05952-6076-T000
NAS 9-4810

LM/AGS FLIGHT EQUATIONS
NARRATIVE DESCRIPTION

25 January 1967

Submitted by: T. S. Bettwy
T.S. Bettwy

Approved by: PD Joseph
P.D. Joseph, Manager
System Analysis and
Software Department

ACKNOWLEDGEMENT

The LM/AGS flight equations explained herein have been developed primarily by the LM Guidance and Navigation Section under the direction of F. A. Evans. Members of this section are E. V. Avery, K. L. Baker, T. S. Bettwy, M. L. Elowitz, R. L. Eshbaugh, C. M. Kawauchi, H. V. Kienberger, and J. T. Wagner. Other members of the System Analysis and Software Department under the direction of P. D. Joseph have contributed at various times in support of this development. This report was written by T. S. Bettwy. K. L. Baker contributed Sections 7.1 and 7.2.

TABLE OF CONTENTS

PART II

DERIVATION AND DISCUSSION

LIST OF FIGURES

1.0 INTRODUCTION

The purpose of this document is twofold:

(1) To present a narrative description of the LM/AGS flight equations as presented in Reference 1 and modified by Reference 2.

(2) To present derivations that are not obvious and additional insight into various portions of the guidance equations.

This document is designed to complement the equation documents (References 1 and 2) and is for information purposes only. There is no plan to update the document for each mission.

The document is basically composed of two parts:

(1) A general narrative description where technical details are kept at a minimum and only logic flow and minor technical points are discussed. This section also contains a functional block diagram of the computer showing inputs and outputs and a description of the coelliptic sequence of maneuvers.

(2) A detailed discussion of various portions of the guidance equations along with derivations of the equations where appropriate. A small section on orbital mechanics is presented in the Appendix so that the derivations will be as self contained as possible. For deeper insight into these areas a list of references has been prepared.

PART I

NARRATIVE DESCRIPTION

2.0 MISSION DESCRIPTION

The mission for which the flight equations are designed is the coelliptic rendezvous of the LM with the CSM, with abort initiation possible at any time after separation of the LM from the CSM. To illustrate all the various maneuvers required, Figure 2.1 shows a coelliptic rendezvous flight profile with abort from the lunar surface.

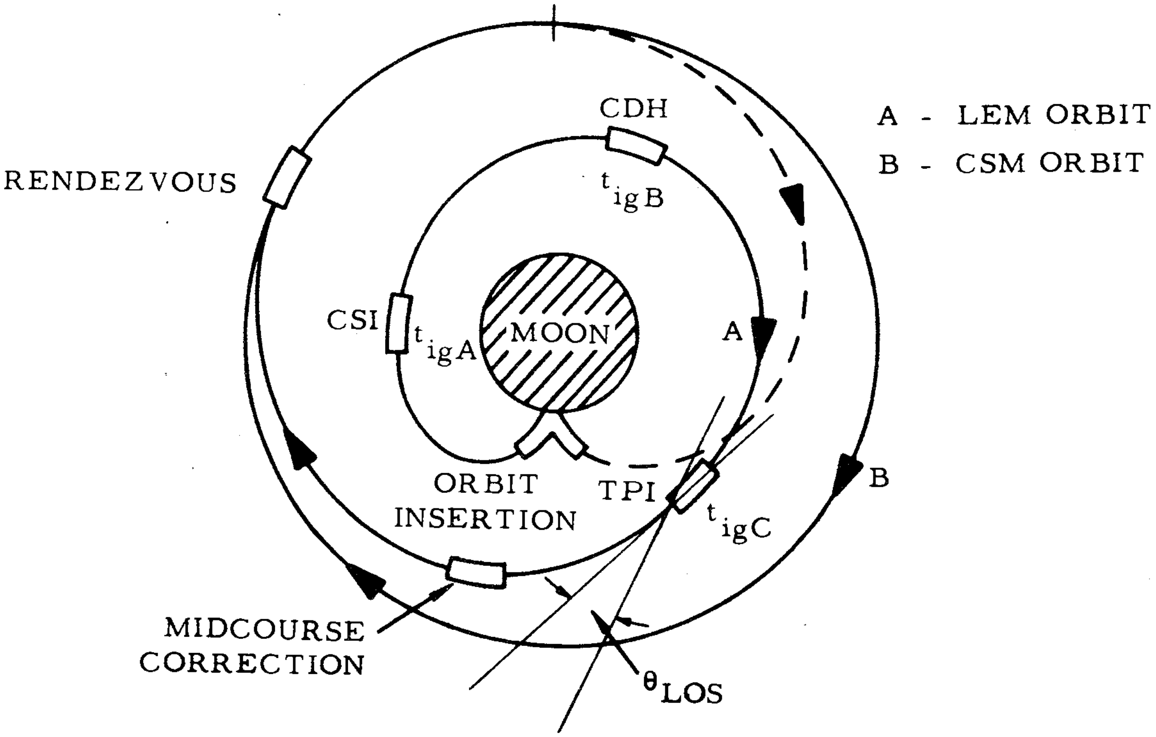


Figure 2.1

Coelliptic Rendezvous Flight Profile

The CSM is in a circular orbit at an altitude of about 80 n mi. 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 absolute time t_{igA}, which brings the LM about 90 deg around the orbit from orbit insertion. At that point the Coelliptic Sequence Initiate (CSI) burn is performed. The magnitude of this CSI burn is determined by an iterative technique whereby trial values of the horizontal burn magnitude are assumed and a resultant error function evaluated. The error function is the difference between the desired central angle difference between the LM and the CSM at the desired time of direct transfer and the computed difference that would be achieved using the trial value of the horizontal velocity. Successive values of the CSI burn are chosen to drive the error function to zero.

The coelliptic (CDH) burn is performed at the predicted time of the LM orbit apofocus (or perifocus). The magnitude of this burn 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 (θ_{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. The midcourse maneuver is obtained in the same manner.

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 direct transfer modes available. Built-in mission planning capabilities are available when these modes are employed. These capabilities and some restrictions are discussed in Section 6.3.7.5.

3.0 COMPUTER INPUTS, OUTPUTS AND OVERALL FUNCTIONAL DIAGRAM

An overall functional block diagram of the LM/AGS is shown in Figure 2.2. Not all functions displayed on this diagram are considered in this document. Only those concerning the flight equations are discussed. Those concerning computer operations such as load and verify routines, telemetry, GSE service routine are not covered in this document.

The functions of the computer program are attitude reference, navigation, abort guidance including attitude commands, Engine ON/OFF command and data display. These functions must be performed based upon the following inputs:

AGS SENSORS

$\Delta\alpha_{xi}$, $\Delta\alpha_{yi}$, $\Delta\alpha_{zi}$ - strapped down gyro inputs along LM X, Y, Z body axis respectively. These inputs are in the form of pulses and 1 pulse represents an angular increment of 2^{-16} radians.

ΔV_{xi}, ΔV_{yi}, ΔV_{zi} - accelerometer inputs along LM X, Y, Z body axis respectively. These inputs are in the form of pulses and 1 pulse represents $\frac{1}{320}$ fps.

PRIMARY GUIDANCE AND NAVIGATION SYSTEM

θ_{pi}, ψ_{pi}, ϕ_{pi} - PGNS 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 - PGNS ephemeris data: L signifies LM, E signifies CSM

RADAR DATA

$\underline{Z}_b^*$ - Z body axis direction cosines stored in the computer when the radar measurement is obtained ($S_{15} = 1$)

R** - Relative range from CSM to LM at the time of taking the radar measurement

$\dot{R}^{**}$ - Range rate between CSM and LM

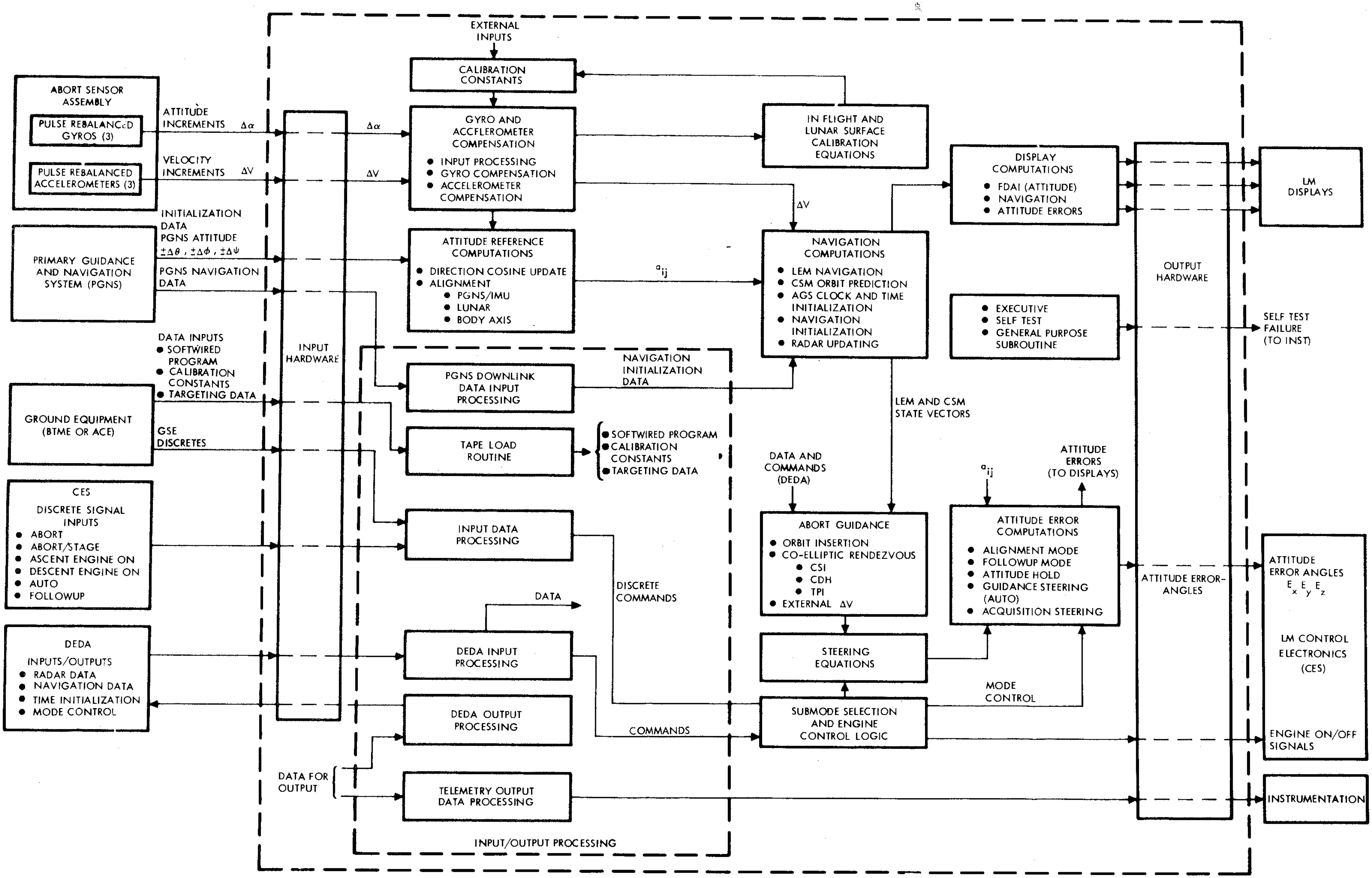


Figure 2.2 **AGS Functional Block Diagram**

DISCRETE SIGNALS

from CES autopilot	β_1	DPS Engine ON ($\beta_1 = 1$) or OFF ($\beta_1 = 0$)
	β_2	APS Engine ON ($\beta_2 = 1$) or OFF ($\beta_2 = 0$)
from Guidance switch on instrument panel	β_3	follow up signal ($\beta_3 = 1$); no follow up ($\beta_3 = 0$)
from 3 state Mode Control switch on instrument panel	β_4	auto ($\beta_4 = 1$); Attitude hold or OFF ($\beta_4 = 0$)
from button on instrument panel	β_5	abort ($\beta_5 = 1$)
from button on instrument panel	β_6	abort stage ($\beta_6 = 1$)
DEDA	K_j^i	input constants
	J^i, J_j^i	input and targeting constants including LM and CSM ephemeris data
	$\dot{R}^{**}$	radar range rate (fps) input as J^{17}
	R^{**}	radar range (n mi) input as J^{18}
	DEDA Control Switches - see below	

DEDA CONTROL SWITCHES

AGS Function Selector (S_{00})

$S_{00} = 0$	Attitude Hold	Inertial Reference Mode
$S_{00} = 1$	Guidance Steering	
$S_{00} = 2$	LOS Acquisition	
$S_{00} = 3$	IMU Align	Align Mode
$S_{00} = 4$	Lunar Align	
$S_{00} = 5$	Body Axis Align	
$S_{00} = 6$	Calibrate	
$S_{00} = 7$	Inflight Accelerometer Calibration	

External ΔV Reference (S_{07})

$S_{07} = 0$ External ΔV vector fixed in $\underline{U}_1$, $\underline{V}_1$, $\underline{W}_1$ local coordinate frame. Sensed velocity increments are set to zero.

$S_{07} = 1$ External ΔV vector fixed in AGS inertial frame. Sensed velocity increments are accumulated.

Guidance Mode Selector (S_{10})

$S_{10} = 0$	Orbit Insertion
$S_{10} = 1$	CSI Maneuver
$S_{10} = 2$	CDH Maneuver
$S_{10} = 3$	Direct Transfer, astronaut inputs T_Δ
$S_{10} = 4$	Direct Transfer, astronaut inputs t_{igc}
$S_{10} = 5$	External ΔV

Engine Select (S_{11})

$S_{11} = 0$	Descent Propulsion System (DPS) or Reaction Control System (RCS)
$S_{11} = 1$	Ascent Propulsion System (APS)

Inflight Self Test Control (S_{12})

$S_{12} = 0$	reset error indications
$S_{12} = 1$	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})

$S_{13} = 0$	No store
$S_{13} = 1$	Store

Navigation Initialization (S_{14})

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

Radar Data (S_{15})

$S_{15} = 0$	no radar data
$S_{15} = 1$	store LM Z axis direction cosines

CDH Apsidal Crossing Selection (S_{16})

$S_{16} = 0$ perform CDH maneuver at first crossing of LM orbit line of apsides

$S_{16} = 1$ perform CDH maneuver at second crossing of LM orbit line of apsides

Elliptical or Circular Orbit Logic (S_{17})

$S_{17} = 0$ elliptical orbit logic
$S_{17} = 1$ circular (near circular) orbit logic

Altitude rate readout (S_{55})

$S_{55} = 0$ select altitude rate readout for lunar missions
$S_{55} = 1$ select altitude rate readout for earth missions

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. Reference 1 contains a short list of these quantities. The overall list will be presented in the Programmed Equations Document.

4.0 COORDINATE SYSTEM DEFINITIONS

Prior to lunar landing the moon centered inertial coordinate system used by the AGS system is defined in the following manner.

> The X axis passes from the center of the moon through the nominal lunar landing site. The Z axis is defined in the downrange direction parallel to the CSM orbit plane and obtained from the equation $\underline{W}_c \times \underline{U}_1$ where $\underline{W}_c$ is the unit vector perpendicular to CSM orbit plane and $\underline{U}_1$ is the unit vector in the direction of the X axis. The third component is defined in the direction of the cross product of the above two.

After landing on the moon and prior to liftoff a new coordinate system is established which is defined exactly as above except that the X axis passes through the nominal launch site rather than the nominal landing site (inertially different because of the moons rotation). If an inflight align is performed the coordinates may be oriented in any orthonormal manner so long as the XZ inertial plane lies within 80° of the LM orbit plane. Reasons for this restriction are discussed in Section 6.3.7.5.1.

The LM vehicle body axes denoted by $\underline{X}_b$, $\underline{Y}_b$, $\underline{Z}_b$ are oriented as shown on Figure 2.3. The direction of the arrows about the vehicle body axes unit vectors indicate positive angular rate (P,Q,R) and displacement (α_X, α_Y, α_Z). The direction of the arrows along the axes indicates the direction of positive translational acceleration and velocity (V_{Xb}, V_{Yb}, V_{Zb}).

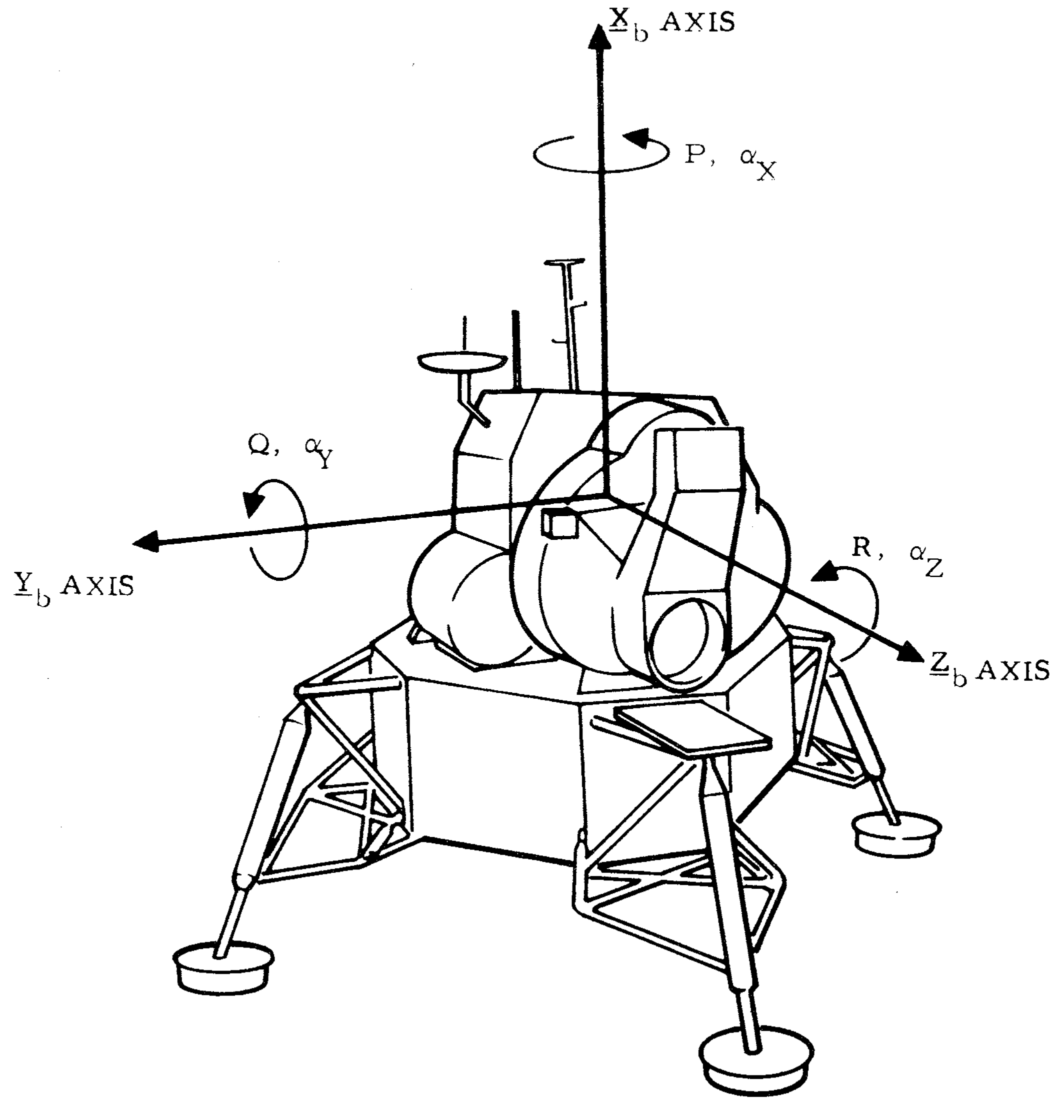


Figure 2.3

Vehicle Reference Axes

5.0 COMPUTATIONAL CYCLES

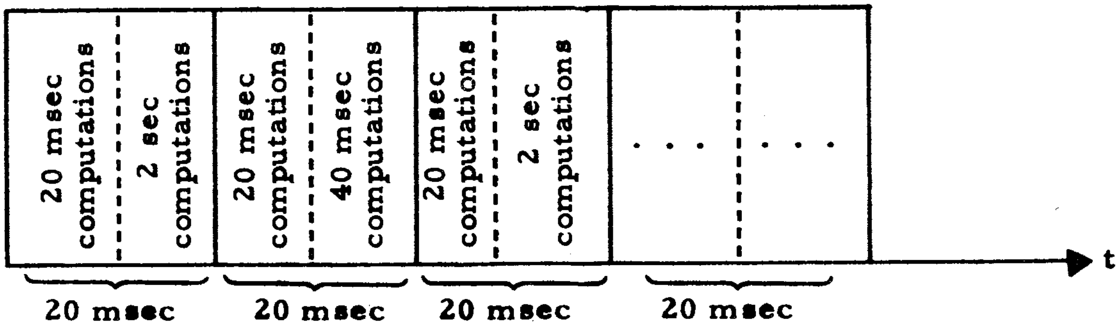
The LM/AGS major computational cycle is 2 sec in duration. In this period of time, the equations update navigation, solve the rendezvous guidance problem, perform inflight self test, and if requested, align the inertial frame of reference. This major 2 sec cycle is broken into 100 segments. Each segment 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; they actually take only about 5 msec. This set of equations is recalculated in the second part of every other 20 msec time segment, thus assuring that this set of equations is recalculated every 40 msec.

The other set of equations must be newly calculated every two 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.

The diagram below illustrates the segmentation of the major two sec computational cycle.



5.1 Timing of the Flight Equation Computations

In the equation flow diagrams of Reference 2, the 20 msec computations, the 40 msec computations, and the 2 sec computations are shown in separate diagrams. Unless otherwise indicated the following figures are those of Reference 2. Figure 3.1 is a systems diagram of the logic connecting the three sets of computations. The important decision blocks in Figure 3.1, from the standpoint of the correct interlacing and proper sequencing of the 40 msec computations and the 2 sec computations are labeled ζ ensures that the 40 msec computations and the 2 sec computations take place in alternating basic 20 msec periods. The 2 sec computations are split among the 50 unused parts. When Cycle B is selected by ζ, the branch control decision block routes the program to the proper set of equations.

Figure 3.1 shows only those portions of the logic and the equations which are hardwired. Included are all of the 20 msec computations, and the 40 msec computations and only portions of the 2 sec computations, including Branch 50. The remainder of the 2 sec computations will be fitted by the programmer into the remaining unused portions of every other 20 msec cycle between Branch 3 and Branch 50.

At the conclusion of the computations in a particular branch (except Branch 50), a dummy variable Q is set to the number of another branch, and then at point Ⓒ of Figure 3.1, the branch control is set to route the program to this block of equations. This means that the next time ζ routes the program to Cycle B, the branch control switch will route the program to the block of equations selected at the end of the previous 2 sec computation. An exception to this sequencing is Branch 50. Counter μ_{10} counts the number of 2 sec computations. Immediately following the block labeled Initial Starting Routine, note that μ_{10} is set equal to zero and branch control set to Branch 50. This ensures that the first branch of the 2 sec computations will be Branch 50. Branch 50 is unique among the 2 sec computations branches in that the first thing done in this routine is to set branch control to Branch 1, and pass from this routine to Branch 1 without going through point Ⓒ. Every time point Ⓒ is passed, the μ_{10} counter is advanced. When $\mu_{10} = 49$, branch control is set to 50. This ensures that no matter what equations of the 2 sec computations are performed in the major 2 sec cycle, the navigation update equations will always be performed at equally spaced intervals. The first Branches 1, 2, and 3, the last Branch, 50, indicate the order which the computations will be performed. As previously stated, all remaining 2 sec computations must be programmed in Branches 4 to 49.

5.2 Basic Block Computation

5.2.1 20-Msec Computations

The most important block in Figure 3.1 from the standpoint of regulating the duration of each basic 20-msec block of computations is decision block Π, which ensures that 20 mseconds of real time have passed before a new basic 20 msecond cycle begins.

The major classification of equations processed in each of the three different computational blocks is listed below.

The 20-msecond computations 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) IMU or body axis align computations

5.2.2 40-Msecond Computations

The 40-msecond computations 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. Hence, the engine OFF command has a 40-msecond timing resolution.

(5) FDAI computations: these are outputs to the astronaut's attitude indicator on the instrument panel.

(6) Lunar align loop: lunar align is a digital servo loop nulling attitude error signal, with a sampling period of 40 mseconds.

(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 Computations

(1) Decision logic for AGS guidance functioning

(2) Navigation updating from previous 2 sec cycle, i.e., the integration of velocity and position (detection of ullage)

(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, 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

(8) Computation of Coelliptic Sequence Initiate (CSI) burn

(9) Computation of Coelliptic Maneuver, CDH Burn

(10) Selection of best of trial values for CSI burn and CDH burn by evaluation of the error function

(11) Calculation of direct intercept transfer orbit (TPI) using p-iteration routine

(12) Computation of an External ΔV maneuver

(13) Computation of velocity-to-be-gained, V_G, and desired vehicle thrust direction

(14) Computation of altitude, altitude rate and body Y-axis velocity every 2 sec and interpolated for output every 200 msec.

6.0 DESCRIPTION OF FLOW CHARTS

This section contains the narrative description of the flight equations. Unless otherwise specified all figures mentioned refer to those of Reference 2. The blocks of equations contained in dashed lines are those programmed in the hardwired memory of the computer.

6.1 20 Msec Computations

6.1.1 Sensor Input Processing

The 20 msec computations begin on Figure 3.2. At the top of the page a check is made on Δt to insure that the equations are calculated every 20 msec. PGNCS Euler Angles, Gyro Data and Accelerometer Data are picked up and appropriately stored. The first calculations are to 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 (1K7) from the number of pulses accumulated and the remainder is the net pulses around a zero acceleration. The net number of pulses is then multiplied by the appropriate scale factor (1K18, 1K20, 1K22) to convert pulses to ft/sec. These scale factors can be varied according to ASA calibration measurements. In a zero g environment the accelerometer may not output exactly 640 pulses so this bias is compensated by the constants 1K19, 1K21, 1K23 for each accelerometer respectively. These numbers are also obtained and varied during calibration.

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

6.1.2 Align or Calibration Mode

If in the align mode the equations determine which specific submode and act accordingly.

6.1.2.1 IMU Align

If an IMU align is being done ($S_{00} = 3$) the logic flow proceeds to the top of Figure 3.5 (point ⒶⒷ) 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 1K25. Operation on these angles yields the desired direction cosines a_{11D}, a_{12D}, a_{13D}, a_{31D}, a_{32D}, a_{33D}. The remaining direction cosines are obtained toward the bottom of the page. Detailed information on the IMU align equations is contained in Section 7.1.2. The quantities $\Delta a_{1j\ rem}$, $\Delta a_{3j\ rem}$, $j = 1, 2, 3$ are then set equal to zero. These quantities will be discussed later. Suffice it to say that when any alignment is performed these equations are zeroed. The logic flow of the equations is then at point BA which is the same as if the inertial reference mode had been selected ($S_{00} < 3$).

6.1.2.2 Lunar Align

If the lunar align mode has been selected ($S_{00} = 4$) on Figure 3.2 the logic flow is to the gyro compensation block. Lunar align is discussed in detail in Section 7.1.3.

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. If the PGNCS is operating, the lunar alignment is performed by going into the IMU align equations.

Basically the lunar align equations compute the [A] transformation matrix (matrix of body axis cosines) which relates the vehicle body axes to the computational inertial reference system, so that the X, Y, Z computational inertial frame coincides with a selenocentric coordinate frame. This selenocentric coordinate system is defined with the X axis along the lunar local vertical positive outward from the lunar center, with the Z axis direction conceptually 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 compensated accelerometer output for leveling and an azimuth reference update constant δ_Δ for the azimuth reference. These equations are computed in the 40 msec computational cycle and are shown on Figure 3.8. Detailed discussion of the equations is presented in Section 7.1.3.

For lunar align to function properly the direction cosines must have been stored soon after LM touchdown on the lunar surface by the setting of $S_{13} = 1$.

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 AC on Figure 3.5 where the direction cosines are initialized as $a_{11} = a_{33} = 1$, $a_{12} = a_{13} = a_{31} = a_{22} = 0$. The remaining direction cosines are computed from these near the bottom of the page. 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 that the XZ inertial plane lies within 80° of the LM orbit plane. Detailed discussion of the mode is contained in Section 7.1.1. The logic flow again arrives at point BA of Figure 3.5.

6.1.2.4 Calibration Submodes

If the equations are in a calibrate mode ($S_{00} = 6,7$) then the lunar surface flag δ_{21} is checked to see if the LM is on the lunar surface ($\delta_{21} = 1$) or in flight ($\delta_{21} = 0$). In either event the logic flow arrives at the gyro compensation block of equations and as in the inertial reference mode to point $\textcircled{BA}$ on Figure 3.5. The actual calibration equations and techniques are discussed in Section 7.2 of this report. S_{00} should NOT be set to 7 when on the lunar surface.

6.1.3 Inertial Reference Mode

If the equations are in the inertial reference mode the logic flow is to the gyro compensation block at the bottom of Figure 3.2. 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 axis except as noted below.

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

$$\Delta\alpha_{x\ rem} = \Delta\alpha_{x\ rem} + K_1^1 + K_2^1 K_3^1\,(\Delta\alpha_{xi} - K_7^1) + K_{14}^1\,\Delta V_x$$

$$\Delta\alpha_x = K_2^1\,(\Delta\alpha_{xi} - K_7^1) + \Delta\alpha_{xA} + \Delta\alpha_{x\ rem|\geq 2^{-16}}$$

$$\Delta\alpha_{x\ rem} = \Delta\alpha_{x\ rem} - \Delta\alpha_{x\ rem|\geq 2^{-16}}$$

These equations do the following: The raw X axis gyro output ($\Delta\alpha_{xi}$) 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 ($\Delta\alpha_{xi} - K_7^1$) by the constant 1K2 converts the pulse count to radians. Multiplication of the angular increment 1K2 ($\Delta\alpha_{xi} - K_7^1$) by the constant 1K3 forms a correction for the attitude rate scale factor error of the X gyro. This quantity has added to it the gyro drift compensation constant 1K1 and the quantity 1K14 (ΔV_x). This latter 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 do not appear in the Y or Z gyro equations because error analyses have indicated them to be unnecessary. $\Delta\alpha_{x\ rem}$ is then formed as the summation of all the terms just discussed and that part of $\Delta\alpha_{x\ rem}$ of the previous computation cycle that did not exceed 2^{-16} radians. Note that $\Delta\alpha_{x\ rem}$ appears as a summation of correction terms to the yaw gyro input. To obtain the compensated gyro input used in the navigation equations ($\Delta\alpha_x$), the part of $\Delta\alpha_{x\ rem}$ that exceeds 2^{-16} radians is added to the raw gyro input converted to radians $[K_2^1(\Delta\alpha_{xi} - K_7^1)]$. In addition the alignment error $\Delta\alpha_{xA}$ 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 $\Delta\alpha_{x\ rem}$ term is set up for the next computing cycle by subtracting from $\Delta\alpha_{x\ rem}$ that part of $\Delta\alpha_{x\ rem}$ greater than 2^{-16}. The flow logic then proceeds to (AA) on Figure 3.3.

On Figure 3.3 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. Scaling at 2^{-6} is necessary to obtain the required dynamic range of rotation about each axis (±25°/sec). Switch-over of the scaling occurs at approximately 5°/sec. The LM usually rotates at a rate less than 5°/sec so that the desired precision is obtained. The logic flow then proceeds to (AD) on Figure 3.4.

Figure 3.4 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 improve the computational accuracy due to the higher scaling used on Figure 3.3. E_1, F_2 and E_{13} are then set to zero.

Regardless of the mode of the equations (Figure 3.2) all logical paths arrive at the point (BA) on Figure 3.5. Here 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} which were evaluated previously in a manner depending upon the mode of the equations. With these direction cosines the velocity increments obtained along LM body axis (from the accelerometers that are mounted on the body axis) are transformed to X,Y,Z inertial coordinates. These velocity increments are denoted by ΔV_{xs}, ΔV_{ys}, ΔV_{zs} respectively. This group of equations are the last computed in the 20 msec subcycle. At the bottom of Figure 3.5 the decision is made to proceed to the 40 msec computations or to the 2 second 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 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 seconds

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

$\underline{V}_g$ 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 seconds.

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

The following chart 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.

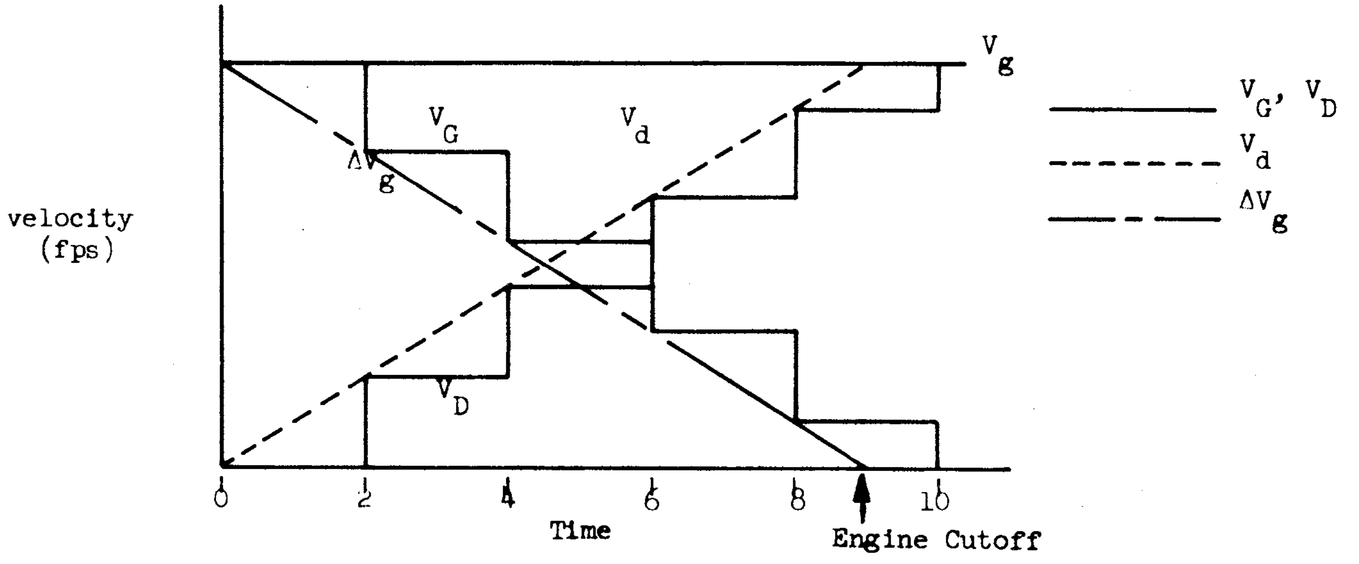


FIGURE 2.4

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 ($\Delta\underline{V}_g$) rather than information that may be up to two seconds old ($\underline{V}_G$). Note that since $\underline{V}_D$ should increase by the same amount that $\underline{V}_G$ decreases every two seconds the quantity $\underline{V}_g$ remains constant. Thus, the velocity remaining to be gained (calculated each 40 msec) is readily computed as

$$\underbrace{\Delta\underline{V}_g}_{\text{velocity to-be-gained}} = \underbrace{\underline{V}_g}_{\text{constant}} - \underbrace{\underline{V}_d}_{\text{velocity already gained updated every 40 msec}}$$

In the first block on Figure 3.6, the quantity V_{dx} (for example) is computed as

$$V_{dx} = V_{dx} + \Delta V_{x,n} + \Delta V_{x,n-1}$$

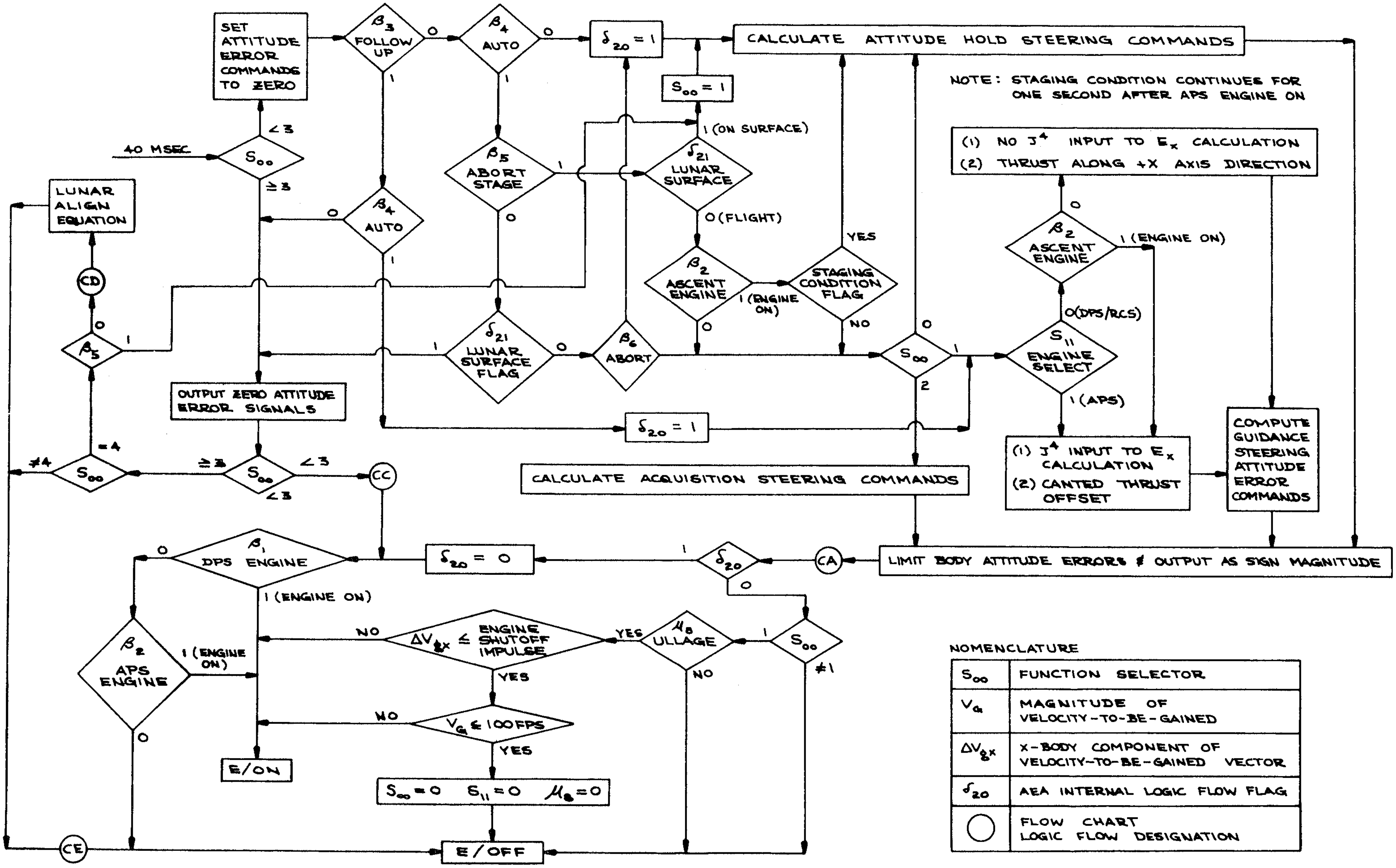
which in words means: set the x component of velocity already gained equal to that gained up to 40 msec ago plus the velocity gained in the latest 20 msec computing cycle plus the velocity gained in the second previous 20 msec cycle.

6.2.1 Establishing Vehicle Status

Following these calculations, the status of the β discretes is checked. Every 40 msec the β discretes 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 δ_2 flag which is equal to one if the descent section has been staged. If the δ_2 flag has not been set to 1, then the β_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 δ_2 is set to 1, meaning the descent section has staged and δ_{21} is set to 0, meaning the LM is not on the surface of the Moon. S_{00} is set to 1 so that the vehicle goes into a guidance steering mode of operation. This is done for the following reason. Prior to lift-off from the lunar surface, a lunar align may be performed. It is desirable to perform this align until nominal liftoff time. Thus, instead of making the astronaut take the equations out of the lunar align mode (switch S_{00} from 4 to 1) it is done automatically. This is also a safeguard against an undesirable engine cutoff since the equations command engine OFF (explained below) when $S_{00} \neq 1$.

The remainder of the flow in the 40 msec subcycle is functionally displayed in Figure 2.4 (this document). This diagram is very useful for considering the result of various switch settings.


SET ATTITUDE ERROR COMMANDS TO ZERO
β_3 FOLLOW UP
β_4 AUTO
$\delta_{20} = 1$
CALCULATE ATTITUDE HOLD STEERING COMMANDS
NOTE: STAGING CONDITION CONTINUES FOR ONE SECOND AFTER APS ENGINE ON
$S_{00} = 1$
1 (ON SURFACE)
40 MSEC
S_{00}
< 3
≥ 3
β_5 ABORT STAGE
δ_{21} LUNAR SURFACE
0 (FLIGHT)
(1) NO J^4 INPUT TO E_x CALCULATION
(2) THRUST ALONG +X AXIS DIRECTION
LUNAR ALIGN EQUATION
β_4 AUTO
β_2 ASCENT ENGINE
1 (ENGINE ON)
YES
STAGING CONDITION FLAG
NO
CD
β_5
δ_{21} LUNAR SURFACE FLAG
β_6 ABORT
S_{00}
S_{11} ENGINE SELECT
0(DPS/RCS)
1 (APS)
OUTPUT ZERO ATTITUDE ERROR SIGNALS
$\delta_{20} = 1$
(1) J^4 INPUT TO E_x CALCULATION
(2) CANTED THRUST OFFSET
COMPUTE GUIDANCE STEERING ATTITUDE ERROR COMMANDS
= 4
≠ 4
CC
CALCULATE ACQUISITION STEERING COMMANDS
β_1 DPS ENGINE
$\delta_{20} = 0$
δ_{20}
CA
LIMIT BODY ATTITUDE ERRORS & OUTPUT AS SIGN MAGNITUDE
1 (ENGINE ON)
β_2 APS ENGINE
$\Delta V_{gx} \leq$ ENGINE SHUTOFF IMPULSE
μ_8 ULLAGE
$V_G \leq 100$ FPS
E/ON
$S_{00}=0$ $S_{11}=0$ $\mu_8=0$
E/OFF
CE
0
1
2

NOMENCLATURE

S_{00}	FUNCTION SELECTOR
V_G	MAGNITUDE OF VELOCITY-TO-BE-GAINED
ΔV_{gx}	X-BODY COMPONENT OF VELOCITY-TO-BE-GAINED VECTOR
δ_{20}	AEA INTERNAL LOGIC FLOW FLAG
○	FLOW CHART LOGIC FLOW DESIGNATION

Figure 2.5 - AGS Logic Flow Diagram

Regardless of the logic flow path just taken, a check is made on S_{00} to see if the system is in an inertial reference mode ($S_{00} < 3$) or in the align and calibrate mode ($S_{00} \geq 3$). If in the align and calibrate mode, zero attitude errors are output. If the specific mode is lunar align ($S_{00} = 4$) a check is made on β_5 (abort stage). If an abort stage is occurring, the equations switch out of the lunar align mode by setting S_{00} to 1 and go to the attitude hold equations. If $\beta_5 = 0$, the logic flow proceeds to Figure 3.8 where the lunar align calculations are done. The logic flow then proceeds to point (CE) on Figure 3.7 where the engine discrete is set to OFF. (CE) is also the point to which the logic flow proceeds if any of the other align or calibrate modes are operating. The check of S_{00} against 4 at the bottom center of Figure 3.6 determines which of the above paths are taken.

6.2.3 Steering Modes

Prior to discussing the logic in the center of Figure 3.6, the various steering modes are considered. The result of all steering mode calculations are attitude errors about the LM body axes denoted by E_x, E_y and E_z. Derivation of the equations for these three quantities (in the various steering modes) is contained in Section 7.4.1 of this report. A general description of the equations is given here.

6.2.3.1 Attitude Hold

The object of the "attitude hold" equations is to generate steering commands such that the vehicle maintains the inertial attitude existing when the attitude hold mode is first entered. This is done by utilizing the δ_5 check at the top center of Figure 3.6. When the attitude hold mode is first entered, δ_5 equals zero. This causes the desired pointing directions $\underline{X}_D$ and Z_D to be established as the present pointing direction of the vehicle $\underline{X}_b$ and $\underline{Z}_b$ respectively. In addition, the flag δ_5 is set equal to 1 so that in the future the operations of setting $\underline{X}_D = \underline{X}_b$ and $\underline{Z}_D = \underline{Z}_b$ are bypassed. 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 the first time through the attitude hold computations.

6.2.3.2 Guidance Steering

The object of the "guidance steering" mode of attitude control is to orient the LM thrust vector along the desired thrust vector. The LM thrust vector is actually oriented differently depending upon which engines are being used. The RCS engines are mounted along the body axes so that 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 approximately 5 degrees downward (toward the positive Z body axis) in pitch and 2 degrees toward the positive Y body axis in yaw. Thus, steering commands must be generated depending upon which engine is used. These equations are derived in Section 7.4.1 and are

$$E_x = - \underline{W}_c \cdot \underline{Z}_b$$

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

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

when the DPS or RCS engines are used. These equations appear at the bottom of Figure 3.6 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 center of Figure 3.6) 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 the angular momentum vector of the CSM orbit pointing toward the lunar surface (down).

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

$$E_x = - \underline{W}_c \cdot \underline{Z}_b - J^4$$
$$E_y = K_7^4 (\underline{X}_b \cdot \underline{X}_{bD}) - (\underline{Z}_b \cdot \underline{X}_{bD})$$
$$E_z = -K_8^4 (\underline{X}_b \cdot \underline{X}_{bD}) + (\underline{Y}_b \cdot \underline{X}_{bD})$$

In the equation for E_x the DEDA input constant has been added to allow S band communication during the orbit insertion phase of the lunar mission regardless of the landing site chosen. J^4 is entered in degrees and should be limited to less than 40° due to accuracy limitations. A positive value of J^4 will drive the positive LM Z body axis toward the negative CSM angular momentum vector. Care should be exercised in using this parameter if the desired thrust direction of the LM is nearly colinear with the CSM angular momentum vector. In this situation the attitude error equations (for nonzero values of J^4) will try to drive the Z body axis to a physically unrealizable position thus causing the vehicle to continuously roll about the X body axis. If this situation occurs at all it probably would be in the final phase of the mission where a value of $J^4 = 0$ should be utilized.

The logic check on the switch S_{11} followed by the check on β_2 at the bottom of Figure 3.6 appears at first glance to be redundant since S_{11} should be switched to 1 when the APS engines are used ($\beta_2 = 1$). This logic has been inserted because β_2 is 0 until the APS engine is turned on at which time it is set to 1. Thus up until this time the desired steering would not consider the canted engine. In order to overcome this the switch S_{11} has been inserted so that the correct steering will be done prior to the initiation of the APS burn. For long APS burns the insertion of the S_{11} switch would not be necessary; however for short duration burns inclusion of this switch is mandatory.

6.2.3.3 Acquisition Steering

The purpose of the "acquisition steering" mode ($S_{00} = 2$) of attitude control is to point the positive Z body axis of the LM in the computed direction of the CSM. This mode is used prior to taking radar data. The error equations are computed as

$$E_z = \underline{W}_c \cdot \underline{X}_b$$

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

$$E_x = - \underline{Y}_b \cdot \underline{Z}_{bD}$$

where $\underline{Z}_{bD}$ is the desired pointing direction of the LM Z body axis. In this mode then, the Z axis is oriented in the direction of the CSM and 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 3.6 is now considered. At the upper left hand side of the page the follow-up flag β_3 is checked. Consider the situation where $\beta_3 = 0$. A check is then made on the "Auto" discrete β_4. If $\beta_4 = 0$ (attitude hold) the flag δ_{20} is set equal to 1 and the attitude hold equations are then computed. δ_{20} is used on Figure 3.7 to control the logic flow. The result of δ_{20} being set equal to 1 maintains the Engine ON/OFF discrete in the status determined by β_1 or β_2. That is, if either β_1 or β_2 equals 1 the engine discrete is set ON, if not then the engine discrete is set OFF.

Back on Figure 3.6 if the β_4 flag equals 1 (rather than 0 as just discussed) then the "abort stage" discrete β_5 is checked. Consider first the case where no abort stage is commanded ($\beta_5 = 0$). Immediately following this the lunar surface flag δ_{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 $\textcircled{CC}$ on Figure 3.7 where again the engine discrete is determined by the status of β_1 and β_2. If not on the lunar surface ($\delta_{21} = 0$) the abort discrete (β_6) is checked. If $\beta_6 = 0$ then again δ_{20} is set to 1 and the logic flow proceeds to attitude hold. If the abort discrete is 1 then either attitude hold, guidance steering or acquisition steering is done depending upon the setting of S_{00}.

The only other alternative remaining is if an abort stage command, i.e., $\beta_5 = 1$. This appears in the center of Figure 3.6. If the LM is on the lunar surface, δ_{00} is set to 1 and 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 3.6 is exercised as discussed above. That is, when the ascent engine comes on ($\beta_2 = 1$), δ_{21} is set to zero so that following the $\beta_5 = 1$ check at the center of the page, the check on δ_{21} now sends the logic flow downward toward the β_2 check. Since β_2 was set to 1, the logic flow proceeds to the u_6 counter check. u_6 is incremented by 1 every 40 msec. The vehicle stays in attitude hold until u_6 exceeds 4K32 at which time the logic flow proceeds to the guidance steering equations. If the abort stage had been set in flight with the ascent engine off ($\beta_2 = 0$), then the u_6 logic would be bypassed (until $\beta_2 = 1$) and the nominal guidance steering continued. The purpose of the u_6 logic is to allow the vehicle to automatically be in attitude hold as the vehicle lifts off the lunar surface and during staging. After a prescribed amount of time (controlled by 4K32), the vehicle will orient to the attitude defined by the guidance steering equations.

Consider now again the check on β_3. In this instance, let the AGS be in follow-up ($\beta_3 = 1$). If $\beta_4 = 0$ (altitude hold), then the attitude errors are zeroed and the logic flow proceeds to CC on Figure 3.7. If $\beta_4 = 1$ (auto), then δ_{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.

6.2.5 Engine ON/OFF Logic

As already indicated, there are four exits from Figure 3.6. These have all been discussed above with the exception of the exit to CA when δ_{20} is zero. CA appears on Figure 3.7. If δ_{20} is zero and the mode of the computer (δ_{00}) is not in "guidance steering", then the engine is commanded OFF. If, however, $\delta_{00} = 1$ (guidance steering) a series of checks are made to determine the proper status of the engine discretes. First, the ullage counter is checked. If this counter has not accumulated to the value 1K9, the engine discrete is set OFF. If $u_8 \geq$ 1K9, 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 4K25 (engine shutdown impulse), the engine discrete is set ON. If less than 4K25, a check on total velocity-to-be-gained is made against 4K26. The total velocity-to-be-gained is here denoted by ΔV_G. This is a dummy variable set equal to V_G near the end of each 2 second computing increment. If $\Delta V_G \geq$ 4K26, the engine discrete is set ON. If less than 4K26, the attitude hold mode is entered, S_{11} is set to zero, the ullage counter is reset 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.

The value of 4K26 has been tentatively set at 100. The reason for the check of ΔV_G against 4K26 is to guard 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 ΔV_{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 ΔV_G against 4K26 insures that the desired conditions are obtained.

The remainder of Figure 3.7 has to do with outputting DEDA words and generating quantities E_1, E_3, E_{13}. These three terms are used for the orthonormal corrections to the direction cosine matrix as derived in Section 7.3.1 of this report. The remainder of the 40 msec computations are done on Figure 3.9 where sin α, cos α, sin γ, cos γ are computed for the FDAI and sent to the D/A converter.

6.3 Two Second Computations

The equations utilized in the 2 second computation interval are selected by the logic on Figure 3.27 called the "Executive Branch". The selection of equations depends primarily upon switch S_{14} which has the following settings.

$$S_{14} = \begin{cases} 0 & \text{Initialization Complete} \\ 1 & \text{Initialize LM and CSM via Downlink} \\ 2 & \text{Initialize LM via DEDA} \\ 3 & \text{Initialize CSM via DEDA} \end{cases}$$

If the switch setting for S_{14} is either 0 or 3 the equations utilized in the 2 second computations are those beginning on Figure 3.12. This is the usual mode of operation. If however, S_{14} is 1 or 2 then only the equations that appear on Figures 3.10 and 3.11 are solved in the 2 second cycle. In other words no guidance computations are done in the 2 second computation interval when S_{14} is either 1 or 2.

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 (explained in mathematical detail in Section 7.4.2 of this report) has the capability of accepting position and velocity of a vehicle at any time, say t_E, and determining the position and velocity of the vehicle at any other time, say $t_E + T_i$. 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 time t_E and $\underline{V}_E$ is the CSM velocity vector at the epoch time t_E. Thus, if it is desired to know the position and velocity of the CSM at say 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 point is greater than 1 orbit old. In that case the epoch position and velocity is updated (bootstrapped, Figure 3.13) one full orbital period. This is done so that the quantity time does not exceed 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. Thus 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

Reference is made to Figure 3.10. If S_{14} is 1 then a CSM and LM ephemeris update is to be performed utilizing information obtained from the LGC downlink via DSKY command. This information is converted to the correct format and stored in the DEDA cells. The quantity t_b (to be discussed below) is set to zero. If $S_{14} = 2$ then the LM ephemeris is to be updated via DEDA inputs in the form of constants 1J1 thru 1J7. No new CSM data is obtained at this time.

S_{14} is automatically set to zero so that in the next 2 second computing increment the normal guidance equations will be utilized. Also the present time, t, is incremented by 2 seconds. The following quantities are obtained for the LM orbit from the "orbit parameters subroutine" that appears on Figure 3.15. This subroutine computes orbital elements from equations described in Appendix A of this document.

r - distance of LM from the center of moon

α - semi major axis of LM orbit

η - LM mean orbital rate

C - product of LM eccentricity and cosine of eccentric anomaly

S - product of LM eccentricity and sine of eccentric anomaly

Definition of these terms is contained in Appendix A.

Figure 3.11 indicates how the LM ephemeris data is updated to the present time. First the quantity T_i is calculated which is the time increment from when the data was valid to the present time. If the vehicle is on the lunar surface ($\delta_{21} = 1$) then the position vector is updated simply as

$$\underline{r} = \underline{r}_o + \underline{V}_o T_1$$

$$\underline{V}_o = \underline{V}_o$$

If, however, δ_{21} is 0 then the present position and velocity of the LM is obtained via the ellipse predictor as indicated previously. Following these updating calculations several quantities are calculated for use in the next 2 sec computing increment.

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

6.3.4 Navigation Equations Description

If S_{14} is neither 1 nor 2 then the two second computations begin on Figure 3.12. The purpose of the equations on this page is to obtain the LM position and velocity based upon the sensed accelerometer ΔV's and attitude, accumulated during the 20 msec computation cycles.

If the LM is on the lunar surface ($\delta_{21} = 1$) the accumulated velocities ($\underline{V}_d$) and the gravity integral $\underline{\Delta IG}$ are set equal to zero and the "navigation update" equations 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 seconds 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 seconds in the past. The thrust acceleration, a_T, is checked against constant 4K35 (.1 fps^2). If $a_T \geq$ 4K35, the ullage counter (μ_8) is increased by 1. If $a_T <$ 4K35 the ullage counter (μ_8) is set to zero. ΔV^*_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 1K35. If $\Delta V^*_S <$ 1K35 it is assumed that no thrusting has occurred and the accumulated sensed velocities (accelerometer biases) are set to zero. Moreover $\underline{V}_d$ is set equal to $\underline{V}_D$.

The first equation in the group of equations marked "Navigation Update" on Figure 3.12 pertains only to the "external ΔV" guidance mode of operation and will be discussed completely below. Suffice it here to note that $\underline{\Delta q}_s$ is the accumulated thrust velocity in inertial coordinates and the equation updates the value every 2 seconds. $\underline{V}_D$ is set equal to $\underline{V}_d$ for use in the calculation of a_T in the next 2 sec computing increment. Next the LM velocity vector is updated to the present time by the relation

$$\underline{V} = \underline{V} + \underline{\Delta IG} + \underline{\Delta V}_S$$

where again $\underline{\Delta V}_S$ is the accumulated sensed velocity in inertial coordinates over the last two seconds and the term $\underline{\Delta IG}$ takes into account the effect of gravity on velocity. $\underline{\Delta IG}$ is equal to 2 (since 2 sec is the computing increment) times the average value of acceleration due to gravity in the previous computing cycle which is assumed to be the value of acceleration one second ago. Pictorially

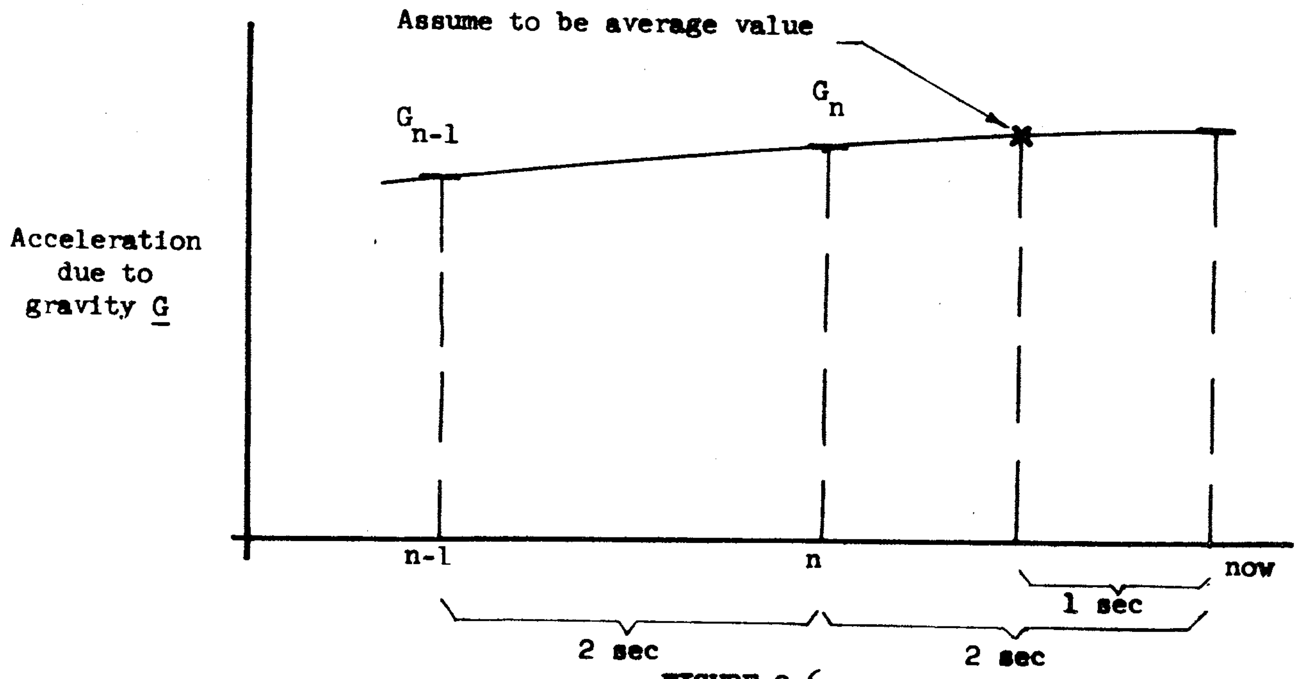

Assume to be average value
G_{n-1}
G_n
Acceleration due to gravity $\underline{G}$
n-1
n
now
1 sec
2 sec
2 sec

FIGURE 2.6

EVALUATION OF GRAVITY VECTOR

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

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

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

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

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 two seconds. 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 $\dot{r}$ is calculated for later guidance purposes and $\dot{h}$ is calculated for display purposes. In addition time is incremented by 2 seconds. The logic flow then proceeds to Figure 3.13 where the CSM orbit parameters are computed from the ephemeris data.

6.3.5 CSM Orbit Parameters

Recall that if $S_{14} = 3$ a new CSM ephemeris point is to be input via the DEDA. At the top of Figure 3.13 a check on S_{14} is made against 3. If it is 3 the new information is entered and the quantities t_b and S_{14} are zeroed. The same orbit parameters as determined for the LM on Figure 3.10 are not obtained for the CSM from the "Orbit Parameter Subroutine" on Figure 3.15. 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 1 CSM orbital period then the epoch time is increased by 1 orbit. This is accomplished at the bottom of Figure 3.13 where T_{CSM} is the CSM orbital period.

The logic flow then proceeds to Figure 3.14 where the calculations are done to call the "ellipse prediction subroutine" and obtain the CSM position and velocity ($\underline{r}_c$ and $\underline{V}_c$) value at the present time. These quantities are then used for the radar filter, Figure 3.16.

6.3.6 Radar Filter

Prior to discussing the equation flow logic of the radar filter a general description of the radar operation procedure is discussed. First, it must be understood that the radar and the AEA are separate equipment units and in no way can radar information be accepted automatically by 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 S_{15} is set to 1. At this same time the astronaut must read the range tape meter and within 30 seconds enter radar range via the DEDA. This is entered as manual constant J^{18}. Explicitly then the procedure is as follows:

(1) The astronaut sets $S_{00} = 2$. 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 angle on his attitude display are zeroed. 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 "follow-up" mode ($\beta_3 = 1$).

(3) When the gimbal angles have been zeroed the astronaut enters via the DEDA the appropriate code that sets $S_{15} = 1$.

(4) "Radar Range" is read from the tape meters and entered as J^{18} via the DEDA. This entry should be made within 30 seconds after taking the radar point.

The radar filter (Figure 3.16) is a simplified Kalman filter. Derivation of the equations are contained in Section 7.3.2 of this report. Briefly, the filter operates by assuming that knowledge of the CSM position and velocity is perfect. LM position and velocity is then updated based upon the navigation estimate of LM present position and velocity, the radar measurement, and the relative "credibility" of these two quantities. This "credibility" is expressed in the equations by the terms P_{11}, P_{12}, P_{22} and σ^2. Definition of these terms are:

P_{11} - navigation position error variance

P_{12} - navigation position, velocity error covariance

P_{22} - navigation velocity error variance

σ^2 - sum of the navigation position and radar measurement error variances

To see how the equation logic works reference is made to Figure 3.16. In the first block of computations the relative range vector $\underline{R}$ between the CSM and LM is computed along with its magnitude. The desired Z body axis direction ($\underline{Z}_{bD}$) is also computed. In addition relative range rate, $\dot{R}$, is computed for use in the range rate filter (described below) and for display purposes. Bypass the check on $\dot{R}^{**}$ for the moment and proceed to the check on δ_{12}. δ_{12} is normally zero except when radar data is being taken. That is, when S_{15} is set to 1 (direction cosines are to be stored) δ_{12} is also set to 1. This is done in the DEDA routine and is indicated at the bottom of Figure 4.6. When δ_{12} is zero and no radar updating is being done, R** (the radar range measurement) is set to zero. The logic flow then proceeds to point (E) on Figure 3.17.

Assume, however, that a radar measurement is to be made. Assume also that the astronaut has lined up the Z body axis of the LM pointing toward the CSM and he enters the code setting $S_{15} = 1$. At this time he notes the radar range on the tape meter. δ_{12} is set to 1 automatically and $S_{15} = 1$. Following the flow diagrams it is seen that a quantity Δt is computed. This quantity is the time elapsed since the previous radar point. Assume this to be the first radar point taken. Then t_1 is zero and Δt is some large number. The calculations of P_{11}, P_{12} and P_{22} have no meaning since they have never been initialized. In this case Δt overflows (in the check following the P_{11}, P_{12}, P_{22} calculations) and the quantities P_{11}, P_{12} and P_{22} are initialized. If this was not the first radar point then the calculations of P_{11}, P_{12}, P_{22} would be valid. These calculations propagate the error covariance from the time of the previous radar point to the present time. In this case, assuming the time since the last radar measurement is not unduly large (less than 2^{10} seconds), no overflow occurs and the block at the bottom center of the figure is entered. Here the weighting is determined (W_1 and W_2) based upon range and the magnitude of P_{11} and P_{12}. In addition the time of the radar measurement is saved as t_1 (for use in the calculation of Δt at the next time a measurement is taken) and S_{15} is set to zero. For later purposes the relative range vector $\underline{R}$ is stored as $\underline{R}^*$ and W_1 is stored as W_1'. This completes the radar computations this 2 second cycle and the logic flow as previously proceeds to the block where R** is set equal to zero. Note that the navigation update based upon the radar measurement has not been performed yet. This is done in the 2 second computer cycle that the astronaut enters radar range as J^{18} into the DEDA.

Each succeeding two second computing increment (until the navigation update is made) the logic path followed after entering at the top of the page is that of $\delta_{12} = 1$ and $S_{15} = 0$. Here W_1 is updated to take into account the fact that the navigation update is not being performed at the time of taking the radar measurement. If the range measurement has not been entered then the logic flow leaves the radar filter page as described above. In the cycle that the radar measurement, R**, is entered to the computer, R** is greater than zero and the computations at the right center of the page are performed. This is where the navigation updating occurs. The quantity $\underline{\delta r}$ is computed which is the difference between the estimated relative range vector at the time the radar measurement was taken and the actual radar measurement. This quantity is then used in conjunction with the filter weights W_1 and W_2 to update the present estimate of LM position and velocity. This is done by the equations

$$\underline{r} = W_1\underline{\delta r} + \underline{r}$$

$$\underline{V} = W_2\underline{\delta r} + \underline{V}$$

Note that the radar measurement taken at some prior point in time is being used to update the present LM position and velocity. Of course the accuracy of this technique diminishes as the time between taking the radar measurement and performing the update increases. Studies are presently being conducted on the maximum allowable duration between these two events. Until these results are obtained a time limit of 30 seconds has been placed on this procedure. After the updating has occurred the magnitude of the error covariances are decreased since the estimate of position and velocity has been improved. Moreover δ_{12} is set to zero so that the radar filter is not entered again until another radar point is to be taken.

The astronaut also has the capability of entering into the computer the radar range rate obtained from a tape meter. This entry does not need to be associated with the range measurement discussed above. When $\dot{R}^{**}$ is entered the LM velocity vector $\underline{V}$ is updated in the logic path followed when $\dot{R}^{**} \neq 0$. The logic flow then leaves the radar filter computations as usual.

6.3.7 Guidance Equations

After leaving the radar filter page, the logic flow proceeds to Figure 3.17 where the actual guidance equation calculations begin.

Entering at the top of Figure 3.17 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 pericynthion of the LM orbit. This quantity later has the value of mean lunar radius, J^5, subtracted from it. q then 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 value 2^{20}.

Immediately following this several unit vectors are generated. Based upon the present position of the LM ($\underline{U}_1 = \underline{r}/|\underline{r}|$) and the unit vector, 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. From $\underline{U}_1$ and $\underline{V}_1$ then the third unit vector $\underline{W}_1$ for the LM is calculated. The CSI and CDH maneuver are both "in-plane" maneuvers so the quantities V_y and y are set equal to zero. These quantities are calculated where needed in the other maneuvers. A dummy variable $\dot{r}_A$ is set equal to $\dot{r}$ (present LM altitude rate) and the flag δ_{10} is set to zero. This flag controls the logic flow later in the equations and is to be zero in all guidance modes except "orbit insertion" where it is set to the value 1. An angle, ξ, is computed for display purposes which is the angle between the Z body axis and the LM local horizontal plane.

Following these computations, a series of checks are made on S_{10} to determine the guidance mode of operation. The various modes of S_{10} are as follows:

$$S_{10} = \begin{cases} 0 & \text{orbit insertion} \\ 1 & \text{CSI} \\ 2 & \text{CDH} \\ 3, 4 & \text{direct transfer (TPI, MCC)} \\ 5 & \text{external } \Delta V \end{cases}$$

The first check on S_{10} is against the value 5. If S_{10} is 5, the external ΔV mode of operation is entered.

6.3.7.1 External ΔV

When S_{10} is 5, the "external ΔV" mode is entered and the first check in this path is against the quantity S_{07}. The first time down this path, S_{07} must be 0. To insure that this is so, S_{07} is set to zero when any other guidance mode is selected. This is the equation just to the left of the check of S_{10} against the value 5. Assuming S_{07} is zero, the next check made is on the ullage counter. The ullage counter must also have an initial value of 0 for the "external ΔV" mode of operation to work properly. This means that the vehicle must not be thrusting in the positive X body axis direction the cycle the external ΔV mode is entered. The next block of equations establish the external ΔV maneuver to be performed. This block is bypassed once thrusting has started. Prior to discussing this initialization, the two types of "external ΔV" maneuvers possible 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 Section 6.2.5). The maneuver will be performed separately along each axis. If thrusting is done first along the positive X axis, no further action need be taken by the astronaut to get the "external ΔV" mode operating correctly. This is so because when thrusting along the positive X axis occurs the ullage counter increments itself and the external ΔV initialization equations are bypassed as desired (S_{07} is set to 1). If, however, thrusting is performed along one of the other axes or along the negative X body axis, then S_{07} must be set to 1 just prior to the initiation of the maneuver by the astronaut. In this situation then the "external ΔV" initialization equations are again bypassed as desired when the maneuver has begun.

(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.

In the "external ΔV" mode the desired maneuver is characterized by the values of the input constants 28J1, 28J2 and 28J3. A value of 28J1 characterizes the velocity-to-be-gained in the LM local horizontal direction parallel to the CSM orbit plane with a positive value indicating a posigrade maneuver. 28J2 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. 28J3 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 and fixed once the maneuver starts, yielding an initial velocity-to-be-gained vector $\underline{\Delta V}$. $\underline{\Delta q}_S$, the accumulated velocity gained in inertial coordinates is set equal to zero until the maneuver starts at which time the initialization block is bypassed and the velocity to be gained vector is computed as

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

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 and δ_{11} set to zero. δ_{11} is another logic routing flag (used in the CSI calculation) that must be zero the first time the CSI calculations are performed. The logic flow then proceeds to $\textcircled{D_7}$ on Figure 3.26. Here, until the velocity-to-be-gained is less than 5K26, 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 5K26 the desired X body axis direction is not updated because of the large attitude maneuvers that may ensue from the indeterminate calculations of $\underline{V}_G/V_G$. The velocity-to-be-gained along each body axis (V_{Gx}, V_{Gy}, V_{Gz}) is then computed and used in the calculation of $\underline{V}_g$ as explained in Section 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. If however the vehicle is in "rendezvous 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 3.26 is common to all guidance modes and will not be discussed in the following subsections.

6.3.7.2 Orbit Insertion

Returning the Figure 3.17 follow the logic flow down the left hand side of the page. Assume that $S_{10} \neq 1$ and note that δ_{11} is set to zero. Again this is so because the first time the path $S_{10} = 1$ is entered δ_{11} must be zero. Assume also that $S_{10} \neq 2$ but rather that S_{10} equals zero which means the guidance equations are in the orbit insertion mode of operations.

This guidance mode has been designed to drive the LM vehicle to a prescribed altitude above the moon with specified values of altitude rate and horizontal velocity. In addition, steering in this mode is such that the LM is driven into the CSM orbit plane at engine cutoff with an out of plane velocity component of zero. Of course, in some abort situations all these conditions cannot be achieved. Comments on these situations are contained in the following discussion.

At the bottom of Figure 3.17 several quantities are established. First δ_{10} is set to 1 to control the logic flow later. The desired final value of horizontal velocity V_{hf} is set to the value of the input constant 24J. The present out of plane component of position (y) and velocity (V_y) are computed and the desired final value of radial rate ($\dot{r}_f$) is established. A detailed discussion of this computation is considered below. Next the in-plane component of horizontal velocity, V_{hA}, is computed and the logic flow proceeds to D8 on Figure 3.25. At this point the magnitude of the velocity-to-be-gained is computed from the following equation

$$V_G = \Big[(\underbrace{V_{hf}}_{\text{final inplane horizontal velocity}} - \underbrace{V_{hA}}_{\text{present inplane horizontal}})^2 + (\underbrace{\dot{r}_f}_{\text{final value of radial rate}} - \underbrace{\dot{r}_A}_{\text{present value of radial rate}})^2 + \underbrace{(V_y)^2}_{\text{present value of out-of-plane velocity}}\Big]^{1/2}$$

The quantity ψ is computed to determine if the thrust maneuver is to be posigrade or retrograde. Since $S_{10} = 0$ the time to burn T_B is computed. Derivation of this equation is contained in Section 7.4.3 of this report.

Following the calculation of T_B the desired values of the derivative of radial rate and the derivative of out of plane velocity are computed. In addition, the predicted value of final radial position and out of plane position are computed. Then since δ_{10} equals 1 the desired values of the second derivative of radial rate and the second derivative of yaw velocity are computed. These calculations are based upon the orbit insertion guidance law which is such as to maintain $\dddot{r}_d$ and $\dddot{y}_d$ constant. Detailed discussion of this guidance law is presented in Section 7.4.3 of this report. Following these computations the value of $\ddot{r}_d$ is modified to account for vehicle motion about a spherical body. The resulting value of $\ddot{r}_d$ is the desired vertical component of the thrust acceleration. The terms $\frac{K_1}{r^2}$ and $\frac{V_h{}^2}{r}$ are the gravitational and centrifugal accelerations. The logic flow then proceeds to point $\textcircled{L}$ on Figure 3.26. Note on Figure 3.25 that $\dddot{r}_d$ and $\dddot{y}_d$ are set to zero when δ_{10} equals zero. This occurs in all guidance modes except orbit insertion.

The purpose of the equations of Figure 3.26 is to generate the desired pointing direction of the vehicle X body axis and to obtain the velocity-to-be-gained along each body axis.

ψ_p is defined as the sine of desired pitch angle and ψ_y as the sine of the desired yaw angle. In general, these quantities are computed as the desired value of radial acceleration divided by thrust acceleration and 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 ψ_p equal to 1 and ψ_y = to 0. Then the desired pointing vector X_{bD} is merely $\underline{U}_1$, the unit vector in the radial direction.

The logic at the top of Figure 3.26 is used to determine if the thrust vector should be in the radial direction or not. The first check is that of LM altitude above the lunar landing site (r-5J) against the constant 21J (25,000 ft). If the altitude is less than 21J a check is made of altitude rate ($\dot{r}$) against the constant 22J (50 fps). If $\dot{r}$ is greater than 22J the desired pitch profile is flown. If not then the vehicle is commanded to thrust vertically. This last check actually controls the time at which the LM starts pitching over after the vertical rise from the lunar surface. For nominal missions this occurs approximately 12 seconds into the powered flight.

Following calculation of the desired pointing vector $\underline{X}_{bD}$ the various components of velocity-to-be-gained are computed as discussed in Sections 6.2.1 and 6.3.7.1. These components are meaningful only near cutoff. This then completes the orbit insertion calculations for any 2 second computing cycle.

Two ideas above have been glossed over and 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 $\dddot{r}_d$ and $\dddot{y}_d$.

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

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

and then limited between the values of 23J and 4K6 as follows

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

Then $\dot{r}_f$ as a function of final altitude appears as follows (for the DMCP trajectory)

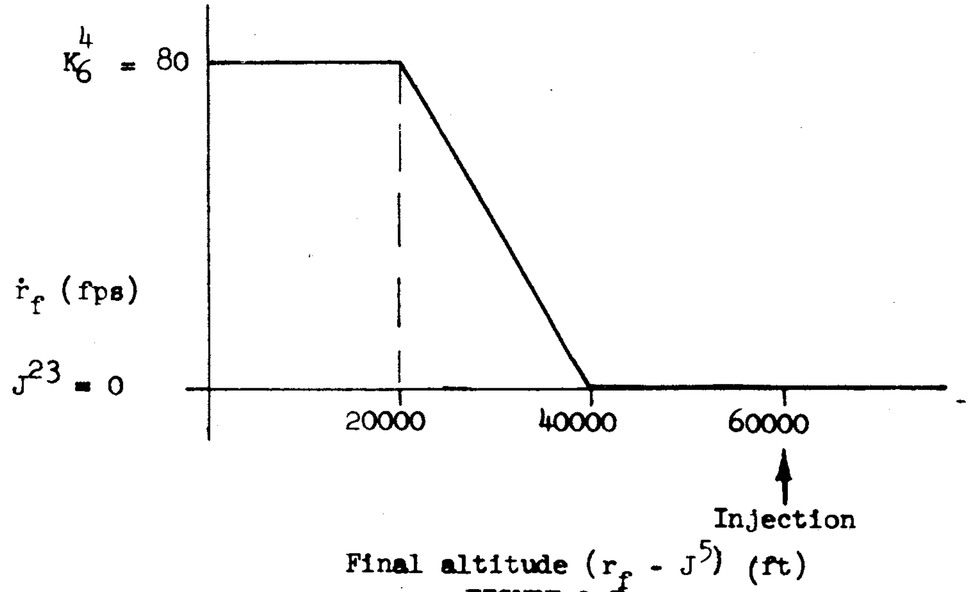


FIGURE 2.7

ALTITUDE RATE VS. FINAL ALTITUDE

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 the low fast abort that was discussed in Section 7.4 of Reference 3.

In this type case, 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 pericynthion. Prescribing $\dot{r}_f$ as above yields this condition.

The calculation of $\dddot{r}_d$ and $\dddot{y}_d$ are contained on Figure 3.26. The equation for $\dddot{r}_d$ contains the term J^{16} which is the desired final altitude at the time of orbit insertion. Derivation of this equation is contained in Section 7.4.3 of this report. The value of $\dddot{r}_d$ is constrained to lie between the values of K^5_{15} and zero. These values are chosen so as to limit the maneuvers of the vehicle. When $\dddot{r}_d$ lies outside either of the limits, the effect is that the equations give up trying to drive the final altitude of the LM to the desired altitude but just make a limited correction. The orbit insertion maneuver will continue, however, until the desired velocity is obtained.

The equation for $\dddot{y}_d$ is similar to that of $\dddot{r}_d$. 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, $\dddot{y}_d$ is limited to the value $\pm K^5_{16}$. If the computed value of $\dddot{y}_d$ is outside the limits, the equations give up trying to drive the LM into the CSM orbit plane at engine cutoff but just make a limit correction. The philosophy used in the simulations made to this date has been to steer out approximately $1/2^o$ 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. Since the CSI and CDH maneuvers are done parallel to the CSM orbit plane, any additional out-of-plane error is removed at the TPI maneuver. For quantitative purposes, two simulation runs were made with the LM 2^o out of the CSM orbit plane at liftoff (see Reference 3), Section 7.2. In the first run constant K^5_{16} was set to steer out $1/2^o$ of this error during orbit insertion and the remainder during the direct transfer phase (TPI maneuver). Total ΔV expended to effect rendezvous in this situation was 6388 fps. In the second simulation run, the constant K^5_{16} was set so that all 2^o were eliminated during orbit insertion. In this situation, the total velocity required to rendezvous was 6576 fps. These results indicate the desirability of the selected orbit insertion philosophy.

6.3.7.3 CSI Routine

The guidance routine under discussion in this section is the CSI routine which is initiated on Figure 3.17. Shortly after the orbit insertion maneuver is completed, the astronaut switches S_{10} from zero to 1 and ascertains that the following targeting constants have been entered:

t_{igA}	absolute time at which the CSI maneuver is to be performed
J^1	absolute time at which the TPI maneuver is to be performed
J^2	desired line-of-sight angle between the LM and CSM at time J^1.

The purpose of the CSI calculation is to determine the magnitude of the horizontal burn to be performed parallel to the CSM orbit plane at CSI time (t_{igA}) such that following the coelliptic maneuver (CDH maneuver) the desired line-of-sight angle (J^2) between LM and CSM will be achieved at time J^1. By definition, coelliptic means that: (1) the product of semi-major axis and eccentricity of the LM trajectory equals the same product of the CSM orbit, and (2) the line of apsides of the two orbits are aligned. For calculation of these quantities, the semi-major axis of the LM orbit is obtained as the semi-major axis of the CSM trajectory minus Δr, where Δr is the distance between the two orbits on the radial line passing through the LM at the time of the CDH maneuver.

The solution of this problem is solved by an iteration technique. In each two-second computing increment, three trial values of horizontal velocity magnitude increments, V_H, are selected and an appropriate error function evaluated for each. The magnitude of the successive velocity increments differ by the quantity Δ_6. The value of the error function associated with each trial value of horizontal velocity increment is examined and the solution corresponding to the minimum value of error retained.

The horizontal velocity magnitude corresponding to the minimum value of error is used as the second of the three trial values in the next 2-second computing increment. To cause the iteration to converge to the correct value, the following rule is used to modify Δ_6. If either the first or third trial value of horizontal velocity increment yielded the minimum cost, increase the value of Δ_6 (up to a limit) by a factor 1.5. If the middle (second) trial value of horizontal velocity increment yielded the minimum cost, decrease the value of Δ_6 by a factor of 0.4. Thus, several sequences of the iteration may appear as follows.

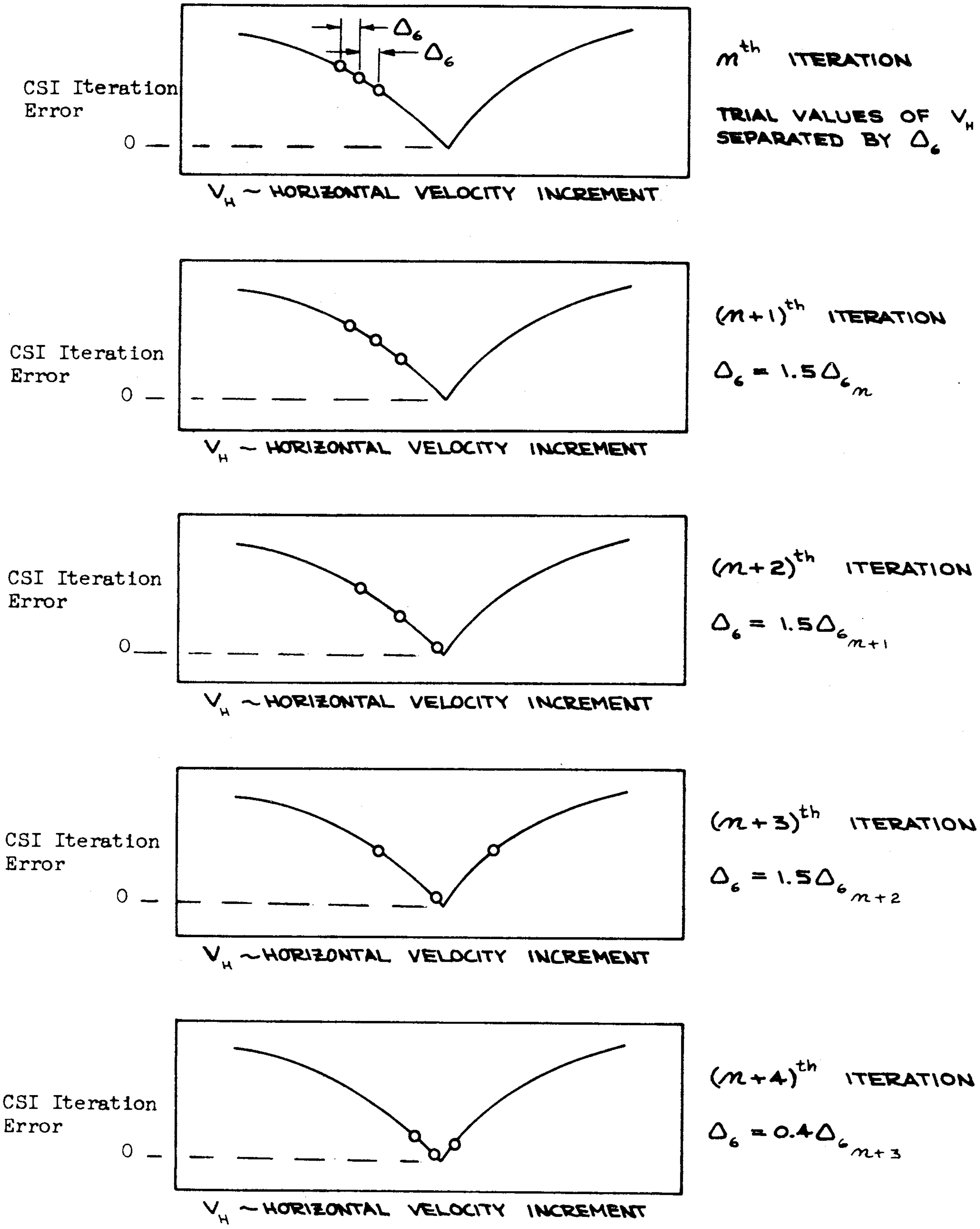


FIGURE 2.8

SEQUENCE OF CSI ITERATIONS

The maximum time observed to date for the iteration routine to converge to within 1/4 fps has been 36 seconds. This situation was obtained under adverse conditions. Usually it takes about 24 seconds to obtain the desired solution.

In order to simplify the discussion of the equation flow logic, derivation of many of the equations have been placed in Part II of this report. The discussion begins with mission time sometime after orbit insertion and prior to CSI. The first time the logic follows the path $S_{10} = 1$, the quantity δ_{11} equals zero. This is shown on Figure 3.17. This initializes the quantity Δ_6 and V_{ho}. δ_{11} is then set to 1 and this block is no longer entered. The flow logic now proceeds to D1 on Figure 3.18 where the quantity T_i is computed. This quantity is the time remaining until the time of the CSI maneuver which is used in the ellipse predictor routine to obtain the predicted position and velocity of the LM at the time of the CSI maneuver. This predicted position and velocity is denoted by $\underline{r}_5$ and $\underline{V}_5$ respectively. T_I is then set to T_i for later use and the horizontal unit vector parallel to the CSM orbit plane, $\underline{V}_{1A}$, is computed. The velocity increment added at CSI time will be in this direction.

The next group of equations set up the iteration routine for the present two-second computing increment. This block is entered only once per two-second computing increment. The various quantities are:

i - The variable used to index the three trial values of horizontal velocity increment. This parameter assumes the values -1, 0, +1. The horizontal velocity increment associated with the index i = 0 is greater than that associated with the index i = -1 and less than that associated with the index i = +1. This indexing then makes it easy to control Δ_6 in the next computing increment because if the index associated with the minimum error this cycle (denoted by i_o) equals zero then the quantity Δ_6 should be decreased because the middle value was chosen as best. If not, then Δ_6 should be increased.

C_o - This quantity is the minimum value of error each two-second computing increment, which is originally set to the maximum value 2^3. Even though the best solution of the three trials per two-second computing increment is saved for use in the next two-second computing increment the value of error associated with this solution is not saved. Thus, the three values of error are compared against each other and not with any from previous computing cycles.

V_H - This quantity is the trial velocity increment and is first computed as

$$V_H = V_{ho} - V_h - \Delta_6$$

where

V_{ho} is the previously determined (from preceeding computer cycles) best value of horizontal velocity increment obtained plus the LM horizontal velocity in that computing increment. The formulation has been done in this way to make the solution valid after the maneuver begins.

V_h is the present value of horizontal velocity

Δ_6 is the difference between trial horizontal velocity magnitudes. To understand what is taking place assume that the CSI calculations are being done for the first time. Then $V_{ho} = V_h$ from the initialization on Figure 3.17 and $\Delta_6 = 2K5$. The quantity $V_H = V_{ho} - \Delta_6 = V_h - V_h - \Delta_6 = -2K5$. Thus the first horizontal velocity increment used is -2K5. As will be seen below the second trial horizontal velocity increment is obtained from the first by the equation

$$V_H = V_H + \Delta_6$$

or in this case

$$V_H = -2K5 + 2K5 = 0$$

Similarly, the third trial value of V_H is obtained by adding Δ_6 to V_H. Thus, the first 2 second computing increment in which the CSI calculations are done, the trial velocity increments are -2K5, 0, +2K5 indexed by i = -1, 0, +1 respectively. The best of the three values of V_H is selected and denoted by V_o and the quantity V_{ho} used in the next 2 second computing increment is obtained as

$$V_{ho} = V_h + V_o$$

Also Δ_6 is modified as described above. Thus when the calculation for V_H is done in the next 2 second computing increment it assumes the value

$$V_H = V_{ho} - V_h - \Delta_6 = V_{h,\ n-1} + V_o - V_{h,\ n} - \Delta_{6_n}$$

where n and n-1 have been used to denote present values and values valid 2 seconds previously. Assume for the moment that $V_{h,\ n-1} = V_{h,\ n}$

then

$$V_H = V_o - \Delta_6$$

is the horizontal velocity increment used as the first trial value. As above, the second trial value is given by $V_H = V_H + \Delta_6 = V_o$ and the third by $V_H = V_H + \Delta_6 = V_o + \Delta_6$. Remembering that V_o was the best horizontal velocity increment from the previous two-second computing increment, it is seen that in the present computing increment the search is made around the previous best value. The process continues in this way until the best value of V_H has been determined. This is characterized by Δ_6 shrinking to a preset value. It is not apparent in the above discussion why the values of V_h have been used. The reason is that once the maneuver starts we desire the iteration routine to follow the effects of the maneuver (maintain the correct answer). This is done by effectively decreasing the value of V_H by the horizontal velocity gained or approximately by $V_{h,\ n-1} - V_h$.

$\dot{r}_5$ - predicted value of LM altitude rate at CSI time.

$\dot{r}_A$ - predicted value of LM altitude rate at CDH time. This quantity is set to zero because the CDH maneuver is programmed to occur at either apocynthion or pericynthion. The CSI calculations assume the CDH maneuver is made impulsively. During the CDH maneuver itself, however, the assumption of zero $\dot{r}_A$ is not made because the actual burn is of finite duration and also may not occur precisely at the line of apsides.

The next group of equations beginning with the calculation of $\underline{V}_{55}$ to the calculation of T_A are done to determine the time from the CSI maneuver to the CDH maneuver for the trial value of horizontal velocity increment under consideration. This desired time is T_A. The equation for $\underline{V}_{55}$ is the first equation to be solved three times per computing increment. That is, later in the flow logic, directions will be given to return to FA on this page. The time of the CDH maneuver is designated as either the first or second crossing of the line of apsides after the CSI maneuver and is controlled by switch S_{16}. If $S_{16} = 0$, the maneuver is to be performed at the first crossing and if $S_{16} = 1$, the maneuver is to be performed at the second crossing of the line of apsides.

The following possibilities then exist

$\dot{r}_5$	S_{16}	CDH Maneuvers Occur at
<0	0	pericynthion
<0	1	aspcynthion
>0	0	apocynthion
>0	1	pericynthion

When $\dot{r}_5$ is close to zero either one of two possibilities exist

(1) The LM is on a near circular orbit in which case quantities such as first and second crossing begin to lose their meaning

(2) The LM is either near apogee or perigee in which case the CDH maneuver is to occur either at 180° away or 360° away.

In either of the situations the switch S_{17} should be set to 1 and S_{16} set to 0 for CDH to occur at 180° or S_{17} set to 1 and S_{16} set to 1 for CDH to occur at 360°. For (1) above it is obvious why this is necessary. The reason why it is necessary in (2) is now presented. Assume the CSI maneuver begins just prior to say pericynthion and it is desired to perform the maneuver at apocynthion and thus on the 2nd crossing of the line of apsides. Normally then S_{17} would be set to 0 and S_{16} set to 1. If the duration of the maneuver is sufficiently long for the LM to pass through pericynthion then this configuration of switch settings would cause a discontinuity in the solution because now the second crossing occurs at pericynthion. For this reason it is suggested that if $|\dot{r}_5|$ is small then S_{17} should be switched to 1.

At the bottom of Figure 3.18 the time T_A is used to predict the value of LM position and velocity $\underline{r}_6$, $\underline{V}_6$ at the time of the CDH maneuver. The logic flow then proceeds to ZB on Figure 3.19. The calculations on this page are used to obtain several quantities required in the error function and to compute the various components of velocity required to perform the CDH maneuver. The first computations on Figure 3.19 compute the CSM position and velocity at the time of the CDH maneuver. The desired quantity from this information is CSM altitude rate, denoted by $\dot{r}_c$ and eventually saved as $\dot{r}_B$. δ_7 is set to zero merely to control the following logic flow. $\dot{r}_c$ is then calculated and stored in $\dot{r}_B$. δ_7 is then set to 1. The purpose of the calculations on the lower right of Figure 3.19 is to obtain the value of Δr which is defined as the distance difference between the CSM and LM orbits as measured on the radial line passing thru the LM at the time of the CDH maneuver. Thus to obtain this, the distance of the CSM must be determined at the position in the orbit specified. Since the LM can be out of the CSM orbit plane the procedure used is to project the LM at CDH time onto the CSM orbit plane and the compute the central angle between this projection and the CSM. This angle is denoted by θ_f and division of θ_f by the mean CSM orbital rate yields the approximate time T_δ that $\underline{r}_7$ and $\underline{V}_7$ (CSM position and velocity at CDH) must be propagated to be on the desired radial line. This propagation is done in the ellipse predictor and the output is again denoted by $\underline{r}_7$, $\underline{V}_7$. From this then the distance of the CSM from the center of the moon can be determined along with radial rate. This is done at the top right hand side of the page. Then since δ_7 has been set to 1 the logic flow is to the left and the calculations performed to obtain the following quantities.

Δr definition above

α_L semi major axis of desired LM orbit equals semi major axis of CSM orbit minus Δr

V_{hA} LM horizontal velocity at time of CDH maneuver

n_L mean orbital rate of desired LM trajectory

$\dot{r}_f$ desired altitude rate of LM at CDH time

V_f desired value of LM velocity at CDH time

v_{hf} desired horizontal velocity at CDH time

After these calculations logic flow proceeds to D8 on Figure 3.25 where the velocity-to-be-gained during the CDH maneuver is calculated as V_G. Then since S_{10}

equals 1 the logic flow returns to D9 on Figure 3.20.

The equations on Figure 3.20 evaluate the error function, store the best solution, generate the best value of the time of CDH maneuver and control the iteration. At the top of Figure 3.20 the quantity b_3 is calculated. b_3 is defined as the desired central angle between LM and CSM at TPI time based upon the desired line of sight angle J^2 and Δr. The error function is the absolute value of the difference between b_3 (the desired) and the actual central angle that would be achieved if this trial trajectory were flown. It is this function that is minimized. Derivation of b_3 and the error function C is contained in Part II of this report. The remainder of the iteration logic operates as follows. Assume the first of three trial values of horizontal velocity magnitude (indexed by i = -1) is being considered. The check of C against C_o will be less than zero because C_o was set to its maximum value at the beginning of the two second computer cycle (see Figure 3.18). This solution then is stored in the block to the right of the $(C-C_o)$ check and i_o is set to 1. Then since i equals -1, V_H is incremented by Δ_6 as explained above and i is increased to 0. The logic flow then goes back to FA on Figure 3.18 and a new trial solution is obtained. Back on Figure 3.20 if the error value for this trial is less than the previous value of error then the new solution is stored and i_o set to 0. If $(C-C_o)$ is greater than zero no new solution is stored. Again i is checked and since i = 0, V_H and i are again incremented and the logic flow again returns to FA on Figure 3.18. Back again on Figure 3.20 the same procedure is followed. This time, however, when the check on i is made the flow is to the right. The direction to point the LM X body axis is obtained from the sign of V_o and used as previously stated. $\dot{r}_f$ is set equal to $\dot{r}_A$ for later purposes and the magnitude of velocity-to-be-gained during the CSI maneuver is obtained as the magnitude of V_o. The best time of CDH maneuver t_{igB} is then computed as $t_{igA} + T_{AO}$ where T_{AO} is the best value of T_A. Then, if i_o (the index of the best solution) is zero, Δ_6 is decreased and limited. If, however, $i_o = 1$ then Δ_6 is increased and limited. The flow logic proceeds to D3 on Figure 3.25.

The purpose of going to Figure 3.25 is to use equations that have already been programmed. In this mode (as in all modes except orbit insertion) δ_{10} is 0 so that $\dddot{r}_d$ and $\dddot{y}_d$ are zero. Then, since $\dot{r}_f = \dot{r}_A$ and V_y equals zero, both $\ddot{r}_d$ and $\ddot{y}_d$ are zero. In this mode of operation these are the only equations of interest on this page. The logic flow then follows exactly as in the orbit insertion mode through Figure 3.26. Now, however, both ψ_p and ψ_y equal zero so that the desired X body axis pointing direction is established to be along the unit vector V_1 in a direction determined by ψ (the sign of the horizontal velocity increment).

6.3.7.4 CDH Routine

After the CSI maneuver has been executed, the astronaut inserts the setting $S_{10} = 2$. Then, during the 2 second computations, Figure 3.17 is entered and the logic flow follows the path $S_{10} = 2$. In this mode the CDH maneuver is computed as if the burn is to be done immediately. The equations to do this have already been discussed in the preceding section.

When $S_{10} = 2$, several quantities are established for succeeding calculations. Both T_I and T_A are set to zero because the calculations are being done as if the maneuver were to occur immediately. The quantity T_Δ (time until CDH) is computed as $t_{igB} - t$. Moreover, $\underline{r}_6$ and $\underline{V}_6$ are set equal to $\underline{r}$ and $\underline{V}$ respectively. The logic flow then proceeds to ZB on Figure 3.19 where the CDH maneuver is calculated as in the CSI computations. Leaving Figure 3.19 the logic flow proceeds to D8 on Figure 3.25 where the velocity to be gained is computed. Note that since S_{10} is not equal to 1, all the CSI function and iteration logic is bypassed. The logic flow proceeds immediately to the calculations to determine the desired pointing direction $\underline{X}_{bD}$. Now, $\ddot{y}_d = 0$

and $\ddot{r}_d = \frac{\dot{r}_f - \dot{r}_A}{T_B}$ as calculated on Figure 3.25. Then the logic flow proceeds to Figure 3.26 as previously and ψ_y is set zero and $\psi_p = \frac{\ddot{r}_d}{a_T}$ which is approximately equal to $\frac{(\dot{r}_f - \dot{r}_A)}{V_G}$. The remainder of the equations are exactly the same as previously.

Note that prior to the maneuver the desired attitude of the LM is not fixed in inertial space but actually rotates with the LM vehicle. This is the same situation that occurs prior to the CSI maneuver.

6.3.7.2 Direct Transfer (TPI)

All possible logic paths on Figure 3.17 have been discussed with the exception of the path going to Z5 at the bottom of the page. This path is used when S_{10} is either 3 or 4, the direct transfer modes.

The direct transfer mode is used in the coelliptic rendezvous scheme following the CDH maneuver. The two previous maneuvers (CSI and CDH) were performed such that at the desired TPI time (J^1) the proper phasing exists between the LM and CSM so that the desired line of sight angle is achieved. For each line of sight angle at TPI time there is a corresponding best time until rendezvous. The values used in the simulations to date have been:

desired line of sight angle (J^2)	26.6°
time of TPI to rendezvous	2880 sec

2880 seconds corresponds to 140° central angle of the CSM in an 80 nm circular orbit.

Because of 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 where the astronaut can determine when the desired line of sight will be achieved and perform the maneuver at this time if so desired. In this mode also, the astronaut could determine 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:

$S_{10} = 3$

The astronaut introduces an increment of time, T_Δ, 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 secs per 2 second because the time of initiation of the burn t_{igC} equals $t + T_\Delta$. T_Δ is a fixed number and t increments two seconds per 2 second. 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.

$S_{10} = 4$

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 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_{igC} 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_{igC}) 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_{igC} is inserted a solution is found in one computing increment.

Assuming the value of t_{igC} has been selected (and $S_{10} = 4$) then the quantity T_Δ 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 on Figure 3.21 it is observed that when $S_{10} = 3$ the input quantity is T_Δ and t_{igC} is computed whereas when $S_{10} = 4, t_{igC}$ is the input quantity and T_Δ is computed. The remainder of the logic is the same for both modes. T_r is the time to rendezvous. If T_Δ is greater than zero (meaning the burn has not been initiated yet) then the position $\underline{r}_5$ and velocity $\underline{V}_5$ of the LM at the time of the maneuver is ascertained from the ellipse predictor. If T_Δ is less than zero it is set to zero for use in the ellipse predictor and $\underline{r}_5$ and $\underline{V}_5$ are the LM present position and velocity. Next, the position $\underline{r}_c$ and velocity $\underline{V}_c$ of the CSM are updated to the same time as that of the LM. From these two quantities the line of sight angle, θ_{LOS}, is computed. 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 $J^3 = 0$)

$$t_i = t_b + T_r$$

where t_b is the time from the CSM epoch point to the present time. The CSM position at rendezvous is denoted by $\underline{r}_T$ and the velocity by $\underline{V}_T$. In preparation for use of the p iterator two other quantities are calculated, T and p. T is the time from present to rendezvous if $t > t_{igC}$ or the time from the TPI maneuver to rendezvous if $t < t_{igC}$. p is an initial guess of the value of the semi latus rectum for use in the p iterator. r_f is the magnitude of the LM radius vector at either the present time, if $t > t_{igC}$, or that of the LM at the time TPI if $t < t_{igC}$. The logic flow then leads directly to Figure 3.22, the first page of calculations of the p iterator.

The p iterator equations contained in Figures 3.22, 3.23 and 3.24 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 $t > t_{igC}$ or the position at the TPI maneuver time if $t < t_{igC}$. The second point is the rendezvous point.

The actual details of the p iterator are contained in Part II of this report. Here only 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 degrees of the LM orbit plane. This restriction comes about because of the calculation (on Figure 3.22)

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

where y, and y_c are the components along the y inertial axis of $\underline{W}_1$ and $\underline{W}_c$ respectively.

The term sgn ($y\ y_c$) is an approximation to the exact expression sgn ($\underline{W}_1 \cdot \underline{W}_c$). The constraint of ± 80° 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 π orbits. 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 ± 10° and for earth missions the constrained region is ± 20°.

In the p iterator, eight trial trajectories are considered each 2 second 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 8 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 2K20 seconds (2 seconds) then the p iterator will not yield a solution. A way to check the p iterator to see if a solution has been obtained is to examine the quantity V_T which is essentially the sum of the magnitudes of the 2 velocities to be gained (initial and final). If the value of V_T equals 2K11 (8000) then the p iterator has not obtained a solution.

The outputs of the p iterator needed to perform the first maneuver (both for steering and for velocity-to-be-gained) are $\dot{r}_f$ - the desired radial rate after the TPI maneuver, V_G - the velocity-to-be-gained, V_y - the out of transfer plane velocity component, ψ - the indication of a posigrade or retrograde maneuver, and $\dot{r}_A$ - the actual radial rate at the time the maneuver is performed. Then the output of the p iterator goes directly to D3 on Figure 3.25 where the computation of T_B is performed. The remainder of the equations are as discussed several times above.

It is interesting to note that if the flight equations are in the "rendezvous steering" mode ($S_{00} = 1$) prior to the time of the TPI maneuver the vehicle will orient itself in inertial space to the correct attitude required at the time of the maneuver. This is in contrast to the CSI and CDH maneuvers where the vehicle will maintain the correct attitude with respect to the local coordinate system ($\underline{U}_1$, $\underline{V}_1$, $\underline{W}_1$).

Before concluding the discussion of the guidance equations it should be noted that it is not necessary to orient the X body axis of the vehicle in the desired direction of thrust, i.e. set $S_{00} = 1$, for these various guidance options (orbit insertion, CSI, CDH, TPI). This has already been pointed out for the "external ΔV" mode where a separate section was presented on placing the vehicle in "attitude hold" ($S_{00} = 0$) and thrusting along each axis individually. This same procedure can be used in all other guidance modes also if desired.* In addition, this is a feasible way of eliminating residual velocity-to-be-gained after engine cutoff. It should be noted, however, that the maneuver must be accomplished with the RCS engines since the engine OFF discrete is set when $S_{00} \neq 1$.

* Obviously, more propellant will be required if each ΔV component is reduced to zero separately.

PART II

DERIVATION AND DISCUSSION

7.0 INTRODUCTION

In this part of the report derivations of many of the important equations are presented and discussed. There are three general categories considered. These categories and the subsections contained within them are:

ALIGNMENT AND CALIBRATION

Body Axis Align

IMU Align

Lunar Align

Gyro Inflight Calibration

Accelerometer Inflight Calibration

Gyro Lunar Calibration

NAVIGATION

Direction Cosine Updating Algorithm

Derivation of Radar Filter Equations

GUIDANCE

Attitude Error Commands

Ellipse Predictor Subroutine

Steering Equations for Orbit Insertion

Derivation of Cost Function for CSI Calculations

Derivation of the Equations to Obtain Coelliptic Orbits

Derivation of p-iterator Equations

7.1 Alignment

The alignment of a strapdown system is essentially the computation of the correct direction cosines which relate the vehicle body axes to the desired inertial coordinate axes. This alignment may be accomplished in many different ways, three of which are mechanized in the LM abort guidance system.

7.1.1 Body Axis Align ($S_{00} = 5$)

In this alignment mode of the AGS, the direction cosines are set equal to a unit matrix. This is equivalent to aligning to an inertial coordinate system which coincides with the vehicle body axes at the instant the alignment is performed.

7.1.2 IMU Align ($S_{00} = 3$)

In the IMU alignment mode, the AGS direction cosines are set equal to the PGNCS direction cosines computed from the PGNCS Euler angles θ_p, ϕ_p, ψ_p. This alignment method is in error due to the quantization of the PGNCS Euler angles (40 seconds of arc), the PGNCS alignment error, and also by the amount the PGNCS inertial platform has drifted since it was aligned. The equations for computing the desired (PGNCS) direction cosines are shown below, (Equation 7.1.2) and are based on the gimbal order of the PGNCS system as expressed in Equation 7.1.1. IMU alignment is performed prior to the inflight gyro calibration, before the descent of the LM vehicle to the lunar surface, and on the lunar surface if the PGNCS is still operating after the lunar landing has occurred.

$$[a_{ij}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\phi & S\phi \\ 0 & -S\phi & C\phi \end{bmatrix} \begin{bmatrix} C\psi & S\psi & 0 \\ -S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\theta & 0 & -S\theta \\ 0 & 1 & 0 \\ S\theta & 0 & C\theta \end{bmatrix} \qquad (7.1.1)$$

where $C\beta = \cos\beta$ and $S\beta = \sin\beta$

$$
\begin{aligned}
a_{11} &= \cos \psi_p \cos \theta_p \\
a_{12} &= \sin \psi_p \\
a_{13} &= -\cos \psi_p \sin \theta_p \\
a_{31} &= \sin \phi_p \sin \psi_p \cos \theta_p + \cos \phi_p \sin \theta_p \\
a_{32} &= -\sin \phi_p \cos \psi_p \\
a_{33} &= \cos \phi_p \cos \theta_p - \sin \phi_p \sin \psi_p \sin \theta_p
\end{aligned}
\qquad (7.1.2)
$$

Since the other three direction cosines may be computed from the above direction cosines, and since these three direction cosines are computed in this manner in the direction cosine update computations, the other three direction cosines are computed as shown in Equation (7.1.3).

$$
\begin{aligned}
a_{21} &= a_{13}a_{32} - a_{12}a_{33} \\
a_{22} &= a_{11}a_{33} - a_{13}a_{31} \\
a_{23} &= a_{12}a_{31} - a_{11}a_{32}
\end{aligned}
\qquad (7.1.3)
$$

Equation (7.1.3) is easily derived as $\underline{X}_b = \begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \end{pmatrix}$,

$$\underline{Y}_b = \begin{pmatrix} a_{21} \\ a_{22} \\ a_{23} \end{pmatrix} ,$$

$$\underline{Z}_b = \begin{pmatrix} a_{31} \\ a_{32} \\ a_{33} \end{pmatrix} , \text{ and } \underline{Y}_b = \underline{Z}_b \times \underline{X}_b$$

7.1.3 Lunar Align Equations

Prior to the descent of the LM to the lunar surface, the X_b, Y_b, Z_b vehicle axes are aligned to a selenocentric coordinate system with the X axis through the intended landing sight at the nominal time of landing positive outward from the lunar center. The Z axis is defined as the cross product of a unit vector along the angular momentum vector of the CSM orbit with a unit vector along the X axis. Thus, the inertial coordinate system used for the descent phase has the X and the Z inertial axes in the CSM orbit plane.

When the LM lands on the lunar surface, the vehicle azimuth with respect to the CSM orbit plane is determined by storing, upon command of the DEDA, the AGS azimuth reference as defined by the appropriate elements of the transformation matrix [A]. This azimuth reference δ_L is defined to be the angle between the Y inertial axis and the projection of the Z body axis on the Y - Z inertial plane, measured positive in the right-hand rotational direction about the X inertial axis. (See Figure 7.1). From this figure, the direction cosines which

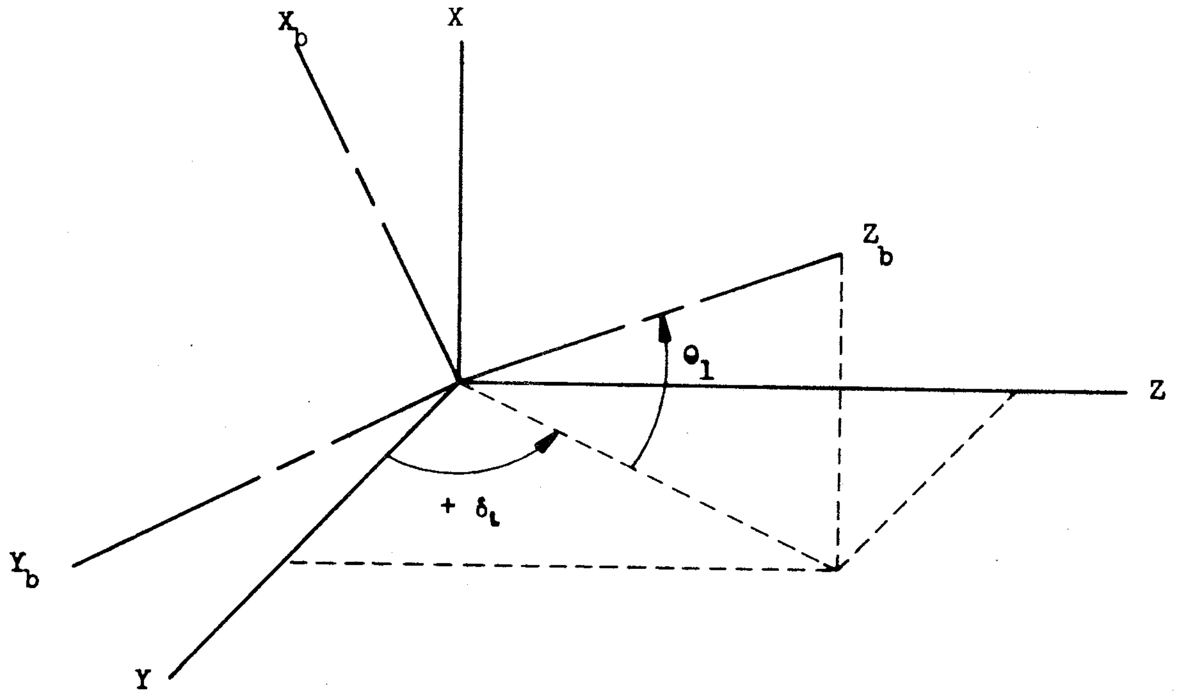


FIGURE 7.1
Lunar Align Geometry

define the Z body axis location with respect to the X, Y, Z inertial axes may be determined, and are presented in Equation (7.1.4).

$$a_{31} = Z_b \cdot X = \sin \theta_1$$
$$a_{32} = Z_b \cdot Y = \cos \theta_1 \cos \delta_L$$
$$a_{33} = Z_b \cdot Z = \cos \theta_1 \sin \delta_L \qquad (7.1.4)$$

Since $a_{31}^2 = \sin^2 \theta_1$, $1 - a_{31}^2 = \cos^2 \theta_1$, the cosine and sine of the landing azimuth δ_L as mechanized in the AGS are computed as shown in Equation (7.1.5). These equations

$$\sin \delta_L = \frac{a_{33}}{[1 - a_{31}^2]^{1/2}}$$

$$\cos \delta_L = \frac{a_{32}}{[1 - a_{31}^2]^{1/2}} \qquad (7.1.5)$$

are valid only when $|\theta_1| < 90$ degrees. Since the maximum vehicle tilt angle is expected to be 30 degrees, and the LM is expected to land within 1 degree from the nominal landing site, the equations used to define δ_L are valid as defined.

After the lunar landing, the AGS is aligned to another selenocentric coordinate system with the inertial x axis through the intended launch site at the time of launch, positive outward from the lunar center. The inertial Z axis direction is defined conceptually by the cross product of a unit vector along the CSM angular momentum vector with a unit vector along the inertial X axis. That is, the Z inertial axis is parallel to the CSM orbit plane.

In the lunar align equations, the direction cosines which relate the vehicle body axes to the local level lunar centered coordinate system described above are computed with a low gain filter. This filter uses the accelerometer outputs to obtain the estimate of the vehicle leveling errors, and an azimuth reference constant δ_A to compute the estimate of the azimuth error. δ_A is the updated value of the landing azimuth reference constant δ_L, where $\delta_A = \delta_L - \Delta_\delta$. Δ_δ is computed on the earth, and corrects for CSM orbit changes and the lunar rotation during the LM's stay on the lunar surface. Since the updated value for the azimuth reference constant is defined by the difference of two angles, the sine and cosine of the updated azimuth reference constant δ_A may be computed from Equation (7.1.6). Using this

$$\begin{aligned} \cos(\alpha - \beta) &= \cos\alpha \cos\beta + \sin\alpha \sin\beta \\ \sin(\alpha - \beta) &= \sin\alpha \cos\beta - \cos\alpha \sin\beta \end{aligned} \qquad (7.1.6)$$

equation, and small angle approximations for Δ_δ, the equations for the sine and cosine of δ_A are

$$\begin{aligned} \cos(\delta_A) &= \cos\delta_L + \Delta_\delta \sin\delta_L \\ \sin(\delta_A) &= \sin\delta_L - \Delta_\delta \cos\delta_L \end{aligned} \qquad (7.1.7)$$

Since the LM AGS is being aligned to a coordinate system with the X inertial axis vertical, the lunar gravity vector $\overline{g}_L = g_L \overline{X}$, where $\overline{X}$ is a unit vector along the x inertial axis and g_L is the magnitude of the lunar gravity. Thus, the components of lunar gravity along the X, Y, Z body axes are

$$\begin{aligned} g_x &= g_L\, a_{11} \\ g_y &= g_L\, a_{21} \\ g_z &= g_L\, a_{31} \end{aligned} \qquad (7.1.8)$$

and the outputs of the body mounted accelerometers are

$$\begin{aligned} \Delta V_{Xi} &= \int g_L\, a_{11}\, dt \approx g_L\, \Delta t\, a_{11} \\ \Delta V_{Yi} &= \int g_L\, a_{21}\, dt \approx g_L\, \Delta t\, a_{21} \\ \Delta V_{Zi} &= \int g_L\, a_{31}\, dt \approx g_L\, \Delta t\, a_{31} \end{aligned} \qquad (7.1.9)$$

The estimates of the leveling errors may be obtained by comparing the output of the Y and Z accelerometers with Equation (7.1.9).

$$\Delta\theta_y = \frac{1}{g_L \Delta t} [\Delta V_z - g_L \ \Delta t \ a_{31}]$$

$$\Delta\theta_z = \frac{1}{g_L \Delta t} [-\Delta V_y - g_L \ \Delta t \ a_{21}] \qquad (7.1.10)$$

The estimates of the azimuth errors are computed from the sine and cosine of δ_A in the following steps. First the desired values for a_{32} and a_{33} are computed, based on the assumption that a_{31} is equal to the sin θ_1 as described in Figure 7.1. These equations are

$$a_{32D} = (1 - a_{31}^2)^{\frac{1}{2}} \cos \delta_A$$

$$a_{33D} = (1 - a_{31}^2)^{\frac{1}{2}} \sin \delta_A \qquad (7.1.11)$$

equivalent to Equations (7.1.5) used to compute the sine and cosine of δ_L, the landing azimuth constant. The computation of a_{32D} and a_{33D} assumes that the leveling error in a_{31} is small. The estimate of the azimuth error is then computed by taking the dot product between the desired orientation of the Z body axis defined by a_{31}, a_{32D}, a_{33D} and the AGS orientation of the Y body axis defined by a_{21}, a_{22} and a_{23}.

$$\Delta\theta \cong (a_{31}a_{21} + a_{32D}a_{22} + a_{33D}a_{23}) \qquad (7.1.12)$$

The approximations in Equation (7.1.12) are that $\Delta\theta_x$ is a small angle, the difference between a_{31} and a_{31D} is very small, and the integral of the lunar gravity times a direction cosine is equal to $g_L \ \Delta t \ a_{ij}$. Since a_{31} is not a function of the azimuth angle δ_A, and is only a function of the leveling angle, a_{31} may be used in place of a_{31D} without any degradation of accuracy. However, the azimuth alignment will not converge until the leveling errors of the lunar align become small. The approximation of $\int_0^\tau a_{ij} \ g_L$ by $a_{ij}(\tau) \ g_L \ \tau$ should cause a negligible error as the direction cosines will be changing at a very slow rate.

The equations mechanized in the LM AGS for the lunar alignment of the transformation matrix are presented in Equation (7.1.13).

$$\begin{aligned} \Delta\alpha_{xA} &= -K^1_{26}\,[a_{31}a_{21} + a_{32D}a_{22} + a_{33D}a_{23}] \\ \Delta\alpha_{yA} &= K^1_{27}\,(\Delta V_z + \Delta V_{z,\ m-1}) - K^1_{28}\,a_{31} \\ \Delta\alpha_{zA} &= K^1_{28}a_{32} - K^1_{27}\,(\Delta V_{y,\ m-1}) \end{aligned} \tag{7.1.13}$$

where

$\Delta\alpha_{xA}$, $\Delta\alpha_{yA}$, $\Delta\alpha_{zA}$ are the alignment correction increments in radians.

a_{ij} are the present AGS direction cosines.

a_{ijD} are the desired direction cosines specified in (Equation (7.1.11)

K^1_{26}, K^1_{27}, K^1_{28} are alignment gains.

In the lunar align equations, the gains presently used are

$$\begin{aligned} K^1_{26} &= 0.007 \\ K^1_{27} &= 0.0435 \\ K^1_{28} &= 0.009264595 \end{aligned} \tag{7.1.14}$$

The quantity $g_L\ \Delta t$ for a 40-msecond alignment cycle is $\frac{g_L}{25}$ and is equal to 0.009264595. Thus, the effective alignment gains in the hardwired portion of the lunar align equations is 0.007 for azimuth and 0.009264595 for leveling. After leaving the hardwired lunar align equations, the leveling angle increments are shifted to the right 2 bits reducing the leveling gains to 0.00231615.

7.2 LM AGS Calibration

The AGS has the capability of performing three calibrations; prelaunch gyro calibration, inflight gyro and accelerometer calibration (during free fall), and lunar gyro calibration.

7.2.1 Gyro Inflight Calibration

Any departure of the LM AGS direction cosines from their correct values may be construed as being due to gyro drift. Since the PGNCS Euler angles, which are used to compute the reference direction cosines, are quantized to 40 seconds of arc, the departure of the AGS direction cosines is a combination of the gyro drift errors and PGNCS Euler angle quantization noise errors. Due to this measurement noise, the inflight gyro calibration requires the use of a filter to achieve the desired accuracy of 0.1 degrees per hour.

A general linear system as represented by a vector state difference equation is shown below in Equation (7.2.1), where $\underline{X}_n$ is a column

$$\underline{X}_{n+1} = \phi_{n+1,\ n}\ \underline{X}_n + \sigma_n\ \underline{U}_n \qquad (7.2.1)$$

vector representing the state of the system, $\phi_{n+1,\ n}$ is the transition matrix from state $\underline{X}_n$ to $\underline{X}_{n+1}$, $\underline{U}_n$ is a general noise vector and σ_n is the corresponding distribution matrix. In the calibration problem considered here, $\underline{X}_n$ represents the coordinate misalignments and instrument error sources which are to be determined. The noise vector $\underline{U}_n$ corresponds to the gyro noise. In this case, the observables $\underline{Y}_n$ are assumed to be related to $\overline{X}_n$ by the linear relationship.

$$\underline{Y}_n = M_n \underline{X}_n + \delta\ \underline{Y}_n \qquad (7.2.2)$$

with $\delta\ \underline{Y}_n$ representing observation noise such as quantization errors. Letting the estimate of the state vector at t_n be denoted by $\hat{\underline{X}}_n$, the general unbiased linear estimate of $\underline{X}_n$ is

$$\hat{\underline{X}}_n = \hat{X}'_n + b_n [\underline{Y}_n - \hat{Y}'_n] \qquad (7.2.3)$$

where $\hat{\underline{X}}'$ is the predicted (extrapolated) value of $\hat{\underline{X}}_{n-1}$

$$\hat{\underline{X}}' = \phi_{n,n-1} \hat{\underline{X}}_{n-1} + \sigma_{n-1} \underline{U}_{n-1} \qquad (7.2.4)$$

and

$$\hat{\underline{Y}}'_n = M_n \hat{\underline{X}}'_n \qquad (7.2.5)$$

and $\underline{U}_{n-1}$ is given by

$$\hat{\underline{U}}_{n-1} = \overline{\underline{U}}_{n-1} = 0 \qquad (7.2.6)$$

$$(\overline{\underline{Z}} = \text{ensemble average of } \underline{Z})$$

In the case of the gyro calibration filter, several simplifying assumptions were made at the outset. The first is that the attitude errors E_x, E_y, E_z about the LM body axes are the integrals of the total gyro drift rates ϵ_x, ϵ_y, ϵ_z about the corresponding axes. Thus, Equation (7.2.1) becomes

$$\begin{bmatrix} E_j & (n+1) \\ \epsilon_j & (n+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} E_j & (n) \\ \epsilon_j & (n) \end{bmatrix} \qquad (7.2.7)$$

where j = X, Y, or Z.
The observable in this case is simply the attitude error E_j (n+1) measured with respect to the PGNCS. Thus, Equation (7.2.2) becomes

$$[E_j\ (n)] = [1 \quad 0] \begin{bmatrix} E_j & (n) \\ \epsilon_j & (n) \end{bmatrix} + \delta E_j(n) \qquad (7.2.8)$$

where j = X, Y, or Z.

If a linear filter such as (7.2.3) is used for updating the estimate of gyro drift rate and attitude error, we have

$$\begin{bmatrix} \hat{E}_j\ (n+1) \\ \hat{\epsilon}_j\ (n+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{E}_j\ (n) \\ \hat{\epsilon}_j\ (n) \end{bmatrix} + \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} [E_j(n+1) - \hat{E}_j(n) - \Delta t \hat{\epsilon}_j(n)] \qquad (7.2.9)$$

For the LM AGS inflight gyro calibration, the first step performed is to compute the PGNCS direction cosines from the PGNCS Euler angles θ_p, ϕ_p, ψ_p as shown in Equation (7.2.10)

$$\begin{aligned} a_{11D} &= \cos \psi_p \cos \theta_p \\ a_{12D} &= \sin \psi_p \\ a_{13D} &= - \cos \psi_p \sin \theta_p \\ a_{31D} &= \sin \phi_p \sin \psi_p \cos \theta_p + \cos \phi_p \sin \theta_p \\ a_{32D} &= - \sin \phi_p \cos \psi_p \\ a_{33D} &= - \sin \phi_p \sin \psi_p \sin \theta_p + \cos \phi_p \cos \theta_p \qquad (7.2.10) \end{aligned}$$

These 6 direction cosines define the position of the X and Z body axes in the PGNCS, and will be used to compute the attitude errors for the gyro calibration, except during the first cycle through the calibration equations. During this cycle, the AGS direction cosines are set equal to the PGNCS direction cosines with IMU align equations.

During each subsequent cycle through the calibration equations, the vehicle attitude errors are computed from the AGS and PGNCS direction cosines as shown in Equation (7.2.11). These attitude errors

$$\begin{aligned} E_x &= - Y_b \cdot Z_D = - [a_{21}a_{31D} + a_{22}a_{32D} + a_{23}a_{33D}] \\ E_y &= - Z_b \cdot X_D = - [a_{31}a_{11D} + a_{32}a_{12D} + a_{33}a_{13D}] \\ E_z &= Y_b \cdot X_D = [a_{21}a_{11D} + a_{22}a_{12D} + a_{23}a_{13D}] \qquad (7.2.11) \end{aligned}$$

as computed are the sines of the negative angle drifted about each axis. Since these drift angles are small, the compensation corrections are updated as shown in Equation (7.2.12).

$$K_1^1 = K_1^1 + K_{34}^1 E_x$$
$$K_6^1 = K_6^1 + K_{34}^1 E_y$$
$$K_{11}^1 = K_{11}^1 + K_{34}^1 E_z \qquad (7.2.12)$$

where

K_1^1, K_6^1, K_{11}^1 are the X, Y, Z gyro compensation constants.

K_{34}^1 is the compensation correction gain, and equals 2×10^{-5}.

Small alignment corrections are also computed based on the attitude errors and are the following equations:

$$\Delta\alpha_{xA} = K_{33}^1 E_X$$
$$\Delta\alpha_{yA} = K_{33}^1 E_Y$$
$$\Delta\alpha_{zA} = K_{33}^1 E_Z \qquad (7.2.13)$$

where

$\Delta\alpha_{xA}$, $\Delta\alpha_{yA}$, $\Delta\alpha_{zA}$ are alignment corrections about the X, Y, Z body axes.

K_{33}^1 is the attitude correction constant, and at present has the value of 0.08.

The calibration at present is performed at a 2-second cycle rate. The effective gains for the calibration are 10^{-3} for K_{34}^1 and 0.08 for K_{33}^1 as the alignment corrections are applied once each calibration cycle, and the compensation corrections are applied each 20 mseconds.

The transient response for this filter is shown in Figure 7.2, and converges to the desired value in approximately 200 seconds. However, this

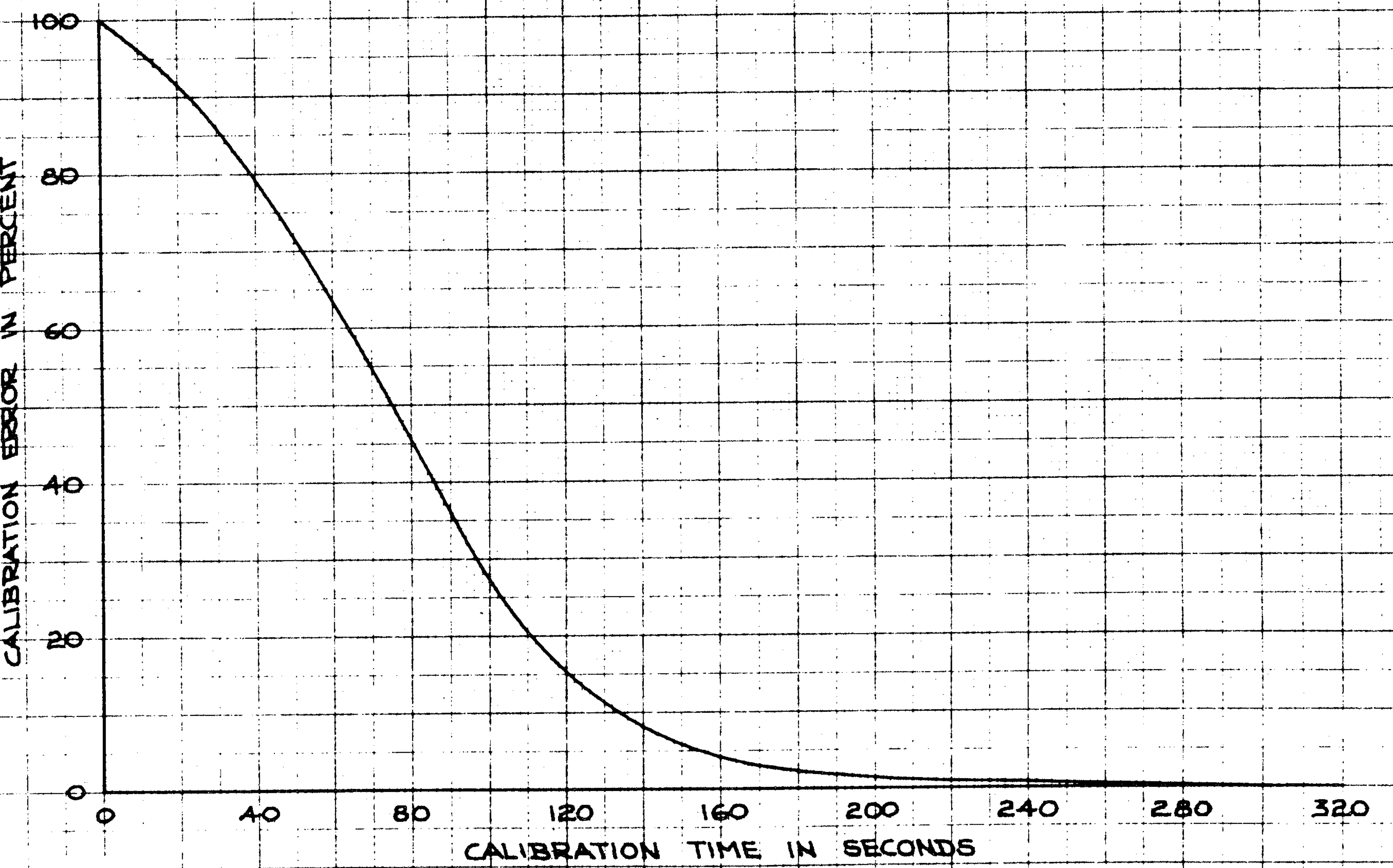


FIGURE 7.2 THEORETICAL CALIBRATION ERROR IN PERCENT VERSUS CALIBRATION TIME

transient response curve does not include the PGNCS quantization noise or the AGS quantization errors which cause a delay in the calibration response as well as an oscillatory type of error in the steady state value.

7.2.2 Accelerometer Inflight Calibration

The AGS accelerometers are calibrated during free fall conditions for bias errors. That is, the outputs of the accelerometers are accumulated for 76 seconds, and are divided by the time interval to obtain the accelerometer bias error. Since the accelerometer bias corrections are used every 20 mseconds in feet per second, the accelerometer bias corrections K^1_{19}, K^1_{21}, and K^1_{23} are computed using the gain shown below in Equation (7.2.14). This gain constant is 0.020/76 which converts the accumulated velocities to the amount of feet per second error the accelerometer bias will cause in 20 mseconds.

$$
\begin{aligned}
K^1_{19} &= K^1_{19} + 2.63158 \times 10^{-4}\ S\Delta V_x \\
K^1_{21} &= K^1_{21} + 2.63158 \times 10^{-4}\ S\Delta V_y \\
K^1_{23} &= K^1_{23} + 2.63158 \times 10^{-4}\ S\Delta V_z
\end{aligned}
\qquad (7.2.14)
$$

where

K^1_{19}, K^1_{21}, K^1_{23} are the X, Y, Z accelerometer compensation values

$S\Delta V_x$, $S\Delta V_y$, $S\Delta V_z$ are the X, Y, Z accelerometer outputs accumulated for 76 seconds.

7.2.3 Gyro Lunar Calibration

The equations mechanized for the lunar calibration are exactly the same as the equations previously described in Section 7.2.1. The only difference between the two calibrations is that angular increments which compensate for the lunar rotation rate are included in the transformation matrix update computations. These equations are approximate corrections, and are shown in Equation (7.2.15).

$$\begin{aligned}\Delta\alpha_{x\ rem} &= \Delta\alpha_{x\ rem} - K^1_{56}\ a_{12}\\ \Delta\alpha_{y\ rem} &= \Delta\alpha_{y\ rem} - K^1_{56}\ a_{22}\\ \Delta\alpha_{z\ rem} &= \Delta\alpha_{z\ rem} - K^1_{56}\ a_{32}\end{aligned} \qquad (7.2.15)$$

where

K^1_{56} is the product of the lunar rotation rate and 20 msecond compute cycle period $\Lambda_M\ \Delta t$ (Radians)

a_{12}, a_{22}, a_{32} are the LM AGS direction cosines which define the inertial Y axis with respect to the X_b, Y_b, Z_b body axes.

$\Delta\alpha_{g\ rem}$ is the angle increment rotated about the g^{th} body axis. This variable has a range of $\pm\ 2^{-13}$ radians with a quantization of 2^{-30} radians, and is used to accumulate small angular increments to increase the accuracy of the LM AGS direction cosines. These angle increments are only used when they exceed 2^{-16} radians.

When the LM is on the lunar surface, it is aligned to a selenocentric coordinate system with the X axis along the lunar local vertical positive outward from the lunar center, and the Z axis obtained by crossing the unit angular momentum vector of the CSM orbit with a unit vector along the X axis. The Y axis is also defined by Y = Z x X and is approximately aligned with the unit angular momentum vector of the CSM orbit.

The equations used to compensate for the lunar rotation rate were designed for the Y axis to be collinear with the lunar rotation axis. However, since the CSM orbit may be inclined to the lunar equator by as much as 10 degrees, the lunar rate compensation may be in error by 0.1 degrees per hour.

7.3 Navigation

7.3.1 Direction Cosine Updating Algorithm

In this section, the derivation of the algorithm used to update the direction cosines (Figure 3.4, Reference 1) is presented. The strapped down system used in the LM/AGS consists of three gyros mounted on the body of the LM. These gyros measure the integral of the turning rate about the three axes. Unit vectors along these three mutually orthogonal axes are denoted by $\underline{X}_b$, $\underline{Y}_b$, $\underline{Z}_b$ respectively.

The navigation equations operate in a selenocentric inertial coordinate system with unit vectors X, Y, Z respectively. In order to maintain knowledge of the LM attitude in the inertial coordinate system a matrix A of direction cosines is maintained. Thus, A is defined as

$$A = \begin{bmatrix} \underline{X}_b \cdot \underline{X} & \underline{X}_b \cdot \underline{Y} & \underline{X}_b \cdot \underline{Z} \\ \underline{Y}_b \cdot \underline{X} & \underline{Y}_b \cdot \underline{Y} & \underline{Y}_b \cdot \underline{Z} \\ \underline{Z}_b \cdot \underline{X} & \underline{Z}_b \cdot \underline{Y} & \underline{Z}_b \cdot \underline{Z} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (7.3.1)$$

Rotations about the vehicle body axes must be reflected in a change in the A matrix.

The derivative of A is given by

$$\dot{A} = \omega A \quad (7.3.2)$$

where

$$\omega = \begin{bmatrix} 0 & \omega_z & -\omega_y \\ -\omega_z & 0 & \omega_x \\ \omega_y & -\omega_x & 0 \end{bmatrix} \quad (7.3.3)$$

and ω_x, ω_y, ω_z are the rotation rates about the X, Y, and Z body axis respectively.

Assume now that the matrix A is known at the $(n-1)^{th}$ computing interval. Then A at the n^{th} computing interval can be obtained as the result of a Taylor series expansion about A_{n-1}. Thus,

$$A_n = A_{n-1} + \dot{A}_{n-1}\,\Delta t + \frac{1}{2}\,\ddot{A}_{n-1}\,\Delta t^2 + \frac{1}{6}\,\dddot{A}_{n-1}\,\Delta t^3 + \ldots \qquad (7.3.4)$$

Upon substituting 7.3.2 into 7.3.4, there results (up to third order)

$$A_n = \left[I + \omega\Delta t + \frac{1}{2}\,\Delta t^2(\dot{\omega} + \omega^2) + \frac{\Delta t^3}{6}\,(\ddot{\omega} + 2\dot{\omega}\omega + \omega\dot{\omega} + \omega^3)\right]A_{n-1} \qquad (7.3.5)$$

where I is the identity matrix

Δt is the time between the $(n-1)^{th}$ and n^{th} computing points.

The gyro outputs at the n^{th} computing increment are

$$\Delta\alpha_i = \int_{(n-1)\Delta t}^{n\Delta t} \omega_i\,dt \qquad i = x,\ y,\ z \qquad (7.3.6)$$

If (7.3.3) is integrated as in (7.3.6), there results the matrix

$$\Delta\alpha = \begin{bmatrix} 0 & \Delta\alpha_z & -\Delta\alpha_y \\ -\Delta\alpha_z & 0 & \Delta\alpha_x \\ \Delta\alpha_y & -\Delta\alpha_x & 0 \end{bmatrix} \qquad (7.3.7)$$

Now $\Delta\alpha$ is expanded in a Taylor series (to third order) and

$$\Delta\alpha = \omega\Delta t + \frac{1}{2}\,\dot{\omega}\Delta t^2 + \frac{1}{6}\,\ddot{\omega}\Delta t^3 \qquad (7.3.8)$$

Substituting 7.3.8 and 7.3.7 into 7.3.5 yields

$$A_n = \left[I + \Delta\alpha + \frac{\Delta\alpha^2}{2} + \frac{\Delta t^3}{12}\,(\dot{\omega}\omega - \omega\dot{\omega} + 2\omega^3)\right] A_{n-1} \qquad (7.3.9)$$

if terms on the order of Δt^4 and higher are neglected. The algorithm used for implementation in the guidance equations (Figure 3.4) is

$$A_n = \left[I + \Delta\alpha + \frac{\Delta\alpha^2}{2}\right] A_{n-1} \qquad (7.3.10)$$

so that the error up to terms of fourth order in Δt is

$$\epsilon A_n = \left[\frac{(\dot{\omega}\omega - \omega\dot{\omega})}{12} + \frac{\omega^3}{6} \right] A_{n-1} \Delta t^3 \tag{7.3.11}$$

To counter computer round-off, orthonormality corrections are derived as follows. Let $\underline{a}_1 = (a_{11}, a_{12}, a_{13})$ and $\underline{a}_3 = (a_{31}, a_{32}, a_{33})$. Because of computational errors, these vectors are not exactly of unit length nor orthogonal to each other. First compute first order corrections E_1 and E_3 so that $\underline{a}_1$ and $\underline{a}_3$ are normalized without a change in orientation; i.e.,

$$|(1 + E_1)\, \underline{a}_1| = 1 \tag{7.3.12}$$

Squaring both sides,

$$(1 + E_1)^2\, \underline{a}_1 \cdot \underline{a}_1 = 1$$

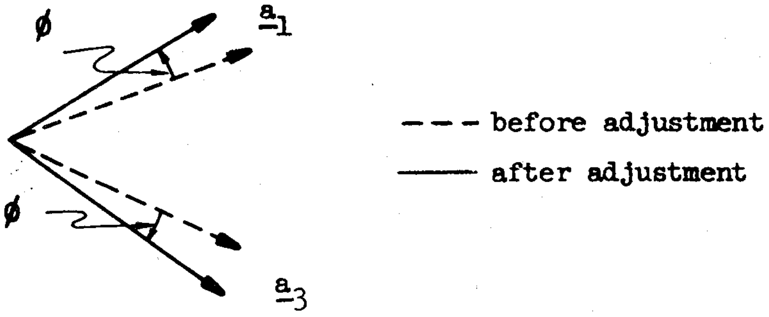
$$E_1 \approx \frac{1}{2} (1 - \underline{a}_1 \cdot \underline{a}_1) \tag{7.3.13}$$

Likewise,

$$E_3 \approx \frac{1}{2} (1 - \underline{a}_3 \cdot \underline{a}_3) \tag{7.3.14}$$

To gain orthogonality, $\underline{a}_1$ and $\underline{a}_3$ are rotated away from each other in their plane through equal angles until they are perpendicular to each other.

$$E_{13} \overset{\Delta}{\equiv} \sin\phi \approx \frac{1}{2}\, \underline{a}_1 \cdot \underline{a}_3$$



Hence, the first order orthonormality corrections to $\underline{a}_1$ and $\underline{a}_3$ are:

$$\underline{a}_1 \leftarrow \underline{a}_1 + E_1\underline{a}_1 - E_{13}\underline{a}_3$$

$$\underline{a}_3 \leftarrow \underline{a}_3 + E_3\underline{a}_3 - E_{13}\underline{a}_1 \qquad (7.3.15)$$

Vector $\underline{a}_2 = (a_{21}, a_{22}, a_{23})$ is then computed as $\underline{a}_2 = \underline{a}_3 \times \underline{a}_1$.

7.3.2 Derivation of the Radar Filter Equations

This section contains a derivation of the radar filter equations. Unless indicated, the symbolism in the derivation does not correspond to that of the flight equations.

DYNAMIC FILTER EQUATIONS

Consider the vector difference equation which represents the actual dynamics of a system.

$$X_{n+1} = U(t_{n+1}, t_n) X_n + q_n \qquad (7.3.16)$$

where

X_n is the state of the system at time t_n

U models the dynamics of the system

q_n is noise added to the system at time t_n with the properties

$$E[q_n] = 0; \; E[q_n q_j^T] = 0 \text{ for } n \neq j; \; E[q_n q_n^T] = Q_n$$

T signifies transpose and E the expected value operation.

If a measurement

$$y_n^* = M_n X_n + S_n \qquad (7.3.17)$$

where S_n is noise with the properties

$$E[S_n] = 0; \; E[S_n S_j^T] = 0 \text{ for } n \neq j; \; E[S_n S_n^T] = \tilde{S}_n$$

is made at time n then the best, linear, unbiased estimate of X_n denoted by $\hat{X}_n$ is given by

$$\hat{X}_n = B_n y_n^* + C_n X_{n/n-1} \qquad (7.3.18)$$

where

$$B_n = P_n M_n^T \{M_n P_n M_n^T + \tilde{S}_n\}^{-1} \qquad (7.3.19)$$

$$C_n = I - B_n M_n \qquad (7.3.20)$$

$$X_{n/n-1} = U(t_n, t_{n-1}) \hat{X}_{n-1} \qquad (7.3.21)$$

$$P_n = U_n P_{n-1/n-1} U_n^T + Q_n \qquad (7.3.22)$$

$$P_{n/n} = C_n P_n \qquad (7.3.23)$$

In words, $X_{n/n-1}$ is the value of the state at time n as obtained from the value of the best estimate of the state at time n-1. $X_{n/n-1}$ is sometimes called the "a priori" estimate of the state at time n. $P_{n/n}$ is the value of the error covariance matrix after the measurement and "a priori" estimate have been combined to form the best estimate $\hat{X}_n$ and P_n is the value of $P_{n-1/n-1}$ updated by the dynamics through time $\Delta t = t_n - t_{n-1}$.

LM FILTER

In constructing the LM filter several assumptions are made. First, the problem can be considered as a one dimensional problem with the resultant filter weights applying to each of the three dimensions. Thus, denoting (in one dimension) p for position and $\dot{p}$ for velocity, the difference equations of motion for each vehicle (LM and CSM) can be expressed as

$$\begin{aligned} p_{n+1} &= p_n + \dot{p}_n \Delta t + \frac{1}{2} a_n \Delta t^2 \\ \dot{p}_{n+1} &= \dot{p}_n + a_n \Delta t \end{aligned} \qquad (7.3.24)$$

a_n is the acceleration due to gravity at time t_n.

Corresponding to Equation (7.3.16), the state vector is taken to be

$$X = \begin{bmatrix} p_L - p_c \\ \dot{p}_L - \dot{p}_c \end{bmatrix} \qquad (7.3.25)$$

where $_L$, c designate LM and CSM respectively. It is assumed that p_c and $\dot{p}_c$ are known exactly. Substituting (7.3.24) into (7.3.25) yields

$$X_{n+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} X_n + \Delta t \begin{bmatrix} (a_{L,n} - a_{c,n}) \frac{\Delta t}{2} \\ a_{L,n} - a_{c,n} \end{bmatrix} \qquad (7.3.26)$$

If it is assumed that gravity acts the same on each vehicle, this equation reduces to

$$X_{n+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} X_n \qquad (7.3.27)$$

Corresponding to equation 7.3.16 then

$$U(t_n, t_{n-1}) = U\ (\Delta t) = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \qquad (7.3.28)$$

Since gravity does not act the same on each vehicle q_n is taken as noise with covariance matrix (in the nomenclature of the flight equations)

$$Q_n = \begin{bmatrix} 2^9 & 0 \\ 0 & K_{10}^2\ \Delta t \end{bmatrix} \qquad (7.3.29)$$

The term $K_{10}^2\ \Delta t$ is an experimentally determined term representing noise on velocity and 2^9 is used to maintain the matrix P_n (see equation 7.3.22) positive definite (2^9 is one quantum of the element P_{11} of P_n). This essentially means that P_{11} is rounded upward rather than downward. From equation 7.3.22 then, the terms of P_n are given by

$$\begin{aligned} P_{11} &= P_{11} + 2\ P_{12}\ \Delta t + P_{22}\ \Delta t^2 + 2^9 \\ P_{12} &= P_{12} + P_{22}\ \Delta t \\ P_{22} &= P_{22} + K_{10}^2\ \Delta t \end{aligned} \qquad (7.3.30)$$

The measurement used is that of relative position of the LM and CSM only so that

$$y_n^* = [1\ 0]\ X_n + S_n \qquad (7.3.31)$$

and

$$M = [1\ 0]$$

It follows from Equation (7.3.19) that

$$B_n = \begin{bmatrix} \dfrac{P_{11}}{P_{11} + \tilde{S}_n} \\ \\ \dfrac{P_{12}}{P_{11} + \tilde{S}_n} \end{bmatrix} \tag{7.3.32}$$

and from equation (7.3.20) that

$$C_n = \begin{bmatrix} 1 - \dfrac{P_{11}}{P_{11} + \tilde{S}_n} & 0 \\ \\ \dfrac{P_{12}}{P_{11} + \tilde{S}_n} & 1 \end{bmatrix} \tag{7.3.33}$$

Following the nomenclature of the flight equations, let

$$\sigma^2 = P_{11} + \tilde{S}_n = P_{11} + K_3^3 R^2 + K_7^3 \tag{7.3.34}$$

where R is the relative position of the CSM with respect to the LM. Then

$$W_1 = \frac{P_{11}}{\sigma^2} \tag{7.3.34}$$

$$W_2 = \frac{P_{12}}{\sigma^2}$$

The terms K_7^3 and $K_3^3R^2$ model the variances of the radar measurement errors.

Rearranging equation (7.3.18) yields

$$\hat{X}_n = B_n[M_n(X_n - X_{n/n-1}) + S_n] + I\,X_{n/n-1} \tag{7.3.37}$$

The term multiplying B_n is the difference between the actual measurement and what the measurement was expected to be. This term is denoted here by δr (a scalar). It should be noted here that even though a simplified model is used for the radar filter, the term δr is obtained as the difference between the radar measurement of range and the navigation estimate of radar range (using the lunar spherical gravitational model). Substitution of the terms in equation (7.3.37) yields

$$\hat{p}_{L,n} = W_1\,\delta r + p_{L,n}$$

$$\dot{p}_{L,n} = W_2\,\delta r + \dot{p}_{L,n} \tag{7.3.38}$$

Since the same weights are used in all dimensions (δr will be three dimensional) equation (7.3.38) is expanded to the vector flight equations.

$$\underline{r} = W_1\underline{\delta r} + \underline{r}$$

$$\underline{V} = W_2\underline{\delta r} + \underline{V} \tag{7.3.39}$$

The updated value of the covariance matrix is obtained directly from equation (7.3.32) and yields

$$P_{22} = P_{22} - W_2 P_{12}$$

$$P_{12} = P_{12}\,(1 - W_1)$$

$$P_{11} = P_{11}\,(1 - W_1) \tag{7.3.40}$$

In the radar filter, two different values of W_1 (denoted by W_1 and W_1') are shown. The reason for this is because the radar information is not used at the time it is received, but rather at a time when the range measurement has been inserted into the computer via the DEDA. Thus, the current values of position and velocity are being updated based upon radar information valid in the past. Therefore, this radar information should not be weighted as much at the current time as it would have been if used immediately. To see how this weighting is changed, consider how the LM position vector is propagated assuming the filtering was done at the time the radar measurement was taken. Denote filtered quantities of position and velocity by $\hat{\underline{r}}$ and $\hat{\underline{V}}$ respectively. These quantities are obtained in equation (7.3.39). Let the subscript, n, denote the n^{th} 2 second computing increments. Two seconds (1 computing increment) later the estimated value of $\underline{r}$ would be

$$\underline{r}_{n+1} = \hat{\underline{r}}_n + 2\,\hat{\underline{V}}_n \tag{7.3.41}$$

or (utilizing equation 7.3.39)

$$\begin{aligned}\underline{r}_{n+1} &= W_1\underline{\delta r} + \underline{r}_n + 2\,\underline{V}_n + 2\,W_2\,\underline{\delta r}\\ &= \underline{r}_n + 2\,\underline{V}_n + (W_1 + 2\,W_2)\,\underline{\delta r}\end{aligned} \tag{7.3.42}$$

If no filtering had been done, the value of $\underline{r}_{n+1}$ would have been (approximately)

$$\underline{r}_{n+1} = \underline{r}_n + 2\,\underline{V}_n \tag{7.3.43}$$

where the uncapped symbols denote unfiltered quantities.
Therefore, letting $W_1 = W_1 + 2\,W_2$ it is seen that the filtering can be accomplished when the range measurement is entered if W_1 is updated every computing increment and if equation (7.3.42) is valid. This equation becomes invalid after a period of time (not completely determined yet) and thus it has been recommended that the radar range be entered into the computer within 30 seconds of taking the radar measurement.

It should be noted that if only 2 or 3 radar points are taken, the velocity estimate of the LM may deteriorate. For this reason, it is also recommended that at least 5 radar points be used. These restrictions are to be examined in a future analysis.

7.4 GUIDANCE

7.4.1 Attitude Error Commands

In this section the derivation of the attitude steering errors E_x, E_y, E_z is presented for the rendezvous steering mode of operation and for the acquisition steering mode of operation.

RENDEZVOUS STEERING

The thrust direction is given by

$$\underline{X}_t = \underline{X}_b + K_8^4 \underline{Y}_b + K_7^4 \underline{Z}_b \qquad (7.4.1)$$

The constants K_7^4 and K_8^4 produce a prescribed thrust offset with respect to the X-body axis. Steering is accomplished by rotating the LM about an instantaneous axis given by $\underline{X}_t \times \underline{X}_D$ where $\underline{X}_D$ is the desired thrust direction. Note that the magnitude of the cross product above equals the angular error to be steered out. The pitch and yaw errors, E_Y and E_Z are computed by scalar multiplication of the cross product with $\underline{Y}_b$ and $\underline{Z}_b$, respectively.

Hence,

$$E_Y = (\underline{X}_t \times \underline{X}_D) \cdot \underline{Y}_b = (\underline{Y}_b \times \underline{X}_t) \cdot \underline{X}_D \qquad (7.4.2)$$

$$E_Z = (\underline{X}_t \times \underline{X}_D) \cdot \underline{Z}_b = (\underline{Z}_b \times \underline{X}_t) \cdot \underline{X}_D \qquad (7.4.3)$$

which yields

$$E_Y = - \underline{Z}_b \cdot \underline{X}_D + K_7^4 (\underline{X}_b \cdot \underline{X}_D) \qquad (7.4.4)$$

$$E_Z = \underline{Y}_b \cdot \underline{X}_D - K_8^4 (\underline{X}_b \cdot \underline{X}_D) \qquad (7.4.5)$$

The above pitch and yaw equations are used to steer the APS. For the DPS or RCS, the K_7^4 and K_8^4 terms are dropped because these engines are assumed to thrust along the LM X-body axis.

The E_x command is

$$E_X = - \underline{W}_c : \underline{Z}_b - J^4 \qquad (7.4.6)$$

When J^4 is zero, the LM Z-body axis is driven parallel to the CSM orbit plane. A positive numerical value of $- \underline{W}_c \cdot \underline{Z}_b - J^4$ will rotate the vehicle in a right hand sense about the + X-body axis.

J^4 is entered in degrees as the desired LM orientation about the X-body axis. The DEDA conversion routine multiplies by the appropriate scale factor to convert the angle to radians. The radian value used is the desired value that the Z-body axis makes with the CSM orbit plane. The error in this approximation is given in Table 7.4.1.

Table 7.4.1

Error in E_X Made by Using Radian Value of J^4 as Desired Value of a_{32}

Error in E_X (Degrees)	Value of J^4 (Degrees)
.4	20
1.6	30
3.8	40
11.0	50
32.7	57.3

Inputs such that the J^4 radian magnitude is greater than one would cause continual rotation of the vehicle about the X-body axis. In practice J^4 inputs should be less than 40° for the sake of accuracy and to preserve a safe margin of gain in the E_X control loop.

ACQUISITION STEERING

In this mode the Z-body axis is to be pointed in the direction of the CSM. Steering is accomplished by rotating the LM about an instantaneous axis given by

$$\underline{Z}_b \times \underline{Z}_{bD}$$

where $\underline{Z}_{bD}$ is the unit vector in the direction of the CSM. The attitude errors E_x and E_y are determined by scalar multiplication of the cross product with $\underline{X}_b$ and $\underline{Y}_b$ respectively. Thus

$$E_x = (\underline{Z}_b \times \underline{Z}_{bD}) \cdot \underline{X}_b = (\underline{X}_b \times \underline{Z}_b) \cdot \underline{Z}_{bD} \qquad (7.4.7)$$

$$E_y = (\underline{Z}_b \times \underline{Z}_{bD}) \cdot \underline{Y}_b = (\underline{Y}_b \times \underline{Z}_b) \cdot \underline{Z}_{bD} \qquad (7.4.8)$$

which yields

$$E_x = -\underline{Y}_b \cdot \underline{Z}_{bD} \tag{7.4.9}$$

$$E_y = \underline{X}_b \cdot \underline{Z}_{bD} \tag{7.4.10}$$

The E_z command is

$$E_z = \underline{W}_c \cdot \underline{X}_b \tag{7.4.11}$$

which orients the X axis parallel to the CSM orbit plane. 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.

ELLIPSE PREDICTOR SUBROUTINE

The purpose of the ellipse predictor subroutine is to propagate a known position $\underline{r}_1$ and velocity $\underline{V}_1$ around an elliptical orbit to obtain the position, $\underline{r}_2$, and velocity, $\underline{V}_2$, T seconds later (or earlier).

To derive the ellipse predictor equations consider equations (A.12) and (A.13) of Appendix A in the form

$$\underline{r}_2 = x_2\underline{P} + y_2\underline{Q} \tag{7.4.12}$$

$$\underline{V}_2 = \dot{x}_2\underline{P} + \dot{y}_2\underline{Q} \tag{7.4.13}$$

where

$$x_2 = \alpha\ (\cos E_2 - e) \tag{7.4.14}$$

$$y_2 = \alpha\ (1-e^2)^{\frac{1}{2}} \sin E_2 \tag{7.4.15}$$

$$\dot{x}_2 = -\sqrt{\mu\alpha}\ \frac{\sin E_2}{r_2} \tag{7.4.16}$$

$$\dot{y}_2 = [\mu\alpha(1-e^2)]^{\frac{1}{2}} \frac{\cos E_2}{r_2} \tag{7.4.17}$$

To solve for $\underline{P}$ and $\underline{Q}$ utilization is made of these same equations valid at the point 1 where everytning is known. i.e.

$$\underline{r}_1 = x_1\underline{P} + y_1\underline{Q} \tag{7.4.18}$$

$$\underline{V}_1 = \dot{x}_1\underline{P} + \dot{y}_1\underline{Q} \tag{7.4.19}$$

and $\underline{r}_1$, $\underline{V}_1$, x_1, y_1, $\dot{x}_1$, $\dot{y}_1$ are known. Then

$$\underline{P} = \frac{\underline{r}_1\dot{y}_1 - \underline{V}_1 y_1}{x_1\dot{y}_1 - y_1\dot{x}_1} \tag{7.4.20}$$

$$\underline{Q} = \frac{\underline{r}_1\dot{x}_1 - \underline{V}_1 x_1}{x_1\dot{y}_1 - y_1\dot{x}_1} \tag{7.4.21}$$

Substituting equation (7.4.20) and (7.4.21) into (7.4.12) and (7.4.13) yields

$$\underline{r}_2 = \left[\frac{x_2\dot{y}_1 - y_2\dot{x}_1}{x_1\dot{y}_1 - y_1\dot{x}_1}\right]\underline{r}_1 + \left[\frac{y_2x_1 - x_2y_1}{x_1\dot{y}_1 - y_1\dot{x}_1}\right]\underline{V}_1 = f\underline{r}_1 + g\underline{V}_1 \tag{7.4.22}$$

$$\underline{V}_2 = \left[\frac{\dot{x}_2\dot{y}_1 - \dot{y}_2\dot{x}_1}{x_1\dot{y}_1 - y_1\dot{x}_1}\right]\underline{r}_1 + \left[\frac{\dot{y}_2x_1 - \dot{x}_2y_1}{x_1\dot{y}_1 - y_1\dot{x}_1}\right]\underline{V}_1 = \dot{f}\underline{r}_1 + \dot{g}\underline{V}_1 \tag{7.4.23}$$

Substituting equations (7.4.14), (7.4.15), (7.4.16), and (7.4.17) with appropriate subscripts into (7.4.22) and (7.4.23) yields

$$f = \frac{\alpha}{r_1}\left[\cos \Delta E - e \cos E_1\right] \tag{7.4.24}$$

$$g = T + \frac{\alpha^{3/2}}{\sqrt{\mu}}\left(\sin \Delta E - \Delta E\right) \tag{7.4.25}$$

$$\dot{f} = \frac{\sqrt{\mu}}{\sqrt{\alpha}\ r_1} \ \frac{\sin \Delta E}{1 - e \cos E_2} \tag{7.4.26}$$

$$\dot{g} = \frac{\text{Cos } \Delta E - e \cos E_2 \cdot}{1 - e \cos E_2} \tag{7.4.27}$$

where $\Delta E = E_2 - E_1$ and T is the time to go from point 1 to point 2. Quantities such as α, r_1, e cos E_1 are readily computed from the known quantities $\underline{r}_1$ and $\underline{V}_1$. In the above equations there are two unknown quantities ΔE and E_2. This can be reduced to one unknown ΔE by the following:

$$\text{Cos } E_2 = \text{Cos } (E_2 - E_1 + E_1) = \text{Cos } (\Delta E + E_1)$$

$$= \text{Cos } \Delta E \text{ Cos } E_1 - \text{Sin } \Delta E \text{ Sin } E_1$$

Then $\dot{f}$ and $\dot{g}$ assume the form

$$\dot{f} = \frac{\sqrt{\mu}}{\sqrt{\alpha}\ r_1} \ \frac{\text{Sin } \Delta E}{(1 - e \text{ Cos } E_1 \text{ Cos } \Delta E + e \text{ Sin } E_1 \text{ Sin } \Delta E)} \tag{7.4.28}$$

$$\dot{g} = \frac{\text{Cos } \Delta E - e \text{ Cos } E_1 \text{ Cos } \Delta E + e \text{ Sin } E_1 \text{ Sin } \Delta E}{1 - e \text{ Cos } E_1 \text{ Cos } \Delta E + e \text{ Sin } E_1 \text{ Sin } \Delta E} \tag{7.4.29}$$

Equations (7.4.24), (7.4.25), (7.4.28), and (7.4.29) are the equations utilized in the ellipse predictor subroutine on Figure 3.15 of Reference 1.

To obtain ΔE use is made of Keplers equation

$$T = \frac{1}{n} [E_2 - E_1 - e \text{ Sin } E_2 + e \text{ Sin } E_1] \tag{7.4.30}$$

Substituting for Sin E_2 the terms Sin ΔE Cos E_1 + Cos ΔE Sin E_1 yields

$$nT = M = \Delta E - e \text{ Cos } E_1 \text{ Sin } \Delta E - e \text{ Sin } E_1 \text{ Cos } \Delta E + e \text{ Sin } E_1 \tag{7.4.31}$$

or

$$M - \Delta E + e \operatorname{Cos} E_1 \operatorname{Sin} \Delta E + e \operatorname{Sin} E_1 \operatorname{Cos} \Delta E - e \operatorname{Sin} E_1 = 0 \tag{7.4.32}$$

This is a transcendental equation and is solved by using a Newton Raphson technique. That is, if an initial guess is used for the ΔE (the unknown) and substituted into equation (7.4.32) the equation will not equal zero but some other value, say x_{11}. The next value tried for ΔE is obtained from the equation

$$\Delta E = \Delta E + \frac{x_{11}}{x_{12}} \tag{7.4.33}$$

where

$$x_{12} = - \frac{dx_{11}}{d\Delta E} = 1 - e \operatorname{Cos} E_1 \operatorname{Cos} \Delta E + e \operatorname{Sin} E_1 \operatorname{Sin} \Delta E \tag{7.4.34}$$

The convergence of this iteration depends upon the eccentricity of the orbit and the initial guess for ΔE. For the trajectories under consideration in the LM program and if the initial guess of ΔE is chosen to be M then a sufficiently accurate answer for ΔE is obtained in two passes through the iterator.

7.4.3 Pitch Steering Equations for Orbit Insertion

A derivation of the constant $\dddot{r}$ pitch steering law is given below. A similar derivation can be followed for the yaw steering.

The final radial position at orbit insertion, r_f, can be expressed in terms of the present radial position, r, and the time to burnout, T_B, as follows:

$$r_f = r + \dot{r}_A T_B + \frac{1}{2}\ddot{r}_d T_B{}^2 + \frac{1}{6}\dddot{r}_d T_B{}^3 \tag{7.4.35}$$

where $\dot{r}_A$ is the stored value of the present LM velocity $\dot{r}$.

Differentiating equation (7.4.35) with respect to T_B yields

$$\dot{r}_f = \dot{r}_A + \ddot{r}_d T_B + \frac{1}{2}\dddot{r}_d T_B{}^2 \tag{7.4.36}$$

Solving equation (7.4.36) for $\ddot{r}_d$ gives

$$\ddot{r}_d = (\dot{r}_f - \dot{r}_A - \frac{1}{2}\dddot{r}_d T_B{}^2)/T_B \tag{7.4.37}$$

In equations (7.4.35) and (7.4.36), the only unknowns are $\ddot{r}_d$ and $\dddot{r}_d$. Solving equations (7.4.35) and (7.4.36) simultaneously for $\dddot{r}_d$ yields

$$\dddot{r}_d = \frac{12}{T_B{}^3}\left[(\dot{r}_f + \dot{r}_A)\frac{T_B}{2} + r - r_f\right] \tag{7.4.38}$$

Taking the differential of equation (7.4.38) yields

$$\Delta\dddot{r}_d = \frac{12}{T_B{}^3}\left[\frac{T_B}{2}(\Delta\dot{r}_f + \Delta\dot{r}_A) + \Delta r - \Delta r_f\right] \tag{7.4.39}$$

In equation (7.4.39), $\Delta\dot{r}_f = \Delta\dot{r}_A = \Delta r = 0$

So,

$$\Delta\dddot{r}_d = \frac{12}{T_B{}^3}\Delta r_f \tag{7.4.40}$$

But

$$\Delta r_f = r_f - J^{16} - J^5 \tag{7.4.41}$$

where

J^{16} is the desired altitude above the lunar surface at orbit insertion

J^5 is the lunar radius at the landing site.

So

$$\dddot{r}_d = \dddot{r}_d + \frac{12}{T_B^{\;3}} (r_f - J^{16} - J^5) \tag{7.4.42}$$

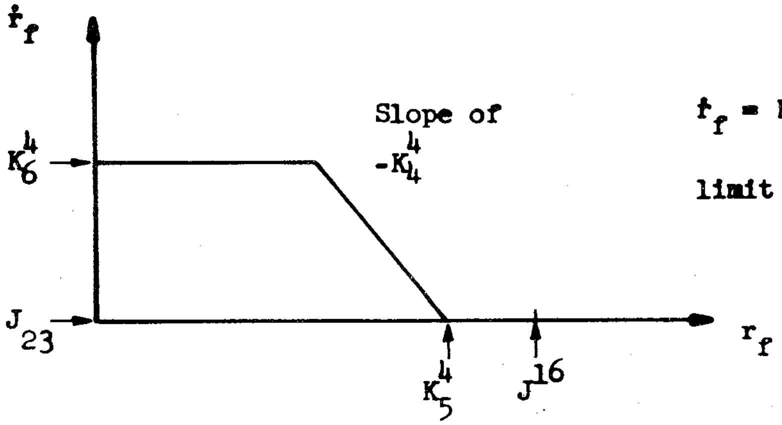
The expression for $\ddot{r}_d$ gives the desired total radial acceleration. The desired radial thrust acceleration must compensate for gravity and centrifugal force. Thus

$$\ddot{r}_d = \ddot{r}_d + \frac{K_1^2}{r^2} - \frac{V_h^2}{r} \tag{7.4.43}$$

The computed value of $\dddot{r}_d$ (and also $\dddot{y}_d$) is limited so that position control by the AGS decreases as V_G decreases. This is desirable for two reasons:

(1) As $V_G \rightarrow 0$, vehicle attitude would otherwise swing rapidly near cutoff in the vain attempt to null trivial altitude (and yaw displacement) errors

(2) The equations are specifically designed not to return to the nominal trajectory in the event of highly perturbed aborts. The vehicle will directly achieve the desired velocity without expending propellant which would otherwise be required for the return to the nominal trajectory. The desired value of $\dot{r}_f$ is computed as a function of r_f.



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

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

$J^{23} = 0$ (for lunar mission)

Function of $\dot{r}_f$ vs. r_f

FIGURE 7.3

The desired value of yaw velocity at burnout (component perpendicular to CSM orbit plane) is zero.

When the AEA indicated altitude is less than J^{21} feet above the landing site and the indicated altitude rate is less than J^{22} fps, then the AGS will command vertical thrusting. (This logic provides for the vertical rise after lunar lift-off and also for emergency abort recovery to avoid impacting the lunar terrain.) Otherwise, the steering proceeds toward the desired burnout velocity and position.

The derivation of the time to engine burnout T_B is based on the "rocket equation".

$$V_G = \int_t^{T_f} a_T dt = \int_t^{T_f} \frac{F}{m_o - \dot{m}t} dt \qquad (7.4.44)$$

where

V_G is the velocity-to-be-gained

a_T is the thrust acceleration

F is force exerted by the engine (assumed constant)

m_o is the initial fuel mass

$\dot{m}$ is the mass flow rate (assumed constant)

t is present time

T_f is the time at orbit insertion

The solution to equation (7.4.44) is

$$\frac{V_G \dot{m}}{F} = \ln \left(\frac{m_o - \dot{m}t}{m_o - \dot{m}T_f}\right) \qquad (7.4.45)$$

or

$$m_o - \dot{m}T_f = (m_o - \dot{m}t) \exp \left[- V_G \dot{m}/_F\right] \qquad (7.4.46)$$

Expanding the exponential term and dropping all terms higher than the cubic gives

$$m_o - \dot{m}T_f = (m_o - \dot{m}t)\ (1 - \frac{V_G\dot{m}}{F} + \frac{V_G^2\dot{m}^2}{2F^2} - \frac{V_G^3\dot{m}^3}{gF^3}\ ) \qquad (7.4.47)$$

Solving equation (7.4.47) for T_f gives

$$T_f = t + (\frac{m_o - \dot{m}t}{F})V_G - (\frac{m_o - \dot{m}t}{2F^2})\ \dot{m}V_G{}^2 + (\frac{m_o - \dot{m}t}{6F^3})\ \dot{m}V_G{}^3 \qquad (7.4.48)$$

But

$$T_B = T_F - t \qquad (7.4.49)$$

and

$$a_T = \frac{F}{m_o - \dot{m}t} \qquad (7.4.50)$$

So equation (7.4.50) reduces to

$$T_B = \frac{V_G}{a_T}\ (1 - V_G\ \frac{\dot{m}}{2F} + V_G{}^2\ \frac{\dot{m}^2}{6F^2}) \qquad (7.4.51)$$

If $K_2^4 = - \frac{\dot{m}}{2F}$ and $K_3^4 = \frac{\dot{m}^2}{6F^2}$

$$T_B = \frac{V_G}{a_T}\ (1 + K_2^4V_G + K_3^4V_G{}^2) \qquad (7.4.52)$$

Notice that $F/_{\dot{m}} = C*$, the effective exhaust velocity of the LM engine. C* is related to engine I_{SP} by the formulae $C* = g_eI_{SP}$ where $g_e = 32.174$.

7.4.4 Derivation of the Error Function for CSI Calculations

The purpose of the CSI calculations is to determine the appropriate CSI maneuver such that the desired line of sight angle between the LM and CSM is achieved at TPI time. The "natural" error function to be minimized in the iteration is the difference between the desired line of sight and the line of sight achieved if the trial CSI maneuver was performed. Computation of the predicted line of sight at TPI time is expensive from the standpoint of the number of computer steps required because in addition to the shuffling of information in and out of the ellipse predictor subroutine a coordinate transformation would have to be performed. A simple alternative solution has been developed based upon the trigonometric relationship involving the desired line of sight, the semi major axis of the CSM trajectory, the differential altitude (Δr) between CSM and LM orbits during the coelliptic coast trajectory and the central angle between the LM and CSM vehicles at TPI time.

What is actually done is the following. Based upon the calculated value of Δr and desired line of sight angle J^2 at TPI time, the desired central angle between the LM and CSM at TPI time can be determined. Denote this angle by θ_D for the moment. Based upon the central angle between the LM and CSM after the CDH maneuver and the known time until the TPI maneuver, the actual central angle $\Delta\theta$ between LM and CSM at TPI time can be determined. The error function is then computed as the absolute value of the difference between θ_D and $\Delta\theta$. The first part of the derivation is the computation of θ_D or $-b_3$ in the nomenclature of the flight equations.

The angle b_3 is the negative of the desired relative central angle of the CSM with respect to the LM at TPI. This angle is a function of Δr and the differential altitude in the coelliptic orbits. The figure below shows the relative positions of the LM and CSM at TPI.

From the law of sines

$$\frac{r_L}{r_E} = \frac{\sin\left(\frac{\pi}{2} - J^2 - \theta_D\right)}{\sin\left(\frac{\pi}{2} + J^2\right)} = \frac{\cos(J^2 + \theta_D)}{\cos J^2} \qquad (7.4.53)$$

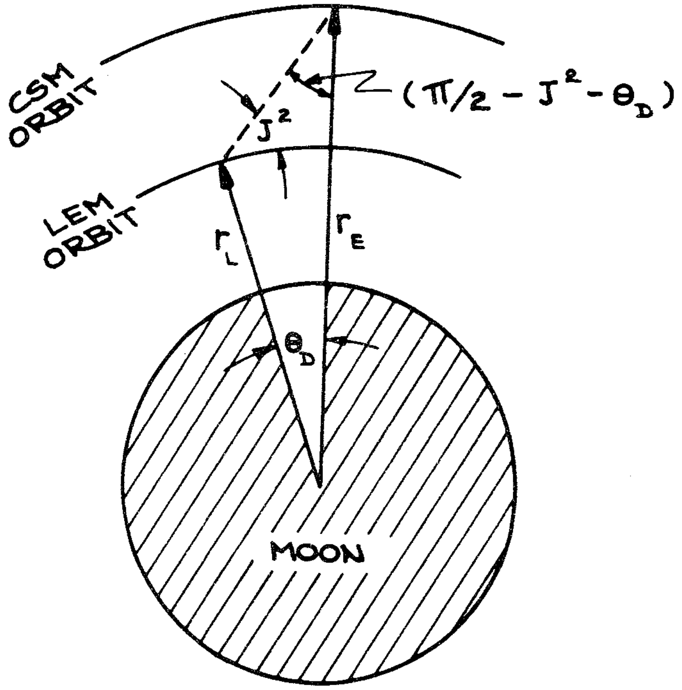


FIGURE 7.4
TPI GEOMETRY

For an elliptical orbit (see Equation A.9 in Appendix A)

$$r = \alpha\ (1 - e \cos E) \tag{7.4.54}$$

Substitution in equation (7.4.53) gives

$$\frac{\alpha_L(1 - e_L \cos E_L)}{\alpha_E(1 - e_E \cos E_E)} = \cos \theta_D - \tan J^2 \sin \theta_D \tag{7.4.55}$$

where the subscript L refers to the LM and the subscript E to the CSM. Since the orbits are coelliptic

$$\alpha_L e_L \cos E_L = \alpha_E e_E \cos E_E \tag{7.4.56}$$

at TPI.

The differential altitude Δr for the coelliptic orbits is given by

$$\Delta r = \alpha_E - \alpha_L \tag{7.4.57}$$

or

$$\alpha_L = \alpha_E - \Delta r$$

The CSM orbit is very nearly circular so

$$e_E \cos E_E \ll 1 \tag{7.4.58}$$

Also assume that

$$\theta_D \ll 1 \tag{7.4.59}$$

Substitution of equation (7.4.57) in equation (7.4.55), expansion of the terms $(1 - e_E \cos E_E)^{-1}$ and $\cos \theta_D$ in series and retention of the first two terms of each series leads to

$$\begin{aligned}(1 - \frac{\Delta r}{\alpha_E})\ (1 - e_L \cos E_L + e_E \cos E_E)& \\ \approx 1 - \theta_D^2 - (\tan J^2)\ \theta_D&\end{aligned} \tag{7.4.60}$$

Assume

$$e_L = e_E \tag{7.4.61}$$

But

$$\begin{aligned}\cos E_E - \cos E_L &= -2 \sin (\frac{E_L + E_E}{2}) \sin (\frac{E_E - E_L}{2}) \\ &\approx -2 \sin (E_E) \sin (\frac{\theta_D}{2}) \\ &\approx -\sin (E_E)\ \theta_D\end{aligned} \tag{7.4.62}$$

Then

$$(1 - \frac{\Delta r}{\alpha_E})\ [1 - (e_E \sin E_E)\ \theta_D] \approx 1 - \frac{\theta_D^2}{2} - (\tan J^2)\ \theta_D \tag{7.4.63}$$

or

$$1 - \frac{\Delta r}{\alpha_E} - (e_E \sin E_E)\ \theta_D = 1 - \frac{\theta_D^2}{2} - (\tan J^2)\ \theta_D \tag{7.4.64}$$

Solving equation (7.4.64) for θ_D leads to

$$\theta_D = \frac{\frac{\Delta r}{\alpha_E}}{\frac{\theta_D}{2} + \tan J^2 - e_E \sin E_E} \tag{7.4.65}$$

For $e_E < .05$ and $\Delta r < 40$ nm, ignoring the $e_E \sin E_E$ term will cause a TPI timing error of no greater than 100 seconds. For circular orbits, the term has value zero. Hence it is dropped from the computation of θ_D.

A first order approximation for θ_D is

$$\theta_D \approx \frac{\Delta r}{(\alpha_E \tan J^2)} \tag{7.4.66}$$

Hence θ_D can be expressed as

$$\theta_D = \frac{\frac{\Delta r}{\alpha_E}}{\frac{\Delta r}{2\alpha_E \tan J^2} + \tan J^2} \tag{7.4.67}$$

The approximation is usable because $\theta_D \ll \tan J^2$.

Let

$b_3 = - \theta_D$

$b_4 = \tan J^2$

$b_5 = - \Delta r/\alpha_E$

then equation (7.4.67) reduces to

$$b_3 = \frac{b_5}{\frac{b_5}{2b_4} - b_4} \tag{7.4.69}$$

The next part of the derivation indicates how the relative central angle between LM and CSM at TPI time is obtained from the relative central angle between LM and CSM at time of CDH.

The true anomaly, θ, can be expressed as an infinite series in the mean anomaly M (see Appendix A)

$$\theta = M + 2e \sin M + \text{higher order terms in } e \tag{7.4.70}$$

The error function is defined as the difference between the desired relative central angle θ_D at TPI and the computed relative central angle, $\Delta\theta$.

The relative central angle may be written as

$$\Delta\theta = \theta_L - \theta_E - \theta_{LO} + \delta\theta_L - \theta_{EO} - \delta\theta_E \qquad (7.4.71)$$

where

$$\delta\theta_L = \theta_L - \theta_{LO} \qquad (7.4.72)$$

θ_{LO}, θ_{EO} are respectively the LM and the CSM true anomalies at CDH. θ_L, θ_E are respectively the LM and the CSM true anomalies at TPI.

$$\Delta\theta = (\theta_{LO} - \theta_{EO}) + (M_L - M_L) + 2e_L \sin M_L - 2e_L \sin M_L + \ldots$$

$$- M_E + M_{EO} - 2e_E \sin M_E + 2e_E \sin M_{EO} + \ldots. \qquad (7.4.73)$$

The terms in e^2 and higher (not actually shown) are dropped. Since

$$M_L - M_{LO} = n_L T_E$$
$$M_E - M_{EO} = n_E t_E \qquad (7.4.74)$$

it follows that

$$\Delta\theta = (\theta_{LO} - \theta_{EO}) + (n_L - n_E)\, T_E$$

$$+ 2e_L \sin M_L - 2e_E \sin M_E$$

$$- 2e_L \sin M_L + 2_E \sin M_{EO} \qquad (7.4.75)$$

For circular CSM orbits, the terms containing e_E and e_L are zero. For non-circular orbits the e_E and e_L terms contribute significantly. In an early formulation of the CSI equations, all terms shown in Equation (7.4.75) were provided. However, to accomodate other functions, later requested, it was agreed not to apply full capability in the CSI equations for handling unexpectedly eccentric orbits and large coelliptic differential altitudes at the same time. To this end, the assumption is now made that at TPI, the angles M_L and M_E are essentially the same, as are the eccentricities. This assumption is based on the fact that for LM/CSM LOS angles greater than 20 degrees, the desired relative central angle at TPI is less than 5 degrees when the coelliptic differential altitude is less than 40 nautical miles. Thus, the terms $2e_L \sin M_L$ and $-2e_E \sin M_E$ are dropped. The angles M_{LO} and M_{EO} are then approximated by the eccentric anomalies E_{LO} and E_{EO}.

A further approximation is made as follows:

$$2e_E \sin E_{EO} - 2e_L \sin E_{LO} = \frac{2r_{EO}\ \dot{r}_{EO}}{\sqrt{K_1^2\ \alpha_E}} - \frac{2r_{LO}\ \dot{r}_{LO}}{\sqrt{K_1^2\ \alpha_L}}$$

$$\approx -2\sqrt{\frac{\alpha_E}{K_1^2}}\ (\dot{r}_{LO} - \dot{r}_{EO}) \qquad (7.4.76)$$

where

r_{LO}, r_{EO} are respectively the radial distances of the LM and the CSM at CDH

$\dot{r}_{LO}$, $\dot{r}_{EO}$ are respectively the radial velocities of the LM and the CSM at CDH

K_1^2 is the lunar gravitational constant

α_L, α_E are respectively the semi-major axis of the LM orbit and the CSM orbit

The cost function is computed using equation (7.4.76)

Let

$$\theta_{LO} - \theta_{EO} = \theta_f$$

$$\dot{r}_f = \dot{r}_{LO}$$

$$\dot{r}_B = \dot{r}_{EO} \qquad (7.4.77)$$

and the desired relative central angle between the LM and the CSM at TPI be $-b_3$, then the cost, C, is

$$C = |b_3 + \theta_f + K_{28}^2\ (\dot{r}_f - \dot{r}_B) + (n_L - n_E)\ T_c| \qquad (7.4.78)$$

where

T_c is the difference in time between the CDH maneuver and the TPI maneuver targeted.

where K_{28}^{2} is the nominal value of $-2\sqrt{\frac{\alpha_E}{K_1^2}}$ (Note that $\sqrt{\frac{K_1^2}{\alpha_E}}$ equals the CSM circular velocity magnitude).

For multiple orbit rendezvous in eccentric orbits (say $e = .03$ and $\Delta r = 40$ nm), the K_{28}^{2} term provides a correction which is not trivial. For eccentricities less than .015, the cost function is accurate for Δr magnitudes of 100 nm or less. If the eccentricities become larger, the allowable Δr magnitude should be decreased accordingly.

7.4.5 Derivation of the Equations to Obtain Coelliptic Orbits

At the coelliptic burn point (CDH), the LM orbit should be adjusted such that the following conditions are true after the CDH burn.

$$\alpha_L = \alpha_E - \Delta r \tag{7.4.79}$$

$$\alpha_L e_L = \alpha_E e_E \tag{7.4.80}$$

$$\theta_L \approx \theta_E \tag{7.4.81}$$

where

α_E, e_E are respectively and the semi-major axis and eccentricity of the CSM orbit

α_L, e_L are respectively the semi-major axis and eccentricity of the LM orbit

Δr is the differential altitude between CSM orbit and LM orbit measured along a radial line through the LM position at CDH

θ_L, θ_E are respectively the true anomalies of the LM and CSM orbits.

If $\dot{r}$, the radial rate of the LM, is chosen properly, then conditions (7.4.79) and (7.4.80) are enough to satisfy (7.4.81). To see this, let E be the eccentric anomaly. For Keplerian orbits (See Appendix A)

$$\cos E_E = \frac{1}{e_E}\left[1 - \frac{r_E}{\alpha_E}\right] \tag{7.4.82}$$

$$\cos E_L = \frac{1}{e_L}\left[1 - \frac{r_L}{\alpha_L}\right] \tag{7.4.83}$$

where

r_E, r_L are respectively the radial distances of the CSM and the LM

E_E, E_L are respectively the eccentric anomalies of the CSM and the LM orbits

Substituting (7.4.79) and (7.4.80) into (7.4.83) and noting that $r_l = r_E - \Delta r$ yields

$$\cos E_l = \frac{\alpha_l - r_l}{\alpha_l e_l} = \frac{\alpha_E - \Delta r - r_E + \Delta r}{\alpha_E e_E} = \frac{\alpha_E - r_E}{\alpha_E e_E} = \cos E_E \qquad (7.4.84)$$

or

$$E_l = \pm E_E \qquad (7.4.85)$$

Thus, if $\dot{r}_l$ is made to have the same sign as $\dot{r}_E$, then $E_l = E_E$.

The relationship between θ and E is given by equation (A.7)

$$\tan \frac{\theta}{2} = \sqrt{\frac{1 + e}{1 - e}} \tan \frac{E}{2} \qquad (7.4.86)$$

Thus, for small values of e the true and eccentric anomalies are essentially the same and hence the coelliptic maneuver aligns the LM's orbital line of apsides with that of the CSM.

At the CDH point, both r_l and α_l are known, so the total desired velocity, V_f, can be obtained from the relation

$$V_f^2 = K_1^2 \left(\frac{2}{r_l} - \frac{1}{\alpha_l}\right) \qquad (7.4.87)$$

where

K_1^2 is the lunar gravitational constant

The desired LM altitude rate is obtained by considering the equations

$$\dot{r}_l r_l = \sqrt{K_1^2 \alpha_l}\, e_l \sin E_l \qquad (7.4.88)$$

$$\dot{r}_E r_E = \sqrt{K_1^2 \alpha_E}\, e_E \sin E_E \qquad (7.4.89)$$

Dividing equation (7.4.88) by equation (7.4.89) yields

$$\frac{\dot{r}_l}{\dot{r}_E} = \frac{r_E}{r_l} \sqrt{\frac{K_1^2 \alpha_l}{K_1^2 \alpha_E}} \; \frac{e_l \sin E_l}{e_E \sin E_E} \qquad (7.4.90)$$

Since

$$E_L = E_E \tag{7.4.91}$$

$$\frac{r_E}{r_L} = \frac{\alpha_E}{\alpha_L} \tag{7.4.92}$$

and

$$\frac{e_L}{e_E} = \frac{\alpha_E}{\alpha_L} \tag{7.4.93}$$

then

$$\frac{\dot{r}_L}{\dot{r}_E} = \sqrt{\frac{K_1^2 \alpha_E^3}{K_1^2 \alpha_L^3}} \tag{7.4.94}$$

The mean orbital angular rates are

$$n_L = \sqrt{\frac{K_1^2}{\alpha_L{}^3}} \tag{7.4.95}$$

$$n_E = \sqrt{\frac{K_1^2}{\alpha_E{}^3}} \tag{7.4.96}$$

It follows that:

$$\dot{r}_L = \frac{n_L}{n_E} \; \dot{r}_E \tag{7.4.97}$$

7.4.6 Derivation of p-Iterator Equations

The purpose of the p-iterator routine is to determine the trajectory that passes between two points ($\bar{r}_5$, $\bar{r}_T$) in a given time T. With this information the impulsive velocity required to achieve this trajectory when the LM reaches the point $\bar{r}_5$ can be determined.

For this discussion, the initial point will be designated with the symbol i and the final point with the letter f. The p-iterator function is to drive the value of T_p in Kepler's equation (7.4.98) to the desired value T.

$$t_f - t_i = T_p = \frac{1}{n}\left[E_f - E_i - e(\sin E_f - \sin E_i)\right] \qquad (7.4.98)$$

To compute T_p, the quantities $\Delta E = E_f - E_i$, $e \sin E_f$, $e \sin E_i$ and n must be obtained. Several relations from the Appendix are rewritten for current use.

$$r = \alpha(1 - e \cos E) \qquad (7.4.99)$$

$$p = r(1 + e \cos E) \qquad (7.4.100)$$

$$\sin E = \frac{r}{p}(1 - e^2)^{\frac{1}{2}} \sin \Theta \qquad (7.4.101)$$

$$n = \sqrt{K_1^2/\alpha^3} \qquad (7.4.102)$$

$$\cos E = \frac{r}{p}(\cos \Theta + e) \qquad (7.4.103)$$

ΔE is computed in the following manner which yields a solution that is not indeterminate when e approaches zero

$$\Delta E = \tan^{-1}\frac{\sin \Delta E}{\cos \Delta E} = \tan^{-1}\left[\frac{\sin(E_f - E_i)}{\cos(E_f - E_i)}\right]$$

$$= \tan^{-1}\left[\frac{\sin E_f \cos E_i - \cos E_f \sin E_i}{\cos E_f \cos E_i + \sin E_f \sin E_i}\right] \qquad (7.4.104)$$

Substituting (7.4.101) and (7.4.103) into (7.4.104) yields

$$\Delta E = \tan^{-1} \frac{\frac{r_f r_i}{p} \sqrt{1-e^2} \left\{ \sin(\Theta_f - \Theta_i) + e \sin \Theta_f - e \sin \Theta_i \right\}}{\frac{r_f r_i}{p} \left\{ \cos(\Theta_f - \Theta_i) + (e^2 + e \cos \Theta_i + e \cos \Theta_f - e^2 \sin \Theta_f \sin \Theta_i) \right\}} \quad (7.4.105)$$

where in the notation of the flight equations

$$\sin(\Theta_f - \Theta_i) = c_2 \quad (7.4.106)$$

$$e \sin \Theta_f = x_4 \quad (7.4.107)$$

$$e \sin \Theta_i = x_3 \quad (7.4.108)$$

$$\cos(\Theta_f - \Theta_i) = c_1 \quad (7.4.109)$$

$$e \cos \Theta_i = x_1 \quad (7.4.110)$$

$$e \cos \Theta_f = x_2 \quad (7.4.111)$$

Equations (7.4.110) and (7.4.111) are obtained directly from equation (7.4.100)

Define

$$\underline{U}_1 = \frac{\underline{r}_i}{|\underline{r}_i|} \quad (7.4.112)$$

and

$$\underline{U}_2 = \frac{\underline{r}_f}{|\underline{r}_f|} \quad (7.4.113)$$

Then

$$c_1 = \underline{U}_1 \cdot \underline{U}_2 \quad (7.4.114)$$

$$c_2 = 1 - c_1^2 \quad (7.4.115)$$

The sign of c_2 is the sign of the component along the y inertial axis of $(W_1 \cdot W_c)$ where

$$\underline{W}_1 = \underline{U}_1 \times \underline{U}_2 \qquad (7.4.116)$$

and

$$\underline{W}_c = \text{the CSM angular momentum unit vector.}$$

The following procedure is used to obtain x_3

$$e \cos \Theta_f = e \cos (\Theta_f - \Theta_i + \Theta_i) = \cos (\Theta_f - \Theta_i) \cos \Theta_i$$
$$- e \sin (\Theta_f - \Theta_i) \sin \Theta_i \qquad (7.4.117)$$

or

$$x_2 = c_1 x_1 - c_2 x_3 \qquad (7.4.118)$$

Thus

$$x_3 = \frac{c_1 x_1 - x_2}{c_2} \qquad (7.4.119)$$

x_4, which is obtained in the same manner is given by

$$x_4 = \frac{x_1 - c_1 x_2}{c_2} \qquad (7.4.120)$$

The quantities e sin E_f and e sin E_i are obtained by substitution in equation (7.4.101) and multiplication by e.

$$e \sin E_f = \frac{r_f}{p} (1 - e^2)^{\frac{1}{2}} e \sin \Theta_f \qquad (7.4.121)$$

or

$$x_9 = \frac{r_f}{p} (1 - e^2)^{\frac{1}{2}} x_4 \qquad (7.4.122)$$

and

$$e \cos E_i = \frac{r_i}{p} (1 - e^2)^{\frac{1}{2}} e \sin \Theta_i \qquad (7.4.123)$$

or

$$x_8 = \frac{r_i}{p} (1 - e^2)^{\frac{1}{2}} x_3 \qquad (7.4.124)$$

Thus

$$T_p = \frac{1}{n} (\Delta E + x_8 - x_9) \qquad (7.4.125)$$

T_p is made to approach T by iterating on the parameter p. Two values of p are used to obtain two values of T_p. With this information, the partial derivative of T_p with respect to p is obtained. Succeeding values for the parameter p are obtained using the Newton-Raphson formula

$$p = p + \frac{(T - T_p)}{\frac{\partial_p}{\partial_T}} \qquad (7.4.126)$$

The actual iteration control logic is shown near the bottom for Figure 3.23 in Reference (1). μ_3 counts the number of iterations performed. When $\mu_3 \geq 2$, two previous trial values of T_p are retained as T_p and T_p', and the two previous trial values of p are retained as p and p'. These quantities are used to compute the derivative required in the iteration scheme (equation 7.4.126). If μ_3 is less than 2K17 (8 iterations), the iterations continue. If μ_3 is 2, the quantity ∂_T is computed. This means that in each computation cycle, the partial derivative is computed at least once. As the trial solutions approach the desired solution, large errors can be obtained in the quantity ∂_T because p approaches p' and T_p approaches T_p'. Thus, after the derivative has been computed once (when $\mu_3 = 2$), a check is made on $|T_p - T_p'|$. If this number is larger than K_{18}^2, the derivative is computed. If not, the last value of the derivative is used in the computation of the increment for p. The increment is computed as

$$\Delta p = (T - T_p)\, \partial_T$$

and Δp is limited to be less than 2^{19} (524288 ft) in magnitude. This is done for the following reason. When the iteration begins, the value of Δp could be quite large because these initial quantities were obtained with just a guess of the ultimate solution. If p changes too much, it is possible for the scaling in the computer to be exceeded. In this case, no solution would be obtained when an actual solution may exist. Thus, the magnitude of Δp is limited.

When 2K17 iterations have been performed, a check is made of the final solution against the desired solution. The difference between T and T_p is checked against the quantity 2K20 (2 sec). If the difference is greater than 2K20, it is assumed the iteration has not converged. If less than 2K20, the iteration has converged to the desired answer and the logic flow continues to Figure 3.24 where the initial and final velocities required to rendezvous are computed.

Current design is to use eight iterations since there is ample time for this number to be done in the 2-second computing interval. For the trajectories under consideration in the LM program, the minimum that should ever be used is five.

APPENDIX A

GENERAL DISCUSSION OF ORBITAL MECHANICS

This section contains a summary of the more important equations of orbital mechanics used in the derivations in the preceding sections. Derivation of these equations and detailed discussions can be found in any of the books on Celestial Mechanics or Astrodynamics (e.g., References 5 and 6).

Consider Figure A1 which depicts a vehicle in free flight above a spherical attracting body which for our purposes is the Moon. The vehicle flies an elliptical trajectory with the position vector $\underline{r}$ and velocity vector $\underline{V}$ both contained in the orbit plane. The vector normal to the orbit plane defined by

$$\underline{h} = \underline{r} \times \underline{V} \tag{A.1}$$

is called the angular momentum vector and the magnitude of $\underline{h}$ is the angular momentum per unit mass. The center of the Moon occupies one focus of the ellipse. The point of closest approach to the Moon is called pericynthion and 180° from pericynthion is apocynthion, the point of greatest distance from the Moon. The central angle between the position of the vehicle and pericynthion is called the true anomaly and is denoted by θ. The semi-latus rectum or parameter of the ellipse is defined by

$$p = \frac{h^2}{\mu} \qquad h = |\underline{h}| \tag{A.2}$$

where μ is the gravitational constant. The velocity at any point on the ellipse can be obtained from the relationship

$$V^2 = \mu \left(\frac{2}{r} - \frac{1}{\alpha} \right) \tag{A.3}$$

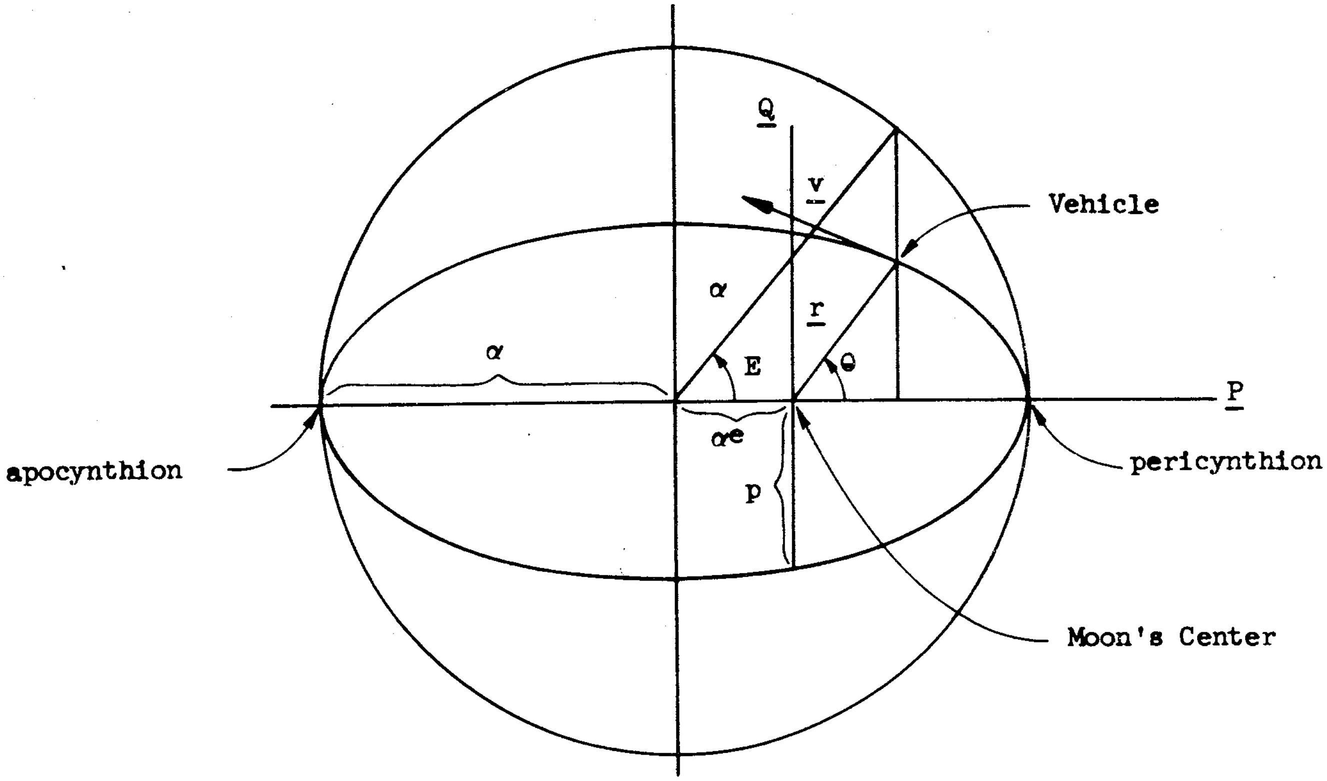
where

$r = |\underline{r}|$

α is the semi-major axis of the ellipse

Eccentricity of the ellipse denoted by e is obtained from the relation

$$e^2 = 1 - \frac{p}{\alpha} \tag{A.4}$$



α - Semi-major axis

e - Eccentricity

E - Eccentric anomaly

Θ - True anomaly

p - Semi-latus rectum

h - Angular momentum

μ - Universal gravity constant times mass of the Moon

P, Q, W - Unit vectors describing orientation of the ellipse in inertial space

FIGURE A1

Elliptical Free Flight Trajectory and Astrodynamic Notation

Another parameter of major importance is the eccentric anomaly E obtained from θ and e by any of the relationships

$$\sin E = \frac{r}{p} (1 - e^2)^{\frac{1}{2}} \sin \theta \tag{A.5}$$

$$\cos E = \frac{r}{p} (\cos \theta + e) \tag{A.6}$$

$$\tan \frac{E}{2} = \sqrt{\frac{1 - e}{1 + e}} \tan \frac{\theta}{2} \tag{A.7}$$

and the quadrant of E is the same as that of θ. From these quantities the radius at any point on the trajectory can be obtained from either

$$r = \frac{p}{1 + e \cos \theta} = \frac{\alpha(1 - e^2)}{1 + e \cos \theta}$$

or

$$r = \alpha(1 - e \cos E) \tag{A.9}$$

Note then that the pericynthion radius is given by

$$r_p = \frac{p}{1 + e} = \alpha(1 - e) \tag{A.10}$$

and apocynthion radius by

$$r_a = \frac{p}{1 - e} = \alpha(1 + e) \tag{A.11}$$

In the $\underline{P}$, $\underline{Q}$, $\underline{W}$ coordinate system shown in Figure A1, the position and velocity vectors at any point 2 can be expressed as

$$\underline{r}_2 = \alpha(\cos E_2 - e)\, \underline{P} + \alpha(1 - e^2)^{\frac{1}{2}} \sin E_2 \underline{Q} \tag{A.12}$$

$$\underline{V}_2 = -\sqrt{\mu\alpha}\, \frac{\sin E_2}{r_2}\, \underline{P} + [\mu\alpha(1 - e^2)]^{\frac{1}{2}}\, \frac{\cos E_2}{r_2}\, \underline{Q} \tag{A.13}$$

These equations were utilized in the derivation of the ellipse predictor equations.

The horizontal velocity of the vehicle at any radius on the orbit is given by

$$V_h = \frac{\sqrt{\mu p}}{r} \qquad (A.14)$$

and thus the radial rate by

$$\dot{r} = \sqrt{V^2 - V_h^2} \qquad (A.15)$$

Radial rate can also be obtained from the expressions

$$\dot{r} = \frac{\sqrt{\mu\alpha}}{r} \; e \sin E \qquad (A.16)$$

or

$$\dot{r} = \sqrt{\frac{\mu}{p}} \, e \sin \Theta \qquad (A.17)$$

The period of the orbit (time to travel 360°) is given by

$$P = 2\pi \sqrt{\frac{\alpha^3}{\mu}} \qquad (A.18)$$

and the mean orbital rate by

$$n = \sqrt{\mu/\alpha^3} \qquad (A.19)$$

The time to travel from point 1 on an orbit to point 2 on an orbit is given by Kepler's equation

$$t = \frac{1}{n} [E_2 - E_1 - e \sin E_2 + e \sin E_1] \qquad (A.20)$$

Mean Anomaly, M, of an orbit is the central angle from pericynthion transversed in time T if the vehicle were traveling at the mean orbital rate n. Thus

$$M = nT \qquad (A.21)$$

In terms of M and e, Θ can be expressed in series form

$$\Theta = M + 2e \sin M + \frac{5}{4} e^2 \sin 2M + \ldots \qquad (A.22)$$

REFERENCES

1. TRW Report No. 05952-6045-R000, Revision B, "LM AGS Lunar Flight Equation Document, Design Mission Computer Program," by T. S. Bettwy, M. J. Laubenstein, E. V. Avery, dated 14 November 1966. (U)

2. TRW Report No. 05952-6113-T000, "LM/AGS Lunar Flight Equations Design Mission Computer Program Equations Document Modification No. 1," by T. S. Bettwy, dated 25 January 1967. (U)

3. TRW Report No. 05952-6040-T000, Revision A, "LM/AGS Scientific Simulation Test Results Document, Design Mission Computer Program," by T. S. Bettwy, dated November 1966. (U)

4. TRW Report No. 05952-6091-T000, "LM AGS Guidance Software, Design Report No. 3 - Earth Prelaunch Gyro Calibration Program," dated 15 December 1966. (U)

5. Fundamentals of Celestial Mechanics, by J. M. A. Danby, the Macmillan Company.

6. Orbital Dynamics of Space Vehicles, by R. Deutsch, Prentice-Hall, Inc.