

## 5.3 POWERED FLIGHT NAVIGATION AND GUIDANCE

### 5.3.1 GENERAL COMMENTS

The objective of the powered flight guidance routines is to maintain an estimate of the CSM state vector during the thrusting maneuvers, and to control the thrust direction such that the desired velocity cut-off conditions are achieved. The powered flight navigation program used to maintain estimate of the vehicle state vector during all thrusting conditions is referred to as the Average-G Routine and is presented in Section 5.3.2.

For the Lunar Landing Mission the basic powered flight guidance concept used in the CMC is a velocity-to-be-gained concept with cross product steering (Section 5.3.3.4) which is used in each of the following two procedures:

1. Lambert Aim Point Maneuver Guidance (Section 5.3.3.2).
2. External  $\Delta V$  Maneuver Guidance (Section 5.3.3.1).

These two procedures, based on the cross product steering concept, differ only in the unique generation of the desired velocity vector,  $\underline{v}_R$  and are used to control all CMC guided powered maneuvers for the Lunar Landing Mission.

5. 3. 2     POWERED FLIGHT NAVIGATION - AVERAGE-G ROUTINE

The purpose of the Powered Flight Navigation Sub-routine is to compute the vehicle state vector during periods of powered flight steering. During such periods the effects of gravity and thrusting are taken into account. In order to achieve a short computation time the integration of the effects of gravity is achieved by simple averaging of the gravity acceleration vector. The effect of thrust acceleration is measured by the IMU Pulsed Integrating Pendulous Accelerometers (PIPA) in the form of velocity increments ( $\Delta \underline{v}$ ) over the computation time interval ( $\Delta t$ ). The computations are, therefore, in terms of discrete increments of velocity rather than instantaneous accelerations. The repetitive computation cycle time  $\Delta t$  is set at 2 seconds to maintain accuracy and to be compatible with the basic powered flight cycle.

The Average-G Routine, in contrast to the Coasting Integration Routine, is used when a short computing time is required such as during powered flight. The Average-G Routine computations are illustrated in Figs. 3. 2-1 and 3. 2-2. The following defines the parameters used in these figures:

- |                    |  |
|--------------------|--|
| $\underline{r}(t)$ | Vehicle position vector at time $t$ .  |
| $\underline{v}(t)$ | Vehicle velocity vector at time $t$ .  |
| $P_C$              | Planet designator $\left\{ \begin{array}{l} 0 \text{ Earth} \\ 1 \text{ Moon} \end{array} \right.$ |
| $\Delta t$         | Computation cycle of 2 seconds.  |

$\Delta \underline{v}(\Delta t)$	The velocity vector change sensed by the IMU PIPA's over the time interval $\Delta t$ . This velocity vector increment is initially sensed in IMU or Stable Member Coordinates and then transformed to the Basic Reference Coordinate System.
$\underline{g}_p(t)$	Previous gravity acceleration vector. This is a required initialization parameter and is supplied by the calling program.
$\underline{u}_r$	Unit vector in the direction of $\underline{r}$ .
$\underline{u}_z$	Unit vector in the direction of the polar axis of the earth.
$\mu_E$	Earth gravitational constant.
$\mu_M$	Moon gravitational constant.
$r_E$	Equatorial radius of the earth.
$J_{2E}$	Second-harmonic coefficient of the earth's potential function.
$\underline{g}_b(t)$	Component of the earth gravity acceleration vector representing earth oblateness effects.

With reference to Fig. 3.2-2 it can be seen that a single oblateness term is included in the earth gravity subroutine computation, but none for the lunar case.

The PIPA measured velocity  $\Delta \underline{v}$  is compensated for instrument errors as described in Section 5.6.13 prior to being transformed into the Basic Reference Coordinate System and processed in the Average-G Subroutine of Fig. 3.2-1.

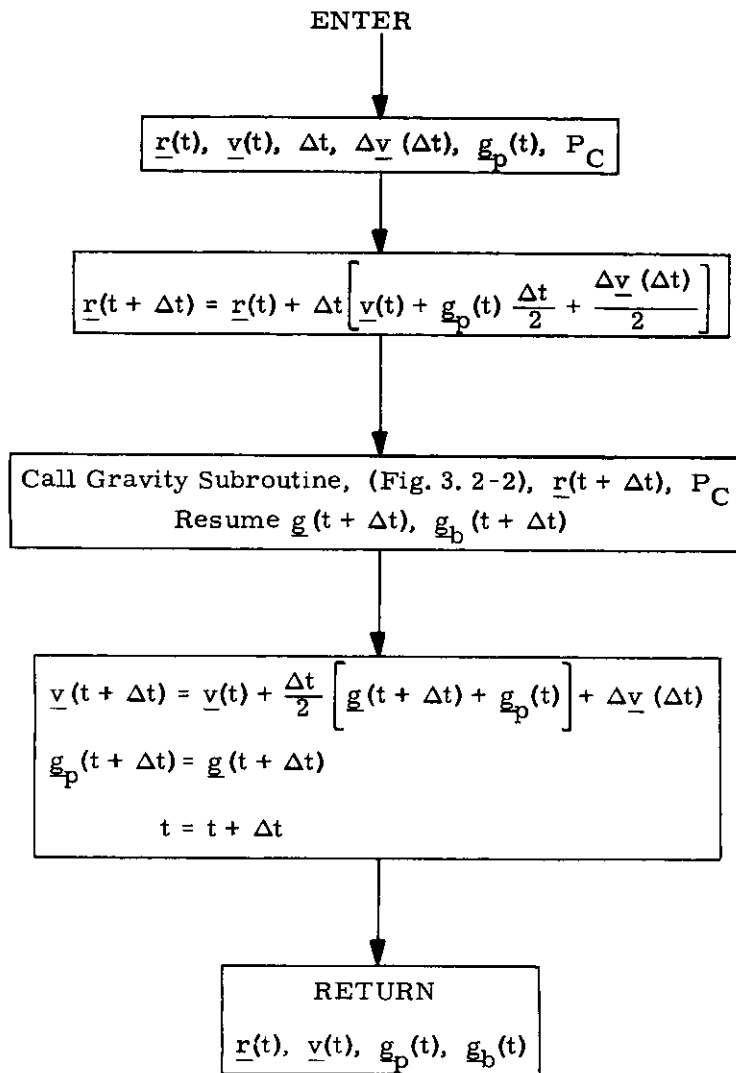


Figure 3. 2-1 Average-G Subroutine

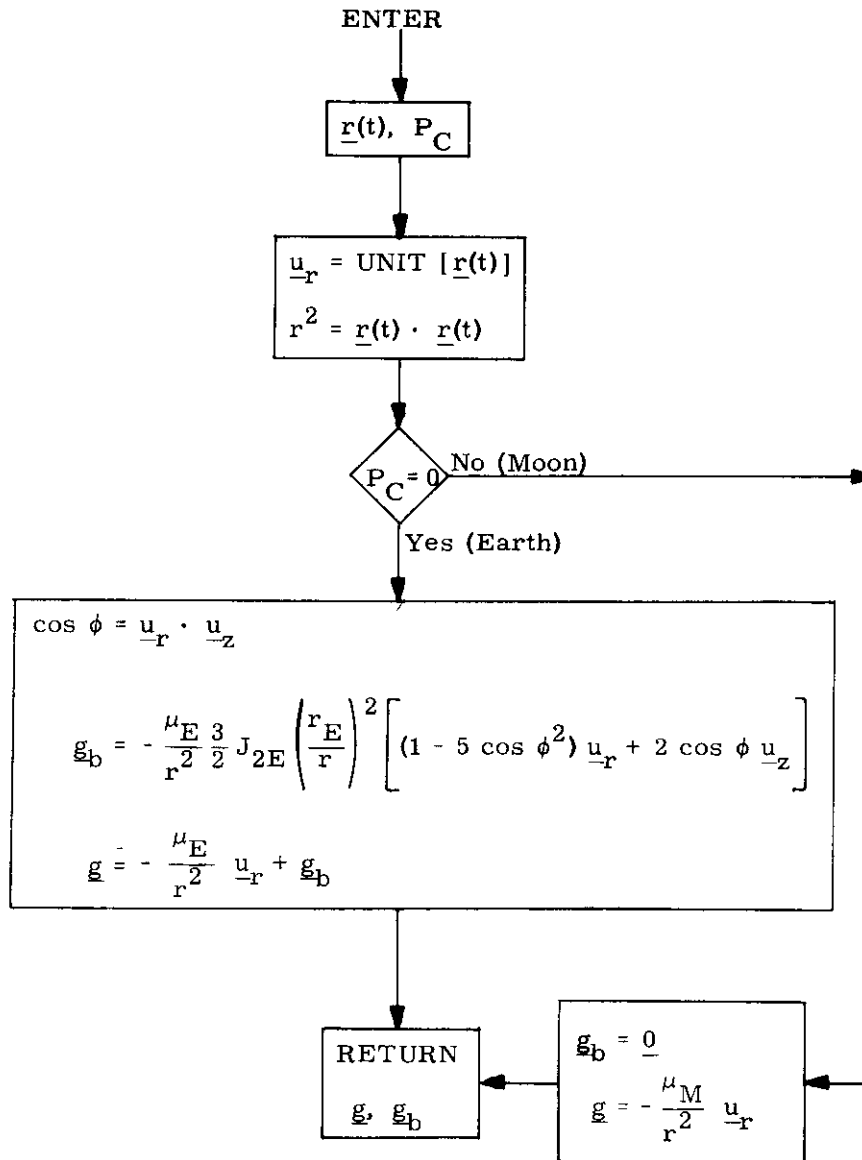


Figure 3. 2-2 Gravity Subroutine

5. 3. 3     POWERED FLIGHT GUIDANCE USING CROSS  
PRODUCT STEERING

5. 3. 3. 1   Introduction

The cross product steering concept is used to control the following maneuvers:

- (a)   Cislunar Midcourse Corrections, P-31 or P-30
- (b)   Lunar Orbit Insertion (LOI) P-31 and P-30
- (c)   Lunar Orbit Plane Change Maneuvers (LOPC), P-31
- (d)   Rendezvous Intercept (P-34) and Midcourse Correction Maneuvers (P-35), Stable Orbit Rendezvous Maneuvers (P-38 and P-39)
- (e)   Return-to-Earth Maneuvers, P-37
- (f)   External  $\Delta V$  and Orbital Phasing Maneuvers (RTCC or LGC Targeted Maneuver) P-30
- (g)   Transearch Injection (TEI) P-31

The External  $\Delta V$  Guidance mode (Section 5. 3. 3. 3. 1) is used for maneuvers in which the required cut-off velocity,  $\underline{v}_R$ , is specified by a source external to the CMC by use of program P-30. All other maneuvers are controlled by the Lambert Aim Point Maneuver Guidance Mode (Section 5. 3. 3. 3. 2) in which the  $\underline{v}_R$  is periodically computed by the Lambert subroutine during the maneuver to establish the desired intercept trajectory. Both External  $\Delta V$  and Lambert Aim Point Guidance modes use the cross product steering concept to control the thrust direction along the velocity-to-be-gained vector, and terminate thrust when the desired velocity increment has been achieved.

Three subroutines are used repetitively in sequence (Section 5.3.3.2) during cross product controlled maneuvers to accomplish this function. These are:

1. The Powered Flight Navigation Average-G Routine which computes the state vector accounting for the effects of thrust acceleration and gravity.
2. The Cross-Product Steering Subroutine which has 3 functions:
  - a) incremental updating of the velocity-to-be-gained vector.
  - b) generation of steering commands to the vehicle autopilot.
  - c) computation of time-to-go before engine shut-off and the issuance of engine-off commands.
3. The Velocity-to-be-Gained Subroutine which repetitively solves the Lambert intercept problem when in the Lambert Aim Point guidance mode.

The Average-G Routine is described in Section 5.3.2. The other subroutines listed above are described in Sections 5.3.3.4 to 5.3.3.6. The Pre-Thrust Subroutines of Section 5.3.3.3 initialize the powered maneuver programs for either the External  $\Delta V$  or Lambert Aim Point guidance modes, and for the selected engine for the maneuver.

The CMC powered flight programs described by the computation subroutines presented in Section 5.3.3 are

P-40 Service Propulsion System (SPS) Thrust Prog.

P-41 Reaction Control System (RCS) Thrust Prog.

Active steering and engine-off commands are provided by program P-40. Maneuvers using RCS translation control (P-41) are manually controlled and terminated by the astronaut while the CMC displays the required velocity-to-be-gained in control coordinates (Section 5.3.3.3.3).

The functions of the External  $\Delta V$  Pre-Thrust Program, P-30 are described in the pre-thrust subroutine description of Section 5.3.3.3.1.

The Lambert Aim Point Maneuver Guidance Program, P-31 is a RTCC targeted pre-thrust program in which all required target parameters are uplinked via the CMC Update Program P-27. Program P-31 uses the same routines as CMC targeted intercept maneuvers as shown in Section 5.3.3.3.2.



### 5.3.3.2 Powered Flight Guidance Computation Sequencing

The time sequencing of SPS powered flight subroutines for External  $\Delta V$  steering is illustrated in Figs. 3.3-1 thru 3.3-3, and for Lambert Aim Point steering in Figs. 3.3-4 thru 3.3-6. These figures represent the sequence of operations for P-40 SPS maneuvers lasting longer than 6 seconds. RCS controlled maneuvers (P-41) require manual steering control, and the general timing sequence is different from that described for SPS maneuvers. The general cross product steering concept is used to compute and display the required velocity-to-be-gained vector, but no engine-off computations are made for RCS maneuvers. The following description of sequence operations is restricted to SPS Maneuvers.

Figures 3.3-1 and 3.3-4 show the sequencing during the ignition count down which starts 30 seconds before the nominal ignition time. Figures 3.3-2 and 3.3-5 show the normal sequencing for an engine-on time greater than 6 seconds as predicted by the pre-thrust subroutines. Figures 3.3-3 and 3.3-6 illustrate the sequencing during engine thrust termination.

The basic computation cycle time of the steering is 2 seconds and, as shown on the above figures, is initiated by the reading of the PIPA  $\Delta v$  registers. The various subroutines utilized during the 2 second cycle are sequenced in time as shown.

During Lambert Aim Point steering, however, the velocity-to-be-gained ( $v_G$ ) updating is anticipated to occur nominally every four seconds. However during SPS maneuvers the CMC computation occasionally prevents completion of the Lambert solution in 2 to 4 seconds. The program is capable of processing an N cycle update where N can be greater than 2. The time sequencing diagrams show only the nominal two and four second  $v_G$  update cases. This allows sequenced parallel computing functions, such as the demands of the autopilots and telemetry, to take place without interference with the basic two second cycle time. Should the  $v_G$  computation be completed in time for use during a given 2 second cycle, it will be utilized.

If the calculations conducted prior to the thrusting period (pre-thrust) indicate that the maneuver objective can be reached with a 6 second thrust period or less, provision is included (switches  $S_I$  and  $S_W$ ) to preclude the generating of steering commands during the thrusting period with the engine-off signal being generated prior to ignition.

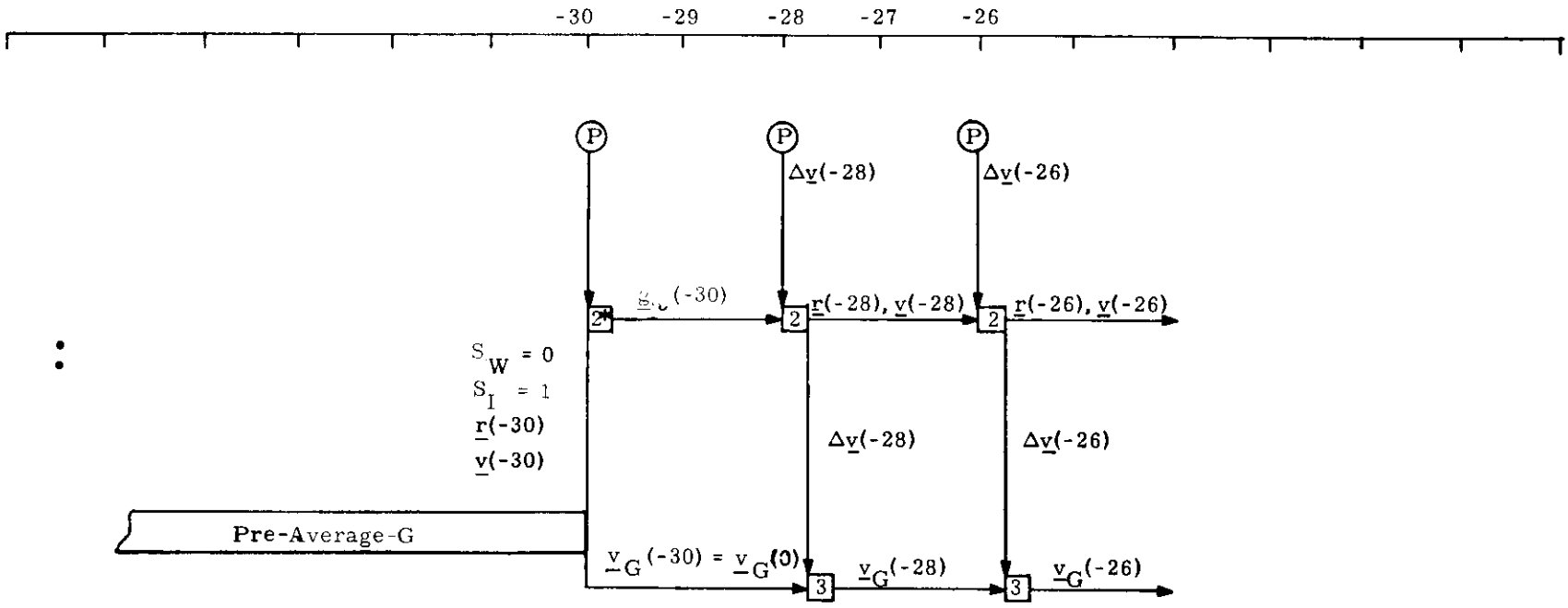
In addition to timing information, the sequence diagrams of Figs. 3.3-1 to 3.3-6 also show the basic information utilized by each subroutine and its source.

The guidance computer program which controls the various subroutines to create a powered flight sequence is called the Servicer Routine. The sequence diagrams of Figs. 3.3-1 to 3.3-6 define what the Servicer Routine does, but do not show the logic details of how these functions are accomplished.

The subroutines listed in Figs. 3.3-1 to 3.3-6 are described in Section 5.3.2 and the following Sections 5.3.3.3 to 5.3.3.6. These sections should be referenced in tracing the powered flight computation sequencing.

Ullage will be turned off at the fixed time of 2 seconds from engine-on command. Steering will be enabled ( $S_W$  set to 1) at the fixed time of 2 seconds from engine-on command.

Seconds from Nominal Ignition Time  $t_{IG}$



\*Initial Average-G cycle computes gravity terms only.

NOTE: SPS ullage will be turned on by the Astronaut at a time depending on the vehicle weight and propellant loading. It will be turned off at the fixed time of 2 secs from the engine on command.

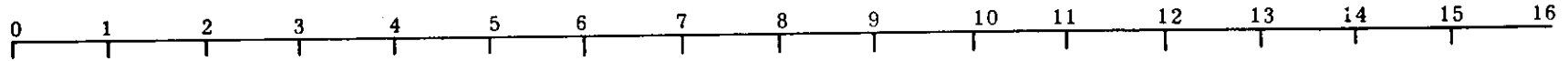
(P)	Read and Zero PIPA Registers
2	Average G Subroutine
3	Cross Product Steering Subroutine

Rev. 3 - 11/68

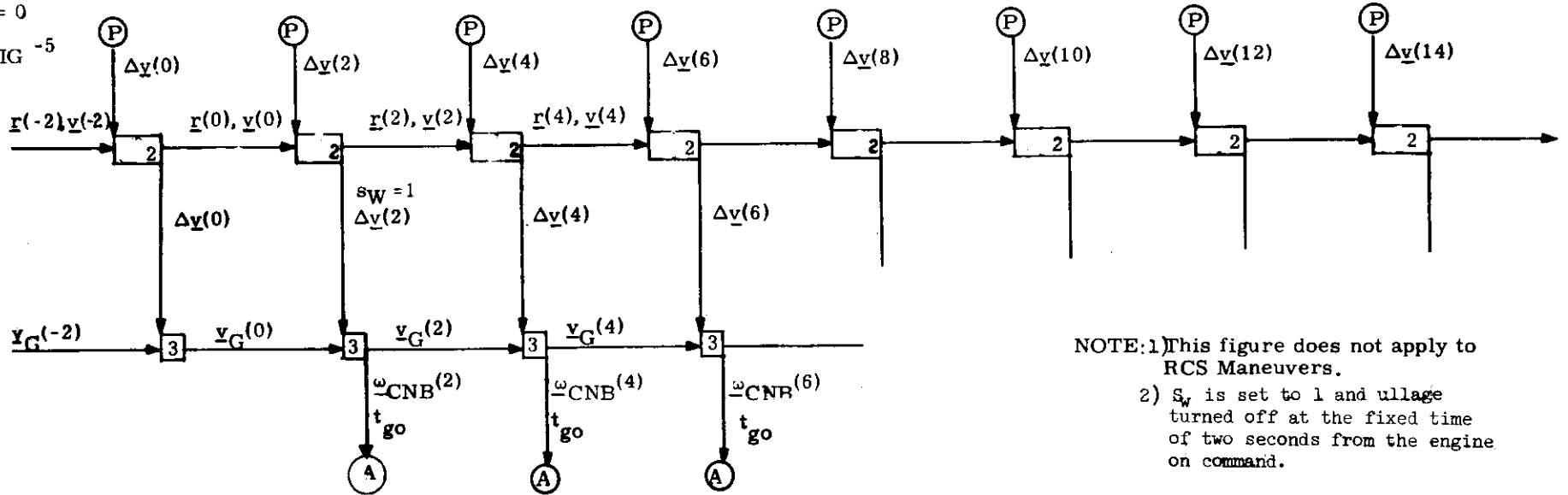
5.3-11

Fig. 3.3-1 Ignition Countdown - External  $\Delta V$  Subroutine Sequencing for SPS Maneuvers

Seconds from Nominal Ignition time -  $t_{IG}$



$s_I = 0$   
at  $t_{IG} - 5$



NOTE: 1) This figure does not apply to RCS Maneuvers.  
2)  $s_w$  is set to 1 and ullage turned off at the fixed time of two seconds from the engine on command.

Rev. 3 - 11/68  
5.3-12

Nominal Ignition Time

Actual Ignition Time

Time at which:  
Go to SPS Thrust Fail Routine (R40) if engine fails to ignite.

- (P) Reading and zeroing of PIPA Registers
- (2) Average G Subroutine
- (3) Cross Product Steering subroutine
- (A) Transmit commands to Autopilot

Fig. 3.3-2 Normal External  $\Delta V$  Subroutine Sequencing (Maneuver time greater than 6 sec) for SPS Maneuvers

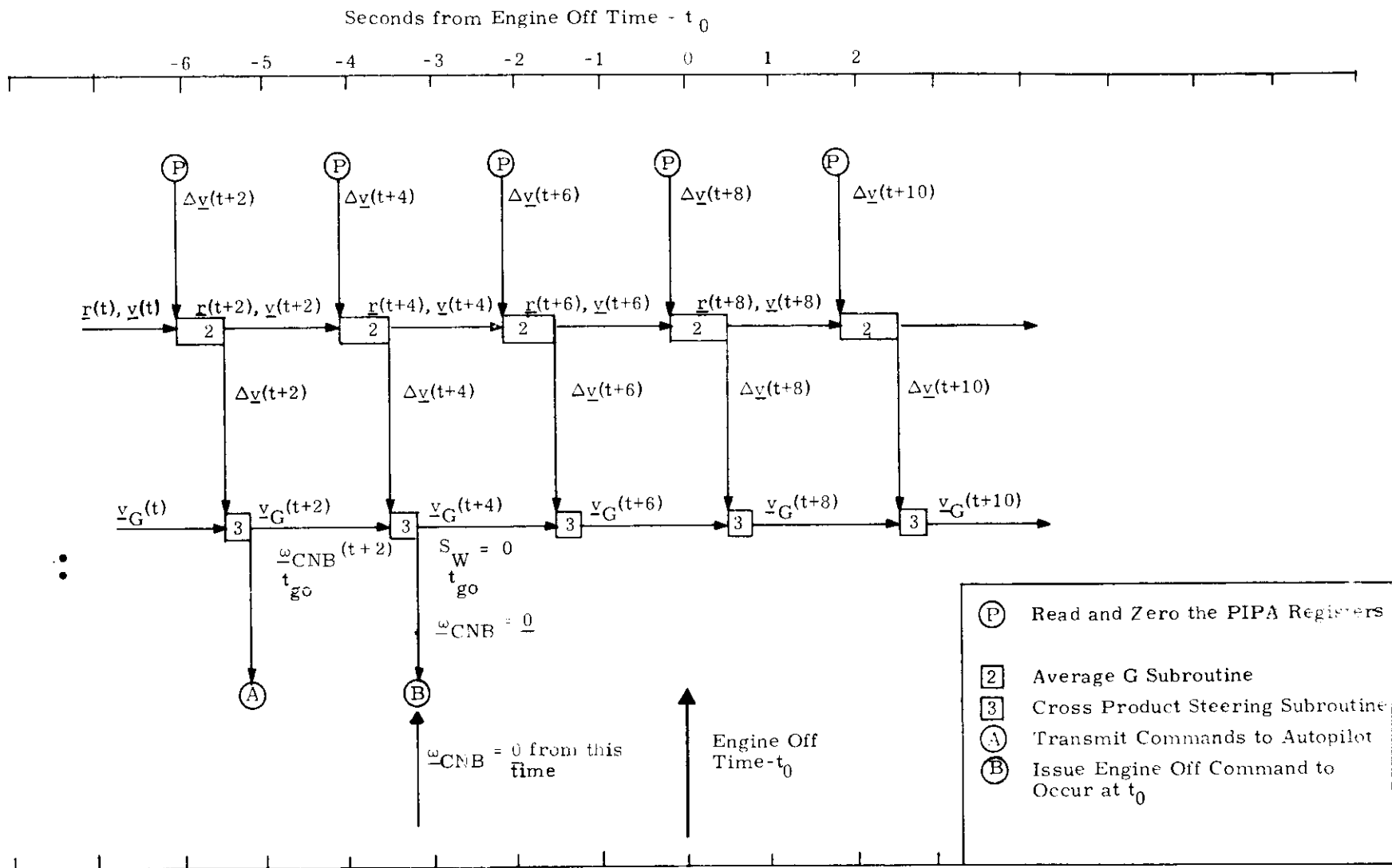


Fig. 3.3-3 Engine Off External  $\Delta V$  Subroutine Sequencing for SPS Maneuvers

Seconds from Nominal Ignition Time -  $t_{IG}$

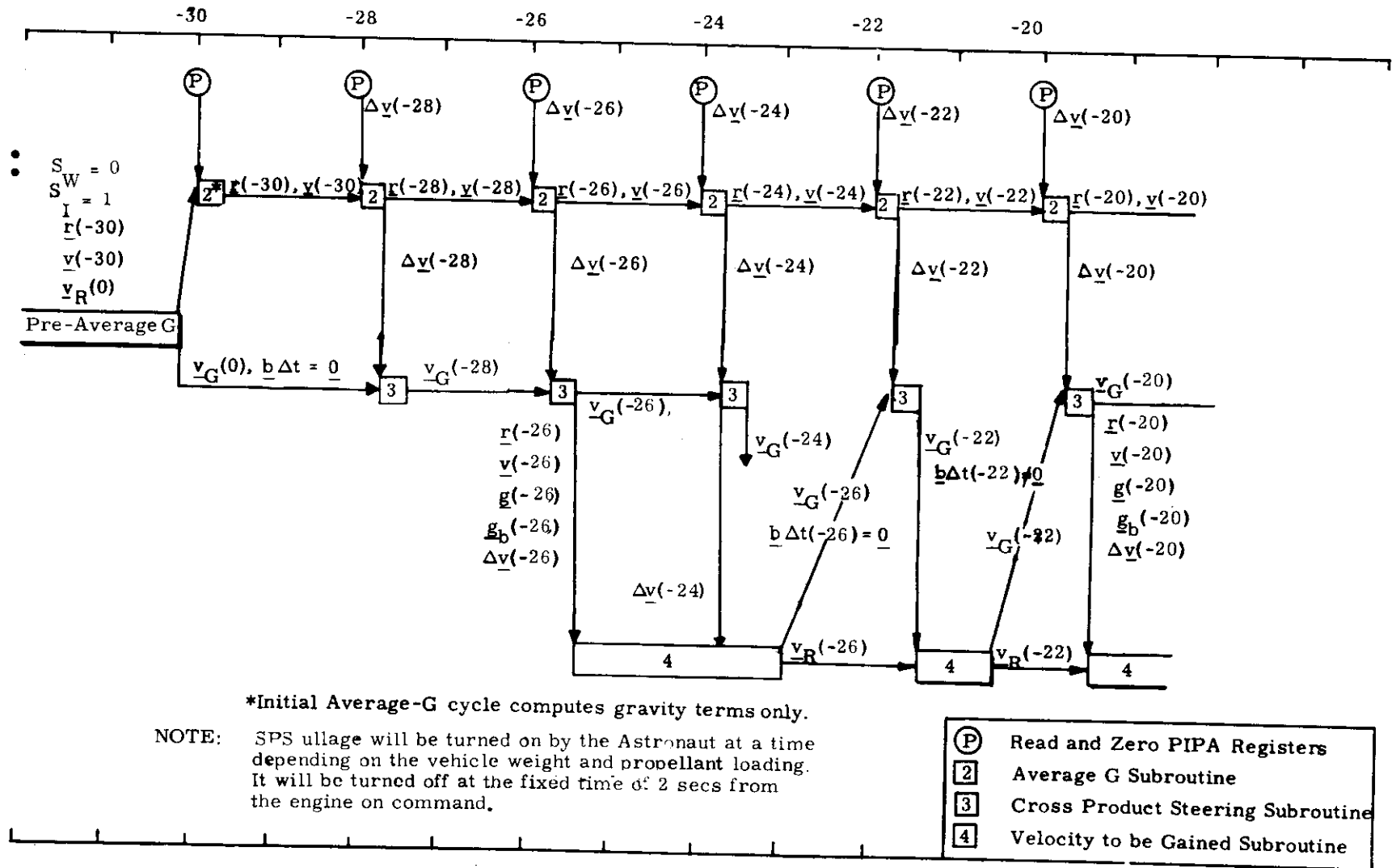


Fig. 3.3-4 Ignition Countdown - Lambert Aim Point Subroutine Sequencing for SPS Maneuvers

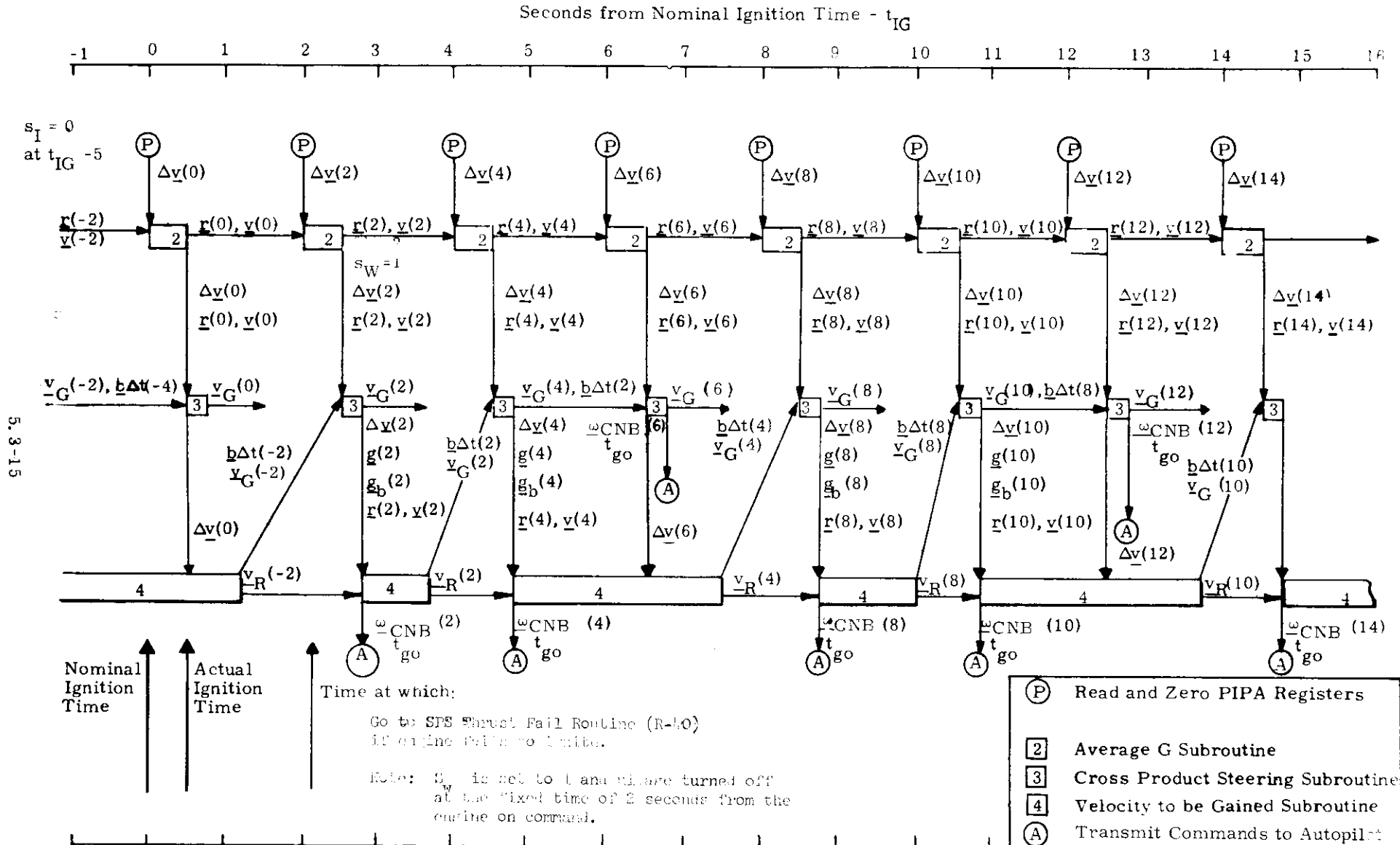


Fig. 3.3-5 Normal Lambert Aim Point Subroutine Sequencing (Maneuver time greater than 6 sec) for SPS Maneuvers

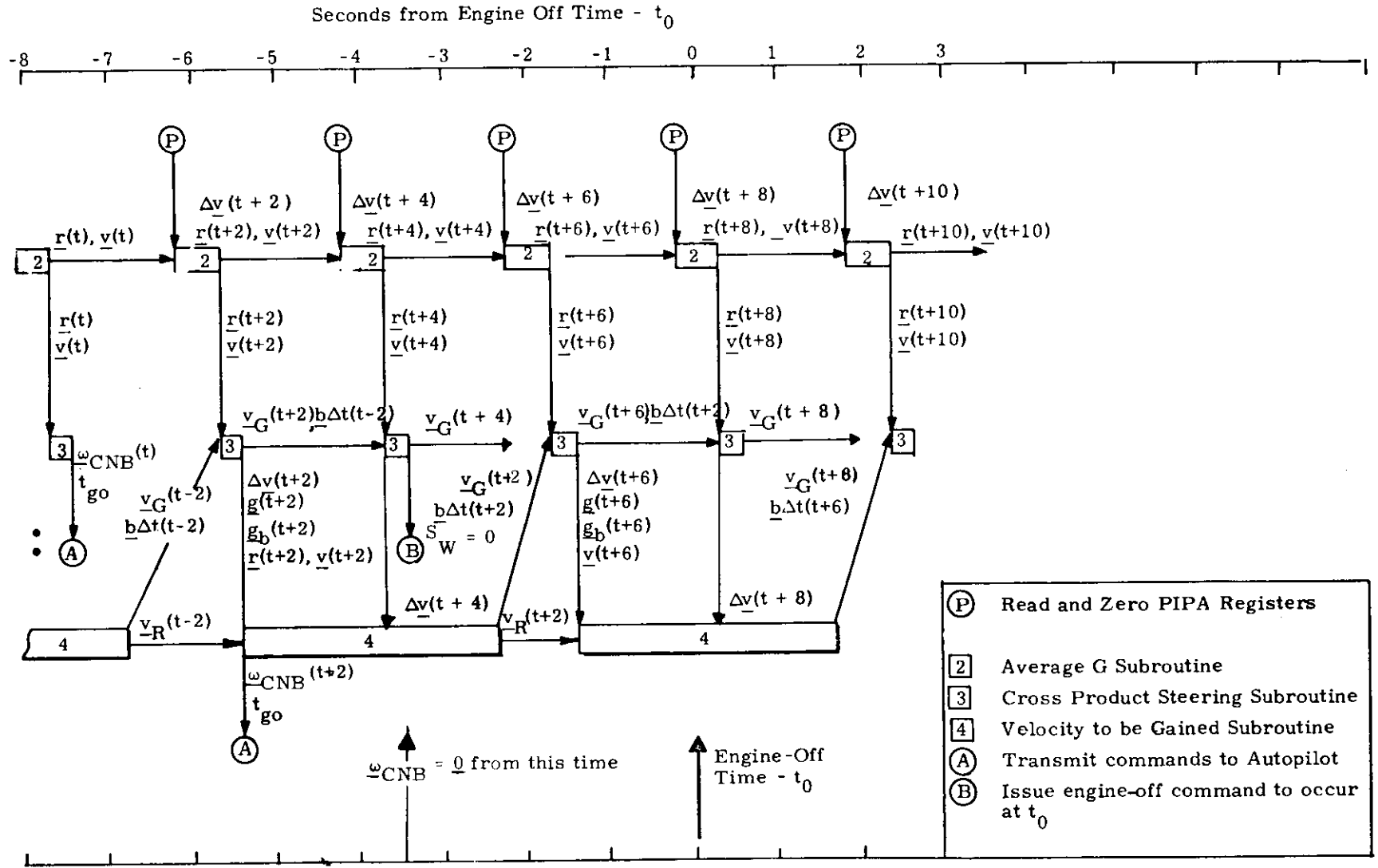


Fig. 3.3-6 Engine-off Lambert Aim Point Subroutine Sequencing for SPS Maneuvers



### 5.3.3.3 Pre-Thrust Computations

The objective of the computations required prior to thrusting maneuvers is to determine the following:

- (1) The desired thrust direction at ignition.
- (2) At time  $t_{IG} - 5$  the duration of the powered maneuver is estimated to determine if there will be enough time to allow active steering for the SPS.
- (3) Whether an IMU realignment is required to avoid gimbal lock.
- (4) Various parameters and variables required by subsequent powered flight routines.

The two major guidance modes using cross product steering are the External  $\Delta V$  guidance mode and the Lambert Aim Point Maneuver guidance mode. The pre-thrust computations required for the External  $\Delta V$  mode are presented in Section 5.3.3.3.1. Those required for the Lambert Aim Point Maneuver mode are described in Section 5.3.3.3.2. The initial IMU alignment computations and maneuver time logic is summarized in Section 5.3.3.3.3.

The following description of pre-thrust computations applies to both SPS (P-40) and RCS (P-41) maneuvers.

#### 5.3.3.3.1 External $\Delta V$ Maneuver Pre-Thrust Computations

External  $\Delta V$  maneuver guidance is normally used to control orbital phasing maneuvers or an externally targeted maneuver in which a constant thrust attitude is desired. The guidance program accepts input data via the DSKY (P-30) or

the telemetry uplink (P-27) in the form of 3 components of an impulsive  $\Delta V_{LV}$  expressed in a local vertical coordinate system of the active vehicle at the ignition time  $t_{IG}$ . An approximate compensation for the finite maneuver time is made within the program by rotating the  $\Delta V_{LV}$  vector, and the guidance program issues commands to the spacecraft control system so as to apply the compensated velocity increment along an inertially fixed direction. The active vehicle state vector is normally either available or can be extrapolated to the ignition time in the CMC.

The pre-thrust computations required for the External  $\Delta V$  guidance mode are shown in Fig. 3.3-7. The following parameter definitions refer to this figure.

$\Delta V_{LV}$  Specified velocity change in the local vertical coordinate system of the active vehicle at the time of ignition. This is an input parameter.

$$\Delta V_{LV} = \begin{pmatrix} \Delta V_X \\ \Delta V_Y \\ \Delta V_Z \end{pmatrix}$$

$\Delta V$  Specified velocity change in Basic Reference Coordinates

$t_{IG}$  Ignition time, an input parameter.

$\Delta V_P$  The inplane velocity components of  $\Delta V_{LV}$  in the Basic Reference Coordinate System.

$\theta_T$  The approximate central angle traveled during the maneuver.

MGA Angle equivalent to the IMU middle gimbal angle when the vehicle X axis is aligned along  $\Delta V$ . This angle is used to check for gimbal lock tolerance.

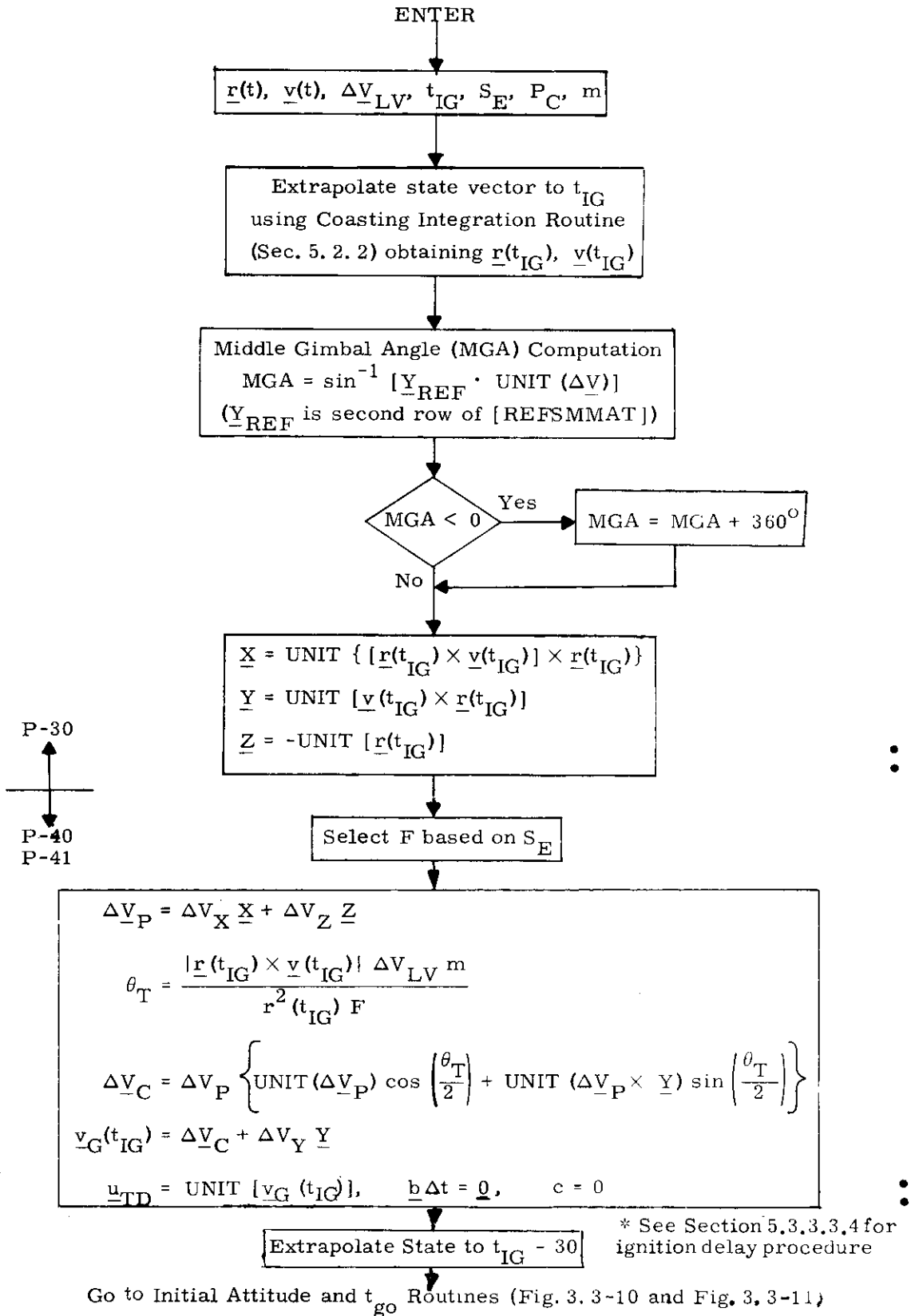


Figure 3.3-7 External  $\Delta V$  Prethrust Routine

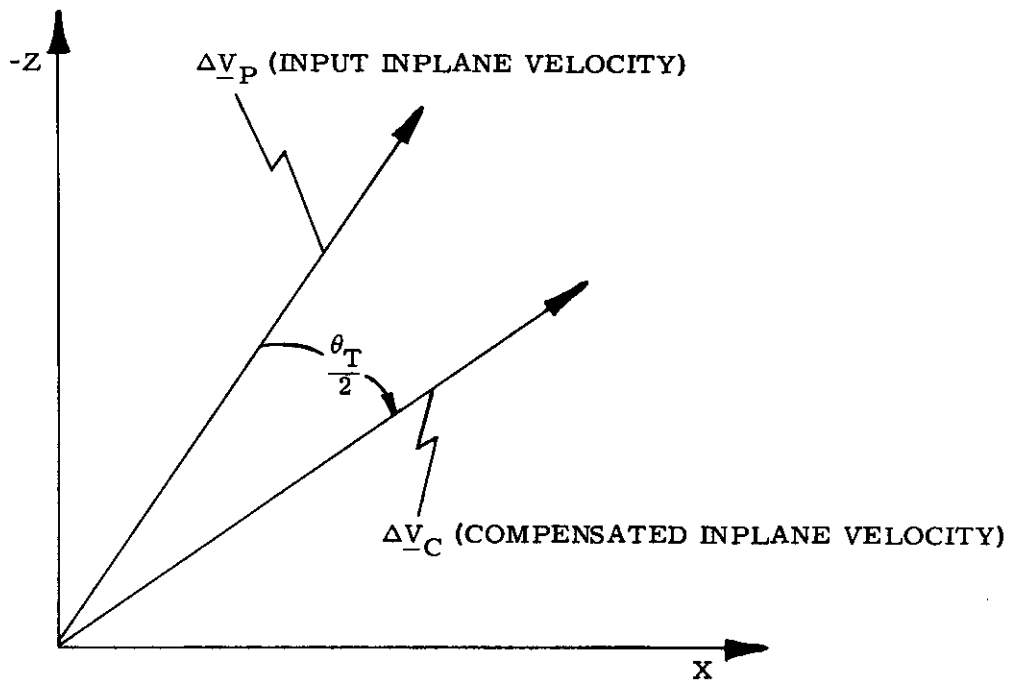


Figure 3. 3-8 Inplane External  $\Delta V$  Maneuver Angle Compensation

m	Vehicle mass.
$S_E$	Engine Select Switch $\begin{cases} 1 & \text{P-40 (SPS)} \\ 0 & \text{P-41 (RCS)} \end{cases}$
F	Prestored nominal thrust based on the engine selection switch $S_E$ $F_{SPS}$ for the SPS (P-40) $F_{RCS}$ for the RCS (P-41) The values of the various engine thrust levels are listed in Section 5.9. In the case of P-41, RCS, the astronaut can select either a 2 or 4 jet translation maneuver.
$\Delta \underline{V}_C$	The compensated inplane velocity-to-be-gained vector.
$\underline{v}_G(t_{IG})$	Total velocity-to-be-gained at $t_{IG}$ .
$\underline{u}_{TD}$	Unit vector in the desired initial thrust direction.
c	Cross Product Steering Constant This steering constant (Section 5.3.3.4) is set equal to zero for External $\Delta V$ maneuvers.

The inplane External  $\Delta V$  maneuver angle compensation involving  $\Delta \underline{V}_P$  and  $\Delta \underline{V}_C$  is illustrated in Fig. 3.3-8.

#### 5.3.3.3.2 Lambert Aim Point Maneuver Pre-Thrust Computations

The objective of the Lambert Aim Point Maneuver Guidance program is to control the cut-off velocity vector such that the resulting trajectory intercepts a specified target position vector at a given time. The target position vector for the TEI, LOI and LOPC maneuvers are determined by the RTCC and no CMC targeting routines exist for these maneuvers. Return to earth aborts initiated within the lunar sphere of influence must also be RTCC targeted.

The table below summarizes the maneuvers performed by Lambert Aim Point Maneuver Guidance.

Maneuvers Controlled by Lambert Aim Point Maneuver Guidance

<u>Maneuver</u>	<u>Program</u>	<u>*Targeting Parameters From</u>
TEI	P-31	RTCC
LOI (First Maneuver)	P-31	RTCC
LOPC	P-31	RTCC
Rendezvous Intercept	P-34	RTCC, CMC or LGC
Rendezvous Midcourse Maneuver	P-35	RTCC, CMC or LGC
Stable Orbit Rendezvous Maneuvers	P-38, P-39	RTCC, CMC or LGC
Some Cislunar Midcourse Corrections	P-31	RTCC (Within Lunar Sphere) RTCC or CMC (Outside Lunar Sphere)
Return-to-Earth Maneuvers	P-37	RTCC (Within Lunar Sphere) RTCC or CMC (Outside Lunar Sphere)

\*Targeting Parameters

- 1) Ignition time  $t_{IG}$
- 2) Time of flight to conic target aim vector ( $t_F$ )
- 3) Conic target aim vector  $\bar{r}(t_2)$  where  $t_2 = t_F + t_{IG}$
- 4)  $c$  cross product steering constant

The Lambert Aim Point Maneuver guidance mode basically uses the Lambert Subroutine of Section 5. 5. 6 called through the Initial Velocity Subroutine of Section 5. 5. 11 during the pre-thrust phase to determine the required initial thrust direction,  $\underline{u}_{TD}$ . The pre-thrust computations required for the Lambert Aim Point guidance mode are shown in Fig. 3. 3-9. The primary operation shown in this figure is the computation of the required intercept velocity vector  $\underline{v}_R$  at the ignition time,  $t_{IG}$ . The following parameter definitions refer to Fig. 3. 3-9.

$\underline{r}(t_2)$	Offset target intercept position vector at time $t_2$ . This parameter is determined by the preceding targeting program and is an input to the Lambert Aim Point prethrust subroutine.
$t_2$	Intercept time of arrival associated with the offset target vector, $\underline{r}(t_2)$ . This is an input parameter.
$S_R$	Target rotation switch set in P-34, P-35, P-38 or P-39 indicating that the target vector was rotated into the orbital plane due to proximity to $180^\circ$ transfer conditions.
	$S_R \begin{cases} 1 & \text{Target vector rotated} \\ 0 & \text{No rotation} \end{cases}$
$N_1$	Number of target offset iterations desired in the Initial Velocity Subroutine of Section 5. 5. 11.
$\epsilon$	Initial Velocity Subroutine parameter.
$\underline{v}_R(t_{IG})$	Required velocity vector at the ignition time $t_{IG}$ to establish the intercept trajectory.
$\underline{g}$	Total gravity vector associated with $\underline{r}(t_{IG})$ .
$\underline{b} \Delta t$	Incremental change of the velocity-to-be-gained vector over one sample period.
MGA	The angle equivalent to the IMU middle gimbal angle when the vehicle X axis is aligned along $\underline{v}_G$ .

c Cross Product Steering Constant

The cross product steering constant,  $c$ , is set equal to one if the pre-thrust program was P-34 P-35, P-38 or P-39, and  $1/2$  if the pre-thrust program was P-37, (Section 5.4). If the pre-thrust program was the External  $\Delta V$  Program, P-30,  $c$  is set equal to zero (Fig. 3.3-7). For RCS maneuvers (P-41) the constant  $c$  is set to zero before the Lambert Aim Point Maneuver Pre-Thrust Computation as shown in Fig. 3.3-9. In pre-thrust program P-31 (Lambert Aim Point Maneuver Guidance) the value of  $c$  is determined by RTCC and uplinked with other target parameters (Fig. 3.3-9).

ec Pad loaded value of the cross product steering constant. RTCC uplink can modify the value of  $ec$  through the range  $-4$  to  $+4$  for Lambert targeting (Fig. 3.3-7).

The rotation switch  $S_R$  is set in the Lambert Aim Point targeting programs P-34, P-35, P-38 and P-39 to indicate whether the target aim point vector was within a specified cone angle,  $\epsilon$ , measured from the  $180^\circ$  transfer angle condition. If the initial target vector was within this cone angle, it is rotated into the active vehicle orbital plane in the Initial Velocity Subroutine of Section 5.5.11 so that excessive plane change and  $\Delta V$  requirements are avoided about the  $180^\circ$  transfer angle condition. In the intercept targeting programs P-34 and P-35 the cone angle  $\epsilon$  is set at  $15^\circ$ , and active vehicle transfer angles between  $165^\circ$  to  $195^\circ$  are normally avoided in the targeting procedure. If a transfer angle condition falling within this  $180^\circ \pm 15^\circ$  sector ( $S_R = 1$ ) is intentionally selected either during the TPI targeting (P-34), or results from a rendezvous midcourse correction maneuver (P-35) during an intercept trajectory targeted for more than  $180^\circ$ , the Lambert Aim Point Maneuver Pre-Thrust Routine of Fig. 3.3-9 increases the Initial Velocity Subroutine cone angle  $\epsilon$  to  $45^\circ$  so that the active vehicle transfer angle will not change from inside to outside the cone angle during the powered maneuver. Such a condition is undesirable since the intercept trajectory would be retargeted during the powered maneuver. Likewise, if the initial transfer central angle falls outside the  $15^\circ$  cone angle  $\epsilon$  of programs P-34 and P-35 ( $S_R = 0$ ),  $\epsilon$  is decreased to  $10^\circ$  in Fig. 3.3-9 to reduce the possibility of the transfer angle changing from outside to inside the cone angle during a powered maneuver. It should be noted that all other targeting programs (P-31, and P-37) set  $S_R$  equal to zero.



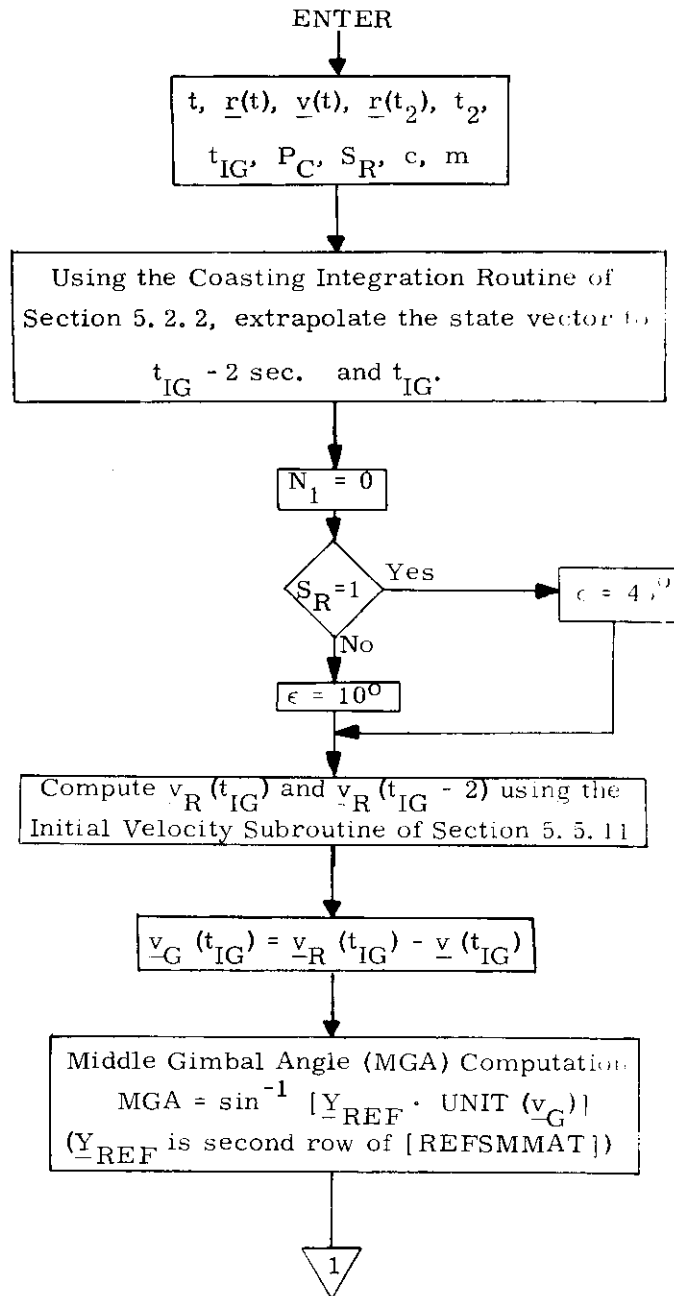
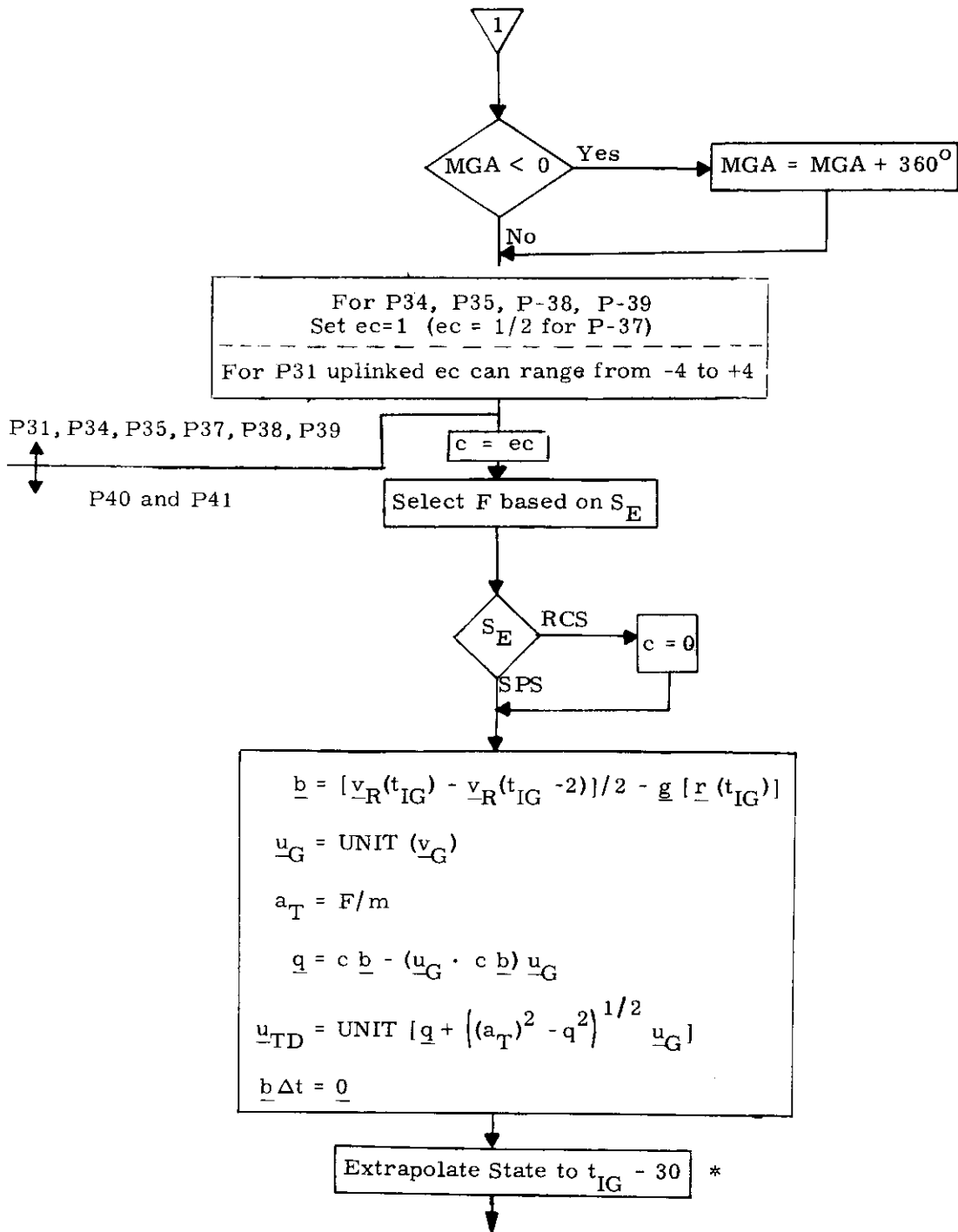


Figure 3.3-9 Lambert Aim Point Maneuver Pre-Thrust Routine  
(page 1 of 2)



Go to Initial Attitude and  $t_{go}$  Routines (Fig. 3.3-10 and Fig. 3.3-11)

\* See Section 5.3.3.4 for ignition delay procedure.

Figure 3.3-9 Lambert Aim Point Maneuver Pre-Thrust Routine  
(page 2 of 2)

5.3-25A

THIS PAGE INTENTIONALLY LEFT BLANK

5.3-25B

Rev. 3 - 11/68

5.3.3.3.3 Initial IMU Alignment and Maneuver Time-to-Go Computations

Initial Attitude Routine

The following pre-thrust computations and functions are required for both the External  $\Delta V$  and Lambert Aim Point guidance modes:

1. Determination of the preferred IMU alignment for the thrusting maneuver (Section 5.1.4.2.1).
2. Alignment of the vehicle thrust axis along the desired initial thrust direction,  $\underline{u}_{TD}$ .
3. Estimate the maneuver time,  $t_{go}$ , prior to engine ignition.

In Fig. 3.3-10 the following parameter definitions apply:

[REFSMMAT]	Transformation matrix from the BRC System to the IMU or Stable Member Coordinate System (Section 5.6.3.3)
[SMNB]	Transformation matrix from the Stable Member Coordinate System to the Navigation Base Coordinate System (Section 5.6.3.2)
$\underline{u}_A$	Unit vector of the assumed thrust acceleration in Navigation Base Coordinates. For the SPS, the unit vector $\underline{u}_A$ is along the engine axis (called engine bell).
$\underline{u}_D$	Unit vector of the desired thrust direction in Stable Member Coordinates.

$p_T = p + p_0$  Total pitch trim offset angle about the spacecraft Y axis.

$y_T = y + y_0$  Total yaw trim offset angle about the spacecraft Z axis.

where:

$p$  and  $y$  Electrical pitch and yaw trim angles respectively.

$p_0$  Mechanical pitch trim angle equal to  $-2.15^\circ$  about the spacecraft Y axis.

$y_0$  Mechanical yaw trim angle equal to  $+0.95^\circ$  about the spacecraft Z axis.

For an SPS burn, because of pitch and yaw trim,  $\underline{u}_A$ , the unit vector of the assumed thrust acceleration vector in Navigation Base Coordinates is not a unit vector along the spacecraft X axis (1, 0, 0) but is a function of the pitch and yaw trim angles given above. In Fig. 3.3-10 it is called the Engine Axis unit vector,  $\underline{u}_A$ . The maneuver then places the engine axis unit vector ( $\underline{u}_A$ ) along the desired unit thrust direction  $\underline{u}_D$ .

#### Time-to-Go Prediction Routine

Very short burns require special consideration since some interval of time elapses before effective steering is achieved. Not only must the autopilot react to the pointing commands, but the engine-off signal may be required before the  $\Delta V$  from the PIPA's can be measured. To this end an estimate of the burn time,  $t_{go}$ , is made on the basis of SPS engine data, prior to engine ignition as shown in Fig. 3.3-11.

5.3-27

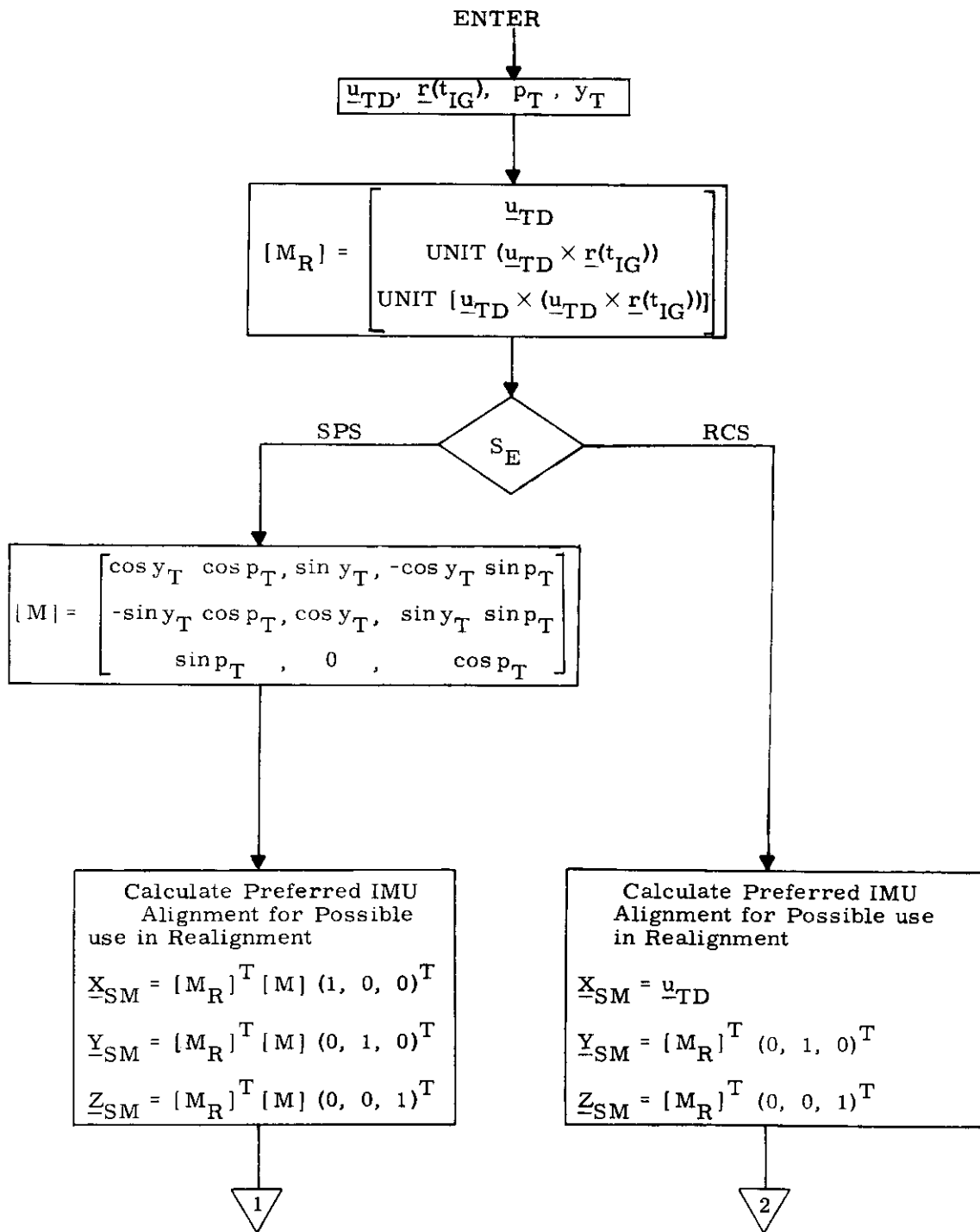


Figure 3.3-10 Initial Attitude Routine  
(page 1 of 2)

5.3-28

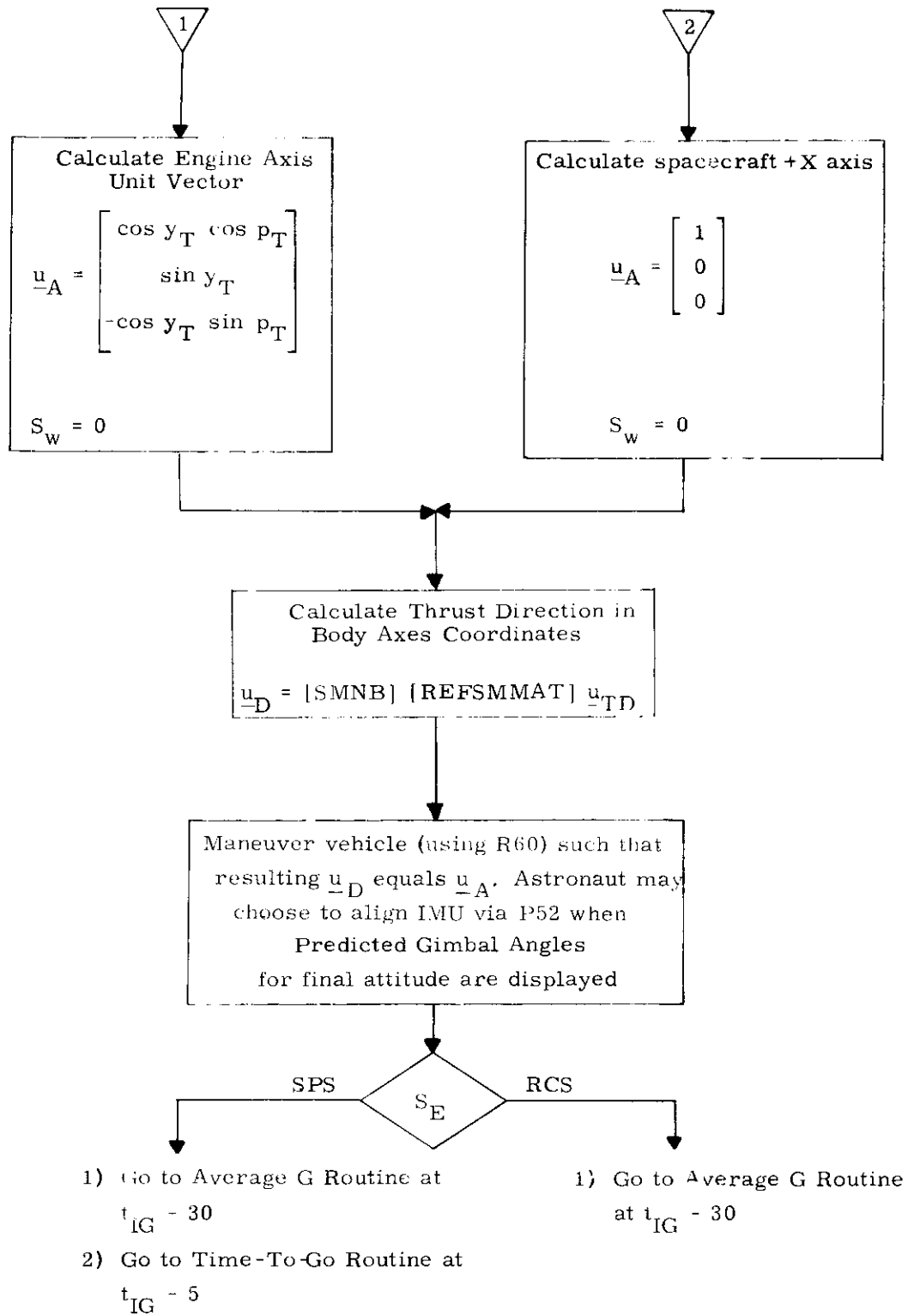


Figure 3.3-10 Initial Attitude Routine  
(page 2 of 2)

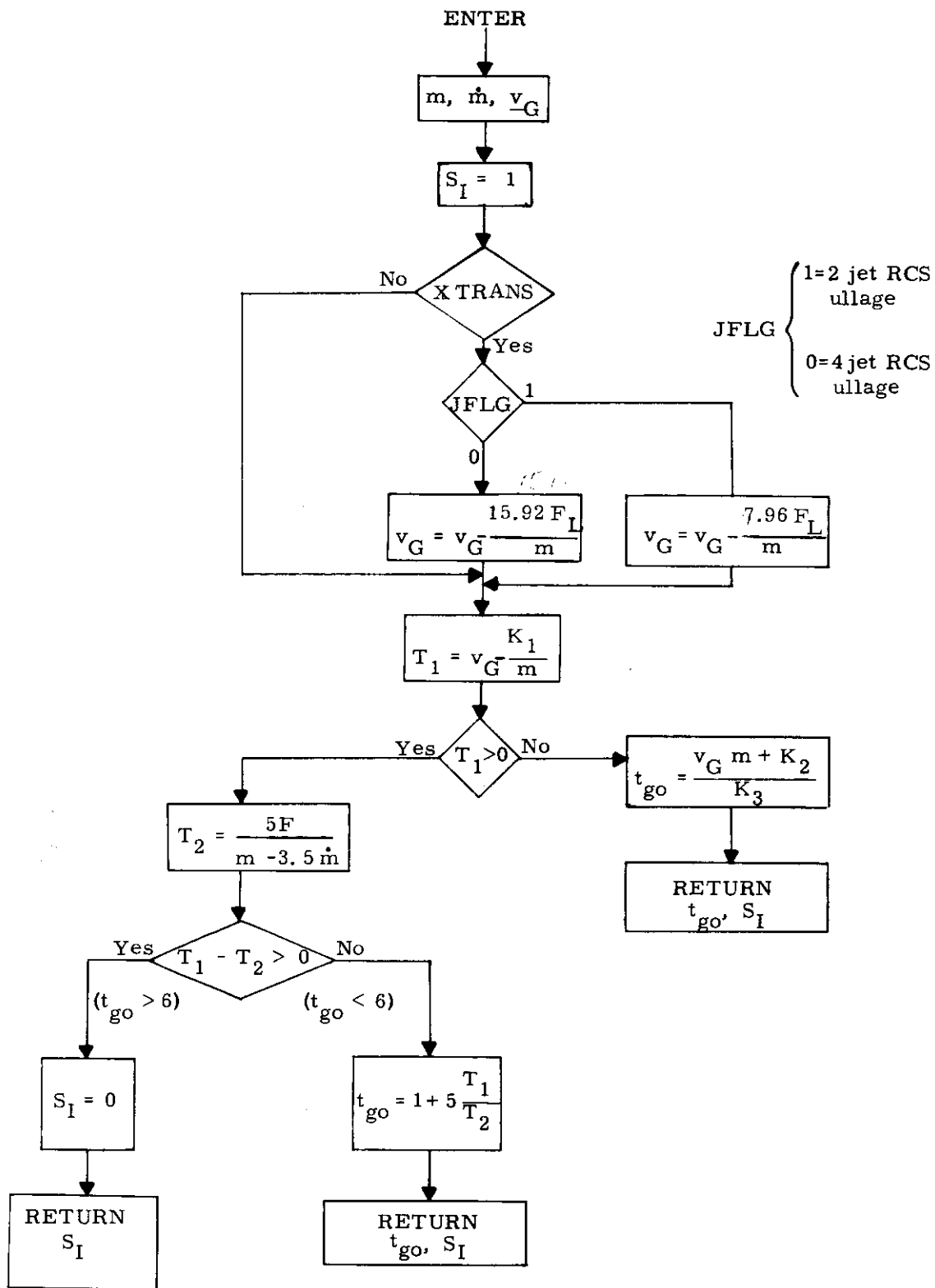


Figure 3.3-11 Time-to-Go Prediction Routine

5.3-30



With reference to Fig. 3.3-11:

$S_I$	Impulse switch
$S_I$	$\left\{ \begin{array}{l} 1 \text{ Indicates a maneuver less than} \\ \text{6 seconds and no steering is required} \\ 0 \text{ Indicates a maneuver duration of at} \\ \text{least 6 seconds, and normal steering} \\ \text{will occur} \end{array} \right.$
$F_L$	Ullage thrust for 2 jets
$K_1$	SPS impulse velocity acquired in a one second maneuver for a unit mass vehicle
$K_2$	SPS minimum impulse constant
$K_3$	SPS minimum impulse constant equal to the slope of minimum impulse curve

The initial computation in Fig. 3.3-11 estimates the velocity-to-be-gained after ullage. There is an option of 2 jet or 4 jet RCS ullage as indicated in Fig. 3.3-11. This subroutine is performed at approximately five seconds prior to ignition, the PIPA's are read at six seconds prior to ignition and the ullage is terminated at two seconds after ignition, consequently it is for this reason about eight seconds of ullage is accounted for as shown in Fig. 3.3-11.

The hand controller signals are observed at the initiation of this routine. The ullage compensation in the time-to-go routine will be made only if +X translation (XTRANS ON) is indicated.

5.3-31

If the SPS were chosen (P-40) Fig. 3.3-11, a check is first made to determine if the maneuver time is less than one second. If the maneuver time is less than one second, the  $t_{go}$  estimate is made on SPS minimum impulse test data represented by the constants  $K_2$  and  $K_3$  (see Section 5.9). In this case no active steering is attempted ( $S_1 = 1$ ). If the maneuver time is greater than one second but less than 6 seconds,  $t_{go}$  is computed as shown in Fig. 3.3-11 and again no active steering is attempted. If the estimated maneuver time is greater than 6 seconds, active steering is used and  $t_{go}$  computations are performed during the maneuver.

With reference to Fig. 3.3-10, if the RCS were chosen for the maneuver, (P-41), the  $t_{go}$  prediction calculation is not made.

If the estimated maneuver time,  $t_{go}$ , for the SPS is less than 6 seconds, the Engine-Off signal is set for the actual ignition time plus  $t_{go}$ .

#### 5.3.3.3.4 Ignition Delay Procedures Caused by Pre-Thrust Computations

The normal pre-thrust computations (sections 5.3.3.3.1 and 5.3.3.3.2) require an extrapolation of the CSM state vector to thirty seconds prior to nominal ignition time, i.e.  $t_{IG} - 30$ . If the Coasting Integration Routine of Sec. 5.2.2 does not complete the extrapolation before  $t_{IG} - 40$  occurs, then an ignition delay procedure occurs as follows:

- 1.) The astronaut is alerted to this condition by a program alarm.
- 2.) The integration continues one step at a time until the CSM state vector time minus the current time minus the integration computing interval (usual 4-5 secs) is greater than 5 seconds. This will permit a 5 second blanking interval (Section 4 State Vector Integration (MID to AVE) Routine R-41).
- 3.) The maneuver ignition time is then redefined to be 30 seconds from the resulting CSM state vector time, and the normal pre-ignition sequence is started.

#### 5.3.3.4 Cross Product Steering Routine

The cross product steering concept is the basic control concept for both External  $\Delta V$  and Lambert Aim Point Guidance modes. The cross product steering logic is shown in Fig. 3, 3-12. The following parameter definitions not previously described apply to this figure.

$\Delta v_{-SM}$	PIPA measured velocity vector over the computation cycle $\Delta t$ in IMU coordinates
$\Delta v$	PIPA measured velocity over the computation cycle $\Delta t$ transformed to the Basic Reference Coordinate System
$\Delta v_p$	A constant which establishes the $\Delta v$ which must be sensed in a 2 second computation interval. It is approximately equivalent to thrusting at 15 % full SPS thrust with fully loaded vehicle for 2 seconds.
$S_w$	A logic switch in the cross product steering routine which when set to 1 allows steering commands and $t_{go}$ calculations to be made. $S_w$ is set to 1 at the fixed time of 2 seconds from engine on command. $S_w$ to set to 0 when the computed time-to-go first becomes less than four seconds.
$\Delta t_{Tail-off}$	A constant representing the duration of a burn at maximum thrust equivalent to the tail-off impulse after the engine-off signal is issued.

$\underline{\Delta v}_G$	$\underline{b} \Delta t - \underline{\Delta v}$
$V_e$	Engine exhaust velocity = $g I_{SP}$
K	Guidance steering gain required for desired dynamic response of the combined powered guidance and thrust control (autopilot) loops
$\underline{\omega}_C$	Commanded attitude rate in the Basic Reference Coordinate System.
$\underline{\omega}_{CNB}$	Commanded attitude rate in Navigation Base Coordinates which is sent to the CSM attitude control system.
c	The cross product steering constant, c, is set equal to one if the pre-thrust program was P-34, P-35, P-38, or P-39, and 1/2 if the pre-thrust program was P-37, (Section 5.4). If the pre-thrust program was the External $\Delta V$ Program, P-30, c is set equal to zero (Fig. 3.3-7). For RCS maneuvers (P-41) the constant c is set to zero before the Lambert Aim Point Maneuver Pre-Thrust computation as shown in Fig. 3.3-9. In pre-thrust program P-31 (Lambert Aim Point Guidance) the value of c is determined by RTCC and uplinked with other target parameters (Fig. 3.3-9).
$\underline{\Delta m}$	$c \underline{b} \Delta t - \underline{\Delta v}$
LOFLG	This flag prevents a premature engine fail indication after a manual engine start  LOFLG $\begin{cases} 1 & \text{steer} \\ 0 & \text{no steer} \end{cases}$
$\Sigma \underline{\Delta v}$	Accumulated PIPA readings used to generate an extrapolation of the Lambert solution. When a new solution becomes available $\Sigma \underline{\Delta v}$ is set to zero.

5.3-34

N	Number of cycles between Lambert solutions. The CMC computation during SPS maneuvers occasionally prevent completion of the Lambert solution in 2 to 4 seconds. The program is capable of processing an N cycle $\underline{v}_G$ update.
$S_F$	Lambert first pass flag. The Lambert first pass occurs at approximately $t_{IG}^{-26}$ . After the first pass it is set to zero.
$S_F$	$\left\{ \begin{array}{l} 1 \text{ First Lambert pass. This results in bypass of the } \underline{b}\Delta t \text{ computation.} \\ 0 \text{ Not first Lambert pass.} \end{array} \right.$

Switch  $S_w$  is set to zero for short duration thrust periods and during the first two seconds of long duration thrust periods. In both of these cases there is no active steering and the vehicle attitude is held at the pre-thrust alignment. When  $S_w$  is set to 1, active steering is performed. The time-to-go,  $t_{go}$  computation and steering commands  $\underline{\omega}_{CNB}$  are performed as shown in Fig. 3.3-12. When the computed  $t_{go}$  becomes less than 4 seconds, then the engine-off signal is set and switch  $S_w$  is reset to zero. For the remainder of the maneuver, no further computations are made except for  $\underline{v}_G$  updating.

With reference to Fig. 3.3-12, the logic switch  $S_w$  is set to zero by the sequencing routine. The only function of the cross product steering routine is to update the velocity-to-be-gained vector  $\underline{v}_G$  with  $\Delta \underline{v}$  and  $\underline{b} \Delta t$ , as long as  $S_w$  is zero.

The steering command generated by this routine is  $\underline{\omega}_{CNB}$  which is in Navigation Base Coordinates. The objective of the cross product steering concept is to control the thrust acceleration vector such that the following condition is satisfied:

$$\underline{v}_G \times (c \underline{b} - \underline{a}_T) = \underline{0}$$

In general, however, there will be a directional error and the steering command  $\underline{\omega}_{CNB}$  tends to align the vehicle such that the above equation is satisfied.

THIS PAGE INTENTIONALLY LEFT BLANK

5. 3-34B

Rev. 3 - 11/68

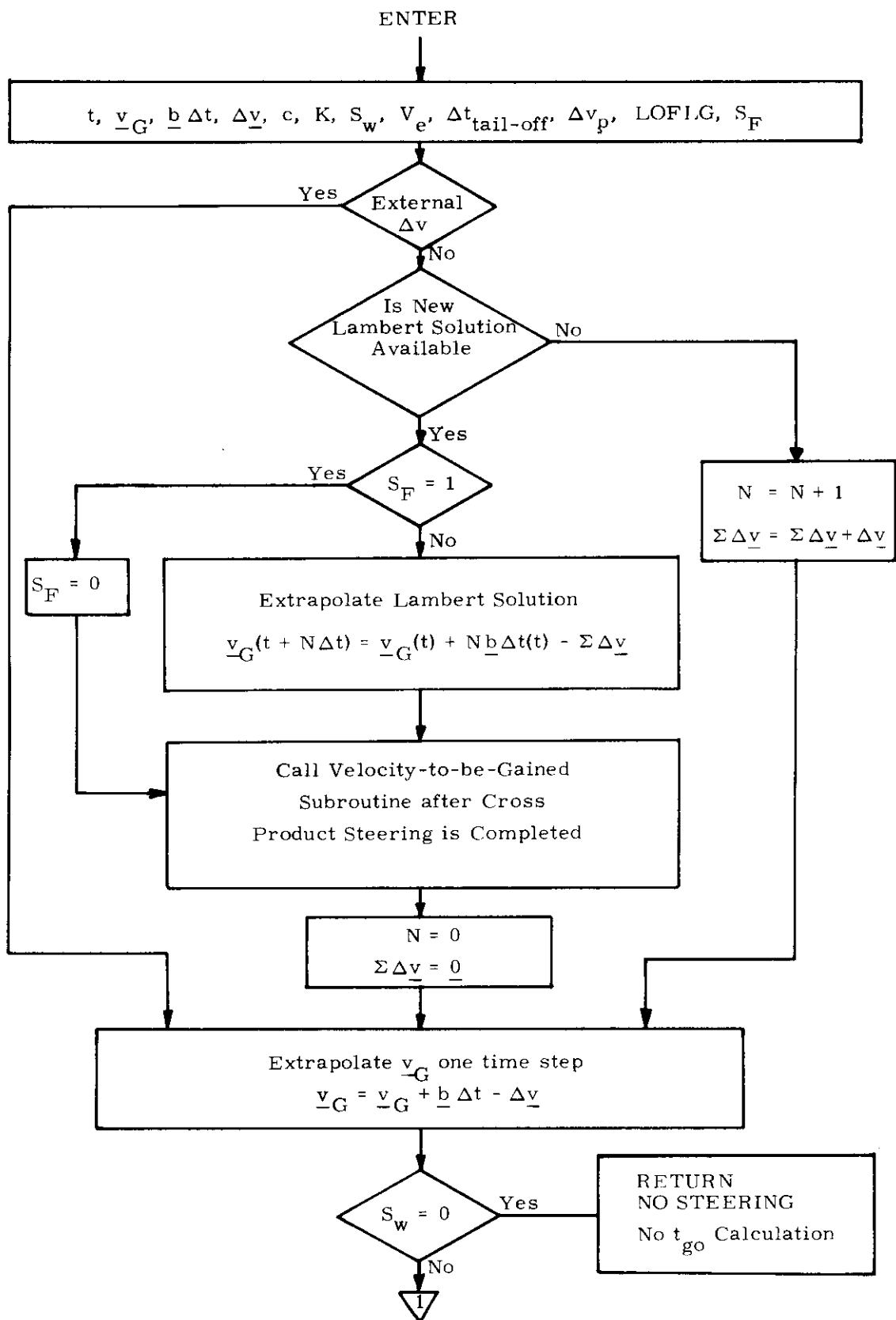


Figure 3.3-12 Cross Product Steering Subroutine  
(page 1 of 2)  
Update  $\underline{v}_G$  Section

5.3-35

Rev. 3 - 11/68

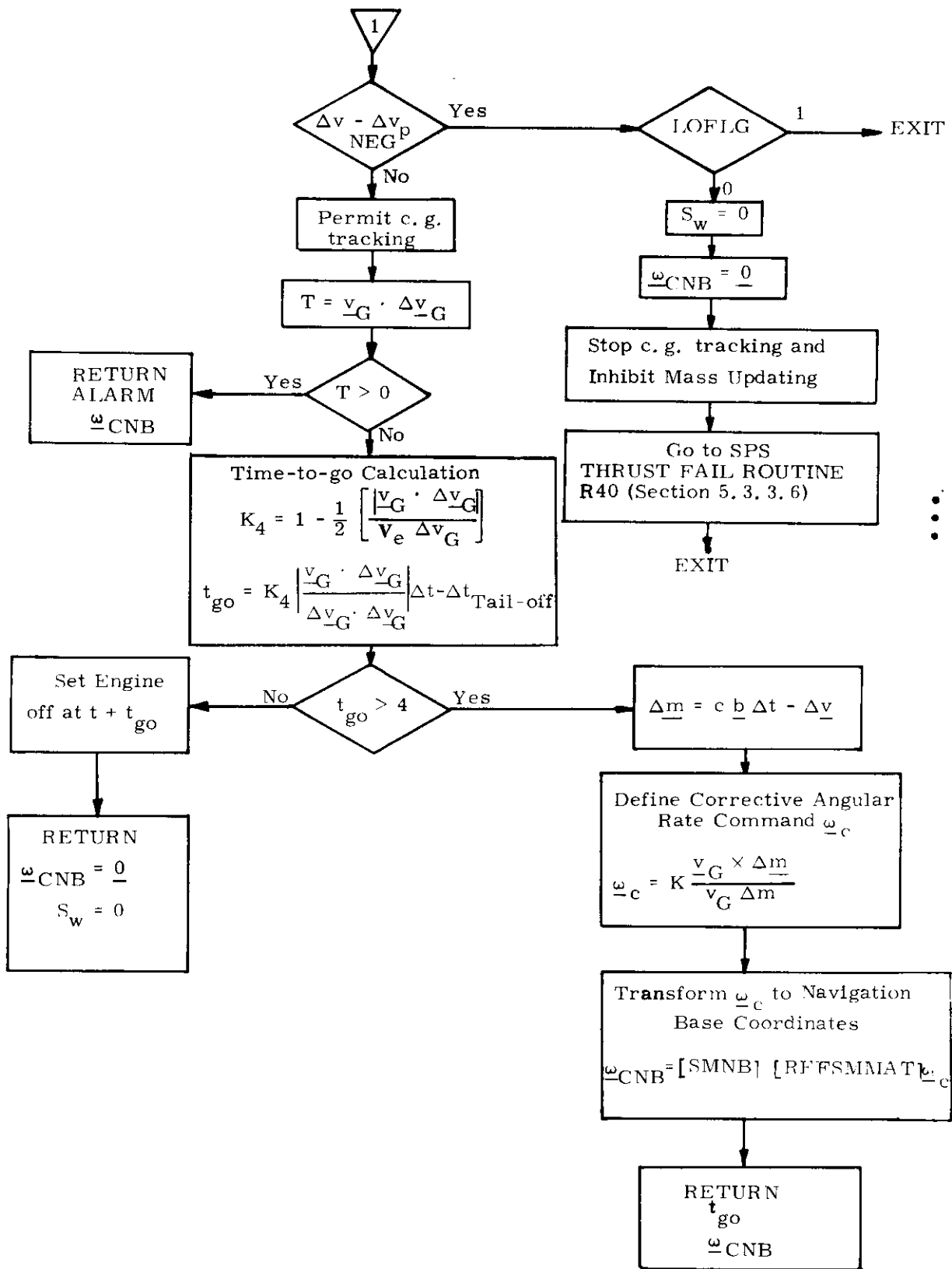


Figure 3.3-12 Cross Product Steering Subroutine  
(page 2 of 2):

$t_{go}$  and Steering Command Computations

5.3-36



The cross-product steering routine, as shown in Fig. 3.3-12 is divided into two sections.

- (a) Update  $\underline{v}_G$ : This portion of the cross-product steering routine provides an up to date velocity-to-be-gained,  $\underline{v}_G$ , for use in the  $t_{go}$  and steering command computations. In the event a new velocity-to-be-gained is not available, the old  $\underline{v}_G$  is extrapolated using the old  $\underline{b} \Delta t$  value. If a new  $\underline{v}_G$  and  $\underline{b} \Delta t$  are available, they are used.
- (b)  $t_{go}$  and steering command calculations: This portion of the cross product steering routine uses the  $\underline{v}_G$  generated in the update  $\underline{v}_G$  section to calculate  $t_{go}$  and steering commands to be sent to the DAP. In addition the following control options are provided:
  - (1) If the present  $\Delta v$  is less than  $\Delta v_p$ , where  $\Delta v_p$  is approximately equivalent to thrusting at 15% full SPS thrust with fully loaded vehicle for 2 secs., the steering rate commands are zeroed, c. g. tracking is stopped and the SPS Thrust Fail Routine (R-40) (Section 5.3.3.5) is entered.  $\Delta v_p$  is an erasable quantity.
  - (2) An alarm is issued if  $\underline{v}_G \cdot \Delta \underline{v}_G$  is positive. This is equivalent to pointing the thrust vector in the wrong direction.

### 5.3.3.5 Velocity-to-be-Gained Routine

The velocity-to-be-gained computations shown in Fig. 3.3-13 are those carried out during the Lambert Aim Point powered flight guidance. The velocity-to-be-gained computation for the External  $\Delta V$  guidance mode is simpler than that for the Lambert Aim Point guidance mode. The External  $\Delta V$  velocity-to-be-gained computation is that shown in the cross product steering routine of Fig. 3.3-12 and is equivalent to

$$\underline{v}_G = \underline{v}_G - \underline{\Delta v}$$

since  $\underline{b}\Delta t = \underline{0}$  for the External  $\Delta V$  mode as shown in Fig. 3.3-7.

The velocity-to-be-gained computations for the Lambert Aim Point guidance mode involve the determination of a new  $\underline{v}_G$  by processing the Lambert Subroutine via the Initial Velocity Subroutine. A second objective is the computation of a new  $\underline{b}\Delta t$  parameter for use by the cross product steering routine. A new required velocity,  $\underline{v}_R$ , is also determined for use in the next computation cycle of the velocity-to-be-gained subroutine. The CMC computations during SPS maneuvers occasionally prevent completion of the Lambert Solution in 2 to 4 seconds. The program is capable of processing an N cycle  $\underline{v}_G$  update (Fig. 3.3-13).

The following parameter definitions refer to Fig. 3.3-13.

$\underline{r}(t)$	}	active vehicle state vector
$\underline{v}(t)$		
t		
$\underline{r}(t_2)$		Offset target intercept position vector at time $t_2$ . This parameter is determined by the preceding targeting program (P-34, P-35, P-37, P-38, P-39, or P-31).

$t_2$	Intercept time of arrival associated with the offset target vector $\underline{r}(t_2)$ . This is an input target parameter.
$t_{IG}$	Nominal ignition time.
$\underline{v}_R(t - \tau)$	Required velocity vector at the preceding Lambert computation cycle.
$\tau$	Time interval between the current and previous Lambert computation cycle = $N\Delta t$ .
$S_R$	Target rotation switch set in P-34, P-35, P-38 or P-39 indicating that the target vector was rotated into the orbital plane due to proximity to $180^\circ$ transfer.
	$S_R \begin{cases} 1 & \text{Target vector rotated} \\ 0 & \text{No rotation} \end{cases}$
$\underline{g}$	Total gravity acceleration vector (Section 5.3.2).
$\underline{g}_b$	Component of the earth gravity acceleration vector representing earth oblateness effects (Section 5.3.2).
$\Delta \underline{v}(t)$	Measured PIPA velocity change in the Basic Reference Coordinate system.
$N_1$	Number of target offset iterations desired in the Initial Velocity Subroutine (Section 5.5.11).
$\epsilon$	Initial Velocity Subroutine parameter.
$\underline{v}_R(t)$	Required velocity vector at time $t$ .
$\underline{v}_G(t)$	Velocity-to-be-gained vector at time $t$ .
$\underline{v}_G(t + \Delta t)$	Extrapolated $\underline{v}_G$ one computation time step.

N            Number of cycles between Lambert solutions. The CMC computation during SPS maneuvers occasionally prevent completion of the Lambert solution in 2 to 4 seconds. The program is capable of processing an N cycle  $v_G$  update.

$S_F$         Lambert first pass flag. The Lambert first pass occurs at approximately  $t_{IG} - 26$ . After the first pass it is set to zero.

$S_F$          $\left\{ \begin{array}{l} 1 \quad \text{First Lambert pass. This results} \\ \quad \text{in bypass of the } \underline{b}\Delta t \text{ computation.} \\ 0 \quad \text{Not first Lambert pass.} \end{array} \right.$

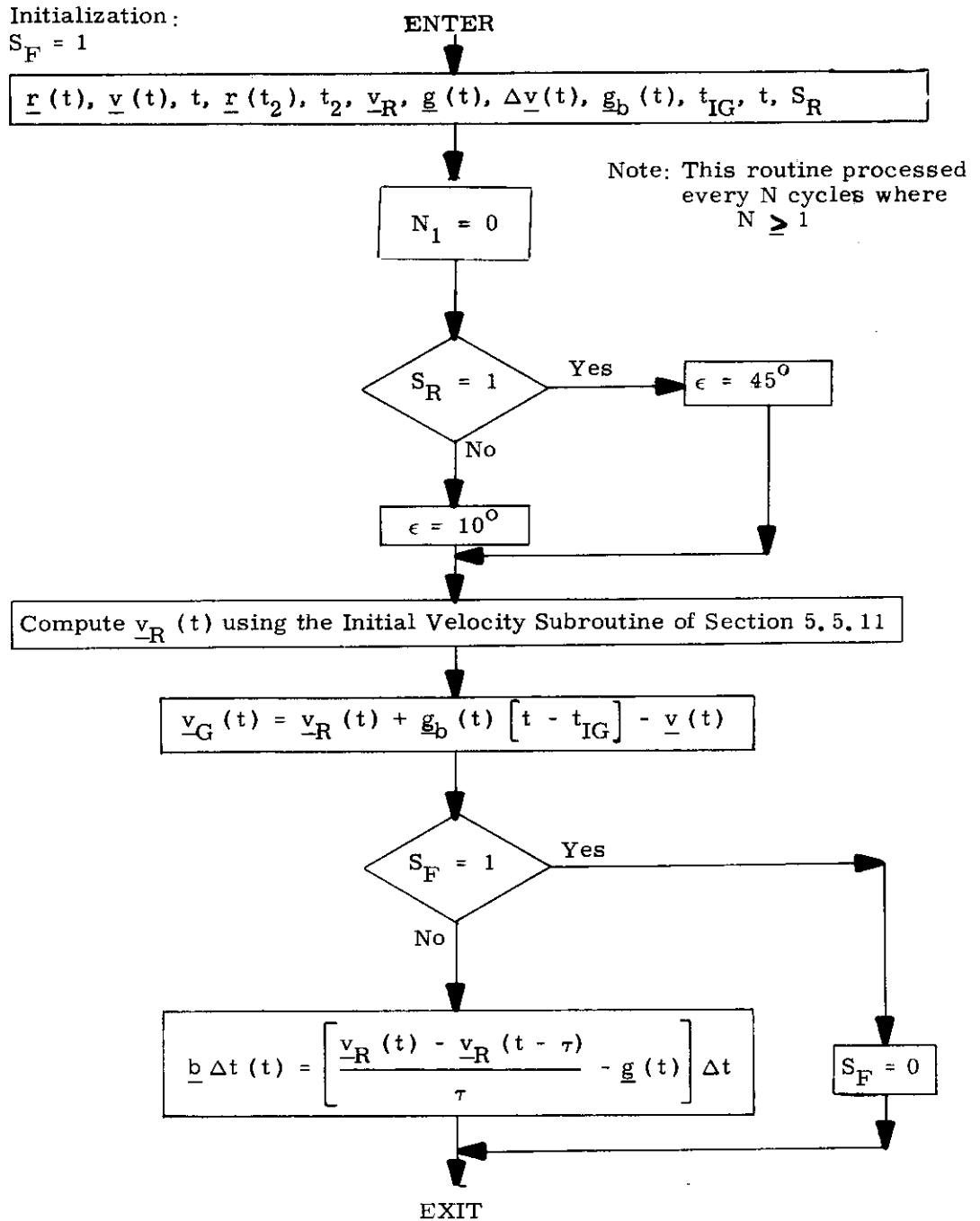


Figure 3.3-13 Velocity-to-be-Gained Subroutine for Lambert Aim Point Guidance

THIS PAGE INTENTIONALLY LEFT BLANK

5.3-42

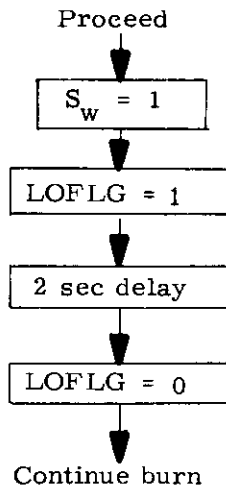
Rev. 3 - 11/68

It should be noted that in Fig. 3.3-13 the velocity-to-be-gained,  $\underline{v}_G$ , derived from the Lambert solution using an offset target vector is modified by the term  $\underline{g}_b(t) [t - t_{IG}]$ . This term is an approximation to the velocity change contributed by the earth oblateness effect. The compensation used in this subroutine is computed as the current oblateness acceleration,  $\underline{g}_b(t)$ , multiplied by the time since nominal ignition  $(t - t_{IG})$  where  $t_{IG}$  is the nominal ignition time. This correction is zero for lunar orbits. The objective of this correction is to reduce cut-off errors due to finite maneuver time effects, and to minimize commanded thrust attitude variations during the maneuver. These two effects occur during long maneuvers because in accounting for earth oblateness effects in the initial targeting programs (P-34 and P-35), it is assumed that an impulsive maneuver will be applied at ignition time. Since a finite maneuver time is required, the pre-computed target aim point becomes less accurate as the maneuver progresses. The  $\underline{g}_b [t - t_{IG}]$  correction is an approximate substitute for a retargeting procedure which cannot be performed during a powered maneuver.

5.3.3.6 SPS Thrust Fail Routine (R-40)

The purpose of the SPS Engine Fail Routine (R-40) is to present the Astronaut with 3 options when a low SPS thrust is detected in the Cross Product Steering Subroutine (Section 5.3.3.4). The three options are:

- (1) Command engine off and terminate P-40.
- (2) Command engine off and return to the P-40 point where the impulse burn test is made.
- (3) Proceed with the burn but with the thrust failure detection inhibited for 2 secs to prevent premature thrust fail indication as shown below:



See Section 4 for complete details of the R-40 routine



#### 5.3.4 THRUST MONITOR PROGRAM

The Thrust Monitor Program, P-47, is used during manual or non-GNCS controlled maneuvers to monitor and display the velocity change applied to the vehicle. The program first suspends state vector updating by the VHF Range-link (resets the update and track flags), and advances the vehicle state vector to the current time by the Coasting Integration Routine of Section 5.2.2. This operation is continued until the state vector is advanced several seconds ahead of the current time as described in Section 5.3.3.3.4. The Average-G Routine of Section 5.3.2 is then initiated to allow thrusting to be started as soon as possible. The Average-G Routine is left on until the program is terminated after completion of the maneuver. The primary output of P-47 is the measured maneuver  $\Delta V$  in vehicle coordinates as described in Section 4.

The two major maneuvers during which the Thrust Monitor Program is normally used are the Translunar Injection (TLI) maneuver controlled by the Saturn guidance system, and the manually controlled terminal rendezvous maneuvers required for a CSM retrieval of the LM. During the TLI maneuver a callable display of inertial velocity, altitude above the launch pad radius and altitude rate is available to the astronaut. The altitude display parameter in this program is only valid outside the lunar sphere of influence.

During active CSM terminal rendezvous maneuvers, the Rendezvous Display Routine R-31 is normally called to display relative range, range rate, and the vehicle X axis to the horizontal plane angle  $\theta$ . The operation of R-31 with the Average-G Routine of P-47 is described in Section 5.6.7.1.

### 5.3.5 EARTH ORBIT INSERTION MONITOR PROGRAM - P-11

#### 5.3.5.1 Introduction

The purpose of this section is to describe the operation and implementation of Program P-11, Earth Orbit Insertion Monitor.

This program is initiated by Program P-02, Gyro Compassing, when the liftoff discrete is detected or by the astronaut backup Verb 75 and it performs the following functions (in the order of occurrence)

- (1) Zeroes the CMC clock at liftoff and updates the reference ephemeris time (see Time Definition, Section 5.1.5.5). (Time Subroutine)
- (2) Computes the CMC state vector (in Basic Reference Coordinates) at liftoff and starts the Average-G computation (see Section 5.3.2) using this state vector. (State Subroutine)
- (3) Computes the matrix REFSMMAT which relates the IMU Stable Member orientation to the Basic Reference Coordinate System. (State Subroutine)
- (4) Computes periodically the error between the nominal desired Saturn Launch Vehicle attitude (as determined by a time dependent polynomial) and the actual Saturn Launch Vehicle attitude (as determined by the CM IMU), and transmits the error to the CDU for display by the FDAI.\* (Attitude Error Subroutine)

---

\* Flight Director Attitude Indicator.

- (5) Periodically computes Saturn Launch Vehicle inertial velocity magnitude, rate of change of the vehicle altitude above the launch pad radius, and the vehicle altitude above launch pad radius and displays this information via the DSKY. (Display Subroutine)

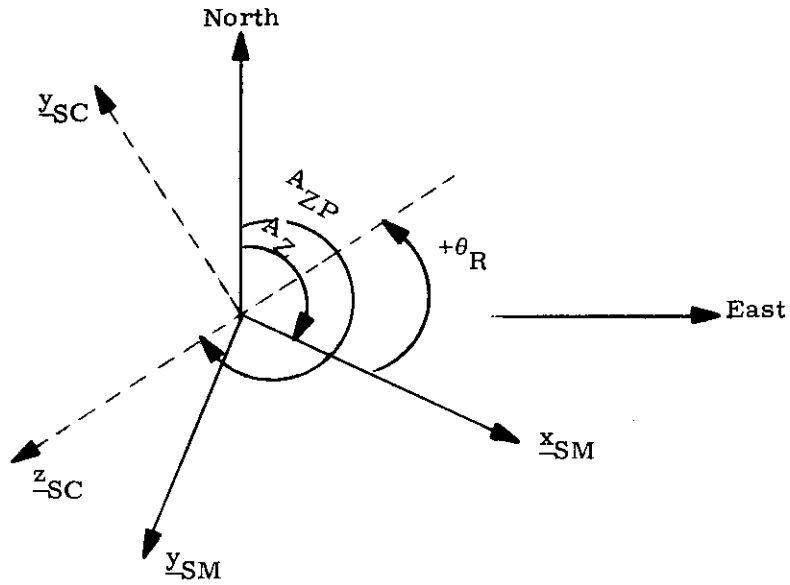
Program P-11, when activated by Program P-02 assumes

- (1) The following information has been pad loaded

- $A_{ZP}$  - Azimuth of the launch vehicle on the pad measured from north to the +Z spacecraft axis positive in a right hand sense about the inward pad local vertical. See figure 3.5-1. •  
•  
•
- $A_Z$  - launch azimuth measured from north to the X IMU Stable member axis positive in the same sense as  $A_{ZP}$ . See figure 3.5-1. •  
•  
•
- $K_r$  - constant denoting the absolute value of the rate at which the Saturn will roll from the pad azimuth to the launch azimuth.
- $t_{E1}$  - the time from liftoff at which the initial Saturn roll will commence.
- $t_{E2}$  - the increment in time after  $t_{E1}$  to when the display of Saturn attitude error on the FDAI will be held constant.
- $a_0, a_1, \dots, a_6$  - coefficients of a 6th order polynomial in time  $t$ , ( $t_{E1} < t < (t_{E1} + t_{E2})$ ) describing the nominal Saturn pitch profile.

- (2) The clock and reference ephemeris time were synchronized previously.

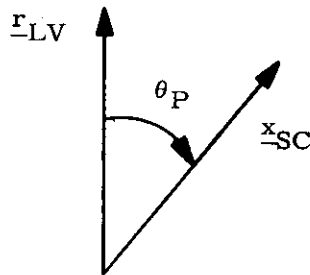
Azimuth and Roll Angle Definitions



- . Plane of the paper is the pad local horizontal
- . Up is out of the paper
- . Spacecraft and IMU axis shown prior to launch
- . After lift-off the Saturn rolls to align the  $-z_{SC}$  with  $x_{SM}$

Pitch Angle Definition

Inertial upward local vertical vector (established at liftoff)



- . Plane of the paper is the instantaneous pitch plane
- .  $y_{SC}$  is into the paper and northerly

Figure 3.5-1 Angle Definitions

•  
•

5. 3. 5. 2 Nomenclature for P-11

t	Computer clock reading at any time t	•
t <sub>0</sub>	See Section 5. 1. 5. 5.	•
h	Vehicle altitude above a sphere whose radius is that of the launch pad	•
$\dot{h}$	Rate of change of altitude measured in the direction of the vehicle radius vector	•
v	Inertial velocity magnitude of the vehicle	•
Lat <sub>P</sub>	Launchpad geodetic latitude	•
Lon <sub>P</sub>	Launchpad geodetic longitude	•
Alt <sub>I</sub>	IMU altitude above the launch pad	•
$\underline{U}_z$	A vector in the Basic Reference Coordinate System in the direction of the earth's rotation	•
$\omega_E$	The angular velocity of the earth	•
[REFSMMAT]	A matrix whose rows are the location of the IMU Stable Member axes in the Basic Reference Coordinate System	•
CDU	The current value of the IMU Gimbal Angles	•
$\theta_P$	The Saturn Vehicle nominal pitch angle measured from the Launch pad local vertical at liftoff to the Saturn X axis (see figure 3. 5-1).	•
$\theta_R$	The Saturn nominal roll angle measured from the X IMU Stable Member axis to the negative Z spacecraft axis. (see Fig. 3. 5-1).	•
$\underline{E}_a$	A vector representing the pitch, yaw and roll errors about CM body axes (in a right hand sense).	•
DGA	A vector representing the Saturn nominal desired IMU gimbal angles.	•
$\left(\frac{b}{a}\right)^2$	Square of the ratio of the earth semi minor axis to the semi major axis.	•

$K_z$  A constant computed in the State Subroutine which is utilized by the Attitude Error Display Subroutine.

$t_{KS}$  The elapsed time from lift-off to when the attitude error display is disabled. (Note this constant is in fixed memory).

Note that pad loaded variables are identified in Section 5.3.5.1.

### 5.3.5.3 Time Sequencing of P-11 Subroutines

Program P-11 is composed of four Subroutines; Time Subroutine, State Subroutine, Attitude Error Subroutine and Display Subroutine.

The Time Subroutine is selected by P-11 within 0.5 second of the lift-off discrete receipt. The State Subroutine is initiated immediately after the Time Subroutine and is through within 1 second after receipt of the lift-off discrete.

The cycling of the Attitude Error Subroutine is then started. This subroutine refreshes the attitude error FDAI display approximately every 0.5 second until  $t = (t_{E1} + t_{E2})$ . From time  $(t_{E1} + t_{E2})$  until 163.84 seconds the error is held constant at its value at time  $(t_{E1} + t_{E2})$ . At  $t = 163.84$  seconds the display is disabled.

At the same time the attitude error display is started, the cycling of the display subroutine is started. This subroutine displays  $v$ ,  $h$ , and  $\dot{h}$  via the DSKY. It is cycled every 2 seconds following the Average-G computations and is out of synchronization with the Attitude Error Subroutine.

### 5.3.5.4 Time Subroutine

The procedure for clock zeroing and presetting of reference time,  $t_0$ , is shown in Fig. 3.5-2. Refer to Time Definitions, Section 5.1.5.5, for a description of how the reference ephemeris time was originally synchronized with the AGC clock prior to lift-off.

This activity does not occur precisely at lift-off, but within 0.5 second maximum of the receipt of the lift-off discrete. In any event, the clock zeroing and the constant  $t_0$  are not changed for the remainder of the mission unless P-27 intercedes.

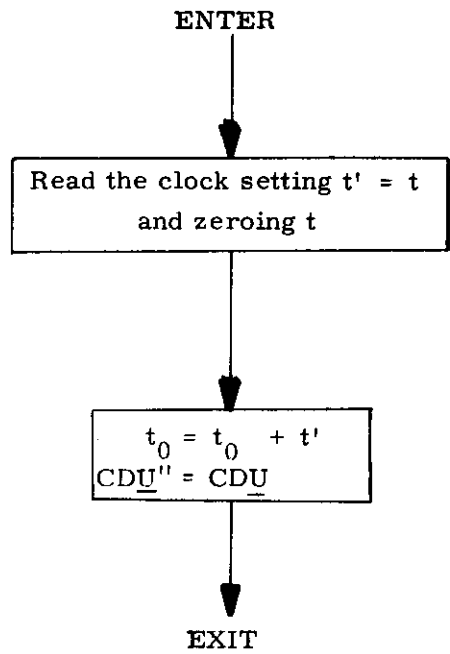


Figure 3.5-2 Time Subroutine



#### 5.3.5.5 State Subroutine

The state vector of the vehicle at lift-off in Basic Reference Coordinates is that imparted by the earth. The first part of Fig. 3.5-3 shows this computation.

REFSMMAT, the matrix which relates the IMU stable member orientation at lift-off to the Basic Reference Coordinates, is determined as shown on page 2 of Fig. 3.5-3. This computation assumes that the stable member X and Y axes are normal to the pad local vertical and the X axis is aligned along the launch azimuth  $A_Z$  pointing down range and the Z axis toward the center of the earth. This computation determines the local vertical utilizing the characteristics of the earth's reference ellipsoid.

These computations take place within 0.75 second of the receipt of the lift-off discrete by Program P-11.

#### Attitude Error Subroutine

This subroutine computes and transmits to the FDAI the difference between a stored nominal Saturn Launch Vehicle attitude profile and the actual attitude profile as measured by the CM inertial measurement unit. Figures 3.5-4 and 3.5-5 present the details of these computations.

This subroutine is cycled approximately every 0.5 seconds and is terminated when mission time  $t$  reaches a preset value  $t_{KS}$ .

#### 5.3.5.6 Display Subroutine

The computations of the display quantities  $v$ ,  $h$ ,  $\dot{h}$  are shown on Fig. 3.5-6. Callable display parameters available during P-11 operation are described in Section 5.6.10, Orbital Parameter Display Computations.

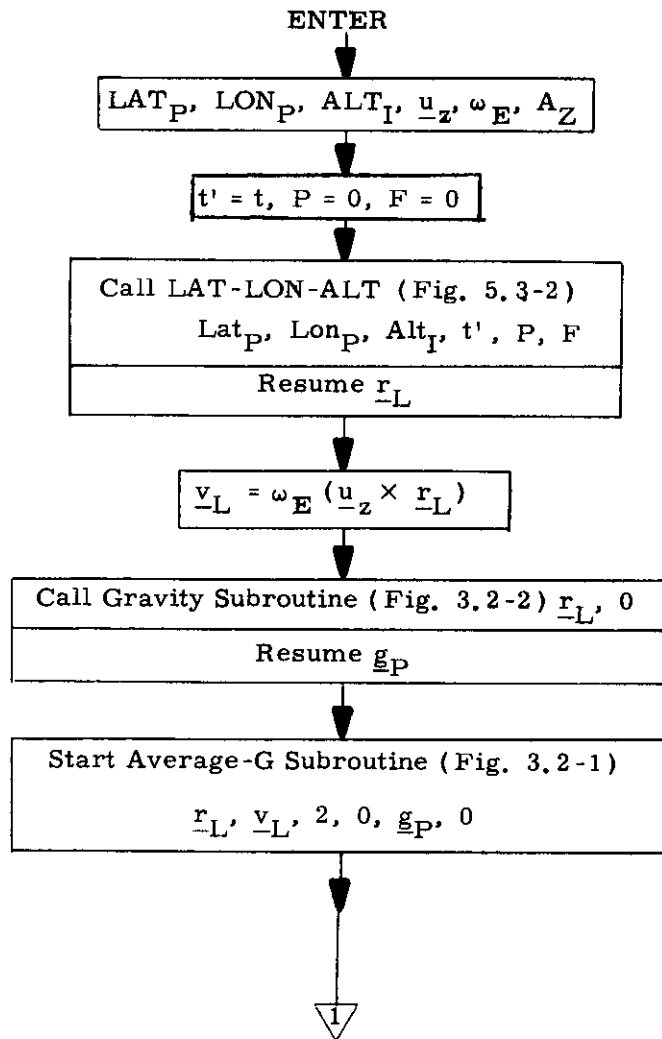


Figure 3.5-3 State Subroutine  
(page 1 of 2)

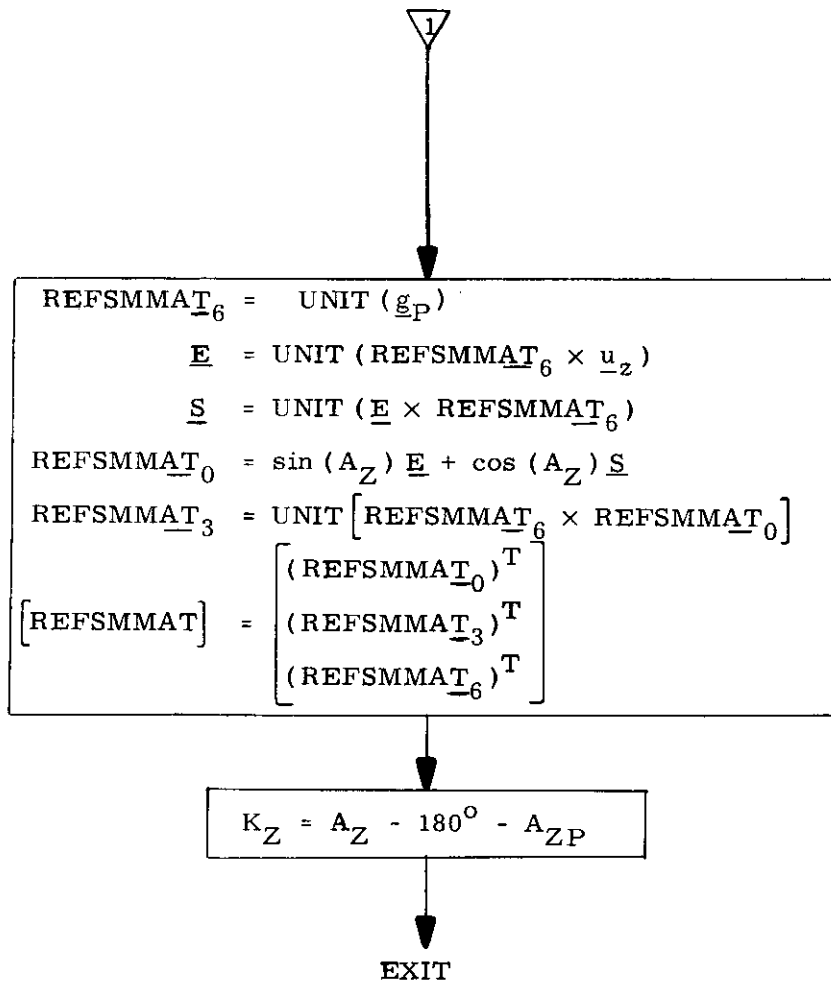


Figure 3.5-3 State Subroutine  
(page 2 of 2)

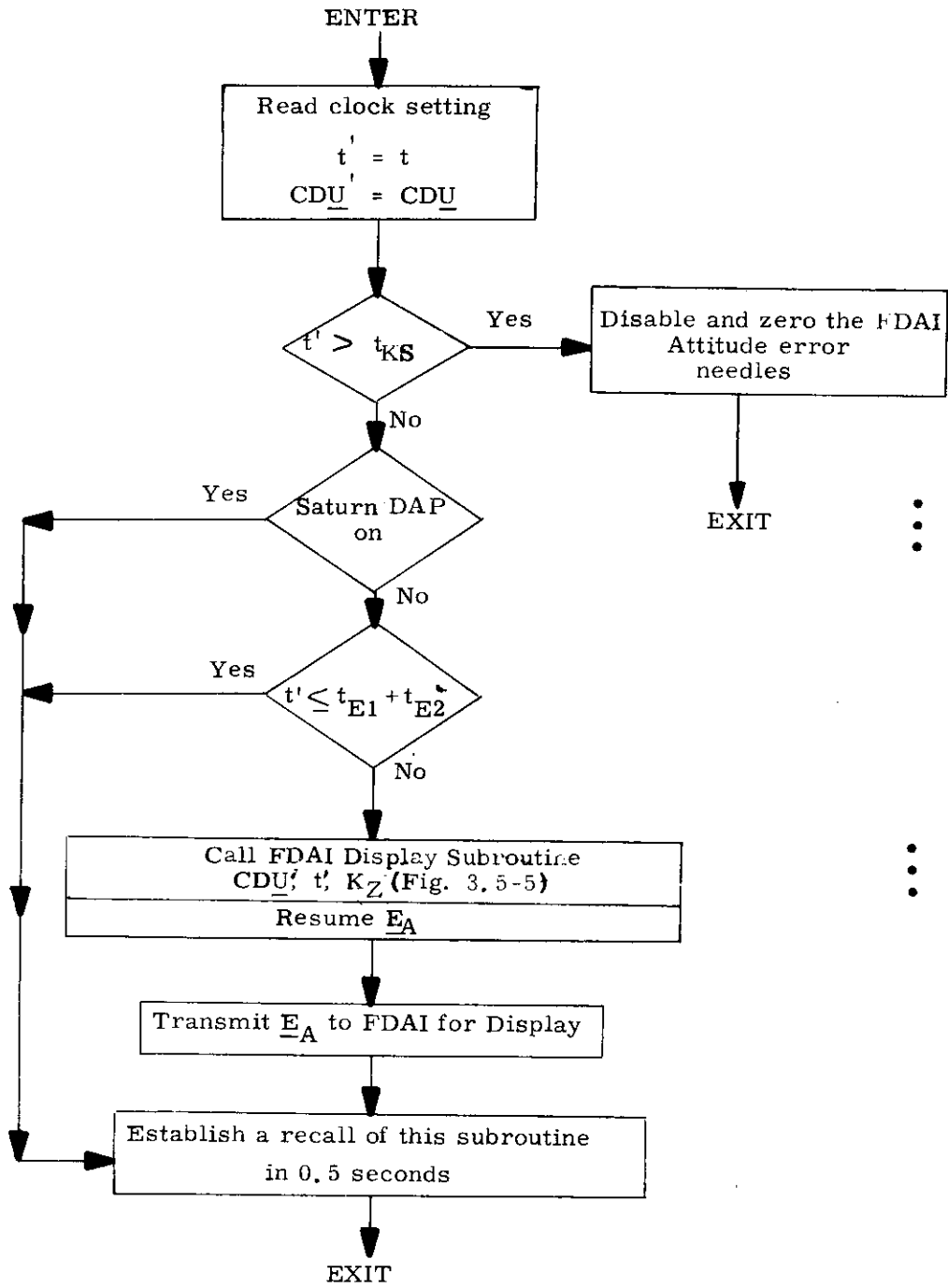


Figure 3.5-4 Attitude Error Subroutine

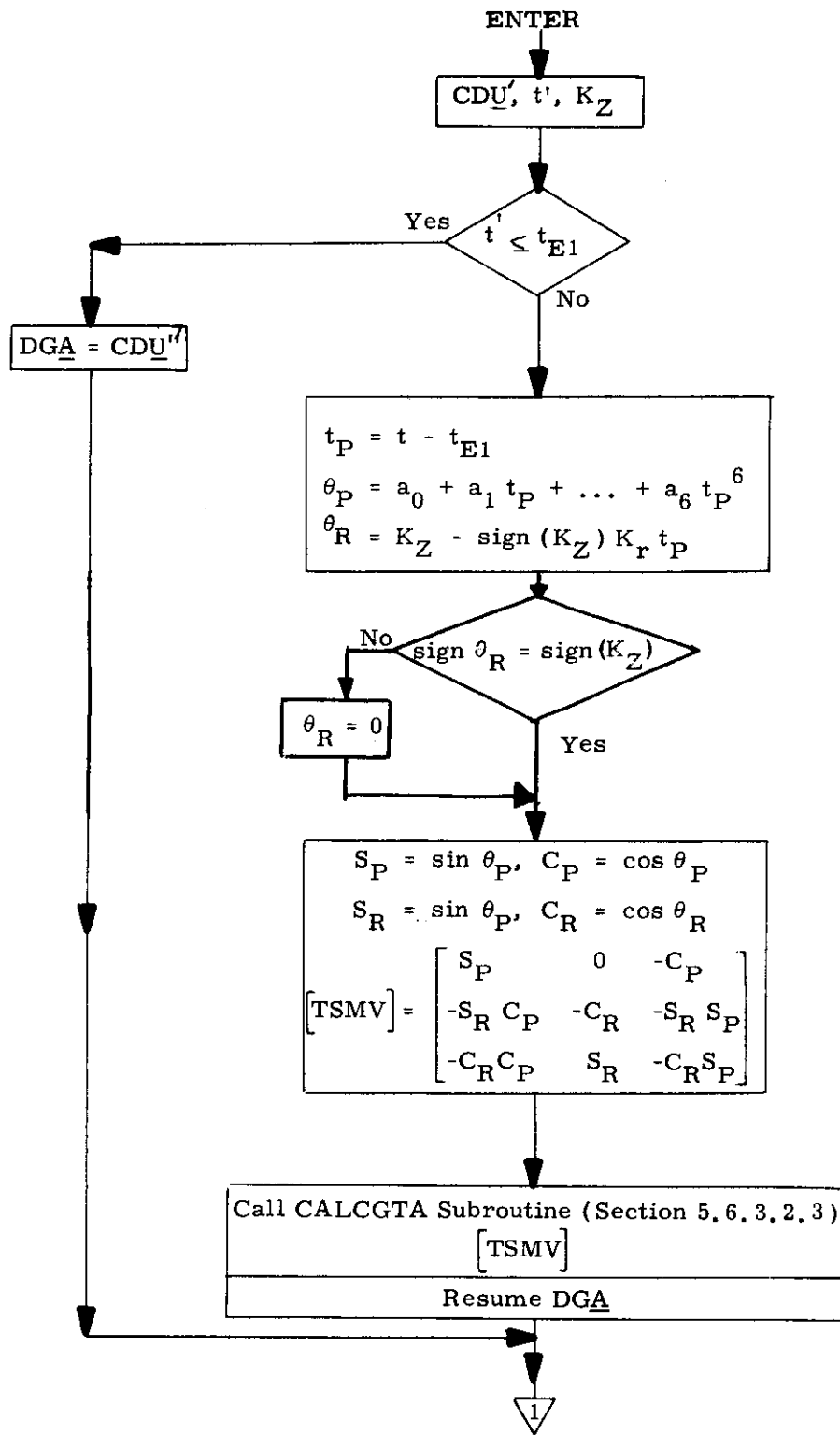


Figure 3. 5-5 FDAI Display Subroutine  
(page 1 of 2)

5. 3-57

Revised COLOSSUS

Added GSOP #R-577 PCR # 628

Rev. 3

Date 11-8-68

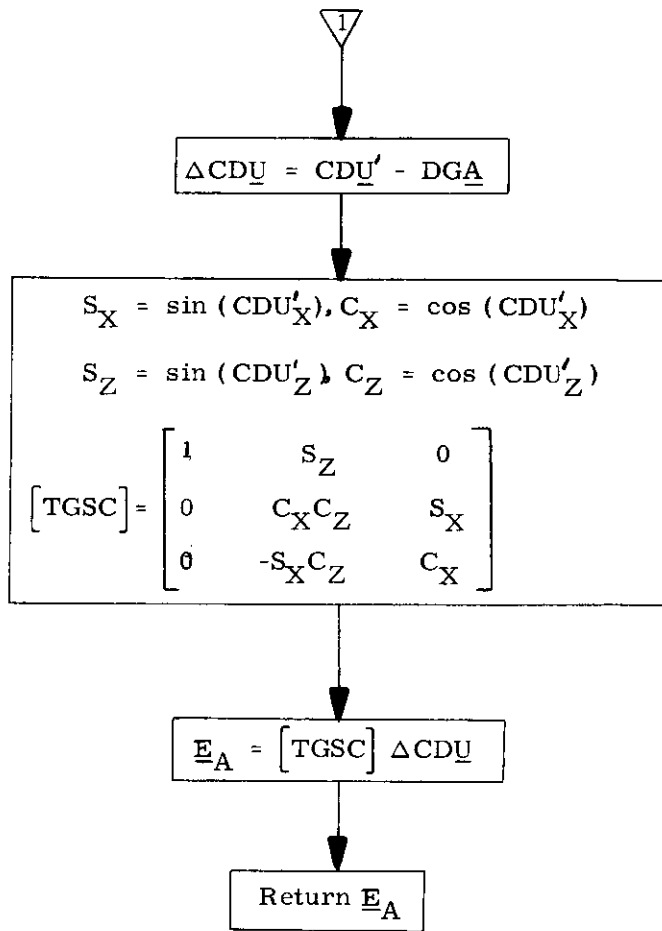


Figure 3.5-5 FDAI Display Subroutine  
(page 2 of 2)

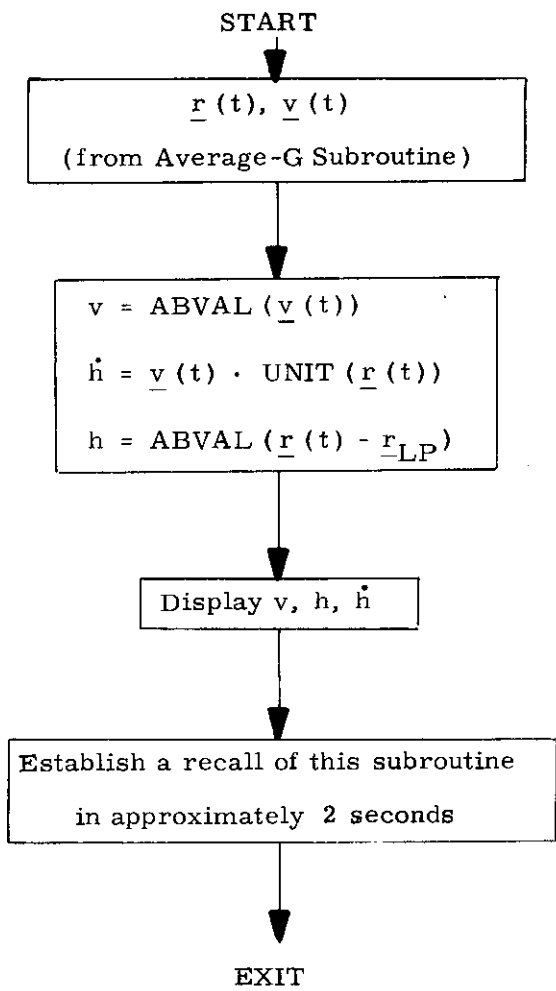


Figure 3.5-6 Display Subroutine

THIS PAGE INTENTIONALLY LEFT BLANK

5.3-60

Rev. 3 - 11/68



## 5.4 TARGETING ROUTINES

### 5.4.1 GENERAL COMMENTS

The objectives of the targeting routines presented in this section are to provide a CMC capability to determine the required input target parameters and set control modes for the various powered flight guidance routines of Section 5.3. It should be noted that the CMC does not have a targeting capability for every phase of the lunar landing mission, and therefore must rely on RTCC targeting for many of the nominal and abort phases of the lunar mission. The restricted targeting capability provided by the CMC is presented in the following subsections:

#### 5.4.4 Rendezvous Retrieval Targeting Routines

These include TPI and Rendezvous Midcourse Correction programs, a TPI Search Program, and Stable Orbit Rendezvous routines.

#### 5.4.5 Return-to-Earth Targeting Routines

These routines are restricted to operation outside the lunar sphere of influence for abort returns to earth.

All CMC targeting programs use the Basic Reference Coordinate System defined in Section 5.1.4.1. The Rendezvous Retrieval Targeting Routines of Section 5.4.4 can be used in either earth or lunar orbits. The basic input parameters required by the targeting routines of this section are the vehicle state vector estimates determined by the navigation programs of Section 5.2.

#### 5. 4. 4 RENDEZVOUS RETRIEVAL TARGETING

##### 5. 4. 4. 1 General

The Rendezvous Guidance Program in the CMC may be used to provide a direct intercept trajectory and the midcourse maneuvers required for rendezvous. The two programs necessary are designated pre-TPI (Transfer Phase Initiation) and Midcourse, and are identical to the last two phases of the Concentric Rendezvous Program in the LM Guidance Computer (LGC).

In addition, the TPI Search Program provides a technique for minimizing the transfer  $\Delta V$  required for rendezvous. An alternate rendezvous procedure is also available using the Stable Orbit Rendezvous Routines.

All computations performed in the Rendezvous Guidance Program use the Basic Reference Coordinate System defined in Section 5. 1. 4. 1.

Rendezvous targeting programs have a basic dependence on the state vector estimates which are obtained from the navigation routines described in Sections 5. 2 and 5. 3. 2.

For the routines in this section there are two associated program numbers based on whether the CSM or the LM is the active vehicle. If the astronaut elects to make the LM active, the only change is the program number selected. All equations and program operations in Section 5. 4. 4 are identical for these two modes of operation. The function  $SGN(X)$  used in Figs. 4. 4-2 and 4. 4-3 is set equal to +1 if  $X$  is zero or a positive number, otherwise it is set equal to -1.

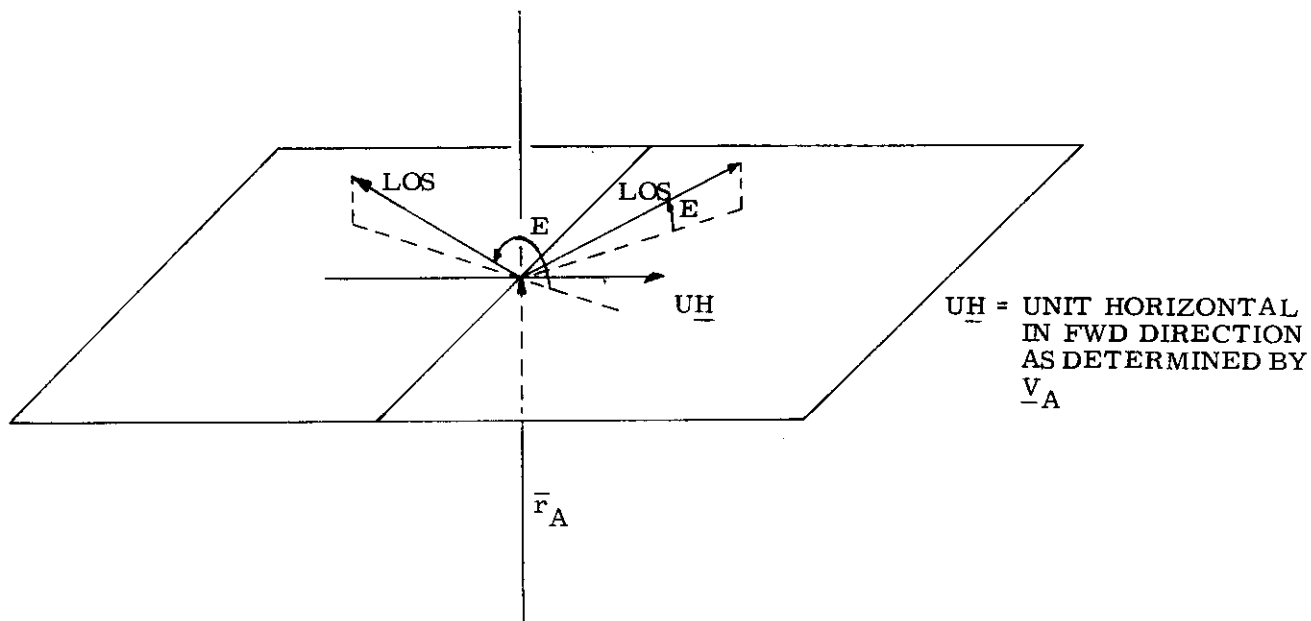
#### 5. 4. 4. 2 Pre-TPI Maneuver

This program corresponds to Program P-34 (CSM active) or P-74 (LM active) of Section 4. Its objective is to establish the terminal phase initiation (TPI) maneuver. The position of the TPI maneuver is determined by specifying either the TPI time or the elevation angle which specified the relative geometry of the vehicles at the TPI point. The astronaut inputs are:

1. Choice of active vehicle (P-34 CSM, P-74 LM)
2. Time  $t$  of the TPI maneuver
3. Elevation angle  $E$  (set equal to zero if  $t$  is specified), defined in Fig. 4. 4-1.
4. Central angle  $\omega t$  of the passive vehicle between the TPF and TPI points.

The active  $\underline{r}_{AI}$ ,  $\underline{v}_{AI}$  and passive vehicle  $\underline{r}_{PI}$ ,  $\underline{v}_{PI}$  state vectors are available in the guidance computer. The program starts with a precision update of these vectors to the TPI time.

If the elevation angle is not specified, it is computed after the precision updating of both vehicles to the TPI maneuver time. If  $E$  is specified, an iteration procedure is initiated to find the TPI time at which  $E$  is attained. This procedure uses the input TPI time as the initial guess and is based on conic trajectories (except for the original precision update). The time correction  $\delta t$  is based on (1) the angular distance between  $\underline{r}_p$  and the desired position of the passive vehicle, obtained by assuming the passive vehicle is in a circular orbit, and (2) assuming the vehicles are moving at a constant angular rate. The flow diagram for this procedure is shown in Fig. 4. 4-2.



- 1) IF THE LOS PROJECTION ON  $\underline{UH}$  IS POSITIVE:
  - a) WHEN THE LOS IS ABOVE THE HORIZONTAL PLANE  
 $0 < E < \pi/2$
  - b) WHEN THE LOS IS BELOW THE HORIZONTAL PLANE  
 $3\pi/2 < E < 2\pi$
  
- 2) IF THE LOS PROJECTION ON  $\underline{UH}$  IS NEGATIVE
  - a) WHEN THE LOS IS ABOVE THE HORIZONTAL PLANE  
 $\pi/2 < E < \pi$
  - b) WHEN THE LOS IS BELOW THE HORIZONTAL PLANE  
 $\pi < E < 3\pi/2$

Figure 4. 4-1 Definition of Elevation Angle, E

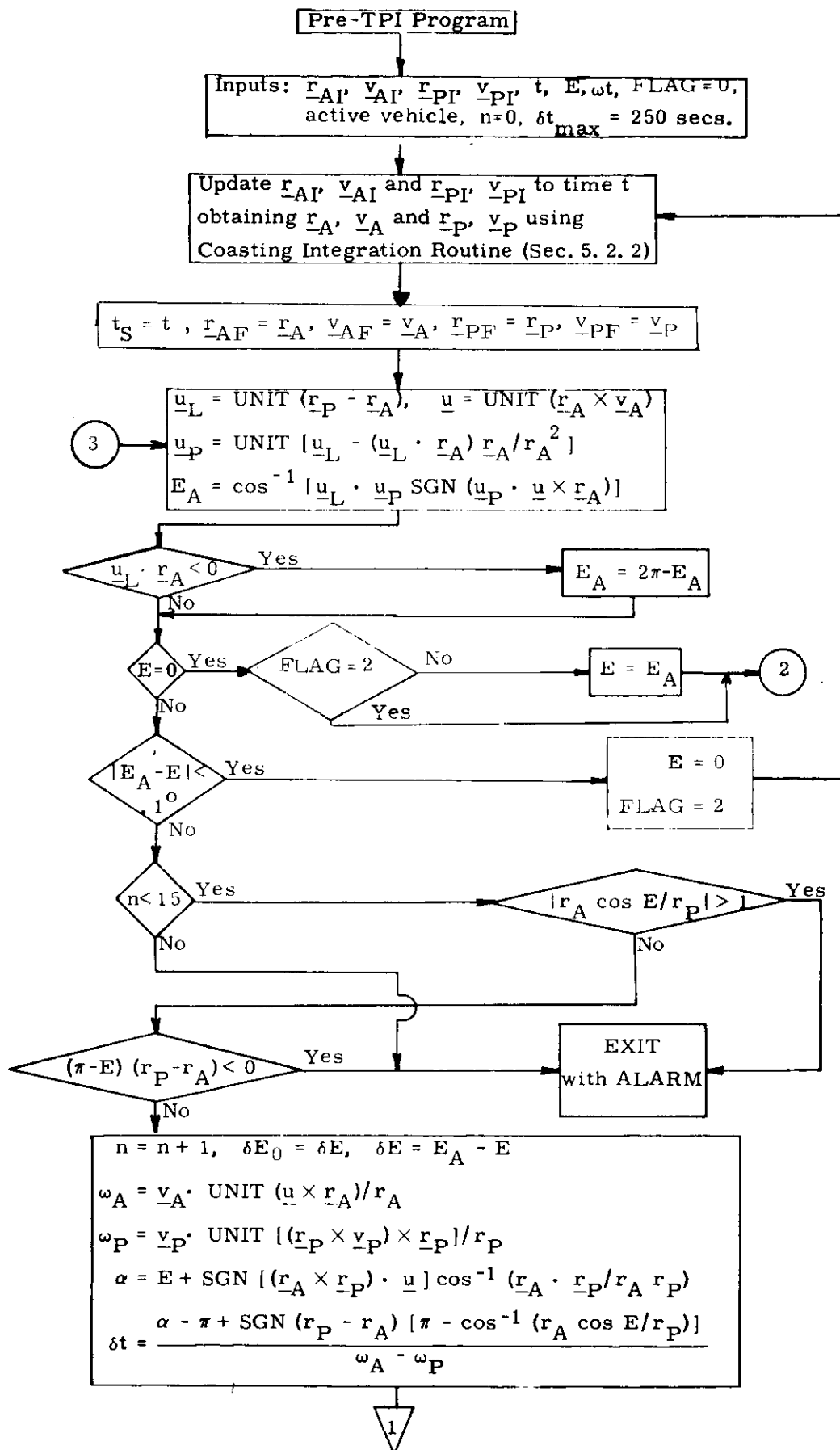


Figure 4. 4-2 Pre-TPI Maneuver Program (page 1 of 3)

5. 4-5

Revised COLOSSUS

Added GSOP #R-577 PCR # 628

Rev. 3

Date 11-6-68

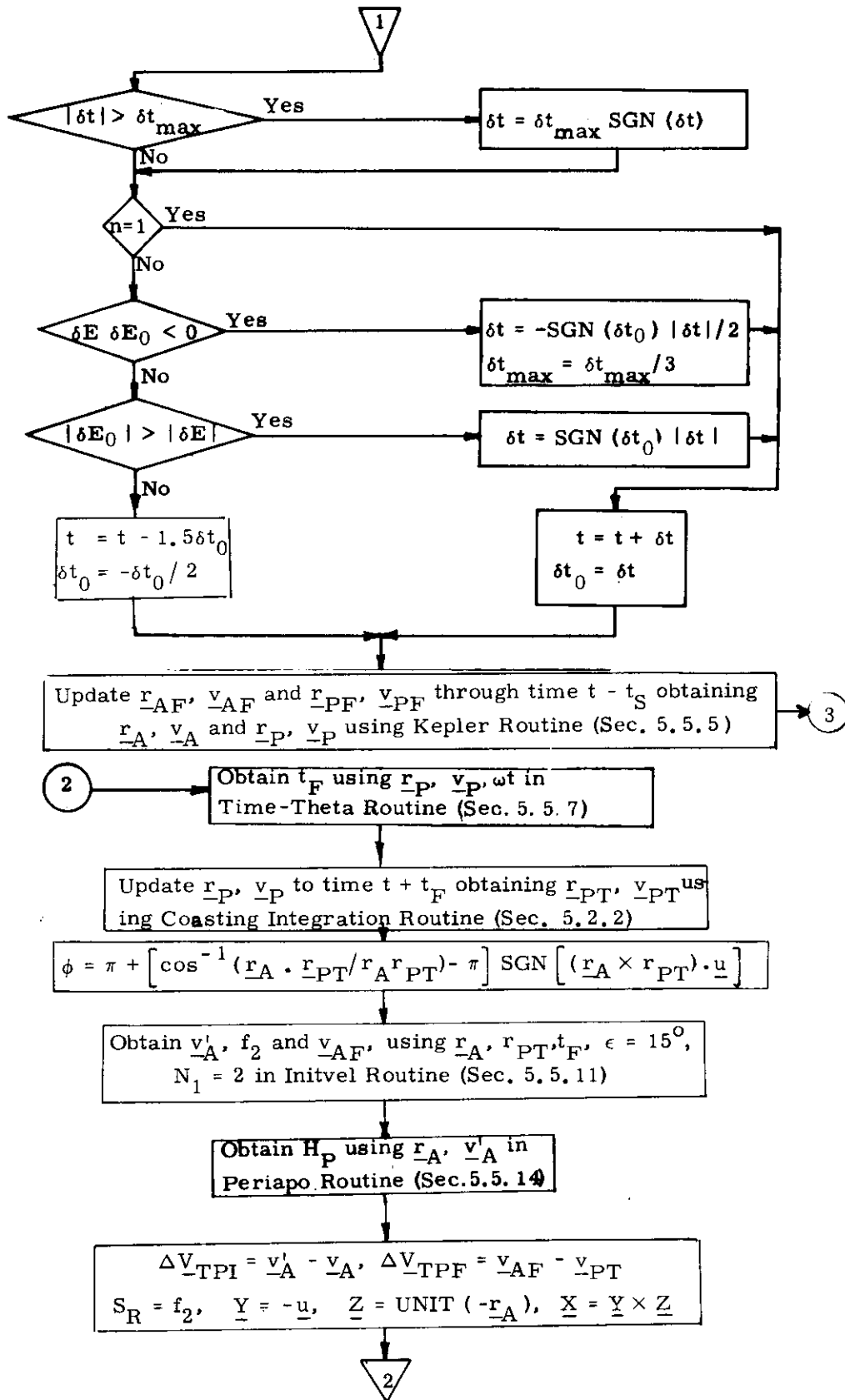


Figure 4.4-2 Pre-TPI Maneuver Program (page 2 of 3)

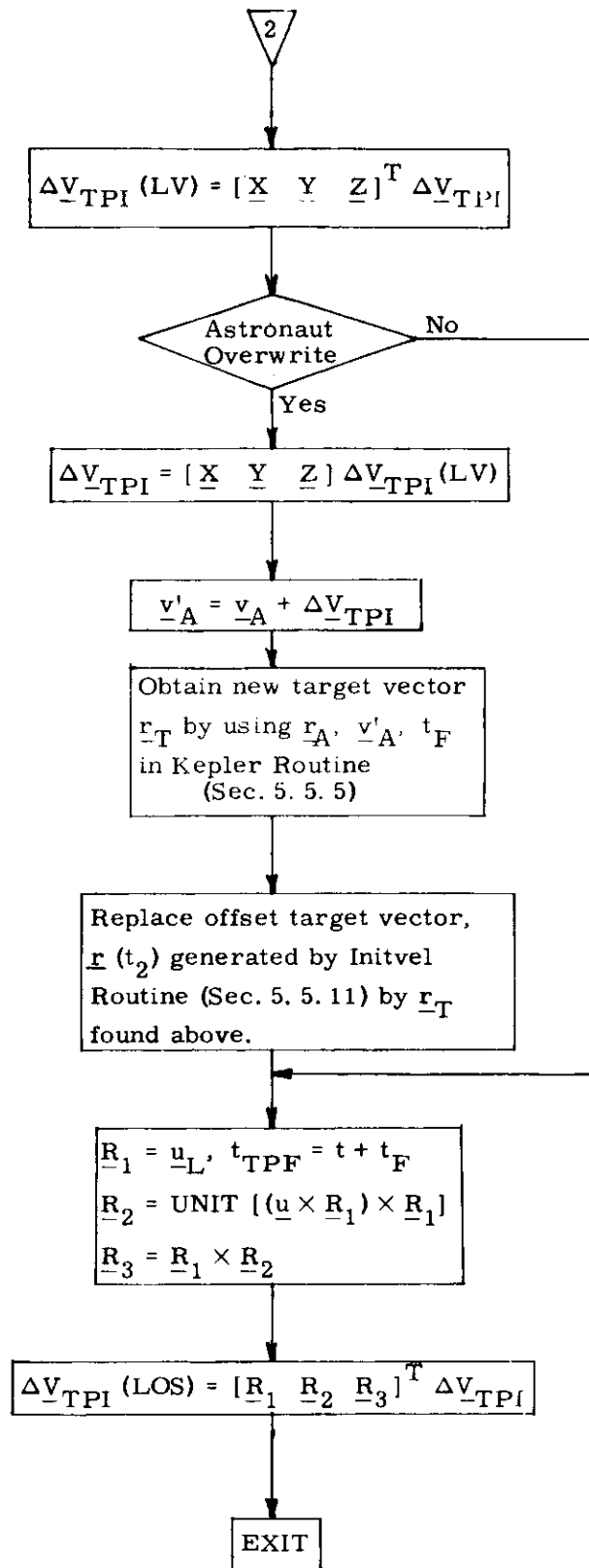


Figure 4. 4-2 Pre-TPI Maneuver Program (page 3 of 3)

The iteration is successful when the computed elevation angle  $E_A$  is sufficiently close to  $E$ .

To help insure convergence, the following steps are taken:

- a. The step size  $\delta t$  is restricted to 250 secs.
- b. If the solution has been passed ( $\delta E \delta E_0 < 0$ ), the step size is halved and forced in the opposite direction of the last step.
- c. If the iteration is converging ( $|\delta E_0| - |\delta E| > 0$ ), the sign of  $\delta t$  is maintained.
- d. If the iteration is proceeding in the wrong direction, the step direction is reversed.

The iteration is terminated for any of the following reasons, and a single alarm as described in Section 4 is set:

1. The iteration counter has exceeded its maximum value of 15.
2. The line of sight emanating from the active vehicle does not intersect the circular orbit with radius equal to that of the passive vehicle.
3. The elevation angle is inconsistent with the relative altitudes of the two vehicles (e. g., if the elevation angle is less than  $180^\circ$  when the active vehicle is above the passive vehicle).

Upon convergence, the state vectors are precision updated to the TPI time.



The TPI-TPF phase of the program starts with the use of the angle  $\omega t$  in the Time-Theta Subroutine (Section 5.5.7) to determine the corresponding transfer time  $t_F$ . After precision updating of the passive vehicle through  $t_F$ , the Initvel Routine (Section 5.5.11) is then called to compute a transfer trajectory between the TPI and TPF points with the switch  $N_1$  set equal to two and the cone angle  $\epsilon$  set equal to  $15^\circ$ . This results in the precision offsetting procedure to be used twice in succession. The rotation projection switch  $f_2$  is obtained from the steering program as  $S_R$ .

The central angle  $\phi$  traversed by the active vehicle from the TPI time to intercept is computed as shown in Fig. 4.4-2 for display if requested by the astronaut. This display is used to avoid  $180^\circ$  transfer angle problems.

The displays for the Pre-TPI program are:

1.  $\Delta V_{TPI}$
2.  $\Delta V_{TPF}$
3.  $\Delta \underline{V}_{TPI}(\text{LOS})$  : Line-of-sight coordinates
4.  $\Delta \underline{V}_{TPI}(\text{LV})$  : Local vertical coordinates
5. Perigee altitude ( $H_P$ ) following the TPI maneuver

The display of  $\Delta \underline{V}_{TPI}(\text{LV})$  may be overwritten by the astronaut. If so, a new target vector is generated as shown in Fig. 4.4-2. This new aimpoint is used for powered flight steering.

The cross product steering constant,  $ec$ , of Section 5.3.3.4 is set equal to unity in P-34, and is an input parameter to the thrust program.

In addition,  $M$ , the number of navigation measurements processed since the last maneuver,  $MGA$ , the middle gimbal angle at the start of the maneuver assuming plus X acceleration, and  $TFI$ , the time between the CSI maneuver and the present time are displayed. The middle gimbal angle is computed with the following equation

$$MGA = \sin^{-1} \left[ \underline{Y}_{REF} \cdot \text{UNIT} (\Delta \underline{V}) \right]$$

if  $-90^\circ \leq MGA < 0^\circ$ ,  $MGA$  is redefined such that  $270^\circ \leq MGA < 360^\circ$ .  $\Delta \underline{V}$  is the maneuver  $\Delta \underline{V}$  in reference coordinates and  $\underline{Y}_{REF}$  is the second row of the REFSMMAT matrix defined in Section 5.6.3.3.

#### 5.4.4.3 Rendezvous Midcourse Maneuver

This program, corresponding to Program P-35 (CSM active) or P-75 (LM active) of Section 4, computes a midcourse correction maneuver. This maneuver insures that the active vehicle will intercept the passive vehicle at the time established in the previous Pre-TPI program. The astronaut may call this program any time after the TPI maneuver, but in general no later than 10 minutes before the intercept time. The flow diagram for this program is shown in Fig. 4.4-3.

There is one astronaut input: choice of the active vehicle (P-35 CSM, P-75 LM). The active  $\underline{r}_A$ ,  $\underline{v}_A$  and passive  $\underline{r}_P$ ,  $\underline{v}_P$  vehicle state vectors, the intercept time  $t_{\text{TPI}}$  (from the Pre-TPI program) and a time delay are available in the guidance computer. The time delay ( $\delta\tau_3$  for P-35 and  $\delta\tau_7$  for P-75) is the time required to prepare for the thrust maneuver and is stored in either of two erasable locations.

When the program is initiated the number of navigation measurements since the last maneuver and the time since TPI are displayed. Based on this information and additional displays discussed in Section 5.6.7, the astronaut may elect to proceed with the midcourse maneuver at some time. When he does so, the program updates the state vectors to the present time plus the time delay using the Coasting Integration Routine. The Initvel Routine is then called to obtain the velocity vector on the transfer ellipse. The cone angle  $\epsilon$  is set equal to  $15^\circ$  and the  $N_1$  switch is set equal to two, resulting in a double use of the offsetting procedure. The rotation projection switch  $f_2$  is obtained from the Initvel Routine for use in the powered flight steering program as  $S_R$ .

The central angle  $\phi$  traversed by the active vehicle from the maneuver time to intercept is computed as shown in Fig. 4.4-3 for display if requested by the astronaut. This display is used to avoid  $180^\circ$  transfer angle problems.

After obtaining the maneuver  $\Delta\underline{V}$ , it is rotated into a local vertical (LV) and a line of sight (LOS) coordinate system obtaining  $\Delta\underline{V}(\text{LV})$  and  $\Delta\underline{V}(\text{LOS})$ , and both are displayed.

The display of  $\Delta\underline{V}(\text{LV})$  may be overwritten by the astronaut. If so, a new target vector is generated as shown in Fig. 4.4-3. This new aimpoint is used for powered flight steering.

In addition, M, MGA and TFI are displayed as in the Pre-TPI program displays.

The cross product steering constant,  $ec$ , is set equal to unity in P-35 as in P-34.

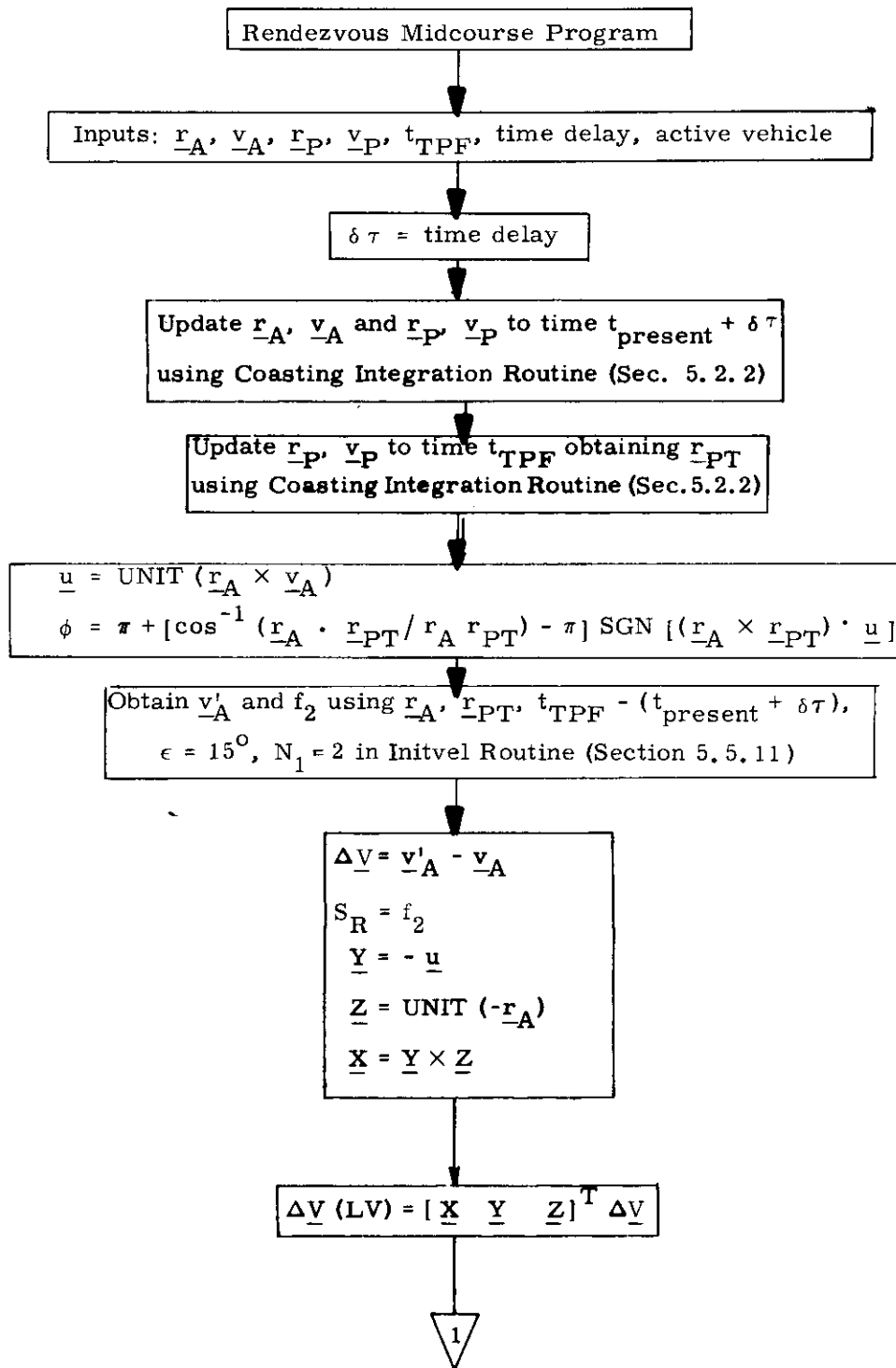


Figure 4.4-8 Rendezvous Midcourse Program  
(page 1 of 2)

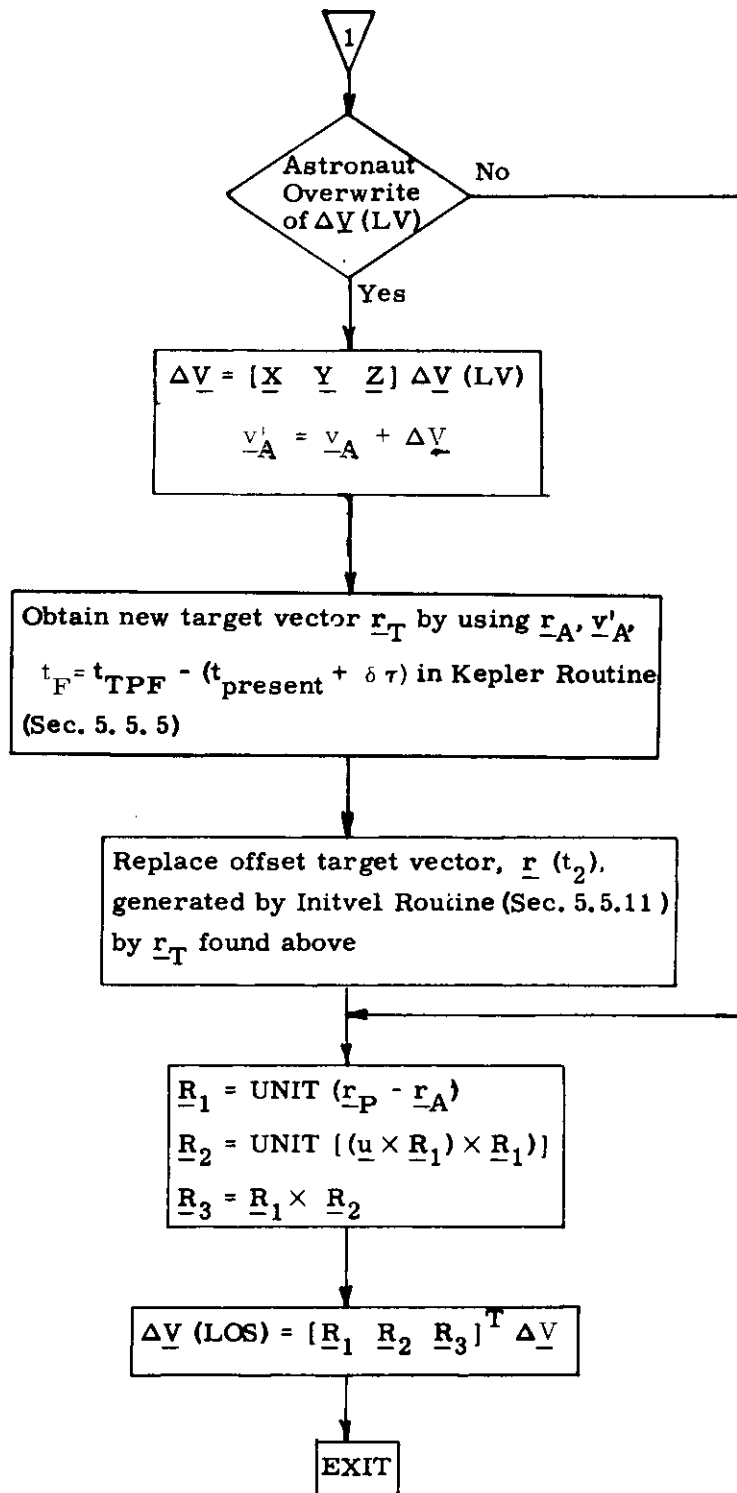


Figure 4.4-3 Rendezvous Midcourse Program  
(page 2 of 2)

#### 5.4.4.4 TPI Search Program

The objective of the TPI Search Program is to determine the transfer trajectory having a minimum total velocity change, within the constraint of a safe pericenter, for a specified TPI time. The minimum total velocity determined is the sum of the impulsive velocities for the TPI and TPF maneuvers. This program is used to establish the target parameters for a direct intercept trajectory between the LM and CSM orbits. The manual terminal rendezvous maneuver is normally initiated from such an intercept trajectory. The TPI Search Program is used to provide the required TPI maneuver parameters for either direct transfer rendezvous profiles or concentric flight plan rendezvous TPI maneuvers if  $\Delta V$  minimization is desired beyond nominal preselected astronaut input target parameters. This program can be used in both earth and lunar orbits and consists of two separate programs in Section 4. When used as program P-17, the active vehicle which establishes the intercept trajectory is the CSM. When used as P-77, the active vehicle is the LM.

The desired time  $t_{TPI}$  of the TPI maneuver is required as an input to this program. The basic output parameters displayed to the astronaut are:

- 1)  $\pm \phi$  Initial relative phase angle between the two vehicles at  $t_{TPI}$ , + indicating that the active vehicle is ahead of the passive vehicle with respect to orbital motion.

5.4-13

Revised COLOSSUS  
 Added GSOP #R-577 PCR # 628 Rev. 3 Date 11-6-68

- 2)  $\pm \Delta h$  Altitude difference between the two vehicles at  $t_{TPI}$ , + indicating that the passive vehicle has a larger radial magnitude than the active vehicle.
- 3) K Preferred central angle search sector. K = +1 for central angles greater than  $180^\circ$  and K = -1 for central angles less than  $180^\circ$ .
- The search option input defined in Section 4 has the following correspondence to the variable K:
- | search option | K  |
|---------------|----|
| 1             | -1 |
| 2             | +1 |
- 4)  $\Delta V_{TPI}$  Impulsive velocity required for the TPI maneuver.
- 5)  $\pm \Delta V_{TPF}$  Impulsive velocity required for the terminal rendezvous maneuver, + indicating that the active vehicle is ascending at TPF, - if descending.
- 6)  $h_P$  Altitude of the pericenter above a reference radius vector.
- 7)  $\omega t$  Central transfer angle of the passive vehicle.
- 8) P Pericenter code. P = 1 if pericenter occurs between the TPI and TPF maneuvers and P = 2 if pericenter occurs after the TPF maneuver.

5.4-14

Revised COLOSSUS

Added GSOP #R-577 PCR # 628 Rev. 3 Date 11-13-68

In general, non-coplanar transfer problems normally involve two minimum total velocity trajectories, one having a central angle greater than  $180^{\circ}$  and the other less than  $180^{\circ}$ . The program selects a sector (determined by the relative altitudes of the two vehicles at the TPI time) in which the most acceptable solution will probably be found. Upon displaying this sector to the astronaut, he may proceed with this choice or change the sector by changing the value of K.

Due to the excessive  $\Delta V$  requirements for non-coplanar two-impulse transfers near central angles of  $180^{\circ}$ , the search sectors are restricted from the  $180^{\circ}$  region by an angle which depends upon the magnitude of the angle between the active vehicle position vector at TPI and the passive vehicle orbital plane.

The general TPI Search Program target vectors and relative angles are illustrated in Fig. 4. 4-4 with the program logic outlined in Fig. 4. 4-5. The subscripts A and P refer to the active and passive vehicles, respectively. The minimum pericenter altitude  $h_L$  is defined as

Lunar Orbit:  $h_L = 35,000$  ft. about the landing site radius.

Earth Orbit:  $h_L = 85$  nm above the launch pad radius.

The following are the major computation sequences:

1. Upon activation of the program, the active and passive vehicle state vectors are advanced to the specified TPI time by means of the Kepler Routine. At this point the relative phase angle  $\phi$ , altitude difference  $\Delta h$ , and search sector flag K are computed and displayed to the astronaut and a response is requested. The astronaut may proceed, command the initial search to the other sector by changing K or change the TPI time.

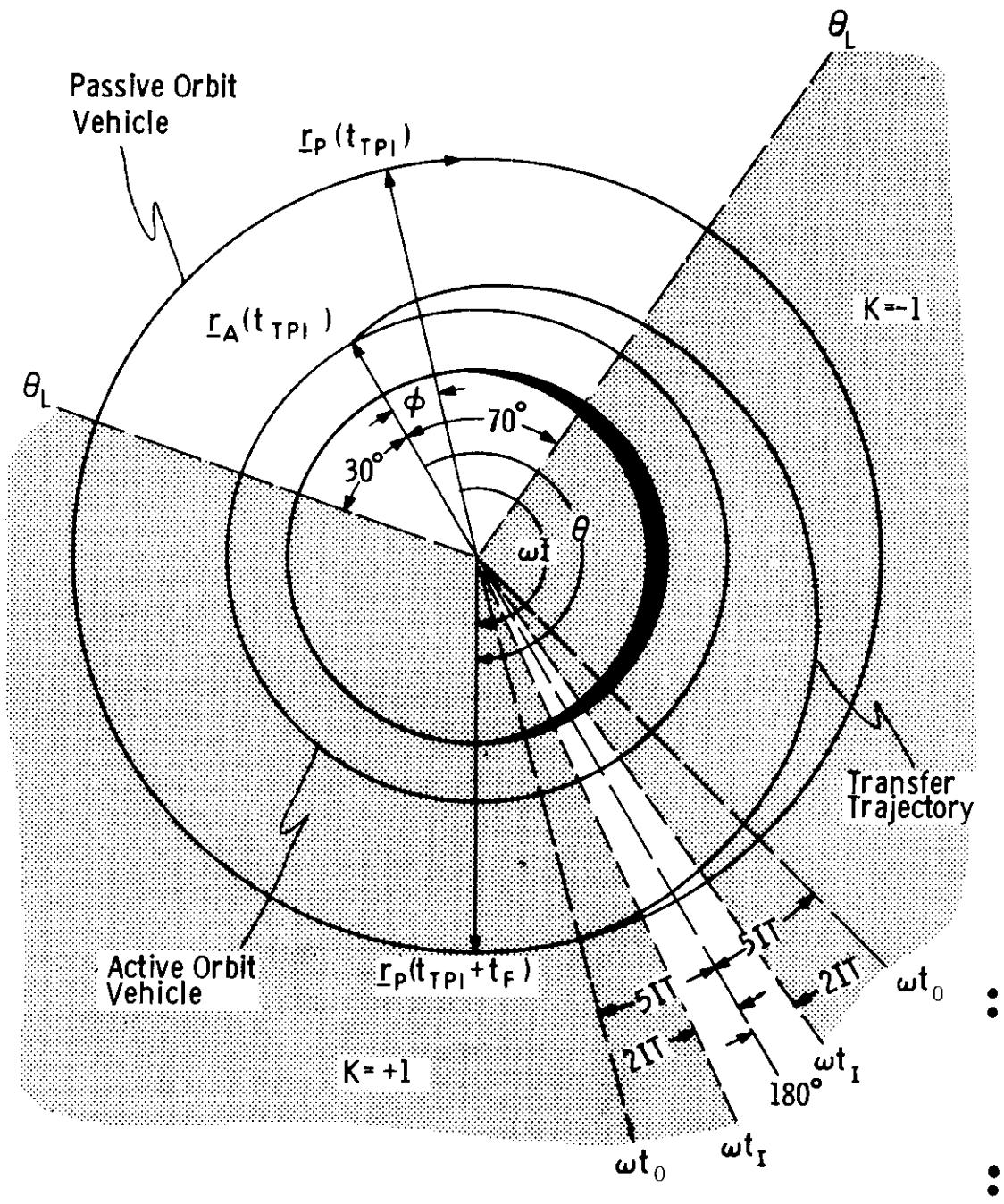


Figure 4. 4-4 Central Angle Sectors for the TPI Search Program



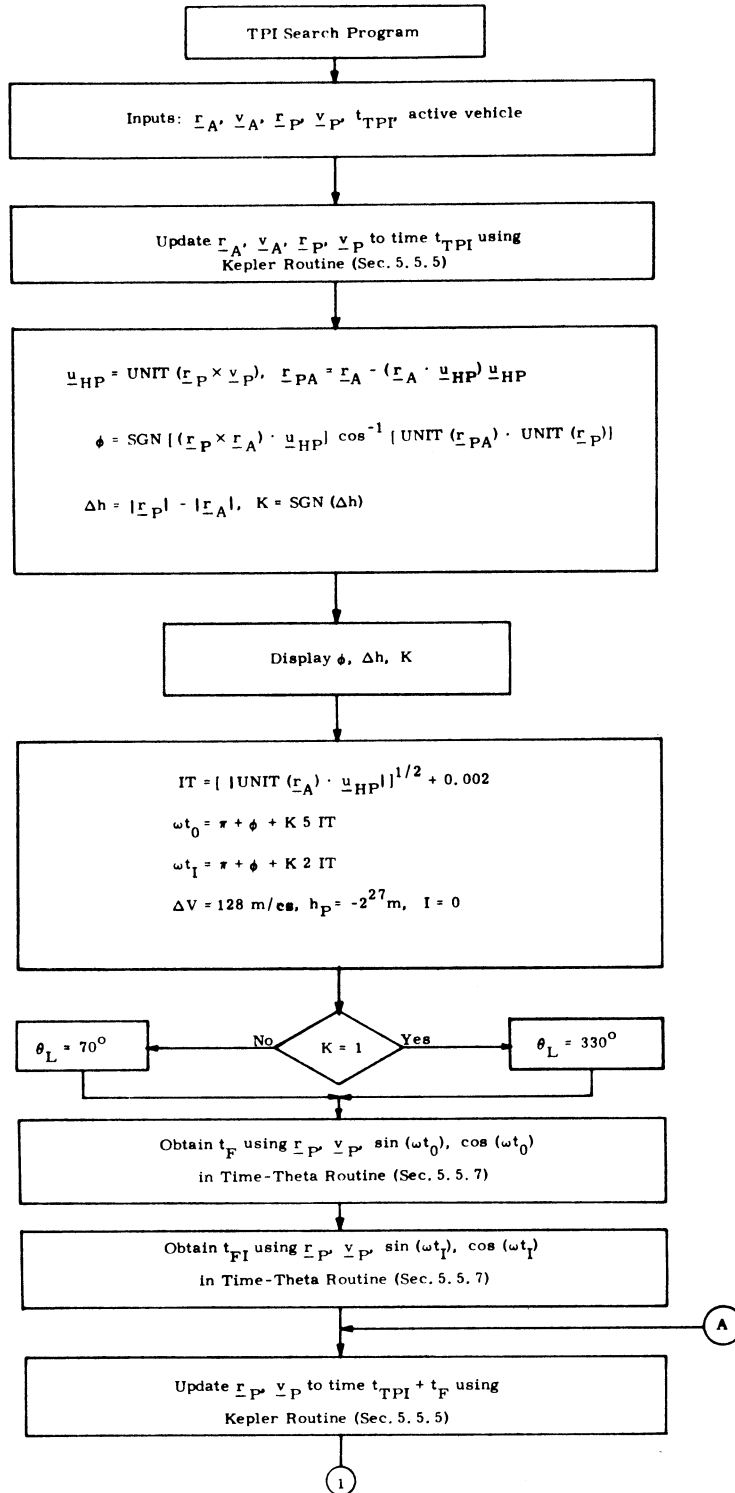


Figure 4.4-5 TPI Search Program Logic Diagram  
(page 1 of 2)

5.4-17

Revised COLOSSUS

Added GSOP #R-577 PCR # 628 Rev. 3 Date 11-6-68

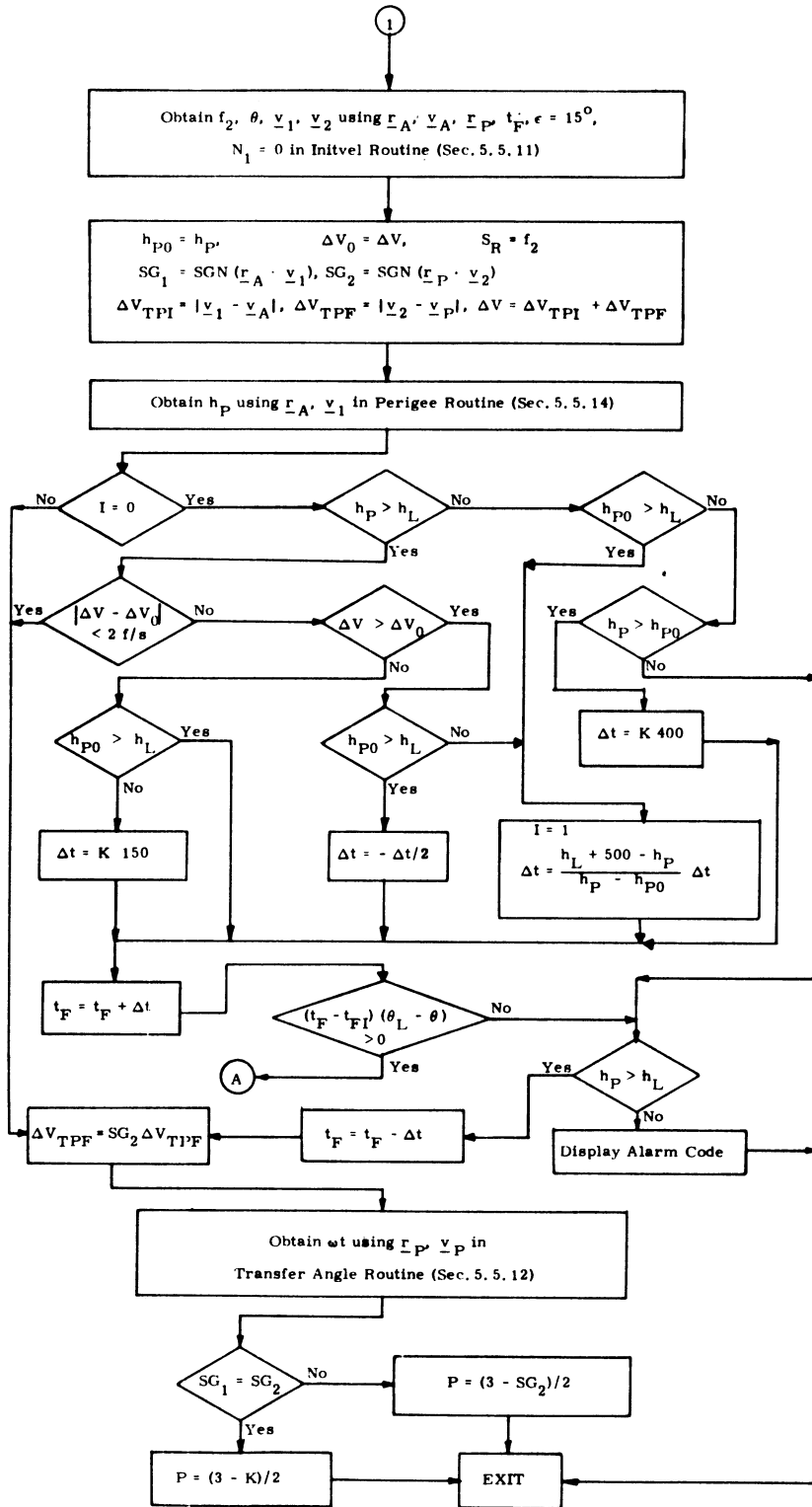


Figure 4.4-5 TPI Search Program Logic Diagram (page 2 of 2)

5.4-18

☑ Revised COLOSSUS

☐ Added GSOP #R-577 PCR # 628

Rev. 3

Date 11-6-68

2. An initial guess  $\omega t_0$  is next made of the transfer angle of the passive vehicle corresponding to a minimum  $\Delta V$  for the active vehicle which is used to start the search iteration. An inner transfer angle limit  $\omega t_I$  for the passive vehicle is set to avoid excessive  $\Delta V$  conditions, and the outer search limits  $\theta_L$  of the active vehicle are set as indicated in Fig. 4.4-5. Initial values are next chosen for  $\Delta V$ ,  $h_P$  and  $L$ . The Time Theta Routine is then used to determine the passive vehicle transfer times  $t_F$  and  $t_{FI}$  to traverse the angles  $\omega t_0$  and  $\omega t_I$  respectively.

3. The minimum  $\Delta V$  trajectory search begins by advancing the passive vehicle to the initial aim point at time  $t_{TPI}$  plus  $t_F$  by means of the Kepler Routine. The Initvel Routine is then called with  $N_1$  and the cone angle  $\epsilon$  equal to zero and  $15^\circ$  respectively. The active vehicle's velocity  $\underline{v}_1$  and  $\underline{v}_2$  after the TPI maneuver and before the TPF maneuver, respectively, along with the transfer angle  $\theta$  and the rotation projection switch  $f_2$  (used in the powered flight steering program as  $S_R$ ) are obtained from this routine. After calculating the maneuver magnitudes, the Periapo Routine is used to calculate the pericenter altitude  $h_P$ .

4. The program enters a control sequence whose effect is as follows:

1. If the height of pericenter  $h_P$  and its previous value  $h_{P0}$  are both less than  $h_L$  and furthermore  $h_P$  is less than  $h_{P0}$  an alarm is displayed indicating that there are no safe pericenters in the sector being searched.

2. If, as before, both  $h_P < h_L$  and  $h_{P0} < h_L$ , but now  $h_P > h_{P0}$  the program sets the time of flight increment  $\Delta t$  equal to  $K 400$  sec., increments the time of flight, and continues searching.

5.4-19

Revised COLOSSUS  
 Added GSOP #R-577 PCR # 628 Rev. 3 Date 11-6-68

3. If the pericenter changes from safe to unsafe, the program interpolates to find the time of flight at which the change occurred, and accepts this value as the solution, together with  $\Delta V_{TPI}$ ,  $\Delta V_{TPF}$ ,  $h_P$ .

4. If the pericenter changes from unsafe to safe, the program also interpolates and accepts the solution provided  $\Delta V$  is greater than its previous value  $\Delta V_0$ . If not it sets  $\Delta t = K150$  sec., increments the time of flight, and continues searching.

5. If the present pericenter is safe and  $|\Delta V - \Delta V_0| < 2$  ft/sec. the present time of flight is accepted as the solution.

6. If none of the above conditions holds, the time of flight is incremented by  $\Delta t$  if  $\Delta V < \Delta V_0$  or diminished by  $\Delta t/2$  if  $\Delta V > \Delta V_0$ , and the search function continues.

7. If the incremented time of flight  $t_F$  exceeds  $t_{FI}$  or the active vehicle transfer angle  $\theta$  exceeds  $\theta_L$ , the last solution is accepted if a safe pericenter exists, otherwise an alarm code is displayed, indicating no solution in this sector.

5. A sign ( $SG_2$ ) is affixed to  $\Delta V_{TPF}$  to indicate whether the active vehicle is ascending or descending on the passive vehicle at TPF and the solution set consisting of  $\Delta V_{TPI} \pm \Delta V_{TPF}$ ,  $h_P$  is displayed to the astronaut. If the astronaut is satisfied with these trajectory parameters, he may request that the central transfer angle  $\omega t$  of the passive vehicle be displayed to be recorded by him as an input parameter for the TPI Pre-thrust program P-34 along with the initial  $t_{TPI}$ . In addition, this display will also contain the pericenter code P which indicates whether the pericenter of the transfer trajectory occurs between the TPI and TPF maneuvers or beyond the TPF maneuver. The other central angle sector may be searched by returning to 1 and changing K, or a new trajectory solution attempted by returning to 1 and changing the input TPI time until a satisfactory transfer trajectory has been found.

5. 4-20

Revised COLOSSUS

Added GSOP # R-577 PCR # 628 Rev. 3 Date 11-13 68

THIS PAGE INTENTIONALLY LEFT BLANK

5. 4-21

Revised COLOSSUS

Added GSOP #R-577 PCR # 628 Rev. 3 Date 11-13 68

THIS PAGE INTENTIONALLY LEFT BLANK

5.4-22

Revised COLOSSUS

Added GSOP # R-577 PCR # 628 Rev. 3 Date 11-13-68

#### 5. 4. 4. 5 Stable Orbit Rendezvous

The Stable Orbit Rendezvous Program P-38 (CSM active) or P-78 (LM active) is defined by the following three maneuvers. The first maneuver (Stable Orbit Initiation SOI) defines a transfer of the active vehicle to a point on the orbit of the passive vehicle, a given distance ahead of, or behind the passive vehicle, as measured along the passive vehicle orbital path. The second maneuver (Stable Orbit Rendezvous SOR) is designed to place the active vehicle in the same orbit as the passive vehicle with a separation between the two vehicles determined by the first maneuver. The position of the active vehicle relative to the passive vehicle after this second maneuver is referred to as the stable orbit point. The third maneuver defines the final transfer from the stable orbit point to the terminal rendezvous point and is calculated using the pre-TPI maneuver program (Section 5.4.4.2). All the maneuvers are based on precision trajectory computations.

After the SOI maneuver, the Rendezvous Midcourse Maneuver Program (Section 5.4.4.5.3) is available for computing midcourse corrections.

The pre-thrust programs for the Stable Orbit Rendezvous (P-38, P-78, P-39, P-79) use the Lambert targeting concept employing two precision off-sets (Section 5.5.6) and the cross-product steering of Section 5.3.3. The cross product steering constant,  $c$ , of Section 5.3.3.4 is set equal to unity in programs P-38 and P-39.

The Stable Orbit Rendezvous Program has two phases, which must be called sequentially to compute the first and second rendezvous maneuvers. The input parameters are:

- 1) Choice of active vehicle (P-38 CSM, P-78 LM).
- 2) Ignition time of maneuver  $t_{IG}$ .
- 3)  $\omega t$ , central angle of travel during the transfer, based on the passive vehicle orbit.
- 4) S (0 for the first maneuver, 1 for the second maneuver).
- 5) Stable orbit displacement distance  $\delta r$ . Positive  $\delta r$  indicates that the stable orbit point is behind the passive vehicle. (Specified only when S = 0).

The active vehicle state vector  $\underline{r}_A$ ,  $\underline{v}_A$  and passive vehicle state vector  $\underline{r}_p$ ,  $\underline{v}_p$  are available in the guidance computer.

The trajectory configuration for the first two maneuvers is illustrated in Fig. 4. 4-6 and the program logic is outlined in Fig. 4. 4-7. Both phases of the program start with the update of the state vectors to the ignition time and the conversion of  $\omega t$  to the transfer time  $t_F$ .

#### 5.4.4.5.1 Stable Orbit Initiation Maneuver

If S is set equal to zero, the program computes the stable orbit initiation SOI maneuver. The passive vehicle is first updated to the intercept time  $t_I$ , equal to  $t_{IG}$  plus  $t_F$ . This point is next used to convert the stable orbit displacement distance  $\delta r$  to a time  $\delta t$  by approximating the orbit of the passive vehicle with a circular orbit and by using an orbital velocity equal to the vehicle's horizontal velocity. The passive vehicle is now updated to the target time  $t_T$ , equal to  $t_I$  minus  $\delta t$ , to obtain the target vector  $\underline{r}_T$  for the first maneuver (see Fig. 4. 4-7.)

5.4-24

Revised COLOSSUS

Added GSOP # R-577 PCR # 628 Rev. 3 Date 11-6-68



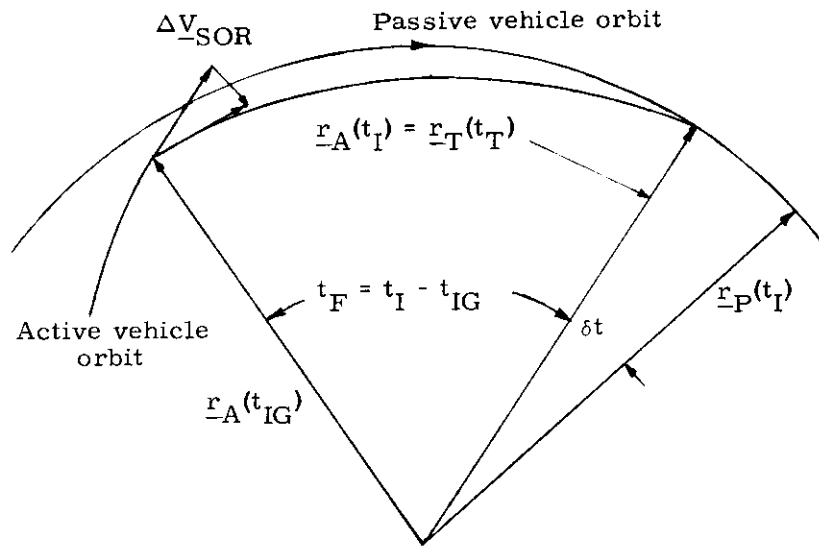
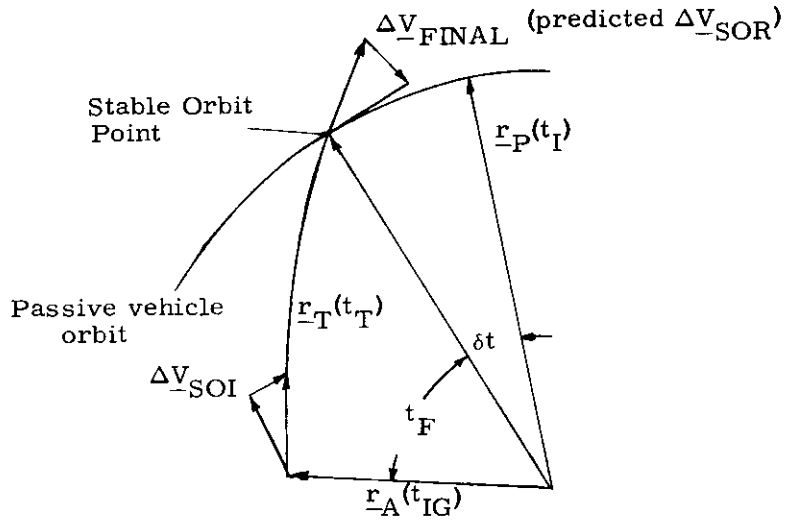


Figure 4.4-6 Geometry of the SOI and SOR Maneuvers

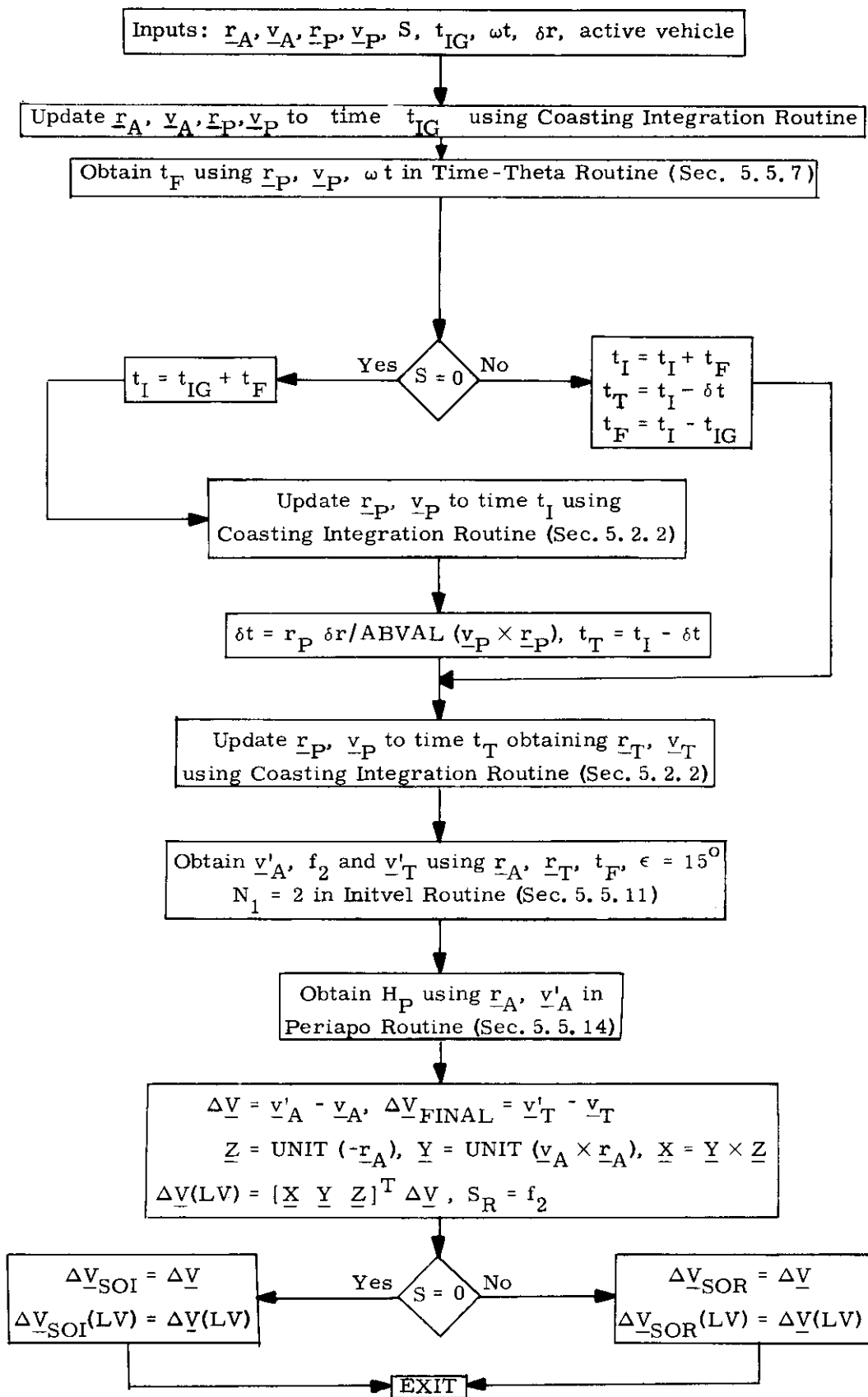


Fig. 4.4-7 Stable Orbit Rendezvous Program

5. 4-26

Rev. 3 - 11/68

The Initvel Routine is called with  $N_1$  and the cone angle  $\epsilon$  equal to two and  $15^\circ$  respectively. This results in the precision offsetting procedure to be used twice in succession. The rotation projection switch  $f_2$  is obtained from the Initvel Routine for use in the powered flight steering program as  $S_R$ .

The following displays are computed.

- 1)  $\Delta V_{SOI}$  :  $\Delta V$  for the SOI maneuver.
- 2)  $\Delta V_{FINAL}$  : Passive vehicle's velocity minus the active vehicle's velocity, both computed at the stable orbit point.
- 3)  $\Delta V_{SOI}(LV)$  : Local vertical coordinates
- 4)  $H_p$  : Perigee altitude following the stable orbit initiation maneuver.
- 5)  $t_I$  : Time the active vehicle intercepts the stable orbit point.

#### 5.4.4.5.2 Stable Orbit Rendezvous Maneuver

If  $S$  is set equal to one, the program computes the stable orbit rendezvous SOR maneuver. In order for this maneuver to place the active vehicle in the orbit of the passive vehicle in a precise manner, the SOR ignition time should be offset to account for the finite burn time. When the transfer trajectory time  $t_F$  is added to the calculated SOI intercept time  $t_I$  (stored in the SOI maneuver program), the SOR intercept time  $t_I$  is obtained (see Fig. 4.4-7).

The target vector  $\underline{r}_T$  for the SOR maneuver can be obtained by updating the passive vehicle to the target time  $t_T$ , equal to  $t_I$  minus  $\delta t$  ( $\delta t$  is stored in the SOI maneuver program). The transfer time  $t_F$ , based on the offset ignition time, is equal to the new intercept time minus the ignition time.

After using the Initvel Routine, as in the Stable Orbit rendezvous maneuver, to compute the transfer trajectory, the following displays are calculated:

- 1)  $\Delta V_{\text{SOR}}$  :  $\Delta V$  for the SOR maneuver
- 2)  $\Delta V_{\text{FINAL}}$  : Approximately equal to zero
- 3)  $\Delta V_{\text{SOR}}(\text{LV})$  : Local vertical coordinates
- 4)  $H_p$  : Perigee altitude following the SOR maneuver

#### 5.4.4.5.3 Stable Orbit Midcourse Maneuver

This program, corresponding to Program P-39 (CSM active) or P-79 (LM active) of Section 4, computes a midcourse correction maneuver. The astronaut may call this program after completing the SOI maneuver, but in general no later than 10 minutes before the intercept time. The program logic is outlined in Fig. 4.4-8.

There is one astronaut input: choice of the active vehicle (P-39 CSM, P-79 LM). The active vehicle state vector  $\underline{r}_A, \underline{v}_A$ , the passive vehicle state vector  $\underline{r}_P, \underline{v}_P$ , the time delay the target time  $t_T$  and the intercept time  $t_I$  are available in the guidance computer. The time delay ( $\delta \tau_3$  for P-39 and  $\delta \tau_7$  for P-79) is the time required to prepare for the thrust maneuver and is stored in either of two erasable locations.

5.4-28

Revised COLOSSUS  
 Added GSOP # R-577 PCR # 628 Rev. 3 Date 11-6-68

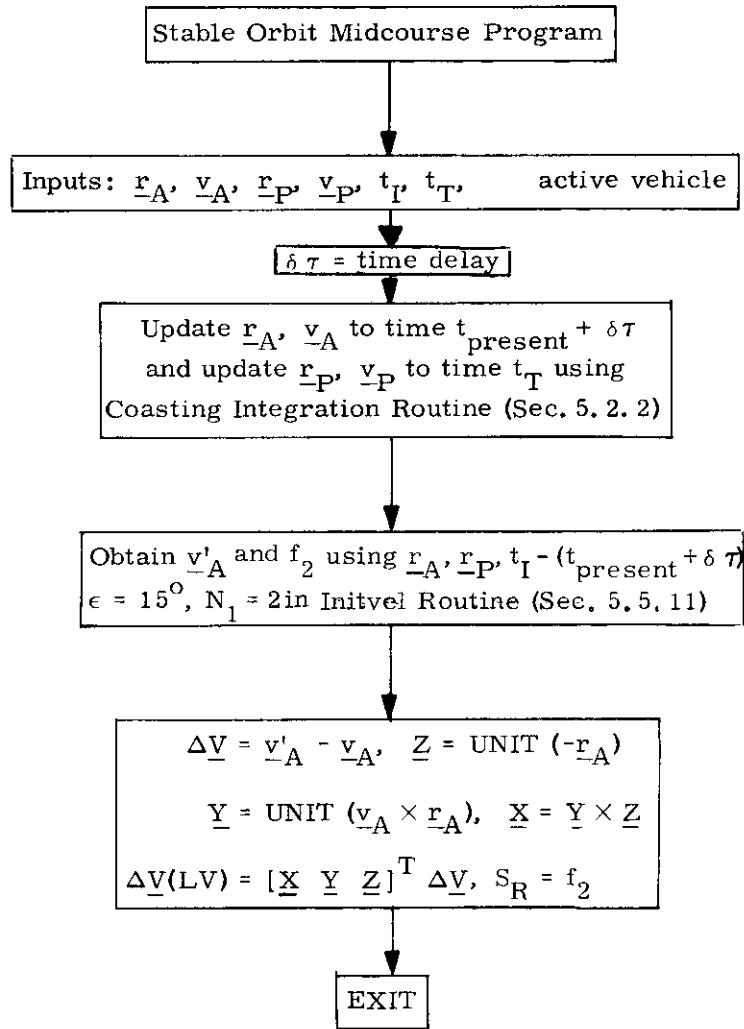


Fig. 4.4-8 Stable Orbit Midcourse Program

When the program is initiated the number of navigation measurements processed since the last maneuver and the time to ignition are displayed. Based on this information and additional displays discussed in Section 5.6.7, the astronaut may elect to proceed with the midcourse maneuver at some time. When he does so, the program updates the active vehicle to the present time plus the time delay  $\delta \tau$  and updates the passive vehicle to the target time using the Coasting Integration Routine. The Initvel Routine is then called with a transfer time equal to the intercept time minus the ignition time to obtain the initial velocity  $v'_A$  for the transfer ellipse. The  $N_1$  switch is set equal to two, resulting in a double use of the offsetting procedure. After obtaining the maneuver  $\Delta V$ , it is rotated into a local vertical coordinate system, resulting in the display  $\Delta V(LV)$ . In addition, M, MGA and TFI are displayed as in the Pre-TPI program displays.

#### 5. 4. 5 RETURN TO EARTH TARGETING

##### 5. 4. 5. 1 Introduction

The Return to Earth Targeting Program provides to the astronaut the ability to compute on board the spacecraft the targeting for a powered maneuver which returns the spacecraft to an earth reentry point with proper conditions for a safe reentry. This program is intended as a backup in the event of a communication or ground system failure and, thus, is completely independent of inputs from the ground system. The method of computation however, does not compromise certain critical trajectory characteristics such as fuel usage and transit time for the expediency of minimum computer storage usage or computational speed.

Specifically this program provides powered maneuver targeting for :

- (1) Returning from near earth orbits
- (2) Returning from trajectories resulting from a translunar injection powered maneuver failure.
- \* (3) Returning from translunar midcourse

---

\* Outside moon's sphere of influence only

- \* (4) Returning from transearth midcourse
- \* (5) Midcourse corrections during transearth midcourse
- \* (6) Changes in transit time during transearth midcourse

Two modes of operation are provided: time critical and fuel critical. The former establishes trajectories which return to earth in a minimum of time, the latter, with a minimum of fuel.

The return trajectory is constrained to be in the plane which the spacecraft was in prior to the return so that optimum fuel utilization will be achieved.

Although explicit control of the spacecraft landing site is not a feature of this program, it is possible to achieve landing site selection through adjustment of the astronaut input to the program. When on translunar or transearth trajectories, the adjusted parameter is the desired fuel utilization. When in orbit near the earth, the designated return time is the adjusted parameter. The astronaut need only to note the earth track of the landing site as a function of the adjusted parameter to select the proper parameter value.

The computations take place in two steps. The first is the generation of a conic two body solution to the problem. This takes a relatively short computational time. The second step, the computation of a precise trajectory, takes a relatively long computational time.



The Return to Earth Program is described in increasing detail in the following subsections:

Sec. 5. 4. 5. 2	Program Input-Output	pg. 5. 4-34
Sec. 5. 4. 5. 3	General Computational Procedure	pg. 5. 4-36
Sec. 5. 4. 5. 4	Logical Description	pg. 5. 4-46
Sec. 5. 4. 5. 5	Detailed Flow Diagrams	pg. 5. 4-59

#### 5.4.5.2 Program Input-Output

The Return to Earth Program may be called by the astronaut at any time the computer is operable. Should the designated time of return place the spacecraft at a point in space where the program is incapable of producing a successful solution, an alarm will occur. The current calling designation for this program is P-37 for both modes.

#### Return-to-Earth Targeting

After calling the program, the astronaut must load the following information into erasable memory via the keyboard.

- (1) objective velocity change to be utilized during the maneuver
- (2) time the maneuver is to be executed
- (3) reentry angle (normally set to zero and internally calculated\*)

Should a minimum fuel return be desired, the objective velocity change should be set to zero. Should the objective change be less than the minimum required, the program will select the minimum. When the objective velocity change is greater than the minimum, the return will require that amount of velocity change.

This program is not designed to operate from low altitude polar orbits unless the astronaut selects the reentry angle.

The program provides the following displays when both the conic trajectory computation and the precision trajectory computation are complete.

- (1) latitude of the landing site
- (2) longitude of the landing site
- (3) velocity change required (vector)

---

\*See 5.4.5.4.2 for detailed explanation of its use

- (4) velocity magnitude at 400,000 ft. entry altitude measured above the Fischer ellipsoid.
- (5) flight path angle at 400,000 ft. entry altitude
- (6) transit time to 400,000 ft. entry altitude from ignition.

The astronaut must select either the SPS or RCS engine after the precision trajectory calculation. The following quantities are displayed after the engine selection:

- (7) middle gimbal angle at ignition
- (8) time of ignition
- (9) time from ignition

The precision trajectory displays can be significantly different from the conic displays, especially in the case of long transit time returns; therefore the astronaut should check the new displays before accepting the precise solution.



The cross product steering constant,  $c$ , of Section 5.3.3.4 is set equal to 0.5 in this program. In addition the rotation switch,  $S_R$ , which is used in the Lambert Aim Point pre-thrust computation of Section 5.3.3.3.2 is set to zero.

### 5.4.5.3 General Computational Procedure

This section contains a general description of the Return-to-Earth computational procedure as illustrated in Figs. 4.5-1, 4.5-2, and 4.5-3.

The Return to Earth Targeting Program, as previously mentioned, may be used in either the fuel critical or time critical mode. When used in the fuel critical mode, the program generates a trajectory which meets specified constraints on reentry flight path angle and reentry altitude, and minimizes the impulsive velocity change required to achieve this trajectory. In the time critical mode, the program generates a trajectory which meets the specified reentry constraints and returns the spacecraft in the shortest possible time. The astronaut selects either mode by entering via the keyboard the objective velocity change ( $\Delta v_D$ ). If  $\Delta v_D$  is set to zero, the routine will provide a fuel critical trajectory. If  $\Delta v_D$  is not set to zero, the routine will provide a time critical trajectory if possible, or select a fuel critical trajectory if  $\Delta v_D$  is less than the minimum required impulsive velocity change.

The Return to Earth Targeting Program has two major phases, an initial two body conic phase illustrated in Fig. 4.5-1 and a numerically integrated precision trajectory phase illustrated in Fig. 4.5-2. A conic routine which provides initial conditions for return trajectories for both phases is illustrated in Fig. 4.5-3.

### 5.4.5.3.1 Nomenclature

The following is a list of the nomenclature used in Figs. 4.5-1, 4.5-2, and 4.5-3

$\underline{y}(t_0) = (\underline{x}(t_0), \underline{v}(t_0))$	CSM state vector maintained in the CMC
$\underline{y}_1(t_1) = (\underline{x}(t_1), \underline{v}_1(t_1))$	pre-return state vector
$\underline{y}_2(t_1) = (\underline{x}(t_1), \underline{v}_2(t_1))$	post-return state vector
$\underline{y}(t_2) = (\underline{x}(t_2), \underline{v}(t_2))$	state vector at reentry
$\underline{y}(t_2)_{\text{PRE}}$	final state vector of a precision trajectory
$R_{\text{CON}}$	final radius of a conic trajectory
$R_{\text{PRE}}$	final radius of a precision trajectory
$R_{\text{D}}$	desired final radius (reentry radius)
$\gamma(t_2)_{\text{CON}}$	final flight path angle of a conic trajectory
$\gamma(t_2)_{\text{D}}$	desired final flight path angle
$\gamma(t_2)_{\text{PRE}}$	final flight path angle of a precision trajectory
$x(t_1)$	cotangent of the post-return flight path angle
$\Delta v$	impulsive velocity change
$\Delta v_{\text{D}}$	desired velocity change
$\epsilon_1$ to $\epsilon_{10}$	tolerances

$t_{21}$	time of flight ( $t_2 - t_1$ )
$n_1$	counter
$f_1$	flag set to $\left\{ \begin{array}{l} 0 \text{ initial conic phase} \\ 1 \text{ precision trajectory phase} \end{array} \right.$

#### 5.4.5.3.2 Discussion of the Conic Phase

The following discussion is intended to supplement the information contained in Fig. 4.5-1.

The pre-return CSM state vector ( $\underline{y}_1(t_1)$ ) is obtained by numerical integration. An initial conic trajectory is then generated which meets the reentry and velocity change constraints of the problem. The reentry radius and flight path angle constraints, however, are known only as a function of the latitude and velocity magnitude at reentry. Therefore an iterative procedure is employed to generate this initial conic trajectory. Based on the pre-return radius magnitude, an estimate of the reentry constraints is made. This initial estimate is used to compute a conic trajectory which satisfies the  $\Delta v_D$  requirement. The final velocity magnitude and latitude are now available for use in a more precise calculation of the reentry constraints. An iterative process has now been started which quickly converges on a suitable initial conic trajectory.

During the above iterative process slight changes in the reentry flight path angle can produce significant changes in the minimum required velocity change. If the time critical mode is being used, it is possible that during a particular pass through the iteration loop the velocity change desired may be less than the minimum required. This does not necessarily preclude a time critical solution, and therefore the iteration is continued using the minimum required velocity change.

A description, in general terms, of the procedure CONICRETURN, used to obtain the initial state vector of the conic return trajectory, will follow later in this section.

After an initial conic trajectory is computed, an estimate is made of the landing site, and the results are displayed. Based on this display the astronaut may elect to continue with the calculation of a numerically integrated trajectory.

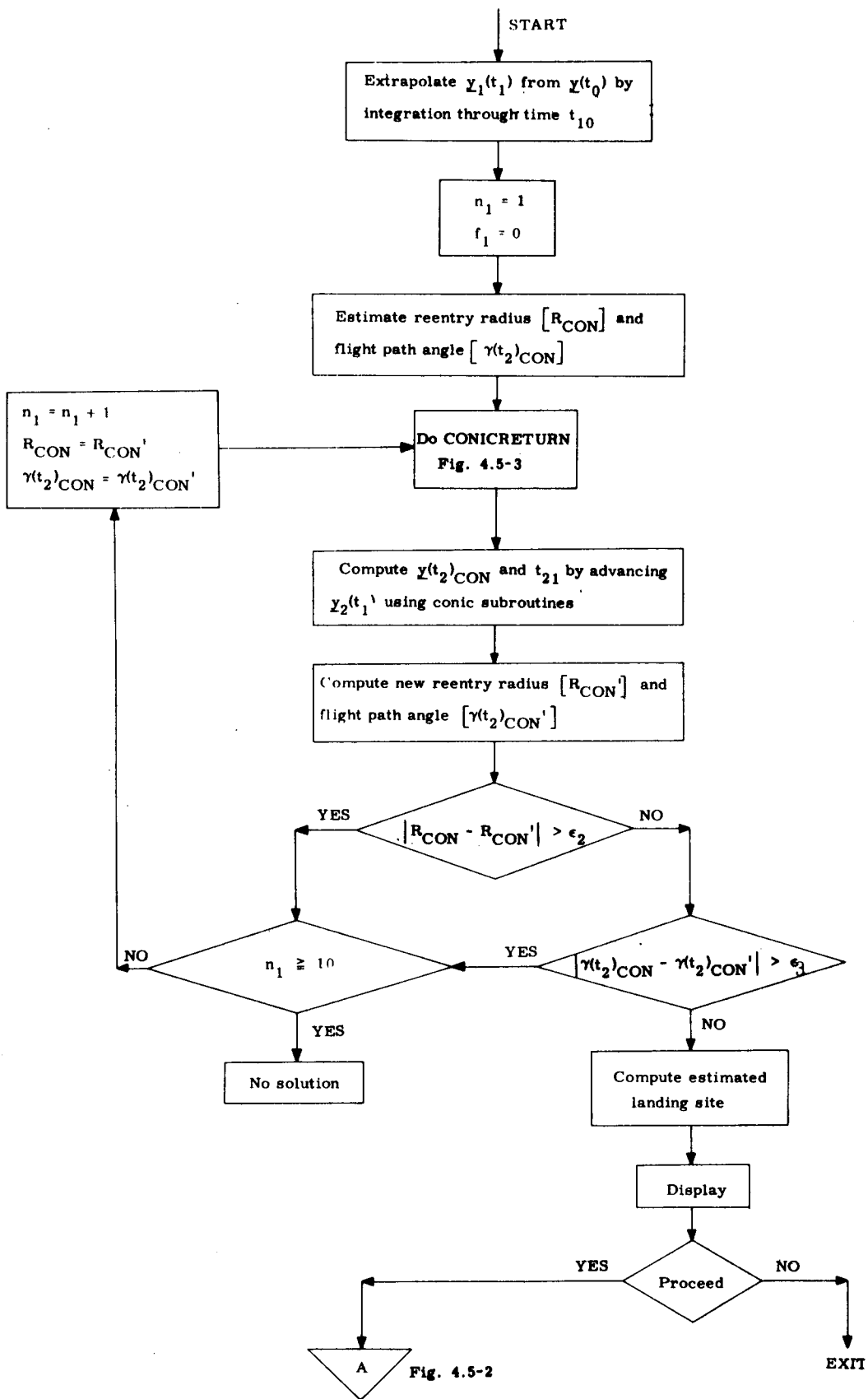


FIGURE 4.5-1 SIMPLIFIED LOGIC DIAGRAM OF RETURN TO EARTH TARGETING



#### 5.4.5.3.3 Discussion of the Precision Phase

The following discussion is intended to supplement the information contained in Fig. 4.5-2.

The numerically integrated trajectory must satisfy the reentry constraints achieved by the initial conic trajectory and meet the  $\Delta v_D$  requirement as closely as possible. An iterative procedure is again employed to generate this trajectory. The post-return state vector of the previous conic trajectory is used to initialize the Coasting Integration Routine. The integration proceeds until the desired reentry flight path angle has been achieved. The reentry radius of the next conic trajectory is then offset to compensate for the error in the reentry radius of the numerically integrated trajectory. An iterative process has now been initiated which converges to the precise trajectory which meets the objective velocity change and the reentry constraints. Next a target is computed for the Lambert Aimpoint Powered Flight Routine, the landing site is calculated, and the results are displayed.

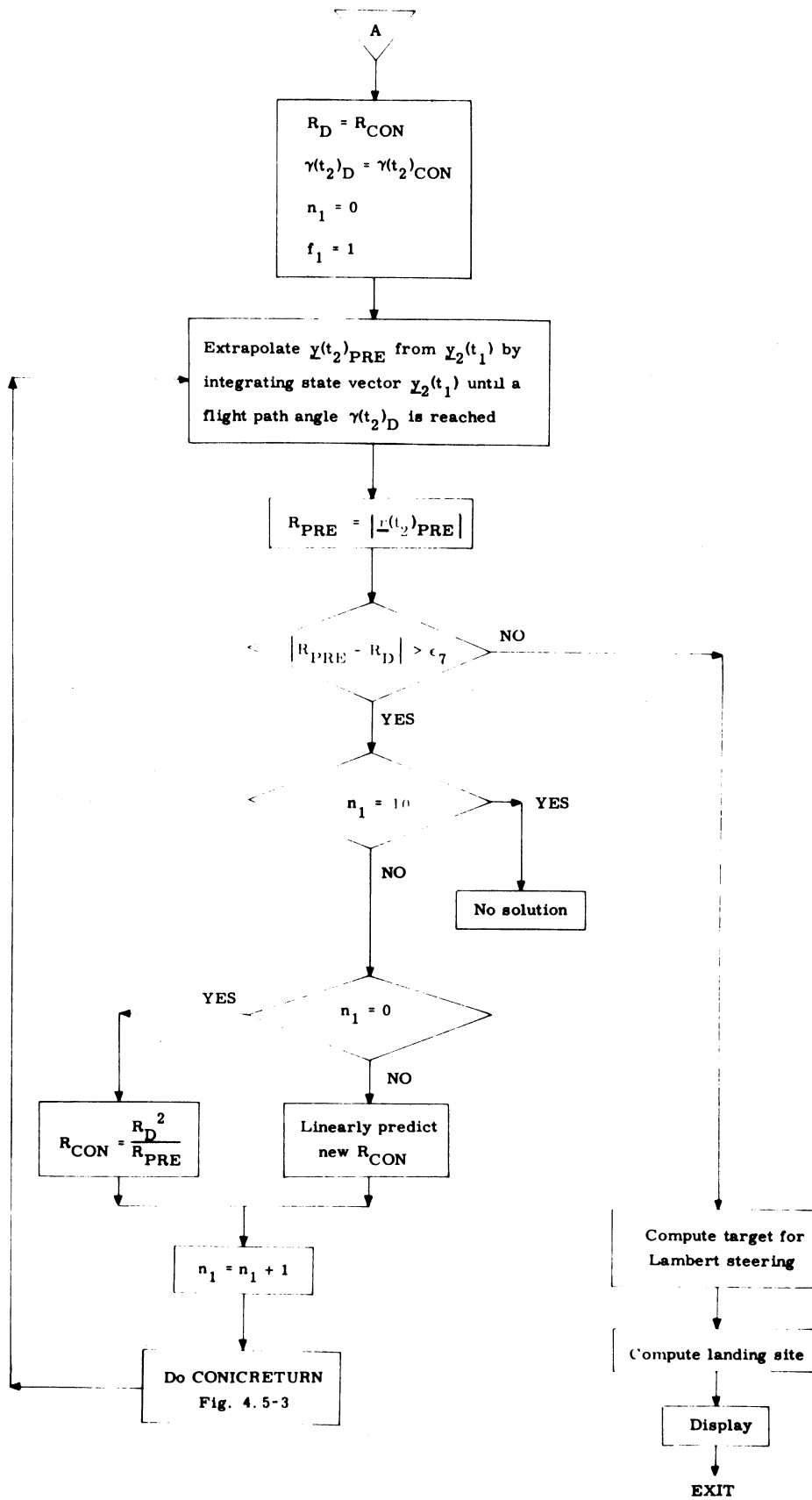


FIGURE 4.5-2 SIMPLIFIED LOGIC DIAGRAM OF RETURN TO EARTH TARGETING

#### 5.4.5.3.4 Discussion of the Conic Routine

The following discussion is intended to supplement the information contained in Fig. 4.5-3.

The conic routine solves the following two conic problems.

- (1) Given an initial position vector ( $\underline{r}(t_1)$ ), a pre-return velocity vector ( $\underline{v}_1(t_1)$ ), a final flight path angle ( $\gamma(t_2)_{CON}$ ), and a final radius magnitude ( $R_{CON}$ ), compute a post-return velocity vector ( $\underline{v}_2(t_1)$ ) which minimizes the velocity change ( $\Delta v$ ).
- (2) Given the above quantities, compute a post-return velocity vector ( $\underline{v}_2(t_1)$ ) which meets the objective velocity change ( $\Delta v_D$ ).

An iterative procedure is again used. The independent variable of the procedure is the cotangent of the post-return flight path angle ( $x(t_1)$ ), and the dependent variable is the velocity change ( $\Delta v$ ).

Initial bounds on  $x(t_1)$  are computed based on the maximum allowable semi-major axes. The upper bound constrains return trajectories which pass through apogee from going too near the moon, and the lower bound keeps the reentry velocity for direct return flights from exceeding the maximum allowable reentry velocity.

During the second phase of the Return to Earth Targeting Program, the calculation of a numerically integrated trajectory, additional bounds are imposed on the independent variable  $x(t_1)$ . In the case of a return from earth orbit, the resulting trajectory may be nearly circular, and will pass through the desired reentry flight path angle twice at nearly the same radius. Additional limits on  $x(t_1)$  are computed to prevent conic trajectories generated during the search for a numerically integrated trajectory from oscillating between these two solutions.

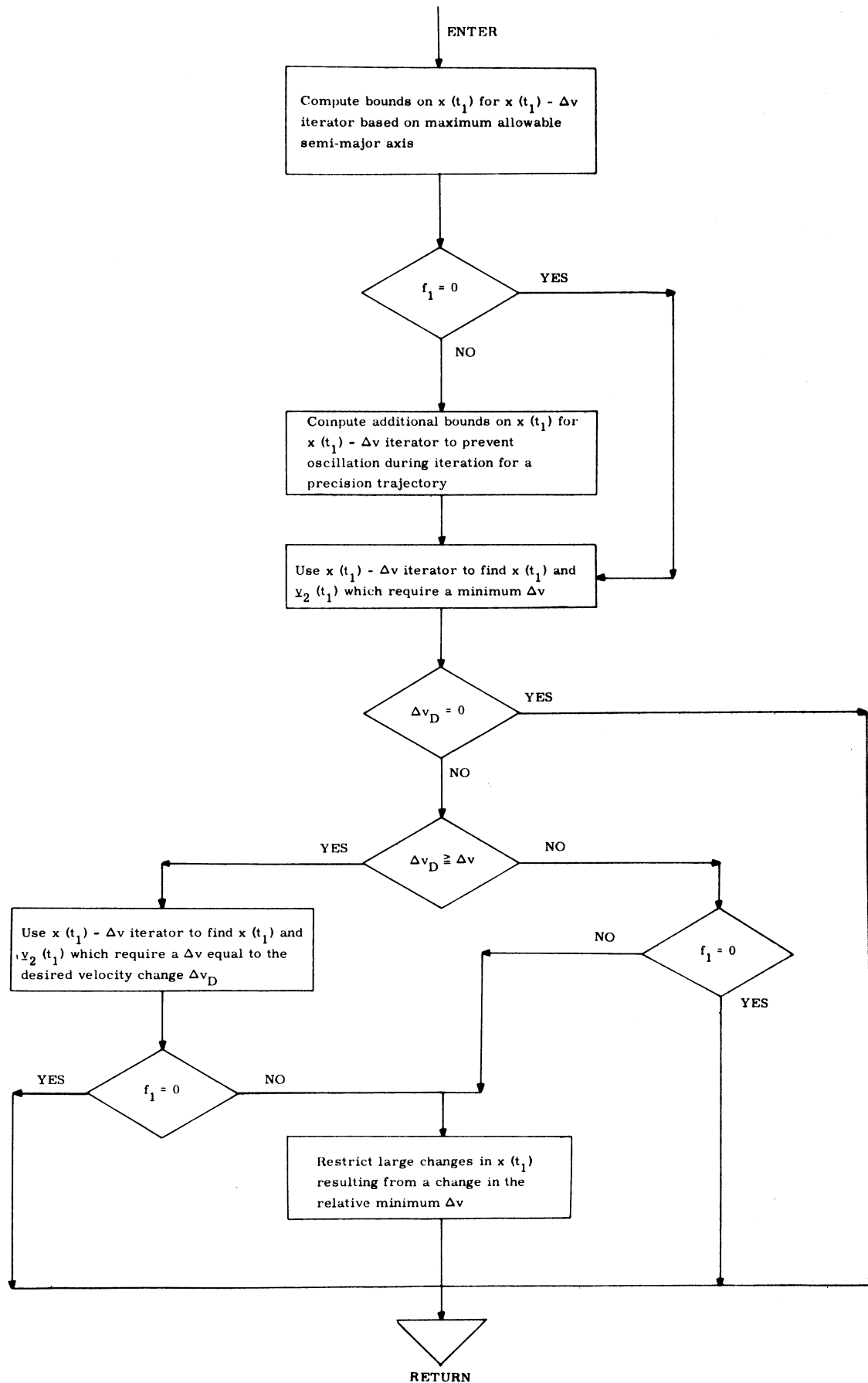


FIGURE 4. 5-3 SIMPLIFIED LOGIC DIAGRAM OF RETURN TO EARTH TARGETING  
- CONIC RETURN ROUTINE

In all cases the  $x(t_1) - \Delta v$  iterator is used to compute a post-abort velocity vector which requires a minimum  $\Delta v$ . In the event that a time critical answer is required, another iteration using the  $x(t_1) - \Delta v$  iterator will be made to compute a post-return velocity vector which requires a  $\Delta v$  equal to  $\Delta v_D$ . If the velocity change desired is less than the minimum required, the second iteration will be omitted.

During the calculation of the numerically integrated trajectory, the intermediate conics are additionally restricted to prevent large changes in the post-return flight path angle. This additional restraint is necessary in the time critical mode for certain near earth returns because both absolute and relative minimum  $\Delta v$  solutions exist.

#### 5.4.5.4 Logical Description

This section contains logical flow diagrams of the outside the sphere portion of the Return to Earth Targeting Program. Some items not fully discussed in the text are explicitly illustrated in the diagrams. Certain self contained procedures are detailed in the next section.

#### 5.4.5.4.1 Nomenclature

A list of the nomenclature used in this section and not previously defined follows.

$t_{IG}$	desired ignition time
$t_{21}$	time of flight ( $t_2 - t_1$ )
$p(t)$	primary body $\left\{ \begin{array}{l} 0 \text{ for earth} \\ 1 \text{ for moon} \end{array} \right.$
$x(t_1)_{LIM}$	limit on the cotangent of the post-return flight path angle
$x(t_1)_{MAX}, x(t_1)_{MIN}$	upper and lower bounds on the cotangent of the post-return flight path angle
$\Delta x(t_1)$	change in $x(t_1)$
$\Delta x(t_1)_{MAX}$	maximum allowed change in $x(t_1)$
$x(t_2)$	final cotangent of the flight path angle
$x(t_2)_{PRE}$	final cotangent of the flight path angle for a precision trajectory
$P_{CON}$	semi-latus rectum of a conic trajectory
$R_{ERR}$	precision trajectory reentry radius error

$\mu_E$	earth's gravitation constant
$MA_1, MA_2$	maximum major axes
$\underline{n}$	normal vector
$\phi_2$	$\left\{ \begin{array}{l} + 1 \text{ near apogee solution} \\ - 1 \text{ near perigee solution} \end{array} \right.$
$\phi_4$	precision trajectory direction switch
$f_2$	iterator mode flag
$n_1, n_2$	counters
$\beta_1$ to $\beta_{14}$	intermediate variables



#### 5.4.5.4.2 Discussion of the Logical Flow Diagram

The following discussion is intended to supplement the information contained in the referenced diagrams.

Three inputs are required in addition to the CSM state vector (Fig. 4.5-4). They are the objective velocity change desired ( $\Delta v_D$ ), the desired time of ignition ( $t_{IG}$ ), and the reentry angle ( $\gamma(t_2)_D$ ). The reentry angle should nominally be set to zero, allowing the program to compute a reentry angle in the center of the reentry corridor. The astronaut may select the reentry angle if a minimum  $\Delta v$  solution lying on the edge of the reentry corridor is required. This may be necessary in the event the reaction control system is used for the return. See Fig. 4.5-15 for a detailed explanation of the reentry angle calculations.

The Coasting Integration Routine is used to advance the state vector to the desired return time. A check is made to insure that the state vector is not inside the moon's sphere of influence. If the state vector is outside the sphere of influence, precomputations are executed. They are explained in detail later. A conic trajectory is then generated which satisfies the estimated reentry constraints and  $\Delta v_D$  requirements. The procedure CONICRETURN, which computes the initial state vector of this conic, will be discussed later. The Time Radius Subroutine, described in Section 5.5.8, is used to compute the time of flight to the desired radius and advance the state vector through that time of flight.

Based on this final state vector (Fig. 4.5-5), a new reentry radius ( $R_{CON}'$ ) and a new value for the cotangent of the reentry flight path angle ( $x(t_2)'$ ) can be computed. If they are sufficiently close to the previously used values, the conic trajectory is accepted. Otherwise a new conic is computed which satisfies the new reentry constraints. A counter is employed to protect against an excessive number of iterations.

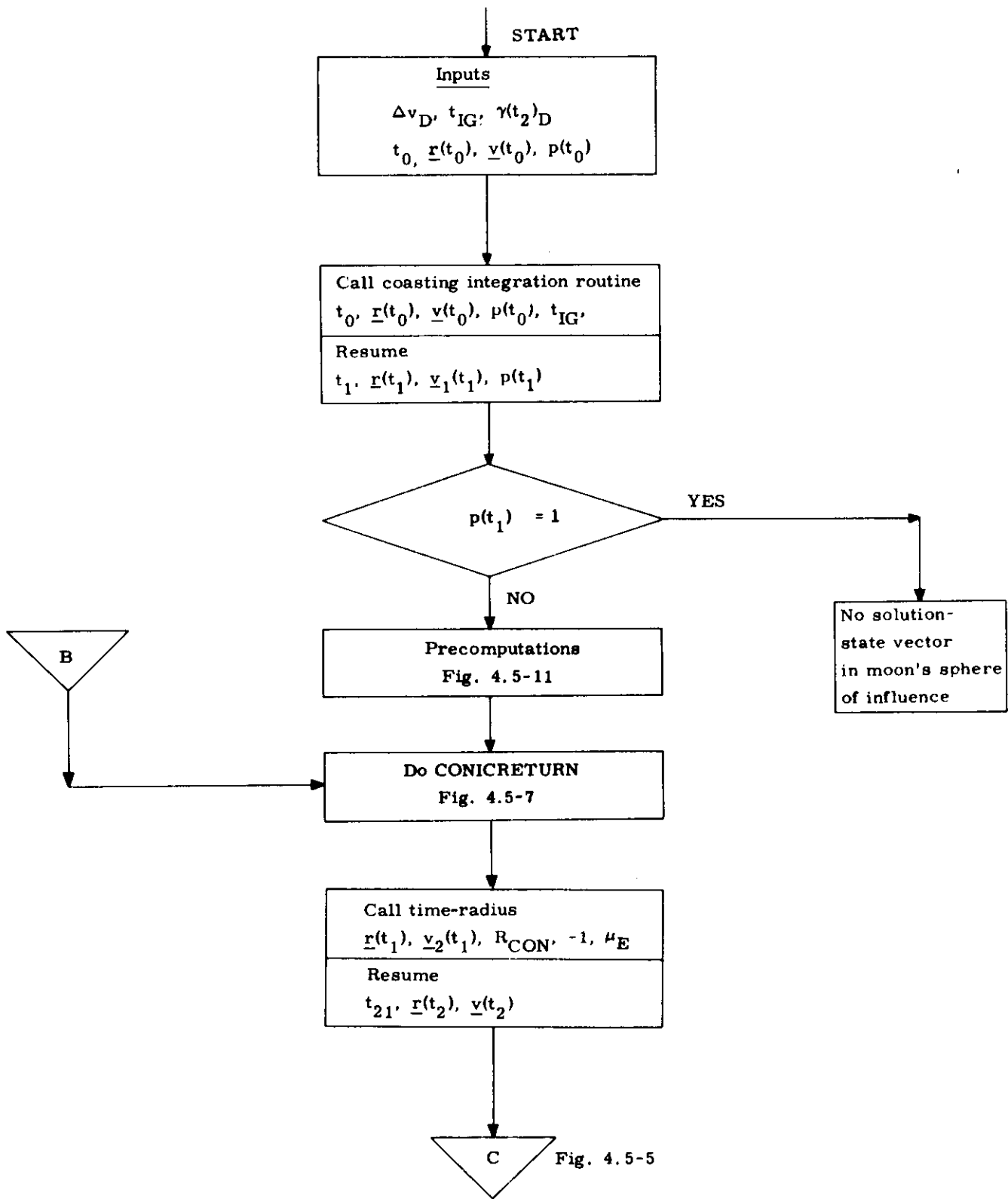


FIGURE 4.5-4 LOGICAL DIAGRAM OF RETURN TO EARTH TARGETING

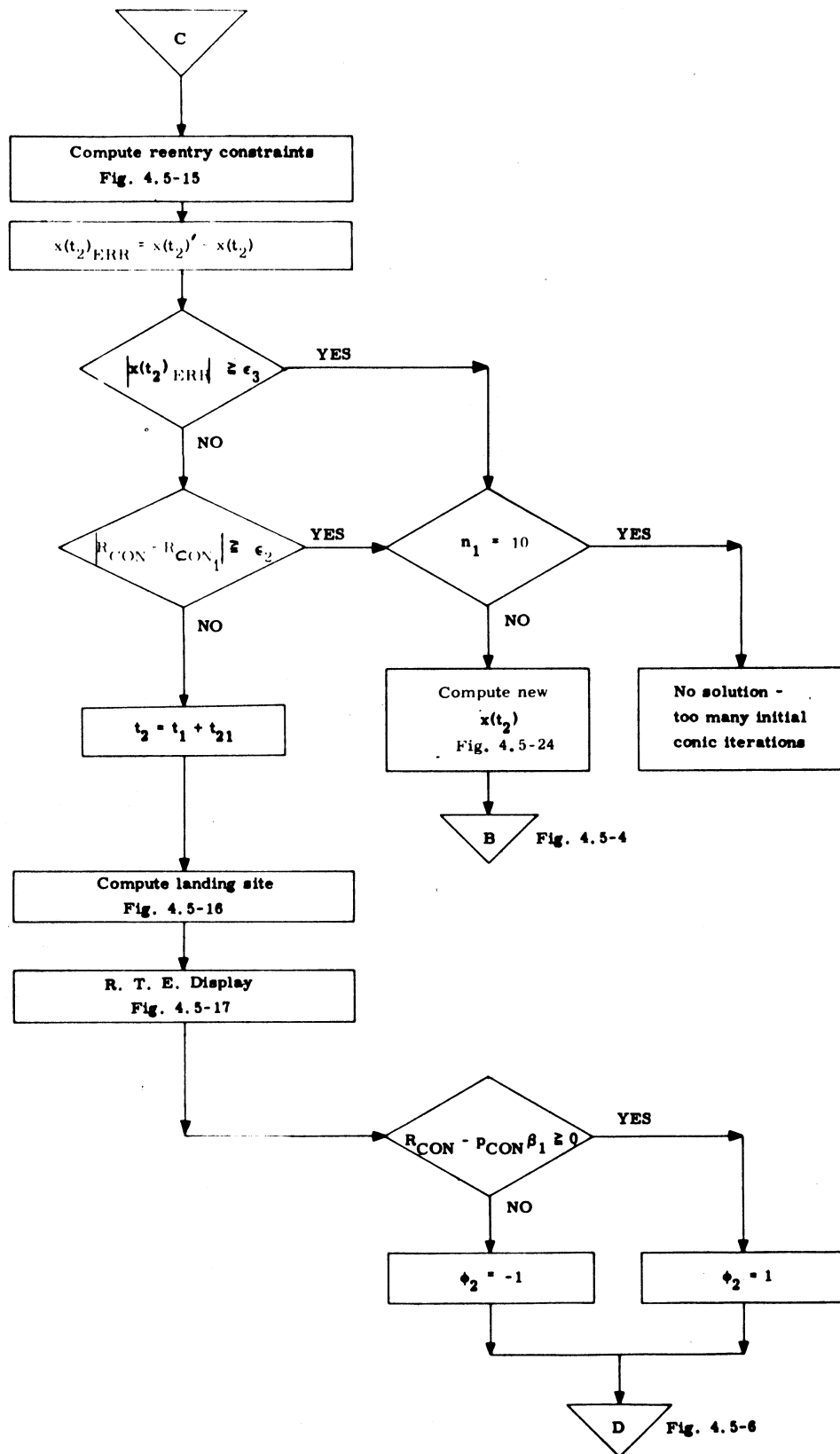


FIGURE 4.5-5 LOGICAL DIAGRAM OF RETURN TO EARTH TARGETING

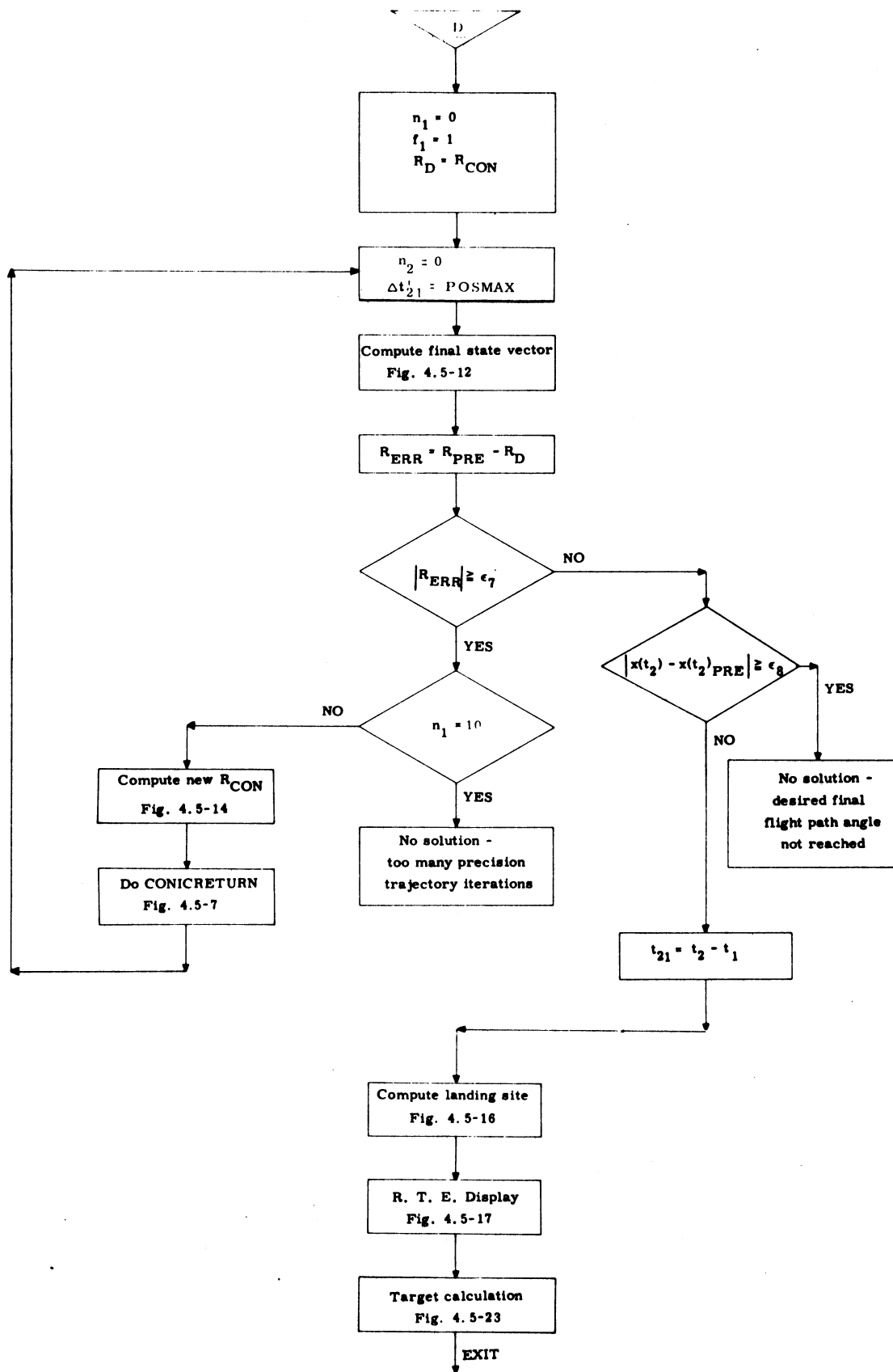


FIGURE 4.5-6 LOGICAL DIAGRAM OF RETURN TO EARTH TARGETING

If the initial conic is accepted, an estimate is made of the latitude and longitude of the landing site. For long flight times, however, differences in time of flight between the conic and numerically integrated trajectory will cause an error in the conically estimated landing site longitude. The results of the preceding calculations are displayed, and based on this information the astronaut may elect to proceed with the calculation of a numerically integrated trajectory or exit the program. As explained in Section 5.4.5.3, return trajectories from near earth orbits may be nearly circular, and therefore pass through the desired flight path angle twice with approximately the same radius. The unique solution reached by the initial conic is tested to determine whether the final state vector was achieved during the first or second pass through the desired reentry flight path angle. A switch  $\phi_2$  is set appropriately, and with one exception all succeeding precision and conic trajectories are required to behave in a similar manner. The exception will be explained in detail in Section 5.4.5.4.3.

The post-return state vector of the conic trajectory is now advanced until the desired flight path angle of reentry is reached (Section 5.4.5.5). A radius error will exist between the desired final radius ( $R_D$ ) and the final radius magnitude of the numerically integrated trajectory. If this error is small, the precision trajectory is accepted. Otherwise a new conic trajectory is generated which uses an offset final radius magnitude. If the error in the final radius is small, a final check is made to insure that the reentry flight path angle is acceptable.

#### 5.4.5.4.3 Discussion of Logical Flow Diagram - CONICRETURN

The procedure CONICRETURN, illustrated in Figures 4.5-7 and 4.5-8, is used to generate initial state vectors for the Return to Earth Targeting Program. Equations required to initialize CONICRETURN each time it is used are described in Section 5.4.5.5. The conic problems solved by the routine were described in Section 5.4.5.3.4.

The logical structure shown in Figs. 4.5-7 and 4.5-8 is based in the following facts.

(1) There is only one minimum if the pre-return velocity vector has a negative radial component.

(2) There may be two minima if the pre-return velocity vector has a positive radial component. In this case the absolute minimum post-return velocity vector has a positive radial component and the relative minimum has a negative component.

As explained in Section 5.4.5.3.4 and 5.4.5.4.2, it is necessary to compute additional bounds on  $x(t_1)$  during the search for a precision trajectory. The logic used to do this is shown on the lower portion of Fig. 4.5-7.

A check is made to determine whether the locus of possible solutions includes trajectories which achieve the radius  $R_{CON}$  and the flight path angle  $\gamma(t_2)$  near apogee, and if so the variable  $\beta_6$  will be positive\*.

---

\*With one exception every conic solution generated by this routine achieves the desired reentry flight path angle ( $\gamma(t_2)$ ) twice during the passage from apogee to perigee. The first time  $\gamma(t_2)$  is reached during the passage from apogee to perigee will be referred to as near apogee, and the second time will be referred to as near perigee. The desired reentry angle will be achieved only once if  $\gamma(t_2)$  equals the maximum flight path angle of the conic.

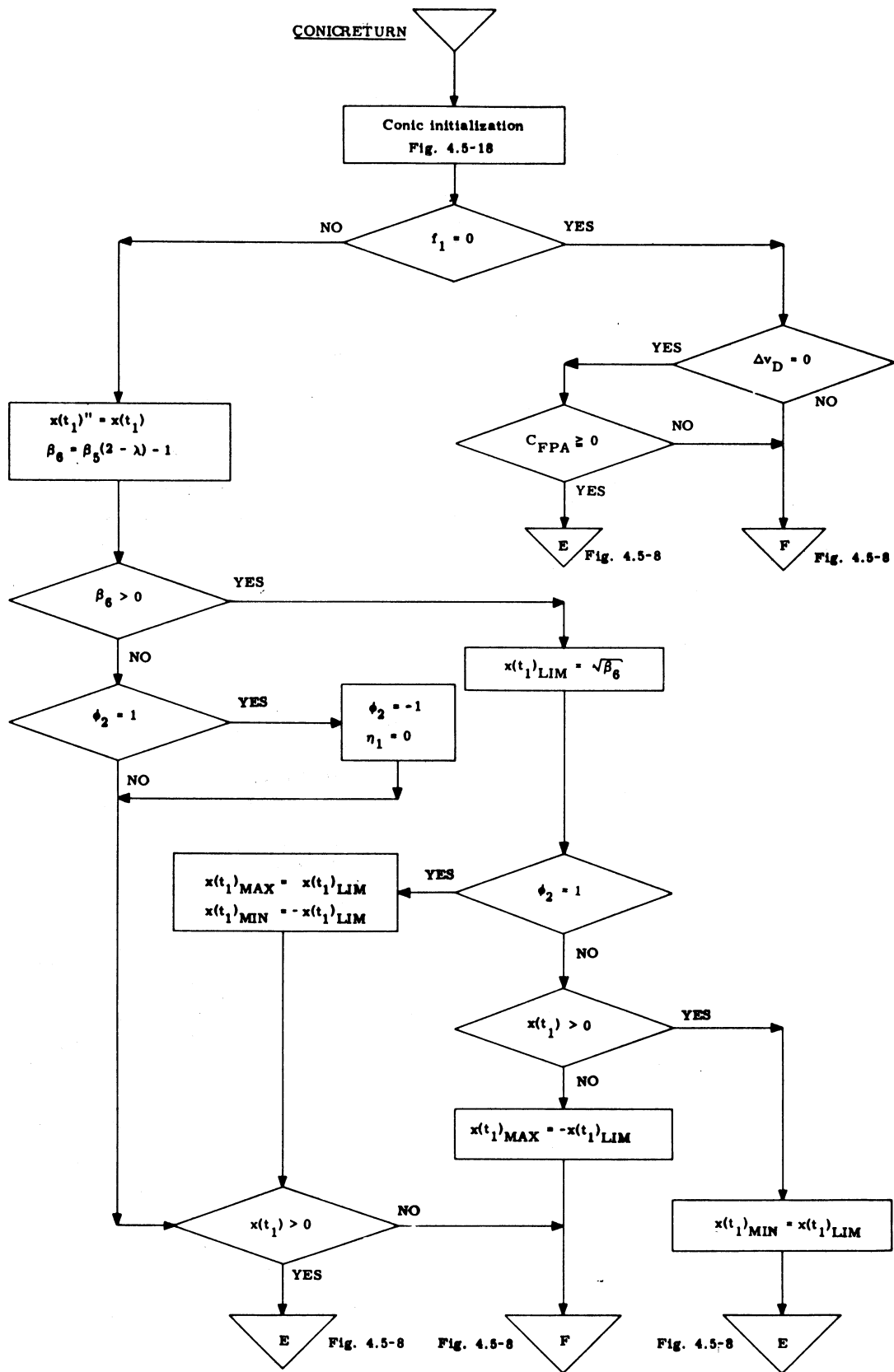


FIGURE 4.5-7 LOGICAL DIAGRAM OF RETURN TO EARTH TARGETING -  
CONIC RETURN ROUTINE

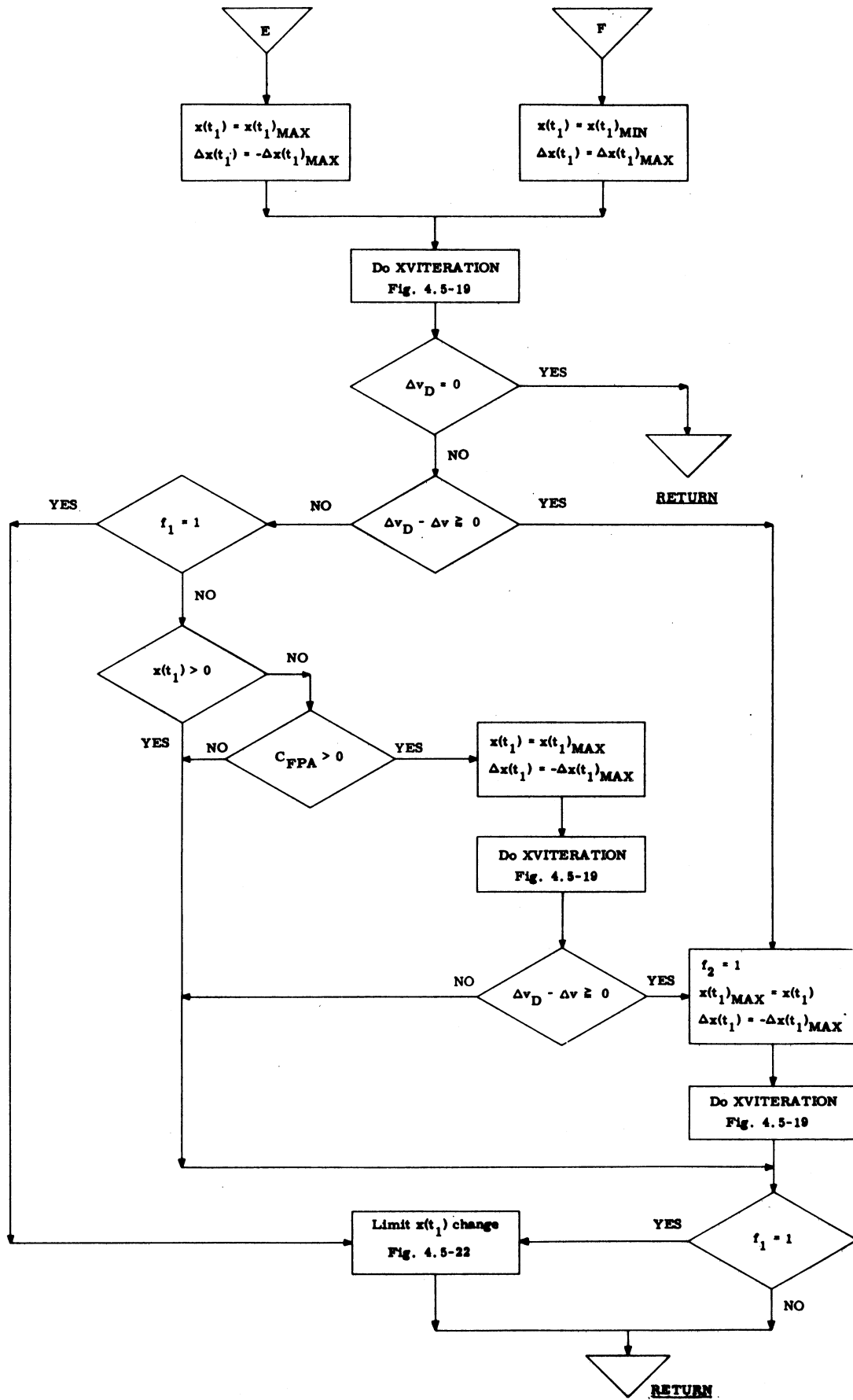


FIGURE 4.5-8 LOGICAL DIAGRAM OF RETURN TO EARTH TARGETING - CONIC RETURN ROUTINE

5.4-56



If  $\beta_6$  is negative, only near perigee solutions are possible and hence no additional bounds on  $x(t_1)$  are computed. The initial conic trajectory may have achieved the desired reentry flight path angle near apogee, however, and this type of solution is no longer possible due to a change in  $R_{CON}$ . This is the one exception mentioned in Section 5.4.5.4.2. The value of  $\phi_2$  must be set to -1, forcing all subsequent conic solutions to reach the desired reentry flight path angle near perigee, and then the entire precision trajectory phase is recycled.

If  $\beta_6$  is positive, it is used to calculate  $x(t_1)_{LIM}$ . Three types of solutions may exist.

$$(1) -x(t_1)_{LIM} < x(t_1) < x(t_1)_{LIM}$$

If  $x(t_1)$  takes a value in this region, the desired reentry flight path angle ( $\gamma(t_2)$ ) will be achieved near apogee.

$$(2) x(t_1) < -x(t_1)_{LIM} \text{ or } x(t_1)_{LIM} < x(t_1)$$

If  $x(t_1)$  takes a value in these regions, the desired reentry flight path angle ( $\gamma(t_2)$ ) will be achieved near perigee.

$$(3) x(t_1) = x(t_1)_{LIM} \text{ or } x(t_1) = -x(t_1)_{LIM}$$

If  $x(t_1)$  takes on either of these values, the desired reentry flight path angle ( $\gamma(t_2)$ ) will equal the maximum angle achieved by the conic.

The above information is used to set new bounds on  $x(t_1)$  which will force the conic to reach a near apogee or near perigee solution as prescribed by the initial conic.

If the program is being operated in the fuel critical mode, the  $x(t_1) - \Delta v$  iterator is used to compute an absolute minimum  $\Delta v$  solution.

If the program is being operated in the time critical mode, the  $x(t_1) - \Delta v$  iterator will be used to compute a solution requiring a velocity change equal to the objective velocity change.

#### 5. 4. 5. 5 Detailed Flow Diagrams

( Certain portions of the flow diagrams mentioned in the previous sections are illustrated in Figs. 4. 5-11 through 4. 5-24.

A list of nomenclature used in the following diagrams and not defined previously follows.

$\lambda$	radius ratio
$p$	semi-latus rectum
$a$	1/semi-major axis
$r_2(t_2)$	target aim point
$r_{LS}$	vector through landing site
$\theta_{LAT}$	landing site latitude
$\theta_{LONG}$	landing site longitude
$\dot{m}$	mass flow rate
$F$	thrust
$f_3$	flag set to $\begin{cases} 0 \text{ RCS jets} \\ 1 \text{ SPS engine} \end{cases}$
$t_{LS}$	time of landing
$\Delta t_B$	length of burn
$\Delta v_{LV}$	velocity change vector in local vertical coordinates
$a_{MG}$	middle gimbal angle
$\theta_1, \theta_2, \theta_3$	intermediate variables
$\Delta t_{21}$	adjustment of $t_{21}$
$\Delta R_{CON}$	change in the final radius of a conic trajectory

5, 4-59

Revised COLOSSUS

Added GSOP #R-577 PCR # 626 Rev. 3 Date 11/1/68

The diagrams are essentially self-explanatory, however, a discussion of two of the diagrams will follow.

#### Precomputations

The equations illustrated in Figure 4.5-11 are used to compute basic variables required during the remaining portions of the program.

As mentioned in the introduction, the pre-return and post-return state vectors are required to lie in the same plane. If the pre-return position and velocity vectors are nearly co-linear, the plane of the return trajectory will be defined to prevent reentry near the earth's poles.

Retrograde return to earth trajectories are not allowed.

#### Final State Vector Computation

The method illustrated in Figures 4.5-12 and 4.5-13 is used to extrapolate a post-return state vector to the desired reentry flight path angle. An iterative procedure is again used.

First the post-return state vector is advanced through the best estimate of the transit time to reentry using the Coasting Integration Routine. Since it is not necessary to explicitly compute the flight path angle of the resulting reentry state vector, the cotangent of the flight path angle is compared with the cotangent of the desired reentry flight path angle. If they are sufficiently close, the resulting reentry state vector is accepted.

If the desired cotangent of the reentry flight path angle is not achieved, an iterative procedure is started. This procedure uses Conic Subroutines to compute the time of flight ( $\Delta t_{21}$ ) to the desired reentry flight path angle.

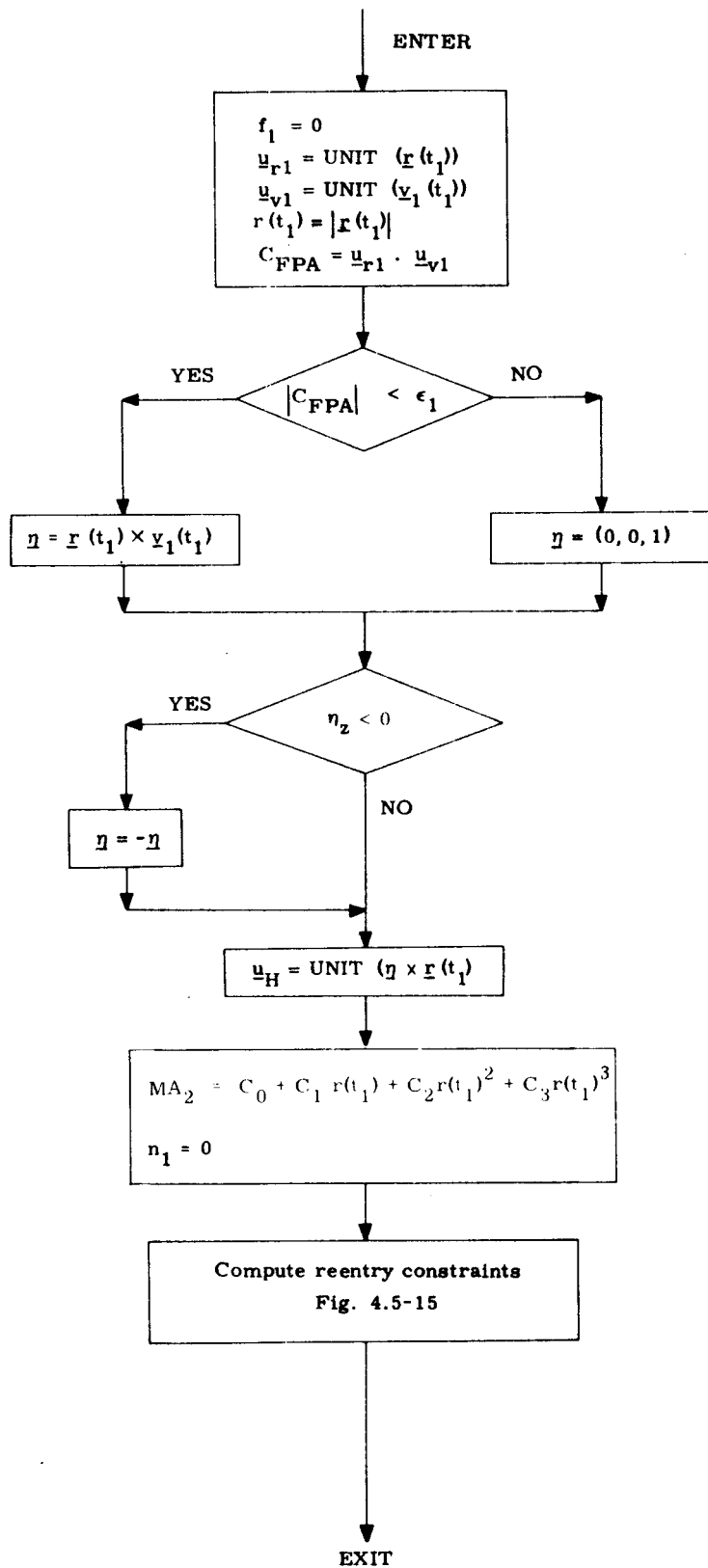


FIGURE 4.5-11 PRECOMPUTATIONS

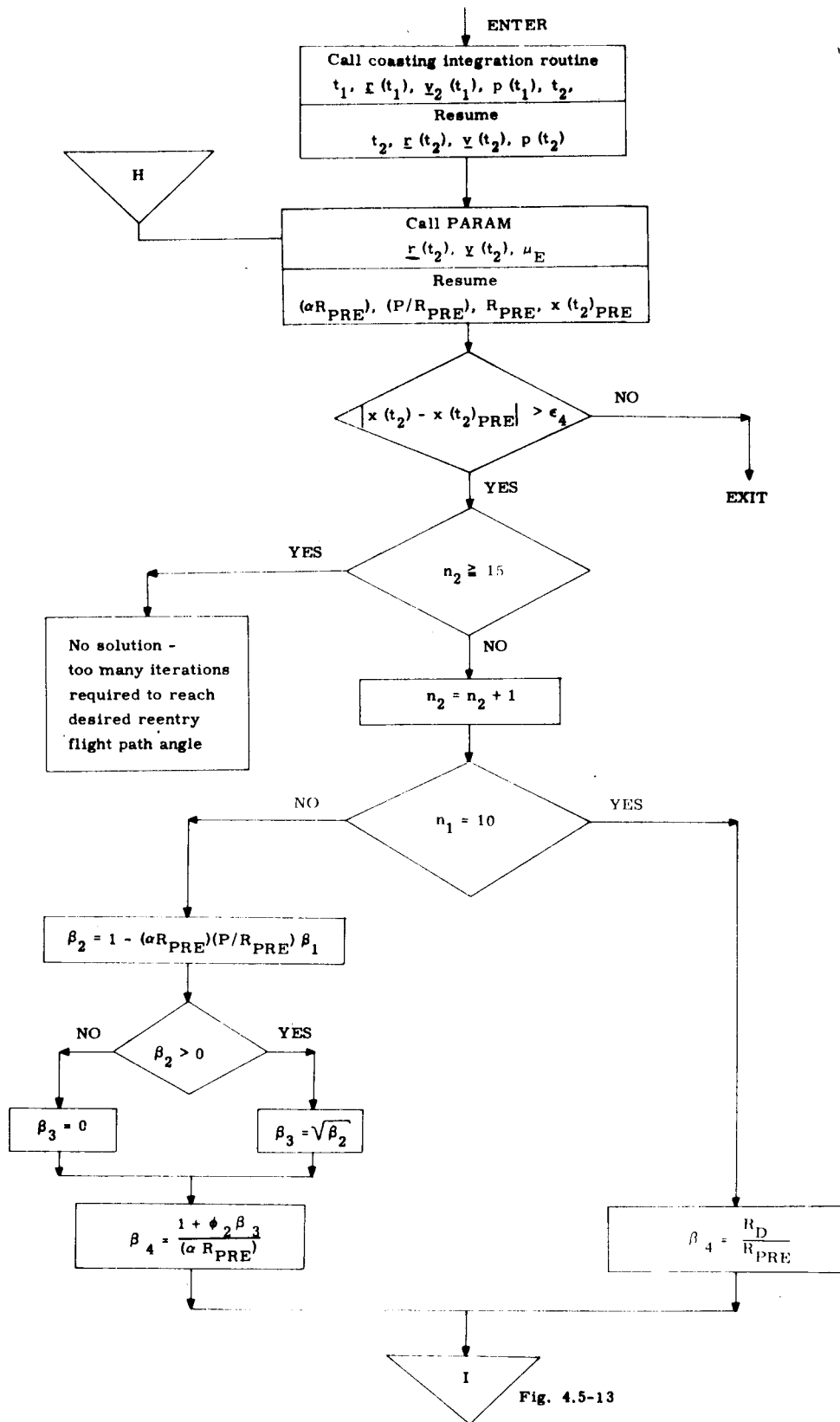


Fig. 4.5-13

FIGURE 4.5-12 FINAL STATE VECTOR COMPUTATION

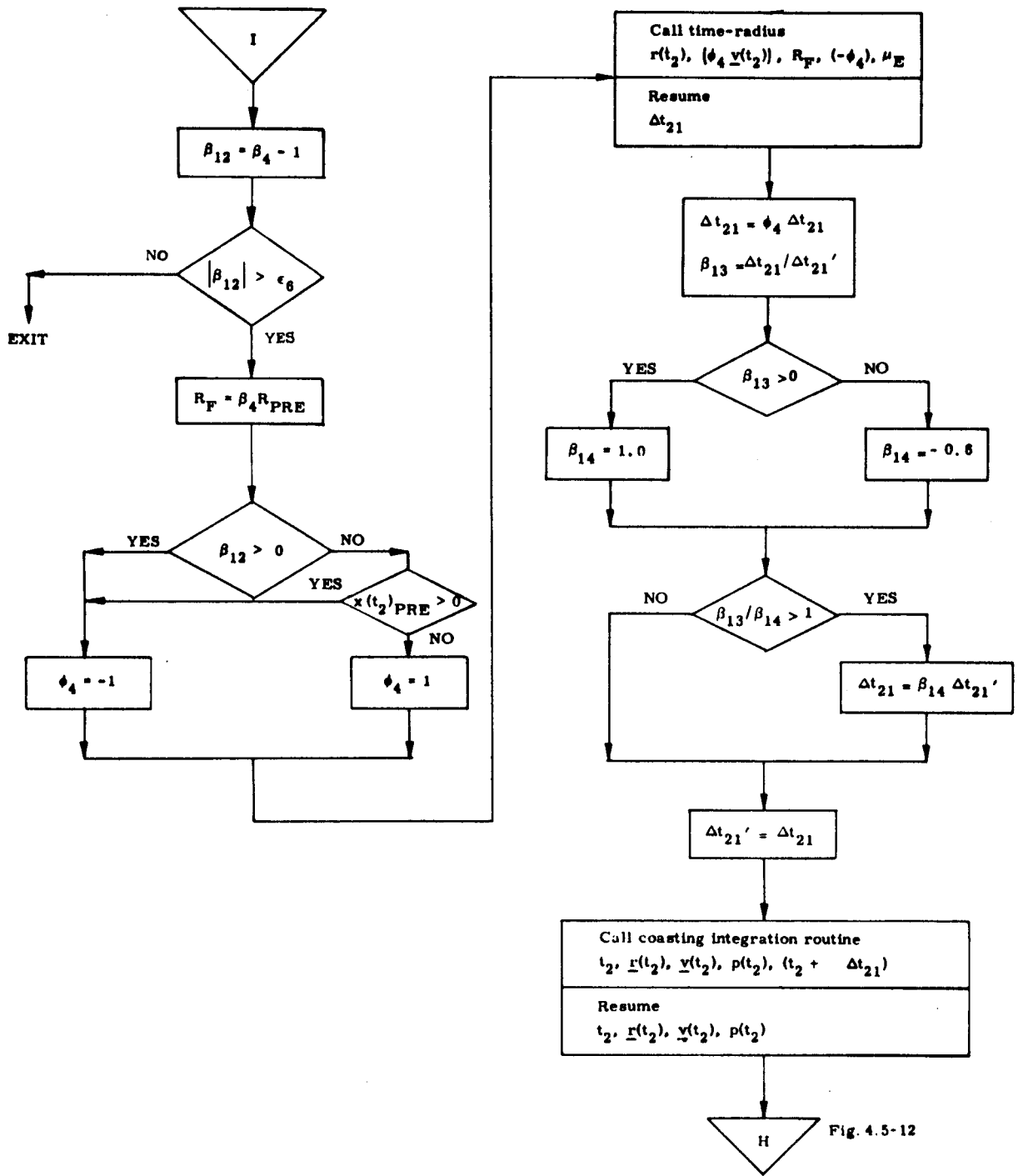


FIGURE 4.5-13 FINAL STATE VECTOR COMPUTATION

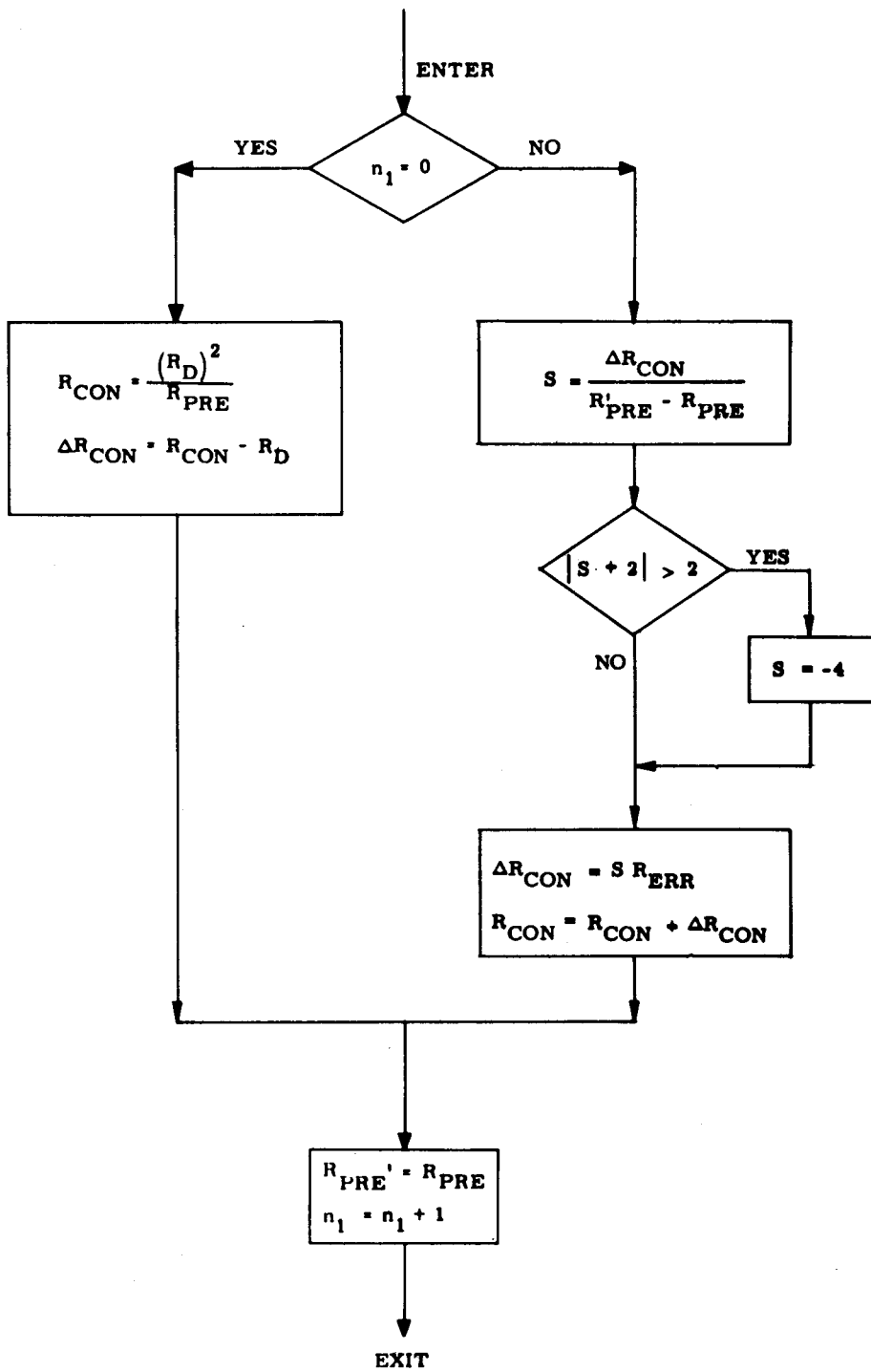


FIGURE 4.5-14 NEW  $R_{CON}$  COMPUTATION



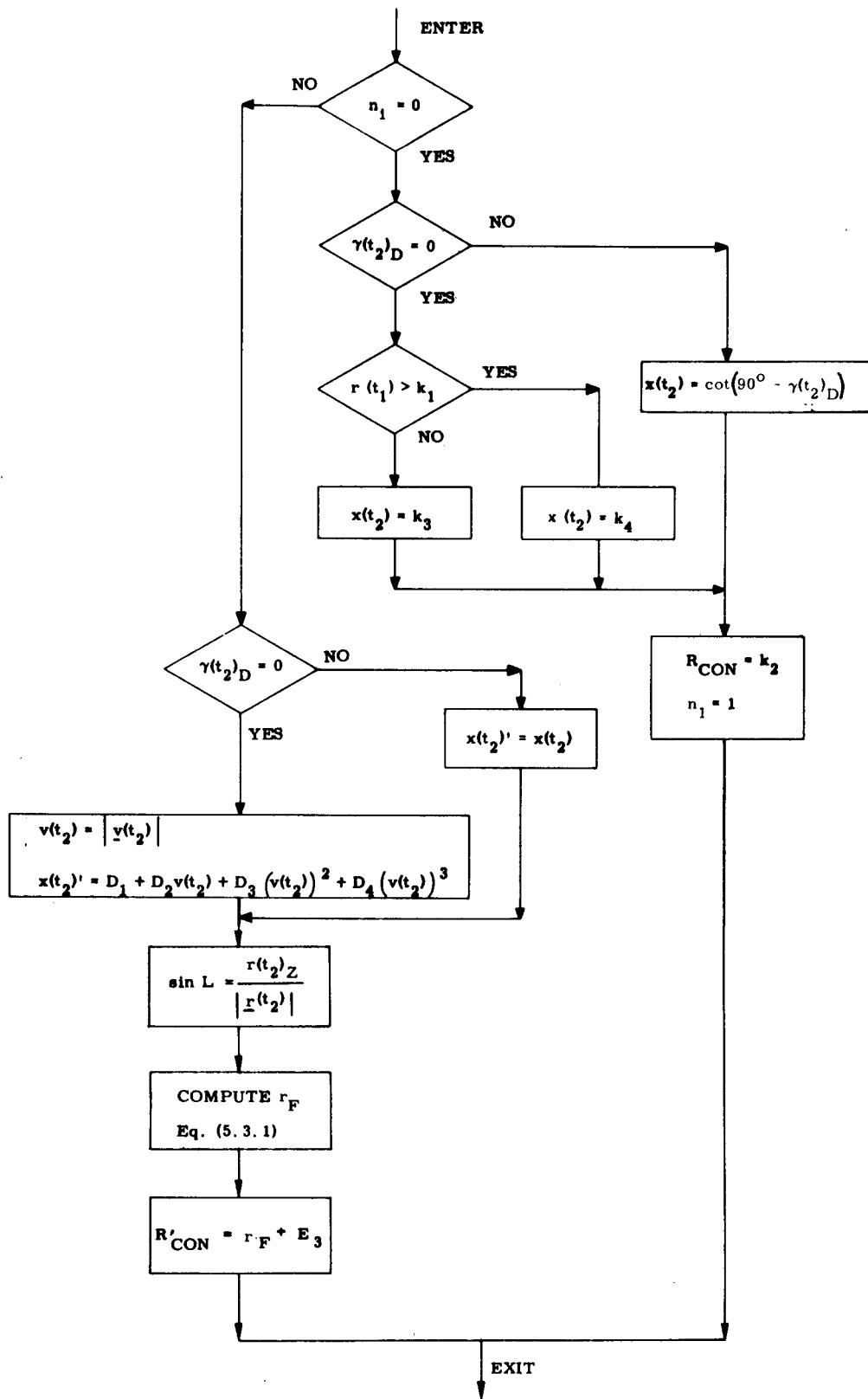


FIGURE 4. 5-15 REENTRY CONSTRAINT COMPUTATION

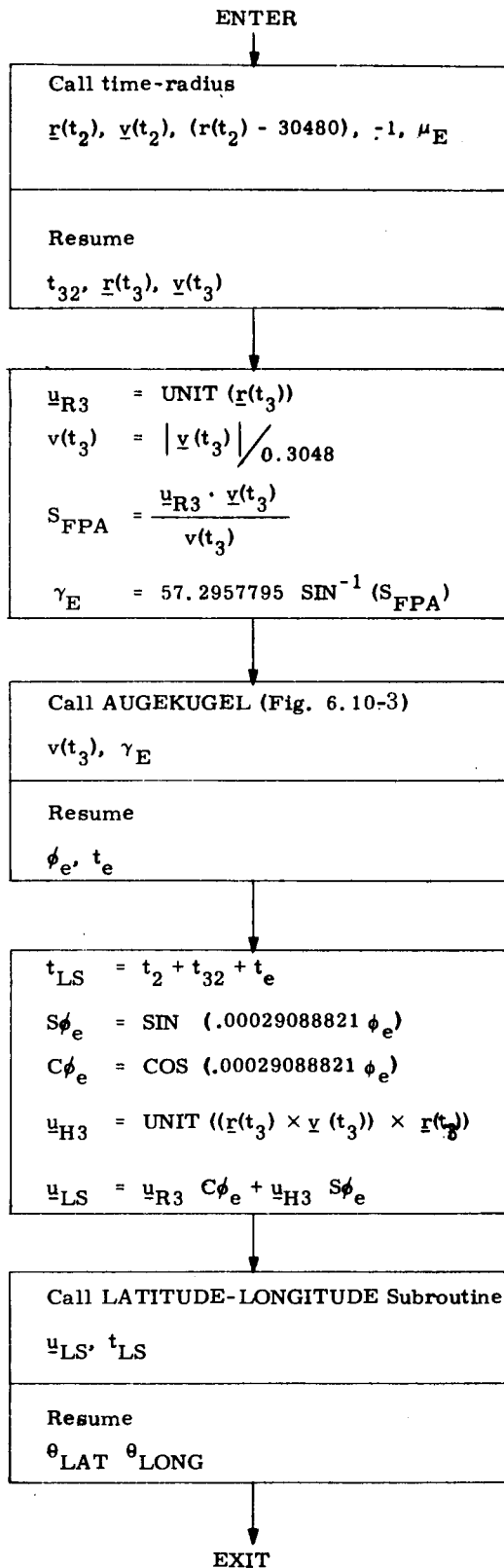


FIGURE 4.5-16 LANDING SITE COMPUTATION

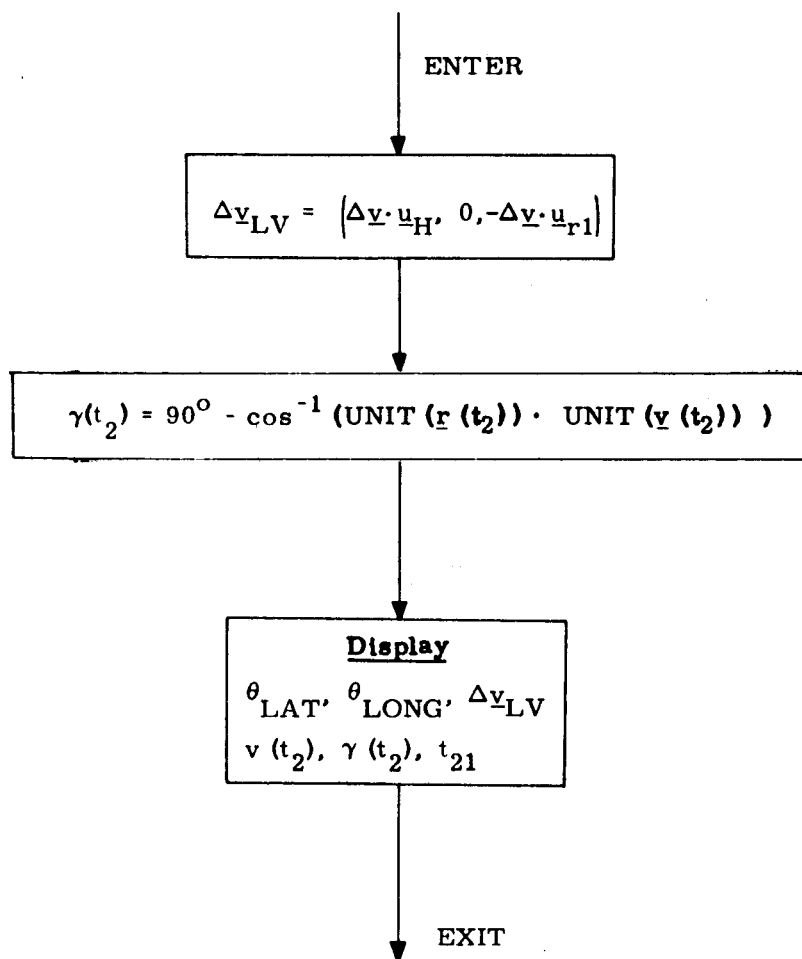


FIGURE 4.5-17 RETURN TO EARTH DISPLAY

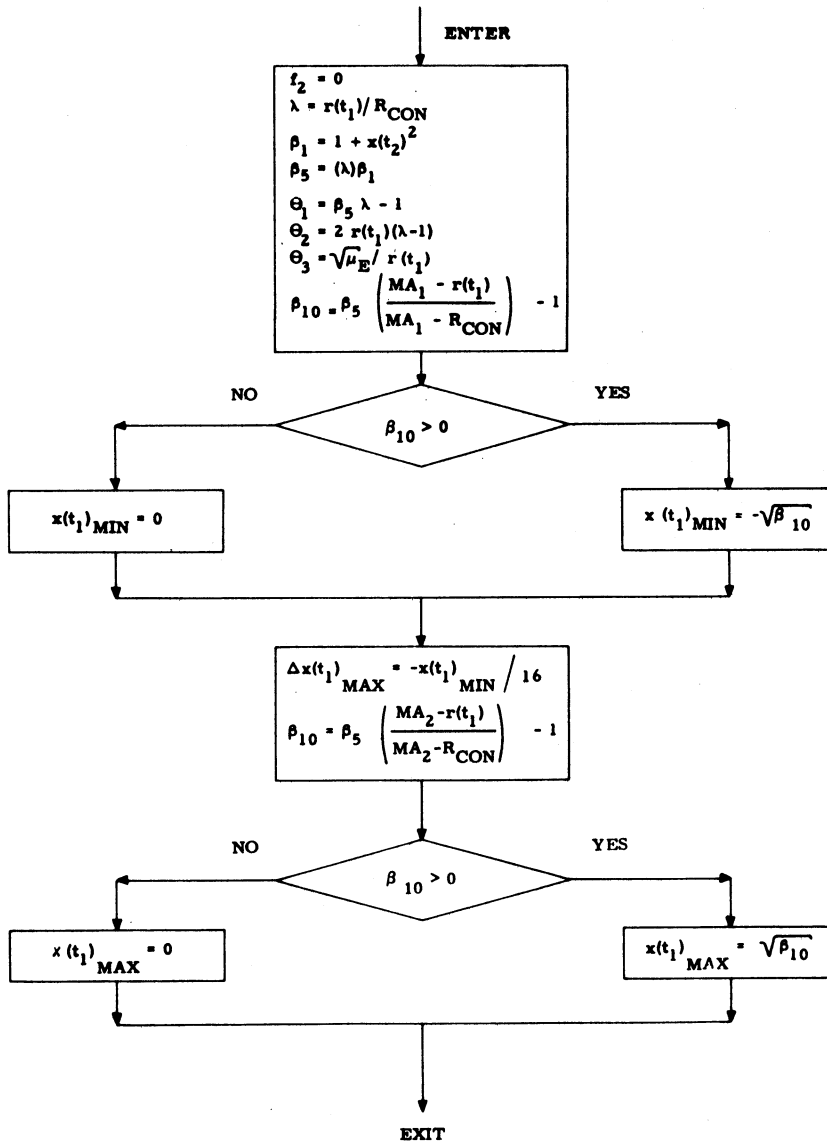


FIGURE 4.5-18 CONIC INITIALIZATION

5.4-68

Rev. 3 - 11/68

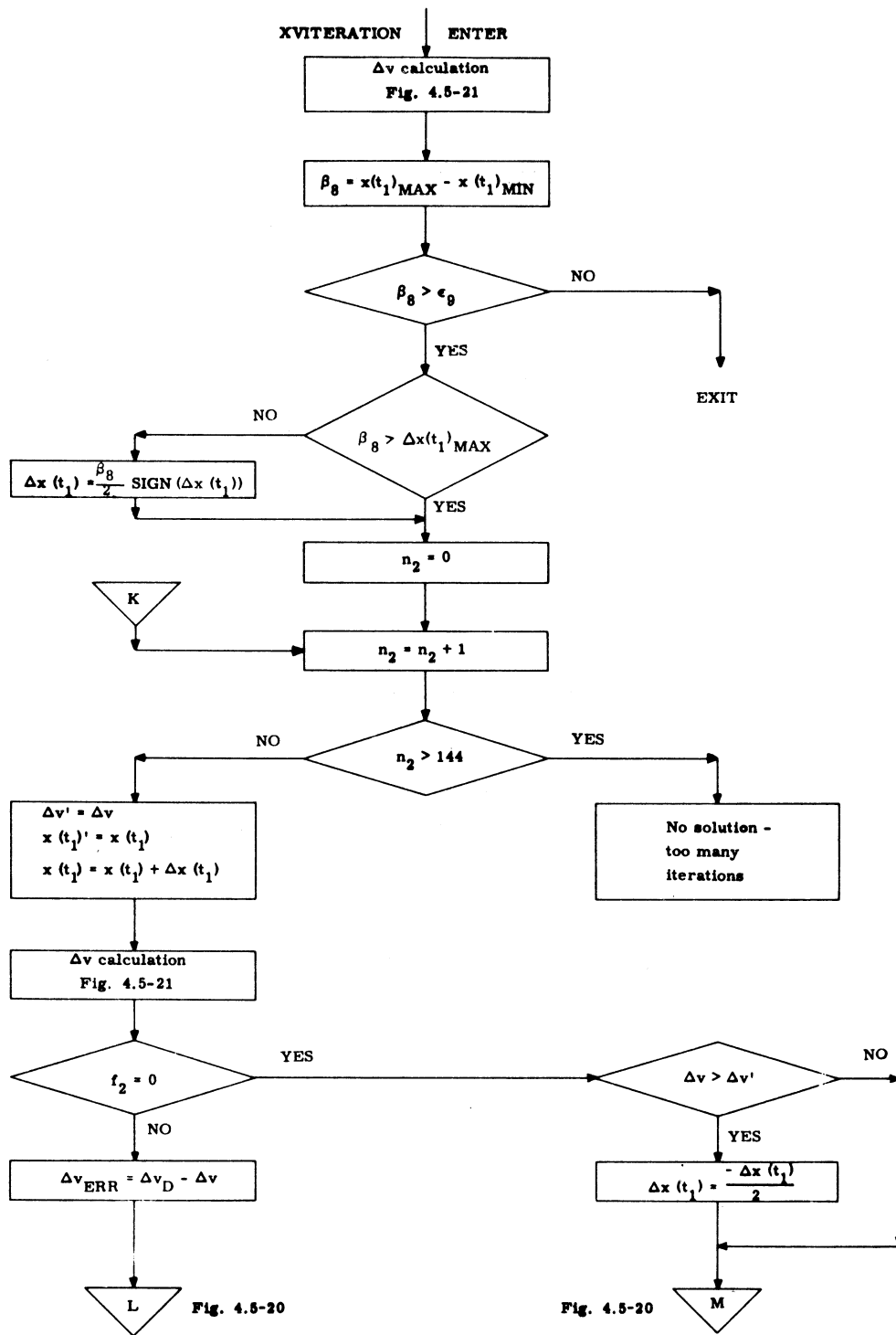


FIGURE 4.5-19  $x(t_1) - \Delta v$  ITERATOR

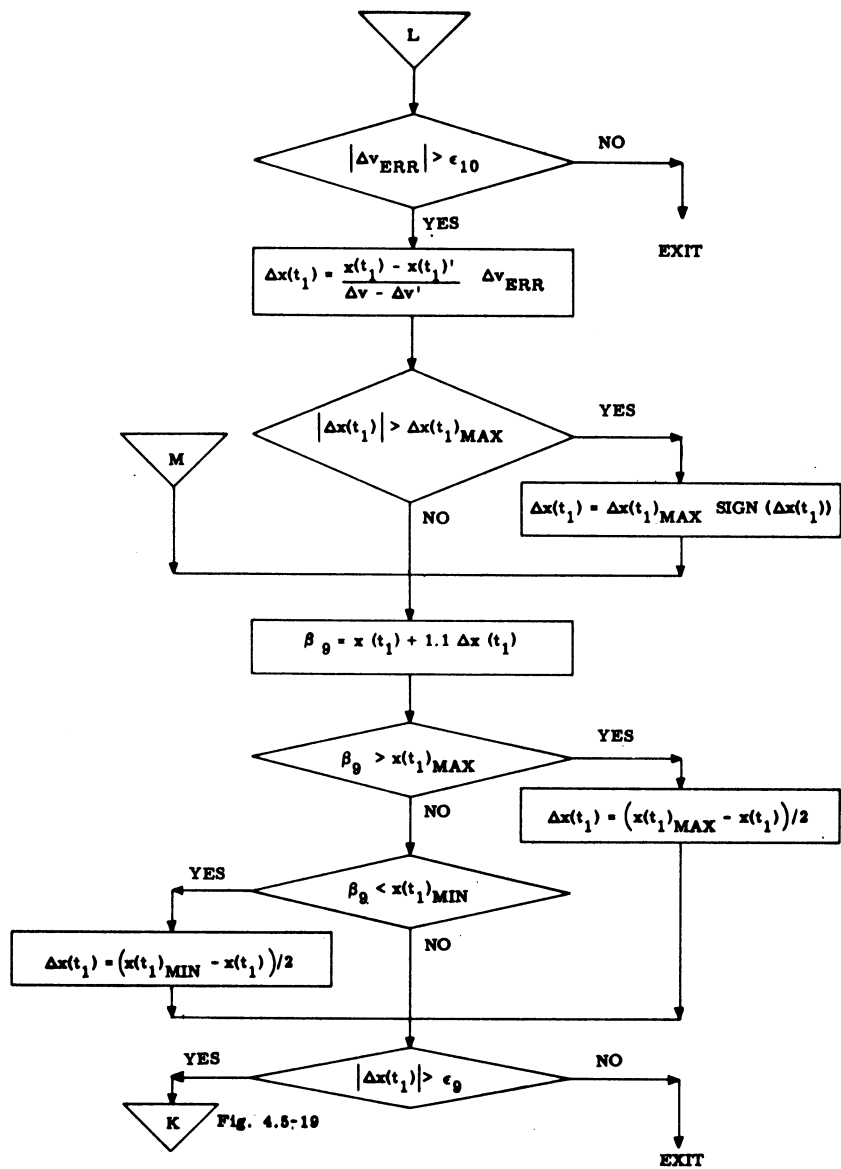


FIGURE 4.5-20  $x(t_1) - \Delta v$  ITERATOR

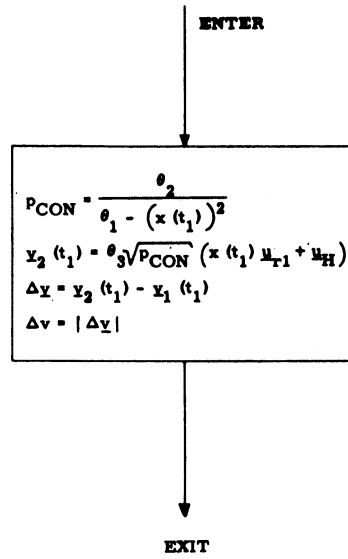


FIGURE 4.5-21  $\Delta v$  CALCULATION

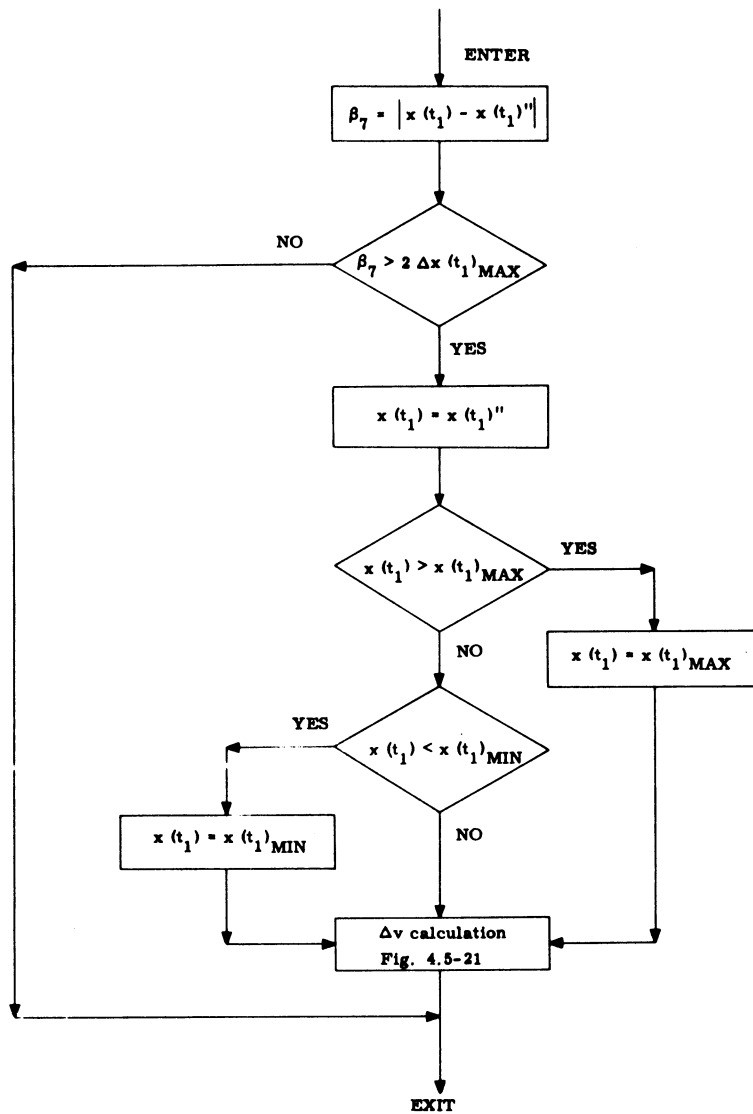


FIGURE 4. 5-22 LIMIT  $x(t_1)$  CHANGE



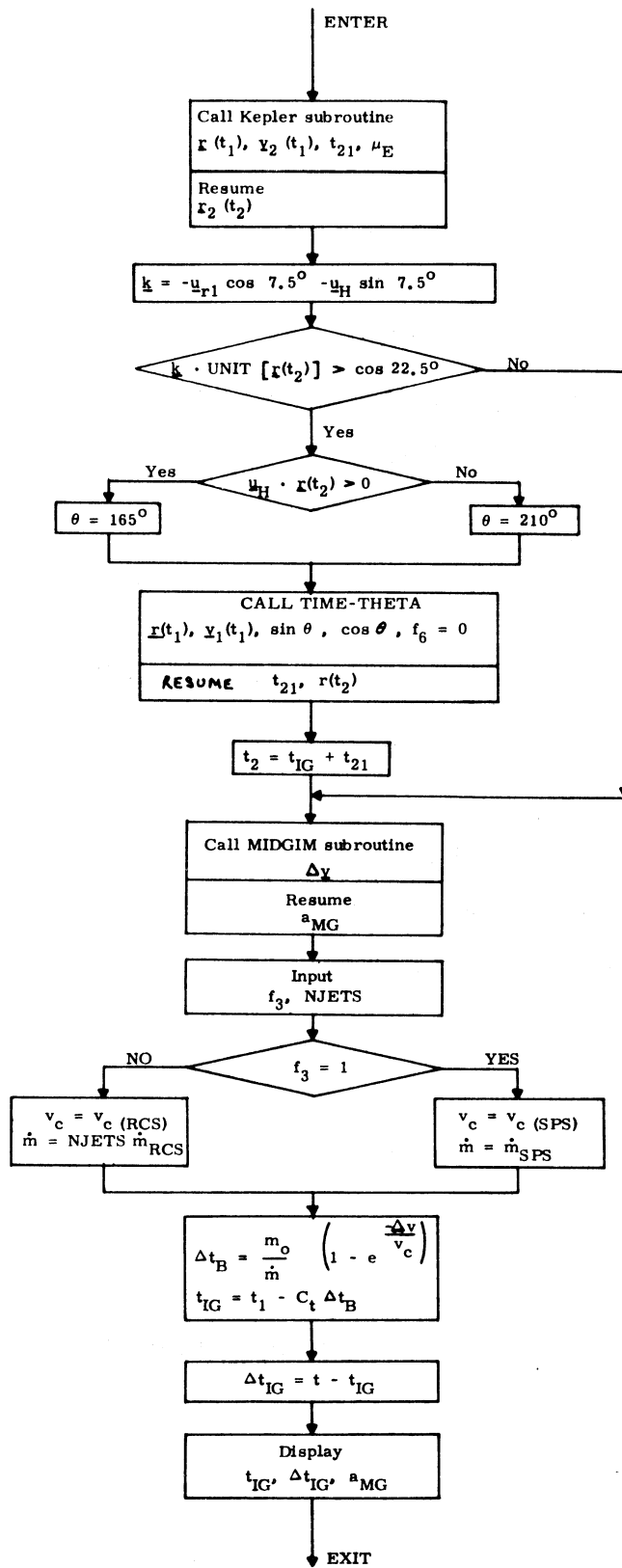


Figure 4-5-23 Target Computation

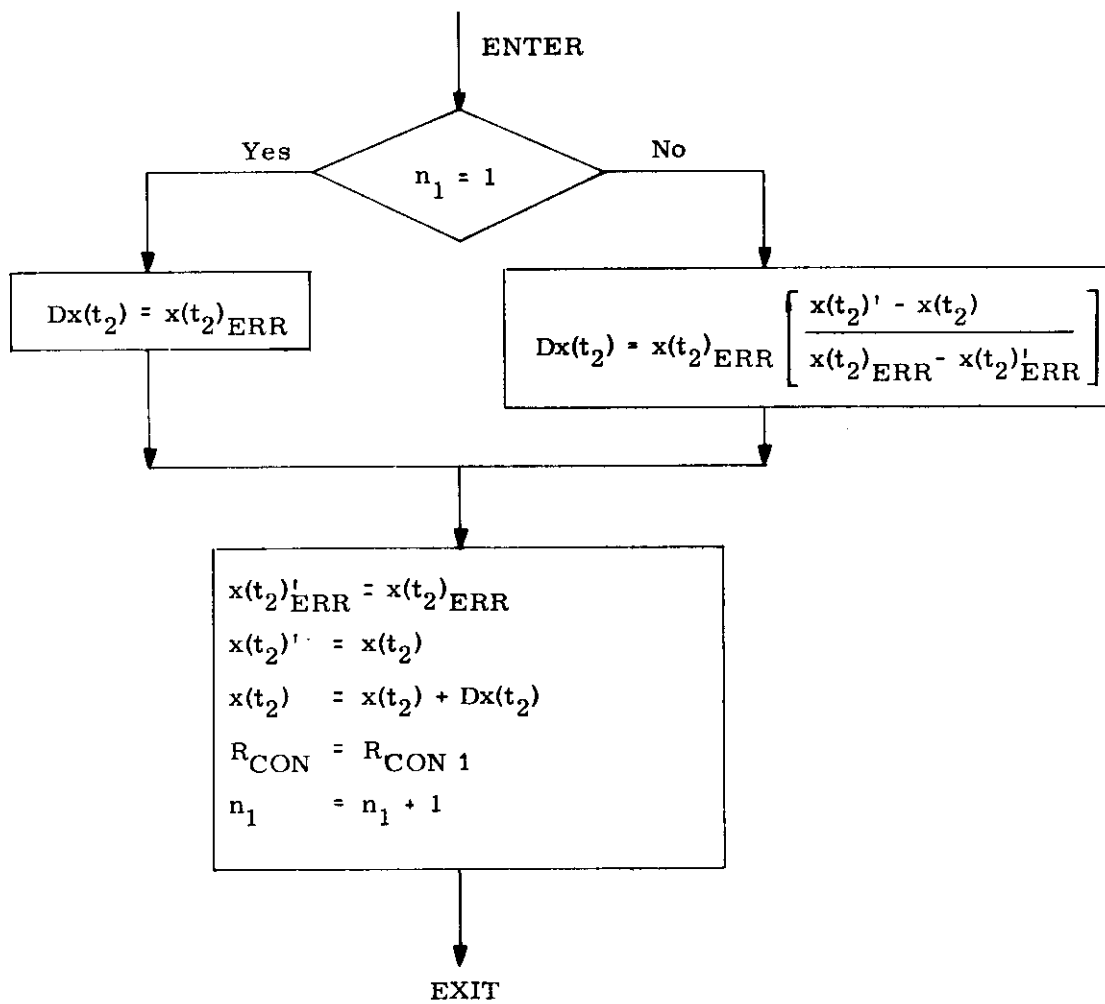


FIGURE 4.5-24 NEW  $x(t_2)$  COMPUTATION

## 5.5 BASIC SUBROUTINES

### 5.5.1 GENERAL COMMENTS

The basic solar system and conic trajectory sub-routines which are used by the various guidance and navigation routines are described in this section.

#### 5.5.1.1 Solar System Subroutines

The subroutines used to determine the translation and rotation of the relevant solar system bodies (earth, moon and sun) are designed specifically for a fourteen day lunar landing mission. The method of computing the position and velocity of the moon and the sun relative to the earth is given in Section 5.5.4. The transformations between the Basic Reference Coordinate System and the Earth- and Moon-fixed Coordinate Systems are described in Section 5.5.2. The procedure for transforming between vectors in the Basic Reference Coordinate System and latitude, longitude, altitude coordinates is given in Section 5.5.3. Although these subroutines are normally used in the lunar landing mission, they are valid for use in any mission of not more than fourteen days duration in earth-moon space.

### 5.5.1.2 Conic Trajectory Subroutines

This is a description of a group of conic trajectory subroutines which are frequently used by higher level routines and programs in both the Command Module and the Lunar Module computers.

These subroutines, whose block diagrams are presented in Sections 5.5.5 to 5.5.10, provide solutions to the following conic problems. (See nomenclature which follows)

- (1) Given  $\underline{r}(t_1)$ ,  $\underline{v}(t_1)$ ,  $t_D$ ; solve for  $\underline{r}(t_2)$ ,  $\underline{v}(t_2)$   
(Kepler Subroutine)
- (2) Given  $\underline{r}(t_1)$ ,  $\underline{r}(t_2)$ ,  $t_{D21}$ ,  $s_G$ ; solve for  $\underline{v}(t_1)$   
(Lambert Subroutine)
- (3) Given  $\underline{r}(t_1)$ ,  $\underline{v}(t_1)$ ,  $\theta$ ; solve for  $t_{21}$ ,  $\underline{r}(t_2)$ ,  $\underline{v}(t_2)$   
(Time-Theta Subroutine)
- (4) Given  $\underline{r}(t_1)$ ,  $\underline{v}(t_1)$ ,  $r(t_2)$ ,  $s_G$ ; solve for  $t_{21}$ ,  $\underline{r}(t_2)$ ,  $\underline{v}(t_2)$   
(Time-Radius Subroutine)
- (5) Given  $\underline{r}(t)$ ,  $\underline{v}(t)$ ; solve for  $r_P$ ,  $r_A$ ,  $e$   
(Apsides Subroutine)

In addition, the following useful subroutines are provided.

- (6) Conic Parameters Subroutine (See Fig. 5.10-1).
- (7) Geometric Parameters Subroutine (See Fig. 5.10-2).
- (8) Iterator Subroutine (See Fig. 5.10-3).

The solutions to the above set of conic problems have stringent accuracy requirements. Programming the fixed-point Apollo computer introduces two constraints which determine accuracy limitations: the 28 bit double precision word length, and the range of variables which is several orders of magnitude for the Apollo mission.

In order to maintain numerical accuracy when these subroutines are programmed into the Apollo computer, floating point programming techniques must be exercised. The effect is for even a simple equation to require a large number of computer instructions. The alternative to this is to separate the problem into phases, each with a different variable range. This, however, requires an even larger number of instructions. These considerations provide the incentive for efficiently organizing the conic equations as shown in the block diagrams.

In addition to the requirement for accuracy, the solution to the Kepler and Lambert Problems must be accomplished in a minimum of computation time in order that the guidance system operate satisfactorily in real time. This additional constraint dictates that a minimum of computer instructions be performed when solving the problem.

#### Method of Solution

To minimize the total number of computer instructions, the problems are solved in the "universal" form; i. e. only equations which are equally valid for the ellipse, parabola and hyperbola are used. Also these subroutines can be used with either the earth or the moon as the attracting body.

Kepler's equation, in the universal form, is utilized to relate transfer time to the conic parameters. All other necessary equations are also universal. The Kepler and Lambert problems are solved with a single iteration loop utilizing a simple first-order slope iterator. In the case of the Kepler problem a third order approximation is available to produce the initial guess for the independent variable (See Eq.(2. 2. 4) of Section 5. 2. 2. 2).

Sections 5. 5. 5 thru 5. 5. 10 provide block diagrams of the detailed computational procedures for solving the various problems. The equations are presented in block diagram form with the nomenclature defined below.

#### Range of Variables

As indicated previously, the programming of the conic subroutines requires a careful balance between accuracy, computational speed and number of instructions. This balance, in the Apollo Guidance Computer, leaves very little margin in any of these areas.

Since the values of problem variables are determined by the solution of the problem being solved and since the problem may originate from the ground system, it is essential that the variable range limitations be defined. The conic routines are incapable of handling problems when the solution lies outside of the range.

The following is a list of the maximum allowable numeric values of the variables. Note that, in addition to fundamental quantities such as position and velocity, there are limitations on intermediate variables and combinations of variables.

Scaling for Conic Subroutines (Sections 5. 5. 5 to 5. 5. 10)

<u>Parameter</u>	<u>Maximum Value*</u>	
	<u>Earth Primary Body</u>	<u>Moon Primary Body</u>
r	2 <sup>29</sup>	2 <sup>27</sup>
v	2 <sup>7</sup>	2 <sup>5</sup>
t	2 <sup>28</sup>	2 <sup>28</sup>
$\alpha^{**}$	2 <sup>-22</sup>	2 <sup>-20</sup>
$\alpha_N^{**}$	2 <sup>6</sup>	2 <sup>6</sup>
$p_N$	2 <sup>4</sup>	2 <sup>4</sup>
$\cot \gamma$	2 <sup>5</sup>	2 <sup>5</sup>
$\cot \frac{\theta}{2}$	2 <sup>5</sup>	2 <sup>5</sup>
x	2 <sup>17</sup>	2 <sup>16</sup>
$\xi = \alpha x^2^{***}$	- 50 + 4 $\pi^2$	- 50 + 4 $\pi^2$
$c_1 = \frac{r \cdot v}{\sqrt{\mu}}$	2 <sup>17</sup>	2 <sup>16</sup>
$c_2 = r v^2 / \mu - 1$	2 <sup>6</sup>	2 <sup>6</sup>
$\lambda = r(t_1) / r(t_2)$	2 <sup>7</sup>	2 <sup>7</sup>
$\cos \theta - \lambda$	2 <sup>7</sup>	2 <sup>7</sup>

\* All dimensional values are in units of meters and centiseconds.

\*\* The maximum absolute value occurs for negative values of this parameter.

\*\*\* Both the maximum and minimum values are listed since neither may be exceeded.

<u>Parameter</u>	<u>Maximum Value*</u>	
	<u>Earth</u>	<u>Moon</u>
e	$2^3$	$2^3$
$x^2$	$2^{34}$	$2^{32}$
$x^2 c(\xi)$	$2^{33}$	$2^{31}$
$x^3 s(\xi)/\sqrt{\mu}$	$2^{28}$	$2^{28}$
$c_1 x^2 c(\xi)$	$2^{49}$	$2^{46}$
$c_2 x^2 s(\xi)$	$2^{35}$	$2^{33}$
$x [c_2 x^2 s(\xi) + r(t_1)]$	$2^{49}$	$2^{46}$
$\xi s(\xi)$	$2^7$	$2^7$
$x^2 c(\xi)/r$	$2^8$	$2^8$
$\sqrt{\mu} x (\xi s(\xi) - 1)/r(t_2)$	$2^{15}$	$2^{13}$
$c(\xi)$	$2^4$	$2^4$
$s(\xi)$	$2^1$	$2^1$

\* All dimensional values are in units of meters and centiseconds.

5. 5-6

Revised COLOSSUS

Added GSOP #R-577' PCR # 628 Rev. 3 Date 11-4-68



Nomenclature for Conic Subroutines (Sections 5.5.5 to 5.5.10)

$\underline{r}(t_1)$	initial position vector
$\underline{v}(t_1)$	initial velocity vector
$\underline{r}(t_2)$	terminal position vector
$\underline{v}(t_2)$	terminal velocity vector
$\underline{u}_N$	unit normal in the direction of the angular momentum vector
$\alpha$	reciprocal of semi-major axis (negative for hyperbolas)
$r_P$	radius of pericenter
$r_A$	radius of apocenter
$e$	eccentricity
$\alpha_N$	ratio of magnitude of initial position vector to semi-major axis
$p_N$	ratio of semi-latus rectum to initial position vector magnitude
$\gamma$	inertial flight path angle as measured from vertical
$\theta$	true anomaly difference between $\underline{r}(t_1)$ and $\underline{r}(t_2)$
$f$	true anomaly of $r(t_2)$

x	a universal conic parameter equal to the ratio of eccentric anomaly difference to $\sqrt{+\alpha}$ for the ellipse, or the ratio of the hyperbolic analog of eccentric anomaly difference to $\sqrt{-\alpha}$ for the hyperbola
x'	value of x from the previous Kepler solution
t <sub>21</sub>	computed transfer time from Kepler's equation (t <sub>2</sub> - t <sub>1</sub> )
t' <sub>21</sub>	transfer time corresponding to the previous solution of Kepler's equation
t <sub>D</sub>	desired transfer time through which the conic update of the state vector is to be made
t <sub>D21</sub>	desired transfer time to traverse from $\underline{r}(t_1)$ to $\underline{r}(t_2)$
t <sub>ERR</sub>	error in transfer time
ε <sub>t</sub>	fraction of desired transfer time to which t <sub>ERR</sub> must converge
Δx	increment in x which will produce a smaller t <sub>ERR</sub>
ε <sub>x</sub>	value of Δx which will produce no significant change in t <sub>21</sub>
Δcot γ	increment in cot γ which will decrease the magnitude of t <sub>ERR</sub>
ε <sub>c</sub>	value of Δcot γ which will produce no significant change in t <sub>21</sub>

$\mu$	product of universal gravitational constant and mass of the primary attracting body
$x_{MAX}$	maximum value of x
$x_{MIN}$	minimum value of x
$\cot_{MAX}$	maximum value of $\cot \gamma$
$\cot_{MIN}$	minimum value of $\cot \gamma$
$\ell_{MAX}$	upper bound of general independent variable
$\ell_{MIN}$	lower bound of general independent variable
$x_{MAX 1}$	absolute upper bound on x with respect to the moon
$x_{MAX 0}$	absolute upper bound on x with respect to the earth
k	a fraction of the full range of the independent variable which determines the increment of the independent variable on the first pass through the iterator
y	general dependent variable
y'	previous value of y
$y_{ERR}$	error in y
z	general independent variable

$\Delta z$	increment in z which will produce a smaller $y_{ERR}$
$s_G$	a sign which is plus or minus according to whether the true anomaly difference between $r(t_1)$ and $r(t_2)$ is to be less than or greater than 180 degrees
$s_r$	a sign which is plus or minus according to whether the desired radial velocity at $r(t_2)$ is plus or minus
$\eta_1$	general vector # 1
$\eta_2$	general vector # 2
$\phi$	angle between $\eta_1$ and $\eta_2$
$f_1$	a switch set to 0 or 1 according to whether a guess of $\cot \gamma$ is available or not
$f_2$	a switch set to 0 or 1 according to whether Lambert should determine $\underline{u}_N$ from $r(t_1)$ and $r(t_2)$ or $\underline{u}_N$ is an input
$f_3$	a tag set to 0 or 1 according to whether the iterator should use the "Regula Falsi" or bias method
$f_4$	a flag set to 0 or 1 according to whether the iterator is to act as a first order or a second order iterator
$f_5$	a flag set to 0 or 1 according to whether Lambert converges to a solution or not

5.5-10

- $f_6$  a switch set to 0 or 1 according to whether or not the new state vector is to be an additional output requirement of the Time-Theta or Time-Radius problems.
- $f_7$  a flag set to 1 if the inputs require that the conic trajectory must close through infinity
- $f_8$  a flag set to 1 if the Time-Radius problem was solved for pericenter or apocenter instead of  $r(t_2)$
- $f_9$  a flag set to 1 if the input to the Time-Radius Subroutine produced an  $e$  less than  $2^{-18}$ .
- $t_p$  period of the orbit
- $t_R$  integral periods subtracted from  $t_D$  to produce a  $t_D$  less than  $t_p$
- $x_R$  value of  $x$  corresponding to  $t_R$
- $k_1$  the minimal acceptance percentage of  $t_{D21}$  to which  $t_{ERR}$  must converge
- $n_1$  a flag set to 0 or 1 according to whether or not the velocity vector at the terminal position is to be an additional output requirement of the Lambert Subroutine

### 5. 5. 2 PLANETARY INERTIAL ORIENTATION SUBROUTINE

This subroutine is used to transform vectors between the Basic Reference Coordinate System and a Planetary (Earth-fixed or Moon-fixed) Coordinate System at a specified time. These three coordinate systems are defined in Section 5. 1. 4.

Let  $\underline{r}$  be a vector in the Basic Reference Coordinate System,  $\underline{r}_P$  the same vector expressed in the Planetary Coordinate System, and  $t$  the specified ground elapsed time (GET). Then,

$$\underline{r}_P = M(t) (\underline{r} - \underline{\ell} \times \underline{r}) \quad (5. 2. 1)$$

and

$$\underline{r} = M^T(t) (\underline{r}_P + \underline{\ell}_P \times \underline{r}_P) \quad (5. 2. 2)$$

where  $M(t)$  is a time dependent orthogonal transformation matrix,  $\underline{\ell}$  is a small rotation vector in the Basic Reference Coordinate System, and  $\underline{\ell}_P$  is the same vector  $\underline{\ell}$  expressed in the Planetary Coordinate System. The vector  $\underline{\ell}$  is considered constant in one coordinate system for the duration of the mission. The method of computing  $M(t)$  and  $\underline{\ell}$  depends on whether the relevant planet is the earth or the moon.

#### Case I - Earth

For the earth, the matrix  $M(t)$  describes a rotation about the polar axis of the earth (the Z-axis of the Earth-fixed Coordinate

System), and the vector  $\underline{\ell}$  accounts for the precession and nutation of the polar axis (the deviation of the true pole from the mean pole).

Let  $A_X$  and  $A_Y$  be the small angles about the X- and Y-axes of the Basic Reference Coordinate System, respectively, that describe the precession and nutation of the earth's polar axis. The values of these two angles at the midpoint of the mission are included in the pre-launch erasable data load and are considered constant throughout the flight. Then,

$$\underline{\ell} = \begin{pmatrix} A_X \\ A_Y \\ 0 \end{pmatrix}$$

$$A_Z = A_{Z0} + \omega_E (t + t_0) \quad (5.2.3)$$

$$M(t) = \begin{pmatrix} \cos A_Z & \sin A_Z & 0 \\ -\sin A_Z & \cos A_Z & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\underline{\ell}_P = M(t) \underline{\ell}$$

where  $A_{Z0}$  is the angle between the X-axis of the Basic Reference Coordinate System and the X-axis of the Earth-fixed Coordinate System (the intersection of the Greenwich meridian and the equatorial plane of the earth) at July 1.0, 1968 universal time (i. e., midnight at Greenwich just prior to July 1, 1968),  $t_0$  is the elapsed time between July 1.0, 1968 universal time and the time that the computer clock was zeroed, and  $\omega_E$  is the angular velocity of the earth.

## Case II - Moon

For the moon, the matrix  $M(t)$  accounts for the difference in orientation of the Basic Reference and Moon-fixed Coordinate Systems in exact accordance with Cassini's laws, and the rotation vector  $\underline{\ell}$  corrects for deviations from the above orientation because of physical libration.

Define the following three angles which are functions of time:

- B = the obliquity, the angle between the mean earth equatorial plane and the plane of the ecliptic.
- $\Omega_1$  = the longitude of the node of the moon's orbit measured from the X-axis of the Basic Reference Coordinate System.
- F = the angle from the mean ascending node of the moon's orbit to the mean moon.

Let  $I$  be the constant angle between the mean lunar equatorial plane and the plane of the ecliptic ( $1^\circ 32.1'$ ). Then, the sequence of rotations which brings the Basic Reference Coordinate System into coincidence with the Moon-fixed Coordinate System (neglecting libration) is as follows:



<u>Rotation</u>	<u>Axis of Rotation</u>	<u>Angle of Rotation</u>
1	X	B
2	Z	$\Omega_I$
3	X	-I
4	Z	$\pi + F$

The transformation matrices for these rotations are, respectively,

$$M_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos B & \sin B \\ 0 & -\sin B & \cos B \end{pmatrix}$$

$$M_2 = \begin{pmatrix} \cos \Omega_I & \sin \Omega_I & 0 \\ -\sin \Omega_I & \cos \Omega_I & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$M_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos I & -\sin I \\ 0 & \sin I & \cos I \end{pmatrix}$$

$$M_4 = \begin{pmatrix} -\cos F & -\sin F & 0 \\ \sin F & -\cos F & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

(5.2.4)

The matrix  $M(t)$  is then given by

$$M(t) = M_4 M_3 M_2 M_1 \quad (5.2.5)$$

The following approximate method is used to determine the transformation between the Basic Reference and Moon-fixed Coordinate Systems.

The angles  $B$ ,  $\Omega_I$  and  $F$  are computed as linear functions of time. Let  $\underline{\ell}_M$  be the value of the vector libration  $\underline{\ell}_P$  (expressed in the Moon-fixed Coordinate System) at the midpoint of the mission. The vector  $\underline{\ell}_M$  is included in the pre-launch erasable data load and is considered constant throughout the flight. Then,

$$\begin{aligned} \underline{\ell}_P &= \underline{\ell}_M \\ t_M &= t + t_0 \\ B &= B_0 + \dot{B} t_M \\ \Omega_I &= \Omega_{I0} + \dot{\Omega}_I t_M \\ F &= F_0 + \dot{F} t_M \end{aligned}$$

$$\underline{a} = \begin{pmatrix} \cos \Omega_I \\ \cos B \sin \Omega_I \\ \sin B \sin \Omega_I \end{pmatrix} \quad (5.2.6)$$

$$\underline{b} = \begin{pmatrix} -\sin \Omega_I \\ \cos B \cos \Omega_I \\ \sin B \cos \Omega_I \end{pmatrix}$$

$$\underline{c} = \begin{pmatrix} 0 \\ -\sin B \\ \cos B \end{pmatrix}$$

$$\underline{d} = \underline{b} C_I - \underline{c} S_I$$

$$\underline{m}_2 = \underline{b} S_I + \underline{c} C_I$$

$$\underline{m}_0 = -\underline{a} \cos F - \underline{d} \sin F \quad (5.2.6)$$

(cont.)

$$\underline{m}_1 = \underline{a} \sin F - \underline{d} \cos F$$

$$M(t) = \begin{pmatrix} \underline{m}_0^T \\ \underline{m}_1^T \\ \underline{m}_2^T \end{pmatrix}$$

$$\underline{\ell} = M^T(t) \underline{\ell}_P$$

where  $B_0$ ,  $\Omega_{I0}$ , and  $F_0$  are the values of the angles  $B$ ,  $\Omega_I$  and  $F$ , respectively, at July 1.0, 1968 universal time;  $\dot{B}$ ,  $\dot{\Omega}_I$  and  $\dot{F}$  are the rates of change of these angles; and  $C_I$  and  $S_I$  are the cosine and sine, respectively, of the angle  $I$ . Time  $t_M$  is defined in Section 5.5.4.

•  
•  
•

### 5. 5. 3 LATITUDE-LONGITUDE SUBROUTINE

For display and data load purposes, the latitude, longitude, and altitude of a point near the surface of the earth or the moon are more meaningful and more convenient to use than the components of a position vector. This subroutine is used to transform position vectors between the Basic Reference Coordinate System and Geographic or Selenographic latitude, longitude, altitude at a specified time.

In the case of the moon, the altitude is computed above either the landing site radius,  $r_{LS}$ , or the mean lunar radius,  $r_M$ . For the earth, the altitude is defined with respect to either the launch pad radius,  $r_{LP}$  or the radius of the Fischer ellipsoid,  $r_F$ , which is computed from

$$r_F^2 = \frac{b^2}{1 - \left(1 - \frac{b^2}{a^2}\right) (1 - \text{SINL}^2)} \quad (5. 3. 1)$$

where  $a$  and  $b$  are the semi-major and semi-minor axes of the Fischer ellipsoid, respectively, and  $\text{SINL}$  is the sine of the geocentric latitude.

The computational procedures are illustrated in Figs. 5. 3-1, 5. 3-2, and 5. 3-3. The calling program must specify either a vector  $\underline{r}$  or latitude (Lat), longitude (Long), and altitude (Alt). In addition, the program must set the time  $t$  and the two indicators  $P$  and  $F$  where

$$P = \begin{cases} 0 & \text{for earth} \\ 1 & \text{for moon} \end{cases}$$

$$F = \begin{cases} 1 & \text{for Fischer ellipsoid or mean lunar radius} \\ 0 & \text{for launch pad or landing site radius} \end{cases}$$

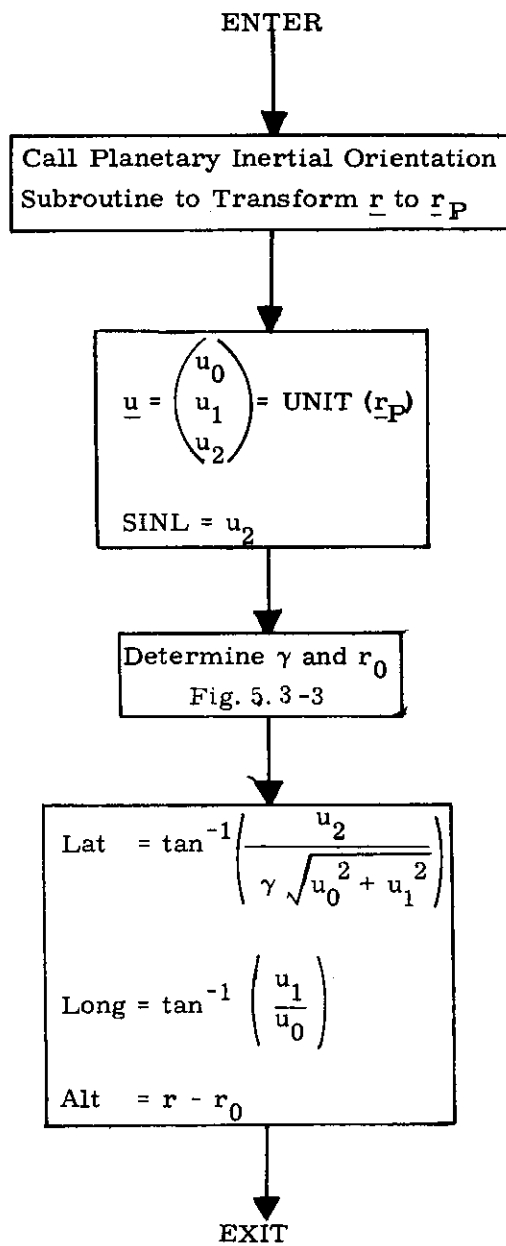


Fig. 5.3-1 Vector to Latitude, Longitude, Altitude Computation Logic Diagram

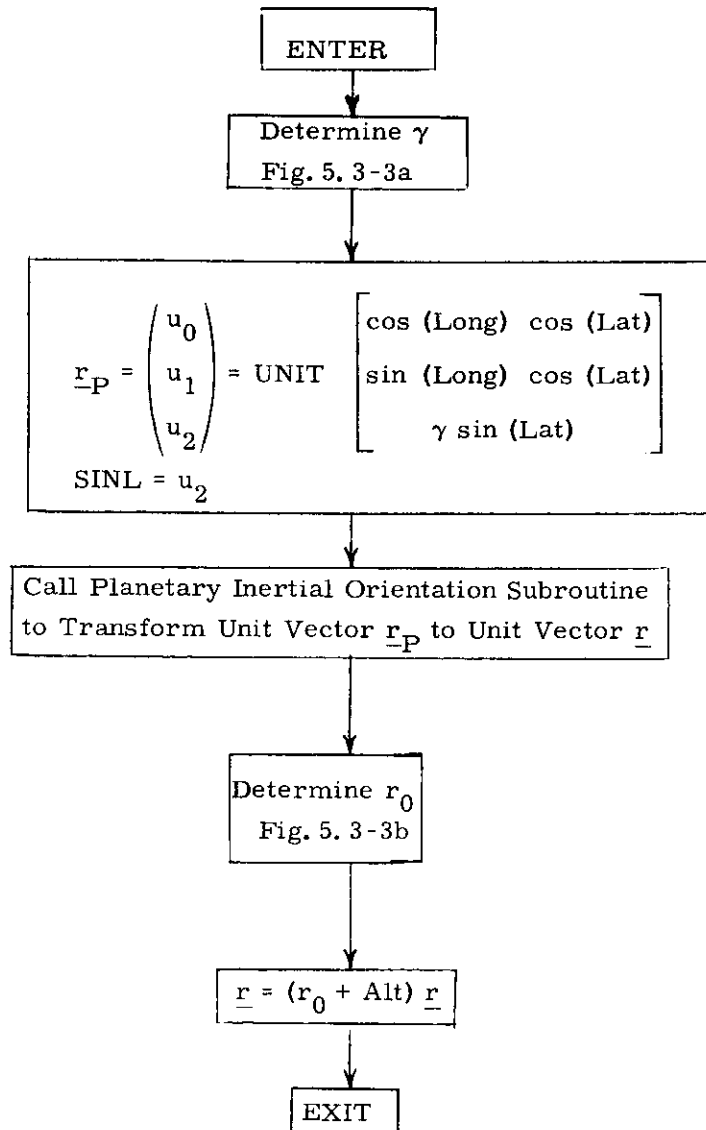


Fig. 5.3-2 Latitude, Longitude, Altitude to Vector Computation Logic Diagram

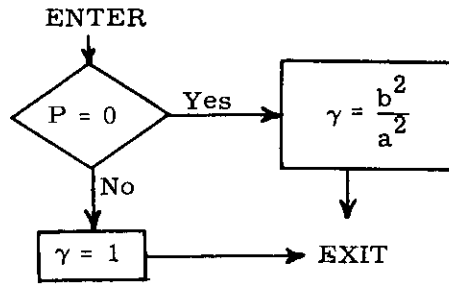


Figure 5.3-3a Determination of  $\gamma$

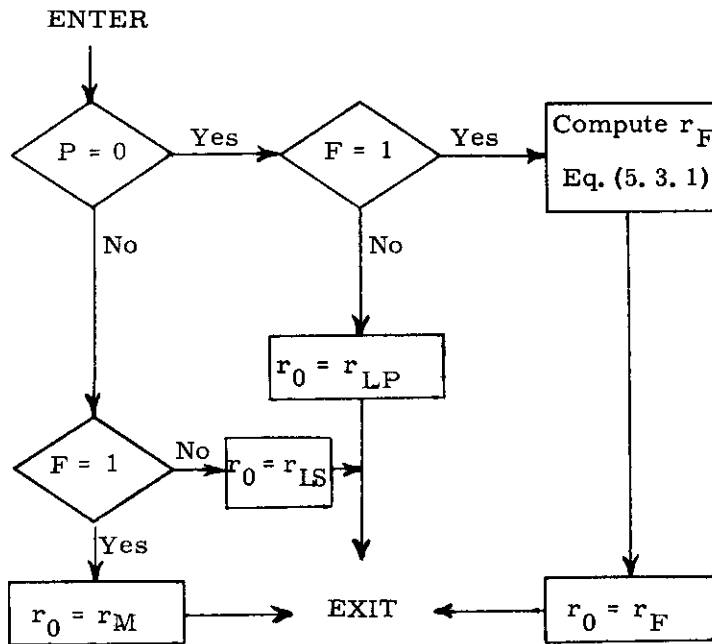


Figure 5.3-3b Determination of  $r_0$



#### 5. 5. 4 LUNAR AND SOLAR EPHEMERIDES

This subroutine is used to determine the position and velocity vectors of the sun and the moon relative to the earth. The position vectors of the moon and the sun are needed by the Coasting Integration Routine to compute gravity perturbations (Section 5. 2. 2. 3). The velocity of the moon is used by the Coasting Integration Routine when a change in the origin of the coordinate system is performed at the sphere of influence of the moon (Fig. 2. 2-3). The velocity of the sun is required, but not very accurately, to compute aberration corrections to optical sightings.

The position of the moon is stored in the computer in the form of a ninth-degree polynomial approximation which is valid over a 14.5 day interval beginning at noon ephemeris time on the day of the launch. The following parameters are included in the pre-launch erasable data load:

- $t_{M0}$  = the elapsed time between July 1.0, 1968 universal time and the time at the center of the range over which the lunar-position polynomial is valid. The value of  $t_{M0}$  will be an integral number of quarter days minus the difference between ephemeris time and universal time.
- $c_0$  to  $c_9$  = vector coefficients

Let  $t$  be the specified ground elapsed time (GET), and  $t_0$  be the elapsed time between July 1.0, 1968 universal time and the time that the computer clock was zeroed. Then, the approximate position and velocity of the moon are computed from

$$t_M = t + t_0 - t_{M0} \quad (5. 4. 1)$$

$$\underline{r}_{EM} = \sum_{i=0}^9 c_i t_M^i \quad (5.4.2)$$

$$\underline{v}_{EM} = \sum_{i=1}^9 i c_i t_M^{i-1} \quad (5.4.3)$$

The approximate position and velocity of the sun are computed from the following items which are included in the pre-launch erasable data load:

$\underline{r}_{ES0}$ ,  $\underline{v}_{ES0}$  = the position and velocity vectors of the sun relative to the earth at time  $t_{M0}$ .

$\omega_{ES}$  = the angular velocity of the vector  $\underline{r}_{ES0}$  at time  $t_{M0}$

Then,

$$\begin{aligned} \underline{r}_{ES} &= \underline{r}_{ES0} \cos (\omega_{ES} t_M) \\ &+ \left[ \underline{r}_{ES0} \times \text{UNIT} (\underline{v}_{ES0} \times \underline{r}_{ES0}) \right] \sin (\omega_{ES} t_M) \end{aligned} \quad (5.4.4)$$

$$\underline{v}_{ES} = \underline{v}_{ES0} \quad (5.4.5)$$

### 5.5.5 KEPLER SUBROUTINE

The Kepler Subroutine solves for the two body position and velocity vectors at the terminal position given the initial position and velocity vectors and a transfer time to the terminal position.

This section contains information to aid the reader in understanding the less obvious aspects of the Kepler Subroutine block diagram depicted in Figs. 5.5-1 thru 5.5-3. The subroutines referred to in these figures are presented in Section 5.5.10. Nomenclature is found in Section 5.5.1.2.

Prior to entering the Kepler Subroutine an initial estimate of  $x$  can be generated via Eq. (2.2.4) of Section 5.2.2.2 with  $\frac{\Delta t}{2} = t_D - t_{21}$  and  $\tau = t_D$ . However,  $x'$  and  $t_{21}$  are non-zero only if the subroutine is being used repetitively.

Although, theoretically, there is no upper bound on  $x$ , the practical bound is set to  $x_{MAX0}$  or  $x_{MAX1}$  to eliminate non-feasible trajectories and increase the accuracy to which  $x$  can be computed. In addition,  $\alpha x^2$  has a practical range of  $-50 < \alpha x^2 < (2\pi)^2$  which determines an independent upper bound on  $x$ . The  $x_{MAX}$  used, then, corresponds to the smaller of the two values.

The transfer time convergence criterion is approximately the same as the granularity of the time input. Since, for some of the problems to be solved, the sensitivity of time to  $x$  is so large that the granularity in  $x$ ,  $\epsilon_x$ , produces a change in time which exceeds the granularity in time, it is necessary to introduce  $\epsilon_x$  as a redundant convergence criterion.

The Kepler Subroutine, provided the parameter range constraints are satisfied will always produce a solution.

A negative value of  $t_D$  will cause the subroutine to update the state vector backward in time. If  $t_D$  is negative but its absolute value is greater than one period, the solution returned will correspond to an update backward of exactly one period. If  $t_D$  is positive but greater than one period the update will correspond to the true, total value of  $t_D$ , just as it does for  $t_D$  less than one period.

5. 5. 26

Rev. 3 - 11/68

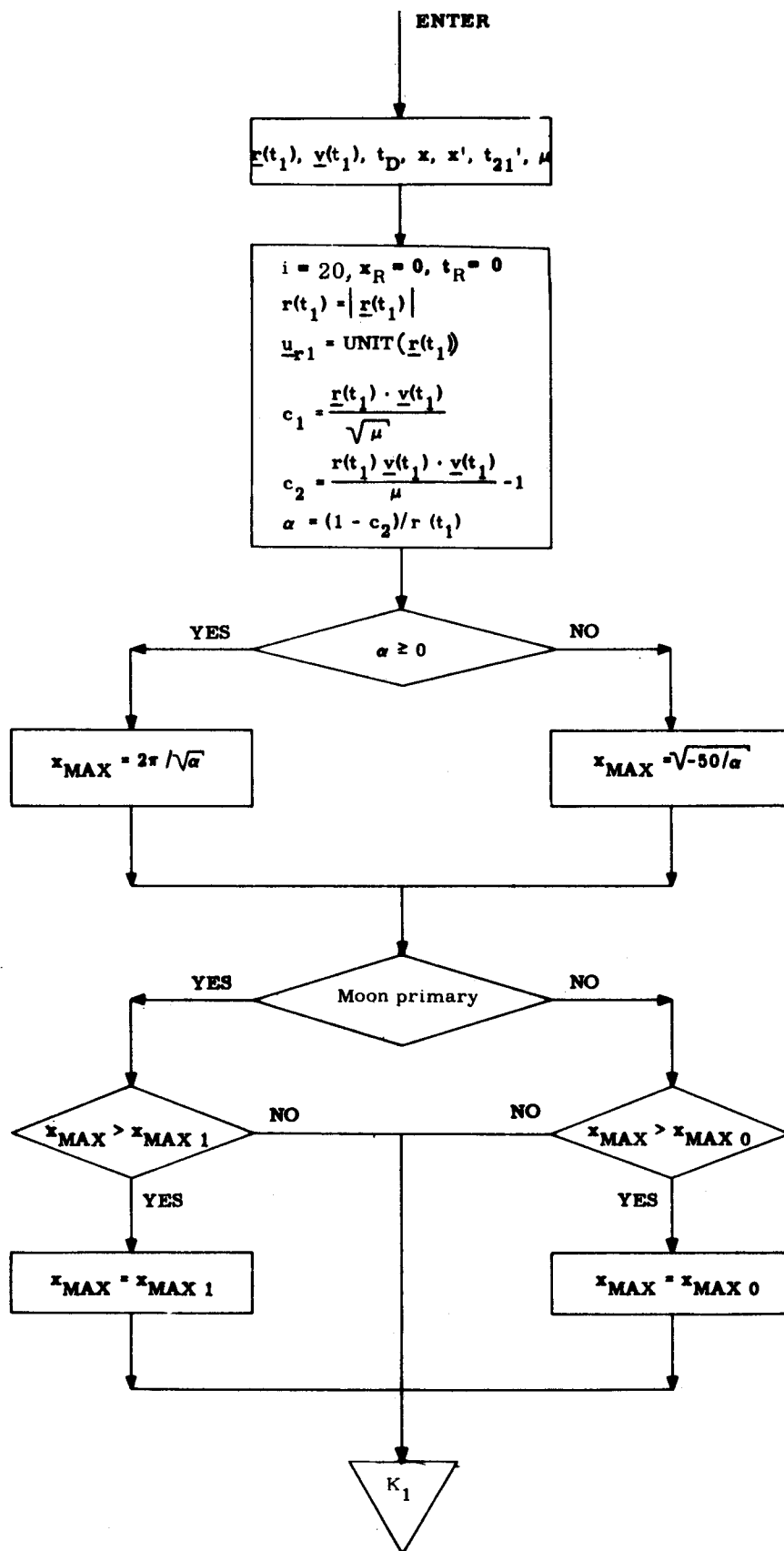


Figure 5.5-1 Kepler Subroutine

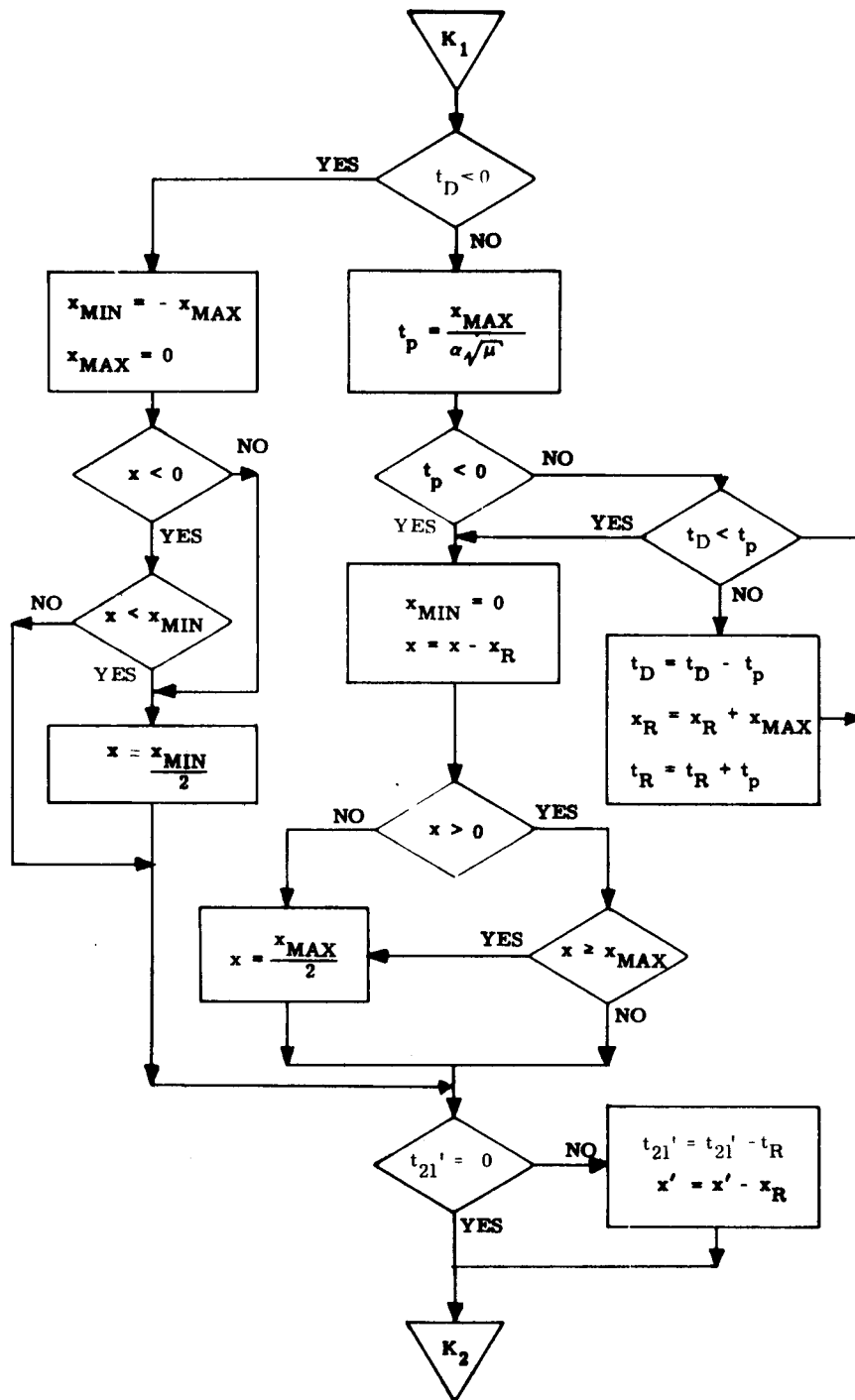


Figure 5.5-2 Kepler Subroutine

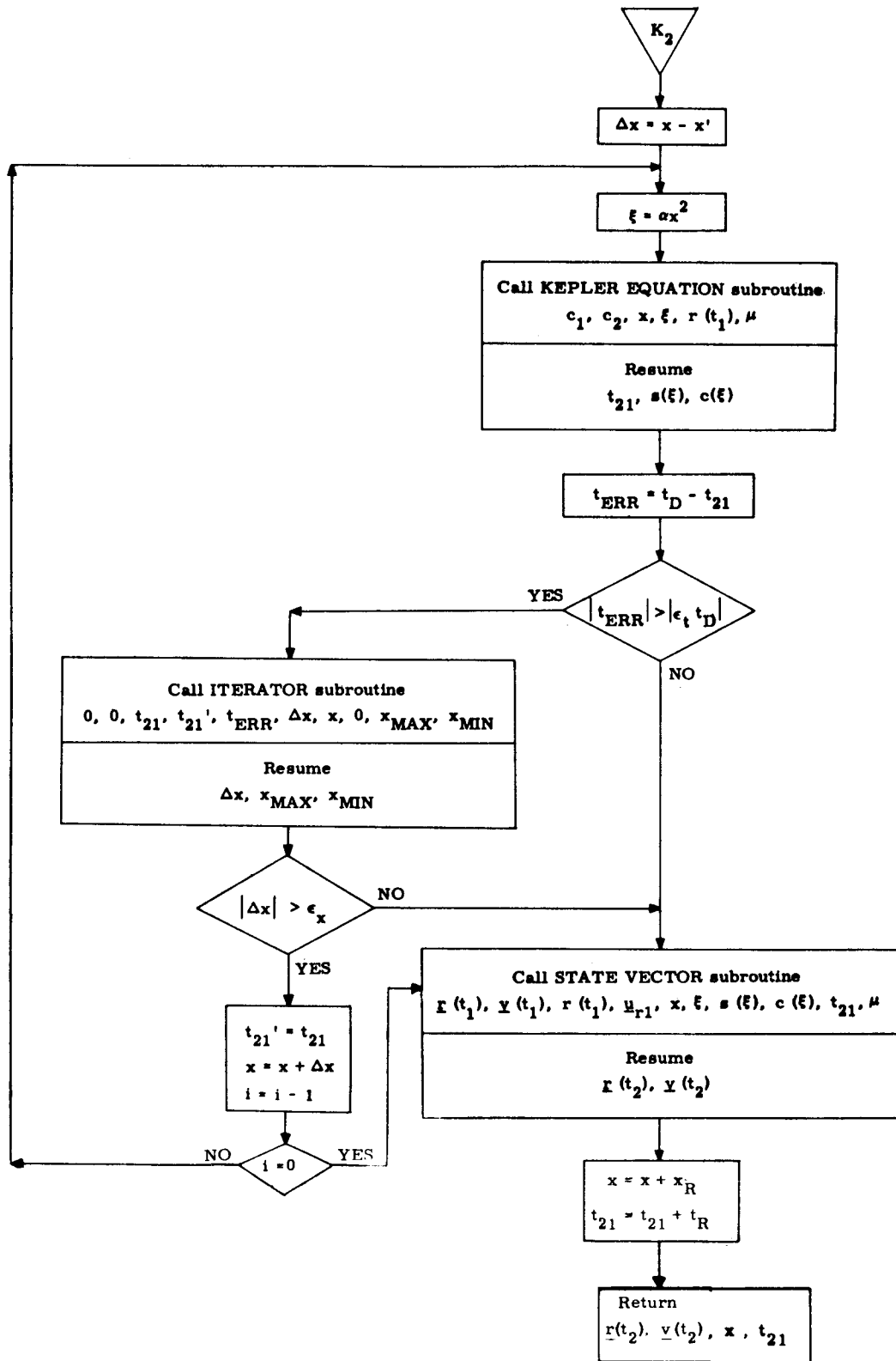


Figure 5.5-3 Kepler Subroutine

## 5. 5. 6     LAMBERT SUBROUTINE

The Lambert Subroutine solves for the two body initial velocity vector given the initial and terminal position vectors and a transfer time between the two.

This section contains information to aid the reader in understanding the less obvious aspects of the Lambert Subroutine block diagrams depicted in Figs. 5. 6-1 and 5. 6-2. The subroutines referred to in these figures are presented in Section 5. 5. 10 and the nomenclature is found in Section 5. 5. 1. 2.

If the Lambert Subroutine is used repetitively and rapid computation is required, the previous value of the independent variable,  $\cot \gamma$ , can be used as a starting point for the new iteration. Flag  $f_1$  provides this option.

The Lambert Subroutine computes the normal to the trajectory,  $\underline{u}_N$ , using the two input position vectors. If these vectors are nearly colinear, it is desirable to specify the normal as an input rather than rely on the ill-defined normal based on the two input position vectors. Flag  $f_2$  provides this option. The presence of the inputs in parentheses, therefore, is contingent upon the setting of these flags.

The theoretical bounds on the independent variable,  $\cot \gamma$ , correspond to the infinite energy hyperbolic path and the parabolic path which closes through infinity. These bounds are dynamically reset by the iterator to provide a more efficient iteration scheme. In addition, if during the course of the iteration,  $\cot \gamma$  causes a parameter of the problem to exceed its maximum as determined by its allowable range, the appropriate bound is reset and the iterator continues trying to find an acceptable solution. (This logic does not appear in Figs. 5. 6-1 and 2



as it is pertinent only to fixed-point programming). If no acceptable solution is reached, the transfer time input was too small to produce a practical trajectory between the input position vectors. When this happens,  $\Delta \cot \gamma$  approaches its granularity limit  $\epsilon_c$  before time converges to within a fraction  $\epsilon_t$  of the desired time. However, this same granularity condition exists when the sensitivity problem described in the Kepler Subroutine, Section 5.5.5., occurs. In this case an acceptable solution does exist. This dual situation is resolved via a third convergence criterion. If the error in transfer time is greater than the usual fraction  $\epsilon_t$  of the desired transfer time, but still less than a slightly larger fraction  $k_1$  of the desired transfer time and  $\Delta \cot \gamma$  is less than  $\epsilon_c$ , then the solution is deemed acceptable and the required velocity is computed.

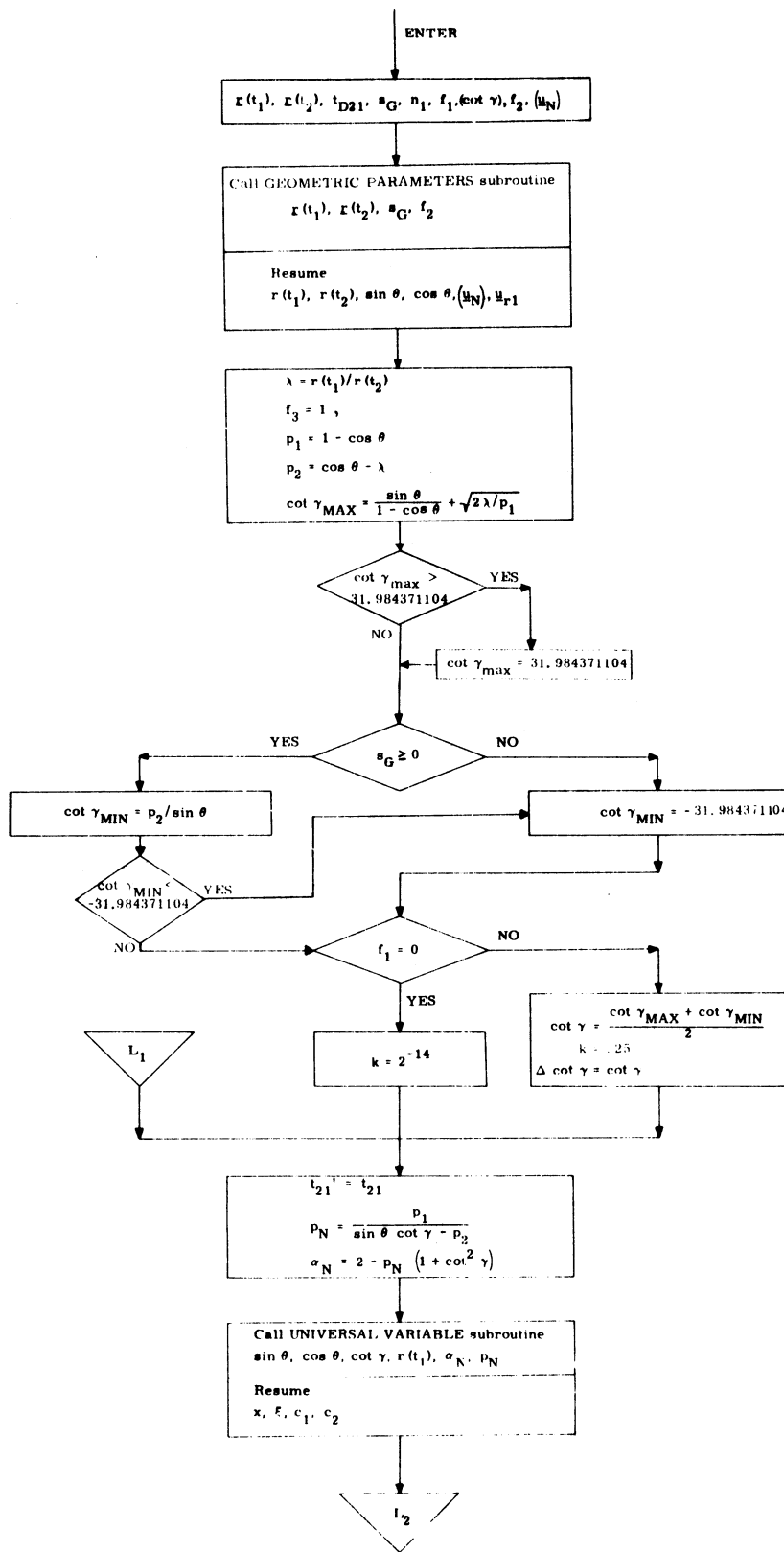


Figure 5.6-1 Lambert Subroutine

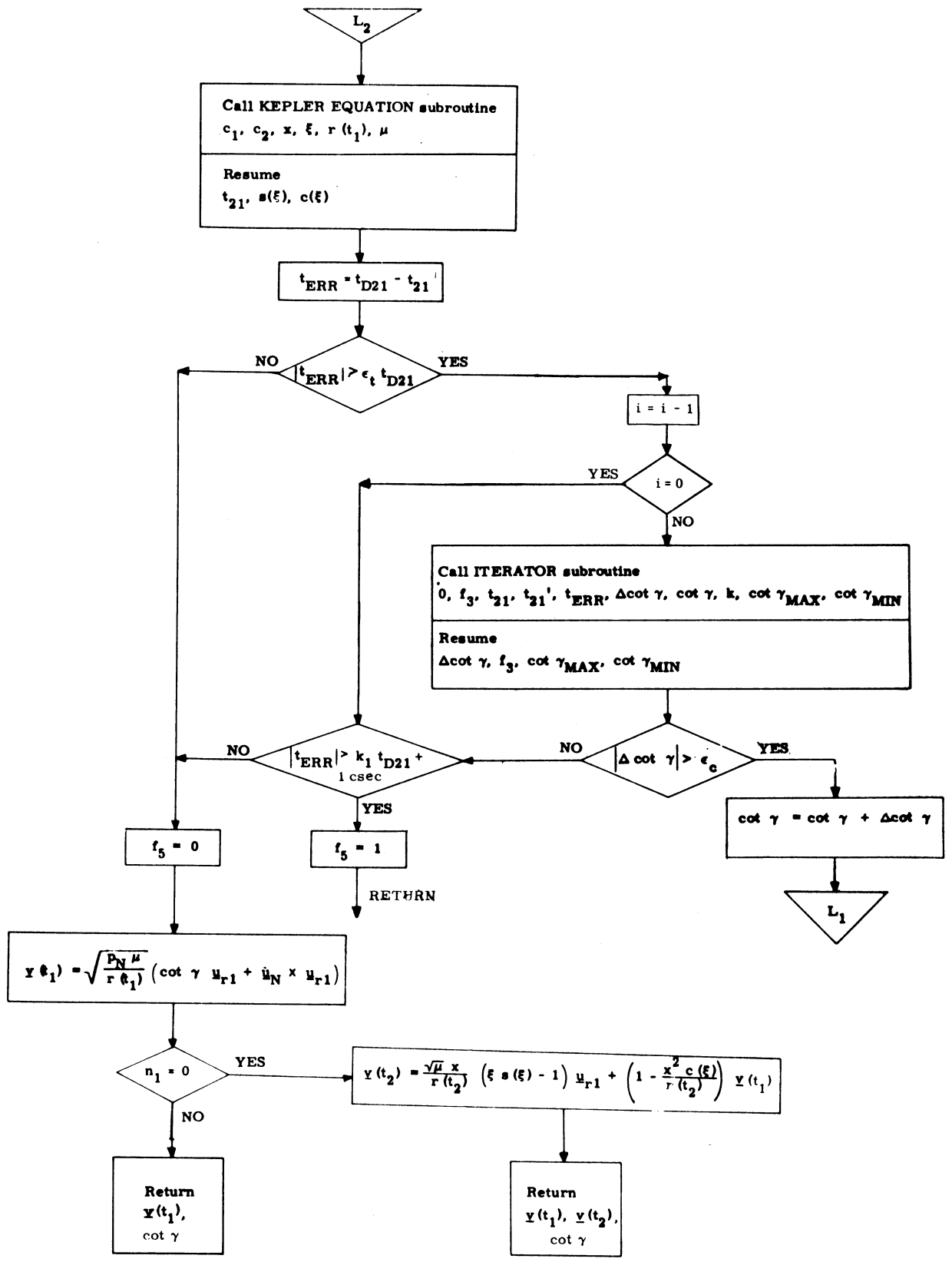


Figure 5.6-2 Lambert Subroutine

### 5. 5. 7 TIME-THETA SUBROUTINE

The Time-Theta Subroutine solves for the two body transfer time given the initial position and velocity vectors and the true anomaly difference (transfer angle) to the terminal position.

This section contains information to aid the reader in understanding the less obvious aspects of the Time-Theta Subroutine block diagram depicted in Fig. 5.7-1. The subroutines referred to in this figure are presented in Section 5.5.10 and the nomenclature is found in Section 5.5.1.2.

The flag  $f_6$  must be zero if the user desires computation of the terminal state vector in addition to the transfer time.

If the conic trajectory is a parabola or hyperbola and the desired transfer angle,  $\theta$ , lies beyond the asymptote of the conic,  $f_7$  will be set indicating that no solution is possible.

In addition to the parameter range constraints imposed on Kepler's equation, the additional restriction on Time-Theta that the trajectory must not be near rectilinear is indicated by the range of  $\cot \gamma$ .

The Time-Theta problem is not well defined for near rectilinear trajectories, i. e. the transfer angle  $\theta$  is no longer a meaningful problem parameter. This will not cause difficulties provided the input variables are within the specified ranges.

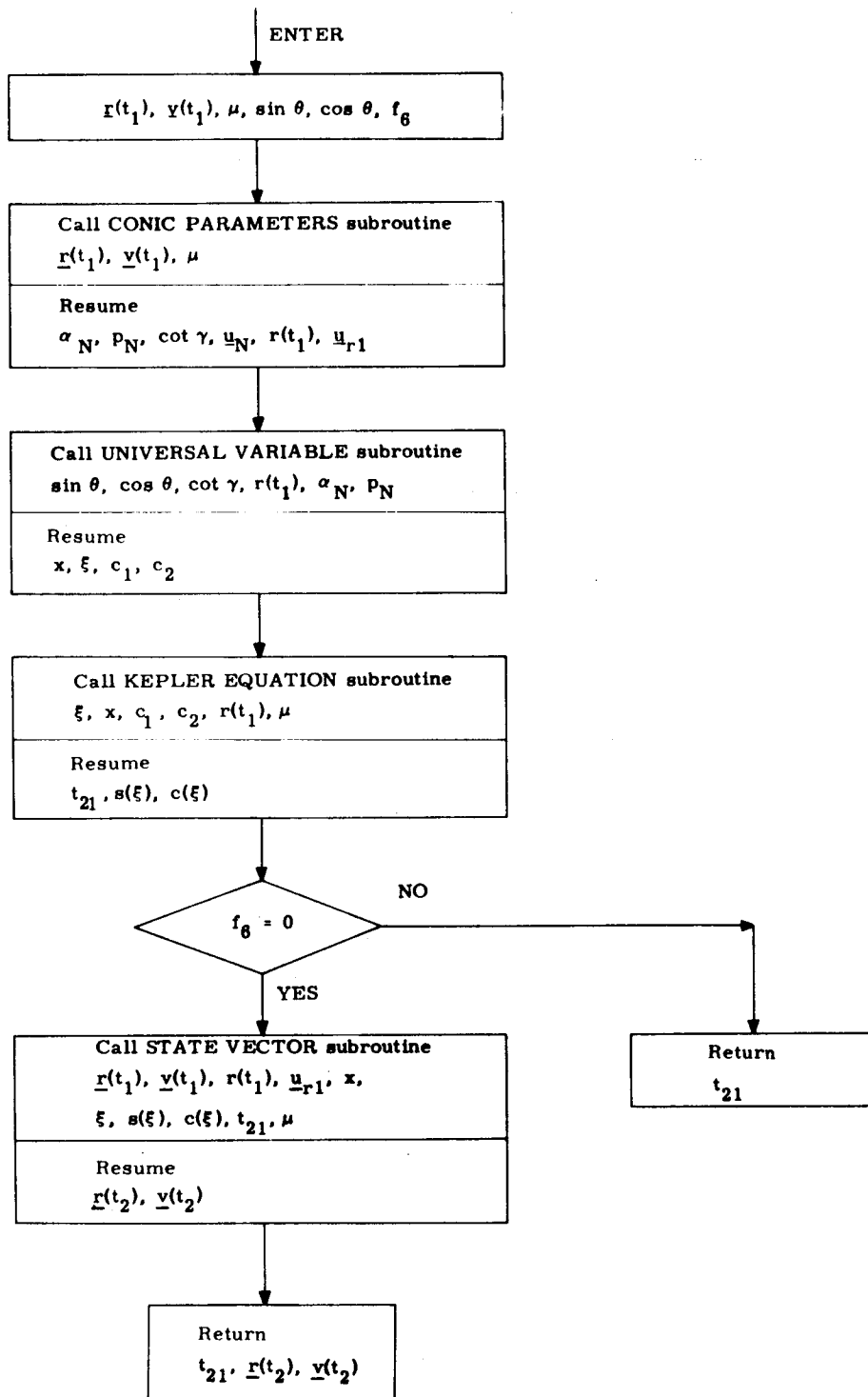


Figure 5.7-1 Time-Theta Subroutine

### 5.5.8 TIME-RADIUS SUBROUTINE

The Time-Radius Subroutine solves for the two body transfer time to a specified radius given in the initial position and velocity vectors and the radius magnitude.

This section contains information to aid the reader in understanding the less obvious aspects of the Time-Radius Subroutine block diagrams depicted in Figs. 5.8-1 and 5.8-2. The subroutines referred to in this figure are presented in Section 5.5.10 and the nomenclature is found in Section 5.5.1.2.

Paragraphs 3, 4 and 5 of Section 5.5.7 apply to the Time-Radius Subroutine as well.

Since an inherent singularity is present for the circular orbit case, near-circular orbits result in a loss of accuracy in computing both the transfer time,  $t_{21}$ , and the final state vector. This is caused by the increasing sensitivity of  $t_{21}$  to  $r(t_2)$  as the circular orbit is approached. In the extreme case that eccentricity is less than  $2^{-18}$ , the problem is undefined and the subroutine will exit without a solution, setting flag  $f_9$  to indicate this.

If  $r(t_2)$  is less than the radius of pericenter or greater than the radius of apocenter, then  $r(t_2)$  will be ignored and the pericenter or apocenter solution, respectively, will be computed. A flag,  $f_8$ , will be set to indicate this.

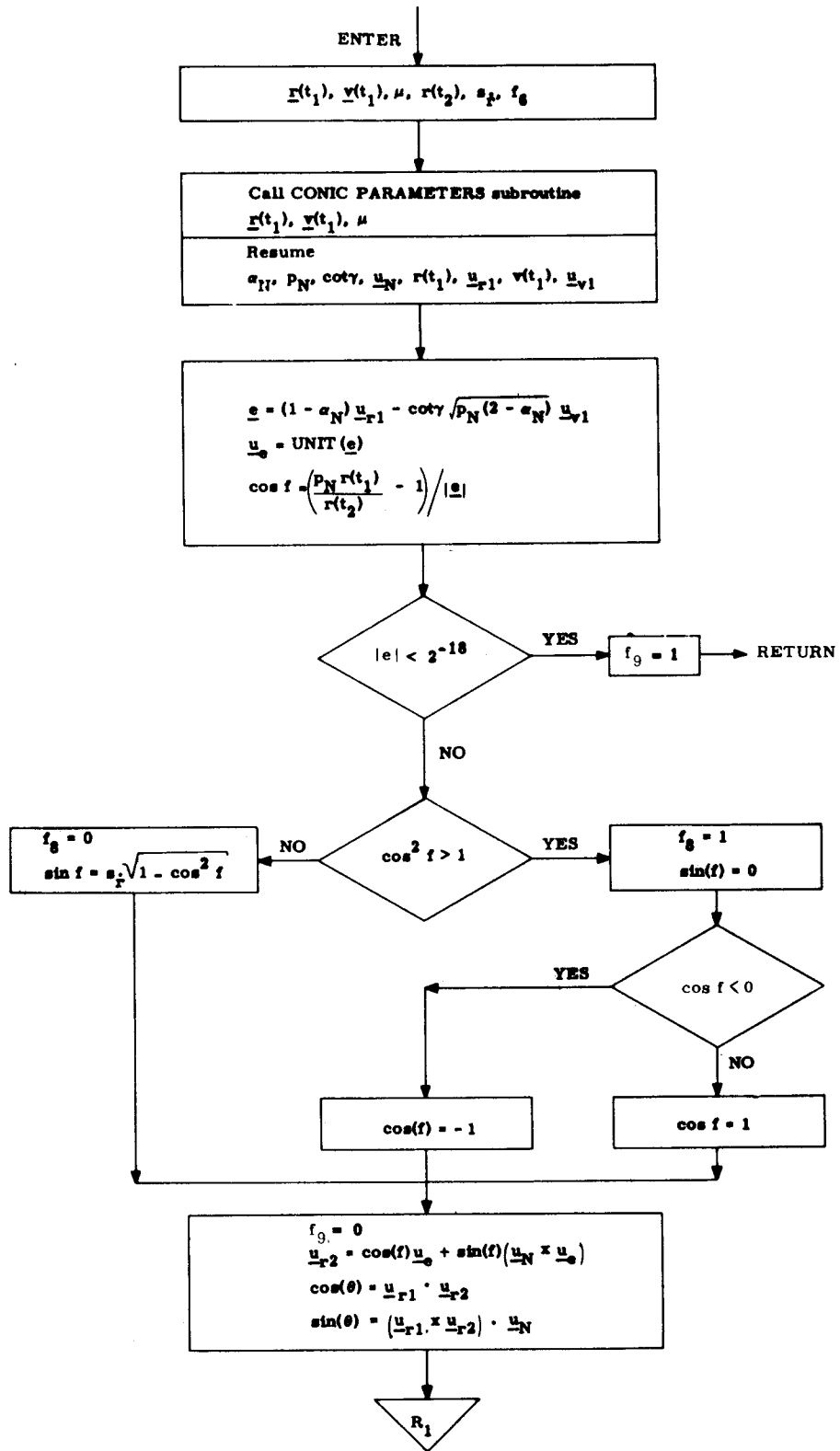


Figure 5.8-1 Time-Radius Subroutine

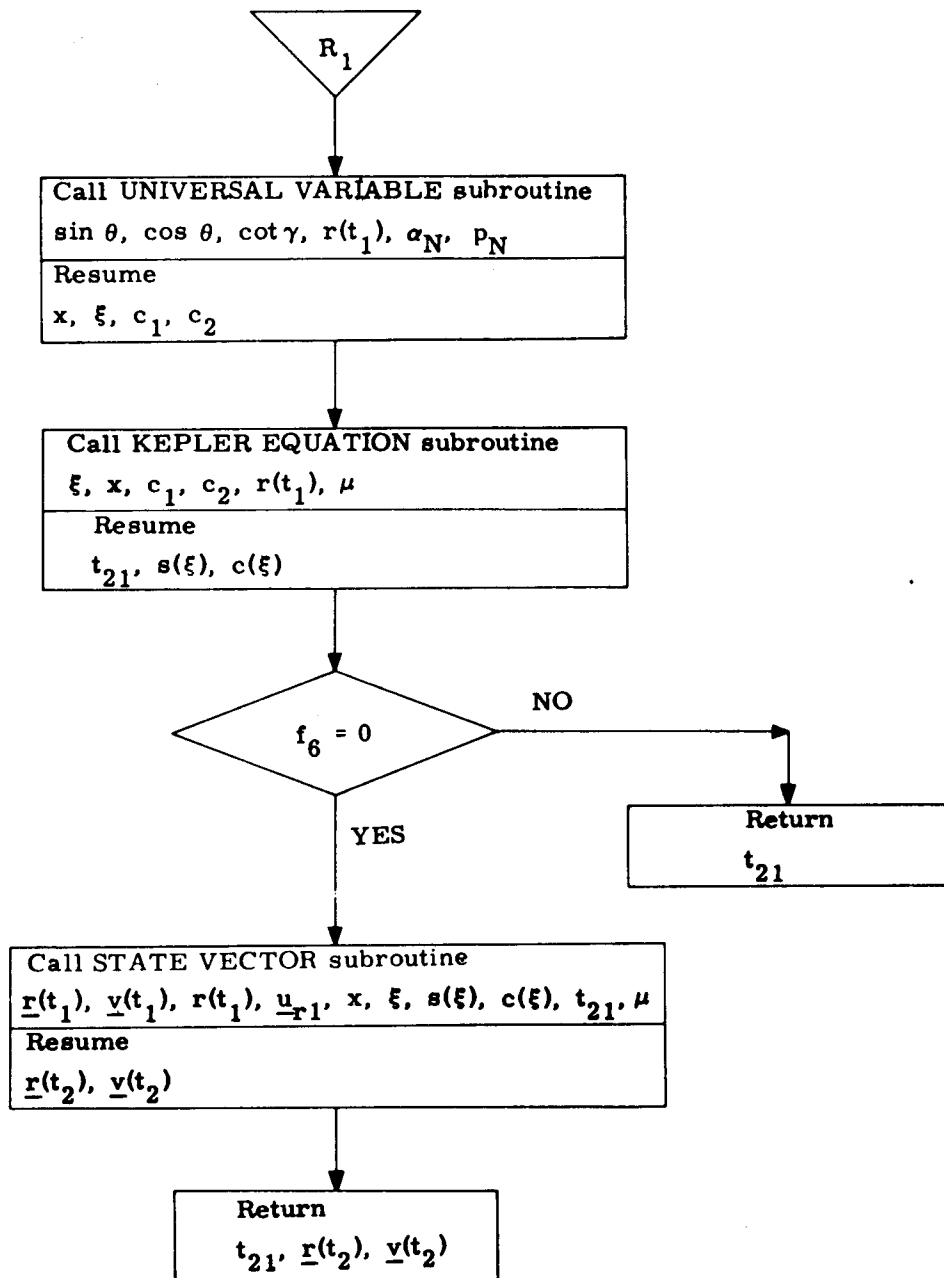


Figure 5.8-2 Time-Radius Subroutine



### 5.5.9 APSIDES SUBROUTINE

The Apsides Subroutine solves for the two body radii of apocenter and pericenter and the eccentricity of the trajectory given the position and velocity vectors for a point on the trajectory.

This subroutine is depicted in Fig. 5.9-1. The subroutines referred to in this figure are presented in Section 5.5.10. Nomenclature is found in Section 5.5.1.2.

It is characteristic of this computation that the apsides become undefined as the conic approaches a circle. This is manifested by decreasing accuracy. When the conic is nearly parabolic, or hyperbolic, the radius of apocenter is not defined. In this event the radius of apocenter will be set to the maximum positive value allowed by the computer.

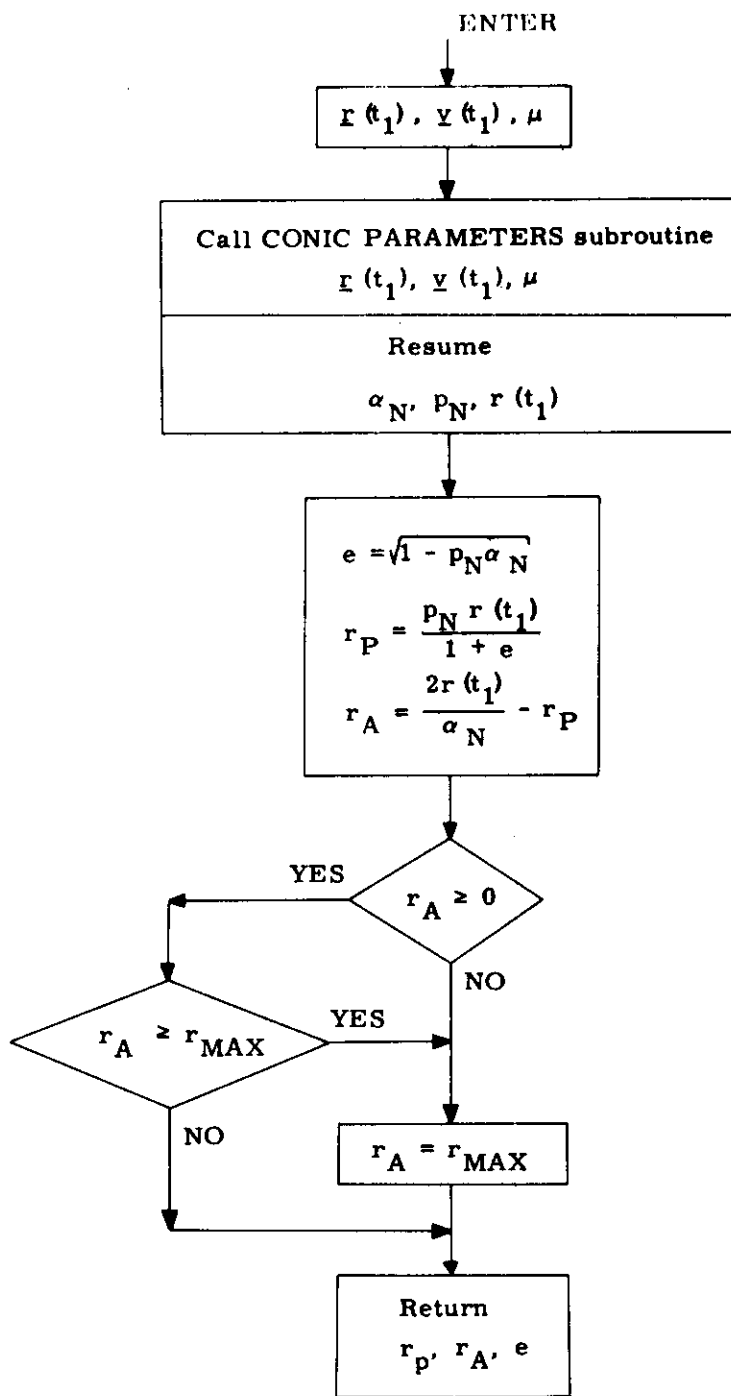


Figure 5.9-1 Apsides Subroutine

#### 5. 5. 10 MISCELLANEOUS SUBROUTINES

There are, as part of the Conic Trajectory Subroutines, three subroutines which are useful in their own right. These are the Conic Parameters, the Geometric Parameters and the Iterator Subroutines which are depicted in Figs. 5. 10-1, 5. 10-2 and 5. 10-3, respectively.

The Conic Parameters and Geometric Parameters Subroutines are self explanatory.

The Iterator Subroutine serves several purposes. It is used when flag  $f_4$  is set to zero to solve for the value of the independent variable which drives the error in the dependent variable to zero, provided the function is monotonically increasing. To improve convergence for functions whose derivative changes rapidly, the limits are reset as shown in the block diagram.

With  $f_4$  set to 1, the Iterator seeks a minimum of the function, provided the first derivative is single-valued between the limits. The inputs are redefined so that "y" is the derivative of the independent variable with respect to the dependent variable, and "x" is the value at which the derivative was computed or approximated. Since the desired value of y is zero,  $y_{ERR} = -y$ .

Since the Iterator uses the "Regula Falsi" technique, it requires two sets of variables to begin iteration. If only one set is available, flag  $f_3$  must be set to 1, causing the iterator to generate the independent variable increment from a percentage of the full range.

In addition to the above subroutines there are three other subroutines of primary interest to the five basic conic subroutines described in Sections 5.5.5 to 5.5.9. These are the Universal Variable Subroutine, the Kepler Equation Subroutine and the State Vector Subroutine shown in Figs. 5.10-4, 5.10-5 and 5.10-6, respectively.

The Universal Variable Subroutine is utilized by the Lambert, the Time-Theta and the Time-Radius Subroutines to compute the universal parameter  $x$  required for the time equation. There are two different formulations required according to the size of the parameter  $w$ .

If the input to the subroutine requires the physically impossible solution that the trajectory "close through infinity", the problem will be aborted, setting flag  $f_7$ .

The Kepler Equation Subroutine computes the transfer time given the variable  $x$  and the conic parameters.

The State Vector subroutine computes the position and velocity vectors at a point along the trajectory given an initial state vector, the variable  $x$  and the transfer time.

The final miscellaneous subroutine, the SETMU Subroutine, is depicted in Fig. 5.10-7. It sets  $\mu$  to the appropriate primary body gravitational constant consistent with the estimated CSM or LM state vector as defined in Section 5.2.2.6.

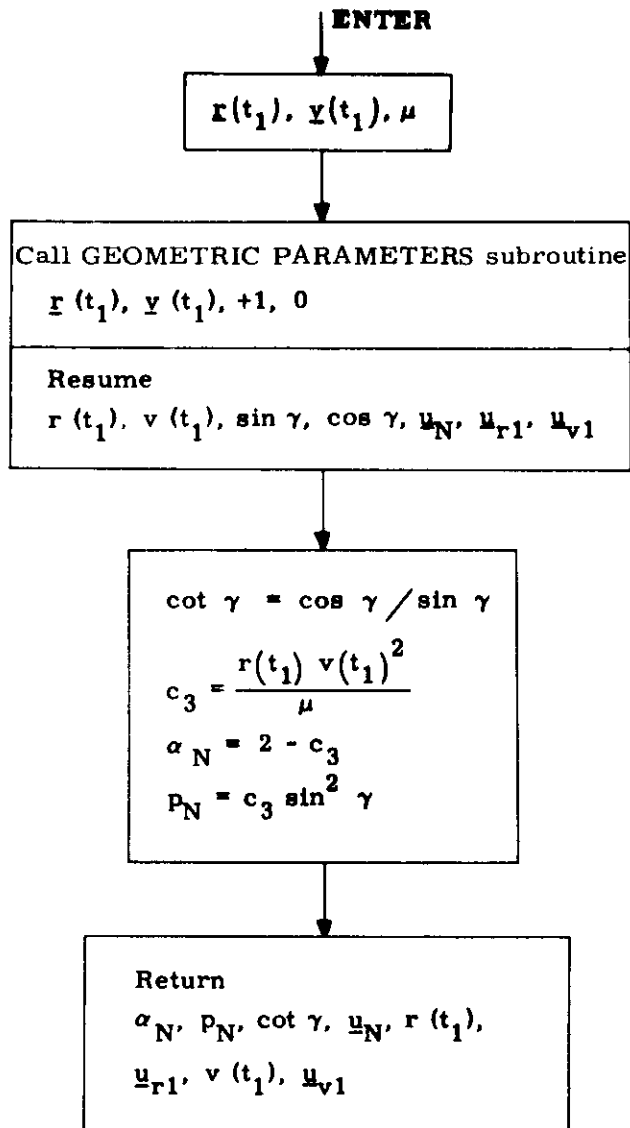


Figure 5.10-1 Conic Parameters Subroutine

5.5-43

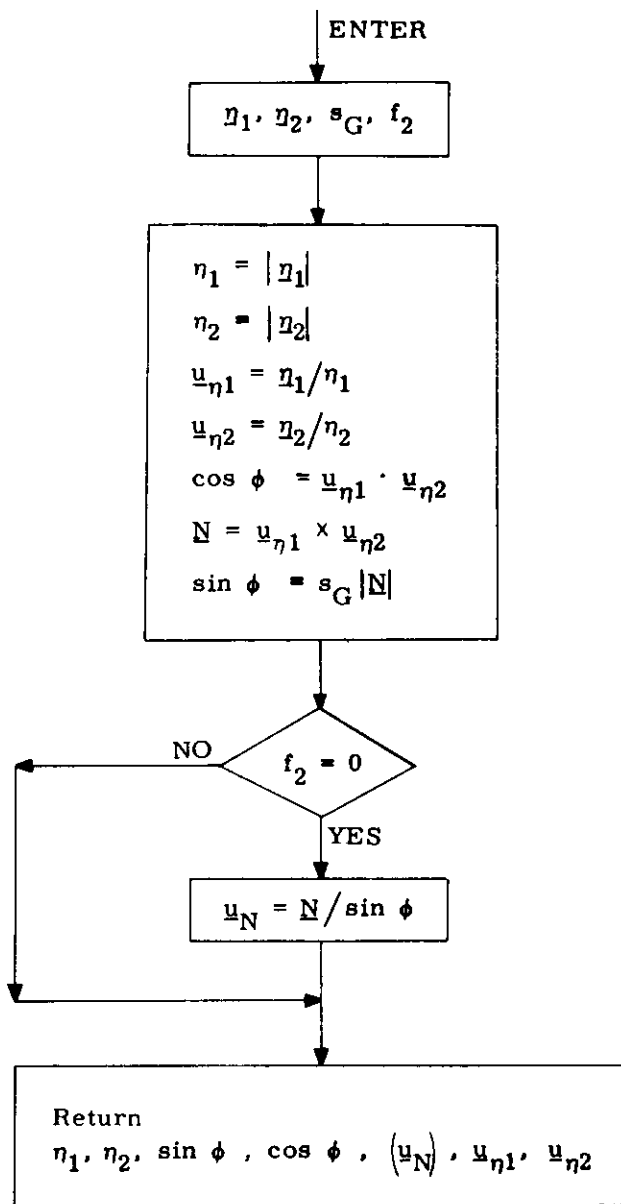


Figure 5.10-2 Geometric Parameters Subroutine

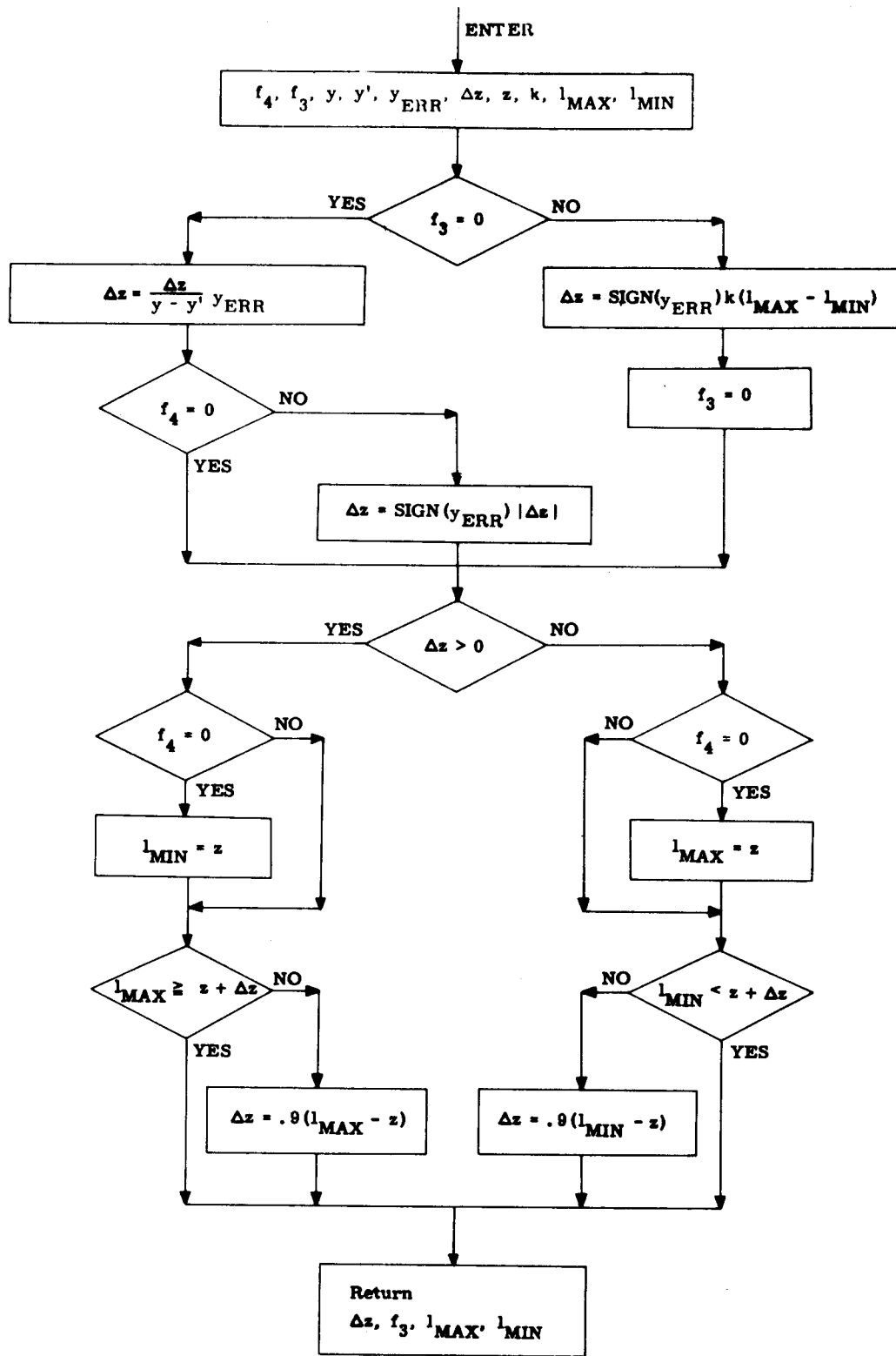


Figure 5.10-3 Iterator Subroutine

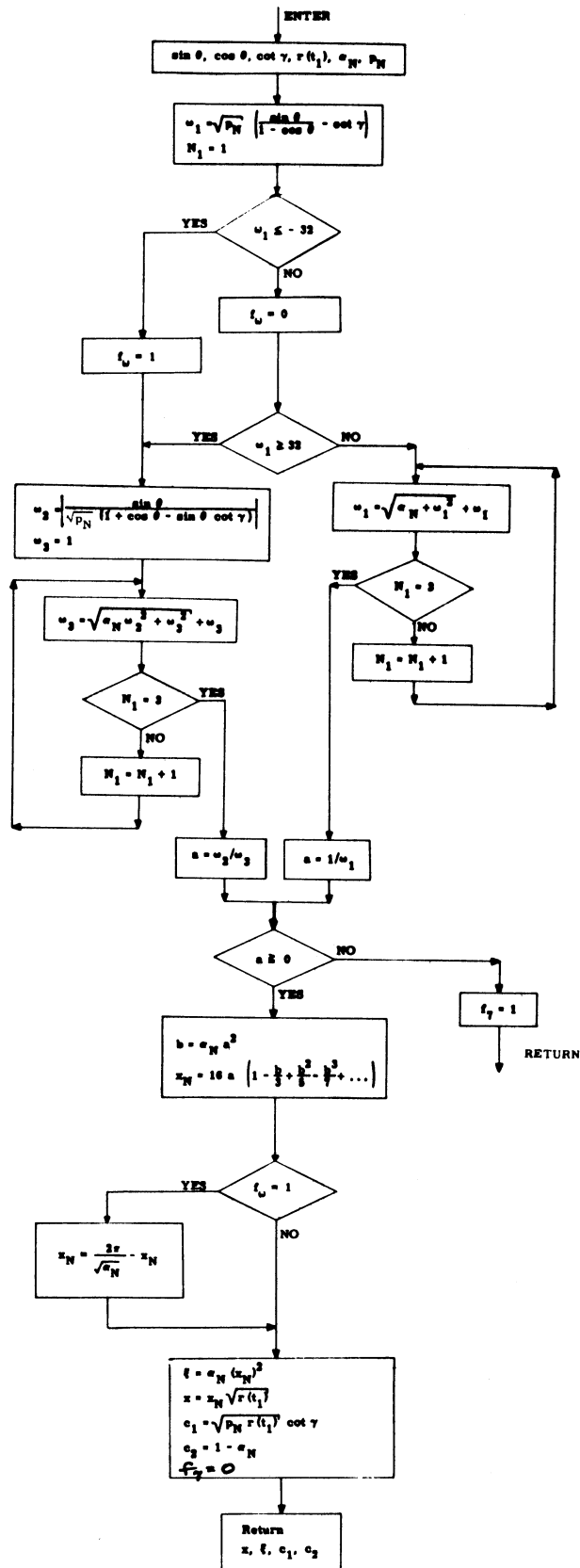


Figure 5.10-4 Universal Variable Subroutine

5. 5-46



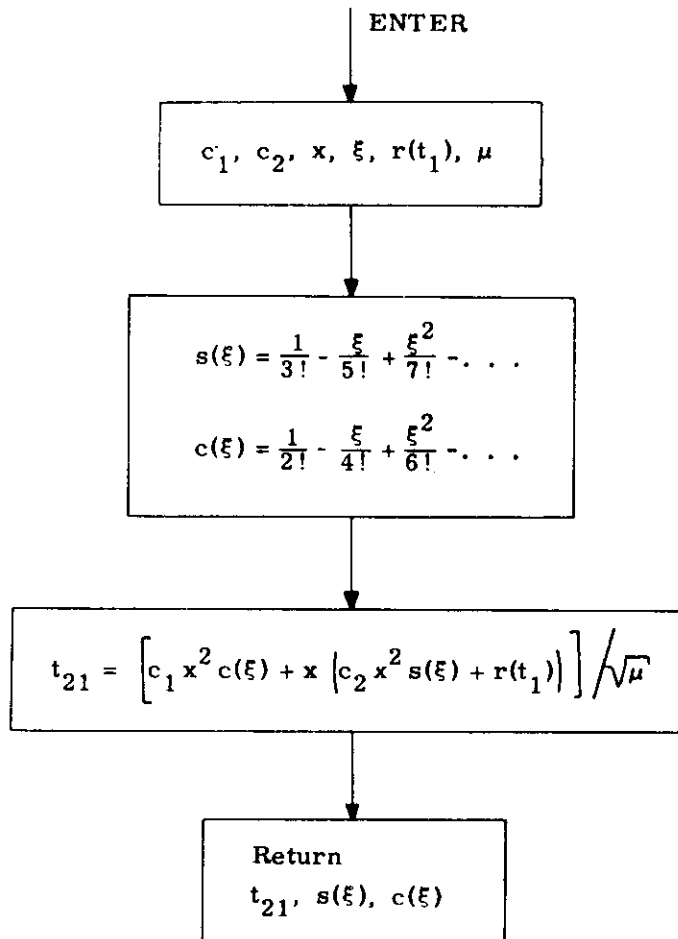


Figure 5.10-5 Kepler Equation Subroutine

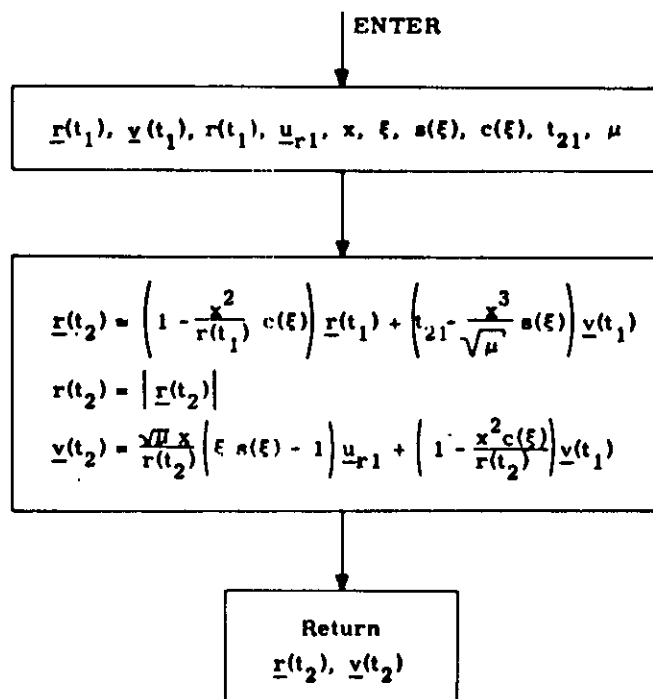


Figure 5.10-6 State Vector Subroutine

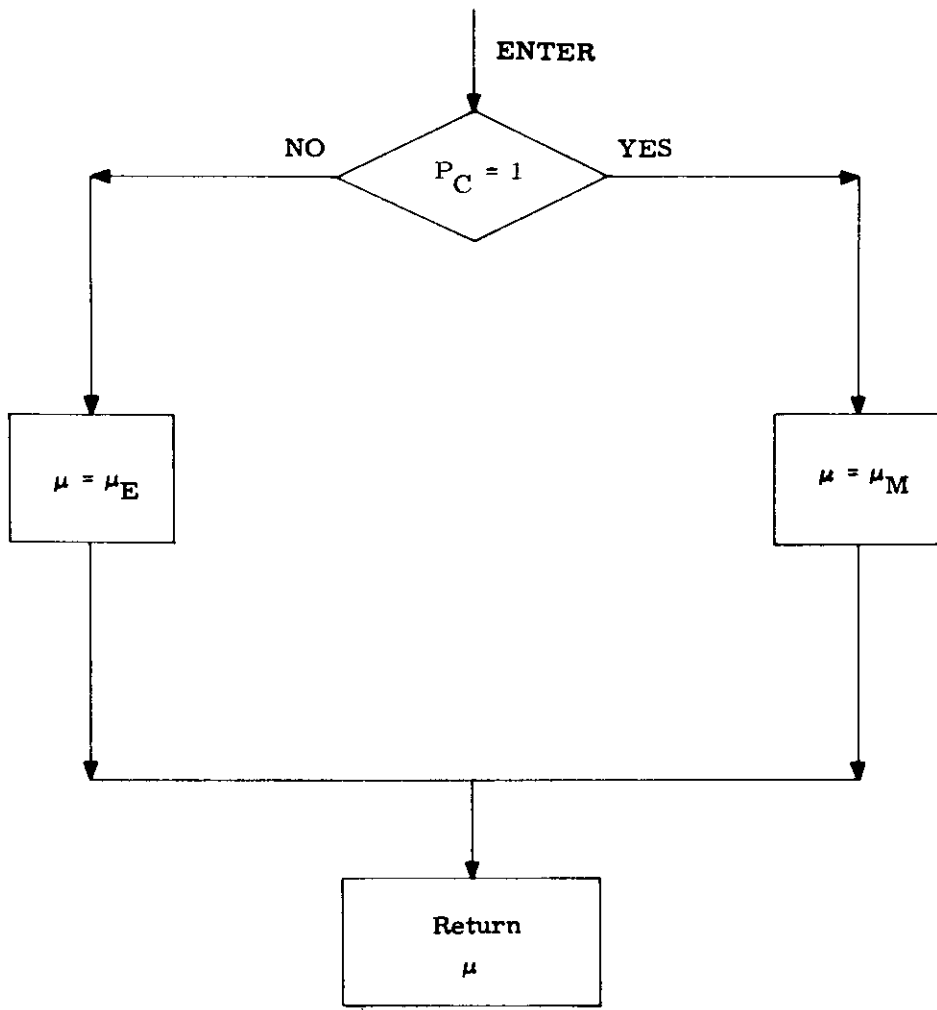


Figure 5.10-7 SETMU Subroutine

#### 5.5.11 INITIAL VELOCITY SUBROUTINE

The Initial Velocity Subroutine computes the required initial velocity vector for a trajectory of specified transfer time between specified initial and target position vectors. The trajectory may be either conic or precision depending on an input parameter (namely, number of offsets). In addition, in the precision trajectory case, the subroutine also computes an "offset target vector", to be used during pure-conic cross-product steering. The offset target vector is the terminal position vector of a conic trajectory which has the same initial state as a precision trajectory whose terminal position vector is the specified target vector.

In order to avoid the inherent singularities in the  $180^\circ$  transfer case when the (true or offset) target vector may be slightly out of the orbital plane, the Initial Velocity Subroutine rotates this vector into a plane defined by the input initial position vector and another input vector (usually the initial velocity vector), whenever the input target vector lies inside a cone whose vertex is the origin of coordinates, whose axis is the  $180^\circ$  transfer direction, and whose cone angle is specified by the user.

The Initial Velocity Subroutine is depicted in Fig. 5.11-1. The Lambert Subroutine, Section 5.5.6, is utilized for the conic computations; and the Coasting Integration Subroutine, Section 5.2.2, is utilized for the precision trajectory computations.

## Nomenclature for the Initial Velocity Subroutine

$\underline{r}(t_1)$	Initial position vector.
$\underline{v}(t_1)$	Vector (usually the actual initial velocity vector) used to determine whether the transfer from the initial position vector to the target vector is through a central angle of less or greater than $180^\circ$ , and also used in certain cases to specify the transfer plane (see text).
$\underline{r}_T(t_2)$	Target Vector (True target vector if $N_1 > 0$ , or Offset target vector if $N_1 = 0$ ).
$t_D$	Desired transfer time from initial position vector to target vector.
$N_1$	Number of offsets to be used in calculating the offset target vector from the true target vector. ( $N_1 = 0$ implies conic calculations only with offset target vector input).
$\epsilon$	Cone Angle of a cone whose vertex is the coordinate origin and whose axis is the $180^\circ$ transfer direction (i. e., the negative initial position direction). The cone angle $\epsilon$ is measured from the axis to the side of the cone.
$f_1$	Switch set to 0 or 1 according to whether a guess of $\cot \gamma$ is input or not.
$\{\cot \gamma\}$	Guess of $\cot \gamma$ .
$\underline{v}_T(t_1)$	Required initial velocity vector of a precision [a conic] trajectory which passes through the true [or offset] target vector, or the rotated true [or offset] target vector if the original target vector was in the cone, at the end of the desired transfer time, if $N_1 > 0$ [or $N_1 = 0$ ].
$\underline{r}(t_2)$	Computed offset target vector.
$\underline{v}_T(t_2)$	Final precision [conic] velocity vector resulting from a precision [conic] update of the initial position vector and the required initial velocity vector $\underline{v}_T(t_1)$ , if $N_1 > 0$ , [or $N_1 = 0$ , respectively].
$\underline{r}_T(t_2)$	Final precision position vector.
$\cot \gamma$	Value to which the Lambert Subroutine converged (for later use as guess to minimize computation time).
$f_2$	Switch set to 0 or 1 according to whether the input (true or offset) target vector was not or was in the cone, and consequently was not or was rotated into the plane.

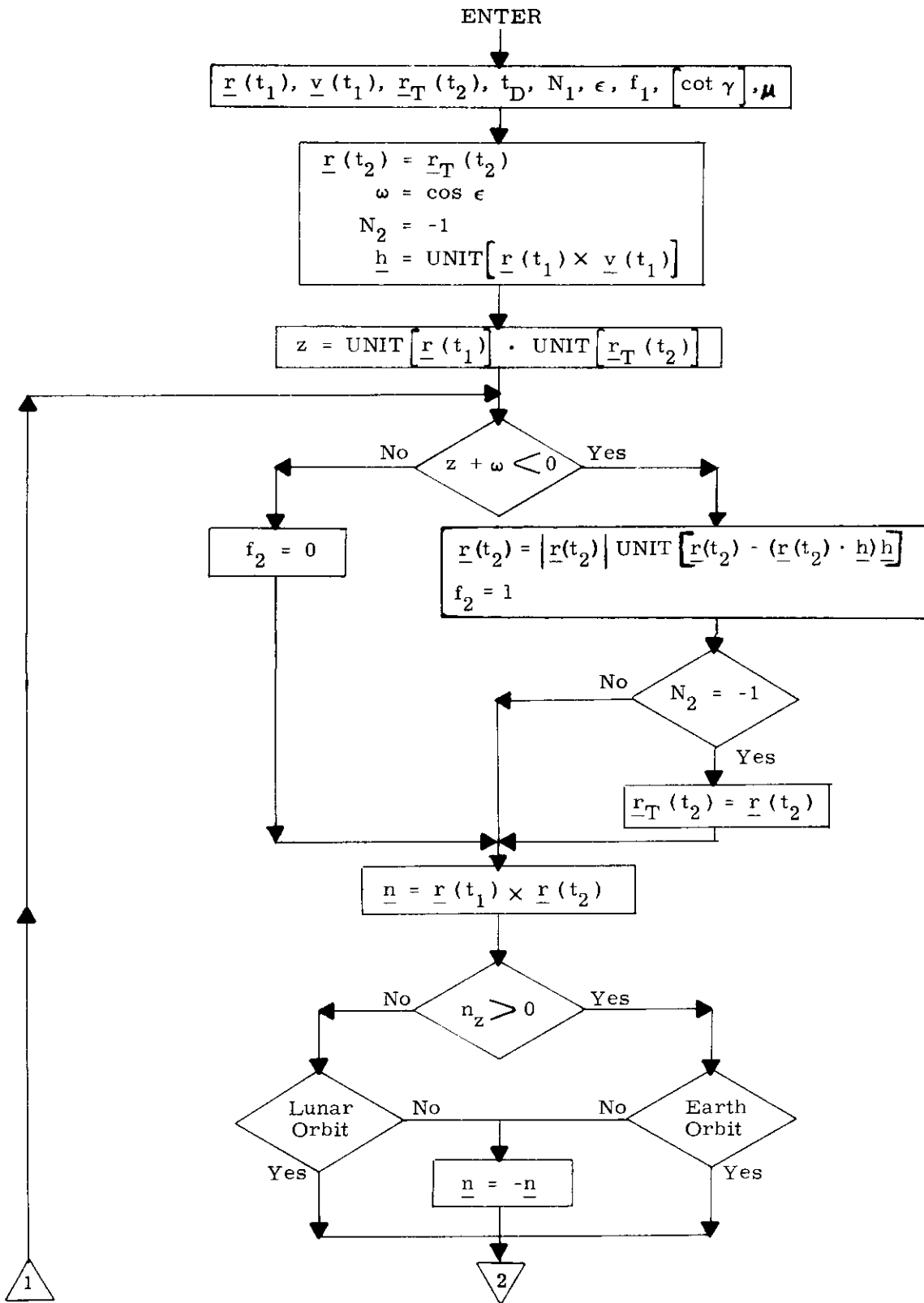


Figure 5.11-1 Initial Velocity Subroutine  
(page 1 of 2)

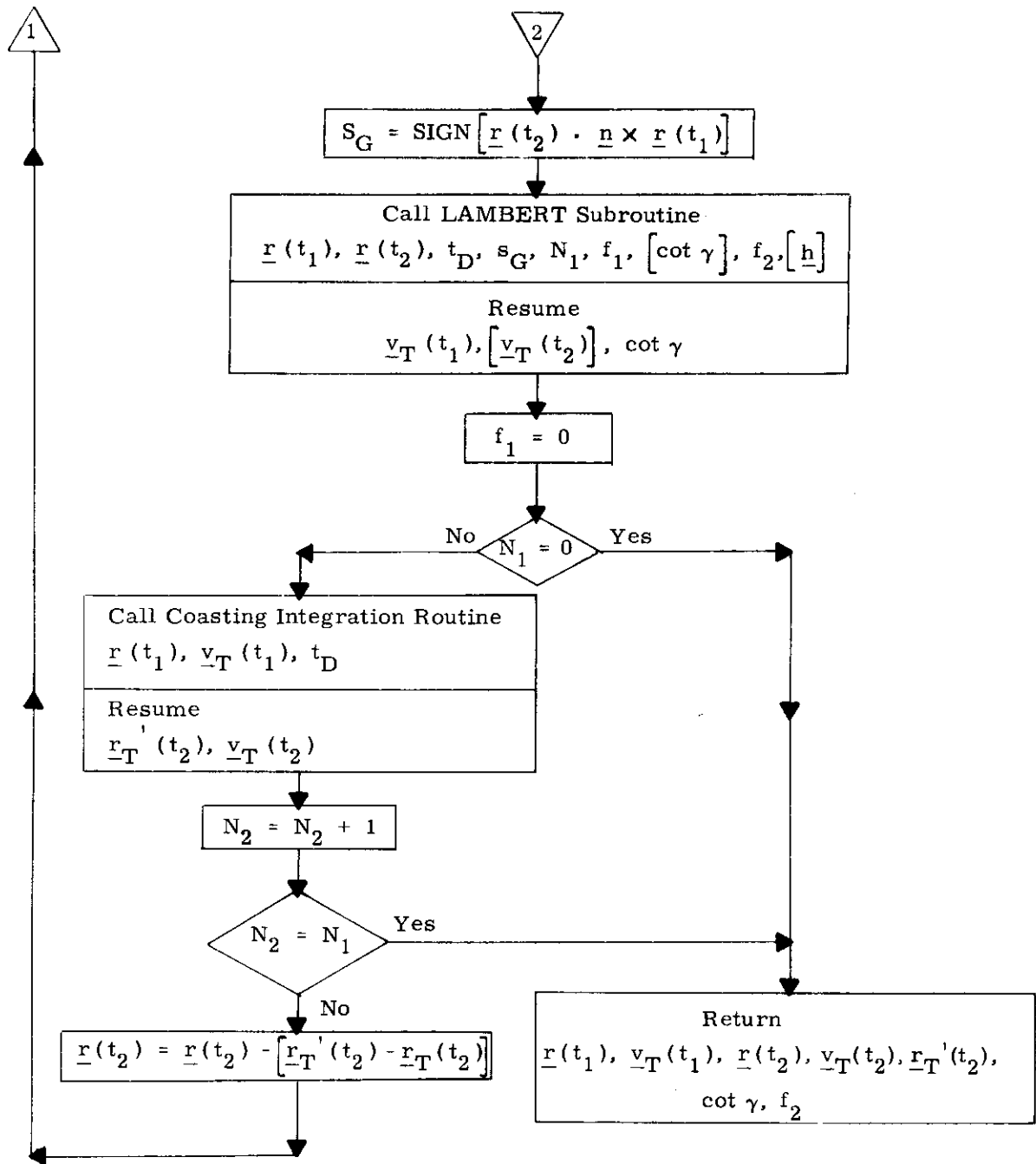


Figure 5.11-1 Initial Velocity Subroutine  
(page 2 of 2)

THIS PAGE INTENTIONALLY LEFT BLANK

5.5-52B

Rev. 3 - 11/68



#### 5.5.12 TRANSFER ANGLE SUBROUTINE

The computation of orbital transfer trajectories generally requires a time of flight,  $t_F$ . However, for purposes of communication between the astronaut and the guidance computer,  $\omega t$ , the central angle of travel of the passive vehicle during the transfer is more convenient. This subroutine is used to convert from  $t_F$  to  $\omega t$ . The conversion is approximate because it is based on the mean motion of the passive vehicle.

The equations and logic used for the conversion are shown in Fig. 5.12-1.

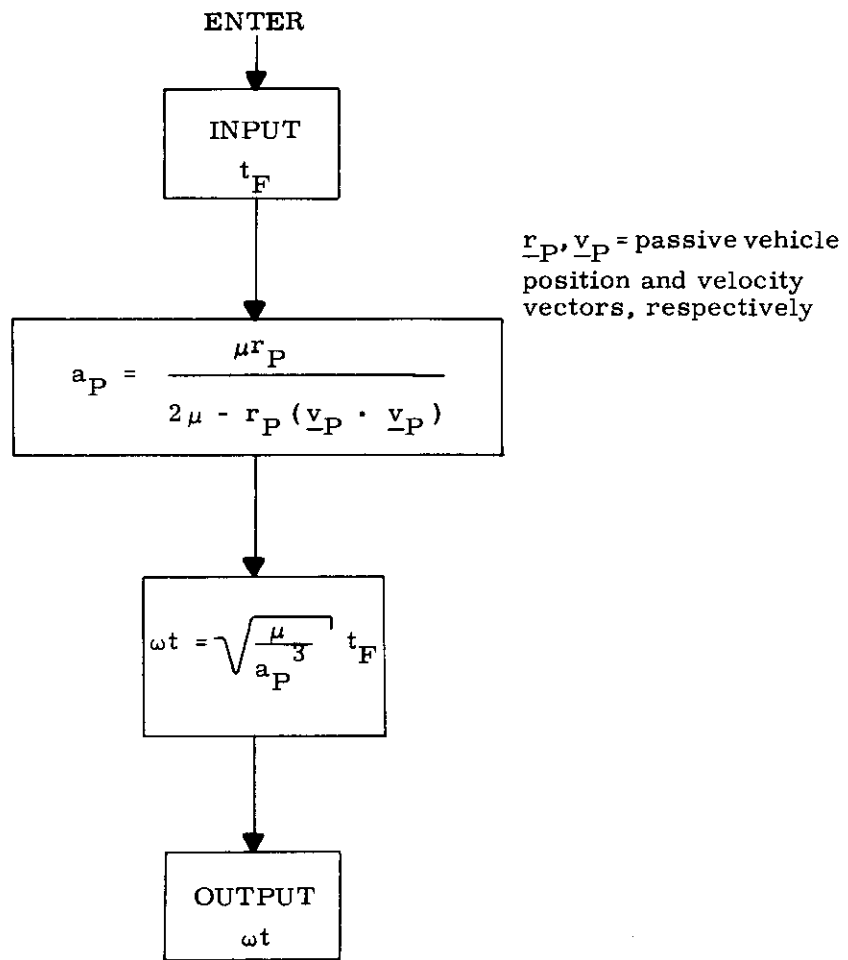


Fig. 5.12-1 Transfer Angle Subroutine

5. 5. 13 LOSSEM SUBROUTINE

The LOSSEM Subroutine computes the lines-of-sight of the Sun, Earth, and Moon with respect to the spacecraft in the Basic Reference Coordinate System. This data is used by the IMU alignment programs whenever the astronaut elects to sight on the Sun, Earth, or Moon instead of a star for purposes of IMU alignment. The data is also used by the Star Selection Routine (Section 5. 6. 4) when testing for star occultation. In addition, this subroutine computes the sizes of the occultation cones used in the Star Selection Routine and the correction for aberration of light which is applied in the IMU alignment programs to the line-of-sight unit vector of a star stored in basic reference coordinates.

The unit vectors  $\underline{u}_S$ ,  $\underline{u}_E$ , and  $\underline{u}_M$  specifying the lines-of-sight to the Sun, Earth, and Moon respectively, in the Basic Reference Coordinate System are computed as follows:

$$\underline{u}_S = \begin{cases} \text{UNIT}(\underline{r}_{ES}) & \text{if } P = E \\ \text{UNIT}(\underline{r}_{ES} - \underline{r}_{EM}) & \text{if } P = M \end{cases} \quad (5. 13. 1)$$

$$\underline{u}_E = \begin{cases} -\text{UNIT}(\underline{r}_C) & \text{if } P = E \\ -\text{UNIT}(\underline{r}_{EM} + \underline{r}_C) & \text{if } P = M \end{cases} \quad (5. 13. 2)$$

$$\underline{u}_M = \begin{cases} \text{UNIT}(\underline{r}_{EM} - \underline{r}_C) & \text{if } P = E \\ -\text{UNIT}(\underline{r}_C) & \text{if } P = M \end{cases} \quad (5. 13. 3)$$

where P, E, M, and S respectively denote the primary body, Earth, Moon, and Sun,  $\underline{r}_C$  is the position vector of the CSM with respect to the primary body, and  $\underline{r}_{EM}$  and  $\underline{r}_{ES}$  are the position vectors of the Moon and Sun with respect to the Earth obtained from the Lunar and Solar Ephemerides Subroutine of Section 5. 5. 4. The line-of-sight vectors are determined for a time specified by the calling program or routine.

The occultation cones used in the Star Selection Routine for the Sun, Earth, and Moon are computed as follows:

$$c_S = \cos 15^\circ \quad (5.13.4)$$

$$c_E = \begin{cases} \cos \left[ 5^\circ + \sin^{-1} \left( \frac{R_E}{r_C} \right) \right] & \text{if } P = E \\ \cos 5^\circ & \text{if } P = M \end{cases} \quad (5.13.5)$$

$$c_M = \begin{cases} \cos 5^\circ & \text{if } P = E \\ \cos \left[ 5^\circ + \sin^{-1} \left( \frac{R_M}{r_C} \right) \right] & \text{if } P = M \end{cases} \quad (5.13.6)$$

where  $c$  is the cosine of one half the total angular dimension of a cone and represents a more convenient way of treating the dimension of a cone in the Star Selection Routine,  $r_C$  is the magnitude of the CSM position vector,  $R_E$  is the equatorial radius (6378.166 km) of the Earth, and  $R_M$  is the mean radius (1738.09 km) of the Moon.

The vector  $\underline{a}$  which is used by the IMU alignment programs to correct the stored star vectors for aberration of light is determined as follows:

$$\underline{a} = \frac{\underline{v}_C - \underline{v}_{ES}}{c} \quad (5.13.7)$$

where  $\underline{v}_C$  is the velocity of the CSM with respect to the primary body,  $\underline{v}_{ES}$  is the velocity of the Sun relative to the Earth, and  $c$  is the speed of light. Note that this correction does not consider the velocity of the Moon relative to the Earth when the primary body is the Moon since the contribution from this source is considered negligible.

5.5.14 PERICENTER-APOCENTER (PERIAPO) SUBROUTINE

The Pericenter - Apocenter Subroutine computes the two body apocenter and pericenter altitudes given the position and velocity vectors for a point on the trajectory and the primary body.

This subroutine is depicted in Fig. 5.14-1.

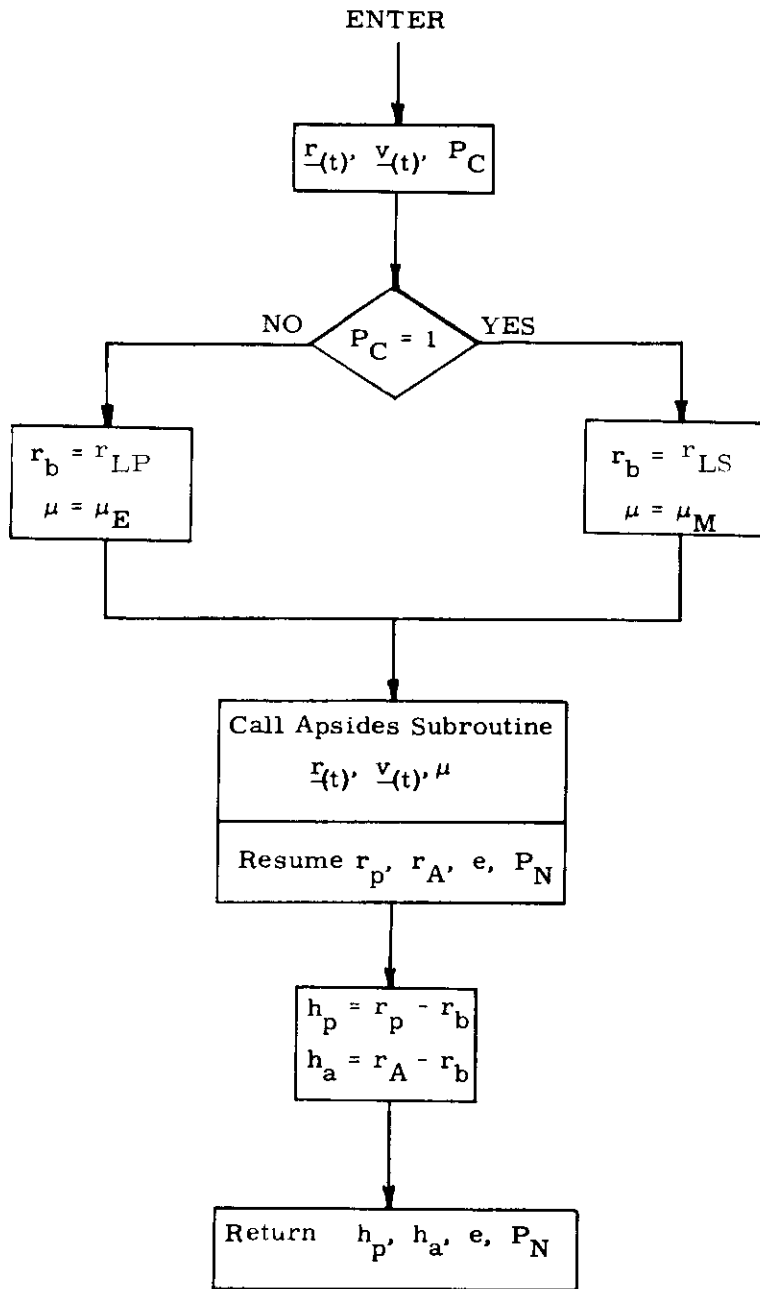


Figure 5.14-1 PERICENTER - APOCENTER SUBROUTINE