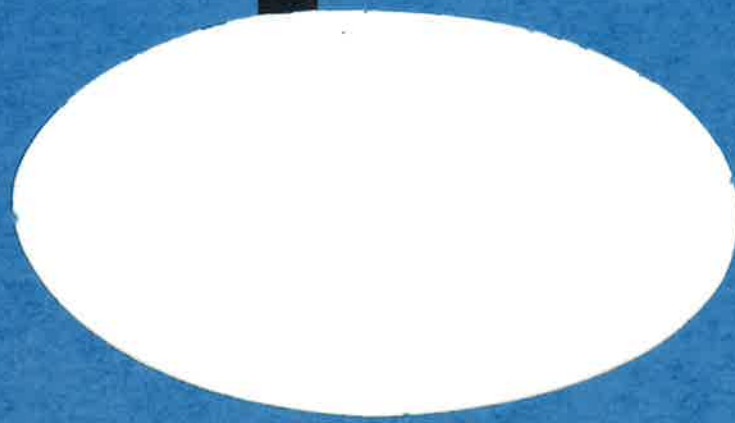


*Scanned

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

APOLLO

GUIDANCE, NAVIGATION AND CONTROL



MIT

**CHARLES STARK DRAPER
LABORATORY**

CAMBRIDGE MASSACHUSETTS 02139

APOLLO

GUIDANCE, NAVIGATION AND CONTROL

Approved: Gerald M. Levine Date: 15 Oct 71
G. M. LEVINE, DIRECTOR, GUIDANCE ANALYSIS
APOLLO GUIDANCE AND NAVIGATION PROGRAM

Approved: Stephen T. Copps Date: 15 Oct 1971
S. L. COPPS, SKYLARK PROJECT MANAGER
APOLLO GUIDANCE AND NAVIGATION PROGRAM

Approved: D. G. Hoag for Date: 15 Oct 71
R. H. BATTIN, DIRECTOR, MISSION DEVELOPMENT
APOLLO GUIDANCE AND NAVIGATION PROGRAM

Approved: David G. Hoag Date: 15 Oct 71
D. G. HOAG, DIRECTOR
APOLLO GUIDANCE AND NAVIGATION PROGRAM

Approved: Ralph R. Ragan Date: 15 Oct 71
R. R. RAGAN, DEPUTY DIRECTOR
INSTRUMENTATION LABORATORY

R-693

GUIDANCE SYSTEM OPERATIONS PLAN FOR
MANNED CSM EARTH ORBITAL MISSIONS
USING PROGRAM SKYLARK 1

SECTION 5 GUIDANCE EQUATIONS

OCTOBER 1971

MIT

CAMBRIDGE, MASSACHUSETTS, 02139

CHARLES STARK DRAPER LABORATORY

INDEXING DATA							
DATE	OPR	#	I	PGM	SUBJECT	SIGNATOR	LOC
10-31-71	MIT	R-693	R	APD	*	MIT	084-44

* scanned #51-208456


ACKNOWLEDGEMENT

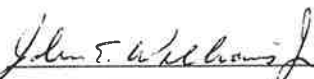
This report was prepared under DSR Project 55-23890, sponsored by the Manned Spacecraft Center of the National Aeronautics and Space Administration through Contract NAS 9-4065.

GUIDANCE SYSTEM OPERATIONS PLAN
FOR MANNED CM EARTH ORBITAL
MISSIONS USING PROGRAM
SKYLARK 1

SECTION 5 GUIDANCE EQUATIONS

Signatures appearing on this page designate approval of this document by NASA/MSC.

Approved:  Date: 4 Oct, '71
John R. Garman
Section Chief, Guidance Program Section
Manned Spacecraft Center, NASA

Approved:  Date: 4 Oct 71
John E. Williams, Jr.
Chief, Simulation and Flight Software Branch
Manned Spacecraft Center, NASA

Approved:  Date: 10/9/71
James C. Stokes, Jr.
Chief, Flight Support Division
Manned Spacecraft Center, NASA

BLANK

FOREWORD

The Guidance System Operations Plan (GSOP) for Program Skylark 1 is published in five sections as separate volumes.

2. Data Links
3. Digital Autopilots
4. Operational Modes
5. Guidance Equations
7. Erasable Programs

Since the information in Section 1 of the Colossus 2E GSOP is also applicable to the Skylark Program, Section 1 will not be republished for Skylark. The reader is referred, therefore, to R577 Colossus 2E GSOP, Section 1, Revision 2, January 1970. Also, Section 6 will not be published for Skylark.

The changes made to this Skylark GSOP Section 5 are enough to warrant its consideration as a new document. For that reason, it is not being treated as a revision of Colossus 3 GSOP 5. Major deletions, additions, and editorial changes have been made to reflect the scope of the new program. Paragraphs 5.1.3, 5.1.5.3, 5.1.5.4, 5.2.4, 5.5.12, 5.6.11, 5.6.16, and 5.6.17 have been deleted and so indicated in the Table of Contents and throughout the document. Also deleted and so indicated in text are a number of equations that do not pertain to the Skylark Program. Subsections 5.4 and 5.8 and paragraphs 5.1.1, 5.1.2, 5.6.1 through 5.6.7.3, 5.6.14, 5.6.9, and 5.9.1 have been rewritten. Appendix A, containing a list of PCRs and PCNs, whose implementation is reflected in this issue, has been added. The rest of the document has undergone extensive editorial changes in the hope of making it more useful to the reader. At the same time, the basic form of the Colossus GSOP 5 has been retained to facilitate comparison with earlier GSOPs.

The volume is published as a control document governing the basic design for the guidance and navigation computations of Skylark 1 and supersedes M. I. T. - DL Section-5 memos published earlier. Revisions constituting changes to the Skylark Program require NASA approval.

BLANK

CONTENTS

<u>Section</u>		<u>Page</u>
5.1	INTRODUCTION	5.1-1
5.1.1	General Comments	5.1-1
5.1.2	Sections 4 and 5 Cross Reference	5.1-2
5.1.3	(Deleted from Skylark)	
5.1.4	Coordinate Systems	5.1-6
5.1.4.1	Basic Reference Coordinate System	5.1-6
5.1.4.2	IMU Stable Member or Platform Coordinate System	5.1-7
5.1.4.3	Vehicle or Body Coordinate System	5.1-9
5.1.4.4	The Earth-fixed Coordinate System	5.1-9
5.1.4.5	Apollo Telescope Mount Sensor Coordinate System	5.1-9
5.1.4.6	Navigation Base Coordinate System	5.1-10
5.1.5	General Definitions and Conventions	5.1-10
5.1.5.1	Error Transition Matrix Maintenance	5.1-10
5.1.5.2	Altitude Parameters Convention	5.1-12
5.1.5.3	(Deleted from Skylark)	
5.1.5.4	(Deleted from Skylark)	
5.1.5.5	Time Definitions	5.1-13
5.2	COASTING FLIGHT NAVIGATION	5.2-1
5.2.1	General Comments	5.2-1
5.2.2	Coasting Integration Routine	5.2-5
5.2.2.1	General Comments	5.2-5
5.2.2.2	Encke's Method	5.2-5
5.2.2.3	Disturbing Acceleration	5.2-8
5.2.2.4	Error-Transition Matrix	5.2-9
5.2.2.5	Numerical Integration Method	5.2-12
5.2.2.6	Coasting Integration Logic	5.2-13

CONTENTS (cont'd)

<u>Section</u>	<u>Page</u>
5.2.3 Measurement Incorporation Routine	5.2-18
5.2.4 (Deleted from Skylark)	
5.2.5 Universal Tracking and Rendezvous Navigation Program (P20)	5.2-25
5.2.5.1 Tracking	5.2-25
5.2.5.2 Rendezvous Tracking Data Processing Routine (R22)	5.2-35
5.2.5.3 Automatic W-matrix Reinitialization (WRI) . .	5.2-51
 5.3 POWERED FLIGHT NAVIGATION AND GUIDANCE	 5.3-1
5.3.1 General Comments	5.3-1
5.3.2 Powered Flight Navigation - Average-G Routine	5.3-1
5.3.3 Powered Flight Guidance Using Cross Product Steering .	5.3-5
5.3.3.1 Introduction	5.3-5
5.3.3.2 Powered Flight Guidance Computation Sequencing	5.3-6
5.3.3.3 Prethrust Computations	5.3-14
5.3.3.4 Cross Product Steering Routine	5.3-32
5.3.3.5 Velocity-to-be-Gained Routine	5.3-37
5.3.3.6 SPS Thrust Fail Routine (R40)	5.3-40
 5.3.4 Thrust Monitor Programs	 5.3-41
5.3.5 Earth Orbit Insertion Monitor Program—P11	5.3-41
5.3.5.1 Introduction	5.3-41
5.3.5.2 Nomenclature for P11	5.3-44
5.3.5.3 Time Sequencing of P11 Subroutines	5.3-45
5.3.5.4 Time Subroutine	5.3-46
5.3.5.5 State Subroutine	5.3-46
5.3.5.5a Attitude Error Subroutine	5.3-46
5.3.5.6 Display Subroutine	5.3-46
 5.4 TARGETING	 5.4-1
5.4.1 Rendezvous Targeting Constraints	5.4-2
5.4.2 NC1 Targeting Program (P31)	5.4-3
5.4.3 NC2 Targeting Program (P32)	5.4-15
5.4.4 NCC Targeting Program (P33)	5.4-16

CONTENTS (cont'd)

<u>Section</u>	<u>Page</u>
5.4.5 NSR Targeting Program (P34)	5.4-17
5.4.6 TPI Targeting Program (P35)	5.4-20
5.4.7 TPM Targeting Program (P36)	5.4-28
5.4.8 NPC Targeting Program (P38)	5.4-31
5.4.9 Targeting Program Routines	5.4-34
5.4.9.1 REVUP Routine	5.4-34
5.4.9.2 RADUP Routine	5.4-34
5.4.9.3 COE Routine	5.4-34
5.4.9.4 ITER Routine	5.4-34
5.4.9.5 QRDTPI Routine	5.4-38
 5.5 BASIC SUBROUTINES	 5.5-1
5.5.1 General Comments	5.5-1
5.5.1.1 Solar System Subroutines	5.5-1
5.5.1.2 Conic Trajectory Subroutines	5.5-1
5.5.1.3 Additional Subroutines	5.5-8
5.5.2 Planetary Inertial Orientation Subroutine	5.5-8
5.5.3 Latitude-Longitude Subroutine	5.5-9
5.5.4 Solar Ephemerides	5.5-13
5.5.5 Kepler Subroutine	5.5-13
5.5.6 Lambert Subroutine	5.5-17
5.5.7 Time-Theta Subroutine	5.5-20
5.5.8 Time-Radius Subroutine	5.5-22
5.5.9 Apsides Subroutine	5.5-22
5.5.10 Miscellaneous Subroutines	5.5-26
5.5.11 Initial Velocity Subroutine	5.5-33
5.5.11.1 Nomenclature for the Initial Velocity Subroutine	5.5-34
5.5.12 (Deleted from Skylark)	
5.5.13 LOCSAM Subroutine	5.5-37
5.5.14 Pericenter-Apocenter (Periapo) Subroutine	5.5-38
5.5.15 Time-of-Longitude Program (P29)	5.5-38

CONTENTS (cont'd)

<u>Section</u>	<u>Page</u>
5.6 GENERAL SERVICE ROUTINES	5.6-1
5.6.1 General Comments	5.6-1
5.6.2 IMU Alignment Modes	5.6-1
5.6.2.1 IMU Orientation Determination Program (P51)	5.6-1
5.6.2.2 IMU Realignment Program (P52)	5.6-4
5.6.2.3 Backup IMU Orientation Determination Program (P53)	5.6-9
5.6.2.4 Backup IMU Realignment Program (P54)	5.6-10
5.6.2.5 ATM Orientation Determination Program	5.6-11
5.6.2.6 ATM Star Tracker Gimbal Angle Program	5.6-15
5.6.3 IMU Routines	5.6-19
5.6.3.1 Sextant Transformations	5.6-19
5.6.3.2 IMU Transformations	5.6-21
5.6.3.3 REFSMMAT Transformations	5.6-26
5.6.3.4 Navigation Base to ATM Transformation (NBATM)	5.6-29
5.6.4 Star Selection Routine	5.6-29
5.6.5 Ground Track Routine	5.6-31
5.6.6 VHF Range-Rate Mark Processing Routine	5.6-31
5.6.6.1 Data Read Routine for Range-Rate Mark Processing	5.6-37
5.6.6.2 Provision for VHF Range Modulus	5.6-38
5.6.6.3 R27 Computation of Theta and Phi	5.6-38
5.6.7 Additional Rendezvous Displays	5.6-38
5.6.7.1 Range, Range Rate, Theta Display (R31)	5.6-42
5.6.7.2 Range, Range Rate, Phi Display (R34)	5.6-44
5.6.7.3 Final Attitude Display (R63)	5.6-45
5.6.7.4 Out-of-Plane Rendezvous Display (R36)	5.6-45
5.6.8 Automatic Optics Positioning Routine	5.6-46
5.6.8.1 General	5.6-46
5.6.8.2 Line-of-Sight Subroutine	5.6-47
5.6.8.3 Command Subroutine	5.6-47

CONTENTS (concl'd)

<u>Section</u>	<u>Page</u>
5.6.9 Optics Angles Transform Routine	5.6-47
5.6.10 Orbital Parameter and Entry Display Computations . .	5.6-48
5.6.10.1 Orbital Parameters Display Routine (R30) . .	5.6-48
5.6.10.2 Splash Error Computation Subroutine (DELRSPL)	5.6-53
5.6.10.3 The Entry Preparation Program (P61) Displays	5.6-57
5.6.10.4 The TFF Subroutines	5.6-61
5.6.10.5 Equations Used in Calculating the Conic Time of Flight	5.6-64
5.6.10.6 Desired Gimbal-angle Display for Entry Attitude Used During P62 and P66	5.6-74
5.6.10.7 Entry Terminal Display of CM Geographic Location	5.6-74
5.6.11 (Deleted from Skylark)	
5.6.12 CMC Idling Program	5.6-76
5.6.13 IMU Compensation	5.6-79
5.6.14 Impulsive ΔV Program (P77)	5.6-80
5.6.15 Estimated RMS Position and Velocity Error Display . .	5.6-80
5.6.16 (Deleted from Skylark)	
5.6.17 (Deleted from Skylark)	
5.7 ENTRY GUIDANCE	5.7-1
5.8 ERASABLE MEMORY PARAMETER LIST	5.8-1
5.9 FIXED MEMORY CONSTANTS	5.9-1
5.9.1 Fixed Constants	5.9-1
5.9.2 References for Fixed Constants	5.9-5
5.9.3 Comments on Fixed Constants	5.9-7
APPENDIX A: PCR-PCN CHECKLIST	A-1

BLANK

ILLUSTRATIONS

Figure		Page
2.1-1	Simplified CMC Rendezvous Navigation Functional Diagram . .	5.2-3
2.2-1	Simplified Coasting Integration Routine Logic Diagram	5.2-16
2.2-2	Coasting Integration Routine Logic Diagram	5.2-17
2.3-1	Measurement Incorporation Procedure	5.2-23
2.3-2	INCORP2 Subroutine Logic Diagram	5.2-24
2.5-1	Rendezvous Navigation Program (P20 options 0 and 4)	5.2-34
2.5-2	(Deleted from Skylark)	
2.5-3	Rendezvous Tracking Data Processing Routine	5.2-36
2.5-4	Rendezvous Navigation Computations	5.2-44
3.2-1	Average-G Subroutine	5.3-3
3.2-2	Gravity Subroutine	5.3-4
3.3-1	Ignition Countdown-External Δv Subroutine Sequencing for SPS Maneuvers	5.3-7
3.3-2	Normal External Δv Subroutine Sequencing for SPS Maneuvers (Maneuver Time Greater Than 6 Sec)	5.3-8
3.3-3	Engine-Off External Δv Subroutine Sequencing for SPS Maneuvers	5.3-9
3.3-4	Ignition Countdown—Lambert Aimpoint Subroutine Sequencing for SPS Maneuvers	5.3-10
3.3-5	Normal Lambert Aimpoint Subroutine Sequencing for SPS Maneuvers (Maneuver Time Greater Than 6 Sec)	5.3-11
3.3-6	Engine-Off Lambert Aimpoint Subroutine Sequencing for SPS Maneuvers	5.3-12
3.3-7	External Δv Prethrust Routine	5.3-16
3.3-8	Inplane External Δv Maneuver Angle Compensation	5.3-18
3.3-9	Lambert Aimpoint Maneuver Prethrust Routine	5.3-22
3.3-10	Initial Attitude Routine	5.3-26
3.3-11	Time-to-go Prediction Routine	5.3-28
3.3-11a	Complex Impulsive Burn Routine	5.3-31
3.3-12	Cross Product Steering Routine—Update v_G Section	5.3-35
3.3-13	Velocity-to-be-Gained Subroutine for Lambert Aimpoint Guidance	5.3-39
3.5-1	Angle Definitions	5.3-43
3.5-2	Time Subroutine	5.3-47
3.5-3	State Subroutine	5.3-48
3.5-4	Attitude Error Subroutine	5.3-50
3.5-5	FDAI Display Subroutines No. 1 and 2	5.3-51
3.5-6	Display Subroutine	5.3-53
3.5-7	Spacecraft Control of Saturn Subroutine	5.3-54

ILLUSTRATIONS (cont'd)

Figure		Page
4. 1-1	Definition of Elevation Angle E_L	5. 4-4
4. 1-2	Illustration of the Skylark Rendezvous Constraints	5. 4-5
4. 2-1	NC1, NC2 Targeting Program Flowchart	5. 4-6
4. 4-1	NCC Targeting Program Flowchart	5. 4-18
4. 5-1	NSR Targeting Program Flowchart	5. 4-21
4. 6-1	TPI Targeting Program Flowchart	5. 4-24
4. 7-1	TPM Targeting Program Flowchart	5. 4-29
4. 8-1	NPC Targeting Program Flowchart	5. 4-32
4. 9-1	REVUP Routine Flowchart	5. 4-35
4. 9-2	RADUP Routine Flowchart	5. 4-35
4. 9-3	COE Routine Flowchart	5. 4-36
4. 9-4	ITER Routine Flowchart	5. 4-37
4. 9-5	QRDTPI Routine Flowchart	5. 4-39
5. 3-1	Vector to Latitude, Longitude, Altitude Computation Logic Diagram	5. 5-10
5. 3-2	Latitude, Longitude, Altitude to Vector Computation Logic Diagram	5. 5-11
5. 3-3	Determination of r_0	5. 5-12
5. 5-1	Kepler Subroutine	5. 5-14
5. 6-1	Lambert Subroutine	5. 5-18
5. 7-1	Time-Theta Subroutine	5. 5-21
5. 8-1	Time-Radius Subroutine	5. 5-23
5. 9-1	Apsides Subroutine	5. 5-25
5. 10-1	Conic Parameters Subroutine	5. 5-27
5. 10-2	Geometric Parameters Subroutine	5. 5-28
5. 10-3	Iterator Subroutine	5. 5-29
5. 10-4	Universal Variable Subroutine	5. 5-30
5. 10-5	Kepler Equation Subroutine	5. 5-31
5. 10-6	State Vector Subroutine	5. 5-32
5. 11-1	Initial Velocity Subroutine	5. 5-35
5. 14-1	Pericenter-Apocenter Subroutine	5. 5-39
5. 15-1	Time-of-Longitude Program (P29)	5. 5-40
6. 4-1	Star Selection Routine	5. 6-30
6. 6-1	Initialization and Recursive Branches for R27	5. 6-33
6. 6-2	Initialization Procedure	5. 6-34
6. 6-3	Recursive Procedure	5. 6-36
6. 6-4	Data Request Routine	5. 6-39

ILLUSTRATIONS (concl'd)

Figure		Page
6.6-5	Polynomial Range Estimate Modification Due to VHF Range Modulus (VHFMOD Sequence)	5.6-40
6.6-6	Computation of Theta and Phi in R27	5.6-41
6.7-1	Definition of Theta	5.6-42
6.7-2	Range, Range Rate, θ	5.6-43
6.7-3	Definition of Phi	5.6-44
6.10-1	Orbital Parameters Display Routine	5.6-51
6.10-2	Splash Error Computation Subroutine	5.6-54
6.10-3	Empirical Relations for Entry Range	5.6-56
6.10-4	Display Calculations for P61	5.6-58
6.10-5	TFFCONIC Subroutine	5.6-65
6.10-6	TFFRP/RA Subroutine	5.6-66
6.10-7	CALCTFF/CALCTPER Subroutine	5.6-67
6.10-8	TFF/TRIG Subroutine	5.6-69
6.10-9	FISHCALC Subroutine	5.6-70
6.10-10	VGAMCALC Subroutine	5.6-71
6.10-11	Calculation of Gimbal Angles Corresponding to Entry Attitude .	5.6-75
6.12-1	CMC Idling Program—State Vector Extrapolation	5.6-78
6.14-1	Impulsive Δv Program (P77)	5.6-81
6.16-1	RSS Position and Velocity Error Display	5.6-83
7.0-0	Entry Computation	5.7-3
7.0-1	Computer Variables	5.7-4
7.0-2	Display Information	5.7-8
7.0-3	Constants, Gains and Conversion Factors	5.7-9
7.0-4	Entry Program Initialization	5.7-10
7.0-5	Average-G Navigation Routine for Entry from Earth Orbit . .	5.7-11
7.0-6	Entry Targeting	5.7-12
7.0-7	Initial Roll (INITROLL)	5.7-13
7.0-8	Huntest	5.7-14
7.0-9	Entry Range Prediction	5.7-15
7.0-10	Constant Drag Control	5.7-16
7.0-11	Entry Upcontrol	5.7-17
7.0-12	Entry Ballistic Phase	5.7-18
7.0-13	Entry Final Phase	5.7-19
7.0-14	Final Phase Reference	5.7-20
7.0-15	Entry G-Limiter	5.7-22
7.0-16	Entry Lateral Logic	5.7-23
9.1-1	Unit Vectors of the Navigational Stars	5.9-4

BLANK

SECTION 5

GUIDANCE EQUATIONS

5.1 INTRODUCTION

5.1.1 General Comments

Section 5 of the Guidance System Operations Plan (GSOP) presents the guidance and navigation computation routines for the CSM Skylark program. These routines are utilized by the programs outlined in Section 4, in which astronaut and other subsystem interface and operational requirements are described. The guidance and navigation equations and procedures presented are primarily concerned with mission programs reflecting the current CSM GNCS computer (CMC) capability. A limited number of CMC service-program operations involving computations are also included.

This section presents the equations for the Skylab mission in the six functional categories listed below. Each category constitutes a subsection of this document.

Coasting Flight Navigation	—	subsection 5.2
Powered Flight Navigation and Guidance	—	subsection 5.3
Targeting	—	subsection 5.4
Basic Subroutines	—	subsection 5.5
General Service Routines	—	subsection 5.6
Entry Guidance	—	subsection 5.7

In addition, subsection 5.1 presents general information to help the reader understand the Skylark program. (For example, paragraph 5.1.2 is a cross reference of GSOP Sections 4 and 5 relating the program numbers and titles of Section 4 to the paragraphs that describe them in Section 5; paragraph 5.1.4 explains the coordinate systems used in the document.) Subsection 5.8 lists the guidance-equation parameters required for program initialization and operation. These selected parameters are stored in the CMC erasable memory. Finally, subsection 5.9 presents those fixed-memory constants—which have not been specified elsewhere—that are used in the equations throughout the document.

The principal contributing authors are listed below:

C. M. Beals	W. M. Robertson
P. M. Kachmar	W. H. Tempelman
G. M. Levine	R. L. White
R. E. Phillips	

Technical writing and publications coordination for the document were performed by H. D. Nayar.

5.1.2 Sections 4 and 5 Cross Reference

Paragraph 5.1.2 provides the reader a correlation between the programs and routines of Section 4 and the specification data of Section 5. The programs are arranged according to mission phases to further provide the reader with a structure for understanding the Skylab mission. Programs and routines are treated as discrete units in Section 4. Because of the arrangement of material in Section 5, however, it is not always possible to reference a specific paragraph for each program or routine. In these cases, either a paragraph that generally describes that program or routine is referenced, or no reference is given.

<u>SECTION 4</u>		<u>SECTION 5</u>
<u>Mission Phase—Program</u>	<u>Title</u>	<u>Principal Paragraph</u>
<u>Prelaunch and Service</u>		
P00	CMC Idling	5.6.12
P01	Initialization	
P02	Gyro Compassing	
P03	Optical Verification of Gyro Compassing	
P06	CMC Power Down	
P07	System Test	
<u>Boost</u>		
P11	Earth Orbit Insertion Monitor	5.3.5

SECTION 4 <u>Mission Phase-Program</u>	<u>Title</u>	SECTION 5 <u>Principal Paragraph</u>
<u>Coast</u>		
P20	Universal Tracking	5.2.5
P21	Ground Track Determination	5.6.5
P25	Contingency VHF Range Rate	5.6.6
P27	CMC Update	
P29	Time of Longitude	5.5.15
<u>Prethrusting</u>		
P30	External Delta V	5.3.3.3.1
P31	NC1 Targeting	5.4.2
P32	NC2 Targeting	5.4.3
P33	NCC Targeting	5.4.4
P34	NSR Targeting	5.4.5
P35	TPI Targeting	5.4.6
P36	TPM Targeting	5.4.7
P37	Rendezvous Final Phase	
P38	Plane Change Targeting	5.4.8
<u>Thrusting</u>		
P40	SPS Thrusting	5.3.3
P41	RCS Thrusting	5.3.3
P47	Thrust Monitor	5.3.4
P48	Rendezvous Thrust Monitor	5.3.4
<u>Alignment</u>		
P50	ATM Orientation Determination	5.6.2.5
P51	IMU Orientation Determination	5.6.2.1
P52	IMU Realign	5.6.2.2

SECTION 4
Mission Phase-Program

Title

SECTION 5
Principal Paragraph

Alignment (cont'd)

P53	Backup IMU Orientation Determination	5.6.2.3
P54	Backup IMU Realign	5.6.2.4
P55	ATM Star Tracker Gimbal Angle	5.6.2.6

Entry

P61	Entry Preparation	5.7
P62	CM-SM Separation and Pre- entry Maneuver	5.7
P63	Entry Initialization	5.7
P64	Post 0.05G	5.7
P65	Up Control	5.7
P66	Ballistic	5.7
P67	Final Phase	5.7

Post-thrusting

P77	CSM Velocity Vector Update	5.6.14
-----	----------------------------	--------

SECTION 4

SECTION 5

<u>Routine</u>	<u>Title</u>	<u>Principal Paragraph</u>
R00	Final Automatic Request Terminate	
R01	Erasable and Channel Modification	
R02	IMU Status Check	
R03	CSM DAP Data Load	
R04	Docked DAP Data Load	
R07	MINKEY Controller	
R08	VHF Range Read	
R21	Rendezvous Tracking Sighting Mark	5. 2. 5. 1
R22	Rendezvous Tracking Data Processing	5. 2. 5. 2
R23	Rendezvous Backup Sighting Mark	5. 2. 5. 1
R27	VHF Range Rate Mark Processing	5. 6. 6
R30	Orbital Parameters Display	5. 6. 10. 1
R31	Rendezvous Parameter Display No. 1	5. 6. 7. 1
R34	Rendezvous Parameter Display No. 2	5. 6. 7. 2,
R36	Rendezvous Out-of-Plane Display	5. 6. 7. 4
R40	SPS Thrust Fail	5. 3. 3. 6
R41	State Vector Integration (MID TO AVE)	5. 3. 3. 3. 4
R50	Coarse Align	5. 6. 2. 2
R52	Automatic Optics Positioning	5. 6. 8

SECTION 4

SECTION 5

<u>Routine</u>	<u>Title</u>	<u>Principal Paragraph</u>
R53	Sighting Mark	5.6.2.1, 5.6.2.2
R54	Sighting Data Display	5.6.2.2
R55	Gyro Torquing	5.6.2.2
R56	Alternate LOS Sighting Mark	5.6.2.3, 5.6.2.4
R60	Attitude Maneuver	
R61	Tracking Attitude	5.2.5.1
R62	Crew-defined Maneuver	
R63	Rendezvous Final Attitude	5.6.7.3
R64	Optics Angles Transform	5.6.9
R67	Rotation	5.2.5.1.3

5.1.3 Paragraph 5.1.3 has been deleted from Skylark.

5.1.4 Coordinate Systems

Six major coordinate systems are used in the navigation and guidance programs for Skylark. These six coordinate systems are defined individually in the following descriptions and referenced to control specifications of paragraph 5.9.2 where applicable. Any other coordinate system used in a specific CMC program is defined in the subsection or paragraph describing that program.

5.1.4.1 Basic Reference Coordinate System

The Basic Reference Coordinate System is an orthogonal inertial coordinate system whose origin is located at the earth center of mass. The orientation of this coordinate system is defined by the line of intersection of the mean earth equatorial plane and the mean orbit of the earth (the ecliptic) at the beginning of the Besselian year that starts January 0.9233573, 1950, E. T. The X-axis (\underline{u}_{XI}) is along this intersection with the positive sense in the direction of the ascending node of the ecliptic on the equator (the equinox); the Z-axis (\underline{u}_{ZI}) is along the mean earth north pole; and the Y-axis (\underline{u}_{YI}) completes the right-handed triad. The Basic Reference Coordinate System is presented in reference 2 of paragraph 5.9.2 as the Ephemeris Coordinate System. Navigation stars and

the solar ephemeris are referenced to this coordinate system; vehicle state vectors are also referenced to this system.

5.1.4.2 IMU Stable Member or Platform Coordinate System

The orthogonal inertial coordinate system defined by the GNCS inertial measurement unit (IMU) is dependent upon the current IMU alignment. Many alignments are possible during a mission, but the primary IMU alignment orientations are described in paragraph 5.6.3.3, are summarized below, and are designated by the subscript SM. The origin of the IMU Stable Member Coordinate System is the center of the IMU stable member.

1. Preferred Alignment

$$\begin{aligned} \underline{u}_{XSM} &= \text{UNIT}(\underline{x}_B) \\ \underline{u}_{YSM} &= \text{UNIT}(\underline{u}_{XSM} \times \underline{r})^* \\ \underline{u}_{ZSM} &= \underline{u}_{XSM} \times \underline{u}_{YSM} \end{aligned} \quad (1.4.1)$$

where:

$$\left. \begin{array}{l} \underline{u}_{XSM} \\ \underline{u}_{YSM} \\ \underline{u}_{ZSM} \end{array} \right\} \begin{array}{l} \text{IMU stable member coordinate unit vectors} \\ \text{referenced to the Basic Reference Coordinate} \\ \text{System} \end{array}$$

\underline{x}_B = vehicle or body X-axis at the preferred vehicle attitude for ignition

\underline{r} = position vector at ignition

2. Nominal Alignment (Local Vertical)

$$\begin{aligned} \underline{u}_{XSM} &= (\underline{u}_{YSM} \times \underline{u}_{ZSM}) \\ \underline{u}_{YSM} &= \text{UNIT}(\underline{v} \times \underline{r}) \\ \underline{u}_{ZSM} &= \text{UNIT}(-\underline{r}) \end{aligned} \quad (1.4.2)$$

* In P41 if the magnitude of the cross product is less than 2^{16} meters or in P40 if the magnitude of $\underline{u}_{XSM} \times \text{UNIT} \underline{r}$ is less than 2^{-12} radians, then $\underline{u}_{YSM} = \text{UNIT}[\underline{u}_{XSM} \times (\text{UNIT} \underline{r} + 0.125 \text{ UNIT} \underline{v})]$ where \underline{v} = velocity vector at ignition.

3. Earth Prelaunch Alignment

Before earth launch the IMU stable member is aligned to a local vertical axis system where

$$\underline{u}_{ZSM} = \text{UNIT}(\underline{-r}) \text{ (local vertical)}$$

$$\underline{u}_{XSM} = \text{UNIT}(\underline{A}) \text{ where } \underline{A} \text{ is a horizontal vector pointed at the desired launch azimuth angle.}$$

$$\underline{u}_{YSM} = \underline{u}_{ZSM} \times \underline{u}_{XSM}$$

4. Plane Change Alignment

Whenever the CSM performs a plane change maneuver to the thrust attitude during MINKEY, the Z-gyro is pulse torqued either through plus 45 deg before, and minus 45 deg after, the maneuver or vice versa. The choice is made as follows. If $\cos(\text{CDUY}) \text{ VGY}_{LV} \geq 0$ the Z-gyro is pulse torqued minus 45 deg and then plus 45 deg. Otherwise the reverse is done.

$$\underline{u}_{XSM} = \text{UNIT}(\underline{u}_{XSM0} \cos 45^\circ - \underline{u}_{YSM0} \sin 45^\circ)$$

(minus 45° then plus 45°)

or

$$\underline{u}_{XSM} = \text{UNIT}(\underline{u}_{XSM0} \cos 45^\circ + \underline{u}_{YSM0} \sin 45^\circ)$$

(plus 45° then minus 45°)

$$\underline{u}_{YSM} = \text{UNIT}(\underline{u}_{ZSM0} \times \underline{u}_{XSM})$$

$$\underline{u}_{ZSM} = \underline{u}_{ZSM0}$$

where 0 refers to the orientation existing before plane change alignment.

5.1.4.3 Vehicle or Body Coordinate System

The Vehicle or Body Coordinate System is the orthogonal coordinate system used for the CSM structural body. The origin of this coordinate system is on the longitudinal axis of the CSM, 1000 inches below the mold line of the heat shield main structure ablator interface. The X-axis (u_{XB}) lies along the longitudinal axis of the vehicle, positive in the nominal SPS thrust direction. The positive Z-axis (u_{ZB}) is defined by an alignment target (labeled +Z) at the top of the service module and is normal to u_{XB} . The Y-axis (u_{YB}) completes the right-handed triad. This coordinate system is defined in reference 1 of paragraph 5.9.2 as the Standard Coordinate System 8c, CSM Structural Body Axes in Figure A-8c.

5.1.4.4 The Earth-fixed Coordinate System

The Earth-fixed Coordinate System is an orthogonal rotating coordinate system whose origin is at the center of mass of the earth. The Z-axis of this coordinate system is defined to be along the earth's true rotational or polar axis (neglecting the instantaneous motion of the pole). The X-axis is defined to be along the intersection of the prime (Greenwich) meridian and the equatorial plane of the earth, and the Y-axis is in the equatorial plane and completes the right-handed triad.

5.1.4.5 Apollo Telescope Mount Sensor Coordinate System

The Apollo Telescope Mount (ATM) Sensor Coordinate System is an orthogonal right-handed coordinate system whose axes are defined with respect to the ATM acquisition sun sensor and the ATM star tracker. The +Z axis is defined to be in the same direction as the ATM acquisition sun sensor; and the -Y axis is defined to be in the same direction as the ATM star tracker when its gimbal angles are zero. It is assumed that the star tracker is perpendicular to the +Z axis when its inner gimbal angle is zero. When the CSM is docked to the workshop, the orientation of this coordinate system with respect to the Navigation Base Coordinate System can be computed by the ATM Orientation Determination Program (P50) of paragraph 5.6.2.5.

5.1.4.6 Navigation Base Coordinate System

The Navigation Base Coordinate System (subscript NB) is an orthogonal coordinate system whose orientation is essentially parallel to that of the CSM Vehicle Coordinate System. The Y_{NB} axis is defined by the centers of the two upper mounts between GNCS navigation base and the CM structure located at vehicle station points $X_C = 71.185$ and $Z_C = 35.735$ in reference 13 of paragraph 5.9.2. The positive Y_{NB} direction is in the same general direction as the CSM +Y axis. The Z_{NB} axis is defined as a line measured from the center line of the optics (shown in Section A-A of reference 1) through an angle of $32^\circ 31' 23.19''$ about the Y_{NB} axis towards the vehicle +Z axis, and located on the Y_{NB} axis half way between the mount points. The positive Z_{NB} axis is in the same general direction as the vehicle +Z axis. The X_{NB} axis is defined as $Y_{NB} \times Z_{NB}$ to complete the right-handed triad.

5.1.5 General Definitions and Conventions

This paragraph defines a selected number of parameters and gives the conventions for using them. Although virtually all the information given here appears elsewhere in this document, this paragraph provides a summary of facts otherwise distributed among various other sections.

5.1.5.1 Error Transition Matrix Maintenance

5.1.5.1.1 Definitions.—The error transition matrix (W-matrix) is defined in paragraph 5.2.2.4 and is used in processing navigation-measurement data. Control of the W-matrix is maintained by means of the W-matrix control flag $RENDWFLG$. (See paragraph 5.2.5.2.2.) If $RENDWFLG$ is equal to one, the W-matrix is valid for processing rendezvous navigation data. If $RENDWFLG$ is equal to zero, the W-matrix is invalid; it will be reinitialized, however, by R22.

5.1.5.1.2 W-Matrix Control Flag.—The W-matrix control flag ($RENDWFLG$) is maintained according to the following rules:

1. $RENDWFLG$ is initially zero.
2. A CSM or WS state-vector update from the ground causes $RENDWFLG$ to be zeroed.
3. A special DSKY verb exists, which the astronaut can use to zero $RENDWFLG$.

4. Initialization of the W-matrix for rendezvous navigation causes RENDWFLG to be set to one.
5. If a computation overflow occurs during extrapolation of the W-matrix, a program alarm results, RENDWFLG is zeroed, and the extrapolation of the state vector continues without the W-matrix.

With regard to item 4 above, a set of initialization parameters for the W-matrix exists in erasable memory; this set consists of two elements, a position element and a velocity element. At the time each item of navigation data is processed, the W-matrix control flag is tested. If the flag is found to be zero, the W-matrix is initialized, consistent with the appropriate erasable parameters, and the flag is set to one. See paragraph 5.2.5.2.2 for precise details of this initialization procedure.

5.1.5.1.3 W-Matrix Extrapolation. — Extrapolation of the W-matrix is described in paragraph 5.2.2.4. Required in this extrapolation is the specification of the appropriate vehicle's state vector with which the W-matrix is extrapolated. Extrapolation occurs during programs P00 and P20, and at the conclusion of programs P40, P41, P47 and P48. The conventions under which the extrapolation occurs during each of these programs are as follows:

<u>Program</u>	<u>Convention</u>
P00	The W-matrix is extrapolated with the WS state vector if RENDWFLG is equal to one; it is not extrapolated if RENDWFLG is equal to zero. (See paragraph 5.6.12.)
P20 (Options 1, 2, 5)	The W-matrix is extrapolated with the state vector that is being updated if RENDWFLG is equal to one; it is not extrapolated if RENDWFLG is equal to zero. (See paragraph 5.2.5.2.2.)

Program

Convention

P11	The result of the maneuver is a final
P40	state vector at the end-of-maneuver time t_F .
P41	The CSM state vector that existed before the
P47	maneuver program will still exist, as will
P48	the pre-maneuver WS state vector and
	W-matrix. The following steps are per-
	formed after the Average-G Routine
	(paragraph 5.3.2) is terminated:
	1. If RENDWFLG is equal to one, the old
	CSM state vector and the W-matrix are
	extrapolated to time t_F .
	2. The CSM state vector is initialized to the
	end-of-maneuver state vector.
	3. The WS state vector is extrapolated to
	time t_F .

If a computation overflow occurs during any of the above W-matrix extrapolations, a program alarm results, RENDWFLG is zeroed, and the extrapolation of the state vector continues without the W-matrix.

5.1.5.2 Altitude Parameters Convention

In the following programs and routines, the display parameters of the vehicle altitude or trajectory pericenter or apocenter altitude are measured with respect to the earth launch-pad radius magnitude r_{LP} , which is stored in fixed memory.

P11	Earth Orbit Insertion Monitor Program	paragraph 5.3.5
P21	Ground Track Determination Program	paragraph 5.6.5
P29	Time-of-Longitude Program	paragraph 5.5.15
P30	External Δv Maneuver Guidance	paragraph 5.3.3.3.1
P47	Thrust Monitor Program	paragraph 5.3.4
R30	Orbital Parameters Display Routine	paragraph 5.6.10.1

In the Entry Preparation Program (P61) (paragraph 5.6.10), the 400,000 feet and EMS altitude parameters are referenced to the Fischer ellipsoid.

5.1.5.3 Paragraph 5.1.5.3 has been deleted from Skylark.

5.1.5.4 Paragraph 5.1.5.4 has been deleted from Skylark.

5.1.5.5 Time Definitions

The time t_0 is the time interval between the astrodynamic epoch and the instant when the guidance computer clock was zeroed. The astrodynamic epoch is the time point corresponding to the initial earth-orientation angle A_{Z0} in the Planetary Inertial Orientation Subroutine (paragraph 5.5.2) and the initial solar longitude LOS_0 and mean anomaly $PHASE_C$ in the Solar Ephemerides Routine (paragraph 5.5.4). The astrodynamic epoch is nominally chosen to be the Greenwich midnight universal time that begins the July 1 that precedes the launch, but may be any other time before liftoff if these three erasables are suitably loaded.

The time t (also known as mission time or ground elapsed time GET) is the time maintained by the guidance computer clock, and hence is the time interval between the zeroing of this clock and the current time.

Shortly before launch, the mission time t is zeroed and t_0 is established utilizing the ground check out system ACE and the uplink (VERB 55) and downlink. Within 0.5 sec of the time the computer receives the liftoff discrete, mission time t is again zeroed and t_0 is incremented by the elapsed time since the last zeroing of mission time. (See program P11.) During a flight, the mission time t can be adjusted (1) as a convenience for the flight controllers so that events can occur at the mission time scheduled in the Flight Plan, and (2) to correct possible drift in the AGC clock. The adjustment is performed by the use of Extended Verb 70; an increment of time is added or subtracted from mission time t , and is subtracted or added, respectively, to t_0 .

BLANK

5.2 COASTING FLIGHT NAVIGATION

5.2.1 General Comments

The CMC coasting flight navigation routines that are presented in this subsection are used during freefall phases of the Skylab mission. The basic objective of these navigation routines is to maintain estimates of the position and velocity vectors of both the CSM and the WS. Let \underline{r} and \underline{v} be the estimates of a vehicle's position and velocity vectors, respectively. Then, the six-dimensional state vector, \underline{x} , of the spacecraft is defined by

$$\underline{x} = \begin{pmatrix} \underline{r} \\ \underline{v} \end{pmatrix}$$

Coasting flight navigation is accomplished by extrapolating the state vector, \underline{x} , by means of the Coasting Integration Routine (paragraph 5.2.2), and updating or modifying this estimated state vector using tracking data by the recursive method of navigation (paragraphs 5.2.3–5.2.5).

The Coasting Integration Routine is used by other navigation and targeting routines to extrapolate the following:

1. Present estimated CSM state vector
2. Present estimated WS state vector
3. An arbitrary specified state vector, such as temporary state vectors used in intermediate calculations.

State vector extrapolation is accomplished by means of Encke's method of differential accelerations. The motion of a spacecraft is dominated by the conic orbit that would result if the spacecraft were in a central force field. In Encke's method the differential equations for the deviations from conic motion are integrated numerically. This technique is in contrast to a numerical integration of the differential equations for the total motion, and it provides a more accurate orbit extrapolation. The numerical integration is accomplished by means of Nystrom's method, which gives fourth-order accuracy while requiring only three computations of the derivatives per time step. The usual fourth-order Runge-Kutta integration methods require four derivative computations per time step.

Regardless of the accuracy of the state vector extrapolation, errors in the initial conditions will propagate and soon grow to intolerable size. Thus, it is necessary periodically to obtain additional data in the form of either new state vector estimates or modifications to the current state-vector estimates. These state-vector modifications are computed from navigation data obtained by means of navigation measurements.

During the rendezvous phase the CSM GNCS normally uses optical angle data from the sextant (SXT) and VHF range data to compute state-vector changes. The six-dimensional state vector of either the CSM or the WS can be updated from the measurement data obtained with the CSM-based sensors. Normally the CSM state vector is updated, but the astronaut can select the WS update mode. The selection of the update mode is based primarily upon which vehicle's state vector is most accurately known initially.

Navigation data is incorporated into the state vector estimates by means of the Measurement Incorporation Routine (paragraph 5.2.3), which is also a subroutine of the Rendezvous Navigation Routine (paragraph 5.2.5).

The navigation procedure, which is illustrated in simplified form in Figure 2.1-1, involves computing an estimated tracking measurement, \hat{Q} , based on the current state-vector estimates. This estimated measurement is then compared with the actual tracking measurement Q to form a measured deviation δQ . A statistical weighting vector, $\underline{\omega}$, is computed from statistical knowledge of state-vector uncertainties and tracking performance, $\overline{\sigma^2}$, plus a geometry vector, \underline{b} , determined by the type of measurement being made. The weighting vector, $\underline{\omega}$, is defined such that a statistically optimum linear estimate of the deviation, $\delta \underline{x}$, from the estimated state vector is obtained when the weighting vector is multiplied by the measured deviation δQ . The vectors $\underline{\omega}$, \underline{b} and $\delta \underline{x}$ are six dimensional.

In an attempt to prevent unacceptably large incorrect state-vector changes, a validity test has been included in the rendezvous procedure.

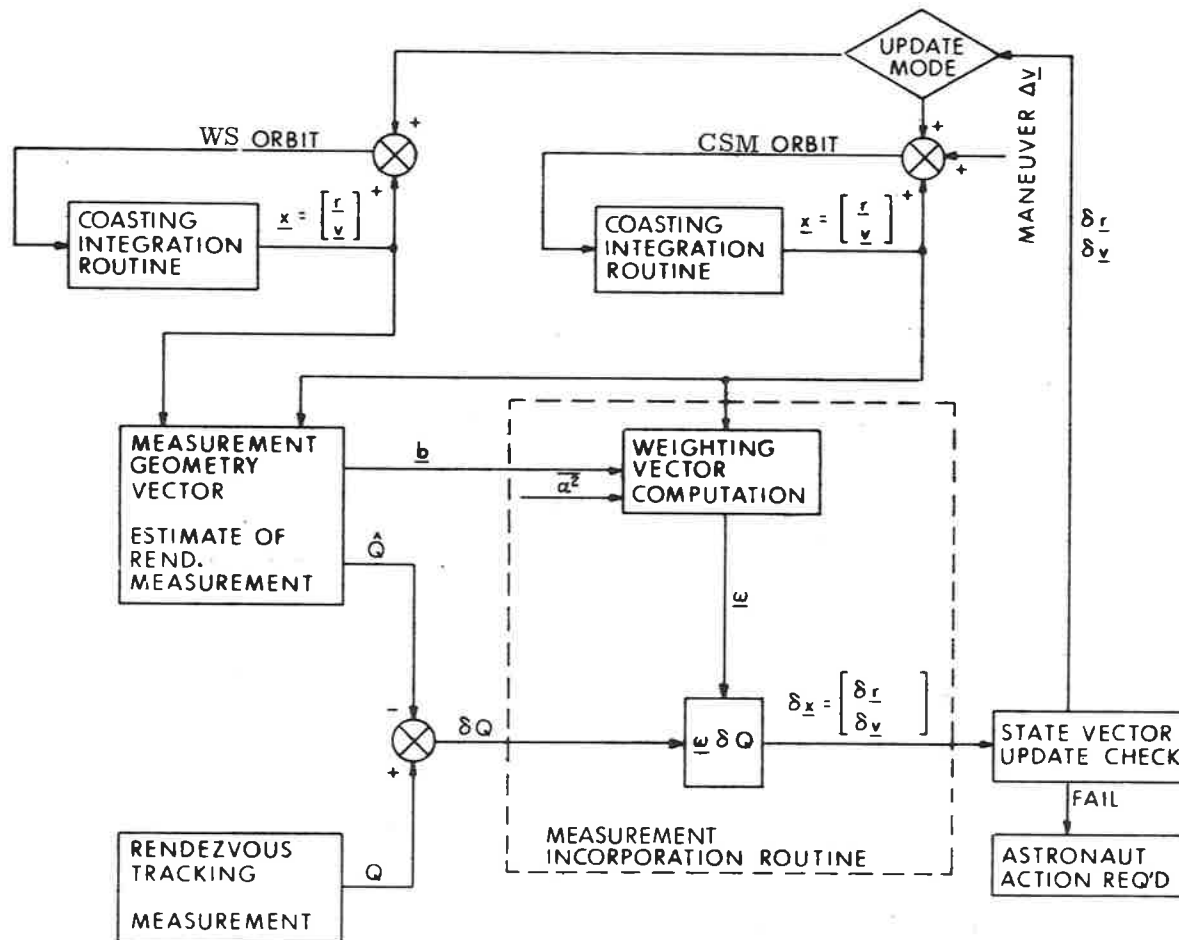


Figure 2.1-1. Simplified CMC Rendezvous Navigation Functional Diagram

In the Rendezvous Navigation Routine (paragraph 5.2.5), measurement data are processed periodically; and it is desirable that the WS be tracked during the entire rendezvous phase up to the manual terminal maneuver. If the magnitudes of the changes in the estimated position and velocity vectors, δr and δv , respectively, are both less than preset tracking-alarm levels, then the selected vehicle's state vector is automatically updated by the computed deviation, $\delta \underline{x}$. No special display is presented, but the tracking-measurement counter is incremented by one. If either δr or δv exceeds its alarm level, then the state vector is not updated, and the astronaut is alerted to this condition by a special display of δr and δv . Included in this display is the source code that indicates whether optical or VHF range-link data caused the display.

If this display should occur because of optics data, the astronaut should recheck the optical tracking and be sure that he is actually tracking the WS. Under certain conditions it is possible to mistake a star for WS reflected sunlight; and it may take a period of a few minutes to determine the WS target by watching the relative motion of the target and its star background. After the tracking has been verified, and navigation data have again been acquired, the astronaut has the option of commanding a state-vector update if the tracking alarm is again exceeded, or of repeating further optical checks before incorporating the measurement data. If the astronaut cannot determine the WS target, because of lack of positive acquisition, (for example, bright background), he can terminate the marking procedure and try to achieve better tracking conditions at a later time.

The displayed values of δr and δv that have not passed the tracking-alarm test will depend upon the statistical parameters stored in the CMC and upon the errors made in the tracking. The tracking-alarm criterion is incorporated in the Rendezvous Navigation Routine to alert the astronaut to the fact that the state-vector update is larger than normally expected, and to prevent the estimated state vector from automatically being updated in such cases. The update occurs only by specific command of the astronaut. The tracking-alarm level beyond which updating is suspended is chosen primarily to avoid false acquisition and tracking conditions. As previously mentioned, this condition is possible in the CSM if a star is mistaken for WS reflected sunlight; and it is, therefore, possible for the alarm level to be exceeded in such cases although the estimated state vectors are essentially correct. It is also possible for the state-vector update alarm level to be exceeded after correct initial acquisition and tracking when a poor estimate of either the CSM or the WS state vector exists. In this case, the astronaut must command the initial state-vector updates—after which the alarm level is seldom exceeded during the remainder of the rendezvous phase.

The parameters required to initialize the navigation routines are (1) the initial estimated CSM state vector, (2) the initial estimated WS state vector, (3) the initial state-vector estimation error covariance matrix in the form of a prestored diagonal error-transition matrix (as defined in paragraph 5.2.2.4), and (4) a priori measurement-error variances. The basic inputs to the navigation routines are (1) SXT tracking-angle data indicated to the CMC by the astronaut when he presses the MARK button—signifying that he has centered the optical reticle on the WS, and, (2) the automatically acquired VHF range-link tracking data. The primary output of the navigation routines is either the estimated CSM state vector, or the estimated WS state vector. The guidance targeting modes outlined in subsection 5.4 are based on the state-vector estimates resulting from these navigation routines.

5.2.2 Coasting Integration Routine

5.2.2.1 General Comments

During all coasting-phase navigation procedures, an extrapolation of position and velocity by numerical integration of the equations of motion is required. The basic equation may be written in the form

$$\frac{d^2}{dt^2} \underline{r}(t) + \frac{\mu_E}{r^3} \underline{r}(t) = \underline{a}_d(t) \quad (2.2.1)$$

where μ_E is the gravitational constant of the earth, and $\underline{a}_d(t)$ is the vector acceleration that prevents the motion of the vehicle from being precisely a conic with focus at the center of the earth. The Coasting Integration Routine is a precision-integration routine in which all significant perturbation effects are included.

An approximate extrapolation of a vehicle state vector in which the disturbing acceleration, $\underline{a}_d(t)$ of Equation (2.2.1), is set to zero can be made by using the Kepler Subroutine (paragraph 5.5.5).

5.2.2.2 Encke's Method

If \underline{a}_d is small compared with the central force field, direct integration of Equation (2.2.1) is inefficient. The extrapolation is therefore made using the technique of differential accelerations attributed to Encke.

At time t_0 the position and velocity vectors, \underline{r}_0 and \underline{v}_0 , define an osculating conic orbit. The position and velocity vectors in the conic orbit, $\underline{r}_{\text{con}}(t)$ and $\underline{v}_{\text{con}}(t)$, respectively, deviate by a small amount from the actual position and velocity vectors.

The conic position and velocity at time t are computed as shown in paragraph 5.5.5. Required in this calculation is the variable x which is the root of Kepler's equation. In order to minimize the number of iterations required in solving Kepler's equation, an estimate of the correct solution for x is obtained as follows:

Let

$$\tau = t - t_0 \quad (2.2.2)$$

During the previous computation cycle, the values

$$\begin{aligned} \underline{r}' &= \underline{r}_{\text{con}} \left(\tau - \frac{\Delta t}{2} \right) \\ \underline{v}' &= \underline{v}_{\text{con}} \left(\tau - \frac{\Delta t}{2} \right) \\ x' &= x \left(\tau - \frac{\Delta t}{2} \right) \end{aligned} \quad (2.2.3)$$

were computed. A trial value of $x(\tau)$ is obtained from

$$x_t = x' + s \left[1 - \gamma s (1 - 2\gamma s) - \frac{1}{6} \left(\frac{1}{r'} - \alpha \right) s^2 \right] \quad (2.2.4)$$

where

$$\begin{aligned} s &= \frac{\sqrt{\mu_E}}{r'} \left(\frac{\Delta t}{2} \right) \\ \gamma &= \frac{\underline{r}' \cdot \underline{v}'}{2r' \sqrt{\mu_E}} \\ \alpha &= \frac{2}{r'} - \frac{(v')^2}{\mu_E} \end{aligned} \quad (2.2.5)$$

After specification of \underline{r}_0 , \underline{v}_0 , x_t and τ , the Kepler Subroutine (paragraph 5.5.5) is used to compute $\underline{r}_{\text{con}}(\tau)$, $\underline{v}_{\text{con}}(\tau)$, and $x(\tau)$.

The true position and velocity vectors deviate from the conic position and velocity since \underline{a}_d is not zero. Let

$$\underline{r}(t) = \underline{\delta}(t) + \underline{r}_{\text{con}}(t) \quad (2.2.6)$$

$$\underline{v}(t) = \underline{\nu}(t) + \underline{v}_{\text{con}}(t)$$

where $\underline{\delta}(t)$ and $\underline{\nu}(t)$ are the position and velocity deviations from the conic. The deviation vector $\underline{\delta}(t)$ satisfies the differential equation

$$\frac{d^2}{dt^2} \underline{\delta}(t) = -\frac{\mu E}{r_{\text{con}}^3(t)} \left[f(q) \underline{r}(t) + \underline{\delta}(t) \right] + \underline{a}_d(t) \quad (2.2.7)$$

subject to the initial conditions

$$\underline{\delta}(t_0) = \underline{0}, \quad \underline{\nu}(t_0) = \underline{0} \quad (2.2.8)$$

where

$$q = \frac{(\underline{\delta} - 2\underline{r}) \cdot \underline{\delta}}{r^2} \quad (2.2.9)$$

$$f(q) = q \frac{3 + 3q + q^2}{1 + (1 + q)^{3/2}} \quad (2.2.10)$$

The first term on the right-hand side of Equation (2.2.7) must remain small, i. e., of the same order as $\underline{a}_d(t)$, if the method is to be efficient. As the deviation vector $\underline{\delta}(t)$ grows in magnitude, this term eventually increases in size. Therefore, in order to maintain the efficiency of the method, a new osculating conic orbit should be defined by the total position and velocity vectors $\underline{r}(t)$ and $\underline{v}(t)$. The process of selecting a new conic orbit from which to calculate deviations is called rectification. When rectification occurs, the initial conditions for the differential equation for $\underline{\delta}(t)$, as well as the variables τ and x , are again zero.

5.2.2.3 Disturbing Acceleration

The form of the disturbing acceleration $\underline{a}_d(t)$ that is used in Equation (2.2.1) can, in general, be quite complicated. For the Skylab Mission, however, only the gravitational perturbations arising from the earth's non-spherical shape are considered. Let \underline{a}_{dE} be the acceleration due to the non-spherical gravitational perturbations of the earth. Then,

$$\underline{a}_d = \underline{a}_{dE} = \frac{\mu_E}{r^2} \sum_{i=2}^4 J_{iE} \left(\frac{r_E}{r} \right)^i \left[P'_{i+1}(\cos \varphi) \underline{u}_r - P'_i(\cos \varphi) \underline{u}_Z \right] \quad (2.2.11)$$

where

$$\begin{aligned}
P_2'(\cos \varphi) &= 3 \cos \varphi \\
P_3'(\cos \varphi) &= \frac{1}{2} (15 \cos^2 \varphi - 3) \\
P_4'(\cos \varphi) &= \frac{1}{3} (7 \cos \varphi P_3' - 4P_2') \\
P_5'(\cos \varphi) &= \frac{1}{4} (9 \cos \varphi P_4' - 5P_3')
\end{aligned}
\tag{2.2.12}$$

are the derivatives of Legendre polynomials,

$$\cos \varphi = \underline{u}_r \cdot \underline{u}_Z$$

\underline{u}_Z = unit vector of the earth's true north polar axis expressed in the Basic Reference Coordinate System

$$\tag{2.2.13}$$

\underline{u}_r = unit vector in the direction of \underline{r} ,

J_2, J_3, J_4 are the coefficients of the second, third, and fourth harmonics of the earth's potential function; and r_E is the mean equatorial radius of the earth.

NOTE. — Equations (2.2.14)–(2.2.21) have been deleted.

5.2.2.4 Error-Transition Matrix

The position and velocity vectors as maintained in the computer are estimates of the true values. As part of the navigation technique, it is also necessary to maintain statistical data in the computer to aid in the processing of navigation measurements.

If $\underline{\epsilon}(t)$ and $\underline{\eta}(t)$ are the errors in the estimates of the position and velocity vectors, respectively, then the six-dimensional covariance matrix $E(t)$ is defined by

$$E(t) = \begin{pmatrix} \overline{\underline{\epsilon}(t) \underline{\epsilon}(t)^T} & \overline{\underline{\epsilon}(t) \underline{\eta}(t)^T} \\ \overline{\underline{\eta}(t) \underline{\epsilon}(t)^T} & \overline{\underline{\eta}(t) \underline{\eta}(t)^T} \end{pmatrix}
\tag{2.2.22}$$

NOTE. — Equation (2. 2. 23) has been deleted.

Rather than use the covariance matrix in the navigation procedure, it is more convenient to utilize a matrix $W(t)$, called the error-transition matrix, and defined by

$$E(t) = W(t) W(t)^T \quad (2. 2. 24)$$

Extrapolation of the six-dimensional matrix $W(t)$ is made by direct numerical integration of the differential equation

$$\frac{d}{dt} W(t) = \begin{pmatrix} O & I \\ G(t) & O \end{pmatrix} W(t) \quad (2. 2. 25)$$

where $G(t)$ is the three-dimensional gravity gradient matrix, and I and O are the three-dimensional identity and zero matrices, respectively. If the W matrix is partitioned as

$$W = \begin{pmatrix} \underline{w}_0 & \underline{w}_1 & \cdot & \cdot & \cdot & \underline{w}_5 \\ \underline{w}_6 & \underline{w}_7 & \cdot & \cdot & \cdot & \underline{w}_{11} \end{pmatrix} \quad (2. 2. 26)$$

then,

$$\left. \begin{aligned} \frac{d}{dt} \underline{w}_i(t) &= \underline{w}_{i+6}(t) \\ \frac{d}{dt} \underline{w}_{i+6}(t) &= G(t) \underline{w}_i(t) \end{aligned} \right\} i = 0, 1, \dots, 5 \quad (2. 2. 27)$$

The extrapolation may be accomplished by successively integrating the vector differential equations

$$\frac{d^2}{dt^2} \underline{w}_i(t) = G(t) \underline{w}_i(t) \quad i = 0, 1, \dots, 5 \quad (2.2.28)$$

The gravity gradient matrix $G(t)$ for earth orbit is given by

$$G(t) = \frac{\mu_E}{r^5(t)} \left[3\underline{r}(t) \underline{r}(t)^T - r^2(t) \mathbf{I} \right] \quad (2.2.29)$$

NOTE. — Equation (2.2.30) has been deleted.

Thus, the differential equations for the $\underline{w}_i(t)$ vectors are

$$\frac{d^2}{dt^2} \underline{w}_i(t) = \frac{\mu_E}{r^3(t)} \left\{ 3 \left[\underline{u}_r(t) \cdot \underline{w}_i(t) \right] \underline{u}_r(t) - \underline{w}_i(t) \right\} \quad (2.2.31)$$

$$i = 0, 1, \dots, 5$$

NOTE. — Equation (2.2.32) has been deleted.

It is possible for a computation overflow to occur during the W -matrix integration if any element of the matrix exceeds its maximum value. This event is extremely unlikely because of the large scale factors chosen. The overflow can occur, however, under one of the following conditions:

1. If any element of the position part (upper half) of the W -matrix becomes equal to or greater than 2^{19} m.
2. If any element of the velocity part (lower half) of the W -matrix becomes equal to or greater than one m/csec.

If overflow should occur, an alarm results, the W -matrix control flag is reset, and either new state-vector estimates must be obtained from RTCC, or a sufficient number of navigation measurements must be made before the state vectors are used in any targeting or maneuver programs.

5. 2. 2. 5 Numerical Integration Method

The extrapolation of navigational data requires the solution of a number of second-order vector differential equations, specifically Equations (2. 2. 7) and (2. 2. 31). These are all special cases of the form

$$\frac{d^2}{dt^2} \underline{y} = \underline{f}(\underline{y}, t) \quad (2. 2. 33)$$

Nystrom's method is particularly well suited to this form and gives an integration method of fourth-order accuracy. The second-order system is written

$$\frac{d}{dt} \underline{y} = \underline{z} \quad (2. 2. 34)$$

$$\frac{d}{dt} \underline{z} = \underline{f}(\underline{y}, t)$$

and the formulas are summarized below.

$$\begin{aligned} \underline{y}_{n+1} &= \underline{y}_n + \underline{\phi}(\underline{y}_n) \Delta t \\ \underline{z}_{n+1} &= \underline{z}_n + \underline{\psi}(\underline{y}_n) \Delta t \\ \underline{\phi}(\underline{y}_n) &= \underline{z}_n + \frac{1}{6} (\underline{k}_1 + 2\underline{k}_2) \Delta t \\ \underline{\psi}(\underline{y}_n) &= \frac{1}{6} (\underline{k}_1 + 4\underline{k}_2 + \underline{k}_3) \\ \underline{k}_1 &= \underline{f}(\underline{y}_n, t_n) \\ \underline{k}_2 &= \underline{f}\left(\underline{y}_n + \frac{1}{2} \underline{z}_n \Delta t + \frac{1}{8} \underline{k}_1 (\Delta t)^2, t_n + \frac{1}{2} \Delta t\right) \\ \underline{k}_3 &= \underline{f}\left(\underline{y}_n + \underline{z}_n \Delta t + \frac{1}{2} \underline{k}_2 (\Delta t)^2, t_n + \Delta t\right) \end{aligned} \quad (2. 2. 35)$$

For efficient use of computer storage as well as computing time, the computations are performed in the following order:

1. Equation (2.2.7) is solved using the Nystrom formulas, Equation (2.2.35). It is necessary to preserve the values of the vector \underline{r} at times t_n , $t_n + \Delta t/2$, $t_n + \Delta t$ for use in the solution of Equations (2.2.31).
2. Equations (2.2.31) are solved one-at-a-time using Equations (2.2.35) together with the values of \underline{r} that resulted from the first step.

The variable Δt is the integration time step and should not be confused with τ , the time since rectification. The maximum value for Δt that can be used for precision integration, Δt_{\max} , is computed from

$$\Delta t_{\max} = \text{minimum} \left(\Delta t_{\text{lim}}, \frac{Kr^{3/2}_{\text{con}}}{\sqrt{\mu_E}} \right) \quad (2.2.36)$$

where

$$\Delta t_{\text{lim}} = 4000 \text{ sec} \quad (2.2.37)$$

$$K = 0.3$$

5.2.2.6 Coasting Integration Logic

Estimates of the state vectors of two vehicles (CSM and WS) are maintained in the computer. Various phases of the mission will require extrapolation of a state vector either alone or with an associated W-matrix of six dimensions.

To accomplish all of these possible procedures, as well as to solve the computer restart problem, three state vectors are maintained in the computer. Let \underline{x}_C and \underline{x}_W be the estimated CSM and WS state vectors, respectively, and let \underline{x} be a temporary state vector. The state vector \underline{x} is a symbolic representation of the following set of variables:

$$\begin{aligned}
 \underline{r}_0 &= \text{rectification position vector} \\
 \underline{v}_0 &= \text{rectification velocity vector} \\
 \underline{r}_{\text{con}} &= \text{conic position vector} \\
 \underline{v}_{\text{con}} &= \text{conic velocity vector} \\
 \underline{\delta} &= \text{position deviation vector} && (2. 2. 38) \\
 \underline{\nu} &= \text{velocity deviation vector} \\
 t &= \text{time associated with } \underline{r}_{\text{con}}, \underline{v}_{\text{con}}, \underline{\delta} \text{ and } \underline{\nu} \\
 \tau &= \text{time since rectification} \\
 x &= \text{root of Kepler's equation}
 \end{aligned}$$

The state vectors \underline{x}_C and \underline{x}_W represent an analogous set of variables.

The Coasting Integration Routine is controlled by the calling program by means of the two indicators D and V. The Flag D indicates if both the state vector and the W-matrix are to be extrapolated (D = 6), or if the only state vector is to be extrapolated (D = 0).

NOTE. — Equation (2. 2. 39) has been deleted.

The variable V indicates the appropriate vehicle as follows:

$$V = \begin{cases} 1 & \text{for CSM} \\ 0 & \text{for WS} \\ -1 & \text{for state vector specified by calling program.} \end{cases} \quad (2. 2. 40)$$

In addition, the calling program must set the desired final time t_F ; and, for V equal to -1, the desired state vector \underline{x} .

There is a procedure for the emergency termination of the Coasting Integration Routine in order to permit correction of wrong erasable memory parameters. This emergency function is described in paragraph 5. 6. 12.

A simplified functional diagram of the Coasting Integration Routine is shown in Figure 2.2-1. In the figure the indicated state vector is being integrated to time t_F . The value of Δt for each time step is Δt_{\max} [Equation (2.2.36)] or the total time-to-go—whichever is smaller. The integration is terminated when the computed value of Δt is less than ϵ_t .

Figure 2.2-2 illustrates in more detail the logic flow of this routine. This figure also illustrates certain items that have not been fully discussed in the text. In particular,

1. The saving of \underline{r} values for W-matrix integration
2. The rectification procedure.

The flag F is used to distinguish between state vector integration (F = 1) and W-matrix integration (F = 0).

In addition to the general criterion discussed in paragraph 5.2.2.2, the requirements for rectification (which are not shown in Figure 2.2-2) are functions of the following:

1. Computer word length
2. Fact that the computations are performed in fixed-point arithmetic
3. Scale factors of the variables
4. Accuracy of the Kepler Subroutine.

If

$$\frac{\delta}{r_{\text{con}}} > 0.01$$

or if

$$\delta > 0.75 \times 2^{22} \text{ m}$$

or if

$$\nu > 0.75 \times 2^3 \text{ m/csec}$$

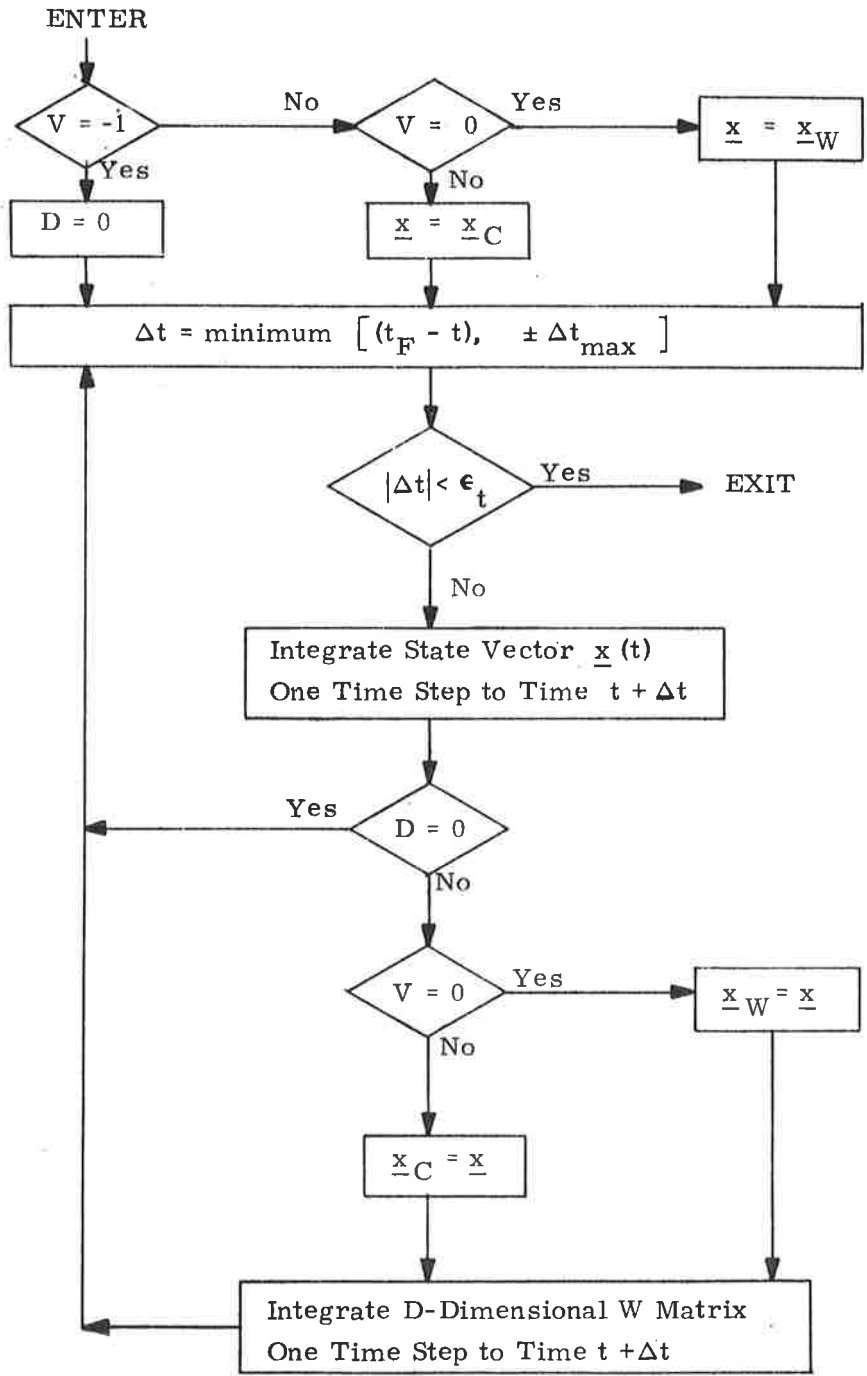


Figure 2. 2-1. Simplified Coasting Integration Routine Logic Diagram

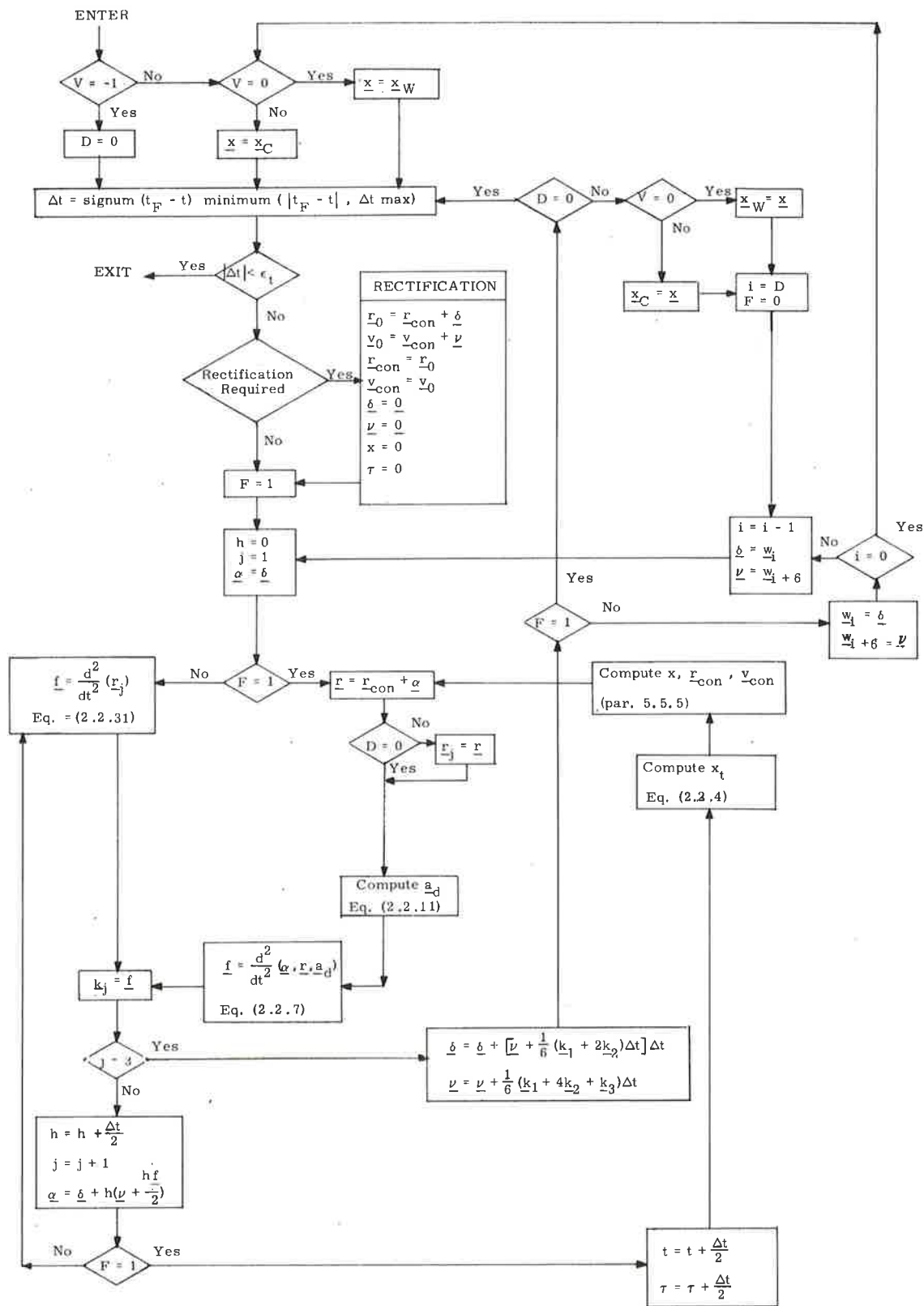


Figure 2.2-2. Coasting Integration Routine Logic Diagram

then rectification occurs at the point indicated in Figure 2.2-2. Also, if the calculation of the acceleration [Equation (2.2.7)] results in overflow (i. e. any component is equal to or greater than 2^{-16}m/csec^2), then the program is recycled to the beginning of the time step and rectification is performed, provided that δ is not identically zero (which may occur if an attempt is made to extrapolate a state vector below the surface). In this exceptional case, an abort occurs with alarm code 20430.

5.2.3 Measurement Incorporation Routine

The estimated position and velocity vectors of the vehicle (CSM or WS) must be periodically updated by means of navigation measurements. At the time a measurement is made, the best estimate of the state vector of the spacecraft is the extrapolated estimate denoted by \underline{x}' . The six components of \underline{x}' are the components of the estimated position and velocity vectors. From this state vector estimate, it is possible to determine an estimate of the quantity measured. When the predicted value of this measurement is compared with the actual measured quantity, the difference is used to update the indicated state vector as well as its associated error transition matrix, as described in paragraph 5.2.1. The error transition matrix, W , is defined in paragraph 5.2.2.4.

The Measurement Incorporation Routine is used to compute deviations to be added to the components of the estimated state vector, and to update the estimated state vector by these deviations provided the deviations pass a state vector update validity test as described in paragraph 5.2.1.

Associated with each measurement are the following parameters, which are to be specified by the program calling this routine:

\underline{b} = Geometry vector of 6 components

$\overline{\alpha^2}$ = A priori measurement error variance

δQ = Measured deviation, the difference between the quantity actually measured and the expected value based on the original value of the estimated state vector \underline{x}' .

The procedure for incorporating measurements into the estimated state vector is as follows:

- 1 Compute a 6-dimensional \underline{z} vector from

$$\underline{z} = W'^T \underline{b} \quad (2.3.1)$$

where W' is the error transition matrix associated with \underline{x}' .

- 2 Compute the 6-dimensional weighting vector, $\underline{\omega}$, from

$$\underline{\omega}^T = \frac{1}{z^2 + \alpha^2} \underline{z}^T W'^T \quad (2.3.2)$$

- 3 Compute the state vector deviation estimates from

$$\delta \underline{x} = \underline{\omega} \delta Q \quad (2.3.3)$$

- 4 If the data pass the validity test, update the state vector and the W matrix by

$$\underline{x} = \underline{x}' + \delta \underline{x} \quad (2.3.4)$$

$$W = W' - \frac{\underline{\omega} \underline{z}^T}{1 + \sqrt{\frac{\alpha^2}{z^2 + \alpha^2}}} \quad (2.3.5)$$

In order to take full advantage of the three-dimensional vector and matrix operations provided by the interpreter in the computer, the six-dimensional W -matrix is partitioned into twelve three-dimensional vectors:

$$\underline{w}_0, \underline{w}_1, \dots, \underline{w}_{11}$$

Refer to paragraph 5.2.2.4 for the definition of the W matrix. Define the three-dimensional matrices

$$W_0 = \begin{pmatrix} \underline{w}_0^T \\ \underline{w}_1^T \\ \underline{w}_2^T \end{pmatrix} \quad W_1 = \begin{pmatrix} \underline{w}_3^T \\ \underline{w}_4^T \\ \underline{w}_5^T \end{pmatrix} \quad W_2 = \begin{pmatrix} \underline{w}_6^T \\ \underline{w}_7^T \\ \underline{w}_8^T \end{pmatrix} \quad W_3 = \begin{pmatrix} \underline{w}_9^T \\ \underline{w}_{10}^T \\ \underline{w}_{11}^T \end{pmatrix} \quad (2.3.6)$$

so that

$$W = \begin{pmatrix} W_0^T & W_1^T \\ W_2^T & W_3^T \end{pmatrix} \quad (2.3.7)$$

Let the six-dimensional vectors $\delta \underline{x}$, \underline{b} , $\underline{\omega}$, and \underline{z} be partitioned as follows:

$$\delta \underline{x} = \begin{pmatrix} \delta \underline{x}_0 \\ \delta \underline{x}_1 \end{pmatrix} \quad \underline{b} = \begin{pmatrix} \underline{b}_0 \\ \underline{b}_1 \end{pmatrix} \quad \underline{\omega} = \begin{pmatrix} \underline{\omega}_0 \\ \underline{\omega}_1 \end{pmatrix} \quad \underline{z} = \begin{pmatrix} z_0 \\ z_1 \\ \vdots \\ z_5 \end{pmatrix} = \begin{pmatrix} \underline{z}_0 \\ \underline{z}_1 \end{pmatrix} \quad (2.3.8)$$

Then, the computations shown in Equations (2.3.1) through (2.3.3) are performed as follows, using three-dimensional operations:

$$\underline{z}_0 = W'_0 \underline{b}_0 + W'_2 \underline{b}_1$$

$$\underline{z}_1 = W'_1 \underline{b}_0 + W'_3 \underline{b}_1$$

$$a = \underline{z}_0 \cdot \underline{z}_0 + \underline{z}_1 \cdot \underline{z}_1 + \alpha^2$$

$$\underline{e}_0^T = \frac{1}{a} \left[\underline{z}_0^T W'_0 + \underline{z}_1^T W'_1 \right] \quad (2.3.9)$$

$$\underline{e}_1^T = \frac{1}{a} \left[\underline{z}_0^T W'_2 + \underline{z}_1^T W'_3 \right]$$

$$\delta \underline{x}_0 = \delta Q \underline{\omega}_0$$

$$\delta \underline{x}_1 = \delta Q \underline{\omega}_1$$

Equation (2.3.5) is written

$$\nu = \frac{1}{1 + \sqrt{\alpha^2/a}}$$

(2.3.10)

$$\underline{w}_{i+6j} = \underline{w}'_{i+6j} - \nu z_i \underline{\omega}_j \quad \begin{pmatrix} i = 0, 1, \dots, 5 \\ j = 0, 1 \end{pmatrix}$$

The Measurement Incorporation Routine is divided into two sub-routines, INCORP1 and INCORP2. The subroutine INCORP1 consists of Equations (2.3.9), while INCORP2 is composed of Equations (2.3.4) and (2.3.10). The method of using these subroutines is illustrated in Figure 2.3-1.

Since the estimated position and velocity vectors are maintained in two pieces, conic and deviation from the conic, Equation (2.3.4) cannot be applied directly. The estimated position and velocity deviations resulting from the measurement, $\delta \underline{x}_0$ and $\delta \underline{x}_1$, are added to the vectors $\underline{\delta}$ and $\underline{\nu}$, the position and velocity deviations from the conics, respectively. Since $\underline{\delta}$ and $\underline{\nu}$ are maintained to much higher accuracy than the conic position and velocity vectors, a possible computation overflow situation exists whenever Equation (2.3.4) is applied. If overflow does occur, then it is necessary to reinitialize the Coasting Integration Routine (paragraph 5.2.2) by the process of rectification as described in paragraph 5.2.2.2. The logic flow of the subroutine INCORP2 is illustrated in detail in Figure 2.3-2.

Overflow occurs when

$$\delta \geq 2^{22} \text{ m}$$

or

$$\nu = 2^3 \text{ m/csec}$$

5.2.4 Paragraph 5.2.4 has been deleted.

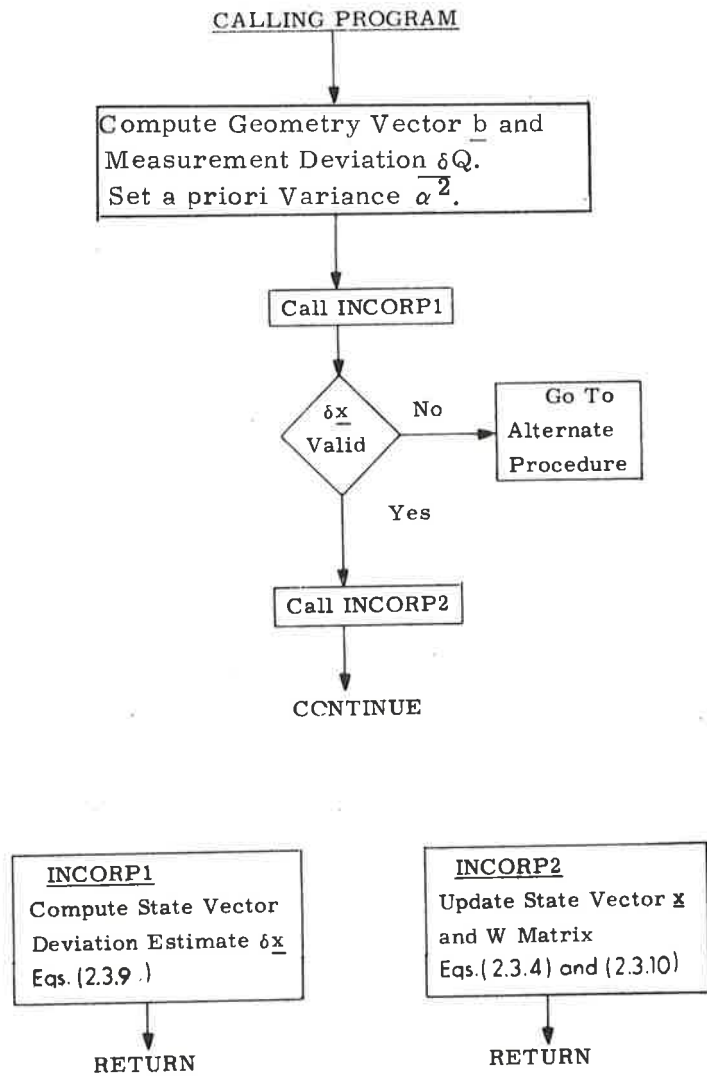


Figure 2.3-1. Measurement Incorporation Procedure

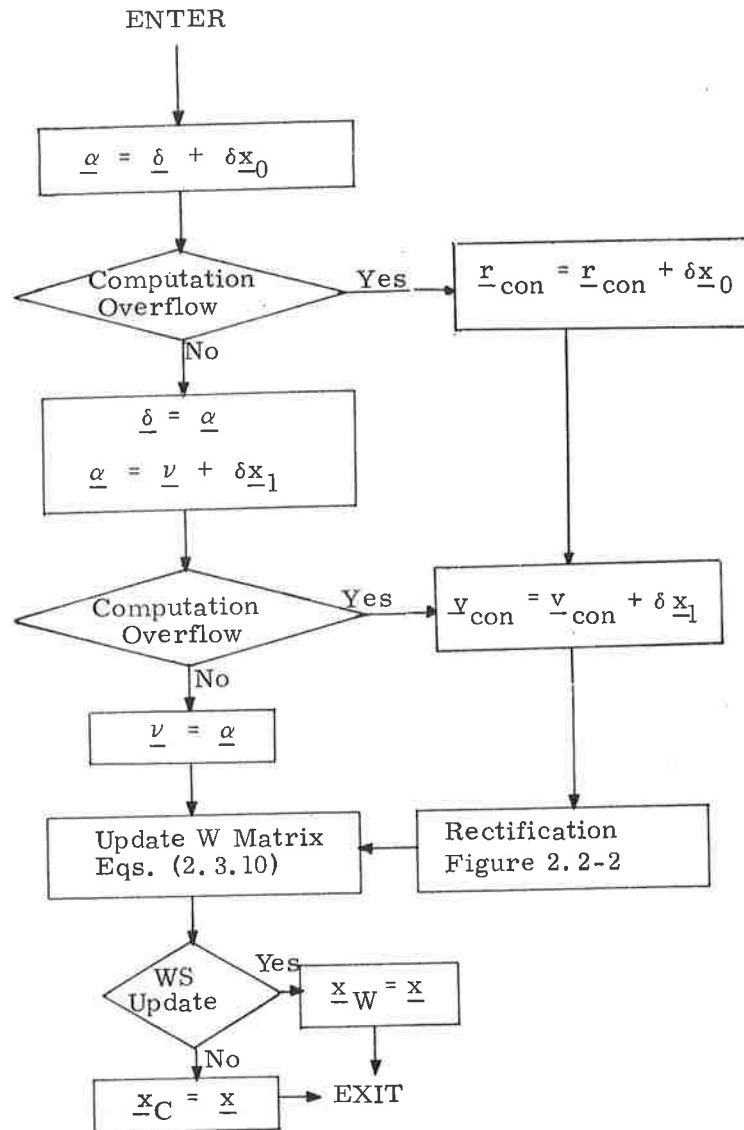


Figure 2,3-2. INCORP2 Subroutine Logic Diagram

5.2.5 Universal Tracking and Rendezvous Navigation Program (P20)

The CMC Universal Tracking and Rendezvous Navigation Program (P20) has two independent functions—tracking and navigation. The tracking function applies to any mission situation requiring the CSM to track a designated body with a specified pointing vector or to any mission situation requiring the CSM to rotate at a constant rate around a specified vector. The navigation function applies only to the Rendezvous Navigation Mode of P20.

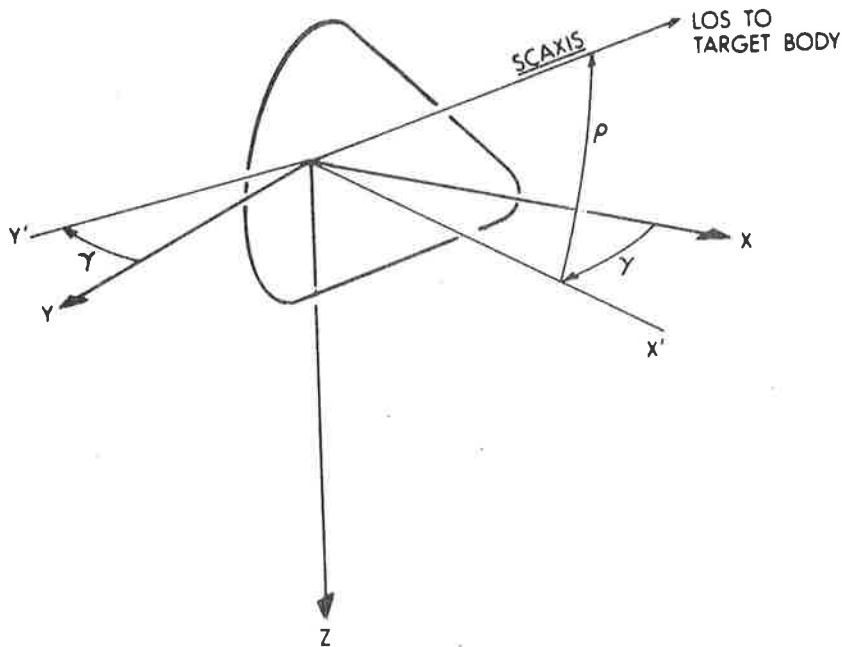
5.2.5.1 Tracking

There are three nonrendezvous and two rendezvous tracking (or pointing) options:

<u>Option</u>	<u>Purpose</u>	<u>Mode</u>
0	Point a specified spacecraft vector at the WS. Do not constrain rotational attitude about this pointing vector. (Use VECPOINT)	Rendezvous
1	Point a specified spacecraft vector at a celestial body. Do not constrain attitude about this vector. (Use VECPOINT)	Nonrendezvous
2	Rotate at a specified rate about a specified spacecraft vector.	Nonrendezvous
4	Same as Option 0, but constraining attitude about the pointing vector to a specified value (OMICRON).	Rendezvous
5	Same as Option 1, but constraining attitude about pointing vector to a specified value (OMICRON).	Nonrendezvous

5.2.5.1.1 Option 0: Orient the Spacecraft Such that a Specified Body-Fixed Pointing Vector Aligns with the Estimated Line of Sight to the WS. —Option 0 is effected by the Tracking Attitude Routine (R61), which is called by P20 as prescribed for this option in GSOP Section 4. The pointing vector (SCAXIS) is defined by two angles:

1. A specified positive angle (γ) from the spacecraft +X-axis about the spacecraft Z axis ($0 \leq \gamma < 360$ deg)
2. A specified positive angle (ρ) about the Y'-axis produced in the XY plane as a result of γ ($-90 \leq \rho \leq 90$ deg).



The LOS is determined by advancing the estimated CSM and WS state vectors to the current time. (See Kepler Subroutine, paragraph 5.5.5.) Once the coincidence of SCAXIS and LOS has been established, R61 establishes a spacecraft angular rate to maintain the coincidence. For Option 0, R61 provides the RCS DAP with the following quantities:

1. $\underline{\omega}_{CA}$ desired LOS rate plus LOS correction rate in control-axis coordinates
2. IGA_D, MGA_D, OGA_D DAP reference angles, i. e., center of the dead-band.
3. ΔGAa vector defining the desired incremental changes in the IMU gimbal angles every 0.1 second.

The desired LOS rate ($\underline{\omega}_{CA}$) is computed as follows:

$$\underline{u}_{LOS} = \underline{r}_W - \underline{r}_C$$

$$\underline{\omega}_{correction} = ([NBSM] \underline{SCAXIS} \times [REFSMMAT] \text{UNIT}(\underline{u}_{LOS})) / \Delta t_C$$

where [NBSM] uses the DAP reference angles THETADX, Y, and Z for the transformation from navigation-base to stable-member coordinates.

If $|\underline{\omega}_{\text{correction}}|$ exceeds MAXRATE, distribute as follows:

$$\underline{\omega}_{\text{correction}} = \text{MAXRATE} [\text{UNIT} (\underline{\omega}_{\text{correction}})]$$

$$\underline{\omega}_{\text{LOS}} = [\text{REFSMMAT}] \left\{ \frac{\text{UNIT} (\underline{u}_{\text{LOS}}) \times (\underline{v}_W - \underline{v}_C)}{|\underline{u}_{\text{LOS}}|} \right\} + \underline{\omega}_{\text{correction}}$$

$$\underline{\omega}_{\text{CA}} = [\text{NBCA}] [\text{SMNB}] \underline{\omega}_{\text{LOS}},$$

where $\underline{\omega}_{\text{LOS}}$ is the LOS rate in stable-member coordinates; [SMNB] and [NBSM] are transformation matrices defined in paragraph 5.6.3.2.1; [NBCA] is the matrix for transforming a vector from navigation-base to control-axis coordinates as defined below; \underline{r}_W , \underline{v}_W , \underline{r}_C , and \underline{v}_C are the position and velocity vectors of the WS and CSM in basic-reference coordinates obtained by using the Kepler Subroutine of paragraph 5.5.5; SCAXIS is the specified axis (γ, ρ); MAXRATE is the maximum correction rate; and Δt_C is a conservative estimate of the R61 computation interval (thus preventing overshoot in vehicle attitude by the time of the next pass).

$$[\text{NBCA}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 7.5 & -\sin 7.5 \\ 0 & \sin 7.5 & \cos 7.5 \end{bmatrix}$$

The vector $\underline{\Delta GA}$ is computed as follows:

$$\underline{\Delta GA} = \begin{bmatrix} \cos \text{IGA} \sec \text{MGA} & 0 & -\sin \text{IGA} \sec \text{MGA} \\ -\cos \text{IGA} \tan \text{MGA} & 1 & \sin \text{IGA} \tan \text{MGA} \\ \sin \text{IGA} & 0 & \cos \text{IGA} \end{bmatrix} (0.1) \underline{\omega}_{\text{LOS}}$$

where $\underline{\omega}_{\text{LOS}}$ is the LOS rate in stable-member coordinates computed previously.

5.2.5.1.2 Option 1: Orient the Spacecraft Such that SCAXIS Aligns with the LOS to a Specified Celestial Body. —Option 1 is effected by the Tracking Attitude Routine (R61), called by P20 as prescribed for this option in GSOP Section 4. SCAXIS is as defined in paragraph 5.2.5.1.1.

For Option 1, the LOS is determined by advancing the estimated CSM state vector to the current time. (See Kepler Subroutine, paragraph 5.5.5.) Once the coincidence of SCAXIS and LOS has been established, R61 establishes a spacecraft angular rate to maintain the coincidence. R61 quantities provided to the RCS DAP under tracking-option 1 are as described for Option 0 (paragraph 5.2.5.1.1), with the following exceptions:

$$\underline{u}_{\text{LOS}} = \text{line of sight direction, including aberration correction} \\ (\text{Refer to paragraph 5.5.13.})$$

and,

for STARCODE = 0-46,

$$\underline{\omega}_{\text{LOS}} = \underline{\omega}_{\text{correction}};$$

for STARCODE = 47,

$$\underline{\omega}_{\text{LOS}} = [\text{REFSMMAT}] \left\{ \frac{\text{UNIT } (\underline{r}_C) \times (\underline{v}_C)}{|\underline{r}_C|} \right\} + \underline{\omega}_{\text{correction}}$$

5.2.5.1.3 Option 2: Rotate the Spacecraft About SCAXIS.—Option 2 is effected by the Automatic Rotation Routine (R67), called by P20 as prescribed for this option in GSOP Section 4. SCAXIS is as defined in paragraph 5.2.5.1.1. R67 initializes COF, BRATE, [DEL], [MIS], and ADB in KALCMANU (GSOP Section 3) and then executes the KALCMANU coding to issue the following quantities to the RCS DAP:

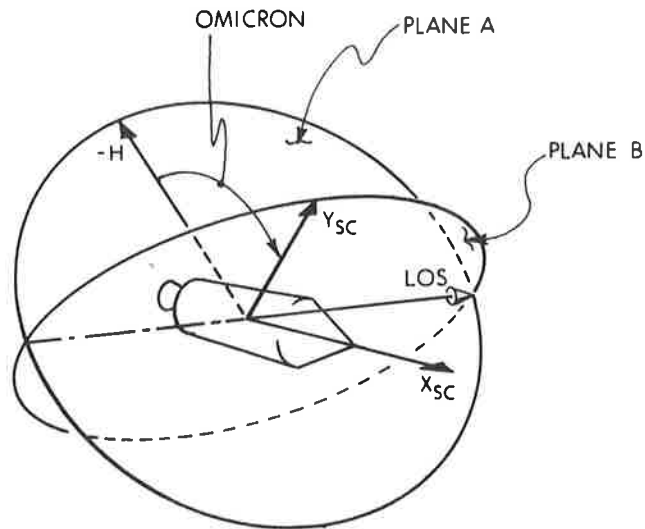
1. $\underline{\omega}_{\text{CA}}$ desired rotation rate
2. IGA_D , MGA_D , OGA_D , DAP reference angles,
i.e., center of dead-band.
3. $\Delta \underline{\text{GA}}$ a vector defining the desired incremental changes in the IMU gimbal angles every 0.1 second.

These calculations are repeated once per second.

5.2.5.1.4 Option 4: Orient the Spacecraft as in Option 0, but Further Specify an Angle (OMICRON) of Rotation About SCAXIS.—Option 4 is effected by the Tracking Attitude Routine (R61), called by P20 as prescribed for this option in GSOP Section 4.

SCAXIS is as defined in paragraph 5.2.5.1.1; OMICRON is the positive-sense angle between planes A and B—where plane A is defined by the negative momentum vector (-H) and the LOS, and plane B is defined by the spacecraft +Y-axis and the LOS.

Alignment of the LOS with either the spacecraft Y-axis or the momentum vector is proscribed since the necessary two-plane condition for specifying OMICRON would not exist.



For Option 4, R61 provides the following quantities to the RCS DAP:

1. $\underline{\omega}_{CA}$ desired LOS rate in control-axis coordinates
2. IGA_D , MGA_D , OGA_D , DAP reference angles,
i. e., center of the
deadband
3. $\underline{\Delta GA}$ a vector defining the desired incremental
changes in the IMU gimbal angles every
0.1 second.

The desired LOS rate ($\underline{\omega}_{CA}$) in control-axis coordinates is obtained as follows:

$$\underline{u}_{LOS} = \underline{r}_W - \underline{r}_C$$

$$\underline{\omega}_{LOS} = [REFSMMAT] \left\{ \frac{UNIT (\underline{u}_{LOS}) \times (\underline{v}_W - \underline{v}_C)}{|\underline{u}_{LOS}|} \right\}$$

$$\underline{\omega}_{CA} = [NBCA] [SMNB] \underline{\omega}_{LOS}$$

where $\underline{\omega}_{LOS}$ is the LOS rate in stable member coordinates; [SMNB] is a transformation matrix defined in paragraph 5.6.3.2.1; [NBCA] is the matrix for transforming a vector from navigation base to control axis coordinates as defined below; \underline{r}_W , \underline{v}_W , \underline{r}_C , and \underline{v}_C are the position and velocity vectors of the WS and CSM in basic reference coordinates obtained by using the Kepler Subroutine of paragraph 5.5.5.

$$[NBCA] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 7.25 & -\sin 7.25 \\ 0 & \sin 7.25 & \cos 7.25 \end{bmatrix}$$

The vector (ΔGA) is computed as follows:

$$\Delta GA = \begin{bmatrix} \cos IGA \sec MGA & 0 & -\sin IGA \sec MGA \\ -\cos IGA \tan MGA & 1 & \sin IGA \tan MGA \\ \sin IGA & 0 & \cos IGA \end{bmatrix} (0.1) \underline{w}_{LOS} + \underline{w}_{correction},$$

where

$$\underline{w}_{correction} = \begin{bmatrix} \theta_{DX} - OGA_D \\ \theta_{DY} - IGA_D \\ \theta_{DZ} - MGA_D \end{bmatrix} \frac{0.1}{\Delta t_C}$$

$\theta_{DX}, \theta_{DY}, \theta_{DZ}$ = desired gimbal angles

Δt_C = conservative estimate of R61 computation interval

The calculations used to compute the desired gimbal angles $\theta_{DX}, \theta_{DY}, \theta_{DZ}$ are similar to those used by the AXISGEN and CALCGA Routines, paragraphs 5.6.3.2.4 and 5.6.3.2.2, respectively. First, four unit vectors are computed as follows:

$$\underline{s}_A = \cos(\text{OMICRON}) \text{unit}(\underline{u}_{LOS} \times [\underline{v}_C \times \underline{r}_C]) + \sin(\text{OMICRON}) \text{unit}\left[\underline{u}_{LOS} \times (\underline{u}_{LOS} \times (\underline{v}_C \times \underline{r}_C))\right]$$

$$\underline{s}'_A = \text{unit}(\underline{SCAXIS} \times \underline{y}_{SC})$$

$$\underline{s}_B = \underline{u}_{LOS}$$

$$\underline{s}'_B = \underline{SCAXIS}$$

Given these unit vectors, the unit vectors \underline{x} , \underline{y} , \underline{z} are calculated defining the desired directions of the x, y, and z axes of the navigation base with respect to the Basic Reference Coordinate System. The unit vectors \underline{x} , \underline{y} , \underline{z} and the vectors \underline{x}_{SM} , \underline{y}_{SM} , \underline{z}_{SM} from the matrix REFSMMAT are then used to compute the desired gimbal angles θ_{DX} , θ_{DY} , θ_{DZ} .

5.2.5.1.5 Option 5: Orient the Spacecraft as in Option 1, but Further Specify an Angle (OMICRON) of Rotation About SCAXIS. —Option 5 is effected by the Tracking Attitude Routine (R61), called by P20 as prescribed for this option in GSOP Section 4.

SCAXIS is as defined in paragraph 5.2.5.1.1; OMICRON is as defined in paragraph 5.2.5.1.4 (Option 4).

For Option 5, R61 quantities provided to the RCS DAP are as in Option 4, except as follows:

\underline{u}_{LOS} = line of sight direction including aberration correction
and,

for STARCODE = 0-46,

$\underline{\omega}_{LOS} = 0;$

for STARCODE = 47,

$\underline{\omega}_{LOS} = [\text{REFSMMAT}] \left\{ \frac{\text{UNIT } (\underline{r}_C) \times (\underline{v}_C)}{|\underline{r}_C|} \right\}$

5.2.5.1.6 P20 General Flow. — For options 1 and 5, P20 causes R61 to be performed at intervals of approximately 3-4 seconds. For Option 2, P20 calls R67, which operates once per second. For all three of these options (1, 2, 5) if P20 is in the mode lights, the Coasting Integration Routine is called in a manner identical to that used by the CMC Idling Program.

For P20 options 0 and 4, the desired attitude and attitude rate are achieved by routine R61; then the Auto Optics Positioning Routine (R52) is called, followed by the Rendezvous Tracking Data Processing Routine (R22). (See Figure 2.5-1.) The purpose of the Auto Optics Positioning Routine (paragraph 5.6.8) is to drive the SXT to the line-of-sight of the WS or a star. To

ensure that the above line-of-sight is that of the WS, program P20 (options 0 or 4) sets the WS Target flag just before calling routine R52. If the WS Target flag is set and the optics mode switch is in the CMC position, routine R52 periodically computes the line-of-sight to the WS and drives the optics to it. The line-of-sight is also computed when the optics mode switch is not in the CMC position, and can be displayed upon astronaut request. In addition, routine R52 calls the Tracking Attitude Routine (R61) each time to re-establish the desired tracking attitude and attitude rate. The manner in which routine R61 is used is given in detail in Section 4 of the GSOP.

The purpose of the Rendezvous Tracking Data Processing Routine (R22) is to periodically process the optical and/or the VHF range-link tracking data to update the state vector of either the CSM or WS, as defined by the state vector update option. This option is automatically set to the CSM at the beginning of program P20 (options 0 or 4) but can later be set to the WS by the astronaut if it is considered necessary. The optical tracking data processed by this routine is obtained by using concurrently either the Rendezvous Tracking Sighting Mark Routine (R21) or the Backup Rendezvous Tracking Sighting Mark Routine (R23). If the astronaut wishes to have the VHF range-link data processed, he should set the VHF range flag (by keying V87E) after leaving Routine R00 and entering Program P20 (options 0 or 4). It is assumed that the astronaut has activated the VHF range-link and has assured himself of target acquisition by the range link before he sets the VHF range flag. The VHF range-link operates in conjunction with the voice link and measures the range between the CSM and the WS. Target acquisition can be ascertained by the astronaut by observing the range display provided with the VHF range link. The range data is sent to the CMC upon request by the Rendezvous Tracking Data Processing Routine (R22). In addition, R22 calls R27—the VHF Range Rate Mark Processing Routine (paragraph 5.6.6) to process VHF range data—if enabled by keying V76E.

As previously mentioned, the optical tracking data processed by routine R22 are obtained by using either the Rendezvous Tracking Sighting Mark Routine (R21) or the Backup Rendezvous Tracking Sighting Mark Routine (R23), both of which are described in detail in Section 4. Routine R21 is activated during the operation of P20 (options 0 or 4) by the astronaut's depressing of the MARK or MARK REJECT button. The sextant (SXT) is used to track the WS because of its greater accuracy and

target-detection capability. After the preferred tracking attitude has been established and the SXT has been driven to the LOS of the WS, the astronaut acquires the target through the SXT and switches the optics mode switch to the MANUAL position so that he can manually center the SXT reticle on the target. When accurate tracking is achieved, he presses the MARK button, which causes the measurement time and the SXT and IMU gimbal angles to be recorded. This data is stored in an erasable memory location denoted as Position 1.

In order to achieve the desired accuracy levels of the vehicle state vector with optical tracking data, it is important that this data be taken at essentially uniform intervals over the tracking period. Uniform tracking intervals are defined to be about once every minute throughout the phase, as opposed to all the tracking data being taken over a short period.

If it becomes necessary to use the Crew Optical Alignment Sight (COAS) instead of the SXT to obtain optical tracking data, use is made of the Backup Rendezvous Tracking Sighting Mark Routine (R23). At the beginning of this routine, the astronaut loads the coordinates of the COAS. These coordinates are the equivalent SXT shaft and trunnion angles of the COAS, which would be the same as those indicated for the SXT if it were possible to point the SXT in the same direction as the COAS. After the +X-axis tracking attitude has been established, the astronaut acquires the WS with the COAS and performs an optical mark by keying an ENTR into the DSKY (instead of depressing the MARK button, since he cannot reach the button when he is using the COAS). Whenever he keys in an ENTR, the measurement time, the IMU gimbal angles, and the previously loaded coordinates of the COAS are stored in the same Position 1 used by routine R21.

Optical tracking of the WS from the CSM and request for data from the VHF range-link are suspended during rendezvous maneuvers. The estimated CSM state vector is updated by means of the Average-G Routine during the maneuver, or by P77 - DSKY entry of the maneuver change in velocity. Mark processing is suspended in the rendezvous targeting programs when the Update flag is cleared, i. e., final computations have begun.

Functions Under Separate Control of Astronaut:

1. Rendezvous Tracking Sighting Mark Routine (R21)
2. Backup Rendezvous Tracking Sighting Mark Routine (R23)
3. Processing of VHF range data in routine R22
4. Enabling VHF Range Rate Mark Processing Routine (R27)

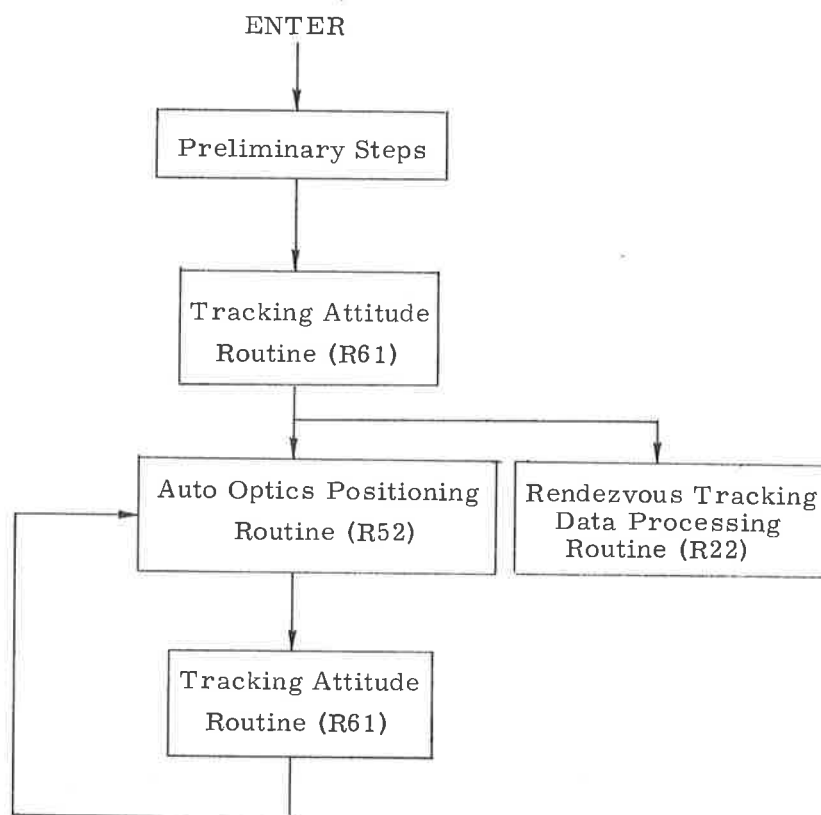


Figure 2.5-1. Rendezvous Navigation Program (P20 options 0 and 4)

NOTE. — Figure 2.5-2 has been deleted.

5. 2. 5. 2 Rendezvous Tracking Data Processing Routine (R22)

In the P20 Rendezvous Mode (Options 0 and 4), Routine R22 processes optics and VHF tracking data to update the state vector of either the CSM or WS (See GSOP Section 4.)

5. 2. 5. 2. 1 Rendezvous Data-Processing Logic. — Figure 2. 5-3 is a flowchart of R22 logic. Observe that the routine alternates between checking the status of the VHF Range flag and checking to see if optics mark data are in Position 1. Preceding these checks in both branches is a check on the R27 flag. If this flag is set, R27 is performed as described in paragraph 5.6.6. Following R27 is a check on SNAP-FLAG. This flag is set in R27 in order to prevent the beginning of a state-vector update within 20 seconds before the recording of the CDU angles for use in the optimization mode of R27, and reset as soon as this time is past. Proceeding down both branches, observe that whenever optics mark data are found in Position 1, they are transferred to Position 2 and used to calculate the correction (or update) to the state vector; and whenever the VHF Range flag is set, the routine calls R08 to read the range from the VHF range-link if at least 60 seconds have expired since the last time range was read. Immediately after reading the range, a check is made to see if the Data Good discrete is being received from the VHF range-link, signifying that the range tracking network is tracking the target satisfactorily. If the Data Good discrete is present, the range data are used to calculate the correction to the state vector. The time t_{VHF} is used as the time of range measurement. If the Data Good discrete is not present, the Tracker Fail Light is turned on.

The manner in which the state vector correction is calculated for either optics mark or range-link data is given in paragraph 5. 2. 5. 2. 2. To distinguish between the two types of data in Figure 2. 5-3 use is made of a Source Code (SC), which is equal to one or two depending on whether it is optics mark or range-link data, respectively.

After the state vector correction has been calculated, it is seen in Figure 2. 5-3 that a check is made to determine if the magnitudes of the proposed corrections in position and velocity (δr and δv) exceed certain threshold limits (δr_{MAX} and δv_{MAX}) stored in erasable memory. The same threshold limits are used in this check whether the state vector correction is based on optics mark or range-link

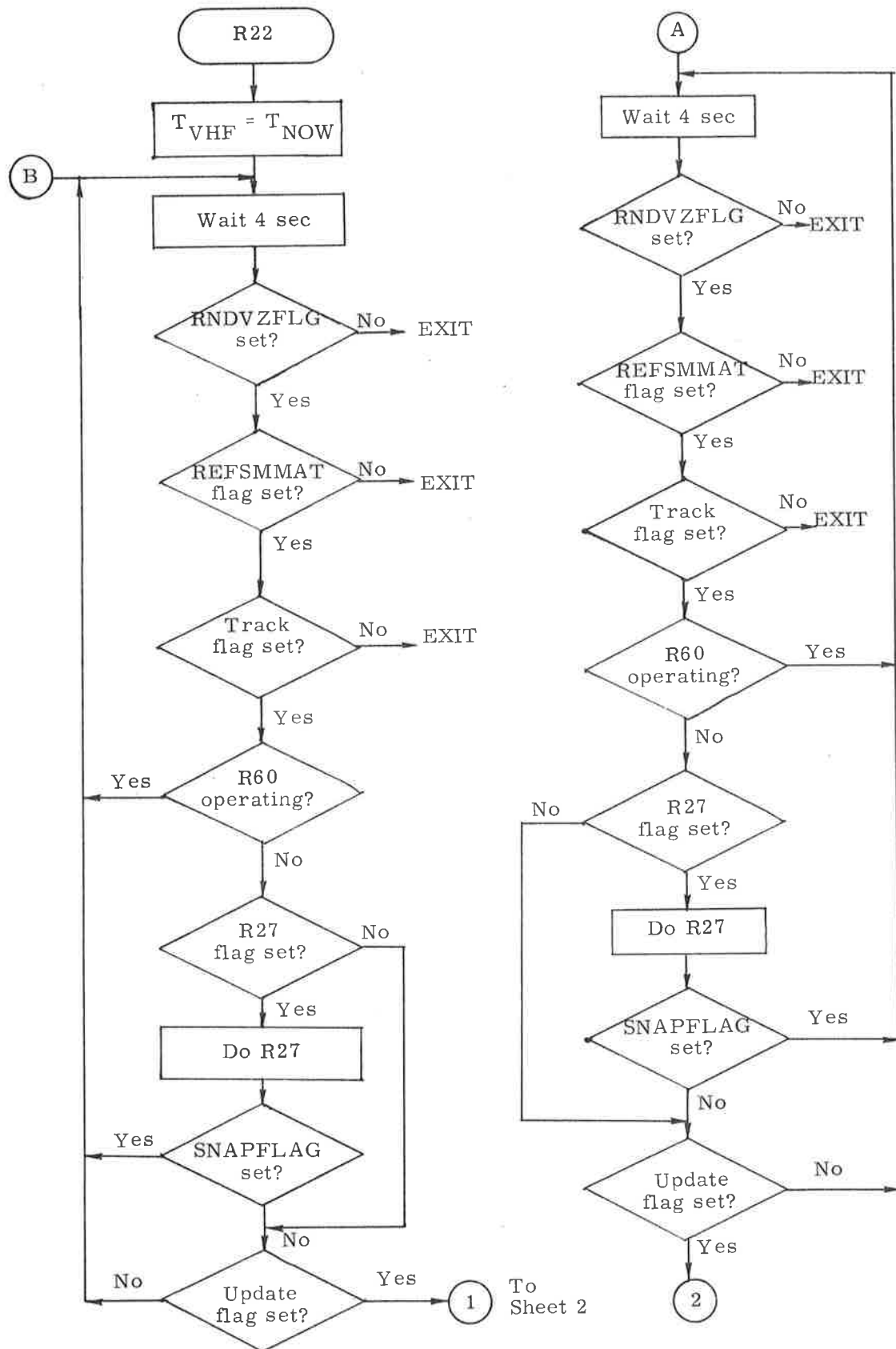


Figure 2.5-3. Rendezvous Tracking Data Processing Routine (Sheet 1 of 3)

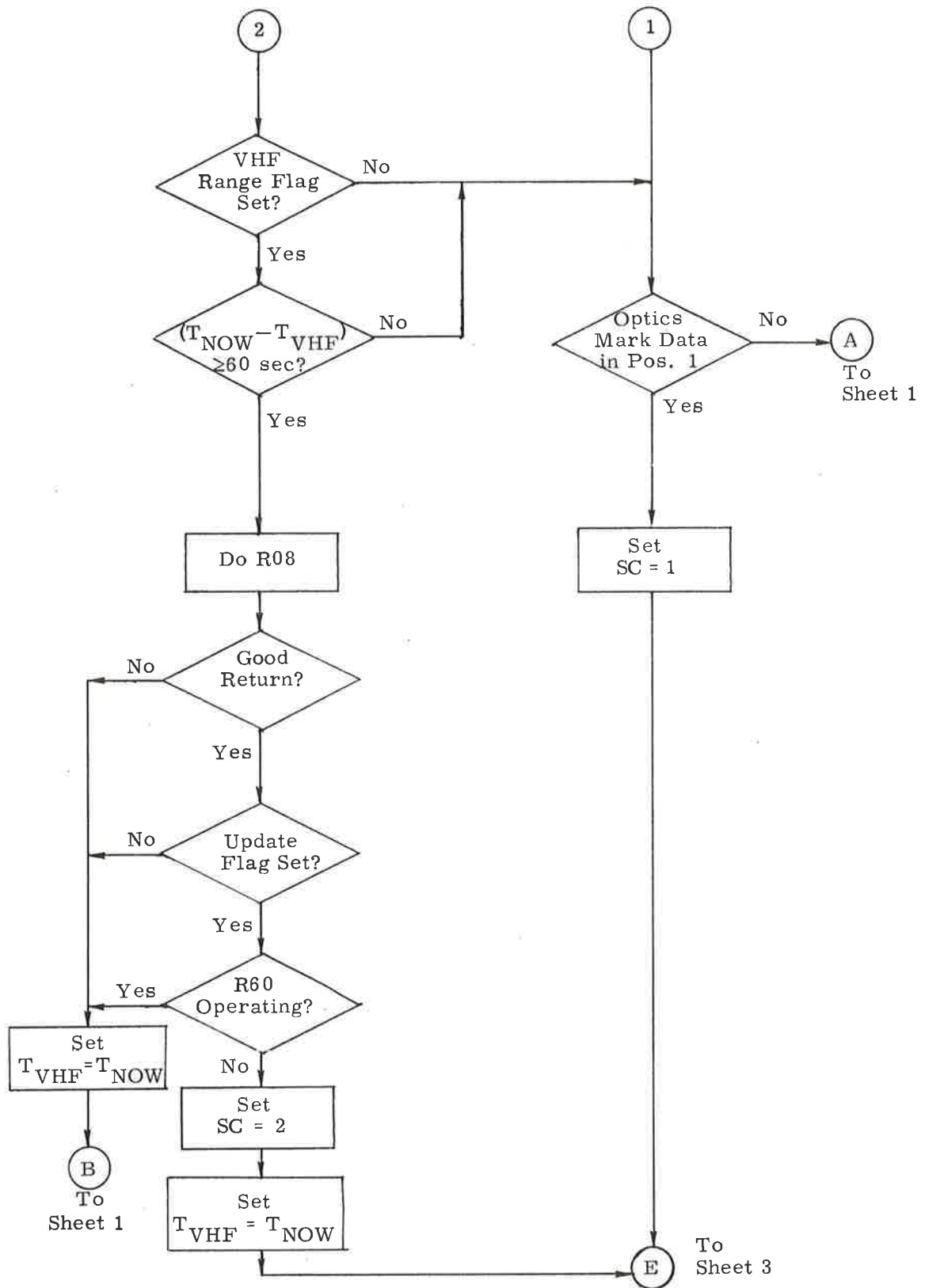


Figure 2.5-3. Rendezvous Tracking Data Processing Routine (Sheet 2 of 3)

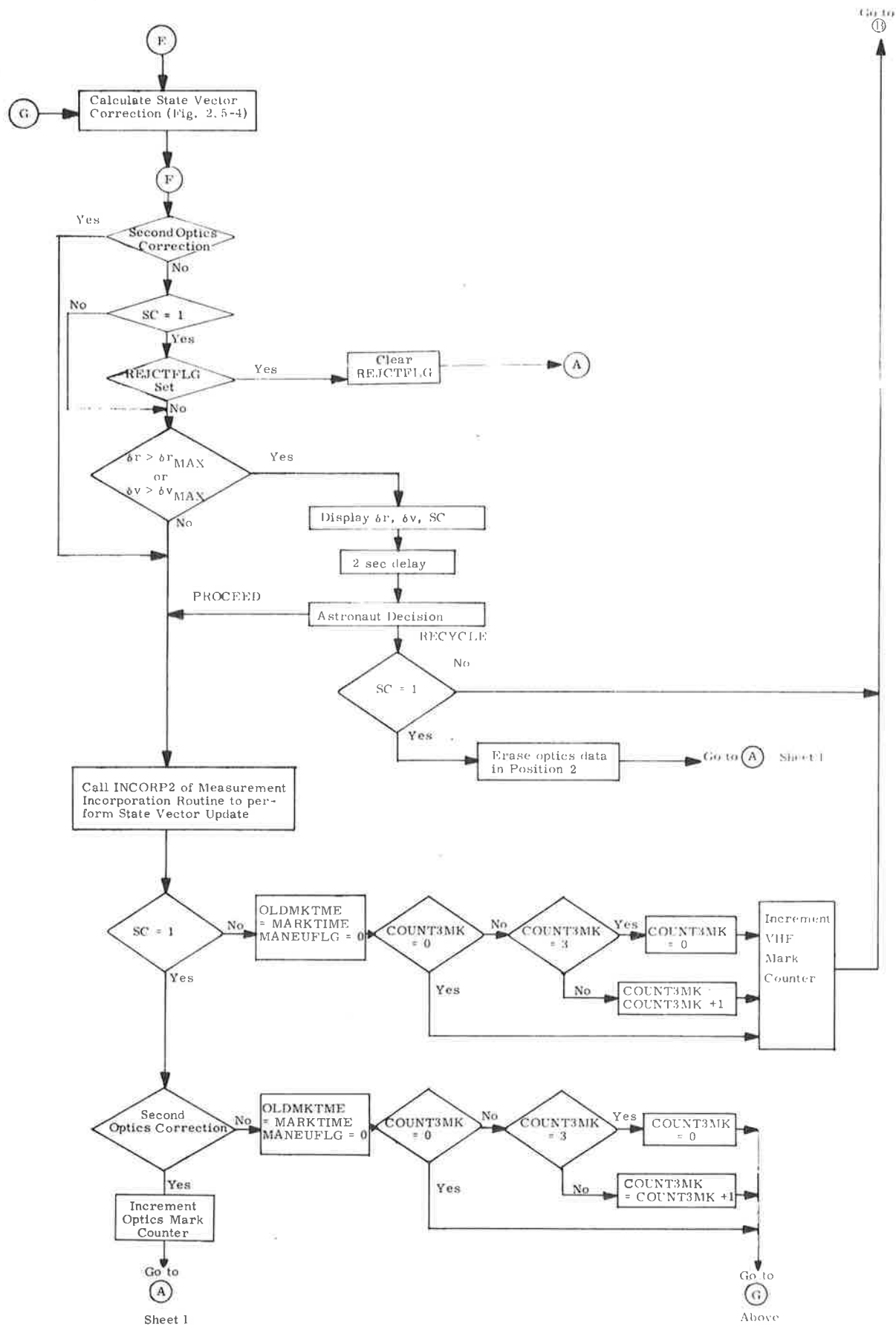


Figure 2.5-3. Rendezvous Tracking Data Processing Routine (Sheet 3 of 3)

data. The purpose of this threshold check is to ensure the validity of the proposed state-vector correction (update). If the proposed correction exceeds either threshold limit, the magnitudes of the correction in position and velocity (δr and δv) and the Source Code (SC) are displayed to the astronaut. If the correction is the result of optical tracking data and the astronaut is sure that he is tracking the WS, he should command the update. Otherwise, he should reject the data and recheck the optical tracking. Additional details on the threshold check are given in paragraph 5.2.1. It should be noted that if optical data are being processed, there will actually be two separate corrections to the state vector, instead of one, due to the manner in which the optics data are used to calculate the state vector correction, as explained in paragraph 5.2.5.2.2. Since the second optics correction is calculated using the state vector updated with the first optics correction, it is unlikely that the magnitude of the second correction will exceed that of the first. Consequently, a threshold check is made in Figure 2.5-3 only on the first optics correction.

At various points in Figure 2.5-3 it is seen that a check is made to determine if the Update flag is set. This flag is cleared when there is no desire to process the optics and range-link data. It is cleared by the CMC during CSM Δv maneuvers, during certain time-consuming computations, during the final computations of the rendezvous targeting programs, and during the CSM Velocity Vector Update Program. If the Update flag is cleared, a check is made on the Track flag. The Track flag is cleared when it is desired to temporarily terminate the rendezvous navigation process. If the Track flag is set, it is seen in Figure 2.5-3 that the routine continues to monitor the Track and Update flags in a standby status until one of the flags changes state.

The range obtained from the VHF range-link by the Rendezvous Tracking Data Processing Routine is that measured by the range-link between the CSM and the WS. These data are sent to the CMC from the range-link as a 15-bit binary data word R_{RL} . In the CMC the range R_m in nautical miles is obtained as follows:

$$R_m = k_{RL} R_{RL}$$

where k_{RL} is the bit weight in nautical miles.

5.2.5.2.2 Rendezvous Navigation Computations. — Each set of optical navigation data contains the time of the measurement and the two optics and three IMU gimbal angles. From these five angles the measured unit vector \underline{u}_M along the CSM-to-WS line-of-sight is computed in the Basic Reference Coordinate System from

$$\underline{u}_M = [\text{REFSMMAT}]^T [\text{NBSM}] \underline{u}_{NB} \quad (2.5.1)$$

where $[\text{REFSMMAT}]$ and $[\text{NBSM}]$ are transformation matrices and \underline{u}_{NB} is the measured line-of-sight vector in navigation base coordinates. All terms of Equation (2.5.1) are defined in paragraph 5.6.3.

For the purpose of navigation, it is convenient to consider the measured unit vector \underline{u}_M to be the basic navigation data. This navigation measurement of the line-of-sight vector \underline{u}_M is mathematically equivalent to the simultaneous measurement of the angles between the lines-of-sight to the WS and two stars. The data are processed by selecting two convenient unit vectors (fictitious star directions), converting the vector \underline{u}_M to an equivalent set of two artificial star-WS measurements, and using the Measurement Incorporation Routine (paragraph 5.2.3) twice, once for each artificial measurement. These two unit vectors are chosen to be perpendicular to each other and to the current estimated line-of-sight vector so as to maximize the convenience and accuracy of the procedure.

Let \underline{r}_C and \underline{r}_W be the estimated CSM and WS position vectors at the time of a given line-of-sight measurement. Then, the first state vector update for the measurement is performed as follows:

- ① Calculate the estimated CSM-to-WS line-of-sight from

$$\underline{r}_{CW} = \underline{r}_W - \underline{r}_C \quad (2.5.2)$$

$$\underline{u}_{CW} = \text{UNIT}(\underline{r}_{CW})$$

- ② Initialize the fictitious star direction to the vector

$$\underline{u}_s = \text{UNIT} (\underline{u}_{CW} \times \underline{u}_M) \quad (2.5.3)$$

If the vectors \underline{u}_{CW} and \underline{u}_M are separated by a small enough angle (possibly as much as $\sqrt{3} \times 2^{-19}$ rad), then a computation overflow occurs in the execution of Equation (2.5.3), and this set of measurement data is discarded because all the components of the estimated state-vector deviation, $\delta \underline{x}$, would be negligible for both state vector updates.

- ③ Compute an artificial star direction from

$$\underline{u}_s = \text{UNIT} (\underline{u}_s \times \underline{u}_{CW}) \quad (2.5.4)$$

- ④ Calculate the six-dimensional geometry vector, \underline{b} , from

$$\underline{b}_0 = \pm \frac{1}{r_{CW}} \underline{u}_s \quad (2.5.5)$$

$$\underline{b}_1 = \underline{0} \quad (2.5.6)$$

where the + (-) sign is selected if the CSM (WS) state vector is being updated, and $r_{CW} = |\underline{r}_{CW}|$

- ⑤ Determine the measured deviation δQ from

$$\begin{aligned} \delta Q &= \cos^{-1} (\underline{u}_s \cdot \underline{u}_M) - \cos^{-1} (\underline{u}_s \cdot \underline{u}_{CW}) \quad (2.5.7) \\ &= \cos^{-1} (\underline{u}_s \cdot \underline{u}_M) - \frac{\pi}{2} \end{aligned}$$

- ⑥ Incorporate the fictitious star-WS measurement using the Measurement Incorporation Routine (paragraph 5.2.3).

Included in Step ⑥ is the state vector update validity check for the first proposed update, as described in paragraph 5.2.1.

It should be noted that the initialization of the star direction \underline{u}_s , which is given by Equation (2.5.3), is such that the first artificial star [computed from Equation (2.5.4)] will in most cases yield the maximum value for the measured deviation δQ that is obtained from Equation (2.5.7).

Assuming that the first state vector update was valid, the second update for this measurement data set is performed by first recomputing the estimated CSM-to-WS line-of-sight vector from Equation (2.5.2) using the updated values of the estimated CSM and WS position vectors \underline{r}_C and \underline{r}_W , respectively. Then, Steps ③ - ⑥ are repeated, this time with no state vector update validity check.

If the first proposed state vector update does not pass the validity check, then the magnitudes of the proposed changes in the estimated position and velocity vectors δr and δv , respectively, are displayed. If the astronaut is sure that he is tracking the WS, then he should command the update. Otherwise, he should reject the data and recheck the optical tracking. A detailed discussion of this state vector update validity check is given in paragraph 5.2.1.

The results of the processing of the measured line-of-sight vector \underline{u}_M are updated values of the estimated position and velocity vectors of the CSM or the WS. These two estimated state vectors are used to compute required rendezvous targeting parameters as described in subsection 5.4.

For convenience of calculation in the CMC, Equations (2.5.4) - (2.5.7) are reformulated and regrouped as follows:

$$\begin{aligned}
 \underline{u}_s &= \text{UNIT} (\underline{u}_s \times \underline{u}_{CW}) \\
 \underline{b}_0 &= \underline{u}_s \\
 \underline{b}_1 &= \underline{0} \\
 \delta Q &= r_{CW} \left[\cos^{-1} (\underline{u}_s \cdot \underline{u}_M) - \frac{\pi}{2} \right]
 \end{aligned}
 \tag{2.5.8}$$

This set of equations is used by the Rendezvous Navigation Routine in processing optical tracking data. To validate the use of Equations (2.5.8) let

$$\overline{\alpha^2} = r_{CW}^2 (\text{var}_{SXT} + \text{var}_{IMU}) + \text{var}_{INT}
 \tag{2.5.9}$$

where var_{SXT} and var_{IMU} are the a priori estimates for the SXT and IMU angular error variances per axis, respectively. The variable var_{INT} is included in Equation (2.5.9) for the purpose of smoothing the effects of coasting integration inaccuracies.

After VHF range-link acquisition is established (paragraph 5.2.5.1), the measured CSM-to-WS range (R_m) is automatically acquired at approximately one-minute intervals. The geometry vector and measured deviation for a VHF range measurement are given by

$$\begin{aligned} \underline{b}_0 &= \pm \underline{u}_{\text{CW}} \\ \underline{b}_1 &= \underline{0} \\ \delta Q &= R_m - r_{\text{CW}} \end{aligned} \tag{2.5.10}$$

where the + (-) sign is selected if the WS (CSM) state vector is being updated. The measurement error variance is computed from

$$\overline{\alpha^2} = \text{maximum} (r_{\text{CW}}^2 \text{var}_R + \text{var}_{\text{INT}}, \text{var}_{R_{\text{min}}}) \tag{2.5.11}$$

where var_R is the range error variance corresponding to a percentage error and $\text{var}_{R_{\text{min}}}$ is the minimum range error variance.

The rendezvous navigation computations are illustrated in Figure 2.5-4. As shown in the figure, this set of computations is entered from two points (E) and (G) of Figure 2.5-3. It is assumed that the following items are stored in erasable memory at the start of the computation shown in the figure:

- \underline{x}_C = Estimated CSM state vector as defined in paragraph 5.2.2.6.
- \underline{x}_W = Estimated WS state vector
- W = Six-dimensional error transition matrix associated with \underline{x}_C or \underline{x}_W as defined in paragraph 5.2.2.4.
- WRDTIME = Specified minimum time between W-matrix reinitializations.

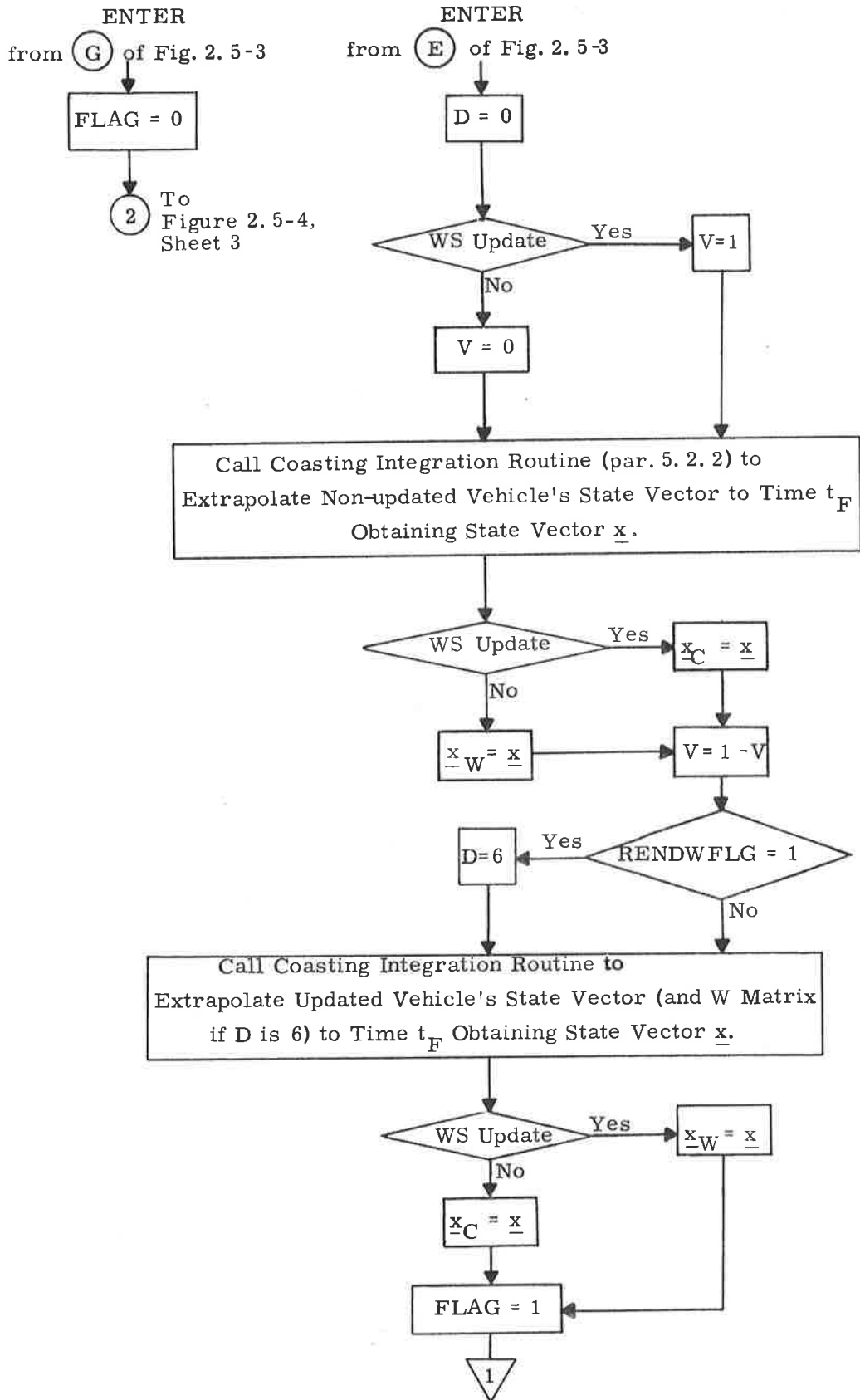


Figure 2.5-4. Rendezvous Navigation Computations (Sheet 1 of 4)

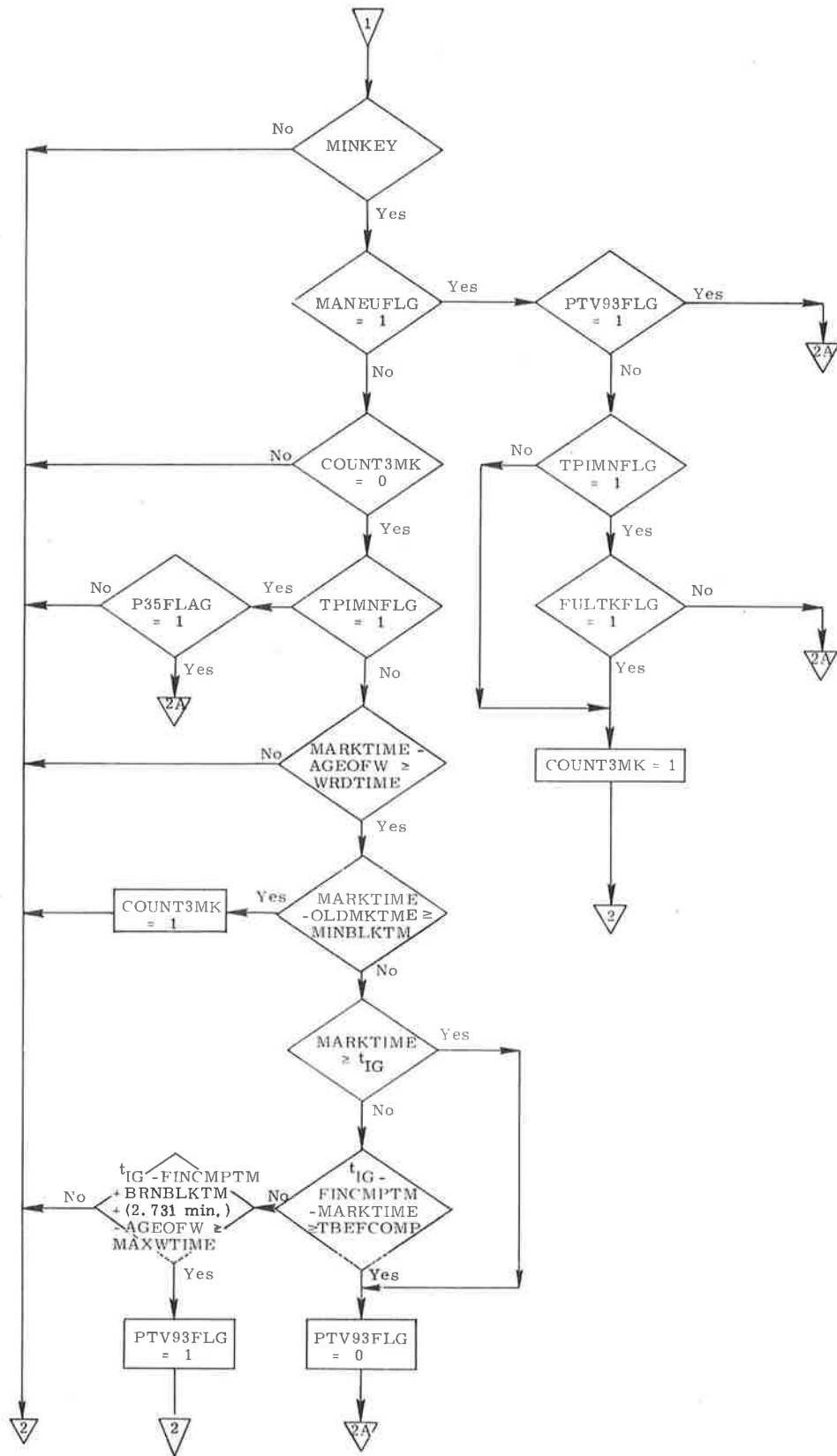


Figure 2.5-4. Rendezvous Navigation Computations (Sheet 2 of 4)

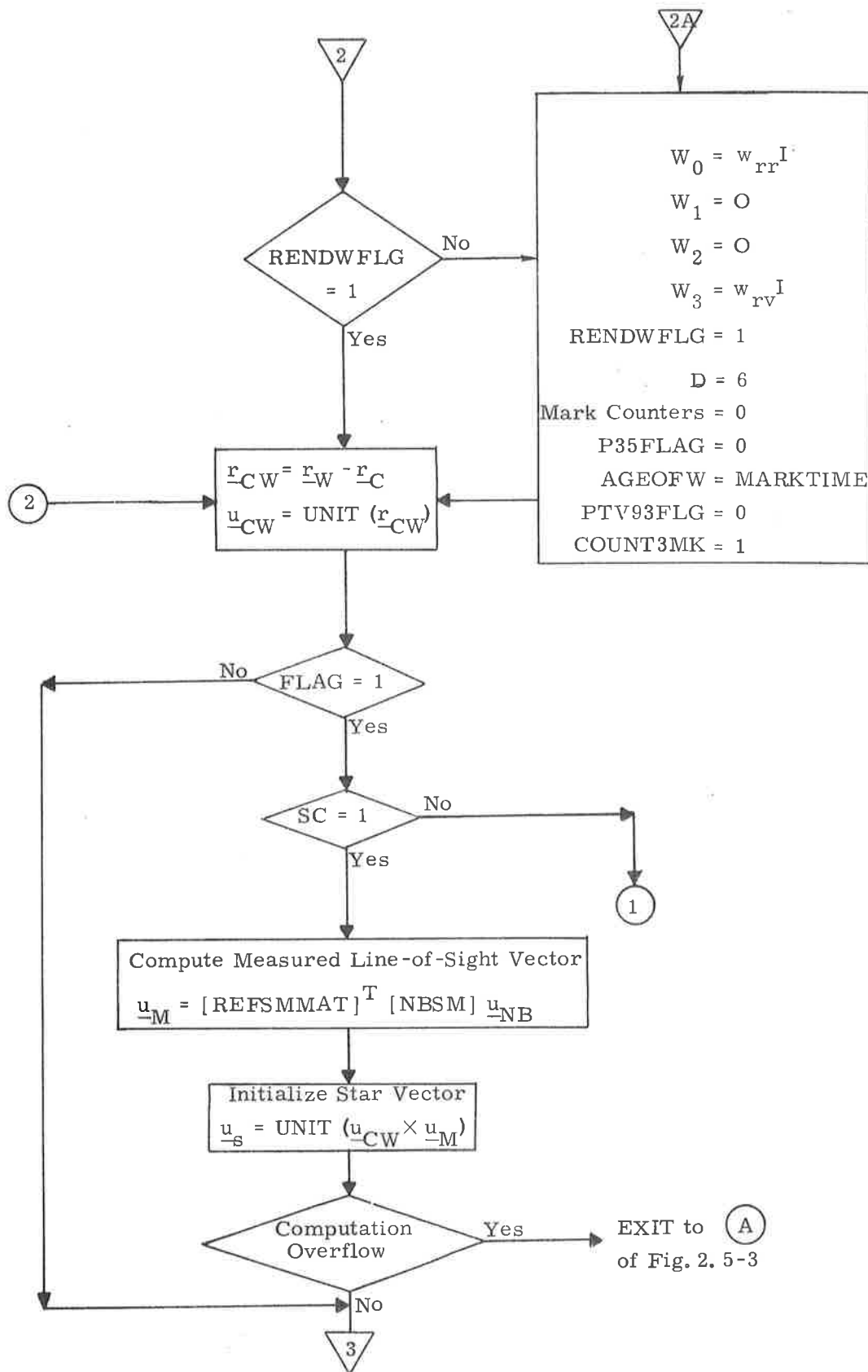


Figure 2.5-4 Rendezvous Navigation Computations (Sheet 3 of 4)

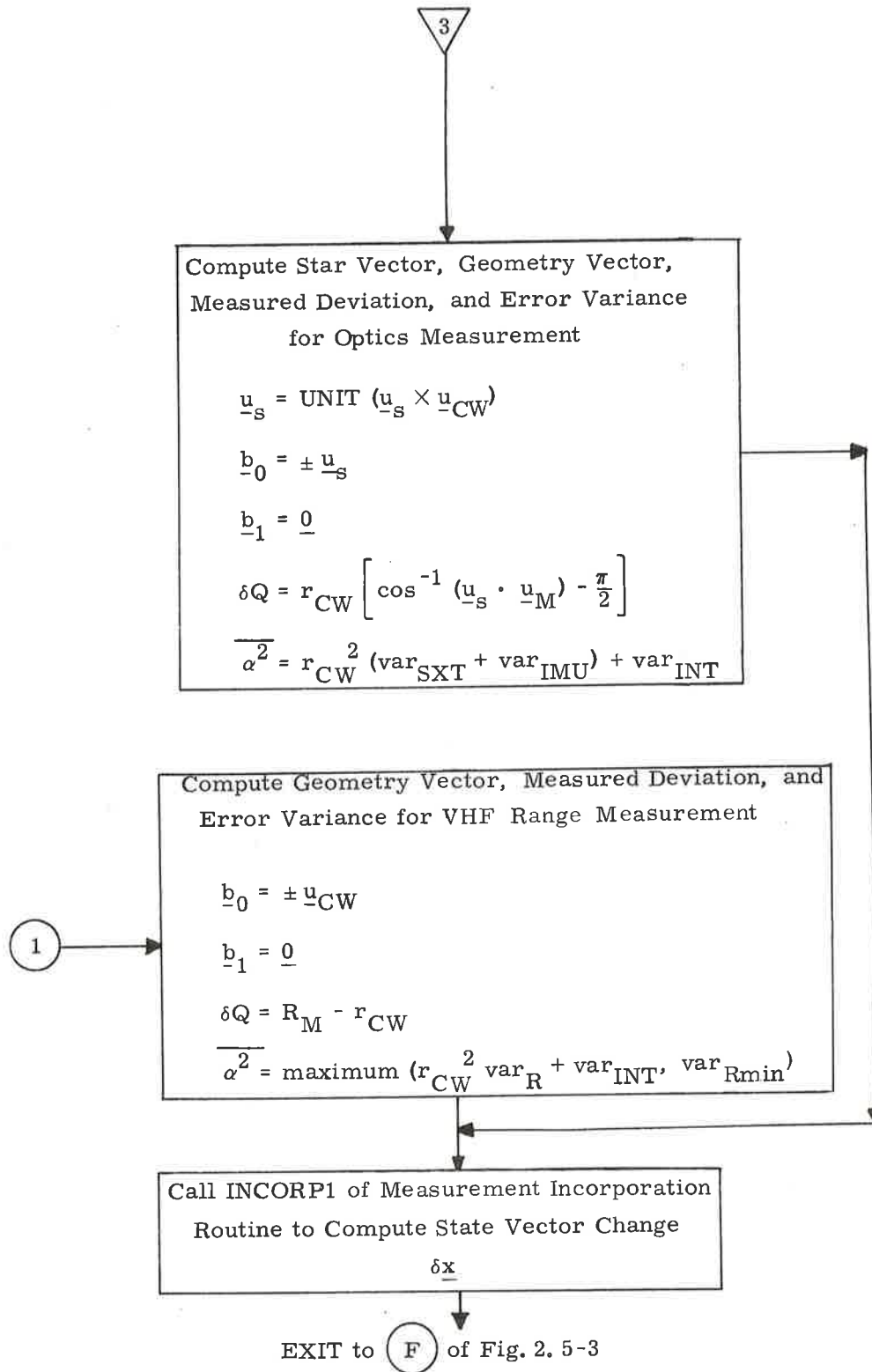


Figure 2. 5-4 Rendezvous Navigation Computations (Sheet 4 of 4)

- MINBLKTM = Specified maximum time after a mark that a W-matrix reinitialization can be performed.
- TBEFCOMP = Specified minimum time before final targeting solution that W-matrix reinitializations will be allowed.
- BRNBLKTM = Estimated total time without marking before and after a maneuver.
- MAXWTIME = Specified maximum time between W-matrix reinitializations.
- FINCMPTM = Specified maximum time before a maneuver that a mark can be taken.

The following definitions apply:

RENDWFLG = $\begin{cases} 1 & \text{for valid W-matrix} \\ 0 & \text{for invalid W-matrix} \end{cases}$

This flag or switch is maintained by programs external to the Rendezvous Navigation Routine. It indicates whether or not the existing W-matrix is valid for use in processing WS tracking data. The flag is set to zero after each of the following procedures:

1. State vector update from ground
2. Astronaut command
3. W-matrix overflow.

[REFSMMAT] = Transformation Matrix: Basic Reference Coordinate System to IMU Stable Member Coordinate System.

t_F = Measurement time.

Five optics and IMU
gimbal angles (or)
 R_m } = Measured range.

w_{rr} and w_{rv} = Preselected W-matrix initial diagonal elements. There is one value for each of these two initial diagonal W-matrix elements stored in the CMC erasable memory. These parameters nominally represent rendezvous initialization state errors. They can be changed during the mission by the astronaut or by RTCC. During MINKEY, these parameters are changed automatically at the completion of each targeting program.

SC = Source Code $\left\{ \begin{array}{l} 1 \text{ Optics Measurement} \\ 2 \text{ Range Measurement} \end{array} \right.$

MANEUFLG (maneuver has been performed)—set on final pass through any rendezvous targeting program except P38

TPIMNFLAG (TPI has been performed)—set on final pass through P35 and on entry to P36

P35FLAG (MCC1 has been performed)—set on final pass through P36

MARKTIME —time of current mark

AGEOFW—time of last W-matrix reinitialization

OLDMKTME—time of last mark

PTV93FLAG—causes W-matrix to be reinitialized on first mark after a maneuver

FULTKFLAG—(single-sensor tracking denoted)—set by crew.

t_{IG} — time of ignition (During P38, this is ignition time of NSR maneuver, not of plane change.)

COUNT3MK—mark counter, counts up to three marks

The variables D and V are indicators that control the Coasting Integration Routine (paragraph 5.2.2) as described in paragraph 5.2.2.6, and I and O are the three-dimensional identity and zero matrices, respectively.

The optical measurement incorporation procedure outlined above should be repeated at about one-minute intervals throughout the rendezvous phase except during final targeting computations and powered maneuvers, as described in paragraph 5.2.5.1. To achieve desired state-vector accuracy, it is important that SXT tracking data be taken over the largest possible angular sector in inertial space (as swept out by the SXT line-of-sight). This is required since the SXT tracking provides information only in directions normal to the line-of-sight [as indicated by Equation (2.5.5)], and, thus, the line-of-sight must be allowed to rotate in inertial space to achieve more complete update data. As mentioned previously, the VHF range data are obtained automatically, approximately every minute.

5.2.5.2.3 Rendezvous Navigation Computations (Alternate Line-of-sight). —During rendezvous, optical tracking data are normally obtained by means of SXT sightings of the WS from the CSM and processed as described in paragraph 5.2.5.2.2. Navigation data can also be obtained by means of a backup optical device. Since data obtained in this manner are much less accurate than SXT sighting data, the backup device should be used for the sightings only if the SXT has failed, or if the astronaut cannot return to the lower equipment bay to make SXT sightings.

The processing of the data from a backup sighting is identical to the procedure described in paragraph 5.2.5.2.2 with the following two exceptions:

1. The values of the shaft and trunnion angles associated with the particular device used are astronaut input items.
2. The measurement error variance, Equation (2.5.9), is replaced by

$$\overline{\alpha^2} = r_{CW}^2 (\text{var}_{ALT} + \text{var}_{IMU}) + \text{var}_{INT} \quad (2.5.12)$$

where var_{ALT} is the a priori estimate for the angular error variance of an alternate line-of-sight measurement per axis.

5.2.5.3 Automatic W-matrix Reinitialization (WRI)

The error-transition (W) matrix is defined in paragraph 5.2.2.4. In the P20 Rendezvous Mode, a capability has been provided for a minimum-keystroke-sequenced (MINKEY) rendezvous. (See GSOP Section 4.) A salient feature of MINKEY is automatic W-matrix reinitialization (WRI). The considerations underlying the automatic W-matrix reinitialization logic in Figure 2.5-4 are (1) to avoid premature reinitializations, which would degrade the error correction and correlation, (2) to avoid late reinitializations, which would allow the W-matrix to become superannuated, i. e., to shrink to the point that incoming measurements are downweighted excessively or to degrade in accuracy to the point that the matrix is no longer reliable.

Loss of filter gain is a function of the measurement geometry and the number of marks taken since the last WRI and is the result of incomplete error modeling; loss of reliability is a function of time since the last WRI and is the result of incorrect error modeling due to computer storage limitations. Since geometry is a factor in the loss of gain, determining when the loss will occur requires a bit-by-bit analysis of simulated computer operations for a particular mission. Determining how long the W-matrix can be extrapolated before it becomes unreliable is more difficult, since a major factor is the accuracy of the state vector which is not being updated. A generalized scheme that can be mechanized for all contingencies is possible, however, by following a minmax concept of reinitializing more often than may be necessary but less often than would significantly degrade performance.

Accordingly, the principal criterion for allowing a WRI after other than a first mark following a maneuver is whether a specified minimum time (WRD TIME) has elapsed since the last WRI. Typically, WRD TIME is pad-loaded with a value of about 40 minutes—long enough for effective correlation and smoothing, short enough to occur well before superannuation. The second criterion is whether no more than a specified maximum time (MINBLK TM, typically 5 minutes) has elapsed since the last mark. If more than MINBLK TM has elapsed since the last mark (VHF or optics), WRI will be inhibited for three marks in order to allow W-matrix correlation to reduce error buildup during the no-mark period. The third criterion for allowing WRI is whether more than a specified minimum time (TB EFCOMP) remains before the final targeting computation. For example, TB EFCOMP will have a preset value of about 10–15 minutes, allowing a sufficient period of post-WRI tracking to provide a correlated W-matrix and a best state-vector estimate for the targeting com-

putation. If the other criteria have been satisfied and more than TBEFCOMP remains before final targeting, WRI is allowed and the mark is incorporated.

Before transfer-phase initiation (TPI), the nominal flow for a first mark following a maneuver is to inhibit WRI until three marks have been accumulated for correlation. The exception to this is when, as a result of the TBEFCOMP restriction preceding a maneuver, the W-matrix age would exceed MAXWTIME (~1 hour) should WRI be inhibited. For this contingency, WRI is specified to occur before the first mark is incorporated.

The other condition when WRI occurs before the first mark is incorporated is in the nominal-flow situation following TPI. The post-TPI geometry is such that filter gain deteriorates more rapidly than in the pre-TPI phase. Consequently, with both VHF and SXT operating, the nominal post-TPI procedure gives higher priority to restoring filter gain than to correlation: WRI occurs on the first mark, and is prohibited on subsequent marks.

Correlation in the unmeasured dimension becomes obligatory, however, when data are being incorporated from only one sensor. The post-TPI, single-sensor strategy is to reinitialize the W-matrix only once between TPI and the second midcourse correction (MCC2). The strategy is mechanized as follows. The first mark following TPI is incorporated without WRI. Since subsequent marks between TPI and the first midcourse correction (MCC1) will not pass the TBEFCOMP criterion, no WRI can occur during the TPI-MCC1 phase. Following MCC1, however, a WRI is specified to occur after the first three marks have been incorporated—regardless of TBEFCOMP. The single WRI, occurring three marks after MCC1, ensures the best post-TPI trade-off between the requirements of one-sensor correlation and filter gain.

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 cutoff conditions are achieved. The powered flight navigation program used to maintain an estimate of the vehicle state vector during all thrusting conditions is referred to as the Average-G Routine and is presented in paragraph 5.3.2.

The basic powered flight guidance concept used in the CMC is a velocity-to-be-gained concept with cross product steering (paragraph 5.3.3.4), which is used in each of the following two procedures:

1. Lambert Aimpoint Maneuver Guidance (paragraph 5.3.3.2).
2. External Δv Maneuver Guidance (paragraph 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.

5.3.2 Powered Flight Navigation - Average-G Routine

The purpose of this powered flight navigation subroutine 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 effect 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 (Δt). The computations are, therefore, in terms of discrete increments of velocity rather than instantaneous accelerations. The repetitive computation cycle time Δt 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, for example, during powered flight. The Average-G Routine computations are illustrated in Figures 3.2-1 and 3.2-2. The following are definitions of the parameters used in these figures:

$\underline{r}(t)$	Vehicle position vector at time t .
$\underline{v}(t)$	Vehicle velocity vector at time t .
Δt	Computation cycle of 2 seconds.
$\underline{\Delta v}(\Delta t)$	The velocity vector change sensed by the IMU PIPA's over the time interval Δt . 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 of earth's true polar axis in the Basic Reference Coordinate System.
μ_E	Earth gravitational constant.
r_E	Equatorial radius of the earth.
J	Gravity harmonic coefficient.
$\underline{g}_b(t)$	Component of the earth gravity acceleration vector representing earth oblateness effects.

In Figure 3.2-2, it can be seen that only a single oblateness term is included in the earth gravity subroutine computation.

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

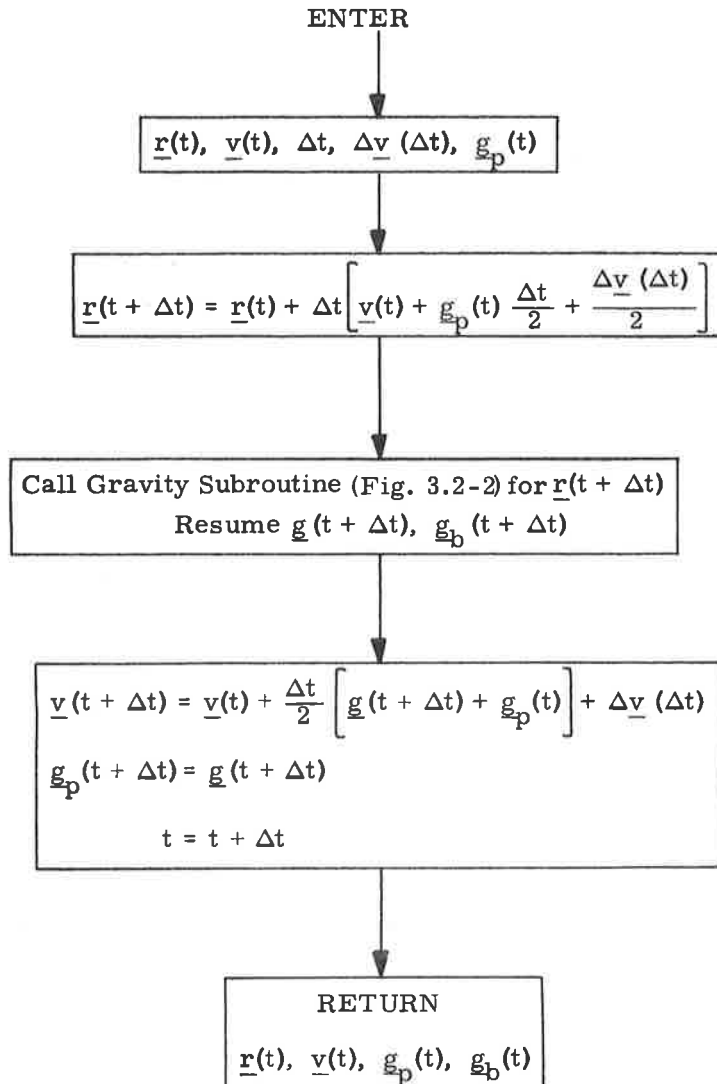


Figure 3.2-1. Average-G Subroutine

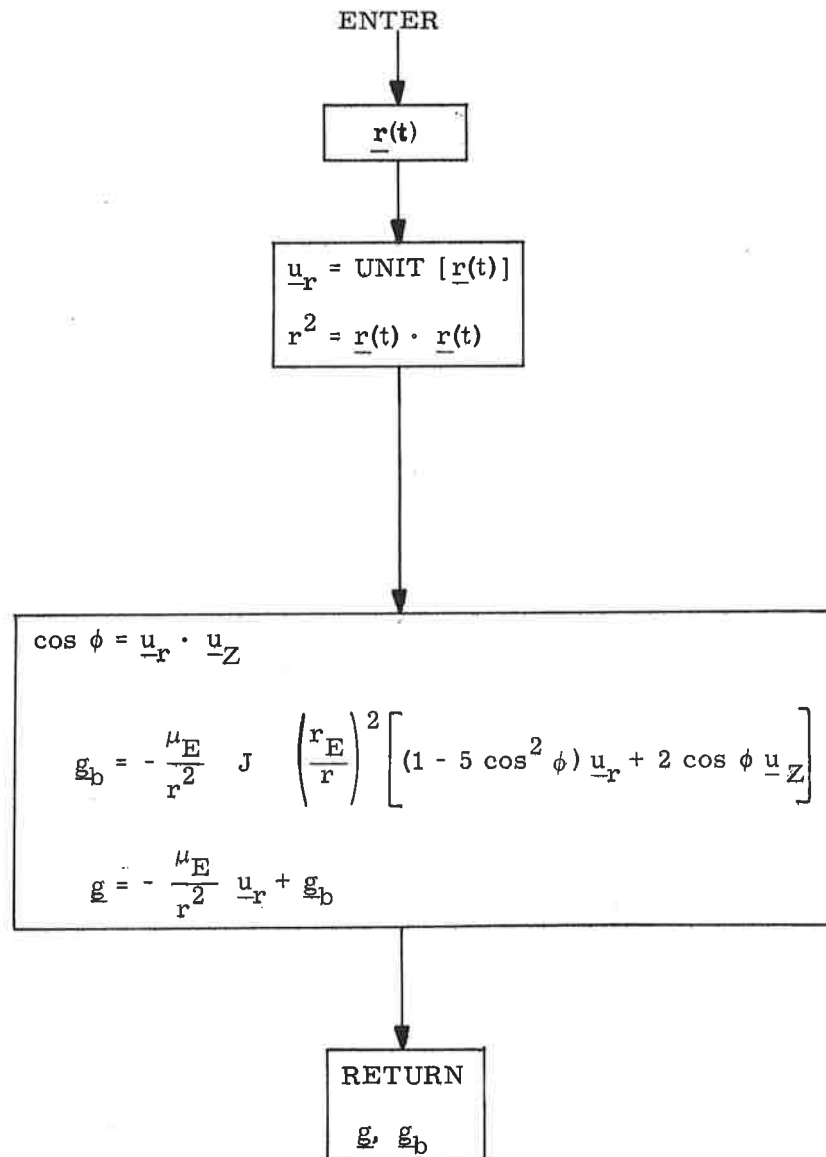


Figure 3.2-2. Gravity Subroutine

5.3.3 Powered Flight Guidance Using Cross Product Steering

5.3.3.1 Introduction

The cross product steering concept can be used to control the following maneuvers that are targeted by the premaneuver programs as listed.

1. Corrective Maneuver 1 (NC1) P31
2. Corrective Maneuver 2 (NC2) P32
3. Corrective Combination Maneuver (NCC) P33
4. Coelliptic Maneuver (Slow Rate) (NSR) P34
5. Terminal Phase Initiation (TPI) P35
6. Terminal Phase Midcourse (TPM) P36
7. Plane Change (NPC) P38
8. External Δv and Orbital Phasing Maneuvers (RTCC) P30.

The External Δv Guidance mode is used for maneuvers in which the required velocity-to-be-gained in local vertical coordinates, $\Delta v(LV)$, is specified by P31, P32, P34, P38 or a source external to the CMC by use of program P30. All other maneuvers are controlled by the Lambert Aimpoint Maneuver Guidance Mode in which the required cutoff velocity v_R is periodically computed by the Lambert subroutine during the maneuver to establish the desired intercept trajectory. Both External Δv and Lambert Aimpoint 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.

The following three subroutines are used repetitively in sequence (paragraph 5.3.3.2) during cross product controlled maneuvers to accomplish this function.

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 4 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
 - d. updating of the CSMMASS.
3. The Velocity-to-be-Gained Subroutine, which repetitively solves the Lambert intercept problem when in the Lambert Aimpoint guidance mode.

The prethrust subroutines of paragraph 5.3.3.3 initialize the powered maneuver programs for either the External Δv or Lambert Aimpoint guidance modes, and for the selected engine.

The CMC powered flight programs described by the computation subroutines presented in this paragraph are:

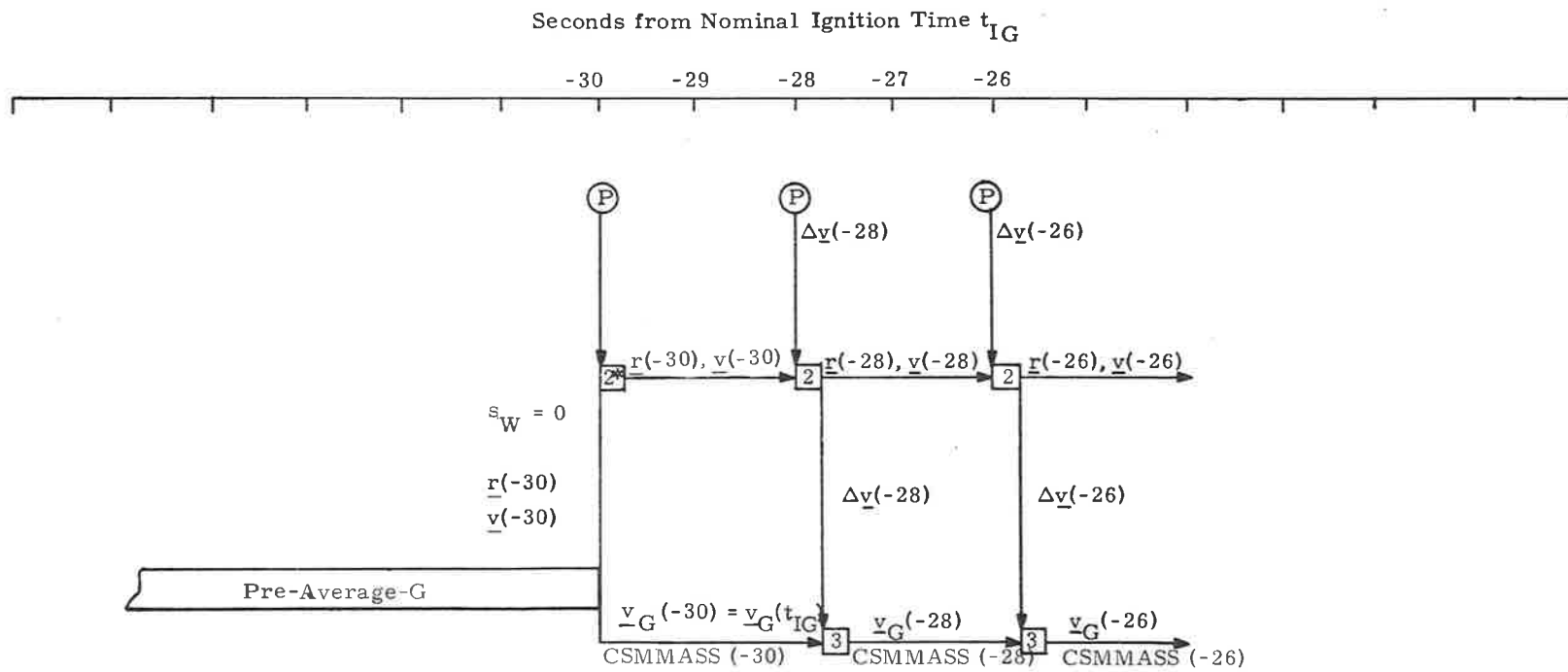
- P40 Service Propulsion System (SPS) Thrust Program
- P41 Reaction Control System (RCS) Thrust Program.

P40 provides active steering and engine-off commands. Maneuvers using RCS translation control (P41) are manually controlled and terminated by the astronaut while the CMC displays the required velocity-to-be-gained in control coordinates (paragraph 5.3.3.3.3).

The functions of the External Δv prethrust program (P30) are described in paragraph 5.3.3.3.1.

5.3.3.2 Powered Flight Guidance Computation Sequencing

The time sequencing of SPS powered flight subroutines for External Δv steering is illustrated in Figures 3.3-1 through 3.3-3, and for Lambert Aimpoint steering in Figures 3.3-4 through 3.3-6. These figures represent the sequence of operations for P40 SPS maneuvers lasting longer than 6 seconds. RCS controlled maneuvers (P41) 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



*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.

- | | |
|---|-----------------------------------|
| Ⓟ | Read and Zero PIPA Registers |
| ② | Average G Subroutine |
| ③ | Cross Product Steering Subroutine |

Figure 3.3-1. Ignition Countdown-External Δv Subroutine Sequencing for SPS Maneuvers

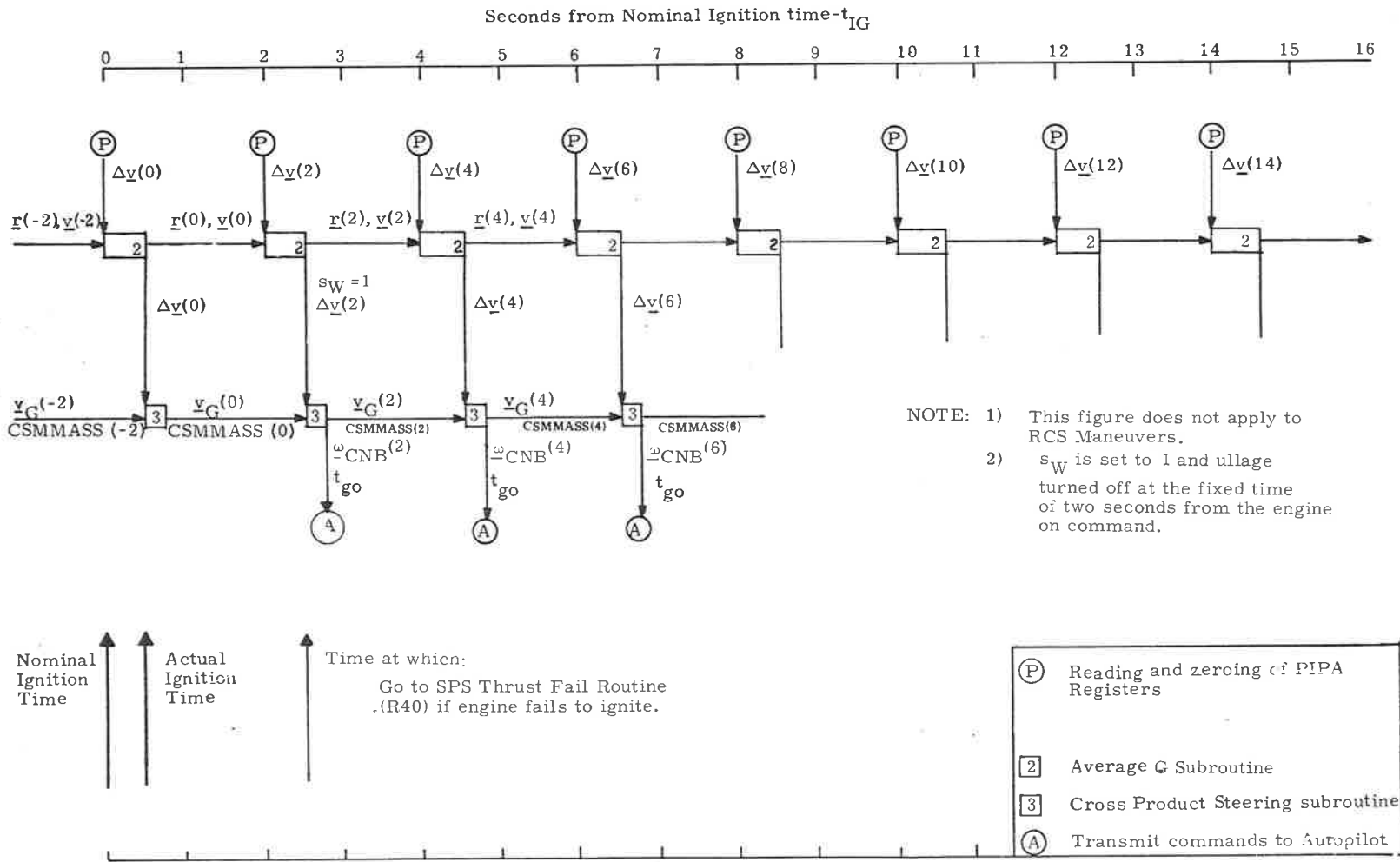


Figure 3.3-2. Normal External Δv Subroutine Sequencing for SPS Maneuvers (Maneuver Time Greater Than 6 Sec)

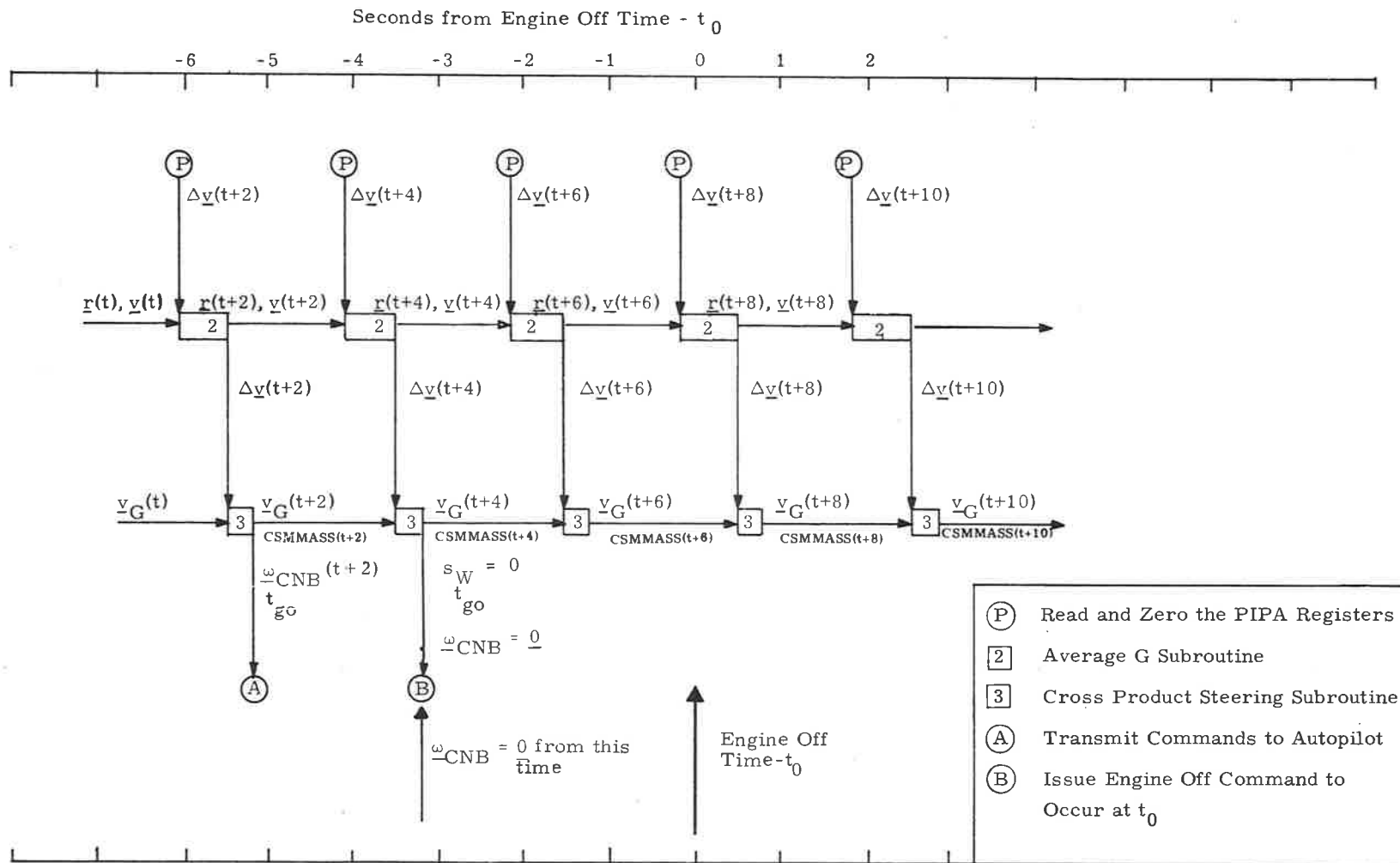


Figure 3.3-3. Engine-Off External Δv Subroutine Sequencing for SPS Maneuvers

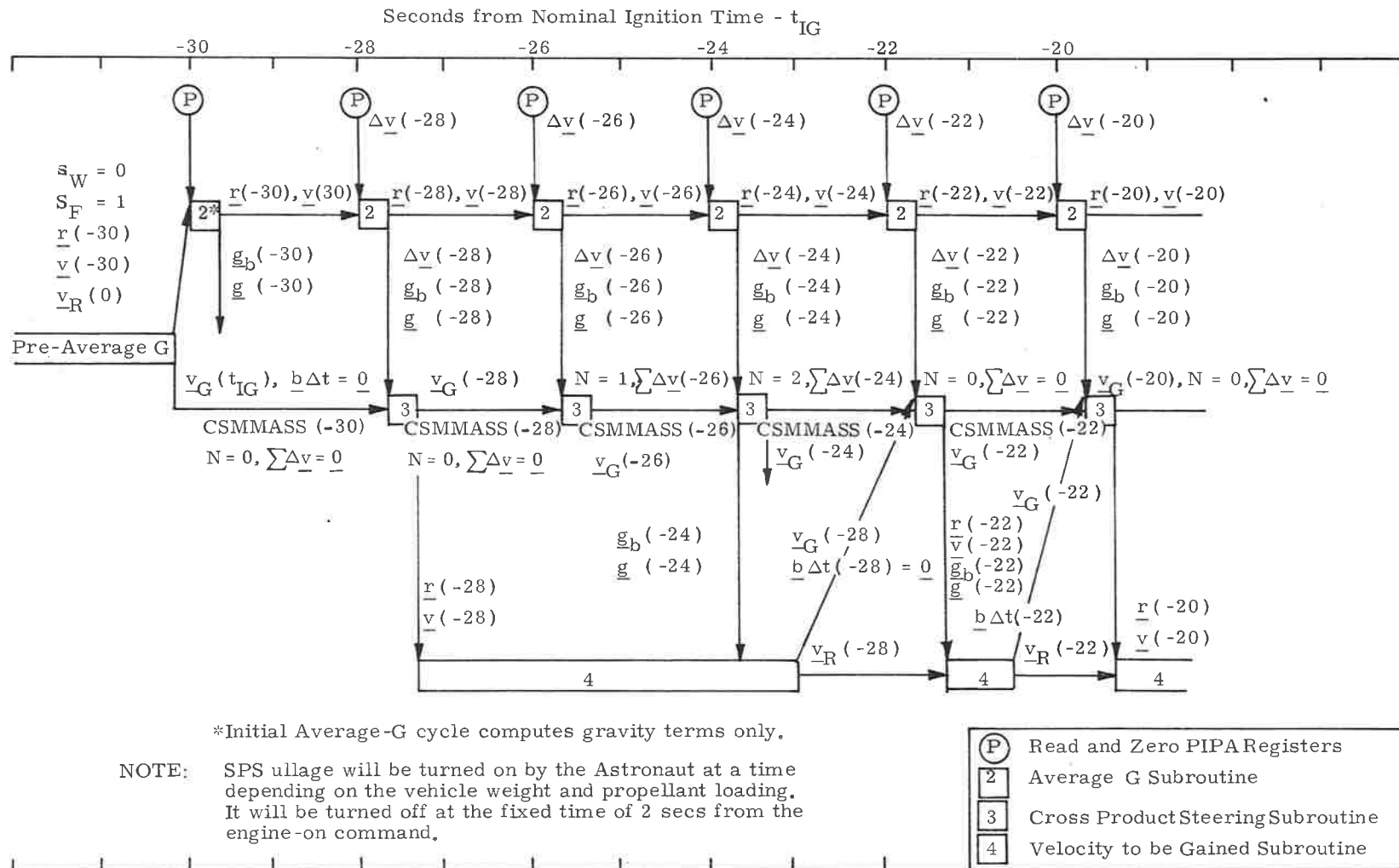


Figure 3.3-4. Ignition Countdown—Lambert Aimpoint Subroutine Sequencing for SPS Maneuvers

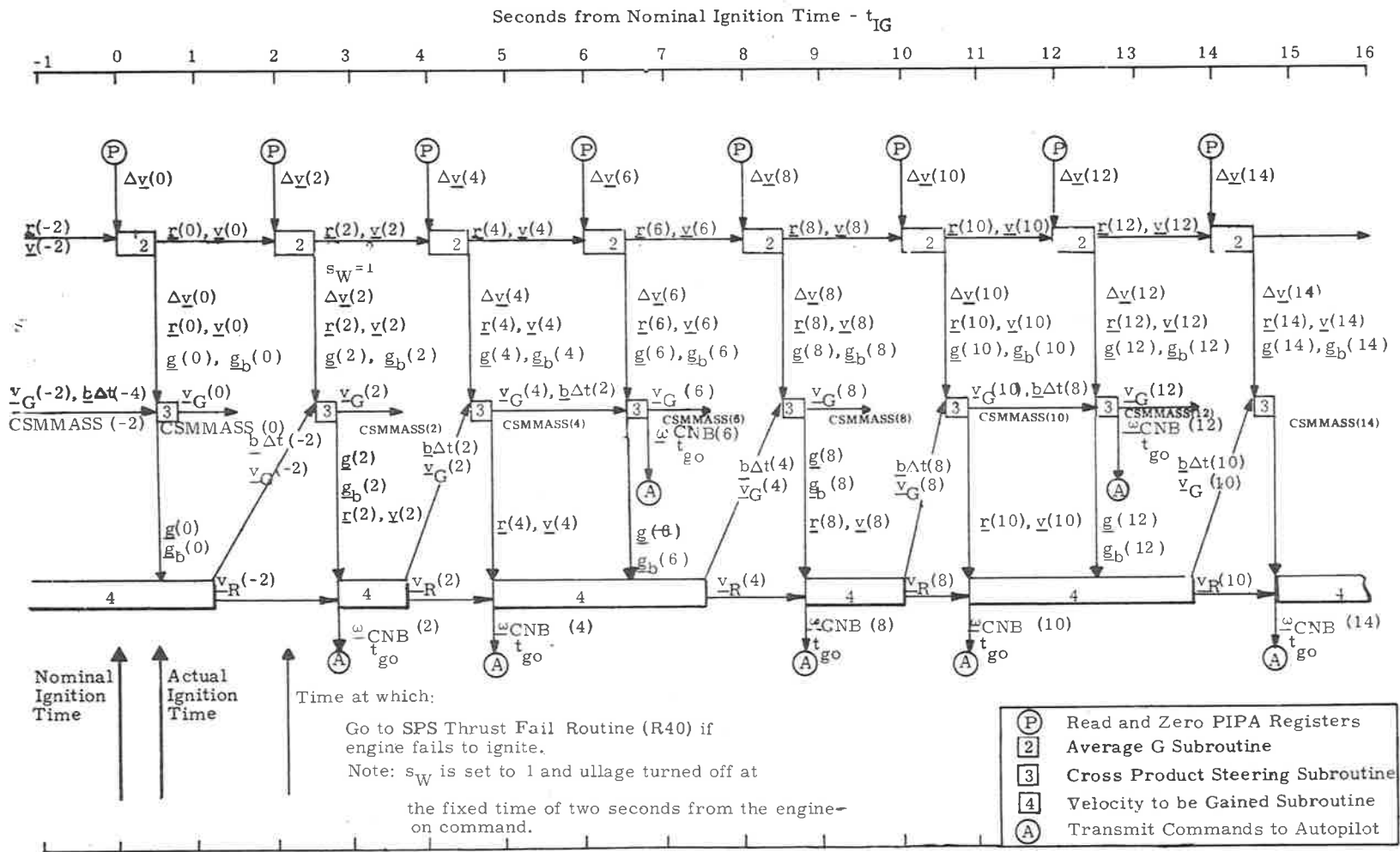


Figure 3.3-5. Normal Lambert Aimpoint Subroutine Sequencing for SPS Maneuvers (Maneuver Time Greater Than 6 Sec)

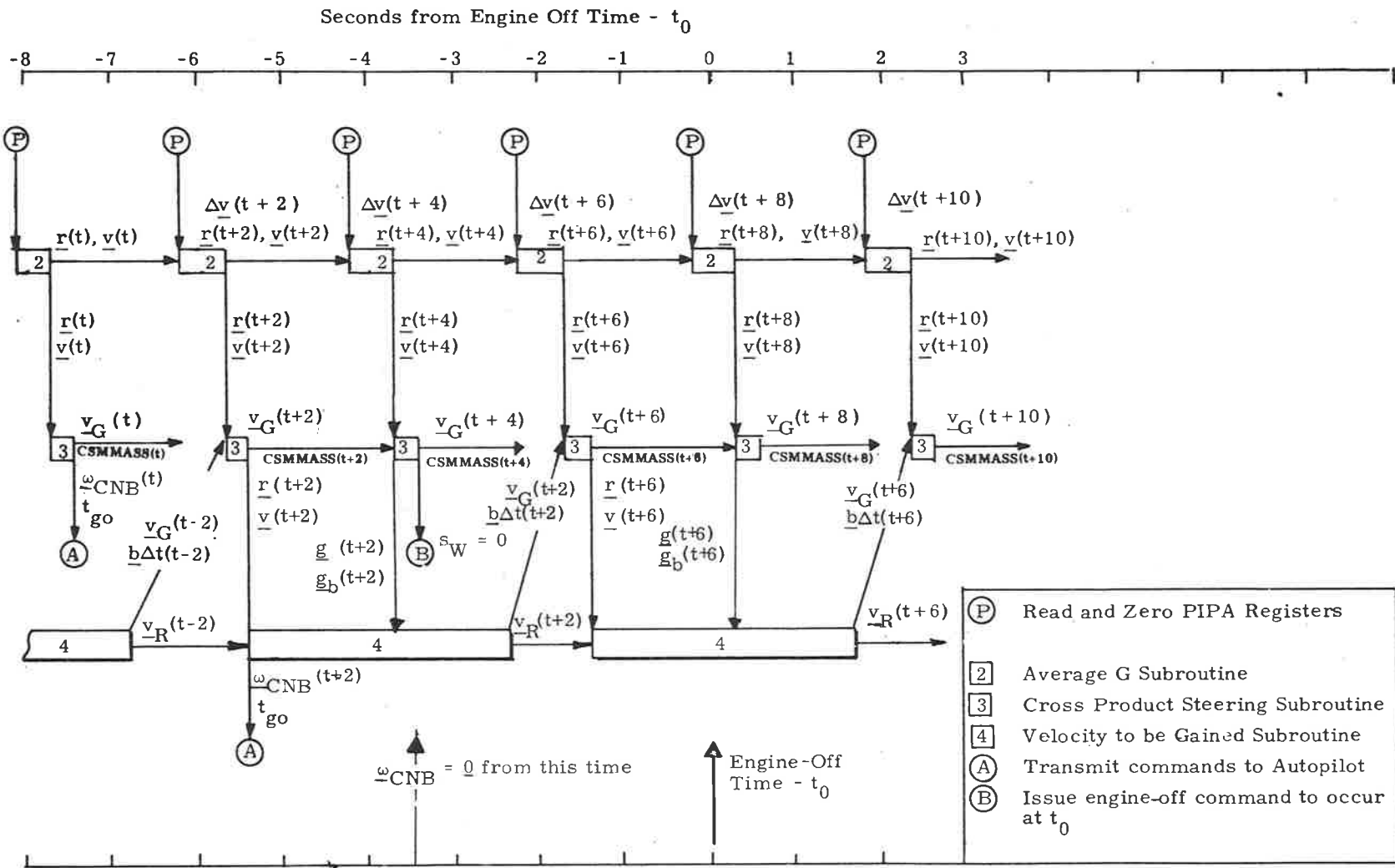


Figure 3.3-6. Engine-Off Lambert Aimpoint Subroutine Sequencing for SPS Maneuvers

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 approximately 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 prethrust 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 two seconds, and, as shown in the preceding figures, is initiated by the reading of the PIPA Δv registers. The various subroutines utilized during the 2-second cycle are sequenced in time as shown.

During Lambert Aimpoint steering, however, the velocity-to-be-gained (v_G) updating is anticipated to occur nominally every four seconds. But during SPS maneuvers the CMC computation occasionally prevents completion of the Lambert solution in two to four seconds. The program is capable of processing an N-cycle update where N can be greater than two. (In most cases, the time sequencing diagrams show only the nominal two- and four-second v_G updates.) This allows sequenced parallel computing functions, such as the demands of the autopilots and telemetry, to take place without interference with the basic 2-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 (prethrust) 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 set to be issued at the end of the estimated thrust period (SBFLAG = 0) or when the required Δv magnitude has been attained (SBFLAG = 1). See paragraph 5.3.3.3.5.

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

The guidance computer program that controls the various subroutines to create the powered flight sequence is called the Servicer Routine. The

sequence diagrams of Figures 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 Figures 3.3-1 to 3.3-6 are described in paragraph 5.3.2, and in paragraphs 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 after the engine-on command. Steering will be enabled (s_W set to 1) at the fixed time of 2 seconds after the engine-on command.

5.3.3.3 Prethrust Computations

The computations required before the thrusting maneuvers determine the following:

1. The desired thrust direction at ignition
2. The duration of the powered maneuver estimated at time $t_{IG} - 5$ to determine if there will be enough time to allow active steering for the SPS
3. The middle gimbal angle required for the burn attitude (indicates 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 Δv guidance mode and the Lambert Aimpoint Maneuver guidance mode. The prethrust computations required for the External Δv mode are presented in paragraph 5.3.3.3.1. Those required for the Lambert Aimpoint Maneuver mode are described in paragraph 5.3.3.3.2. The initial IMU alignment computations and maneuver time logic is summarized in paragraph 5.3.3.3.3. The following description of prethrust computations applies to both SPS (P40) and RCS (P41) maneuvers.

5.3.3.3.1 External Δv Maneuver Prethrust Computations. - External Δv maneuvers guidance is normally used to control orbital phasing maneuvers or externally targeted maneuvers in which a constant thrust attitude is desired. The guidance program accepts input data via the DSKY (P30, P31, P32, P34, and P38) or

the telemetry uplink (P27) in the form of three components of an impulsive Δv (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 Δv (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 CMC state vector is extrapolated to ignition time for these computations.

The prethrust computations required for the External Δv guidance mode are shown in Figure 3.3-7. The following parameter definitions refer to this figure.

Δv (LV) Specified velocity change in the local vertical coordinate system of the CMC at the time of ignition. This is an input parameter.

$$\Delta v$$
 (LV) =
$$\begin{pmatrix} \Delta v_X \\ \Delta v_Y \\ \Delta v_Z \end{pmatrix}$$

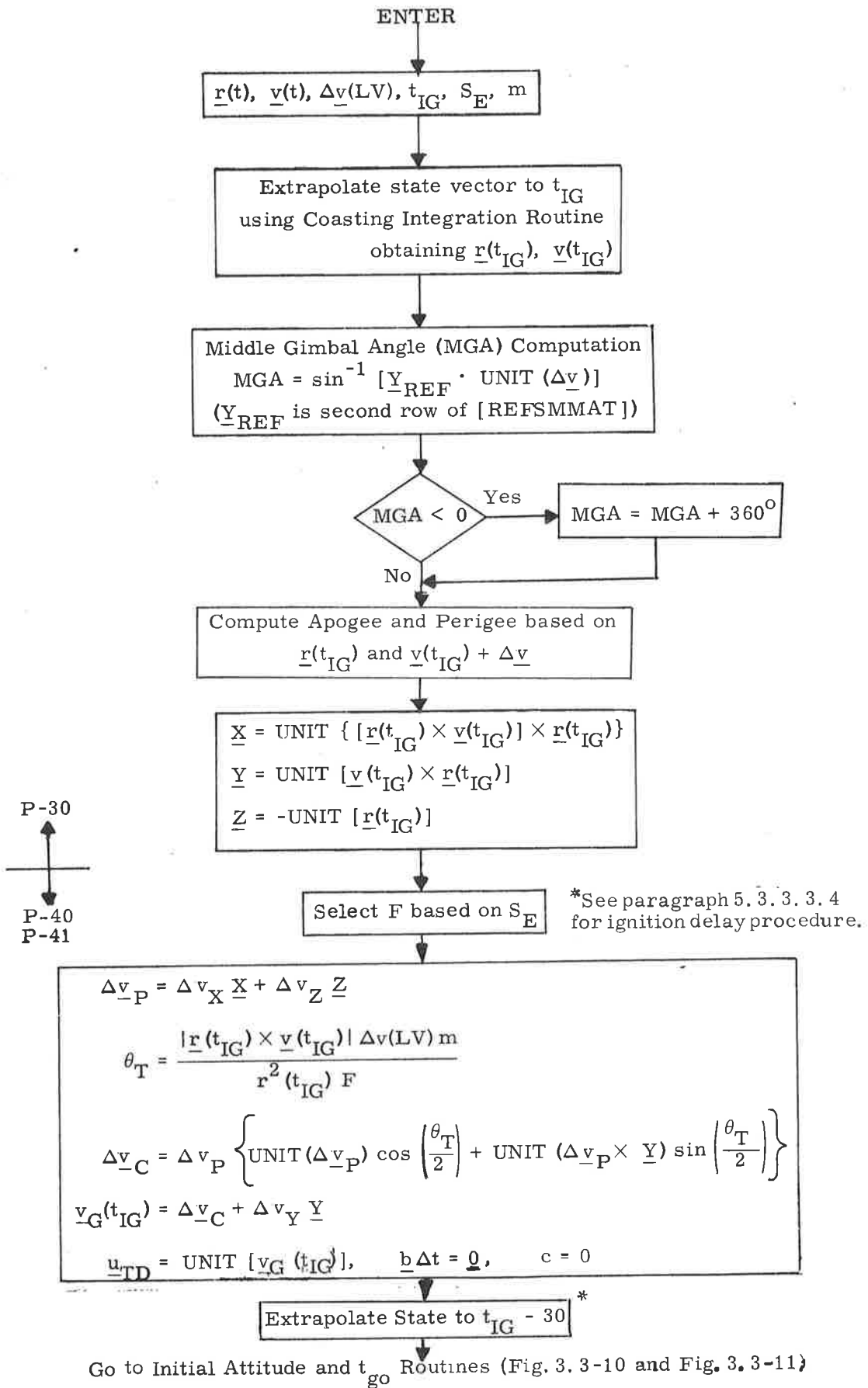
Δv Specified velocity change in Basic Reference Coordinates.

t_{IG} Ignition time, an input parameter.

Δv_P The inplane velocity components of Δv (LV) in the Basic Reference Coordinate System.

θ_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 Δv . This angle is observed by the crew to check for gimbal lock tolerance.



P-30

P-40
P-41

Figure 3.3-7. External Δv Prethrust Routine

m	Vehicle mass
S_E	Engine select switch $\begin{cases} 0 & \text{P40 (SPS)} \\ 1 & \text{P41 (RCS)} \end{cases}$
F	Prestored nominal thrust based on the engine selection switch S_E $F = F_{SPS}$ for the SPS (P40) $F = F_{RCS}$ for the RCS (P41) The values of the various engine thrust levels are listed in subsection 5.9. In the case of P41, RCS, the astronaut can select either a 2- or 4-jet translation maneuver.
Δv_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 constant is set equal to zero for External Δv maneuvers.
$\underline{b}\Delta t$	Δt times the \underline{b} -vector ($\dot{\underline{v}}_R - \underline{g}$), computed as the incremental change of the velocity-to-be-gained vector over one sample period minus current gravity.

The inplane External Δv maneuver angle compensation involving Δv_P and Δv_C is illustrated in Figure 3.3-8.

5.3.3.3.2 Lambert Aimpoint Maneuver Prethrust Computations.—The objective of the Lambert Aimpoint 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. Targeting parameters are determined by CMC targeting programs P33, P35, or P36.

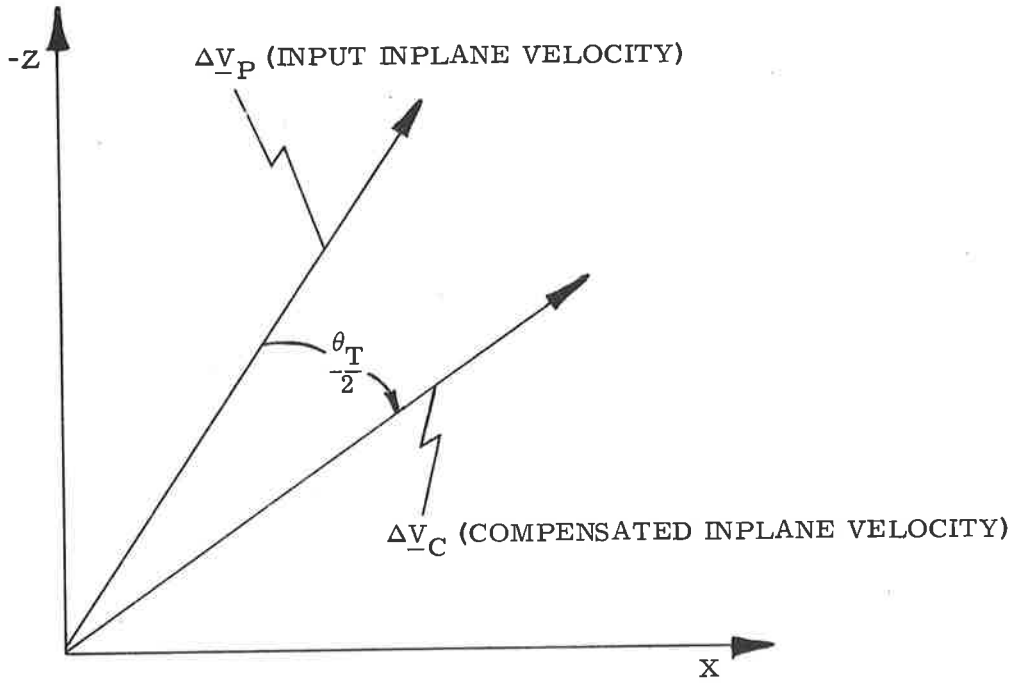


Figure 3.3-8. Inplane External Δv Maneuver Angle Compensation

The table below summarizes the maneuvers that can be performed by Lambert Aimpoint Maneuver Guidance and shows the maneuver's corresponding targeting program.

Maneuvers Controlled by Lambert Aimpoint Guidance	
<u>Maneuver</u>	<u>Premaneuver Targeting Program</u>
Corrective Combination (NCC)	P33
Terminal Phase Initiation (TPI)	P35
Terminal Phase Midcourse (TPM)	P36
Targeting parameters computed by these programs are:	
1. Ignition time t_{IG}	
2. Time of intercept of conic target aimpoint vector t_2	
3. Conic target aimpoint vector $\underline{r}(t_2)$	

The Lambert Aimpoint Maneuver guidance mode basically uses the Lambert Subroutine of paragraph 5.5.6 called through the Initial Velocity Subroutine of paragraph 5.5.11 during the prethrust phase to determine the required initial thrust direction \underline{u}_{TD} . The prethrust computations required for the Lambert Aimpoint guidance mode are shown in Figure 3.3-9. The primary operation shown in this figure is the computation of the required intercept velocity \underline{v}_R at the ignition time t_{IG} . The following parameter definitions refer to Figure 3.3-9.

$\underline{r}(t_2)$ Offset target intercept position vector at time t_2 . This parameter is an input to the Lambert Aimpoint prethrust subroutine.

- t_2 Intercept time of arrival associated with the offset target vector $\underline{r}(t_2)$. This is an input parameter.
- S_F See discussion in paragraphs 5.3.3.4 and 5.3.3.5—controls first pass through cross-product steering routine and first pass through velocity-to-be-gained subroutine.
- S_R Target rotation switch set in P33, P35 and P36, indicating whether the target vector was rotated into the orbital plane due to proximity to 180° 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 paragraph 5.5.11.
- ϵ 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 $\underline{r}(t_{IG})$.
- $\underline{b}\Delta t$ Δt times the \underline{b} -vector ($\dot{\underline{v}}_R - \underline{g}$), computed as the incremental change of the velocity-to-be-gained vector over one sample period minus current gravity.
- MGA The angle equivalent to the IMU middle gimbal angle when the vehicle X axis is aligned along \underline{v}_G .

- c The cross product steering constant.
Set equal to zero if the prethrust program was the External Δv Program P30 (Figure 3.3-7). P41 sets $c = 0$ (Figure 3.3-9). P40 sets $c = 0$ for all external Δv burns and sets $c = 1$ for Lambert computations.

The rotation switch S_R is set in the Lambert Aimpoint targeting programs P33, P35 and P36 to indicate whether the target aimpoint vector was within a specified cone angle, ϵ , measured from the 180-deg 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 so that excessive plane change and Δv requirements are avoided about the 180-deg transfer-angle condition. In the intercept targeting programs P33, P35, and P36 the cone angle ϵ is set equal to 15 deg; and CSM transfer-angles between 165 deg and 195 deg are normally avoided in the targeting procedure. If a transfer angle condition falling within this 180 deg \pm 15 deg sector ($S_R = 1$) is intentionally selected, or results from a rendezvous mid-course correction maneuver (P36) during an intercept trajectory targeting for more than 180 deg, the Lambert Aimpoint Maneuver Prethrust Routine of Figure 3.3-9 increases the Initial Velocity Subroutine cone angle ϵ to 45 deg so that the CSM 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-deg cone angle ϵ of programs P33, P35 and P36 ($S_R = 0$), ϵ is decreased to 10 deg in Figure 3.3-9 to reduce the possibility of the transfer angle's changing from outside to inside the cone angle during a powered maneuver.

5.3.3.3.3 Initial IMU Alignment and Maneuver Time-to-Go Computations. —
Initial Attitude Routine. The following prethrust computations and functions are required for both the External Δv and Lambert Aimpoint guidance modes:

1. Determination of the preferred IMU alignment for the thrusting maneuver (paragraph 5.1.4.2.1).

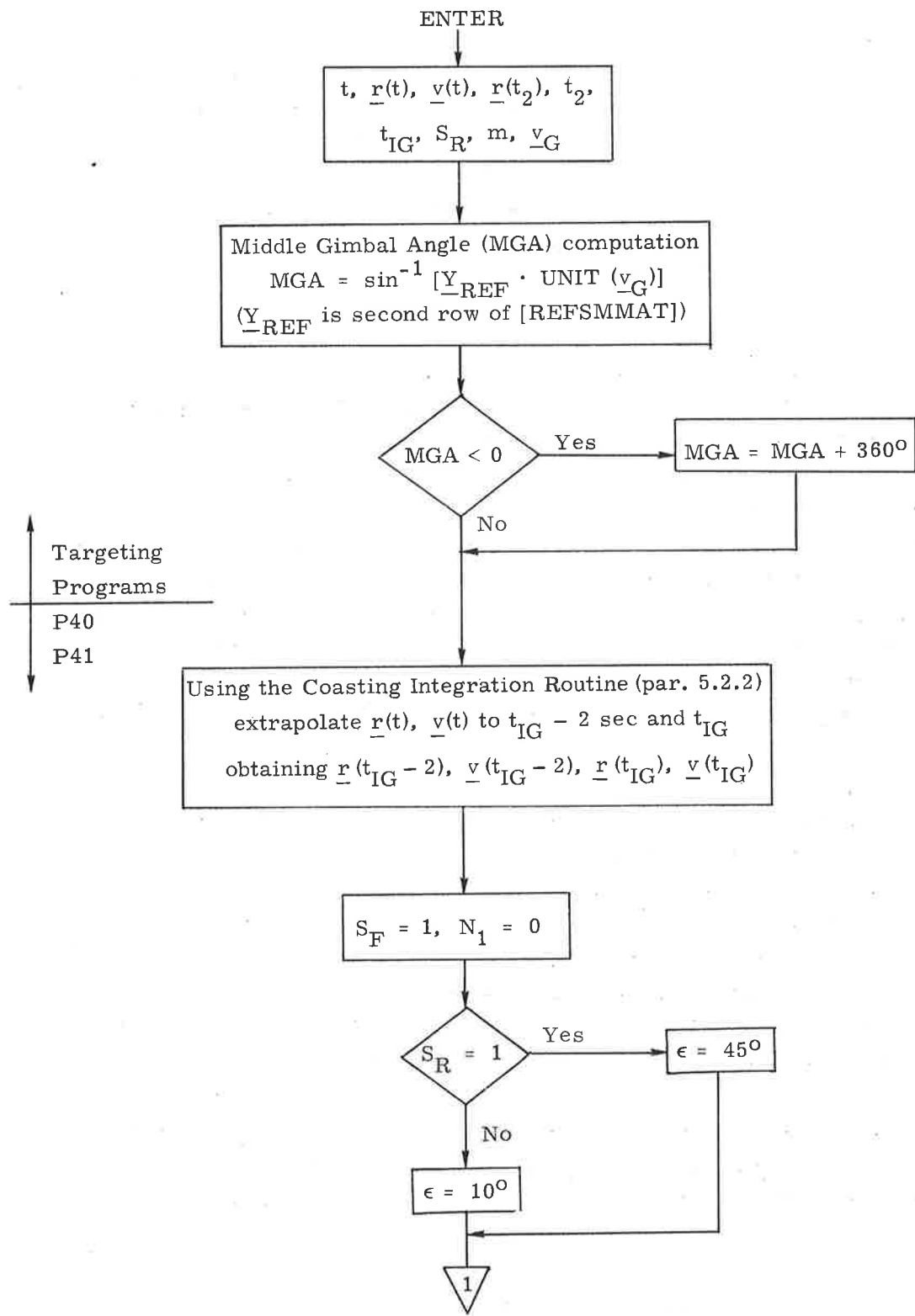


Figure 3.3-9. Lambert Aimpoint Maneuver Prethrust Routine (Sheet 1 of 2)

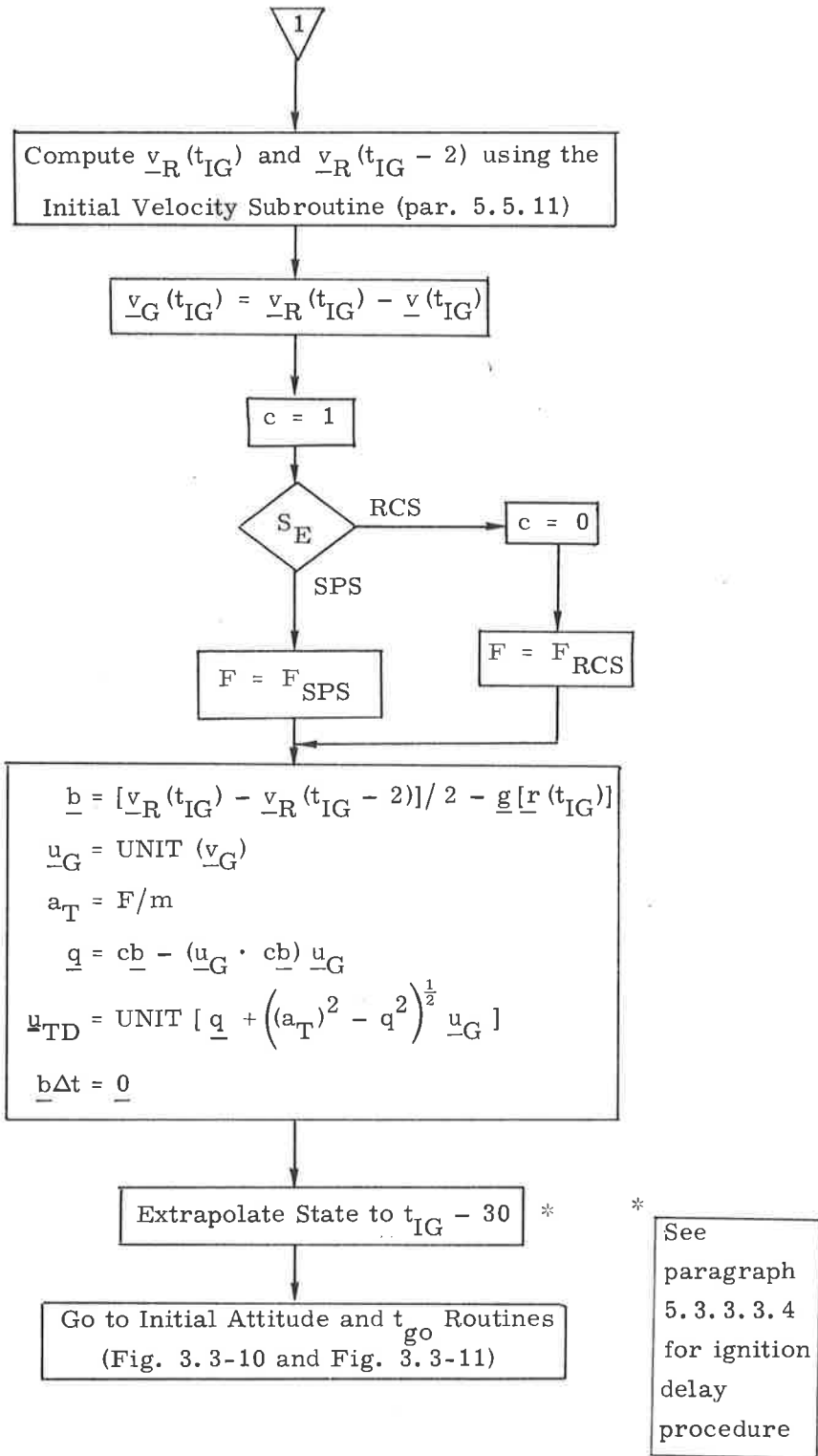


Figure 3.3-9. Lambert Aimpoint Maneuver Prethrust Routine (Sheet 2 of 2)

2. Alignment of the vehicle thrust axis along the desired initial thrust direction, \underline{u}_{TD} .
3. Estimation of the 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 (paragraph 5.6.3.3).
[SMNB]	Transformation matrix from the Stable Member Coordinate System to the Navigation Base Coordinate System (paragraph 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 are the electrical pitch and yaw trim angles, respectively.
	p_0 is the mechanical pitch trim angle equal to -2.15 deg about the spacecraft Y axis.
	y_0 is the mechanical yaw trim angle equal to +0.95 deg 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 Figure 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 Δv 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 before engine ignition, as shown in Figure 3.3-11.

With reference to Figure 3.3-11

s_I	Impulse switch
	s_I $\left\{ \begin{array}{l} 1 \text{ Indicates a maneuver less than} \\ \text{6 seconds and no steering is} \\ \text{required} \\ 0 \text{ Indicates a maneuver duration} \\ \text{of at least 6 seconds, and} \\ \text{normal steering will occur} \end{array} \right.$
F_L	Ullage thrust for 2 jets
K_1	SPS impulse velocity acquired in a one-second maneuver for a unit mass vehicle; this is an erasable-memory parameter.
F_{01}	SPS minimum impulse constant equal to the slope of minimum impulse curve from 0 to 1 second; this is an erasable-memory parameter.
F_{16}	Steady state thrust of the SPS engine. This value is used only in the short burn logic and is an erasable-memory parameter.
V_e	Exhaust velocity.

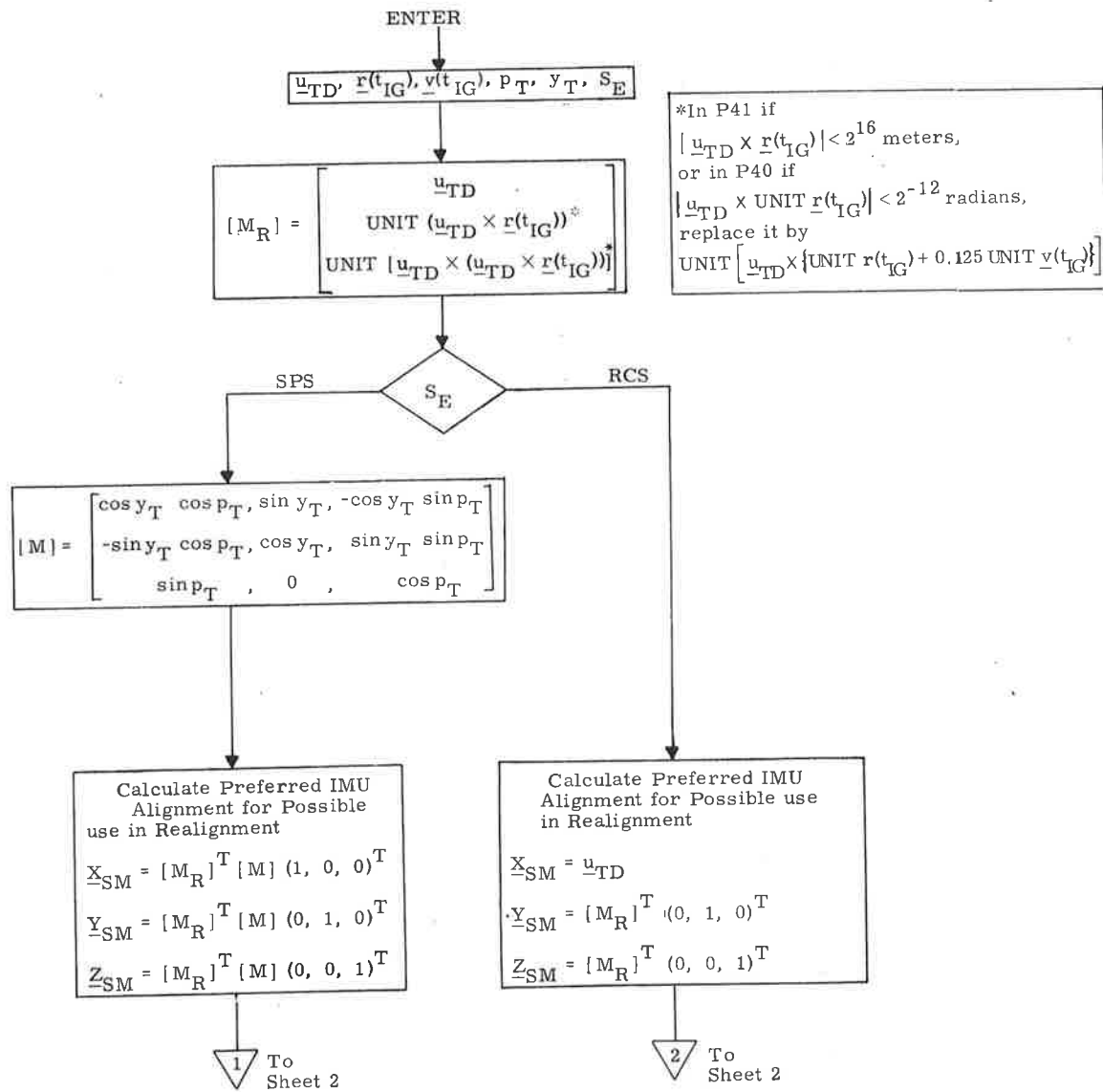


Figure 3.3-10. Initial Attitude Routine (Sheet 1 of 2)

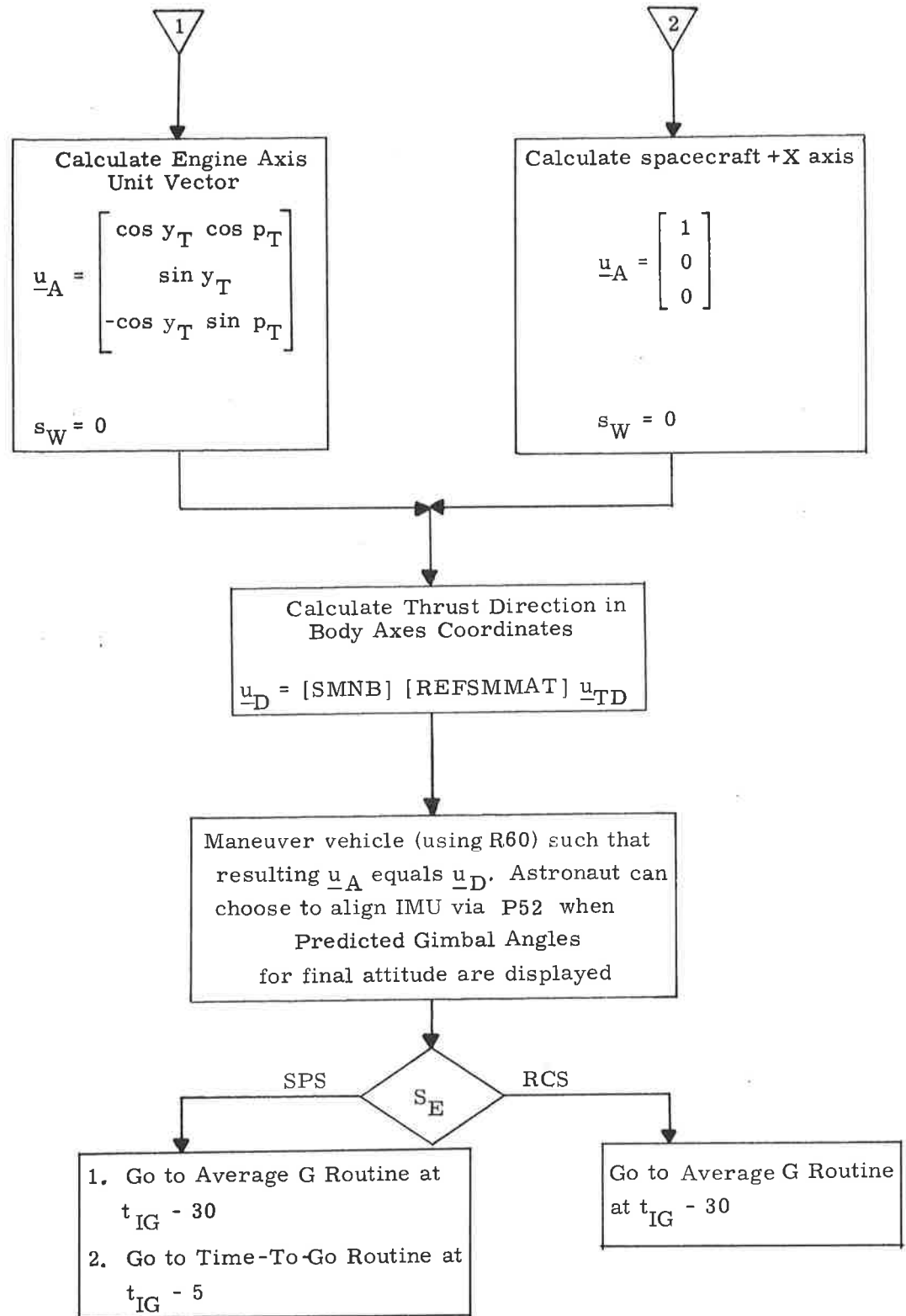


Figure 3.3-10. Initial Attitude Routine (Sheet 2 of 2)

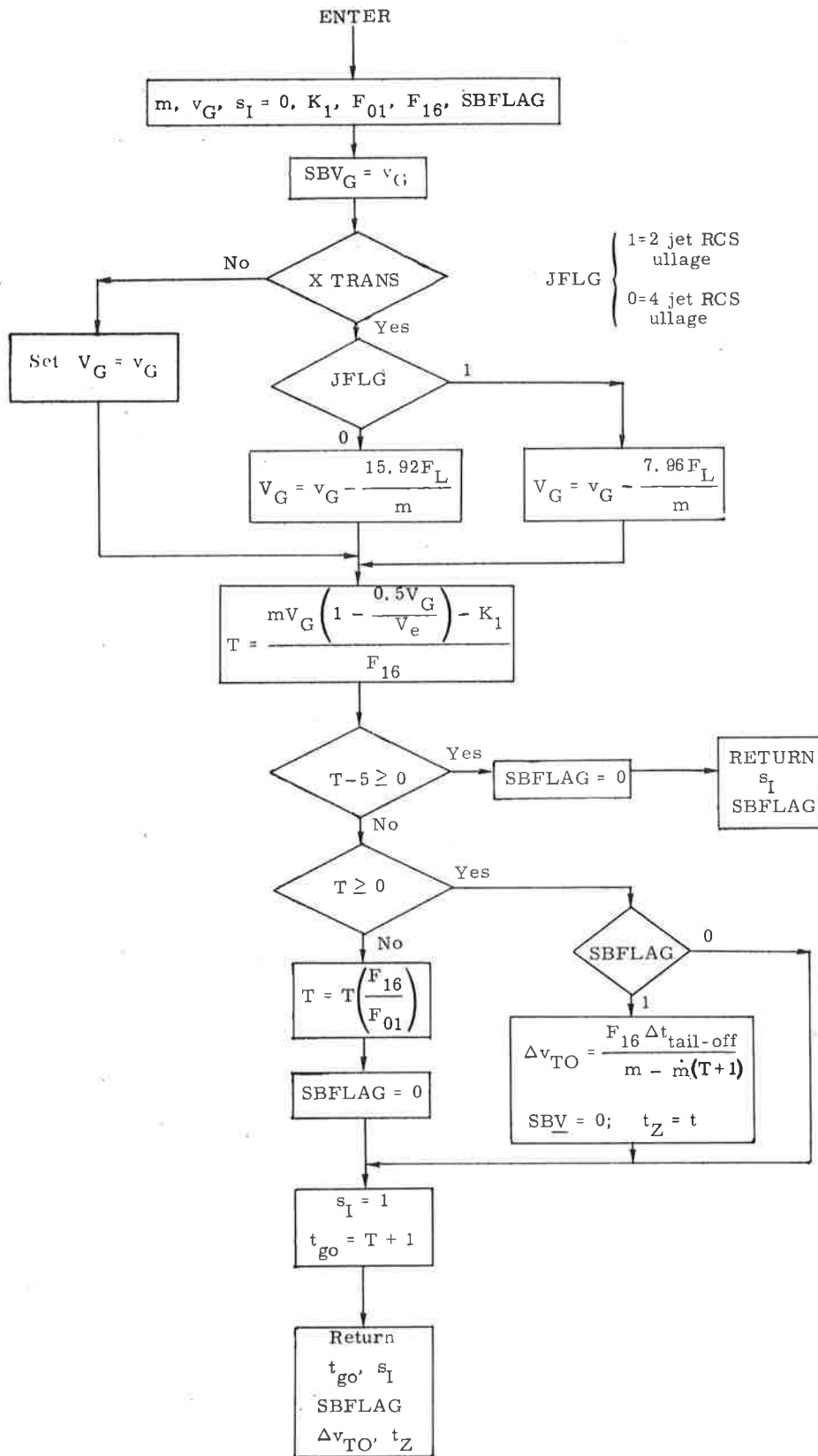


Figure 3.3-11. Time-to-go Prediction Routine

SBV _G	Targeted velocity magnitude
Δv_{TO}	Expected tailoff velocity magnitude
SBV	Velocity change from t_Z to last PIPA read
t_Z	Time SBV zeroed

The initial computation in Figure 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 Figure 3.3-11. This subroutine is performed at approximately five seconds before ignition, the PIPAs are read at six seconds before ignition, and the ullage is terminated at two seconds after ignition. Therefore, about eight seconds of ullage are accounted for, as shown in Figure 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 (X TRANS) is indicated.

If the SPS (P40) is chosen, a check is made (Figure 3.3-11) 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_1 and F_{01} . (See subsection 5.8.) In this case, SBFLAG = 0 and no active steering is attempted ($s_I = 1$). If the maneuver time is greater than one second but less than 6 seconds, t_{go} is computed as shown in Figure 3.3-11, and again no active steering is attempted. If SBFLAG = 1, the complex impulsive burn logic variables are initialized as shown in Figure 3.3-11. If the estimated maneuver time is greater than 6 seconds, SBFLAG = 0, active steering is used, and t_{go} computations are performed during the maneuver. If the RCS (P41) is chosen for the maneuver, the t_{go} prediction calculation is not made. If the estimated maneuver time t_{go} for the SPS is less than 6 seconds, either the engine-off signal is set for the actual ignition time plus t_{go} (SBFLAG = 0), or the Complex Impulsive Burn Routine is initiated shortly after ignition (SBFLAG = 1).

5.3.3.3.4 Ignition Delay Procedures Caused by Prethrust Computations.—The normal prethrust computations (paragraphs 5.3.3.3.1 and 5.3.3.3.2) require an extrapolation of the CSM state vector to thirty seconds before nominal ignition time, i. e., $t_{IG}-30$. If the Coasting Integration Routine does not complete the extrapolation before $t_{IG}-42.5$ 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 is greater than 12.5 seconds. This will permit a 5 second blanking interval [Section 4 State Vector Integration (MID to AVE) Routine (R41)]. R41 allows Average-G initialization to occur at least 5.6 seconds after completion of last integration step.
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.

5.3.3.3.5 Complex Impulsive Burn Routine.— This routine (called at ignition only if both s_I and SBFLAG = 1) measures the accumulated Δv every 0.1 seconds and shuts down the SPS engine when the desired v_G has been attained. The parameters used in the routine are defined as follows—with particular reference to Figure 3.3-11a:

Δv_{pr}	Velocity magnitude for previous pass
t_p	Time at start of previous pass
Δv	Velocity magnitude for present pass
t	Time at start of present pass
$\left. \begin{array}{l} PIPAX_p \\ PIPAY_p \\ PIPAZ_p \end{array} \right\}$	Components of velocity change from t_Z to last PIPA read, SBV

See also paragraph 5.6.13.

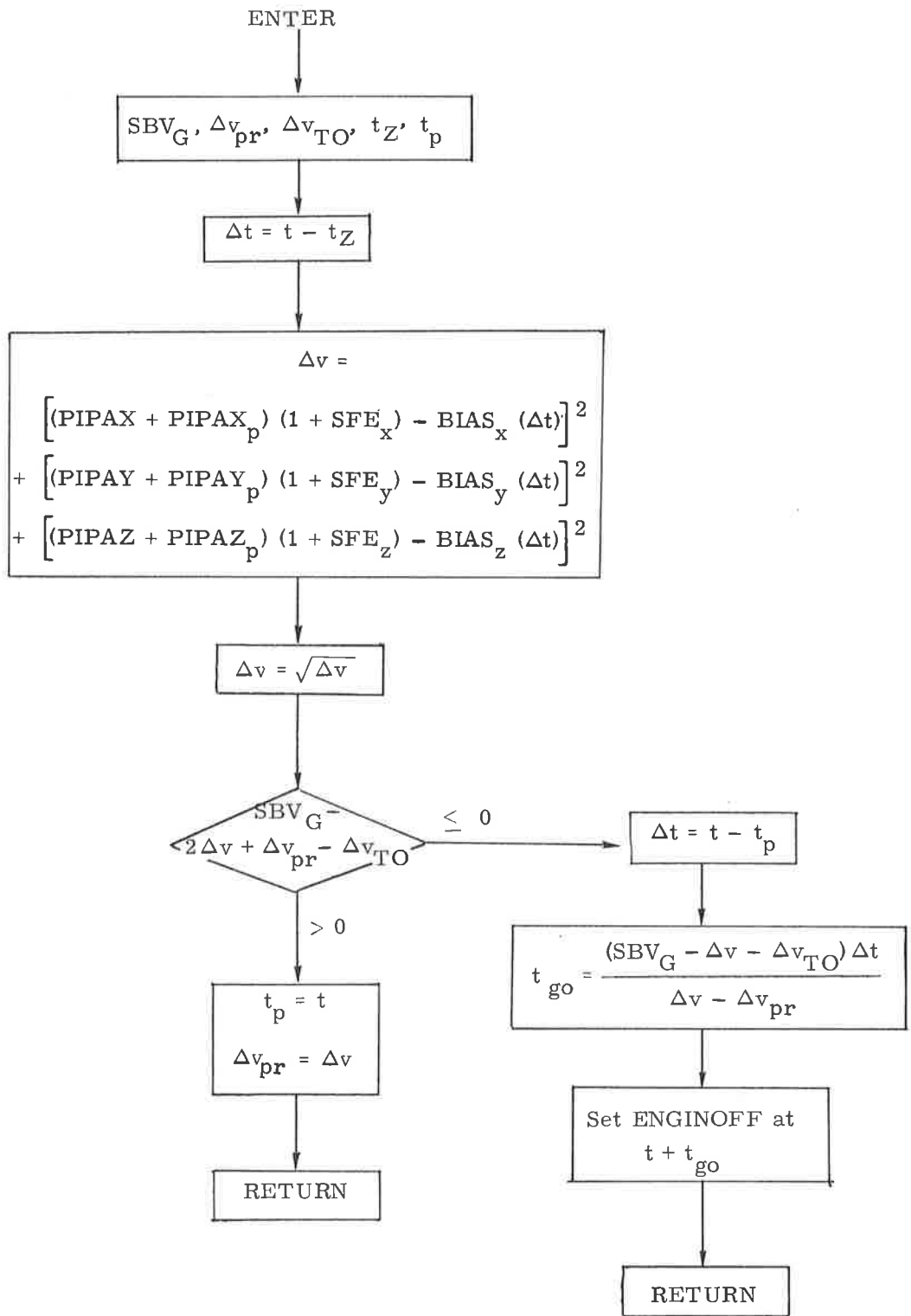


Figure 3.3-11a. Complex Impulsive Burn Routine

5.3.3.4 Cross Product Steering Routine

The cross product steering concept is the basic control concept for both External Δv and Lambert Aimpoint Guidance modes. The cross product steering logic is shown in Figure 3.3-12. The following parameter definitions apply to this figure.

Δv_{SM}	PIPA measured velocity vector over the computation cycle Δt in IMU coordinates
Δv	PIPA measured velocity vector over the computation cycle Δt expressed in the Basic Reference Coordinate System.
Δv_p	A constant that establishes the Δv that must be sensed in a 2-second computation interval. It is 2 ft/sec (erasable), which is approximately equivalent to thrusting at 5 percent 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 if non-impulsive. s_W is 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.
Δv_G	$b\Delta t - \Delta v$.
V_e	Engine exhaust velocity = gI_{SP} .
K	Guidance steering gain required for desired dynamic response of the combined powered guidance and thrust control (autopilot) loops.
ω_C	Commanded attitude rate in the Basic Reference Coordinate System.
ω_{CNB}	Commanded attitude rate in Navigation Base Coordinates, which is sent to the CSM attitude control system.

c The cross product steering constant c, used by P40 and P41 in the equation

$$\underline{\Delta m} = \underline{cb\Delta t} - \underline{\Delta v}$$

and certain pre-thrust computations. It will be zeroed by P40 prethrust for external Δv burns and by P41 pre-thrust. It is set by P40 prethrust to 1 for Lambert burns.

$\underline{\Delta m}$ $\underline{cb\Delta t} - \underline{\Delta v}$

LOFLG This flag prevents a premature engine fail indication after a manual engine start

$$\text{LOFLG} \begin{cases} 1 & \text{steer} \\ 0 & \text{no steer} \end{cases}$$

$\underline{\Sigma\Delta v}$ Accumulated PIPA readings used to generate an extrapolation of the Lambert solution. When a new solution becomes available $\underline{\Sigma\Delta v}$ is set to zero.

N Number of cycles between Lambert solutions. 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. When a new solution becomes available, N = 0.

S_F Lambert first pass flag. The Lambert first pass starts at approximately $t_{IG} - 28$. After the first pass is completed, it is set to zero.

$$S_F \begin{cases} 1 & \text{First Lambert pass. This results in bypass of the } \underline{b\Delta t} \text{ computation.} \\ 0 & \text{Not first Lambert pass.} \end{cases}$$

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 prethrust alignment. When s_W is

set to 1, active steering is performed. The time-to-go t_{go} computation, and steering commands ω_{CNB} are performed as shown in Figure 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 Figure 3.3-12, the logic switch s_W is set to zero by the sequencing routine. When s_W is zero, the only functions of the cross product steering routine are to update the velocity-to-be-gained vector \underline{v}_G with $\Delta\underline{v}$ and $\underline{b}\Delta t$ and to update the CSMMASS.

The steering command generated by this routine is ω_{CNB} 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:

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

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

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

1. 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. N and $\Sigma\Delta\underline{v}$ are used when a new Lambert solution is available.
2. 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:
 - a. If the present Δv is less than Δv_p , where Δv_p is approximately equivalent to thrusting at 5 percent full SPS thrust with fully loaded vehicle for 2 sec, the steering rate commands are zeroed, c. g. tracking is stopped and the SPS Thrust Fail Routine (R40) (paragraph 5.3.3.6) is entered. Δv_p is an erasable quantity. If the present Δv is greater than Δv_p , the CSMMASS is decremented by 2 m.
 - b. 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.

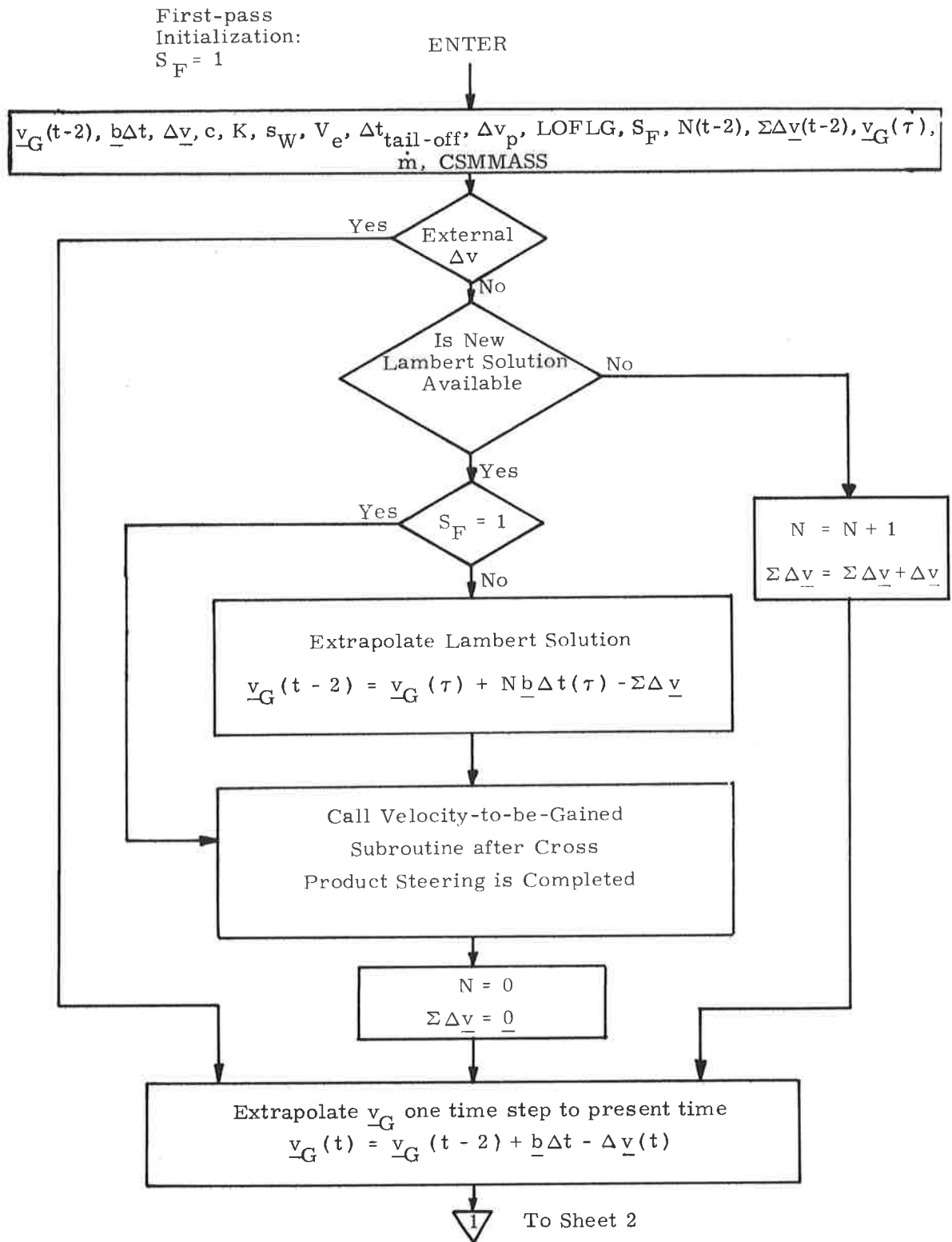


Figure 3.3-12. Cross Product Steering Routine—Update \underline{v}_G Section (Sheet 1 of 2)

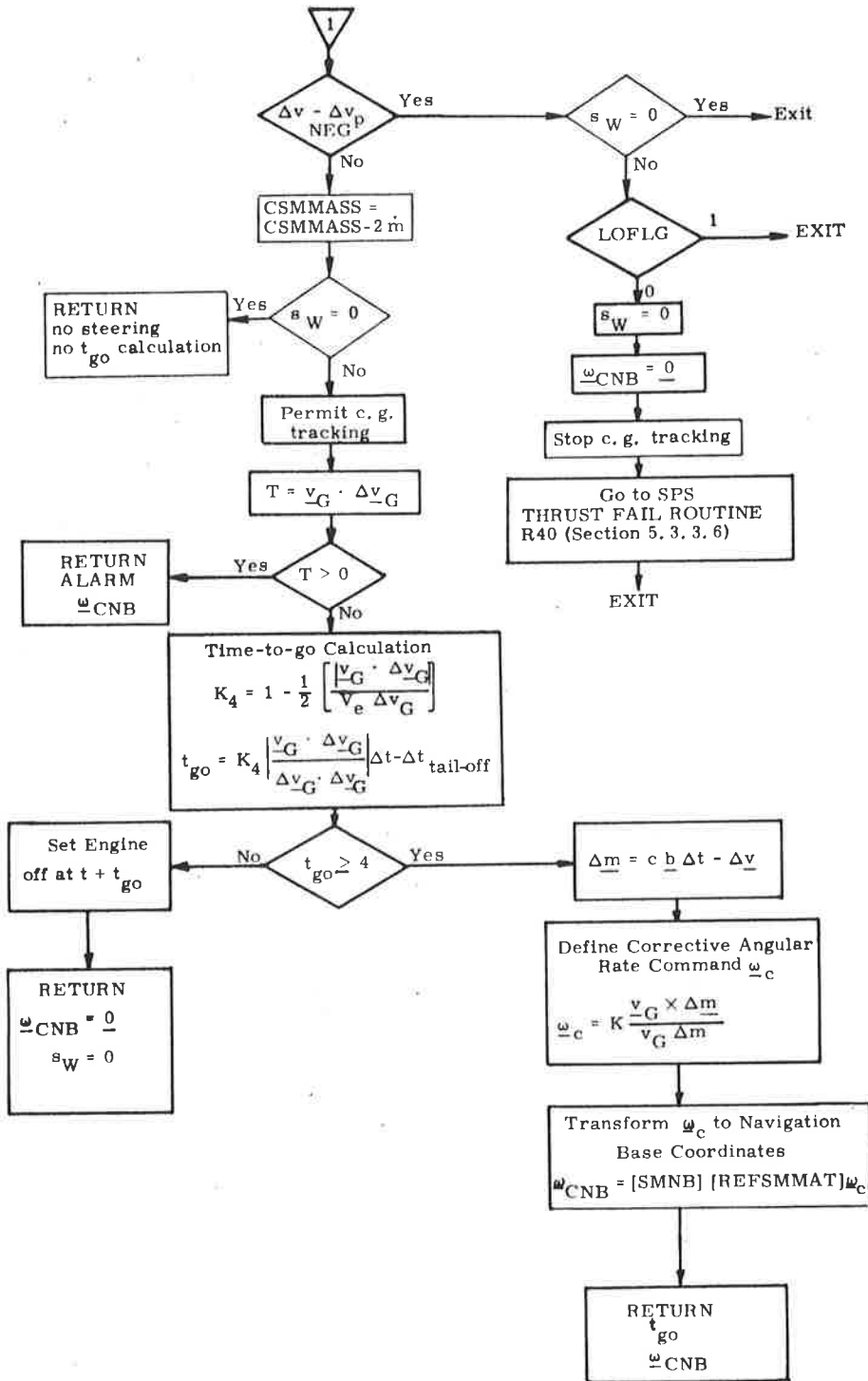


Figure 3.3-12. Cross Product Steering Subroutine — t_{go} and Steering Command Computations (Sheet 2 of 2)

5.3.3.5 Velocity-to-be-Gained Routine

The velocity-to-be-gained computations shown in Figure 3.3-13 are those carried out during the Lambert Aimpoint powered flight guidance. The velocity-to-be-gained computation for the External Δv guidance mode is simpler than that for the Lambert Aimpoint guidance mode. The External Δv velocity-to-be-gained computation is that shown in the Cross Product Steering Routine of Figure 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 Δv mode as shown in Figure 3.3-7.

The velocity-to-be-gained computations for the Lambert Aimpoint 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 two to four seconds. The program is capable of processing an N-cycle \underline{v}_G update (Figure 3.3-13).

The following parameter definitions refer to Figure 3.3-13.

$\left. \begin{array}{l} \underline{r}(\tau) \\ \underline{v}(\tau) \\ \tau \end{array} \right\}$	Active vehicle state vector at start of Lambert solution.
$\underline{r}(t_2)$	Offset target intercept position vector at time t_2 . This parameter is determined by the preceding targeting program (P33, P35 or P36).
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.
$\underline{v}_R(\tau - \Delta\tau)$	Required velocity vector from the preceding Lambert computation cycle.

$\Delta\tau$	Time interval between the current and previous Lambert computation cycle = $(N+1)\Delta t$.
S_R	Target rotation switch set in P33, P35, or P36, indicating that the target vector was rotated into the orbital plane due to proximity to 180° transfer.
S_R	$\left\{ \begin{array}{l} 1 \text{ Target vector rotated} \\ 0 \text{ No rotation} \end{array} \right.$
$\underline{g}(t)$	Most recent total gravity acceleration vector (paragraph 5.3.2).
$\underline{g}_b(t)$	Component of the most recent earth gravity acceleration vector representing earth oblateness effects (paragraph 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 (paragraph 5.5.11).
ϵ	Initial Velocity Subroutine parameter.
$\underline{v}_R(\tau)$	Required velocity vector at time t.
$\underline{v}_G(\tau)$	Velocity-to-be-gained vector at time t.
$\underline{b}\Delta t(\tau)$	\underline{b} -vector solution from most recent Lambert computation.
N	Number of cycles between Lambert solutions. The CMC computations during SPS maneuvers occasionally prevent completion of the Lambert solution in two to four seconds.
S_F	Lambert first pass flag. The Lambert first pass begins at approximately $t_{IG}-28$. After the first pass is completed, it is set to zero.
S_F	$\left\{ \begin{array}{l} 1 \text{ First Lambert pass. This results in bypass of the } \underline{b}\Delta t \text{ computation.} \\ 0 \text{ Not first Lambert pass.} \end{array} \right.$

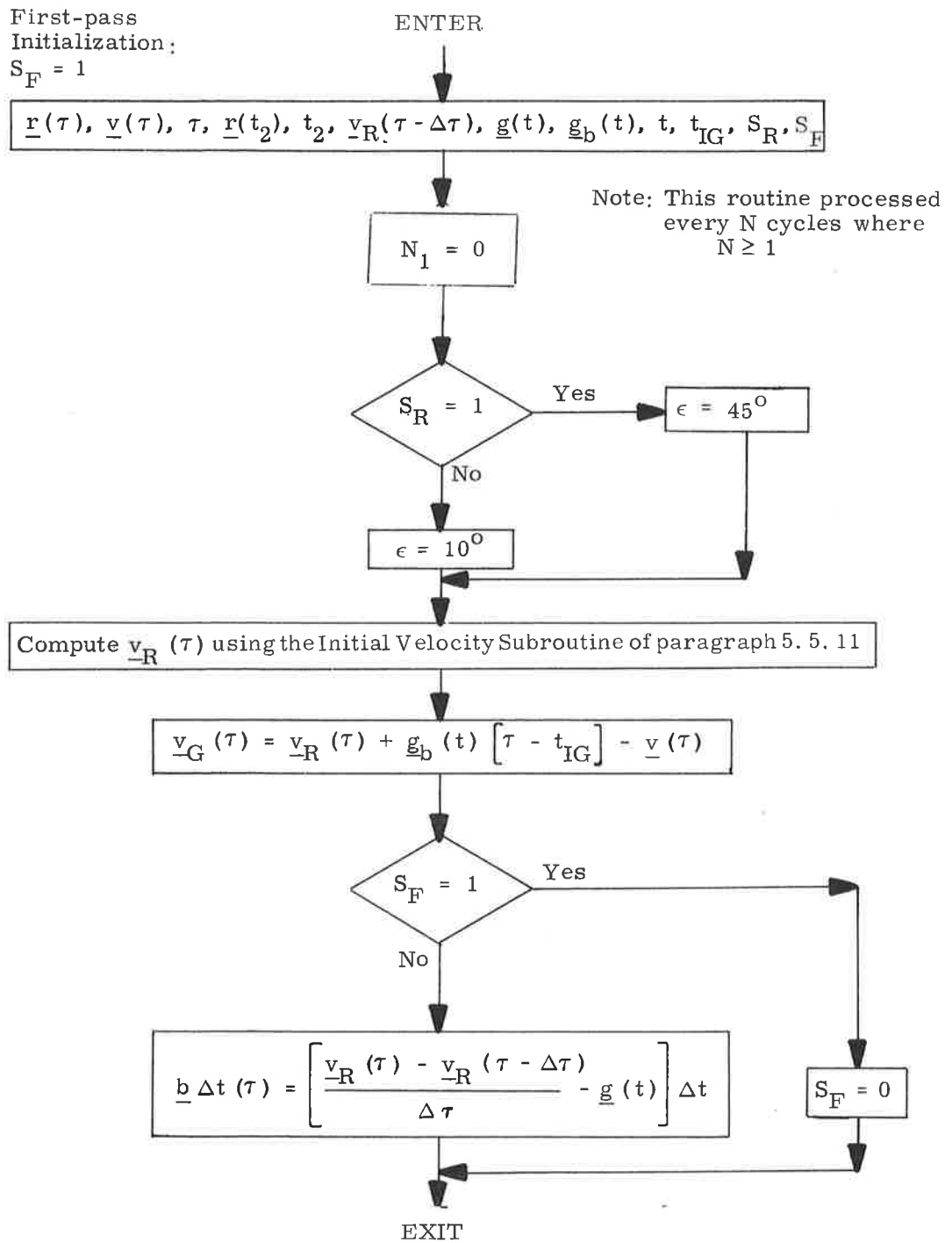


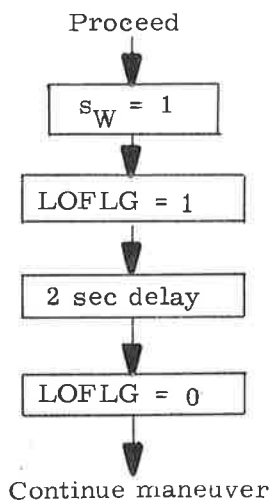
Figure 3.3-13. Velocity-to-be-Gained Subroutine for Lambert Aimpoint Guidance

It should be noted that in Figure 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 $\underline{g}_b(t) [\tau - t_{IG}]$. 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 ($\tau - t_{IG}$) where t_{IG} is the nominal ignition time. The objective of this correction is to reduce cutoff 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 (P33, P35 and P36), it is assumed that an impulsive maneuver will be applied at ignition time. Since a finite maneuver time is required, the precomputed target aimpoint becomes less accurate as the maneuver progresses. The $\underline{g}_b[\tau - t_{IG}]$ correction is an approximate substitute for a retargeting procedure that cannot be performed during a powered maneuver.

5.3.3.6 SPS Thrust Fail Routine (R40)

The purpose of the SPS Thrust Fail Routine (R40) is to present the astronaut with three options when a low SPS thrust is detected in the Cross Product Steering Subroutine (paragraph 5.3.3.4). The three options are:

1. Command engine off and terminate P40.
2. Command engine off and return to the P40 point where the impulse maneuver test is made.
3. Proceed with the maneuver but with the thrust failure detection inhibited for two sec to prevent premature thrust fail indication as shown below:



See Section 4 for complete details of the R40 routine.

5.3.4 Thrust Monitor Programs

The Thrust Monitor Program, P47, 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 (paragraph 5.2.2). This operation is continued until the state vector is advanced several seconds ahead of the current time as described in paragraph 5.3.3.3.4. The Average-G Routine of paragraph 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 P47 is the measured maneuver velocity change in vehicle coordinates as described in Section 4. The program can also be used for a manually controlled maneuver required after separation from the SIVB.

The Rendezvous Thrust Monitor Program, P48, performs the same function as P47 without performing the subroutine that provides the display of inertial velocity magnitude, altitude rate, and altitude above the launch pad radius, but with the additional capability provided by R27 (paragraph 5.6.6). P48 calls R27 after completion of the current Average-G cycle—provided that the previous R27 computations have been completed. If not, the R27 call is skipped until the next Average-G cycle is completed.

5.3.5 Earth Orbit Insertion Monitor Program—P11

5.3.5.1 Introduction

This paragraph describes the operation and implementation of Program P11, the Earth Orbit Insertion Monitor. P11 is initiated by Program P02, Gyro Compassing, when the liftoff discrete is detected or by the astronaut backup Verb 75. The program is actually composed of five subroutines and performs the following functions (in the order of occurrence).

1. Zeroes the CMC clock at liftoff and updates the reference ephemeris time. (Time Subroutine) (See Time Definition, paragraph 5.1.5.5.)
2. Computes the CMC state vector (in Basic Reference Coordinates) at liftoff and starts the Average-G computation (paragraph 5.3.2) using this state vector. (State Subroutine)
3. Finishes remainder of prelaunch torquing.
4. Computes the matrix REFSMMAT that relates the IMU Stable Member orientation to the Basic Reference Coordinate System. (State Subroutine)

5. Periodically computes 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)
6. 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 P11, when activated by Program P02, assumes the following:

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 estimated time from liftoff at which the initial Saturn pitch and roll maneuvers will commence.

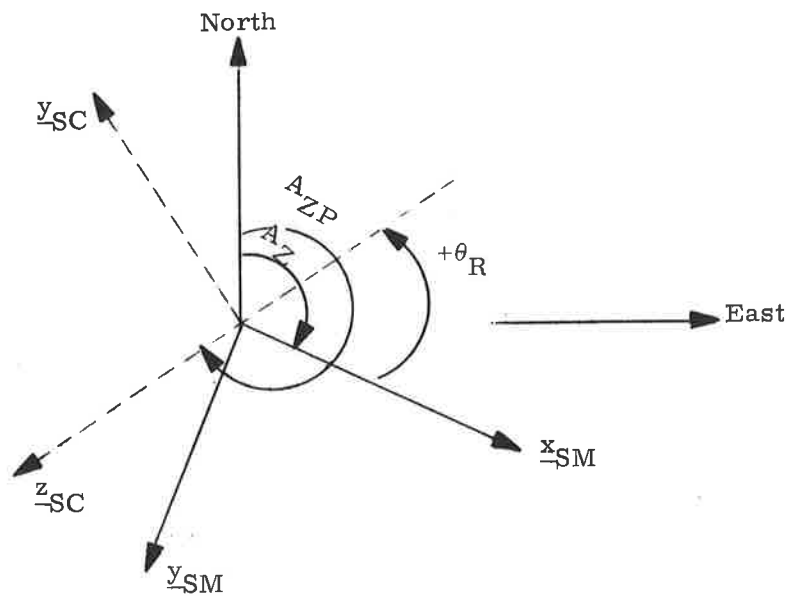
t_{E2} - time interval beginning at t_{E1} used by the Attitude Error subroutine. During t_{E2} , desired gimbal angles computed using polynomial functions.

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 previously synchronized.

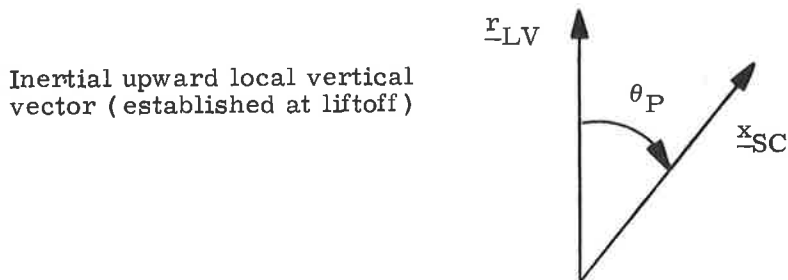
*Flight Director Attitude Indicator

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



- 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 P11

t	Computer clock reading at any time t
t_0	See paragraph 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_P	Launchpad geodetic latitude
Lon_P	Launchpad geodetic longitude
Alt_I	IMU altitude above the launch pad
\underline{u}_Z	A vector in the Basic Reference Coordinate System in the direction of the earth's true pole
ω_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
\underline{CDU}	The current value of the IMU Gimbal Angles
θ_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.)
θ_R	The Saturn nominal roll angle measured from the X IMU Stable Member axis to the negative Z spacecraft axis. (See Figure 3.5-1.)
\underline{E}_a	A vector representing the roll, pitch, and yaw errors about CM body axes (in a right hand sense).
\underline{DGA}	A vector representing the Saturn nominal desired IMU gimbal angles (x, y, z).
\underline{r}_L	Position on the launch pad.
\underline{v}_L	Velocity on the launch pad.

K_z A constant computed in the State Subroutine and used by the Attitude Error Display Subroutine.

Note that pad-loaded variables are identified in paragraph 5.3.5.1.

5.3.5.3 Time Sequencing of P11 Subroutines

Program P11 is composed of four subroutines: Time Subroutine, State Subroutine, Attitude Error Subroutine, and Display Subroutine. The Time Subroutine is selected by P11 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 1/2 second until P11 is exited. Body axis attitude errors are computed from the differences between the desired gimbal angles (DGA) and the actual gimbal angles (CDU). DGA is obtained from a 3-part model of the Saturn SIC attitude profile.

1. Before pitchover, $t \leq t_{E1}$, DGA is the lift off attitude (from the Time Subroutine). Note that the Saturn yaw maneuver for tower clearance is not modeled.
2. During the maneuver, $t_{E1} < t \leq t_{E1} + t_{E2}$, DGA is obtained from a sixth order pitch polynomial and a first order roll polynomial (FDAI Display Subroutine No. 1).
3. When $t > t_{E1} + t_{E2}$ (termination of the pitch maneuver, i. e., tilt arrest) DGA is held constant at its last computed value.

The values of t_{E1} and t_{E2} are chosen so as to provide a reasonably smooth model of expected Saturn performance.

At the same time the attitude error display is started, the cycling of the Display Subroutine is started. This subroutine displays v, h, and 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 Figure 3.5-2. Refer to Time Definitions, paragraph 5.1.5.5, for a description of the reference ephemeris time synchronization with the AGC clock before 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 P27 intercedes.

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 Figure 3.5-3 shows this computation.

REFSMMAT, the matrix that relates the IMU stable member orientation at lift-off to the Basic Reference Coordinates, is determined as shown on page 2 of Figure 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.

The state vector and REFSMMAT computations take place within 0.75 second of the receipt of the lift-off discrete by Program P11.

5.3.5.5a. 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.

5.3.5.6 Display Subroutine

The computations of the display quantities v , h , \dot{h} are shown on Figure 3.5-6. Callable display parameters available during P11 operations are described in paragraph 5.6.10, Orbital Parameter and Entry Display Computations.

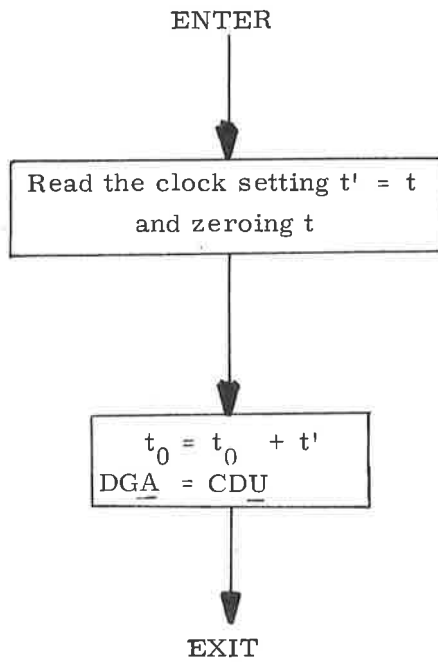


Figure 3.5-2. Time Subroutine

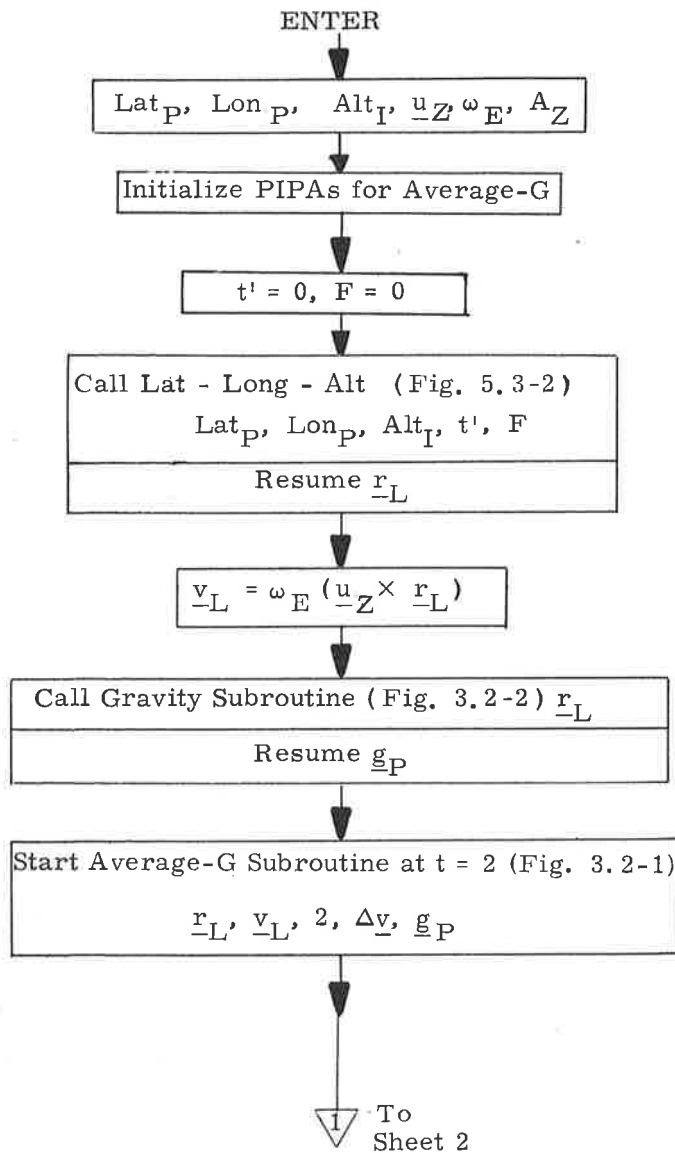


Figure 3.5-3. State Subroutine (Sheet 1 of 2)

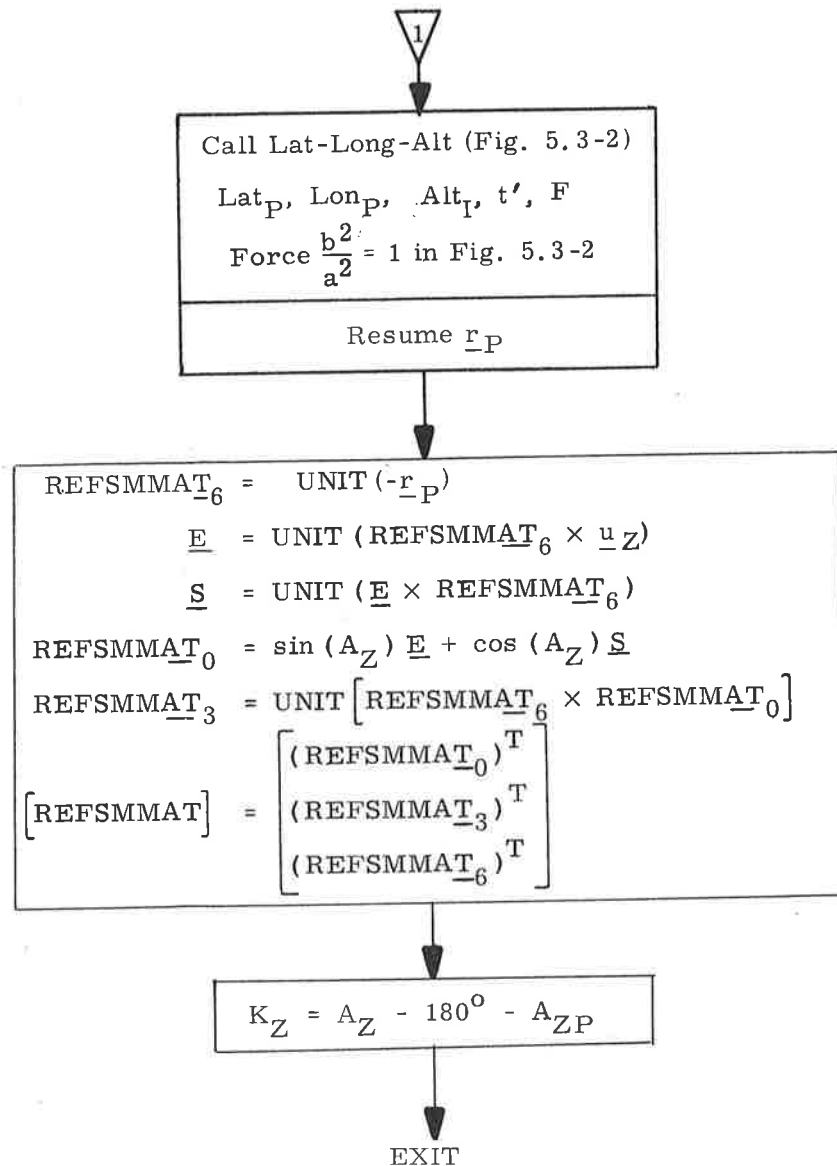


Figure 3.5-3. State Subroutine (Sheet 2 of 2)

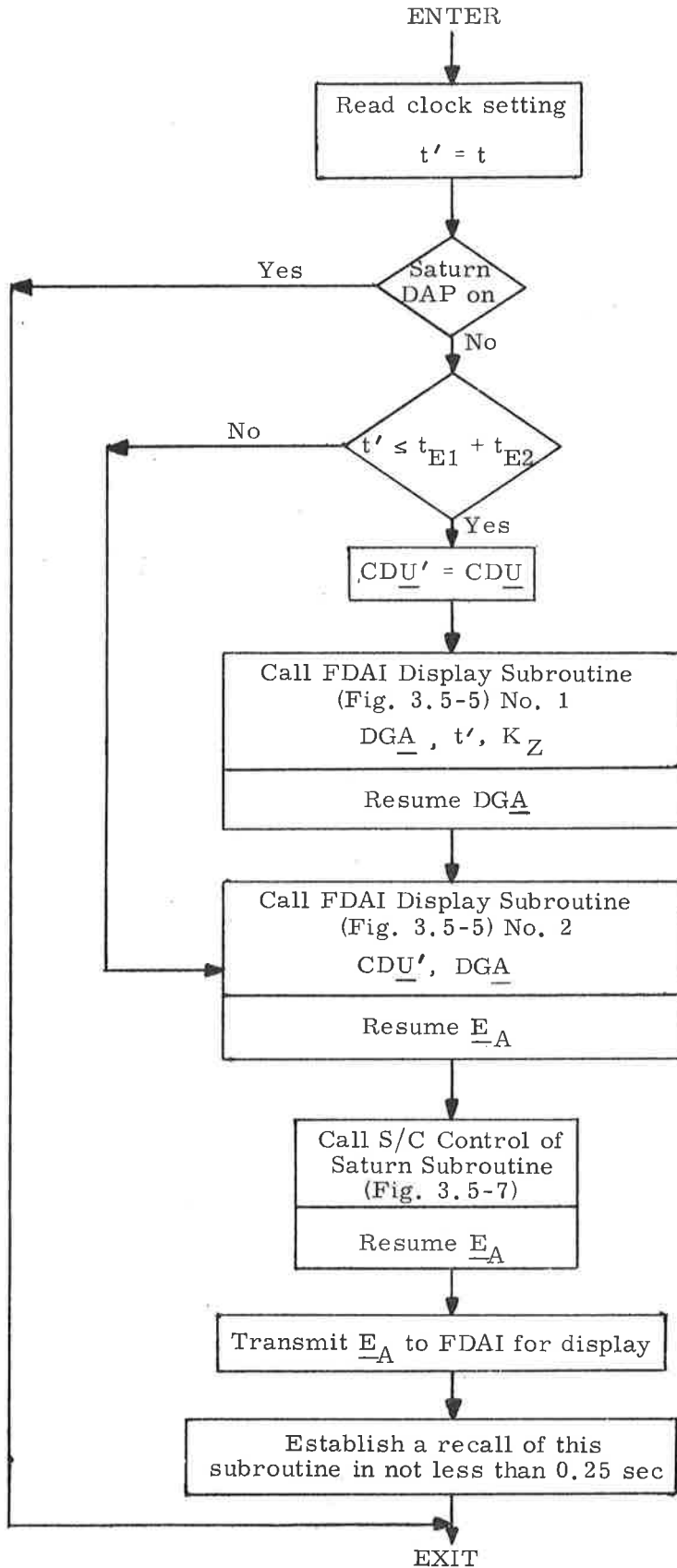


Figure 3.5-4. Attitude Error Subroutine

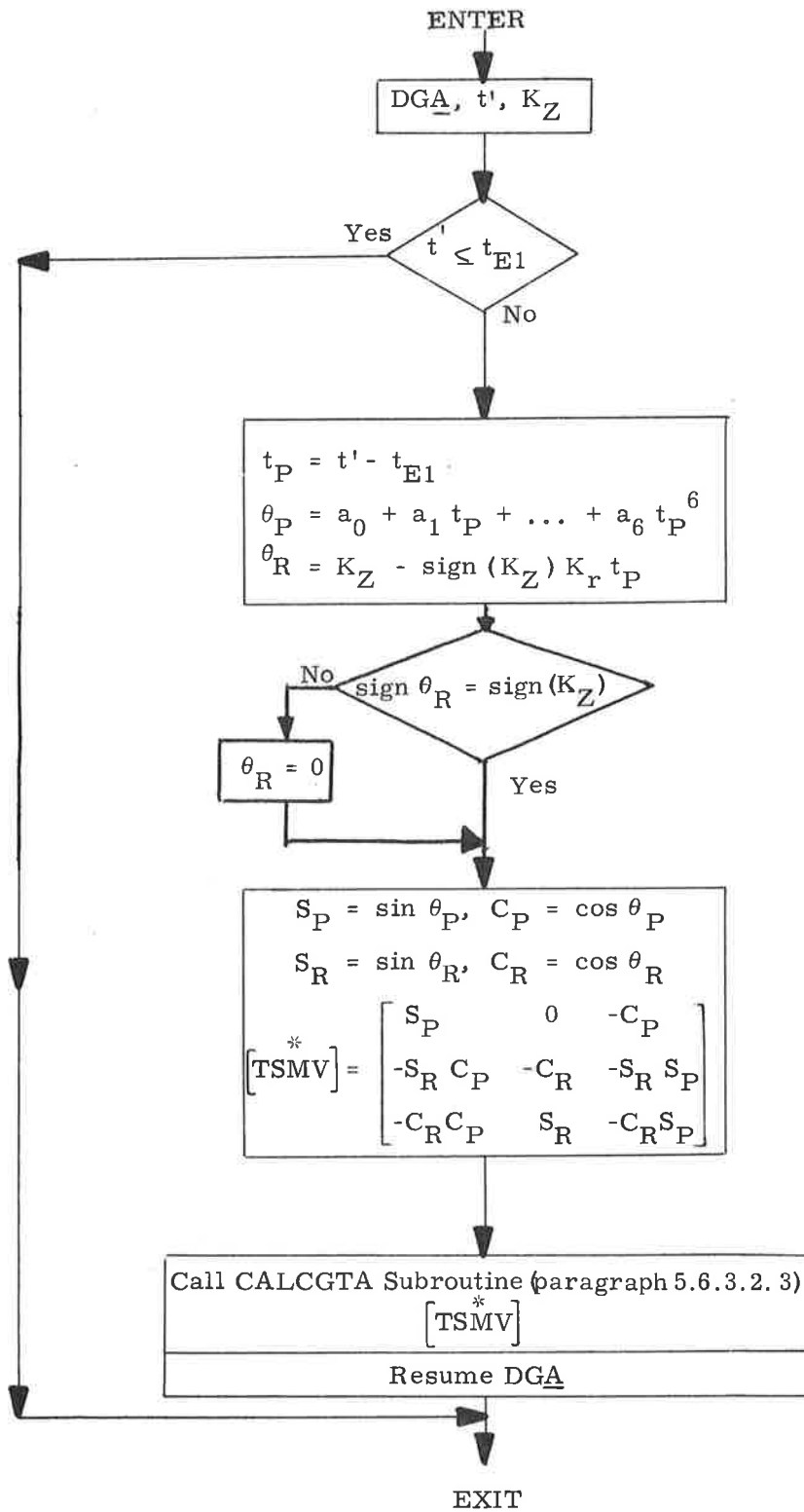


Figure 3.5-5. FDAI Display Subroutine No. 1 (Sheet 1 of 2)

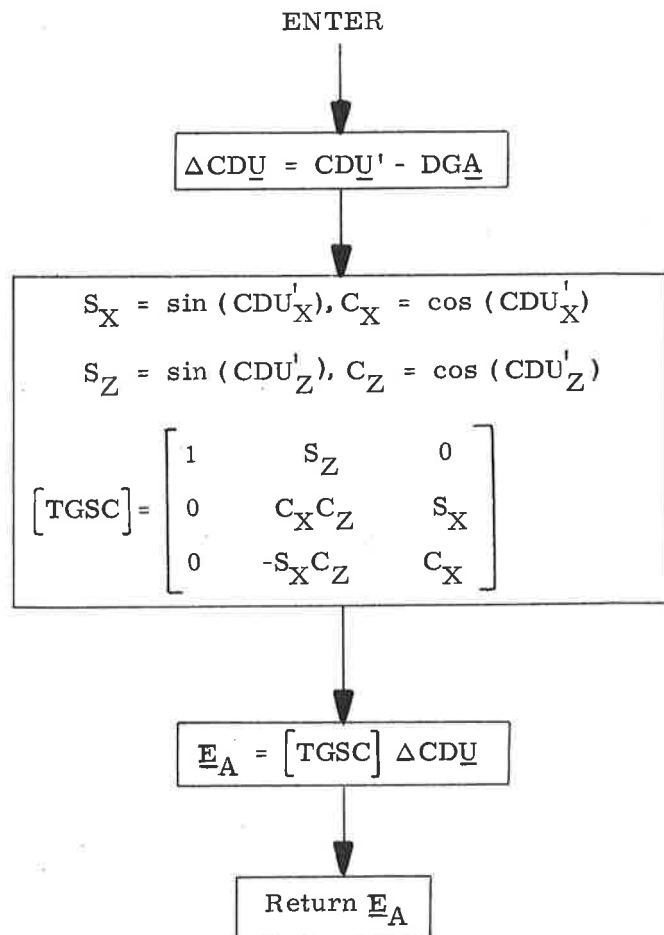


Figure 3.5-5. FDAI Display Subroutine No. 2 (Sheet 2 of 2)

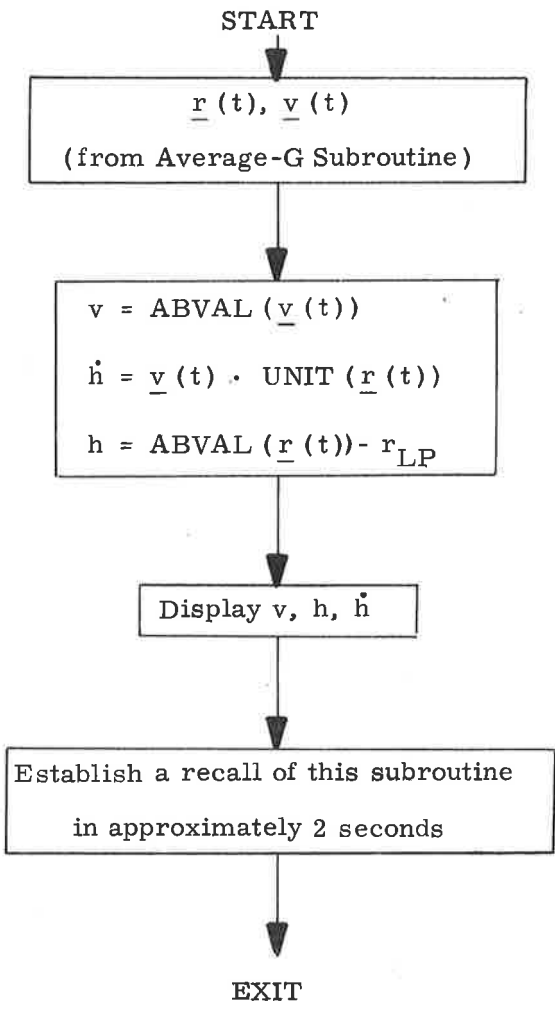


Figure 3.5-6. Display Subroutine

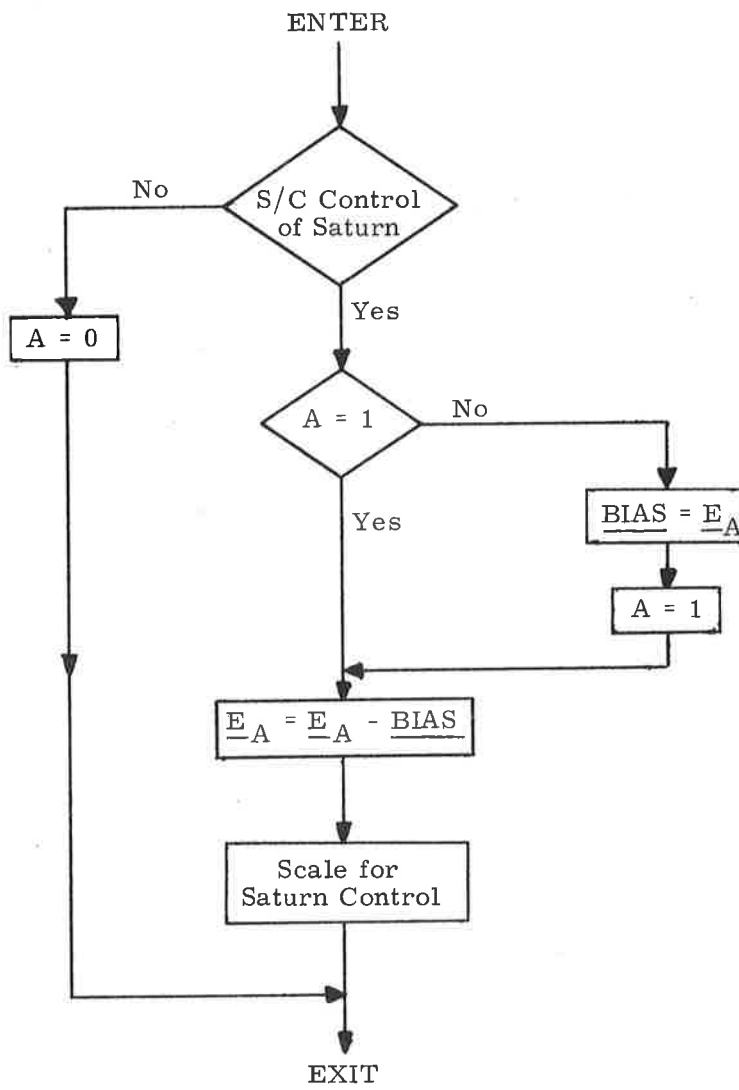


Figure 3.5-7. Spacecraft Control of Saturn Subroutine

5.4 TARGETING

The rendezvous targeting programs presented in this section provide a CMC capability to determine the required input target parameters and to set control modes for the various powered-flight guidance programs described in subsection 5.3. The rendezvous targeting programs use the Basic Reference Coordinate System defined in paragraph 5.1.4.1; they require, as basic input parameters, the vehicle state-vector estimates determined by the navigation programs described in subsection 5.2 and paragraph 5.3.2.

The Skylark rendezvous is implemented through the use of seven targeting programs, and involves the following maneuvers:

NC1	—	Corrective maneuver no. 1
NC2	—	Corrective maneuver no. 2
NCC	—	Corrective combination maneuver
NSR	—	Coelliptic maneuver (slow rate)
TPI	—	Terminal-phase initiation maneuver
TPM	—	Terminal-phase midcourse maneuver
NPC	—	Plane-change maneuver

The terminal phase of rendezvous (TPF) is accomplished by a series of manual braking maneuvers. These maneuvers do not require a targeting program.

The targeting program numbers, corresponding program names, and the special maneuver data each program displays are as follows:

<u>Program Number</u>	<u>Program</u>	<u>Special Maneuver Displays</u>
P31	NC1 Targeting	Δv_{NC2} , Δv_{NCC} , Δv_{NSR}
P32	NC2 Targeting	Δv_{NCC} , Δv_{NSR} , Δv_{NSR}
P33	NCC Targeting	Δv_{NSR}
P34	NSR Targeting	
P35	TPI Targeting	Δv_{TPF}
P36	TPM Targeting	
P38	NPC Targeting	

Programs P31, P32, P34, and P38 use External Δv guidance (paragraph 5.3.3.3.1); programs P33, P35, and P36 use the Lambert targeting concept.

Lambert targeting can be based on conic or precision updates depending on the value chosen for the variable N_1 . If N_1 equals zero, targeting is performed conically. If N_1 is greater than zero, N_1 represents the number of precision offsets used to generate the offset-target vector. In this case, N_1 is generally set equal to two. All seven targeting programs use the Cross Product Steering Routine described in paragraph 5.3.3.4. The function SIGN (x) used in the figures in this subsection is set equal to plus one if x is zero or a positive number; otherwise, it is set equal to minus one.

A display common to all the programs in this subsection is M_{OPT} and M_{VHF} , the number of optics and VHF navigation measurements processed since the last W-matrix reinitialization; MGA, the middle gimbal angle at the start of the maneuver—assuming plus-x acceleration; and TFI, the time from ignition. The data are displayed in R1, R3, and R2 of verb 16 noun 45, respectively.

5.4.1 Rendezvous Targeting Constraints

The nature of the Skylab rendezvous is dictated by constraints on the maneuvers, the maneuver times, and the trajectory. The NC1 and NC2 maneuvers are constrained to be horizontal maneuvers. During the NC1 and NC2 targeting programs the NCC maneuver is also constrained to be a horizontal maneuver. During the NCC program, however, the NCC maneuver is dictated by a time-constrained traverse between two position vectors (Lambert's problem). The TPI maneuver is similarly constrained. The NSR maneuver is constrained to result in coelliptic orbits following the maneuver. Coelliptic orbits have an approximately constant radial distance between the orbits.

Four of the maneuvers have time constraints. The TPI time and the time between the NCC and NSR maneuvers are specified. The times between the NC1 and NC2 maneuvers and the NC2 and NCC maneuvers are indirectly specified by specifying the number of revolutions (may be given by a fraction) involved in the transfers between the maneuvers. The time between the TPI and TPF maneuvers is also indirectly specified via specification of the central angle that the WS traverses between TPI and TPF.

The rendezvous also has three trajectory constraints—two altitude and an elevation-angle constraint. The altitudes between the WS orbit and the CM are specified for the NCC and the TPI times. Because the NCC altitude is specified only during the NC1 program, however, this constraint is not satisfied in the final trajectory if a navigation update occurs between NC1

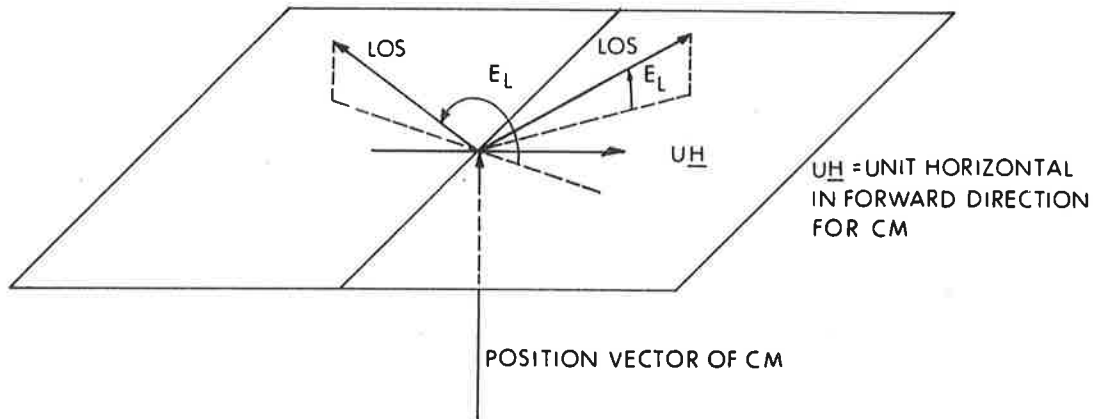
and NC2. The elevation angle specified is that angle(E_L) measured between the CM local horizontal plane and the CM line of sight to the workshop at TPI time. In addition to the three trajectory constraints, rendezvous also requires specification of the workshop central angle between TPI and TPF. Refer to Figure 4.1-1 for a definition of the elevation angle, and to Figure 4.1-2 for an illustration of the rendezvous constraints.

5.4.2 NC1 Targeting Program (P31)

The NC1 Program (P31) computes the first phasing maneuver, NC1, and the NC2, NCC, and NSR maneuver magnitudes for display. The required inputs to this program are:

1. CM and WS state vectors, r_C, v_C, r_W, v_W
2. Time of the NC1 maneuver, t_C
3. Time between the NCC and NSR maneuvers, Δt
4. Time of TPI maneuver, t_F
5. Altitude between the WS orbit and the CM at TPI time, Δh_F
6. Elevation angle, E_L
7. NC1-NC2 mode switch, $s = 1$
8. Number of revolutions between the NC2 and NCC maneuvers, n_{H1}
9. Altitude between the WS orbit and the CM at NCC time, Δh
10. Number of revolutions between the NC1 and NC2 maneuvers, n_C .

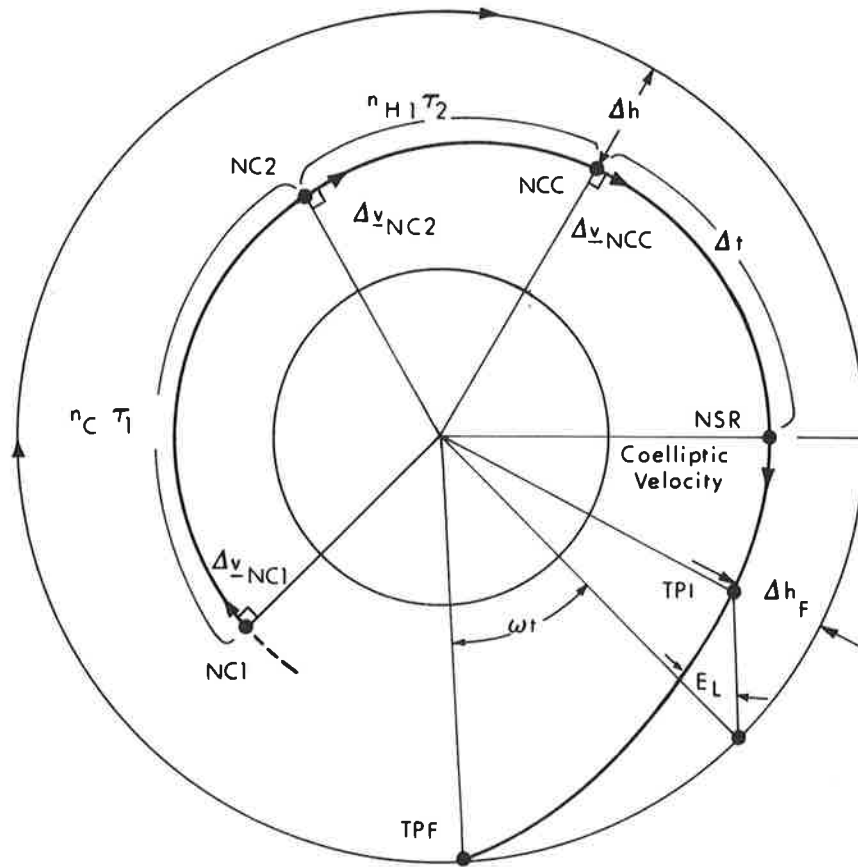
The NC1 program logic is illustrated in the flowchart in Figure 4.2-1. The program uses two independent iterative loops, embedded in an outer iterative loop, to compute a trajectory that satisfies the Skylab constraints. The two inner loops use the magnitudes of the NC2 and NCC maneuvers as independent variables, while the outer loop uses the magnitude of the NC1 maneuver. Nominal values for these maneuvers are stored in the program for use as initial estimates. Because of the nature of the constraints being satisfied by the iterative loops, the NC2 and NCC maneuvers are considered height maneuvers; the NC1 maneuver is considered a phase maneuver.



1. If the LOS projection on \underline{UH} is positive:
 - a. When the LOS is above the horizontal plane,
 $0 < E_L < \pi/2$
 - b. When the LOS is below the horizontal plane,
 $3\pi/2 < E_L < 2\pi$
2. If the LOS projection on \underline{UH} is negative:
 - a. When the LOS is above the horizontal plane,
 $\pi/2 < E_L < \pi$
 - b. When the LOS is below the horizontal plane,
 $\pi < E_L < 3\pi/2$

Figure 4.1-1. Definition of Elevation Angle E_L

Orbit of WS Vehicle



τ_1 = Period of post NC1 maneuver orbit
 τ_2 = Period of post NC2 maneuver orbit

Figure 4.1-2. Illustration of the Skylark Rendezvous Constraints

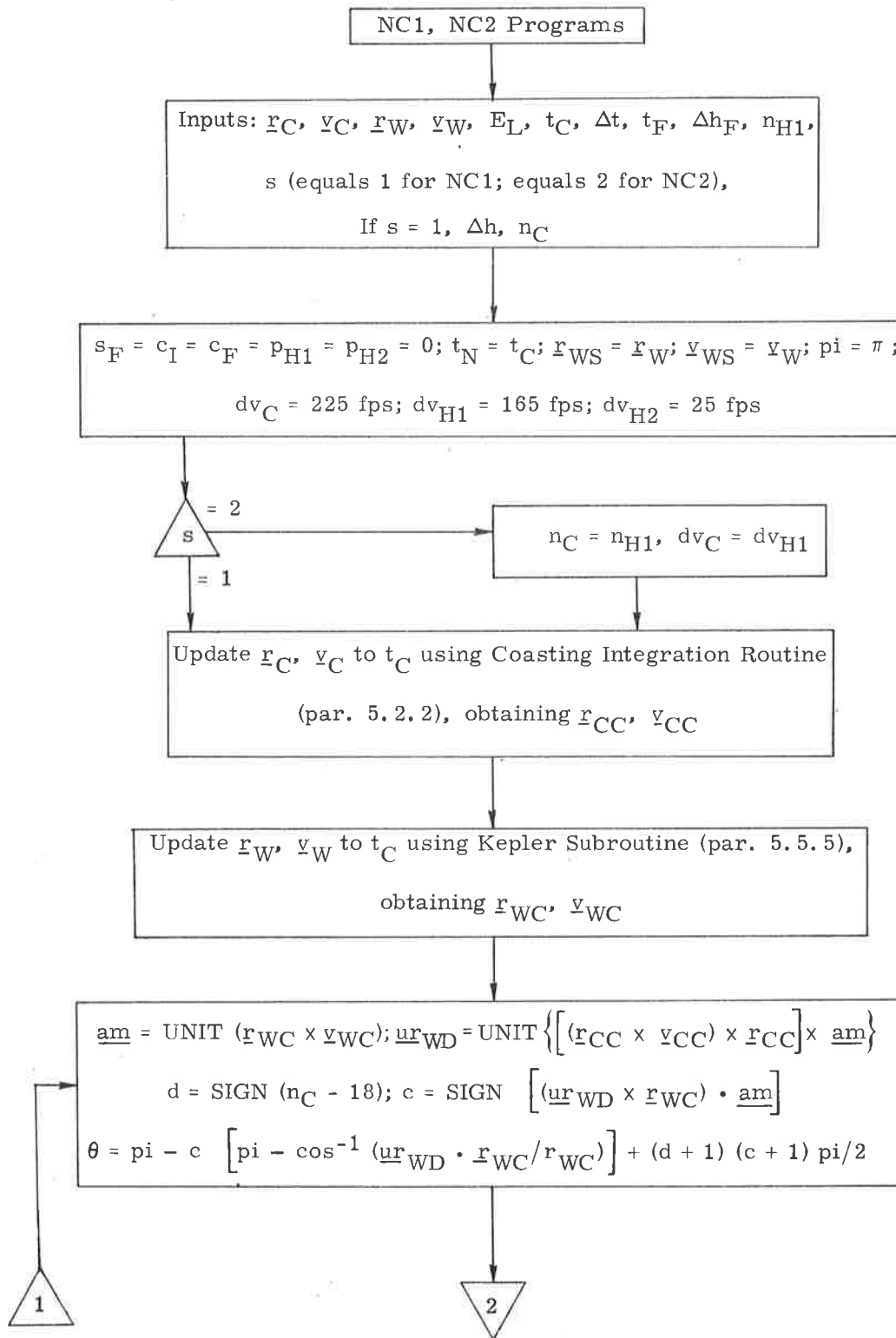


Figure 4.2-1. NC1, NC2 Targeting Program Flowchart (Sheet 1 of 7)

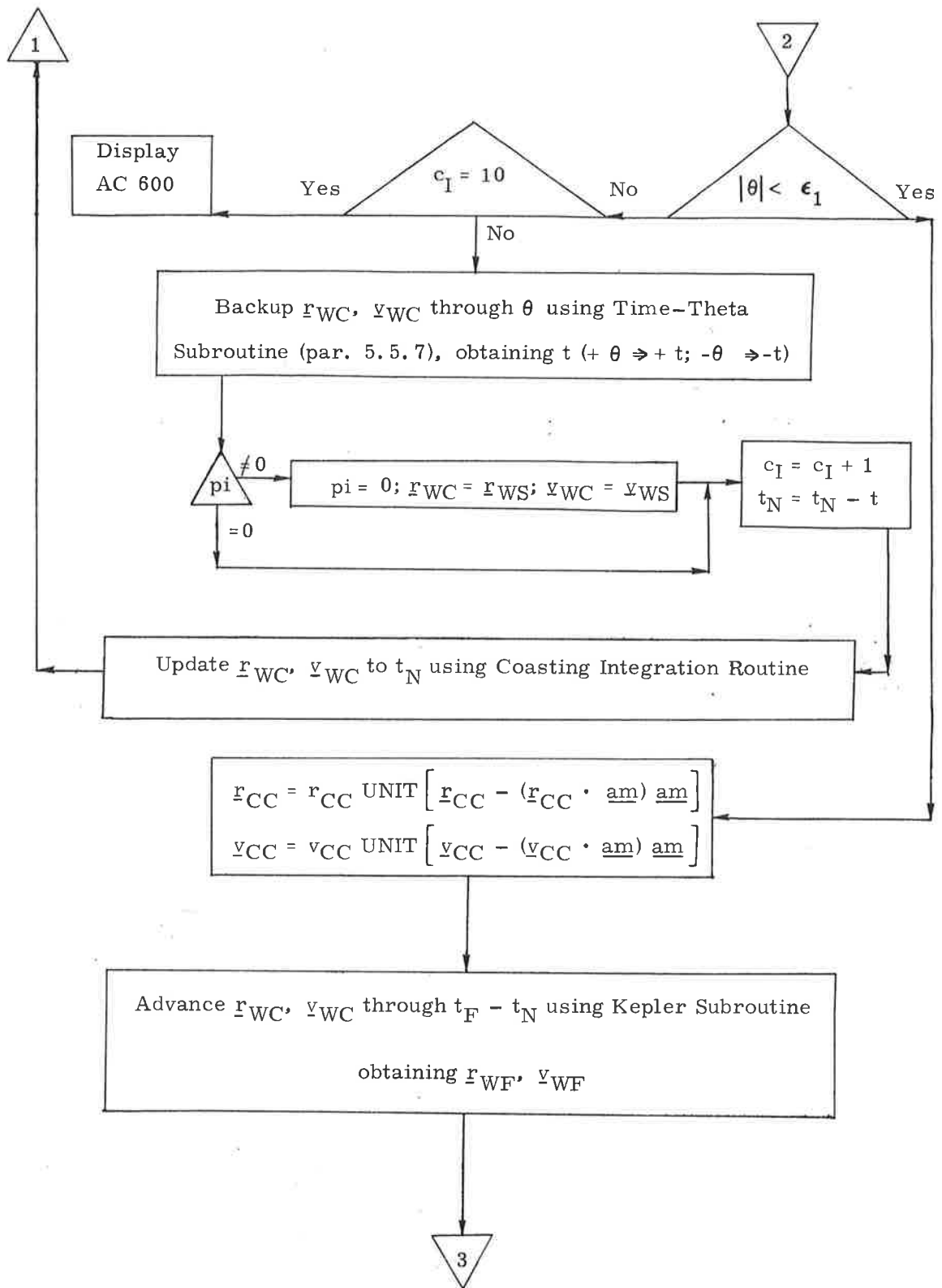


Figure 4.2-1. NC1, NC2 Targeting Program Flowchart (Sheet 2 of 7)

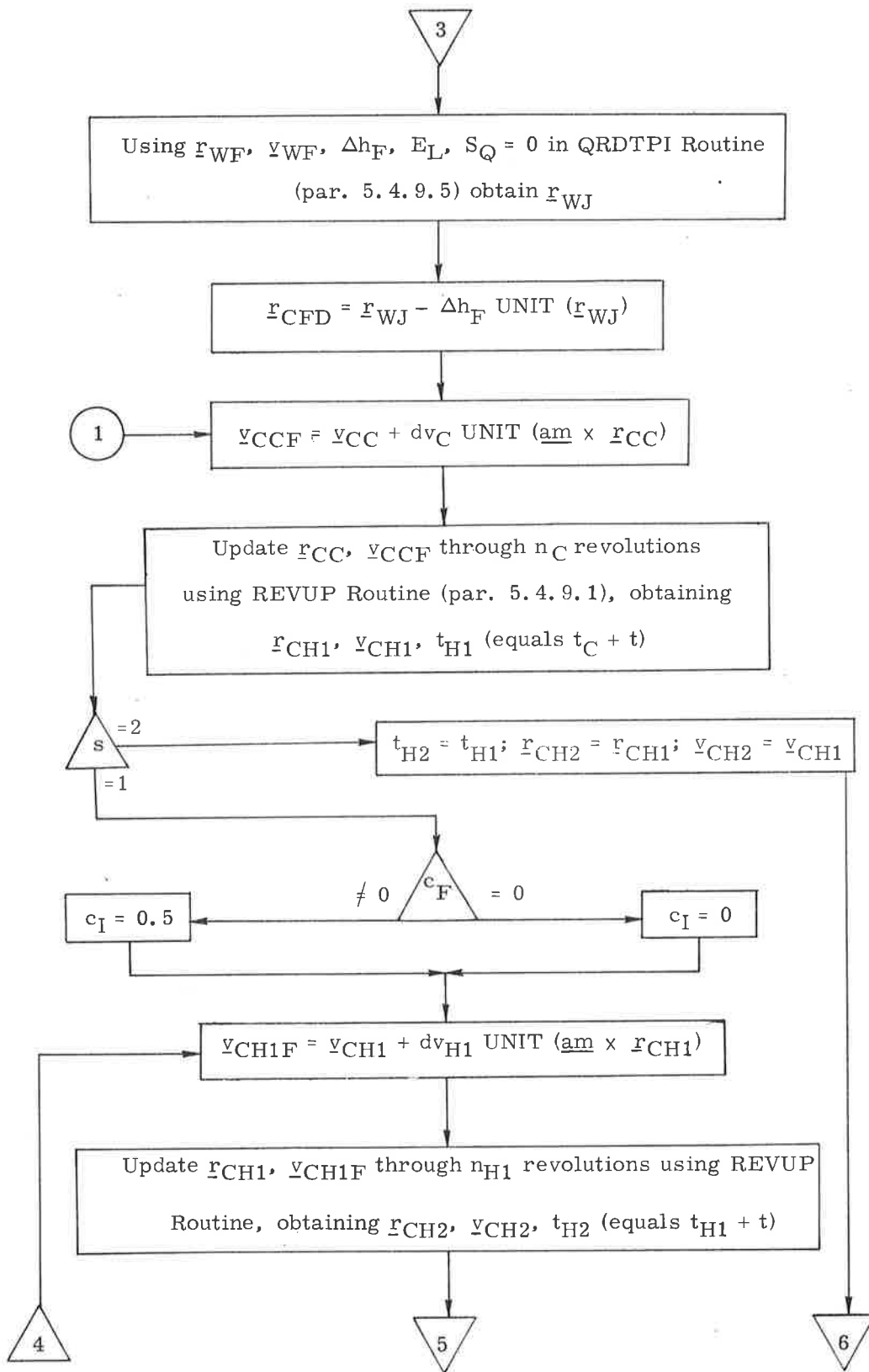


Figure 4.2-1. NC1, NC2 Targeting Program Flowchart (Sheet 3 of 7)

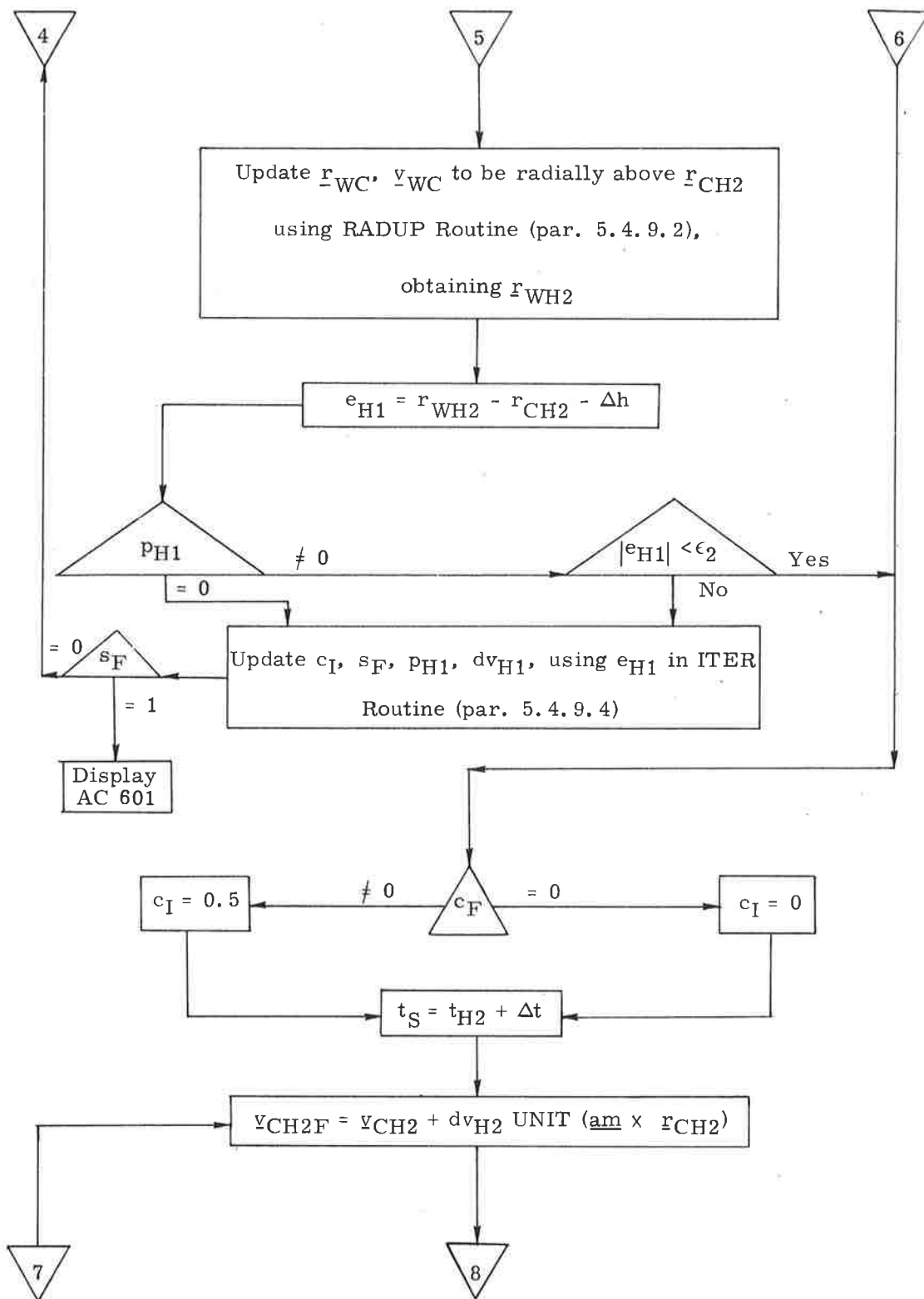


Figure 4.2-1. NC1, NC2 Targeting Program Flowchart (Sheet 4 of 7)

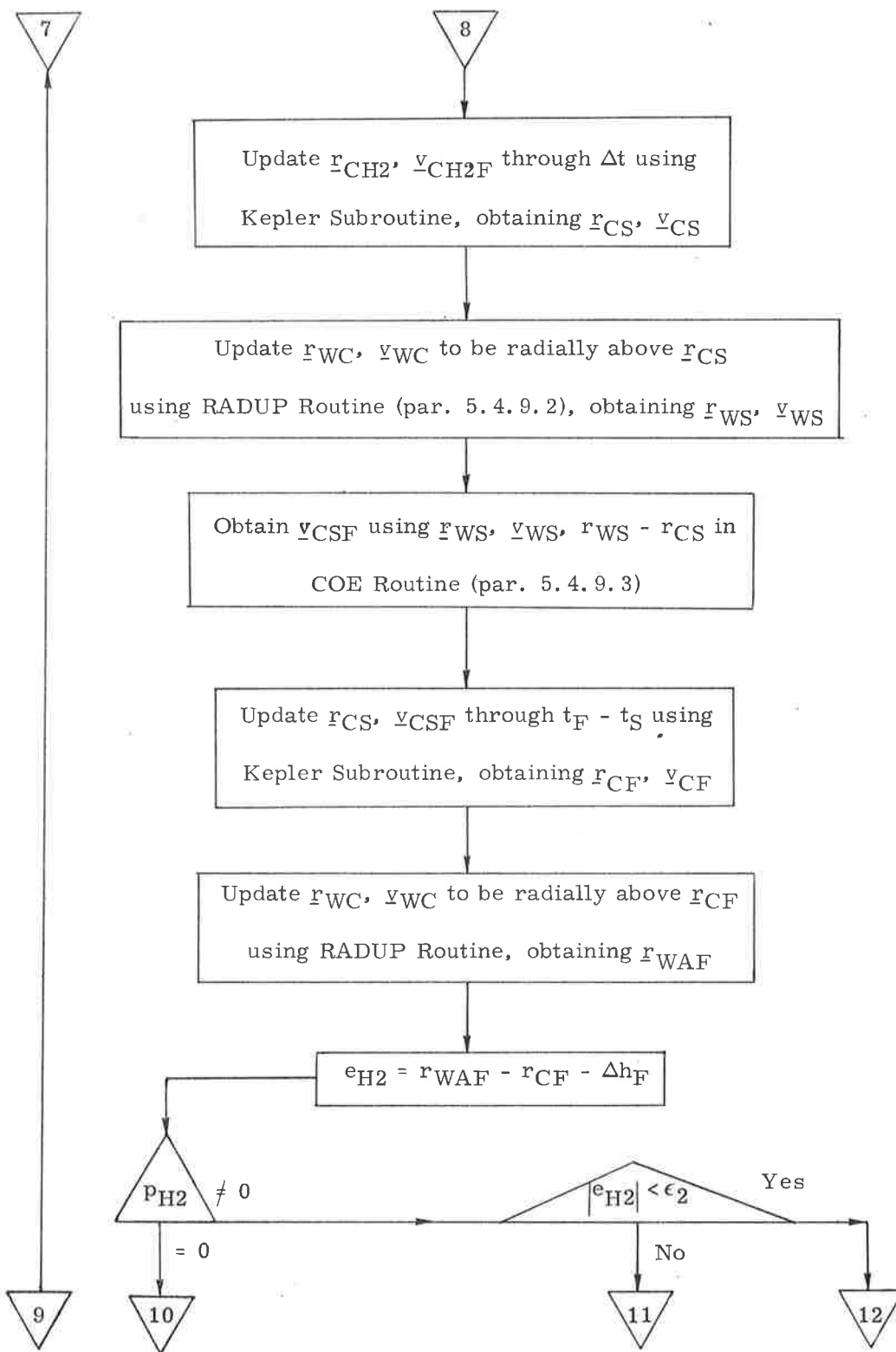


Figure 4.2-1. NC1, NC2 Targeting Program Flowchart (Sheet 5 of 7)

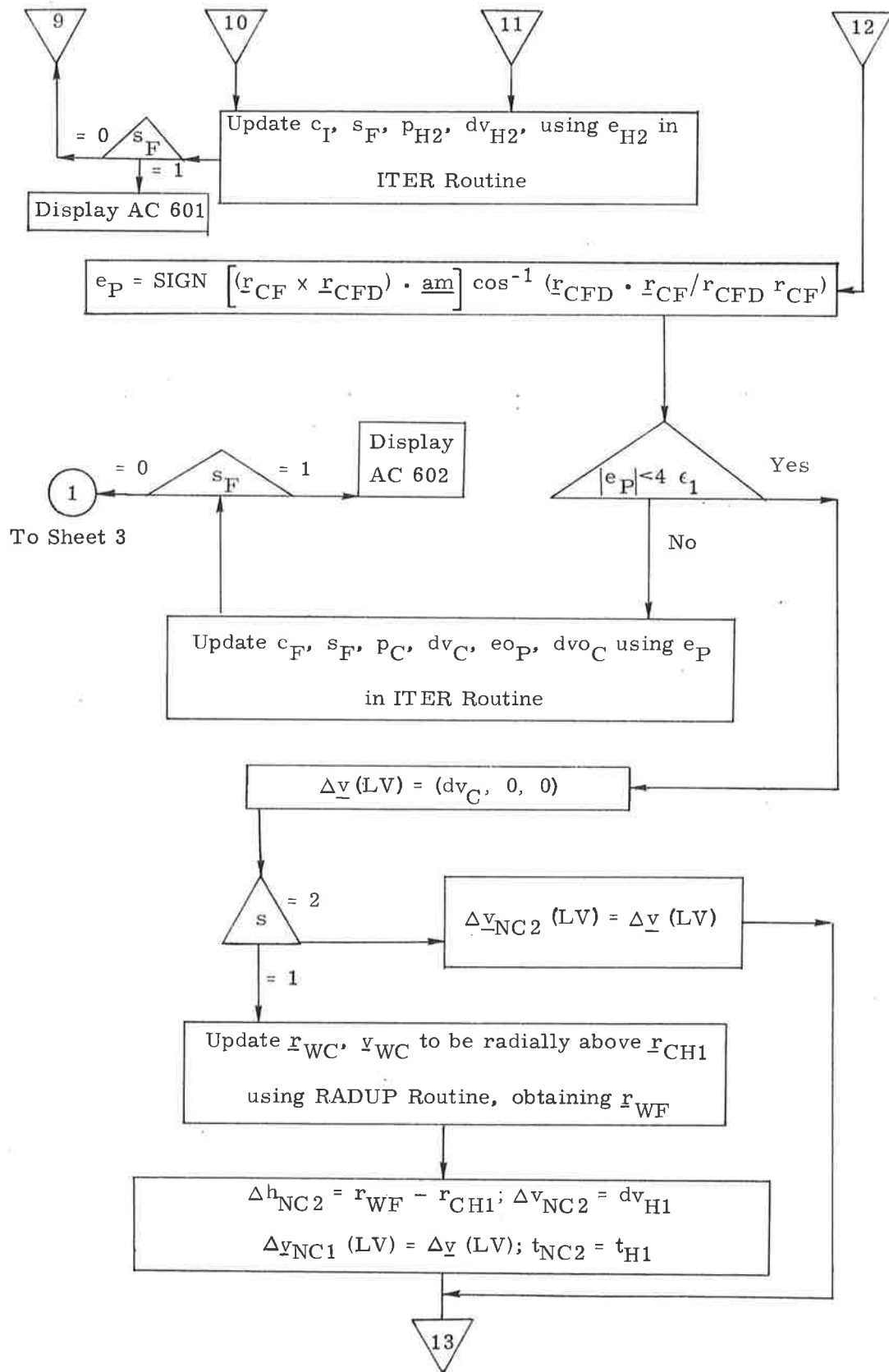


Figure 4.2-1. NC1, NC2 Targeting Program Flowchart (Sheet 6 of 7)

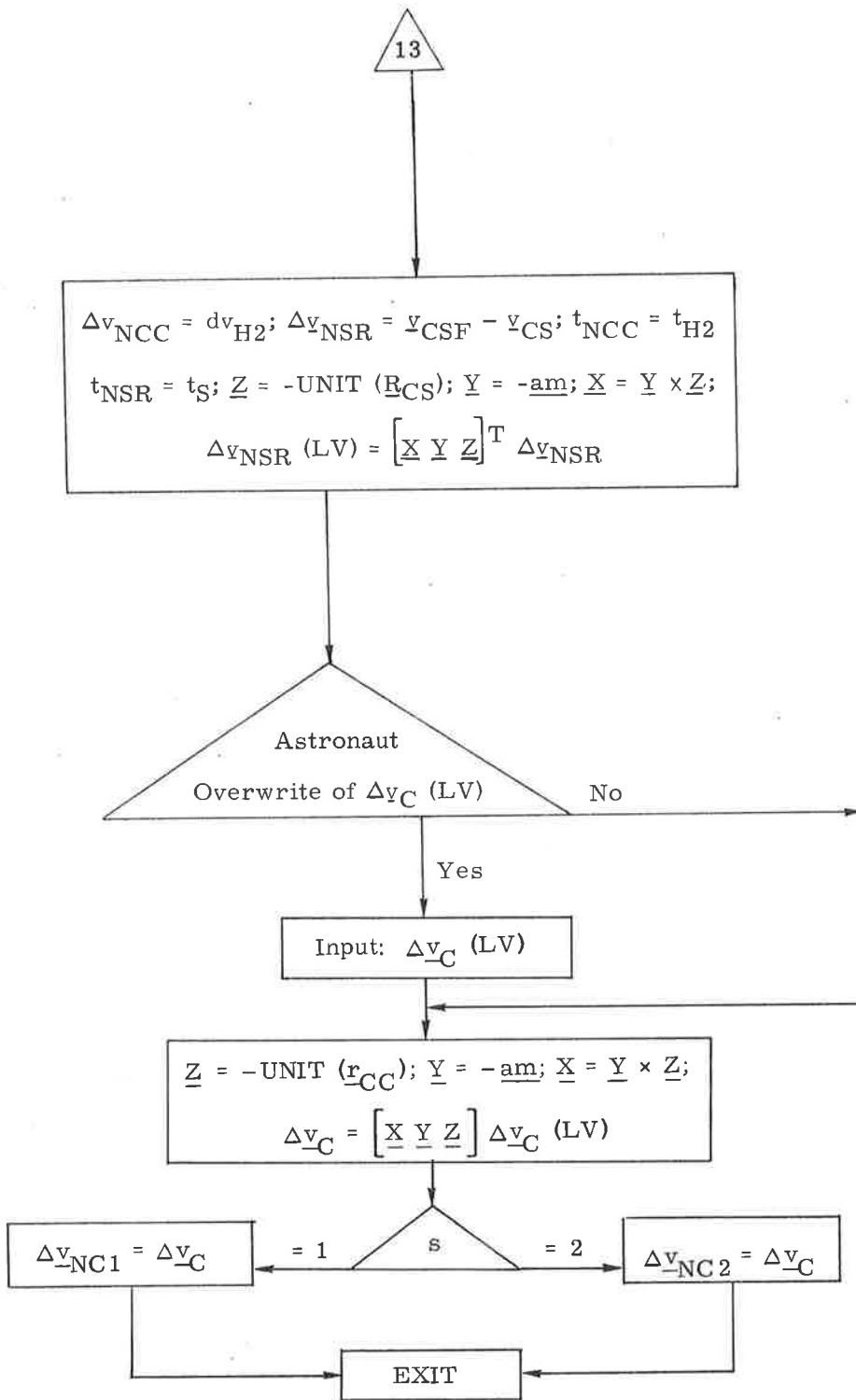


Figure 4.2-1. NC1, NC2 Targeting Program Flowchart (Sheet 7 of 7)

Before the program enters the iterative loops, the WS state vector is "phase matched" to the CM state vector at the time of NC1. This results in the CM and WS traversing approximately the same central angle during the NC1 to TPI phase of the rendezvous. Since the relative motion of two vehicles that traverse the same central angle in approximately the same orbits can be predicted based on either precision or conic updates, the NC1 program can exclusively employ conic updates without losing significant accuracy (compared with a program using precision updates).

The phase-match procedure starts by updating the CM and WS state vectors to the NC1 time with precision and conic updates, respectively. A unit vector u_{WD} is then calculated, which is colinear with the intersection of the WS plane and the plane perpendicular to the CM plane that contains the CM position vector. The angle θ between the two vehicles is calculated next; it is defined as a positive angle for the WS leading the CM. After θ is converted to a corresponding time, the WS state vector is precision updated to the estimated phase match time t_N . The updated position is then checked to determine if it is approximately colinear with u_{WD} . If not, the calculation is repeated starting with the calculation of θ . Alarm code 600 is displayed if this iteration fails.

When the number n_C of revolutions between the first two maneuvers exceeds 18, there is a possibility that the WS might lead the CM by a central angle greater than 360 degrees. To allow for this possibility, the equation for θ includes a term that will increase θ by 360 degrees when n_C exceeds 18 and the passive vehicle leads the active vehicle by less than 180 degrees.

When phase match is successfully completed, the CM state vector is rotated into the plane of the WS orbit. The WS state vector is then updated to the TPI time. The desired CM position vector r_{CFD} at TPI is established using the QRDTPI routine (paragraph 5.4.9.5) to determine the WS position vector r_{WJ} radially above r_{CFD} . This routine uses the elevation angle E_L and the altitude Δh_F between the WS orbit and the CM, at the TPI time, in an iterative search that solves for the TPI geometry.

The outer loop is initiated by adding the NC1 maneuver to the CM NC1 velocity vector. Using the REVUP Routine (paragraph 5.4.9.1), the new state vector is updated through n_C revolutions to the NC2 point.

Before the first height iterative loop is entered, the iteration counter switch c_I is set equal to zero if it is the first pass or to 0.5 if it is a subsequent pass through the outer loop. This height loop starts by adding the estimated NC2 maneuver to the CM velocity vector. The CM state vector is then advanced to the NCC time using n_{H1} in the REVUP Routine. To verify that the height constraint Δh is satisfied, the WS vehicle must first be updated to be radially above the CM. This is accomplished using the RADUP Routine (paragraph 5.4.9.2), which computes the central angle between the two vehicles and updates the WS state vector through that angle. An altitude error e_{H1} is then defined and checked to determine if it is sufficiently small. This test is bypassed if the partial p_{H1} used in the iterative loop is zero so that a value for p_{H1} can be computed for use in subsequent passes through the NC2 maneuver computation. If e_{H1} passes the test, the program proceeds to the second height maneuver. Otherwise, the ITER Routine (paragraph 5.4.9.4) is used to obtain a new estimate of the NC2 maneuver, and the first height maneuver calculation is repeated. ITER encompasses the Newton-Raphson iterative scheme based on numerical partials.

Following convergence of the first height maneuver (NC2), the calculation of the second height maneuver (NCC) starts by setting the iteration counter c_I to zero or to 0.5 depending on the value of c_F . The estimated value of the NCC maneuver is then added to the CM velocity vector at NCC. After the CM state vector is updated through the time Δt to obtain the CM state at NSR, the WS state vector is updated to be radially above the CM position vector using the RADUP Routine. The coelliptic velocity for the CM is then computed using the COE Routine (paragraph 5.4.9.3) and the CM state vector is updated to the TPI time t_F . The WS state vector is again updated to be radially above the CM position vector using the RADUP Routine.

Based on the desired altitude difference Δh_F , an altitude error e_{H2} is defined. If the partial p_{H2} used in the iterative loop is not zero, and this error is sufficiently small, the program proceeds to the outer loop. Otherwise, the ITER Routine is called to obtain a new value of the NCC maneuver and updated values of c_I and the partial p_{H2} . The second height maneuver calculation is then repeated until the TPI altitude constraint is satisfied. If either the first or second height maneuver iterations fail, alarm code 601 is displayed.

In addition to a constraint on the altitude at the TPI point, there is a constraint on the phasing. The updated CM position vector r_{CF} should be approximately colinear with r_{CFD} . An angular error e_p is defined, which represents the central angle between r_{CF} and r_{CFD} . If this error is sufficiently small,

the program is terminated after calculating the required displays. Otherwise, the ITER Routine is called to obtain a new value of the NC1 maneuver and to update values of the outer loop iteration counter c_F , the partial p_C , and the stored values of the error e_{oP} , and maneuver magnitude dvo_C . If this iterative loop fails, alarm code 602 is displayed.

The astronaut can change the computed NC1 maneuver by overwriting the displayed maneuver velocity vector in local-vertical coordinates. In this case, the astronaut input then becomes the defined velocity-to-be-gained in the External Δv maneuver guidance (paragraph 5.3.3.3.1) used to control the maneuver.

The quantities computed for display are:

- | | | | |
|----|----------------------------------|---|---|
| 1. | $\Delta v_{NC1}(LV)$ | — | NC1 maneuver in local-vertical coordinates |
| 2. | $\Delta v_{NC2}, \Delta v_{NCC}$ | — | Magnitudes of the NC2 and NCC maneuvers |
| 3. | $\Delta v_{NSR}(LV)$ | — | NSR maneuver in local vertical coordinates |
| 4. | $t_{NC2}, t_{NCC}, t_{NSR}$ | — | Times of the NC2, NCC and NSR maneuvers |
| 5. | Δh_{NC2} | — | Altitude difference between the WS orbit and the CM at the time of the NC2 maneuver |

5.4.3 NC2 Targeting Program (P32)

The NC2 program (P32), in addition to computing the second phasing maneuver, NC2, also computes and displays NCC and NSR maneuver magnitudes. The NC2 program logic is incorporated into the NC1 logic as shown in Figure 4.2-1. The required inputs are identical to the first eight inputs of the NC1 program (paragraph 5.4.2) with the exception that t_C represents the time of the NC2 maneuver. The NC1-NC2 mode switch is set equal to two upon selection of this program.

NC2 begins by phase matching the CM and WS state vectors, as in the NC1 program. After rotating the CM state vector into the plane of the WS orbit, the initial estimate of the NC2 maneuver is added to the CM velocity vector.

The CM state vector is then updated through n_{H1} revolutions to obtain the state at NCC. The NCC maneuver is next calculated with the same algorithm used in the NC1 program for the NCC maneuver. Finally, the NC2 maneuver is varied to attain the desired phasing at TPI (analogous to varying the NC1 maneuver in the NC1 targeting program).

The astronaut has the option of overwriting the display of the NC2 maneuver in local-vertical coordinates. The NC2 program utilizes the same alarm codes as the NC1 program.

The quantities computed for display are:

- | | | | |
|----|----------------------------------|---|--|
| 1. | Δv_{NC2} (LV) | — | NC2 maneuver in local-vertical coordinates |
| 2. | $\Delta v_{NCC}, \Delta v_{NSR}$ | — | Magnitudes of the NCC and NSR maneuvers |
| 3. | Δv_{NSR} (LV) | — | NSR maneuver in local-vertical coordinates |
| 4. | t_{NCC}, t_{NSR} | — | Times of the NCC and NSR maneuvers |

5.4.4 NCC Targeting Program (P33)

The objective of the NCC Targeting Program (P33) is the computation of the corrective combination maneuver. In addition, the program computes and displays the NSR maneuver.

The inputs to the NCC program are:

1. CM and WS state vectors, r_C, v_C, r_W, v_W
2. Time of the NCC maneuver, t_C
3. Time between the NCC and NSR maneuver, Δt
4. Time of the TPI maneuver, t_F
5. Altitude between the WS orbit and the CM at the TPI time, Δh
6. Elevation angle, E_L

Figure 4.4-1 is a flowchart of the program; it starts with precision updates of the CM and WS state vectors to t_C and t_F , respectively. The QRDTPI Routine (paragraph 5.4.9.5) is then called to obtain the WS state vector \underline{r}_{WJ} , \underline{v}_{WJ} radially above the desired CM position vector at TPI. Using the WS state vector and the altitude difference Δh , the COE Routine (paragraph 5.4.9.3) is called to obtain the CM state \underline{r}_{CF} , \underline{v}_{CF} prior to the TPI maneuver. This state vector is then integrated backwards, using the Coasting Integration Routine, to the time of the NSR maneuver, obtaining the target vector \underline{r}_{CT} for the NCC maneuver. The Initial Velocity Subroutine (paragraph 5.5.11) is called next to obtain the NCC and NSR velocities based on setting the number of precision offsets N_1 equal to two and the cone angle equal to 15 degrees. The rotation projection switch f_2 and $\underline{r}(t_2)$ are obtained from the Initial Velocity Subroutine for use in the powered-flight steering program as S_R and the offset target vector. After the NCC and NSR maneuvers are calculated, the NCC maneuver is rotated into local-vertical coordinates for display. At the end of P33, the astronaut has the option of overwriting the out-of-plane component of the NCC maneuver. If he does, the program recomputes the aimpoint vector $\underline{r}(t_2)$ by updating the NCC state vector based on the new maneuver.

The quantities computed for display are:

1. $\Delta \underline{v}_{NCC} (LV)$ — NCC maneuver in local-vertical coordinates
2. $\Delta \underline{v}_{NSR} (LV)$ — NSR maneuver in local-vertical coordinates
3. t_{NSR}, t_{TPI} — times of the NSR and TPI maneuvers.

5.4.5 NSR Targeting Program (P34)

The objective of the NSR Targeting Program (P34) is the computation of the coelliptic maneuver. The inputs to this program are:

1. CM and WS state vectors, $\underline{r}_C, \underline{v}_C, \underline{r}_W, \underline{v}_W$
2. Elevation angle, E_L
3. Time of the NSR maneuver, t_2
4. Time of the TPI maneuver, t_3 .

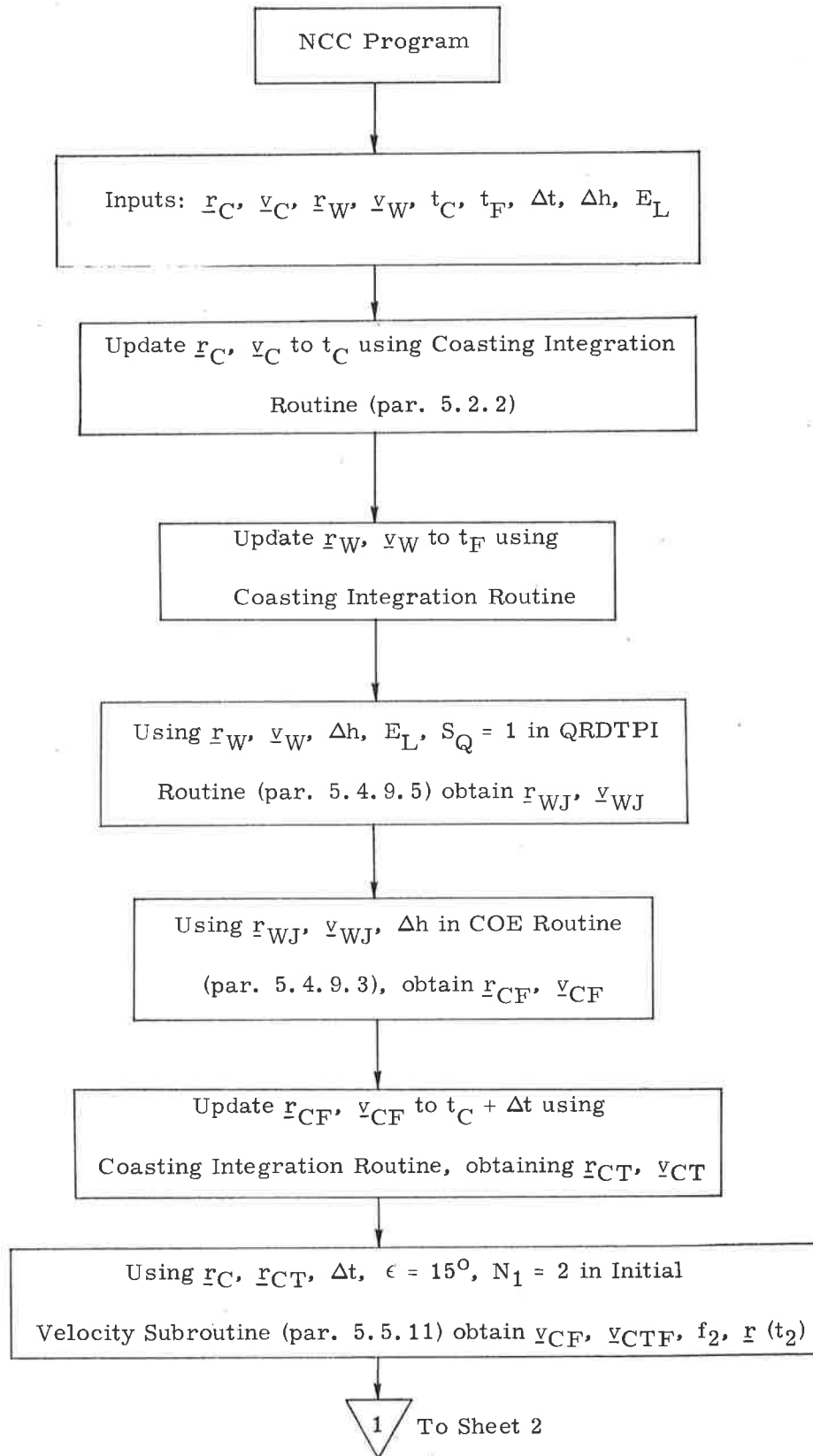


Figure 4.4-1. NCC Targeting Program Flowchart (Sheet 1 of 2)

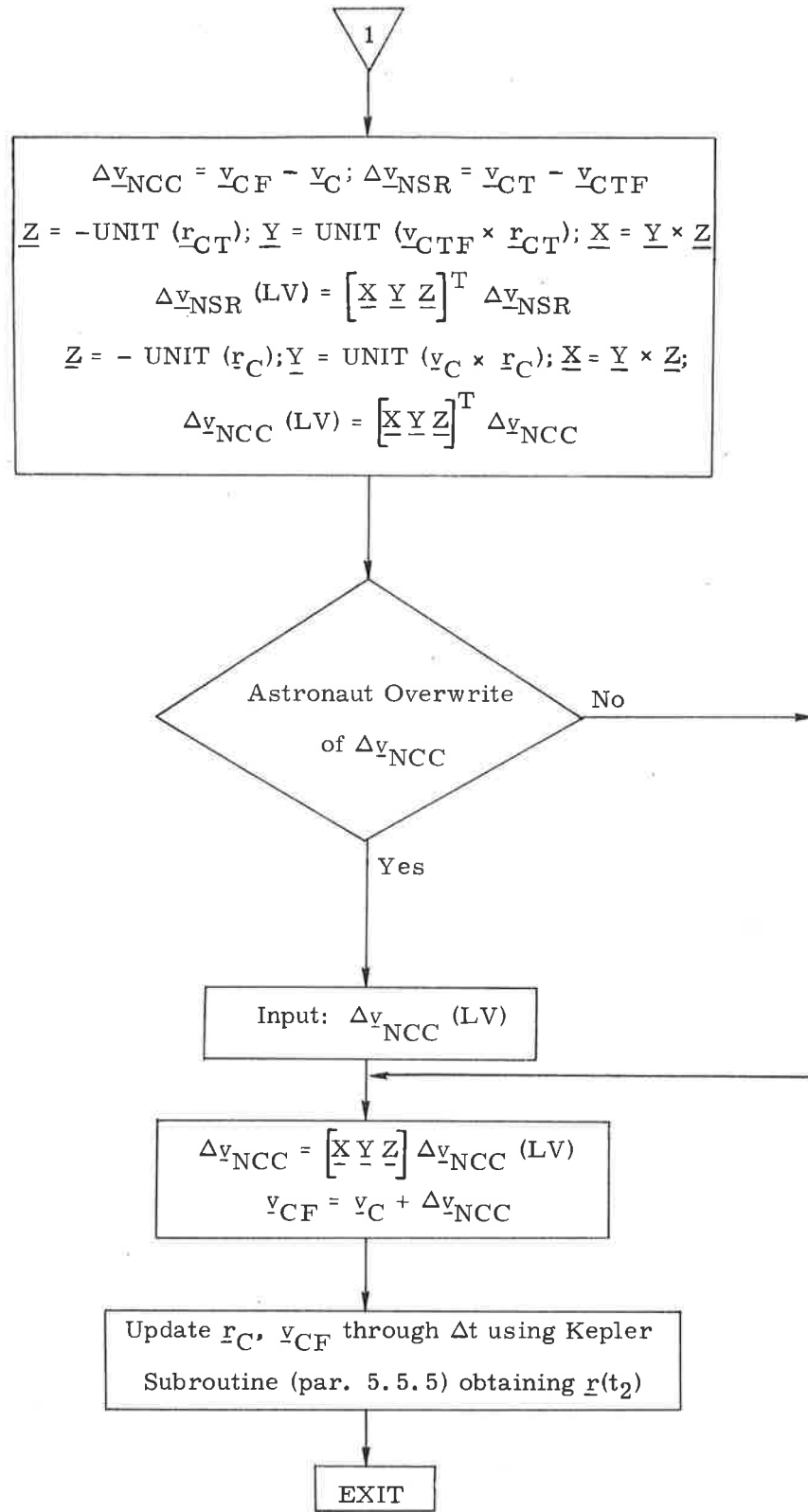


Figure 4.4-1. NCC Targeting Program Flowchart (Sheet 2 of 2)

The NSR program logic is illustrated in Figure 4.5-1. The program starts with a precision update of the CM and WS state vectors to the NSR maneuver time. After the CM state vector is rotated into the plane of the WS orbit, the WS state vector is updated to be colinear with the CM position vector using the Time-Theta Routine (paragraph 5.5.7).

The "coelliptic" maneuver constraint determines the semimajor axis a_C and vertical component of velocity v_{CV} following the maneuver. After calculating the CM post-NSR velocity vector v_{C2}' , both the CM and WS state vectors are conically updated to the time of the TPI maneuver. The TPI program is then used to calculate the time at which the specified elevation angle is attained. If the iteration is not successful, an alarm code is displayed. At this point, the astronaut can either recycle the program or proceed with the calculation of the time displays.

The following quantities are computed for display:

1. Δy_{NSR} (LV) — NSR maneuver in local-vertical coordinates
2. Δt_3 — Time between the computed TPI time and the NSR maneuver time
3. Δt_{TPI} — Time between the computed TPI time and the input TPI time
4. Δh_{NSR} — Altitude difference between the WS orbit and the CM at the time of the NSR maneuver
5. $Y_C, \dot{Y}_C, \dot{Y}_W$ — Out-of-plane parameters at t_{IG} :
 Y_C and \dot{Y}_C — CM out-of-plane position and velocity relative to WS orbital plane
 \dot{Y}_W — WS out-of-plane velocity relative to CM orbital plane

5.4.6 TPI Targeting Program (P35)

The Terminal Phase Initiation (TPI) Targeting Program (P35) computes the terminal phase initiation maneuver. The position of this maneuver is determined by specifying either the TPI time or the elevation angle that specifies the

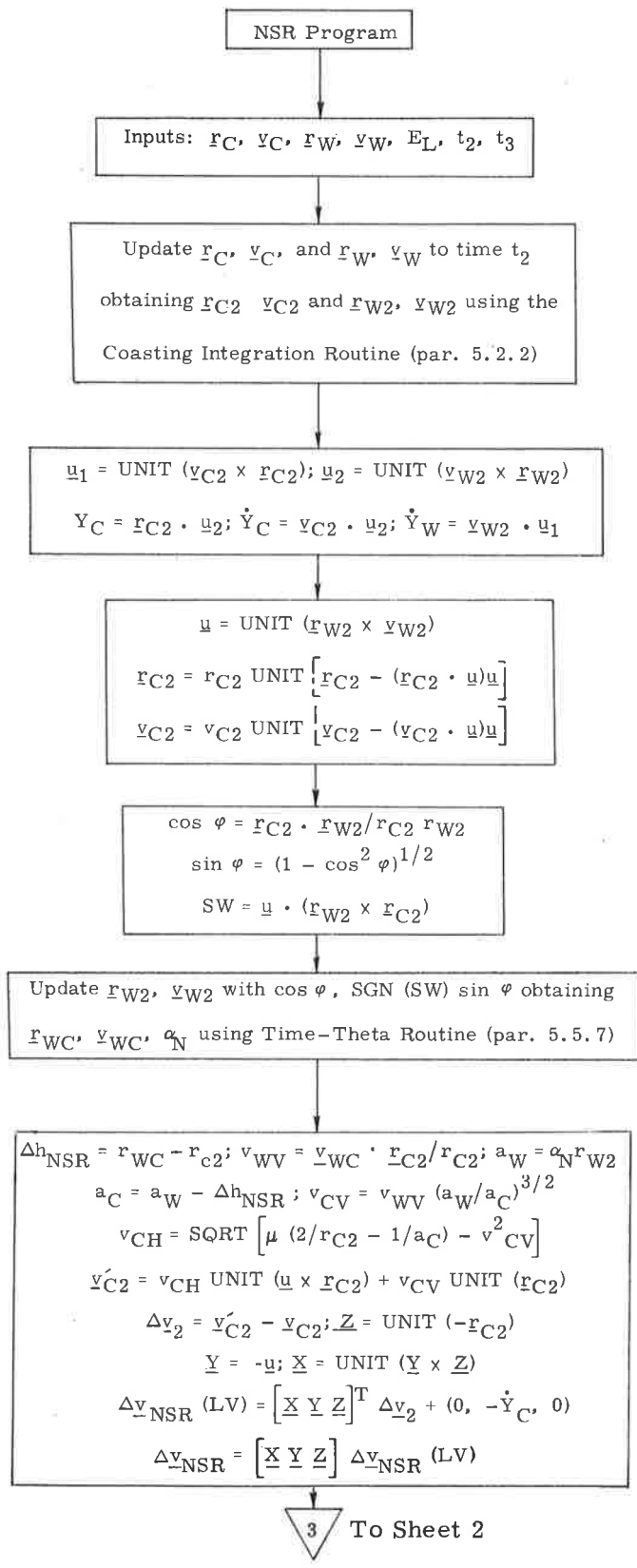


Figure 4.5-1. NSR Targeting Program Flowchart (Sheet 1 of 2)

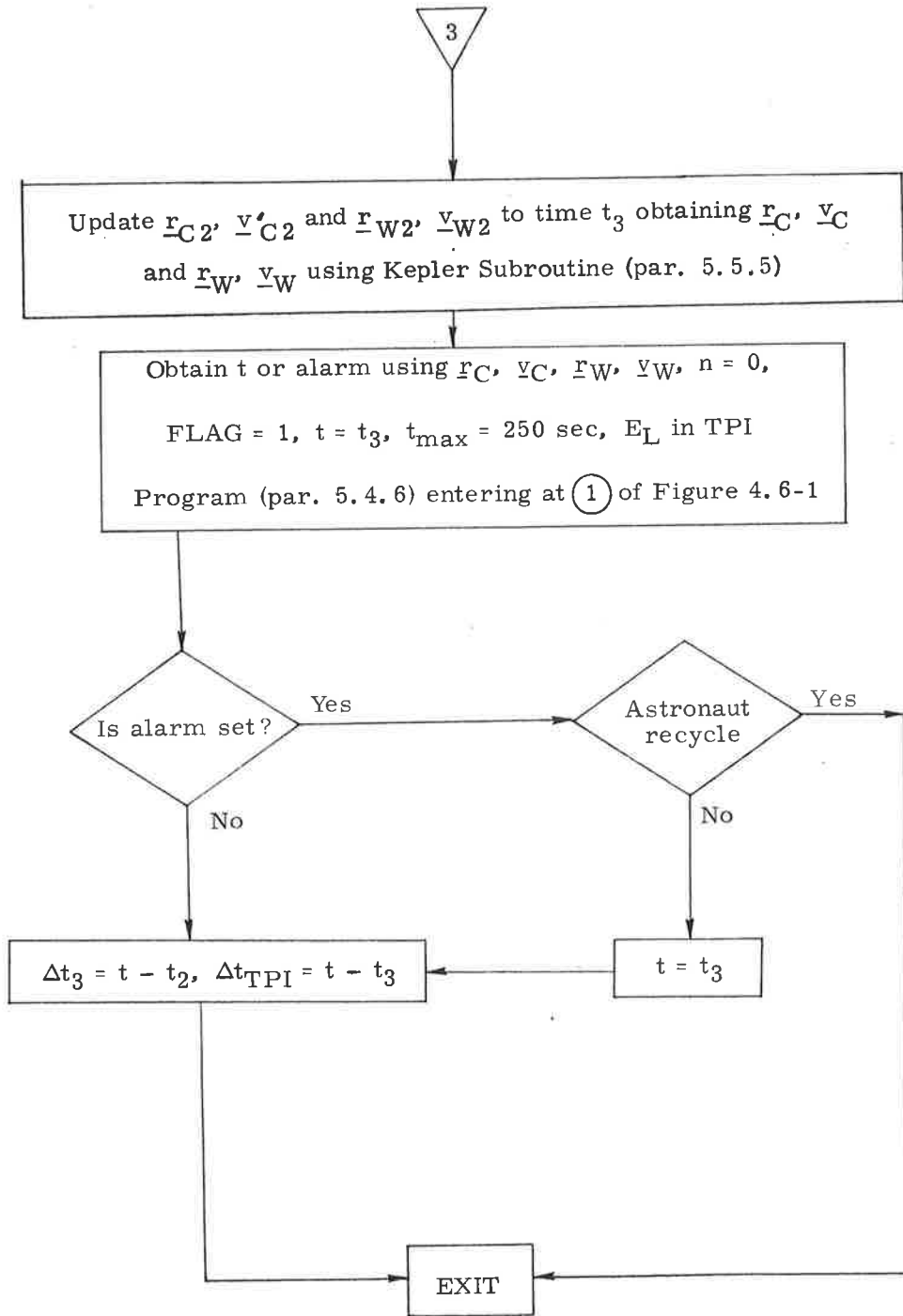


Figure 4.5-1. NSR Targeting Program Flowchart (Sheet 2 of 2)

relative geometry of the vehicles at the TPI point. The inputs to this program are:

1. CM and WS state vectors, x_{CI} , y_{CI} , x_{WI} , y_{WI}
2. Time t of the TPI maneuver
3. Elevation angle E_L (set equal to zero if t is specified)
4. Central angle ωt of the WS between the TPF and TPI points
5. Number N_1 of precision offsets made in generating the offset target vector for TPI maneuver; if set equal to zero, conic trajectories are used to generate the target vector.

The TPI program logic is illustrated in Figure 4.6-1. The program starts with a precision update of the CM and WS state vectors to the TPI time. It then computes the elevation angle—if it has not been specified—and initiates an iteration procedure to determine the TPI time at which the elevation angle is attained. This procedure — based on conic trajectories—uses the input TPI time as an initial estimate. Modifications to the TPI time are based on an empirical relationship using the angular distance between the current position of the WS and the desired position of the WS. This relationship assumes that the WS is in a circular orbit and that both the CM and WS are moving at a constant angular rate.

To help ensure convergence during the iteration procedure, the following steps are taken:

1. The TPI time step size δt is restricted to 250 sec.
2. If the solution has been passed ($\delta E_L \delta E_0 < 0$), the step size is halved and forced in the direction opposite to the last step.
3. If the iteration is converging ($|\delta E_0| - |\delta E_L| > 0$), the sign of δt is maintained.
4. If the iteration is proceeding in the wrong direction, the step direction is reversed.

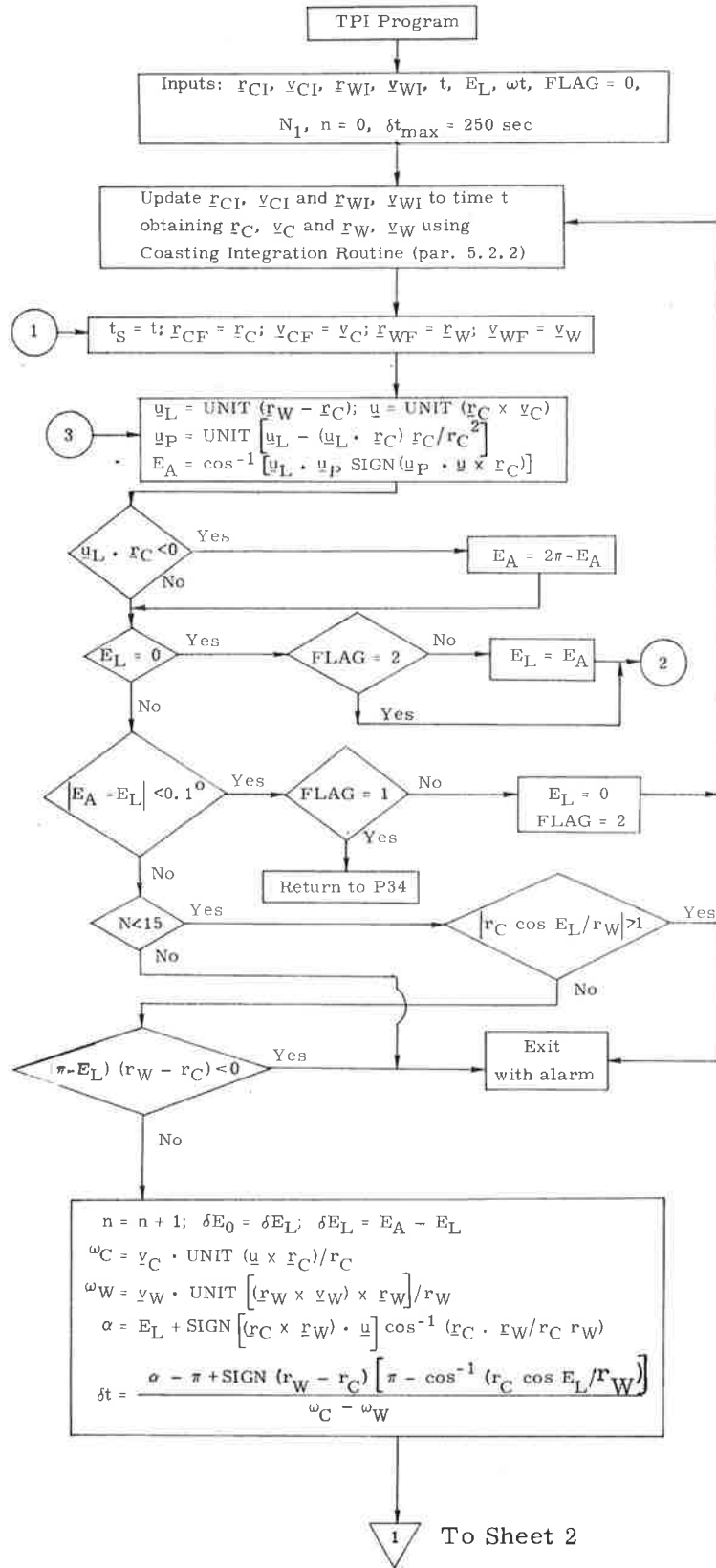


Figure 4.6-1. TPI Targeting Program Flowchart (Sheet 1 of 3)

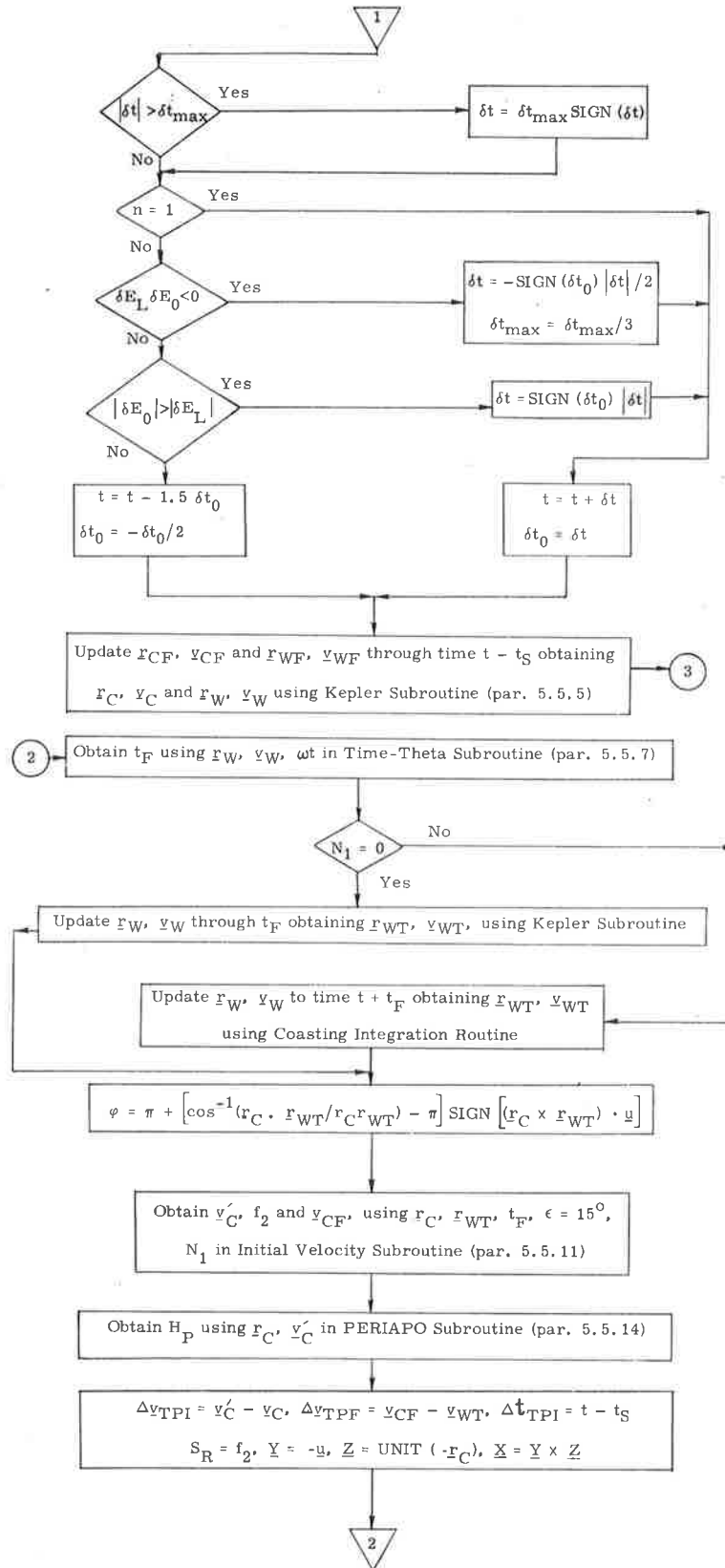


Figure 4.6-1. TPI Targeting Program Flowchart (Sheet 2 of 3)

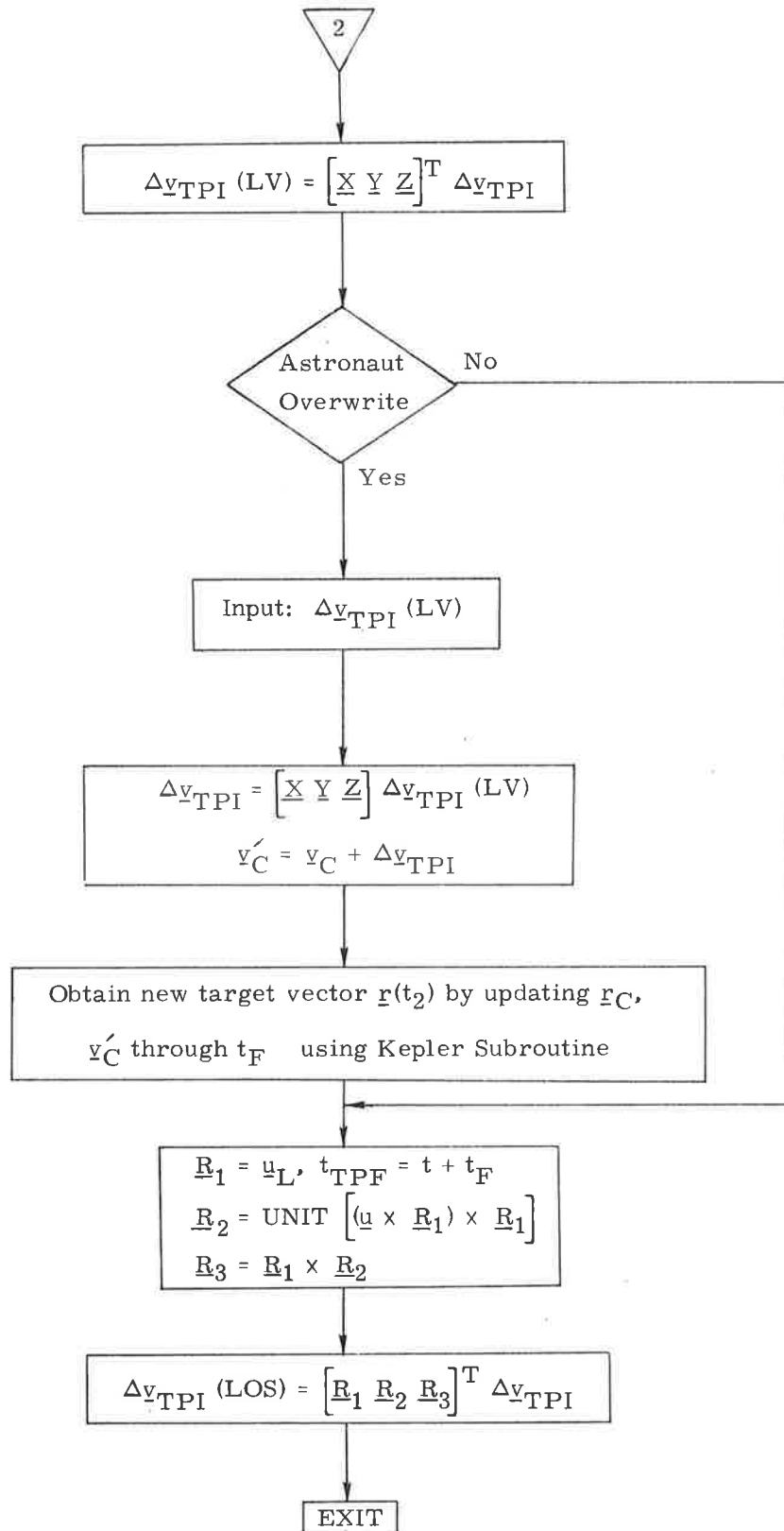


Figure 4.6-1. TPI Targeting Program Flowchart (Sheet 3 of 3)

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 CM does not intersect the circular orbit with radius equal to that of the WS.
3. The elevation angle is inconsistent with the relative altitudes of the two vehicles (e. g., if the elevation angle is less than 180° when the CM is above the WS).

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 ωt in the Time-Theta Subroutine to determine the corresponding transfer time t_F . The WS state vector is then updated through t_F using either a conic or precision updating routine, depending on the value of N_1 . The Initial Velocity Subroutine (paragraph 5.5.11) is then called to compute a transfer trajectory between the TPI and TPF points with the cone angle ϵ set equal to 15 degrees. The rotation projection switch f_2 and $\underline{r}(t_2)$ are obtained from this routine for use in the powered-flight steering program as S_R and the offset target vector.

The display of Δv_{TPI} (LV) may be overwritten by the astronaut. If so, a new offset target vector is generated and Δv_{TPI} (LOS) is recomputed as shown in Figure 4.6-1. This new aimpoint is used for powered flight steering.

The TPI program computes the following data for display:

1. Δv_{TPI} — TPI maneuver magnitude
2. Δv_{TPI} (LV) — TPI maneuver in local-vertical coordinates
3. Δv_{TPI} (LOS) — TPI maneuver in line-of-sight coordinates
4. Δv_{TPF} — TPF maneuver magnitude
5. Δt_{TPI} — Time between the computed TPI time and the input TPI time

- | | | | |
|----|-----------|---|---|
| 6. | t | — | Computed TPI time |
| 7. | E_L | — | Computed elevation angle |
| 8. | φ | | CM central angle between TPI
and intercept |

5.4.7 TPM Targeting Program (P36)

The Terminal Phase Midcourse (TPM) Targeting Program (P36) computes the terminal phase midcourse correction maneuver. This maneuver ensures CM-WS rendezvous at the time established in the TPI program. The astronaut can call this program any time after the TPI maneuver, but in general no later than 10 minutes before the intercept time. Figure 4.7-1 flowcharts the TPM Program.

The inputs to program P36 available in the guidance computer are as follows:

1. CM and WS state vectors, $\underline{r}_C, \underline{v}_C, \underline{r}_W, \underline{v}_W$
2. Intercept time, t_{TPI}
3. Number of precision offsets, N_1
4. Time delay, δt (the time required to prepare for the maneuver)

When the program is initiated, the number of navigation measurements since the last W-matrix reinitialization and the time since last maneuver, or maneuver calculation are displayed. Based on this information and the additional displays discussed in paragraph 5.6.7, the astronaut can choose to proceed with the midcourse maneuver at some time. When he does so, the program precision updates the state vectors to the present time plus the time delay. After updating the WS state vector to the time t_{TPI} to obtain the target vector $\underline{r}_{\text{WT}}$, the Initial Velocity Subroutine is called, with the cone angle ϵ set equal to 15° , to obtain the velocity vector \underline{v}'_C on the transfer ellipse. The rotation projection switch f_2 and $\underline{r}(t_2)$ are obtained from this routine for use in the powered-flight steering program as S_R and the offset target vector.

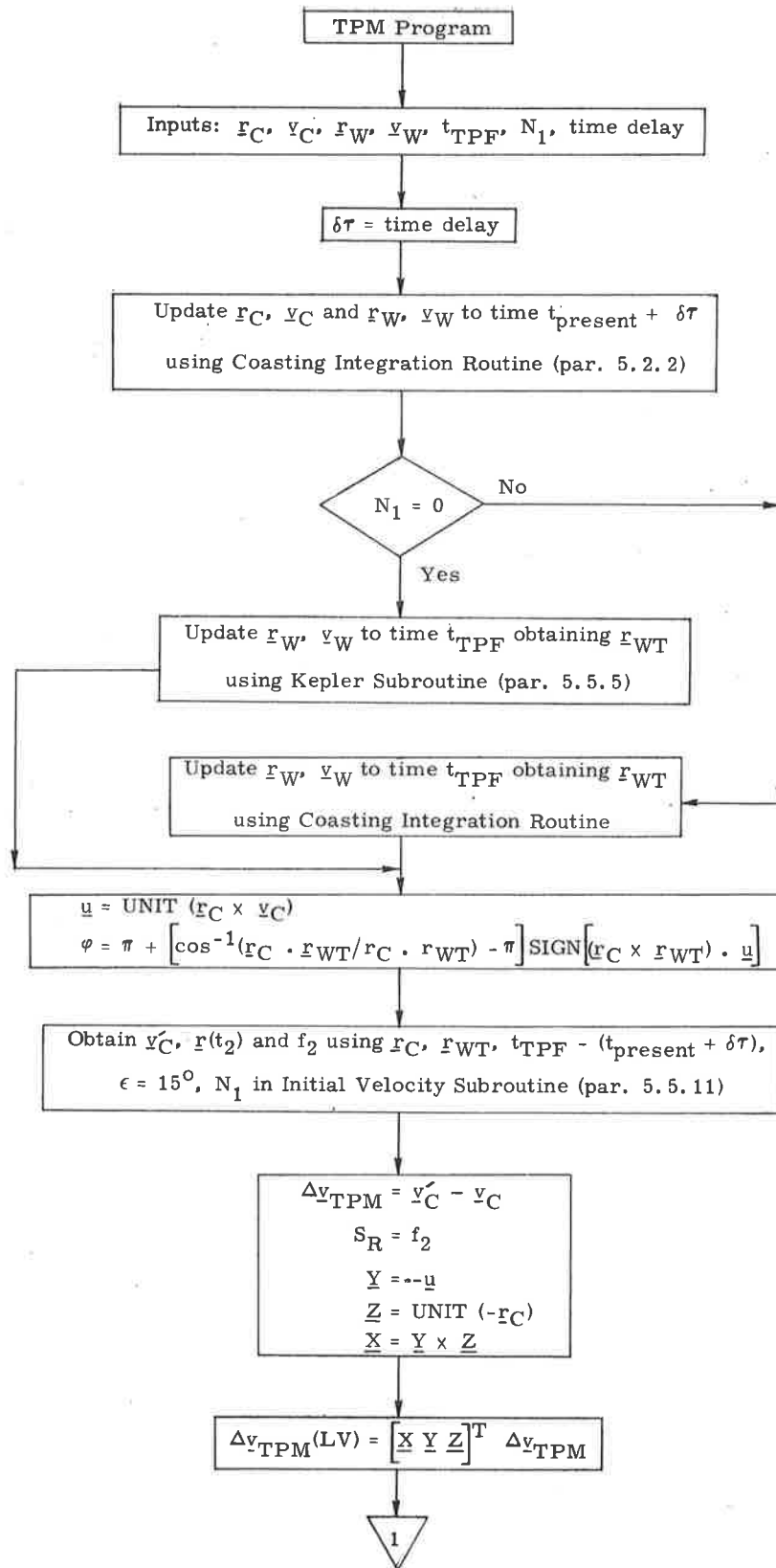


Figure 4.7-1. TPM Targeting Program Flowchart (Sheet 1 of 2)

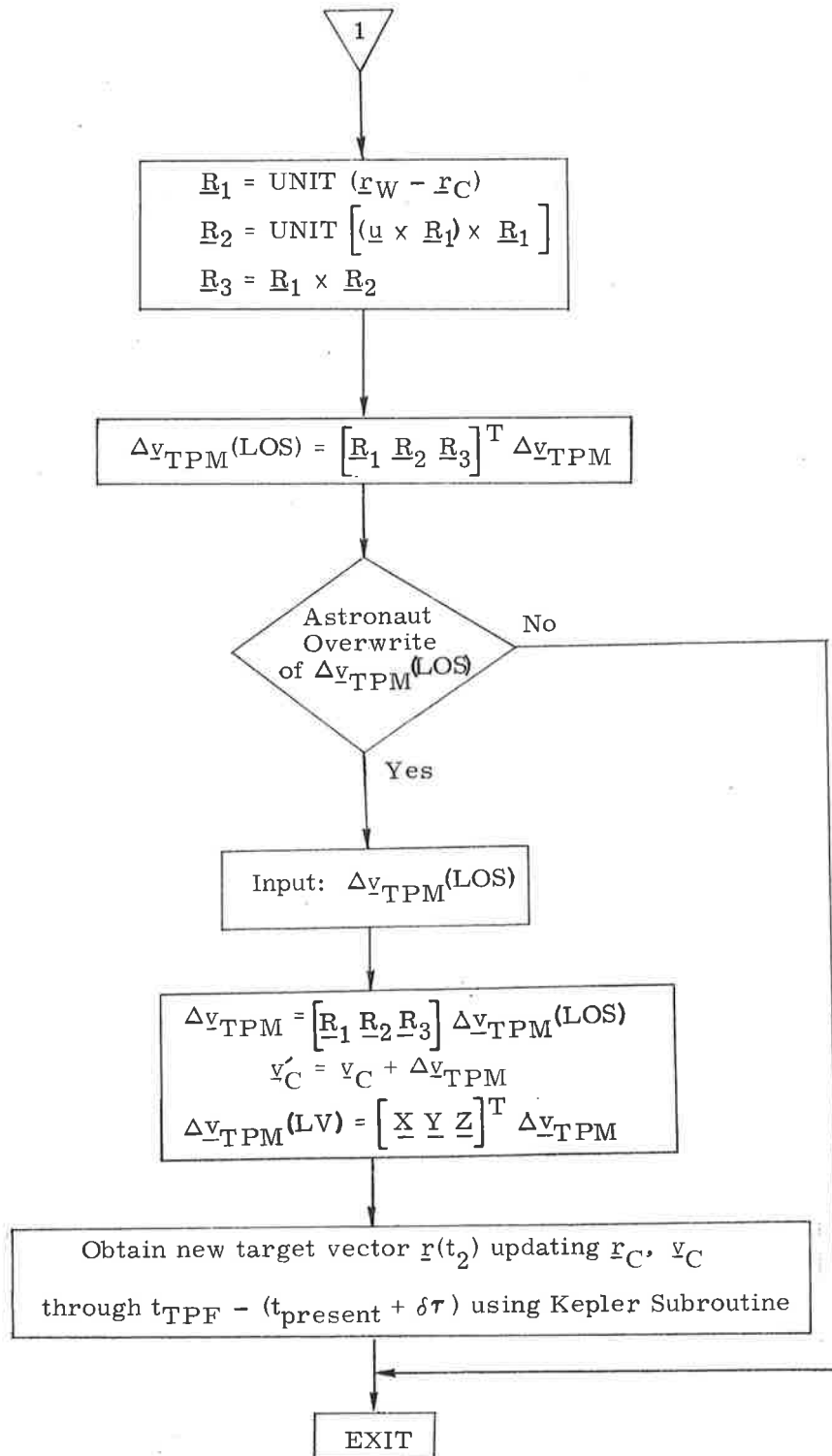


Figure 4.7-1. TPM Targeting Program Flowchart (Sheet 2 of 2)

P35 computes the following quantities for display:

1. Δy_{TPM} (LV) — TPM maneuver in local-vertical coordinates
2. Δy_{TPM} (LOS) — TPM maneuver in line-of-sight coordinates
3. φ — CM central angle between TPM and intercept

The display of Δy (LOS) can be overwritten by the astronaut. If so, a new offset target vector is generated and the Δy_{TPM} (LV) is recomputed as shown in Figure 4.7-1. This new aimpoint is used for powered-flight steering.

5.4.8 NPC Targeting Program (P38)

The plane change NPC Targeting Program (P38) computes the plane-change (PC) maneuver. The inputs to this program are the state vectors of the CM and WS and the time t of the last maneuver. This last maneuver, made by the CM, is assumed to result in a nodal point after a 90-degree traverse from the maneuver position.

Figure 4.8-1 contains a flow diagram of the program, which starts with a precision integration of the CM state vector to the time of the last maneuver. The time t_{PC} of the nodal crossing is then computed by using the Time-Theta Routine to update the CM state vector through 90 degrees. Following the precision updating of both the CM and WS state vectors to the time of the plane change maneuvers, the following quantities are computed for display:

1. Δy_{PCM} (LV) — PC maneuver in local-vertical coordinates
2. t_{PC} — Time of PC maneuver
3. $Y_C, \dot{Y}_C, \dot{Y}_W$ — Out-of-plane displays (See NSR displays.)

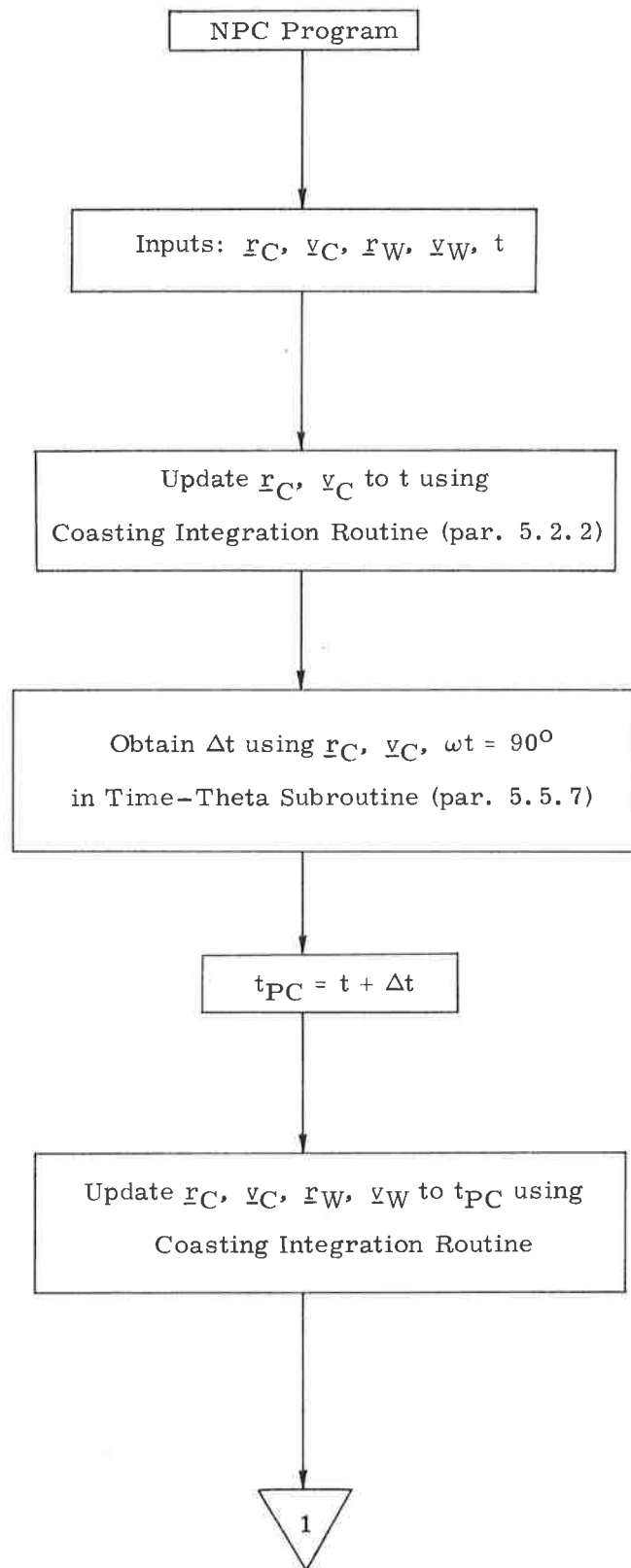


Figure 4.8-1. NPC Targeting Program Flowchart (Sheet 1 of 2)

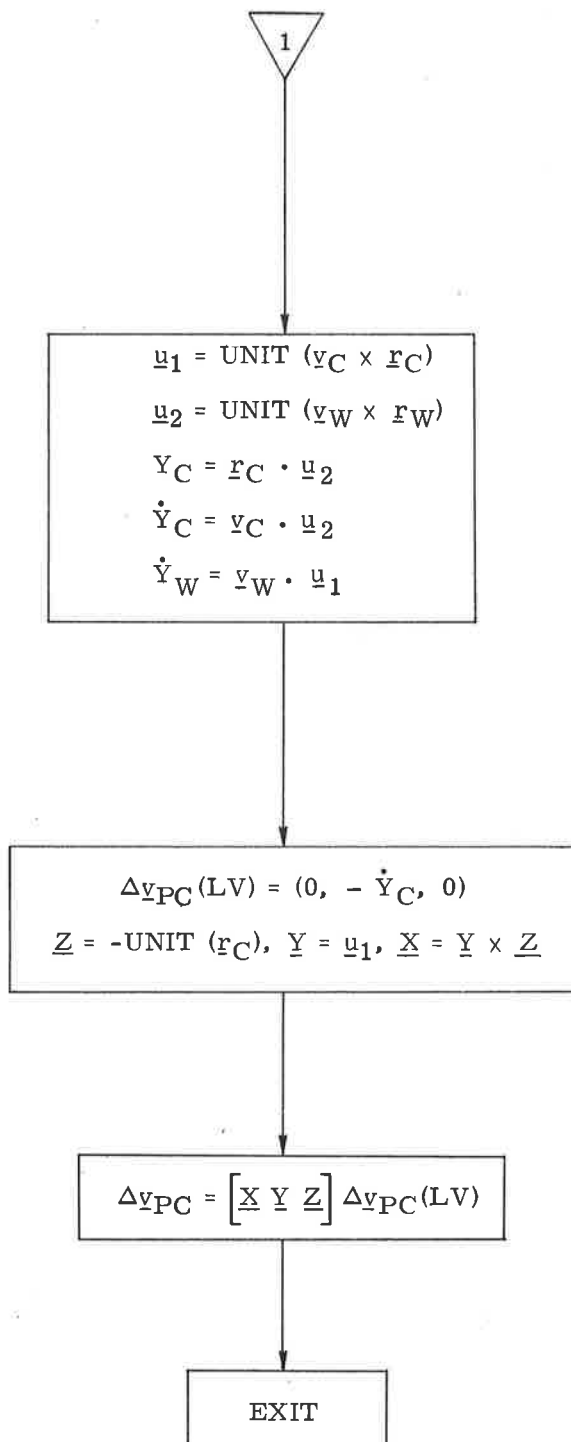


Figure 4.8-1. NPC Targeting Program Flowchart (Sheet 2 of 2)

5.4.9 Targeting Program Routines

The routines described in this section are used by the NC1, NC2, and NCC targeting programs.

5.4.9.1 REVUP Routine

The REVUP Routine (Figure 4.9-1) conically updates a state vector \underline{r} , \underline{v} through a specified number n of revolutions. The semimajor axis a is first calculated, and then used to convert n revolutions into a corresponding time by multiplying n by the period of the orbit. After conically updating the state through this time, the program returns the new state and the time.

5.4.9.2 RADUP Routine

The RADUP Routine (Figure 4.9-2) conically updates the WS state vector \underline{r}_W , \underline{v}_W to be radially above the position of the CM \underline{r}_C . The routine initially calculates the central angle θ between the two vehicles as a positive angle—if the CM leads the WS. The WS state vector is then updated through θ , using the Time-Theta Subroutine to obtain the desired state.

5.4.9.3 COE Routine

The COE Routine (Figure 4.9-3) computes the CM coelliptic state based on the state vector \underline{r} , \underline{v} of the WS and a delta altitude h . After computing the semimajor axis a of the WS orbit, COE calculates the desired semimajor axis a_D of the CM orbit. This data is then used to compute the desired radial component of velocity v_V and the velocity vector \underline{v}_C for the coelliptic orbit.

5.4.9.4 ITER Routine

The ITER Routine (Figure 4.9-4) encompasses a Newton-Raphson iteration scheme based on empirical partials. It utilizes as inputs:

1. Current x and old x_0 values of the independent variable (The independent variable is velocity (Δv measured in feet per sec) when ITER is called by the NC1 and NC2 programs and time (t measured in sec) when called by the QRDTPI Routine.)
2. Current e and old e_0 values of the dependent variable
3. Estimate of the partial $p = de/dx$

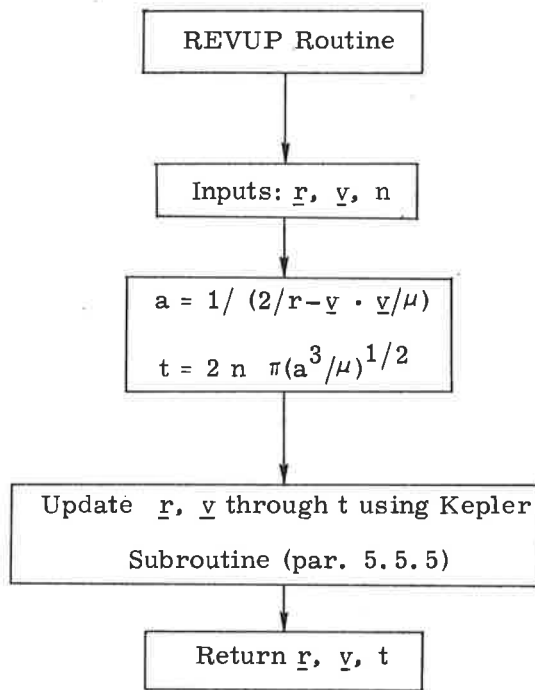


Figure 4.9-1. REVUP Routine Flowchart

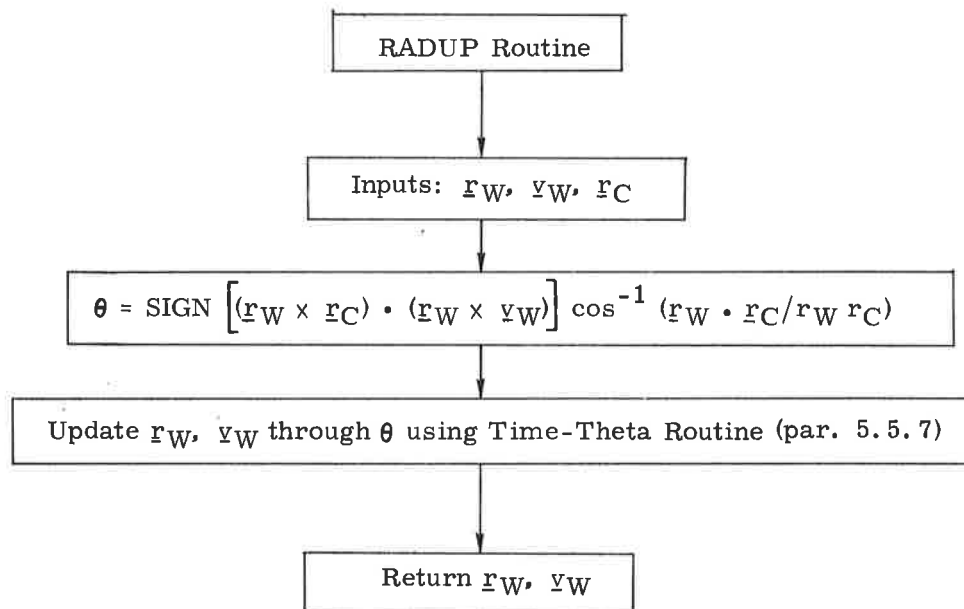


Figure 4.9-2. RADUP Routine Flowchart

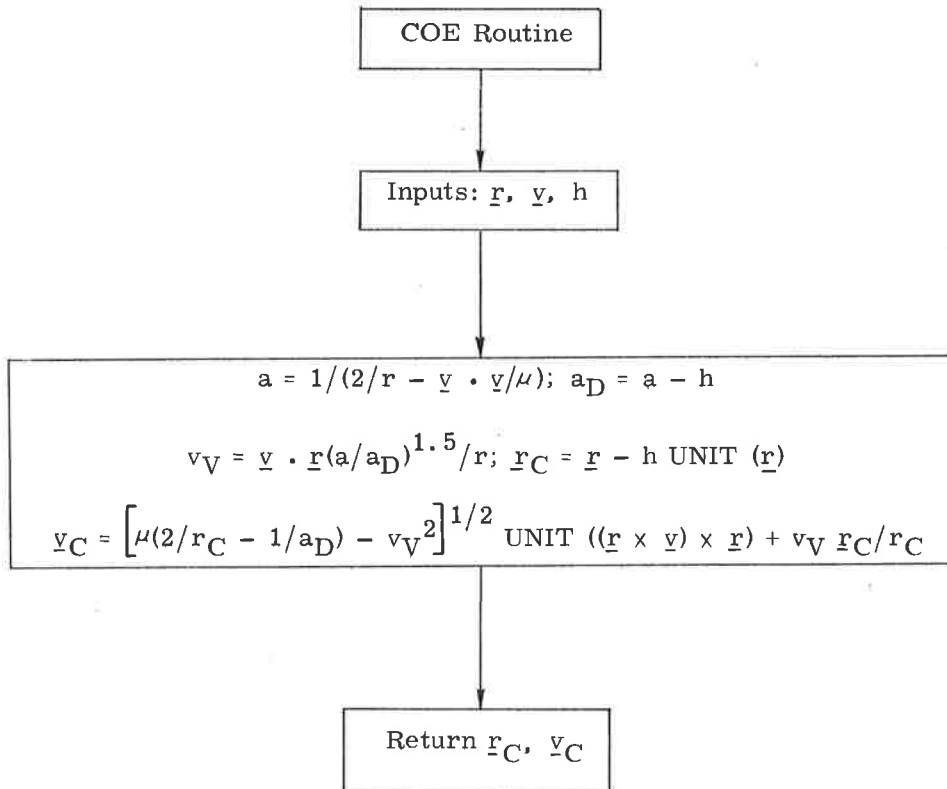


Figure 4.9-3. COE Routine Flowchart

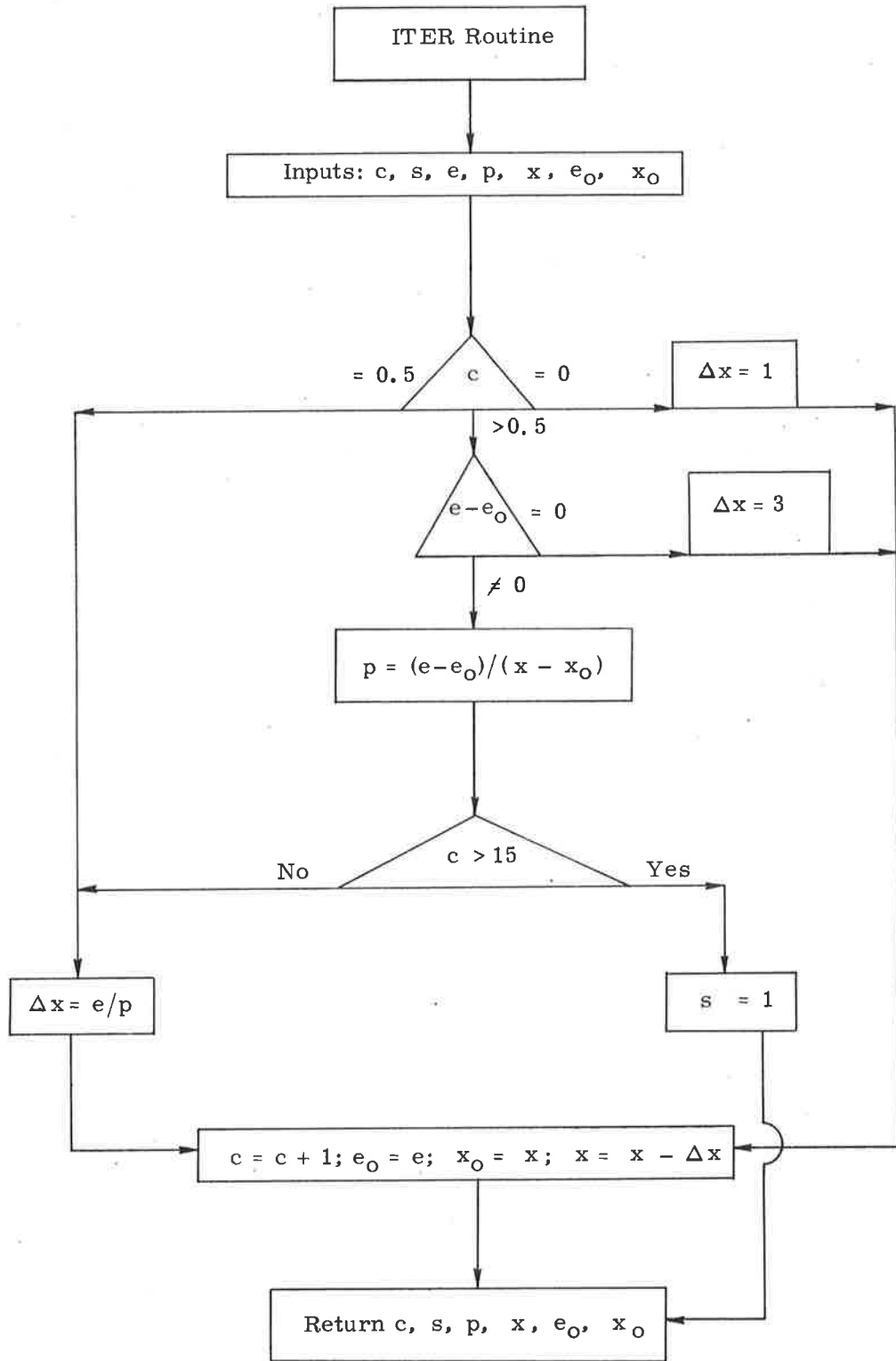


Figure 4.9-4. ITER Routine Flowchart

4. An iteration counter c that also functions as a control switch as follows:
 - $c = 0$ — First pass, p equals zero; increment x by one
 - $c = 0.5$ — First pass with nonzero p ; compute change in x to be e/p
 - $c > 0.5$ — Change x based on new partial computed from e , e_o , x , and x_o
5. A switch s , which is set equal to one in case the iteration counter exceeds 15.

In addition to computing a new estimate of the independent variable, ITER updates the quantities e_o , x_o , p , and the iteration counter c . Whenever e equals e_o , ITER increments x by 3.

5.4.9.5 QRDTPI Routine

The QRDTPI Routine (Figure 4.9-5) determines the WS state vector radially above the CM at the TPI time. The inputs to this subroutine are:

- \underline{r} , \underline{v} — WS state vector at the TPI time
- h — Altitude between the WS orbit and the CM at the TPI time
- E_L — Elevation angle
- s — Switch (1 for coasting integration; 0 for conic update)

QRDTPI uses the ITER Routine (paragraph 5.4.9.4) in an iterative search for the desired WS state vector \underline{r}_J , \underline{v}_J —which is radially above the desired CM TPI position vector. Each pass through the iterative loop involves an update of the WS state vector to the estimated time associated with the desired \underline{r}_J . The routine then calculates an error e_T that represents the difference between the desired central angle between the CM and WS position vectors at the TPI time and the actual central angle. QRDTPI then iterates until e_T is smaller than ϵ_1 , using the Newton-Raphson iterative scheme contained in the ITER Routine. If the iteration fails, alarm code 603 is displayed.

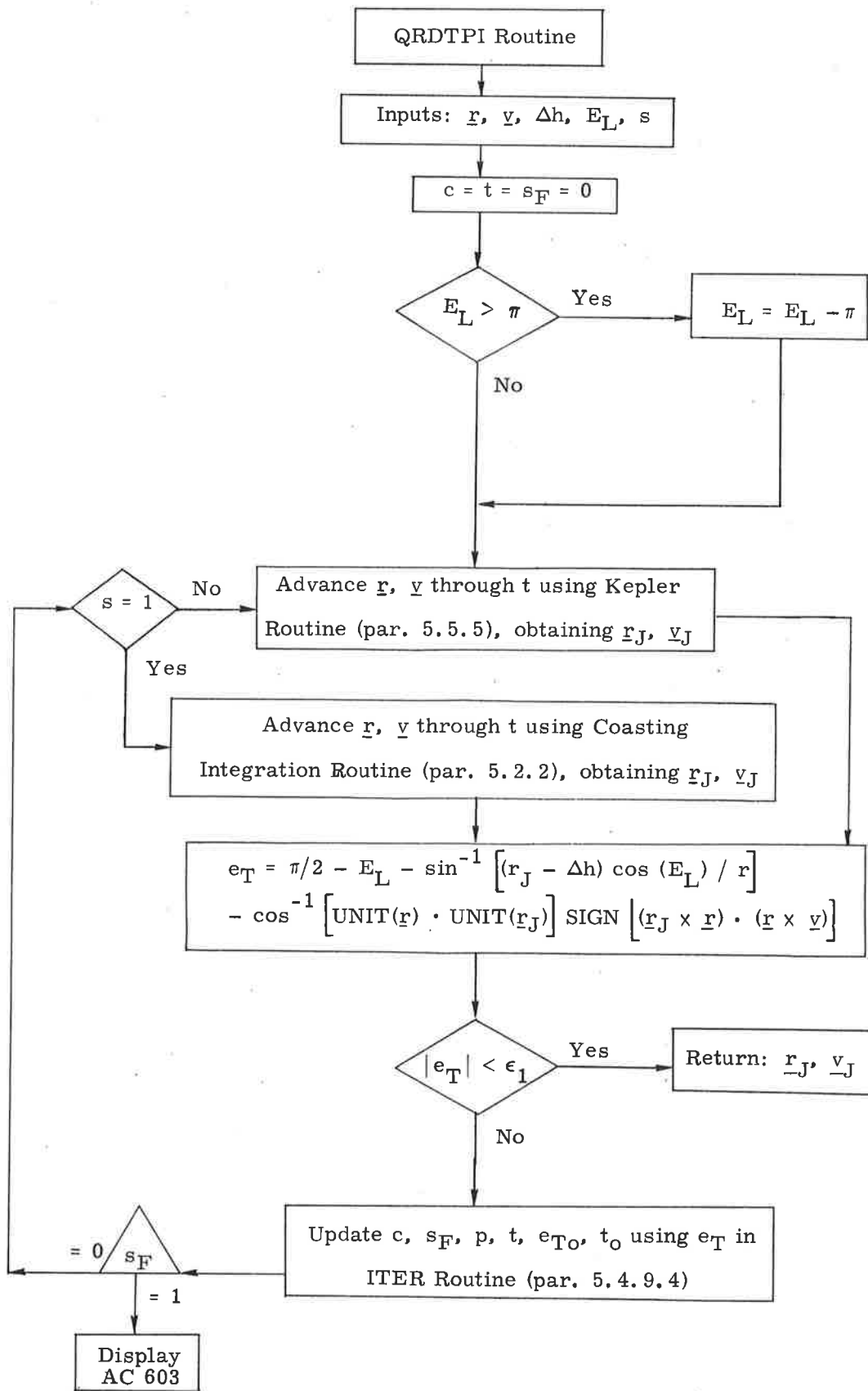


Figure 4.9-5. QRDTPI Routine Flowchart

BLANK

5.5 BASIC SUBROUTINES

5.5.1 General Comments

This section describes the basic solar system and conic trajectory subroutines that are used by the various guidance and navigation routines.

5.5.1.1 Solar System Subroutines

The transformation between the Basic Reference Coordinate System and the Earth-fixed Coordinate System is described in paragraph 5.5.2. The procedure for transforming between vectors in the Basic Reference Coordinate System and latitude, longitude, and altitude coordinates is given in paragraph 5.5.3. The method of computing the unit position vector of the sun relative to the earth is given in paragraph 5.5.4.

5.5.1.2 Conic Trajectory Subroutines

This paragraph describes a group of conic-trajectory subroutines, which are frequently used by higher level routines and programs.

These subroutines, whose block diagrams are presented in paragraphs 5.5.5 to 5.5.10, provide solutions to the following conic problems. (See nomenclature that 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 $\underline{r}(t_1)$, $\underline{v}(t_1)$, θ ; 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_r ; 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 paragraph describes six subroutines related to the five listed above.

5.5.1.2.1 Method of Solution. —To minimize the total number of computer instructions, the problems are solved in the "universal" form, i.e., only equations that are equally valid for the ellipse, parabola and hyperbola are used.

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 estimate for the independent variable. (See Equation (2.2.4) of paragraph 5.2.2.2.)

Paragraphs 5.5.5 through 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.

5.5.1.2.2 Range of Variables. —Since the values of problem variables are determined by the solution of the problem being solved, and since the problem can 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 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, limitations exist on intermediate variables and combinations of variables.

5.5.1.2.3 Scaling for Conic Subroutines (paragraphs 5.5.5 to 5.5.10)

<u>Parameter</u>	<u>Maximum Value*</u>
r	2^{29}
v	2^7
t	2^{28}
α^{**}	2^{-22}
α_N^{**}	2^6

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

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

<u>Parameter</u>	<u>Maximum Value</u> *
p_N	2^4
$\cot \gamma$	2^5
$\cot \frac{\theta}{2}$	2^5
x	2^{17}
$\xi = \alpha x^{2^{**}}$	- 50 + $4\pi^2$
$c_1 = \frac{r \cdot v}{\sqrt{\mu}}$	2^{17}
$c_2 = r v^2 / \mu - 1$	2^6
$\lambda = r(t_1) / r(t_2)$	2^7
$\cos \theta - \lambda$	2^7
e	2^3
x^2	2^{34}
$x^2 c(\xi)$	2^{33}
$x^3 s(\xi) / \sqrt{\mu}$	2^{28}
$c_1 x^2 c(\xi)$	2^{49}
$c_2 x^2 s(\xi)$	2^{35}
$x [c_2 x^2 s(\xi) + r(t_1)]$	2^{49}
$\xi s(\xi)$	2^7
$x^2 c(\xi) / r$	2^8
$\sqrt{\mu} x (\xi s(\xi) - 1) / r(t_2)$	2^{15}
$c(\xi)$	2^4
$s(\xi)$	2^1

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

** Both the maximum and minimum values are listed, since neither can be exceeded.

5. 5. 1. 2. 4 Nomenclature for Conic Subroutines (paragraphs 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
α	reciprocal of semimajor axis (negative for hyperbolas)
r_P	radius of pericenter
r_A	radius of apocenter
e	eccentricity
α_N	ratio of magnitude of initial position vector to semi-major axis
p_N	ratio of semi-latus rectum to initial position vector magnitude
γ	inertial flight path angle as measured from vertical
θ	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_{21}	computed transfer time from Kepler's equation ($t_2 - t_1$)
t'_{21}	transfer time corresponding to the previous solution of Kepler's equation
t_D	desired transfer time through which the conic update of the state vector is to be made
t_{D21}	desired transfer time to traverse from $\underline{r}(t_1)$ to $\underline{r}(t_2)$
t_{ERR}	error in transfer time
ϵ_t	fraction of desired transfer time to which t_{ERR} must converge
Δx	increment in x which will produce a smaller t_{ERR}
ϵ_x	value of Δx which will produce no significant change in t_{21}
$\Delta \cot \gamma$	increment in $\cot \gamma$ which will decrease the magnitude of t_{ERR}
ϵ_c	value of $\Delta \cot \gamma$ which will produce no signifi- cant change in t_{21}
μ_E	product of universal gravitational constant and earth's mass
x_{MAX}	maximum value of x
x_{MIN}	minimum value of x
\cot_{MAX}	maximum value of $\cot \gamma$
\cot_{MIN}	minimum value of $\cot \gamma$

l_{MAX}	upper bound of general independent variable
l_{MIN}	lower bound of general independent variable
x_{MAX0}	absolute upper bound on x
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
Δz	increment in z that will produce a smaller y_{ERR}
s_G	a sign that is plus or minus according to whether the true anomaly difference between $\underline{r}(t_1)$ and $\underline{r}(t_2)$ is to be less than or greater than 180 degrees
s_r	a sign that is plus or minus according to whether the desired radial velocity at $r(t_2)$ is plus or minus
$\underline{\eta}_1$	general vector # 1
$\underline{\eta}_2$	general vector # 2
ϕ	angle between $\underline{\eta}_1$ and $\underline{\eta}_2$
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 $\underline{r}(t_1)$ and $\underline{r}(t_2)$ 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
- 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
- k_1 the minimal acceptance fraction of t_{DZ1} 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

5.5.1.3 Additional Subroutines

The computation of the required initial velocity vector for a trajectory of specified transfer between initial and target position vectors is described in paragraph 5.5.11. Paragraph 5.5.13 discusses a subroutine which computes the lines-of-sight of the sun and earth relative to the spacecraft in the Basic Reference Coordinate System. The computation of the two-body apocenter and pericenter altitudes, given the position and velocity vectors for a point on the trajectory and the primary body, is described in paragraph 5.5.14. Finally, paragraph 5.5.15 discusses a program for computing the time of the first crossing of a specified longitude after a specified base time—as well as the corresponding latitude—for earth orbit.

5.5.2 Planetary Inertial Orientation Subroutine

This subroutine is used to transform vectors between the Basic Reference Coordinate System and the Earth-fixed Coordinate System at a specified time. These coordinate systems are defined in paragraph 5.1.4.

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

$$\underline{r}_E = M(t) L \underline{r} \quad (5.2.1)$$

and

$$\underline{r} = L^T M(t)^T \underline{r}_E \quad (5.2.2)$$

where $M(t)$ is a time dependent orthogonal transformation matrix, and L is a 3×3 matrix included in the prelaunch erasable data load. The third row of the L matrix is identical to the unit vector \underline{u}_Z of the earth's true north polar axis expressed in the Basic Reference Coordinate System. The vector \underline{u}_Z is used, for example, in the Coasting Integration Routine (paragraph 5.2.2.3), in the Earth Orbit Insertion Monitor, P11 (paragraph 5.3.5), and in calculating the displays described in paragraph 5.6.10.6. The matrix L is considered constant for the duration of the mission, although for long missions L can be updated if desired.

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 matrix L accounts for the effects of precession and nutation.

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

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

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 earth's equatorial plane) at the time the computer clock is zeroed, linearly backdated to the astrodynamical epoch;* t_0 is the elapsed time between the astrodynamical epoch and the time the computer clock was zeroed; and ω_E is the angular velocity of the earth.

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 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 latitude, longitude, and altitude at a specified time.

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 semimajor and semiminor axes of the Fischer ellipsoid, respectively, and SINL is the sine of the geocentric latitude.

The computational procedures are illustrated in Figures 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

*The astrodynamical epoch is nominally taken to be the Greenwich midnight universal time that begins the July first preceding the launch.

t and the indicator F, where

$$F = \begin{cases} 1 & \text{for Fischer ellipsoid radius} \\ 0 & \text{for launch pad radius.} \end{cases}$$

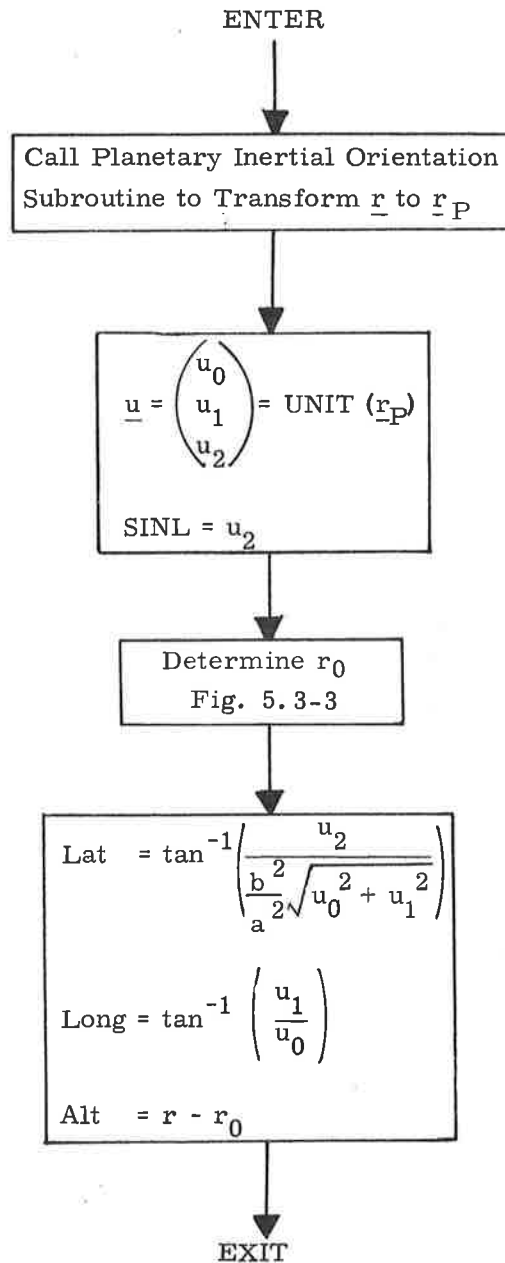


Figure 5.3-1. Vector to Latitude, Longitude, Altitude Computation Logic Diagram

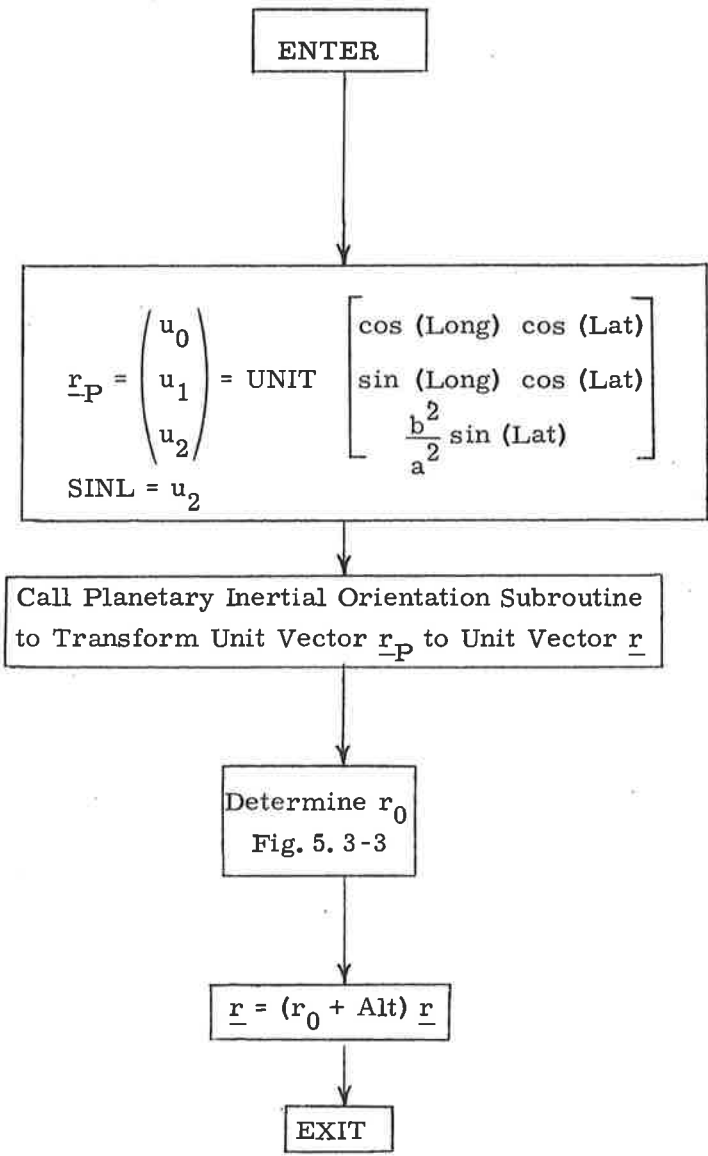


Figure 5.3-2. Latitude, Longitude, Altitude to Vector Computation Logic Diagram

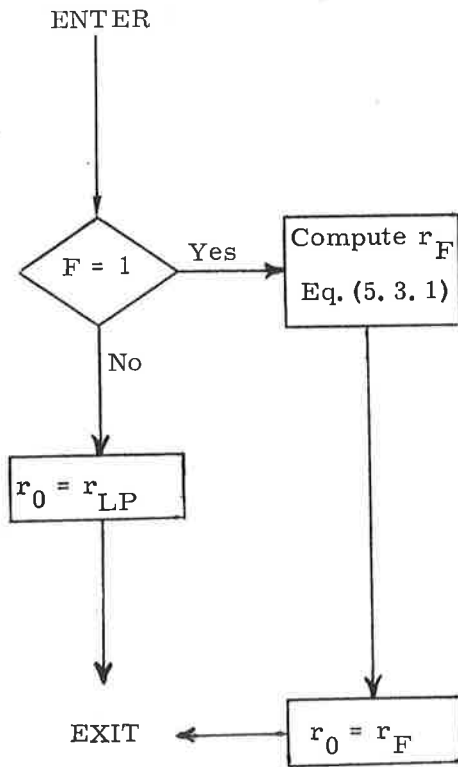


Figure 5.3-3. Determination of r_0

5. 5. 4 Solar Ephemerides

This subroutine is called by LOCSAM (paragraph 5. 5. 13) to obtain the line-of-sight unit vector of the sun with respect to the earth, which may be used for IMU alignment.

The unit position vector of the sun with respect to the earth in the Basic Reference Coordinate System is given by:

$$\underline{u}_{ES} = \begin{bmatrix} \cos (\text{LOS}) \\ K_1 \sin (\text{LOS}) \\ K_3 \sin (\text{LOS}) \end{bmatrix} \quad (5. 4. 1)$$

where the true longitude of the sun is approximated by:

$$\text{LOS} = \text{LOS}_0 + \text{LOS}_R (t + t_0) - C \sin [\text{OMEGA}_C (t + t_0) + \text{PHASE}_C], \quad (5. 4. 2)$$

and where t and t_0 are defined in Time Definitions (paragraph 5. 1. 5. 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 Figure 5.5-1. The subroutines referred to in this figure are presented in paragraph 5.5.10, and the nomenclature is found in paragraph 5.5.1.2.4.

Before entering the Kepler Subroutine an initial estimate of x can be generated via Equation (2.2.4) of paragraph 5.2.2.2 with

$$\frac{\Delta t}{2} = t_D - t_{21} \text{ and } \tau = t_D$$

However, x' and t_{21} are non-zero only if the subroutine is being used repetitively.

Although there is no upper bound, theoretically, on x , the practical bound is set to x_{MAX0} to eliminate non-feasible trajectories and increase the accuracy to which x can be computed. In addition, αx^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.

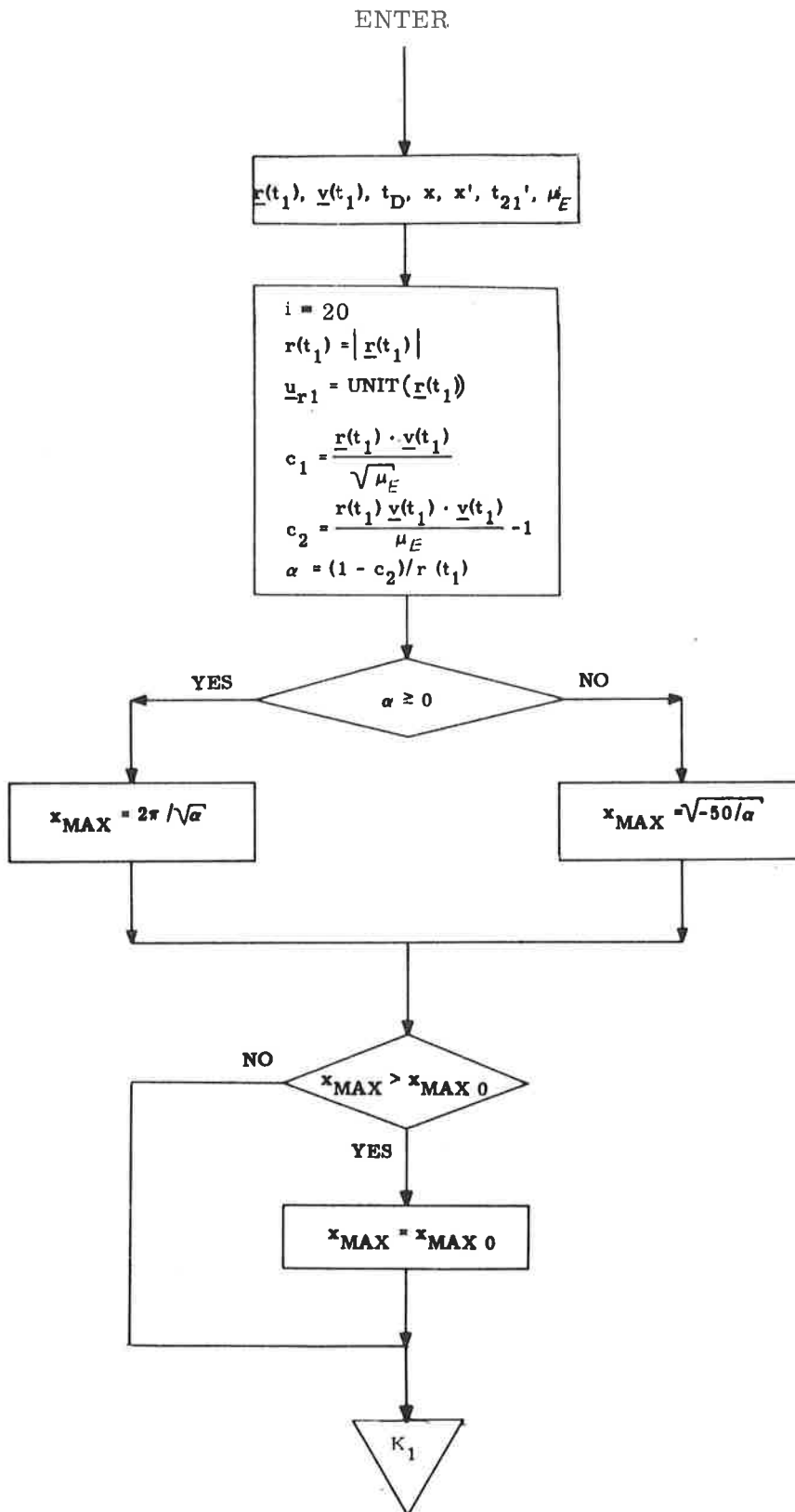


Figure 5.5-1. Kepler Subroutine (Sheet 1 of 3)

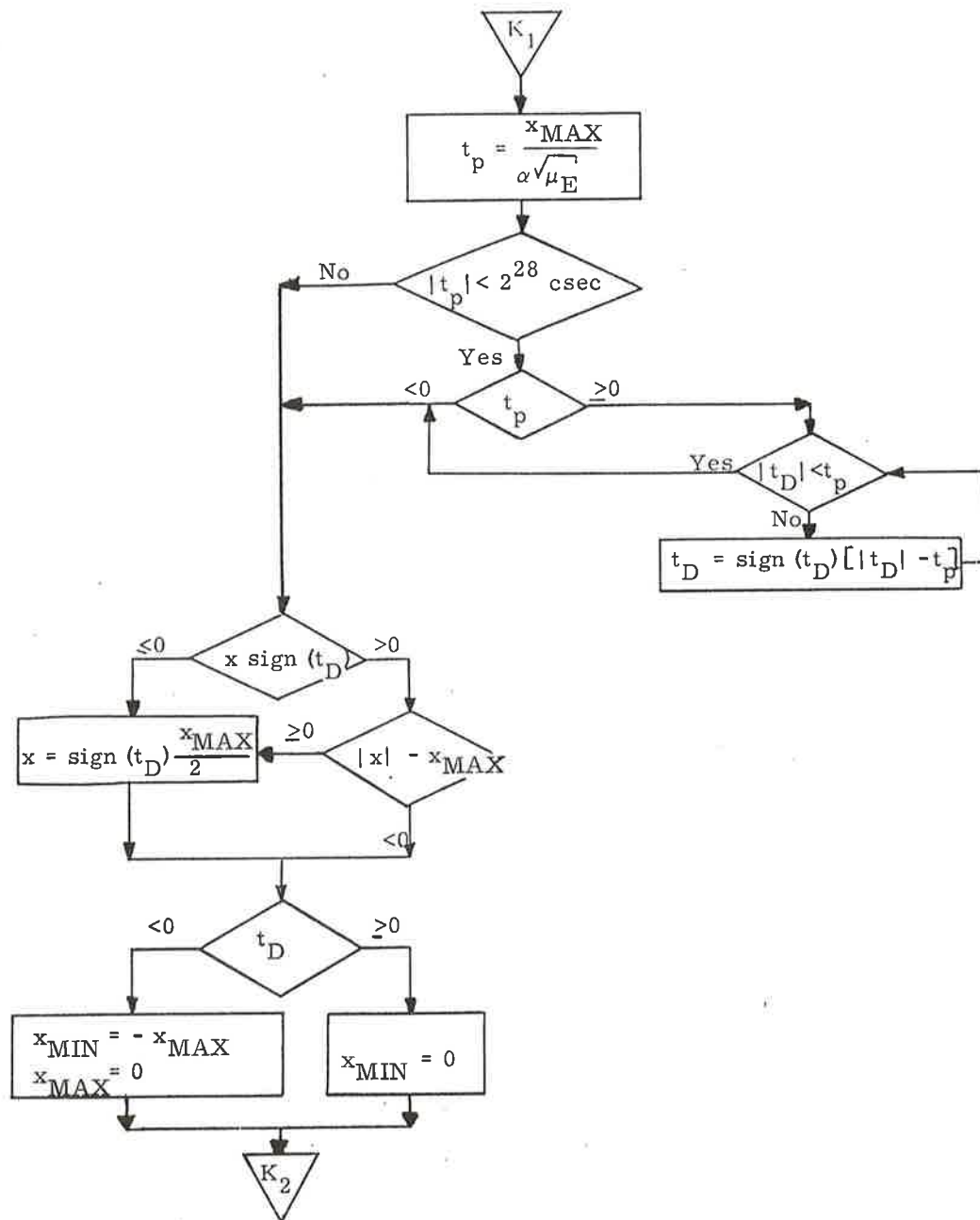


Figure 5.5-1. Kepler Subroutine (Sheet 2 of 3)

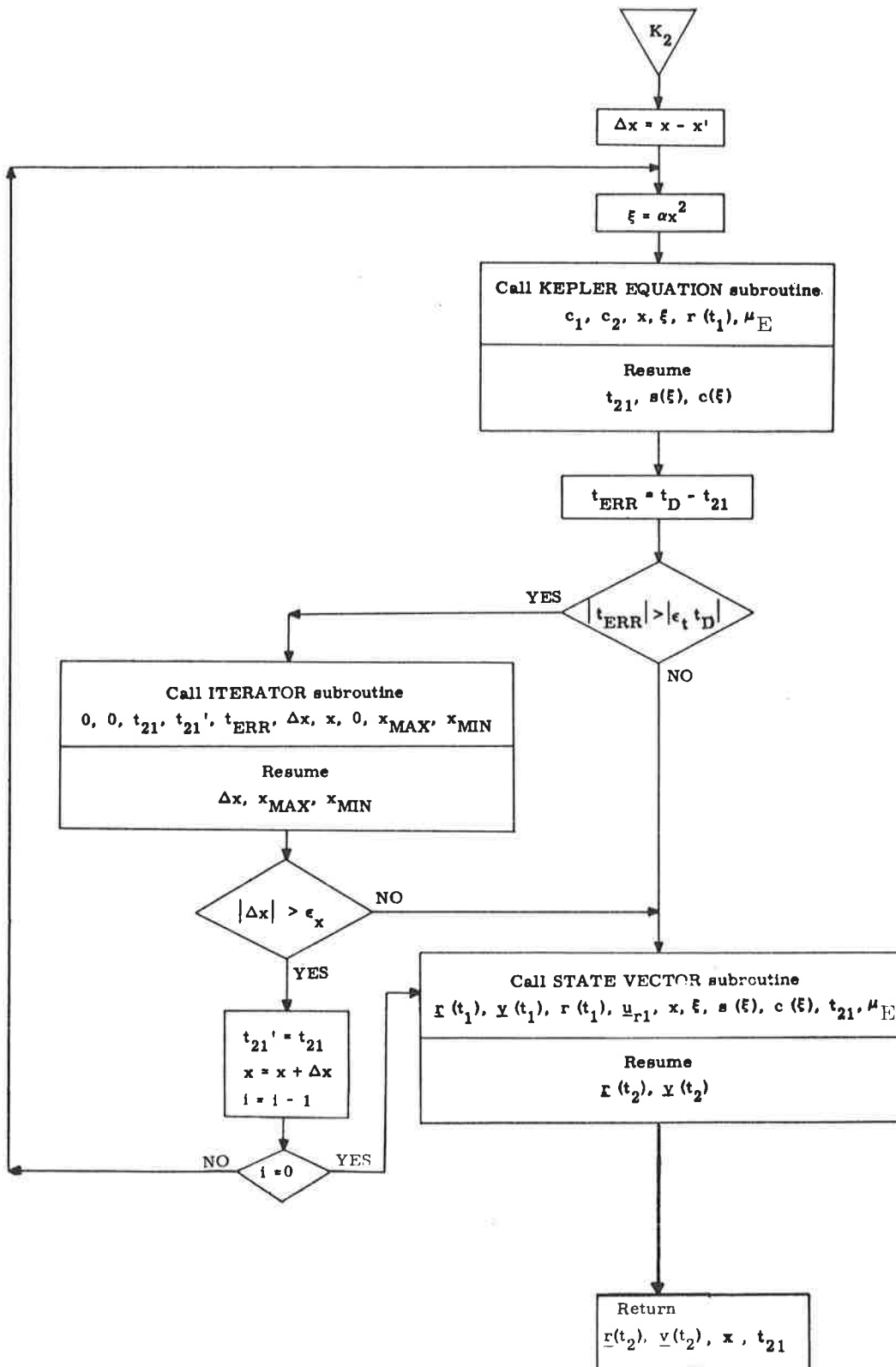


Figure 5.5-1. Kepler Subroutine (Sheet 3 of 3)

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 that exceeds the granularity in time, it is necessary to introduce ϵ_x as a redundant convergence criterion.

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

A negative value of t_D will cause the subroutine to update the state vector backward in time (i. e., backdate the state vector). The subroutine can be called to update or backdate for any amount of time; there are no restrictions on whether the time t_D is less than a period.

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 paragraph contains information to aid the reader in understanding the less obvious aspects of the Lambert Subroutine block diagrams depicted in Figure 5.6-1. The subroutines referred to in this figure are presented in paragraph 5.5.10; the nomenclature is found in paragraph 5.5.1.2.4.

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 that 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

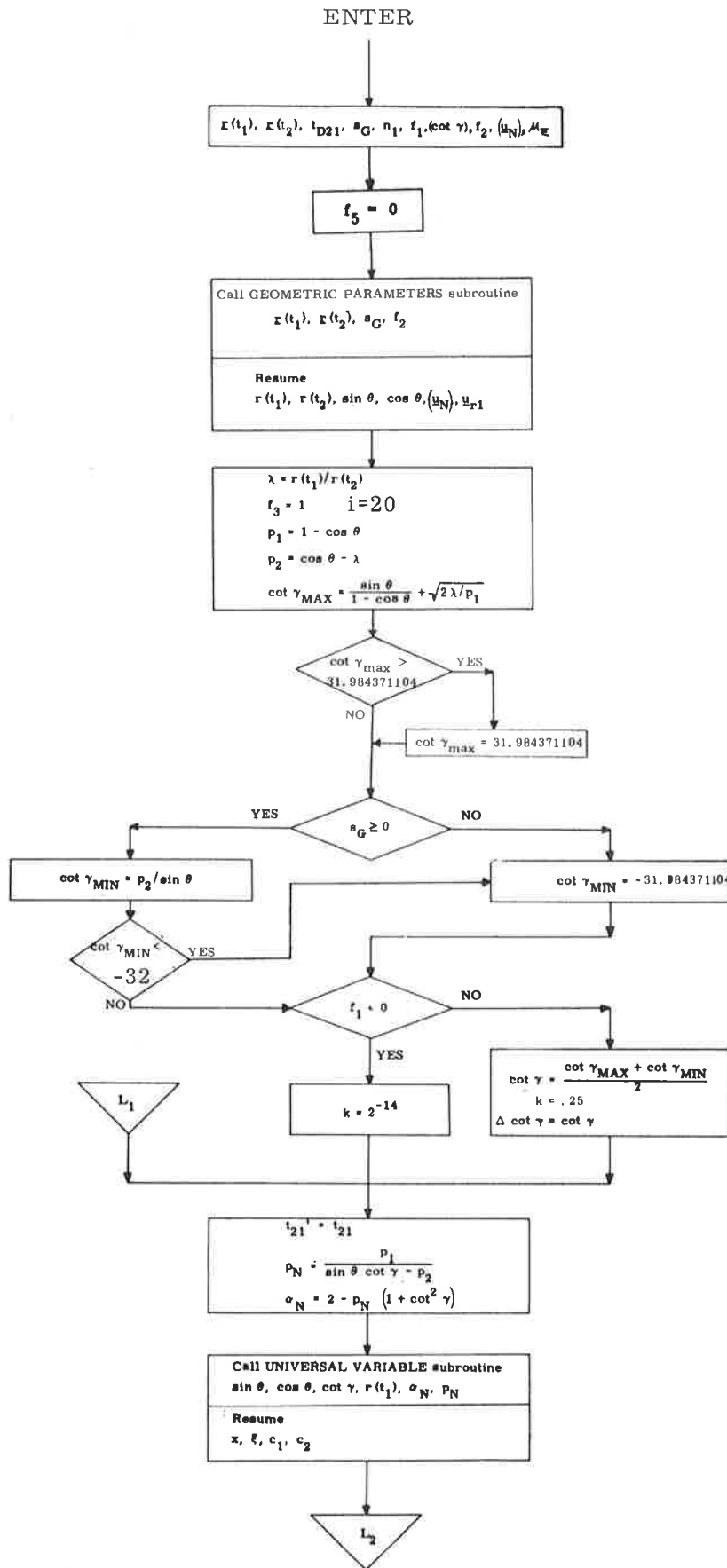


Figure 5.6-1. Lambert Subroutine (Sheet 1 of 2)

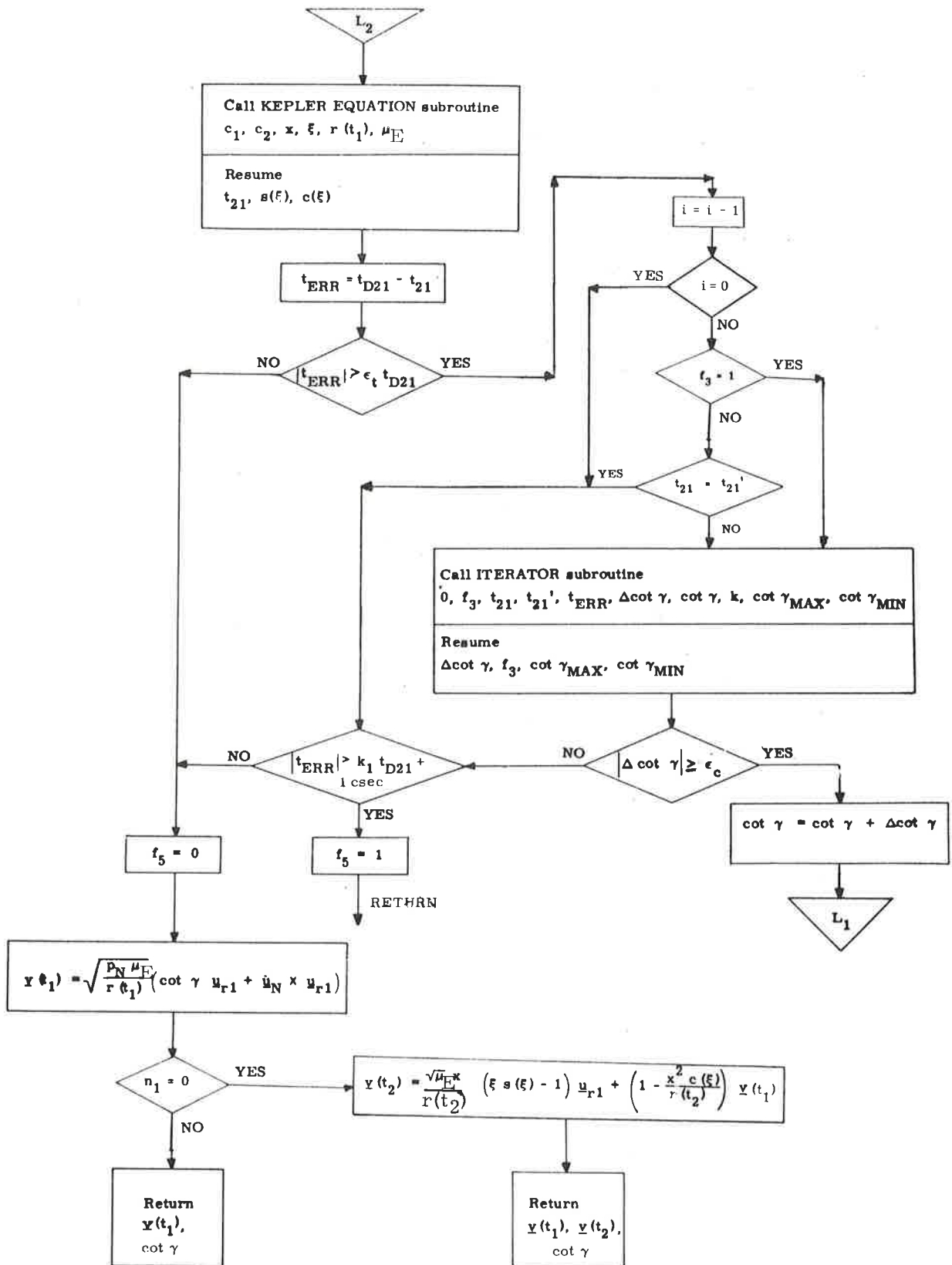


Figure 5.6-1. Lambert Subroutine (Sheet 2 of 2)

trying to find an acceptable solution. (This logic does not appear in Figures 5.6-1 and 5.6-2, 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 ϵ_c before time converges to within a fraction ϵ_t of the desired time. This same granularity condition exists, however, when the sensitivity problem occurs as described in the Kepler Subroutine, paragraph 5.5.5. 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 ϵ_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 ϵ_c , then the solution is deemed acceptable and the required velocity is computed.

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 paragraph contains information to aid the reader in understanding the less obvious aspects of the Time-Theta Subroutine block diagram, Figure 5.7-1. The subroutines referred to in this figure are presented in paragraph 5.5.10, and the nomenclature is found in paragraph 5.5.1.2.4.

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, θ , 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 θ is no longer a meaningful problem parameter. This will not cause difficulties provided the input variables are within the specified ranges.

*If the Time-Theta Subroutine is called with inputs for which no solution is possible (for either or both of these two reasons), the subroutine will abort with an alarm code of 20607.

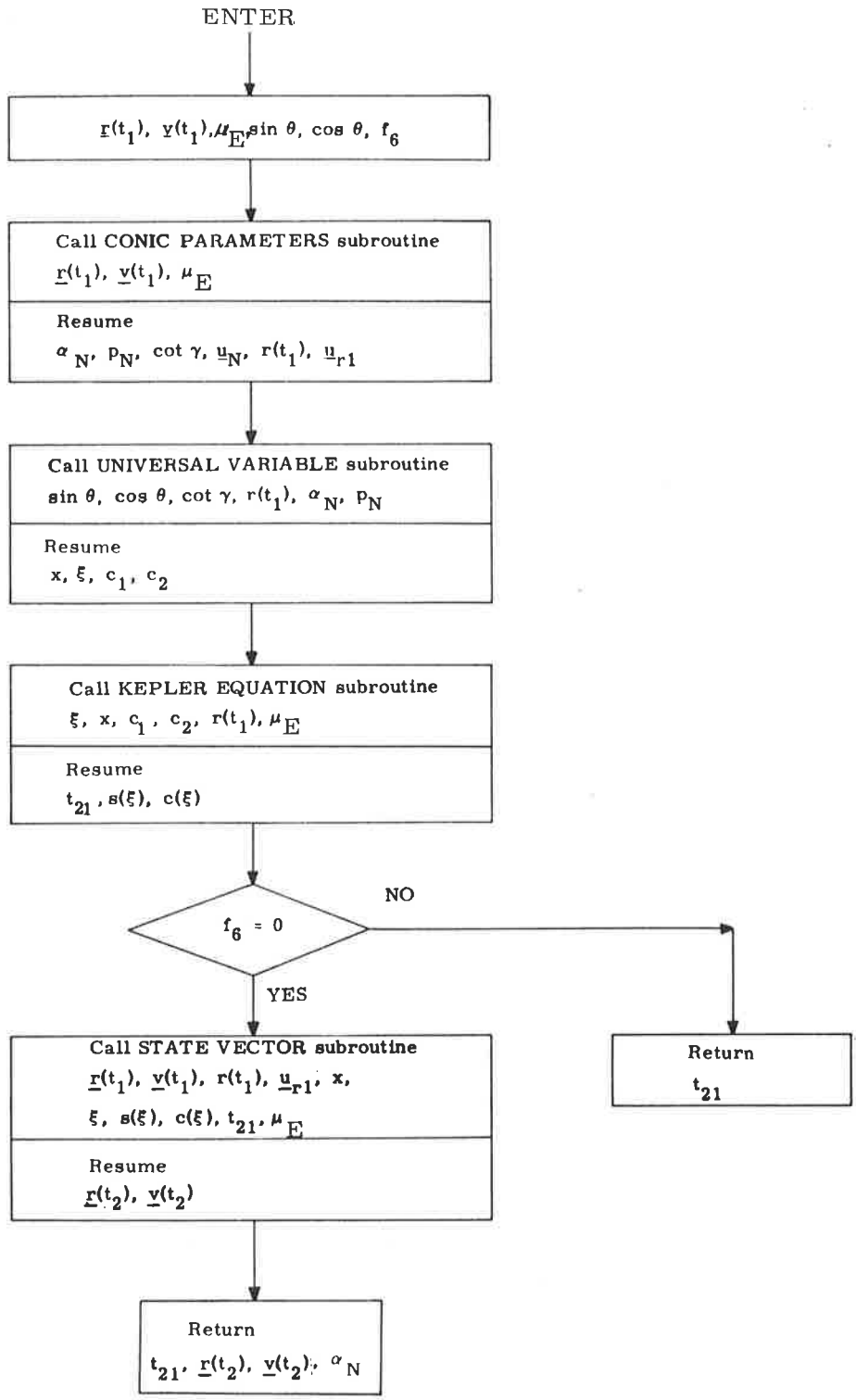


Figure 5.7-1. Time-Theta Subroutine

5. 5. 8 Time-Radius Subroutine

The Time-Radius Subroutine solves for the two-body transfer time to a specified radius given 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 diagram depicted in Figure 5.8-1. The subroutines referred to in this figure are presented in paragraph 5.5.10, and the nomenclature is found in paragraph 5.5.1.2.4. The third, fourth, and fifth paragraphs of paragraph 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 when the eccentricity is less than approximately 2^{-18} , the problem is undefined and the subroutine exits without a solution, setting flag f_9 to indicate this.* (The precise conditions are given in Figure 5.8-1.)

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

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 Figure 5.9-1. The subroutines referred to in this figure are presented in paragraph 5.5.10, and the nomenclature is found in paragraph 5.5.1.2.4.

*If the Time-Radius Subroutine is called with inputs for which no solution is possible (for any one or more of the reasons given in paragraphs 4 or 5 of paragraph 5.5.7 or paragraph 3 above), the subroutine will abort with an alarm code of 20607.

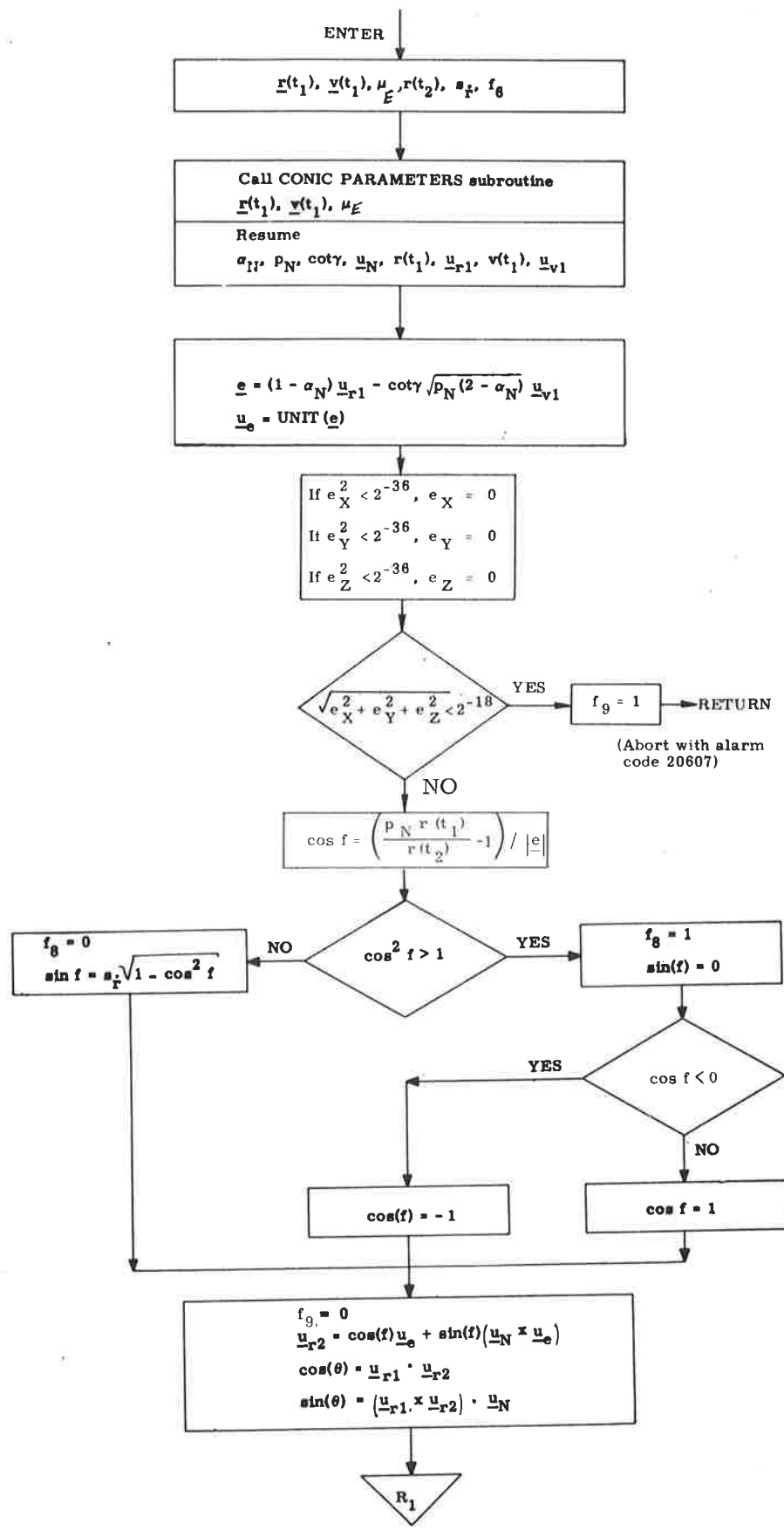


Figure 5.8-1. Time-Radius Subroutine (Sheet 1 of 2)

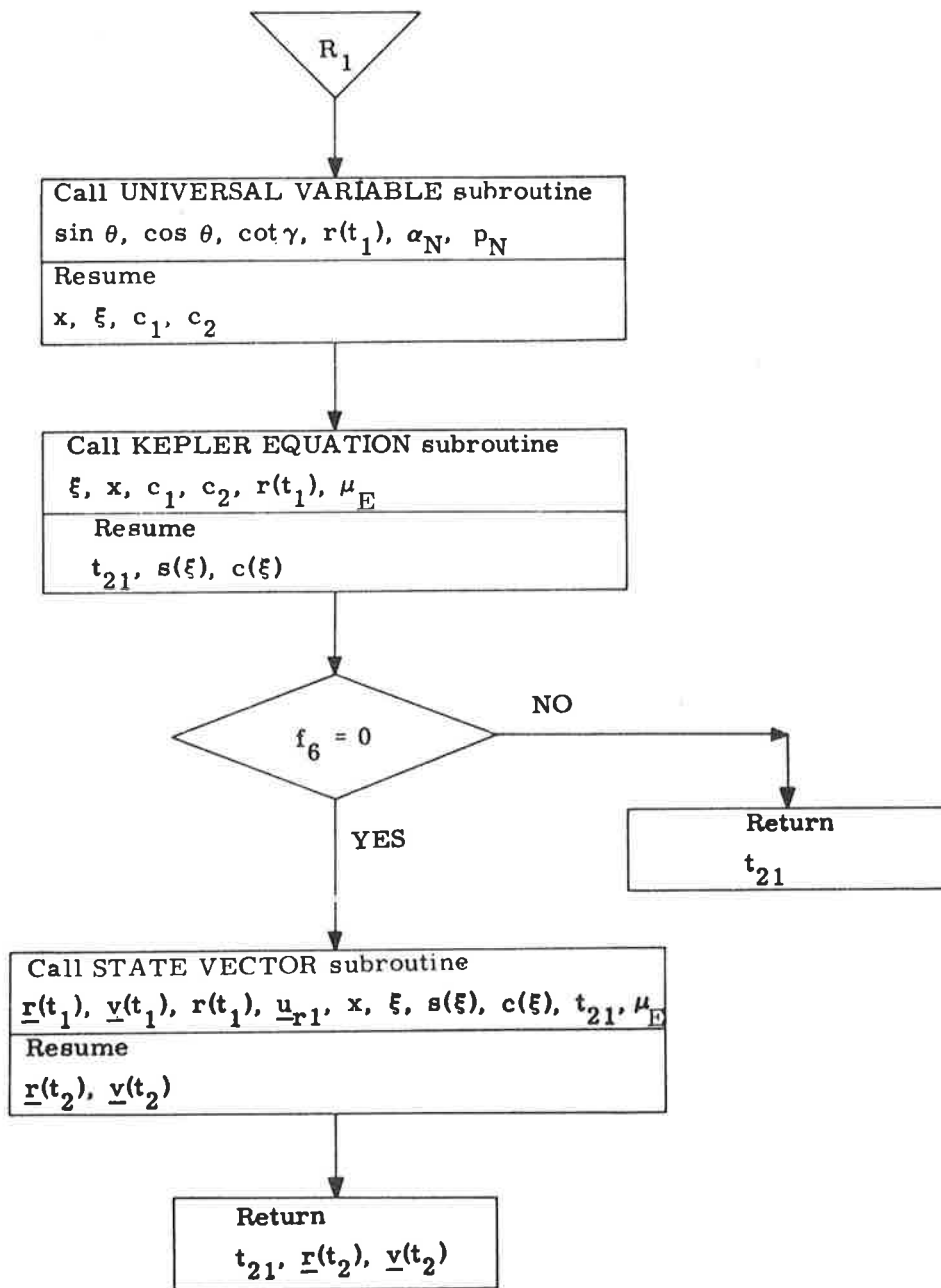


Figure 5.8-1. Time-Radius Subroutine (Sheet 2 of 2)

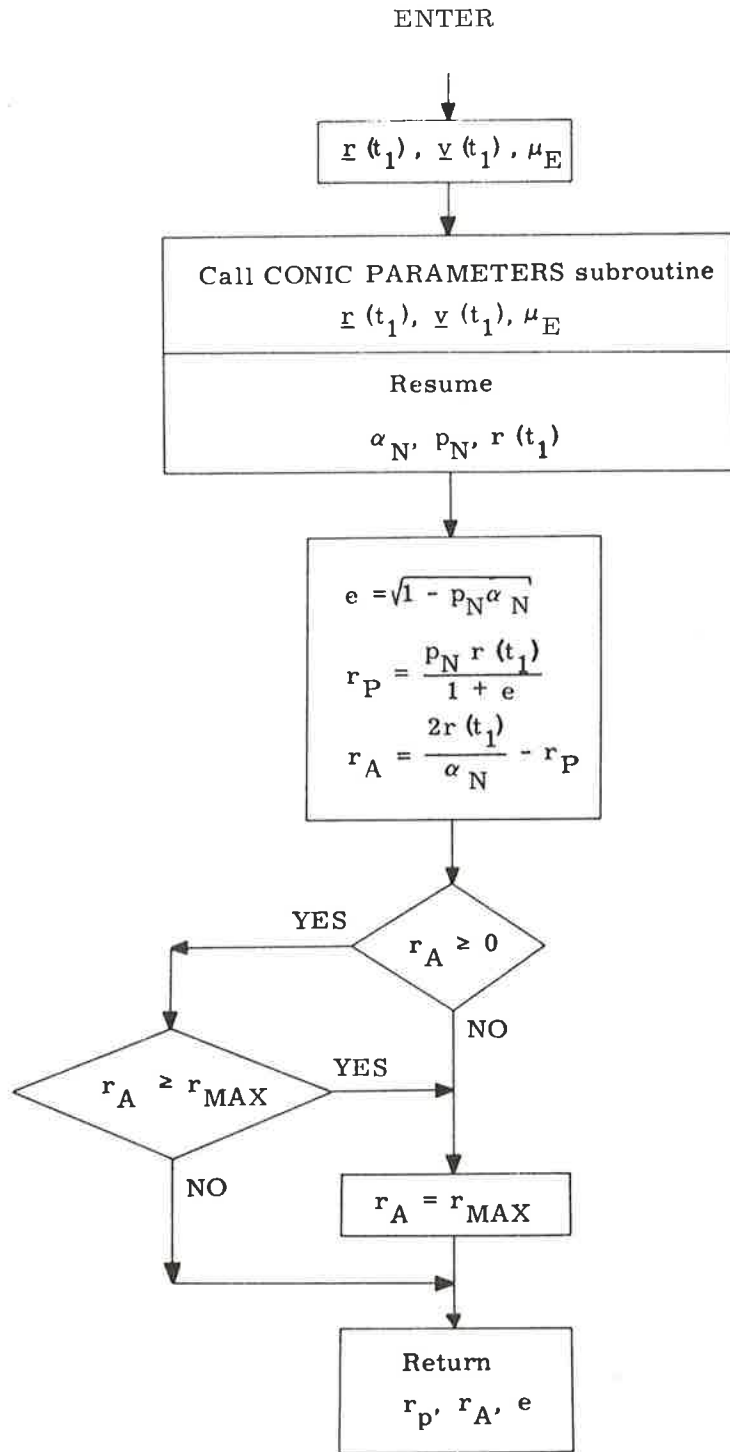


Figure 5.9-1. Apsides Subroutine

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 is set to the maximum positive value allowed by the computer.

5.5.10 Miscellaneous Subroutines

Three subroutines that are part of the conic trajectory subroutines are useful in their own right. These are the Conic Parameters, the Geometric Parameters and the Iterator Subroutines which are depicted in Figures 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, however, and requires more explanation. It is used when flag f_4 is set to zero to solve for the value of the independent variable that 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 paragraphs 5.5.5 to 5.5.9. These are the Universal Variable Subroutine, the Kepler Equation Subroutine and the State Vector Subroutine, shown in Figures 5.10-4, 5.10-5 and 5.10-6, respectively.

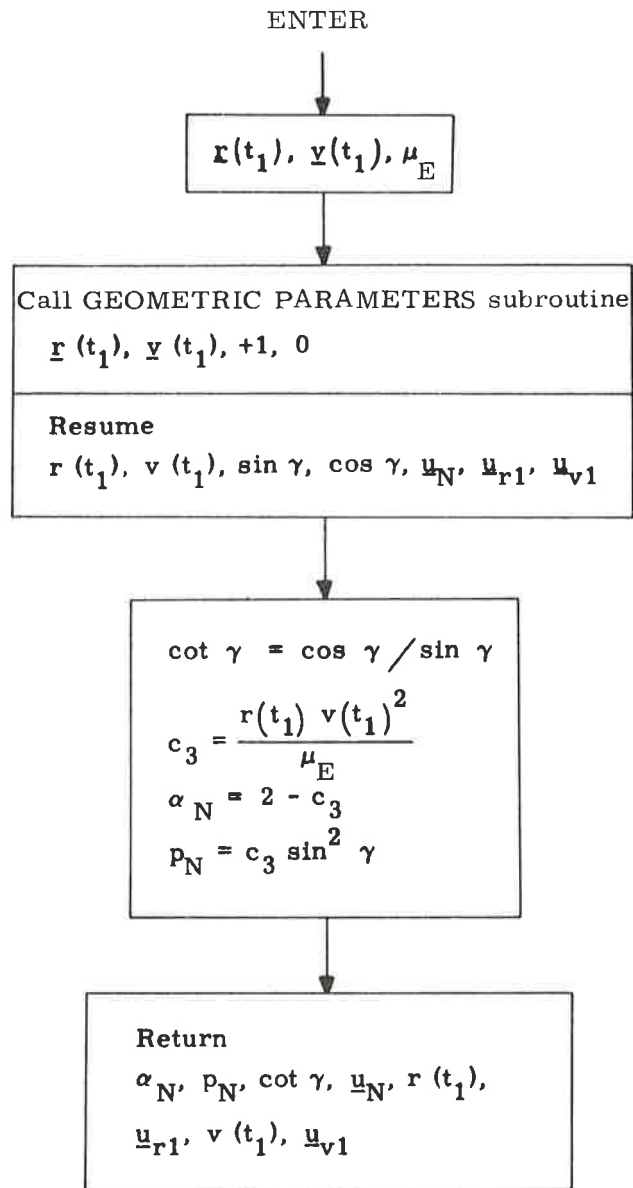


Figure 5.10-1. Conic Parameters Subroutine

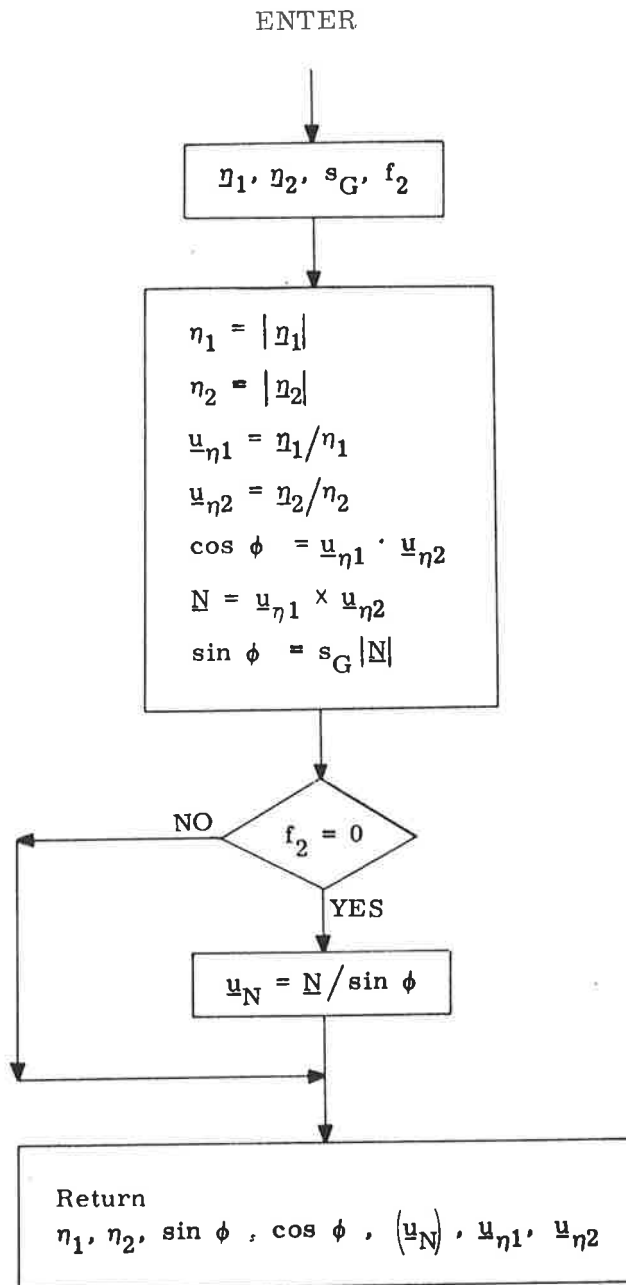


Figure 5.10-2. Geometric Parameters Subroutine

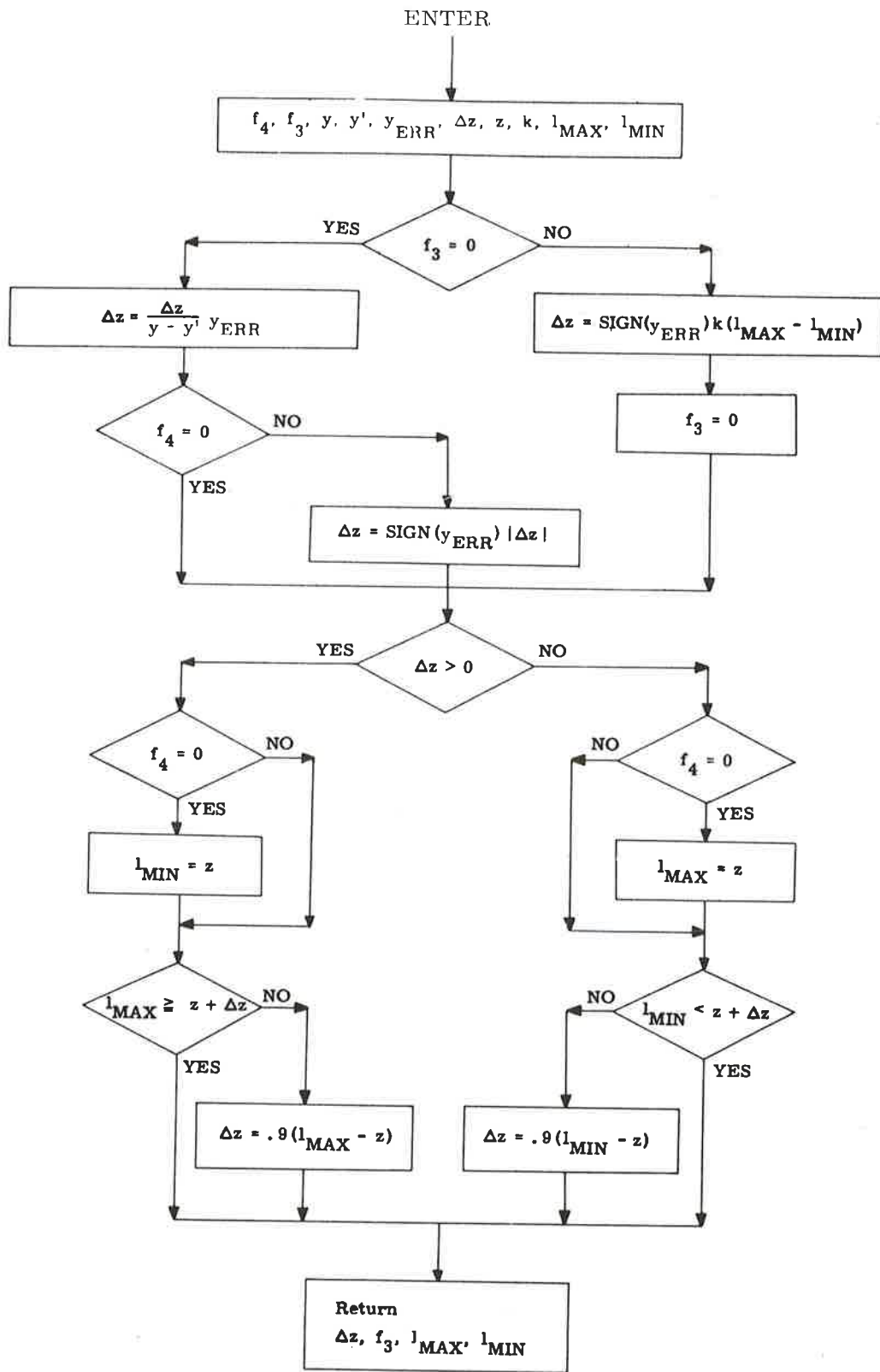


Figure 5.10-3. Iterator Subroutine

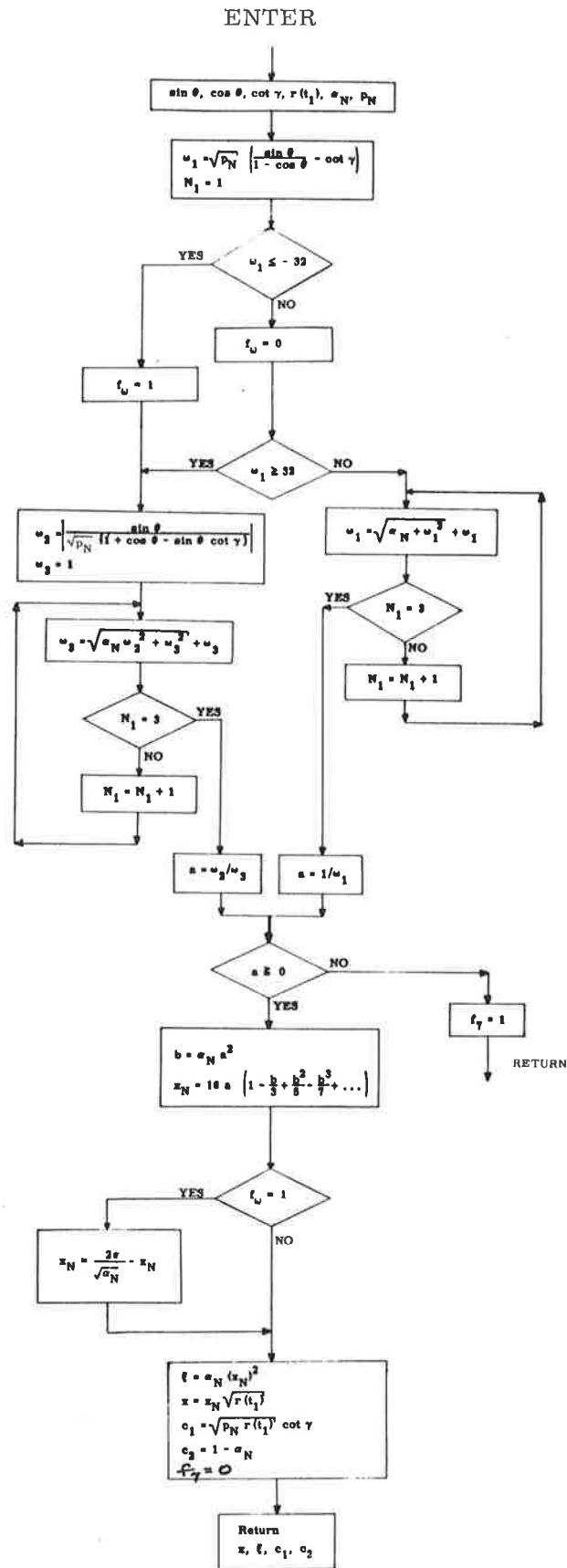


Figure 5.10-4. Universal Variable Subroutine

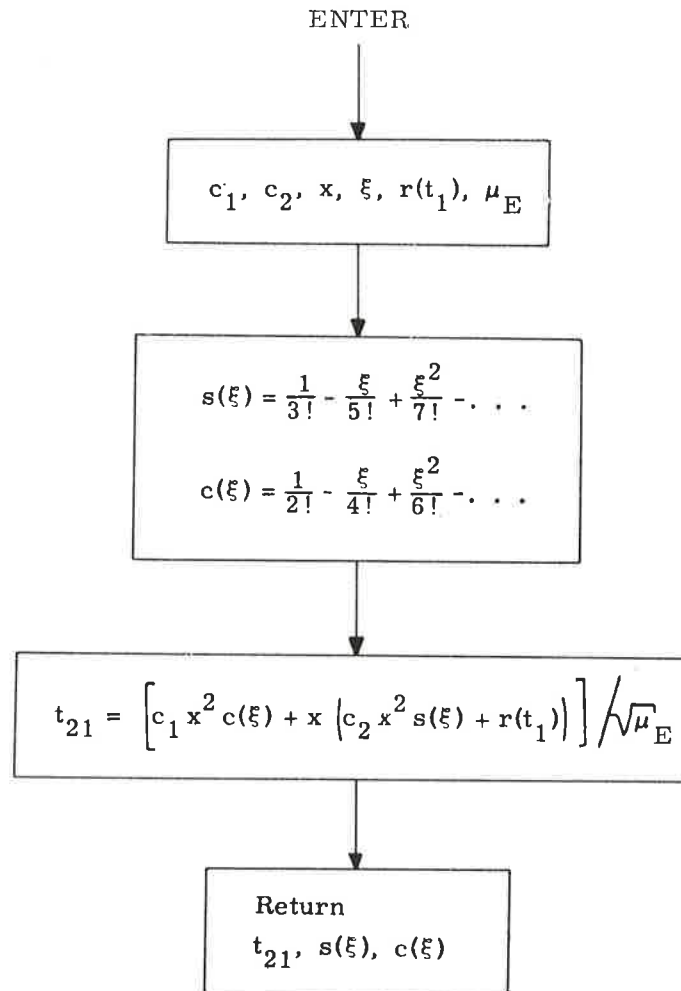


Figure 5.10-5. Kepler Equation Subroutine

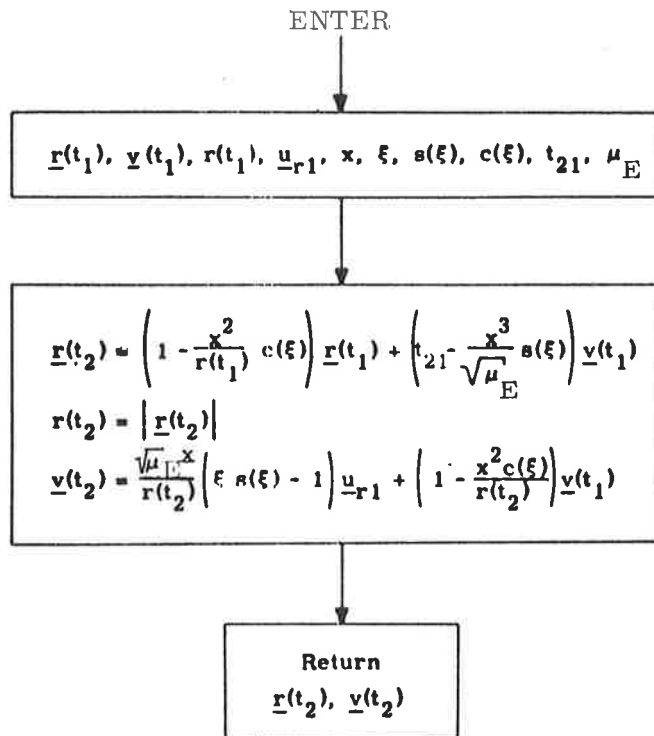


Figure 5.10-6. State Vector Subroutine

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. Two different formulations are required depending upon 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.

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 can be either conic or precision depending on an input parameter (namely, the number of offsets). In addition, for a precision trajectory, 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 that 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-deg 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-deg transfer direction, and whose cone angle is specified by the user.

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

5. 5. 11. 1 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° , 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).
ϵ	Cone Angle of a cone whose vertex is the coordinate origin and whose axis is the 180° transfer direction (i. e., the negative initial position direction). The cone angle ϵ 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 γ]	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.

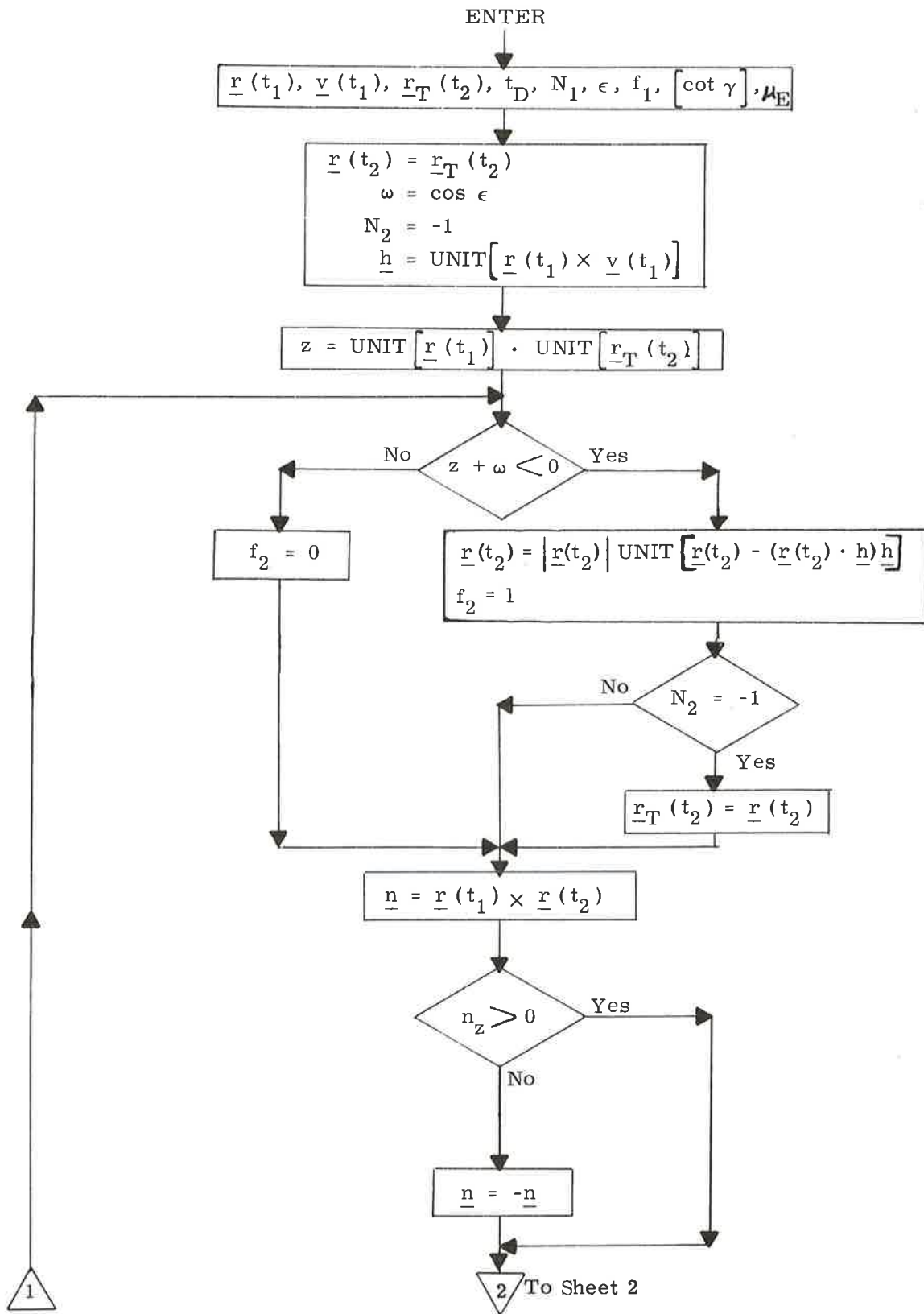


Figure 5.11-1. Initial Velocity Subroutine (Sheet 1 of 2)

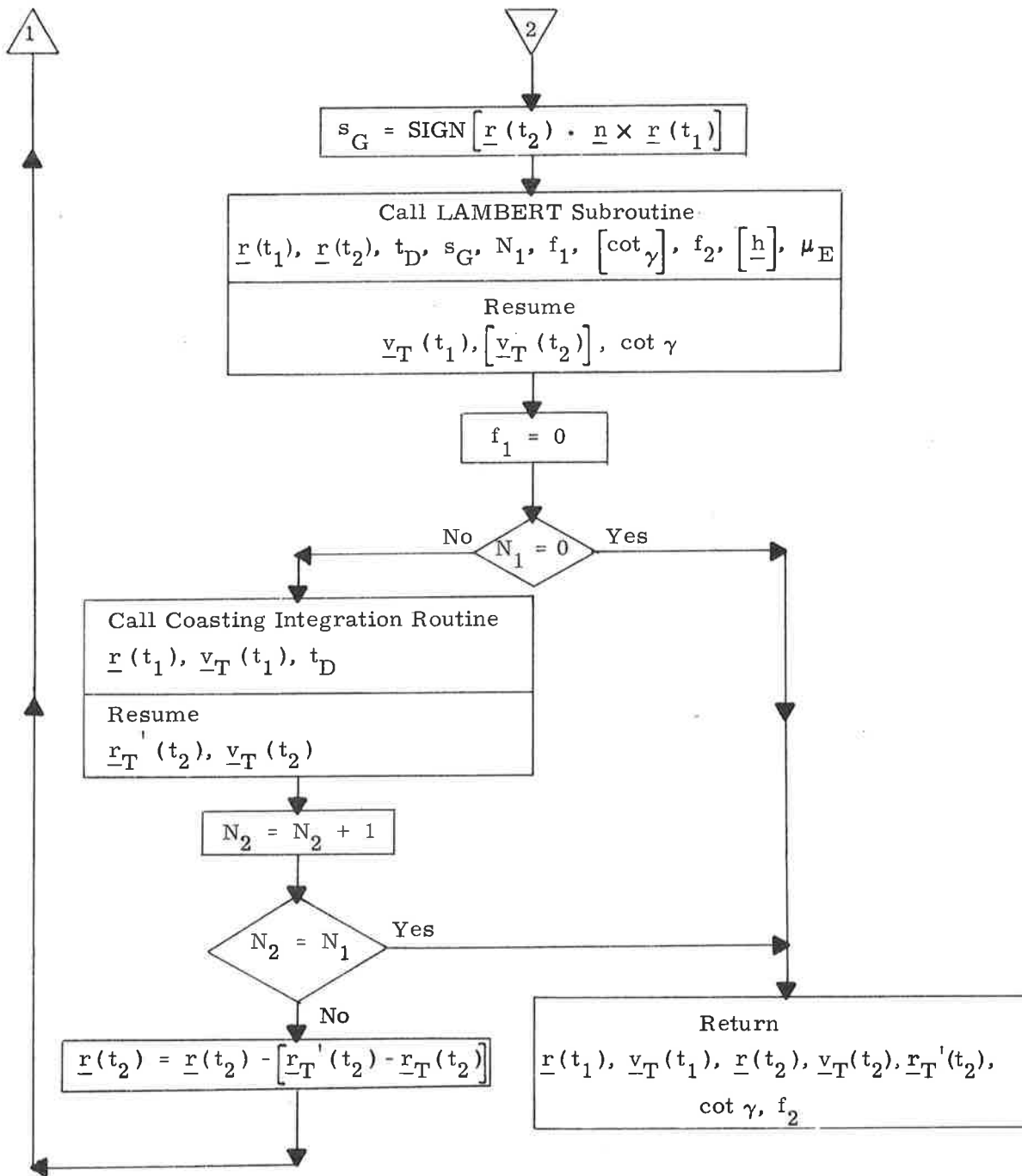


Figure 5.11-1. Initial Velocity Subroutine (Sheet 2 of 2)

5. 5. 12 Paragraph 5. 5. 12 has been deleted.

5. 5. 13 LOCSAM Subroutine

The LOCSAM Subroutine computes the lines-of-sight of the sun and earth with respect to the spacecraft in the Basic Reference Coordinate System. These data are used by the IMU alignment programs whenever the astronaut elects to sight on the sun or earth, instead of a star, for purposes of IMU alignment. The data are also used by the Star Selection Routine (paragraph 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 that 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 and \underline{u}_E specifying the lines-of-sight to the sun and earth, respectively, in the Basic Reference Coordinate System are computed as follows:

$$\underline{u}_S = \underline{u}_{ES} \quad (5. 13. 1)$$

$$\underline{u}_E = -\text{UNIT}(\underline{r}_C) \quad (5. 13. 2)$$

where E and S, respectively, denote the earth and sun; \underline{r}_C is the position vector of the CSM, and \underline{u}_{ES} is the unit position vector of the sun with respect to the earth obtained from the Solar Ephemerides Subroutine of paragraph 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 and earth are computed as follows:

$$c_S = \cos 15^\circ \quad (5. 13. 3)$$

$$c_E = \cos \left[5^\circ + \sin^{-1} \left(\frac{R_E}{r_C} \right) \right] \quad (5. 13. 4)$$

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; and R_E is the equatorial radius (6378. 166 km) of the earth.

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

$$\underline{a} = \underline{u}_S \times \underline{P}_{Ec1} + \frac{\underline{v}_C}{c} \quad (5.13.5)$$

where \underline{u}_S is as calculated by Equation (5.13.1); \underline{v}_C is the velocity of the CSM with respect to the earth body; c is the speed of light; and

$$\begin{aligned} \underline{P}_{Ec1} &= (\text{aberration constant}) [0, -\sin(\text{obliquity}), \cos(\text{obliquity})] \\ &= c_A (0, -K_3, +K_1) \end{aligned} \quad (5.13.6)$$

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 Figure 5.14-1.

5.5.15 Time of Longitude Program (P29)

The Time of Longitude Program (P29) computes the time of the first crossing of a specified longitude after a specified base time, as well as the corresponding latitude, for earth orbit. The program is depicted in Figure 5.15-1.

Inputs to the program consist of the base time t_0 , the desired longitude ϕ_d , and a vehicle option (CSM or WS). The base time is measured in terms of Ground Elapsed Time, and is not to be confused with the t_0 defined in paragraph 5.1.5.5. The desired longitude is measured positive in an eastward direction from the conventional zero meridian of the earth; west longitudes can be specified negatively, if desired.

The permanent state vector corresponding to specified vehicle is transferred to a temporary location and precision-integrated to the base time t_0 by the Coasting Integration Routine. Let \underline{r}_0 and \underline{v}_0 be the position and velocity vectors at the time t_0 . It is re-emphasized that the program finds the Ground Elapsed Time of the first crossing after the base time t_0 ,—regardless of where either vehicle is at the "present" time—provided that \underline{r}_0 and \underline{v}_0 define an elliptic orbit.

The unit angular momentum vector $\underline{\mu}_n$ is formed, and used to calculate a supplementary unit vector $\underline{\mu}_s$ that will be needed later to determine whether a transfer is through θ or $360-\theta$ degrees. The direction of the vector $\underline{\mu}_n$ is then placed in the same hemisphere with the earth's north polar unit axis $\underline{\mu}_Z$. The unit local east vector $\underline{\mu}_E$ at the base position \underline{r}_0 , and the unit vector $\underline{\mu}_c$ normal to both $\underline{\mu}_E$ and $\underline{\mu}_Z$ (and pointing to the center or $\underline{\mu}_Z$ axis) are calculated.

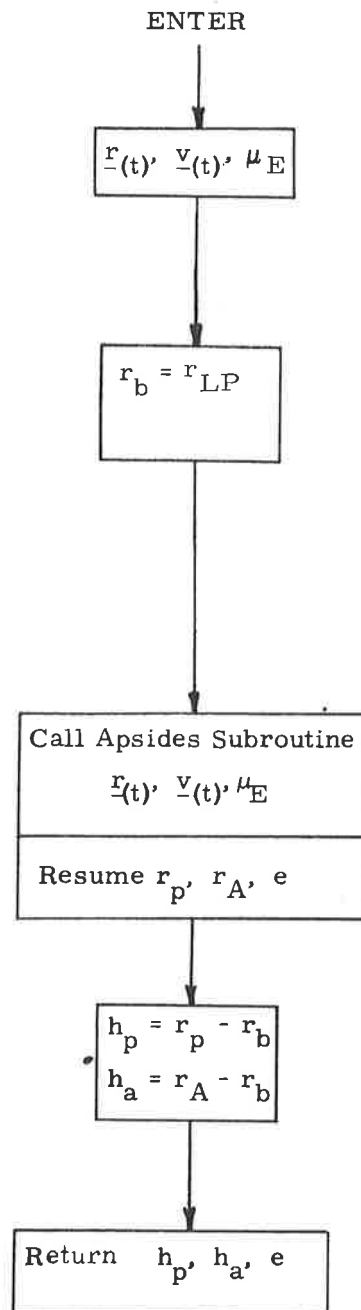


Figure 5.14-1. Pericenter-Apocenter Subroutine

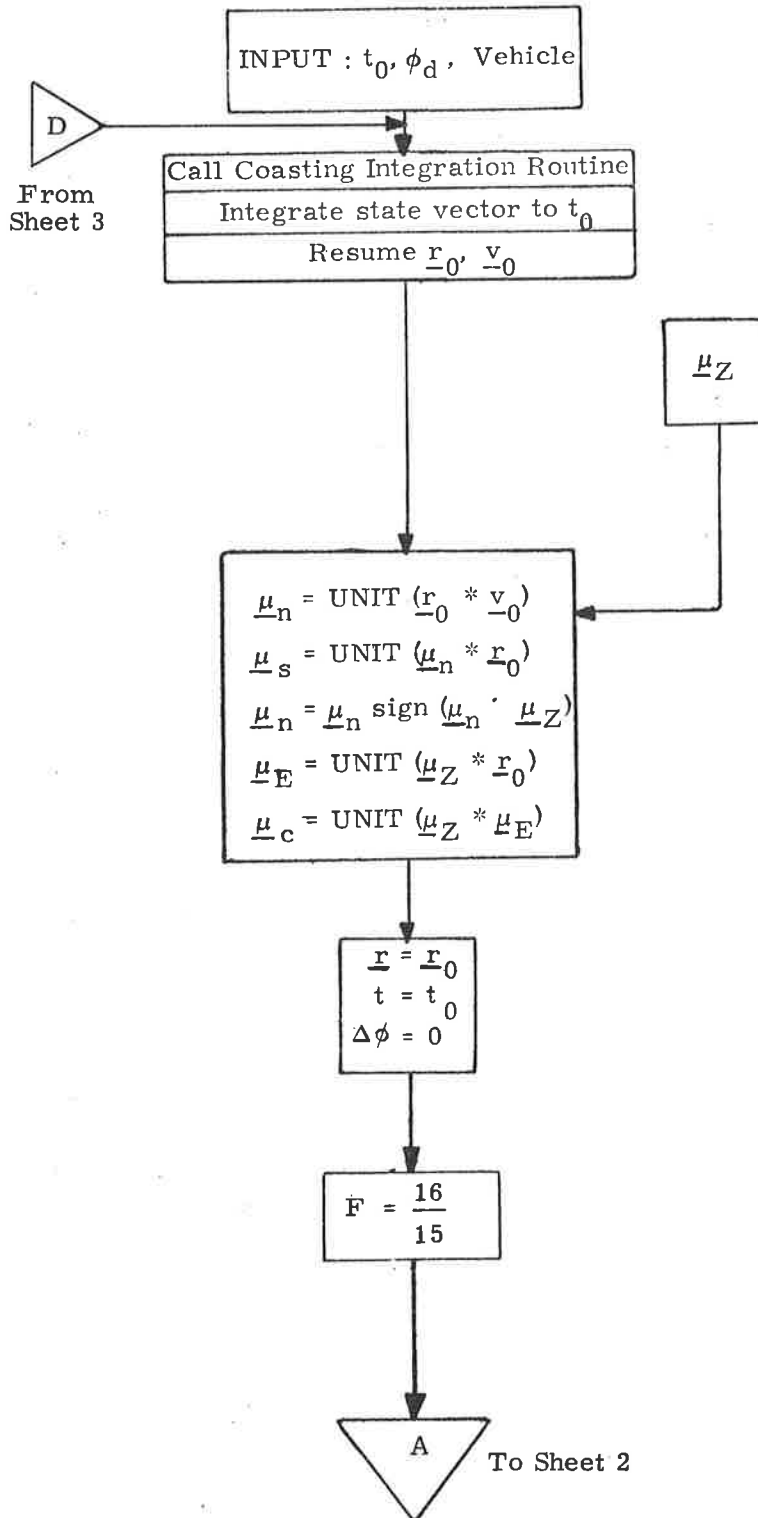


Figure 5, 15-1. Time of Longitude Program (P29) (Sheet 1 of 3)

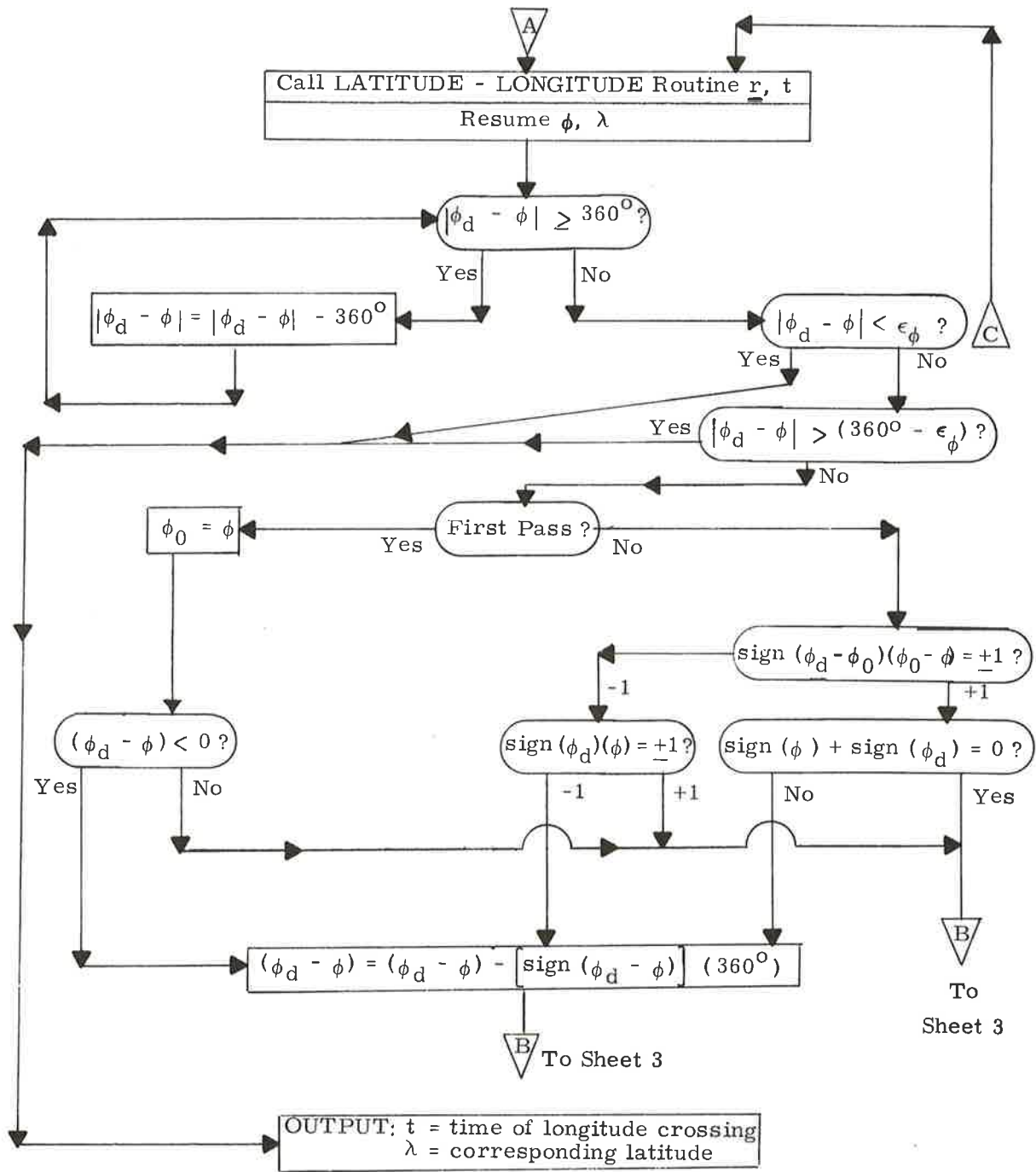


Figure 5.15-1. Time of Longitude Program (P29) (Sheet 2 of 3)

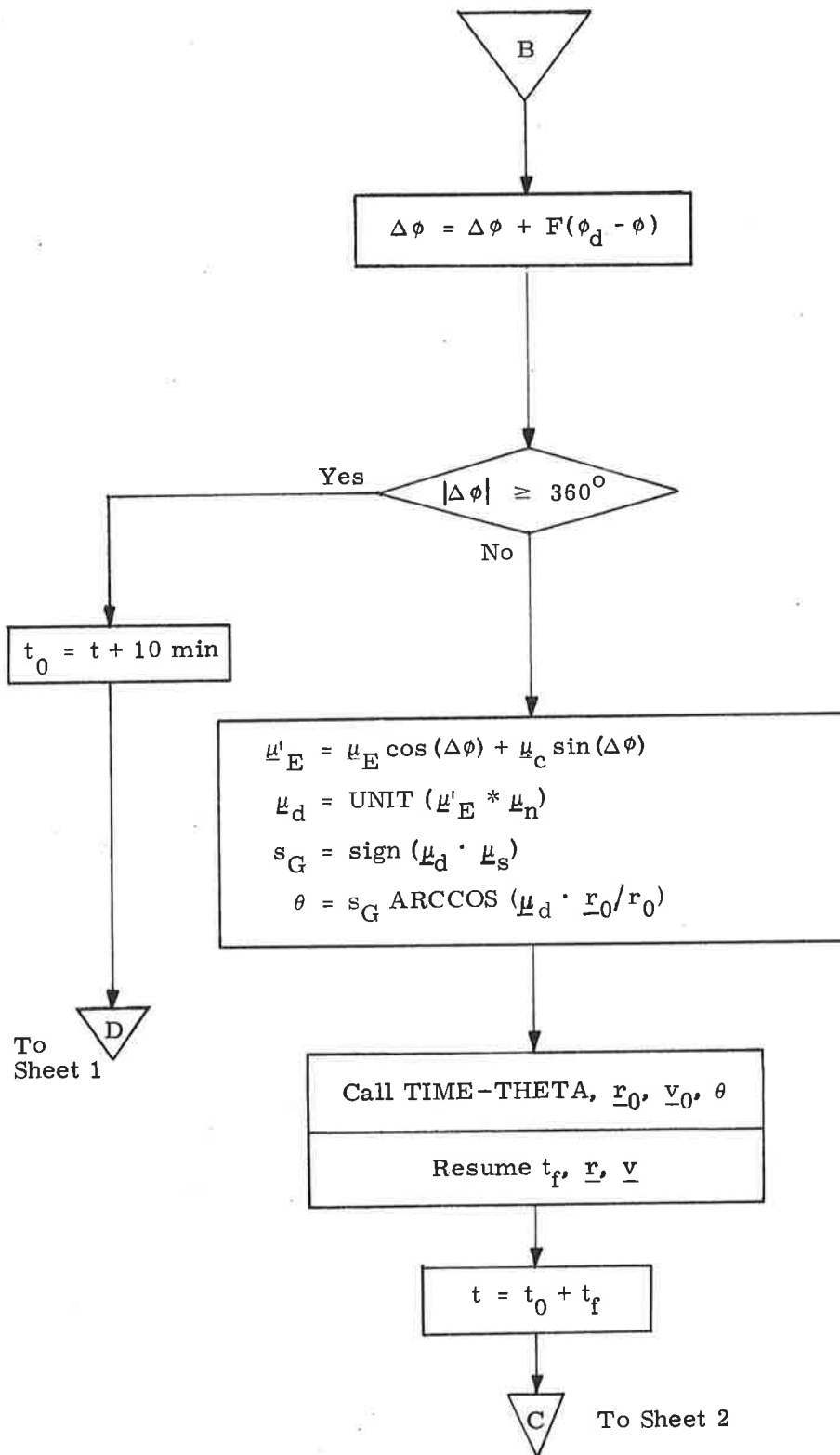


Figure 5.15-1. Time of Longitude Program (P29) (Sheet 3 of 3)

The iteration variables, \underline{r} , t , and $\Delta\phi$ correspond to the estimate on the latest iteration of the position, time, and longitude difference from the longitude of \underline{r}_0 , of the spacecraft at the desired longitude; these variables are initialized to start the iteration process from the base point \underline{r}_0 . The factor 16/15 reduces the number of iterations required for convergence of the procedure by approximating the ratio of the number of revolutions of the spacecraft with respect to inertial space to the number with respect to a fixed longitude in equal time periods. The value corresponds to an earth-orbiting spacecraft with a period of 90 minutes.

The iteration loop is entered with a call to the Latitude-Longitude Routine, which transforms the inertial position vector \underline{r} at the t into body-fixed spherical coordinates so that its longitude can be compared with the desired longitude. If the two longitudes are within the tolerance ϵ_ϕ , the iteration process has converged and the corresponding time t and latitude λ are output for display.

If the two longitudes are not within the tolerance ϵ_ϕ , a new current iterative approximation $\Delta\phi$ of the longitude difference between the longitude of \underline{r}_0 and the desired longitude is calculated. Because $\Delta\phi$ must always be positive for the factor F to correctly compensate (reflecting the prograde motion around the earth) and because the angle values used for the calculations are modulo 360 deg, a complicated series of tests is necessary to determine whether 360 deg should, or should not, be added or subtracted to $(\phi_d - \phi)$. After the first pass, the tests pick out the eight cases in which a 360-deg adjustment is necessary from the 24 possible arrangements of ϕ , ϕ_d , ϕ_0 , and 0. These tests hold for all longitudes between -360 deg and +360 deg.

After the new current estimate of $\Delta\phi$ has been determined, the unit local east vector $\underline{\mu}_E^1$ at the current iterative approximation of the desired longitude is calculated, and from this is obtained the current iterative unit position vector $\underline{\mu}_d$, which yields the current approximation of the transfer angle θ . Using the conic Time-Theta Routine, the base state vector $(\underline{r}_0, \underline{v}_0)$ is updated through θ , to a new terminal position \underline{r} , with which the iteration loop is re-entered.

BLANK

5.6 GENERAL SERVICE ROUTINES

5.6.1 General Comments

The routines presented in this section include the following general-service functions:

1. IMU alignment modes
2. Basic coordinate transformations
3. Computer initialization procedures
4. Special displays that can be called by the astronaut
5. Automatic optics positioning.

5.6.2 IMU Alignment Modes

5.6.2.1 IMU Orientation Determination Program (P51)

The IMU Orientation Determination Program (P51) is used during free fall to determine the present IMU stable-member orientation, with respect to the Basic Reference Coordinate System, by sighting on two navigation stars or known celestial bodies with the sextant (SXT) or the scanning telescope (SCT). When the CSM is docked to the workshop, the crew can also use two other sighting-data sources—the acquisition sun sensor and the star tracker of the Apollo telescope mount (ATM). The astronaut indicates the sighting data source at the time he identifies the celestial body he is using with the data source. It should be noted that any two of the four data sources can be used in a single alignment.

At the start of program P51, if the astronaut plans to use the CSM optics (SXT or SCT) and the CSM is not docked to the workshop, he can acquire the desired celestial bodies by maneuvering the spacecraft until they are visible in the instrument. During this acquisition phase, he monitors the FDAI Ball to avoid IMU gimbal lock. If he desires, he can also coarse align the IMU gimbal angles to zero.

After establishing satisfactory IMU gimbal angles, the astronaut performs the following on each of two celestial bodies with whatever sighting data sources he plans to use:

1. When the astronaut depresses the MARK pushbutton, the Sighting Mark Routine (R53 of Section 4) records the time of measurement and the CSM optics and IMU gimbals angles. If the SXT or SCT is being used, the astronaut marks when he has centered the instrument on the celestial body. If the ATM sun sensor is being used, the astronaut marks when he is satisfied that the workshop is in the solar inertial attitude, which usually implies that the sensor is locked onto the sun. If he is using the ATM star tracker, the astronaut marks at the same time he records the star-tracker gimbal angles for later entry in the computer.

2. After the mark is made, the astronaut identifies the celestial body and the sighting data source by using a celestial body code and a source code. Separate celestial body code numbers are provided for the 37 navigation stars, the sun, the earth, and a general celestial body (referred to as the planet code). Separate source code numbers are provided for CSM optics (SXT or SCT), the ATM sun sensor, and the ATM star tracker. If he selects the ATM star tracker source code, a request will be made for the star-tracker gimbal angles recorded previously.

3. The unit vector \underline{u}'_{CB} defining the direction of the celestial body as a result of the measurement with the selected sighting data source is obtained in the IMU Stable Member Coordinate System as follows:

If CSM optics (SXT or SCT) was used, then:

$$\underline{u}'_{CB} = [NBSM] [SBNB] \begin{bmatrix} \sin TA \cos SA \\ \sin TA \sin SA \\ \cos TA \end{bmatrix} \quad (6.2.1)$$

where $[NBSM]$ is the transformation matrix in paragraph 5.6.3.2.1 using the IMU gimbal angles stored for the optical mark; $[SBNB]$ is the matrix in paragraph 5.6.3.1.1; and SA and TA are, respectively, the shaft and trunnion angles of the SXT or SCT stored for the optical mark.

If the ATM sun sensor was used, then:

$$\underline{u}'_{CB} = [NBSM] \underline{z}_{ATM} \quad (6.2.2)$$

where $[NBSM]$ is the transformation matrix in paragraph 5.6.3.2.1

using the IMU gimbal angles stored for the mark; and \underline{z}_{ATM} is a unit vector, previously computed and stored by the ATM Orientation Determination Program (P50, described in paragraph 5.6.2.5), which represents the direction of the +Z-axis of the ATM sensor coordinate system in navigation-base coordinates.

If the ATM star tracker was used, then:

$$\underline{u}'_{CB} = [NBSM] [ATMNB] \begin{bmatrix} \cos\psi_1 & \sin\psi_3 \\ -\cos\psi_1 & \cos\psi_3 \\ -\sin\psi_1 & \end{bmatrix} \quad (6.2.3)$$

where $[NBSM]$ is the matrix in paragraph 5.6.3.2.1 using the IMU gimbal angles stored for the mark; $[ATMNB]$ is the transformation matrix described in paragraph 5.6.3.4, and ψ_1 and ψ_3 are, respectively, the inner and outer gimbal angles of the ATM star tracker recorded at the time of the mark.

4. After \underline{u}'_{CB} is computed, the subroutine LOCSAM (paragraph 5.5.13) is used to compute the correction for aberration of light, and the line-of-sight vectors of the sun and earth in the Basic Reference Coordinate System for the time of the mark. If the sun or earth code was selected, the vector for that body will be used later.
5. The unit vector \underline{u}^*_{CB} defining the direction of the indicated celestial body in the Basic Reference Coordinate System is then determined as follows. If the celestial body is one of 37 navigation stars shown in Figure 9.1-1, the vector is obtained from the star catalog in fixed memory. If the celestial body is the sun or earth, the vector previously computed by LOCSAM for that body is used. If the planet code is selected, a request will be made for the astronaut to load the vector components of the desired body.
6. The vector \underline{u}^*_{CB} is then corrected for aberration of light as follows:

$$\underline{u}_{CB} = \text{UNIT} (\underline{u}_{CB}^* + \underline{a}) \quad (6.2.4)$$

where \underline{a} is the aberration correction vector computed by the subroutine LOCSAM when this subroutine was called after the celestial body sighting, and \underline{u}_{CB}^* and \underline{u}_{CB} are, respectively, the uncorrected and corrected unit vectors to the celestial body in basic reference coordinates. Although this correction is correct only for distant objects such as stars, it is also applied, for program simplification, to the unit vectors of the sun, earth, or a planet, if any of these bodies are being used. It should be noted that the aberration correction is negligible in comparison to the errors in sighting on the sun or earth.

When the above steps have been performed for two celestial bodies, the computer has the unit line-of-sight vectors for the two bodies in both IMU stable-member and basic-reference coordinates. Let \underline{u}'_{CBA} and \underline{u}'_{CBB} be the unit vectors for the two celestial bodies (A and B), which are obtained with Equations (6.2.1) through (6.2.3), and let \underline{u}_{CBA} and \underline{u}_{CBB} be the unit vectors for the same bodies obtained with Equation (6.2.4). At this point in the program, the Sighting Data Display Routine (R54) computes the angle between the unit line-of-sight vectors (\underline{u}'_{CBA} and \underline{u}'_{CBB}) obtained for the two bodies in stable-member coordinates and the angle between the corresponding unit line-of-sight vectors (\underline{u}_{CBA} and \underline{u}_{CBB}) for the two bodies in basic-reference coordinates. The magnitude of the difference between the two angles and the angle between \underline{u}'_{CBA} and \underline{u}'_{CBB} are displayed to the astronaut and either he accepts the results or he repeats the IMU orientation determination process.

If he accepts the results of the Sighting Data Display Routine, the unit vectors \underline{u}'_{CBA} , \underline{u}'_{CBB} , \underline{u}_{CBA} , and \underline{u}_{CBB} are used to determine the present stable-member orientation and REFSMMAT, using the procedure given in paragraph 5.6.3.3.1.

5.6.2.2 IMU Realignment Program (P52)

The IMU Realignment Program (P52) is used during free fall to re-align the IMU to its presently assumed orientation or to align it from a known orientation to one of the desired orientations given in paragraph 5.6.3.3 of this document,

and in P52 of GSOP Section 4. This alignment is made by sighting on two navigation stars or known celestial bodies with the sextant (SXT) or the scanning telescope (SCT). When the CSM is docked to the workshop, the crew can also use the ATM sun sensor and star tracker to obtain sighting data, as explained in paragraph 5.6.2.1.

At the beginning of program P52 the astronaut indicates which of the following stable-member orientations he desires:

1. Preferred—for thrusting maneuvers
2. Nominal—for alignment with respect to local vertical
3. REFSMMAT—for realignment to presently assumed orientation.

The preferred and nominal orientations are defined in paragraph 5.6.3.3. If the astronaut selects the nominal orientation, it is computed by P52 in the manner shown in the referenced paragraphs, and in P52 of GSOP Section 4. The preferred orientation must be computed before entering P52. Whenever the astronaut selects the preferred or nominal orientation, the program also computes and displays the IMU gimbal angles for the desired stable-member orientation using the present vehicle attitude. These angles are computed by the CALCGA Routine described in paragraph 5.6.3.2.2. The inputs to this routine are the unit vectors (\underline{u}_{XSM} , \underline{u}_{YSM} , \underline{u}_{ZSM}) defining the desired stable-member axes with respect to the Basic Reference Coordinate System, and the unit vectors (\underline{x}_{NB} , \underline{y}_{NB} , \underline{z}_{NB}) defining the present navigation-base axes with respect to the Basic Reference Coordinate System; the latter are computed as follows:

$$\begin{aligned}\underline{x}_{NB} &= [\text{REFSMMAT}]^T \underline{x}'_{NB} \\ \underline{y}_{NB} &= [\text{REFSMMAT}]^T \underline{y}'_{NB} \\ \underline{z}_{NB} &= [\text{REFSMMAT}]^T \underline{z}'_{NB}\end{aligned}\tag{6.2.5}$$

where $[\text{REFSMMAT}]^T$ is the transpose of the present $[\text{REFSMMAT}]$ and \underline{x}'_{NB} , \underline{y}'_{NB} , and \underline{z}'_{NB} define the navigation base-axes with respect to the present Stable Member Coordinate System and are computed by the CALCSMSC Routine described in paragraph 5.6.3.2.5.

If the computed IMU gimbal angles are unsatisfactory, the astronaut maneuvers the vehicle to a more suitable attitude and has the program recompute

and display the new gimbal angles. Once satisfactory angles are displayed, the astronaut keys PRO; he is then requested to indicate whether the IMU is to be aligned to the desired orientation by using the Coarse Alignment Routine (R50 of Section 4 GSOP) or by torquing the gyros. If he chooses to have the gyros torqued to the desired orientation, the gyro-torquing angles are computed by the CALCGTA Routine (paragraph 5.6.3.2.3). The inputs to this routine are the unit vectors (\underline{x}_D , \underline{y}_D , \underline{z}_D) defining the desired stable-member axes with respect to the present Stable Member Coordinate System; they are computed as follows:

$$\begin{aligned}\underline{x}_D &= [\text{REFSMMAT}] \underline{u}_{\text{XSM}} \\ \underline{y}_D &= [\text{REFSMMAT}] \underline{u}_{\text{YSM}} \\ \underline{z}_D &= [\text{REFSMMAT}] \underline{u}_{\text{ZSM}}\end{aligned}\tag{6.2.6}$$

where $\underline{u}_{\text{XSM}}$, $\underline{u}_{\text{YSM}}$, and $\underline{u}_{\text{ZSM}}$ were previously defined. During the period when the gyros are being torqued, the IMU gimbal angles are displayed to the astronaut so that he can avoid gimbal lock by maneuvering the spacecraft. After the gyro-torquing process, the astronaut either terminates program P52 or performs a fine alignment using sighting data on two celestial bodies.

If the astronaut elects to have the IMU aligned to the desired orientation via the Coarse Alignment Routine (R50), R50 will first compute the IMU gimbal angles for the desired stable-member orientation, just as done previously. The routine then determines if all the required gimbal angle changes are less than, or equal to, one degree. If so, the IMU is automatically gyro torqued to the desired orientation using torquing angles computed as follows.

The unit vectors $\underline{x}_{\text{NB}}$ and $\underline{y}_{\text{NB}}$ defining the orientations of the navigation base x and y axes with respect to the present IMU Stable Member Coordinate System are computed by the CALCSMSC Routine (paragraph 5.6.3.2.5). The unit vectors $\underline{x}'_{\text{NB}}$ and $\underline{y}'_{\text{NB}}$ defining the orientations of the same two axes with respect to the desired IMU Stable Member Coordinate System are computed as follows:

$$\begin{aligned}\underline{x}'_{\text{NB}} &= [\text{NBSM}]_D \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ \underline{y}'_{\text{NB}} &= [\text{NBSM}]_D \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}\end{aligned}\tag{6.2.7}$$

where $[NBSM]_D$ is the matrix for transforming vectors from navigation-base to desired stable-member coordinates using the desired IMU gimbal angles computed initially by Routine R50. Once the vectors \underline{x}_{NB} , \underline{y}_{NB} , \underline{x}'_{NB} , and \underline{y}'_{NB} have been computed, they are used in the AXISGEN Routine (paragraph 5.6.3.2.4) to compute the vectors \underline{x} , \underline{y} , and \underline{z} defining the orientations of the desired stable-member axes with respect to the present IMU Stable Member Coordinate System; these vectors are then used by the CALCGTA Routine to compute the gyro-torquing angles.

If R50 determines that at least one gimbal angle must be changed by more than one degree, the IMU is first coarse aligned by driving the gimbals to the desired values, and then gyro torqued to trim any residual errors in obtaining the desired gimbal angles. The gyro-torquing angles for this case are computed in the same manner as for the other case. It should be noted that any change in vehicle attitude during the time the gimbals are first driven to the desired values will introduce error in the alignment.

Afterwards, the astronaut can maneuver the CSM to a desired attitude for celestial-body acquisition if he plans to use the CSM optics and if the CSM is not docked to the workshop. He can use the Star Selection Routine (paragraph 5.6.4) to establish which celestial bodies he will use with the CSM optics or he can make his own selection. If he uses the Star Selection Routine, program P52 will call the LOCSAM Subroutine (paragraph 5.5.13) before calling the Star Selection Routine, since LOCSAM computes the directions and the occultation cones of the sun and earth for the occultation tests in the Star Selection Routine. When LOCSAM is called for this purpose, the specified input time is the present time plus an additional amount (T_{SS}) to ensure that the time used in computing the LOS vectors for the sun and earth is near the middle of the sighting-mark process. The time increment T_{SS} is based upon estimates of the time required to do the Star Selection Routine and to perform the optical sightings.

If the Star Selection Routine is unable to find two satisfactory stars at the present vehicle attitude, the astronaut either repeats the above process of changing the vehicle attitude and using the Star Selection Routine, or selects his own celestial bodies.

After making the selection, the astronaut performs the following steps on each of the two celestial bodies:

1. The astronaut identifies the celestial body and the sighting data source by using a celestial body code and a source code, as explained in P51 (paragraph 5.6.2.1). If the Star Selection Routine was used earlier, the celestial body code number for one of the stars found by that routine will be displayed for his perusal.
2. The LOCSAM Subroutine is then used to compute the correction for aberration of light and the line-of-sight vectors of the sun and earth in the Basic Reference Coordinate System for the current time. If the sun or earth code was selected, the vector for that body will be used later.
3. The unit vector \underline{u}_{CB} defining the direction of the indicated celestial body in the Basic Reference Coordinate System is then determined as in steps 5 and 6 of P51. This vector will be used to designate the CSM optics (SXT and SCT) if the Auto Optics Positioning Routine (R52) is used in step 4 below.
4. If the CSM optics source code was selected in step 1, the Auto Optics Positioning Routine is used to drive the CSM optics to the LOS of the celestial body. At the same time, the Sighting Mark Routine (R53) is called by routine R52 to request a mark after the astronaut has manually centered the SXT or SCT on the celestial body. If the ATM sun sensor or star tracker source code was selected in step 1, only the Sighting Mark Routine (R53) will be called at this point. The time when a mark is made for either ATM source is given in step 1, paragraph 5.6.2.1 for P51.
5. After the mark is made, steps 2 through 6, shown in paragraph 5.6.2.1, are performed, with the exception that the unit vector \underline{u}_{CB} (see Equation (6.2.4) in step 6 of paragraph 5.6.2.1) is expressed in terms of the desired IMU Stable Member Coordinate System instead of the Basic Reference Coordinate System as follows:

$$\underline{u}_{CB} = \text{UNIT} \left\{ [\text{REFSMMAT}]_D (\underline{u}_{CB}^* + \underline{a}) \right\} \quad (6.2.8)$$

where \underline{a} is the aberration correction vector computed by the LOCSAM Subroutine, and $[\text{REFSMMAT}]_D$ is the matrix (paragraph 5.6.3.3) for transforming vectors from the Basic Reference Coordinate System to the desired IMU Stable Member Coordinate System.

After the above steps have been performed on two celestial bodies, the computer has the unit line-of-sight vectors for the two bodies in both present and desired IMU stable-member coordinates. Let \underline{u}'_{CBA} and \underline{u}'_{CBB} be the unit vectors for the two bodies (A and B), which are obtained with Equations (6.2.1) through (6.2.3) and let \underline{u}_{CBA} and \underline{u}_{CBB} be the unit vectors for the same bodies obtained with Equation (6.2.8). At this point in the program the Sighting Data Display Routine (R54 of Section 4) computes the angle between \underline{u}_{CBA} and \underline{u}_{CBB} , and the angle between \underline{u}'_{CBA} and \underline{u}'_{CBB} , and displays the magnitude of the difference between the two angles and the angle between \underline{u}_{CBA} and \underline{u}_{CBB} to the astronaut. If he accepts the results, the four vectors are used by the AXISGEN Routine of paragraph 5.6.3.2.4 to compute the desired stable-member axes with respect to the present Stable Member Coordinate System, which are used in the Gyro Torquing Routine (R55 of Section 4). Routine R55 computes the gyro-torquing angles required to drive the IMU stable member to the desired orientation by using the above vectors in the CALCGTA Routine of paragraph 5.6.3.2.3. The gyro-torquing angles are displayed to the astronaut so that he can decide whether to have the gyros torqued through these angles or not. If he is not satisfied with the results of the Sighting Data Display Routine or the Gyro Torquing Routine, he can repeat the optical sightings without terminating the program.

It should be noted that the ground can uplink, to the CMC, the present stable-member orientation or a desired stable-member orientation. If the present orientation is uplinked it is done by transmitting a REFSMMAT to the CMC. Under normal circumstances, however, it is not desirable for the ground to uplink the present stable-member orientation since this orientation should be determined by the CSM GNCS. If ground desires an orientation different from the present orientation, however, this desired orientation can be transmitted as a preferred orientation. By treating a desired stable-member orientation in this manner, program P52 will be able to correct for any large differences between the present and desired orientations by coarse alignment. In addition, this approach avoids the introduction of orientation errors that affect celestial-body acquisition.

5.6.2.3 Backup IMU Orientation Determination Program (P53)

The Backup IMU Orientation Determination Program (P53) is used during free fall to determine the present IMU stable-member orientation with respect to the Basic Reference Coordinate System by sighting on two navigation stars or known celestial bodies with some optical device other than the sextant

or scanning telescope. An optical device that is a backup to the normal CSM optics is the Crew Optical Alignment Sight (COAS). Like program P51 this program can also use the ATM sun sensor and star tracker to obtain sighting data. This program is identical to program P51 except that the Alternate LOS Sighting Mark Routine (R56 of Section 4) is used in place of the Sighting Mark Routine (R53) to sight on the celestial bodies. When P53 calls R56, R56 requests the astronaut to load the coordinates of the backup optical device before performing the optical sighting. These coordinates are the equivalent SXT shaft and trunnion angles of the device that would be indicated for the SXT if it were possible to direct the SXT along the same direction as the optical device. If the ATM sun sensor or star tracker is being used, these coordinates are irrelevant and will not be used. When an optical sighting is performed using Routine R56, the measurement time, the IMU gimbal angles, and the equivalent shaft and trunnion angles of the backup device are stored by the computer. After the optical sighting the astronaut identifies the celestial body and the sighting data source via a celestial body code and a source code, as explained in P51, except that the source code for CSM optics now indicates the use of a backup optical device other than the ATM sensors. If the source code for the ATM star tracker is selected, a request is made for the star-tracker gimbal angles recorded at the time of the mark.

The manner in which the IMU orientation is determined with the sighting data is the same as that given for program P51, with the exception that the angles SA and TA of Equation (6. 2. 1) are those indicated for the backup device whenever one is used.

5.6.2.4 Backup IMU Realignment Program (P54)

The Backup IMU Realignment Program (P54) is used during free fall to realign the IMU to its presently assumed orientation, or to align it from a known orientation to one of the desired orientations given in paragraph 5.6.3.3 and in P54 of Section 4 by sighting on two navigation stars or known celestial bodies with some optical device other than the sextant or scanning telescope. For example, the optical Crew Optical Alignment Sight (COAS) can be used. Like program P52 this program can also use the ATM sun sensor and star tracker to obtain sighting data. This program is identical to program P52 except that the Alternate LOS Sighting Mark Routine (R56 of Section 4) is used in place of the Sighting Mark Routine (R53) to sight on the celestial bodies. In addition, the Auto Optics Positioning Routine (R52) is not used in this program. A description of the Alternate LOS Sighting Mark Routine is given in Section 4 and in paragraph 5.6.2.3.

The manner in which the astronaut indicates the celestial body and the source used to sight on it is the same as that given in program P52 except that the source code for CSM optics now indicates the use of a backup optical device other than the ATM sensors.

5.6.2.5 ATM Orientation Determination Program

5.6.2.5.1 Introduction. —The ATM Orientation Determination Program (P50) is used to compute and store the orientation of the Apollo Telescope Mount Sensor Coordinate System with respect to the Navigation Base Coordinate System when the CSM is docked to the workshop. This data is required whenever an IMU alignment is performed in P51 or P52 with sighting data from the ATM sun sensor or the ATM star tracker. In addition, the data is required in the ATM Star Tracker Gimbal Angle Program (P55 of paragraph 5.6.2.6) in order to compute the gimbal angles as required by the star tracker to point at a designated celestial body.

The orientation data computed by P50 is in the form of two unit vectors \underline{y}_{ATM} and \underline{z}_{ATM} , which define the respective directions of the +Y-axis and +Z-axis of the ATM Sensor Coordinate System in navigation-base coordinates when the CSM is docked to the workshop. It is assumed that the +Z-axis of the ATM Sensor Coordinate System is pointing in the same direction as the ATM acquisition sun sensor, and that the -Y-axis is pointing in the same direction as the ATM star tracker when the star tracker gimbal angles are zero.

P50 provides three techniques (or options) for computing the two ATM vectors. It is assumed that the expected nominal values of the two vectors will be loaded into erasable memory before launch and that P50 will be used during the mission to make partial or complete updates of the vectors. Whenever new ATM vectors are computed by P50, the equivalent docking angles, which describe the orientation of the ATM Sensor Coordinate System with respect to the Navigation Base Coordinate System, are computed and displayed to the astronaut for his approval. At this point he can either accept or reject the proposed new vectors, or he can load his own docking angle values, which will then be used to compute new ATM vectors for storage. It should be noted that the ATM vector values present in storage before this program is initiated will not be replaced by new values until the astronaut so indicates in the above display.

5.6.2.5.2 Option 1—ATM Vector Determination Using ATM Sun Sensor. —Option 1 is selected by the astronaut when he wants the two ATM vectors \underline{y}_{ATM} and \underline{z}_{ATM} to be updated using only the ATM sun sensor data. It is assumed that the IMU is at a known orientation (i. e., a REFSMMAT is present) and that the ATM is in the solar-inertial attitude—which implies that the ATM sun sensor is pointing at the

sun. At this time, it is possible to determine the direction of the +Z-axis of the ATM Sensor Coordinate System in navigation-base coordinates without using any knowledge of the structural orientation of the ATM with respect to the CSM navigation base (for example, the docking-ring angle). This is accomplished as follows. The Sighting Mark Routine (R53) is used to request a mark from the astronaut. When the astronaut is satisfied that the ATM is in the solar-inertial attitude, he performs a mark, which records the present time and IMU gimbal angles. These data are then used as shown below to obtain the vector \underline{z}'_{ATM} that represents the estimated direction of the +Z-axis of the ATM Sensor Coordinate System in navigation base coordinates.

$$\underline{z}'_{ATM} = [\text{SMNB}] [\text{REFSMMAT}] \underline{u}_{ES} \quad (6.2.9)$$

where $[\text{SMNB}]$ is the matrix in paragraph 5.6.3.2.1 using the IMU gimbal angles recorded for the mark; $[\text{REFSMMAT}]$ is the matrix in paragraph 5.6.3.3; and \underline{u}_{ES} is the unit LOS vector of the sun in basic-reference coordinates computed for the time of the mark by the Solar Ephemerides Subroutine (paragraph 5.5.4).

When the calculation is made, \underline{z}'_{ATM} and the previously stored unit vector \underline{y}_{ATM} are used in the following manner as inputs to the AXISGEN Routine (paragraph 5.6.3.2.4) to obtain a new estimate \underline{y}'_{ATM} of the direction of the +Y-axis of the ATM Sensor Coordinate System in navigation-base coordinates:

$$\begin{aligned} \underline{s}_A &= \underline{z}'_{ATM} \\ \underline{s}_B &= \underline{y}_{ATM} \\ \underline{s}'_A &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ \underline{s}'_B &= \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \end{aligned} \quad (6.2.10)$$

The output of AXISGEN will be \underline{x}'_{ATM} , \underline{y}'_{ATM} , and \underline{z}'_{ATM} , where the latter happens to be the same as the \underline{z}'_{ATM} used in the input. It should be noted that \underline{y}'_{ATM} in this case, is simply \underline{y}_{ATM} corrected in the plane of \underline{z}'_{ATM} and \underline{y}_{ATM} so as to be perpendicular to \underline{z}'_{ATM} . In other words, \underline{y}'_{ATM} represents a partial update of \underline{y}_{ATM} based upon the new estimate (\underline{z}'_{ATM}) of the ATM +Z-axis.

After the unit vectors (\underline{x}'_{ATM} , \underline{y}'_{ATM} , \underline{z}'_{ATM}) have been computed by AXISGEN, they are used with the following unit vectors

$$\underline{x}_{NB} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \underline{y}_{NB} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \text{and} \quad \underline{z}_{NB} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

in the CALCGA Routine of (paragraph 5.6.3.2.2) to compute a set of pseudo-IMU gimbal angles (OGA, IGA, MGA) that describe the orientation of the ATM Sensor Coordinate System with respect to the Navigation Base Coordinate System. These angles would be the same as those indicated by the IMU CDU's if the IMU stable member were at the same orientation, with respect to the navigation base, as the ATM Sensor Coordinate System. This orientation is displayed to the astronaut by three docking angles (α_{ATM} , β_{ATM} , γ_{ATM}), which are related to the pseudo-gimbal angles as follows:

$$\begin{aligned} \alpha_{ATM} &= 180^\circ - \text{OGA} \\ \beta_{ATM} &= \text{IGA} \\ \gamma_{ATM} &= \text{MGA} \end{aligned} \tag{6.2.11}$$

The nominal values of α_{ATM} , β_{ATM} , and γ_{ATM} expected in a mission are 145 deg, 180 deg, and 0 deg, respectively. The angle α_{ATM} represents the value the docking-ring angle should have after the nominal value has been corrected by the angular deviation indicated on the docking-ring scale located between the CSM and the workshop. If the astronaut is satisfied with these angles, he keys PRO, which causes \underline{y}'_{ATM} and \underline{z}'_{ATM} to replace \underline{y}_{ATM} and \underline{z}_{ATM} in storage. If he is not satisfied with the angles, he can load his own values, which will then be used to compute new values of \underline{y}_{ATM} and \underline{z}_{ATM} for storage. These vectors are obtained by first converting the docking angles to the equivalent IMU gimbal angles according to the relationships in Equation 6.2.11, and using the results in the CALCSMSC Routine (paragraph 5.6.3.2.5) to obtain the unit vectors \underline{x}_{NB} , \underline{y}_{NB} , and \underline{z}_{NB} , which represent the directions of the navigation-base axes in the ATM Sensor Coordinate System. The desired vectors are then obtained as follows (the numerical subscript identifies the component of the vector):

$$\underline{y}_{ATM} = \begin{bmatrix} x_{NB2} \\ y_{NB2} \\ z_{NB2} \end{bmatrix} \tag{6.2.12}$$

$$\underline{z}_{\text{ATM}} = \begin{bmatrix} x_{\text{NB3}} \\ y_{\text{NB3}} \\ z_{\text{NB3}} \end{bmatrix}$$

(6.2.12 cont.)

5.6.2.5.3 Option 2-ATM Vector Determination Using ATM Sun Sensor and Star

Tracker.—Option 2 is selected by the astronaut when he wants $\underline{y}_{\text{ATM}}$ and $\underline{z}_{\text{ATM}}$ to be computed using both the ATM sun sensor and the ATM star tracker. Unlike option 1, which updates the ATM sensor orientation in only two dimensions, this option performs a complete update. When using this option it is assumed that a REFSMMAT is present, the ATM is in the solar-inertial attitude, and the star tracker is locked onto a celestial body.

At the beginning of this option the Sighting Mark Routine (R53) makes a request for a single mark, which is made when the astronaut is satisfied that the ATM is in the solar inertial attitude and the ATM star tracker is locked onto a celestial body. At the time of the mark, the astronaut records the star-tracker gimbal angles for later entry in the computer. After the mark, the astronaut indicates the celestial body being tracked by the star tracker and loads the star-tracker gimbal angles recorded during the mark.

In option 2 the method for computing the unit vector $\underline{z}'_{\text{ATM}}$ is the same as that in option 1. (See Eq. 6.2.9.) The AXISGEN Routine is used to compute $\underline{y}'_{\text{ATM}}$. The inputs are

$$\underline{s}_A = \underline{z}'_{\text{ATM}}$$

$$\underline{s}_B = [\text{SMNB}] [\text{REFSMMAT}] \text{UNIT} (\underline{u}_{\text{CB}}^* + \underline{a})$$

(6.2.13)

$$\underline{s}'_A = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\underline{s}'_B = \begin{bmatrix} \cos\psi_1 & \sin\psi_3 \\ -\cos\psi_1 & \cos\psi_3 \\ & -\sin\psi_1 \end{bmatrix}$$

where \underline{u}_{CB}^* is the unit LOS vector of the indicated celestial body in basic-reference coordinates; this vector was either taken from the computer's star catalog in fixed memory, calculated by the computer, or loaded by the astronaut; \underline{a} is the light-aberration correction vector computed by the LOCSAM Subroutine (paragraph 5.5.13); and ψ_1 and ψ_3 are the inner- and outer-gimbal angles of the star tracker, respectively. Although the correction for aberration of light is not required in this program, it is applied because of commonality of coding with P51 and P52.

The output (x'_{ATM} , y'_{ATM} , z'_{ATM}) of AXISGEN is then handled in the same way as described for option 1 for the computation and display of the equivalent docking angles and the decisions made by the astronaut.

5.6.2.5.4 Option 3-ATM Vector Determination Using Independent Docking-Angle

Data. -Option 3 is selected by the astronaut when he wants y_{ATM} and z_{ATM} to be updated with docking-angle data from sources other than the ATM sun sensor and the ATM sun tracker; for example, the docking-ring scale indication. This option can also be selected simply for displaying the docking angles for the presently stored values of y_{ATM} and z_{ATM} .

Initially, this option computes and displays the docking angles in the same manner as in options 1 and 2, except that the presently stored values of y_{ATM} and z_{ATM} are used instead of the values (y'_{ATM} and z'_{ATM}), which are estimated by options 1 and 2. From this point on, the astronaut can perform as described in option 1. If he wishes to change any of the angles, he can do so and new values of y_{ATM} and z_{ATM} will be computed and stored.

It should be noted that the docking angle α_{ATM} computed in options 1 and 2 will usually be more accurate than that computed in option 3 via the docking-ring scale. The greater accuracy of α_{ATM} is due to its lack of structural mounting errors and to the fact that it contains only the relatively small errors of IMU alignment and the determination of the direction of the sun in basic-reference coordinates.

5.6.2.6 ATM Star Tracker Gimbal Angle Program

5.6.2.6.1 Introduction. -The ATM Star Tracker Gimbal Angle Program (P55) computes and displays the gimbal angles required by the ATM star tracker to point at a desired celestial body. The data computed by P55 can be used by the astronaut to point the star tracker at a desired object or can be loaded into the ATM computer to provide it with inertial-attitude information in the event of a

star-tracker failure.

P55 provides two options (or techniques) for computing the desired star-tracker gimbals angles. Both options require the astronaut to identify the celestial body to be acquired by the star tracker; it is assumed that the ATM Orientation Determination Program (P50) has previously been used to determine the ATM sensor attitude vectors (\underline{y}_{ATM} and \underline{z}_{ATM}).

5.6.2.6.2 Option 1 - Star Tracker Gimbal Angle Determination Using an Aligned

IMU, -In option 1, the IMU must be aligned to a known orientation (i. e., a REFSMMAT must be present). When the astronaut selects this option, the computer requests that he identify the celestial body to be acquired by the star tracker. He then chooses a celestial body code from those provided for the 37 navigation stars, the sun, the earth, or the general celestial body (normally referred to as planet). It is assumed that the astronaut will never select the sun or earth for the star tracker. If he selects the planet code, the astronaut must load the coordinates of this celestial body in the Basic Reference Coordinate System.

After the astronaut identifies the celestial body, the unit LOS vector (\underline{u}_{CB}) of the body in ATM sensor coordinates is computed as follows:

$$\underline{u}_{CB} = [NBATM][SMNB][REFSMMAT] \text{ UNIT } (\underline{u}_{CB}^* + \underline{a}) \quad (6.2.14)$$

where \underline{u}_{CB}^* is the unit LOS vector of the indicated celestial body, in basic-reference coordinates, which was either taken from the computer's star catalog in fixed memory, calculated by the computer, or loaded by the astronaut; \underline{a} is the light-aberration correction vector computed by the LOCSAM Subroutine, and $[NBATM]$ is the matrix defined in paragraph 5.6.3.4. Although the correction for aberration is not required in this program, it is applied because of commonality of coding with P51 and P52.

To obtain the desired star-tracker gimbal angles (ψ_3 and ψ_1) the following calculations are made:

$$\begin{aligned}
\underline{u}_{SP} &= \text{UNIT} \begin{bmatrix} u_{XCB} \\ u_{YCB} \\ 0 \end{bmatrix} \\
\cos \psi_3 &= \underline{u}_{SP} \cdot (-\underline{y}_{ATM}) \\
\sin \psi_3 &= \underline{u}_{SP} \cdot \underline{x}_{ATM} \\
\psi_3 &= \text{ARCTRIG}(\sin \psi_3, \cos \psi_3) \\
\cos \psi_1 &= \underline{u}_{SP} \cdot \underline{u}_{CB} \\
\sin \psi_1 &= \underline{u}_{CB} \cdot (-\underline{z}_{ATM}) \\
\psi_1 &= \text{ARCTRIG}(\sin \psi_1, \cos \psi_1)
\end{aligned} \tag{6.2.15}$$

where u_{XCB} and u_{YCB} are the X and Y components of \underline{u}_{CB} ; ARCTRIG is a routine that computes the angle, choosing either \sin^{-1} or \cos^{-1} so as to yield maximum accuracy; and

$$\underline{x}_{ATM} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \underline{y}_{ATM} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \underline{z}_{ATM} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

After ψ_3 and ψ_1 are computed, a check is made to determine if they are within the following angular limits of the star tracker:

$$\begin{aligned}
-87^\circ < \psi_3 < +87^\circ \\
-40^\circ < \psi_1 < +40^\circ
\end{aligned} \tag{6.2.16}$$

If the angles are not within these limits the PROG alarm light illuminates and an alarm code is stored. In either case, however, the angles are displayed in arc minutes decimal. By keying PRO the astronaut can also have ψ_3 displayed in a special format suitable for loading into the ATM computer in the event of star-tracker failure. If the astronaut is not satisfied with the angles, he can recycle for either display, and select another celestial body for gimbal-angle computation.

5.6.2.6.3 Option 2—Star Tracker Gimbal Angle Determination Using CSM

Optics. —Option 2 does not require that the IMU be on since IMU orientation with respect to the Basic Reference Coordinate System is not used. Instead, the orientation of the ATM Sensor Coordinate System is determined with respect to the Basic Reference Coordinate System by sighting on one celestial body with the CSM optics while the ATM is in the solar-inertial attitude. The known LOS of the sun then serves as the other spatial direction in space that must be determined.

At the beginning of option 2, the astronaut uses the SXT or SCT in conjunction with the Sighting Mark Routine (R53) to perform a sighting on one celestial body; he later identifies this body via the celestial body code. The AXISGEN Routine (paragraph 5.6.3.2.4) then provides the matrix [BRATM] for transforming vectors from basic reference to ATM sensor coordinates. The inputs to this routine are:

$$\begin{aligned}
 \underline{s}'_A &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
 \underline{s}'_B &= [\text{NBATM}][\text{SBNB}] \begin{bmatrix} \sin TA \cos SA \\ \sin TA \sin SA \\ \cos TA \end{bmatrix} \\
 \underline{s}_A &= \underline{u}_{ES} \\
 \underline{s}_B &= \text{UNIT} (\underline{u}_{CB}^* + \underline{a})
 \end{aligned} \tag{6.2.17}$$

where SA and TA are the shaft and trunnion angles recorded for the mark; [SBNB] is the matrix in paragraph 5.6.3.1; [NBATM] is the matrix in paragraph 5.6.3.4; \underline{u}_{ES} is the unit LOS vector of the sun in basic-reference coordinates computed for the time of the mark by the Solar Ephemerides Subroutine of paragraph 5.5.4; \underline{a} is the correction vector for aberration of light, and \underline{u}_{CB}^* is the unit LOS vector of the indicated celestial body in basic-reference coordinates; this unit vector was either taken from fixed memory, calculated by the computer, or loaded by the astronaut.

After AXISGEN determines [BRATM], the astronaut identifies the celestial body to be acquired by the star tracker, and the unit LOS vector (\underline{u}_{CB}) of the body in ATM sensor coordinates is computed as follows:

$$\underline{u}_{CB} = [\text{BRATM}] \text{UNIT} (\underline{u}_{CB}^* + \underline{a}) \quad (6.2.18)$$

where \underline{a} is the correction vector for aberration of light, and \underline{u}_{CB}^* is the unit LOS vector of the indicated celestial body in basic-reference coordinates.

From this point on, the desired star-tracker gimbals angles are computed and displayed as they are in option 1. It should be noted that option 2 assumes that the orientation of the ATM with respect to the Basic Reference Coordinate System will remain fixed from the time of the optical sighting with the SXT or SCT until the star-tracker gimbals angles are used by the crew.

5.6.3 IMU Routines

5.6.3.1 Sextant Transformations

5.6.3.1.1 Sextant Base-Navigation Base. —To transform a vector from sextant-base to navigation-base coordinates, the following transformation matrix is used:

$$[\text{SBNE}] = \begin{pmatrix} \cos \alpha & 0 & -\sin \alpha \\ 0 & 1 & 0 \\ \sin \alpha & 0 & \cos \alpha \end{pmatrix} \quad (6.3.1)$$

where $\alpha = -32^{\circ} 31' 23.19''$

This matrix and its transpose [NBSB] are both present in computer memory.

The sextant-base coordinate system referred to above is frequently called the Block I navigation base coordinate system in deference to the present navigation base coordinate system (Block II). Transformation of vectors between the sextant base and the present navigation base requires use of the above transformations.

5.6.3.1.2 Sextant to Navigation Base (SXTNB). -To obtain a unit vector (\underline{u}_{NB}), which specifies the direction of the line-of-sight of the sextant in navigation-base coordinates, use is made of the following:

$$\underline{u}_{NB} = [\text{SBNB}] \begin{pmatrix} \sin TA \cos SA \\ \sin TA \sin SA \\ \cos TA \end{pmatrix} \quad (6.3.2)$$

where $[\text{SBNB}]$ is the transformation matrix given in paragraph 5.6.3.1.1, and TA and SA are the trunnion (precision) and shaft angles, respectively, of the sextant.

5.6.3.1.3 Calculation of Sextant Angles (CALCSXA). -Given a unit star vector \underline{s}_{SM} in stable-member coordinates, this routine computes the angles SA and TA required to position the optics such that the line of sight lies along the star vector.

The star vector is first transformed to the sextant base coordinate system as follows:

$$\underline{s}_{SB} = [\text{NBSB}] [\text{SMNB}] \underline{s}_{SM} \quad (6.3.3)$$

where $[\text{NBSB}]$ and $[\text{SMNB}]$ are given in paragraphs 5.6.3.1.1 and 5.6.3.2.1, respectively.

Next, a unit vector \underline{u}_{TDA} , defining the direction of the trunnion (precision) drive axis in sextant-base coordinates, is obtained.

$$\underline{u}_{TDA} = \text{UNIT} \left[\underline{z}_{SB} \times \underline{s}_{SB} \right] \quad (6.3.4)$$

where

$$\underline{z}_{SB} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Using \underline{s}_{SB} and \underline{u}_{TDA} , the angles SA and TA are determined as follows:

$$\begin{aligned} \sin SA &= \underline{u}_{TDA} \cdot (-\underline{x}_{SB}) \\ \cos SA &= \underline{u}_{TDA} \cdot \underline{y}_{SB} \\ SA &= \text{ARCTRIG}(\sin SA, \cos SA) \\ TA &= \cos^{-1} \left[\underline{z}_{SB} \cdot \underline{s}_{SB} \right] \end{aligned} \tag{6.3.5}$$

where ARCTRIG implies computing the angle, choosing either \sin^{-1} or \cos^{-1} so as to yield maximum accuracy, and

$$\underline{x}_{SB} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad \underline{y}_{SB} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

5.6.3.2 IMU Transformations

5.6.3.2.1 Stable Member-Navigation Base.—Let IGA, MGA, OGA be the IMU inner, middle and outer gimbal angles, respectively. Define the following matrices:

$$Q_1 = \begin{pmatrix} \cos IGA & 0 & -\sin IGA \\ 0 & 1 & 0 \\ \sin IGA & 0 & \cos IGA \end{pmatrix} \tag{6.3.6}$$

$$Q_2 = \begin{pmatrix} \cos MGA & \sin MGA & 0 \\ -\sin MGA & \cos MGA & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6.3.7}$$

$$Q_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos OGA & \sin OGA \\ 0 & -\sin OGA & \cos OGA \end{pmatrix} \quad (6.3.8)$$

Stable Member to Navigation Base Transformation

$$\underline{U}_{NB} = Q_3 Q_2 Q_1 \underline{U}_{SM} \quad (6.3.9)$$

$$\left[\text{SMNB} \right] = Q_3 Q_2 Q_1$$

Navigation Base to Stable Member Transformation

$$\underline{U}_{SM} = Q_1^T Q_2^T Q_3^T \underline{U}_{NB}, \quad \left[\text{NBSM} \right] = Q_1^T Q_2^T Q_3^T \quad (6.3.10)$$

5.6.3.2.2 Calculation of Gimbal Angles (CALCGA). --Given a stable-member orientation and a navigation-base orientation, both referred to the same coordinate system, the following procedure is used to compute the corresponding gimbal angles.

$$\underline{a}_{MG} = \text{UNIT} (\underline{x}_{NB} \times \underline{y}_{SM})$$

$$\cos OGA = \underline{a}_{MG} \cdot \underline{z}_{NB} \quad (6.3.11)$$

$$\sin OGA = \underline{a}_{MG} \cdot \underline{y}_{NB}$$

$$\text{OGA} = \text{ARCTRIG} (\sin \text{OGA}, \cos \text{OGA})$$

$$\cos \text{MGA} = \underline{y}_{\text{SM}} \cdot (\underline{a}_{\text{MG}} \times \underline{x}_{\text{NB}})$$

$$\sin \text{MGA} = \underline{y}_{\text{SM}} \cdot \underline{x}_{\text{NB}}$$

$$\text{MGA} = \text{ARCTRIG} (\sin \text{MGA}, \cos \text{MGA}) \quad (6.3.11 \text{ cont.})$$

$$\cos \text{IGA} = \underline{a}_{\text{MG}} \cdot \underline{z}_{\text{SM}}$$

$$\sin \text{IGA} = \underline{a}_{\text{MG}} \cdot \underline{x}_{\text{SM}}$$

$$\text{IGA} = \text{ARCTRIG} (\sin \text{IGA}, \cos \text{IGA})$$

where the inputs are three vectors along the stable-member axes and three vectors along the navigation-base axes.

5.6.3.2.3 Calculation of Gyro Torquing Angles (CALCGTA). —In the fine align procedure, after the present platform orientation is determined, the torquing angles required to move the platform into the desired orientation must be computed. The computation is made as follows.

Let \underline{x}_D , \underline{y}_D , and \underline{z}_D be the desired stable member axes referred to the present stable member orientation, and let \underline{x}_P , \underline{y}_P , and \underline{z}_P denote the present stable member axes. The rotations are performed in three steps: (1) rotating through θ_y about the present y axis, yielding \underline{x} , \underline{y}_P , and \underline{z} ; (2) rotating through θ_z about the z axis, yielding \underline{x}_D , \underline{y} , \underline{z} ; (3) and finally rotating through θ_x about the \underline{x}_D axis, yielding \underline{x}_D , \underline{y}_D , \underline{z}_D . The relevant equations are as follows:

$$\underline{z} = \text{UNIT} (-x_{D,3}, 0, x_{D,1}) \quad (6.3.12)$$

$$\sin \theta_y = z_1$$

$$\cos \theta_y = z_3$$

$$\theta_y = \text{ARCTRIG}(\sin \theta_y, \cos \theta_y)$$

$$\sin \theta_z = x_{D, 2}$$

$$\cos \theta_z = z_3 x_{D, 1} - z_1 x_{D, 3}$$

(6. 3. 12 cont.)

$$\theta_z = \text{ARCTRIG}(\sin \theta_z, \cos \theta_z)$$

$$\cos \theta_x = \underline{z} \cdot \underline{z}_D$$

$$\sin \theta_x = \underline{z} \cdot \underline{y}_D$$

$$\theta_x = \text{ARCTRIG}(\sin \theta_x, \cos \theta_x)$$

The required inputs are the three coordinate axes of the desired stable-member orientation referred to the present stable-member orientation.

5. 6. 3. 2. 4 Coordinate Axes Generator (AXISGEN). — Given two unit vectors (usually star vectors), \underline{s}_A and \underline{s}_B , expressed in two coordinate systems, denoted by primed and unprimed characters, i. e., \underline{s}'_A , \underline{s}'_B , \underline{s}_A , \underline{s}_B , this routine computes the unit vectors \underline{x} , \underline{y} , \underline{z} that are the primed coordinate system axes referred to the unprimed coordinate system. This is accomplished by defining two orthogonal coordinate sets, one in each system, in the following manner:

$$\underline{u}'_X = \underline{s}'_A$$

$$\underline{u}'_Y = \text{UNIT}(\underline{s}'_A \times \underline{s}'_B)$$

$$\underline{u}'_Z = \underline{u}'_X \times \underline{u}'_Y$$

$$\underline{u}_X = \underline{s}_A$$

(6. 3. 13)

$$\underline{u}_Y = \text{UNIT}(\underline{s}_A \times \underline{s}_B)$$

$$\underline{u}_Z = \underline{u}_X \times \underline{u}_Y$$

The primed coordinate system axes expressed in terms of the unprimed coordinate system axes are:

$$\begin{aligned}\underline{x} &= u'_{X1}u_{-X} + u'_{Y1}u_{-Y} + u'_{Z1}u_{-Z} \\ \underline{y} &= u'_{X2}u_{-X} + u'_{Y2}u_{-Y} + u'_{Z2}u_{-Z} \\ \underline{z} &= u'_{X3}u_{-X} + u'_{Y3}u_{-Y} + u'_{Z3}u_{-Z}\end{aligned}\tag{6.3.14}$$

It should be noted that vectors can be transformed from the unprimed to the primed coordinate systems by using the following matrix constructed with the output (Equation 6.3.14) of AXISGEN:

$$\begin{bmatrix} \underline{x}^T \\ \underline{y}^T \\ \underline{z}^T \end{bmatrix}\tag{6.3.15}$$

5.6.3.2.5 Calculation of Stable Member Coordinates of the Spacecraft

(CALCSMSC). --To determine the directions of the X, Y, and Z axes of the present vehicle-coordinate system or the navigation-base coordinate system with respect to the IMU Stable Member Coordinate System, the CALCSMSC Routine is used.

The unit vectors \underline{x}_{NB} , \underline{y}_{NB} , and \underline{z}_{NB} defining the directions of the navigation-base coordinate system axes with respect to the IMU Stable Member Coordinate System are determined as follows:

$$\underline{x}_{NB} = \begin{pmatrix} \cos IGA & \cos MGA \\ \sin MGA \\ -\sin IGA & \cos MGA \end{pmatrix}\tag{6.3.15a}$$

$$\underline{z}_{NB} = \begin{pmatrix} \cos IGA & \sin OGA & \sin MGA \\ + \cos OGA & \sin IGA & \\ -\sin OGA & \cos MGA & \\ \cos OGA & \cos IGA & \\ -\sin OGA & \sin MGA & \sin IGA \end{pmatrix} \quad (6.3.15b)$$

$$\underline{y}_{NB} = \underline{z}_{NB} \times \underline{x}_{NB} \quad (6.3.15c)$$

where IGA, MGA, and OGA are the inner, middle, and outer IMU gimbal angles, respectively. It should be noted that the rows of the transformation matrix [SMNB] also give the above vectors.

5.6.3.3 REFSMMAT Transformations

The matrix required to transform a vector from the Basic Reference Coordinate System to the IMU Stable Member Coordinate System is referred to as REFSMMAT. This matrix can be constructed as follows with the unit vectors \underline{u}_{XSM} , \underline{u}_{YSM} , and \underline{u}_{ZSM} defining the orientations of the stable-member axes with respect to reference coordinates.

$$\text{REFSMMAT} = \begin{bmatrix} \underline{u}_{XSM}^T \\ \underline{u}_{YSM}^T \\ \underline{u}_{ZSM}^T \end{bmatrix} \quad (6.3.16)$$

5.6.3.3.1 Present REFSMMAT From Star Sightings. —The present IMU stable-member orientation with respect to the Basic Reference Coordinate System, and the associated REFSMMAT, can be determined by sighting on two navigation stars with the CSM optics. If \underline{s}'_A and \underline{s}'_B are the unit vectors defining the measured directions of the two stars in the present Stable Member Coordinate System, and \underline{s}_A and \underline{s}_B are the unit vectors to the corresponding stars as known in the Basic Reference Coordinate System, then these vectors can be used as the input to the AXISGEN Routine (paragraph 5.6.3.2.4) to obtain the present IMU orientation and REFSMMAT [Equations (6.3.14) and (6.3.15)].

5.6.3.3.2 Alignment for Thrusting Maneuvers (Preferred Orientation). —During certain thrusting maneuvers the IMU will be aligned according to the following equations.

$$\begin{aligned} \underline{u}_{XSM} &= \text{UNIT}(\underline{x}_B) \\ \underline{u}_{YSM} &= \text{UNIT}(\underline{u}_{XSM} \times \underline{r})^* \\ \underline{u}_{ZSM} &= \underline{u}_{XSM} \times \underline{u}_{YSM} \end{aligned} \tag{6.3.17}$$

where \underline{x}_B is the vehicle or body X-axis at the preferred vehicle attitude for ignition and \underline{r} is the CSM position vector. The associated transformation matrix (REFSMMAT) is given by Equation (6.3.16).

5.6.3.3.3 Alignment to Local Vertical in Orbit (Nominal Orientation). —The IMU stable member can be aligned to the local vertical at a specified time. For this type of orientation, the stable-member axes are found from the following.

$$\begin{aligned} \underline{u}_{XSM} &= (\underline{u}_{YSM} \times \underline{u}_{ZSM}) \\ \underline{u}_{YSM} &= \text{UNIT}(\underline{v} \times \underline{r}) \\ \underline{u}_{ZSM} &= \text{UNIT}(-\underline{r}) \end{aligned} \tag{6.3.18}$$

where \underline{r} and \underline{v} are the position and velocity vectors of the CSM at the specified time. The vectors \underline{r} and \underline{v} are computed by the Coasting Integration Routine (paragraph 5.2.2). The REFSMMAT associated with this IMU orientation is found from Equation (6.3.16).

*In P41, if $|\underline{u}_{XSM} \times \underline{r}| < 2^{16}$ meters, or in P40 if $|\underline{u}_{XSM} \times \text{UNIT}(\underline{r})| < 2^{-12}$ radians, then $\underline{u}_{YSM} = \text{UNIT}[\underline{u}_{XSM} \times (\text{UNIT}(\underline{r}) + 0.125 \text{UNIT}(\underline{v}))]$ where \underline{v} is the CSM velocity vector.

5.6.3.3.4 Plane Change Alignment. —The plane change alignment is used whenever the CSM performs a plane change maneuver to the thrust attitude during MINKEY. The alignment is accomplished by pulse torquing the Z-gyro either through minus 45 deg, before the maneuver, and through plus 45 deg after, or through plus 45 deg before and minus 45 deg after. The purpose of the pulse torquing after the maneuver is to return the IMU to its original orientation. The choice of pulse torquing plus or minus 45 deg before the maneuver is made as follows. If $\cos(\text{CDUY})\text{VG}_{LV} \geq 0$ the Z-gyro is first pulse torqued minus 45 deg and then plus 45 deg. Otherwise, the reverse is done.

$$\begin{aligned} \underline{u}_{XSM} &= \text{UNIT} (\underline{u}_{XSM0} \cos 45^\circ - \underline{u}_{YSM0} \sin 45^\circ) & (6.3.18a) \\ & \text{(minus } 45^\circ \text{ then plus } 45^\circ) \end{aligned}$$

or

$$\begin{aligned} \underline{u}_{XSM} &= \text{UNIT} (\underline{u}_{XSM0} \cos 45^\circ + \underline{u}_{YSM0} \sin 45^\circ) & (6.3.18b) \\ & \text{(plus } 45^\circ \text{ then minus } 45^\circ) \end{aligned}$$

$$\underline{u}_{YSM} = \text{UNIT} (\underline{u}_{ZSM0} \times \underline{u}_{XSM}) \quad (6.3.18c)$$

$$\underline{u}_{ZSM} = \underline{u}_{ZSM0} \quad (6.3.18d)$$

where 0 refers to the orientation existing before plane change alignment.

5.6.3.3.5 Earth Prelaunch Alignment. —Prior to earth launch the IMU stable member is aligned to a local-vertical axis system.

$$\underline{u}_{ZSM} = \text{UNIT} (-\underline{r}) \text{ (local vertical)}$$

$$\underline{u}_{XSM} = \text{UNIT} (\underline{A}) \text{ where } \underline{A} \text{ is a horizontal vector} \quad (6.3.18e)$$

pointed at the desired launch azimuth angle.

$$\underline{u}_{YSM} = \underline{u}_{ZSM} \times \underline{u}_{XSM}$$

The REFSMMAT associated with this IMU orientation is given in Equation (6.3.16).

5.6.3.4 Navigation Base to ATM Transformation (NBATM)

To transform a vector from navigation base to ATM sensor coordinates use is made of the matrix [NBATM] which is computed as follows:

$$[\text{NBATM}] = \begin{bmatrix} (\underline{y}_{\text{ATM}} \times \underline{z}_{\text{ATM}})^T \\ \underline{y}_{\text{ATM}}^T \\ \underline{z}_{\text{ATM}}^T \end{bmatrix} \quad (6.3.19)$$

where $\underline{y}_{\text{ATM}}$ and $\underline{z}_{\text{ATM}}$ are unit vectors that define the corresponding directions of the +Y-axis and +Z-axis of the ATM Sensor Coordinate System in navigation-base coordinates when the CSM is docked to the workshop. It is assumed that anticipated nominal values of $\underline{y}_{\text{ATM}}$ and $\underline{z}_{\text{ATM}}$ will be loaded in erasable memory before launch and that they will be updated by the ATM Orientation Determination Program (P50) after docking to the workshop. To transform a vector from ATM sensor to navigation base coordinates use is made of the matrix

$$[\text{ATMNB}] = [\text{NBATM}]^T \quad (6.3.20)$$

5.6.4 Star Selection Routine

The Star Selection Routine is used by the IMU Realignment Program (P52) and the Backup IMU Realignment Program (P54) to select the best pair of stars in the viewing cone of the sextant for fine alignment of the IMU. The logic diagram for this routine is shown in Figure 6.4-1.

This routine tests each pair of stars from the computer catalog of 37 stars to determine if both stars are within a 76-degree viewing cone centered with respect to the shaft drive axis of the sextant (SXT). The routine then checks to determine if the angle of separation between the stars is at least 30 degrees.

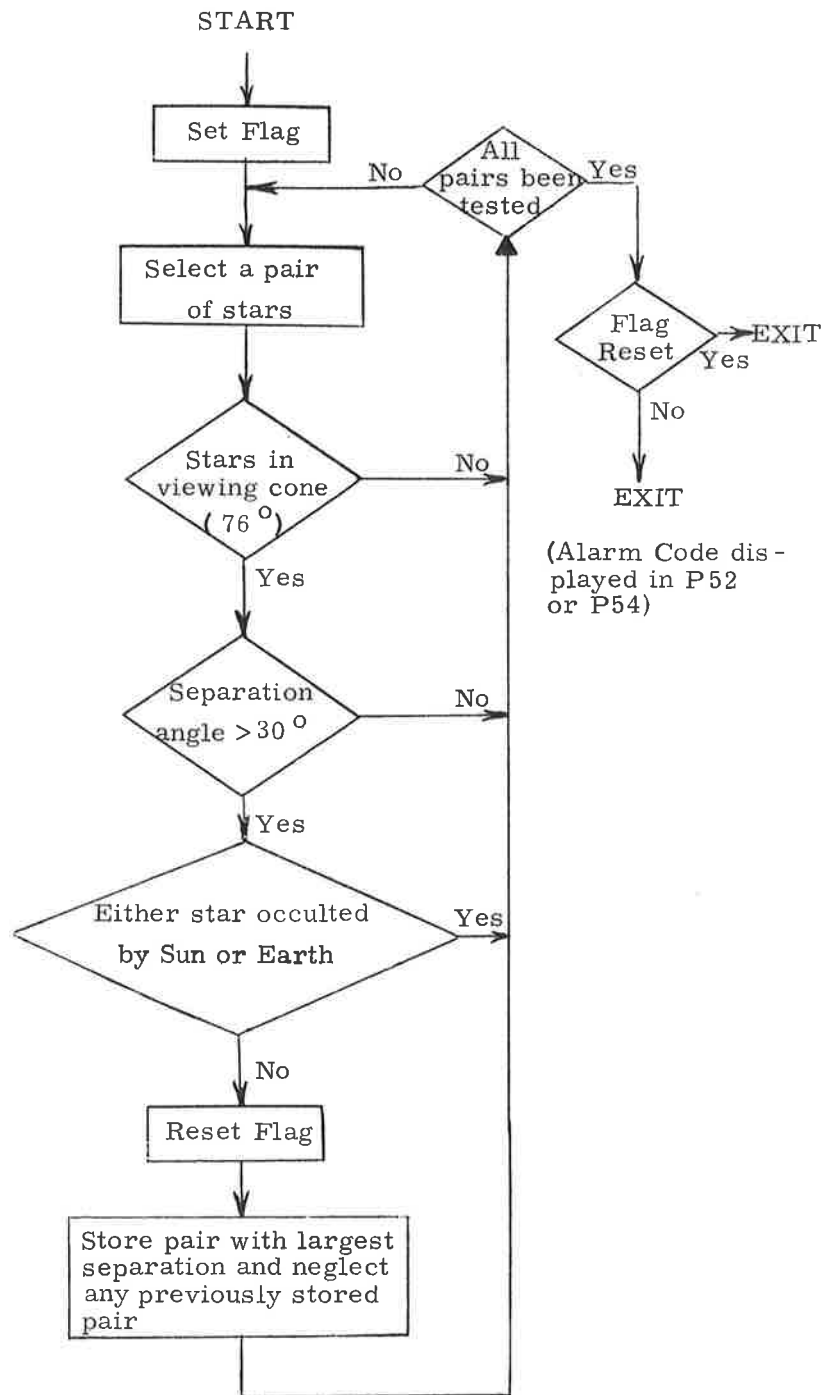


Figure 6.4-1. Star Selection Routine

If a pair of stars passes both tests, the routine further checks to determine if either star is occulted by the sun or earth. The sizes of the occultation cones are such as to not only account for true occultation, but also to prevent the selection of stars too near the bodies because of visibility problems. The directions and the associated occultation cone sizes of the two bodies are actually computed by the LOCSAM Subroutine (paragraph 5.5.13), which the IMU Realignment Programs call before they call the Star Selection Routine.

The pair of stars that passes all the above tests and that has the largest angular separation is chosen by this routine. If the routine is unable to find a satisfactory pair of stars after testing all combinations, it is seen in programs P52 and P54 of Section 4 that an alarm code is displayed; the astronaut can then either repeat the star selection process at a different spacecraft attitude, or select his own stars later.

5.6.5 Ground Track Routine

This routine is used by the astronaut in near-earth orbit to obtain CSM or WS trajectory information. The astronaut specifies a time (GET) and a vehicle (CSM or WS). The routine uses the Coasting Integration Routine (paragraph 5.2.2) to extrapolate the desired vehicle's state vector to the specified time. The resulting estimated position vector is converted to latitude, longitude, altitude coordinates by means of the Latitude-Longitude Subroutine (paragraph 5.5.3), and these data are displayed. Altitude is defined with respect to the launch pad radius for earth orbit. The astronaut can request the state-vector extrapolation to continue in ten-minute steps, or to another specified time, and obtain additional displays of the coordinates of points in the spacecraft's orbit. Alternatively, the astronaut can request a display of altitude (to 10 n. mi.), inertial velocity magnitude (to 1 ft/sec), and flight path angle (in degrees) at an astronaut-specified time.

5.6.6 VHF Range Rate Mark Processing Routine

The VHF Range Rate Mark Processing Routine, R27, provides a least-mean squares fit of VHF range data to a second-order polynomial in time. It also includes the computation of ϕ , the angle between the local horizontal and the sextant LOS, or θ , the angle between the local horizontal and the +X body axis. Range data are incorporated into the expansion recursively. The polynomial is expanded about either the current time or some fixed time supplied by the astronaut. The estimates of

range and range rate are optimized at this fixed time for use in chart solutions for rendezvous maneuvers.

The inputs to the routine are:

- σ_v — VHF ranger variance
- \vec{r} — a priori acceleration error used to initialize W_8
- $\overline{\vec{r}}$ — a priori value of \vec{r} used to degrade the W-matrix
- t_d — the fixed time at which the optimum estimate of range and range rate are desired
- t_b — a time interval during which range and range-rate values are optimized.

The inputs σ_v , and t_b are fixed-memory constants; \vec{r} and t_d are pad-loaded erasable-memory constants; and $\overline{\vec{r}}$ is an erasable-memory constant set to zero at initialization of R27. Although t_d is set to zero at pad load, its value can be changed at the discretion of the crew. The outputs of the routine are range and range rate at the current or fixed time, and the angle θ or ϕ .

When one of the three programs—P20, P25, or P48—is selected, two flags associated with R27—R27UP1 and R27UP2—are set to zero. The first two VHF range measurements are used to generate an initial state; succeeding measurements are incorporated recursively. The flowchart in Figure 6.6-1 shows the branches for the initialization and recursive procedures. The initialization procedure forms an estimated state, \underline{x} , and weighting matrix, W , as shown in the flowchart, Figure 6.6-2. The three dimensional polynomial state, \underline{x} , is made up of the three elements:

$$\begin{aligned}
 x_0 &= r_0 \text{ range at } t_0 \\
 x_1 &= v_0 \text{ range rate at } t_0 \\
 x_2 &= a_0 \text{ range acceleration at } t_0
 \end{aligned}
 \tag{6.6.1}$$

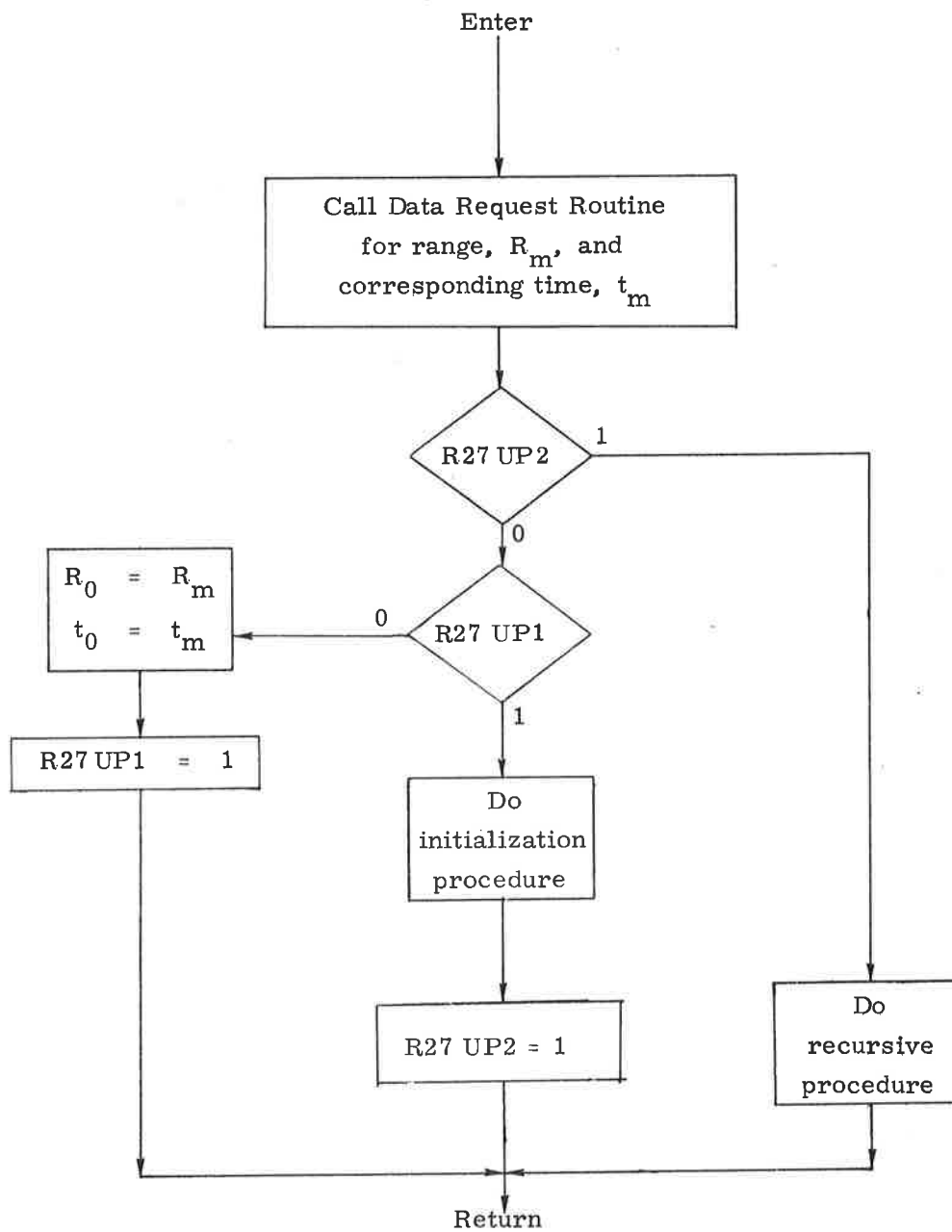


Figure 6.6-1. Initialization and Recursive Branches for R27

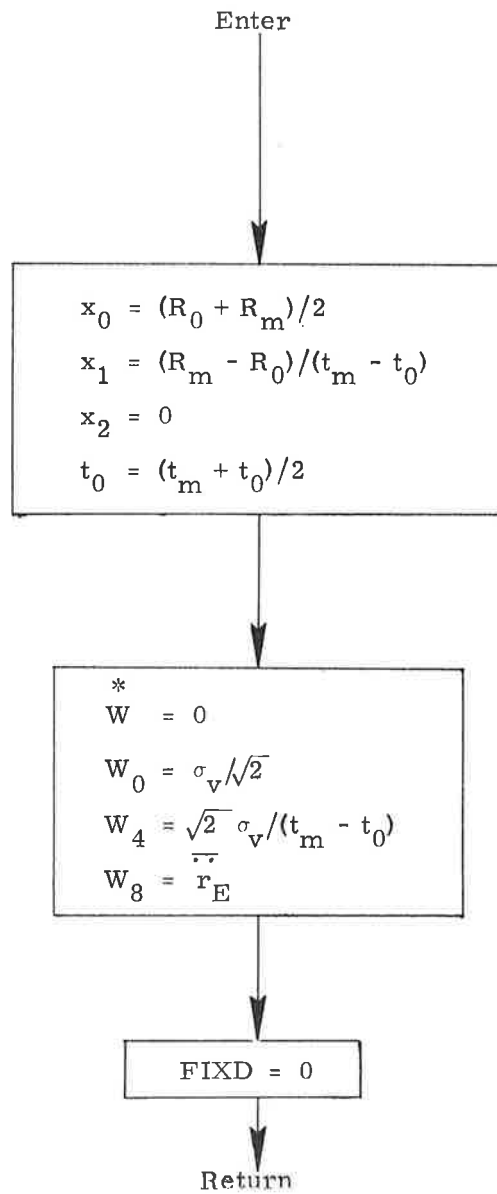


Figure 6.6-2. Initialization Procedure

The weighting matrix, W , is composed of nine elements numbered as follows:

$$W = \begin{bmatrix} W_0 & W_1 & W_2 \\ W_3 & W_4 & W_5 \\ W_6 & W_7 & W_8 \end{bmatrix} \quad (6.6.2)$$

The rows of this matrix may be thought of as vectors and numbered as follows:

$$\begin{aligned} \underline{W}_0 &= (W_0, W_1, W_2) \\ \underline{W}_3 &= (W_3, W_4, W_5) \\ \underline{W}_6 &= (W_6, W_7, W_8) \end{aligned} \quad (6.6.3)$$

Since two range measurements cannot yield an estimate of acceleration, x_2 is set equal to zero. The corresponding diagonal term of the W matrix, W_8 , is set equal to an a priori value, \bar{r} . The fixed-display flag, $FIXFLAG$, is set to 0 to indicate that the range and range rate are to be extrapolated so that the state is at the current time.

Following initialization, measurements are incorporated recursively as shown in the block diagram, Figure 6.6-3. The power series coefficients can be considered a three-dimensional state, $\underline{x} = (r_0, v_0, a_0)$, at t_0 . The estimated range at the time of the measurement t_m as a function of the state is,

$$r_e(t_m) = r_0 + v_0(t_m - t_0) + \frac{1}{2}a_0(t_m - t_0)^2 \quad (6.6.4)$$

The difference in the measured and estimated range, δQ , can thus be written,

$$\begin{aligned} \delta Q &= R_m - r_e(t_m) \\ &= R_m - \underline{b} \cdot \underline{x} \end{aligned} \quad (6.6.5)$$

The vector \underline{b} is

$$\frac{\partial}{\partial x_i} r_e(t_m) \text{ or } \underline{b} = (1, \Delta t, \frac{1}{2} \Delta t^2) \quad (6.6.6)$$

Using this \underline{b} vector the weighting vector \underline{w} can be calculated and the state, \underline{x} , and W-matrix, W , can be updated.

Following the update, a set of logical choices is made depending on the values of the quantities FIXFLAG, t_m , t_d , and t_b . The function of this set of choices is to decide whether \underline{x} is to be the state at the current time or at some fixed time t_d and to set the time Δt to extrapolate the state and W-matrix as needed. The result of the logic is to:

- a. Compute the current time state if the time of the measurement is not within t_b of t_d in either direction.
- b. Begin computing an optimum estimate of range and range rate at t_d as soon as t_m is greater than $t_d - t_b$.
- c. Replace the last computed current state for the time $t_d - t_b$ by the optimized state at the new time t_d as soon as t_m is greater than t_d .
- d. Continue improving and updating the optimum estimate until t_m is greater than $t_d + t_b$, then begin computing the current state once again.

If required, the state and W-matrix are extrapolated in time by the amount Δt set in the previous logic. For each extrapolation, the W-matrix is degraded by an amount dependent on the value of \underline{r} . The effect on the covariance matrix $\underline{W}\underline{W}^T$ is to increase the third diagonal term $\overline{e_r e_r}$ by the amount $(\underline{r} \Delta t)^2$. The purpose of this "degrading" of the W-matrix is to take into account errors in the estimated state due to truncated, but non-zero, higher order range derivatives.

5.6.6.1 Data Read Routine for Range Rate Mark Processing

The granularity in the VHF range data is 0.01 n. mi. This leads to errors in the measured range between zero and 60.76 feet. This error can be

reduced by making a rapid succession of VHF measurements and accepting a value when the reading changes. Figure 6.6.4 shows the logic for choosing the VHF measurement at the "change" time.

5.6.6.2 Provision for VHF Range Modulus

If the filter is being used in conjunction with P20 or P48, the range returned from R08 will be the true range; if it is used with P25, however, the range will be modulo 327.68 n. mi. In the former case, the basis for constructing the true range from the modulo 327.68 n. mi. VHF ranger reading is the range computed in R61 from the CM and WS state vectors. These vectors are assumed to be unavailable for P25 operation; therefore a logical sequence—indicated VHFMOD in Figure 6.6-4 and illustrated in Figure 6.6-5—is designed to keep the first component of the polynomial state, x_0 , between the values of zero and 327.68 n. mi. It is left to the crew to add multiples of 327.68 n. mi. according to their knowledge of the range between the vehicles.

5.6.6.3 R27 Computation of Theta and Phi

The angles ϕ (in P20) and θ (in P48) are defined completely in paragraphs 5.6.7.1 and 5.6.7.2. They are computed in R27 independently of R31 and R34.

In P20, accuracy requirements for ϕ at $t = t_d$ make precision integration necessary in extrapolating the CSM state vector to this time. Otherwise, for $t < t_d$, computation of ϕ is done with conic integration.

In P48, use is made of the current Average-G state vector $\underline{r}(t)$, $\underline{v}(t)$ in updating θ every two seconds.

The logic for the computation of the angles ϕ and θ in R27 is shown in Figure 6.6-6.

5.6.7 Additional Rendezvous Displays

During the final phases of rendezvous the following four routines may be called by the astronaut for the purpose of computing and displaying special quantities related to the rendezvous geometry.

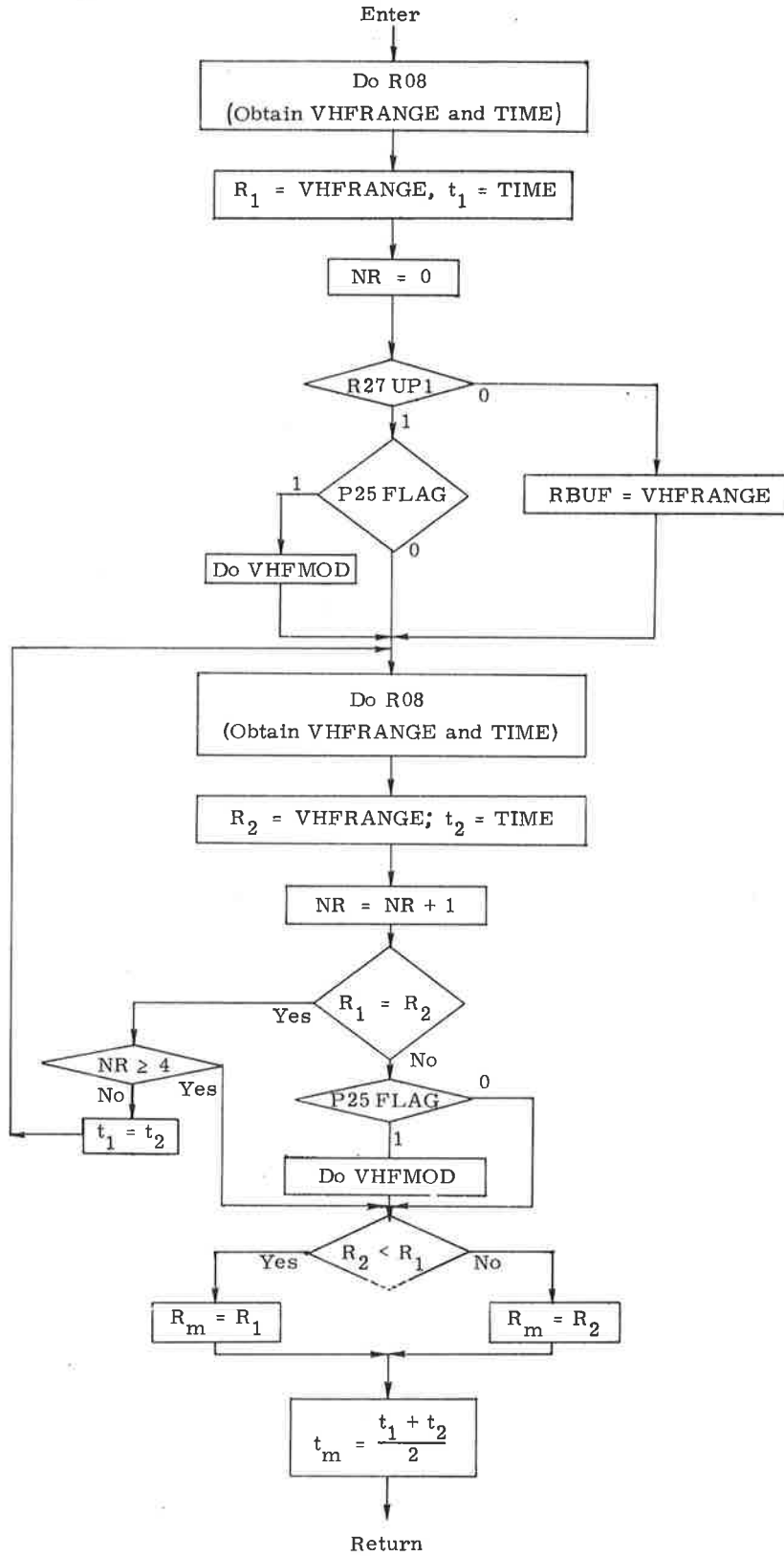
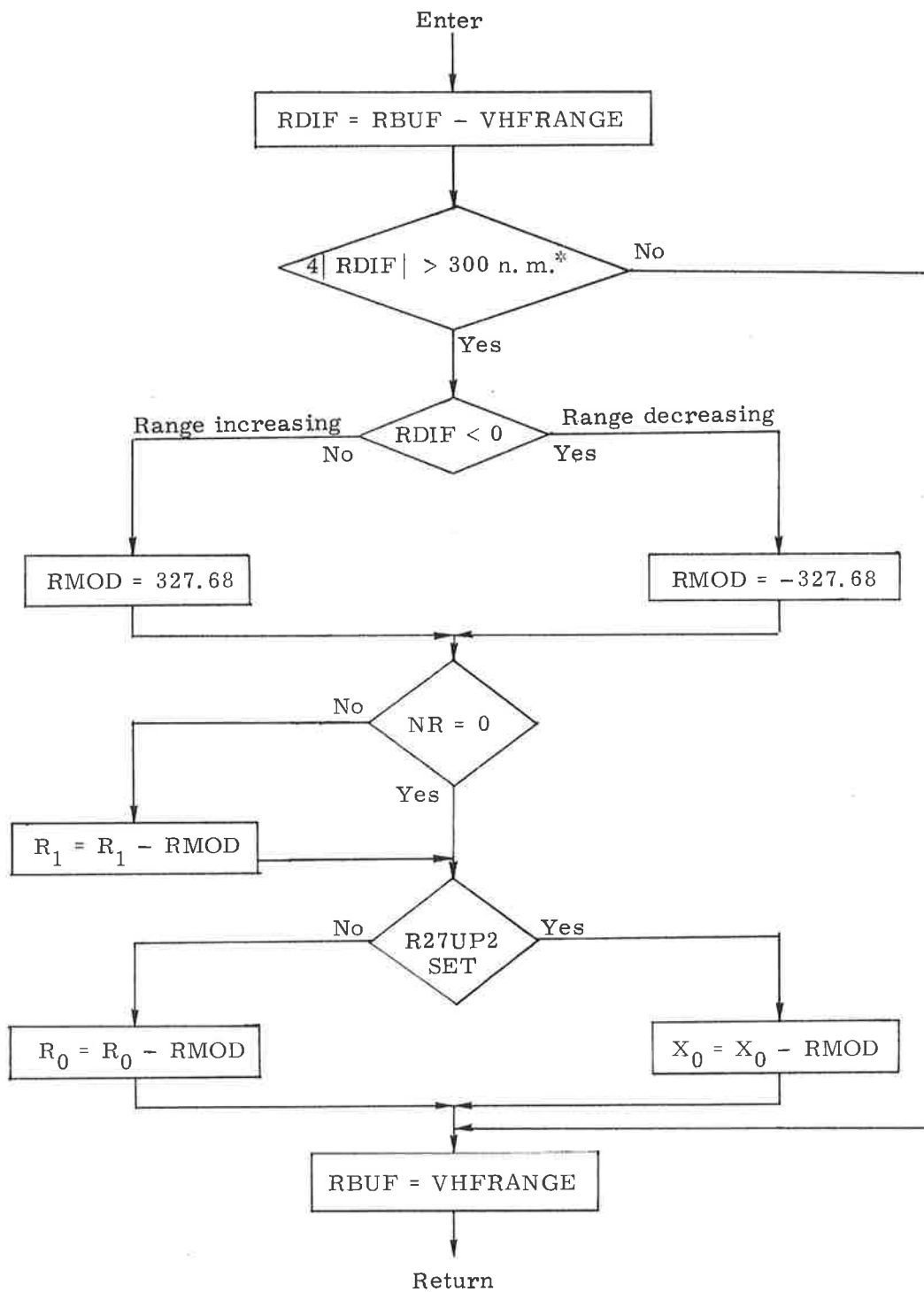


Figure 6.6-4. Data Request Routine



*The program actually uses 75 n. m. and not 300 n. m.

Figure 6.6-5. Polynomial Range Estimate Modification Due to VHF Range Modulus (VHFMOD Sequence)

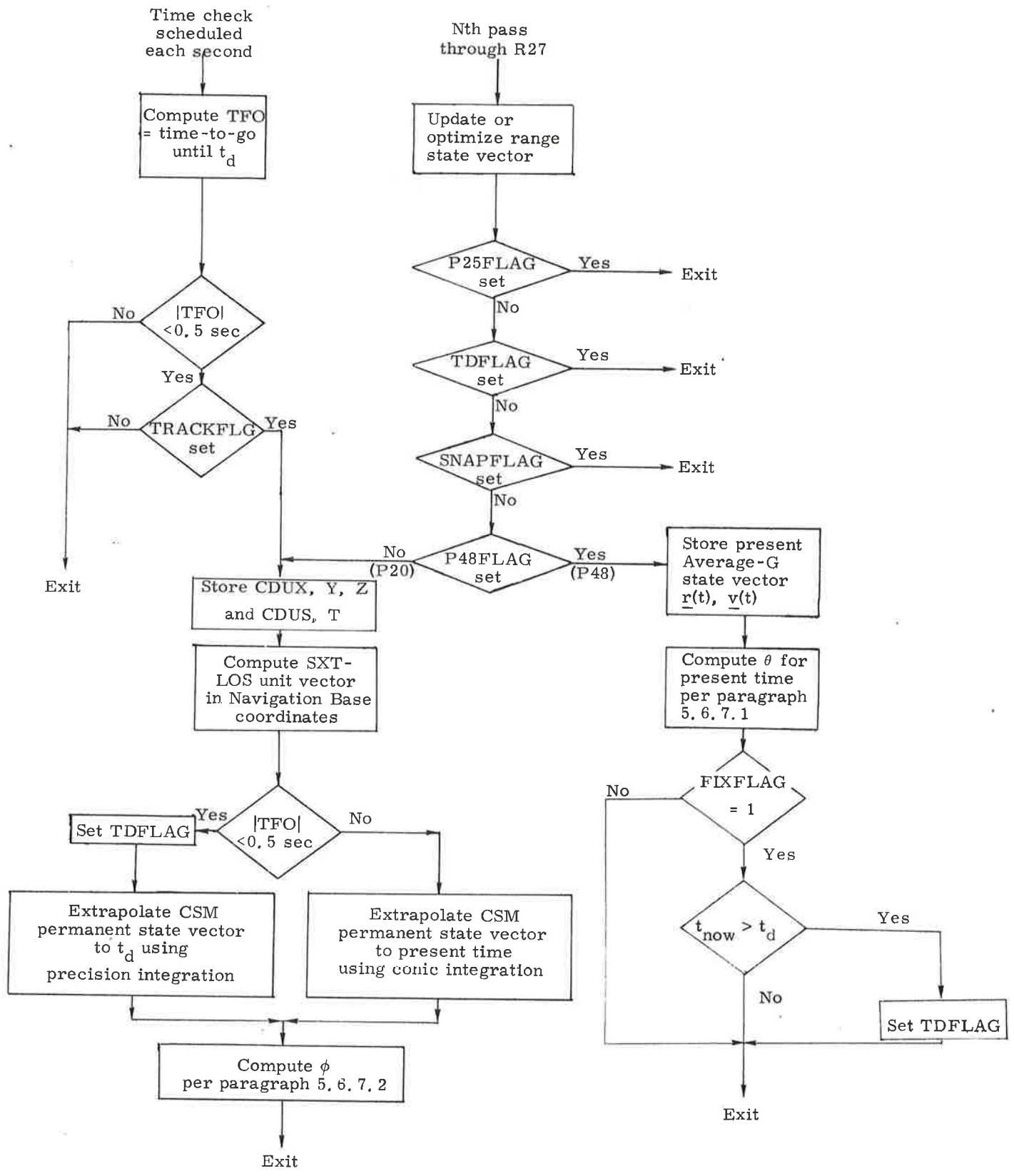


Figure 6.6-6. Computation of Theta and Phi in R27

5.6.7.1 Range, Range Rate, Theta Display (R31)

Routine R31 may be called upon to compute and display the range and range rate between the two vehicles and an angle θ shown in Figure 6.7-1.

The angle θ represents the angle between the CSM X-body axis and the local horizontal plane. It is defined in a manner completely analogous to the definition of E_L in subsection 5.4. Theta, therefore, can have values between 0 and 360 deg.

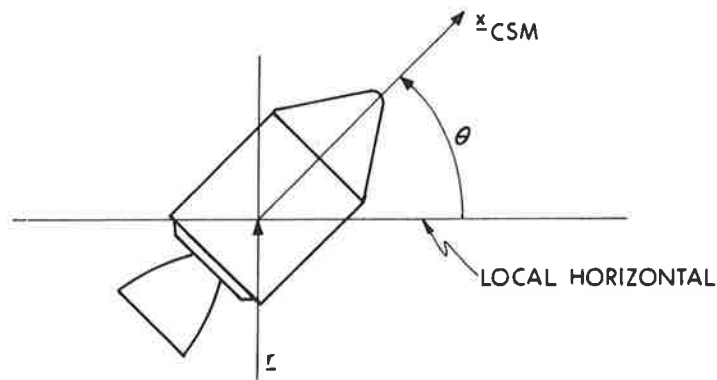


Figure 6.7-1 Definition of Theta

The equations used to compute the display parameters for R31 are given below.

$$\begin{aligned} \underline{R} &= \underline{r}_W - \underline{r}_C \\ \underline{u}_R &= \text{UNIT}(\underline{R}) \\ \text{RANGE} &= |\underline{R}| \end{aligned} \tag{6.7.1}$$

$$\text{RANGE RATE} = (\underline{v}_W - \underline{v}_C) \cdot \underline{u}_R \tag{6.7.2}$$

To compute θ the following vector is defined

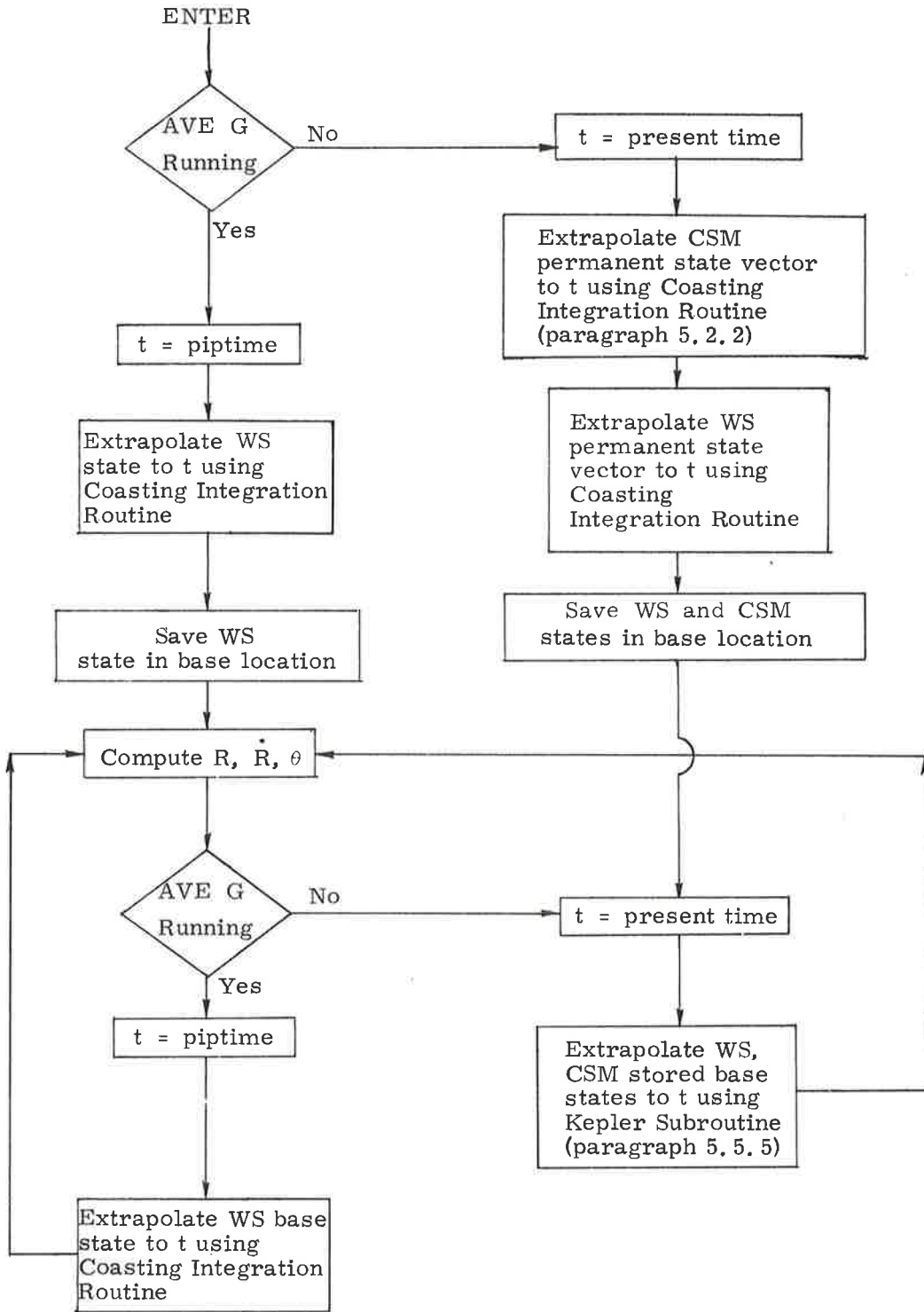


Figure 6.7-2. Range, Range Rate, θ

$$\underline{u}_X = [\text{REFSMMAT}]^T [\text{NBSM}] \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

where NBSM and REFSMMAT are defined in paragraph 5.6.3 and \underline{u}_X is a unit vector along the X-body axis expressed in basic reference coordinates. The angle θ is then found as follows:

$$\begin{aligned} \underline{u} &= \text{UNIT}(\underline{r}_C \times \underline{v}_C) \\ \underline{u}_P &= \text{UNIT} \left[\underline{u}_X - \left(\frac{\underline{u}_X \cdot \underline{r}_C}{r_C^2} \right) \underline{r}_C \right] \\ \theta &= \cos^{-1} \left[\underline{u}_X \cdot \underline{u}_P \text{SGN}(\underline{u}_P \cdot \underline{u} \times \underline{r}_C) \right] \end{aligned} \tag{6.7.3}$$

$$\text{If } \underline{u}_X \cdot \underline{r}_C < 0; \theta = 2\pi - \theta$$

The three displays of R31 are automatically updated until R31 is terminated by the astronaut. The logic flow required to accomplish this update is shown in Figure 6.7-2.

5.6.7.2 Range, Range Rate, Phi Display (R34)

Routine R34 may be initiated by the astronaut to display the computed range and range rate between the two vehicles as well as an angle ϕ shown in Figure 6.7-3.

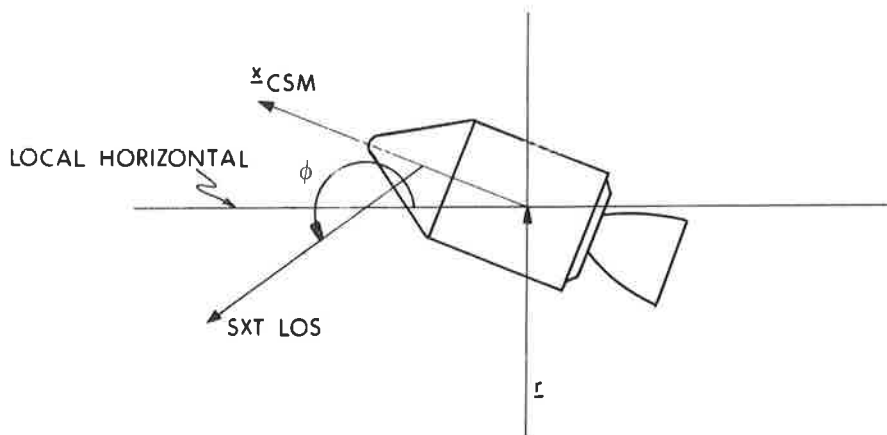


Figure 6.7-3. Definition of Phi

The angle ϕ represents the angle between the SXT line of sight and the local horizontal plane. It is determined in the same way as θ in Equation (6.7.3) except that

$$\underline{u}_X = [\text{REFSMMAT}]^T [\text{NBSM}] \underline{u}_{\text{NB}}$$

where $\underline{u}_{\text{NB}}$ is the sextant line of sight in navigation base coordinates as given in paragraph 5.6.3.1.2.

The three displays of R34 are automatically updated in a manner similar to the R31 update.

5.6.7.3 Final Attitude Display (R63)

Routine R63 may be used to compute and display the gimbal angles required to point a specified spacecraft vector, SCAXIS (as defined in paragraph 5.2.5.1), at the WS. After initiation of this routine the state vectors of both vehicles are extrapolated to the present time plus one minute. Based on these new state vectors the required gimbal angles are computed and displayed.

There is no automatic display update. However, R63 can easily be recycled to manually accomplish an update of the display.

5.6.7.4 Out-of-Plane Rendezvous Display (R36)

Routine R36 can be used during any phase of the rendezvous sequence to provide information about the out-of-plane position and velocity of the CSM relative to the WS orbital plane and the out-of-plane velocity of the WS relative to the CSM orbital plane. These three quantities are computed for a given time. The present time of ignition is automatically selected in the absence of a time input by the astronaut.

The definition of the three quantities computed is given by the following set of equations where \underline{r}_{CM} , \underline{v}_{CM} is the state vector of the CSM and \underline{r}_{WS} , \underline{v}_{WS} is the state vector of the WS.

$$\underline{u}_{CM} = \text{UNIT} (\underline{v}_{CM} \times \underline{r}_{CM}) \quad (6.7.4)$$

$$\underline{u}_{WS} = \text{UNIT} (\underline{v}_{WS} \times \underline{r}_{WS}) \quad (6.7.5)$$

$$\dot{\underline{Y}}_{CM} = \underline{v}_{CM} \cdot \underline{u}_{WS} \quad (6.7.6)$$

$$\underline{Y}_{CM} = \underline{r}_{CM} \cdot \underline{u}_{WS} \quad (6.7.7)$$

$$\dot{\underline{Y}}_{WS} = \underline{v}_{WS} \cdot \underline{u}_{CM} \quad (6.7.8)$$

5.6.8 Automatic Optics Positioning Routine

5.6.8.1 General

This routine (corresponding to R52 of Section 4) is used during alignment and navigation procedures to automatically point the optics in the direction of a specified tracking target to aid the astronaut in the acquisition of the target. The two target modes available in this routine are as follows:

1. Star mode
2. WS mode

The calculations and positioning commands in this routine are repeated approximately every two seconds provided the optics mode switch is set to CMC. The routine consists of two subroutines, Line-of-Sight and Command. Let t be the time (GET) at the beginning of a cycle and let

$$t_{\delta} = t + \delta t \quad (6.8.1)$$

where δt is a lead term to compensate for system lags. During the first part of the cycle the Line-of-Sight Subroutine computes the desired line-of-sight vector at time t_{δ} , \underline{u}_{LOS} . The Command Subroutine then computes updated values of the desired optics angles from the desired line-of-sight vector, \underline{u}_{LOS} , and the current vehicle attitude. The entire procedure is then repeated. A basic CMC routine drives the optics to the orientation indicated by the desired angles.

5.6.8.2 Line-of-Sight Subroutine

5.6.8.2.1 Star Mode. —This mode is used during IMU alignment procedures to point the optics in the direction of a specified celestial body. The desired line-of-sight vector, \underline{u}_{LOS} , is the specified unit vector of the celestial body which was either obtained from fixed memory, computed by the CMC, or loaded by the astronaut.

5.6.8.2.2 WS Mode. —This mode is used during workshop (WS) tracking phases to point the optics in the direction of the WS. The Kepler Subroutine (paragraph 5.5.5) is used to compute \underline{r}_C and \underline{r}_W , the estimated conic position vectors of the CSM and the WS, respectively, at time t_δ . Then, the desired line-of-sight vector is given by

$$\underline{u}_{LOS} = \text{UNIT}(\underline{r}_W - \underline{r}_C) \quad (6.8.2)$$

5.6.8.3 Command Subroutine

During this subroutine the line-of-sight vector, \underline{u}_{LOS} , is transformed to stable member coordinates by means of

$$\underline{s}_{SM} = [\text{REFSMMAT}] \underline{u}_{LOS} \quad (6.8.3)$$

The desired optics trunnion (precision) and shaft angles, TA and SA, respectively, are computed as described in paragraph 5.6.3.1.3.

5.6.9 Optics Angles Transform Routine

The Optics Angles Transform Routine (R64 of Section 4) computes and displays the tracking angles that correspond to given sextant shaft and trunnion angles. The computed tracking angles specify the spacecraft axis to be pointed by the tracking program in use when R64 is called.

Given the shaft and trunnion angles SA and TA, loaded by the astronaut, the corresponding tracking angles γ and ρ (yaw and pitch, respectively) are computed as follows:

$$\begin{aligned} \text{Pitch } (\rho) &= -\sin^{-1} u_{NB,3} \\ \text{Yaw } (\gamma) &= \tan^{-1} \left(\frac{u_{NB,2}}{u_{NB,1}} \right) \end{aligned}$$

where \underline{u}_{NB} is the sextant line of sight in navigation base coordinates as given in paragraph 5.6.3.1.2.

5.6.10 Orbital Parameter and Entry Display Computations

This paragraph presents the CMC computations required for the displays in the following routines and programs:

- a. The Orbital Parameters Display Routine R30, which is callable by the astronaut via an extended verb.
- b. The Splash Error Computation Subroutine DELRSPL, which is automatically called by R30 when either the Earth Orbital Insertion Monitor Program (P11) or the CMC Idling Program (P00) is running.
- c. The Entry Preparation Program (P61), which provides displays to initialize the EMS.
- d. The CM-SM Separation and Pre-Entry Maneuver Program (P62) and the Entry Ballistic Program (P66), which display IMU gimbal angles corresponding to current CM hypersonic trim attitude.
- e. The Entry Final Phase Program (P67), which displays the current CM geographic location.

5.6.10.1 Orbital Parameters Display Routine(R30)

The Orbit Parameters Display Routine (R30) can be called by the astronaut via an extended verb (V82) to compute and display the orbital parameters defined below. This display is automatically updated only when Average-G is running. The option to select the vehicle for which these orbit parameters are to be displayed and the time of the state vector to be used for these calculations, however, will be available only when Average-G is not running.

Normally, the apocenter altitude, pericenter altitude, and the time from a reference altitude (300,000 ft above launch-pad radius) are displayed. If the chosen orbit does not intercept this altitude, the third display is -59B59. Under certain circumstances—explained in Section 4—an additional display of time from pericenter can also be requested by the astronaut. The details of each option and the acceptable astronaut responses are discussed in Section 4.

The computational logic and equations used in R30 are shown in Figure 6.10-1. The following is a list of parameter definitions that apply to these figures.

\underline{r}	}	State vector of the selected vehicle
\underline{v}		
h_a		Apocenter altitude
h_p		Pericenter altitude
r_{LP}		Earth launch-pad radius
t_{ff}		Time from a reference altitude
t_{PER}		Time from pericenter
SPLERROR		See paragraph 5.6.10.2

The TFF Subroutines, TFFCONIC, TFFRP/RA, CALCTPER, and CALCTFF, which are called by R30 to perform the mathematical computations, are described in paragraph 5.6.10.4. The Splash Error Computation Subroutine DELRSPL, which is also called by R30 if P11 or P00 is running, is described in paragraph 5.6.10.2.

It should be noted that R30 and DELRSPL use the standard values of the gravitational constants. The TFF Subroutines, however, use "adjusted" values for the near-earth trajectory computations involved in the P61 displays.

As a convenience, the display conventions for t_{ff} and t_{PER} , which are defined in Section 4, are summarized here. The display conventions differ, depending on whether or not Average-G is running.

During Average-G, t_{ff} and t_{PER} represent the transfer times between the present state vector (supplied by Average-G) and the specified terminal radius (to the reference altitude or to pericenter). The displays are periodically recomputed, and change as Average-G updates the state vector. The displayed transfer times are the negative of the values calculated by the TFF Subroutines, as in count-down. When the state vector supplied by Average-G lies between perigee and terminal radius, a positive time representing time-since-terminal-point is displayed. When the supplied position passes beyond perigee, negative time to terminal radius is displayed once again. For outbound hyperbolas or parabolas, however, positive time since terminal point is displayed.

When Average-G is not running, a single state vector is supplied to the TFF Subroutines, and the calculations are performed only once. The crew designates which vehicle state vector is to be used and the GET of the state vector. Precision integration obtains the designated state vector and supplies it to the TFF Subroutines. The transfer time between the supplied position and the terminal radius is computed, as described in paragraph 5.6.10.4. This transfer time is corrected to account for the difference between the specified state vector time and the present time. Thus t_{ff} and t_{PER} reflect the time of freefall from the present position of the specified vehicle to the terminal radius. Since the transfer time is computed only once, a 1-second clock is used to update the display.

The sign of the display depends on the position of the specified vehicle with respect to the terminal radius. Thus, if the specified vehicle is approaching terminal radius, the displayed transfer time is negative. The display is incremented each second, counting towards zero. If the specified vehicle has passed terminal radius, the displayed transfer time is positive, and increases as it is incremented by the 1-second clock. In either case, any change in the display is due to the 1-second clock; and the sign of the display changes only if the value passes through zero.

When Average-G is not running, the crew can designate a new time for the state vector by exercising the recycle option (V32E in response to V1G N44). A new time of transfer is computed and displayed, according to the conventions given above.

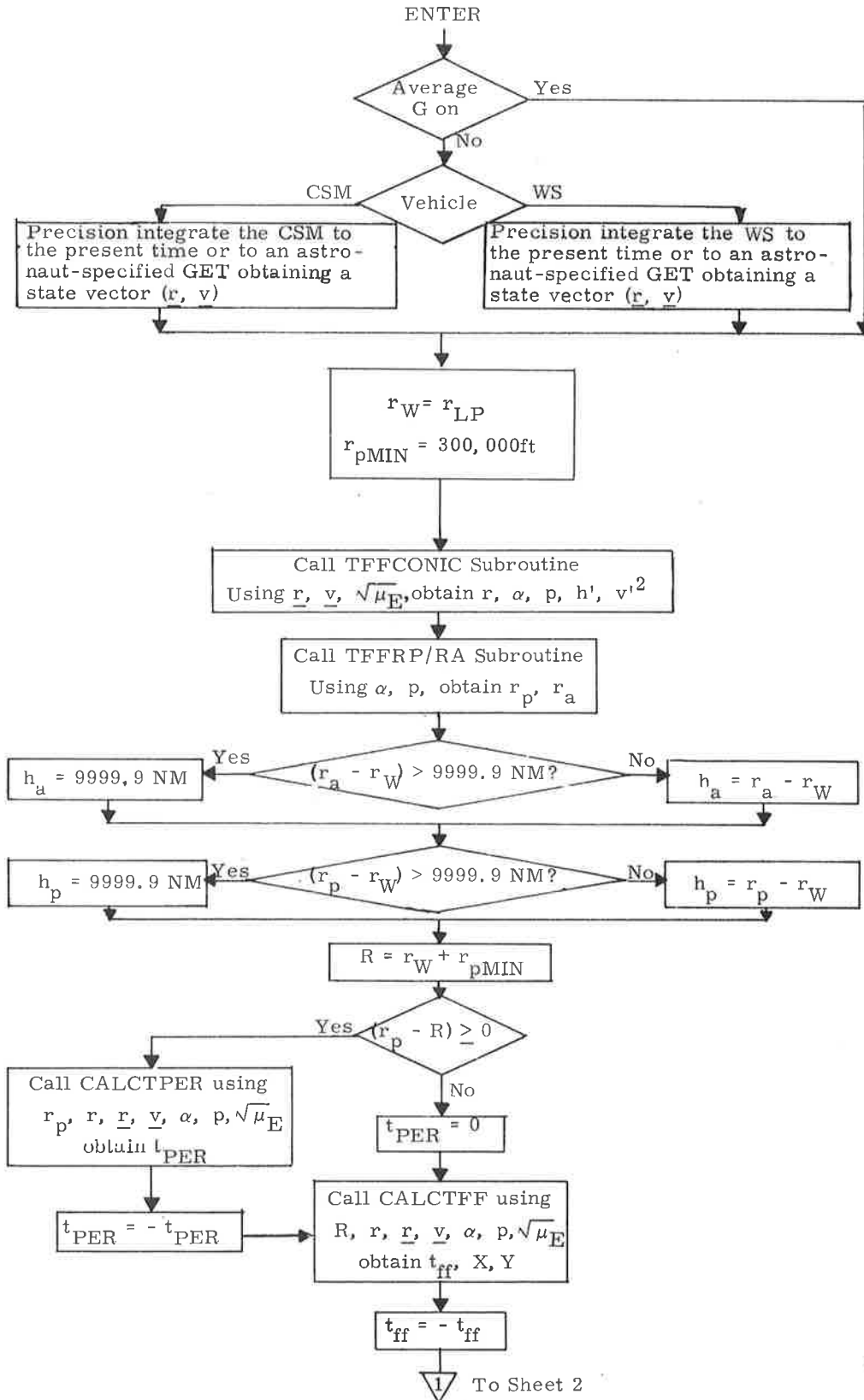


Figure 6.10-1. Orbital Parameters Display Routine (Sheet 1 of 2)

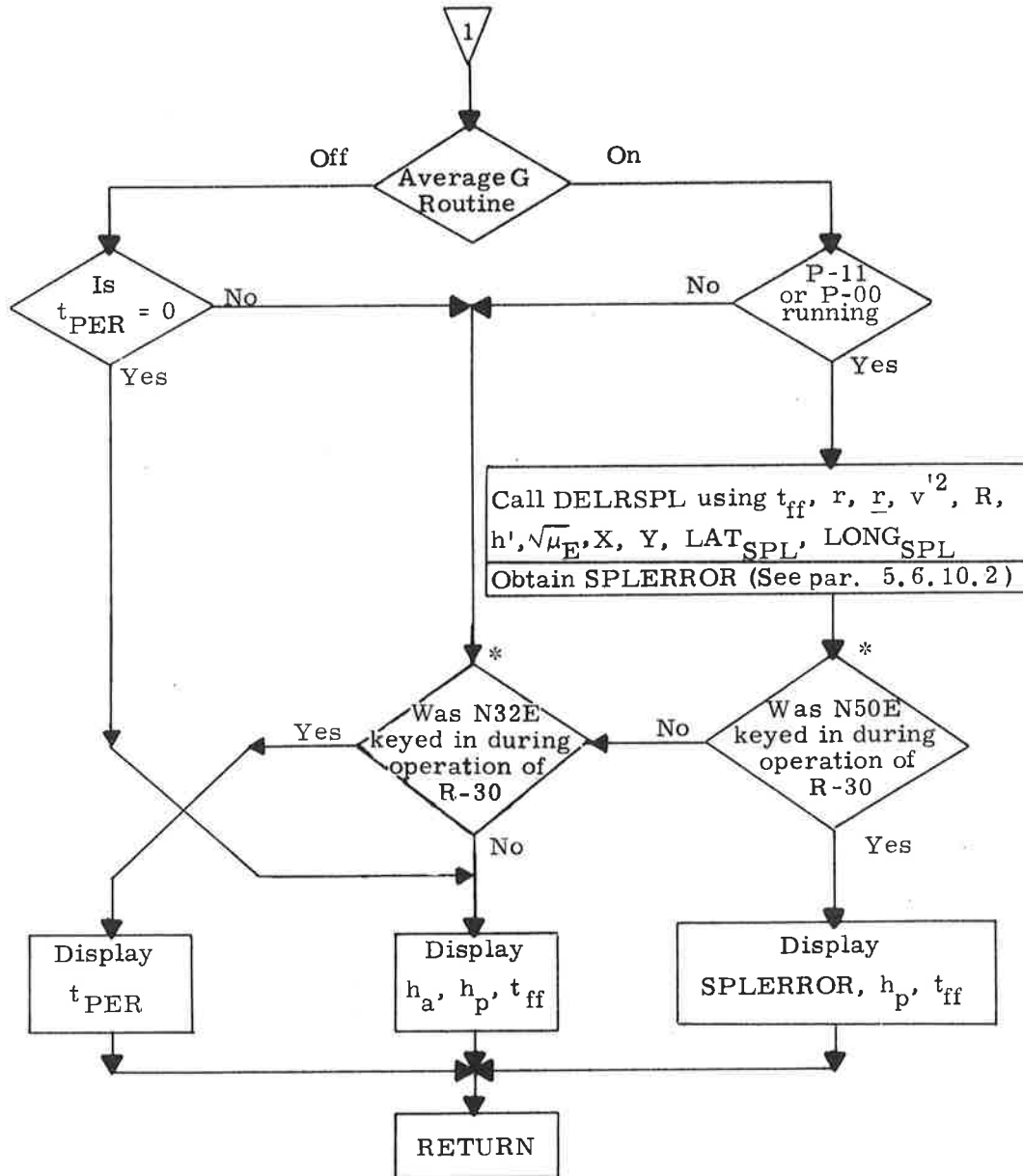


Figure 6.10-1. Orbital Parameters Display Routine (Sheet 2 of 2)

NOTE.— The values of t_{ff} and t_{PER} as displayed on the DSKY are first corrected for the difference between the state vector time and the present time before being made to actually "count down" every second, i. e., the DSKY displays are automatically updated once a second, but only if Average-G is not on. This is not shown explicitly in the above figure.

*These blocks are indicative of use of routine. For complete specification of DSKY interface, see Section 4 GSOP.

5.6.10.2 Splash Error Computation Subroutine (DELRSP)

The Splash Error Computation Subroutine (DELRSP) (Figure 6.10-2) is automatically called by the Orbital Parameters Display Routine R30 if the Earth Orbital Insertion Monitor Program (P11) or P00 is running.

DELRSP provides an approximate indication of landing point miss distance, SPLERROR. The error is computed as the difference between an approximation to the range to be covered and the range to the predicted target point. This assumes that all range components lie in the same plane. The range to the predicted target point is the distance between the present position and the target—as specified by the target coordinates and an estimate of the time of impact. The flight time is the sum of the conic freefall time to 300,000 ft altitude and an estimate of the entry duration. The approximate range to be covered is the sum of the conic freefall range to 300,000 ft altitude and an estimate of the entry range. If the conic path from the present position does not intercept the 300,000 ft altitude, however, or if the vehicle's present location has already passed below the intercept altitude, the miss distance is defined and displayed as minus the range to the target. In both cases, the sign convention is that negative ranges correspond to undershooting the target.

DELRSP uses the VGAMCALC and TFF/TRIG Subroutines, which are described in paragraph 5.6.10.4. It also uses one of the Latitude-Longitude Subroutines described in paragraph 5.5.3 for converting a known latitude and longitude into a position vector. Further, DELRSP uses the empirical formulas for the entry range and entry time given by the AUGKUGEL Subroutine. (See Figure 6.10-3.)

The coefficient 3437.7468, used twice in the DELRSP calculations, is a conversion factor of the reentry calculations, and is defined in Subsection 5.7 of this GSOP. (It is referred to as ATK in that subsection.)*

The parameters LAT_{SPL} and $LONG_{SPL}$ are the geodetic latitude and longitude of the splash point. These quantities are either pad loaded, uplinked, or entered via the DSKY by the astronaut.

It should be noted that the Earth Orbital Insertion Monitor Program (P11) has automatic displays that are not associated with R30 or DELRSP. These displays are described in paragraph 5.3.5.6.

* See note on page 5.7-2.

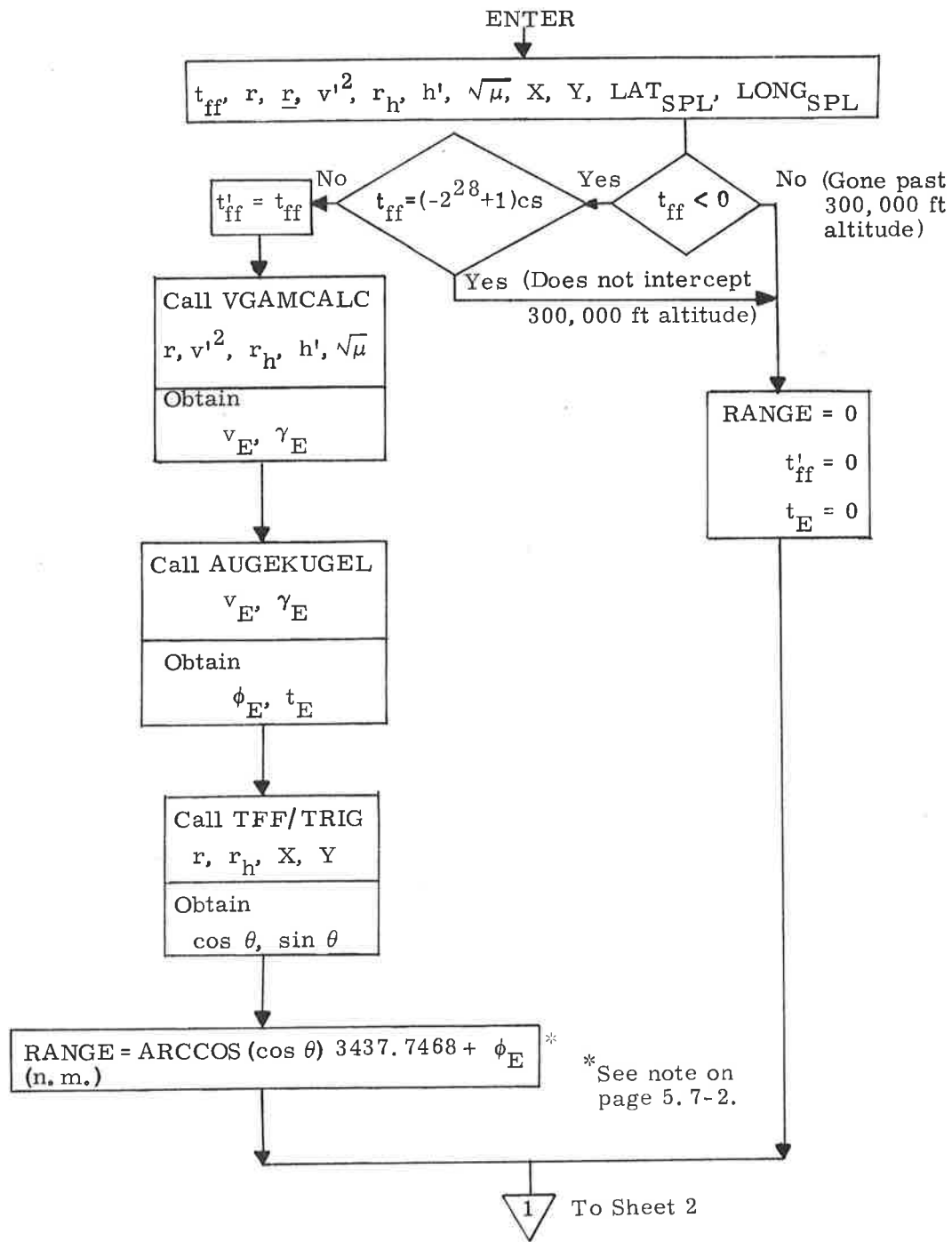
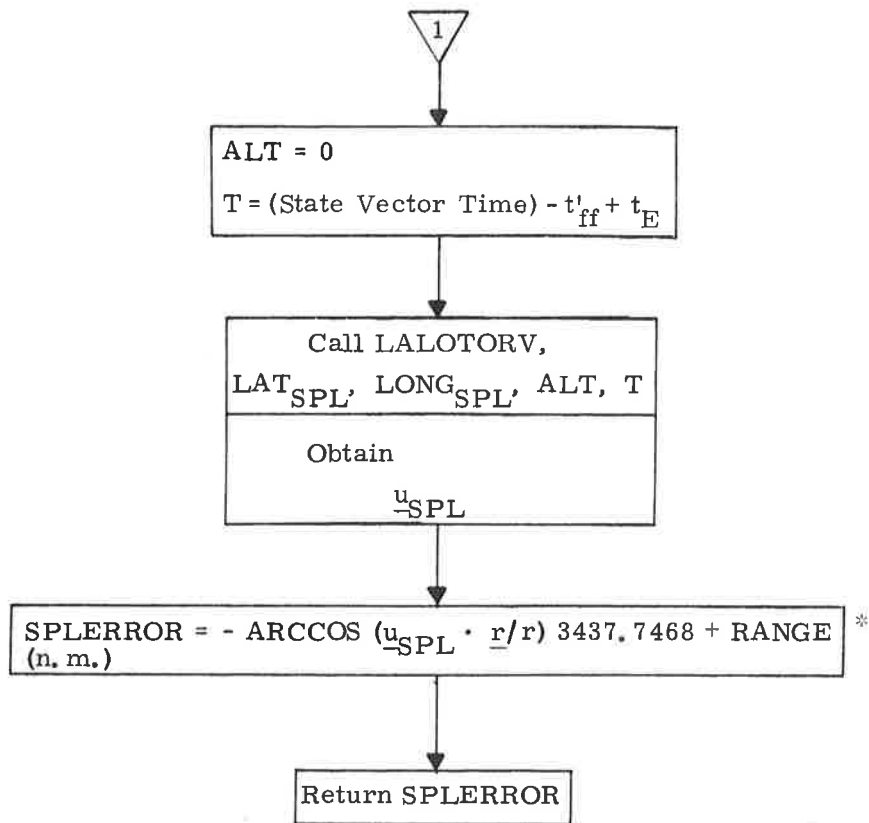


Figure 6.10-2. Splash Error Computation Subroutine (Sheet 1 of 2)



* See note on page 5.7-2.

Figure 6.10-2. Splash Error Computation Subroutine (Sheet 2 of 2)

AUGEKUGEL

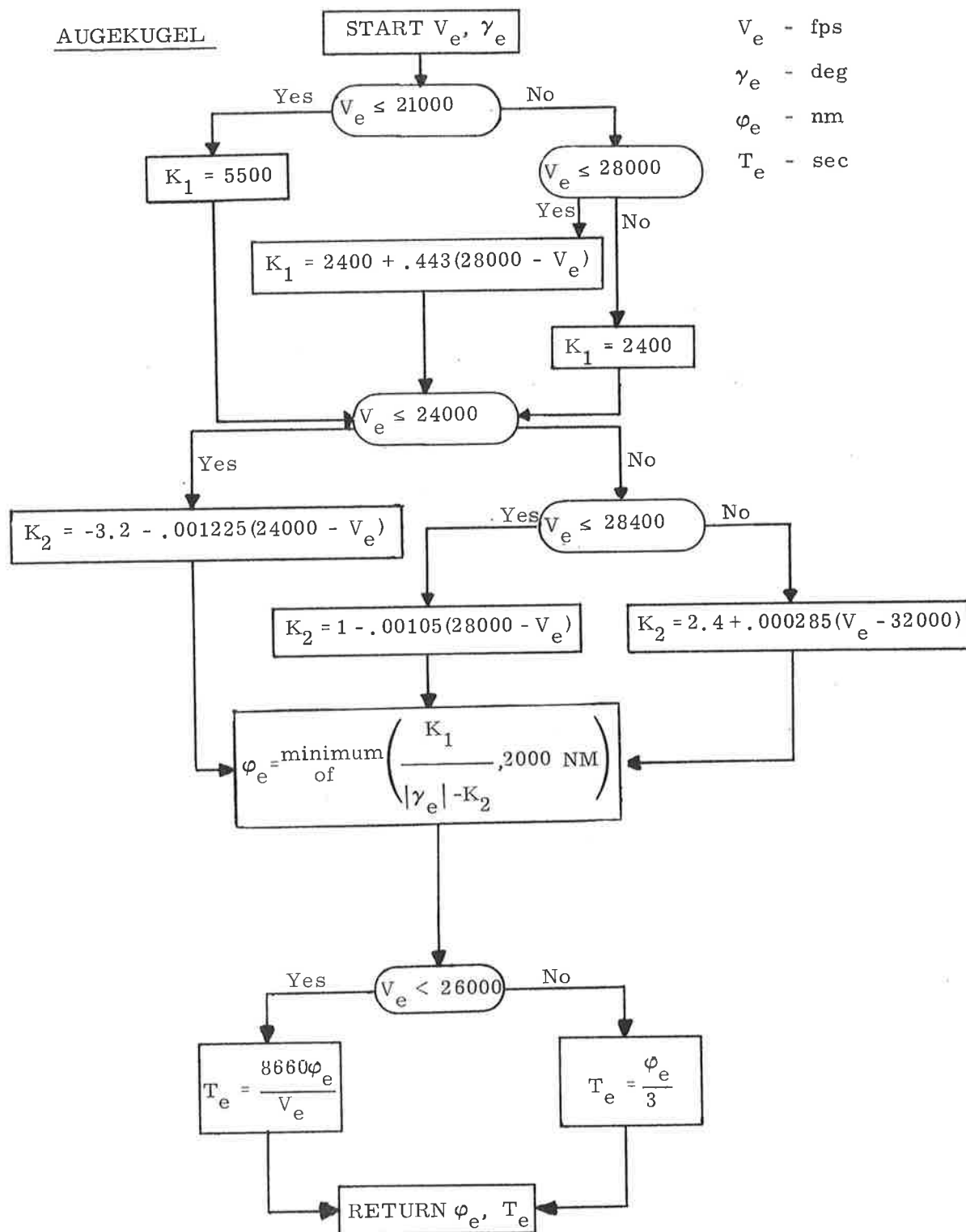


Figure 6.10-3, Empirical Relations for Entry Range

5.6.10.3 The Entry Preparation Program (P61) Displays

The following display computations are made in program P61:

- GMAX — Maximum predicted deceleration for entry at the nominal roll angle ($L/D = 0.3$).
- VPRED — Predicted inertial velocity at an altitude of 400,000 feet.
- GAMMAEI — Predicted flight path angle between the inertial velocity vector and the local horizontal at an altitude of 400,000 feet.

The 400,000-ft. altitude is referenced to the Fischer ellipsoid, and a two-step iteration is made to calculate this radial length.

The following three display parameters are required to initialize the entry monitor system (EMS):

- RTGO — Range-to-go from the EMS altitude to the predicted splash point.
- VIO — Predicted inertial velocity at the EMS altitude.
- TTE — Time-to-go from the current time to the EMS altitude (negative and decreasing). TTE is decremented every two seconds until 0.05g is sensed.

The EMS altitude is measured above the Fischer ellipsoid and is stored in erasable memory during the prelaunch erasable load. (EMSALT)

The principal computation required for these display parameters is the conic calculation of time to a radius of specified length. The intersection of a conic with an altitude above the Fischer ellipsoid is a two-step iteration. An initial calculation is made using an estimate of the radius length. The latitude of the conic with this assumed length is then used to calculate the radius of the Fischer ellipsoid. This calculated radius is then used to calculate a second radius length; and display quantities are referred to this second radius. Additional iterations are not required to achieve the necessary display accuracy.

The computation logic for the P61 displays is presented in Figure 6.10-4. These computations use the TFF Subroutines that are described in paragraph 5.6.10.4.

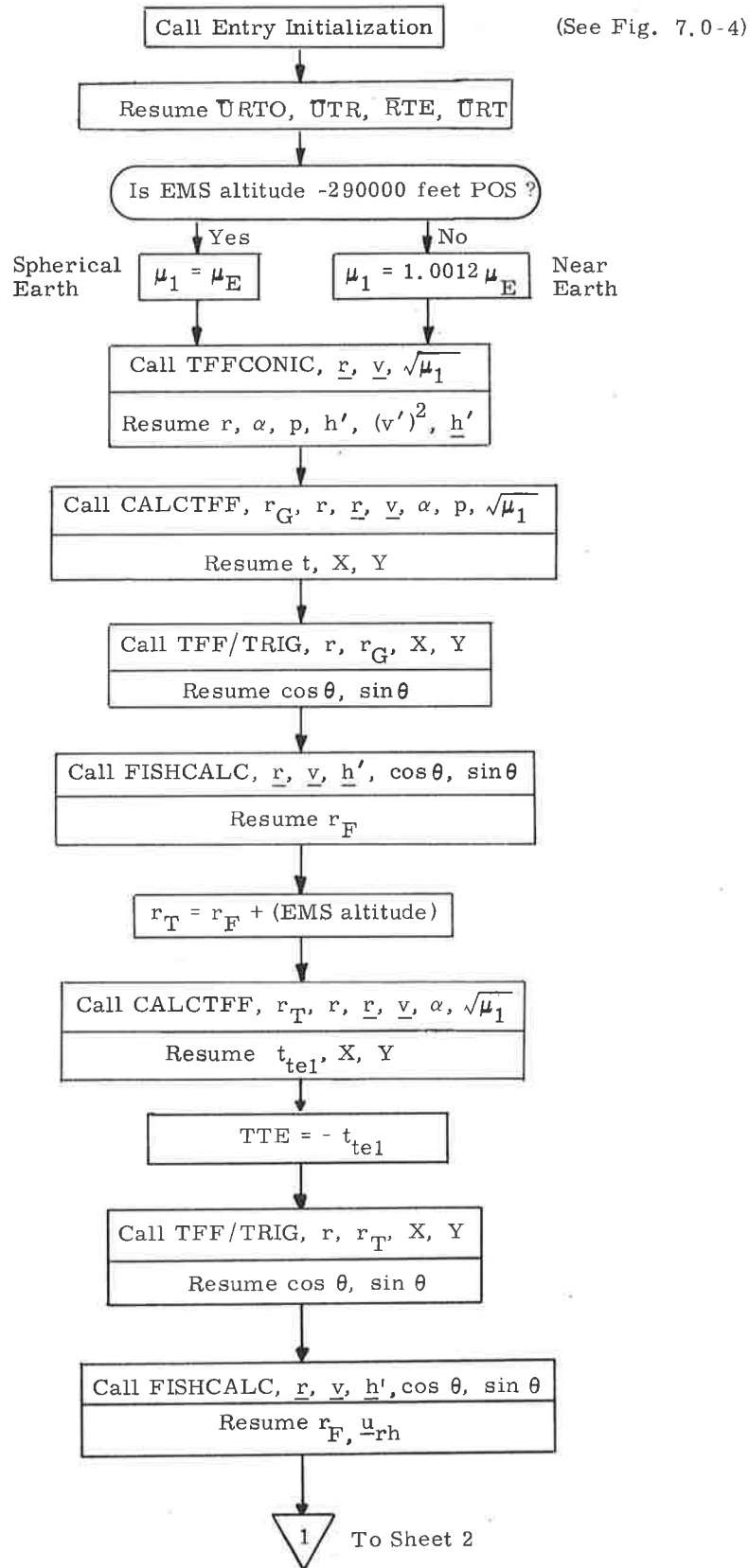


Figure 6.10-4. Display Calculations for P61 (Sheet 1 of 3)

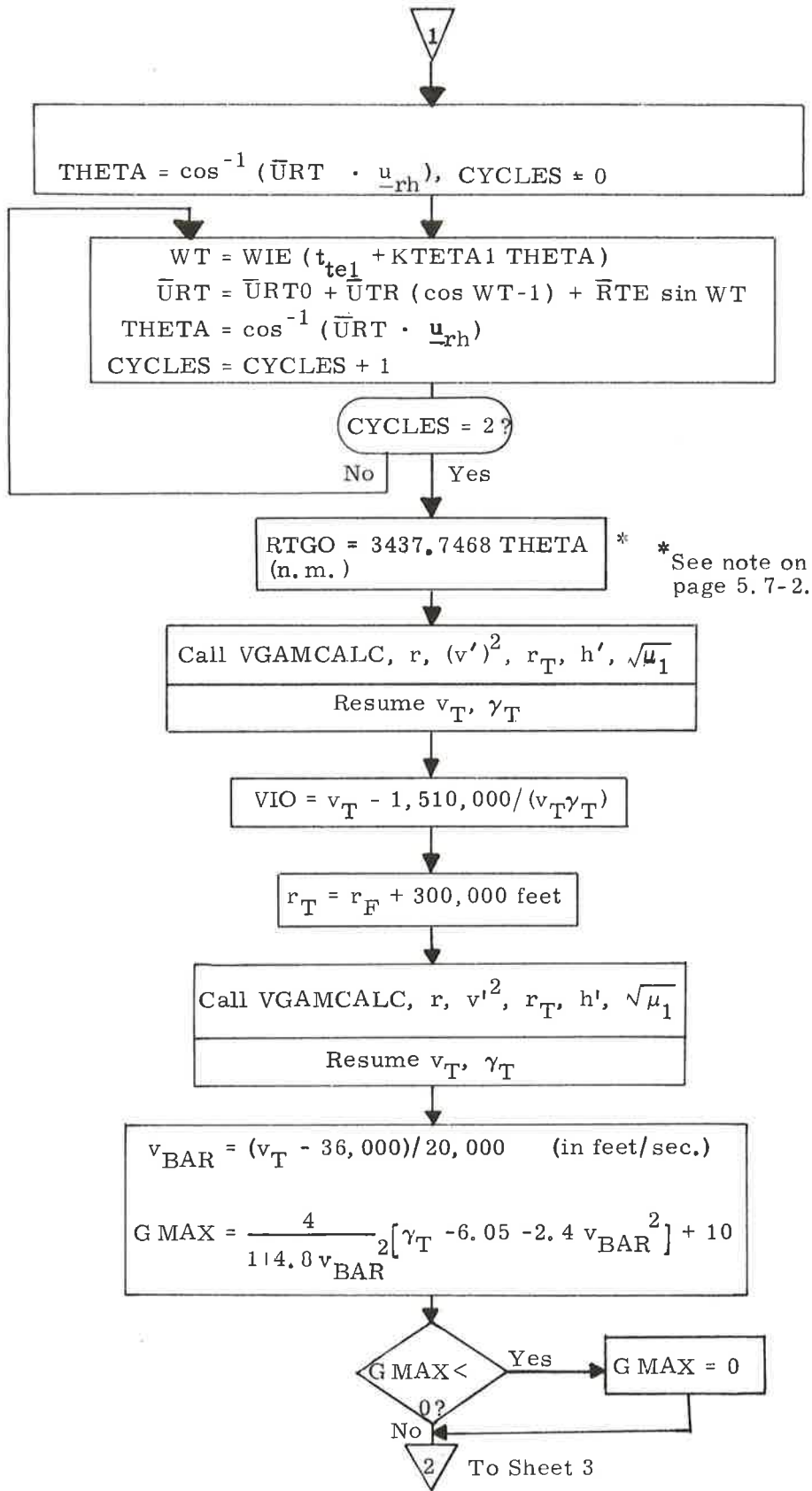
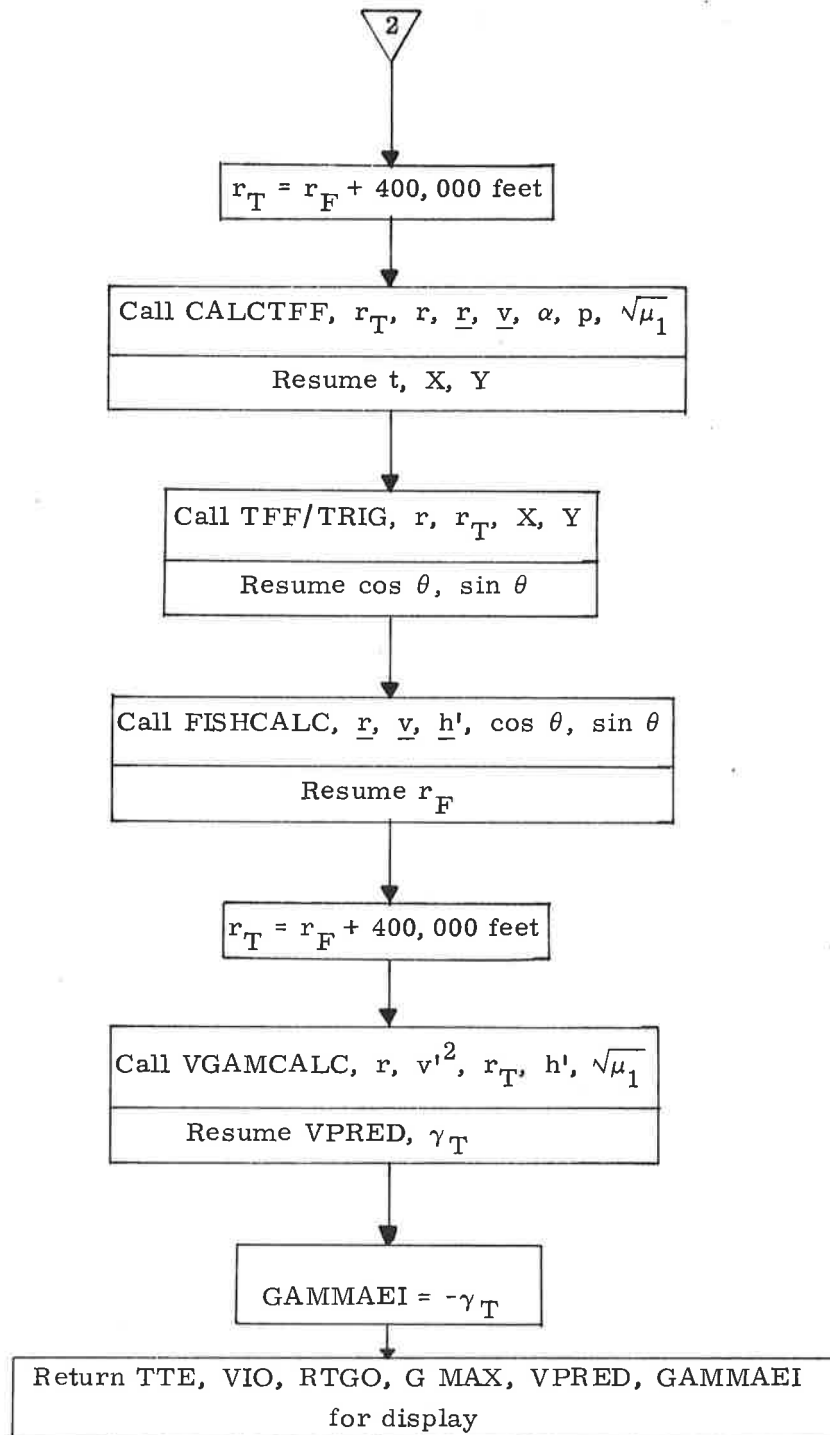


Figure 6.10-4. Display Calculations for P61 (Sheet 2 of 3)



* See note on page 5.7-2.

Figure 6.10-4. Display Calculations for P61 (Sheet 3 of 3)

The earth gravitation constant used in these calculations can be increased to take into account missions that are confined to low inclination angles from the equator where oblate earth effects can be significant.* The drag loss in reaching the EMS point (0.05g) is also approximated for these display computations.

Besides the TFF Subroutine symbols (which are defined in paragraph 5.6.10.4), and the symbols that are defined above, the following additional symbols are used in Figure 6.10-4.

- THETA (θ) — The angle between the present position and the initial target vector at the nominal time of arrival. (This is a reentry variable; see subsection 5.7.)
- KTETA1 — Time of flight constant = 1100
- $\left. \begin{array}{l} \underline{r} \\ \underline{v} \end{array} \right\}$ — State vector of the vehicle at the present time
- r_G — First estimate of the terminal radius
 = $r_{PAD} + 284,643$ feet
 = 21,194,545 feet
 (since $r_{PAD} = 6,373,338m = 20,909,902$ ft)
- μ_1 — Gravitational constant used with an adjusted value that is 0.12 percent greater than the standard value of the gravitational constant, used only if the EMS altitude is less than a specified value.
- μ_E — Gravitational constant for spherical earth.

5.6.10.4 The TFF Subroutines

The TFF Subroutines are used to calculate various conic parameters, such as the semi-latus rectum, the semi-major axis, apocenter and pericenter radii, the transfer time from an initial or present position to specified terminal radii (which may include apocenter and pericenter), the transfer angle of such transfers, and the velocity magnitude and flight-path angle at the terminal position. For these subroutines, the terminal radius is supplied by the calling program, and defined to be on the inbound side of the trajectory.

*The choice to use the spherical earth gravitational constant or the adjusted constant depends upon the value of the pad-loaded EMS altitude.

The sign conventions used by the TFF Subroutines are defined by the logic flow of paragraph 5.6.10.5. To summarize these conventions it is convenient to think of a succession of points along a conic path, each representing a state vector that may be supplied to the TFF Subroutines by the calling program. This point can be the present state vector of the CSM or WS; it can be an estimation; or it can be a previous or future state vector as calculated by the calling program. If the supplied point occurs before the terminal radius, a positive (decreasing) transfer time is computed by the TFF Subroutines. In the event that the supplied position lies between the terminal radius and perigee, a negative (increasing) time representing time since terminal point passage is computed. When the supplied point occurs beyond perigee, positive (decreasing) time to (subsequent) terminal radius is computed for ellipses. However, for the open-ended conics—outbound hyperbolas or parabolas—negative (increasing) time since terminal point passage is computed. Since the supplied state vector may not be the current vehicle state vector, and since a count-down display is preferred by the crew, the transfer time, as displayed on the DSKY, can, in general, differ in sign and magnitude from that calculated by the TFF Subroutines. The display conventions are defined in Section 4, and are repeated, for convenience, where appropriate in Section 5. (See, for example, R30.)

The TFFCONIC Subroutine (Figure 6.10-5) calculates various conic parameters and stores them for short-term use. It also computes other intermediate quantities to avoid duplication by following subroutines.

The TFFRP/RA Subroutine (Figure 6.10-6) calculates pericenter and apocenter radii for the general conic. For hyperbolic or parabolic orbits, or for those elliptic orbits in which the apocenter radius exceeds the scaling, r_a is set equal to $r_{a \text{ MAX}}$, which corresponds to the largest expressible number in the double-precision CMC word.

The CALCTFF/CALCTPER Subroutine (Figure 6.10-7) calculates the time of freefall to a given radius or to pericenter depending on the entry point to the subroutine. The subroutine also calculates the quantity Y required by the TFF/TRIG routine in its computation of the transfer angle. In the event that the terminal radius does not lie on the present conic, t is set to the largest time expressible in double precision, $2^{28} - 1$ csec, as a flag. After TFFCONIC has been called, CALCTFF/CALCTPER can be called, as desired, with different terminal radii to obtain the various times of freefall to these radii for the same conic.

The TFF/TRIG Subroutine (Figure 6.10-8) calculates the sine and cosine of the transfer angle θ from the initial position vector to the terminal position vector. The computation of the intermediate parameter Y is done during the appropriate branch in CALCTFF/CALCTPER and is saved temporarily. Thus, the caller of TFF/TRIG can make only one call for each call to CALCTFF/CALCTPER.

FISHCALC Subroutine (Figure 6.10-9) calculates the sine of the geocentric latitude of the terminal position vector, and then calls the Fischer Ellipsoid Radius Routine (part of the Latitude-Longitude Subroutine of paragraph 5.5.3) to obtain the radius of the Fischer ellipsoid, using Equation (5.3.1).

VGAMCALC Subroutine (Figure 6.10-10) calculates the velocity magnitude at the terminal position and the flight-path angle relative to the local horizontal at the terminal position. The user of the subroutine must supply the proper sign to the flight-path angle.

Nomenclature of the TFF Subroutines:

\underline{r} or \underline{r}_0	}	Present or initial state vector
\underline{v} or \underline{v}_0		
μ	Gravitational constant	
α	Reciprocal of the semi-major axis of conic (negative for hyperbolas)	
p	Semi-latus rectum of conic	
e	Eccentricity of conic	
r_p	Pericenter radius	
r_a	Apocenter radius	
r_h	Terminal position radius	
v_h	Velocity magnitude at terminal position	
γ_E	Flight-path angle relative to local horizontal at the terminal position	
S_{TFF}	Switch set to 0 or 1 according to whether the transfer time is calculated to the terminal position radius (CALCTFF) or to the pericenter (CALCTPER)	

θ	Transfer angle
r_F	Radius of Fischer ellipsoid along the terminal position unit vector
t	Conic transfer time from the present or initial position to the terminal position (CALCTFF) or to pericenter (CALCTPER)
r_{aMAX}	The largest apocenter radius expressible in the double precision CMC word at the relevant scaling.

5.6.10.5 Equations Used in Calculating the Conic Time of Flight

The time of flight along any conic trajectory proceeding from an arbitrary radius r_0 in the direction of velocity v_0 to a radius of specified length r_h is provided by Kepler's equation. The following formulation, while not quite universal, is explicit and direct. With a single equation, time of flight can be calculated for conic paths that are hyperbolic or parabolic and for those elliptic paths having $|\Delta E| \leq 90$ deg. An alternate equation is needed for the elliptic paths having $|\Delta E| > 90$ deg. Highly circular orbits are excluded by the constraint of length for the terminal radius. In the absence of a vector direction, highly circular cases are not treated.

The two conic parameters that appear in the time of flight equations are given by

Angular momentum	$\underline{H} = r_0 \times v_0$	
Semi-latus rectum	$p = \frac{\underline{H} \cdot \underline{H}}{\mu}$	(6.10.1)
Reciprocal semi-major axis	$\alpha = \frac{2}{r_0} - \frac{v_0 \cdot v_0}{\mu}$	

The equation for reciprocal semi-major axis α has the usual sign convention; α is negative for hyperbolic orbits, zero for parabolic orbits, and positive for elliptic orbits.

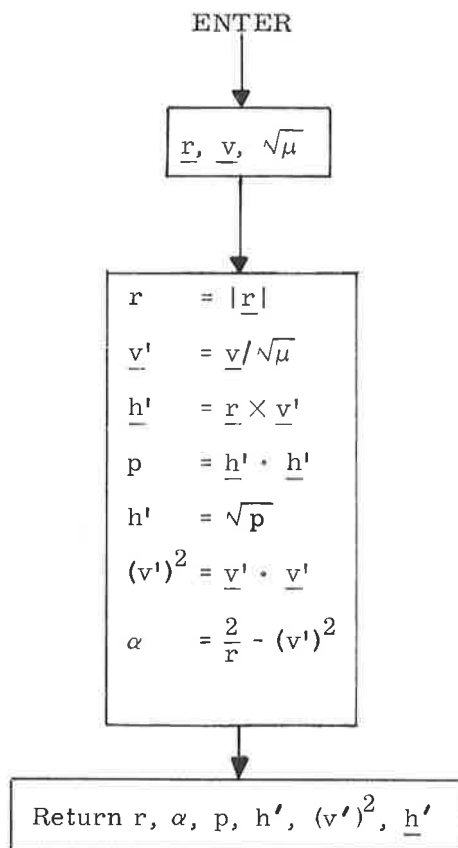


Figure 6.10-5. TFFCONIC Subroutine

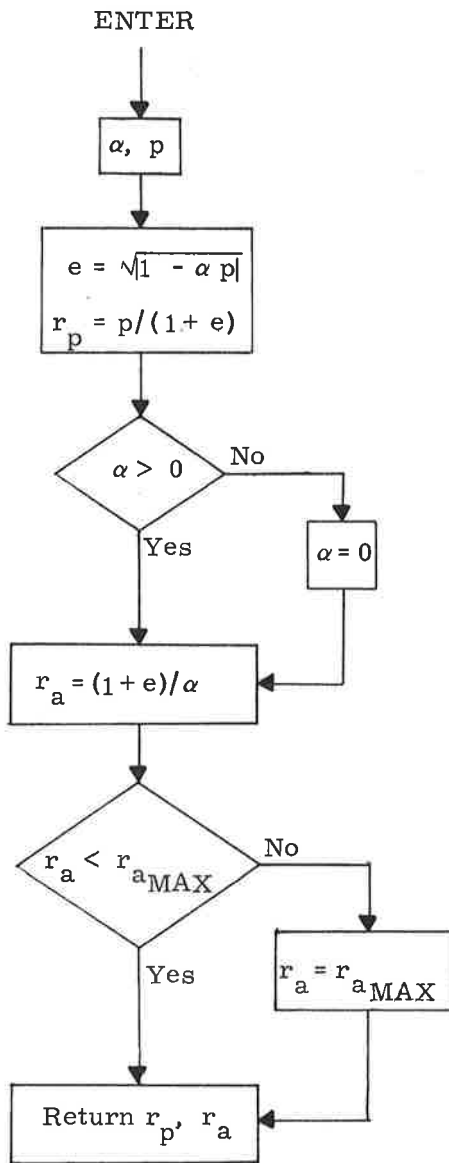
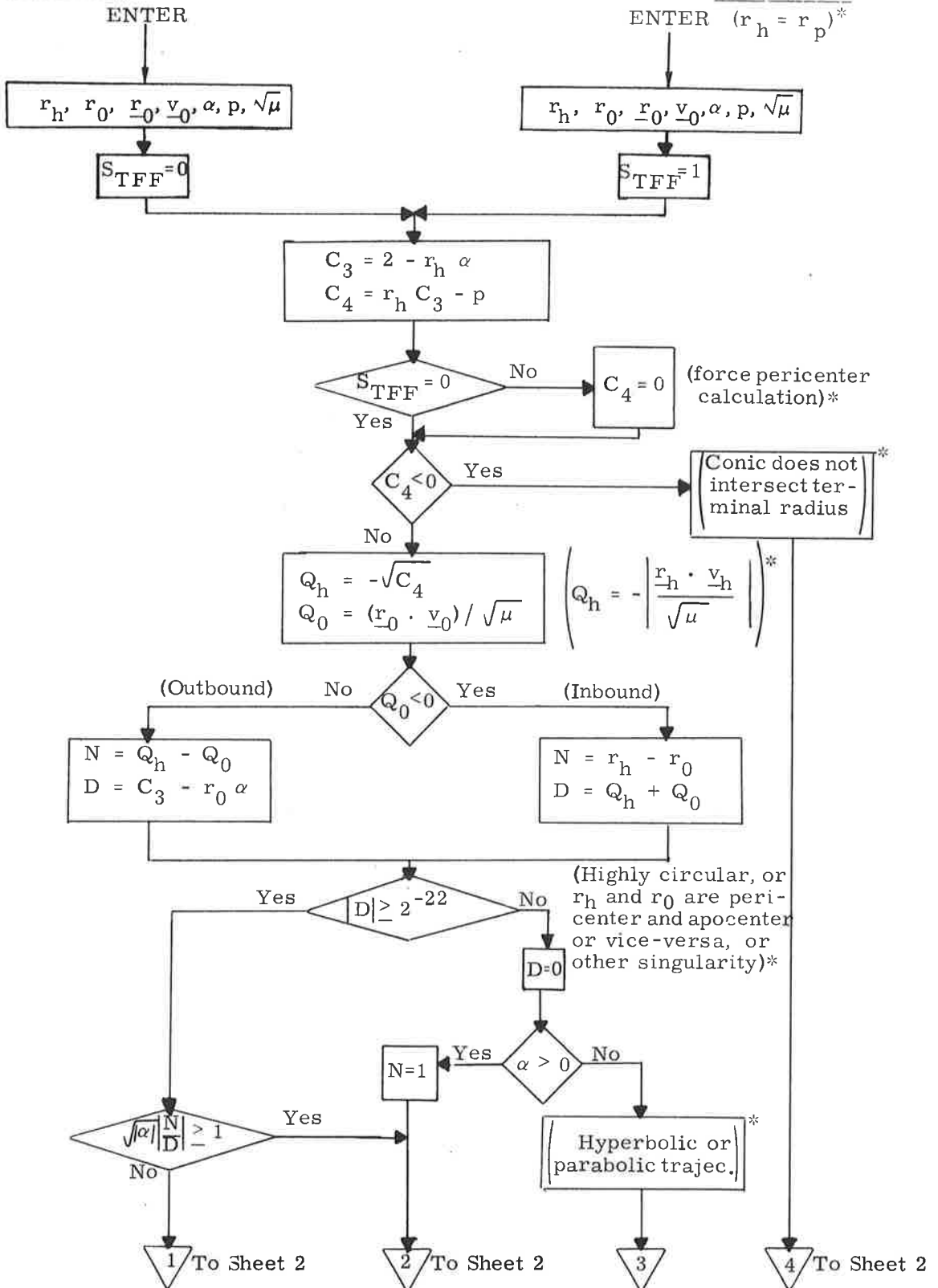


Figure 6.10-6. TFFRP/RA Subroutine

CALCTFF

CALCTPER



*Supplementary Information

Figure 6.10-7. CALCTFF/CALCTPER Subroutine (Sheet 1 of 2)

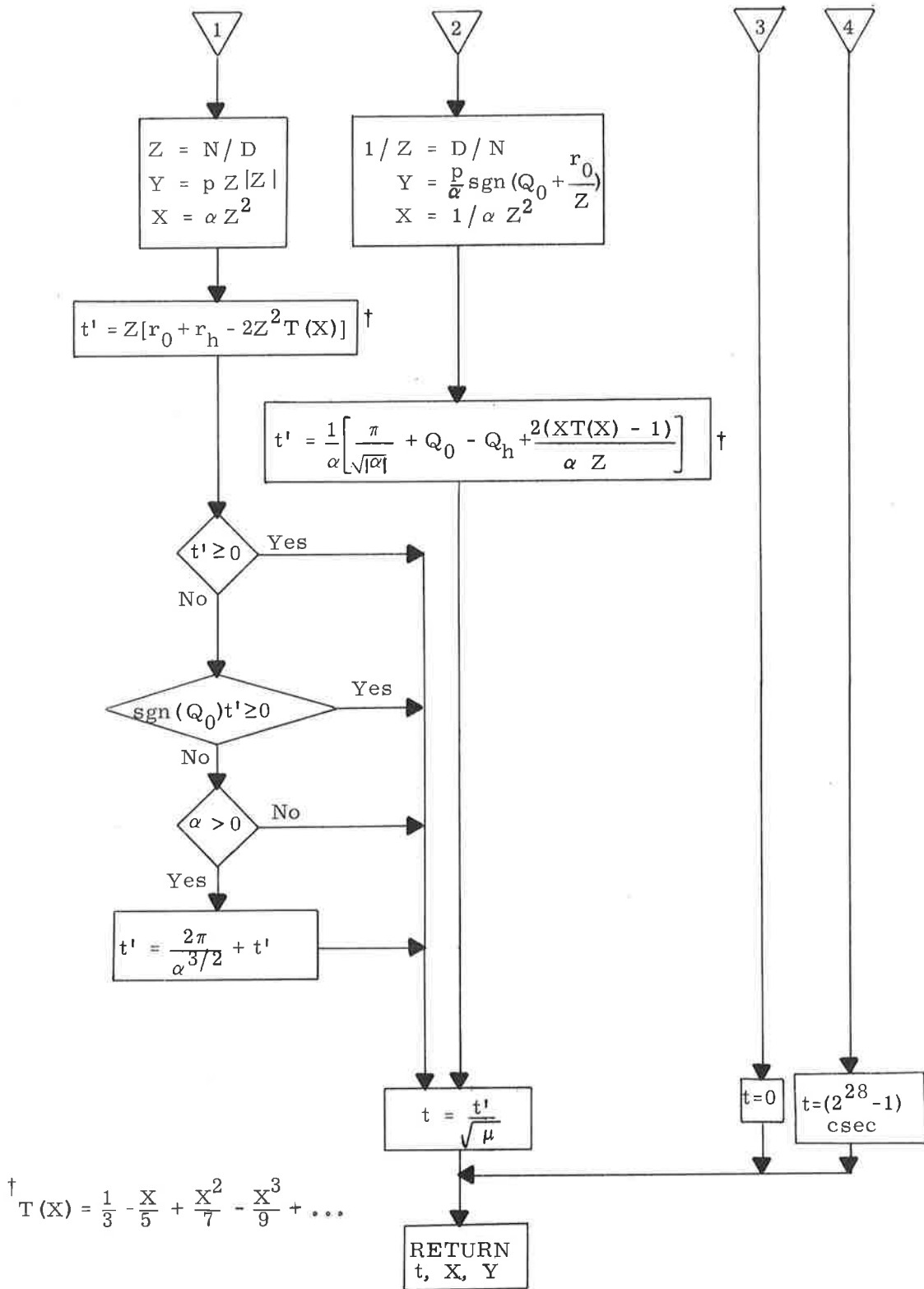


Figure 6.10-7. CALCTFF/CALCTPER Subroutine (Sheet 2 of 2)

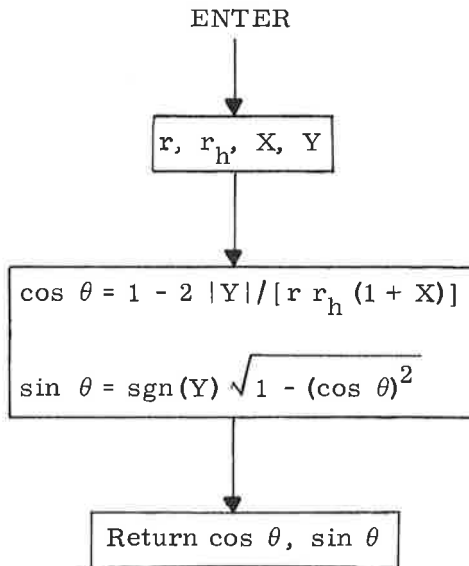


Figure 6.10-8. TFF/TRIG Subroutine

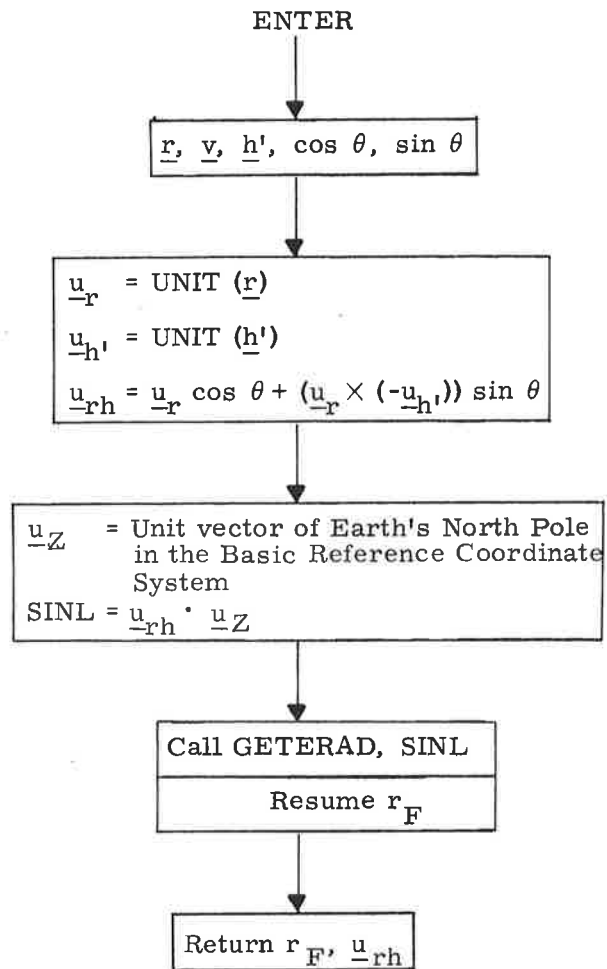


Figure 6.10-9. FISHCALC Subroutine

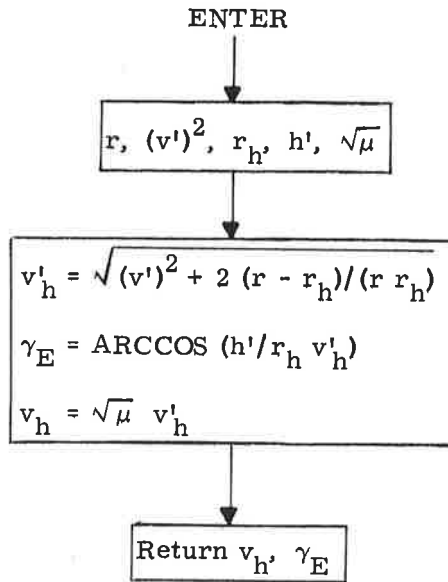


Figure 6.10-10. VGAMCALC Subroutine

Two intermediate quantities, given in two useful equivalent forms, are:

$$Q_0 = \text{sgn}(\dot{r}_0) \sqrt{r_0 (2 - \alpha r_0) - p} = \underline{r}_0 \cdot \underline{v}_0 / \sqrt{\mu} \quad (6.10.2)$$

$$Q_h = \text{sgn}(\dot{r}_h) \sqrt{r_h (2 - \alpha r_h) - p} = \underline{r}_h \cdot \underline{v}_h / \sqrt{\mu}$$

It is these quantities that exclude highly circular orbits, since, being proportional to radial velocity, they become uselessly small. For TFF application to reentry, \underline{r}_h is on the returning side, so the sign of \dot{r}_h is chosen as negative. Excepting ellipses for which $|\Delta E| \geq 90$ deg, the conic time of flight is given by

$$t = (Z/\sqrt{\mu}) (r_0 + r_h - 2 Z^2 T(X)) \quad (6.10.3)$$

where Z is the obtained by one of the following:

if Q_0 and Q_h have opposite signs:

$$Z = (Q_h - Q_0) / (2 - \alpha (r_0 + r_h)) \quad (6.10.4a)$$

if Q_0 and Q_h have like signs:

$$Z = (r_h - r_0) / (Q_h + Q_0) \quad (6.10.4b)$$

and where if

$$\alpha Z^2 < 1.0$$

then

$$X = \alpha Z^2 \quad (6.10.5a)$$

For those elliptic cases having $|\Delta E| \geq 90$ deg,

then

$$\alpha Z^2 \geq 1.0$$

and

$$X = 1/(\alpha Z^2) \quad (6.10.5b)$$

The conic time of flight is given by

$$t = \left(\frac{1}{\alpha \sqrt{\mu}} \right) \left(\frac{\pi}{\sqrt{\alpha}} - Q_h + Q_0 + \frac{2 (XT(X) - 1)}{\alpha Z} \right) \quad (6.10.6)$$

The function $T(X)$ is defined by the series

$$\begin{aligned} T(X) &= \frac{1}{3} - \frac{X}{5} + \frac{X^2}{7} - \frac{X^3}{9} + \dots \\ &= \frac{\sqrt{X} - \arctan(\sqrt{X})}{X \sqrt{X}} \quad \text{if } 0 \leq X \leq 1 \\ &= -\frac{\sqrt{-X} - \operatorname{arctanh}(\sqrt{-X})}{X \sqrt{-X}} \quad \text{if } X < 0 \end{aligned} \quad (6.10.7)$$

The Equations (6.10.4) are equivalent, except that each has a point of indeterminacy. The selection used in Equation (6.10.4) excludes the indeterminate point of each from the region of application. The geometric significance of opposite signs for the Q 's is that r_0 and r_h lie on opposite sides of the conic axis of symmetry. For like signs, the radii lie on the same side. In application to spacecraft landing, r_h is negative and Equation (4a) applies to outbound trajectories, while Equation (4b) applies to inbound ones.

The choice of Equation (6.10.3) or Equation (6.10.6) depends on whether αZ^2 is less than or greater than one. Since both Equation (6.10.3) and Equation (6.10.6) reduce to the same form when $X = 1.0$, the division made above is arbitrary. Equation (6.10.3) gives both positive and negative values, indicating time to terminal point, and time since terminal point passage. A positive value, time to terminal point, is always provided for the ellipse by replacing negative t by $2\pi/\alpha\sqrt{\alpha\mu} + t$. Equation (6.10.6) yields positive values only.

The time of flight equations are exact and involve approximations only to the extent that the function $T(X)$ is represented by a polynomial. Furthermore, Equation (6.10.3) is continuous as the conic changes from hyperbolic to elliptic.

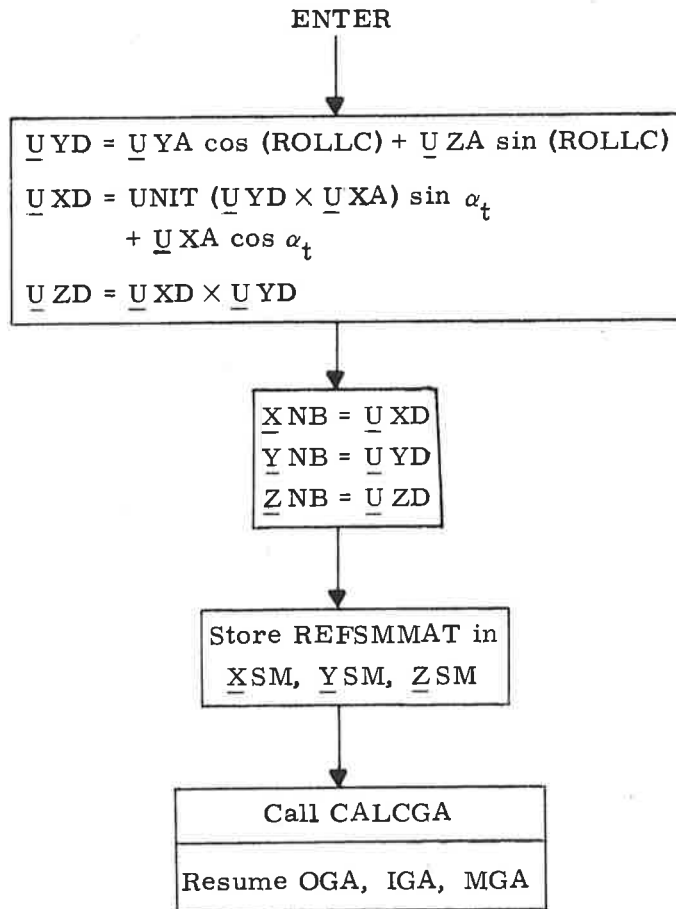
5.6.10.6 Desired Gimbal-angle Display for Entry Attitude Used During P62 and P66

In the CM-SM Separation and Pre-Entry Maneuver Program (P62) and in the Entry Ballistic Program (P66), the desired IMU gimbal angles corresponding to entry attitude (hypersonic trim with respect to the computed wind axis) at the present time are calculated and may be displayed. The calculations are shown in Figure 6.10-11. The following nomenclature is used:

KWE	Equatorial Earth Rate = 1546.70168 ft/sec. (This is a reentry constant; see subsection 5.7).
\underline{u}_Z	Unit vector of earth's North Pole direction, left by pad load.
\underline{U}_{XA}	The trajectory triad computed each two seconds in reference coordinates by CM/POSE (part of the DAP calculations).
\underline{U}_{YA}	
\underline{U}_{ZA}	
\underline{U}_{XD}	Desired body triad for trimmed flight with respect to the relative velocity vector, using roll command and trim angle of attack.
\underline{U}_{YD}	
\underline{U}_{ZD}	
OGA	Outer Gimbal Angle-Roll
IGA	Inner Gimbal Angle-Pitch
MGA	Middle Gimbal Angle-Yaw
ROLLC	Commanded Roll Angle
α_t	Hypersonic Trim Angle of Attack for CM, stored in erasable memory during the prelaunch memory load.

5.6.10.7 Entry Terminal Display of CM Geographic Location

During the Entry Final Phase Program (P67), after the relative wind velocity becomes less than VQUIT (1000 fps), the Entry guidance ceases to compute a new roll command for specifying the CM lift vector direction. Instead, the current value is kept constant and the DAP continues attitude control for the remainder of the flight. The guidance mode SELECTOR is changed to compute



where:

$$\underline{U} XA = - \text{UNIT} (\underline{V} + \text{KWE} (\text{UNIT} (\underline{r}) \times \underline{u}_Z))$$

$$\underline{U} YA = - \text{UNIT} (\underline{U} XA \times \text{UNIT} (\underline{r}))$$

$$\underline{U} ZA = (\underline{U} XA \times \underline{U} YA)$$

Figure 6.10-11. Calculation of Gimbal Angles Corresponding to Entry Attitude

the CM geographic coordinates for display to the crew. The computation is as follows:

Call Latitude-Longitude Subroutine (paragraph 5.5.3) with the parameters as defined in that paragraph and having the following values:

\underline{r} = Average-G position vector
t = time of position vector
F = 0

Obtain

Lat = present geodetic latitude

Long = present longitude

5.6.11 Paragraph 5.6.11 has been deleted.

5.6.12 CMC Idling Program

The CMC Idling Program (P00) is used to maintain the CMC in a state of readiness for entry into any other program. While the idling program is in operation, the Coasting Integration Routine (paragraph 5.2.2) is used to advance the estimated CSM and WS state vectors. This periodic state vector extrapolation is not necessary from a theoretical point of view, but does have two practical purposes. First, it is advisable to maintain current (or at least nearly current) state-vector estimates in case an emergency situation arises. Second, a significant amount of computation time is transferred from a period of high computer activity (for example, navigation measurement processing, targeting) to a period of low activity.

The state vectors of both vehicles are extrapolated even if the vehicles are attached. A special DSKY verb exists whose use allows the WS state vector to be made equal to the CSM state vector.

In order to use the Coasting Integration Routine efficiently, the maximum value for the integration time step Δt_{\max} is computed as described in paragraph 5.2.2.5. The value of Δt_{\max} is a function of radial distance and varies from step to

step. Let t_C be the time associated with the estimated CSM state vector and t_1 be the current time. The estimated CSM state vector is extrapolated ahead when

$$t_1 > t_C + 4 \Delta t_{\max} \quad (6.12.1)$$

The integration is terminated when Δt_{\max} is more than the integration time-to-go. In this manner no extra and smaller-than-maximum integration time steps are performed, and the periodic integration is accomplished most efficiently.

The estimated WS state vector (if applicable) is then extrapolated to the CSM state vector time.

The error transition matrix W (see paragraph 5.2.2.4) is extrapolated with the estimated WS state vector if RENDWFLG indicates that the W -matrix is valid. RENDWFLG is defined in paragraph 5.2.5.2.

The logic for the periodic state-vector extrapolation is illustrated in Figure 6.12-1. The variables D and V are indicators that control the Coasting Integration Routine. The quantities \underline{x}_C and \underline{x}_W are the estimated CSM and WS state vectors, respectively, and \underline{x} is a temporary state vector used for integration. Refer to paragraph 5.2.2.6 for precise definitions of these items.

As shown in the figure, time synchronization of the two state vectors is achieved and maintained by this program. The purpose of the state vector synchronization is to guarantee correct W -matrix extrapolation during rendezvous navigation.

In order to permit correction of wrong erasable memory parameters, which have caused, or could cause, an invalid and excessively lengthy integration process to begin, there is an emergency special DSKY verb to terminate or inhibit the Coasting Integration Routine. This special verb causes the following to occur:

1. If the Coasting Integration Routine is in operation, it is terminated at the end of the current time step.
2. The current program is terminated.
3. The CMC Idling Program (P00) is activated.
4. The P00 state vector test is bypassed so that no state vector integration occurs until a new program is entered or P00 is reselected.

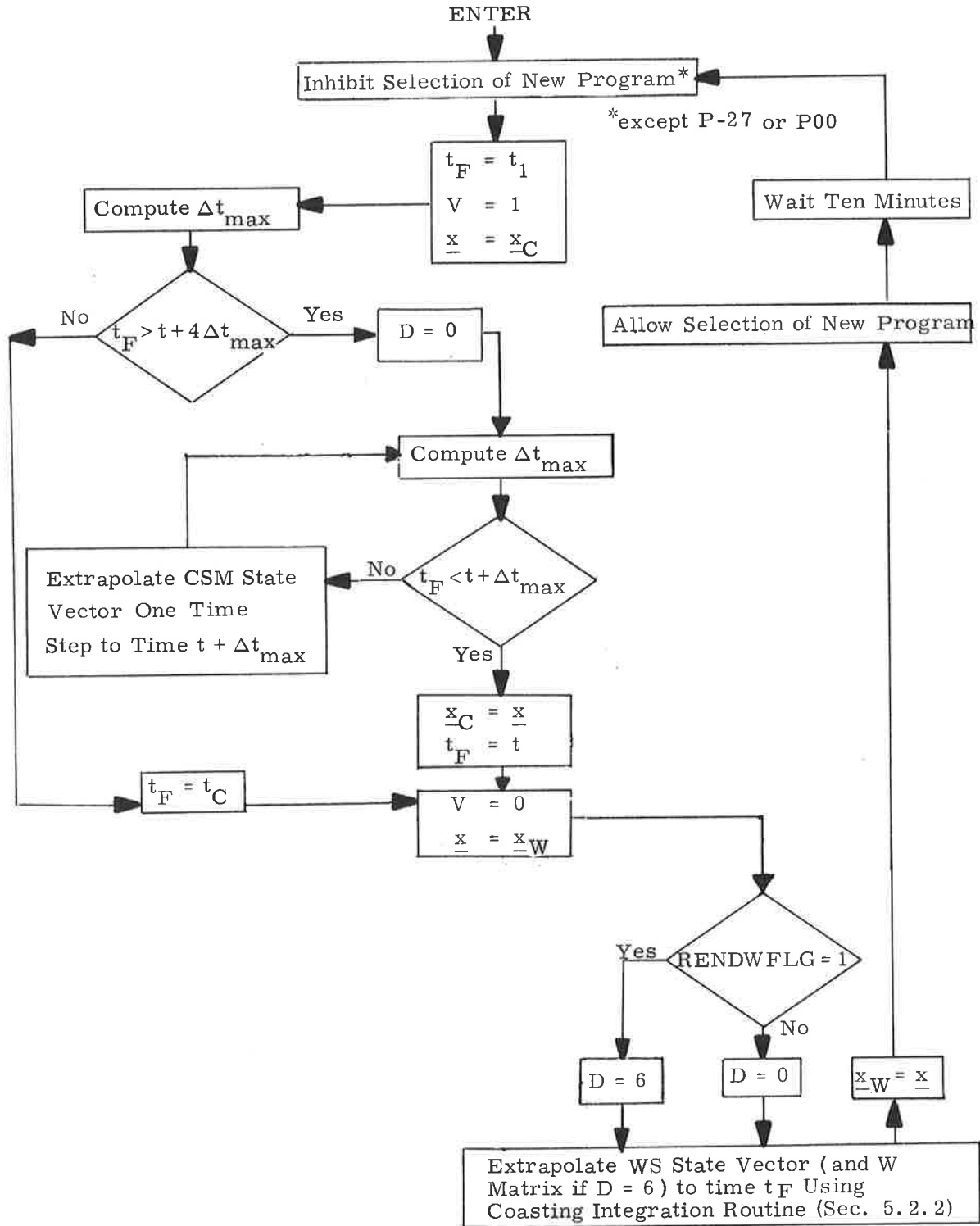


Figure 6.12-1. CMC Idling Program—State Vector Extrapolation

Note that this operation does not maintain state vector synchronization and can, therefore, cause incorrect W-matrix extrapolation in rendezvous navigation.

5.6.13 IMU Compensation

The IMU Compensation is designed to compensate for PIPA bias and scale factor error, and, at the same time, to accumulate gyro torquing commands necessary to compensate for the associated bias and acceleration caused gyro drifts. The correction to the PIPA's is

$$\text{PIPA}_C = (1 + \text{SFE}_I) \text{PIPA}_I - \text{BIAS}_I \Delta t$$

where

PIPA_C is the compensated data for the I^{th} PIPA denoted PIPAX_C , PIPAY_C , PIPAZ_C

$$\text{SFE} = \frac{\text{SF} - \text{SF}_{\text{nom}}}{\text{SF}_{\text{nom}}} \quad (\text{erasable load}^*)$$

$$\text{SF} = \text{Scale-factor} \frac{\text{CM/Sec}}{\text{Pulse}}$$

BIAS_I is the bias for the I^{th} PIPA (an erasable load)

The compensated data is then used to compute the IRIG torquing necessary to cancel the NBD, ADIA, and ADSRA gyro coefficients. The computations are

$$\text{XIRIG} = -\text{ADIAX} \text{PIPAX}_C + \text{ADSRAX} \text{PIPAY}_C - \text{NBDX} \Delta t$$

$$\text{YIRIG} = -\text{ADIAY} \text{PIPAY}_C + \text{ADSRAY} \text{PIPAZ}_C - \text{NBDY} \Delta t$$

$$\text{ZIRIG} = -\text{ADIAZ} \text{PIPAZ}_C - \text{ADSRAZ} \text{PIPAY}_C + \text{NBDZ} \Delta t$$

*The term "erasable load" refers to data entered into CMC erasable memory just before launch.

where

XIRIG, YIRIG, ZIRIG are gyro drift compensation

NBDX, NBDY, NBDZ are gyro bias drifts (an erasable load)

ADSRAX, ADSRAY, ADSRAZ are gyro drifts due to acceleration in spin reference axis (an erasable load)

ADIAX, ADIAY, ADIAZ are gyro drifts due to acceleration in the input axis (an erasable load)

When the magnitude of any IRIG command exceeds two pulses, the commands are sent to the gyros.

During freefall, only NBDX, NBDY, NBDZ are the relevant coefficients; and the routine is so ordered that only these terms are calculated for the gyro compensation.

5.6.14 Impulsive ΔV Program (P77)

The purpose of the Impulsive ΔV Program (P77) is to update the estimated CSM state vector after the execution of a thrusting CSM maneuver not monitored by the GNCS.

The logic for this program is shown in Figure 6.14-1. In the figure, ΔV is the CSM velocity change, expressed in the CSM local vertical coordinate system; and $t_{\Delta V}$ is the time of the maneuver.

5.6.15 Estimated RMS Position and Velocity Error Display

In order to provide the capability for astronaut monitoring of the GNCS estimate of state vector accuracy, a special DSKY verb exists that causes the RSS of the estimated RMS position and velocity errors to be computed from the W-matrix and to be displayed. Based upon the values in this display and the details of the particular mission, the astronaut elects to stop the navigation that is in progress, to resume, or continue with the current navigation procedure, or to reinitialize the W-matrix and continue navigating. The capability of selecting the W-matrix initialization parameters is also included in this process.

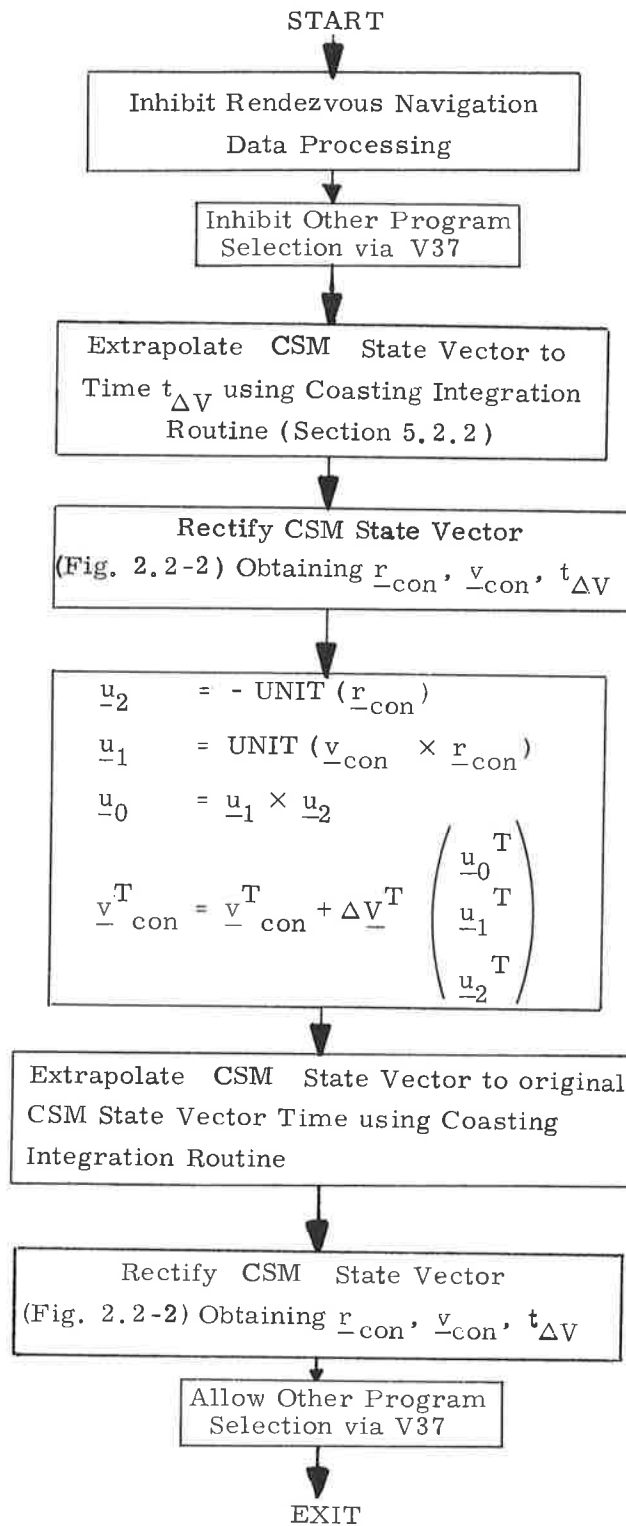
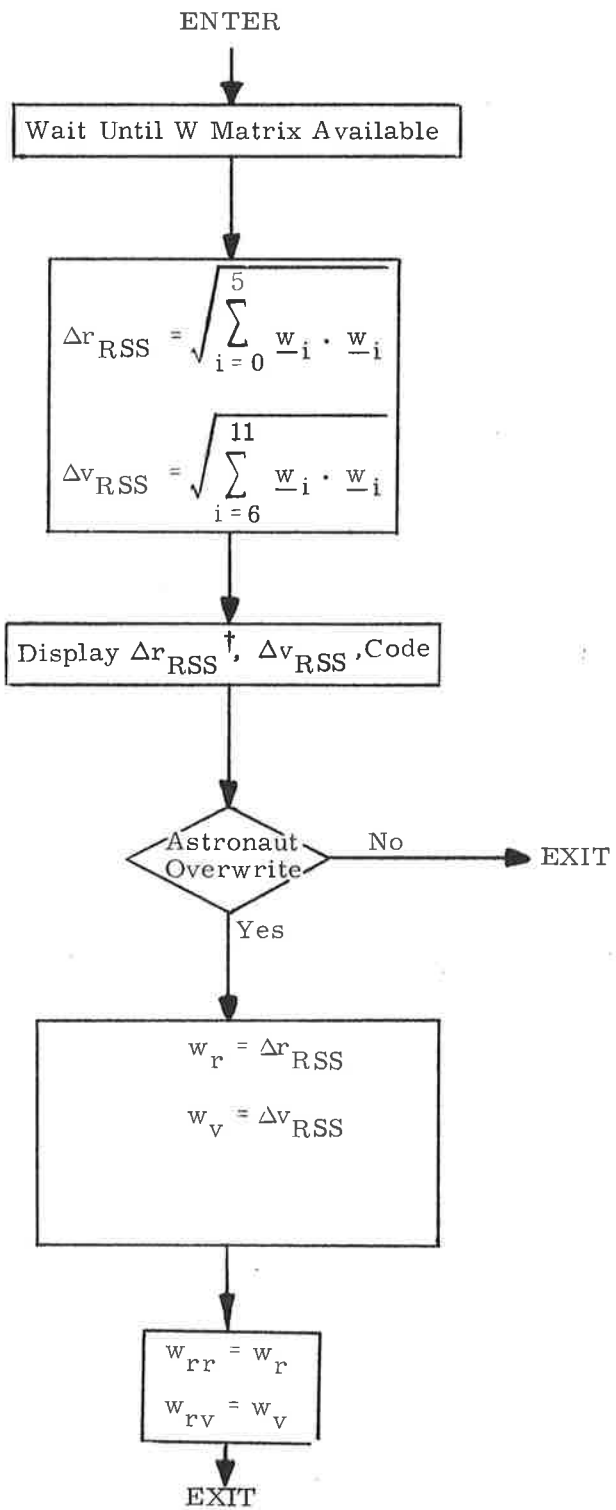


Figure 6.14-1. Impulsive Δv Program (P77)

The logic for the RSS position error (Δr_{RSS}) and RSS velocity error (Δv_{RSS}) display is illustrated in Figure 6.16-1. The vectors \underline{w}_i are partitions of the W-matrix as defined in Equation (2.2.26) of paragraph 5.2.2.4. The variables w_{rr} and w_{rv} are W-matrix initialization parameters, and RENDWFLG is the W-matrix validity flag. See paragraph 5.2.5.2 for further definitions and usage of these terms.

5.6.16 Paragraph 5.6.16 has been deleted.

5.6.17 Paragraph 5.6.17 has been deleted.



†Largest value displayed will be 99997 ft. for $\Delta r_{RSS} > 30479$ meters.

Figure 6.16-1. RSS Position and Velocity Error Display

BLANK

5.7 ENTRY GUIDANCE

The entry guidance is used for control of the CM entry vehicle and is described by the logic flowcharts in Figures 7.0-1 through 7.0-16. A detailed description of the guidance and steering concept is presented in MIT/DL Report R-532, "Reentry Guidance for Apollo", R. Morth, January 1966.

For the Skylab Mission, the entry guidance program retains its high velocity capabilities, as is reflected in the following flowcharts, although only the earth-orbital logic is used. In certain flowcharts, the earth-orbital flow sequence is designated by the heavy lines to aid the reader.

NOTE:— P65 and P66 have not been tested and are not, therefore, considered operational for Skylab.

Figure 7.0-0 illustrates the overall picture of operations during entry. Each block in Figure 7.0-0 is described in detail in following figures. Figure 7.0-1 defines the symbols that represent computed variables stored in erasable memory. Figure 7.0-2 tabulates the variables used in Section 5 with their descriptions, maximum values, and corresponding Section 4 display nouns. (Those parameters that are computed only for display purposes during the entry phase are presented in paragraph 5.6.10.) Figure 7.0-3 gives the values and definitions of constants. The initialization routine shown in Figure 7.0-4 is entered only once at the start of entry. Besides setting the appropriate variables to their initial values, this routine presets the variable SELECTOR to INITROLL.

Every pass through the entry equations (done once every 2 seconds) begins with the powered flight navigation—Average-G Routine—(see Figure 7.0-5), presented in paragraph 5.3.2, which integrates to determine the vehicle's new position and velocity vectors. The navigation routine begins before encountering the entry interface at 400,000 feet altitude, and operates continuously from that time to landing. The targeting section of entry (Figure 7.0-6) then computes the relevant data, shown in the figure, based on the vehicle's position and velocity and the position of the landing site.

The mode selector chooses the next sequence of calculations depending upon the phase of entry trajectory that is currently being flown. The initial roll section (Figure 7.0-7) maintains the roll angle that was previously selected in the initialization routine and determines when to start the next phase. For entry from earth orbit, the CM velocity is less than 27,000 fps. The commanded roll angle is maintained at the initially designated value until the drag exceeds 0.05 g, where a crossrange correction is made. The commanded roll angle again remains constant until the drag exceeds 0.2 g. This effect is obtained as follows. When

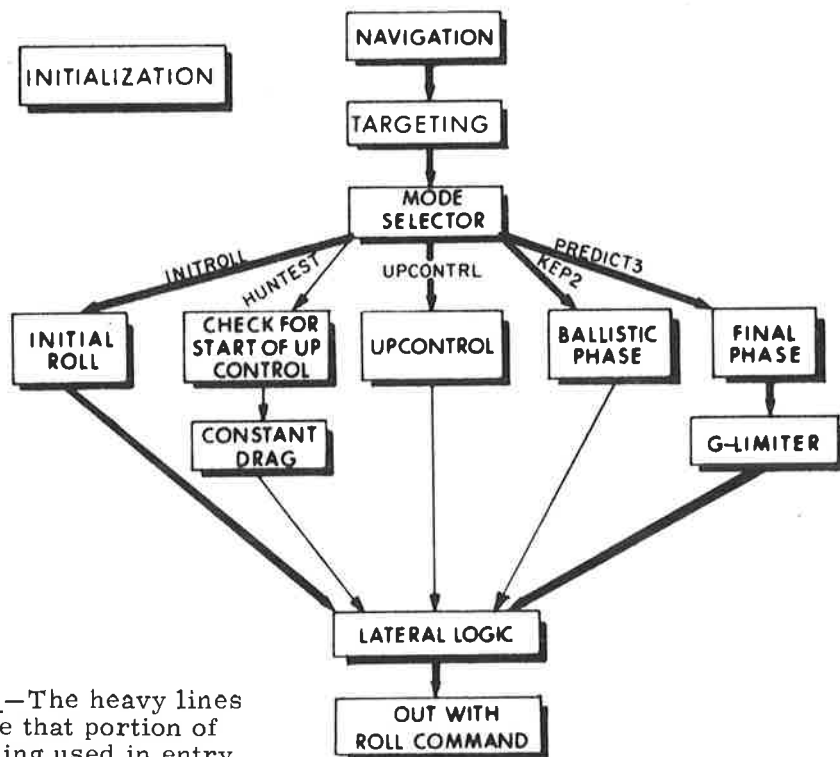
the drag exceeds 0.05 g, INITROLL (Figure 7.0-7) changes the selector to KEP 2 (Figure 7.0-12), but continues on to the lateral logic shown in Figure 7.0-16. Subsequent passes go to KEP 2. When the drag exceeds 0.2 g in Figure 7.0-12, the selector is set to PREDICT 3 of Figure 7.0-13, and the final phase is entered.

For a high velocity entry, the program proceeds from INITROLL to the next phase that maintains a constant drag trajectory while testing to see if it is time to enter the upcontrol phase. The testing is presented in Figures 7.0-8 and 7.0-9. The constant drag equations are given in Figure 7.0-10. The other phases (upcontrol, ballistic, and final) are described in Figures 7.0-11, 7.0-12, and 7.0-13. (See Figure 7.0-0.)

The final phase is accomplished using a stored reference trajectory with linear interpolation between the stored points. The characteristics of this trajectory as well as the steering gains are stored as shown in Figure 7.0-14. The routine that prevents excessive acceleration build-up (G-limiter) is given in Figure 7.0-15. Finally, that portion of entry that performs the lateral logic calculations and computes the commanded roll angle is described in Figure 7.0-16.

For detailed descriptive material about the interaction of Entry Guidance with the Entry DAP and with the mission control programs P61-P67, see Section 3.

PINBALL (DSKY Display) NOTE: In the CMC, range is measured along a great circle and is expressed in revolutions for nouns 50, 63, 64, 66, and 67. (Noun 50, splash error, is discussed in paragraph 5.6.10.2; the other nouns are defined in Figure 7.0-2.) The DSKY display of the range components of these nouns uses the PINBALL scale factor 3441.327214 n. mi./rad. The conventional scaling is ATK of Figure 7.0-3, corresponding to 60 n. mi. is equal to one deg of arc. Since the flowcharts are drawn in terms of nautical miles, the DSKY-displayed range is larger than the flowchart range value by the factor 1.001041507.



NOTE.—The heavy lines indicate that portion of the coding used in entry from earth orbit, i. e., Skylab Mission.

Figure 7.0-0. Entry Computation

<u>VARIABLE</u>	<u>DESCRIPTION</u>	<u>MAXIMUM VALUE *</u>	<u>COMPUTER NAME</u>
URTO	INITIAL TARGET VECTOR	2 (UNIT VECTOR)	= RTINIT
UZ	UNIT VECTOR TRUE NORTH IN BASIC REFERENCE COORDINATE SYSTEM	1	= UNITW
V	VELOCITY VECTOR	2 VSAT	= VEL
R	POSITION VECTOR	2 EXP 29 METERS	= RN
VI	INERTIAL VELOCITY	128 M/CENTISEC	= VN
RTE	VECTOR EAST AT INITIAL TARGET	2	= RTEAST
UTR	NORMAL TO RTE AND UZ	2	= RTNORM
URT	TARGET VECTOR	2	= RT
UNI	UNIT NORMAL TO TRAJECTORY PLANE	2	
DELV	INTEGRATED ACCEL. FROM PIPAS	5.85 16384 CM/S	
G	GRAVITY VECTOR	128 M/CENTISEC	= GDT/2
A0	INITIAL DRAG FOR UPCONTRL	805 FPSS	FPSS=FT/SEC/SEC
AHOOKDV	TERM IN GAMMAL CALC. = AHOOK DVL	8	
A1	DRAG VALUE IN FACTOR CALCULATION	805 FPSS	
ALP	CONST FOR UPCONTRL	1	
ASKEP	KEPLER RANGE	21600 NM	NM = NAUTICAL MILE
ASPI	FINAL PHASE RANGE	21600 NM	
ASPUP	UP-RANGE	21600 NM	
ASP3	GAMMA CORRECTION	21600 NM	
ASPDWN	RANGE DOWN TO PULL-UP	21600 NM	
ASP	PREDICTED RANGE	21600 NM	NOT STORED
COSG	COSINE(GAMMAL)	2	= COSG/2
D	TOTAL ACCELERATION	805 FPSS	
DO	CONTROLLED CONSTANT D	805 FPSS	
DHOOK	TERM IN GAMMAL COMPUTATION	805 FPSS	
DIFF	THETA ATK-ASP (RANGE DIFFERENCE)	21600 NM	
DIFFOLD	PREVIOUS VALUE OF DIFF	21600 NM	
DLEWD	CHANGE IN LEWD	1	
DR	REFERENCE DRAG FOR DOWNCONTROL	805 FPSS	NOT STORED
DREFR()	REFERENCE DRAG (FINAL PHASE TABLE)	805 FPSS	NOT SAVED
DVL	VS1-VL	2 VSAT	
E	ECCENTRICITY	4	NOT STORED
F1()	DRANGE/D DRAG (FINAL PHASE TABLE)	2700/805	= FX +5
F2()	DRANGE/D RDOT (FINAL PHASE TABLE)	2700/2VS NM/FPS	= FX +4 /8
F3()	DRANGE/D (L/D) (FINAL PHASE TABLE)	2700 NM	= FX

* MAXIMUM VALUE DENOTES UNSCALED VARIABLE VALUE WHEN SCALED VARIABLE HAS MAXIMUM VALUE OF ONE.

Figure 7.0-1. Computer Variables (Sheet 1 of 4)

5.7-5

<u>VARIABLE</u>	<u>DESCRIPTION</u>	<u>MAXIMUM VALUE</u>	<u>COMPUTER NAME</u>
FACT1	CONST FOR UPCONTRL	2 VSAT	
FACT2	CONST FOR UPCONTRL	1/805 FPSS	
FACTOR	USED IN UPCONTRL	1	
GAMMAL	FLIGHT PATH ANGLE AT VL	1 RADIAN	
GAMMAL1	SIMPLE FORM OF GAMMAL	1 RADIAN	
HEADSUP	INDICATOR FOR INITIAL ROLL	1	
KA	DRAG TO LIFT UP IF DOWN	805 FPSS	= KAT
KLAT	LATERAL SWITCH GAIN	1	(NOM = .0125)
KZROLL	INDICATOR FOR ROLL SWITCH	1	
LAD	MAX L/D (MIN ACTUAL VEHICLE L/D)	1	
LADPAD	NOMINAL VEHICLE L/D, SP PAD LOAD	1	(NOM = 0.3)
LATANG	LATERAL RANGE	4 RADIANS	
LAT(SPL)	GEODETTIC LATITUDE OF TARGET	1 REVOLUTION	
LNG(SPL)	LONGITUDE OF TARGET	1 REVOLUTION	
LEQ	EXCESS C.F. OVER GRAV=(VSO-1)GS	128.8 FPSS	
LEWD	UPCONTROL REFERENCE L/D	1	
LOD	FINAL PHASE L/D	1	(NOM = 0.18)
LODPAD	FINAL PHASE L/D, SP PAD LOAD	1	
L/D	DESIRED LIFT TO DRAG RATIO (VERTICAL PLANE)	1	
L/D1	TEMP STORAGE FOR L/D IN LATERAL	1	
L/DCMINR	LAD COS(15DEG)	1	(NOM = 0.2895)
PREDANGL	PREDICTED RANGE (FINAL PHASE)	2700 NM	= PREDANG
Q2	FINAL PHASE RANGE	21600 NM	
Q7	MINIMUM DRAG FOR UPCONTROL	805 FPSS	
ROOT	ALTITUDE RATE	2 VSAT	
ROOTREF	REFERENCE RDOT FOR UPCONTROL	2 VSAT	
ROOTREF()	REFERENCE RDOT (FINAL PHASE TABLE)	2 VSAT	= FX +3 /8
RDR	REFERENCE RDOT FOR DOWNCONT	2 VSAT	NOT SAVED
ROLLC	ROLL COMMAND	1 REVOLUTION	
RTOGO()	RANGE TO GO (FINAL PHASE TABLE)	2700 NM	= FX +2
SL	SINE OF LATITUDE	1	NOT SAVED
T	TIME	8 28 CENTISEC	= PIPTIME-TIME/RTO
THETA	DESIRED RANGE (RADIANS)	2 PI RADIANS	= THETAH
V	VELOCITY MAGNITUDE	2 VSAT	
V1	INITIAL VELOCITY FOR UPCONTROL	2 VSAT	
VL	EXIT VELOCITY FOR UPCONTROL	2 VSAT	
VMAGI	INERTIAL VELOCITY MAGNITUDE, VII	128 M/CENTISEC	
VREF	REFERENCE VELOCITY FOR UPCONTROL	2 VSAT	
VREF()	REFERENCE V (FINAL PHASE TABLE)	2 VSAT	NOT STORED
VS1	VSAT OR V1, WHICHEVER IS SMALLER	2 VSAT	
VBARS	$VL / VSAT$	4	
VSO	NORMALISED VEL. SQUARED = $V / VSAT$	4	= VSQUARE
WT	EARTH RATE TIMES TIME	1 REVOLUTION	NOT SAVED
X	INTERMEDIATE VARIABLE IN G-LIMITER	2 VSAT	= WIE (DTEAROT)
Y	LATERAL MISS LIMIT	4 RADIANS	NOT SAVED

Figure 7.0-1. Computer Variables (Sheet 2 of 4)

EXTRA COMPUTER ERASABLE LOCATIONS: NOT SHOWN ON FLOW CHARTS

VARIABLE	DESCRIPTION	MAXIMUM VALUE	
GOTOADDR	ADDRESS SELECTED BY SEQUENCER		
XPIPBUF	BUFFER TO STORE X PIPA COUNTS		
YPIPBUF	BUFFER TO STORE Y PIPA COUNTS		
ZPIPBUF	BUFFER TO STORE Z PIPA COUNTS		
PIPCTR	COUNTS PASSES THRU PIPA READ ROUTINE		
JJ	INDEX IN FINAL PHASE TABLE LOOK-UP		
MM	INDEX IN FINAL PHASE TABLE LOOK-UP		
GRAD	INTERPOLATION FACTOR IN FINAL PHASE		
FX	DRANGE/D L/D = F3	2700 NM	
FX + 1	AREF	805 FPSS	
FX + 2	RTOGO	2700 NM	
FX + 3	RDOTREF	VSAT/4	
FX + 4	DRANGE/D ROOT = F2	21600/2VS NM/FPS	
FX + 5	DRANGE/D DRAG = F1	2700/805 NM/FPSS	
TEM1B	TEMPORARY LOCATION		
TIME/RTO	TIME OF INITIAL TARGET RTINIT	B 28 CENTISEC	
DTEAROT	EST TIME BETWEEN RTINIT AND RT	B 28 CENTISEC	
-			
UNITV	UNIT V VECTOR	2	
-			
UNITR	UNIT R VECTOR	2	
-			
-VREL	NEGATIVE VELOCITY REL TO ATMOSP	2 VSAT	
<u>COMPUTER SWITCHES</u>		<u>INITIAL STATE</u>	<u>CM/FLAGS = STATE +6</u>
GONEPAST	INDICATES OVERSHOOT OF TARGET	BRANCH (1)	95D, BIT 10
RELVELSW	RELATIVE VELOCITY SWITCH	NON-BRANCH (0)	96D, BIT 9
EGSW	FINAL PHASE SWITCH	NON-BRANCH (0)	97D, BIT 8
NOSWITCH	INHIBIT LATERAL SWITCH	NON-BRANCH (0)	98D, BIT 7
HIND	INDICATES ITERATION IN HUNTEST	NON-BRANCH (0)	99D, BIT 6
INRLSW	INDICATES INIT ROLL ATTITUDE SET	NON-BRANCH (0)	100D, BIT 5
LATSW	INHIBIT DOWNLIFT SWITCH IF NOT SET	BRANCH (1)	101D, BIT 4
.05GSW	INDICATES DRAG EXCEEDS .05 GS	NON-BRANCH (0)	102D, BIT 3
GONEBY	INDICATES GONE PAST TARGET (SET)	SELF-INITIALIZING	112D, BIT 8 IN STATE + 7

5.7-6

Figure 7.0-1. Computer Variables (Sheet 3 of 4)

5.7-7

BODY ATTITUDE QUANTITIES (CM/POSE)		
<u>VARIABLE</u>	<u>DESCRIPTION</u>	<u>MAXIMUM VALUE</u>
- -VREL	NEGATIVE VELOCITY REL TO ATMOS.	2 VSAT
- OLDUYA	USED FOR UYA BELOW 1000 FPS	2
- UXA/2	UNIT VECTOR TRIAD	2
- UYA/2	BASED ON	2
- UZA/2	THE TRAJECTORY.	2
- UBX/2	UNIT VECTOR	2
- UBY/2	BODY TRIAD	2
- UBZ/2	FOR CM.	2

Figure 7.0-1. Computer Variables (Sheet 4 of 4)

DISPLAY QUANTITIES*

(FOR USAGE, SEE SECTION 4: P 61 THROUGH P 67)

THIS TABLE PROVIDES THE READER WITH DISPLAY QUANTITY INFORMATION. THE VARIABLE NAMES REFLECT SECTION 5 USAGE. EXCEPTING IMBEDDED BLANKS, THE SECTION 4 NAMES ARE PARENTHESESIZED, IF DIFFERENT. SIGN CONVENTIONS ARE FOUND IN SECTION 4. IF THE DISPLAY VARIABLE NAME DIFFERS FROM SECTION 5 USAGE, SAY BECAUSE OF SCALE FACTOR, THE ALTERNATE SECTION 5 NAME IS ENCLOSED IN ANGULAR BRACKETS: < >. DISPLAY QUANTITY INFORMATION APPEARING ELSEWHERE IN SECTION 5 OTHER THAN IN 5.7 MAY BE FOUND THUS:

N 60 & N 63 (ALL) FIGURE 6.10-4 IN 5.6.10.3
 N 22 (ALL) FIGURE 6.10-11 IN 5.6.10.6
 N 67 (LAT, LONG) SECTION 5.6.10.7

VARIABLE	DESCRIPTION	MAXIMUM VALUE	DISPLAY NOUNS
GMAX	PREDICTED MAXIMUM ENTRY ACCEL	163.84 GS	N 60
VPRD	PREDICTED VELOCITY AT ALTITUDE 400K FT ABOVE FISCHER RADIUS.	128 M/CENTISEC	N 60
GAMMAEI	PREDICTED GAMMA AT ALTITUDE 400K FT ABOVE FISCHER RADIUS.	1 REVOLUTION	N 60
RTGO	RANGE ANGLE TO SPLASH (RTOGO) FROM EMSALT FT ABOVE FISCHER RADIUS. (IN NM)	1 REVOLUTION	N 63
VIO	PREDICTED VELOCITY AT ALTITUDE EMSALT FT ABOVE FISCHER RADIUS.	128 M/CENTISEC	N 63
TTE	TIME OF FREE FALL (TFE) TO ALTITUDE EMSALT FT ABOVE FISCHER RADIUS.	8 28 CENTISEC	N 63
OGA	OUTER GIMBAL ANGLE (OG ROLL)	1 REVOLUTION	N 22
IGA	INNER GIMBAL ANGLE (IG PITCH)	1 REVOLUTION	N 22
MGA	MIDDLE GIMBAL ANGLE (MG YAW)	1 REVOLUTION	N 22
	ALL ABOVE ANGLES FOR HYPERSONIC TRIM		
D	DRAG ACCELERATION (G)	805 FPSS	N 64, N 74
VMAGI	INERTIAL VELOCITY MAGNITUDE (VI)	128 M/CENTISEC	N 64, N 68, N 74
ROLLC	ROLL COMMAND (BETA)	1 REVOLUTION	N 66, N 68, N 69, N 74
XRNGERR	CROSS-RANGE ERROR <LATANG IN NM>	1 REVOLUTION	N 66
DNRNGERR	DOWN RANGE ERROR ** <PREDANG - THETAH IN NM> ** <DNRNGERR DISPLAYS 9999.9 IF GONEPAST =1 IN P 67>	1 REVOLUTION	N 66
HDDT	ALTITUDE RATE <RDOT>	128 M/CENTISEC	N 68
Q7	EXIT DRAG LEVEL (DL)	805 FPSS	N 69
VL	EXIT VELOCITY FOR UP-CONTROL	2 VSAT	N 69
RTOGO	DESIRED RANGE ANGLE <THETAH> <-THETAH IF GONERY =0, THETAH IF GONERY =1>	1 REVOLUTION	N 67, N 64
LAT	GEODETTIC LATITUDE OF VEHICLE	1 REVOLUTION	N 67
LONG	LONGITUDE OF VEHICLE	1 REVOLUTION	N 67
THE FOLLOWING IS A DATA INPUT NOUN (REFER TO SECTION 4, P 61 & P 62)			
HEADSUP	INDICATOR FOR INITIAL ROLL	1 REVOLUTION	N 61
LAT(SPL)	GEODETTIC LATITUDE OF TARGET (IMPACT LAT)	1 REVOLUTION	N 61
LNG(SPL)	LONGITUDE OF TARGET (IMPACT LONG)	1 REVOLUTION	N 61

* See note on page 5.7-2.

5.7-8

Figure 7.0-2. Display Information

CONSTANTS AND GAINS		VALUE	
C1	FACTOR IN ALP COMPUTATION	1.25	
C16	CONSTD GAIN ON DRAG	0.01	
C17	CONSTD GAIN ON ROOT	0.002	
C18	BIAS VEL. FOR FINAL PHASE START	500	FPS
C20	MAX DRAG FOR DOWN-LIFT	210	FPSS
C21	DRAG FOR NO LATERAL SWITCH	140	FPSS
CHOOK	FACTOR IN AHOOK COMPUTATION	0.25	
CHI	FACTOR IN GAMMAL COMPUTATION	1.0	
COS15	COS(15 DEG)	0.965	
DLEWDD	INITIAL VARIATION IN LEWD	-0.05	
DT	COMPUTATION CYCLE TIME INTERVAL	2	SEC.
GMAX	MAXIMUM ACCELERATION	257.6	FPSS (R G-S)
K41	FACTOR IN KA CALC	1.3	GS
K42	FACTOR IN KA CALC	0.2	GS
K43	FACTOR IN DO CALC	90	FPSS
K44	FACTOR IN DO CALC	40	FPSS
KR1	OPTIMIZED UPCONTROL GAIN	2.5	
KR2	OPTIMIZED UPCONTROL GAIN	0.0025	
KDMIN	INCREMENT ON Q7E TO DETECT END OF KEPLER PHASE	0.5	FPSS
KTFTA	TIME OF FLIGHT CONSTANT	1000	
KLAT1	FACTOR IN KLAT CALC	1/24	
K44	GAIN USED IN INITIAL ROLL SECTION	19749550	FPS
LATHIAS	LATERAL SWITCH BIAS TERM	0.41252961	NM
LEWD1	NOMINAL UPCONTROL L/D	0.15	
P0INT1	FACTOR TO REDUCE UPCONTROL GAIN	0.1	
Q3	FINAL PHASE DRANGE/D V	0.07	NM/FPS
Q5	FINAL PHASE DRANGE/ () GAMMA	7050	NM/RAD
C6	FINAL PHASE INITIAL FLIGHT PATH ANGLE	0.0349	RAD
Q7E	MIN DRAG FOR UPCONTROL	6	FPSS
Q7MIN	MIN VALUE FOR Q7 IN FACTOR CALCULATION	805	FPSS (DISABLE FACTOR)
Q19	FACTOR IN GAMMAL1 CALCULATION	0.5	
VFINAL1	VELOCITY TO START FINAL PHASE ON INITIAL ENTRY	27000	FPS
VFINAL	FACTOR IN INITIAL UP-DOWN CALC	26600	FPS
VLMIN	MINIMUM VL	18000	FPS
VMIN	VELOCITY TO SWITCH TO RELATIVE VEL	VSAT/2	
VRCONTRL	ROOT TO START INTO HUNTEST	700	FPS
	VRCONT = COMPUTER NAME		
25NM	TOLERANCE TO STOP RANGE ITERATION	25	NM
VQUIT	VELOCITY TO STOP STEERING	1000	FPS

ATK	ANGLE IN RAD TO NM	3437.7468	NM/RAD
GS	NOMINAL G VALUE FOR SCALING	32.2	FPSS
HS	ATMOSPHERE SCALE HEIGHT	28500	FT
J	GRAVITY HARMONIC COEFFICIENT	0.00162346	
KWE	EQUATORIAL EARTH RATE	1546.70168	FPS
MUE	EARTH GRAVITATIONAL CONSTANT	3.986032233	E 14 CUBIC M/ SEC SEC
RE	EARTH RADIUS	21202900	FT
REQ	EARTH EQUATORIAL RADIUS	20925738.2	FT
VSAT	SATELLITE VELOCITY AT RE	25766.1973	FPS
WIE	EARTH RATE	0.0000729211505	RAD/SEC

Figure 7.0-3. Constants, Gains and Conversion Factors

INITIALIZATION

Use N61 input data to obtain the initial target vector:
Call LATITUDE-LONGITUDE SUBROUTINE (paragraph 5.5.3)
with the parameters as defined there having the following
values:

Lat = LAT(SPL), Long = LNG(SPL), Alt = 0, F = 0
t = TIME/RTO, where TIME/RTO = current state vector
time

Obtain $\bar{U}RTO$

$\bar{U}RTO = \text{UNIT}(\bar{U}RTO)$

$\bar{U}RTO = \text{TARGET VECTOR AT TIME OF INITIALIZATION}$

$\bar{R}TE = \bar{U}Z * \bar{U}RTO$

$\bar{U}TR = \bar{R}TE * \bar{U}Z$

$\bar{U}RT = \text{INITIAL TARGET VECTOR AT NOMINAL TIME OF ARRIVAL}$
(FOR THIS CALCULATION)

$= \bar{U}RTO + \bar{U}TR(\text{COS } WT - 1) + \bar{R}TE \text{ SIN } WT$
WHERE $WT = W \text{ TNOM}$, $\text{TNOM} = 500 \text{ SEC}$

$\bar{U}NI = \text{UNIT}(\bar{V} * \text{UNIT}(\bar{R}))$

LATANG = $\bar{U}RT \cdot \bar{U}NI$

THETA = $\text{ARCCOS}(\bar{U}RT \cdot \text{UNIT}(\bar{R}))$

K2ROLL = $-\text{SIGN}(\text{LATANG})$

Q7 = Q7F

SELECTOR = INITROLL

FACTOR = 1

LOD = LODPAD

LAD = LADPAD

KLAT = KLAT1 LAD

L/DCMINR = LAD COS15

L/D = $- \text{LAD SGN}(\text{HEADSUP})$

DIFFOLD = 0

DLEWD = DLEWDO

LEWD = LEWD1

} *Initialized at start of*
HUNTEST

Q2 = $-1992 + 3500 \text{ LAD}$

INITIALIZE SWITCHES

GONEPAST = 1

EGSW = 0

HIND = 0

LATSW = 1

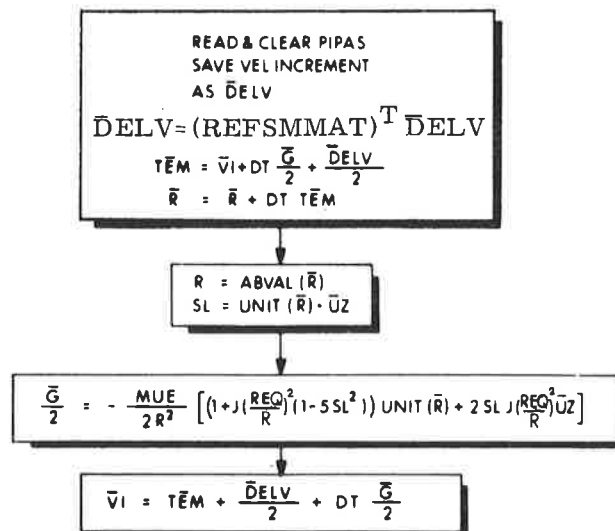
RELVELSW = 0

.05GSW = 0

INRLSW = 0

NGSWITCH = 0

Figure 7.0-4. Entry Program Initialization

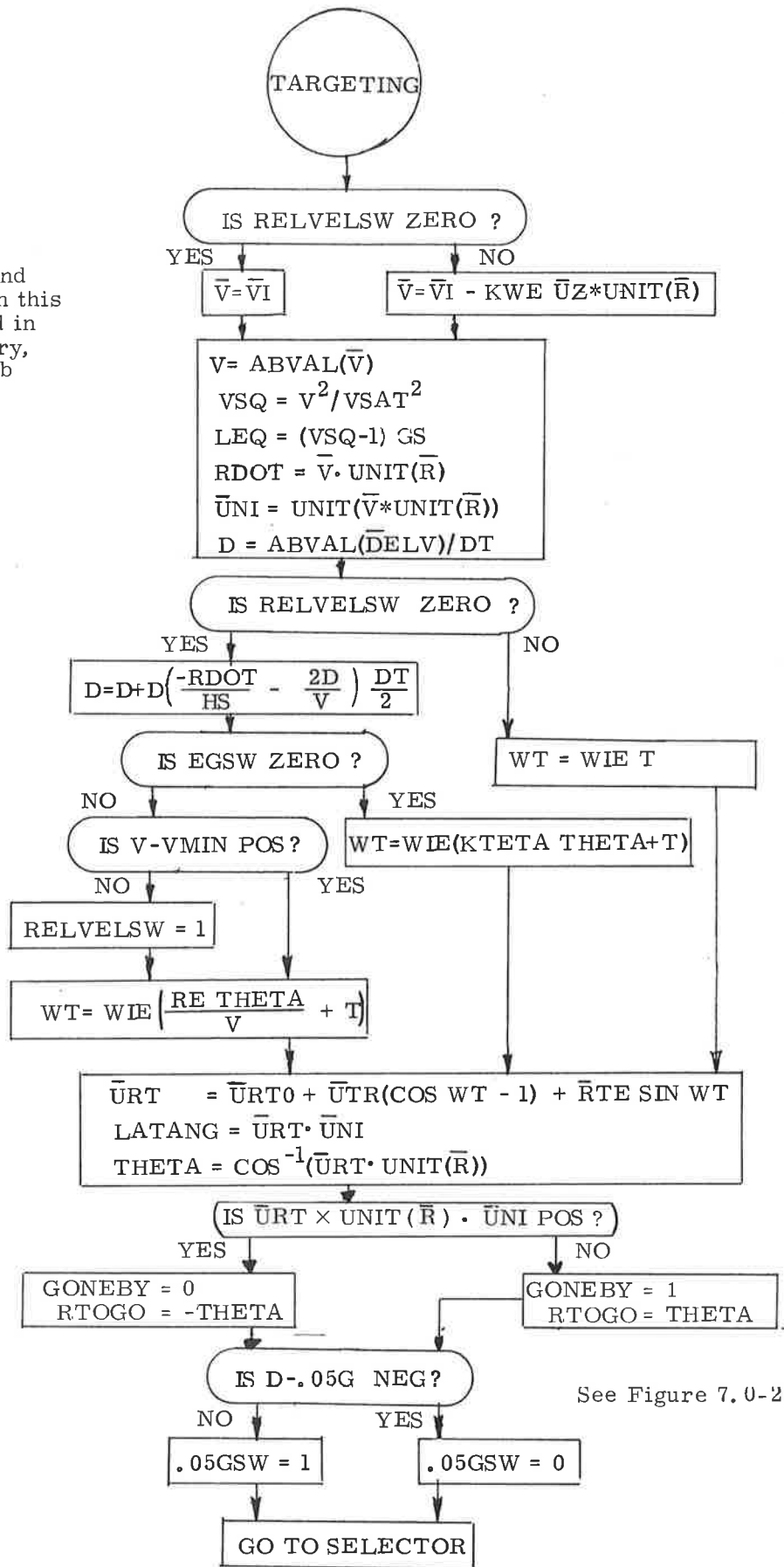


NOTE. — The computations shown in this figure are used in the Skylab Mission.

NOTE. — This routine is identical to Average-G Routine, paragraph 5.3.2.

Figure 7.0-5. Average-G Navigation Routine for Entry from Earth Orbit

NOTE. - The computations and coding shown in this figure are used in earth-orbit entry, i. e., the Skylab Mission.



See Figure 7.0-2

Figure 7.0-6. Entry Targeting

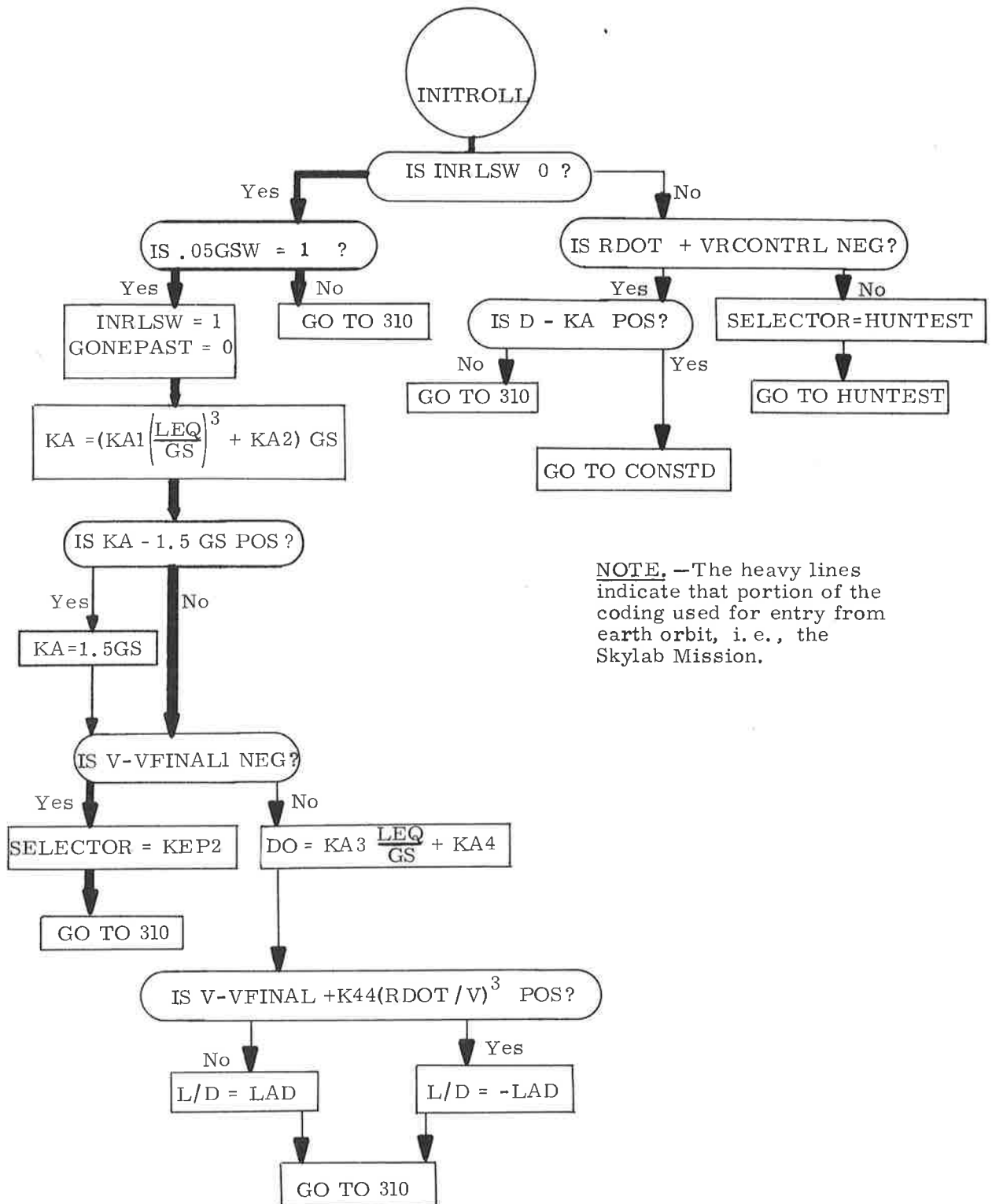
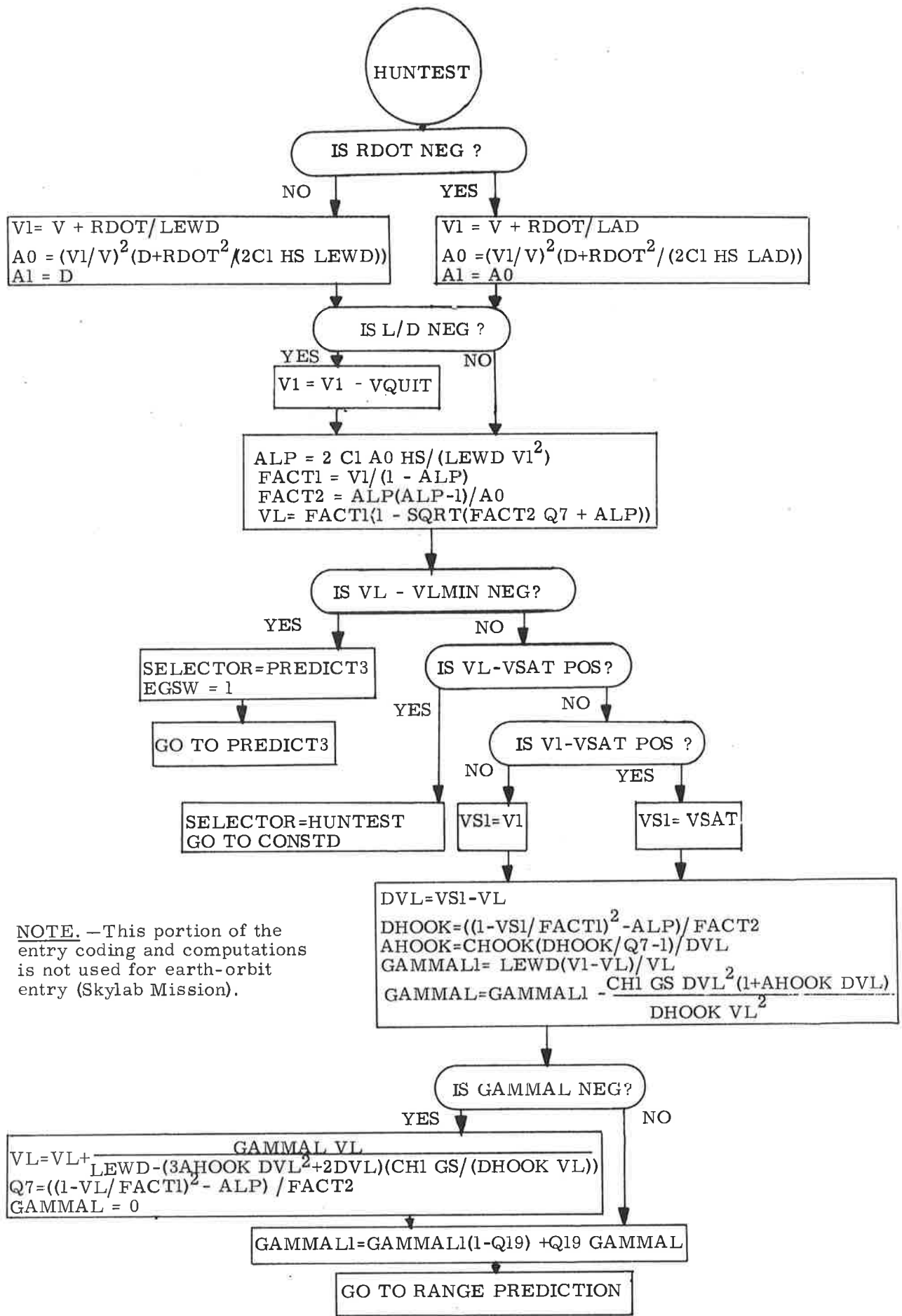
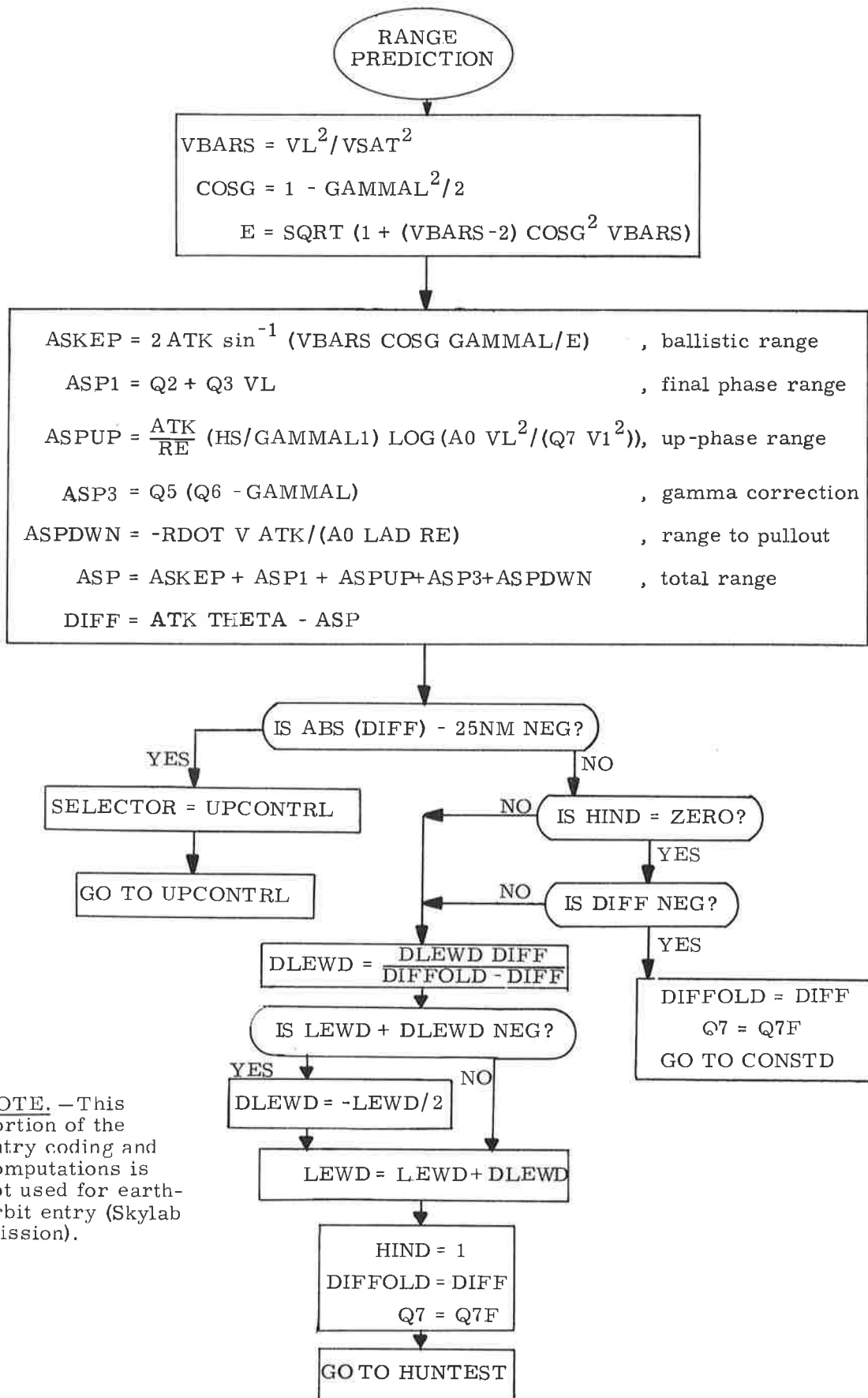


Figure 7.0-7. Initial Roll (INITROLL)



NOTE. — This portion of the entry coding and computations is not used for earth-orbit entry (Skylab Mission).

Figure 7.0-8. Hunttest



NOTE. — This portion of the entry coding and computations is not used for earth-orbit entry (Skylab Mission).

Figure 7.0-9. Entry Range Prediction

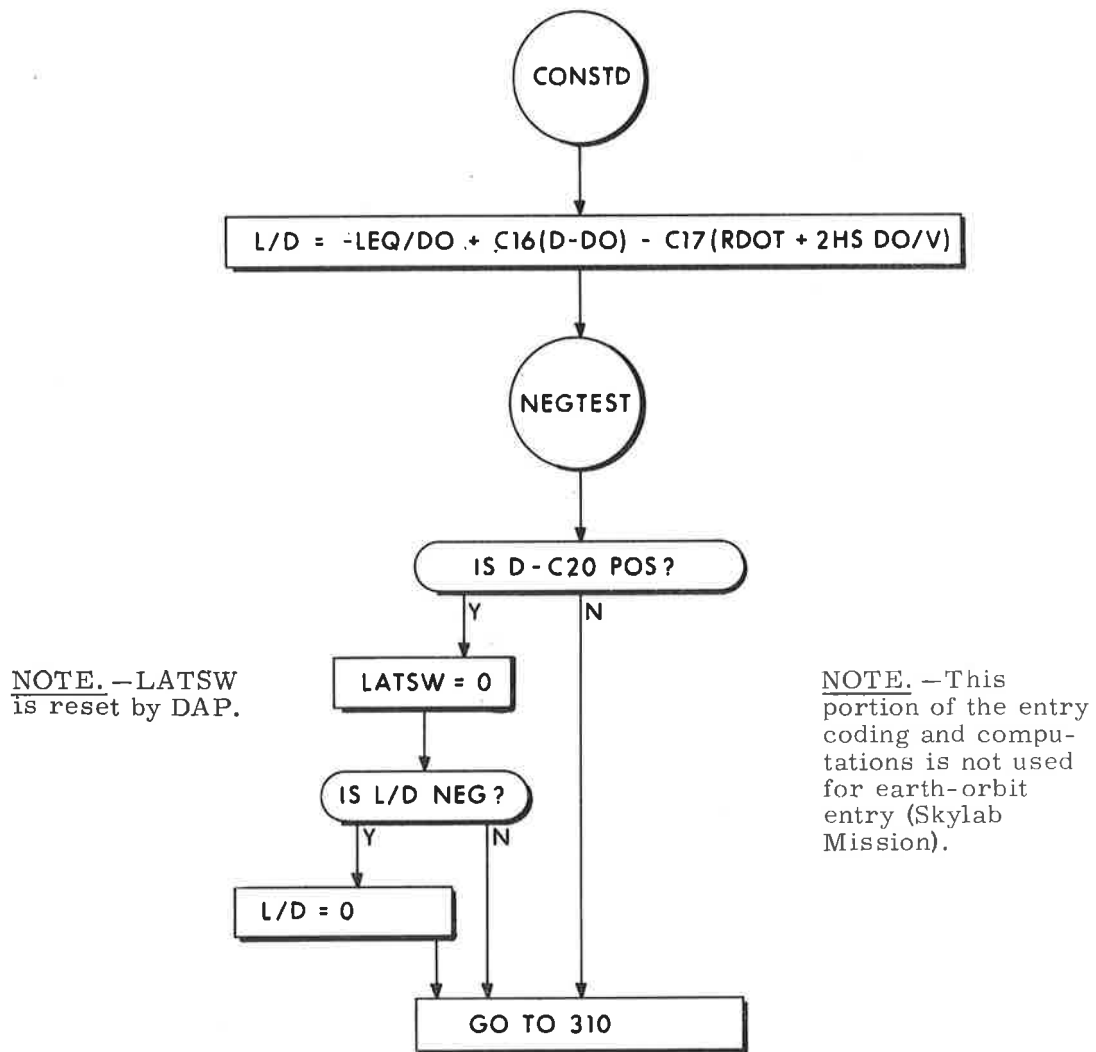


Figure 7.0-10. Constant Drag Control

NOTE. - This portion of entry coding and computations is not used for earth-orbit entry (Skylab Mission).

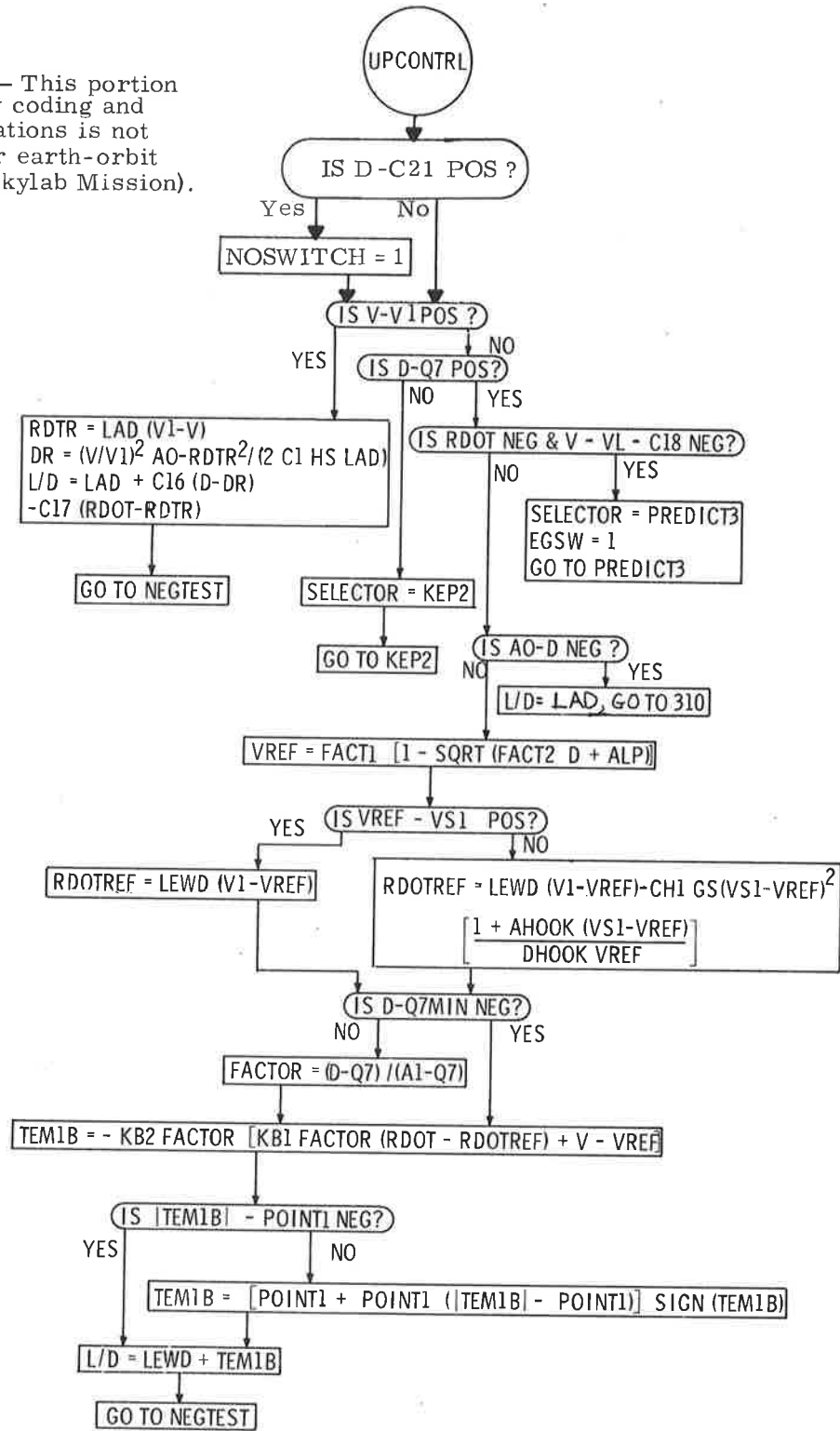
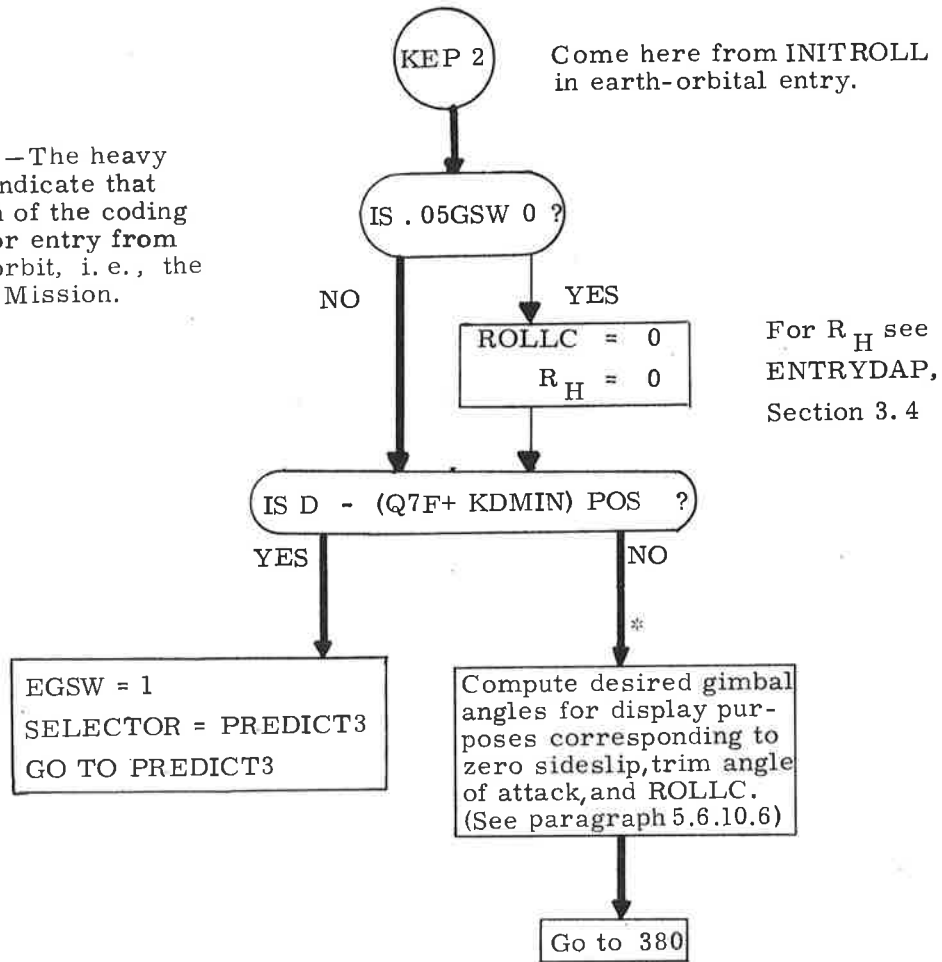


Figure 7.0-11. Entry Upcontrol

NOTE. - The heavy lines indicate that portion of the coding used for entry from earth-orbit, i. e., the Skylab Mission.



* During ballistic mode, DAP maintains attitude control to zero sideslip, trim angle of attack, and ROLLC (Subsection 3.4). During earth-orbital entry, ROLLC is constant and has crossrange correction.

Figure 7.0-12. Entry Ballistic Phase

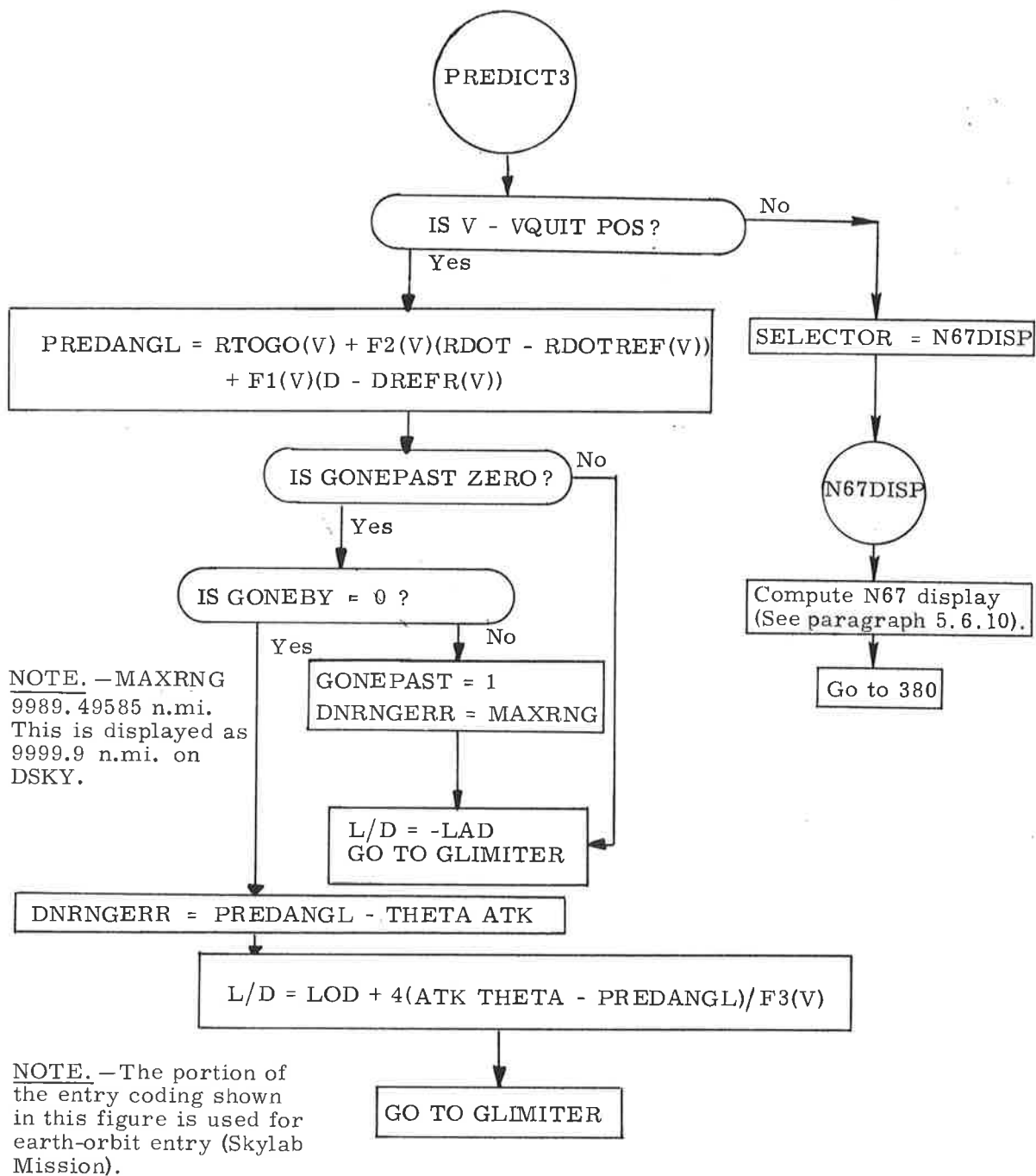


Figure 7.0-13. Entry Final Phase

5.7-20

VREF	RDOTREF	DREFR	DR/DRDOT F2	DR/DA F1	RTOGO	DR/DL/D F3
FPS	FPS	FPSS	NM/FPS	NM/FPSS	NM	NM
994	-690.9	41.15	.002507	-.0346	3.7*	6.10 X2
2103	-719	60.	.003582	-.05551	10.4*	10.91 X2
3922	-694	81.5	.007039	-.09034	23.6*	21.64 X2
6295	-609	93.9	.01446	-.1410	46.3	48.35 X2
8531	-493	98.5	.02479	-.1978	75.4	93.72 X2
10101	-416	102.3	.03391	-.2372	99.9	141.1 X2
14014	-352	118.7	.06139	-.3305	170.9	329.4
15951	-416	125.2	.07683	-.3605x	210.3	465.5
18357	-566	120.4	.09982	-.4956x	266.8	682.7
20829	-781	95.4	.1335	-.6483x	344.3	980.5
23090	-927	28.1	.2175	-2.021	504.8	1385
23500	-820	6.4	.3046	-3.354	643.0	1508
35000	-820	6.4	.3046	-3.354	794.3	1508

*Changed by PCR #803

xRestored by PCR SL 050

Figure 7.0-14. Final Phase Reference (Sheet 1 of 2)

Interpolation Procedure

Start with table entries. See Figure 7.0-14, Sheet 1.

$$VREF(i) \quad , \quad i = 1, 13$$

$$RDOTREF(i), \quad i = 1, 13$$

etc

Vary the index i until velocity magnitude lies between two values in the table. That is:

$$VREF(i) \leq V \leq VREF(i+1).$$

Save the value i and form the linear interpolation factor

$$GRAD = \frac{V - VREF(i)}{VREF(i+1) - VREF(i)}$$

Then form interpolated values. For example

$$RDOTREF(V) = RDOTREF(i) + GRAD [RDOTREF(i+1) - RDOTREF(i)]$$

Figure 7.0-14. Final Phase Reference (Sheet 2 of 2)

NOTE. —The portion of the entry coding shown in this figure is used for earth-orbit entry (Skylab Mission).

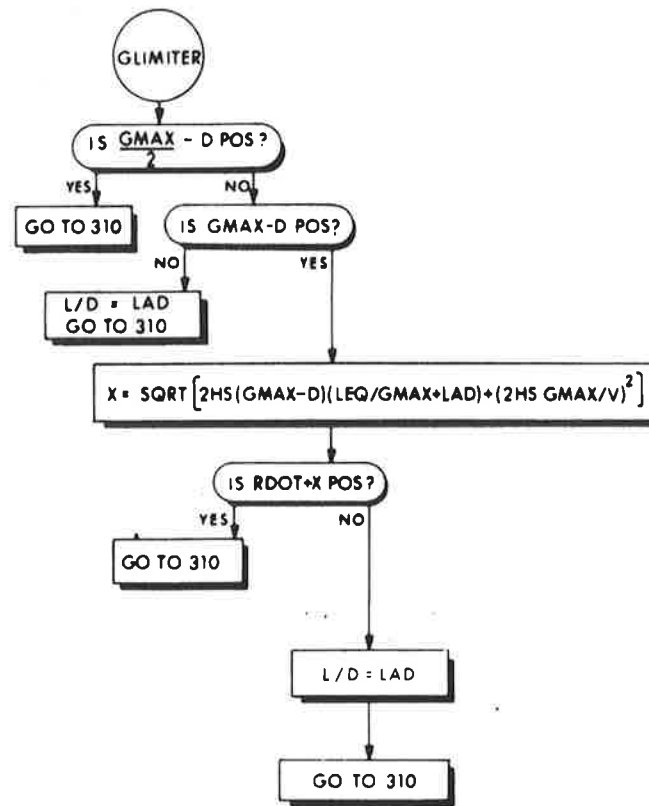


Figure 7.0-15. Entry G-Limiter

NOTE. —The portion of the entry coding shown in this figure is used in earth-orbit entry (Skylab Mission).

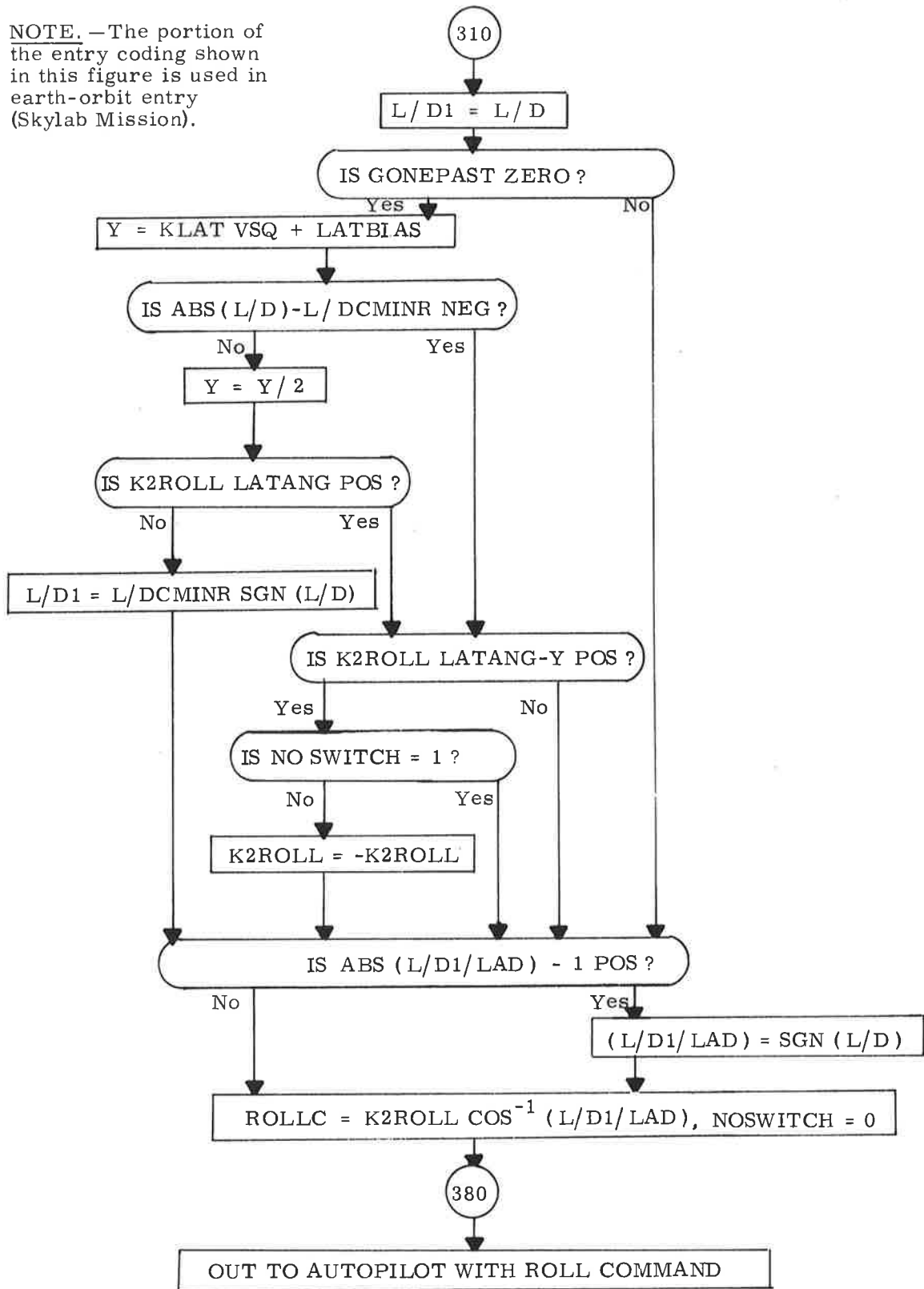


Figure 7.0-16. Entry Lateral Logic

BLANK

5.8 ERASABLE MEMORY PARAMETER LIST

This subsection presents a list of selected CMC erasable parameters required for Skylark program operation. It should be noted, however, that this list is limited in scope. The objective of this selected list is to identify those parameters that can be stored in erasable memory and that are required to initialize the operation of programs and routines. Some parameters can vary continually throughout the mission (for example, vehicle state vectors); others are constant for any one mission phase, but can vary between mission phases; and, finally, some can be constant for one mission, but can change for subsequent missions. The parameters are listed according to the subsection in which they mainly appear. The program mnemonic corresponding to the Section 5 notation is also given.

<u>Subsection</u>	<u>Parameter</u>	<u>Program Mnemonic</u>
5.2	$\underline{r}_C, \underline{r}_W$	RRECTCSM, RRECTLEM
	$\underline{v}_C, \underline{v}_W$	VRECTCSM, VRECTLEM
	$\underline{r}_{C \text{ con}}, \underline{r}_{W \text{ con}}$	RCVCSM, RCVLEM
	$\underline{v}_{C \text{ con}}, \underline{v}_{W \text{ con}}$	VCVCSM, VCVLEM
	$\underline{\delta}_C, \underline{\delta}_W$	DELTA CS M, DELTA ILEM
	$\underline{\nu}_C, \underline{\nu}_W$	NUVCSM, NUVLEM
	$\underline{t}_C, \underline{t}_W$	TETCSM, TETLEM
	$\underline{\tau}_C, \underline{\tau}_W$	TCCSM, TCLEM
	$\underline{x}_C, \underline{x}_W$	XKEPCSM, XKEPLEM
	δr_{MAX}	RMAX
	δv_{MAX}	VMAX
	var_{INT}	INTVAR
	var_R	RVAR
	$\text{var}_{R \text{ min}}$	RVARMIN
	WRD TIME	WRD TIME

<u>Subsection</u>	<u>Parameter</u>	<u>Program Mnemonic</u>	
5.2 (con't)	MINBLKTM	MINBLKTM	
	TBEFCOMP	TBEFCOMP	
	BRNBLKTM	BRNBLKTM	
	MAXWTIME	MAXWTIME	
	FINCMPTM	FINCMPTM	
	w_{rr}	WRENDPOS	
	w_{rv}	WRENDVEL	
	var_{ALT}	ALTVAR	
	5.3	$\Delta v_{(LV)}$	DELVSLV
		t_{IG}	TIG
m		CSMMASS, LEMMASS	
p		PACTOFF	
y		YACTOFF	
F_{16}		EFIMP16	
F_{01}		EFIMP01	
K_1		EIMP1SEC	
Δv_p (SPS)		DVTHRESH	
$\Delta t_{tail-off}$ (SPS)		ETDECAY	
K		EKPRIME	
\dot{m} (SPS)		EMDOT	
A_{ZP}		AZIMUTH	
A_Z		LAUNHAZ	

<u>Subsection</u>	<u>Parameter</u>	<u>Program Mnemonic</u>
5.3 (con't)	K_r	SATRLRT
	t_{E1}	RPSTART
	t_{E2}	POLYSTOP
	a_0, a_1, \dots, a_6	POLYNUM+1
	Lat_P	LATITUDE
	Lon_P	PADLONG
	Alt_I	PGNCSALT
5.4	Δt	TCS
	ϵ_1	EPS1
	ϵ_2	EPS2
	Δh	DHNCC
	Δh_F	DELHI
	N_{H1}	NH1
	E_L	ELEV
5.5	L	LMATRIX
	A_{Z0}	AZO
	t_0	TEPHEM
	LOS_0	LOSO
	C	CMOD
	$PHASE_C$	CARG

<u>Subsection</u>	<u>Parameter</u>	<u>Program Mnemonic</u>
5.6	EMS Altitude	EMSALT
	α_t (See GSOP Section 3.)	ALFAPAD
	SFE ₁ , SFE ₂ , SFE ₃	PIPASCFX, Y, Z
	BIAS ₁ , BIAS ₂ , BIAS ₃	PBIASX, Y, Z
	ADIAX, ADIAY, ADIAZ	
	ADSRAX, ADSRAY, ADSRAZ	
	NBDX, NBDY, NBDZ	
	\bar{y}_{ATM}	NBOA +6D
	\bar{z}_{ATM}	NBOA +12D
	[REFSMMAT]	REFSMMAT
	\bar{CDU}	CDUX, CDUY, CDUZ
	t_d	FIXTIME
	$\bar{\bar{r}}$	RTRIDOT
	\bar{r}	SVEC +4
	LAT _{SPL}	LAT(SPL)
	LONG _{SPL}	LNG(SPL)
	5.7	LADPAD
LODPAD		

5.9 FIXED MEMORY CONSTANTS

Paragraph 5.9.1 contains those fixed memory constants—and their values—discussed in subsections 5.2 through 5.6, but not specified elsewhere in this document. Those constants considered to be control data are so indicated by listing their source references, given in the list of references, paragraph 5.9.2. Those constants needing further elucidation also list a reference to the explanatory material in paragraph 5.9.3. The fixed memory constants listed here are referenced by the subsection in which they are mainly discussed, although they may appear elsewhere in this section of the GSOP.

5.9.1 Fixed Constants

<u>Sub-section</u>	<u>Constant</u>	<u>Unit</u>	<u>Value</u>	<u>Reference (Par. 5.9.2)</u>	<u>Comments (Par. 5.9.3)</u>
5.2	μ_E	m ³ /sec ²	0.3986012×10^{15}	2, 3	
	J_{2E}	—	0.10827×10^{-2}	2, 3	
	J_{3E}	—	-0.256×10^{-5}	2, 3	
	J_{4E}	—	-0.158×10^{-5}	2, 3	
	r_E	m	6,378,160.0	2, 3	
	ϵ_t	csec	3.0		
	var_{IMU}	(mr) ²	0.04	5	
	Δt_C	sec	20		
	MAXRATE	deg/sec	0.1		
	k_{RL}	nm/bit	0.01	6	
	var_{SXT}	(mr) ²	0.04	4	

<u>Sub-section</u>	<u>Constant</u>	<u>Unit</u>	<u>Value</u>	<u>Reference (Par. 5. 9. 2)</u>	<u>Comments (Par. 5. 9. 3)</u>
5. 3	J	—	1.62405×10^{-3}	2, 3	
	F _{SPS}	pounds	20, 500		14
	F _{RCS}	pounds	196. 568 or 393. 136	9	
	F _L	pounds	197	9	10
	V _e (SPS)	m/sec	3151. 0396	20	
5. 4	ωt	deg	130. 00		
	δt	sec	180. 00		
5. 5	(Kepler) ϵ_t	—	2^{-22}	12	
	(Lambert) ϵ_t	—	2^{-19}	12	
	ϵ_x	$m^{1/2}$	2^{-12}	12	
	ϵ_c		2^{-23}	12	
	k_1		2^{-2}	12	
	ω_E	rad/ u. sec	$7.29211514667 \times 10^{-5}$		15
	a	m	6, 378, 166. 0	2, 3	
	b	m	6, 356, 784. 2836	2, 3	
	r _{LP}	m	6, 373, 338	11	
	K ₁		0. 917457730	7	5
	K ₃		0. 397833274	7	5

<u>Sub-section</u>	<u>Constant</u>	<u>Unit</u>	<u>Value</u>	<u>Reference (Par. 5.9.2)</u>	<u>Comments (Par. 5.9.3)</u>
5.5 (cont)	LOS_R	rad/ u. sec	$1.99098665(10)^{-7}$	7	5
	$OMEGA_C$	rad/ u. sec	$1.99096877(10)^{-7}$	7	5
	c	ft/sec	9.835712×10^8	2, 3, 4	
	c_A	arc-sec	20.496	8, 15	
	ϵ_φ	deg	0.01		
5.6	T_{SS}	sec	240		6
	δt	sec	2.4		
	σ_V	ft	30		
	t_b	sec	95		
5.9	Star Table		See Figure 9.1-1.		13

<u>CATALOGUE NO.</u> (octal)		<u>STAR NAME</u>	<u>VISUAL MAGNITUDE</u>	<u>X COORDINATE</u>	<u>Y COORDINATE</u>	<u>Z COORDINATE</u>
1	α	Andromedae (Alpheratz)	2.1	+ .8759127335	+ .0221767921	+ .4819596176
2	β	Ceti (Diphda)	2.2	+ .9344221007	+ .1693349531	- .3133384932
3	γ	Cassiopeiae (Navi)	2.2	+ .4792524242	+ .1143361938	+ .8701978789
4	α	Eridani (Achernar)	0.6	+ .4911095585	+ .2182968779	- .8433017695
5	α	Ursae Minoris (Polaris)	2.1	+ .0150728422	+ .0077535749	+ .9998563354
6	θ	Eridani (Acamar)	3.4	+ .5461199432	+ .5290559424	- .6495019764
7	α	Ceti (Menkar)	2.8	+ .7065135180	+ .7044331667	+ .0679158478
10	α	Persei (Mirfak)	1.9	+ .4142717678	+ .4969671969	+ .7624975459
11	α	Tauri (Aldebaran)	1.1	+ .3552697275	+ .8910553149	+ .2824957457
12	β	Orionis (Rigel)	0.3	+ .2051938364	+ .9681249050	- .1436302809
13	α	Aurigae (Capella)	0.2	+ .1416151994	+ .6807526554	+ .7186939246
14	α	Carinae (Canopus)	-0.9	- .0603437771	+ .6034300291	- .7951294413
15	α	Canis Majoris (Sirius)	-1.6	- .1784151505	+ .9412937204	- .2865905892
16	α_2	Canis Minoris (Procyon)	0.5	- .4076119204	+ .9083859733	+ .0932064683
17	γ	Velorum (Regor)	1.9	- .3601048130	+ .5763392844	- .7335922251
20	ι	Ursae Majoris (Dnoces)	3.1	- .4622023601	+ .4795583503	+ .7459173994
21	α	Hydrae (Alphard)	2.2	- .7717862824	+ .6187078821	- .1467872297
22	α	Leonis (Regulus)	1.3	- .8583271674	+ .4674686420	+ .2115361489
23	β	Leonis (Denebola)	2.2	- .9649190792	+ .0569171730	+ .2563037376
24	γ	Corvi (Gienah)	2.8	- .9533561730	- .0550800729	- .2967797042
25	α	Crucis (Acrux)	1.6	- .4542931560	- .0473466895	- .8895931763
26	α	Virginis (Spica)	1.2	- .9189360613	- .3461098015	- .1891150984
27	η	Ursae Majoris (Alkaid)	1.9	- .5810208598	- .2883158706	+ .7611101885
30	θ	Centauri (Menkent)	2.3	- .6928505073	- .4151341670	- .5895946046
31	α	Bootis (Arcturus)	0.2	- .7878732623	- .5182718149	+ .3326410200
32	α	Coronae Borealis (Alphecca)	2.3	- .5350048409	- .7136778495	+ .4521435028
33	α	Scorpii (Antares)	1.2	- .3561948806	- .8224933890	- .4434296246
34	α	Trianguli Austr. (Atria)	1.9	- .1179551093	- .3394503938	- .9331988117
35	α	Ophiuchi (Rasalhague)	2.1	- .1163388632	- .9689823983	+ .2180329804
36	α	Lyrae (Vega)	0.1	+ .1195024086	- .7708087413	+ .6257579872
37	σ	Sagittarii (Nunki)	2.1	+ .2021911120	- .8729034786	- .4440250795
40	α	Aquilae (Altair)	0.9	+ .4500934054	- .8799661957	+ .1519059600
41	β	Capricorni (Dabih)	3.2	+ .5479677261	- .7957767725	- .2578191213
42	α	Pavonis (Peacock)	2.1	+ .3165704246	- .4450237255	- .8376974691
43	α	Cygni (Deneb)	1.3	+ .4530708165	- .5412631657	+ .7083509163
44	ϵ	Pegasi (Enif)	2.5	+ .8118187290	- .5593567634	+ .1675421212
45	α	Piscis Austr. (Fomalhaut)	1.3	+ .8322543673	- .2429657768	- .4983174684

Figure 9.1-1. Unit Vectors of the Navigational Stars
(FK 4 Catalog at 1973.0 in 1950.0 BRCS)

5.9.2 References for Fixed Constants

1. Project Apollo Coordinate System Standards, SE008-001-1, Office of Manned Space Flight - Apollo Program, National Aeronautics and Space Administration, Washington, D. C., June 1965.
2. Natural Environment and Physical Standards for the Apollo Program and the Apollo Applications Program, NASA-M-DE-8020-008C, SE 015-001-1B, July 1969.
3. Trajectory Prediction Parameters for Skylab, Space Station/Space Shuttle, and Interplanetary Missions, MSC Internal Note 70-FM-121, Revision 1, August 31, 1970.
4. Apollo Missions and Navigation Systems Characteristics, Apollo Navigation Working Group Technical Report No. AN-1.3, December 15, 1967.
5. Airborne Guidance and Navigation Equipment - Block II For Apollo Command Module, Part I, MEI No. 2015000, MIT, June 21, 1965, (C).
6. Command Module Guidance Computer Electrical Interfaces - Block II NAA-MIT, MHO1-01380-216, North American Aviation, Inc.
7. Program Change Request SL408 Revision 1, page 9, Solar Ephemeris Constants (Source: W. M. Robertson).
8. Explanatory Supplement to the Astronomical Ephemeris and the American Ephemeris and Nautical Almanac, 1961.
9. GFE Guidance, Navigation, and Control Performance and Interface Specification Block II, SID 65-299A, North American Aviation, Inc., June 7, 1967.

10. Performance Data Supplement to Mission Modular Data Book-Block II Earth Orbital Mission, SID 66-1501 A, North American Aviation, Inc., Revised March 15, 1967.
11. NASA letter of direction to MIT 67-FS55-91-329.
12. The Fixed Memory Constants in the Conic Subroutines, W. M. Robertson, MIT SGA Memo No. 8-68, March 26, 1968.
13. NAR-MIT Interface Document, MHO1-01301-116 (Rev. B) Block II G & N Navigation Base and Optics Assembly to Command Module Structure.
14. Astronautical Guidance, Richard H. Battin, McGraw Hill Book Co., 1964.
15. The American Ephemeris and Nautical Almanac, Nautical Almanac Office, U.S. Naval Observatory, 1973.
16. Apollo Spacecraft Weight and Mission Performance Definition, Manned Spacecraft Center, NASA, December 12, 1967.
17. This reference has been deleted.
18. This reference has been deleted.
19. NASA Letter of Direction to M. I. T. 68-FS55-19-172.
20. CSM/LM Spacecraft Operational Data Book, Volume I, CSM Data Book, NASA, SNA-8D-027, May, 1968.
21. Vierter Fundamentalkatalog (FK4), Veröffentlichungen des Astronomischen Rechen-Instituts Heidelberg Nr. 10, Verlag G. Braun (1963).
22. FK4 Star Positions in 1950.0 Coordinates at 1973.0 for Use in the AGC Skylark Programs, J. Marshall Reber, MIT/DL Space Guidance Analysis Memo #10-71 (March 1971).

5.9.3 Comments on Fixed Constants

1. The quantity ϵ_t is the limit used for the minimum time step in the Coasting Integration Routine. When the time step drops below ϵ_t , the routine is terminated.

NOTE. —Comments 2-4 have been deleted.

5. The Solar Ephemeris Constants K_1 , K_3 , LOS_R and $OMEGA_C$ were computed according to the method outlined in MIT/DL Space Guidance Analysis Memo 8-71 dated March 16, 1971 and Addendum 1 thereof dated April 20, 1971. The correction factor to convert from ephemeris time rate to universal time rate as explained on page 9 of the memo has been used. The time point t_0 was chosen as July 1.0, 1972. The constants K_1 and K_3 are precisely valid at 1973.0 (January 0.4938182, 1973) which is also the time-point to which the stars have been proper-motioned. The constants LOS_R and $OMEGA_C$ are precisely valid at July 1.0, 1973, which is a convenient mid-point of the overall SKYLAB program. The erasable constants LOS_0 , C , and $PHASE_C$ should be made precisely valid at July 1.0, 1973 also.

6. The quantity T_{SS} is a rough estimate of the time between the initiation of the Star Selection Routine in the IMU Realignment Program (P52) or the Back-up IMU Realignment Program (P54) and the midpoint of the optical sightings on the two celestial bodies.

NOTE. —Comments 7-9 have been deleted.

10. The quantity F_L is ullage thrust using two RCS jets where consideration has been made for the 10 deg offset of these jets.

NOTE. —Comments 11 and 12 have been deleted.

13. The direction of each of the 37 navigation stars in Figure 9-1 is expressed as the components of a unit vector in the Basic Reference Coordinate System. These star directions are the mean places of the stars at the beginning of the Besselian year 1973 and are computed by MIT from the data of Reference 21. See Reference 22 for details.

14. The precise value of thrust is not available. This value is a good working number.

15. Calculated using ΔT (1.0 July 1972) = 42.4 sec.
 ΔT (1973.0) = 42.9 sec.

16. This comment has been deleted.

Appendix A
PCR-PCN Checklist

The preparation of this issue of GSOP Section 5 reflects the implementation of the Skylark Program PCRs and PCNs listed below.

<u>PCR/PCN</u>	<u>Title</u>
003	Improved Short Burn Logic
008	Routine 57 Deletion
009	Routine 33 Deletion
010	Routine 05 Deletion
011	Programs 72-79 Deletion
012 (Note only)	
013	Deletion of Lunar Surface Alignment Option
016	Program 37 Deletion
017	Program 23 Deletion
018	Program 22 Deletion
019	Program 24 Deletion
021	Program 32 Deletion
022	Lock Out Moon Midcourse Perturbations
025	Extended Range Capability
030	Addition of NCC Maneuver Computation Capability
032	VHF Range Rate Computation and Display
033 (See also 043.)	Fixed Coordinate System for Skylab
034 (See also 043.)	Improvements to Ephemerides
036	Compute ATM Star Tracker Gimbal Angles
039 (See also 043.)	DKI Maneuver Sequence (NC1, NH, NSR) Computation Capability
042	Skylab Four Maneuver DKI Sequence
043 (See also 034 and 404.)	Moon Ephemeris Suppression in Skylab
050	Skylab Final Phase Gain Change
400	Program 15 Deletion
402	Eliminate 481 Day Limitation in Lunar-Solar Ephemerides Routine
404 (See also 043.)	Lunar-Solar Ephemerides Routines
405	Transform Optics Angles to Tracking Angles
408 (Rev 1)	Astrodynamic Fixed Memory Constants
409*	Modification to PCR SL022

* PCN

<u>PCR/PCN</u>	<u>Title</u>
410*	Delete Lunar Capability
411*	Delete HAM Targeting Program
412*	Delete Ecsteer
413	ATM Orientation Determination Program (P50)
415	Docked Alignment Capability in P52
416	Add Gyro Trim to R50
417*	Deletion of 9-Dimensional Capability
420	Display Separation Angle in N05
421	Allow Astronaut Overwrite of N59 (Δv_{LOS}) in P36
422*	Initialize Rendezvous Navigation to Update the CSM State Vector
424	Improve Minkey Gyro Torquing Logic
428	Modification of Skylark Memo #14: Preliminary Skylab Rendezvous Targeting Plan for the Skylab GSOP
434	Correct α_{ATM} in P50
436*	Nominal Use of ATM Sources in P52 and P54
438*	Incorrect Star Tracker Angle in P55
439	VHF Range Rate Filter Enable/Disable by Extended Verb
440*	Editorial Update to Skylark Memo No. 11
442*	Modification to R22
445*	Modification No. 3 to Skylark Memo No. 14
448	Modification No. 4 to Skylark Memo No. 14
458 (Rev 1)	Fundamental Astrodynamics Fixed Memory Constants
459	VHF \dot{R} Changes
463*	Iteration Constant Change in Iter Routine

*PCN

BLANK

SKYLARK 1 (GSOP)

R693

Internal Distribution List

Group 23A	<u>D. Lutkevich</u> Brennan Brand Gustafson Higgins Kachmar Klumpp Kriegsman Levine (4)	<u>DL7-211</u> Muller Philliou Phillips Pu Reber Robertson Tempelman	(18)
Group 23B	<u>C. Flynn</u> Klawnsnik Nayar (11)	<u>DL7-221L</u> Reed	(13)
Group 23B	<u>C. Taylor</u> Barnert Beals Brodeur Cramer Goode Hamilton Haslam Lollar	<u>DL7-221L</u> McCoy Neville Ostaneck Rye Smith Whittredge Zeldin	(15)
Group 23B	<u>J. Flaherty</u> Albert Berman Engel	<u>DL7-238A</u> Millard Rosenberg Schulenberg	(6)
Group 23C	<u>M. Erickson</u> Bairnsfather Basile Croopnick Fraser Jones Kalan Maybeck	<u>DL7-215J</u> Penchuk Potter Schlundt Turnbull Womble Work	(13)

Group 23D	<u>S. Beaulieu</u> Drake Dunbar Groome Johnson Kiburz	<u>DL7-213</u> Larson Metzinger Olsson Walsh	(9)
Group 23H	<u>R. Shane</u> Cook Goldberger	<u>DL7-272</u> Kossuth O'Connor	(4)
Group 23N	<u>G. Grover</u> Blanchard Johnson	<u>DL11-201</u> Ogletree Parr	(4)
Group 23P	<u>E. Talbot</u> Greene	<u>DL7-252</u> Stubbs	(2)
Group 23P	<u>B. Hwoschinsky</u> Battin	<u>DL7-203</u> Copps	(2)
Group 23S	<u>P. Amsler</u> Adams Canepa Felleman Heinemann	<u>DL7-140</u> McOuat Werner White Woosley	(8)
Group 23T	<u>J. Grinnel</u> Edmonds Grace Kido Lones	<u>DL7-148B</u> Megna Sheridan St. Amand	(7)
Group 23	<u>D. Farrell</u> Lawrence	<u>DL7-283</u>	(1)
Group 33	<u>J. Hargrove</u> Drane Glick	<u>DL7-111</u> Johnson Mimno	(4)
APOLLO Library			(2)
CSDL Technical Documentation Center			(2)

External Distribution List

SKYLARK 1

Charles Stark Draper Laboratory (5)
P. O. Box 21025
Kennedy Space Center
Attn: Mr. R. O'Donnell

Charles Stark Draper Laboratory (3)
Code EG/MIT Building 16
NASA Manned Spacecraft Center
Houston, Texas 77058
Attn: Mr. G. Silver

NASA MSC HW (10)
Building M7-409
Kennedy Space Center
Florida 32815
Attn: Mr. F. Hughes

Mr. A. Metzger (NASA/RO) (1)

Delco Electronics Division (2)
Milwaukee, Wisconsin 53201
Attn: Mr. J. Stridde, Dept. 42-02
Mr. W. Siarnicki, Dept. 94-02
Building 2-C

Delco Electronics Division (2)
P.O. Box 21027
Kennedy Space Center
Florida 32815
Attn: Mr. J. Kaiser

Delco Electronics Division (1)
P.O. Box 265
Bethpage, Long Island
New York 11714
Attn: Mr. D. Dettmann

Delco Electronics Division (1)
P.O. Box 734
Downey, California 90241
Attn: Mr. D. Karstedt

Kollsman Instrument Corporation (1)
575 Underhill Boulevard
Syosset, Long Island
New York
Attn: Mr. F. McCoy

Raytheon Company Boston Post Road Sudbury, Mass. 01776 Attn: Mr. R. Zagrodnick		(6)
NASA/MSFC	National Aeronautics and Space Administration George C. Marshall Space Flight Center Huntsville, Alabama 35812	(10)
	Attn: J. Mack R-ASTR-5 (1) V. Buckelew S&E-AERO-MFG (1) A. Deaton R-AERO-DG (1) F. Moore R-ASTR-N (1) H. Hosenthien R-ASTR-F (1) D. Germany I-I/IB-E (1) R. Barraza I-V-E (1) W. Chubb R-ASTR/NG (1) J. McCullough I-VE/T (1)	
NASA/MSC	National Aeronautics and Space Administration Manned Spacecraft Center APOLLO Document Control Group (BM86) Houston, Texas 77058 Attn: Mr. J. Garman, FS6 (letter of transmittal only)	(180 + 1R)
BELLCOMM	Bellcomm, Inc. 1100 17th Street N. W. Washington, D. C. 20036 Attn: Information Analysis Section	(6)
LINK	LINK Group, GPSI SIMCOM 1740 A NASA Boulevard Houston, Texas 77058 Attn: Mr. D. Klingbell	(3)
NASA/GSFC	National Aeronautics and Space Administration Goddard Space Flight Center Greenbelt, Maryland 20771 Attn: Mr. P. Pashby, Code 813	(2)
NASA/RASPO NAR	National Aeronautics and Space Administration Resident APOLLO Spacecraft Program Office North American Rockwell, Inc. Space and Information Systems Division CSM Data Management D/096-704 AE99 12214 Lakewood Boulevard Downey, California 90241	(1 + 1R)
NASA/KSC	National Aeronautics and Space Administration Kennedy Space Center Florida 32899 Attn: Technical Document Control Office	(3)
NASA/RASPO GE	General Electric Company APOLLO Systems P. O. Box 2500 Daytona Beach, Florida 32015 Attn: Mr. E. P. Padgett, Jr./Unit 509	(1)

NASA/HQ	NASA Headquarters 600 Independence Avenue SW Washington, D. C. 20546 Attn: Mission Director, Code MA (1) Robert Aller, Code MAO (1)	(2)
NASA/LEWIS	National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135 Attn: Library	(2)
NASA/FRC	National Aeronautics and Space Administration Flight Research Center Edwards AFB, California 93523 Attn: Research Library	(1)
NASA/LRC	National Aeronautics and Space Administration Langley Research Center Langley AFB, Virginia 23365 Attn: Mr. A. T. Mattson	(2)
NAR/KSC	Kennedy Space Center Florida 32815 M. S. O. B. North American Mail Station ZK69 Attn: Mr. D. Matteson	(1)
NASA/RASPO GAC	National Aeronautics and Space Administration Resident APOLLO Spacecraft Program Officer Grumman Aerospace Corporation Bethpage, Long Island New York 11714	(1)
NASA/WSMR	National Aeronautics and Space Administration Post Office Drawer MM Las Cruces, New Mexico 88001 Attn: RH4 Documentation	(2)

BLANK