MSC - 04217
VOL. 3

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

PROJECT/SPACE SHUTTLE

SPACE SHUTTLE GUIDANCE, NAVIGATION AND CONTROL DESIGN EQUATIONS

VOLUME III
ORBITAL OPERATIONS

APRIL 15, 1971

SYSTEMS ANALYSIS BRANCH
GUIDANCE AND CONTROL DIVISION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS

Sat-65279

MSC - 04217

NASA SPACE SHUTTLE PROGRAM WORKING PAPER

SPACE SHUTTLE GUIDANCE, NAVIGATION AND CONTROL DESIGN EQUATIONS

VOLUME III
ORBITAL OPERATIONS

April 15, 1971

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS

Prepared by
Systems Analysis Branch
Guidance and Control Division

K. J. Cox
K. J. Cox, Chief
Systems Analysis Branch

Authorized for Distribution
William C. Bradford for
Maxime A. Faget
Director of Engineering and Development

TABLE OF CONTENTS

PAGE

VOLUME I

1. PURPOSE 1-1
2. SCOPE 2-1
3. APPLICABILITY 3-1
4. DEFINITION OF TERMS 4-1
5. REFERENCE DOCUMENTS 5-1
6. GN&C SYSTEM DESCRIPTION 6-1
7. GN&C SOFTWARE FUNCTIONAL REQUIREMENTS 7-1
8. COORDINATE SYSTEMS AND TRANSFORMATIONS 8-1

VOLUME II - PREFLIGHT THROUGH ORBIT INSERTION

9. DESCRIPTION OF EQUATIONS 9-1
   - 9.1 Preflight 9.1-1
     - 9.1.1 GN&C System Initialization, Monitor, Test and Checkout. 9.1-1
     - 9.1.2 Prelaunch Targeting 9.1-1
     - 9.1.3 Sensor Calibration and Alignment 9.1-1
   - 9.2 Boost Phase Program 9.2-1
     - 9.2.1 Rapid Real-Time State Advancement During Specific Force Sensing 9.2-4
     - 9.2.2 Boost Monitor 9.2-15
   - 9.3 Separation 9.3-1
     - 9.3.1 Attitude Control 9.3-1
   - 9.4 Orbit Insertion Program 9.4-1
     - 9.4.1 Navigation 9.4-1
     - 9.4.2 Powered Ascent Guidance 9.4-3
     - 9.4.3 Attitude Control. 9.4-25

TABLE OF CONTENTS (CONTINUED)

PAGE

9.5 Ascent Aborts 9.5-1

9.5.1 Abort Decision and Mode Determination 9.5-1
9.5.2 Abort Targeting 9.5-1
9.5.3 Abort Guidance 9.5-1
9.5.4 Navigation 9.5-1
9.5.5 Attitude Control 9.5-1

VOLUME III - ORBITAL OPERATIONS

9.6 Orbital Coast 9.6-1

9.6.1 Navigation 9.6-1
9.6.1.1 Conic State Extrapolation 9.6-3
9.6.1.2 Precision State and Filter Weighting Matrix Extrapolation 9.6-33
9.6.2 Sensor Calibration and Alignment 9.6-62
9.6.3 Orbital Coast Attitude Control 9.6.62

9.7 Orbital Powered Flight 9.7-1

9.7.1 Targeting 9.7-2
9.7.2 Navigation 9.7-2
9.7.3 Guidance 9.7-2
9.7.4 Attitude Control 9.7-2

9.8 Rendezvous Mission Phase 9.8-1

9.8.1 Rendezvous Targeting 9.8-3
9.8.2 Relative State Updating 9.8-90
9.8.3 Rendezvous Guidance 9.8-132
9.8.4 Rendezvous Attitude Control 9.8-132

9.9 Station Keeping Mission Phase 9.9-1

9.9.1 Relative State Estimation 9.9-2
9.9.2 Station Keeping Guidance 9.9-2
9.9.3 Station Keeping Attitude Control 9.9-2

PAGE

9.10 Docking and Undocking 9.10-1
9.10.1 Docking and Undocking Navigation 9.10-5
9.10.2 Automatic Docking Control Law 9.10-32
9.11 Docked Operations 9.11-1
APPENDIX A - APPLICABLE APOLLO SOFTWARE A-1

VOLUME IV - DEORBIT AND ATMOSPHERIC OPERATIONS

9.12 Deorbit and Entry 9.12-1
9.12.1 Deorbit Targeting 9.12-1
9.12.2 Entry Navigation 9.12-1
9.12.3 Entry Guidance 9.12-4
9.12.4 Reentry Autopilot 9.12-16
9.13 Transition. 9.13-1
9.13.1 Transition Attitude Control 9.13-2
9.14 Cruise and Ferry Cruise 9.14-1
9.14.1 Navigation 9.14-1
9.14.2 Guidance 9.14-1
9.14.3 Attitude Control 9.14-1
9.15 Approach and Landing 9.15-1
9.15.1 Navigation 9.15-1
9.15.2 Guidance 9.15-1
9.15.3 Control 9.15-1
9.16 Horizontal Takeoff 9.16-1
9.17 Communications and Pointing 9.16-1
9.18 Failure Detection 9.16-1

VOLUME V - FLOW DIAGRAMS

10. FLOW DIAGRAMS . . . . . .

TABLE OF CONTENTS (CONTINUED)

PAGE

VOLUME VI - CONSTANTS AND KEYBOARD ACCESSIBLE PARAMETERS

11. SUMMARY OF CONSTANTS . . . . . .

12. REQUIRED KEYBOARD ACCESSIBLE PARAMETERS . . . . . .

PURPOSE

The purpose of this document is to specify the equations necessary to perform the guidance, navigation and control onboard computation functions for the space shuttle orbiter vehicle. This equations document will: (1) establish more specifically, than on a functional level, the GN&C computational requirements for computer sizing, (2) provide GN&C design equations specification to develop demonstration software for hardware feasibility testing, and (3) define the hardware interface requirements with the GN&C subsystem software. The document will provide a standard of communication of information concerning the GN&C equations, and will provide a means of coordination of GN&C equation development.

SCOPE

This is Volume III of the document which defines the Guidance, Navigation and Control (GN&C) design equations sequencing and interfaces for the computations required in the GN&C Subsystem for all mission phases of the Shuttle Orbiter flight. The equations are intended to satisfy the functional requirements specified in Reference a. This document will describe in mathematical, logical, and operational language all the details necessary to initiate and carry out the design of the required computer modules (subprograms) for the GN&C functions.

The document will be organized into six volumes. Volume I contains Sections 1 through 8, which provide introductory information for the document. Volume II contains the current detailed equations for the preflight, boost, separation, orbit insertion and ascent abort phases of the Orbiter operation. Detailed equations for orbital operations of the Orbiter, which include the orbital coast, orbital powered flight, rendezvous, station keeping, docking and undocking, and docked operations phases, are presented in Volume III. Volume IV contains the current detailed equations for the deorbit and entry, transition, cruise and ferry, approach and landing, and horizontal takeoff phases of the Orbiter. Also, this volume will contain the equations for communications and pointing functions and the failure detection function. For this issue of the document, only Volumes I through IV are being published. In future issues, Volume V will contain the detailed flow diagrams for the equations. For the initial issue, the flow diagrams for the approved equations are included with the equations in Volumes II, III or IV. For future issues, the constants used in the equations will be summarized in Section 11 and the GN&C parameters and variables which can be entered or called via the keyboard will be enumerated in Section 12. These two sections will be contained in Volume VI of the document.

APPLICABILITY

This document is applicable to the Guidance, Navigation, and Control (GN&C) Subsystem of the Electronics System of the Space Shuttle Orbiter Vehicle.. It is applicable to the definition of the shuttle computational requirements for the subsystem listed above. It is applicable to the Phase B and Phase C subsystem development. It defines the Manned Spacecraft Center Guidance and Control Division inhouse sutdy baseline equations design.

9.6 ORBITAL COAST

The following GN&C software functions are envisioned for the orbital coast phase:

Sensor Alignment and Calibration

1. Perform automatic calibration of sensors and compute compensation values during coasting orbital flight.

2. Perform automatic sensor pointing and alignment during coasting orbital flight.

Orbit Navigation

3. Advance inertial vector with conic solutions from an initial state to a final state as a function of time or anomaly.

4 Augment conic state advancement with numerical integration to account for complex gravity potential models.

5. Reduce uncertainties in inertial state by accepting and processing data from navigation sensors (ground beacons, radar altimeter).

Attitude Control

6. Maintain attitude-hold about a desired orientation.

7. Provide attitude rate-hold about a desired rate for orbital rate control, station keeping, passive thermal control or other constant-rate maneuvers.

8. Provide semi-automatic control by initializing attitude hold following manual maneuvers.

9. Implement minimum-impulse jet firings when required by the autopilot or selected by the crew for manual control.

10. Maintain attitude for target visibility at crew and radar locations during the coast periods of rendezvous, station keeping and docking approach.

9.6.1 Orbital Navigation

SPACE SHUTTLE

GN&C SOFTWARE EQUATION SUBMITTAL

Software Equation Section: Conic State Extrapolation Submittal No. 6

Function: Advance inertial state with conic solutions

Module No. ON2 Function No. 1 (MSC 03690)

Submitted by: W. M. Robertson Co. MIT No. 3-71

Date: Feb 1971

NASA Contact: J. Suddath Organization: GCD

Approved by Panel III: K. Cox K.J. Cox Date: 3/10/71

Summary Description: Provides the capability to advance a geocentric inertial state as a function of time or true anomaly. The extrapolation is done analytically assuming Keplerian motion.

Shuttle Configuration: These equations are independent of Shuttle configuration.

Comments:

(Design Status) _____

(Verification Status) _____

Panel Comments:

9.6.1.1 Conic State Extrapolation

1. INTRODUCTION

The Conic State Extrapolation Routine provides the capability to conically extrapolate any spacecraft inertial state vector either backwards or forwards as a function of time or as a function of transfer angle. It is merely the coded form of two versions of the analytic solution of the two-body differential equations of motion of the spacecraft center of mass. Because of its relatively fast computation speed and moderate accuracy, it serves as a preliminary navigation tool and as a method of obtaining quick solutions for targeting and guidance functions. More accurate (but slower) results are provided by the Precision State Extrapolation Routine.

NOMENCLATURE

a	Semi-major axis of conic
c_1	First conic parameter $(\underline{r}_0 \cdot \underline{v}_0)/\mu_E$
c_2	Second conic parameter $(r_0 v_0^2/\mu_E - 1)$
c_3	Third conic parameter $(r_0 v_0^2/\mu_E)$
$C(\xi)$	Power series in ξ defined in text
E	Eccentric anomaly
f	True anomaly
H	Hyperbolic analog of eccentric anomaly
i	Counter
p	Semilatus rectum of conic
p_N	Normalized semilatus rectum (p/r_0)
P	Period of conic orbit
r_0	Magnitude of $\underline{r}_0$
$\underline{r}_0$	Inertial position vector corresponding to initial time t_0
r	Magnitude of $\underline{r}(t)$
$\underline{r}(t)$	Inertial position vector corresponding to time t
s	Switch used in Secant Iterator to determine whether secant method or offsetting will be performed

$S_i(\xi)$	Power series in ξ defined in text
t	Final time (end of time interval through which an extrapolation is made)
t_0	Initial time (beginning of time interval through which an extrapolation is to be made)
$(t - t_0)$	Specified transfer time interval
$(t - t_0)_c$	Value of the transfer time interval calculated in the Universal Kepler Equation as a function of x and the conic parameters
$(t - t_0)'_c$	Previous value of $(t - t_0)_c$
$(t - t_0)_c^{(i)}$	The "i-th" value of the transfer time interval calculated in the Universal Kepler Equation as a function of the "i-th" value x_i of x and the conic parameters
t_{ERR}	Difference between specified time interval and that calculated by Universal Kepler Equation
v_0	Magnitude of $\underline{v}_0$
$\underline{v}_0$	Inertial velocity vector corresponding to initial time t_0
$\underline{v}(t)$	Inertial velocity vector corresponding to time t
x	Universal eccentric anomaly difference (independent variable in Kepler iteration scheme.)
x'	Previous value of x
x_c	Value of x to which the Kepler iteration scheme converged
x'_c	Previous value of x_c

x_i	The "i-th" value of x
x_{min}	Lower bound on x
x_{max}	Upper bound on x
α_0	Reciprocal of semi-major axis at initial point $\underline{r}_0$
α_N	Normalized semi-major axis reciprocal (αr_0)
γ_0	Angle from $\underline{r}_0$ to $\underline{v}_0$
Δt_{max}	Maximum time interval which can be used in computer due to scaling limitations
Δx	Increment in x
ϵ_t	Relative convergence tolerance factor on transfer time interval
ϵ_x	Convergence tolerance on independent variable x
θ	Transfer angle (true anomaly increment)
μ_E	Gravitational parameter of the earth
ξ	Product of α_0 and square of x
$\chi_0, \chi_1, \chi_2, \chi_3$	Coefficients of power series inversion of Universal Kepler Equation
$\underline{1}_{r_0}$	Unit vector in direction of $\underline{r}_0$
$\underline{1}_{v_0}$	Unit vector in direction of $\underline{v}_0$

2. FUNCTIONAL FLOW DIAGRAM

The Conic State Extrapolation Routine basically consists of two parts – one for extrapolating in time and one for extrapolating in transfer angle. Several portions of the formulation are, however, common to the two parts, and may be arranged as subroutines on a computer.

2.1 Conic State Extrapolation As A Function Of Time (Kepler Routine)

This routine involves a single loop iterative procedure, and hence is organized in three sections: initialization, iteration, and final computations, as shown in Fig. 1. The variable "x" is the independent variable in the iteration procedure. For a given initial state, the variable "x" measures the amount of transfer along the extrapolated trajectory. The transfer time interval and the extrapolated state vector are very conveniently expressed in terms of "x". In the iteration procedure, "x" is adjusted until the transfer time interval calculated from it agrees with the specified transfer time interval (to within a certain tolerance). Then the extrapolated state vector is calculated from this particular value of "x".

2.2 Conic State Extrapolation As A Function Of Transfer Angle (Theta Routine)

This routine makes a direct calculation (i.e. does not have an iteration scheme), as shown in Fig. 2. Again, the extrapolated state vector is calculated from the parameter "x". The value of "x" however, is obtained from a direct computation in terms of the conic parameters and the transfer angle θ. It is not necessary to iterate to determine "x", as was the case in the Kepler Routine.

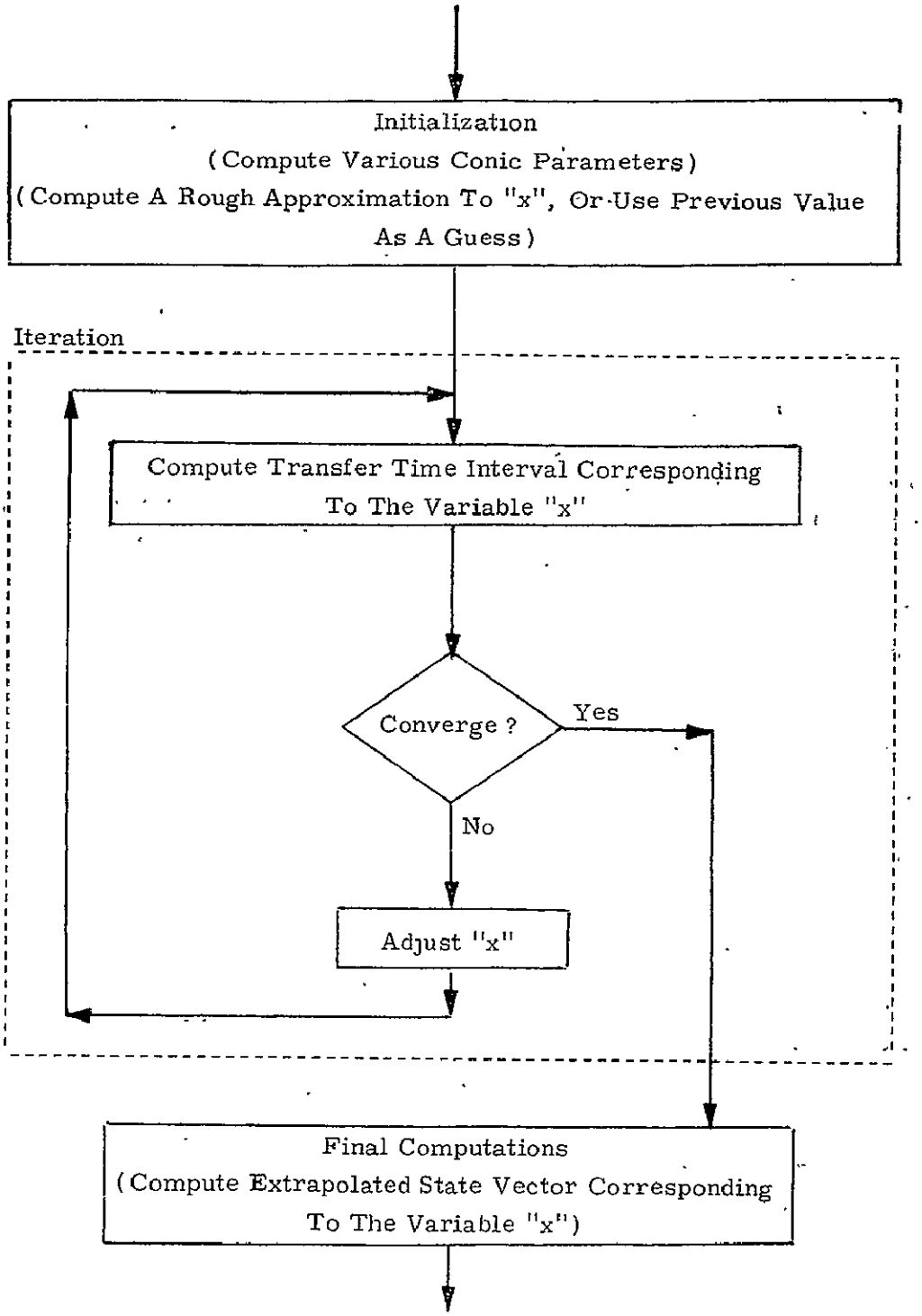


Figure 1 KEPLER ROUTINE FUNCTIONAL FLOW DIAGRAM

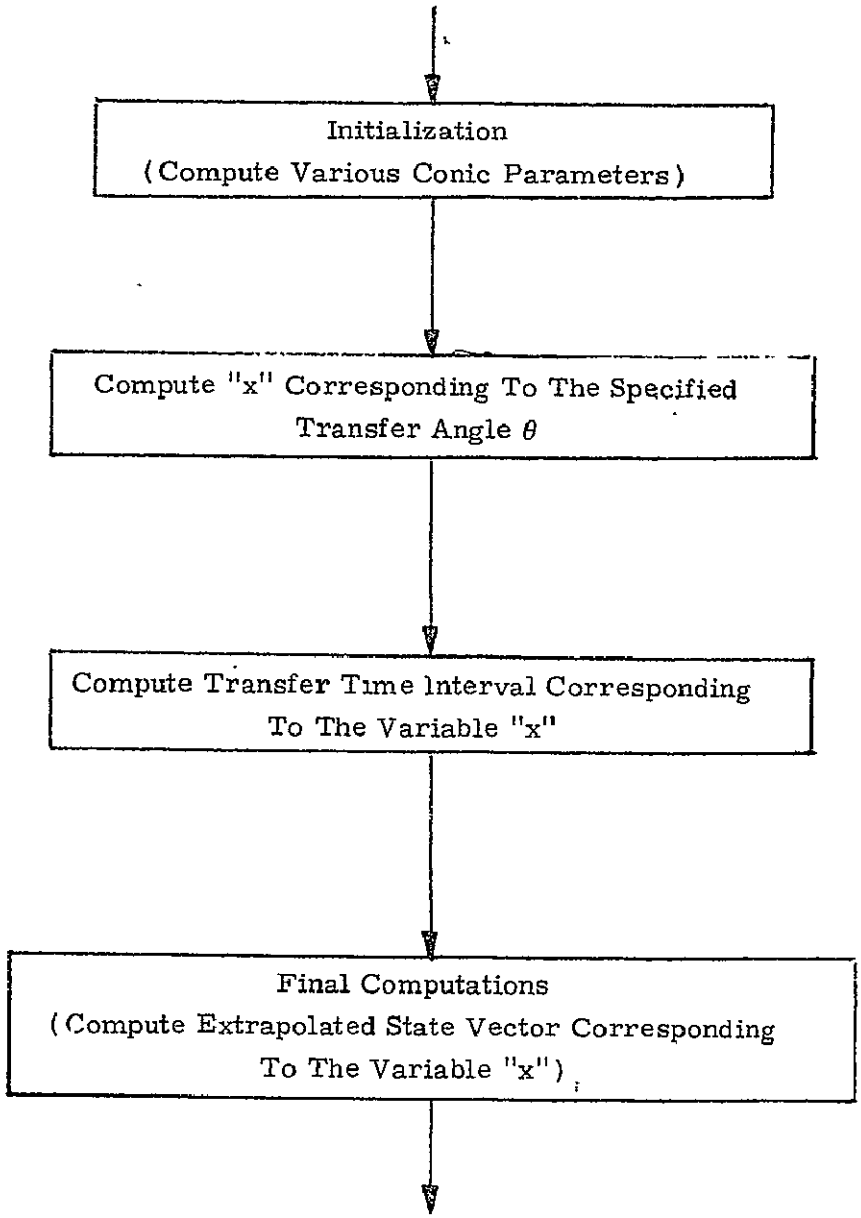


Figure 2 THETA ROUTINE FUNCTIONAL FLOW DIAGRAM

3. ROUTINE INPUT-OUTPUT

The Conic State Extrapolation Routine has only one system parameter input: the gravitational parameter of the earth. Its principal real-time inputs are the inertial state vector which is to be extrapolated and the transfer time interval or transfer angle through which the extrapolation is to be made. Several optional secondary inputs may be supplied in the transfer time case in order to speed the computation. The principal real-time output of both cases is the extrapolated inertial state vector.

3.1 Conic State Extrapolation As A Function of Transfer Time Interval (Kepler Routine)

Input Parameters

System

μ_E — Gravitational parameter of the earth (Product of earth's mass and universal gravitational constant).

Real-Time (Required)

$(\underline{r}_0, \underline{v}_0)$: Inertial state vector which is to be extrapolated (corresponds to time t_0).

$(t - t_0)$: Transfer time interval through which the extrapolation is to be made.

Real-Time (Optional)

x : Guess of independent variable corresponding to solution in Kepler iteration scheme. (Used to speed convergence).

$(t - t_0)'_c$ Value of dependent variable (the transfer time interval) in the Kepler iteration scheme, which was calculated in the last iteration of the previous call to Kepler.

x'_c Value of the independent variable in the Kepler iteration scheme, to which the last iteration of the previous call to Kepler had converged.

Output Parameters

$(\underline{r}(t), \underline{v}(t))$ Extrapolated inertial state vector (corresponds to time t).

$(t - t_0)_c$ Value of the dependent variable (the transfer time interval) in the Kepler iteration scheme, which was calculated in the last iteration (should agree closely with $(t - t_0)$).

x_c Value of the independent variable in the Kepler iteration scheme to which the last iteration converged.

3.2 Conic State Extrapolation As A Function Of Transfer Angle (Theta Routine)

Input Parameters

System

μ_E Gravitational parameter of the earth (Product of earth's mass and universal gravitational constant).

Real-Time

$(\underline{r}_0, \underline{v}_0)$: Inertial state vector which is to be extrapolated.

θ : Transfer angle through which the extrapolation is to be made.

Output Parameters

$(\underline{r}, \underline{v})$: Extrapolated inertial state vector.

$(t - t_0)_c$: Transfer Time Interval corresponding to the conic extrapolation through the transfer angle θ.

4. DESCRIPTION OF EQUATIONS

4.1 Conic State Extrapolation As A Function Of Time (Kepler Routine)

The universal formulation of Stumpff-Herrick-Battin in terms of the universal eccentric anomaly difference is used. This variable, usually denoted by x, is defined by the relations:

$$x = \begin{cases} \sqrt{a}\,(E - E_0) \text{ for ellipse} \\ \sqrt{p}\,(\tan f/2 - \tan f_0/2) \text{ for parabola} \\ \sqrt{-a}\,(H - H_0) \text{ for hyperbola} \end{cases}$$

where a is the semi-major axis, E and H are the eccentric anomaly and its hyperbolic analog, p is the semi-latus rectum and f the true anomaly. The expressions for the transfer time interval $(t - t_0)$ and the extrapolated position and velocity vectors $(\underline{r}, \underline{v})$ in terms of the initial position and velocity vectors $(\underline{r}_0, \underline{v}_0)$ as functions of x are:

(Universal Kepler Equation)

$$(t - t_0) = \frac{1}{\sqrt{\mu_E}}\left[\frac{\underline{r}_0 \cdot \underline{v}_0}{\sqrt{\mu_E}} x^2 C(\alpha_0 x^2) + (1 - r_0 \alpha_0) x^3 S(\alpha_0 x^2) + r_0 x\right]$$

$$\underline{r}(t) = \left[1 - \frac{x^2}{r_0} C(\alpha_0 x^2)\right]\underline{r}_0 + \left[(t - t_0) - \frac{x^3}{\sqrt{\mu_E}} S(\alpha_0 x^2)\right]\underline{v}_0$$

$$\underline{v}(t) = \frac{\sqrt{\mu_E}}{r\, r_0}\left[\alpha_0 x^3 S(\alpha_0 x^2) - x\right]\underline{r}_0 + \left[1 - \frac{x^2}{r} C(\alpha_0 x^2)\right]\underline{v}_0$$

where

$$\alpha_0 = \frac{1}{a_0} = \frac{2}{r_0} - \frac{v_0^2}{\mu_E}$$

and

$$S(\xi) = \frac{1}{3!} - \frac{\xi}{5!} + \frac{\xi^2}{7!} - \ldots$$

$$C(\xi) = \frac{1}{2!} - \frac{\xi}{4!} + \frac{\xi^2}{6!} - \ldots$$

Since the transfer time interval $(t - t_0)$ is given, it is desired to find the x corresponding to it in the Universal Kepler Equation, and then to evaluate the extrapolated state vector $(\underline{r}, \underline{v})$ expression using that value of x. Unfortunately, the Universal Kepler Equation expresses $(t - t_0)$ as a transcendental function of x rather than conversely, and no power series inversion of the equation is known which has good convergence properties for all orbits, so it is necessary to solve the equation iteratively for the variable x.

For this purpose, the secant method (linear inverse interpolation/ extrapolation) is used. It merely finds the increment in the independent variable x which is required in order to adjust the dependent variable $(t - t_0)_c$ to the desired value $(t - t_0)$ based on a linear interpolation/extrapolation of the last two points calculated on the $(t - t_0)_c$ vs x curve. The method uses the formula

$$(x_{n+1} - x_n) = - \frac{(t - t_0)_c^{(n)} - (t - t_0)}{(t - t_0)_c^{(n)} - (t - t_0)_c^{(n-1)}} (x_n - x_{n-1})$$

where $(t - t_0)_c^{(i)}$ denotes the evaluation of the Universal Kepler Equation using the value x_i. In order to prevent the scheme from taking an increment back into regions in which it is known from past iterations that the solution does not lie, it has been found convenient to establish upper and lower bounds on the independent variable x which are continually reset during the course of the iteration as more and more values of x are found to be too large or too small. In addition, it has also been found expedient to damp by 10% any increment in the independent variable which would (if applied) take the value of the independent variable past a bound.

To start the iteration scheme, some initial guess x_0 of the independent variable is required as well as a previous point $(x_{-1}, (t - t_0)_c^{(-1)})$ on the $(t - t_0)_c$ vs x curve. If no previous point is available the point (0, 0) may be used as it lies on all $(t - t_0)_c$ vs. x curves. The closer the initial guess x_0 is to the value of x corresponding to the solution, the faster the convergence will be. One method of obtaining such a guess x_0 is to use a truncation of the infinite series obtained by direct inversion of the Kepler Equation (expressing x as a power series in $(t - t_0)$). It must be pointed out that this series diverges even for "moderate" transfer time intervals $(t - t_0)$; hence an iterative solution must be used to solve the Kepler equation for x in the general case. A third order truncation of the inversion of the Universal Kepler Equation is:

$$x = \sum_{n=0}^{3} \chi_n (t - t_0)^n$$

where

$$\chi_0 = 0, \quad \chi_1 = \sqrt{\mu_E}/r_0,$$

$$\chi_2 = -\frac{1}{2} \frac{\mu_E}{r_0^3} \left(\frac{\underline{r}_0 \cdot \underline{v}_0}{\sqrt{\mu_E}}\right),$$

$$\chi_3 = \frac{1}{6r_0} \left(\frac{\sqrt{\mu_E}}{r_0}\right)^3 \left[\frac{3}{r_0} \left(\frac{\underline{r}_0 \cdot \underline{v}_0}{\sqrt{\mu_E}}\right)^2 - (1 - r_0 \alpha_0)\right],$$

with $\alpha_0 = 2/r_0 - v_0^2/\mu_E$.

4.2 Conic State Extrapolation As A Function of Transfer Angle (Theta Routine)

As with the Kepler Routine, the universal formulation of Stumpff-Herrick-Battin in terms of the universal eccentric anomaly difference x is used in the Theta Routine. A completely analogous iteration scheme could have been formulated with x again as the independent variable and the transfer angle θ as the dependent variable using Marscher's universally valid equation:

$$\cot\frac{\theta}{2} = \frac{r_0\left[1 - \alpha_0 x^2 S(\alpha_0 x^2)\right]}{\sqrt{p}\, x\, C(\alpha_0 x^2)} + \cot\gamma_0$$

where

$$p = \left(\frac{r_0 v_0}{\sqrt{\mu_E}}\right)^2 \sin^2\gamma_0$$

and

$$\gamma_0 = \text{angle from } \underline{r}_0 \text{ to } \underline{v}_0.$$

However, in contrast to the Kepler equation, it is possible to invert the Marscher equation into a power series which can be made to converge as rapidly as desired, by means of which x may be calculated as a universal function of the transfer angle θ. Knowing x, we can directly calculate the transfer time interval $(t - t_0)_c$ and subsequently the extrapolated state vectors using the standard formulae.

The sequence of computations in the inversion of the Marscher Equation is as follows:

Let

$$p_N = p/r_0,\ \alpha_N = \alpha\, r_0$$

and

$$W_1 = \sqrt{p_N}\left(\frac{\sin\theta}{1-\cos\theta} - \cot\gamma_0\right).$$

If

$$|W_1| > 1, \text{ let } V_1 = 1.$$

Let

$$W_{n+1} = +\sqrt{W_n^2 + \alpha_N} + |W_n| \qquad (|W_1| \leq 1)$$

or

$$V_{n+1} = +\sqrt{V_n^2 + \alpha_N (|1/W_1|)^2} + V_n \qquad (|W_1| > 1).$$

Let

$$\omega_n = W_n \qquad (|W_1| \leq 1)$$

or

$$1/\omega_n = (|1/W_1|)/V_n \qquad (|W_1| > 1).$$

Let

$$\Sigma = \frac{2^n}{\omega_n} \sum_{j=0}^{\infty} \frac{(-1)^j}{2j+1} \left(\frac{\alpha_N}{\omega_n^2}\right)^j$$

where n is an integer ≥ 4. Then

$$x/\sqrt{r_0} = \begin{cases} \Sigma & (W_1 > 0) \\ 2\pi/\sqrt{\alpha_N} - \Sigma & (W_1 < 0) \end{cases}$$

The above equations have been specifically formulated to avoid certain numerical difficulties.

5. DETAILED FLOW DIAGRAMS

5.1 Conic State Extrapolation As A Function of Time (Kepler Routine)

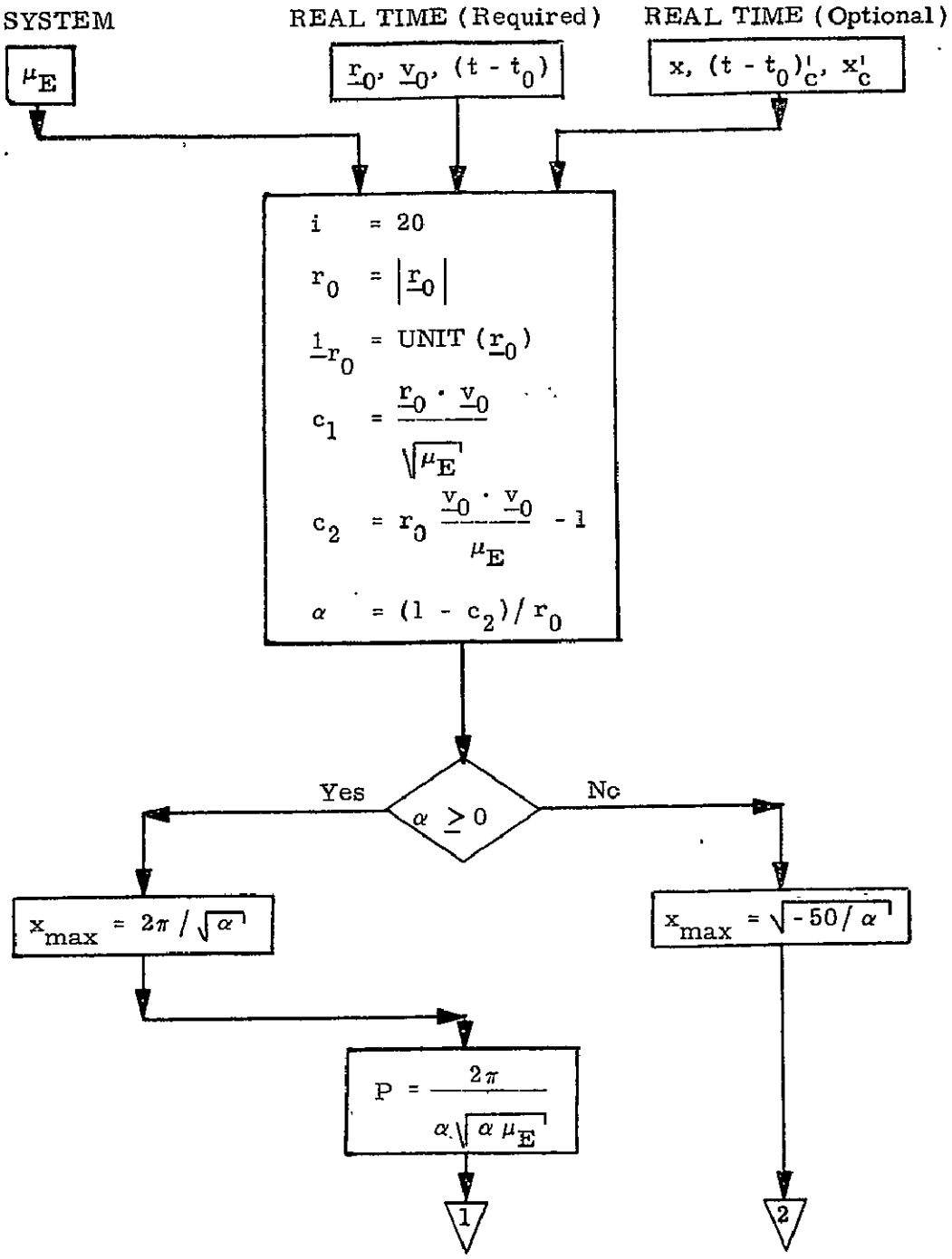


Figure 3a KEPLER ROUTINE DETAILED FLOW DIAGRAM

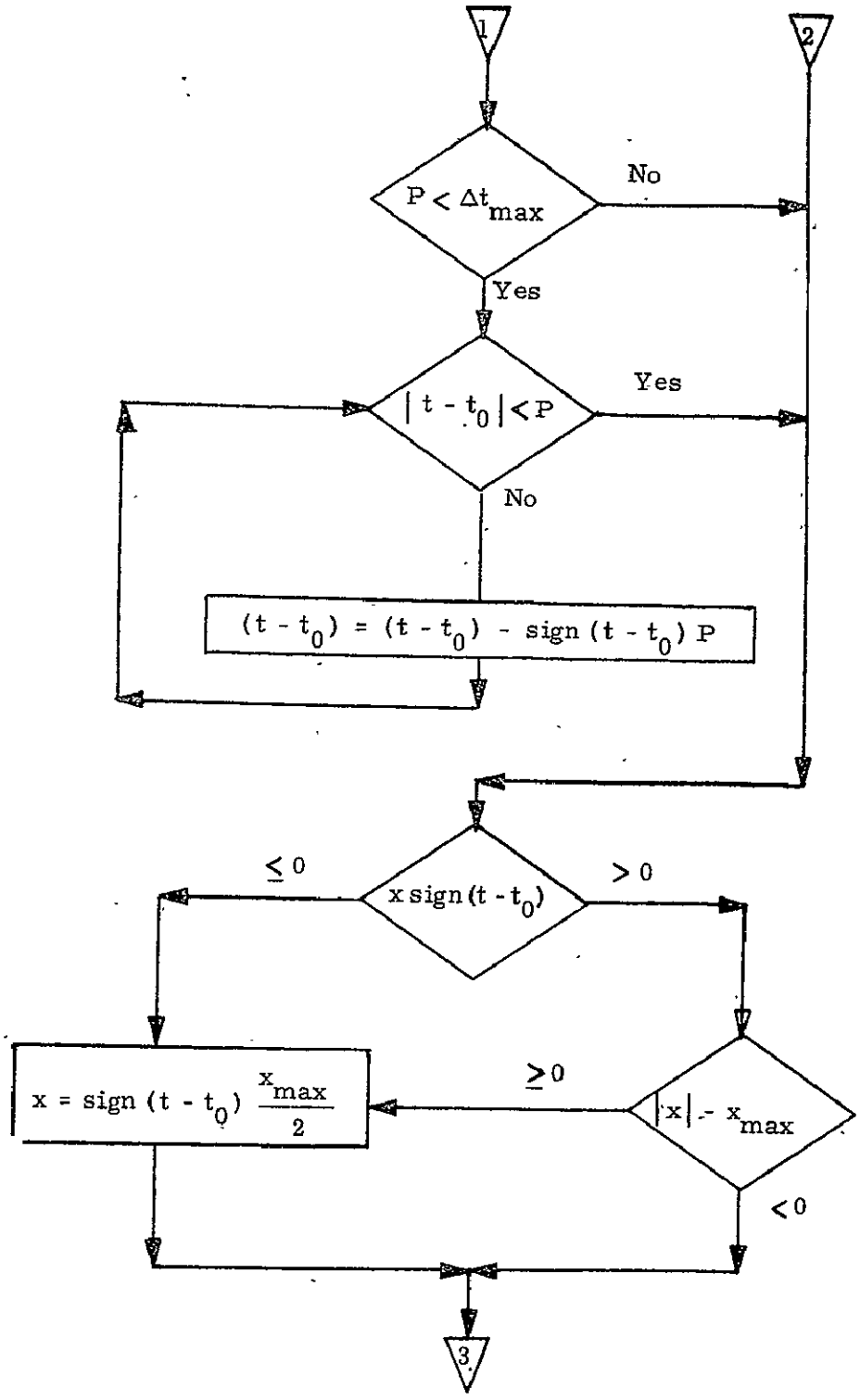


Figure 3b KEPLER ROUTINE DETAILED FLOW DIAGRAM

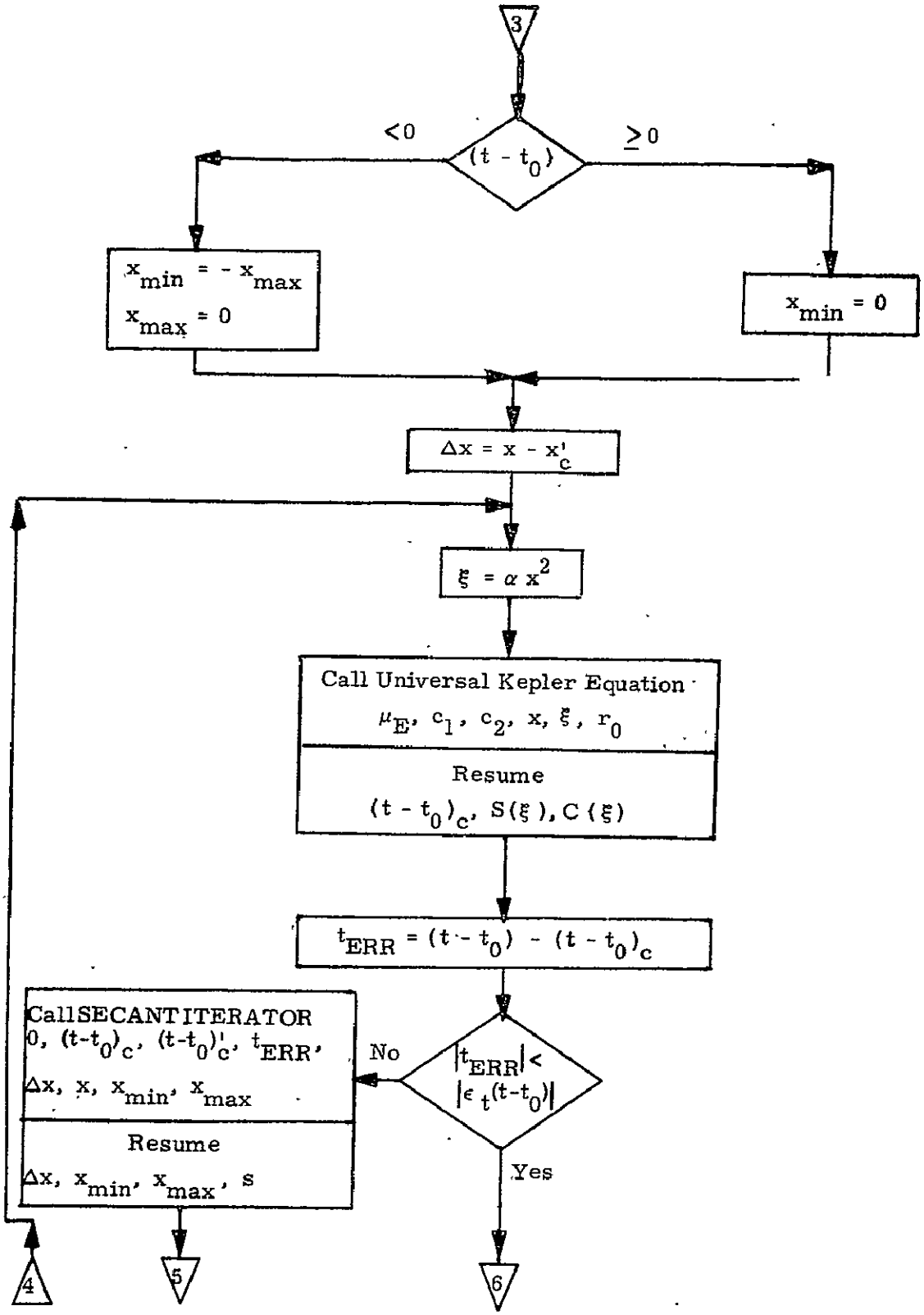


Figure 3c KEPLER ROUTINE DETAILED FLOW DIAGRAM

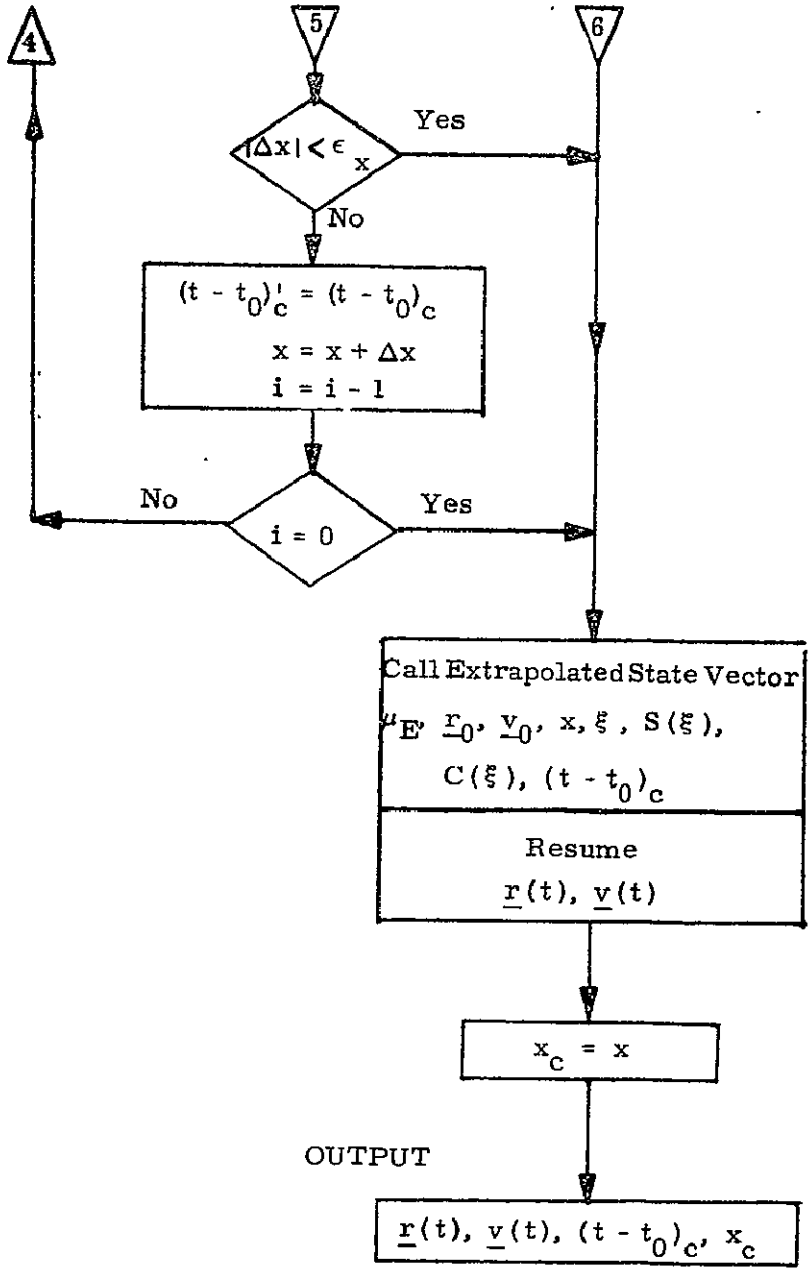


Figure 3d KEPLER ROUTINE DETAILED FLOW DIAGRAM

5.2 Conic State Extrapolation As A Function of Transfer Angle

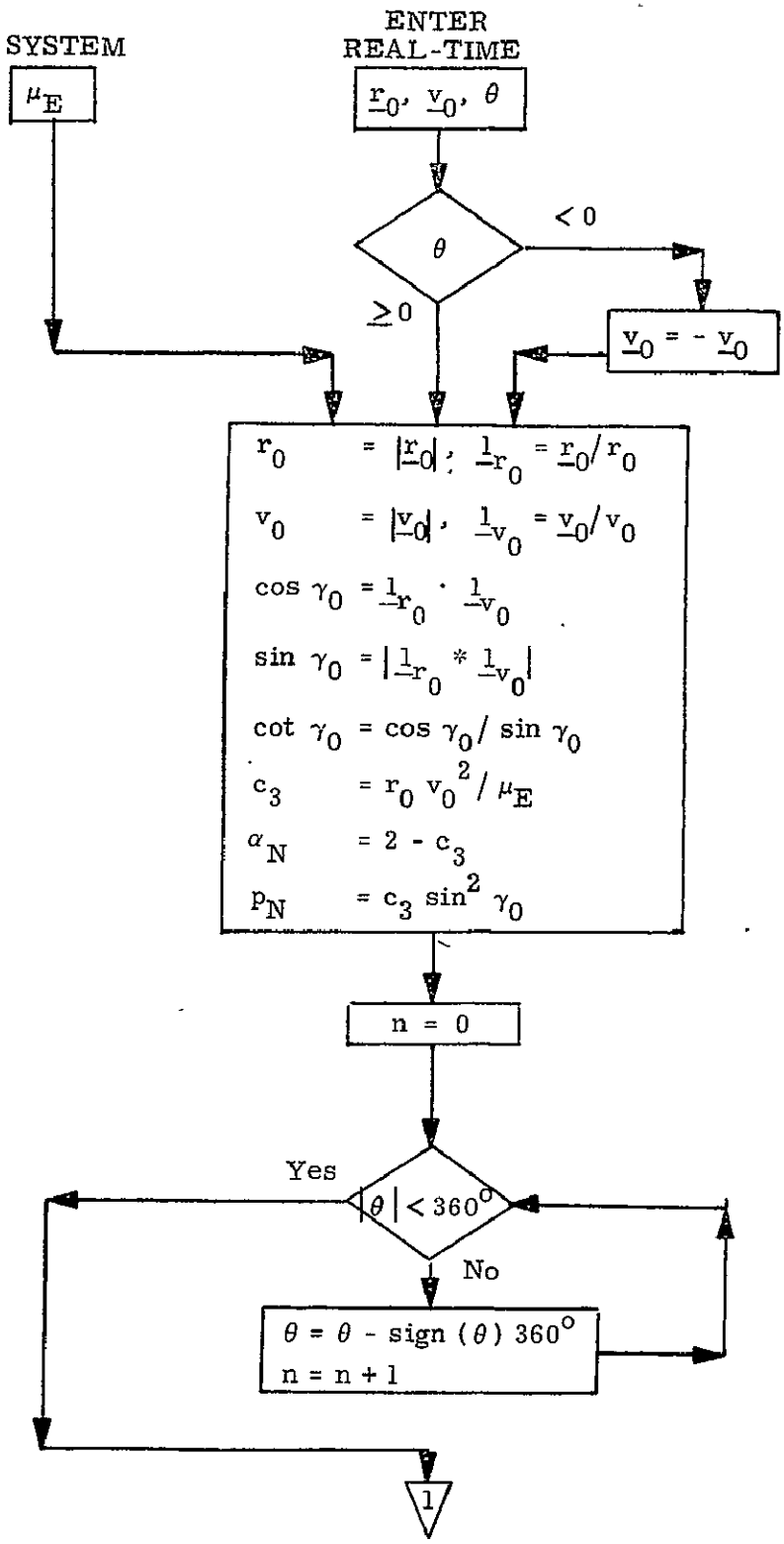


Figure 4a THETA ROUTINE DETAILED FLOW DIAGRAM

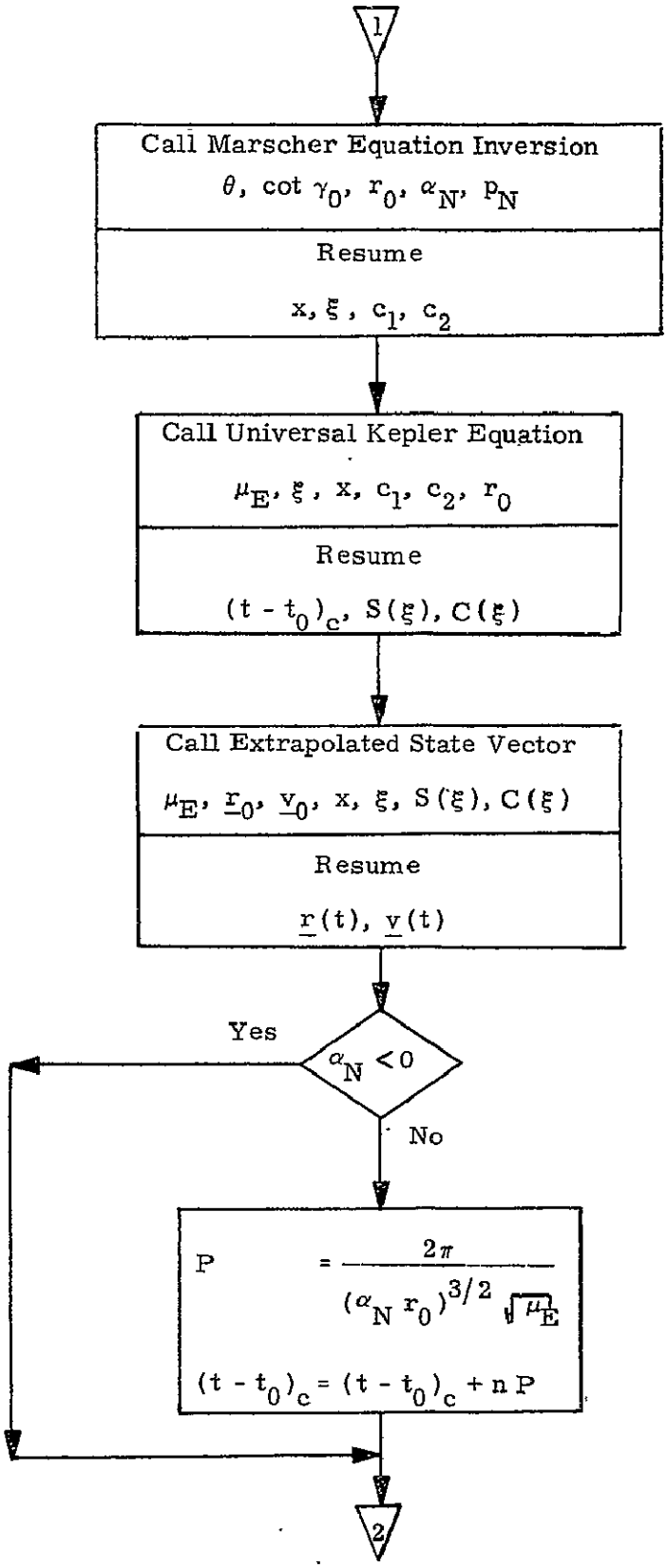


Figure 4b THETA ROUTINE DETAILED FLOW DIAGRAM

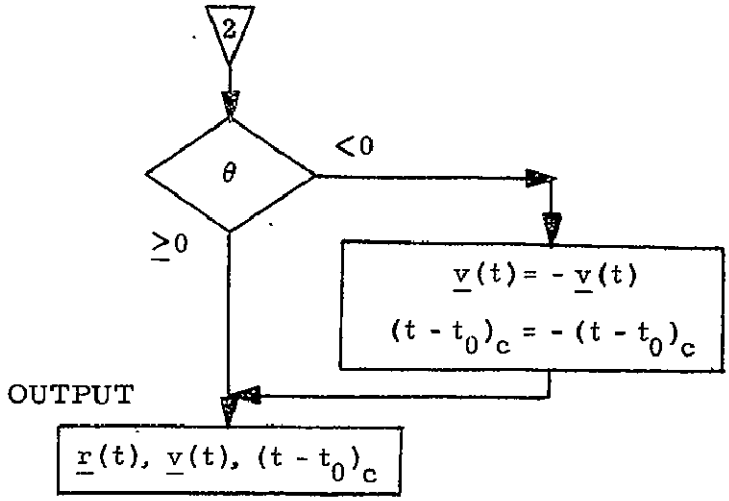


Figure 4c THETA ROUTINE DETAILED FLOW DIAGRAM

5.3 Subroutines Used By The Transfer Time or Transfer Angle Conic Extrapolation Routines

5.3.1 Universal Kepler Equation

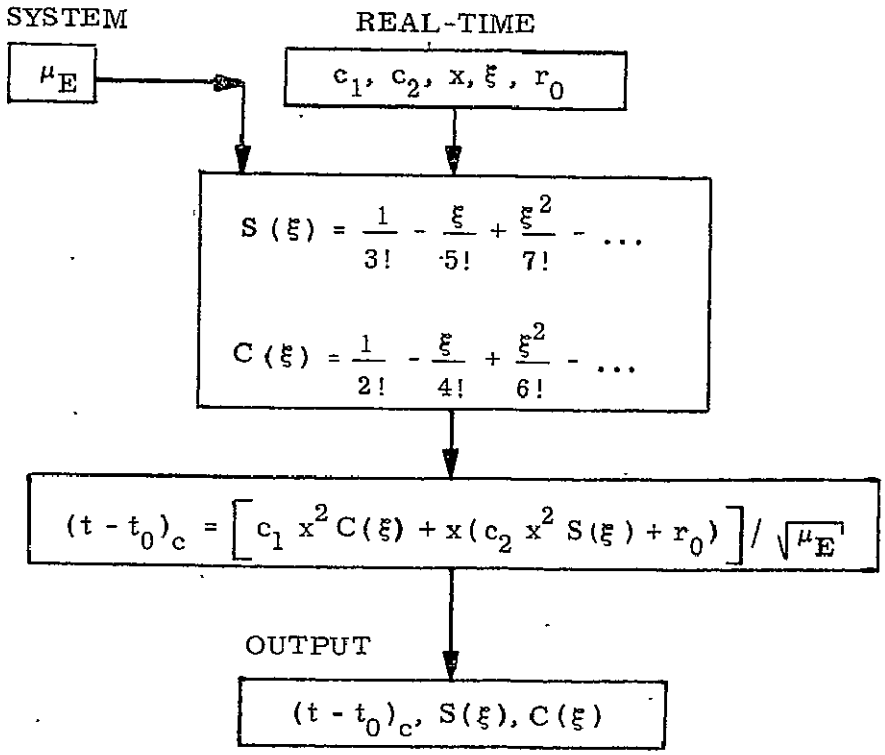


Figure 5 UNIVERSAL KEPLER EQUATION DETAILED FLOW DIAGRAM

5.3.2 Extrapolated State Vector

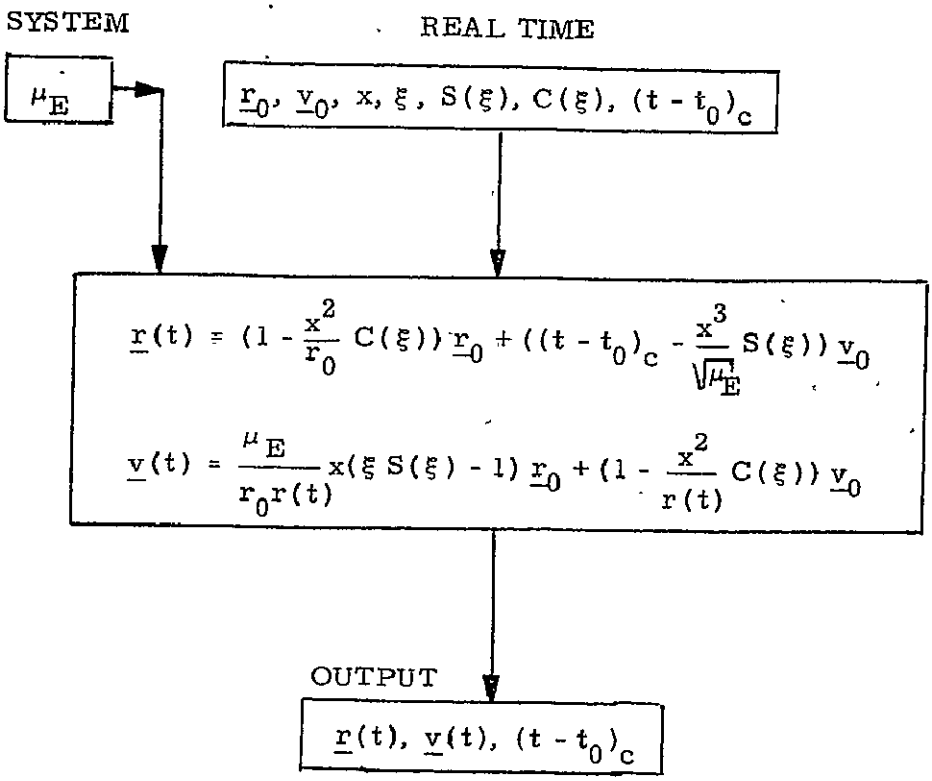


Figure 6 EXTRAPOLATED STATE VECTOR EQUATION
DETAILED FLOW DIAGRAM

5.3.3 Secant Iterator

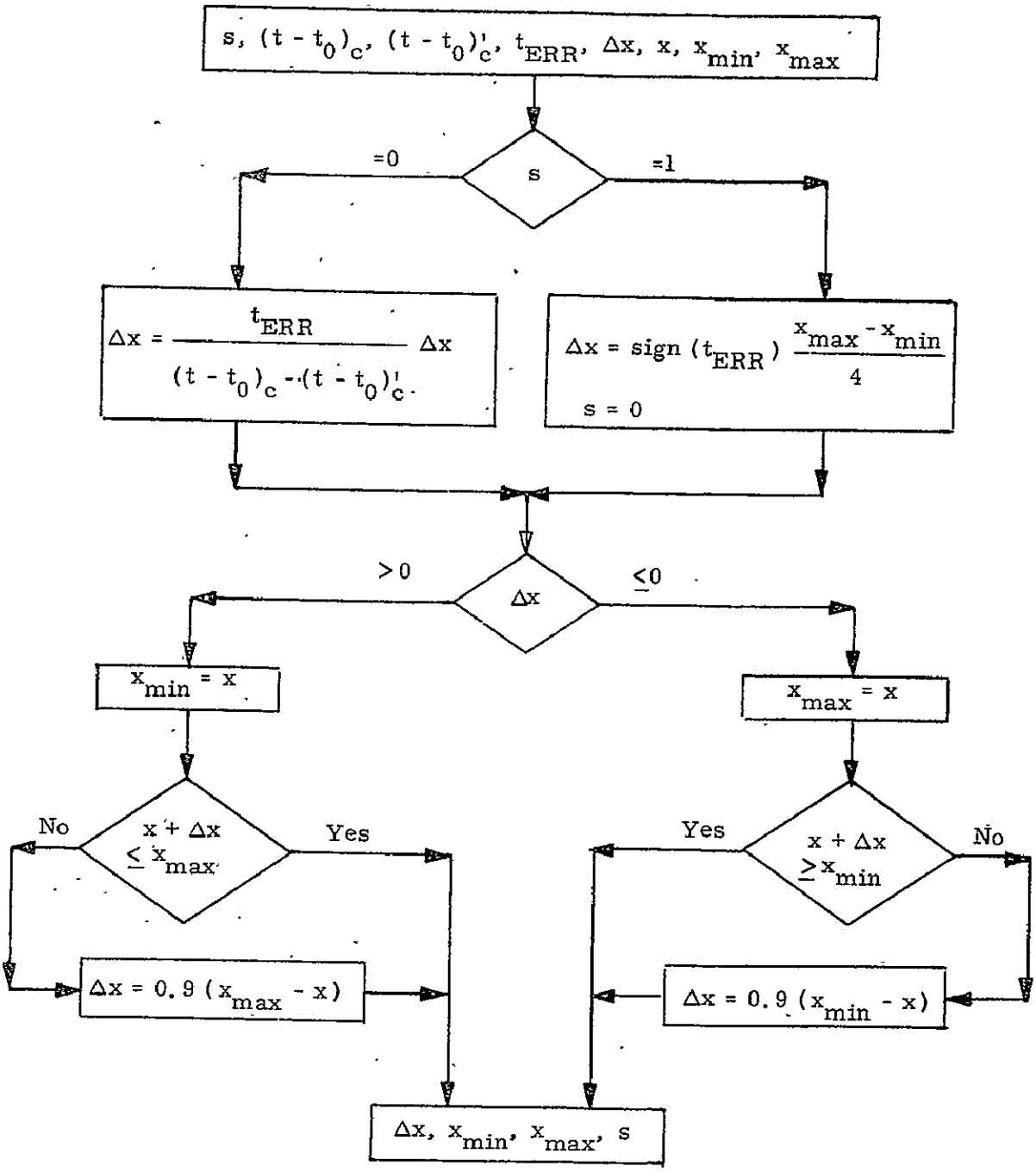


Figure 7 SECANT ITERATOR DETAILED FLOW DIAGRAM

5.3.4 Marscher Equation Inversion

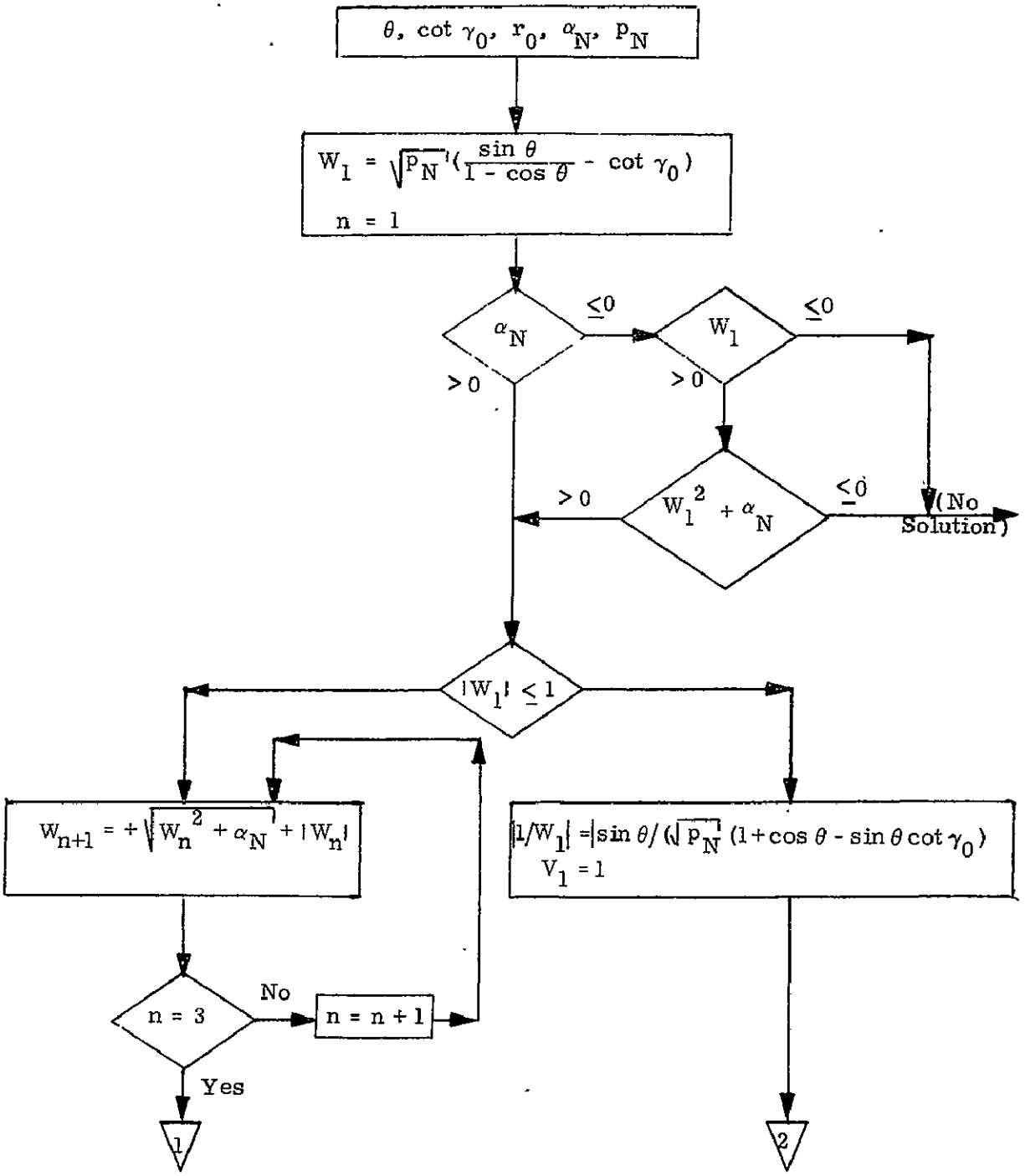


Figure 8a MARSCHER EQUATION INVERSION DETAILED FLOW DIAGRAM

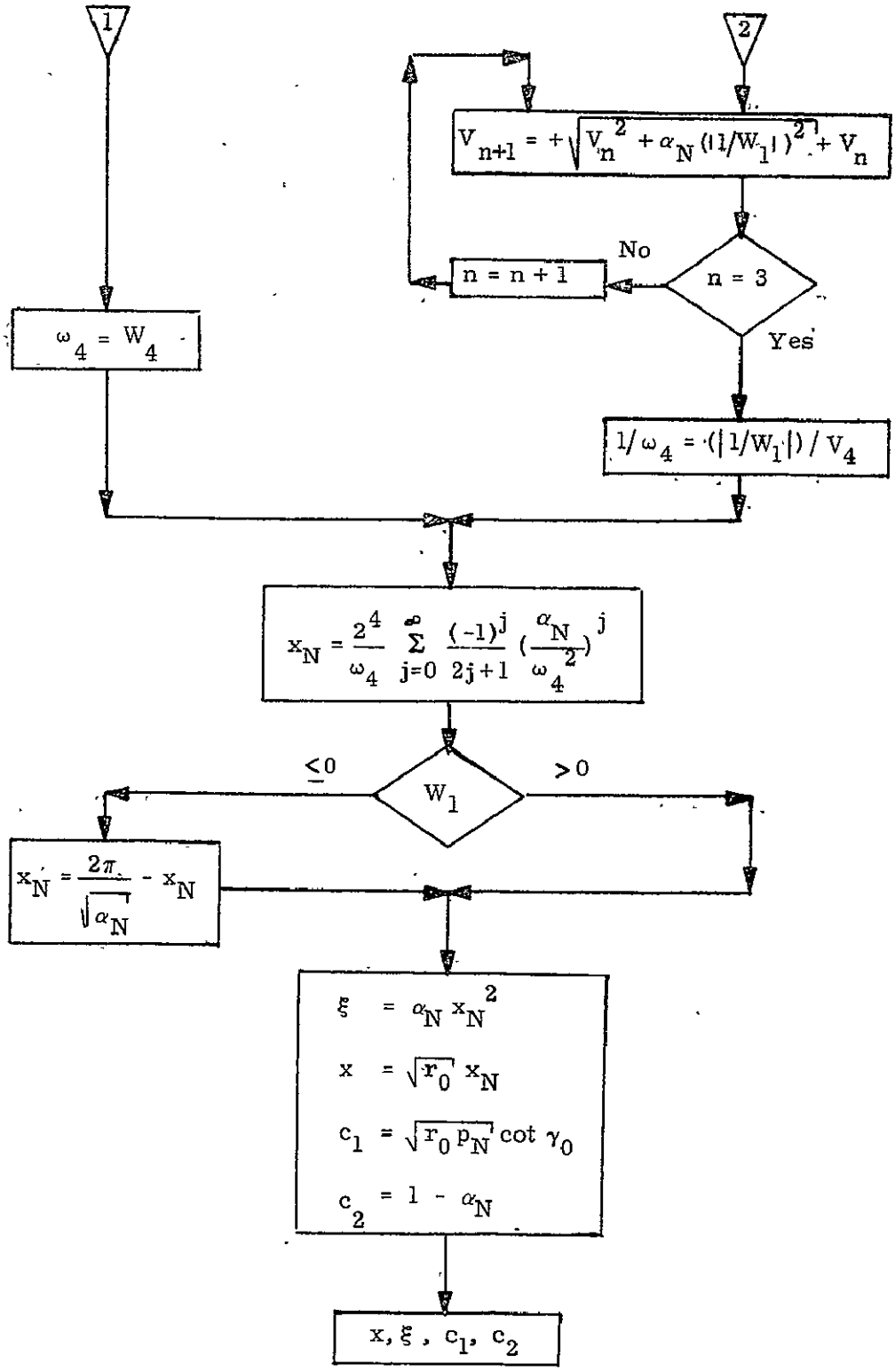


Figure 8b MARSCHER EQUATION INVERSION DETAILED FLOW DIAGRAM

6. SUPPLEMENTARY INFORMATION

The analytic expressions for the Universal Kepler Equation and the extrapolated position and velocity vectors are well known and are given by Battin (1964). Battin also outlines a Newton iteration technique for the solution of the Universal Kepler Equation; this technique converges somewhat faster than the secant technique but requires the evaluation of the derivative. It may be shown that if the derivative evaluation by itself takes more than 44% of the computation time used by the other calculations in one pass through the loop, then it is more efficient timewise to use the secant method.

Marscher's universal equation for $\cot \theta/2$ was derived by him in his report (Marscher, 1965), and is the generalization of his "Three-Cotangent" equation:

$$\cot \frac{\theta}{2} = \frac{r_0}{\sqrt{p\,a}} \cot \frac{(E - E_0)}{2} + \cot \gamma_0$$

Marscher has also outlined in the report an iterative method of extrapolating the state based on his universal equation. The inversion of Marscher's universal equation was derived by Robertson (1967a).

Krause organized the details of the computation in both routines.

A derivation of the coefficients in the inversion of the Universal Kepler Equation is given in Robertson (1967 b) and Newman (1967).

References

1. Battin, R.H., 1964, Astronautical Guidance, McGraw-Hill.

2. Krause, K.W., 1968, A Unified Method of Solving Initial Value and Boundary Value Conic Trajectory Problems, TRW Interoffice Correspondence #3424.9-15 (January 1968).

3. Marscher, W., 1965, A Unified Method of Generating Conic Sections, MIT/IL Report R-479, February 1965.

4. Newman, C.M., 1967, The Inversion of Kepler's Equation, MIT/IL Space Guidance Analysis Memo #14-67.

5. Robertson, W.M., 1967a, Explicit Universal Series Solutions for the Universal Variable x, MIT/IL Space Guidance Analysis Memo #8-67.

6. Robertson, W.M., 1967b, Time-Series Expansions of the Universal Variable x, MIT/IL Space Guidance Analysis Memo #13-67.

SPACE SHUTTLE

GN&C SOFTWARE EQUATION SUBMITTAL

Software Equation Section: Precision State and Filter Weighting Matrix Extrapolation

Submittal No. 7

Function: Provide precision advancement of the Inertial State and Weighting Matrix.

Module No. ON-2 Function No. 2,3 (MSC 03690)

Submitted by: W. M. Robertson Co. MIT No. 4-71

Date: Feb 1971

NASA Contact: J. Suddath Organization: GCD

Approved by Panel III: R.J. Cat Date: 3/10/71

Summary Description: Provides the capability to extrapolate any spacecraft geocentric state vector either backwards or forwards in time through a force field in which all significant perturbation effects have been included. Also provided is the capability of extrapolating the filter-weighting matrix along the precision trajectory.

Shuttle Configuration: These equations are independent of the Shuttle configuration.

Comments:

(Design Status) _____

(Verification Status) _____

Panel Comments:

9.6.1.2 Precision State and Filter Weighting Matrix Extrapolation (continued)

1. INTRODUCTION

The Precision State and Filter Weighting Matrix Extrapolation Routine provides the capability to extrapolate any spacecraft geocentric state vector either backwards or forwards in time through a force field in which all significant perturbation effects (such as the earth's oblateness, the attraction of the sun and moon, and atmospheric drag) have been included, as well as the earth's dominant central-force attraction. The Routine also provides the capability of extrapolating the filter-weighting matrix along the precision trajectory. This matrix, which is also known as the "W-matrix", contains statistical information relative to the accuracies of the state vectors and certain other quantities.

The routine is merely a coded algorithm for the numerical solution of modified forms of the basic differential equations which are satisfied by the geocentric state vector of the spacecraft's center of mass and by the filter-weighting matrix, namely

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

and

$$\frac{d}{dt}\,W(t) = F(t)\,W(t),$$

where $\underline{a}_d(t)$ is the vector sum of all the perturbing accelerations, and $F(t)$ is a matrix containing the gravity gradient matrix as one of its sub-blocks.

Because of its high accuracy and its capability of extrapolating the filter weighting matrix, this routine serves as the computational foundation for precise space navigation. It suffers from a relatively slow computation speed in comparison with the Conic State Extrapolation Routine.

NOMENCLATURE

$\underline{a}_d(t)$	Total perturbing acceleration at time t
$\underline{a}_{dDRAG}$	Perturbing acceleration due to atmospheric drag
a_{dMOON}	Perturbing acceleration due to moon's gravitational attraction
$\underline{a}_{dOBL}$	Perturbing acceleration due to earth's oblateness
$\underline{a}_{dSUN}$	Perturbing acceleration due to sun's gravitational attraction
$C_D A/M$	Spacecraft drag constant
C_{22}	Constant describing earth's oblateness
$(\cos \phi)$	Cosine of colatitude of spacecraft
D	Dimension of Filter Weighting Matrix (D = 0, 6, 7, 8, 9)
$E_n(t)$	Covariance matrix of dimension n
f(q)	Special function of q defined in text
f_{DRAG}	Flag to indicate whether drag perturbations are to be included
f_{SM}	Flag to indicate whether sun and moon perturbations are to be included
f_W	Flag controlling whether state or filter-weighting matrix integration is being performed (used only internally in routine)
G(t)	Gravity Gradient Matrix
h	Altitude

9.6.1.2 Precision State and Filter Weighting Matrix Extrapolation (continued)

I_3	Three-dimensional Identity Matrix
J_2, J_3, J_4	Constants describing earth's oblateness
M_ρ, M_T	Matrices of constants describing earth's atmospheric density
$P_i'(\cos\phi)$	Derivative of i-th Legendre polynomial
q	Special function of $\underline{r}$ and $\underline{\delta}$ defined in text
q_M	Special function of $\underline{r}$ and $\underline{r}_{EM}$ defined in text
q_S	Special function of $\underline{r}$ and $\underline{r}_{ES}$ defined in text
$\underline{r}_0$	Geocentric position vector at time t_0
$\underline{r}(t)$	Geocentric position vector at time t
$r(t)$	Magnitude of geocentric position vector
$\underline{r}_{con}(t)$	Reference conic position vector at time t
$r_{con}(t)$	Magnitude of reference conic position vector at time t
r_E	Mean equatorial radius of the earth
$\underline{r}_{EM}$	Position vector of moon with respect to earth
$\underline{r}_{ES}$	Position vector of sun with respect to earth
$\underline{r}_j$	Intermediate values of $\underline{r}$
r_{MC}	Distance of spacecraft from moon
r_{SC}	Distance of spacecraft from sun
S_{22}	Constant describing earth's oblateness
t_0	Initial time point

t_F	Time to which it is desired to extrapolate ($\underline{r}_0$, $\underline{v}_0$) and optionally W_0.
T_∞	Exospheric temperature
$\underline{v}_0$	Geocentric velocity vector at time t_0
$\underline{v}_{con}(t)$	Reference conic velocity vector at time t
$\underline{v}_{rel}$	Velocity vector of spacecraft relative to the atmosphere
W_0	Filter Weighting Matrix at time t_0
$W(t)$	Filter Weighting Matrix at time t
$\underline{w}_i$	Three-dimensional column vectors into which the W-matrix is partitioned
x_E	First component of $\underline{r}$ in earth-fixed coordinates
y_E	Second component of $\underline{r}$ in earth-fixed coordinates
z_E	Third component of $\underline{r}$ in earth-fixed coordinates
$\underline{\delta}(t)$	Position deviation vector of true position from reference conic position at time t
δ_{max}	Maximum value of $\lvert\underline{\delta}\rvert$ permitted (used as rectification criterion)
Δt	Time-step size in numerical integration of differential equation
Δt_{max}	Maximum permissible time-step size
Δt_{nom}	Nominal integration time-step size
ϵ_t	Time convergence tolerance criterion

$\underline{\epsilon}(t)$	Error in the estimate of the position vector
$\underline{\eta}(t)$	Error in the estimate of the velocity vector
λ_S	Local hour angle of the sun
μ_E	Earth's gravitational parameter
μ_M	Moon's gravitational parameter
μ_S	Sun's gravitational parameter
$\underline{\nu}(t)$	Velocity deviation vector of true velocity from reference conic velocity at time t
ν_{max}	Maximum value of $\lvert\underline{\nu}\rvert$ permitted (used as rectification criterion)
$\rho(\underline{r})$	Atmospheric density at $\underline{r}$
ϕ	Geocentric latitude
ω_E	Angular rotation rate of earth
$\underline{1}_r$	Unit vector in the direction of $\underline{r}$
$\underline{1}_x$, $\underline{1}_y$, $\underline{1}_z$	Earth-fixed vectors (1, 0, 0), (0, 1, 0), (0, 0, 1) transformed into reference coordinates

2. FUNCTIONAL FLOW DIAGRAM

The Precision State and Filter Weighting Matrix Extrapolation Routine performs its functions by integrating modified forms of the basic differential equations at a sequence of points separated by intervals known as time-steps, which are not necessarily of the same size. The routine automatically determines the size to be taken at each step. In close earth orbit about 21 steps are taken per trajectory revolution.

As shown in Fig. 1, the state vector and (optionally) the filter-weighting matrix are updated one step at a time along the precision trajectory until the specified overall transfer time interval is exactly attained. (The size of the last time-step is adjusted as necessary to make this possible.)

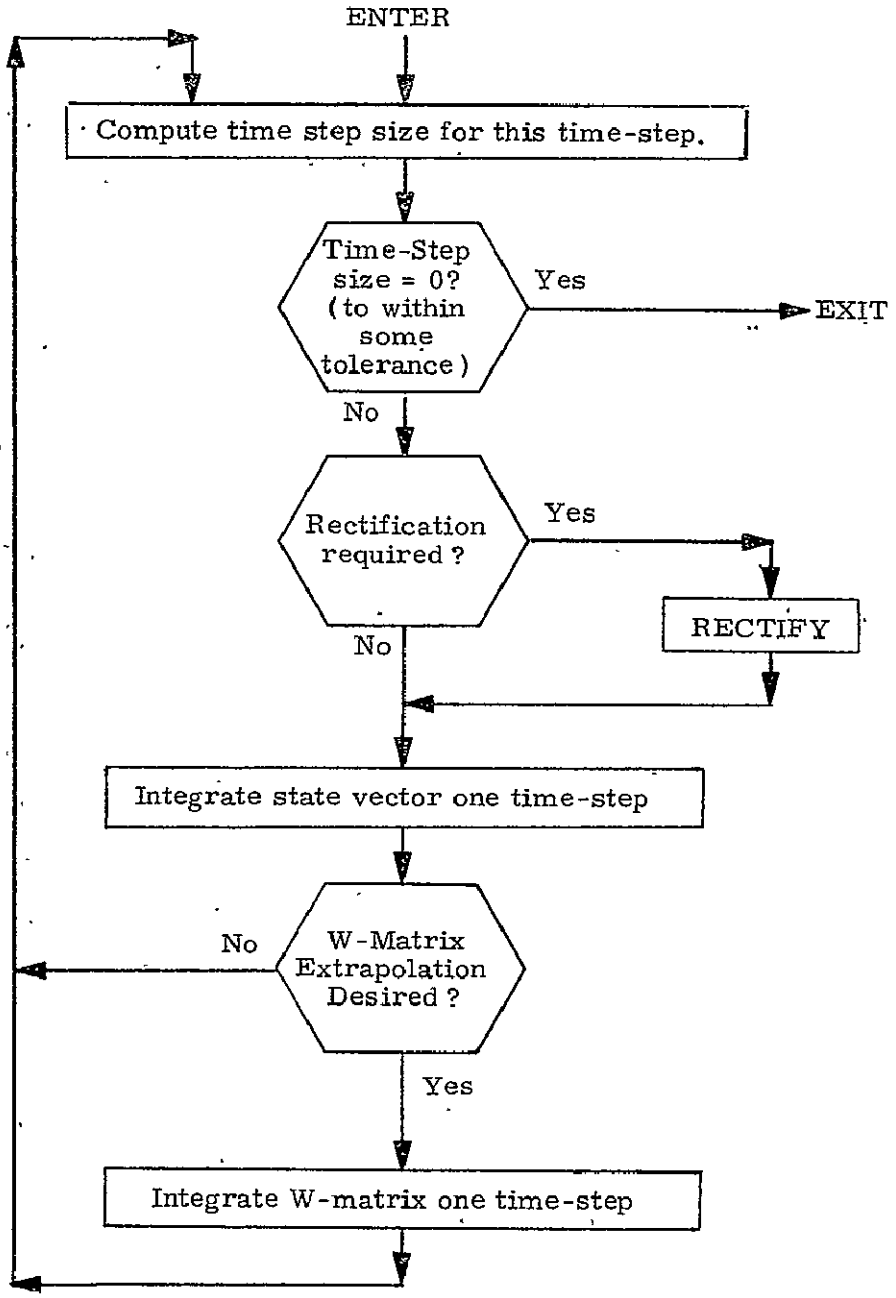


Figure 1 FUNCTIONAL FLOW DIAGRAM, PRECISION STATE AND FILTER WEIGHTING MATRIX EXTRAPOLATION ROUTINE

3. PROGRAM INPUT-OUTPUT

The Precision State and Filter Weighting Matrix Extrapolation Routine has the following input and output parameters:

Input Parameters

System	
μ_E	Gravitational parameter of the earth
J_2, J_3, J_4, C_{22}, S_{22}	Constants describing the earth's oblateness
(TBD)	Constants describing the sun's motion
(TBD)	Constants describing the moon's motion
M_ρ, M_T	Matrices of constants describing the earth's atmospheric density
$C_D A/M$	Spacecraft drag constant
ω_E	Angular rotation rate of the earth

Real-Time	
$(\underline{r}_0, \underline{v}_0)$	Geocentric state vector to be extrapolated
t_0	Time associated with $(\underline{r}_0, \underline{v}_0)$ and W_0
t_F	Time to which it is desired to extrapolate $(\underline{r}_0, \underline{v}_0)$ and optionally W_0.
W_0	Filter Weighting (W) matrix to be extrapolated (optional)
D	Dimension of W-matrix (0 indicates omit W-matrix)
f_{SM}	Flag to indicate whether sun and moon perturbations are to be included
f_{DRAG}	Flag to indicate whether drag perturbations are to be included

Output Parameters

$(\underline{r}(t_F), \underline{v}(t_F))$	Extrapolated geocentric state vector
$W(t_F)$	Extrapolated filter-weighting matrix

4. DESCRIPTION OF EQUATIONS

4.1 Precision State Extrapolation Equations

Since the perturbing acceleration is small compared with the central force field, direct numerical integration of the basic differential equations of motion of the spacecraft state vector is inefficient. Instead, a technique due to Encke is utilized in which only the deviations of the state from a reference conic orbit are numerically integrated The positions and velocities along the reference conic are obtained from the Kepler routine.

At time t_0 the position and velocity vectors, $\underline{r}_0$ and $\underline{v}_0$, define an osculating conic orbit. Because of the perturbing accelerations, the true position and velocity vectors $\underline{r}(t)$ and $\underline{v}(t)$ will deviate as time progresses from the conic position and velocity vectors $\underline{r}_{con}(t)$ and $\underline{v}_{con}(t)$ which have been conically extrapolated from $\underline{r}_0$ and $\underline{v}_0$. Let

$$\underline{\delta}(t) = \underline{r}(t) - \underline{r}_{con}(t)$$
$$\underline{\nu}(t) = \underline{v}(t) - \underline{v}_{con}(t)$$

be the vector deviations. It can be shown that the position deviation $\underline{\delta}(t)$ satisfies the differential equation

$$\frac{d^2}{dt^2}\underline{\delta}(t) + \frac{\mu_E}{r_{con}^3(t)}\left[f(q)\,\underline{r}(t) + \underline{\delta}(t)\right] = \underline{a}_d(t)$$

with the initial conditions

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

where

$$q = \frac{(\underline{\delta} - 2\underline{r}) \cdot \underline{\delta}}{r^2},\quad f(q) = q\,\frac{3 + 3q + q^2}{1 + (1+q)^{3/2}},$$

and $\underline{a}_d(t)$ is the total perturbing acceleration. The term

$$\frac{\mu_E}{r_{con}^3}\left[f(q)\,\underline{r}(t) + \underline{\delta}(t)\right]$$

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 will eventually increase in size. When

$$|\delta(t)| > 0.01\,|\underline{r}_{con}(t)| \text{ or } |\underline{\nu}(t)| > 0.01\,|\underline{v}_{con}(t)|$$

or when

$$|\underline{\delta}(t)| > \delta_{max} \text{ or } |\underline{\nu}(t)| > \nu_{max},$$

a new osculating conic orbit is established based on the latest precision position and velocity vectors $\underline{r}(t)$ and $\underline{v}(t)$, the deviations $\underline{\delta}(t)$ and $\underline{\nu}(t)$ are zeroed, and the numerical integration of $\underline{\delta}(t)$ and $\underline{\nu}(t)$ continues. The process of establishing a new conic orbit is called rectification.

4.2 Explicit Form of the Perturbing Acceleration

The perturbing acceleration $a_d(t)$ is the vector sum of four separate accelerations, namely those due to the earth's oblateness $\underline{a}_{dOBL}$, the gravitational attractions of the sun $\underline{a}_{dSUN}(t)$ and moon $\underline{a}_{dMOON}(t)$, and the earth's atmospheric drag $\underline{a}_{dDRAG}(t)$. Thus:

$$\underline{a}_d(t) = \underline{a}_{dOBL} + \underline{a}_{dSUN} + \underline{a}_{dMOON} + \underline{a}_{dDRAG}.$$

4.2.1 The earth oblateness acceleration is given by the truncated spherical harmonic expansion involving the zonal terms J_2, J_3, J_4, and the sectorial term J_{22}:

$$\underline{a}_{dOBL} = \frac{\mu_E}{r^2}\left\{\sum_{i=2}^{4} J_i\left(\frac{r_E}{r}\right)^i\left[P'_{i+1}(\cos\phi)\,\underline{1}_r - P'_i(\cos\phi)\,\underline{1}_z\right]\right.$$

$$+ 3\left(\frac{r_E}{r}\right)^2\left[\left(\frac{2x_E}{r}\,\underline{1}_x - \frac{2y_E}{r}\,\underline{1}_y - \frac{5(x_E^2 - y_E^2)}{r^2}\,\underline{1}_r\right) C_{22}\right.$$

$$\left.\left. + \left(\frac{2y_E}{r}\,\underline{1}_x + \frac{2x_E}{r}\,\underline{1}_y - \frac{10x_E\,y_E}{r^2}\,\underline{1}_r\right) S_{22}\right]\right\}$$

where

$\underline{1}_r$ is the unit position vector in reference coordinates,

$\underline{1}_x$, $\underline{1}_y$, $\underline{1}_z$ are the earth-fixed vectors (1,0,0), (0,1,0), (0,0,1) transformed into reference coordinates,

x_E, y_E, z_E are the components of $\underline{r}$ in earth-fixed coordinates,

$\cos\phi = \underline{1}_r \cdot \underline{1}_z$,

r_E is the mean equatorial radius of the earth,

and where

$$P'_2(\cos\phi) = 3\cos\phi$$
$$P'_3(\cos\phi) = \frac{1}{2}(15\cos^2\phi - 3)$$
$$P'_4(\cos\phi) = \frac{1}{3}(7\cos\phi\,P'_3 - 4P'_2)$$
$$P'_5(\cos\phi) = \frac{1}{4}(9\cos\phi\,P'_4 - 5P'_3)$$

are the derivatives of the Legendre polynomials.

4.2.2 The gravitational accelerations due to the sun and moon on a spacecraft in earth orbit are given by:

$$\underline{a}_{dS} = \frac{-\mu_S}{r_{SC}^3}\left[f(q_S)\,\underline{r}_{ES} + \underline{r}\right]$$

$$\underline{a}_{dM} = \frac{-\mu_M}{r_{MC}^3}\left[f(q_M)\,\underline{r}_{EM} + \underline{r}\right]$$

where $\underline{r}_{ES}$ and $\underline{r}_{EM}$ are the position vectors of the sun and moon with respect to the earth, r_{SC} and r_{MC} are distances of the spacecraft from the sun and moon ($\underline{r}_{SC} = \underline{r} - \underline{r}_{ES}$, $\underline{r}_{MC} = \underline{r} - r_{EM}$) and the arguments q_S and q_M are computed from

$$q_S = \frac{(\underline{r} - 2\underline{r}_{ES}) \cdot \underline{r}}{r_{ES}^2}$$

$$q_M = \frac{(\underline{r} - 2\underline{r}_{EM}) \cdot \underline{r}}{r_{EM}^2}$$

and $f(q_S)$ and $f(q_M)$ are calculated from the equation for the function f given in Section 4.1.

4.2.3 The drag acceleration is given by:

$$\underline{a}_{dDRAG} = -\frac{1}{2}\left(\frac{C_D A}{M}\right)\rho(\underline{r})\,|\underline{v}_{rel}|\,\underline{v}_{rel}$$

where $(C_D A / M)$ is a constant depending on the coefficient of drag, and the cross sectional area and mass of the spacecraft, $\rho(\underline{r})$ is the density of the atmosphere at the point in question, and $\underline{v}_{rel}$ is the velocity vector of the spacecraft relative to the atmosphere and is computed by assuming the atmosphere is rotating at the same angular velocity as the earth, i.e.

$$\underline{v}_{rel} = \underline{v} - (\underline{1}_z * \underline{r})\ \omega_E$$

where $\underline{v}$ is the geocentric velocity, $\underline{1}_z$ the unit vector of the earth's north pole expressed in reference coordinates, and ω_E is the angular rotation rate of the earth. The density ρ is obtained from an approximation to the Jacchia density model in the form of two binomial fits; the exospheric temperature is approximated by a binomial in latitude and local hour angle of the sun, and the density is then approximated by a binomial in the exospheric temperature and the altitude. Specifically, Let ϕ, λ_S, h, T_∞ be the geocentric latitude, local hour angle of the sun, altitude and exospheric temperature respectively at the position $\underline{r}$; and let

$$\phi' = \phi - \phi_0, \quad \lambda'_S = \lambda_S - \lambda_{S_0}$$
$$h' = h - h_0, \quad T'_\infty = T_\infty - T_{\infty_0}$$

be "adjusted" values of these parameters. Further, let $\underline{\phi}'$, $\underline{\lambda}'_S$, $\underline{h}'$, $\underline{T}'$ denote the vectors

$$\underline{\phi}' = \begin{bmatrix} 1 \\ \phi' \\ (\phi')^2 \\ (\phi')^3 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}, \quad \underline{\lambda}'_S = \begin{bmatrix} 1 \\ \lambda'_S \\ (\lambda'_S)^2 \\ (\lambda'_S)^3 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}, \quad \underline{h}' = \begin{bmatrix} 1 \\ h' \\ (h')^2 \\ (h')^3 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}, \quad \underline{T}' = \begin{bmatrix} 1 \\ T'_\infty \\ (T'_\infty)^2 \\ (T'_\infty)^3 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

whose components are the various powers of the adjusted parameters. Then the double binomial approximation to the atmospheric density may be written as:

$$= (\underline{\phi}')^T M_T (\underline{\lambda}'_S), \quad \rho = (\underline{T}')^T M_\rho (\underline{h}')$$

where M_T and M_ρ are rectangular matrices whose elements are constants. For an altitude range of 60 nautical miles, it has been found sufficient to have $\underline{\phi}'$ and $\underline{\lambda}_S'$ of dimension six, $\underline{h}'$ of dimention nine, and $\underline{T}'$ of dimension four; the matrices M_T and M_ρ are thus 6 x 6 and 4 x 9 in size.

4.3 Filter-Weighting (W) Matrix Extrapolation

The position and velocity vectors as maintained in the spacecraft's computer are only estimates of the true values. As part of the navigation technique it is necessary also to maintain statistical data in the computer to aid in the processing of navigation measurements. The filter weighting W-matrix is used for this purpose.

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 error covariance matrix $E_6(t)$ is defined by:

$$E_6(t) = \begin{pmatrix} \overline{\underline{\epsilon}(t)\,\underline{\epsilon}(t)^T} & \overline{\underline{\epsilon}(t)\,\underline{\eta}(t)^T} \\ \overline{\underline{\eta}(t)\,\underline{\epsilon}(t)^T} & \overline{\underline{\eta}(t)\,\underline{\eta}(t)^T} \end{pmatrix}$$

In certain applications it becomes necessary to expand the state vector and the covariance matrix to more than six dimensions so as to include the estimation of various other quantities such as landmark locations during orbit navigation and certain instrument biases during co-orbiting vehicle navigation. For this purpose a nine-dimensional covariance matrix is defined as follows:

$$E_9(t) = \begin{pmatrix} & & \overline{\underline{\epsilon}(t)\,\underline{\beta}^T} \\ & E_6(t) & \overline{\underline{\eta}(t)\,\underline{\beta}^T} \\ \overline{\underline{\beta}\,\underline{\epsilon}(t)^T} & \overline{\underline{\beta}\,\underline{\eta}(t)^T} & \overline{\underline{\beta}\,\underline{\beta}^T} \end{pmatrix}$$

where the components of the three-dimensional vector $\underline{\beta}$ are the errors in the estimates of three variables which are estimated in addition to the components of the spacecraft state vector. (In some navigation procedures, only one or two additional quantities are estimated).

Finally if the position and velocities of two separate spacecraft are to be maintained simultaneously as well as the additional estimated quantities $\underline{\beta}$, then a fifteen dimensional covariance matrix $E_{15}(t)$ is defined:

$$E_{15}(t) = \left(\begin{array}{c|c} E_6(t) & O \\ \hline O & E_9(t) \end{array}\right)$$

Rather than use one of the covariance matrices E_6, E_9, or E_{15} in the navigation procedure, it is more convenient to use a matrix $W_n(t)$ having the same dimension n as the covariance matrix $E_n(t)$ and defined by

$$E_n(t) = W_n(t)\, W_n(t)^T.$$

$W_n(t)$ is called the filter weighting matrix, and is (in a certain sense) the square root of the covariance matrix.

9.6.1.2 Precision State and Filter Weighting Matrix Extrapolation (continued)

Extrapolation of the six and nine-dimensional W matrices is made by direct numerical integration of the basic differential equations

$$\frac{d}{dt} W_6(t) = \begin{pmatrix} O & I_3 \\ G(t) & O \end{pmatrix} W_6(t)$$

$$\frac{d}{dt} W_9(t) = \begin{pmatrix} O & I_3 & O \\ G(t) & O & O \\ O & O & O \end{pmatrix} W_9(t)$$

where I_3 is the 3 x 3 identity matrix and G(t) is the 3 x 3 gravity gradient matrix

$$G(t) = \frac{\mu_E}{r^5(t)} \left[3\underline{r}(t)\,\underline{r}(t)^T - r^2(t)\, I_3 \right]$$

If the W matrix is partitioned as

$$W_9 = \begin{pmatrix} \underline{w}_0 & \underline{w}_1 & \cdots & \underline{w}_8 \\ \underline{w}_9 & \underline{w}_{10} & \cdots & \underline{w}_{17} \\ \underline{w}_{18} & \underline{w}_{19} & \cdots & \underline{w}_{26} \end{pmatrix}$$

where the $\underline{w}_i$ are three dimensional column vectors, then the basic differential equations may be written:

$$\left.\begin{aligned} \frac{d}{dt}\underline{w}_i(t) &= \underline{w}_{i+9}(t) \\ \frac{d}{dt}\underline{w}_{i+9}(t) &= G(t)\,\underline{w}_i(t) \\ \frac{d}{dt}\underline{w}_{i+18}(t) &= \underline{0} \end{aligned}\right\} \quad i = 0, 1, \ldots, 8$$

Hence, the extrapolation of the W_9 matrix may be accomplished by successively integrating the second order vector differential equations

$$\frac{d^2}{dt^2}\underline{w}_i(t) = G(t)\,\underline{w}_i(t) \qquad i = 0, 1, \ldots, 8,$$

or equivalently

$$\frac{d^2}{dt^2}\underline{w}_i(t) = \frac{\mu_E}{r^5(t)}\left\{3\left[\underline{r}(t)\cdot\underline{w}_i(t)\right]\underline{r}(t) - r^2(t)\,\underline{w}_i(t)\right\}$$

$$i = 0, 1, \ldots, 8$$

and utilizing the identities:

$$\left.\begin{aligned} \underline{w}_{i+9}(t) &= \frac{d}{dt}\underline{w}_i(t) \\ \underline{w}_{i+18}(t) &= \text{constant} \end{aligned}\right\} \quad i = 0, 1, \ldots, 8.$$

The derivatives $\frac{d}{dt}\underline{w}_i(t)$ $i = 0, 1, \ldots, 8$ are a by-product of the numerical integration of the second order equations.

Similarly, the extrapolation of the W_6 matrix involves the same equations with the index i running only up to five ($i = 0, 1, \ldots, 5$), while the extrapolation of the W_{15} matrix is merely the separate extrapolation of independent W_6 and W_9 matrices.

4.4 Numerical Integration Method

The extrapolation of inertial state vectors and filter weighting matrices requires the numerical solution of two second-order vector differential equations, which are special cases of the general form

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

where

$$\underline{z} = \frac{d}{dt}\,\underline{y}\,.$$

Nystrom's standard fourth-order method is utilized to numerically solve this equation. The algorithm for this method is:

$$\underline{y}_{n+1} = \underline{y}_n + \underline{z}_n\,\Delta t + \frac{1}{6}(\underline{k}_1 + \underline{k}_2 + \underline{k}_3)\,(\Delta t)^2$$

$$\underline{z}_{n+1} = \underline{z}_n + \frac{1}{6}(\underline{k}_1 + 2\underline{k}_2 + 2\underline{k}_3 + \underline{k}_4)\,\Delta t$$

$$\underline{k}_1 = \underline{f}\,(t_n, \underline{y}_n, \underline{z}_n)$$

$$\underline{k}_2 = \underline{f}\,(t_n + \frac{1}{2}\Delta t,\ \underline{y}_n + \frac{1}{2}\underline{z}_n\,\Delta t + \frac{1}{8}\underline{k}_1(\Delta t)^2,\ \underline{z}_n + \frac{1}{2}\underline{k}_1\,\Delta t)$$

$$\underline{k}_3 = \underline{f}\,(t_n + \frac{1}{2}\Delta t,\ \underline{y}_n + \frac{1}{2}\underline{z}_n\,\Delta t + \frac{1}{8}\underline{k}_1(\Delta t)^2,\ \underline{z}_n + \frac{1}{2}\underline{k}_2\,\Delta t)$$

$$\underline{k}_4 = \underline{f}\,(t_n + \Delta t,\ \underline{y}_n + \underline{z}_n\,\Delta t + \frac{1}{2}\underline{k}_3(\Delta t)^2,\ \underline{z}_n + \underline{k}_3\,\Delta t)$$

where

$$\underline{y}_n = \underline{y}(t_n),\ \underline{z}_n = \underline{z}(t_n)$$

and

$$t_{n+1} = t_n + \Delta t$$

20

As can be seen, the method requires four evaluations of $\underline{f}(t, \underline{y}, \underline{z})$ per integration step Δt as does the classical fourth-order Runge-Kutta method when it is extended to second-order equations. However, if $\underline{f}$ is independent of $\underline{z}$, then Nystrom's method above only requires three evaluations per step since $\underline{k}_3 = \underline{k}_2$. (Runge-Kutta's method will still require four).

The integration time step Δt may be varied from step to step. The nominal integration step size is

$$\Delta t_{nom} = K \, r_{con}^{3/2} / \sqrt{\mu_E}$$

with K = 0.3. The actual step size is however limited to a maximum of

$$\Delta t_{max} = 4000 \text{ seconds.}$$

Also, in the last step, the actual step size is taken to be the interval between the end of the previous step and the desired integration end-point, so that the extrapolated values of the state or W-matrix are immediately available. Thus the integration step-size Δt is given by the formula

$$\Delta t = \pm \text{minimum} \left(\left| t_F - t \right|, \Delta t_{nom}, \Delta t_{max} \right)$$

where t_F is the desired integration end-point and t is the time at the end of the previous step. The plus sign is used if forward extrapolation is being performed, while the negative sign is used in the back-dating case.

5. DETAILED FLOW DIAGRAMS

This section contains detailed flow diagrams of the Precision State and Filter Weighting Matrix Extrapolation Routine.

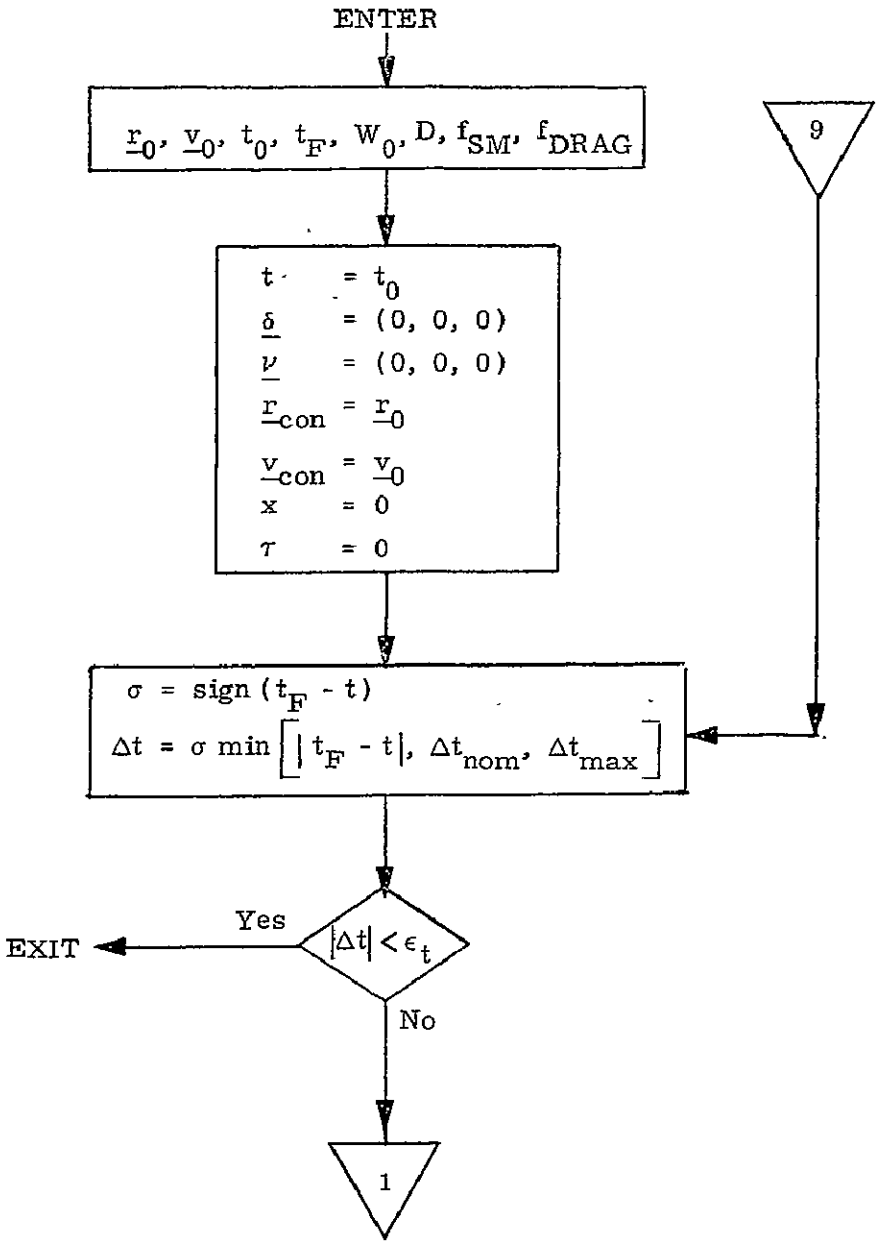


Figure 2a DETAILED FLOW DIAGRAM

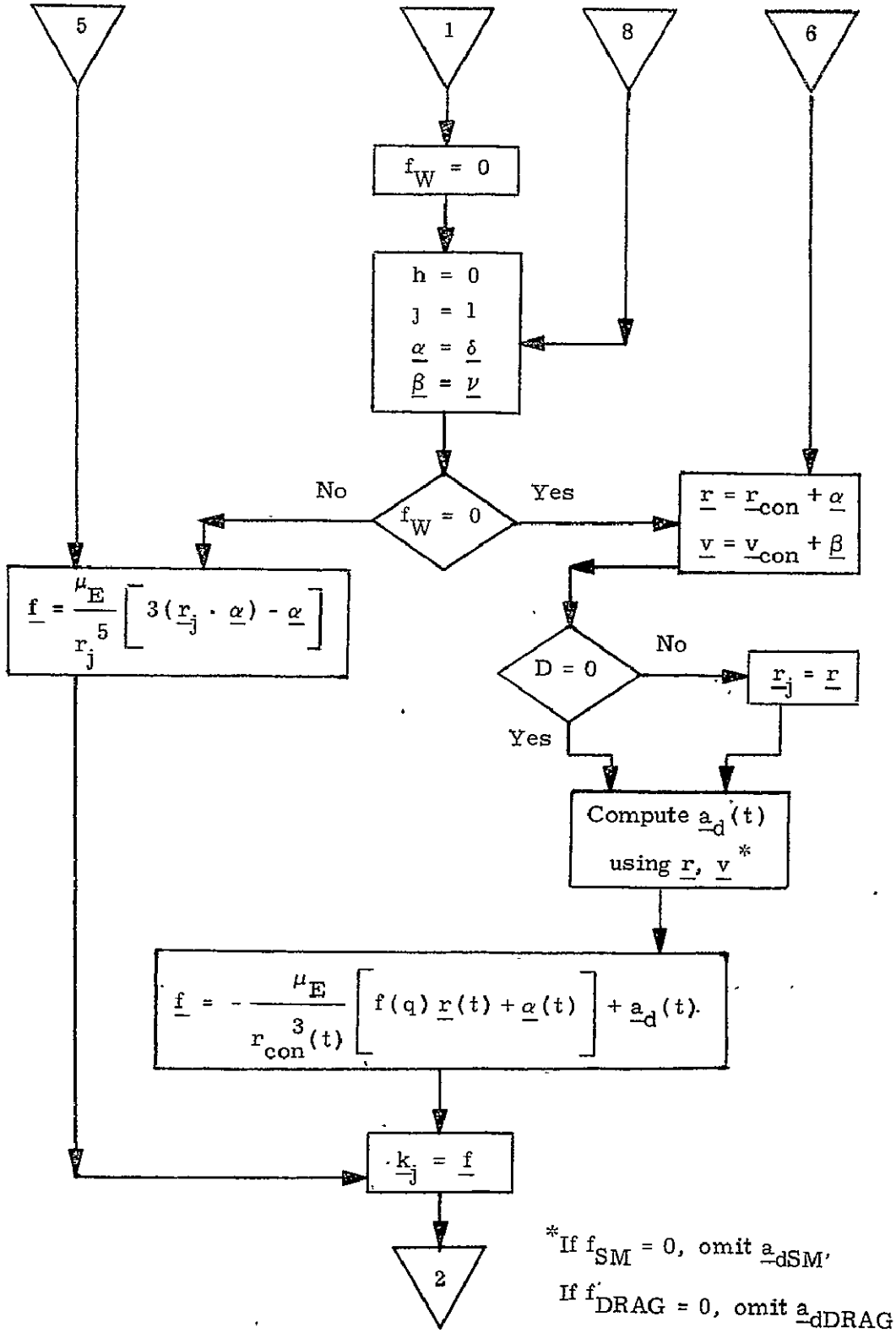


Figure 2b DETAILED FLOW DIAGRAM

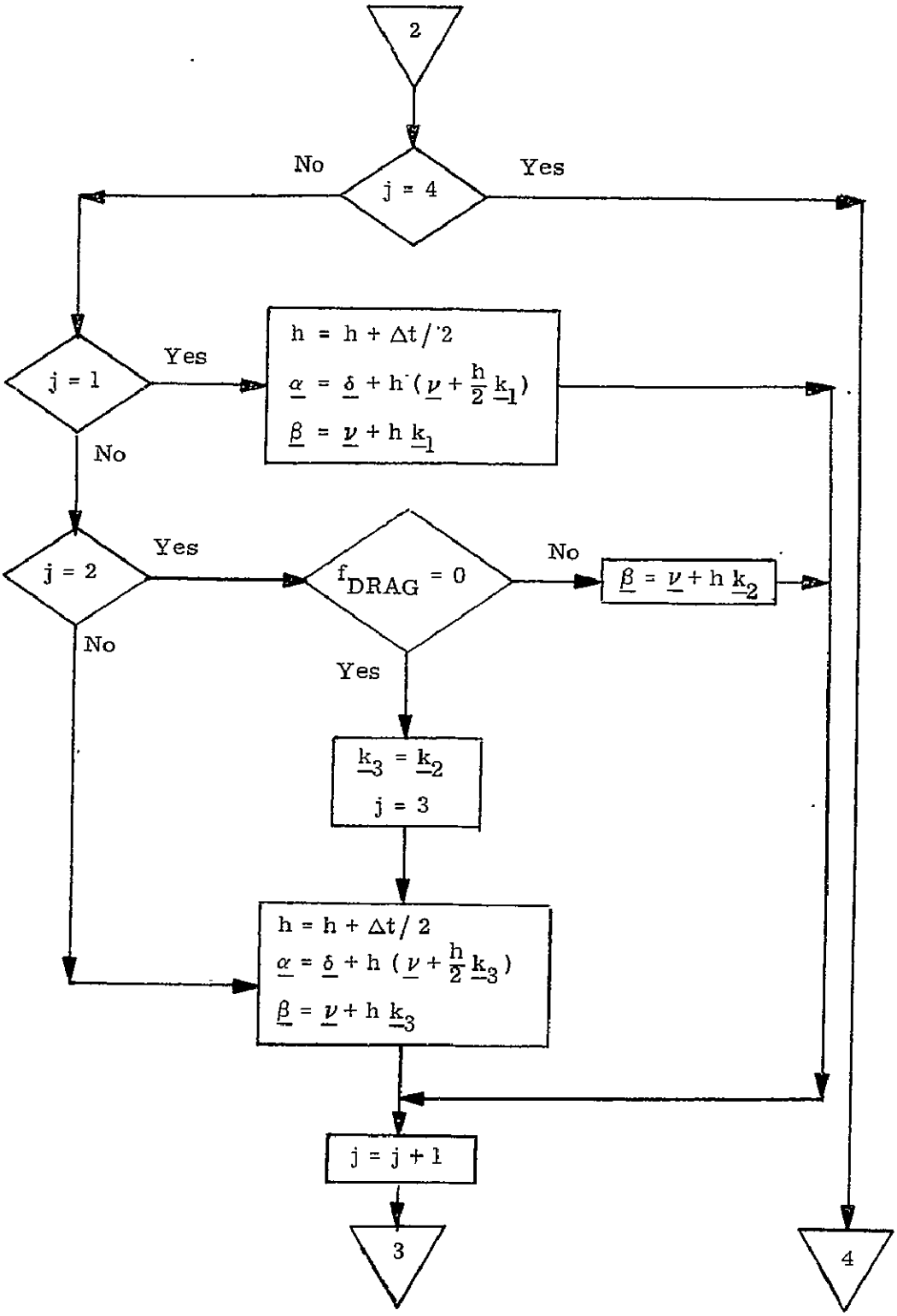


Figure 2c DETAILED FLOW DIAGRAM

9.6.1.2 Precision State and Filter Weighting Matrix Extrapolation (continued)

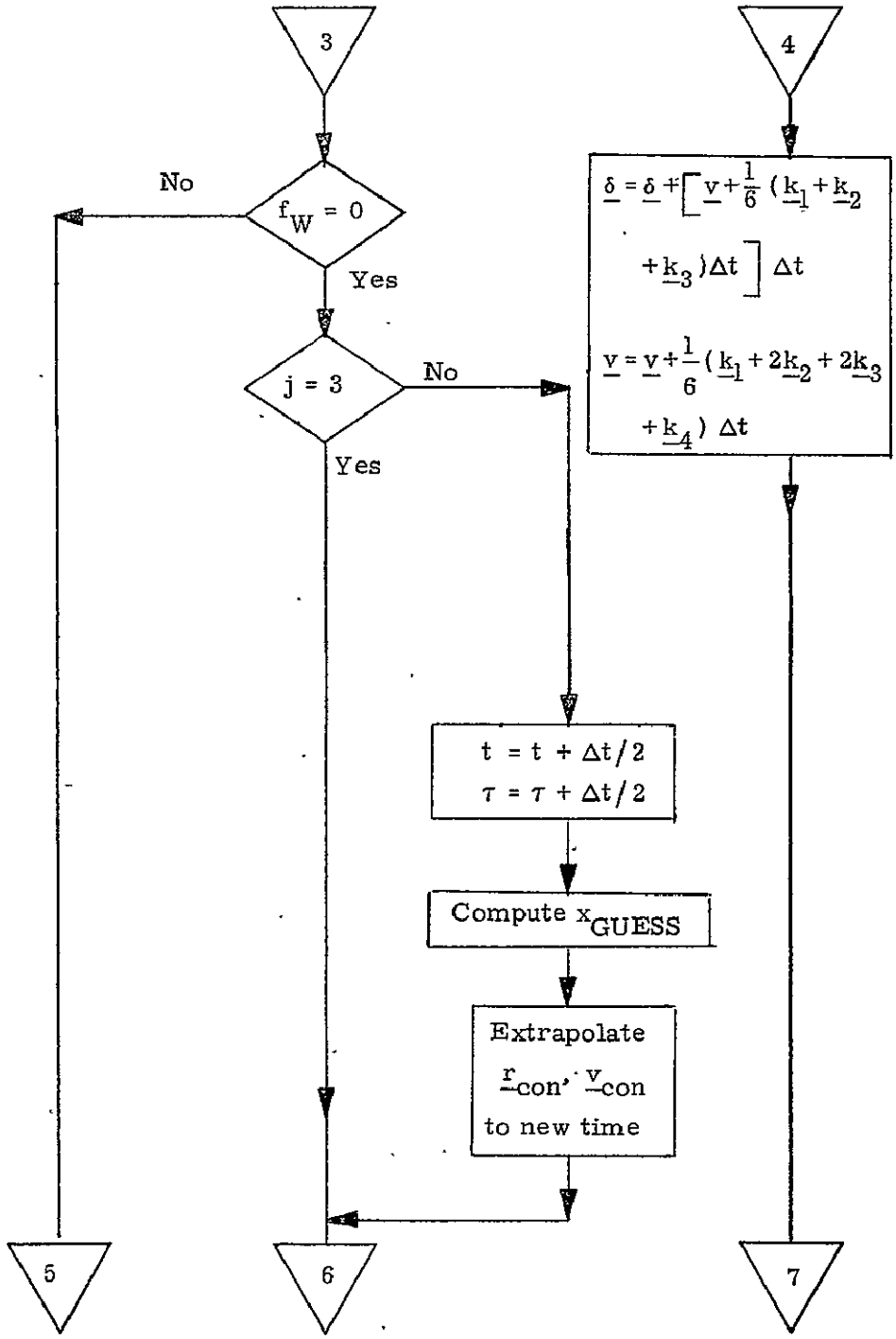


Figure 2d DETAILED FLOW DIAGRAM

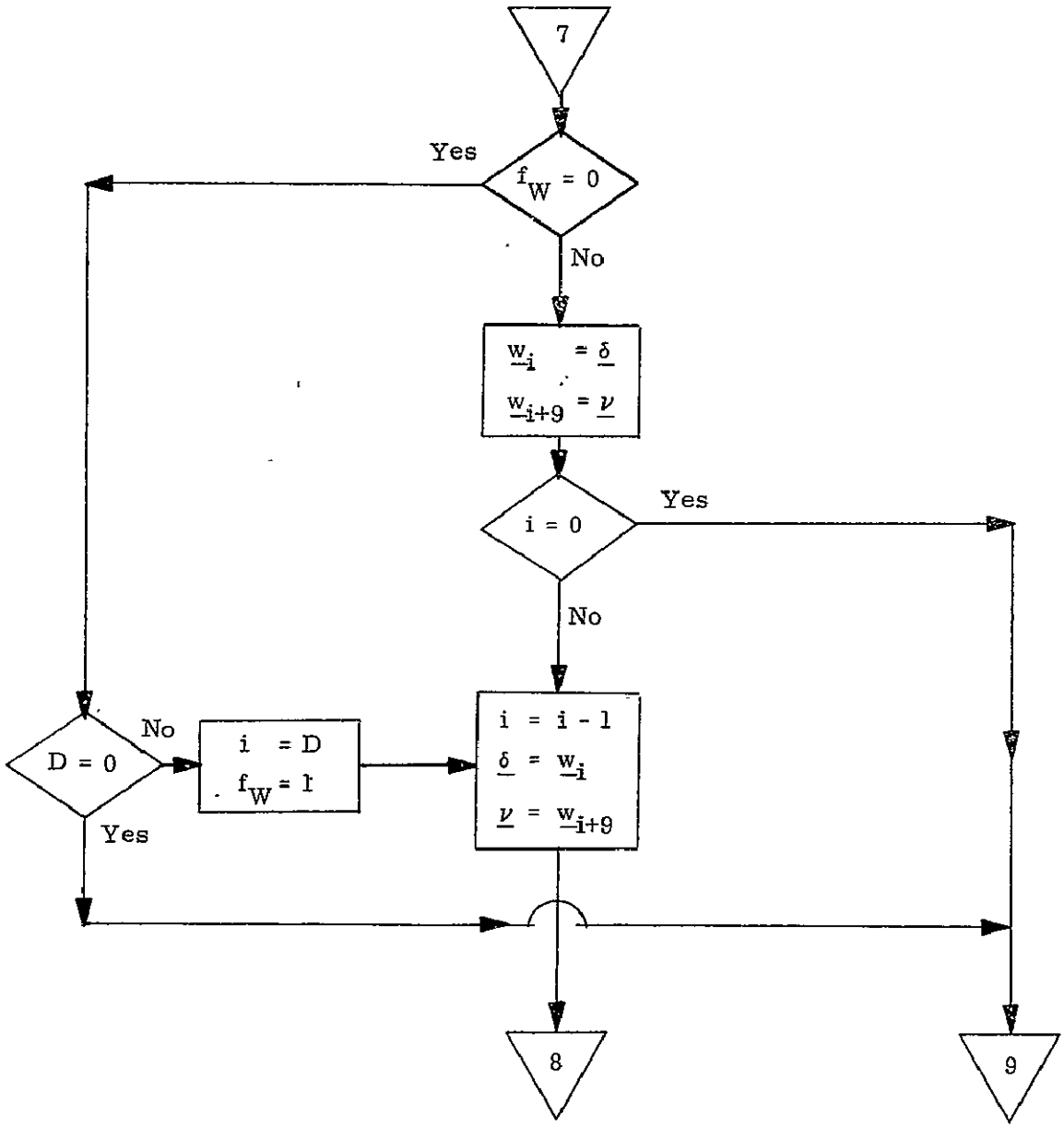


Figure 2e DETAILED FLOW DIAGRAM

6. SUPPLEMENTARY INFORMATION

Encke's technique is a classical method in astrodynamics and is described in all standard texts, for example Battin (1964). The f(q) function used in Encke's technique and in the lunar-solar perturbing acceleration computations has generally been evaluated by a power series expansion; the closed form expression given here was derived by Potter, and is described in Battin (1964).

The oblateness acceleration in terms of a spherical harmonic expansion may be calculated in a variety of ways; three different algorithms are given in Gulick (1970). For low order expansions, especially those involving mostly zonal terms, the formulation presented in Section 4 is generally superior computation-time-wise, as only the non-zero terms enter into the calculation. The general expression for the zonal terms is given by Battin (1964), while Zeldin and Robertson (1970) give explicit analytic expressions for each of the tesseral terms up through fifth order; hence additional terms may easily be included in the oblateness acceleration $\underline{a}_{dOBL}$ by consulting the formulations in these two references.

The lunar-solar perturbing acceleration formulation has also been described by Battin (1964) and many other texts.

The drag acceleration utilizes the standard fundamental expression in terms of the $C_D A/M$ constant, the density, and the velocity with respect to the atmosphere. The double binomial approximation to Jacchia's density model has been developed by the McDonnell-Douglas Corporation (1970); the model itself is described in Jacchia (1970). (A new model by Jacchia is about to appear).

A full discussion of the use of covariance matrices in space navigation is given in Battin (1964). Potter (1963) suggested the use of the W-matrix and developed several of its properties. It should be noted that strictly the gravity gradient matrix **G(t)** should also include the gradient of the perturbing acceleration; however, these terms are so small that they may be neglected for our purposes. The use of only the conic gravity gradient, however, does not imply the W-matrix is being extrapolated conically. (Conic extrapolation of the W-matrix

can be performed by premultiplying the W-matrix by the conic state transition matrix, which can be expressed in closed form). Rather the W-matrix is here extrapolated along the precision (perturbed) trajectory, as can be seen from the detailed flow diagram of Section 5.

The Nystrom numerical integration technique was first conceived by Nystrom (1925), and is described in all standard texts on the numerical integration of ordinary differential equations, such as Henrici (1962). Parametric studies carried out by Robertson (1970) on the general fourth-order Runge-Kutta and Nystrom integration techniques indicate that the "classic" techniques are the best overall techniques for a variety of earth orbiting trajectories in the sense of minimizing the terminal position error for all the trajectories, although for any one trajectory a special technique can generally be found which decreases the position error after ten steps by one or two orders of magnitude for only that trajectory. The classical fourth-order Runge-Kutta and Nystrom techniques are approximately equally accurate, but the latter possesses the computational advantage of requiring one less perturbing acceleration evaluation per step when the perturbing acceleration is independent of the velocity. This fact has been taken into account in the detailed flow diagram of Section 5, in that the extra evaluation is performed only when atmospheric drag is included. Some past Apollo experience has suggested that extra evaluation effect with drag is so small as to be negligible; further analysis will confirm or deny this for the Space Shuttle. In regard to step-size, the constants and the functional form of the nominal and maximum time-step expressions have been determined by Marscher (1965).

References

1. Battin, R.H., 1964, Astronautical Guidance, McGraw-Hill.

2. Gulick, L.J., 1970, A Comparison of Methods for Computing Gravitational Potential Derivatives, Environmental Science Services Administration Technical Report C&GS 40, Rockville Maryland, September 1970.

3. Henrici, P., 1962, Discrete Variable Methods in Ordinary Differential Equations, Wiley & Sons.

4. Jacchia, L.G., 1970, New Static Models of the Thermosphere and Exosphere with Empirical Temperature Profiles, Smithsonian Astrophysical Observatory Special Report #313, May 1970.

5. Marscher, W.F., 1965 , A Modified Encke, MIT/IL Space Guidance Analysis Memo #2-65, January 1965.

6. McDonnell-Douglas Corporation, 1970, Feasibility of Incorporating Aerodynamic Drag in the CMC Equations of Motion, Presented at the Skylab Rendezvous Mission Techniques Meeting on 8 October 1970.

7. Nystrom, E.J., 1925, Uber die Numerische Integration von Differentialgleichungen, Acta Soc. Sci., Fenn, 50 (1925), No. 13, 1-55.

8. Potter, J.E., 1963, New Statistical Formulas, MIT/IL Space Guidance Analysis Memo #40, April 1963.

9. Robertson, W.M., 1970, Parametric Investigations of the Free Parameters in Fourth-Order Runge-Kutta and Nystrom Integration Techniques, Supplement to MIT/DL Report # DLMC-70-3, June 1970.

(10) Zeldin, S. and Robertson, W. M., 1970, Explicit Analytic Expressions for the Terms in a Spherical Harmonic Expansion of the Gravitational Acceleration due to Oblateness, MIT/DL Space Guidance Analysis Memo #4-70, May 1970.

9.6.2 Sensor Calibration and Alignment (TBD)

9.6.3 Orbital Coast Attitude Control (TBD)

9.7 ORBITAL POWERED FLIGHT

The following GN&C software functions are envisioned for the orbital phase:

Guidance Functions

1. Perform orbit modifications targeting, or accept ground/base targeting solutions as initialization.
2. Compute and command initial thrusting attitude.
3. Command orbiter maneuvering engine on.
4. Compute velocity to be gained vector (before and during burn).
5. Provide orbit maneuver steering commands to autopilot.
6. Compute time-to-cut-off and issue engine off commands.
7. Provide commands to null residual velocities.

Navigation Functions

1. Specific Force Integration - Advance the inertial state utilizing accelerometer measurement of thrust and aerodynamic forces.
2. Update inertial state from other navigation sensor data if available. An example is radar altimeter data.
3. Provide coordinate transformations for state vectors as required.
4. Compare state with that calculated by other vehicle during launch for use in decision making and possible updating.

Control Functions

1. Perform vehicle stabilization and control during TVC by engine gimbal commands.
2. Provide vehicle roll stabilization during single-engine burns using RCS.
3. Perform attitude-hold RCS ΔV maneuvers.
4. Perform steered-attitude RCS ΔV maneuvers for docking if required.

5. Do cg/trim estimation during TVC burns

6. Make high-frequency steering estimates between guidance samples for docking if required.

7. Perform adaptive-loop gain calculation if required.

9.7.1 Targeting (TBD)

9.7.2 Navigation (same as Rapid Real-Time State Advancement During Specific Force Sensing)

9.7.3 Guidance (TBD)

9.7.4 Attitude Control (TBD)

9.8 RENDEZVOUS MISSION PHASE

The rendezvous mission phase begins at the completion of orbit insertion with the computation of the rendezvous plan. During this coasting period of time, the mission planning software will utilize the rendezvous targeting routines to insure that the insertion cut-off conditions (i.e., cut-off state vector) are within the rendezvous corridor defined by pre-mission and crew option inputs. When a satisfactory plan is established, the various tasks of the rendezvous will be assigned a preliminary schedule. The implementation of this plan will represent the remainder of the rendezvous mission phase.

The SW functions required in this mission phase are the following:

1. Estimate relative state of target vehicle based on external measurements (if available).

2. Estimate absolute states of both shuttle and target vehicle.

3. Target the rendezvous ΔV's required, their direction, and the time's of ignition.

4. Execute rendezvous maneuvers by commanding engine's on, providing attitude commands during the maneuvers, and commanding engines off.

5. Powered flight navigation.

6. Provide RCS engine commands to achieve commanded attitude during ΔV maneuvers and during coast periods (digital autopilot).

7. Provide data for failure analysis.

8. Provide data for crew display.

The guidance and navigation software during the burns will be the same as described in Orbital Powered Flight. The estimates of absolute states will be performed as described in Orbital Coast.

SPACE SHUTTLE

GN&C SOFTWARE EQUATION SUBMITTAL

Software Equation Section: Rendezvous Targeting Submittal No. 21

Function: Provide targeting solutions for rendezvous

Module No. OG3 Function No.: 1,2,3,4,5,6 (MSC 03690)

Submitted by: W. H. Tempelman, P. M. Kachman Co. MIT No. 7-71

Date: Feb 1971

NASA Contact: J. Suddath Organization: GCD

Approved by Panel III K. J. Cox Date: 3/10/71

Summary Description: Rendezvous targeting is concerned with selecting the number of maneuvers, their spacing, magnitudes and directions. These maneuvers and the resulting trajectory must satisfy lighting, fuel, communication and navigation constraints and have terminal conditions conducive to crew monitoring and backup procedures.

Shuttle Configuration: This software is essentially independent of the shuttle configuration.

Comments:

(Design Status)

(Verification Status)

Panel Comments:

9.8.1 Rendezvous Targeting

1. INTRODUCTION

The rendezvous of the Shuttle with the Space Station is accomplished by maneuvering the Shuttle into a trajectory which intercepts the Space Station orbit at a time which results in the rendezvous of the two vehicles. The Shuttle maneuvers by intermittent thrusting of its rocket engines. Rendezvous targeting is concerned with selecting the number of maneuvers and their spacing, magnitudes and directions. These maneuvers and the resulting trajectory configuration must satisfy lighting, communication and navigation constraints and have a terminal configuration conducive to astronaut monitoring and backup maneuver procedures. In addition, the rendezvous should not use excessive amounts of fuel.

1.1 Possible Maneuver Sequences

During the Gemini and Apollo flights and in the design of the Skylab rendezvous trajectories various numbers of maneuvers were involved in making the rendezvous. The range went from two (Apollo 14) to six (Skylab). Rendezvous targeting is primarily concerned with the targeting of the maneuvers which proceed the two terminal maneuvers, which are targeted separately. The number of preterminal maneuvers involved in a rendezvous presents one way of categorizing the rendezvous scheme. For instance, one possible rendezvous nomenclature is

Preterminal Rendezvous Sequences

ZMS - Zero maneuver sequence

SMS - Single maneuver sequence

DMS - Double maneuver sequence (eg, Apollo: CSI, CDH)

TMS - Triple (Three) maneuver sequence

FMS - Four maneuver sequence (eg, Skylab: NC1, NC2, NCC, NSR)

It would be premature to select one of the above sequences as involving an optimum number of maneuvers for the Shuttle rendezvous at this time as the desired targeting constraints (which are a function of the constraints mentioned above) have not yet been formulated. This report, therefore, will emphasize certain basic maneuvers to be used as building blocks for the Shuttle rendezvous scheme employing N maneuvers.

As each maneuver has its own targeting program, the above nomenclature can be applied to all of the above preterminal maneuvers. For instance, if the first maneuver performed is a four maneuver sequence (FMS), it will be chronologically followed by a TMS, DMS and a SMS. The next maneuver will then be the first maneuver of the terminal phase of the rendezvous.

1.2 Number of Targeting Constraints Involved in the Preterminal Phase of Rendezvous

The number of constraints involved in the preterminal rendezvous sequences equals the number of degrees of freedom implicitly contained in the sequence. To establish this number, a rendezvous configuration can be constructed by imposing arbitrary constraints until the configuration is uniquely defined. A four maneuver coplanar sequence is shown in Fig. 1 , followed by a coast to a terminal point. Using the constraints v_i (velocity magnitude), r_i and θ_i, it is easy to establish that the total number involved is 12, assuming the time of the first maneuver has been established. Removing one maneuver will reduce the number of degrees of freedom by three. Hence, the number of constraints necessary to uniquely determine the maneuver sequences are

ZMS - 3
SMS - 3
DMS - 6
TMS - 9
FMS - 12

If the above rendezvous are not coplanar rendezvous, one additional constraint has to be added to each sequence to allow for the out-of-plane component.

If the desired number of targeting constraints falls short of the number of constraints required for a particular rendezvous sequence then the rendezvous trajectory is not uniquely defined and it cannot be determined. The deficiency in constraints can be eliminated by introducing sufficient variables to complete the determination of the rendezvous trajectory and then determining values for these variables by optimizing the fuel used by varying the variables. This procedure, although not used on previous missions, might be desirable for the Shuttle.

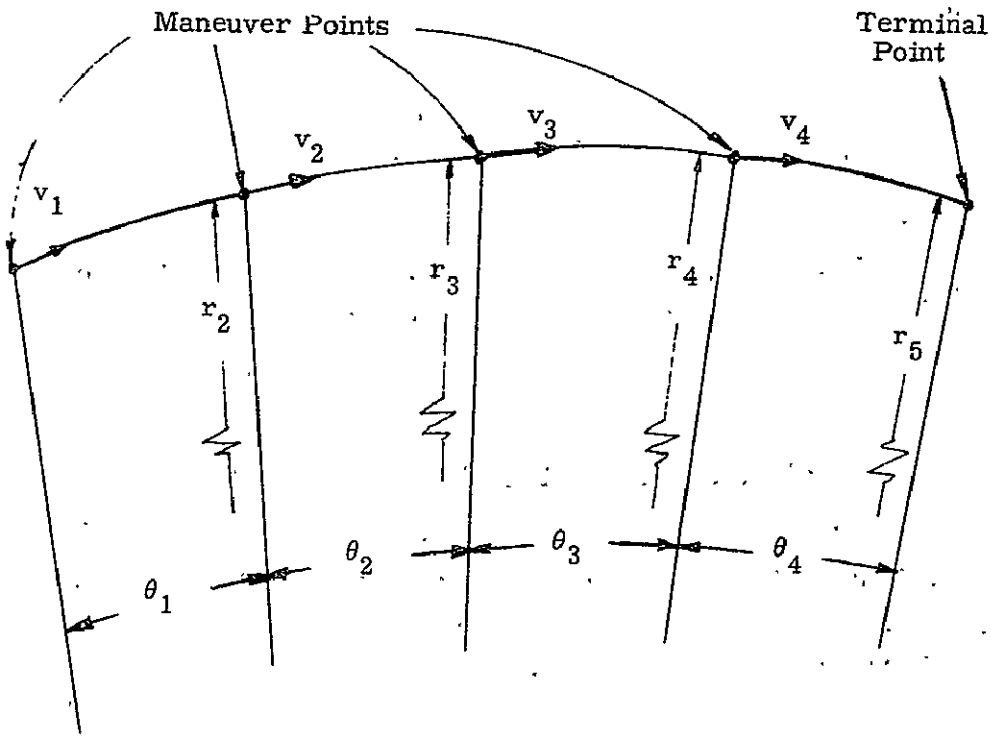


Figure 1 A Possible Set of Constraints Involved in a FMS Rendezvous

1.3 Basic Preterminal Maneuvers

Each maneuver in the above sequences can have associated with it targeting constraints governing the maneuver itself, the duration of the coast phase following the maneuver and trajectory constraints defined either with respect to the active vehicle or relative to the passive vehicle. The possible constraints are multifarious. However, Apollo and Skylab utilized three basic maneuvers, each with a set of unique constraints.

NP - Nominal phase maneuver
NH - Nominal height maneuver
NC - Nominal coelliptic maneuver
NL - Nominal Lamber Maneuver

In addition, many other types of maneuvers could be defined, depending on the targeting constraints imposed.

1.3.1. Nominal Phasing Maneuver

The nominal phase maneuver is a horizontal maneuver which adjusts the active vehicle's orbit to satisfy a terminal constraint on phasing. This phasing constraint is generally imposed at the TPI point (see below), using the desired relative geometry between the two vehicles. Also assoicated with this maneuver is a constraint on the elasped time before the next maneuver. In the Apollo concentric flight plan NP was the first maneuver in the rendezvous sequence and was referred to as the CSI maneuver. In the Skylab NC1 and NC2 programs it was also the first maneuver and was referred to as the NC1 and NC2 maneuvers, respectively.

1.3.2. Nominal Height Maneuver

The nominal height maneuver is a horizontal maneuver whose purpose is to achieve a specified differential altitude with the passive vehicle orbit at the time of the following maneuver, which also has a constraint imposed on its timing. Generally, the latter constraint consists of the fraction of a revolution which elapses between the two maneuvers. The NC2 and NCC maneuvers during the

Skylab NC1 program were height maneuvers and the NCC maneuver was a height maneuver during the NC2 program.

1.3.3. Nominal Coelliptic Maneuver

The coelliptic maneuver computes a post maneuver velocity which results in coelliptic (concentric) orbits, whereby the two orbits maintain an approximately constant, radial separation distance. This maneuver generally preceeds the TPI maneuver. In Apollo and Skylab, CDH and NSR were the coelliptic maneuvers, respectively.

1.3.4. Nominal Lambert Maneuver

The Lambert maneuver computes a maneuver based on computing a trajectory between two position vectors with a constraint on the traverse time. In Apollo and Skylab, TPI and NCC were Lambert maneuvers.

1.4 Terminal Maneuvers

The terminal phase of the rendezvous includes three maneuvers:

TPI - Terminal phase initiation
TPM - Terminal phase midcourse
TPF - Terminal phase final

These maneuvers result in the rendezvous of the active vehicle with the passive vehicle. TPF will be covered by documentation to be submitted at a latter date and is not herein considered.

1.4.1 Terminal Phase Initiation Maneuver

The time of this maneuver is determined either by assuming a time or by specifying an elevation angle. This angle allows the maneuver time to be determined in an iterative search for the vehicle's relative positions which satisfies the elevation angle constraint.

The TPI maneuver determines a trajectory which results in a rendezvous with the passive vehicle. This maneuver is determined with a precision Lambert routine.

1.4.2 Terminal Phase Midcourse Maneuver

This program computes a midcourse maneuver to insure that the active vehicle will intercept the passive vehicle at the time estab-

lished in the TPI program. The precision Lambert routine is used in this program as in the TPI program.

1.5 Additional Maneuvers and Programs

This section contains a discussion of the plane change (PC) maneuver and the phase match (PM) program.

1.5.1. Plane Change Maneuver

The purpose of the plane change maneuver is to make the active and passive vehicle orbits coplanar. This maneuver is made by the active vehicle following a 90 degree central angle traverse from the previous maneuver, which must result in the active vehicle's velocity vector being coplanar with the orbital plane of the passive vehicle. This produces a nodal point 90 degrees from the maneuver.

1.5.2. Phase Match Program

The correct rendezvous configuration would be derived if the targeting programs used precision extrapolation programs exclusively. However, as the targeting programs generally employ iterative searches, the use of precision extrapolation routines will generally be rather costly from the time standpoint. For coplanar problems, the "phase match" technique allows the use of conic extrapolation routines without significantly degrading the accuracy of the calculation. This technique results in updating both vehicle's states over the same sector of the earth's surface during the targeting program. As the vehicles are in approximately the same orbits, the effects of the oblate earth will be the same in each orbit, resulting in an approximately unperturbed relative motion.

1.6 Organization

The remainder of this document is divided into two main sections, 1) an individual description of each of the above maneuvers and 2) an assembly of some of these maneuvers into a four maneuver preterminal rendezvous requence. In addition there is a section containing some basic subroutines used in several of the programs.

1.7 Nomenclature

The nomenclature below is a listing of the major symbols used in this report. It is not a complete listing due to the large number of basically independent programs contained in this report.

c	Iteration counter (used in Newton Raphson iterative loop)
DELH	Delta altitude as defined in TPI program
dv	Maneuver magnitude
e, e_L, E	Elevation angle (different symbols apply in different programs)
h	Height
n	Number of revolutions between maneuvers
p	Partial (used in Newton Raphson iteration loop)
r	Position
RNGE	Distance along orbit as defined in TPI program
s	Switch
t	Time
v	Velocity
X, Y, Z	Cartesian axes
ϵ	Iterative cutoff limit on the dependent variable.
τ	Orbital period

Subscripts

A	Active
D	Desired
F	Final
H	Height maneuver
P	Passive, phasing maneuver

The function SIGN (x) (in the TPI program this function is SGN) equals 1 if x is zero or a positive number; otherwise equals -1.

2. MANEUVER PROGRAM DESCRIPTIONS

The descriptions of the maneuver program are divided into five sections each

1) Program objectives and general comments
2) Program input - outputs
3) Program description
4) Program functional flow chart
5) Program detailed flow chart

The program descriptions contain equations only when they are of special significance. In general, few equations are listed as the maneuver programs mostly contain statements pertaining to state vector extrapolations or trivial equations.

The functional flow charts seek to describe with words exactly what the detailed flow charts describe with mathematical symbols. The two flow charts are, thus, both self sufficient (on their respective levels) and self explanatory.

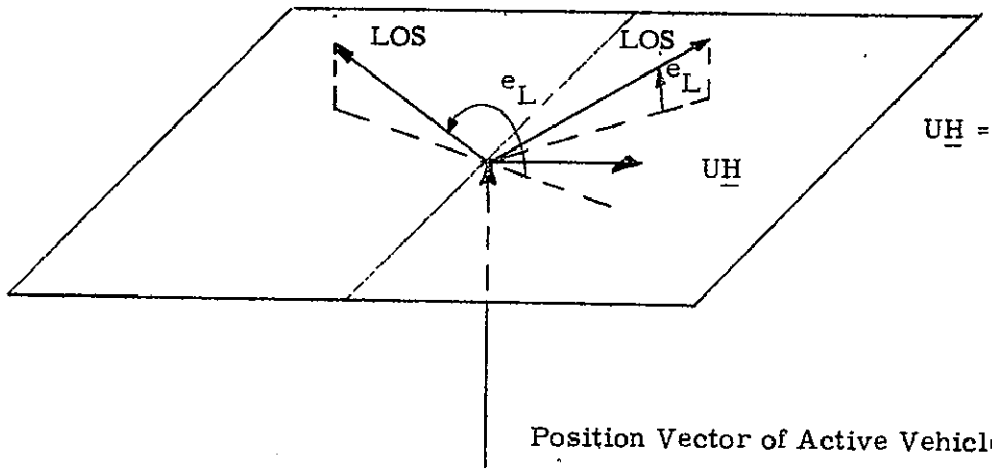
2.1 Phase Maneuver Program (NP)

2.1.1. NP Program Objectives

The main objective of the Phase Maneuver NP Program is to compute a maneuver which will satsify a terminal phasing constraint. This constraint is generally imposed at the TPI point by specifying the desired relative geometry of the two vehicles. The geometry is usually defined in terms of the elevation angle (see Fig. 2) and a delta altitude between the passive orbit and the active vehicle at the TPI time. In addition, this maneuver must satisfy the constraints

1. $\Delta \underline{v}_p$ horizontal
2. Next maneuver occurs after transfer time $n\tau$, where n is specified and τ is the post NP orbital period.

Additional maneuvers, which can be independently calculated, can be inserted between the NP and TPI maneuvers.



1) If the LOS Projection on 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 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 2 Definition of Elevation Angle, e_L

The calculation of the NP maneuver involves a Newton Raphson interation scheme to determine the magnitude of the maneuver, using a phasing error as the dependent variable. A functional flow chart of the NP maneuver is contained in Fig . 2.1.4.

2.1.2 NP Program Input-Output

The required inputs to the Phase Maneuver Program are

$\underline{r}_A$, $\underline{v}_A$, $\underline{r}_P$, $\underline{v}_P$	Active and passive state vectors at time of NP maneuver
n	Number of revolutions before next maneuver
Δh	Desired relative altitude at TPI (positive if passive above active)
e	Elevation angle (see Fig. 2)
v	Initial guess of NP maneuver magnitude
t_{TPI}	Time of the TPI maneuver

The outputs to this program are

$\Delta\underline{v}_P$	NP maneuver $\Delta\underline{v}$
t	Time of next maneuver

2.1.3. NP Program Description

A detailed flow chart of the program is shown in Fig. 2.1.5.

The program starts with an update of the passive state to the TPI time. The QRDTPI routine is then used to obtain the active vehicle's position vector $\underline{r}_{AD}$ at the TPI time. This vector defines the desired TPI phasing.

After adding the guess of the NP maneuver to the active vehicle's velocity vector, the REVUP subroutine is used to obtain the active state at the next maneuver point. This state can then be updated through additional maneuvers as required, so long as the maneuvers are self contained, until the TPI time is reached. An

Phase Maneuver

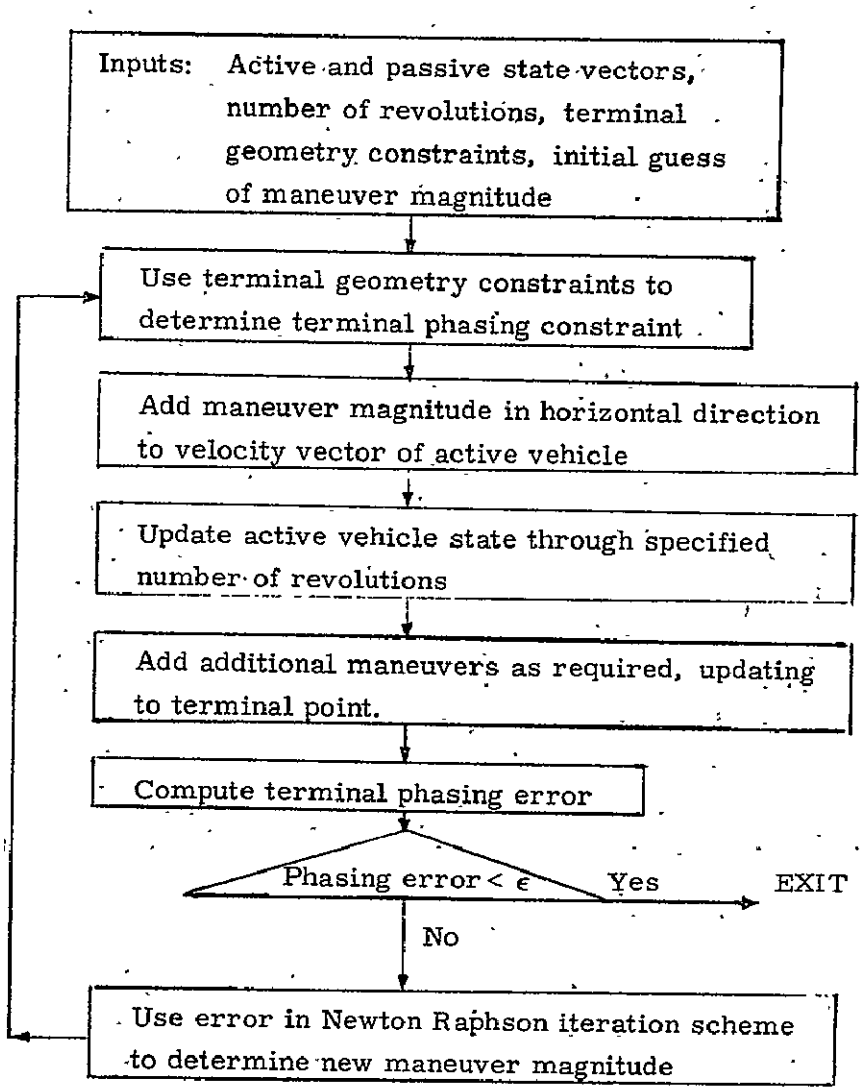


Fig. 2.1.4 NP Functional Flow Chart

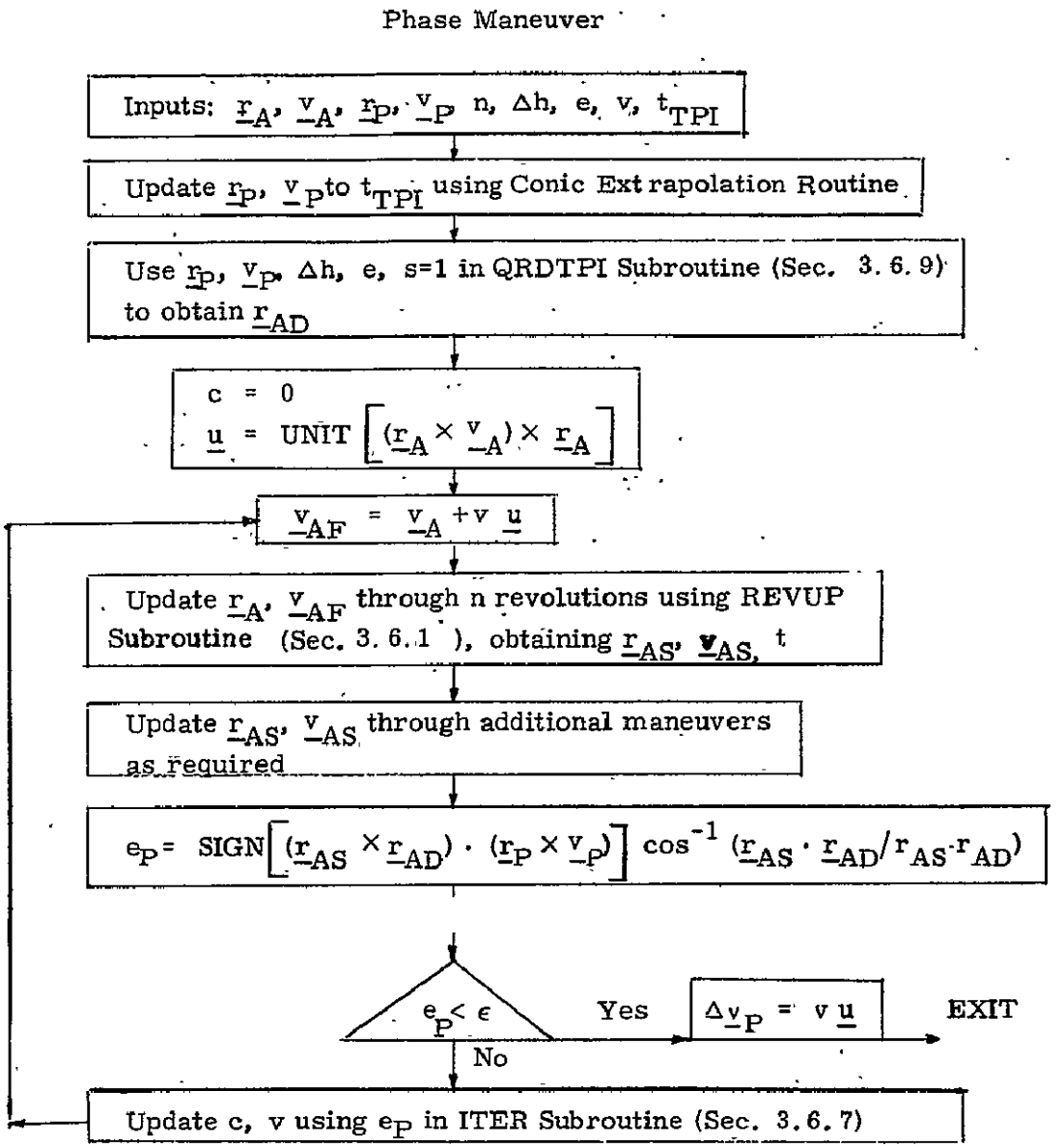


Fig. 2.1.5 NP Detailed Flow Chart

angular phasing error is then defined as

$$e = \text{SIGN}\left[(\underline{r}_{AS} \times \underline{r}_{AD}) \cdot (\underline{r}_P \times \underline{v}_P)\right] \cos^{-1}(\underline{r}_{AS} \cdot \underline{r}_{AD} / r_{AS}\, r_{AD})$$

where $\underline{r}_{AS}$ is the active vehicle's position at the TPI time. This error is then used in the ITER subroutine to vary the phasing maneuver until the phasing constraint is satisfied.

2.2 Height Maneuver Program (NH)

2.2.1 NH Program Objectives

The objective of the height maneuver NH program is to compute a maneuver which satisfies the following constraints

1. $\Delta\underline{v}_H$ horizontal
2. Next maneuver occurs after transfer time $n\tau$, where n is specified and τ is the post NH orbital period.
3. Attitude between passive orbit and active vehicle is specified.

Generally, the third constraint is assumed to occur at the maneuver following the NH maneuver. However, a self contained maneuver and coast phase can be inserted into this calculation such that the third constraint applies at the second maneuver point following the NH maneuver.

The calculation of the NH maneuver involves a Newton Raphson iteration scheme to determine the magnitude of the maneuver, using an error in altitude as the dependent variable. A functional flow chart of the NH maneuver is contained in Fig. 2.2.4.

2.2.2 NH Program Input-Output

The required inputs to the Height Maneuver Program are

$\underline{r}_A, \underline{v}_A, \underline{r}_P \underline{v}_P$	Active and passive state vectors at time of NH maneuver
n	Number of revolutions before mext maneuver

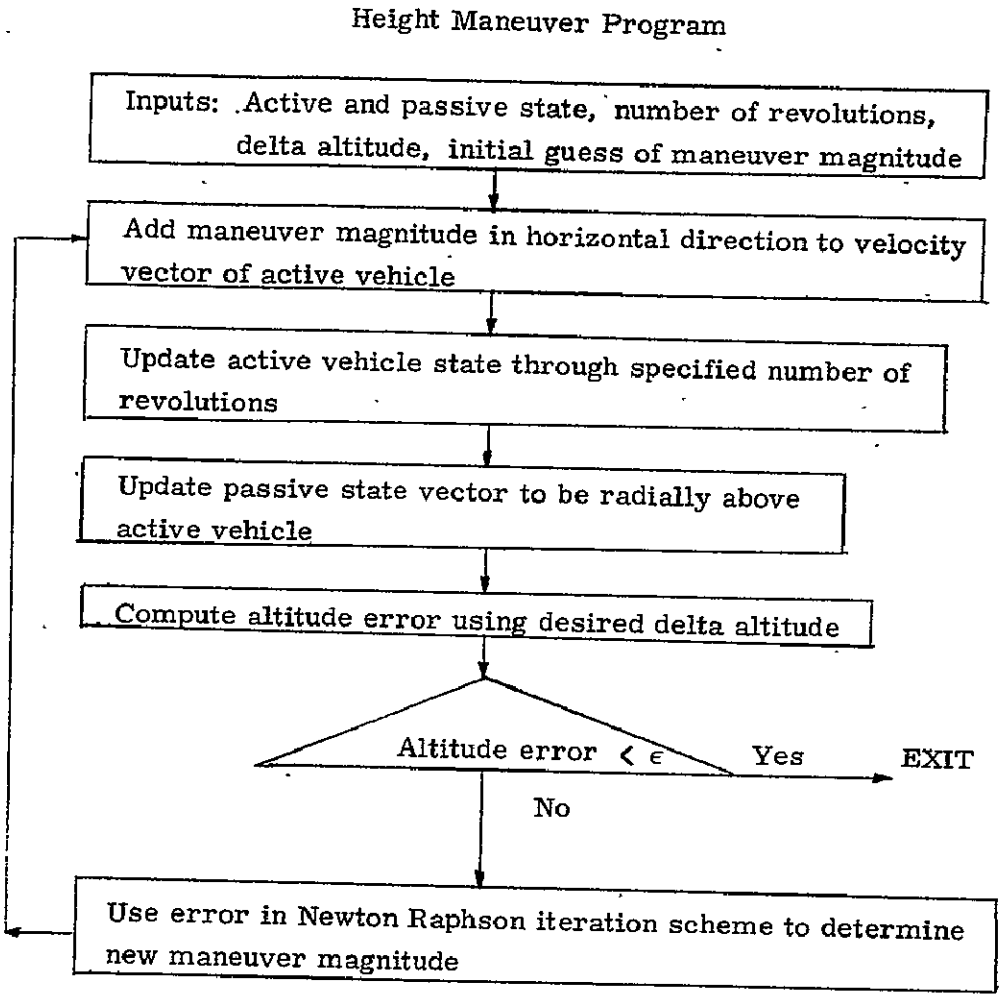


Fig. 2.2.4 NH Functional Flow Chart

Δh	Delta Altitude
v	Initial guess of NH maneuver magnitude

The outputs to this program are

$\Delta \underline{v}_H$	NH maneuver $\Delta \underline{v}$
t	Time of the next maneuver
$\underline{r}_{AS}$, $\underline{v}_{AS}$	Active state at the next maneuver point

2.2.3 NH Program Description

A detailed flow chart of the program is shown in Fig. 2.2.5.

The program starts with the addition of the guess of the NH maneuver to the active vehicle velocity vector. The active state is then updated through n revolutions using the REVUP subroutine. Following the updating of the passive state to be radially above the active position vector using the RADUP subroutine, the height error is computed to be

$$e = r_{PS} - r_{AS} - \Delta h$$

representing the error in attaining the desired altitude difference. This error is driven smaller than ϵ using the Newton Raphson iteration scheme contained in the ITER subroutine by varying the magnitude of the NH maneuver.

2.3 Coelliptic Maneuver Program (NC)

2.3.1. NC Program Objectives

The objective of the coelliptic maneuver NC program is to compute a maneuver which will result in the passive and active orbits being coelliptic following the maneuver. Coelliptic orbits have an approximately constant radial separation distance.

Fig. 2.3.4 is a functional flow chart of the NC program. The post maneuver velocity, which determines the active orbit, is a function of the passive state radially above the active vehicle at the NC time.

Height Maneuver Program

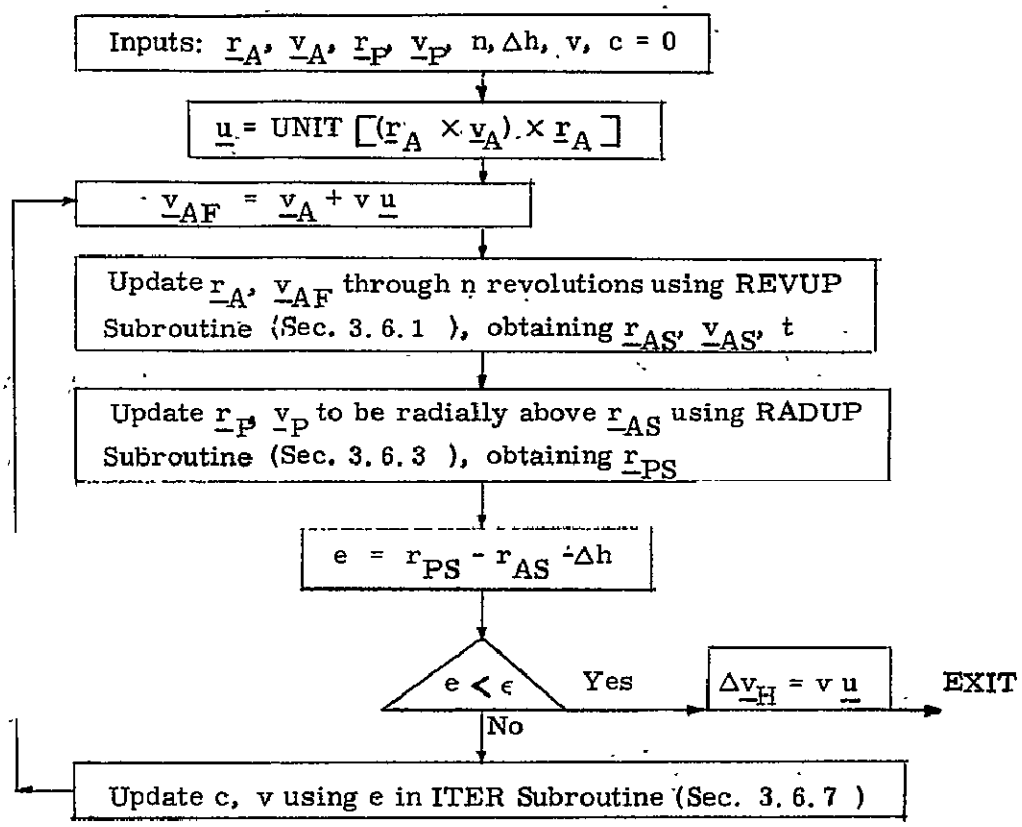


Fig. 2.2.5 NH Detailed Flow Chart

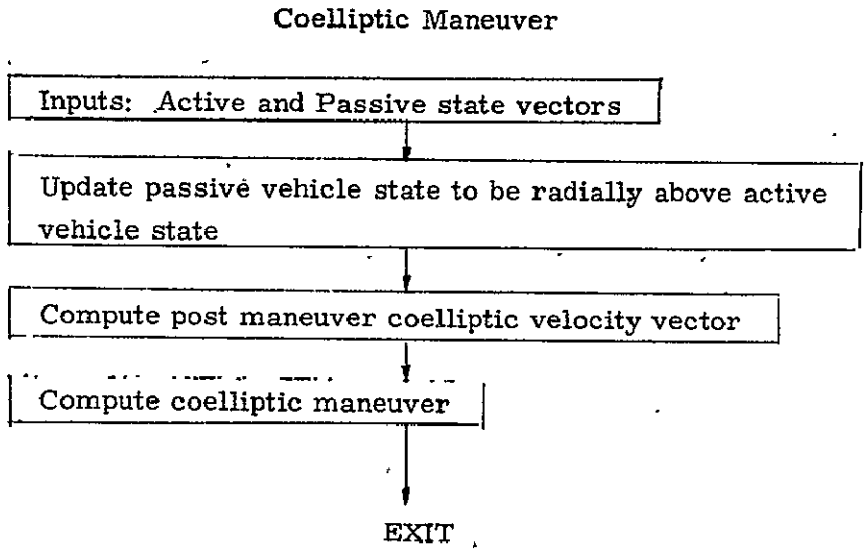


Fig. 2.3.4 NC Functional Flow Chart

2.3.2 NC Program Input-Output

The required inputs to the Coelliptic Maneuver Program are

$\underline{r}_A$, $\underline{v}_A$, $\underline{r}_P$ $\underline{v}_P$ Active and passive state vectors at time of NC maneuver

The outputs to this program are

$\Delta\underline{v}_C$ NC maneuver $\Delta\underline{v}$

2.3.3 NC Program Description

A detailed flow chart of the program is shown in Fig. 2.3.5

The program starts with an update of the passive state to a point radially above the active position vector using the RADUP subroutine. After obtain the altitude difference between the two position vectors, the COE subroutine is used to obtain the post maneuver velocity vector of the active vehicle. This is then used to obtain the NC maneuver.

2.4 Lambert Maneuver Program (NL)

2.4.1 NL Program Objectives

The main objective of the Lambert Maneuver Program is to compute a maneuver that will result in a trajectory which will pass through a specified target position vector at a given time. The target vector can be established by either a precision or conic update of the passive vehicle. The Lambert maneuver can also be executed either in an oblate or two-body gravity field. A functional flow chart of this maneuver is shown in Fig. 2.4.4.

2.4.2 NL Program Input - Outputs

The required inputs to the Lambert Maneuver Program are:

$\underline{r}_A$, $\underline{v}_A$, $\underline{r}_P$, $\underline{v}_P$ Action and passive state vectors at time of NL maneuver

Time associated with the target vector

Coelliptic Maneuver

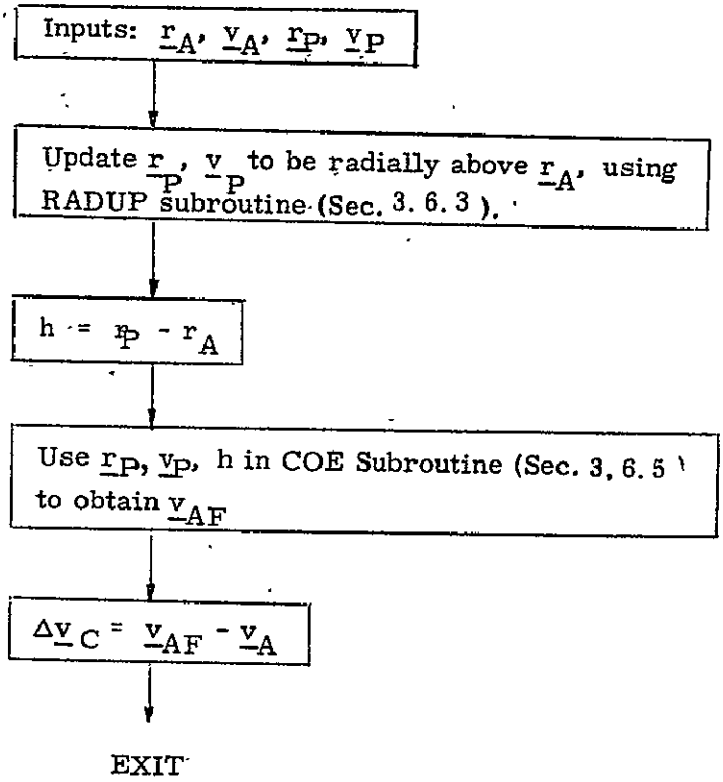


Fig. 2.3.5 NC Detailed Flow Chart

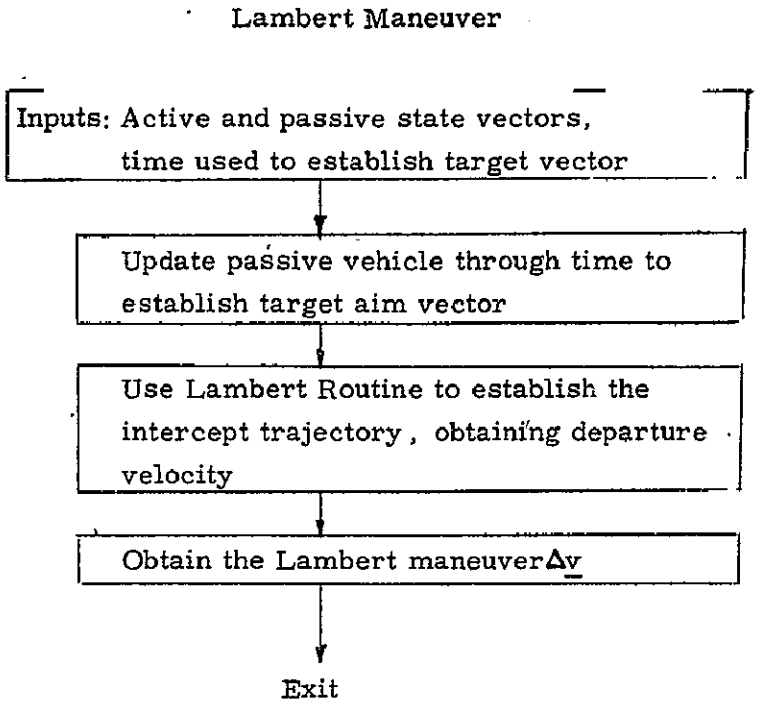


Fig. 2.4.4 NL Functional (Flow Chart)

The outputs to this program are:

$\Delta \underline{v}_L$ NL maneuver $\Delta \underline{v}$

2.4.3 NL Program Description

A detailed flow chart of the program is shown in Fig. 2.4.5.

The program starts with an update of the passive state to the target time to establish the target vector. This vector is then used in a Lambert routine to obtain the post NL maneuver velocity, which determines the NL maneuver.

2.5 Terminal Phase Initiation Program (TPI)

2.5.1 TPI Program Objectives

The objective of the Terminal Phase Initiation (TPI) program is to compute the TPI maneuver and the associated target vector required to initiate the terminal phase of the rendezvous sequence. The functional flow is illustrated in Fig. 2.5.4.

The position of the TPI maneuver is determined by specifying either the time of the maneuver or the elevation angle which specifies the relative geometry of the vehicles at the TPI point. If the elevation angle is specified an iterative procedure is initiated to find the TPI time at which this desired angle is attained.

The TPI maneuver can satisfy either of two objectives. It can place the active vehicle on an intercept trajectory with the passive vehicle or it can cause the active vehicle to pass through a target vector which has a particular relative geometry with respect to the passive vehicle at the intercept time.

If a coplanar intercept is desired, i.e. the active and passive vehicles orbital planes aligned after TPI and before the final intercept of the target vector, the computed TPI maneuver is modified by nulling the active vehicle's out-of-plane velocity relative to the passive vehicle's orbital plane. This causes the orbital planes to intersect ninety degrees from the active vehicle TPI position vector. At the time corresponding to this nodal crossing a mid-course cor-

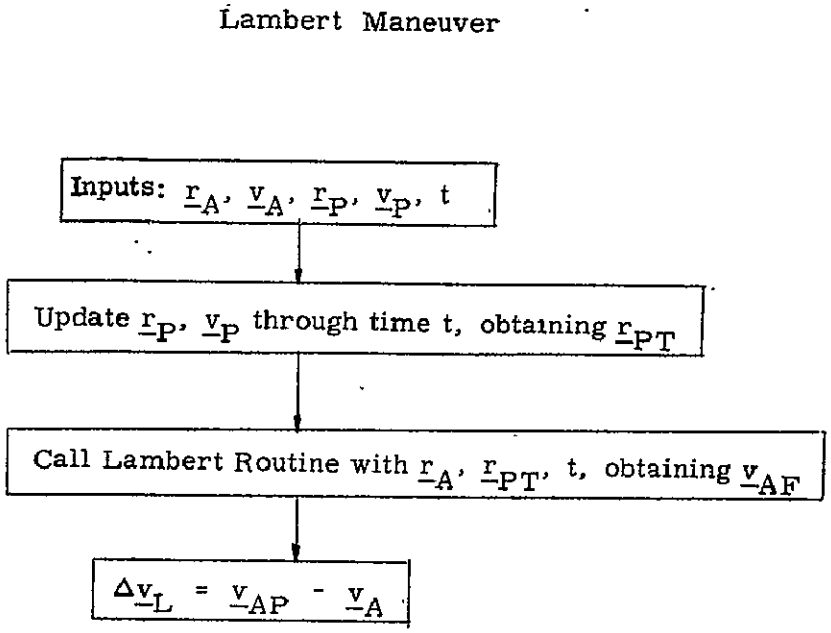


Fig. 2.4.5 NL Detailed Flow Chart

rection is performed which aligns the two orbital planes. The trajectories then remain coplanar throughout the rest of the terminal phase.

The differences between this TPI program and that used in Apollo are: target offsets are used automatically in computing the TPI maneuver to account for oblateness affects; all updates are done with precision extrapolations; modification of the TPI target vector using range and altitude offsets; and the addition of a coplanar intercept mode.

2.5.2 TPI Program Input-Outputs

The required inputs to the TPI program are:

$\underline{r}_{AI}$, $\underline{v}_{AI}$	-	Active vehicle state vector
$\underline{r}_{PI}$, $\underline{v}_{PI}$	-	Passive vehicle state vector
t_{TPI}	-	TPI ignition time
E	-	Elevation angle of passive vehicle from the active vehicle at TPI
ωt	-	Central angle of travel of the passive vehicle, from its TPI position to its final position (TPF) yielding the TPI target vector. Also determines the transfer time, t_F.
RNGE	-	Distance along the passive vehicle orbit from its position at TPF which defines a new target vector.
DELH	-	Altitude above or below the passive vehicle orbit, measured along the target vector at TPF, which defines a new target vector
CPI	-	Flag which determines if the coplanar intercept mode is to be used from TPI to TPF

The outputs from this program are:

$\Delta\underline{v}_{TPI}$	-	TPI deltav in Inertial Reference Coordinates
$\Delta\underline{v}_{TPI}(LV)$	-	TPI deltav in active vehicle local vertical coordinates
$\Delta\underline{v}_{TPI}(LOS)$	-	TPI deltav in active vehicle line-of-sight coordinates
$\underline{r}_{PT}$	-	Target vector used in the powered flight routine for the TPI maneuver

2.5.3 TPI Program Description

The TPI program is entered during the rendezvous targeting sequence when the terminal phase of the rendezvous is to be initiated.

The on-board estimated states of the active and passive vehicles are precision updated to the input TPI time. If the elevation angle has been input as zero, the TPI maneuver is executed at this time.

If, however, the elevation angle, E, at TPI is specified, an iterative procedure is initiated to find the TPI time at which this E occurs. This procedure uses the input TPI time as the initial guess. The input TPI time is changed by a time correction δt which is based on: (1) the angular distance between the passive vehicle position and the desired position of the passive vehicle obtained by assuming the target vehicle is in a circular orbit and (2) assuming the vehicles are moving at a constant angular rate. The following equation for δt is used:

$$\delta t = \frac{\alpha - \pi + SGN(r_P - r_A)\left[\pi - \cos^{-1}(r_A \cos E / r_P)\right]}{\omega_A - \omega_P}$$

where

$$\alpha = E + SGN\left[(\underline{r}_A \times \underline{r}_P) \cdot \underline{u}\right] \cos^{-1}(\underline{r}_A \cdot \underline{r}_P / r_A r_P)$$

and ω_A, ω_P are the angular velocities of the active and passive vehicles respectively.

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

To help insure convergence, the following steps are taken:

a. The step size δt is restricted to 250 secs.

b. If the solution has been passed ($\delta E \, \delta E_0 < 0$), the step size is halved and forced in the opposite direction of the last step.

c. If the iteration is converging ($|\delta E_0| - |\delta E| < 0$), the sign of δt is maintained.

d. If the iteration is proceeding in the wrong direction, the step direction is reversed.

The iteration is terminated for any of the following reasons:

1. The iteration counter has exceeded its maximum value of 15.

2. The line-of-sight emanating from the active vehicle does not intersect the circular orbit with radius equal to that of the target vehicle.

3. The elevation angle is inconsistent with the relative altitudes of the two vehicles (e.g., if the elevation angle is less than 180^o when the active vehicle is above the target vehicle).

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

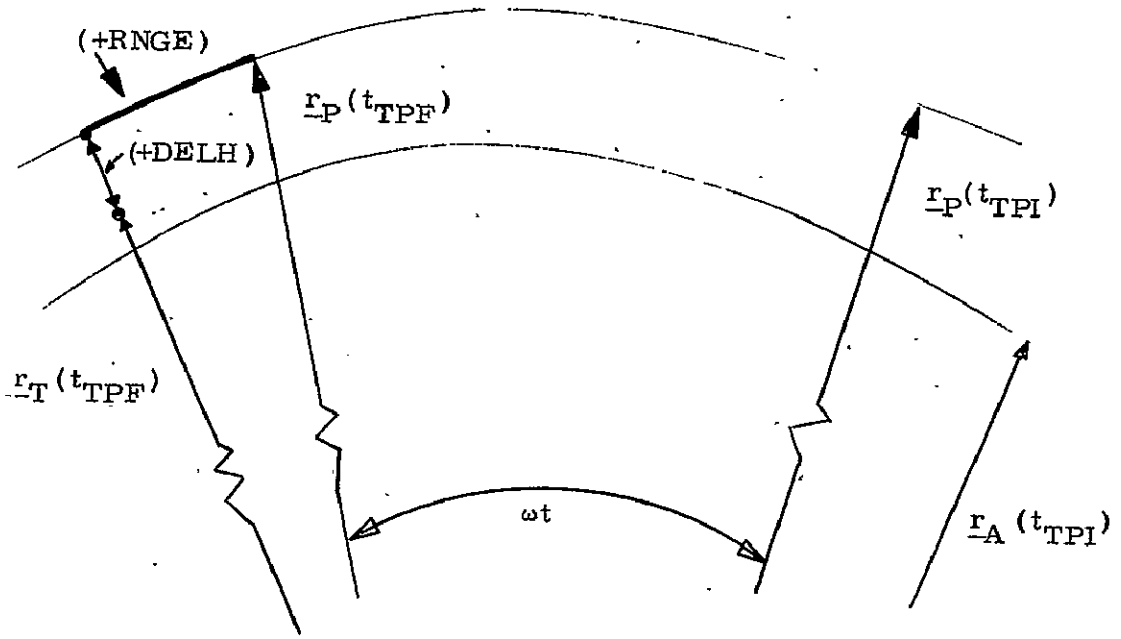
The transfer time t_F is then computed using ωt in the Conic Extrapolation Routine (Time-Theta option) and the passive vehicle

state vector is precision updated through this time yielding the target vector for the TPI maneuver. If an offset range, RNGE (see following diagram) is input, the passive vehicle state at t_{TPF} ($t_{TPF} = t_{TPI} + t_F$) is precision updated through an additional Δt to yield an offset target vector. The time increment Δt is given by the equation

$$\Delta t = \frac{RNGE}{(r_{PT}\ \omega_{PT})}$$

which assumes the passive vehicle is moving at a constant angular rate.

This modified target vector leads or trails the passive vehicle position at the intercept time and is in the same orbit as the passive vehicle.



If an altitude differential between the passive vehicle orbit and the target vector altitude is input, DELH, the target vector magnitude is increased or decreased accordingly.

The precision target vector thus determined, whether it be the passive vehicle position at t_{TPF} or a target offset from the passive vehicle position at t_{TPF}, is used in the Precision Lambert Routine to obtain the TPI maneuver. This routine uses the offset aimpoint technique to compensate for the earth's oblateness effects.

The coplanar intercept flag indicates if the TPI maneuver just computed and its associated aimpoint vector are to be modified or not. If the flag is set, the maneuver is modified in the following way: the computed TPI deltav is expressed in a frame (LV_P) where the Z axis is in the negative direction of the active vehicle position vector, the Y axis is normal to the passive vehicle plane and X forms a right handed system. The 3 components of deltav are ΔV_x, ΔV_y, ΔV_z. The active vehicle out-of-plane velocity relative to the passive vehicle orbital plane, $\dot{y}$, is then computed. The negative of this velocity overwrites Δv_y, and the modified TPI velocity vector in the (LV_P) frame becomes (Δv_x, $-\dot{y}$, Δv_z). This velocity is then expressed in the inertial reference frame and in the active vehicle local vertical frame.

A new target vector is then computed by precision integrating the active vehicle state at TPI with the modified deltav added in to the time t_{TPF}. The time of the nodal crossing at which a midcourse maneuver is to be performed to make the orbits coplanar is then calculated using the Time-Theta option of the Conic Extrapolation Routine with $\omega t = 90^{\circ}$.

The TPI maneuver $\Delta \underline{V}_{TPI}$, the precision target vector $\underline{r}_{PT}$ and the time of flight t_F are used in the powered flight routine which controls the vehicle and determines thrusting attitude during the maneuver.

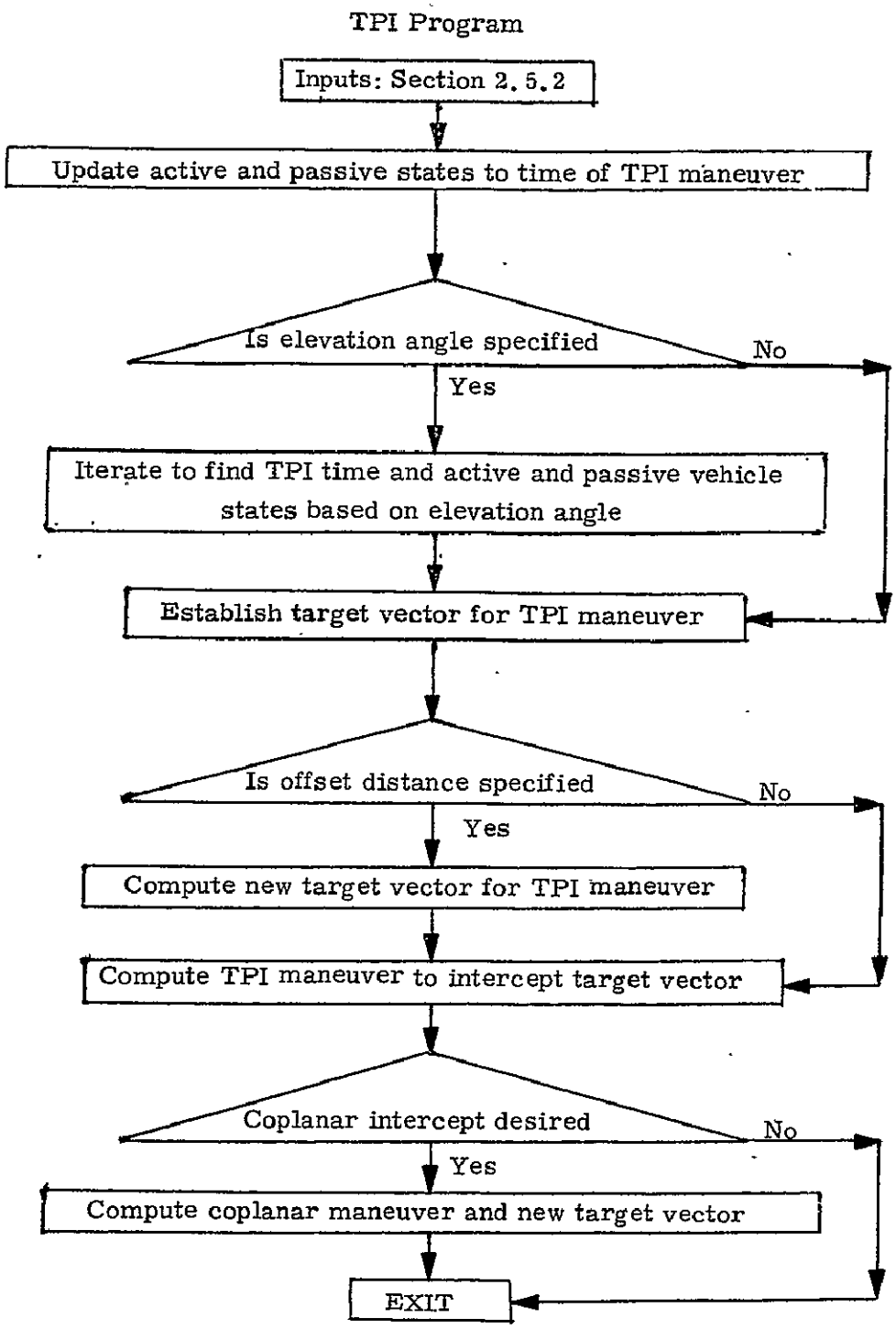


Figure 2.5.4 TPI Functional Flow Chart

TPI Program

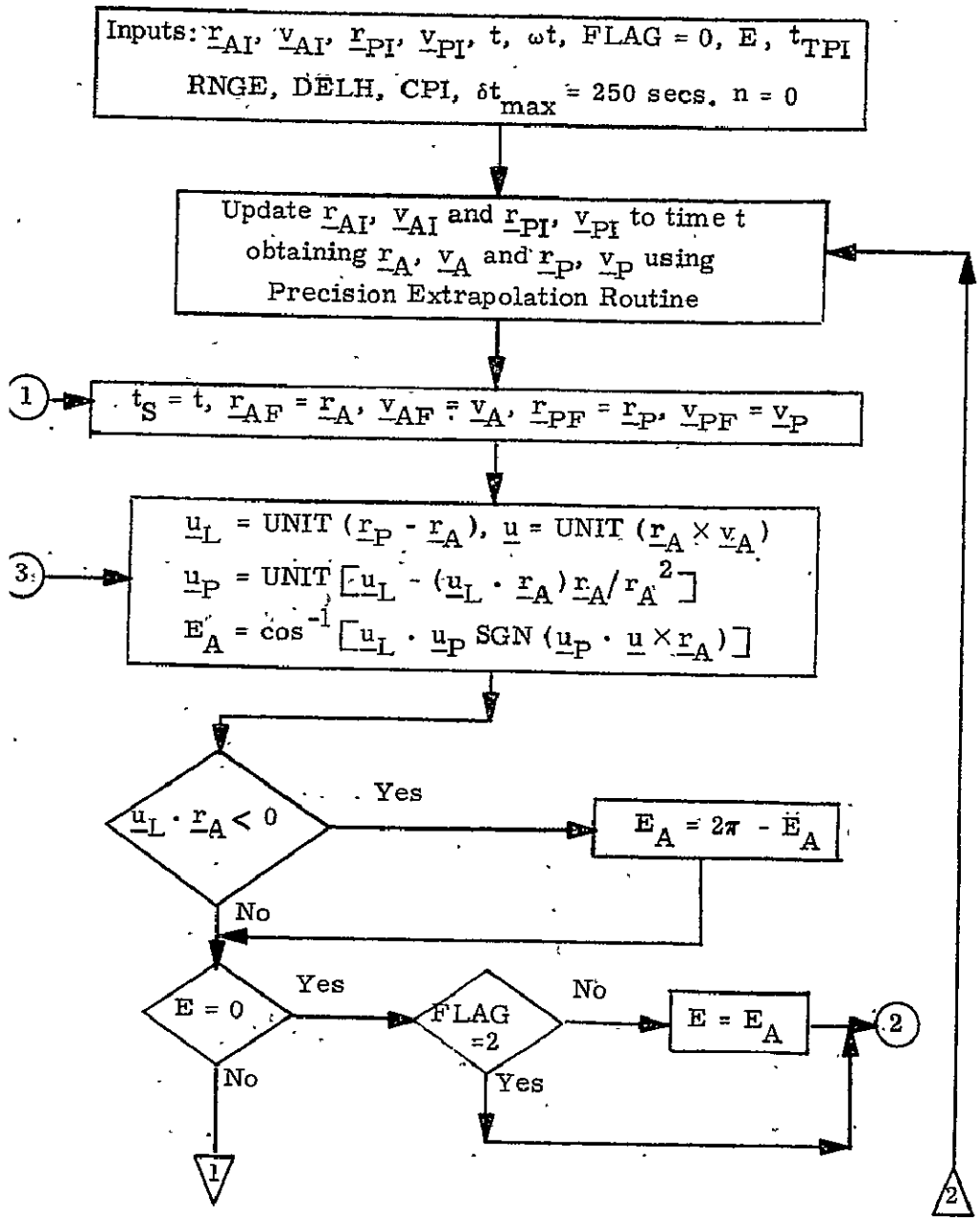


Figure 2.5.5 TPI Detailed Flow Chart

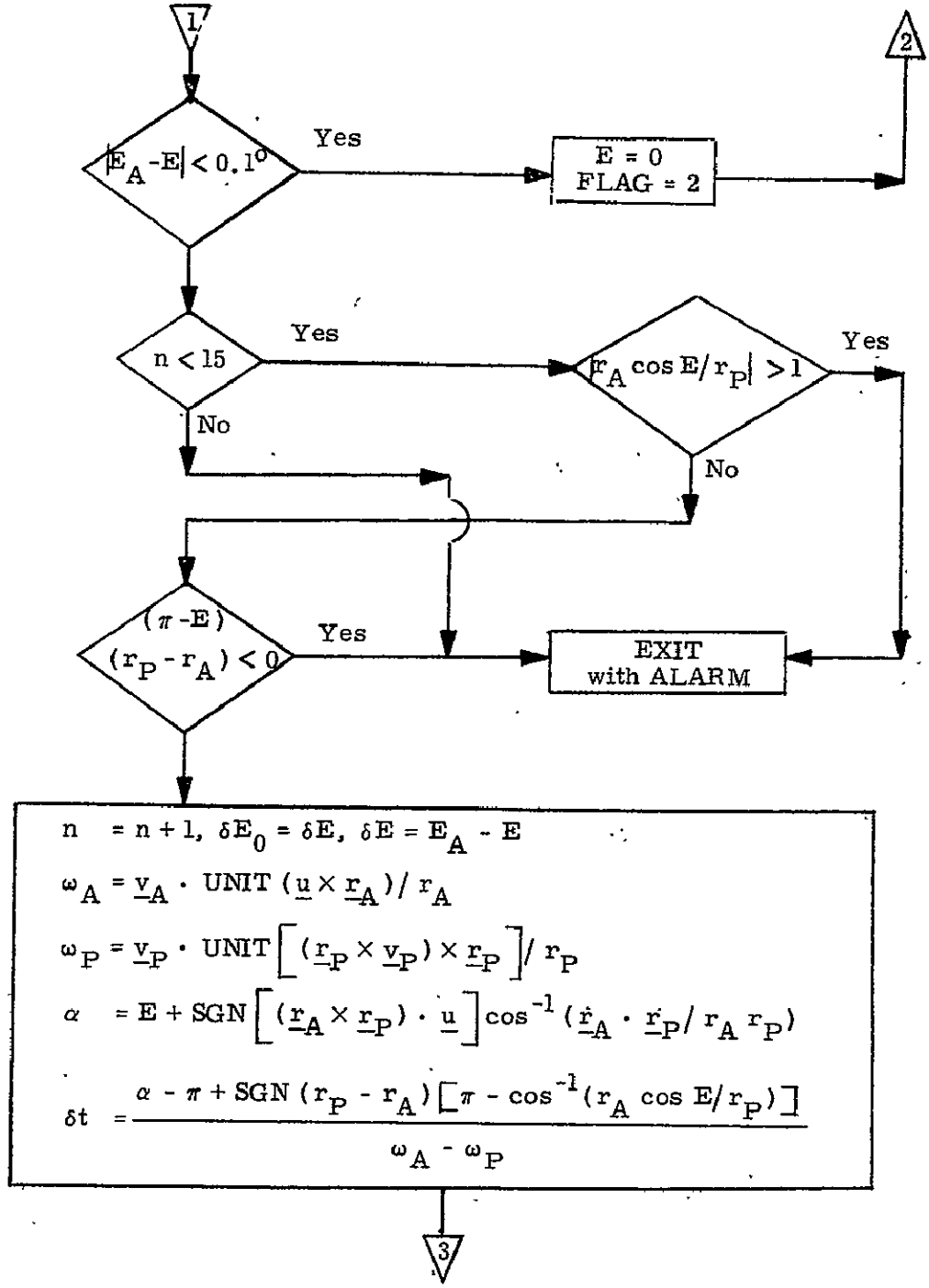


Figure 2.5.5 TPI Detailed Flow Chart

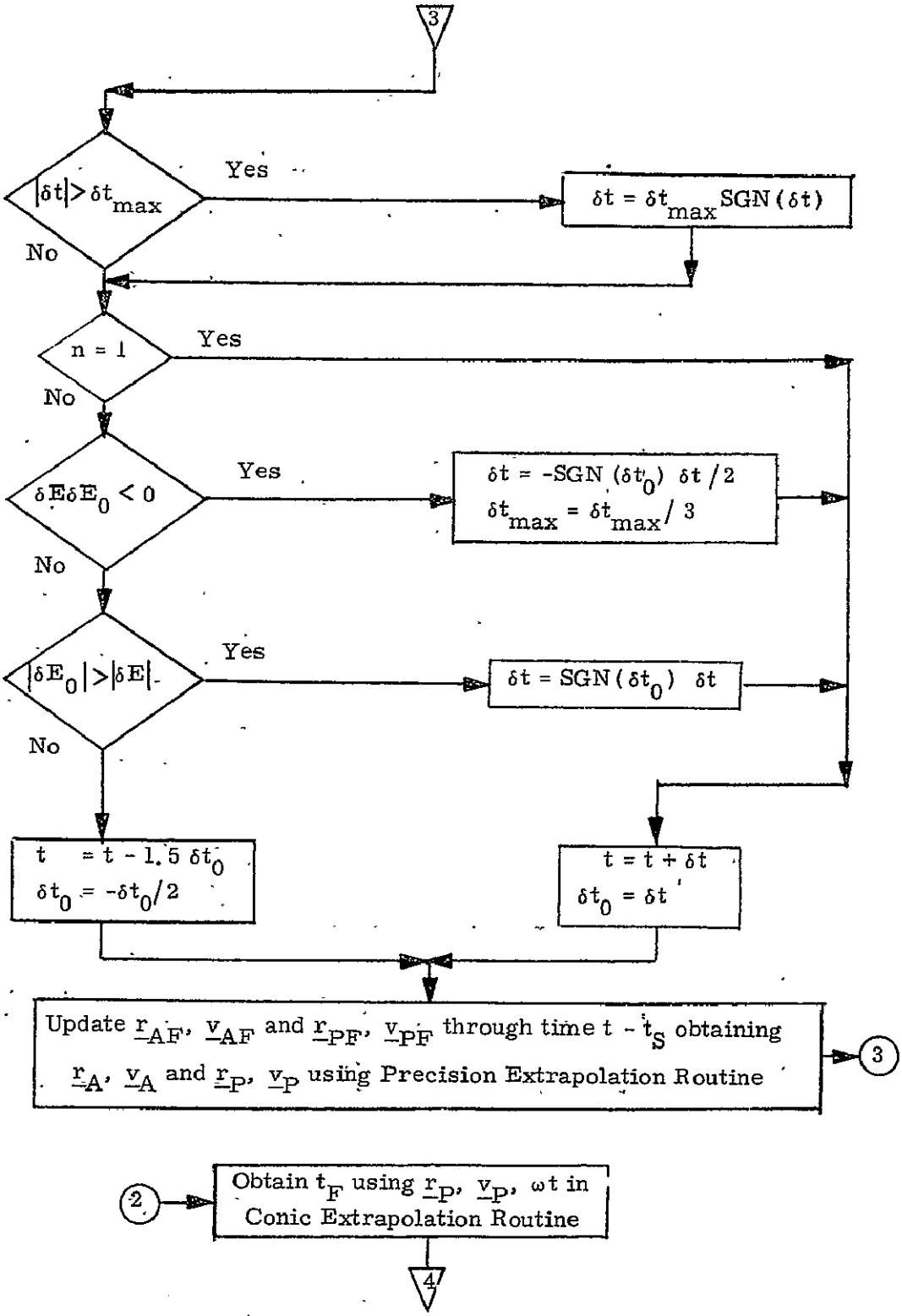


Figure 2.5.5 TPI Detailed Flow Chart

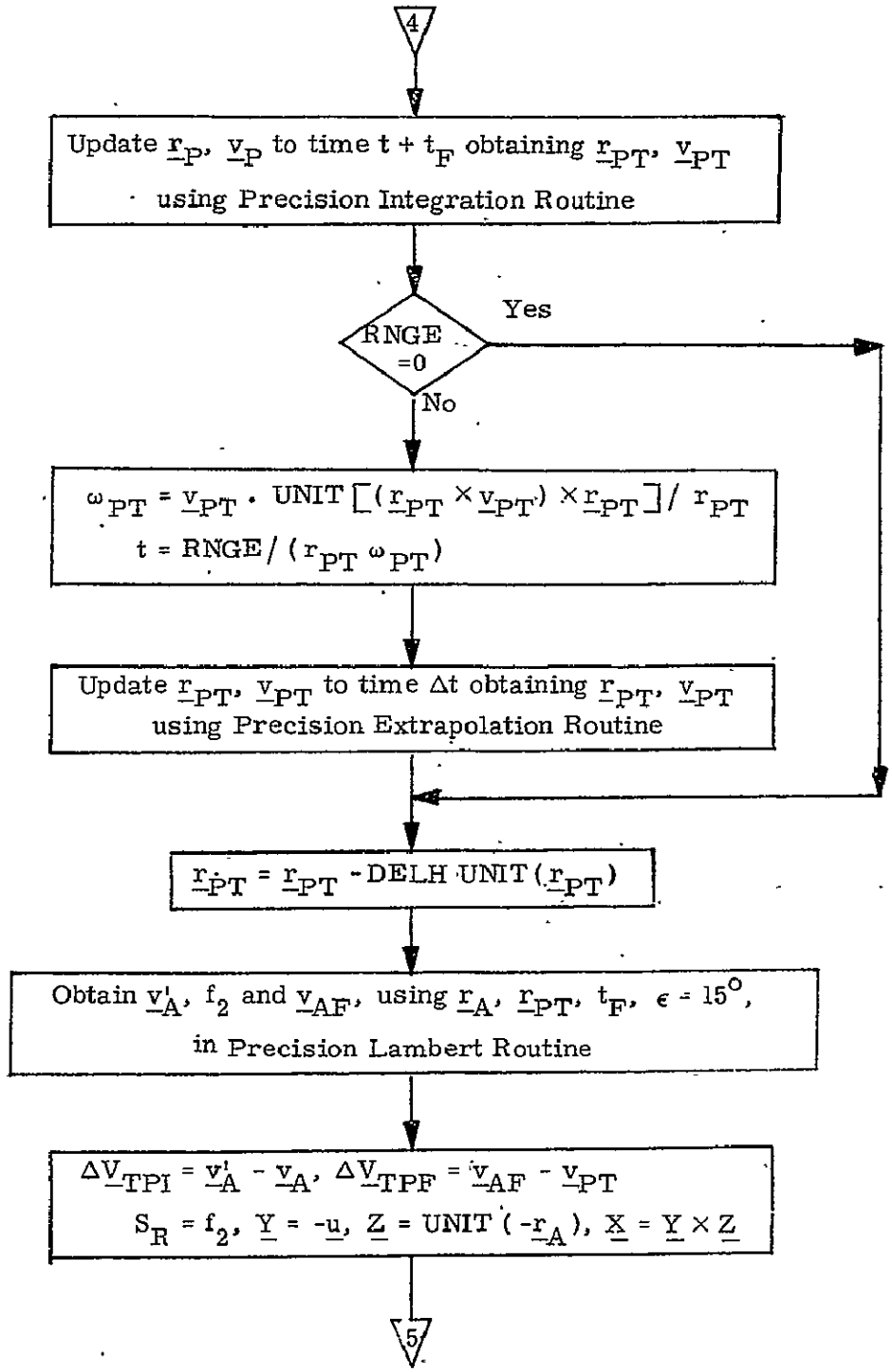


Figure 2.5.5 TPI Detailed Flow Chart

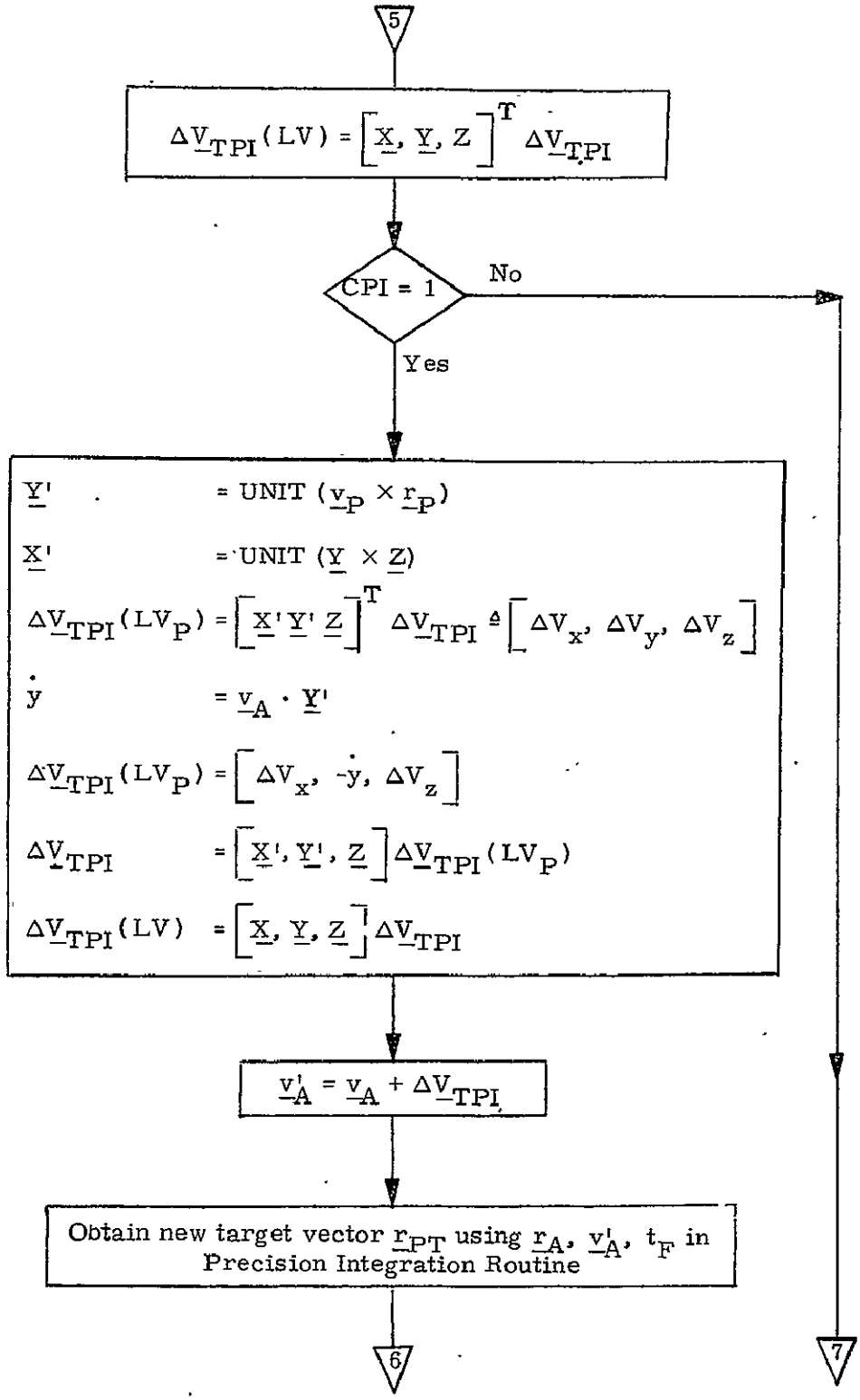


Figure 2.5.5 TPI Detailed Flow Chart

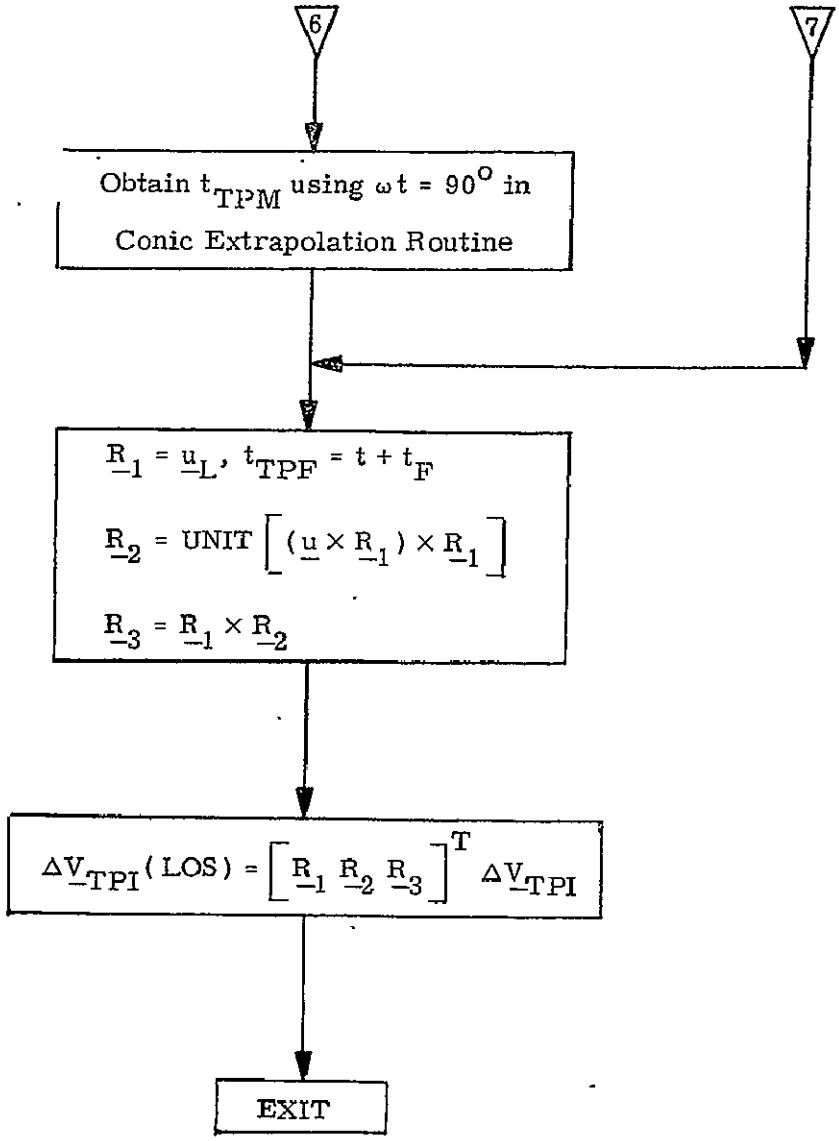


Figure 2.5.5 TPI Detailed Flow Chart

2.6 Rendezvous Midcourse Maneuver Program (TPM)

2.6.1 TPM Program Objectives

The Rendezvous Midcourse Maneuver Program computes a midcourse correction (TPM) which insures that the active vehicle will intercept the desired target vector at the desired intercept time established in the TPI program. The functional flow is illustrated in Fig. 2.6.4

The time of the midcourse correction is chosen so that an adequate time interval from the preceeding rendezvous maneuver (either TPI or a midcourse correction) occurs during which rendezvous navigation can be performed.

If the coplanar intercept mode was used at TPI then the program is entered for the first midcourse correction at the time corresponding to the nodal crossing of the active and passive vehicle orbital planes. This time is computed in the TPI program.

The TPM program recomputes the target vector to achieve the same objective as the TPI program attempted to satisfy. This objective is to either intercept the passive vehicle or achieve a desired target position relative to the passive vehicle at the intercept time.

2.6.2 TPM Program Input-Output

The required inputs to the Terminal Phase Midcourse program are described below:

$\underline{r}_A$, $\underline{v}_A$ - active vehicle state vector

$\underline{r}_P$, $\underline{v}_P$ - passive vehicle state vector

t_{TPI}	-	time of TPI
t_{TPM}	-	time of the midcourse correction
t_F	-	transfer time between TPI and TPF
RNGE	-	distance along the passive vehicle orbit from its position at the intercept time which defines a new target vector
DELH	-	altitude above or below the passive vehicle orbit, measured along the target vector at the intercept time which defines a new target vector

The outputs from this program are:

$\Delta\underline{V}_{TPM}$	-	TPM deltav in inertial reference frame
$\Delta\underline{V}_{TPM}$(LOS)	-	TPM deltav in the line-of-sight coordinate frame
$\underline{r}_{PT}$		target vector for the midcourse maneuver

2.6.3 TPM Program Description

The Terminal Phase Midcourse program is entered after the TPI maneuver or a preceeding midcourse correction. The time of the midcourse correction is chosen so that an adequate time period from the previous maneuver occurs allowing rendezvous navigation to be performed.

If the midcourse is the first one after a TPI maneuver in which the coplanar intercept mode was used, then the time of the midcourse is that which was determined in the TPI program.

The on-board estimated states of the active and passive vehicles are precision extrapolated to the midcourse correction time, t_{TPM}. The passive vehicle state is then precision updated to the final intercept time, t_{TPF}, yielding the target vector for the midcourse maneuver.

If an offset range, RNGE, is input, the passive vehicle state at the intercept time is updated through an additional Δt to yield an offset target vector. The time increment Δt is given by the equation

$$\Delta t = \frac{RNGE}{(r_{PT}\,\omega_{PT})}$$

which assumes the passive vehicle is moving at a constant angular rate.

This modified target vector leads or trails the passive vehicle position at t_{TPF} and is in the same orbit as the passive vehicle.

If an altitude differential between the passive vehicle orbit and the target vector magnitude is input, DELH, the target vector magnitude is increased or decreased accordingly.

The precision target vector thus determined is used in the Precision Lambert Routine to obtain the midcourse correction maneuver. This routine uses the offset aimpoint technique to compensate for the earth's oblateness effects.

Terminal Phase Midcourse Program

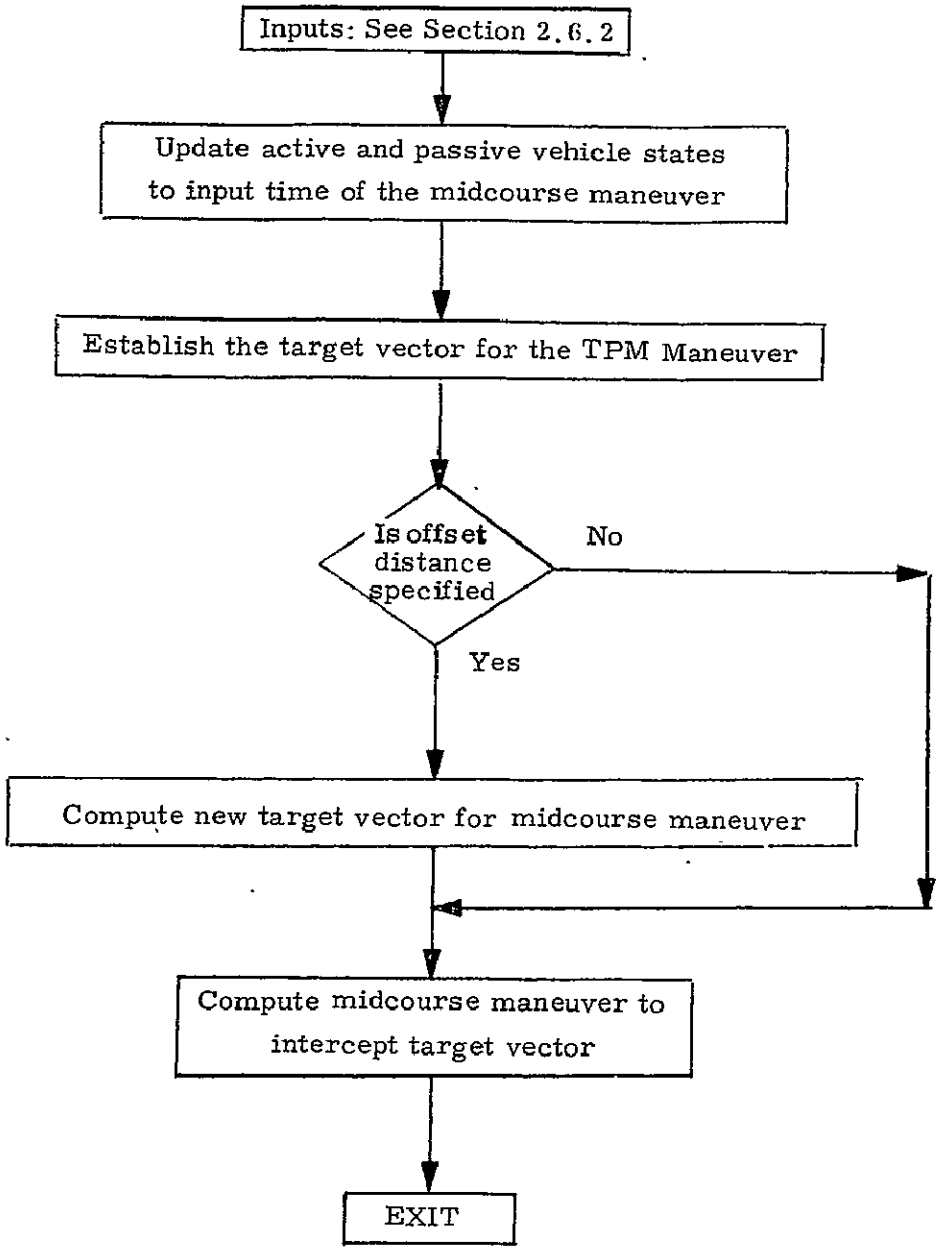


Figure 2.6.4 TPM Functional Flow Chart

Terminal Phase Midcourse Program

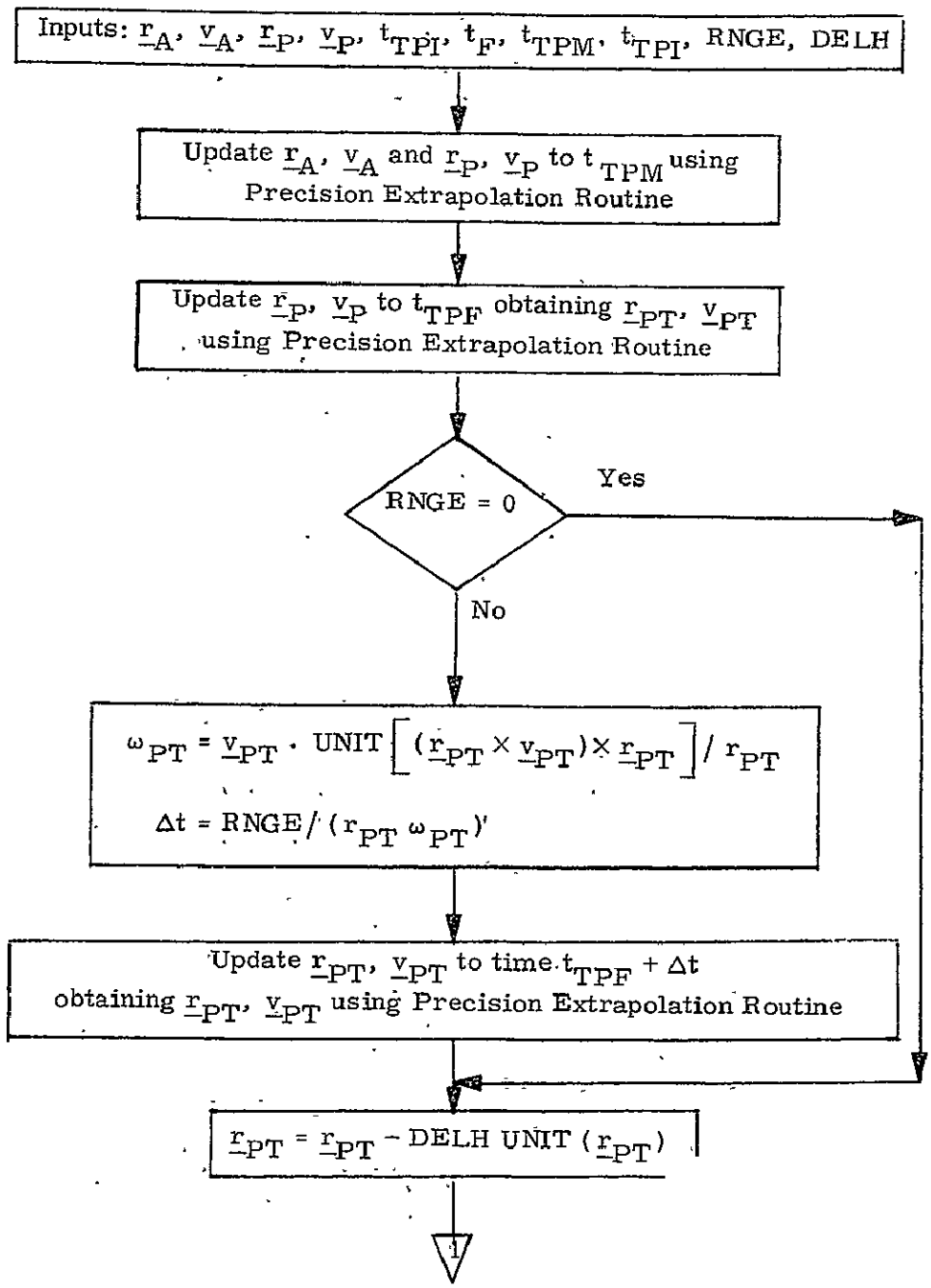


Figure 2.6.5 TPM Detailed Flow Chart

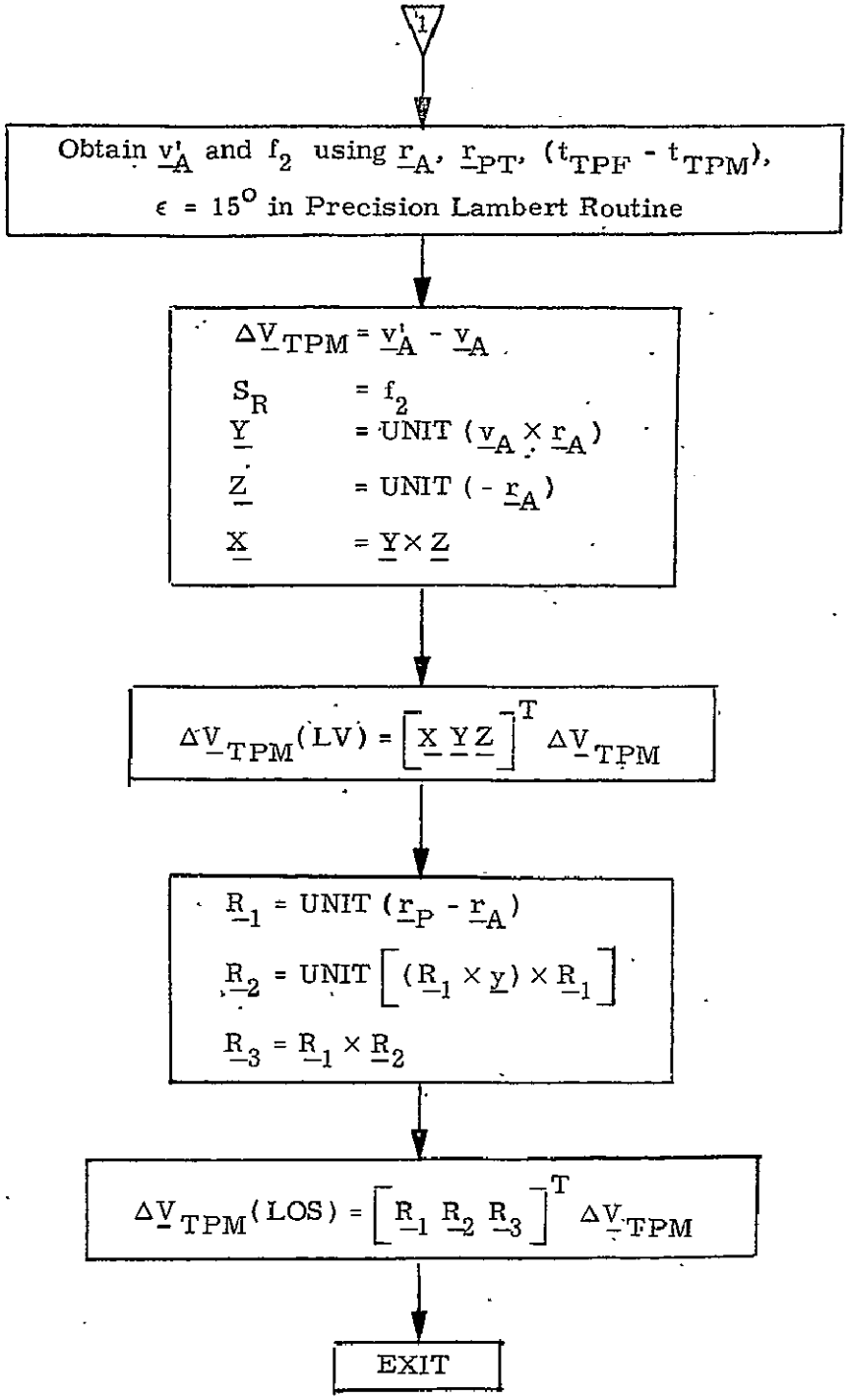


Figure 2.6.5 TPM Detailed Flow Chart

2.7 Plane Change Program (PC)

2.7.1 PC Program Objectives

The objective of the Plane Change Maneuver is to compute a maneuver which will place the active vehicle into the plane of the passive vehicle. This is accomplished by nulling the out-of-plane velocity at a nodal point established by the previous maneuver. Fig. 2.7.4 is functional flow chart of the PC program.

2.7.2 PC Program Input-Outputs

The required inputs to the Plane Change Program are

$\underline{r}_A$, $\underline{v}_A$, $\underline{r}_P$, $\underline{v}_P$	Active and passive state vectors
t	Time of the plane change maneuver

The outputs to this program are:

$\Delta\underline{v}_{PC}$	PC maneuver Δv

2.7.3 PC Program Description

A detailed flow chart of this program is shown in Fig. 2.7.5.

The program starts with a precision extrapolation of the state vectors to the time of the PC maneuver. The out-of-plane velocity of the active vehicle is then calculated and inserted into the PC maneuver.

Plane Change

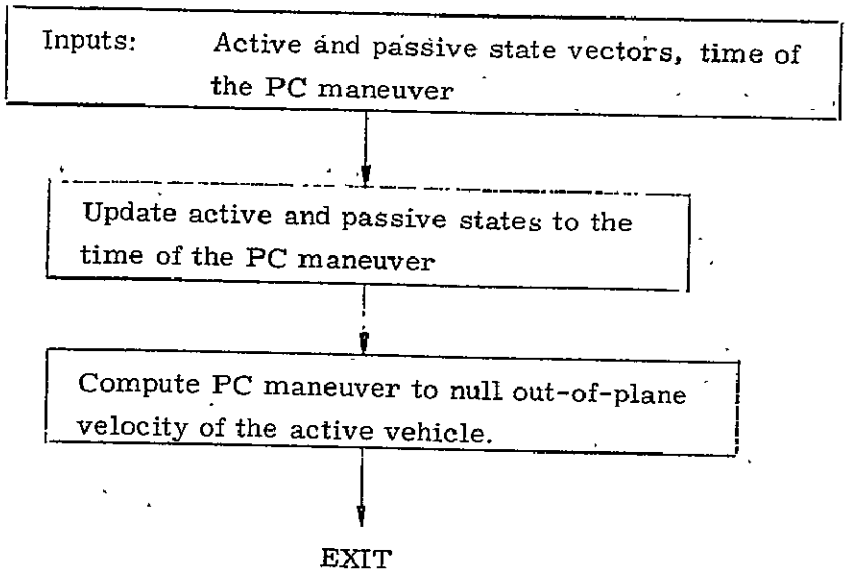


Fig. 2.7.4 PC Functional Flow Chart

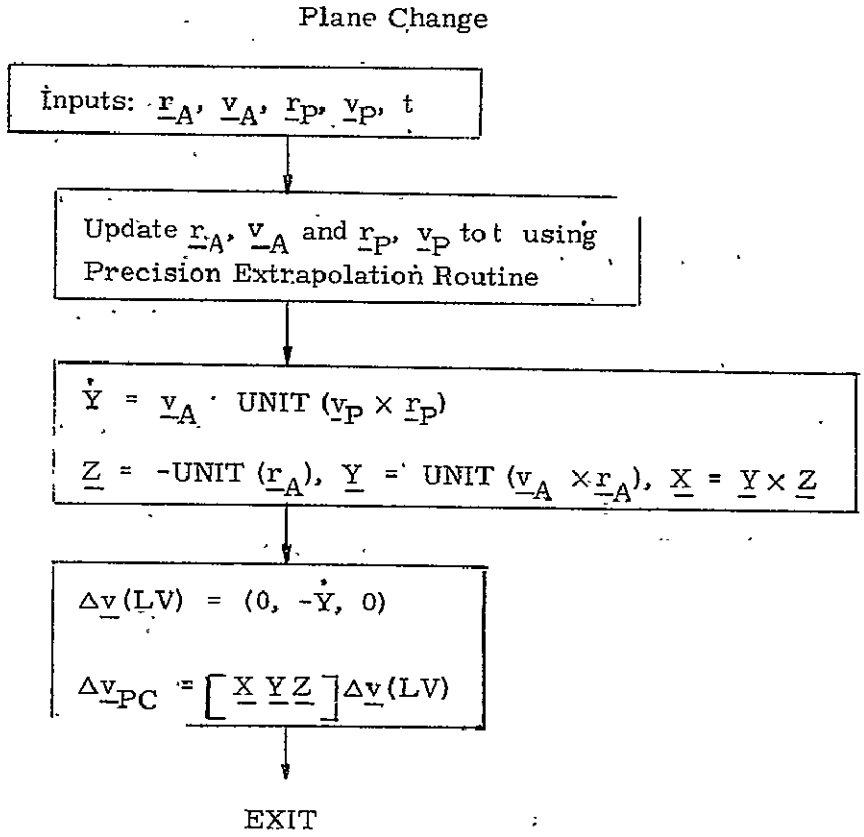


Fig. 2.7.5 PC Detailed Flow Chart

2.8 Phase Match Program (PM)

2.8.1 PM Program Objectives

The objective of the Phase Match Program is to compute a set of state vectors that are "phase matched". These vectors, in the coplanar case, will be colinear. Hence, when entering a rendezvous targeting program with phase matched vectors, the program will approximately update the two state vectors over the same sector of the earth's surface.

In general, at the start of a typical targeting program, the passive vehicle will lead the active vehicle. In this case the passive vehicle has to be backed up using precision extrapolation to establish the phase matched vectors. The functional and detailed flow charts are shown for this case in Fig. 2.8.4 and 2.8.5.

2.8.2 PM Program Input-Output

The required inputs to the Phase Match Program are

$\underline{r}_A$, $\underline{v}_A$, $\underline{r}_P$, $\underline{v}_P$	Active and passive state vectors at time of first maneuvers to be employed in the targeting program

The outputs to this program are

$\underline{r}_P$, $\underline{v}_P$	Passive state vector at phase match
t	Time associated with the new passive state vector

2.8.3 PM Program Description

A detailed flow chart of this program is shown in Fig. 2.8.5.

The program starts by computing the desired unit vector of the passive vehicle at phase match.

$$\underline{u}_D = \text{UNIT}\left\{\left[(\underline{r}_A \times \underline{v}_A) \times \underline{r}_A\right] \times (\underline{r}_P \times \underline{v}_P)\right\}$$

3 ⊕

This vector is colinear with the intersection of the passive vehicle plane and the plane perpendicular to the active plane which contains the active vehicle's position vector. The central angle θ is then computed between the passive vehicle's position vector and $\underline{u}_D$ as a positive angle when the passive vehicle leads the active vehicle. After checking to see if this angle is sufficiently small to terminate the iteration, the central angle is converted into a corresponding time Δt using a conic extrapolation routine. The passive state is then precision extrapolated through Δt to obtain a new estimate of the desired phase matched vector.

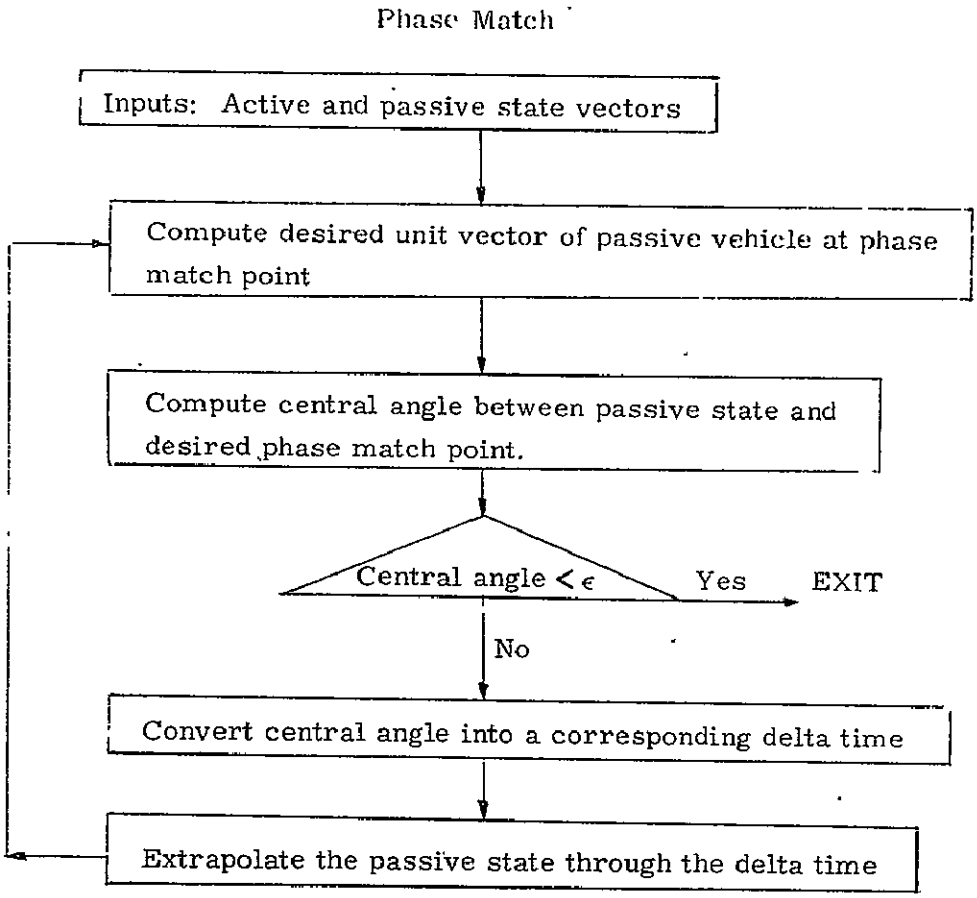


Fig. 2.8.4 PM Functional Flow Chart

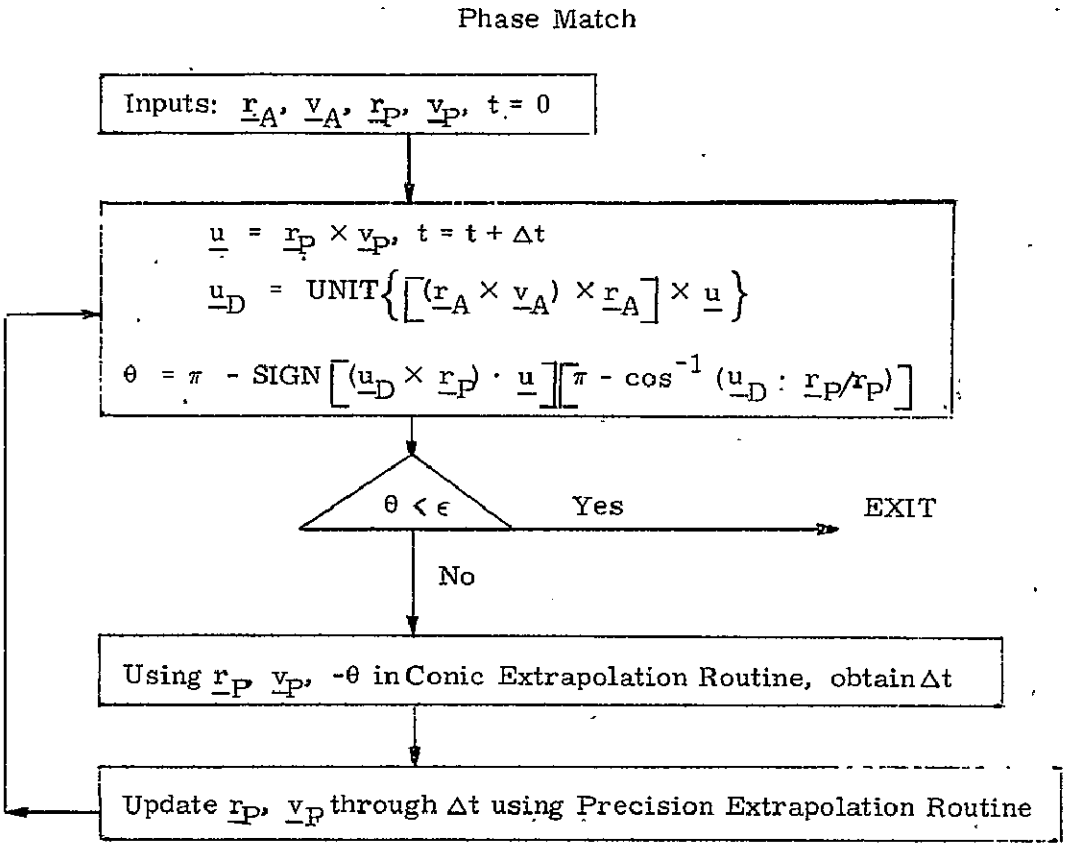


Fig. 2.8.5 PM Detailed Flow Chart

3. THE ASSEMBLY OF THE INDEPENDENT MANEUVERS INTO PRETERMINAL RENDEZVOUS PROGRAMS

As an application of the assembly of the basic preterminal maneuvers into an integrated rendezvous program, a Four Maneuver Sequence (FMS) will be considered. This FMS will closely parallel the four maneuver Skylab NC1 program. The EMS consists of a phase maneuver enclosing two height maneuvers; the latter enclosing a coelliptic maneuver. Also included is a phase match operation prior to the phase maneuver.

Following the FMS targeting and the addition of the phase maneuver to the active vehicle state, the Triple Maneuver Sequence TMS will define the targeting for the three remaining maneuvers. This sequence parallels the NC2 program in Skylab. It consists of a phase maneuver which encloses a height maneuver; the latter enclosing a coelliptic maneuver. The TMS also is initiated with a phase match operation.

Following the TMS targeting and the addition of the new phase maneuver to the active vehicle state, the Double Maneuver Sequence DMS will define the targeting for the two remaining maneuvers. This sequence parallels the NCC program in Skylab. It consists of a Lambert maneuver using an aim point established by inserting a coelliptic orbit at the TPI point using the TPI geometry constraints.

Following the DMS targeting and the addition of the new Lambert maneuver to the active vehicle state, the Single Maneuver Sequence SMS will define the targeting for the final preterminal maneuver. This maneuver is a coelliptic maneuver resulting in orbits that are separated by an approximately constant radial distance.

The maneuver designations for each rendezvous sequence are:

Sequence	Maneuvers
FMS	NP, NH1, NH2, NC
TMS	NP, NH, NC
DMS	NL, NC
SMS	NC

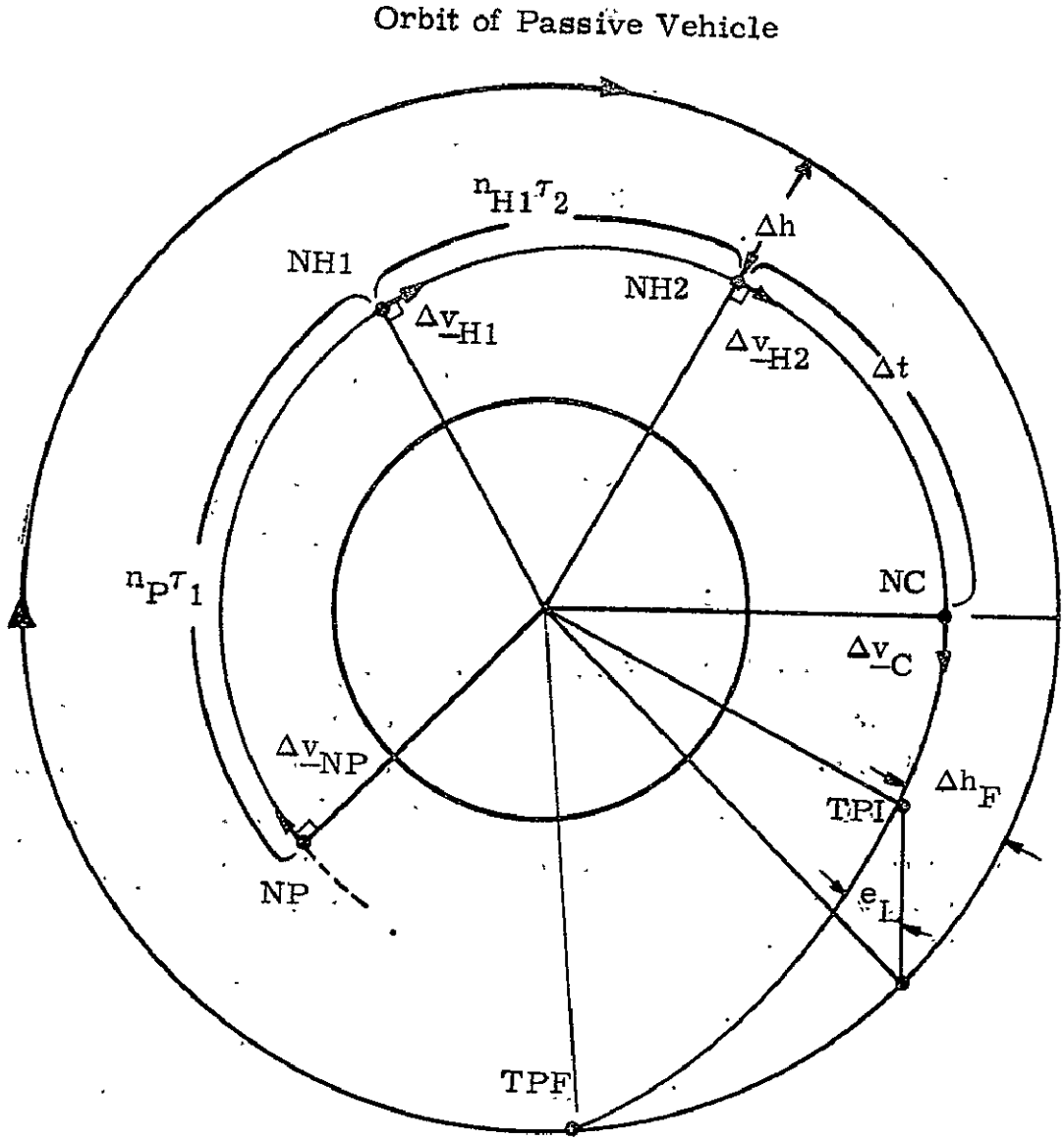
The FMS and TMS programs are combined into one flow chart. Fig. 3 is an illustration of the FMS rendezvous.

This section is divided into five parts. The first section lists the constraints which each maneuver sequence satisfies. The following sections are devoted to the individual maneuver programs.

3.1 Constraints Associated with the Preterminal Rendezvous Programs

Sufficient constraints must be imposed on each maneuver sequence to uniquely define the trajectory (see Sec. 1.2). In general, the constraints for the N-1 maneuver sequence can be obtained by deleting three constraints from the N maneuver sequence. The targeting constraints associated with the FMS mentioned above are

1. $\Delta \underline{v}_{NP}$ horizontal
2. $\Delta \underline{v}_{NH1}$ horizontal
3. $\Delta \underline{v}_{NH2}$ horizontal
4. NH1 maneuver occurs after a transfer time of $n_1 \tau$, where n_1 is a specified number of revolutions and τ is the period of the post NP maneuver orbit.
5. NH2 maneuver occurs after a transfer time of $n_2 \tau$, where n_2 is a specified number of revolutions and τ is the period of the post NH1 maneuver orbit.
6. NC occurs a specified time Δt from the NH2



τ_1 = Period of post NP maneuver orbit

τ_2 = Period of post NH1 maneuver orbit

Figure 3 Illustration of the FMS Rendezvous

maneuver.

7. Altitude between passive orbit and active vehicle at the NH2 time, Δh.
8. Radial velocity at NC computed by coelliptizing the orbit.
9. Horizontal velocity at NC computed by coelliptizing the orbit.
10. TPI time, t_f.
11. TPI elevation angle, e_L.
12. Altitude between passive orbit and active vehicle at TPI time, Δh_f.

The targeting constraints for the TMS are obtained by removing the constraints 1, 4 and 7 from the FMS. For DMS, the additional constraints 2, 3 and 5 are removed. The constraints on the coelliptic orbit are applied at the TPI point during the DMS program. For SMS, the coelliptic maneuver requires constraints 8 and 9. Summarizing, the constraints which apply to the rendezvous sequences are shown below:

Sequence	Applicable Constraints
TMS	2, 3, 5, 6, 8, 9, 10, 11, 12
DMS	6, 8, 9, 10, 11, 12
SMS	8, 9

As the coelliptic maneuver is independent of the TPI elevation angle and altitude constraints, these constraints will not be satisfied by the active vehicle at the TPI time.

3.2 Four Maneuver Sequence (FMS)

3.2.1 FMS Program Objectives

The objective of the FMS program is to compute a rendezvous trajectory which satisfies the constraints listed in Sec. 3.1. The program computes the NP maneuver as well as the magnitudes of NH1, NH2 and NC maneuvers.

This program contains two independent iteration loops embedded in an outer iteration loop. Each loop is iterated with a Newton Raphson iterative scheme. The two inner loops determine the two height maneuvers (NH1 and NH2) and the outer loop determines the phase maneuver (NP).

A functional flow chart of the FMS program is shown in Fig. 3.2.4. If the time of the nodal crossing is desired following the NP maneuver, the active vehicle is extrapolated through 90 degrees of central angle travel following the NP maneuver to establish this time.

3.2.2 FMS Program Input - Output

The inputs for the Four Maneuver Sequence Program are:

$\underline{r}_A, \underline{v}_A, \underline{r}_P, \underline{v}_P$	State vectors of active and passive vehicle
t_P	Time of the NP maneuver
Δt	Time between the NH2 and NC maneuvers
t_F	Time of the TPI maneuver
Δh_F	Altitude between the passive orbit and the active vehicle at the TPI time

e_L	Elevation angle
S=1	FMS - TMS mode switch
n_P	Number of revolutions between the NP and NH1 maneuvers
n_{H1}	Number of revolutions between the NH1 and NH2 maneuvers
Δh	Altitude between the passive orbit and the active vehicle at the NH2 time
spc	Plane change maneuver switch, (= 1 for computation of t_{PC})
dv_P, dv_{H1} , dv_{H2}	Nominal magnitudes of phase and height maneuvers

The outputs to the FMS Program are:

$\Delta\underline{v}_P$	NP maneuver $\Delta\underline{v}$
$\Delta\underline{v}_P$(LV)	NP maneuver $\Delta\underline{v}$ in local vertical coordinates
t_{H1}	Time of the NH1 maneuver
t_{PC}	Time of the plane change maneuver (if required)
Δv_{H1}	Magnitude of the NH1 maneuver
Δv_{H2}	Magnitude of the NH2 maneuver
Δv_C	Magnitude of the NC maneuver

3.2.3 FMS Program Description

The FMS program logic is illustrated in the detailed flow chart in Fig. 3.2.5. This program involves two independent iterative loops embedded in an outer iterative loop in order to compute a trajectory which satisfies the FMS constraints. The two inner loops use the magnitudes of the NH1 and NH2 maneuvers as independent variables, while the outer loop utilizes the magnitude of the NP maneuver. Nominal values for these maneuvers are used in the program as initial guesses in the iterative loops.

Prior to entering the iterative loops, the passive state vector is "phase matched" to the active state vector at the time of the NP maneuver. This results in the active and passive vehicles traversing approximately the same central angle during the NP to TPI phase of the rendezvous. As 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 NP program exclusively employs conic updates without losing significant accuracy compared with a program using precision updates.

The phase match procedure starts by updating the active and passive vehicles to the NP time with precision and conic updates, respectively. A unit vector $\underline{urp}_D$ is then calculated which is colinear with the intersection of the passive vehicle plane and the plane perpendicular to the active plane which contains the active vehicle's position vector.

$$\underline{urp}_D = \text{UNIT}\{[(\underline{r}_{AP} \times \underline{v}_{AP}) \times \underline{r}_{AP}] \times (\underline{r}_{PP} \times \underline{v}_{PP})\}$$

The angle θ between the two vehicles is next calculated and defined as a positive angle for the passive vehicle leading the active vehicle. After converting θ into a corresponding time, the passive vehicle is precision updated to the estimated phase match time t_N. The updated position is then checked to see if it is approximately

colinear with $\underline{urp}_D$. If not, the calculation is repeated starting with the calculation of θ.

For n_P's in excess of 18 there is a possibility that the passive vehicle might lead the active vehicle by a central angle which exceeds 360 degrees. To allow for this possibility, the equation for θ includes a term which will increase θ by 360 degrees when n_P exceeds 18 and the passive vehicle leads the active vehicle by less than 180 degrees. The final formula for θ is

$$\theta = pi - c[pi - \cos^{-1}(\underline{urp}_D \cdot \underline{r}_{PP}/r_{PP})] + (d+1)(c+1)pi/2$$

where $d = SIGN(n_P - 18)$

$$c = SIGN[(\underline{urp}_D \times \underline{r}_{PP}) \cdot UNIT(\underline{r}_{PP} \times \underline{v}_{PP})]$$

and $pi = \pi$ on the first pass through θ and otherwise equals zero.

Upon successfully completing phase match, the active vehicle's state vector is rotated into the plane of passive vehicle's orbit. The passive vehicle is then updated to TPI time and the desired active vehicle's position vector $\underline{r}_{AFD}$ at TPI is established using the QRDTPI subroutine (sec. 3.6.9). This subroutine uses the elevation angle and the altitude between the passive orbit and the active vehicle at the TPI time in an iterative search which solves for the TPI geometry.

The outer loop is initiated by adding the NP maneuver to the active vehicle's NP velocity vector. Using the REVUP subroutine (Sec. 3.6.1), the new state vector is updated through n_P revolutions to the NH1 point.

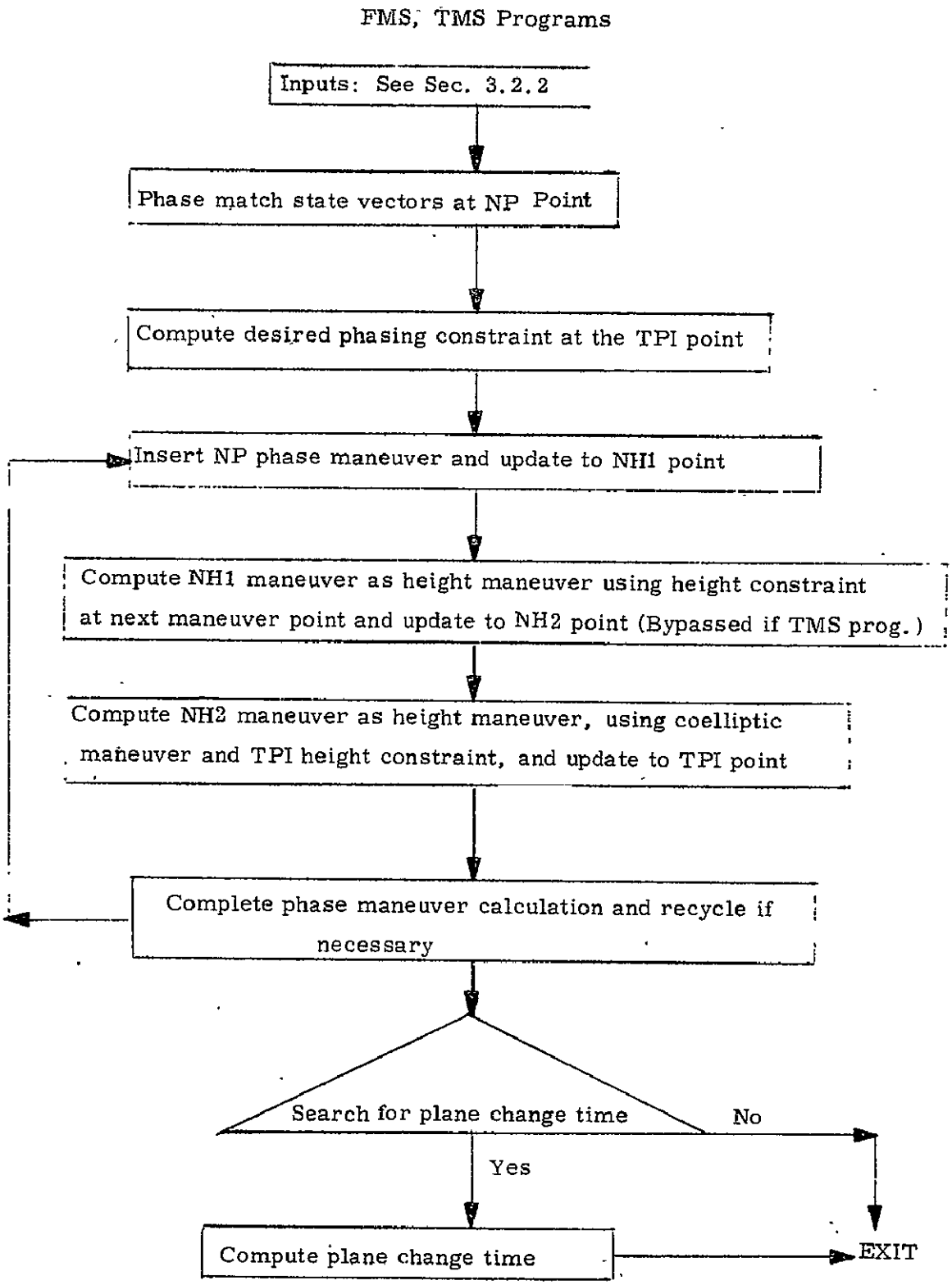
Prior to entering the first height iterative loop, the iteration counter switch c_I is set equal to 0 or .5 if it is the first or subsequent pass through the outer loop, respectively. This height loop starts by adding the estimated NH1 maneuver to the active vehicle's velocity vector. The state is then advanced to the NH2 maneuver time using n_{H1} in the REVUP routine. To verify that the height

constraint Δh is satisfied, the passive vehicle must first be updated to be radially above the active vehicle. This is accomplished using RADUP subroutine (Sec. 3.6.3) which computes the central angle between the two vehicles and updates the passive vehicle state through that angle. An altitude error e_{H1} is then defined and checked to see if it is sufficiently small. This test is bypassed if the partial p_{H1} employed in the iterative loop is zero in order that a value for p_{H1} may be computed for use in subsequent passes through the NH1 maneuver computation. If e_{H1}, passes the test, the program proceeds to the second height maneuver. Otherwise, the ITER subroutine (Sec. 3.6.7) is used to obtain a new estimate of the NH1 maneuver and the first height maneuver calculation is repeated. ITER is a subroutine encompassing a Newton Raphson iterative scheme based on numerical partials.

Following convergence of the first height maneuver, the calculation of the second height maneuver NH1 starts by setting the iteration counter c_I to 0 or .5 depending on the value of c_F. The estimated value of the NH2 maneuver is then added to the active vehicle's state at NH2. After updating the state through the time Δt to obtain the active vehicle's state at the NC point, the passive state is updated to be radially above the active vehicle's position vector using the RADUP subroutine. The coelliptic velocity for the active vehicle is then computed using the COE subroutine (Sec. 3.6.6) and the active's state vector is updated to the TPI time t_F. The passive state is again updated to be radially above the active vehicle's position vector using the RADUP subroutine.

Based on the desired altitude difference Δh_F, an altitude error e_{H2} is defined. If the partial p_{H2} employed in the iterative loop is zero and this error is sufficiently small, the program proceeds to the outer loop. Otherwise, the ITER subroutine is called to obtain a new value of the NH1 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.

In addition to a constraint on the altitude at the TPI point, there is a constraint on the phasing. The updated active vehicle's position vector $\underline{r}_{AF}$ should be approximately colinear with $\underline{r}_{AFD}$. An angular error e_P is defined which represents the central angle between $\underline{r}_{AF}$ and $\underline{r}_{AFD}$. If this error is sufficiently small the program is terminated after calculating the required displays. Otherwise, the ITER subroutine is called to obtain a new value of the NP maneuver and to update values of the outer loop iteration counter c_F, the partial p_C and the stored values of the error eo_P and maneuver magnitude dvo_C.


FMS, TMS Programs
Inputs: See Sec. 3.2.2
Phase match state vectors at NP Point
Compute desired phasing constraint at the TPI point
Insert NP phase maneuver and update to NH1 point
Compute NH1 maneuver as height maneuver using height constraint at next maneuver point and update to NH2 point (Bypassed if TMS prog.)
Compute NH2 maneuver as height maneuver, using coelliptic maneuver and TPI height constraint, and update to TPI point
Complete phase maneuver calculation and recycle if necessary
Search for plane change time
No
Yes
Compute plane change time
EXIT

FMS, TMS Programs

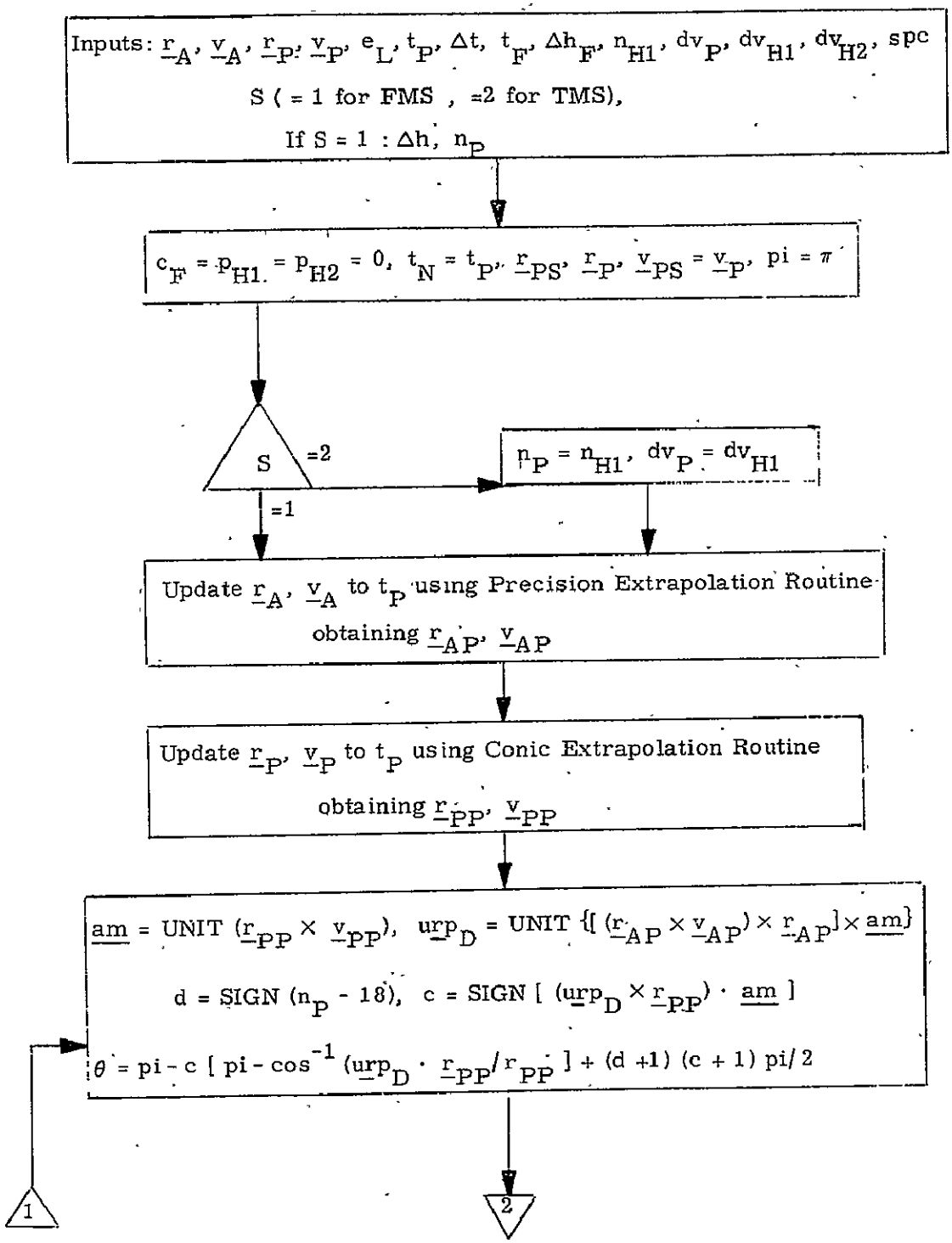


Figure 3.2.5 FMS, TMS Detailed Flow Chart

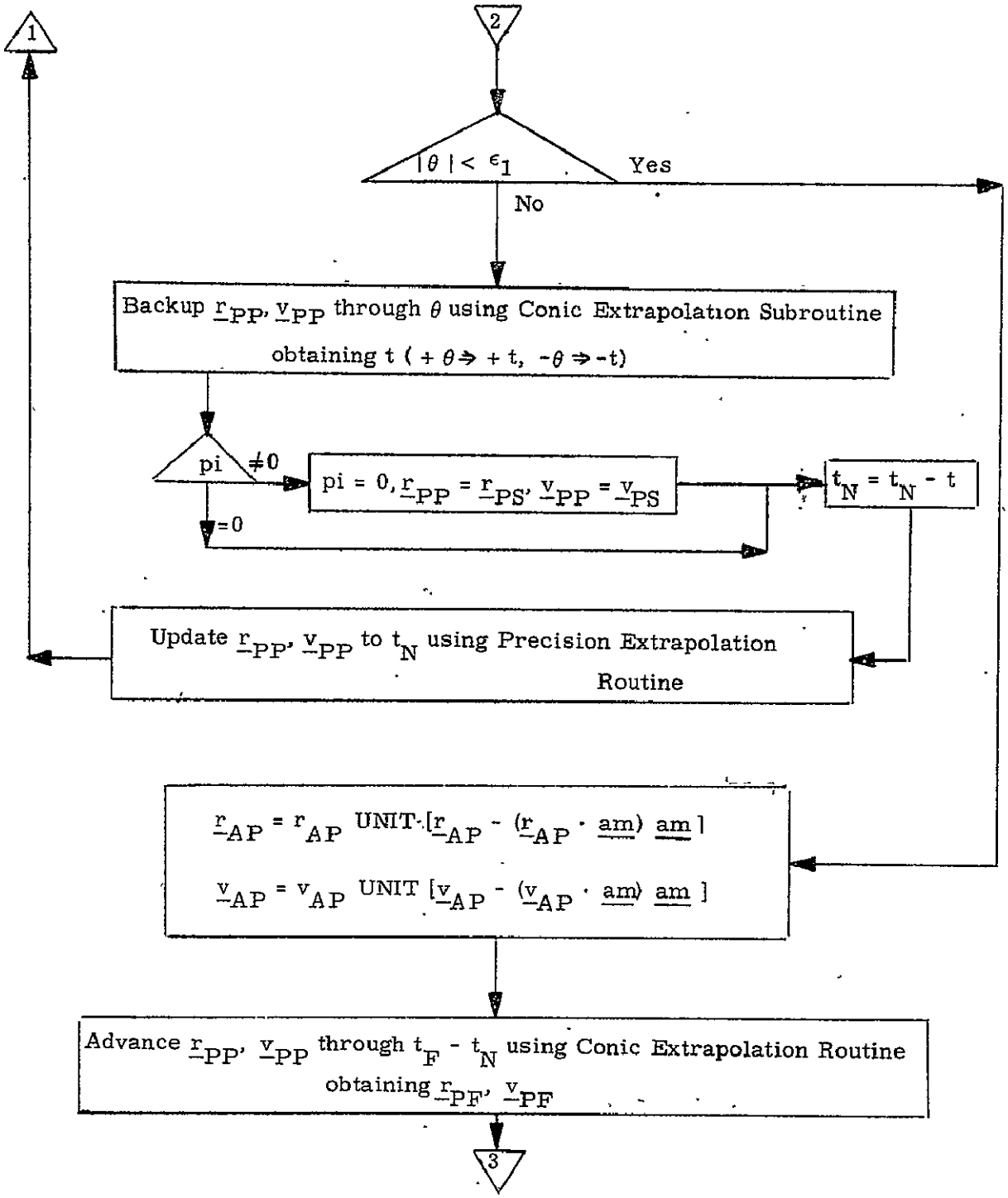


Figure 3.2.5 FMS, TMS Detailed Flow Chart
(page 2 of 6)

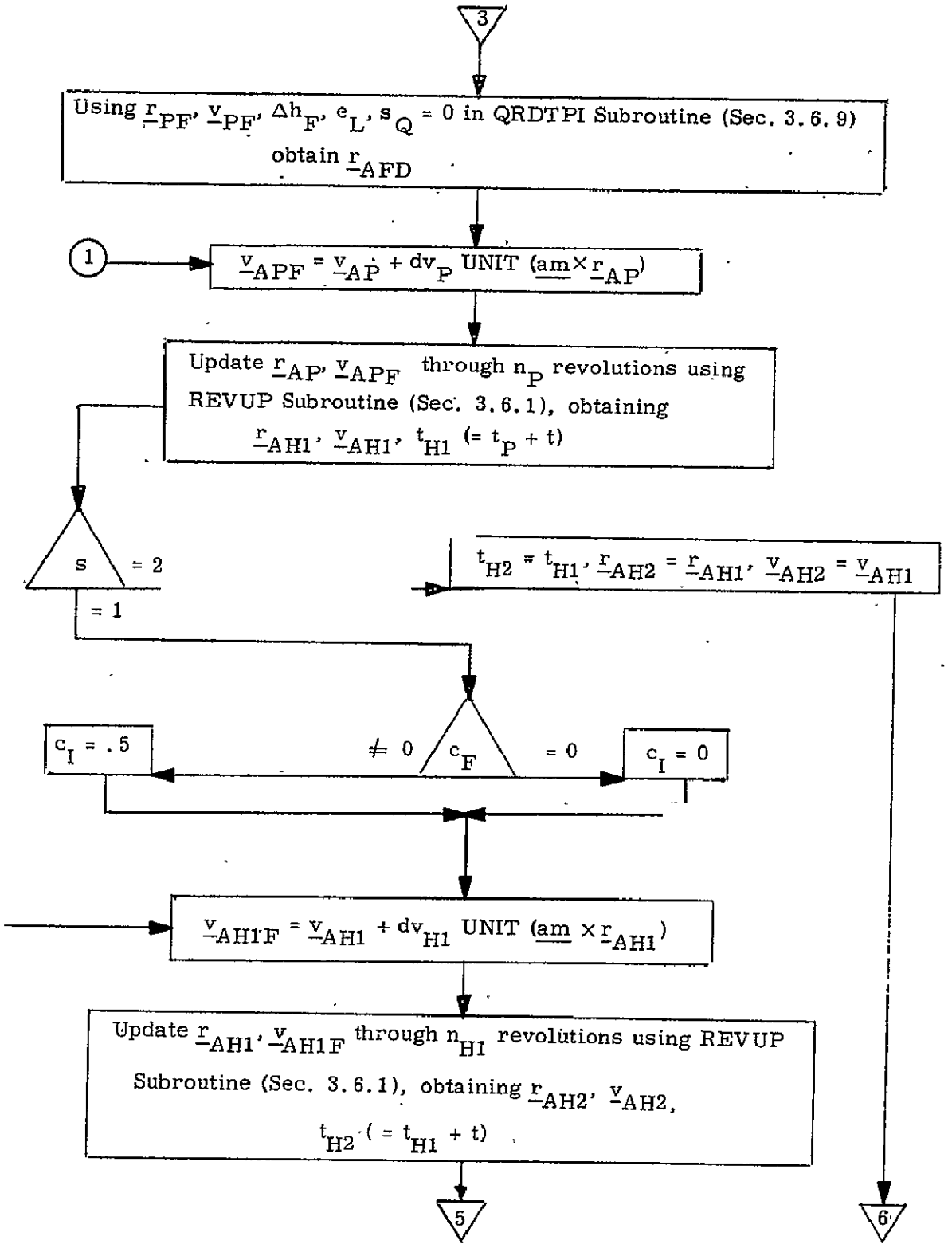


Figure 3.2.5 FMS, TMS Detailed Flow Chart

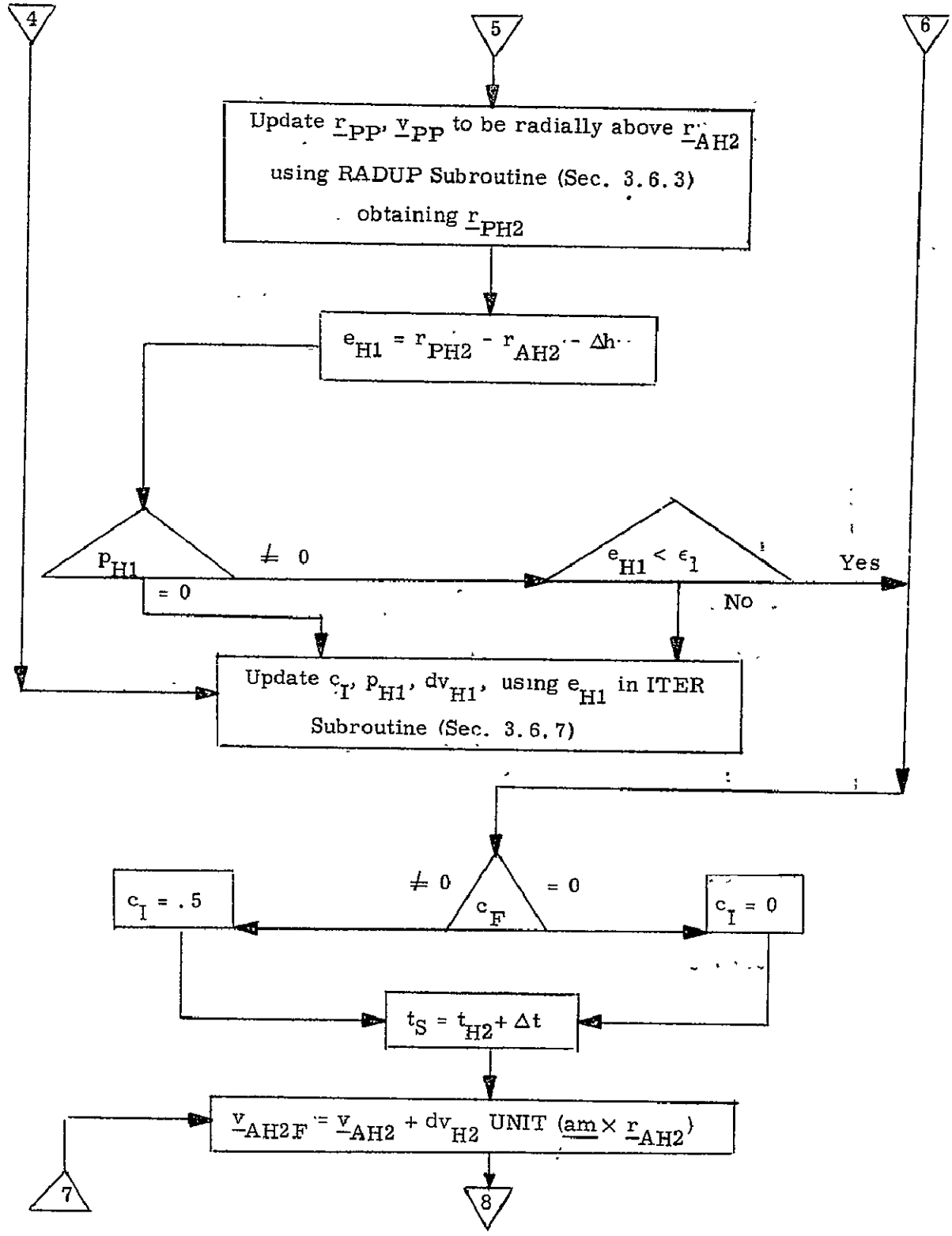


Figure 3.2.5 FMS, TMS Detailed Flow Chart

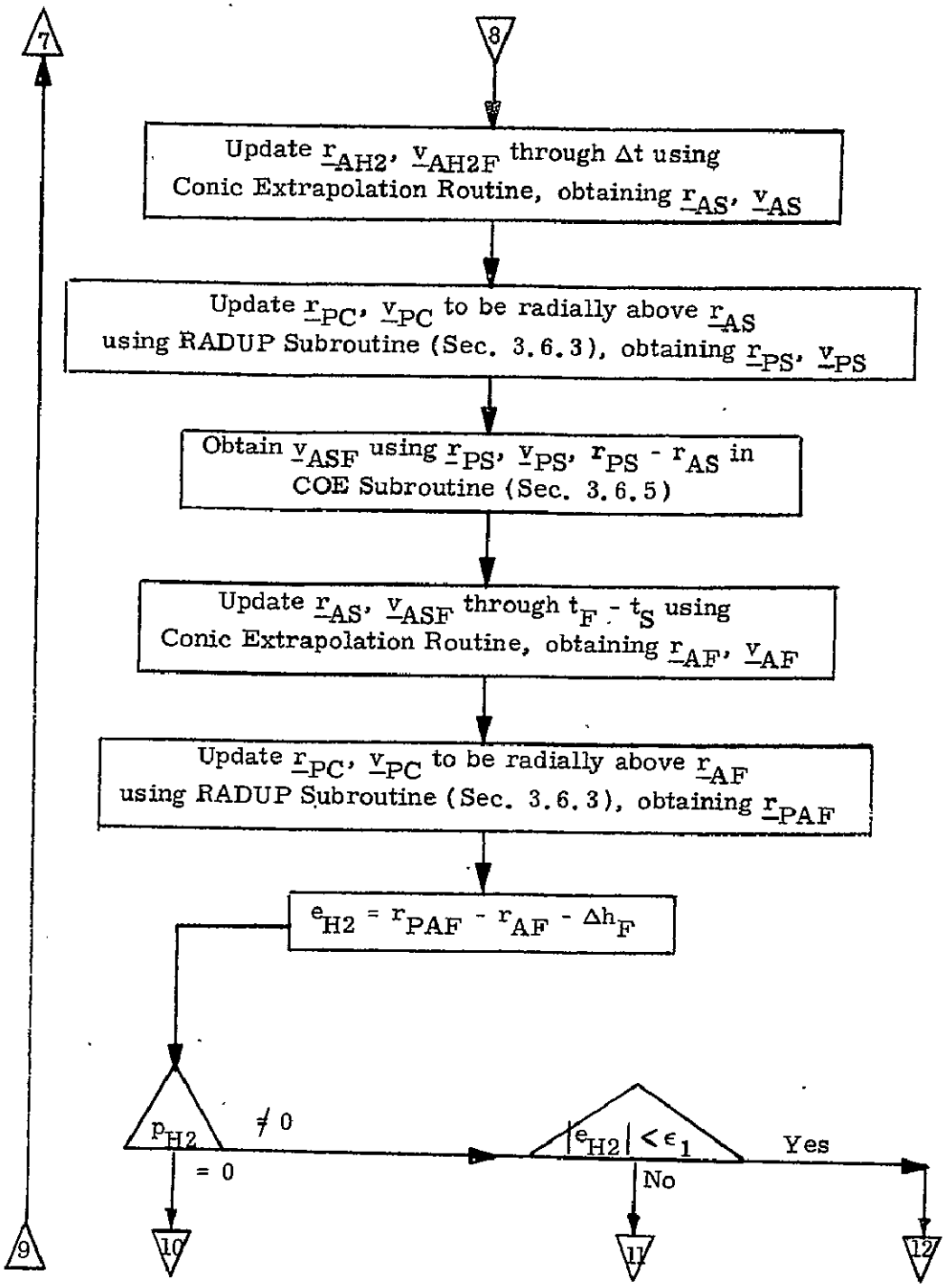


Figure 3.2.5 FMS, TMS Detailed Flow Chart

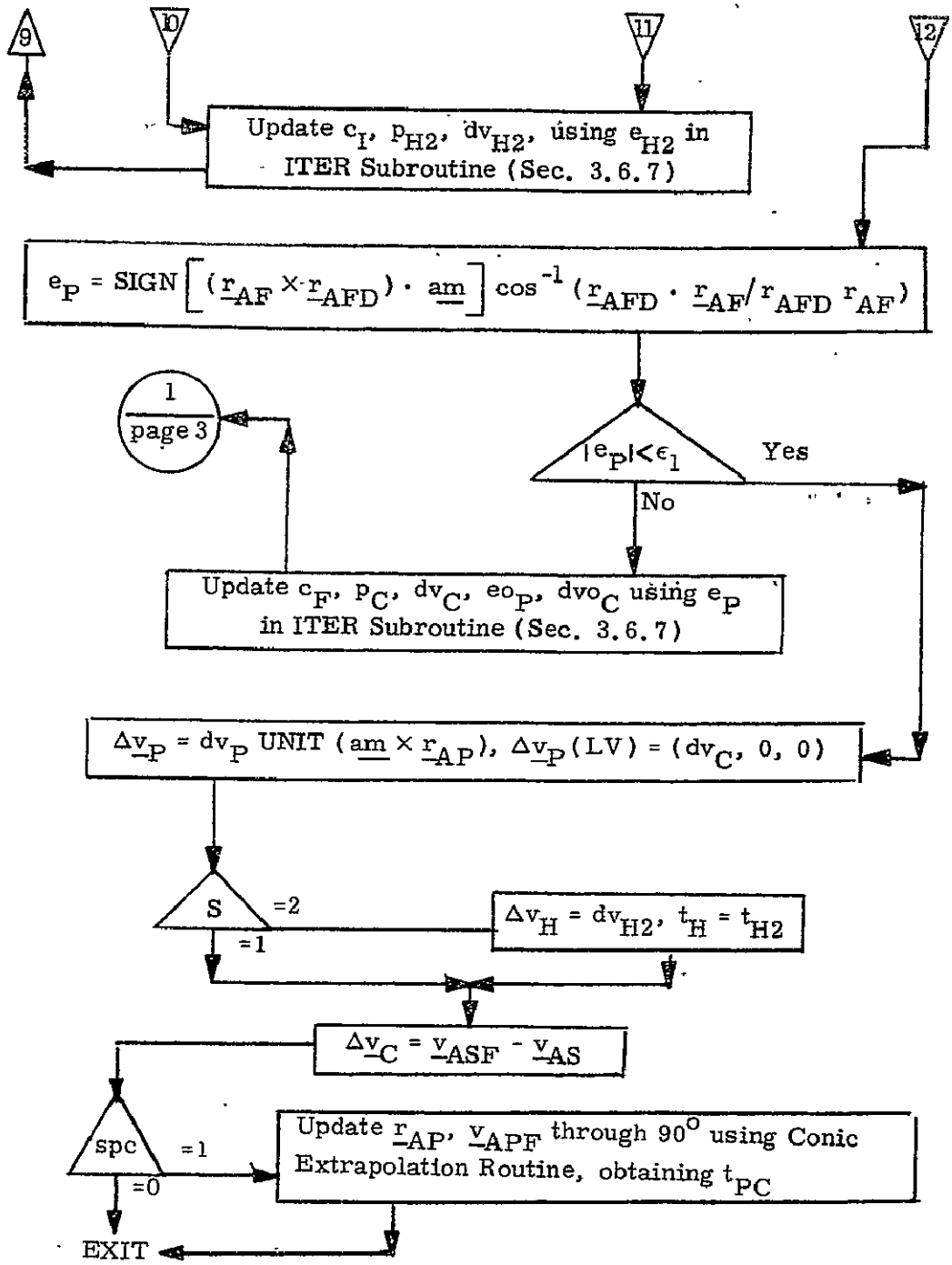


Figure 3.2.5 FMS, TMS Detailed Flow Chart

3.3 Triple Maneuver Sequence Program (TMS)

3.3.1 TMS Program Objectives

The objective of the Three Maneuver Sequence program is to compute a rendezvous trajectory which satisfies the constraints listed in Sec. 3.1. The program computes a phasing maneuver and the magnitudes of a height and a coelliptic maneuver.

This program contains an outer iterative loop to determine the phase maneuver and an inner loop to calculate the height maneuver. The program's logic is very similar to the FMS program, and is herein treated as a subset of the FMS program logic, as represented in the functional flow chart in Fig. 3.2.4.

3.3.2 TMS Program Input-Output

The inputs for the Three Maneuver Sequence are:

$\underline{r}_A, \underline{v}_A, \underline{r}_P, \underline{v}_P$	Active and passive state vectors
t_P	Time of the NP maneuver
Δt	Time between the NH and NC maneuvers
t_F	Time of the TPI maneuver
Δh_F	Altitude between the passive orbit and the active vehicle at the TPI time
e_L	Elevation angle
$S = 2$	FMS - TMS mode switch
n_P	Number of revolutions between the NP and NH maneuvers
spc	Plane change maneuver switch (= 1 for computation of t_{PC})

$\Delta v_P, \Delta v_H$	Nominal magnitudes of the phase and height maneuvers

The outputs to the TMS program are:

Δv_P	NP maneuver $\Delta \underline{v}$
$\Delta v_P(LV)$	NP maneuver $\Delta \underline{v}$ in local vertical coordinates
t_H	Time of the NH maneuver
t_{PC}	Time of the plane change maneuver (if required)
Δv_H	Magnitude of the NH maneuver
Δv_C	Magnitude of the NC maneuver

3.3.3 TMS Program Description

The detailed flow chart for Triple Maneuver Sequence program is incorporated in FMS flow chart in Fig. 3.2.4. As the TMS logic is just a subset of the logic of the FMS (i.e., it eliminates one of the height maneuvers in the FMS program), the reader is referred to Sec. 3.2.3 for the TMS program description.

3.4 Double Maneuver Sequence Program (DMS)

3.4.1 DMS Program Objectives

The objective of the Double Maneuver Sequence program is the computation of a Lambert maneuver based on a target vector obtained by determining a coelliptic orbit at the TPI point.

Fig. 3.4.4 is a functional flow chart of the DMS program.

3.4.2 DMS Program Input-Outputs

The required inputs to the double maneuver sequence program are:

$\underline{r}_A, \underline{v}_A, \underline{r}_P, \underline{v}_P$	Active and passive state vectors
t	Time of the Lambert maneuver
t_F	Time of the TPI maneuver
e_L	Elevation angle
Δh	Altitude between the passive orbit and tha active vehicle at the TPI time
Δt	Time between the Lambert and coelliptic maneuvers

The outputs to this program are:

$\Delta \underline{v}_L$	Lambert maneuver $\Delta \underline{v}$
Δv_C	Coelliptic maneuver Δv

3.4.3 DMS Program Description

Fig. 3.4.5 is a detailed flow chart of the Double Maneuver Sequence program.

The program starts with precision updates of the active and

passive vehicle's state vectors to t_L and t_F, respectively. The QRDTPI Subroutine is then called to obtain the passive vehicle's state $\underline{r}_{PF}$, $\underline{v}_{PF}$ radially above the desired active vehicle's position vector at TPI. Using the passive vehicle's state vector and the altitude difference Δh, the COE subroutine is called to obtain the a active vehicle's state $\underline{r}_{AF}$, $\underline{v}_{AF}$ prior to the TPI maneuver. This state vector is then integrated backwards using the Precision Extrapolation Routine to the time of the NC maneuver, obtaining the target vector $\underline{r}_{AT}$ for the Lambert maneuver. The Precision Lambert routine is next called to obtain the NL and NC 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 is obtained from this routine for use in the powered flight steering program as S_R. After calculating the NL and NC maneuvers, the NL maneuver is rotated into a local vertical coordinate system for display purposes.

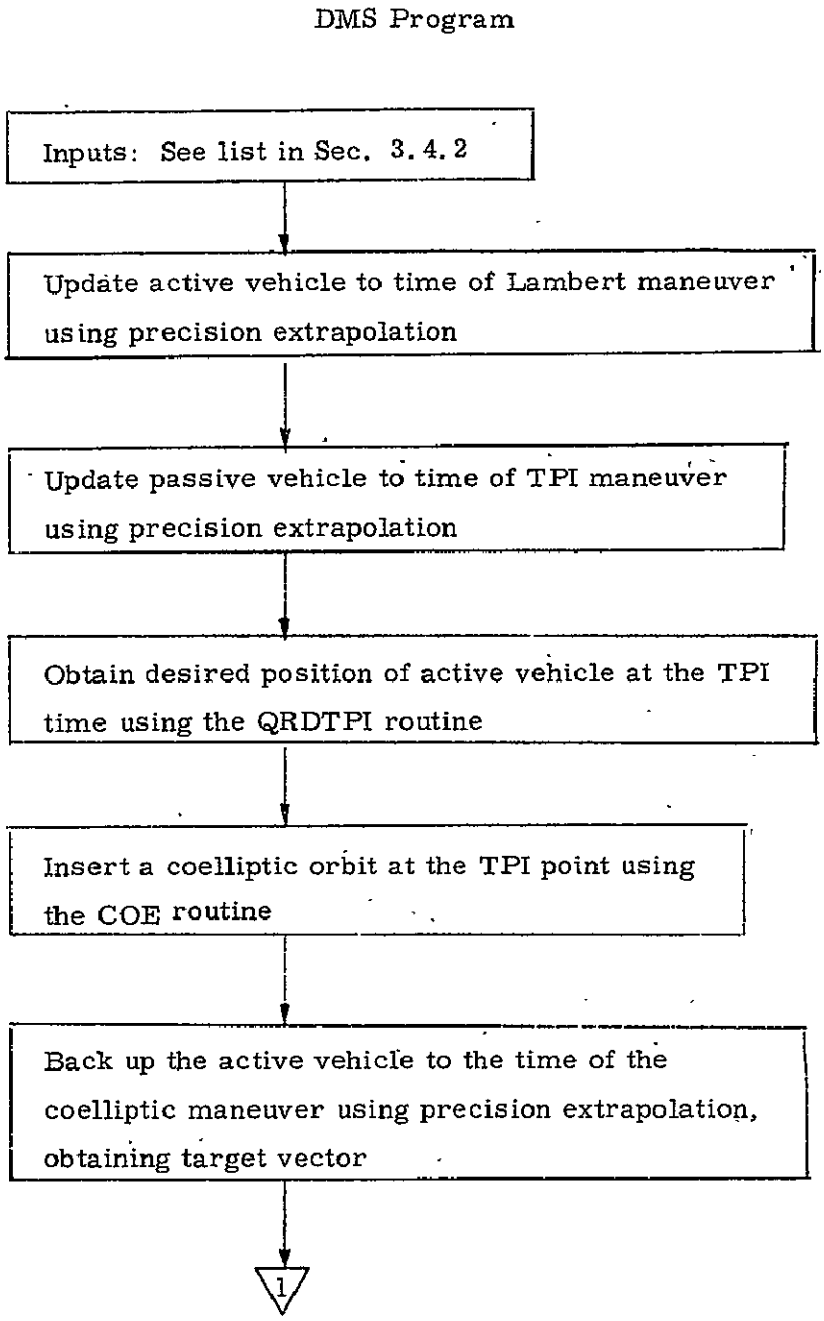


Fig. 3.4.4 DMS Functional Flow Chart

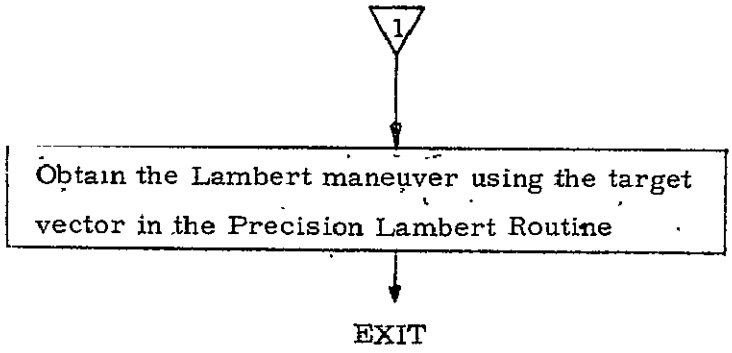


Fig. 3.4.4 DMS Functional Flow Chart

DMS Program

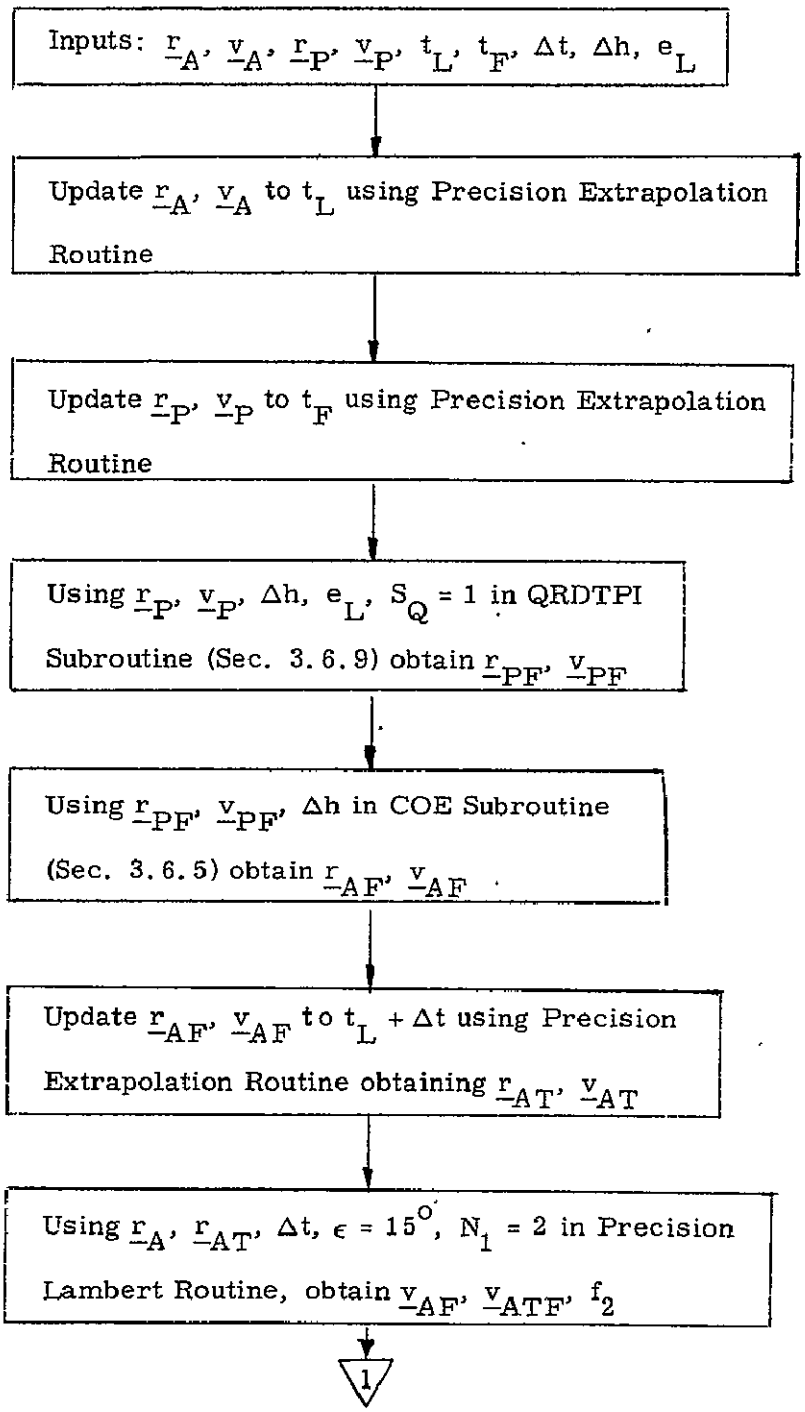


Fig. 3.4.5 DMS Detailed Flow Chart

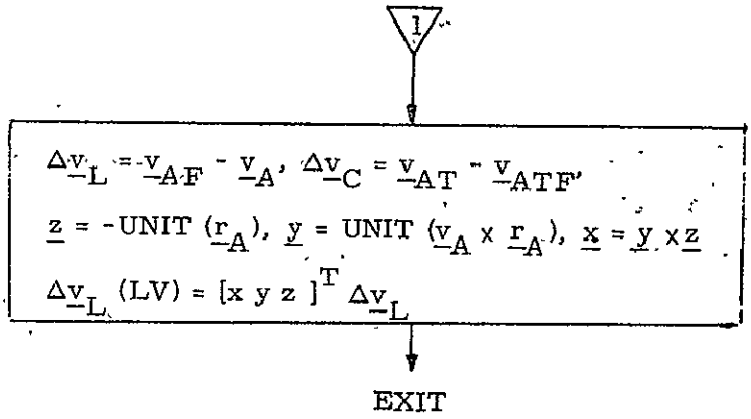


Fig. 3.4.5 DMS Detailed Flow Chart

3.5 Single Maneuver Sequence Program (SMS)

3.5.1 SMS Program Objectives

The objective of the Single Maneuver Sequence Program is the computation of the coelliptic maneuver (NC). Fig. 3.5.4 is a functional flow chart of the SMS program.

3.5.2 SMS Program Input-Outputs

The required inputs to the single maneuver sequence program are:

$\underline{r}_A, \underline{v}_A, \underline{r}_P, \underline{v}_P$	Active and passive state vectors
t_C	Time of the NC maneuver
t_F	Time of the TPI maneuver
e_L	Elevation angle

The outputs to the program are:

$\Delta\underline{v}_C$	Coelliptic manevuer $\Delta\underline{v}$

3.5.3 SMS Program Description

Fig. 3.5.5 is a detailed flow chart of the Single Maneuver Sequence program.

The program starts with precision updates of the active and passive vehicle's state vecotrs to time t_C. The out-of-plan parameters are next computed. Following rotating the active state into the plane of the passive orbit, the passive state is updated to be radially above the active vehicle's position vector at the NC point. Using the COE subroutine, the coelliptic maneuver is computed such that the resulting post maneuver velocity vector of the active vehicle will be parallel to the plane of the passive vehicle's orbit. After updating to the TPI time, the TPI program is used to obtain the TPI time which corresponds to a specified elevation angle.

Single Maneuver Sequence Program

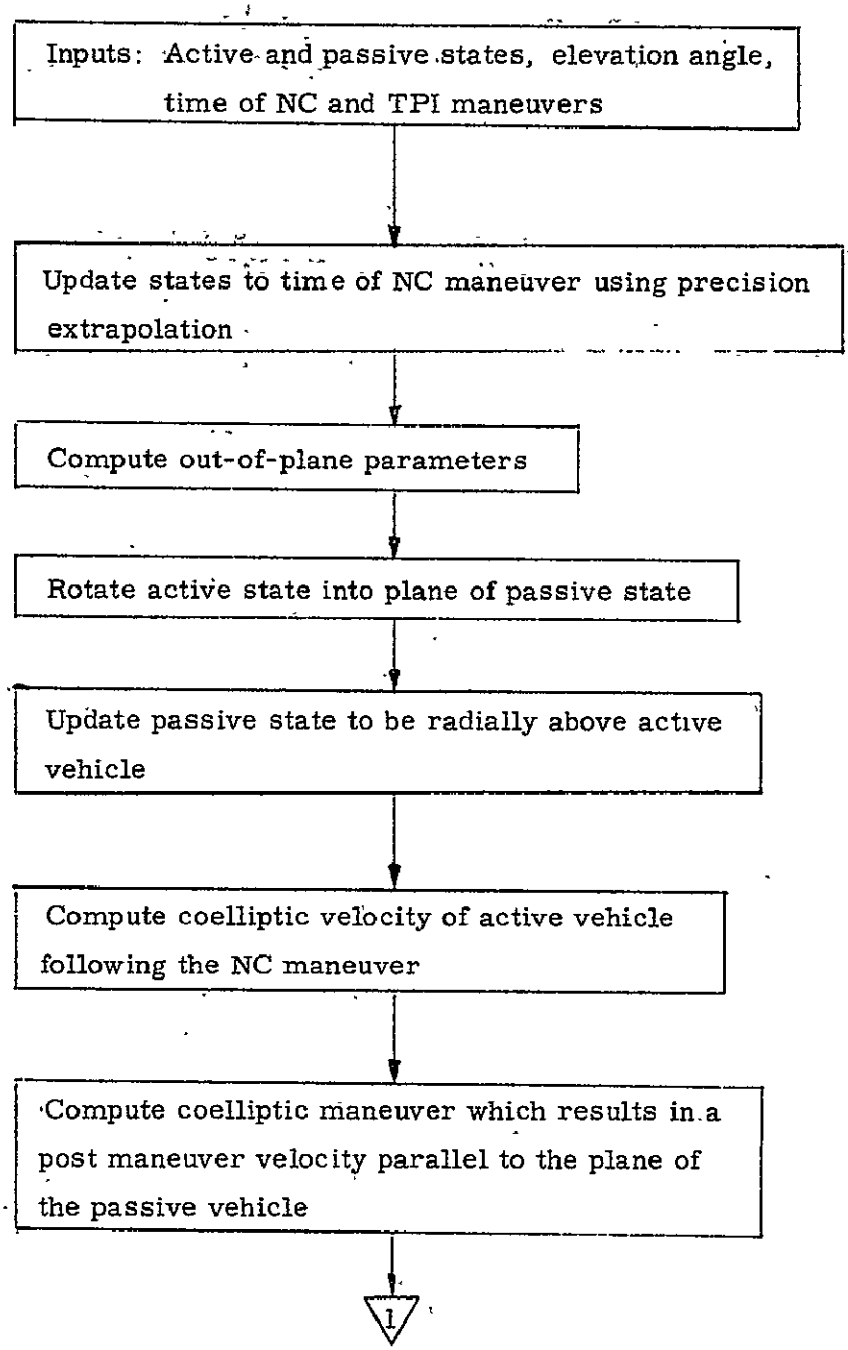


Fig. 3.5.4 SMS Functional Flow Chart

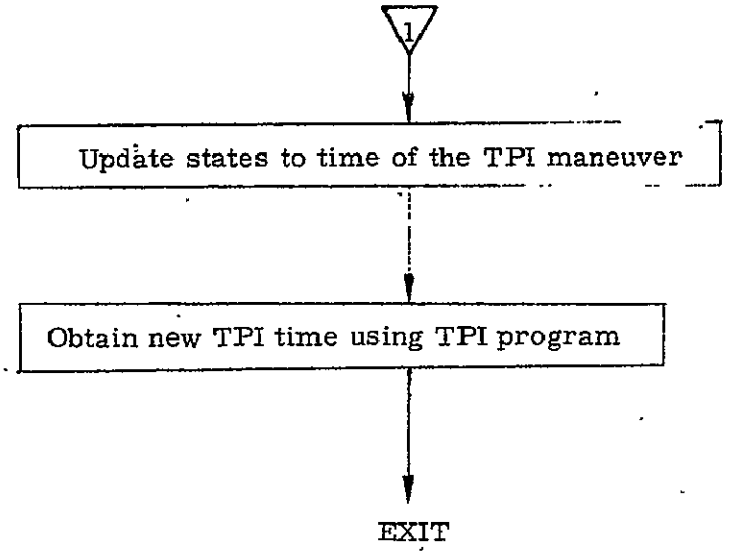


Fig. 3.5.4 SMS Functional Flow Chart

Single Maneuver Sequence Program

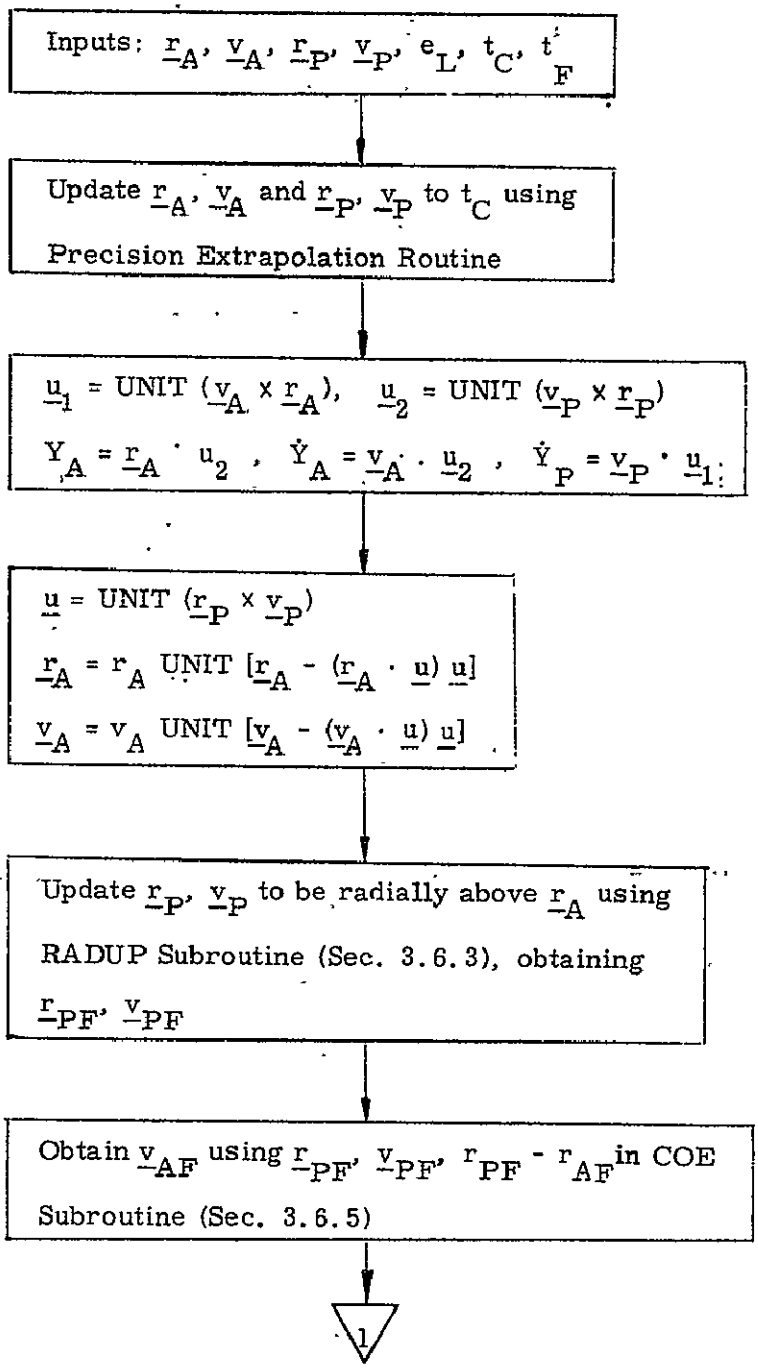


Fig. 3.5.4 SMS Detailed Flow Chart

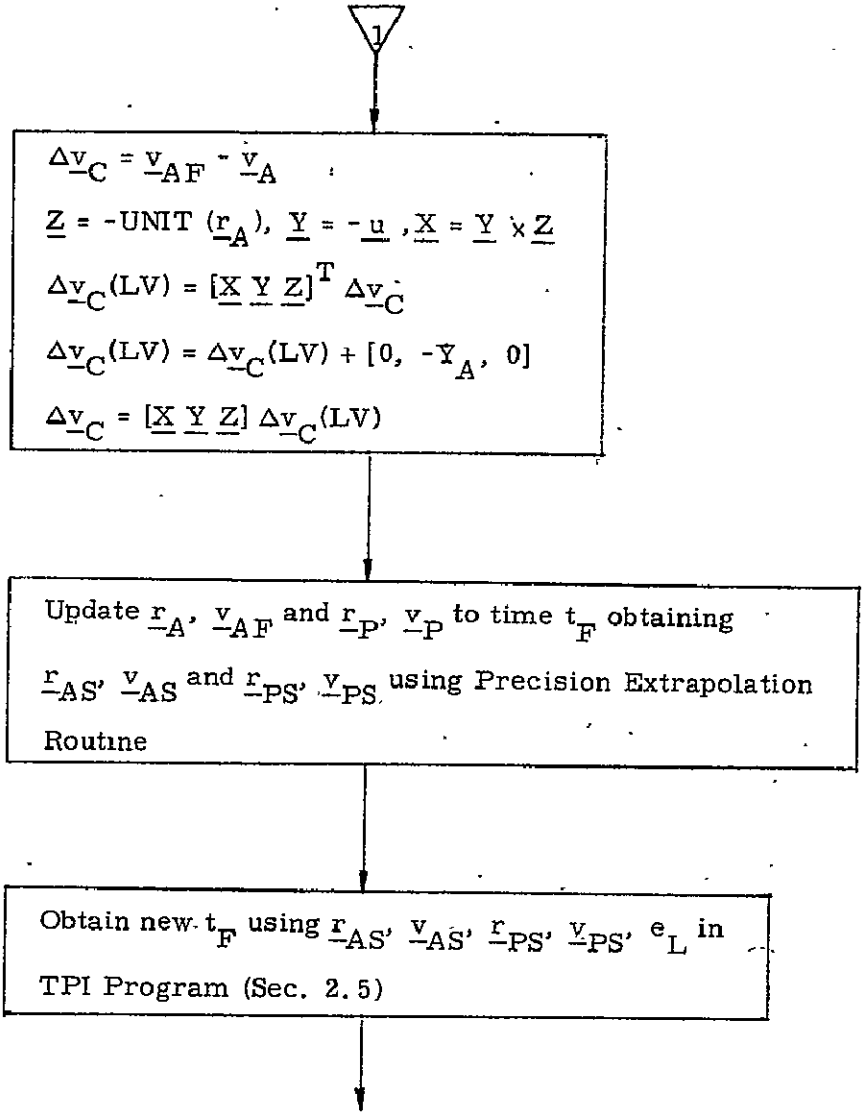


Fig. 3.5.5 SMS Detailed Flow Chart

3.6 Miscellaneous Subroutines Used in the FMS

This section contains a discussion of the special subroutines that are used in the FMS rendezvous targeting programs contained in Sections 3.2 - 3.5. As these subroutines are short and easy to follow, the input-outputs are sometimes only listed on the detailed flow charts and the objectives of the subroutines are contained in the description of the subroutine.

3.6.1 REVUP Subroutine

The REVUP Subroutine (Fig. 3.6.2) conically updates a state vector $\underline{r}$, $\underline{v}$ through a specified number of n of revolutions. The semi-major 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 concially updating the state through this time, the program returns the new state and the time.

3.6.3 RADUP Subroutine

The RADUP Subroutine (Fig. 3.6.4) conically updates the passive vehicle's state vector $\underline{r}_P$, $\underline{v}_P$ to be radially above the position of the active vehicle $\underline{r}_A$. The central angle θ between the two vehicles is initially calculated as a positive angle if the active vehicle leads the passive vehicle. The passive vehicle state is then updated through θ using the Time-Theta Subroutine to obtain the desired state.

3.6.5 COE Subroutine

The COE Subroutine (Fig. 3.6.6) computes the active vehicle's coelliptic state based on the state vector $\underline{r}$, $\underline{v}$ of the passive vehicle and a delta altitude h. After computing the semi-major axis a of the passive orbit, the desired semi-major axis a_D of the active vehicle's orbit is calculated. This is then used to compute the desired radial component of velocity v_V and the velocity vector $\underline{v}_A$ for the coelliptic orbit.

3.6.7 ITER Subroutine

The ITER Subroutine (Fig. 3.6.8) encompasses a Newton Raphson iteration scheme based on empirical partials. It utilizes as inputs:

1. Current v and old v_o values of the independent variable.
2. Current e and old e_o values of the dependent variable.
3. Estimate of the partial p = de/dt.
4. A switch c which can assume three values:

   c = 0 - First pass, p equals zero so increment v by one.

   c = .5 - First pass with non zero p, so compute change in v to be e/p.

   c .5 - Change v based on new partial computed from e, e_o, v and v_o.

In addition to computing a new estimate of the independent variable, this routine updates the quantities e_o, v_o, p and the iteration counter c. It also displays an Alarm Code whenever the counter exceeds 15.

3.6.9 QRDTPI Subroutine

The QRDTPI Subroutine (Fig. 3.6.10) determines the desired active vehicle's position vector $\underline{r}_A$ at the TPI time. The inputs to this subroutine are:

- $\underline{r}$, $\underline{v}$ - TPI passive vehicle's state vector.
- h - Altitude between the passive vehicle's orbit and the active vehicle at the TPI time.
- e - Elevation angle.
- s - Switch: 0 for coasting integration, 1 for conic update.

The subroutine utilizes the ITER Subroutine (Sec. 3.6.7) in an iterative search for the desired $\underline{r}_A$. Each pass through the iterative loop involves an update of the passive vehicle's state vector to the estimated time associated with the passive vehicle's position radially above $\underline{r}_A$. An error e_T is then calculated which represents the difference between the desired central angle between the active and passive position vectors at the TPI time and the actual central angle. This error is driven to be smaller than ϵ_1 using the Newton Raphson iterative scheme contained in the ITER Subroutine.

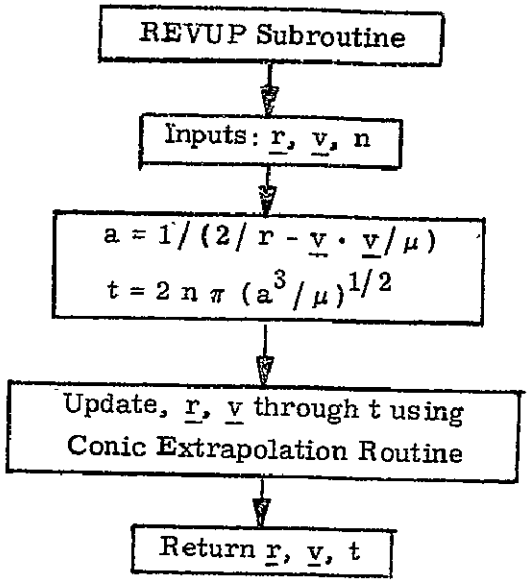


Figure 3.6.2 REVUP Subroutine Flow Chart

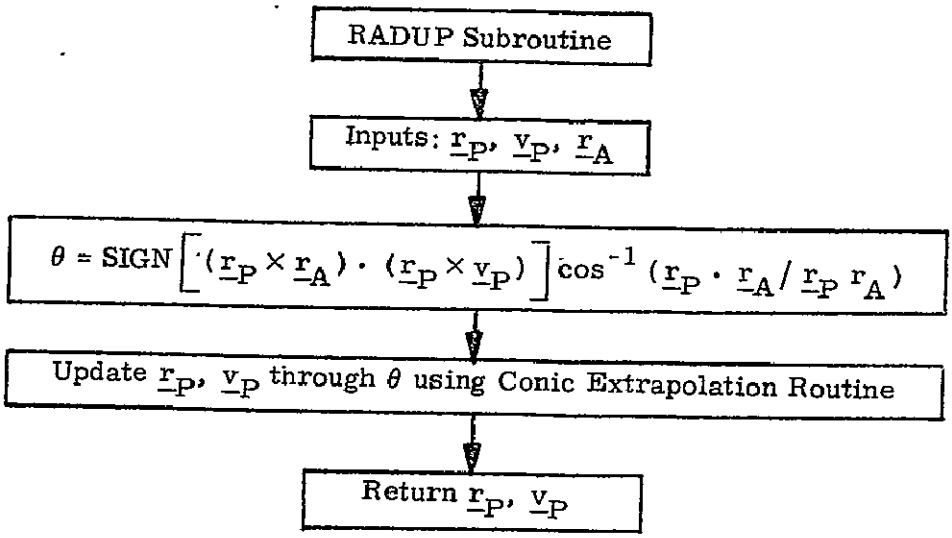


Figure 3.6.4 RADUP Subroutine Flow Chart

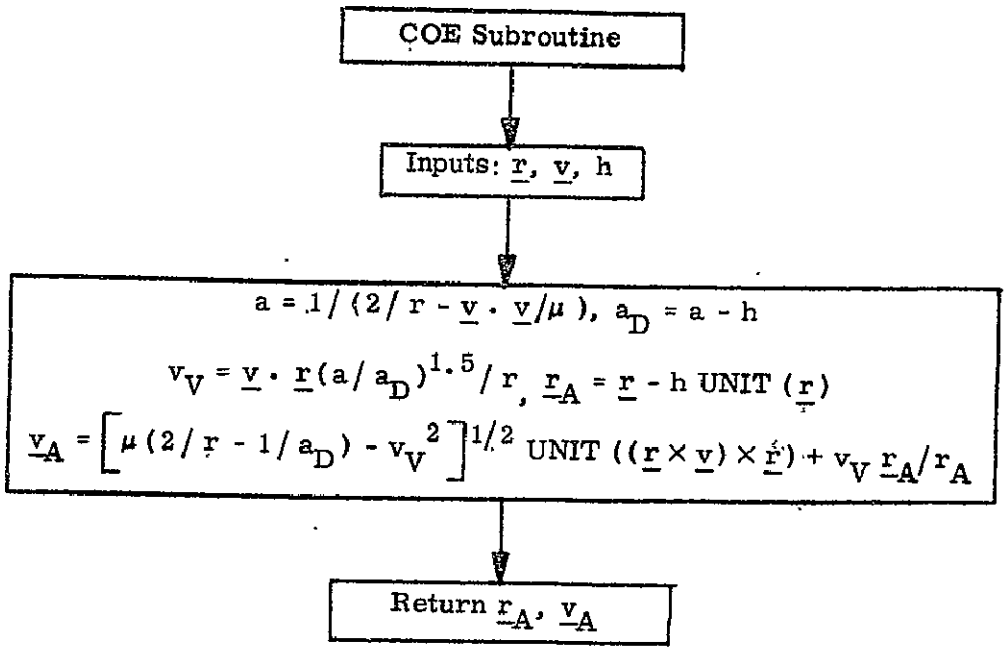


Figure 3.6.6 COE Subroutine Flow Chart

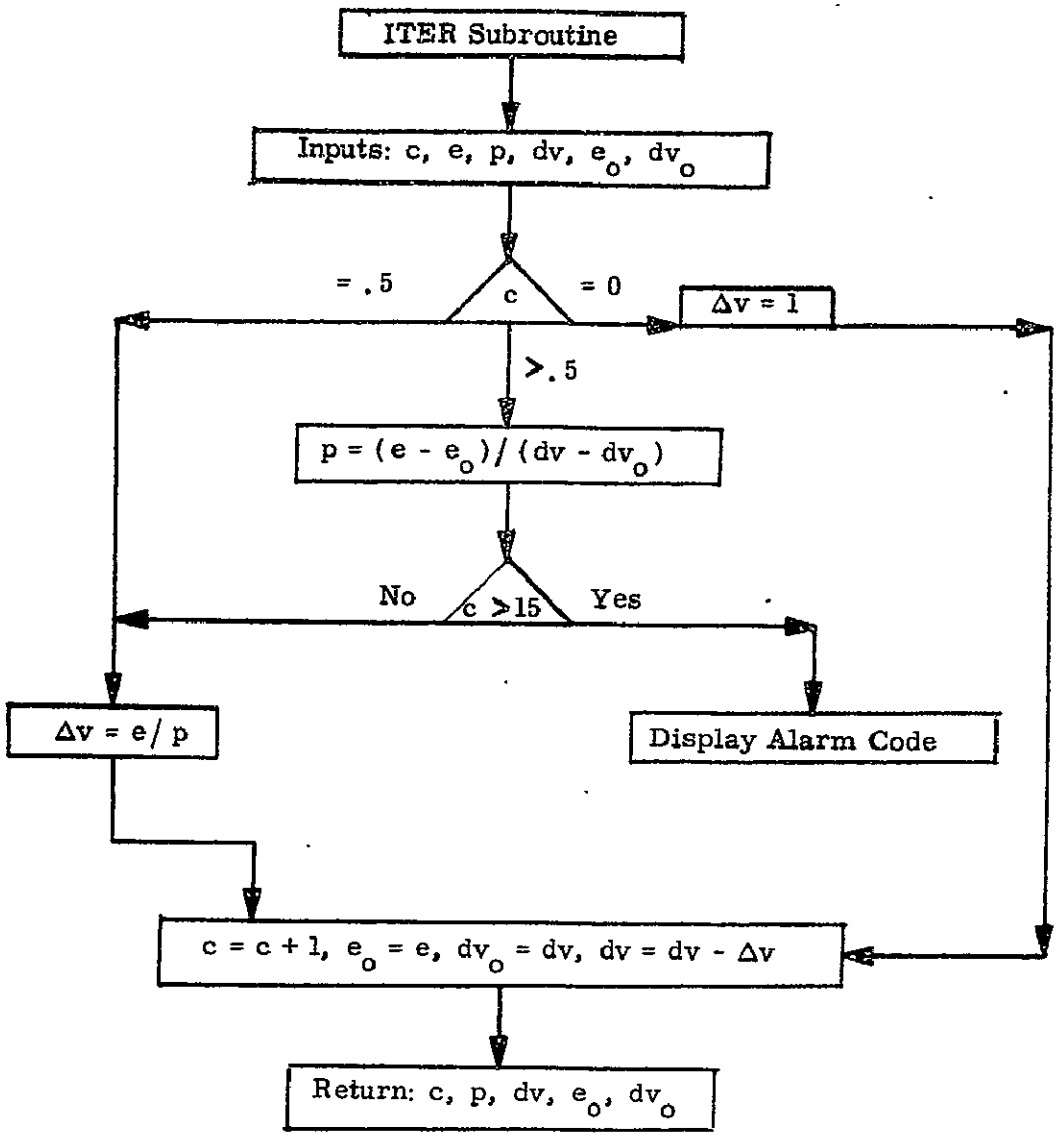


Figure 3.6.8 ITER Subroutine Flow Chart

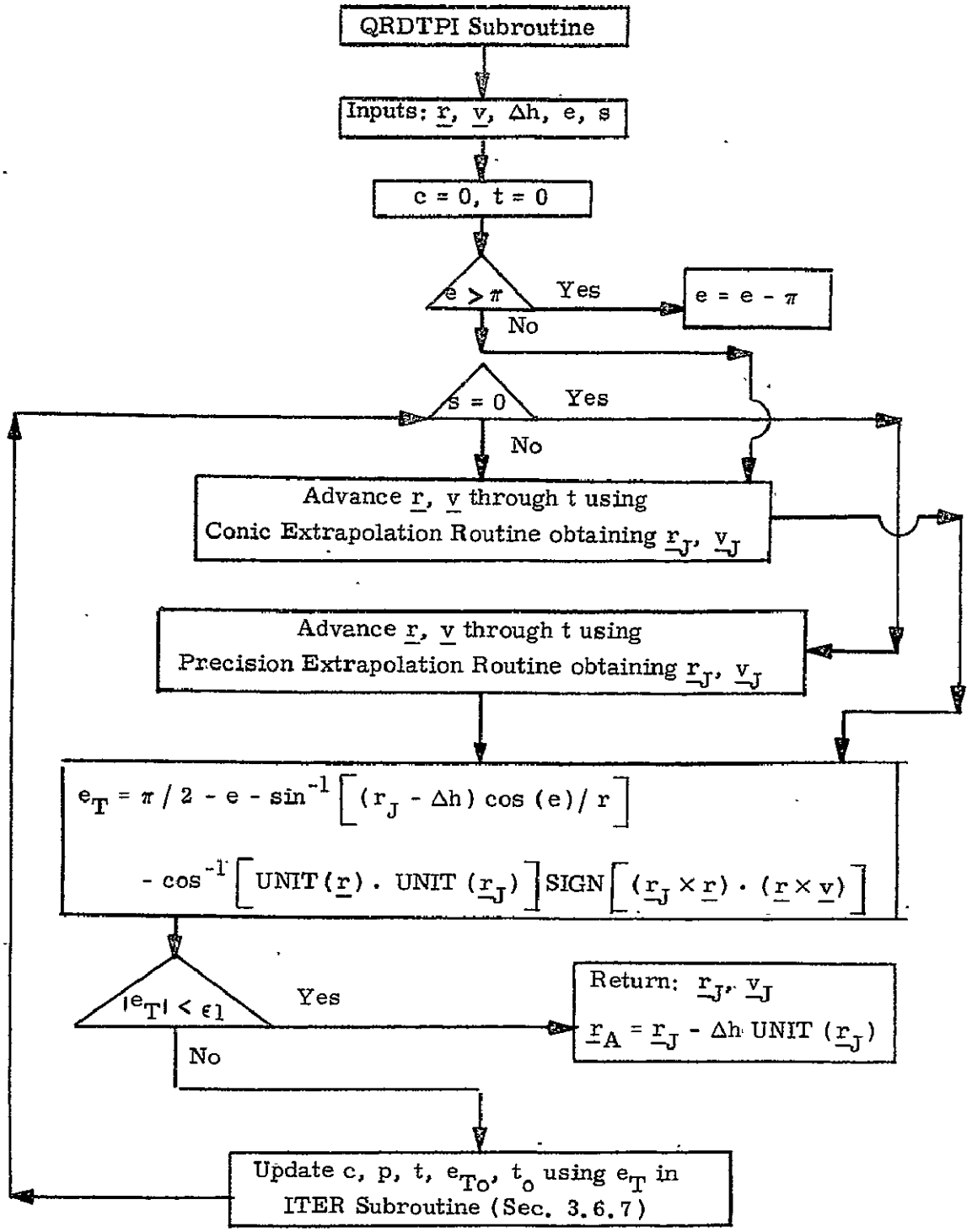


Figure 3.6.10 QRDTPI Subroutine Flow Chart

4. SUPPLEMENTARY INFORMATION

There are several steps that have to be completed prior to the selection of the preterminal rendezvous scheme. First, the mission and operational constraints that impinge on the rendezvous configuration have to be defined. These next have to be converted into a set of targeting constraints which relate to the rendezvous trajectory itself. (For example, if the TPI maneuver is to be aligned along the line of sight, it might be necessary to constrain the TPI elevation angle and altitude as well as the distance between TPI and TPF.) Upon completing this task, the minimum number of maneuvers involved in the preterminal targeting scheme will have been determined. This is because the rendezvous configuration must contain a sufficient number of maneuvers such that the corresponding degrees of freedom equals or exceeds the number of targeting constraints imposed. The number of degrees of freedom involved in an N maneuver rendezvous was originally defined in Ref. 1, and used to define the allowable number of Skylab targeting constraints in Ref. 2 and 3.

Assuming that the number of required targeting constraints derived from the mission constraints is less than the number of degrees of freedom involved in the rendezvous, additional constraints must be conceived to uniquely define the rendezvous configuration. Considerable work has been directed toward the problem of selecting rendezvous trajectories that minimize the fuel consumed (Ref. 4). Some very simple procedures might be used to accomplish this goal. For instance, maybe the specification of fractional orbital periods between maneuvers could be eliminated and the corresponding maneuver times optimized to conserve fuel. It would also conserve some fuel if the maneuvers were applied along the velocity vector instead of in the horizontal direction.

The TPI program (a carryover from Apollo and Skylab) contains an iterative search to determine the time that corresponds to a specified elevation angle. This iteration scheme fails in some cases where the height separation between the orbits becomes small (approximately four miles or less). It appears that a redesign of the iteration scheme would eliminate this possibility.

Most of the targeting constraints and types of maneuvers considered herein originated with NASA MSC.

The authors recommend investigating an approach whereby all the basic maneuvers are programmed as separate entities and are available for assembly into different rendezvous schemes as selected by the astronaut during the flight. Most of these maneuvers would contain the options of 1) phase match prior to the maneuver, 2) conic or precision extrapolations 3) generation of the plane change maneuver time and 4) astronaut overwrites. The storage requirements of such a general purpose rendezvous targeting program have to be determined. Hopefully, the increased requirements, which are not necessarily significant, will be offset by the advantage of the general program's flexibility.

5. REFERENCES

1. Tempelman, W., A Three Dimensional Analysis of Minimum Energy Intercepts, Transfers and Rendezvous Using N Impulses, ARS 15 th Annual Meeting, Washington, D.C., Preprint 1474-60, December 1960.

2. Tempelman, W. and Phillips, R., Basic Targeting Considerations for the Skylab Rendezvous Profile, 23A SKYLAB Memo #6-70, C.S. Draper Lab, November 5, 1970.

3. Tempelman, W. and Phillips, R., SKYLAB Rendezvous Targeting: Pre-NC1 and Pre-NC2 Programs, 23A SKYLAB Memo #5-70, C.S. Draper Lab, November 4, 1970.

4. Gobetz, F. and Doll, J., A Survey of Impulsive Trajectories, AIAA Journal, Vol. 7, No. 5, May 1969, pp. 801-834.

SPACE SHUTTLE

GN&C SOFTWARE EQUATION SUBMITTAL

Software Equation Section: Relative State Updating Submittal No. 20

Function: Compute Relative State of Orbiter with another Satellite

Module No. OG1 Function No. 2,5 (MSC03690)

Submitted by: E. S. Muller, R. E. Phillips Co. MIT No. 6-71

Date: Feb 1971

NASA Contact: J. Suddath Organization: GCD

Approved by Panel III: K. J. Cox Date: 3/10/71

Summary Description: Provides a means of automatically and autonomously improving the estimate of relative state between the Orbiter and another orbiting vehicle. This function may be required in a rendezvous mission or as an orbit navigation mode which utilizes the tracking of satellites.

Shuttle Configuration: Sensors are not specified.

Comments:

(Design Status)

(Verification Status)

Panel Comments:

9.8.2 Relative State Updating

1. INTRODUCTION AND FUNCTIONAL FLOW DIAGRAM

The purpose of the Relative State Updating function is to provide a means of automatically and autonomously improving onboard knowledge of the relative state between the SSV (primary vehicle) and another orbiting vehicle (target vehicle). This knowledge would be required in (a) rendezvous missions as inputs to rendezvous targeting programs to compute maneuvers which effect rendezvous between the primary and target vehicles or (b) orbit navigation modes which utilize tracking of navigation satellites or satellites ejected from the primary vehicle.

Rendezvous navigation sensor data, consisting of measurements of some portion of the relative state, are accepted at discrete "measurement incorporation times". Relative state updating is accomplished at each of these times by sequentially processing the components of the relative state measured by the sensor. A precision extrapolation routine extrapolates the primary and target vehicle state vectors and the filter weighting matrix from one "measurement incorporation time" to the next. A typical measurement incorporation sequence is thus:

(a) Extrapolate primary and target vehicle state vectors and filter weighting matrix to measurement incorporation time (t_m)

(b) Accept set of rendezvous navigation sensor data taken at time = t_m. This will consist of k components of the relative state at t_m given by Q_i ($i = 1, 2, \ldots, k$). A measurement code (c_i) is associated with each Q_i to identify the type of measurement taken.

(c) Process Q_1 in the Measurement Incorporation Routine. If more than one component of the relative state is being sensed, process Q_2, Q_3, ..., Q_k sequentially in the Measurement Incorporation Routine.

(a) thru (c) are then repeated for the next measurement incorporation time.

A general flow diagram of this function is presented in Fig. 1. The inputs required by this function are:

1. On-board estimate of primary vehicle state ($\underline{x}_P$) with time tag.
2. On-board estimate of target vehicle state ($\underline{x}_T$) with time tag.
3. Initial filter weighting matrix (W) (not required if computed using Automatic Initialization Routine).
4. A priori sensor measurement variances.
5. Rendezvous sensor measurements.

The output of this function is an updated n-dimensional state ($\underline{x}$) which minimizes the mean squared uncertainty in the estimate of the relative state. This output is available after each measurement incorporation.

The operations shown in Fig. 1, with the exception of precision extrapolation, belong in this function. The Precision Extrapolation Routine is described in another report. The bulk of the equations involved in this function are associated with the Measurement Incorporation Routine. The equations involved in an optional Automatic Initialization Routine (initializes the filter weighting matrix) will also be described. The equations associated with reading the rendezvous navigation sensor will be described in a later report.

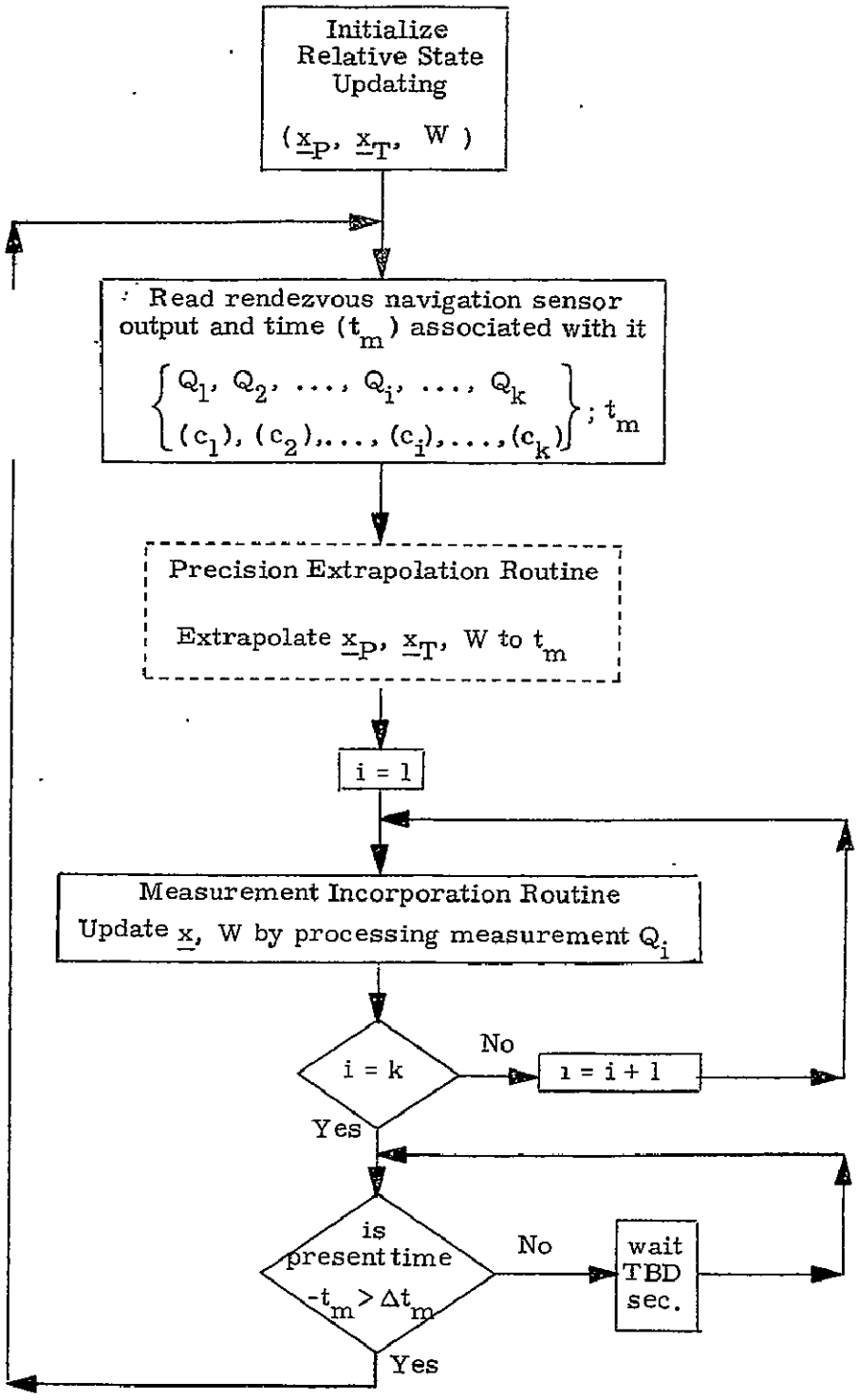


Figure 1 RELATIVE STATE UPDATING FLOW DIAGRAM

NOMENCLATURE

$\underline{b}$	n-dimensional measurement geometry
$\underline{b}_{P0}$, $\underline{b}_{P3}$	3 dimensional measurement geometry vectors associated with $\underline{r}_P$, $\underline{v}_P$
$\underline{b}_\gamma$	J-dimensional measurement geometry vector associated with γ_B
c_i	Measurement code identifying i th measurement at t_m
FIRSTMEAS	Initially set to "1" and reset to "0" after first entrance into Measurement Incorporation Routine
FULLTRACK	"1" if angles and relative range measurements have been taken prior to final intercept maneuver. "0" if angles or relative range measurements only have been taken prior to final intercept maneuver.
MANEUVER	Initially set to "0". Set to "1" at completion of maneuver
MANNOTRK1	Assumed "no track" time immediately prior to maneuver
MANNOTRK2	Assumed "no track" time immediately following maneuver
MANTM	Predicted time of next maneuver (either from preloaded input or previous targeting routine)
M_{NB-m}	Transformation matrix from navigation base axes to rendezvous sensor axes. M_{NB-m} is fixed according to spacecraft configuration.

M_{R-SM}	Transformation matrix from reference coordinate frame (in which initial state is expressed and computations are performed) to stable member axes. M_{R-SM} is given from specified platform alignment
M_{SM-NB}	Transformation matrix from stable member axes to navigation base axes on which IMU is mounted. M_{SM-NB} is determined from IMU gimbal angles
NOTRACKTM	Maximum break in tracking threshold - if time of "no track" period exceeds this, W reinitialization is inhibited until after 3 measurement incorporation times
POSTMANWR	Initially set to "0". If set to "1", forces W reinitialization prior to first mark after maneuver.
Q_{EST}	On-board estimate of measured parameter
Q_i	i th measured parameter at t_m
r_P	Magnitude of vector $\underline{r}_P$
$\underline{r}_P$	Primary vehicle position vector
$\underline{r}_T$	Target vehicle position vector
$\underline{r}_{TP}$	Relative position vector
$\underline{r}_U$	Position vector found in Automatic Initialization Routine (A.I.R)
RENDWFLAG	"0" - W is left as extrapolated value from Precision Integration routine (initially set to "0") "1" - W is set to pre-loaded value given by W_R
t_m	Measurement Incorporation Time
TBEFCOMP	Minimum time required prior to a final targeting computation to allow requested W reinitialization to be performed

TPIMAN	Initially set to "0". Set to "1" at completion of final intercept maneuver (TPI)
UNIT ($\underline{r}_P$)	Unit vector ($\underline{r}_P / r_P$)
UPD	= 1 find state of primary vehicle = 2 find state of target vehicle
$\underline{v}_P$	Primary vehicle velocity vector
$\underline{v}_T$	Target vehicle velocity vector
$\underline{v}_{TP}$	relative velocity vector
VAR	A priori random measurement error variance
$\underline{v}_U$	Velocity vector found in A.I.R.
W	n x n filter weighting matrix associated with $\underline{x}$
WAIT3TM	Initially set to "0" and reset to "1" in order to inhibit W reinitialization until after 3 measurement inforporation times
W_I	Pre-loaded value of initial filter weighting matrix
W_F	Pre-loaded value to which W is reinitialized
WMAXTM	Maximum threshold value - if time since last W reinitialization exceeds this, a W reinitialization is forced to occur prior to the first mark after the next maneuver
WRTM	Normal threshold value - if time since last W reinitialization exceeds this, a W reinitialization is requested

$\underline{x}$	n-dimensional state vector
$\underline{x}_P = \begin{vmatrix} \underline{r}_P \\ \underline{v}_P \end{vmatrix}$	6 dimensional primary vehicle state vector
$\underline{x}_T = \begin{vmatrix} \underline{r}_T \\ \underline{v}_T \end{vmatrix}$	6 dimensional target vehicle state vector
$\underline{x}_U = \begin{vmatrix} \underline{r}_U \\ \underline{v}_U \end{vmatrix}$	6 dimensional state vector found in A.I.R.
Δt_m	Time increment between measurement incorporation times
$\delta \underline{x}$	n-dimensional navigation update of $\underline{x}$
$\underline{\epsilon}_P$	A priori standard deviation of stable member misalignment
$\underline{\epsilon}_T$	A priori standard deviation of misalignment between sensor measurement frame and navigation base
η_r, η_β, η_θ	A priori standard deviation of sensor bias errors in range, gimbal angles β and θ (Fig. 2)
$\underline{\gamma}_B$	j dimensional sensor bias vector
σ_r, σ_β, σ_θ	A priori standard deviation of sensor random errors in range, β, θ
μ	Gravitational constant

2. DESCRIPTION OF EQUATIONS

The recursive navigation equations presented in the Measurement Incorporation section are general with respect to the dimension of the state vector to be updated. These equations are therefore applicable to any one of the following navigated state vectors which is selected for the shuttle relative state updating function. (This selection will ultimately be based on shuttle G& N computer capacity, expected target vehicle state uncertainties, and the error characteristics of shuttle navigation sensors.)

Table I

Possible Navigated States

Navigated State ($\underline{x}$)	Parameters Updated	State dimensions (n)
A. $\underline{x} = \underline{x}_P$ or $\underline{x}_T$	primary vehicle state or target vehicle state	6
B. $\underline{x} = \begin{pmatrix} \underline{x}_P \text{ or } \underline{x}_T \\ \underline{\gamma}_B \end{pmatrix}$	primary or target vehicle state plus j components of sensor bias	6+j
C. $\underline{x} = \begin{pmatrix} \underline{x}_P \\ \underline{x}_T \end{pmatrix}$	primary and target vehicle states	12
D. $\underline{x} = \begin{pmatrix} \underline{x}_P \\ \underline{x}_T \\ \underline{\gamma}_B \end{pmatrix}$	primary and target vehicle states plus j components of sensor bias	12+j

For any of these navigated state vectors, the relative state does not appear directly, but is updated implicitly as a result of the update of either or both vehicle inertial states. Utilizing the state vectors (A or B) results in the Apollo rendezvous navigation filter, whereas either state vector (C or D) results in an optimum rendezvous navigation filter. Specifying the dimension (n) of the navigated state vector automatically specifies the dimension of the measurement geometry vector $\underline{b}$ to be (n), and the filter weighting matrix W to be n x n.

2.1 Measurement Incorporation Routine

As discussed above, this routine is entered k (number of measured components from sensor) times at each measurement incorporation time (t_m). The equations presented below are identical for incorporation of each of these components with the exception of equations for $\underline{b}$, Q_{EST} and VAR which depend on the component incorporated. Equations for $\underline{b}$, Q_{EST}, and VAR are given for typical relative measurement parameters and bias estimation, since the precise parameters will not be known until the rendezvous sensor (s) are selected. The assumed sensor coordinate frame geometry is shown in Fig. 2. (Gimbal limits are assumed to be between $\pm$ 90°).

The precedure for computing $\underline{b}$, Q_{EST} and VAR is as follows:

(1) Compute the relative state ($\underline{x}_R$) from:

$$\underline{x}_R = \begin{pmatrix} \underline{R}_{TP} \\ \underline{V}_{TP} \end{pmatrix} = \underline{x}_T - \underline{x}_P$$

and

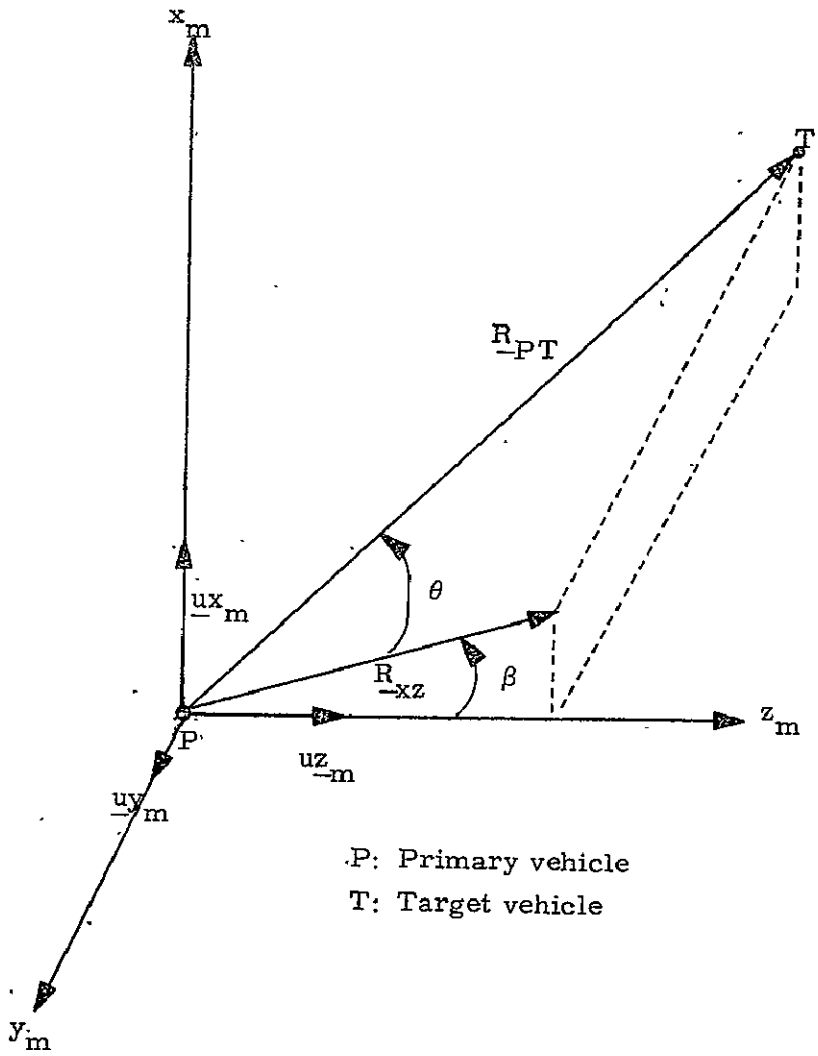


Figure 2 RENDEZVOUS SENSOR COORDINATE FRAME GEOMETRY

$$\underline{UR}_{TP} = \text{UNIT}\,(\underline{R}_{TP})$$

From the measurement code (c_i), compute $\underline{b}$, Q_{EST} and VAR appropriate to this measurement.

For sensor gimbal angle (β, θ) measurements, make the following preliminary computations:

(1a) Compute the unit vectors of the sensor coordinate frame $\underline{ux}_m$, $\underline{uy}_m$, $\underline{uz}_m$ from:

$$\begin{pmatrix} \underline{ux}_m^T \\ \underline{uy}_m^T \\ \underline{uz}_m^T \end{pmatrix} = M_{NB-m}\, M_{SM-NB}\, M_{R-SM}$$

(1b) Compute $\sin(\theta)$, R_{xz} (Fig. 1) from:

$$S = -\,\underline{uR}_{TP} \cdot \underline{uy}_m$$

and

$$R_{xz} = R_{PT}\sqrt{1 - S^2}$$

Computation of $\underline{b}$

Depending on the ultimate selection of the navigated state ($\underline{x}$ of Table I), the vector $\underline{b}$ will take on the following definitions:

If:

$$\underline{x} = \underline{x}_P \quad , \underline{b} = \underline{b}_P = \begin{pmatrix} \underline{b}_{P0} \\ \underline{b}_{P3} \end{pmatrix}$$

$$\underline{x} = \underline{x}_T \quad , \underline{b} = -\underline{b}_P$$

$$\underline{x} = \begin{pmatrix} \underline{x}_P \\ \underline{\gamma}_B \end{pmatrix}, \underline{b} = \begin{pmatrix} \underline{b}_{P0} \\ \underline{b}_{P3} \\ \underline{b}_{\gamma} \end{pmatrix}$$

$$\underline{x} = \begin{pmatrix} \underline{x}_T \\ \underline{\gamma}_B \end{pmatrix}, \underline{b} = \begin{pmatrix} -\underline{b}_{P0} \\ -\underline{b}_{P3} \\ \underline{b}_{\gamma} \end{pmatrix}$$

$$\underline{x} = \begin{pmatrix} \underline{x}_P \\ \underline{x}_T \end{pmatrix} , \underline{b} = \begin{pmatrix} \underline{b}_P \\ -\underline{b}_P \end{pmatrix}$$

$$\underline{x} = \begin{pmatrix} \underline{x}_P \\ \underline{x}_T \\ \underline{\gamma}_B \end{pmatrix} , \underline{b} = \begin{pmatrix} \underline{b}_P \\ -\underline{b}_P \\ \underline{b}_{\gamma} \end{pmatrix}$$

(2) Compute $\underline{b}_{P0}$ and $\underline{b}_{P3}$ from the appropriate equations in the following table using c_i to identify the type of measurement

Measurement	$\underline{b}_{P0}$	$\underline{b}_{P3}$
Relative Range	$-u\underline{R}_{TP}$	$\underline{0}$
Range Rate	$u\underline{R}_{TP} \times (u\underline{R}_{TP} \times \underline{V}_{TP})/R_{TP}$	$-u\underline{R}_{TP}$
Sensor Angle (β)	$UNIT(u\underline{R}_{TP} \times \underline{uy}_m)/R_{xz}$	$\underline{0}$
Sensor Angle (θ)	$(u\underline{R}_{TP} \times \underline{uy}_m) \times u\underline{R}_{TP}/R_{xz}$	$\underline{0}$

(3) Compute $\underline{b}_{\gamma}$

If $\underline{\gamma}_B$ is included in the navigated state, $\underline{b}$ will be computed based on the selection of bias parameters to be estimated. The following are equations for some possible $\underline{b}_{\gamma}$'s, with c_i used to identify the measurement type

(a) Estimating a single bias (γ_B) in measurement code = j): For measurement (code = c_i)

$$b_\gamma(\text{scalar}) = \begin{cases} 1 & c_i = j \\ 0 & c_i \neq j \end{cases}$$

(b) Estimating bias (γ_B) in m of the total of k measurements, the measurement codes of the m measurements being: 1, 2, ..., m:

$$\underline{b}\ (\text{m-vector}) = \underset{c_i=1}{\begin{pmatrix} 1 \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix}}\ ;\ \underset{c_i=2}{\begin{pmatrix} 0 \\ 1 \\ . \\ . \\ . \\ 0 \end{pmatrix}}\ \ldots\ \underset{c_i=m}{\begin{pmatrix} 0 \\ 0 \\ . \\ . \\ . \\ 1 \end{pmatrix}}$$

$\underline{b}_\gamma = \underline{0}$ for $c_i > m$

(c) Estimating three angles (α_x, α_y, α_z) of the stable member misalignment about x, y, z axes of stable member, i.e.

$$\underline{\gamma}_B = \begin{pmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{pmatrix}$$

Measurement	$\underline{b}_\gamma$ (3 dimensional)
Relative Range	$\underline{0}$
Range Rate	$\underline{0}$
Sensor Angle (β)	$R_{PT}/R_{xz}\left[M_{R\text{-}SM}(u\underline{R}_{TP} \times (u\underline{R}_{TP} \times u\underline{y}_m))\right]$
Sensor Angle (θ)	$M_{R\text{-}SM}(\text{UNIT}(u\underline{R}_{TP} \times u\underline{y}_m))$

Computation of Q_{EST}

(4) Compute estimate of bias ($\hat{\gamma}$) in measurement form:

$$\hat{\gamma} = \underline{b}_{\gamma} \cdot \hat{\underline{\gamma}}_B$$

($\underline{\gamma}_B$ is initially set to $\underline{0}$ and attains a value after measurement incorporation for a state ($\underline{x}$) which contains $\underline{\gamma}_B \cdot \hat{\underline{\gamma}}_B$ will be the corresponding components of $\delta\underline{x}$ in Eq. (11)).

(5) Compute Q_{EST} from the appropriate equation identified by the measurement code (c_i):

Measurement	Q_{EST}
Relative Range	$R_{TP} + \hat{\gamma}$
Range Rate	$\underline{V}_{TP} \cdot \underline{uR}_{TP} + \hat{\gamma}$
Sensor Angle (β)	$\mathrm{Tan}^{-1}\left(\frac{\underline{uR}_{TP} \cdot \underline{ux}_m}{\underline{uR}_{TP} \cdot \underline{uz}_m}\right) + \hat{\gamma}$
Sensor Angle (θ)	$\mathrm{Sin}^{-1}(s) + \hat{\gamma}$

Computation of VAR

Equations for VAR can not be anticipated as easily as was done for $\underline{b}$ and Q_{EST} since it is so strongly a function of the error model for the particular rendezvous sensor selected for the final configuration. The measurement variance can be a constant or some function of relative range, range rate, etc and it may have a minimum threshold. Consequently, equations for VAR will not be given at this time.

State Vector and Filter Update at Measurement Incorporation Time

The n × n filter weighting matrix (W) is available from one of the following sources:

At the first measurement incorporation:

1. Pre-loaded values based on mission simulations
2. As an output of the Automatic Initialization Routine

Between measurement incorporations at a given t_m:

3. From the computation (below) after a measurement incorporation

At the first measurement incorporation of new t_m:

4. From the Precision Extrapolation Routine*
5. From the Automatic Reinitialization Routine

(6) Compute n-dimensional $\underline{z}$ vector for measurement (c_i) from:

$$\underline{z} = W^T \underline{b}$$

(7) Compute n-dimensional weighting vector $\underline{\omega}$ from:

$$\underline{\omega} = \frac{1}{\underline{z} \cdot \underline{z} + VAR} W\underline{z}$$

*This routine provides the state and bias portions of W when time invariant biases are modeled. For estimation of biases modeled as time variant, appropriate equations in the Precision Extrapolation Routine will be formulated.

(8) Compute n-dimensional navigation update of $\underline{x}$ for measured parameter Q_i (code = c_i) from:

$$\delta\underline{x} = \underline{\omega}(Q_i - Q_{EST})$$

(9) Update $\underline{x}$ by:

$$\underline{x} = \underline{\hat{x}} + \delta\underline{x}$$

(10) Update W by:

$$W = W - \underline{\omega}\,\underline{z}^T / \left(1 + \sqrt{\frac{VAR}{\underline{z} \cdot \underline{z} + VAR}}\right)$$

2.2 Automatic Filter Weighting Matrix (W) Reinitialization

If reinitialization of the filter weighting matrix is required (e.g. if navigated states A or B of Table I are utilized), this operation may be accomplished automatically by the Automatic Reinitialization Routine. This routine consists almost entirely of logic statements so that there is no real need to present a description of equations here. Instead, the detailed description of the routine will be provided by the detailed flow diagrams, and a brief description of the approach will be given in this section.

A conservative approach is taken in that W is reinitialized to pre-stored values more often than actually required but not at a time which would violate accepted W matrix reinitialization ground rules. The only exception to this is the case in which not reinitializing will most probably produce a greater performance degradation than a reinitialization. The ground rules which prohibit reinitialization are:

1. No reinitialization unless a minimum time (TBEFCOMP) exists prior to the final targeting computation for a rendezvous maneuver.

2. No reinitialization following a "no tracking" interval greater than NO TRACKTM seconds, until after 3 measurement incorporation times.

The only exception occurs when a maximum time has passed without a reinitialization (WMAXTM) because of (1) or (2). In this case a reinitialization is forced to occur immediately following a rendezvous maneuver (representing a "no track" interval) instead of waiting the required 3 measurement incorporation times as specified by (2).

2.3 Automatic Initialization Routine

2.3.1 Introduction

This routine provides a means for computing an initial filter weighting matrix for recursive navigation which is closely related to the actual errors in the computed relative state. Two position fixes are required. The equations described relate to the problem of finding the inertial state of one vehicle given in the inertial position of the other and the relative position of the two.

The routine might be used if the inertial state of the primary vehicle is poorly known. That is, the estimate of the relative state is so bad that the (linear) recursive navigation filter does not converge. This situation might arise, for instance, when (sensor) acquisition does not occur until the range between the vehicles is of the same order of magnitude as the relative error between them.

2.3.2 Program Input-Output

The required inputs to the routine are two sets of sensor measurements at t_1 and t_2, and two inertial positions at t_1 and t_2. Also required are various assumed values for instrument performance to be used in forming the W matrix.

r_1, β_1, θ_1 } sensor measurements at t_1

r_2, β_2, θ_2 } sensor measurements at t_2

$M_{R\text{-}SM}$, $M_{SM\text{-}NB}$ (at t_1 and t_2), $M_{NB\text{-}m}$ } rotation matrices

$\underline{r}_{T1}$ } known inertial position of target at t_1

$\underline{r}_{T2}$ } known inertial position of target at t_2

σ_r, σ_β, σ_θ } a priori standard deviation of sensor random measurement errors

η_r, η_β, η_θ } a priori standard deviation of bias in sensor measurement

$$\begin{bmatrix} \alpha_T \\ \beta_T \\ \gamma_T \end{bmatrix} = \underline{\epsilon}_T$$ a priori standard deviation of misalignment between sensor measurement frame and navigation base

$$\begin{bmatrix} \alpha_P \\ \beta_P \\ \gamma_P \end{bmatrix} = \underline{\epsilon}_P$$ a priori standard deviation of stable member misalignment

The output of the program is $\underline{x}_P$ (or $\underline{x}_T$) at t_2 and an n x n W matrix to use in relative state updating.

2.3.3 Description of Equations

The following equations are in two parts, computing the state of the unknown vehicle and computing the related covariance and W matrix. The first set of equations uses two position "fixes" to solve Lambert's problem for the velocity connecting the positions.

Calculation of the State

Let r_1, β_1, θ_1 and r_2, β_2, θ_2 be the measurements made by the sensor at the times t_1 and t_2. Find the cartesian vector $\underline{r}_{TPS\ 1}$ in the sensor frame shown in Fig. 2.

$$r_{TPS\ 1,0} = r_1 \cos\theta_1 \sin\beta_1$$
$$r_{TPS\ 1,1} = -r_1 \sin\theta_1$$
$$r_{TPS\ 1,2} = r_1 \cos\theta_1 \cos\beta_1$$

Using the same relations define $\underline{r}_{TPS\ 2}$. Transform the vector $\underline{r}_{TPS\ 1}$ from sensor frame to reference frame.

$$\underline{R}_{TP\ 1} = M^{T}_{R\text{-}SM},\ M^{T}_{SM\text{-}NB\ 1}\ M^{T}_{NB\text{-}m}\ \underline{r}_{TPS\ 1}$$

Similarly define the vector $\underline{R}_{TP\ 2}$ from (r_2, β_2, θ_2). Depending on the value of logic switch, UPD, extrapolate either the primary or target vehicle to the times of the two fixes t_1 and t_2. Using these two inertial positions the two relative positions, and the time interval $\Delta t = t_2 - t_1$ find the velocity $\underline{v}_{U\ 2}$ at t_2 via the Lambert Routine. The six-dimensional state vector $\underline{x}_{U\ 2}$ at t_2 is:

$$\underline{x}_{U\ 2} = \begin{bmatrix} \underline{r}_{U\ 2} \\ \\ \underline{v}_{U\ 2} \end{bmatrix}$$

From two position measurements it is impossible to estimate any bias, so those components, if included in the state, are set to zero.

Calculation of the W-Matrix

In rendezvous navigation it is the relative state which is measured and used to update either (or both) of the inertial vectors. Associated with the relative state is the relative covariance matrix. As an example the W matrix for a 9-dimensional state including constant sensor bias is computed.

The error in the relative state is due to errors in the sensor measurements r, β, and θ and to errors in the transformation matrices, $M_{R\text{-}SM}$, $M_{SM\text{-}NB}$, $M_{NB\text{-}m}$. The measured quantities r_m, β_m, and θ_m include noise $\underline{\sigma}$ and bias $\underline{\eta}$.

$$\underline{\sigma} \equiv (\sigma_r, \sigma_\beta, \sigma_\theta)$$
$$\underline{\eta} \equiv (\eta_r, \eta_\beta, \eta_\theta)$$

Errors in the transformation matrices are due to stable member misalignment and structural deformation between the sensor and the navigation base. These errors in the two matrices $M_{R\text{-}SM}$ and $M_{NB\text{-}m}$ are expressed as $\underline{\epsilon}_P$ and $\underline{\epsilon}_T$. These pseudo-vectors represent one standard deviation small rotations about three othogonal axes. From the values of $\underline{\sigma}$, $\underline{\eta}$, $\underline{\epsilon}_T$ and $\underline{\epsilon}_P$ two diagonal 9 x 9 matrices SIG and GAM are formed. It should be noted that η_r^2, η_β^2 and η_θ^2 appear in SIG if each of those components of bias is to be estimated (as is done here), otherwise they appear in GAM.

Combinations of several 3 x 3 partial derivative matrices make up a 9 x 9 matrix relating state error to the matrix SIG. Those component matrices will now be computed.

The partial derivative matrix of relative position error in the sensor frame due to error in r, β and θ, $DRDM_S$, is computed by simply taking the necessary derivatives of the geometric relations:

$$r_{k,0} = r_k \cos\theta_k \sin\beta_k$$
$$r_{k,1} = -r_k \sin\theta_k$$
$$r_{k,2} = r_k \cos\theta_k \cos\beta_k$$

Combined with transformation matrices the partial derivative matrices allow the partial derivative matrix of relative position error in the inertial frame to be written:

$$DRDM_k = M_{R\text{-}SM}^T \, M_{SM\text{-}NB\,k}^T \, M_{NB\text{-}m}^T \, DRDM_{S\,k}$$

9.8.2 Relative State Updating (cont'd)

The dependence of the velocity deviations $\delta \underline{v}_{U\,2}$ on the two position errors must be computed. The two matrices $DVDR_2$ and $DVDR_1$ are derivatives of Lambert's solution for the velocity at the second point. They may be computed from values of semi-major axis, $1/\alpha$, eccentric anomalies, E_k, and S and C (Battin's special transcendental functions) found in the Lambert Routine.

α = reciprocal of semi-major axis

$x = (E_1 - E_2)/\sqrt{\alpha}$ (E = eccentric anomaly)

$y = x^2 / c$

$S(\alpha x^2)$, $C(\alpha x^2)$ } Battin's special transcendental functions

Using these variables and the following definitions proceed:

$r_1 = |\underline{r}_1|$

$r_2 = |\underline{r}_2|$

$\hat{r}_1 = \text{UNIT}(\underline{r}_1)$

$\hat{r}_2 = \text{UNIT}(\underline{r}_2)$

$q = \sqrt{2 - \alpha x^2 c}$

$D_S = (c - 3S)/(2\alpha x^2)$

$D_C = (1 - \alpha x^2 S - 2C)/(2\alpha x^2)$

$D_Q = -\sin(\sqrt{\alpha x^2})/(4 q \sqrt{\alpha x^2})$

$Q_d = (3 x S y D_C)/(2C^2) - x^3 D_S$

$F = \sqrt{\mu / y / \alpha}$

$H = y / r_1 - 1$

$$G = F H$$

$$Y_c = F / r_1 - G / 2y$$

$$A_c = - G / \alpha$$

$$Q_y = \alpha / (2\sqrt{y}) + 3 S x / (2C)$$

$$D = Q_d + Q_y \alpha D_Q$$

$$\nabla_1 \alpha = (\hat{r}_1 + \hat{r}_2) r_2 / 2\alpha$$

$$\nabla_2 \alpha = (\hat{r}_1 + \hat{r}_2) r_1 / 2\alpha$$

$$\nabla_1 \alpha x^2 = ((\sqrt{y} - Q_y q) \nabla_1 \alpha + Q_y \hat{r}_1) / D$$

$$\nabla_2 \alpha x^2 = ((\sqrt{y} - Q_y q) \nabla_2 \alpha + Q_y \hat{r}_2) / D$$

$$\nabla_1 y = \hat{r}_1 - q \nabla_1 \alpha - \alpha D_Q \nabla_1 \alpha x^2$$

$$\nabla_2 y = \hat{r}_2 - q \nabla_2 \alpha - \alpha D_Q \nabla_2 \alpha x^2$$

$$\nabla_1 F = - F (\nabla_1 \alpha / \alpha + \nabla_1 y / 2y)$$

$$\nabla_2 F = - F (\nabla_2 \alpha / \alpha + \nabla_2 y / 2y)$$

$$\nabla_2 G = Y_c \nabla_2 y + A_c \nabla_2 \alpha$$

$$\nabla_1 G = Y_c \nabla_1 y + A_c \nabla_1 \alpha - F y \hat{r}_1 / r_1^2$$

$$DVDR_2 = G I + \hat{r}_2 \nabla_1 F^T + \hat{r}_1 \nabla_1 G^T$$

$$DVDR_1 = F I + \hat{r}_2 \nabla_2 F^T + \hat{r}_1 \nabla_2 G^T$$

(I is the 3 dimensional identity matrix)

Combined with $DRDM_1$ and $DRDM_2$ the above matrices yield the 9 x 9 partial derivative matrix relating the state to the matrix of sensor random and bias errors.

$$DVDM_1 = DVDR_1 \; DRDM_1$$
$$DVDM_2 = DVDR_2 \; DRDM_2$$
$$DVDE = DVDM_1 + DVDM_2$$

$$DSDS_{(9\times9)} = \begin{bmatrix} DVDM_2 & 0 & DRDM_2 \\ DVDM_2 & DVDM_1 & DVDE \\ 0 & 0 & I \end{bmatrix}$$

A second 9 x 9 partial matrix relates state errors to unestimated sensor bias and the two misalignments. The additional needed 3 x 3 component matrices are computed now.

The matrix $DRDE_{TS\,1}$ relates position error at the first "fix" in the sensor frame to misalignment between sensor and navigation base

$$DRDE_{TS\,1} = \begin{bmatrix} 0 & r_{TPS\,k,2} & -r_{TPS\,k,1} \\ -r_{TPS\,k,2} & 0 & r_{TPS\,k,0} \\ r_{TPS\,k,1} & -r_{TPS\,k,0} & 0 \end{bmatrix}$$

This matrix rotated into the reference frame is:

$$DRDE_{T\,1} = M^T_{R\text{-}SM} \; M^T_{SM\text{-}NB\,1} \; M^T_{NB\text{-}m} \; DRDE_{TS1}$$

In the same way $DRDE_{T\,2}$ is computed.

The matrices relating stable member misalignment to error in position in the reference frame $DRDE_P$ is computed in the same way. It consists of a matrix composed of elements of the relative position vector in the reference frame $\underline{r}_{TP}$.

Using the chain rule allows the computation of the matrices relating velocity to the two misalignments:

$$DVDE_T = DVDR_2\ DRDE_{T\,2} + DVDR_1\ DRDE_{T\,1}$$

$$DVDE_P = DVDR_2\ DRDE_{P\,2} + DVDR_1\ DRDE_{P\,1}$$

The dependence of estimated bias in r, β and θ on the two misalignments is given by the following two matrices.

$$DBDE_T = DRDM_{S\,1}^{-1}\ DRDE_{TS\,1}$$

$$DBDE_P = DRDM_{S\,1}^{-1}\ M_{NB\text{-}m}\ M_{SM\text{-}NB\,1}\ M_{R\text{-}SM}\ DRDE_{TS\,1}$$

The complete 9 x 9 partial matrix is thus:

$$DSDG = \begin{bmatrix} DRDM_2 & DRDE_{T\,2} & DRDE_{P\,2} \\ DVDE & DVDE_T & DVDE_P \\ I & DBDE_T & DBDE_P \end{bmatrix}$$

The covariance matrix of errors in the relative state in the reference frame is:

$$\overline{\begin{bmatrix} \delta r \\ \delta v \\ \delta b \end{bmatrix} \begin{bmatrix} \delta r\ \delta v\ \delta b \end{bmatrix}} = \text{DSDS SIG DSDS}^T + \text{DSDG GAM DSDG}^T$$

The W matrix can be found from the above covariance matrix by forming a diagonal matrix E_c consisting of the square roots of the diagonalized covariance matrix. If the rows of the matrix RV are eigenvectors of COV; that is RV is defined to be:

$$\begin{bmatrix} E^2_{c_{11}} & & \\ & E^2_{c_{22}} & \\ & & \text{etc} \end{bmatrix} = \text{RV COV RV}^T$$

The W matrix is then:

$$W = RV^T \begin{bmatrix} E_{c_{11}} & & \\ & E_{c_{22}} & \\ & & \text{etc} \end{bmatrix}$$

(Note: The above indicates symbolically the definition of W but the actual routine to compute W may or may not use the above steps). The vector state $\left[\underline{r}_{U2}, \underline{v}_{U2}\right]$, the time t_2 and the relative W matrix are returned to the calling program.

3. DETAILED FLOW DIAGRAMS

This section contains detailed flow diagrams for the Automatic Initialization Routine and Measurement Incorporation Routine of the Co-orbiting Vehicle Navigation Module. A nine dimensional W-matrix is computed. The three adjoined elements are for constant sensor bias in r, β, and θ. These particular biases were chosen only as an example.

Two routines used are not yet documented: the Lambert Routine and the Eigenvalue Routine.

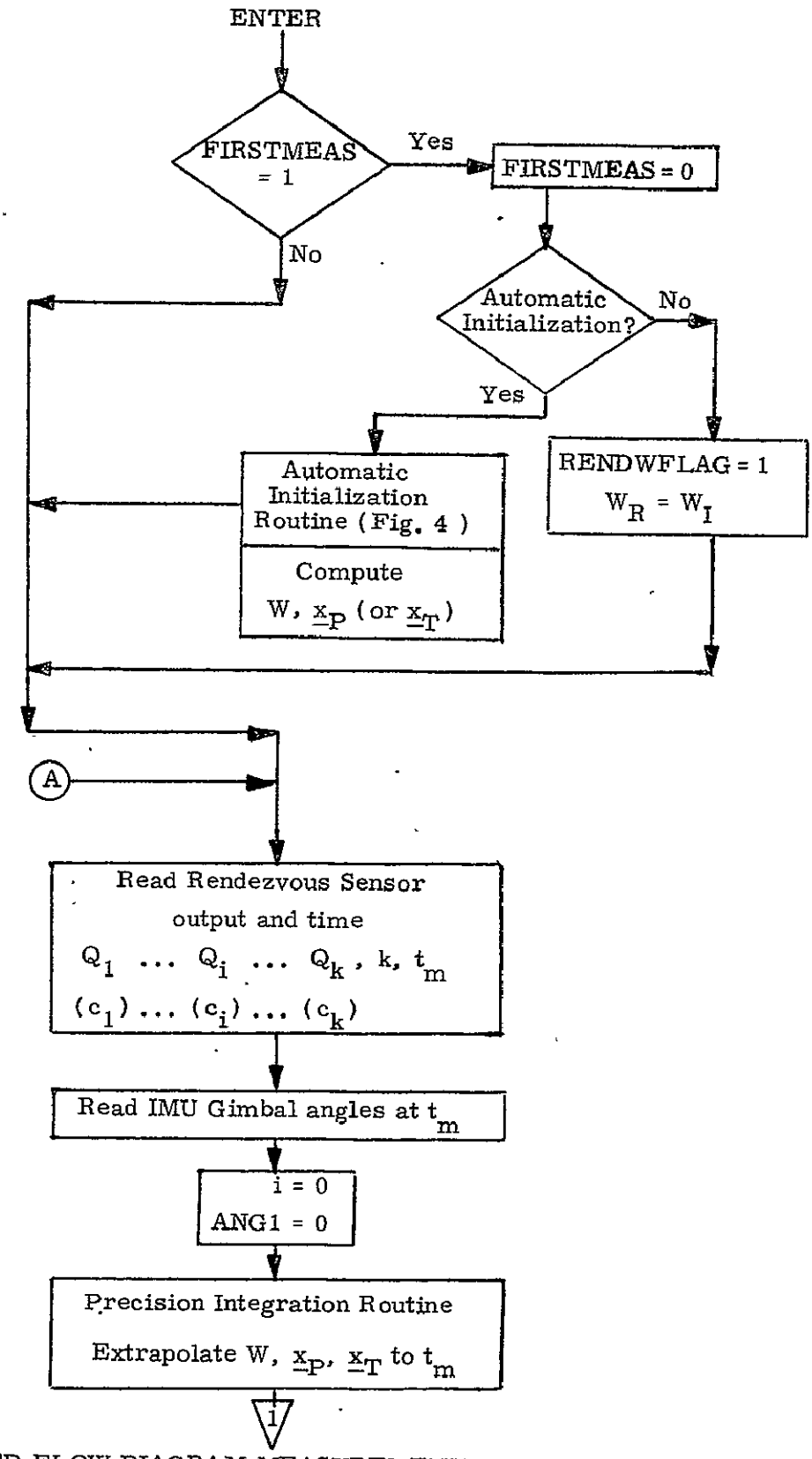


Figure 3a DETAILED FLOW DIAGRAM, MEASUREMENT INCORPORATION ROUTINE

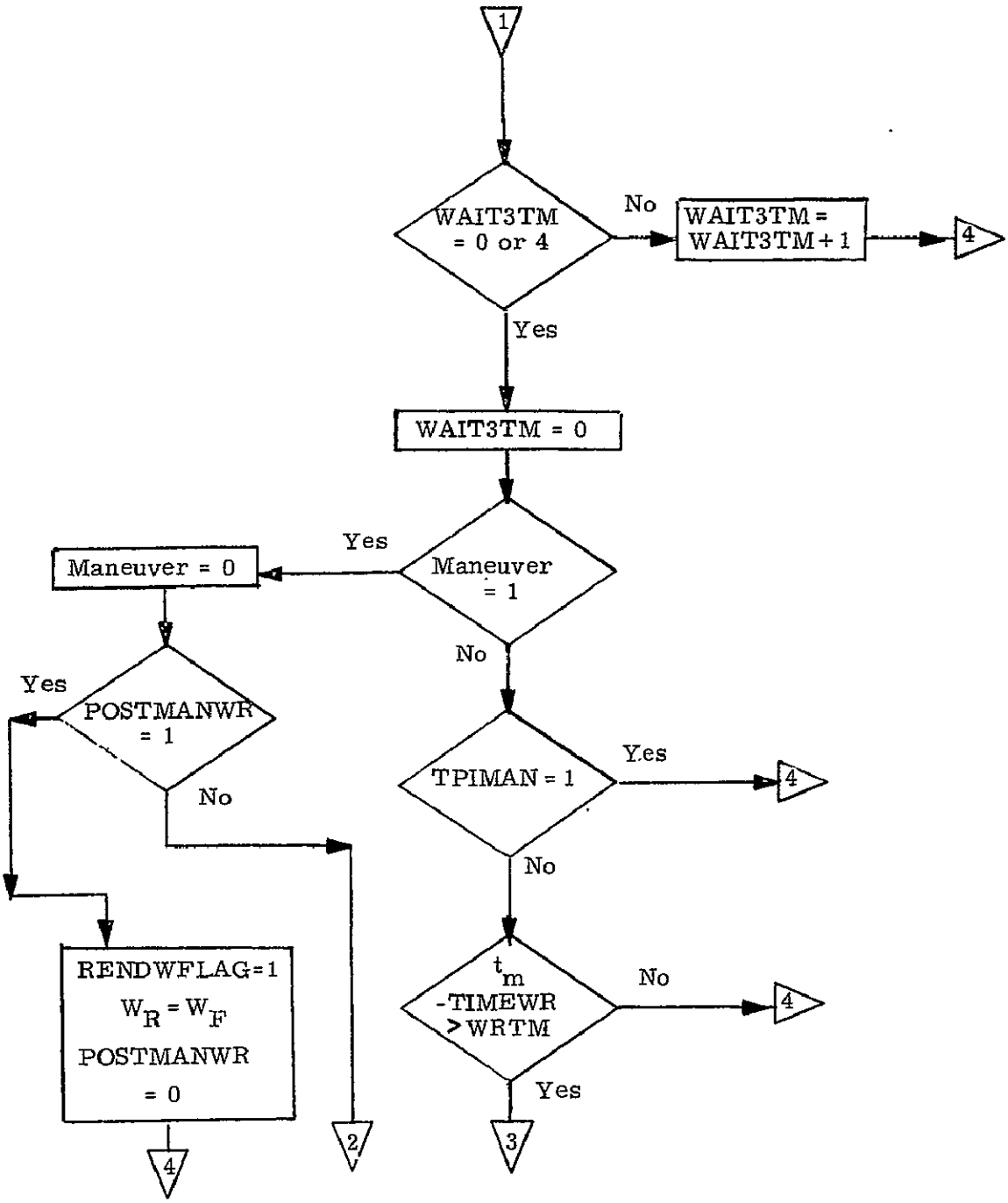


Figure 3b DETAILED FLOW DIAGRAM, MEASUREMENT INCORPORATION ROUTINE

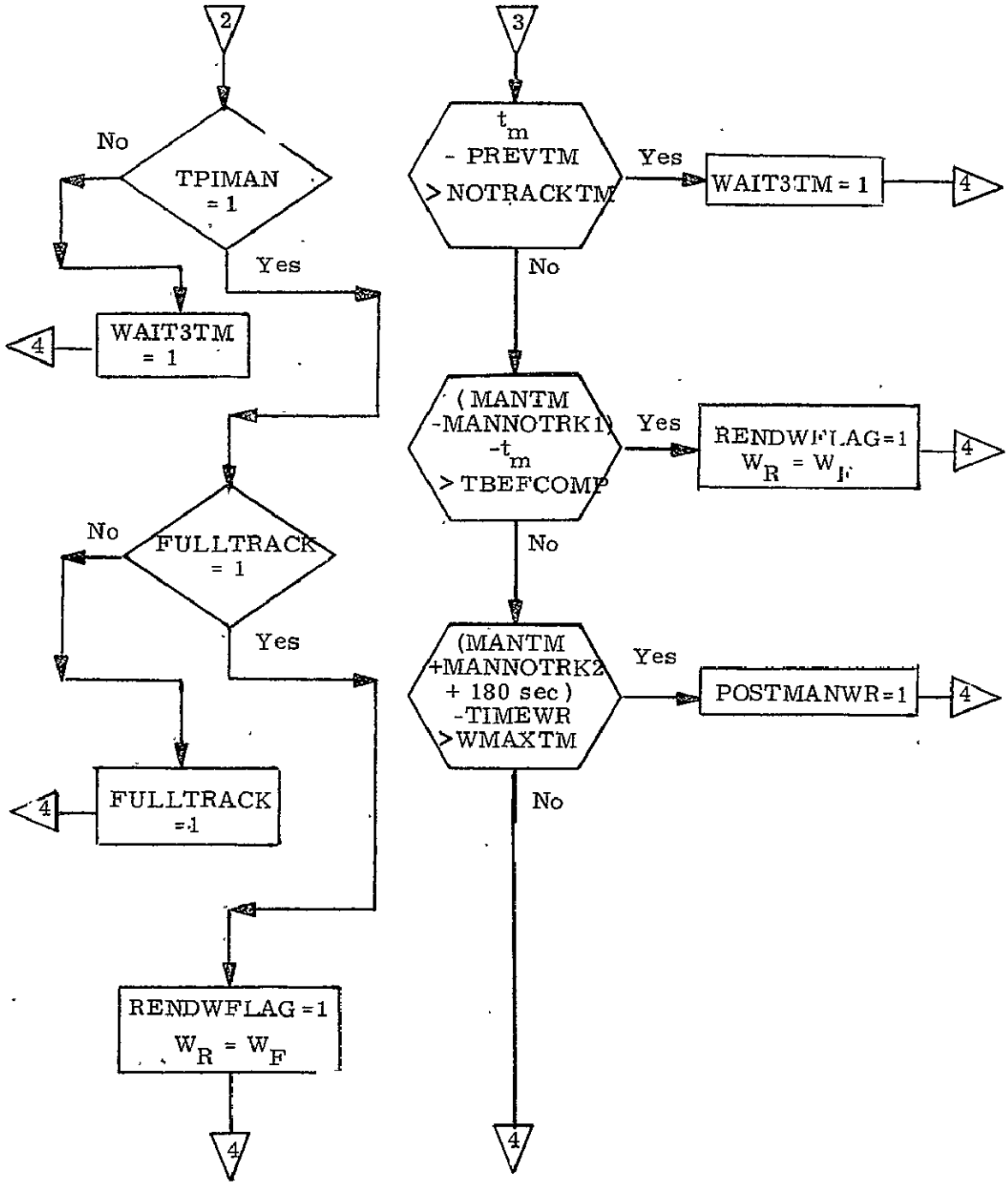


Figure 3c DETAILED FLOW DIAGRAM, MEASUREMENT INCORPORATION ROUTINE

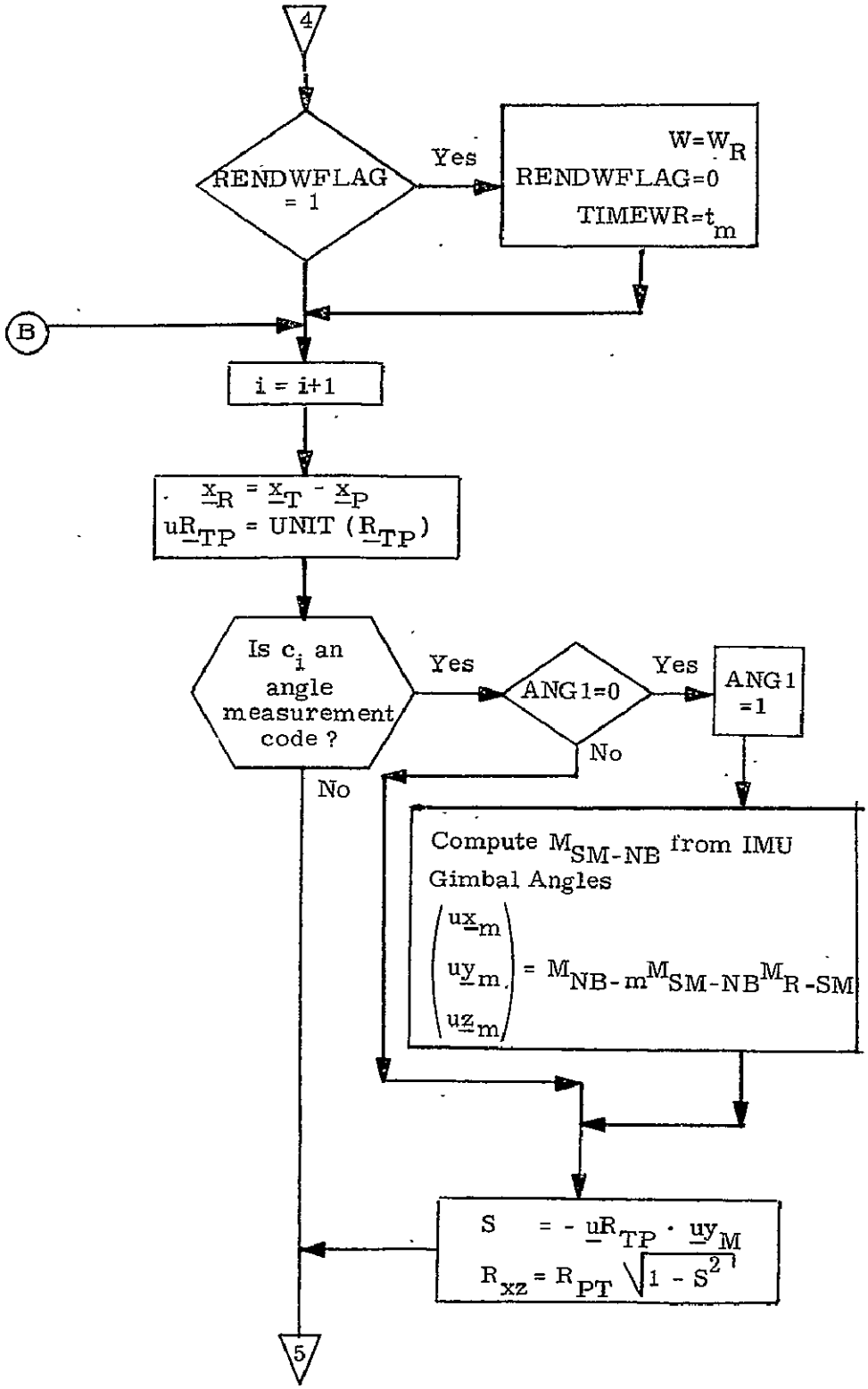


Figure 3d DETAILED FLOW DIAGRAM, MEASUREMENT INCORPORATION ROUTINE

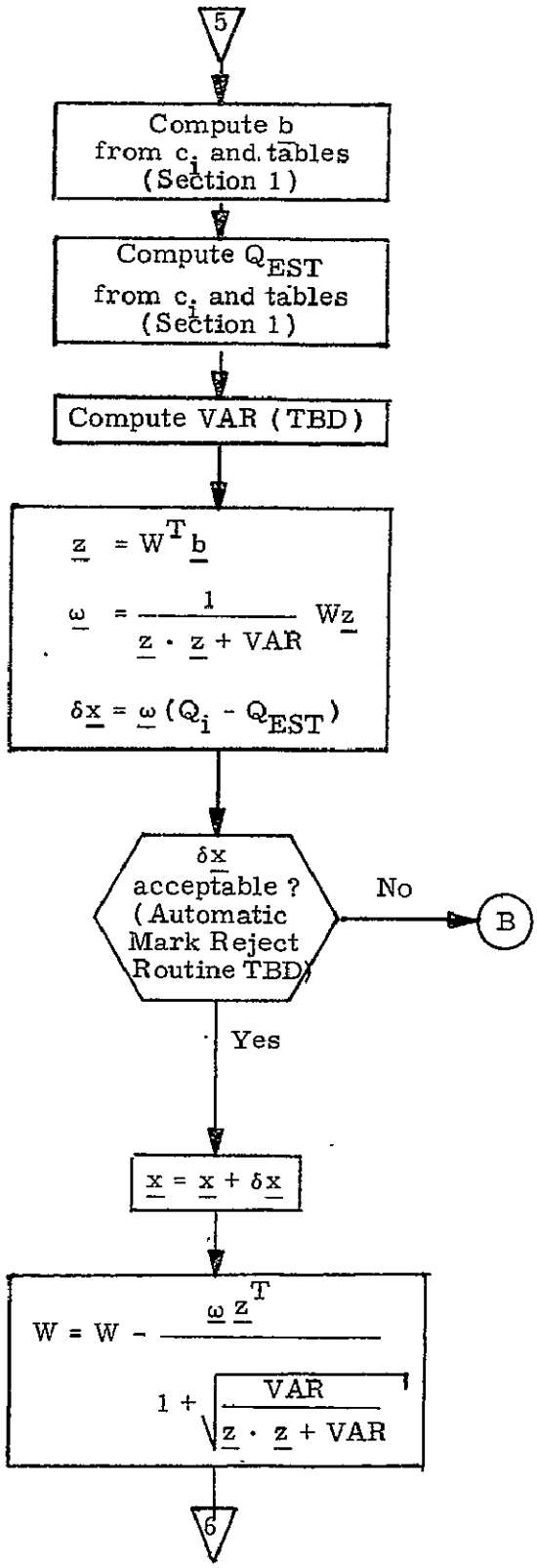


Figure 3e DETAILED FLOW DIAGRAM, MEASUREMENT INCORPORATION ROUTINE

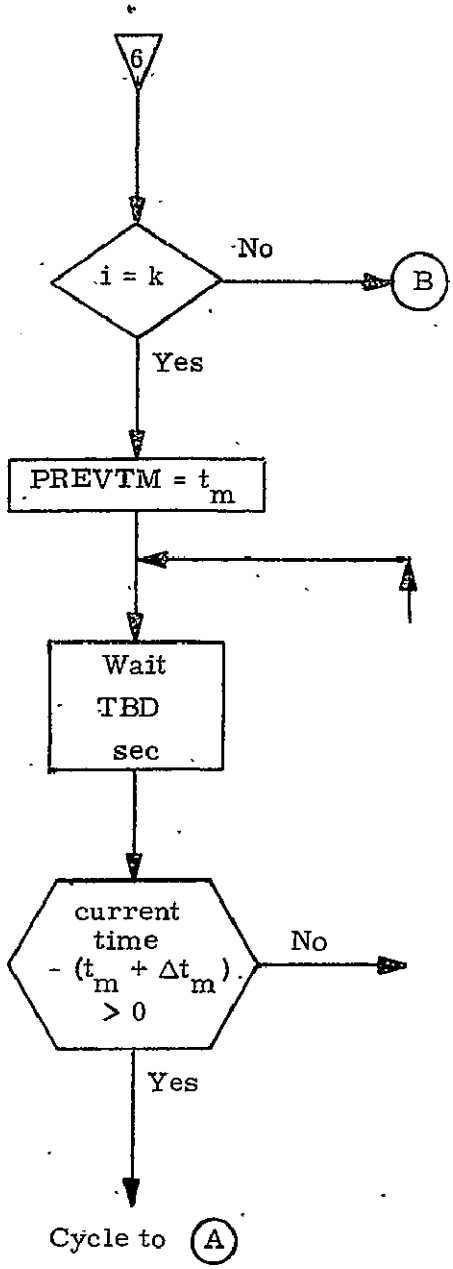


Figure 3f DETAILED FLOW DIAGRAM,MEASUREMENT INCORPORATION ROUTINE

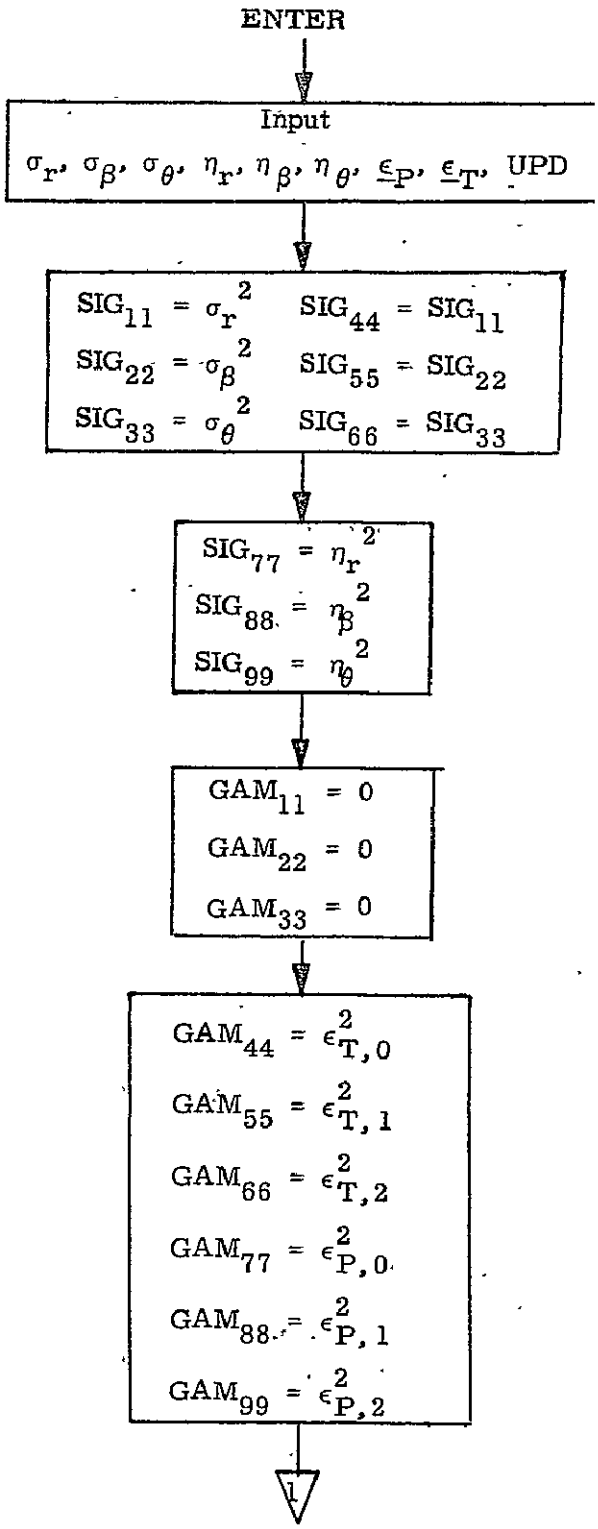


Figure 4a DETAILED FLOW DIAGRAM,AUTOMATIC INITIALIZATION ROUTINE

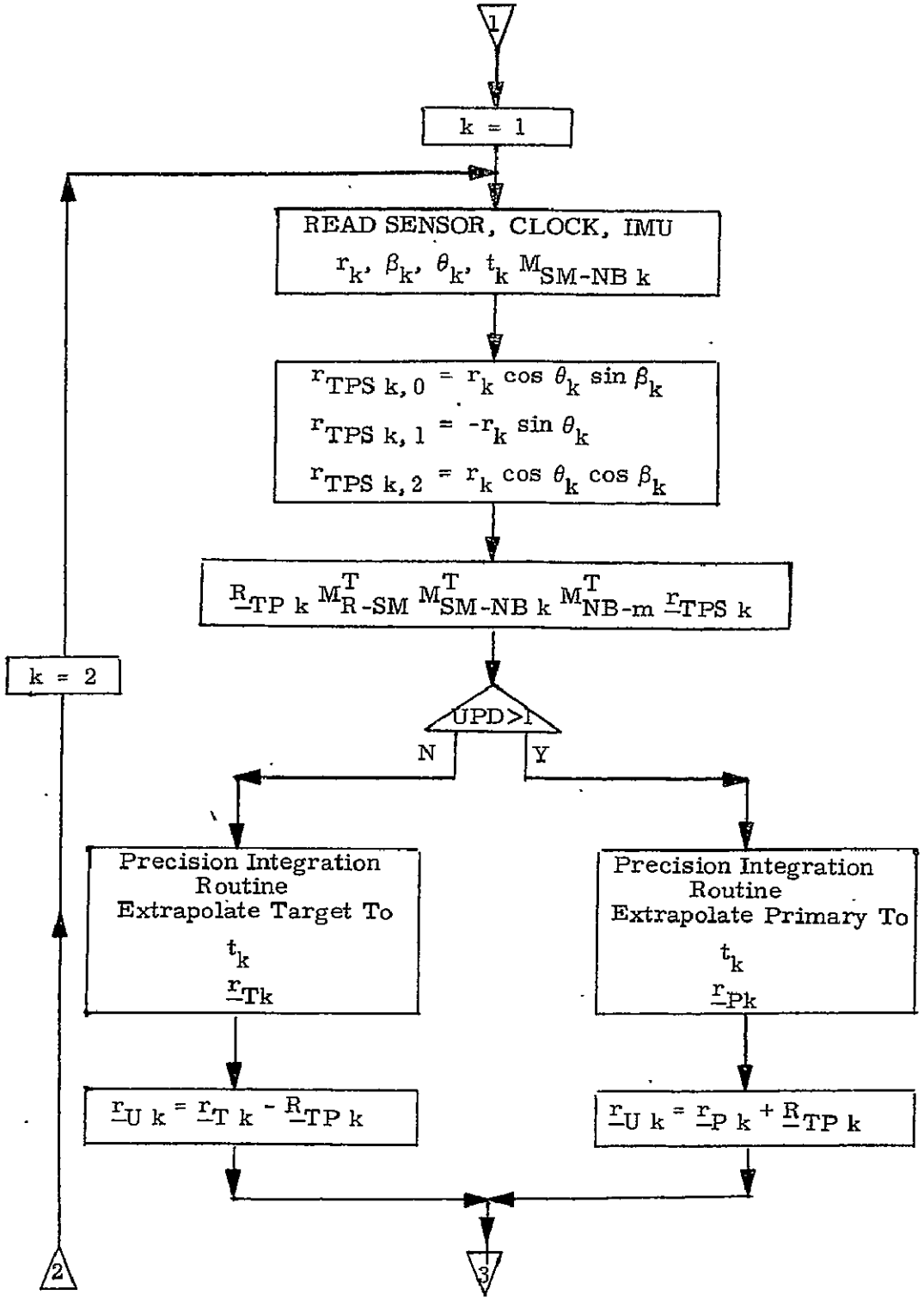


Figure 4b DETAILED FLOW DIAGRAM,AUTOMATIC INITIALIZATION ROUTINE

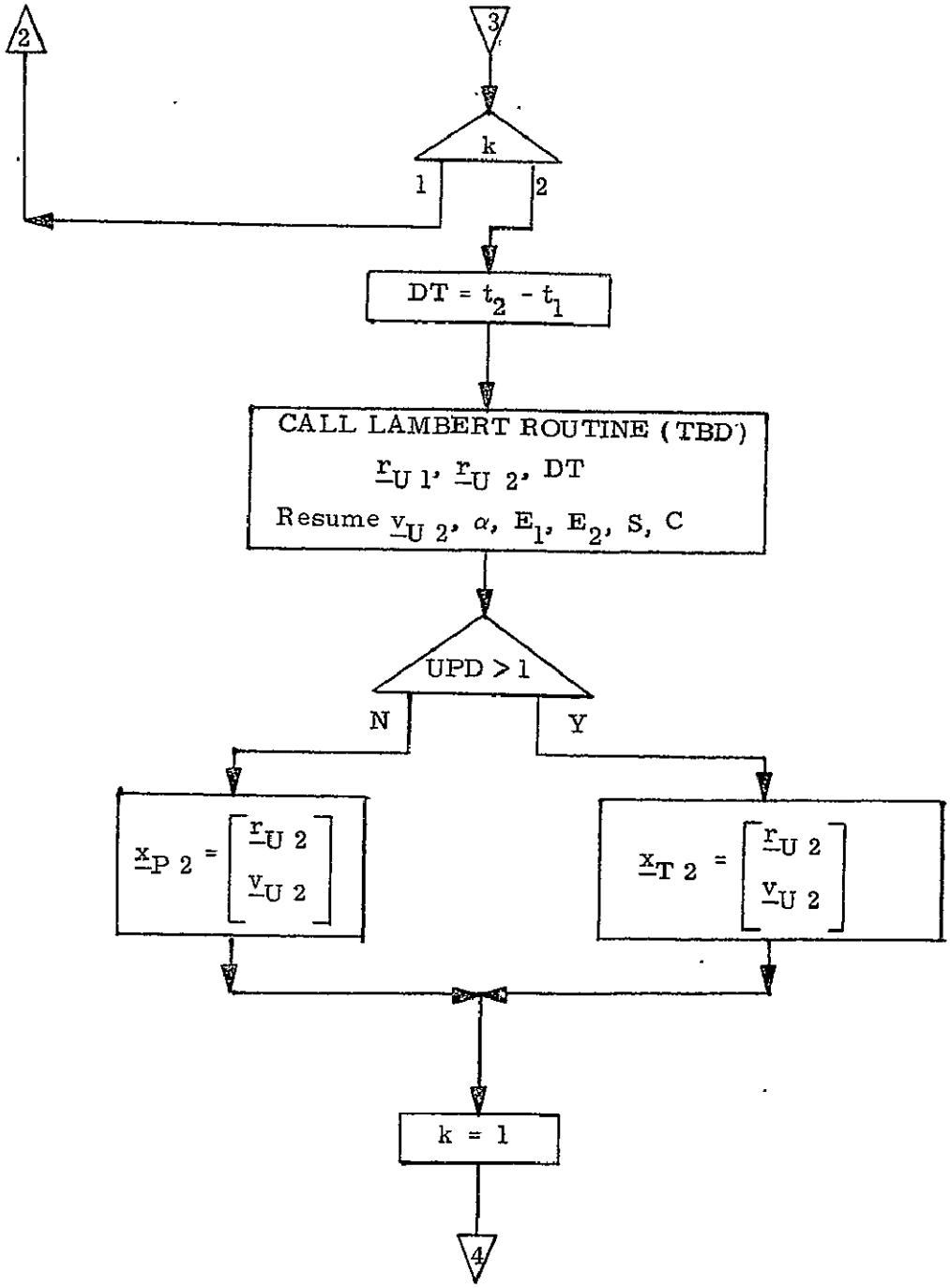


Figure 4c DETAILED FLOW DIAGRAM,AUTOMATIC INITIALIZATION ROUTINE

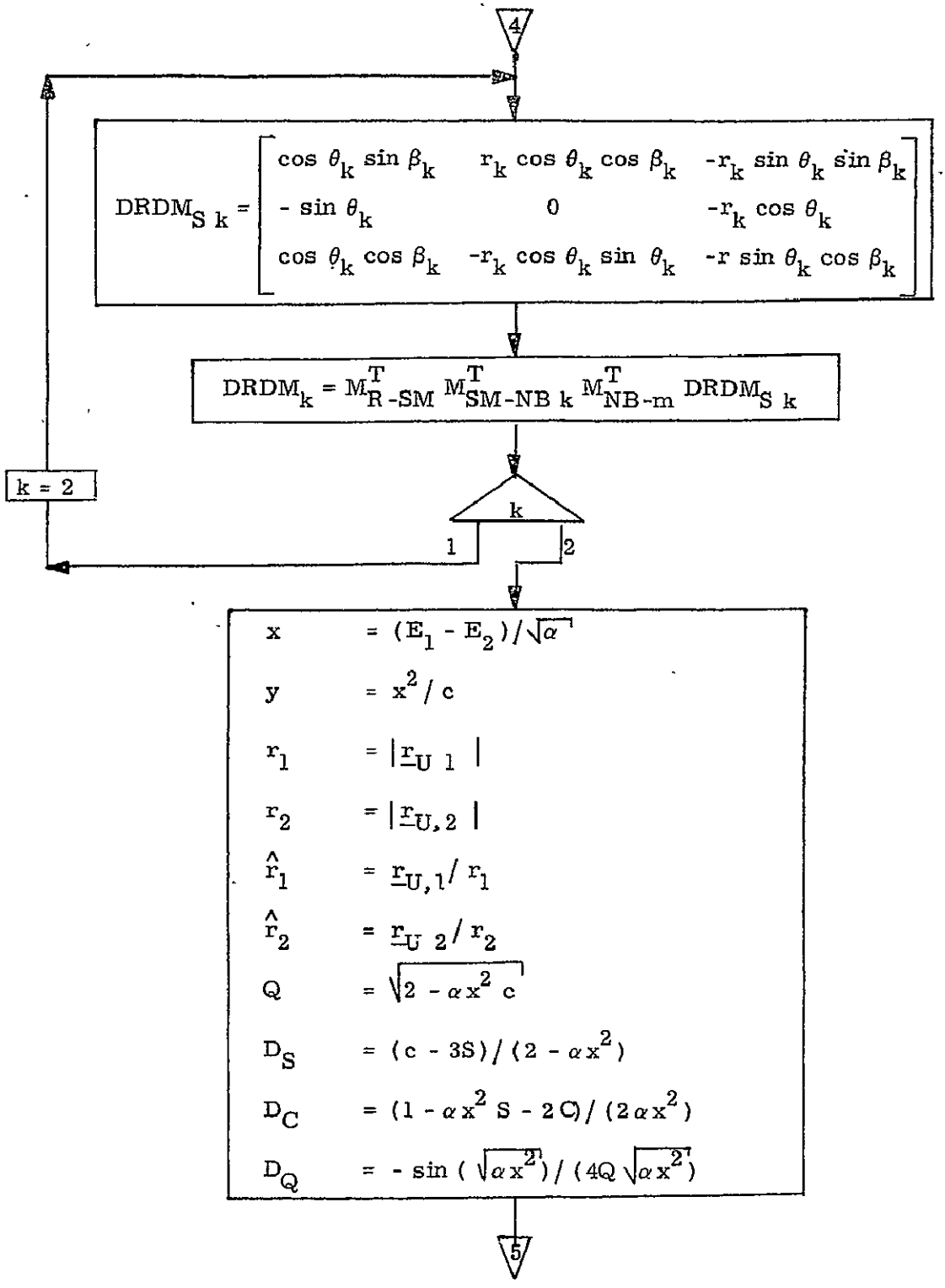


Figure 4d DETAILED FLOW DIAGRAM, AUTOMATIC INITIALIZATION ROUTINE

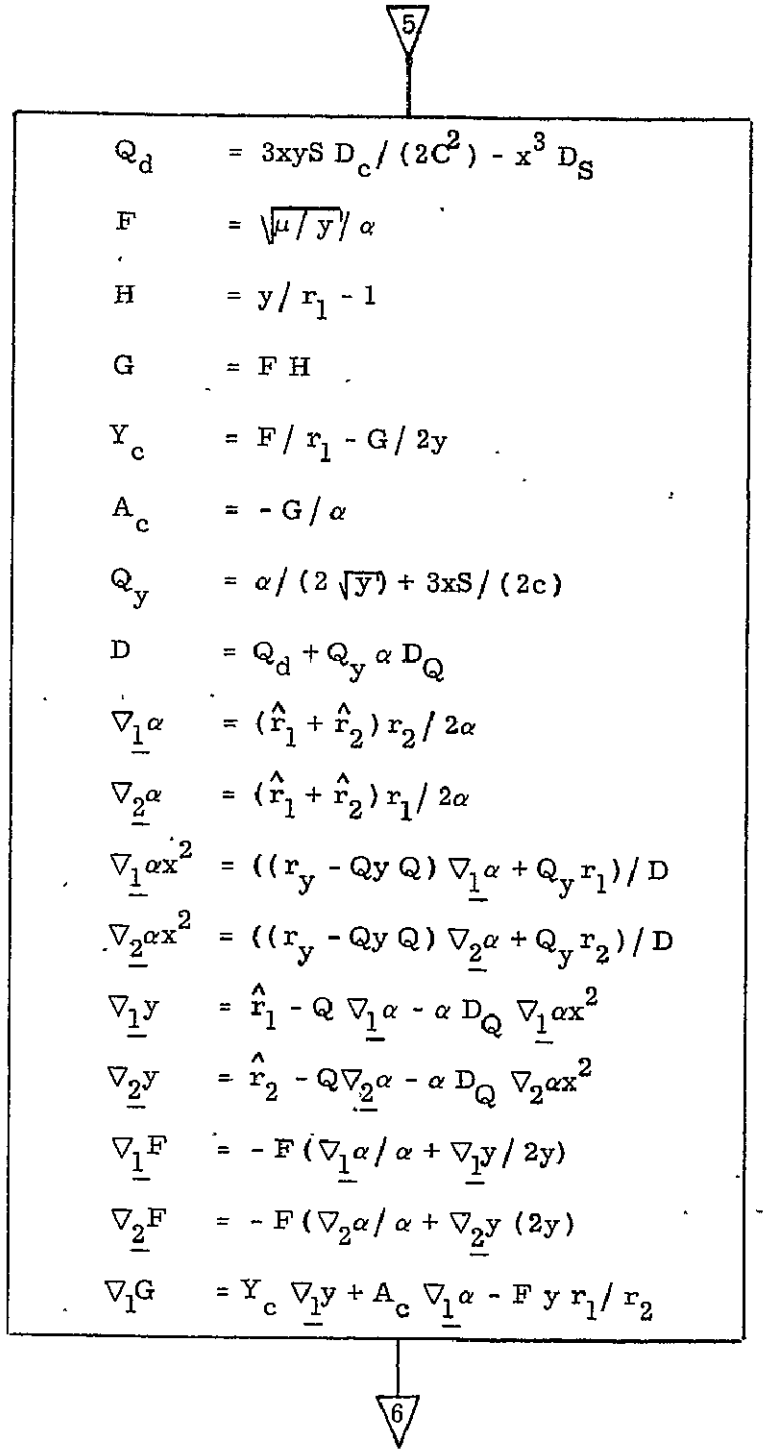
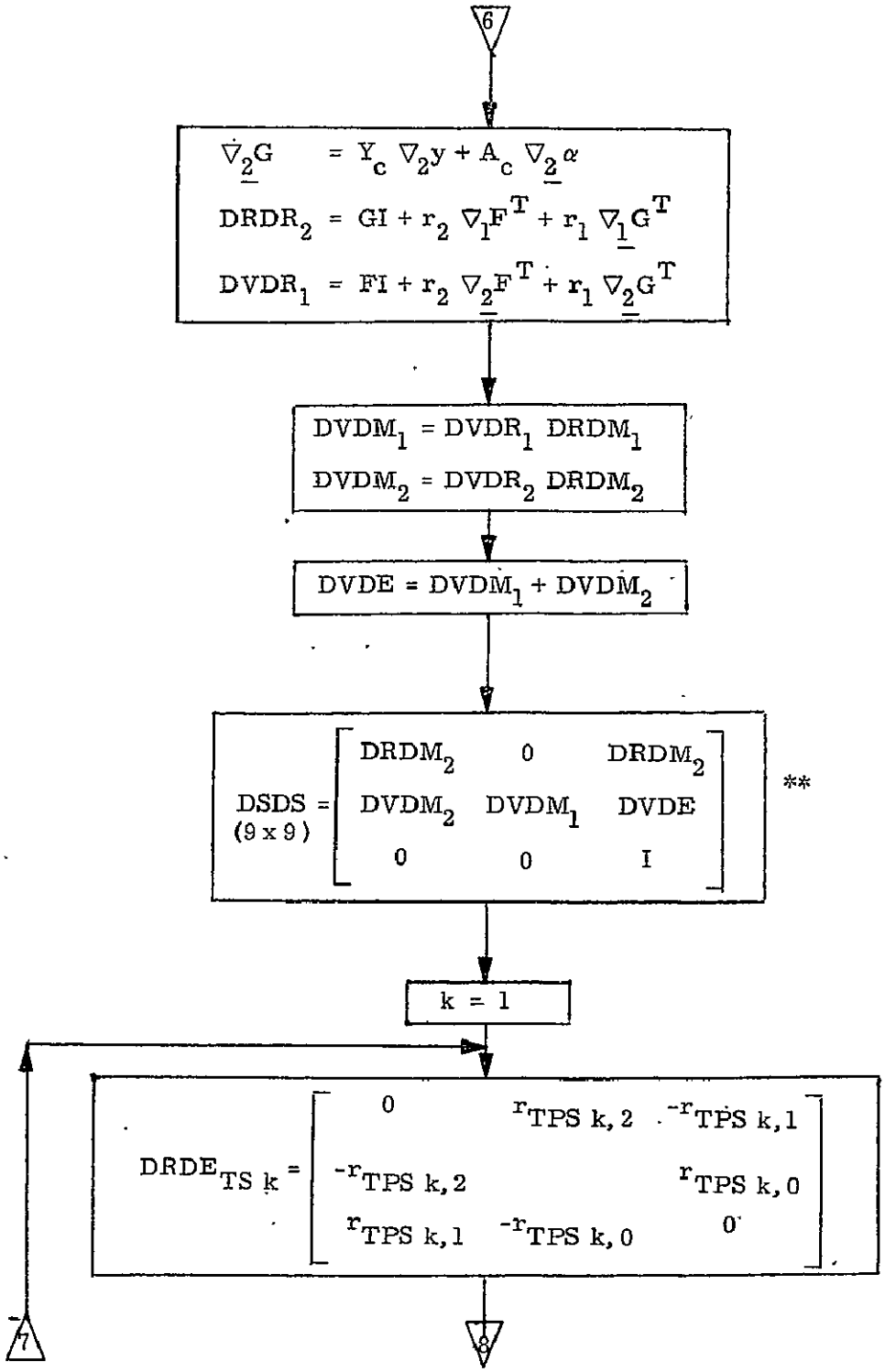


Figure 4e DETAILED FLOW DIAGRAM, AUTOMATIC INITIALIZATION ROUTINE



*I is the 3D identity matrix

**0 is the 3D null matrix

Figure 4f DETAILED FLOW DIAGRAM, AUTOMATIC INITIALIZATION ROUTINE

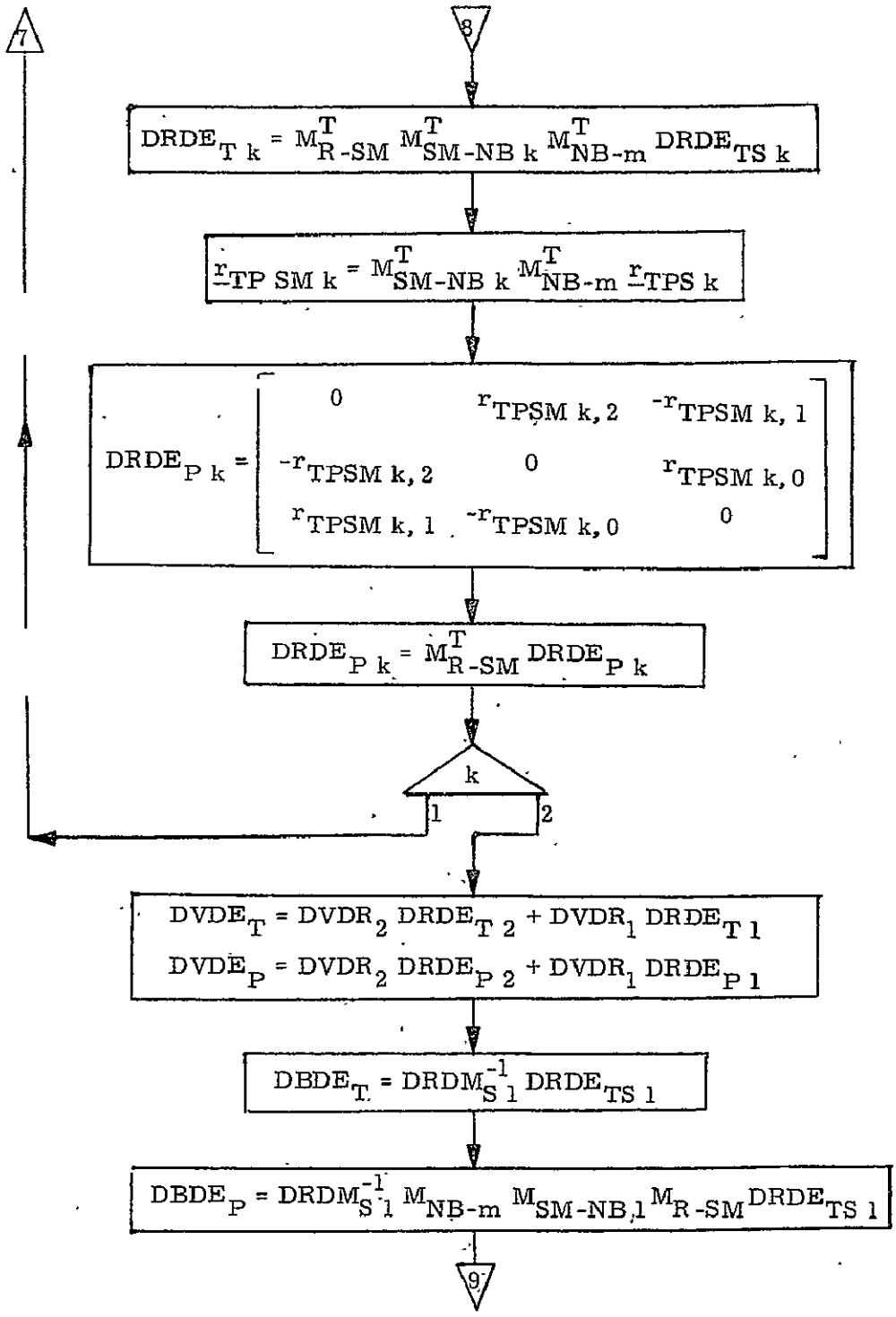


Figure 4g DETAILED FLOW DIAGRAM,AUTOMATIC INITIALIZATION ROUTINE

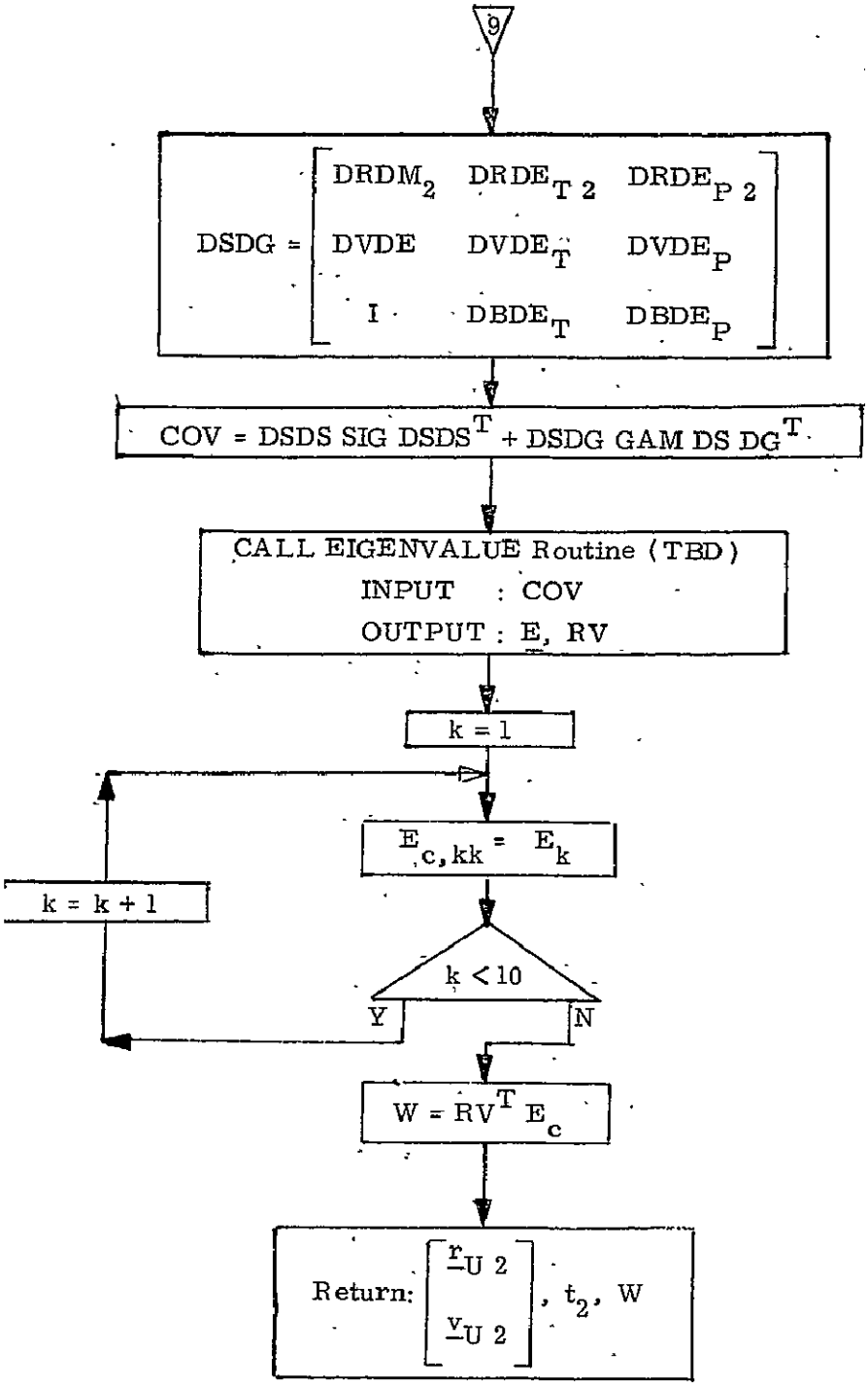


Figure 4h DETAILED FLOW DIAGRAM, AUTOMATIC INITIALIZATION ROUTINE

4. SUPPLEMENTARY INFORMATION

The equations presented in this report are the results to date of studies performed under a G & C shuttle task to develop G & N equations for automatic rendezvous. Two fundamental approaches were taken in these studies: (1) automate proven Apollo rendezvous navigation equations; (2) develop optimum rendezvous navigation equations. By presenting the equations in the general form shown, they are made to reflect formulations developed using both approaches (1) and (2). Analyses performed to evaluate the filter equations are reported in the references.

To complete the automation of the Apollo filter, an automatic mark reject routine remains to be formulated.

References

1. Muller, E.S., Kachmar, P.M., The Apollo Rendezvous Navigation Filter-Theory, Description and Performance, Vol. 1 of 2, Draper Lab Report R-649, June 1970, MIT.

2. Muller, E. and Kachmar, P., STS Progress Report - Rendezvous Studies, Memo 23A STS #11-70, May 1970, MIT.

3. Phillips, R., Lambert Determination of the Relative State for Rendezvous, Memo 23A STS #6-70, 28 April 1970, MIT.

4. Tempelman, W., Non-Cooperative Rendezvous Trajectories, 23A STS Memo #4-70, 30 April 1970, MIT.

5. Sears, N.E., et. al., STS Avionic Specification Presentation, C-3397, 12 November 1969, MIT.

6. Users Guide to Minkey Rendezvous, E-2448, 17 July 1970, MIT.

9.8.3 Rendezvous Guidance (TBD)

9.8.4 Rendezvous Attitude Control (TBD)

9.9 STATION KEEPING MISSION PHASE

Station keeping begins with the targeting for braking as the Shuttle approaches the target vehicle sometime after TPI. This phase includes braking targeting, braking, positioning for station keeping, automatic station keeping, repositioning to station keep at a different position relative to the target vehicle and/or in preparation for docking. Automatic station keeping here means the preservation of a precise relative position with the target vehicle with no requirement for manual commands. Automatic station keeping may occur before docking, after docking, and on missions in which docking does not occur. This phase ends when the docking maneuver begins, or when the shuttle is separated from the target vehicle with no intention of preserving a precise relative position with it.

The software functions required in this mission phase are the following:

1. Estimate relative state of target vehicle based on external measurements.
2. Estimate absolute states of both shuttle and target vehicle.
3. Compute (target) the braking $\Delta V(s)$ required, their direction, and the time(s) of ignition.
4. Execute braking maneuver by commanding engine(s) on, providing attitude commands during braking, and commanding engine(s) off.
5. Powered flight navigation.
6. Automatically preserve a relative position and attitude with the target vehicle by periodic RCS engine on/off commands with a minimum-fuel technique. Spatial and angular requirements and allowable variations during automatic station keeping are TBD.
7. Provide RCS engine commands to achieve commanded attitude during ΔV maneuvers and during coast periods (digital autopilot).

Repositioning for docking maneuver initiation, for a separation maneuver, or for station keeping at a different relative position is assumed to be a manual function and therefore no software for performing these

maneuvers automatically is required.

A flow of software functions during station keeping appears in Figure 1. Some functions overlap with other mission phases and only those equations not provided in earlier sections are discussed here.

9.9.1 Relative State Estimation (TBD)

9.9.2 Station Keeping Guidance (TBD)

9.9.3 Station Keeping Attitude Control (TBD)

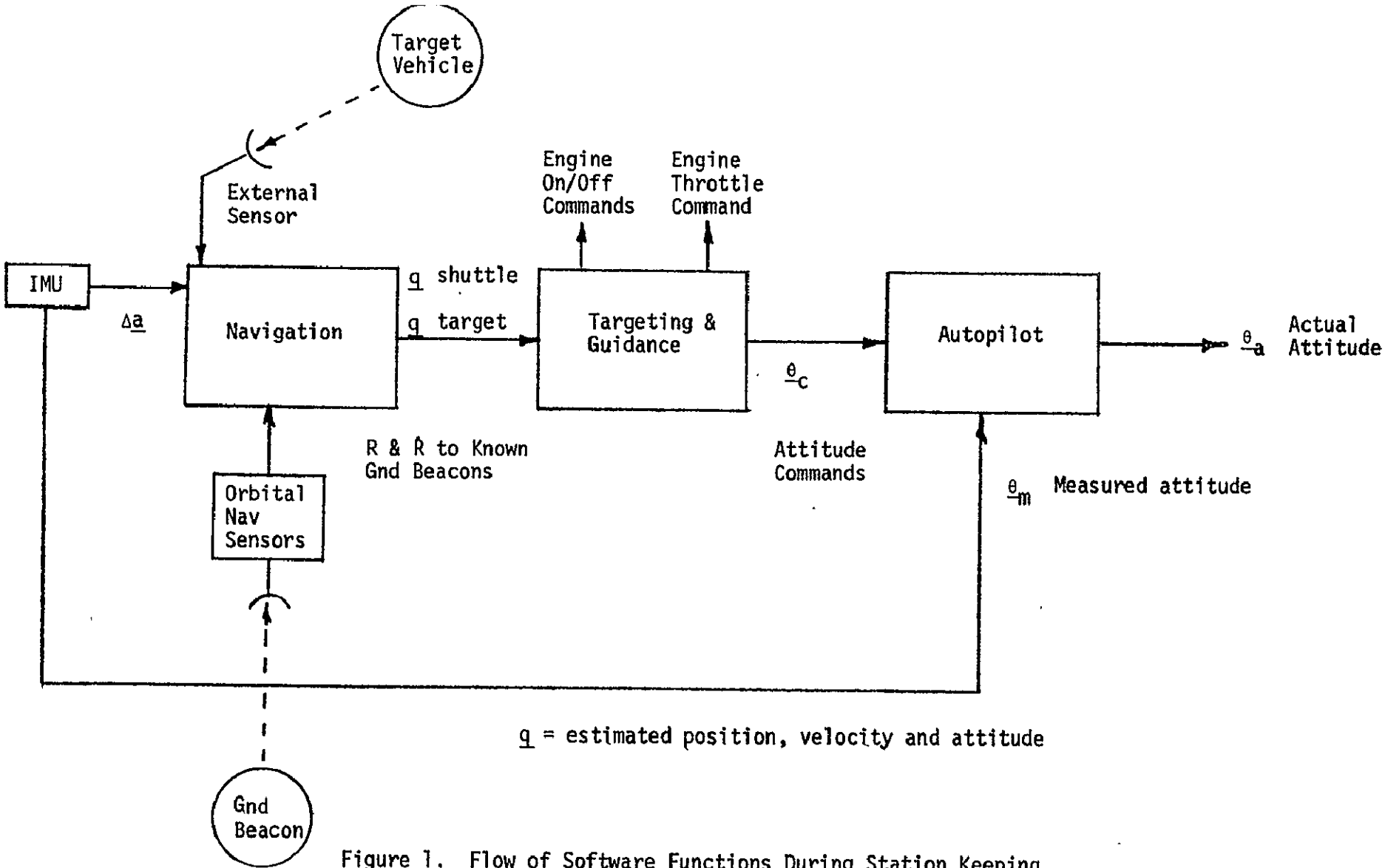


Figure 1. Flow of Software Functions During Station Keeping

9.10 DOCKING AND UNDOCKING

The two distinct events are described as one phase since the events are essentially reversals of one another. The distinction between the docking event and terminal rendezvous is the point at which the maneuver defined by the docking constraints on such variables as range, range rate, attitude, and attitude rate is initiated.

The mode of docking is still open; that is, it has not been determined whether the docking will be performed manually or automatically with a manual backup capability. The GN&C software functions to be performed during this phase are based on an automatic docking with manual backup. The docking SW functions are:

a) Specific force integration updates of relative states during translational burns. This function will maintain the relative state between the orbiter and its co-orbiting target during orbiter burns.

b) Maintain attitude-hold about a desired orientation.

c) Compute and command steered-attitude RCS ΔV maneuvers for docking.

d) Make high-frequency steering estimates between guidance samples for docking.

e) Provide three-axis translation control.

The SW functions for undocking are:

a) Configure all GN&C systems for the next mission phase.

b) Schedule undocking.

c) Compute and command ΔV translations.

d) Provide capability to advance inertial state vector from an initial state to a final state.

e) Provide for specific force integration updates of relative state during burns associated with undocking.

f) Compute and command attitude-hold RCS ΔV maneuvers.

Figure 1 displays a function flow diagram of the docking GN&C software functions.

9.10 DOCKING AND UNDOCKING (con't)

Presently, no specific sensors for automatic docking have been baselined. However, control laws and a navigation routine have been approved by the GN&C Software Equation Formulation and Implementation Panel. These equation formulations are described in the following references:

a) E. T. Kubiak, "Automatic Docking Control Law," MSC EG2-3-71, date 5 January 1971.

b) E. P. Blanchard, G. M. Levine, "Docking and Undocking Navigation," MIT No. 2-71, dated January 1971.

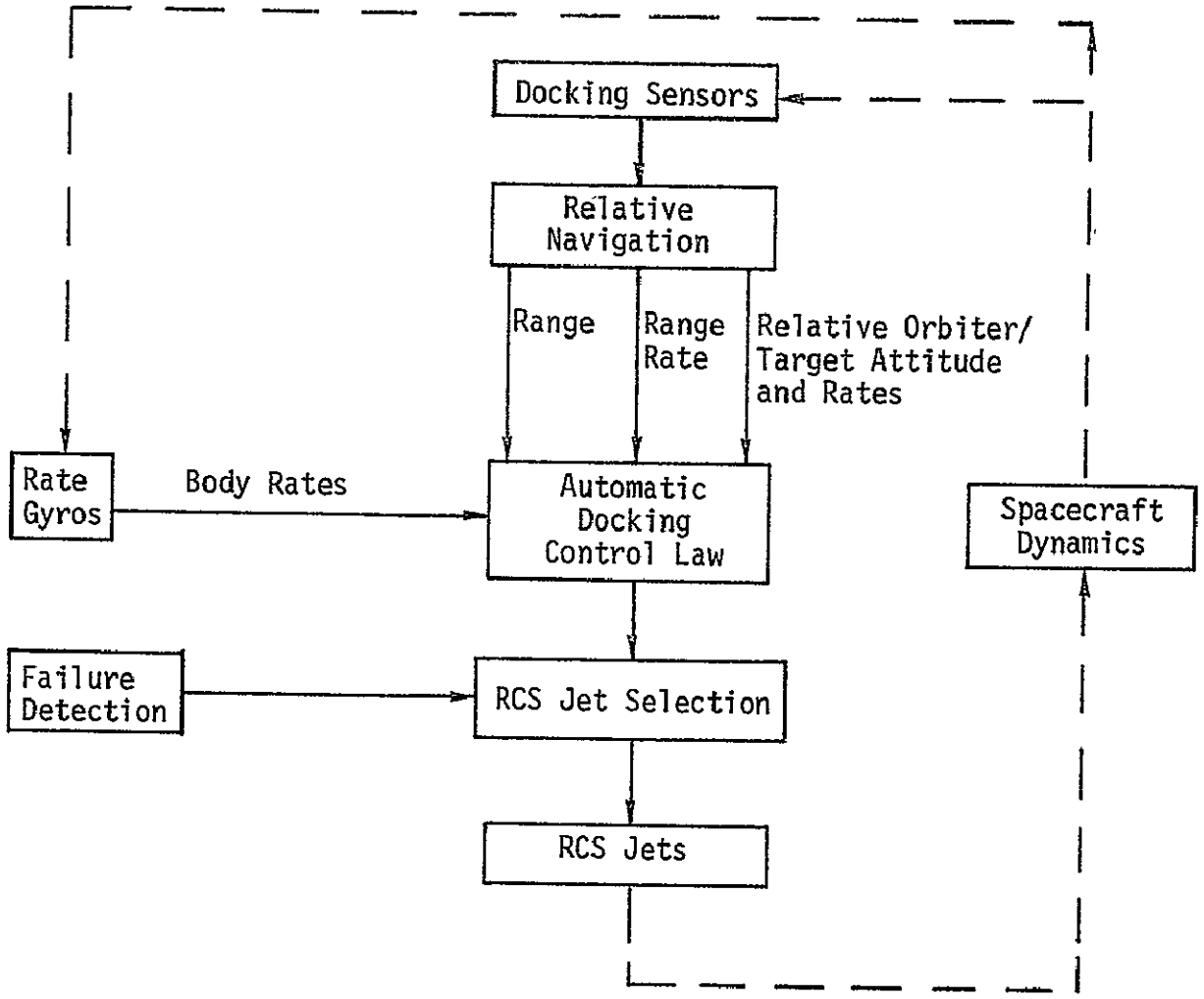


Figure 1

Overall Functional Flow Diagram
for Docking and Undocking

SPACE SHUTTLE

GN&C SOFTWARE EQUATION SUBMITTAL

Software Equation Section Docking and Undocking Submittal No. -

Function Relative Navigation

Module No. ON3 Function No. -2, -5, -8 (MSC 03690)

Submitted By: E. P. Blanchard, G. M. Levine Co. MIT
(Name)

Date: January 1971

NASA Contact: W. H. Peters Organization EG2
(Name)

Approved by Panel III K. J. Cox K.J. Cox Date 3/10/71
(Chairman)

Summary Description: The objective of the Docking and Undocking Navigation Program is to use the data from the docking sensor to determine the relative position and attitude of the target vehicle with respect to the shuttle. These quantities and their rates are computed periodically and used in the generation of guidance commands during both the docking and undocking procedure.

Shuttle Configuration: (Vehicle, Aero Data, Sensor, Et Cetera)
Assumes a docking sensor which measures the azimuth and elevation angles to each of four sources located on the target vehicle.

Comments:

(Design Status) The algorithm for source identification is TBD.

(Verification Status) Open-loop testing has been performed simulating the sensor-target geometry and the sensor.

Panel Comments: The equations are baselined subject to the qualification that they are based on a sensor configuration which has not been baselined. Also, the range and range rate computations must be coordinated with those in the Automatic Docking Control Law.

1. INTRODUCTION

The objective of the Docking and Undocking Navigation Program is to use the data from the docking sensor to determine the relative position and attitude of the target vehicle with respect to the shuttle. These quantities and their rates are computed periodically and used in the generation of guidance commands during both the docking and undocking procedure.

The docking sensor measures the azimuth and elevation angles to each of four sources located on the target vehicle. The configuration of these four sources is designed to permit recognition of one source by its angular position relative to the other sources under all allowable rotations of the shuttle with respect to the target vehicle within certain restricted operating limits. As long as the operating-limit restrictions are satisfied, it is not necessary for the sensor to identify individually the sources; i.e., the sensor portion of the system does not have to associate a particular source with each set of azimuth and elevation angles, that process can be accomplished computationally. Furthermore, in this case, the data from only three of the four sources are required to obtain a complete relative position and attitude solution. The velocity and attitude rates are determined by numerically differencing two position and attitude solutions.

On the other hand, if the operating-limit restrictions are violated, then the equations have multiple solutions, and all four sets of data must be used to resolve the ambiguities.

An additional reason for the presence of four sources is to provide an option for selecting the best combination of three sources; i.e., at close range to permit selection of sources which fall within the sensor field of view, and at long range at provide a combination of three sources which yield a more accurate solution.

NOMENCLATURE

A	Intermediate matrix
a_i	Azimuth angle to source i
B	Intermediate matrix
C	Cos 40^o
e_i	Elevation angle to source i
f	Rate indicator
FLAG	Flag used in iteration
$FLAG_m$	Flag used in rate calculation
I	Negative radicand indicator
$\underline{i}_{XS}$, $\underline{i}_{YS}$, $\underline{i}_{ZS}$	Unit vectors along shuttle coordinate axes
$\underline{i}_{XT}$, $\underline{i}_{YT}$, $\underline{i}_{ZT}$	Unit vectors along target vehicle coordinate axes
K	0.4 or 2.5 depending on selected source set
k	Index used in rate calculations
M	Transformation matrix
m	Index used in rate calculations
m_{ij}	Element of M
n	Index used in rate calculations
p	Source set indicator

$\underline{r}$	Relative position vector between docking hatches
$\dot{\underline{r}}$	Rate of change of $\underline{r}$
$\underline{r}_i$	Vector from sensor to source i
r_i	Magnitude of $\underline{r}_i$
$\underline{r}_{ij}$	Vector from source i to source j
r_{ij}	Magnitude of $\underline{r}_{ij}$
r_M	Maximum value of s_1
r_{new}, r_{old}	Iteration interval end points
S	Sin 40^o
s_1	Trial value of r_1
$\underline{y}$	Vector from sensor to shuttle docking hatch
$\underline{z}$	Vector from source 1 to target vehicle docking hatch
$\underline{\gamma}$	γ_1, γ_2, γ_3
$\dot{\underline{\gamma}}$	Rate of change of $\underline{\gamma}$
γ_1, γ_2, γ_3	Rotation angles
Δr	$s_1 - r_1$
Δr_{old}	Previous value of Δr
Δt	Navigation cycle time
ϵ	Error tolerance
θ_{ij}	Angle between lines-of-sight to sources i and j
ρ	Scaling factor
Subscript S	Shuttle coordinates
Subscript T	Target vehicle coordinates

2. SOURCE CONFIGURATION AND OPERATING LIMITS

In this section, the configuration of the four sources on the target vehicle is described, and the operating limits under which a unique relative position and attitude solution can be obtained is discussed.

Refering to Fig. 1, define a coordinate system fixed in the target vehicle with origin at source 1; X axis parallel to the docking axis; and $\underline{i}_{XT}$, $\underline{i}_{YT}$, and $\underline{i}_{ZT}$ unit vectors along the three axes. Let $\underline{r}_{ij}$ be the vector from source i to source j. Then the locations of sources 2, 3, and 4 are defined by

$$\underline{r}_{12T} = 0.4\rho \begin{pmatrix} \cos 40^{\circ} \\ -\sin 40^{\circ} \\ 0 \end{pmatrix}$$

$$\underline{r}_{13T} = \rho \begin{pmatrix} \cos 40^{\circ} \\ 0 \\ \sin 40^{\circ} \end{pmatrix}$$

$$\underline{r}_{14T} = 2.5\,\rho \begin{pmatrix} \cos 40^{\circ} \\ -\sin 40^{\circ} \\ 0 \end{pmatrix}$$

where the subscript T denotes target vehicle coordinates and ρ is a scaling factor.

In order to discuss the restricted operating limits, define a coordinate system centered at the docking sensor in the shuttle with unit vectors $\underline{i}_{XS}$, $\underline{i}_{YS}$, and $\underline{i}_{ZS}$ along its axes. Again, let the X axis be parallel to the shuttle docking axis. Let γ_1, γ_2, and γ_3 be the three rotation angles which make the shuttle coordinate system parallel to the target vehicle system (the condition required for docking); i.e., a rotation of the shuttle system about the X axis through an angle γ_1, then a rotation about the resulting Y axis through an angle γ_2, and finally a rotation about the resulting Z axis through an angle γ_3 make the two systems parallel.

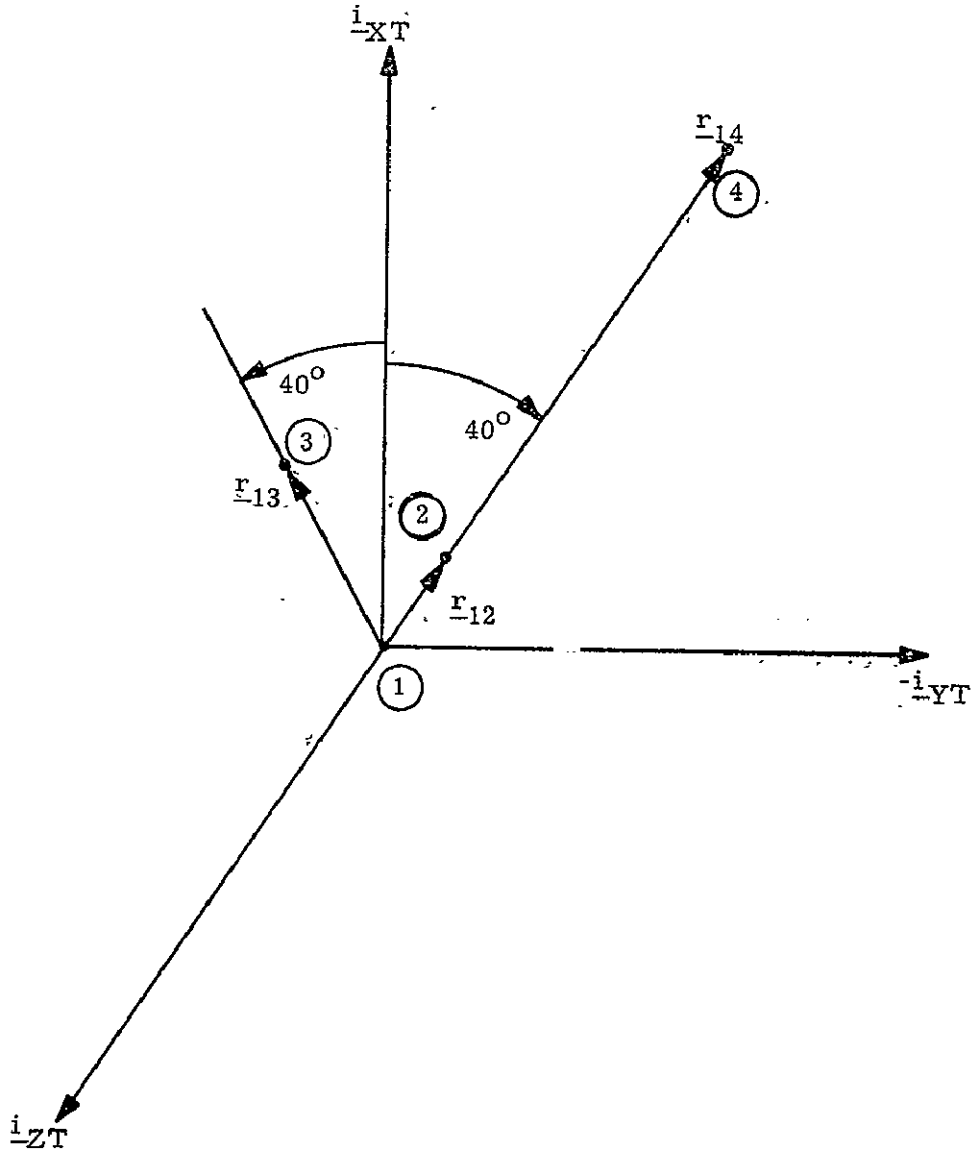


Figure 1 TARGET VEHICLE COORDINATE SYSTEM AND SOURCE CONFIGURATION

If the X (docking) axis of the shuttle is kept within 30° of the target vehicle docking axis, then an identification of the four sources can be made. Figure 2 illustrates the appearance of the sources for various relative vehicle attitudes for the case of zero γ_1. The center illustration shows the appearance of the sources when the two vehicles are properly aligned for docking; the other eight illustrations show the appearance at various points on the surface of the 30° cone defining the operation region.

For all relative vehicle orientations within the operating region, the following two facts hold:

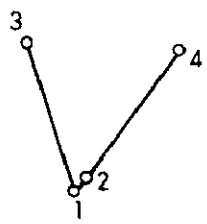
1) Sources 1, 2, and 4 lie on a straight line.

2) The observed distance between sources 1 and 2 always has the same ratio with respect to the observed distance between sources 1 and 4.

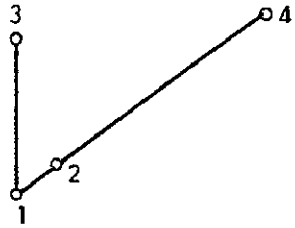
These two facts permit identification of the four sets of paired azimuth and elevation angles with the four sources.

The source configuration has also been selected to assure that for all relative vehicle orientations within the 30° operating region the distances from the sensor to the sources will satisfy the relationship $r_1 < r_2 < r_3 < r_4$. This relationship provides the resolution of the multiple solutions which would otherwise exist in the navigation equations.

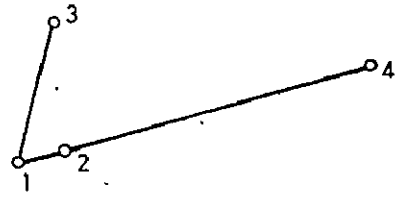
$\gamma_1 = 0$
$\gamma_2 = -\gamma_3 = -21.2^\circ$



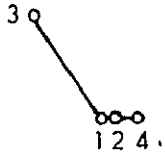
$\gamma_1 = \gamma_3 = 0$
$\gamma_2 = -30^\circ$



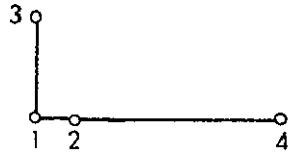
$\gamma_1 = 0$
$\gamma_2 = \gamma_3 = -21.2^\circ$



$\gamma_1 = \gamma_2 = 0$
$\gamma_3 = +30^\circ$

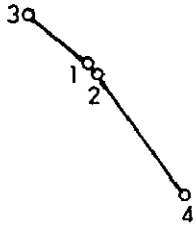


$\gamma_1 = \gamma_2 = \gamma_3 = 0$

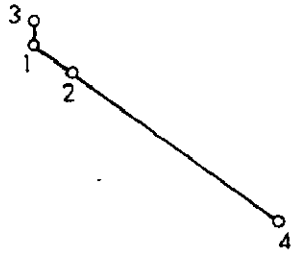


$\gamma_1 = \gamma_2 = 0$
$\gamma_3 = -30^\circ$

$\gamma_1 = 0$
$\gamma_2 = \gamma_3 = +21.2^\circ$



$\gamma_1 = \gamma_3 = 0$
$\gamma_2 = +30^\circ$



$\gamma_1 = 0$
$\gamma_2 = -\gamma_3 = +21.2^\circ$

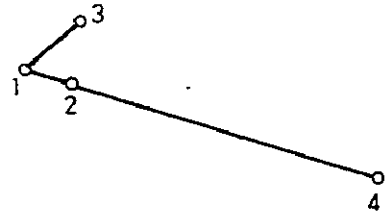


Figure 2 APPEARANCE OF SOURCES VS RELATIVE VEHICLE ATTITUDE

3. FUNCTIONAL FLOW DIAGRAM

The sequencing of functions performed by the Docking and Undocking Navigation Program is described in this section and illustrated by the functional flow diagram in Fig. 3.

The program is called periodically by the Docking and Undocking Guidance Program. The first function performed is to identify the four sources from the two facts discussed in Section 2. Next, the appropriate sources are selected and the unique relative position and attitude solution is determined. Included in this solution is the relative position of the two docking hatches. The final step is to compute velocity and angle rates by differencing two solutions for position and angle.

4. PROGRAM INPUT-OUTPUT

The required inputs to the program are the four sets of azimuth and elevation angles of the four sources relative to the docking sensor; and two indicators, the first of which indicates which of the two combinations of three sources (1, 2, and 3) or (1, 2, and 4) have been selected, and the second is used in the rate calculations. The outputs of the program are solutions for the relative position of the two docking hatches, the rotation angles between the two vehicles, and the rates of change of these quantities.

Input Parameters

(a_1, e_1) (a_2, e_2) (a_3, e_3) (a_4, e_4)	Four sets of paired azimuth and elevation angles but not identified with any of the four sources

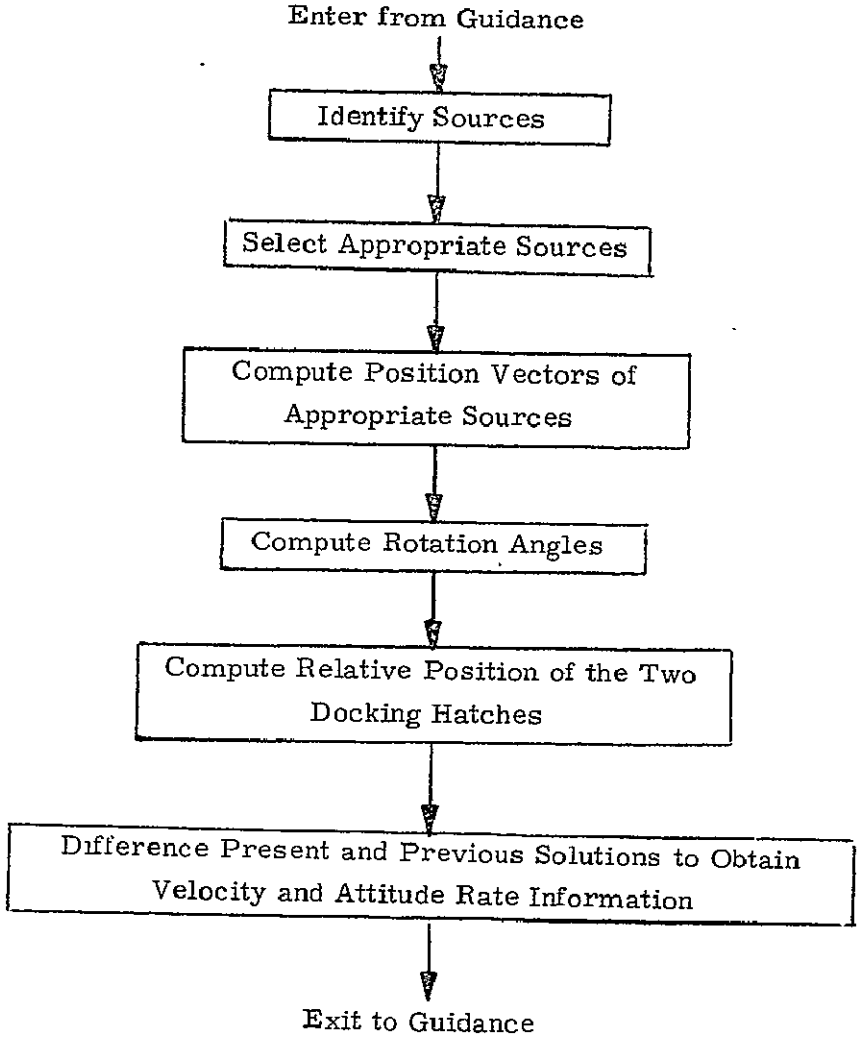


Figure 3 FUNCTIONAL FLOW DIAGRAM

The elevation angle is the angle between the line-of-sight and the XY plane of the shuttle coordinate system. The azimuth angle is the angle between the X axis of the shuttle coordinate system and the projection of the line-of-sight on the XY plane. See Figure 4.

p Source set indicator $= \begin{cases} 2 \text{ if selected source set is } (1,2,3) \\ 4 \text{ if selected source set is } (1,3,4) \end{cases}$

f Rate indicator = Number of cycles separating differenced solutions in rate calculations.

Output Parameters

$\underline{r}_S$ Position vector of target vehicle docking hatch relative to shuttle docking hatch in shuttle coordinates

$\dot{\underline{r}}_S$ Rate of change of $\underline{r}_S$

$\gamma = (\gamma_1, \gamma_2, \gamma_3)$ = Rotation angles

$\dot{\gamma} = (\dot{\gamma}_1, \dot{\gamma}_2, \dot{\gamma}_3)$ = Rates of change of rotation angles

5. DESCRIPTION OF EQUATIONS

The computational sequence during the Docking and Undocking Navigation Program and the related equations are described in this section. These equations are recomputed every guidance cycle.

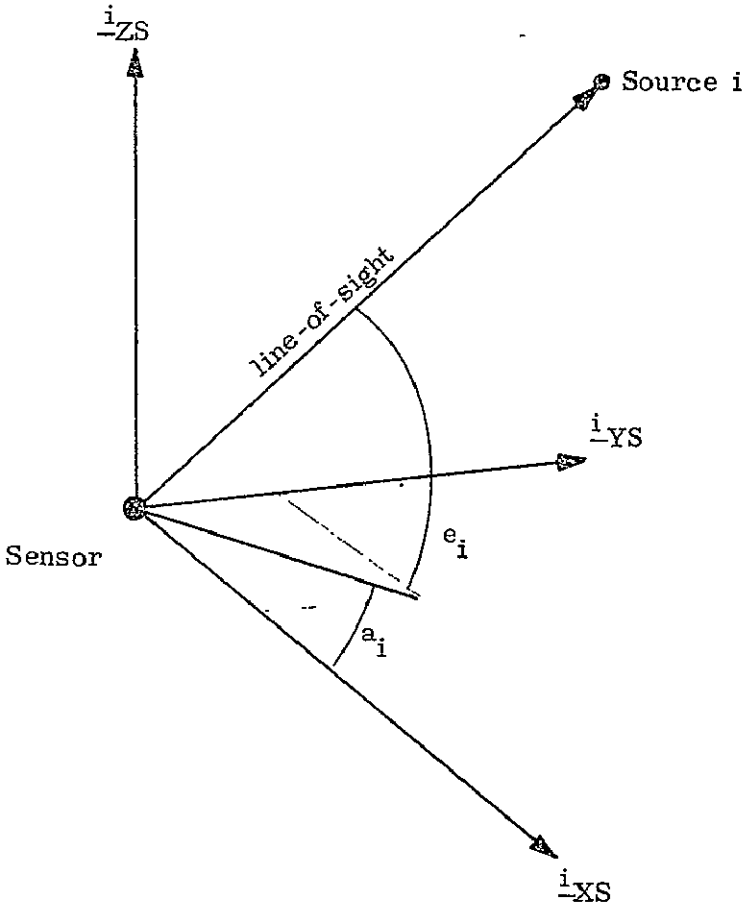


Figure 4 DEFINITION OF AZIMUTH AND ELEVATION ANGLES

5.1 Source Identification (TBD)

The first step in the program is to associate each of the four sets of azimuth and elevation angles with a particular source. The procedure for performing the association is based on the two facts discussed in Section 2; i.e.,

1) Sources 1, 2, and 4 lie on a straight line.

2) The observed distance between sources 1 and 2 has the same ratio with respect to the observed distance between sources 1 and 4.

The algorithm used is TBD.

5.2 Angles Between Lines-of-Sight

The cosines of the three angles between the lines-of-sight from the sensor to the sources in the selected set (based on indicator p) are computed from

$$\cos \theta_{ij} = \cos e_i \cos e_j \cos (a_i - a_j) + \sin e_i \sin e_j$$

for

$$ij = 13,\ 1p,\ \text{and}\ 3p$$

5.3 Distances to Sources

Let $\underline{r}_1$, $\underline{r}_2$, $\underline{r}_3$, and $\underline{r}_4$ be the vectors from the sensor to the four sources. The magnitudes of the three vectors associated with the selected sources satisfy

$$r_3 = r_1 \cos\theta_{13} + \sqrt{r_{13}^2 - (1 - \cos^2\theta_{13})\, r_1^2}$$

$$r_2 = r_3 \cos\theta_{32} - \sqrt{r_{32}^2 - (1 - \cos^2\theta_{32})\, r_3^2}$$

$$r_1 = r_2 \cos\theta_{12} - \sqrt{r_{12}^2 - (1 - \cos^2\theta_{12})\, r_2^2}$$

or

$$r_3 = r_1 \cos\theta_{13} + \sqrt{r_{13}^2 - (1 - \cos^2\theta_{13})\, r_1^2}$$

$$r_4 = r_3 \cos\theta_{34} + \sqrt{r_{34}^2 - (1 - \cos^2\theta_{34})\, r_3^2}$$

$$r_1 = r_4 \cos\theta_{14} - \sqrt{r_{14}^2 - (1 - \cos^2\theta_{14})\, r_4^2}$$

These equations are solved by an iterative interval-halving process in which s_1, a trial value of r_1, is used as input to compute an output value of r_1 by means of

$$r_3 = r_1 \cos\theta_{13} + \sqrt{r_{13}^2 - (1 - \cos^2\theta_{13})\, s_1^2}$$

$$r_p = r_3 \cos\theta_{3p} \mp \sqrt{r_{3p}^2 - (1 - \cos^2\theta_{3p})\, r_3^2} \qquad (1)$$

$$r_1 = r_p \cos\theta_{1p} - \sqrt{r_{1p}^2 - (1 - \cos^2\theta_{1p})\, r_p^2}$$

where the upper and lower signs correspond, respectively, with p=4 and p=2. Agreement between s_1 and r_1 indicates a correct solution.

The iteration is initiated by computing the maximum possible value for r_1 based on the sensor measurements

$$r_M = \frac{r_{13}}{\sqrt{1 - \cos^2 \theta_{13}}}$$

Then, using r_M as the first value for s_1; values for r_3, r_p, and r_1 are computed from Eq. (1). During these calculations, it is possible for one of the radicands to be negative, in which case the selected value of s_1 is too large. If this occurs, the value of s_1 is halved, and the computations are repeated. The process continues until three real numbers are obtained for r_3, r_p, and r_1 as functions of s_1. (It should be noted that once a value of s_1 which produces a real solution has been determined, then all smaller values of s_1 will also yield a real solution.)

The difference between the input and output values of r_1 is computed from

$$\Delta r = s_1 - r_1 \qquad (2)$$

Assuming that a negative radicand did not occur, the value of s_1 is halved, and new values for r_3, r_j, r_1, and Δr are computed. If no sign change in Δr occurs, then s_1 is again halved and the procedure repeated until a polarity change in Δr occurs. When the sign change does occur, the last selected value of s_1 is increased by one half its value and the polarity of the new resulting Δr is tested. This interval-halving procedure, increasing or decreasing s_1 by one half of each increment taken, is repeated until the difference Δr is less than the desired error level ϵ.

This procedure is based on the fact that Eqs. (1) and (2) represent Δr as a continuous function of s_1. If there are two values of s_1, one of which yields a positive value of Δr and the other a negative value, then there is some value of s_1 between these two values for which Δr is zero - the desired condition.

During the first calculation of Δr, if a negative radicand results, then a special procedure must be followed after the value of s_1 which yields real values is found. Whereas in the first case it is known that the correct value of s_1 is not larger than r_M, in this case the solution could be larger than the value of s_1 for which real (but incorrect) values of r_3, r_p, and r_1 resulted. This ambiguity is resolved by performing one pass through Eqs. (1) and (2) with s_1 equal to zero. Comparison of the sign of the resulting Δr with the sign of the previous Δr indicates whether s_1 should be increased or decreased. This same procedure is used if, during an increase in s_1, a negative radicand occurs.

The details of the iterative procedure are shown in the flow diagrams of Section 6.

5.4 Source Position Vectors

The position vectors of the three selected sources are obtained from

$$\underline{r}_{iS} = r_i \begin{pmatrix} \cos e_i & \cos a_i \\ \cos e_i & \sin a_i \\ \sin e_i & \end{pmatrix} \qquad (i = 1, 3, p)$$

where the subscript S denotes shuttle coordinates.

5.5 Transformation Matrix

The transformation matrix M from shuttle to target vehicle coordinates is computed from

$$M^T = AB$$

where

$$A = (\underline{r}_{13S} \quad \underline{r}_{1pS} \quad \underline{r}_{13S} \times \underline{r}_{1pS})$$

$$B = (\underline{r}_{13T} \quad \underline{r}_{1pT} \quad \underline{r}_{13T} \times \underline{r}_{1pT})^{-1}$$

$$\begin{bmatrix} \frac{1}{S^2}\begin{pmatrix} -C \\ -\frac{C^2}{S} \\ \frac{1}{S} \end{pmatrix} & \frac{1}{KS^2}\begin{pmatrix} C \\ \frac{C^2 - S^2}{S} \\ -\frac{C^2}{S} \end{pmatrix} & \frac{1}{K\rho S^2}\begin{pmatrix} 1 \\ \frac{C}{S} \\ -\frac{C}{S} \end{pmatrix} \end{bmatrix}$$

$$C = \cos 40^{o}$$

$$S = \sin 40^{o}$$

$$K = \begin{cases} 0.4 & \text{if } p = 2 \\ 2.5 & \text{if } p = 4 \end{cases}$$

$$\underline{r}_{13S} = \underline{r}_{3S} - \underline{r}_{1S}$$

$$\underline{r}_{1pS} = \underline{r}_{pS} - \underline{r}_{1S}$$

5.6 Rotation Angles

The rotation angles γ_1, γ_2, and γ_3 are obtained from

$$\gamma_2 = \sin^{-1}(m_{31})$$

$$\gamma_3 = -\sin^{-1}\left(\frac{m_{21}}{\cos \gamma_2}\right)$$

$$\gamma_1 = -\sin^{-1}\left(\frac{m_{32}}{\cos \gamma_2}\right)$$

where m_{31}, m_{21}, m_{32} are elements of M according to

$$M = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix}$$

5.7 Relative Position Vector Between Docking Hatches

The position of the target vehicle docking hatch relative to the shuttle docking hatch is computed from

$$\underline{r}_S = -\underline{y}_S + \underline{r}_{1S} + M^T \underline{z}_T$$

where $\underline{y}$ and $\underline{z}$ are the locations of the shuttle and target vehicle docking hatches relative to their respective coordinate system origins, and the S and T subscripts indicate shuttle and target vehicle coordinates. Note that $\underline{y}_S$ and $\underline{z}_T$ are fixed constants.

5.8 Velocity and Attitude Rate

The estimated relative velocity and estimated relative attitude rate of the two vehicles are computed by differencing the current relative position and attitude solution with the solution f cycles in the past as follows:

$$\dot{\gamma}_i = \left[\gamma_i(t) - \gamma_i(t - f\Delta t) \right] / f\Delta t \qquad (i = 1, 2, 3)$$

$$\dot{\underline{r}}_S = \left[\underline{r}_S(t) - \underline{r}_S(t - f\Delta t) \right] / f\Delta t$$

During the first cycle, no rate information can be computed, and during cycles 2 through f, the current and the first solutions are used in the calculations.

This procedure provides smoother estimates of the rates from cycle to cycle than if successive values of relative position and attitude were used.

6. DETAILED FLOW DIAGRAMS

This section contains detailed flow diagrams of the Docking and Undocking Navigation Program.

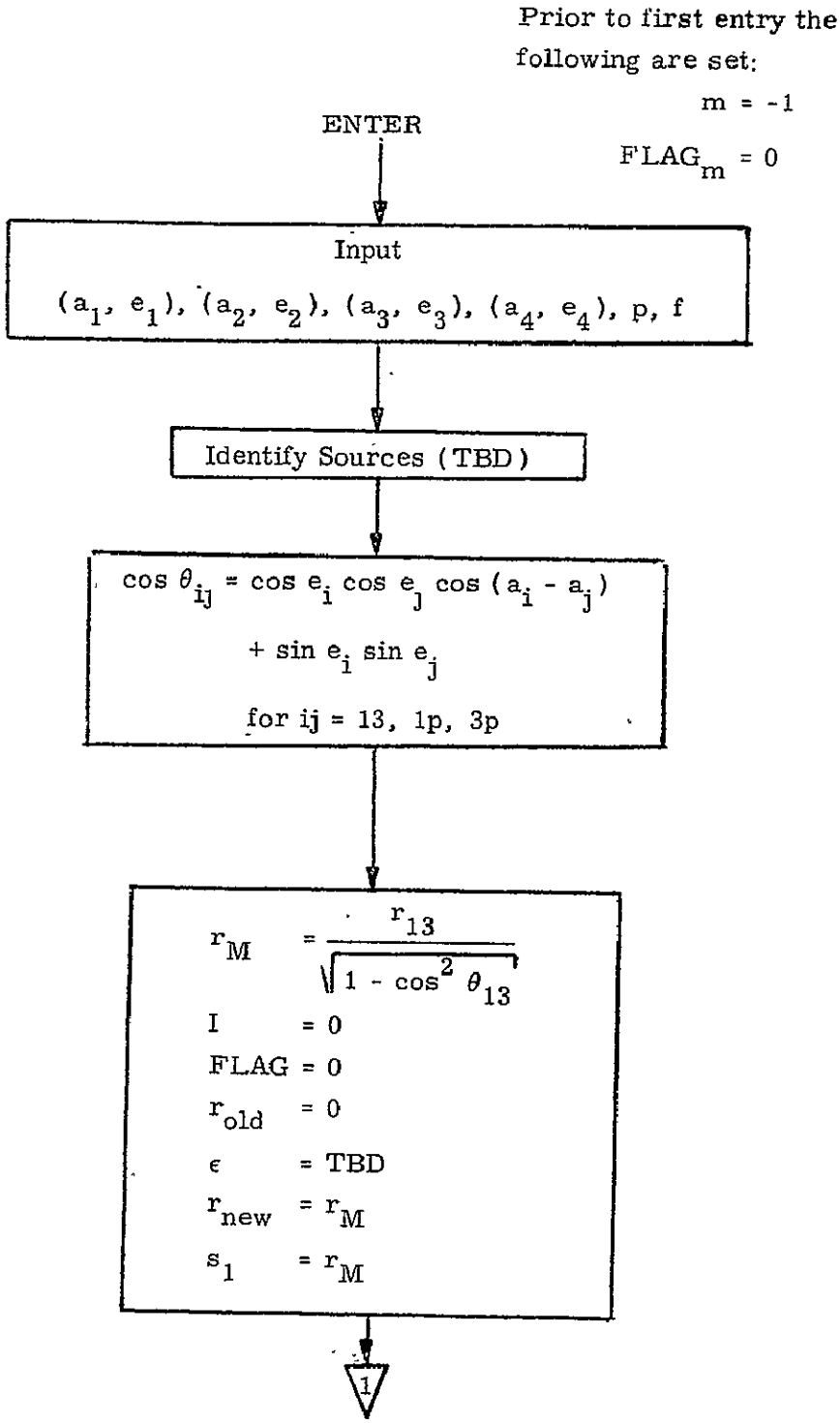


Figure 5a DETAILED FLOW DIAGRAM

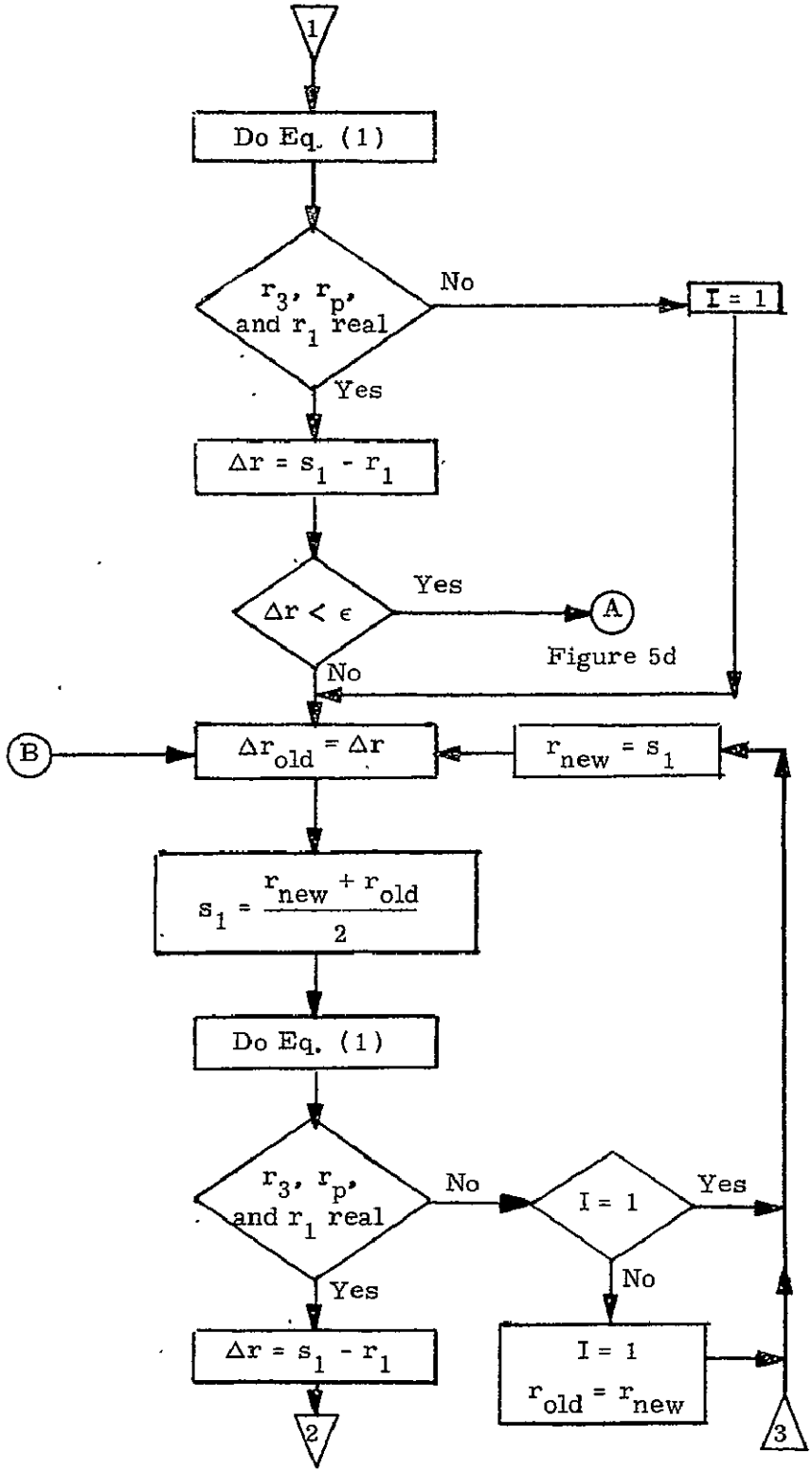


Figure 5b DETAILED FLOW DIAGRAM

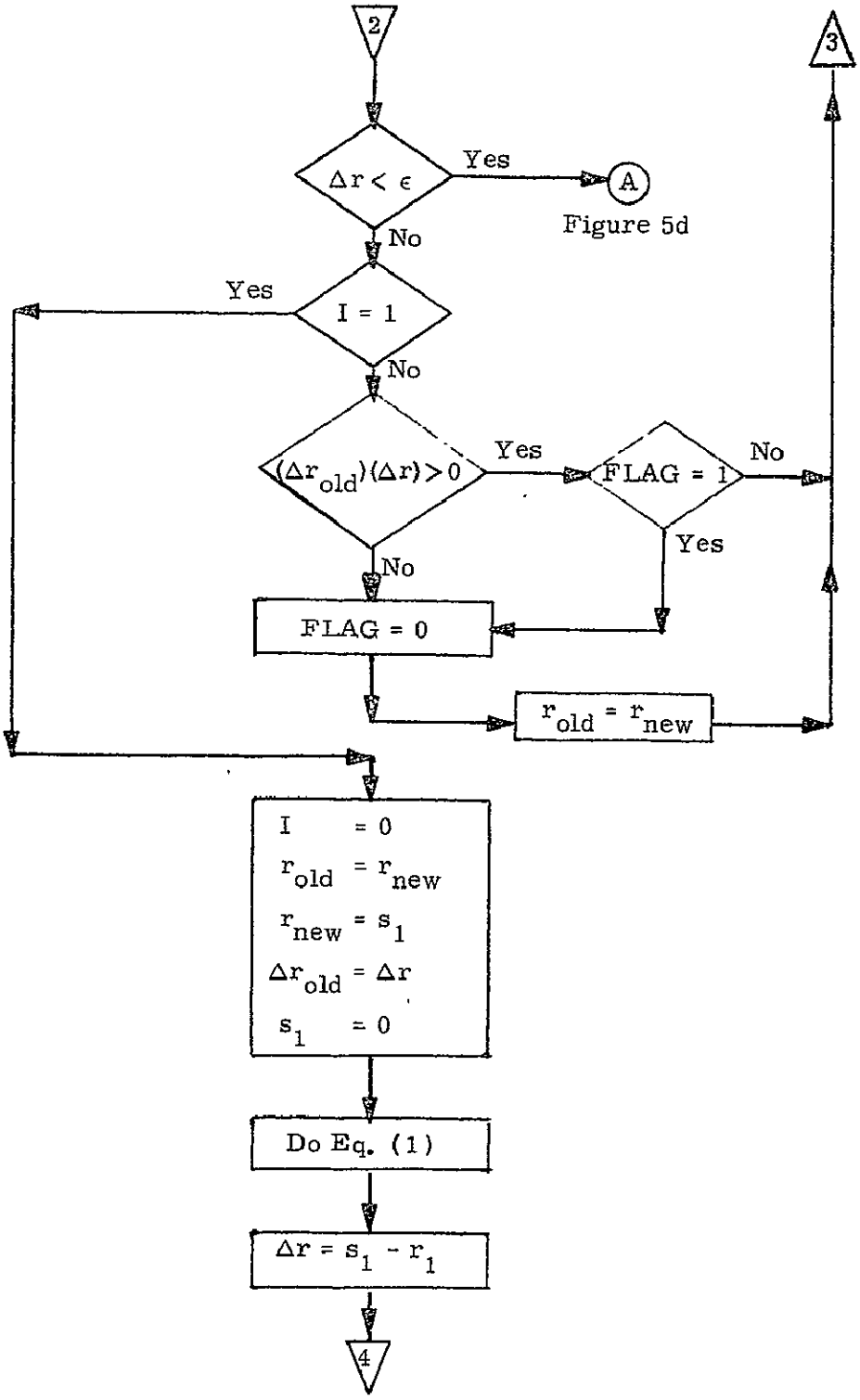


Figure 5c DETAILED FLOW DIAGRAM

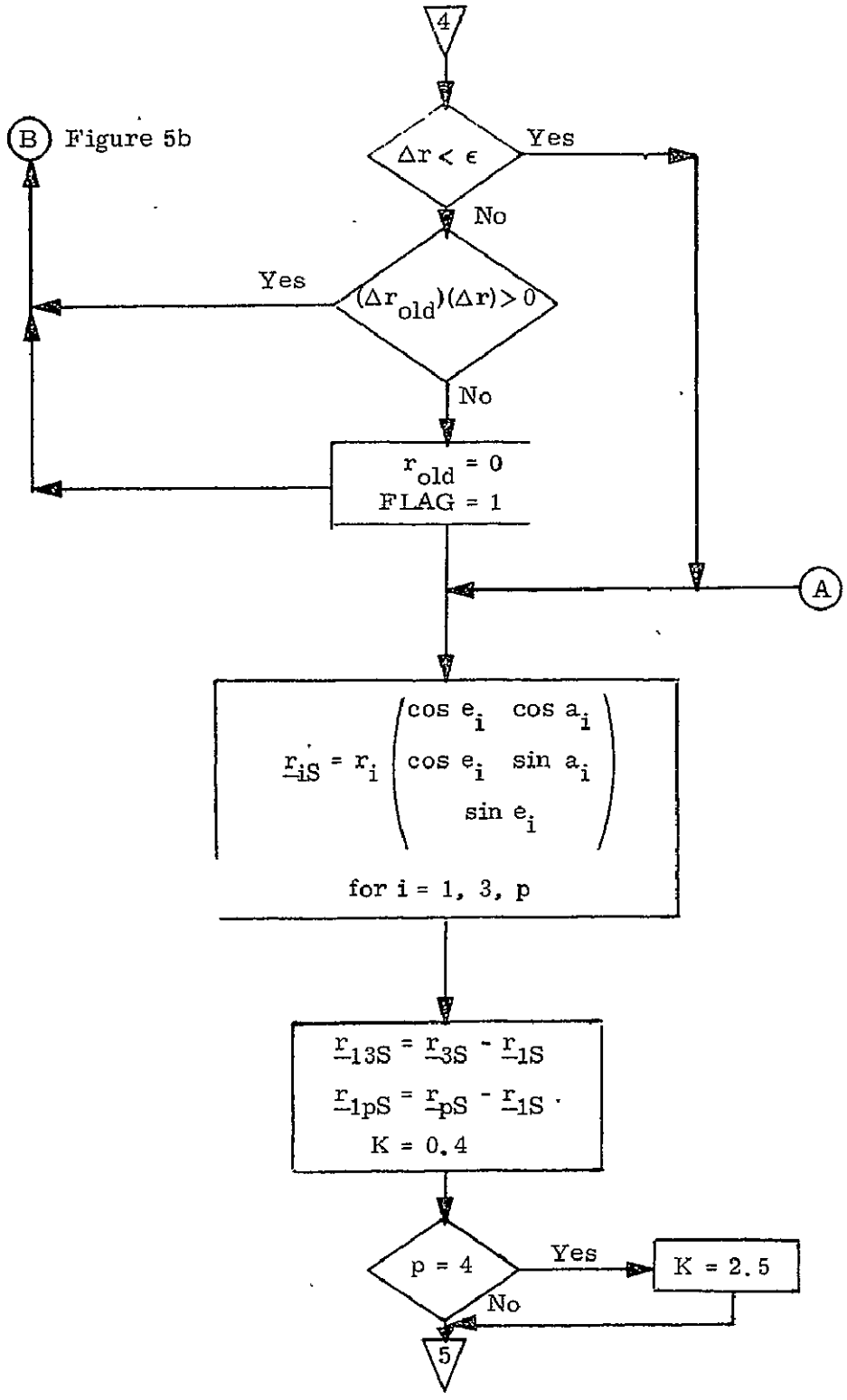


Figure 5d DETAILED FLOW DIAGRAM

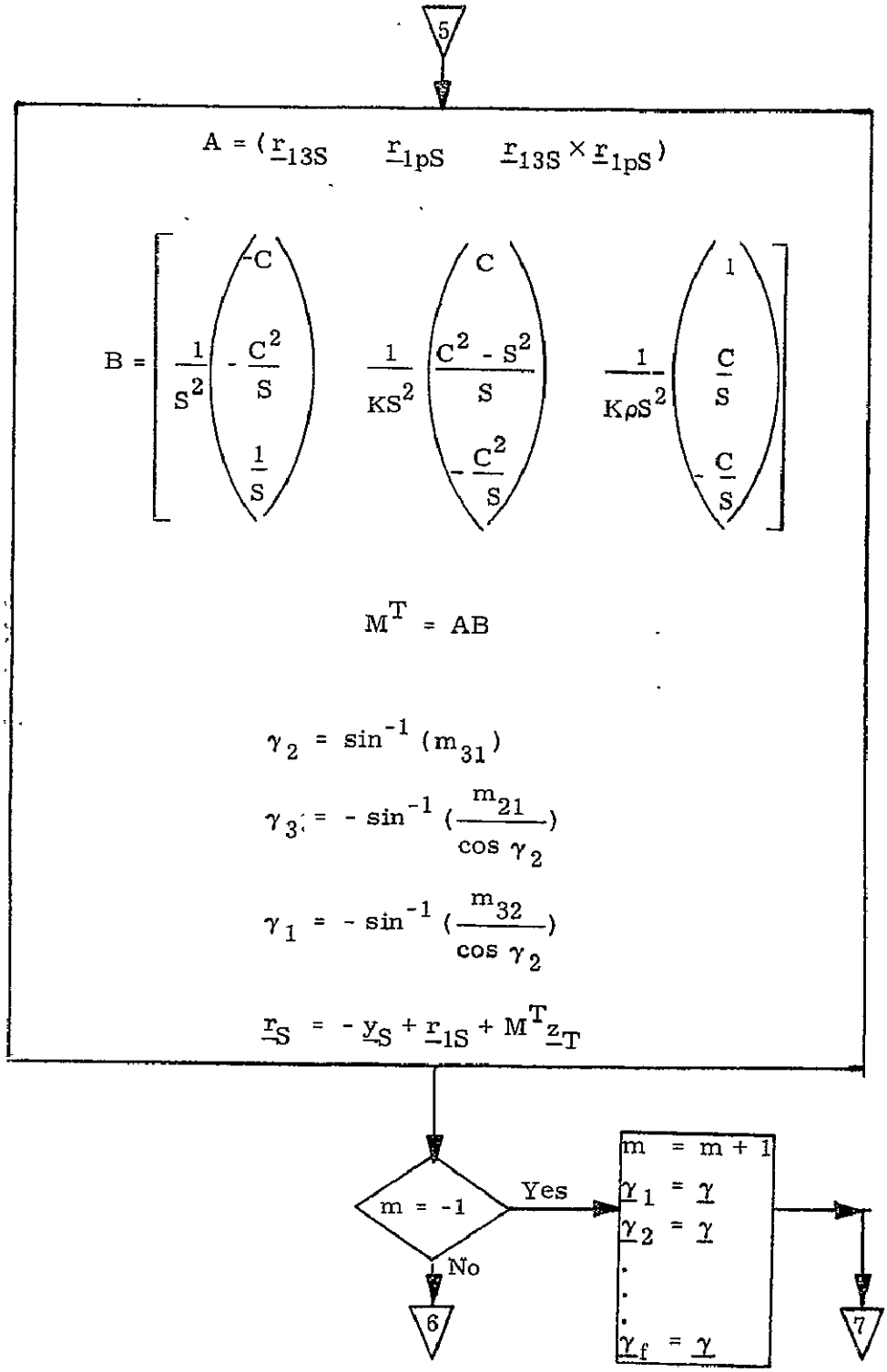


Figure 5e DETAILED FLOW DIAGRAM

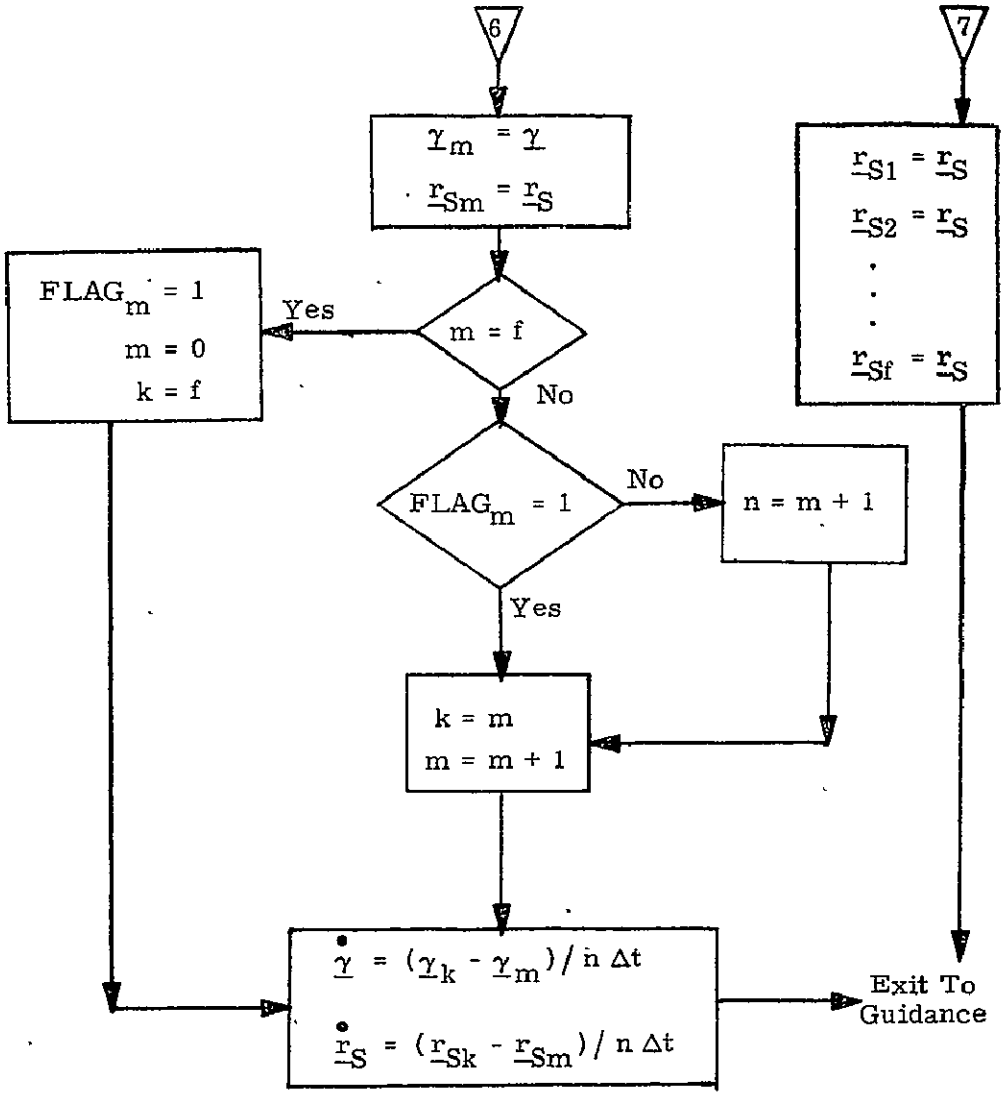


Figure 5f DETAILED FLOW DIAGRAM

7. SUPPLEMENTARY INFORMATION

The Docking and Undocking Navigation Program described in this report has been operated as an open loop, simulating the sensor-target geometry, the sensor, and the computations yielding as outputs the relative state vector and attitude between vehicles. The program is valid and the configuration chosen performs as expected. A chart and tabulated results appear in Ref. 1.

It is planned to continue the present program effort to provide a closed loop capability which will include a guidance law* for Docking and Undocking, and an autopilot with capability to operate with the guidance law and the vehicle and engine characteristics. The navigation program will be modified to incorporate Kalman Filtering which should enhance the navigation and provide better assessment of the relative state vector. It is also planned to add a scale change or zoom capability to the sensor model used such that improvement in the accuracy of the state vector can be achieved at long ranges.

*A simplified guidance law will be implemented initially with growth to more sophisticated guidance laws as deemed necessary.

9.10.1 Docking and Undocking Navigation

Reference

1. Blanchard, Earle P., NAS 9-10268 Automatic Docking GN& C Equation Development, 21 December 1970, 70-408L-7.

SPACE SHUTTLE

GN&C SOFTWARE EQUATION SUBMITTAL

Software Equation Section Docking and Undocking Submittal No. 3

Function Automatic Docking Control Law

Module No. OC4 Function No. -4, -6 (MSC 03690)

Submitted By: E. T. Kubiak Co. MSC/GCD
(Name)

Date: January 26, 1971

NASA Contact: W. H. Peters Organization EG2
(Name)

Approved by Panel III K. J. Cox K.J.Cox Date January 26, 1971

Summary Description: The automatic docking control laws provide the attitude and translational commands for the docking procedure which is defined to begin at a range of 1000 ft. The procedure involves two sequential control tasks. The first brings the orbiter within stationkeeping range (≈150 ft.) and the second accomplishes docking with minimum docking hardware contact position dispersions.

Shuttle Configuration: (Vehicle, Aero Data, Sensor, Et Cetera)
No docking sensor configuration is defined but jet accelerations are assumed.

Comments:

(Design Status) The design is in the conceptual stage with required filters still to be designed.

(Verification Status) Will be simulated on an orbiter docking engineering simulator.

Panel Comments: The range, range rate, and relative attitude computations in these equations must be coordinated with similar computations in the Docking and Undocking Navigation equations.

9.10.2 Automatic Docking Control Law

1. Introduction

The docking procedure is defined to begin at a range of approximately 1000 feet. From this point, there are two sequential control tasks. The first task is to bring the orbiter within stationkeeping range, say 150 feet, with a lateral displacement of 10 feet or less from the desired approach path and relative rates of one half ft/sec/axis or less. The second control task is a successful docking with minimum docking hardware contact position dispersion and transmitted impulses.

Significant improvements over the original control law (Reference 1) are (1) minimum use of relative angle measurements which have large errors, (2) direct control of the probe tip which provides tighter control, and (3) reduced time for the docking procedure due to improved logic. The first two points are also discussed in the reference.

In generating this control law, the following assumptions have been used as ground rules:

a) Measured quantities available from the sensors are range, R; LOS (line-of-sight) angles for pitch, α, and yaw, β; and relative orbiter/target attitude (ϕ_R, θ_R, Ψ_R). As the orbiter is to be autonomous, no other information (e.g., target position or attitude) is available from ground tracking or computer initialization.

b) Range and LOS angle measurements will have greater accuracy than relative attitude angle measurements (particularly at longer ranges).

c) It is desirable to have at least a brief station keeping period prior to the final phase (assumed to begin at 100 ft range) of docking, providing the opportunity for a final check of thrusters, docking mechanisms, GN&C systems, and sensor systems.

d) The docking procedure begins at approximately 1000 ft range and should conclude in 5 to 10 minutes (plus any time spent in the station keeping mode).

e) Sensor measurements provide the only available information with regards to the passive vehicle's relative state (no data link).

f) It is assumed that the target vehicle is under attitude control and that any target vehicle motion due to attitude control limit cycling is negligible (a good assumption for CMG control).

Nomenclature

a	Translational acceleration
α	LOS pitch angle
β	LOS yaw angle
K	Factor in phase-plane switching lines representing the relative importance of time vs. fuel minimization
ℓ	Distance along +X body axis from orbiter c.g. to sensor location
LOS	Line-of-sight
ϕ_R	Relative orbiter/target roll attitude
Ψ_R	Relative orbiter/target yaw attitude
R	Range
T	Total closure time
θ_R	Relative orbiter/target pitch attitude
u_Ψ	$\pm\Psi$ yaw torques
u_θ	$\pm\theta$ pitch torques
u_y	$\pm$y thruster forces
u_z	$\pm$Z thruster forces
$\underline{\omega}_{BODY}$	Orbiter body rate

9.10.2 Automatic Docking Control Law (con't)

$\underline{\omega}_{LOS}$	LOS angular rate
$\dot{X}_0$	Initial separation distance
$\ddot{X}_0$	Initial closing rate
X_{cg}, Y_{cg}, Z_{cg}	C.G. position errors
X_{LCS}, Y_{LCS}, Z_{LCS}	Position errors in LOS coordinate system
X_p, Y_p, Z_p	Probe position errors

2. Coordinate System Definition

Before proceeding to equation formulation, the following coordinate systems must be defined (see also Figure 1).

a) Body coordinate system (BCS) - origin at c.g. of +X axis towards nose along centerline, +Y towards right wing, +Z down.

b) Sensor coordinate system (SCS) - sensor and docking mechanism location assumed coincident along +X body axis at distance ℓ from orbiter c.g., which also defines the origin location. Direction of axes, same as body axes.

c) LOS Coordinate System (LCS) - origin same as the SCS. Direction of axes defined by LOS pitch and yaw rotations from SCS +X axis.

d) Target coordinate system (TCS) - origin located at passive vehicle docking mechanism assumed coincident with reflectors. -X axes defines the desired final approach path. Y and Z complete the right hand system

3. Functional Flow Diagram

The sequencing of functions performed by the Automatic Docking Control Law is described in this section and illustrated in the functional flow diagram in Figures 2a and 2b.

The program calculates the probe to target vector and determines whether Phase 1 or Phase 2 control is desired. If Phase 1 control is required, calculate the position and velocity errors for phase-plane control using the sensor measured pitch and yaw LOS angles, c.g. to target range, and the estimated vehicle to target attitude. Based on these values for position and velocity errors, enter the X, Y, and Z-axis phase-plane control logic and compute translational commands.

For Phase 2 control, compute the range position error using the sensor measured pitch and yaw LOS angles, c.g. to target range, and the estimated vehicle to target attitude. Passing this signal through a

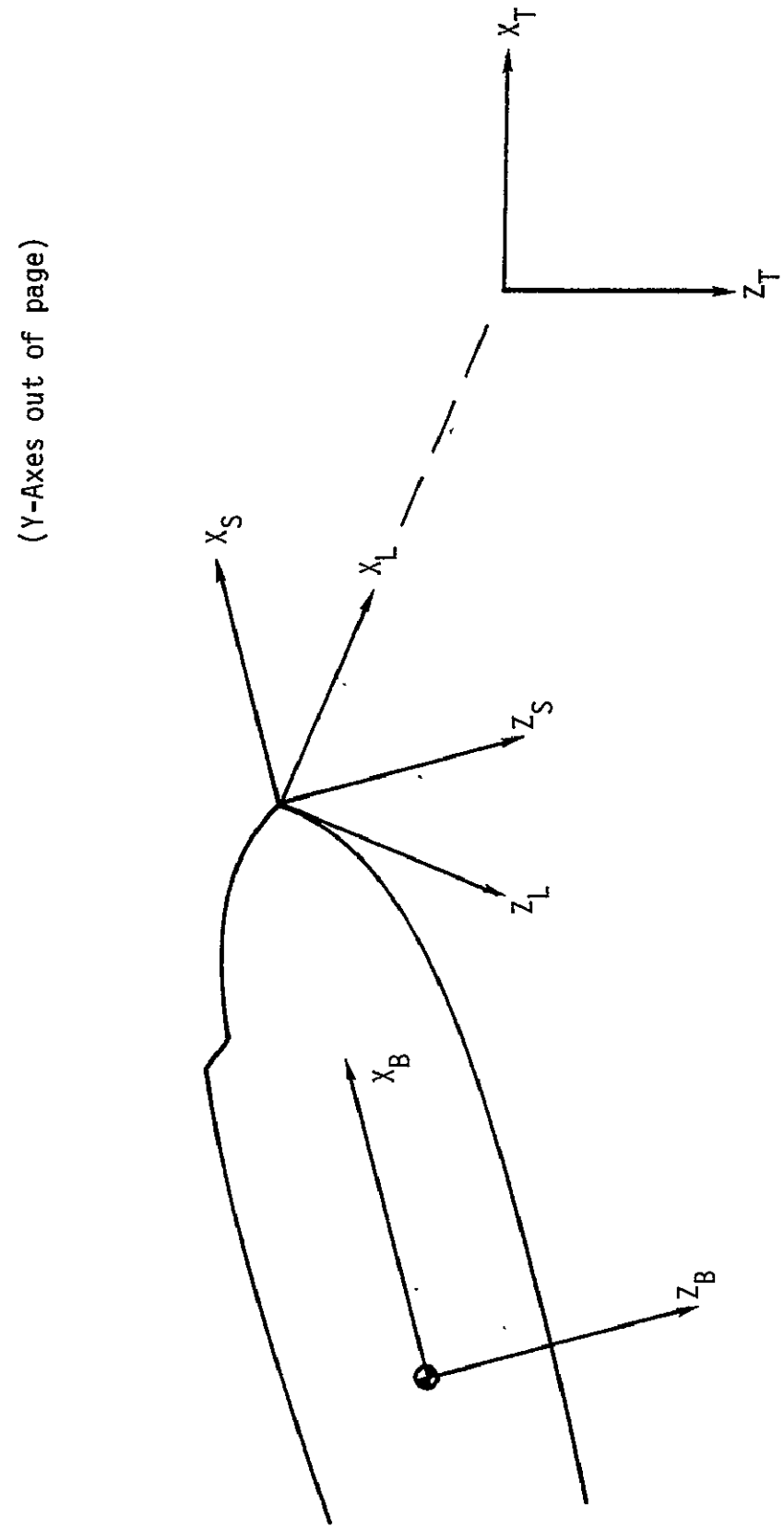


Figure 1

Coordinate System Definition

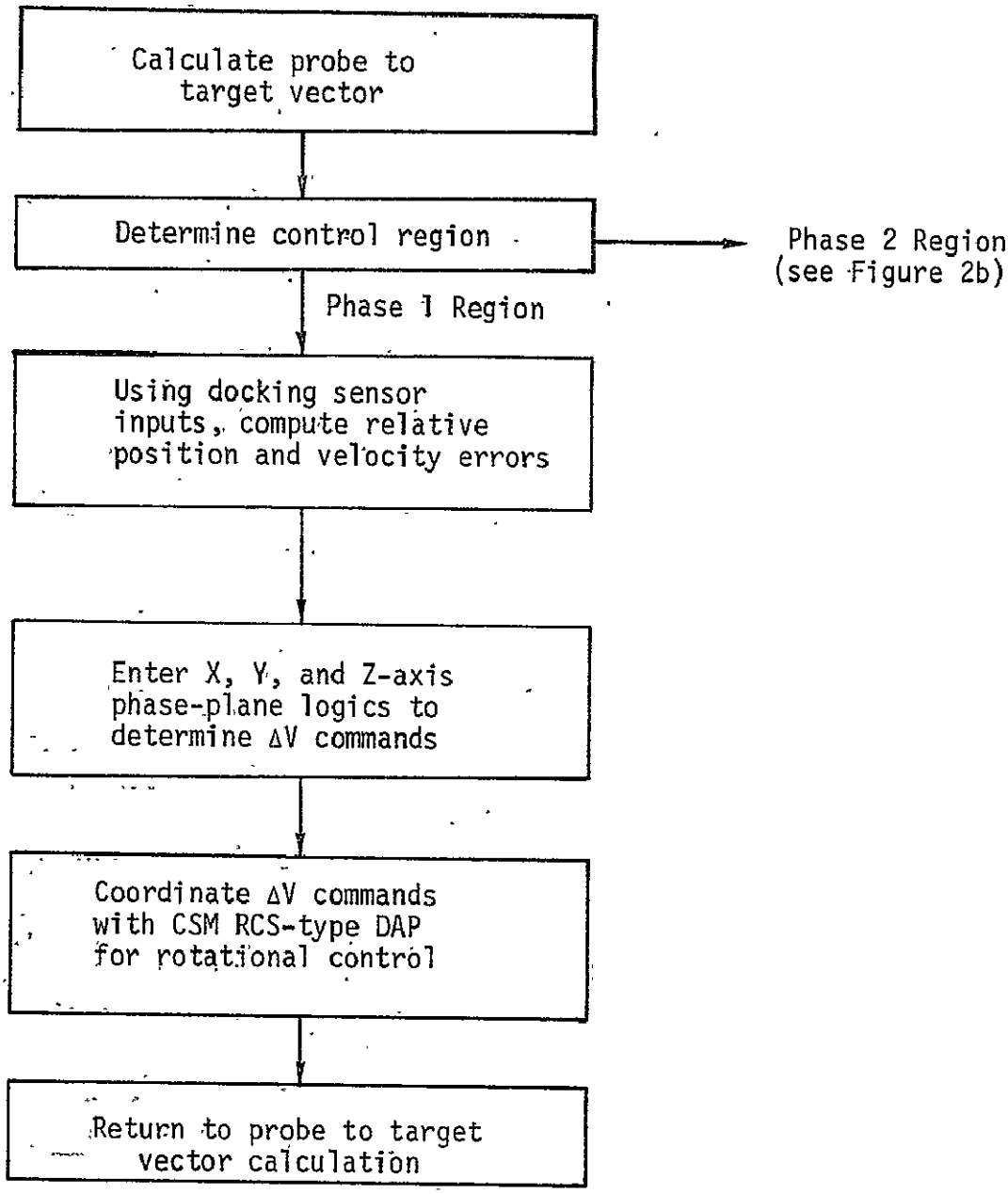


Figure 2a. Phase I Control Functional Flow Diagram

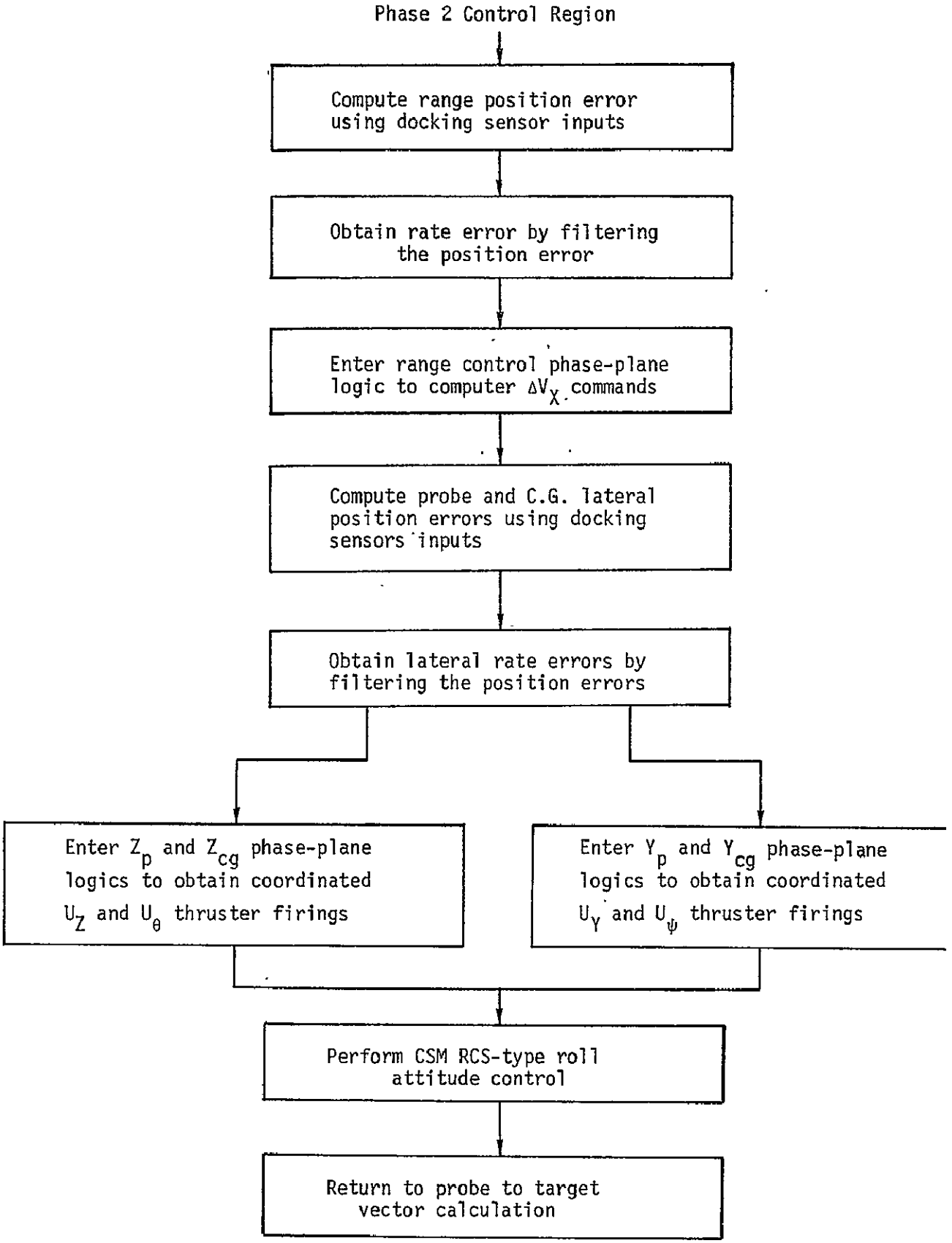


Figure 2b. Phase II Control Functional Flow Diagram

filter, obtain the rate error. Enter the range control phase-plane logic to compute jet firing times for X-axis translation control. Compute the lateral position errors of the probe and c.g. and determine the respective rates by a filtering routine. Enter the c.g. and probe phase-plane logics to obtain coordinated ±Z thruster firings and ±θ pitch torques for Z-axis and pitch control, and coordinated ±Y thruster firings and ±Ψ yaw torques for Y-axis and yaw control.

4. Program Input-Output

The docking sensors have not been baselined, but in this development, basic inputs have been identified. These inputs include range, LOS pitch and yaw angles, relative orbiter/target attitude, body rates, the distance between the orbiter c.g. and probe as measured along the +X body axis, and estimates of the RCS jet control authorities. The outputs of the program are RCS jet firing times.

Input Parameters

R	Range between orbiter and target vehicle
α	LOS pitch angle
β	LOS yaw angle
ϕ_R	Relative orbiter/target roll angle
θ_R	Relative orbiter/target pitch angle
Ψ_R	Relative orbiter/target yaw angle
$\underline{\omega}_{BODY}$	Orbiter angular rates
ℓ	Distance between orbiter c.g. and probe as measured along +X body axis
a	Translational acceleration capability of the orbiter (lateral and ±X body)
U_Y, U_Z	RCS translational acceleration along Y and Z axes

u_θ, u_ψ	RCS angular acceleration about pitch and yaw axes

Output Parameters

$t_{\text{Region X}}$	RCS jet firing time (and sign) for various regions of the phase-plane logics

5. Description of Equation

5.1 Phase I Control

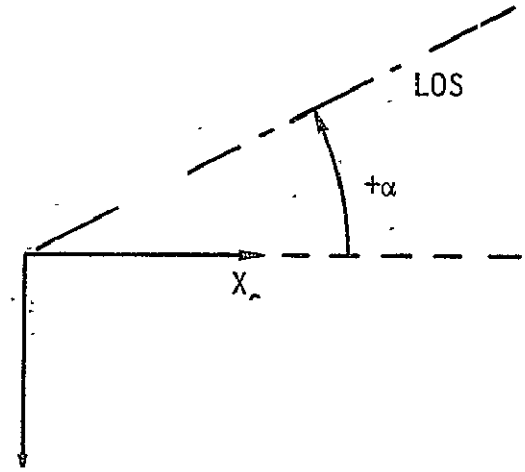
Phase 1 is defined as the control period during which the orbiter is brought from some post rendezvous state (range about 1000 feet) into the stationkeeping state. In the sequence of control actions, the first step is to define as a pitch/yaw reference, the LOS vector from the sensor to the target (i.e., $\alpha = \beta = 0$, see Figure 3).

The roll reference is defined such that $\dot{Z}$ is parallel to Z_T (i.e., the relative roll angle is zero). The attitude error, (ϕ_R, α, β) will change slowly due to relative motion and vehicle body rotation. This error will be measured and filtered once per second. Control logic will be basically the same as the CSM RCS DAP with a deadband of 5°. When the vehicle's attitude is within the deadband for all three axis translational control is begun.

In the translational control formulation the TCS is considered to be inertial (orbiter mechanics neglected). The control problem is to translate the orbiter from its initial state to a limit cycle region which has as its position reference (-150, 0, 0) in the TCS. The ideal trajectory, time and fuel-wise, is the straight line between the initial condition and (-150, 0, 0) in the TCS. One of the more precise control processes which could be used to follow this trajectory is:

a) Generate displacement and rate vector in the TCS from measurements and matrix computations.

PITCH (X_c, Z_c) PLANE



YAW (X_s, Y_s) PLANE

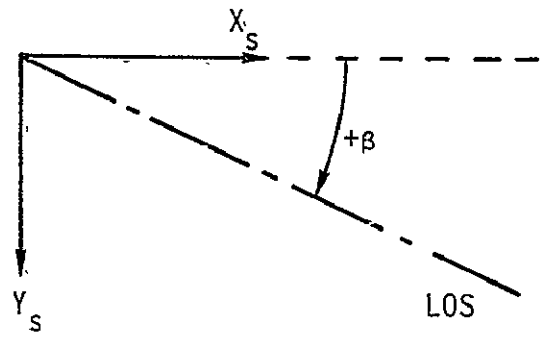


Figure 3

LOS Angle Definition

b) Select a delta V to (1) null velocity component normal to displacement vector, and (2) provide the desired closing rate along the displacement vector.

c) Determine components of delta V in BCS and implement commands.

d) Reiterate computations to null residual errors.

The performance of such a process would be very dependent on the relative angle measurements used in numerous matrix multiplications. As these measurements are not highly accurate, particularly at initialization range (1000 ft or more), another process will be used which performs the same function and requires much less computation.

a) Compute position error and vehicle relative rates in LCS.

b) Input position errors and relative rates to phase plane switching logic to determine delta V commands

c) Recycle according to some selected sample frequency.

For this scheme, Figure 4a shows how the Z_{LCS} position error is determined to be

$$Z_{LCS} = 150 \sin(\alpha + \theta_R) - \ell \sin\alpha$$

Similarly, Figure 4b indicates

$$Y_{LCS} = -150 \sin(\beta + \Psi_R) + \ell \sin\beta$$

Finally, the X position error is

$$X_{LCS} = R + \ell \cos\theta_R \cos\Psi_R - 150$$

The relative velocity of the orbiter with respect to the target vehicle in the LCS is equal to the negative of the derivative of $(\underline{R} + \underline{\ell})$. As $\underline{R}$ is rotating in inertial space with an angular velocity of $(\omega_{LOS} + \omega_{BODY})$, the expression for the derivative is

$$\frac{d}{dt}(\underline{R} + \underline{\ell})_{LOS} = \dot{\underline{R}} + (\underline{\omega}_{LOS} + \underline{\omega}_{BODY}) \times \underline{R} + \underline{\omega}_{BODY} \times \underline{\ell}$$

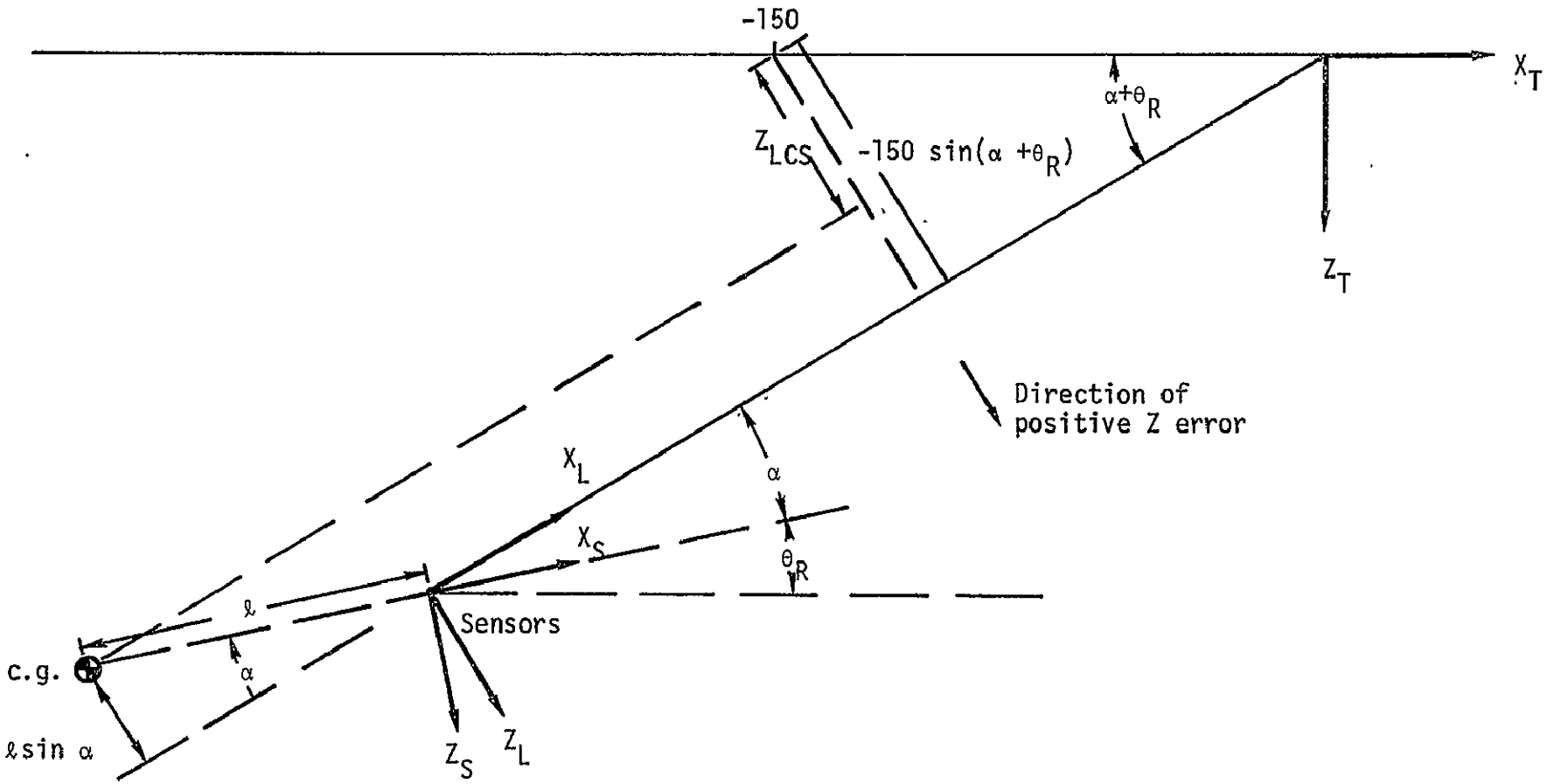


Figure 4a

Calculation of Z_{LCS}

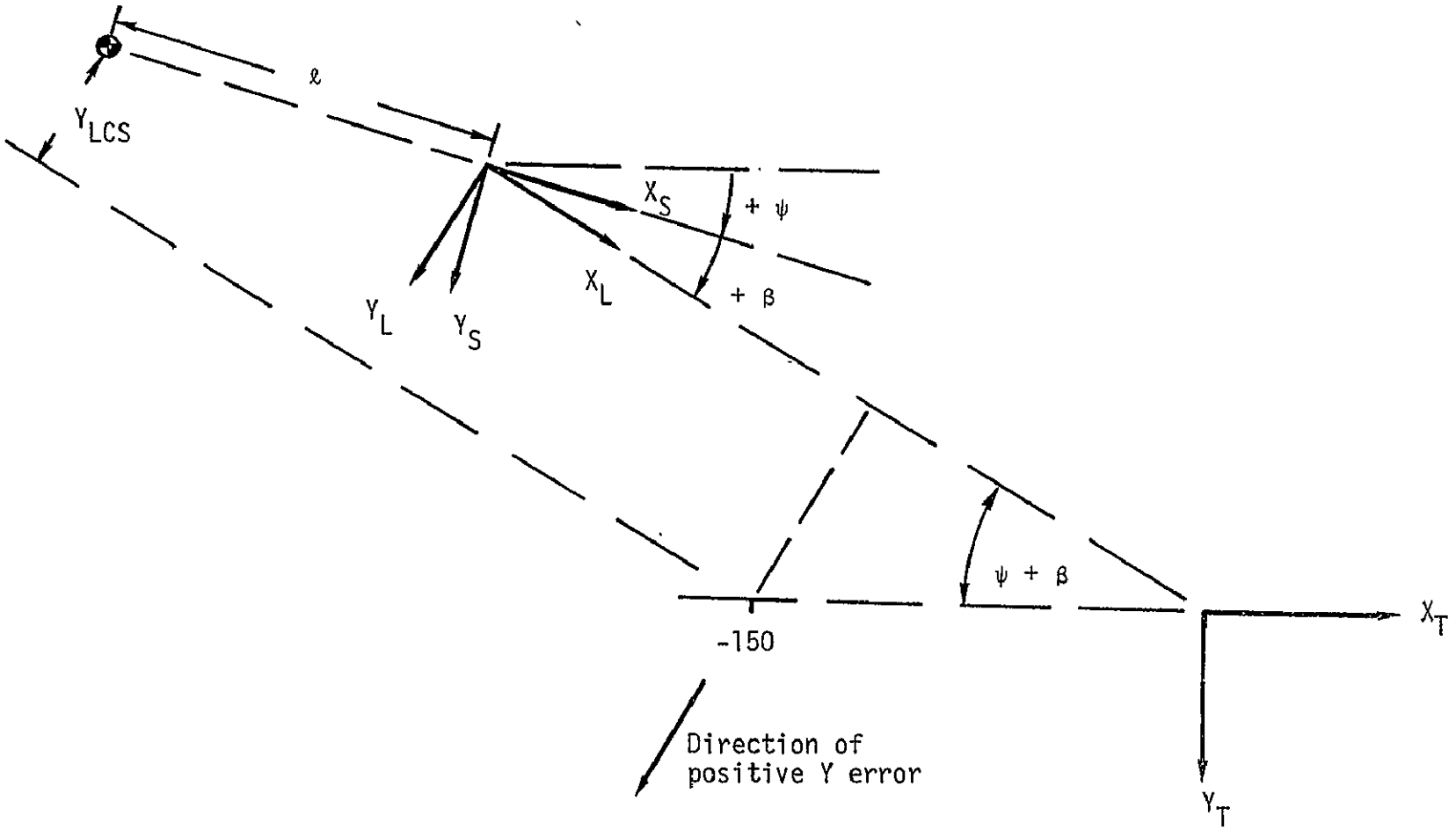


Figure 4b

Calculation of Y_{LCS}

Assuming $\underline{\ell}$ is colinear with the X_{LCS} axis, then $\underline{\ell} = [\ell\ 0\ 0]^T$

Also,

$$\underline{R} = \begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix} \qquad \underline{\omega}_{LOS} = \begin{bmatrix} 0 \\ \alpha \\ \beta \end{bmatrix} \qquad \underline{\omega}_{BODY} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}$$

which yields the following derivatives

$$\dot{X}_{LCS} = -\dot{R}$$

$$\dot{Y}_{LCS} = -R(\dot{\psi}_R + \dot{\beta}) - \ell\dot{\psi}_R$$

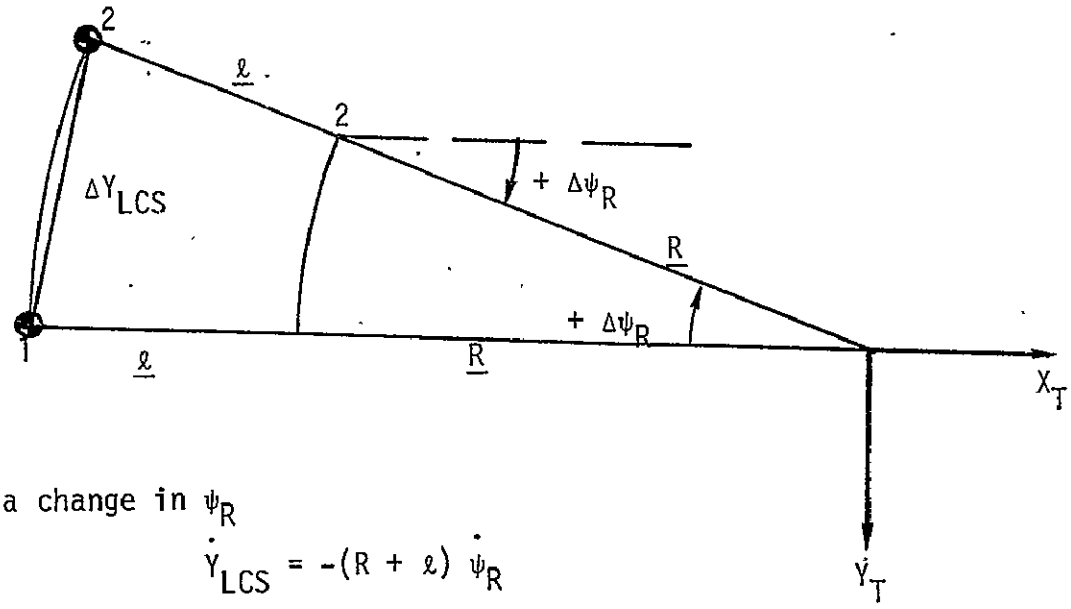
$$\dot{Z}_{LCS} = R(\dot{\theta}_R + \dot{\alpha}) + \ell\dot{\theta}_R$$

Figures 5a and 5b show the geometry relating to the $\dot{Y}_{LCS}$ and $\dot{Z}_{LCS}$ equations. It should be noted in the position and rate equations that R and ℓ are always positive quantities.

Also, as $\dot{\theta}_{BODY}$ and $\dot{\psi}_{BODY}$ very nearly equal $\dot{\theta}_R$ and $\dot{\psi}_R$ and further as the body rates may be known much more accurately than the relative angle rates, $\dot{\theta}_R$ and $\dot{\psi}_R$ may be replaced in the $\dot{Y}_{LCS}$ and $\dot{Z}_{LCS}$ computations by $\dot{\theta}_{BODY}$ and $\dot{\psi}_{BODY}$.

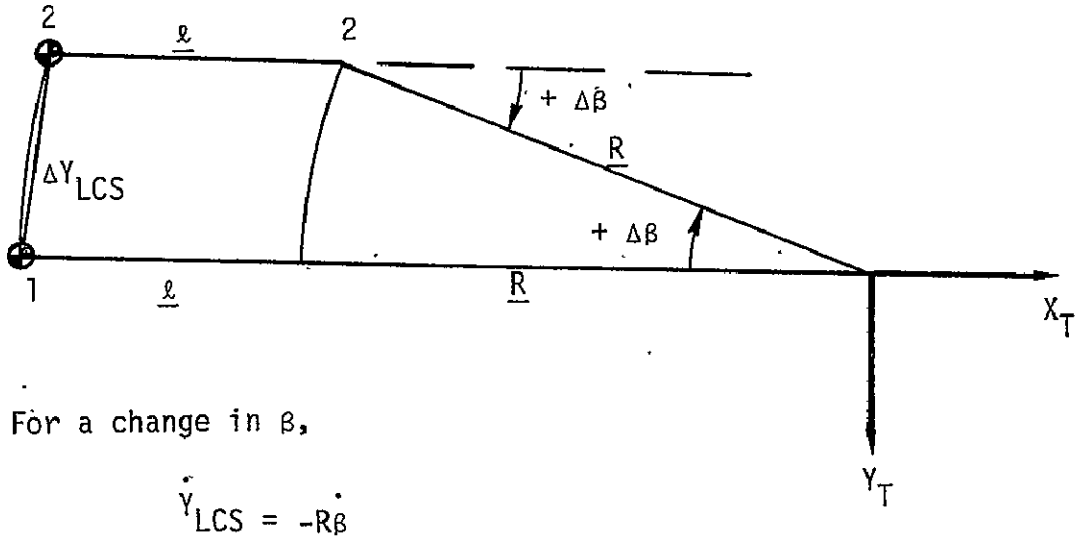
The translational control law is based upon the parabolic switching logic which is the optimal control for minimizing time and fuel for a $1/s^2$ or double integrator plant. Figure 6 illustrates this optimal logic where the available control acceleration is $u = \pm a$.

The factor K in the $\dot{f}_2(X)$ and $\dot{f}_4(X)$ switching curves is the relative importance of time vs. fuel minimization (i.e., increasing K decreases time and increases fuel and vice versa). As $K \to \infty$, $\dot{f}_2(X) \to \dot{f}_1(X)$ and $\dot{f}_4(X) \to \dot{f}_3(X)$, which is the time optimal solution (no coast zones). The docking logic will have separate values of K for range control (X) and lateral control (Y, Z) and those will be selected from the allowable docking time constraints.



For a change in ψ_R

$$\dot{Y}_{LCS} = -(R + \ell)\,\dot{\psi}_R$$



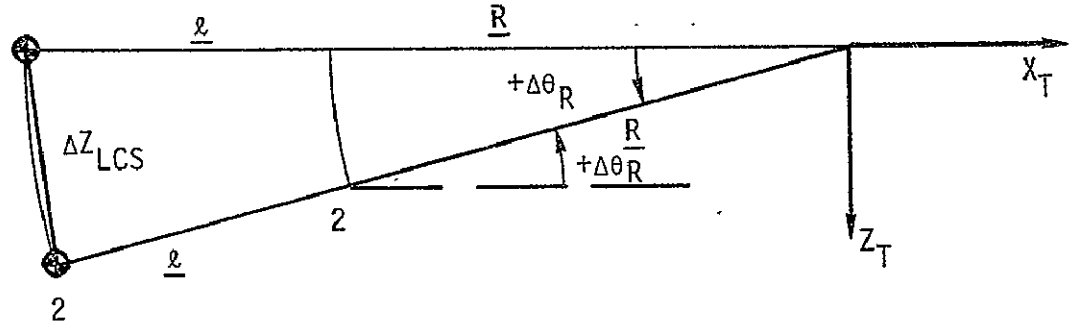
For a change in β,

$$\dot{Y}_{LCS} = -R\dot{\beta}$$

$$\text{Total } \dot{Y}_{LCS} = -R(\dot{\psi} + \dot{\beta}) - \ell\dot{\psi}$$

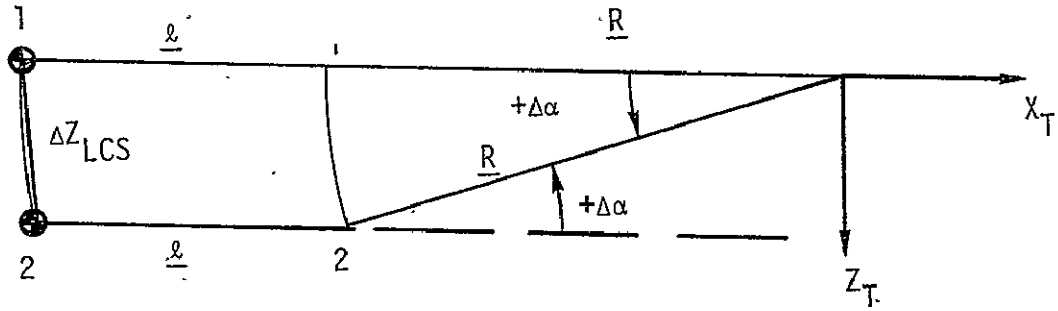
Figure 5a

$\dot{Y}_{LCS}$ Calculation



For change in θ_R,

$$\dot{Z}_{LCS} = (R + \ell)\,\dot{\theta}_R$$



For change in α,

$$\dot{Z}_{LCS} = R\dot{\alpha}$$

$$\text{Total } \dot{Z}_{LCS} = R(\dot{\theta}_R + \dot{\alpha}) + \ell\dot{\theta}_1$$

Fig

$\dot{Z}_{LCS}$ Calculation

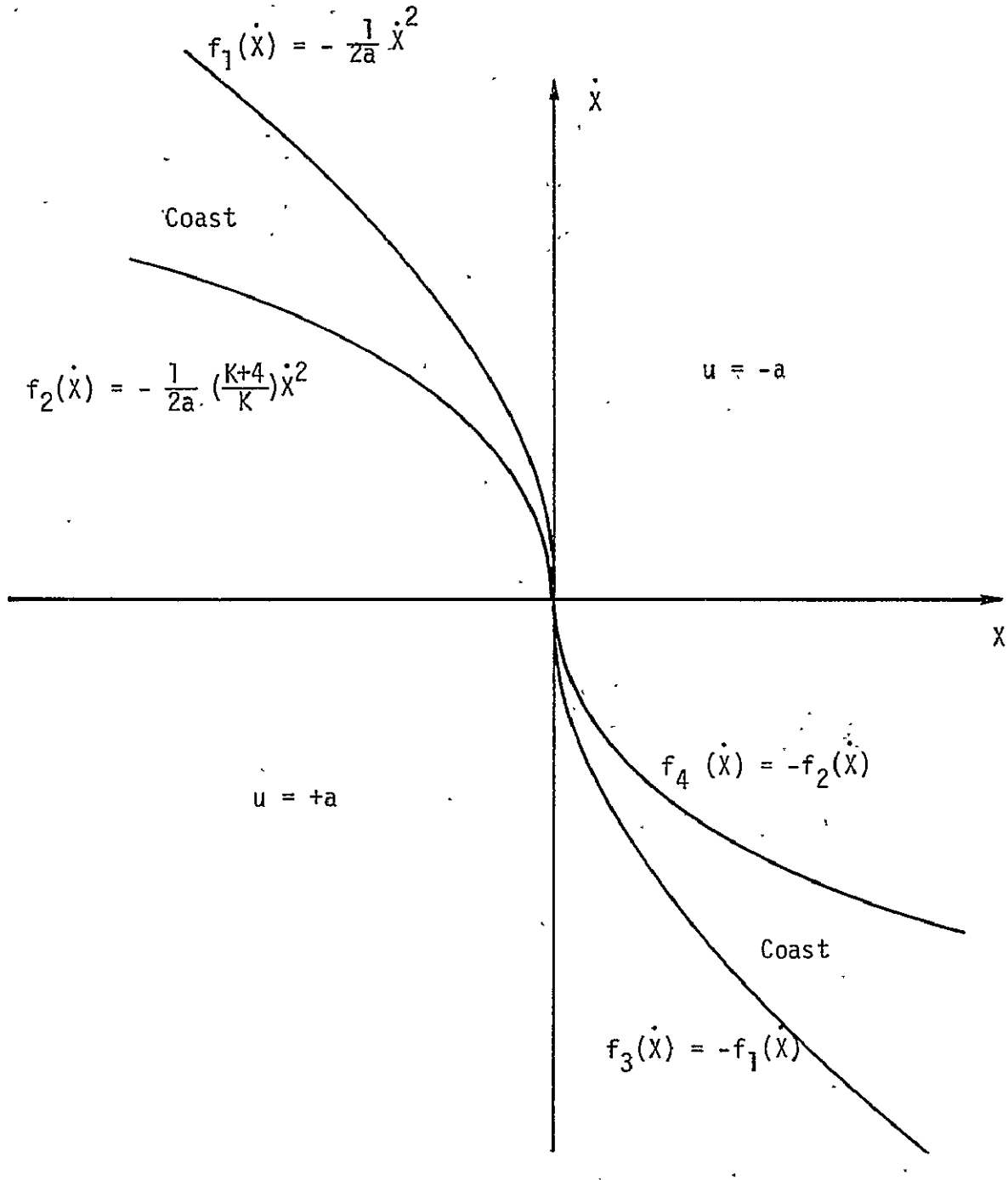


Figure 6

Time-Fuel Optimal Control Logic

5.2 Phase I Range Control

The maximum desired docking time is 5 minutes or 300 seconds and the control acceleration is .2 ft/sec^2 (using two of the available four thrusters for finer control). Assuming a worst case initial separation and closing rate of 1500 feet and zero, respectively, the slowest possible path is shown in Figure 7 (A to B to C).

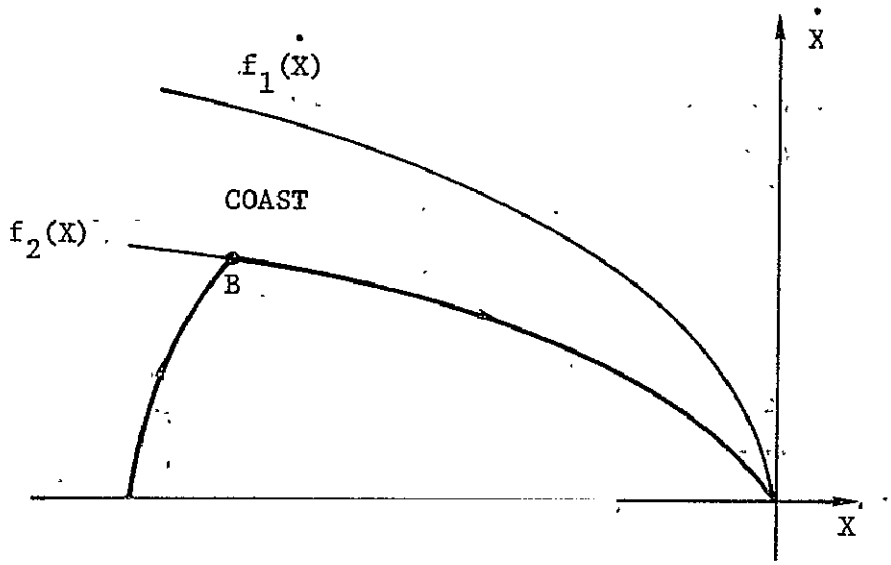


Figure 7 - Maximum Closure Time Trajectory

A to B is the control trajectory and B to C is the slowest trajectory in the coast zone. Hence, the total closure time is

$$T = t_{AB} + t_{BC}$$

The equation for total position change is,

$$X_O = \frac{1}{2}a\, t^2_{AB} + \frac{1}{2}a \left(\frac{K}{K+4}\right) t^2_{BC}$$

Also, as the rate changes from A to B and B to C must be equal,

$$a\, t_{AB} = a \left(\frac{K}{K+4}\right) t_{BC}$$

From these three equations, one can solve for K and $K/(K + 4)$

$$K = \frac{8X_0}{aT^2 - 4X_0} \qquad \frac{K}{K+4} = \frac{2X_0}{aT^2 - 2X_0}$$

For the given values of a, T and X_0, $K = 1$ and $K/(K + 4) = 0.02$. (1)

Finally, to permit coasting between some position deadband, modifications must be made to the optimal logic shown in Figure 6. Assuming a ±5 foot deadband the complete modified logic is shown in Figure 8.

Figure 9 defines the phase plane switching regions. In Region I, the desired control action is to drive the rate to 0.25 ft/sec (line segment AB). The thruster firing time is determined from $\Delta\dot{X} = at$ or

$$t_{\text{Region I}} = \frac{\dot{X} - .25}{.2} = 5\dot{X} - 1.25 \qquad (2)$$

This firing time, of course, should be no longer than the control sample period to make use of feedback. Because of inaccuracies in modeling it may be necessary to include a hysteresis line bordering Region I to eliminate chattering (see Figure 9). This will be determined at a future date.

In Region II, the desired control action brings the state into the coast zone with an opposite rate sign (example trajectory CD shown in Figure 9). The desired rate change can be found by first writing the equation for the trajectory CD and then simultaneously solving this equation with $f_4(X)$. The former equation is

$$X - X' = -\frac{1}{2a}\dot{X}^2$$

where $$X' = X_0 + \frac{1}{2}\frac{\dot{X}_0^2}{a}$$

and the latter $$X = \frac{1}{2a}\left(\frac{K+4}{K}\right)\dot{X}^2$$

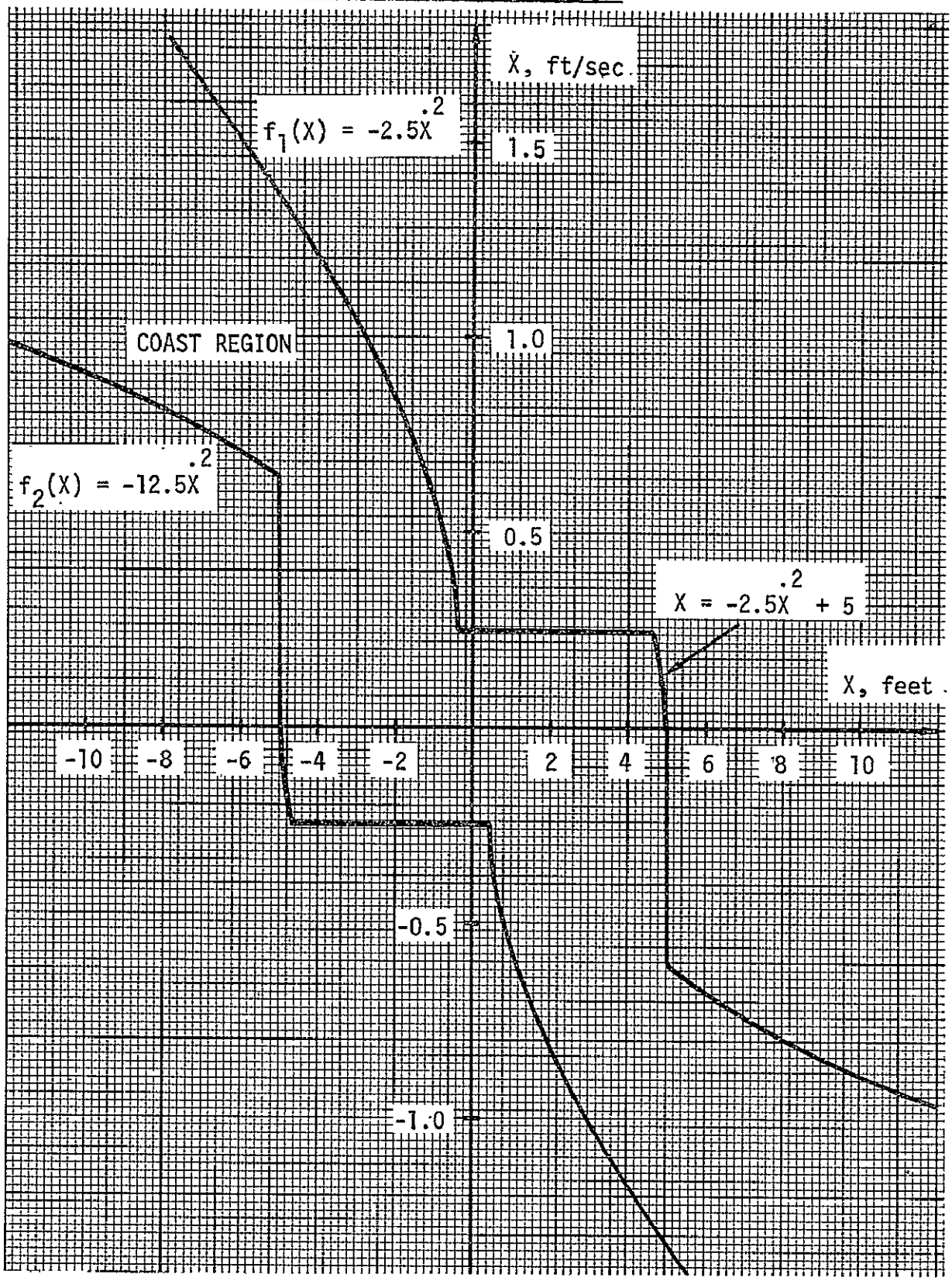


Figure 8

Phase 1 Range Control

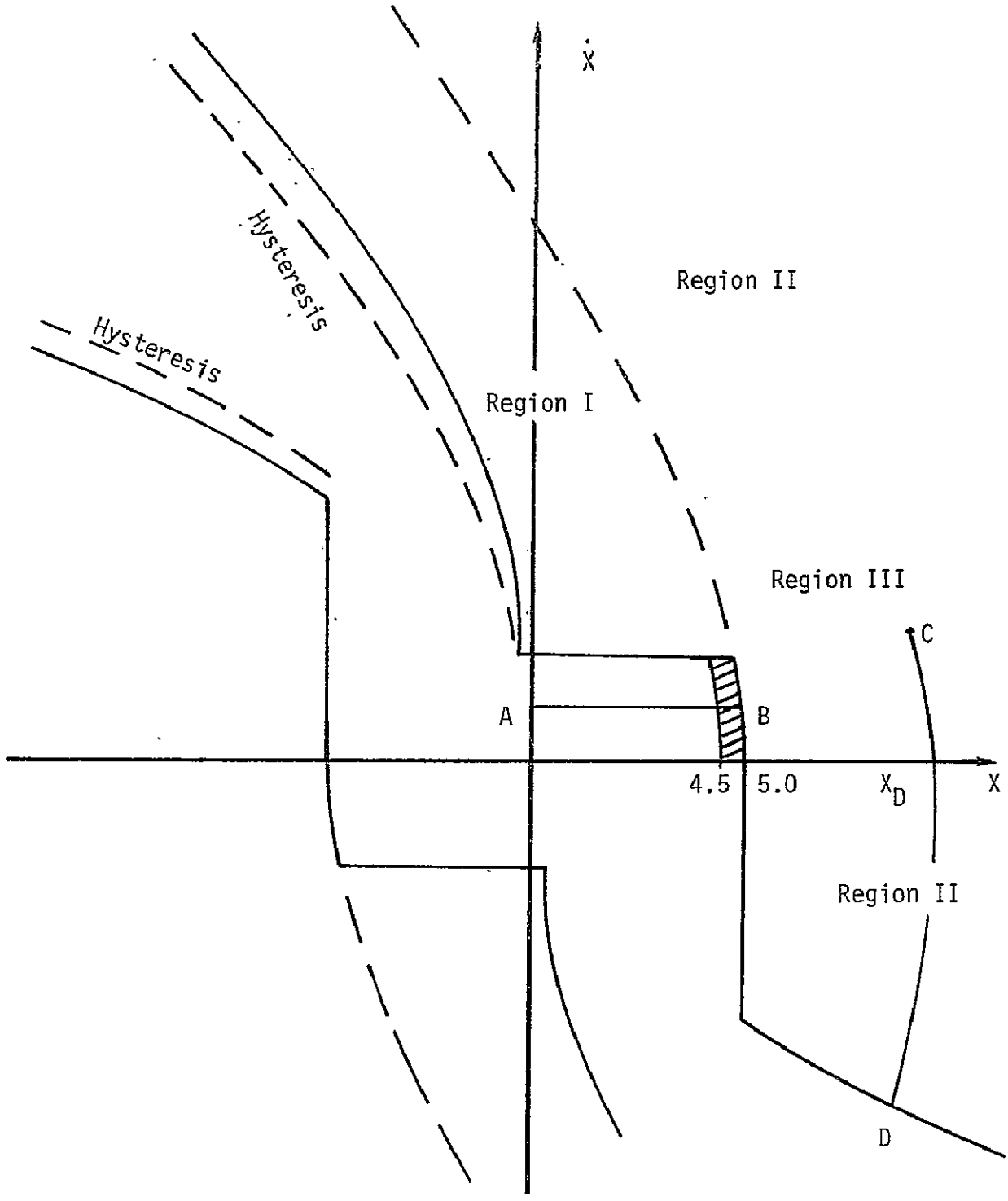


Figure 9

Phase 1 Range Control Regions

The intersection is given by

$$X_{INT} = \left(\frac{K+4}{K+2}\right)\frac{X'}{2}$$

$$\dot{X}_{INT} = -\sqrt{\frac{aKX'}{K+2}}$$

Hence, the firing time is given by

$$t_{\text{Region II}} = \frac{\dot{X}_0 + \sqrt{\frac{aKX'}{K+2}}}{a} = \frac{\dot{X}_0}{a} + \sqrt{\frac{K}{K+2}\cdot\frac{X'}{a}} \qquad (3)$$

Finally, Region III is designed to provide smooth limit cycle operation, the control action is to drive the rate to zero. Hence,

$$t_{\text{Region III}} = \frac{\dot{X}_0}{a} = 5\dot{X}_0 \qquad (4)$$

5.3 Phase 1 Lateral Control

Similarly, K for lateral control, can also be found from the constraints, the acceleration is .2 ft/sec^2, maximum docking time equals 300 seconds and maximum initial position and velocity errors of 150 feet and 3 ft/sec, respectively. Figure 10 shows the slowest trajectory.

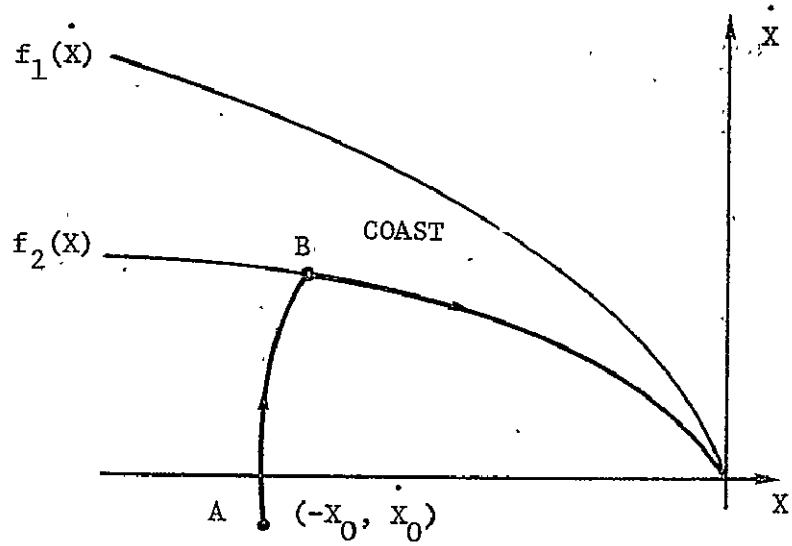


Figure 10 - Maximum Lateral Closure Time

9.10.2 Automatic Docking Control Law (con't)

Again, three equations can be found to solve for K,

time, $T = t_{AB} + t_{BC}$

position, $X_0 = -|\dot{X}_0|T + \frac{1}{2} a t^2_{AB} + \frac{1}{2} a \left(\frac{K}{K+4}\right) t^2_{BC}$

rate, $a\, t_{AB} = a\left(\frac{K}{K+4}\right) t_{BC} + |\dot{X}_0|$

The equations for K and $\left(\frac{K}{K+4}\right)$ are found to be

$$K = \frac{8a\,(X_0 + |\dot{X}_0|T) - 4|\dot{X}_0|^2}{|\dot{X}_0|^2 + (aT)^2 - 4a\,(X_0 + |\dot{X}_0|T)}$$

$$\frac{K}{K+4} = \frac{2a\,(X_0 + |\dot{X}_0|T) - |\dot{X}_0|^2}{(aT)^2 - 2a\,(X_0 + |\dot{X}_0|T)}$$

K and $\left(\frac{K}{K+4}\right)$ are calculated to be 0.594 and 0.129, respectively. Finally, Figure 11 depicts the lateral Y, Z control logic which also has a ±5 foot deadband modification. The control regions and thruster firing times are of the same format as that shown for range control.

5.4 Phase 2 Control

Phase 2 control begins at the stationkeeping state and ends at contact. For a minimum dispersion docking the following parameters and their derivatives need to be controlled:

a) Range

b) Lateral probe position errors

c) Lateral c.g. position errors

d) Relative roll

e) Relative pitch and yaw

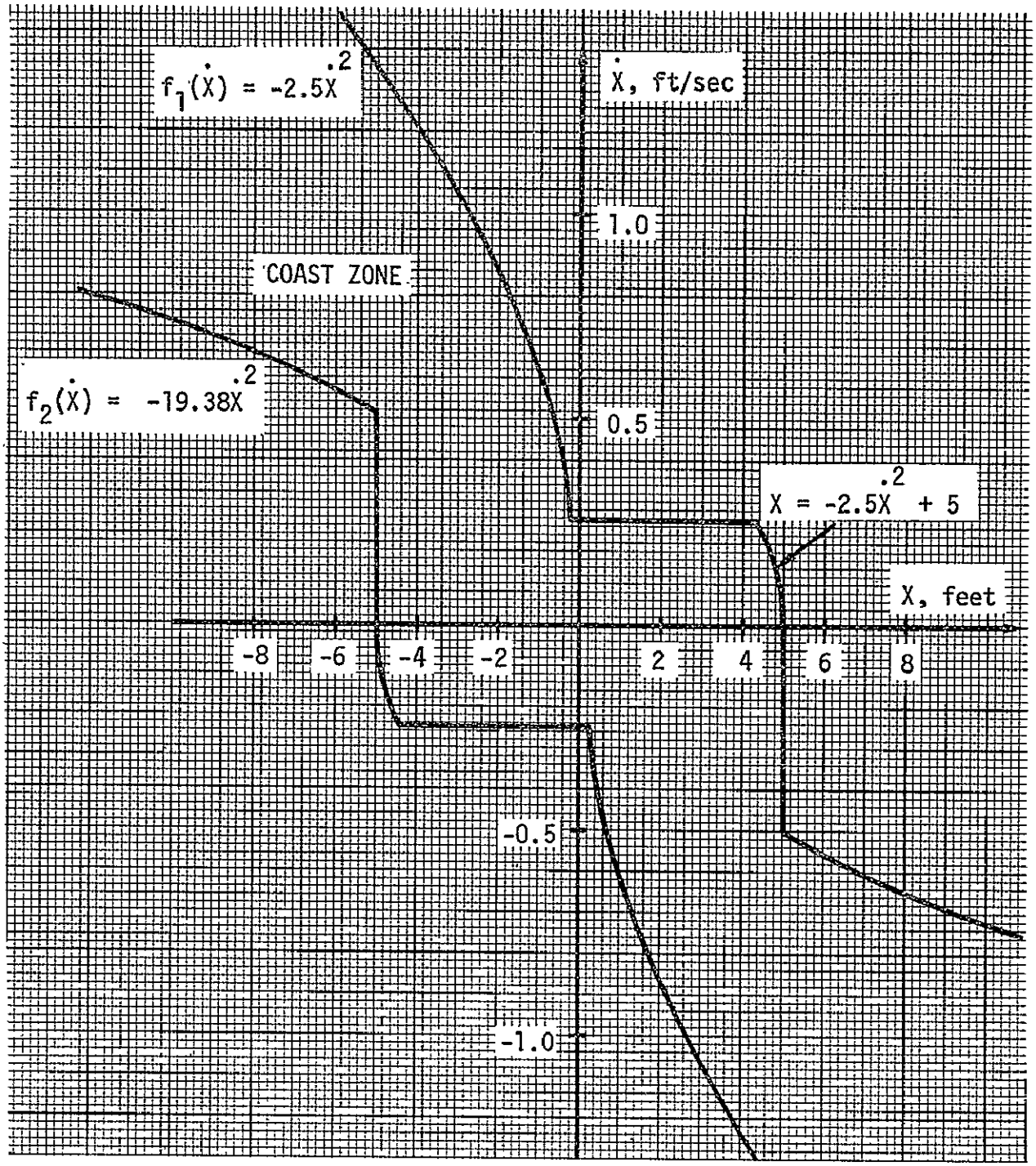


Figure 11

Phase 1 Lateral (Y,Z) Control

5.5 Phase 2 Range Control

In addition to the obvious constraints of maximum time to docking and impact velocity, there may be other constraints; for example, jet plume impingement restrictions. However, until these later constraints are defined, they will be neglected.

The coordinate system used is the TCS. The position error is given by

$$X = R \cos(\theta_R + \alpha) \cos(\Psi_R + \beta) + \ell\cos\theta_R\cos\Psi_R - \ell$$

This quantity will be filtered to provide $\dot{X}$.

The control law will basically be the same as the previously discussed time-fuel optimal logic with the addition of a rate limiting zone for coasting during the final "d" feet of the docking maneuver (see Figure 12). $f_1(X)$ is the curve dictated by two jet braking. K for $f_2(X)$ can be determined by choosing a maximum time for reaching the rate limiting logic for a worst case set of initial conditions. Selecting a maximum time of 3 minutes and a worst case I.C. of $\dot{X}_0 = .25$ ft/sec and $X_0 = 150$ feet, than $K = 0.2734$ and $\left(\frac{K}{K+4}\right) = 0.06404$. For this value of K, it can be shown that the maximum closing rate is less than 2 ft/sec.

The upper boundary in the rate limiting zone is set by the maximum impact velocity constraint which is assumed to be 0.1 ft/sec. The lower boundary is a function of maximum allowable time for coast and the distance for coast, d. Assuming a 100 seconds and 5 feet, respectively, the lower limit is 0.05 ft/sec.

There are three control regions. The first is the one lying to the right of the parabolic coast zone and above the rate limiting coast zone. Here the control should aim for a rate of 0.075 ft/sec (mid-way in rate limiting zone).

$$t_{\text{Region I}} = \frac{\dot{X}_0 - .075}{a} \qquad (5)$$

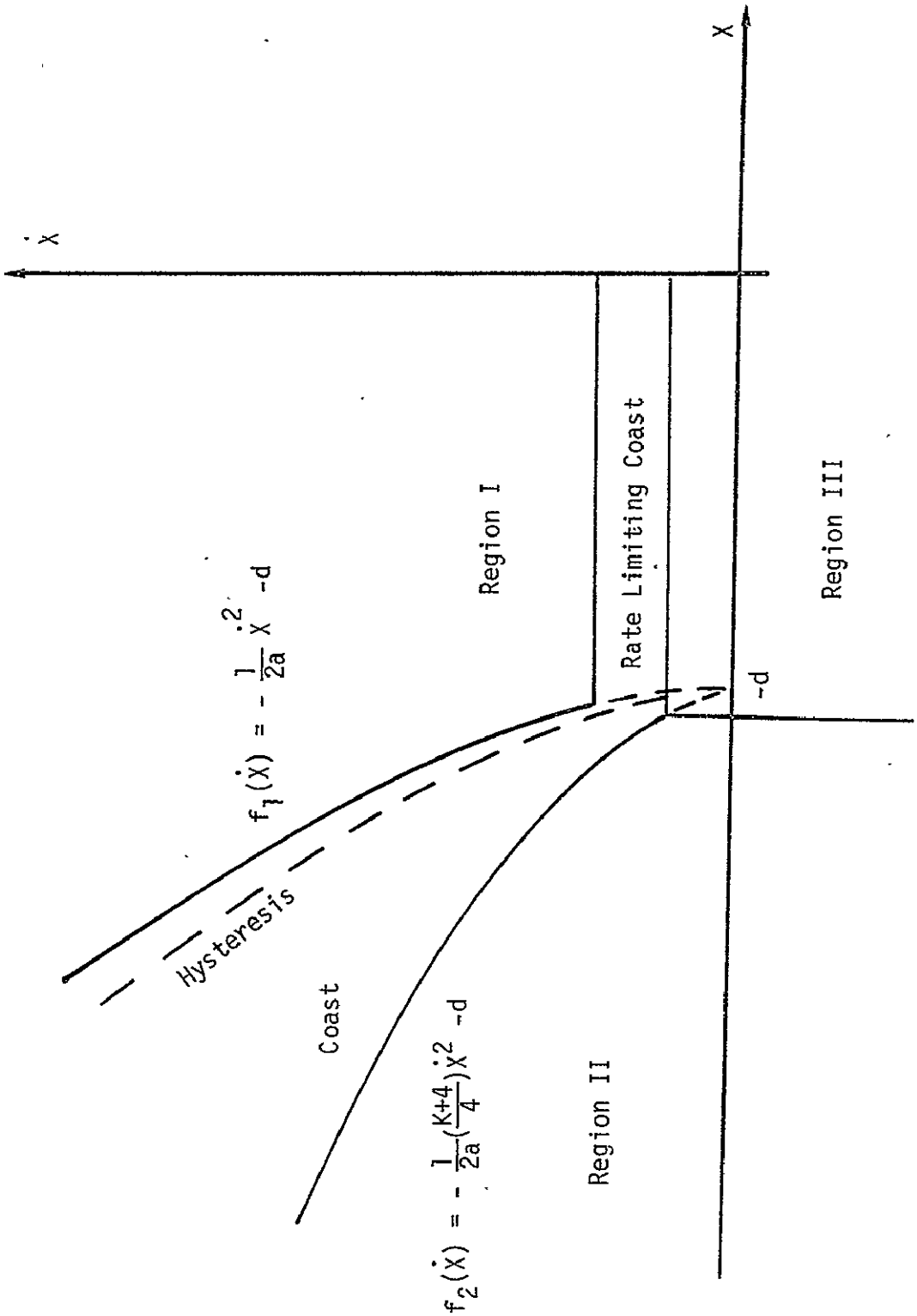


Figure 12
Phase 2 Range Control Phase Plane

9.10.2 Automatic Docking Control Law (con't)

Region II lies to the left of the parabolic coast zone and the line $X = -(d + .1)$. Control requirement for this region is to bring the state up to the lower parabolic switch line. Firing has been derived previously in a similar calculation.

$$t_{\text{Region II}} = \left\| \frac{X_0}{a} + \sqrt{\left(\frac{K}{K+2}\right) \left| \left| \frac{X'}{a} \right| \right|} \right\| \tag{6}$$

where $X' = X_0 + \frac{1}{2} \frac{X_0^2}{a}$

Finally, Region III lies to the right of the line $X = -(d + .1)$ and below the rate limiting coast zone. The firing time is

$$t_{\text{Region III}} = \frac{0.075 - \dot{X}}{a} \tag{7}$$

5.6 Control of Lateral Probe and C. G. Position Errors and Relative Pitch and Yaw Angles

Lateral probe position error should be controlled directly because the allowable lateral probe displacement at impact is likely to be quite small (one foot or less). Indirect control, by simply nulling c.g. position and relative pitch and yaw attitudes, can cause significant lateral dispersions (see reference). However, as lateral probe position error is a function of lateral c.g. position errors and the relative pitch and yaw angles, the controls for all three must be coordinated.

Considering the X-Z plane-first there are three pairs of variables to be controlled $(Z_{cg}, \dot{Z}_{cg})$ $(Z_p, \dot{Z}_p)$ and $(\theta_R, \dot{\theta}_R)$. During this phase of control the two translational parameters will be calculated as follows in the TCS (see Figure 13).

$$Z_{cg} = R \sin (\theta_R + \alpha) + \ell \sin \theta_R$$

$$Z_p = R \sin (\theta_R + \alpha)$$

$$Z_{cg} = R \sin (\theta_R + \alpha) + \ell \sin \theta_R$$

$$Z_p = R \sin (\alpha + \theta_R)$$

Note: Negative θ_R shown (wrt TCS)

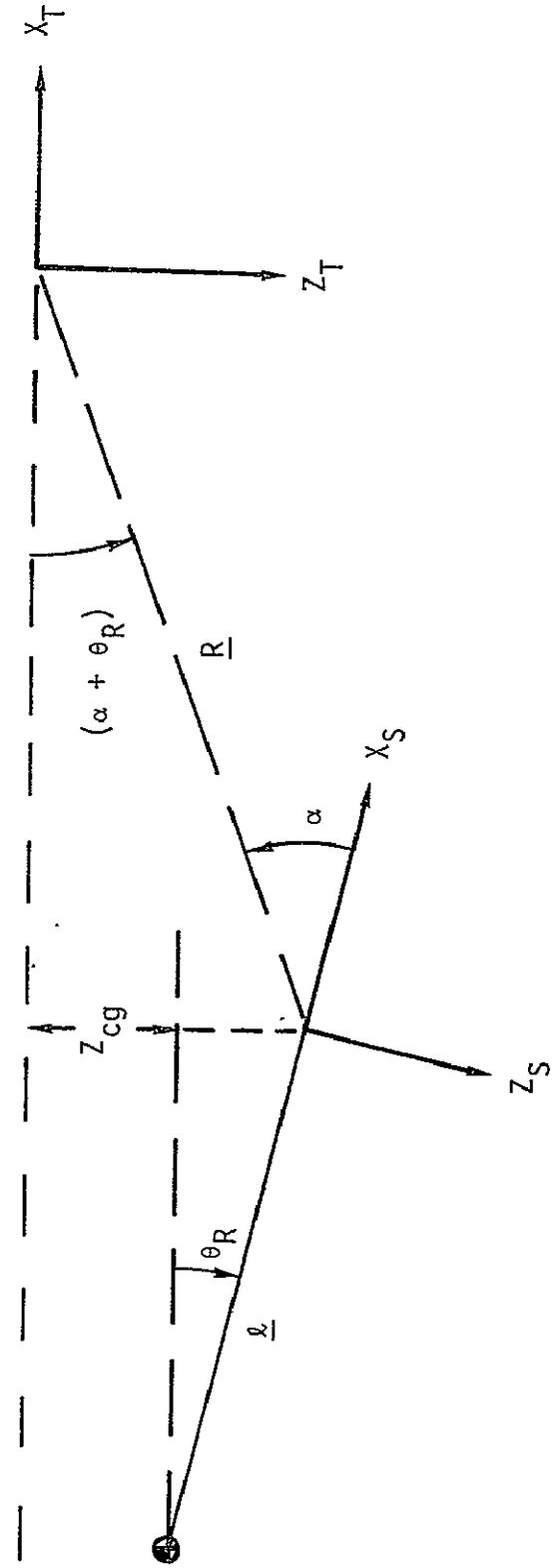


Figure 13
Illustration of Z_{cg} and Z_p

The respective rates will be obtained by a filtering routine to be determined.

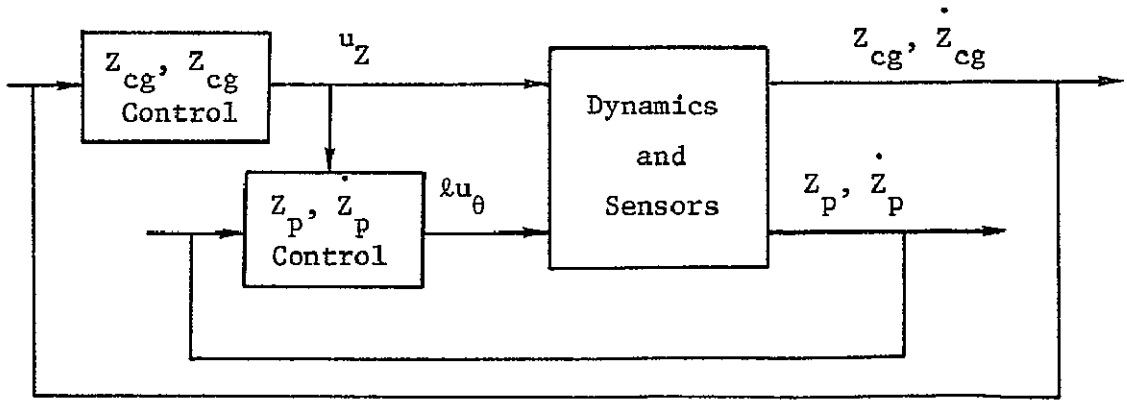
The control inputs are, of course, the ±Z thruster forces, u_Z, and the ±θ pitch torques, u_θ, which are applied in the following manner:

$$\ddot{Z}_p = u_Z - \ell u_\theta$$

$$\ddot{Z}_{cg} = u_Z$$

$$\ddot{\theta}_R = u_\theta$$

Intuitively, it can be seen that explicit control of any two of the variables (Z_p, Z_{cg}, θ_R), implicitly controls the third. For example, a control which forces two of the variables into prescribed limit cycles, indirectly bounds the remaining variable into some limit cycle. As Z_p has already been chosen as one of the variables to be controlled directly, it only remains to select either Z_{cg} or θ_R for the other directly controlled variable. Either is acceptable; however, Z_{cg} is chosen because u_θ has five times more control authority than u_Z and by this choice ℓu_θ can be used exclusively for Z_p control. Summarizing, at this point we have u_Z for exclusive control of Z_{cg} and ℓu_θ for exclusive control of Z_p where u_Z is a known disturbance of Z_p. In block diagram form, this is represented as:



9.10.2 Automatic Docking Control Law (con't)

As an aid in defining the control, it is helpful to visualize the desired control state. Realizing that there is a minimum control impulse on Z_{cg} and θ_R which necessitates deadbands, Figure 14 indicates the ideal control state. Effectively, we have Z_p located on the approach path with Z_{cg} moving up and down in a deadband. Requirements for this condition are

$$(1) \quad \dot{Z}_{cg} - \ell\dot{\theta}_R = 0$$

$$(2) \quad \dot{Z}_p = 0$$

A method of approximating this control state is to (1) drive $(Z_{cg}, \dot{Z}_{cg})$ into a deadbanded limit cycle (consistent with allowable θ_R range), (2) drive $(Z_p, \dot{Z}_p)$ into a very small deadband limit cycle, and (3) use differential jet firings to approach $Z_p = 0$ (i.e., take advantage of the small control impulse available from $u_Z - \ell u_\theta$).

A final consideration is that Z_{cg} and Z_p should be within their deadbands before the range control has reached the rate limiting zone. As the maximum closing rate is 2 ft/sec, the minimum time for this is 150/2 or 75 seconds.

Switching logic for $(Z_{cg}, \dot{Z}_{cg})$

The switching logic will have the same form as that used for Phase I except there will be a different deadband and value of K for the $f_2(X)$ function. The deadband is dependent on the per axis allowable misalignment angle, γ.

Hence, we have

$$Z_{cg} \text{ DEADBAND} = \ell \sin\gamma$$

Assuming a γ of 2° the Z_{cg} deadband is 2.5 feet. To determine K, we insert T = 75 seconds and worst case initial conditions into the previously derived formula

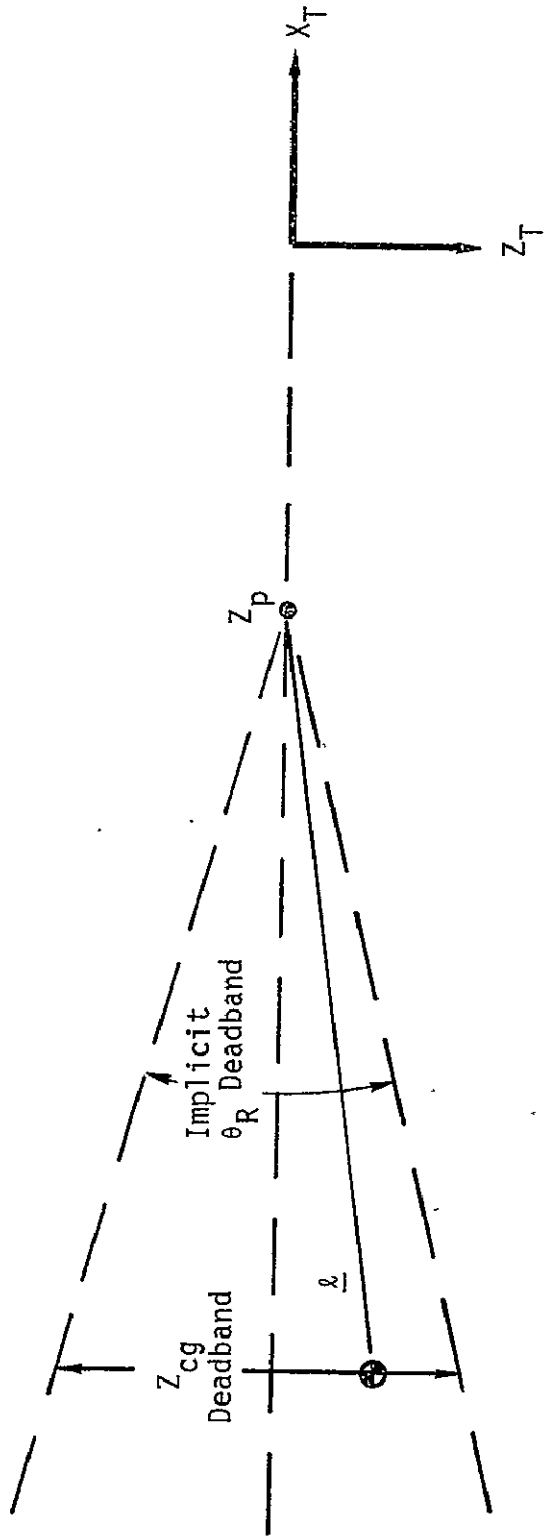


Figure 14
Desired Control State

9.10.2 Automatic Docking Control Law (con't)

$$K = \frac{8a\,(X_0 + |\dot{X}_0|T) - 4\,|\dot{X}_0|^2}{|\dot{X}_0|^2 + a^2T^2 - 4a(X_0 + |\dot{X}_0|T)}$$

Assuming X_0 = 10 ft and $\dot{X}_0$ = -.25 ft/sec, K = .2265. However, as the value of K used for Phase 1 control, 0.594, is more conservative (i.e., longer hence quicker) it will be used for simplicity of logic coding. A final modification from the Phase 1 logic is necessitated by the differential jet firing technique which will be used to null Z_p. It requires the minimum delta V impulses available from the Z-translation and pitch jets to be the same. As u_Z is approximately five times smaller than u_θ, the minimum impulse from u_Z must be increased proportionately. The control regions are the same as that previously used for lateral control.

5.7 Switching logic for $(Z_p, \dot{Z}_p)$

The control for $(Z_p, \dot{Z}_p)$ will also be a modified form of the time - fuel optimal switching logic. Figure 15 illustrates this logic. The linear acceleration from the pitch thrusters is five times greater than that from the translational thrusters, hence, for two thruster acceleration

$$f_1(\dot{Z}_p) = \frac{1}{2a}(\dot{Z}_p)^2 = \frac{1}{2}\dot{Z}_p^2$$

To determine K for the $f_2(Z_p)$ function we must again revert back to the worst case initial conditions and maximum allowable time (this was chosen in the range control law to be 3 minutes). Worst case initial conditions from the stationkeeping phase are shown in Figure 16. The position error is seen to be about -11 feet whereas velocity error is about -1.2 ft/sec (due to minimum impulse rates from translational and rotational control). Using this I.C., K is found to be 0.0575 and K/(K + 4) equals 0.0142. These small values may be increased slightly because of the high sensitivity of the $f_2(Z_p)$ function to a rate error in Z_p. The desired deadband as shown

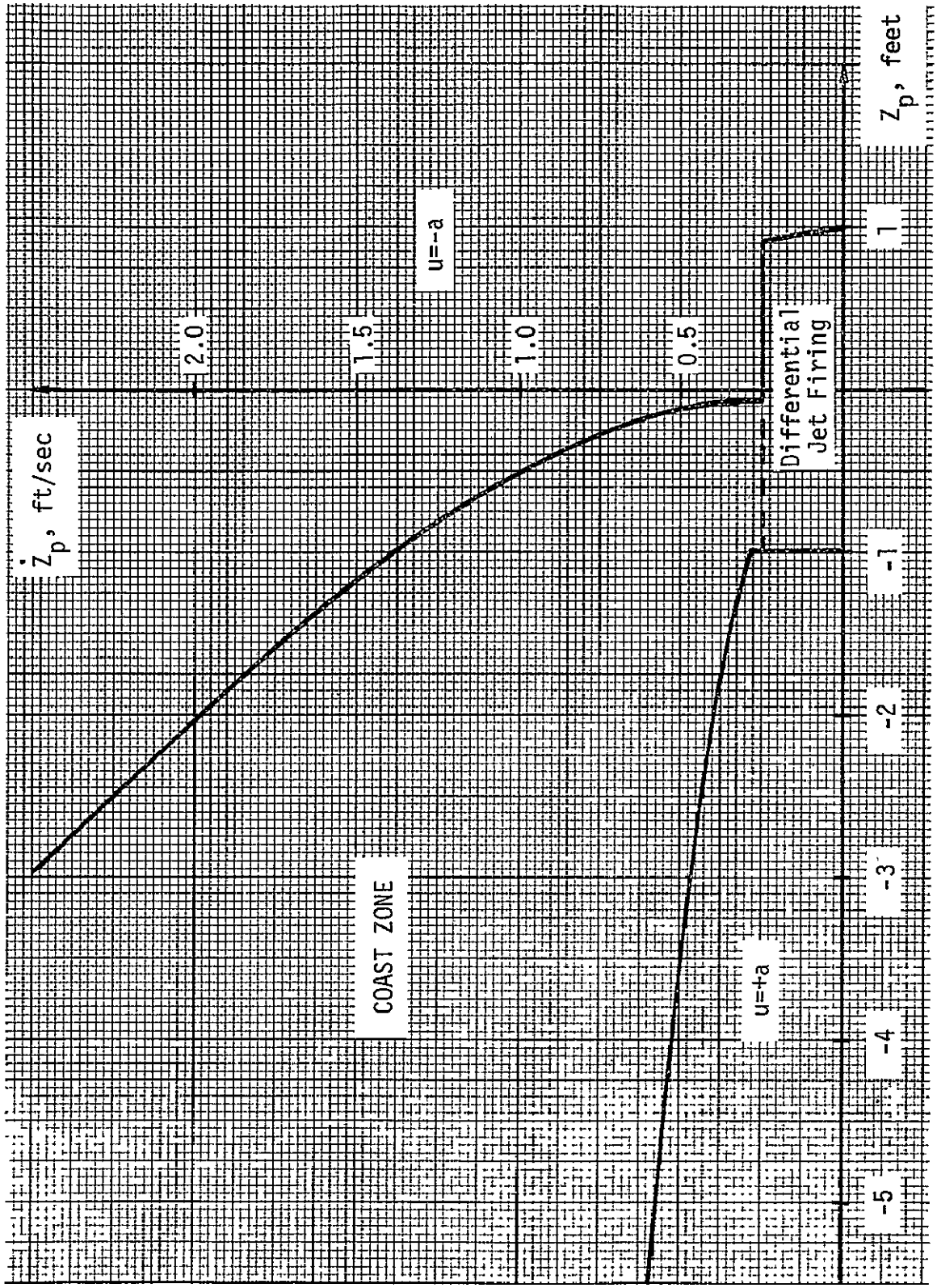


Figure 15
Z_p, $\dot{Z}_p$ Phase Plane

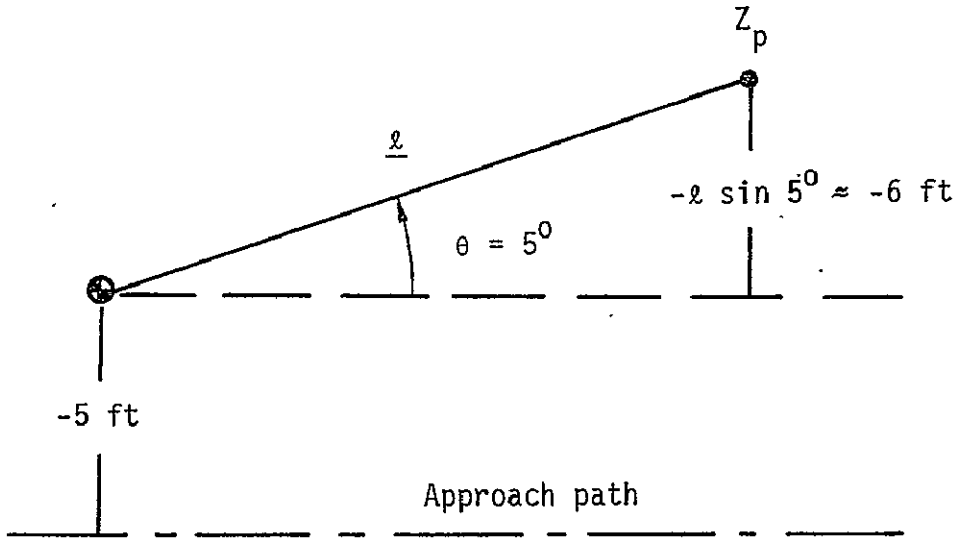


Figure 16

Worst Case Position I.C.

in Figure 15 is one foot. To account for u_Z thruster firings the phase plane is broken into several control regions, as illustrated in Figure 17. With no Z-thruster control disturbances, u_Z, the firing times for Regions I, II, and III are calculated in the same manner as the other phase planes. Namely,

$$t_{\text{Region I}} = \frac{\dot{X} - .125}{a} = \dot{X} - .125 \tag{8}$$

$$t_{\text{Region II}} = \frac{X_0}{a} + \sqrt{\left(\frac{K}{K+2}\right)\frac{X'}{a}} \tag{9}$$

$$t_{\text{Region III}} = \frac{\dot{X}_0}{a} = \dot{X}_0 \tag{10}$$

However, in Regions I, II, and III, if a non-zero command is scheduled from the $(Z_{cg}, \dot{Z}_{cg})$ phase plane, and this would cause the $(Z_p, \dot{Z}_p)$ state to diverge (because of disagreement in sign), then this command is treated as a disturbance and the $(Z_p, \dot{Z}_p)$ firing time is increased proportionately for opposing commands

$$t_{\text{Region I, II}} = t_{\text{Region I, II}} + \frac{1}{5} t_{(Z, \dot{Z})} \tag{11}$$

However, as the net acceleration during this disturbance period is reduced by 20 percent, convergence time in the phase plane may be increased. If this proves a significant factor, K will be increased. Commands in the proper direction are not compensated for as this would cause chattering during long Z firing times. In Region III a u_Z command in either direction should be compensated for

$$t_{\text{Region III}} = t_{\text{Region III}} \pm \frac{1}{5} t_{(Z,\dot{Z})} \tag{12}$$

as the desired control action is to drive the rate to zero.

Finally, in Region IV, no control action is taken unless there is a u_Z command; then the desired control action is to drive the rate to

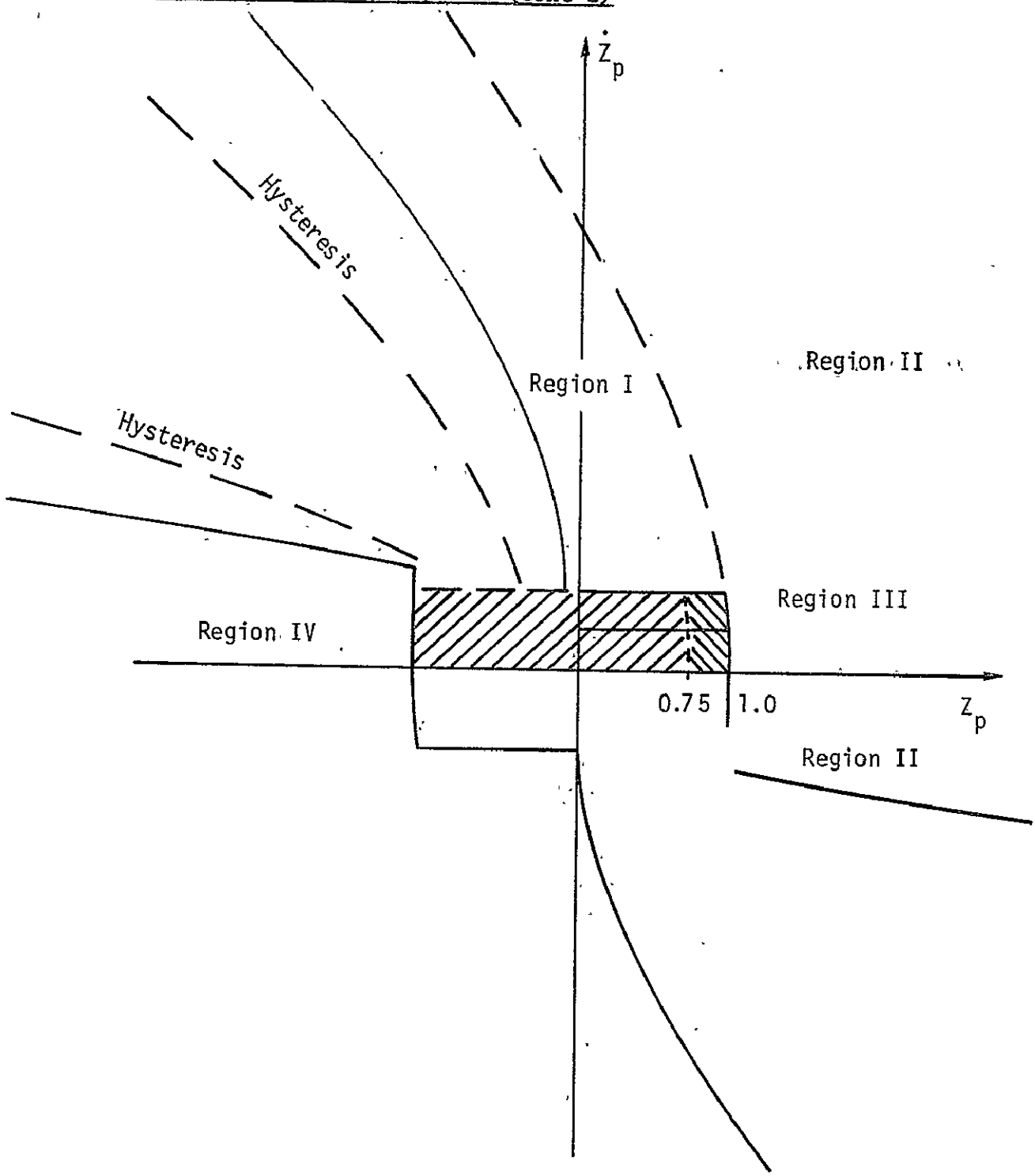


Figure 17

Z_p, $\dot{Z}_p$ Control Regions

zero through differential jet firings. The firing time equation is

$$\dot{Z}_p + u_Z\, t_{(Z,\dot{Z})} - \ell u_\theta\, t_{(Z_p,\, \dot{Z}_p)} = 0$$

Hence, the sign of the control is given by

$$\text{sign}\,(\ell u_\theta) = \text{sign}\,(\dot{Z}_p + u_Z T_{(Z,\, \dot{Z})}) \tag{13}$$

The firing time by

$$t_{\text{Region IV}} = \frac{\dot{Z}_p + u_Z\, t_{(Z,\, \dot{Z})}}{\ell u_\theta} \tag{14}$$

A dual relation exists for control in the X-Y plane (see Figure 18). The position errors are

$$Y_{cg} = -R \sin(\Psi_R + \beta) - \ell \sin\Psi_R$$

$$Y_p = -R \sin(\Psi_R + \beta)$$

Applicable control accelerations are

$$\ddot{Y}_p = u_Y + \ell u_\Psi$$

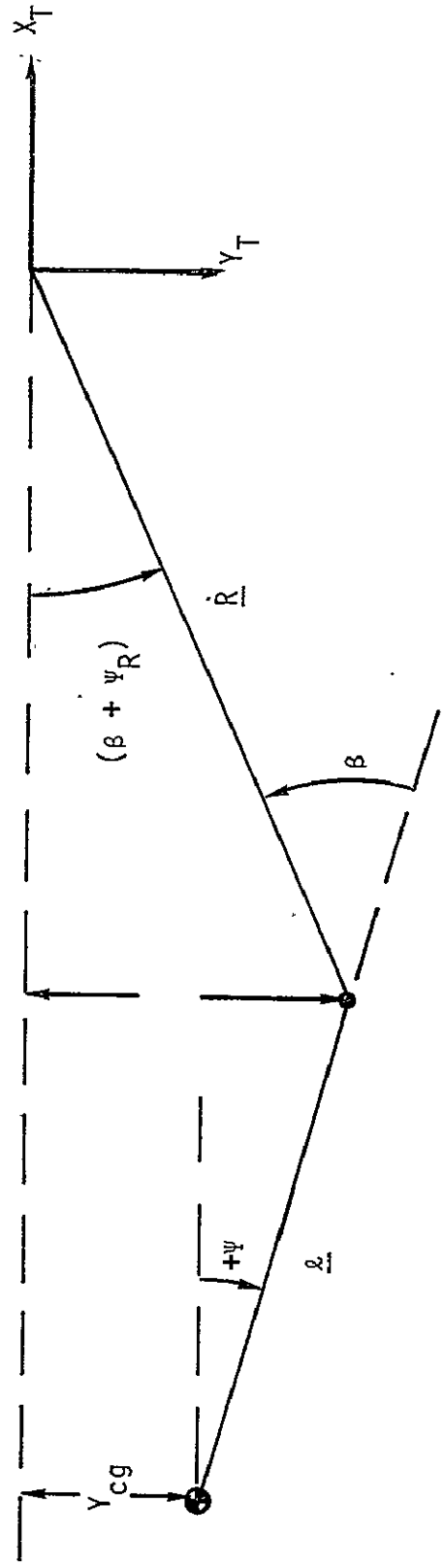
$$\ddot{Y}_{cg} = u_Y$$

$$\ddot{\Psi}_R = u_\Psi$$

The desired end condition requires

(1) $Y_{cg} + \ell \dot{\Psi}_R = 0$

(2) $Y_p,\ \dot{Y}_p = 0$



$Y_{cg} = -R \sin (\psi_R + \beta) - \ell \sin \psi_R$

$Y_p = -R \sin (\psi_R + \beta)$

Note: Negative β shown

Figure 18
Illustration of Y_{cg} and Y_p

The switching logic is identical except for Region IV where the sign of the control effort is given by

$$\text{Sign } (\ell u_\psi) = -\text{sign } (Y_p + u_Y \, t_{(Y,\, \dot{Y})}) \tag{15}$$

and the firing time is given by

$$t_{\text{Region IV}} = \frac{\dot{Y}_p + u_Y \, t_{(Y,\, \dot{Y})}}{-\ell u_\psi} \tag{16}$$

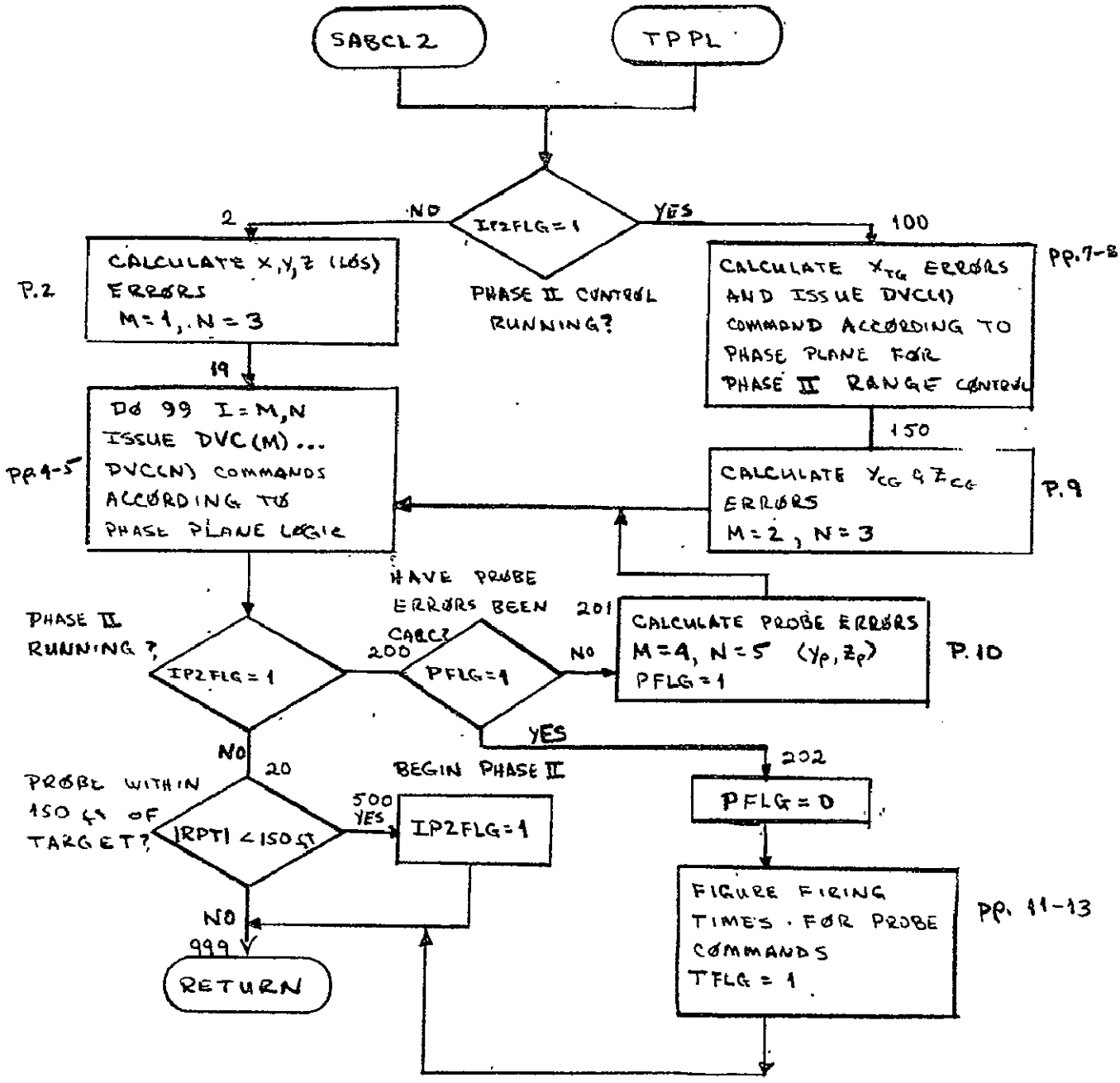
5.8 Relative Roll Control

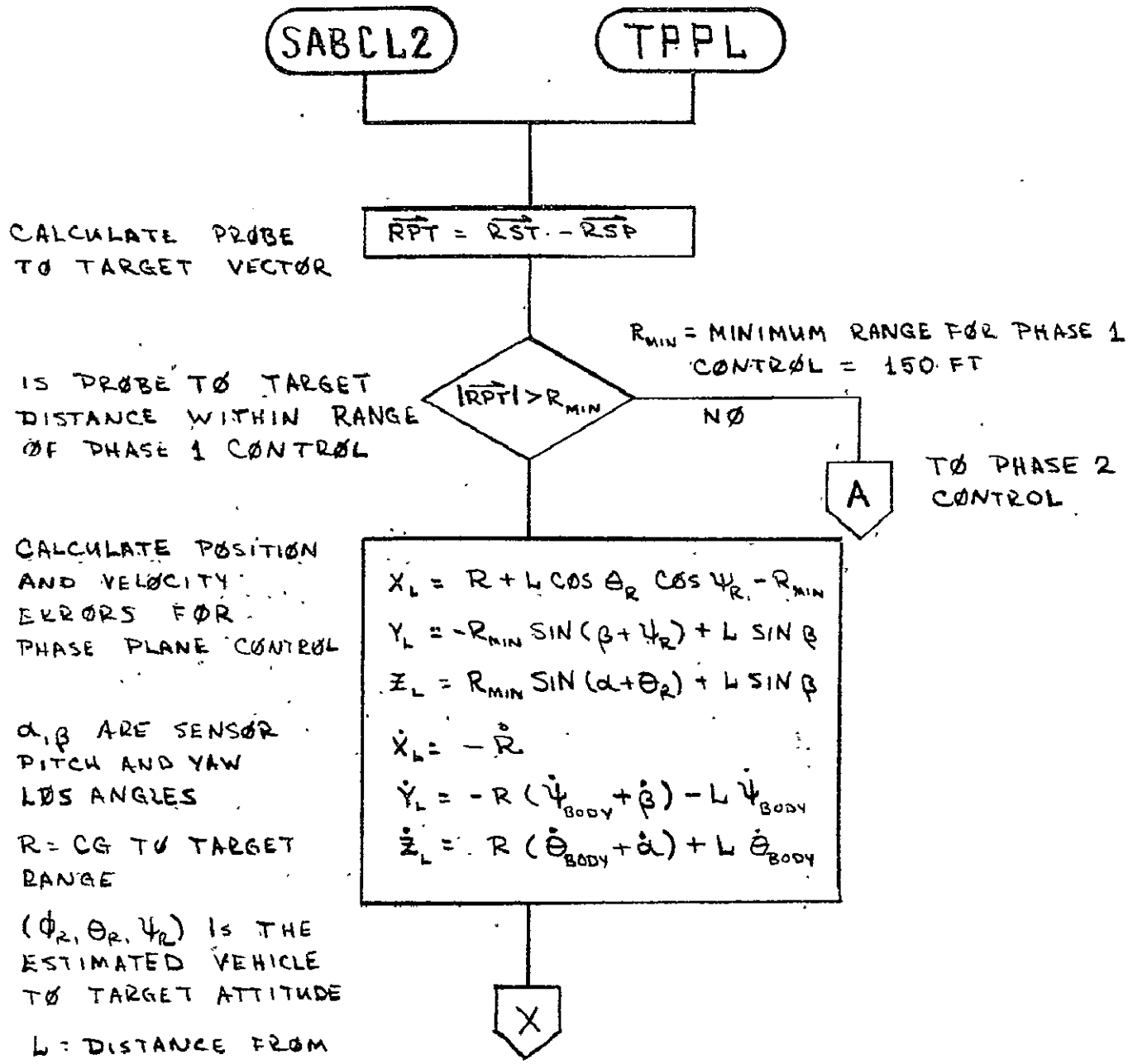
Relative roll control will be the same as that used in Phase 1 except that the deadband will be reduced to comply with docking constraints and close-in measurement accuracies. Two degrees will be assumed initially.

6. Detailed Flow Diagrams

This section contains the flow diagrams for the Automatic Docking Control Law.

THE FOLLOWING FLOW CHART SHOWS THE FUNCTIONAL FLOW OF PHASE I AND PHASE II CONTROL.





α, β ARE SENSOR PITCH AND YAW LOS ANGLES

R = CG TO TARGET RANGE

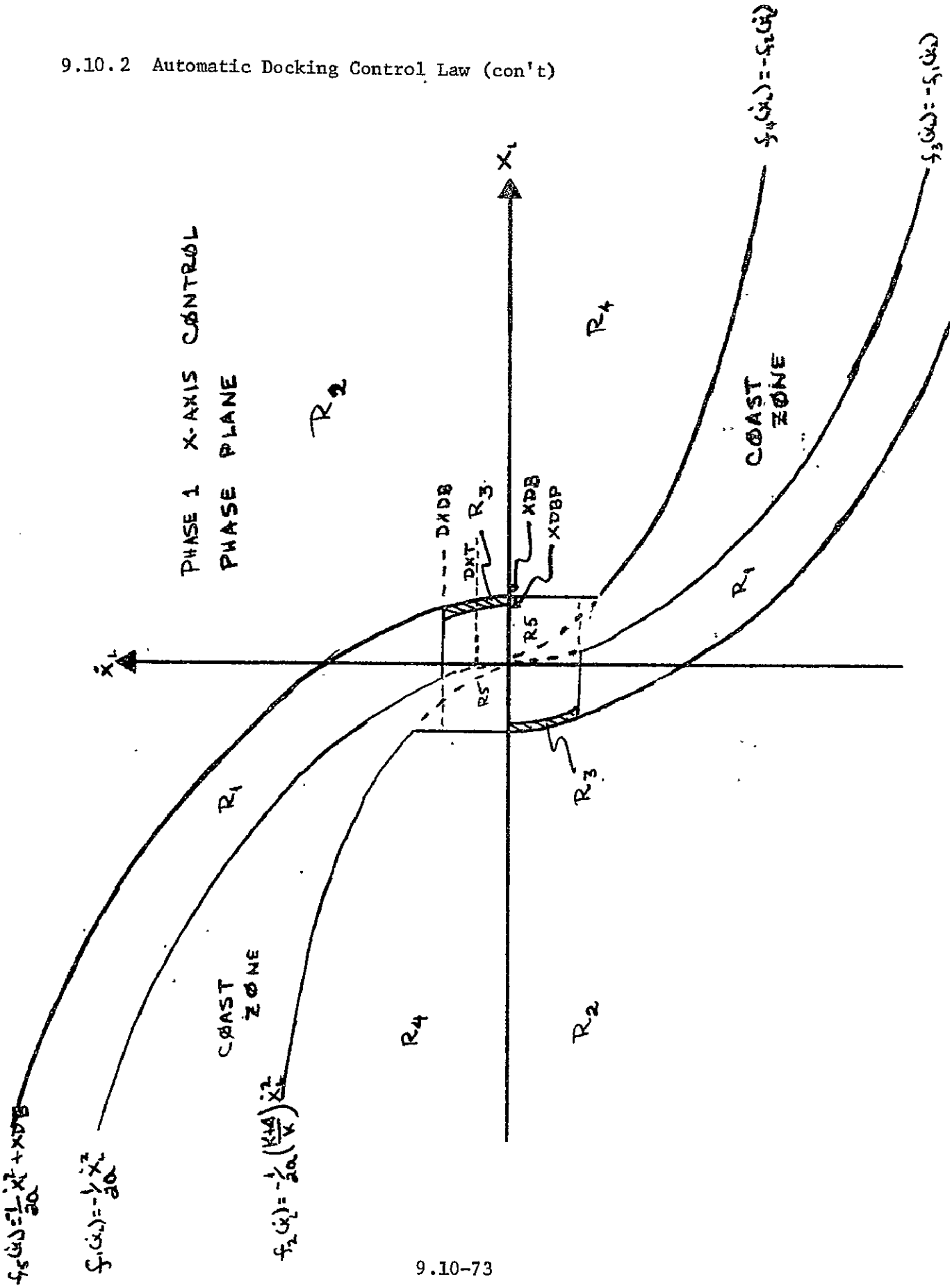
$(\phi_R, \theta_R, \psi_R)$ IS THE ESTIMATED VEHICLE TO TARGET ATTITUDE

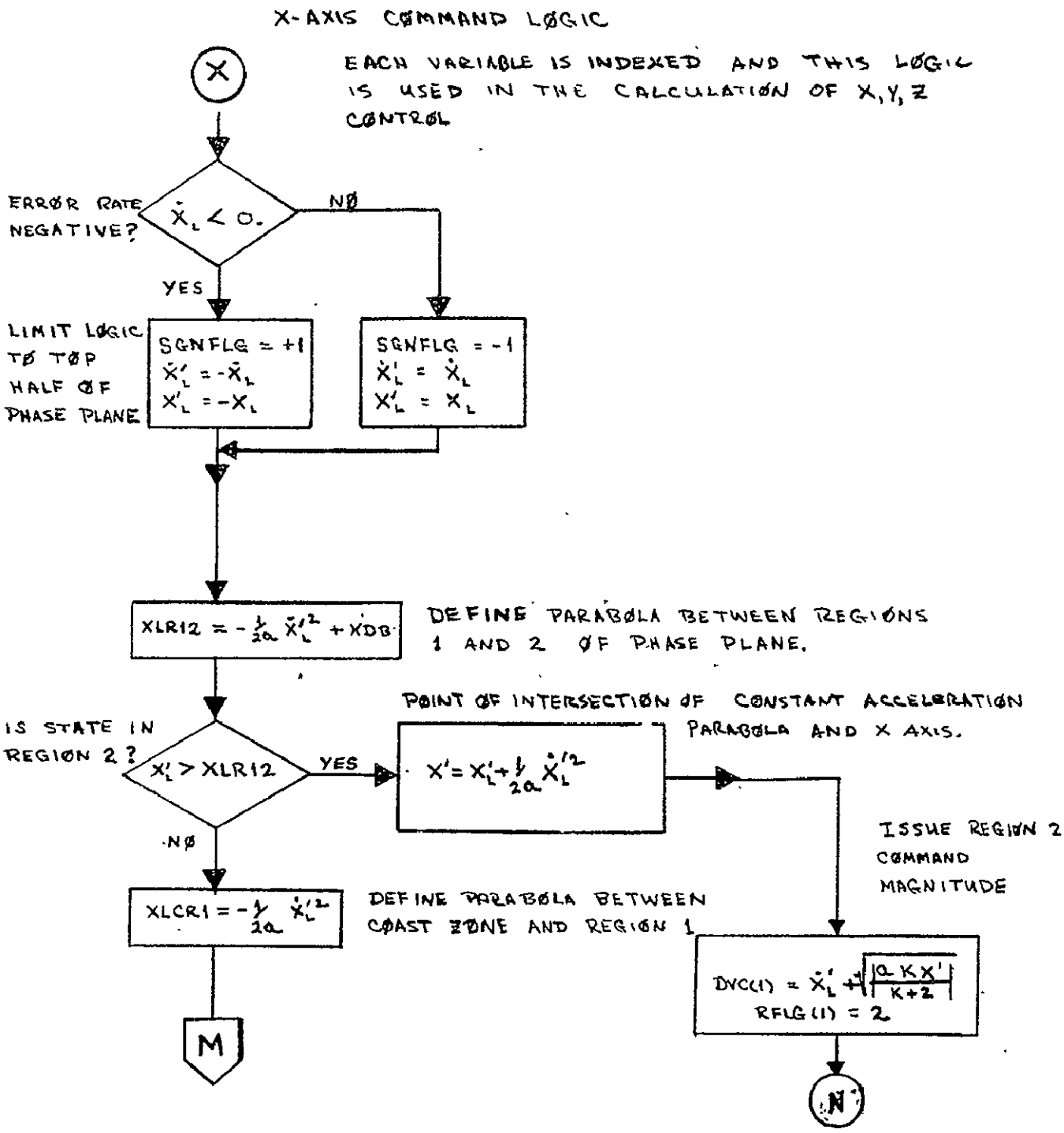
L = DISTANCE FROM CG TO SENSOR

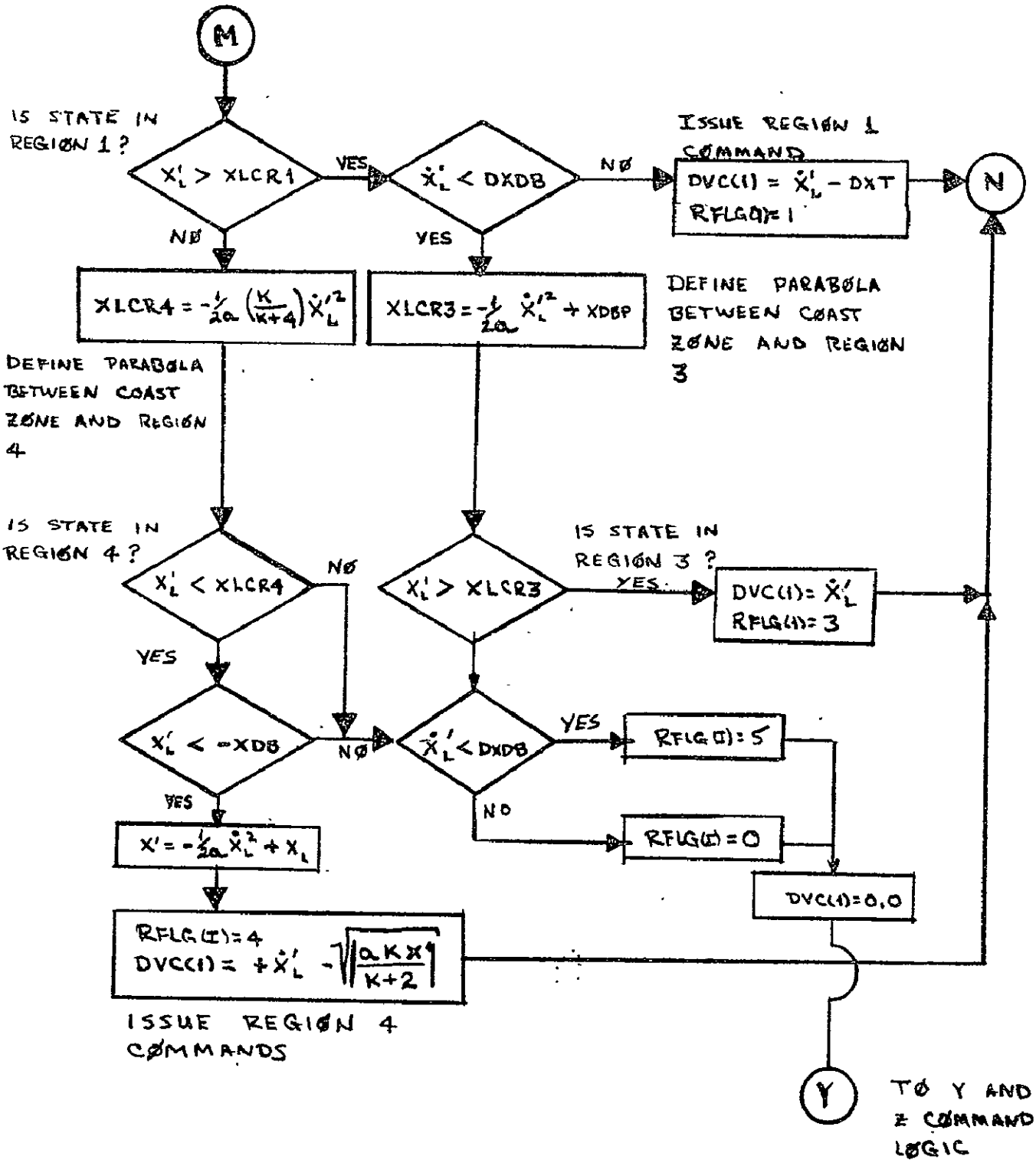
$(\dot{\phi}_{BODY}, \dot{\theta}_{BODY}, \dot{\psi}_{BODY})$ IS THE VEHICLE BODY RATE

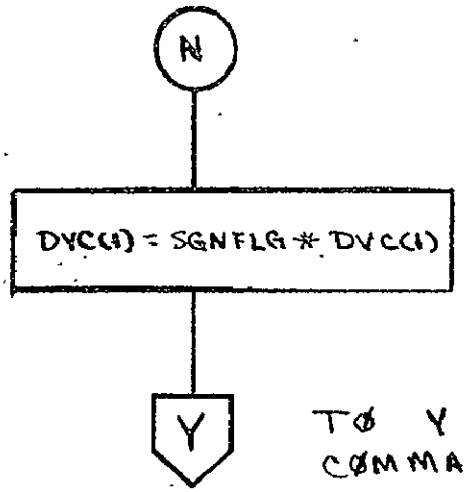
$\dot{\alpha}, \dot{\beta}$ = LOS RATES

$\dot{R}$ = CG TO TARGET CLOSING RATE


PHASE 1 X-AXIS CONTROL
PHASE PLANE
$\dot{X}_L$
X_L
R_1
R_2
R_3
R_4
R_5
R_5'
COAST ZONE
COAST ZONE
DXDB
DXT
XDB
XDBP
$f_1(\dot{x}_L) = -\frac{1}{2a}\dot{x}_L^2$
$f_2(\dot{x}_L) = -\frac{1}{2a}\left(\frac{K+4}{K}\right)\dot{x}_L^2$
$f_3(\dot{x}_L) = -f_1(\dot{x}_L)$
$f_4(\dot{x}_L) = -f_2(\dot{x}_L)$
$f_5(\dot{x}_L) = \frac{1}{2a}\dot{x}_L^2 + XDB$


X-AXIS COMMAND LOGIC
X
EACH VARIABLE IS INDEXED AND THIS LOGIC IS USED IN THE CALCULATION OF X, Y, Z CONTROL
ERROR RATE NEGATIVE?
$\dot{X}_L < 0.$
NO
YES
LIMIT LOGIC TO TOP HALF OF PHASE PLANE
SGNFLG = +1
$\dot{X}'_L = -\dot{X}_L$
$X'_L = -X_L$
SGNFLG = -1
$\dot{X}'_L = \dot{X}_L$
$X'_L = X_L$
$XLR12 = -\frac{1}{2a}\dot{X}'^2_L + XDB$
DEFINE PARABOLA BETWEEN REGIONS 1 AND 2 OF PHASE PLANE.
IS STATE IN REGION 2?
$X'_L > XLR12$
YES
NO
POINT OF INTERSECTION OF CONSTANT ACCELERATION PARABOLA AND X AXIS.
$X' = X'_L + \frac{1}{2a}\dot{X}'^2_L$
ISSUE REGION 2 COMMAND MAGNITUDE
$XLCR1 = -\frac{1}{2a}\dot{X}'^2_L$
DEFINE PARABOLA BETWEEN COAST ZONE AND REGION 1
$DVC(1) = \dot{X}'_L + \sqrt{\frac{a K X'}{K+2}}$
$RFLG(1) = 2.$
M
N


M
IS STATE IN REGION 1 ?
$X'_L > XLCR1$
YES
$\dot{X}'_L < DXDB$
NO
ISSUE REGION 1 COMMAND
$DVC(1) = \dot{X}'_L - DXT$
$RFLG(1) = 1$
N
NO
YES
$XLCR4 = -\frac{1}{2a}\left(\frac{K}{K+4}\right)\dot{X}'^2_L$
$XLCR3 = -\frac{1}{2a}\dot{X}'^2_L + XDBP$
DEFINE PARABOLA BETWEEN COAST ZONE AND REGION 3
DEFINE PARABOLA BETWEEN COAST ZONE AND REGION 4
IS STATE IN REGION 4 ?
$X'_L < XLCR4$
NO
YES
$X'_L < -XDB$
NO
YES
$X' = -\frac{1}{2a}\dot{X}'^2_L + X_L$
$RFLG(I) = 4$
$DVC(1) = +\dot{X}'_L - \sqrt{\frac{aKX'}{K+2}}$
ISSUE REGION 4 COMMANDS
IS STATE IN REGION 3 ?
$X'_L > XLCR3$
YES
$DVC(1) = \dot{X}'_L$
$RFLG(1) = 3$
$\dot{X}'_L < DXDB$
YES
$RFLG(I) = 5$
NO
$RFLG(I) = 0$
$DVC(1) = 0.0$
Y
TO Y AND Z COMMAND LOGIC



CØNSTANTS FØR X-AXIS CØMMAND LØGIC

a = 0.2 FT/SEC^2 = CØNSTANT ACCELERATIØN AVAILABLE ALØNG X-AXIS

K = 1.0 = ØPTIMIZATIØN FACTØR FØR TIME VS. FUEL

XDB = 5.0 FT. (SEE PHASE PLANE IN THE FIGURE)

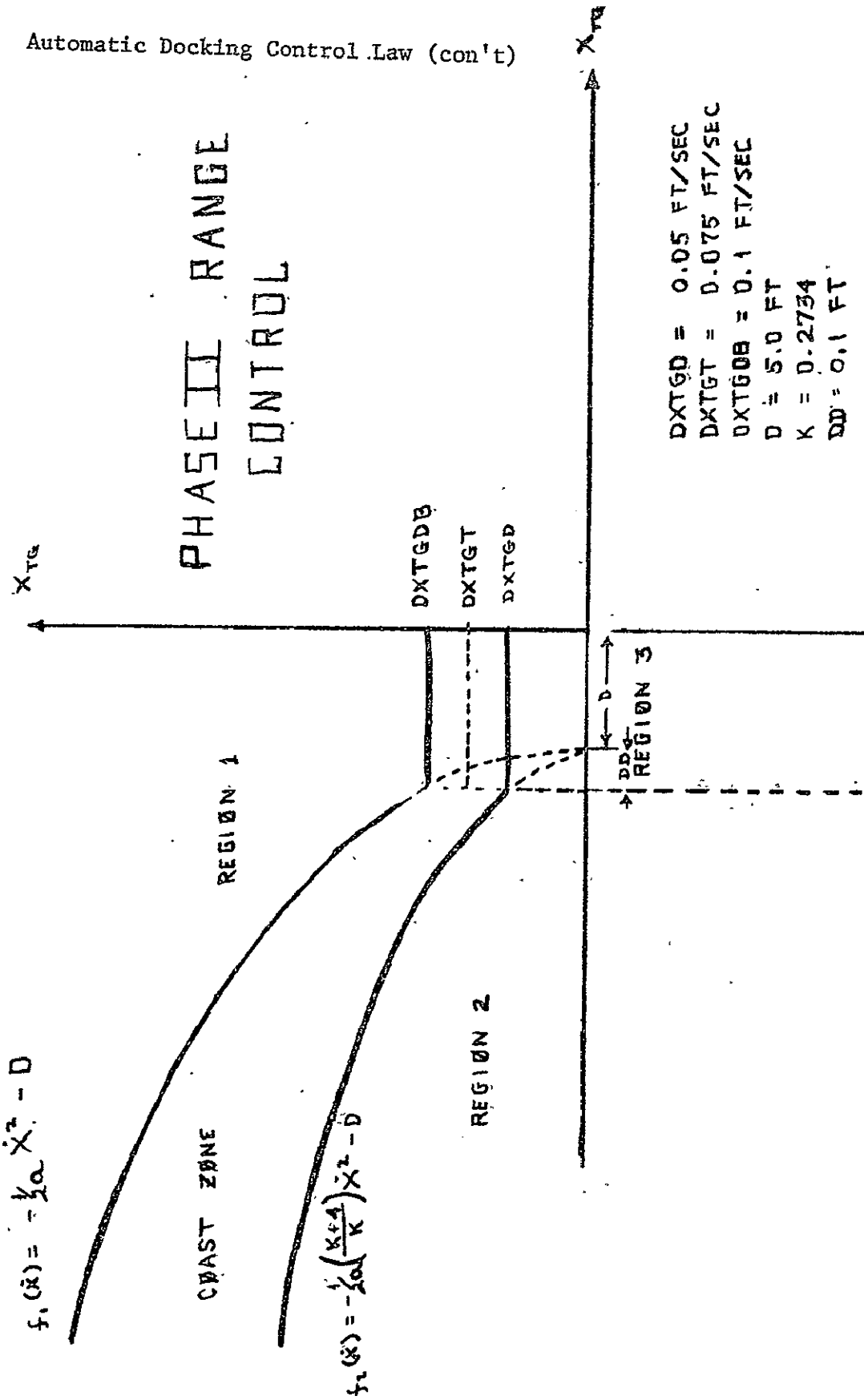
DXDB = 0.25 FT/SEC (SEE PHASE PLANE)

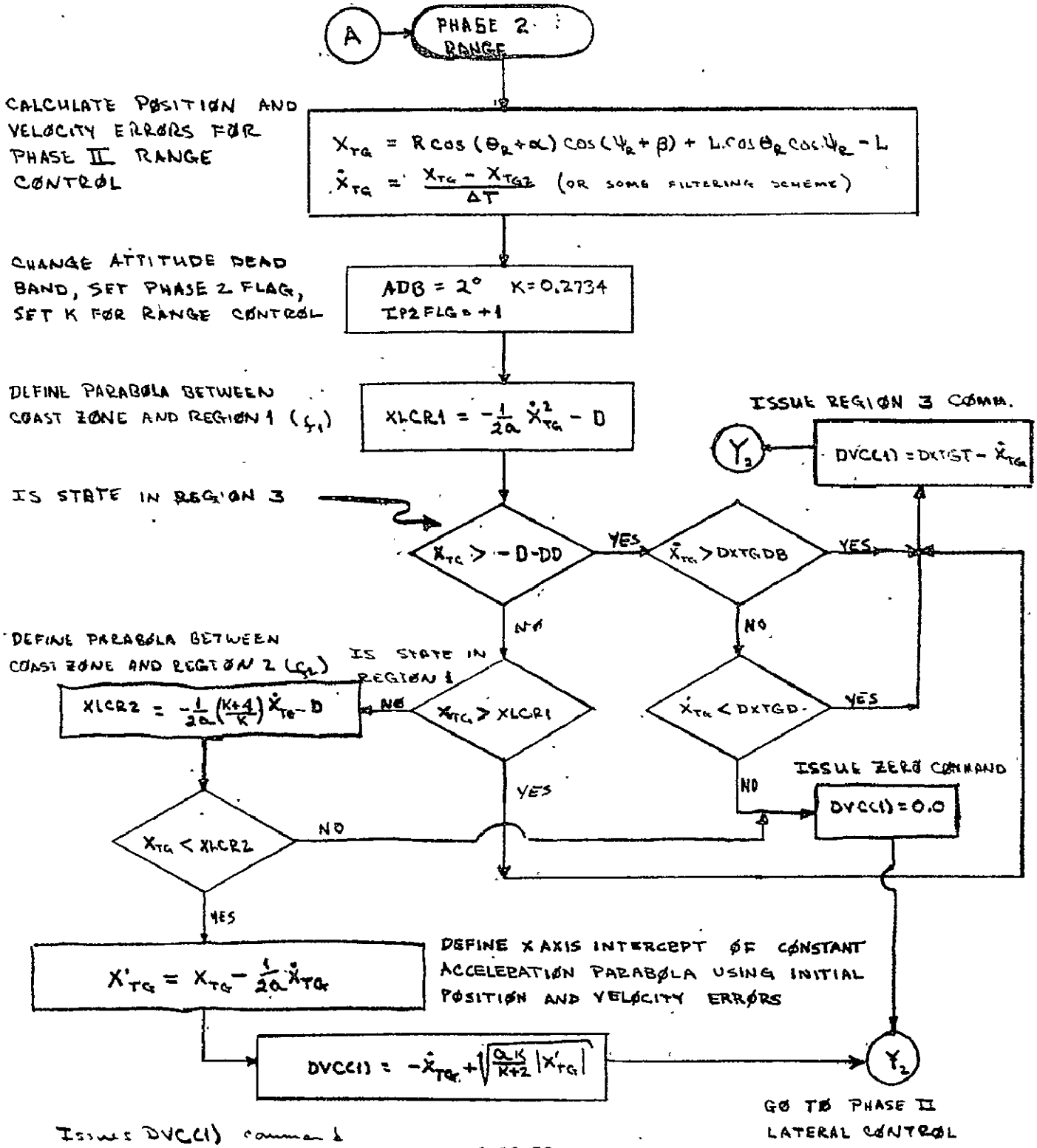
XDBP = 4.5 FT (SEE PHASE PLANE)

DXT = 0.125 FT/SEC = $\dot{X}$ DESIRED FØR REGIØN 1

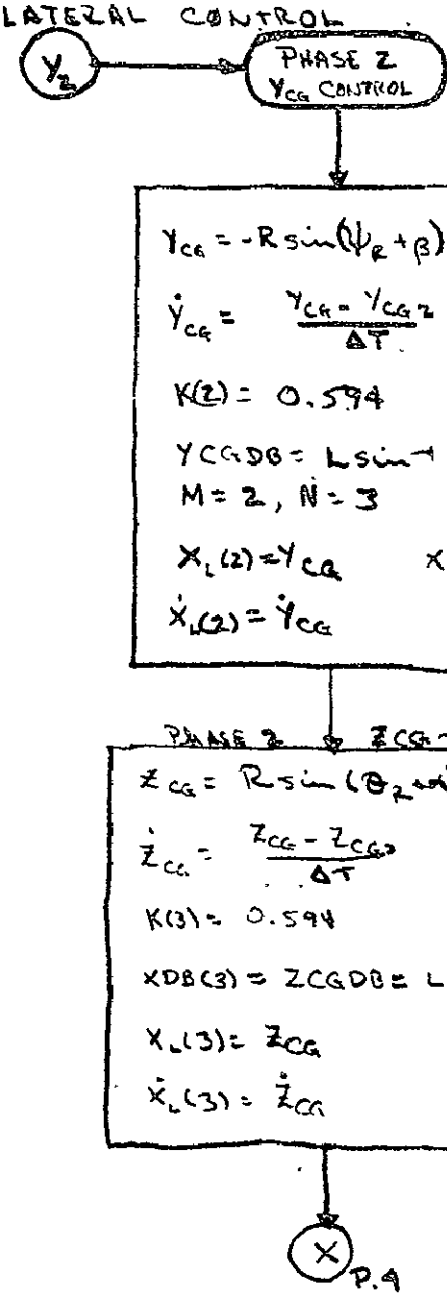
* Y AND Z-AXIS CØMMAND LØGIC IS EXACTLY LIKE THE X-AXIS LØGIC EXCEPT FØR THE FØLLØWING CØNSTANT

K = 0.594

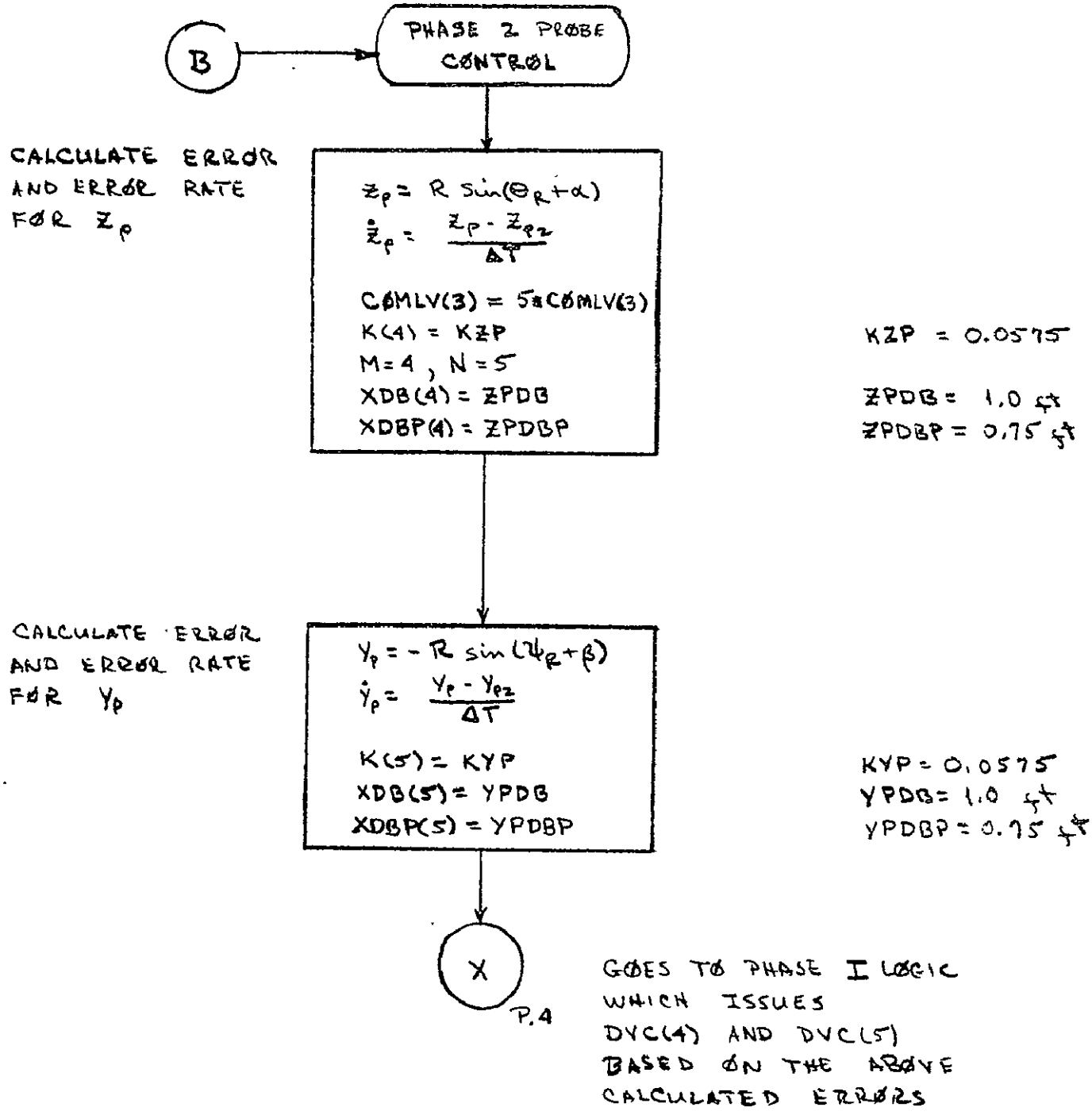

PHASE II RANGE CONTROL
$\dot{X}_{TG}$
X_{TG}
DXTGDB
DXTGT
DXTGD
DXTGD = 0.05 FT/SEC
DXTGT = 0.075 FT/SEC
DXTGDB = 0.1 FT/SEC
D = 5.0 FT
K = 0.2734
DD = 0.1 FT
REGION 1
REGION 2
REGION 3
COAST ZONE
D
DD
$f_1(\dot{X}) = -\frac{1}{2}a\dot{X}^2 - D$
$f_2(\dot{X}) = -\frac{1}{2}a\left(\frac{K+1}{K}\right)\dot{X}^2 - D$


A
PHASE 2 RANGE
CALCULATE POSITION AND VELOCITY ERRORS FOR PHASE II RANGE CONTROL
$X_{TG} = R\cos(\theta_R+\alpha)\cos(\psi_R+\beta) + L\cos\theta_R\cos\psi_R - L$
$\dot{X}_{TG} = \frac{X_{TG} - X_{TG2}}{\Delta T}$ (OR SOME FILTERING SCHEME)
CHANGE ATTITUDE DEAD BAND, SET PHASE 2 FLAG, SET K FOR RANGE CONTROL
ADB = 2° K=0.2734
IP2FLG = +1
DEFINE PARABOLA BETWEEN COAST ZONE AND REGION 1 (C_1)
$XLCR1 = -\frac{1}{2a}\dot{X}_{TG}^2 - D$
ISSUE REGION 3 COMM.
Y_2
DVCC(1) = DXTST - $\dot{X}_{TG}$
IS STATE IN REGION 3
$\dot{X}_{TG} > -D-DD$
YES
$\dot{X}_{TG} > DXTGDB$
YES
NO
NO
DEFINE PARABOLA BETWEEN COAST ZONE AND REGION 2 (C_2)
IS STATE IN REGION 1
$XLCR2 = -\frac{1}{2a}\left(\frac{K+4}{K}\right)\dot{X}_{TG} - D$
NO
$X_{TG} > XLCR1$
$\dot{X}_{TG} < DXTGD$
YES
ISSUE ZERO COMMAND
YES
NO
DVCC(1) = 0.0
$X_{TG} < XLCR2$
NO
YES
$X'_{TG} = X_{TG} - \frac{1}{2a}\dot{X}_{TG}^2$
DEFINE X AXIS INTERCEPT OF CONSTANT ACCELERATION PARABOLA USING INITIAL POSITION AND VELOCITY ERRORS
$DVCC(1) = -\dot{X}_{TG} + \sqrt{\frac{2aK}{K+2}|X'_{TG}|}$
Y_2
GO TO PHASE II LATERAL CONTROL
Issues DVCC(1) comma b

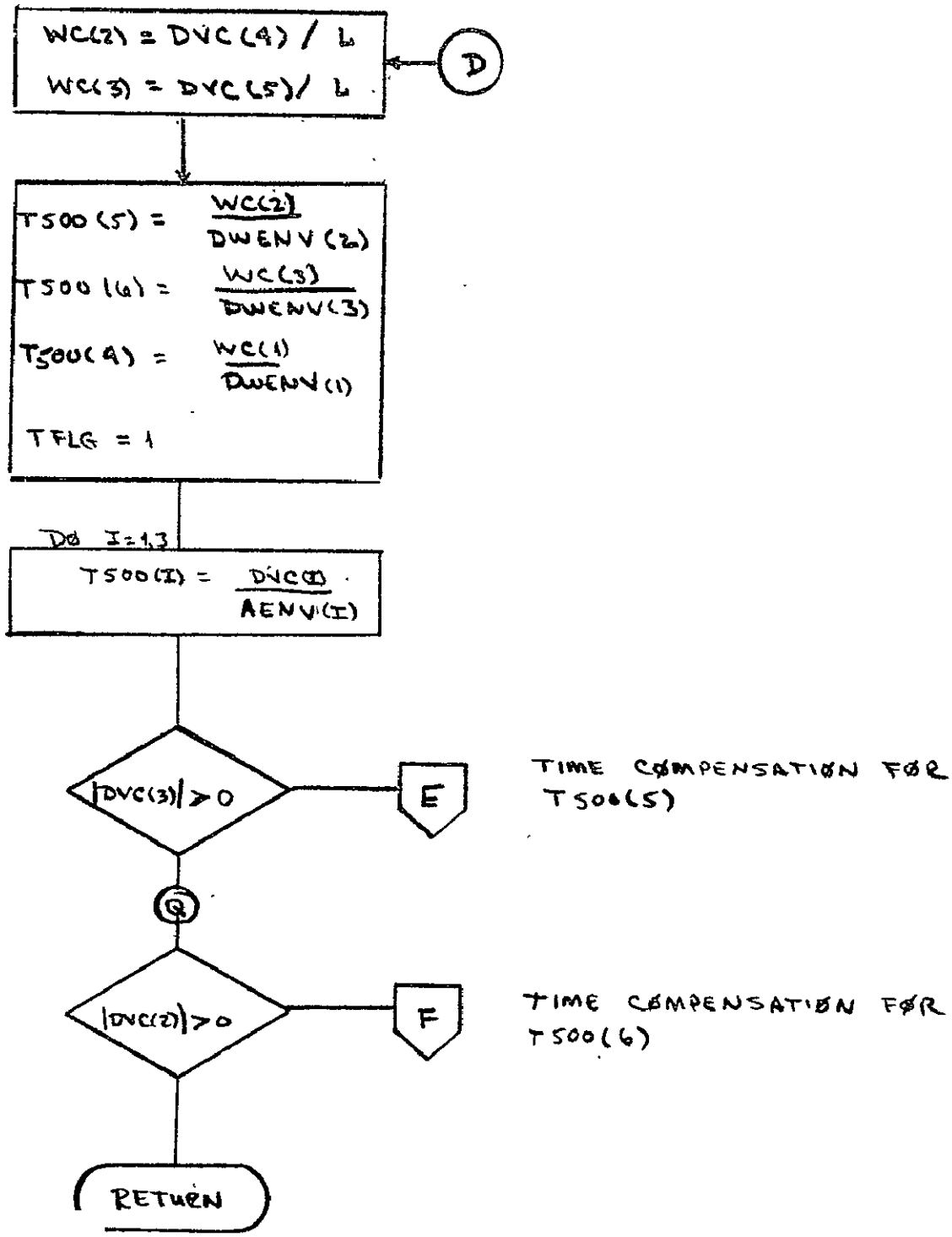
PHASE II LATERAL CONTROL

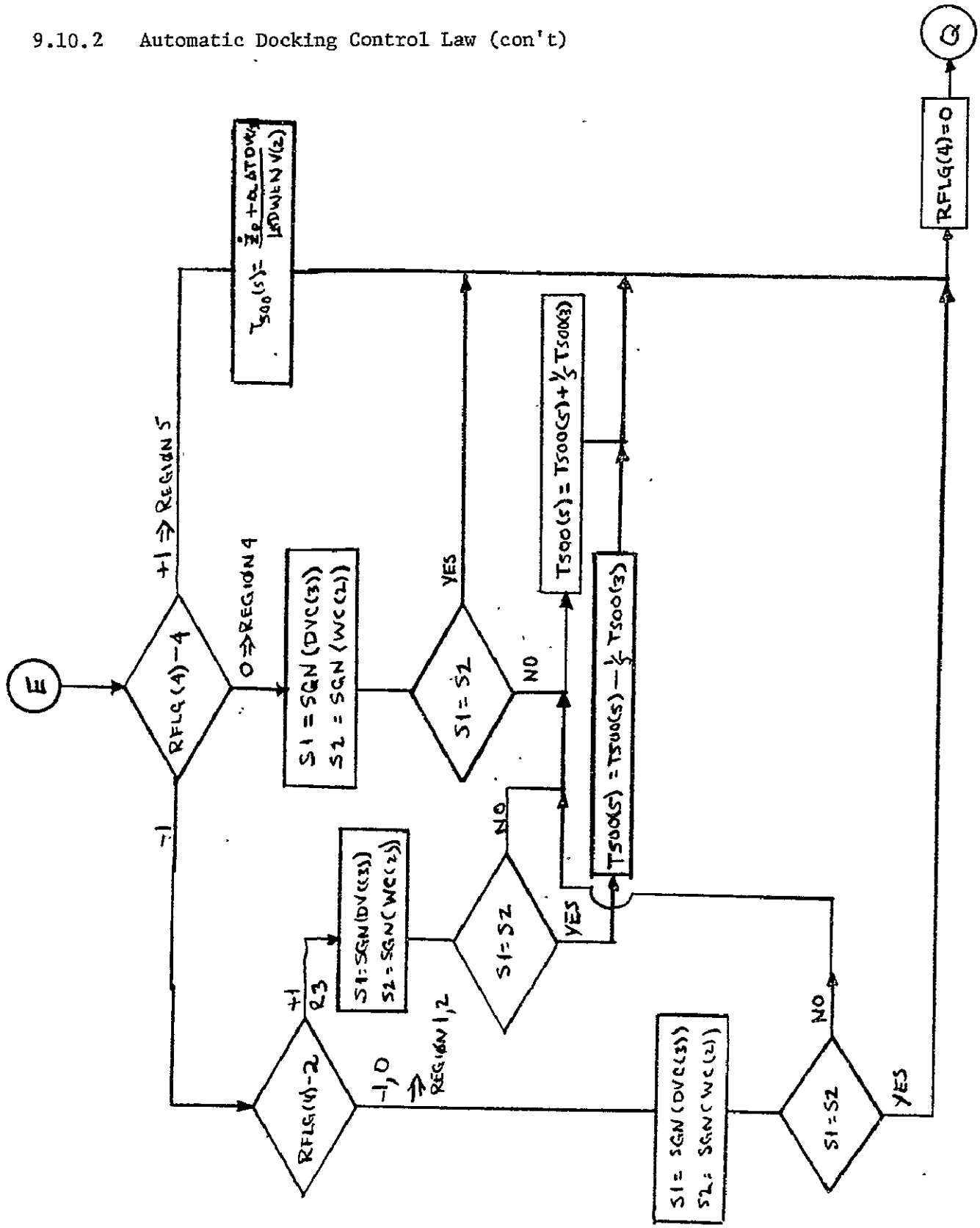


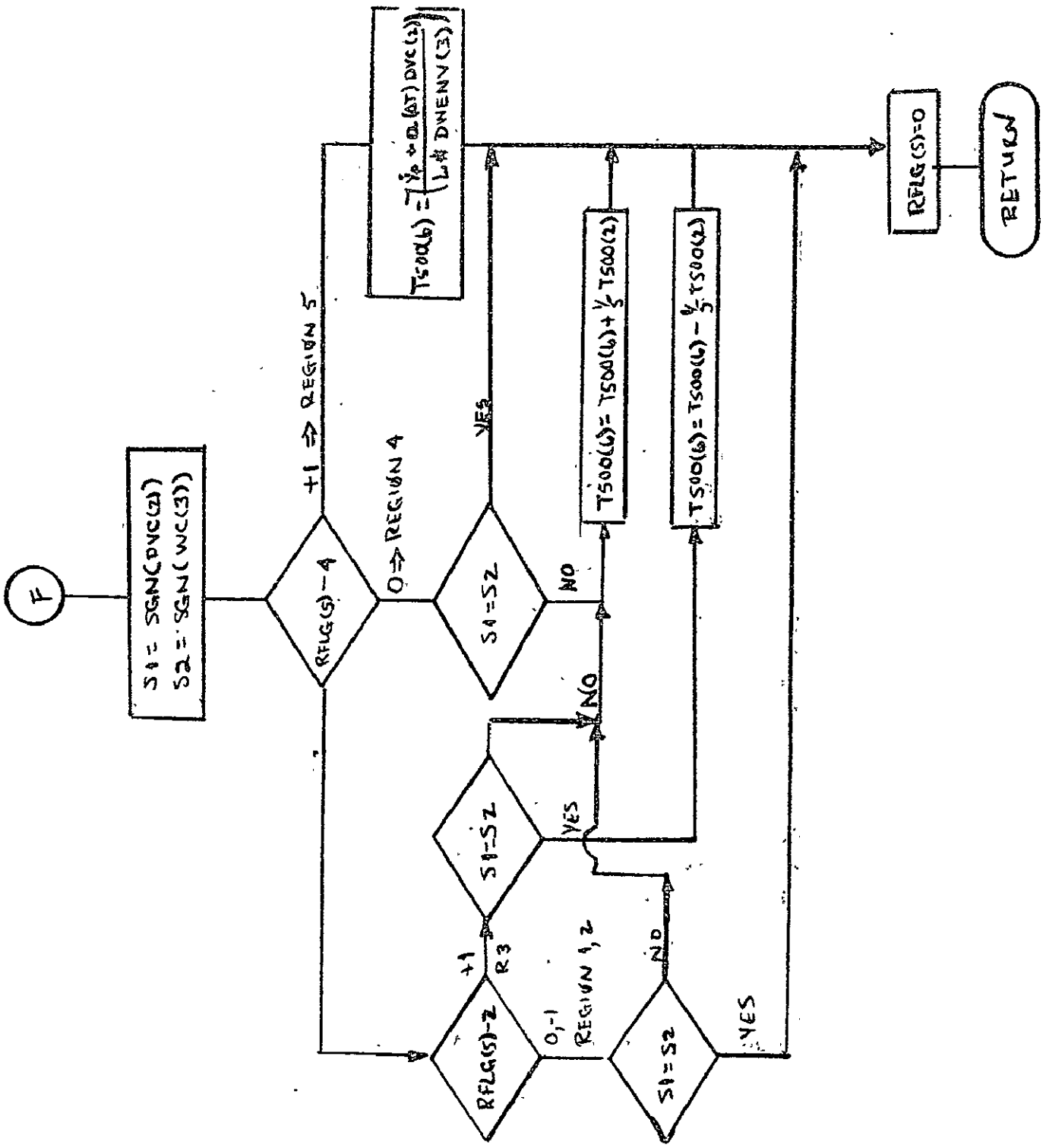
CONTINUE THROUGH PHASE I LOGIC
UPON COMPLETION TEST IP2FLG
IF IT IS NOT SET, RETURN.
IF SET, LOAD VARIABLES FOR
PROBE LOGIC (SEE ENTRY B)

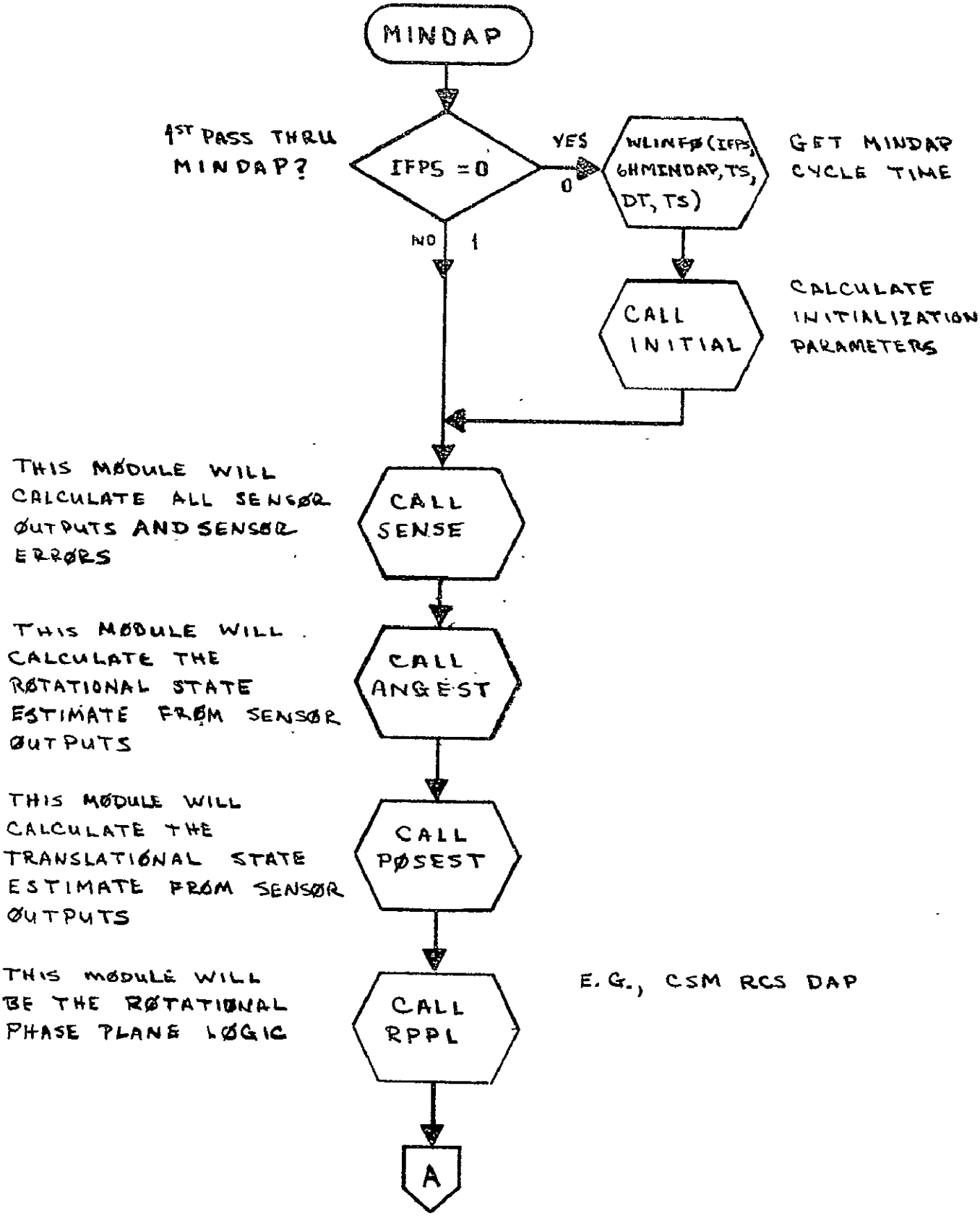

B
PHASE 2 PROBE CONTROL
CALCULATE ERROR AND ERROR RATE FOR Z_p
$Z_p = R \sin(\Theta_R + \alpha)$
$\dot{Z}_p = \frac{Z_p - Z_{p2}}{\Delta T}$
COMLV(3) = 5*COMLV(3)
K(4) = KZP
M=4 , N=5
XDB(4) = ZPDB
XDBP(4) = ZPDBP
KZP = 0.0575
ZPDB = 1.0 ft
ZPDBP = 0.75 ft
CALCULATE ERROR AND ERROR RATE FOR Y_p
$Y_p = -R \sin(\Psi_R + \beta)$
$\dot{Y}_p = \frac{Y_p - Y_{p2}}{\Delta T}$
K(5) = KYP
XDB(5) = YPDB
XDBP(5) = YPDBP
KYP = 0.0575
YPDB = 1.0 ft
YPDBP = 0.75 ft
X
P.4
GOES TO PHASE I LOGIC WHICH ISSUES DVC(4) AND DVC(5) BASED ON THE ABOVE CALCULATED ERRORS

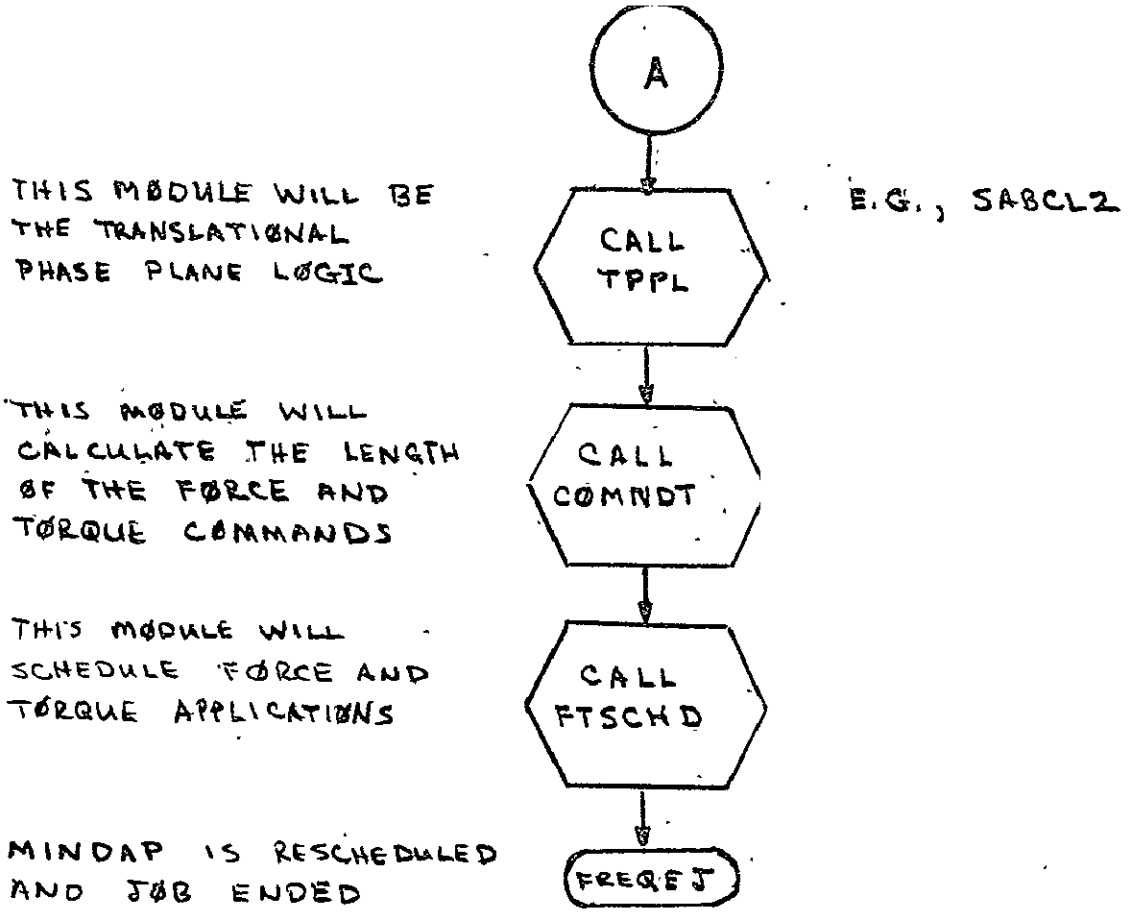
THE FOLLOWING LOGIC COMPENSATES FOR $(\dot{Z}, \dot{\theta})_{CG}$ COMMANDS.



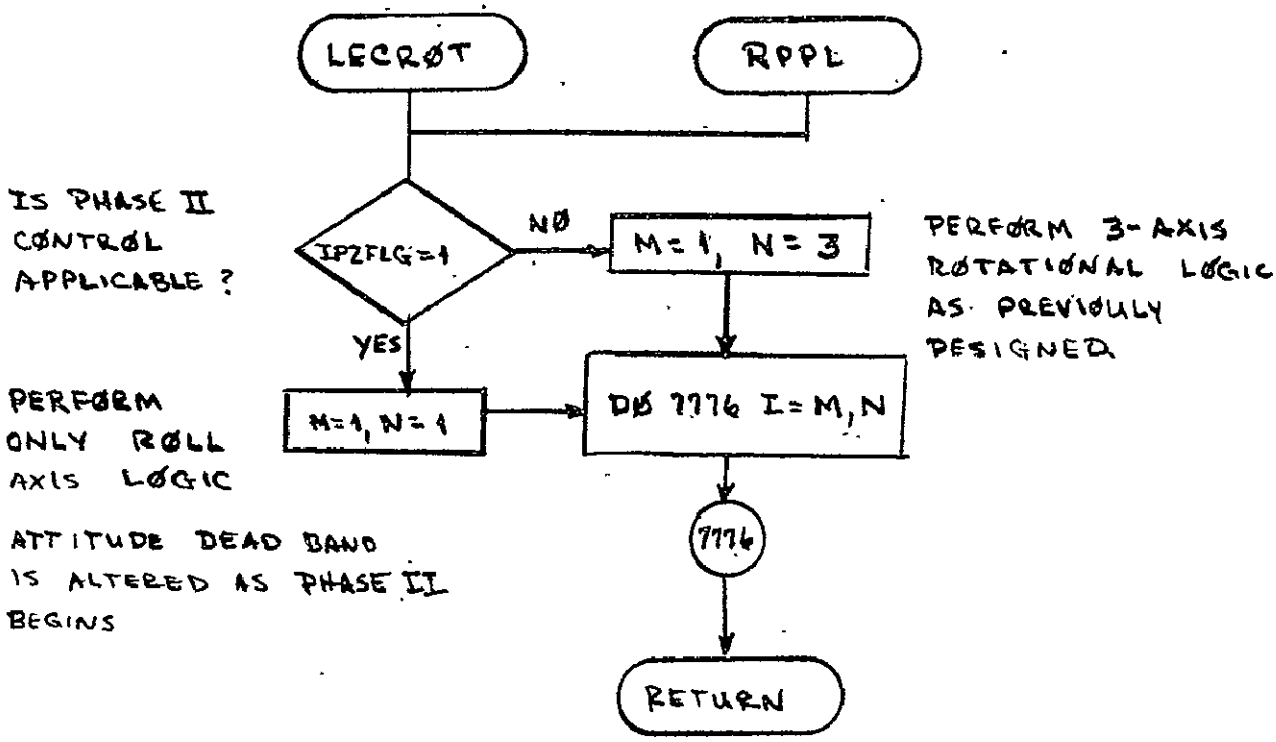

E
RFLG(4)=4
+1 ⇒ REGION 5
0 ⇒ REGION 4
-1
S1 = SGN (DVC(3))
S2 = SGN (WC(2))
S1 = S2
YES
NO
TSUB(S) = TSUB(S) + ½ TSUB(3)
TSUB(S) = TSUB(S) - ½ TSUB(3)
RFLG(4)=0
Q
RFLG(4)=2
+1
R3
-1,0
⇒ REGION 1,2
S1=SGN(DVC(3))
S2=SGN(WC(2))
S1 = S2
YES
NO
S1= SGN (DVC(3))
S2= SGN(WC(2))
S1 = S2
NO
YES


F
S1 = SGN(DVC(2))
S2 = SGN(WC(3))
RFLG(6) = 4
+1 ⇒ REGION 5
0 ⇒ REGION 4
S1=S2
YES
NO
TS00(6) = (½ + Ω(ΔT) DVC(2)) / (L# DWENV(3))
TS00(6) = TS00(6) + ½ TS00(2)
TS00(6) = TS00(6) - ½ TS00(2)
S1=S2
YES
NO
RFLG(5) = 2
+1
R3
0,-1
REGION 1, 2
RFLG(5)=0
RETURN

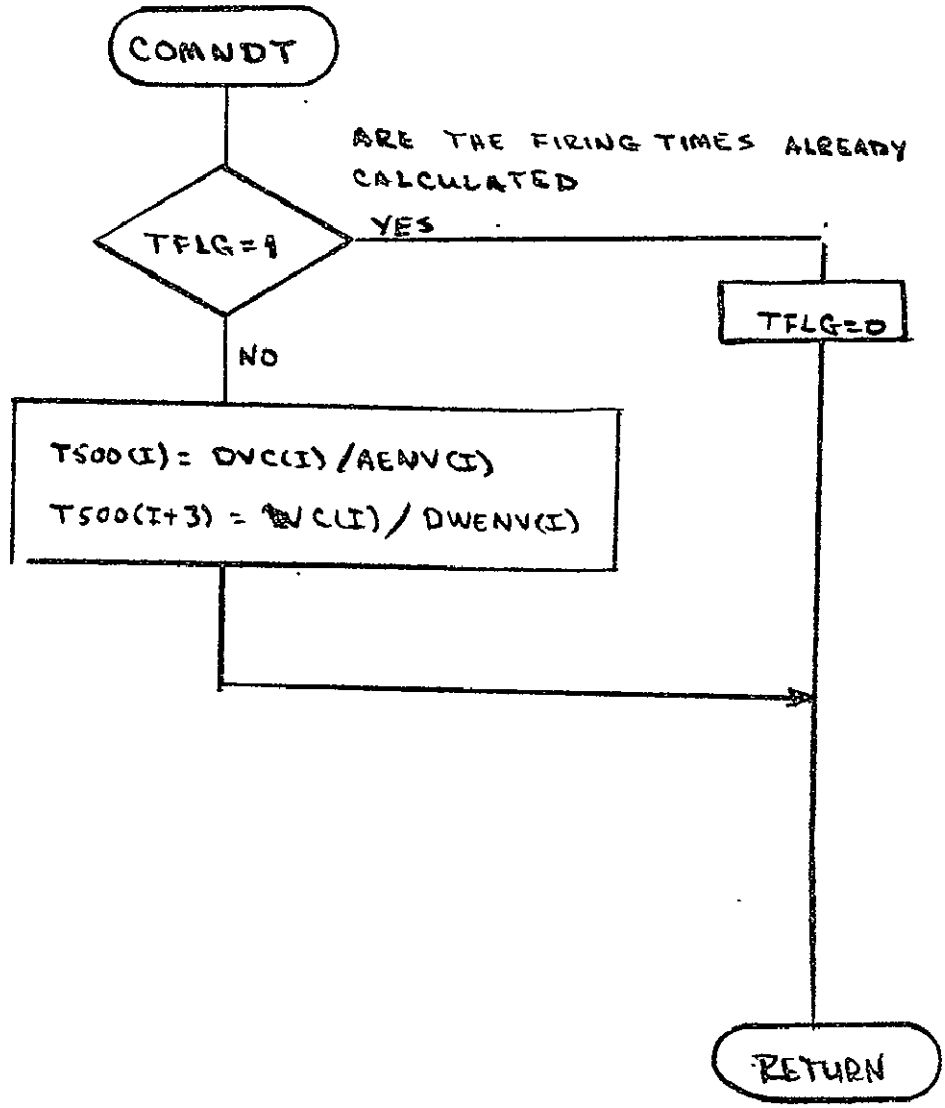

MINDAP
1ST PASS THRU MINDAP?
IFPS = 0
YES
0
NO
1
WLINFØ (IFPS, 6HMINDAP, TS, DT, TS)
GET MINDAP CYCLE TIME
CALL INITIAL
CALCULATE INITIALIZATION PARAMETERS
THIS MØDULE WILL CALCULATE ALL SENSØR ØUTPUTS AND SENSØR ERRØRS
CALL SENSE
THIS MØDULE WILL CALCULATE THE RØTATIONAL STATE ESTIMATE FRØM SENSØR ØUTPUTS
CALL ANGEST
THIS MØDULE WILL CALCULATE THE TRANSLATIONAL STATE ESTIMATE FRØM SENSØR ØUTPUTS
CALL PØSEST
THIS MØDULE WILL BE THE RØTATIONAL PHASE PLANE LØGIC
CALL RPPL
E.G., CSM RCS DAP
A


A
THIS MODULE WILL BE THE TRANSLATIONAL PHASE PLANE LOGIC
CALL TPPL
E.G., SABCL2
THIS MODULE WILL CALCULATE THE LENGTH OF THE FORCE AND TORQUE COMMANDS
CALL COMNDT
THIS MODULE WILL SCHEDULE FORCE AND TORQUE APPLICATIONS
CALL FTSCHD
MINDAP IS RESCHEDULED AND JOB ENDED
FREQFJ

THE FOLLOWING IS A MODIFICATION TO THE ROTATIONAL LOGIC TO INCORPORATE PHASE II ROTATIONAL LOGIC.



ILCULATE LENGTH
: TIME FOR
)RCE AND
SRGUE COMMANDS



Reference

EG 2-70-149, "Docking Sensor Error Model," dated 16 September 1970.

60

9.11 DOCKED OPERATIONS

The GN&C functions during docked operations are undefined. Some of the candidate functions are the following:

1. Targeting for Rendezvous, Deorbit, Orbit modification.
2. Absolute and Relative Navigation.
3. Provide Guided ΔV's to the Space Station.
4. Attitude Control of the docked cluster.
5. Sensor Calibration and Alignment.
6. System Monitor, Test and Checkout.

APPENDIX A

APPICABLE APOLLO SOFTWARE

The concepts represented by portions of the Apollo on-board software have been designated as being applicable to the Shuttle Onboard Software. These applicable Apollo concepts identified in the references below represent a foundation on which to develop shuttle software:

Guidance System Operations Plan for Manned CM Earth Orbital and Lunar Missions Using Program COLOSSUS 2E Section 5. Guidance Equations (Rev. 12)

P30 - External ΔV Maneuver Guidance pg. 5.3-17

P51 - IMU Orientation Determination pg. 5.6-2

P52 - IMU Realignment Program pg. 5.6-6