

Real Time Computer Complex

Apollo Programming Systems

BOOK: *MISSION SYSTEMS*

AS-200 MISSION PLANNING

(NASA-CR-130938) APOLLO PROGRAMMING
SYSTEMS. VOLUME 1: BOOK - MISSION
SYSTEMS, AS-200 SYSTEMS INTEGRATION
(International Business Machines Corp.)

N73-71844

Unclas

00/99 17096

Volume 1

IBM

Federal Systems Division

NAS 9-996

IBM

RTCE APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

Title Page
DATE 11/29/66
REV
PAGE i of ii

APOLLO PROGRAMMING SYSTEMS

Mission Systems

Contract No. NAS 9-996

Federal Systems Division
International Business Machines Corporation
1322 Space Park Drive
Houston, Texas 77058

Copy No. 27

IBM**RTCC APOLLO PROGRAMMING SYSTEMS**

BOOK: MISSION SYSTEMS | - AS-200 Mission Planning

Control Sheet

DATE 11/29/66

REV

PAGE ii of ii

INSERTION CONTROL SHEET

As each program is distributed, please enter the date you receive it opposite the insertion number listed below. If you miss an insertion, or if you want to verify that your book is up-to-date, please call Ray Lundin and: (1) request the missing program or (2) ask what was the number of the last insertion issued.

Insertion Date	Insertion Date	Insertion Date	Insertion Date
1. <u>4/21/67</u>	23. <u>11-14-67</u>	45. _____	67. _____
2. <u>6/16/67</u>	24. <u>11-21-67</u>	46. _____	68. _____
3. <u>6/19/67</u>	25. <u>11-27-67</u>	47. _____	69. _____
4. <u>6/22/67</u>	26. <u>12-11-67</u>	48. _____	70. _____
5. <u>7-7-67</u>	27. <u>12-13-67</u>	49. _____	71. _____
6. <u>7-13-67</u>	28. <u>1-2-68</u>	50. _____	72. _____
7. <u>7-28-67</u>	29. _____	51. _____	73. _____
8. <u>7-31-67</u>	30. _____	52. _____	74. _____
9. <u>8-12-67</u>	31. _____	53. _____	75. _____
10. <u>8-11-67</u>	32. <u>1-8-68</u>	54. _____	76. _____
11. <u>8-25-67</u>	33. _____	55. _____	77. _____
12. <u>8-28-67</u>	34. <u>1-31-68</u>	56. _____	78. _____
13. <u>8-31-67</u>	35. <u>2-28-68</u>	57. _____	79. _____
14. <u>9-15-67</u>	36. <u>4-12-68</u>	58. _____	80. _____
15. <u>9-21-67</u>	37. <u>5/10/68</u>	59. _____	81. _____
16. <u>9-28-67</u>	38. <u>5/24/68</u>	60. _____	82. _____
17. <u>10-3-67</u>	39. <u>6/28/68</u>	61. _____	83. _____
18. <u>10-10-67</u>	40. <u>10/21/68</u>	62. _____	84. _____
19. <u>10-23-67</u>	41. _____	63. _____	85. _____
20. <u>10-31-67</u>	42. _____	64. _____	86. _____
21. <u>11-8-67</u>	43. _____	65. _____	87. _____
22. <u>11-10-67</u>	44. _____	66. _____	88. _____

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS

Foreword
DATE 7/31/67
REV
PAGE i of xiii

FOREWORD

Introduction

Mission Systems program documentation is published in a set of seventeen books which are part of the Apollo Programming Systems documentation series. (This series includes Mission Systems, Simulation, and RTOS User's Guide.) Each Mission Systems Department subsystem maintains two books (one for the Apollo 200 series missions and one for the Apollo 500 series missions). Each book is titled with the name of the subsystem to which it applies and contains the task and load module writeups for that subsystem. The Mission Systems Integration Department also maintains a book for each series. The contents of these books include task and load module writeups for Mission Control programs and data tables, MED formats and library routine writeups for all subsystems. (See Figure 1 for book titles.)

This documentation is intended to:

- Assist programmers in maintaining and coding programs
- Provide an information tool for managers
- Provide NASA with a description of the completed programs
- Provide the library with reference and historical information about the project.

Publication Procedures

The publication procedures as shown in Figure 2 are described in the following paragraphs. After a program has been coded and described in a writeup (format discussed on page iv), the programmer completes a Technical Services Request Form (form #66-5000, available in secretarial areas) and attaches it to the writeup. Figure 3 illustrates information to be supplied by the programmer.

The writeup is submitted to the programmer's manager for technical approval. The manager reviews the program, signs the request form and forwards the document to the Mission Systems Documentation Coordinator.

The Coordinator logs receipt of the document and examines it for correct format. Format errors are called to the programmer's attention for correction.

Copies of the completed document are made for the Mission Program Design Guide (MPDG) and the original is forwarded to Technical Services. The MPDG

**RTCC APOLLO PROGRAMMING SYSTEMS**

BOOK: MISSION SYSTEMS

Foreward

DATE 7/31/67

REV

PAGE ii of xiii

MISSION SYSTEMS - General
MISSION SYSTEMS - AS-200 Systems Integration
MISSION SYSTEMS - AS-500 Systems Integration
MISSION SYSTEMS - AS-200 Digital Command System
MISSION SYSTEMS - AS-500 Digital Command System
MISSION SYSTEMS - AS-200 Launch/High Speed Abort
MISSION SYSTEMS - AS-500 Launch/High Speed Abort
MISSION SYSTEMS - AS-200 Mission Planning
MISSION SYSTEMS - AS-500 Mission Planning
MISSION SYSTEMS - AS-200 Orbit Trajectory Computations
MISSION SYSTEMS - AS-500 Orbit Trajectory Computations
MISSION SYSTEMS - AS-200 Reentry Computations
MISSION SYSTEMS - AS-500 Reentry Computations
MISSION SYSTEMS - AS-200 Telemetry
MISSION SYSTEMS - AS-500 Telemetry
MISSION SYSTEMS - AS-200 Trajectory Determination
MISSION SYSTEMS - AS-500 Trajectory Determination

Figure 1. Apollo Programming Systems Books*

*Used For Documentation of Mission Systems Programs

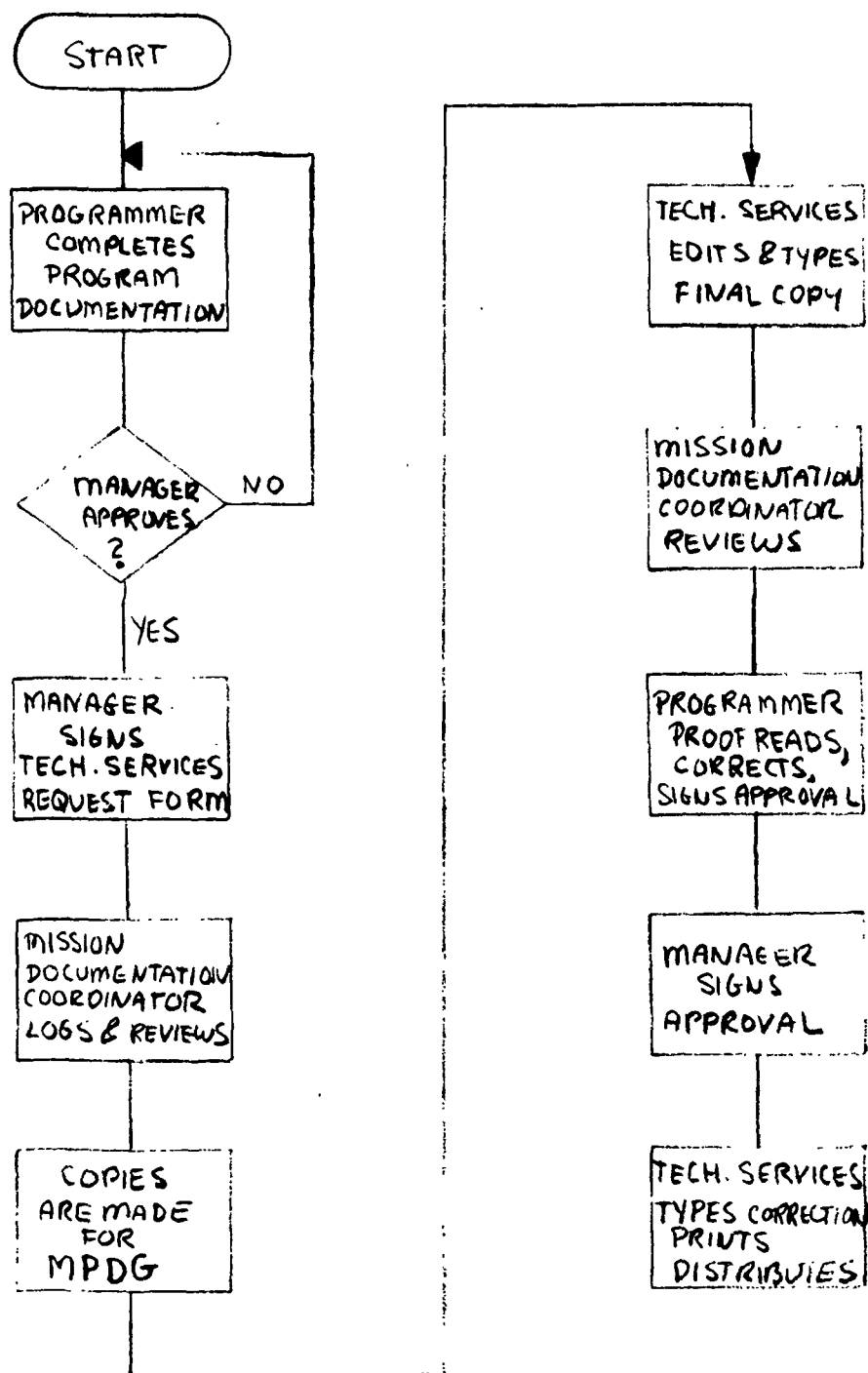


Figure 2. Publication Procedure



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS

Foreword

DATE 7/31/67
REV
PAGE iv of xiii

Document or Job Number	Charge	No. Pages	No. Copies	Assigned to:
	2010	10		
TITLE EMMJC LNI (206)				
Originator	Phone	Mail Code	Department	Location
J. SMITH	3456	14	HRS	D13
INSTRUCTIONS Complete this portion for all requests				
PREPARATION		The check items below include only standard publication services. All other services should be requested under SPECIAL INSTRUCTIONS.		
☒ EDIT	☐ ART	Line	Slide	
☒ TYPE	(Circle)	Flipchart	Sign	
REPRODUCTION (Circle)				
REPRO METHOD	REPRODUCE ON	FINAL SIZE		
Press Bruning	Bond Vellum	8.5 x 11	☐ UNPUNCHED	
Xerox AB Dick	IBM Card Cover	11 x 17	☐ 7-HOLE PUNCH	
			☐ COLLATE ☐ STAPLE	
			☐ CERLOX BINDING	
SPECIAL INSTRUCTIONS				

ITEM

INFORMATION SUPPLIED

Charge
No. Pages
Title
Originator
Phone, Dept.,
Mail Code,
Location
Preparation
Documents
Manager

Work Package Number
Number of written faces
Program name, data table name, etc
Programmer's name

Self explanatory

Check(☒) Edit and Type
Check(☒) Release or Revision
Programmer's Department Manager's signature

Figure 3. Technical Services Request Form

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS

Foreword
DATE 7/31/67
REV
PAGE v of xiii

copies are maintained for programmer reference until the document is published. Guides are located in each Mission Systems secretarial area.

The Technical Services Department edits the document, types it in final form for photographic reproduction, and returns the original and typed copies to the Mission Systems Documentation Coordinator for review of typed format. He forwards the document to the programmer.

The programmer proofreads the typed document and makes any necessary corrections. He then signs the first approval block and sends the original copy and the corrected typed copy to his manager. The manager signs the second approval block and returns both copies to Technical Services.

The approved program writeups are printed and distributed to holders of the applicable documentation books. Original drafts are retained by Technical Services. The programmer receives a published copy of his document. Any revisions to the document are processed in the same manner.

Program Writeup Procedure

The programmer documents each task, load module, MED entry and data table assigned to him. The required information is outlined on standard draft forms (Figures 4-13) available in Mission Department secretarial areas. The following paragraphs describe the forms and procedures to be used in preparation of documentation.

Draft forms have standard headings which permit the programmer to supply all information necessary to identify the location of writeups when published. Each of the following must be provided.

Book — Name of the Apollo Programming Systems book in which the writeup will be published (see Figure 1 for names). When the document applies to both 200 and 500 series missions, both book names are to be provided.

Section — Section identification consists of section number concatenated with symbolic name of task, load module MED entry of data table. Section numbers are determined by referencing the book table of contents.

Page — Pages of the draft are numbered sequentially beginning with one (1). Page one of the draft should indicate the total number of pages included (i. e., "1 of x", when x is the total of written faces including flowcharts).

The programmer is required to complete only those forms which are applicable to his assignment; however, each item outlined on the selected forms is to be completed. When the requested information is not available, an indication is to be made as to whether the information is not applicable (NA) or to be provided later.

Selection of appropriate forms should be based on the following:

- a. Task Writeups - Each independent task is to be documented. Four forms are provided to be used as indicated below.
 1. TASK WRITEUP (Figure 4) - Required for each task
 2. TASK RESOURCE TABLE (Figure 5) - Required if applicable
 3. TASK SUBPOOL LIST (Figure 6) - Required if applicable
 4. FLOWCHART (Figure 7) - Required for each task; provides a general logic flow showing the relationships of load modules required to perform the task.
- b. Program Writeups - Each load module is to be documented. Five forms are provided to be used as indicated below. Additional tables may be provided as the user requires.
 1. PROGRAM WRITEUP (Figures 8) - Required for each load module. For mathematical routines, "METHOD" should contain the derivation and proof of the method used. In addition, the source of the derivation or reference to the source document should be detailed enough to explain the method. Additional comments which may assist the reader may be included.

Any data tables and MED entries used by the load module are to be referenced in the appropriate place in the writeup; however, detailed documentation of these items is to be published separately.
 2. DTRROUTE, RTQCNTRL, RTATTACH MACROS (Figure 9) - Required when used by the load module; provides format and purpose of each use in program.



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS

Foreword

DATE 7/31/67

REV

PAGE vii of xiii

3. SYSTEM PARAMETERS (Figure 10) - Required when used by load module; provides a list of all Mission Systems Parameters used in program.
 4. PARAMETER LIST (Figure 11) - Required when used by load module; one form prepared for each list.
 5. FLOWCHART (Figure 7) - Required for each load module; provides detailed logic flow of program.
- c. MED Formats (Figure 12) - One form is to be completed for each MED code.
- d. Data Table Formats (Figures 13) - A form is to be completed for each data table. The columns, BLOCK, WORD, BYTE and BIT refer to actual location of data within the table. Only those columns necessary to describe data are to be used. If the data is handled as separate bits, the bit location is noted and the type column is blank. If the item is a full or double word, byte and bit location are not supplied.

966-6 SW
IBM RTCC APOLLO PROGRAMMING SYSTEMS
 BOOK MISSISSIMM SYSTEMS - AS200/500 DES
 SECTION LECTURE DISK
 DATE _____
 REV 2 of 9
 PAGE _____
 MISSION AS200/500

Task Resource Table

Byte No.	Content
1	Status byte for csm Navigation Update
2	" " " Descent Target Primary Update
3	" " " Descent Target Manual A "
4	" " " " " B "
5	" " " Descent Target A Update
6	" " " " " B "
7	" " " Minimum Angular Target Update
8	" " " Time Churnout Update
9	" " " Lift-off time Update
10	" " " Universal Target A Update
11	" " " " " B "
12	" " " TFFMIN Update
13	" " " REE-SMAT Update
14-40	Reserved
41	Status byte for SV Line Base Update
42	" " " Sequence Estimate Update Mail
43	" " " " " 2
44	" " " " " 3

66 3600-1

Figure 5. Task Resource Table

IBM RTCC APOLLO PROGRAMMING SYSTEMS
BOOK. MISSION SYSTEMS - AS 200/500 DCS
TASK WRITEUP

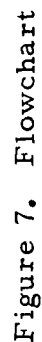
Section L. CDRSJSX
DATE _____
REV _____
PAGE 1 of 9
MISSION 200/500

a. Task Symbol — CDRJSX
b. Task Name — DCS SUBSYSTEM TASK
c. Task Type — Independent ☒ or dependent (check one)
d. Programming Group — DIGITAL COMMAND SUBSYSTEM
e. Writeup Completion Date — 8/22/66
f. _____
g. _____
h. _____
i. _____

FUNCTION
The DCS Subsystem is responsible for
generation, display and transmission of command loads
for the SUB, CSM, and LEM. Output to the program are
from the MEPS and PRT's located on the Computer
Command Console. Command loads generated are
formatted and sent to CCATS for transmission to the
proper command site.

USAGE
a. Inputs — 1- MDP Request see load module
2- DTV Request writeups for
3- PRT Request detail
b. Outputs — 1- DTV Output
2- CSM, SUB add LEM command loads

Figure 4. Task Writeup

Figure 6. Task Subpool List

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS

Foreword
DATE 7/31/67
REV
PAGE x of xiii

IBM RTCC APOLLO PROGRAMMING SYSTEMS
BOOK: MISSION SYSTEMS - AS 200600 TRAJECTORY DETERMINATION

SECTION 2: BMQLSMED
DATE 10614
REV PAGE
MISSION AS-200600 TRAJECTORY DETERMINATION AS-201

e. Data Restrictions and Limits - Restrictions checked by the MED decoder are described in the MED format Tables. Vehicle and beacon numbers are checked by BMQLSMED

f. Error - If an invalid vehicle or beacon number is received, an error message is printed and the MED is rejected by BMQLSMED

g. Other -

METHOD

Given detailed description of the method, refer to the program flow chart. A brief description of the purpose of the two entries to BMQLSMED is given below.

QUEID = 1

The external station numbers and radar type numbers from GESTCH are placed in BZLSPEM where they are used by BMQLSIMP to identify the source of input radar messages.

QUEID = 2

The routine of low-speed radar data to BMQLSIMP is initiated.

QUEID = 3

Routine of low-speed radar to

66-3598-1

(Sheet 2 of 2)

IBM RTCC APOLLO PROGRAMMING SYSTEMS
BOOK: MISSION SYSTEMS - AS 200600 TRAJECTORY DETERMINATION

SECTION 2: BMQLSMED
DATE 10614
REV PAGE
MISSION AS-200600 TRAJECTORY DETERMINATION

PROGRAM WRITEUP

a. Symbol - BMQLSMED

b. Program Name - LOW SPEED MED PROCESSOR

c. Programming Group - TRAJECTORY DETERMINATION

d. Programmer(s) - CAROLYN WARD

e. Language - 360/ASSEMBLER

f. Data Tables Used - BZLSPEM, BZLSIMP, GESTCH

g. MED's Used - S24, S25, S26, S27, S28, S29, S30

h. Writeup Completion Date - 9/30/66

FUNCTION

- Process MED's which control low speed radar data collection.
- Control routine of low speed radar inputs to BMQLSIMP
- Anticipate the station characteristics-dependent portion of BZLSPEM

USAGE

a. Calling Sequence -
ATTACH TASK=BZLSDC, EP=BMQLSMED

b. Inputs - REFER TO MED FORMAT TABLES

c. Outputs and Exit - Output is to data table BZLSPEM and BZLSIMP. Exit via RETURN XCTL

d. Other Programs Used - BMQCHAIN, BMQLSIMP, GMSMAP

66-3598-1

(Sheet 1 of 2)

Figure 8. Program Writeup

[illegible]

Figure 10. System Parameters

IBM

RTCC APOLLO PROGRAMMING SYSTEMS
ASCO
BOOK: MISCELLANEOUS SYSTEMS - ASCO TRAJECTORY

966-6 53

17

Section: 2
Date: 20 AUGUST 68

Rev: 6 of 14
Page: 200 of 501

DETERMINATION

DTROUTE, RTQCNTL, RTATTACH Macro

Macro - DTROUTE BLSEADR, KMODE=ENAGE, RTYPE=DATA,
TASK=RTLSDE, EP=ENASIMP, DECE=CDL, MSGEN=1,
PURPOSE=ID=1, MESHT=1,8000000,8000000
PURPOSE- Activate Navstar of low-speed radar data.
Macro - DTROUTE BRSEADR, RMODE=DISAB

Purpose - Stop routing of low-speed radar

Macro - RTATTACH TASK=(TISLAY,EP=GMSMAP,ID=3
PARAM=(IFN,4),GSPV=2
Purpose - Update the low-speed radar display
when a display quantity is changed.
Macro -

Purpose -

Macro -

Purpose -

Macro -

Purpose -

66 3579

Figure 9. Macro Writeup

MED Format Table

MED Code: B02
Purpose: REQUEST TRANSMISSION OF ACQUISITION DATA
Load Module: EMRTRAT

Task: GTXSACQ WKALK=2
EP: GMSSACQ
QNAME:
QUEID: 6

Desc. \ Item	1	2	3	4	5	6
Item Name	VEHICLE	REV	RADAR	ΔT		
Input Format	XXXX	XXX	XXXX	HH:MM:SS		
Input Units	EPICDIC		RADAR CODE SOUND CODE			
Program Units	NUMERIC CODE		EPICDIC	FIXED POINT NUMERIC SECONDS		
Scale Factor						
Checking Option	ACTIVE	$\neq$ (PRESENT) = (REV - 1)	BAWD CODE = C/S			
Missing Item Option	ERROR	ERROR	ERROR	= ZERO		
Item Destination	Implementation 0					

Initial Logic:
Final Logic: All output shall be passed in subpool 1
Illegal Phases: All except Orbit, Reentry
Notes:

- Notes:
1. Items 1 and 2 shall be stored as half words
 2. Items 3 and 4 shall be stored as full words

Initiator: P. Pestel
8/31/66

Subsystem Approval: B. Follis
9/1/66

Requirements Control: E. Schommer
66-2656

66-2656

Figure 12. MED Format Table

Section 2 GMSD-1A5E

Sect:
DATE
REV

BOOK: MISSION SYSTEMS - ASAD SYSTEMS INTEGRATION PAGE MISSIUN 306

Parameter List

QNAME -

QNAME —
To-Task — GTPHASE

GP-GMSPI125

QUEID - O
From - GMS PHASE

Purpose - Print ONLINE COMMENT

[illegible]

- 2 - Single-protection (flaming-point)
- D - Double-protection (flaming-point)
- 3 - Nonvolatiles

Type	Used - before 2011 word	Used - after 2011 word
1 - Fixed-price	22	22
2 - Fixed-price	22	22

56 3603

Figure 11. Parameter List

IBM RTCC APOLLO PROGRAMMING SYSTEMS
BOOK: MISSION SYSTEMS - AS200/500 SYSTEMS
AS 9-995

SECTION: 7.E ZMTCSM
DATE: 10/2
REV: 10/2
PAGE: 10/2
MISSION: AS-200/500

Symbolic Name -- ERKMSM (ERKMS-15, ERKMSM)
Programmer -- Q. NAME
Table Name -- CSM EPHEMERIS DATA TABLE (SYB, LEM)
Initialization of any Element Required -- NO

Dimensions

Block -- 806
Header Block -- 48 Bytes
Total Words -- 38,628 Bytes
Bytes

General Description and Usage

This data table is a "circular" ephemeris table containing a header block and 806 data records.

IBM RTCC APOLLO PROGRAMMING SYSTEMS
BOOK: MISSION SYSTEMS - AS200/500 SYSTEMS
AS 9-995

SECTION: 7.E ZMTCSM
DATE: 10/2
REV: 10/2
PAGE: 10/2
MISSION: AS-200/500

Detailed Content:

Block	Word	Byte	Bit	Type	Units	Description and Range
1	1			F		Left limit of ephemeris
	2			F		Right limit of ephemeris
	3			F		Time stop
	4	1		H		Update number
	5	1	5			Update in progress
	6	1	6			Kicker at altitude reached
	7			F		Time fired right limit of ephemeris
	8			D		Time
	9-10			D		Left limit of ephemeris
	11-12			D		Right limit of ephemeris
	13-14			D		Time stop
	15-16			3D		Position vector
	17-18			3D		Velocity vector
	19-20					Same as block 2

Figure 13. Data Table Format

TABLE OF CONTENTS

Foreword

Functional Specifications

Section 1: TASKS

Section 2: LOAD MODULES

<u>Symbolic Name</u>	<u>Mission Number</u>
PCHAPE	207
PIPASVEL	207
PMDDMT	206
PMMCMU	206
PMMDMT	206
PMMEPHEM	205
PMMFUD	206
PMMMPT	206
PMMSCH	205
PMMWTC	206
PMSEXE	206

MISSION PLANNING SUBSYSTEM
FUNCTIONAL SPECIFICATIONS FOR THE MISSION SYSTEM
EARTH ORBITAL RENDEZVOUS PROGRAM

MISSION PLANNING SUBSYSTEM
FUNCTIONAL SPECIFICATIONS FOR THE MISSION SYSTEMS
EARTH ORBITAL RENDEZVOUS PROGRAM

INTRODUCTION

The purpose of the Mission Planning Subsystem is to plan and control all maneuver computations necessary for the Apollo earth-orbit rendezvous missions, to provide the necessary computations to target the chaser vehicle to the orbit of the target vehicle, and to communicate with the Marshall Space Flight Center in Huntsville, Alabama. This subsystem provides the NASA flight controller with a real time prelaunch and midcourse maneuver planning capability.

All programs included in the Mission Planning Subsystem execute under the control of the same real time task (PTMANC). This subsystem is divided into the following logical processing units:

- a. Saturn Launch Vehicle (SLV) Targeting
- b. General Purpose Maneuver (GPM)/Perigee Adjust
- c. Two-Impulse (T/I)
- d. Docking Initiate (DKI)/ Co-elliptic Sequence (SPQ)
- e. Maneuver Control

The programs are designed to give the flight controller maximum flexibility and control of all maneuver processing.

Approval	Approval
<i>S.C. James</i> <i>A. Ariatti</i>	<i>J. L. Ste...</i>

Figure 1. Introduction
Flowchart

MISSION PLANNING SUBSYSTEM UNITS

Saturn Launch Vehicle (SLV) Targeting Unit

The Saturn Launch Vehicle Targeting Program is exercised in the period of time between the determination of the first vehicle (target) trajectory and the second vehicle (chaser) liftoff. This period of time is called the prelaunch 2 phase of the mission.

The main purpose of the SLV Targeting Program is to compute the Saturn guidance parameters necessary for the second vehicle to be inserted into an orbit whose relationships with the first vehicle orbit are such that rendezvous can be achieved.

The second vehicle ephemeris which may be generated in the prelaunch 2 phase can be used for mission planning purposes. Among these are rendezvous and reentry planning.

The program can be initiated by the following manual requests:

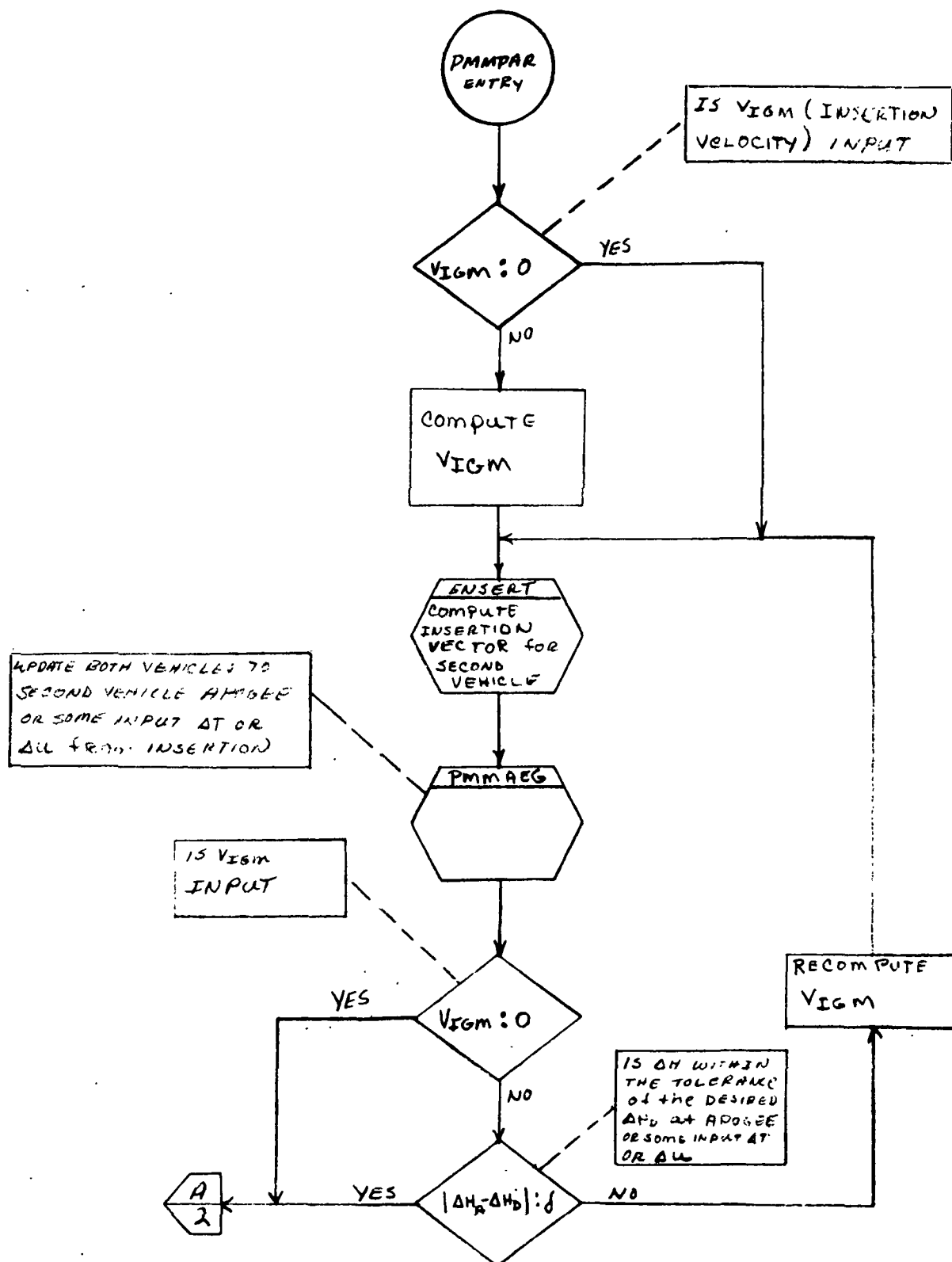
- a. Compute and display recommended liftoff time ($GMTLO_R$) and insertion velocity (V_{IGM}) associated with an input liftoff time in the plane of the launch window at which the second vehicle liftoff must occur in order to achieve the nominal rendezvous plan for the mission under consideration. V_{IGM} is the insertion velocity needed to place the second vehicle into an orbit such that at first apogee (or some input angle or delta time measured from insertion), it is above or below the target orbit by a desired delta altitude, ΔH , (n.m.). If desired a specific velocity can be input.
- b. Compute and display the Saturn guidance parameters associated with an input $GMTLO$ and V_{IGM} .
- c. Perform Step b. In addition route the nominal insertion vector used to compute the targeting parameters to an ephemeris generation program. The insertion vector will be stored for the launch program.
- d. If the condition-for-launch has occurred, perform Step b and save the insertion vector for the launch program. The vector is not routed to generate an ephemeris.

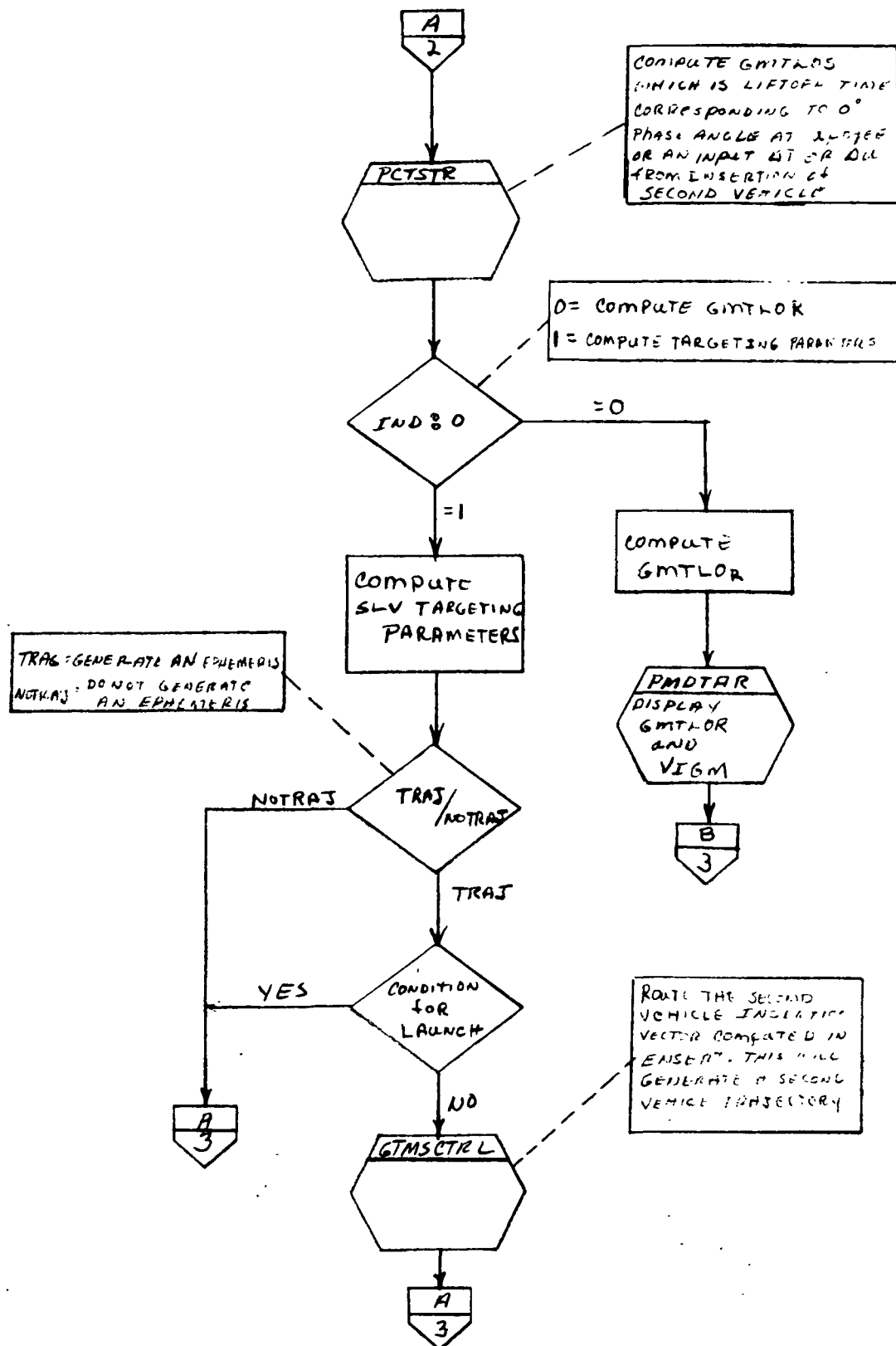
IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

EORP
DATE 6/19/67
REV
PAGE 4 of 42

Also during prelaunch 2 phase there is an Ephemeris Data and Targeting Parameter Transmission Program which is initiated by a manual request. The purpose of this program is to format the current ephemeris data and Saturn guidance parameters in a teletype format and send it to Marshall Space Flight Center (MSFC) simulation computer at Huntsville. When the program is entered, which is prior to second vehicle liftoff, the current ephemeris data is gotten at the input Epoch time, and the Saturn guidance parameters which have been computed are formatted and sent to MSFC.

Figure 2. Saturn Launch Vehicle (SLV) Targeting Unit
Flowchart


 Figure 2. Saturn Launch Vehicle (SLV) Targeting Unit
 Flowchart

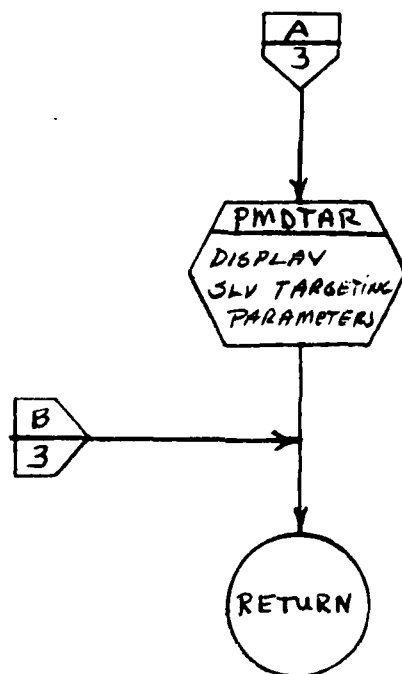


Figure 2. Saturn Launch Vehicle (SLV) Targeting Unit
Flowchart

Load Module Descriptions — The load modules described below are included in the Saturn Launch Vehicle (SLV) Targeting Unit.

PMMPAR — This load module is designed to take an input-planned liftoff time and compute the recommended liftoff time. Then using the recommended liftoff time, the SLV targeting parameters are computed and a trajectory update is generated using the insertion vector which was used to compute the targeting parameters.

PMXTAR — PMXTAR (SLV Transmission Load Module) formats the SLV targeting parameters and transmits the information to Huntsville by a teletype message.

PMDTAR — This load module displays the SLV targeting parameters.

General Purpose Maneuver (GPM)/Perigee Adjust Maneuver Unit

Load Module Descriptions — Described below are the load modules for the GPM/Perigee Adjust Maneuver Unit.

PMMGPM — The function of the General Purpose Maneuver Processor is to provide the flight controller with two main capabilities:

- a. To determine the effect that a specified incremental velocity applied at a given maneuver point (along a given pitch and yaw) will have on the orbit;
- b. To determine the maneuver required to obtain a specified orbit or orbital change.

The first capability is more commonly known as a flight controller special maneuver request and has six options for the maneuver point.

- a. An equatorial (nodal) crossing
- b. A specified longitude
- c. A specified time
- d. A specified height
- e. An apogee crossing
- f. A perigee crossing

The second capability is divided into eight types with various maneuver point options.

- a. A plane change at a certain equatorial crossing, longitude, time or height,
- b. A circularization maneuver at a longitude or height,
- c. A maneuver at perigee to adjust apogee or vice-versa,
- d. A maneuver to adjust the height 180° around from the maneuver point at a longitude or time,
- e. A maneuver to shift the ascending node at an optimum time, longitude, time or height,
- f. A maneuver to obtain a specified apogee and perigee at an optimum time, longitude, time, or height,
- g. A maneuver to shift the ascending node and adjust the height 180° around from the maneuver point at a longitude or time,

- h. A maneuver to shift the line of apsides to the maneuver point and obtain a specified height 180° around from the maneuver point at a longitude, time, or height.

The output of the GPM Processor is a display containing such maneuver information as ΔV , pitch, yaw, maneuver time, maneuver height, etc., and such post maneuver information as apogee and perigee heights, longitude of the ascending node, eccentricity, etc. A table containing the elements before and after the maneuver at the impulsive time is also output so the maneuver may be transferred to the Mission Plan Table if desired.

PMDGPM — The Display Load Module formats all necessary information required on the GPM Display.

PMMPAD — The function of the Perigee Adjust Processor is to compute a series of maneuvers each of which will raise perigee to a specified height while keeping the thrust direction at a constant angle to the horizon. A threshold time, desired perigee height and a desired time increment are the main inputs, while the main display values are the begin time burn, the velocity increment required, and the resultant apogee and perigee values for each of the six maneuvers computed. A table containing the elements before and after each of the maneuvers is also provided so that a selected maneuver can be transferred if so desired.

PMMPAD — The Display Load Module formats all necessary information on the Perigee Adjust Display.

Two Impulse Unit (T/I)

The purpose of the Two-Impulse Unit for the AS-258 Mission is to provide the flight controller with a set of two maneuvers which can be transferred to the Mission Plan Table (MPT) to achieve desired orbit conditions of the maneuvering vehicle.

The unit works in terms of chaser and target vehicle, and in terms of a transfer initiation time and transfer completion time. The first maneuver computed is a chaser transfer initiation maneuver needed to intercept a point in space which is defined in terms of the target vehicle. This intercept point is defined using phase angle and height difference offsets from the target vehicle at the transfer completion time. The second maneuver of the set, which is done at this intercept point, is a co-elliptic maneuver. For example, if the desired offsets are both zero at the transfer completion time, the transfer initiation maneuver will cause the chaser to intercept the target at transfer completion time, and the transfer completion maneuver would be a velocity matching maneuver.

The Two-Impulse Unit has two major areas to accomplish its purpose: (1) the regular Two-Impulse or Multiple Solution option and (2) a Corrective Combination or Digital Option.

The regular Two-Impulse request is initiated via Manual Entry Device (MED) input. A maximum of thirteen solutions may be displayed on the Multiple Solution Display. If more than thirteen solutions are computed, the thirteen which require the least fuel expenditure are displayed. The flight controller may fix the time of the first maneuver and vary the time of the second maneuver through a time increment; he may hold the second maneuver time fixed and vary the first maneuver time; or he may compute only one solution at the input times. An option also exists to compute the time of the first maneuver in terms of an elevation angle to the target. The time of the second maneuver may also be computed in terms of a target vehicle central angle of travel from transfer initiation time to transfer completion time. These two options are normally used for the terminal phase maneuvers, i. e., zero offsets.

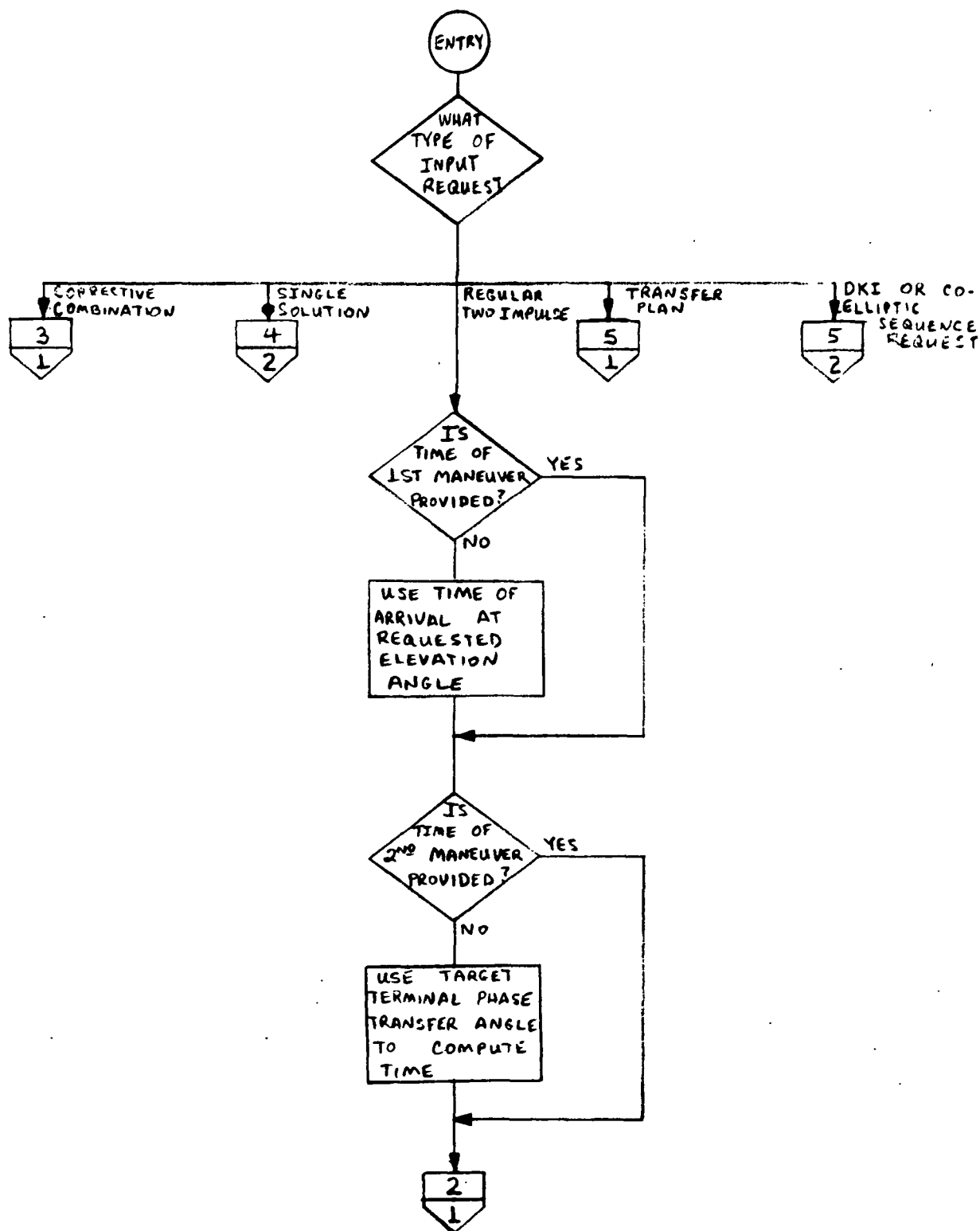
The Corrective Combination area of the Two-Impulse Unit is normally used to obtain satisfactory co-elliptic conditions prior to terminal phase. This may be used when launch dispersion or improper maneuver execution may cause other rendezvous techniques some problems. There are two methods of

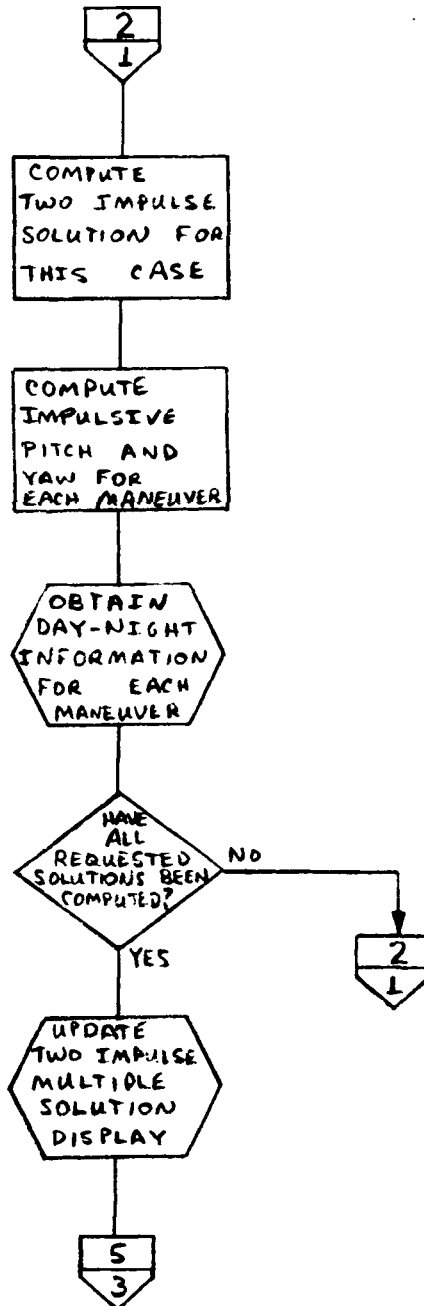
obtaining the optimum corrective combination solution:

- a. The first method requires a range of height differences to be input. A range of times for the transfer completion maneuver are also input, and the phase angle for each time is computed. For each height difference the range of solutions are computed, and the impulsive solution at each height difference that requires the least amount of fuel expenditure is displayed.
- b. The second method involves a series of solutions that also requires a range of height differences at a transfer completion time. For each height difference requested, a range of phase angles at transfer completion time is used, and the optimum solution for each height is displayed. This range of phase angles for each height is computed from a time limit on each side of terminal phase initiation, with an input increment determining the number of solutions attempted. A terminal phase slip time is computed for each solution displayed.

Another display that is driven by this unit is the Single Solution Display. The flight controller may view any one solution from either the multiple or corrective combination table. This display contains components in external ΔV coordinates, daylight-darkness information, and pitch and yaw angles so that the window is along a given line of sight, and contains range, range rate and azimuth approach data at requested times.

Should the flight controller desire to insert the solution into the MPT, a direct transfer option of a particular solution, either multiple or corrective combination, is available by MED request.

Figure 3. Two-Impulse Unit
Flowchart

Figure 3. Two-Impulse Unit
Flowchart

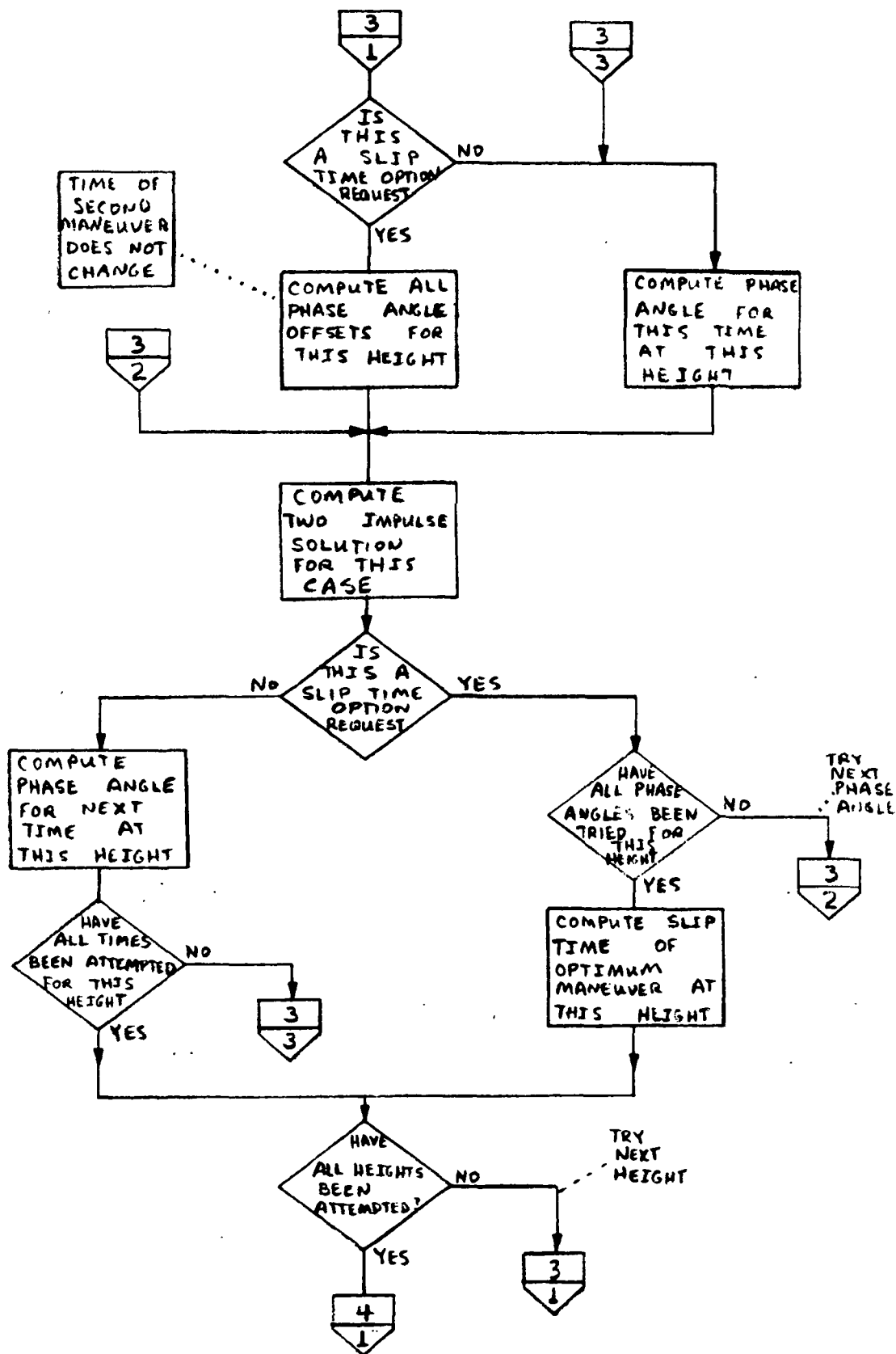


Figure 3. Two-Impulse Unit Flowchart

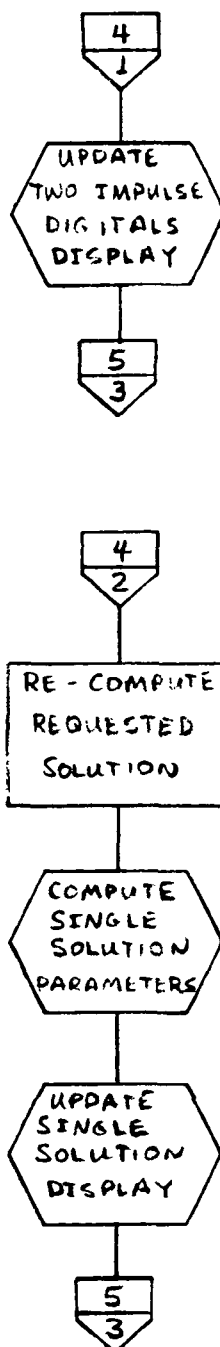


Figure 3. Two-Impulse Unit
Flowchart

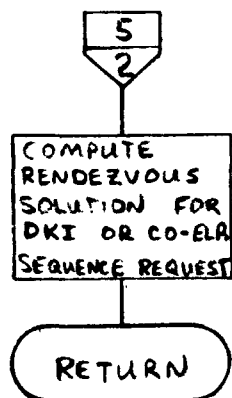
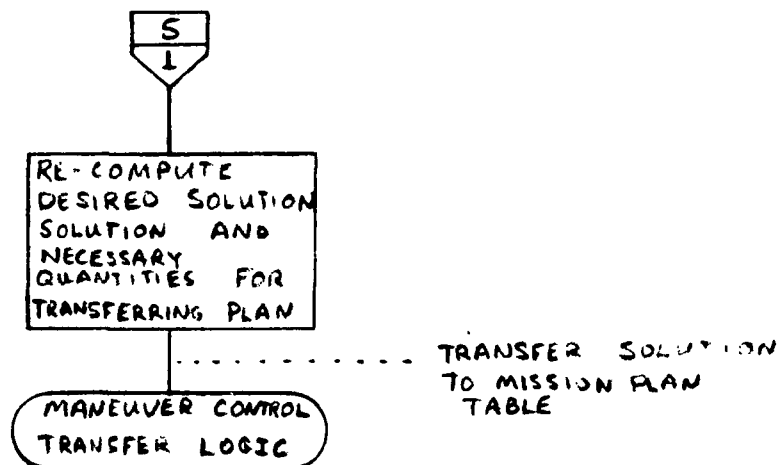


Figure 3. Two-Impulse Unit Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

EORP

DATE 6/19/67

REV

PAGE 18 of 42

Load Module Descriptions — Described below are the load modules contained in the Two-Impulse Unit.

PMSTICN — This load module contains both the control logic and math computations necessary to generate either a multiple solution or a corrective combination set, using Lambert's conic fit technique. It also contains the logic necessary to build an elements table for a transfer of any of the above solutions, and contains the logic to compute the two maneuvers and related elements table to pass to the Docking Initiate (DKI) program to achieve rendezvous from the point in the orbit passed from the DKI (usually TPI point).

PMSTISS — This load module contains the logic and math computation to build the single solution table for display purposes from either the multiple solution table or corrective combination table. This is determined by the Manual Entry Device (MED) input.

PMDTIMP — The Display Load Module formats all necessary information required on the Two-Impulse Multiple Solution Display.

PMDDTVCC — The Display Load Module formats all necessary information required on the Two-Impulse Digitals Display.

PMDTISS — The Display Load Module formats all necessary information required on the Two-Impulse Single Solution Display.

Docking Initiate (DKI)/Co-elliptic Sequence (SPQ) Unit

The basic function of the DKI/SPQ unit is to compute midcourse impulsive maneuvers for up to seven mission plans; the result of these maneuvers is the rendezvous of the CSM and LM spacecraft.

The technique used to determine the necessary times and velocities for the maneuvers is divided into two major areas of logic: (1) the docking initiate (DKI) and (2) the co-elliptic sequence (SPQ). Each method is designed to put the vehicles in a co-elliptic orbit so that a terminal phase maneuver (TPI) and a terminal velocity match maneuver (TPF) may be performed to achieve the actual rendezvous. In addition, the DKI/SPQ provides the data necessary for the transfer of a particular plan to the Mission Plan Table (MPT).

Docking Initiate — The DKI attains a co-elliptic orbit by doing up to four maneuvers: (1) plane change, (2) phase, (3) height, and (4) a co-elliptic maneuver that puts the chaser in a co-elliptic orbit with the target.

The rendezvous point for the DKI is defined for a specified maneuver counter line crossing, with the possible constraints that rendezvous occur in daylight, darkness, or at a longitude crossing. The rendezvous point dictates what phase angle is needed at the co-elliptic maneuver point (NSR) for rendezvous to occur. The NSR maneuver is always done, and always occurs at an apogee end of an orbit.

The three other maneuvers are done to get the spacecraft to NSR with a phase angle, height difference, and wedge angle so that subsequent maneuvers will produce rendezvous. These maneuvers are also defined by a maneuver counter line crossing (N_a).

The plane change maneuver point (NPC) is at a common node between two successive counter line crossings, namely $NPC + .25$, and $NPC - .25$. The height maneuver point (NH) is performed at a perigee counter line value (i.e. $N_a + .5$). The phase maneuver can be done at either a perigee or apogee counter line. The point depends upon the phase angle desired at NSR, and how much phase must change to obtain the desired phase at NSR. To distinguish the two phase possibilities, two modes are defined.

- a. Upper (Mode 1) for phase at apogee (C1), and
- b. Lower (Mode 2) for phase at perigee (CH1).

The flight controller has the option to input the maneuver line number, delete the maneuver, or have them scheduled automatically according to ground rules. They may be for Mode 1 or Mode 2; but if Mode 1 fails to produce rendezvous conditions, the program automatically shifts to a Mode 2 in an attempt to produce rendezvous. If this fails, the program terminates processing.

Upon computing a plan, sufficient information is saved so that the plan may be transferred to the Mission Plan Table. Also the values are saved that will be displayed on the rendezvous evaluation and rendezvous planning displays.

Co-elliptic Sequence — The Co-elliptic Sequence Program computes the incremental velocities required to meet some relative vehicle positioning at the terminal phase initiation (TPI) point. The maneuvers of the iterative sequence are the concentric sequence initiation maneuver (CSI), the co-elliptic maneuver (CDH), and a plane change maneuver (PC) which may or may not be scheduled.

The desired relative positioning at TPI has two options: (1) that the active vehicle have a certain elevation angle to the target, or (2) that a desired differential altitude exist in the co-elliptic orbit configuration.

The time of the CSI maneuver, if scheduled, is input. The desired CSI ΔV which will only be horizontal, may be computed with respect to a desired elevation angle at TPI or a desired ΔH at CDH.

There are three options for the computation of the CDH maneuver:

- a. Arrival at the apsis may be computed by using the Analytic Ephemeris Generator (AEG),
- b. It may be computed after some desired travel angle from CSI, or
- c. It may be scheduled at an input time.

The CDH maneuver may be deleted if the CSI maneuver is also deleted.

The CSI and CDH maneuvers may be applied in-plane or parallel to the target plane; if a plane change is scheduled, only the in-plane option is available. A plane change may be scheduled for either vehicle and before TPI. The threshold time and subsequent common node to perform the PC are input, or the threshold time may be designated as either the CSI or CDH time. TPI time may be input or computed as a time of target arrival at a desired sun-lighting condition or at a desired earth longitude.

An input flag allows the CSI phase angle to be defined in one of three ways:

- a. As the phase angle having smallest absolute value ($-\pi$ to $+\pi$),
- b. As the positive phase angle (0 to 2π) or
- c. As the negative phase angle (0 to -2π).

The constraints checked by this program are in three classifications:

- a. Resulting orbits should have minimum perigees,
- b. A minimum time should exist between the maneuvers, and
- c. The total ΔV expenditure should not exceed the ΔV remaining.

If desired, an option is available to recycle the iterator when a constraint violation occurs by redefining the CSI phase angle as the complement of the original value.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning
DKI Control Supervisor

EORP

DATE 6/19/67

REV

PAGE 22 of 42

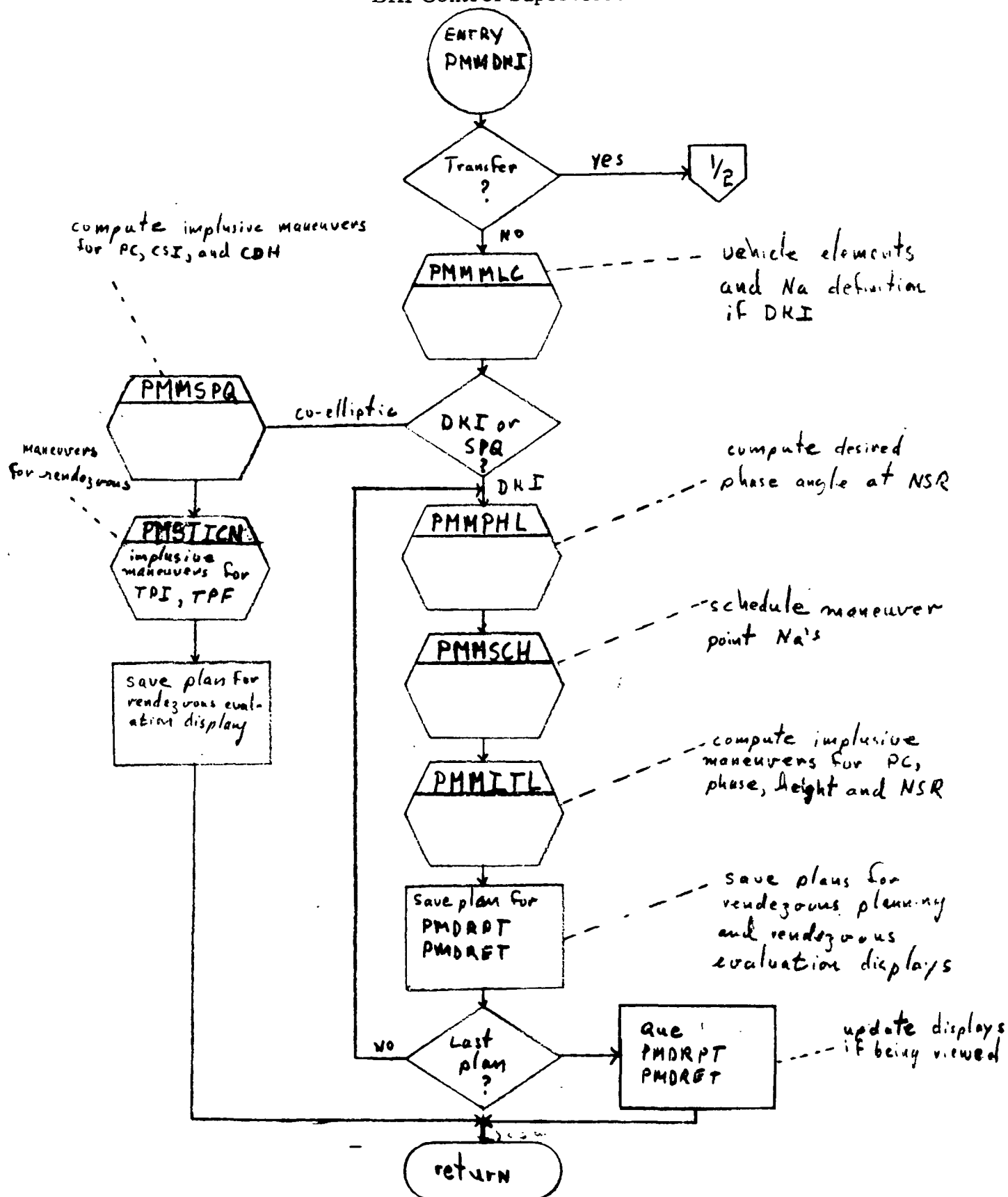


Figure 4. Docking Initiate/Co-elliptic Sequence Unit
Flowchart (Sheet 1 of 2)

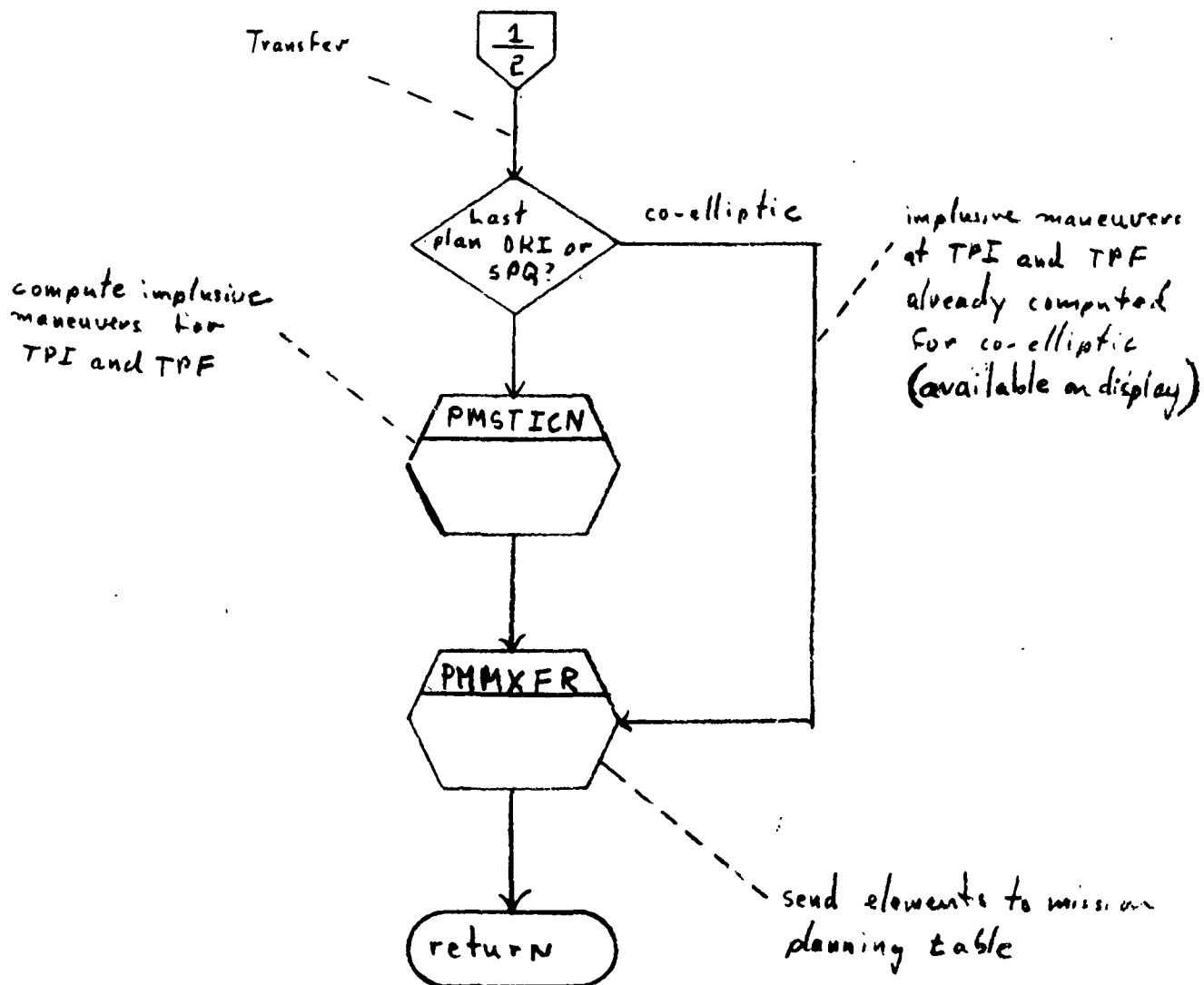


Figure 4. Docking Initiate/Co-elliptic Sequence Unit
Flowchart (Sheet 2 of 2)

Load Module Descriptions — The load modules described below are included in the DKI/SPQ Unit.

PMMPHL — The program computes the phase angle needed at the NSR maneuver point to achieve rendezvous at a specified elapsed time into daylight or darkness or to achieve rendezvous at a specified longitude crossing.

PMMSCH — The program schedules input maneuver points according to ground rules and sequences them in ascending order. If a phase maneuver is scheduled, then the target vehicle liftoff delay time is computed.

PMMITL — The program computes impulsive maneuvers at specified maneuver lines. The maneuvers are computed by an iterative process to converge on a specified phase, height, and co-planar relationships at the circularization (NSR) point.

PMMSPO — The program computes impulsive maneuvers for specified CSI, CDH, PC, and TPI points. See the Unit Writeup for further description.

PMMLC — This program computes and defines the chaser vehicle's maneuver line, updates according to specified option, and returns the chaser and target elements at specified time.

PMDRET — This program displays times, velocities, maneuver N_a 's, height of apogee and perigee, and phase and height difference for the maneuvering vehicle.

PMDRPT — This program displays up to seven DKI plans, indicating plan modes, maneuvers N_a 's, rendezvous m-lines, plus other quantities used to evaluate a plan.

Maneuver Control Unit

The maneuver control unit is responsible for processing various inputs related to maneuver control and trajectory updates. The main function of this unit is to control the generation and maintenance of the CSM and LM Mission Plan Tables. The main types of processing related to this unit are as follows:

- a. Maneuver generation and confirmation
- b. Vector routing
- c. Maneuver control maintenance
- d. Maneuver control displays

Load Module Descriptions (Maneuver Generation and Confirmation) — The Maneuver Generation and Confirmation Programs are responsible for defining maneuvers in the appropriate Mission Plan Table and for integrating through these periods of powered flight. A logical decision making module (PMMXFR) receives and interprets MED information and routes data to the appropriate maneuver processing modules. Initial finite burn computations for the direct input of a future maneuver or for the confirmation of a history maneuver are performed by PMMMCD. Initial finite burn computations for the transfer of impulsive maneuvers as computed by rendezvous planning units are performed by the impulsive maneuver transfer module PMMMPT. The integration through these maneuvers is performed during the process of performing a trajectory update by PMMRKJ. Maneuver execution and postburn data is then computed for the Detailed Maneuver Table display by PMMDMT. The following is a description of each of these load modules.

PMMXFR — The Maneuver Transfer Module solves the ground rules and controls the processing necessary to place the following classes of maneuvers into the MPT.

- a. Contingency Insertion Maneuver
- b. Perigee Adjust Maneuver
- c. Direct Input Maneuver
- d. General Purpose Maneuver
- e. Two-Impulse Plan
- f. DKI Plan
- g. Co-elliptic Sequence Plan

It also solves the ground rules and controls the processing necessary to confirm an executed maneuver which exists in the Mission Plan Table.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

EORP
DATE 6/19/67
REV
PAGE 26 of 42

The PMMMCD module has the capability to compute initial finite burn parameters necessary for the transfer of a Direct Input Maneuver or for the confirmation of an executed maneuver. These parameters are also computed by PMMMCD for the Contingency Insertion Maneuver or for a perigee adjust maneuver as the result of a simulated direct input from PMMXFR.

The parameters defining the Contingency Insertion Maneuver are made available by processing during the CSM launch phase. (No provision is made for a LM contingency insertion burn since this is not within the capabilities of the hardware). Upon manual entry of the MED M01 indicating that this maneuver is desired in the Mission Plan Table, PMMXFR obtains the maneuver parameters along with the launch computed preburn vector, and simulates a direct input to PMMMCD. The transfer of this maneuver automatically deletes all other maneuvers from the Mission Plan Table. The Contingency Insertion Maneuver will be performed using the SPS thruster in the fixed inertial mode. Spacecraft attitudes at burn initiate will be such that the initial thrust is in the orbital plane and in the direction of the earth's horizon. The computations necessary to define the maneuver are performed by PMMMCD. An ephemeris anchored on the preburn vector is then generated.

The maneuvers contained in the Perigee Adjust Maneuver Table are computed assuming a fixed inertial thrust alignment. The alignment is defined by aligning the spacecraft in a "heads down" orientation with the spacecraft X-body axis aligned at the appropriate angle (presently 31.7 degrees) above the in-plane line of sight to the horizon. Maneuvers transferred from this table to the CSM Mission Plan Table will assume the same initial attitude and the same attitude control mode as the Perigee Adjust Processor and will use the SPS thruster. The computations of the initial finite burn parameters are performed by PMMMCD.

A future maneuver may be placed in the Mission Plan Table by manual input of the parameters which describe the maneuver. PMMXFR sets up inputs to the direct input maneuver program, PMMMCD, which performs the necessary computations to describe the maneuver for the powered flight integrator. These inputs include the MED data, a vector prior to the maneuver, and maneuver control indicators necessary for placing the maneuver in the Mission Plan Table.

Any maneuver in the Mission Plan Table that has a begin-burn time prior to present time may be confirmed by manual input of MED M60. PMMXFR sets up the inputs to the maneuver confirmation program, PMMMCD, which performs

the necessary computations to redefine the maneuver so that the resultant trajectory is consistent with the input maneuver execution parameters. These inputs include the MED data, the last anchor vector prior to the maneuver and the control indicators necessary for placing the maneuver in the Mission Plan Table.

The remaining future maneuver classes all require initial finite burn computations by PMMMPT. PMMXFR recognizes the origin of the maneuver(s) and solves appropriate ground rules. Inputs to PMMMPT are the same in all cases and are basically the orbital elements before and after the maneuver at the impulsive maneuver time.

In all cases except maneuver confirmation, after the PMMMCD or PMMMPT processing is completed, the anchor vector is routed to PMSVCT to perform a trajectory update. For maneuver confirmation, after the PMMMCD processing is completed, the anchor vector is routed to Task GTMSCTRL to be moved to present time and is then routed to PMSVCT to perform a trajectory update.

PMMMCD — The Maneuver Confirmation and Direct Input Load Module is responsible for the direct input of a future maneuver and for the confirmation of a history (executed) maneuver. In both cases, the primary concern is the computation of finite burn parameters for the specified maneuver. These parameters are used as inputs to the maneuver integrator. In the case of direct input, a new future maneuver is built based upon the input data and transferred into the appropriate Mission Plan Table. In the case of a confirmation, certain quantities for a specified past maneuver are changed based upon the input data so that this past maneuver and the resultant orbit are consistent with the actual execution of the maneuver by the spacecraft.

A maneuver can be performed with the following thrusters:

- a. CSM or LM RCS $\pm$ X (2 or 4 thrusters firing)
- b. CSM SPS
- c. LM DPS
- d. LM APS

in any of the following attitude control modes:

- a. Fixed inertial
- b. Primary external $\Delta \bar{V}$
- c. Manual holding body orientation invariant

Also an S-IVB blowdown vent maneuver can be performed using the fixed inertial attitude mode.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

EORP

DATE 6/19/67

REV

PAGE 28 of 42

A CSM or LM maneuver can be a single vehicle maneuver, a docking maneuver, an undocking maneuver, or a docked maneuver. (If the CSM and the LM are docked, the maneuver may be placed in either the CSM or LM Mission Plan Table or both). An S-IVB blowdown vent maneuver must be a docked maneuver. The basic inputs for a direct input or confirmation are the following:

- a. Thruster and attitude control mode indicators.
- b. Ignition time of specified thruster.
- c. The ΔV magnitude, or the ΔV along the X-body axis, or the Δt , or the external $\overline{\Delta V}$ vector, or in the case of a confirmation, the maneuver $\overline{\Delta V}$ residuals which have been read by an astronaut.
- d. The pitch, yaw, and roll in the LVLH (Local Vertical, Local Horizontal) System or in the IMU (Inertial Measuring Unit) System. (The attitudes do not have to be specified if the mode of the maneuver is primary external $\overline{\Delta V}$.).
- e. The Δt of ullage and number of RCS thrusters used for ullage (specified only for SPS, DPS, or APS thruster).
- f. A reference matrix indicator.
- g. Vehicle heads-up/heads-down indicator.
- h. The Δt of 10% thrust level for DPS thruster.
- i. The trim angle indicator for SPS or DPS thruster.

In both cases, control is given to PMMMCD from PMMXFR. When processing has been completed, control is returned to PMMXFR. In the case of a direct input, control is then relinquished to PMSVCT, and a trajectory update is performed. In the case of a confirmation, a queue is sent to GMSCTRL (Task GTMCTRL), which moves the history vector up to present time and sends a queue to PMSVCT. A trajectory update is then performed.

PMMPPT — The Mission Plan Table Impulsive Maneuver Load Module is responsible for the transfer of an impulsive maneuver into the appropriate Mission Plan Table. The maneuver is generated by one of the following sources:

- a. General Purpose Maneuver Program
- b. Two-Impulse Program
- c. DKI Program
- d. Co-elliptic Sequence Program

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

EORP
DATE 6/19/67
REV
PAGE 29 of 42

This program takes osculating orbital elements before and after an impulsive maneuver and computes the finite burn parameters necessary as inputs to the powered flight numerical integration program. A maneuver can be performed with the following thrusters:

- a. CSM or LM RCS $\pm X$ (2 or 4 thrusters firing)
- b. CSM SPS
- c. LM DPS
- d. LM APS

in any of the following attitudes control modes:

- a. Fixed inertial
- b. Primary external $\overline{\Delta V}$
- c. Manual holding body orientation invariant.

The basic input parameters are the following:

- a. Osculating orbital elements before and after the impulsive maneuver with both sets of elements given at the impulsive maneuver time.
- b. Thruster and attitude control mode indicators.
- c. Δt of ullage and number of ullage thrusters to be used (applicable only for SPS, DPS, or APS thrusters).
- d. Vehicle heads up/heads down indicator.
- e. Δt of 10% thrust level for DPS thruster.
- f. Thruster trim angle indicator.

Maneuvers transferred to the Mission Plan Table by this program may be single vehicle or docked maneuvers; however, no change in the vehicle configuration is allowed.

PMMRKJ — The CSM/LM Powered Flight Integrator Load Module is a program designed to numerically integrate through periods of powered flight and short spans of free flight. The types of maneuvers available are (1) inertial, (2) manual holding body attitude invariant in inertial space, and (3) primary external $\overline{\Delta V}$.

These maneuvers can be performed using the following thrusters:

- a. CSM RCS $\pm X$ (2 or 4 thrusters)
- b. LM RCS $\pm X$ (2 of 4 thrusters)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

EORP
DATE 6/19/67
REV
PAGE 30 of 42

- c. CSM SPS
- d. LM APS
- e. LM DPS
- f. S-IVB blowdown vent

For maneuvers using multiphase thrusters (SPS, APS, DPS), two or four RCS thrusters for ullage may be specified. A fourth order Runge-Kutta, Nyström technique is used to generate the following:

- a. An ephemeris table of position and velocity vectors.
- b. An auxiliary output table containing critical values at burn initiate and burnout.
- c. A table of weights.

PMMDMT — The Detailed Maneuver Table Load Module is responsible for (1) maintenance of the areas and weights after the maneuver; (2) for storing integrator output in the appropriate Mission Plan Table; (3) for the computation of quantities based on the position and velocity vector at ignition; (4) for post-burn calculations which describe the post-maneuver orbit; and (5) for post-maneuver rendezvous related target-chaser quantities at maneuver burnout (height differential, central angle difference and difference in orbit planes). The results are stored in the appropriate Mission Plan Table and are available for display. A restart vector is printed on-line at the end of a trajectory update from the burnout vectors of the first future maneuver.

Load Module Descriptions (Vector Routing) — The primary functions of the Vector Routing processing is to select vectors from the current vehicle ephemeris for the purpose of performing an ephemeris update and to initiate trajectory updates as the result of the selection of new vehicle anchor vectors.

PMSVCT — The Vector Routing Load Module will be entered as the result of various maneuver control functions which result in a change in the Mission Plan Table data that affects the generation of the current CSM or LM ephemeris.

These functions include the following:

- a. The insertion, deletion, or change of a future MPT maneuver.
- b. The change of the current orbit density multiplier (K-factor).
- c. The change of a vehicle weight or cross-sectional area.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

EORP
DATE 6/19/67
REV
PAGE 31 of 42

In addition, PMSVCT will be entered to perform an "ephemeris extension" when the length of the future ephemeris fails to meet the requirement as specified in TR165. Since these functions do not affect the anchor vector for the ephemeris, a "present position" vector based upon the current anchor vector is selected from the vehicle ephemeris. This "present position" vector will be the latter of (1) the burnout vector of the last executed maneuver, or (2) the first free-flight three-minute vector prior to present time.

New anchor vectors for an ephemeris update may originate from various sources. CSM and LM launch programs compute and route insertion vectors when the launch-to-orbit transition is made. Rendezvous planning programs compute and route a nominal insertion vector during LM pre-launch. CSM and LM orbit-correction programs provide anchor vectors based on high and low-speed radar collection and high and low-speed telemetry. An anchor vector may be re-routed for trajectory update as a result of a maneuver confirmation. Restart vectors for the CSM and LM may be manually entered for the purpose of a type "C" restart. A manual insertion of a nominal insertion vector may be entered during prelaunch. Prior to being routed to PMSVCT, each of these vectors, except nominal insertion vectors, will be integrated to present time reflecting all maneuvers in that time span. This present position vector will then be routed to EMSTRAJ to complete the trajectory update.

Load Module Descriptions (Maneuver Control Maintenance) — The Maneuver Control Maintenance Programs are responsible for the general upkeep of the CSM and LM Mission Plan Tables. They set indicators and change data as the result of an internal real time queue or a MED request. These modules are also required to logically solve given sets of ground rules as a part of their task.

PMMFUD — The Freeze/Unfreeze/Delete Load Module solves the ground rules associated with freezing, unfreezing, or deleting maneuvers in the CSM or LM Mission Plan Table. It identifies the maneuvers as being frozen or unfrozen and maintains the Mission Plan Table when deleting executed or non-executed maneuvers. The deletion of executed maneuvers will not cause a trajectory update. The deletion of a non-executed maneuver will result in a trajectory update being performed as a result of relinquishing control to PMSVCT.

PMSEXE — The Maneuver Execution Load Module performs various maneuver control functions related to the execution of maneuvers in the CSM or LM Mission Plan Table. When present time reaches the begin time of a maneuver, a queue is passed to PMSEXE to print on-line the burn-out vector for the maneuver along with the corresponding weight of the vehicle. Flags are set in the MPT to indicate that the maneuver has been executed. A queue is sent to the Supervisor providing it with the time of the next maneuver execution queue. The MPT and DMT displays are updated as a result of releasing control to GTISPLAY, a display task.

PMMWTC — The Weight Change Module performs the function of changing the CSM, LM or S-IVB vehicle area and weight in the CSM or LM MPT. An input of a new vehicle area or weight and time to be applied will result in those values being stored as the single vehicle area and weight at the end of the last maneuver prior to the given time. Those values will then be used to compute a new vehicle configuration area and weight at the end of the maneuver. This program also accepts manual inputs of orbit density multiplier (K-factor) for the CSM or LM and accepts the manual input of the initial vehicle configuration for the CSM or LM ephemeris. Any entry to this program will result in a queue to PMSVCT to perform a trajectory update.

Load Module Descriptions (Maneuver Control Displays) — The Maneuver Control Display Programs present summary and detail-maneuver related quantities and post-maneuver quantities to the flight controllers. These displays are automatically updated as a result of a maneuver execution, a history maneuver deletion, a maneuver being frozen or unfrozen, or a trajectory update.

PMDMPT — The Mission Plan Table Display Module formats maneuver-related quantities from the CSM and LM Mission Plan Tables and causes the information to be put to the D/TV system.

PMDDMT — The Detailed Maneuver Table Display Module provides detailed maneuver information for any specified maneuver in either the CSM or LM Mission Plan Table. The basic quantities displayed are finite burn parameters, post-maneuver trajectory parameters, vehicle weights before and after the maneuver, target parameters (applicable to guided maneuvers only), and some rendezvous parameters.

All displayed quantities except the vehicle IMU attitude quantities are picked up from the appropriate MPT block by PMDDMT. The display request MED specifies the required REFSMAT for IMU attitude computation. If the desired (DES) REFSMAT is designated, an indicator must be input specifying whether a heads up or heads down platform orientation is desired. The vehicle IMU attitudes for the specified REFSMAT are computed by PMDDMT.

Maneuver Control Functional Flow

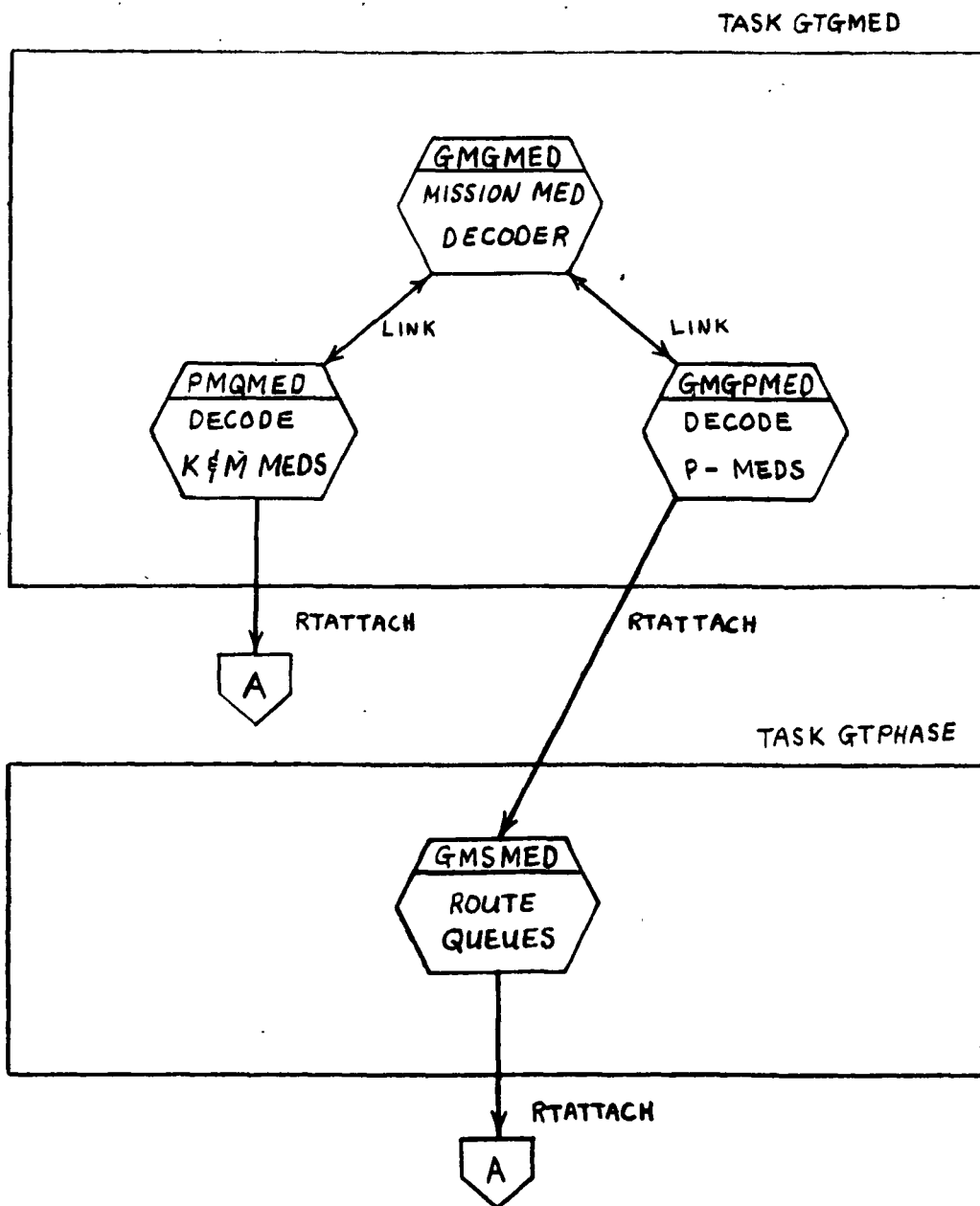
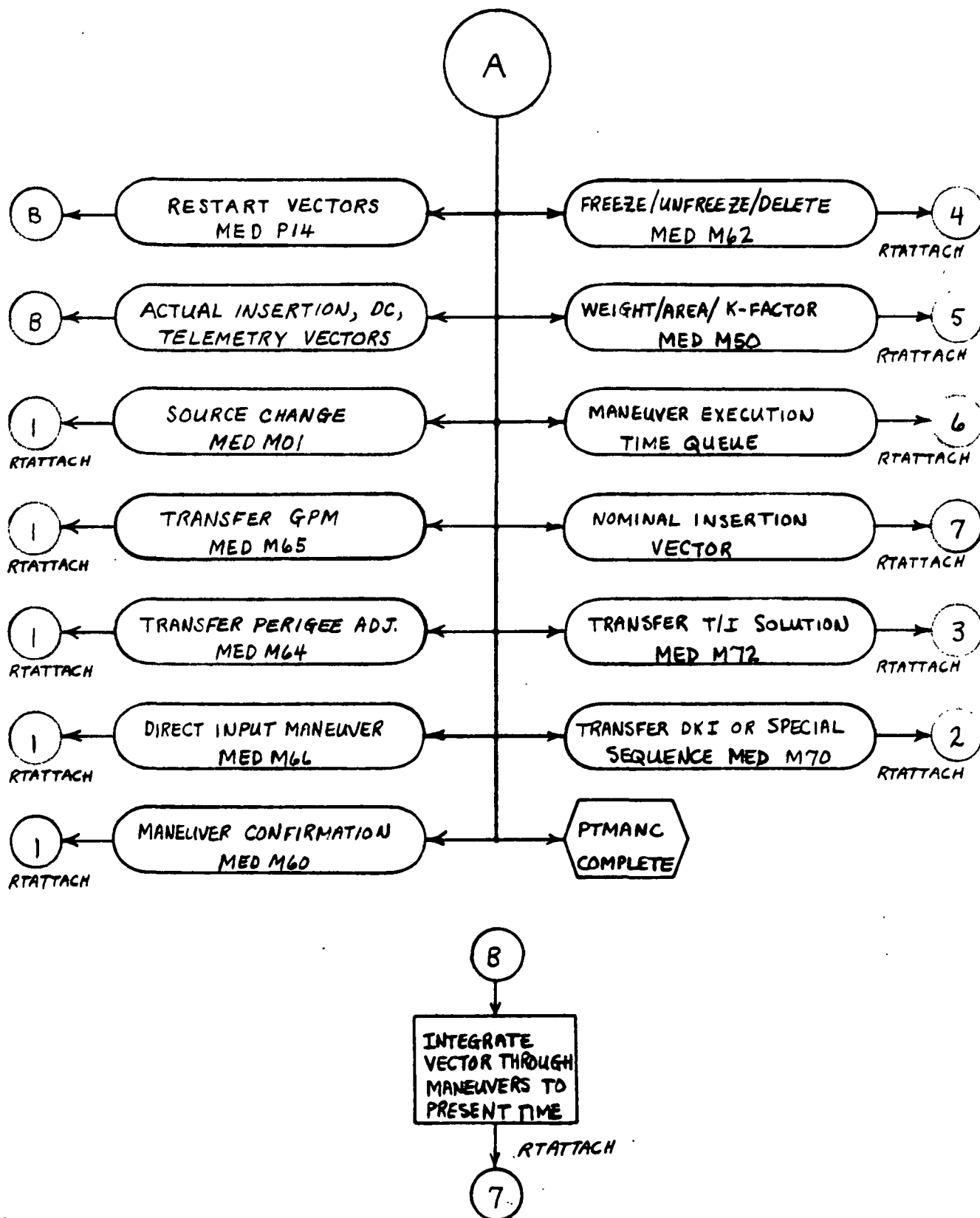


Figure 6. Maneuver Control Unit
Flowchart

Task PTMANC

Figure 7. Maneuver Control Unit
Flowchart

Task PTMANC

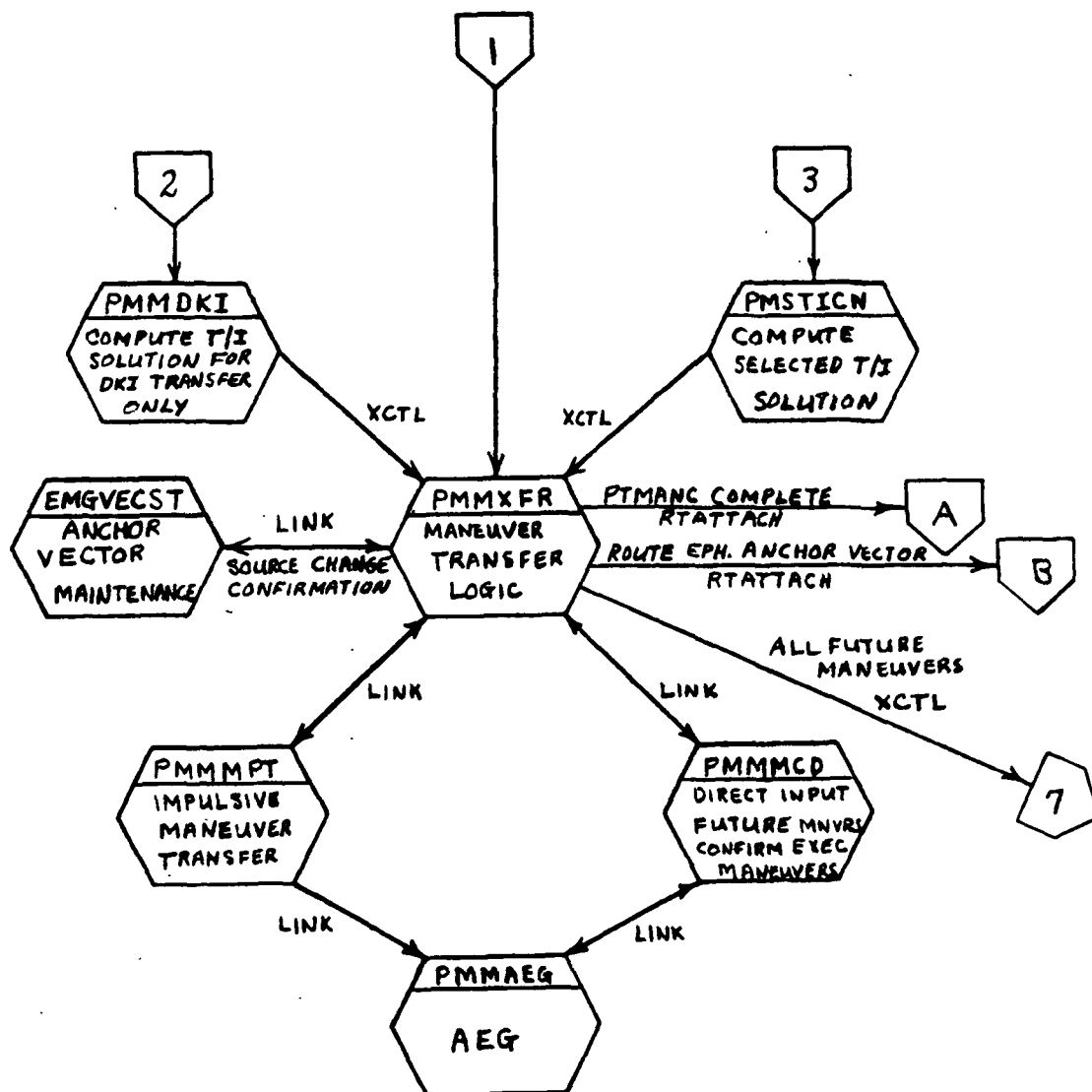


Figure 8. Maneuver Control Unit Flowchart

Task PTMANC

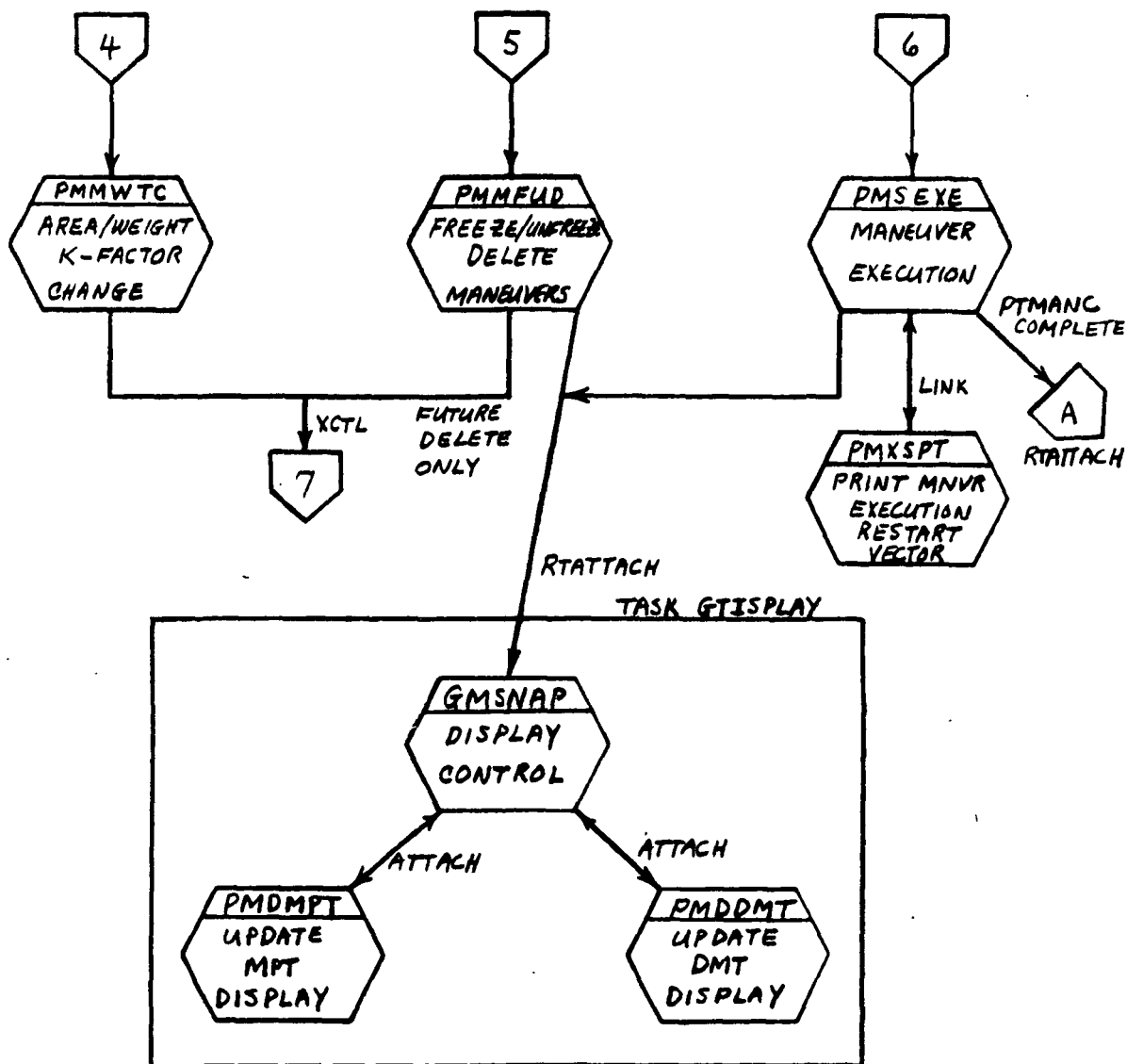
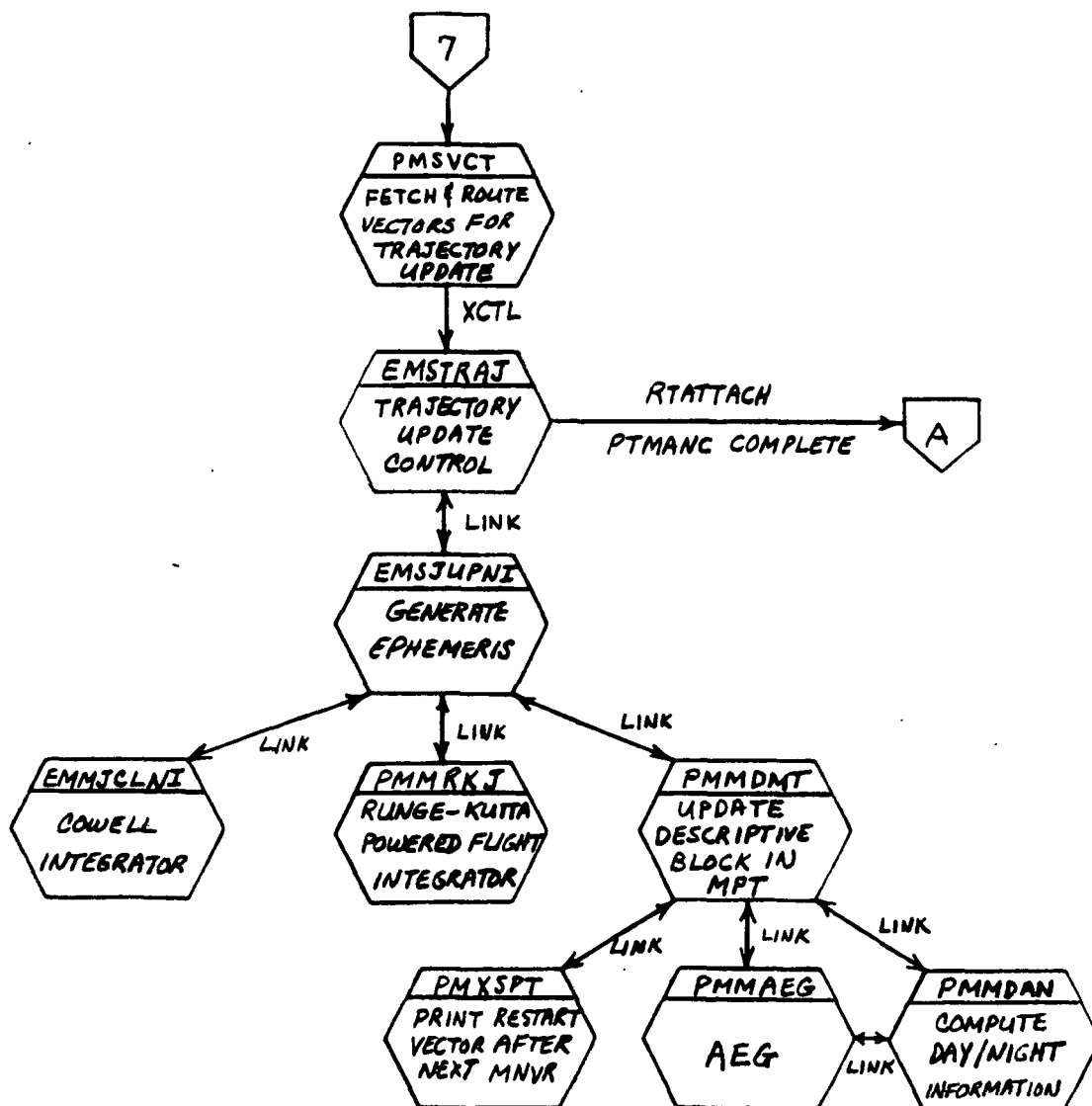


Figure 9. Maneuver Control Unit Flowchart

Task PTMANC

Figure 10. Maneuver Control Unit
Flowchart

MISSION PLANNING UTILITY PROGRAMS

Load Module Descriptions — The Mission Planning Utility Programs include the load modules described below.

PMMAEG — The Earth Satellite Analytic Ephemeris Generator Load Module (AEG) can be separated into four logical units. The units are (1) initialization of a set of Keplerian elements, (2) updating of the elements, (3) initialization and updating of elements, and (4) processing of multiple blocks. Each of these logical units are discussed in more detail below.

- a. Initialization of Elements — To initialize elements the AEG will compute the osculating and mean elements, the rates, argument of latitude, radius magnitude, and true anomaly based on a set of input osculating elements at a given time.
- b. Updating of Elements — To update elements, the AEG must have a set of the initialized elements described above. The AEG can update a set of these elements by six different methods. The methods are (1) time, (2) mean anomaly, (3) argument of latitude, (4) maneuver counter line, (5) time match, and (6) position match. Each of these methods are discussed in more detail below.
 - (1) Time — Update the initialize input elements to an input time.
 - (2) Mean Anomaly — Update the initialize input elements to the input mean anomaly in the orbit ΔN orbits from the input elements.
 - (3) Argument of Latitude — Update to the input argument of latitude in the orbit ΔN orbits from the input elements.
 - (4) Maneuver Counter Line — Update to a maneuver counter line which is ΔN counter lines from the reference counter line.
 - (5) Time Match — Update the input elements to the time associated with output elements of the previous block processed. The AEG also computes the phase angle for output.
 - (6) Position Match — The AEG can update to position match in two different ways. They are (1) update the input elements to position match with the previous block processed by going backward, and (2) update the input elements to position match with the previous block processed by going forward if the phase angle at time match is greater than 180° , or by going backward

if the phase angle at time match is less than 180° . The AEG also returns the time lag, height difference, and phase angle as output at time match. When updating the input elements, the AEG has the option to update with or without drag depending upon the input.

- c. Initialization and Updating of Elements — Depending upon the input, the AEG can take a set of input osculating elements and initialize the elements (described above in A) and then update the initialize elements (described in B above) with one entry.
- d. Processing of Multiple Blocks — The AEG can process multiple number of input blocks. When updating to a time or position match, multiple block processing must be used as a previous block must have been processed to use this update option.

PMSVEC — Given either one or two vector times, and a vehicle code for the first or only time, the Rendezvous Planning Vector Fetch Load Module fetches a vector from the corresponding vehicle ephemeris, if available, thus satisfying all criteria concerning maneuvers in the Mission Plan Table (MPT). The logic then converts the vector to Keplerian elements, and passes these elements, the corresponding time, and the area, weight, and K-factor at that time to the requesting program in a format suitable for use with the Analytic Ephemeris Generator (AEG) Program. If two vector times are requested, two AEG blocks are passed, one for each vehicle involved in the mission. The station ID used to build the current ephemeris is also made available to the user.

At the time of the writing, the General Purpose Maneuver (GPM), Perigee Adjust, Two-Impulse, and Maneuver Line Definition programs were using utility program.

PMMDAN — Given a set of satellite elements (AEG format), the Day/Night Load Module Program will utilize the AEG to determine the time and nature of the next one or two environmental changes. The routine will use a sun vector at the input time. The sun's rays are considered to be parallel, and no attention is given to reflection, diffusion or semidiameter of parallax. For the purpose of this program these factors do not introduce significant error.

PMXSPT — The Mission Planning Print Module prints all on-line messages related to Mission Planning.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

EORP
DATE 6/19/67
REV
PAGE 41 of 42

PMQMED — The Mission Planning MED Decoder Load Module checks the validity of all MED's related to Mission Planning and sets up the data to be routed to the appropriate task.

Subroutine Descriptions — The subroutines described below are included in the Mission Planning Utility Programs.

PIEXDV — The Primary External $\Delta\bar{V}$ Coordinate Transformation Subroutine basically performs a special coordinate transformation on an input velocity increment vector $\Delta\bar{V}$. If the input $\Delta\bar{V}$ is in the external $\Delta\bar{V}$ uplink coordinate system the transformed $\Delta\bar{V}$ is in the RTCC inertial coordinate system. If the input $\Delta\bar{V}$ is in the RTCC inertial coordinate system, the transformed $\Delta\bar{V}$ is in the external $\Delta\bar{V}$ uplink coordinate system. The program is consistent with the on-board capabilities for the first earth-orbit rendezvous mission as presented in the MIT CSM and LM R547 documents dated October, 1966. The subroutine is designed for the CSM GNCS and LM PGNCS external $\Delta\bar{V}$ guidance and cannot be used for LM AGS external $\Delta\bar{V}$ computations.

GIMIKC — The Inertial to Keplerian Element Conversion Subroutine converts position and velocity vectors in the RTCC inertial coordinate system to Keplerian orbital elements. The program can be used for elliptic and circular orbits and cannot be used for parabolic or hyperbolic orbits.

GIMKIC — The Keplerian to Inertial Element Conversion Subroutine converts Keplerian orbital elements to position and velocity vectors in the RTCC inertial coordinate system. The program can be used for elliptic and circular orbits and cannot be used for parabolic or hyperbolic orbits.

PIDREF — The Orbit Desired REFSMAT Computation Subroutine computes the orbit phase desired IMU stable member alignment REFSMAT for the CSM or LM. There is an option for either a heads up or a heads down IMU stable member orientation.

PIDVRS — The Maneuver Confirmation $\Delta\bar{V}$ Residual Subroutine is designed to allow confirmation of an executed CSM or LM maneuver when the maneuver $\Delta\bar{V}$ residuals in the vehicle RCS control axes are known. The inputs are the $\Delta\bar{V}$ residuals, the IMU attitude angles at the time of the reading of the residuals, the applicable REFSMAT, the original maneuver $\Delta\bar{V}$, and a vehicle indicator. The actual maneuver velocity increments $\Delta\bar{V}_A$ is computed and can be used by other programs to confirm the executed maneuver.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

EORP

DATE 6/19/67

REV

PAGE 42 of 42

GIMGBL — NASA will provide tables for the Thruster Gimbal Angle Computation Subroutine which give the vehicle center of gravity location as a function of vehicle weight for each of the following vehicles:

- a. CSM
- b. LM with Ascent Propulsion System attached
- c. LM with Ascent and Descent Propulsion Systems attached

The capability will exist to update each of these tables in real time. The Thruster Gimbal Angle Computation Subroutine will use these tables and the input vehicle configuration code, docking angle, and thruster indicator to compute the center of gravity for the specified configuration and the thruster gimbal angles necessary to align the thrust vector through the center of gravity.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

1. PMMCMU
DATE 11/22/66 (205)
REV
PAGE 1 of 8

1. PMMCMU

- a. Symbol — PMMCMU
- b. Program Name. — Current Matrix Update
- c. Programming Group — Mission Planning
- d. Programmer (s) — R. J. Taylor
- e. Language — Assembler
- f. Data Tables Used — PZMPTCSM
- g. MED's Used — None
- h. Writeup Completion Date — 11/23/66
- i. Parameter List —
- j. System Parameters — Table PMMCMU-2.
- k. Flowchart — Figure PMMCMU-1.

FUNCTION

When definition is desired of new unit vectors for the platform axes, this load module places the new matrix in the Mission Plan Table for all maneuvers after the time of the new definition. New values for pitch, yaw, roll, and the velocity vector are computed for those maneuvers.

USAGE

- a. Calling Sequence — LINK EP = PMMCMU, PARAM = (VEH)
- b. Inputs — PZMPTCSM, Vehicle Code
- c. Outputs and Exit — Updated PZMPTCSM, return
- d. Other Programs Used — EIGMTX, GLMDPD, GLMXPD, GLQATN, PMDDMT, PMXSPT
- e. Data Restrictions and Limits — None
- f. Error — If no matrix is returned from EIGMTX, an on-line message "No matrix available" is printed and the program exits.

METHOD

The subroutine EIGMTX is used to obtain the current matrix, time of the matrix, and its ID. All maneuvers at or past the matrix time are updated by storing the new definition and its ID, and by using the values of pitch, yaw, roll, and the velocity vector to find the new numbers for those quantities.

Approval	Approval
<i>R. Taylor</i>	<i>J. Ste...</i>

- $\hat{X}_{p_1}$ = unit vector in direction of platform X-axis
 $\hat{Y}_{p_1}$ = unit vector in direction of platform Y-axis
 $\hat{Z}_{p_1}$ = unit vector in direction of platform Z-axis
 V_{g_1} = velocity vector in platform coordinate system
 P_1 = pitch
 Y_1 = yaw
 R_1 = roll

The above values are from a maneuver block in the Mission Plan Table.

- $\hat{X}_p$ = unit vector in direction of X-axis
 $\hat{Y}_p$ = unit vector in direction of Y-axis
 $\hat{Z}_p$ = unit vector in direction of Z-axis
- } new platform

$$\begin{bmatrix} \hat{X}_n \\ \hat{Y}_n \\ \hat{Z}_n \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos R_1 & \sin R_1 \\ 0 & -\sin R_1 & \cos R_1 \end{bmatrix} \begin{bmatrix} \cos Y_1 & \sin Y_1 & 0 \\ -\sin Y_1 & \cos Y_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos P_1 & 0 & -\sin P_1 \\ 0 & 1 & 0 \\ \sin P_1 & 0 & \cos P_1 \end{bmatrix} \begin{bmatrix} \hat{X}_{p_1} \\ \hat{Y}_{p_1} \\ \hat{Z}_{p_1} \end{bmatrix}$$

$$Y = \sin^{-1} \left(\hat{X}_n \cdot \hat{Y}_p \right) \quad 0 \leq Y < \pi/2, \quad 3\pi/2 < Y < 2\pi$$

If $Y \neq \pi/2$ or $3\pi/2$:

$$P = \tan^{-1} \left[\frac{-\hat{X}_n \cdot \hat{Z}_p}{\hat{X}_n \cdot \hat{X}_p} \right]$$

$$R = \tan^{-1} \left[\frac{-\hat{Z}_n \cdot \hat{Y}_p}{\hat{Y}_n \cdot \hat{Y}_p} \right]$$

Otherwise:

$$P = \tan^{-1} \left[\frac{\hat{Z}_p \times \hat{Z}_n \cdot \hat{Y}_p}{\hat{Z}_p \cdot \hat{Z}_n} \right]$$

$$R = 0.0$$

Then:

$$\overline{\Delta V} = V_{g_{x_1}} \hat{X}_{p_1} + V_{g_{y_1}} \hat{Y}_{p_1} + V_{g_{z_1}} \hat{Z}_{p_1}$$

$$V_{g_x} = \overline{\Delta V} \cdot \hat{X}_p$$

$$V_{g_y} = \overline{\Delta V} \cdot \hat{Y}_p$$

$$V_{g_z} = \overline{\Delta V} \cdot \hat{Z}_p$$

The values for P, Y, R, V_{g_x} , V_{g_y} , and V_{g_z} are placed in the maneuver block, and the same sequence is used for the remaining maneuvers.

Table PMMCMU - 2. System Parameters

[illegible]

*Type
H — Fixed-point half word
F — Fixed-point full word
E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

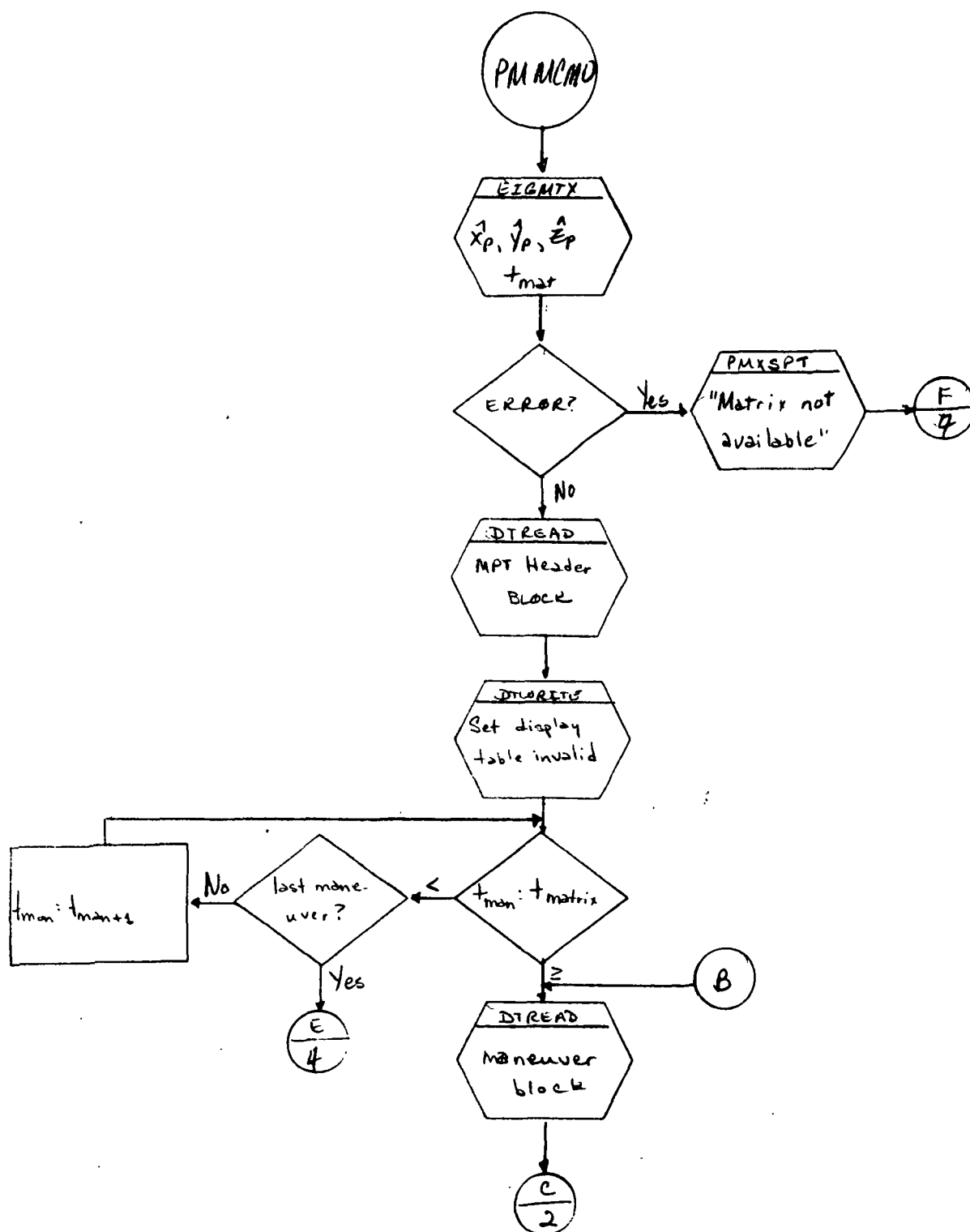


Figure PMMCMU - 1.

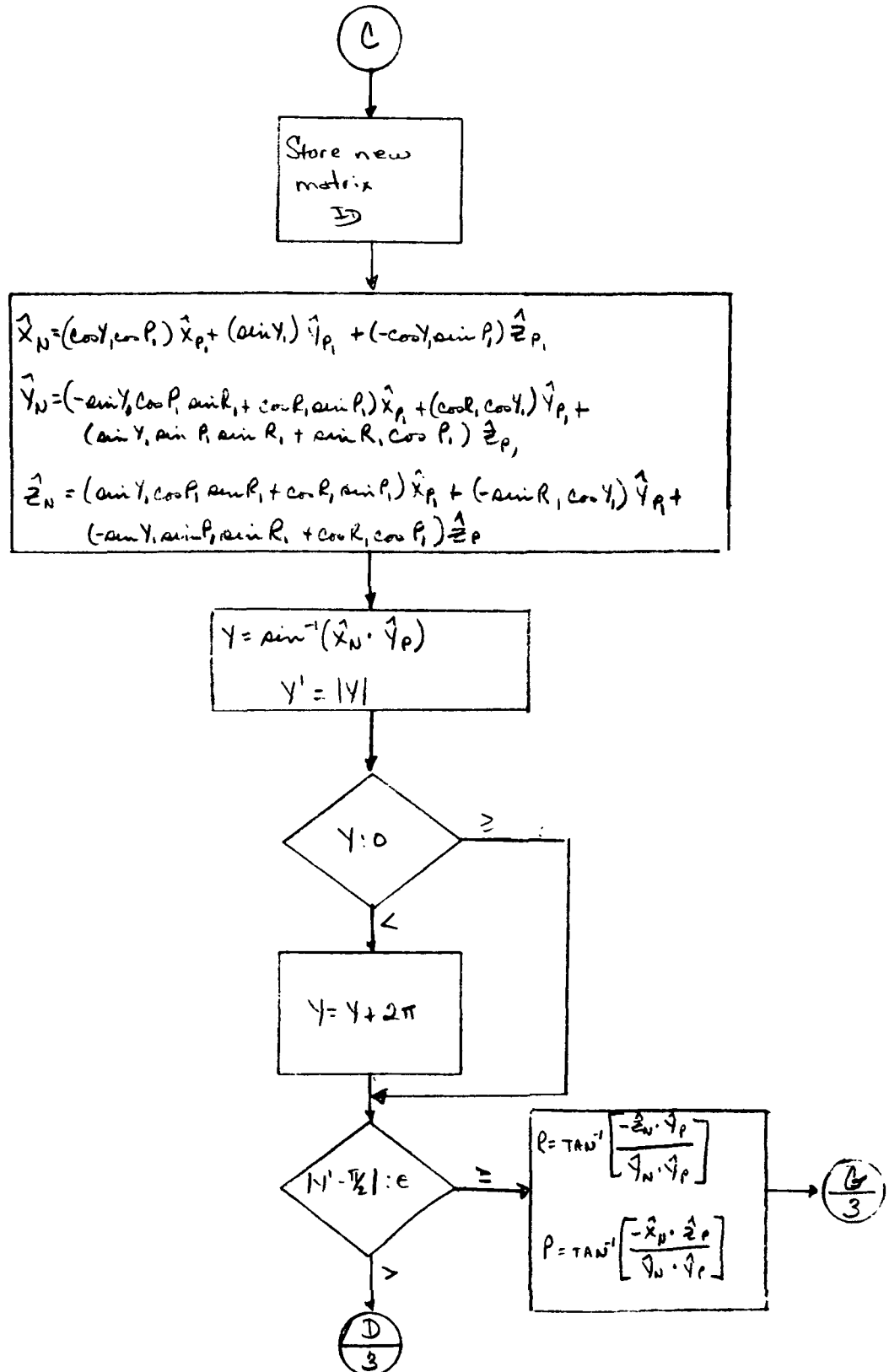


Figure PMMCMU - 1.

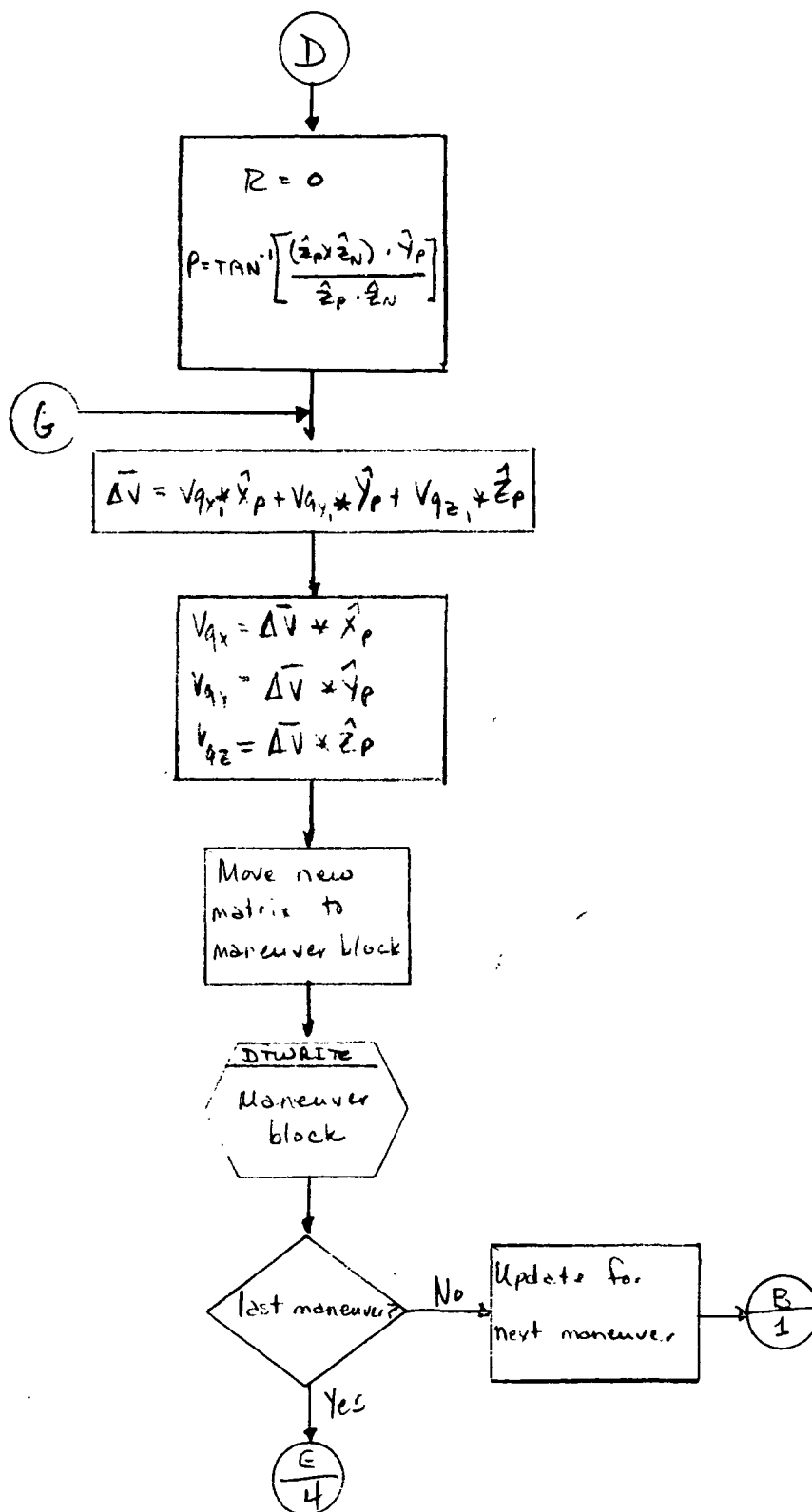


Figure PMMCMU - 1.

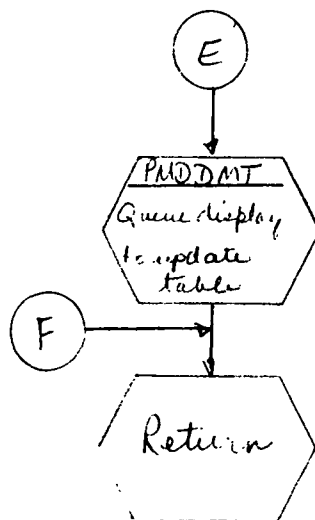


Figure PMMCMU - 1.

EORP LOAD MODULE CROSS-REFERENCE

Listed below are the load modules that are included in the Earth Orbital Rendezvous Package (EORP). Those load modules followed by a mission number in parentheses have been documented for that mission and that version applies to EORP. Writeups for these load modules will not be republished. All other load modules have been published for EORP.

ProgramBook

BMDCSM (206)	Trajectory Determination
BMDHAP	
BMDLSDS (205)	
BMDLSSU	
BMDMVMP	
BMDRLS	
BMDVEC	
BMDVPS	
BMGCHAIN (206)	
BMGHSPR	
BMGPRIME (205)	
BMGSTA (206)	
BMMCNV (206)	
BMMELL (206)	
BMMFED (206)	
BMMIDG (205)	
BMMLPH (206)	
BMMMAN (206)	
BMMMDP	
BMMNFM (205)	
BMMPRG (205)	
BMMSEL (205)	
BMMTPE	
BMPRAW (205)	
BMQCBAND	
BMQDCMED (205)	
BMQHIGHS	
BMQHIGHI	
BMQLSINP (206)	
BMQLSMED (206)	
BMSANM (206)	
BMSHSDS	
BMSINIT	
BMSINT (206)	
BMSMED	

EORP LOAD MODULE CROSS-REFERENCE
(Continued)ProgramBook

BMSPBI
BMSPCL
BMSPDR
BMSPMI
BMSPNM (206)
BMSPPP (205)
BMSRNM (206)
BMSVEC
BMSVPS
BMTEST
CMDAXTDV
CMDCMNAV
CMDERMEM
CMDGMTLU
CMDLIFTF
CMDLXTDV
CMDPTBKU
CMDPTPCP
CMDPTPRI
CMDRFMAT
CMDRXTDV
CMDSEQIN (205)
CMDSVGEN
CMDSVNAV
CMDSWSEL (205)
CMDSWSLT (205)
CMDTMBSE (205)
CMDTMEIN
CMMAXTDV
CMMCNAV
CMMERMEM
CMMGMTLU
CMMLIFTF
CMMLMNAV
CMMLXTDV
CMMPTBKU
CMMPTPRI
CMMRFMAT
CMMRXTDV
CMMSEQIN (205)
CMMSKENT

Trajectory Determination



Digital Command System



EORP LOAD MODULE CROSS-REFERENCE
(Continued)ProgramBook

CMMSVGEN
CMMSWSEL (205)
CMMTMBSE (205)
CMMTMEIN
CMRMEDIN
CMSDCSXT
CMSDTVLU
CMSEQNUM
CMSPBIDC (205)
CMSTATUS (205)
CMSWITCH
EMDASCND
EMDCHECK
EMDEXSIT
EMDGAGSN
EMDGLMST
EMDGSSTB
EMDGSUPP
EMDGTDGL
EMDPSAD (206)
EMDSSMMD
EMDSTAC
EMDXDDRR (206)
EMGPRINT
EMGTVMED (206)
EMGVECST
EMGXVTRA
EMMASCND
EMMDYNEL (206)
EMMDYNMC
EMMEACC (206)
EMMEXSIT
EMMGLCVP
EMMGLSTM
EMMGLSXT
EMMGSSMP
EMMGSSXT
EMMJCLNI
EMMJEWST (206)
EMMRMD
EMMVEN (206)

Digital Command System

Orbit Trajectory Computations

EORP LOAD MODULE CROSS-REFERENCE
(Continued)

<u>Program</u>	<u>Book</u>
EMMVRTTY	Orbit Trajectory Computations
EMRTRAJ	
EMSGSUPP	
EMSJMSNI	
EMSJUPNI	
EMSLSUPP	
EMSTAGEN (206)	
EMSTAG2 (206)	
EMSTRAJ	
EMXJMATX	
GMDTVGID (206)	
GMGCARD	
GMGDMED	
GMGMED	
GMGPMED	
GMLCSLQD	
GMLGSTCH	
GMLRESRT	
GMMCBND	Orbit Trajectory Computations
GMMDUD	Orbit Trajectory Computations
GMMGRP	Reentry Computations
GMMSBND	Orbit Trajectory Computations
GMMYNI	Reentry Computations
GMPREINT	Systems Integration
GMSCTRL	Orbit Trajectory Computations
GMSDELQG	Systems Integration
GMSDETPH	Systems Integration
GMSDISP	Systems Integration
GMSFILL	Systems Integration
GMSINT	Systems Integration
GMSMDDD	Systems Integration
GMSMED	Systems Integration
GMSNAP	Orbit Trajectory Computations
GMSPHASE	Systems Integration
GMSPRINT	Systems Integration
GMSREMED	Systems Integration
GMSSACQ	Orbit Trajectory Computations
GMSTIME (205)	Orbit Trajectory Computations
GMSWAQ	Systems Integration

EORP LOAD MODULE CROSS-REFERENCE
(Continued)ProgramBook

GMSWITCH	Systems Integration
GMTLMTRJ	Orbit Trajectory Computations
GMXERRER	Orbit Trajectory Computations
LMGHINIT	Launch/High-Speed Abort
LMGINITZ	
LMMAPPDS (206)	
LMMRKN1	
LMMRKN2	
LMQAGCTM	
LMQIPRRD (206)	
LMQIPSMO (206)	
LMQLGCTM	
LMQUSBLR	
LMRKN1	
LMRMANU1	
LMRMANU2	
LMRMANU3	
LMRMANU4	
LMRMANU5	
LMRMANU6	
LMRMANU7	
LMSACTRL (206)	
LMSBSUPR	
LMSNCTRL (205)	Digital Command System
LMSHLDPH	Launch/High-Speed Abort
LMSSUPER	
LMXAINIT (206)	
LMXANDTV (206)	
LMXDDD	
LMXDDD5 (206)	
LMXDTV	
LMXGNGPR	
LMXHSDTV (206)	
LMXINITL	
LMXPHASE	
LMXPHRT	
LMXPPB	
LMXPPBI	
LMXPRNTR (206)	
LMXQLGIC	
LMXSPSPR	

EORP LOAD MODULE CROSS-REFERENCE
(Continued)

<u>Program</u>	<u>Book</u>
LMXTVGEN	Launch/High-Speed Abort
LMXZSHEL	Launch/High-Speed Abort
PMDDMT	Mission Planning
PMDDTVCC	
PMDGPM (206)	
PMDMPT (206)	
PMDPAD	
PMDRET	
PMDRPT	
PMDTAR	
PMDTIMP	
PMDTIPSS	
PMMDAN	
PMMDMT	
PMMFSET	
PMMFUD (206)	
PMMGPM (206)	
PMMMCD	
PMMMLC	
PMMMPT	
PMMPAD	
PMMPAR	
PMMPHL	
PMMPNE	
PMMRKJ	
PMMSCH (205)	
PMMS PQ	
PMMTISS	
PMMUDT	
PMMWTC	
PMMXFR	
PMQMED (206)	
PMSEXE (206)	
PMSTICN	
PMSVCT	
PMSVEC	
PMXSPT (206)	
PMXTAR	
RMDABRT	
RMDATVR	
RMDDBMP	
RMDDPSA	

Reentry Computations

EORP LOAD MODULE CROSS-REFERENCE
(Continued)

<u>Program</u>	<u>Book</u>
RMDEIED	Reentry Computations ↓
RMDGEDD	
RMDRBT	
RMDREAD	
RMDREC	
RMDRET	
RMDRTSD	
RMDRTSEP	
RMDRXDV	
RMDTADD	
RMDTARG	
RMGEPH	
RMGMSG	
RMGSCP	
RMGTTF	
RMMBAR	
RMMBGC	
RMMDBF	
RMMDBM	
RMMDBN	
RMMSAG	
RMMUDAT	
RMMXDV	
RMRRMED	
RMSDBMP	
RMSDEAD	
RMSPECT	
RMSSCS	
RMSTTF	
RMVKTR	
TAGPDUMP	Telemetry ↓
TMCDDDL	
TMCENTRF	
TMCLDTAB	
TMCRESET	
TMCMCSM	
TMCMXSIU	
TMDSPREQ	
TMLEMCMP	
TMLGCCMP	
TMLRESET	
TMQLSDCD	

EORP LOAD MODULE CROSS-REFERENCE
(Continued)ProgramBookTMQTMED
TMSDCNSL
TMSHSAGC
TMSHSAGS
TMSHSAMD
TMSHSCSM
TMSHSLEM
TMSHSLGC
TMSHSSIB
TMSHSSIU
TMSMEKIN
TMSTATUS
TMSUPER1
TMSUPER2
TMTVDBUG
TMXGDCTL
TMXLSREB
TMXPAPER
TMXSYSPL

Telemetry



GIMAPD APOGEE-PERIGEE DETERMINATION SUBROUTINE

REFERENCES

1. Programming Group — Rendezvous Planning
2. Programmer — M. Jones, J. Steed, R. Gayle
3. Language — H Level FORTRAN
4. Writeup Completion Date — 1/15/68
5. Data Tables Used — None
6. MED's Used — None
7. Flowchart — Figure 1

FUNCTION

Given a set of input elements, GIMAPD will determine the upcoming apogee and/or perigee information.

USAGE

1. Calling Sequence — CALL GIMAPD with a labeled common, i. e., COMMON/GIMAPD/Z(60), A(5), P(5), XLO, KAOP, KE.
2. Inputs — AEG input elements in Z(1) through Z(27), λ_0 , and an option indicator.

Z — A set of AEG elements are input in Z(1) - Z(27). If they are initialized, they should be exactly as output by the AEG. If they are not initialized, the following cells should be set. (Note $KZ \cong Z$.)

Z(8) = Osculating a (CUL)
 Z(9) = Osculating e
 Z(10) = Osculating i (rad.)
 Z(11) = Osculating l (rad.)
 Z(12) = Osculating g (rad.)
 Z(13) = Osculating h (rad.)
 Z(20) = T_0 - time of the above elements (flt. pt. hr.)
 Z(24) = Same as Z(20)
 KZ(2) = 0
 KZ(4) = As desired (usually non-zero)
 Z(2) = As desired (usually non-zero)

Approval	Approval
<i>J. R. Steed</i>	<i>A. Amitt</i>

$Z(3) = \text{Area (ft.}^2\text{)}$

$Z(4) = \text{Weight (lbs.)}$

$KZ(3) = 0$ (Not necessary for GIMAPD, but a good practice to follow.)

KAOP indicates information desired:

$= -1$ Perigee information only

$= 0$ Perigee and apogee information

$= +1$ Apogee information only

XLO λ_0 (inertial longitude of Greenwich at $T_0 = 0$) radians $0 \leq \lambda_0 < 2\pi$

3. Outputs and Exit — Apogee information is in A(1) through A(5), perigee information is in P(1) through P(5) and an error indicator KE.

Z(60):

If KAOP = -1, the AEG elements at perigee will be in Z(1) - Z(27).

If KAOP = 0, the AEG elements at apogee will be in Z(1) - Z(27), and the AEG elements at perigee will be in Z(28) - Z(54).

If KAOP = +1, the AEG elements at apogee will be in Z(1) - Z(27).

In all cases, the elements will be initialized. However, KZ(3) may be left non-zero.

KE < 0 An AEG processing error has occurred if GIMAPD returns to the calling program with a negative result.

KE = 0 GIMAPD has evaluated the request. This is the normal final return.

A(5) contains apogee information as outlined below

A(1) = T_{AP} - Time of apogee crossing (flt. pt. hr.),

A(2) = r_{AP} - Radius magnitude at apogee (CUL),

A(3) = φ'_{AP} - Geocentric latitude of apogee
(radians; $-\pi/2 \leq \varphi_{AP} \leq \pi/2$),

A(4) = λ_{AP} - Earth-fixed longitude of apogee at T_{AP} (rad),

A(5) = $h_{AP_{OB}}$ - Height of apogee above an oblate earth (CUL).

P(5) contains perigee information as outlined for apogee.

The following are constants:

TWOPI - 2π
 PI - π
 PI2 - $\pi/2$
 C2 - $(A^2 - B^2)/B^2$ B = polar radius of the Earth.
 C3 - A (CUL) Equatorial radius of the Earth.
 C4 - ω_e (rad./hr.) Earth rotation rate.

These are geophysical constants and do not concern the calling program.

4. Other Programs Used — PMMAEG - Analytical Ephemeris Generator.
5. Data Restrictions and Limits — Input AEG elements should be acceptable for the AEG to process. GIMAPD will use the full 60-word array that is input.
6. Error — KE, the AEG error return will be set negative when an AEG processing error occurs.
7. Other — All inputs are doubleword floating point except KAOP and KE, which are half-word fixed point integers.

METHOD

The program formulation is basically divided into two parts, high e and low e. The dividing point presently is

$e < .005$ low e
 $e \geq .005$ high e.

The same dividing point is currently used by the AEG. However, if the input elements are initialized, we use the AEG cell that indicates that the elements were initialized high e or low e. This is desirable because the osculating e may be initialized $\geq .005$, but in the course of updating, the value becomes less than .005 (or vice versa).

In the high e case, apogee and perigee information is obtained by using the AEG option to advance to a given mean anomaly (l). Apogee is at $l = \pi$ and perigee is at $l = 0$.

The low e case is considerably more complicated. The problem arises because neither l nor the argument of perigee (g) remains well defined. However, the

argument of latitude remains well defined for both low e and high e cases (as long as we are not in an equatorial orbit). The method used for low e is to write the radius magnitude (r) as a function of u , using a sine wave approximation (MSC Internal Note No. 64-FM-58).

$$(1) \quad r = r_o + X_r \sin (u - u_o)$$

where

r_o = mean radius of the orbit

u_o = u corresponding to r_o

X_r = amplitude of sine wave.

Since there are three unknowns, three positions in the orbit must be obtained. We use the AEG to obtain (u_1, r_1) , (u_2, r_2) , (u_3, r_3) at 15-minute intervals. Thus, by solving the 3 simultaneous equations we can obtain

$$(2) \quad u_o = \tan^{-1} \left(\frac{\sin u_1 - \sin u_2 - \gamma(\sin u_1 - \sin u_3)}{\gamma(\cos u_3 - \cos u_1) - \cos u_2 + \cos u_1} \right)$$

$$u_o \geq 0 \quad \text{where } \gamma = (r_1 - r_2)/(r_1 - r_3).$$

An analysis of (1) would eventually yield

$$u_x = u_o + \pi/2$$

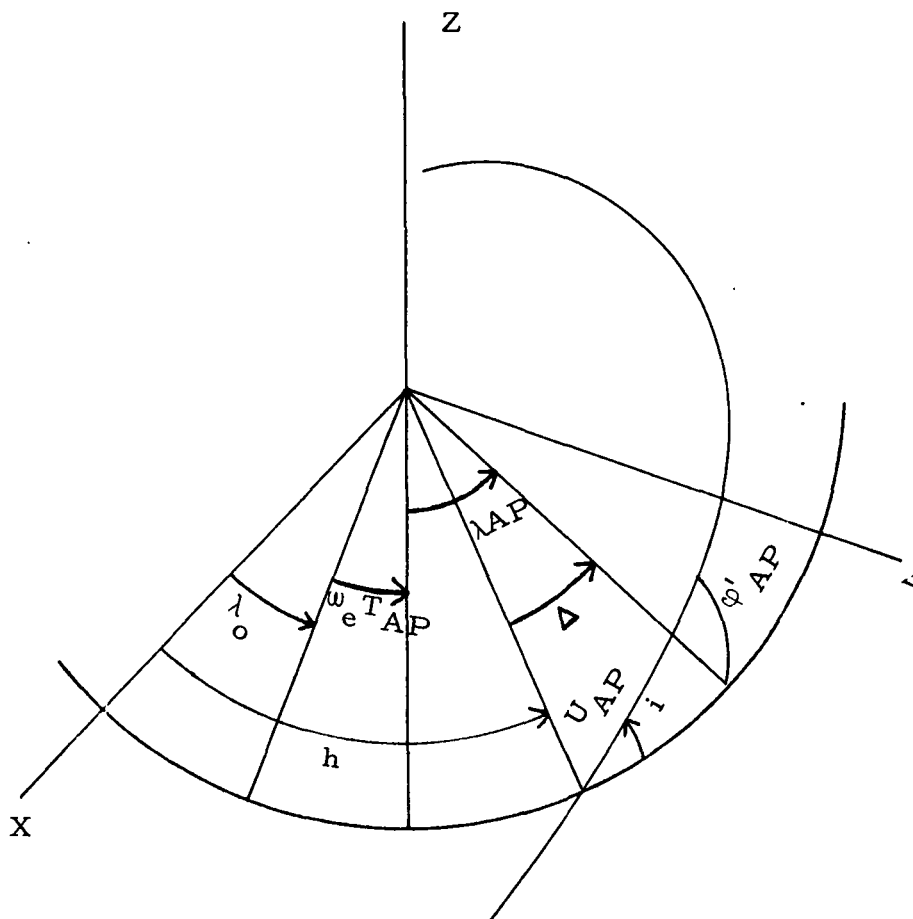
$$u_y = u_o - \pi/2.$$

In these equations, u_x and u_y are the arguments of latitude of apogee and perigee (not necessarily respectively).

We could then advance the AEG to u_x and u_y and pick up r_x and r_y . A comparison of r_x and r_y would differentiate apogee and perigee (obviously perigee being the smaller of the two).

It should be noted that a little special logic is necessary to handle $r_1 = r_3$.

Once we have the elements at apogee and perigee, we obtain the other desired quantities by the following means:



$$\varphi'_{AP} = \sin^{-1} (\sin u_{AP} \sin i_{AP})$$

$$-\pi/2 \leq \varphi'_{AP} \leq \pi/2$$

$$\Delta = \tan^{-1} \left(\frac{\sin u_{AP} \cos i_{AP}}{\cos u_{AP}} \right)$$

$$0 \leq \Delta < 2\pi$$

$$\lambda_{AP} = (u_{AP} + \Delta) - (\lambda_o + \omega_e T_{AP})$$

$$h_{AP_{OB}} = r_{AP} - \left(\sqrt{1 + \frac{A^2 - B^2}{B^2} \sin^2 \varphi'_{AP}} \right)$$

(The values of A, B, ω_e , & λ_o are defined under "Outputs and Exit".)

The same equations are also used on the elements at perigee.

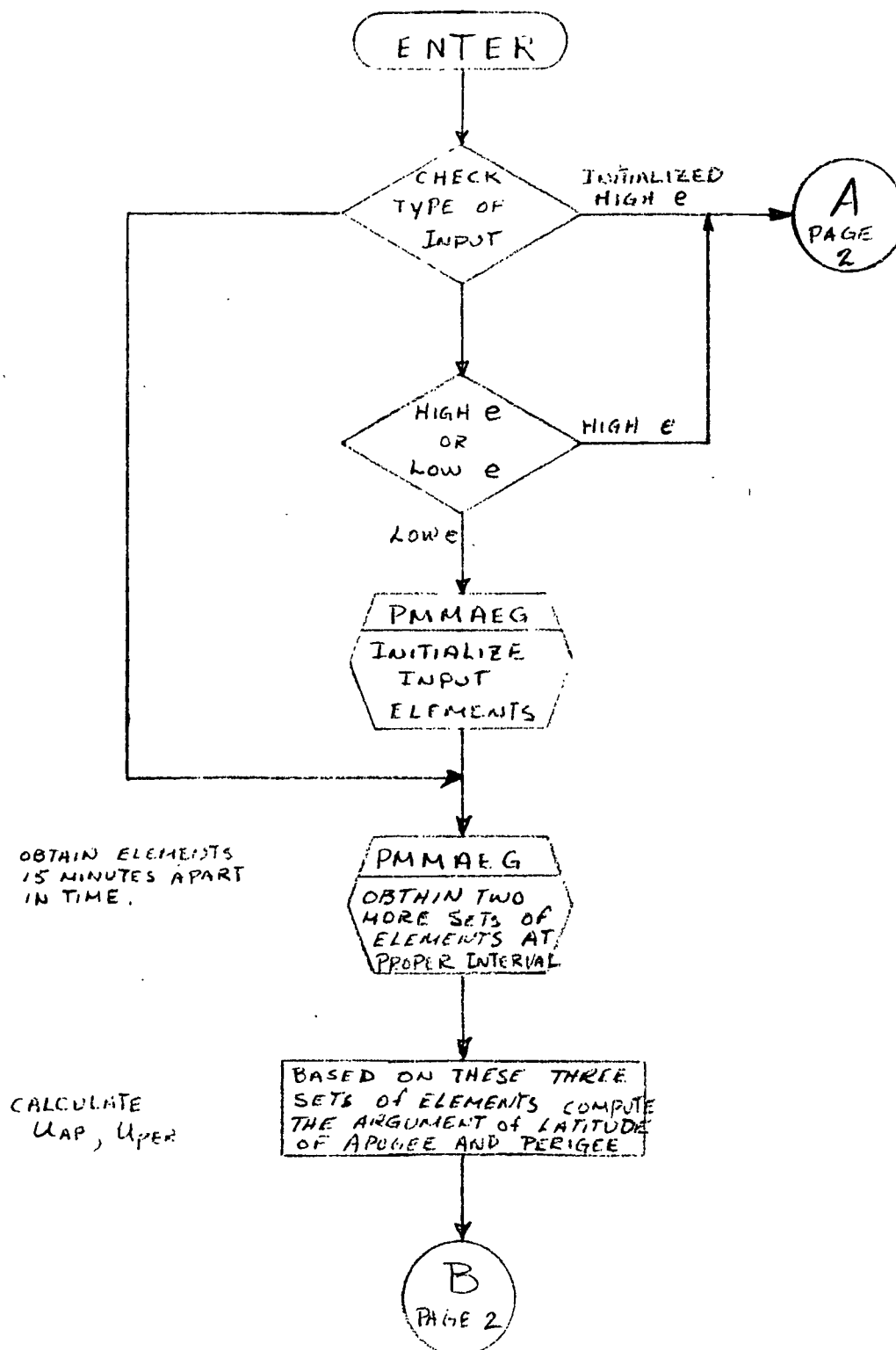


Figure 1

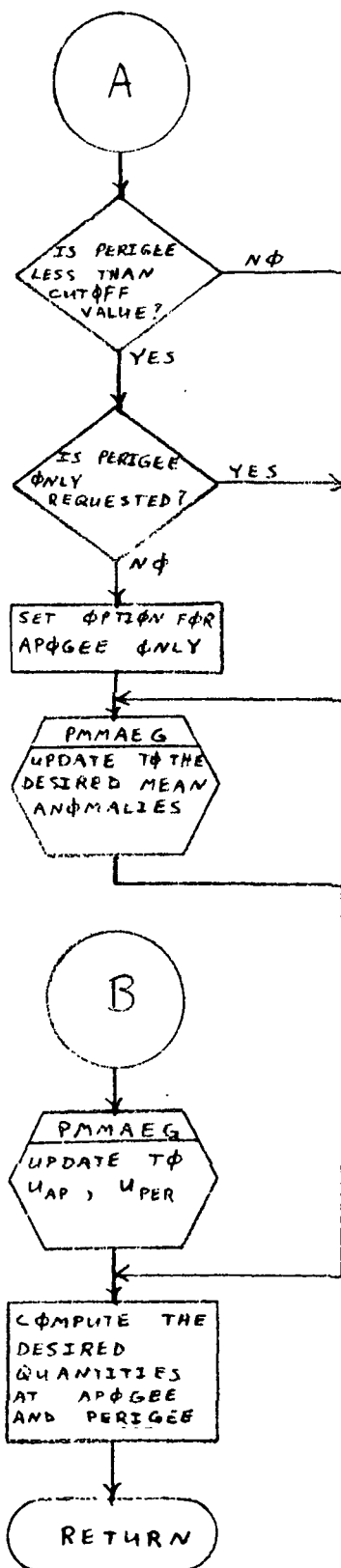


Figure 1

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. GIMIKC

DATE 7/7/67 (206)

REV

PAGE 1 of 3

- a. Symbol — GIMIKC
- b. Program Name — Inertial to Keplerian Conversion Subroutine
- c. Programming Group — Mission Planning
- d. Programmer — L. Janes
- e. Language — Assembler
- f. Data Tables Used — None
- g. MED's Used — None
- h. Write-up Completion Date — 9/15/66
- i. Flowchart — Figure GIMIKC-1

FUNCTION

GIMIKC converts the inertial $\bar{R}$ and $\bar{V}$ vectors to Keplerian elements (A, E, I, ℓ , g, h).

USAGE

- a. Calling Sequence — CALL GIMIKC, (RV, KEP, SQRTMU)
- b. Inputs — RV — 6 double words of $\bar{R}$ and $\bar{V}$ components.
SQRTMU — 1 double word $-\sqrt{\mu}$
Note: SQRTMU may be within the calling program, or it may be a system parameter.
- c. Outputs and Exit — Output — KEP — 6 double words of Keplerian elements. Exit via RETURN.
- d. Other Programs Used — DSQRT, DARCOS, GLQATN
- e. Data Restrictions and Limits — Elliptic orbits only
- f. Error — None
- g. Other — Since the calling program provides $\sqrt{\mu}$, $\bar{R}$ and $\bar{V}$ may be in any units if $\bar{R}$, $\bar{V}$, and μ are consistent. (A and $\bar{R}$ will have the same units.)

METHOD

$\bar{V} = \bar{V} / \sqrt{\mu}$ $\bar{R}$ and $\bar{V}$ now form a "canonical" set

$\bar{H} = \bar{R} \times \bar{V}$ $R = |\bar{R}|$, $V = |\bar{V}|$

Approval	Approval
<i>L. Janes</i>	<i>L. Janes</i>

METHOD

Solve $M = EA - E \sin EA$
for EA using Newton's method of successive approximations.
(E = eccentricity, EA = eccentric anomaly, $M = \int =$ mean anomaly.)

The $X'' - Y''$ coordinate system is in the plane of the orbit with X'' directed toward pericenter:

$$\begin{aligned} R_{X''} &= A (\cos EA - E) & (A = \text{semi-major axis}) \\ R_{Y''} &= A \sqrt{1 - E^2} \sin EA \\ V_{X''} &= \sqrt{\mu} \sin EA / \sqrt{A} (1 - E \cos EA) \\ V_{Y''} &= \sqrt{\mu} \sqrt{1 - E^2} \cos EA / \sqrt{A} (1 - E \cos EA) \end{aligned}$$

The $X' - Y'$ coordinate system is the result of rotating the $X'' - Y''$ coordinate system through the angle ω ($\omega = g$ = argument of perigee) in the plane of the orbit so that X' is directed toward the ascending node:

$$\begin{aligned} X' &= X'' \cos \omega - Y'' \sin \omega \\ Y' &= X'' \sin \omega + Y'' \cos \omega \end{aligned}$$

(These transformations are performed on both the $\vec{R}$ and $\vec{V}$ vectors to get $R_{X'}, R_{Y'}, V_{X'}, V_{Y'}$.)

Z'' and Z' are perpendicular to the orbit plane and so the $\vec{R}$ and $\vec{V}$ components in that direction have been zero.

The inertial $X - Y - Z$ coordinate system can be achieved by rotating the $X' - Y' - Z'$ coordinate system through the angles $-I$ (I = inclination) about the X' axis and $-\Omega$ ($\Omega = h$ = argument of ascending node) about the Z axis:

$$\begin{aligned} X &= X' \cos \Omega - (Y' \cos I) \sin \Omega \\ Y &= X' \sin \Omega + (Y' \cos I) \cos \Omega \\ Z &= Y' \sin I \end{aligned}$$

(These transformations are also performed on both $\vec{R}$ and $\vec{V}$ vectors to get $R_X, R_Y, R_Z, V_X, V_Y, V_Z$ ($X, Y, Z, \dot{X}, \dot{Y}, \dot{Z}$.)

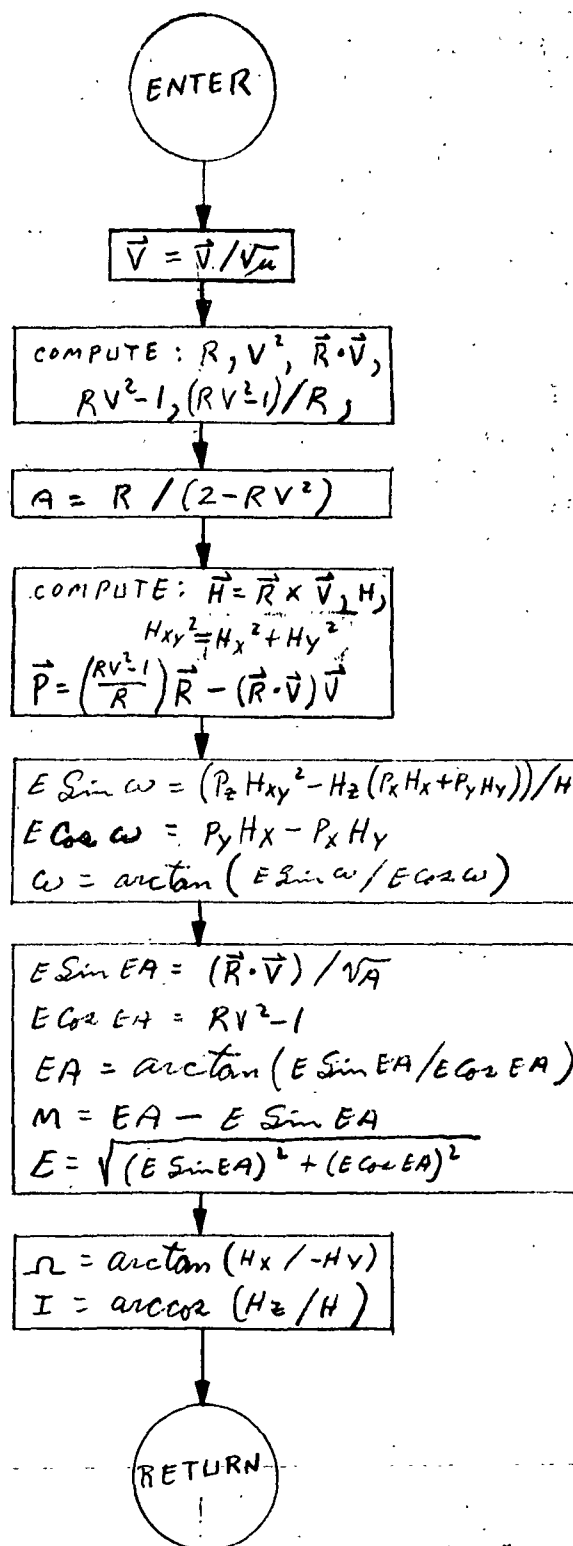


Figure GIMIKC-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. GIMKIC
 DATE 7/7/67 (206)
 REV
 PAGE 1 of 3

- a. Symbol — GIMKIC
- b. Program Name — Keplerian to Inertial Conversion Subroutine
- c. Programming Group — Mission Planning
- d. Programmer — L. Janes
- e. Language — Assembler
- f. Data Tables Used — None
- g. MED's Used — None
- h. Writeup Completion Date — 9/15/66
- i. Flowchart — Figure GIMKIC-1

FUNCTION

GIMKIC converts the Keplerian elements (A, E, I, l, g, h) to inertial $\bar{R}$ & $\bar{V}$ vectors.

USAGE

- a. Calling Sequence — CALL GIMKIC, (KEP, RV, SQRTMU)
- b. Inputs — KEP - 6 double words of Keplerian elements
SQRTMU - 1 double word - $\sqrt{\mu}$
- c. Outputs and Exit — RV - 6 double words of $\bar{R}$ & $\bar{V}$ components.
Exit via RETURN.

NOTE: SQRTMU may be either a number within the calling program or a system parameter.

- d. Other Programs Used — DSIN, DCOS, DSQRT.
- e. Data Restrictions and Limits — Elliptic orbits only
- f. Error — None
- g. Other — The units of the output $\bar{R}$ and $\bar{V}$ are determined by the input A and $\sqrt{\mu}$ (A and $\bar{R}$ will have the same units).

Approval	Approval
<i>L. Janes</i>	<i>L. Janes</i>

$$\vec{P} = (RV^2 - 1) * \vec{R}/R - (\vec{R} \cdot \vec{V}) * \vec{V}$$

$\vec{P}$ is directed toward perigee.

$$A = R/(2 - RV^2) \quad (A = \text{semi-major axis})$$

$$E * \sin \omega = (P_Z(H_X^2 + H_Y^2) - H_Z(P_X H_X + P_Y H_Y)) / H \quad (H = |\vec{H}|)$$

$$E * \cos \omega = P_Y H_X - P_X H_Y$$

$$\omega = \arctan (E * \sin \omega / E * \cos \omega) \quad (\omega = g = \text{argument of perigee})$$

$$E * \cos EA = RV^2 - 1 \quad (EA = \text{eccentric anomaly})$$

$$E * \sin EA = (\vec{R} \cdot \vec{V}) / \sqrt{A}$$

$$EA = \arctan (E * \sin EA / E * \cos EA)$$

$$M = EA - (E * \sin EA) \quad (M = \ell = \text{mean anomaly})$$

$$E = \sqrt{(E * \sin EA)^2 + (E * \cos EA)^2} \quad (E = \text{eccentricity})$$

$$\Omega = \arctan (H_X / -H_Y) \quad (\Omega = h = \text{argument of ascending node})$$

$$I = \arccos (H_Z / H) \quad (I = \text{inclination angle})$$

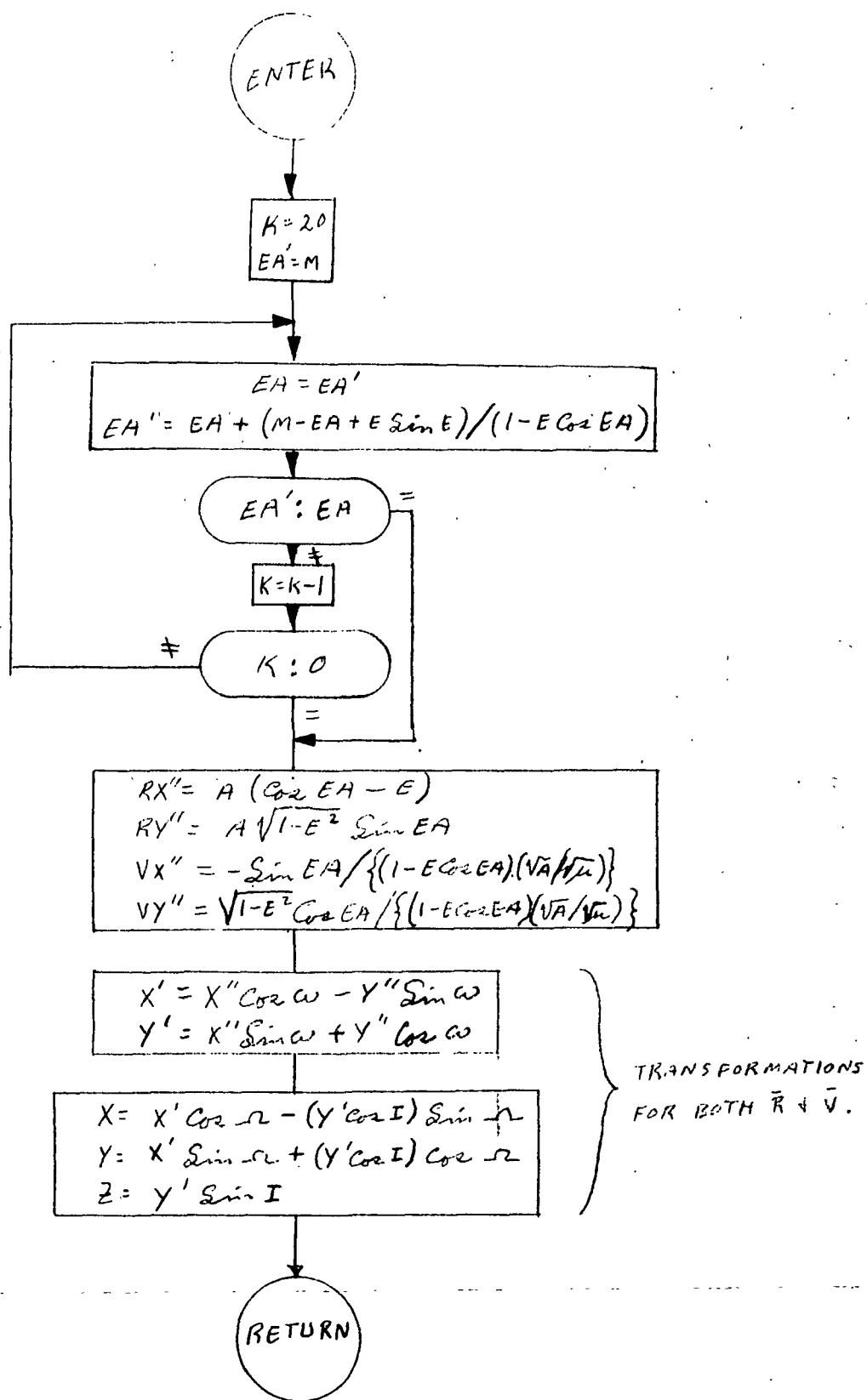


Figure GIMKIC-1. Flowchart

(Sheet 1 of 1)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

2. GLMRTM-GLMRTV

DATE 1/31/68 (206)

REV

PAGE 1 of 8

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

GLMRTM-GLMRTV
MATRIX/VECTOR ROTATION ROUTINEREFERENCES

1. Programming Group — Maneuver Control (Mission Planning)
2. Programmers — M. Eagle, M. Banks
3. Language — Assembler
4. Writeup Completion Date — 12/21/67
5. Data Tables Used — None
6. MED's Used — None
7. Flowchart — Figure 1

FUNCTION

This routine rotates a given matrix or vector about a given axis(es) through a given angle(s). The matrix or vector must be stored by row. All inputs, outputs and computations are in floating-point long precision.

USAGE

1. Calling Sequence —

Matrix - CALL GLMRTM, (SAVE, A, B, THET1, K1, THET2, K2, ... THETN, KN)

Vector - CALL GLMRTV, (SAVE, A, B, THET1, K1, THET2, K2, ... THETN, KN)

2. Inputs —

SAVE = Work area of 229 bytes on a doubleword boundary

A = Matrix or vector, floating-point long precision

THETi = Angle of ith rotation, floating-point long precision

Ki = Axis of ith rotation, fixed point: 1=X-axis, 2=Y-axis, 3=Z-axis

3. Outputs and Exit — B = Rotated matrix or vector, stored by row floating-point long precision

4. Other Programs Used — DSIN, DCOS

5. Data Restrictions and Limits — The matrices are dimensioned (3 x 3), and the vectors are dimensioned (3 x 1).

6. Error — None

7. Other — Storage information: GLMRTM/GLMRTV-344 bytes

Approval	Approval
<i>M. Banks</i>	<i>S.E. James</i>

METHOD

For matrix rotation:

$$C^{(i)} = \begin{pmatrix} C_{11}^{(i)} & C_{12}^{(i)} & C_{13}^{(i)} \\ C_{21}^{(i)} & C_{22}^{(i)} & C_{23}^{(i)} \\ C_{31}^{(i)} & C_{32}^{(i)} & C_{33}^{(i)} \end{pmatrix} = R^{(i)} \begin{pmatrix} C_{11}^{(i-1)} & C_{12}^{(i-1)} & C_{13}^{(i-1)} \\ C_{21}^{(i-1)} & C_{22}^{(i-1)} & C_{23}^{(i-1)} \\ C_{31}^{(i-1)} & C_{32}^{(i-1)} & C_{33}^{(i-1)} \end{pmatrix}$$

For vector rotation:

$$C^{(i)} = \begin{pmatrix} C_{11}^{(i)} \\ C_{21}^{(i)} \\ C_{31}^{(i)} \end{pmatrix} = R^{(i)} \begin{pmatrix} C_{11}^{(i-1)} \\ C_{12}^{(i-1)} \\ C_{13}^{(i-1)} \end{pmatrix} \quad \begin{array}{l} \text{where } A = C^{(0)} \\ B = C^{(N)} \\ \text{and } i = 1, 2, \dots, N \end{array}$$

For Ki = 1:

$$R^{(i)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_i & \sin \theta_i \\ 0 & -\sin \theta_i & \cos \theta_i \end{pmatrix} \quad \begin{array}{c} \text{Ki=2} \\ R^{(i)} = \begin{pmatrix} \cos \theta_i & 0 & -\sin \theta_i \\ 0 & 1 & 0 \\ \sin \theta_i & 0 & \cos \theta_i \end{pmatrix} \end{array}$$

Ki=3

$$R^{(i)} = \begin{pmatrix} \cos \theta_i & \sin \theta_i & 0 \\ -\sin \theta_i & \cos \theta_i & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

where $\theta_i = \text{THETi}$.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

DATE 1/31/68 (206)

REV

PAGE 3 of 8

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

Type = 1 CC rotation of + θ about +X axis

$$\begin{pmatrix} \bar{X}' \\ \bar{Y}' \\ \bar{Z}' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} \bar{X} \\ \bar{Y} \\ \bar{Z} \end{pmatrix} = \begin{pmatrix} \bar{X} \\ \bar{Y} \cos \theta + \bar{Z} \sin \theta \\ -\bar{Y} \sin \theta + \bar{Z} \cos \theta \end{pmatrix}$$

Type = 2 CC rotation of + θ about +Y axis

$$\begin{pmatrix} \bar{X}' \\ \bar{Y}' \\ \bar{Z}' \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} \bar{X} \\ \bar{Y} \\ \bar{Z} \end{pmatrix} = \begin{pmatrix} \bar{X} \cos \theta - \bar{Z} \sin \theta \\ \bar{Y} \\ \bar{X} \sin \theta + \bar{Z} \cos \theta \end{pmatrix}$$

Type = 3 CC rotation of + θ about +Z axis

$$\begin{pmatrix} \bar{X}' \\ \bar{Y}' \\ \bar{Z}' \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \bar{X} \\ \bar{Y} \\ \bar{Z} \end{pmatrix} = \begin{pmatrix} \bar{X} \cos \theta + \bar{Y} \sin \theta \\ -\bar{X} \sin \theta + \bar{Y} \cos \theta \\ \bar{Z} \end{pmatrix}$$

CALL {GLMRTM
GLMRTV (SAVE, C₁, C₂, θ_1 , TYPE₁, θ_2 , TYPE₂, θ_3 , TYPE₃,)

where $\left\{ \begin{array}{l} C_1 \text{ is old coordinate system (3x3 matrix)} \\ \text{Input } \left\{ \begin{array}{l} \theta \text{ is angle of rotation} \\ \text{Type tells axis of rotation (as defined above)} \end{array} \right. \\ \text{Output } [C_2 \text{ is rotated coordinate system (3x1 matrix)}] \end{array} \right.$

One possible scheme is to designate the row vectors as

$$\begin{pmatrix} \bar{A}' \\ \bar{B}' \\ \bar{C}' \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \bar{A} \\ \bar{B} \\ \bar{C} \end{pmatrix}$$

Then use the rotation about +X as the rotation formula and define A^1 , B^1 , C^1 , $\bar{A}$, $\bar{B}$, $\bar{C}$ according to which axis is the axis of rotation. This can be done in assembly language by initializing with the proper vector address.

$$\bar{A}' = \bar{A}$$

$$\bar{B}' = \bar{B} \cos \theta + \bar{C} \sin \theta$$

$$\bar{C}' = -\bar{B} \sin \theta + \bar{C} \cos \theta$$

$$\text{For type} = 1 : \quad \bar{A}=\bar{X}, \bar{B}=\bar{Y}, \bar{C}=\bar{Z} \quad \left\{ \begin{array}{l} \bar{X}' = \bar{X} \\ \bar{Y}' = \bar{Y} \cos \theta + \bar{Z} \sin \theta \\ \bar{Z}' = -\bar{Y} \sin \theta + \bar{Z} \cos \theta \end{array} \right.$$

$$\text{For type} = 2 : \quad \bar{A}=\bar{Y}, \bar{B}=\bar{Z}, \bar{C}=\bar{X} \quad \left\{ \begin{array}{l} \bar{Y}' = \bar{Y} \\ \bar{Z}' = \bar{Z} \cos \theta + \bar{X} \sin \theta \\ \bar{X}' = -\bar{Z} \sin \theta + \bar{X} \cos \theta \end{array} \right.$$

$$\text{For type} = 3 : \quad \bar{A}=\bar{Z}, \bar{B}=\bar{X}, \bar{C}=\bar{Y} \quad \left\{ \begin{array}{l} \bar{Z}' = \bar{Z} \\ \bar{X}' = \bar{X} \cos \theta + \bar{Y} \sin \theta \\ \bar{Y}' = -\bar{X} \sin \theta + \bar{Y} \cos \theta \end{array} \right.$$

As an additional aid in using this routine the user should give special attention to the following notation consideration.

$$\begin{pmatrix} \bar{X} \\ \bar{Y} \\ \bar{Z} \end{pmatrix} = \begin{pmatrix} \bar{A} \\ \bar{B} \\ \bar{C} \end{pmatrix} = \begin{pmatrix} A_x & A_y & A_z \\ B_x & B_y & B_z \\ C_x & C_y & C_z \end{pmatrix} = \begin{bmatrix} L(1) & L(2) & L(3) \\ L(4) & L(5) & L(6) \\ L(7) & L(8) & L(9) \end{bmatrix}$$

where $L(N)$ = Relative location in some contiguous doubleword storage area with $N=1, 2, 3, \dots, 9$.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

DATE 1/31/68 (206)

REV

PAGE 5 of 8

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

Special care should be exercised when using the double subscript notation in FORTRAN to define the above matrix areas. It is suggested that three contiguous single subscripted variables be used for clarity, e.g., COMMON X(3), Y(3), Z(3), A(3), B(3), C(3).

Furthermore, it is suggested that all assembler language users pass all arguments to this routine in a parameter list.

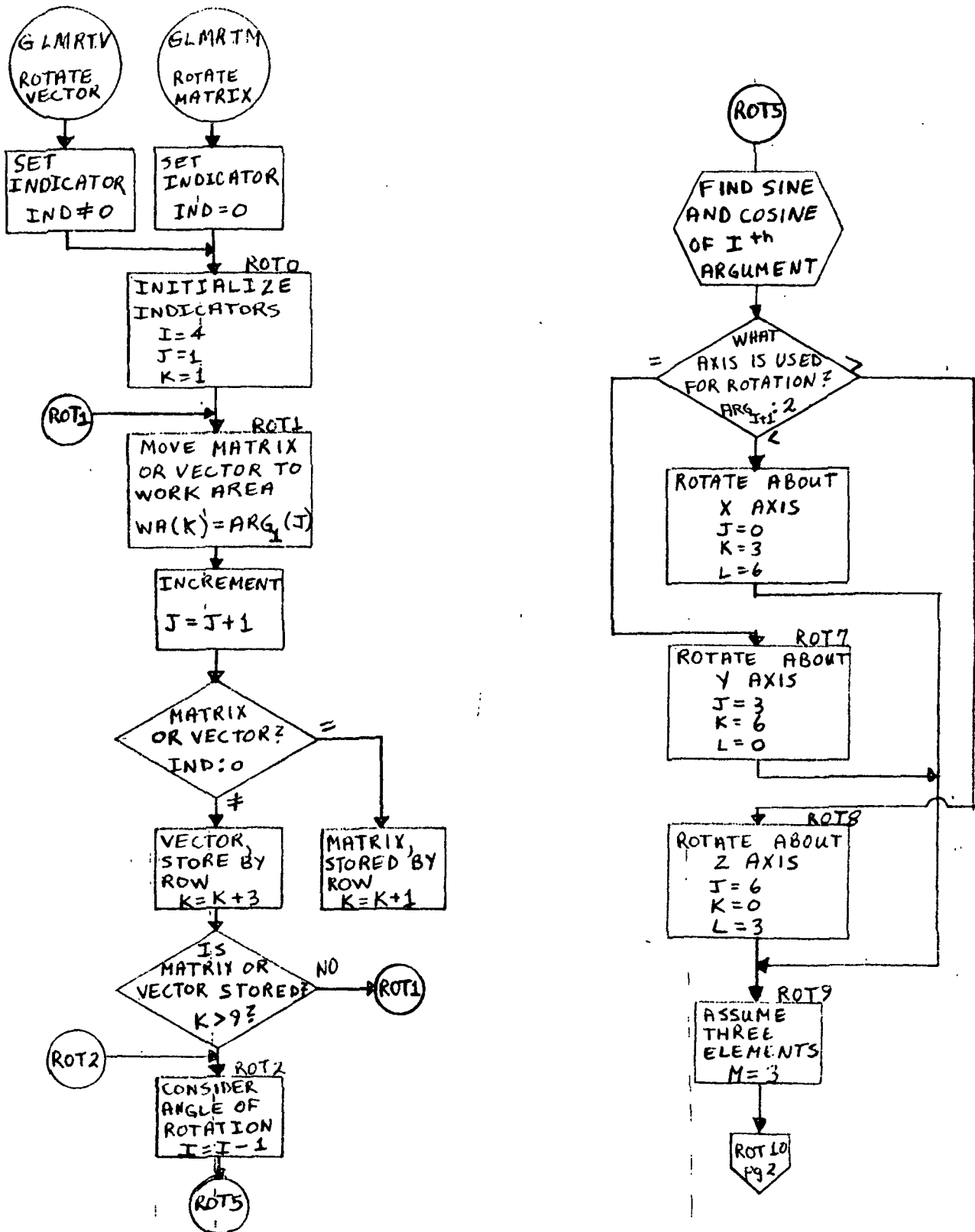


Figure 1

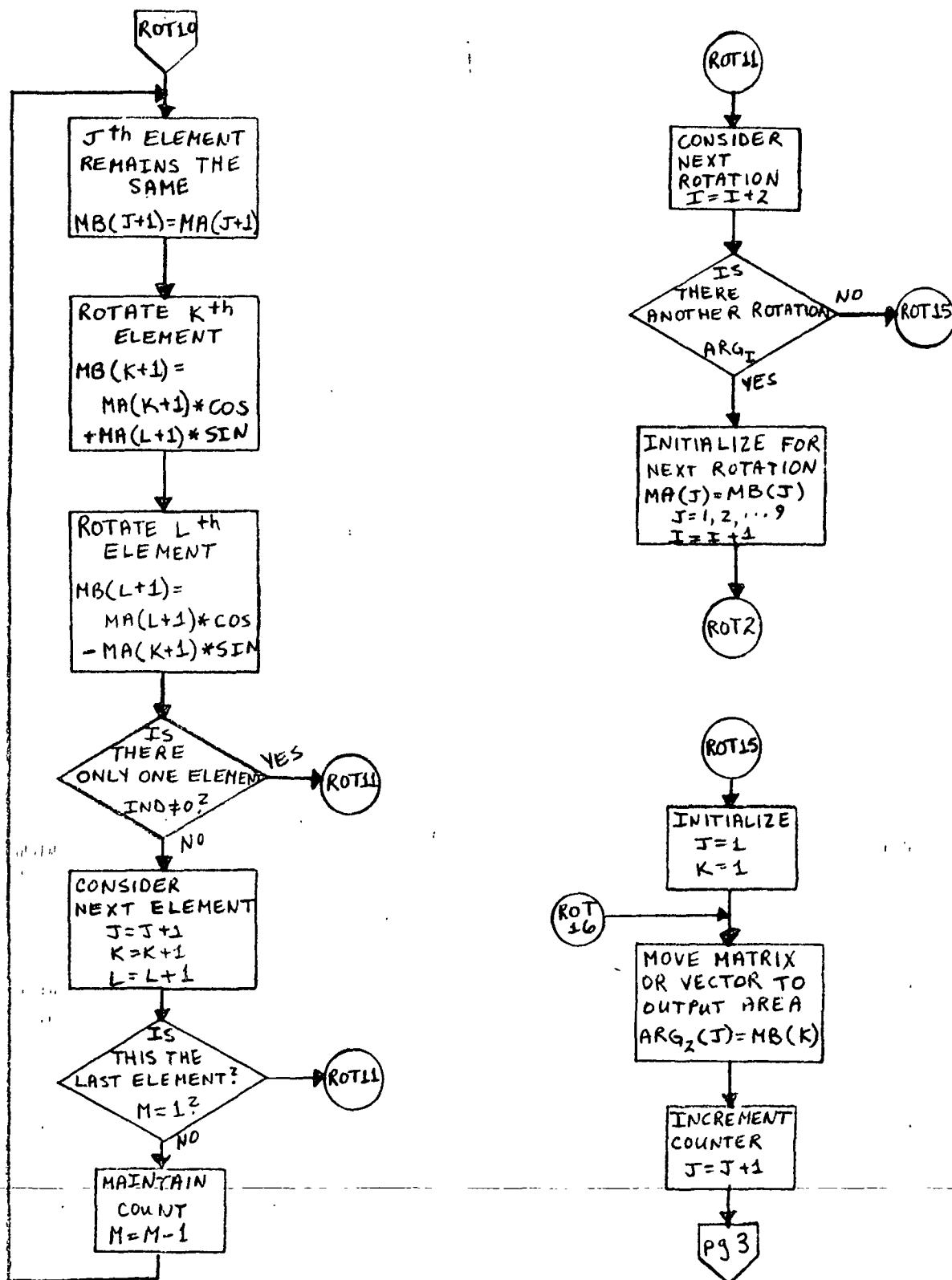


Figure 1

(Sheet 2 of 3)

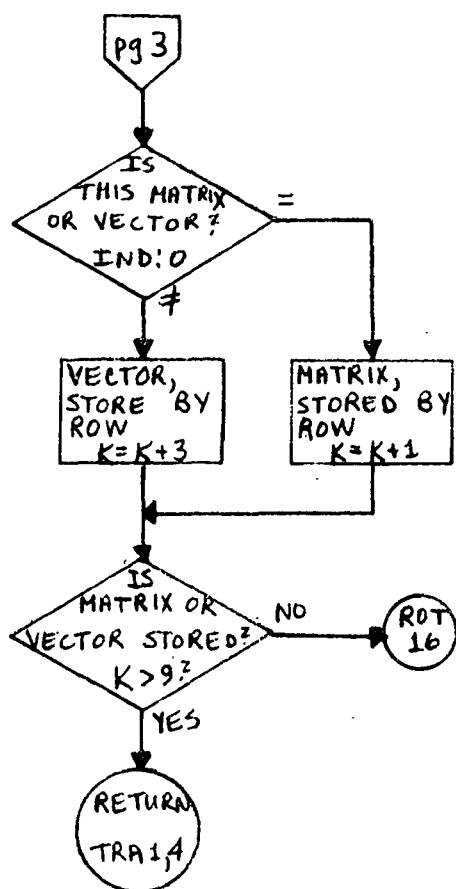


Figure 1

(Sheet 3 of 3)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. GIMGBL
 DATE 8/25/67 (EORP)
 REV
 PAGE 1 of 10

- a. Symbol — GIMGBL
- b. Program Name — Gimbal, Thrust, and Weight Loss Rate Subroutine
- c. Programming Group — Maneuver Control (Mission Planning)
- d. Programmer(s) — R. Cusey
- e. Language — Fortran
- f. Data Tables Used — None
- g. MED's Used — None
- h. Writeup Completion Date — 7/20/67
- i. Flowcharts — Figures GIMGBL-1 through 3

FUNCTION

Given the vehicle configuration and weight, tables of vehicle center of gravity location versus vehicle weight for each vehicle, and tables of thrust and weight loss rate versus vehicle weight for each thruster, this subroutine determines thruster gimbal angles and/or thrust and weight loss rate using a table look-up procedure and linear interpolation.

USAGE

- a. Calling Sequence — CALL GIMGBL (WT, RY, RZ, T, WDOT, ITC, IC, IA, IJ, D)
- b. Inputs —
 - WT= Weights (lb.) for which values are desired (two double words), where the first double word is CSM weight and the second double word is LM weight. If the vehicles are not docked, any value can be used for the weight of the vehicle which is not being considered. Positive or negative weights are valid inputs.
 - ITC= Thruster code (half word)
 - 33 for SPS
 - 34 for APS
 - 35 for DPS
 - IC= Configuration code at beginning of the maneuver (halfword)
 - 0 for CSM
 - 1 for LM
 - 2 for CSM and LM (docked)
 - IA= Trim angle indicator (half word)
 - 1 for trim angles, thrust, and weight loss rate desired outputs
 - 0 for thrust and weight loss rate desired outputs
 - +1 for trim angles desired outputs
 - IJ= Indicator to tell if LM descent stage is included in the configuration at the beginning of this maneuver (halfword). This indicator is applicable only if ITC=33 and IC=2.
 - 0 for descent stage included
 - 1 for descent stage not included
 - D= Docking angle (radians) (double word)

Approval	Approval
<i>R. E. Cusey</i>	<i>S. E. James</i>

- c. Outputs and Exit — EXIT VIA RETURN.
 RY = Rotation (radians) about the y-body axis (double word), and
 RZ = Rotation (radians) about the z-body axis (double word), such that

$$\begin{bmatrix} \hat{X}_T \\ \hat{Y}_T \\ \hat{Z}_T \end{bmatrix} = R_Z(RZ) R_Y(RY) \begin{bmatrix} \hat{X}_B \\ \hat{Y}_B \\ \hat{Z}_B \end{bmatrix}$$

T = Thrust (lb.) (double word)
 WDOT = Weight loss rate (lb./hr.) (double word)

- d. Other programs Used — SIN, COS, SQRT, ATAN, ARSIN
 e. Data Restrictions and Limits — If ITC = 34 or if IA = 0, the contents of double words RY and RZ are not changed by GIMGBL.
 - If IA = +1, the contents of double words T and WDOT are not changed by GIMGBL. If either of the input weights is outside the range of the applicable table(s), output values correspond to the first or last entry in the table(s), whichever is appropriate. No indication of this condition is given by GIMGBL.
 f. Error — None
 g. Other — All internal calculations are single precision. Two auxiliary subroutines, GIMGB1 and GIMGB2, are used internally by GIMGBL. These subroutines are not available for general use.

METHOD

Tables of center of gravity (CG) location (in a reference coordinate system) versus vehicle weight for the CSM, the LM with the descent stage attached, and the LM without the descent stage, and tables of thrust and weight loss rate versus vehicle weight for the SPS, the DPS, and the APS are stored as system parameters.

Determination of Gimbal Angles —

Four cases can be defined:

- Case 1 — The vehicle is the CSM and the thruster is the SPS.
- Case 2 — The vehicle is the LM and the thruster is the DPS.
- Case 3 — The vehicle is the CSM/LM (docked) and the thruster is the SPS.
- Case 4 — The vehicle is the CSM/LM (docked) and the thruster is the DPS.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. GIMGBL
DATE 8/25/67 (EORP)
REV
PAGE 3 of 10

The CG location (in reference coordinates) of each vehicle involved is determined from the appropriate CG table (using linear interpolation if necessary). The CG location of each vehicle relative to the designated thruster hinge point is determined from a coordinate translation:

$$\text{Case 1} - X_C = X_{C_{REF}} - K_2; Y_C = Y_{C_{REF}}; Z_C = Z_{C_{REF}}$$

$$\text{Case 2} - X_L = X_{L_{REF}} - K_1; Y_L = Y_{L_{REF}}; Z_L = Z_{L_{REF}}$$

$$\text{Case 3} - X_C, Y_C, Z_C \text{ same as for Case 1.}$$

$$X'_L = X_{L_{REF}} - (K_1 + K_3); Y'_L = Y_{L_{REF}}; Z'_L = Z_{L_{REF}}$$

$$\text{Case 4} - X'_C = X_{C_{REF}} - (K_2 + K_3); Y'_C = Y_{C_{REF}}; Z'_C = Z_{C_{REF}}$$

$$X_L, Y_L, Z_L \text{ same as for Case 2.}$$

where X_C, Y_C, Z_C are CSM CG coordinates, X_L, Y_L, Z_L are LM CG coordinates, subscript REF indicates reference coordinates, primes indicate target vehicle, and K_1, K_2 , and K_3 are 154 inches, 833.2 inches, and 435.55 inches respectively by definition of the reference coordinate system.

For Cases 3 and 4 the CG coordinates of the target vehicle must be rotated counterclockwise about the x-axis through an angle of $(240^\circ - D)$ to properly align body axes. Here D is the docking angle in degrees. Since the x-body axes of the docked vehicles lie in opposite directions, the target vehicle CG coordinates computed above must be further rotated about the z-axis through 180° . Thus for Case 3

$$\begin{bmatrix} X_L \\ Y_L \\ Z_L \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(240^\circ - D) & \sin(240^\circ - D) \\ 0 & -\sin(240^\circ - D) & \cos(240^\circ - D) \end{bmatrix} \begin{bmatrix} X'_L \\ Y'_L \\ Z'_L \end{bmatrix}$$

and for Case 4

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 \cos(240^\circ - D) & \sin(240^\circ - D) \\ 0 - \sin(240^\circ - D) & \cos(240^\circ - D) \end{bmatrix} \begin{bmatrix} X'_C \\ Y'_C \\ Z'_C \end{bmatrix}$$

For Cases 1 and 2 the CG location (X, Y, Z) of the vehicle relative to the designated thruster hinge point is given by $\{X, Y, Z\} = \{X_C, Y_C, Z_C\}$ and $\{X, Y, Z\} = \{X_L, Y_L, Z_L\}$ respectively. For Cases 3 and 4 the CG location of the docked vehicles relative to the designated thruster hinge point is given by

$$X = (W_C X_C + W_L X_L) / (W_C + W_L)$$

$$Y = (W_C Y_C + W_L Y_L) / (W_C + W_L)$$

$$Z = (W_C Z_C + W_L Z_L) / (W_C + W_L)$$

where W_C and W_L are the CSM and LM weights respectively.

Knowing the CG location of the vehicle configuration, the thruster gimbal angles, P_G and Y_G , are determined from

$$P_G = \begin{cases} \tan^{-1}(-\frac{Z}{X}) & \text{for } X \neq 0 \\ 0 & \text{for } X = 0 \text{ or } R = 0 \end{cases} \quad -\frac{\pi}{2} \leq P_G \leq \frac{\pi}{2}$$

$$Y_G = \begin{cases} \sin^{-1}(\frac{Y}{R}) & \text{for } R \neq 0 \\ 0 & \text{for } R = 0 \end{cases} \quad -\frac{\pi}{2} \leq Y_G \leq \frac{\pi}{2}$$

where $R^2 = X^2 + Y^2 + Z^2$.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. GIMGBL
DATE 8/25/67 (EORP)
REV :
PAGE 5 of 10

Determination of Thrust and Weight Loss Rate —

Knowing the weight of the vehicle on which the designated thruster is located, the appropriate thrust-weight loss rate table is searched and the thrust and weight loss rate are determined using linear interpolation.

SUBROUTINE
GIMGBL

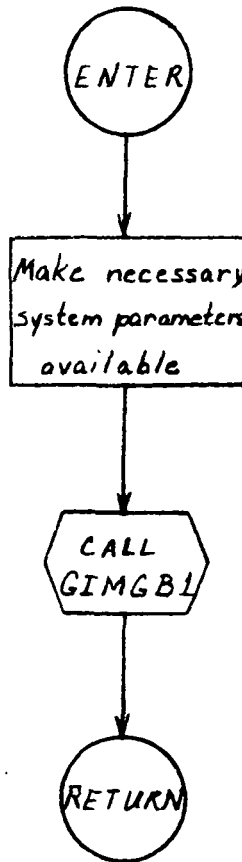


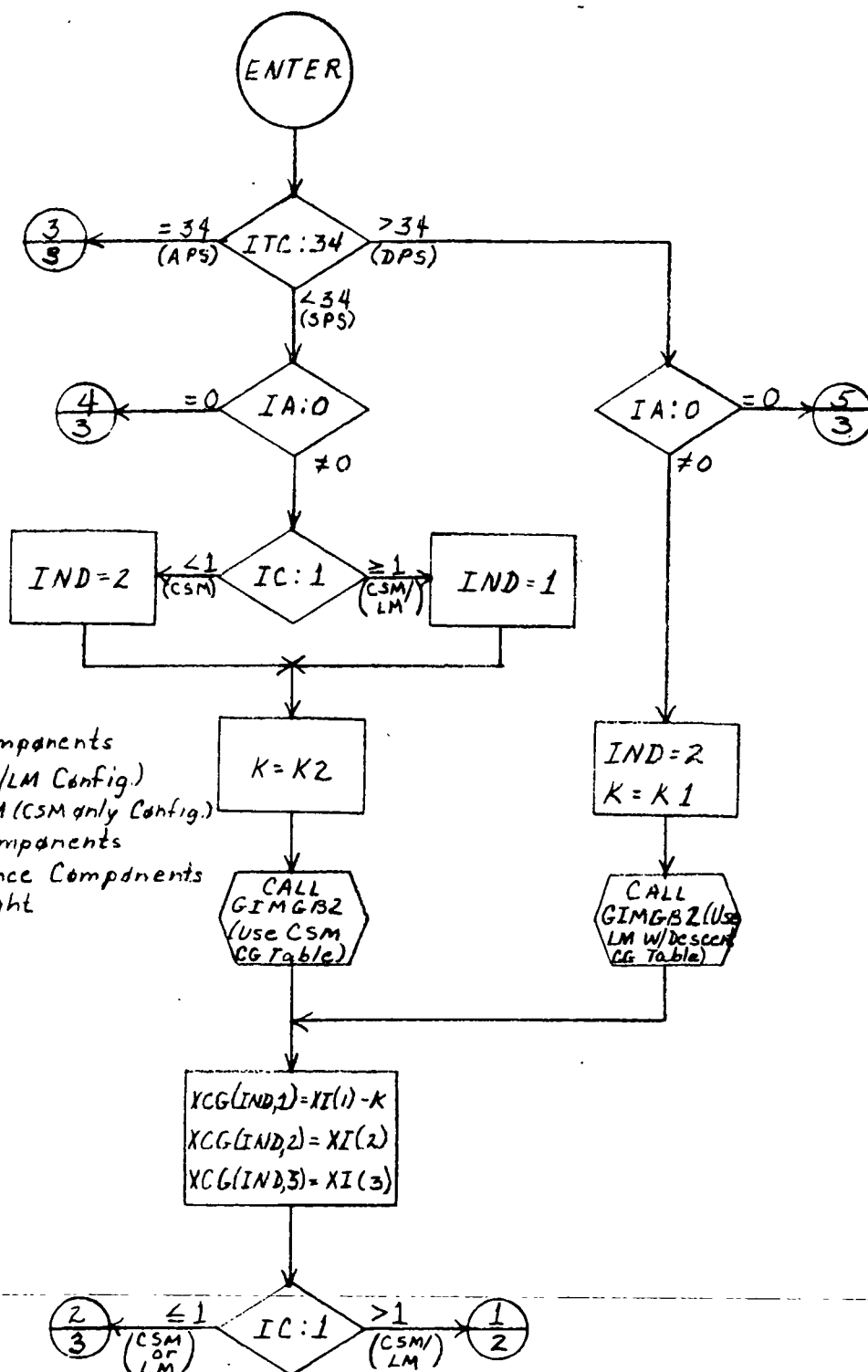
Figure GIMGBL-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. GIMGBL
DATE 8/25/67 (EORP)
REV
PAGE 7 of 10

Subroutine
GIMGB1



Note: XCG(IND,J)=CG Components

IND=1 ⇒ CSM (CSM/LM Config.)

IND=2 ⇒ LM or CSM (CSM only Config.)

J=1,2,3 ⇒ X,Y,Z Components

XI(J)=CG Reference Components

W(I)=Vehicle Weight

I=1 ⇒ CSM

I=2 ⇒ LM

I=3 ⇒ CSM/LM

Figure GIMGBL-2. Flowchart

(Sheet 1 of 3)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. GIMGBL
 DATE 8/25/67 (EORP)
 REV
 PAGE 8 of 10

Subroutine
 GIMGB1 (Cont.)

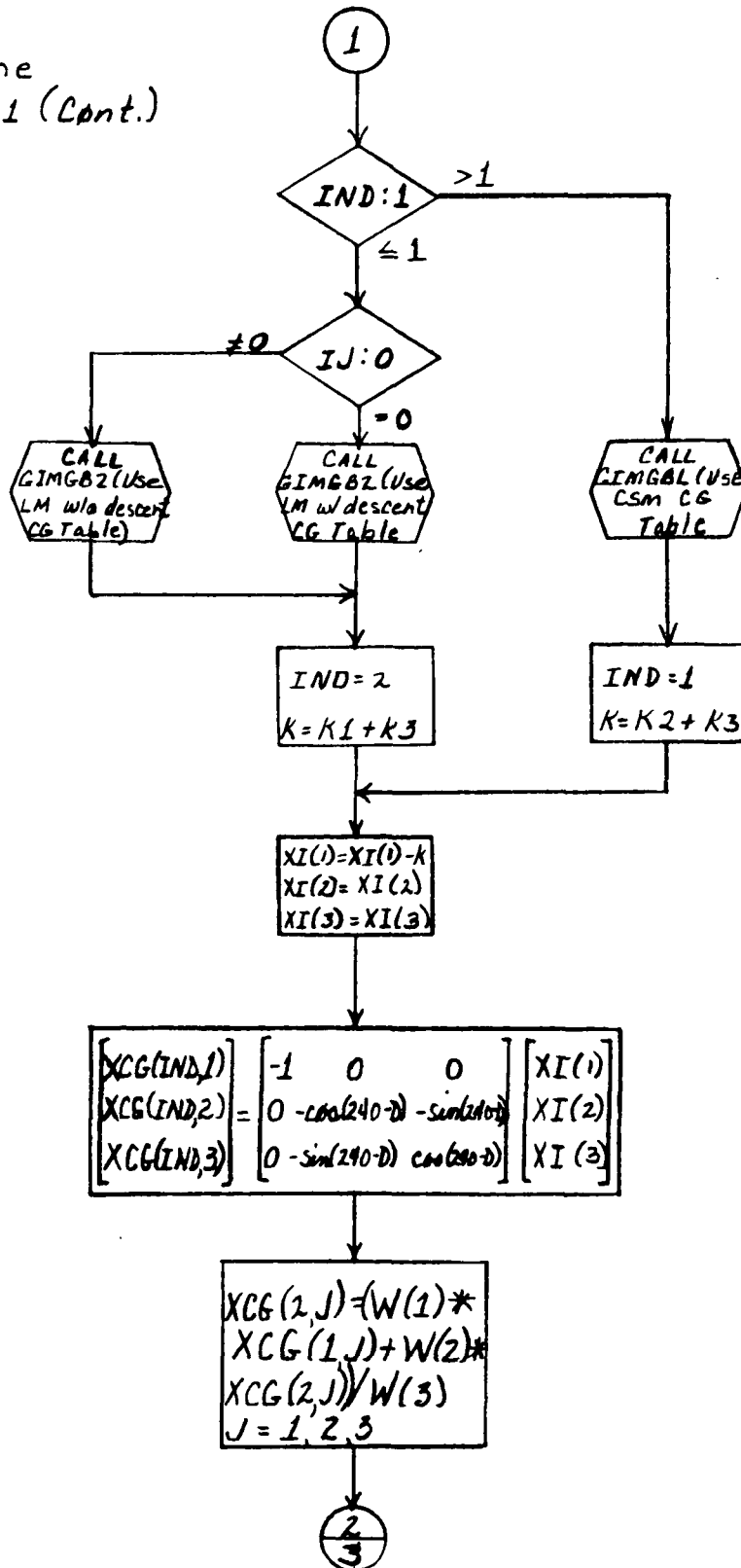


Figure GIMGBL-2, Flowchart

(Sheet 2 of 3)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. GIMGBL

DATE 8/25/67 (EORP)

REV

PAGE 9 of 10

Subroutine
GIMGB1 (Cont.)

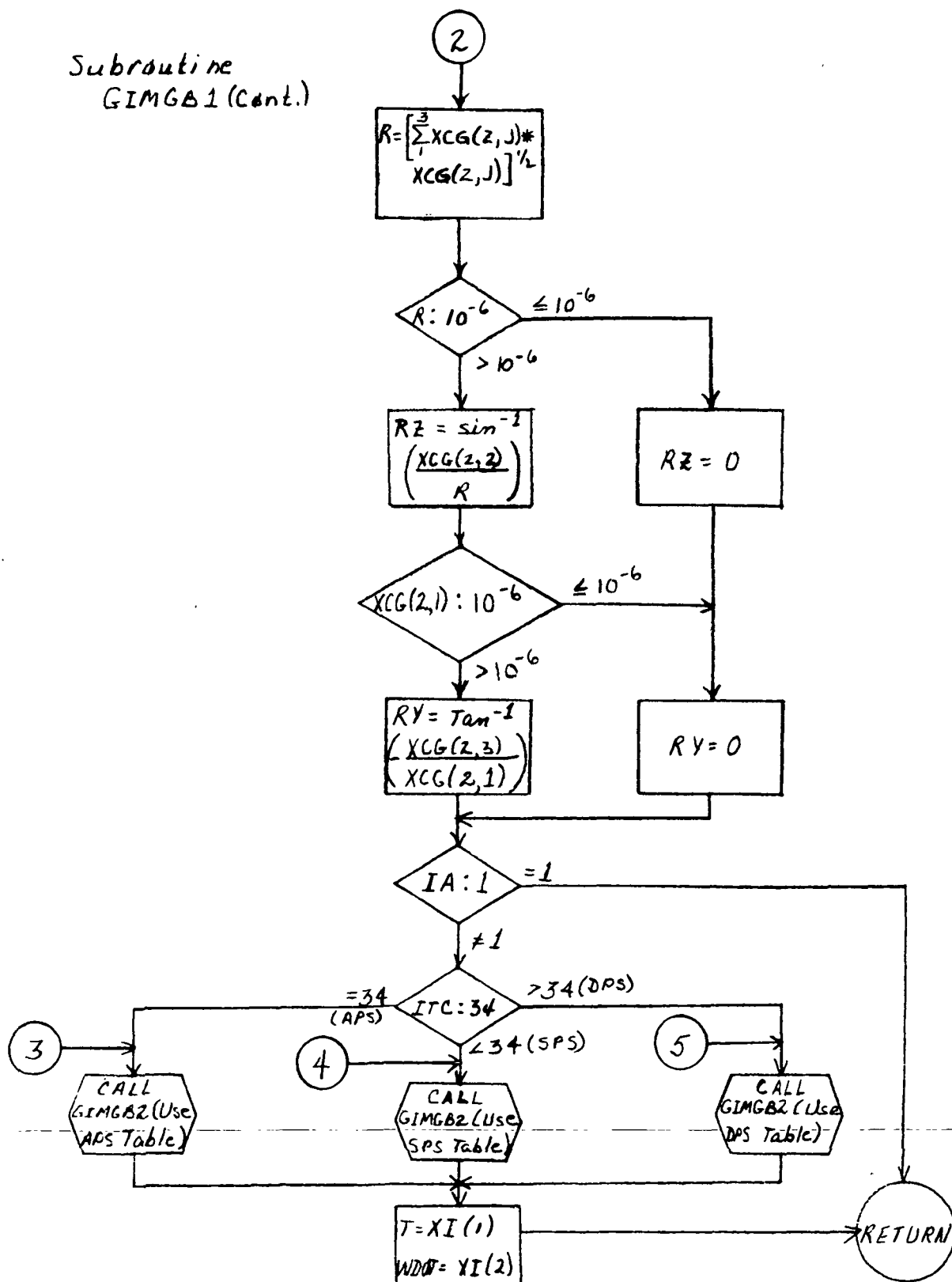


Figure GIMGBL-2. Flowchart

(Sheet 3 of 3)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. GIMGBL
 DATE 8/25/67 (EORP)
 REV
 PAGE 10 of 10

Subroutine
 GIMGB2

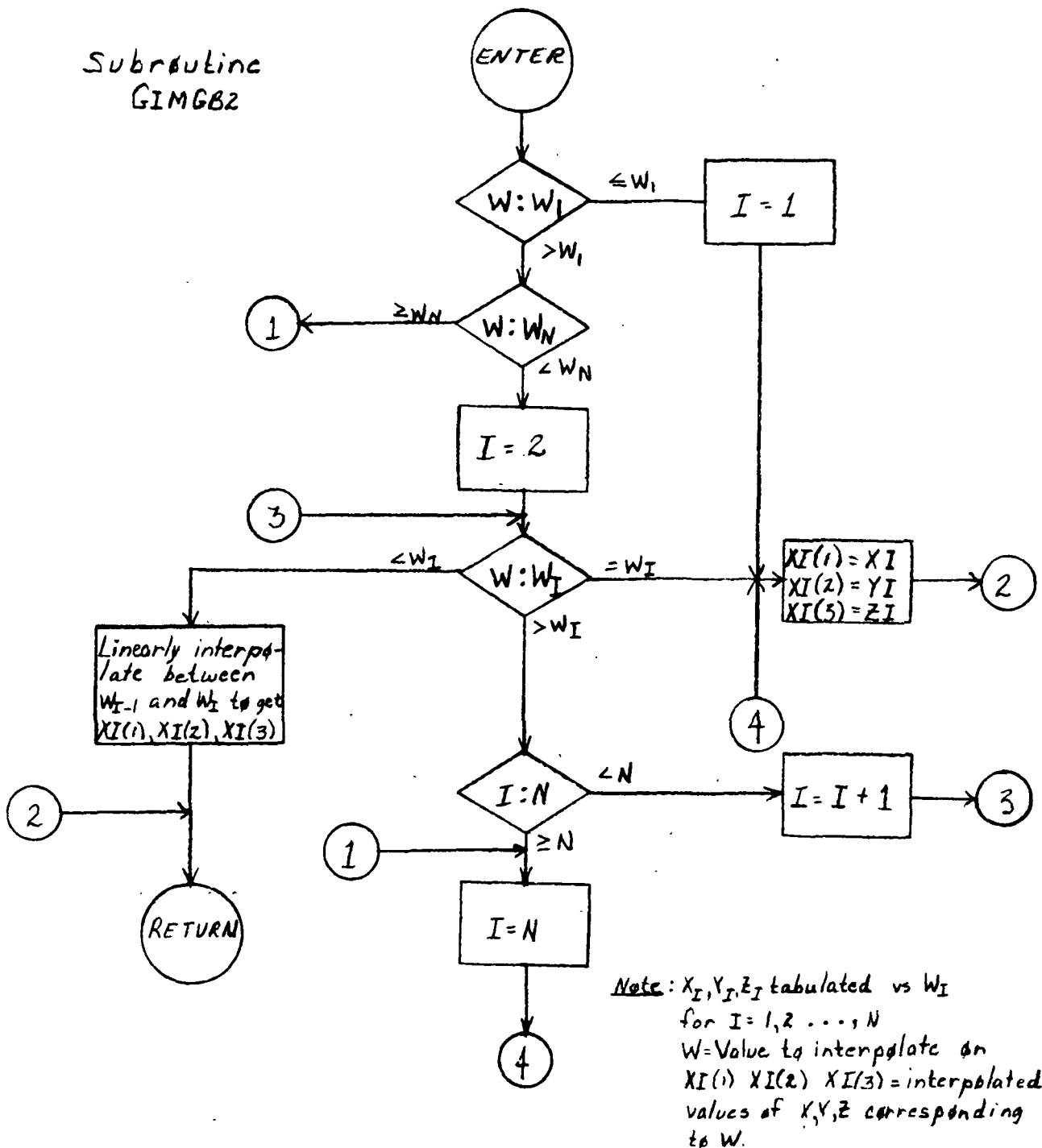


Figure GIMGBL-3. Flowchart

(Sheet 1 of 1)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PCGUID
 DATE 11/14/67 (EORP)
 REV
 PAGE 1 of 10

- a. Symbol — PCGUID
- b. Program Name — Guidance and Time-to-Go Subroutine
- c. Programming Group — Maneuver Control
- d. Programmer — J. Gardner
- e. Language — FORTRAN
- f. Data Tables Used — None
- g. MED's Used — None
- h. Writeup Completion Date — 10/13/67
- i. Flowchart — Figure PCGUID-1

FUNCTION

This subroutine performs the short maneuver logic, computes the thrust direction for guided maneuvers, and computes the engine-off time.

USAGE

- a. Calling Sequence — CALL PCGUID
- b. Inputs — All input is via COMMON
- c. Outputs and Exit — All output is via COMMON
- d. Other Programs Used — DSQRT

METHOD

The method for computing the thrust direction is given in the writeup of PCMRKJ. The time-to-go computations is given by:

$$T_{go} = \frac{W}{\dot{W}} \left[1 - \frac{-\dot{W} V_g / G T_c}{\dot{W}_o} \right] \frac{\dot{W}_{to}}{\dot{W}_o} \Delta T_{to} \text{ where}$$

- W = current weight;
- w = current weight loss rate;
- v_g = |v_g| = velocity to be gained;
- G = gravitational acceleration;
- T_c = current thrust level;
- w_o = full load weight loss rate;
- w_{to} = weight loss rate of tailoff based on full load rate loss rate;
- ΔT_{to} = ΔT of tailoff.

Approval	Approval
<i>J. Gardner</i>	<i>S.E. James</i>

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PCGUID
DATE 11/14/67 (EOR)
REV
PAGE 2 of 10

The T_{go} calculation requires two successive computation cycles at essentially the same thrust level for accurate performance. The Descent Propulsion System (DPS) engine-off logic for short maneuvers is complicated by the DPS start sequence. The DPS start sequence after ullage is ten percent thrust for the first twenty-six seconds at which time the DPS throttle is normally increased to its maximum setting for the duration of the maneuver. Before integration is begun through the maneuver, some short maneuver tests are made.

For the Ascent Propulsion System (APS) and the Service Propulsion System (SPS) engine choice, the short maneuver logic differs only in the constants used. The v_g is biased by an assumed seven-second, two jet ullage. If the v_g is less than that obtained in one second, the time to go is computed using the minimum impulse curve. For a v_g greater than that obtained in one second, a test is made to see if it can be obtained in five additional seconds. The v_g for the five additional seconds is computed using the average thrust level and average weight loss rate with the weight compensated by the weight loss rate. If the v_g cannot be gained in five additional seconds a time to go is computed.

$$T_{go} = 1 + 5 v_{go} / v_{gs} \text{ where}$$

v_{go} = Velocity to be gained after compensating for ullage and one-second burn

v_{g5} = Velocity to be gained for a five-second burn

When the time to go is less than six seconds the initial direction at main engine on is held throughout the burn of T_{go} seconds.

For the Reaction Control System (RCS) engine choice a time to go is computed using the designated thrust level and weight loss rate. If this value is less than six seconds, the initial direction is held throughout the burn of T_{go} seconds.

For the DPS engine choice a time to go is computed using the v_g biased by an assumed seven-second, two jet ullage and the ten percent values for the thrust level and weight loss rate. If this time is less than six seconds, the direction computed at main engine on is held throughout the burn of T_{go} seconds. If this time is greater than six seconds, a test is made to see if it is less than eighty-three seconds. If it is less than eighty-three seconds, the ten percent level is held throughout the burn.

For the DPS engine choice the short maneuver logic applies to all types of maneuvers. For the RCS, SPS, and APS engine choices the short maneuver logic applies only to the external ΔV maneuver.

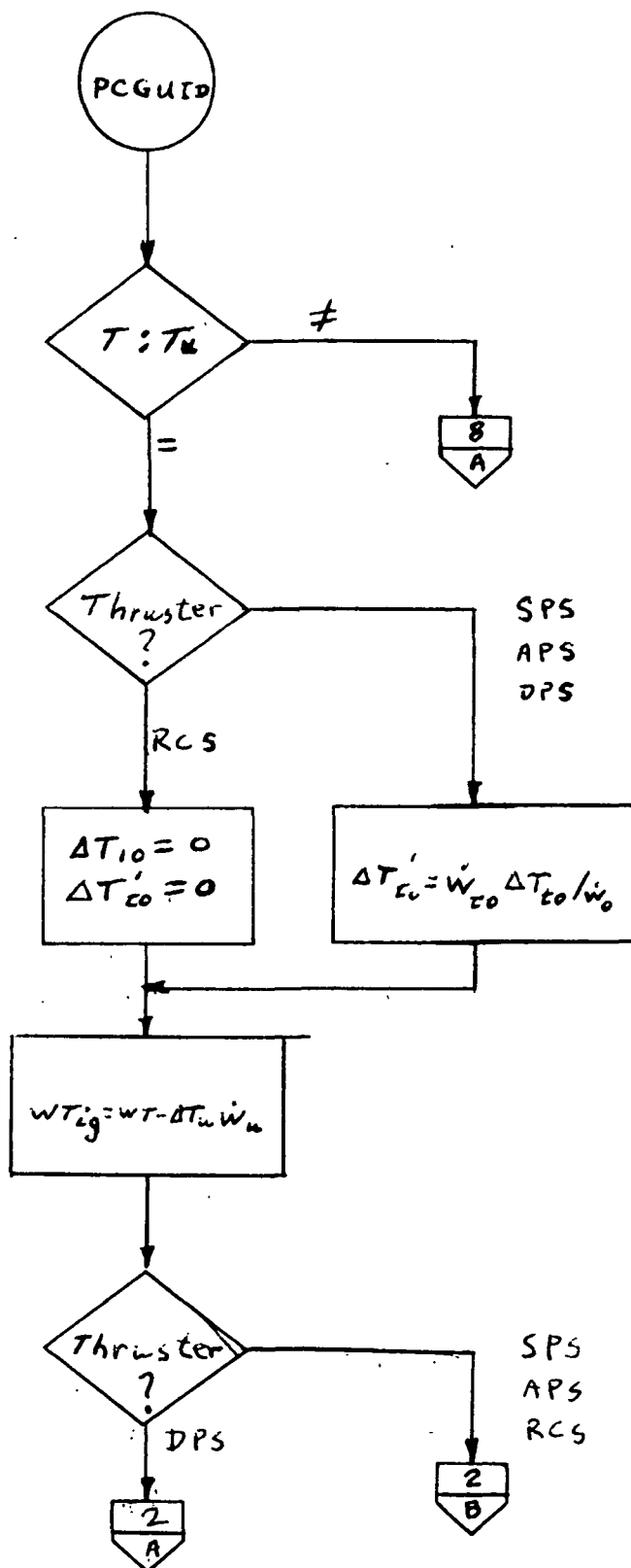


Figure PCGUID-1. Flowchart

(Sheet 1 of 8)

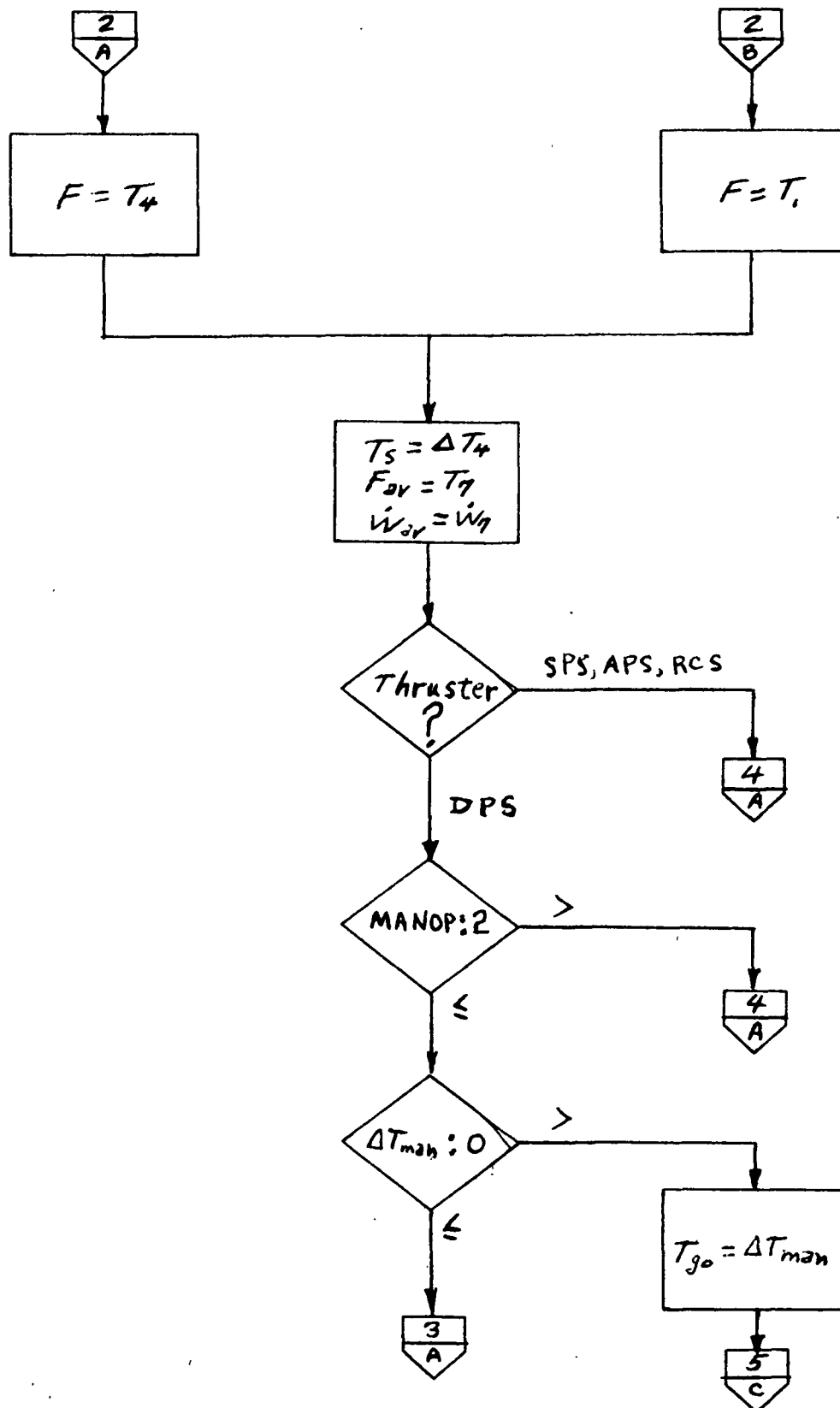


Figure PCGUID-1. Flowchart

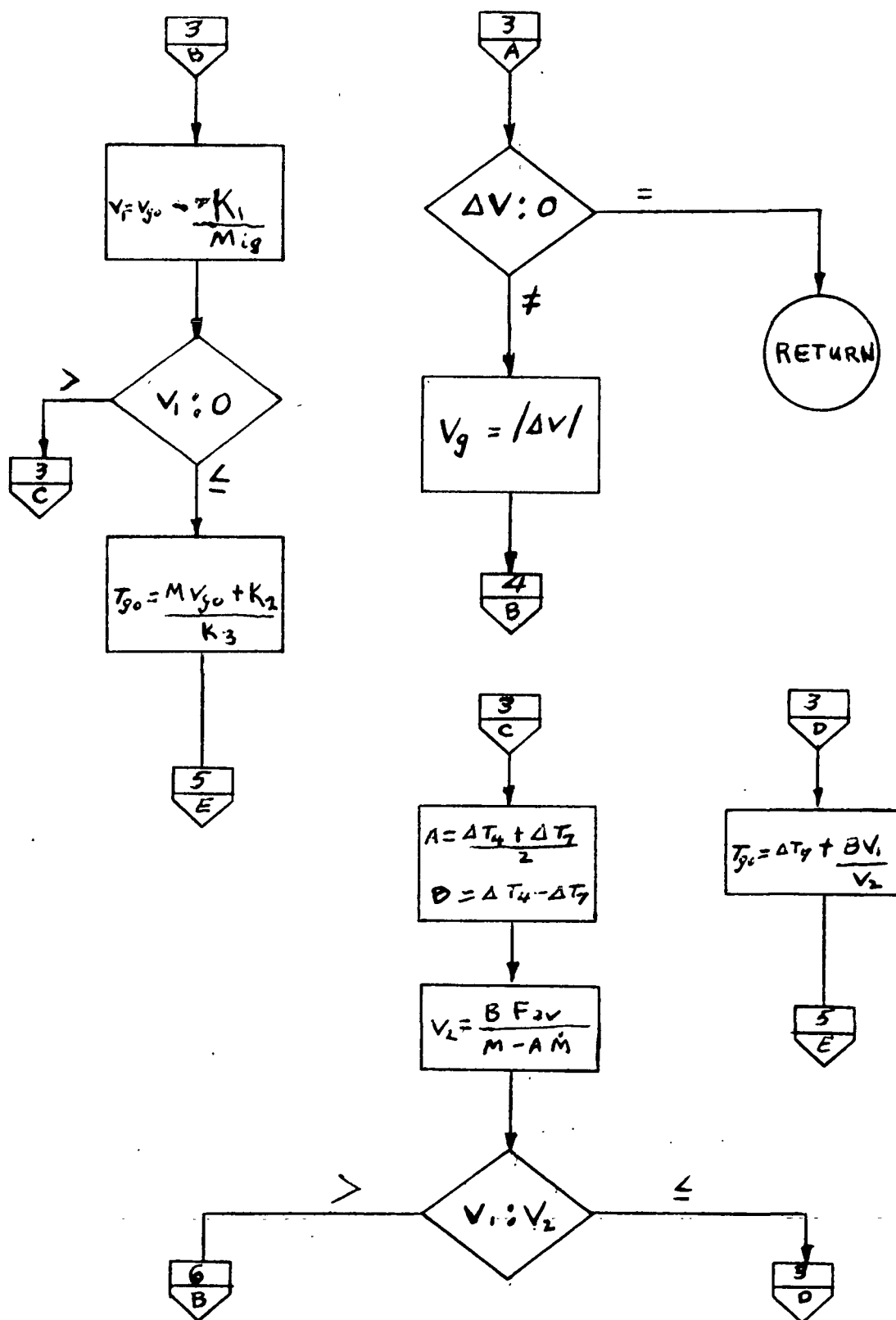


Figure PCGUID-1. Flowchart

(Sheet 3 of 8)

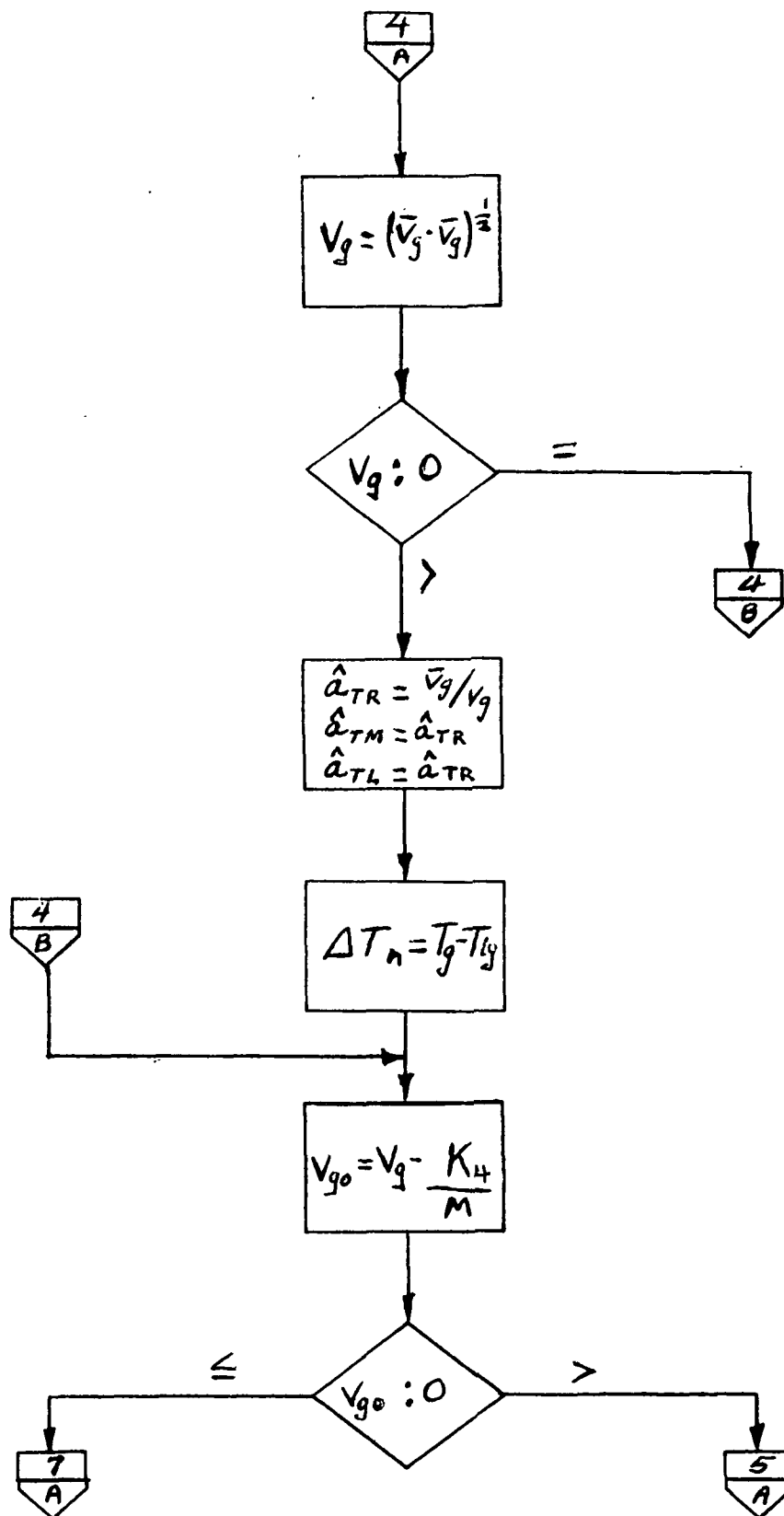


Figure PCGUID-1. Flowchart

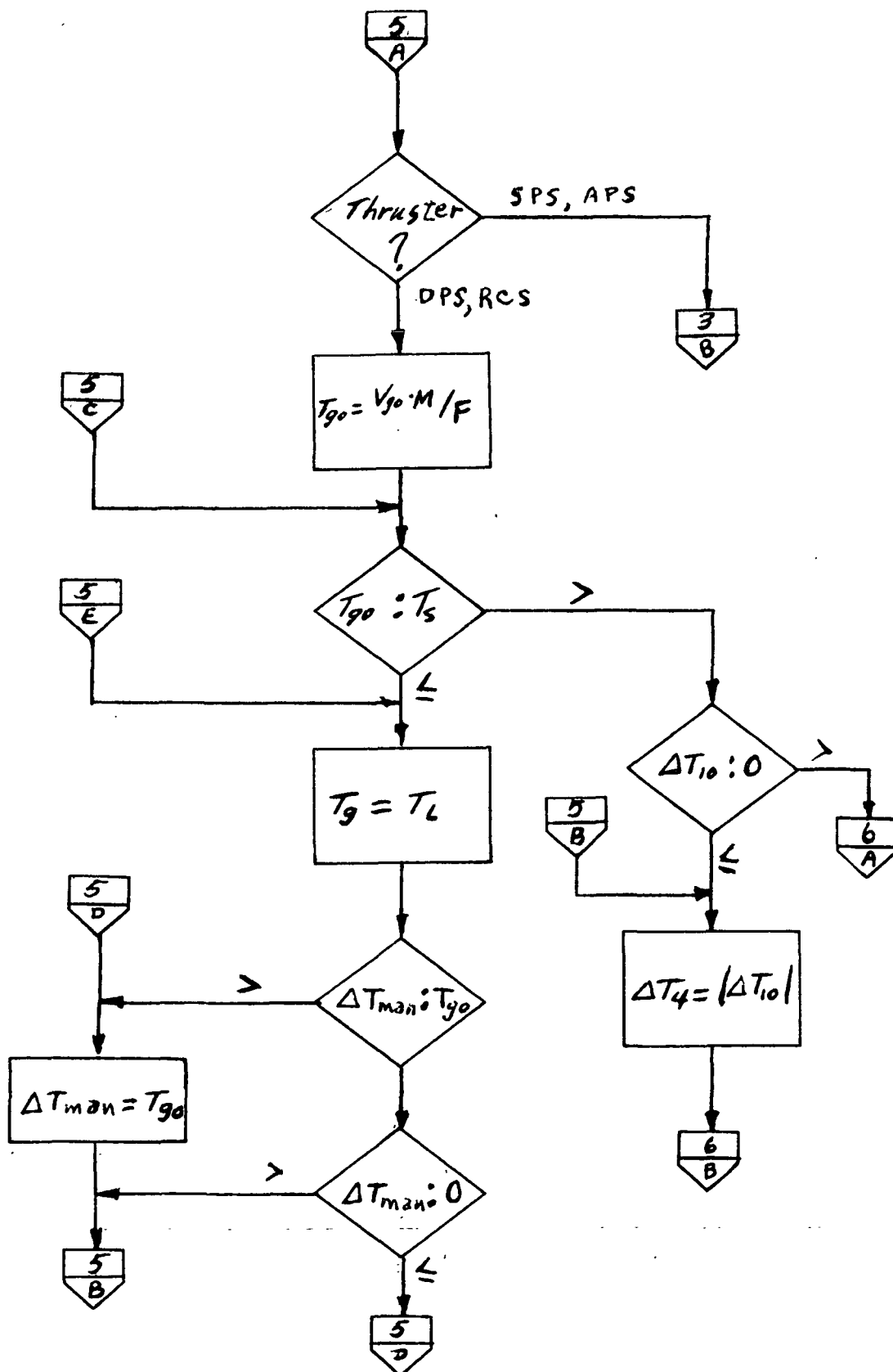


Figure PCGUID-1. Flowchart

(Sheet 5 of 8)

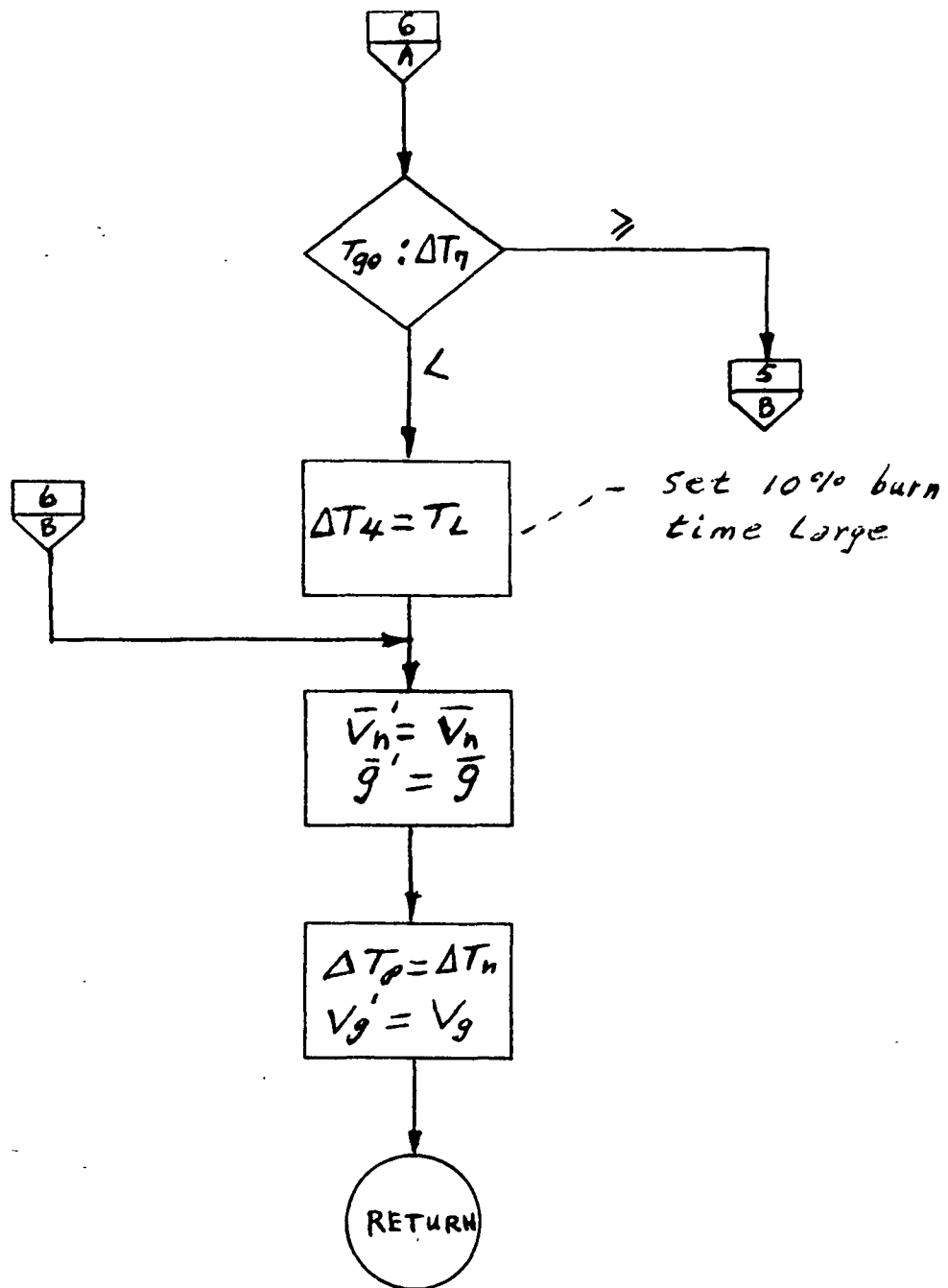


Figure PCGUID-1. Flowchart

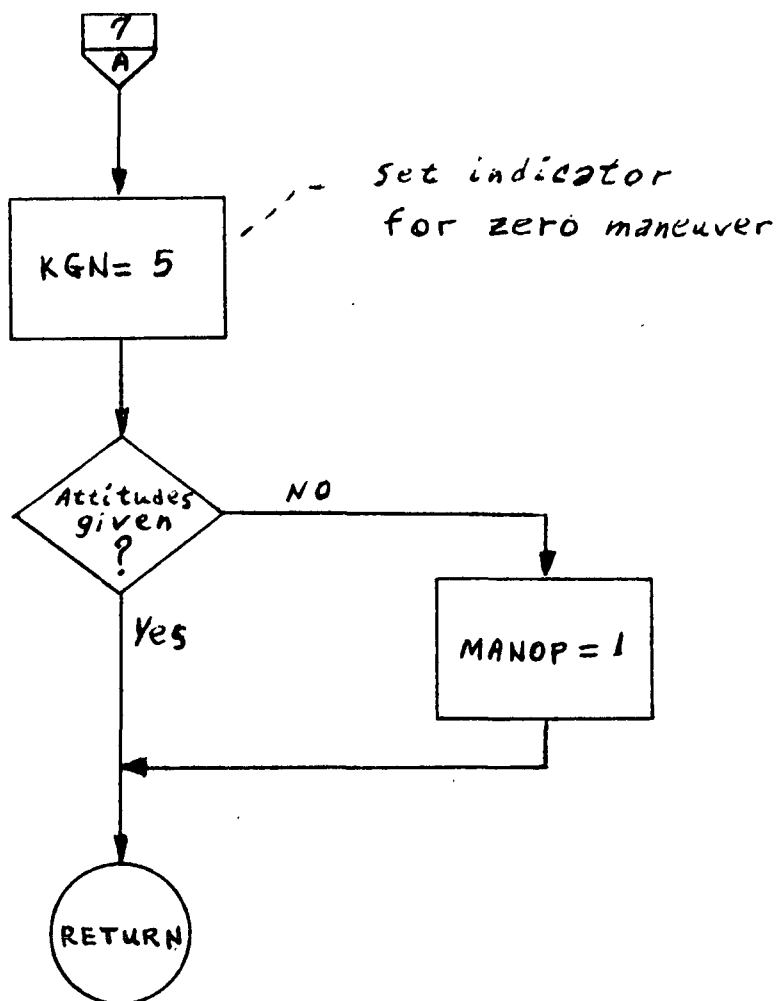


Figure PCGUID-1. Flowchart

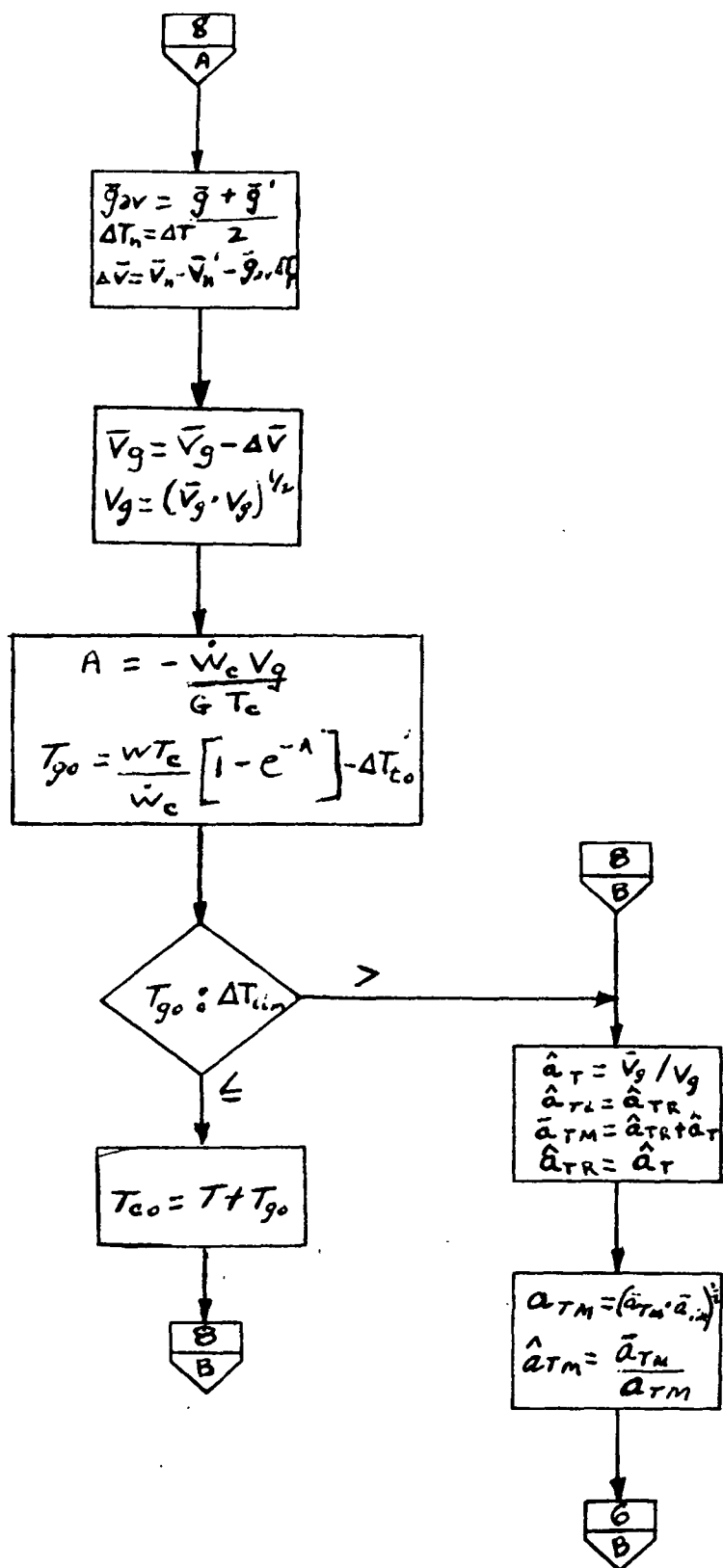


Figure PCGUID-1. Flowchart

(Sheet 8 of 8)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PCINIT
DATE 11/14/67 (EORP)
REV
PAGE 1 of 4

- a. Symbol — PCINIT
- b. Program Name — System Parameter Initialization Routines
- c. Programming Group — Maneuver Control
- d. Programmer(s) — J. Gardner
- e. Language — FORTRAN
- f. Data Tables Used — None
- g. MED's Used — None
- h. Writeup Completion Date — 10/13/67
- i. Flowchart — Figure PCINIT-1

FUNCTION

This routine initializes the thrust constants as a function of the thruster requested.

USAGE

- a. Calling Sequence — CALL PCINIT (TAB, MANOP, ITYPE) where TAB is the table to store the appropriate constants, MANOP is the maneuver type, and ITYPE is the thruster indicator.
- b. Inputs —
- c. Outputs and Exit — All output is stored in the table specified in calling sequence
- d. Other Programs Used — None

METHOD

The method is reflected in the flowchart.

Approval	Approval
<i>J Gardner</i>	<i>SE James</i>

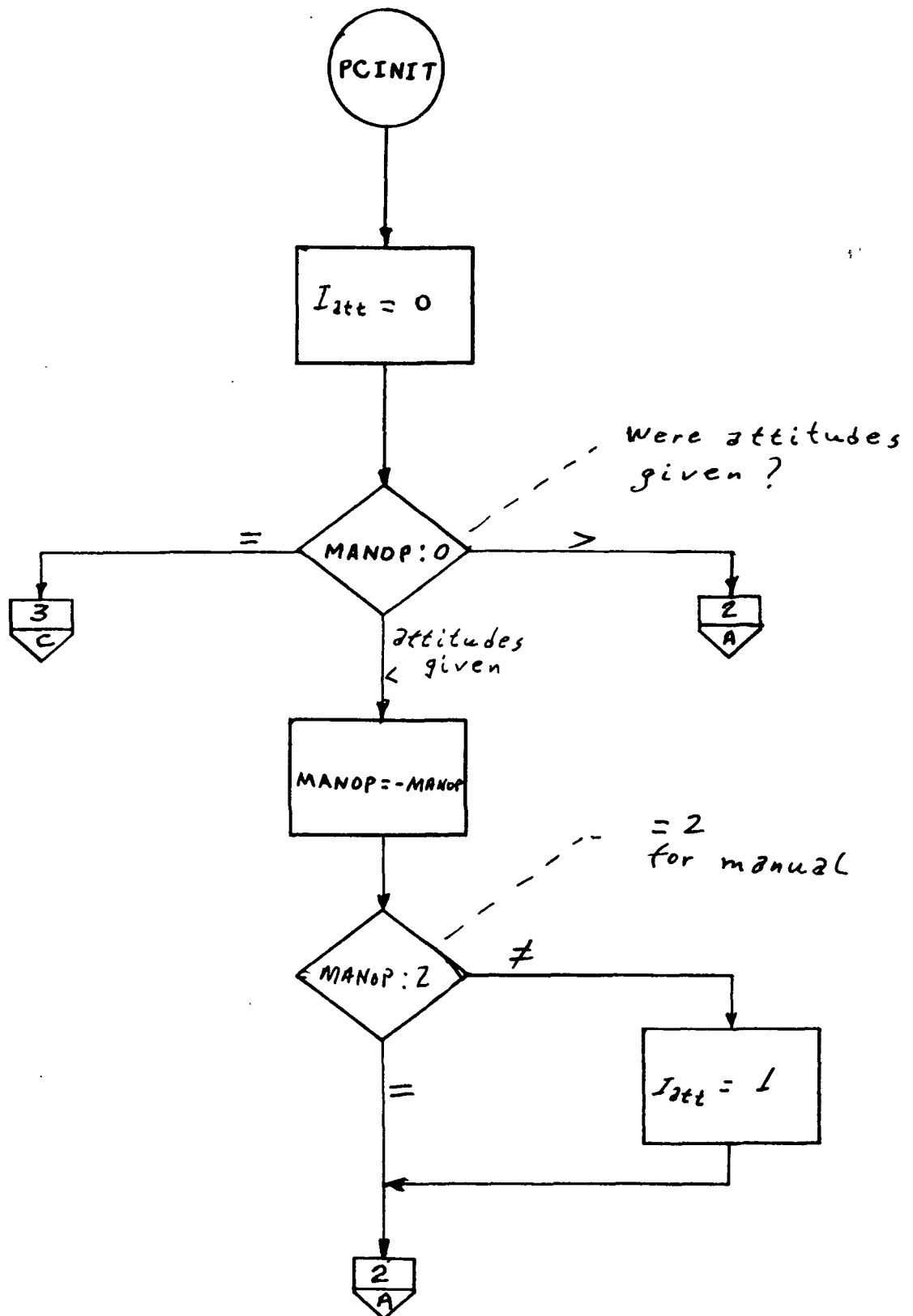


Figure PCINIT-1. Flowchart

(Sheet 1 of 3)

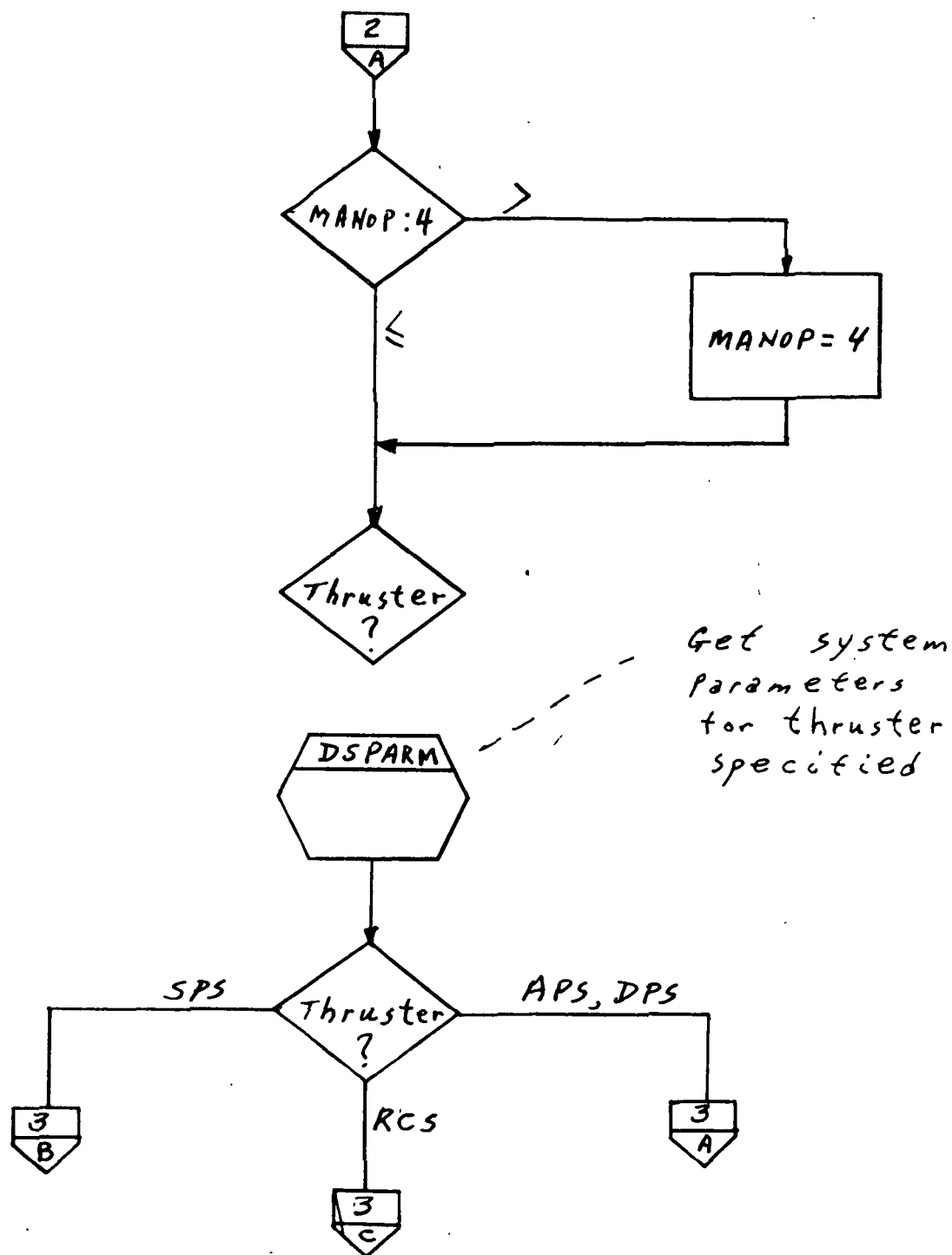


Figure PCINIT-1. Flowchart

(Sheet 2 of 3)

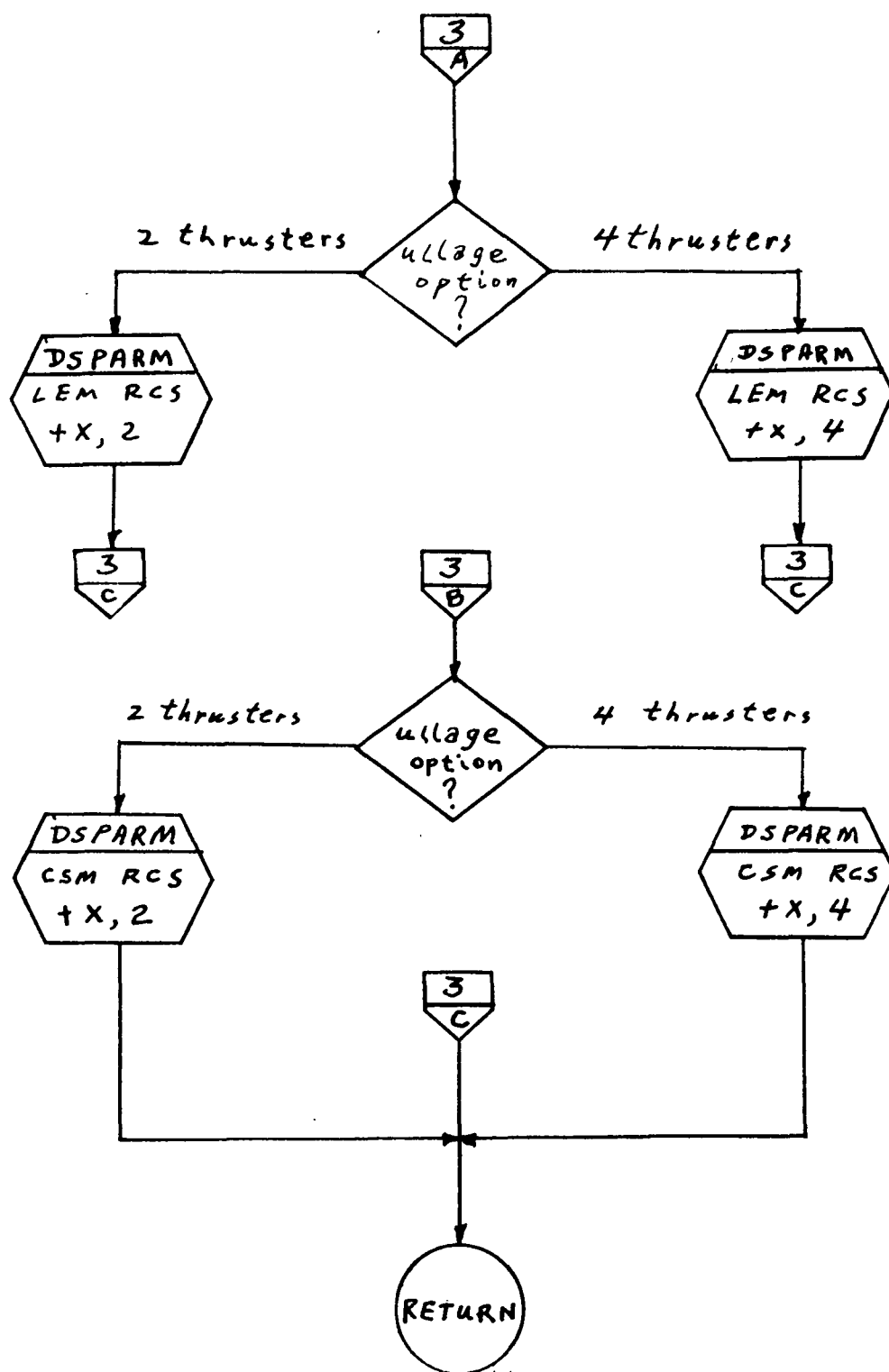


Figure PCINIT-1. Flowchart

(Sheet 3 of 3)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PCRDD
 DATE 11/14/67 (EORP)
 REV
 PAGE 1 of 10

- a. Symbol — PCRDD
- b. Program Name — Derivative Control Subroutine
- c. Programming Group — Maneuver Control
- d. Programmer(s) — J. Gardner
- e. Language — FORTRAN
- f. Data Tables Used — None
- g. MED's Used — None
- h. Writeup Completion Date — 10/13/67
- i. Flowchart — Figure PCRDD-1

FUNCTION

This program computes the second derivative of $\bar{r}$ with respect to time.

USAGE

- a. Calling Sequence — CALL PCRDD
- b. Inputs — All input is via COMMON.
- c. Outputs and Exit — All output is via COMMON.
- d. Other Programs Used — PCGUID, GIMGBL, GLMRTM, GLFDEN, DLOG, DCOS, DSQRT

METHOD

This routine computes the acceleration vector given by:

$$\ddot{\bar{r}} = -\frac{\partial V}{\partial \bar{r}} + \bar{F}$$

The method is given in the writeup of PMMRKJ and is reflected in the flowchart.

Approval	Approval
<i>J Gardner</i>	<i>SE James</i>

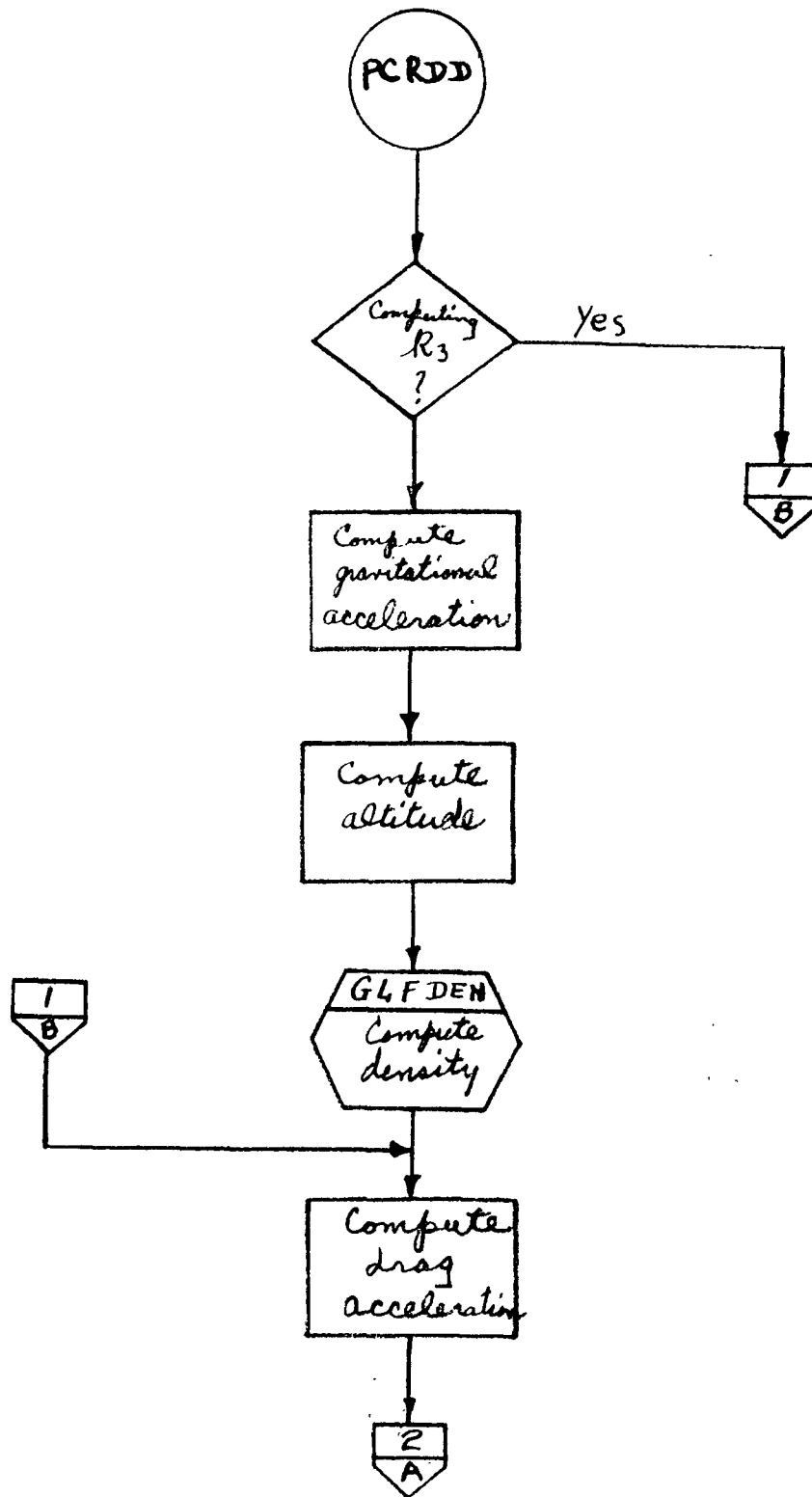


Figure PCRDD-1. Flowchart

(Sheet 1 of 9)

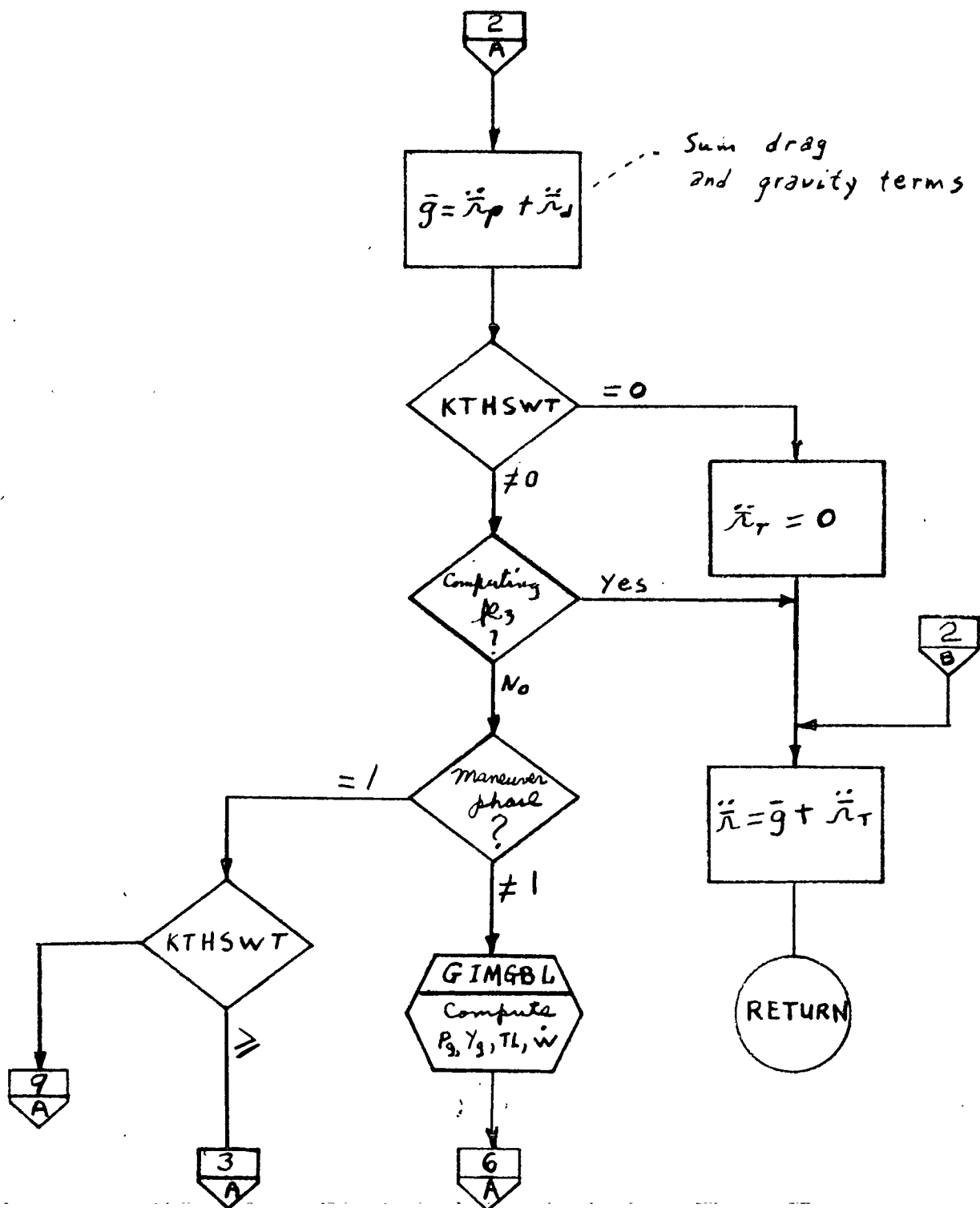


Figure PCRDD-1. Flowchart

(Sheet 2 of 9)

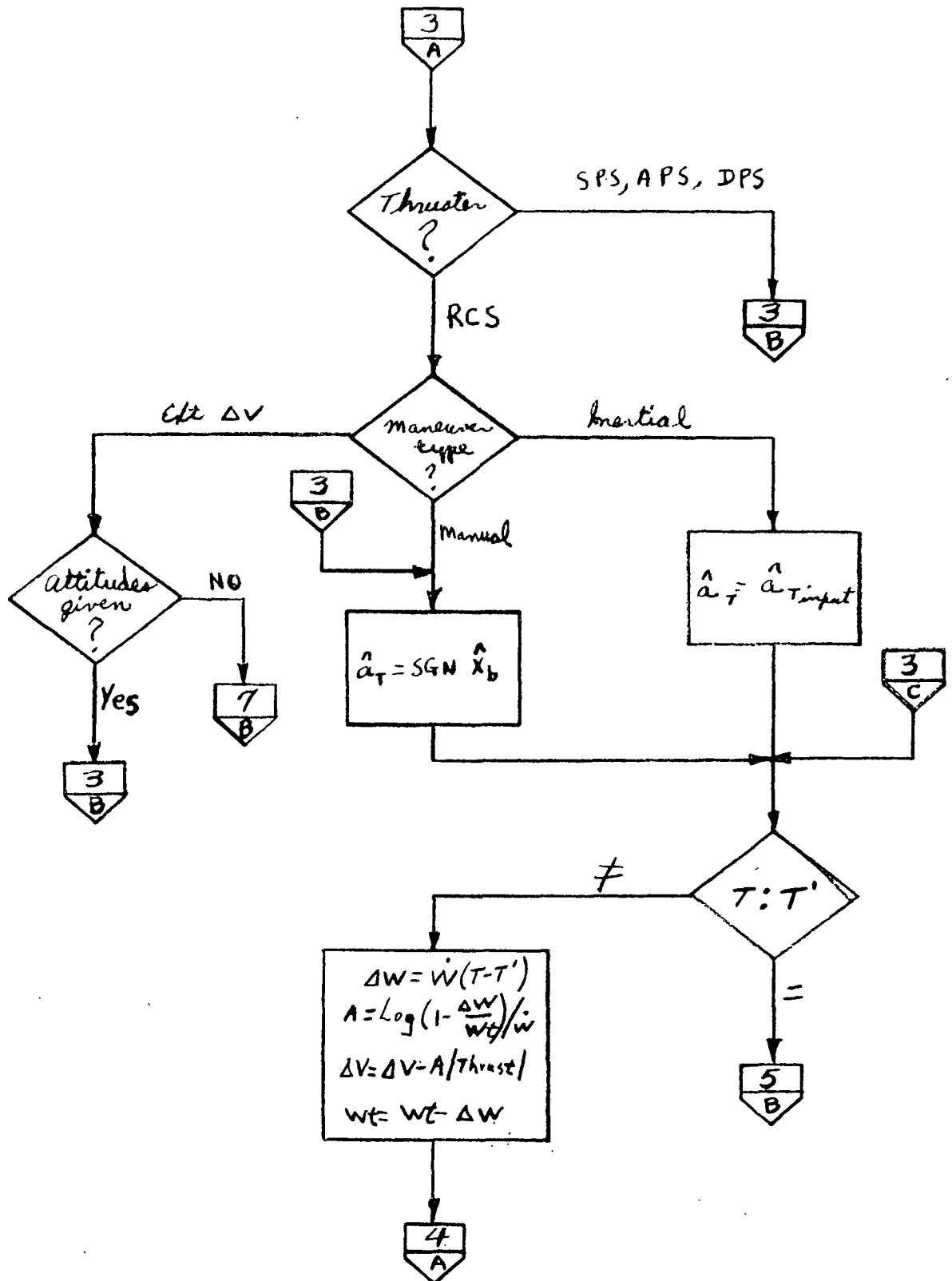


Figure PCRDD-1. Flowchart

(Sheet 3 of 9)

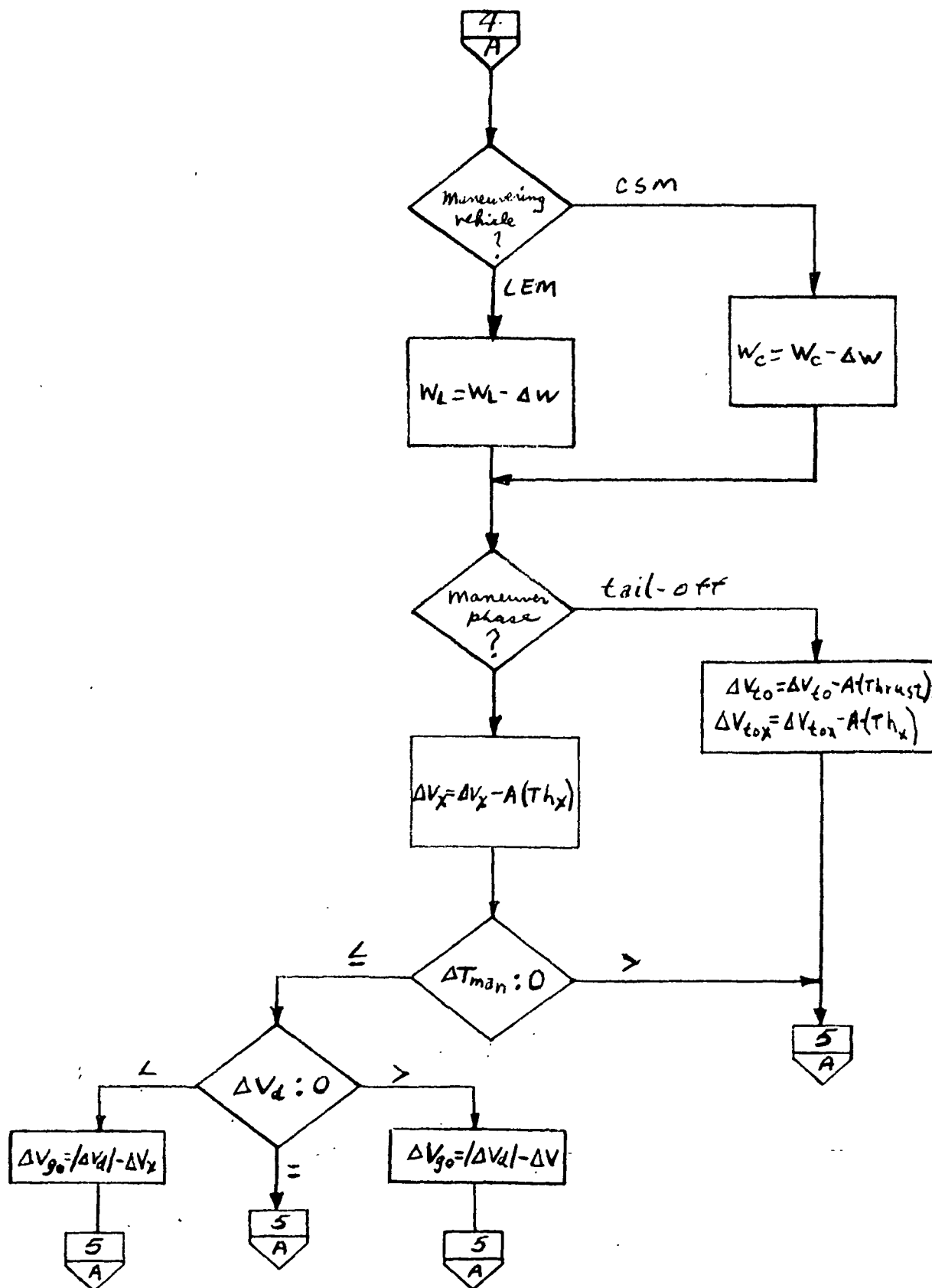


Figure PCRDD-1. Flowchart

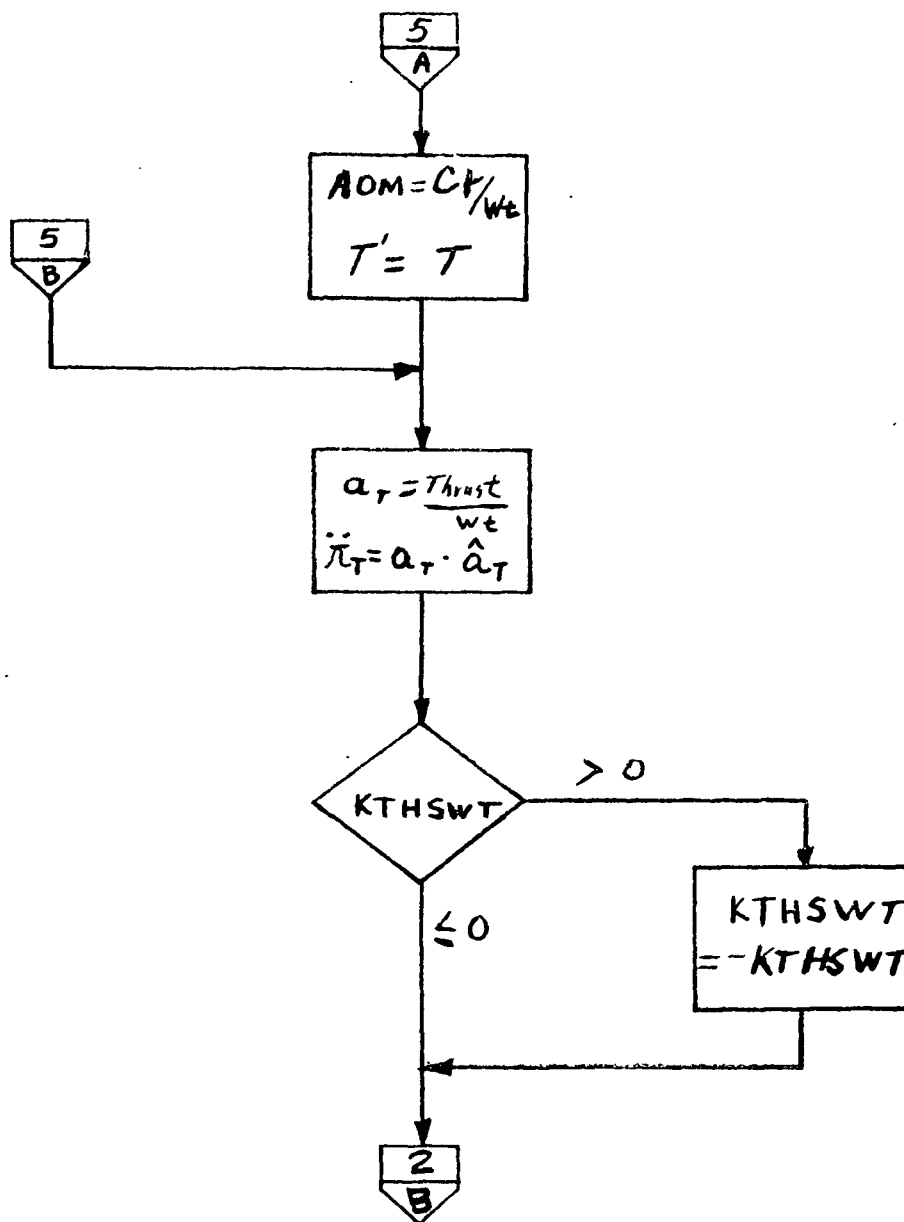


Figure PCRDD-1. Flowchart

(Sheet 5 of 9)

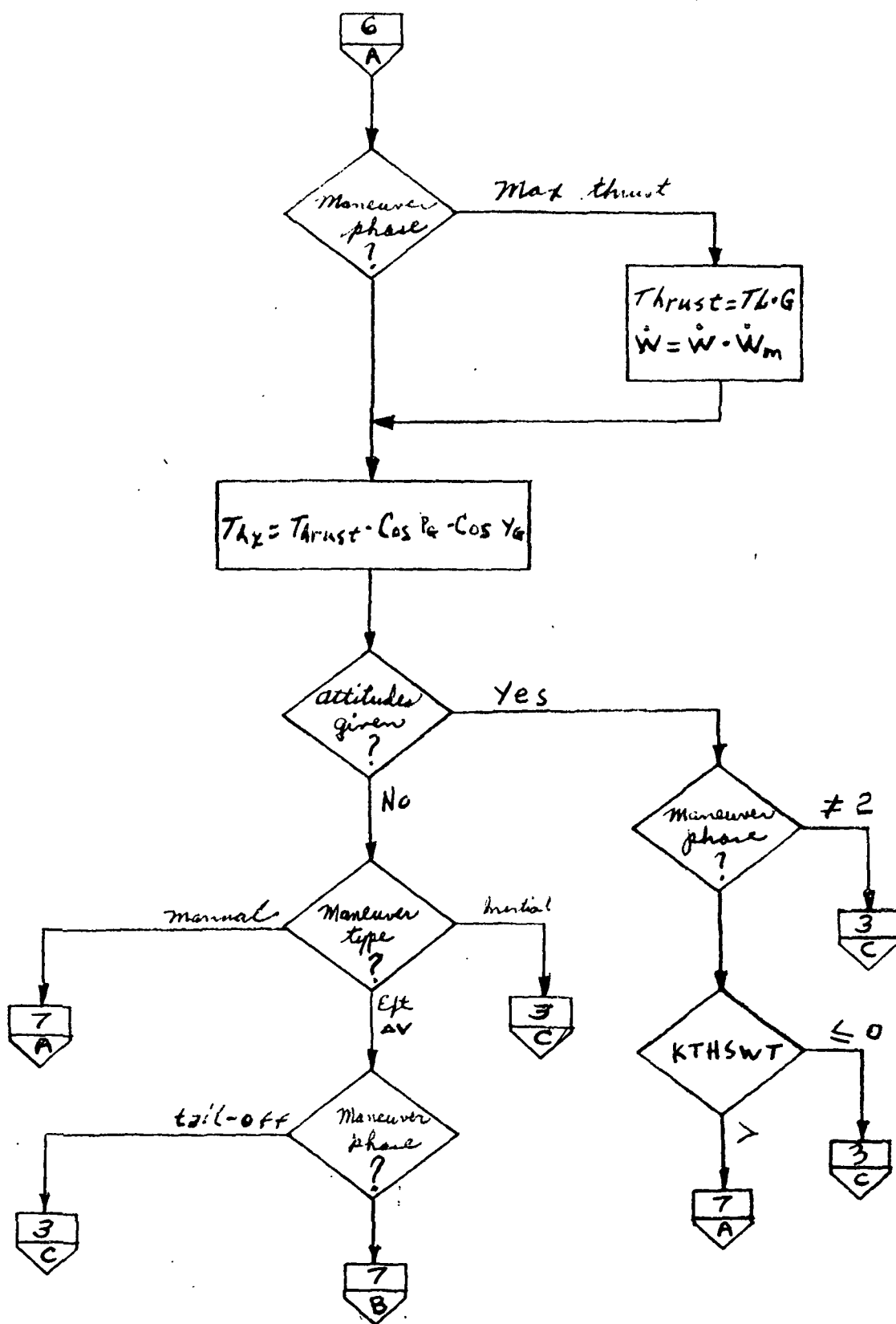


Figure PCRDD-1. Flowchart

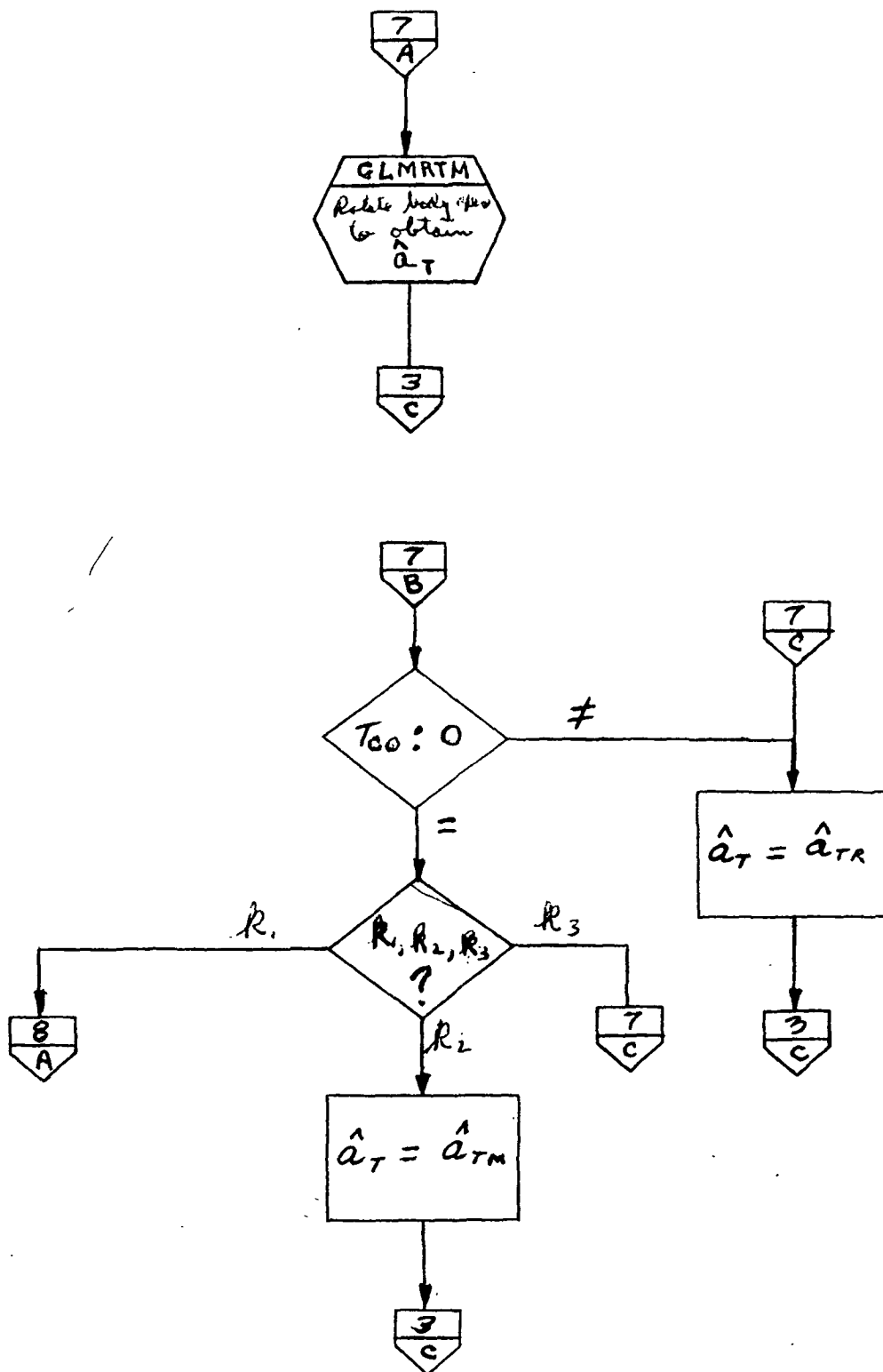


Figure PCRDD-1. Flowchart

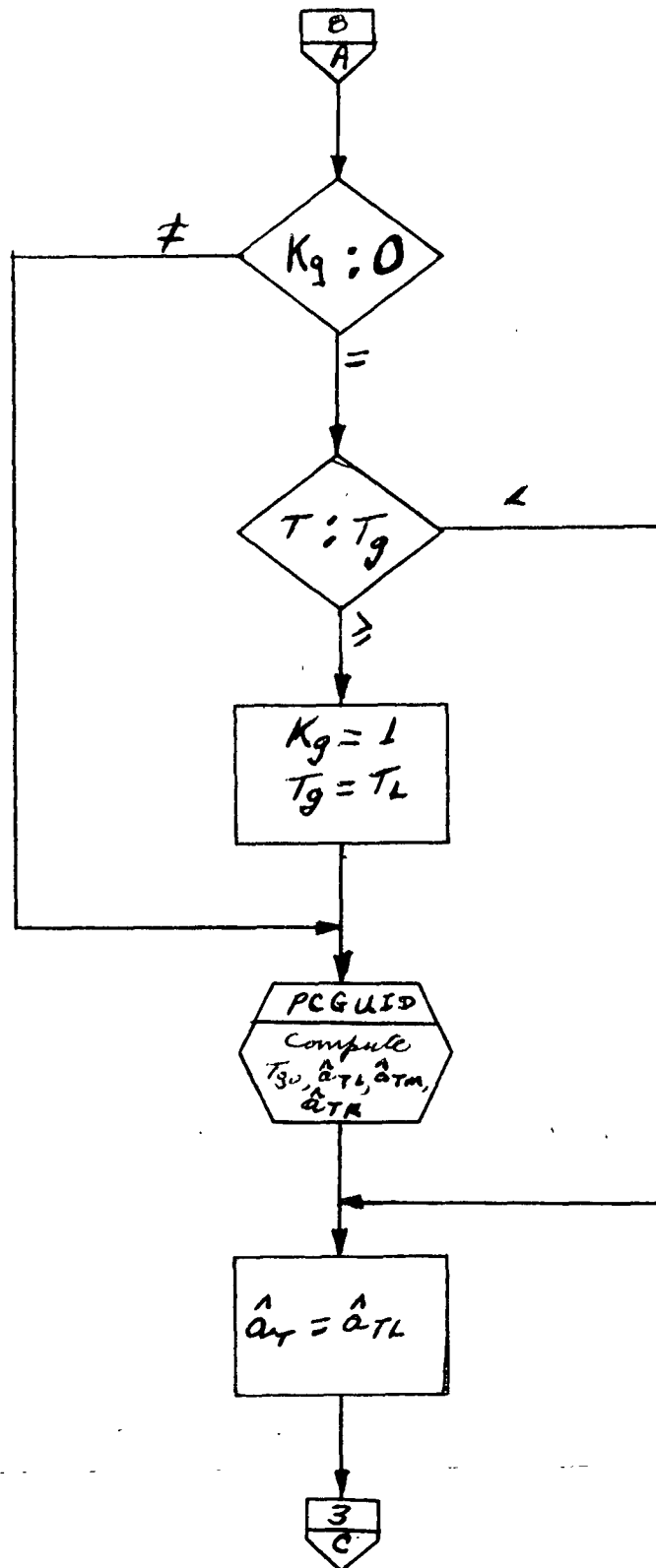


Figure PCRDD-1. Flowchart

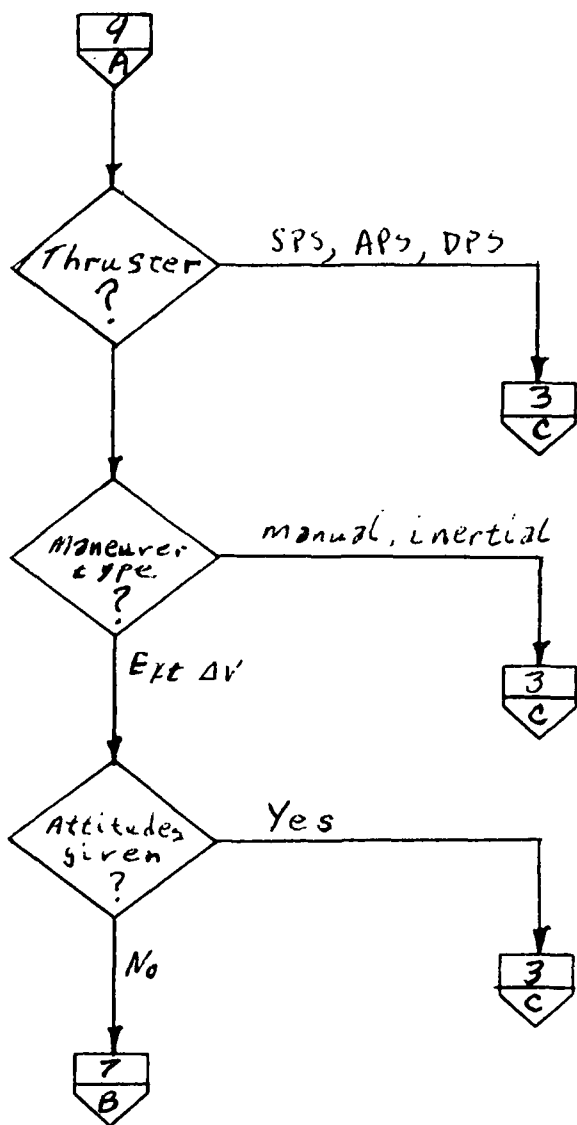


Figure PCRDD-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PCRUNG

DATE 11/14/67 (EORP)

REV

PAGE 1 of 9

- a. Symbol — PCRUNG
- b. Program Name — Runge Kutta Integration Subroutine
- c. Programming Group — Maneuver Control
- d. Programmer(s) — J. Gardner
- e. Language — FORTRAN
- f. Data Tables Used — None
- g. MED's Used — None
- h. Writeup Completion Date — 10/13/67
- i. Flowchart — Figure PCRUNG-1

FUNCTION

This routine is used to integrate $\ddot{r}$ and $\ddot{v}$ from "time t" to "time t plus Δt ".

USAGE

- a. Calling Sequence — CALL PCRUNG (E, W) where E is ephemeris table and W is weight table
- b. Inputs — All input is via COMMON.
- c. Outputs and Exit — All output is via COMMON and ephemeris tables.
- d. Other Programs Used — PCRDD, GLFSAT
- e. Error — If the kickout altitude is reached, the error code is set to 1. If the fuel is exhausted, the error code is set to 2.

METHOD

The method is given in the writeup of PMMRKJ and is reflected in the flowchart.

Approval	Approval
<i>J. Gardner</i>	<i>S. E. James</i>

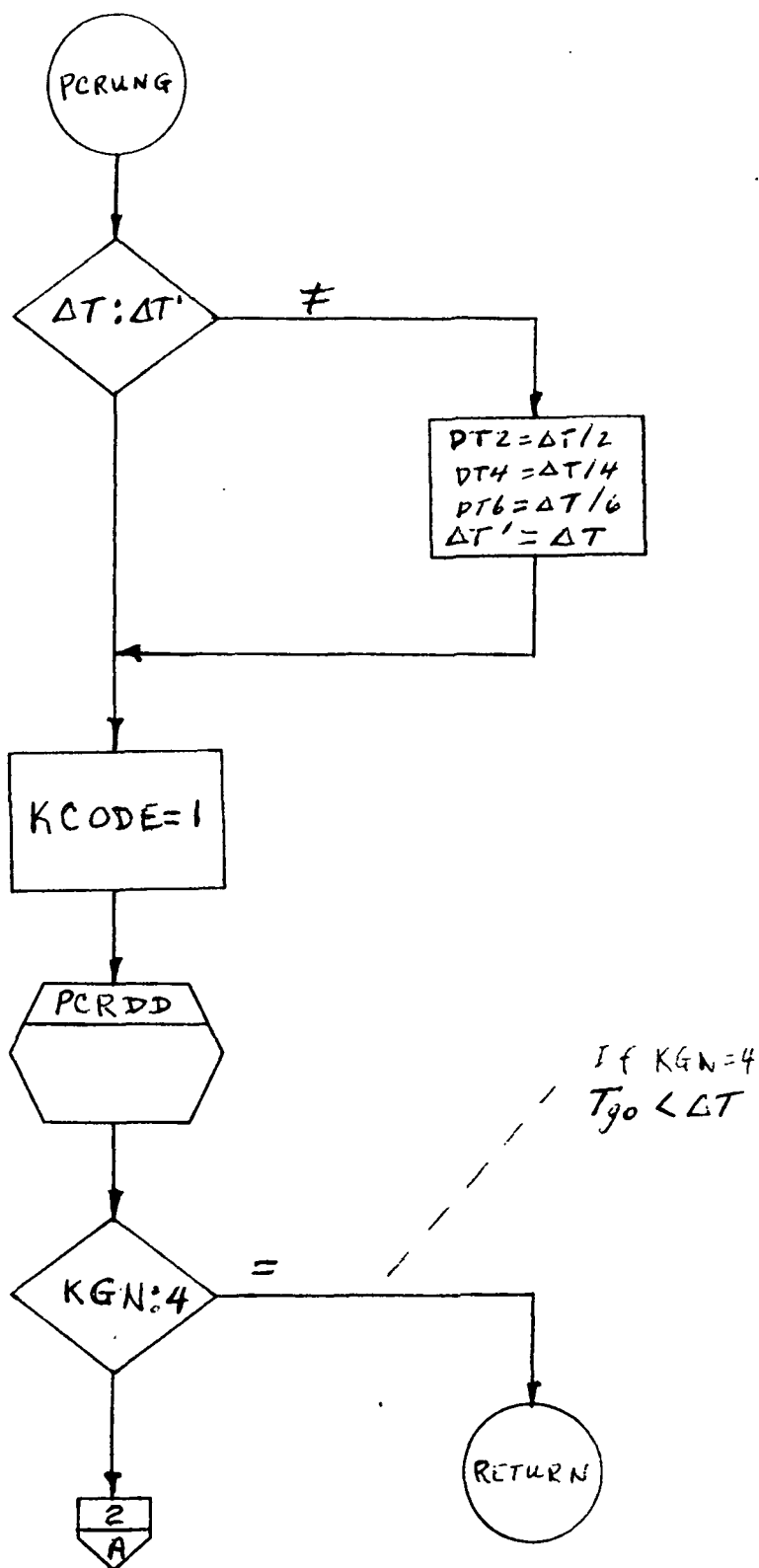


Figure PCRUNG-1. Flowchart

(Sheet 1 of 8)

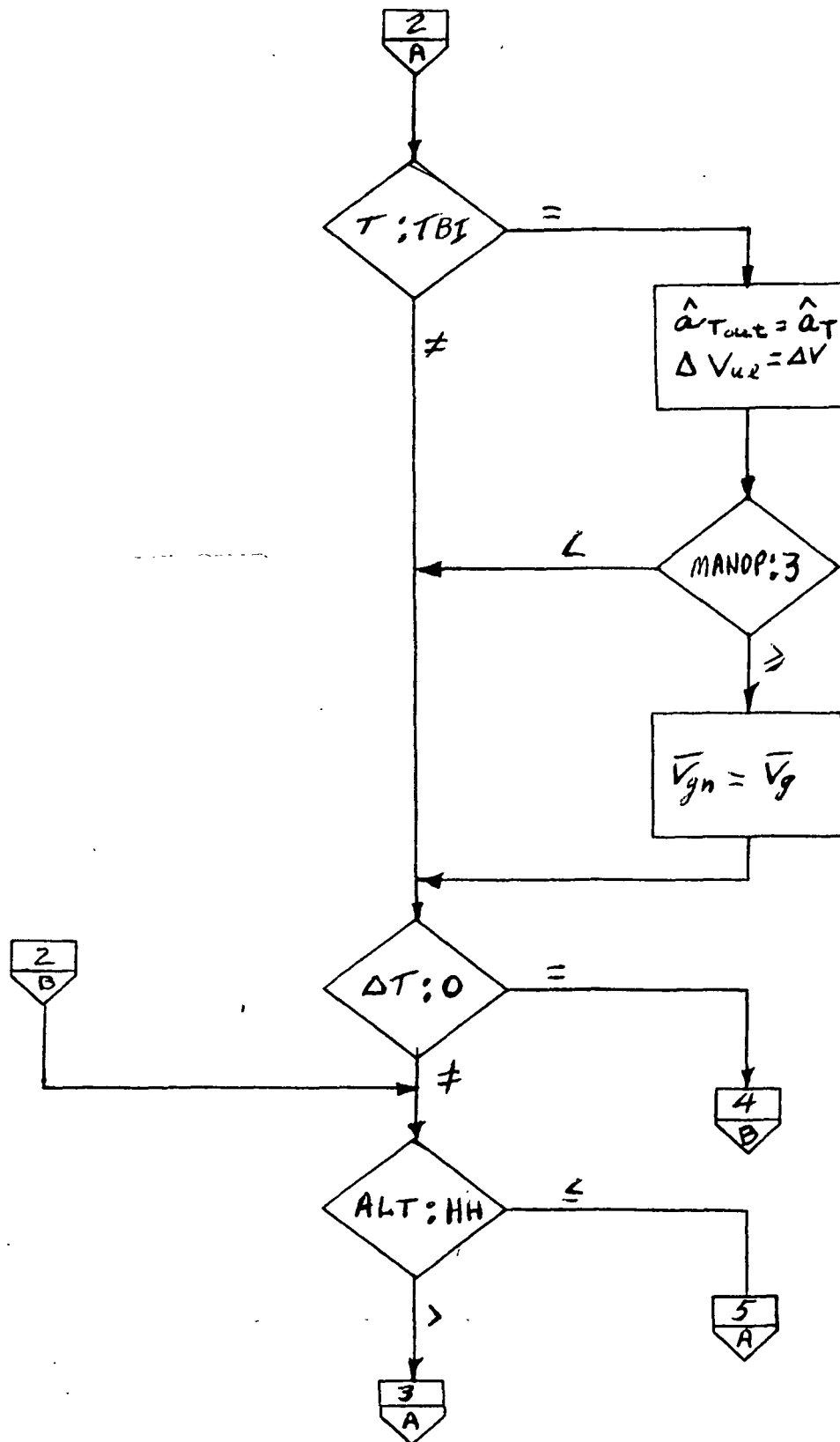


Figure PCRUNG-1. Flowchart

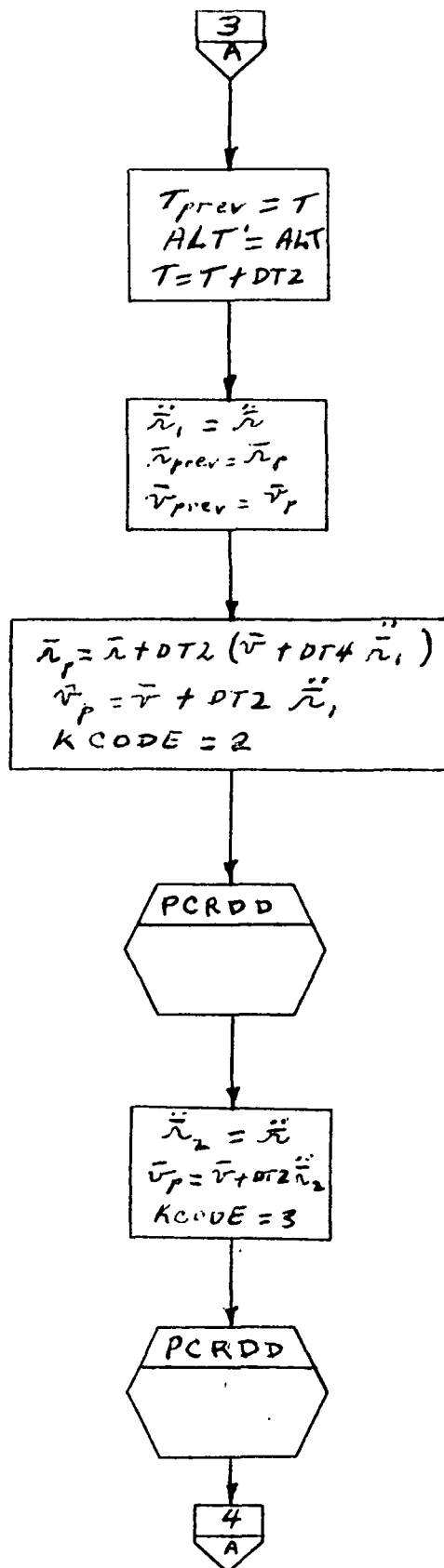


Figure PCRUNG-1. Flowchart

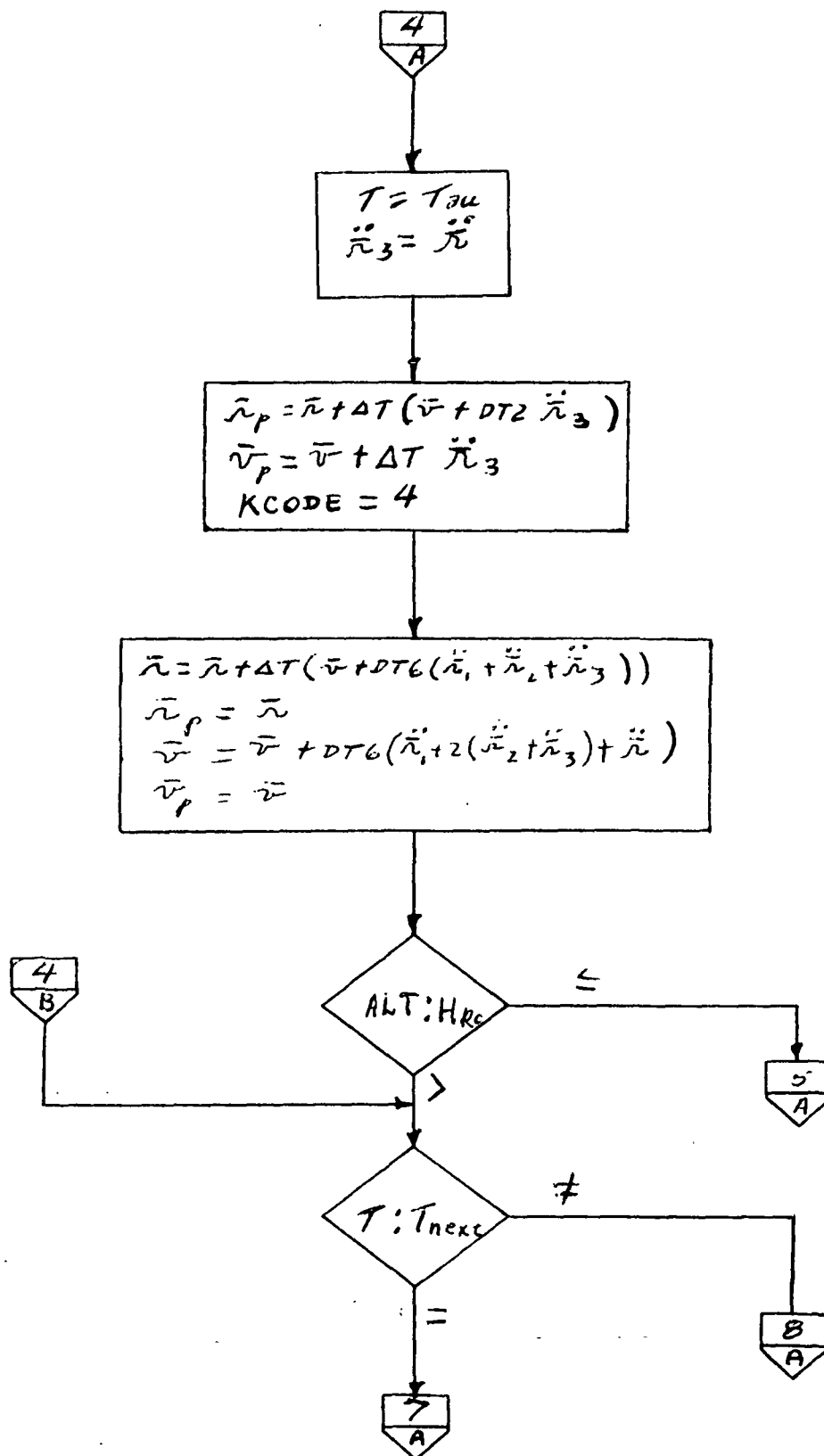


Figure PCRUNG-1. Flowchart

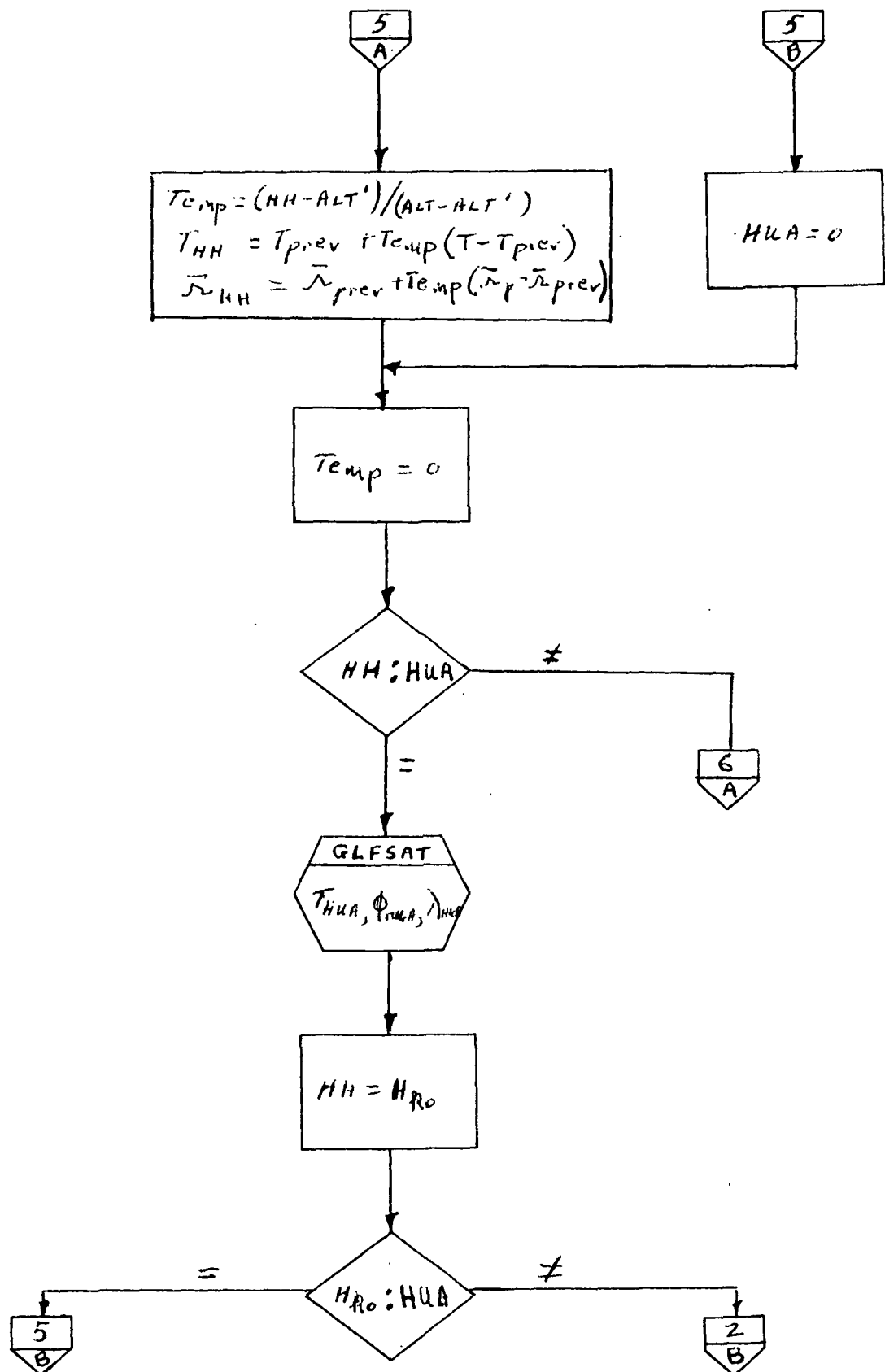


Figure PCRUNG-1. Flowchart

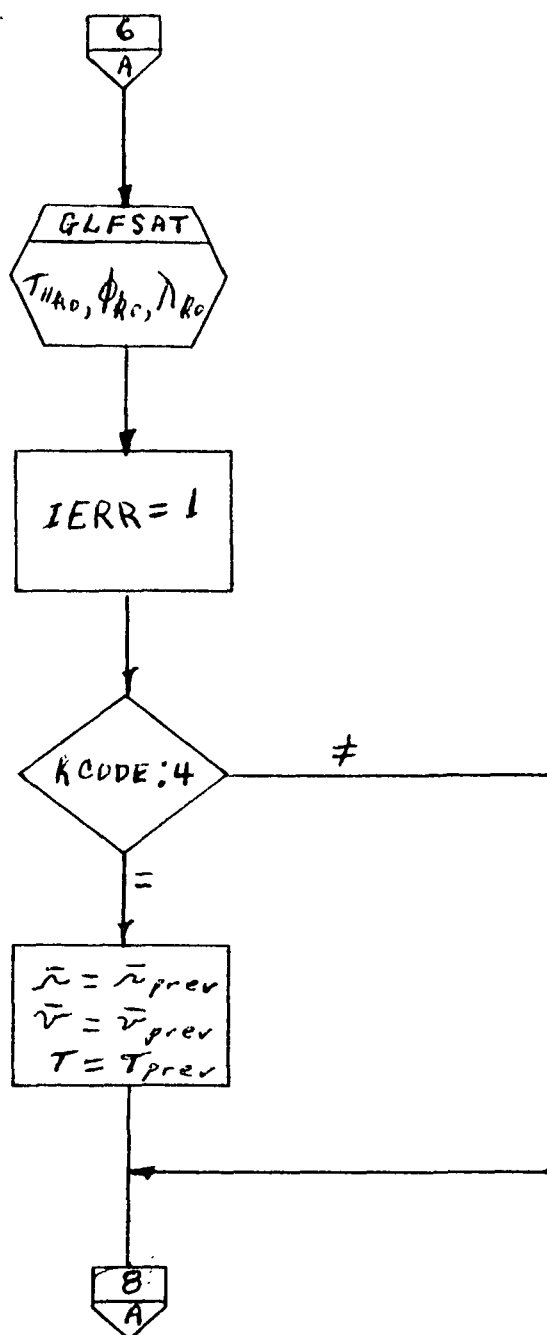


Figure PCRUNG-1. Flowchart

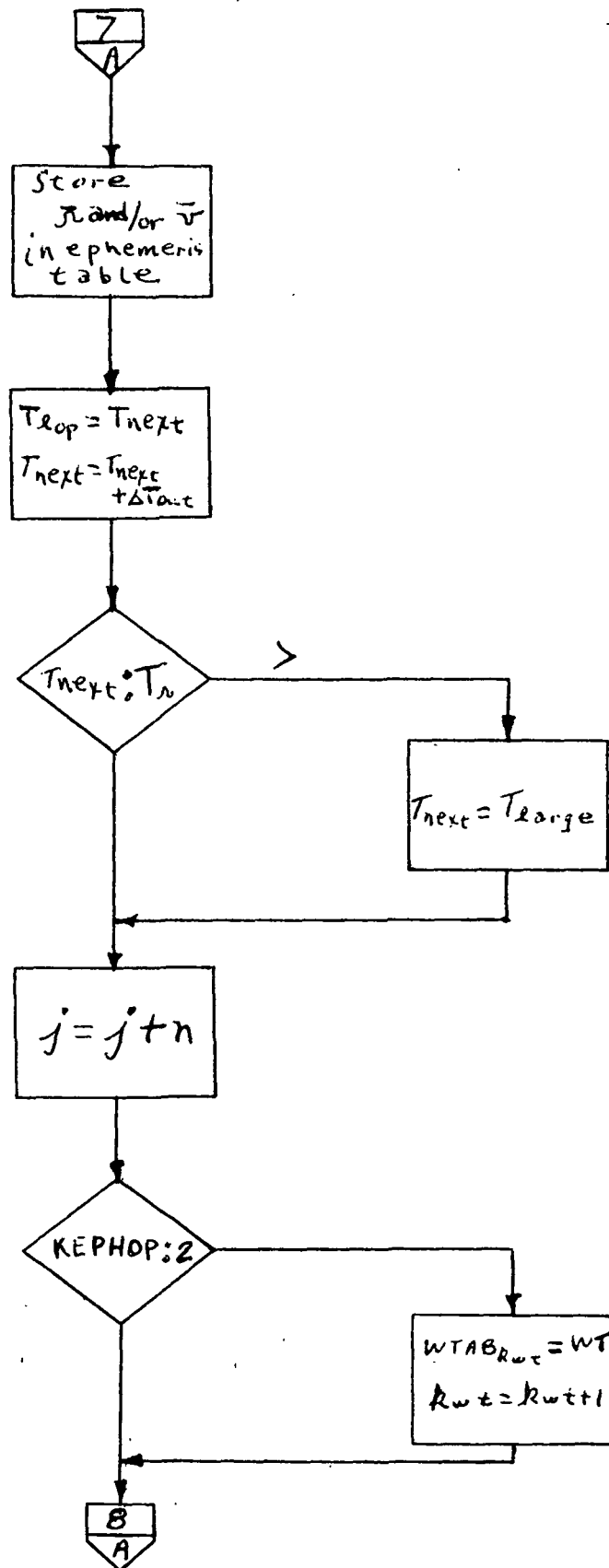


Figure PCRUNG-1. Flowchart

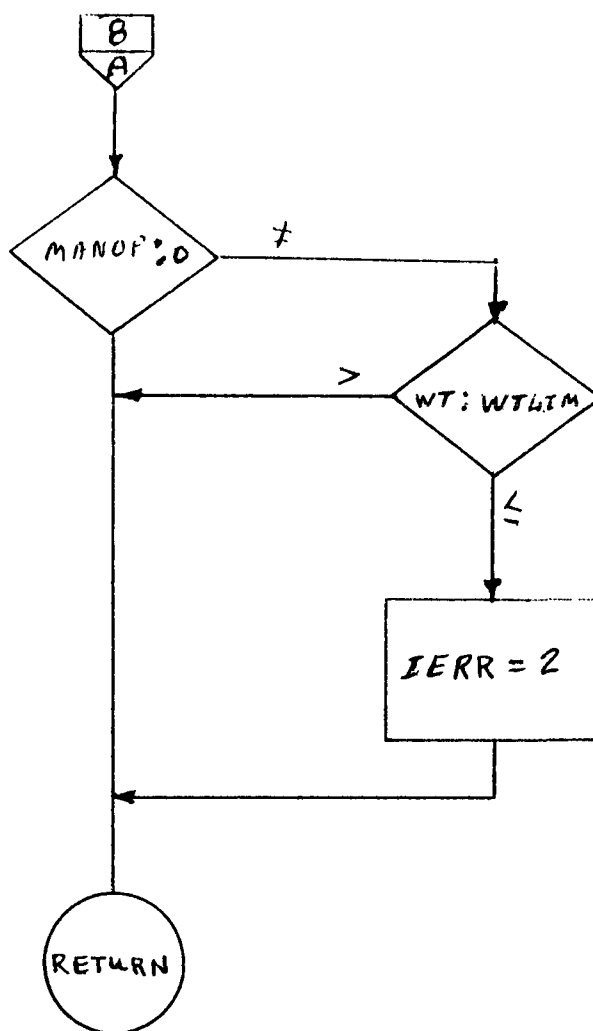


Figure PCRUNG-1. Flowchart

(Sheet 8 of 8)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PIAEDV

DATE 7/31/67(EORP)

REV

PAGE 1 of 3

- a. Symbol — PIAEDV
- b. Program Name — LM Abort Guidance System (AGS) External ΔV Coordinate Transformation Subroutine
- c. Programming Group — Mission Planning - Maneuver Control
- d. Programmer — R.E. Cusey
- e. Language — FORTRAN
- f. Data Tables Used — None
- g. MED's Used — None
- h. Writeup Completion Date — 6/6/67
- i. Flowchart — Figure PIAEDV-1

FUNCTION

Given a LM AGS external $\Delta \bar{V}$ vector, this subroutine transforms the vector from the RTCC Inertial Coordinate System to the Command and Display Coordinate System or from the Command and Display Coordinate System to the RTCC Inertial Coordinate System.

USAGE

- a. Calling Sequence — CALL PIAEDV (DV, RC, RL, I)
- b. Inputs —
 - DV - components of external $\Delta \bar{V}$ - (3 double words)
 - RC - componenets of CSM position and velocity vectors, $\bar{r}_{CSM}$ and $\bar{V}_{CSM}$, at ullage begin time - (6 double words).
 - RL - components of LM position vector, $\bar{r}_{LM}$ at ullage begin time - (3 double words).
 - I - transformation indicator: I = 0 for transformation from RTCC inertial coordinates to command and display coordinates; I = 1 for transformation from command and display coordinates to RTCC inertial coordinates - (1 full word integer).
- c. Outputs and Exit — DV - (same as input array). Exit via RETURN
- d. Other Programs Used — DSQRT
- e. Data Restrictions and Limits — None
- f. Error — None
- g. Other — The input $\Delta \bar{V}$ is overwritten by the output $\Delta \bar{V}$.

Approval	Approval
<i>R. E. Cusey</i>	<i>[Signature]</i>

METHOD

The method is described as follows:

Define Command and Display Coordinate System axes, $\hat{X}_{PA}$, $\hat{Y}_{PA}$, $\hat{Z}_{PA}$, such that $\hat{Z}_{PA} = -\bar{r}_{LM}/r_{LM}$, $\hat{X}_{PA} = \hat{Z}_{PA} \times (\bar{r}_{CSM} \times \bar{V}_{CSM}) / |\hat{Z}_{PA} \times (\bar{r}_{CSM} \times \bar{V}_{CSM})|$, and $\hat{Y}_{PA} = \hat{Z}_{PA} \times \hat{X}_{PA}$. Let the input $\Delta\bar{V}$ be $\Delta\bar{V}_{in} = (V'_x, V'_y, V'_z)$ and the output $\Delta\bar{V}$ be $\Delta\bar{V}_{out} = (V'_x, V'_y, V'_z)$.

If the external $\Delta\bar{V}$ vector is to be transformed from the RTCC Inertial Coordinate System to the Command and Display Coordinate System,

$$V'_x = \Delta\bar{V}_{in} \cdot \hat{X}_{PA}$$

$$V'_y = \Delta\bar{V}_{in} \cdot \hat{Y}_{PA}$$

$$V'_z = \Delta\bar{V}_{in} \cdot \hat{Z}_{PA}$$

If the external $\Delta\bar{V}$ vector is to be transformed from the Command and Display Coordinate System to the RTCC Inertial Coordinate System,

$$\Delta\bar{V}_{out} = V'_x \hat{X}_{PA} + V'_y \hat{Y}_{PA} + V'_z \hat{Z}_{PA}.$$

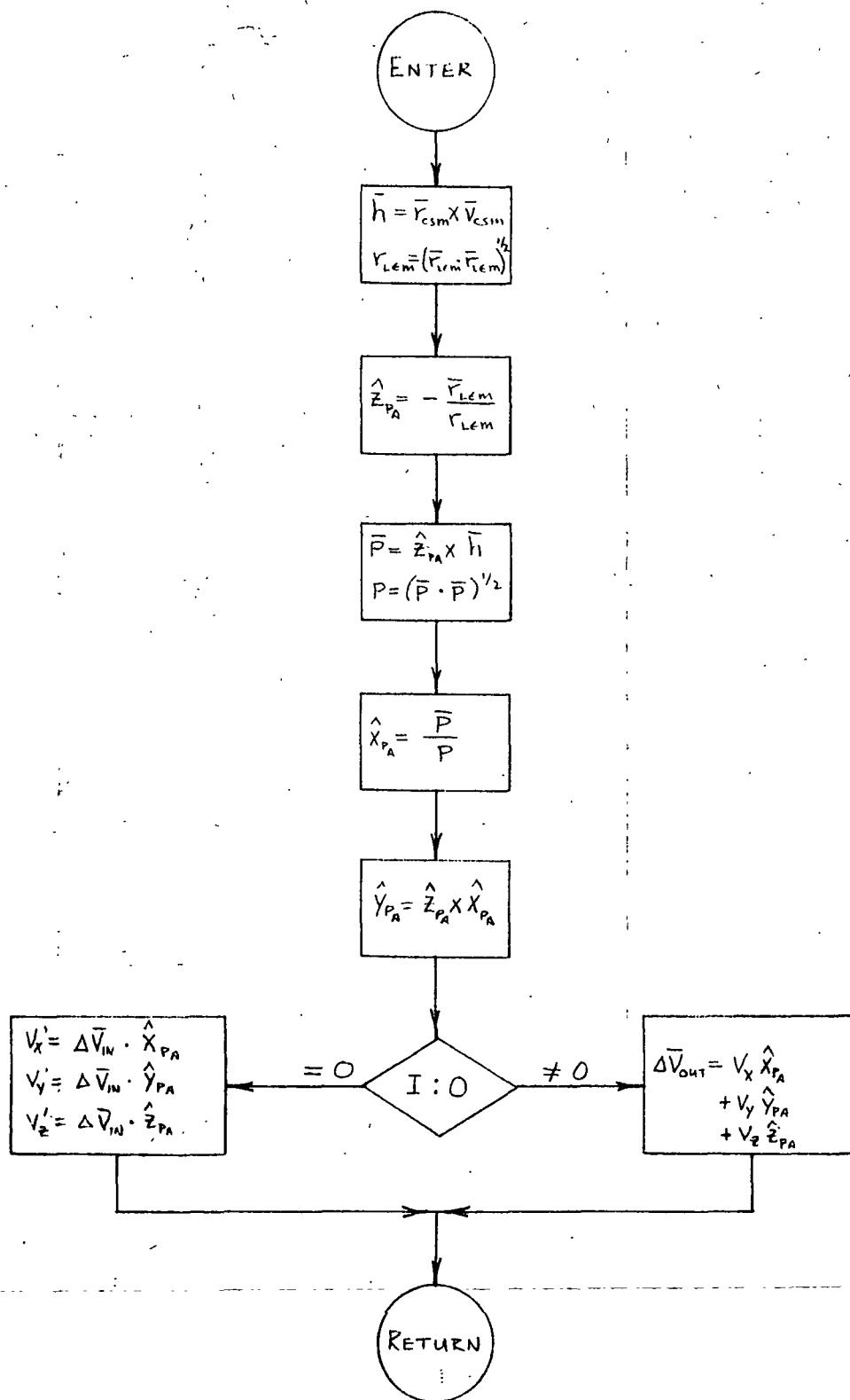


Figure PIAEDV-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PIDREF
 DATE 7/31/67 (EORP)
 REV
 PAGE 1 of 3

- a. Symbol — PIDREF
- b. Program Name — Orbit Desired Refsmat Computation Subroutine
- c. Programming Group — Mission Planning - Maneuver Control
- d. Programmer — R.E. Cusey
- e. Language — FORTRAN
- f. Data Tables Used — None
- g. MED's Used — None
- h. Writeup Completion Date — 6/6/67
- i. Flowchart — Figure PIDREF-1

FUNCTION

Given CSM or LM engine trim angles, position, velocity, attitude indicator, and unit thrust vector (all at GMTI), this subroutine computes the desired inertial measurement unit IMU coordinate axes in the RTCC Inertial Coordinate System.

USAGE

- a. Calling Sequence — CALL PIDREF (AT, R, PGYG, K, XP)
- b. Inputs —
 - AT - components of unit thrust vector, $\hat{a}_T$, at GMTI - (3 double words).
 - R - components of position and velocity vectors, $\bar{r}$ and $\bar{v}$, at GMTI - (6 double words).
 - PGYG - engine trim angles (in radians): pitch, P_G , and yaw, Y_G , for CSM or pitch, P_G , and roll, R_G , for LM - (2 double words).
 - K - attitude indicator: K = 0 for heads-down; K = 1 for heads-up (1 full word integer)
- c. Outputs and Exit — XP - components of desired IMU coordinate axes, $\hat{X}_P$, $\hat{Y}_P$, $\hat{Z}_P$ - (9 double words). Exit via RETURN.
- d. Other Programs Used — DSQRT, DSIN, DCOS
- e. Data Restrictions and Limits — None
- f. Error — None

Approval	Approval
R. E. Cusey	<i>[Signature]</i>

METHOD

The method is described as follows:

Define a right-handed set of thrust axes, $\hat{X}_T, \hat{Y}_T, \hat{Z}_T$, such that $\hat{X}_T = \hat{a}_T, \hat{Y}_T = \pm (\bar{r} \times \hat{X}_T)$, and $\hat{Z}_T = \hat{X}_T \times \hat{Y}_T$ where the positive sign is taken for $\hat{Y}_T$ if the vehicle attitude is heads-down, and the negative sign is taken if the vehicle attitude is heads-up. If $\hat{X}_T$ is parallel to $\bar{r}$ (within 0.1 degree), define $\hat{Y}_T$ as $\hat{Y}_T = (\bar{r} \times \bar{v}) / |\bar{r} \times \bar{v}|$.

Rotate $\hat{X}_T, \hat{Y}_T, \hat{Z}_T$ clockwise through angle Y_G about $\hat{Z}_T$, then clockwise through angle P_G about $\hat{Y}_T$ to obtain the desired IMU coordinate axes $\hat{X}_P, \hat{Y}_P, \hat{Z}_P$:

$$\begin{bmatrix} \hat{X}_P \\ \hat{Y}_P \\ \hat{Z}_P \end{bmatrix} = \begin{bmatrix} \cos P_G & 0 & \sin P_G \\ 0 & 1 & 0 \\ -\sin P_G & 0 & \cos P_G \end{bmatrix} \begin{bmatrix} \cos Y_G & -\sin Y_G & 0 \\ \sin Y_G & \cos Y_G & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{X}_T \\ \hat{Y}_T \\ \hat{Z}_T \end{bmatrix}$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PIDREF

DATE 7/31/67 (EORP)

REV

PAGE 3 of 3

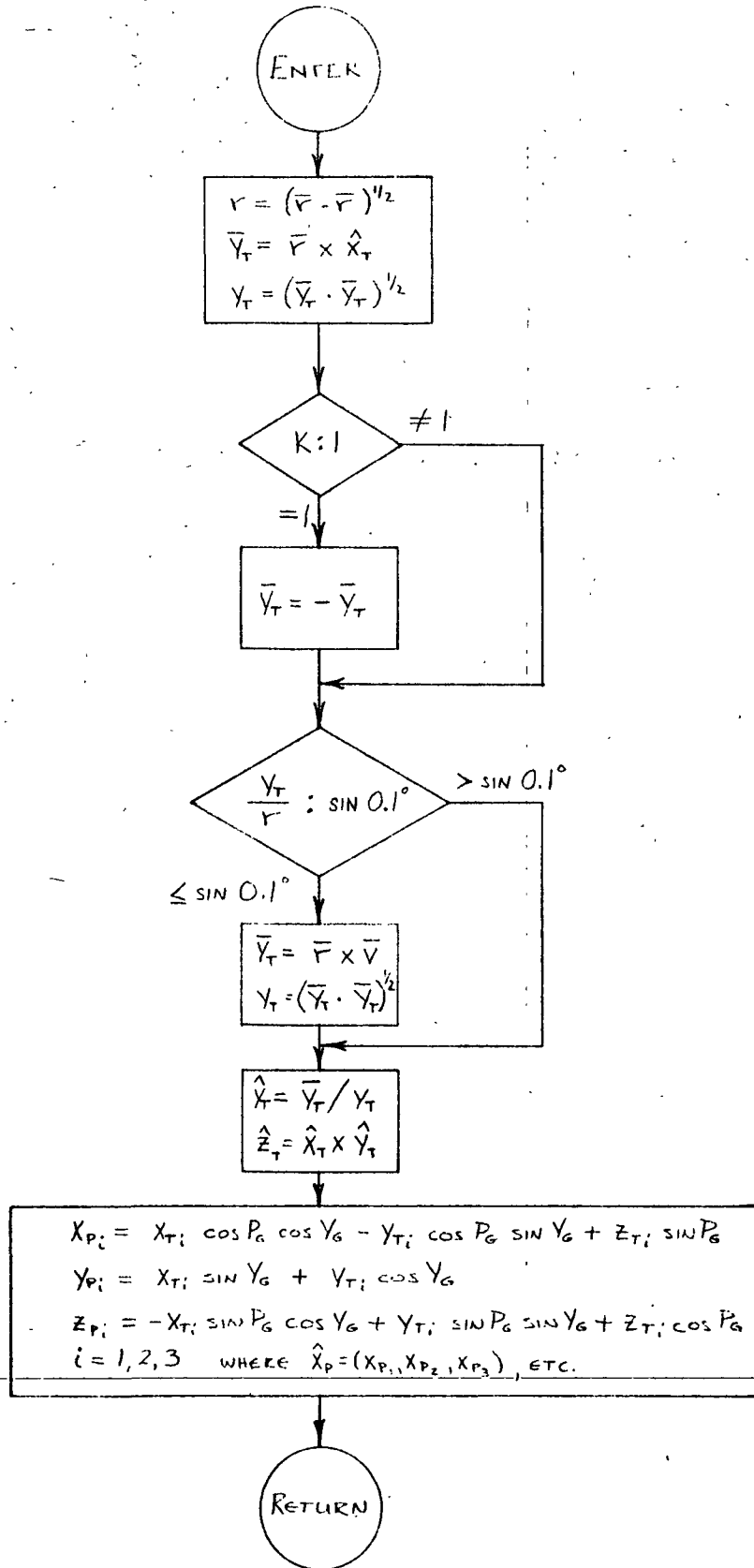


Figure PIDREF-1. Flowchart

- a. Symbol — PIDVRS
- b. Program Name — Maneuver Confirmation $\overline{\Delta V}$ Residuals Subroutine
- c. Programming Group — Maneuver Control
- d. Programmer (s) — J. Myers
- e. Language — Assembler
- f. Data Tables Used — None
- g. MED's Used — None
- h. Writeup Completion Date — 8/10/67
- i. System Parameters — Table PIDVRS-1
- j. Flowchart — Figure PIDVRS-1

FUNCTION

The purpose of this subroutine is to compute the actual maneuver velocity increment for a CSM or LM executed maneuver from the original ΔV vector and the maneuver $\overline{\Delta V}$ residuals (in the vehicle RCS control axes system). The actual maneuver velocity increment can be used to confirm the executed maneuver.

USAGE

- a. Calling Sequence — CALL PIDVRS, (DVR, VEH, PYR, DVO, RM)
- b. Inputs — The inputs are the following:
 - (1) DVR is the address of three doublewords which are the maneuver $\overline{\Delta V}$ residuals in the vehicle RCS control axes system ($\Delta V_X''$, $\Delta V_Y''$, $\Delta V_Z''$).
 - (2) VEH is the address of a halfword which is the vehicle indicator (1 for CSM and 3 for LM).
 - (3) PYR is the address of three doublewords which are the Pitch, Yaw, and Roll in the Inertial Measuring Unit (IMU) system at the time the residuals were read.
 - (4) DVO is the address of three doublewords which are the components of the original planned ΔV vector in the RTCC inertial system ($\overline{\Delta V}_O$).
 - (5) RM is the address of nine doublewords which are the inertial coordinates of the unit vectors in the direction of the platform X, Y, and Z axes ($\hat{X}_p$, $\hat{Y}_p$, $\hat{Z}_p$).

Approval	Approval
<i>J. A. Myers</i>	<i>A. E. James</i>

- c. Outputs and Exit — The outputs are the components of the actual maneuver velocity vector ($\overline{\Delta V}_A$), which are stored in DVO. PIDVRS exists by returning to the caller.
- d. Other Programs Used — DSIN, DCOS
- e. Data Restrictions and Limits — None
- f. Error — None

METHOD

For the LM the RCS control axes are parallel to the vehicle body axes. For the CSM the vehicle body axes are obtained by rotating the RCS control axes counter-clockwise through the angle θ (presently 7.25 degrees) about the X-RCS control axis.

$$\begin{bmatrix} \hat{X}_B \\ \hat{Y}_B \\ \hat{Z}_B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \hat{X}_{RCS} \\ \hat{Y}_{RCS} \\ \hat{Z}_{RCS} \end{bmatrix}$$

The maneuver $\overline{\Delta V}$ residuals in the RCS control axes system ($\Delta V_X''$, $\Delta V_Y''$, $\Delta V_Z''$) are transformed into the platform coordinate system by the following:

$$\begin{bmatrix} \Delta V_x' \\ \Delta V_y' \\ \Delta V_z' \end{bmatrix} = \begin{bmatrix} \cos P & 0 & \sin P \\ 0 & 1 & 0 \\ -\sin P & 0 & \cos P \end{bmatrix} \begin{bmatrix} \cos M & -\sin M & 0 \\ \sin M & \cos M & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \Delta V_x'' \\ \Delta V_y'' \\ \Delta V_z'' \end{bmatrix}$$

For the LM, P is the Pitch, M is the Roll, and ϕ is the Yaw in the Inertial Measuring Unit (IMU) System at the time the residuals were read; for the CSM, P is the Pitch, M is the Yaw, and ϕ is the Roll minus θ in the IMU System at the time the residuals were read.

The maneuver $\overline{\Delta V}$ residuals in the platform coordinate system $(\Delta V_x', \Delta V_y', \Delta V_z')$ are transformed into the RTCC inertial coordinate system by the following :

$$\begin{bmatrix} \Delta V_{RES_x} \\ \Delta V_{RES_y} \\ \Delta V_{RES_z} \end{bmatrix} = \begin{bmatrix} Xp_x & Yp_x & Zp_x \\ Xp_y & Yp_y & Zp_y \\ Xp_z & Yp_z & Zp_z \end{bmatrix} \begin{bmatrix} \Delta V_x' \\ \Delta V_y' \\ \Delta V_z' \end{bmatrix}$$

where (Xp_x, Xp_y, Xp_z) , (Yp_x, Yp_y, Yp_z) , (Zp_x, Zp_y, Zp_z) are the inertial coordinates of the unit vectors in the direction of the platform X, Y, and Z axes.

The actual maneuver velocity vector is then computed as follows:

$$\overline{\Delta V}_A = \overline{\Delta V}_O - \overline{\Delta V}_{RES}.$$

Table PIDVRS-1. System Parameters

[illegible]

*Type
H — Fixed-point half word
F — Fixed-point full word
E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

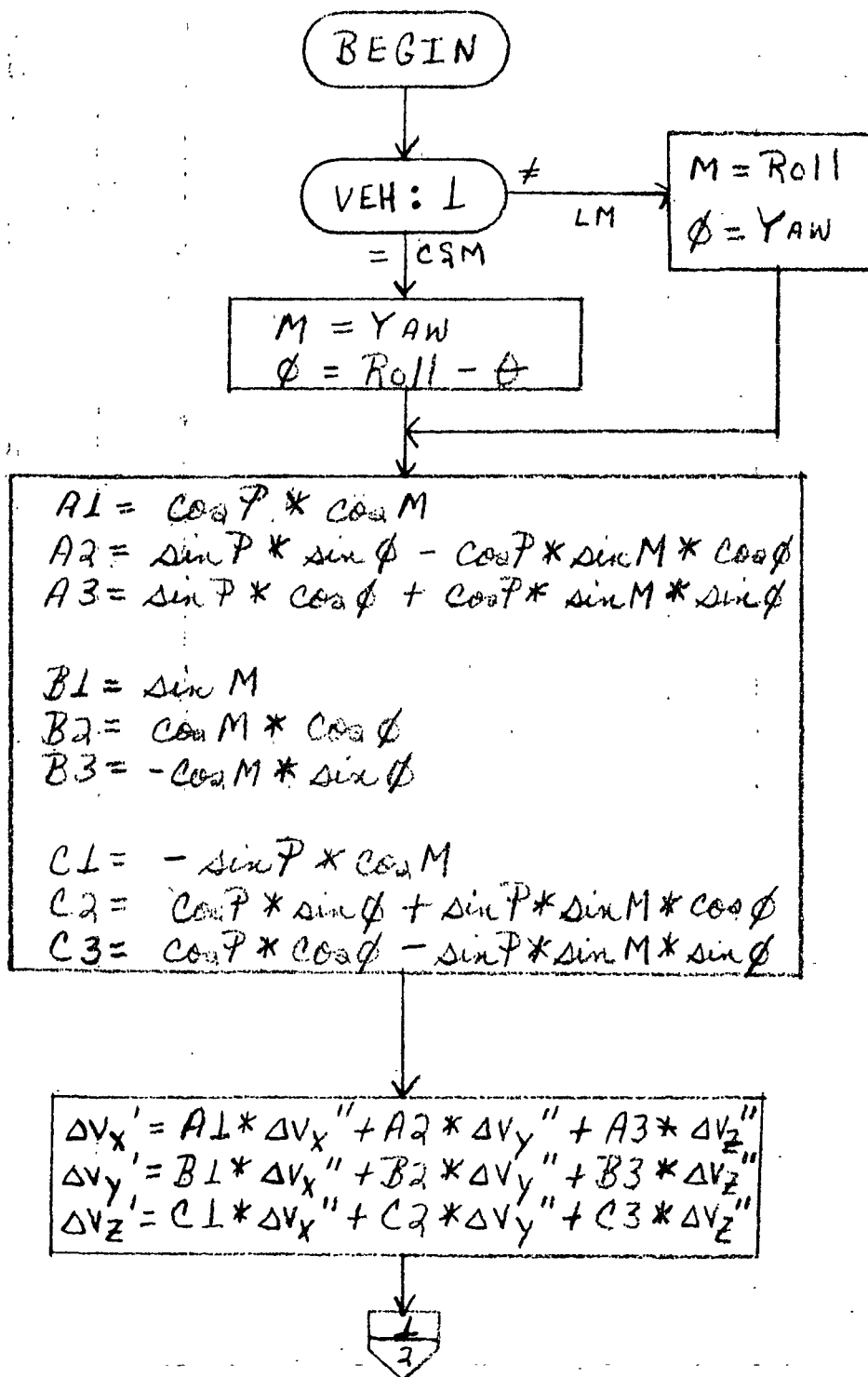


Figure PIDVRS-1. Flowchart

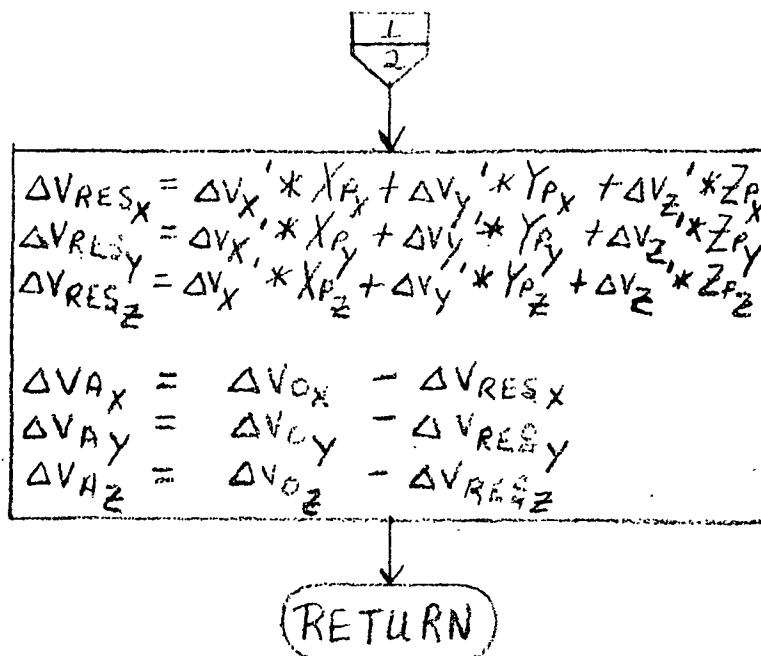


Figure PIDVRS-1. Flowchart

(Sheet 2 of 2)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PIEXDV
DATE 8/11/67 (EORP)
REV
PAGE 1 of 4

- a. Symbol — PIEXDV
- b. Program Name — External $\overline{\Delta V}$ Coordinate Transformation Subroutine
- c. Programming Group — Mission Planning
- d. Programmer(s) — E. Kramer
- e. Language — Assembler
- f. Data Tables Used — None
- g. MED's Used — None
- h. Writeup Completion Date — 5/9/67
- i. System Parameters — Table PIEXDV-1.
- j. Flowchart — Figure PIEXDV-1.

FUNCTION

This subroutine basically performs a special coordinate transformation on an input velocity increment vector $\overline{\Delta V}$. If the input $\overline{\Delta V}$ is in the external $\overline{\Delta V}$ uplink coordinate system the transformed $\overline{\Delta V}$ is in the RTCC inertial coordinate system. If the input $\overline{\Delta V}$ is in the RTCC inertial coordinate system, the transformed $\overline{\Delta V}$ is in the external $\overline{\Delta V}$ uplink coordinate system. The program is consistent with the on-board capabilities for the first Earth-orbit rendezvous mission as presented in the MIT CSM and LM R547 documents dated October, 1966. The subroutine is designed for the CSM GNCS and LM PGNCs external $\overline{\Delta V}$ guidance and cannot be used for LM AGS external $\overline{\Delta V}$ computations.

USAGE

- a. Calling Sequence — CALL PIEXDV, ($\overline{\Delta V}$, $\overline{R}_{ig}$, T, W_{ig}) or for FORTRAN - PIEXDV (ΔV , R_{ig} , T, W_{ig})
- b. Inputs — See "Method"
- c. Outputs and Exit — See "Method"
- d. Other Programs Used — DSQRT, DSIN, DCOS
- e. Data Restrictions and Limits — None
- f. Error — None

Approval	Approval
<i>[Signature]</i>	<i>[Signature]</i>

METHOD

The calling program passes 4 pointers: (1) $\overline{\Delta V}$ - The input external ΔV vector in double precision; (2) $\overline{R}_{ig}$ - The position and velocity vectors at GMTI in double precision; (3) T - The thrust level of the designated thruster in double precision; and (4) W_{ig} - The weight of the vehicle at GMTI in double precision. If this weight is positive, the input $\overline{\Delta V}$ is transformed from the linkup coordinate system to the RTCC inertial coordinate system. If the weight is negative, the inverse transformation is done. the three component $\overline{\Delta V}$ output vector is stored in the same storage as the input $\overline{\Delta V}$; therefore, this area must be under the users protect key. The other inputs are not disturbed.

*Type
H — Fixed-point half word
F — Fixed-point full word
E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

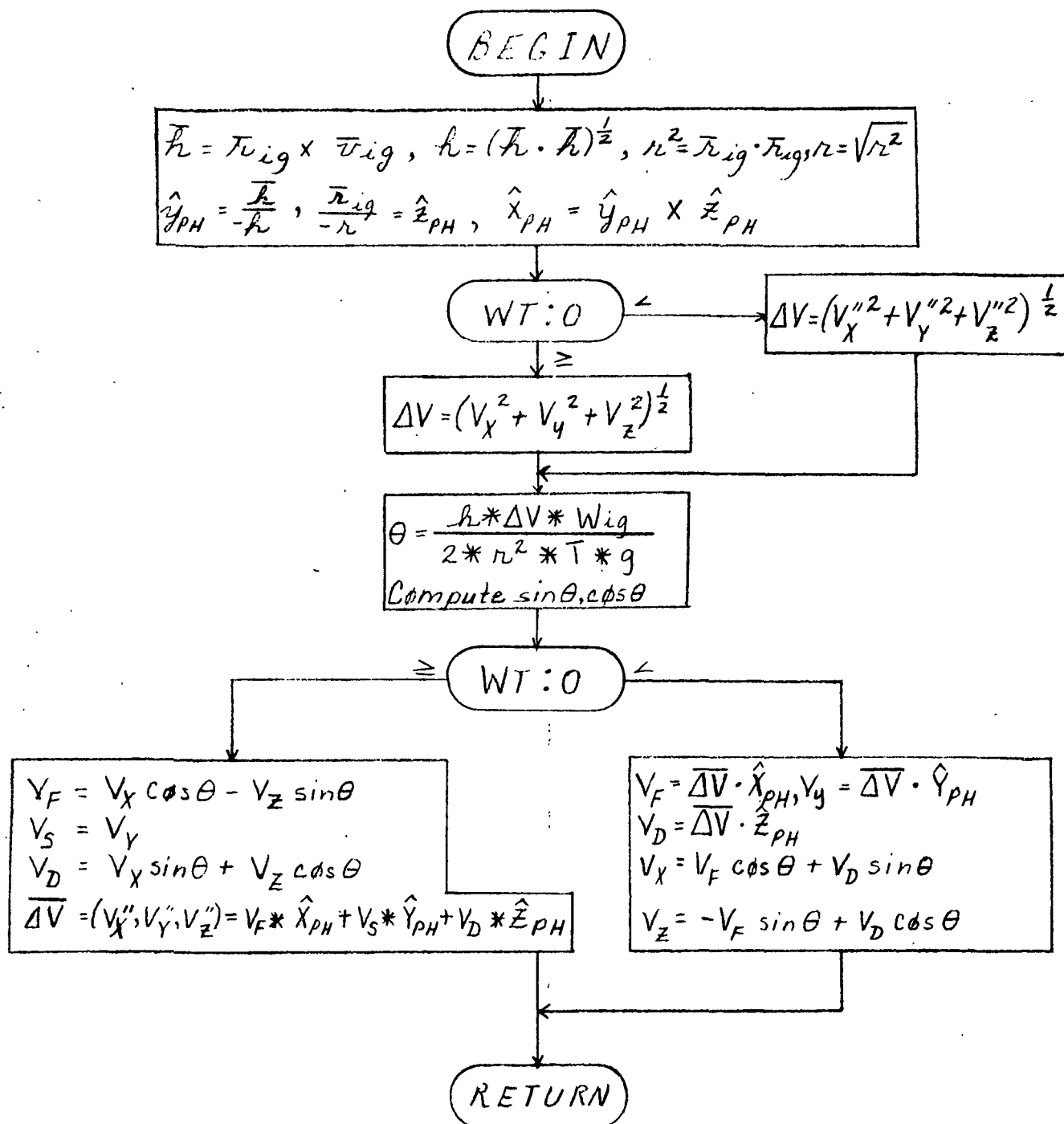


Figure PIEXDV-1. Flowchart

(Sheet 1 of 1)

PIFAAP
APOGEE/PERIGEE MAGNITUDE DETERMINATIONREFERENCES

1. Programming Group — Rendezvous Planning
2. Programmer — C. Peabody
3. Language — Assembler
4. Writeup Completion Date — 3/18/68
5. Data Tables Used — None
6. MED's Used — None
7. Flowchart — Figure 1

FUNCTION

The function is to compute apogee and perigee magnitudes. A first zonal harmonic (J_2) and a low eccentricity approximation are used which results in a small fast routine.

USAGE

1. Calling Sequence — CALL PIFAAP (AREA)
2. Inputs — a (semimajor axis), e (eccentricity), i (angle of inclination), f (true anomaly), u (argument of latitude), r (radius magnitude) in AREA(1) through AREA(6). Units are E_r and radians.
3. Outputs and Exit — AREA(7) = radius magnitude at apogee (E_r). AREA(8) = radius magnitude at perigee (E_r).
4. Other Programs Used — DSIN, DCOS, DATAN2, DSQRT
5. Data Restrictions and Limits — The program is for elliptical orbits only and works best for low eccentricities.

METHOD

A reference orbit is selected given the osculating elements at an epoch so that the variation of the osculating radius within one orbit is minimum. The reference radius at epoch is selected as the actual radius at that time. The remaining elements, a and e, are derived. The osculating elements of the actual and the reference orbits are propagated from the epoch time by Keplerian motion. The two points in the Keplerian orbits are determined where the difference between the radii are maximum and minimum. These two differences are added to the reference orbit radius at the epoch time to obtain the actual apogee and perigee radii.

Approval	Approval
<i>C. Peabody</i>	<i>A. Amietti</i>

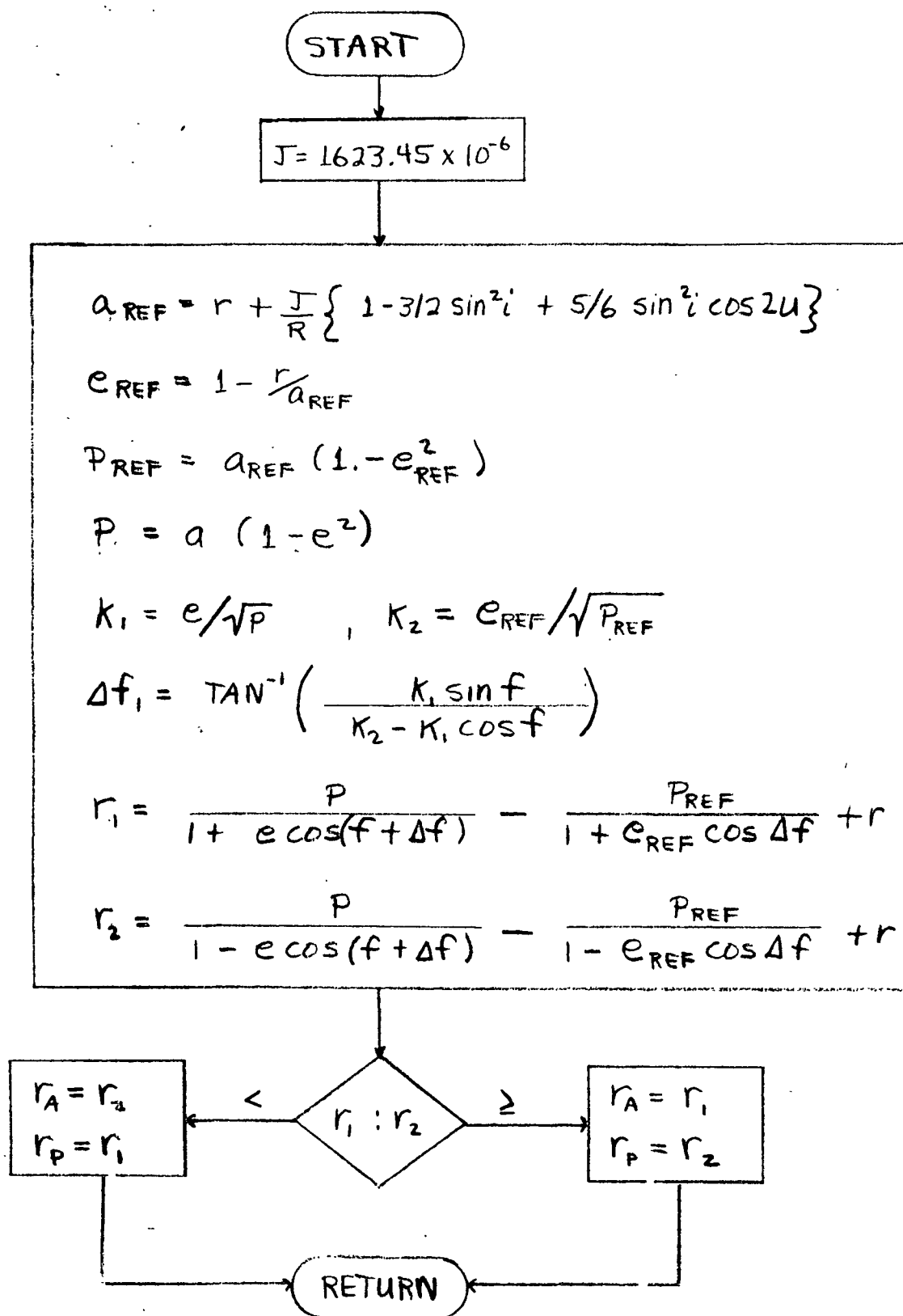


Figure 1

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMDDMT
DATE 6/16/67 (206)
REV
PAGE 1 of 4

- a. Symbol — PMDDMT
- b. Program Name — Detailed Maneuver Table Display
- c. Programming Group — Maneuver Control
- d. Programmer — R. Taylor
- e. Language — Assembler
- f. Data Tables Used — PZMPTLEM, EZETVMED, PZDMTLEM
- g. MED's Used — U20
- h. Writeup Completion Date — 3/21/67
- i. System Parameters — Table PMDDMT-1
- j. Flowchart — Figure PMDDMT-1

FUNCTION

This load module formats the Detailed Maneuver Table Display.

USAGE

- a. Calling Sequence — ATTACH
- b. Inputs — Internal format number, entry queue, vehicle code, maneuver number
- c. Outputs and Exit — PZDMTLEM. Exit is via a return.
- d. Other Programs Used — RHFDFCNV
- e. Data Restrictions and Limits — None
- f. Error — If the requested maneuver is not in the Mission Plan Table or if there are no maneuvers, "No Maneuver in Table" is displayed. An "*" is displayed for an update in process.

METHOD

This load module takes the MED data in EZETVMED and formats that maneuver according to the NASA format. The maneuver data is written in PZDMTLEM for use in command loads.

Approval	Approval
<i>R. Taylor</i>	<i>S. E. James</i>

Table PMDDMT-1. System Parameters

[illegible]

*Type
H — Fixed-point half word
F — Fixed-point full word
E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

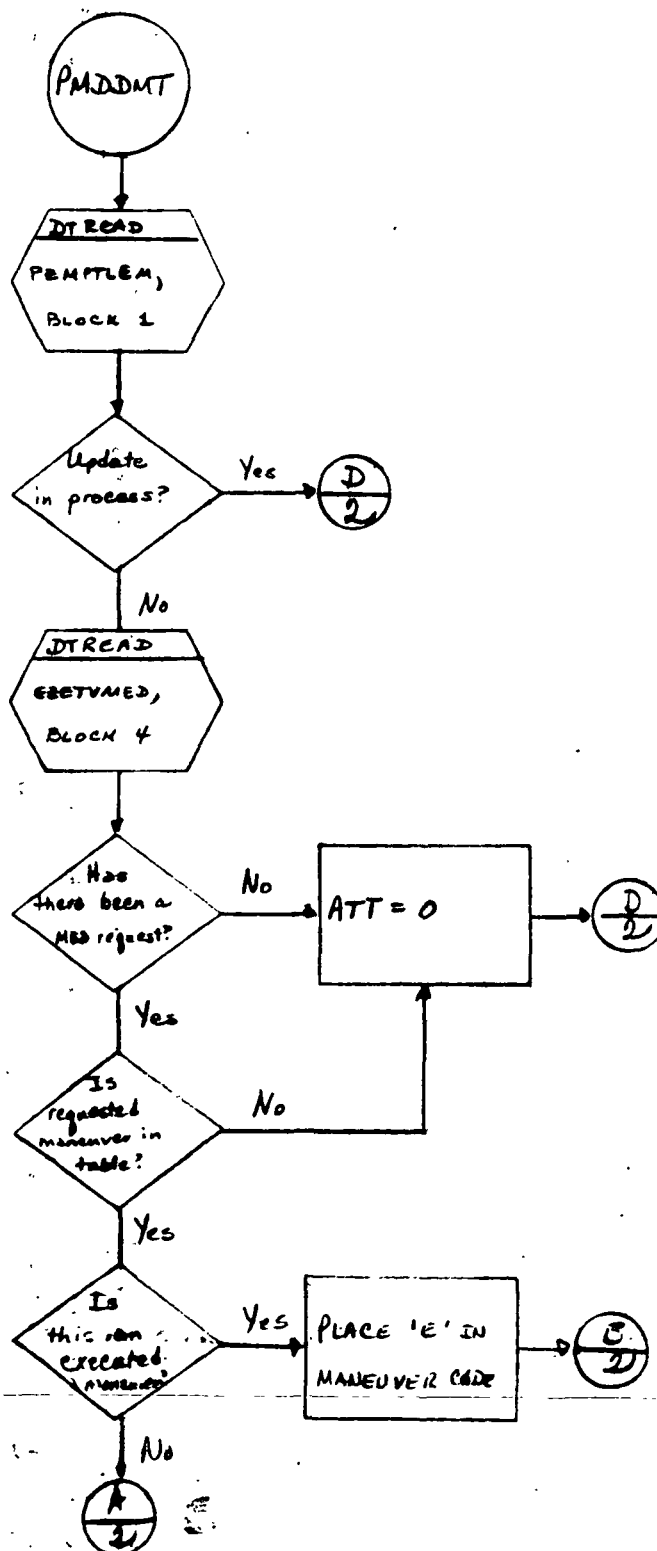


Figure PMDDMT-1. Flowchart

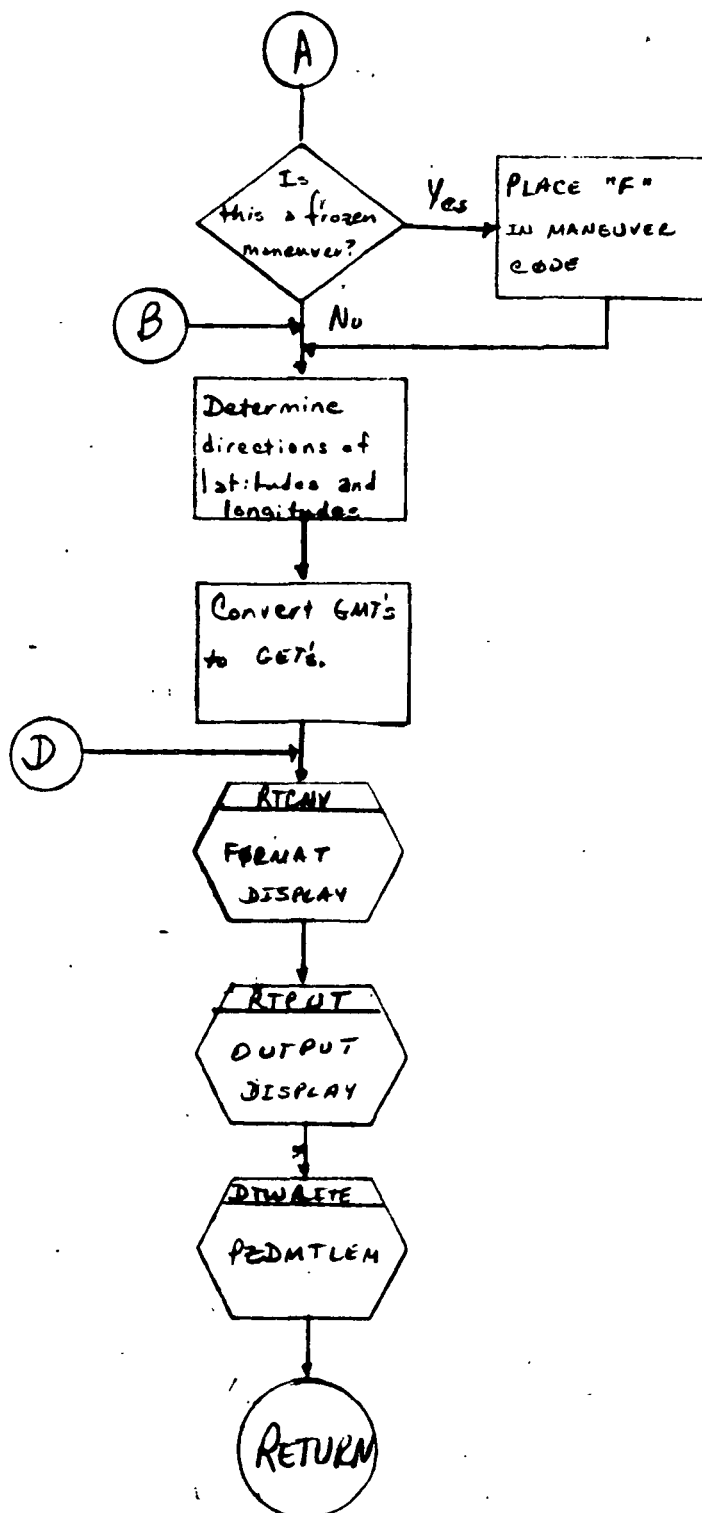


Figure PMDDMT-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMDDMT

DATE 8/22/67 (EORP)

REV 10/10/67

PAGE 1 of 16

- a. Symbol — PMDDMT
- b. Program Name — Detailed Maneuver Table Display
- c. Programming Group — Mission Planning
- d. Programmer(s) — R. Taylor
- e. Language — Assembler
- f. Data Tables Used — PZMPTCSM, PZMPTLEM, TZGDNSAV, EZETVMED, GZLTRA
- g. MED's Used — U20
- h. Writeup Completion Date — 7/28/67
- i. System Parameters — Table PMDDMT-1
- j. Flowchart — Figure PMDDMT-1

FUNCTION

This load module formats the Detailed Maneuver Table, which includes providing IMU attitudes and velocity components based on a specified reference matrix, two-vehicle quantities (θ , $\dot{\theta}$, δ , Δh), and the AGS external ΔV vector in the command and display system.

USAGE

- a. Calling Sequence — ATTACH
- b. Inputs — Internal format number, entry queue, vehicle code, maneuver number, reference matrix, heads up/down indicator.
- c. Outputs and Exit — Detailed Maneuver Table display; Exit via return.
- d. Other Programs Used — DARSIN, RHFD FCNV, DSQRT, ELGETE, GLQATN, GLMXPDP, GLMDPD, PMMAEG, EIGMTX, PIAEDV, PIDREF, GIMIKC, GIMKIC
- e. Data Restrictions and Limits — None
- f. Error — Conditions and the associated message are (1) Maneuver requested not in Mission Plan Table — "MANEUVER NOT IN TABLE"; (2) Trajectory update in process — "!", "UPDATE IN PROCESS"; (3) Reference matrix not available — "MATRIX NOT AVAILABLE, AFFECTS P, Y, R, VGX, VGY, VGZ"; (4) Vector from other ephemeris not fetched — "VECTOR NOT AVAILABLE, AFFECTS θ , $\dot{\theta}$, δ , Δh "; (5) Error return from PMMAEG — "AEG ERROR, AFFECTS θ , $\dot{\theta}$, δ , Δh "; (6) Vector for AGS computations not fetched — "VECTOR NOT AVAILABLE, AFFECTS V_X , V_Y , V_Z "; (7) PMMAEG error return — "AEG ERROR, AFFECTS V_X , V_Y , V_Z "; (8) Incomplete or invalid maneuver — "MANEUVER QUANTITIES NOT CURRENT"; (9) Burnout vector replaced by anchor vector — "ANCHOR VECTOR WITHIN MANEUVER".

Approval	Approval
<i>R. Taylor</i>	<i>SE James</i>

METHOD

This load module uses information from the U20 MED to format the Detailed Maneuver Table display (MSK 54).

Rendezvous - two vehicle quantities are computed by getting a vector from the ephemeris for the other vehicle. Then

$\bar{r}_m$ Position vector for maneuvering vehicle

$\bar{V}_m$ Velocity vector for maneuvering vehicle.

$\bar{r}_\alpha$ Position vector for other vehicle.

$\bar{V}_\alpha$ Velocity vector for other vehicle

$$\bar{h}_m = \bar{r}_m \times \bar{V}_m$$

$$\bar{h}_\alpha = \bar{r}_\alpha \times \bar{V}_\alpha$$

$$\bar{N} = \bar{h}_\alpha \times \bar{h}_m$$

$$h_m = |\bar{h}_m|$$

$$h_\alpha = |\bar{h}_\alpha|$$

$$V_\alpha^2 = \bar{V}_\alpha \cdot \bar{V}_\alpha$$

$$V_m^2 = \bar{V}_m \cdot \bar{V}_m$$

$$a_\alpha = \frac{r_\alpha}{2 - \frac{r_\alpha V_\alpha^2}{\mu}}$$

$$a_m = \frac{r_m}{2 - \frac{r_m V_m^2}{\mu}}$$

$$\dot{\theta} = 2\pi \left[\sqrt{\frac{a_{\alpha}^3}{a_m^3}} - 1 \right]$$

$$\delta = \sin^{-1} \left[\frac{|\bar{N}|}{h_m h_{\alpha}} \right]$$

θ and Δh are computed using PMMAEG, update option 6. The phase angle θ is computed at time match (measured from other vehicle to maneuvering vehicle) and Δh at position match.

The reference matrix requested on the MED is fetched using the subroutine EIGMTX except for a desired reference matrix request. For a desired request subroutine PIDREF calculates the matrix using the heads up/heads down attitude*. The IMU attitudes are calculated as follows:

$$\left. \begin{matrix} \hat{X}_p \\ \hat{Y}_p \\ \hat{Z}_p \end{matrix} \right\} \begin{matrix} \text{Reference matrix returned by} \\ \text{EIGMTX or PIDREF; or, for S-IVB blow-} \\ \text{down vent, fetched from GZLTRA} \end{matrix}$$

* For an S-IVB maneuver, the S-IVB reference matrix from launch is used with CSM logic.

$$\left. \begin{matrix} \hat{X}_B \\ \hat{Y}_B \\ \hat{Z}_B \end{matrix} \right\} \text{ vehicle body axis from Mission Plan Table}$$

$$A = \sin^{-1} (\hat{X}_B \cdot \hat{Y}_P)$$

$$B = \tan^{-1} \left[\frac{-\hat{Z}_B \cdot \hat{Y}_P}{\hat{Y}_B \cdot \hat{Y}_P} \right]$$

$$P = \tan^{-1} \left[\frac{-\hat{X}_B \cdot \hat{Z}_P}{\hat{X}_B \cdot \hat{X}_P} \right]$$

For the CSM, yaw = A, roll = B, and pitch = P. LM attitudes are roll = A, yaw = B, and pitch = P. S-IVB attitudes are computed as for the CSM. If A is equal to $\pi/2$ or $3\pi/2$, B and P are computed differently.

$$\begin{matrix} B = 0.0 \\ P = \text{TAN}^{-1} \left[\frac{\hat{X}_P \cdot \hat{Z}_B}{\hat{Z}_P \cdot \hat{Z}_B} \right] \end{matrix}$$

The IMU delta velocity vector is based on a ΔV vector stored in the Mission Plan Table.

$$V_{gx} = \overline{\Delta V} \cdot \hat{X}_P$$

$$V_{gy} = \overline{\Delta V} \cdot \hat{Y}_P$$

$$V_{gz} = \overline{\Delta V} \cdot \hat{Z}_P$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMDDMT
DATE 8/22/67 (EORP)
REV 10/10/67
PAGE 5 of 16

For a LM AGS attitude maneuver, the external ΔV vector is computed by the subroutine PIAEDV using vectors for both vehicles at the time to begin the maneuver.

The initial vehicle weights are determined by using the configuration code at ignition and the associated total configuration weight.

The information is then formatted by RHFDFCNV and transmitted to the D/TV.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMDDMT

DATE 8/22/67 (EORP)

REV 10/10/67

PAGE 6 of 16

Table PMDDMT-1. System Parameters

Symbol in Program	Symbol In Master Tape	Type *	Units	Comments
--	MCGMTL	D	hrs.	Lift-off time of first vehicle
TWOPI	MCMTPI	D	rad.	2π
SQRMU	MCERMU	D	$(E_r^3/\text{hr.}^2)^{1/2}$	Square root of μ
PI	MCMPPIR	D	rad.	π
NM	MCCNMC	D	nm/ E_r	Conversion of Earth radii to nautical miles
DG	MCCDPR	D	deg./rad.	Conversion of radians to degrees
SC	MCCVCU	D	ft/sec E_r/hr	Conversion of $E_r/\text{hr.}$ to ft./sec.
HAFPI	MCMHPI	D	rad:	$\pi/2$
MU	MCEMUU	D	$E_r^3/\text{hr.}^2$	μ

*Type

H - Fixed-point half word

D - Double-precision (floating-point)

F - Fixed-point full word

E - Single-precision (floating-point)

D - Double-precision (floating-point)

X - Hexadecimal

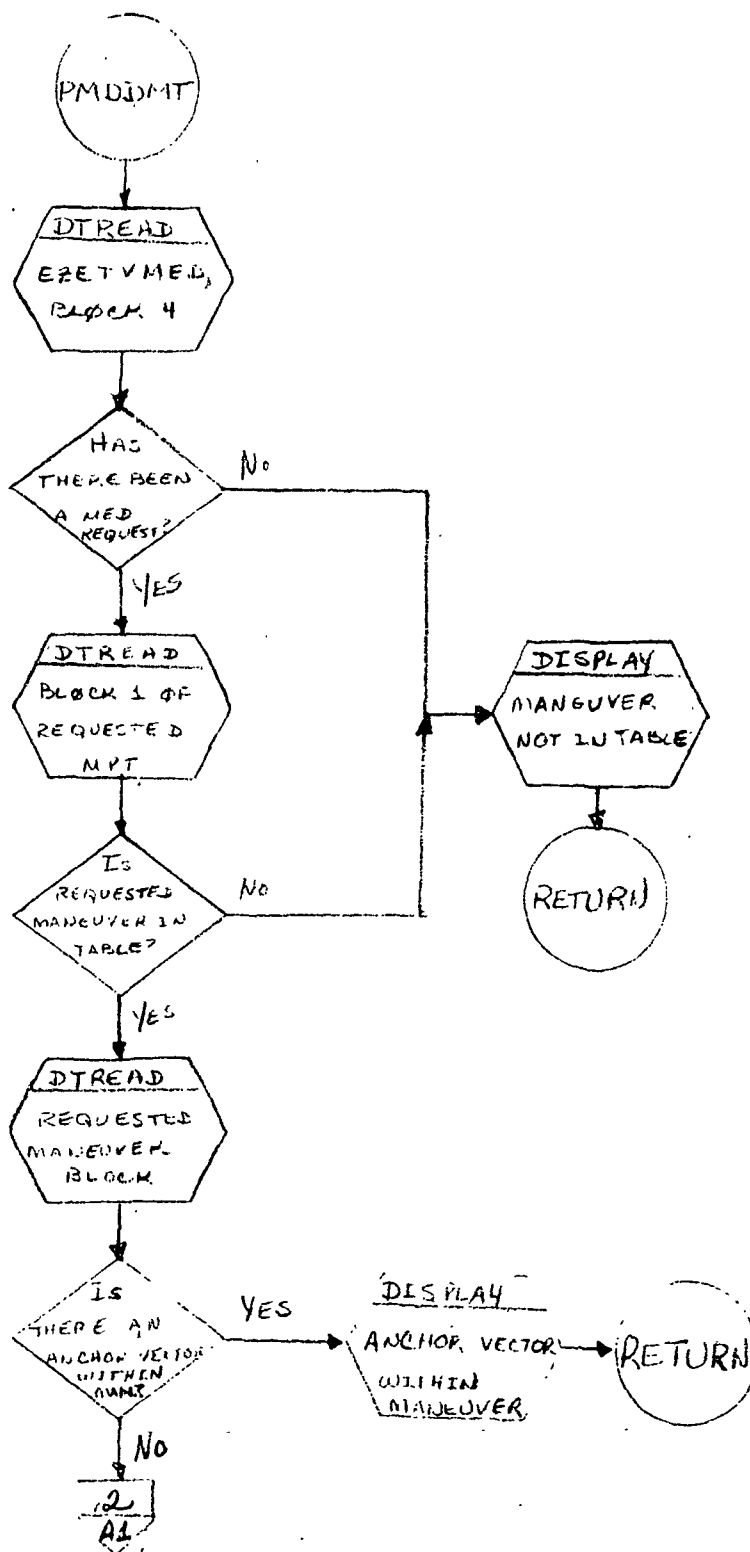


Figure PMDDMT-1. Flowchart

(Sheet 1 of 10)

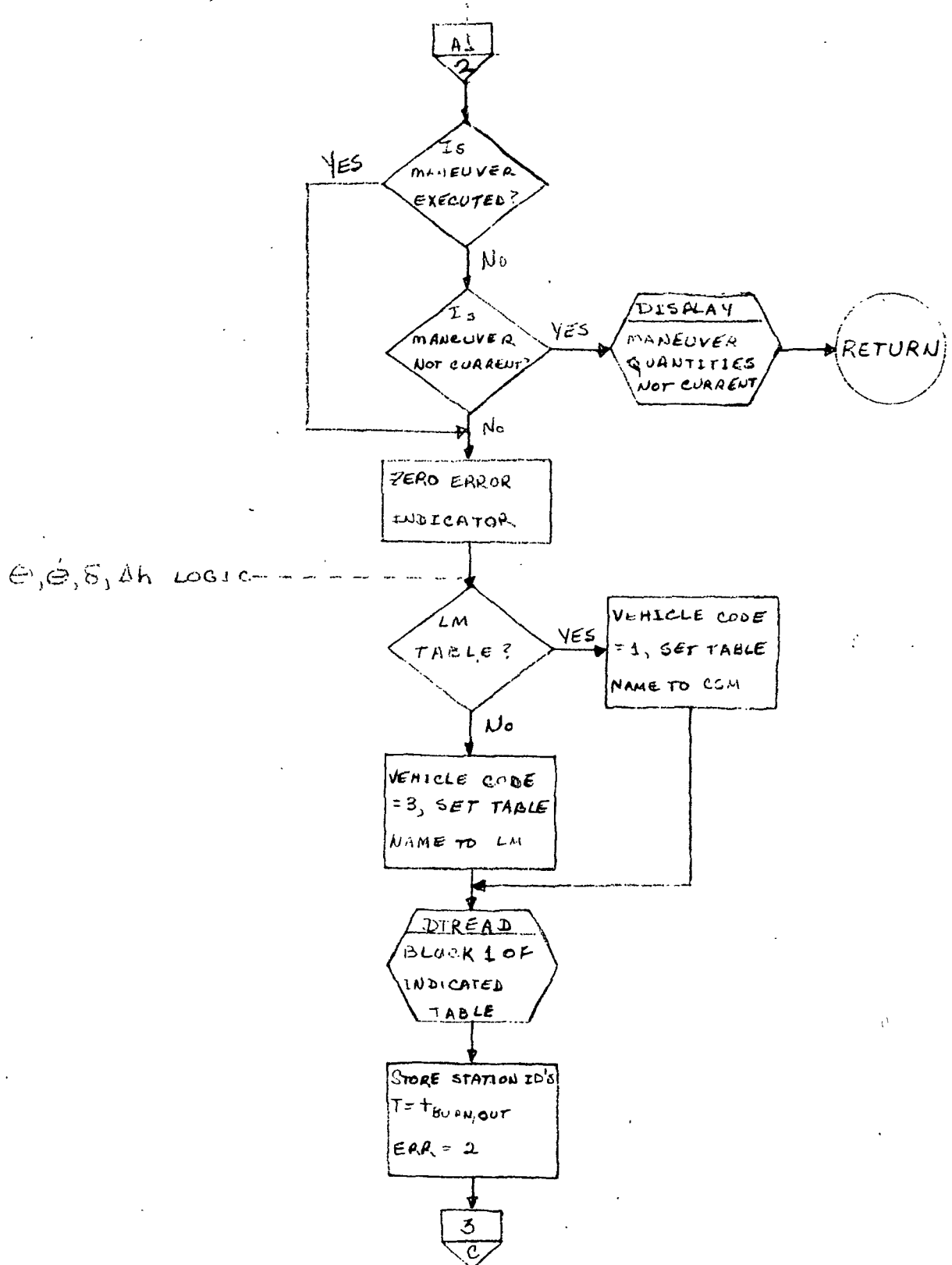


Figure PMDDMT-1. Flowchart

(Sheet 2 of 10)

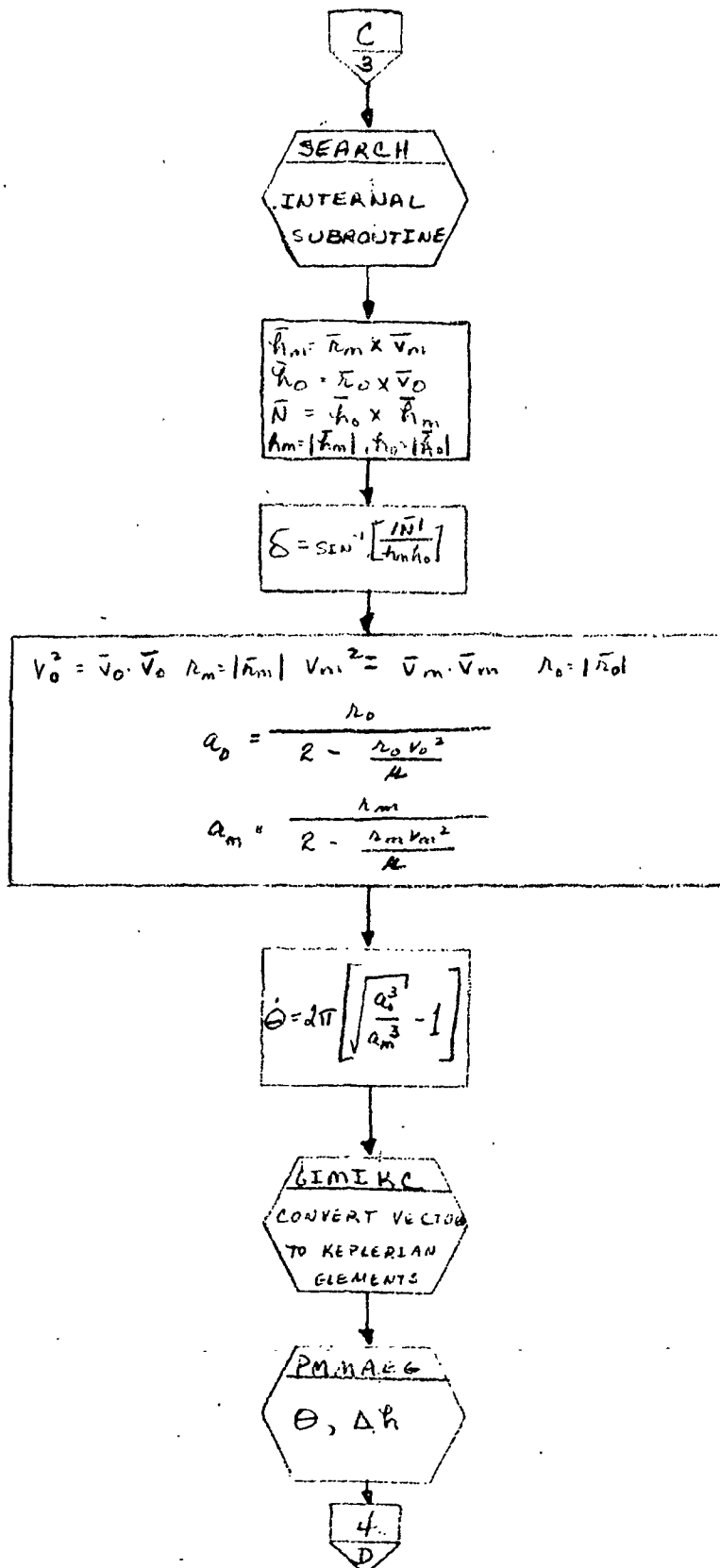


Figure PMDDMT-1. Flowchart

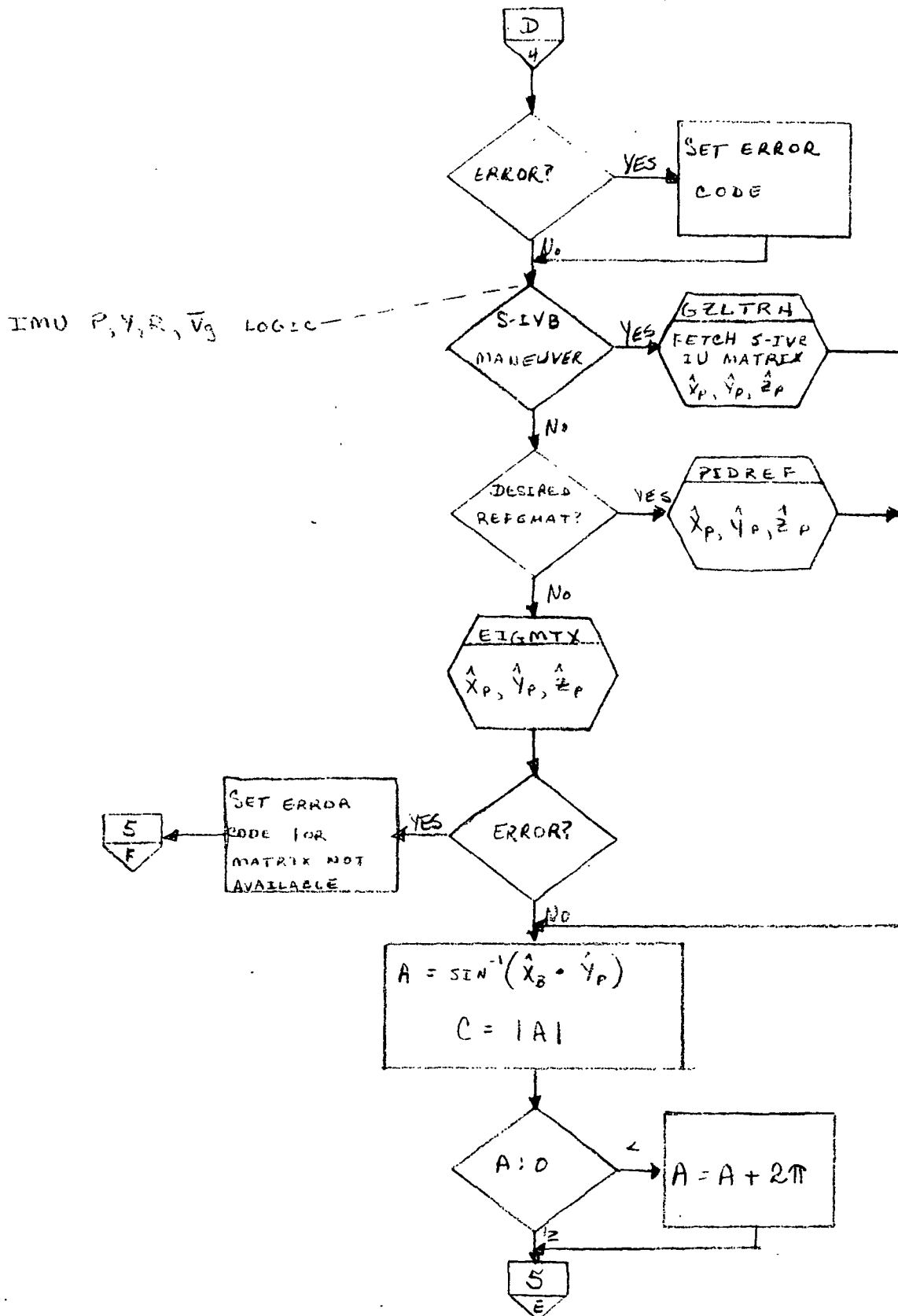


Figure PMDDMT-1. Flowchart

(Sheet 4 of 10)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMDDMT

DATE 8/22/67 (EORP)

REV 10/10/67

PAGE 11 of 16

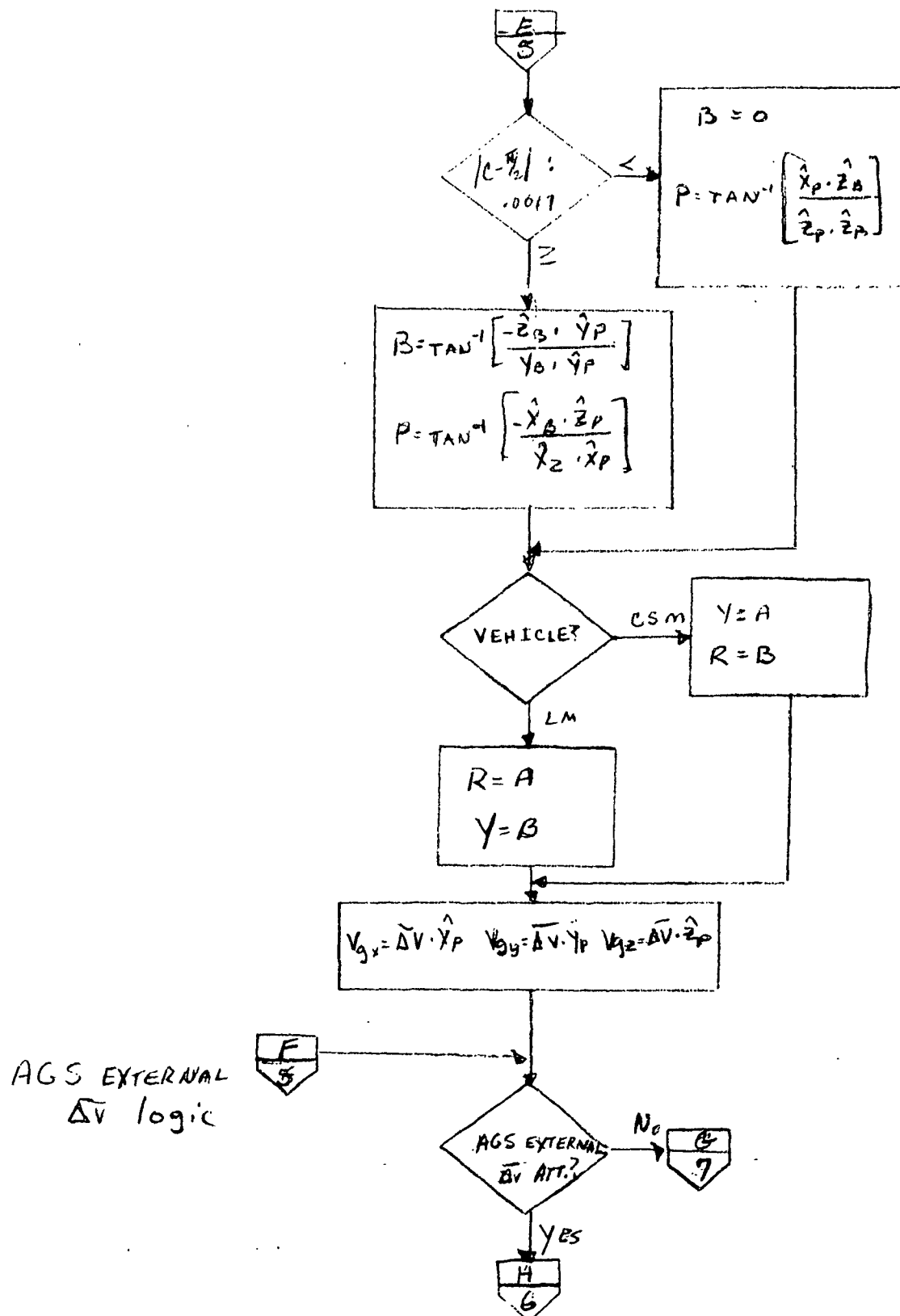


Figure PMDDMT-1. Flowchart

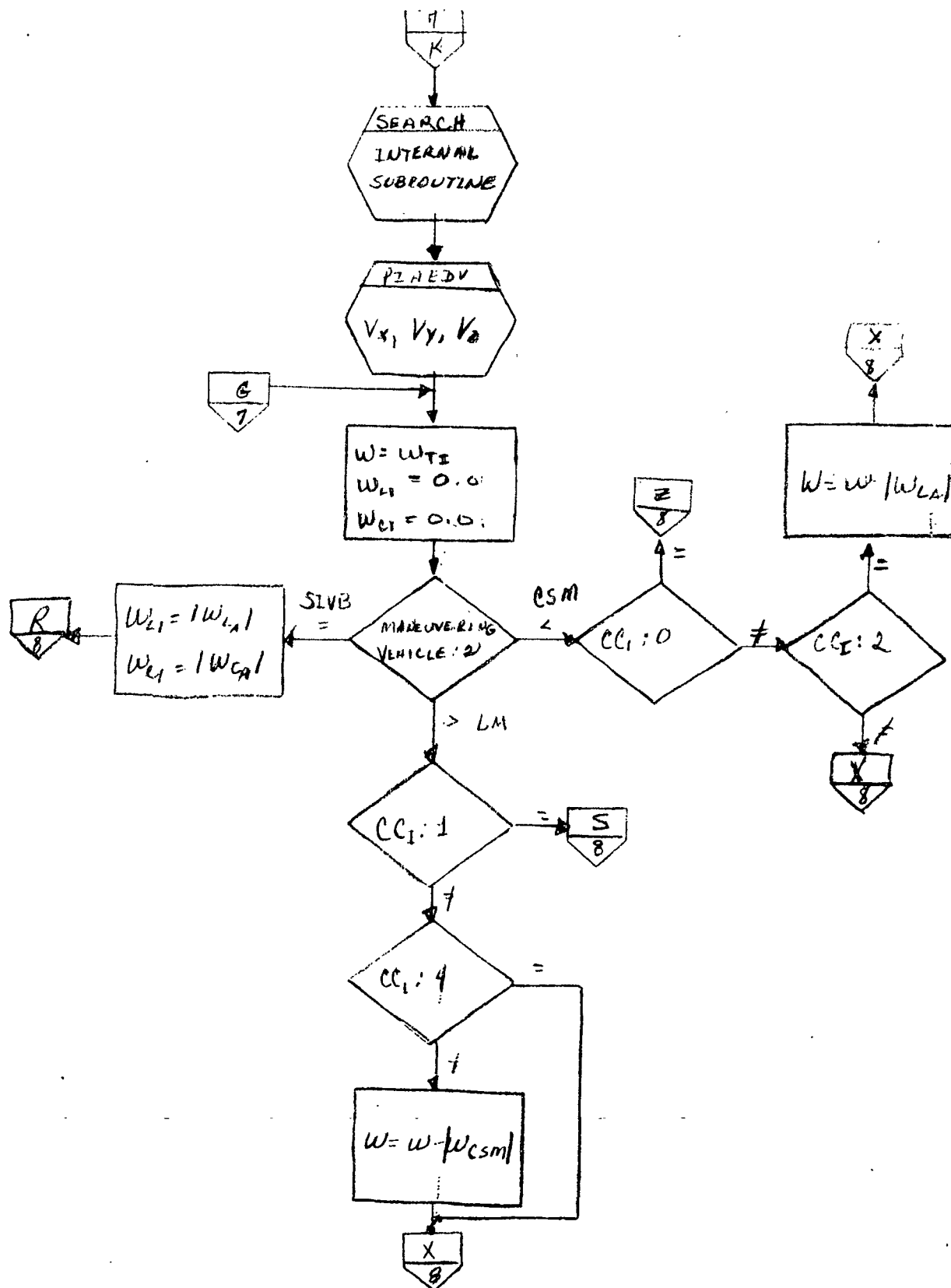


Figure PMDDMT-1. Flowchart

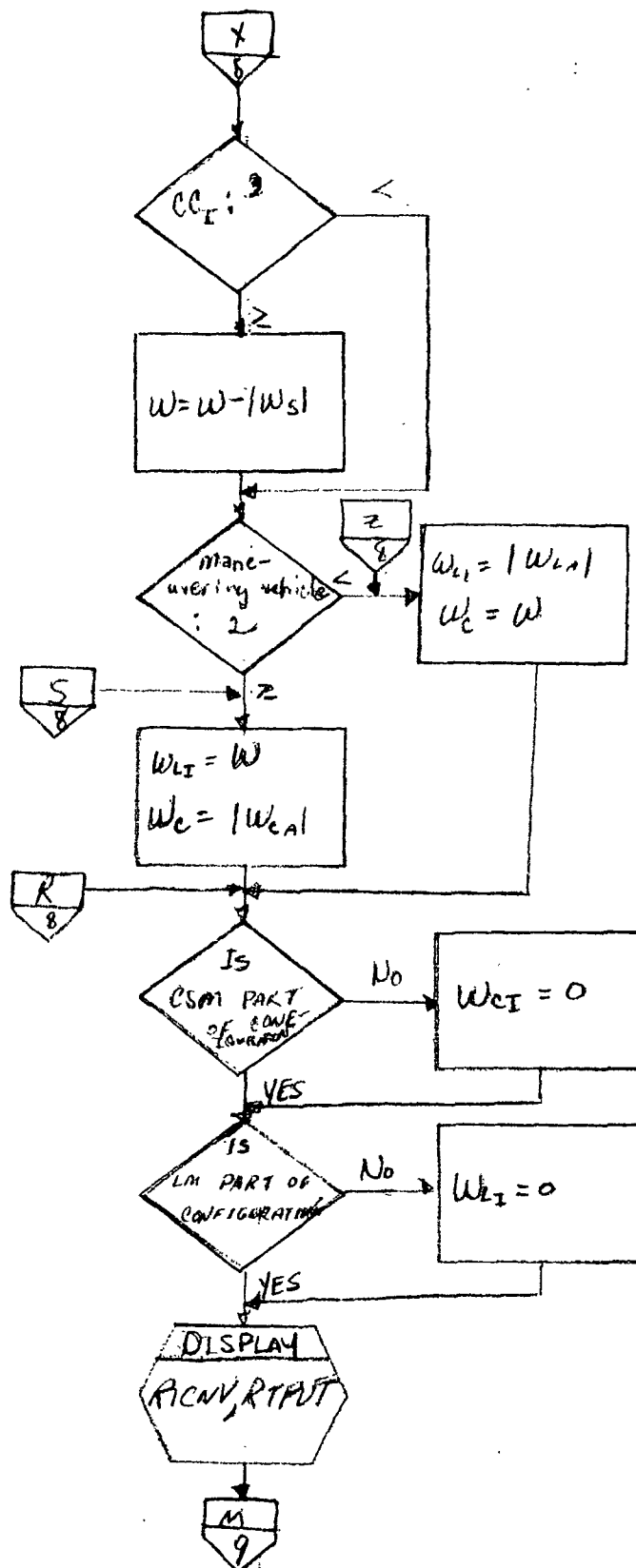


Figure PMDDMT-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMDDMT

DATE 8/22/67 (EORP)

REV 10/10/67

PAGE 15 of 16

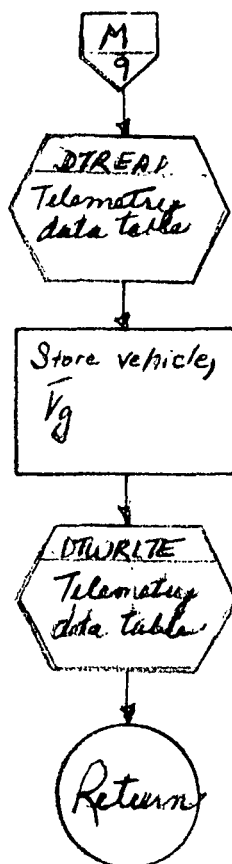


Figure PMDDMT-1. Flowchart.

(Sheet 9 of 10)

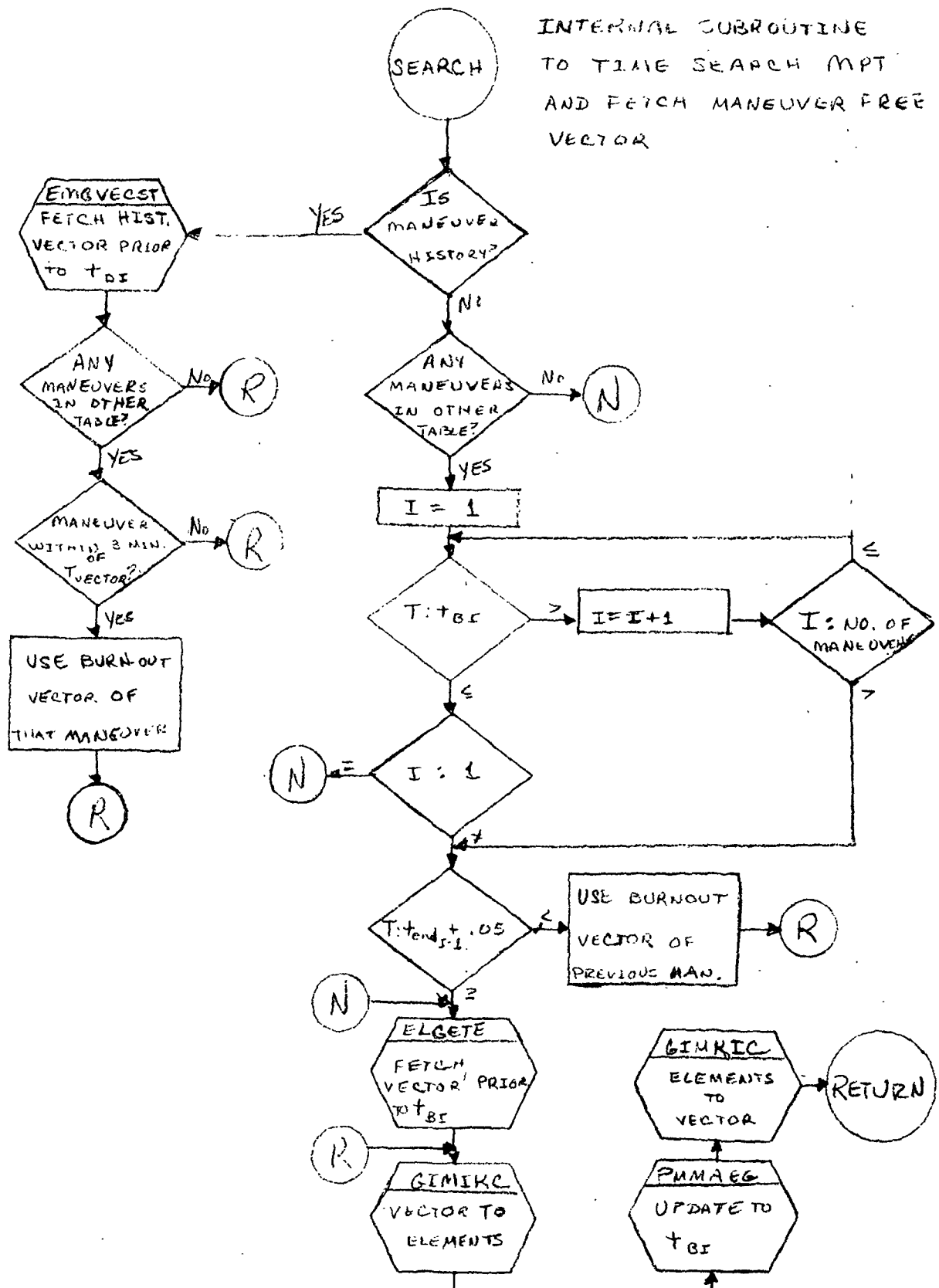


Figure PMDDMT-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMDDTVCC

DATE 12/11/67 (EORP)

REV

PAGE 1 of 4

- a. Symbol — PMDDTVCC
- b. Program Name — Two-impulse Corrective Combination Display
- c. Programming Group — Rendezvous Planning
- d. Programmer(s) — T. Hickey
- e. Language — Assembler
- f. Data Tables Used — PZTIPCCD
- g. MED's Used — None
- h. Writeup Completion Date — 11/9/67
- i. System Parameters — Table PMDDTVCC-1
- j. List of Items Displayed — Table PMDDTVCC-2
- k. Flowchart — Figure PMDDTVCC-1

FUNCTION

This display formats and displays the parameters associated with the Two-impulse Digitals (Corrective Combination) Display, and queues the Philco D/TV equipment for purposes of display. The MSK is 64.

USAGE

- a. Calling Sequence — ATTACH EP = PMDDTVCC
- b. Inputs — Data control word
- c. Outputs and Exit — Builds buffer load to Philco D/TV hardware.
Exit is via "RETURN" macro.
- d. Other Programs Used — RHFDFCNV
- e. Data Restrictions and Limits — None

METHOD

After reading the Two-impulse Corrective Combination Display Table (PZTIPCCD), a check is made to determine whether there are any plans in the table.

If there are none, another check is made to determine if the table is in an update status. If this is the case, a message entitled "TABLE BEING UPDATED" is formatted; if not, a message entitled "NO CORRECTIVE COMBINATION PLANS" is formatted.

If there are solutions in the table, all of the appropriate parameters are converted to the desired units, the data is formatted and placed in a buffer, and the buffer is then RTPUT to the Philco D/TV equipment for display purposes.

Approval	Approval
<i>Thomas M. Hickey</i>	<i>A. Amieatt</i>

Table PMDDTVCC-1. System Parameters

[illegible]

*Type
H — Fixed-point half word
F — Fixed-point full word
E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexidecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMDDTVCC

DATE 12/11/67 (EORP)

REV

PAGE 3 of 4

Table PMDDTVCC-2. List of Items Displayed

ITEM	DESCRIPTION	UNITS
CSM STA ID	Current CSM STA ID in MPT	
LM STA ID	Current LM STA ID in MPT	
GETNCC	Impulsive elapsed time of first maneuver of all solutions	HR: MIN: SEC
GMTNCC	Impulsive GMT of same	DY: HR: MIN: SEC
CODE	Numeric code specifying the plan number of each solution in the table	
GETNSR	Impulsive elapsed time of the second maneuver of the plan	HR: MIN: SEC
GMTNSR	Impulsive GMT of same	DY: HR: MIN: SEC
ΔV_T	Sum of the impulsive ΔV 's for both maneuvers plus an approximation of the ΔV required for terminal phase	FT/SEC
Δh	Height offset after the second maneuver	N. M.
$\Delta \theta$	Phase angle after the second maneuver	DEG
Δt	Time interval between first and second maneuvers	MIN.
VEH	Maneuver vehicle code (first solution only)	
T_{SLIP}	Amount of time terminal phase initiation will differ from the nominal if this solution is used	MIN.

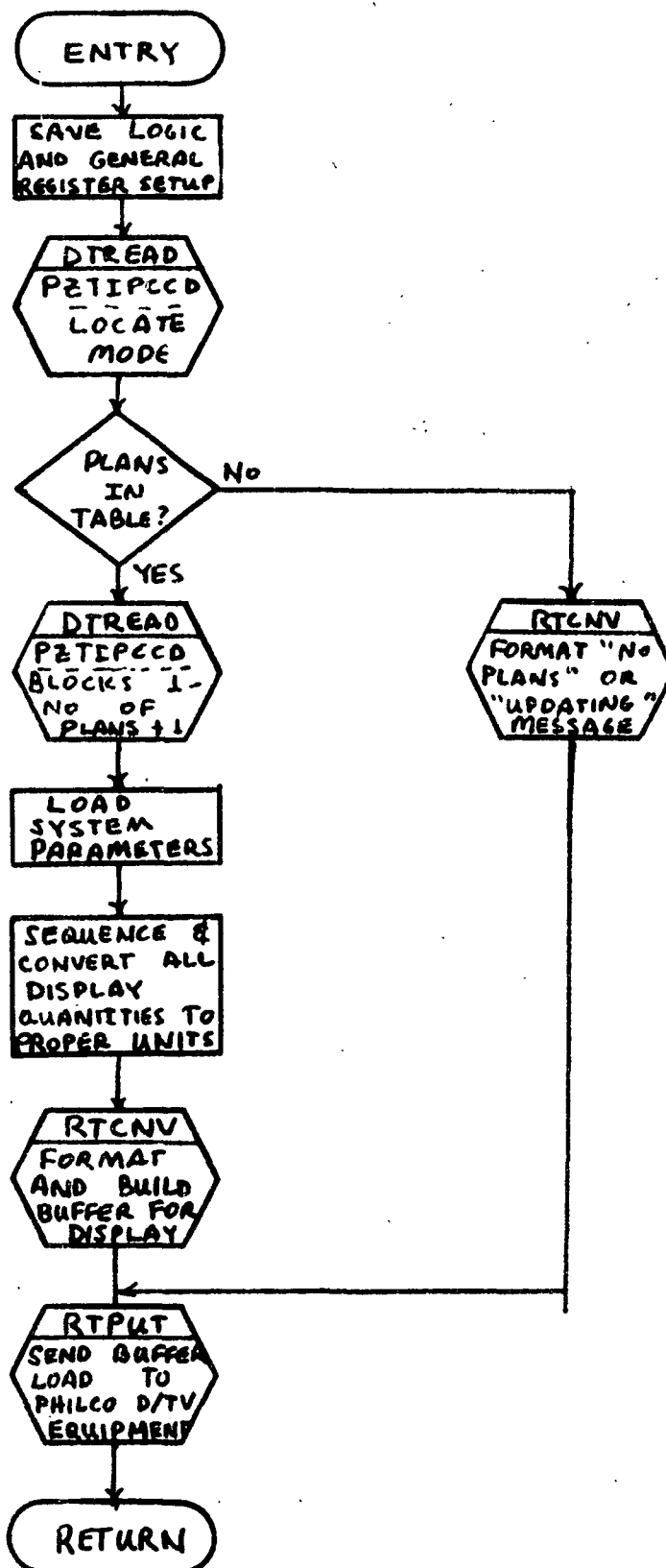


Figure PMDDTVCC-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMDGPM
DATE 12/11/67(206)
REV
PAGE 1 of 5

- a. Symbol — PMDGPM
- b. Program Name — General Purpose Maneuver Display Load Module
- c. Programming Group — Rendezvous Planning
- d. Programmer — T. Hickey
- e. Language — Assembler
- f. Data Tables Used — PZGPMDIS, EZCCSM or EZCS4B or EZCLEM
- g. MED's Used — None
- h. Writeup Completion Date — 11/10/67
- i. System Parameters — Table PMDGPM-1
- j. List of Items Displayed — Table PMDGPM-2
- k. Flowchart — Figure PMDGPM-1

FUNCTION

This load module formats and displays the parameters associated with the general purpose maneuver last computed. The MSK number is 48.

USAGE

- a. Calling Sequence — ATTACH EP = PMDGPM
- b. Inputs — Data control word
- c. Outputs and Exit — Builds buffer load to Philco D/TV hardware. Exit is by "RETURN" macro.
- d. Other Programs Used — RHFDFCNV
- e. Data Restrictions and Limits — None

METHOD

After reading the GPM display data table (PZGMDIS), a check is made to ensure that a maneuver is present. If no maneuver is in the table, a buffer load with a message entitled "NO MANEUVERS IN GPM TABLE" is formatted and RTPUT to the Philco D/TV equipment for displaying. If a maneuver is present, all of the desired parameters are converted to their proper units, all data is formatted as required, and the buffer is then RTPUT to the Philco D/TV equipment for display purposes.

Approval	Approval
<i>James M. Hickey</i>	<i>A. Ametti</i>

Table PMDGPm-1. System Parameters

[illegible]

*Type
H — Fixed-point half word
F — Fixed-point full word
E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMDGPM

DATE 12/11/67 (206)

REV

PAGE 3 of 5

Table PMDGPM-2. List of Items Displayed

ITEM	DESCRIPTION	UNITS
MAN VEH STA ID	Call letters of station used for last trajectory update of the maneuvering vehicle	
GETAV	GET of anchor vector for last trajectory update	HR:MIN:SEC
WEIGHT	Weight of vehicle at time of the maneuver	LBS.
CODE	Maneuver code	
h_A	Resultant apogee height	N. M.
λ_A	Apogee longitude after maneuver	DEG:MIN $\begin{pmatrix} E \\ W \end{pmatrix}$
ϕ_A	Geocentric latitude of apogee after maneuver	DEG:MIN $\begin{pmatrix} N \\ S \end{pmatrix}$
GET _A	GET of apogee after maneuver	HR:MIN:SEC
h_P	Resultant perigee height	N. M.
λ_P	Perigee longitude after maneuver	DEG:MIN $\begin{pmatrix} E \\ W \end{pmatrix}$
ϕ_P	Geocentric latitude of perigee after maneuver	DEG:MIN $\begin{pmatrix} N \\ S \end{pmatrix}$
GET _P	GET of perigee after maneuver	HR:MIN:SEC
a	Semi-major axis of resultant orbit	N. M.
e	Eccentricity of resultant orbit	
i	Inclination angle of resultant orbit	DEG.
λ_{AN}	Longitude of first ascending node resulting from the maneuver	DEG:MIN $\begin{pmatrix} E \\ W \end{pmatrix}$
Ω	Amount of apsidal shift of resultant orbit	DEG:MIN $\begin{pmatrix} E \\ W \end{pmatrix}$
GET	GET of the impulsive maneuver	HR:MIN:SEC
GETTH	Threshold time to find maneuver point	HR:MIN:SEC
h_{MAN}	Height of vehicle at impulsive maneuver point	N. M.
λ_{MAN}	Longitude of vehicle at impulsive maneuver point	DEG:MIN:SEC $\begin{pmatrix} E \\ W \end{pmatrix}$

IBM RTCC APOLLO PROGRAMMING SYSTEMS.

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMDGPM
DATE 12/11/67 (206)
REV
PAGE 4 of 5

Table PMDGPM-2. List of Items Displayed (Cont'd.)

ITEM	DESCRIPTION	UNITS
ϕ_{MAN}	Geocentric latitude of vehicle at impulsive maneuver point	DEG:MIN:SEC ^(N) _(S)
ΔV	Velocity increment required to perform maneuver	FT. /SEC.
P_H	Pitch attitude of vehicle - LVLH system	DEG.
Y_H	Yaw attitude of vehicle - LVLH system	DEG.
REV	Revolution number in which the maneuver is performed	

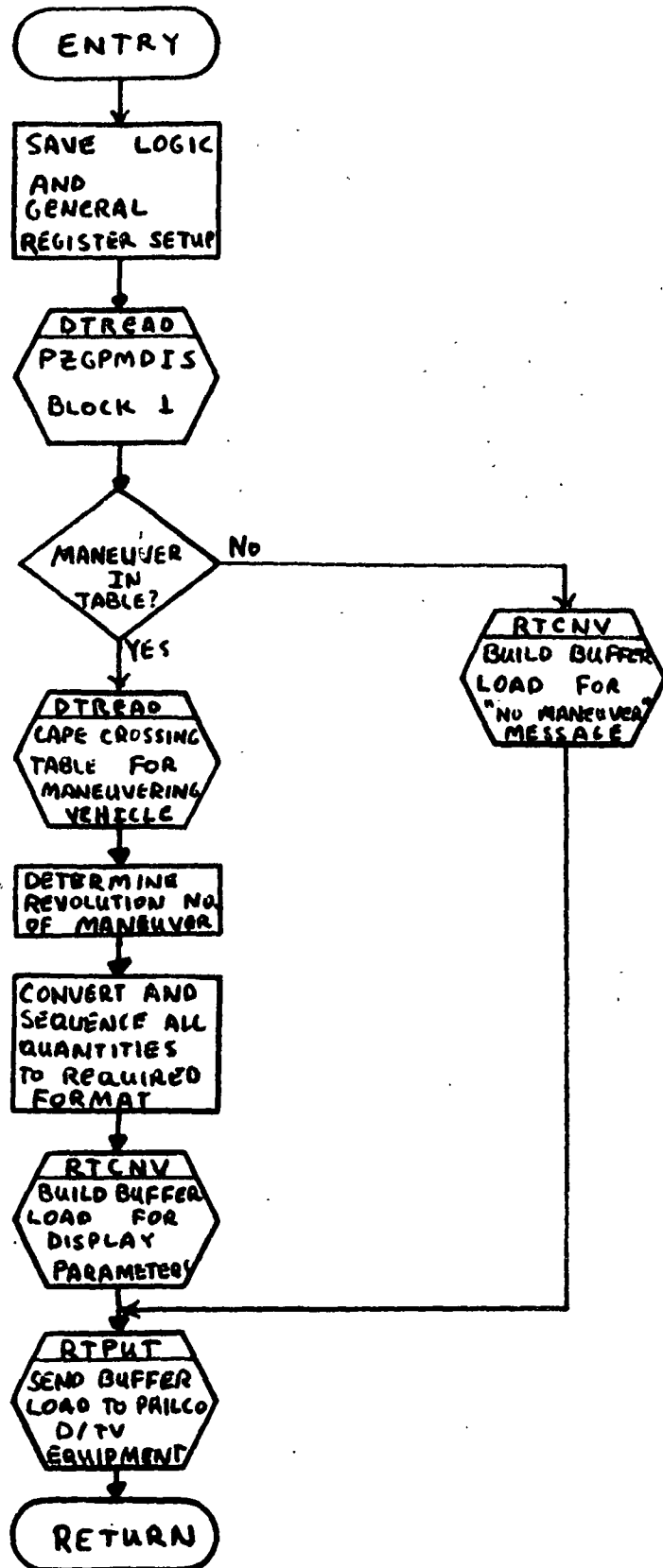


Figure PMDGPM-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMDMPT
DATE 8/31/67 (206)
REV
PAGE 1 of 4

- a. Symbol — PMDMPT
- b. Program Name — Mission Plan Table Display
- c. Programming Group — Mission Planning
- d. Programmer — E. Kramer
- e. Language — Symbolic
- f. Data Tables Used — PZMPTLEM
- g. MED's Used — None
- h. Writeup Completion Date — 8/4/67
- i. System Parameters — Table PMDMPT-1.
- j. List of Items Displayed — Table PMDMPT-2.
- k. Flowchart — Figure PMDMPT-1.

FUNCTION

PMDMPT formats and outputs to the D/TV hardware (via RTPUT) various quantities for each maneuver in the Mission Plan Table. See Table 2 for a description of these quantities.

USAGE

- a. Calling Sequence — ATTACH EP = PMDMPT
- b. Inputs — Data Control word
- c. Outputs and Exit — Outputs formatted data for display and returns
- d. Other Programs Used — RHFDFCNV
- e. Data Restrictions and Limits — None
- f. Error — NA
- g. Other — If a trajectory update is in process the following message is displayed: "UPDATE IN PROCESS".

METHOD

The required data is taken from the MPT and formatted using the RTCNV and RTFMT macros, and the formatted buffer is RTPUT to the D/TV buffer for displaying.

Approval	Approval
<i>[Signature]</i>	<i>[Signature]</i>

Table PMDMPT-1. System Parameters

[illegible]

*Type
H — Fixed-point half word
F — Fixed-point full word
E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMDMPT

DATE 8/31/67 (206

REV

PAGE 3 of 4

Table PMDMPT-2. List of Items Displayed

ITEM	DESCRIPTION	UNITS
LEM STAID	Call Letters of Station Used for Last Update of CSM Trajectory	--
GETAV	GET of Anchor Vector for CSM Update	h:m:s
GETBI	GET of the Burn Initiation	h:m:s $\frac{\text{sec.}}{10}$
ΔT	Time Between Maneuvers	m:s
ΔV	Incremental Velocity Required to Complete the Maneuver	ft/sec.
ΔV_{REM}	Maneuvering Vehicle Velocity Capability Remaining after the Maneuver for Indicated Thruster	ft/sec.
H_A	Height of Apogee above Spherical Earth After the Maneuver	NM
H_P	Height of Perigee above Spherical Earth After the Maneuver	NM
CODE	Code Assigned to the Maneuver in the MPT	--

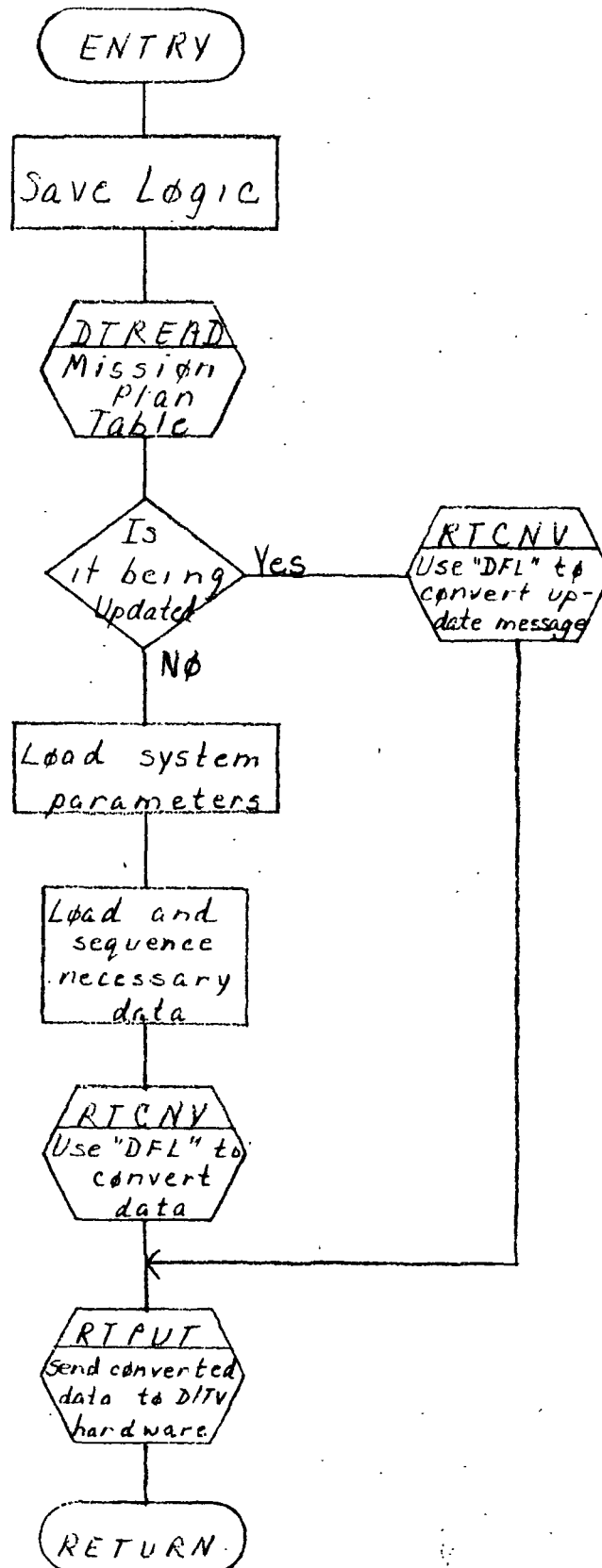


Figure PMDMPT-1. Flowchart

- a. Symbol — PMDPAD
- b. Program Name — Perigee Adjust Display
- c. Programming Group — Rendezvous Planning
- d. Programmer — C. Peabody
- e. Language — Assembler
- f. Data Tables Used — PZPADDIS
- g. MED's Used —
- h. Writeup Completion Date — 6/14/67
- i. Flowchart — Figure PMDPAD-1

FUNCTION

PMDPAD displays burn time, velocity increment, burn duration, true anomaly, height above spherical earth, and height of resultant apogee for six maneuvers, any one of which will cause perigee to be raised to some requested value. These calculations are made in PMMPAD.

USAGE

- a. Calling Sequence — LINK EP=PMDPAD
- b. Inputs — Perigee Adjust Display Table PZPADDIS
- c. Outputs and Exit — Display buffer to D/TV system
- d. Other Programs Used — None
- e. Data Restrictions and Limits — None
- f. Error — None

METHOD

The first word of the data table is compared to zero. If it is equal, the perigee requested is less than the current perigee and current perigee is displayed. If not, the desired perigee is converted from Earth radii to nautical miles. The following conversions are done for all six maneuvers: ΔV from Er./hr. to ft/sec, true anomaly from radians to degrees, height above Earth from Er. to miles, height at resultant apogee from Er. to miles. The above information plus the timestep, the GMT of burn initiate, and the burn time are then displayed. However, if the GMT of burn initiate is negative, there is no maneuver and thus only the time is displayed.

Approval	Approval
<i>C. Peabody</i>	<i>Q. Steiner</i>

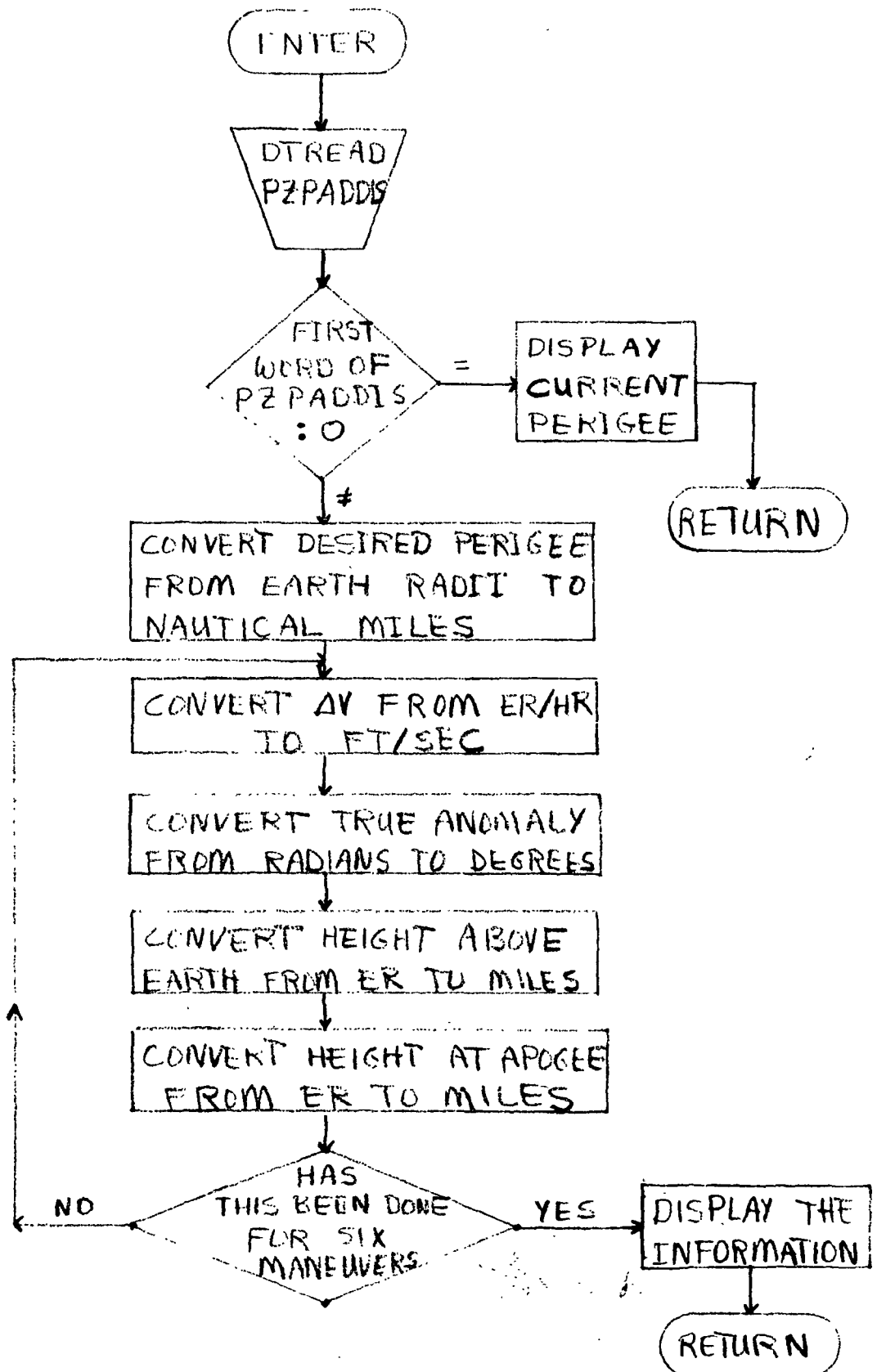


Figure PMDPAD-1. Flowchart

Data Table Formats

Symbolic Name — PZPADDISProgrammer — C. PeabodyTable Name — Perigee AdjustInitialization of any Element Required —Dimensions

Blocks — 1

Words/Blocks — 50 double words

Total Words — 50 double words

General Description and Usage

PMDPAD will be attached from the display supervisor. Information for six maneuvers to raise perigee to the requested value will be displayed. This information is in the PZPADDIS table.

Approval	Approval
<i>C. Peabody</i>	<i>J. Little</i>

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMDPAD
DATE 7/28/67 (EOR)
REV
PAGE 4 of 4

Detailed Contents

Block	Word	Byte	Bit	Type*	Units	Description and Range
	1			D		Solution Indicator
	2			D	hrs.	Time of Elements
	3			D	hrs.	Thresh Time
	4			D	hrs.	Time Step
	5			D	Er	Apogee Height
	6			D	Er	Perigee Height
	7			D		Not Used
	8			D	rad.	Pitch at Ignition
	9			D	hrs.	Burn Time
	10			D	Er/hr	Velocity Increment
	11			D	hrs.	Burn Duration
	12			D	rad.	True Anomaly
	13			D	Er.	Height Above Earth
	14			D	Er.	Height of Resultant Apogee
	15-21			D		Same as Words 8-14 (for 2nd Maneuver)
	22-28			D		Same as Words 8-14 (for 3rd Maneuver)
	29-35			D		Same as Words 8-14 (for 4th Maneuver)
	36-42			D		Same as Words 8-14 (for 5th Maneuver)
	43-49			D		Same as Words 8-14 (for 6th Maneuver)
	50			D		Not Used

*Type

H — Fixed-point half word
F — Fixed-point full wordE — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK:MISSION SYSTEMS - AS-200 Mission Planning

2. PMDRET
DATE 12/11/67 (EOR)
REV
PAGE 1 of 4

- a. Symbol — PMDRET
- b. Program Name — Rendezvous Evaluation Display Load Module
- c. Programming Group — Rendezvous Planning
- d. Programmer(s) — R. Barrett, T. Hickey
- e. Language — Assembler
- f. Data Tables Used — PZDKIT, EZETVMED, GZGENCSN
- g. MED's Used — U06
- h. Writeup Completion Date — 11/17/67
- i. List of Displayed Parameters — Table PMDRET-1
- j. Flowchart — Figure PMDRET-1

FUNCTION

This load module formats and builds a buffer containing all appropriate parameters for the Rendezvous Evaluation display, according to the latest U06 MED request. It also queues the Philco D/TV equipment to display this buffer load.

USAGE

- a. Calling Sequence — ATTACH EP = PMDRET
- b. Inputs — Data control word
- c. Outputs and Exit — Builds buffer load to Philco D/TV hardware. Exit is via "RETURN" macro.
- d. Other Programs Used — RHFDFCNV
- e. Data Restrictions and Limits — None

METHOD

After reading the DKI planning table (PZDKIT), a check is made to determine whether there are any solutions available. If not, the format is set to display either a "NO PLANS IN TABLE" or "UPDATE IN PROCESS" message. If the plan number requested is contained in the table, all of the appropriate parameters are converted to the desired units, the data is formatted and placed in a buffer, and the buffer is then RTPUT to the Philco D/TV equipment for display purposes.

Approval	Approval
<i>Thomas M. Hickey</i>	<i>A. Quinn H.</i>



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMDRET
 DATE 12/11/67 (E)
 REV
 PAGE 2 of 4

Table PMDRET-1. List of Displayed Parameters

ITEM	DESCRIPTION	UNITS
ID	Row number of plan from Rendezvous Planning Table	
M	Orbit number in which rendezvous is to occur	
GET	Elapsed impulsive time of the maneuver	HR:MIN:SEC
Δt	Amount of time between each maneuver of the same vehicle	MIN:SEC
Δv	Velocity increment for the maneuver	FT/SEC
VEH	Maneuvering vehicle code	
ID	Maneuver purpose code of this maneuver	
θ	Phase angle after the maneuver	DEG.
Δh	Height offset after the maneuver	N. M.
h_p / h_a	Height of perigee and height of apogee	N. M.

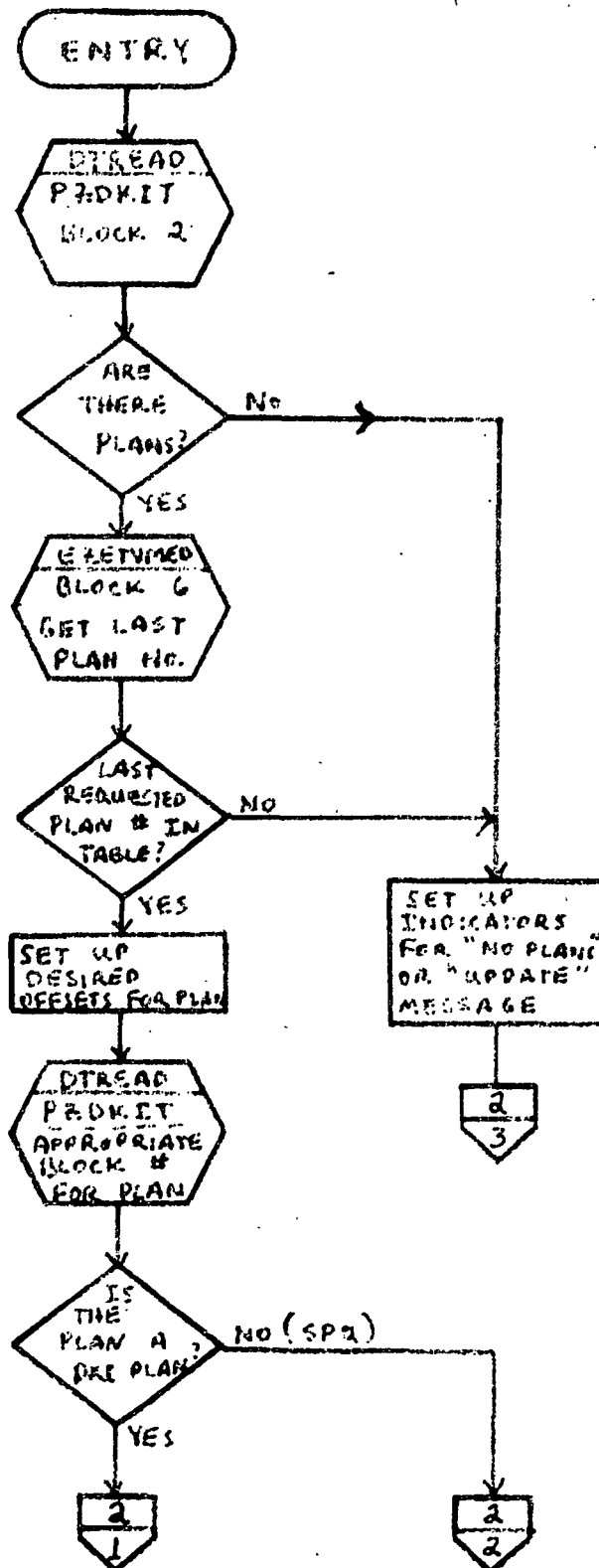


Figure PMDRET-1. Flowchart

(Sheet 1 of 2)

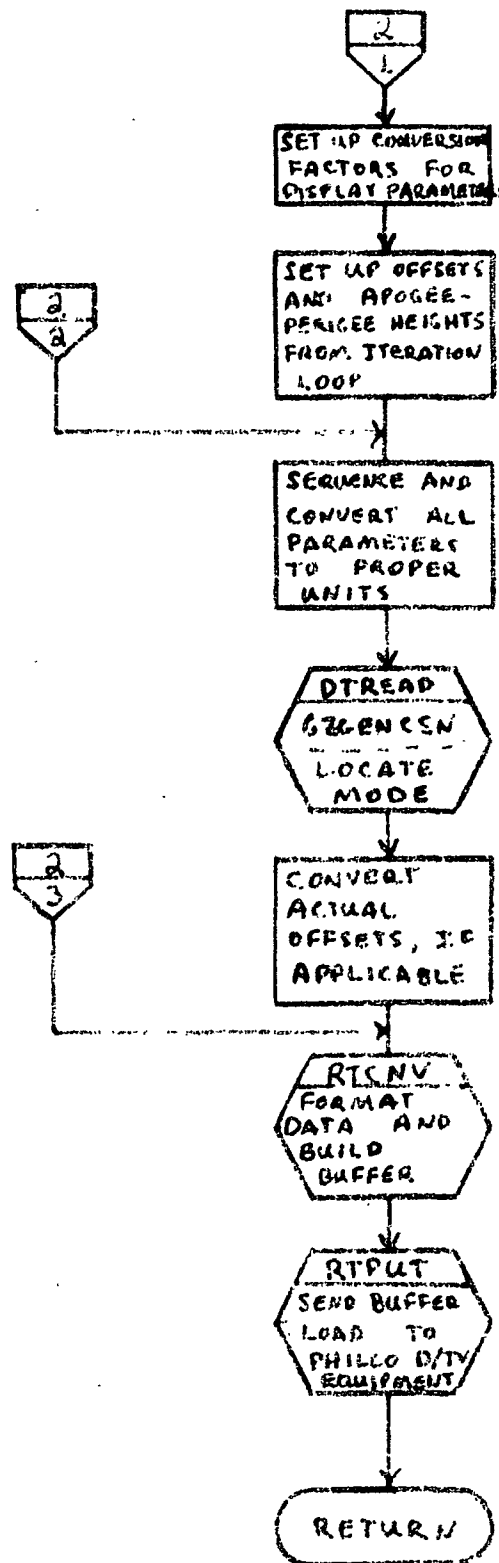


Figure PMDRET-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMDRPT
DATE 12/11/67 (EORP)
REV
PAGE 1 of 3

- a. Symbol — PMDRPT
- b. Program Name — Rendezvous Plan Table Display
- c. Programming Group — Rendezvous Planning
- d. Programmer(s) — M. Secondo
- e. Language — Assembler
- f. Data Tables Used — PZDKIT, PZSLVTAR
- g. MED's Used — None
- h. Writeup Completion Date — 11/15/67
- i. System Parameters — Table PMDRPT-1
- j. Flowchart — Figure PMDRPT-1

FUNCTION

PMDRPT provides display information for a maximum of seven DKI rendezvous plans.

USAGE

- a. Calling Sequence — RTATTACH TASK=GTISPLAY, EP=GMSTAP, ID=7, PARAM = (Word, 4, Fword, 4)
- b. Inputs — TVGUIDE word
- c. Outputs and Exit — D/TV Formatted display. Exit is by return to calling program.
- d. Other Programs Used — RHFDFCNV
- e. Data Restrictions and Limits — None
- f. Error — None

METHOD

Upon entry to the program, a data table, PZDKIT, is read to determine whether a DKI plan or Coelliptic Sequence plan is requested. For a Coelliptic Sequence entry, a message is displayed indicating that no rendezvous plans are available; if a DKI plan is being computed, the message will state that the table is being updated. After all of the DKI plan computations are completed, the display quantities are converted to the proper units and transmitted to the D/TV.

For each plan, the quantities displayed include: (1) an identification number for that plan, (2) the requested M-LINE, (3) ΔV for each vehicle, (4) GET, (5) the maneuver line points for each maneuver scheduled, and (6) the amount of time that liftoff can be delayed and still achieve rendezvous.

Approval	Approval
<i>Mitchell R. Secondo</i>	<i>A. Amietti</i>

Table PMDRPT-1. System Parameters

[illegible]

E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal
 Type
 H — Fixed-point half word
 F — Fixed-point full word

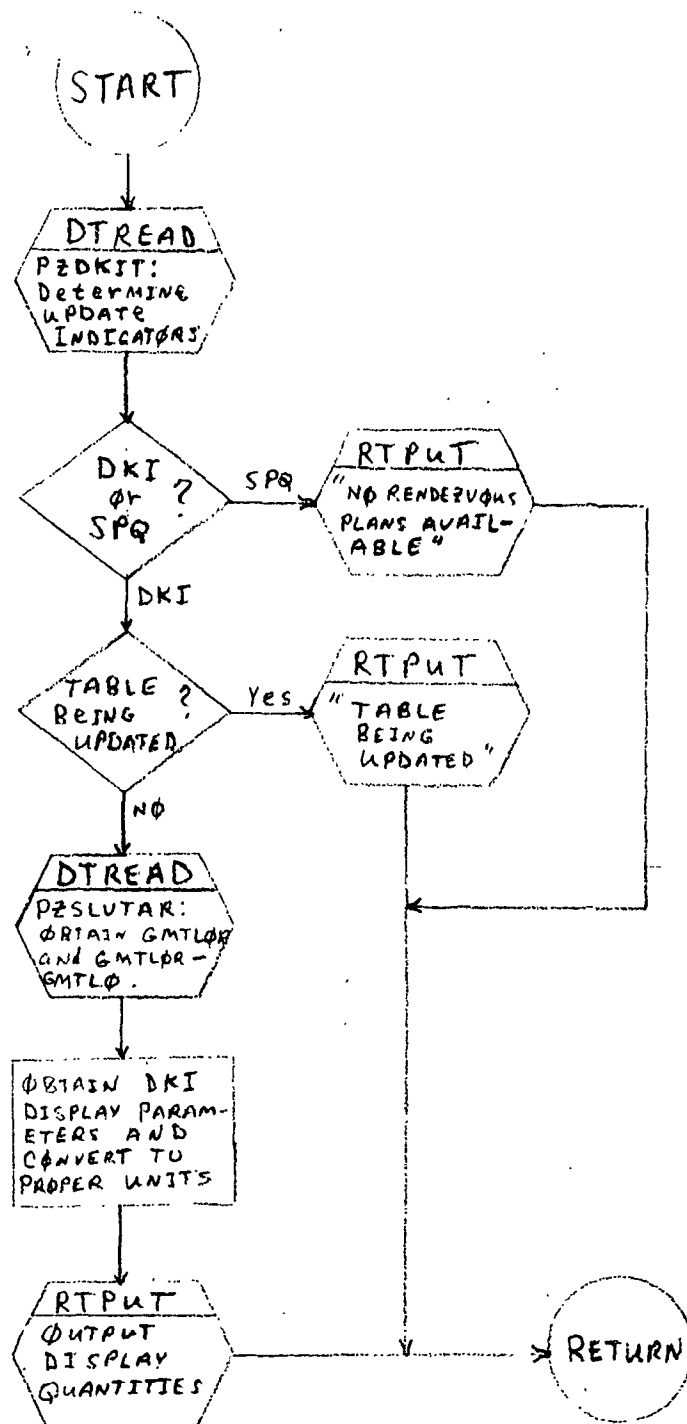


Figure PMDRPT-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS -AS-200 Mission Planning

2. PMDTAR
DATE 7/31/67(EORP)
REV
PAGE 1 of 2

- a. Symbol — PMDTAR
- b. Program Name — Saturn Launch Vehicle (SLV) Targeting Display
Load Module
- c. Programming Group — Rendezvous Planning
- d. Programmer(s) — T. Copper
- e. Language — Assembler
- f. Data Tables Used — PZSLVTAR
- g. MED's Used — None
- h. Writeup Completion Date — 5/24/67
- i. Flowchart — Figure PMDTAR-1

FUNCTION

PMDTAR generates a format for displaying the SLV targeting parameters, and transmits the format to the D/TV system.

USAGE

- a. Calling Sequence — RTATTACH TASK = GTISPLAY, EP = GMSNAP, I3 = 3, PARAM = (WORD, 4, FWORD, 4)
- b. Inputs — Data Table (PZSLVTAR), TV GUIDE WORD
- c. Outputs and Exit — Output is on D/TV screen. Control is return to calling program.
- d. Other Programs Used — DTREAD, RTCNV, RTPUT
- e. Data Restrictions and Limits — None
- f. Error — None

METHOD

When PMDTAR is entered, it reads in the input SLV Targeting Data Table (PZSLVTAR) using a DTREAD. By using the data format language package, the input data is converted to the D/TV display format and stored in a buffer area. After all data has been formatted, the buffer area is put out to the D/TV hardware to display the SLV Targeting Digital Display by an RTPUT.

Approval	Approval
<i>T. Copper</i>	<i>Arthur Amietti</i>

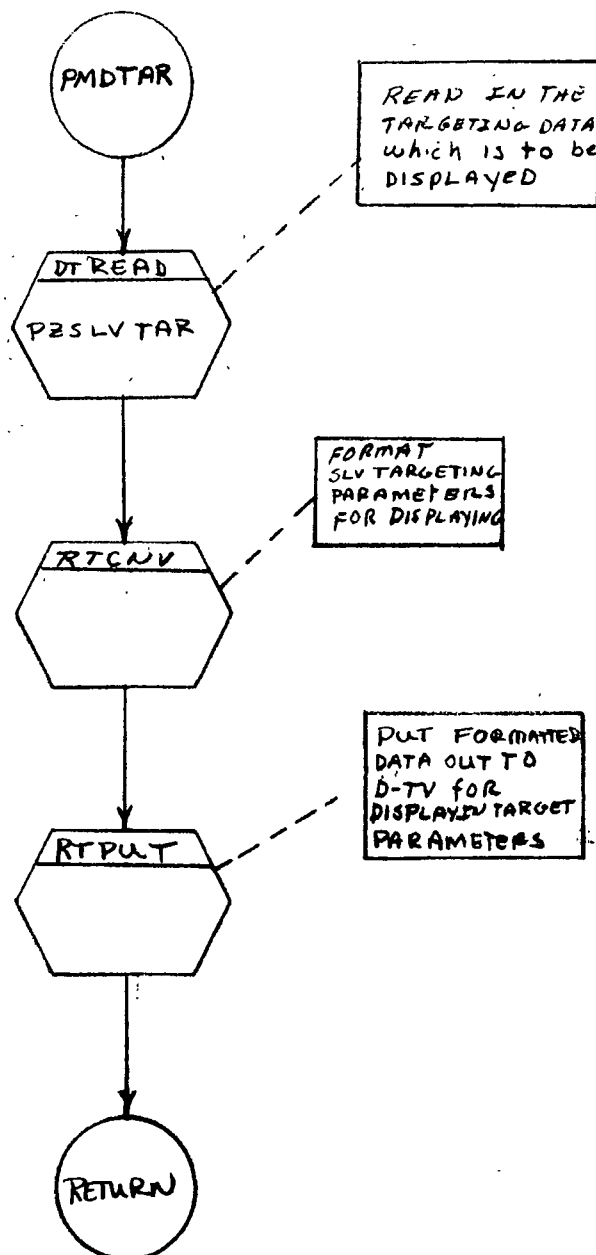


Figure PMDTAR-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMDTIMP
DATE 12/11/67 (EORP)
REV
PAGE 1 of 5

- a. Symbol — PMDTIMP
- b. Program Name — Two-impulse Multiple Solution Display
- c. Programming Group — Rendezvous Planning
- d. Programmer(s) — T. Hickey
- e. Language — Assembler
- f. Data Tables Used — PZTIPREG, GZGENCSN
- g. MED's Used — None
- h. Writeup Completion Date — 11/17/67
- i. System Parameters — Table PMDTIMP-1
- j. List of Items Displayed — Table PMDTIMP-2
- k. Flowchart — Figure PMDTIMP-1

FUNCTION

This display formats and queues the Philco D/TV equipment for purposes of displaying the two-impulse multiple solution display parameters. The MSK is 63.

USAGE

- a. Calling Sequence — ATTACH EP = PMDTIMP
- b. Inputs — Data control word
- c. Outputs and Exit — Builds buffer load to Philco D/TV hardware.
Exit is via "RETURN" macro.
- d. Other Programs Used — RHFDFCNV
- e. Data Restrictions and Limits — None

METHOD

After reading the Two-impulse Multiple Solution Display Table (PZTIPREG), a check is made to determine whether there are any plans in the table. If there are none, another check is made to determine if the table is in an update status. If this is the case, a message entitled "TABLE BEING UPDATED" is formatted; if not, a message entitled "NO TWO IMPULSE PLANS AVAILABLE" is formatted.

If there are solutions in the table, all of the appropriate parameters are converted to the desired units, the data is formatted and placed in a buffer, and the buffer is then RTPUT to the Philco D/TV equipment for display purposes.

Approval	Approval
<i>Thomas M. Hickey</i>	<i>A. Amietti</i>

Table PMDTIMP-1. System Parameters

[illegible]

*Type
 H — Fixed-point half word
 F — Fixed-point full word
 E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMDTIMP

DATE 12/11/67 (EORP)

REV

PAGE 3 of 5

Table PMDTIMP-2. List of Items Displayed

ITEM	DESCRIPTION	UNITS
CSM STA ID	Current CSM STA ID in MPT	HR:MIN:SEC
LM STA ID	Current LM STA ID in MPT	
GETTH CSM	Ground elapsed time tag of vector from CSM ephemeris used for computations	
GETTH LM	Same as above except for LM	HR:MIN:SEC
MAN VEH	Maneuvering vehicle indicator	
ω_T	Desired target terminal phase transfer angle	DEG.
$\Delta\theta$	Desired phase angle after second maneuver	DEG.
Δh	Desired height offset after second maneuver	N. M.
GET ₁	Elapsed time of the frozen maneuver	HR:MIN:SEC
GMT	GMT of frozen maneuver	DY:HR:MIN:SEC
OPTION	Word indicating whether the first, second, or both maneuver time(s) is(are) frozen	
MINUTES UNTIL	Number of minutes until the environment change indicated from the frozen maneuver of the first solution	MINUTES—(DAY NIGHT)
ΔV_1	Impulsive velocity increment of the frozen maneuver	FT/SEC

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMDTIMP
DATE 12/11/67 (EOB)
REV
PAGE 4 of 5

Table PMDTIMP-2. List of Items Displayed (Cont.)

ITEM	DESCRIPTION	UNITS
YAW ₁	Yaw altitude of the frozen maneuver	DEG.
PITCH ₁	Pitch altitude of the frozen maneuver	DEG.
GET ₂	Elapsed time of the variable maneuver	HR:MIN:SEC
ΔV_2	Impulsive velocity increment of the variable maneuver	FT/SEC
YAW ₂	Yaw altitude of the variable maneuver	DEG.
PITCH ₂	Pitch altitude of the variable maneuver	DEG.
L	Letter D or N indicating whether the variable maneuver occurs in daylight or darkness	
C	Numeric code specifying the plan number of each solution in the table	

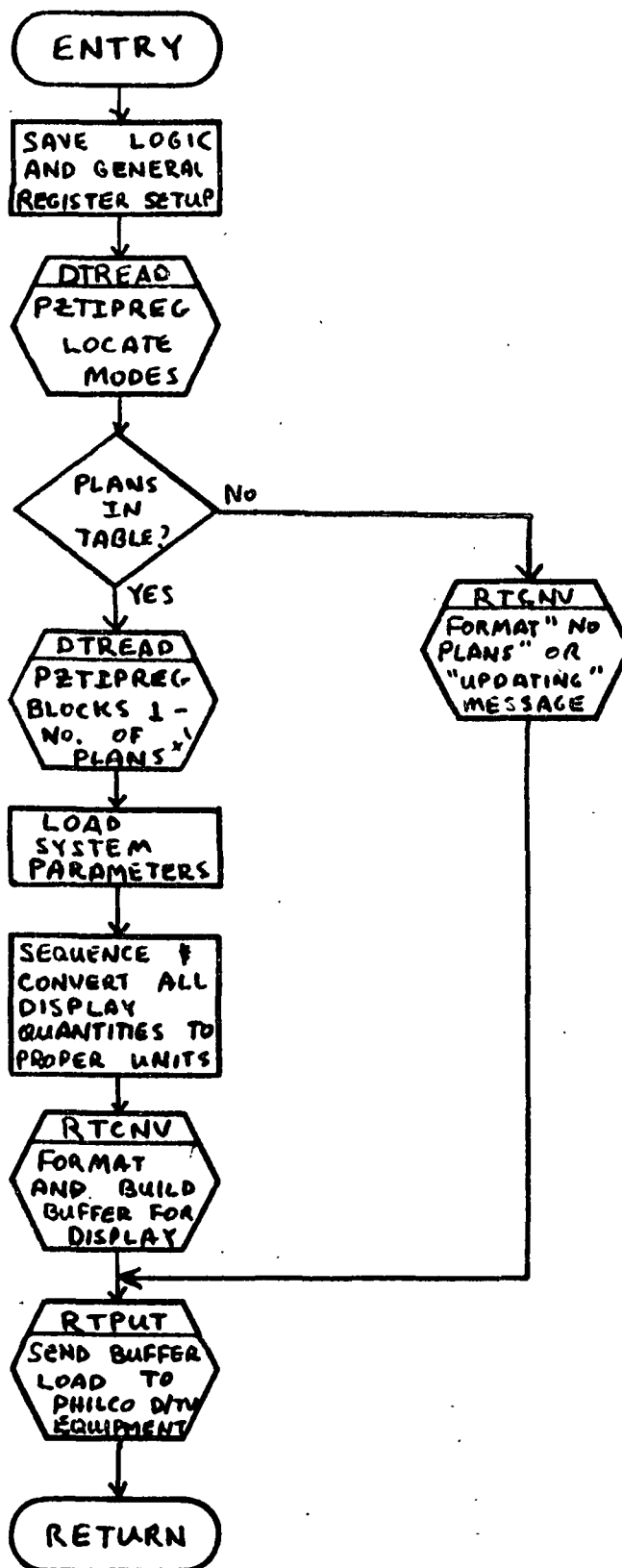


Figure PMDTIMP-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMDTIPSS
DATE 7/31/67 (EORP)
REV
PAGE 1 of 6

- a. Symbol — PMDTIPSS
- b. Program Name — Two-impulse Single Solution Display
- c. Programming Group — Rendezvous Planning
- d. Programmer(s) — J. A. Steed
- e. Language — Assembler
- f. Data Tables Used — PZTIPSS
- g. MED's Used — K31 (initializes the math module which fills PZTIPSS)
- h. Writeup Completion Date — 6/14/67
- i. System Parameters — Table PMDTIPSS-1
- j. Data Table Formats — Table PZTIPSS-2
- k. Flowchart — Figure PMDTIPSS-1

FUNCTION

PMDTIPSS displays two-impulse single solution information, which includes expanded and detailed information concerning a set of maneuvers.

USAGE

- a. Calling Sequence — LINK EP = PMDTIPSS
- b. Inputs — Internal Format Number; Data Table, PZTIPSS.
- c. Outputs and Exit — Display. Return to calling program
- d. Other Programs Used — RHFDFCNV
- e. Data Restrictions and Limits — See Table PZTIPSS-2
- f. Error — None

METHOD

PMDTIPSS reads in the data table PZTIPSS. If there is no data in the table, a message to that effect is displayed using DFL. If there is data, the necessary conversions are made, and the desired information is displayed.

Approval	Approval
J. A. Steed	<i>[Signature]</i>

Table PMDTIPSS-1. System Parameters

[illegible]

*Type
H — Fixed-point half word
F — Fixed-point full word
E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

Data Table Formats

Symbolic Name — PZTIPSS

Programmer — T. M. Hickey

Table Name — Two-impulse Single Solution

Initialization of any Element Required — No

Dimensions

Blocks — 1
Words/Blocks — 103 double
Total Words — 103 double

General Description and Usage

Used only by the Two-impulse Single Solution Display, this program contains all necessary quantities for that display. It is not used for any other purpose.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMDTIPSS
 DATE 7/31/67 (EORP)
 REV
 PAGE 4 of 6

Detailed Contents

Block	Word	Byte	Bit	Type*	Units	Description and Range
1	1			D		LM STAID
	2			D		CSM STAID
	3			D	hrs.	LM GETTHS
	4			D	hrs.	CSM GETTHS
	5	1-2		H		Man. veh. code 1 = CSM 3 = LM 2 = S-IVB
		3-4		H		Mode 1 = LOS 2 = TGI
		5-6		H		ID = no. of solution 1 = MS
		7-8		H		Table requested 2 = CC
	6			D	rad.	$\Delta\theta$ = phase offset
	7				CUL	ΔH = height offset
	8				rad.	ωT
	9				rad.	Δ pitch
	10				hrs.	Time of first man. - GET
	11				rad.	E_{HOR} of first man.
	12				ERH	ΔV - impulsive for first man.
	13				rad.	Pitch - impulsive for first man.
	14				rad.	Yaw - impulsive for first man.
	15				ERH	V_x
	16				ERH	V_y
	17				ERH	V_z
	18				rad.	Pitch of active veh. along given LOS
	19				rad.	Yaw of active veh. along given LOS
	20				ERH	$\dot{X}$
	21				ERH	$\dot{Y}$
	22				ERH	$\dot{Z}$
	23				hrs.	ΔT + F - A
	24				hrs.	ΔT + L - R
	25				hrs.	ΔT + D - U

*Type

H - Fixed-point half word
F - Fixed-point full wordE - Single-precision (floating-point)
D - Double-precision (floating-point)
X - Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMDTIPSS
DATE 7/31/67 (EORP)
REV
PAGE 5 of 6

Detailed Contents

Block	Word	Byte	Bit	Type*	Units	Description and Range
	26			D		Min. until daylight or darkness
	27				CUL	Height of apogee after first man.
	28				CUL	Height of perigee after first man.
	29				hrs.	GET of approach data
	30				rad.	Target Azimuth + = L - = R
	31				rad.	Target elevation + = U - = D
	32				CUL	Range
	33				ERH	Range rate + L
	34				CUL	X - T
	35				CUL	Z + B - A
	36				CUL	Y + R - L
	37-52					2 duplicates of 29-36
	53-103					Duplicate of 10-52 with 29-36 four times

*Type

H - Fixed-point half word
F - Fixed-point full wordE - Single-precision (floating-point)
D - Double-precision (floating-point)
X - Hexadecimal

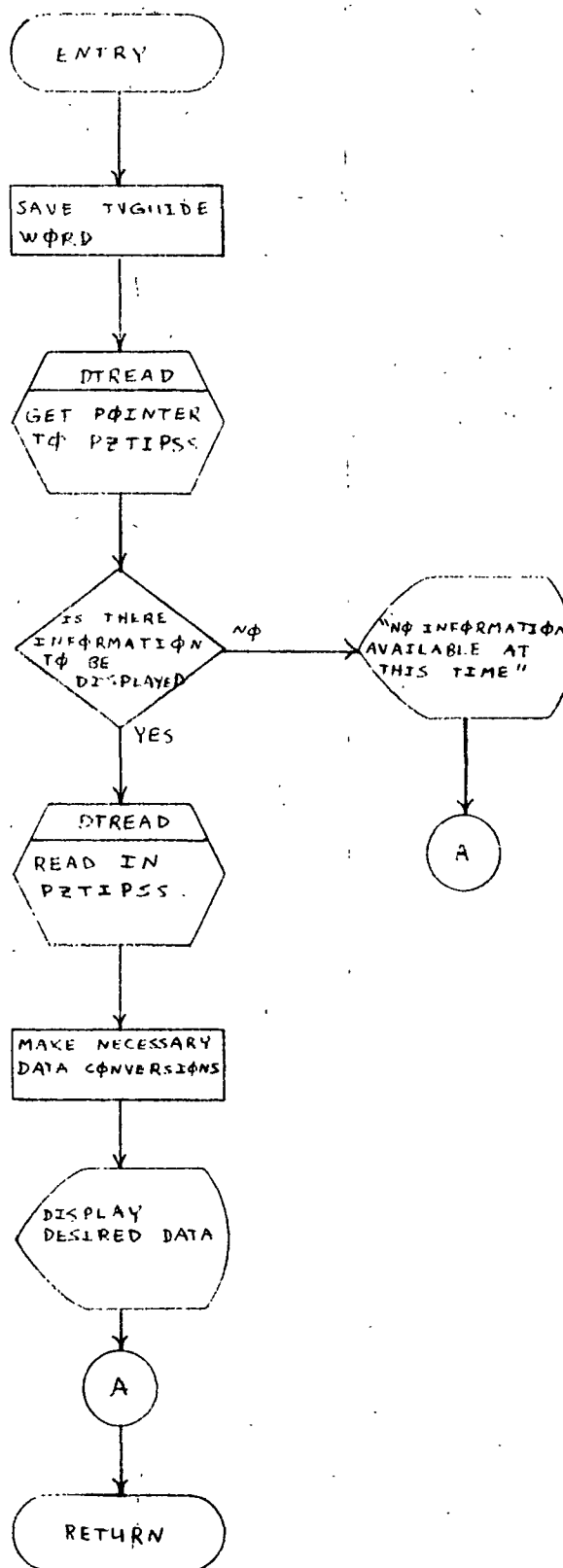


Figure PMDTIPSS-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

2. PMMAEG
 DATE 10/21/68
 REV
 PAGE 1 of 54
 MISSION EORP

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

PMMAEG
 EARTH ANALYTIC EPHEMERIS GENERATOR

REFERENCES

1. Programming Group — Rendezvous Planning
2. Programmers — T. Copper and J. Myers
3. Language — Assembler
4. Data Tables Used — None
5. MED's Used — None
6. Header — Table 1
7. Input Block Format — Table 2
8. Output Block Format — Table 3
9. PMMAEG Flowchart — Figure 1
10. TATSTOR Flowchart — Figure 2
11. DRAG Flowchart — Figure 3

FUNCTION

Given an initial set of orbit-defining elements (t, a, e, i, l, g, h), the AEG will initialize and/or update those elements to a specified time, accounting for the first three (or combinations thereof) surface spherical zonal harmonics (J_2 , J_3 , J_4), with or without drag.

USAGE

1. Calling Sequence — LINK EP=PMMAEG, PARAM=(HEADER, INPUT, OUTPUT)
2. Inputs — First two parameters of parameter list (see Note)
3. Outputs and Exit — Third parameter in parameter list (see Note)
4. Other Programs Used — None
5. Data Restrictions and Limits — See Note
6. Error — See Note
7. Note —

The AEG, PMMAEG, is being set up so that it can be called by using the LINK or ATTACH macro. The location of the first byte of the input/output tables, which must be located on doubleword boundaries, will be communicated to the AEG via key word PARAM. If ATTACH is used and a WAIT macro is issued by the user, the user must give key word ECB with the ATTACH so that the supervisor can do the posting; there will be no provisions for the AEG to issue a POST macro.

The parameter list must have the following format:

PARAM=HEADER, INPUT, OUTPUT

Approval	Approval
<i>J. Copper</i>	<i>M.E. Jones</i>

HEADER consists of eight bytes.
INPUT is one or more input blocks.
OUTPUT is one or more output blocks.

METHOD

In addition to updating a set of orbit-defining elements to a desired time, the AEG has built-in subroutines to obtain other desired end conditions. These routines are generally known as TIMA (time of arrival) and TAU (phase angle related). There are six specific options available as follows.

1. Mean Anomaly (l) — this option provides for the computation of the arrival time at any desired mean anomaly. A given mean anomaly occurs once every orbit; hence, an orbit counter (ΔN) is provided; it denotes the number of times to pass through the reference line ($l=0$; commonly known as peri-focus). For nearly circular orbits (e (eccentricity) $< .005$) the mean anomaly becomes ill defined, and failure to converge may occur.
2. Argument of Latitude (u) — this option corresponds to the mean anomaly except that the reference line ($u=0$) is known as the ascending node, and the argument of latitude remains well defined regardless of eccentricity. However, it would become unstable for nearly equatorial orbits.
3. Maneuver Line (NA) — this option provides the ability to advance a requested number (ΔN) of maneuver line crossings from a reference maneuver line. This reference line is defined by an argument of latitude (u) and a number (NA). The NA is a multiple of .5 with the whole numbers (e.g., 2.0) representing apofocal ends of the maneuver line, and the fractional numbers (e.g., 2.5) representing perifocal ends. The maneuver line would be fixed in the orbital plane except that it is allowed to move at the apsidal precession rate ($\dot{g}$). Hence, successive crossings of the maneuver line are separated by a central angle of π plus a small precession factor.
4. Phase Angle (θ_r) — this option (as well as the next two) requires at least two AEG blocks. The current block will be advanced to the output time from the previous blocks and the vehicles will be at time match. The central angle difference (relative to the previous vehicle) between the two vehicles will then be computed, and this is the phase angle. (It could be noted that if the vehicles are not coplanar, the angle is their relative difference from a common node. If they are coplanar the phase angle becomes the difference in arguments of latitude).
5. Position Match — in effect this option takes the current block to the time where the phase angle is zero, thus yielding position match. This is accomplished by taking the current block to time match, computing the phase angle, and then backing the current vehicle through the phase angle. For this purpose the phase angle is considered to be $0 \leq \theta_r < 2\pi$. The phase angle and the height difference are returned.
6. Position Match — this option is the same as (5) except that for position match the phase angle is considered to be $-\pi < \theta_r \leq \pi$.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMAEG
 DATE 10/21/68
 REV
 PAGE 3 of 54
 MISSION EORP

Table 1. Header

All angles are in the interval $(0, 2\pi)$ radians.

Item No.	Type	Byte Length	Identification
1	Integer	2	Spare
2	Integer	2	Error indicator
3	Integer	2	No. blocks to process
4	Integer	2	Spare

Explanation

Item 2 — The error indicator is an output quantity which will have one of four values:

1. 0: There were no errors.
2. -1:
 - a. The mean elements could not be computed from the osculating elements.
 - b. The input osculating elements were rejected because the elements did not fall within the following limits:
 - . $4 \leq a \leq 9.0$ CUL
 - . $0 \leq e \leq .85$ ND
 - . $.18 \leq i \leq 1.05$ radians
 - . $0 \leq l \leq 2\pi$ radians
 - . $0 \leq g \leq 2\pi$ radians
 - . $0 \leq h \leq 2\pi$ radians
 - c. The amount of time to update the elements exceeds four days. If this occurs there will be a message printed on line.
3. -2: The trajectory being updated became reentrant.
4. -3: Convergence for a mean anomaly, argument of latitude, or position match update was not obtained. There will be a message printed on line if it fails to converge on position match.

Item 3 — The number of blocks to process is an input quantity which should be an integer greater than zero.

Item 1 and item 3 will not be altered by the AEG.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMAEG
 DATE 10/21/68
 REV
 PAGE 4 of 54
 MISSION EORP

Table 2. Input Block Format

All angles are in the interval $(0, 2\pi)$ radians.

Item No.	Type	Byte Length	Units	Identification
1	Integer	2		Spare
2	Integer	2		Initialize/update indicator
3	Integer	2		Update option indicator
4	Integer	2		Harmonics indicator
5	Real	8		Drag indicator/K factor
6	Real	8	ft. ²	Vehicle area
7	Real	8	lbs.	Vehicle weight
8	Real	8	rad.	l, U, or Uoc
9	Real	8	hrs.	Toc
10	Real	8		ΔN
11	Real	8	ER	Osculating a
12	Real	8		Osculating e
13	Real	8	rad.	Osculating i
14	Real	8	rad.	Osculating l
15	Real	8	rad.	Osculating g
16	Real	8	rad.	Osculating h
17	Real	8	ER	Mean a
18	Real	8		Mean e
19	Real	8	rad.	Mean i
20	Real	8	rad.	Mean l
21	Real	8	rad.	Mean g
22	Real	8	rad.	Mean h
23	Real	8	hrs.	Time associated with items
			hrs.	11-16
24	Real	8	rad. /hr.	l
25	Real	8	rad. /hr.	g
26	Real	8	rad. /hr.	h
27	Real	8		Time to which to update
			hrs.	elements
28	Real	8	rad.	f
29	Real	8	rad.	U
30	Real	8	ER	R

IBM RTCC APOLLO PROGRAMMING SYSTEMS

2. PMMAEG
DATE 10/21/68
REV
PAGE 5 of 54
MISSION EORP

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

Explanation - Table 2

Item 2 — The Initialize/initialize-update indicator can have three values:

1. 0: Only osculating elements are provided. Initialize the AEG and update according to item 3.
2. 1: Osculating and mean elements, i , g , h , f , U , and R are provided. Update the elements according to item 3 using the low e formulation in the AEG.
3. 2: Osculating and mean elements, i , g , h , f , U , and R are provided. Update the elements according to Item 3 using the high e formulation in the AEG.

Item 3 — The update option indicator can be one of seven values:

1. 0: Update to the time given in item 27.
2. 1: Update to the mean anomaly given in item 8 in the orbit ΔN orbits from the input elements. ΔN is given in item 10.
3. 2: Update to the argument of latitude given in item 8 in the orbit ΔN orbits from the input elements. ΔN is given in item 10.
4. 3: Update to a maneuver counter line which is ΔN (item 10) counter lines from the reference counter line whose argument of latitude is given in item 8, and whose crossing time is given in item 9.
5. 4: Update the input elements to the time associated with output elements of the previous block which was processed. Compute the phase angle and return as output.
6. 5: Update the input elements to position match (phase angle equal to zero) with the previous block by backing through the phase angle 0 to 2π . In other words the vehicle under consideration is assumed to be leading the other vehicle represented by the previous block. Compute the phase angle at time match, and the time lag and height difference at position match, and return them in the output block.
7. 6: Update the input elements to position match with the previous block which was processed by going forward if the phase angle at time match is greater than 180° , or by going backward if the phase angle at time match is less than 180° . Time lag, height difference, and phase angle at time match are returned in the output block.

When using update option 4, 5, or 6 the phase angle will be returned between -180° and 180° . When updating to position match, option 6 should be used for Apollo rendezvous missions.

Item 4 — The harmonics indicator can have one of three values:

1. 0: Second and fourth harmonics of the potential function will be considered.
2. 1: Second, third, and fourth harmonics will be considered.
3. -1: Second and third harmonics will be considered.

Only (2) should be used in the real time system.

Item 5 — The drag indicator can be zero or nonzero. If it is zero, no drag will be used. If it is nonzero, the magnitude should be the K factor, which is nominally equal to 1. If it is nonzero, a reentry test will be made in the drag program, and the error indicator set to -2 if a radius magnitude in the update interval becomes less than one Earth radius plus fifty nautical miles.

Items 6 and 7 — The vehicle's area and weight should be furnished if item 5 is nonzero.

Item 8 — See item 3, (2), (3), and (4).

Item 9 — See item 3 (4).

Item 10 — See item 3 (2), (3), and (4).

Items 11-16 — Osculating Keplerian elements to be updated.

Items 17-22 — Mean Keplerian elements, which should be furnished if item 2 is nonzero.

Item 23 — Time associated with items 11-16.

Item 24 — Mean anomaly secular rate, which should be furnished if item 2 is nonzero.

Item 25 — Argument of perigee secular rate, which should be furnished if item 2 is nonzero.

Item 26 — Argument of ascending node secular rate, which should be furnished if item 2 is nonzero.

Item 27 — If item 3 is zero, this cell should contain the time to which to update the input elements. If item 3 is nonzero, this cell should contain the time associated with the input elements, which is the same as item 23.

Item 28 — True anomaly associated with input elements, to be furnished if item 2 is nonzero.

Item 29 — Argument of latitude associated with input elements, to be furnished if item 2 is nonzero.

Item 30 — Position magnitude associated with input elements, to be furnished if item 2 is nonzero.

Table 3. Output Block Format

Item No.	Type	Byte Length	Units	Identification
1	Integer	2		Spare
2	Integer	2		Initialize/update indicator
3	Integer	2		Update option indicator
4	Integer	2		Harmonics indicator
5	Real	8		Drag indicator/K factor
6	Real	8	ft. ²	Vehicle area
7	Real	8	lbs.	Vehicle weight
8	Real	8	ER	Height difference (if item 3 = 5 or 6; otherwise, same as input)
9	Real	8	hrs.	Time lag (if item 3 = 5 or 6; otherwise, same as input)
10	Real	8	rad.	Phase lag (if item 3 is 4, 5, or 6; otherwise, same as input)
11	Real	8	ER	Osculating a
12	Real	8		Osculating e
13	Real	8	rad.	Osculating i
14	Real	8	rad.	Osculating l
15	Real	8	rad.	Osculating g
16	Real	8	rad.	Osculating h
17	Real	8	ER	Mean a
18	Real	8		Mean e
19	Real	8	rad.	Mean i
20	Real	8	rad.	Mean l
21	Real	8	rad.	Mean g
22	Real	8	rad.	Mean h
23	Real	8	hrs.	Time associated with updated elements
24	Real	8	rad. /hr.	$\dot{l}$
25	Real	8	rad. /hr.	$\dot{g}$
26	Real	8	rad. /hr.	$\dot{h}$
27	Real	8	hrs.	Time associated with updated elements
28	Real	8	rad.	f
29	Real	8	rad.	U
30	Real	8	ER	R

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMAEG
DATE 10/21/68
REV
PAGE 7 of 54
MISSION EORPExplanation - Table 3

Item 2 — This indicator is set to 1 if the low e initialization path was taken, or if it was 1 when input. It is set to 2 if the high e initialization path was taken, or if it was 2 when input.

Items 3-7 — Same as input.

Items 8-10 — As defined in table.

Items 11-16 — Updated osculating elements.

Items 17-22 — Mean elements associated with input/output items 11-16.

Item 23 — Time associated with updated elements.

Items 24-26 — Secular rates of mean anomaly, argument of perigee, and longitude of ascending node, respectively.

Items 28, 30 — True anomaly, argument of latitude, and radius associated with updated elements.

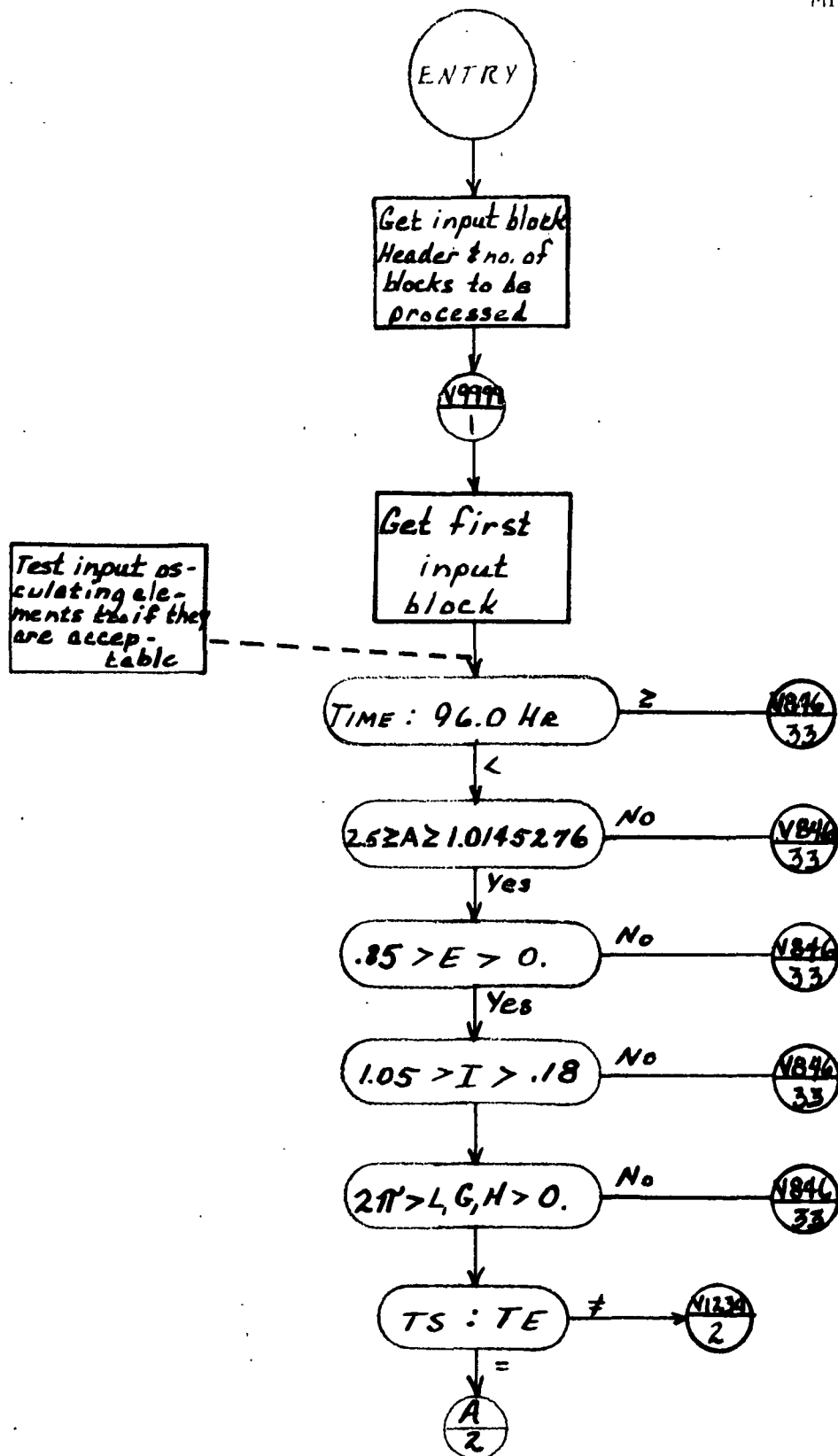


Figure 1. PMMAEG Flowchart

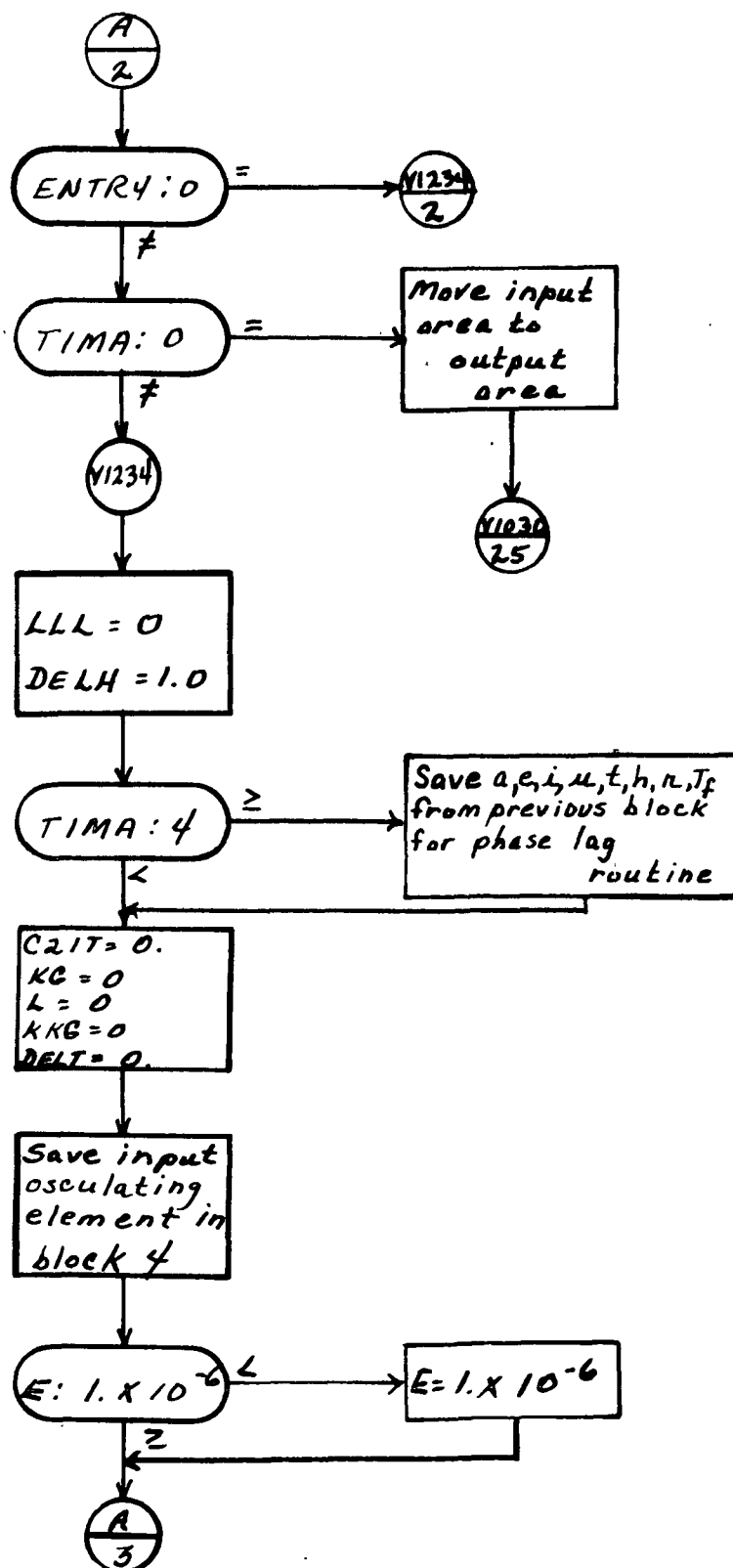


Figure 1. PMMAEG Flowchart

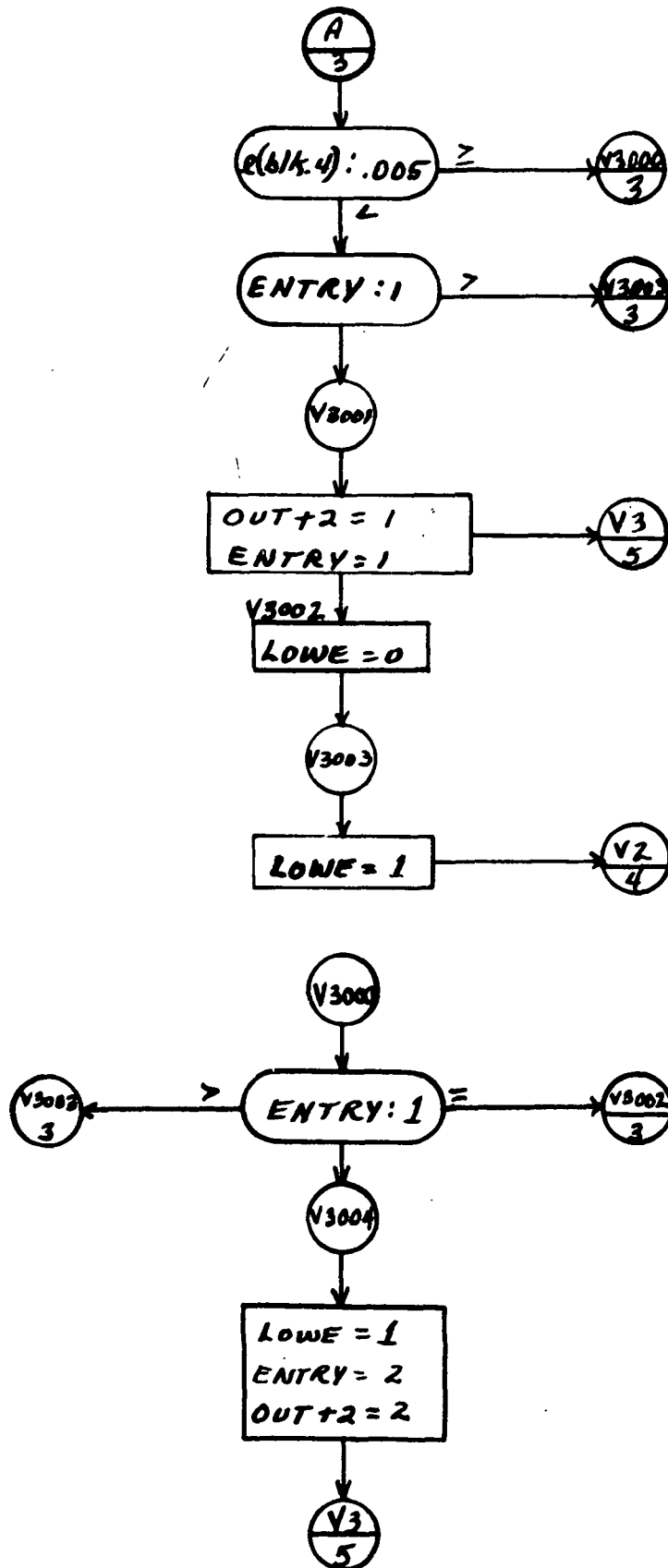


Figure 1. PMMAEG Flowchart

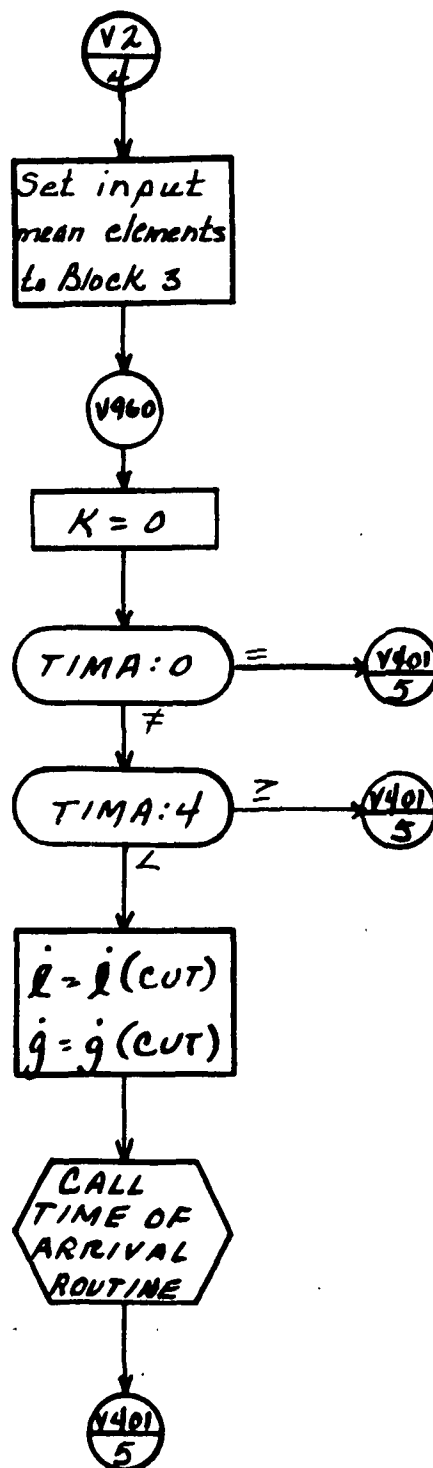


Figure 1. PMMAEG Flowchart

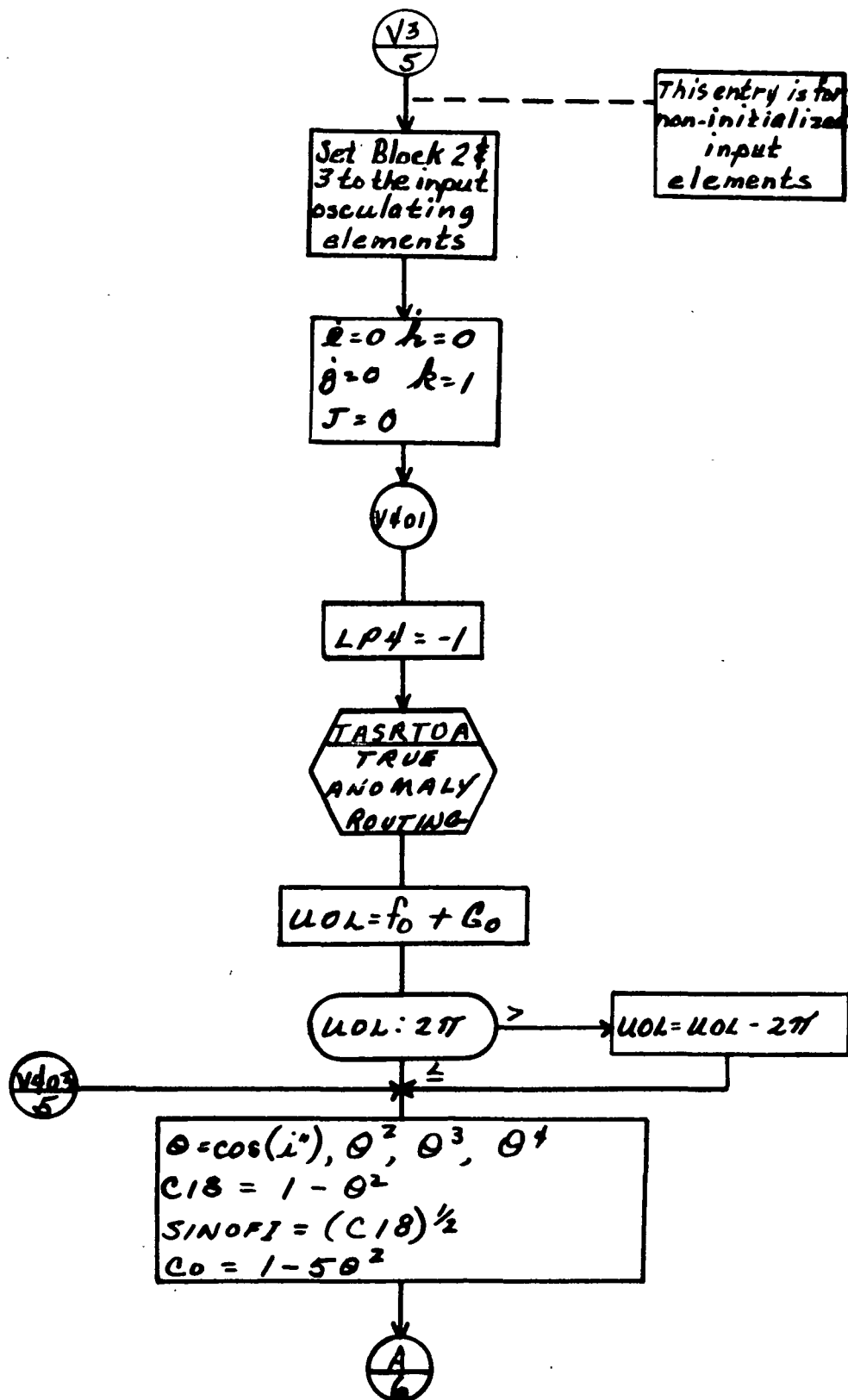


Figure 1. PMMAEG Flowchart

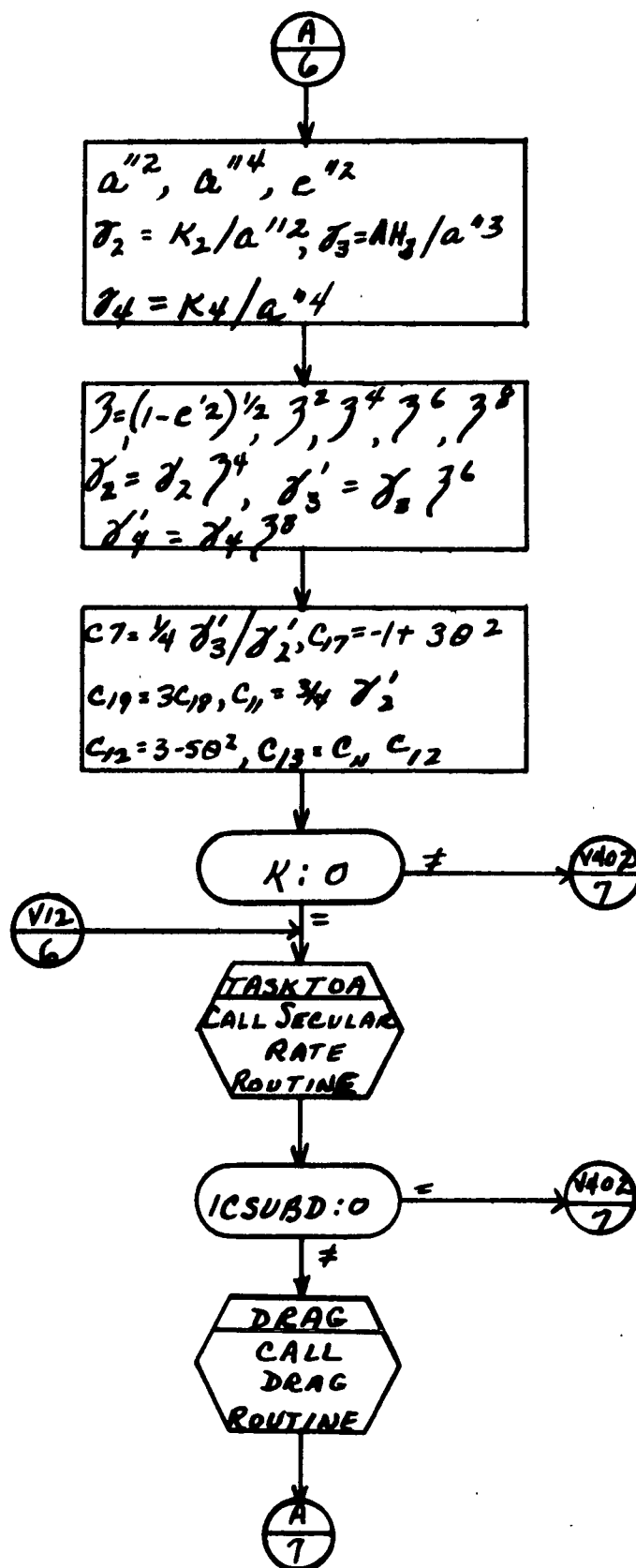


Figure 1. PMMAEG Flowchart

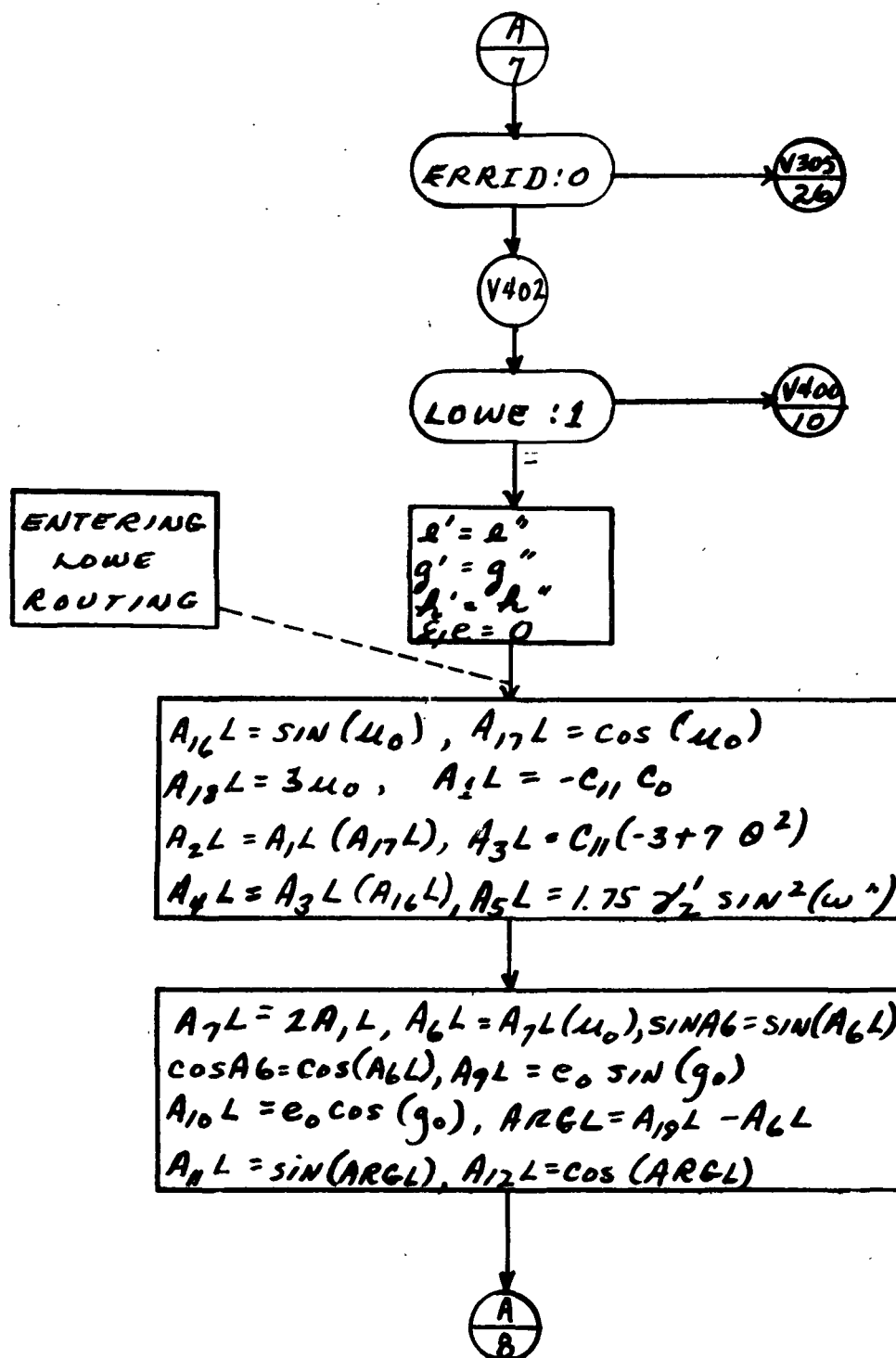


Figure 1. PMMAEG Flowchart

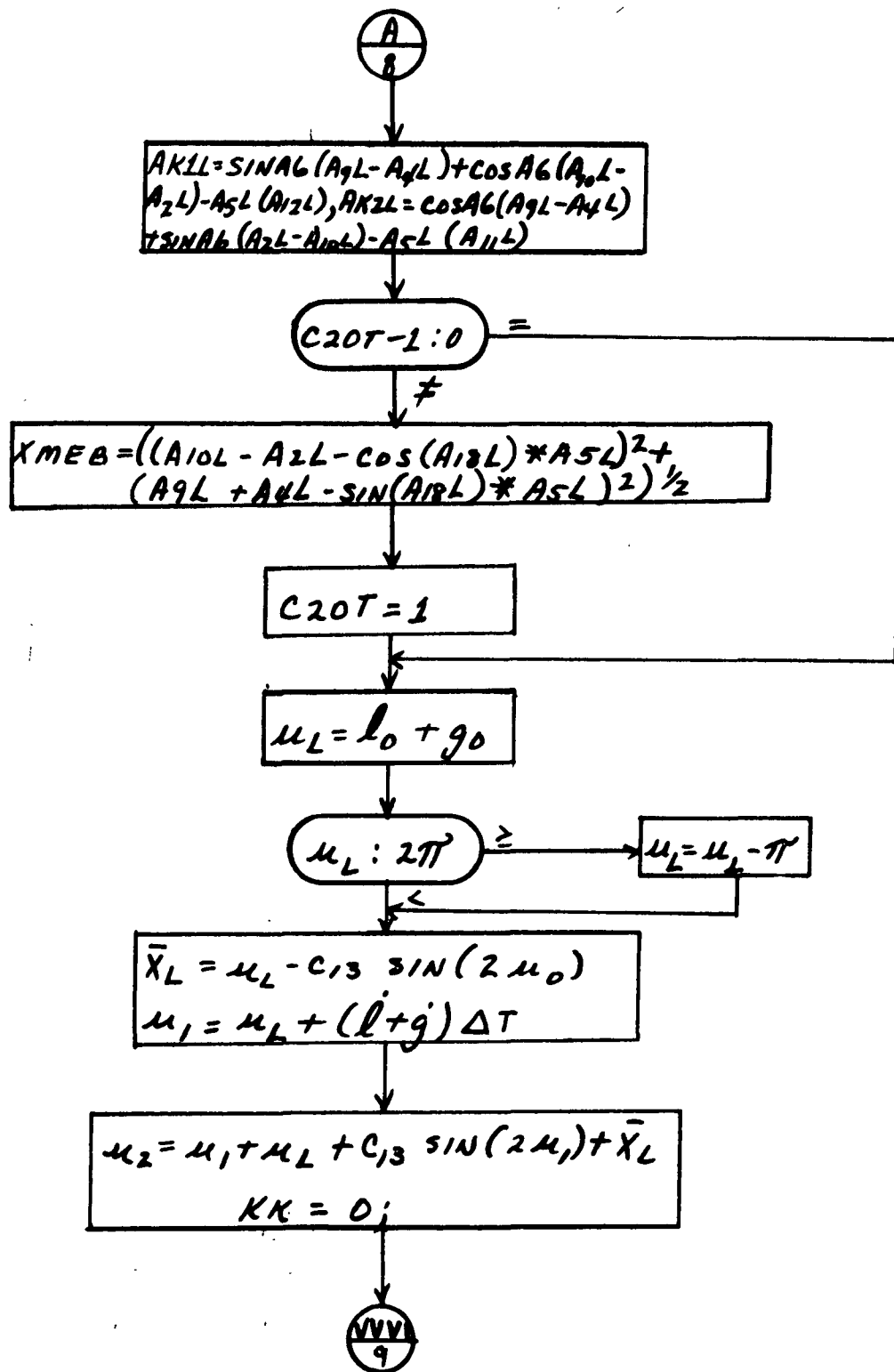


Figure 1. PMMAEG Flowchart

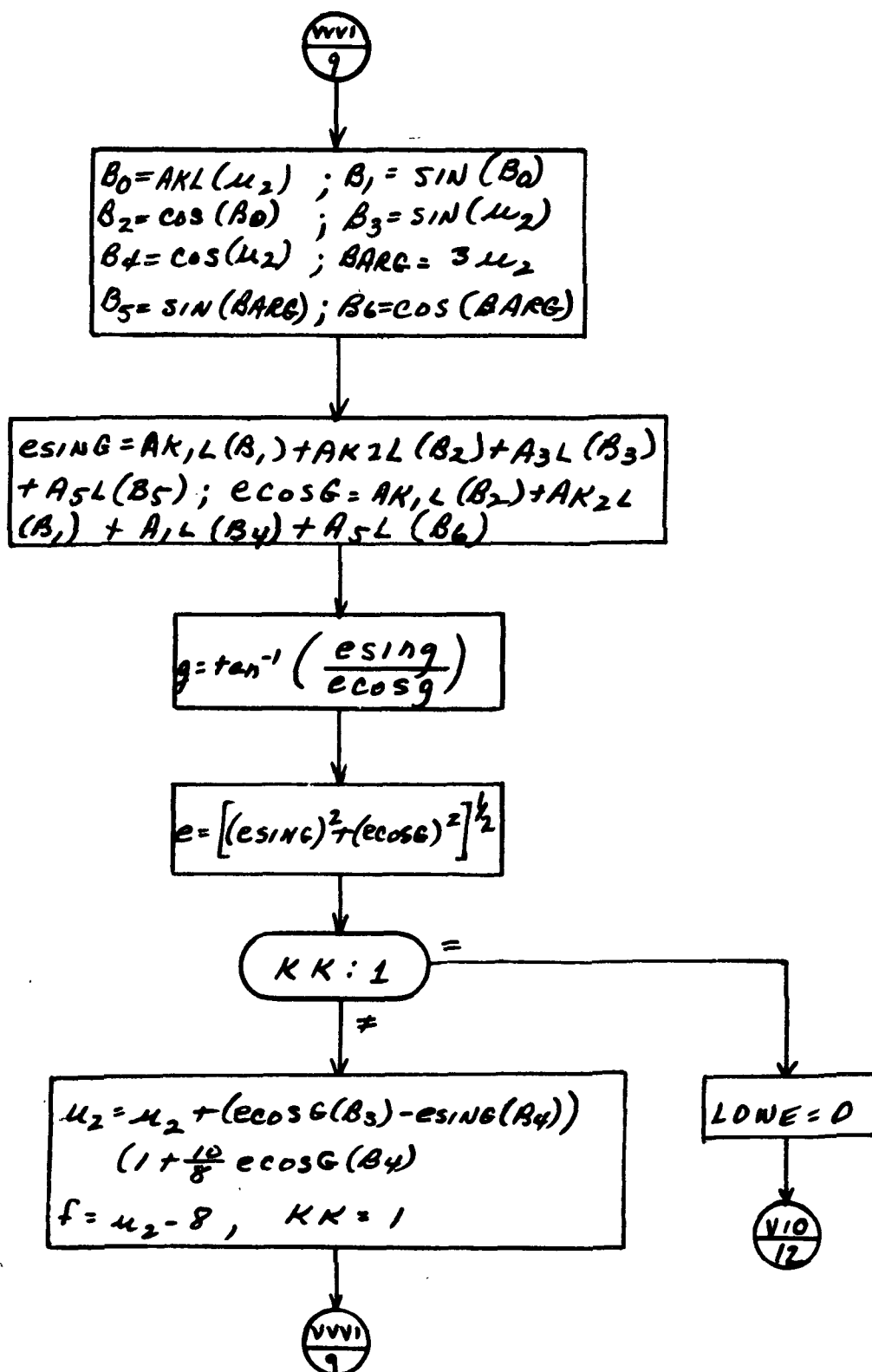


Figure 1. PMMAEG Flowchart

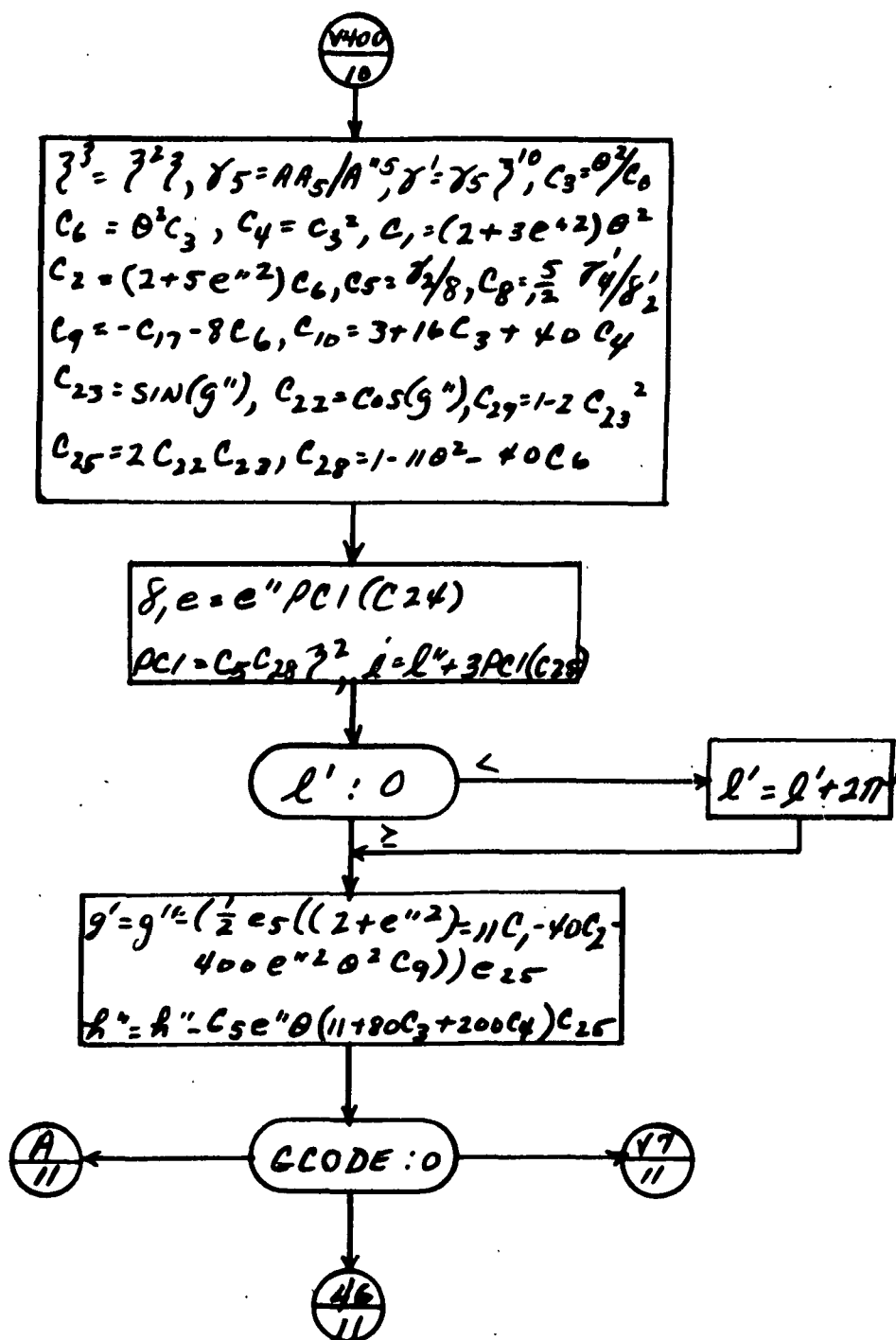


Figure 1. PMMAEG Flowchart

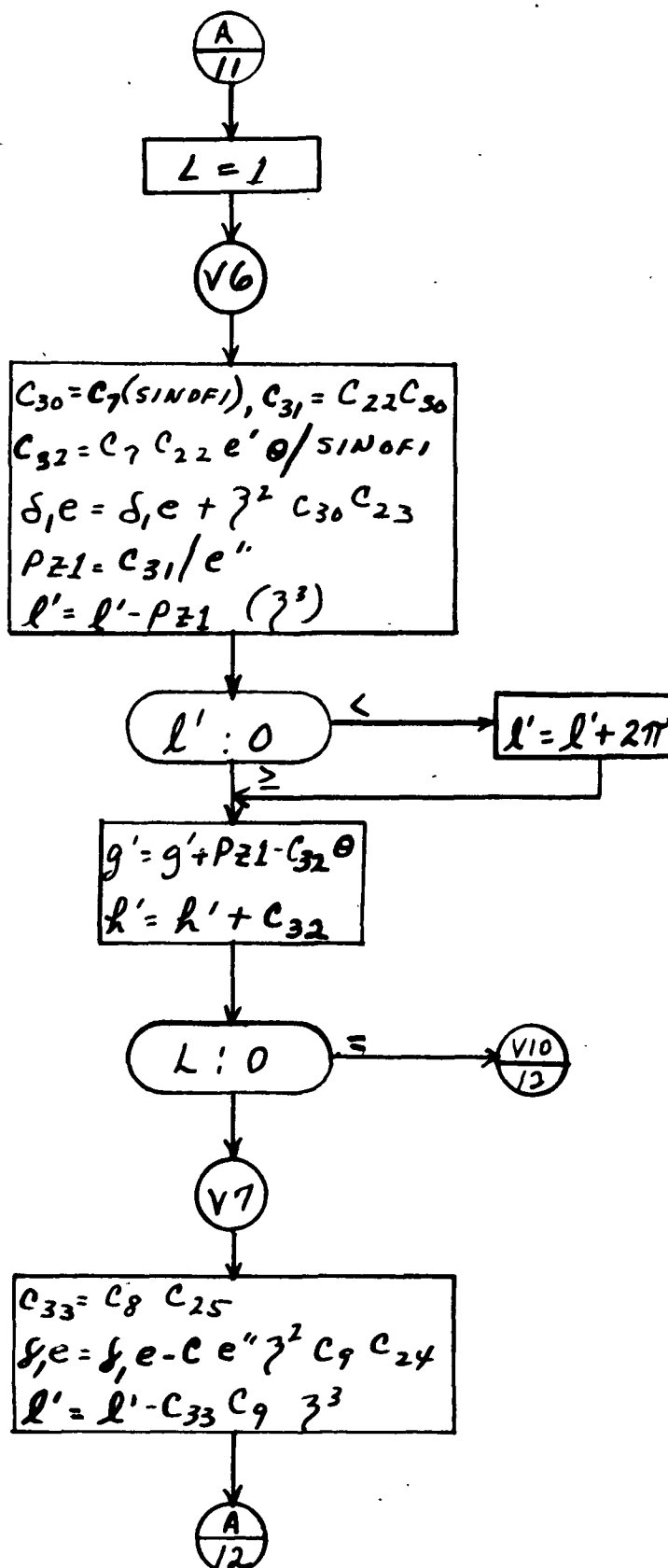


Figure 1. PMMAEG Flowchart

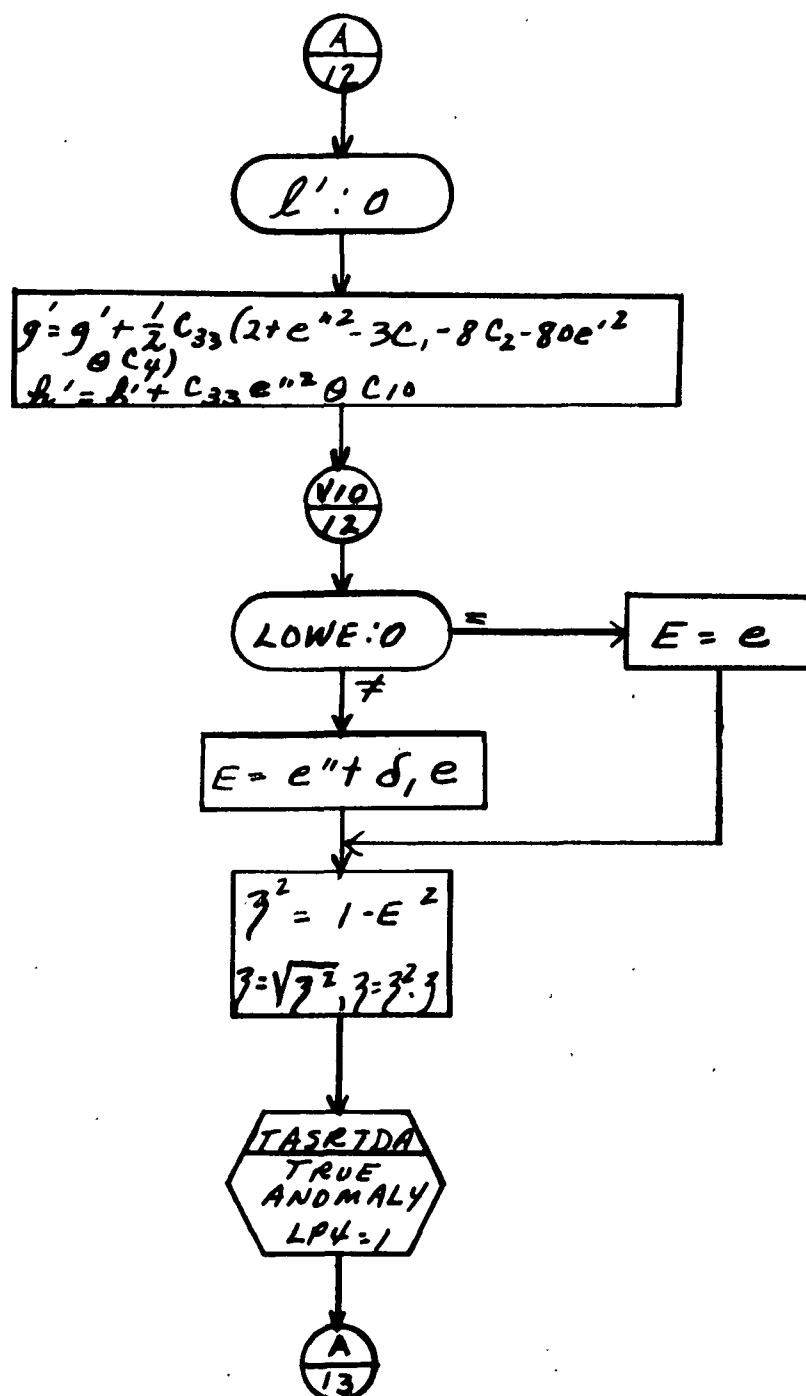


Figure 1. PMMAEG Flowchart

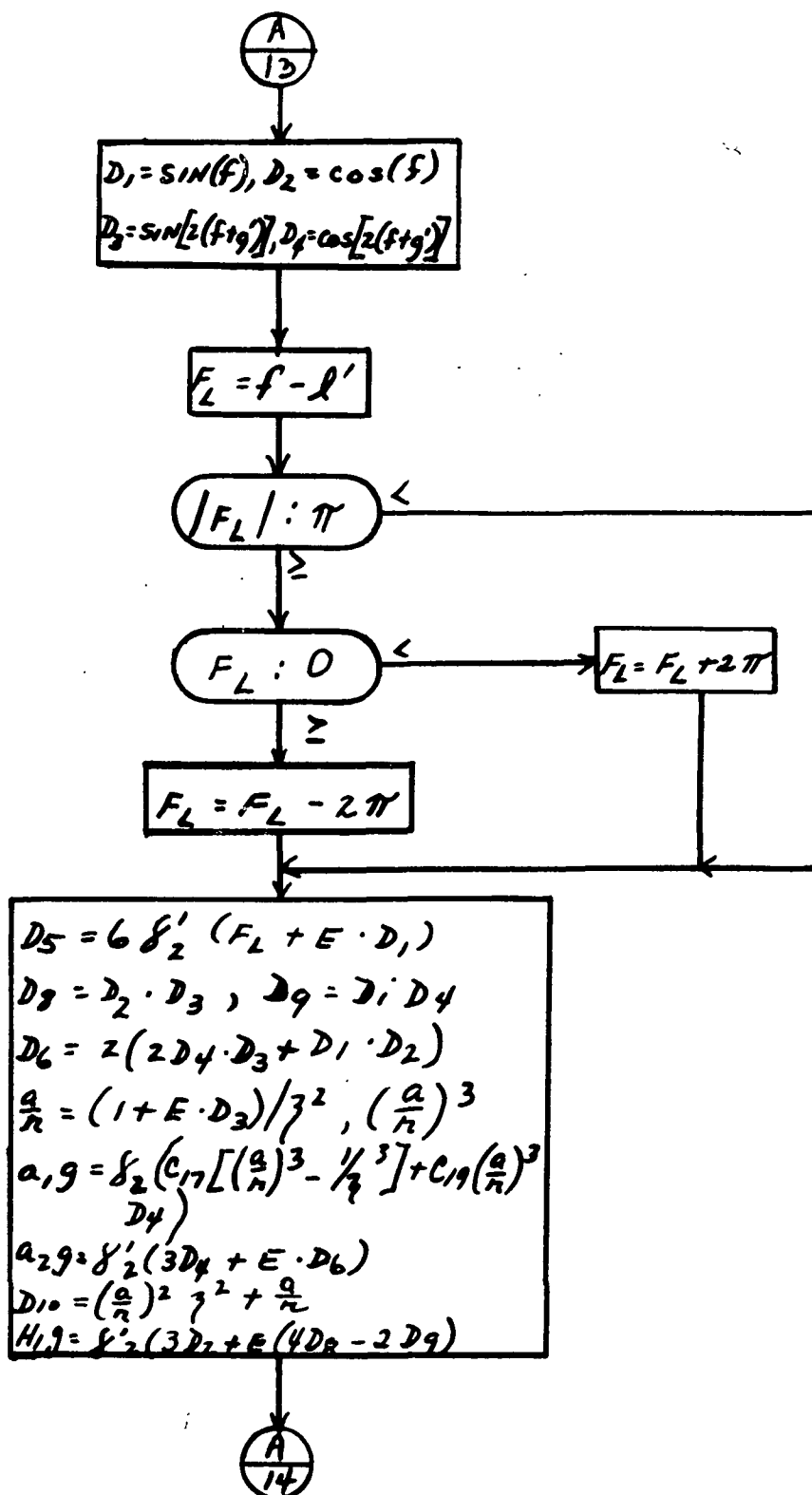


Figure 1. PMMAEG Flowchart

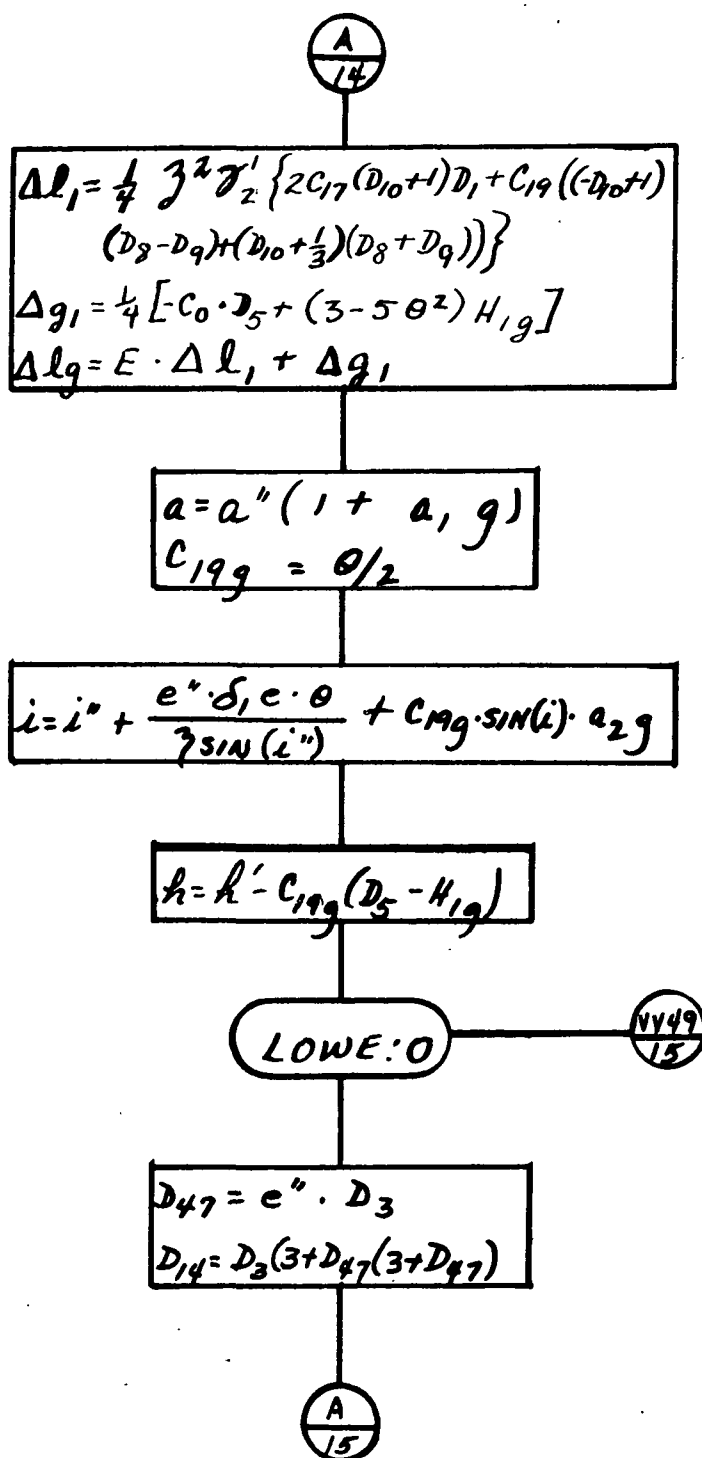


Figure 1. PMMAEG Flowchart

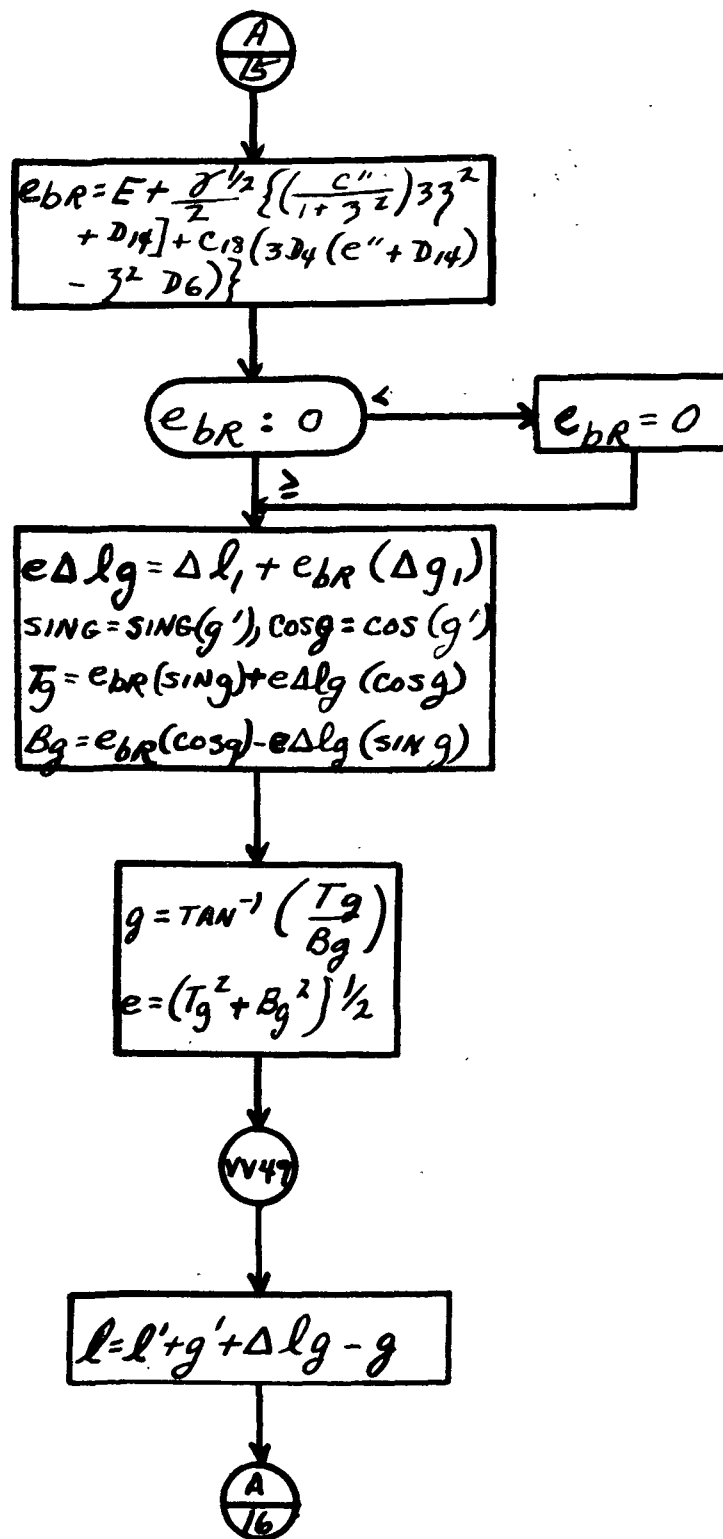


Figure 1. PMMAEG Flowchart

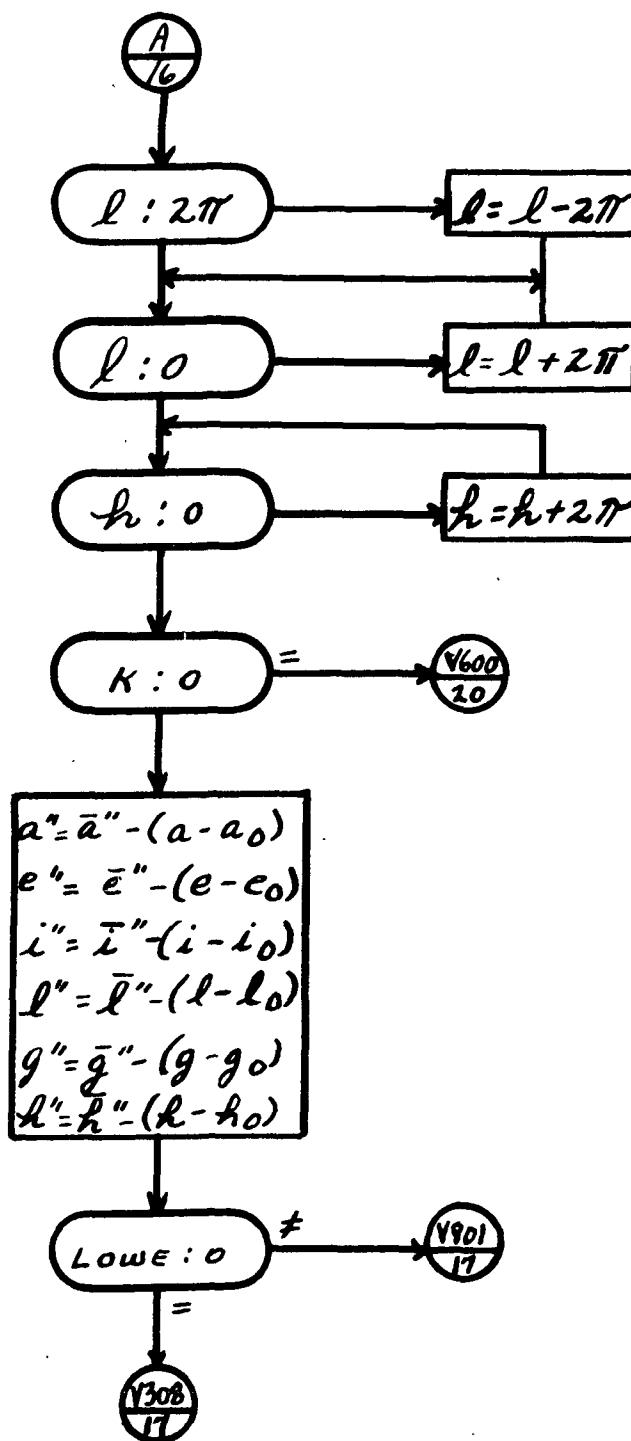


Figure 1. PMMAEG Flowchart

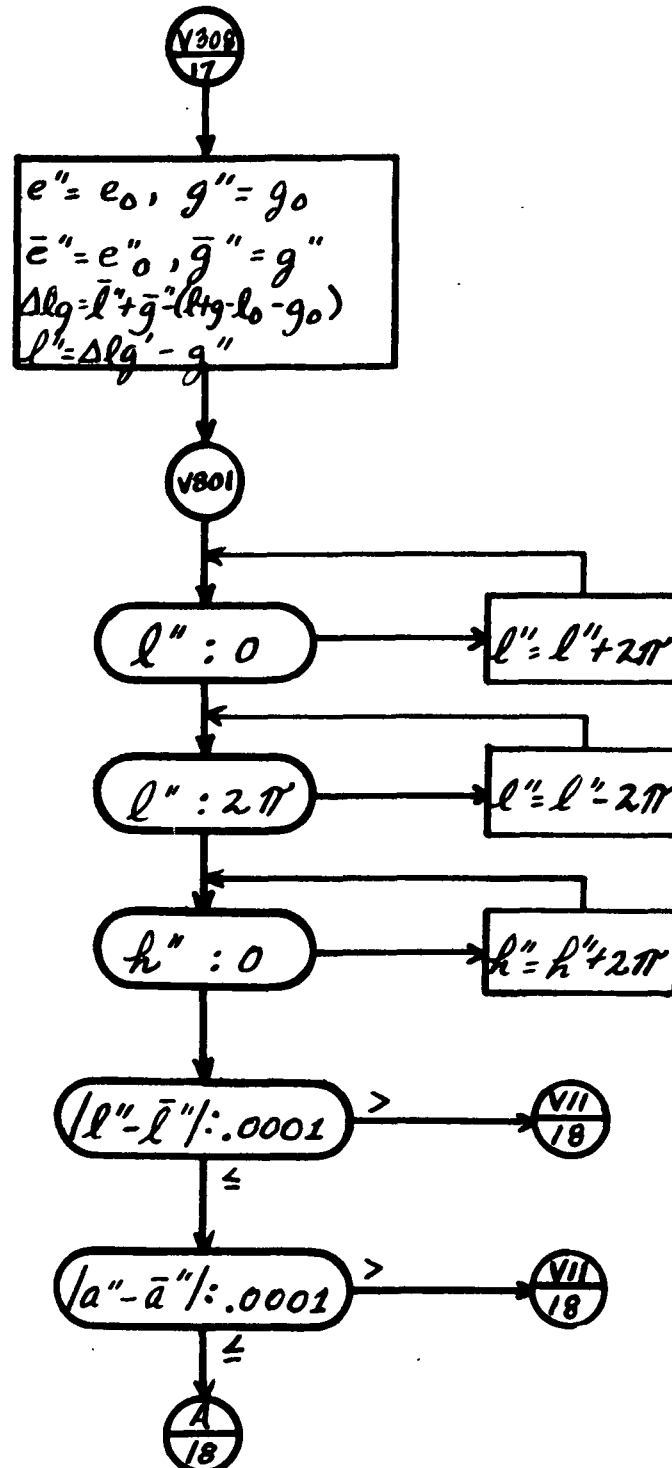


Figure 1. PMMAEG Flowchart

(Sheet 17 of 33)

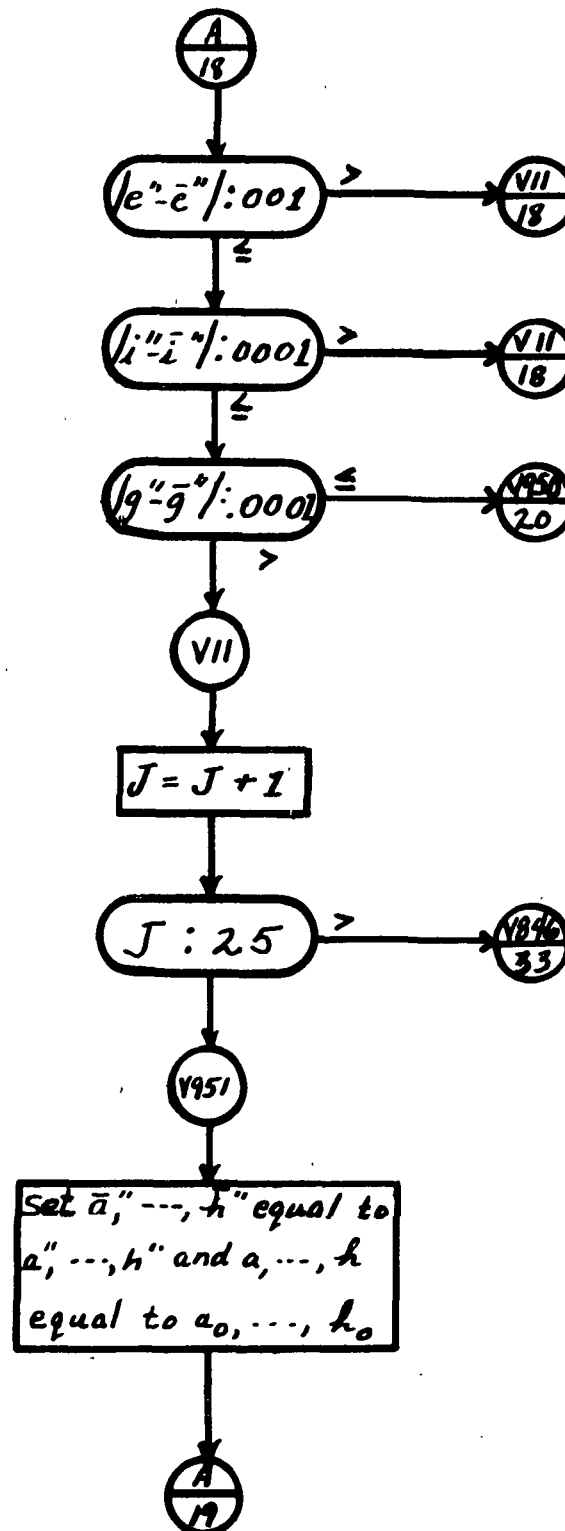


Figure 1. PMMAEG Flowchart

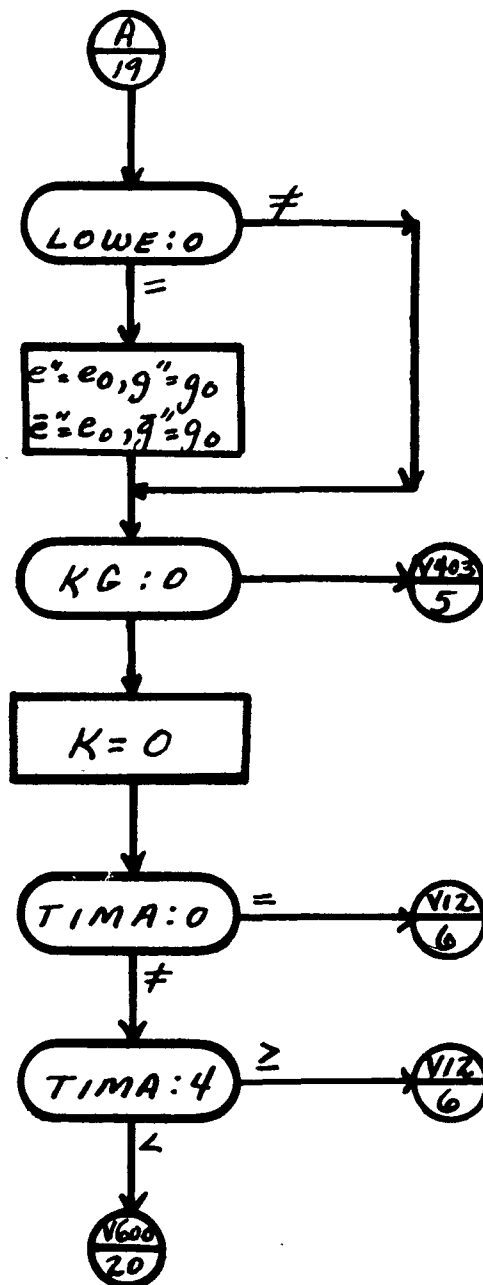


Figure 1. PMMAEG Flowchart

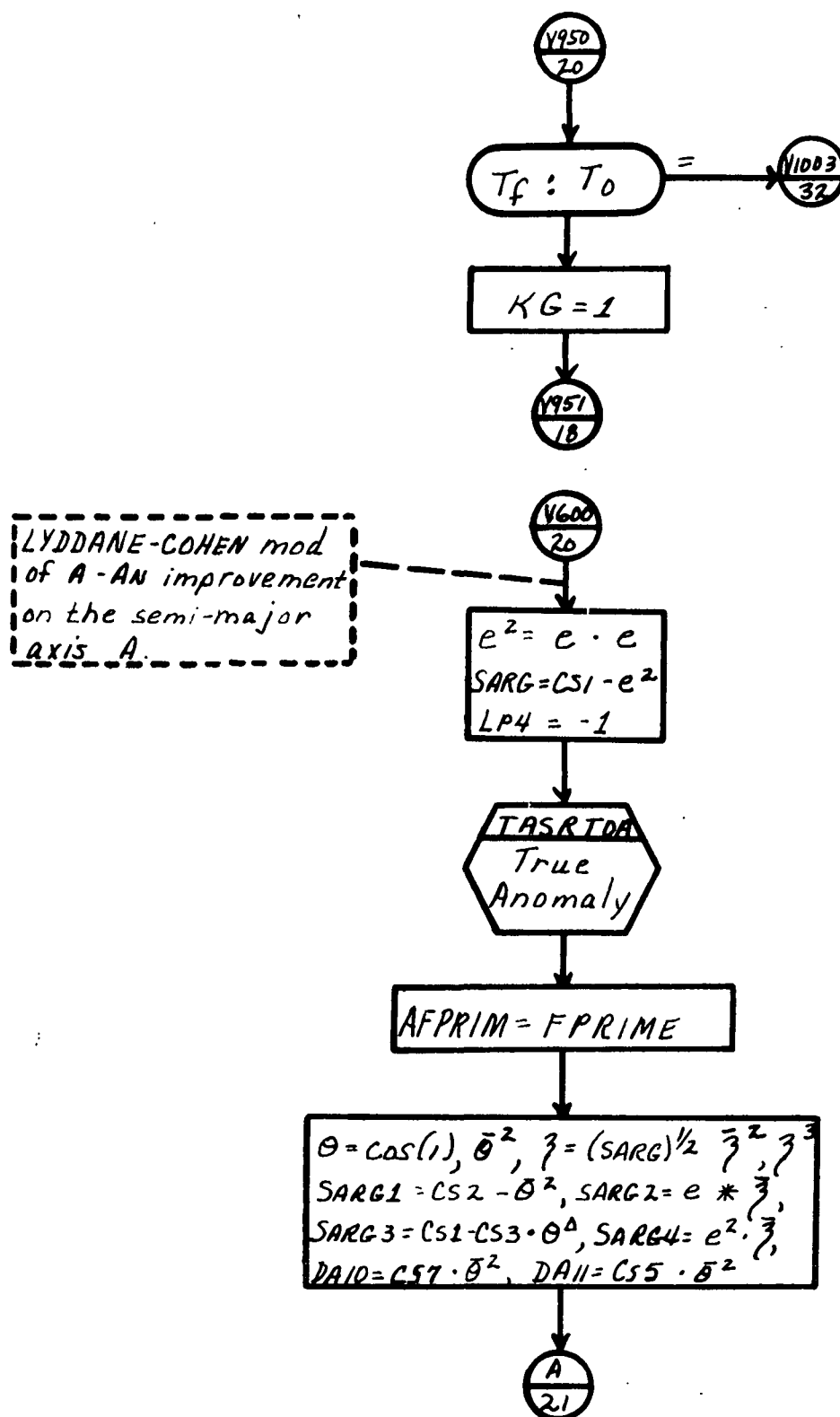


Figure 1. PMMAEG Flowchart

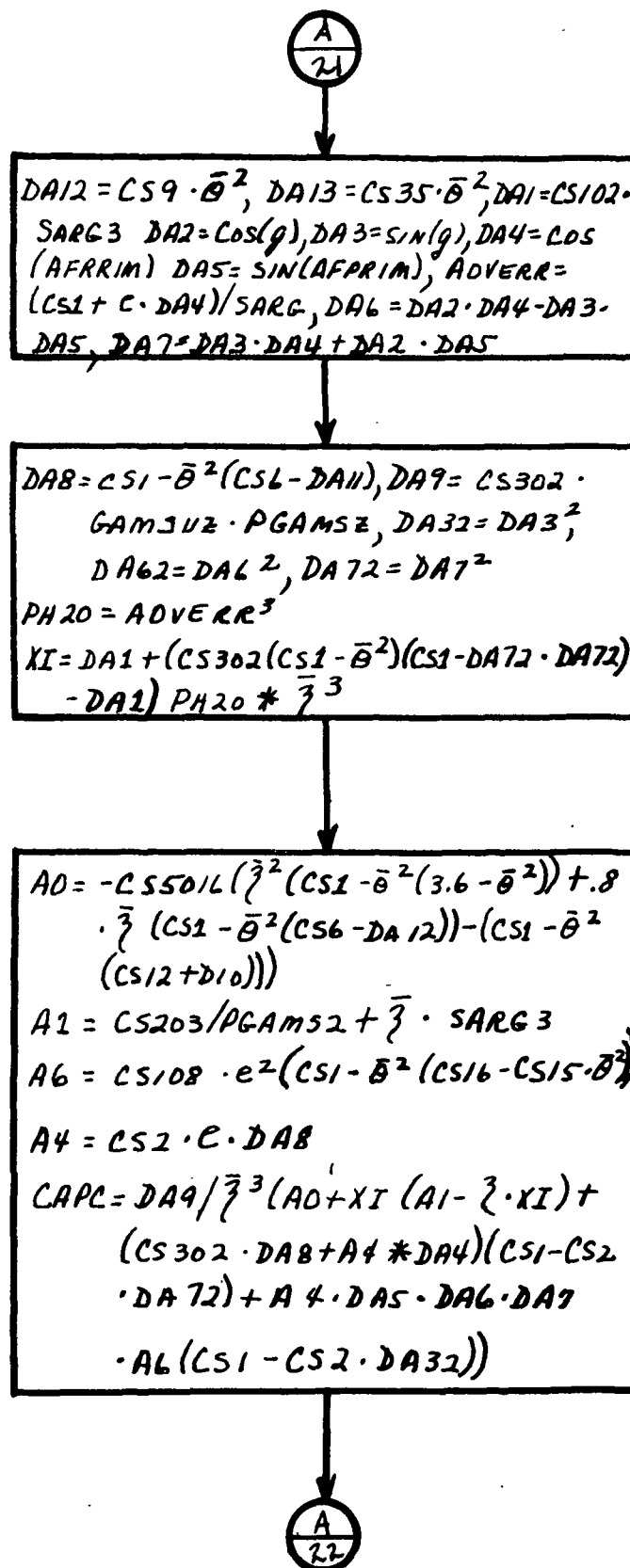


Figure 1. PMMAEG Flowchart

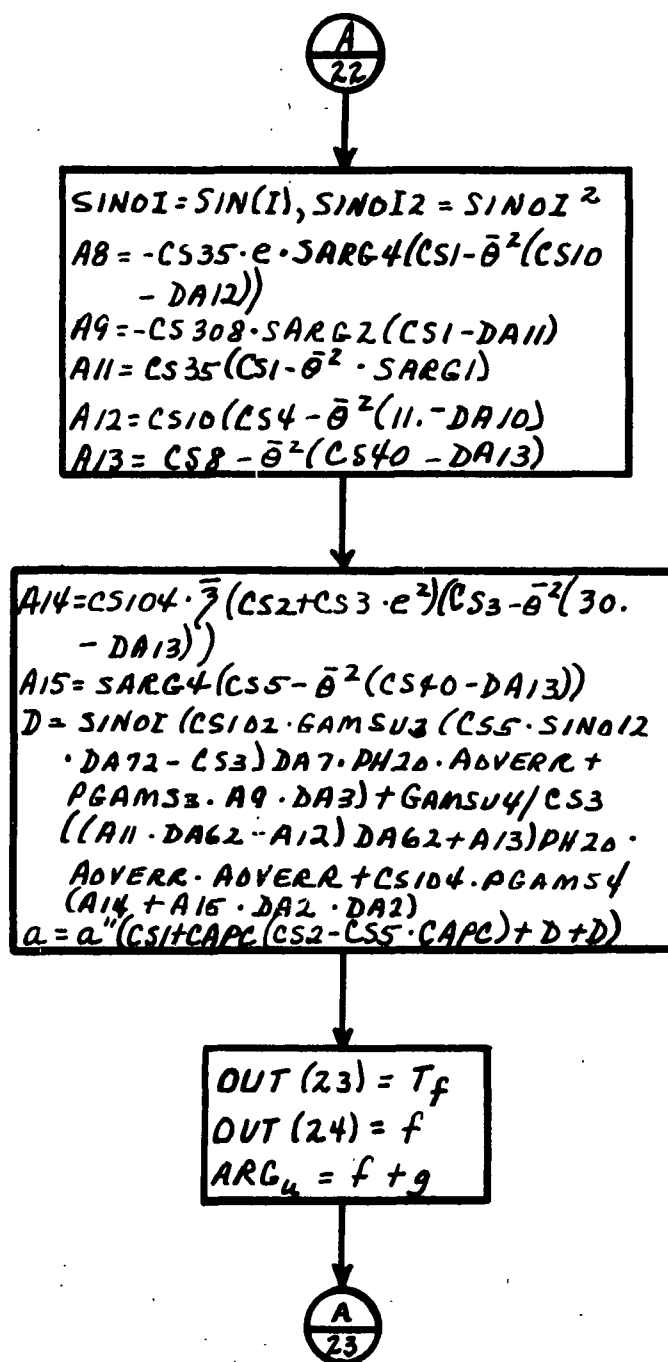


Figure 1. PMMAEG Flowchart

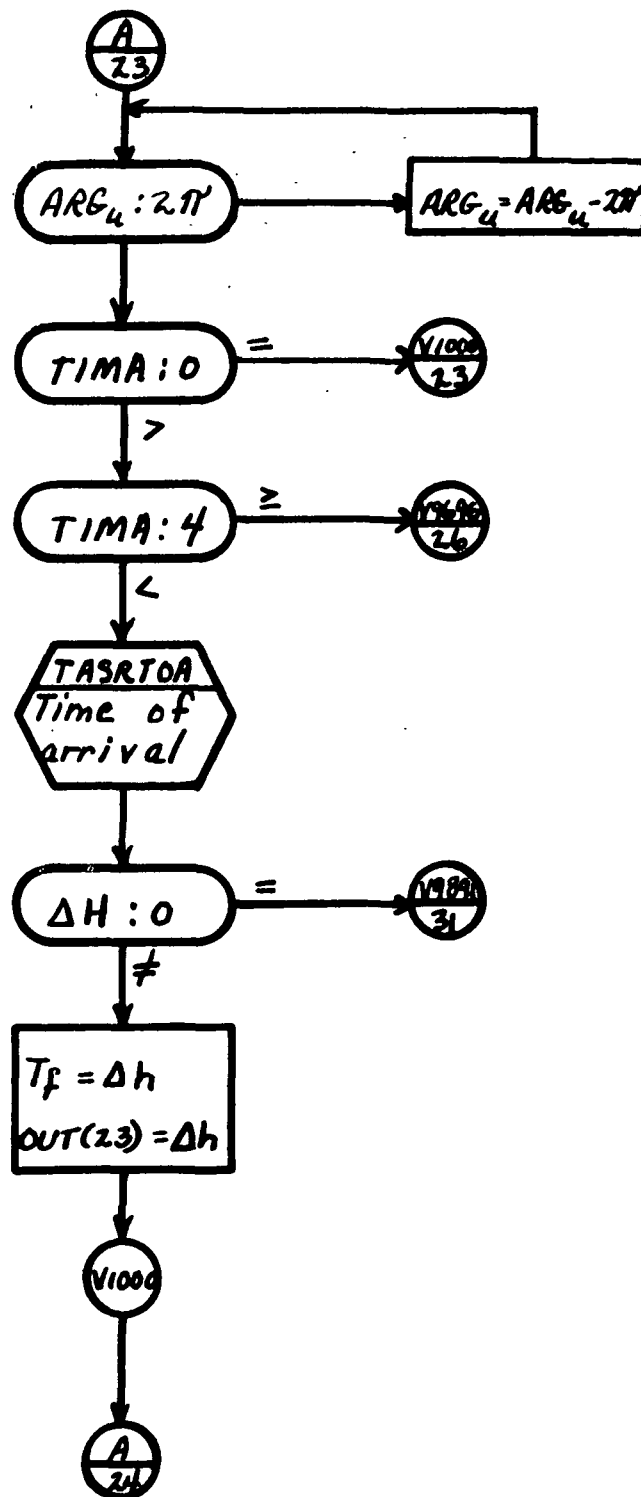


Figure 1. PMMAEG Flowchart

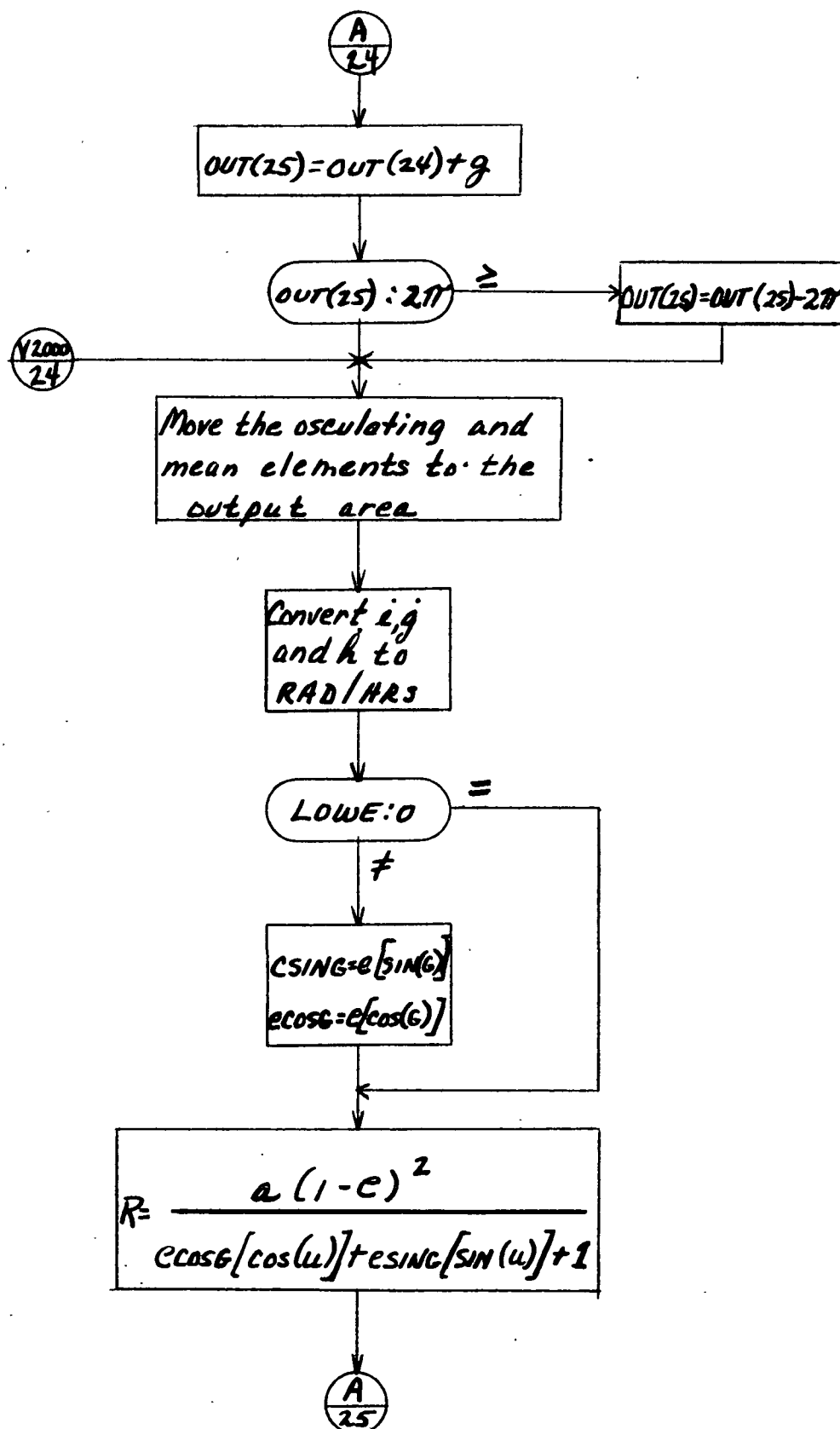


Figure 1. PMMAEG Flowchart

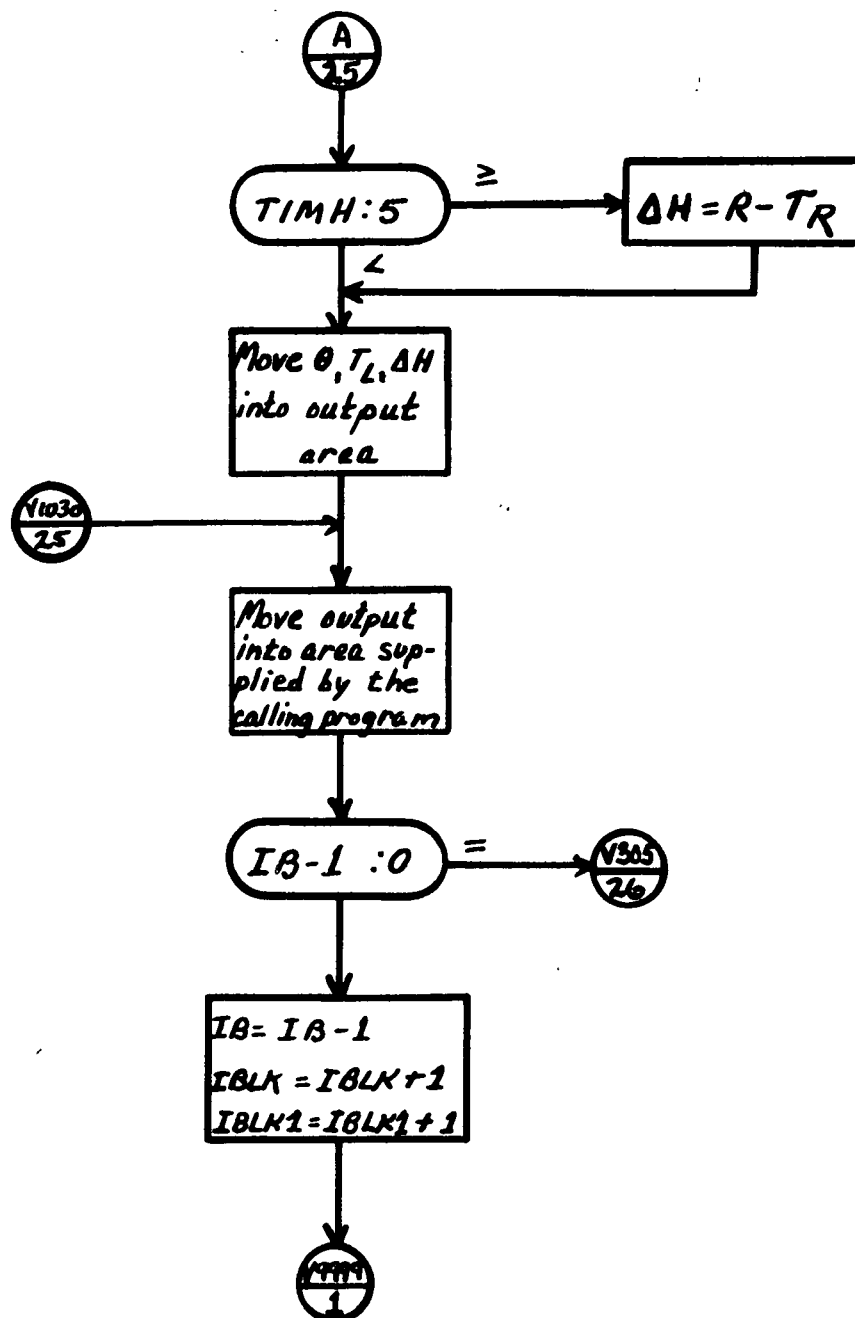


Figure 1. PMMAEG Flowchart

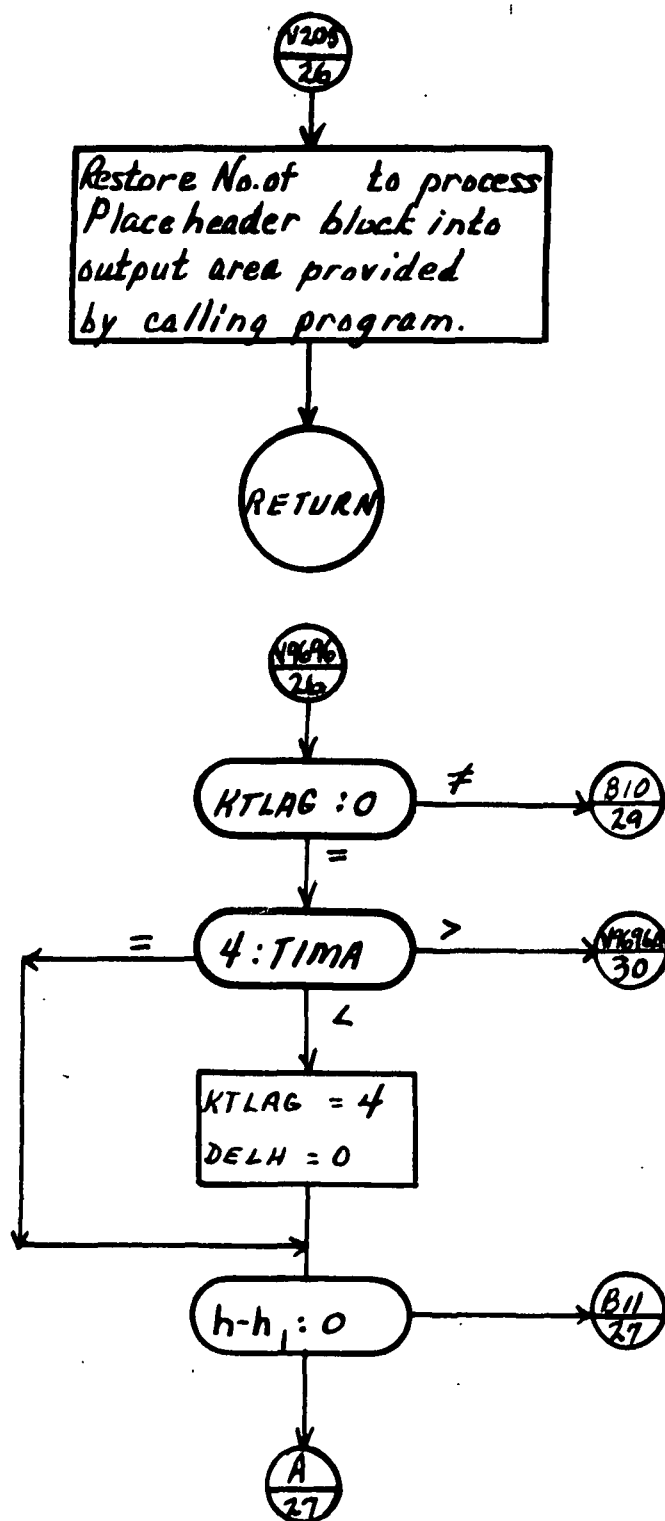


Figure 1. PMMAEG Flowchart

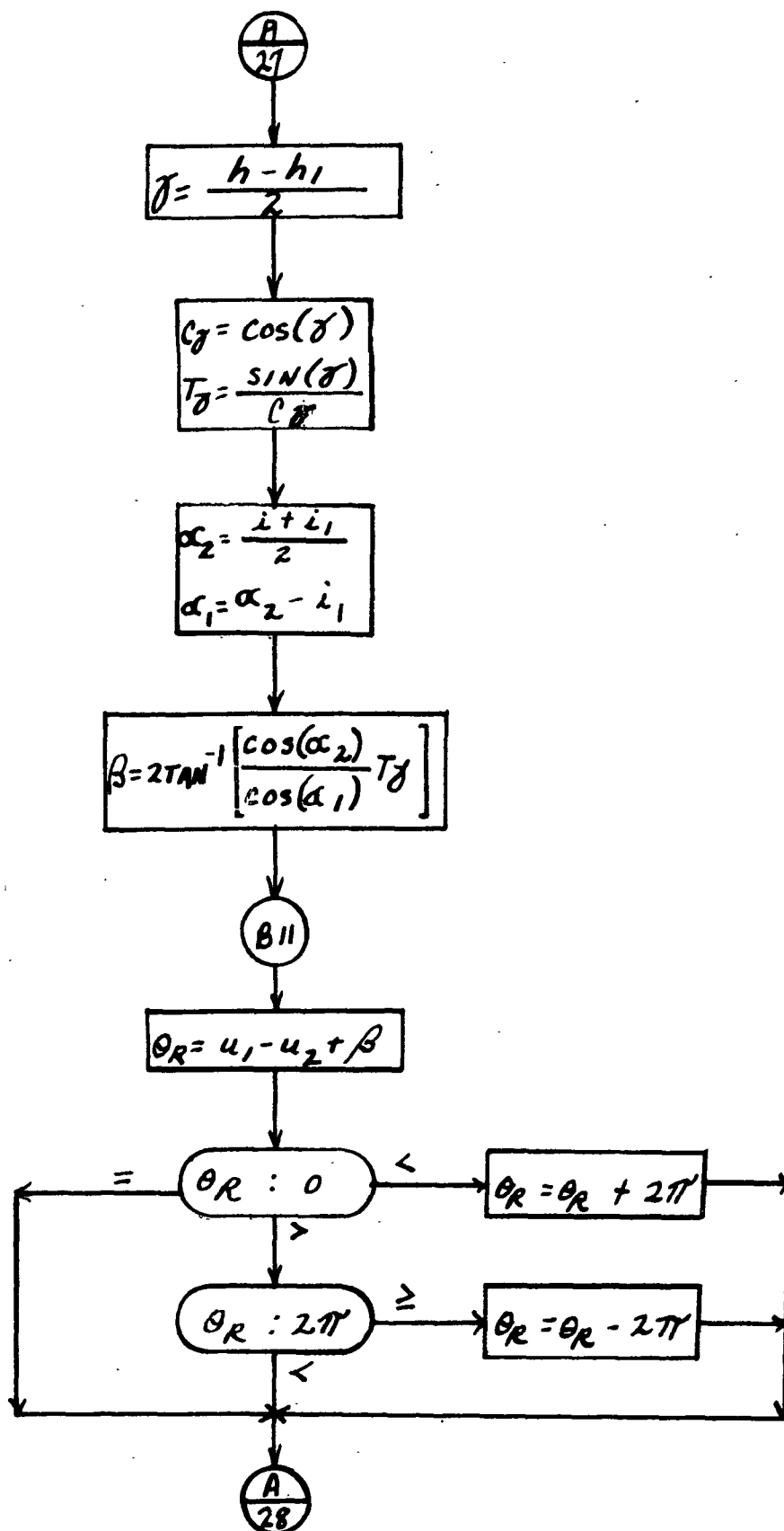


Figure 1. PMMAEG Flowchart

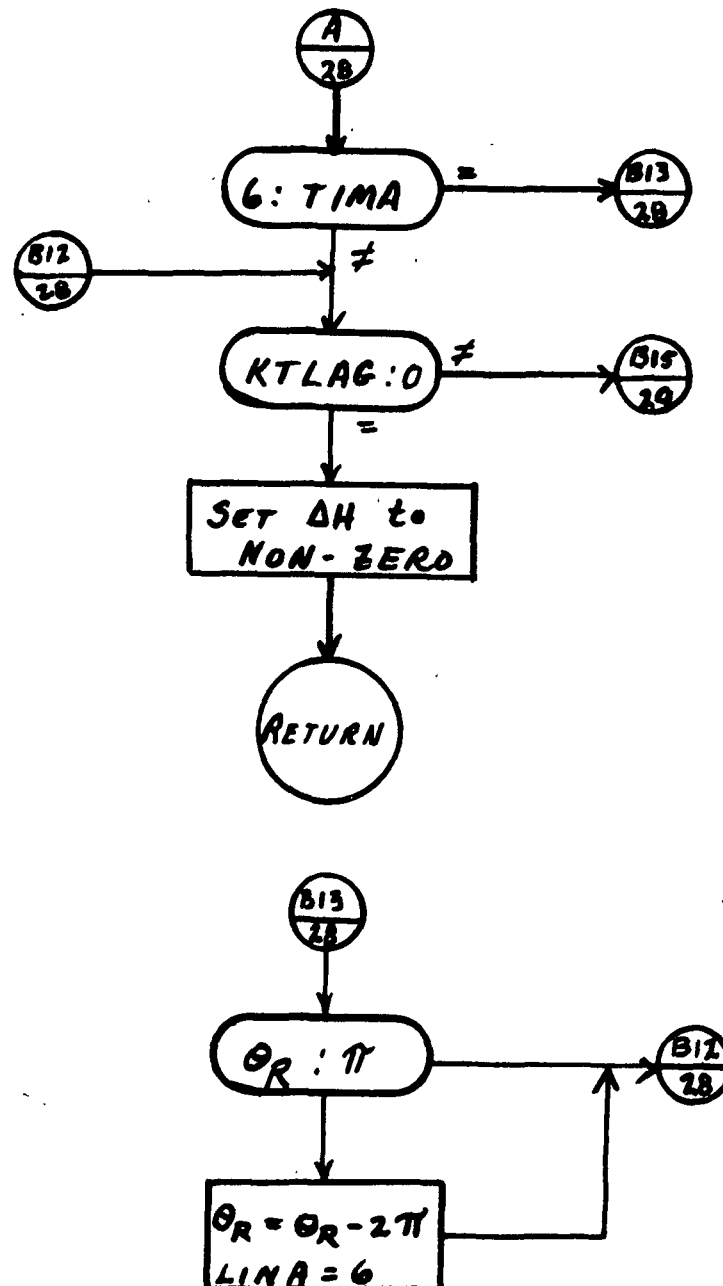


Figure 1. PMMAEG Flowchart

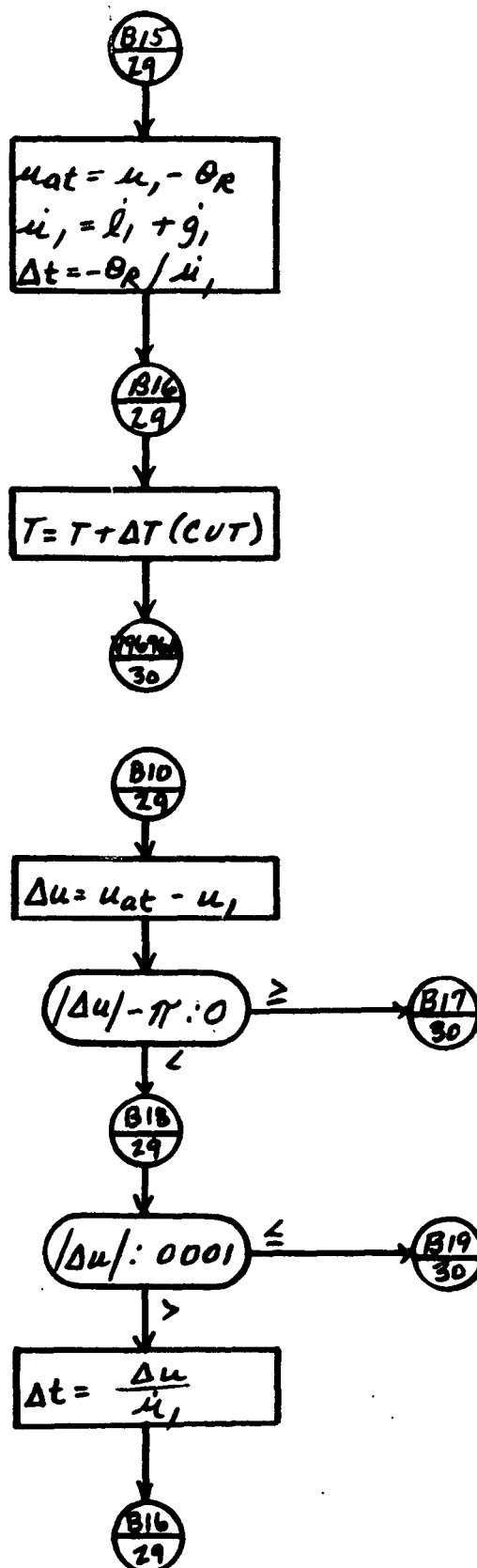


Figure 1. PMMAEG Flowchart

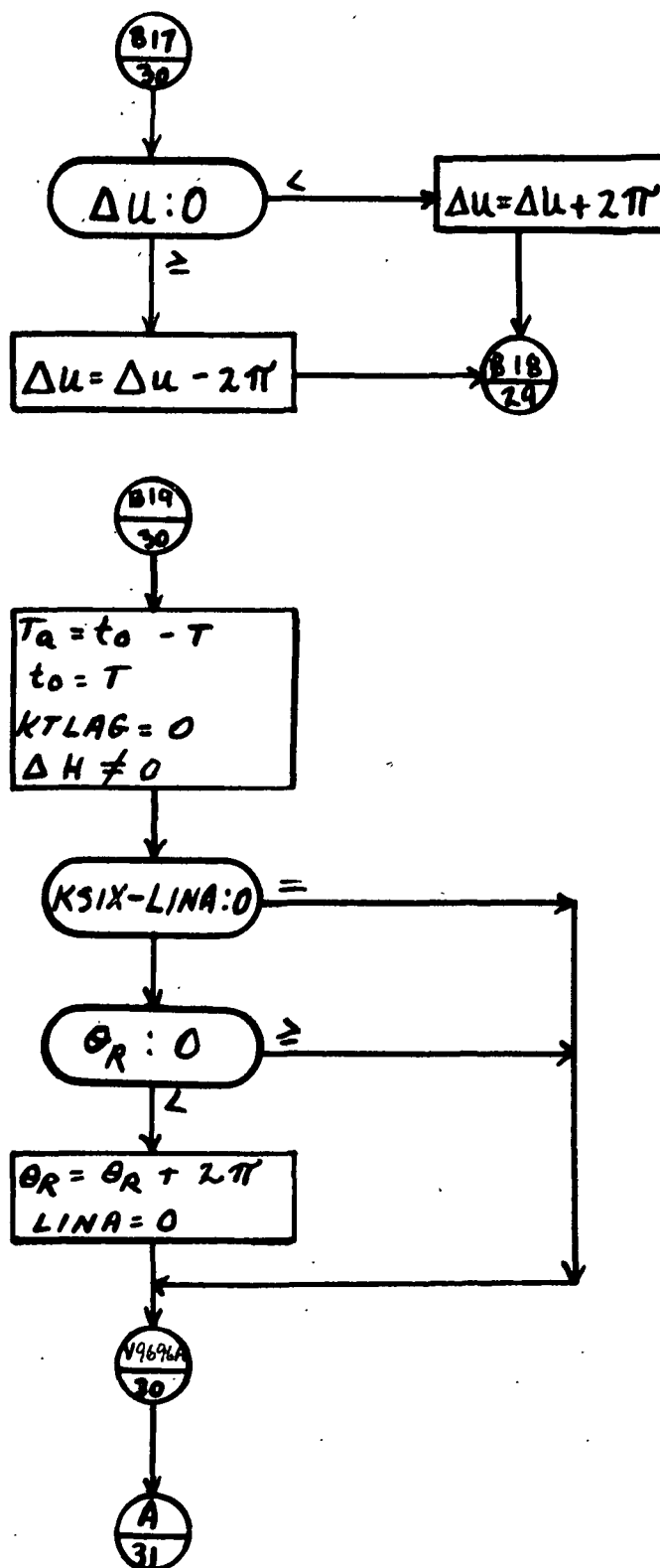


Figure 1. PMMAEG Flowchart

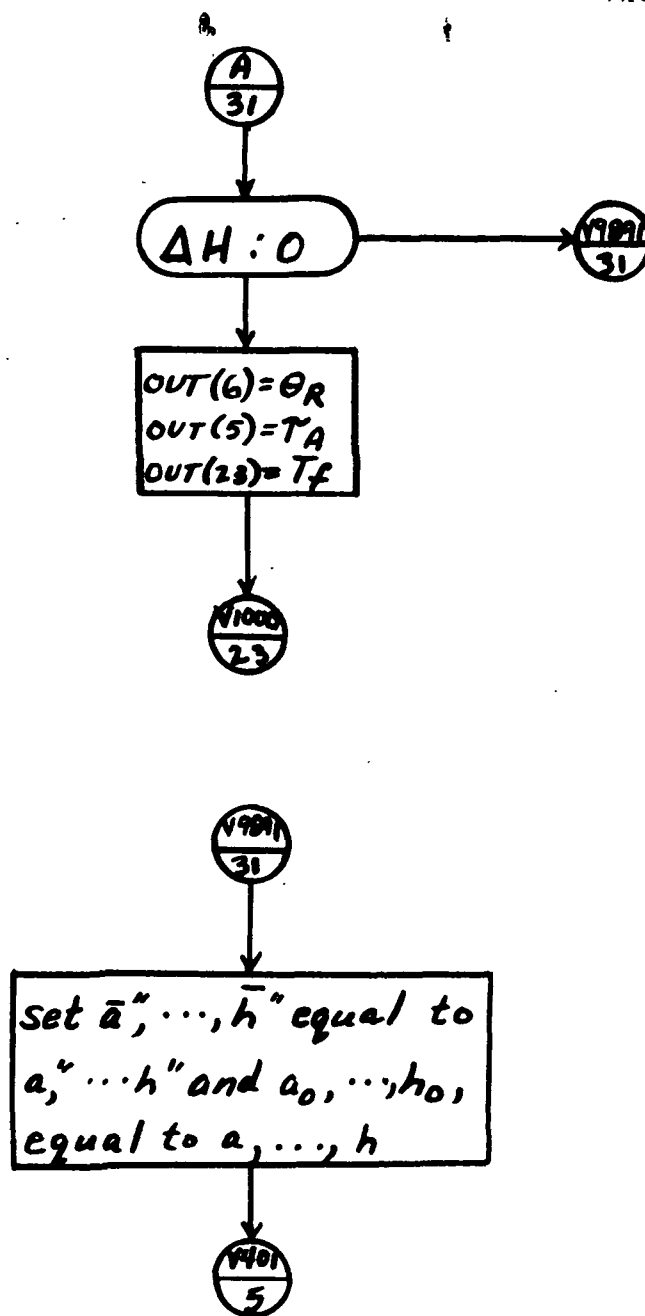


Figure 1. PMMAEG Flowchart

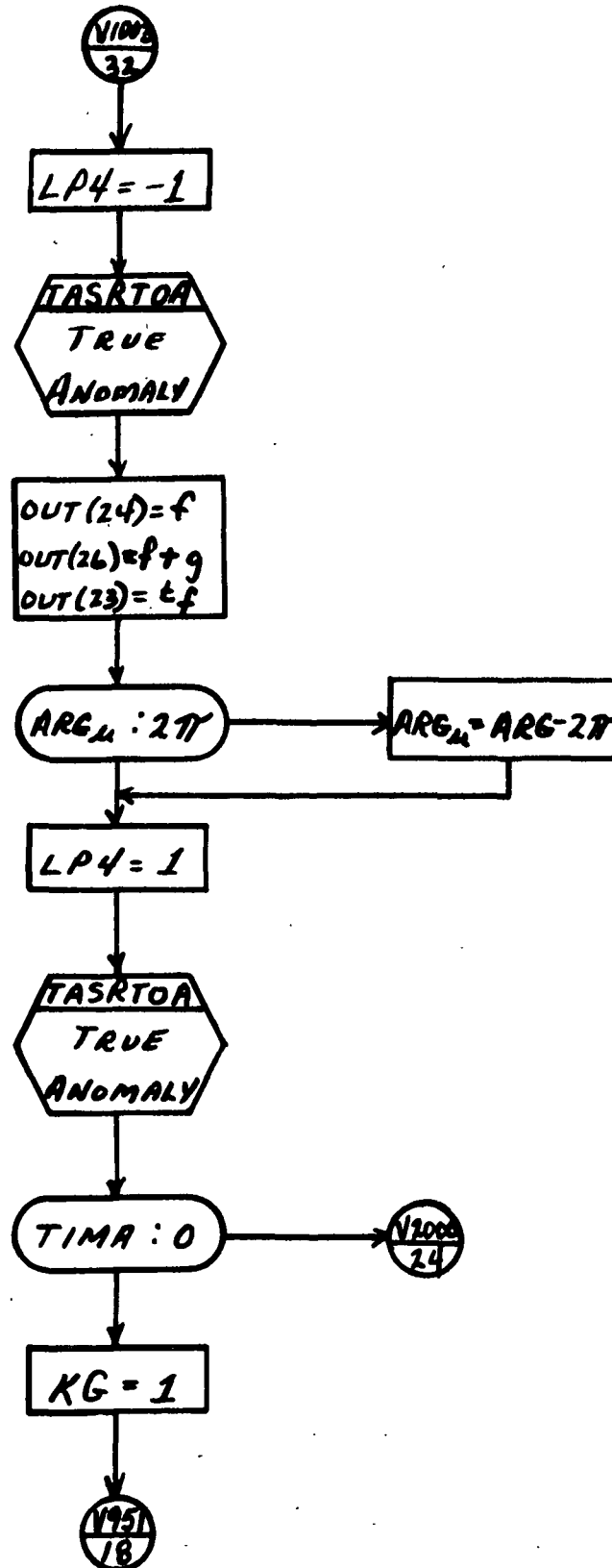


Figure 1. PMMAEG Flowchart

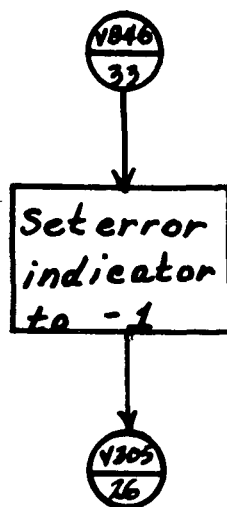


Figure 1. PMMAEG Flowchart

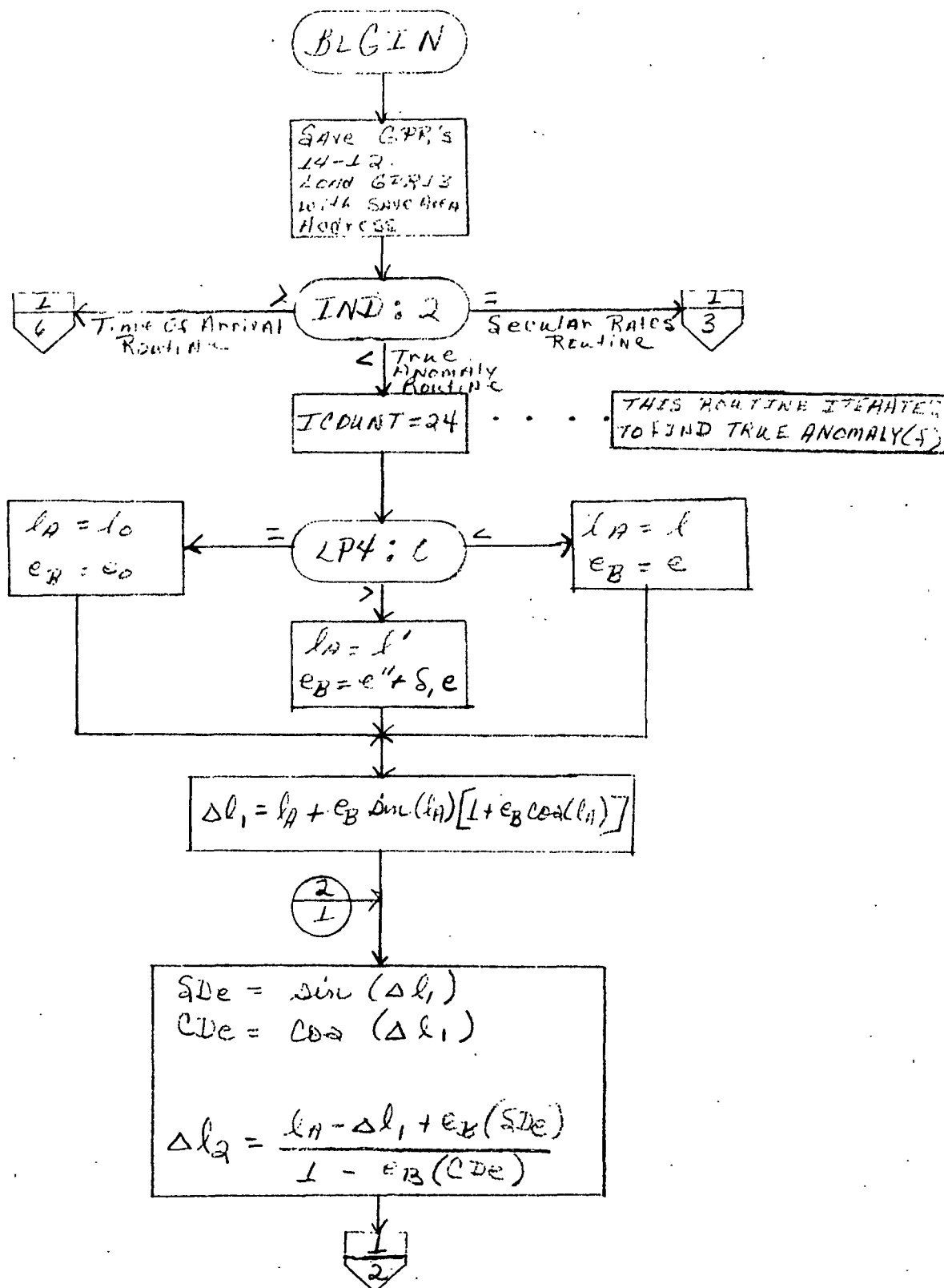


Figure 2. TASRTOA Flowchart

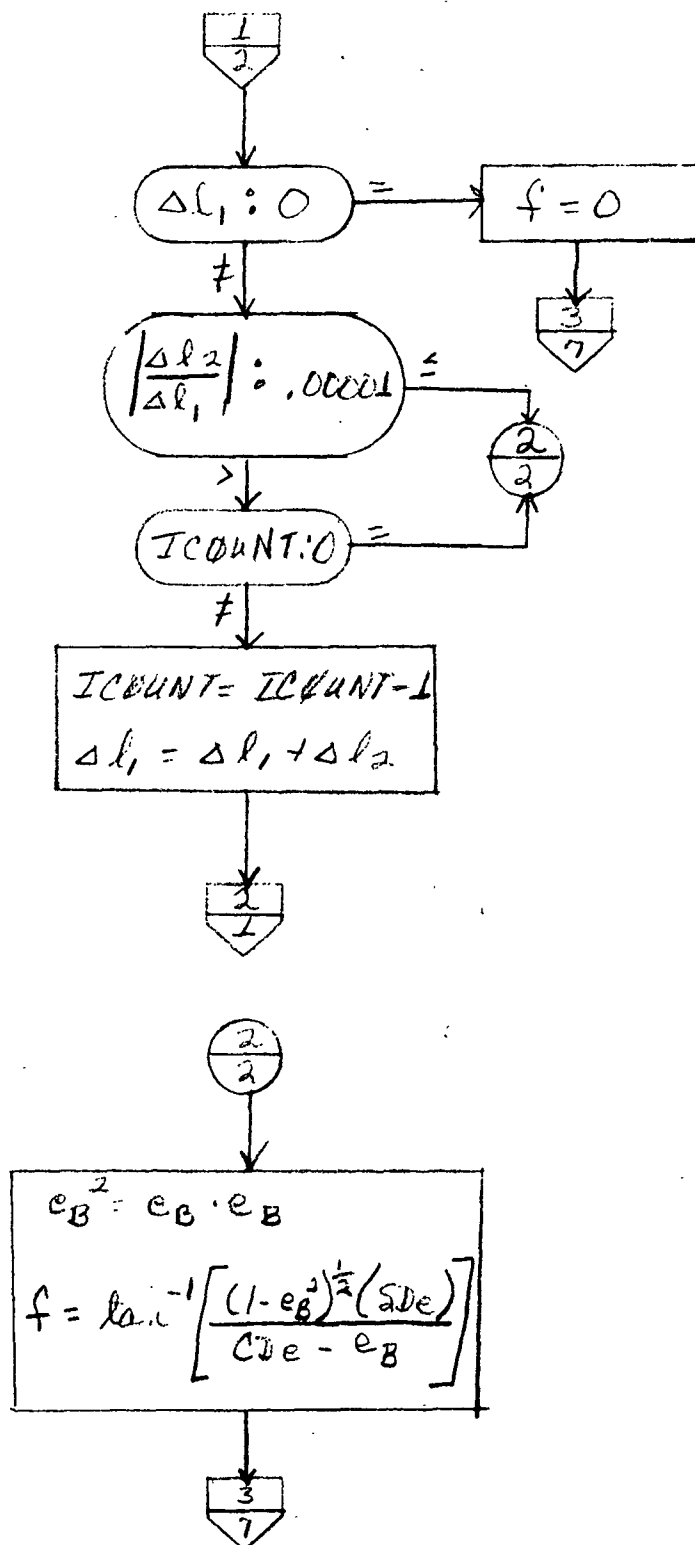


Figure 2. TASRTOA Flowchart

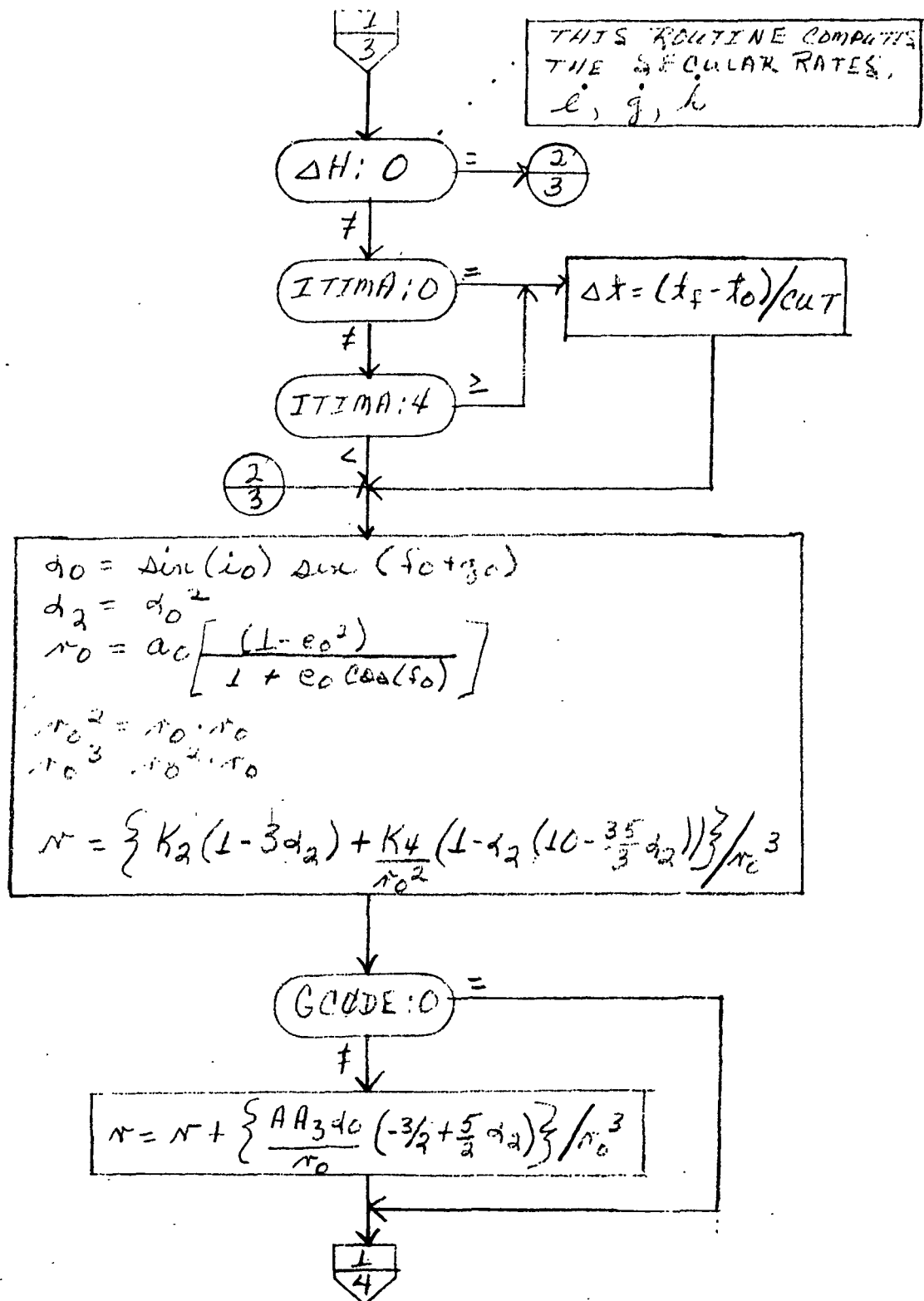


Figure 2. TASRTOA Flowchart

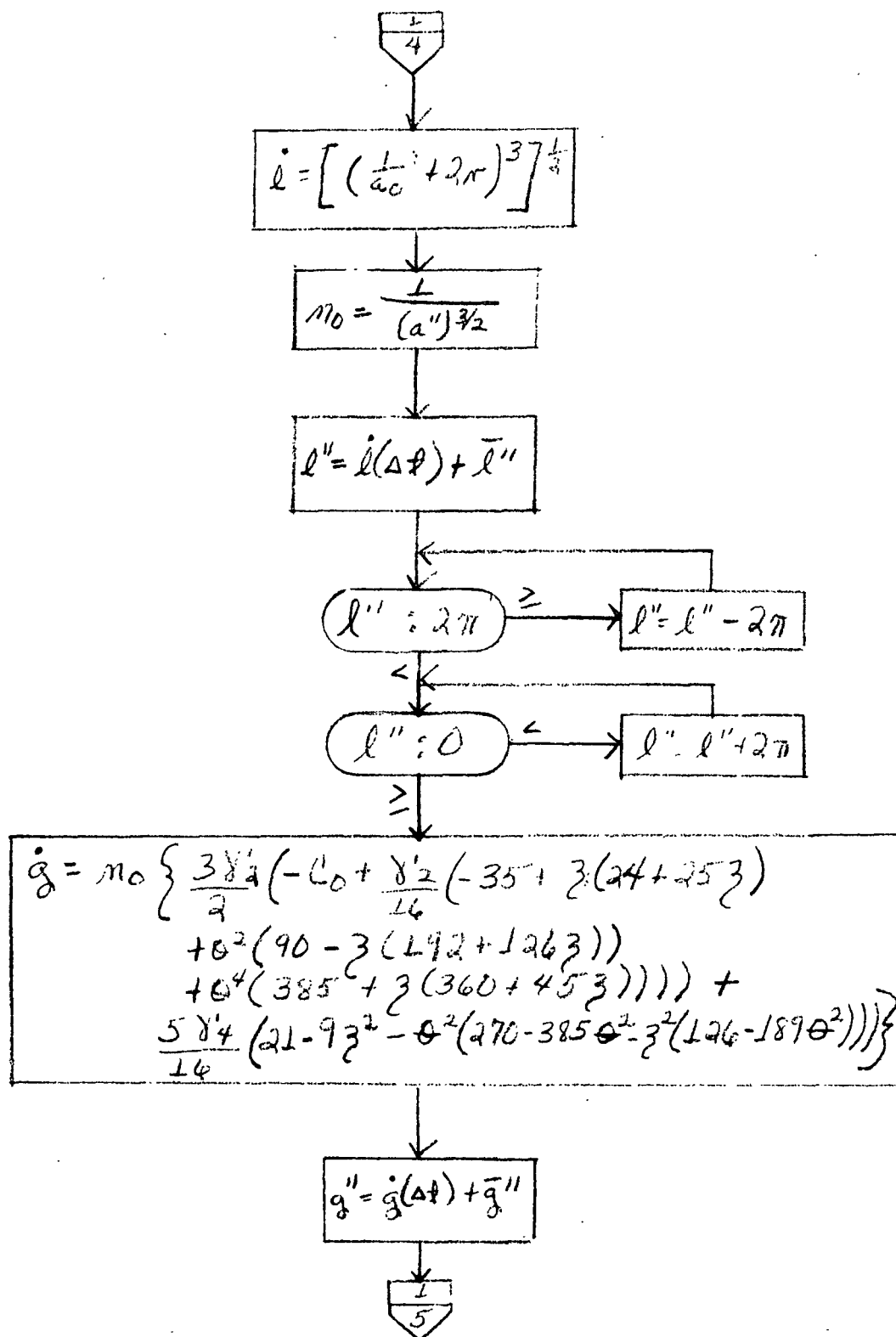


Figure 2. TASRTOA Flowchart

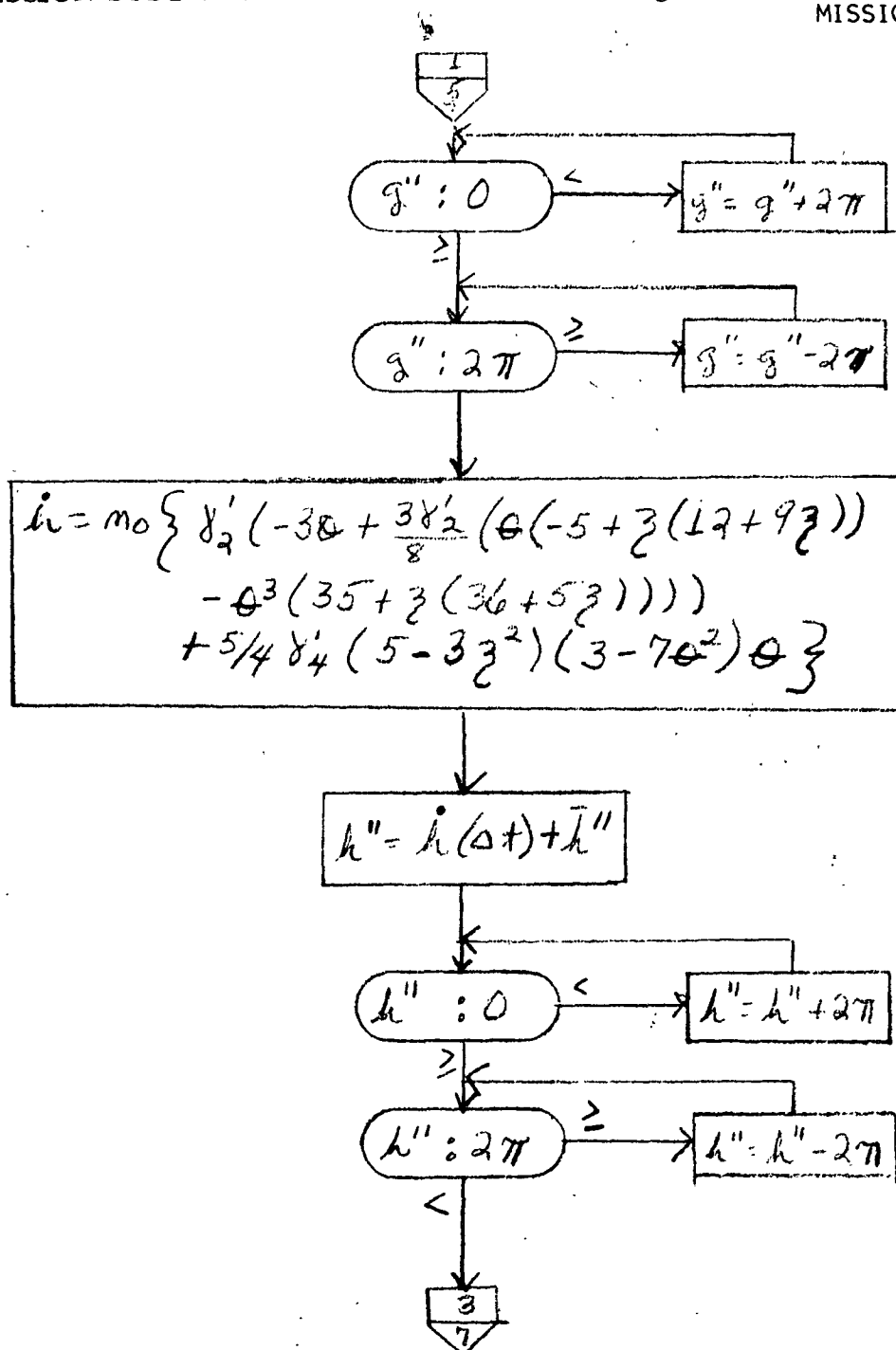


Figure 2. TASRTOA Flowchart

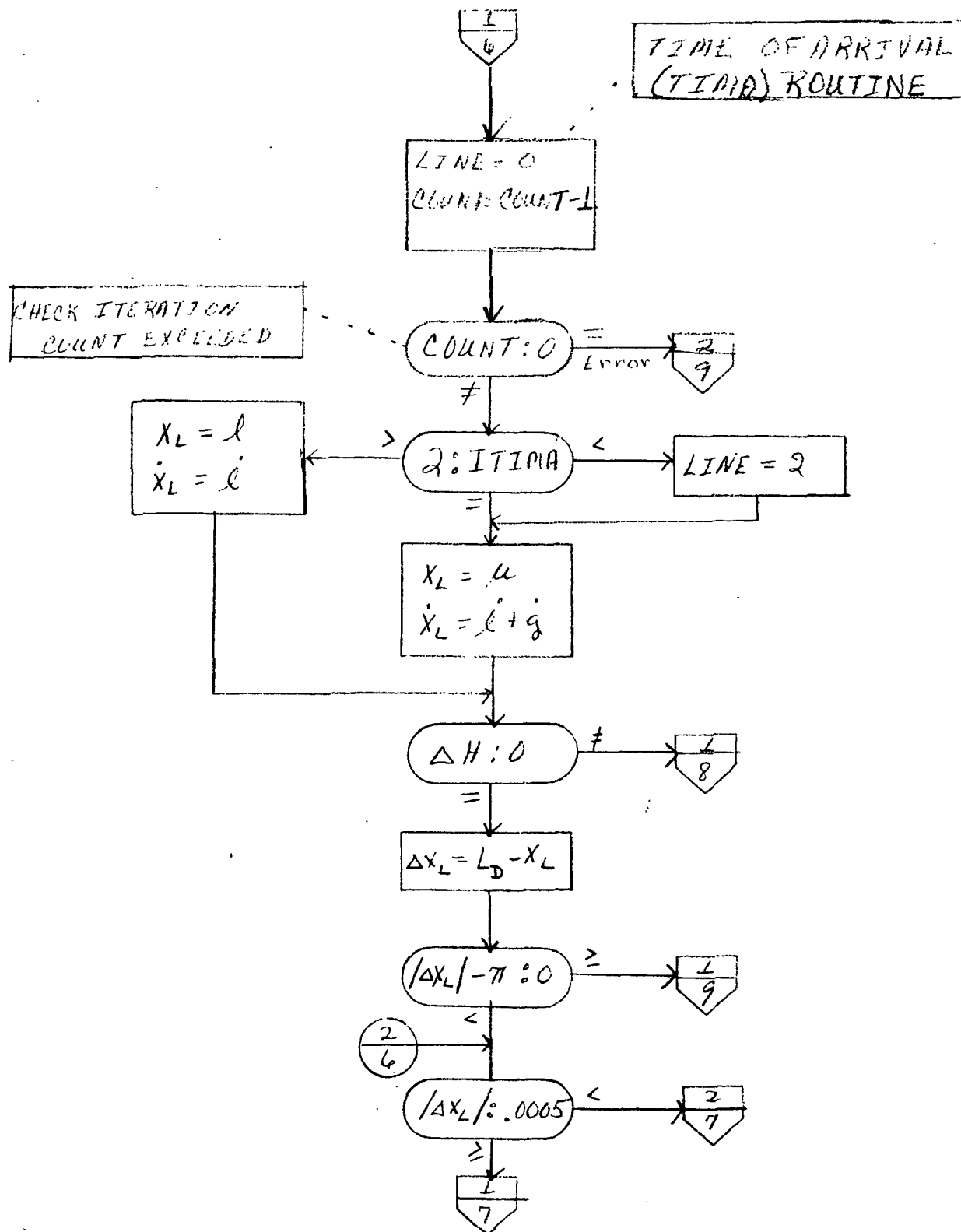


Figure 2. TASRTOA Flowchart

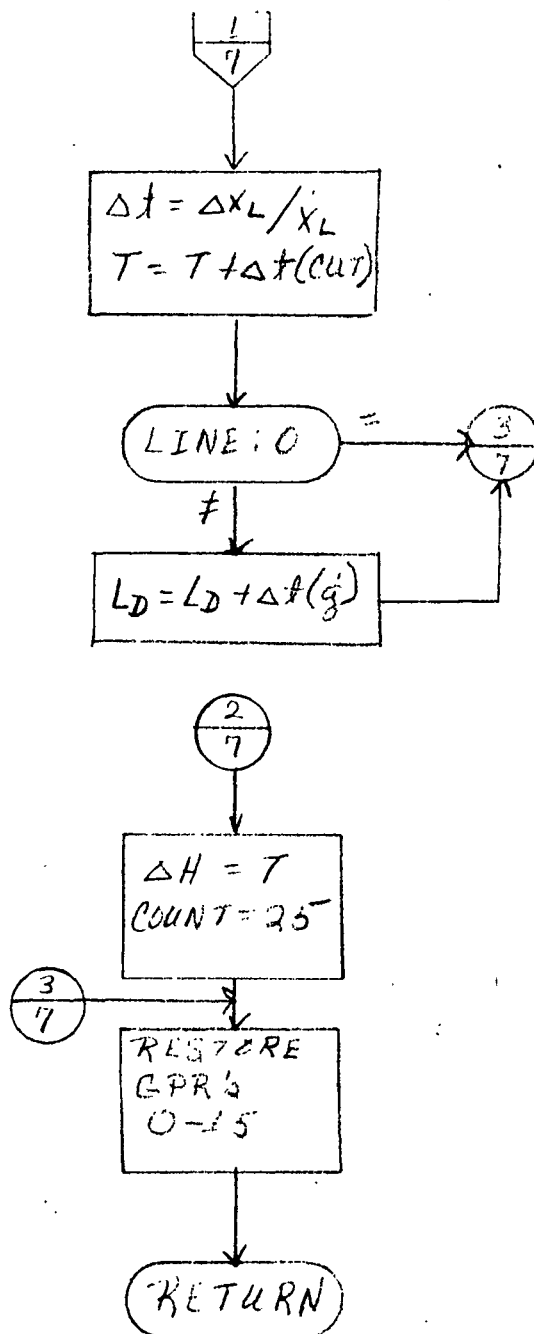


Figure 2. TASRTOA Flowchart

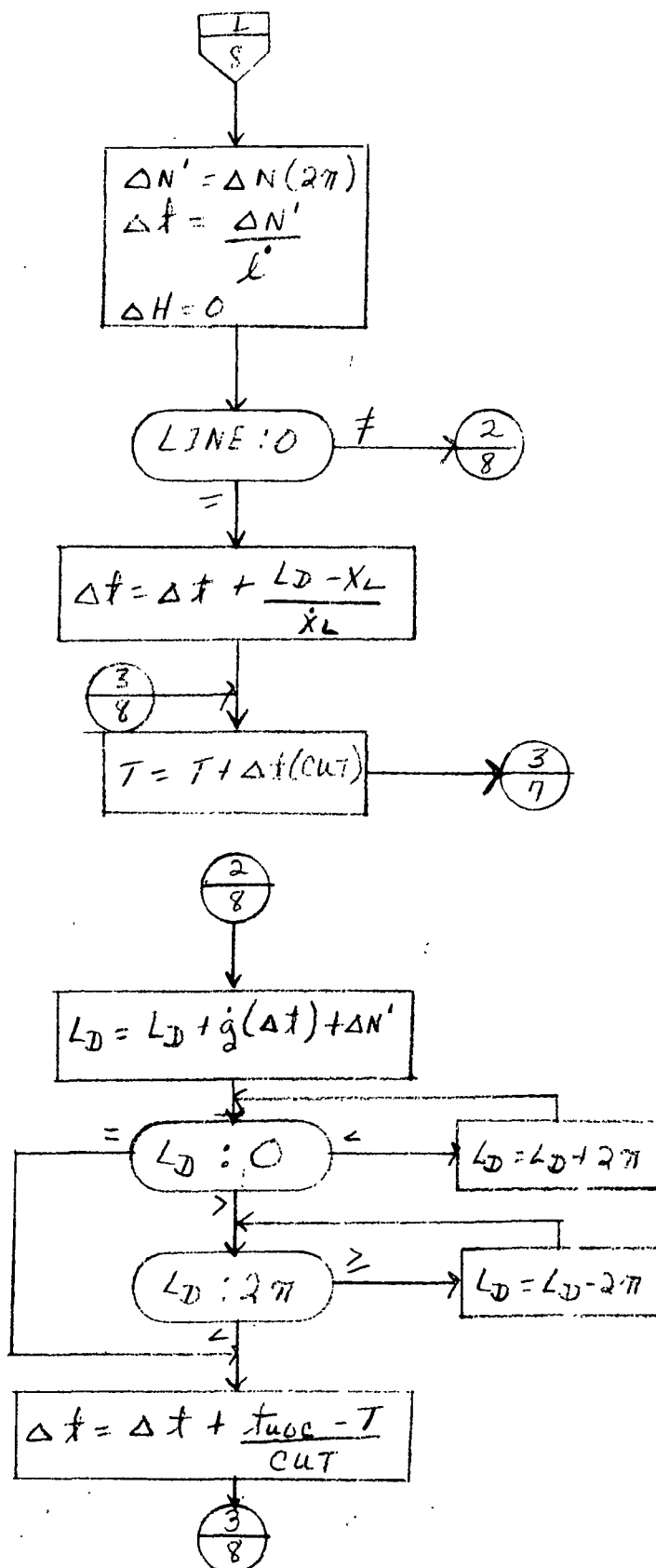


Figure 2. TASRTOA Flowchart

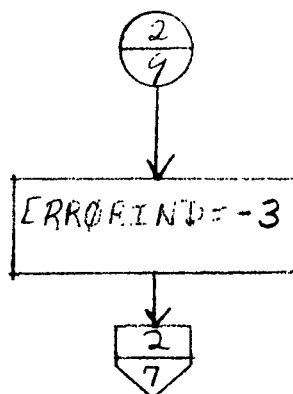
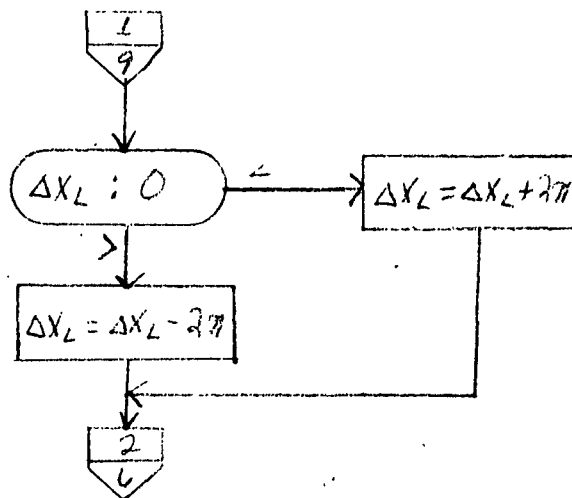


Figure 2. TASRTOA Flowchart

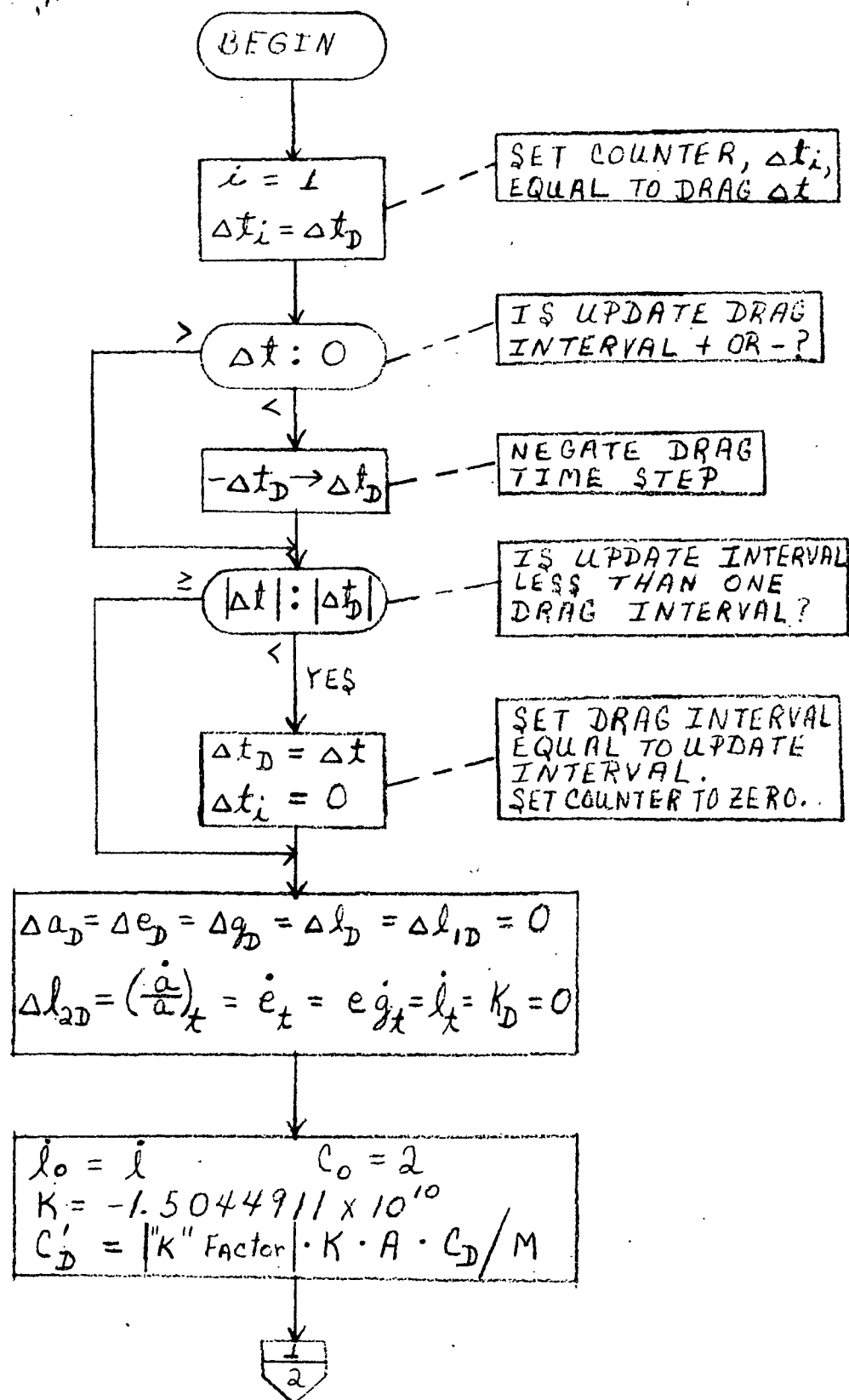


Figure 3. DRAG Flowchart

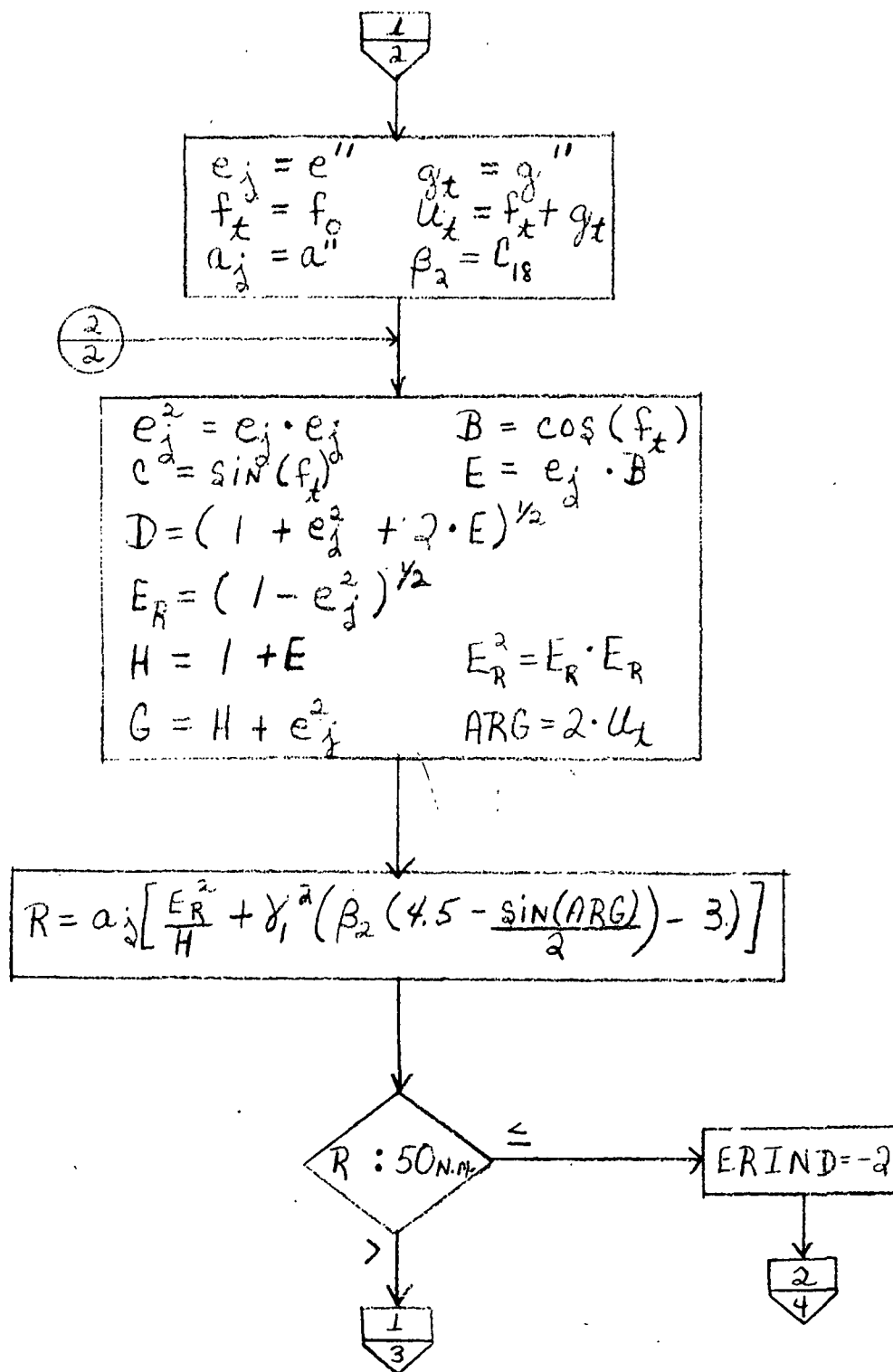


Figure 3. DRAG Flowchart

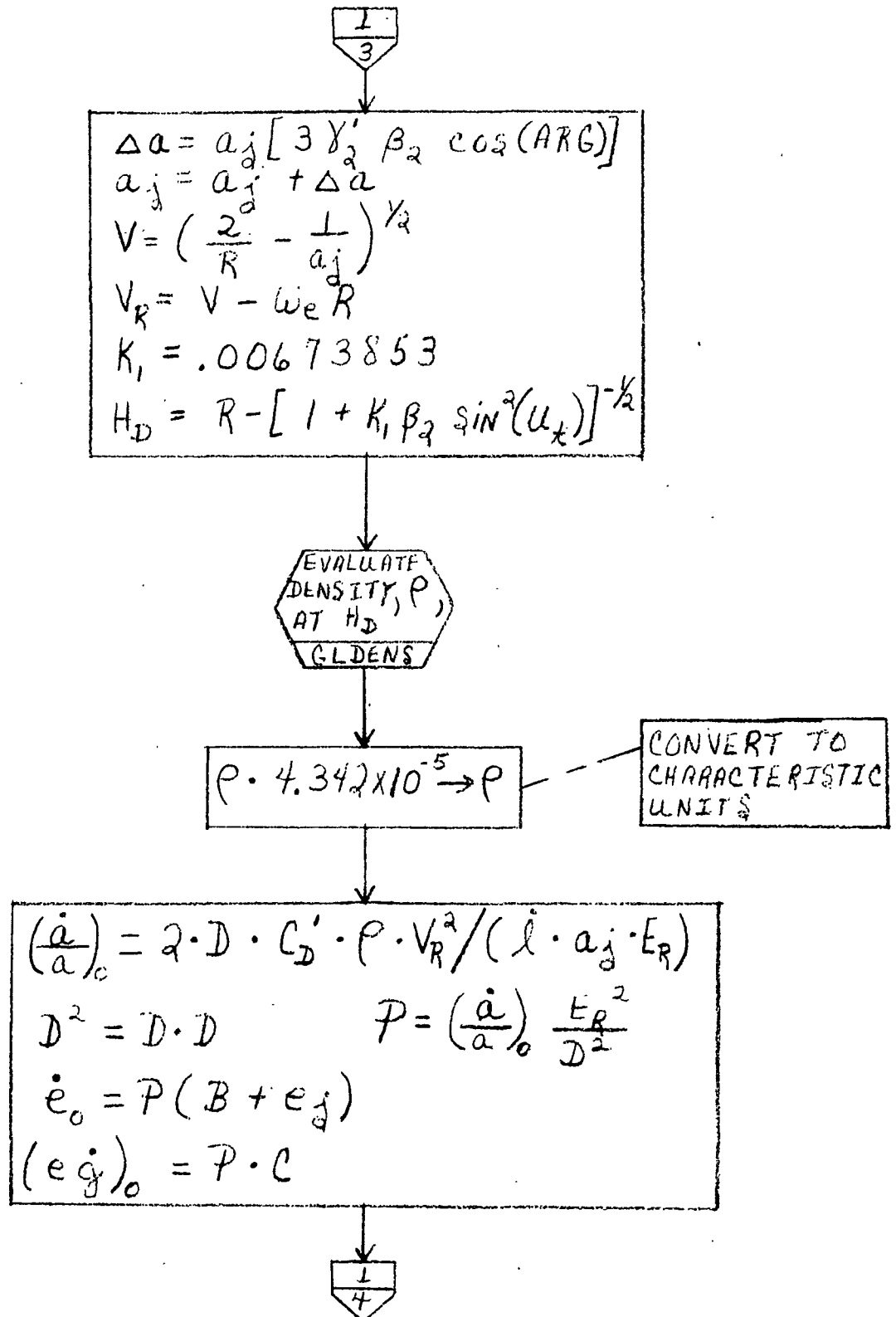


Figure 3. DRAG Flowchart

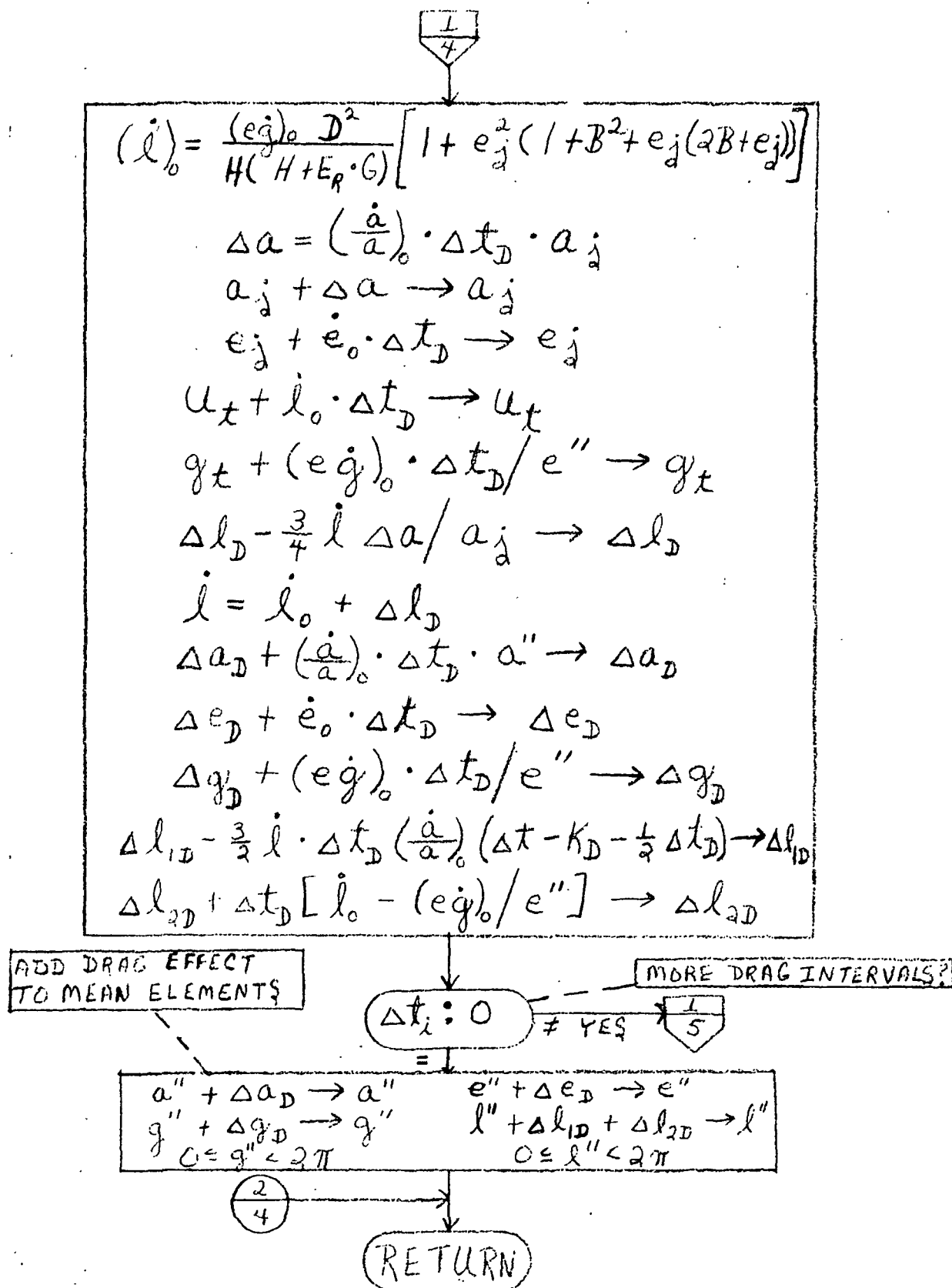


Figure 3. DRAG Flowchart

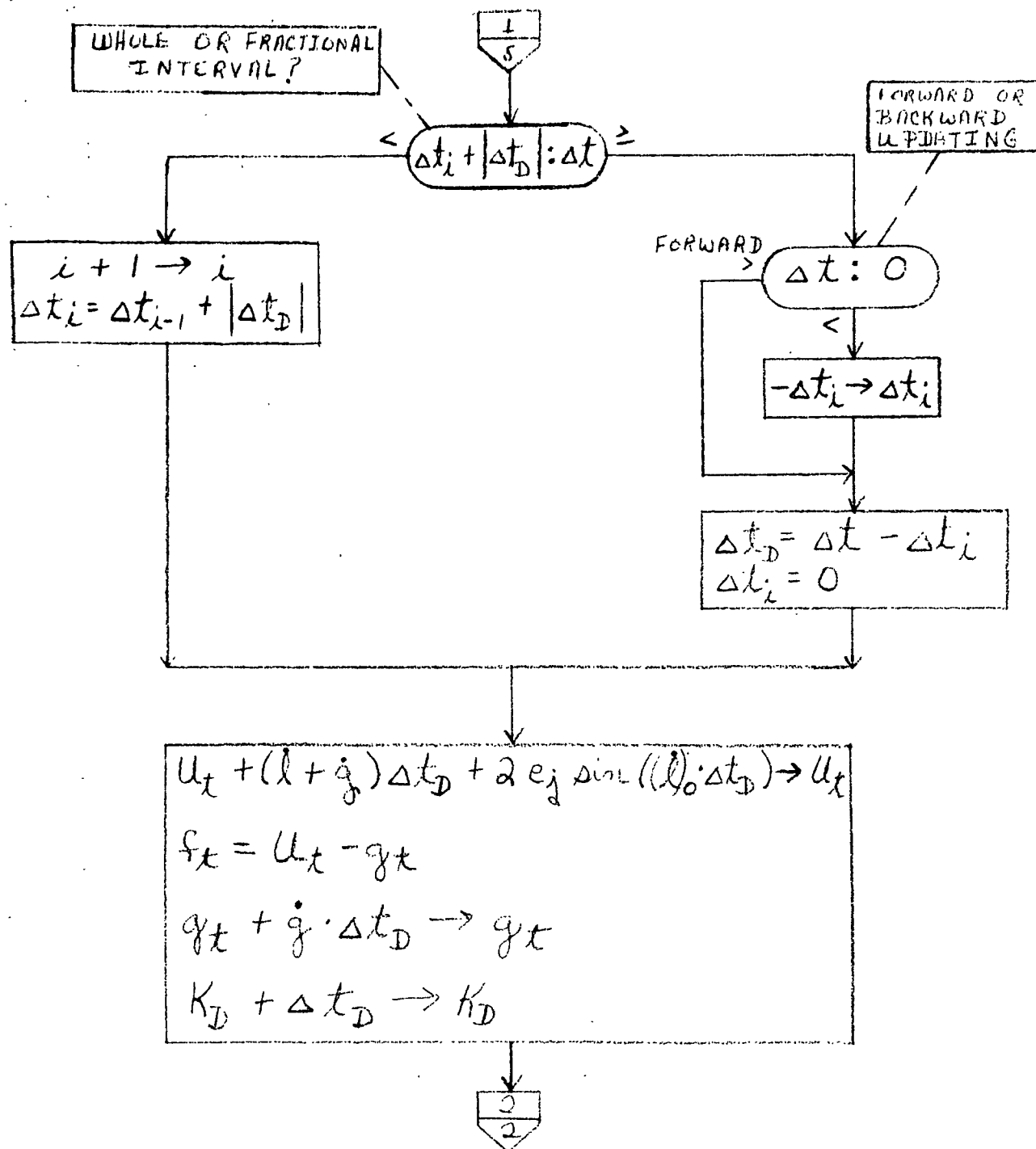


Figure 3. DRAG Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMDAN
 DATE 1/2/68 (EORP)
 REV
 PAGE 1 of 10

- a. Symbol — PMMDAN
- b. Program Name — Analytic Ephemeris Generator (AEG) Day/Night Determination
- c. Programming Group — Rendezvous Planning
- d. Programmer — M. R. Secondo
- e. Language — Assembler
- f. Data Tables Used — GZGENCSN
- g. MED's Used — P80
- h. Writeup Completion Date — 11/16/67
- i. System Parameters — Table PMMDAN-1
- j. Flowchart — Figure PMMDAN-4

FUNCTION

Given a set of vehicle elements (AEG format), PMMDAN will return the time and nature of the next one or two (optional) environment changes.

USAGE

- a. Calling Sequence — LINK EP=PMMDAN, PARAM=(ELM, IND, ERR, T1, T2)
- b. Inputs — The following quantities must be provided by the calling program:
 - 1. ELM is a 27 doubleword array in standard AEG format; PMMDAN uses this array. No provisions are made for saving the input quantities.
 - 2. IND is one word of storage indicating whether one or two predictions are desired; that is, IND=1 or IND=2.
- c. Outputs and Exit — The following quantities will be returned by PMMDAN:
 - 1. ERR is one word for an error return. The values are indicated in the Usage section of this writeup.
 - 2. T1 is a doubleword for the time of the upcoming environment change.
 - 3. T2 is a doubleword for the time of the next environment change.

For both T1 and T2, a positive time indicates an upcoming environment of daylight and a negative time for darkness. If an error exists, both of these values will be set to zero.
- d. Other Programs Used — GIMIKC, PMMAEG, DARCOS, DARSIN, DSQRT, PIEFEM

Approval	Approval
M. R. Secondo	A. Ametti

- e. Data Restrictions and Limits — None
- f. Error — ERR < 0; same as PMMAEG error returns
 ERR = 1, orbit is totally in daylight
 ERR = 2, convergence failure
 ERR = 0, no error

METHOD

The derivation is basically a function of the sun's position, satellite orbital elements, and the shadow of a spherical Earth. From this information, the time and nature of the upcoming environment change(s) can be computed. The sun's rays are considered to be parallel; hence no attention is given to refraction, diffusion, semidiameter or parallax. For the purposes of this program, these factors do not introduce significant error.

From Figure PMMDAN-1,

$$\cos \theta = \frac{\bar{R} \cdot \bar{R}_s}{|\bar{R}| |\bar{R}_s|} \quad \begin{array}{l} \bar{R} = \text{satellite position vector} \\ \bar{R}_s = \text{sun position vector} \end{array}$$

For $\cos \theta \geq 0$, the satellite is in daylight because it is less than (or equal to) 90° from the sun (central angle). If $\cos \theta < 0$, more information is needed.

Noting that $\sin \beta = -\cos \theta$, then $\cos \beta = +\sqrt{1 - (-\cos \theta)^2}$. Now if

$|\bar{R}| \cos \beta \geq R_e$ (radius of the Earth), it can be seen that the satellite is not yet in the Earth's shadow, and hence the present environment is daylight. Consequently, if $|\bar{R}| \cos \beta < R_e$, the present satellite environment is darkness.

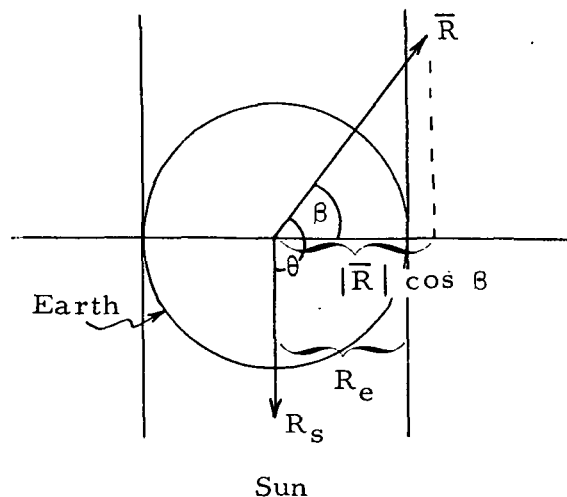


Figure PMMDAN-1

The nature of the upcoming environment change is now known (by virtue of knowing the present environment). To determine the time of the environment change, it is first necessary to compute

$$\bar{N}^* = \frac{\bar{R}_s \times \bar{h}}{|\bar{R}_s \times \bar{h}|} \quad \text{where } \bar{h} = \bar{R} \times \bar{V} \quad (\bar{V} = \text{satellite velocity vector}).$$

$\bar{N}^*$ lies along the intersection of the orbital plane and the plane normal to $\bar{R}_s$. If the present environment is daylight, $\bar{N}^*$ is set negative; otherwise it is used as computed.

Basically, the three angles φ_1 , φ_2 , and φ_3 , will be computed. The angle between $\bar{R}$ and $\bar{N}^*$ is φ_1 ($0 \leq \varphi_1 \leq \pi$). The angle between $\bar{N}^*$ and environment change is φ_2 ($0 < \varphi_2 < \pi/2$). The angle through which the satellite must travel to change environment is φ_3 ($0 \leq \varphi_3 < 2\pi$); φ_3 is computed from φ_1 and φ_2 .

It can be seen that

$$\varphi_1 = \cos^{-1} \left(\frac{\bar{R} \cdot \bar{N}^*}{|\bar{R}|} \right)$$

And, for the computation of φ_2

$$\cos \alpha = \frac{\bar{R}_e}{|\bar{R}|} \quad \sin \alpha = \sqrt{1 - \cos^2 \alpha}$$

$$\cos \eta = \frac{\bar{R}_s \cdot \bar{h}}{|\bar{R}_s| |\bar{h}|} \quad \sin \eta = \sqrt{1 - \cos^2 \eta}$$

From the spherical triangle in Figure PMMDAN-2,

$$\frac{\sin \varphi_2}{\sin \pi/2} = \frac{\sin \alpha}{\sin \eta}$$

$$\text{or} \quad \varphi_2 = \sin^{-1} \left(\frac{\sin \alpha}{\sin \eta} \right) \quad 0 < \varphi_2 < \pi/2$$

$\sin \eta \leq \alpha$ indicates that the upcoming orbit is totally in sunlight, and, for purposes of this program, an error exists.

After φ_3 is properly determined, it is divided by the mean motion of the satellite in order to add the time increment to the present time and advance the AEG to the new time. This will give a new set of elements for recomputing φ_3 . The process is repeated until a time increment test is satisfied, or an iteration limit is exceeded. The final result is the time of the upcoming environment change.

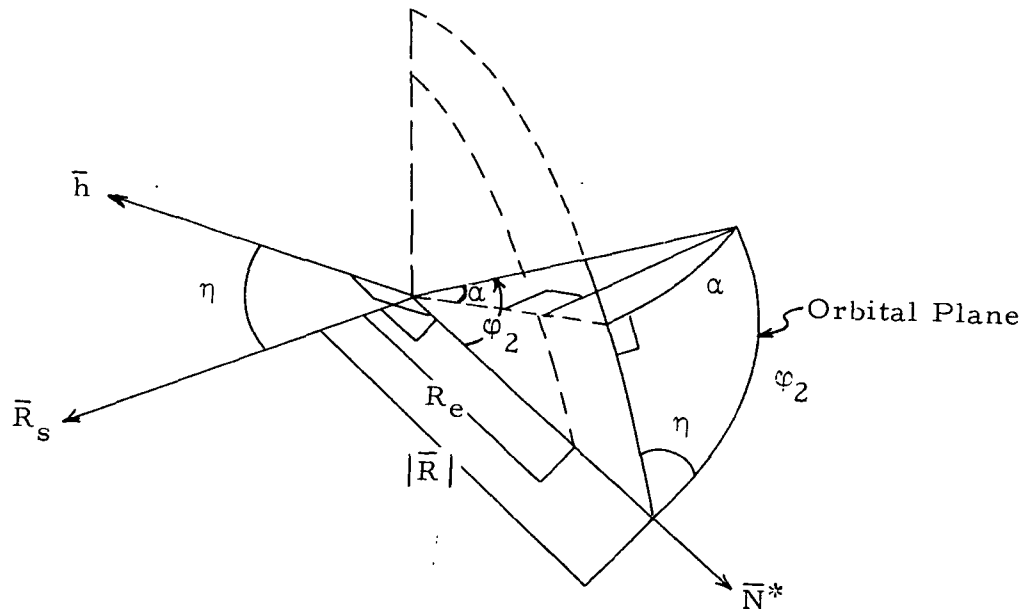


Figure PMMDAN-2.

If two predictions are desired, the same method is used for finding the time and nature of the first environment change. To compute the second change, an estimate is made for φ_3 , and the iterative process is used for finding the time of the next change.

That is, let

$\varphi_3 = \pi + 2\varphi_2$	$0 < \varphi_2 < \pi/2$	for an upcoming change to darkness
	$-\pi/2 < \varphi_2 < 0$	for an upcoming change to daylight

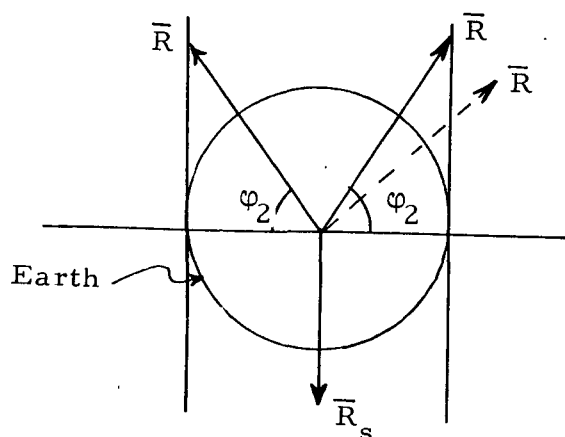


Figure PMMDAN-3

If the satellite is at the entrance point to darkness,

$$\Delta T = \frac{\varphi_3}{(1 + g)}$$

will put it at the entry point into daylight. The converse also holds true. The next step is to reenter the program, and, using the AEG, find a time corresponding to this next environment change

Table PMMDAN-1. System Parameters

[illegible]

*Type
H — Fixed-point half word
F — Fixed-point full word
E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

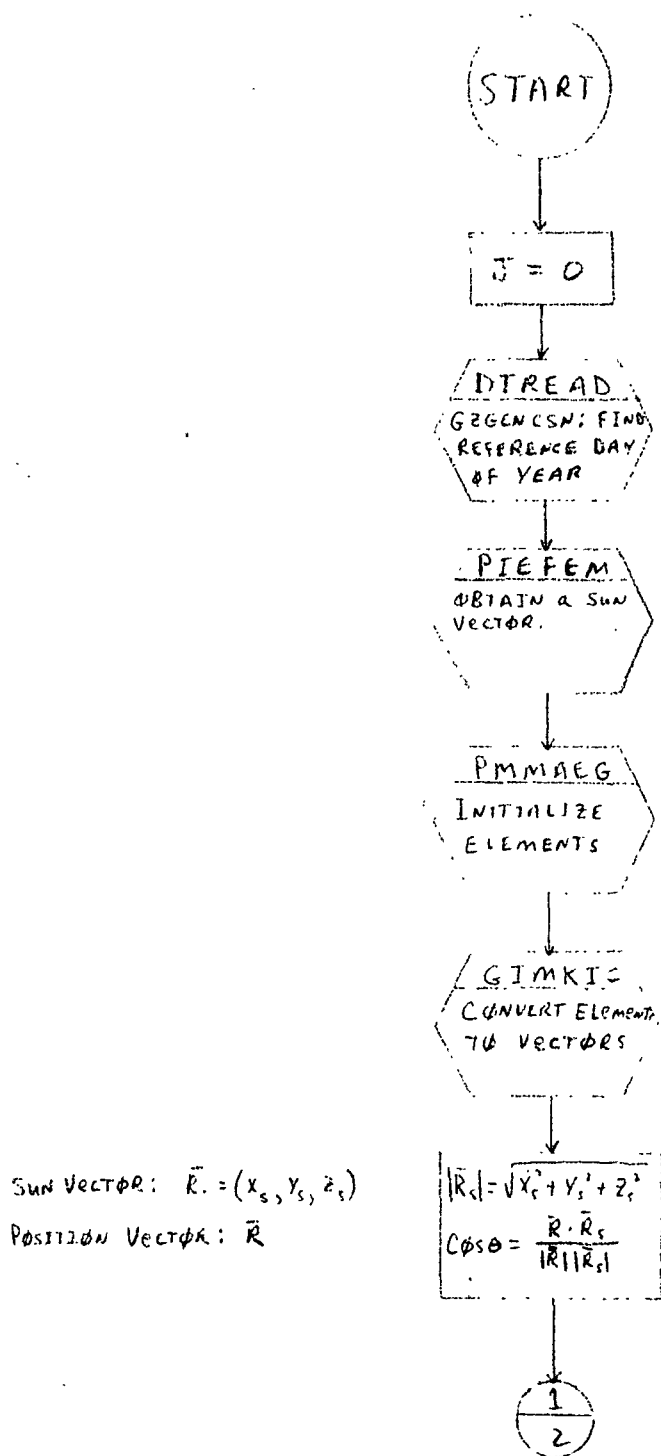


Figure PMMDAN-4. Flowchart

(Sheet 1 of 4)

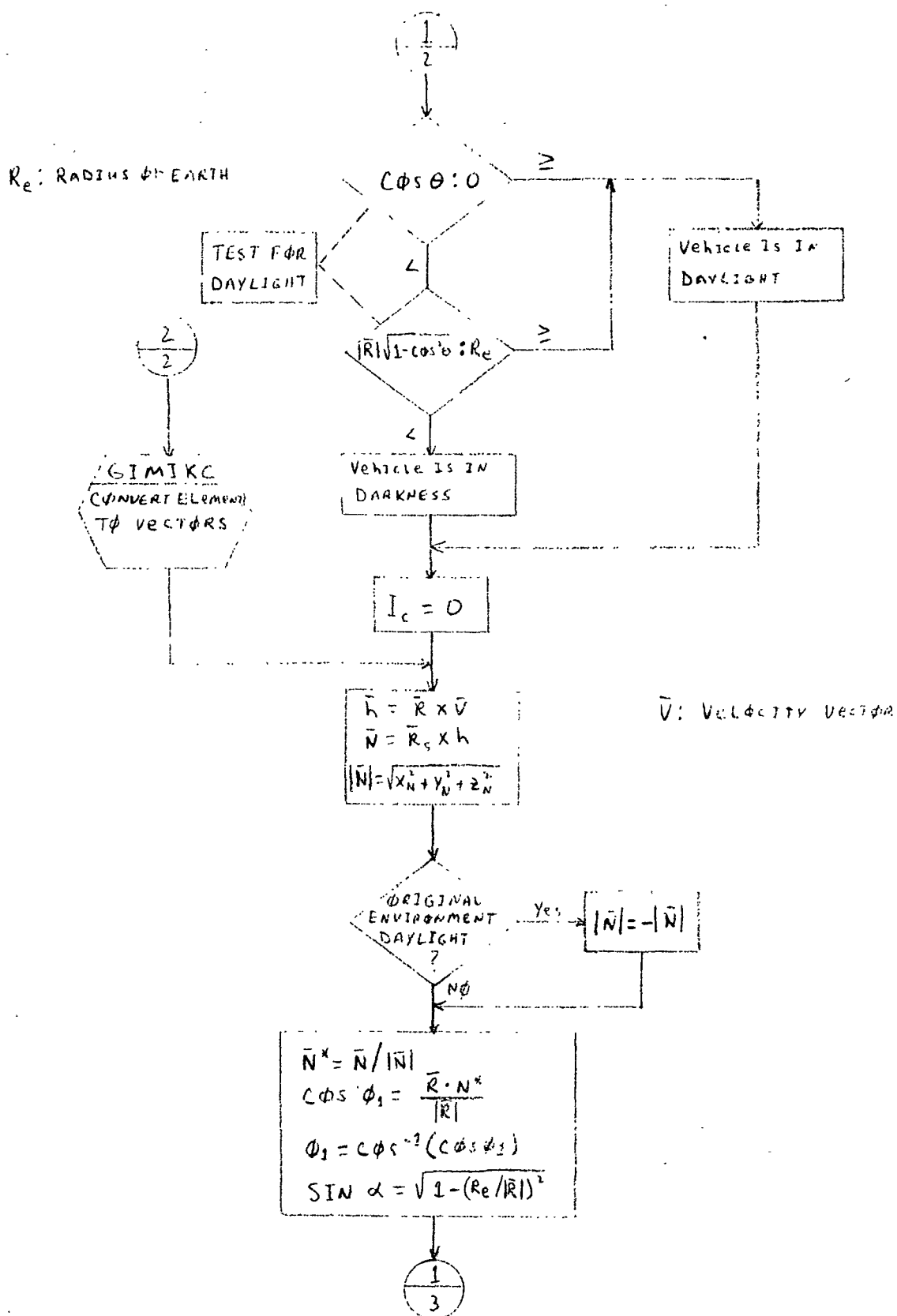


Figure PMMDAN-4. Flowchart

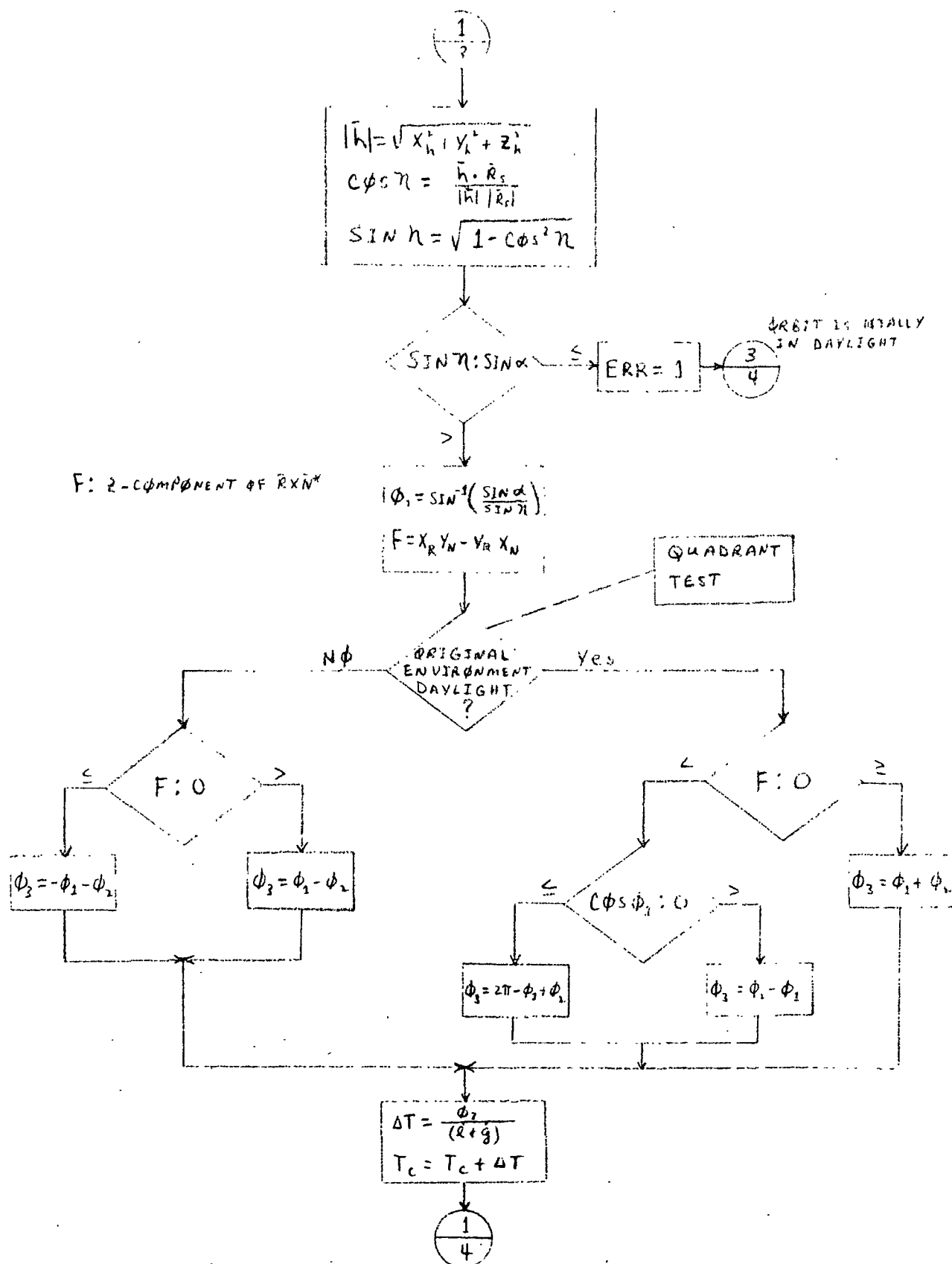


Figure PMMDAN-4. Flowchart

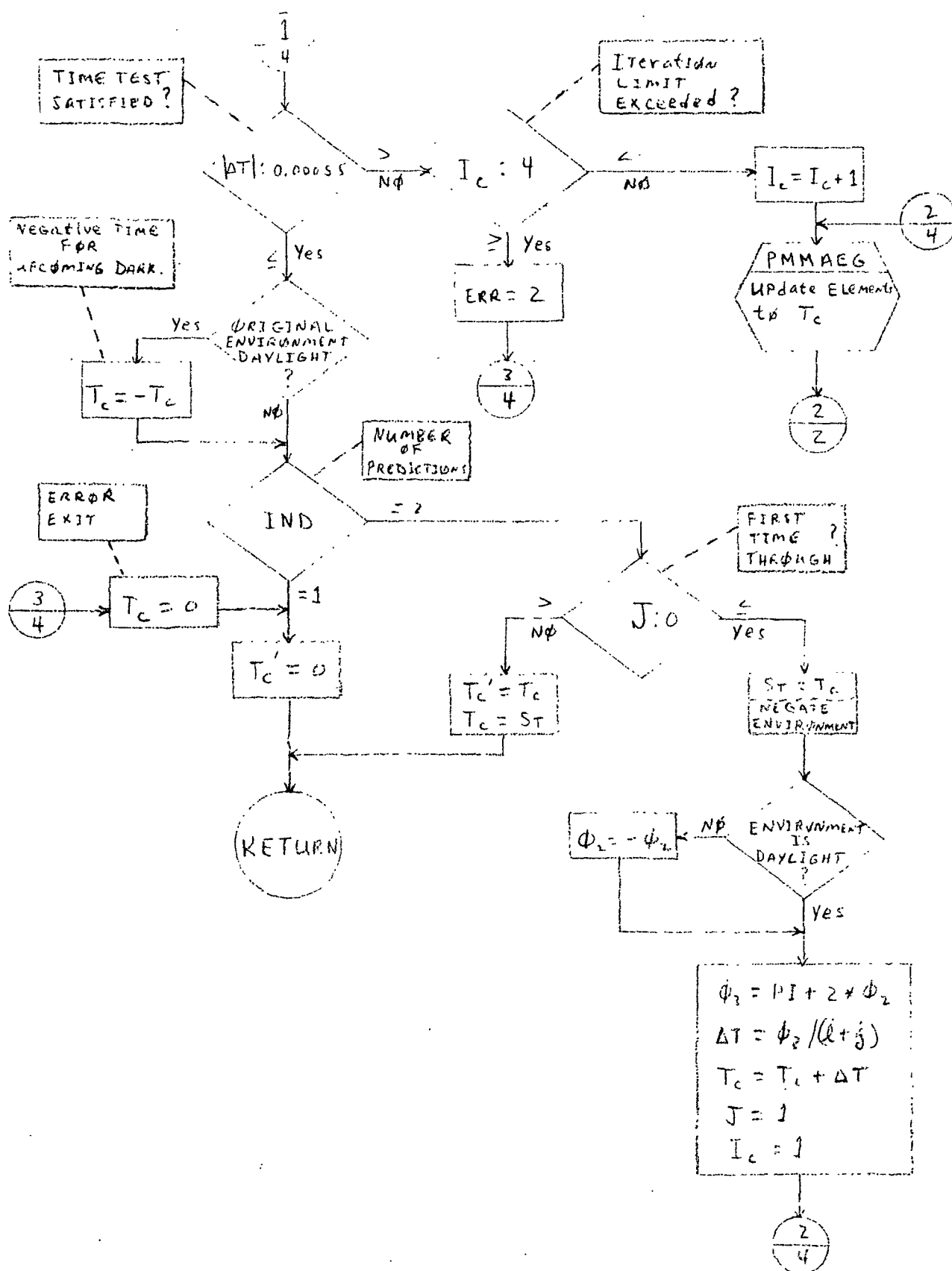


Figure PMMDAN-4. Flowchart

PMMDKI-PCMDKI
DOCKING INITIATION CONTROL LOGIC
SUBROUTINEREFERENCES

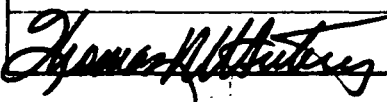
1. Programming Group — Rendezvous Planning
2. Programmers — R. Barrett, T. Hickey
3. Language — Assembler
4. Writeup Completion Date — 5/13/68
5. Data Tables Used — PZDKIT, PZDKIELM, GZGENCSN
6. MED's Used — K05, K00, K01, M07
7. Flowchart — Figure 1

FUNCTION

This section of the load module PMMDKI has the control logic responsibilities for the entire unit. All MED requests are routed to this CSECT for control of all calculations using the math CSECT's PCPICK, PCTETR, and the necessary load modules for other computations.

USAGE

1. Calling Sequence — All MED requests = RTATTACH EP = PMMDKI
2. Inputs — K00 or K01 MED parameters, and initialization quantities from K05 MED
3. Outputs and Exit — Data tables PZDKIT, and PZDKIELM. Exit is via both a XCTL to PMMXFR for M70, and a RETURN macro for K00 and K01.
4. Other Programs Used — CSECT PCPICK, PMMAEG, PMXSPT, PMMSPQ, PMSTICN, PMMMLC, PMMPHL, PMMITL, PMMSCH
5. Data Restrictions and Limits — None other than restrictions controlled by MED logic
6. Error — The following error messages may be printed on line during execution of this load module:
 - a. "PMMDKI - A REENTRANT TRAJECTORY IN PMMPHL. PLAN
 - b. "PMMDKI - A REENTRANT TRAJECTORY IN PMMSPQ. NO PLAN GENERATED."
 - c. "PMMDKI - UNABLE TO SCHEDULE PLAN IN PMMSCH. PLAN DELETED."
 - d. "PMMDKI - PLAN x FAILED TO CONVERGE IN PMMITL. PLAN RETAINED."

Approval	Approval
	

- e. "PMMDKI - UNABLE TO OBTAIN TWO IMPULSE SOLUTION. NO PLAN TRANSFER."
- f. "PMMDKI - UNABLE TO ADD TWO IMPULSE SOLUTION TO SPQ. NO PLAN GENERATED."
- g. "PMMDKI - CONSTRAINT MINIMUM — VIOLATED IN PMMSPQ. (TIME, DELTA V, OR PERIGEE)
- h. "PMMDKI - DKI OR SPQ ELEVATION ANGLE PRIOR TO PREVIOUS MANEUVER. NO PLAN GENERATED."

All other error conditions are set internally for the appropriate action.

METHOD

1. DKI Generate

Upon receipt of the K00 MED, the DKI control logic reads the GZGENSCN data table for its initialization quantities and moves these and the MED quantities into a common area for further use. A queue is sent to the Maneuver Line Definition Program (PMMMLC) to set up the analytic ephemeris generator (AEG) initial block and a maneuver line definition.

For each plan requested, a queue is sent to the maneuver scheduling unit (PMMSCH), the maneuver computational unit (PMMITL), and to the terminal phase maneuver computations (PMSTICN). If all units are able to complete their computations, the plan is saved, and the elements before and after each maneuver are stored.

2. SPQ Generate

Upon receipt of the K01 MED, the logic loads the common area of PMMSPQ and PMMDKI with the MED requests, the K05 initialization quantities, and the AEG elements for the chaser vehicle and target vehicle. The AEG elements are computed by PMMMLC as in the K00 MED, but no maneuver line is computed. A queue is sent to the co-elliptic sequence unit (PMMSPQ) to compute a rendezvous solution. If the unit completes its computations, the terminal phase maneuvers are computed by PMSTICN, and the plan is saved as well as the elements before and after each maneuver.

Upon completion of both the K00 and K01 computations, a queue is sent through the display control logic GMSNAP to update the Rendezvous Evaluation Table Display and the Rendezvous Planning Table Display.

IBM RTCC APOLLO PROGRAMMING SYSTEMS2. PMMDKI-PCMDKI
DATE 6/28/68 (EORP)
REV
PAGE 3 of 15

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

3. Transfer DKI

Upon receipt of the DKI transfer request, the plan requested is read from the data table, the elements after the last maneuver are updated to TPI time, and the terminal phase maneuvers are computed by PMSTICN. A queue is then sent to the maneuver control logic PMMXFR for transferring the plan to the Mission Plan Table.

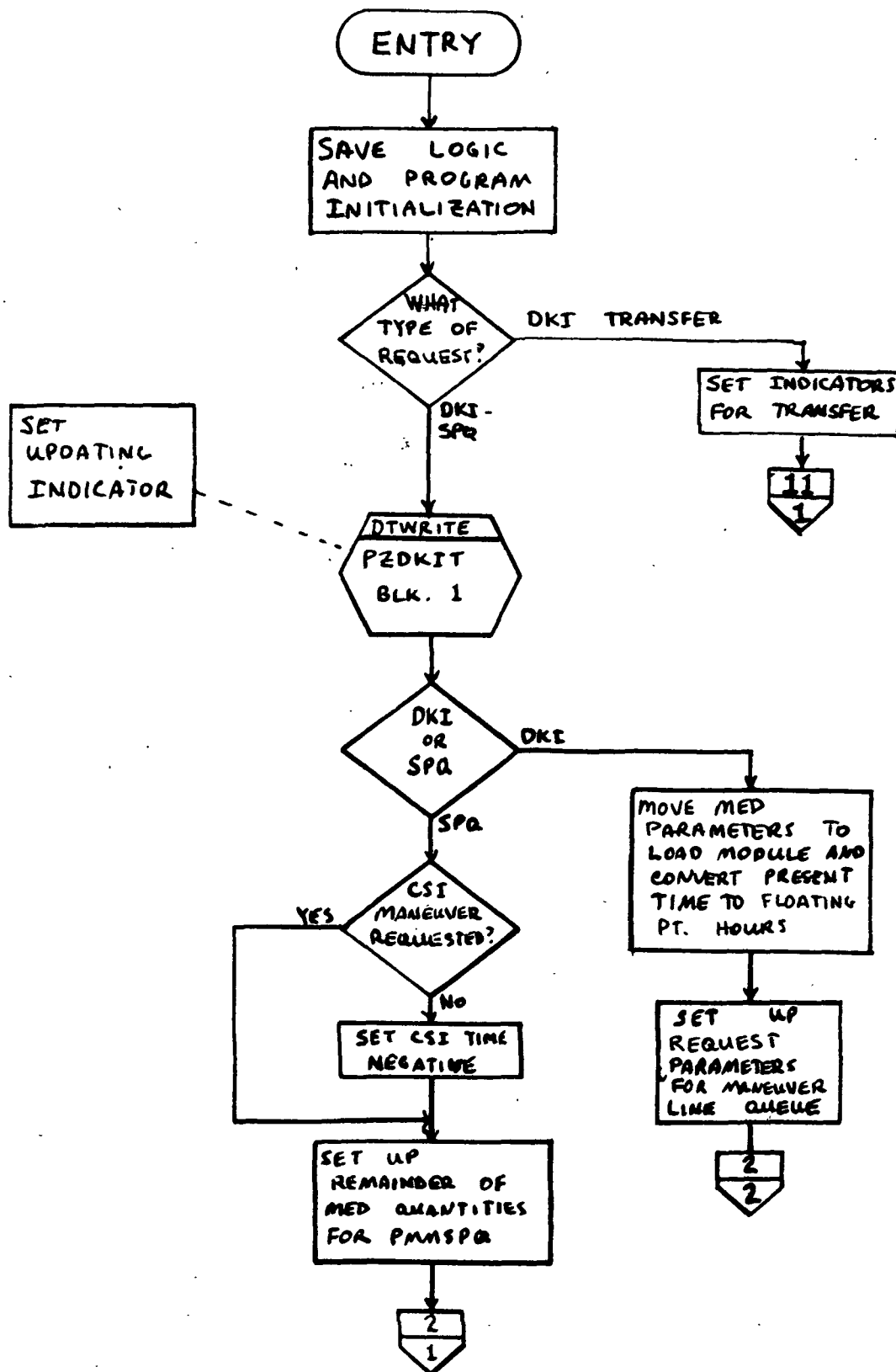


Figure 1

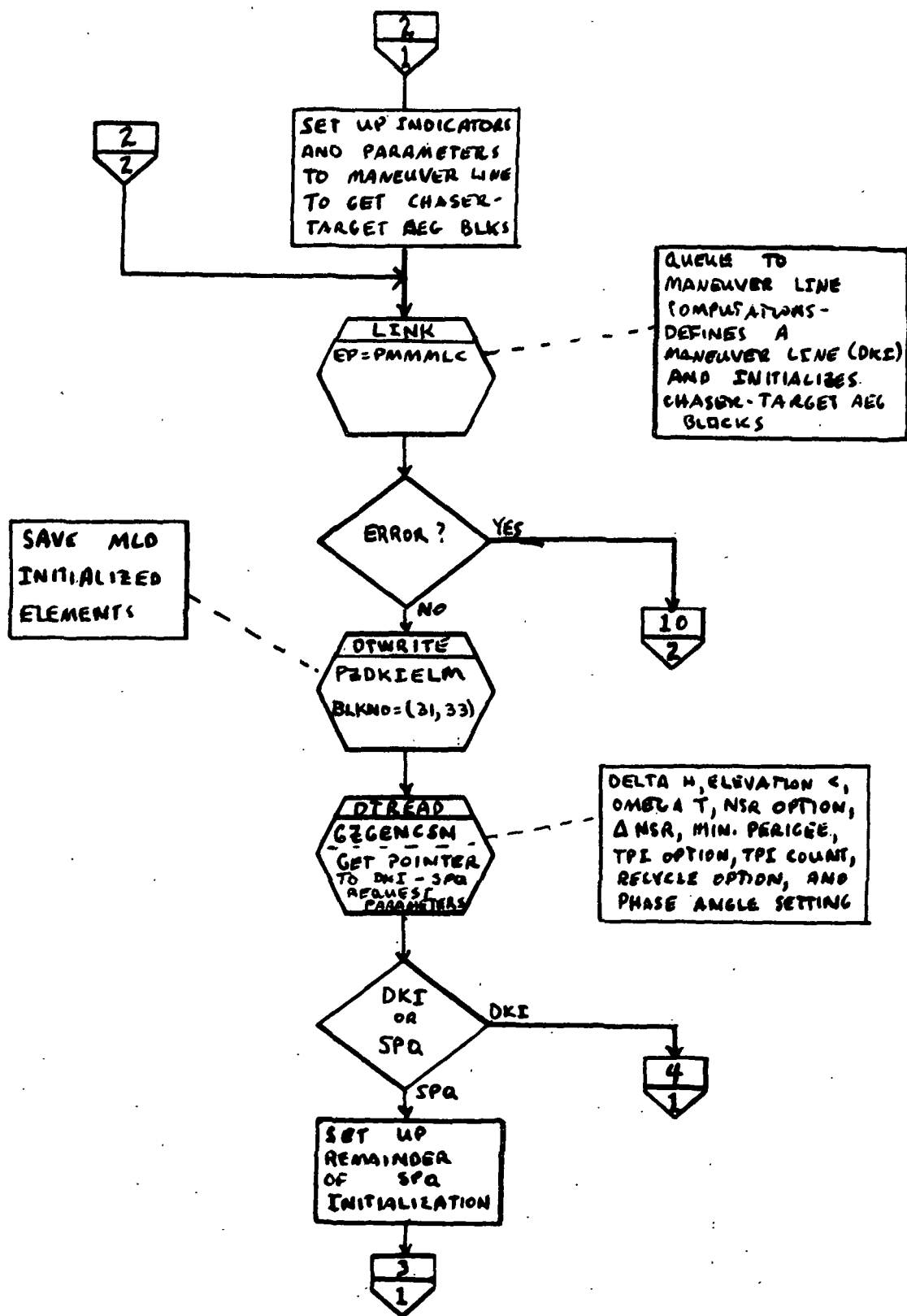


Figure 1

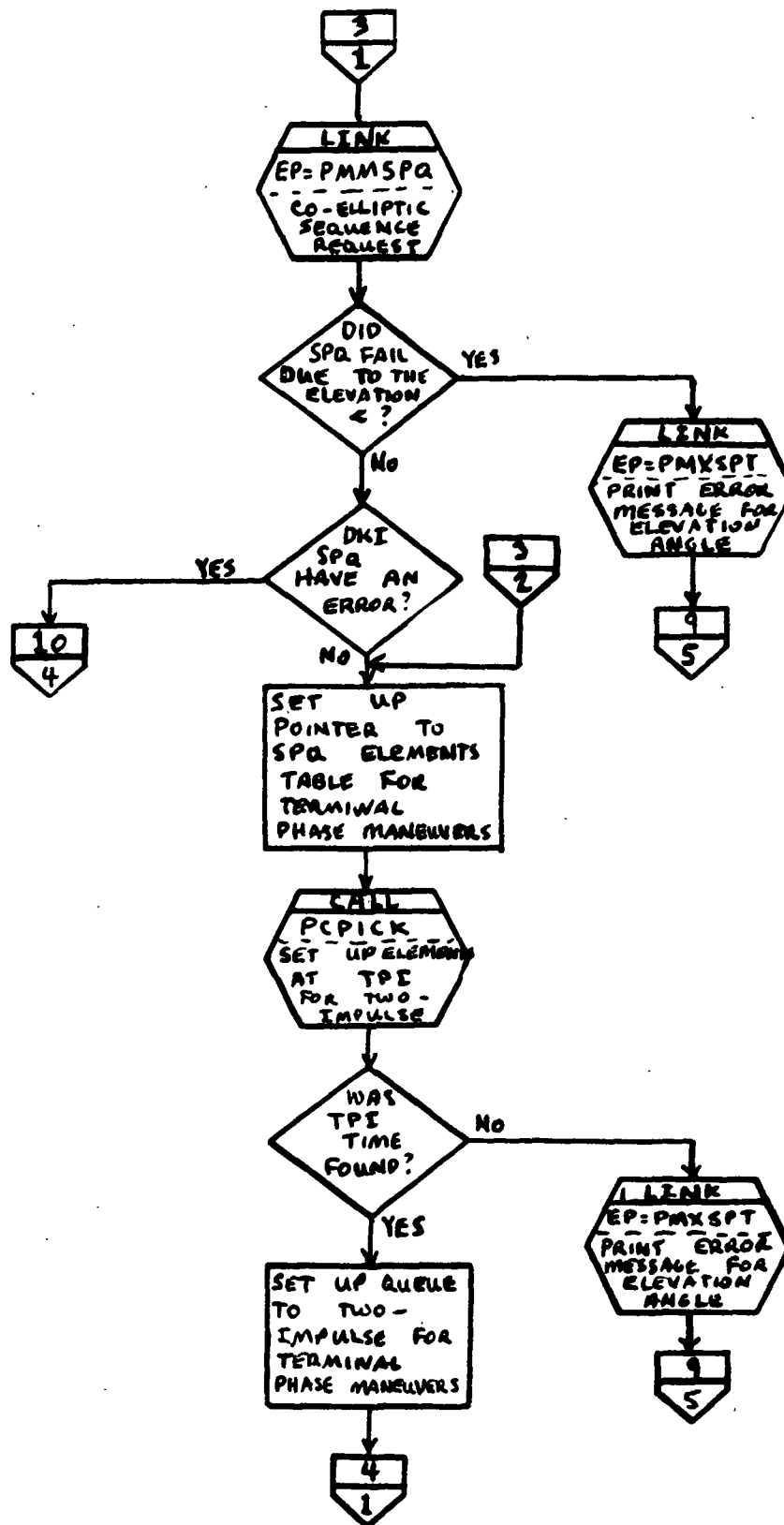


Figure 1

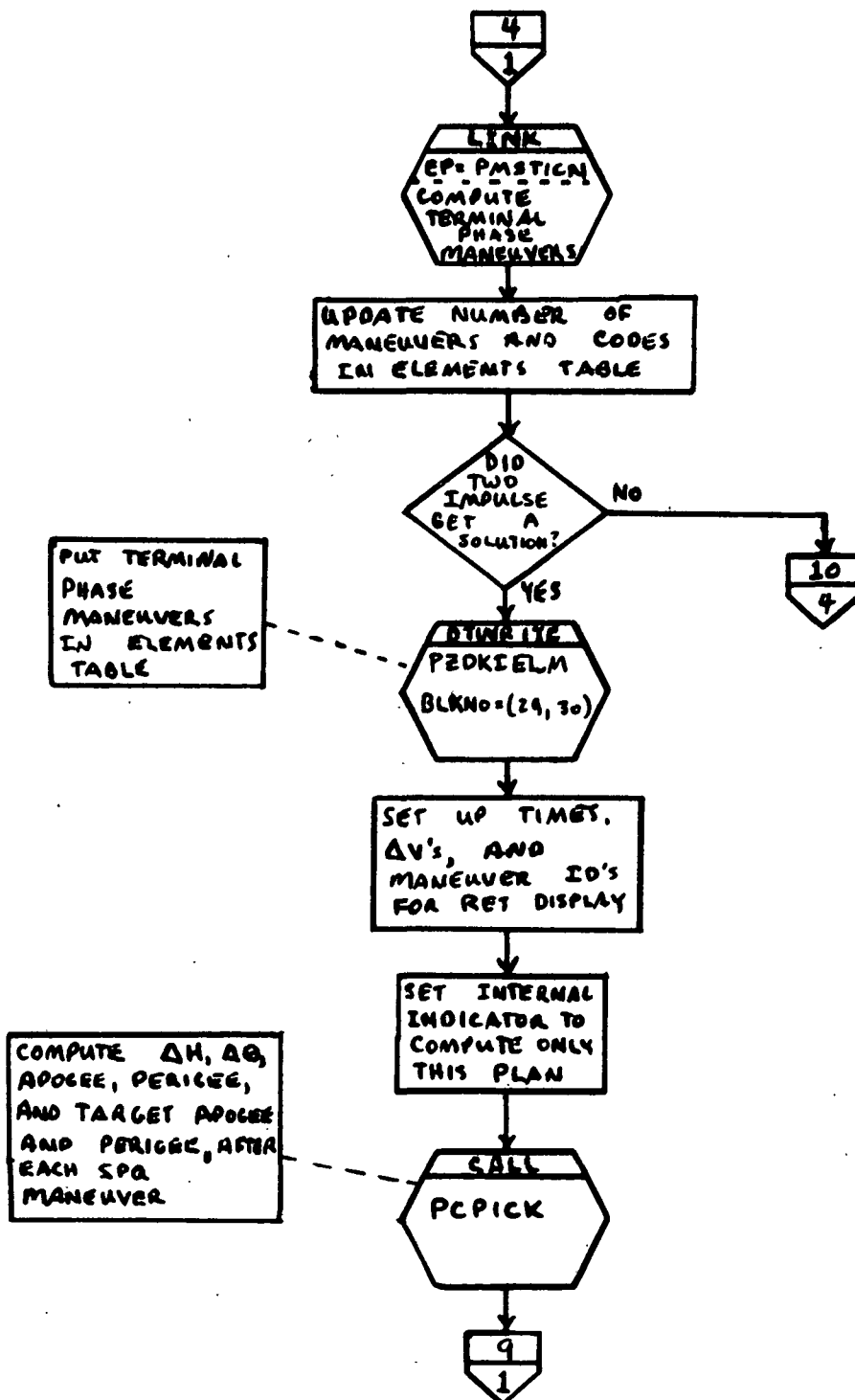


Figure 1

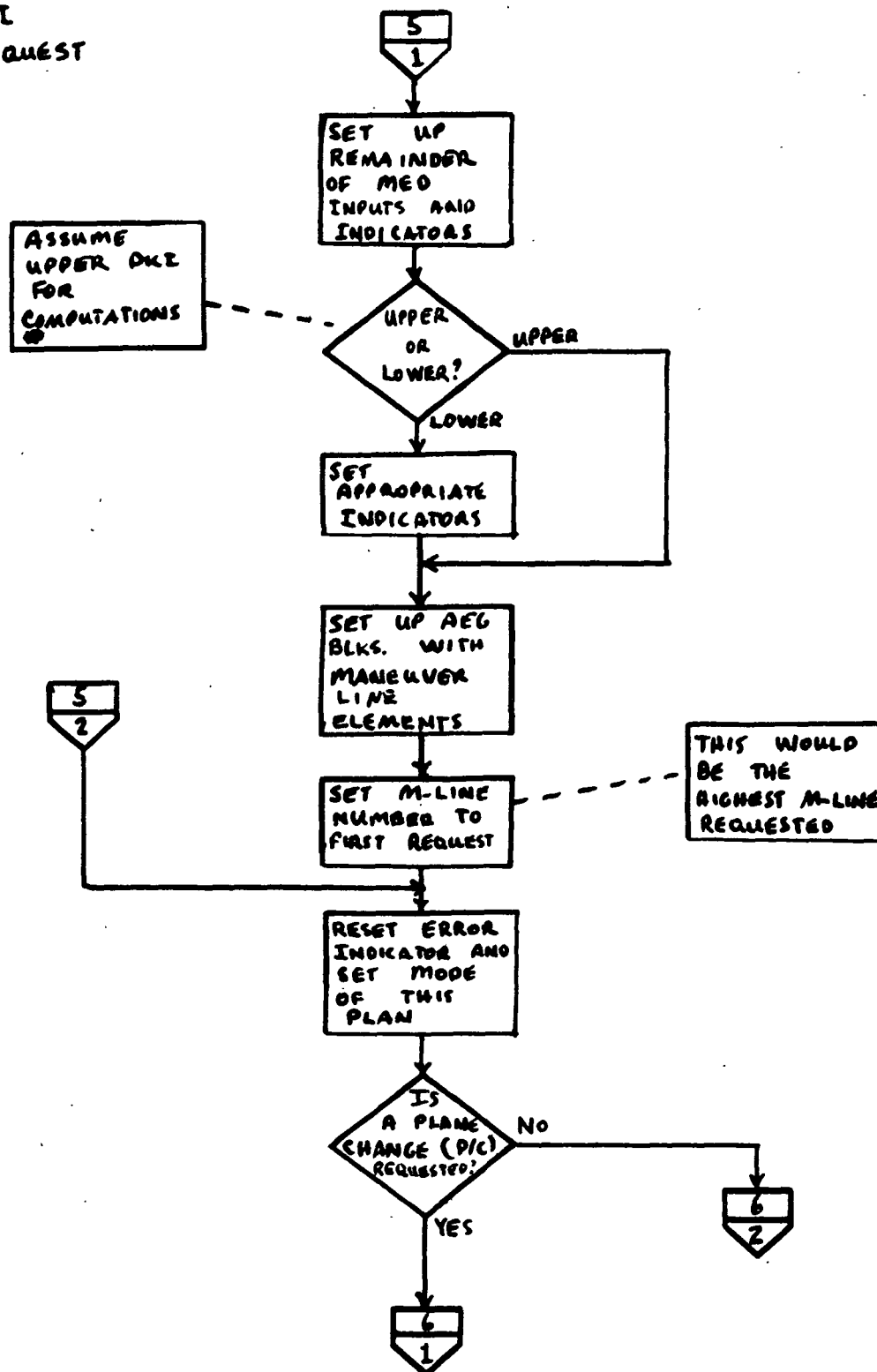
DKI
REQUEST

Figure 1

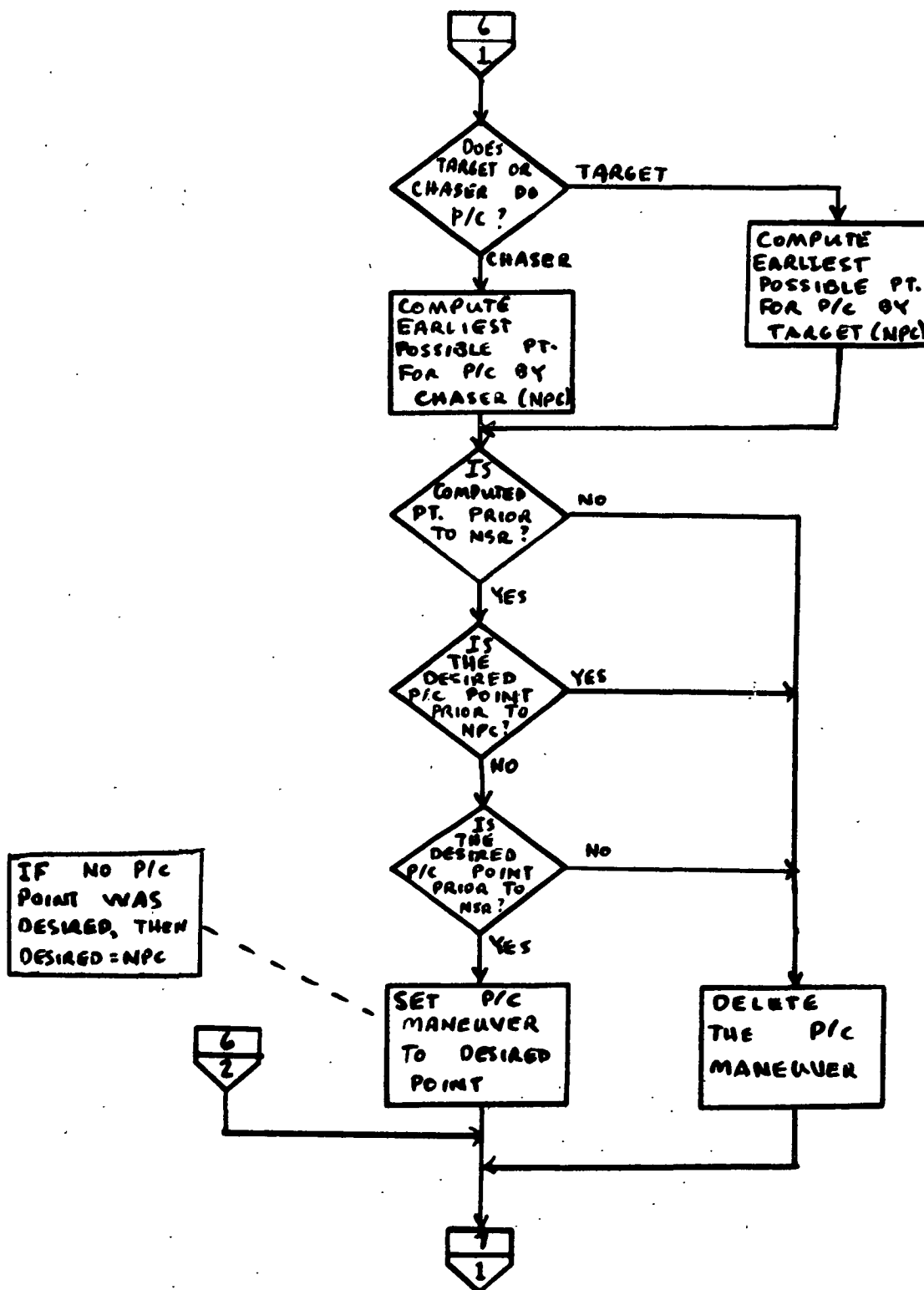


Figure 1

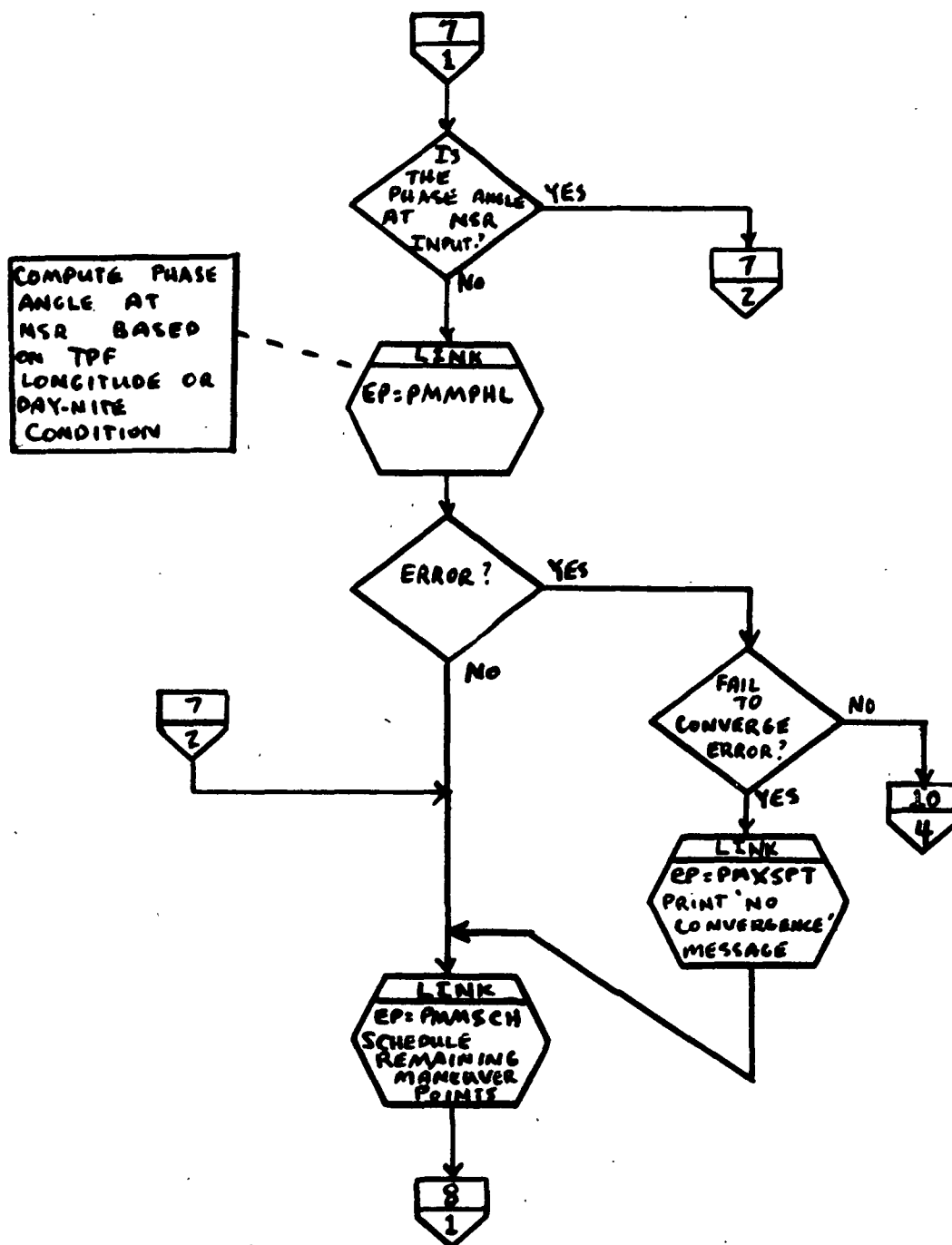


Figure 1

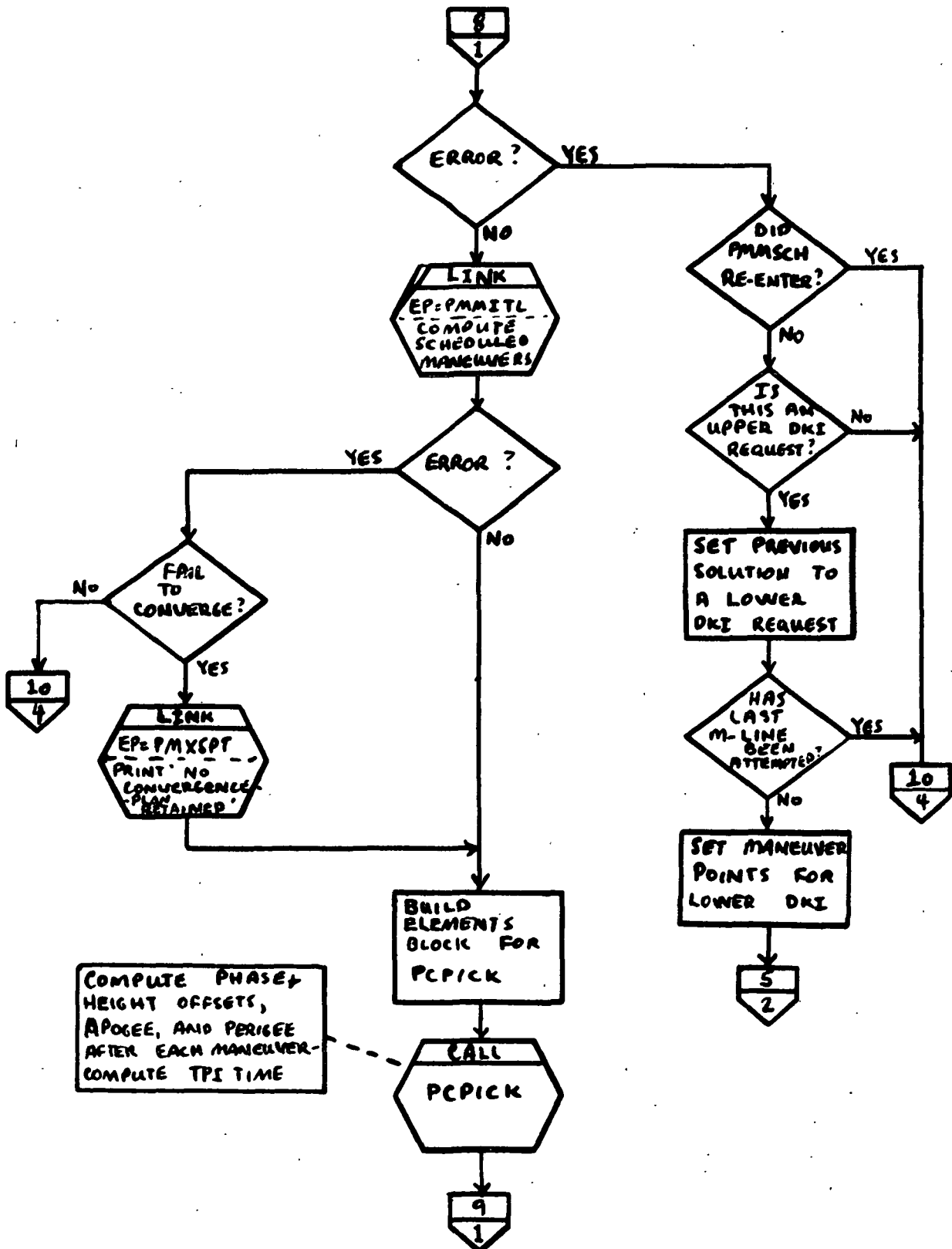


Figure 1

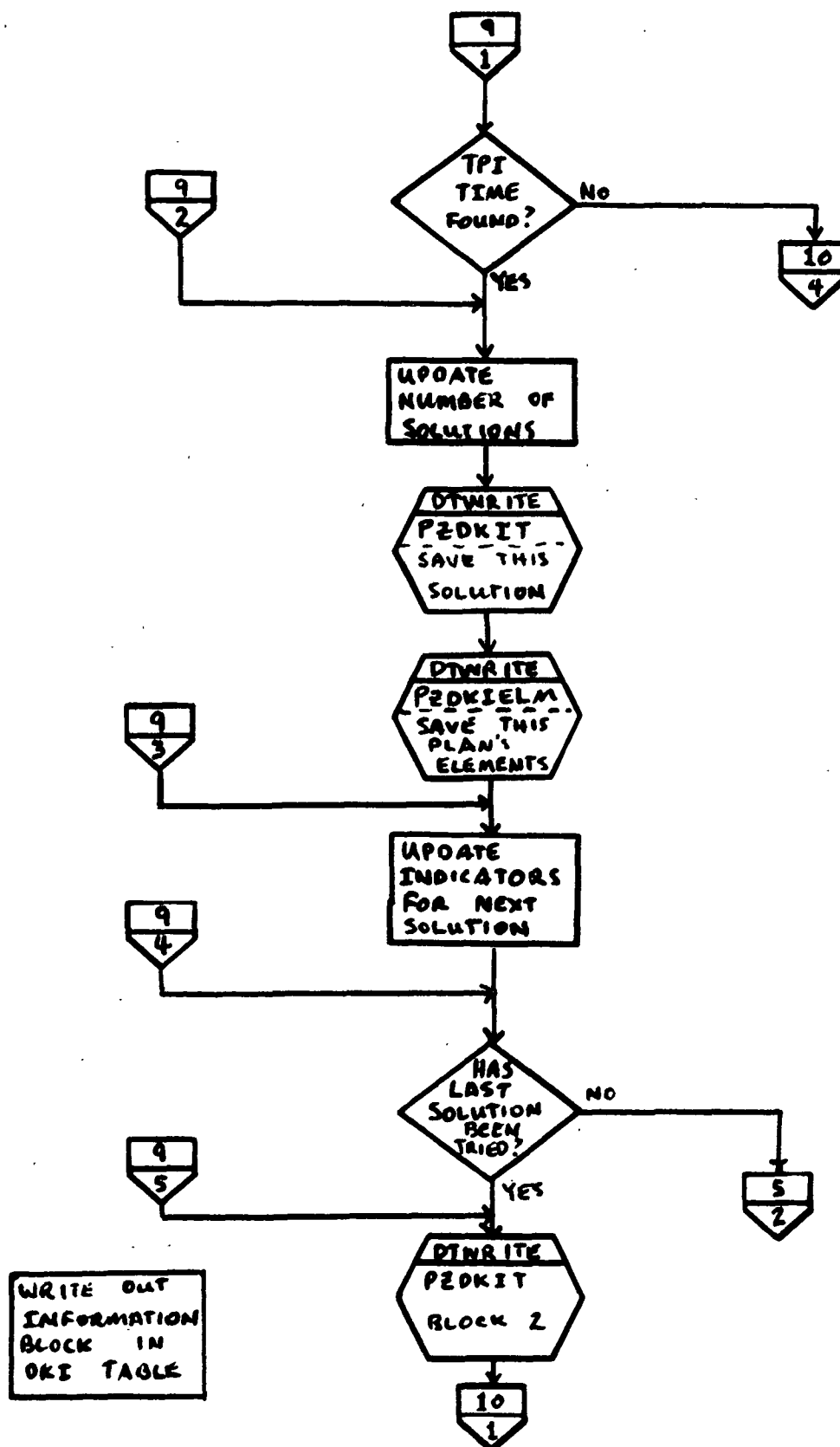


Figure 1

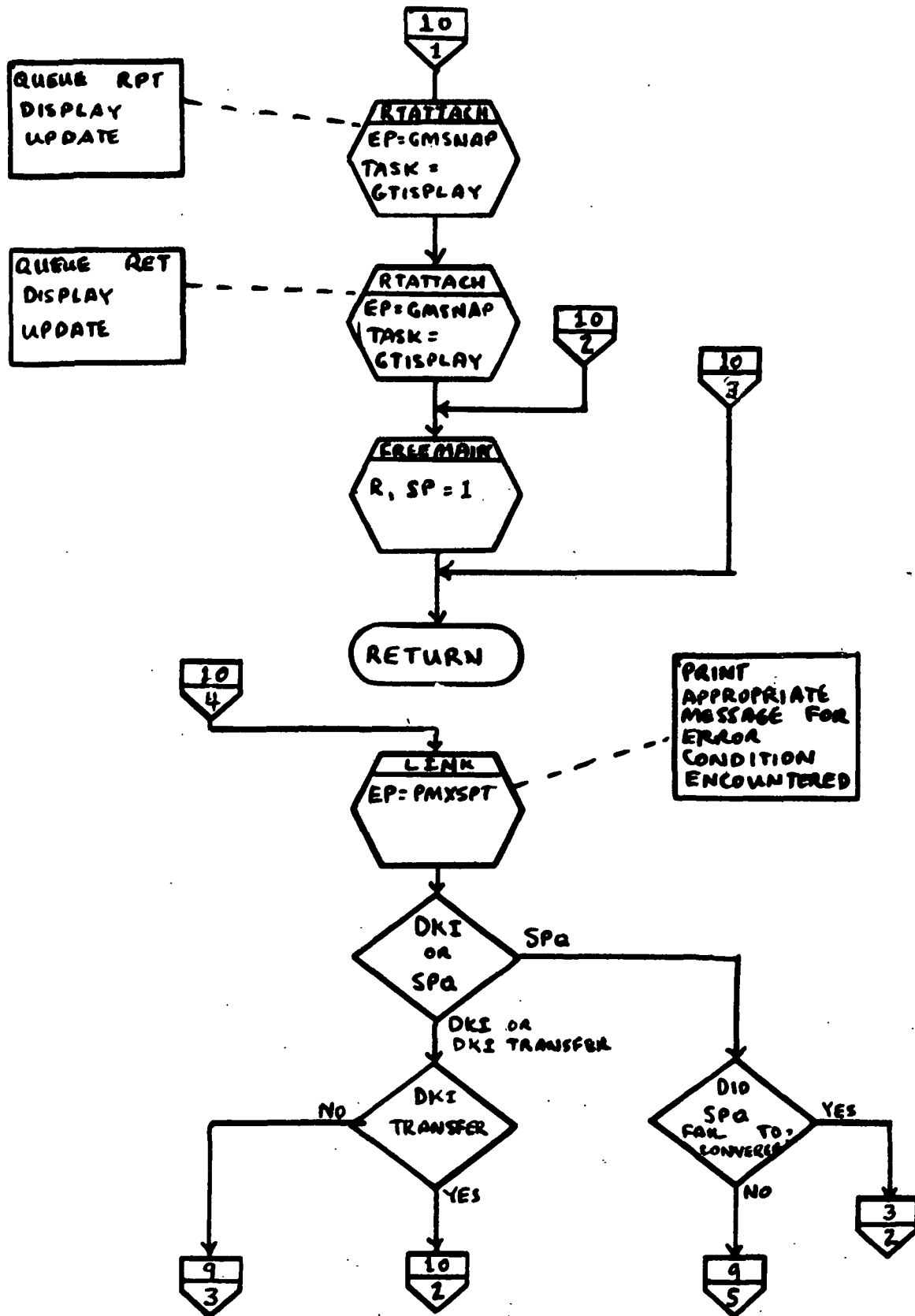


Figure 1

DKI TRANSFER

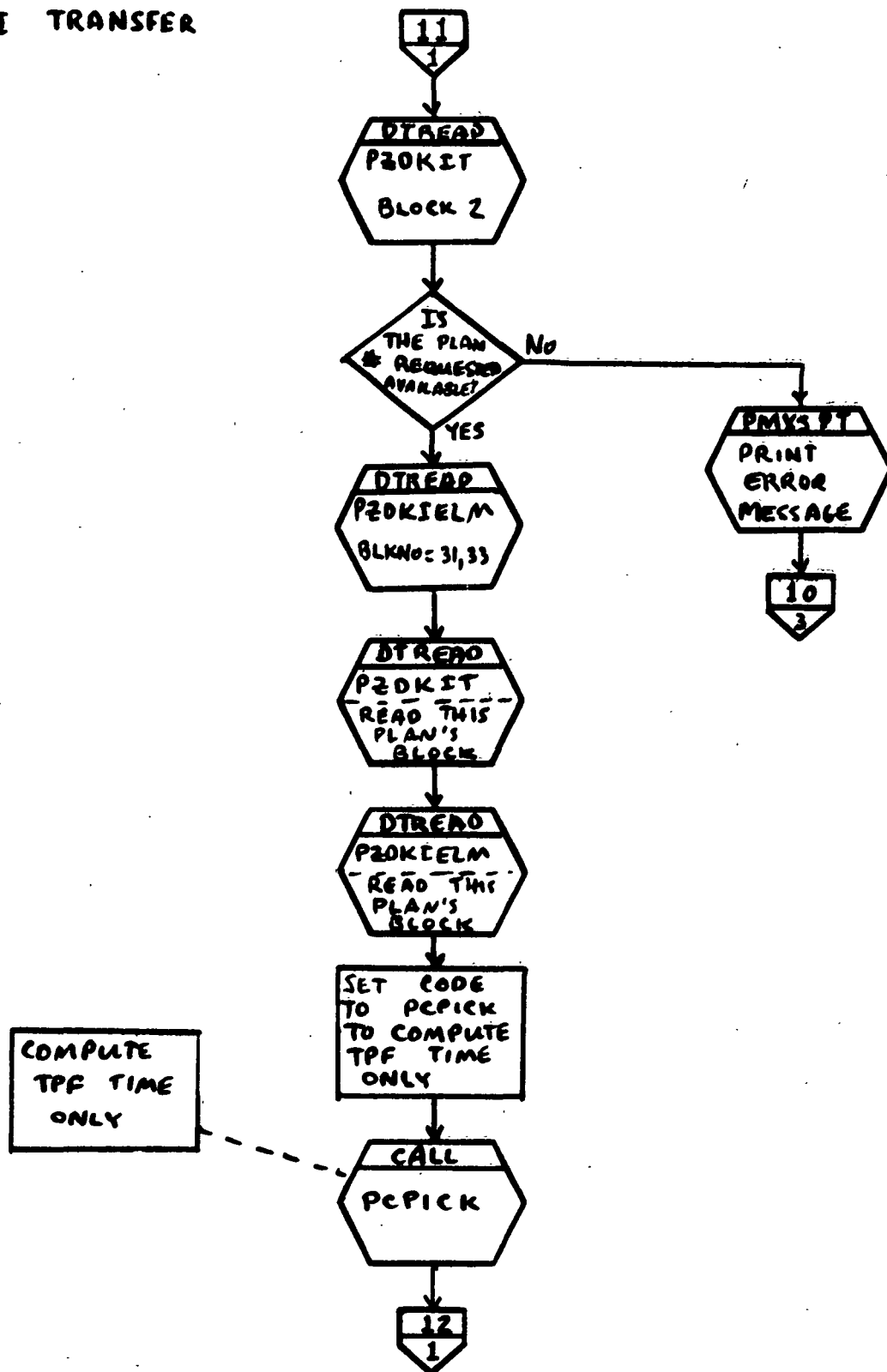


Figure 1

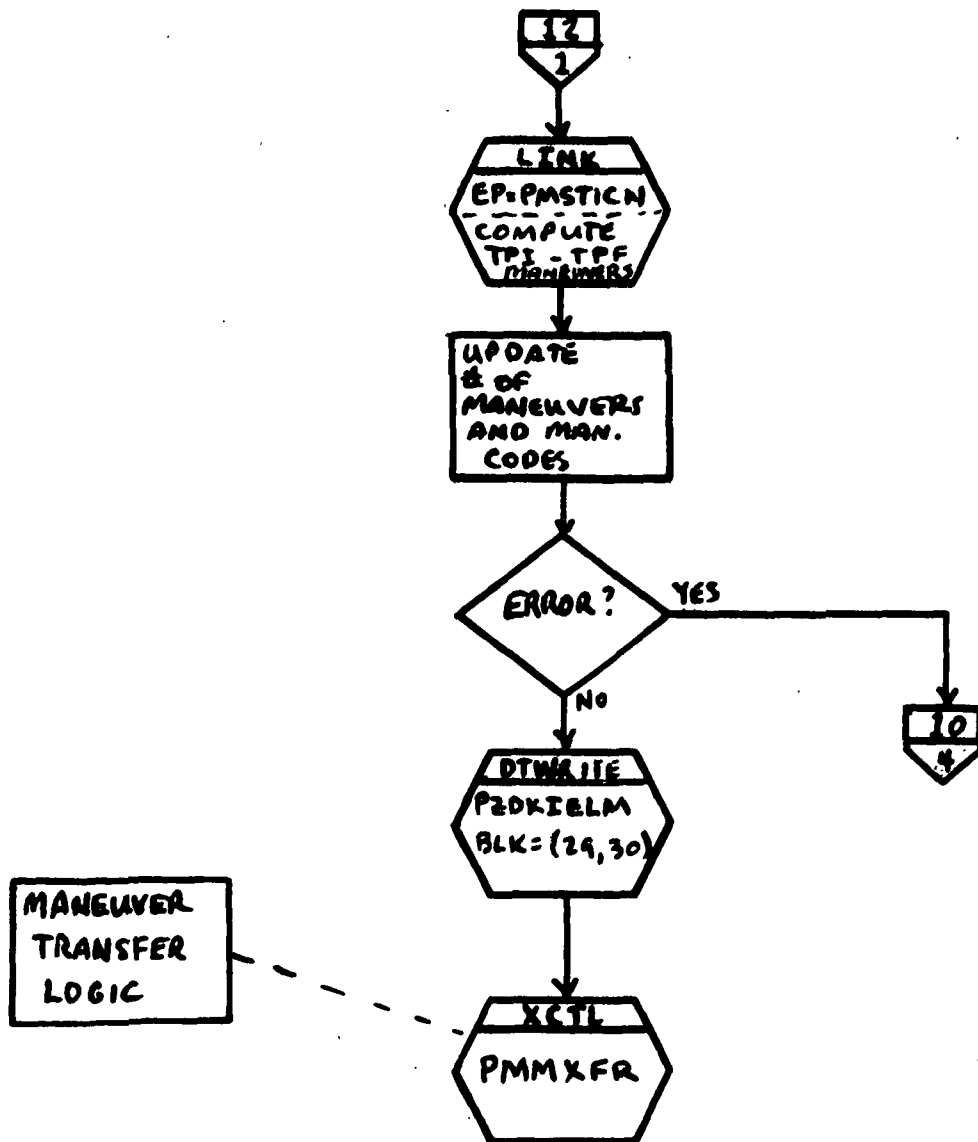


Figure 1

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMDKI-PCPICK
DATE 6/28/68 (EORP)
REV
PAGE 1 of 4PMMDKI-PCPICK
APOGEE, PERIGEE, AND OFFSET DETERMINATIONREFERENCES

1. Programming Group — Rendezvous Planning
2. Programmer — R. Barrett
3. Language — FORTRAN IV
4. Writeup Completion Date — 5/13/68
5. Data Tables Used — None
6. MED's Used — None
7. Flowchart — Figure 1

FUNCTION

The purpose of PCPICK is to compute phase, delta h, apogee and perigee heights after a maneuver or to update the chaser and target vehicles to the time of the TPI maneuver.

USAGE

1. Calling Sequence — CALL PCPICK (DKIZ, EL, QHAP)
2. Inputs — AEG input block for chaser and target (DKIZ) and elements block (EL)
3. Outputs and Exit — delta height, delta theta, apogee, and perigee after each maneuver (QHAP)
4. Other Programs Used — PMMAEG, PCTETR, Apogee/Perigee Determination Routine
5. Data Restrictions and Limits — None
6. Error — AEG error. An error results if PCPICK is unable to get TPI time from PCTETR; error indicator is passed in AEG block, word 56;

METHOD

The basic problem involved in either of the two entries is to get the chaser and target elements after each maneuver. This is done by going through the elements table and initializing the chaser/target elements after each maneuver. If apogee/perigee heights are desired, PCHAPE is called to compute these values. After the last maneuver, TPI is computed, and the AEG blocks are updated to TPI time, if desired. Also an options is provided to compute the target's apogee and perigee height after the last maneuver.

Approval	Approval
<i>James H. [Signature]</i>	<i>A. Amietti</i>

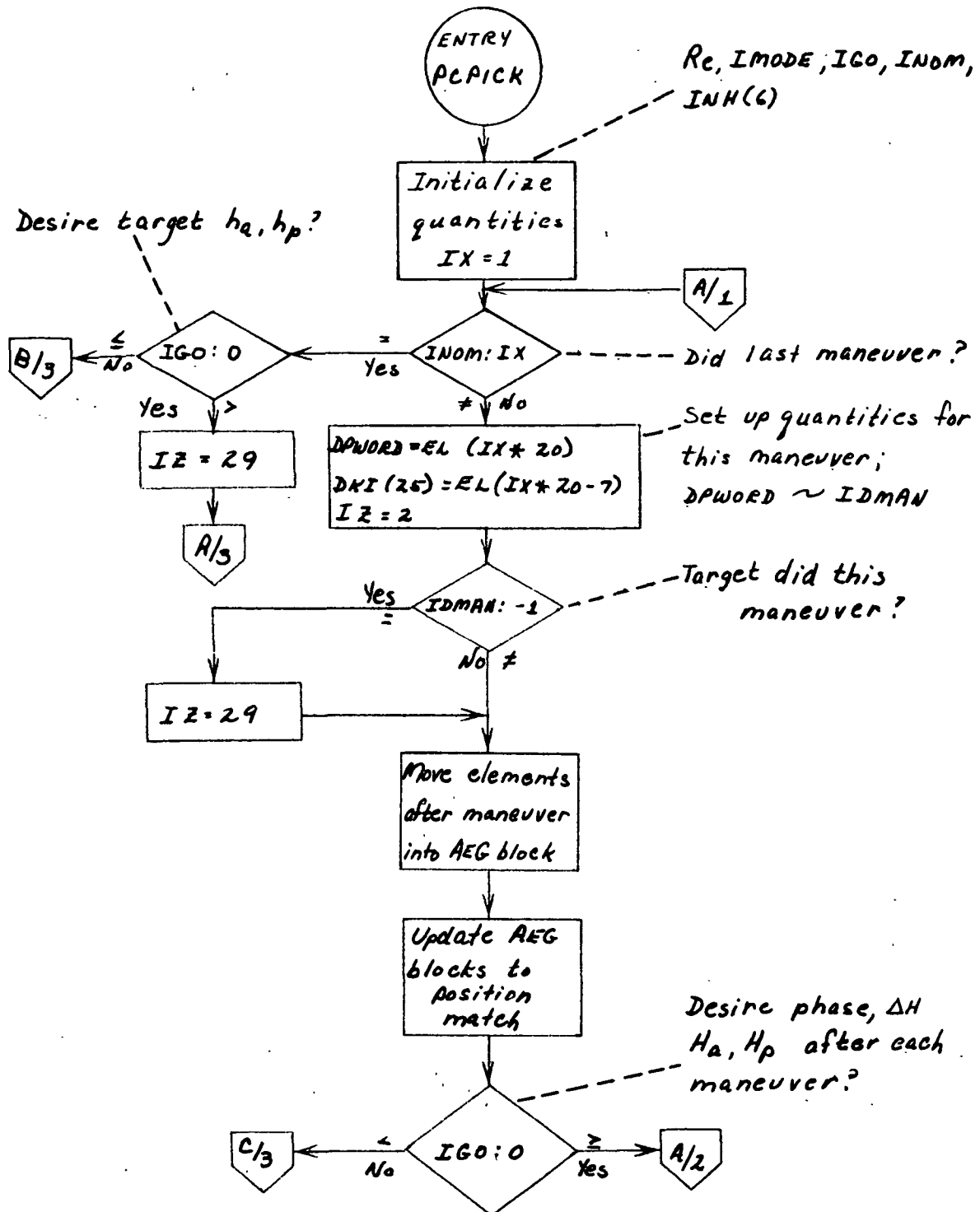


Figure 1

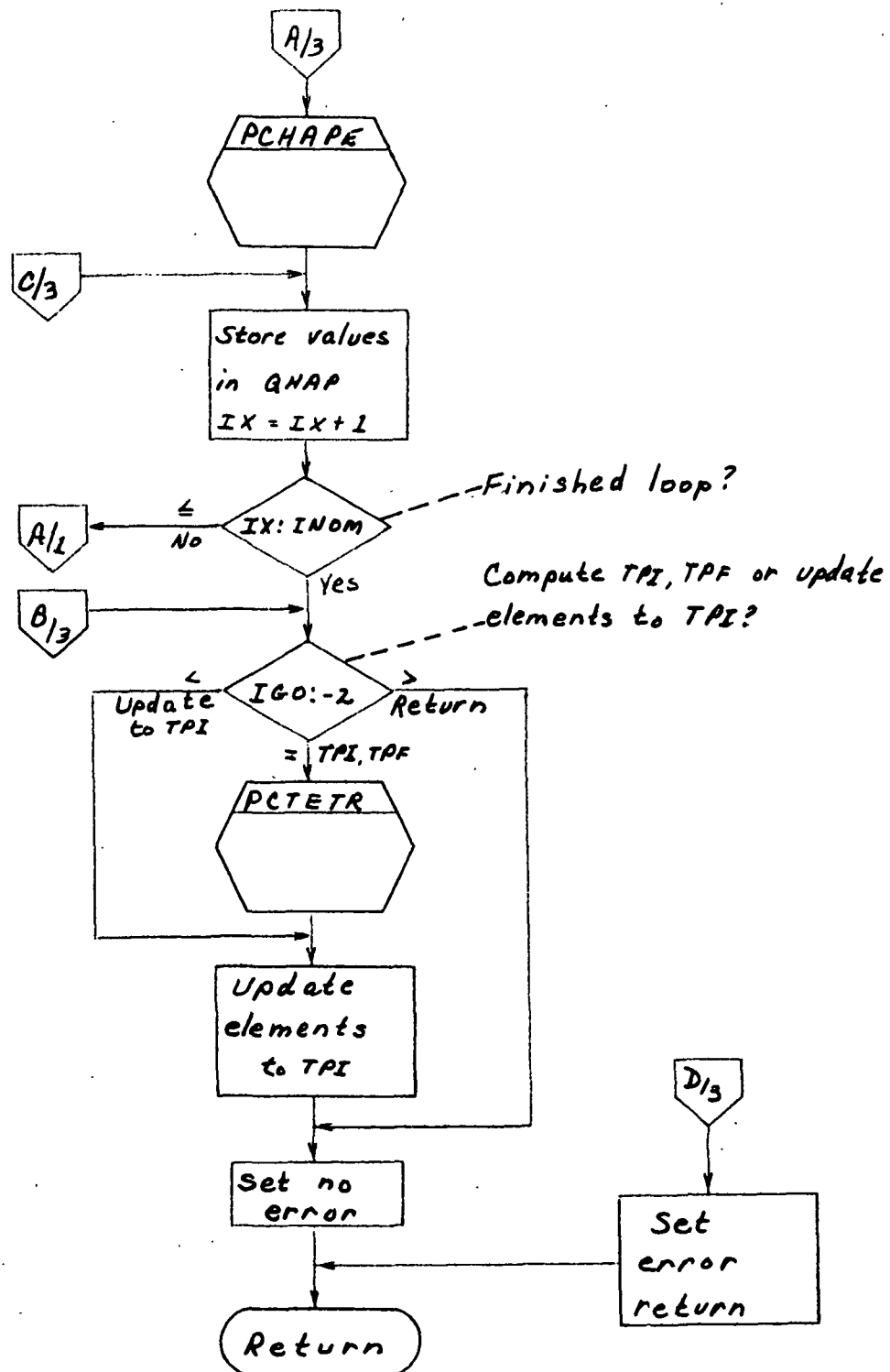


Figure 1

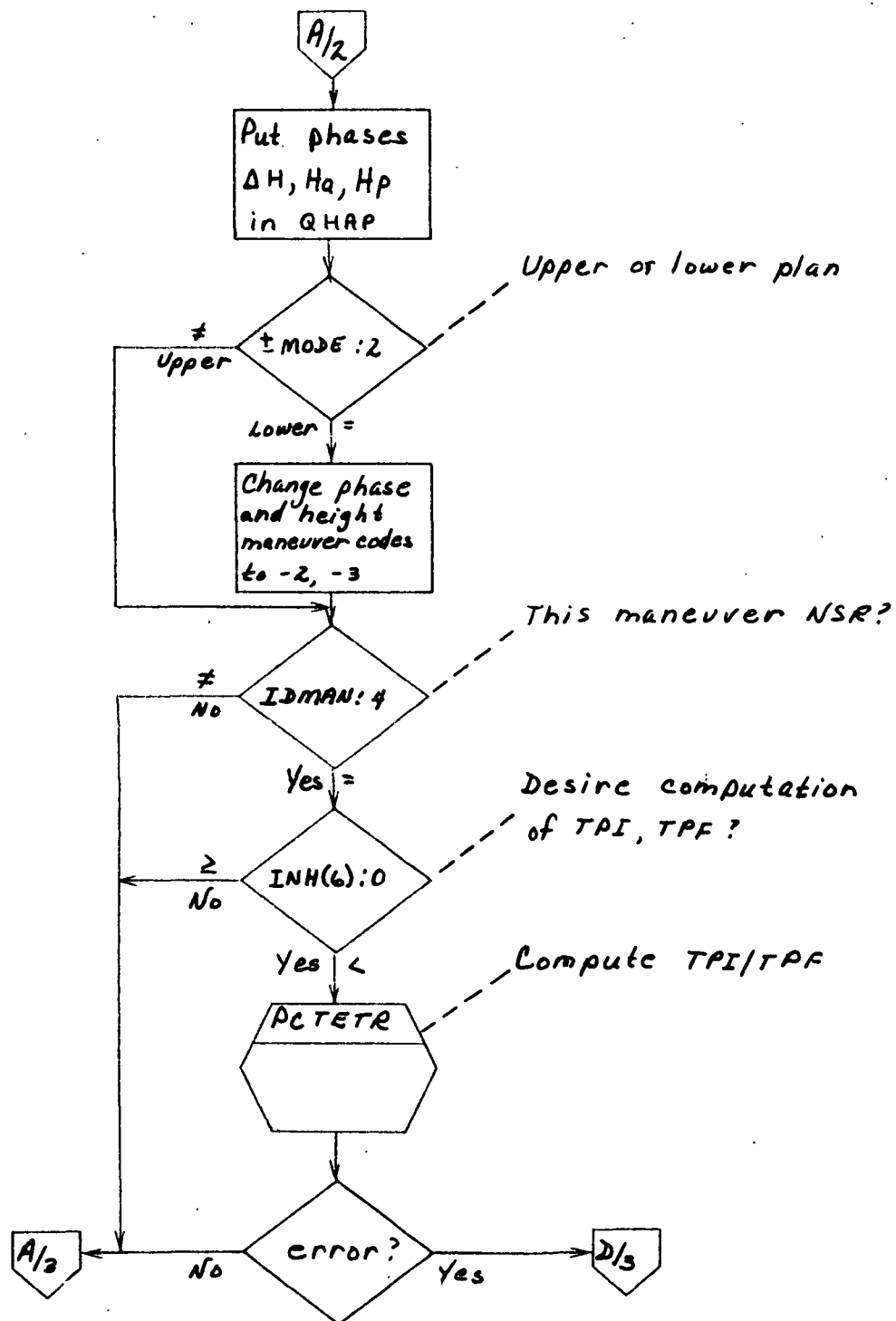


Figure 1

PMMDKI-PCTETR
ELEVATION ANGLE SEARCH SUBROUTINEREFERENCES

1. Programming Group — Rendezvous Planning
2. Programmer — R. Barrett
3. Language — FORTRAN IV
4. Writeup Completion Date — 5/13/68
5. Data Tables Used — None
6. MED's Used — None
7. Flowchart — Figure 1

FUNCTION

PCTETR is designed to compute, from input target and chaser elements, the time of the TPI maneuver based on an input elevation angle and the time of the TPF maneuver based on an input target terminal phase travel angle.

USAGE

1. Calling Sequence — CALL PCTETR (AEGBLK, WT, ESP, TR, TESP)
2. Inputs — AEG input block (AEGBLK), terminal phase travel angle (WT) and desired elevation angle (ESP).
3. Outputs and Exit — Time of arrival at elevation angle (TESP) and time of rendezvous (TR). Exit is by RETURN macro.
4. Other Programs Used — PMMAEG
5. Data Restrictions and Limits — AEGBLK is chaser and target elements at time match, plus AEG header and doubleword 56. First halfword is error exit indicator.
6. Error — An error indicator = -1 if no solution and = 0 if valid solution.

Approval	Approval
<i>James M. Fisher</i>	<i>A. Annette</i>

METHOD

PCTETR computes a phase angle, θ_i , using the target radius (R_T), chaser radius (R_c) and the input desired phase angle (E^*) from the following formulas:

$$\theta_i = \cos^{-1} \left[\frac{-c_2 - \sqrt{c_4}}{2c_1} \right]$$

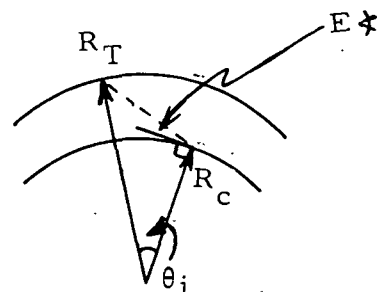
where:

$$c_1 = -(R_T / \cos(E^*))^2$$

$$c_2 = 2R_T R_c$$

$$c_3 = -c_1 - R_T^2 - R_c^2$$

$$c_4 = c_2^2 - 4c_1 c_3$$



A time increment ΔT_i is found from the following:

$$\Delta T_i = (\theta - \theta_i) / \Delta \dot{u}$$

where:

$$\Delta \dot{u} = \dot{L}_c + \dot{g}_c - \dot{L}_T - \dot{g}_T = \text{catchup rate}$$

θ = actual phase as returned from the AEG.

If $\Delta T_i \leq \sigma t$ is a convergence tolerance, then $Q \approx Q_i$; hence chase and target elements are TPI. If $\Delta T_i > \sigma t$, then the chaser and target elements are updated to $T_c + \Delta T$, and the process is repeated.

Error conditions may occur for two reasons. The TPI time was not found after a maximum number of iterations, or it is determined that the chaser and target will never achieve a configuration such that the desired elevation angle (E^*) requirements will be achieved. Therefore the triangle using R_c , R_T and E^* has no solution.

TPF time is computed from the formula below:

$$\text{TPF} = \text{TPI} + \omega t / (\dot{L}_T + \dot{g}_T)$$

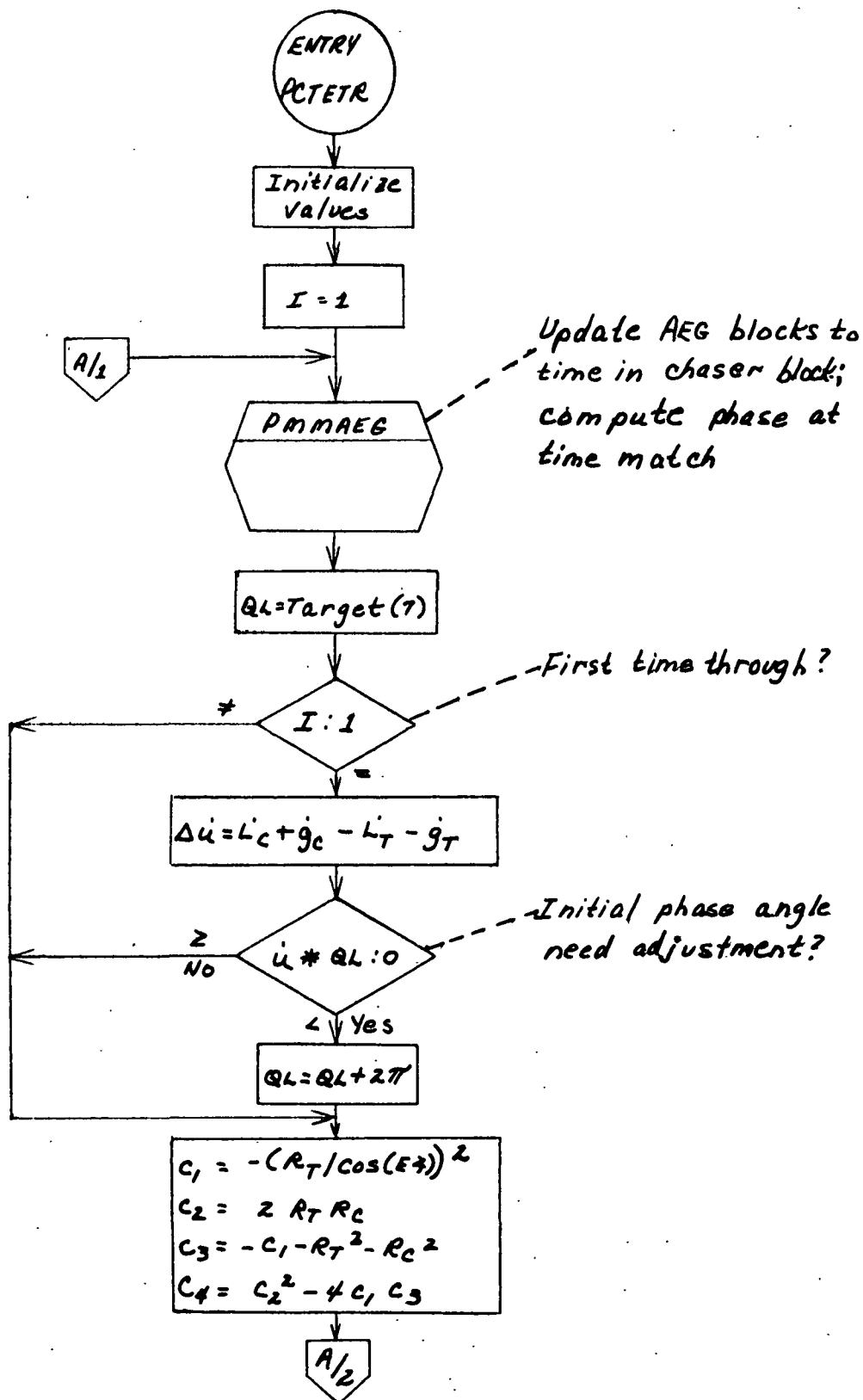


Figure 1

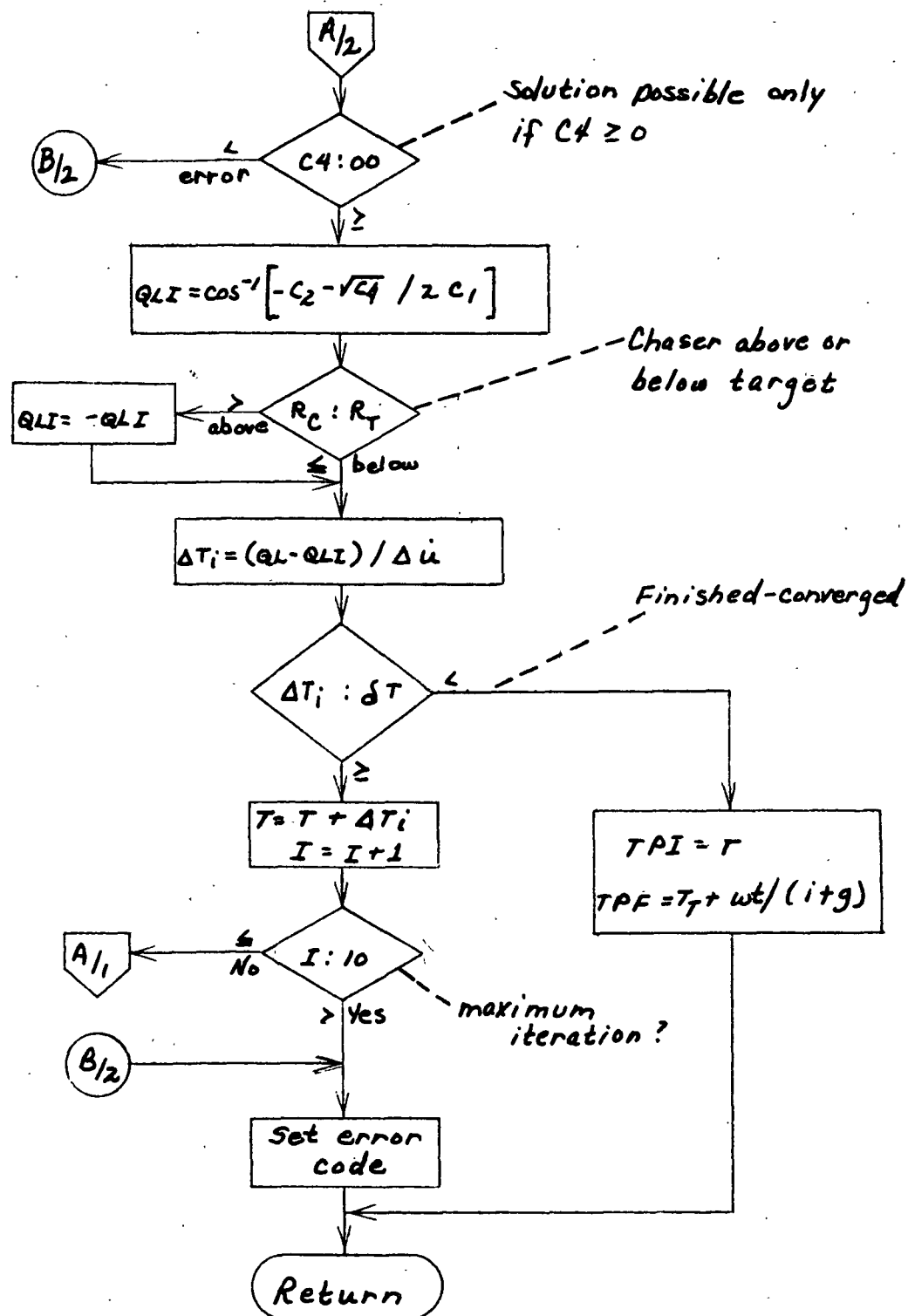


Figure 1

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS-AS-200 Mission Planning

2. PMMDMT
DATE 6/22/67 (206)
REV
PAGE 1 of 20

- a. Symbol - PMMDMT
- b. Program Name - Detailed Maneuver Table Math
- c. Programming Group - Maneuver Control
- d. Programmer(s) - R. Taylor
- e. Language - Assembler
- f. Data Tables Used - PZMPTLEM, PZDMTLEM, EZETVMED ✓
- g. MED's Used - None
- h. Writeup Completion Date - 3/17/67
- i. System Parameters - Table PMMDMT-1
- j. Flowchart — Figure PMMDMT-1

FUNCTION

This program uses the auxiliary output table from the LM Runge-Kutta integrator to supply finite burn and post maneuver quantities for the Mission Plan Table and prints a restart vector after the last executed maneuver.

USAGE

- a. Calling Sequence - LINK EP=PMMDMT
- b. Inputs - Address of auxiliary output table, vehicle code, maneuver number, scratch area
- c. Outputs and Exit - Updated Mission Plan Table, updated PZDMTLEM, and restart vector
- d. Other Programs Used - DLOG, GLMDPD, DSQRT, GIMAPD, GIMIKC, DSIN, GLQATN, DARSIN, DCOS, PMMAEG, PMXSPT
- e. Data Restrictions and Limits - None
- f. Error return from PMMAEG and error return from GIMAPD

METHOD

For a queue to update a maneuver block in the Mission Plan Table (MPT), the burn initiate and burn-out vectors, ΔV of the maneuver, Δt of burn, cutoff time, ΔV along x-body axis with ullage and tail-off, and ΔV along x-body axis including ullage are stored in the indicated block; and end-time of the maneuver is stored in the header block. If no weight change MED has been executed, the weight after the maneuver; ΔV remaining for the appropriate thruster; APS, DPS, and RCS fuels remaining; and LM ascent and descent dry weights are computed.

For the guided maneuvers--descent orbit injection guidance, powered landing guidance, and ascent guidance-- ΔV of tail-off, pitch, yaw, roll, pitch gimbal, and yaw gimbal values are transferred from the integrator output

Approval	Approval
<i>R. Taylor</i>	<i>A. E. James</i>

table. The velocity components ($\bar{V}_g$) are converted to the platform coordinate system. The values for P_H , Y_H , R_H , V_F , V_D , and V_N are computed using pitch, yaw, roll, the reference matrix, burn initiate vectors, and $\bar{V}_g$. In this case, $\bar{\Delta V} = \bar{V}_g$.

$$\begin{bmatrix} \hat{X}_N \\ \hat{Y}_N \\ \hat{Z}_N \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos Y & \sin Y \\ 0 & -\sin Y & \cos Y \end{bmatrix} \begin{bmatrix} \cos R & \sin R & 0 \\ -\sin R & \cos R & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos P & 0 & -\sin P \\ 0 & 1 & 0 \\ \sin P & 0 & \cos P \end{bmatrix} \begin{bmatrix} \hat{X}_P \\ \hat{Y}_P \\ \hat{Z}_P \end{bmatrix}$$

The local vertical local horizontal (LVLH) platform coordinate axes are defined as:

$$\hat{Z}_{P_{HV}} = -\frac{\bar{r}}{r}, \quad \hat{Y}_{P_{HV}} = \frac{-(\bar{r} \times \bar{v})}{|\bar{r} \times \bar{v}|}, \quad \hat{X}_{P_{HV}} = \hat{Y}_{P_{HV}} \times \hat{Z}_{P_{HV}}$$

where: $\bar{r}$ = position vector at burn initiate
 $\bar{v}$ = velocity vector at burn initiate

The final computations are:

$$P_H = \sin^{-1} \left[\frac{\hat{Z}_{P_{HV}} \cdot \hat{X}_N}{|\hat{Z}_{P_{HV}}| |\hat{X}_N|} \right]$$

$$Y_H = \tan^{-1} \left[\frac{\hat{Y}_{P_{HV}} \cdot \hat{X}_N}{\hat{X}_{P_{HV}} \cdot \hat{X}_N} \right]$$

$$R_H = \tan^{-1} \left[\frac{\hat{Z}_{P_{HV}} \cdot \hat{Y}_N}{\hat{Z}_{P_{HV}} \cdot \hat{Z}_N} \right]$$

If P_H is equal to $\frac{1}{2}$, the equations for Y_H and R_H are:

$$R_H = 0.0$$

$$Y_H = \tan^{-1} \left[\frac{\left(\hat{Y}_{P_{HV}} \times \hat{Y}_N \right) \cdot \hat{Z}_{P_{HV}}}{\hat{Y}_{P_{HV}} \cdot \hat{Y}_N} \right]$$

The V_F , V_D , V_N values are computed as:

$$V_F = \hat{X}_{P_{HV}} \cdot \Delta \bar{V}$$

$$V_D = \hat{Z}_{P_{HV}} \cdot \Delta \bar{V}$$

$$V_N = \hat{Y}_{P_{HV}} \cdot \Delta \bar{V}$$

For inertial maneuvers using Reaction Control System (RCS) -X thrusters with ΔV option, only the total ΔV is taken from the integrator table. For $\Delta+$ option, the ΔV -body values are set equal to ΔV of the maneuver and new values for V_{g_x} , V_{g_y} , V_{g_z} are found as follows:

$$C = \frac{\Delta V_{MAN(NI)}}{\Delta V_{MAN(MPT)}} -$$

$$V_{g_x} = V_{g_x(MPT)} * C$$

$$V_{g_y} = V_{g_y(MPT)} * C$$

$$V_{g_z} = V_{g_z(MPT)} * C$$

For both options, the velocity increment vector $\Delta \bar{V}$ is:

$$\Delta \bar{V} = V_{g_x} * \hat{X}_P + V_{g_y} * \hat{Y}_P + V_{g_z} * \hat{Z}_P$$

Computations of P_H , Y_H , R_H , V_F , V_D , and V_N then are as from one.

Post maneuver quantities consist of the height, latitude, and longitude of burn initiate, longitude of the next ascending node; apogee and perigee heights, latitudes, longitudes, and times.

Burn initiate quantities are determined by the position vector at burn initiate.

$$r = \sqrt{X^2 + Y^2 + Z^2}$$

$$H = r - R_e$$

$$\phi_c = \sin^{-1} \left(\frac{Z}{r} \right)$$

$$\lambda_{ef} = \tan^{-1} \left(\frac{Y}{X} \right) - (\lambda_o + \omega_e T)$$

Where: $\bar{r}$ is the position vector

X, Y, Z components of $\bar{r}$

H height above spherical Earth

R_E radius of spherical Earth

ϕ_c geocentric latitude $-\pi/2 < \phi_c \leq \pi/2$

λ_{ef} Earth fixed longitude $0 \leq \lambda_{ef} < 2\pi$

λ_o inertial longitude of Greenwich at $T=0$

ω_e Earth rotational rate

T time

The longitude of the upcoming ascending node is based on Keplerian elements at that point found by converting the vector after the maneuver and using the Analytical Ephemeris Generator (AEG) to locate the node.

$$\lambda_{ef} = h - (\lambda_o + \omega_e T) \quad 0 \leq \lambda_{ef} < 2\pi$$

Where h is the inertial argument of the ascending node.

Apogee and perigee quantities are provided by GIMAPD.



The final queue to this load module results in the trajectory update number being set positive; the output table for the Detailed Maneuver Table (DMT) display, PZDMTLEM, being updated; and the restart vector at burn-out of the first non-executed maneuver being printed on line.

Other computations shown on the flowchart are for two-vehicle capability and will not be used in AS-206.



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMDMT
 DATE 6/22/67 (206)
 REV
 PAGE 6 of 20

Table PMMDMT-1. System Parameters

Symbol in Program	Symbol In Master Tape	Type *	Units	Comments
LAMBDAO	MCLAMD	D	radians	Lambda Naught
OMEGAE	MCERTS	D	rad./hr..	Earth Rotational Rate
RE	MCECAP	D	Er.	Radius of Earth
TWOPI	MCMTPI	D	radians	2π
SPAPS	MCTASI	D	hr..	Specific Impulse A_e
SPDPS	MCTDSI	D	hr..	Specific Impulse DPS
ACCG	MCEACG	D	(Er./hr.) ²	Acceleration of Gravity
SPRCS	MCTRII	D	hr..	Specific Impulse RCS
EPSILON	MCMDMT	D	-	Epsilon
HAFPI	MCMHPI	D	radians	$\pi/2$

*Type
 H - Fixed-point half word
 F - Fixed-point full word
 E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal

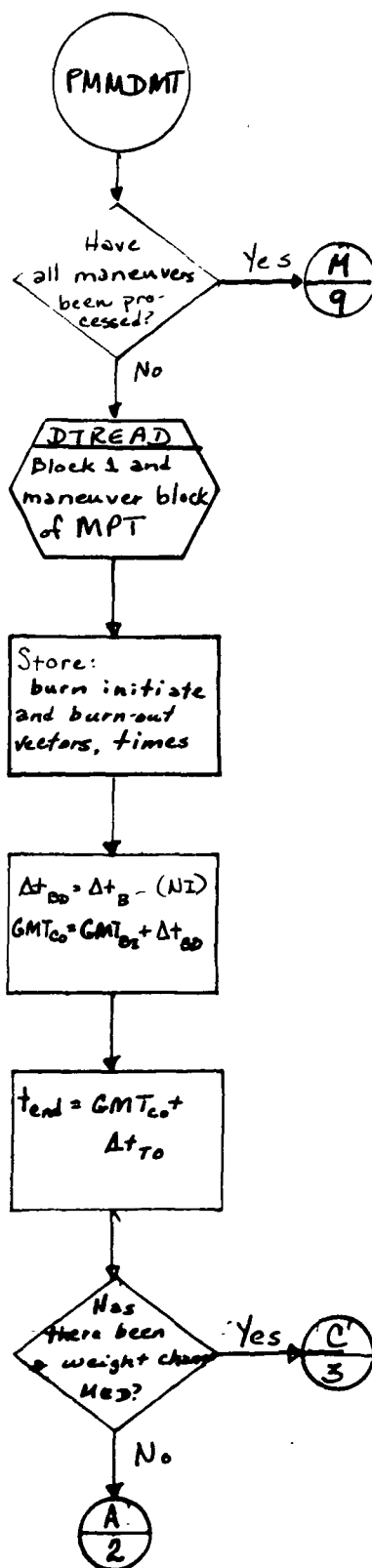


Figure PMMDMT-1. Flowchart

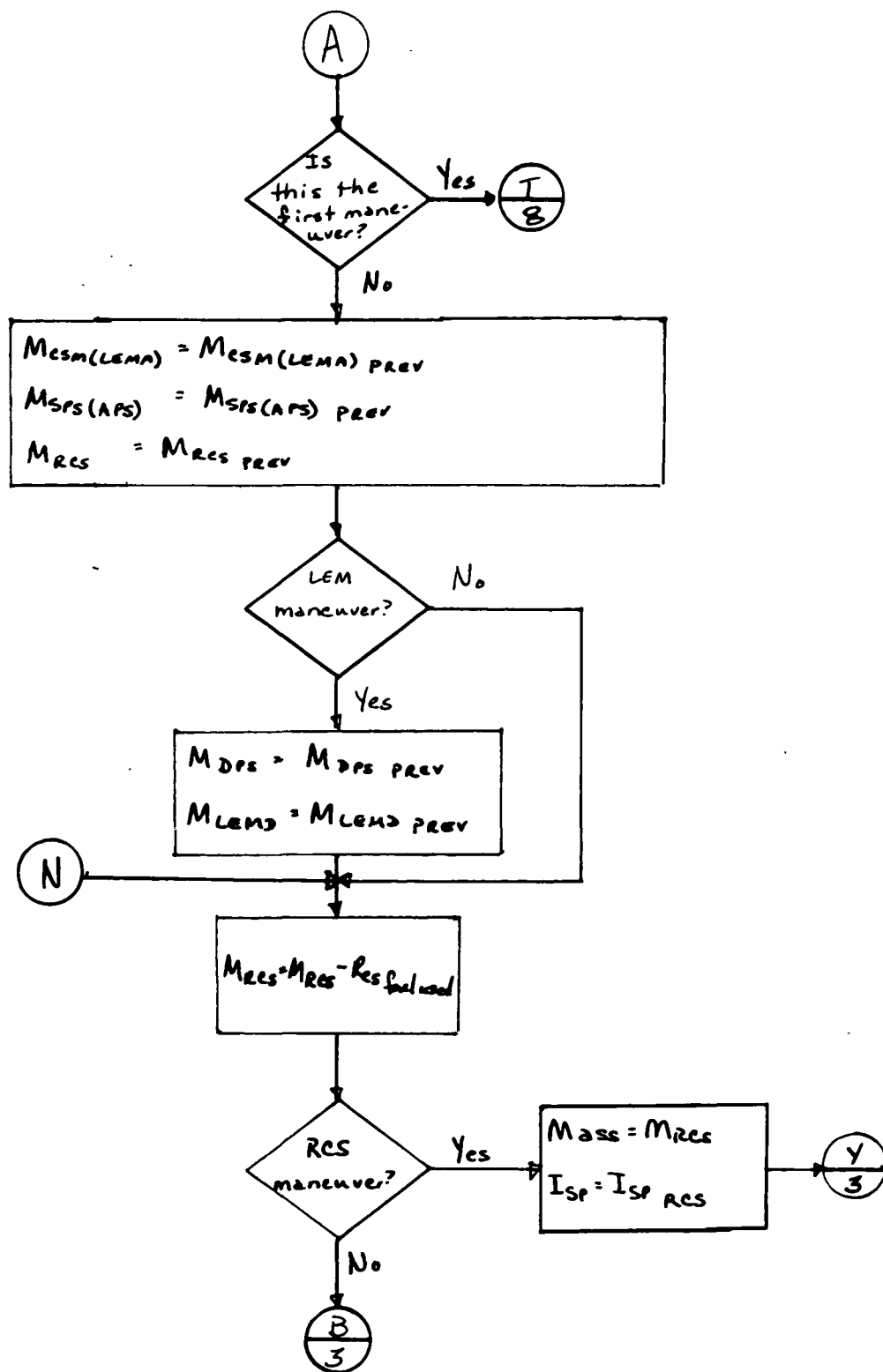


Figure PMMDMT-1. Flowchart

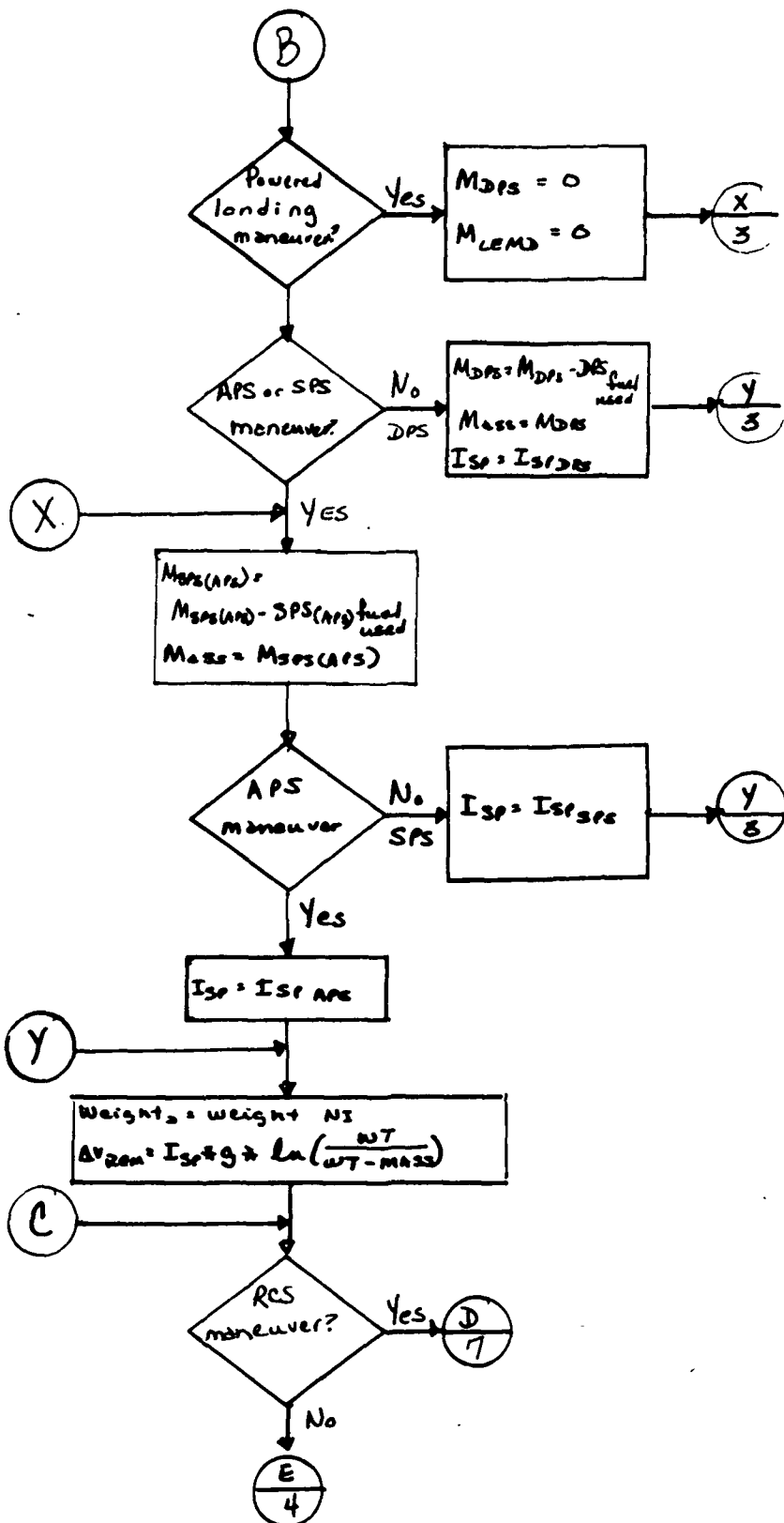


Figure PMMDMT-1. Flowchart

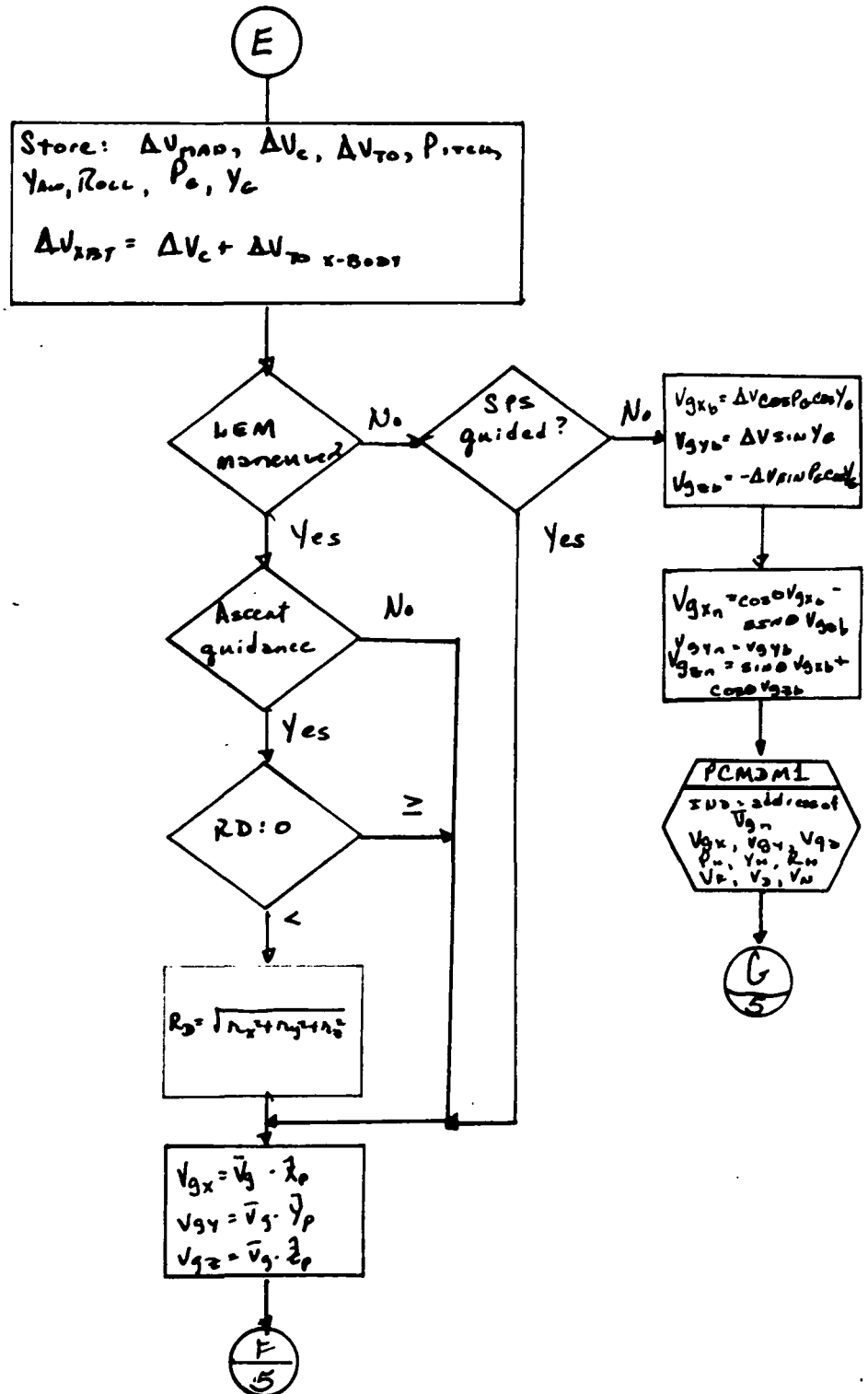


Figure PMMDMT-1. Flowchart

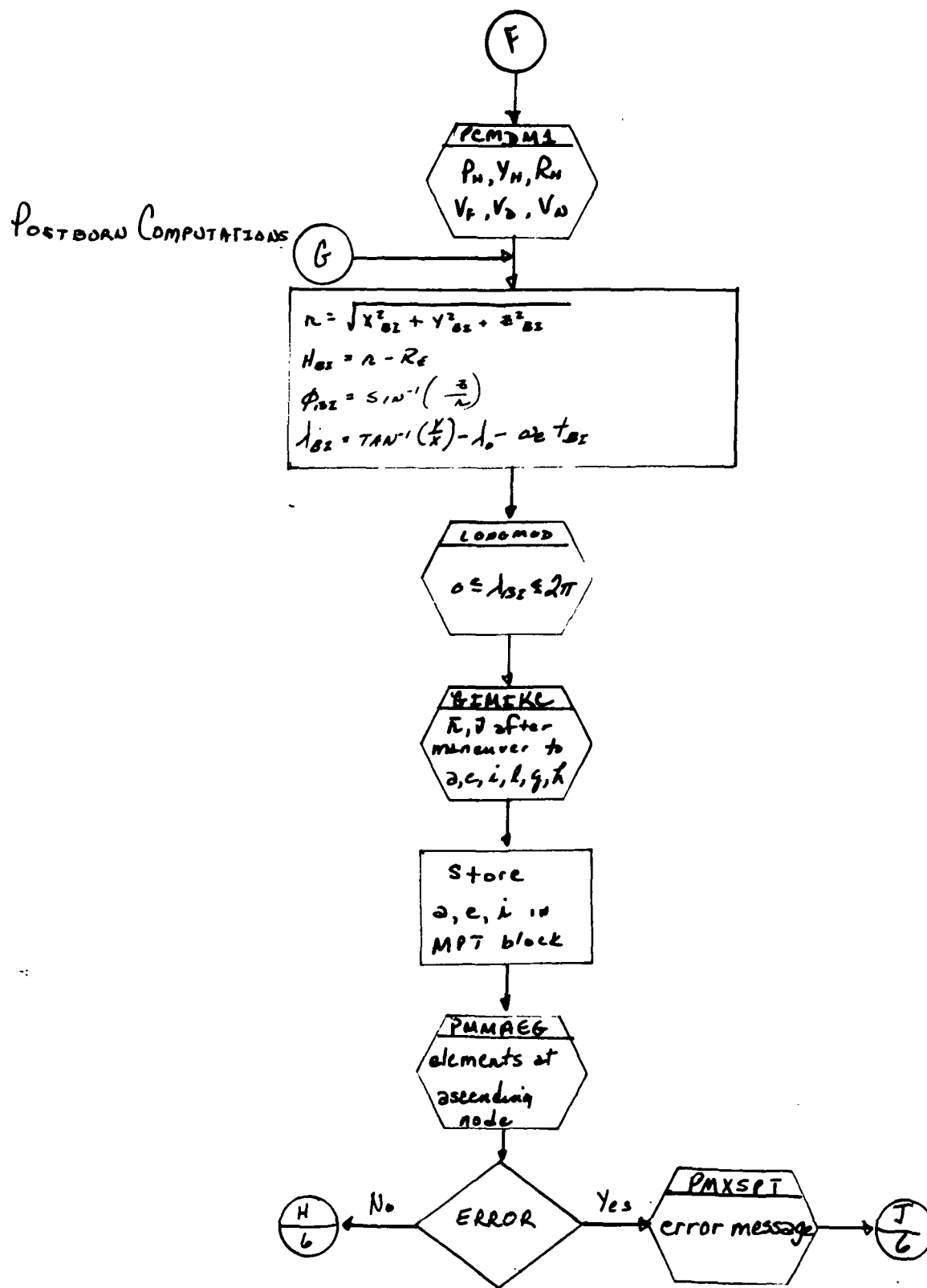


Figure PMMDMT-1. Flowchart

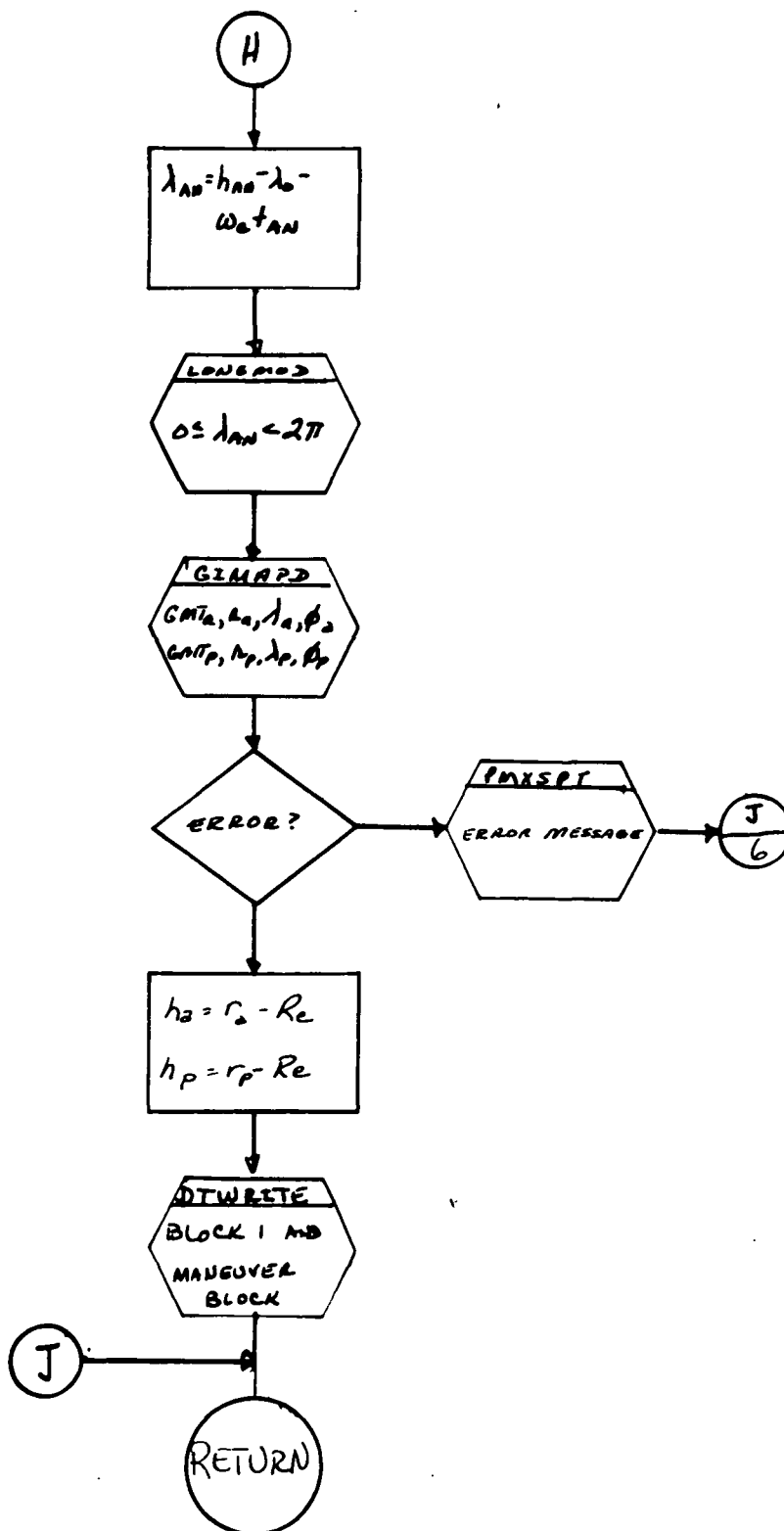


Figure PMMDMT-1. Flowchart

RCS maneuvers

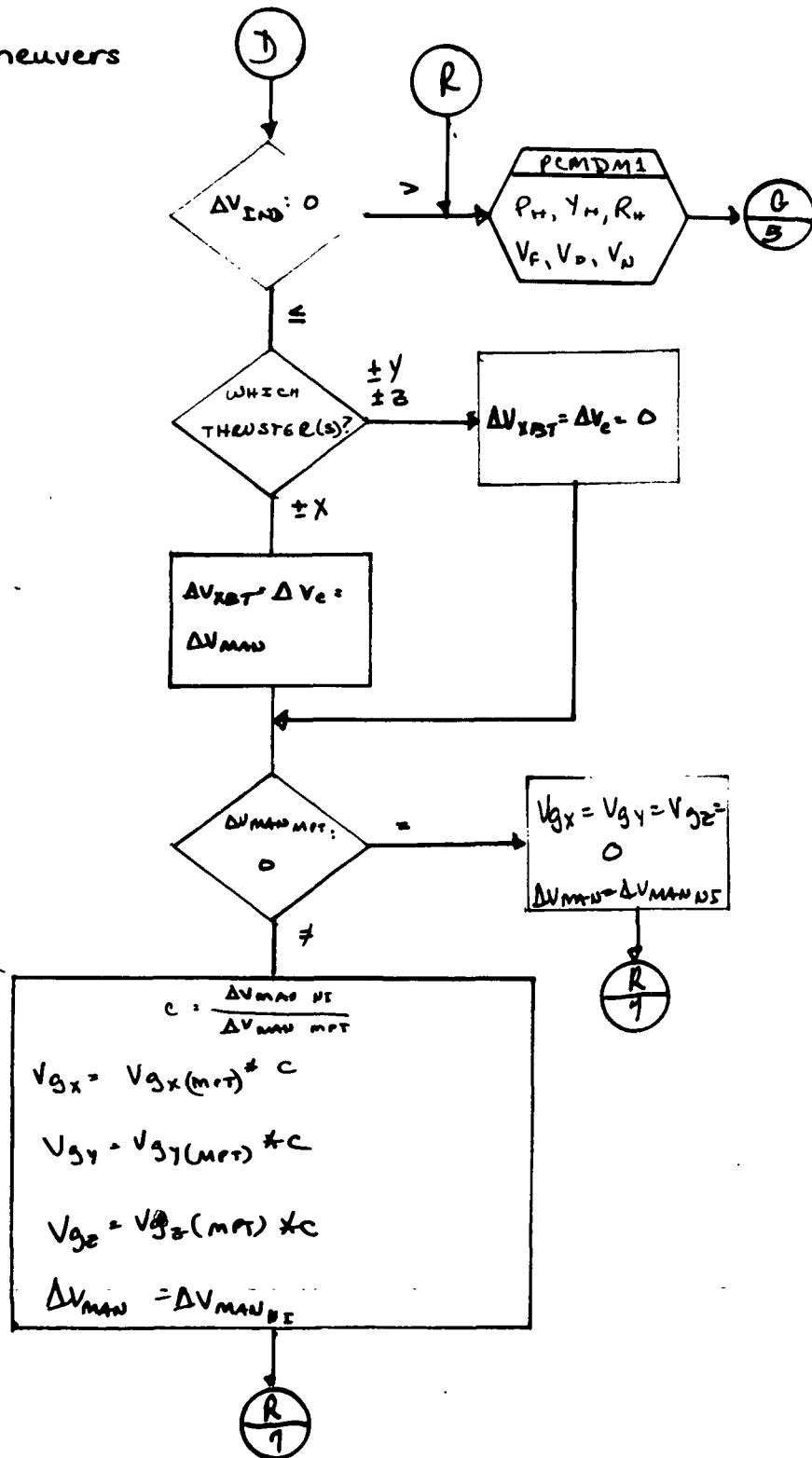


Figure PMMDMT-1. Flowchart

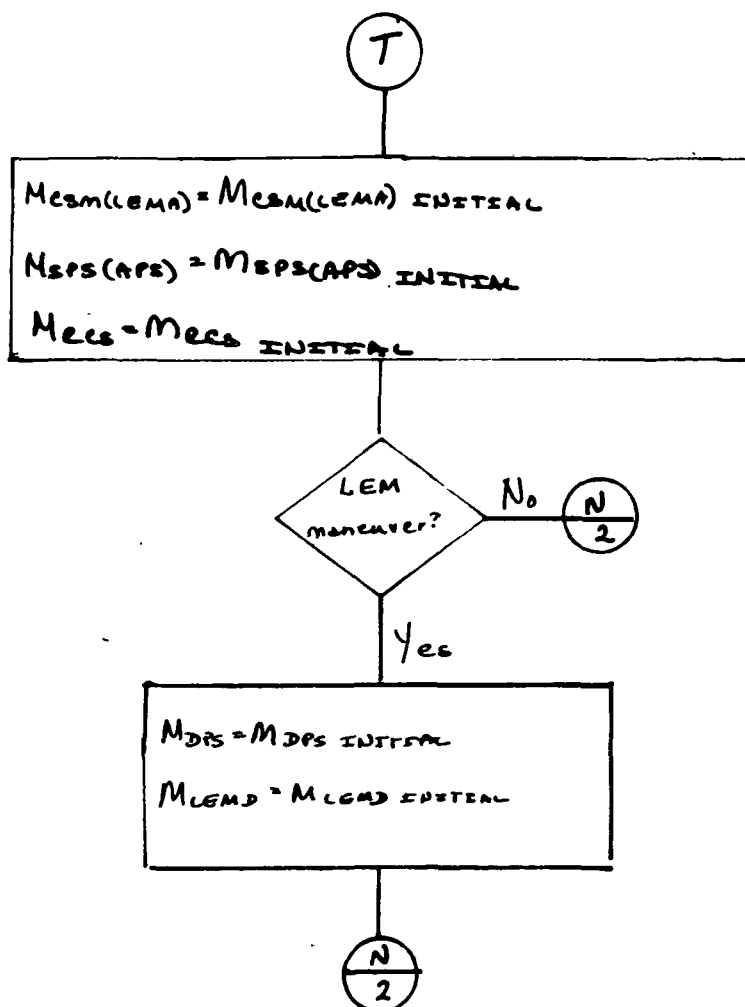


Figure PMMDMT-1. Flowchart

(Sheet 8 of 14)

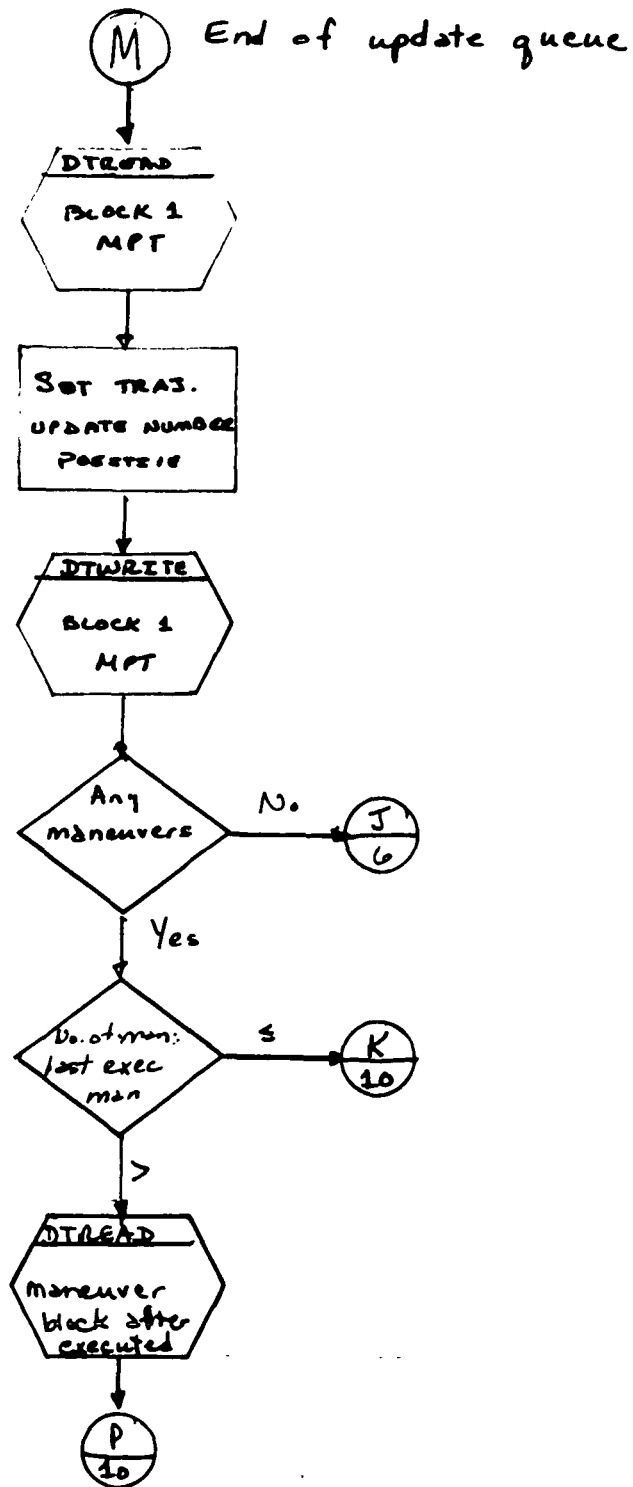


Figure PMMDMT-1. Flowchart

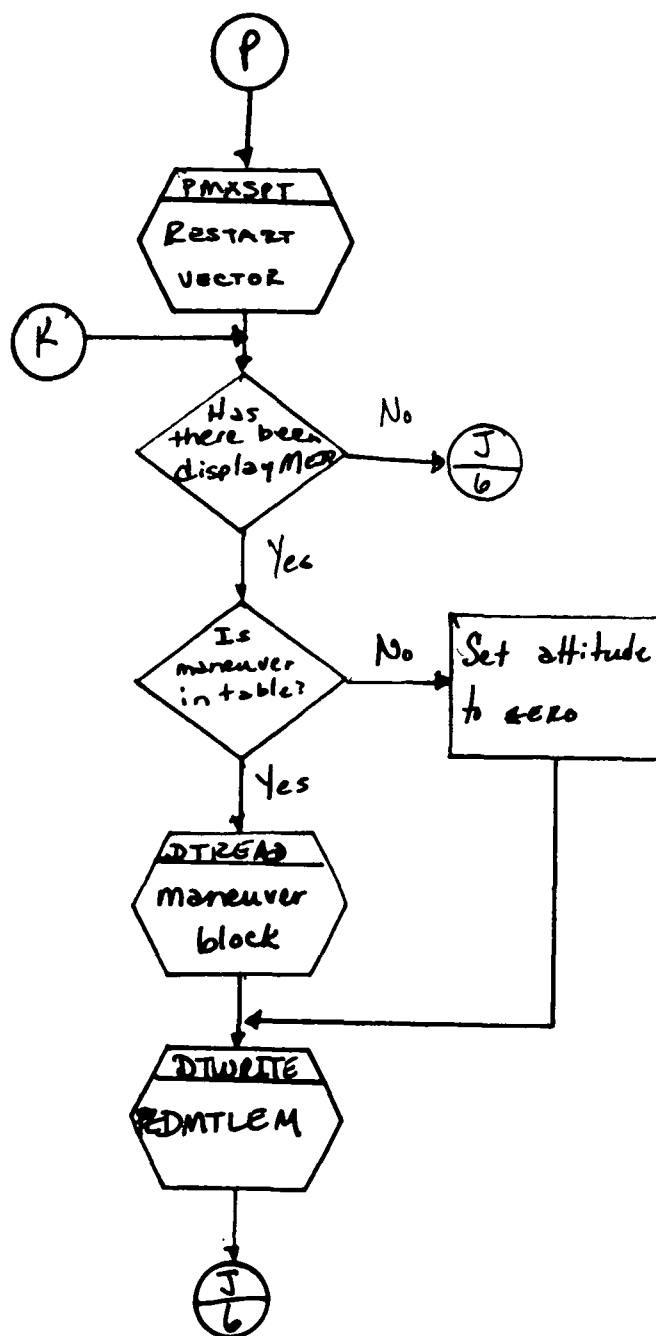


Figure PMMDMT-1. Flowchart

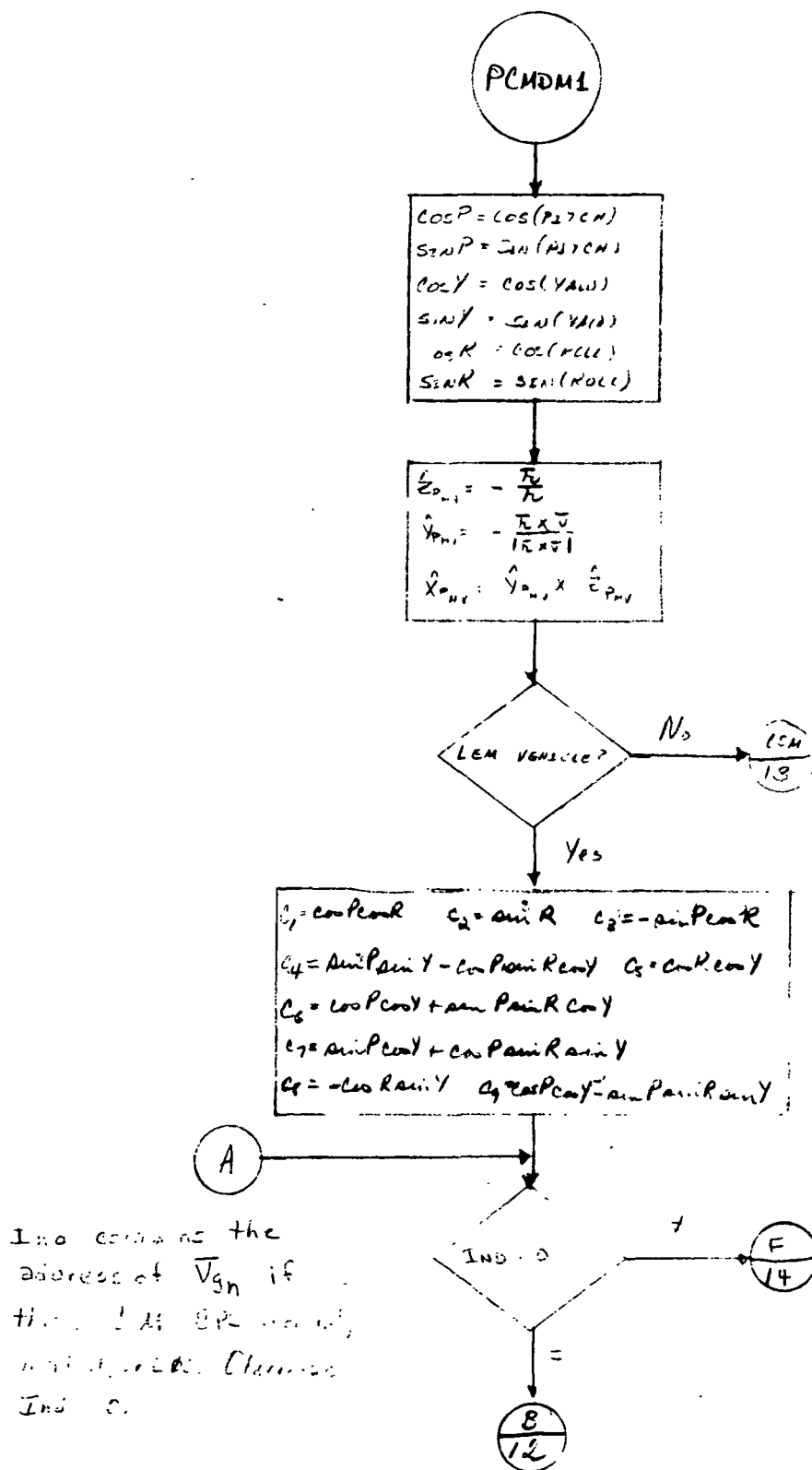


Figure PMMDMT-1. Flowchart

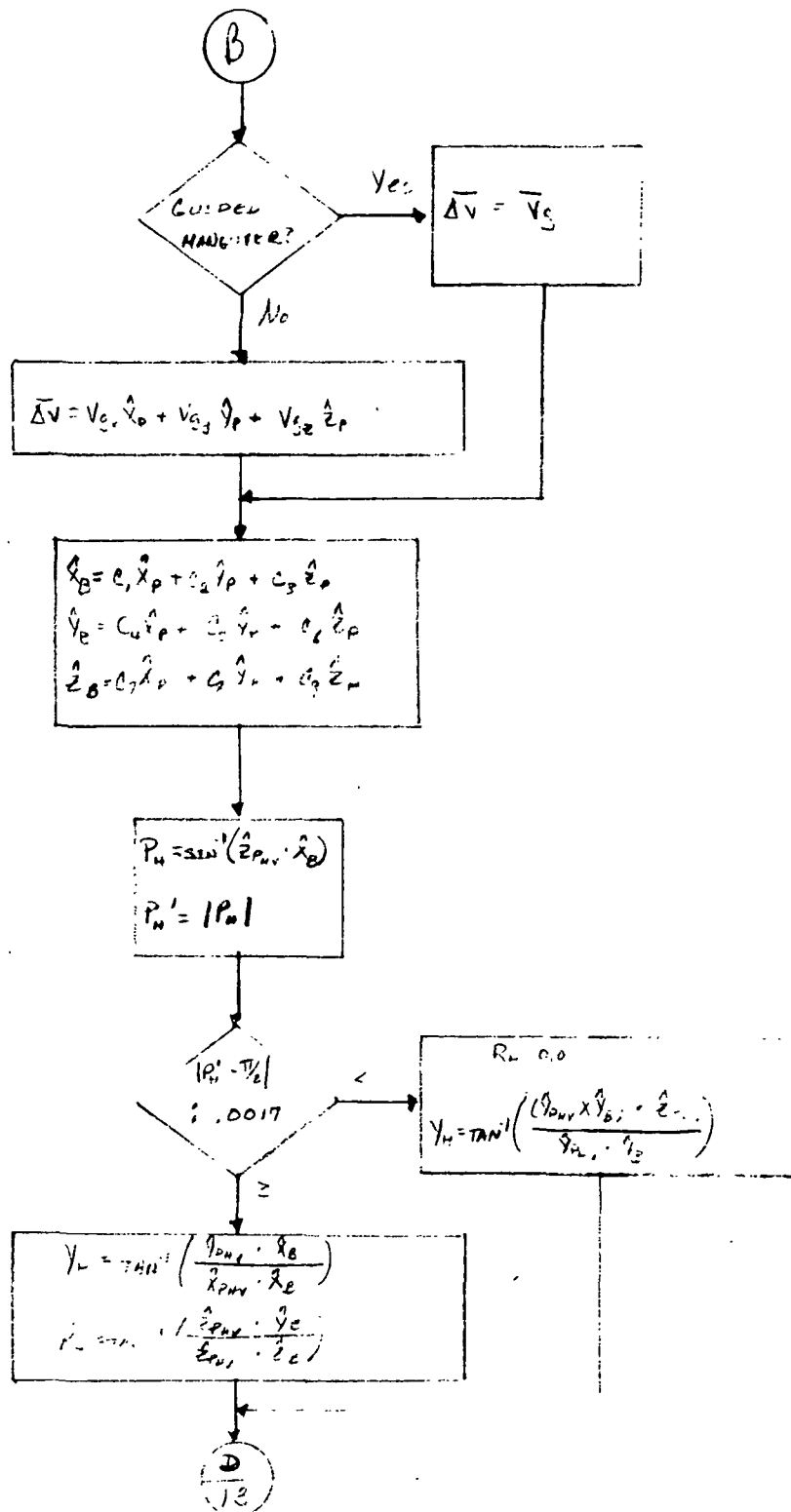


Figure PMMDMT-1. Flowchart

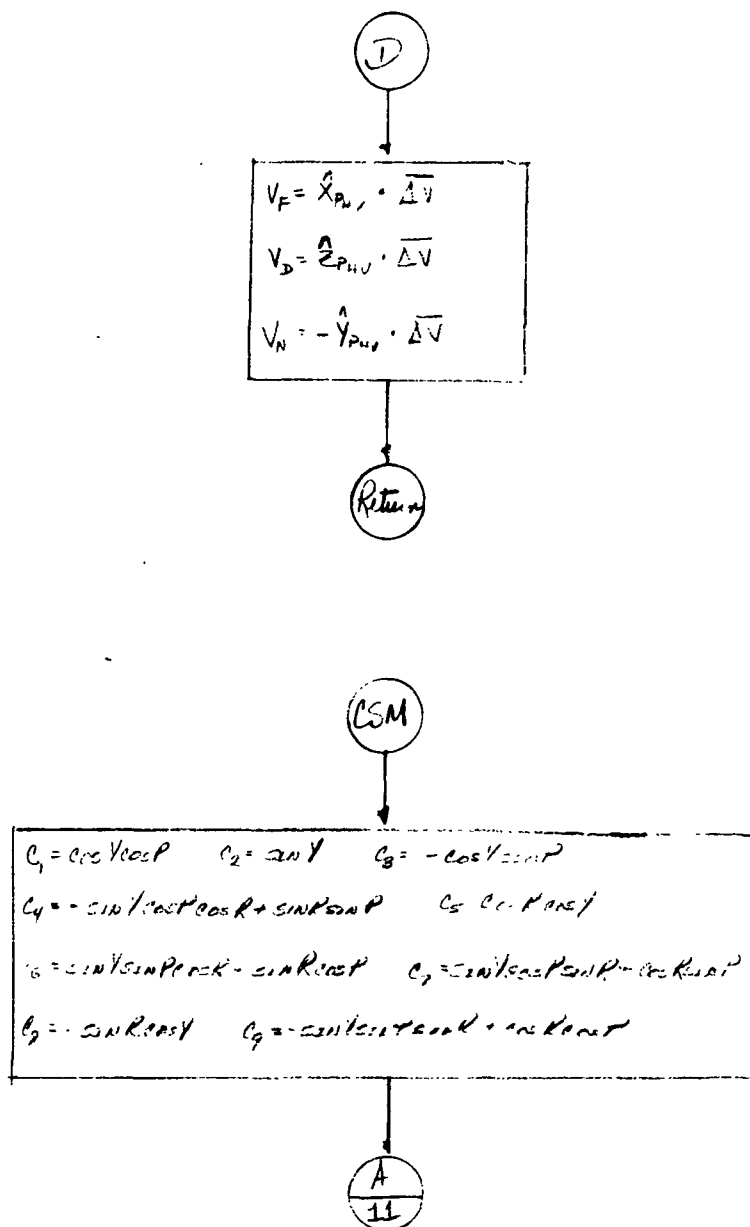


Figure PMMDMT-1. Flowchart

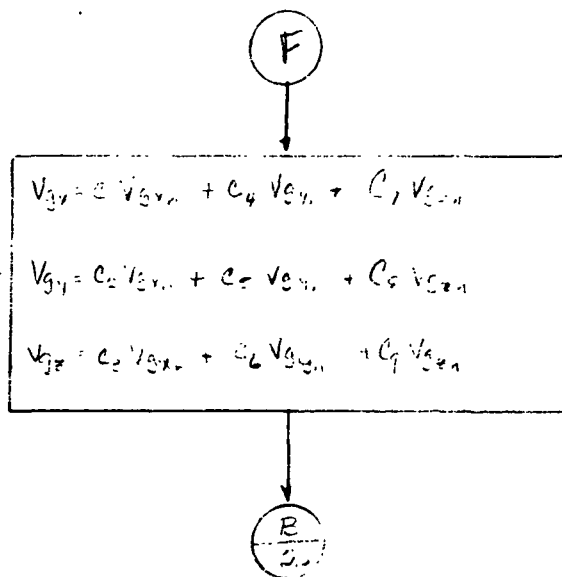


Figure PMMDMT-1. Flowchart

$(\bar{Y}_T \cdot \hat{Y}_B) < 0$ indicates a heads down attitude should be used, so $Y_T = -Y_T$.
If $(\bar{Y}_T \cdot \hat{Y}_B) \geq 0$, Y_T is unchanged for heads up attitude.

$$\hat{Y}_T = \frac{\bar{Y}_T}{Y_T}$$

$$\hat{Z}_T = \hat{a}_T \times \hat{Y}_T$$

$$\hat{X}_B = \cos P \cos Y * \hat{a}_T - \cos P \sin Y * \hat{Y}_T + P * \hat{Z}_T$$

$$\hat{Y}_B = \sin Y * \hat{a}_T + \cos Y * \hat{Y}_T$$

$$\hat{Z}_B = \hat{X}_B \times \hat{Y}_B$$

P - Pitch gimbal angle

Y - Yaw gimbal angle (SPS) or roll gimbal angle (DPS)

The local vertical local horizontal (LVLH) computations are the same for all attitudes and thrusters. The ignition vectors ($\bar{r}_I$, $\bar{v}_I$) and a ΔV vector ($\Delta \bar{V}$) are used.

$$\hat{Z}_{P_{HV}} = -\frac{\bar{r}_I}{|\bar{r}_I|}$$

$$\hat{Y}_{P_{HV}} = -\frac{(\bar{r} \times \bar{v})}{|\bar{r} \times \bar{v}|}$$

$$\hat{X}_{P_{HV}} = \hat{Y}_{P_{HV}} \times \hat{Z}_{P_{HV}}$$

$$P_H = \sin^{-1}(\hat{Z}_{P_{HV}} \cdot \hat{X}_B)$$

$$Y_H = \tan^{-1} \left[\frac{\hat{Y}_{P_{HV}} \cdot \hat{X}_B}{\hat{X}_{P_{HV}} \cdot \hat{X}_B} \right]$$

$$R_H = \tan^{-1} \left[\frac{\hat{Z}_{P_{HV}} \cdot \hat{Y}_B}{\hat{Z}_{P_{HV}} \cdot \hat{Z}_B} \right]$$

If P_H is equal to $\pm \pi/2$, the equations for Y_H and R_H are:

$$R_H = 0.0$$

$$Y_H = \text{TAN}^{-1} \left[\frac{-\hat{Y}_B \cdot \hat{X}_{P_{HV}}}{\hat{Y}_B \cdot \hat{Y}_{P_{HV}}} \right]$$

where: $\hat{X}_B, \hat{Y}_B, \hat{Z}_B$ Unit vectors in the direction of the vehicle body X, Y, and Z axes.

P_H, Y_H, R_H Pitch, yaw, and roll respectively in the LVLH coordinate system.

The velocity components are computed as:

$$\begin{aligned} \overline{\Delta V} &= \Delta V_M * \hat{a}_T \\ V_F &= \hat{X}_{P_{HV}} \cdot \overline{\Delta V} \\ V_S &= \hat{Y}_{P_{HV}} \cdot \overline{\Delta V} \\ V_D &= \hat{Z}_{P_{HV}} \cdot \overline{\Delta V} \end{aligned}$$

where V_F Component of ΔV in the local horizon plane at ignition, perpendicular to the position vector and positive in the direction of motion.

V_S Component of ΔV along the negative of the angular momentum vector at ignition

V_D Component of ΔV along the radius vector at ignition, positive toward the earth.

The primary external ΔV attitude requires computing the ΔV components in the command and display coordinate system.

$$\begin{aligned} V_{F_X} &= \bar{V} \cdot \hat{X}_{P_{HV}} & \bar{V} & \text{External } \overline{\Delta V} \text{ in RTCC inertial coordinate system} \\ V_{S_X} &= \bar{V} \cdot \hat{Y}_{P_{HV}} & V_{F_X}, V_{S_X}, V_{D_X} & \text{defined as } V_F, V_S, V_D \\ V_{D_X} &= \bar{V} \cdot \hat{Z}_{P_{HV}} & \text{previously. } \hat{X}_{P_{HV}}, \hat{Y}_{P_{HV}}, \hat{Z}_{P_{HV}} & \text{as defined previously,} \end{aligned}$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMEPHEM
 DATE 4/21/67 (205)
 REV
 PAGE 1 of 12

2. PMMEPHEM

- a. Symbol — PMMEPHEM
- b. Program Name — Sun - Moon Ephemeris Generator
- c. Programming Group — Mission (500 Apollo Flight Plans)
- d. Programmer(s) — C. R. Scott
- e. Language — Assembler
- f. Data Tables Used — PZEFDATA and PZEFEM
- g. MED's Used — None
- h. Writeup Completion Date — 9/22/66
- i. Flowchart — Figure PMMEPHEM-1.

FUNCTION

This load module provides, as a function of time, ephemeris data about the Sun, Moon, and Earth for any time from year 1965, hour 3803.99, up to year 1970, hour 4811.99.

USAGE

- a. Calling Sequence — This load module is called with a LINK or a LOAD and CALL to PMMEPHEM, with one input array.
- b. Inputs — Input is a five-word array. The first two words are a double precision hour (universal time), and the third word is an integer year. They give the time for which data are requested. Every pass through the routine with a nonzero year indicates an initialization pass; a year equal to zero indicates that an initialization pass has been made and the hour given is a delta hour past the time of the last initialization pass. The fourth word of the array is an integer indicating the type of output requested (see USAGE, f). The fifth word is used for an error indicator output.

Ephemeris data are input from data table PZEFEM (see writeup).

- c. Outputs and Exit — The routine outputs the following data in data table PZEFDATA (see writeup):

Cartesian geocentric Moon position $(\vec{r}_{M/E})$ or
 selenocentric Earth position $(\vec{r}_{E/M})$

Approval	Approval
C. R. Scott	R. O. Sogard

Cartesian geocentric Moon velocity ($\vec{v}_{M/E}$) or
selenocentric Earth velocity ($\vec{v}_{E/M}$)

Delta time (Δt) from initialization corresponding to the most recent output from PMMEPHEM.

Position (r) magnitude of $\vec{r}_{M/E}$ or $\vec{r}_{E/M}$

Position (r^2) magnitude squared

Position (r^3) magnitude cubed

Velocity (v) magnitude of $\vec{v}_{M/E}$ or $\vec{v}_{E/M}$

Velocity (v^2) magnitude squared

Velocity (v^3) magnitude cubed

Cartesian geocentric ($\vec{r}_{S/E}$) or selenocentric ($\vec{r}_{S/M}$)
Sun position

Moon's 3x3 libration matrix (XLIB) used for converting from
selenocentric to selenographic coordinates.

The following combinations of output are triggered by an input indicator set to the following values:

<u>Indicator</u>	<u>Output</u>
1	$\vec{r}_{M/E}$, Δt
2	$\vec{r}_{E/M}$, Δt
3	$\vec{r}_{M/E}$, Δt , $\vec{r}_{S/E}$
4	$\vec{r}_{E/M}$, Δt , $\vec{r}_{S/M}$, XLIB
5	$\vec{r}_{M/E}$, $v_{M/E}$, Δt , r , r^2 , r^3 , v , v^2 , v^3 , $\vec{r}_{S/E}$, XLIB

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMEPHEM
DATE 4/21/67 (205)
REV
PAGE 3 of 12

<u>Indicator</u>	<u>Output</u>
6	$\vec{r}_{M/E}, \vec{v}_{M/E}, \Delta t$
7	$\vec{r}_{E/M}, \vec{v}_{E/M}, \Delta t$
8	$\Delta t,$ XLIB

The only exit is a return to the calling program.

- d. Other Programs Used — None
- e. Data Restrictions and Limits — The user's input indicator is not checked, and an invalid one will cause the routine to transfer unpredictably.

The input year must be either one less than the Besselian year on PZEFEM or equal to it. The input (universal) hour must be equal to or greater than 35.99 hours past the first (ephemeris) hour on the table and less than 1403.99 hours past it.

- f. Errors — The error indicator is stored in the fifth word of the user's input array. The value of it corresponds to a certain type of error. For a nonzero value, no other outputs are given.

<u>Indicator</u>	<u>Error</u>
0	No error; output is valid
1	Input year is negative
2	Input year is not within range of data on PZEFEM
3	Input hour is not within range of data on PZEFEM
4	Input year is zero, but base time has not been initialized
5	Data table PZEFEM is not initialized

METHOD

An indicator is checked to determine whether the ephemeris data table (PZEFEM) has been DTLOADED. If not, it is DTLOADED and the indicator is set accordingly.

If the input year indicates an initialization pass, the year is compared with the Besselian year (epoch) on the data table to determine whether data of that year can be referenced to that epoch. The requested hour is calculated as the



number of hours from January 0 (ephemeris time) of the year previous to the epoch. If it is not an initialization pass, the delta time is added to the base time.

The hour is then normalized with respect to the hour of the beginning of the data table and is checked to determine if enough data are available on the data table to interpolate for the requested hour.

The input indicator is tested to determine the type of data requested, and the data are calculated with the Lagrangian Interpolation Formula. The formula is

$$U_x = \frac{(x-b)(x-c) \cdots (x-k)}{(a-b)(a-c) \cdots (a-k)} U_a + \cdots + \frac{(x-a)(x-b) \cdots (x-j)}{(k-a)(k-b) \cdots (k-j)} U_k$$

in which the values $U_a, U_b, \cdots, U_k$ of the dependent variable are known for the values $a, b, \cdots, k$ of the independent variable.

A fifth-order polynomial is used, that is,

$$U_x = \sum_{i=1}^6 C_i U_i \text{ in which}$$

$$C_i = \frac{\prod_{j \neq i} (t - t_j)}{\prod_{j \neq i} (t_i - t_j)}$$

t is the relative position of x in the interval (x_3, x_4)

$$t_1 = -3$$

$$t_2 = -2$$

$$t_3 = -1$$

$$t_4 = 0$$

$$t_5 = +1$$

$$t_6 = +2$$



The interval of the dependent variable table to which the input independent variable corresponds is calculated; then its relative position in that interval is computed. From those and the pointer to the first known value of the dependent variables requested, the six known values of the function are picked up, and the output value is calculated. The known values for the next dependent variable are picked up for interpolation, and this continues until all the dependent variables requested are calculated.

The information is put into an array, the error indicator is set to zero, and the array is DTWRITE onto PZEFDATA, the output data table for PMMEPHEM.

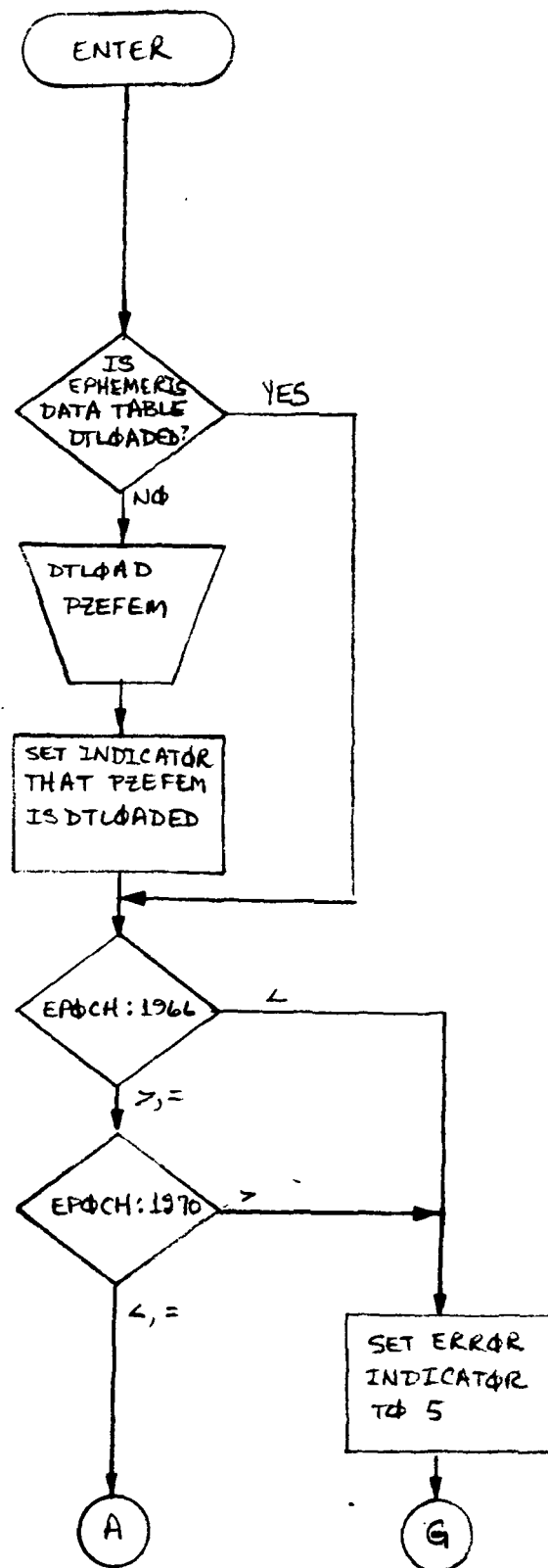


Figure PMMEPHEM-1. Flowchart

(Sheet 1 of 7)

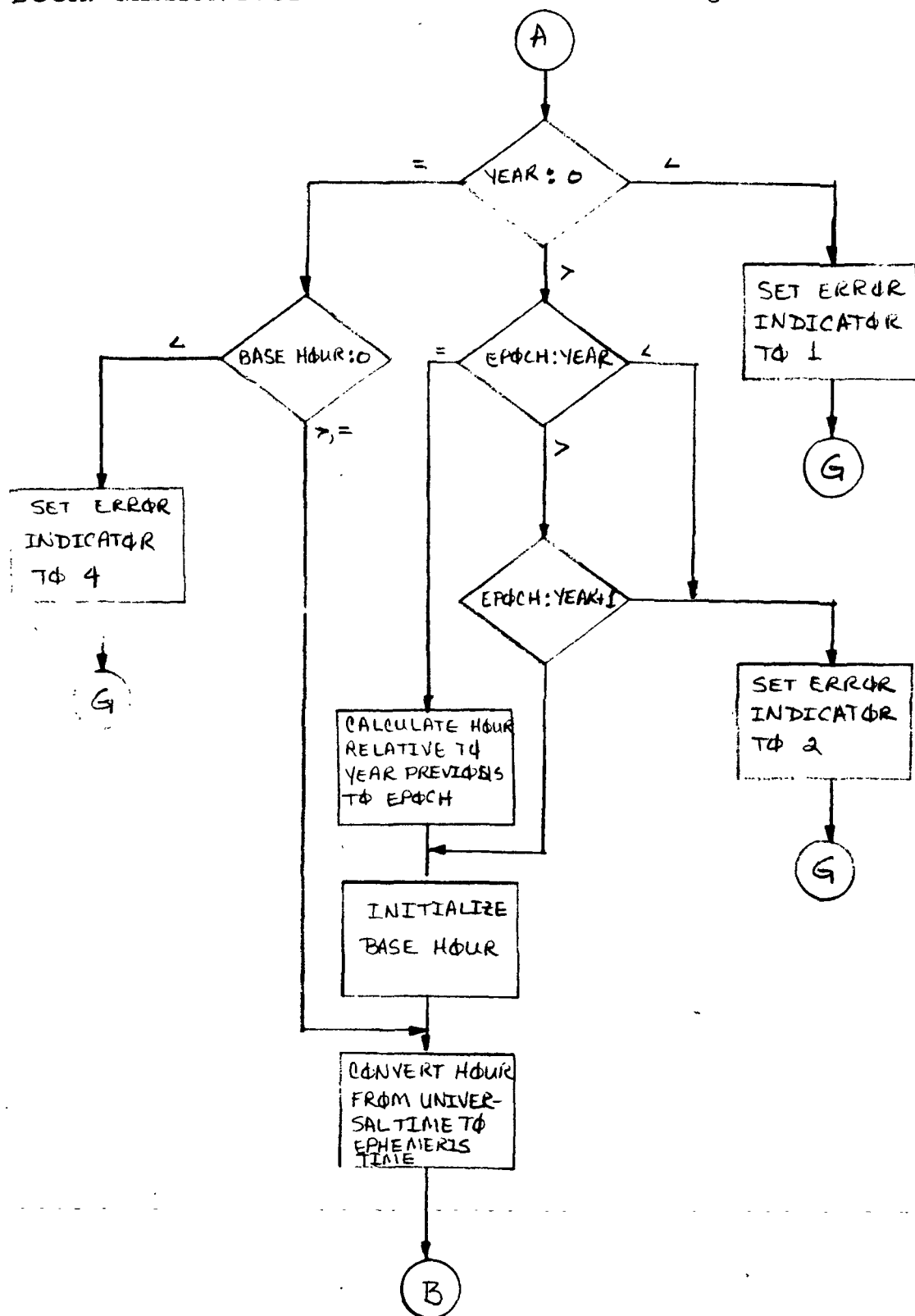


Figure PMMEPHEM-1. Flowchart

(Sheet 2 of 7)

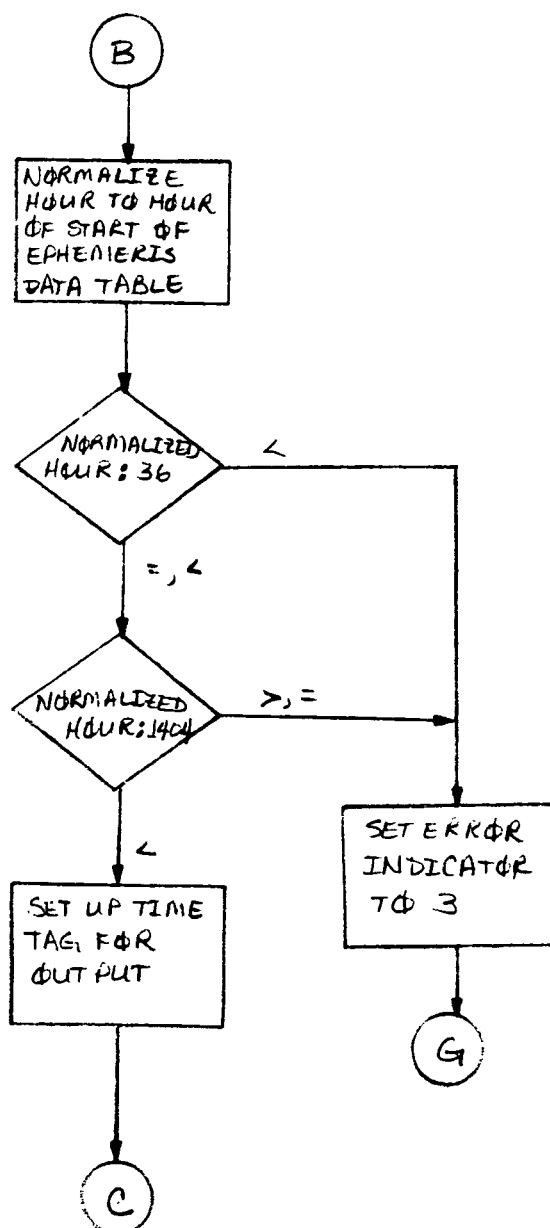


Figure PMMEPHEM-1. Flowchart

(Sheet 3 of 7)



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMEPHEM

DATE 4/21/67 (205)

REV

PAGE 9 of 12

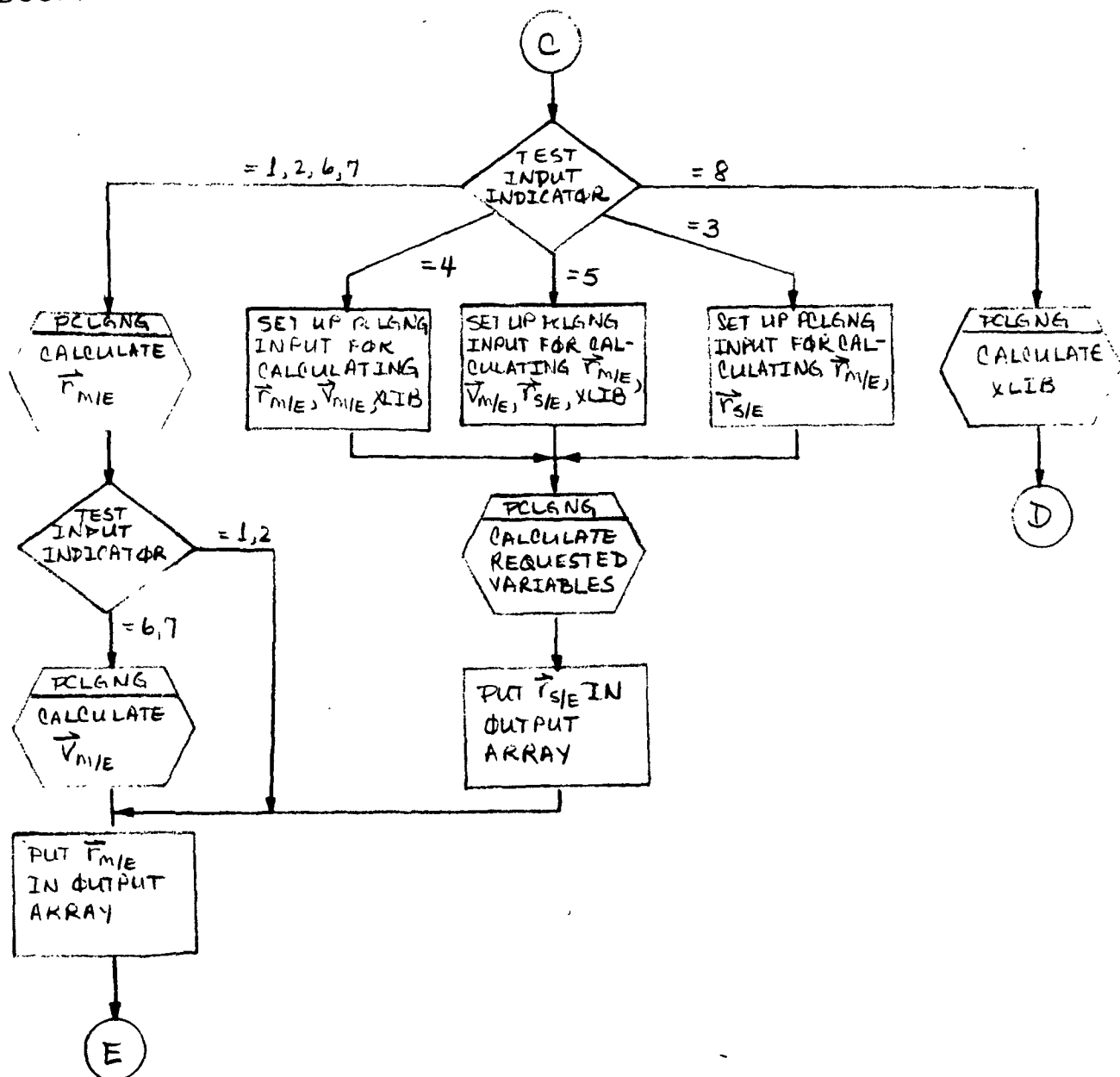


Figure PMMEPHEM-1. Flowchart

(Sheet 4 of 7)

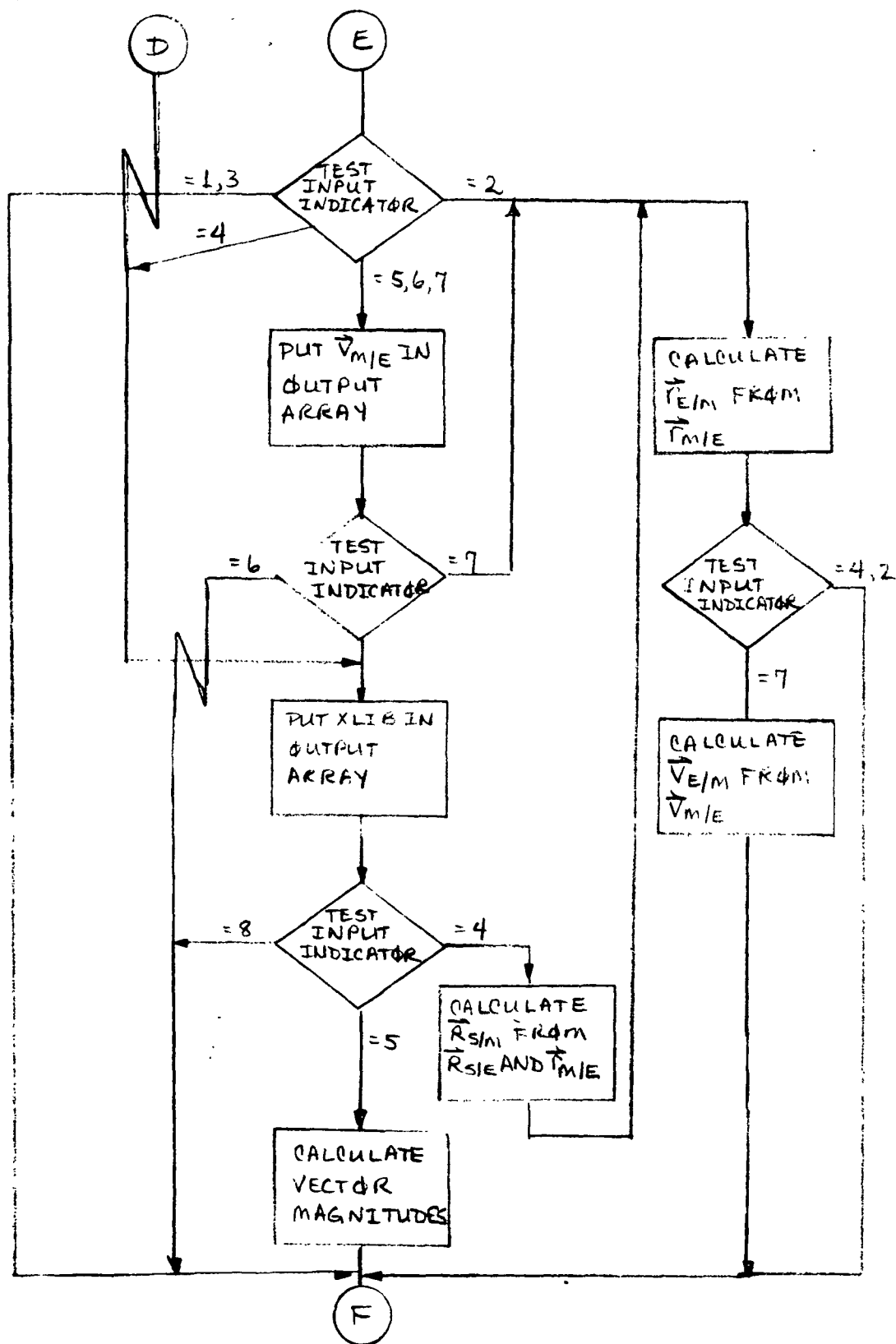


Figure PMMEPHEM-1. Flowchart

(Sheet 5 of 7)

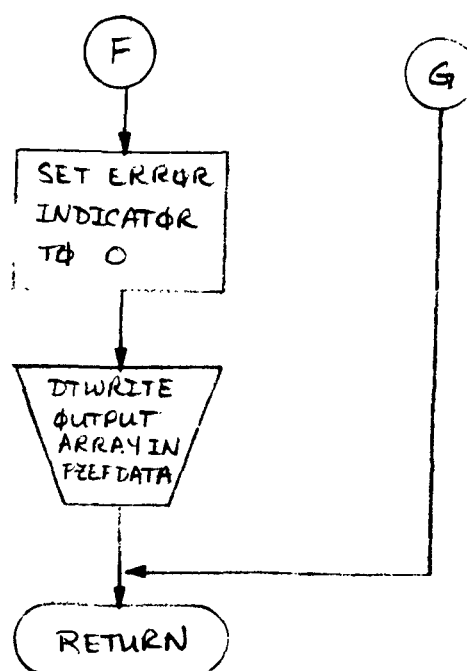


Figure PMMEPHEM-1. Flowchart

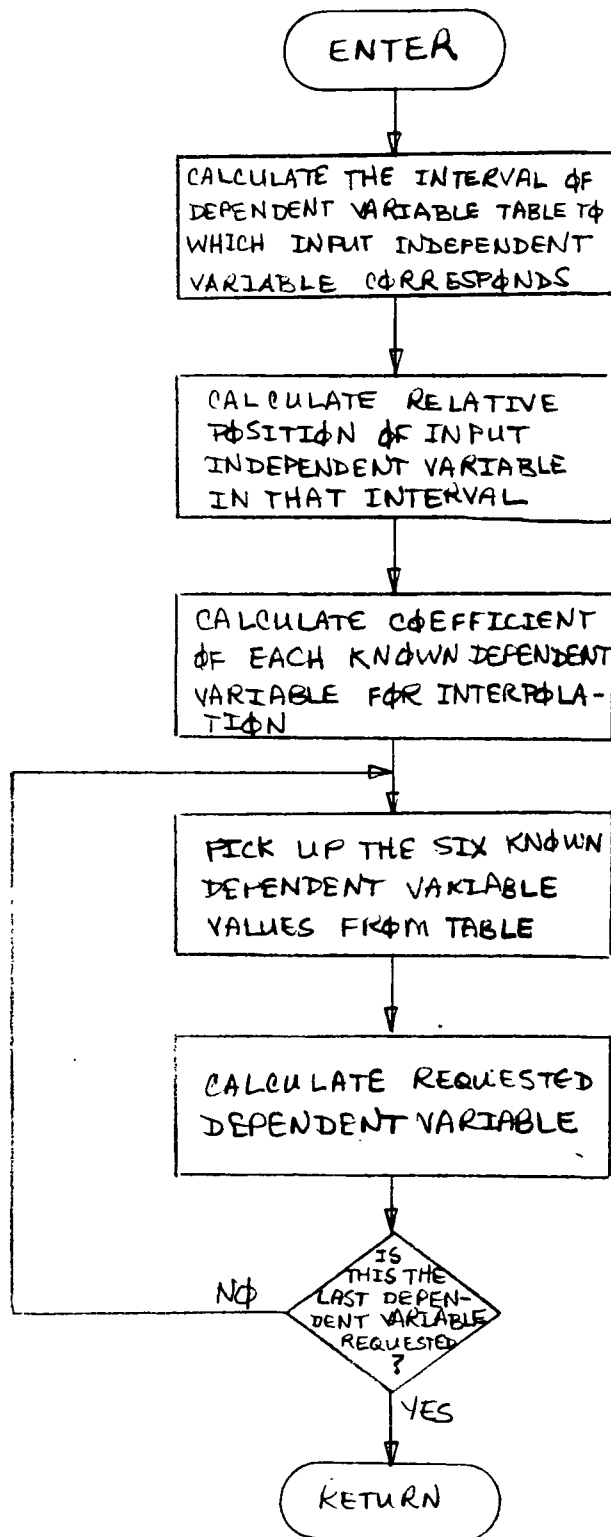


Figure PMMEPHEM-1. Flowchart

(Sheet 7 of 7)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMFSET
DATE 9/21/67 (EORP)
REV
PAGE 1 of 7

- a. Symbol — PMMFSET
- b. Program Name — Computation of Actual Height, Phase, and Wedge Angle Offsets
- c. Programming Group — Rendezvous Planning (Mission Planning)
- d. Programmer — J. A. Steed
- e. Language — Assembler
- f. Data Tables Used — GZGENCSN, PZMPTCSM, PZMPTLEM
- g. MED's Used — None
- h. Writeup Completion Date — 8/15/67
- i. System Parameters — Table PMMFSET-1
- j. Flowchart — Figure PMMFSET-1

FUNCTION

PMMFSET computes the height, phase, and wedge angle offsets after a CDH or NSR maneuver, and the elevation angle after a TPI maneuver. It then queues the DKI Rendezvous Evaluation Table to display these values.

USAGE

- a. Calling Sequence — RTATTACH TASK = PTMANC, EP = PMMFSET
- b. Inputs — None
- c. Outputs and Exit — Stores the computed values in words 36-38 of GZGENCSN. RTATTACH TASK=GTISPLAY, EP = GMSNAP, ID = 3, PARAM = (WORD, 4, FKWORD, 4) where WORD is the RET display TVGUIDE word, 4 the length, and FKWORD the end of parameters.
- d. Other Programs Used — PMSVEC, PMMAEG, DCOS, DSIN, DARCOS, DSQRT
- e. Data Restrictions and Limits — None
- f. Error — If an error is received from PMSVEC or PMMAEG, an = X 'FF' is stored in the first byte of word 36 in GZGENCSN, and the above exit is taken.

METHOD

PMMFSET obtains the height difference and the phase angle and computes the wedge angle at the endtime of either a NSR (DKI circularization) or a CDH (SPQ co-elliptical orbit) maneuver. If the maneuver was an NSR, the three values are placed in the General Constraints Table; the Rendezvous Evaluation Table Display is queued; and the exit is taken. If the maneuver was a CDH, the TPI (Terminal Phase Initiate) maneuver is sought. Elements for both vehicles are obtained at the beginning of this maneuver, and the phase angle

Approval	Approval
<i>J. A. Steed</i>	<i>A. Ametti</i>

obtained at its impulsive time. The phase angle is used in computing the elevation angle, which is stored with the height difference and the wedge angle in GZGENCSN; the RET Display is queued; and the exit taken. If there is no TPI maneuver, the elevation angle is set to zero, and the above procedure is followed. (For further details, see Figure PMMFSET-1.)

The following formulas are used in the computation of the wedge angle and the elevation angle:

$$\text{wedge angle} = \cos^{-1} \left[\cos(i_c) \cos(i_T) + \sin(i_c) \sin(i_T) \cos(h_c - h_T) \right]$$

$$\text{elevation angle} = \cos^{-1} \left[\sqrt{\frac{(r_T)^2 (\sin^2 \theta)}{(r_c)^2 + (r_T)^2 - 2(r_T)(r_c)(\cos \theta)}} \right]$$

where:

- c = the chaser (maneuvering vehicle),
- T = target (non-maneuvering vehicle),
- i = angle of inclination,
- h = the argument of the ascending node,
- r = position magnitude, and
- θ = the phase angle.

*Type
H — Fixed-point half word
F — Fixed-point full word
E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

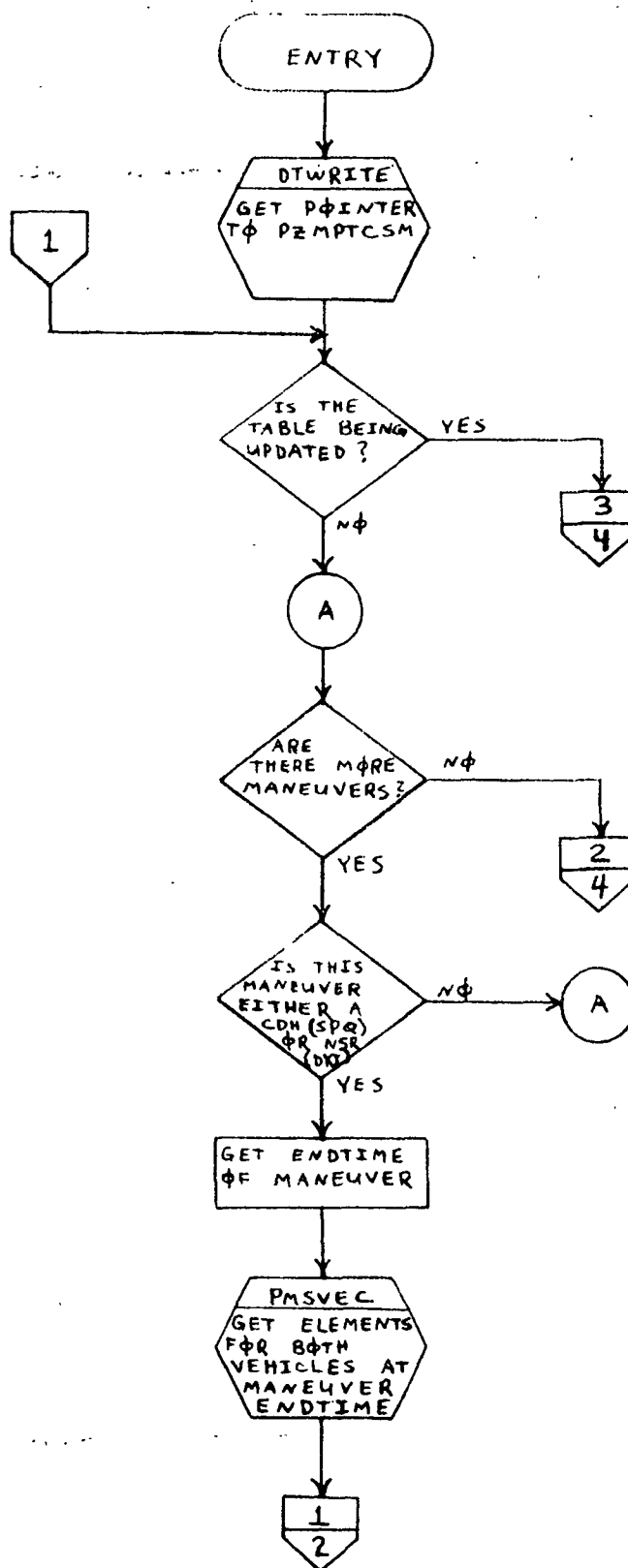


Figure PMMFSET-1. Flowchart

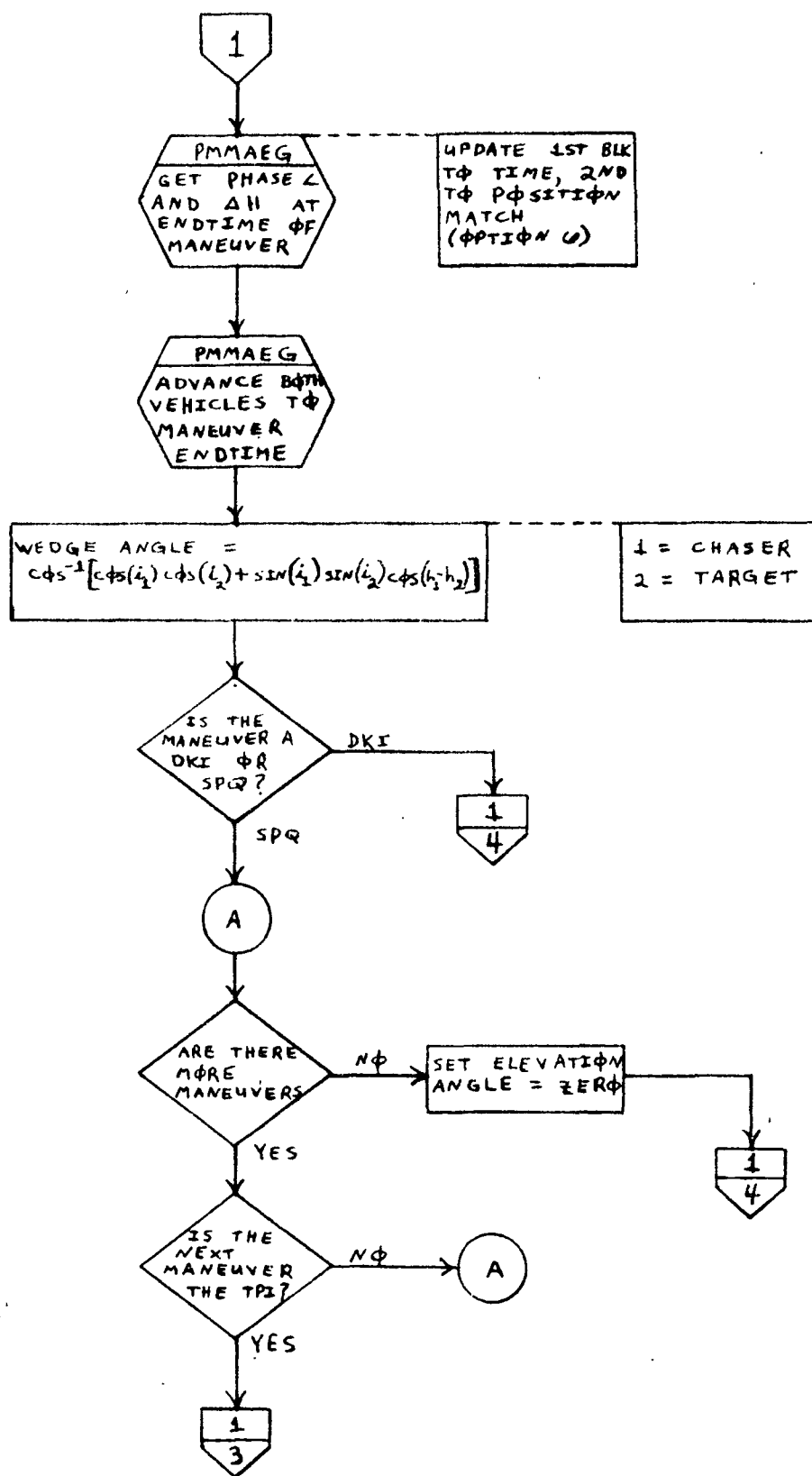


Figure PMMFSET-1. Flowchart

(Sheet 2 of 4)

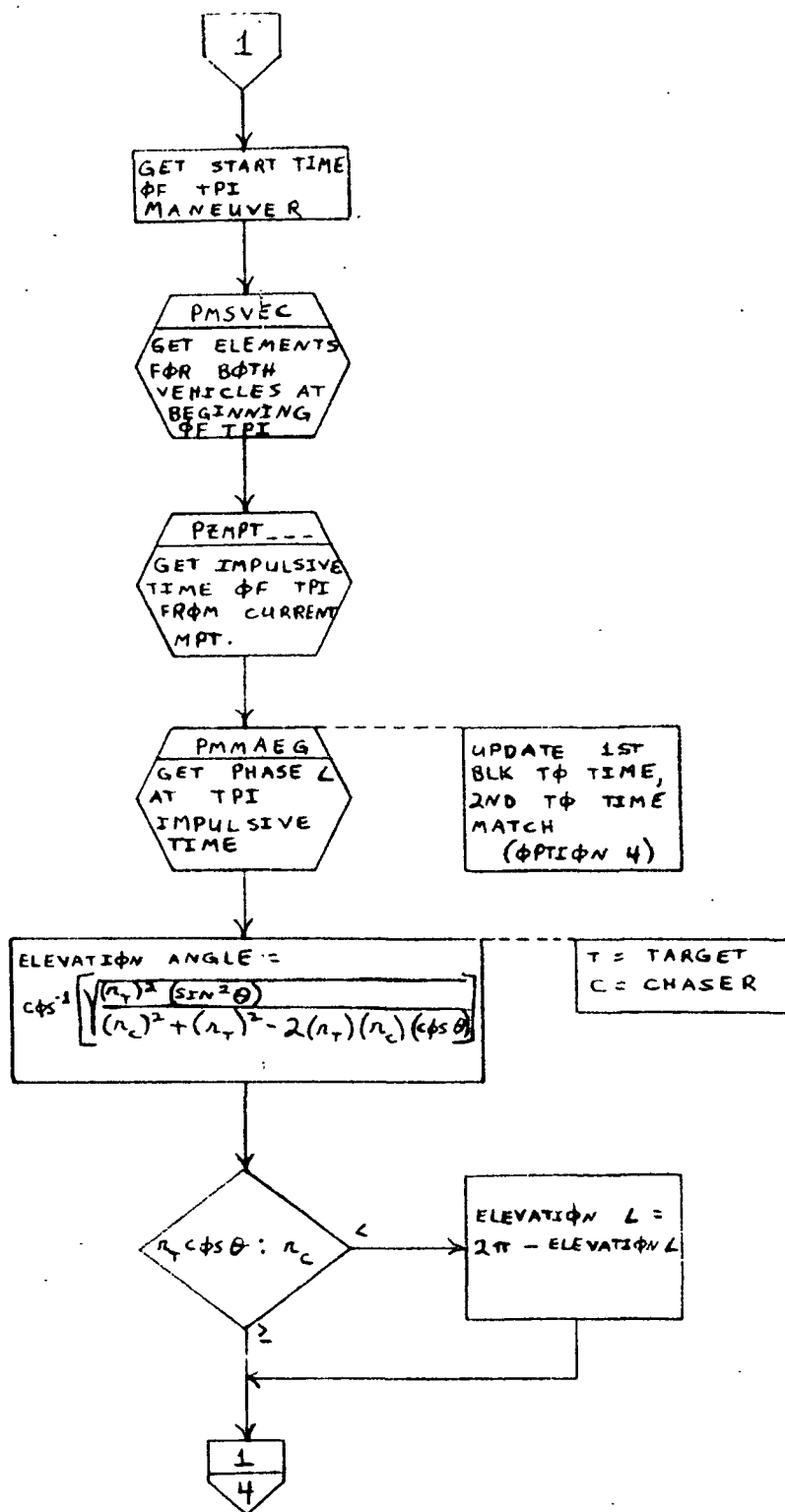


Figure PMMFSET-1. Flowchart

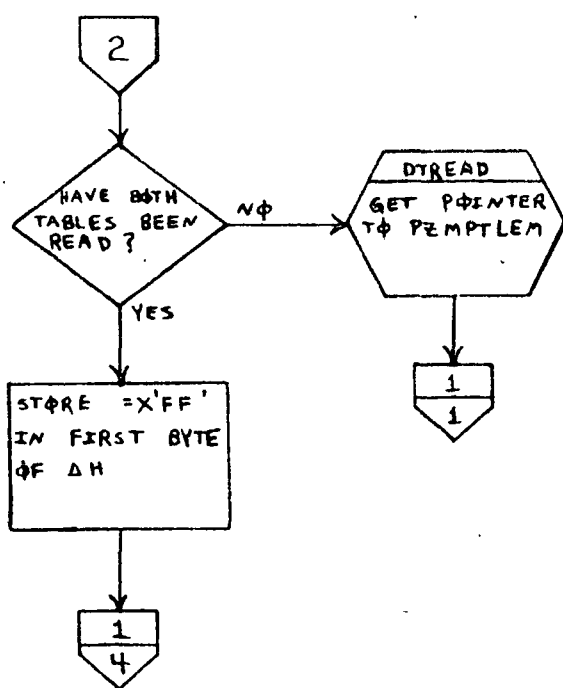
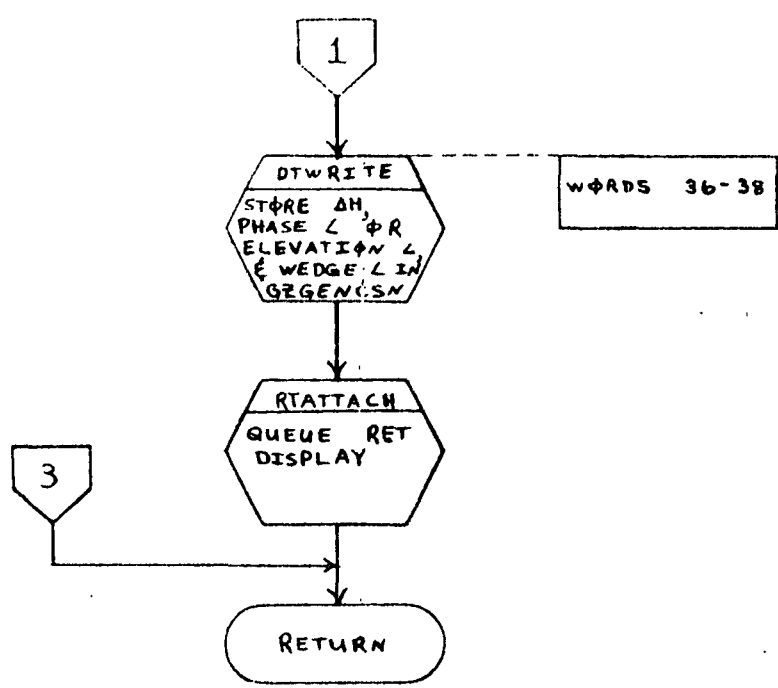


Figure PMMFSET-1. Flowchart (Sheet 4 of 4)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS -AS-200 Mission Planning

2. PMMFUD
DATE 6/22/67 (206)
REV
PAGE 1 of 10

- a. Symbol - PMMFUD
- b. Program Name - Freeze, Unfreeze, Delete Processor
- c. Programming Group - Mission Planning
- d. Programmer(s) - E. Kramer
- e. Language - Assembler
- f. Data Tables Used - PZMPTLEM, PZMPTS4B
- g. MED's Used - M62
- h. Writeup Completion Date - 5/09/67
- i. Parameter List - Table PMMFUD-1
- j. Task Subpool List - Table PMMFUD-2
- i. Flowchart — Figure PMMFUD-1

FUNCTION

PMMFUD freezes, unfreezes, deletes, and history-deletes maneuvers in the Mission Planning Table (MPT).

USAGE

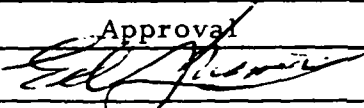
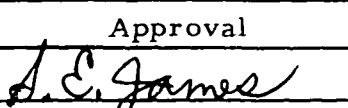
- a. Calling Sequence - RTATTACH TASK = PTMANC
EP = PMMFUD, ID = 40.
- b. Inputs - Vehicle Code, maneuver number, desired action
- c. Outputs and Exit - Change MPT according to action if valid and RETURN except on future delete on which it XCTL's to PMSVCT.
- d. Other Programs Used - None
- e. Data Restrictions and Limits - None
- f. Error - If any error conditions are encountered, an appropriate comment is printed on line, and the program exits.

METHOD

There are four possible actions taken by the program: freeze maneuver(s), unfreeze maneuver(s), delete future maneuver(s), and delete executed maneuver(s).

If freezing is requested, there is a check to see if the indicated maneuver is frozen; if not, it (and all prior maneuvers to it) is frozen. If it already is frozen, an error comment is printed on line.

If unfreezing is requested, there is a check to see if the indicated maneuver is frozen; if so, it and all subsequent maneuvers are unfrozen. If it was not frozen, an error comment is printed on line.

Approval 	Approval 
---	--

If a future delete is requested, a check is made to see if the maneuver is non-executed, and non-frozen; if so, it and all subsequent maneuvers are deleted, and a trajectory update is queued. If one of the tests fail, an appropriate comment is printed on line.

If a history delete is requested, a check is made to see if the maneuver is executed; if so, it and all prior maneuvers are deleted. If it is not executed, an error comment is printed on line.

IBM RTCC APOLLO PROGRAMMING SYSTEMS
 2. PMMFUD
 DATE 6/22/67 (206)
 REV
 PAGE 3 of 10

BOOK: MISSION SYSTEMS -AS-200 Mission Planning

Table PMMFUD-1. Parameter List

QUEID — 40

QNAME — GQ1CTL

From — GMSCTRL

To-Task — PTMANC

Purpose — Freeze, unfreeze, or delete
maneuvers

EP — PMMFUD

Byte No.	Type*	Units	Value / Description / Comments
1-4	F	-	Address of Subpool 1

*Type

H — Fixed-point half word

F — Fixed-point full word

E — Single-precision (floating-point)

D — Double-precision (floating-point)

X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMFUD
DATE 6/22/67 (206)
REV
PAGE 4 of 10

Table PMMFUD-2. Task Subpool List

Subpool Number 1

Symb. Ref.	Byte No.	Type*	Units	Value / Description / Comments
VEHICLE	1-4	F	-	Address of vehicle ID
	5-8	F	-	Address of maneuver number
	9-12	F		Address of desired action code
	13-16	F		Spare
	17-20	F		Vehicle ID 1 = CSM 2 = S-IVB 3 = LM
MANVNUM	21-24	F		Maneuver number
ACTION	25-28	F		Desired action 0 = Future delete 1 = History delete 2 = Freeze 3 = Unfreeze

*Type
H — Fixed-point half word
F — Fixed-point full word

E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

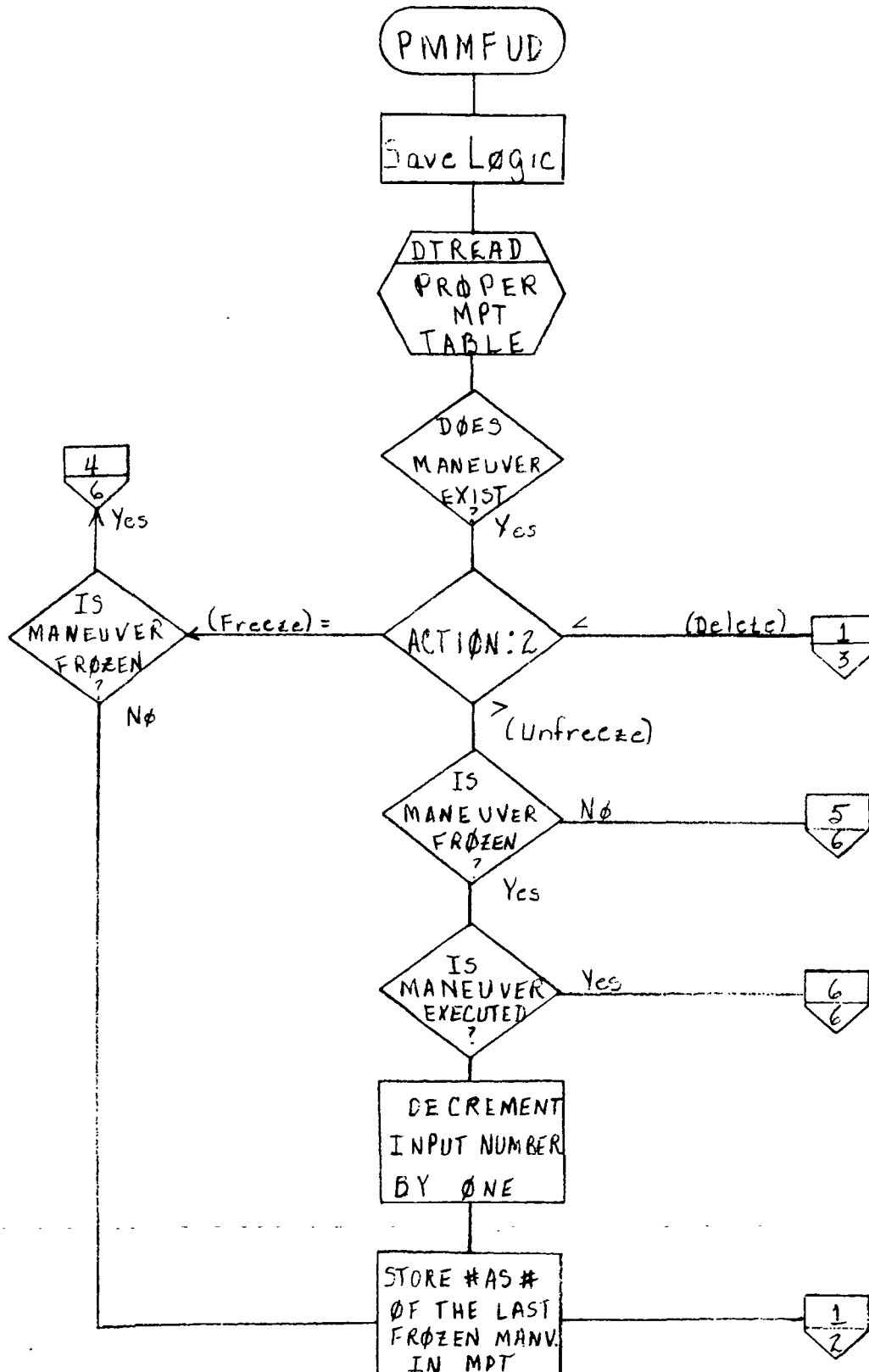


Figure PMMFUD-1. Flowchart

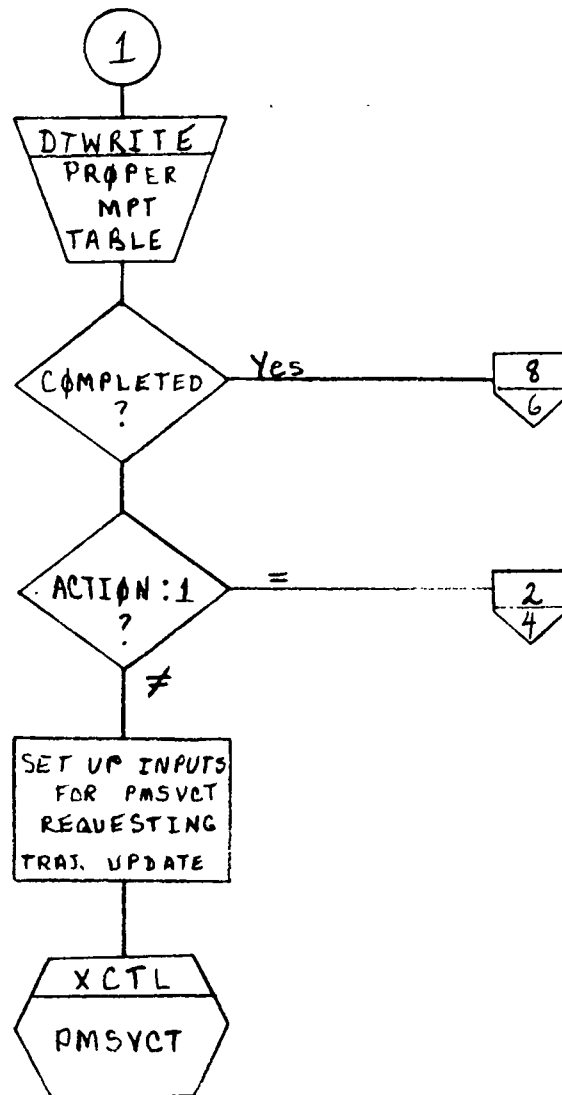


Figure PMMFUD-1. Flowchart

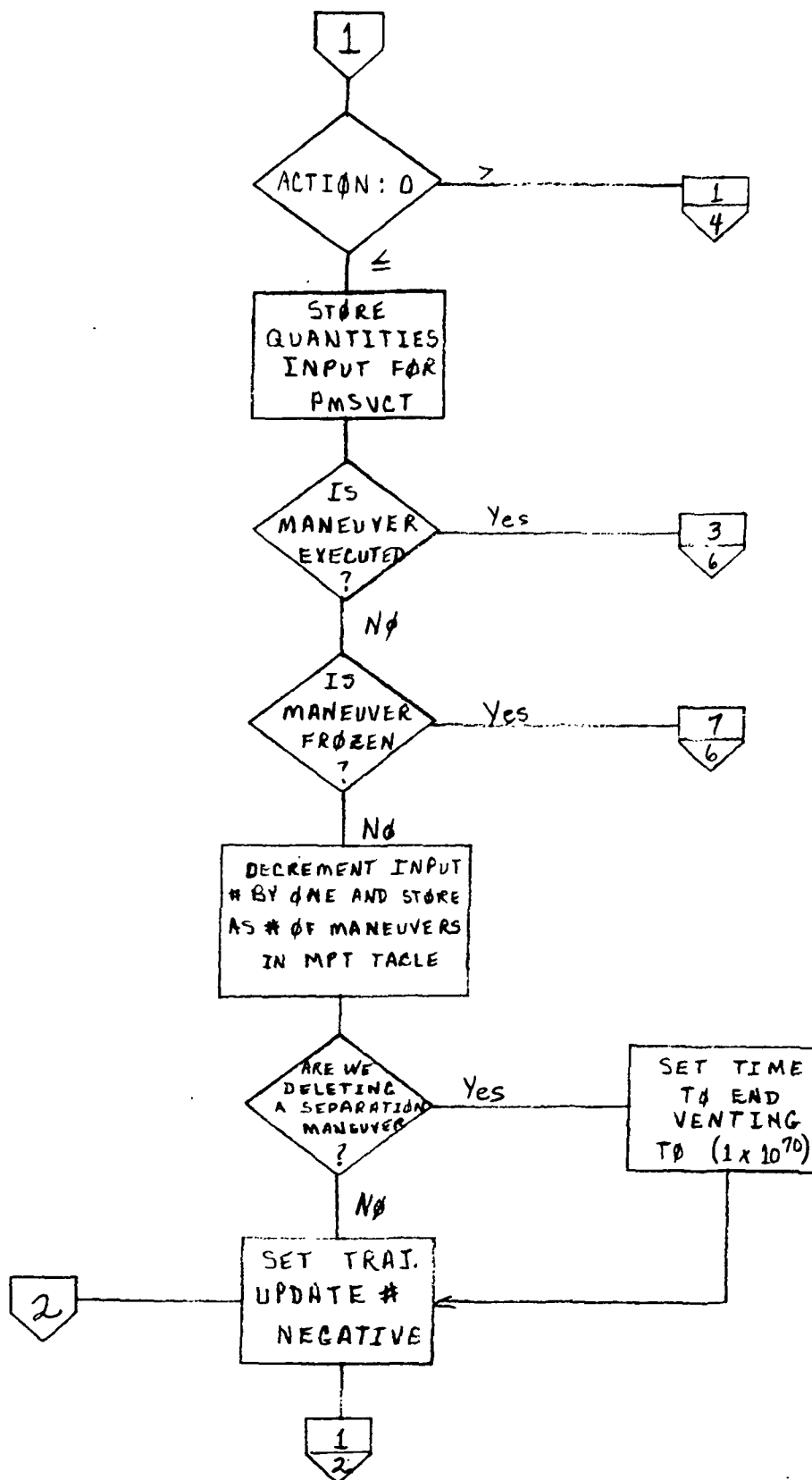


Figure PMMFUD-1. Flowchart

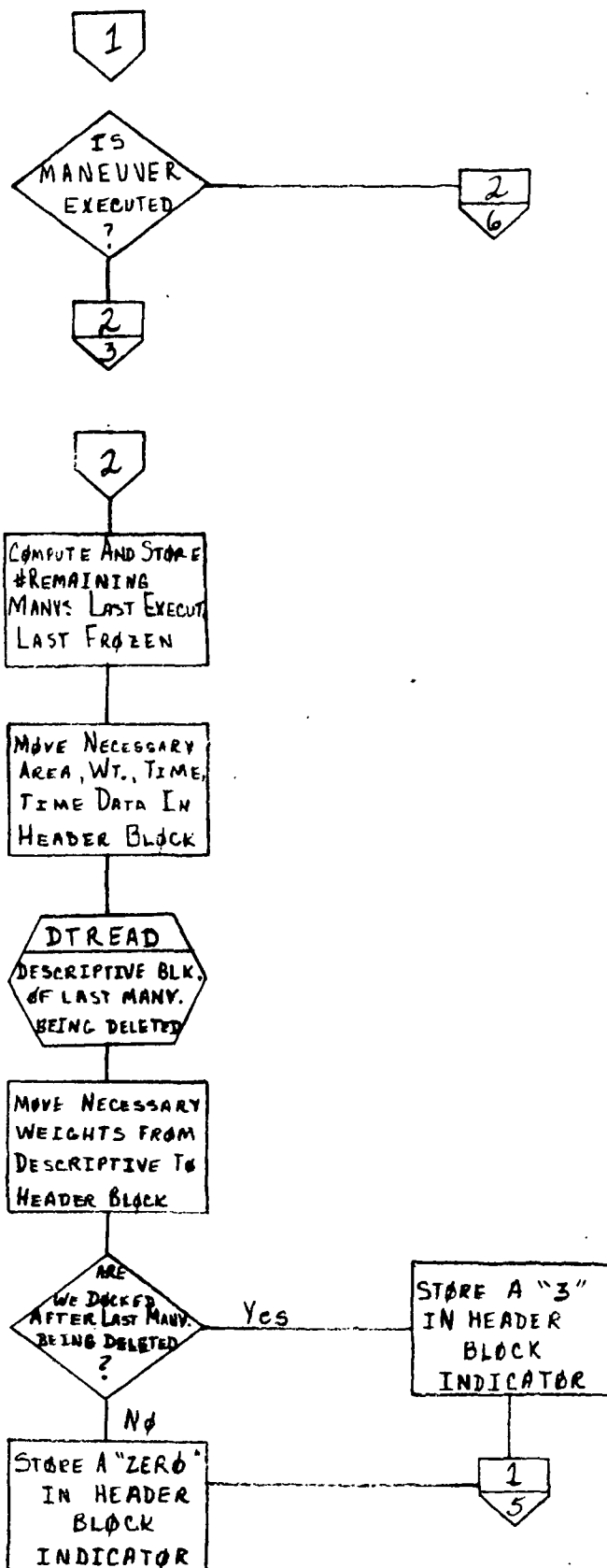


Figure PMMFUD-1. Flowchart

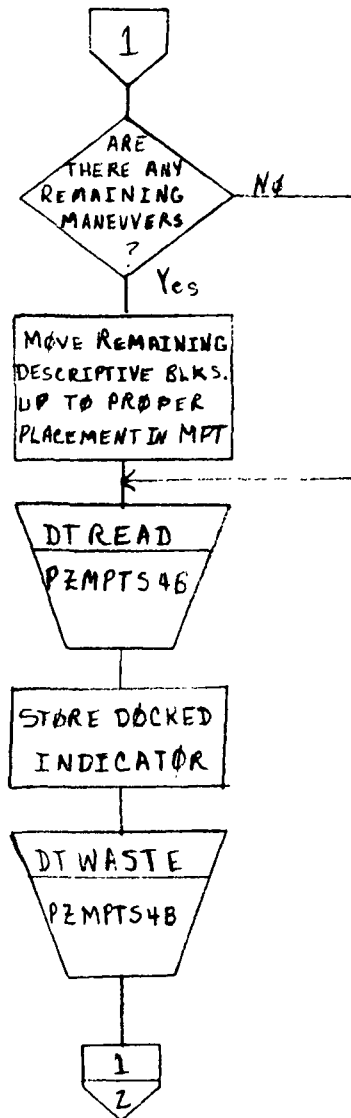


Figure PMMFUD-1. Flowchart

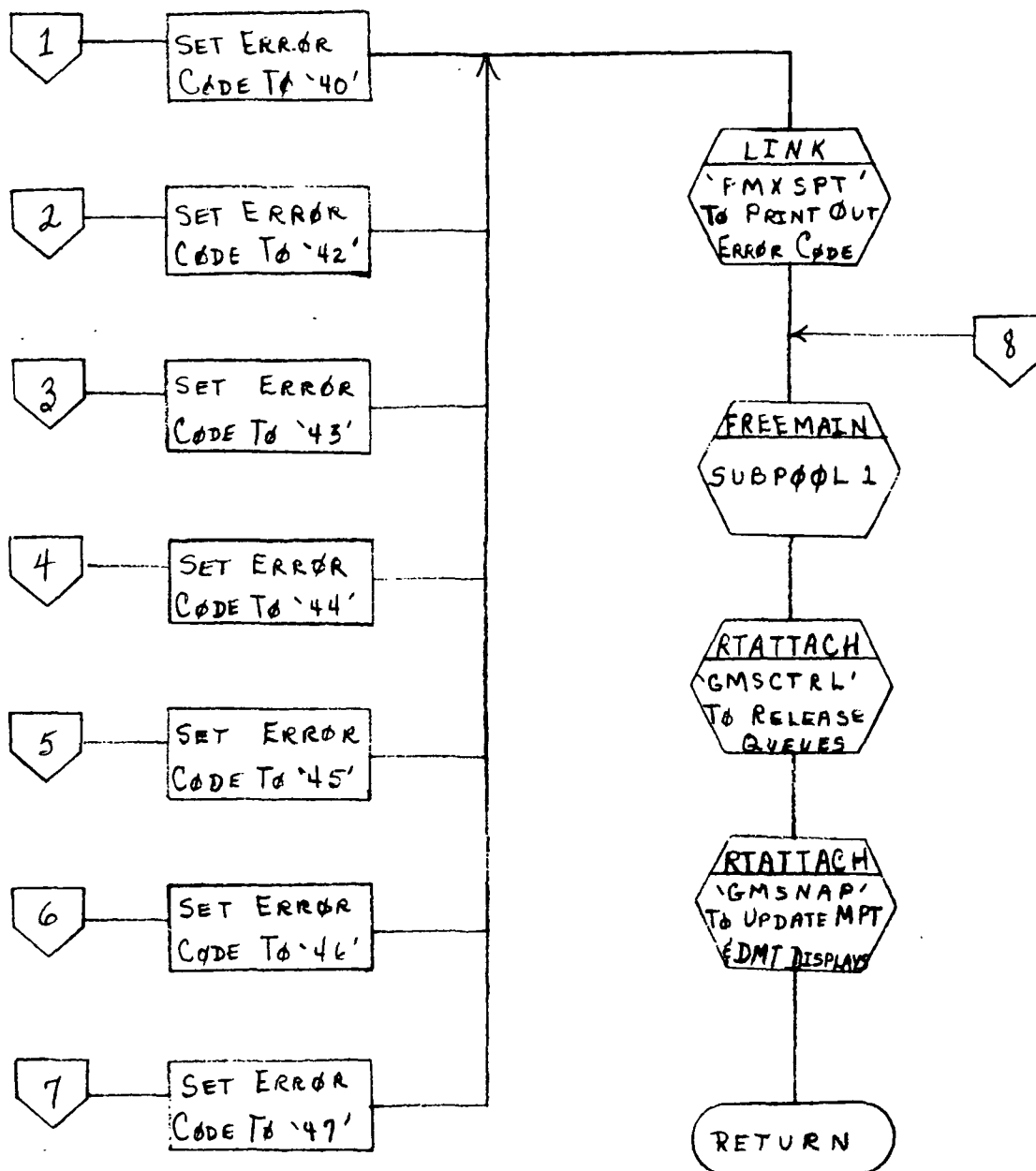


Figure PMMFUD-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

2. PMMGPM
 DATE 11/21/67 (206)
 REV
 PAGE 1 of 34

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

- a. Symbol - PMMGPM
- b. Program Name - General Purpose Maneuver Load Module
- c. Programming Group - Rendezvous Planning
- d. Programmers - J. W. Loyde, T. M. Hickey, E. M. Kramer
- e. Language - Assembler, FORTRAN IV
- f. Data Tables Used - PZGPMDIS, PZGPMELM, GZGENCSN
- g. MED's Used - K20, K21, K22, K23, K24
- h. Writeup Completion Date - 10/27/67
- i. Optional Inputs - Table PMMGPM-1
- j. System Parameters - Table PMMGPM-2
- k. Parameter List - Table PMMGPM-3
- l. Functional Flowchart (PCGPMP) - PMMGPM-1
- m. Mathematical Flowchart (PCGPMP) - PMMGPM-2

FUNCTION

PMMGPM computes any one of several types of general purpose maneuvers (as specified by optional inputs) to provide for display a table of maneuver execution parameters and a table of trajectory elements associated with the maneuver.

USAGE

- a. Calling Sequence - XCTL
- b. Inputs - See Parameter List, Table PMMGPM-3.
- c. Outputs and Exit - PZGPMELM, PZGPMDIS. Exit via RETURN.
- d. Other Programs Used - LINK, PMMAEG, PMXSPT, DTREAD, DTRITE, DEXP, DSIN, DCOS, DSQRT, DATAN2, DARSIN, DARCOS, GIMAPD
- e. Data Restrictions and Limits - See documentation of the K20, K21, K22, K23, and K24 MED's. Also see METHOD below.
- f. Error - One or more of the following messages may result from the computation of a GPM.

PMMGPM - GOOD GPM MANEUVER.

PCGPMP - ILLEGAL REQUEST, MANEUVER CANNOT BE PERFORMED IN THIS ORBIT.
 PROCESSING HALTED.

PCGPMP - ILLEGAL REQUEST, MANEUVER CANNOT BE PERFORMED AT THIS POINT IN
 THE ORBIT. PROCESSING HALTED.

PCGPMP - AEG FAILED TO CONVERGE. PROCESSING CONTINUED.

PCGPMP - UNRECOVERABLE AEG ERROR. PROCESSING HALTED.

PCGPCR - AEG FAILED TO CONVERGE. PROCESSING CONTINUED.

PCGPCR - UNRECOVERABLE AEG ERROR. PROCESSING HALTED.

PCGPCR - CANNOT FIND DESIRED HEIGHT. (SMALL E CASE) PROCESSING
 CONTINUED.

PCGPCR - FAILED TO CONVERGE ON DESIRED RADIUS. PROCESSING CONTINUED.

Approval	Approval
Merritt E. Jones	A. Anicatti

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMGPM
DATE 11/21/67 (20)
REV
PAGE 2 of 34

PCGPLN - AEG FAILED TO CONVERGE. PROCESSING CONTINUED.
PCGPLN - UNRECOVERABLE AEG ERROR. PROCESSING HALTED.
PCGPLN - FAILED TO CONVERGE ON TIME OF ARRIVAL AT GIVEN LONGITUDE.
PROCESSING CONTINUED.

g. Other - There are seven CSECTS that comprise PMMGPM. A list and functional description follows:

1. PCMGPM - In addition to being the entry point, this program arranges the MED inputs, fetches the system parameters, and queues the math program (PCGPMP) and display program.
2. PCGPMP - This is the math program and contains the equations necessary for computations of GPM's. See METHOD.
3. PCGFCI - This CSECT computes the Flight Controller Special Inputs options and is inherently part of PCGPMP. It is a separate CSECT due to early FORTRAN Compiler restrictions. See METHOD.
4. PCGPLN - This program computes time of arrival at a desired longitude. The method used is as documented in AMGPLO.
5. PCGPCR - This program computes time of arrival at a desired height. The method is as documented in MLTCIR.
6. PCBLAK - This program sets up the data table names.
7. GMAPD - This program determines apogee/perigee information and is documented in GIMAPD.

METHOD

To assist in describing the method used in PCGPMP and PCGPFI, an additional flowchart has been included, which shows the mathematical equations used in the formulation. To describe the method used, it is first necessary to discuss the logic used to arrive at a maneuver point. There are three general cases: (1) the maneuver is to be done when the elements are input to the GPM at that time which is the maneuver point; (2) the maneuver is to be performed at a longitude (Earth-fixed), altitude, apogee, perigee, or equatorial crossing (in which case, the elements are input at a threshold time, and the first upcoming maneuver point is chosen by using auxiliary routines such as GIMAPD, PCGPLN, PCGPCR, and the AEG); (3) the maneuver is of the optimum type (in which case, the elements are input at a threshold time, and the upcoming maneuver point is chosen by the GPM).

After determining the maneuver point, the elements at this point are used to compute the display quantities: height, latitude, and longitude of the maneuver. Following these computations, the appropriate formulation is executed to determine the orbital elements after the maneuver. For all maneuvers except the flight controller inputs, the ΔV , pitch, and yaw are computed next based on the elements before and after the maneuver. These quantities are input for the flight controller input maneuvers. The ΔT of burn duration is approximated using SPS thruster information and neglecting ullage and tail-off. Finally, the post-maneuver quantities are calculated based on the orbital elements after the maneuver.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMGPM

DATE 11/21/67 (206)

REV

PAGE 3 of 34

More specific information about the methods used in each type of maneuver is given below.

- a. Plane Change - It should be emphasized that this is a plane change and not an inclination change. The orbital plane is rotated through the input wedge angle about the position vector maintaining the original velocity. The wedge angle is considered positive counter-clockwise, as viewed by looking down the position vector.
- b. Circularization - A first estimate is made at circularization by setting the eccentricity and the radial component of velocity to zero, and computing the appropriate velocity for a circular orbit at the maneuver point. Then the formulation for a height change is used by setting the desired height (180° around from the maneuver point) equal to the height of the maneuver point. See case d of Method.
- c. Optimum Maneuver to Change the Height of Apogee or Perigee - The optimum place at which to alter apogee height is perigee; and conversely, the optimum place for altering perigee height is apogee. An impulsive ΔV correction is computed based on the height change desired. The ΔV is applied along the velocity vector, and the elements after the maneuver are then computed based on the new velocity. The resultant apogee or perigee height is then found. This procedure is repeated until the height tolerance is met or until five iterations are made. It should be noted that if the maneuver request results in apogee and perigee being within twenty-four nautical miles of each other, the formulation for a height change (180° around) is used. This is to preclude the switching back and forth of apogee and perigee due to Keplerian approximations as the program iterates.
- d. Height Change 180° Around from the Maneuver Point - An impulsive ΔV is computed based on the desired height change. This ΔV is applied in the local vertical - local horizontal plane, and the elements after the maneuver are computed based on the new velocity. The height 180° around is found based on the new elements. This procedure is repeated as in case c, but it is insensitive to the switching of apogee and perigee. For many cases, it may not be too efficient in the application of ΔV .
- e. Optimum Nodal Shift - The method used is to change only the argument of the ascending node and leave the remaining orbital elements unchanged.* The maneuver point is computed on the basis of the

* It should be noted that this is not actually optimum (from a ΔV viewpoint) but for relatively small shifts (5° or so) it is very close to optimum and has the desirable feature of not changing the inclination.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMGPM
DATE 11/21/67
REV
PAGE 4 of 34

(206

inclination after being equal to the inclination before the maneuver. With the elements at the maneuver point, the general nodal-shift formulation is used. (See case f of Method.) It might be noted that no iteration is done on the maneuver point because it is only affected by the variation in inclination, which is small.

- f. Nodal Shift - The desired amount of nodal shift is used to compute the elements after the maneuver by a method similar to that described in case a. The shift is considered positive eastward.
- g. Optimum Maneuver to Change Apogee and Perigee Heights - There are four cases to be considered:
 - 1. Raise apogee and perigee
 - 2. Lower apogee and perigee
 - 3. Raise apogee and lower perigee
 - 4. Lower apogee and raise perigee.

The assumption made for cases 1 and 2 is that the optimum way to perform the maneuver is to thrust along the velocity vector. This assumption is theoretically correct. The maneuver point is located by assuming the flight-path angle after the maneuver is equal to the flight-path angle before the maneuver, and by solving for the necessary height of the maneuver point. The assumption made for cases 3 and 4 is to change the a and e without shifting the line of apsides. In other words, the true anomaly after the maneuver is equal to the true anomaly before the maneuver. This assumption is apparently not the theoretical minimum, but tests have shown it to be sufficiently close.

Because of the variations in a and e , the maneuver point is iterated upon in all cases. After the maneuver point is located, the general formulation for changing apogee and perigee heights is used. (See case h of Method.) Before performing any computations, a check is made to see if the maneuver can be performed (the present and desired orbits must have an intersection), and the request is rejected if illegal.

- h. Maneuver to Change Apogee and Perigee Heights - After reaching the requested maneuver point, the elements are computed based on the desired apogee and perigee heights. The apsidal heights actually produced by the elements are then computed. Corrections are made based on the last apsidal values obtained, and this procedure is repeated. The possibility exists that the maneuver cannot be performed at the requested maneuver point, but could be at some other point in the orbit. The height of the requested maneuver point must be between the requested apogee and perigee heights, or the maneuver cannot be computed.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMGPM
DATE 11/21/67 (206)
REV
PAGE 5 of 34

- i. Maneuver to Shift the Node and Change the Height 180° Around from the Maneuver Point - This maneuver is computed using the formulation for cases d and f of Method. Once at the desired maneuver point, case f is used to compute the nodal shift. The elements after the nodal shift are then used as the elements before the height change, for which case d is used. The elements after the final step show the effect of the nodal shift and height change.
- j. Flight Controller Inputs - The ΔV , pitch, yaw, and maneuver point are input. The ΔV is resolved into three components: the horizontal in-plane, horizontal out-of-plane, and radial. The radial and horizontal in-plane components (there is no out-of-plane) of the original orbit at the maneuver point are computed, and the new velocity is calculated. The orbital elements after the maneuver can then be computed from the new velocity components.
- k. Maneuver to Shift the Line of Apsides and Obtain a Height 180° Around from the Maneuver Point - This maneuver results in an orbit in which the maneuver point is one apsis, and the requested height is the other apsis. A first estimate to the required elements is made, and then the formulation for d is used. It should be noted that this maneuver (AHL, AHT, AHH) applies only for EORP.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMGPM
DATE 11/21/67 (206
REV
PAGE 6 of 34

DTRÖUTE, RTQCNTRL, RTATTACH Macros

Macros - RTATTACH TASK = GTISPLAY, EP = GMSNAP, ID = 3, PARAM = (IFN, 4, MINUS, 4),
DPRIOR = PRIOR75

Purpose - Update the GPM Display

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2.PMMGPM
 DATE 11/21/67 (206)
 REV
 PAGE 7 of 34

Maneuver Code	Item 4	Item 5	Item 6	Item 7	Decoding MED
11 PCE			ΔW^*		K20
12 PCL		λ_D	ΔW		
13 PCT			ΔW		
14 PCH	H_D		ΔW		
15 CRL		λ_D			
16 CRH	H_D				
17 HAO	ΔH_D				
18 HPO	ΔH_D				
19 HOL	ΔH_D	λ_D			
20 HOT	ΔH_D				
21 NSO			Δh		
22 NSL		λ_D	Δh		
23 NST			Δh		
24 NSH	H_D		Δh		
25 HBO	H_A	H_P			K22
26 HBL	H_A	H_P		λ_D	
27 HBT	H_A	H_P			
28 HBH	H_A	H_P	H_D		
29 NHL	ΔH_D	λ_D	Δh		K23
30 NHT	ΔH_D		Δh		K23
31 FCE	ΔV	P	Y		K21
32 FCL	ΔV	P	Y	λ_D	
33 FCT	ΔV	P	Y		
34 FCH	ΔV	P	Y	H_D	
35 FCA	ΔV	P	Y		
36 FCP	ΔV	P	Y		
37 AHL		λ_D	H		K24
38 AHT			H		K24
39 AHH	H_D		H		K24

* ΔW - Desired wedge angle change

λ_D - Desired maneuver longitude

H_D - Desired maneuver height

ΔH_D - Desired change in height

Δh - Desired change in the ascending node

H_A - Desired apogee height after the maneuver

H_P - Desired perigee height after the maneuver

ΔV - Input incremental velocity magnitude of the maneuver

P - Input pitch of the maneuver

Y - Input yaw of the maneuver

H - Desired height 180° around from the maneuver point



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMGPM
 DATE 11/21/67
 REV
 PAGE 8 of 34

(206)

Table PMMGPM-2. System Parameters

Symbol in Program	Symbol in Master Tape	Type *	Units	Comments
TWOPI	MCMTPI	D	radians	
PI	MCMPPIR	D	radians	
WE	MCERTS	D	radians hour	
XMU	MCEMUU	D	$Er^3/hr.^2$	
SQRMU	MCERMU	D	$Er^{3/2}/hr.$	
DKFACT		D		
RADE	MCECAP	D	$Er.$	
ACG	MCEACG	D	$Er/hr.^2$	
XISP	MCTSIP	D	hr.	
WDOT	MCTWLR	D	lbs./hr.	

*Type
 H - Fixed-point half word
 F - Fixed-point full word
 E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMGPM

DATE 11/21/67 (206)

REV

PAGE 9 of 34

Table PMMGPM-3. Parameter List

QUEID — NA

QNAME — NA

From — NA

To-Task — NA

Purpose — NA

EP — NA

Byte No.	Type*	Units	Value / Description / Comments
0	F		Vehicle Code
4	F		Maneuver Code
8	D	hr.	Vector time
16	D	hr.	Threshold time
24	D		Item 4 (see Table PMMGPM-1)
32	D		Item 5 (see Table PMMGPM-1)
40	D		Item 6 (see Table PMMGPM-1)
48	D		Item 7 (see Table PMMGPM-1)
56	F		Thruster - vehicle 1 (not used)
60	F		Attitude - vehicle 1 (not used)
64	F		Thruster - vehicle 2 (not used)
68	F		Attitude - vehicle 2 (not used)
72-399			
400	D	Er	a
408	D		e
416	D	rad.	i
424	D	rad.	l
432	D	rad.	g
440	D	rad.	h
448	D	hr.	Time of the above elements
456	D		K-factor (drag coefficient)
464	D	ft. ²	Area
472	D	lb.	Weight
480	D		Station ID
488	F		QUEID
492	F		

*Type
H - Fixed-point half word
F - Fixed-point full word

E - Single-precision (floating-point)
D - Double-precision (floating-point)
X - Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMGPM
 DATE 11/21/67
 REV
 PAGE 10 of 34

(206)

LOGIC FLOW

PCGPMP

1

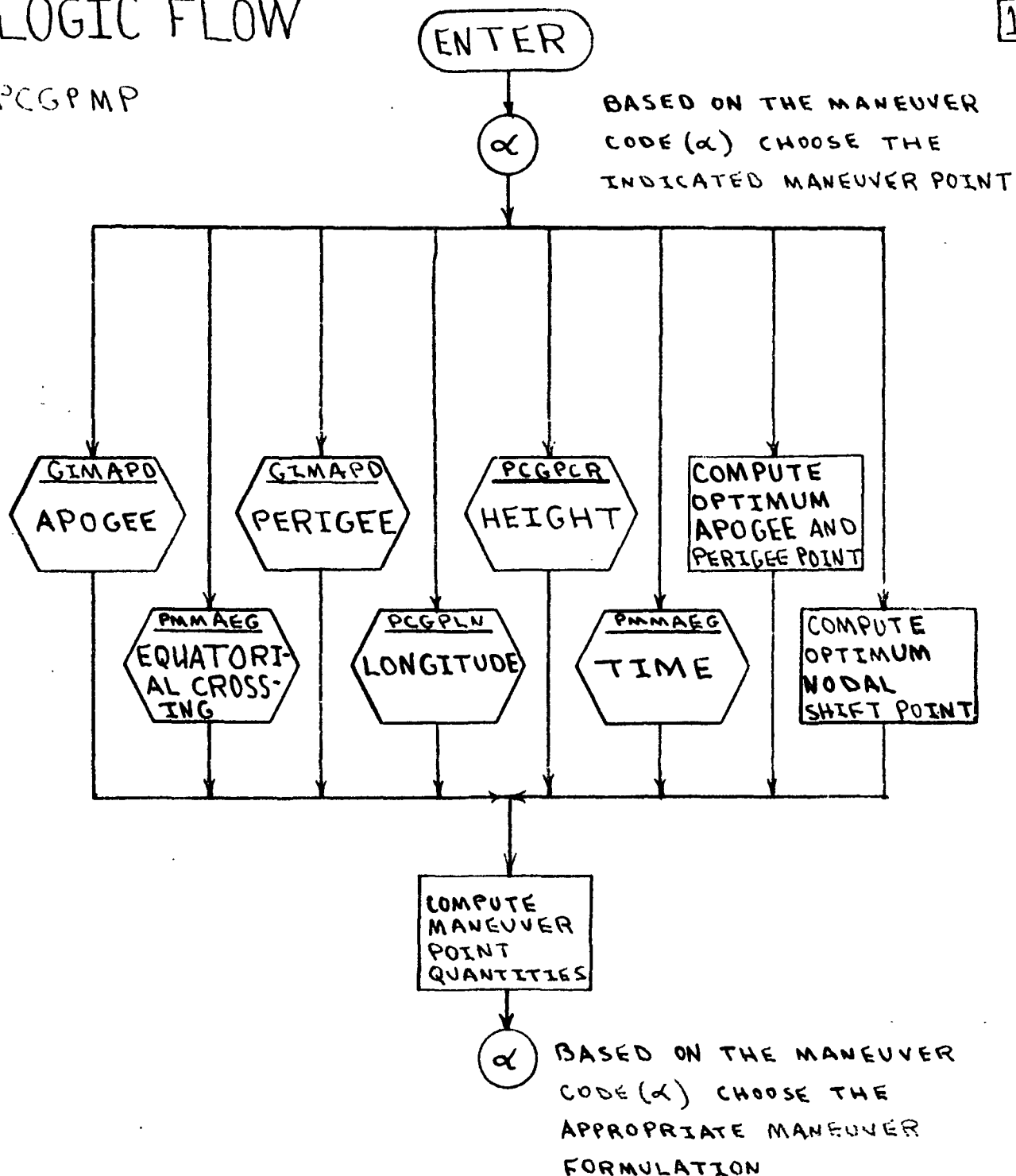


Figure PMMGPM-1. Functional Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMGPM

DATE 11/21/67 (206)

REV

PAGE 11 of 34

2

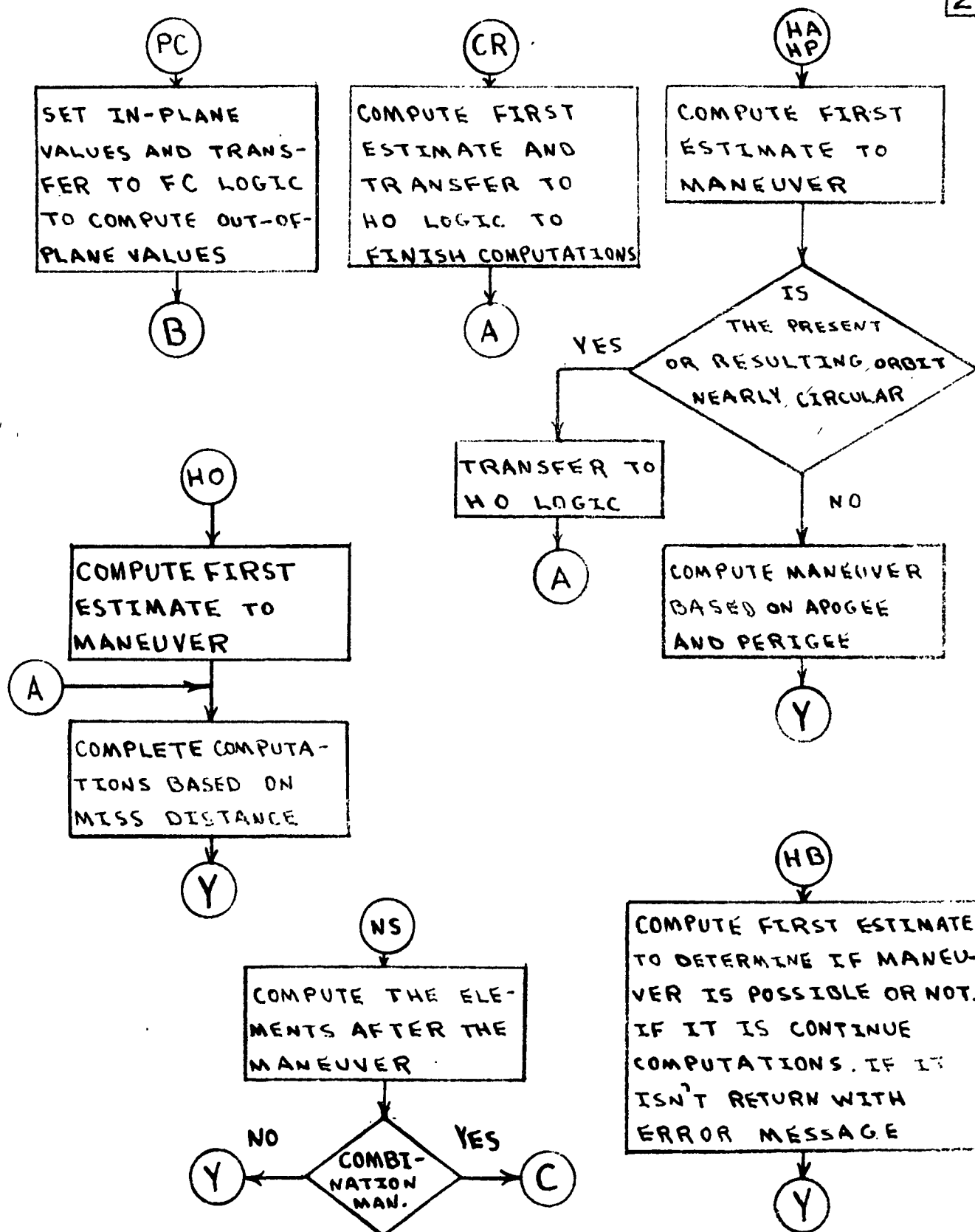


Figure PMMGPM-1. Functional Flowchart

(Sheet 2 of 3)

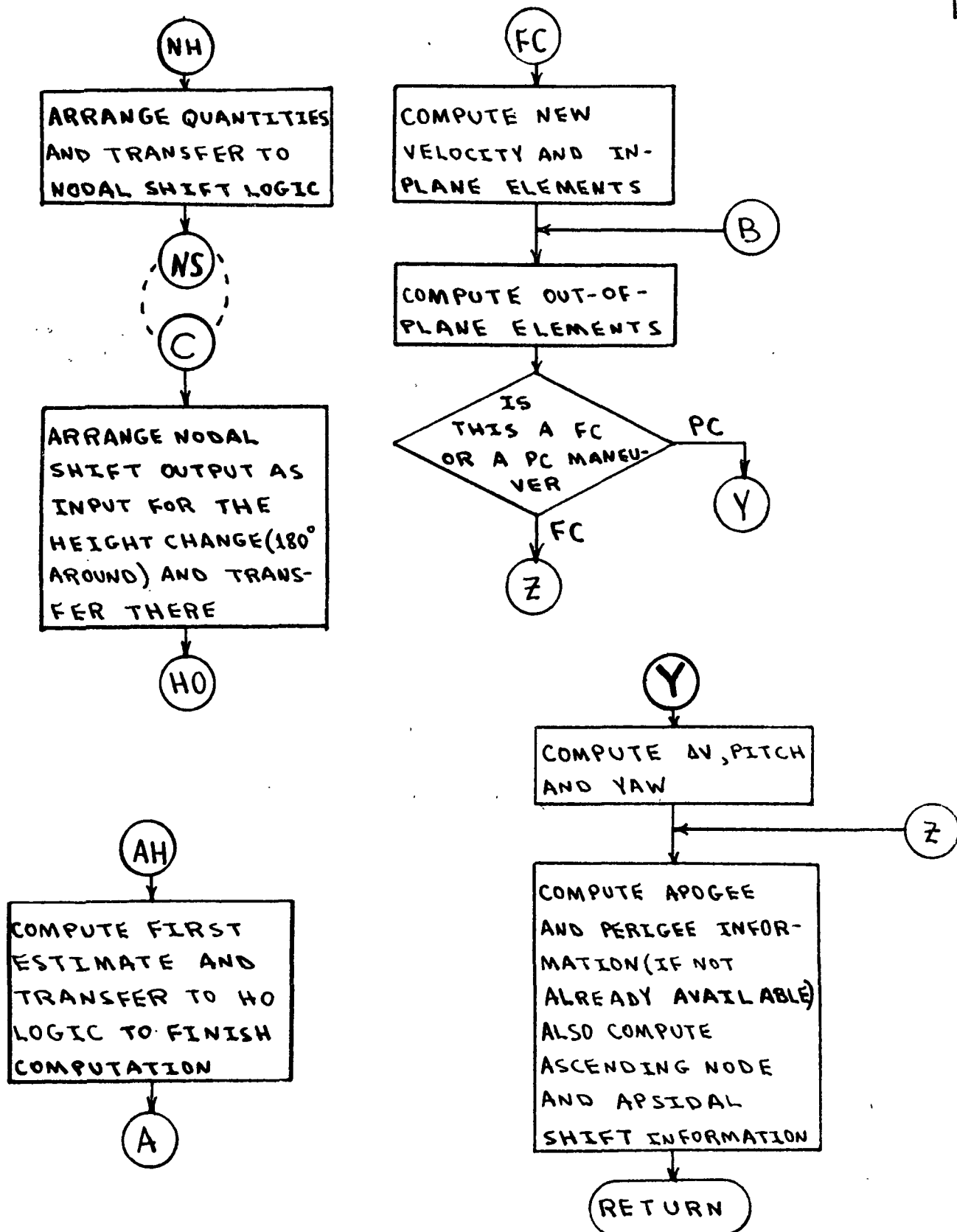


Figure PMMGPM-1. Functional Flowchart

MATH FLOW PCGPMP

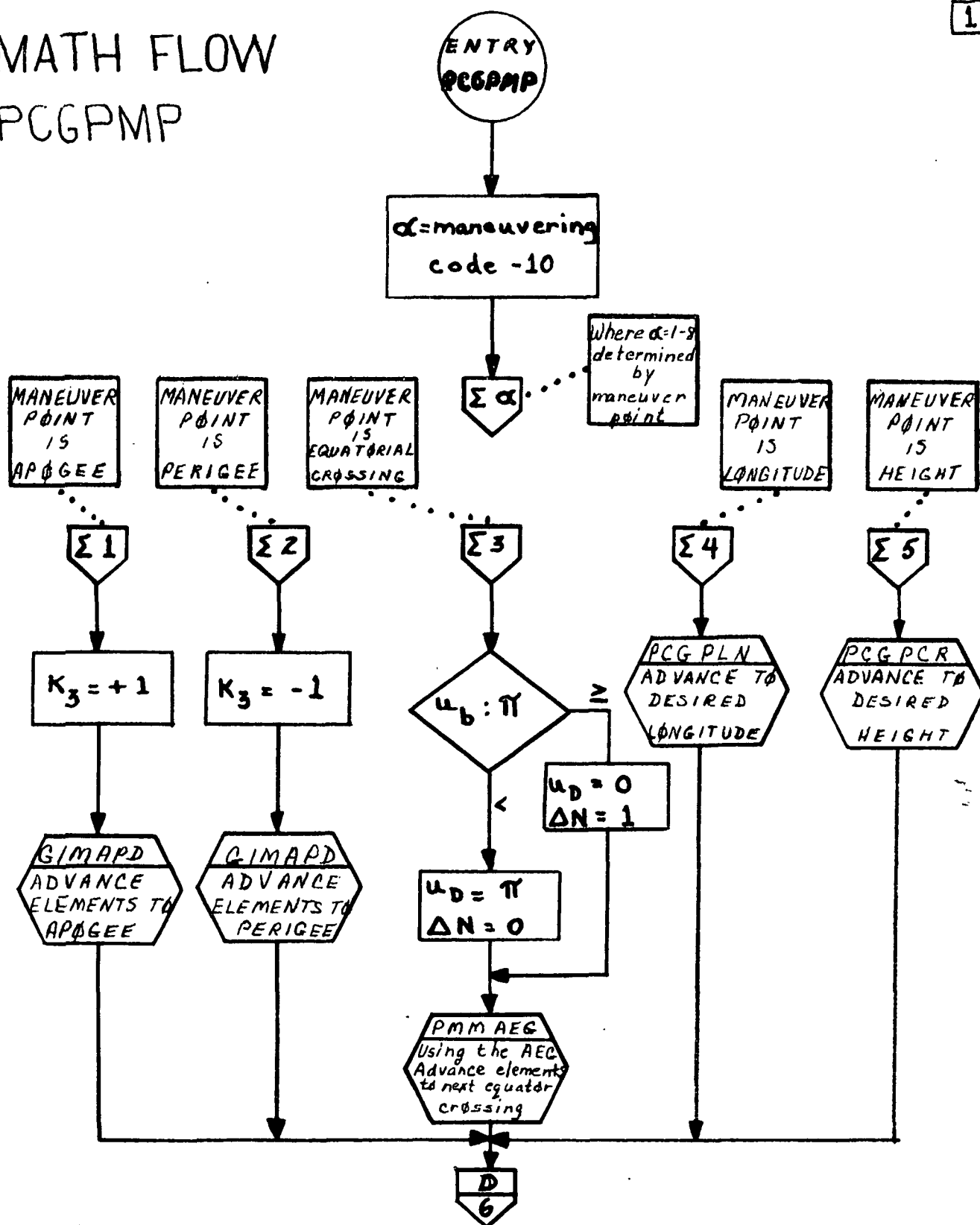


Figure PMMGPM-2. Mathematical Flowchart

2

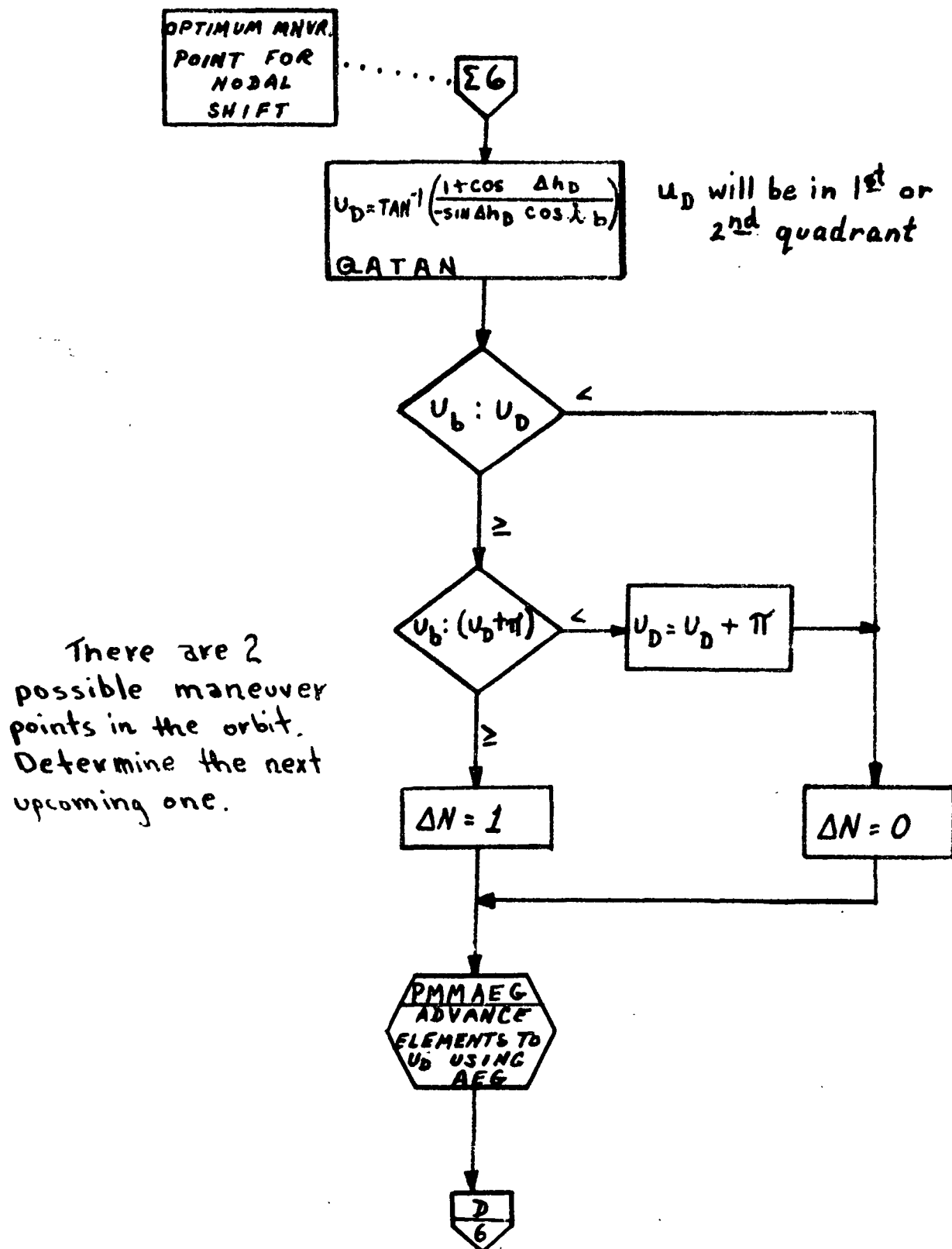


Figure PMMGPM-2. Mathematical Flowchart

3

Optimum maneuver
to obtain desired
apogee and
perigee heights

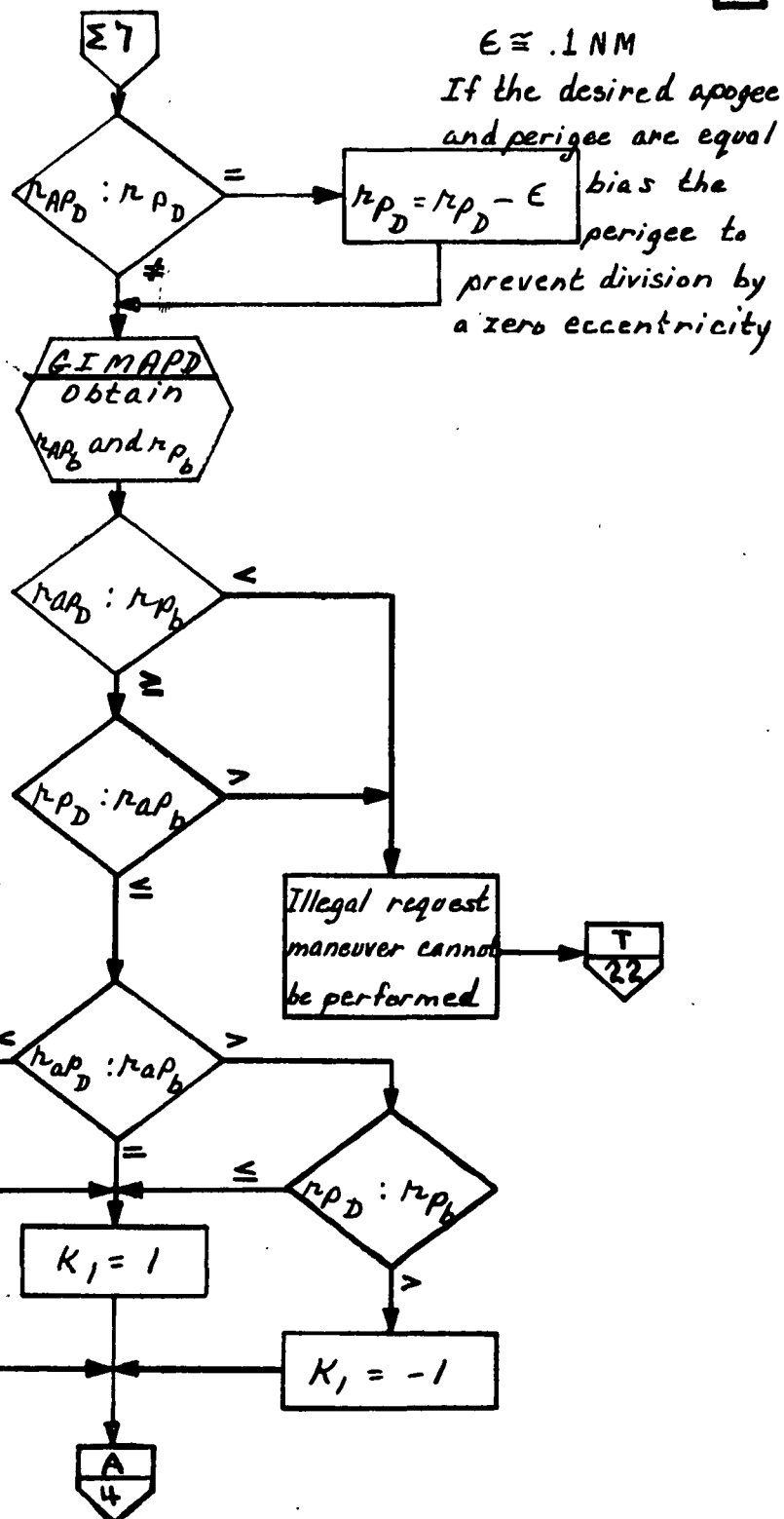


Figure PMMGPM-2. Mathematical Flowchart

a_p and e_p are the elements at perigee as returned by GIMAPD. Similarly for a_{AP} and e_{AP}

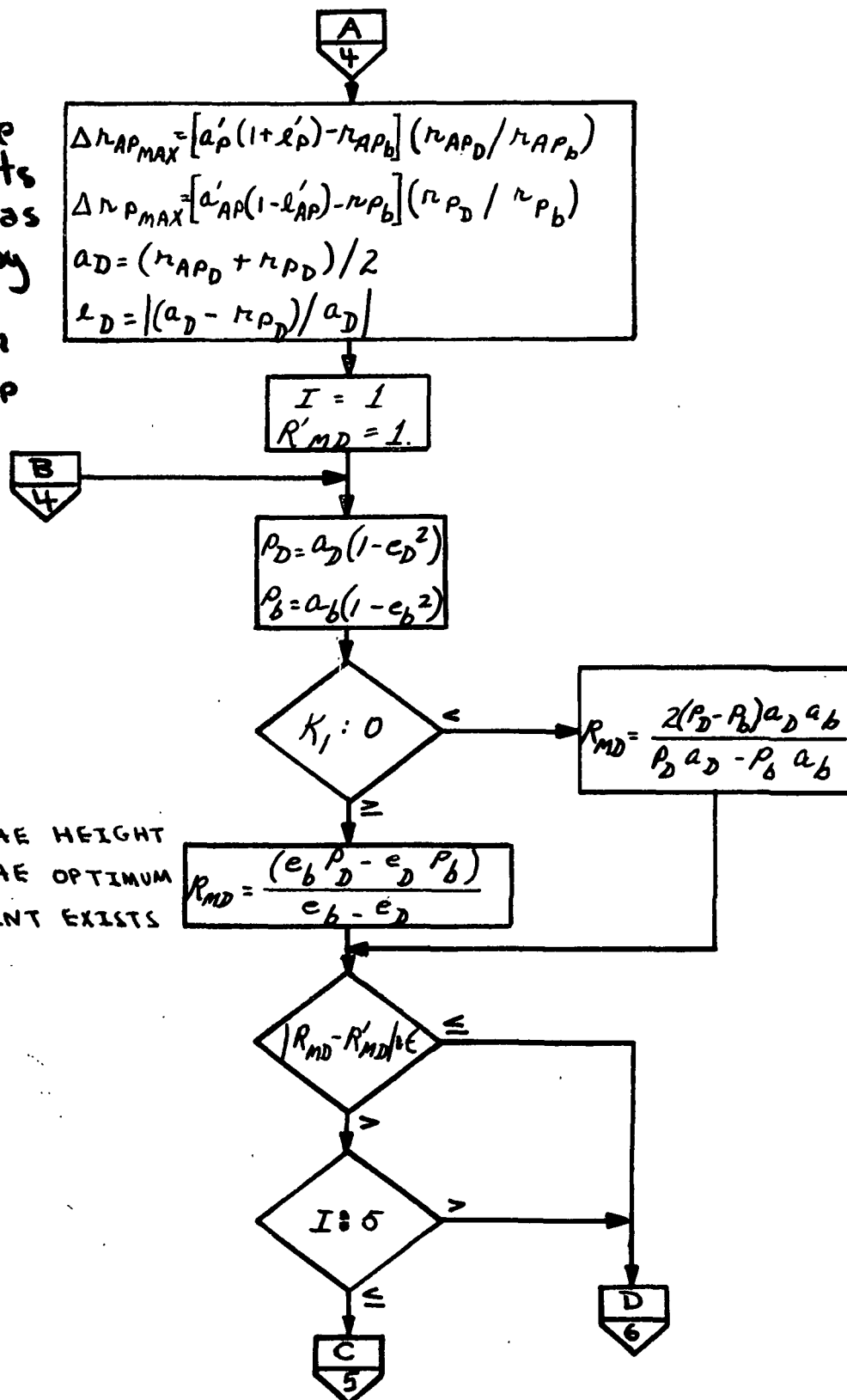


Figure PMMGPM-2. Mathematical Flowchart

5

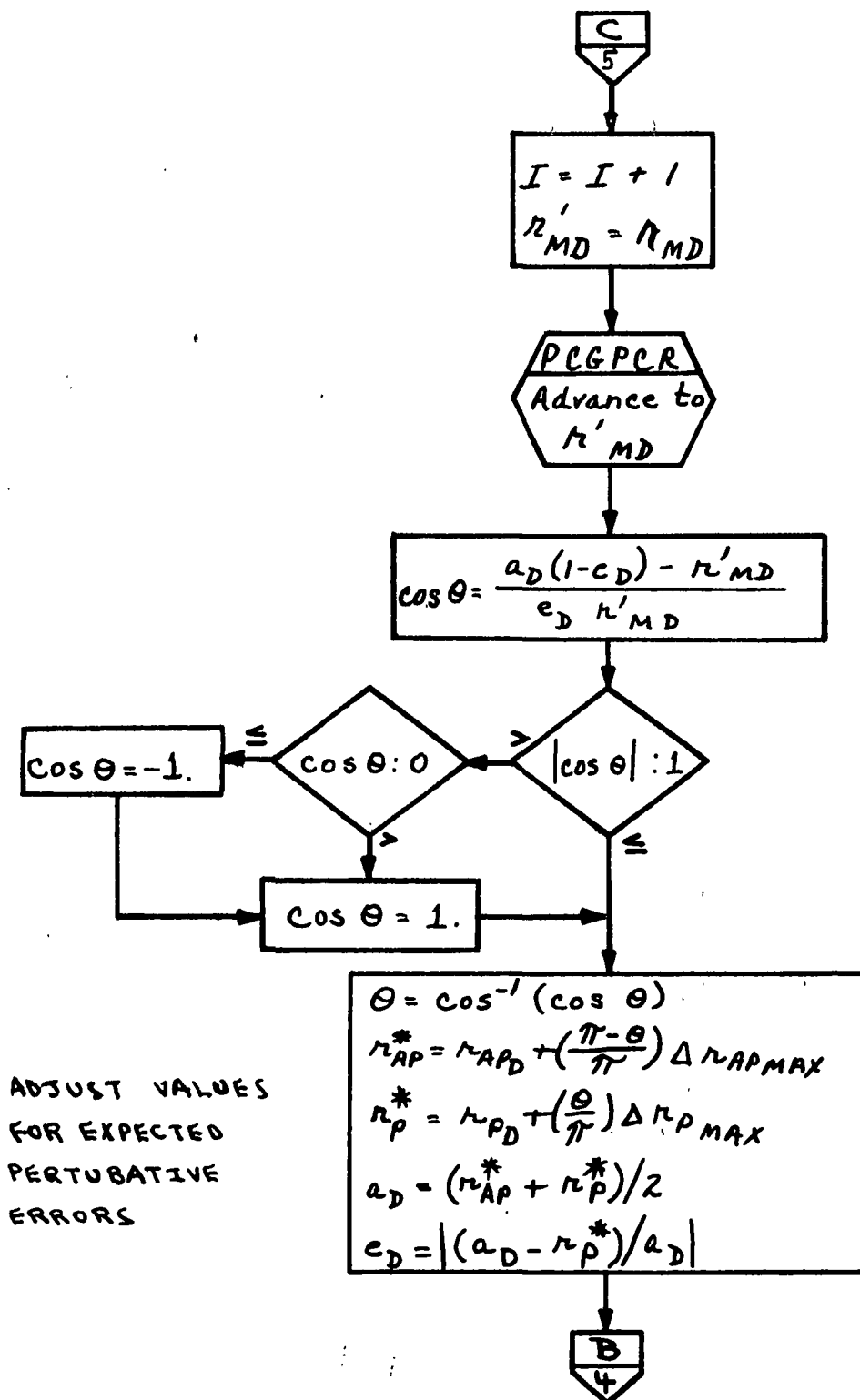


Figure PMMGPM-2. Mathematical Flowchart

6

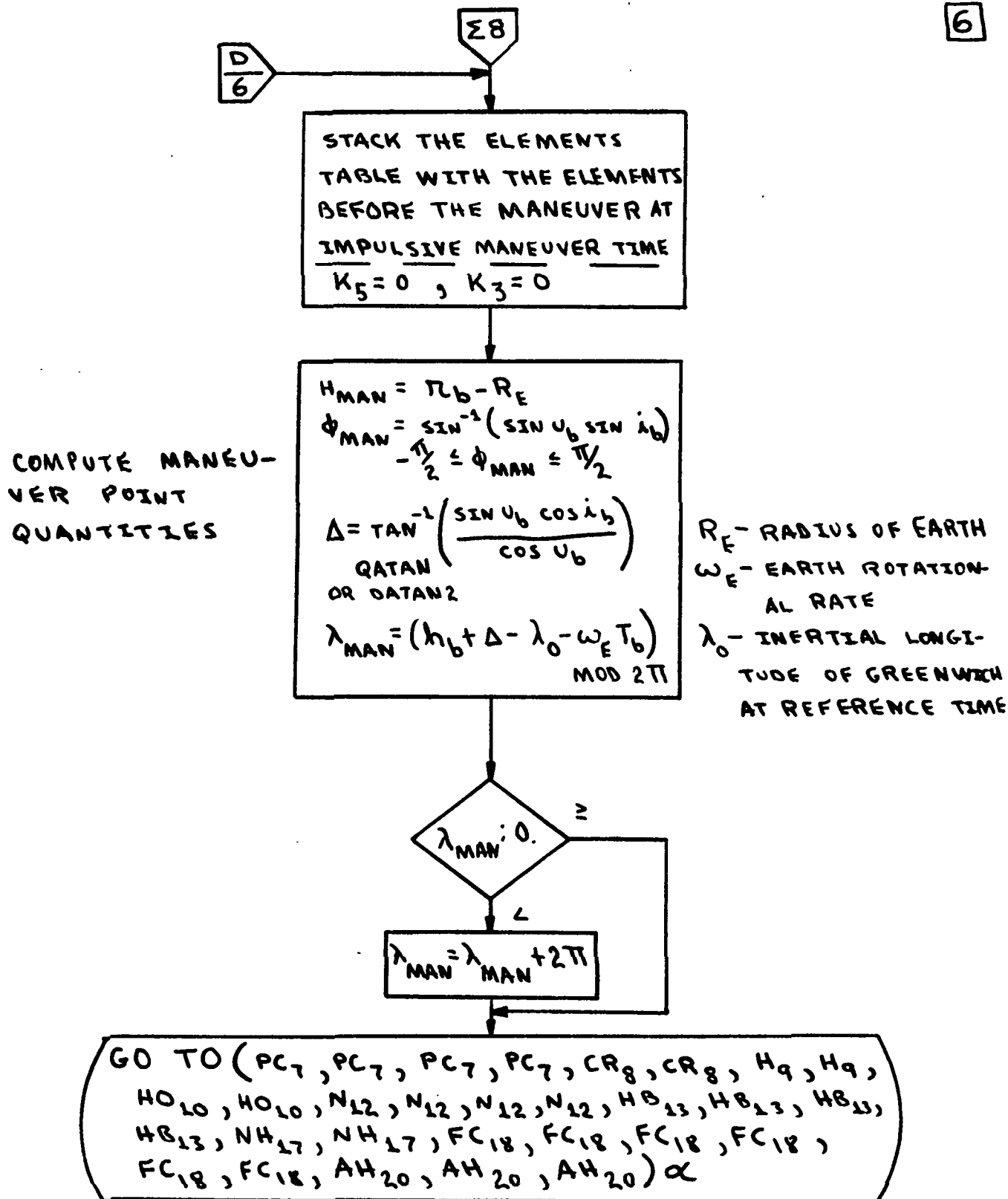
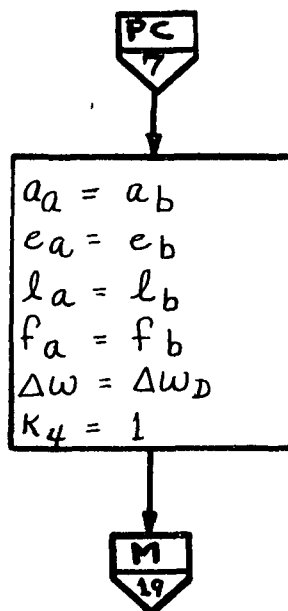


Figure PMMGPM-2. Mathematical Flowchart

7

PCE
PCL
PCT
PCH



*The other elements can
be computed using the
Flight Controller Input
Logic*

Figure PMMGPM-2. Mathematical Flowchart

(Sheet 7 of 22)

CRL
CRH

COMPUTE FIRST
ESTIMATE TO
POST-MANEUVER
QUANTITIES

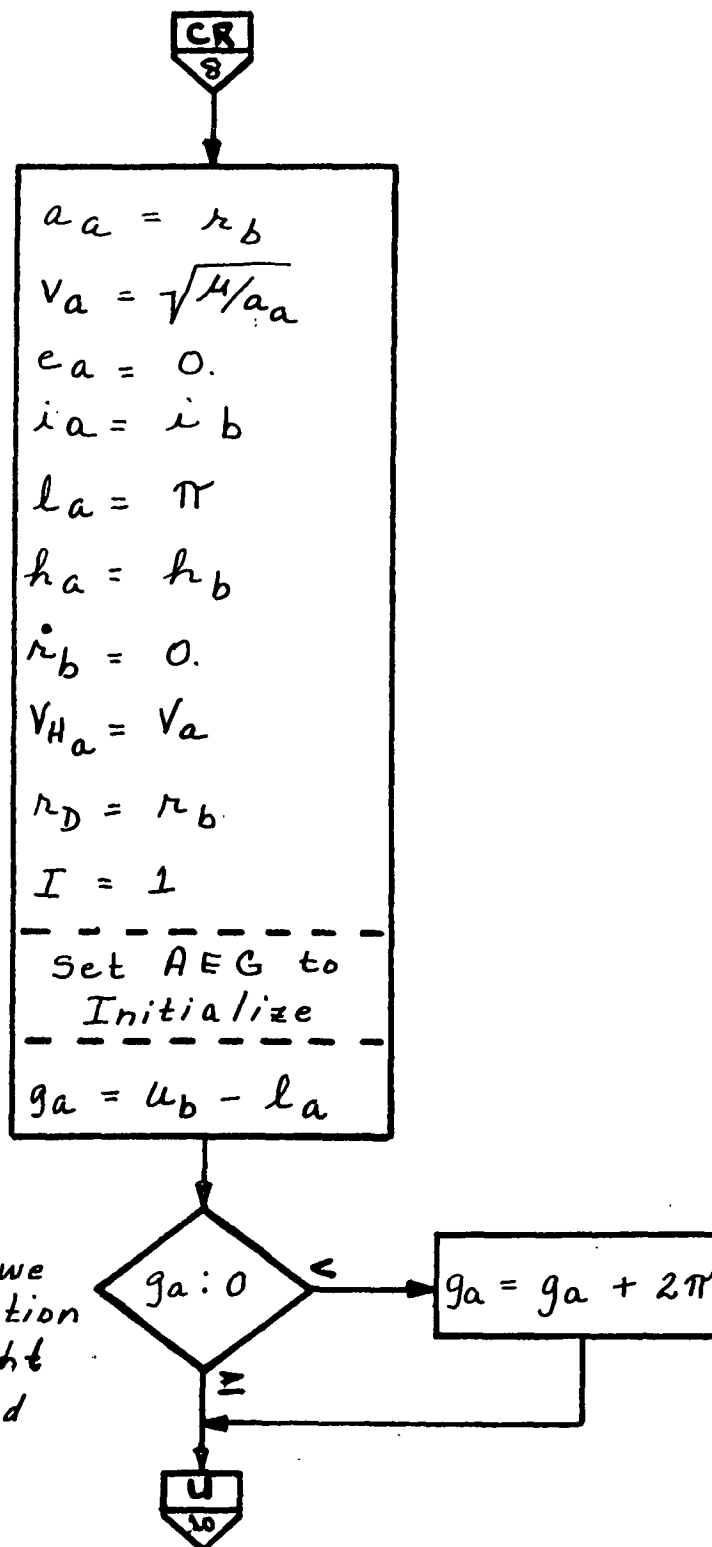


Figure PMMGPM-2. Mathematical Flowchart

9

HAO
HPO

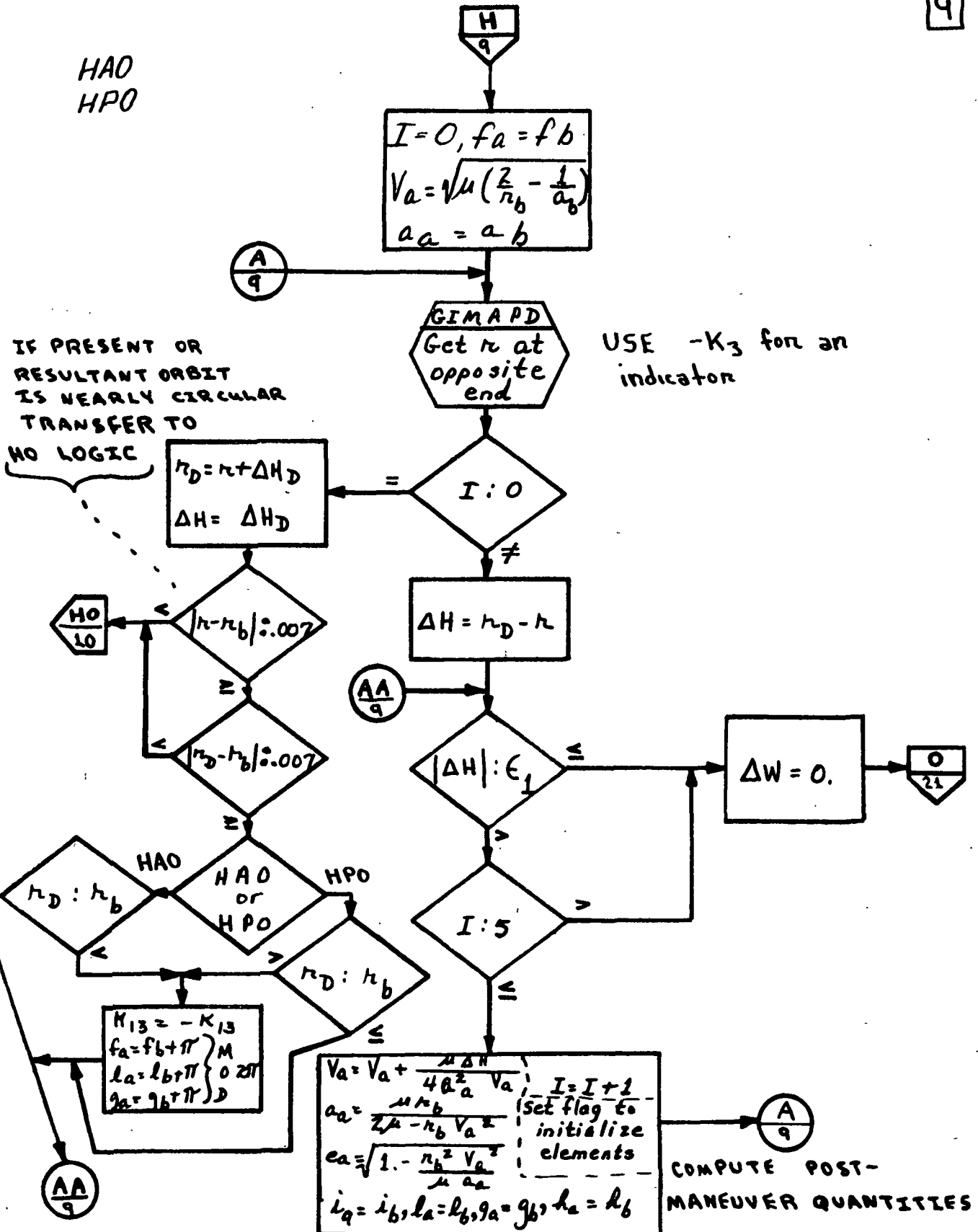
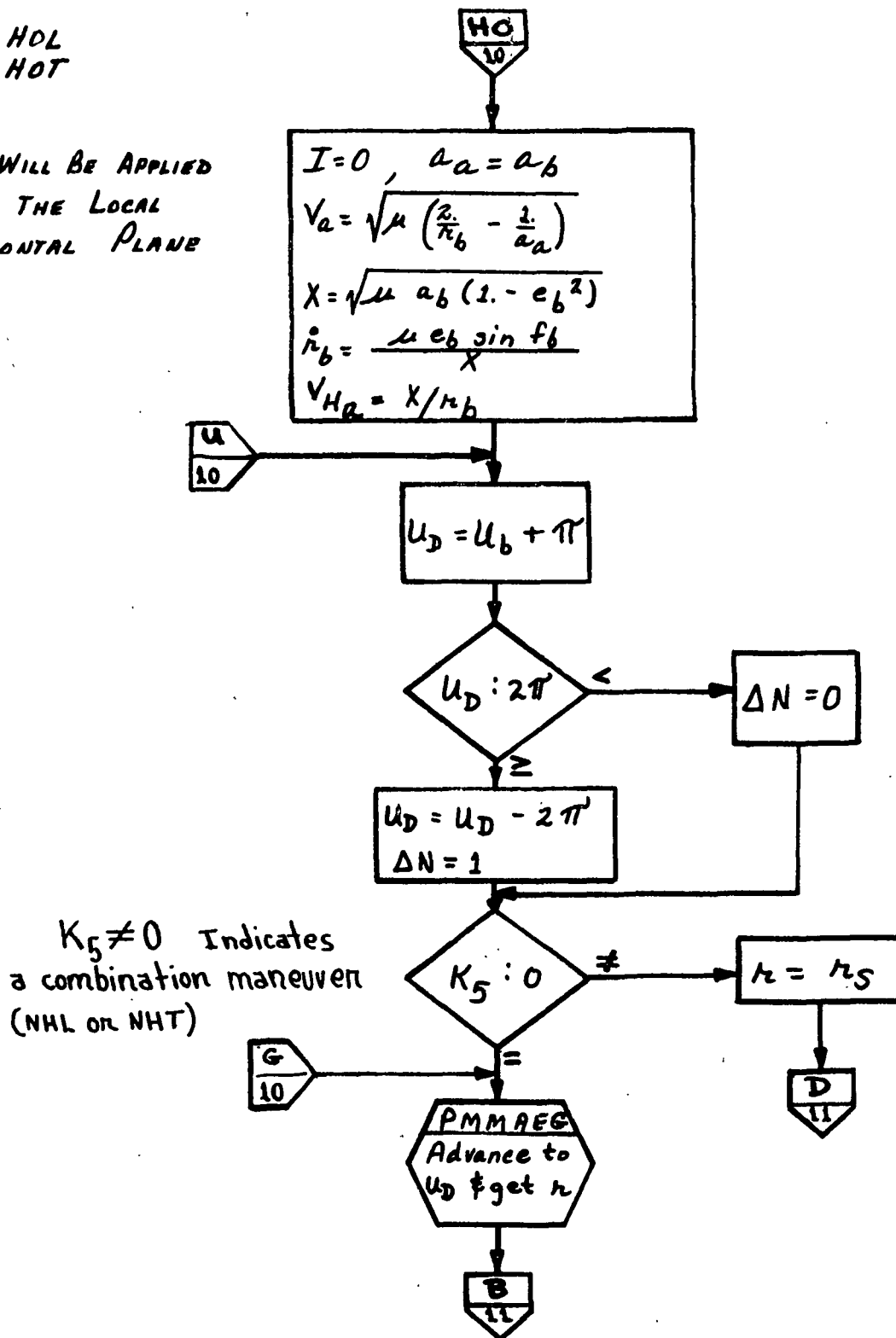


Figure PMMGPM-2. Mathematical Flowchart

10

HOL
 HOT

ΔV WILL BE APPLIED
 IN THE LOCAL
 HORIZONTAL PLANE



$K_5 \neq 0$ Indicates
 a combination maneuver
 (NHL or NHT)

Figure PMMGPM-2. Mathematical Flowchart

11

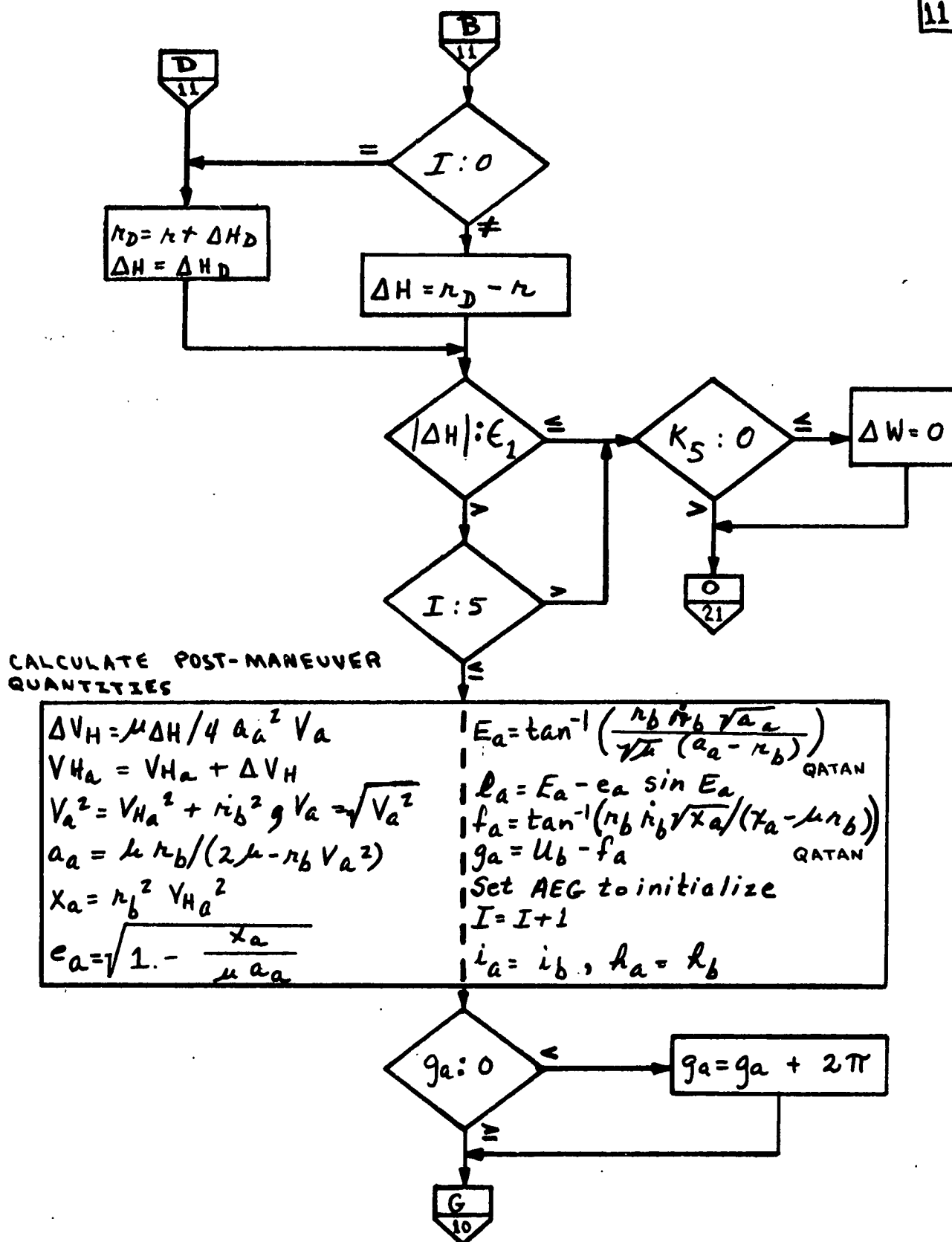


Figure PMMGPM-2. Mathematical Flowchart

NSL
NST
NSH
(NSO)

12

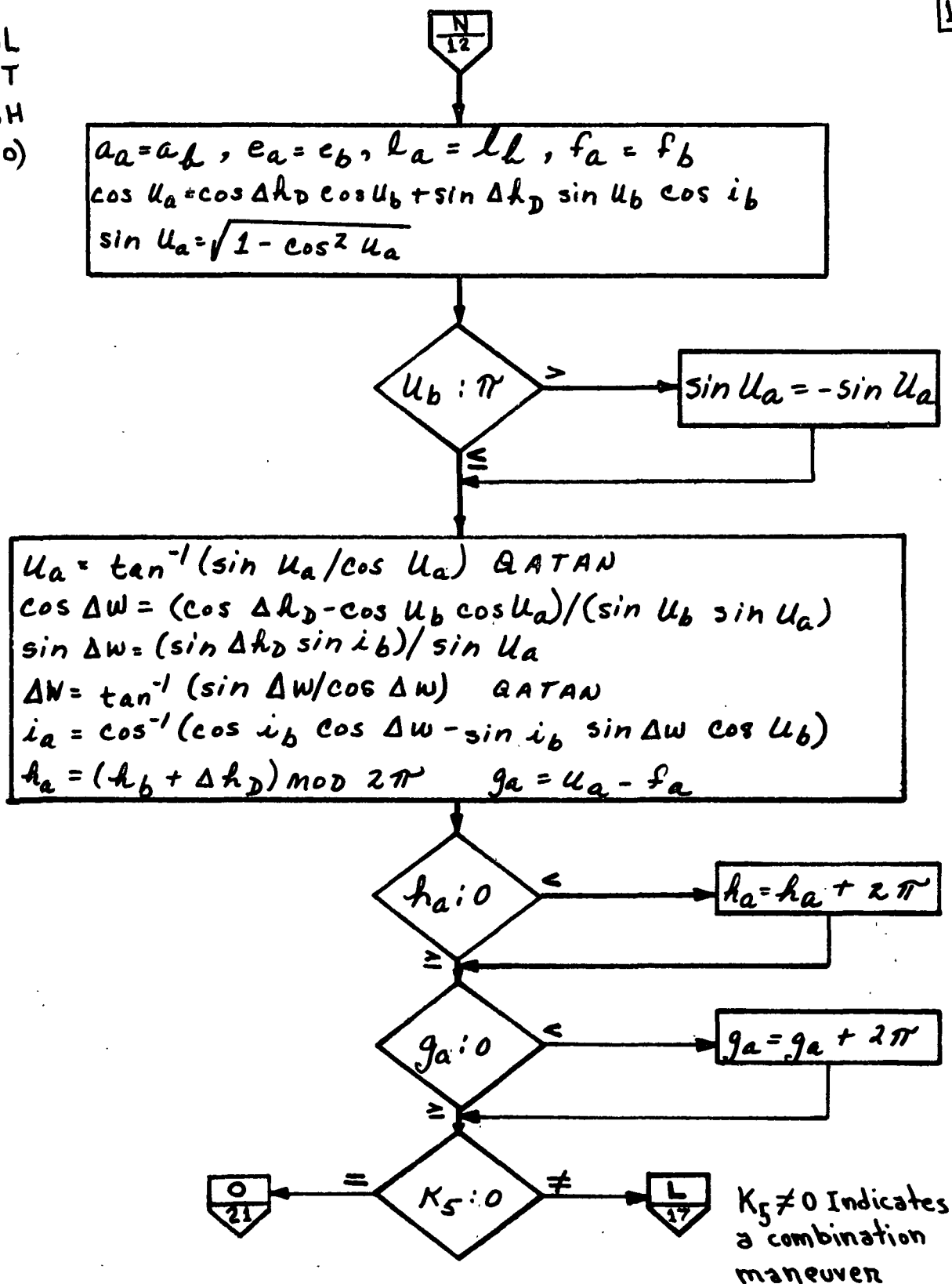


Figure PMMGPM-2. Mathematical Flowchart

14

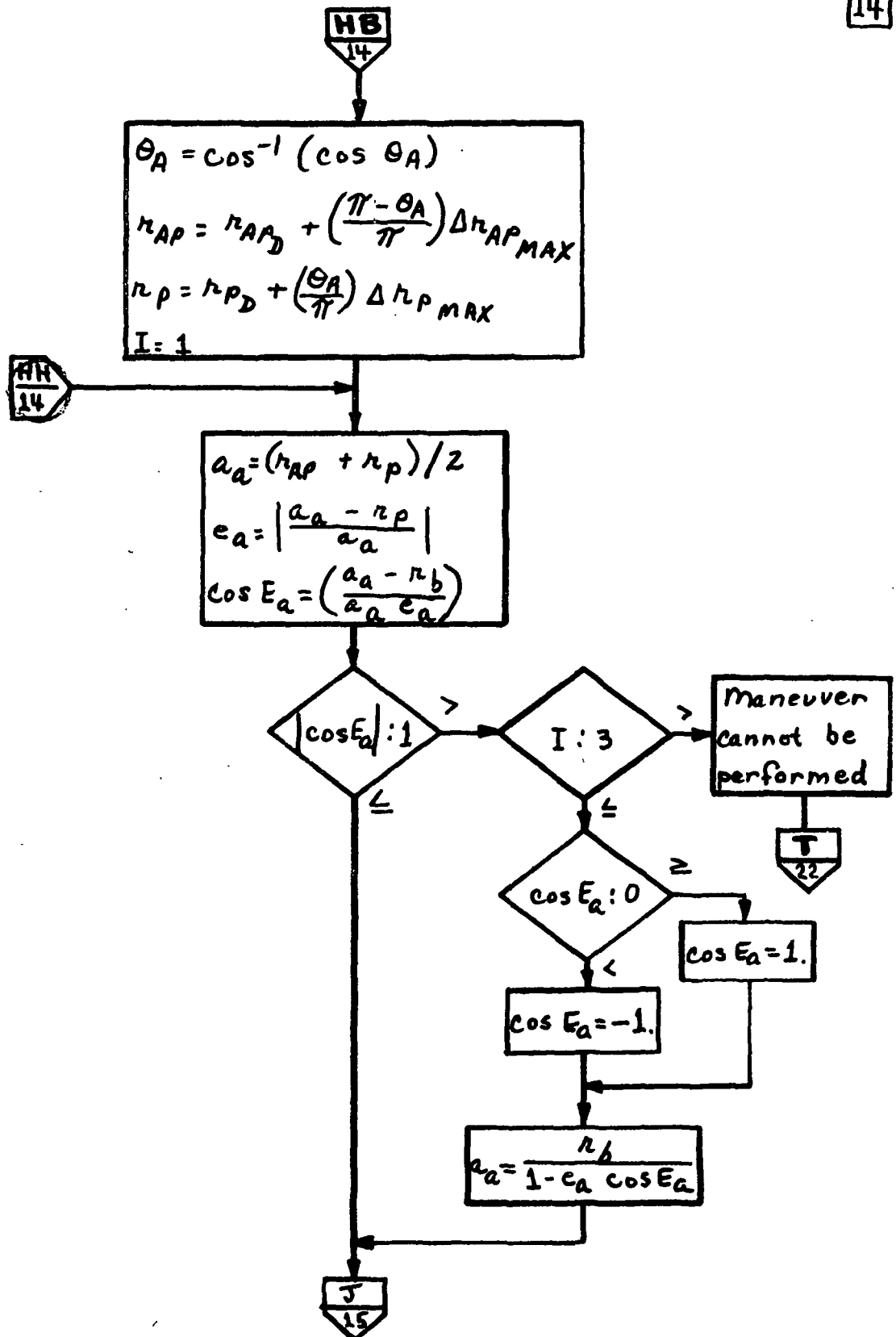


Figure PMMGPM-2. Mathematical Flowchart

15

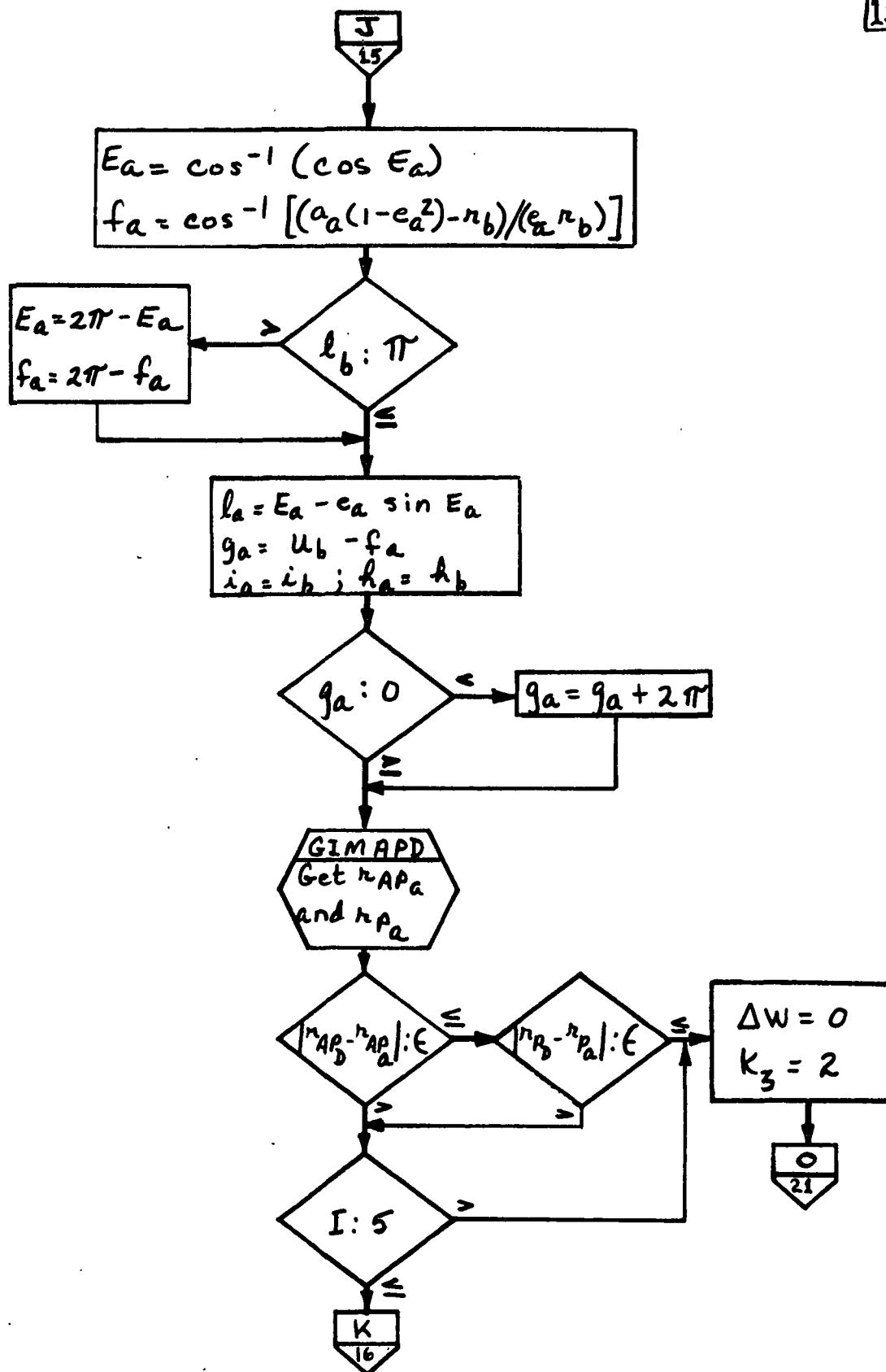


Figure PMMGPM-2. Mathematical Flowchart

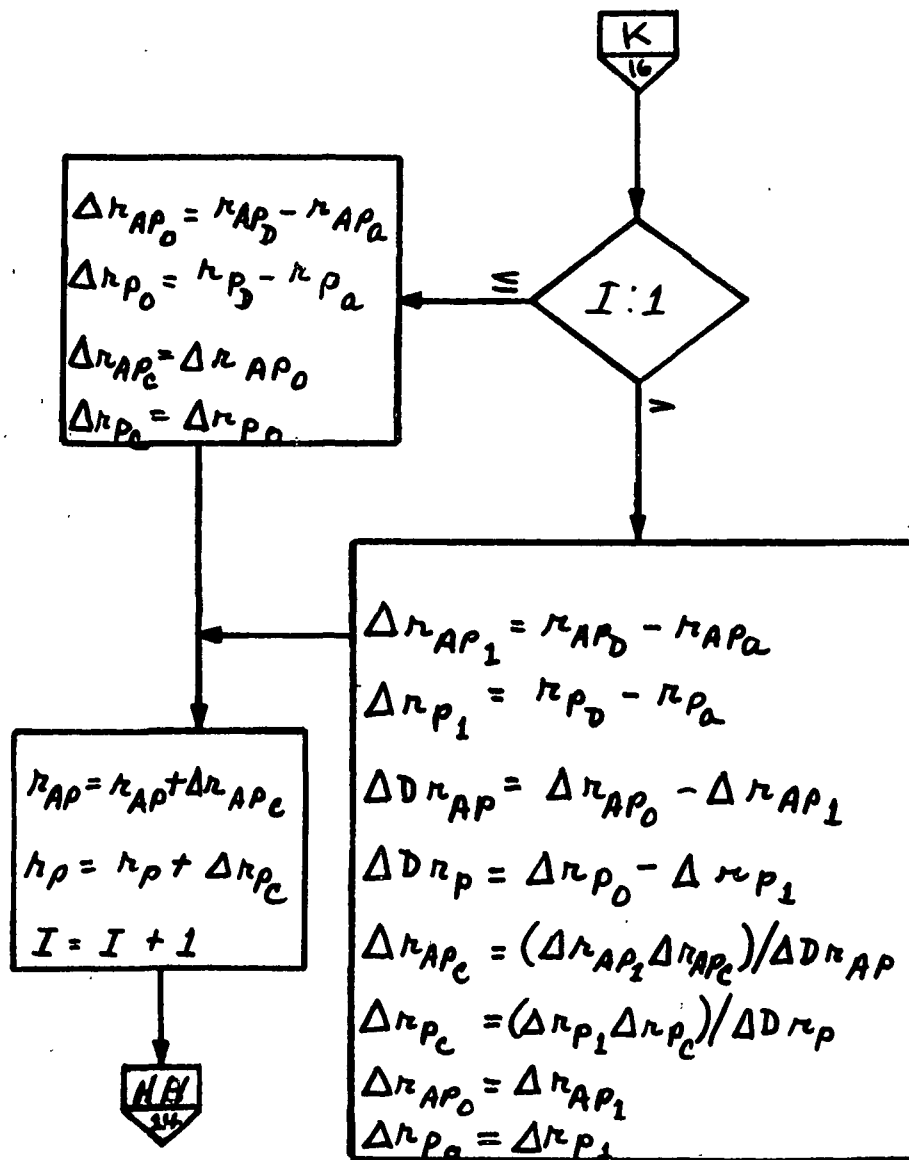
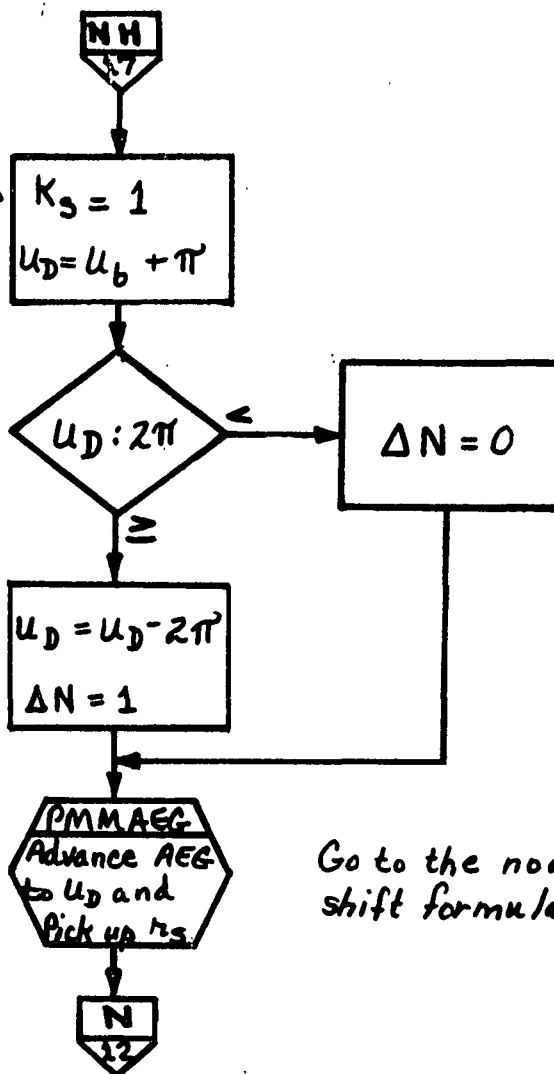


Figure PMMGPM-2. Mathematical Flowchart

17

NHL
NHT

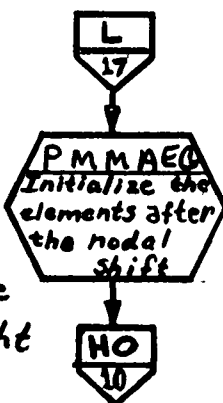
The subscript b refers to the elements at impulsive maneuver time.



Don't overlay the elements at impulsive maneuver time.

Go to the nodal shift formulation.

The elements are to be initialized at impulsive maneuver time. Then they are to be considered the elements before (from the point of view of the height maneuver).



Go to the height change formulation.

Figure PMMGPM-2. Mathematical Flowchart

18

FCE
FCL
FCT
FCH
FCA
FCP



$$\begin{aligned}\Delta V_{HIP} &= \Delta V \cos P \cos Y \\ \Delta V_{HOP} &= \Delta V \cos P \sin Y \\ \Delta \dot{r} &= \Delta V \sin P\end{aligned}$$

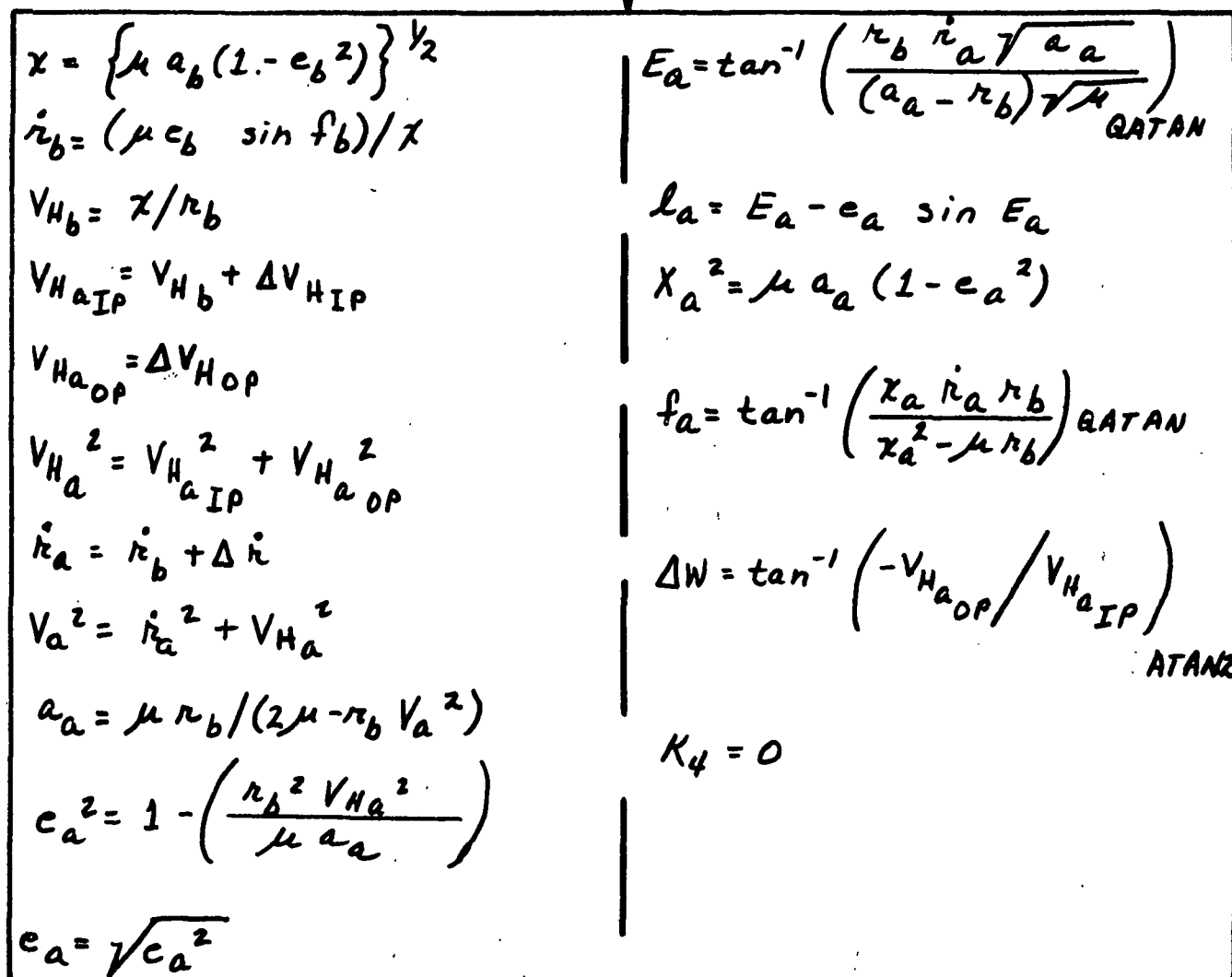


Figure PMMGPM-2. Mathematical Flowchart

19

M
19

$$i_a = \cos^{-1}(\cos i_b \cos \Delta W - \sin i_b \sin \Delta W \cos u_b)$$

$$\sin \Delta h = (\sin \Delta W \sin u_b) / \sin i_a$$

$$\cos \Delta h = \frac{\cos \Delta W - \cos i_b \cos i_a}{\sin i_b \sin i_a}$$

$$\Delta h = \tan^{-1}(\sin \Delta h / \cos \Delta h) \text{ QATAN}$$

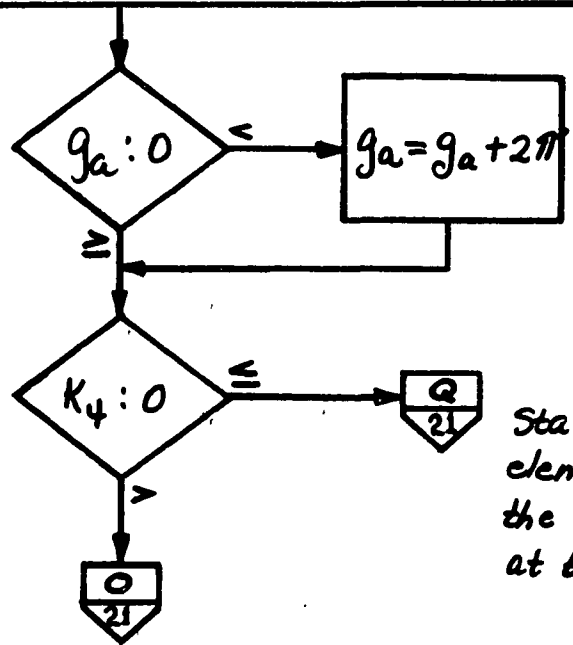
$$h_a = (h_b + \Delta h) \text{ MOD } 2\pi$$

$$\sin u_a = (\sin u_b \sin i_b) / \sin i_a$$

$$\cos u_a = \cos u_b \cos \Delta h + \sin u_b \sin \Delta h \cos i_b$$

$$u_a = \tan^{-1}(\sin u_a / \cos u_a) \text{ QATAN}$$

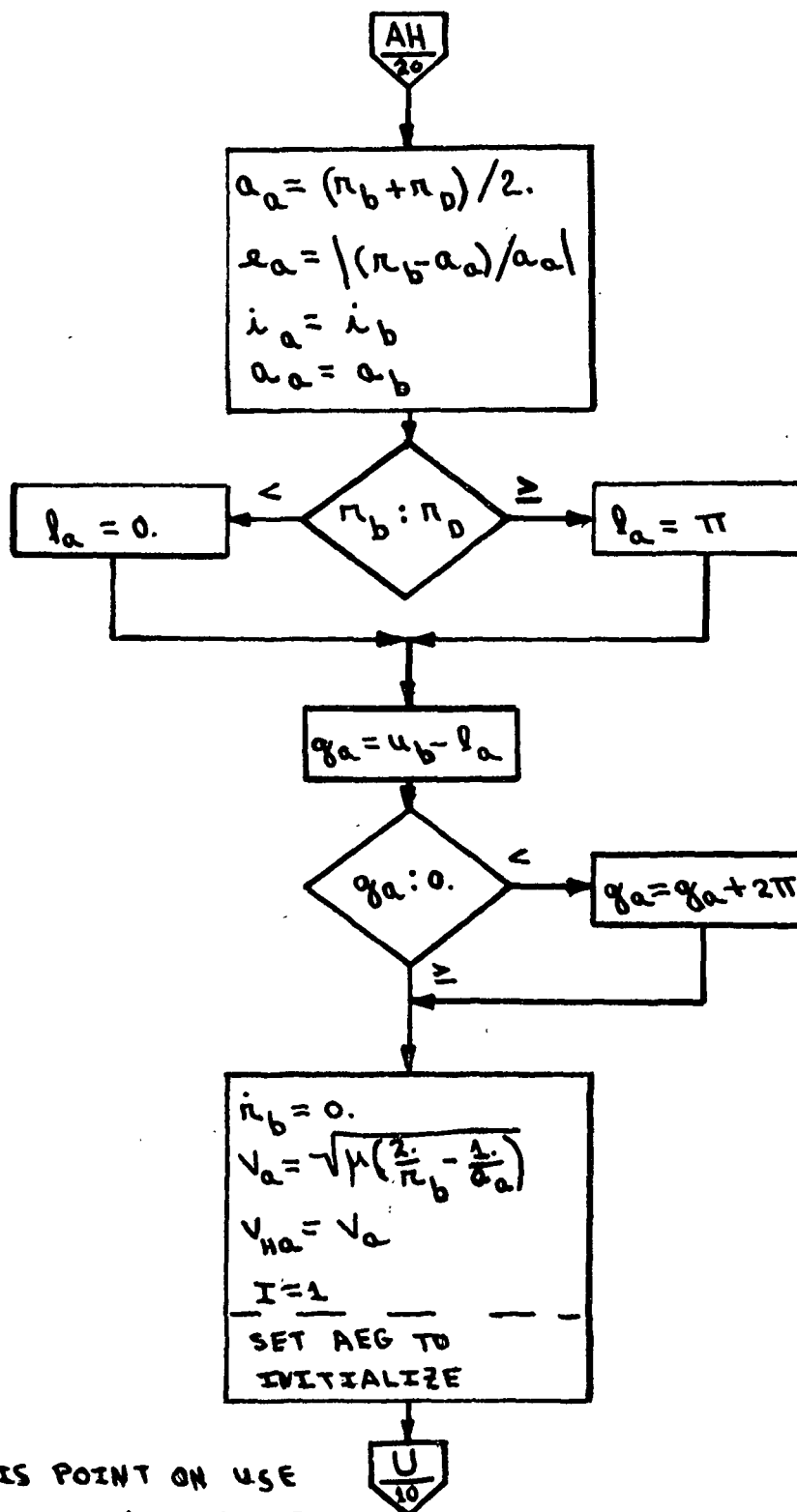
$$g_a = u_a - f_a$$



Stack the
elements at
the maneuver
at this point.

Figure PMMGPM-2. Mathematical Flowchart

20

 AHL
 AHT
 AHH


FROM THIS POINT ON USE
 THE FORMULATION FOR THE
 DESIRED HEIGHT CHANGE
 180° AROUND

Figure PMMGPM-2. Mathematical Flowchart

21

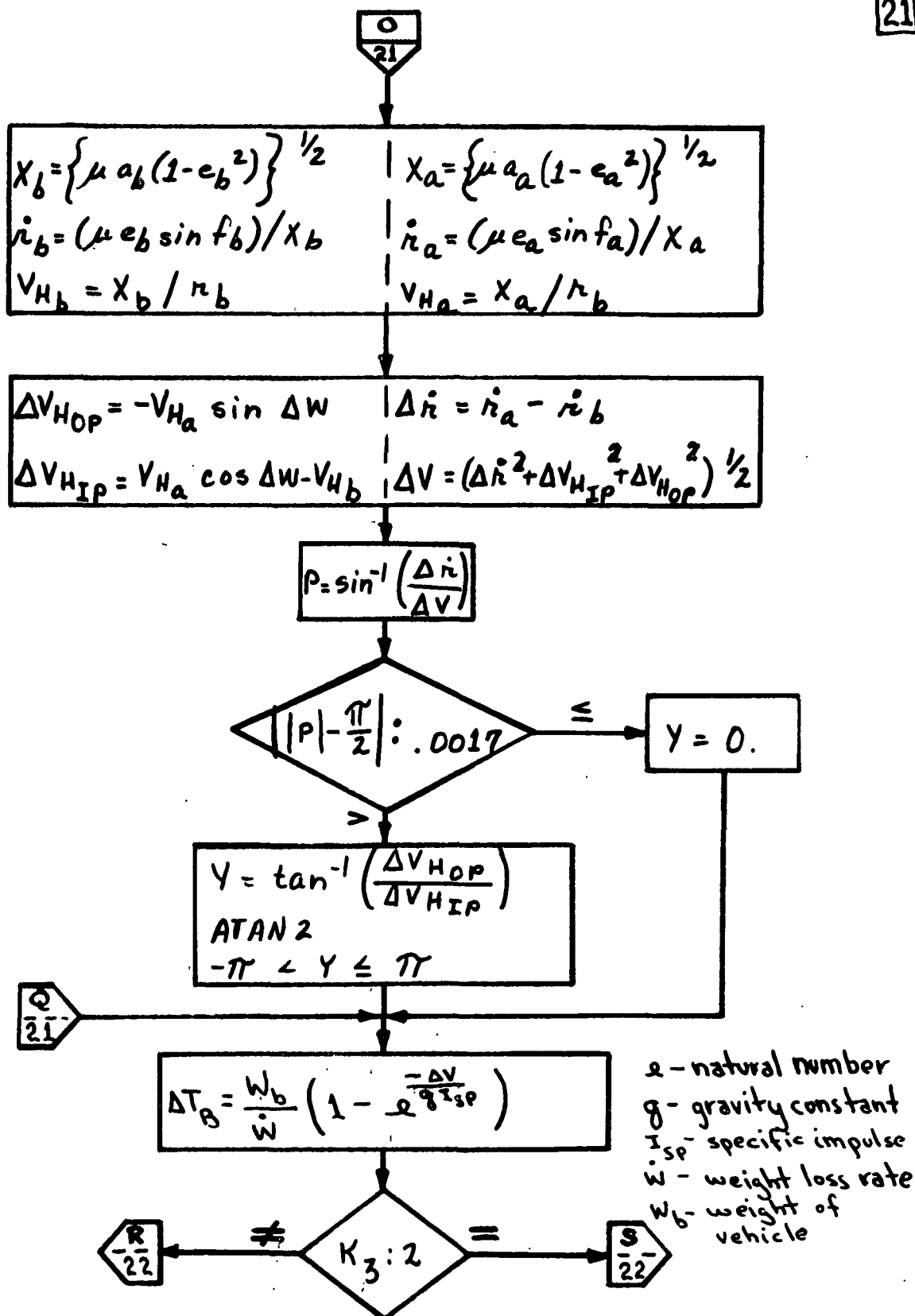


Figure PMMGPM-2. Mathematical Flowchart

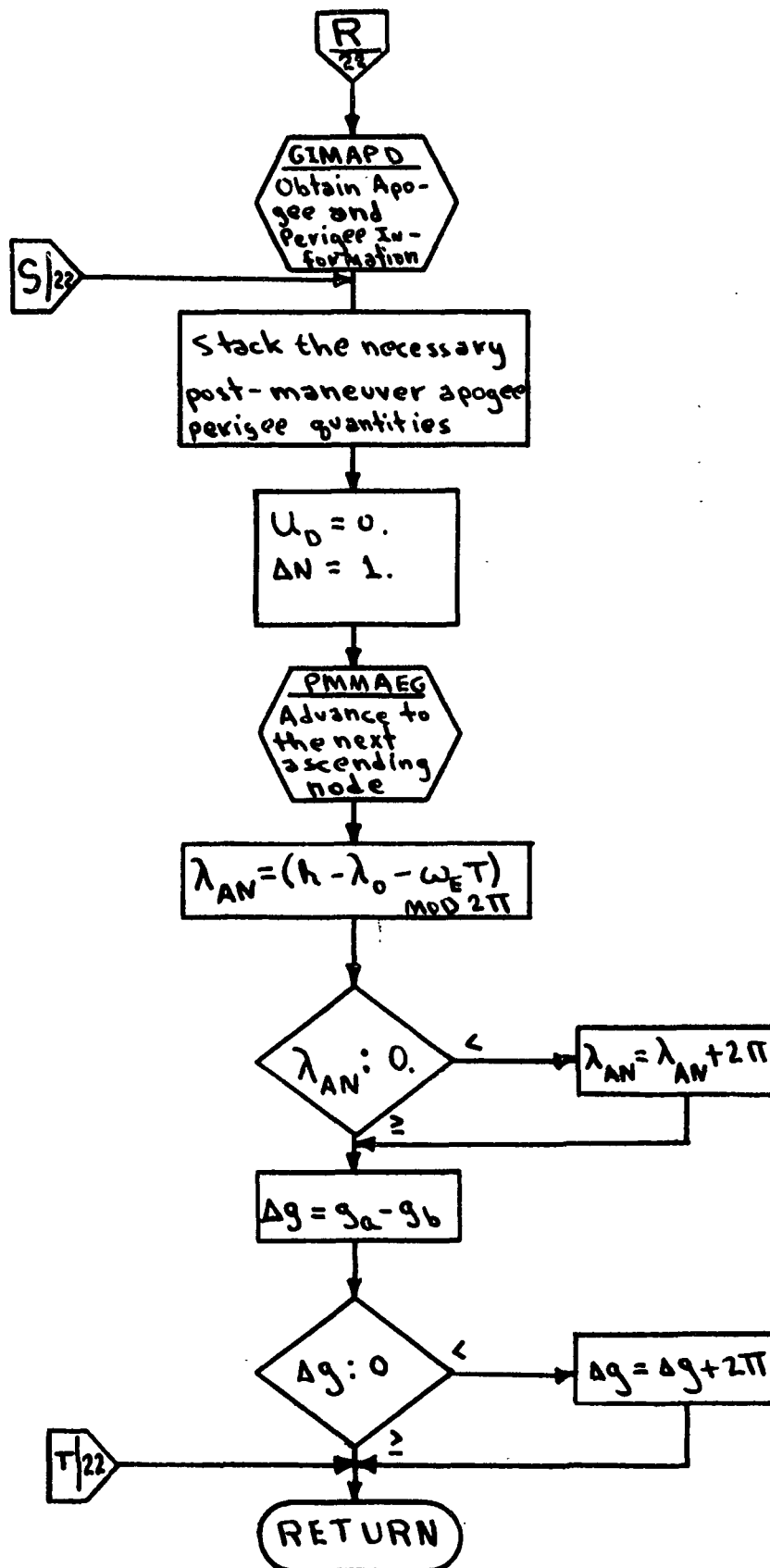


Figure PMMGPM-2. Mathematical Flowchart

- a. Symbol — PMMHMN
- b. Program Name — History Maneuver Load Module
- c. Programming Group — Maneuver Control
- d. Programmer(s) — J. Myers
- e. Language — Assembler, FORTRAN IV
- f. Data Tables Used — PZMPTLEM, GZLTRA
- g. MED's Used — M60
- h. Writeup Completion Date — 3/22/67
- i. Parameter List — Table PMMHMN-1 and 2
- j. Task Subpool List — Table PMMHMN-3
- k. System Parameters — Table PMMHMN-4
- l. Flowchart — Figure PMMHMN-1

FUNCTION

The purpose of PMMHMN is to provide the capability for confirming all types of LM maneuvers that have been previously executed. It provides the capability to change certain maneuver quantities in the past so that current mission data will reflect maneuvers as they actually were performed. This load module updates specific words in block one and in the input history maneuver's block in the LM Mission Plan Table (PZMPTLEM), but it does not completely update the maneuver's block.

USAGE

- a. Calling Sequence — LINK EP=PMMHMN, PARAM=(MED, VECT)
- b. Inputs — See Tables PMMHMN-1 and 3
- c. Outputs and Exit —

For the input History Maneuver the following quantities are updated in the LM Mission Plan Table (PZMPTLEM):

1. In all cases, the following are updated:
 - (a) Trajectory update number (block 1)
 - (b) Time to begin maneuver (block 1 and word 3 of maneuver's block)
 - (c) Attitude and Thruster Code (bytes 1-4 of word 1 of maneuver's block)
 - (d) Termination code (bytes 1-2 of word 2 of maneuver's block)
 - (e) Δt of ullage (word 4 of maneuver's block)
 - (f) Δt of maneuver and Δv of maneuver (words 5 and 6 of maneuver's block)

Approval	Approval
<i>J. A. Myers</i>	<i>J. E. James</i>



- (g) $\hat{X}_p, \hat{Y}_p, \hat{Z}_p$ (words 10-18 of maneuver's block)
 - (h) GMTI (word 75 of maneuver's block)
 - (i) Vehicle code (bytes 5, 6 of word 115 of maneuver's block)
 - (j) Dock/Undock Indicator (bytes 7, 8 of word 115 maneuver's block)
 - (k) Δt of tail off (word 116 of maneuver's block)
 - (l) Thruster and attitude part of EBCDIC maneuver purpose code (bytes 3, 4 of word 117 of maneuver's block)
2. For a LM Reaction Control System (RCS) maneuver, the following are updated in input maneuver's block:
 - (a) Pitch, Yaw, Roll (words 7-9)
 - (b) ΔV_{man} (word 65), ΔV_{To} (word 68); and if ΔV was input, ΔV_C (word 66) and ΔV_{XBT} (word 67).
 - (c) V_{gx}, V_{gy}, V_{gz} (words 76-78)
 - (d) P_g, Y_g (words 85 and 86)
 3. For an LM descent orbit injection guidance maneuver, the following are updated in input maneuver's block:
 - (a) Yaw (word 8)
 - (b) Ullage thruster option (bytes 5, 6 of word 2)
 - (c) r_D (word 19)
 4. For a LM powered landing guidance maneuver, the following are updated in input maneuver's block:
 - (a) Yaw (word 8)
 - (b) Ullage thruster option (bytes 5, 6 of word 2)
 - (c) CPT6 in RTCC inertial coordinate system (words 19-21)
 - (d) $\overline{a_T}$ (words 22-24)
 - (f) $\overline{t_F}$ (word 25)
 - (g) CPT6 in LM IMU coordinate system (words 31-33)
 5. For an LM ascent guidance maneuver, the following are updated in input maneuver's block:
 - (a) Yaw (word 8)
 - (b) Ullage thruster option (bytes 5, 6 of word 2)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMHMN
 DATE 8/12/67 (206)
 REV
 PAGE 3 of 22

- (c) t_A (word 19)
- (d) $\overline{r}_T$ in RTCC inertial coordinate system (words 20-22)
- (e) R_B (word 23)
- (f) $\overline{r}_T$ in LM IMU coordinate system (words 31-33)

- 6. For a separation maneuver, the time to end venting (word 11 of block 1) is updated.

PMMHMN exits by returning to GMSCTRL

- d. Other Programs Used — GIMKIC, GIMIKC, PMMAEG, PMXSPT, EMMRLM, GLQATN
- e. Data Restrictions and Limits — None
- f. Error — There are two error conditions: (1) if there is an error return of -1 or -2 from PMMAEG; (2) if the time to begin maneuver is within the time span of another maneuver, prior to a previous maneuver, or after the next maneuver. In both cases an on-line message is printed, processing is terminated, and GPR15 is set to nonzero. If GPR15 equals zero, then no error was discovered.

METHOD

As stated previously, the function of the History Maneuver Load Module is to update specific maneuver quantities in the LM Mission Plan Table. Most of these specified quantities are updated directly based upon the input MED data. The following paragraphs discuss the method used in computing certain specified quantities from the input MED data.

The following is the method for converting Local Vertical, Local Horizontal (LVLH) Pitch, Yaw, and Roll to Inertial Measuring Unit (IMU) Pitch, Yaw, and Roll:

Given,

- 1. $\overline{F}$, $\overline{v}$ at GMTI
- 2. P_H , Y_H , R_H (pitch, yaw, roll in LVLH coordinate system)
- 3. $\hat{X}_P$, $\hat{Y}_P$, $\hat{Z}_P$ (unit vectors in the direction of the platform X, Y, and Z axes)

The inertial coordinates of the unit vectors representing the LVLH coordinate system are computed from the given $\bar{r}$ and $\bar{v}$.

$$\hat{Z}_{PHV} = -\frac{\bar{r}}{r}, \quad \hat{Y}_{PHV} = -\frac{\bar{r} \times \bar{v}}{|\bar{r} \times \bar{v}|}$$

$$\hat{X}_{PHV} = \hat{Y}_{PHV} \times \hat{Z}_{PHV}$$

The inertial coordinates of the unit vectors representing the body axes when pitch, yaw, roll are to be applied in the sequence yaw, pitch, roll are obtained as follows:

$$\begin{bmatrix} \hat{X}_B \\ \hat{Y}_B \\ \hat{Z}_B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos R_H & \sin R_H \\ 0 & -\sin R_H & \cos R_H \end{bmatrix} \begin{bmatrix} \cos P_H & 0 & -\sin P_H \\ 0 & 1 & 0 \\ \sin P_H & 0 & \cos P_H \end{bmatrix} \begin{bmatrix} \cos Y_H & \sin Y_H & 0 \\ -\sin Y_H & \cos Y_H & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{X}_{PHV} \\ \hat{Y}_{PHV} \\ \hat{Z}_{PHV} \end{bmatrix}$$

$$\text{Let } a = \sin P_H \sin R_H, \quad b = \sin P_H \cos R_H$$

Then,

$$\hat{X}_B = \begin{bmatrix} \cos Y_H \cos P_H \end{bmatrix} \hat{X}_{PHV} + \begin{bmatrix} \sin Y_H \cos P_H \end{bmatrix} \hat{Y}_{PHV} + \begin{bmatrix} -\sin P_H \end{bmatrix} \hat{Z}_{PHV}$$

$$\hat{Y}_B = \begin{bmatrix} a \cos Y_H - \sin Y_H \cos R_H \end{bmatrix} \hat{X}_{PHV} + \begin{bmatrix} a \sin Y_H + \cos Y_H \cos R_H \end{bmatrix} \hat{Y}_{PHV} + \begin{bmatrix} \cos P_H \sin R_H \end{bmatrix} \hat{Z}_{PHV}$$

$$\hat{Z}_B = \begin{bmatrix} b \cos Y_H + \sin Y_H \sin R_H \end{bmatrix} \hat{X}_{PHV} + \begin{bmatrix} b \sin Y_H - \cos Y_H \sin R_H \end{bmatrix} \hat{Y}_{PHV} + \begin{bmatrix} \cos P_H \cos R_H \end{bmatrix} \hat{Z}_{PHV}$$

The IMU pitch, yaw, and roll are then computed by the following equations, which are derived as follows:

The IMU attitudes are applied in the sequence pitch, roll, yaw about the Y_B , Z_B , X_B axes, respectively.

Therefore,

$$\begin{bmatrix} \hat{X}_B \\ \hat{Y}_B \\ \hat{Z}_B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos Y & \sin Y \\ 0 & -\sin Y & \cos Y \end{bmatrix} \begin{bmatrix} \cos R & \sin R & 0 \\ -\sin R & \cos R & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos P & 0 & -\sin P \\ 0 & 1 & 0 \\ \sin P & 0 & \cos P \end{bmatrix} \begin{bmatrix} \hat{X}_P \\ \hat{Y}_P \\ \hat{Z}_P \end{bmatrix}$$

$$\hat{X}_B = \begin{bmatrix} \cos P \cos R \end{bmatrix} \hat{X}_P + \begin{bmatrix} \sin R \end{bmatrix} \hat{Y}_P + \begin{bmatrix} -\sin P \cos R \end{bmatrix} \hat{Z}_P$$

$$\begin{aligned} \hat{Y}_B &= \begin{bmatrix} \sin P \sin Y - \cos P \sin R \cos Y \end{bmatrix} \hat{X}_P + \begin{bmatrix} \cos R \cos Y \end{bmatrix} \hat{Y}_P \\ &\quad + \begin{bmatrix} \cos P \sin Y + \sin P \sin R \cos Y \end{bmatrix} \hat{Z}_P \end{aligned}$$

$$\begin{aligned} \hat{Z}_B &= \begin{bmatrix} \sin P \cos Y + \cos P \sin R \sin Y \end{bmatrix} \hat{X}_P + \begin{bmatrix} -\cos R \sin Y \end{bmatrix} \hat{Y}_P \\ &\quad + \begin{bmatrix} \cos P \cos Y - \sin P \sin R \sin Y \end{bmatrix} \hat{Z}_P \end{aligned}$$

By taking the appropriate dot products and solving the resulting expressions, the equations for pitch, yaw, and roll are obtained:

$$\text{Pitch} = \tan^{-1} \left[\frac{-\hat{X}_B \cdot \hat{Z}_P}{\hat{X}_B \cdot \hat{X}_P} \right]$$

$$\text{Roll} = \sin^{-1} \left[\hat{X}_B \cdot \hat{Y}_P \right]$$

$$\text{Yaw} = \tan^{-1} \left[\frac{-\hat{Z}_B \cdot \hat{Y}_P}{\hat{Y}_B \cdot \hat{Y}_P} \right]$$

These equations are valid except where roll = 90 degrees or 270 degrees. In these cases, the following equations should be used for pitch and yaw:

$$\begin{aligned} \text{Yaw} &= 0 \\ \text{Pitch} &= \tan^{-1} \left[\frac{\hat{Z}_B \cdot \hat{X}_P}{\hat{Z}_B \cdot \hat{Z}_P} \right] \end{aligned}$$

The following is the method used to compute V_{gx} , V_{gy} , V_{gz} (coordinates of ΔV vector in the platform coordinate system):

Given,

1. V_T (magnitude of thrust ΔV)
2. $\hat{X}_B$ (unit vector representing the X-body axis)
3. $\hat{X}_P$, $\hat{Y}_P$, $\hat{Z}_P$ (unit vectors in the direction of the platform X, Y, and Z axes)
4. Thruster designation

The coordinates of the ΔV vector in the body coordinate system are obtained as follows:

$$\text{for RCS + X: } \Delta \bar{V} = V_T * \hat{X}_B$$

$$\text{for RCS - X: } \Delta \bar{V} = -V_T * \hat{X}_B$$

then,

$$V_{gx} = \Delta \bar{V} \cdot \hat{X}_P$$

$$V_{gy} = \Delta \bar{V} \cdot \hat{Y}_P$$

$$V_{gz} = \Delta \bar{V} \cdot \hat{Z}_P$$

The following is the method used to convert a vector from the LM IMU coordinate system to the RTCC inertial coordinate system:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} X_{P_X} & Y_{P_X} & Z_{P_X} \\ X_{P_Y} & Y_{P_Y} & Z_{P_Y} \\ X_{P_Z} & Y_{P_Z} & Z_{P_Z} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

where: (X, Y, Z) are the coordinates of the vector in LM IMU coordinate system,

(X', Y', Z') are the coordinates of the vector in RTCC inertial coordinate system, and $(X_{P_X}, X_{P_Y}, X_{P_Z})$, $(Y_{P_X}, Y_{P_Y}, Y_{P_Z})$, $(Z_{P_X}, Z_{P_Y}, Z_{P_Z})$ are the

inertial coordinates of the unit vectors in the direction of the platform X, Y, and Z axes.

The following is the method for computing GMTI (ignition time) from the input t_{DI} (burnout time) for a Descent Orbit Injection Guidance maneuver:

$$\hat{i} = \frac{(\bar{r} \times \bar{v}) \times \bar{r}}{|\bar{r} \times \bar{v}| |\bar{r}|} = \frac{r^2 \bar{v} - (\bar{r} \cdot \bar{v}) \bar{r}}{|r^2 \bar{v} - (\bar{r} \cdot \bar{v}) \bar{r}|}$$

where $\bar{r}, \bar{v}$ are the position and velocity vectors in the original orbit at t_{DI}

$$V_D = \left[\frac{2\mu r_D}{r^2 + r r_D} \right]^{1/2}$$

where r_D is the input desired radius magnitude 180 degrees from t_{DI} and r is the magnitude of position vector in the original orbit at t_{DI}

$$\bar{V}_D = V_D \hat{i}$$

$$\bar{V}_G = \bar{V}_D - \bar{v}$$

$$V_G = \sqrt{\bar{V}_G \cdot \bar{V}_G}$$

$$\text{GMTI} = t_{\text{DI}} - \left[C_1 + \frac{V_G - C_2}{C_3} \right]$$

where,

C_1 = Δt of 10 percent phase for DPS thruster.

C_2 = Approximate ΔV gained through ullage and the 10 percent thrust phase for DPS thruster.

C_3 = Approximate acceleration from DPS maximum thrust.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMHMN
 DATE 8/12/67 (206)
 REV
 PAGE 9 of 22
Table PMMHMN-1. Parameter List

QUEID —

QNAME —

From — GMSCTRL

To-Task —

Purpose — Update Mission Plan Table

EP — PMMHMN

LINK

Byte No.	Type*	Units	Value / Description / Comments
1-4	F		Pointer to data within MED subpool
5-8	F		Pointer to anchor vector (standard format: T, $\bar{R}$, $\bar{V}$, source ID)

Table PMMHMN-2. Parameter List

QUEID —

QNAME —

From — PMMHMN

To-Task —

 Purpose — Mission Plan Table
 Update Complete

EP — GMSCTRL

LINK RETURN

Byte No.	Type*	Units	Value / Description / Comments
			If Register 15 $\neq$ 0, an error has occurred and the Mission Plan Table was not updated.

 *Type
 H — Fixed-point half word
 F — Fixed-point full word

 E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMHMN
 DATE 8/12/67 (206)
 REV
 PAGE 10 of 22

Table PMMHMN-3. Task Subpool List

Subpool Number 1

 Given to GTMSCTRL; used
 and released by PMMHMN

Symb. Ref.	Byte No.	Type*	Units	Value / Description / Comments
	1-4	F		Pointer to first MED data item (HVEH), byte N, aligned on a doubleword boundary.
HVEH	N	F		Vehicle code (=3)
HMCODE	N+4	F		Maneuver sequence number
HMSTART	N+8	D	hrs.	GMTI
	N+16	F		Thruster Code: DP = 10 ; R2 = 13 AP = 11 ; R3 = 14 R1 = 12 ; R4 = 15
	N+20	F		Attitude mode: I = 1 ; PG = 9 DG = 8 ; AG = 10
	N+24	D	Er/hr.	ΔV
	N+32	D	hrs.	Δt
	N+40	D	hrs.	Δt of ullage
	N+48	F		Coordinate Indicator LVLH = 0 ; IMU = 1
	N+52	F		Not used
	N+56	D	radians	Pitch LVLH: $-\pi/2 \leq \text{pitch} \leq \pi/2$ IMU : $0 \leq \text{pitch} < 2\pi$
	N+64	D	radians	Yaw LVLH: $-\pi \leq \text{yaw} \leq \pi$ IMU : $0 \leq \text{yaw} < 2\pi$
	N+72	D	radians	Roll LVLH: $-\pi \leq \text{roll} \leq \pi$ IMU : $0 \leq \text{roll} \leq \pi/2, 3\pi/2 \leq \text{roll} < 2\pi$
	N+80	F		Separation Maneuver Indicator 0 = Separation maneuver +1 = Separation maneuver being confirmed to zero -1 = Not separation maneuver
	N+84	F		Ullage thruster, option 0 = 2 RCS thrusters 1 = 4 RCS thrusters

 *Type
 H - Fixed-point half word
 F - Fixed-point full word

 E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMHMN
 DATE 8/12/67 (206)
 REV
 PAGE 11 of 22

Table PMMHMN-3. Task Subpool List (Cont'd)

Subpool Number 1

Symb. Ref.	Byte No.	Type*	Units	Value / Description / Comments
	N+88	D	Er	P1: r_D
			—	P2: $CPT6_X$
			Er	P3: R_B
	N+96	D	—	P2: $CPT6_Y$
			Er	P3: r_{TX}
	N+104	D	—	P2: $CPT6_Z$
			Er	P3: r_{TY}
	N+112	D	Er	P3: r_{TZ}
	N+120	D	hrs.	P3: t_A
	M	D	hrs.	GMT of anchor vector
	M+8	D	Er	X
	M+16	D	Er	Y
	M+24	D	Er	Z
	M+32	D	Er/hr.	$\dot{X}$
	M+40	D	Er/hr.	$\dot{Y}$
	M+48	D	Er/hr.	$\dot{Z}$
				Position Vector
				Velocity Vector

*Type

H — Fixed-point half word
F — Fixed-point full wordE — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMHMN
 DATE 8/12/67 (206)
 REV
 PAGE 12 of 22

Table PMMHMN-4. System Parameters

Symbol in Program	Symbol In Master Tape	Type *	Units	Comments
TWOPI	MCMTP1	D	radians	360 degrees in radians
PIO2	MCMHPI	D	radians	90 degrees in radians
SRMU	MCERMU	D	$(E_r)^3 / hr^2$	Square root of μ
DTULAP(1)	MCTDUP	D	hrs.	Delta T of ullage overlap for DPS thruster
DTULAP(2)	MCTAUP	D	hrs.	Delta T of ullage overlap for APS thruster
DTTO(1)	MCTDTF	D	hrs.	Delta T of Tail Off for DPS thruster
DTTO(2)	MCTATF	D	hrs.	Delta T of Tail Off for APS thruster
CONS(1)	MCTTFT	D	hrs.	Delta T of 10% phase for DPS thruster
CONS(2)	MCTDTV	D	E_r / hr	Approximate ΔV gained through ullage and the 10% thrust phase for DPS thruster
CONS(3)	MCTDFA	D	E_r / hr^2	Approximate acceleration from DPS maximum thrust

*Type
 H — Fixed-point half word
 F — Fixed-point full word
 E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

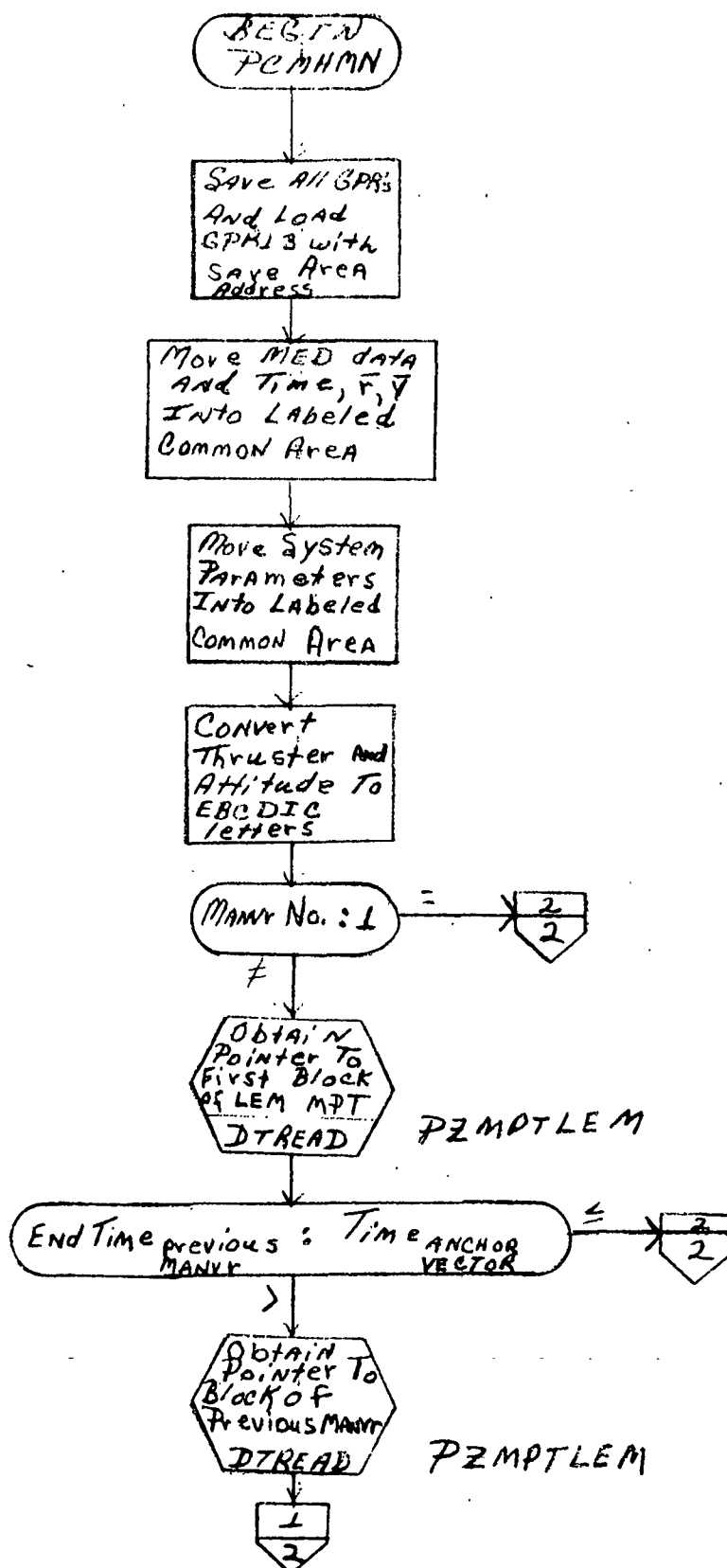


Figure PMMHMN-1. Flowchart
(Sheet 1 of 10)

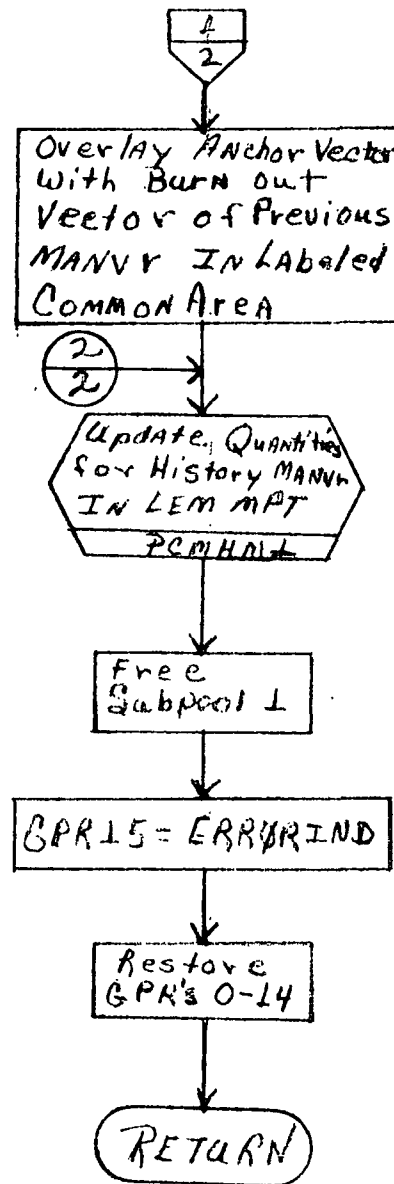


Figure PMMHMN-1. Flowchart
(Sheet 2 of 10)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMHMN

DATE 8/12/67 (206)

REV

PAGE 15 of 22

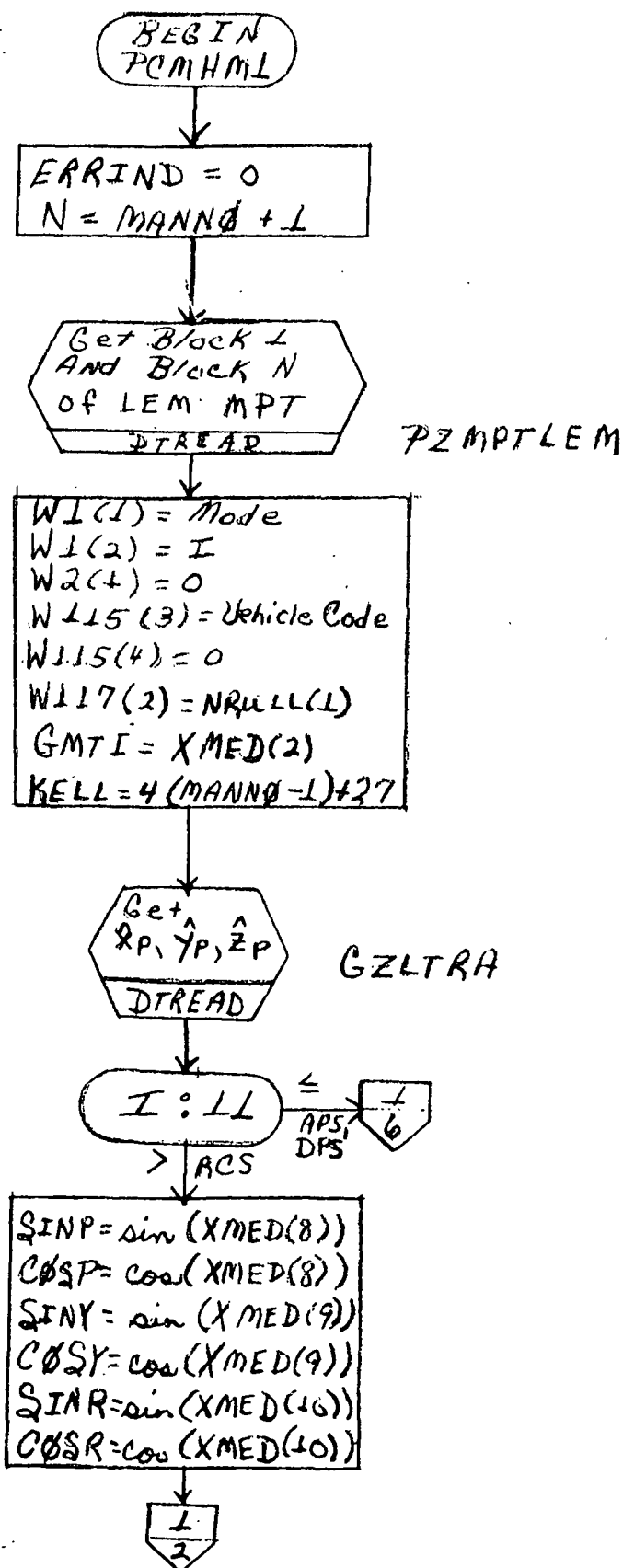


Figure PMMHMN-1. Flowchart
(Sheet 3 of 10)

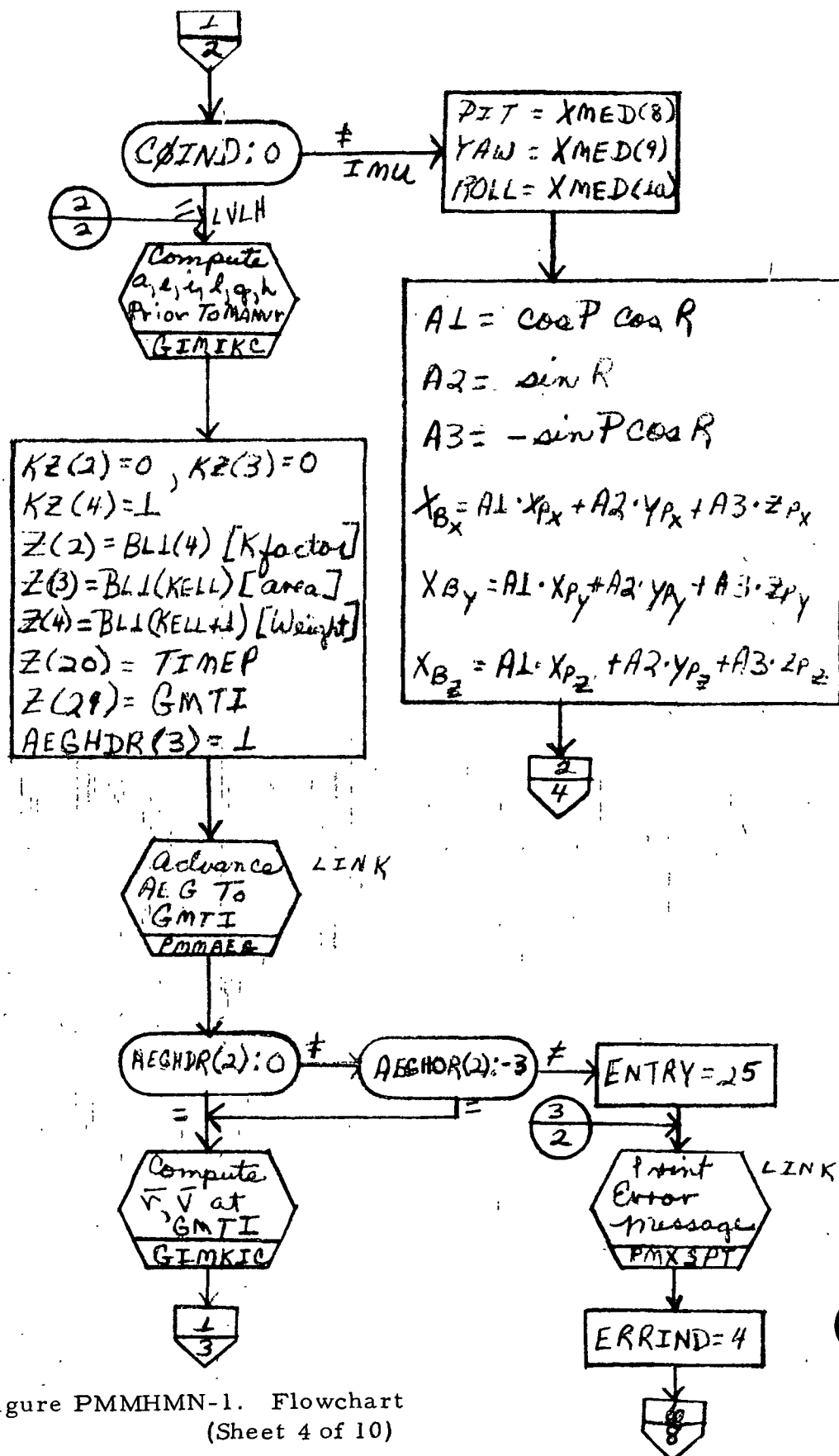


Figure PMMHMN-1. Flowchart
(Sheet 4 of 10)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMHMN

DATE 8/12/67 (206)

REV

PAGE 17 of 22

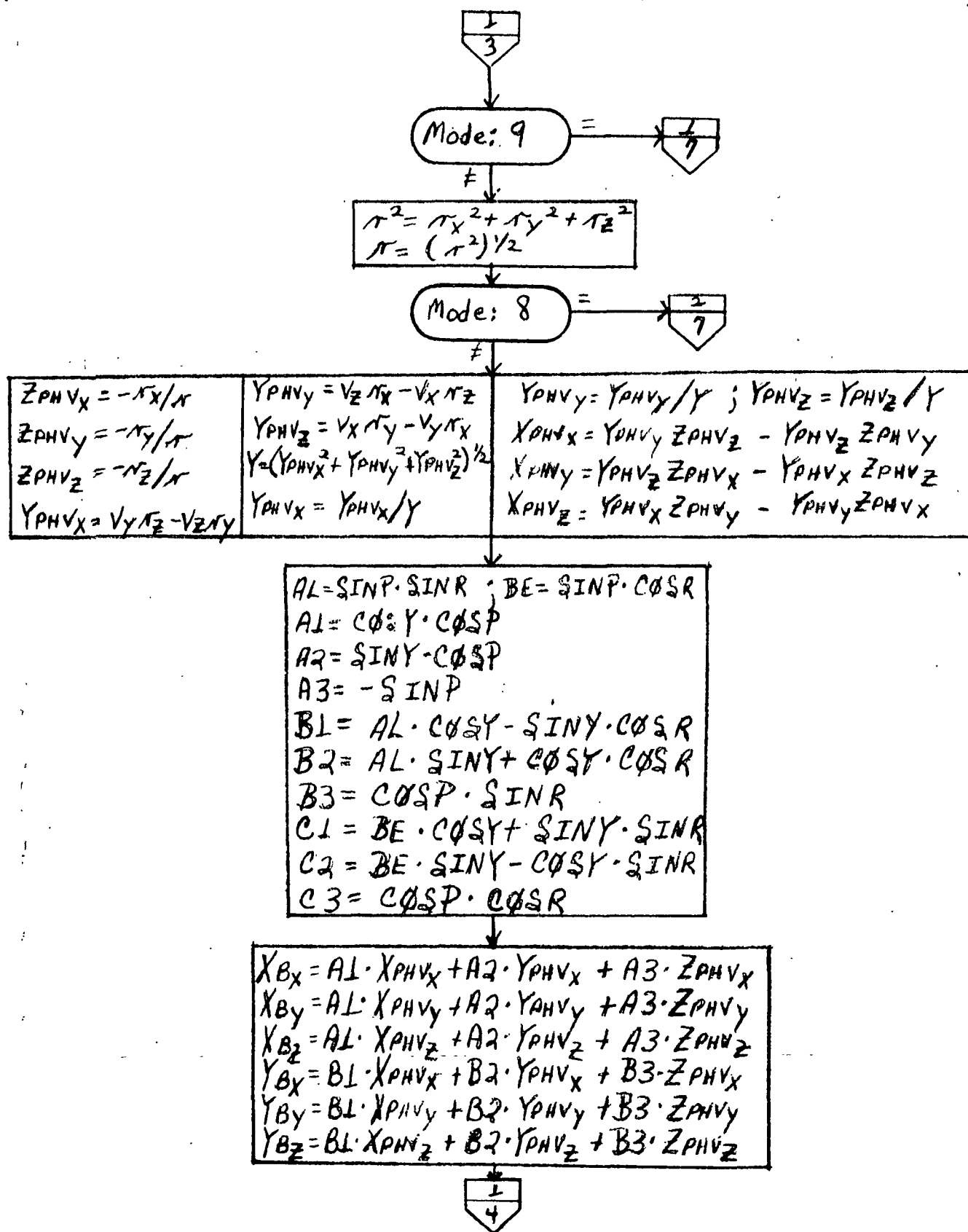
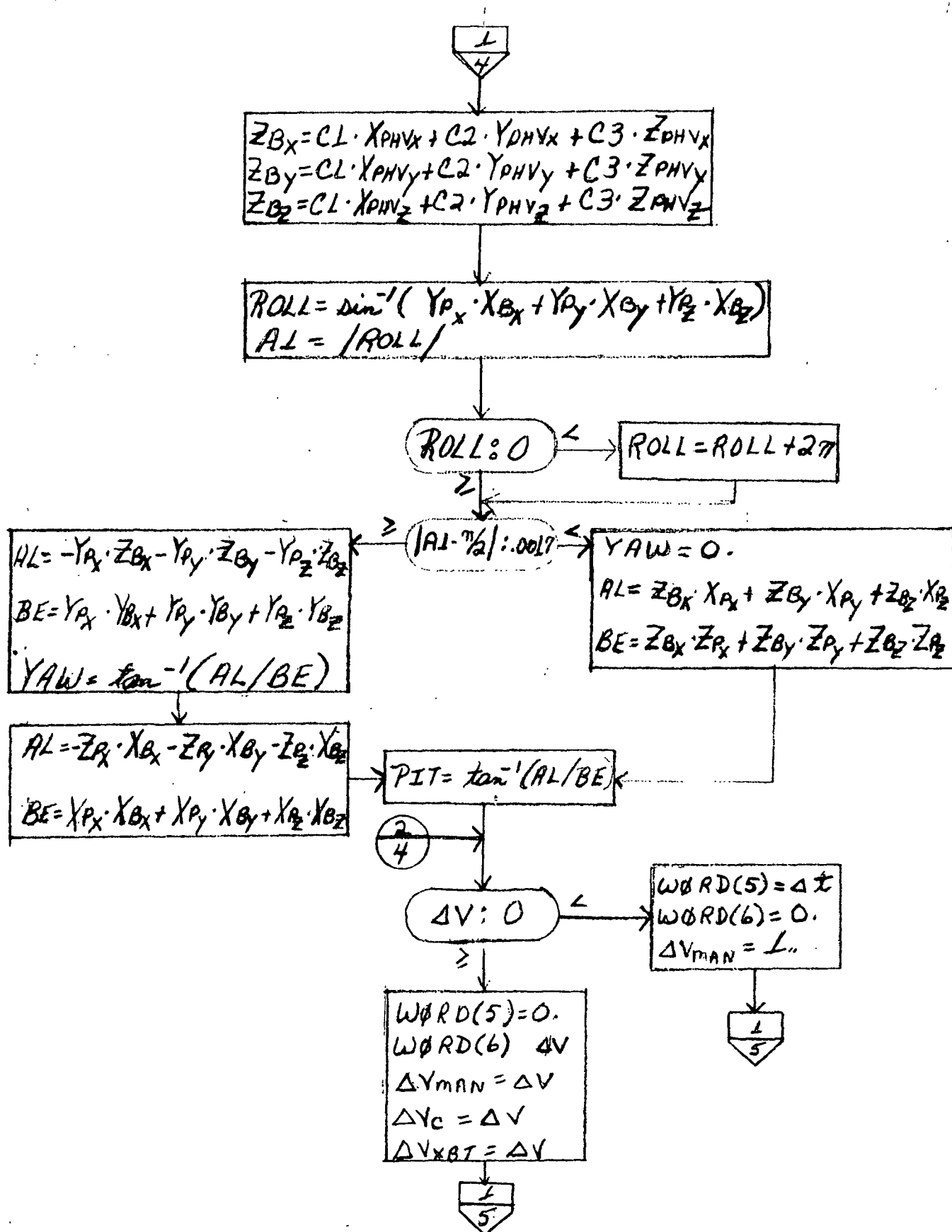


Figure PMMHMN-1. Flowchart

(Sheet 5 of 10)


 Figure PMMHMN-1. Flowchart
 (Sheet 6 of 10)

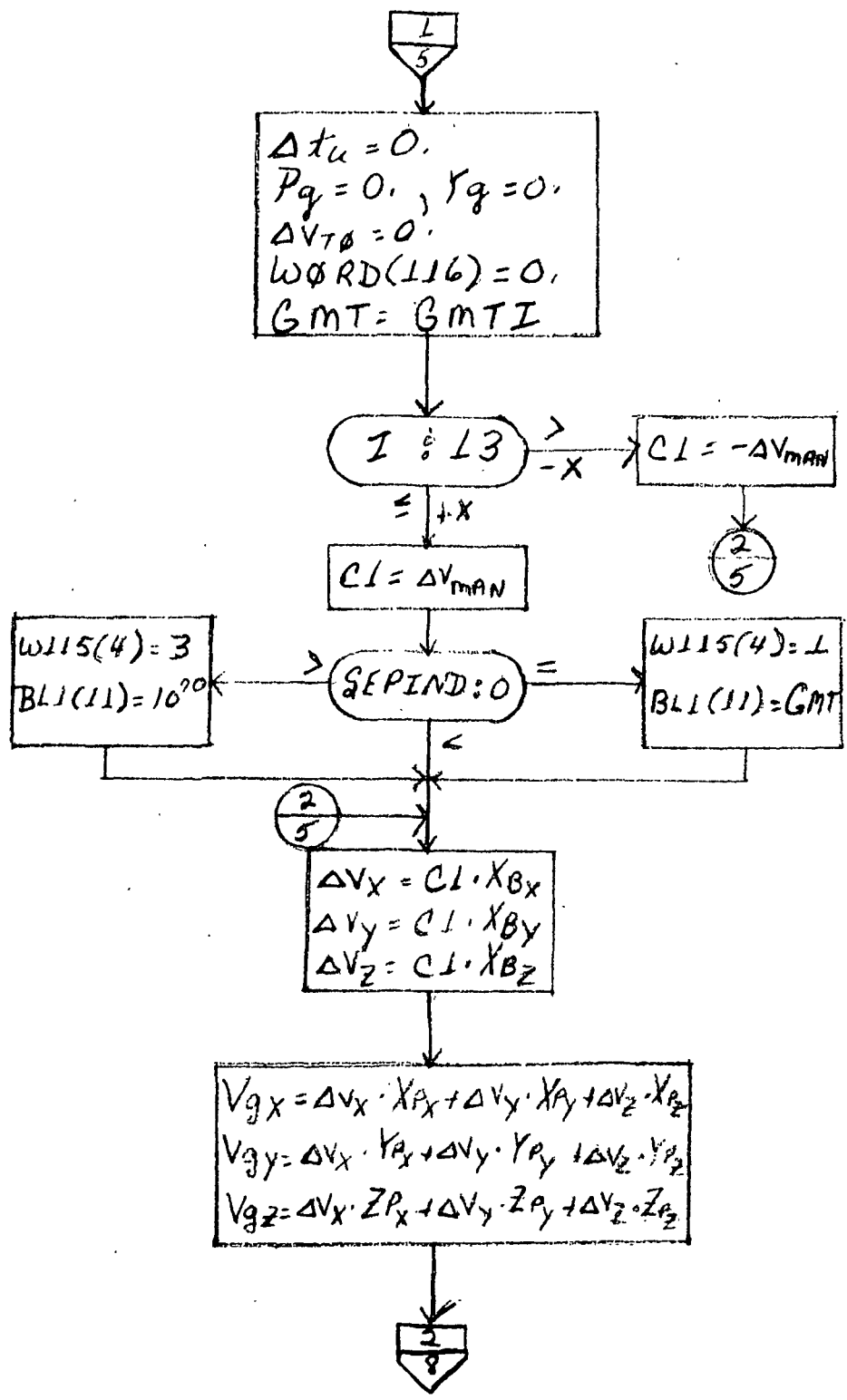


Figure PMMHMN-1. Flowchart
(Sheet 7 of 10)

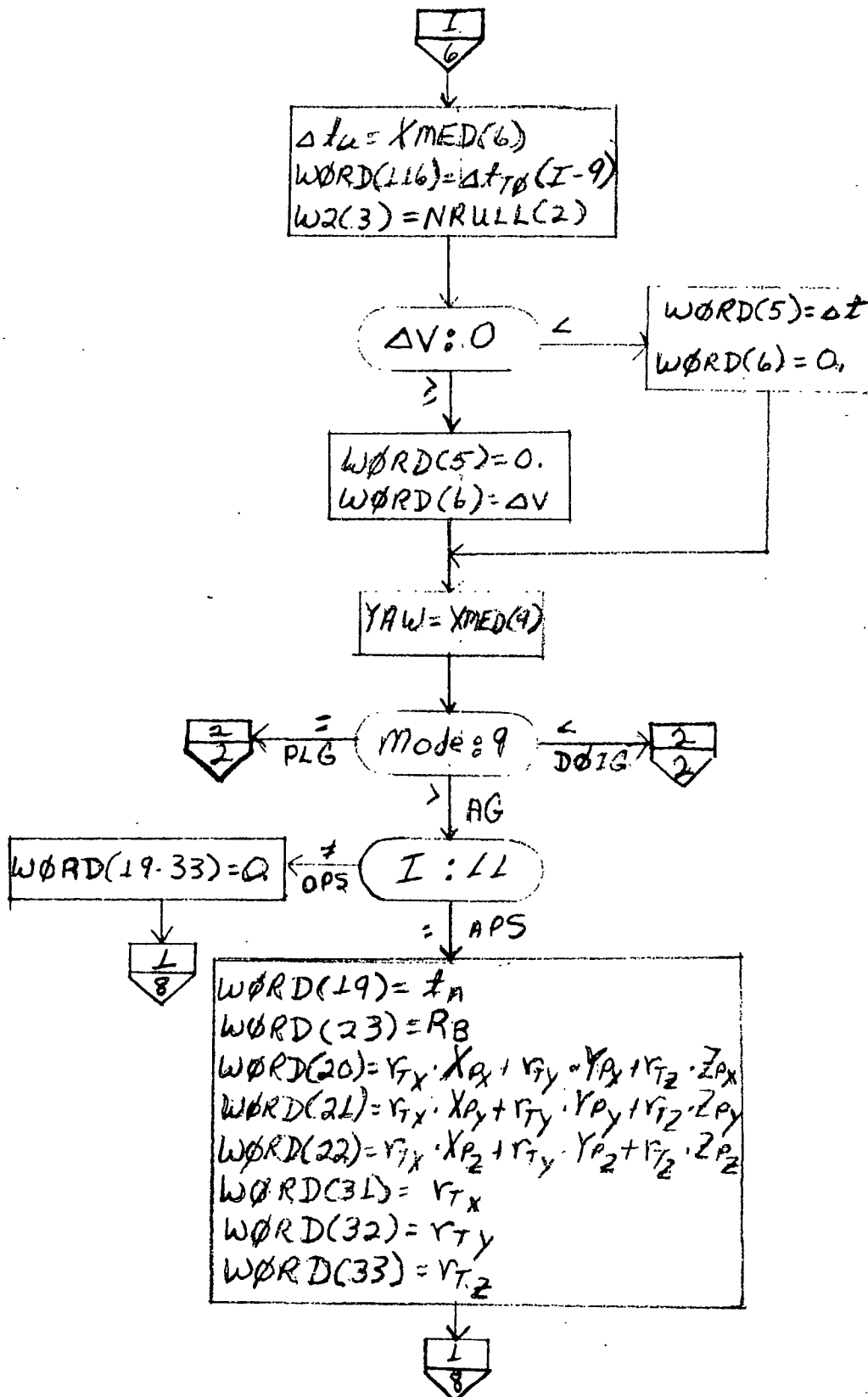


Figure PMMHMN-1. Flowchart
(Sheet 8 of 10)

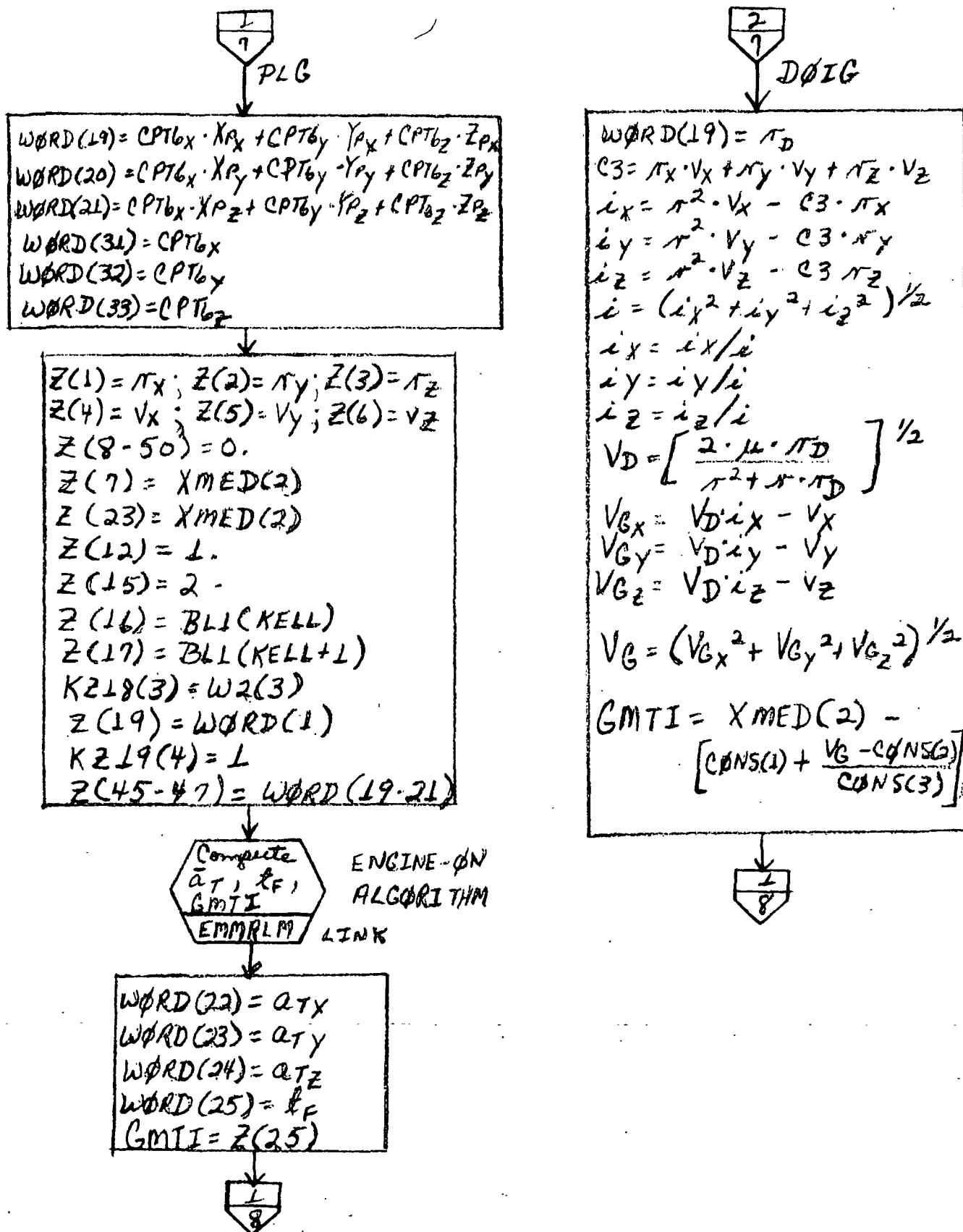


Figure PMMHMN-1. Flowchart

(Sheet 9 of 10)

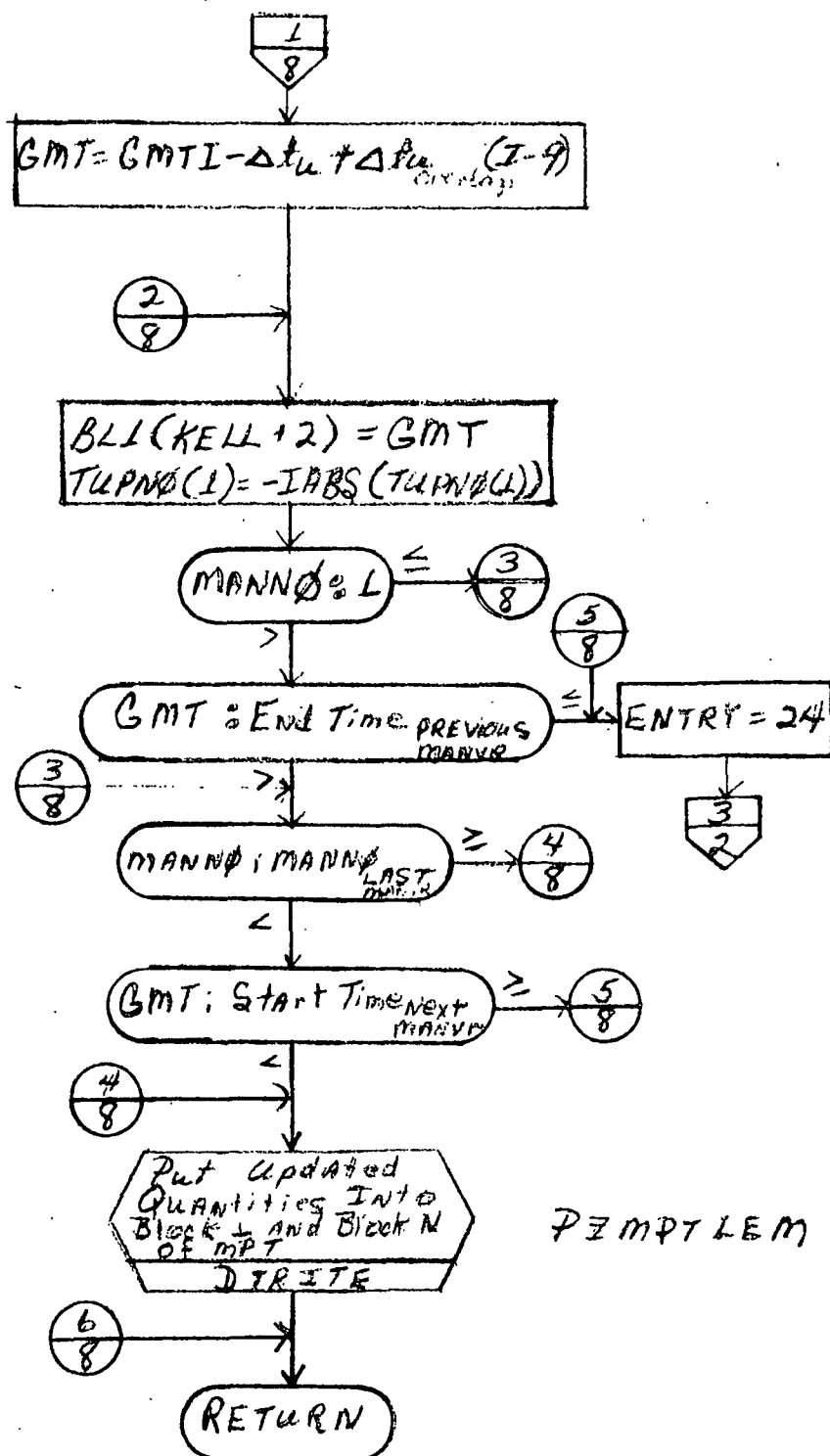


Figure PMMHMN-1. Flowchart
(Sheet 10 of 10)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMCD
DATE 10/31/67 (EORP)
REV
PAGE 1 of 29

- a. Symbol — PMMMCD
- b. Program Name — Maneuver Confirmation and Direct Input Load Module
- c. Programming Group — Maneuver Control
- d. Programmer(s) — J. Myers
- e. Language — Assembler
- f. Data Tables Used — PZMPTCSM, PZMPTLEM, GZLTRA
- g. MED's Used — M60, M66
- h. Writeup Completion Date — 9/12/67
- i. System Parameters — Table PMMMCD-1
- j. Flowchart — Figure PMMMCD-1

FUNCTION

The Maneuver Confirmation and Direct Input Load Module provides the capability for the direct input and for the confirmation of an S-IVB blowdown vent, CSM, or LM maneuver. In the case of a direct input, a new future maneuver is built based upon the input data and transferred into the appropriate Mission Plan Table. In the case of a confirmation, certain quantities for a specified history (executed) maneuver are changed based upon the input data so that this past maneuver and the resultant orbit are consistent with the actual execution of the maneuver by the spacecraft.

USAGE

- a. Calling Sequence — LINK EP=PMMMCD
- b. Inputs — Upon entry to this load module, GPR 4 contains the address of a block of input data which has the following format:

	Units	
Doublewords 1-52		Not used
Doubleword 53 - bytes 1-4		(Maneuver Number-1) *32
bytes 5-8		Pointer to previous maneuver block of MPT
Doublewords 54-77		Not used
Doubleword 78	lbs.	Vehicle weight at maneuver initiation
Doubleword 79 - bytes 1-4		Not used
bytes 5-8		Pointer to MPT
		EBCDIC name

Approval	Approval
<i>J.A. Myers</i>	<i>S.E. James</i>

	Units	
Doubleword 80 - bytes 1-2		Maneuver block number
bytes 3-4		Maneuver block number
bytes 5-8		Confirmation/Direct Input
		Indicator (Confirmation=36,
		Direct Input=33)
Doubleword 81 - bytes 1-4		Table indicator
bytes 5-8		(CSM=1, LM=3)
Doubleword 82	hrs.	Maneuver Number
		GMTI (ignition time for
		specified thruster)
Doubleword 83 - bytes 1-4		Thruster Code:
		2 CSM RCS + X = 1
		4 CSM RCS + X = 2
		2 CSM RCS - X = 3
		4 CSM RCS - X = 4
		S-IVB blowdown vent = 16
		2 LM RCS + X = 17
		4 LM RCS + X = 18
		2 LM RCS - X = 19
		4 LM RCS - X = 20
		CSM SPS = 33
		LM APS = 34
		LM DPS = 35
bytes 5-8		Attitude Mode:
		Fixed Inertial = 1
		Manual holding body
		orientation invariant = 2
		External ΔV (primary)=4
		External ΔV (LM AGS)=5
Doublewords 84-86		Burn Parameters
	Er/hr.	P1:
		ΔV of maneuver
		ΔV indicator (MAG=0, DVC=1)
	hrs.	Δt of maneuver
	Er/hr.	P2:
		V_x } Components of
		V_y } external ΔV vector
		V_z } in display and
		command system
	Er/hr.	P3:
		V_{gx} } Components of ΔV
		V_{gy} } vector in IMU
		V_{gz} } coordinate system

IBM RTCC APOLLO PROGRAMMING SYSTEMS

2. PMMMCD
 DATE 10/31/67 (EORP)
 REV
 PAGE 3 of 29

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

	Units	
	Er/hr.	V_F
	Er/hr.	V_S
	Er/hr.	V_D
		} Components of ΔV vector in LVLH coordinate system
	Er/hr.	ΔV_x
	Er/hr.	ΔV_y
	Er/hr.	ΔV_z
		} Maneuver $\overline{\Delta V}$ residuals in RCS control axes system
Doubleword 87 - byte 1		P4: Burn parameter indicator (P1=1, P2=2, P3=3, P4=4, P5=5)
bytes 2-8		Not used
Doubleword 88 - bytes 1-4		Not used
bytes 5-8		Coordinate indicator (LVLH=0, IMU=1)
Doubleword 89	radians	Pitch
Doubleword 90	radians	Yaw
Doubleword 91	radians	Roll
Doubleword 92	hrs.	Δt of ullage
Doubleword 93 - bytes 1-4		Ullage thruster option (0=2 RCS thrusters, 1=4 RCS thrusters)
bytes 5-8		Vehicle heads-up/heads-down indicator (0=down, 1=up)
Doubleword 94	hrs.	Δt of 10% thrust level for DPS thruster
Doubleword 95 - bytes 1-4		Reference matrix code
		CUR=1 DMT=6
		PCR=2 DOD=7
		TLM=3 DOK=7
		OST=4 LCV=8
		MED=5 AGS=9
bytes 5-8		LM Ascent/Descent Separation flag (=0 means set the time of LM A/D separation; $\neq 0$ means do not set the time of LM A/D separation)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMCD

DATE 10/31/67 (EORP)

REV

PAGE 4 of 29

	<u>Units</u>	
Doubleword 96 - bytes 1-4		Configuration change indicator
		0=No change from configuration at end of previous maneuver
		1=Undocking
		2=Docking
		3=Separation of LM Ascent from LM Descent
bytes 5-8		Configuration code at maneuver initiation
		0=CSM
		1=LM
		2=CSM and LM
		3=CSM and S-IVB
		4=LM and S-IVB
		5=CSM, LM, S-IVB
Doubleword 97 - bytes 1-4		Configuration code at the end of the maneuver
		0=CSM
		1=LM
		2=CSM and LM
		3=CSM and S-IVB
		4=LM and S-IVB
		5=CSM, LM, S-IVB
bytes 5-8		Maneuvering vehicle code
		1=CSM, 2=S-IVB, 3=LM
Doubleword 98	radians	Docking Angle
Doubleword 99 - bytes 1-4		Trim angle indicator (input only for confirmation)
		0=Trim angles are to be computed
		2=System parameters are used for trim angles
bytes 5-8		Not used
Doublewords 100-104		Not used
Doubleword 105	hrs.	Time of anchor vector
Doubleword 106	Er	X } Components of position vector
Doubleword 107	Er	
Doubleword 108	Er	

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2, PMMMCD
DATE 10/31/67 (EORP)
REV
PAGE 5 of 29

	Units	
Doubleword 109	Er/hr.	$\dot{X}$
Doubleword 110	Er/hr.	$\dot{Y}$
Doubleword 111	Er/hr.	$\dot{Z}$
Doubleword 112	ft. ²	} Components of velocity vector
Doubleword 113	lbs.	
Doubleword 114	hrs.	
Doubleword 115	hrs.	Area prior to maneuver
Doubleword 116		Weight prior to maneuver
		Start time limit
		End time limit
		EBCDIC maneuver purpose code
Doublewords 117-127		Not used
Doublewords 128-136		Time of vector, position vector, velocity vector, area prior to maneuver, and weight prior to maneu- ver for the CSM, input only when the attitude mode is LM AGS external ΔV
		Block 1 of the Mission Plan Table
Doublewords 137-256		

c. Outputs and Exit — The following quantities are output in Block 1 of the appropriate Mission Plan Table (MPT) (either PZMPTCSM or PZMPTLEM):

1. Trajectory update number
2. Time to begin this maneuver
3. Time of LM jettison of descent stage (if the LM Ascent/Descent separation flag is zero).

In all cases, the following quantities are output in the maneuver's descriptive block of the appropriate MPT:

1. Attitude mode (bytes 1, 2 of word 1)
2. Thruster code (bytes 3, 4 of word 1)
3. Configuration code at maneuver initiation (bytes 1, 2 of word 2)
4. Time to begin this maneuver (word 3)
5. Δt of ullage (word 4)
6. Δt of maneuver (word 5)
7. ΔV of maneuver (word 6)
8. $\hat{A}_T$ (Unit thrust vector at designated thruster ignition time in RTCC inertial coordinate system) (words 7-9)
9. $\hat{X}_B, \hat{Y}_B, \hat{Z}_B$ (RTCC inertial coordinates of the unit vectors in the direction of the vehicle body X, Y, and Z axes, respectively) (words 10-18)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMMCD
 DATE 10/31/67 (EORP)
 REV
 PAGE 6 of 29

10. Maneuvering vehicle code (bytes 1, 2 of word 20)
11. Docking angle (word 21)
12. Configuration code at the end of this maneuver (bytes 1, 2 of word 44)
13. Configuration change indicator (bytes 3, 4 of word 44)
14. GMTI (ignition time for specified thruster) (word 75)
15. Impulsive maneuver time (word 77)
16. EBCDIC maneuver code (word 117)
17. Maneuver uncertainties (words 118-120).

If the thruster is CSM or LM RCS, the mode is fixed inertial or manual holding body orientation invariant, and the ΔV or ΔV vector was input, then the following quantities are also output in the maneuver's descriptive block of the appropriate MPT:

1. ΔV_{MAN} (total ΔV) (word 65)
2. ΔV_C (ΔV along X-body axis excluding tailoff) (word 66)
3. ΔV_{xbt} (total ΔV along X-body axis) (word 67).

If the thruster is CSM SPS, LM APS, or LM DPS, then the following quantities are also output in the maneuver's descriptive block of the appropriate MPT:

1. Ullage thruster option (bytes 5, 6 of word 1)
2. Δt of 10% thrust level for DPS thruster (output only if thruster is LM DPS) (word 19)
3. Thruster trim angle indicator (output only if CSM SPS or LM DPS thruster) (bytes 3, 4 of word 20).

If the attitude mode is primary external $\overline{\Delta V}$ or LM AGS external $\overline{\Delta V}$, then the following quantities are also output in the maneuver's descriptive block of the appropriate MPT:

V_X, V_Y, V_Z (components of external ΔV vector in RTCC inertial system) (words 24-26).

In the case of a confirmation, the following quantities are also output in the maneuver's descriptive block of the appropriate MPT:

1. CSM area and weight after this maneuver (words 45, 46)
2. S-IVB area and weight after this maneuver (words 47, 48)
3. LM area and weight after this maneuver (words 49, 50).

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMMCD
 DATE 10/31/67 (EORP)
 REV
 PAGE 7 of 29

In all cases, the remaining words of the maneuver's descriptive block of the appropriate MPT are set to zero.

PMMMCD exits by returning to PMMXFR.

- d. Other Programs Used — GIMIKC, GIMKIC, PIAEDV, PIEXDV, EIGMTX, PIDVRS, GIMGBL, PMMAEG
- e. Data Restrictions and Limits — None
- f. Error — There are three error conditions. The first is an error return of -1 or -2 from PMMAEG. The second is an error return from EIGMTX. A third error condition exists if the time to begin maneuver is within the time span of another maneuver, prior to a previous maneuver, or after the next maneuver. In all cases, processing is terminated, GPR 15 is set to non-zero, and an on-line message is printed. If GPR 15 is zero, then no error was discovered.

METHOD

The function of this load module is to output specific maneuver-related quantities that are contained in the CSM or LM Mission Plan Table. Some of these specified quantities are set according to the input MED data. The following paragraphs discuss the method used in computing certain specified quantities from the input MED data.

The following is the method for computing the vehicle body defining vectors in RTCC inertial coordinate system ($\hat{X}_B$, $\hat{Y}_B$, $\hat{Z}_B$) when the LVLH (Local Vertical, Local Horizontal) Pitch (P_H), Yaw (Y_H), and Roll (R_H) are given.

The inertial coordinates of the unit vectors representing the LVLH coordinate system are computed from the $\bar{r}$ and $\bar{v}$ at GMTI.

$$\hat{Z}_{PHV} = \frac{-\bar{r}}{r}, \quad \hat{Y}_{PHV} = \frac{-\bar{r} \times \bar{v}}{|\bar{r} \times \bar{v}|}$$

$$\hat{X}_{PHV} = \hat{Y}_{PHV} \times \hat{Z}_{PHV}$$

For the CSM, LM, and S-IVB (blowdown vent thruster), the LVLH attitudes are applied in the sequence Yaw, Pitch, Roll about the Z_B , Y_B , X_B axes, respectively.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMMCD
 DATE 10/31/67 (EORI)
 REV
 PAGE 8 of 29

Therefore,

$$\begin{bmatrix} \hat{X}_B \\ \hat{Y}_B \\ \hat{Z}_B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos R_H & \sin R_H \\ 0 & -\sin R_H & \cos R_H \end{bmatrix} \begin{bmatrix} \cos P_H & 0 & -\sin P_H \\ 0 & 1 & 0 \\ \sin P_H & 0 & \cos P_H \end{bmatrix} \begin{bmatrix} \cos Y_H & \sin Y_H & 0 \\ -\sin Y_H & \cos Y_H & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{X}_{PHV} \\ \hat{Y}_{PHV} \\ \hat{Z}_{PHV} \end{bmatrix}$$

$$\text{Let } a = \sin P_H \sin R_H, \quad b = \sin P_H \cos R_H$$

Then,

$$\hat{X}_B = [\cos Y_H \cos P_H] \hat{X}_{PHV} + [\sin Y_H \cos P_H] \hat{Y}_{PHV} + [-\sin P_H] \hat{Z}_{PHV}$$

$$\begin{aligned} \hat{Y}_B &= [a \cos Y_H - \sin Y_H \cos R_H] \hat{X}_{PHV} + [a \sin Y_H + \cos Y_H \cos R_H] \hat{Y}_{PHV} \\ &\quad + [\cos P_H \sin R_H] \hat{Z}_{PHV} \end{aligned}$$

$$\begin{aligned} \hat{Z}_B &= [b \cos Y_H + \sin Y_H \sin R_H] \hat{X}_{PHV} + [b \sin Y_H - \cos Y_H \sin R_H] \hat{Y}_{PHV} \\ &\quad + [\cos P_H \cos R_H] \hat{Z}_{PHV} \end{aligned}$$

The following is the method for computing the vehicle body defining vectors in RTCC inertial coordinate system ($\hat{X}_B$, $\hat{Y}_B$, $\hat{Z}_B$) when the IMU (Inertial Measuring Unit) Pitch (P), Yaw (Y), and Roll (R) are given.

For the CSM and the S-IVB (blowdown vent thruster), the IMU attitudes are applied in the sequence Pitch, Yaw, Roll about the Y_B , Z_B , X_B axes, respectively.

Therefore,

$$\begin{bmatrix} \hat{X}_B \\ \hat{Y}_B \\ \hat{Z}_B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos R & \sin R \\ 0 & -\sin R & \cos R \end{bmatrix} \begin{bmatrix} \cos Y & \sin Y & 0 \\ -\sin Y & \cos Y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos P & 0 & -\sin P \\ 0 & 1 & 0 \\ \sin P & 0 & \cos P \end{bmatrix} \begin{bmatrix} \hat{X}_P \\ \hat{Y}_P \\ \hat{Z}_P \end{bmatrix}$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMCD

DATE 10/31/67 (EORP)

REV

PAGE 9 of 29

where $\hat{X}_P$, $\hat{Y}_P$, $\hat{Z}_P$ are the unit vectors representing the IMU coordinate system.

Let $a = \cos P \sin Y$, $b = \sin P \sin Y$.

Then,

$$\hat{X}_B = [\cos P \cos Y] \hat{X}_P + [\sin Y] \hat{Y}_P + [-\sin P \cos Y] \hat{Z}_P$$

$$\hat{Y}_B = [\sin P \sin R - a \cos R] \hat{X}_P + [\cos Y \cos R] \hat{Y}_P + [\cos P \sin R + b \cos R] \hat{Z}_P$$

$$\hat{Z}_B = [\sin P \cos R + a \sin R] \hat{X}_P + [-\cos Y \sin R] \hat{Y}_P + [\cos P \cos R - b \sin R] \hat{Z}_P$$

For the LM, the IMU attitudes are applied in the sequence Pitch, Roll, Yaw about the Y_B , Z_B , X_B axes, respectively.

Therefore,

$$\begin{bmatrix} \hat{X}_B \\ \hat{Y}_B \\ \hat{Z}_B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos Y & \sin Y \\ 0 & -\sin Y & \cos Y \end{bmatrix} \begin{bmatrix} \cos R & \sin R & 0 \\ -\sin R & \cos R & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos P & 0 & -\sin P \\ 0 & 1 & 0 \\ \sin P & 0 & \cos P \end{bmatrix} \begin{bmatrix} \hat{X}_P \\ \hat{Y}_P \\ \hat{Z}_P \end{bmatrix}$$

Let $a = \cos P \sin R$, $b = \sin P \sin R$.

Then,

$$\hat{X}_B = [\cos P \cos R] \hat{X}_P + [\sin R] \hat{Y}_P + [-\sin P \cos R] \hat{Z}_P$$

$$\hat{Y}_B = [\sin P \sin Y - a \cos Y] \hat{X}_P + [\cos R \cos Y] \hat{Y}_P + [\cos P \sin Y + b \cos Y] \hat{Z}_P$$

$$\hat{Z}_B = [\sin P \cos Y + a \sin Y] \hat{X}_P + [-\cos R \sin Y] \hat{Y}_P + [\cos P \cos Y - b \sin Y] \hat{Z}_P$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMCD

DATE 10/31/67 (EORF)

REV

PAGE 10 of 29

The following is the method for computing the vehicle body defining vectors in RTCC inertial coordinate system ($\hat{X}_B, \hat{Y}_B, \hat{Z}_B$) when the unit thrust vector ($\hat{A}_T$) is given and the vehicle attitude angles are not given.

$$\hat{X}_T = \hat{A}_T$$

$$\bar{h}' = \bar{r} \times \hat{X}_T$$

$\bar{r}$ is position vector at GMTI

$$h' = \sqrt{\bar{h}' \cdot \bar{h}'}$$

If $h' = 0$,

$$\text{then } \hat{Y}_T = \frac{\bar{r} \times \bar{v}}{|\bar{r} \times \bar{v}|}$$

$\bar{v}$ is the velocity vector at GMTI

If $h' \neq 0$,

$$\text{then for heads up } \hat{Y}_T = \frac{-\bar{h}'}{h'}$$

$$\text{then for heads down } \hat{Y}_T = \frac{\bar{h}'}{h'}$$

$$\hat{Z}_T = \hat{X}_T \times \hat{Y}_T$$

For the CSM and LM RCS + X thrusters and the S-IVB blowdown vent thruster:

$$\hat{X}_B = \hat{X}_T, \quad \hat{Y}_B = \hat{Y}_T, \quad \hat{Z}_B = \hat{Z}_T.$$

For the CSM and LM RCS - X thrusters:

$$\hat{X}_B = -\hat{X}_T, \quad \hat{Y}_B = -\hat{Y}_T, \quad \hat{Z}_B = \hat{Z}_T.$$

For the CSM SPS, LM DPS, and LM APS thrusters:

$$\begin{bmatrix} \hat{X}_B \\ \hat{Y}_B \\ \hat{Z}_B \end{bmatrix} = \begin{bmatrix} \cos P_g & 0 & \sin P_g \\ 0 & 1 & 0 \\ -\sin P_g & 0 & \cos P_g \end{bmatrix} \begin{bmatrix} \cos Y_g & -\sin Y_g & 0 \\ \sin Y_g & \cos Y_g & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{X}_T \\ \hat{Y}_T \\ \hat{Z}_T \end{bmatrix}.$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMCD
DATE 10/31/67 (EORP)
REV.
PAGE 11 of 29

Then,

$$\hat{X}_B = [\cos P_g \cos Y_g] \hat{X}_T + [-\cos P_g \sin Y_g] \hat{Y}_T + [\sin P_g] \hat{Z}_T$$

$$\hat{Y}_B = [\sin Y_g] \hat{X}_T + [\cos Y_g] \hat{Y}_T$$

$$\hat{Z}_B = [-\sin P_g \cos Y_g] \hat{X}_T + [\sin P_g \sin Y_g] \hat{Y}_T + [\cos P_g] \hat{Z}_T$$

(For the CSM SPS thruster, P_g is the pitch gimbal angle and Y_g is the yaw gimbal angle; for the LM DPS thruster, P_g is the pitch gimbal angle and Y_g is the roll gimbal angle; for the LM APS thruster, P_g is the pitch trim angle and Y_g is the roll trim angle.)

The following is the method for computing the unit thrust vector in RTCC inertial coordinate system ($\hat{A}_T$).

When the vehicle attitude angles are not given, the $\hat{A}_T$ is computed from the $\overline{\Delta V}$ of the maneuver in RTCC inertial coordinate system:

$$\hat{A}_T = \frac{\overline{\Delta V}}{|\overline{\Delta V}|}$$

When the vehicle attitude angles are given, the $\hat{A}_T$ is computed as follows:

For the CSM and LM RCS + X thrusters:

$$\hat{A}_T = \hat{X}_B$$

For the CSM and LM RCS - X thrusters:

$$\hat{A}_T = -\hat{X}_B$$

For the CSM SPS, LM DPS, and LM APS thrusters:

$$\begin{bmatrix} \hat{X}_T \\ \hat{Y}_T \\ \hat{Z}_T \end{bmatrix} = \begin{bmatrix} \cos Y_g & \sin Y_g & 0 \\ -\sin Y_g & \cos Y_g & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos P_g & 0 & -\sin P_g \\ 0 & 1 & 0 \\ \sin P_g & 0 & \cos P_g \end{bmatrix} \begin{bmatrix} \hat{X}_B \\ \hat{Y}_B \\ \hat{Z}_B \end{bmatrix}$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMCD

DATE 10/31/67 (EORP)

REV

PAGE 12 of 29

$$\hat{A}_T = \hat{X}_T = [\cos Y_g \cos P_g] \hat{X}_B + [\sin Y_g] \hat{Y}_B + [-\cos Y_g \sin P_g] \hat{Z}_B$$

For the S-IVB blowdown vent thruster:

$$\hat{A}_T = a_{X_B} * \hat{X}_B + a_{Y_B} * \hat{Y}_B + a_{Z_B} * \hat{Z}_B$$

$$\hat{A}_T = [\cos \theta_1] \hat{X}_B + [-\sin \theta_1 \cos \theta_2] \hat{Y}_B + [\sin \theta_1 \sin \theta_2] \hat{Z}_B$$

(Where a_{X_B} , a_{Y_B} , a_{Z_B} are the components of the unit thrust vector in the S-IVB body coordinate system, θ_1 is the S-IVB blowdown vent nozzle cant angle away from vehicle longitudinal axis, and θ_2 is the S-IVB blowdown vent nozzle cant angle in roll axis.)

The following is the method for computing the approximate Impulsive Maneuver Time.

If the Δt of the maneuver is not given, it is computed as follows:

$$\Delta t = \frac{W_{MI} * \Delta V}{T * g}$$

where

W_{MI} is the weight of the vehicle at maneuver initiation,

ΔV is the velocity increment of the maneuver,

T is the thrust level for the specified thruster,

g is the acceleration of gravity.

The approximate Impulsive Maneuver Time (IMT) is then obtained as follows:

$$IMT = t_B + \frac{\Delta t}{2}$$

where t_B is the time to begin the maneuver.

The following is the method for computing the components of the ΔV vector in the RTCC inertial coordinate system (ΔV_X , ΔV_Y , ΔV_Z) when the components

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMCD

DATE 10/31/67 (EORP)

REV

PAGE 13 of 29

of the ΔV vector in the IMU coordinate system (V_{g_X} , V_{g_Y} , V_{g_Z}) or the LVLH coordinate system (V_F , V_S , V_D) are input.

$$\begin{bmatrix} \Delta V_X \\ \Delta V_Y \\ \Delta V_Z \end{bmatrix} = \begin{bmatrix} X_{P_X} & Y_{P_X} & Z_{P_X} \\ X_{P_Y} & Y_{P_Y} & Z_{P_Y} \\ X_{P_Z} & Y_{P_Z} & Z_{P_Z} \end{bmatrix} \begin{bmatrix} V_{g_X} \\ V_{g_Y} \\ V_{g_Z} \end{bmatrix}$$

Then,

$$\overline{\Delta V} = V_{g_X} * \hat{X}_P + V_{g_Y} * \hat{Y}_P + V_{g_Z} * \hat{Z}_P$$

$$\begin{bmatrix} \Delta V_X \\ \Delta V_Y \\ \Delta V_Z \end{bmatrix} = \begin{bmatrix} X_{PHV_X} & Y_{PHV_X} & Z_{PHV_X} \\ X_{PHV_Y} & Y_{PHV_Y} & Z_{PHV_Y} \\ X_{PHV_Z} & Y_{PHV_Z} & Z_{PHV_Z} \end{bmatrix} \begin{bmatrix} V_F \\ V_S \\ V_D \end{bmatrix}$$

Then,

$$\overline{\Delta V} = V_F * \hat{X}_{PHV} + V_S * \hat{Y}_{PHV} + V_D * \hat{Z}_{PHV}$$

where $\hat{X}_P$, $\hat{Y}_P$, $\hat{Z}_P$ are the unit vectors representing the IMU coordinate system and $\hat{X}_{PHV}$, $\hat{Y}_{PHV}$, $\hat{Z}_{PHV}$ are the unit vectors representing the LVLH coordinate system.



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMMCD
 DATE 10/31/67
 REV
 PAGE 14 of 29

Table PMMMCD-1. System Parameters

Symbol in Program	Symbol In Master Tape	Type*	Units	Comments
	MCTCT5	D	lbs.	On-board computer thrust level for CSM RCS $\pm X$ thruster (2 quads)
	MCTCT6	D	lbs.	On-board computer thrust level for CSM RCS $\pm X$ thruster (4 quads)
	MCERMU	D	$(E_r^3 / \text{hr.})^{2/3}$	Square root of μ
	MCTSPG	D	radians	Pitch gimbal angle for CSM SPS thruster
	MCTAPG	D	radians	Pitch trim angle for LM APS thruster
	MCTDPG	D	radians	Pitch gimbal angle for LM DPS thruster
	MCTSYG	D	radians	Yaw gimbal angle for CSM SPS thruster
	MCTARG	D	radians	Roll trim angle for LM APS thruster
	MCTDRG	D	radians	Roll gimbal angle for LM DPS thruster
	MCTSD9	D	hrs.	Δt of ullage overlap for CSM SPS thruster
	MCTAD9	D	hrs.	Δt of ullage overlap for LM APS thruster
	MCTDD2	D	hrs.	Δt of ullage overlap for LM DPS thruster
G	MCEACG	D	$E_r / \text{hr.}^2$	Acceleration of gravity
	MCTJT7	D	lbs.	Thrust level for S-IVB blowdown vent thruster

*Type
 H - Fixed-point half word
 F - Fixed-point full word
 E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMCD

DATE 10/31/67 (EORP)

REV

PAGE 15 of 29

Table PMMMCD-1. System Parameters, (Cont'd.)

Symbol in Program	Symbol in Master Tape	Type *	Units	Comments
	MCTLT5	D	lbs.	On-board computer thrust level for LM RCS $\pm$ X thruster (2 quads)
	MCTLT6	D	lbs.	On-board computer thrust level for LM RCS $\pm$ X thruster (4 quads)
	MCTST9	D	lbs.	On-board computer thrust level for CSM SPS thruster
	MCTAT9	D	lbs.	On-board computer thrust level for LM APS thruster
	MCTDT9	D	lbs.	On-board computer thrust level for LM DPS thruster
SINT1	MCVST1	D		Sine of S-IVB blowdown vent nozzle cant angle away from vehicle longitudinal axis
COST1	MCVCT1	D		Cosine of S-IVB blowdown vent nozzle cant angle away from vehicle longitudinal axis
SINT2	MCVST2	D		Sine of S-IVB blowdown vent nozzle cant angle in roll axis
COST2	MCVCT2	D		Cosine of S-IVB blowdown vent nozzle cant angle in roll axis

*Type

H - Fixed-point half word
F - Fixed-point full wordE - Single-precision (floating-point)
D - Double-precision (floating-point)
X - Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMCD
DATE 10/31/67 (EORP)
REV
PAGE 16 of 29

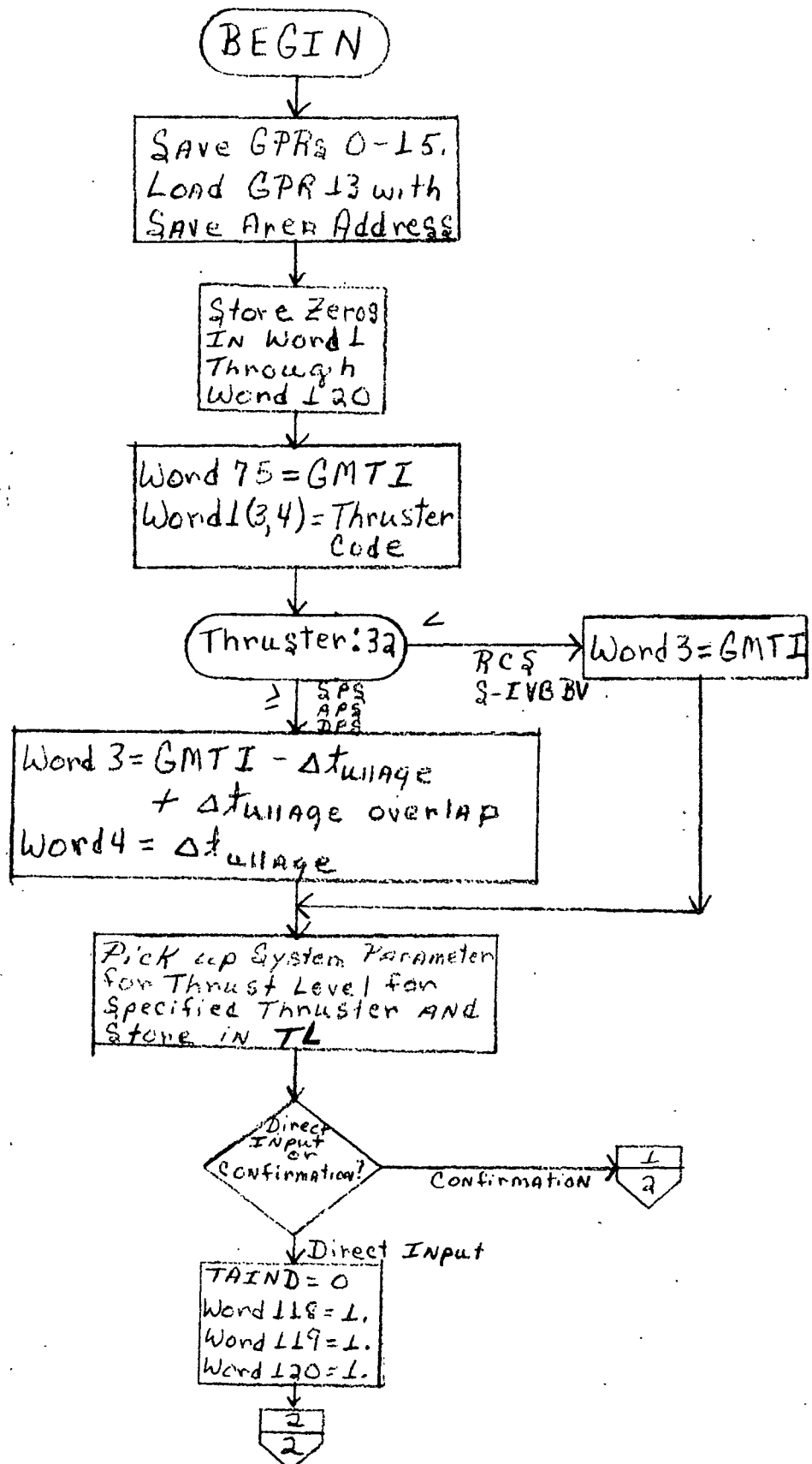


Figure PMMMCD-1. Flowchart

(Sheet 1 of 14)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

2. PMMMCD
 DATE 10/31/67 (EORP)
 REV.
 PAGE 17 of 29

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

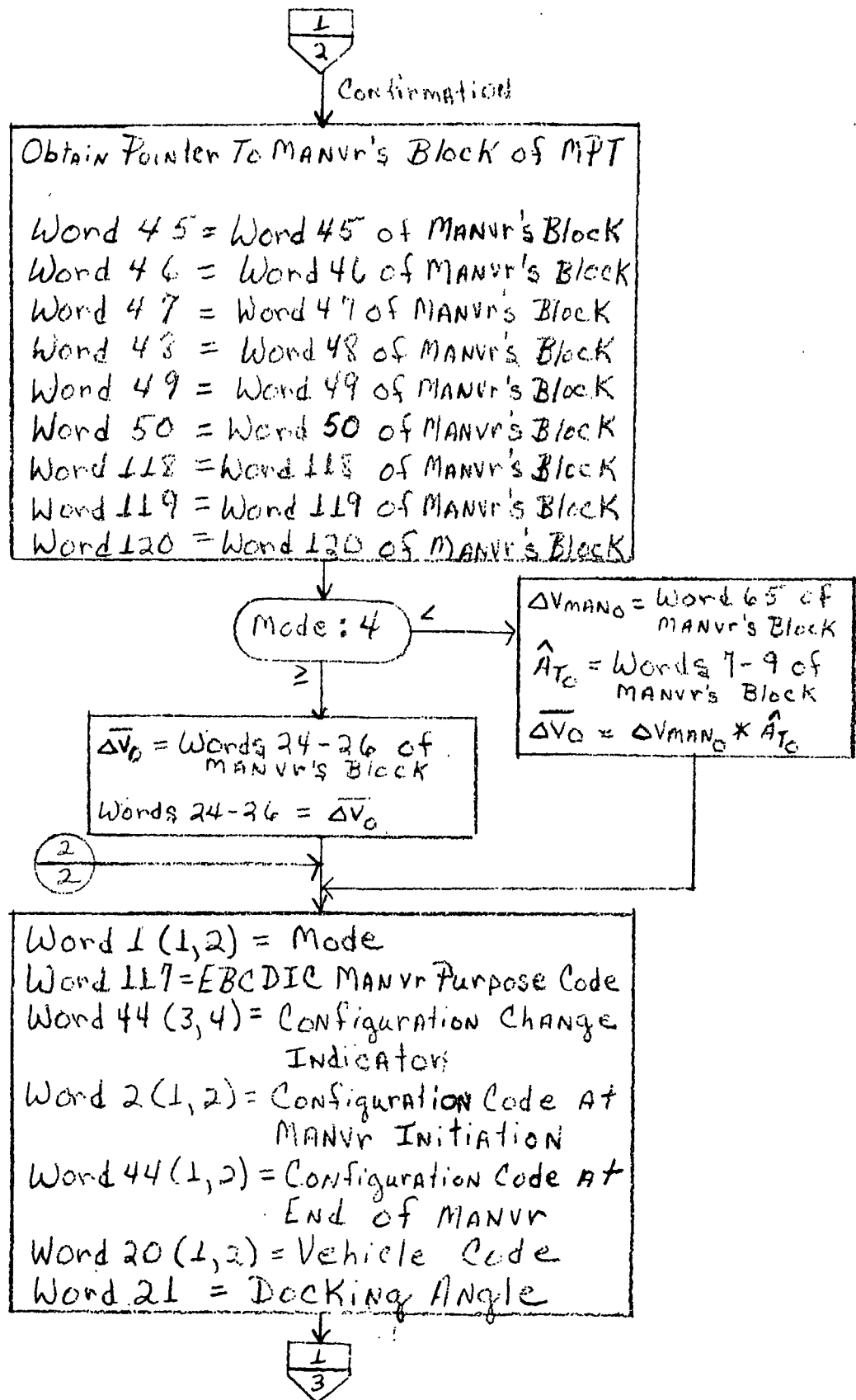


Figure PMMMCD-1. Flowchart

(Sheet 2 of 14)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMCD
DATE 10/31/67 (EORP)
REV
PAGE 18 of 29

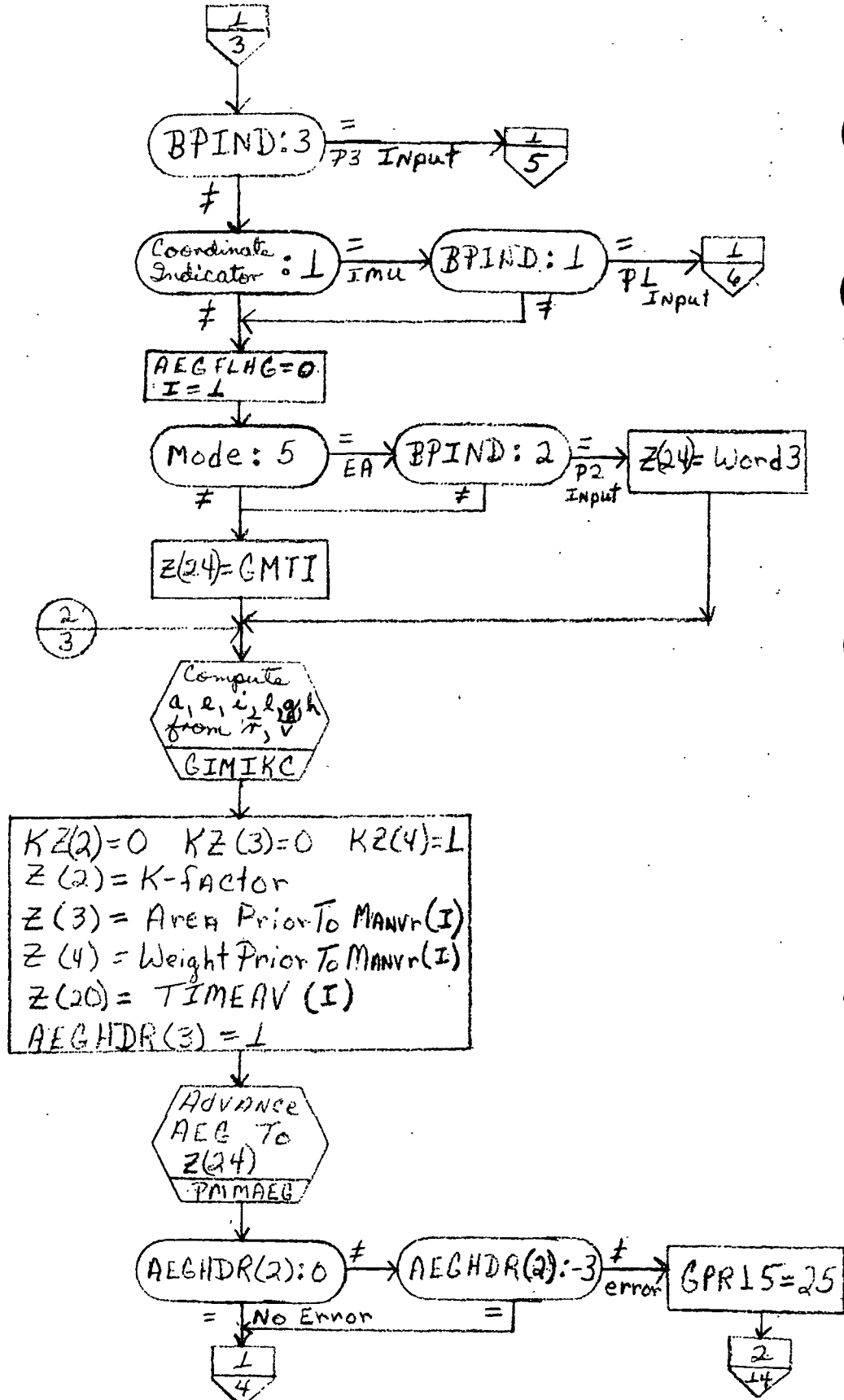


Figure PMMMCD-1. Flowchart

(Sheet 3 of 14)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMCD

DATE 10/31/67 (EORP)

REV

PAGE 19 of 29

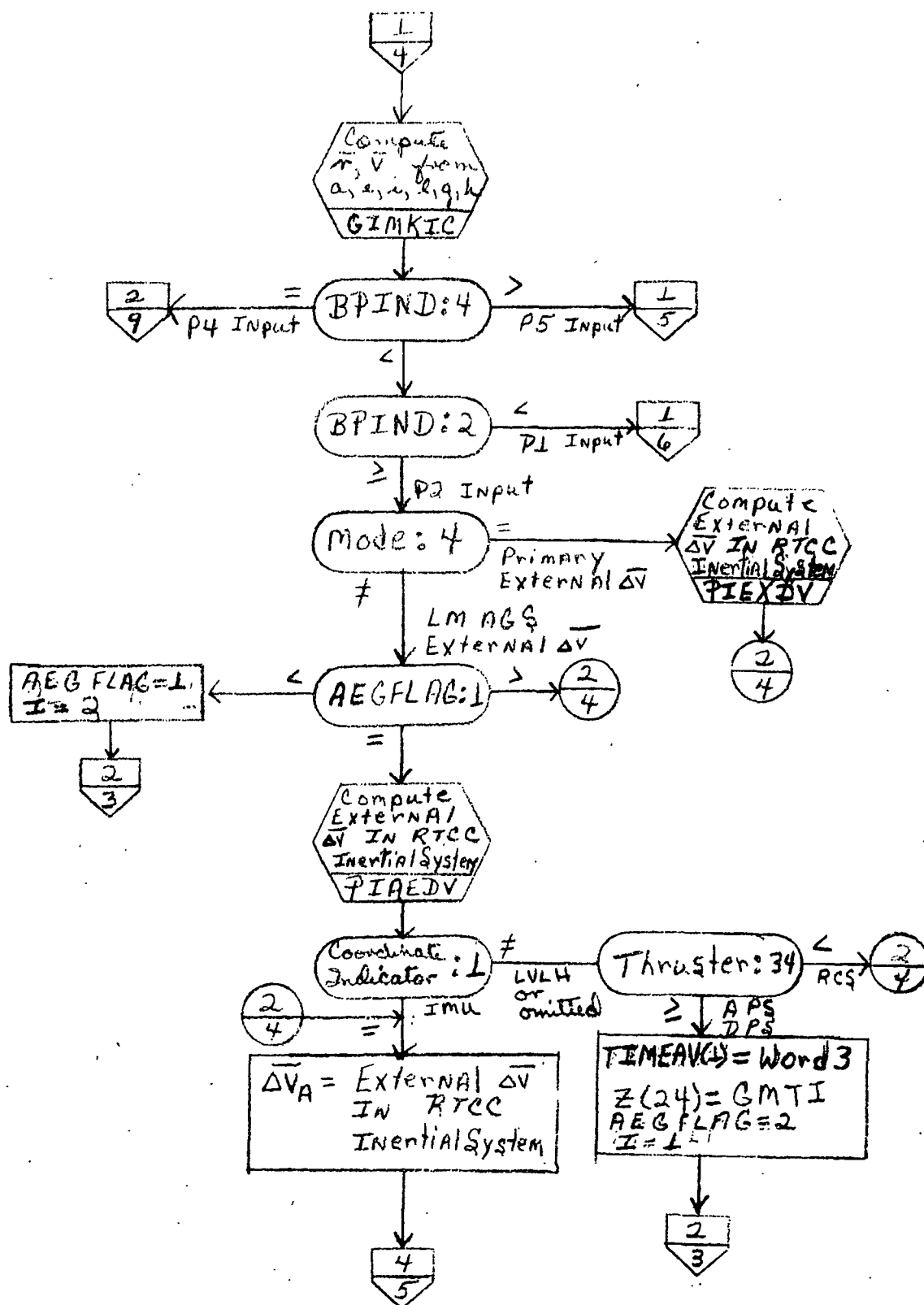


Figure PMMMCD-1. Flowchart

(Sheet 4 of 14)

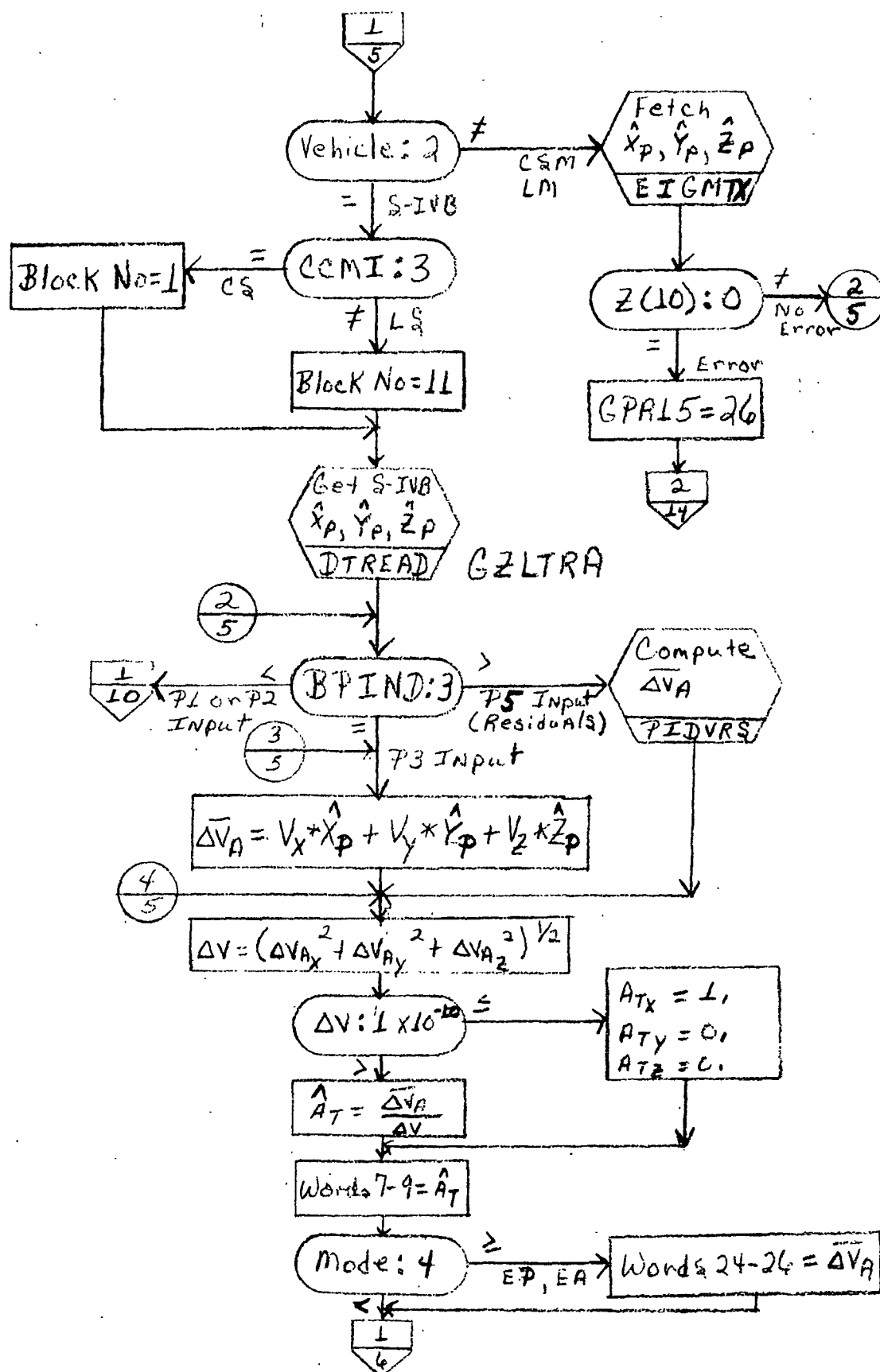


Figure PMMMCD-1. Flowchart

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

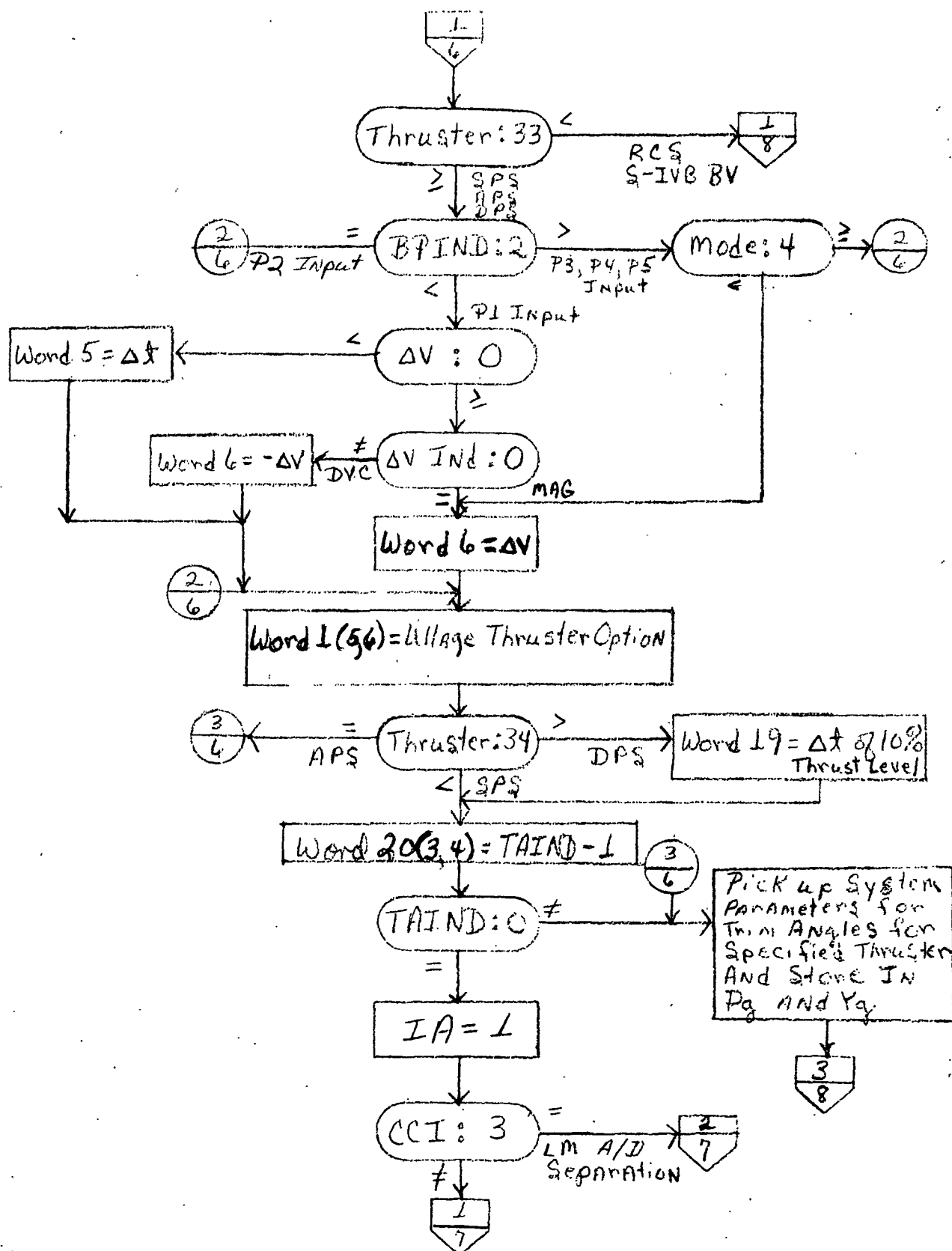


Figure PMMMCD-1. Flowchart

(Sheet 6 of 14)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

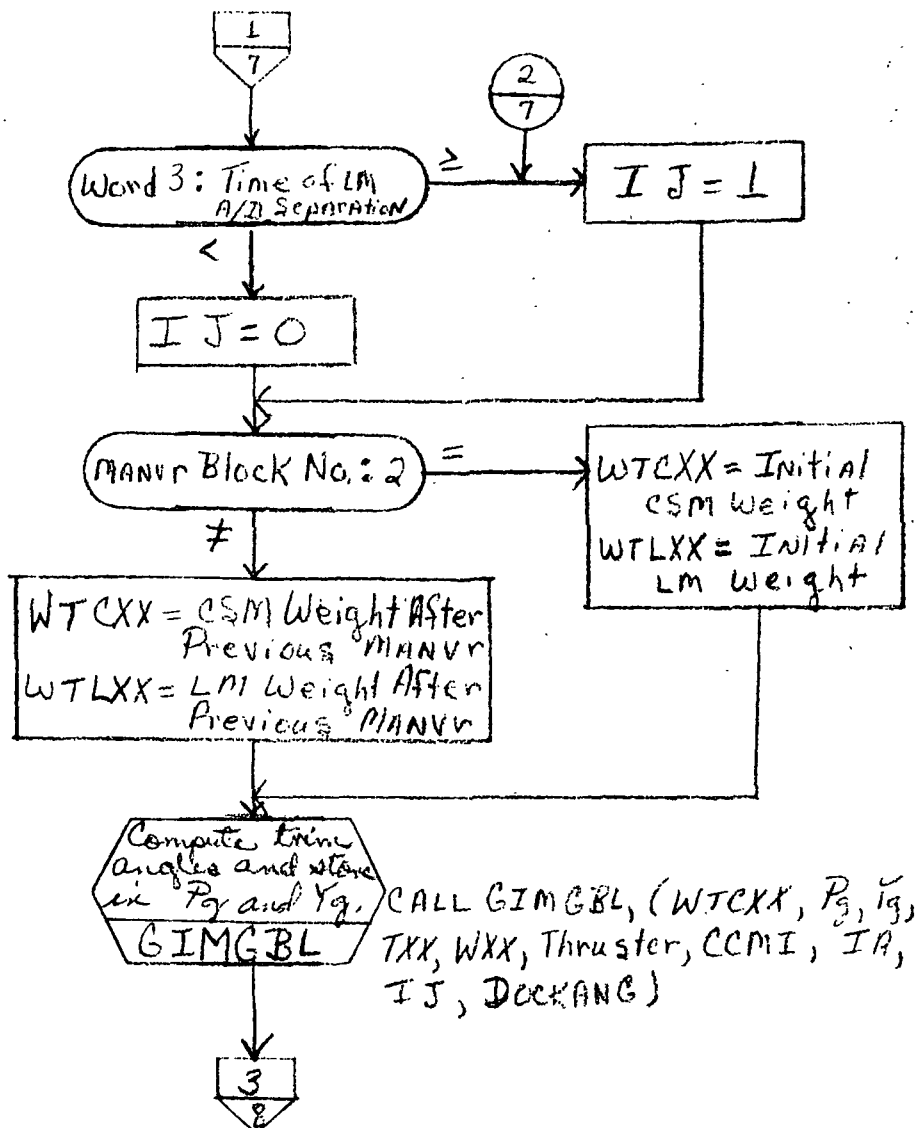
 2. PMMMCD
 DATE 10/31/67 (EOR)
 REV
 PAGE 22 of 29


Figure PMMMCD-1. Flowchart

(Sheet 7 of 14)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMCD

DATE 10/31/67 (EORP)

REV

PAGE 23 of 29

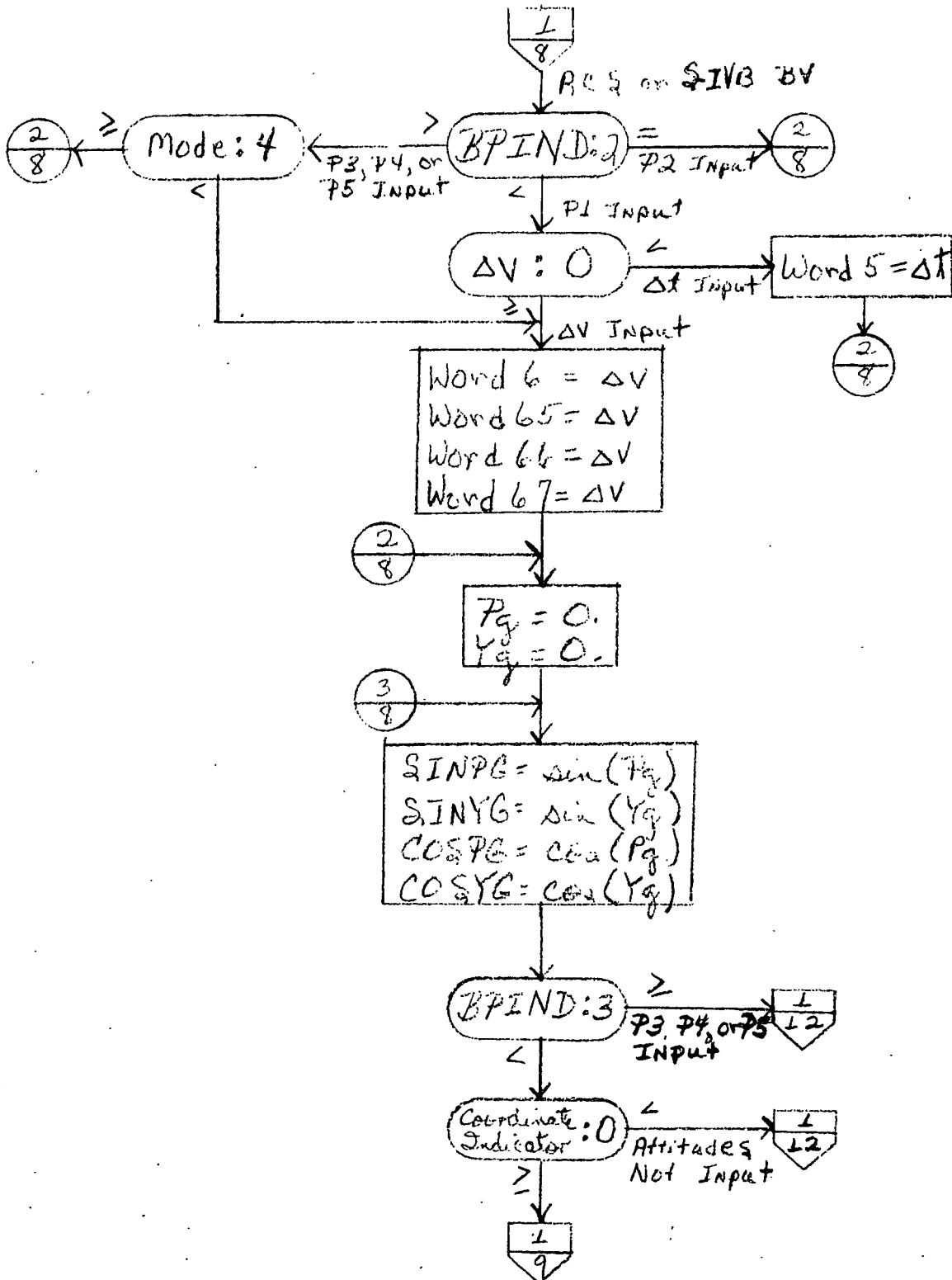


Figure PMMMCD-1. Flowchart

(Sheet 8 of 14)

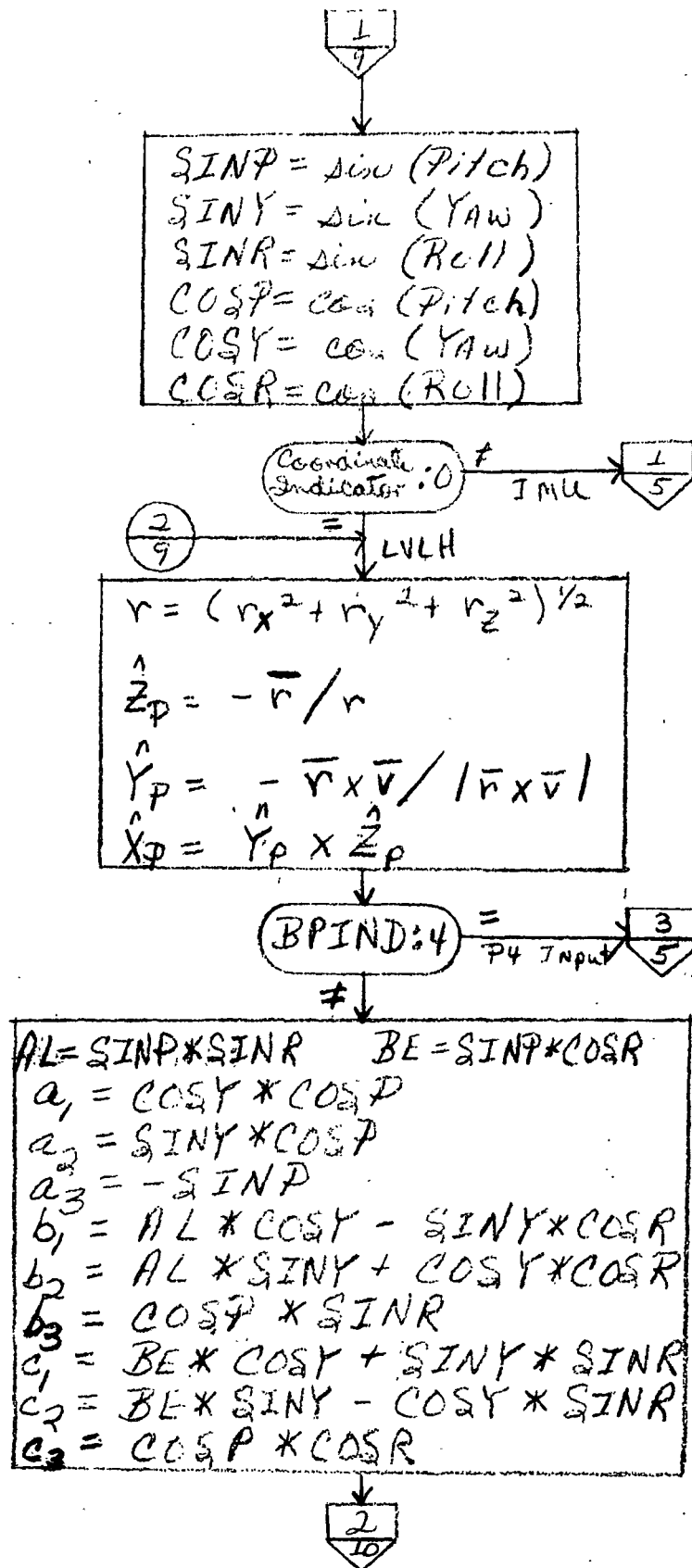


Figure PMMMCD-1. Flowchart

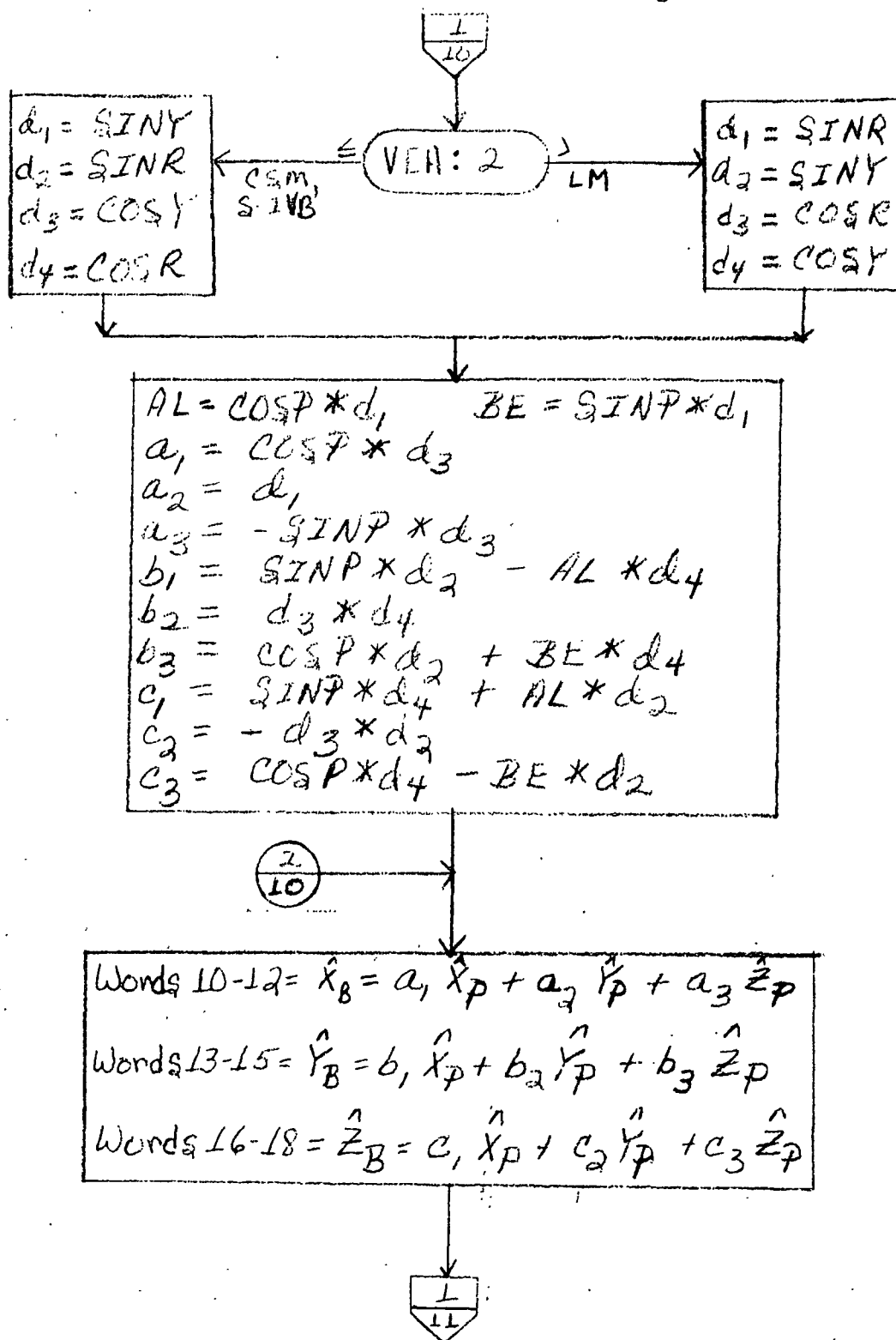


Figure PMMMCD-1, Flowchart

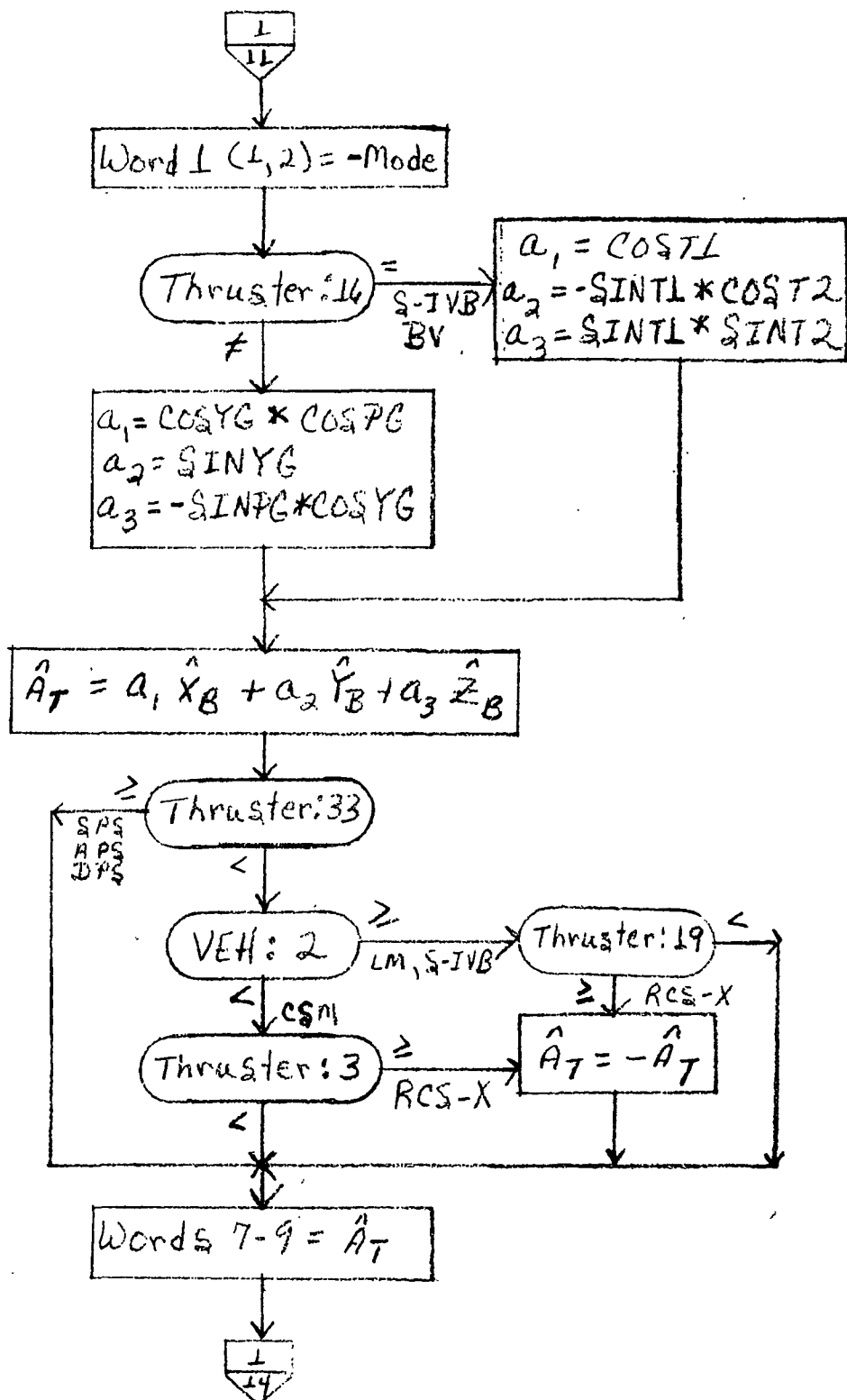


Figure PMMMCD-1. Flowchart

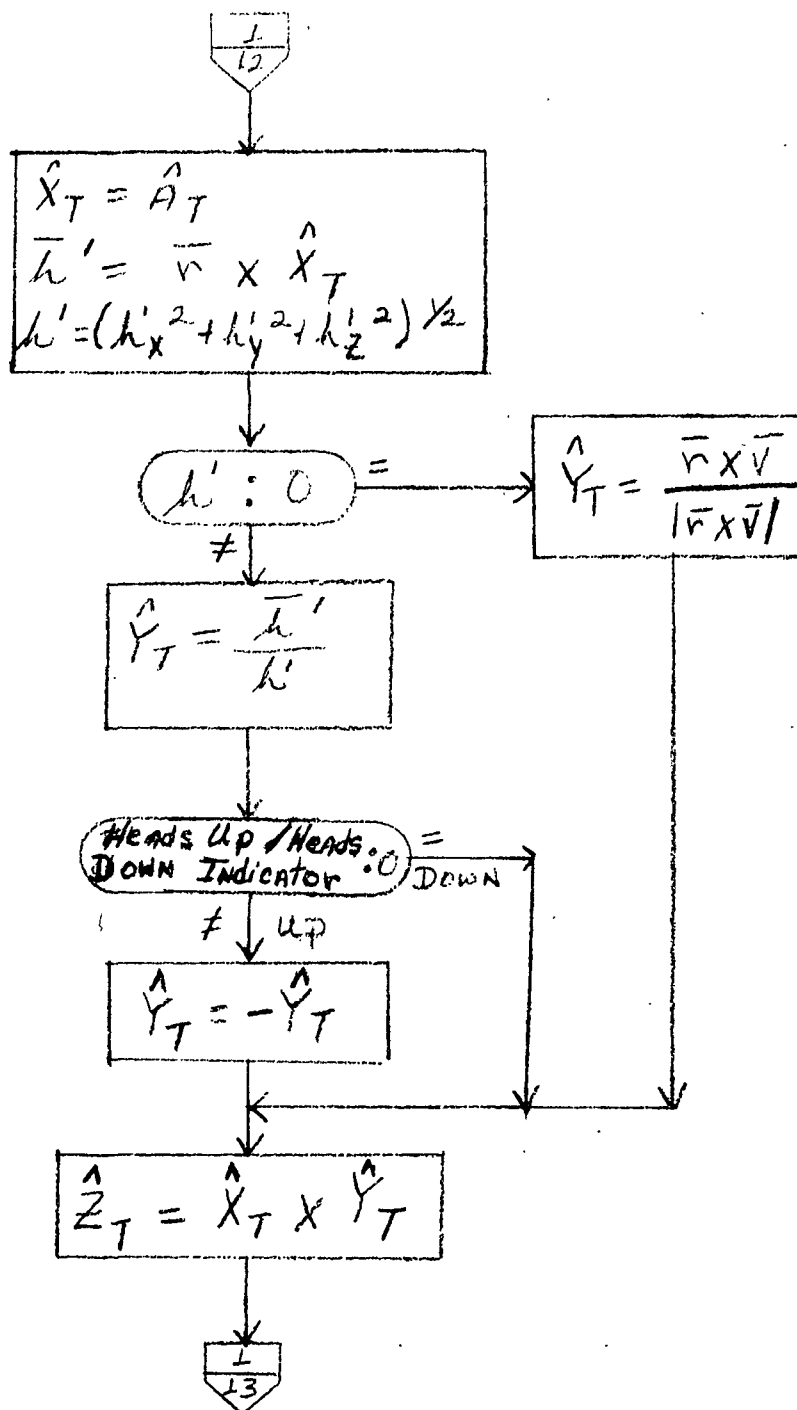


Figure PMMMCD-1. Flowchart

(Sheet 12 of 14)

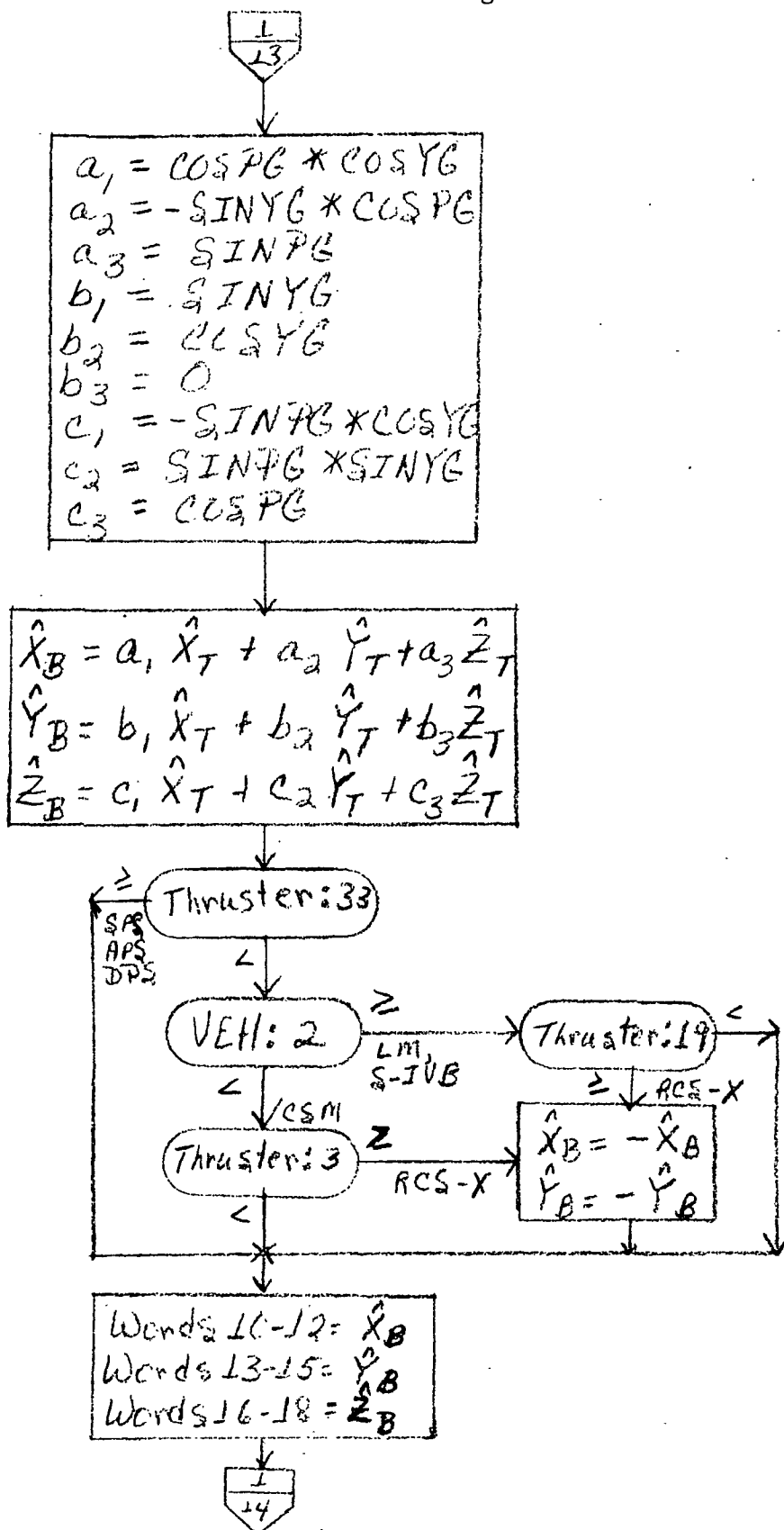


Figure PMMMCD-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMCD

DATE 10/31/67 (EORP)

REV

PAGE 49 of 29

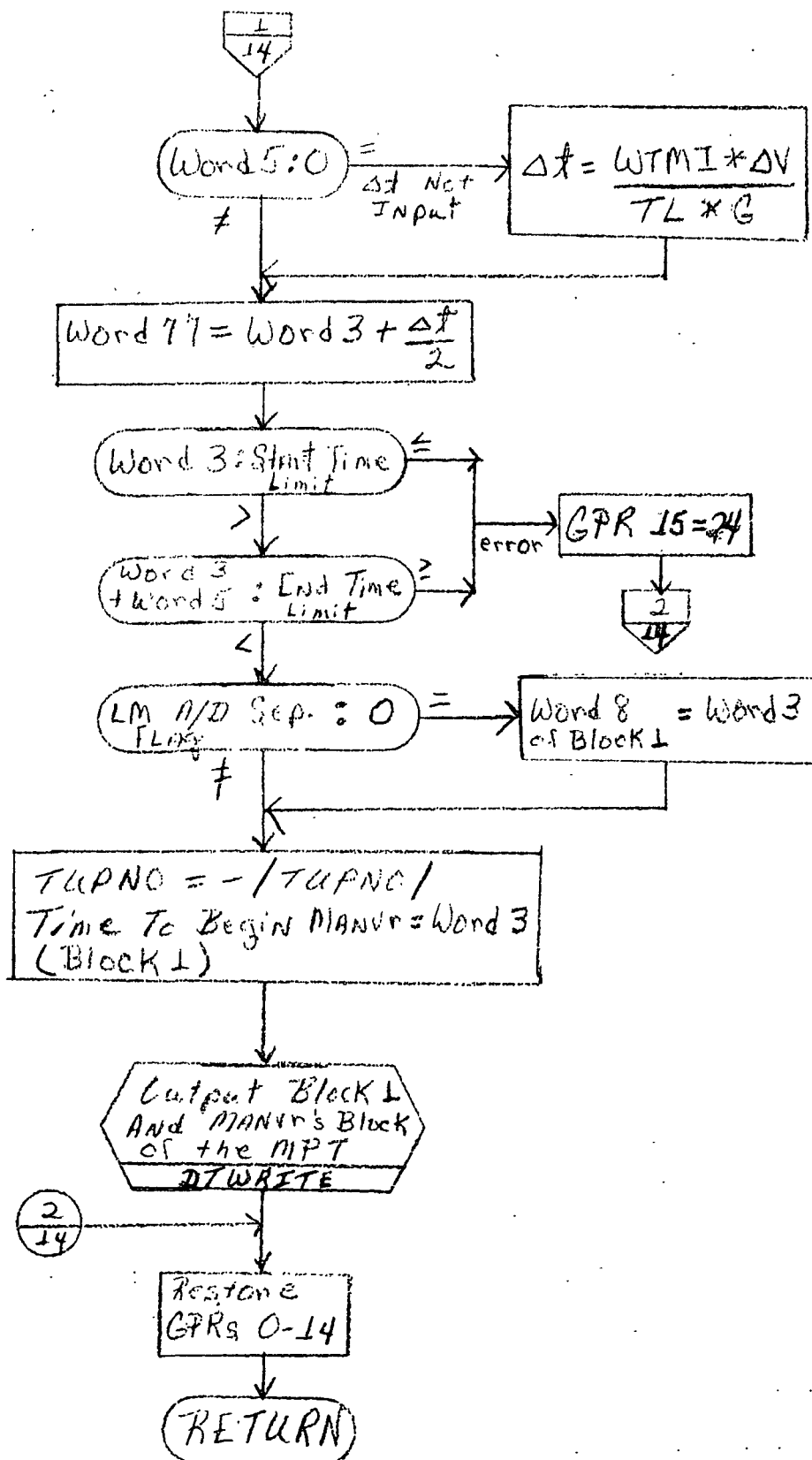


Figure PMMMCD-1. Flowchart

(Sheet 14 of 14)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMMLC
 DATE 9/19/67 (EORP)
 REV
 PAGE 1 of 21

- a. Symbol — PMMMLC
- b. Program Name — Maneuver Line Control Module
- c. Programming Group — Rendezvous Planning
- d. Programmer — J. Steed
- e. Language — Assembler
- f. Data Tables Used — PZMPTCSM, PZMPTLEM, PZCMLD, GZGENCSN
- g. MED's Used — K10 (Initialization MED)
- h. Writeup Completion Date — 6/15/67
- i. System Parameters — Table PMMMLC-1
- j. Parameter List — Table PMMMLC-2
- k. Description of Terms — Table PMMMLC-3
- l. Flowchart — Figure PMMMLC-1

FUNCTION

PMMMLC controls and computes the maneuver line definition and relocates the maneuver line along the line of apsides or along the optimum maneuver line.

USAGE

- a. Calling Sequence — LINK EP=PMMMLC, PARAM=(TIME, IND, AEGADDR, SPADDR)
- b. Inputs — PZCMLD; maneuver line vehicle ID in first halfword of AEG header (1=CSM, 3=LM); see attached parameter list.
- c. Outputs and Exit — PZCMLD; AEG Blocks; MLD, NAG, STAIID at beginning of subpool; return to calling program
- d. Other Programs Used — PMSVEC, PMXSPT, PMMAEG, DSQRT, DATAN, DCOS, DSIN
- e. Data Restrictions and Limits — PZCMLD must be initialized and ML must be defined for relocation.
- f. Error — If PMMMLC detects an error, a (-1) will be returned to the calling program in the second halfword of the AEG header, and an appropriate message will be printed on-line.

METHOD

PMMMLC may be entered with any of eight possible options.

Options 0-3: The maneuver line definition, the NAG, and the updated

Approval	Approval
<i>J. A. Steed</i>	<i>A. Amietti</i>

AEG blocks will be returned by PMMMLC when given one of the following: (1) a threshold time, a specified argument of latitude, and an NA; (2) a specified time and an NA; (3) a specified argument of latitude (threshold time taken as the greater of present time or target insertion time) and an NA; or (4) a specified elapsed time from target insertion time and an NA.

Option 4: The present MLD is relocated along the line of Apsides of the chaser as defined by the chaser elements corresponding to the present MLD time. The NA remains the same or is increased or decreased by .5 depending on the amount and direction of shift in the maneuver line which may have occurred during relocation. The NAG is also computed.

Option 5: Except for the fact that relocation occurs along a given argument of latitude, the optimum relocation ($\dot{r}$) is the same as the relocation along the line of Apsides. This option is used in conjunction with co-elliptic orbits. It returns the optimum point (in terms of fuel required) to move the chaser from its present orbit to a co-elliptic orbit based on the present target orbit. The formulas used to compute the optimum point are as follows:

$$a_{ce} = a_1 - \Delta H_{DES}$$

$$e_{ce} = a_1 e_1 / a_{ce}$$

$$k_{ce} = e_{ce} \sqrt{\mu} / \sqrt{a_{ce} (1 - e_{ce}^2)}$$

$$k_2 = e_2 \sqrt{\mu} / \sqrt{a_2 (1 - e_2^2)}$$

$$\alpha' = g_2 - g_1 \cong f_{ce} - f_2$$

$$\dot{r}_{ce} = k_{ce} \sin f_{ce}$$

$$\dot{r}_2 = k_2 \sin f_2$$

where ΔH_{DES} = desired co-elliptic difference

1 refers to the target

2 refers to the chaser

ce refers to co-elliptic

By setting $r_{ce} = r_2$ and substituting $f_{ce} = f_2 + \alpha$ it yields:

$$f_r = \tan^{-1} \left(\frac{k_{ce} \sin \alpha}{k_2 - k_{ce} \cos \alpha} \right) \text{ and } U_r = f_r + g_2$$

$$0 \leq f_r < 2\pi$$

$$0 \leq U_r < 2\pi$$

The computed desired argument of latitude (U_r) is the optimum point in the present chaser orbit to perform the maneuver. The elements are advanced to U_r and the height difference between the chaser and target orbits is obtained. The elements are then advanced to $U_r + \pi$ and the height difference is again obtained. The end with the smallest height difference is designated the apogee end of the optimum maneuver line.

Option 7: When a specified time is input, the elements of both vehicles are advanced to that time, and the updated AEG blocks are returned.

Table PMMLC-1. System Parameters

[illegible]

*Type
H — Fixed-point half word
F — Fixed-point full word
E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMLC
DATE 9/19/67 (EORP)
REV
PAGE 5 of 21

Table PMMMLC-2. Parameter List

QUEID —

QNAME —

From —

To-Task —

Purpose —

EP —

Byte No.	Type*	Units	Value / Description / Comments
0-3	F	hrs.	Address of Present Time, or any desired time for Option 7*
4-7	F		Option Indicator 1 = Definition, Options 0-3 2 = Relocation along Apsides, Option 4 3 = RDOT Relocation, Option 5 4 = Current Definition, Option 6 5 = Option 7
8-11	F		Address of AEG Blocks
12-15	F		Address of Subpool
* Present time (in hrs.) must be followed by another doubleword containing the time to be used to obtain elements for the second vehicle.			

*Type
H — Fixed-point half word
F — Fixed-point full word

E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMLC
DATE 9/19/67 (EORP)
REV
PAGE 6 of 21

Table PMMMLC-3. Description of Terms

Term	Description
CHASER	Maneuver Line Vehicle
ΔH	Height difference of chaser and target orbits
ΔN	Rev. Counter for AEG
IND	Indicator passed as second element of parameter list; 1 = Options 0-3; 2 = Option 4; 3 = Option 5; 4 = Option 6; 5 = Option 7
MLD	Maneuver Line Definition
MLDIND	First word of PZCMLD, input by MED; 0 = Option 0, Threshold Time; 1 = Option 1, T, NA; 2 = Option 2, U, NA; 3 = Option 3, Δt , NA.
NA	Counter for the chaser crossings of the maneuver line
NAG	Counter for the target crossings of the maneuver line
T_{INS}	Target Insertion Time
T_{MAN}	Time of last frozen chaser maneuver
T_P	Present Time
T_{REF}	Target Reference Time
TARGET	Non-maneuver Line Vehicle
TNA	Time of the MLD
U	Argument of Latitude
UOC	AEG Input Argument of Latitude
U_i	Argument of Latitude of Optimum i
ΔT	Time Increment

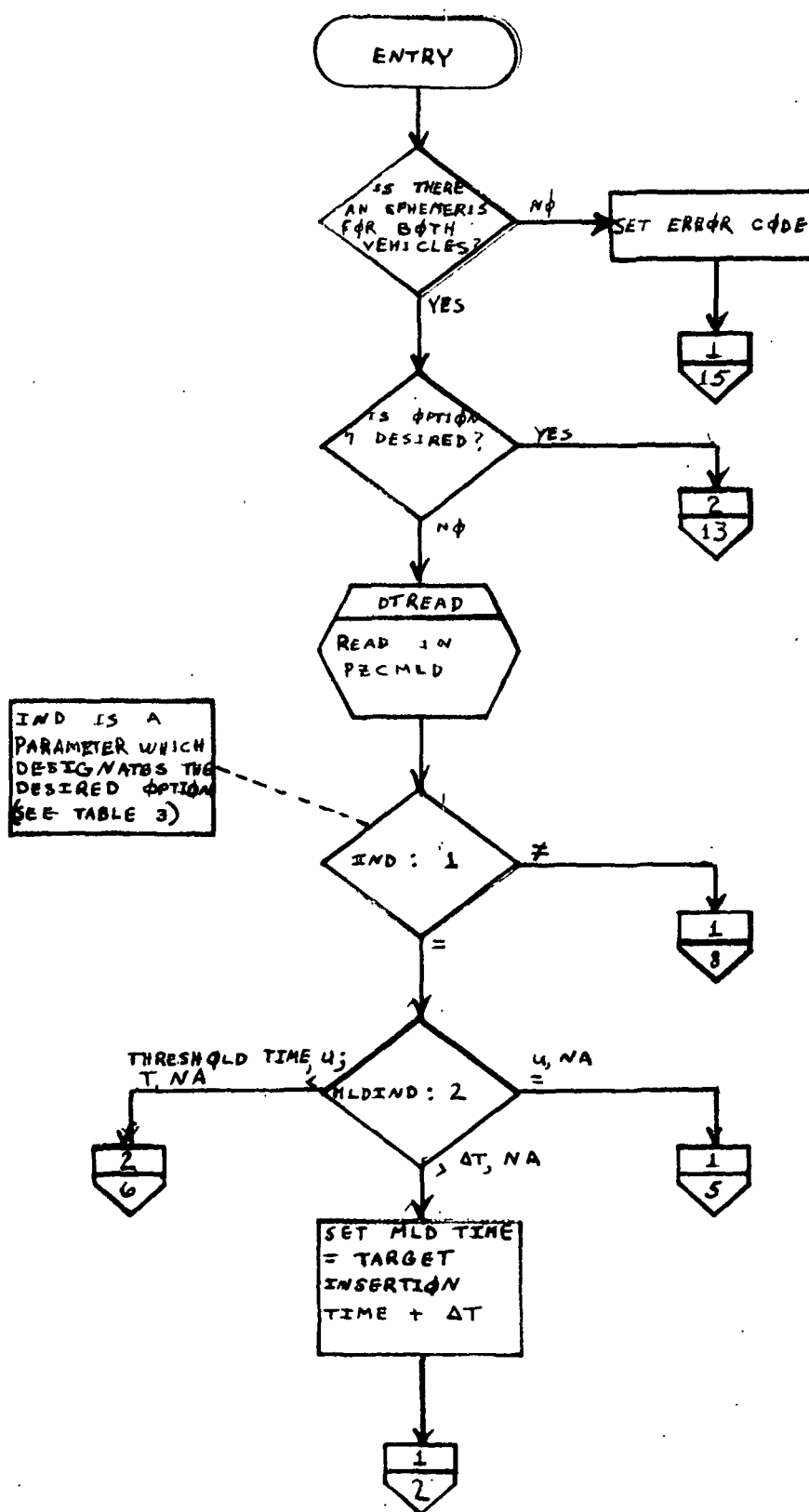


Figure PMMMLC-1. Flowchart

(Sheet 1 of 15)

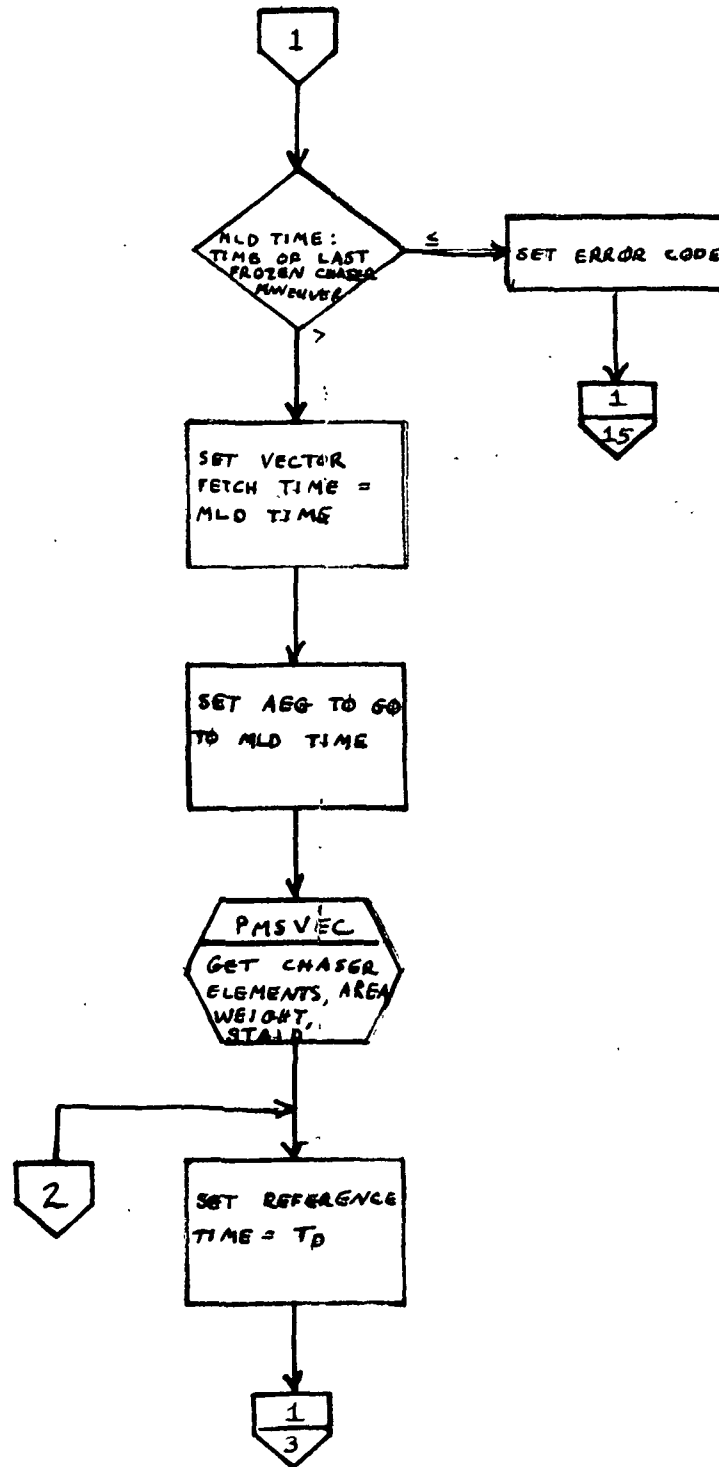


Figure PMMMLC-1. Flowchart

(Sheet 2 of 15)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMLC
 DATE 9/19/67 (EORP)
 REV
 PAGE 9 of 21

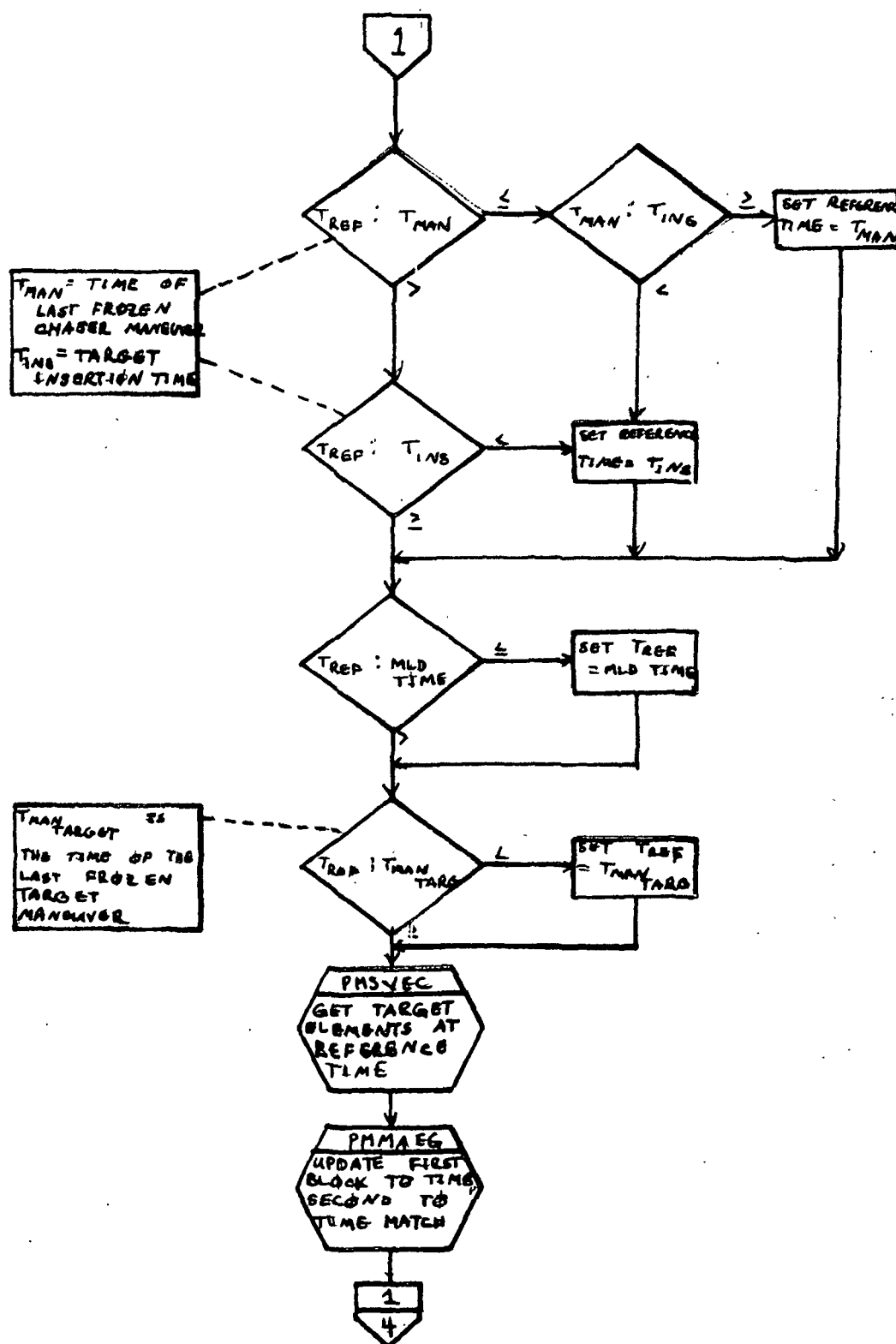


Figure PMMMLC-1. Flowchart

(Sheet 3 of 15)

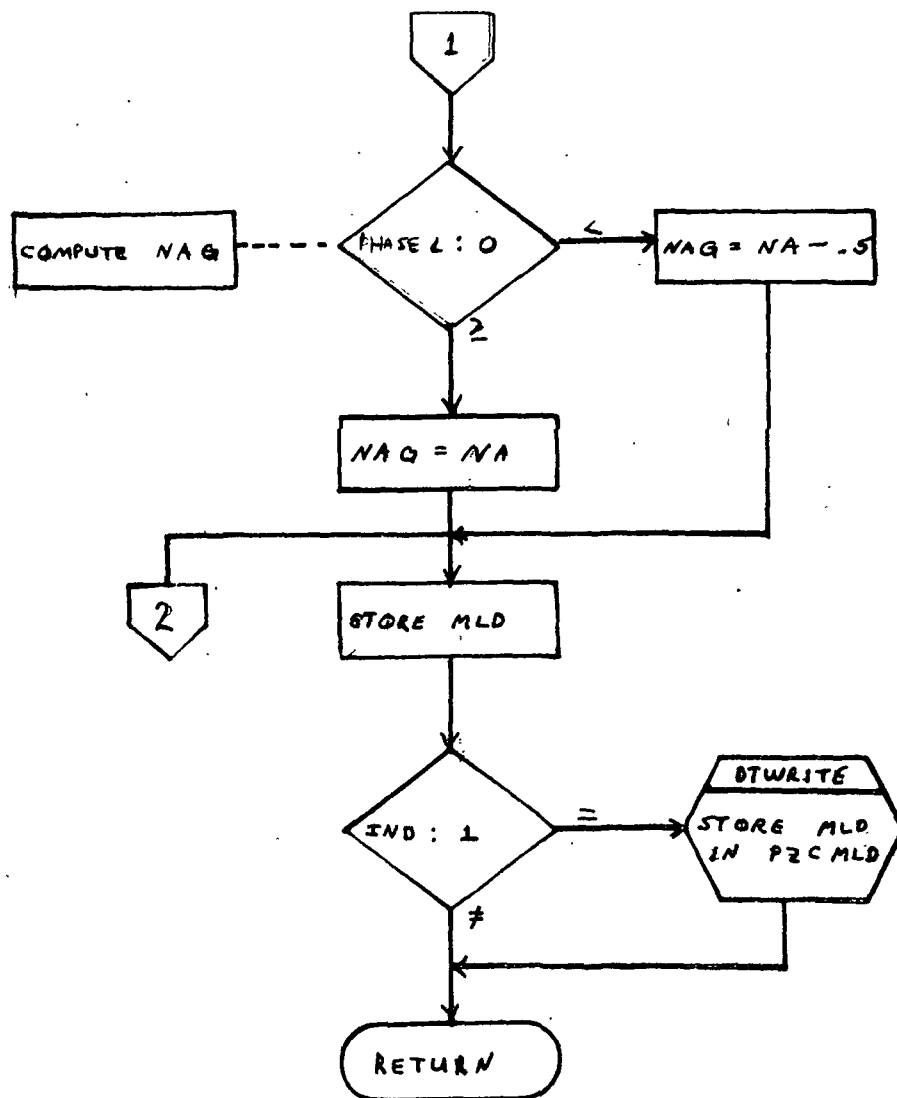


Figure PMMMLC-1. Flowchart

(Sheet 4 of 15)

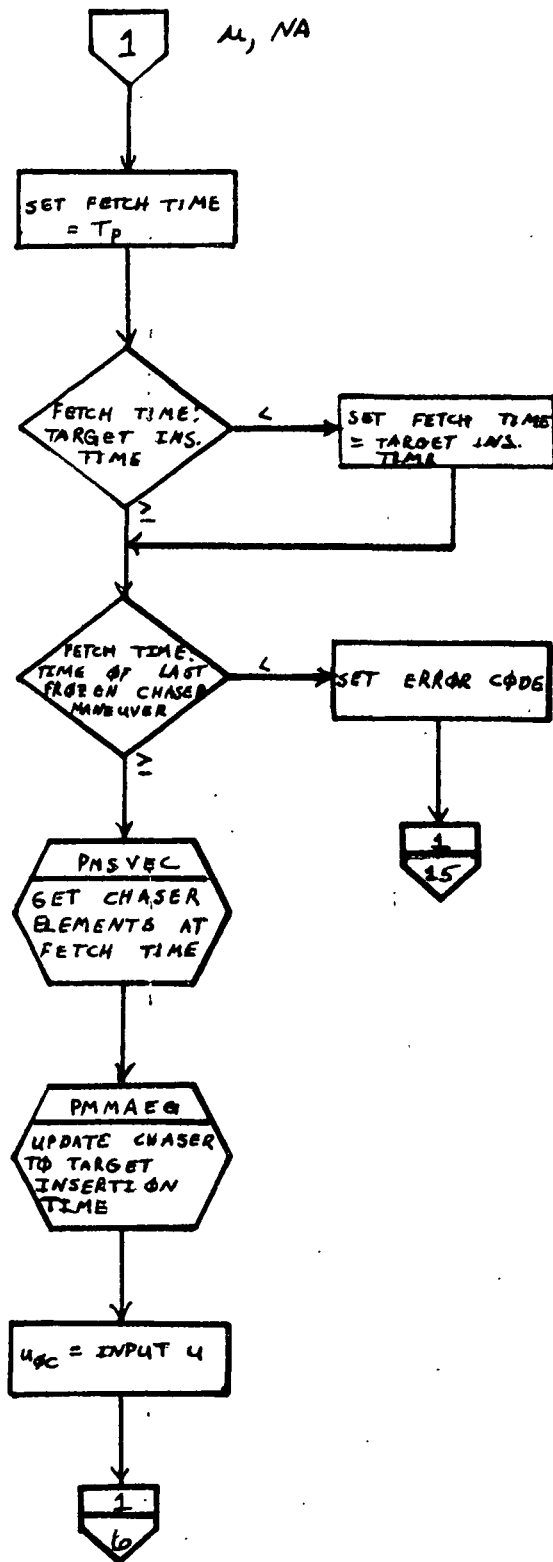


Figure PMMMLC-1. Flowchart

(Sheet 5 of 15)

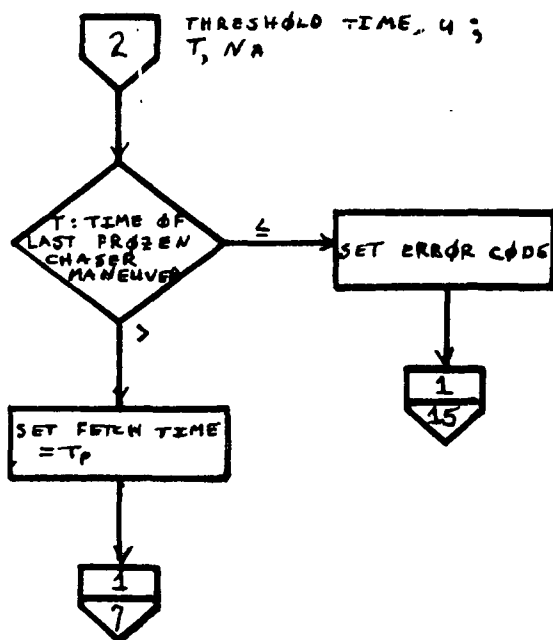
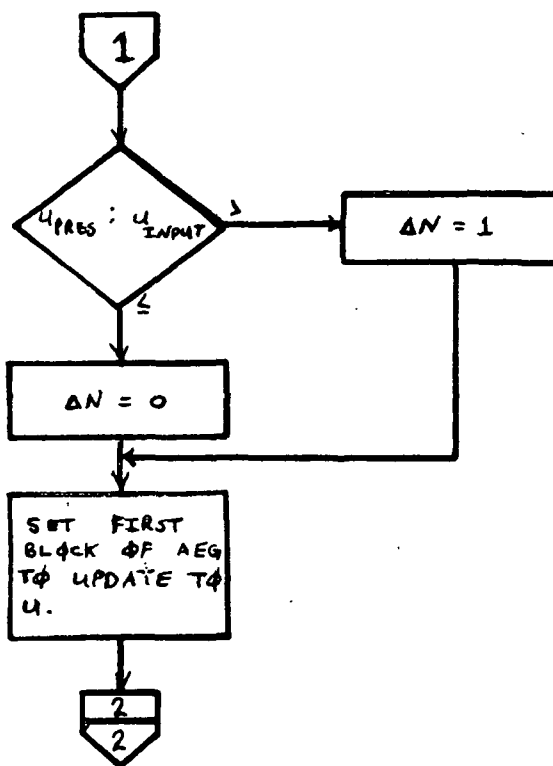


Figure PMMMLC-1. Flowchart

(Sheet 6 of 15)

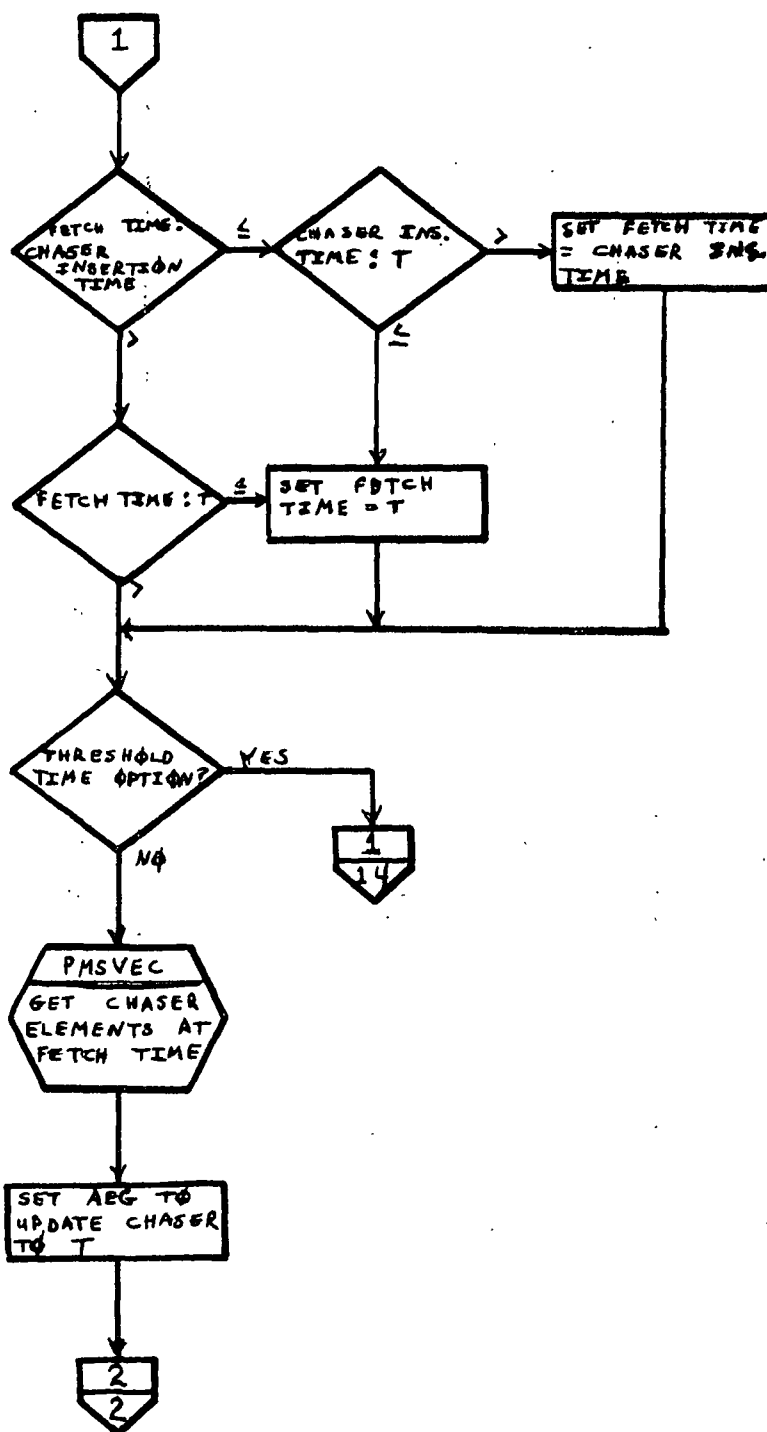


Figure PMMMLC-1. Flowchart

(Sheet 7 of 15)

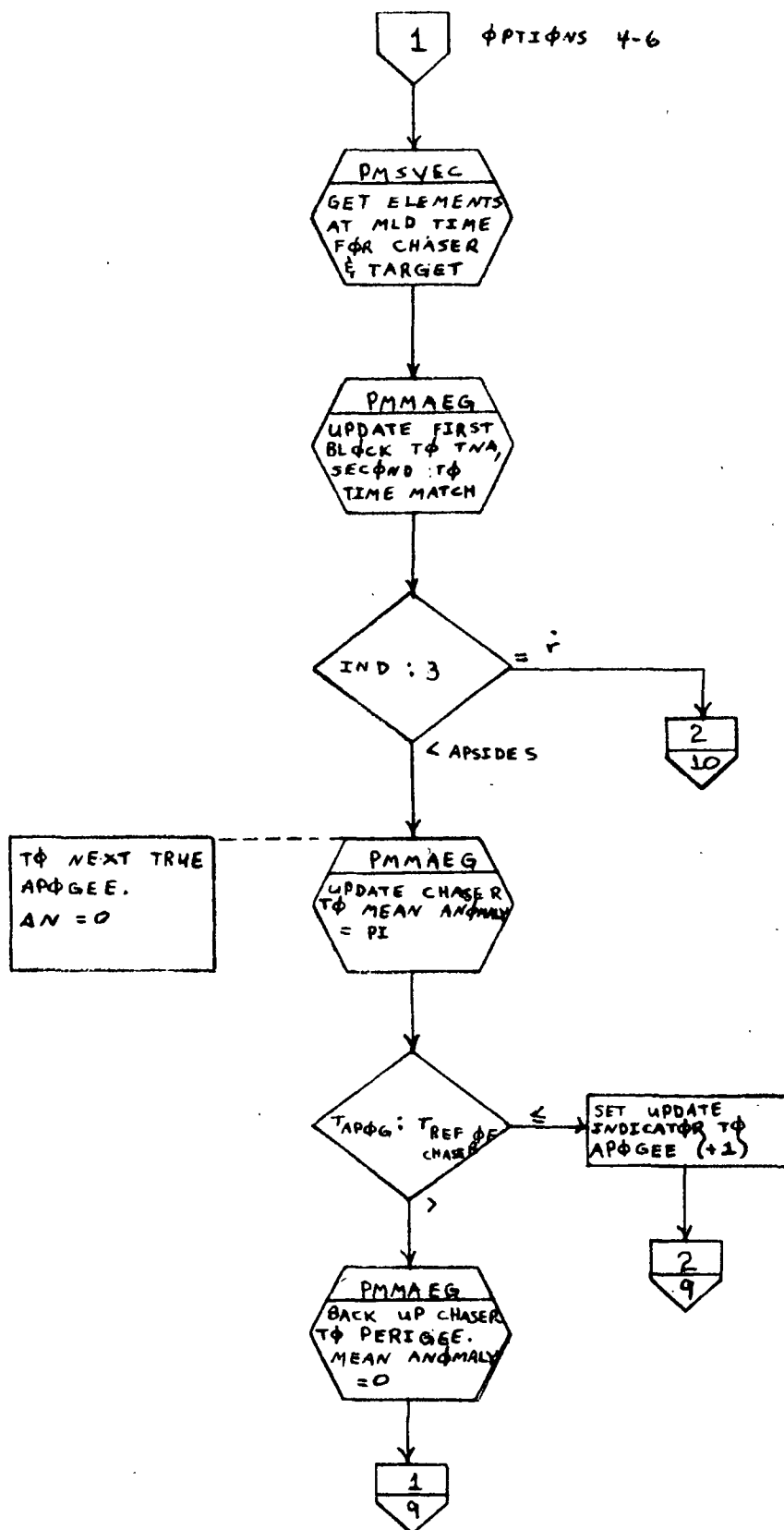


Figure PMMMLC-1. Flowchart

(Sheet 8 of 15)

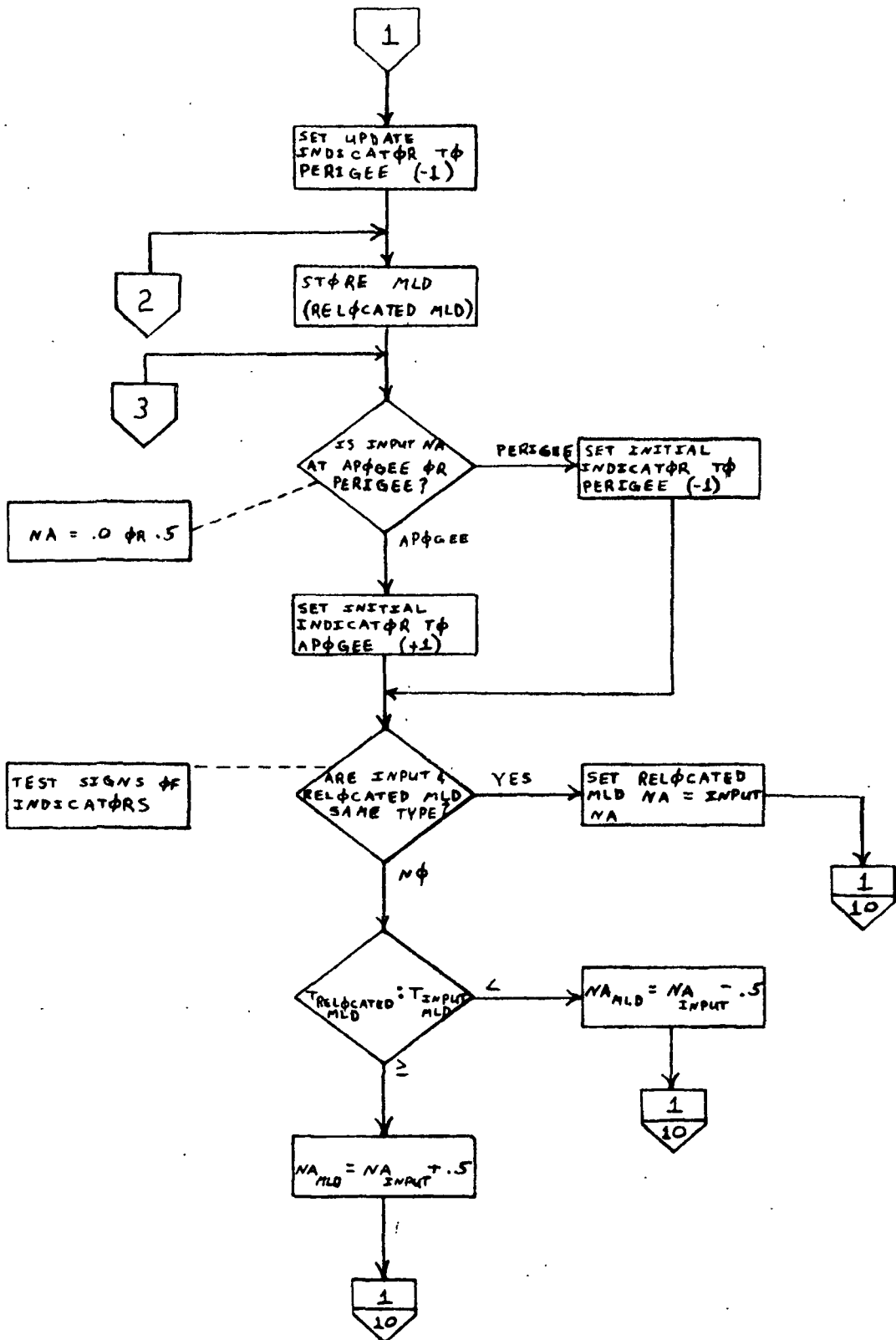
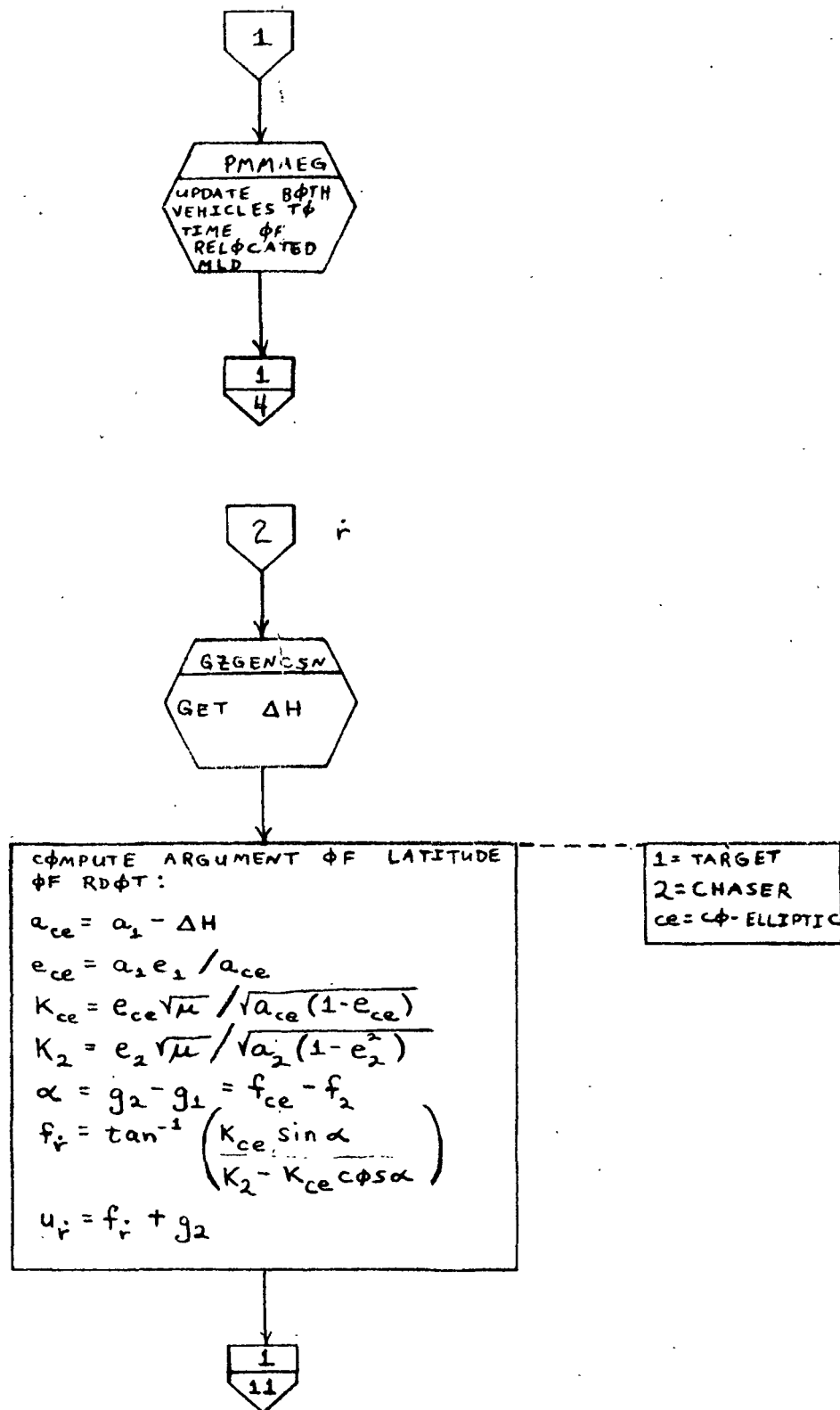


Figure PMMMLC-1. Flowchart

Figure PMMMLC-1. Flowchart
(Sheet 10 of 15)

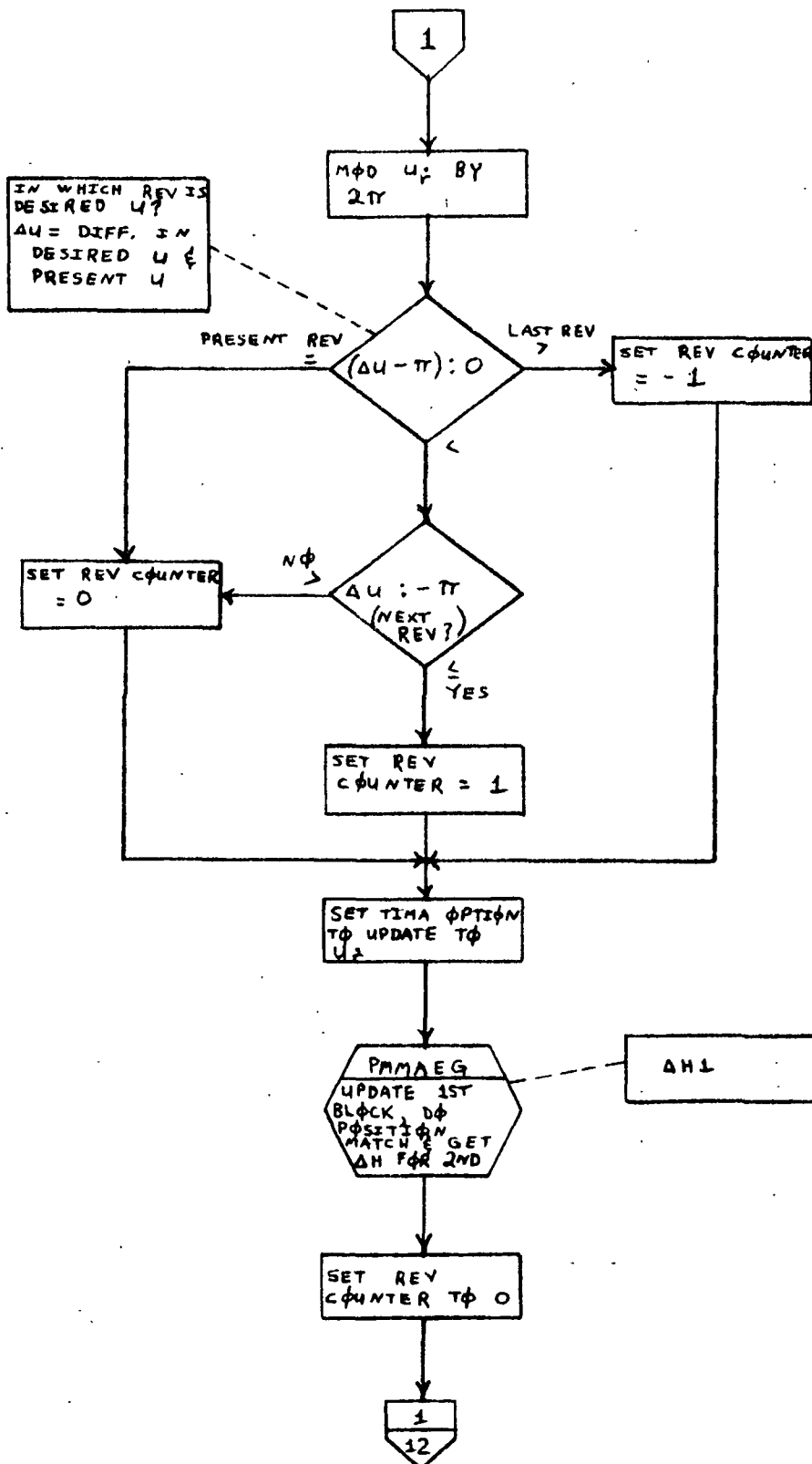


Figure PMMMLC-1. Flowchart

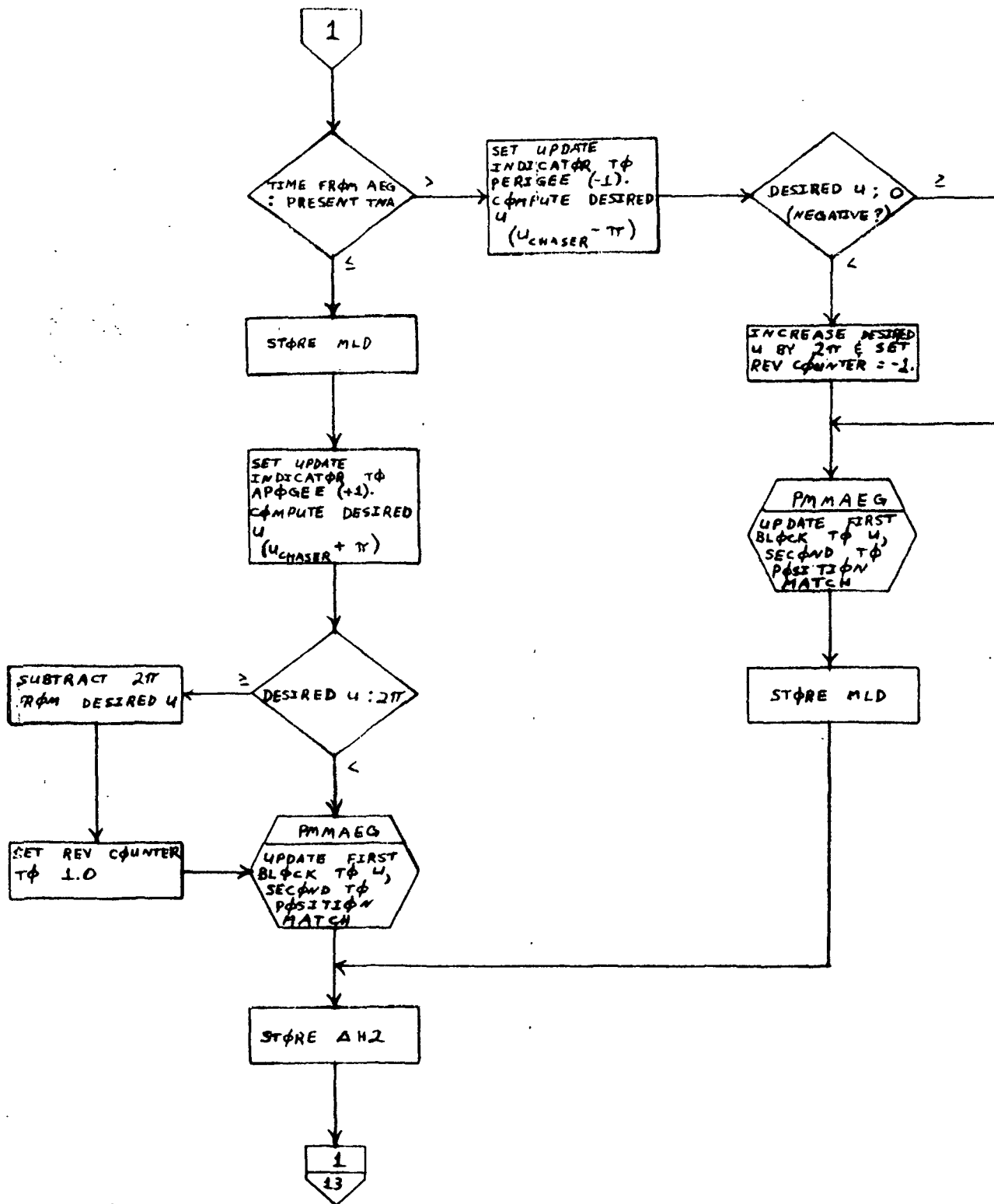


Figure PMMMLC-1. Flowchart

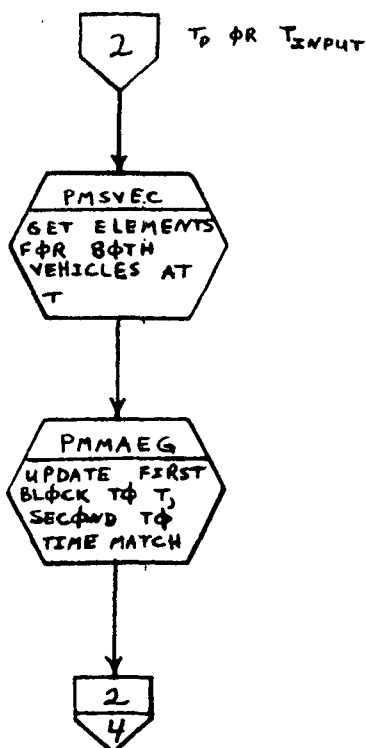
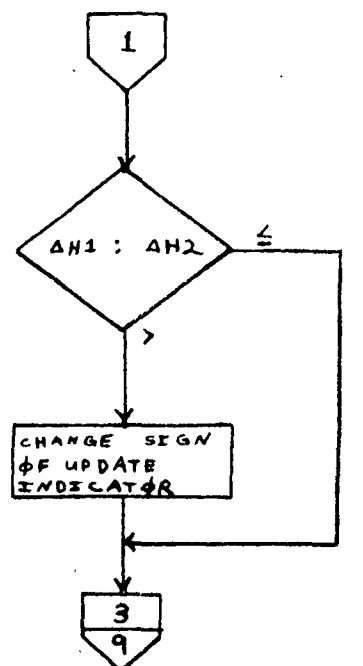


Figure PMMMLC-1. Flowchart

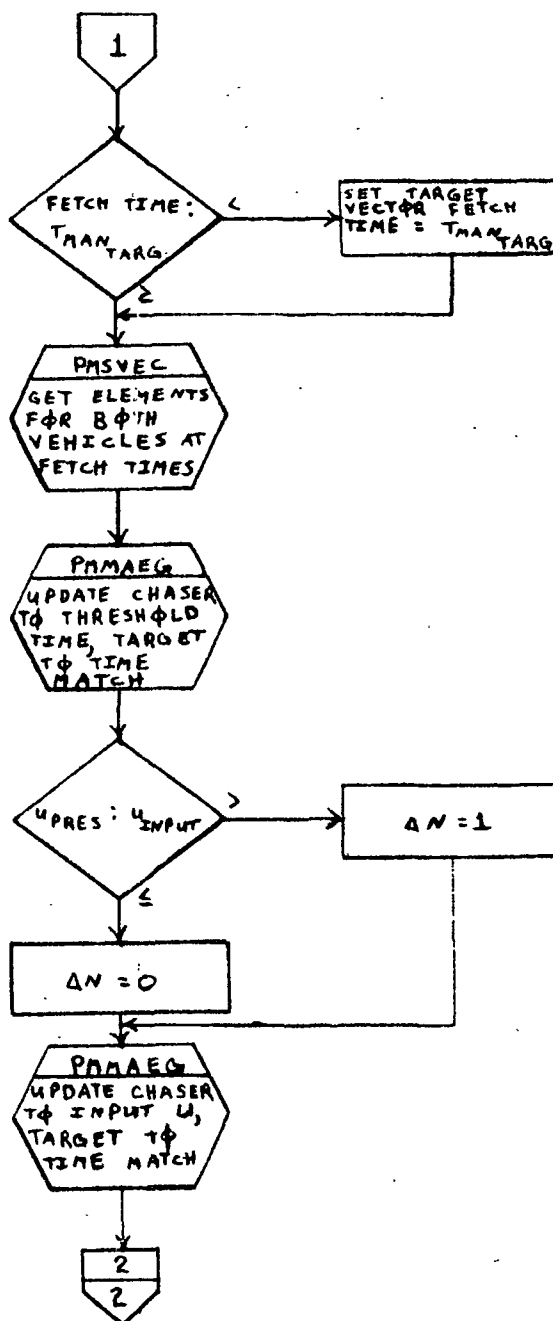


Figure PMMMLC-1. Flowchart

(Sheet 14 of 15)

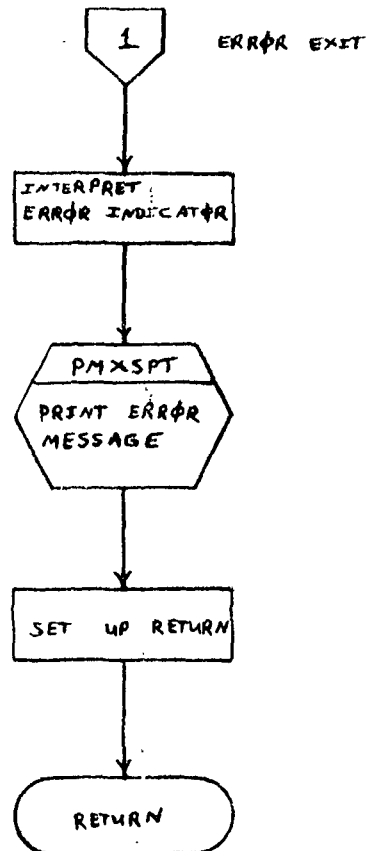


Figure PMMMLC-1. Flowchart

(Sheet 15 of 15)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMPT
DATE 6/22/67 (206)
REV
PAGE 1 of 24

- a. Symbol — PMMMPT
- b. Program Name — 206 Mission Plan Table
- c. Programming Group — Mission Planning
- d. Programmer — W. Stone
- e. Language — Assembler
- f. Data Tables Used — None (directly) - output into table specified in calling sequence
- g. MED's Used — M66, M65
- h. Writeup Completion Date — 5/09/67
- i. System Parameters — Table PMMMPT-1
- j. DSECT - TJUN — Table PMMMPT-2
- k. Flowchart — Figure PMMMPT-1

FUNCTION

For an M66 entry, the finite burn parameters in the input table are placed into the Mission Plan Table. If powered landing guidance is specified, the begin burn time is computed by "ENGINE ON ALGORITHM". For an M65 entry, the finite burn quantities are computed from the input impulsive maneuver elements and are placed in the Mission Plan Table.

USAGE

- a. Calling Sequence — LINK EP = PMMMPT, PARAM = (TJUN)
- b. Inputs — From DSECT; see writeup of input DSECT (TJUN)
- c. Outputs and Exit — Fills a block of the Mission Plan Table and puts it as well as header block of MPT to corefile.
- d. Other Programs Used — GLQATN, GIMIKC, GIMKIC, LLBNTD, PMMAEG, EMMRLM
- e. Data Restrictions and Limits — Maneuver must be executable, trajectory must be non-reentrant.
- f. Error — Error flag set into input "DSECT" (see writeup of input "DSECT").

METHOD

Definition of terms used in describing the method:

$$\begin{bmatrix} \hat{X}_N \\ \hat{Y}_N \\ \hat{Z}_N \end{bmatrix}$$

unit vectors defining the IMU navigation base coordinate axis in the RTCC inertial coordinate system.

Approval	Approval
<i>W. A. Stone</i>	<i>S. E. James</i>

$$\begin{bmatrix} \hat{X}_P \\ \hat{Y}_P \\ \hat{Z}_P \end{bmatrix}$$

unit vectors defining the IMU stable member (platform) coordinate axis in the RTCC inertial coordinate system.

$$\begin{bmatrix} \hat{X}_{PH} \\ \hat{Y}_{PH} \\ \hat{Z}_{PH} \end{bmatrix}$$

unit vectors defining the local vertical local horizontal (LVLH) coordinate system in the RTCC inertial coordinate system

- P = IMU related pitch angle (positive for counterclockwise rotation about Y-axis)
 Y = IMU related yaw angle (positive for clockwise rotation about X-axis)
 R = IMU related roll angle (positive for counterclockwise rotation about Z-axis)
 Y_H = LVLH related yaw angle (positive for counterclockwise rotation about X-axis)
 P_H = LVLH related pitch angle (positive for counterclockwise rotation about Y-axis)
 R_H = LVLH related roll angle (positive for counterclockwise rotation about Z-axis)

$$\begin{bmatrix} \hat{X}_B \\ \hat{Y}_B \\ \hat{Z}_B \end{bmatrix}$$

= unit vector defining the vehicle body axis in the RTCC inertial coordinate system.

$$\begin{bmatrix} V_{GX} \\ V_{GY} \\ V_{GZ} \end{bmatrix}$$

coordinates of the V_G vector in the IMU platform coordinate system for a guided maneuver, V_G is computed by the guidance equations. For an RCS maneuver, V_G represents the total maneuver ΔV in the directions of the calculated thrust at RCS on time.

- $\hat{CPT6}$ = unit normal vector in IMU coordinate system, used by engine on algorithm.
 $\hat{CPT6}_X$ = unit normal vector in RTCC inertial coordinates system, used by engine on algorithm.
 $\hat{RCS}_M$ = Target position vector for ascent guidance in IMU coordinate system.

$\hat{R}_{CSM_{RTCC}}$ = Target position vector for ascent guidance in RTCC inertial coordinate system.
 $\overline{\Delta V}$ = Inertial coordinates of finite burn velocity increment to be applied in order to duplicate the results of the given impulsive maneuver
 ΔV Magnitude of above.
 R_D Radius to the target position, if set negative ignore radius computation.

The following notation will be used to denote coordinate system rotations.

$$R_X(\pm \theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \pm \sin \theta \\ 0 & \mp \sin \theta & \cos \theta \end{bmatrix}$$

$$R_Y(\pm \theta) = \begin{bmatrix} \cos \theta & 0 & \mp \sin \theta \\ 0 & 1 & 0 \\ \pm \sin \theta & 0 & \cos \theta \end{bmatrix}$$

$$R_Z(\pm \theta) = \begin{bmatrix} \cos \theta & \pm \sin \theta & 0 \\ \mp \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The program is designed to handle two basic types of inputs; the following is an indication of mathematical equations necessary for each type of input.

- a. Case where finite burn parameters are input; i.e., M66. If $Y_{H'}$, $P_{H'}$, $R_{H'}$ are given to specify the vehicle orientation at maneuver begin time, the IMU related P , Y , R must be calculated. The LVLH attitude sequence is $Y_{H'}$, $P_{H'}$, $R_{H'}$ and they reference the vehicle body orientation with respect to the local horizontal local vertical at maneuver begin time. The following expression is used to determine the body axis coordinate system.

$$\begin{bmatrix} \hat{X}_B \\ \hat{Y}_B \\ \hat{Z}_B \end{bmatrix} = R_X(R_{H'}) * R_Y(P_{H'}) * R_Z(Y_{H'}) \begin{bmatrix} \hat{X}_{PH} \\ \hat{Y}_{PH} \\ \hat{Z}_{PH} \end{bmatrix}$$

The IMU attitude sequence is P, R, Y. It follows that

$$\begin{bmatrix} \hat{X}_B \\ \hat{Y}_B \\ \hat{Z}_B \end{bmatrix} = R_X(-Y) * R_Z(R) * R_Y(P) \begin{bmatrix} \hat{X}_P \\ \hat{Y}_P \\ \hat{Z}_P \end{bmatrix}$$

Solving this matrix equation for P, Y, R gives

$$P = \tan^{-1} \left[\frac{\hat{X}_P \cdot \hat{Z}_B}{\hat{Z}_P \cdot \hat{Z}_B} \right] \quad 0 \leq P \leq 2\pi$$

$$Y = \tan^{-1} \left[\frac{\hat{Y}_P \cdot \hat{X}_B}{\hat{Y}_P \cdot \hat{Y}_B} \right] \quad 0 \leq Y < 2\pi$$

$$R = \sin^{-1} \left[-\hat{Y}_P \cdot \hat{Z}_B \right] \quad \begin{matrix} 0 \leq R \leq \frac{\pi}{2} \\ \frac{3\pi}{2} \leq R < 2\pi \end{matrix}$$

The preceding expressions for P, Y are undefined for $R = \frac{\pi}{2}$ or $\frac{3\pi}{2}$.
The following equations should be used in such cases.

$$R = 0$$

$$P = \tan^{-1} \left[\frac{\hat{X}_P \cdot \hat{Z}_B}{\hat{Z}_P \cdot \hat{Z}_B} \right] \quad 0 \leq P < 2\pi$$

If an RCS thruster is specified

$$\overline{\Delta V} = V_T * \hat{X}_B$$

where V_T = magnitude of ΔV if ΔV option specified

and V_T = 1. if ΔT option specified

Then if (RCS - X) is the thruster set $\overline{\Delta V} = -\overline{\Delta V}$

$$\begin{aligned} V_{GX} &= \overline{\Delta V} \cdot \hat{X}_P \\ V_{GY} &= \overline{\Delta V} \cdot \hat{Y}_P \\ V_{GZ} &= \overline{\Delta V} \cdot \hat{Z}_P \end{aligned}$$

Most of the first case is where a guided maneuver is directly input and must be broken down into sub cases depending upon thruster.

Consider first the case of an APS thruster, which means ascent guidance is to be used.

$$\begin{bmatrix} \hat{R}_{CSM} \\ RTCC \end{bmatrix} = \begin{bmatrix} X_P & Y_P & Z_P \end{bmatrix} \begin{bmatrix} \hat{R}_{CSM} \\ IMU \end{bmatrix} \quad \text{Note: } \hat{R}_{CSM} \text{ IMU is the input}$$

The above target vector is put directly into the MPT for use by the integrator.

Next consider the case of the DPS thruster, which is actually broken down into two cases. One is with powered landing guidance. Here the input CPT_6 vector is converted from the IMU coordinate system to the RTCC inertial coordinate system by the following

$$\begin{bmatrix} CPT_6_X \\ RTCC \end{bmatrix} = \begin{bmatrix} \hat{X}_P & \hat{Y}_P & \hat{Z}_P \end{bmatrix} \begin{bmatrix} CPT_6 \\ IMU \end{bmatrix}$$

This vector, together with the time to begin search, is input into the engine on algorithm, which then returns a unit thrust vector and the finite begin burn time. The CPT_6_X vector, unit thrust vector and time of finite burn are put into the Mission Plan Table for use by the integrator.

The other with the DPS thruster is the descent injection guidance. This case actually is an adjustment in the orbit radius 180 degrees away from maneuver burnout and the only inputs required are desired radius of the orbit and other directly input quantities.

- b. Case where Keplerian orbital elements before and after an impulsive maneuver are input (M65).

The impulsive elements before and after the maneuver at the impulsive maneuver time. The basic function of this program is to calculate finite begin burn time and the target parameters that produce an equivalent maneuver. (Note: all impulsive transfers for this mission will be guided maneuvers).

A Keplerian to inertial conversion routine is used to obtain

$\bar{V}_B$, $\bar{V}_A$ where:

$\bar{V}_B$ = inertial velocity vector before the impulsive maneuver

$\bar{V}_A$ = inertial velocity vector after the impulsive maneuver.

$\Delta \bar{V}$ = $\bar{V}_A - \bar{V}_B$

ΔV = $|\Delta \bar{V}|$

In calculating the finite time to begin burn for this mission the important assumption that was made was to put $1/2 \Delta V$ on each side of the impulsive maneuver point.

Please refer to the flowchart pages 9 and 10 for the resulting equations that are used to compute GMTBB (time of begin finite burn). One big exception to be noted is the case where the attitude mode of maneuver is to be powered landing guidance. For this case the above GMTBB is used as time to begin search for the engine or algorithm and the actual GMTBB is computed by the engine on algorithm.

After the finite begin burn time is computed, the program takes separate directions depending upon thruster.

For the APS thruster, a target vector some distance ahead of present position must be computed. This is computed by updating the impulsive elements after the maneuver through an angle θ_c (Currently 140 degrees) using two-body motion. The resulting vector is $\hat{RCSM}_{RTCC}$, which is placed into the mission plan table for use by the integrator.

NOTE: $\hat{RCSM}_{RTCC}$ is also converted to $\hat{RCSM}_{IMU}$ for display by the following

$$\begin{bmatrix} \hat{R}_{CSM} \\ IMU \end{bmatrix} = \begin{bmatrix} \hat{X}_P \\ \hat{Y}_P \\ \hat{Z}_P \end{bmatrix} \begin{bmatrix} \hat{R}_{CSM} \\ RTCC \end{bmatrix}.$$

For the DPS thruster, two possible attitude modes must be considered. The first is the powered landing guidance mode. Here a unit normal vector $CPT6_x$ must be computed as follows.

$$\begin{bmatrix} CPT6_x \end{bmatrix} = \begin{bmatrix} \sin i_a & \sin h_a \\ -\sin i_a & \cos h_a \\ \cos i_a \end{bmatrix} \quad \text{where subscript "a" means elements}$$

after the maneuver.

This vector ($CPT6_x$) is necessary as input to the engine on algorithm and therefore must be computed prior to calling such to get finite begin burn time. Also the $CPT6_x$ vector is converted to the IMU coordinate system for display purposes only. The conversion is as follows

$$\begin{bmatrix} CPT6 \\ IMU \end{bmatrix} = \begin{bmatrix} \hat{X}_P \\ \hat{Y}_P \\ \hat{Z}_P \end{bmatrix} \begin{bmatrix} CPT6 \\ RTCC \end{bmatrix}$$

Both the $CPT6_{IMU}$ and $CPT6_{RTCC}$ are stored into the Mission Plan Table.

The second attitude mode for the DPS thrusted is the descent orbit injection guidance. The only input that is required in this case is the target radius 180 degrees ahead of the current impulsive time. Since this maneuver is used to shift the line of apsides of the current orbit and thus define a new apogee and perigee. R_D is computed as follows.

$$R_D = a_a (1 + e_a) \text{ if apogee is desired.}$$

$$R_D = a_a (1 - e_a) \text{ if perigee is desired.}$$

NOTE: The radius 180 degrees ahead is wanted; therefore, the test on apogee or perigee is opposite from what it would be if current position was being used.



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMPT

DATE 6/22/67 (206)

REV

PAGE 8 of 24

Table PMMMPT-1. System Parameters

Symbol in Program	Symbol In Master Tape	Type *	Units	Comments
SRMU	MCERMU	D	$\sqrt{E r^3 / 2 M^2 H}$	Square root MU
PI2	MCETPI	D	none	2π
GRAV	MCEACG	D	$E r / \text{hr}^2$	Acceleration of gravity
UOAPS	MCTANP	D	hr.	Ullage overlap APS
DTTAPS	MCTATF	D	hr.	Time of tailoff APS
IAPS	MCTASI	D	hr.	Specific impulse APS
UODPS	MCTDUP	D	hr.	Ullage overlap DPS
DTTDPS	MCTDIF	D	hr.	Time of tailoff DPS
IDPS	MCTDSI	D	hr.	Specific impulse DPS full
IDPST	MCTDIT	D	hr.	Specific impulse DPS 10 percent
THRD1	MCTTDT	D	lbs.	Thrust DPS 10 percent
THRD2	MCTTDP	D	lbs.	Thrust DPS full
UDDPS1	MCTUDT	D	lbs/hr.	Weight loss rate DPS 10 percent
UDDPS2	MCTUDP	D	lbs/hr	Weight loss rate DPS full

*Type
 H - Fixed-point half word
 F - Fixed-point full word
 E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMMPT
 DATE 6/22/67 (206)
 REV
 PAGE 9 of 24

Table PMMMPT-1. System Parameters (Cont'd.)

Symbol in Program	Symbol In Master Tape	Type *	Units	Comments
DELTAI	MCTTFT	D	hr.	Time of DPS 10 percent
UDRP4	MCTPX4	D	lbs/hr.	Weight loss rate RCS +X(4)
UDRP2	MCTPX2	D	lbs/hr.	Weight loss rate RCS +X(2)
TRCS4	MCTP4X	D	lbs.	Thrust level RCS +X(4)
TRCS2	MCTP2X	D	lbs.	Thrust level RCS +X(2)
UDAPS	MCTUAP	D	lbs/hr.	Weight loss rate APS thrust
THETAC	MCVTHC	D	none	θ_c (angle of advance for APS guidance)
PI	MCMPPIR	D	none	π

*Type
 H - Fixed-point half word
 F - Fixed-point full word
 E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMPT
DATE 6/22/67 (20
REV
PAGE 10 of 24

Table PMMMPT-2. DSECT - TJUN (Inputs)

Byte No.	Type*	Units	Value / Description / Comments
1-960			First block of mission plan table See writeup of such
961-964	F		Vehicle code
965-968	F		Maneuver sequence
969-976	D		GMTDI
977-980	F		Thruster code
981-984	F		Attitude code
985-992	D		Delta velocity of maneuver
993-1000	D		Delta time of burn
1001-1008	D		Time of ullage
1009-1012	F		Coordinant Indicator
1013-1016	F		Spare
1017-1024	D		Pitch
1025-1032	D		Yaw
1033-1040	D		Roll
1041-1044	F		Separation Indicator
1045-1048	F		Ullage thruster
1049-1056	D		Target input
1057-1064	D		Target input
1065-1072	D		Target input
1073-1080	D		Target input
1081-1088	D		Target input
1089-1092	F		Error flag (output)
1093-1096	F		Output block number
1097-1100	F		Direct input flag
1101-1104	F		Spare

*Type

D — Double-precision (floating-point)

F — Fixed-point full word

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMMPT
 DATE 6/22/67 (20
 REV
 PAGE 11 of 24

Table PMMMPT-2. DSECT - TJUN (Inputs) (Cont'd.)

Byte No.	Type*	Units	Value / Description / Comments
1105-1112	D		Time of vector
1113-1120	D		R_X
1121-1128	D		R_Y
1129-1136	D		R_Z
1137-1144	D		V_X
1145-1152	D		V_Y
1153-1160	D		V_Z
1161-1168	D		Area
1169-1176	D		Weight
1177-1184	D		Start limit
1185-1192	D		End limit
1193-1200	D		A before
1201-1208	D		E before
1209-1216	D		I before
1217-1224	D		L before
1225-1232	D		G before
1233-1240	D		H before
1241-1248	D		A after
1249-1256	D		E after
1247-1264	D		I after
1265-1272	D		L after
1273-1280	D		G after
1281-1288	D		H after
1289-1296	D		Time impulsive
1299-1304	D		F before
1305-1312	D		F after

*Type

D — Double-precision (floating-point)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMPT

DATE 6/22/67 (2

REV

PAGE 12 of 24

Table PMMMPT-2. DSECT - TJUN (Inputs) (Cont d.)

Byte No.	Type*	Units	Value / Description / Comments
1313-1320	D		Spare
1321-1328	D		Spare
1329-1336	D		Spare
1337-1344	D		Spare
1345-1346	H		Number of maneuvers in table
1347-1348	H		Maneuver purpose
1349-1352	F		Spare
1353-1356	F		Location of EBCDIC table name
1347-1360	F		Spare
1361-1368	D		Spare
1369-1376	D		Undocking area
1377-1384	D		Undocking weight
1385-1520	17D		Spare

*Type

D — Double-precision (floating-point)

F — Fixed-point full word

H — Fixed-point half word

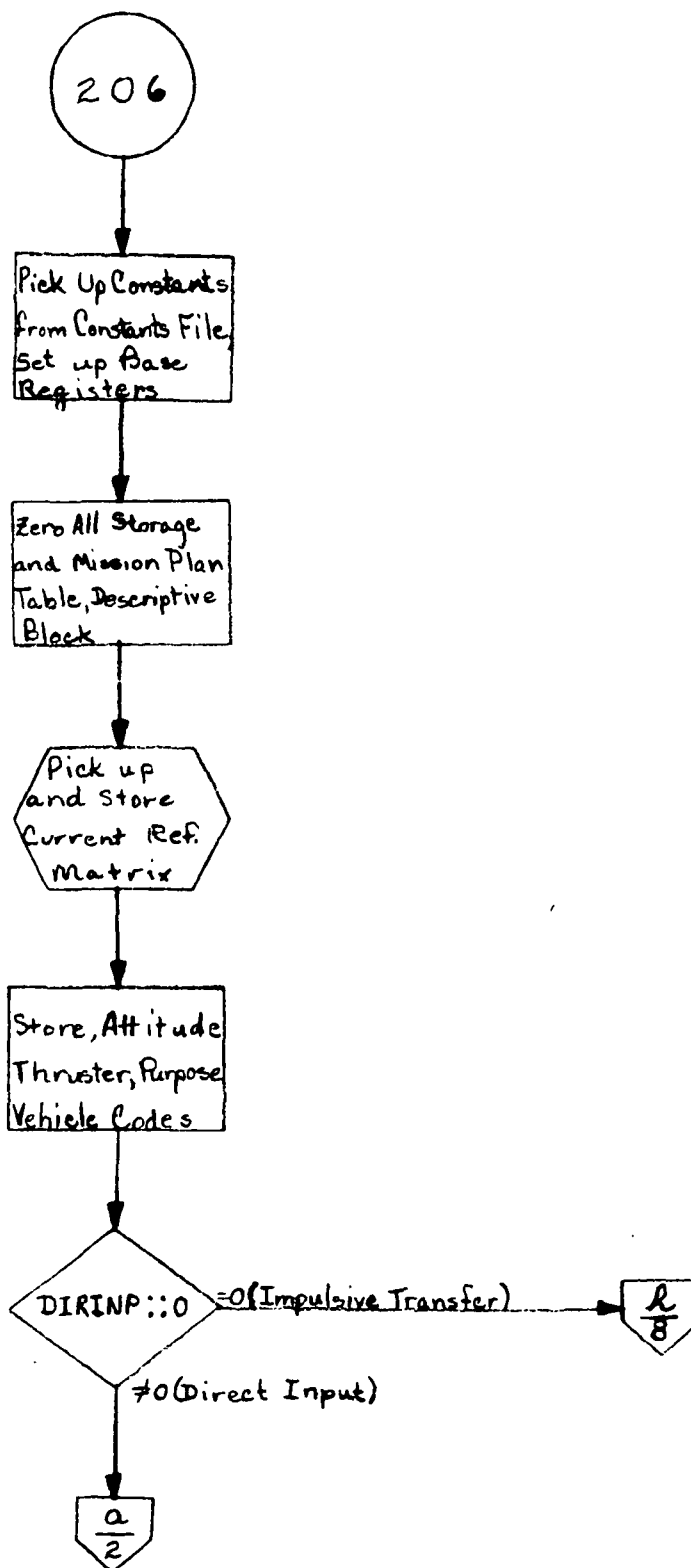


Figure PMMMPT-1. Flowchart

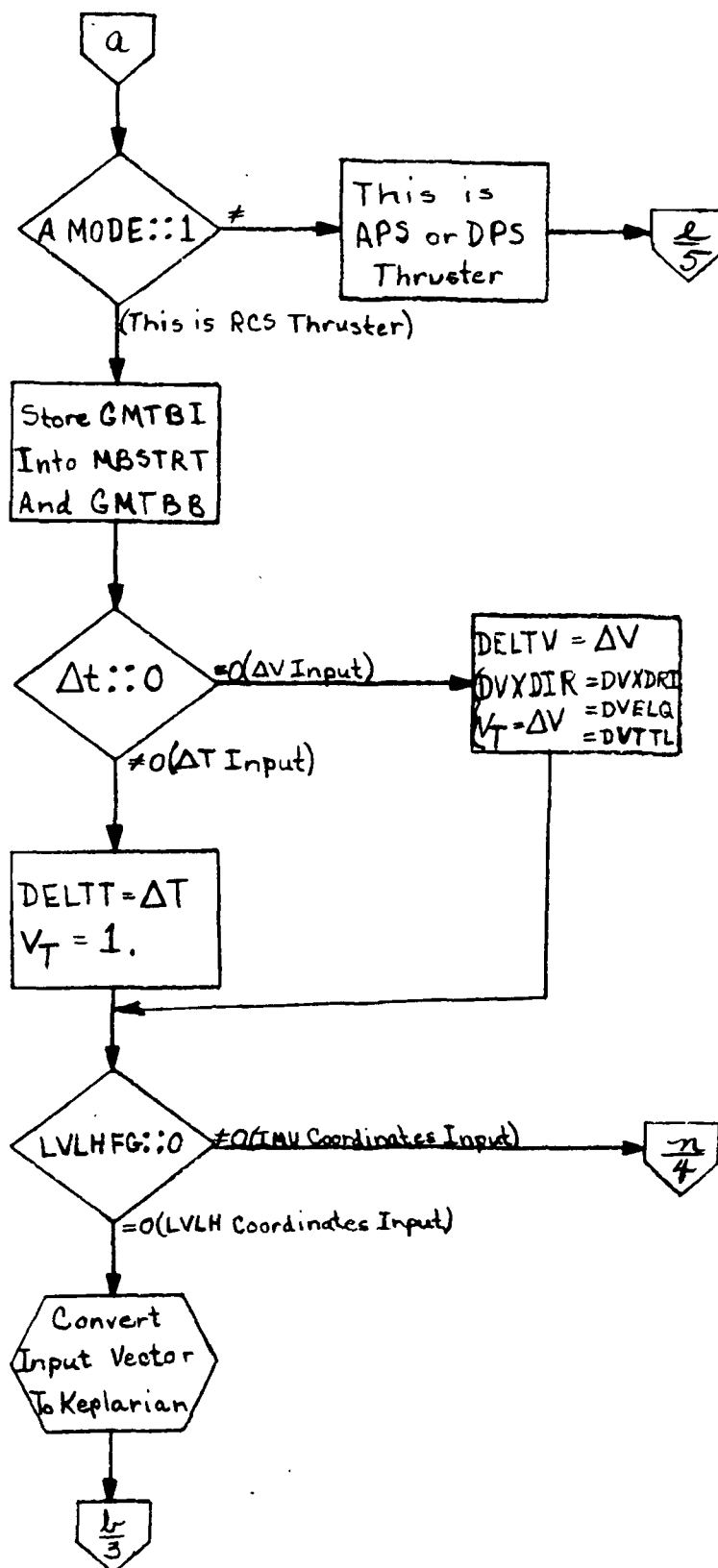


Figure PMMMPT-1. Flowchart

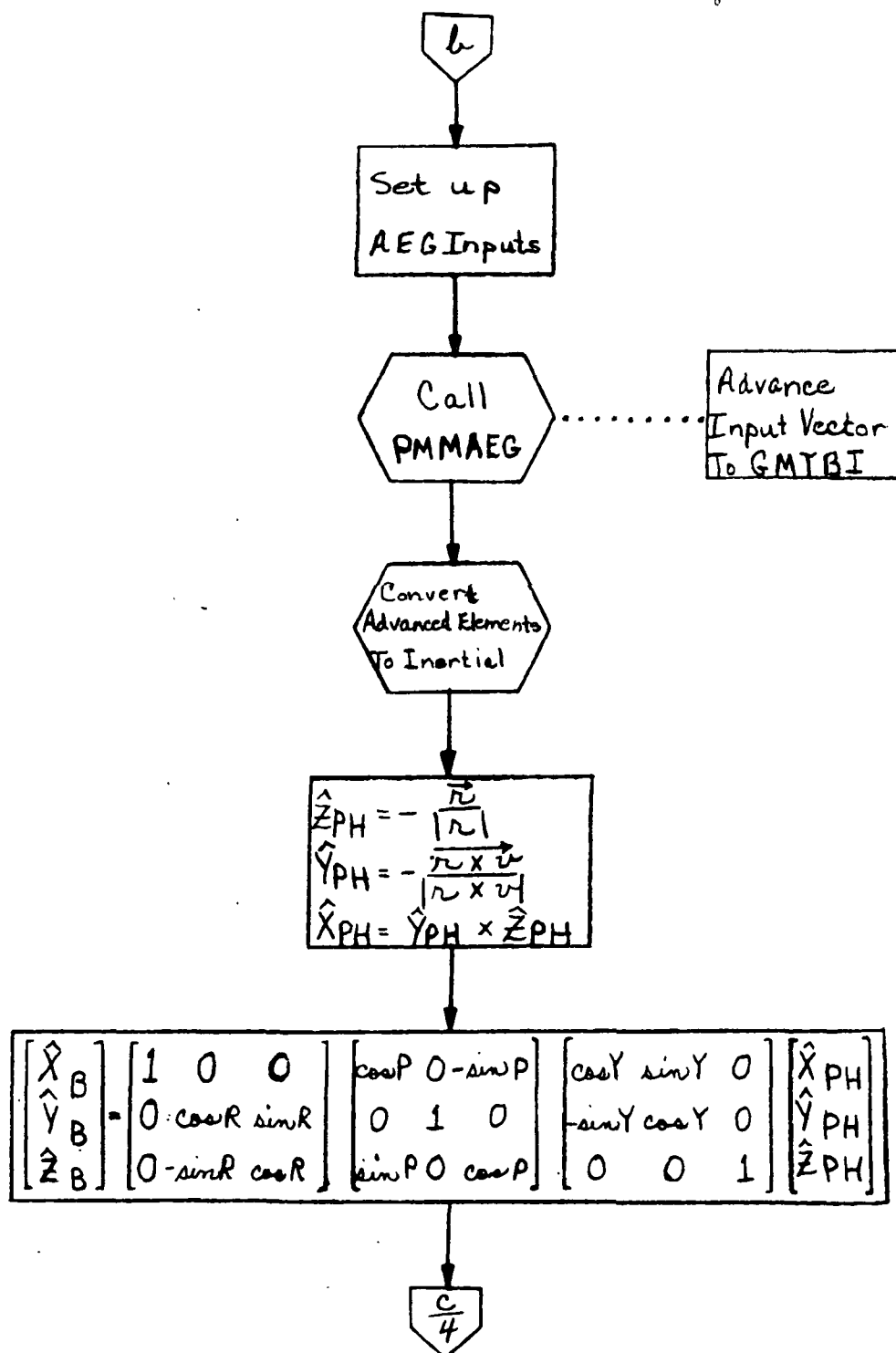


Figure PMMMPT-1. Flowchart

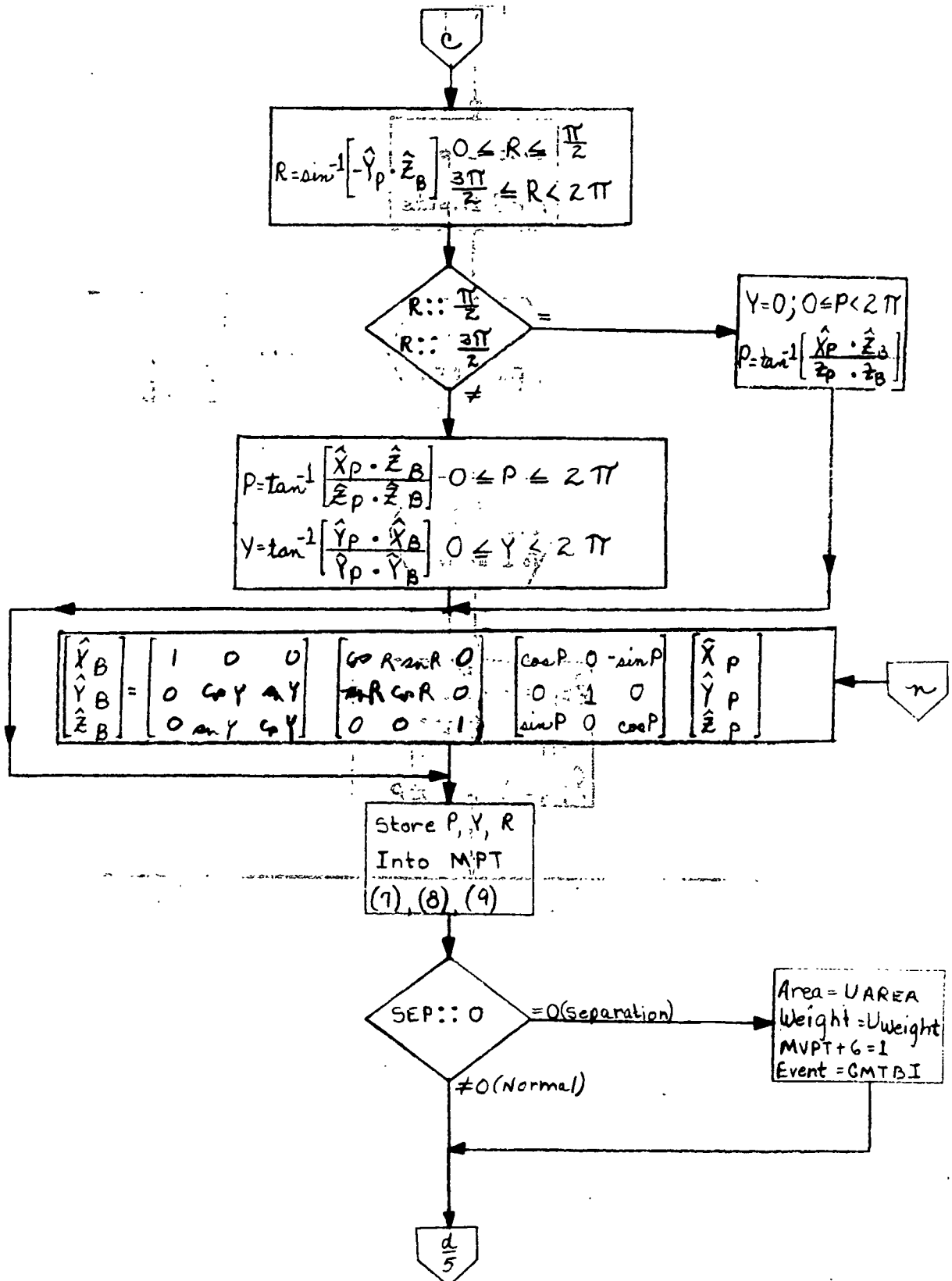


Figure PMMMPT-1. Flowchart

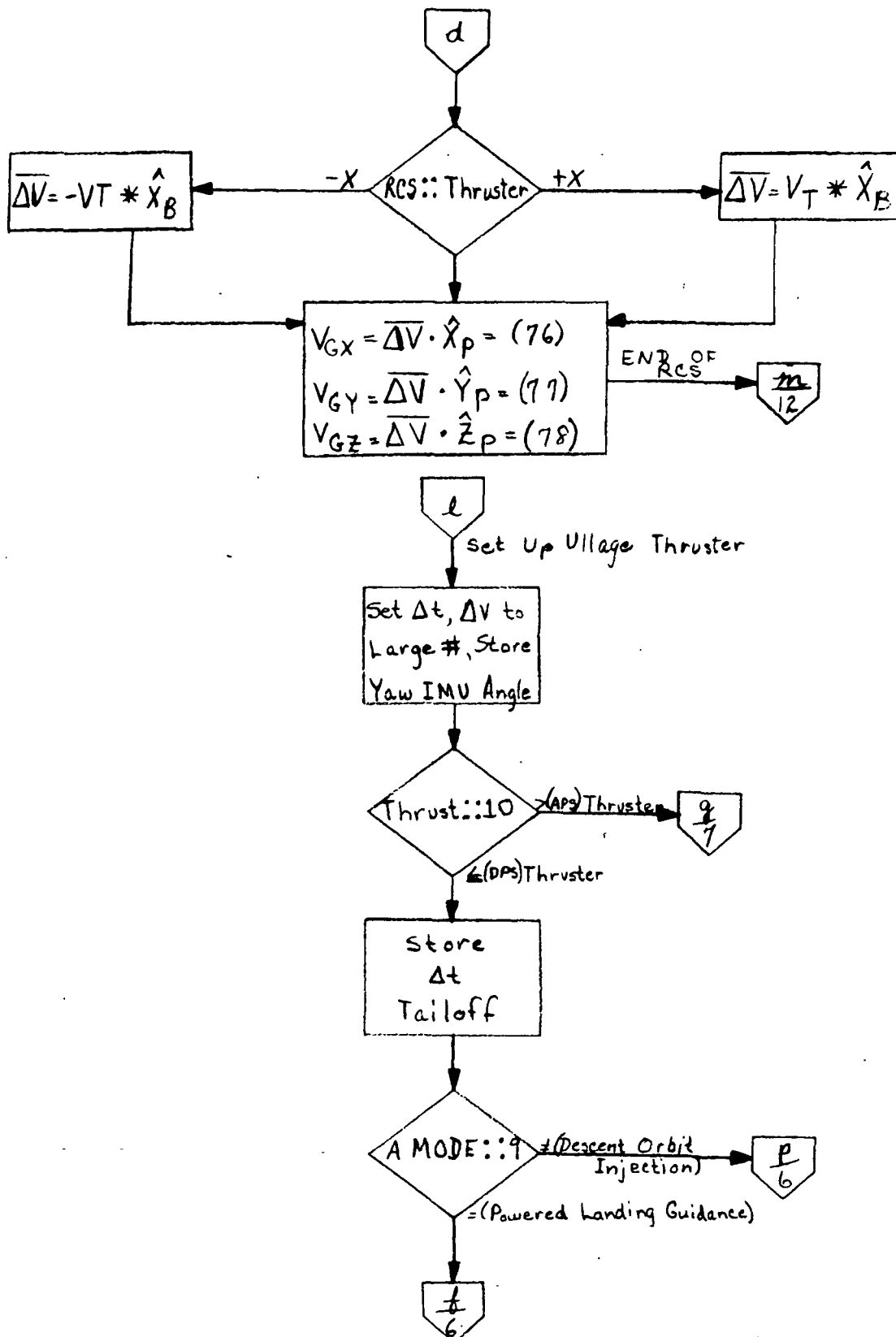


Figure PMMMPT-1. Flowchart

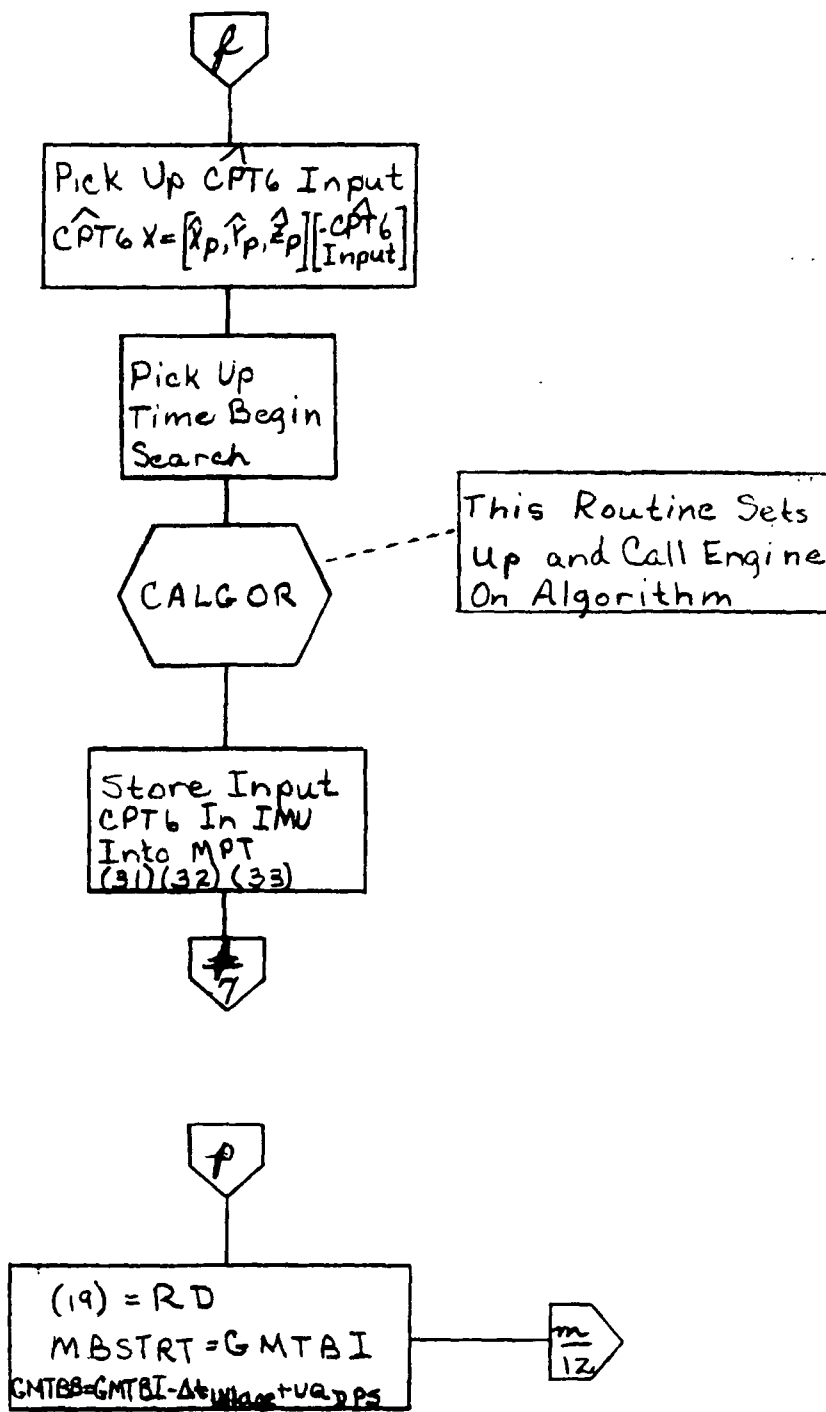


Figure PMMMPT-1. Flowchart

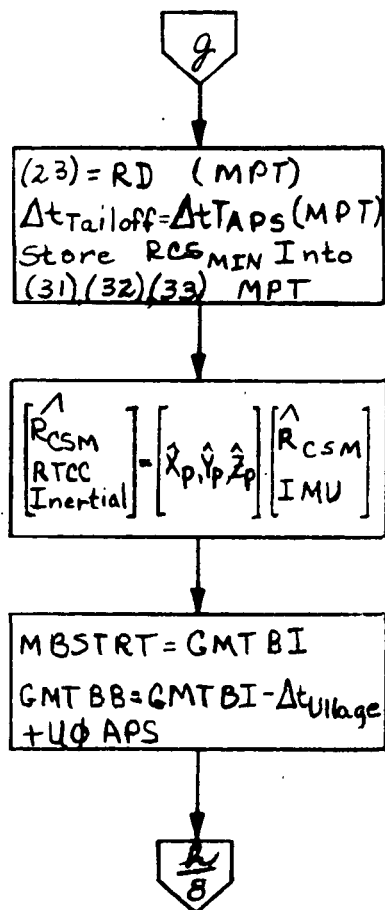


Figure PMMMPT-1. Flowchart

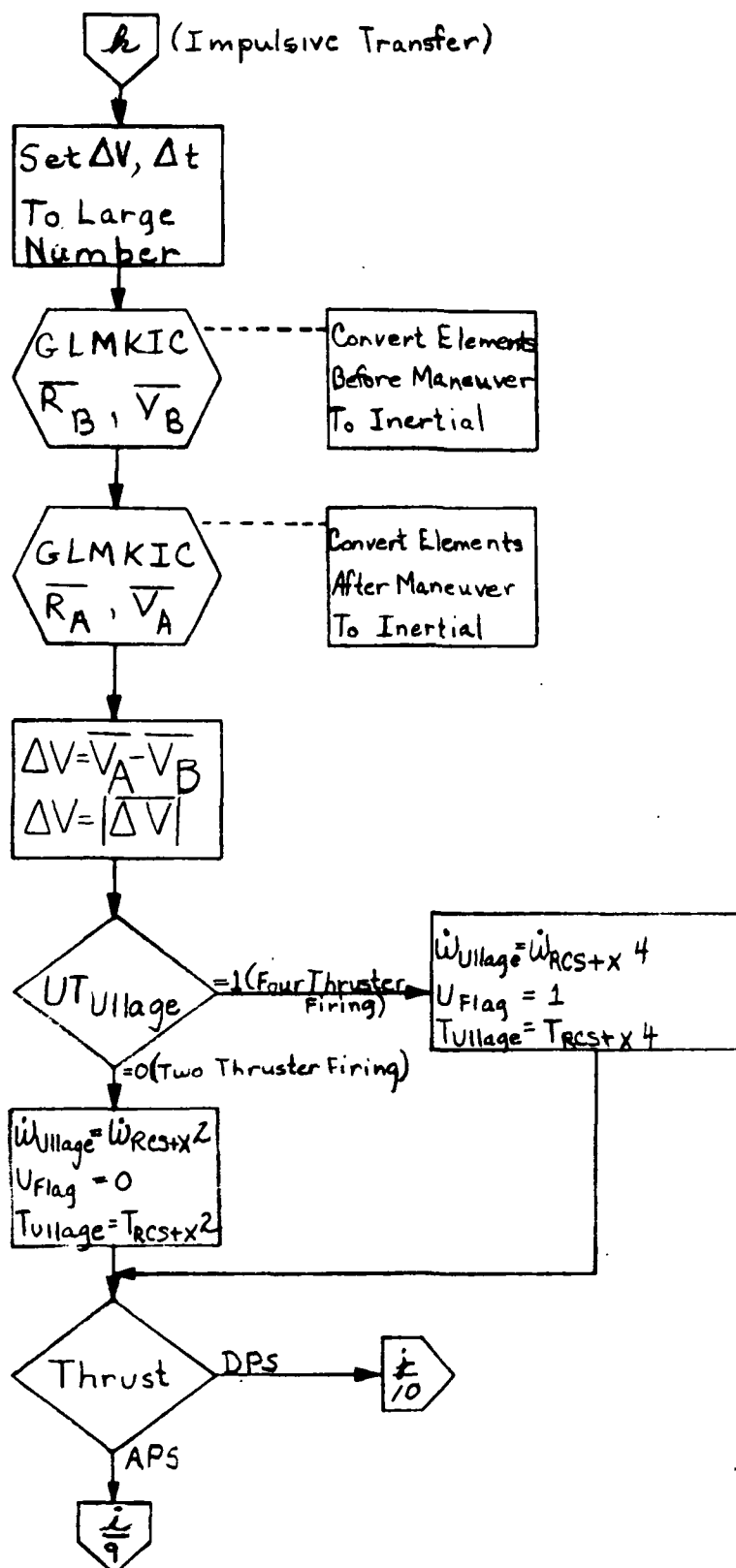


Figure PMMMPT-1. Flowchart

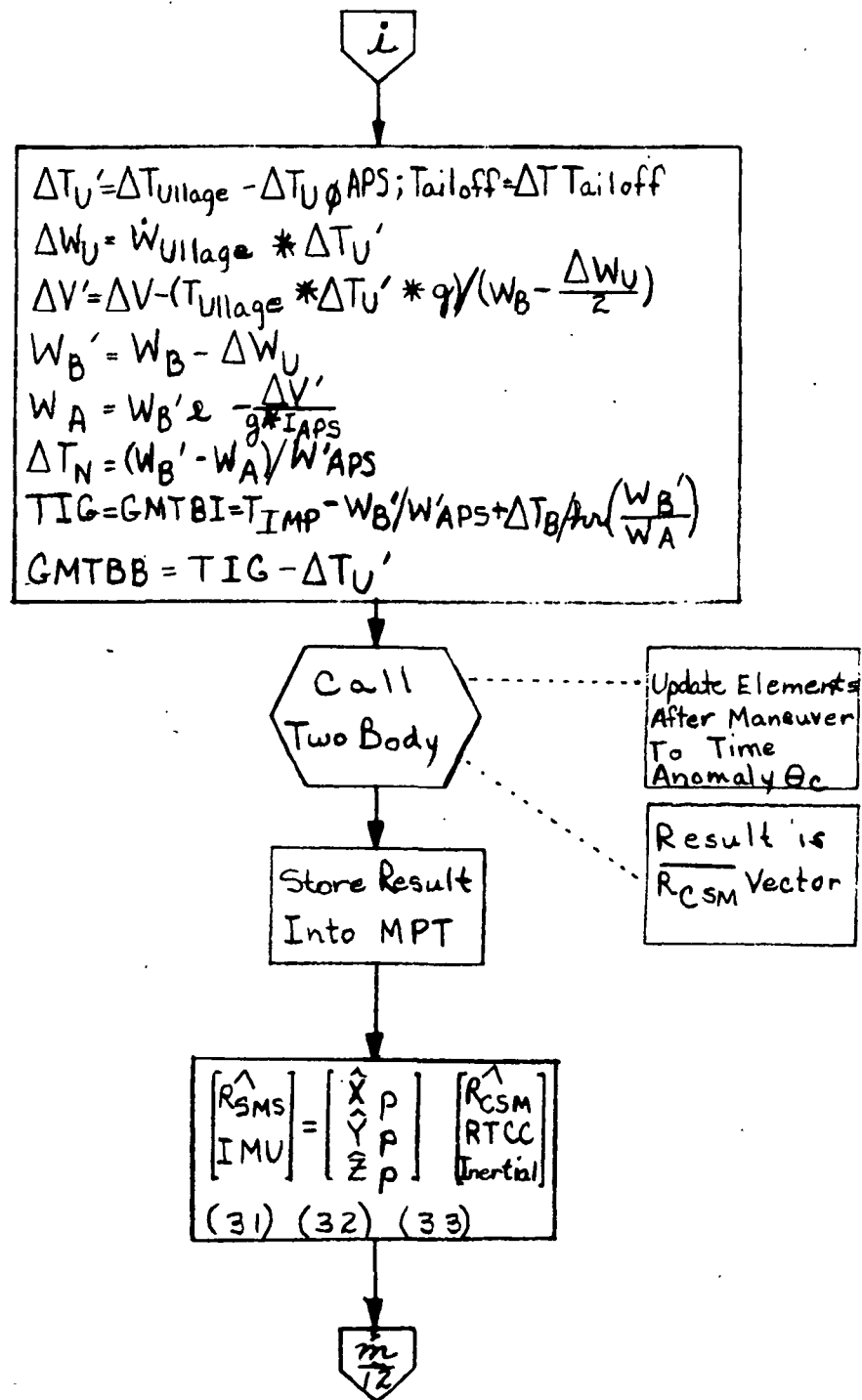


Figure PMMMPT-1. Flowchart

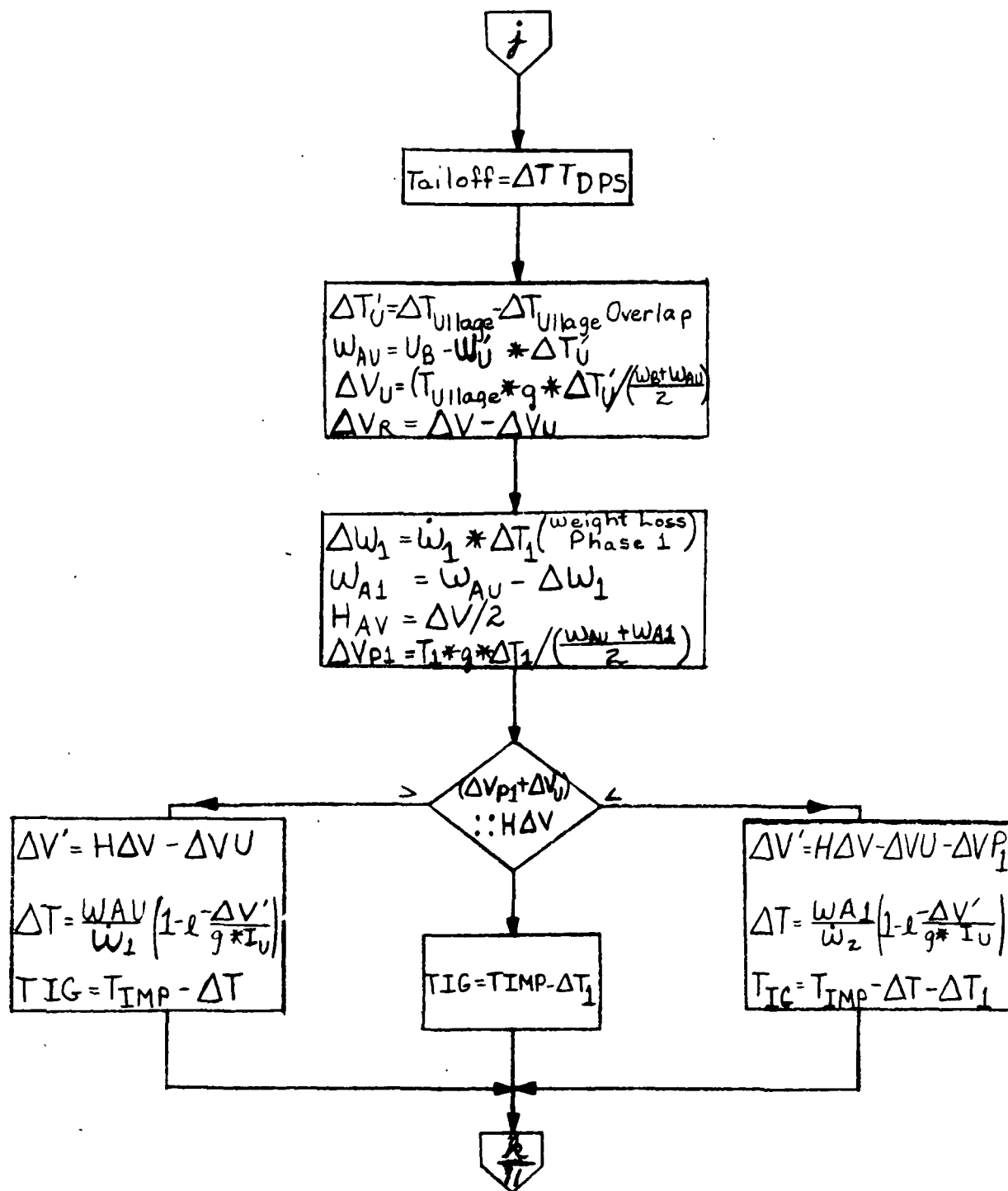


Figure PMMMPT-1. Flowchart

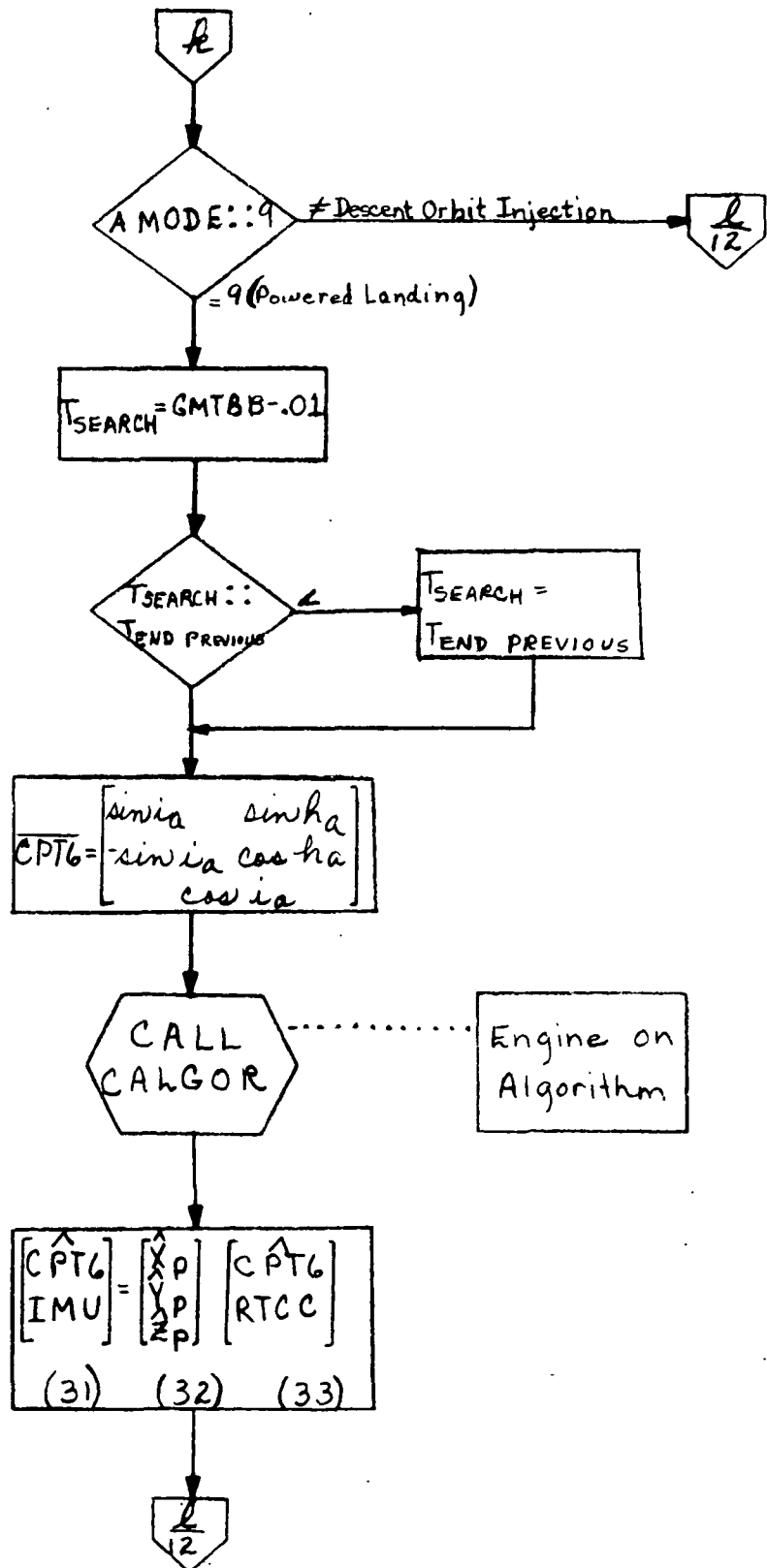


Figure PMMMPT-1. Flowchart

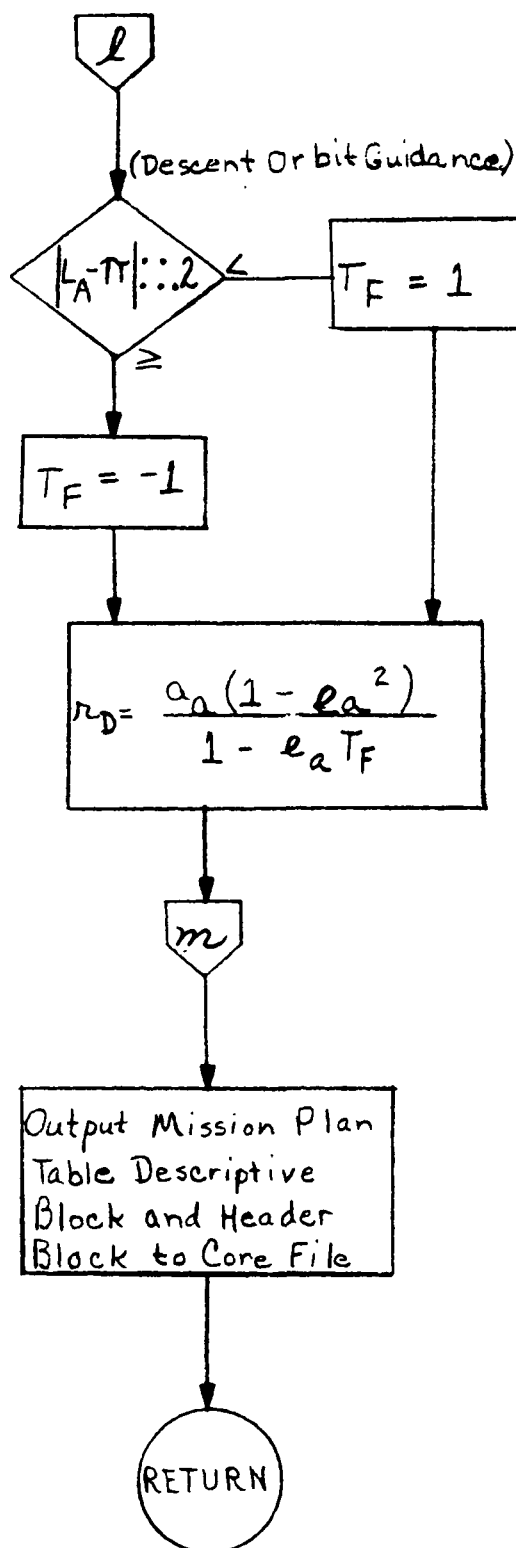


Figure PMMMPT-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMPT
DATE 10/3/67 (EORP)
REV
PAGE 1 of 22

- a. Symbol — PMMMPT
- b. Program Name — Maneuver Transfer Processor
- c. Programming Group — Maneuver Control (Mission Planning)
- d. Programmer(s) — W. Stone
- e. Language — Assembler
- f. Data Tables Used — None (directly). Output is put into table by PMMXFR
- g. MED's Used — M65, M70, M72
- h. Writeup Completion Date — 9/1/67
- i. System Parameters — Table PMMMPT-1
- j. Parameter List — Table PMMMPT-2
- k. Flowchart — Figure PMMMPT-1

FUNCTION

PMMMPT sets up inputs for use by powered flight integrator. For all entries, the finite burn quantities are computed from the input impulse maneuver elements and are placed in an output array for input to Mission Plan Table. Inputs to this program are of the same format regardless of MED used.

USAGE

- a. Calling Sequence — LINK EP=PMMMPT
NOTE: REG 10 points to output array; REG 11 points to input array.
- b. Inputs — See input parameter list, Table PMMMPT-2.
- c. Outputs and Exit — A block of Mission Plan Table is computed and is outputted to area pointed to by REG 10.
- d. Other Programs Used — GIMGBL, GIMIKC, GIMKIC, GLQATN, PMMAEG, and FORTRAN library routines.
- e. Data Restrictions and Limits — Maneuver must not be overlapping another maneuver, not during reentry, and must be at or ahead of present time.
- f. Error — Error flag is set into REG 15 upon return. Maneuver time error = 52; AEG events error = 51

METHOD

The input orbital elements before and after the maneuver are at the impulsive maneuver time. The basic function of this program is to compute: (1) the begin burn time of the maneuver, (2) the initial body axis in RTCC inertial coordinate system, (3) unit thrust vector in RTCC inertial coordinate

Approval	Approval
<i>W. Stone</i>	<i>S. E. James</i>

system, if not a guided maneuver, (4) RTCC inertial delta velocity vector, if a guided maneuver.

A Keplerian to inertial conversion routine is used to obtain $\bar{V}_B$, $\bar{V}_A$ where:

$\bar{V}_B$ = inertial velocity vector before the impulsive maneuver at impulsive time

$\bar{V}_A$ = inertial velocity vector after the impulsive maneuver at impulsive time

$\bar{\Delta V} = \bar{V}_A - \bar{V}_B$ (this is RTCC inertial delta velocity vector if guided maneuver)

$\Delta V = |\bar{\Delta V}|$ magnitude of above

$\bar{U}_T = \frac{\bar{\Delta V}}{\Delta V}$ (unit thrust vector in RTCC inertial coordinate system for non-guided maneuver)

The body axis is computed as follows: take the impulsive elements prior to the maneuver backup free flight to the begin burn time. Convert the elements at the begin burn to the RTCC inertial coordinate system $\bar{r}_B$, $\bar{V}_B$. Then define the thrust coordinate system as:

$$\hat{X}_T = \bar{\Delta V} / \Delta V$$

$$\hat{Y}_T = \frac{\bar{r}_B \times \hat{X}_T}{|\bar{r}_B \times \hat{X}_T|} \quad \text{Note: If } \bar{\Delta V} \text{ along } \bar{r}_B, \text{ then } \hat{Y}_T = \frac{\bar{r}_B \times \bar{V}_B}{|\bar{r}_B \times \bar{V}_B|}$$

$$\hat{Z}_T = \hat{X}_T \times \hat{Y}_T$$

For all RCS + X maneuvers for both the CSM and LM vehicles, the body axes are as follows: $\hat{X}_B = \hat{X}_T$, $\hat{Y}_B = \hat{Y}_T$, $\hat{Z}_B = \hat{Z}_T$. For all RCS - X maneuvers for both the CSM and LM vehicles, the body axes are as follows: $\hat{X}_B = -\hat{X}_T$, $\hat{Y}_B = -\hat{Y}_T$, $\hat{Z}_B = +\hat{Z}_T$.

For all other cases, the thrust axes are rotated about the Z-axis through the (-Yaw Gimbal) angle and then rotated about the Y-axis through the (-Pitch Gimbal) angle. Note: If thruster is APS or DPS, Yaw Gimbal is replaced by Roll Gimbal angle, but the computation is unaffected.

$$\begin{bmatrix} \hat{X}_B \\ \hat{Y}_B \\ \hat{Z}_B \end{bmatrix} = \begin{bmatrix} \cos PG & 0 & \sin PG \\ 0 & 1 & 0 \\ -\sin PG & 0 & \cos PG \end{bmatrix} \begin{bmatrix} \cos YG_{(RG)} & -\sin YG_{(AG)} & 0 \\ \sin YG_{(AG)} & \cos YG_{(RG)} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{X}_T \\ \hat{Y}_T \\ \hat{Z}_T \end{bmatrix}$$

The remaining quantity to be computed is time-to-begin burn. This then is computed such that the centroid time of burn is the impulsive time of the maneuver. Since by definition, the thrust acceleration profile has zero first moments about its centroid time, the following equation must hold true:

$$\sum_{i=1}^n \left[T_i g (t_{bi} - t_c) \int_{t_{bi}}^{t_{ei}} \frac{dt}{w_{oi} - \dot{w}_i (t - t_{bi})} + T_i g \int_{t_{bi}}^{t_{ei}} \frac{(t - t_{bi}) dt}{w_{oi} - \dot{w}_i (t - t_{bi})} \right] \quad (1)$$

where: T_i = Thrust level during i^{th} phase of burn (assume to be constant)

g = Acceleration of gravity

t_{bi} = Time-to-begin burn i^{th} phase

t_{ei} = Time-to-end burn i^{th} phase

w_{oi} = Weight at start of i^{th} phase

$\dot{w}_i$ = Weight flow rate of i^{th} phase (assume to be constant)

The above equation is true in general for any "n" phase meaning, however, the time of burn for each phase must be known. In order to determine this, we must consider three separate cases.

Case 1. Only an RCS thruster will be fired. Pickup the thrust level and weight loss rate for the specified thruster. The specific impulse

is defined as $I_{SP} = T / \dot{w}$. Now delta time of burn is

$$\Delta t_M = \frac{w}{\dot{w}} \left(1 - e^{-\frac{\Delta V}{g * I_{SP}}} \right)$$

and weight after maneuver is $w_a = w - \dot{w} * \Delta t_M$

For this case "n" in equation (1) is equal to 1 upon integration. Equation (1), the following result is obtained:

$$T_B = GMTBB = T_{IMP} - \frac{w}{\dot{w}} + \frac{t_e - t_B}{\ln\left(\frac{w}{w_a}\right)}$$

Note: $t_e - t_B = \Delta t_M$ (time of burn)

Case 2. The thruster is either LM APS or CSM SPS. The method for computing begin burn time is as follows:

Pickup the thrust level and weight loss rate for the ullage thruster. Then, $\Delta t_U' = \Delta t_U - \Delta t_U$ (overlap)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMPT
DATE 10/3/67 (EORP)
REV
PAGE 5 of 22

where: Δt_U = time of ullage
 Δt_U (overlap) = time of ullage burn overlap main burn
 $\Delta t'_U$ = result time of ullage

$\dot{w}_U$ = weight loss rate ullage thruster

w'_B = weight after ullage

T_U = thrust level ullage thruster

$\Delta V' = \Delta V$ required after ullage

$\dot{w}_s$ = weight loss rate main thruster

T_s = thrust level main thruster

w_a = weight after maneuver.

$$\Delta w_U = \dot{w}_U * \Delta t'_U$$

$$w'_B = w - \Delta w_U$$

$$\Delta V' = \Delta V - \frac{T_U * \Delta t'_U * g}{(w + w'_B) / 2}$$

$$w_a = w'_B \left(e^{-\frac{\Delta V' * \dot{w}_s}{g * T_s}} \right)$$

$$\Delta t_M = \frac{w'_B - w_a}{\dot{w}_s}$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMMPT
DATE 10/3/67 (EORP)
REV
PAGE 6 of 22

The thrust profile first moments due to ullage are ignored with only first moments of the main thruster being balanced about the centroid (impulsive) time. Making this assumption for this case, "n" in equation (1) is equal to 1. Therefore,

$$\text{GMTI} = \text{TIMP} - \frac{w'_B}{\dot{w}_s} + \frac{\Delta t_M}{\ln\left(\frac{w_{B'}}{w_a}\right)}$$

Since above is time to start main thruster, then

$$\text{GMTBB} = \text{GMTBI} - \Delta t_{U'} \quad \text{is time for ullage on.}$$

Case 3. The specified thruster is LM DPS.

$$\Delta t_M = \Delta t_{U'} = \Delta t_U - \Delta t_U \quad (\text{overlap}) \quad \text{actual ullage time before main burn.}$$

There are several points to be noted for this case. If the time of 10% burn is input as positive, a test is made to determine whether the maneuver can be completed in some maximum 10% burn time (currently 88 seconds). If the 10% burn time is input zero, a system parameter is used for its value. If the time of 10% in input negative, the absolute value of it is used, and the test for maximum time of 10% burn is ignored.

When the phase of the burn is known, which will end burn, then the time of burn for that phase is computed as follows:

$$\Delta t_M = \frac{w}{\dot{w}} \left(1 - e^{\frac{-\dot{w} * \Delta V}{g * T}} \right)$$

where: w = weight at start of this phase

$\dot{w}$ = weight flow rate

T = thrust level

ΔV = velocity required

g = acceleration of gravity

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMMPT
 DATE 10/3/67 (EORP)
 REV
 PAGE 7 of 22

Note the begin burn time must be computed for this case. Since we are considering all first moments of the thrust profile, "n" in equation (1) is equal to "3". Upon integrating equation (1) and rearranging items, the following result is obtained.

$$S = T * g / \dot{w}$$

$$A = S h \frac{w_d}{w_f}$$

$$B = \frac{w_d}{\dot{w}} + \Delta t_M A - S \Delta t_M$$

T = thrust of phase under study

g = acceleration of gravity

$\dot{w}$ = weight flow rate of phase understudy

w_d = weight at start of phase, understudy

w_f = weight at end of phase understudy

Δt_m = time of burn of phase understudy

$$D = \Sigma B$$

$$C = \Sigma A$$

$$GMTBB = TIMP - \frac{D}{C}$$

Time to begin ullage

$$GMTBI = GMTBB + \Delta t'U$$

Time of DPS thruster on.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMMPT
 DATE 10/3/67 (EORP)
 REV
 PAGE 8 of 22

Table PMMMPT-1. System Parameters

Symbol in Program	Symbol in Master Tape	Type *	Units	Comments
GRAV	MCEAGC	D	E_r/hr^2	g (acceleration of gravity)
SRMU	MCERMU	D	$\sqrt[3]{E_r/hr^2}$	$\sqrt{u}$
	MCTCW1	D	lbs./hr.	Weight flow rate CSM RCS + X (2 quads)
	MCTCT1	D	lbs.	Thrust of CSM RCS + X (2 quads)
	MCTCW2	D	lbs./hr.	Weight flow rate CSM RCS + X (4 quads)
	MCTCT2	D	lbs.	Thrust of CSM RCS + X (4 quads)
	MCTCW3	D	lbs./hr.	Weight flow rate CSM RCS - X (2 quads)
	MCTCT3	D	lbs.	Thrust of CSM RCS - X (2 quads)
	MCTCW4	D	lbs./hr.	Weight flow rate CSM RCS - X (4 quads)
	MCTCT4	D	lbs.	Thrust of CSM RCS - X (4 quads)
	MCTLW1	D	lbs./hr.	Weight flow rate LM RCS + X (2 quads)
	MCTLT1	D	lbs.	Thrust of LM RCS + X (2 quads)
	MCTLW2	D	lbs./hr.	Weight flow rate LM RCS + X (4 quads)
	MCTLT2	D	lbs.	Thrust of LM RCS + X (4 quads)

*Type
 H - Fixed-point half word
 F - Fixed-point full word
 E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMMPT
 DATE 10/3/67 (EORP)
 REV.
 PAGE 9 of 22

Table PMMMPT-1. System Parameters (Cont'd.)

Symbol in Program	Symbol In Master Tape	Type *	Units	Comments
	MCTLW3	D	lbs./hr.	Weight flow rate LM RCS - X (2 quads)
	MCTLT3	D	lbs.	Thrust of LM RCS - X (2 quads)
	MCTLW4	D	lbs./hr.	Weight flow rate LM RCS - X (4 quads)
	MCTLT4	D	lbs.	Thrust of LM RCS - X (4 quads)
	MCTSD9	D	hrs.	Ullage overlap the SPS thruster
	MCTSW4	D	lbs./hr.	Weight flow rate SPS thruster
	MCTAD9	D	hrs.	Ullage overlap the APS thruster
	MCTAW4	D	lbs./hr.	Weight flow rate APS thruster
	MCTAT4	D	lbs.	Thrust level APS thruster
	MCTDD2	D	hrs.	Ullage overlap DPS burn
	MCTDW5	D	lbs./hr.	Weight flow rate DPS thruster full
	MCTDT5	D	lbs.	Thrust level DPS thruster full
	MCTDT3	D	lbs.	Thrust level DPS thruster 10%
	MCTDW3	D	lbs./hr.	Weight flow rate DPS thruster 10%

*Type
 H - Fixed-point half word
 F - Fixed-point full word
 E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal



Table PMMMPT-1. System Parameters (Cont'd.)

Symbol in Program	Symbol in Master Tape	Type*	Units	Comments
	MCTDD3	D	hrs.	System parameter value for DPS 10% burn
	MCTDD7	D	hrs.	Max. 10% burn time
	MCTAPG	D	rad.	APS Pitch Gimble angle constant
	MCTARG	D	rad.	APS Roll Gimble angle constant
	MCTST4	D	lbs.	Thrust level SPS thruster

*Type
 H — Fixed-point half word
 F — Fixed-point full word
 E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

2. PMMMPT
 DATE 10/3/67 (EORP)
 REV
 PAGE 11 of 22

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

Table PMMMPT-2. Parameter List

QUEID — NA

QNAME — NA

From — NA

To-Task — NA

Purpose — NA

EP — NA

GP REG 11 must point to this list.

Byte No.	Type*	Units	Value / Description / Comments	
1-272			44D Not used	
273-512			20D Search area for PMMXFR	
513-616			13D Search area for this program	
617-624			D Spare	
625-656			4D Not used	
657-660	F	None	Thruster Code	
661-664	F	None	Attitude Mode Code	
665-669	F		Spare	
669-672	F	None	Ullage Thruster Code 0 = Two thrusters 1 = Four thrusters	
673-680	D	hrs.	Time of ullage burn	
681-688	D	hrs.	Time of DPS 10% burn	
689-760			9D Spare	
761-764	F		Spare	
765-768	F	None	Configuration code at start	
769-772	F		Spare	
773-776	F	None	Vehicle code	
777-784	D	rad.	Docking angle if vehicles docked	
785-888			13D Not used	
889-896	D	sq. ft.	Vehicle area	
897-904	D	lbs.	Vehicle weight	
905-912	D	hrs.	Start burn limit ¹	
913-920	D	hrs.	End burn limit	
921-928	Alpha	hrs.	EBCDIC code for maneuver	
929-936	D	Er	Semi major axis before maneuver	a
937-944	D	None	Orbit ecc. before maneuver	e
945-952	D	rad.	Orbit inclination before maneuver	i
953-960	D	rad.	Mean anomaly before maneuver	l
961-968	D	rad.	Argument of perigee before maneuver	g
969-976	D	rad.	Argument of ascending mode before maneuver	h
977-984	D	Er	Semi-major axis after maneuver	a
985-992	D	None	Orbit ecc. after maneuver	e
993-1000	D	rad.	Orbit inclination after maneuver	i

*Type
 H — Fixed-point half word
 F — Fixed-point full word

E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS
 2. PMMMPT
 DATE 10/3/67 (EOR)
 REV
 PAGE 12 of 22

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

Table PMMMPT-2. Parameter List (Cont'd.)

QUEID — NA

QNAME — NA

From — NA

To-Task — NA

Purpose — NA

EP — NA

Byte No.	Type*	Units	Value / Description / Comments
1001-1008	D	rad.	Mean anomaly after maneuver
1009-1016	D	rad.	Argument of perigee after maneuver
1017-1024	D	rad.	Argument ascending node after maneuver
1025-1032	D	hrs.	Impulsive time of maneuver
1033-1088			Remaining quantities of impulsive elements not used by this program
1089-2048			Header block of Mission Plan Table

 *Type
 H — Fixed-point half word
 F — Fixed-point full word

 E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

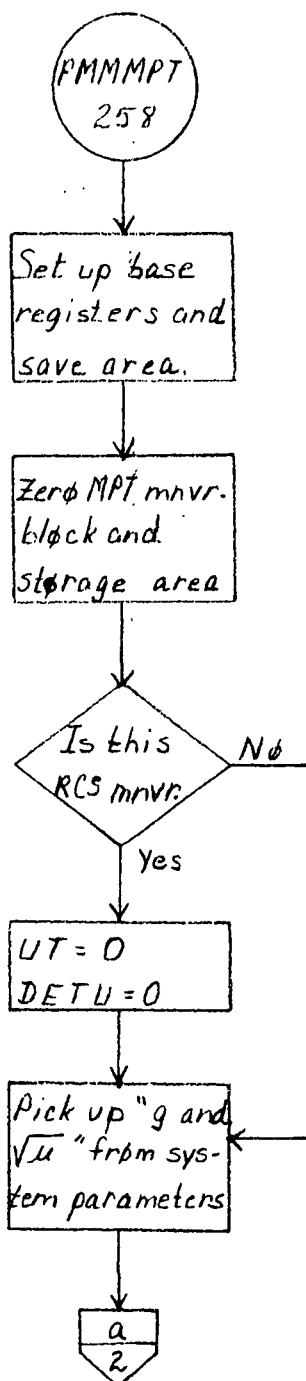


Figure PMMMPT-1. Flowchart

(Sheet 1 of 10)

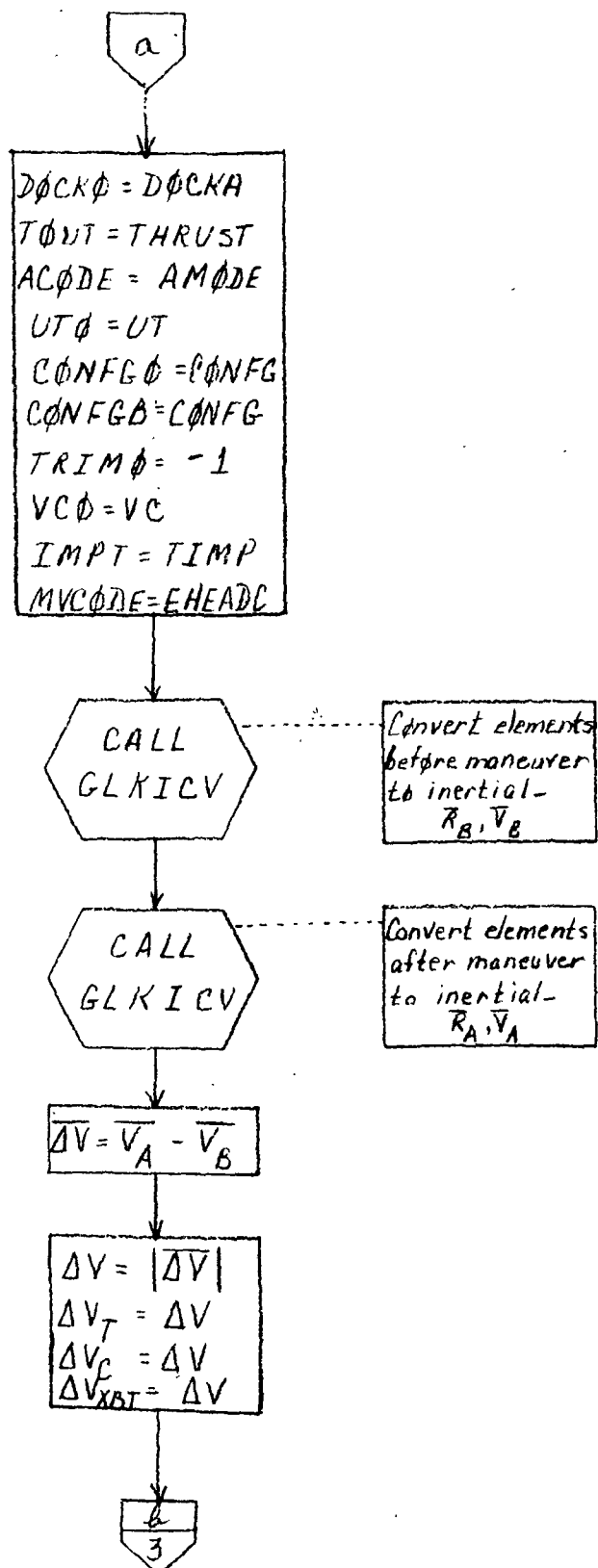


Figure PMMMPT-1. Flowchart

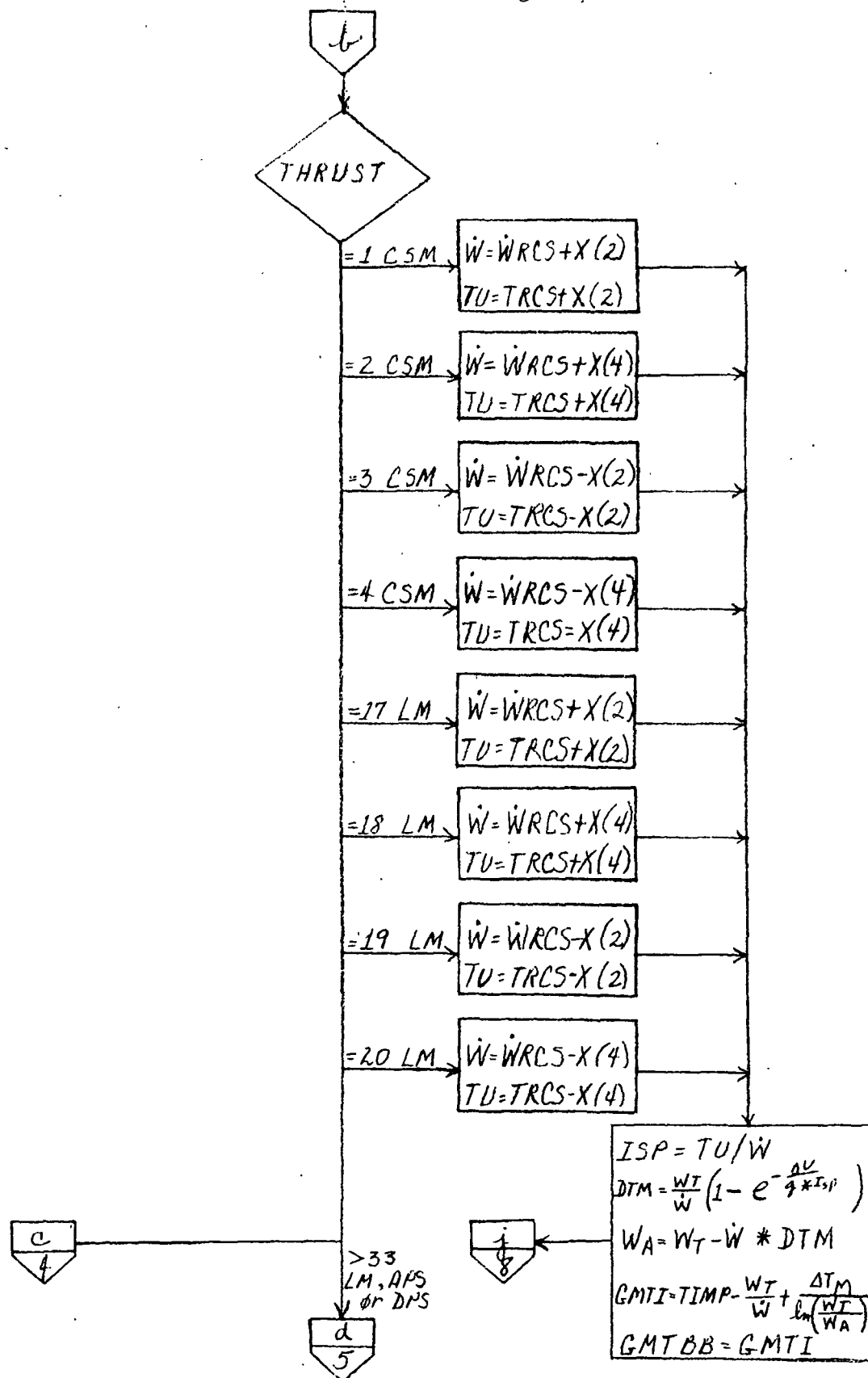


Figure PMMMPT-1. Flowchart

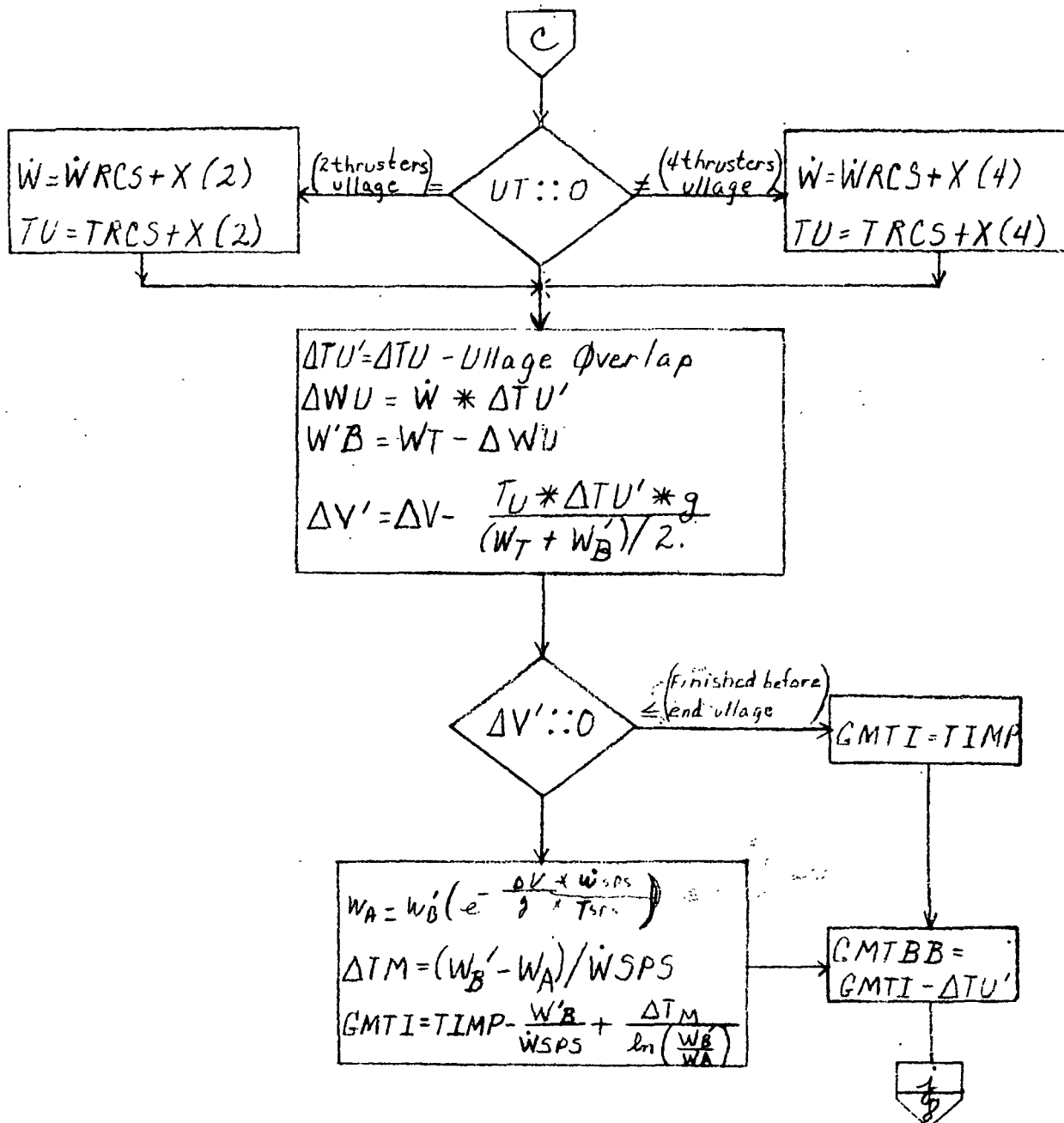


Figure PMMMPT-1. Flowchart

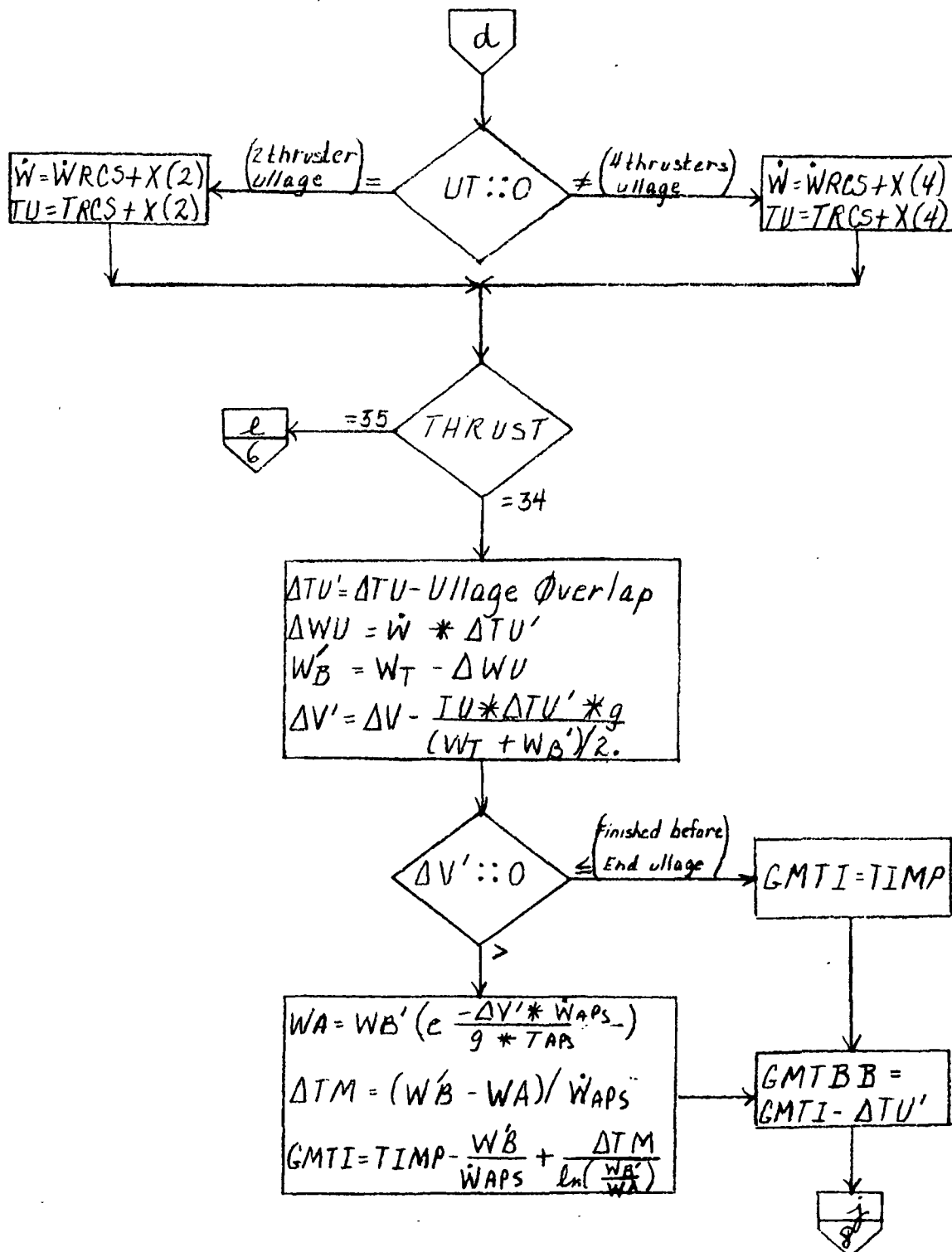


Figure PMMMPT-1. Flowchart

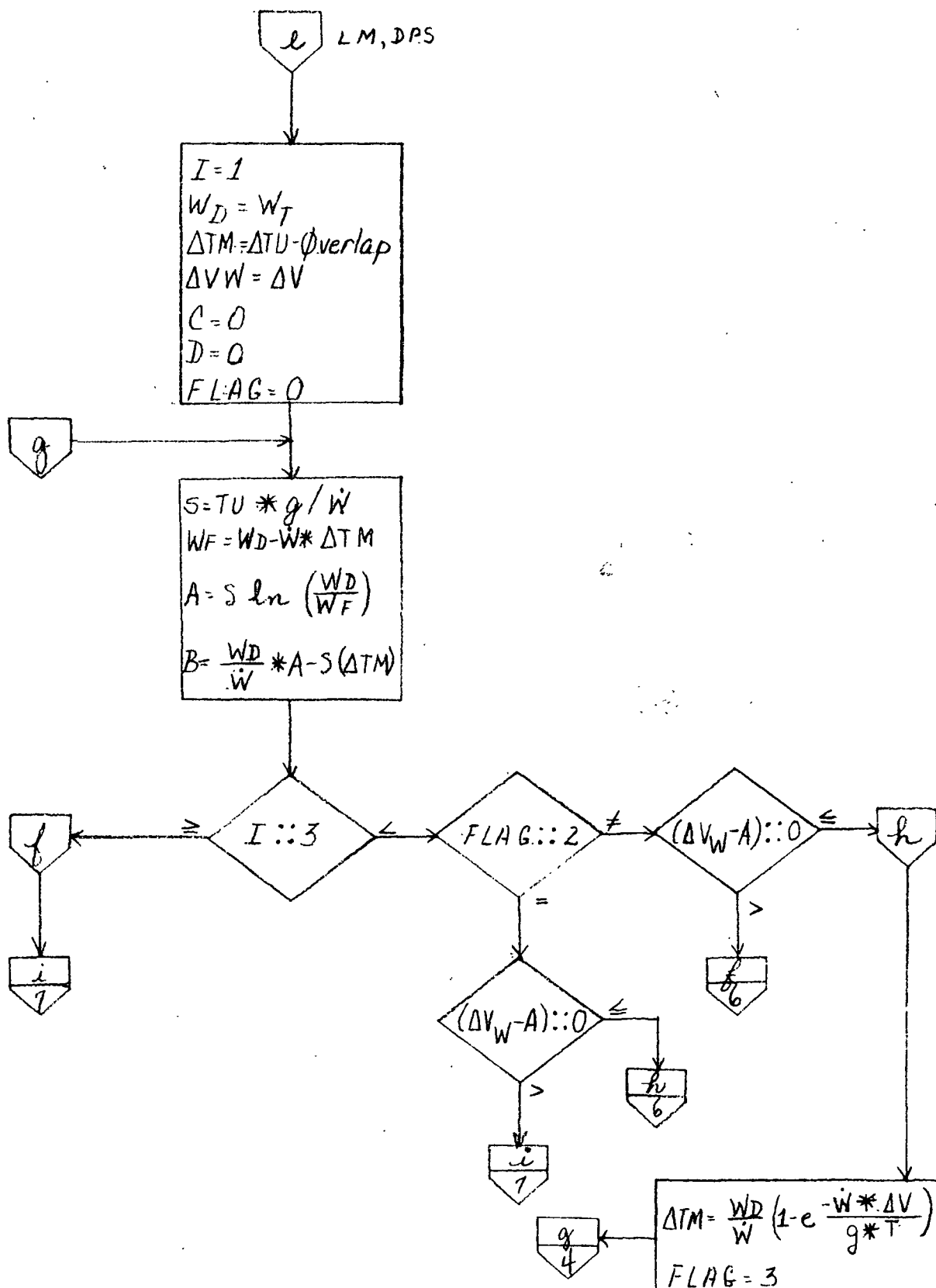


Figure PMMMPT-1. Flowchart

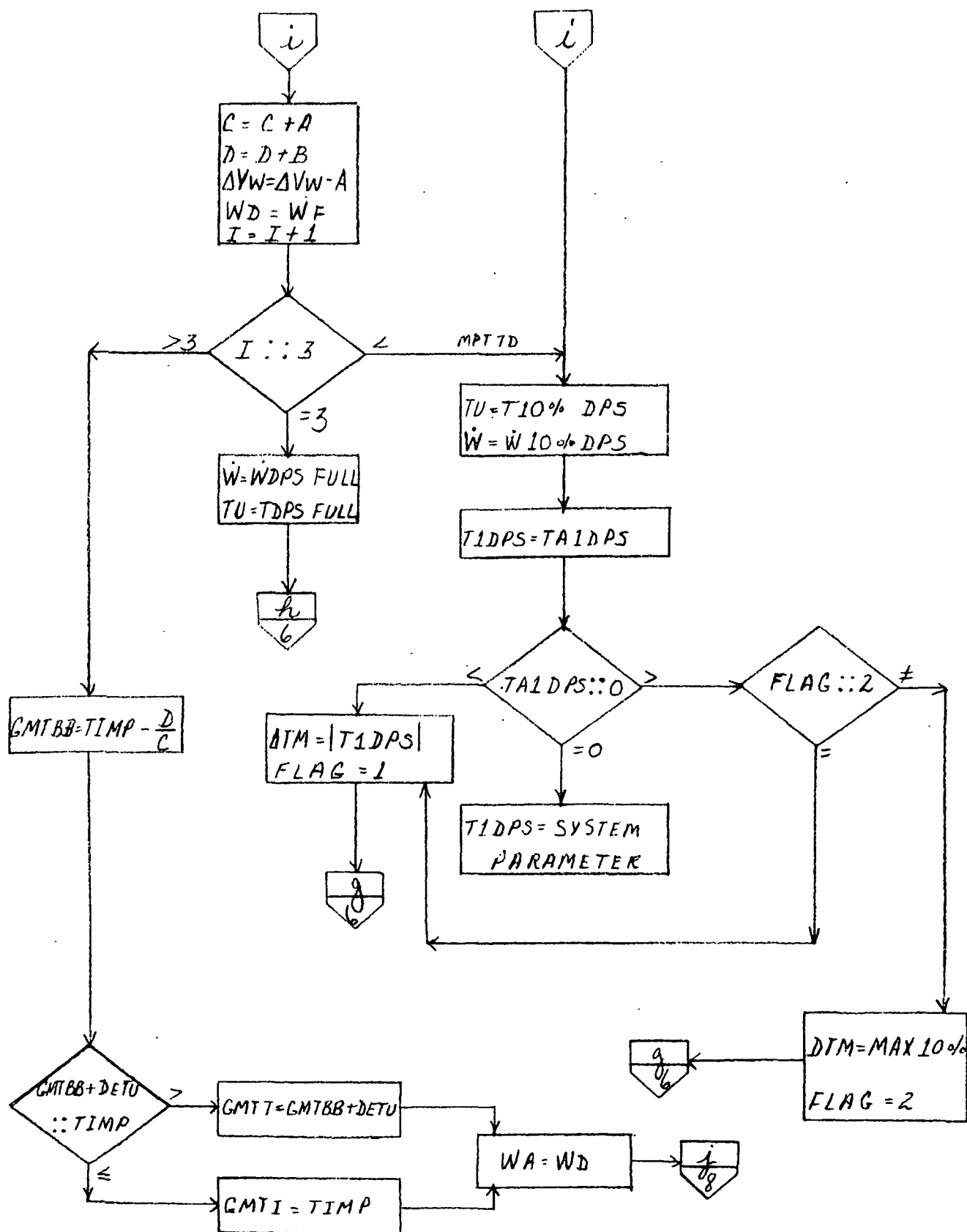


Figure PMMMPT-1. Flowchart

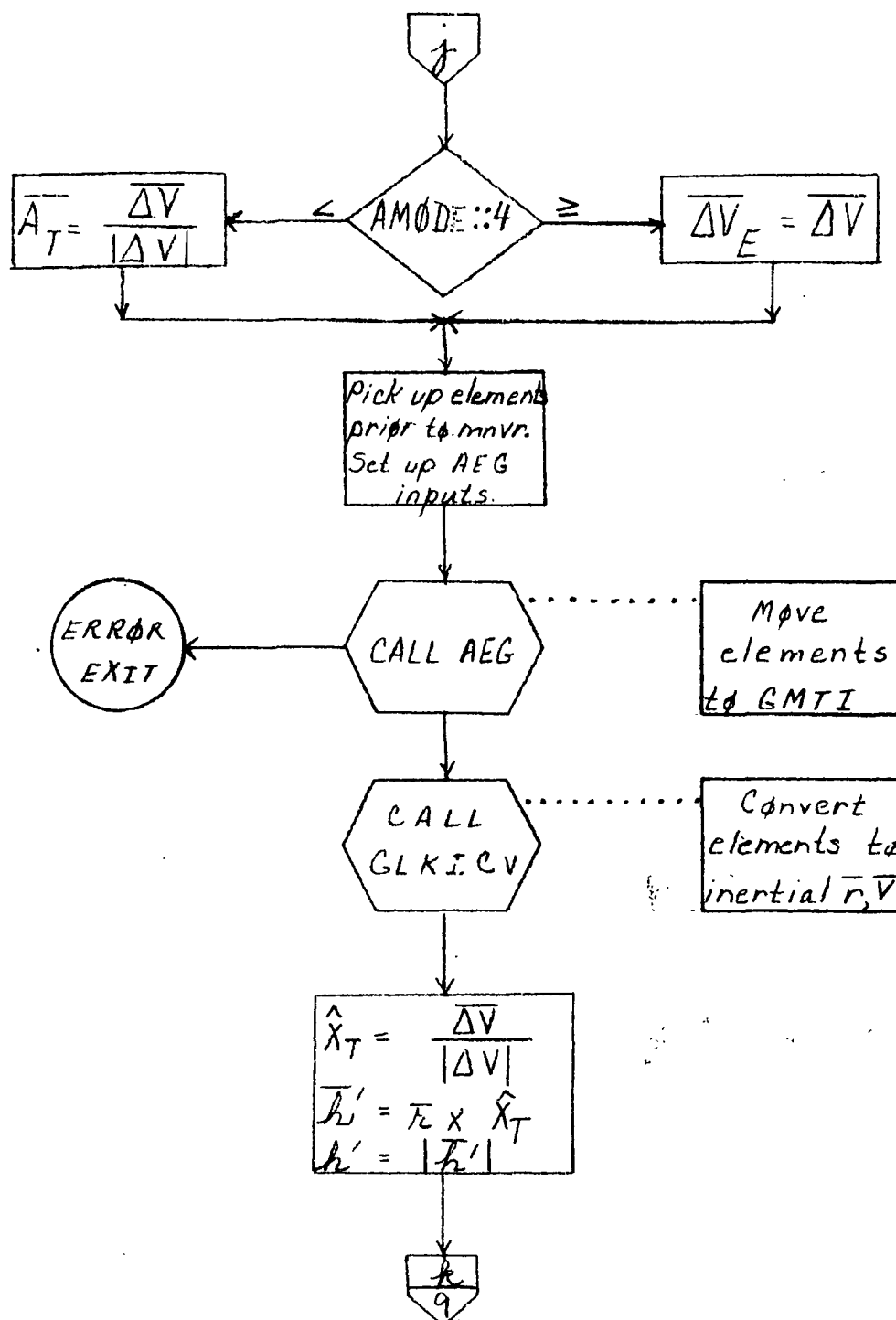


Figure PMMMPT-1. Flowchart

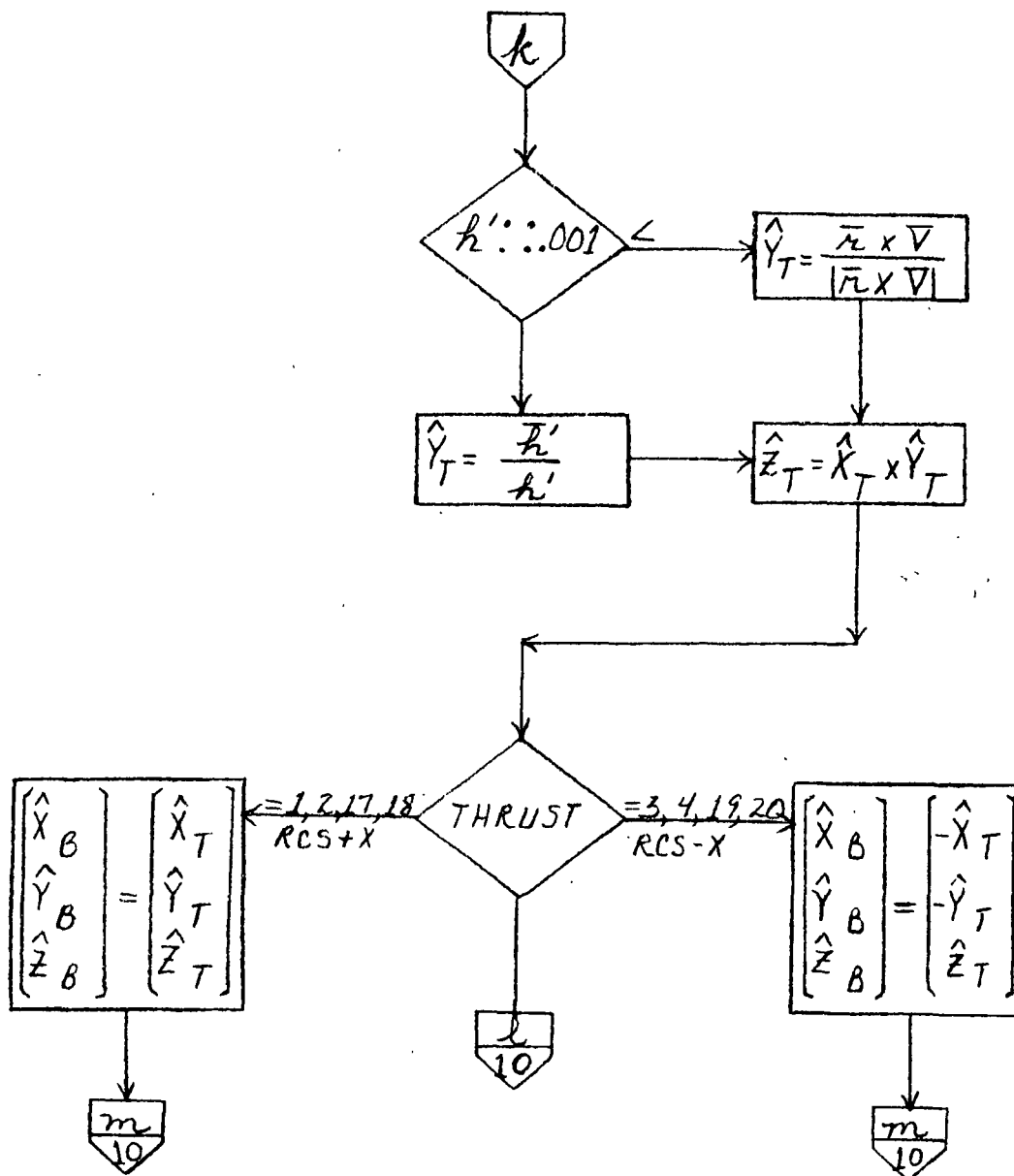


Figure PMMMPT-1. Flowchart

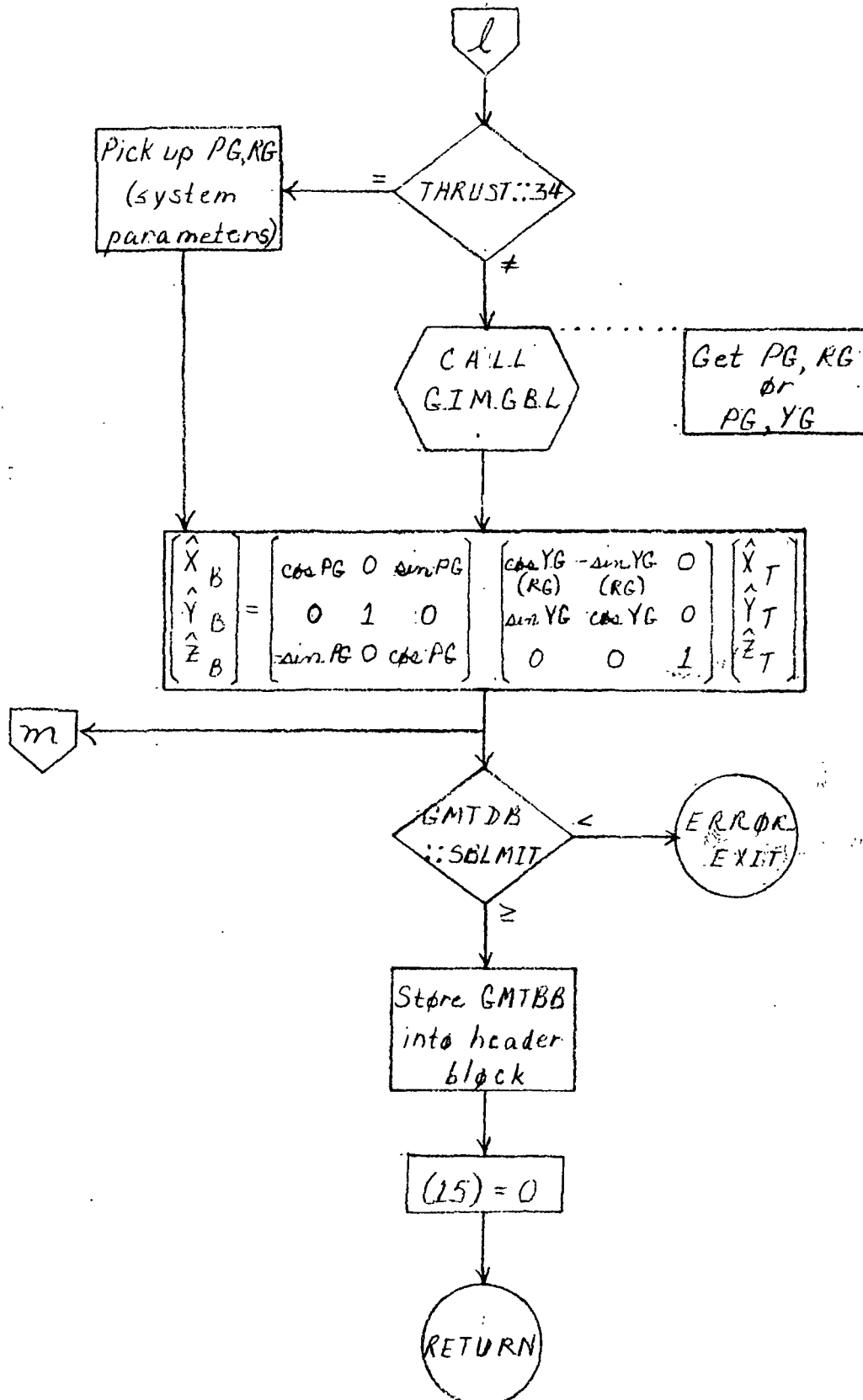


Figure PMMMPT-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMPAD

DATE 8/25/67 (EORP)

REV

PAGE 1 of 14

- a. Symbol — PMMPAD
- b. Program Name — Perigee Adjust Computations
- c. Programming Group — Rendezvous Planning
- d. Programmer(s) — C. Peabody, R. Taylor, R. Gayle
- e. Language — FORTRAN IV - H Level
- f. Data Tables Used — PZPADDIS
- g. MED's Used — K-28 - Perigee Adjust Request
- h. Writeup Completion Date — 8/31/67
- i. Data Table Format — Table PMMPAD-1
- j. Flowchart — Figure PMMPAD-1

FUNCTION

PMMPAD generates pitch, burn time, velocity increment, burn duration, true anomaly, height above spherical Earth, and height of resultant apogee for six maneuvers (if possible), any one of which will cause perigee to be raised to some requested value.

USAGE

- a. Calling Sequence — LINK EP=PMMPAD
- b. Inputs — AEG inputs at a vector time, a threshold time, Δt , apogee and perigee heights
- c. Outputs — Maneuver Information into a display table
- d. Other Programs Used — PMMAEG, GIMAPD
- e. Data Restrictions and Limits — Solutions cannot be obtained near perigee. Perigee height requested can not be greater than current maneuver height.
- f. Error — If number of maneuvers = 0, current perigee is greater than requested, and no maneuvers will be computed. If there are no maneuvers, the times are set negative.

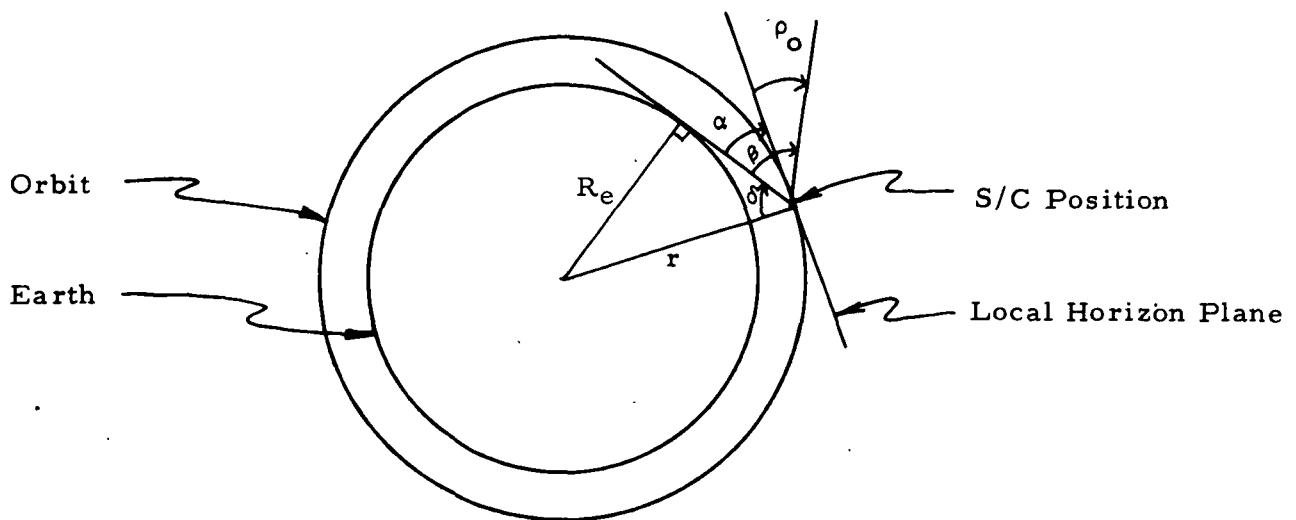
METHOD

If present perigee is less than requested, the elements at threshold time are found and used along with the pitch, computed from the angle with the horizon to compute ΔV of the maneuver. Pitch and ΔV are then used to find the elements after the maneuver, which in turn are used to find perigee. If perigee is not the desired, an iterative procedure is used to find the desired. Apogee is then found, and display quantities computed. This procedure is repeated for six maneuvers.

Approval	Approval
<i>C. Peabody</i>	<i>A. Aricetti</i>

To apply velocity at a constant angle β with the horizon, pitch is computed as:

$$\rho_o = \beta - \cos^{-1} \frac{R_e}{r}$$



Where:

R_e = Radius of the Earth

r = Radius magnitude at maneuver time

ρ_o = Pitch angle

$\rho_o = \beta - \alpha$

$\alpha = 90^\circ - \delta$ or $\delta = 90^\circ - \alpha$

$$\sin \delta = \frac{R_e}{r} = \sin (90^\circ - \alpha) = \cos \alpha$$

$$\text{or } \alpha = \cos^{-1} \frac{R_e}{r}$$

hence

$$\rho_o = \beta - \cos^{-1} \frac{R_e}{r}$$

To compute the ΔV (magnitude increment) needed to raise perigee to the desired magnitude (R_{PD}), the following method is used:

Subscript b signifies before maneuver and a, after.

Where:

a = Semimajor axis

e = Eccentricity

R = Desired magnitude of perigee

h = Magnitude of angular momentum vector

r = Radius magnitude

V = Velocity

ρ = Pitch angle

V_H = Horizontal component of velocity

$\dot{r}$ = Rate of change of position magnitude

$h^2 = a(1 - e^2) = a(1 - e)(1 + e) = R(1 + e)$, where R is defined as
 $a(1 - e)$

Solving for e and $1 - e$,

$$e = \frac{h^2 - R}{R}$$

$$1 - e = 1 + \frac{R - h^2}{R} = \frac{2R - h^2}{R}$$

Defining $a = \frac{r}{2 - rv^2}$ and substituting these values for a and e in the equation $R = a(1 - e)$, results in

$$R = \frac{(2R - h^2)}{R} \cdot \frac{r}{2 - rv^2} = \frac{2Rr - h^2 r}{2R - Rrv^2}$$

$$2R^2 - R^2 rv^2 = 2Rr - h^2 r$$

$$rh^2 - R^2 rv^2 = 2Rr - 2R^2$$

$$h^2 - R^2 v^2 = \frac{2R(r - R)}{r} \quad \text{for later reference} \quad (1)$$

By definition:

$$\Delta V_H = \Delta V \cos \rho$$

$$\Delta \dot{r} = \Delta V \sin \rho$$

$$V_a^2 = (V_{H_b} + \Delta V_H)^2 + (\dot{r}_b + \Delta \dot{r})^2 = (V_{H_b} + \Delta V \cos \rho)^2 + (\dot{r}_b + \Delta V \sin \rho)^2$$

Expanding and combining:

$$V_a^2 = V_{H_b}^2 + 2V_b \Delta V \cos \rho + \Delta V^2 \cos^2 \rho + \dot{r}_b^2 + 2\dot{r}_b \Delta V \sin \rho + \Delta V^2 \sin^2 \rho$$

$$= V_b^2 + \Delta V (2V_{H_b} \cos \rho + 2\dot{r}_b \sin \rho) + \Delta V^2 (\cos^2 \rho + \sin^2 \rho)$$

$$V_a^2 = V_b^2 + \Delta V [2(V_{H_b} \cos \rho + \dot{r}_b \sin \rho)] + \Delta V^2 \quad (2)$$

$$h_a = r V_{H_a}$$

$$V_{H_a} = V_{H_b} + \Delta V \cos \rho$$

$$h_a^2 = r^2 V_{H_a}^2 = r^2 (V_{H_b} + \Delta V \cos \rho)^2$$

$$h_a^2 = r^2 (V_{H_b}^2 + 2\Delta V V_{H_b} \cos \rho + \Delta V^2 \cos^2 \rho) \quad (3)$$

Substituting equations (2) and (3) into equation (1),

$$r^2 (V_{H_b}^2 + 2\Delta V V_{H_b} \cos \rho + \Delta V^2 \cos^2 \rho) - R^2 \left\{ V_b^2 + \Delta V \left[2(V_{H_b} \cos \rho + \dot{r}_b \sin \rho) \right] + \Delta V^2 \right\} = \frac{2R(r - R)}{r} = 0$$

Combining and grouping,

$$\Delta V^2 (\cos^2 \rho r^2 - R^2) + \Delta V \left[2V_{H_b} \cos \rho (r^2 - R^2) - 2R^2 \dot{r}_b \sin \rho \right] + \left[r^2 V_{H_b}^2 - R^2 V_b^2 + \frac{2R(R - r)}{r} \right] = 0$$

Solving for ΔV by quadratic formula,

$$A = r^2 \cos^2 \rho - R^2$$

$$B = 2V_{H_b} \cos \rho (r^2 - R^2) - 2R^2 \dot{r}_b \sin \rho$$

$$C = r^2 V_{H_b}^2 - R^2 V_b^2 + \frac{2R(R - r)}{r}$$

$$D = B^2 - 4AC$$

$$\Delta V_1 = \frac{-B + \sqrt{D}}{2A}$$

$$\Delta V_2 = \frac{-B - \sqrt{D}}{2A}$$

ΔV is chosen as the minimum positive value.

Elements after the maneuver are then computed as follows:

$$\dot{r}_a = \dot{r}_b + \Delta V \sin \rho$$

$$V_{H_a} = V_{H_b} + \Delta V \cos \rho$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMPAD
DATE 8/25/67 (EORP)
REV
PAGE 6 of 14

$$V_a^2 = \dot{r}_a^2 + V_{Ha}^2$$

$$A_a = \frac{\mu r}{2\mu - rV_a^2}$$

$$e_a \cos E_a = \frac{a_a - r}{a_a}$$

$$e_a \sin E_a = \frac{\dot{r}_a r}{\mu a_a}$$

$$E_a = \tan^{-1} (e_a \sin E_a / e_a \cos E_a) \quad 0 \leq E_a < 2\pi$$

$$e_a = [(e_a \sin E_a)^2 + (e_a \cos E_a)^2]^{1/2}$$

$$l_a = E_a - e_a \sin E_a$$

$$i_a = i_b$$

$$h_a = h_b$$

$$f_a = \tan^{-1} [\sin E_a \quad 1 - e_a^2 / (\cos E_a - e_a)] \quad 0 \leq f_a < 2\pi$$

$$g_a = u_b - f_a \quad 0 \leq g_a < 2\pi$$

These elements are then used to find the resulting perigee. For the first pass

$$R = R_{PD} + (R_{PKEP} - R_{PC})$$

Where,

$$R_{PD} = \text{radius of perigee desired}$$

$$R_{PKEP} = \text{radius of perigee from elements} = a(1 - e)$$

$$R_{PC} = \text{current radius at perigee}$$

If this value does not result in a perigee radius within some epsilon range of the desired, the R is computed the first time as

$$R = R - \Delta R_{PER_c}$$

Where,

$$\Delta R_{PER_o} = R_{PD} - R_{PC}$$

$$\Delta R_{PER_c} = \Delta R_{PER_o}$$

For successive tries, the correction procedure is:

$$\Delta R_{PER_1} = R_{PD} - R_{PC}$$

$$\Delta \Delta R_{PER} = \Delta R_{PER_o} - \Delta R_{PER_1}$$

$$\Delta R_{PER_c} = \frac{\Delta R_{PER_1} \Delta R_{PER_c}}{\Delta \Delta R_{PER}}$$

$$\Delta R_{PER_o} = \Delta R_{PER_1}$$

$$R = R + \Delta R_{PER_c}$$

The pitch that was previously computed was based on alignment with the horizon at impulsive maneuver time. The manner in which the actual maneuver is to be performed is to line up on the horizon at maneuver initiation time and then to hold that attitude inertially. However, maneuver initiation time and pitch are at that time dependent upon each other, and hence an iteration scheme is necessary. The equation used is

$$P = P_o + \Delta P$$

where

$$P_o = \text{Pitch at impulsive time.}$$

$$\Delta P = \text{Correction made to } P_o \text{ to approximate pitch at maneuver initiation time } (\Delta P = \text{approximate central angle difference between initiation and impulsive time})$$

$$\Delta P = \frac{\Delta t_{\text{MAN}}}{2} \dot{f}$$

where Δt_{MAN} is approximated by

$$\Delta t_{\text{MAN}} = \frac{\Delta V W_E}{W I_{sp} G}$$

which is derived from

$$F = m A \quad (\text{Newton})$$

$$F = \dot{W} I_{sp} \quad (\text{by definition})$$

$$m = W/G \quad (\text{by definition})$$

$$\Delta t_{\text{MAN}} = \frac{\Delta V}{A} \quad \text{assuming constant acceleration.}$$

$$\text{Also } \dot{f} = \frac{\sqrt{\mu a_a (1 - e_a^2)}}{r^2}$$

$$G = \text{Acceleration due to gravity at the reference latitude}$$

$$I_{sp} = \text{Specific impulse for SPS thruster}$$

$$W_B = \text{Weight of vehicle prior to maneuver}$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMPAD

DATE 8/25/67 (EORP)

REV

PAGE 9 of 14

 ΔW_{MAN} = Weight of propellant used W = Weight flow rate Δt_{MAN} = Maneuver duration ΔV = Velocity increment magnitude m = Mass A = Acceleration F = Force (thrust level)

After an acceptable value for perigee has been computed, a more accurate value for the time required for the maneuver is computed as

$$\Delta W_{MAN} = W_B \left(1 - e^{\frac{-\Delta V}{gI_{sp}}} \right)$$

where e = the natural number (2.71828...).

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMPAD
DATE 8/25/67 (EORP)
REV
PAGE 10 of 14

Table PMMPAD-1. Data Table Formats

Symbolic Name — PZPADDISProgrammer — C. PeabodyTable Name — Perigee AdjustInitialization of any Element Required — NADimensions

Blocks — 1
Words/Blocks — 50 double words
Total Words — 50 double words

General Description and Usage

This table contains information for six maneuvers to raise perigee to the requested value. The program PMDPAD will display this information.



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMPAD

DATE 8/25/67 (EORP)

REV

PAGE 11 of 14

Detailed Contents

Block	Word	Byte	Bit	Type*	Units	Description and Range
	1			D		Solution Indicator
	2			D	hrs.	Time of Elements
	3			D	hrs.	Thresh Time
	4			D	hrs.	Time Step
	5			D	Er	Apogee Height
	6			D	Er	Perigee Height
	7			D		Blank
	8			D	rad.	Pitch at Ignition
	9			D	hrs.	Burn Time
	10			D	Er/hr.	Velocity Increment
	11			D	hrs.	Burn Duration
	12			D	rad.	True Anomaly
	13			D	Er	Height Above Earth
	14			D	Er	Height of Resultant Apogee
	15-21			D		Same as Words 8-14 for Second Maneuver
	22-28			D		Same as 8-14 for Third Maneuver
	29-35			D		Same as 8-14 for Fourth Maneuver
	36-42			D		Same as 8-14 for Fifth Maneuver
	43-49			D		Same as 8-14 for Sixth Maneuver
	50			D		Blank

*Type

H — Fixed-point half word
F — Fixed-point full wordE — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMPAD
 DATE 8/25/67 (EORP)
 REV
 PAGE 12 of 14

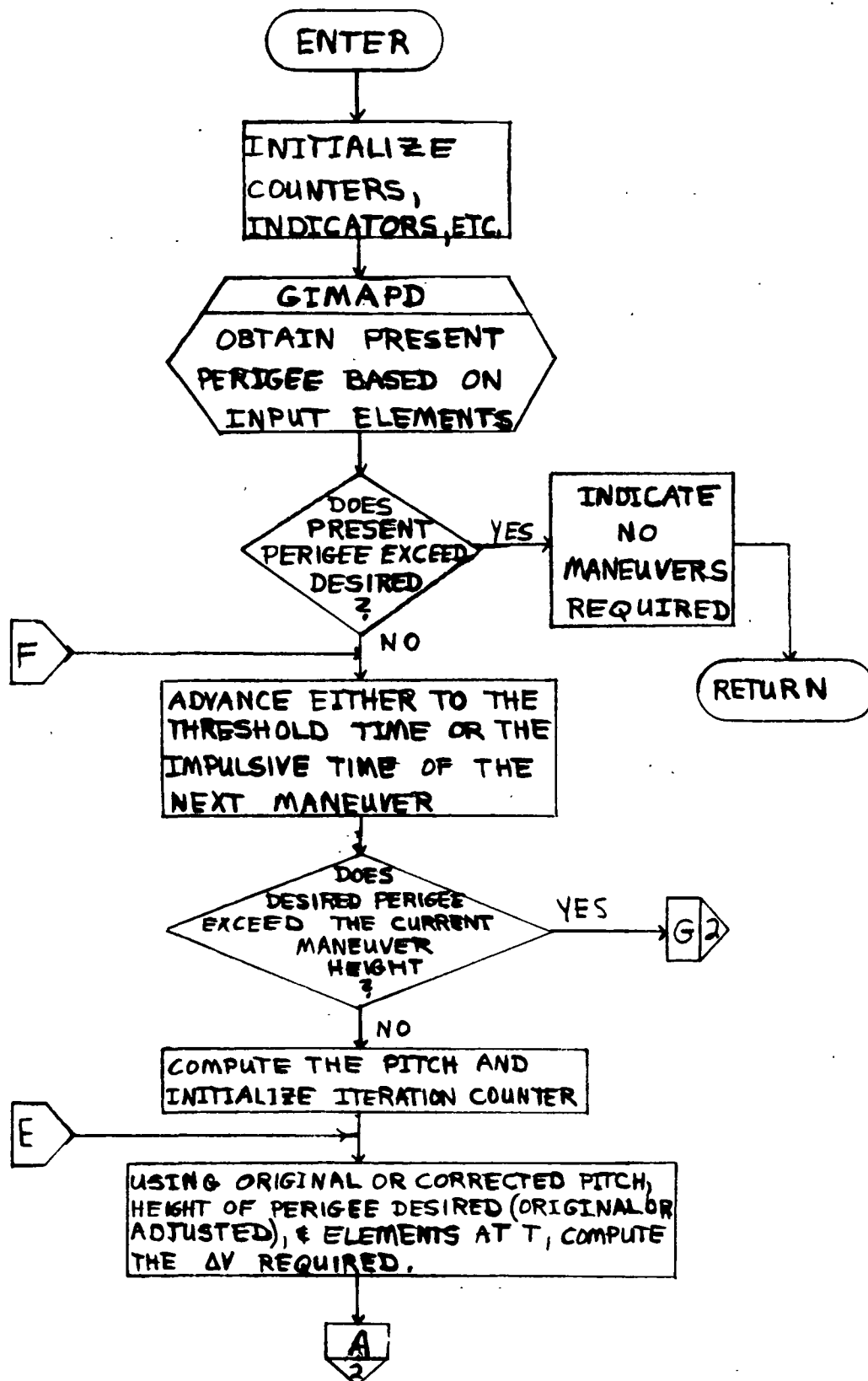


Figure PMMPAD-1. Flowchart

(Sheet 1 of 3)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMPAD

DATE 8/25/67 (EORP)

REV

PAGE 13 of 14

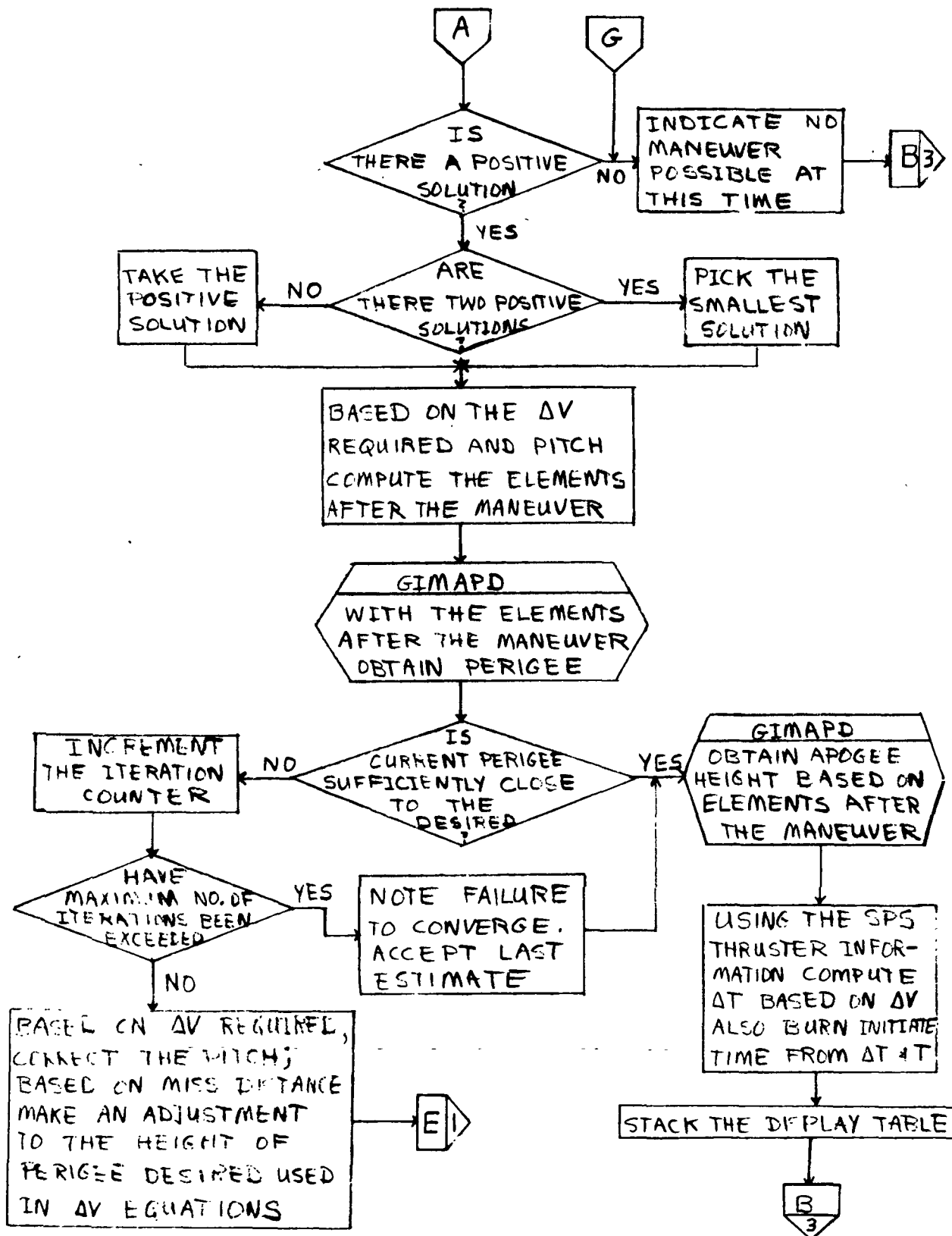


Figure PMMPAD-1. Flowchart

(Sheet 2 of 3)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

2. PMMPAD

DATE 8/25/67 (EORP)

REV

PAGE 14 of 14

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

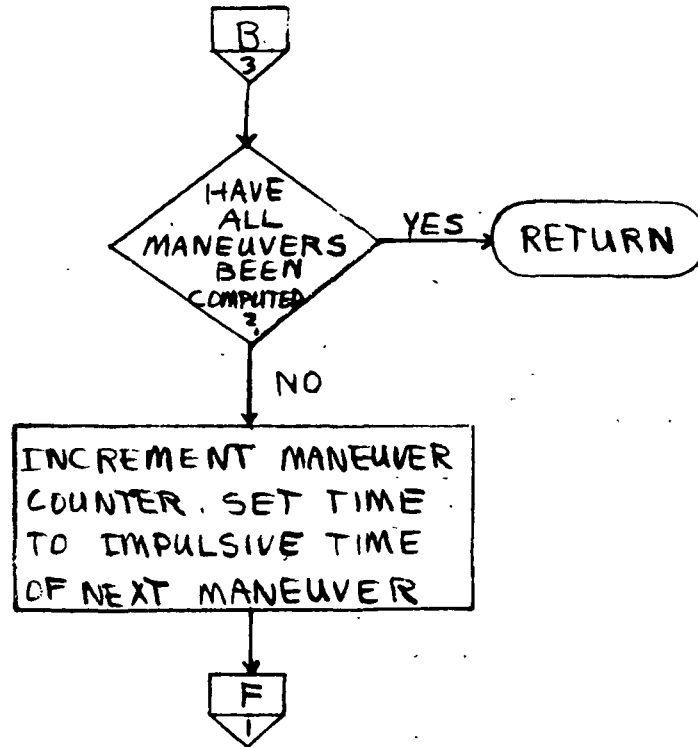


Figure PMMPAD-1. Flowchart

(Sheet 3 of 3)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

2. PMMPAR
 DATE 11/21/67 (EORP)
 REV
 PAGE 1 of 26

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

- a. Symbol - PMMPAR
- b. Program Name - Saturn Launch Vehicle (SLV) Targeting Load Module
- c. Programming Group - Rendezvous Planning
- d. Programmer - T. Copper
- e. Language - Assembler
- f. Data Tables Used - PZMPTCSM, PZMPTLEM, PZSLVCON, PZSLVTAR, GZLTRA
- g. MED's Used - P10, K40
- h. Writeup Completion Date - 10/17/67
- i. System Parameters - Table PMMPAR-1
- j. Flowchart - Figure PMMPAR-1

FUNCTION

PMMPAR directs and computes the targeting quantities appearing on the SLV targeting display.

USAGE

- a. Calling Sequence - RTATTACH TASK=PTMANC, EP=PMMPAR, PARAM=(AREA, 72)
- b. Inputs - PZSLVCON, PZMPTCSM, PZMPTLEM
- c. Outputs and Exit - PZSLVTAR, GZLTRA RTATTACH TASK = GTMSCTRL, EPGMSCTRL, ID = 0, QRANL = 5
- d. Other Programs Used - DSQRT, DSIN, DCOS, DARSIN, ELEGETE, GIMIKC, PMMAEG
- e. Data Restrictions and Limits - None
- f. Error - This is an on-line message if there is an error from PMMAEG and ELGETE.

METHOD

The SLV Targeting Load Module (PMMPAR) is exercised during the period of determination of the first vehicle (target) trajectory and the lift-off time of the second vehicle (chaser); this is called the Prelaunch II phase of the mission.

The main function of PMMPAR is to compute the recommended lift-off time (GMTLOR), insertion velocity (V_{IGM}), and the SLV guidance parameters for the chaser vehicles.

The targeting unit uses two MED's, K40 and P10. The K40 MED is used to initialize or change constants that will be used by PMMPAR during a mission. The P10 MED is used only to queue PMMPAR during Prelaunch II. There are two basic options that can be used to queue the targeting unit.

1. Compute GMTLOR and V_{IGM} of the chaser vehicle for a first guess lift-off time of the chaser vehicle.
2. Compute the SLV guidance parameters and generate a chaser vehicle ephemeris using the computed nominal insertion vector based on the chaser's input lift-off time (GMTLO).

Approval	Approval
<i>J. Copper</i>	<i>A. Amicchi</i>

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMPAR
DATE 11/21/67 (EORP)
REV
PAGE 2 of 26

When the first option is used, a vector is fetched from the target vehicle ephemeris at GMTLO of the chaser vehicle, which was input. The chaser vehicle insertion velocity (V_{IGM}) is computed and based on the V_{IGM} , target vehicle ephemeris, and input constants; the nominal insertion vector for the chaser vehicle is then computed. Using the AEG, the chaser vehicle elements are updated to first apogee or some angle or time measured from chaser vehicle insertion; the target vehicle is updated to position match with the chaser vehicle. The height differences (ΔH) computed by the AEG are compared with the desired ΔH . If the differences in the ΔH fall outside the tolerance, the above process is repeated by adjusting the V_{IGM} . When the ΔH tolerance is met, GMTLO* is computed. GMTLO* is the time at which the chaser vehicle should lift off to have a zero degree phase angle at chasers' first apogee or some ΔT or ΔU measure from chaser vehicle insertion. To accomplish this, the phase lag at first apogee or some ΔT or ΔU measured from the insertion of the chaser vehicle is calculated by the AEG using the chaser and target vehicle element at chaser lift-off time. Using the computed phase angle, a time increment is then calculated to adjust the chaser lift-off time. The process is repeated until the chaser lift-off time results in a zero phase angle. The recommended lift-off time is GMTLOR = GMTLO* + BIAS (input constant).

When the second option is used, the same process is executed as in the first, except for the computations of GMTLOR. In addition to the above process, the chaser nominal insertion vector is stored for the launch program; and the SLV targeting parameters are computed and stored in PZSLVTAR for displaying. These parameters are sent by teletype format to Marshall Space Flight Center. The method used in computing the targeting parameters is referenced in MSC Internal Note, No. 64-FM-103, dated October 14, 1966, and the flowchart included in this writeup.

Depending on the option on the P10, the nominal insertion vector for the chaser vehicle may be passed on to a program to generate a trajectory for the chaser vehicle.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMPAR
 DATE 11/21/67
 REV
 PAGE 3 of 26

(EORP)

Table PMMPAR-1. System Parameters

Symbol in Program	Symbol In Master Tape	Type *	Units	Comments
FSPER	MCCVCU	D	ft./sec.	Feet per second per Earth radii hours
CUL	MCCNMC	D	nm	Nautical miles per CUL
CULF	MCCFCU	D	ft.	Feet per CUL
RAD	MCCDPR	D	deg.	Degrees per radian
WE	MCERTS	D	rad./hr.	Earth's rotational rate
RCAPE	MCECAP	D	CUL	Radius at Cape
JC	MCMSHP	D		Second harmonic
PI	MCPIR	D	rad.	π
TPI	MCMTPI	D	rad.	2π
UU	MCEMUU	D	E_r^3/PEH^2	Earth's gravitational constant
	MCERMU	D		Square root of MCCMUU

*Type
 H - Fixed-point half word
 F - Fixed-point full word
 E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMPAR
DATE 11/21/67 (EORP)
REV
PAGE 4 of 26

DTRROUTE, RTQCNTL, RTATTACH Macros

Macro - RTATTACH TASK = GTISPLAY, EP = GMSNAP, ID = 3 PARAM = (word, 4, FKWORD)

Purpose - To display the SLV Targeting display MSK 263

Macro - RTATTACH = QLOC = QNAME, TASK = GTMSCTRL, EP = GMCCTRL QRANK = 5
ID = 0

Purpose - Generate a trajectory for second vehicle to lift off

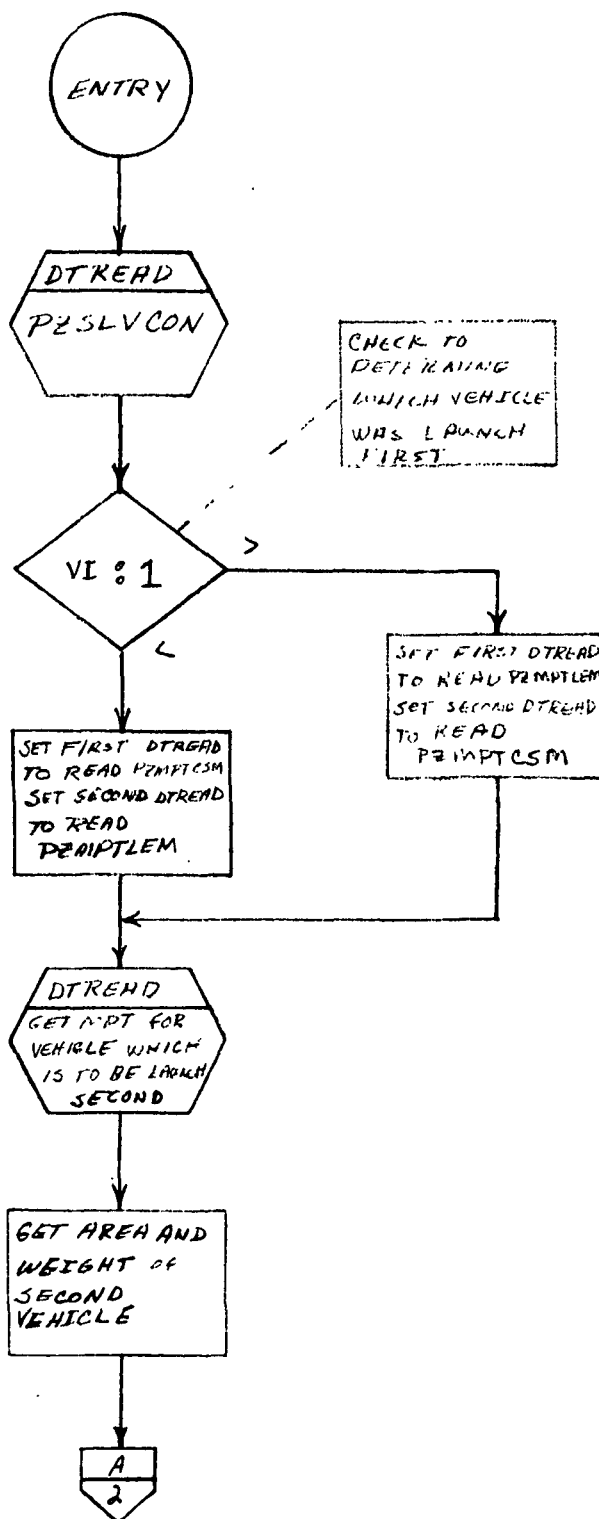


Figure PMMPAR-1. Flowchart

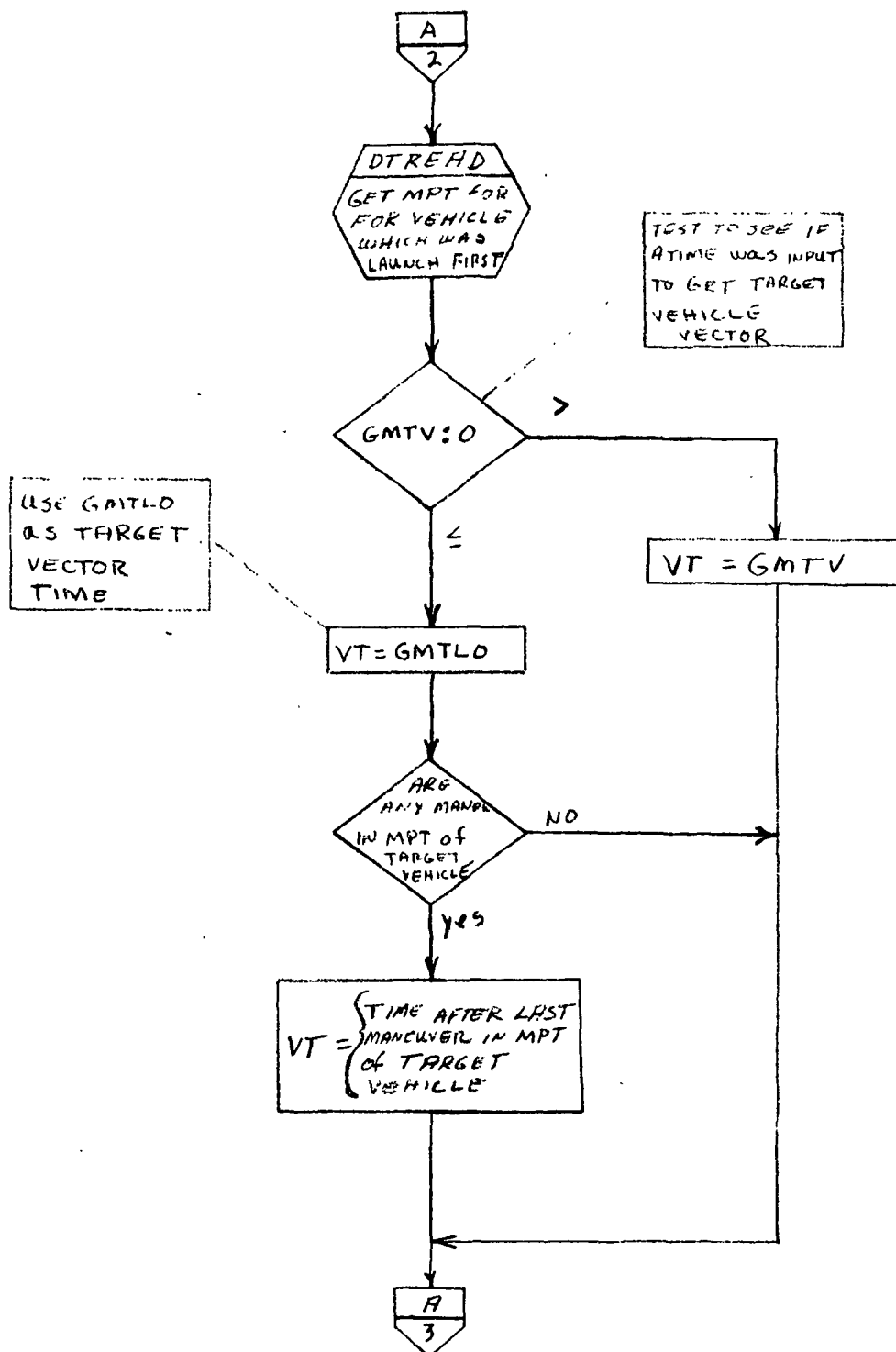


Figure PMMPAR-1. Flowchart

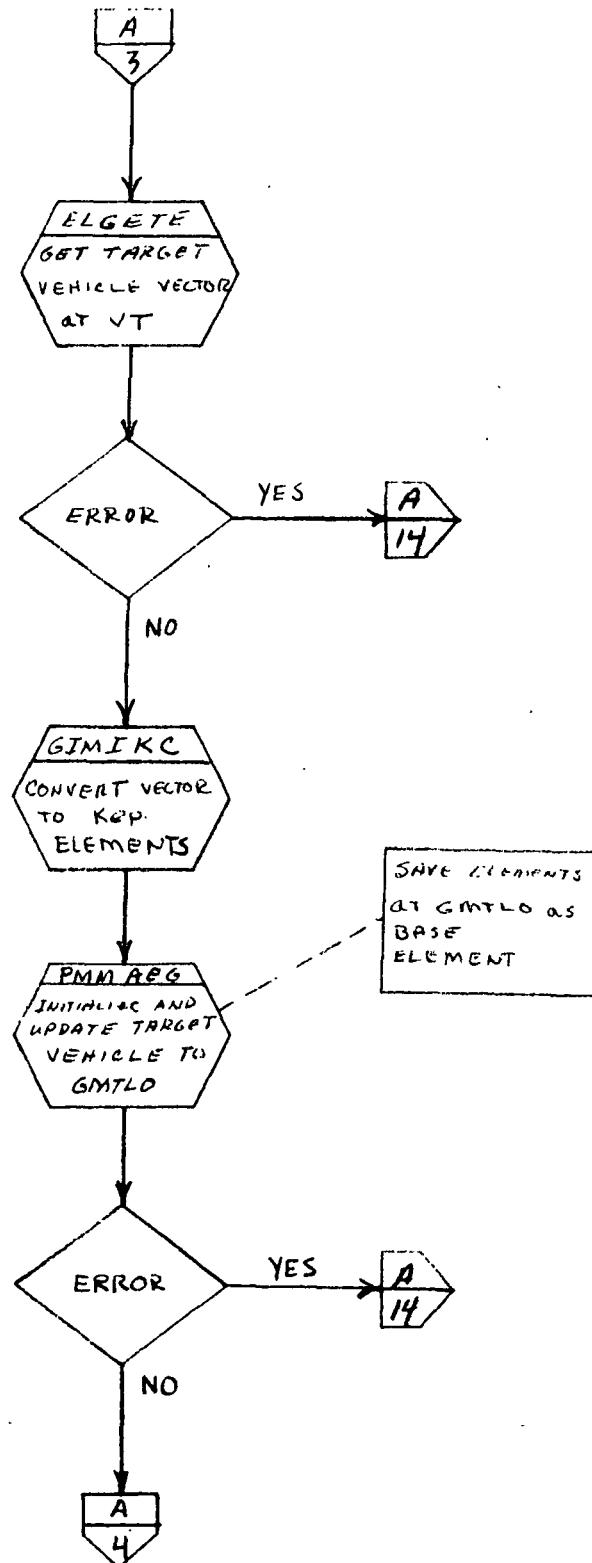


Figure PMMPAR-1. Flowchart

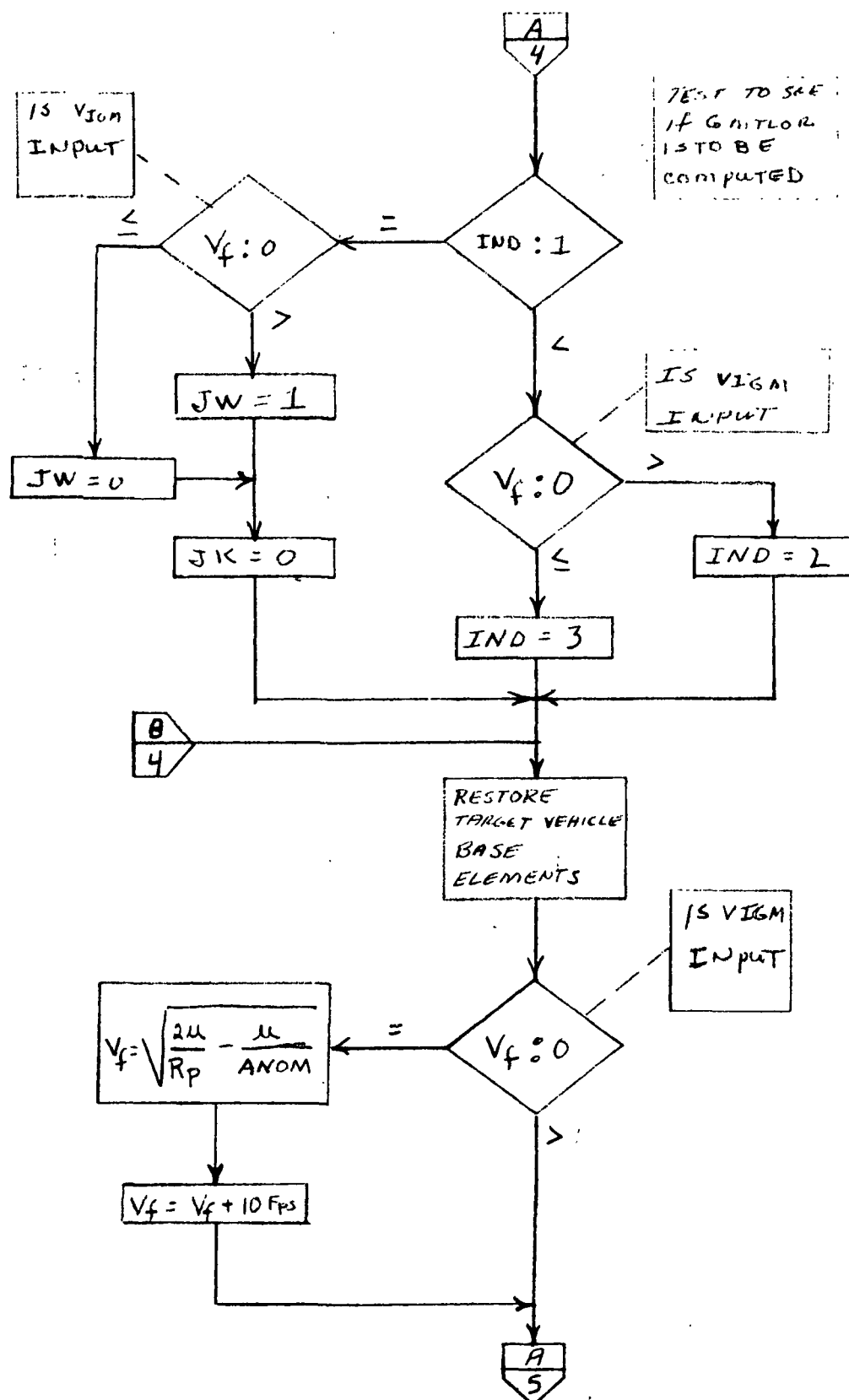


Figure PMMPAR-1. Flowchart

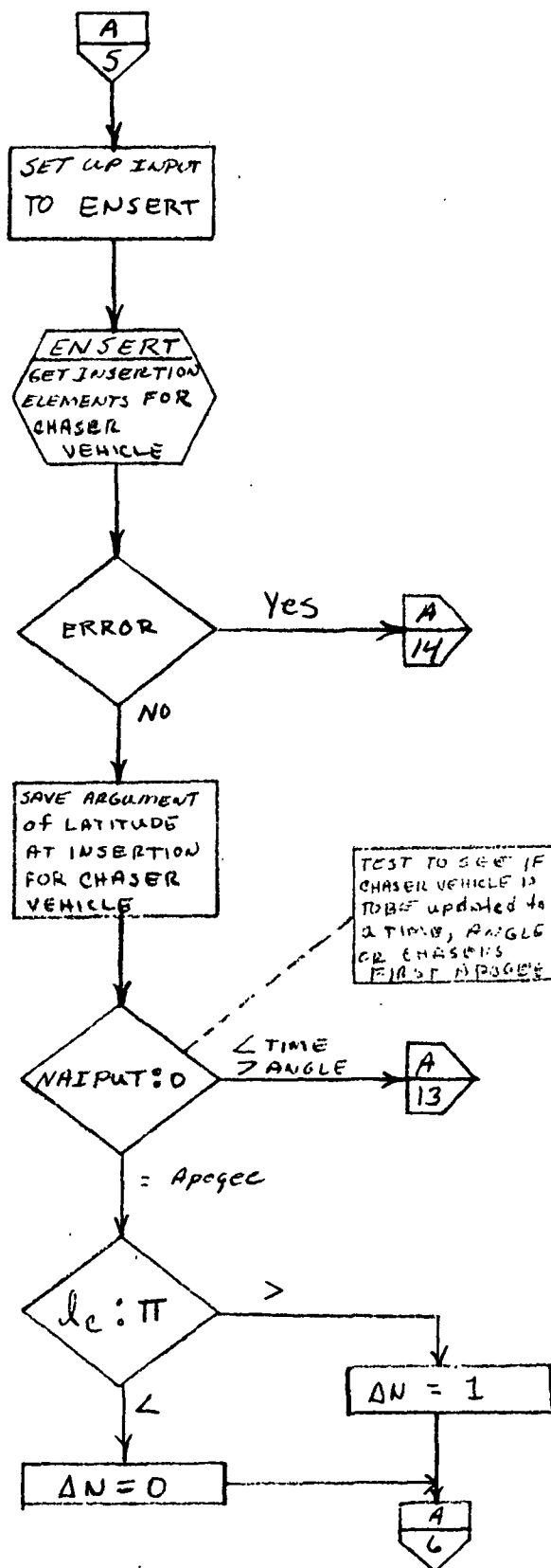


Figure PMMPAR-1. Flowchart

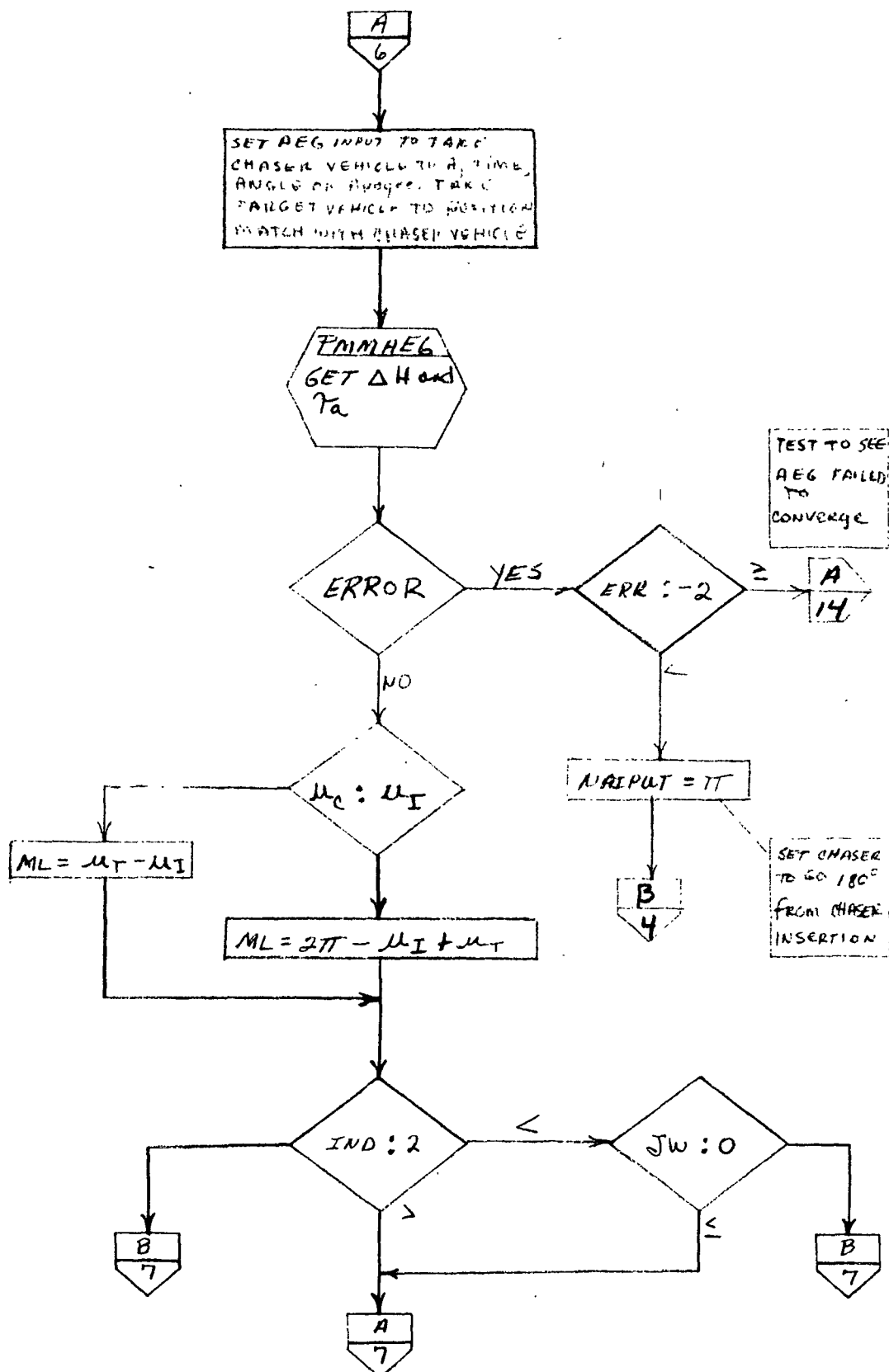


Figure PMMPAR-1. Flowchart

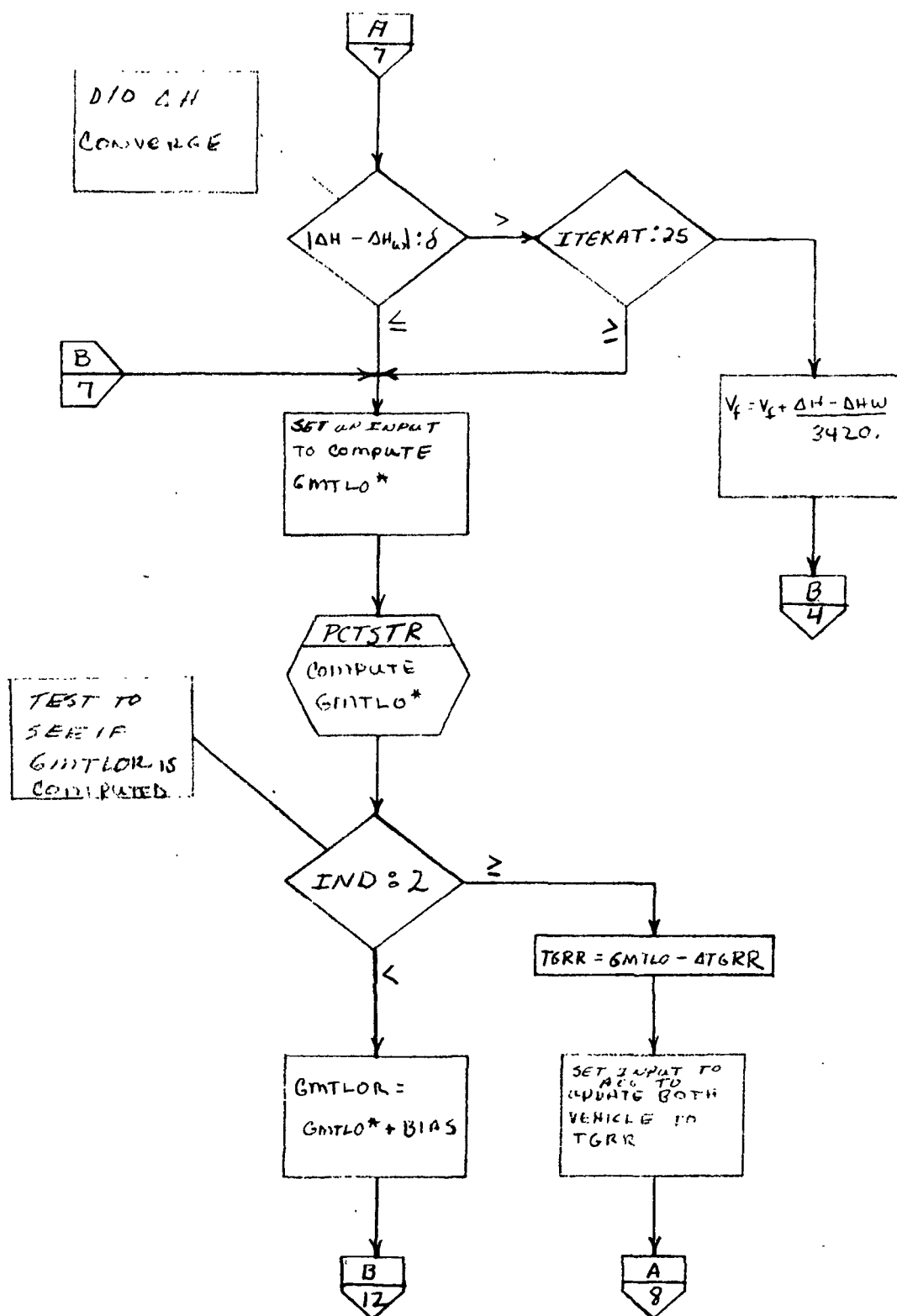


Figure PMMPAR-1. Flowchart

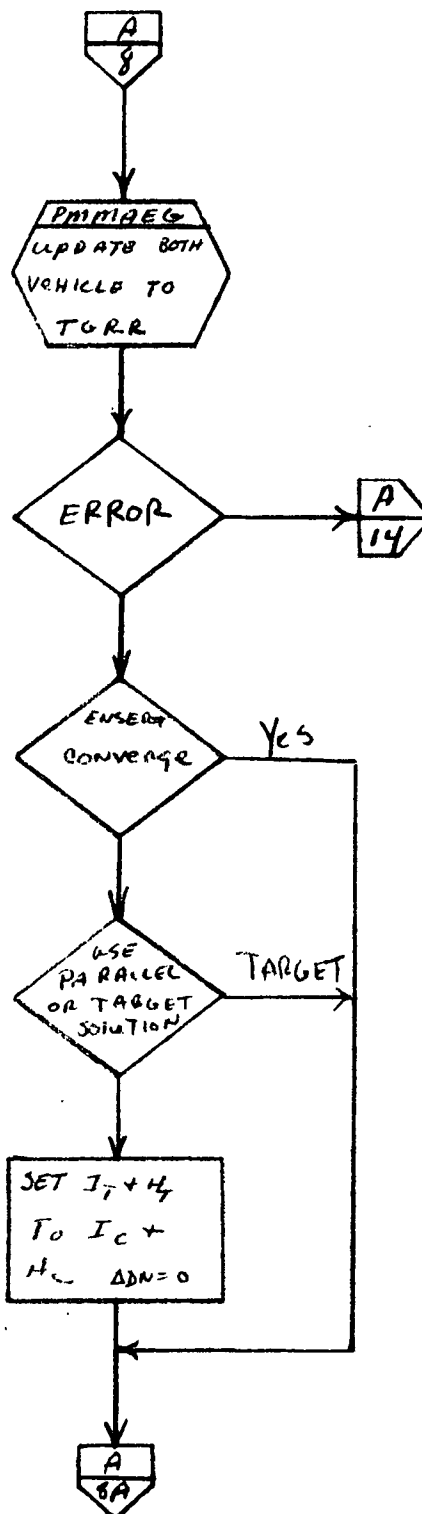


Figure PMMPAR-1. Flowchart

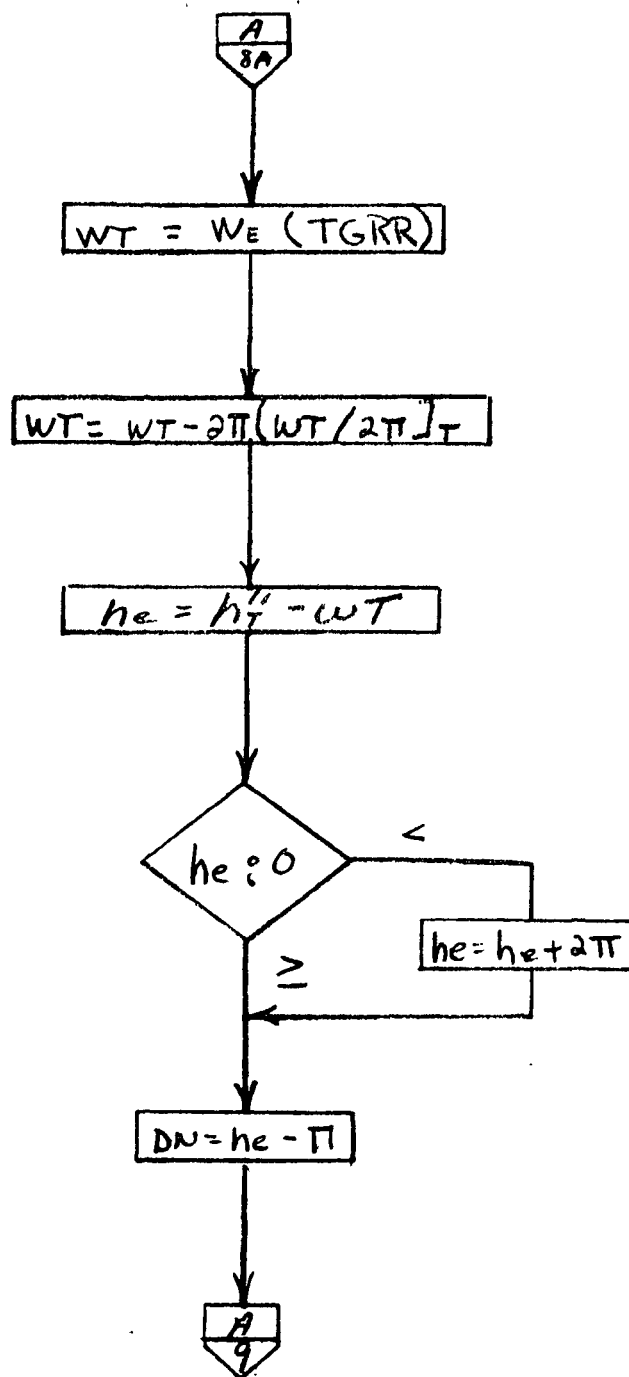


Figure PMMPAR-1. Flowchart

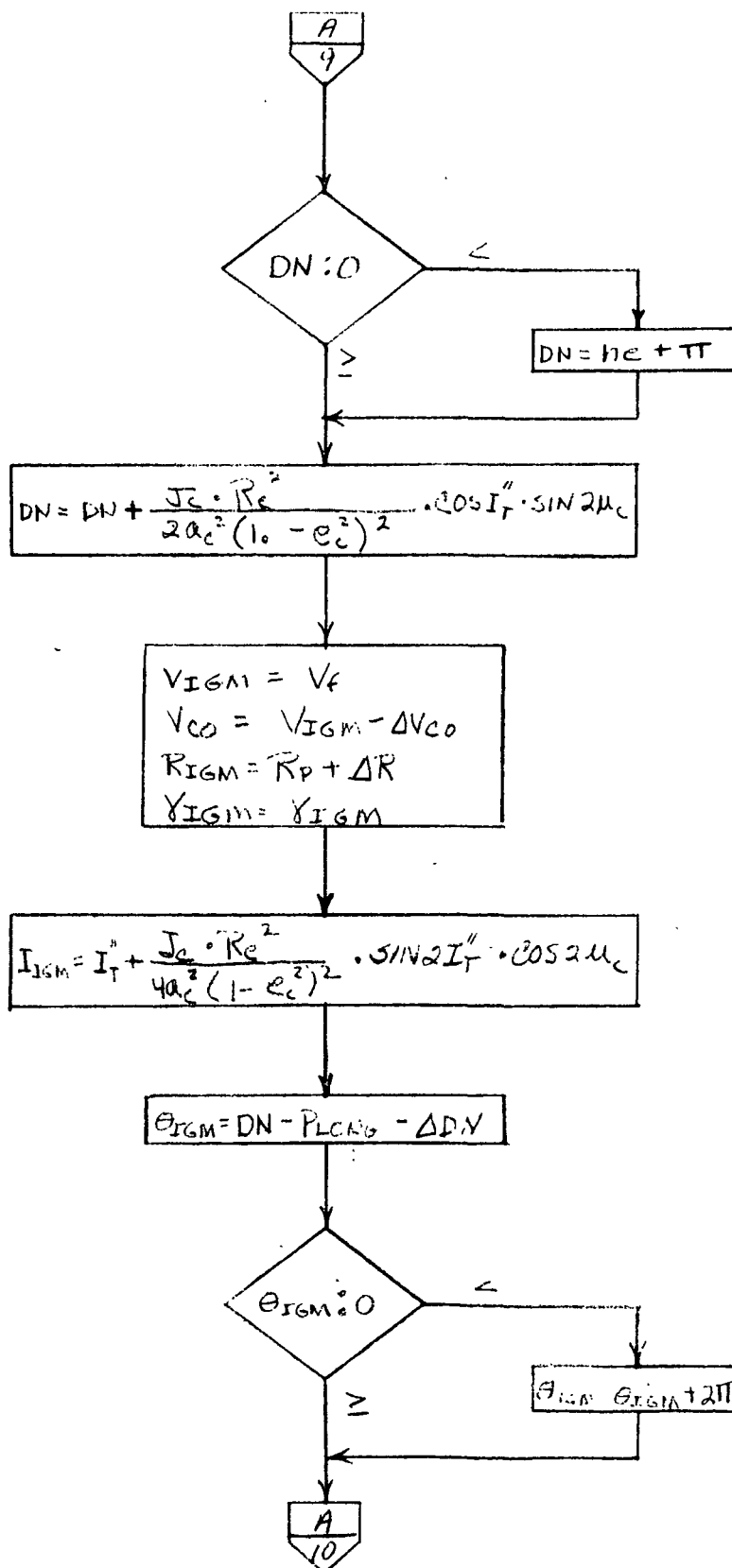


Figure PMMPAR-1. Flowchart

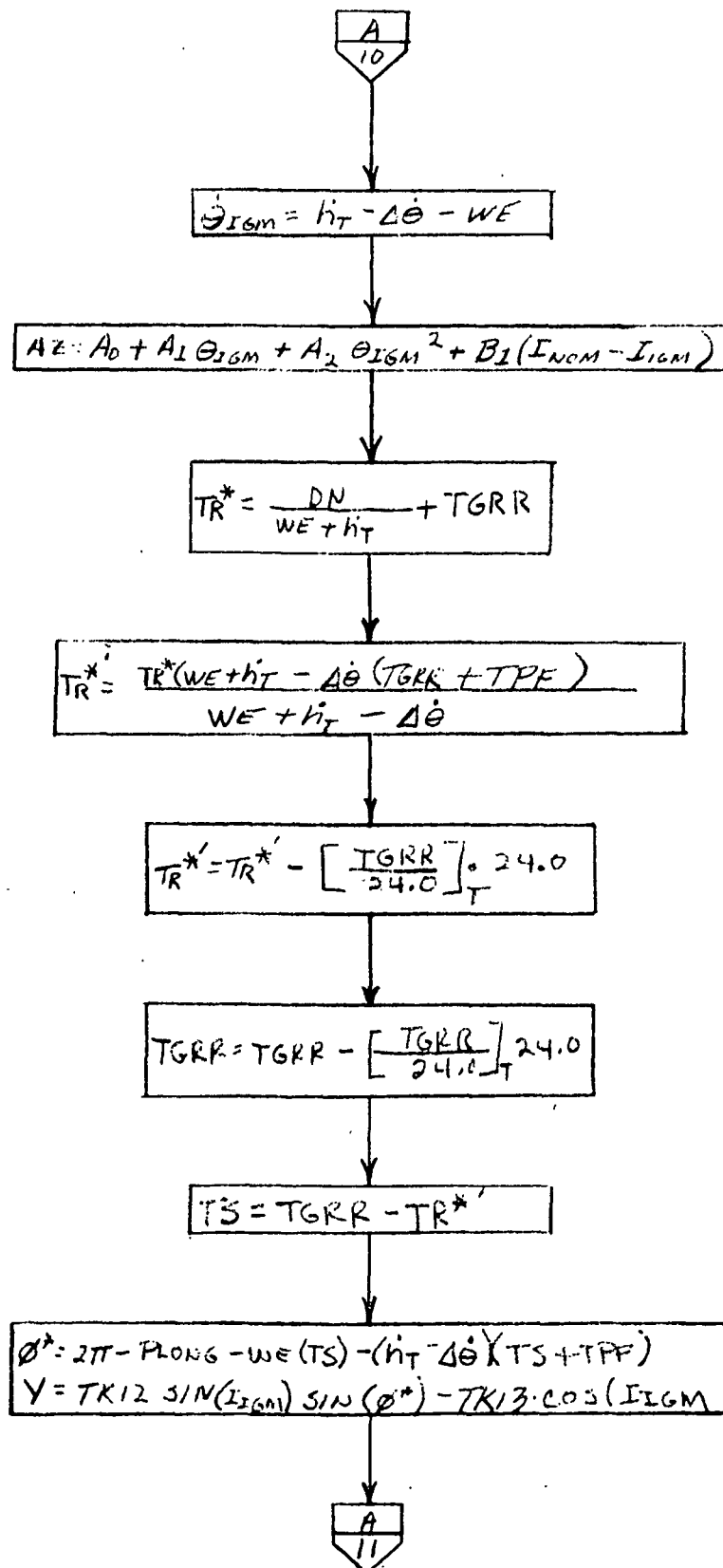


Figure PMMPAR-1. Flowchart

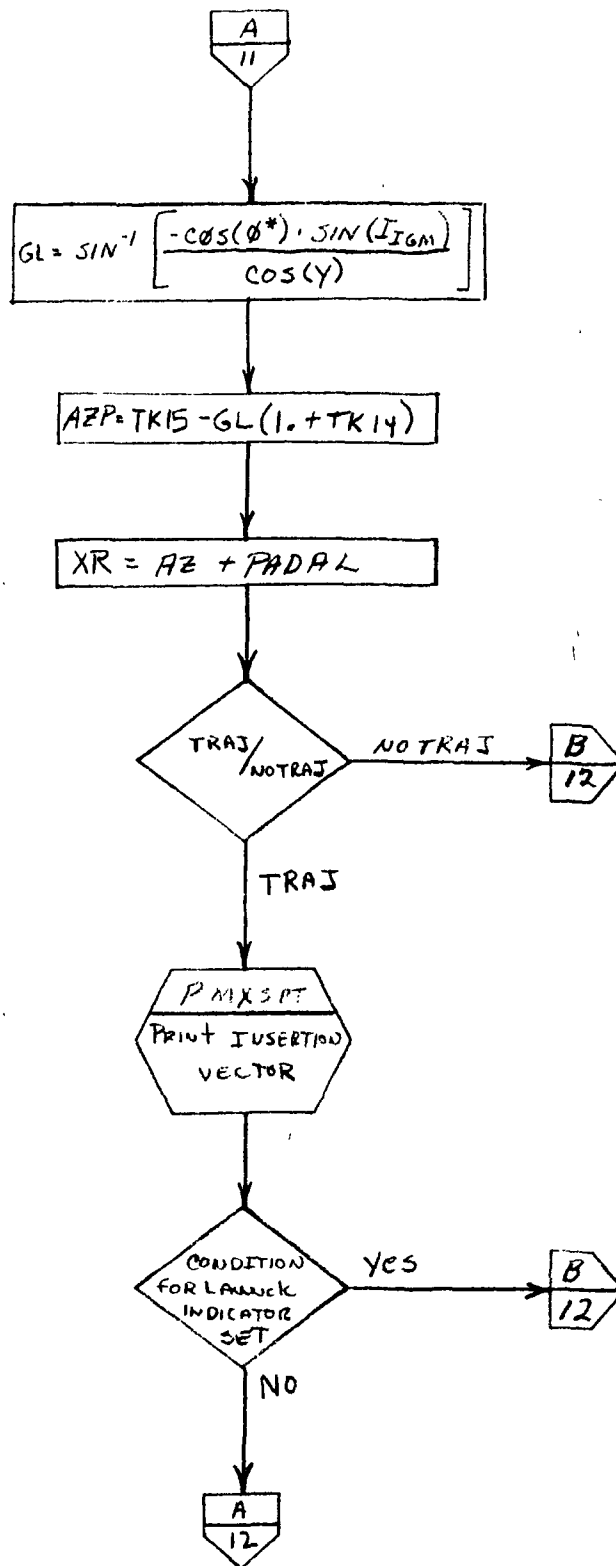


Figure PMMPAR-1. Flowchart

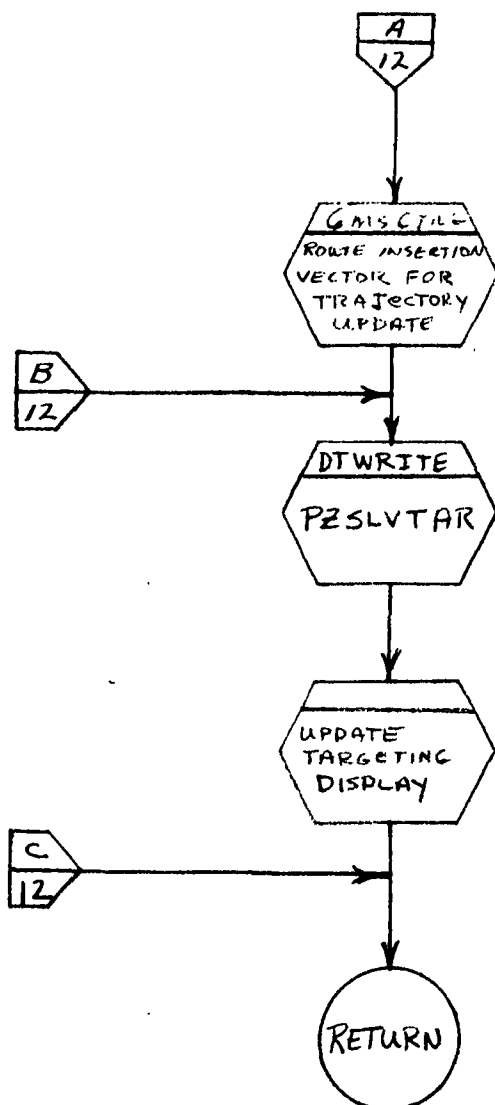


Figure PMMPAR-1. Flowchart

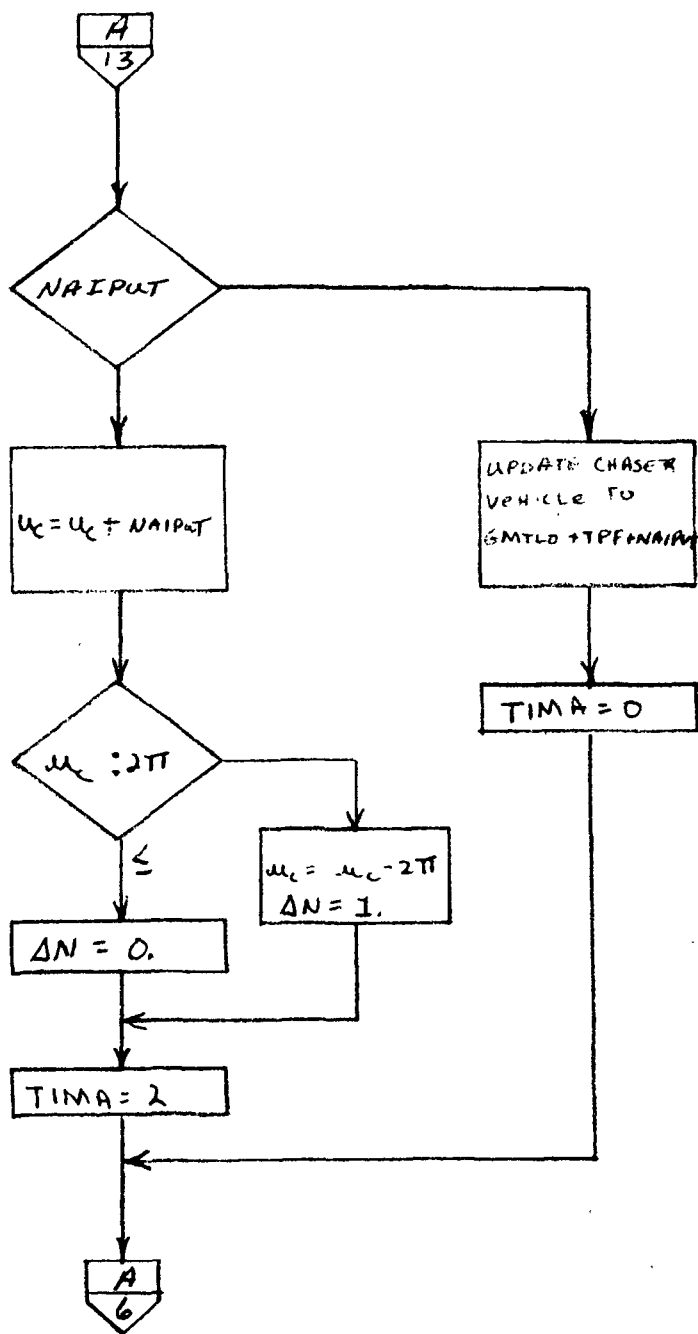


Figure PMMPAR-1. Flowchart

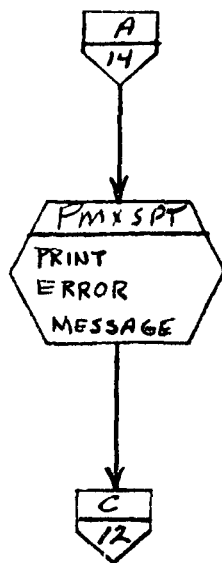


Figure PMMPAR-1. Flowchart

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

SUBROUTINE ENSERT

Subroutine to Compute an Insertion Vector

Input: Target vector (R_i, V_i) at insertion

λ_{LS} Launch site longitude
 ϕ_{LS} Launch site latitude
 t_{PF} Powered flight time
 θ_{PF} Powered flight arc
 $GMTLO$ Liftoff time
 V_i Insertion velocity
 α Flight path angle
 R_i Insertion radius
 R_L Radius of launch site
 Sys Yaw steer capability
 VS Yaw steer return code
 ΔDN Input bias

$TINS = GMTLO + t_{PF}$
 $SAVNODE = H(1)$
 $H(1) = H(1) - \Delta DN$
 Initialize AEG

$TINS = GMTLO + t_{PF}$
 $\lambda_I = \lambda_{LS} + \cos(GMTLO)$
 $R_p = R_L \cos \phi_{LS}$
 $R_{LSX} = R_p \cos \lambda_I$
 $R_{LSY} = R_p \sin \lambda_I$
 $R_{LSZ} = R_L \sin \phi_{LS}$
 $R_{LS} = R_{LSX} X + R_{LSY} Y + R_{LSZ} Z$

$$\bar{H} = \frac{\bar{R}_i \times \bar{V}_i}{|\bar{R}_i \times \bar{V}_i|}$$

$$\bar{K} = \frac{\bar{H} \times \bar{R}_{LS}}{|\bar{H} \times \bar{R}_{LS}|}$$

$$\bar{J} = \frac{\bar{R} \times \bar{H}}{|\bar{R} \times \bar{H}|}$$



Figure PMMPAR-1. Flowchart

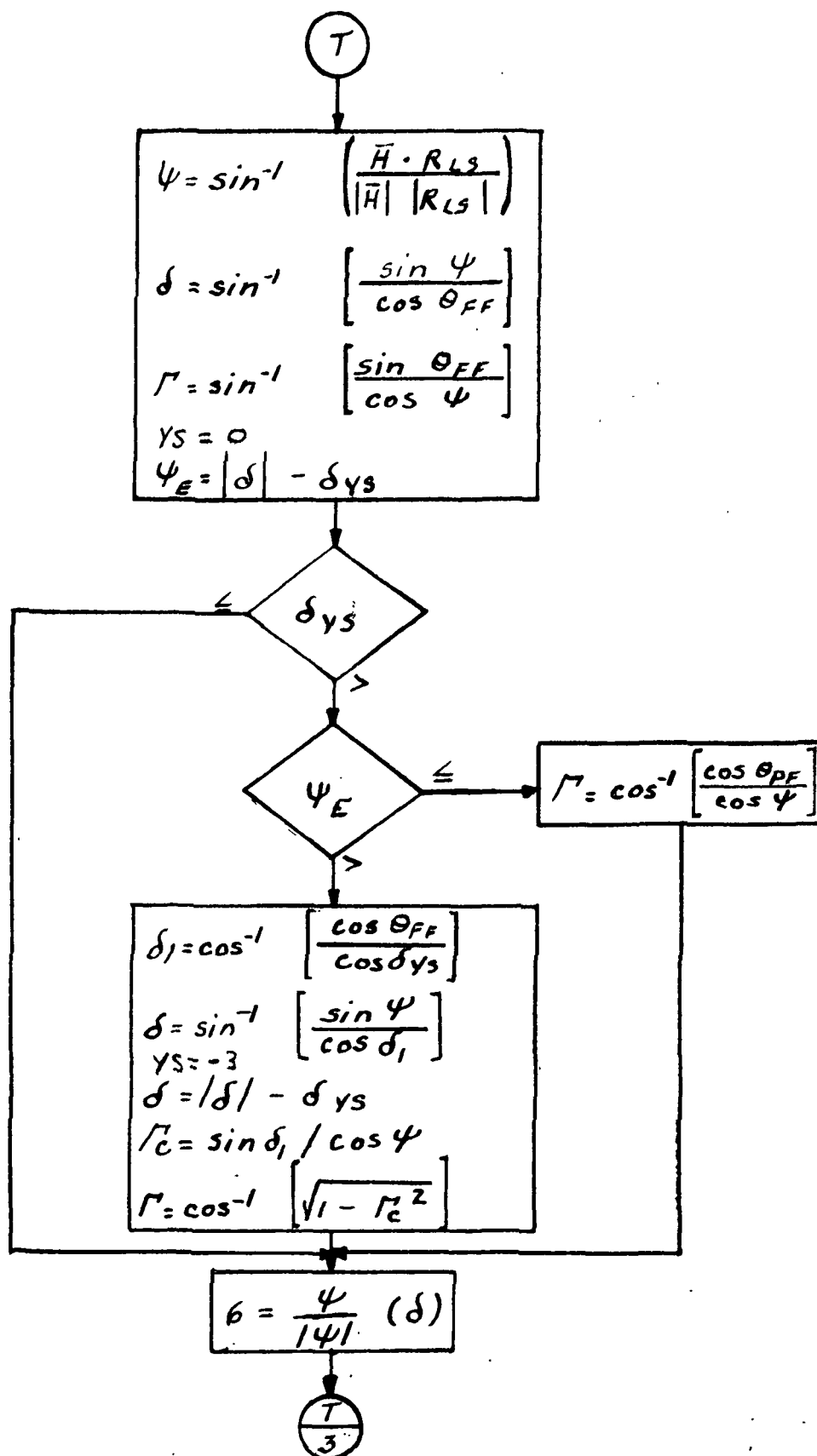


Figure PMMPAR-1. Flowchart

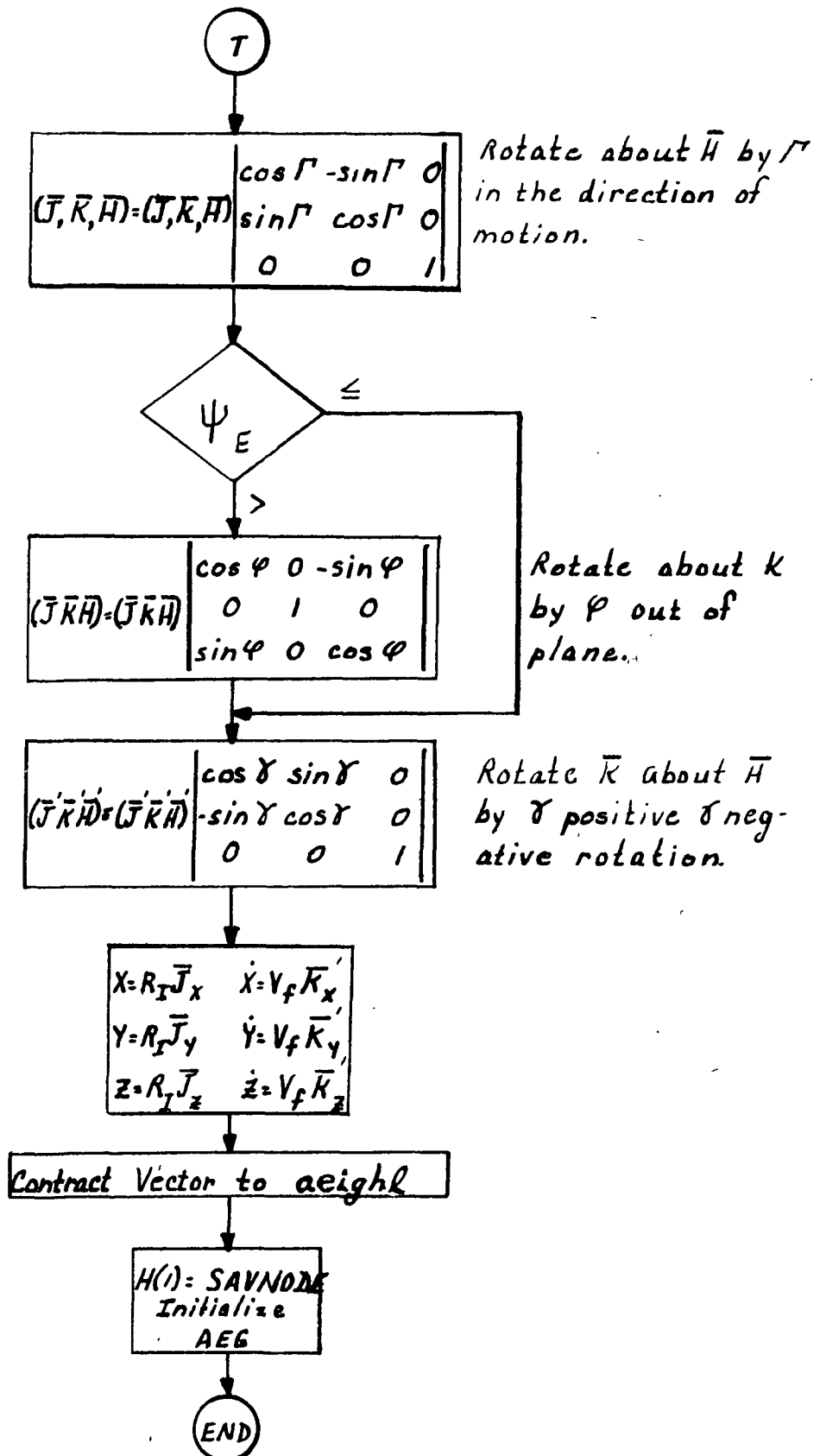


Figure PMMPAR-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMPAR
 DATE 11/21/67 (EORP)
 REV
 PAGE 23 of 26

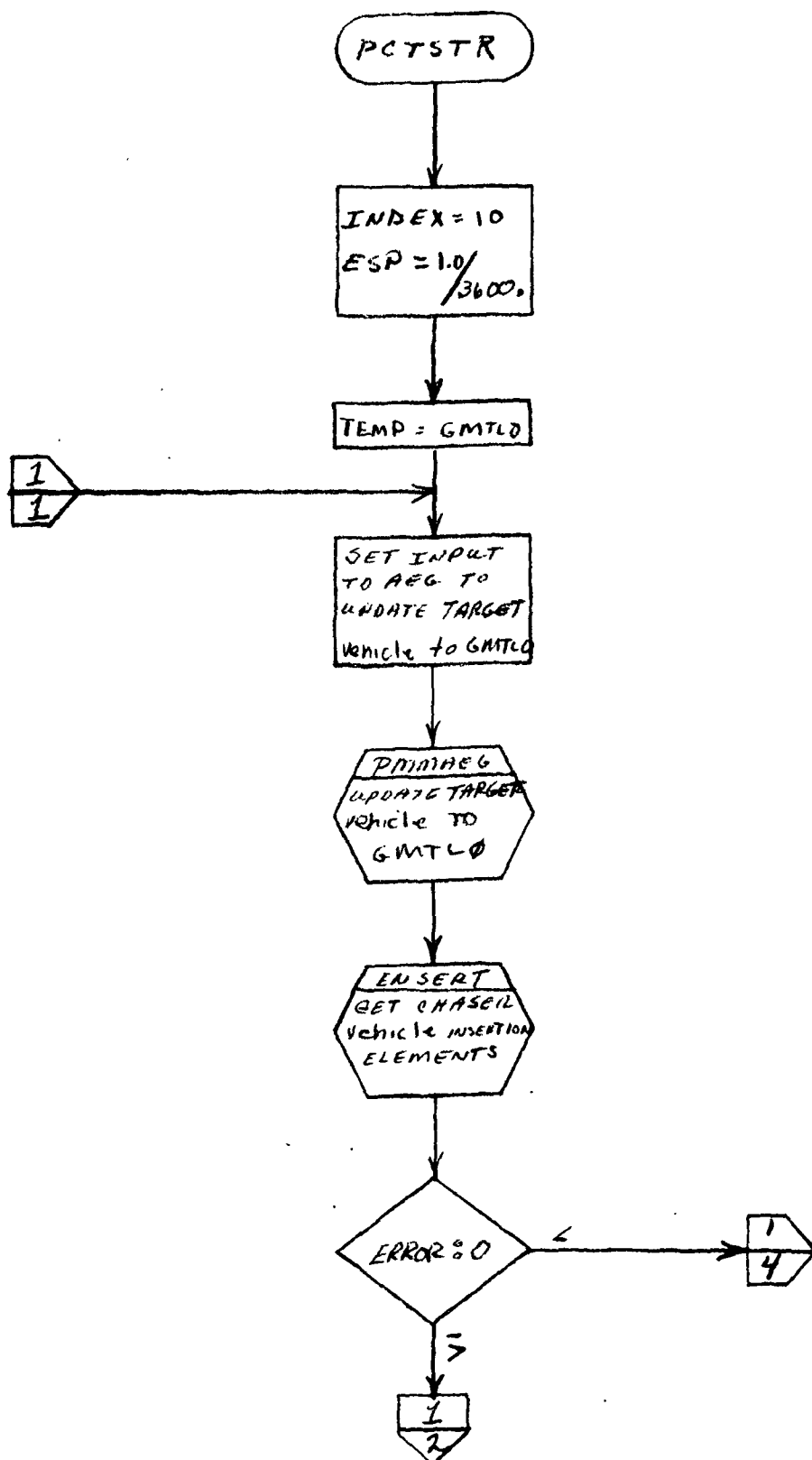


Figure PMMPAR-1. Flowchart

(Sheet 19 of 22)

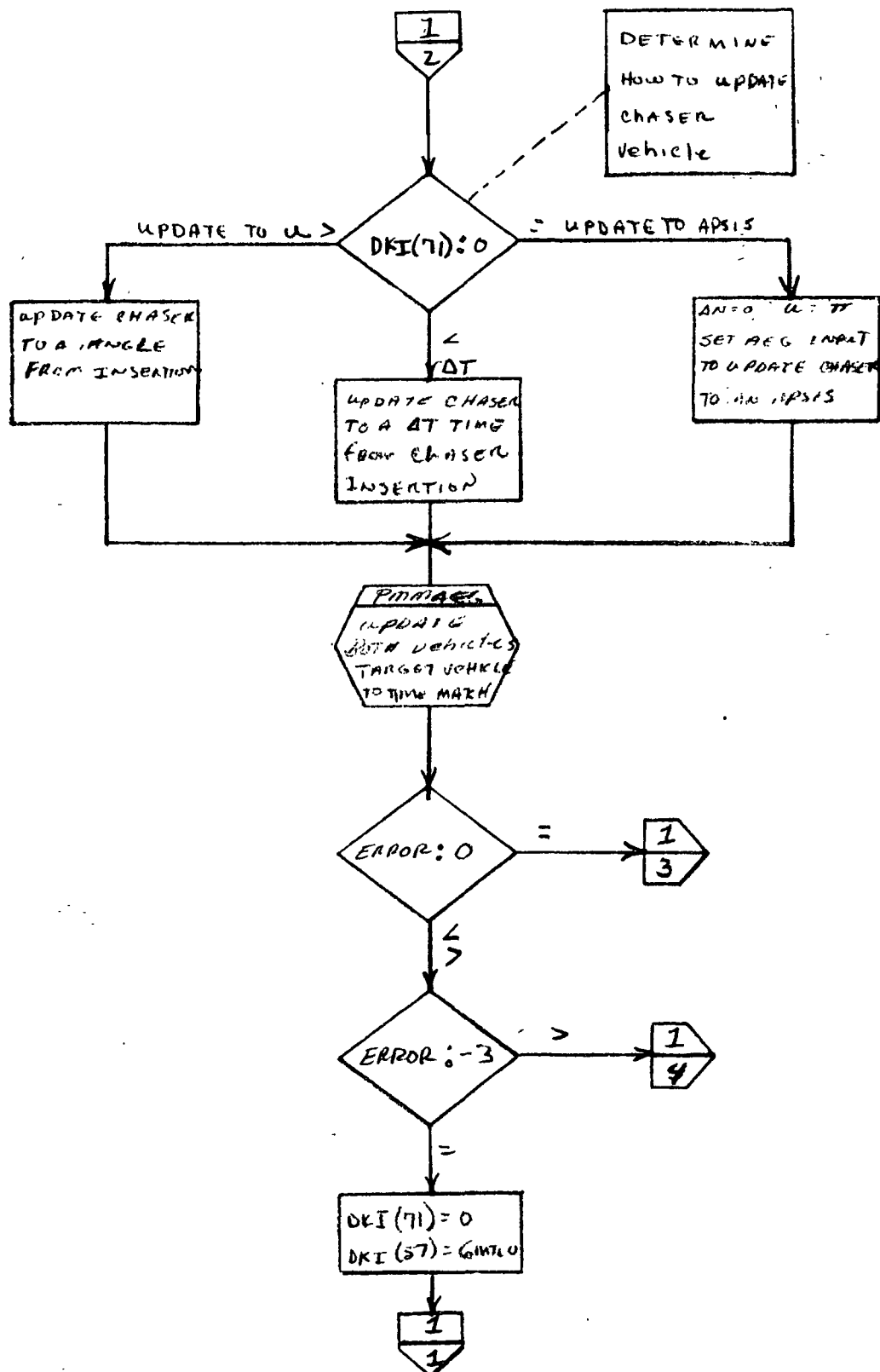


Figure PMMPAR-1. Flowchart

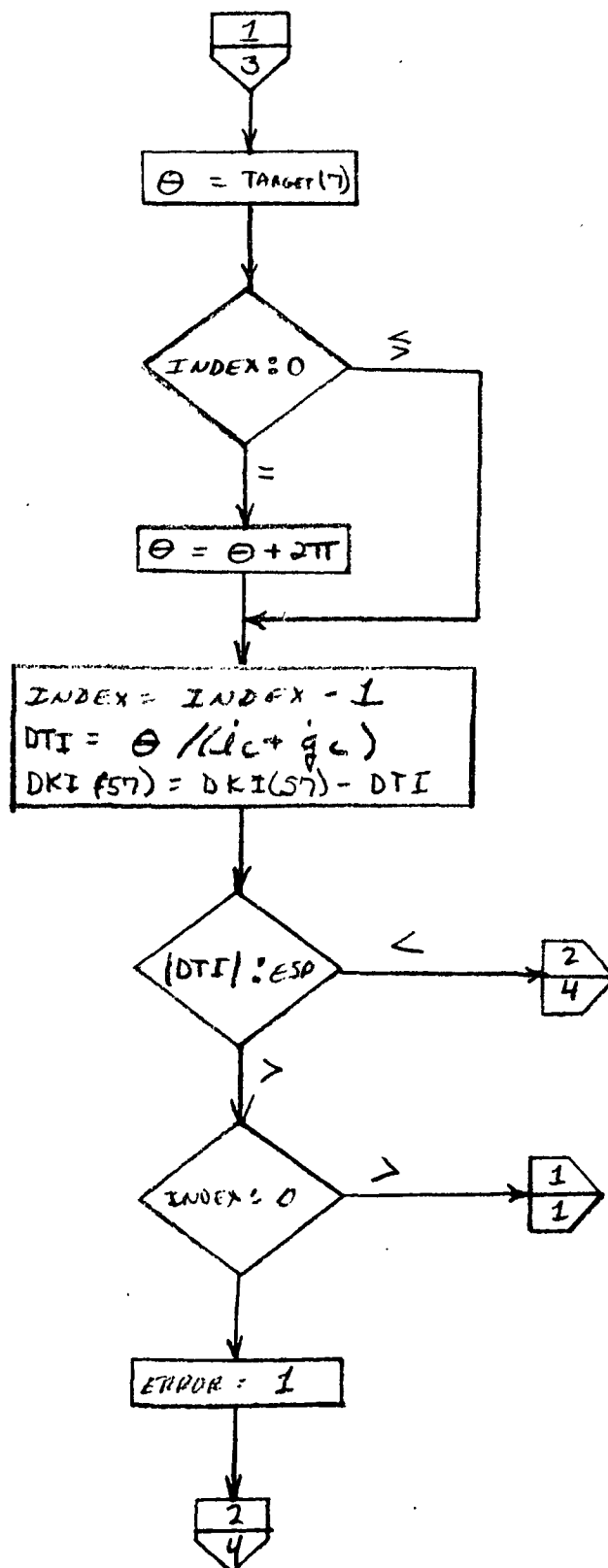


Figure PMMPAR-1. Flowchart

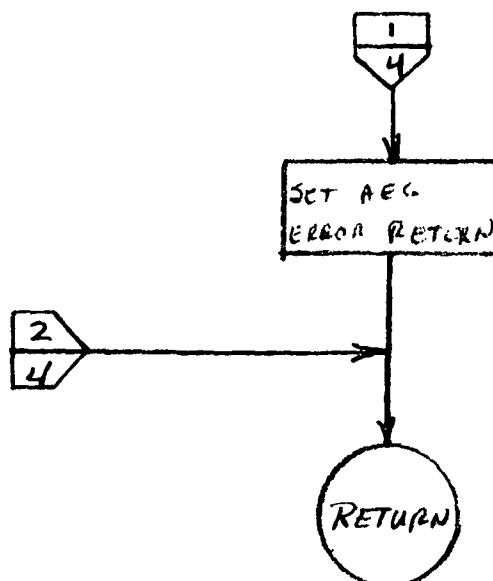


Figure PMMPAR-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMPHL

DATE 1/2/68 (EORF)

REV

PAGE 1 of 10

- a. Symbol — PMMPHL
- b. Program Name — Phaselag
- c. Programming Group — Rendezvous Planning
- d. Programmer(s) — M. R. Secondo
- e. Language — FORTRAN
- f. Data Tables Used — None
- g. MED's Used — None
- h. Writeup Completion Date — 11/16/67
- i. System Parameters — Table PMMPHL-1
- j. Flowchart — Figure PMMPHL-1

FUNCTION

This program will compute the phase angle needed at the circularization maneuver point to achieve rendezvous at a desired day/night condition or at a longitude crossing.

USAGE

- a. Calling Sequence — LINK EP=PMMPHL, PARAM=(DKI)
- b. Inputs — DKI is the common block used by PMMDKI
- c. Outputs and Exit — (1) Desired phase angle at NSR; (2) time of rendezvous; (3) transfer initiation time
- d. Other Programs Used — DSIN, DCOS, DSQRT, DATAN2, DARCOS, DTAN, PMMAEG
- e. Data Restrictions and Limits — None
- f. Error — Same as error returns for PMMAEG

METHOD

Upon entry to the program, a check is made to determine whether or not a height maneuver is scheduled. If no maneuver is scheduled, then the chaser vehicle is advanced to the next perigee point, and a differential height difference is found by taking the target vehicle to a position match. With an input desired height difference, this computation is not done.

By updating to the circularization maneuver point, NSR, and using the previously determined height difference, elements can be evaluated for the chaser vehicle, which would circularize the chaser vehicle with a definite height differential with respect to the target vehicle.

Approval	Approval
<i>M. R. Secondo</i>	<i>A. Arnett</i>

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMPhL
DATE 1/2/68
REV
PAGE 2 of 10

(EO

Letting the subscript "C" represent the chaser vehicle and "T" for the target vehicle, the semimajor axis is

$$a_C = a_T - \Delta h$$

and for the eccentricity,

$$e_C = a_T e_T / a_C.$$

The inclination angles and arguments of the ascending nodes would be equal.

$$i_C = i_T$$

$$h_C = h_T$$

To find the mean anomaly, l_C , and argument of perigee, g_C it is necessary to find the true anomaly.

$$(e \cos f)_C = [a_C(1 - e_C^2) - R_C] / R_C \quad \text{where } R_C = R_T - \Delta h$$

$$(e \sin f)_C = (\text{sign of } \sin f_T) [e_C^2 - (e \cos f)_C^2]$$

which gives

$$f_C = \tan^{-1} \left[\frac{(e \sin f)_C}{(e \cos f)_C} \right]$$

and

$$l_C = f_C - 2(e \sin f)_C [1 - 3/4(e \cos f)_C].$$

Since $u_C = u_T$, then the argument of perigee is

$$g_C = u_C - f_C.$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMPHL

DATE 1/2/68

(EORP)

REV

PAGE 3 of 10

With these elements computed, the target vehicle is taken to T' where

$$T' = T_{NSR} + 2\pi(\Delta NSR - 1)/(\dot{i} + \dot{g})_T.$$

This will place the vehicle within one maneuver line of the rendezvous point.

Using either the day/night option or the longitude crossing routine, the time of rendezvous can be found. From this, the transfer initiation time can be calculated using the terminal phase travel angle.

$$T_I = T_R - wt/\dot{i}_T$$

If there is not enough time between T_I and the $(M - 1)$ maneuver line, then the time of rendezvous is recomputed in the next orbit, as well as T_I .

After taking the target vehicle to T_I , the desired phase angle, θ_{wT} , is computed.

$$R_C = R_T - \Delta h$$

$$C_1 = - \left[\frac{R_C}{\cos E} \right]^2$$

$$C_2 = 2R_T R_C$$

$$\theta_{wT} = \cos^{-1} \left[\frac{-C_2 - \sqrt{C_2^2 + 4C_1(C_1 + R_C^2 + R_T^2)}}{2C_1} \right]$$

With this desired phase angle, the time increment from NSR to transfer initiation can be computed by finding the argument of latitude and mean anomaly increments.

$$\Delta u_C = u_T - \theta_{wT} - u_C$$

$$\Delta l_C = \Delta u_C - e_C [2 \sin(f_C + \Delta u_C) - 2 \sin(f_C) - 3/4 e_C (\sin(2f_C + 2\Delta u_C) - \sin(2f_C))]]$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMPHL

DATE 1/2/68

REV

PAGE 4 of 10

(EC

which gives

$$\Delta t_C = [\Delta l_C + 2\pi(\Delta NSR - 1)]/\eta_C$$

where η_C is the mean motion of the chaser vehicle.

$$\eta_C = \sqrt{\mu/a_C^3}$$

Therefore the NSR time, which is assigned to the chaser vehicle, is

$$T_{NSR} = T_I - \Delta t_C$$

Using the AEG, both vehicles are taken to the transfer initiation time and the resulting phase angle, θ_T , is used as a correction to the computed desired phase angle, θ_{wT} , in determining θ_{NSR} .

$$\theta_{NSR} = \theta_{wT} + (T_I - T_{NSR})(\dot{l}_C - \dot{l}_T) + (\theta_{wT} - \theta_T)$$

The computed quantities, θ_{NSR} , T_I and T_{NSR} are returned as output to the calling program.

*Type
H — Fixed-point half word
F — Fixed-point full word
E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

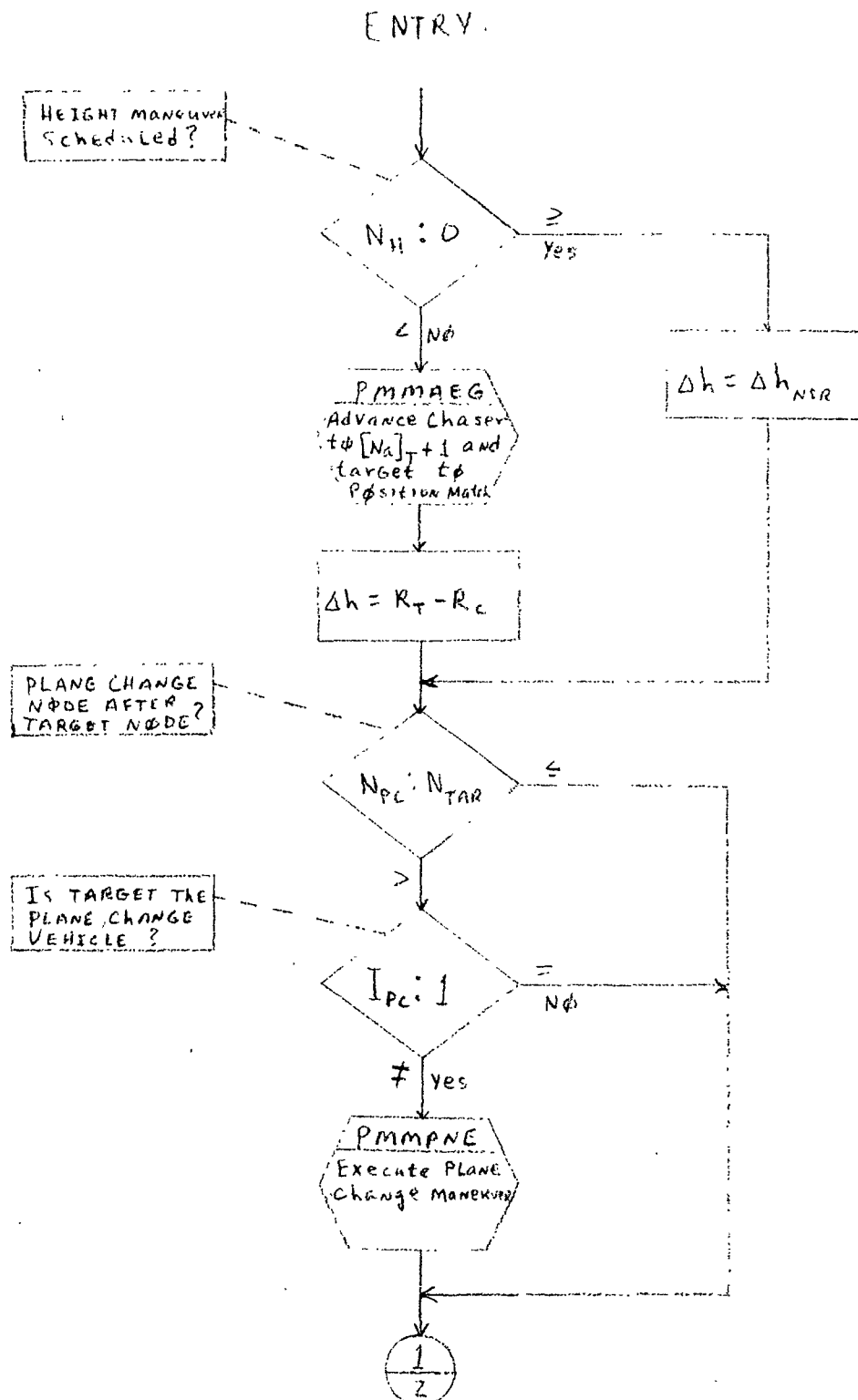


Figure PMMPHL-1. Flowchart

(Sheet 1 of 5)

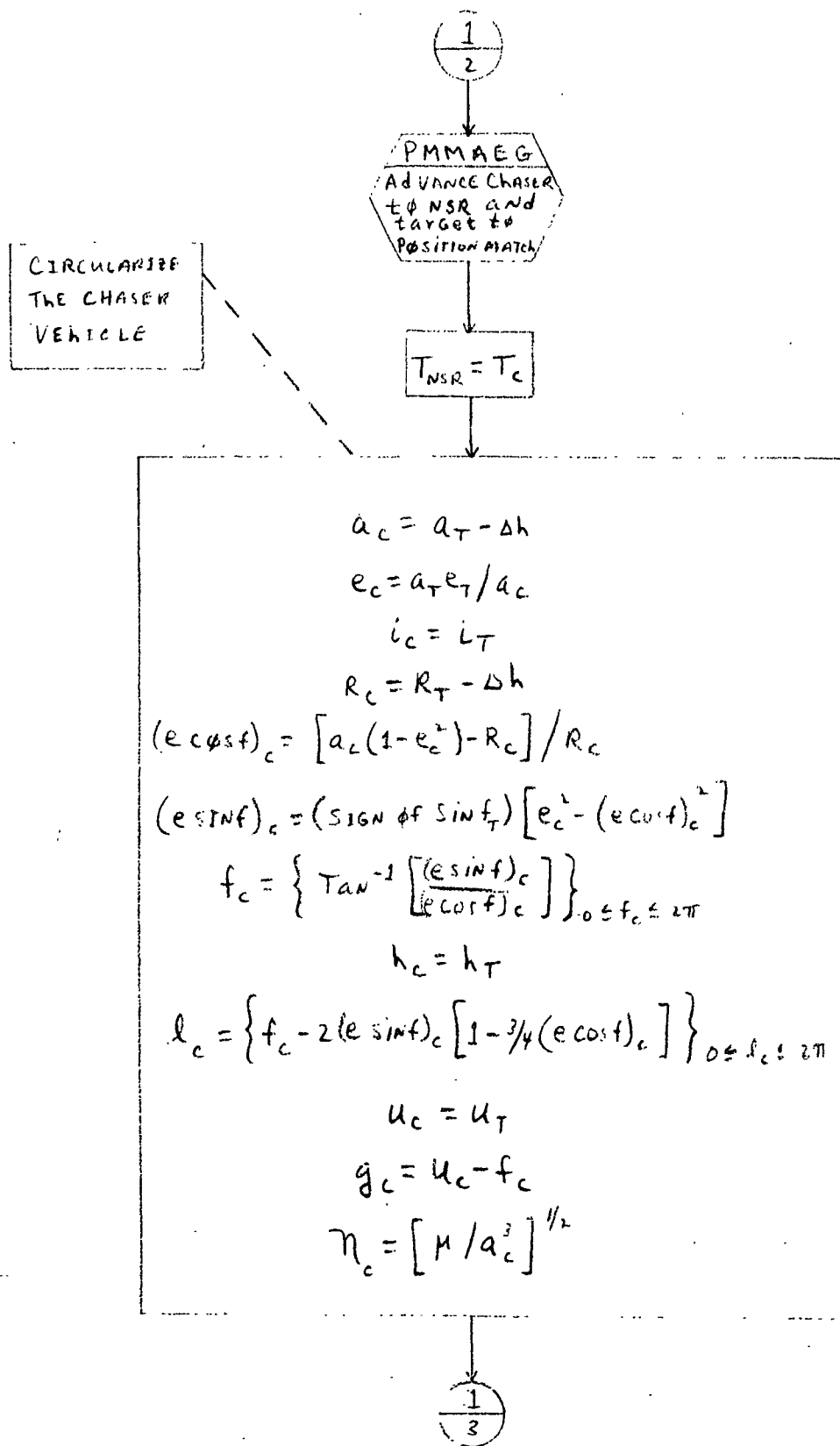


Figure PMMPHL-1. Flowchart

(Sheet 7 of 5)

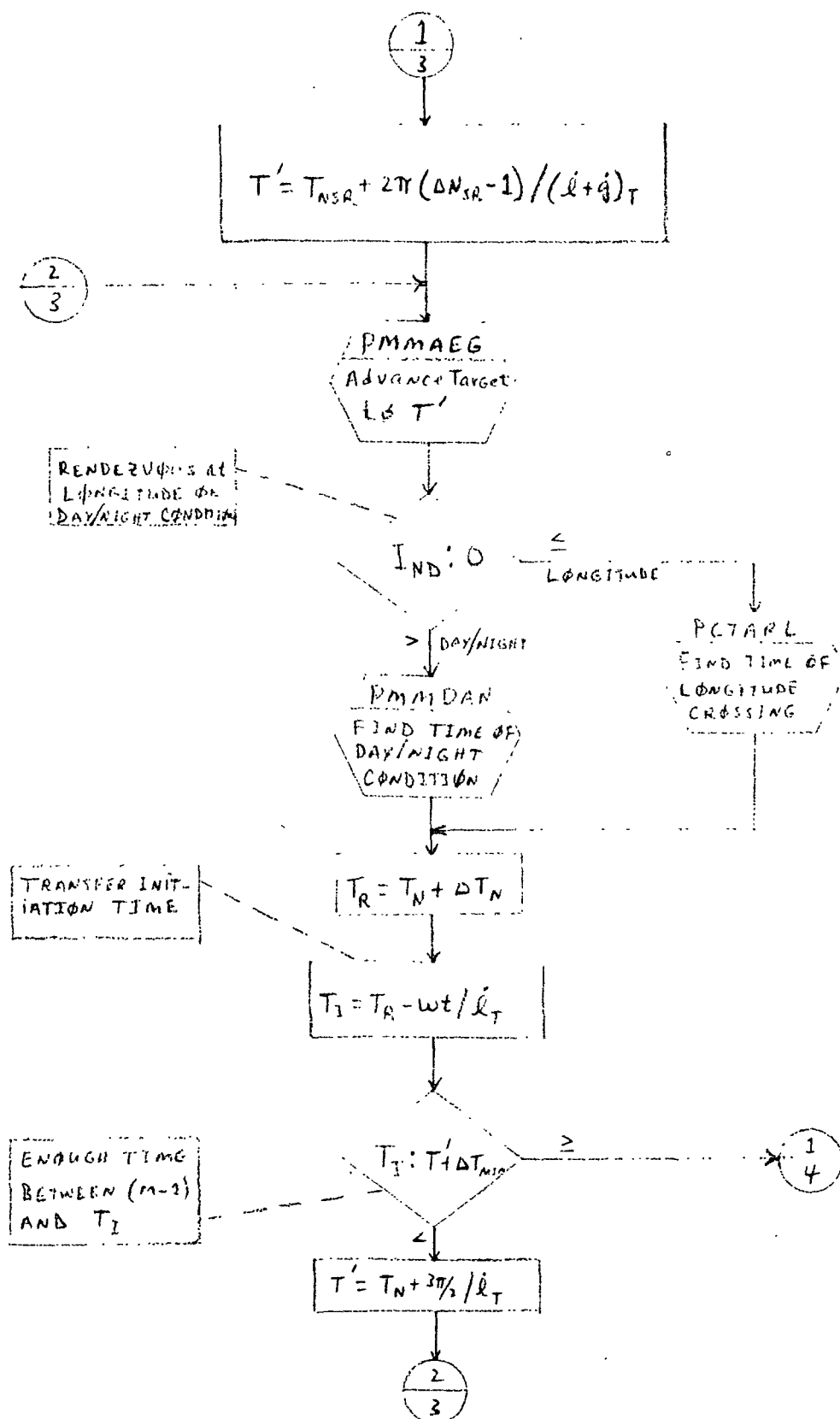


Figure PMMPL-1. Flowchart

(Sheet 3 of 5)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMPHL

DATE 1/2/68

(EORP)

REV

PAGE 9 of 10

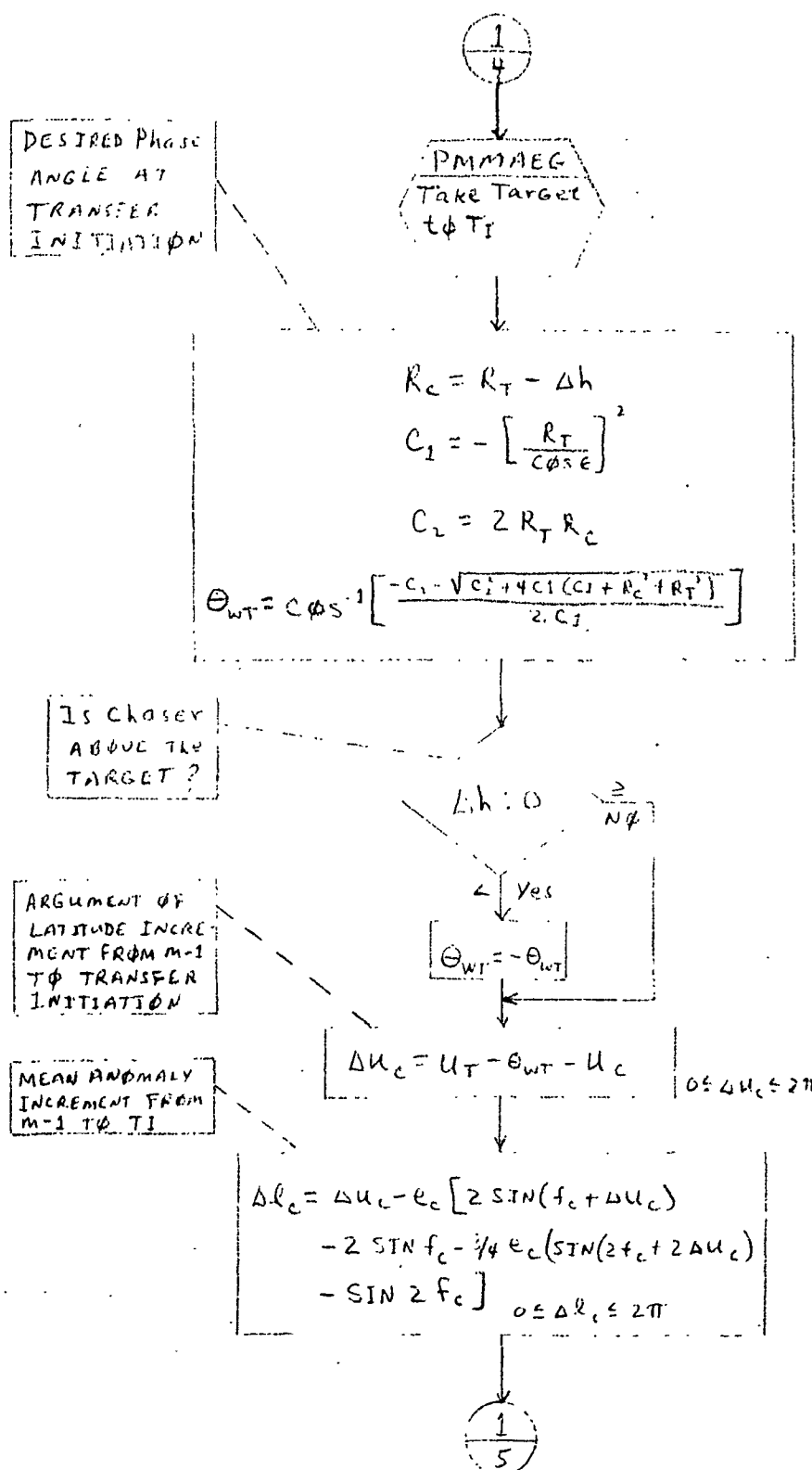


Figure PMMPHL-1. Flowchart

(Sheet 4 of 5)

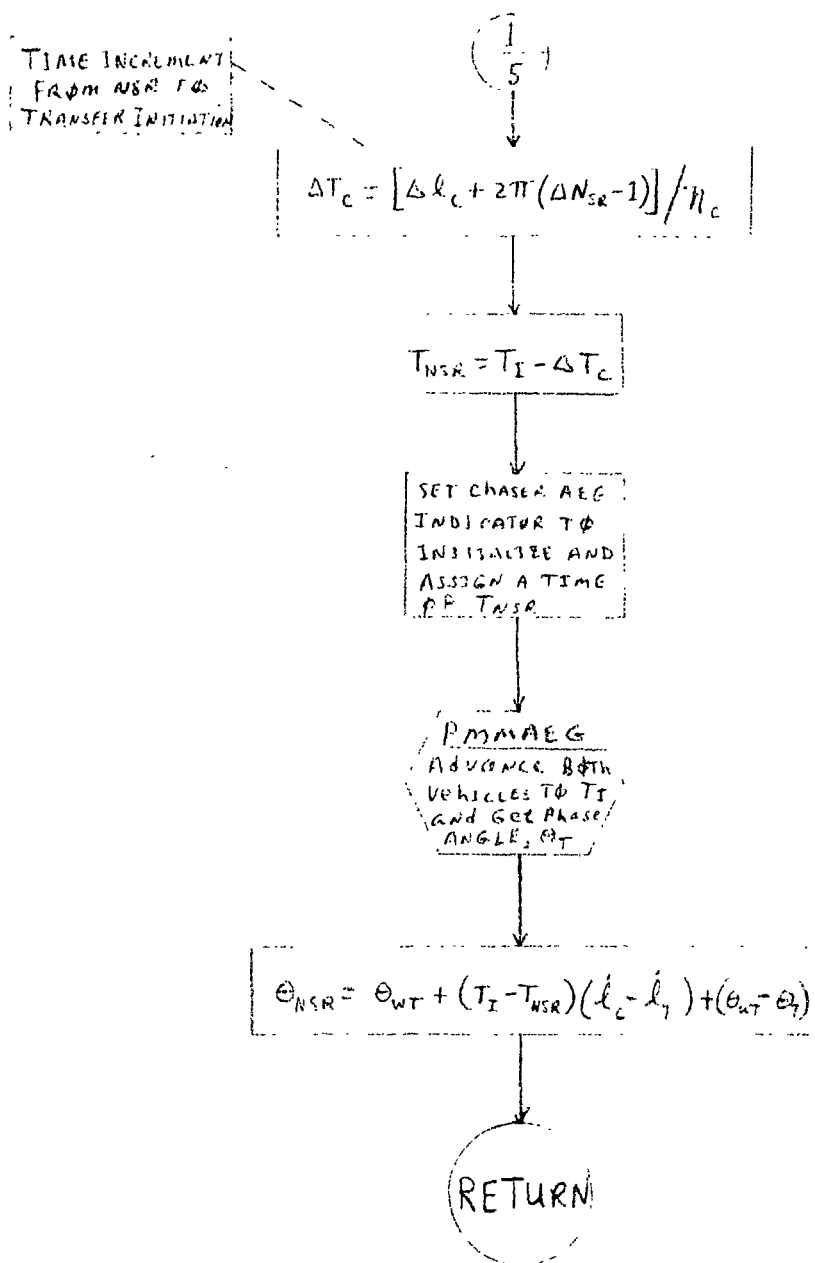


Figure PMMPHL-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMPNE
 DATE 1/2/68 (EORP)
 REV
 PAGE 1 of 13

- a. Symbol — PMMPNE
- b. Program Name — Plane Change Time and Velocity Increments Computations
- c. Programming Group — Rendezvous Planning (Mission Planning)
- d. Programmer(s) — G. Deaton, M. Secondo
- e. Language — FORTRAN
- f. Data Tables Used — None
- g. MED's Used — None
- h. Writeup Completion Date — 11/16/67
- i. System Parameters — Table PMMPNE-1
- j. Flowchart — Figure PMMPNE-1

FUNCTION

Chaser and target vehicle osculating elements, reference time, and node number are input. The plane change maneuver is computed at the input node number beyond reference time. The program computes the time of the plane change maneuver and the out-of-plane and horizontal velocity increments, which are applied to the chaser at the given node number. This application achieves a trajectory that is coplanar with the target trajectory at phase match.

USAGE

- a. Calling Sequence — LINK EP = PMMPNE PARAM = (TAB, TOPTREF, FNPC, KPC, HIPC)
- b. Inputs —
 - 1. TAB — Fifty-five doubleword array, AEG header, chaser and target blocks in standard initialized format.
 - 2. TREF — Five doubleword array consisting of a boundary time and $\sqrt{u}$ as input. The last three doublewords are output values: actual plane change time, horizontal and out-of-plane velocity components.
 - 3. FNPC — Forty-six doublewords of which only the first word is input and is the node number of the plane change. The other forty-five doublewords are used internally.
 - 4. KPC — A word that indicates the entry desired. See "METHOD".
 - 5. HIPC — A four halfword array in which the second halfword indicates the vehicle performing the maneuver: +1 for the chaser, -1 for the target. The other halfwords are used internally.
- c. Outputs and Exit —
 - 1. TOP — Fifty-five double words containing the AEG header and both vehicle's AEG blocks at the plane change point (before maneuver).
 - 2. TREF — The last three doublewords of this five doubleword array are referenced in the description of inputs.

Approval	Approval
<i>M. R. Secondo</i>	<i>R. Amietti</i>

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK:MISSION SYSTEMS - AS-200 Mission Planning

2. PMMPNE
DATE 1/2/68 (EO
REV
PAGE 2 of 13

- d. Other Programs Used — PMMAEG, GIMIKC, DSQRT, DATAN, DATAN2, DTAN; DARCOS
- e. Data Restrictions and Limits — None
- f. Errors — None

METHOD

The plane change subroutine is designed to be used to compute the out-of-plane and the horizontal velocity increments, which are applied by a chaser vehicle at a specified common node. This application will achieve a coplanar orbit with a target vehicle at the point where the phase angle becomes zero. The program can be used in conjunction with an iterative rendezvous program, such as the Coelliptic Rendezvous Program.

An input flag, KPC, determines the program requesting the plane change computations. For a coelliptic request, $KPC < 0$; for a DKI request, $KPC > 0$.

There are three entries to the program. For the first entry ($KPC = \pm 1$), a lower boundary time, T_B , for the plane change maneuver is computed in terms of the input reference time and the number of common nodes beyond the reference time at which the plane change maneuver ΔV 's are desired. The chaser vehicle is updated to the boundary time and the target to position match using the Analytic Ephemeris Generator (AEG). The common node for the plane change is constrained to be between the chaser vehicle's argument of latitude, U_L , at T_B and $U_L + \pi$.

At this point, the period, inclination, and ascending node differentials are computed. If the period differential magnitude is greater than one second, the ascending node differential is adjusted to account for the differential nodal, regression, which will occur until phase match takes place.

It is now possible to form a "phantom" vehicle whose semimajor axis, eccentricity, and mean anomaly are the same as those of the chaser and whose inclination and longitude of the ascending node are the same as those of the chaser plus the computed differentials. The phantom orbit argument of perigee is obtained by reflecting the chaser argument of perigee into the phantom plane.

The chaser and phantom vehicles are advanced to their common node by recursively updating the chaser to the computed common node argument of latitude and the phantom vehicle to position match using the AEG. This is done three times, after which the phantom vehicle osculating elements are considered to be those needed by the chaser vehicle after the plane change maneuver.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMPNE
DATE 1/2/68 (EORP)
REV
PAGE 3 of 13

The chaser and phantom elements are converted to inertial position and velocity vectors. The vectors, along with the plane change velocity magnitude required, are manipulated to get the out-of-plane and horizontal ΔV components needed to apply the maneuver. The chaser elements and time at the plane change maneuver common node are output so that the user can apply the maneuver if he wishes.

If the plane change program is being used by a program that produces in-plane rendezvous maneuvers, it may be that the predicted point at which coplanar trajectories are required is erroneous because of in-plane maneuvers that follow the plane change maneuver. To compute a reliable plane change maneuver under these circumstances, a second entry ($KPC=\pm 2$) to the plane change program should be made. This entry should be made if there is a wedge angle violation in the vicinity of the rendezvous point. If the entry is made, the chaser and target vehicle AEG input blocks should be at the point where the wedge angle used in the violation test was computed. On this entry, the inclination and ascending node differentials are computed. These are used as corrections to the differentials that were computed when the program was called the first time. The only purpose for the second entry is to compute the corrections.

The indicator for the third type of entry ($KPC = \pm 3$) to the plane change program should now be set in the calling program. The iterative procedure that produces the in-plane maneuvers should be restarted. The last set of in-plane maneuvers that was computed will be use for a first guess. When the plane change subroutine is entered, the same computations will be made as were made on the first entry type, except that the corrections computed on the second entry type are applied.

The formulae used in the program are contained in the detailed flowchart (Figure PMMPNE - 1).

Table PMMPNE-1. System Parameters

[illegible]

*Type
H ~ Fixed-point half word
F ~ Fixed-point full word
E ~ Single-precision (floating-point)
D ~ Double-precision (floating-point)
X ~ Hexadecimal

SUBSCRIPT "C" REFERS TO CHASER, "T" TO TARGET, "PC" TO PLANE CHANGE VEHICLE, "PH" TO PHANTOM, "NOPC" TO NON-PLANE CHANGE VEHICLE

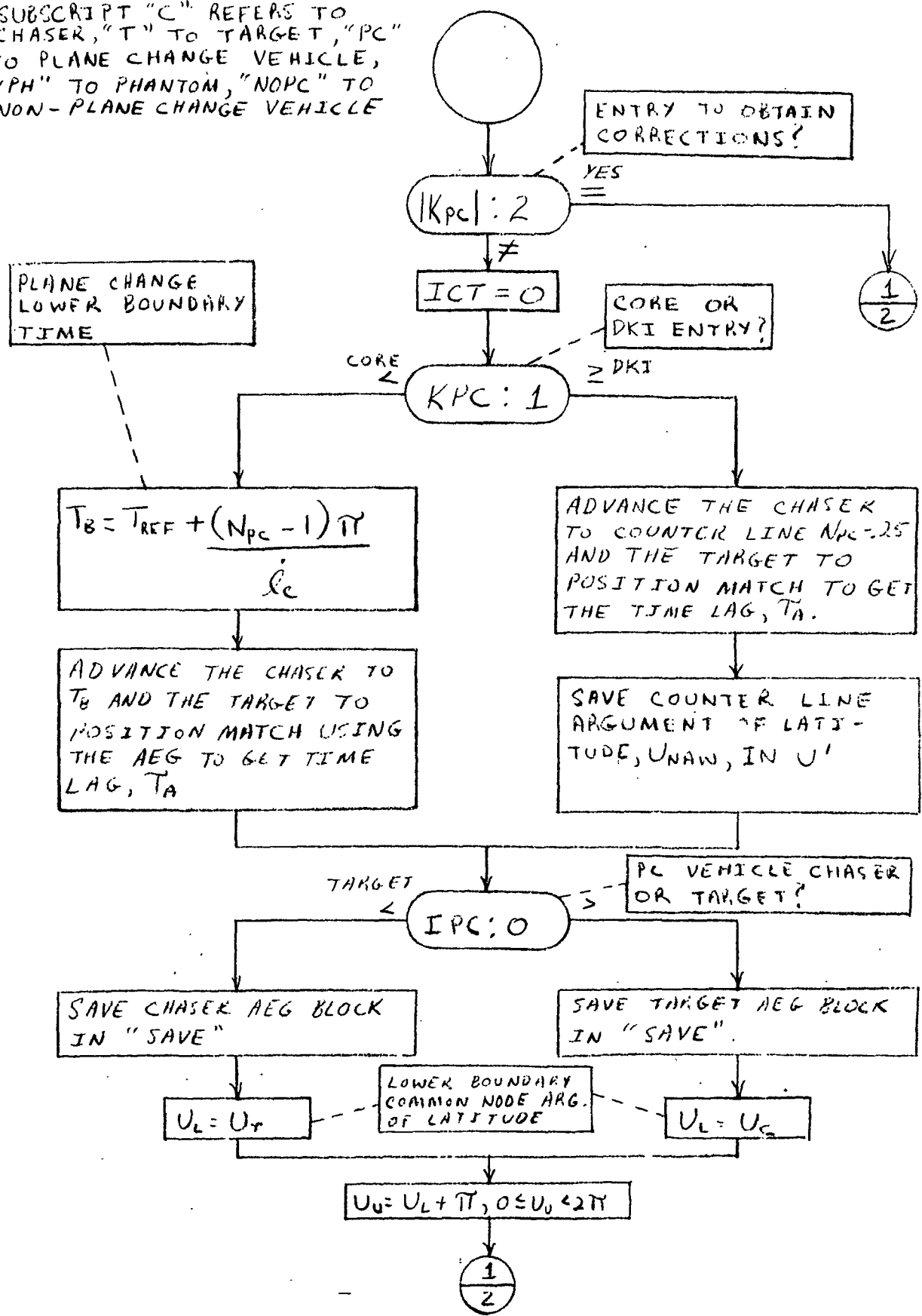


Figure PMMPNE-1. Flowchart (Sheet 1 of 9)

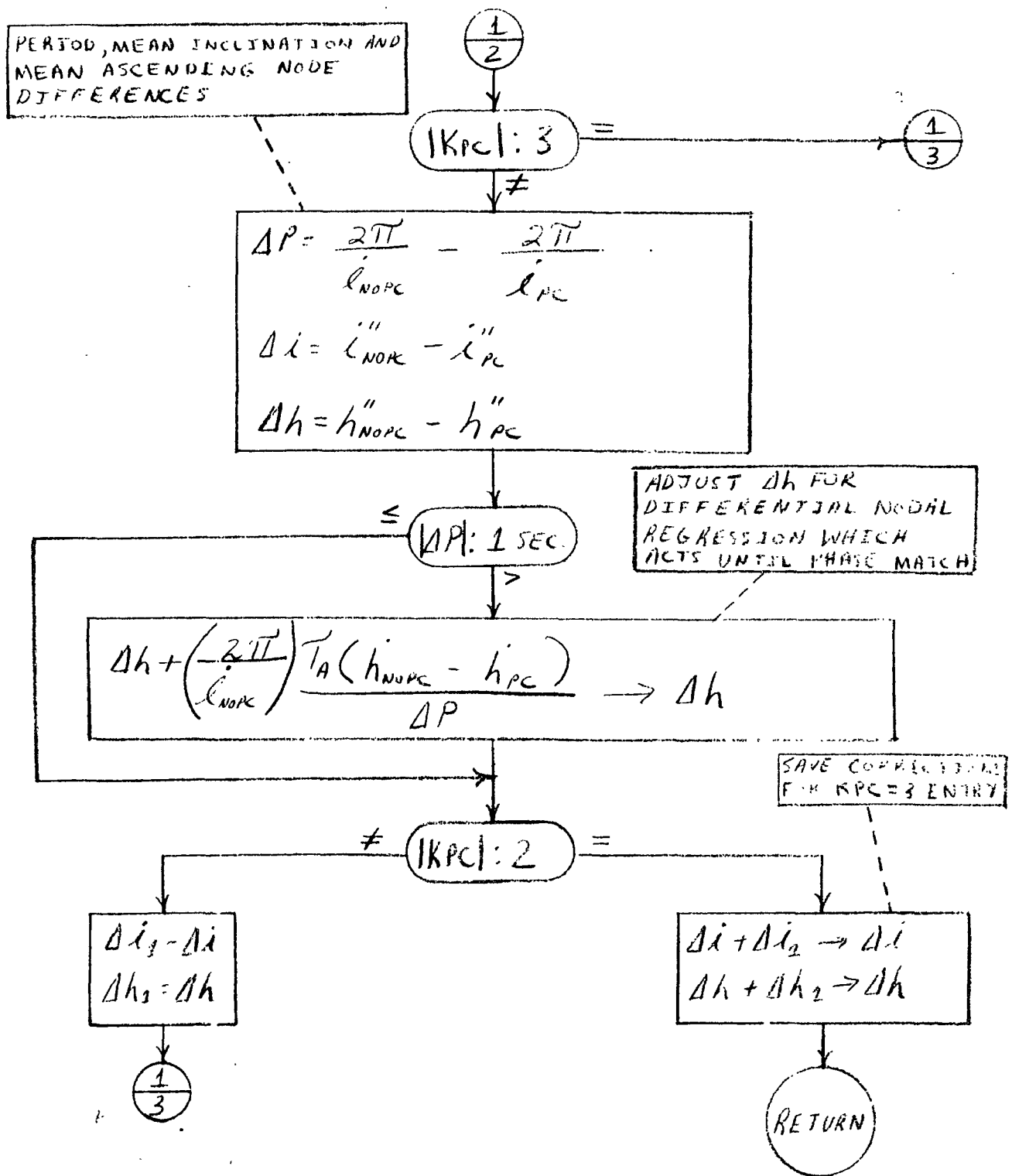


Figure PMMPNE-1. Flowchart (Sheet 2 of 9)

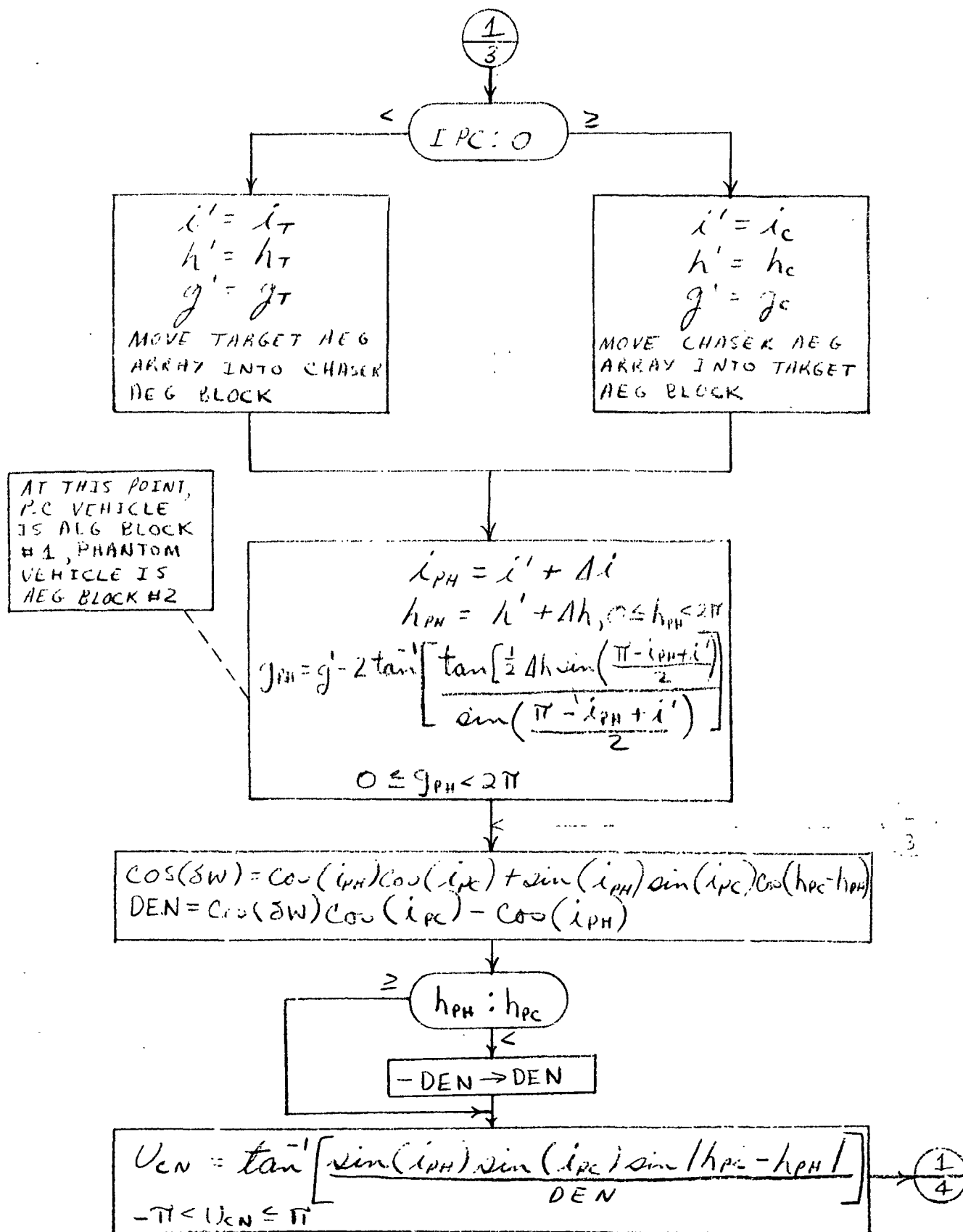


Figure PMMPNE-1. Flowchart (Sheet 3 of 9)

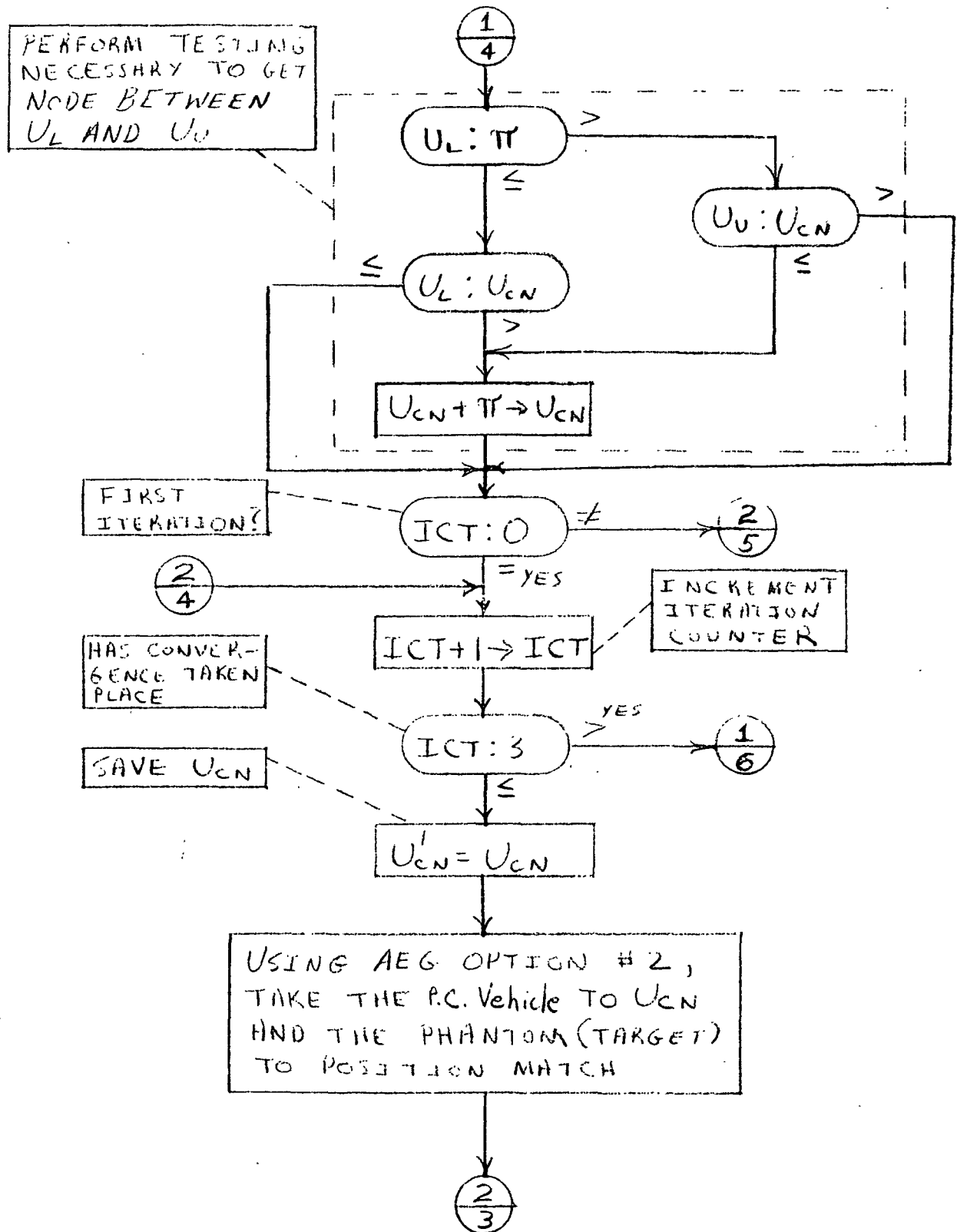


Figure PMMPNE-1. Flowchart (Sheet 4 of 9)

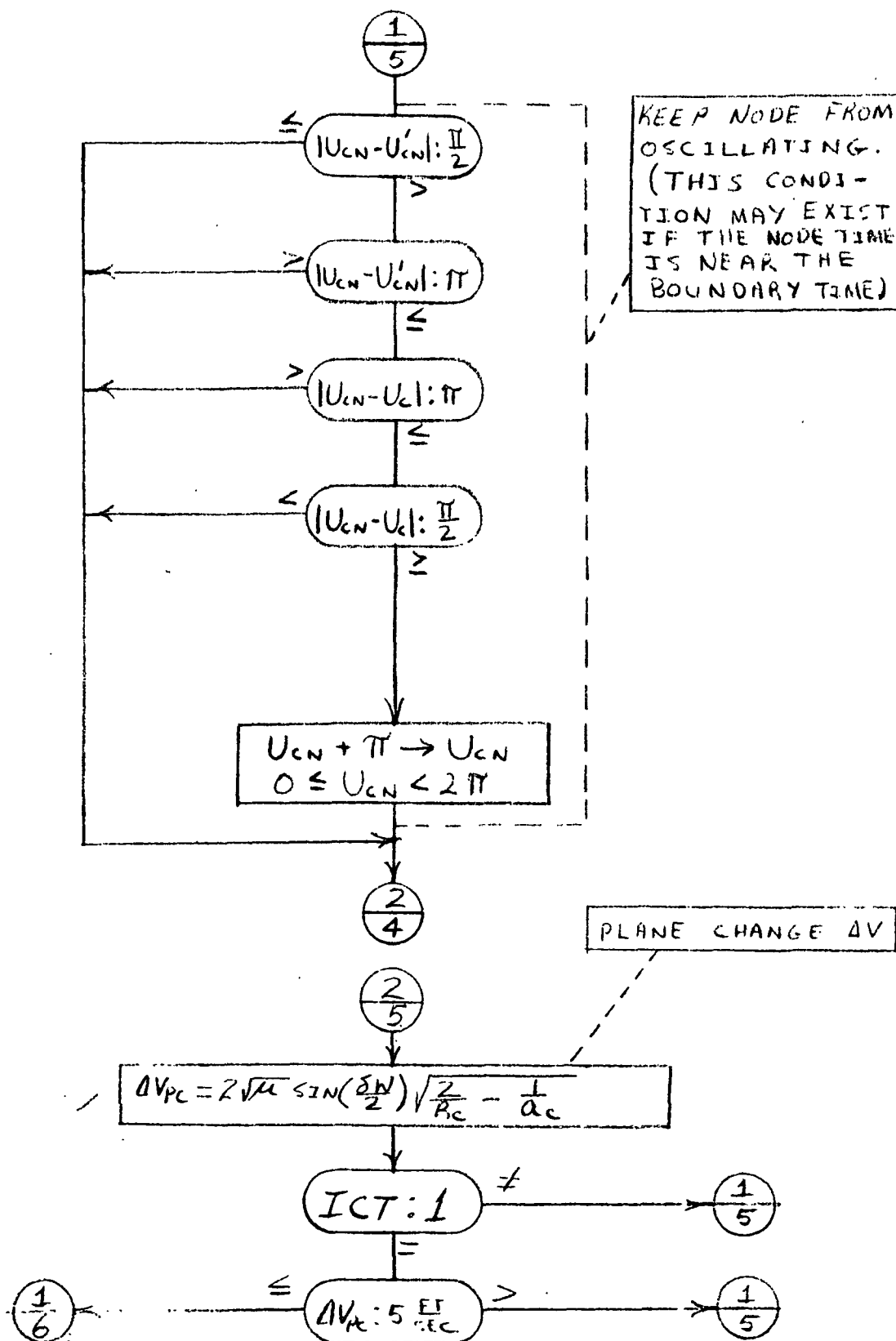


Figure PMMPNE-1. Flowchart (Sheet 5 of 9)

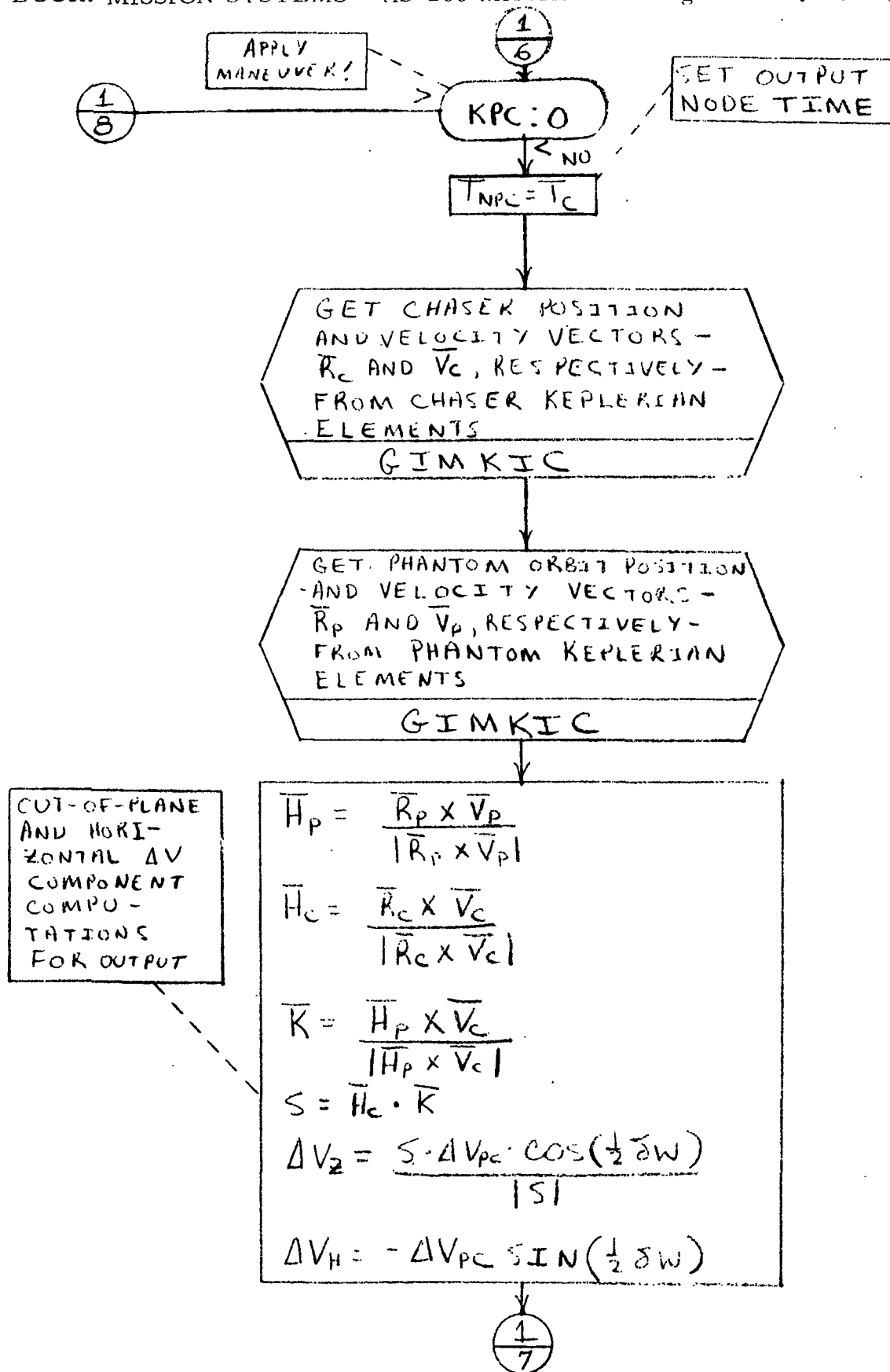


Figure PMMPNE-1. Flowchart (Sheet 6 of 9)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMPNE

DATE 1/2/68 (EORP)

REV

PAGE 11 of 13

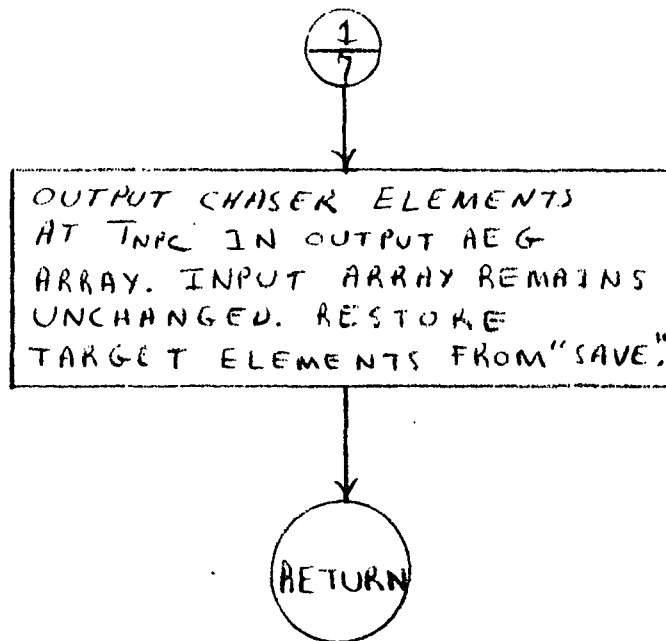


Figure PMMPNE-1. Flowchart (Sheet 7 of 9)

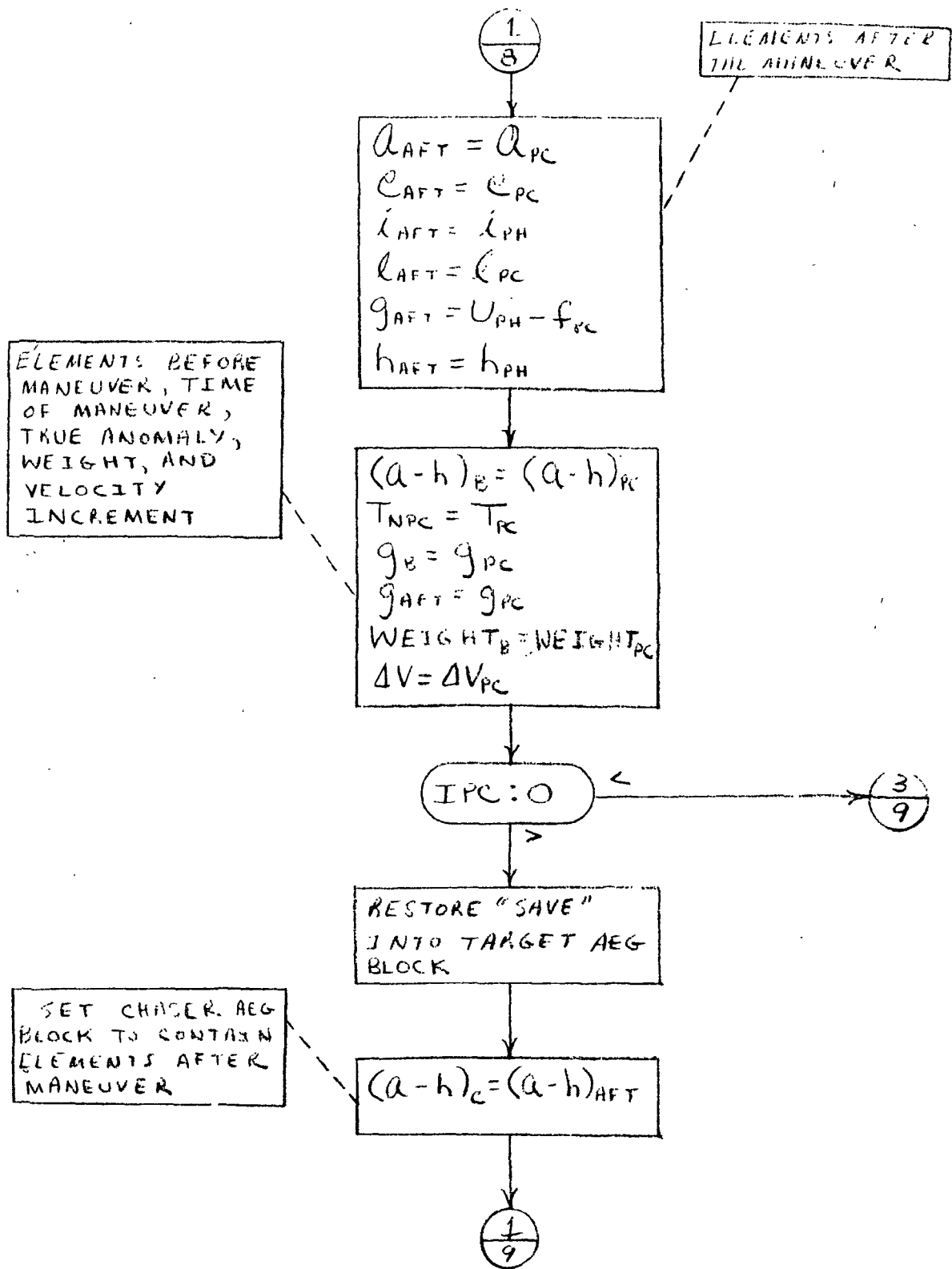


Figure PMMPNE-1. Flowchart (Sheet 8 of 9)

COUNTER LINE ARGUMENT
OF LATITUDE REFLECTED
INTO CHASER PLANE AFTER
PLANE CHANGE MANEUVER.

1
9

$$U_{NAW} = U' - 2 \tan^{-1} \left[\frac{\cos\left(\frac{\lambda_B + \lambda_{AFT}}{2}\right) \tan\left(\frac{h_{AFT} - h_E}{2}\right)}{\cos\left(\frac{\lambda_{AFT} - \lambda_B}{2}\right)} \right]$$

SET UP AEG INPUTS AND TAKE
CHASER ELEMENTS AFTER MANEUVER
TO U_{NAW}

DEFINE CHASER COUNTER LINE
TO BE CURRENT U AND TIME

2
9

OUTPUT NEW AEG
ARRAYS AND ELE-
MENTS TABLE BLOCK

RETURN

3
9

$(a-h)_T = (a-h)_{AFT}$
 $TIME_T = T_{NPC}$
SET AEG INITIALIZATION
INDICATOR.

RESTORE "SAVE" INTO
CHASER AEG BLOCK

2
9

Figure PMMPNE-1. Flowchart (Sheet 9 of 9)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMRKJ

DATE 11/10/67 (205)

REV

PAGE 1 of 47

- a. Symbol — PMMRKJ
- b. Program Name — CSM/LM Powered Flight Numerical Integration and Guidance Module
- c. Programming Group — Maneuver Control
- d. Programmer — J. Gardner
- e. Language — FORTRAN
- f. Data Tables Used — None
- g. MED's Used — None
- h. Writeup Completion Date — 10/13/67
- i. Input — Table PMMRKJ-1
- j. Auxiliary Output Table — Table PMMRKJ-2
- k. Ephemeris Table — Table PMMRKJ-3
- l. Table of Weights — Table PMMRKJ-4
- m. System Parameters — Table PMMRKJ-5
- n. Description of Terms — Table PMMRKJ-6
- o. Flowchart — Figure PMMRKJ-1

FUNCTION

Load Module PMMRKJ is a program designed to integrate through periods of powered flight and short spans of free flight. A fourth order Runge-Kutta, Nystrom, numerical integration technique is used to generate an ephemeris table of position and velocity vectors, an auxiliary output table, and a table of weights.

USAGE

- a. Calling Sequence — In assembly language:
LINK EP = PMMRKJ, PARAM=(T, I, E, A, W)
In FORTRAN:
CALL LINK (PMMRKJ, T, I, E, A, W)
where:
T = 60 double-word input table
I = error indicator (half word)
E = output ephemeris table
A = 40-word auxiliary output table for maneuver
A = 20-word auxiliary output table for free flight
W = output table of weights
- b. Inputs — Table PMMRKJ-1
- c. Outputs
 - 1. Table PMMRKJ-2 auxiliary output table
 - 2. Table PMMRKJ-3 ephemeris table
 - 3. Table PMMRKJ-4 table of weights

Approval	Approval
<i>J. Gardner</i>	<i>L. E. James</i>

- d. Other Programs Used — GIMGBL, GLFSAT, GLFDEN, GLMRTM, DEXP, DCOS, DSQRT, DLOG, PCRUNG, PCRDD, PCGUID, PCINIT, DSPARM
- e. Data Restrictions and Limits
 - 1. All times input must be positive and conform to the following restrictions: anchor time $\leq$ left limit $\leq$ right limit
anchor time $\leq$ time to begin maneuver
 - 2. The auxiliary output table is initialized to zero; thus, conditional type entries in this table will be zero unless the condition is met.
 - 3. For RCS maneuvers ΔV desired must be ≥ 0 . That is, a desired ΔV along the X-body axis is not allowed.
- f. Error — Upon exit the error indicator will be set as follows:
 - 0 for no errors
 - 1 for impacted before reaching end conditions
 - 2 for exceeded available fuel
 - 3 for error in guided maneuver definition

If an error condition is met, the $\bar{r}$, $\bar{v}$, and t at the time the error occurred will be stored in words 8-14 of the auxiliary output table.

Table PMMRKJ-1. Input

<u>Word</u>	<u>Byte</u>	<u>Type</u>	<u>Units</u>	<u>Description and Range</u>
1-3	1-8	D	Er	Starting position vector
4-6	1-8	D	Er/hr.	Starting velocity vector
7	1-8	D	hr.	Starting time
8	1-8	D	hr.	Ephemeris output time cycle
9	1-8	D	hr.	Left integration limit
10	1-8	D	hr.	Right integration limit
11	1-8	D	hr.	Kickout altitude (if ≤ 0 , the system parameter value will be used)
12	1-8	D	-	Weight loss rate multiplier (should normally be 1.0)
13	1-8	D	hr.	Time of LM jettison of descent stage
14	1-8	D		Open for expansion
15	1-8	D		Density multiplier (should normally be 1.0)
16	1-8	D	ft ²	Vehicle cross sectional area
17	1-8	D	lbs.	Weight of specified configuration
18	1,2	H		Ephemeris output option: -1 for $\bar{r}$ only 0 for none +1 for $\bar{r}$ and $\bar{v}$ +2 for $\bar{r}$ and $\bar{v}$ and table of weights

Table PMMRKJ-1. Input (Cont'd.)

<u>Word</u>	<u>Byte</u>	<u>Type</u>	<u>Units</u>	<u>Description and Range</u>
18	3, 4	H		Auxiliary output option: 1 for table desired 0 for table not desired
18	7, 8			Open for expansion
19	1-8			CSM WT
20	1-8			LM WT
21	1, 2	H		Attitude code: 0 - no maneuver ±1 - fixed inertial ±2 - manual holding body orientation invariant ±3 - Lambert Guidance* ±4 - external ΔV (primary) ±5 - external ΔV (LM, AGS) negative sign indicates initial attitudes are siren.
21	3, 4	H		Thruster code: 1 - CSMRCS+X (2 thrusters) 2 - CSMRCS+X (4 thrusters) 3 - CSMRCS-X (2 thrusters) 4 - CSMRCS-X (4 thrusters) 16 - S-IVB blowdown vent 17 - LEMRCS+X (2 thrusters) 18 - LEMRCS+X (4 thrusters) 19 - LEMRCS-X (2 thrusters) 20 - LEMRCS-X (4 thrusters) 33 - CSMSPS 34 - LEMAPS 35 - LEMDPS
21	5, 6	H		Ullage option: 0 - two thrusters 1 - four thrusters
21	7, 8			Open for expansion
22	1, 2	H		Configuration code at maneuver initiation: 0 for CSM 1 for LM 2 for CSM and LM
23	1-8	D	hr.	Time to begin maneuver (first phase)
24	1-8	D	hr.	Δt of ullage (=0 for RCS)
25	1-8	D	hr.	Δt of maneuver (no pure ullage or tailoff) (=0 if ΔV is given)

*Lambert Guidance not a requirement for EORP.

Table PMMRKJ-1. Input (Cont'd.)

Word	Byte	Type	Units	Description and Range
26	1-8	D	Er/hr.	ΔV desired (if < 0 it is ΔV along X-body excluding tailoff)
*27				$\left. \begin{matrix} X \\ Y \\ Z \end{matrix} \right\} \begin{matrix} \hat{a} \\ \hat{a} \\ \hat{a} \end{matrix} \begin{matrix} T \\ T \\ T \end{matrix}$ RTCC inertial coordinates of unit thrust vector at burn initiate
28				
29				
**30	1-8	D		$\left. \begin{matrix} X \\ Y \\ Z \end{matrix} \right\} \begin{matrix} \hat{X} \\ \hat{X} \\ \hat{X} \end{matrix} \begin{matrix} B \\ B \\ B \end{matrix}$ RTCC inertial coordinates of the unit vectors in the direction of the vehicle body X, Y, Z axes, respectively
31	1-8	D		
32	1-8	D		
33	1-8	D		$\left. \begin{matrix} X \\ Y \\ Z \end{matrix} \right\} \begin{matrix} \hat{Y} \\ \hat{Y} \\ \hat{Y} \end{matrix} \begin{matrix} B \\ B \\ B \end{matrix}$ RTCC inertial coordinates of the unit vectors in the direction of the vehicle body X, Y, Z axes, respectively
34	1-8	D		
35	1-8	D		
36	1-8	D		$\left. \begin{matrix} X \\ Y \\ Z \end{matrix} \right\} \begin{matrix} \hat{Z} \\ \hat{Z} \\ \hat{Z} \end{matrix} \begin{matrix} B \\ B \\ B \end{matrix}$ axes, respectively
37	1-8	D		
38	1-8	D		
39	1-8	D	hr.	Δt at 10% thrust level for DPS thruster: Set negative if DPS. Short burn test is to be ignored.
40	1-2	H		Maneuvering vehicle code: 1 - for CSM 2 - for S-IVB 3 - for LM
40	3-4			Thruster trim angle indicator: -1 if angles to be computed +1 if system parameters are to be used
40	5-8			Open for expansion
41	1-8			Docking angle
42	1-8			Open for expansion
43	1-8			Open for expansion
44	1-8	D	Er/hr.	$\left. \begin{matrix} V_X \\ V_Y \\ V_Z \end{matrix} \right\}$ External $\overline{\Delta V}$ in RTCC inertial coordinate system
45	1-8	D	Er/hr.	
46	1-8	D	Er/hr.	
47-60	1-8	D		Open for expansion

*Words 27, 28, 29 need be set only for inertial maneuvers.

**Words 30, 31, 32 must be set for all multi-phase maneuvers.

Table PMMRKJ-2. Auxiliary Output

<u>Word</u>	<u>Byte</u>	<u>Units</u>	<u>Description and Range</u>
1-3	1-8	Er	For no maneuvers: $\bar{r}$, $\bar{v}$, t at t_L
4-6	1-8	Er/hr.	For maneuvers: $\bar{r}$, $\bar{v}$, t at burn initiate
7	1-8	hr.	
8-10	1-8	Er	For no maneuvers: $\bar{r}$, $\bar{v}$, t at t_R
11-13	1-8	Er/hr.	For maneuvers: $\bar{r}$, $\bar{v}$, t at end of maneuver
14	1-8	hr.	Time at 400K feet
15	1-8	hr.	Geodetic latitude at 400K feet
16	1-8	rad.	Longitude at 400K feet
17	1-8	rad.	Time at 25K feet
18	1-8	hr.	Geodetic latitude at 25K feet
19	1-8	rad.	Longitude at 25K feet
20	1-8	rad.	
21	1-8	Er/hr.	V_{gx} } Velocity to be gained at G and N
22	1-8	Er/hr.	V_{gy} } initiate in inertial coordinates
23	1-8	Er/hr.	V_{gz} }
24	1-8	Er/hr.	ΔV of maneuver along X-body axis excluding tailoff
25	1-8	Er/hr.	Total ΔV of tailoff
26	1-8	Er/hr.	Total ΔV of maneuver
27	1-8	hr.	Δt of maneuver excluding tailoff
28	1-8	lbs.	Weight at end of maneuver
29	1-8		Weight loss during maneuver
30	1-8		Area
31	1-8		$\hat{a}_T$ } inertial coordinates of unit
32	1-8		$\hat{a}_T$ } thrust vector at burn initiate
33	1-8		
34	1-8	Er/hr.	ΔV of tailoff along X-body axis (does not apply for RCS maneuvers)
35	1-8	rad.	Pitch gimbal at burn initiate
36	1-8	rad.	Yaw gimbal at burn initiate
37	1-8		ΔV of pure ullage
38	1-8	hrs.	Δt of tailoff
39	1-8		Weight at main engine on
40	1-8		Open for expansion

Table PMMRKJ-3. Ephemeris

<u>Word</u>	<u>Byte</u>	<u>Units</u>	<u>Description and Range</u>
1	1-8	hr.	Timer interval at which ephemeris is stored
2	1-8	hr.	Time associated with first entry in ephemeris
3	1-8	hr.	Time associated with last entry in ephemeris
4-6	1-8	Er	Position vector - 1st entry
7-9	1-8	Er/hr.	Velocity vector - 1st entry
10-12	1-8	Er	Position vector - 2nd entry
13-15	1-8	Er/hr.	Velocity vector - 2nd entry
			Etc.

Table PMMRKJ-4. Table of Weights

<u>Word</u>	<u>Byte</u>	<u>Units</u>	<u>Description and Range</u>
1	1-8	lbs.	Weight - 1st entry
2	1-8	lbs.	Weight - 2nd entry
			Etc.

METHOD

This program integrates the Newtonian equation of motion,

$$\ddot{\vec{r}} = -\frac{\partial V}{\partial \vec{r}} + \vec{F},$$

where

$$\frac{-\partial V}{\partial \vec{r}} = \frac{-\mu \vec{r}}{r^3} \left[1 + J_2 \left(\frac{a}{r} \right)^2 \left(1 - 5 \left(\frac{z}{i} \right)^2 \right) \right] - \frac{2\mu Z}{r^5} J_2 a^2 \hat{Z}$$

where $\hat{Z} = (0, 0, 1)$

$$-\frac{\partial V}{\partial X} = \frac{-\mu X}{r^3} \left[1 + J_2 \left(\frac{a}{r} \right)^2 \left(1 - 5 \left(\frac{Z}{r} \right)^2 \right) \right]$$

$$-\frac{\partial V}{\partial Y} = \frac{-\mu Y}{r^3} \left[1 + J_2 \left(\frac{a}{r} \right)^2 \left(1 - 5 \left(\frac{Z}{r} \right)^2 \right) \right]$$

$$-\frac{\partial V}{\partial Z} = \frac{-\mu Z}{r^3} \left[1 + J_2 \left(\frac{a}{r} \right)^2 \left(3 - 5 \left(\frac{Z}{r} \right)^2 \right) \right]$$

where $J_2 = (1.62345) (10)^{-3}$ = second harmonic coefficient,
 a = Earth's equatorial radius
 μ = gravitational constant times mass of Earth.

For free-flight integration, $\bar{F}$ is given by

$$\bar{F} = \bar{D}$$

where D = atmospheric drag acceleration.

The three components of drag acceleration are given by:

$$D_x = -1/2 \rho \frac{A}{M} C_D V_r \dot{X}_r,$$

$$D_y = -1/2 \rho \frac{A}{M} C_D V_r \dot{Y}_r,$$

$$D_z = -1/2 \rho \frac{A}{M} C_D V_r \dot{Z}_r,$$

where $\dot{X}_r = \dot{X} + Y\omega$

$$\dot{Y}_r = \dot{Y} - X\omega$$

$$\dot{Z}_r = \dot{Z}$$

$$V_r = \left(\dot{X}_r^2 + \dot{Y}_r^2 + \dot{Z}_r^2 \right)^{1/2},$$

ω = Earth's rotational velocity,

A = cross sectional area,

M = mass

C_d = coefficient of atmospheric drag,

ρ = atmospheric density.

The thrust acceleration vector is given by:

$$\bar{T} = \frac{T_o G}{W} \hat{A}_T$$

where T_o = thrust level,

W = weight,

G = acceleration constant,

$\hat{A}_T$ = unit thrust vector

The maneuver thrust acceleration vector is a function of the engine, maneuver phase, and maneuver type. For the LM RCS, and CSM RCS engines the thrust level is constant throughout the burn. The unit thrust vector for RCS maneuvers is determined as follows:

Fixed Inertial

The unit thrust vector is input and is held throughout the maneuver.

Manual

The unit thrust vector is given by $\hat{A}_T = \hat{X}_b$ where $\hat{X}_b$ = inertial coordinates of X body axis

External $\Delta\bar{V}$

When the initial attitudes are given, the initial unit thrust vector is given by

$$\hat{A}_T = \hat{X}_b$$

Steering is suppressed for two computational cycles (4 seconds). At this time the desired unit thrust vectors for the upcoming cycle are given by

$$\hat{A}_{TR} = \text{Unit}(\bar{V}_g)$$

where

$$\bar{V}_g = \text{input external } \Delta\bar{V}$$

$\hat{A}_{TR}$ = unit thrust vector at the end of current computational cycle

$$\hat{A}_{TL} = \hat{A}_T$$

$$\hat{A}_{TM} = \text{unit} (\hat{A}_{TL} + \hat{A}_{TR})$$

The unit thrust vector for the upcoming computational cycle is then given by:

$$\hat{A}_T = \hat{A}_{TL} \text{ for } k_1,$$

$$\hat{A}_T = \hat{A}_{TM} \text{ for } k_2 \text{ and } k_3,$$

$$\hat{A}_T = \hat{A}_{TR} \text{ for } k_4$$

When the initial attitudes are not given, the initial unit thrust vector is given by $\hat{A}_T = \text{unit} (\bar{v}_g)$. The above procedure is repeated each computational cycle with $\bar{v}_g$ being updated by

$$\bar{v}_g = \bar{v}_g - (\bar{v}_n - \bar{v}_n') - \frac{1}{2} \Delta T_p (\bar{g} + \bar{g}')$$

where

$\bar{v}_n$ = current velocity vector,

$\bar{v}_n'$ = previous velocity vector,

$\bar{g}$ = current acceleration due to drag and Earth's potential

$\bar{g}'$ = previous $\bar{g}$

ΔT_p = elapsed time from $\bar{v}_n'$ to $\bar{v}_n$

The APS and SPS maneuvers are five phase:

Phase 1 - ullage

Phase 2 - ullage plus buildup

Phase 3 - ullage plus main engine

Phase 4 - main engine

Phase 5 - tailoff

The DPS maneuver is six phase:

- Phase 1 - ullage
- Phase 2 - ullage plus ten percent
- Phase 3 - ten percent
- Phase 4 - buildup of main engine
- Phase 5 - main engine
- Phase 6 - tailoff

The thrust level for each phase, except tailoff and main engine phase, is a system parameter. The thrust level during main engine phase is a function of the weight and is computed by subroutine GIMGBL. The thrust level for tailoff is

$$T = \frac{T_c}{T_o} T_{to}$$

where

T_c = thrust at engine off,

T_o = full load thrust,

T_{to} = tailoff thrust corresponding to full load thrust

During the ullage phase of APS, DPS, and SPS maneuvers the unit thrust vector is given by

$$\hat{A}_T = \hat{X}_b.$$

At main engine on the unit thrust vector is determined as follows:

Fixed Inertial

When the initial attitudes are given the unit thrust vector is determined in the same manner as for a manual maneuver. When the initial attitudes are not given, the unit thrust vector is input. In both cases the initial direction is held throughout the burn.

Manual

The unit thrust vector is determined from a sequence of rotations:

$$\begin{bmatrix} \hat{A}_T \\ \hat{Y}_T \\ \hat{Z}_T \end{bmatrix} = R_Z(\theta_Z) R_Y(\theta_Y) \begin{bmatrix} \hat{X}_b \\ \hat{Y}_b \\ \hat{Z}_b \end{bmatrix}$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMRKJ
 DATE 11/10/67 (205)
 REV
 PAGE 11 of 47

where

- $\hat{X}_b$ = inertial coordinates of X body axis,
 $\hat{Y}_b$ = inertial coordinates of Y body axis,
 $\hat{Z}_b$ = inertial coordinates of Z body axis,
 R_Y = rotation about Y body axis,
 R_Z = rotation about Z body axis,
 θ_Z = cant angles for APS thruster,
 θ_Y = cant angles for APS thruster,
 θ_Z = gimbal angles so as to fire through the
 θ_Y = center of gravity for SPS and DPS
 thrusters.

External $\Delta\bar{V}$

When the initial attitudes are given, the initial unit thrust vector is determined in the same manner as for a manual maneuver. Steering is suppressed for two computational cycles after main engine on. After this time, the unit thrust vector is determined as described for RCS external $\Delta\bar{V}$ maneuvers. When the initial attitudes are not given, the initial unit thrust vector is determined as described for RCS external $\Delta\bar{V}$ maneuver.

The numerical integration technique used to integrate the equations of motion is the Nystrom modification of a fourth-order Runge-Kutta scheme given by:

$$r_{n+1,i} = r_{n,i} + h \left[r_{n,i} + \frac{1}{6} (K_{1,i} + K_{2,i} + K_{3,i}) \right],$$

$$r_{n+1,i} = r_{n,i} + \frac{1}{6} (K_{1,i} + 2K_{2,i} + 2K_{3,i} + K_{4,i}),$$

where:

$$K_{1,i} = h \ddot{r}(t_n, r_{n,i}, \dot{r}_{n,i}),$$

$$K_{2,i} = h \ddot{r} \left[\left(t_n + \frac{h}{2} \right), \left(r_{n,i} + \frac{h}{2} \dot{r}_{n,i} + \frac{h}{8} K_{1,i} \right), \left(\dot{r}_{n,i} + \frac{1}{2} K_{1,i} \right) \right],$$

$$K_{3,i} = h \ddot{r} \left[\left(t_n + \frac{h}{2} \right), \left(r_{n,i} + \frac{h}{2} \dot{r}_{n,i} + \frac{h}{8} K_{1,i} \right), \left(\dot{r}_{n,i} + \frac{1}{2} K_{2,i} \right) \right],$$

$$K_{4,i} = h \ddot{r} \left[(t_n + h), \left(r_{n,i} + h \dot{r}_{n,i} + \frac{h}{2} K_{3,i} \right), \left(\dot{r}_{n,i} + K_{3,i} \right) \right],$$

where:

h = integration time step,

r = position vector,

$\dot{r}$ = velocity vector,

$\ddot{r}$ = acceleration vector,

for i = x, y, z, and $n = 1, 2, 3, \text{-----}$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMRKJ

DATE 11/10/67 (205)

REV

PAGE 13 of 47

Table PMMRKJ-5. System Parameters

Symbol in Program	Symbol In Master Tape	Type *	Units	Comments
DTSPAN(2)	MCTDD2		hrs.	DPS phase 2 burn time
DTSPAN(4)	MCTDD5		hrs.	DPS burn time for short maneuver time
DTSPAN(5)	MCTDD4		hrs.	DPS build-up to maximum burn time
DTSPAN(6)	MCTAD6		hrs.	DPS elapsed time from engine on to initial steering
DTSPAN(7)	MCTDD7		hrs.	DPS time for ten percent test
DTSPAN(8)	MCTDD6		hrs.	DPS tail-off burn time
XK1	MCTSK1	D	lb-sec.	Impulse in one secone SPS burn
XK2VAL	MCTSK2	D	lb-sec.	Initial value of SPS minimum impulse curve
XK3VAL	MCTSK3	D	lb.	Slope of SPS curve
XK4	MCTSK4	D	lb-sec.	CSM RCS 7-second 2-jet ullage impulse
XK1	MCTAK1	D	lb-sec.	Impulse in one second APS burn
XK2VAL	MCTAK2	D	lb-sec.	Initial value of APS minimum impulse curve
XK3VAL	MCTAK3	D	lb.	Slope of APS curve
XK4	MCTAK4	D	lb-sec.	LM RCS 7-second 2-jet ullage impulse

*Type
 H - Fixed-point half word
 F - Fixed-point full word
 E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexidecimal



Table PMMRKJ-5. System Parameters (Cont'd.)

Symbol in Program	Symbol In Master Tape	Type *	Units	Comments
THPS(1)				
(2)	MCTDT2	D	lbs.	DPS phase 2 - thrust level
(3)				
(4)	MCTDT3		lbs.	DPS 10% thrust level
(5)	MCTDT4		lbs.	DPS buildup to maximum thrust level
(6)				
(7)				
(8)	MCTPT6		lbs.	DPS tail-off thrust level
(9)				
(10)	MCTDT5		lbs.	DPS full load thrust level
WDOTPS(1)				
(2)	MCTDW2		lbs. /hr.	DPS phase 2 weight loss rate
(3)				
(4)	MCTDW3	▼	lbs. /hr.	DPS weight loss rate at 10% thrust

*Type
 H - Fixed-point half word
 F - Fixed-point full word
 E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMRKJ
 DATE 11/10/67
 REV
 PAGE 15 of 47

(205)

Table PMMRKJ-5. System Parameters (Cont'd.)

Symbol in Program	Symbol in Master Tape	Type *	Units	Comments
WDOTPS(5)	MCTDW4	D	lbs./hr.	DPS buildup to maximum weight loss rate
(6)				
(7)				
(8)	MCTDW6		lbs./hrs.	DPS tail-off weight loss rate
(9)				
(10)	MCTDW5		lbs./hr.	DPS full load weight loss rate
THPS(2)	MCTAT2		lbs.	APS phase 2 thrust level
THPS(3)	MCTAT3		lbs.	APS phase 3 thrust level
THPS(7)	MCTAT6		lbs.	APS thrust level for short maneuver test
THPS(8)	MCTAT5		lbs.	APS tail-off thrust level
THPS(10)	MCTAT4		lbs.	APS full load thrust level
WDOTPS(2)	MCTAW2		lbs./hr.	APS phase 2 weight loss rate
WDOTPS(3)	MCTAW3		lbs./hr.	APS phase 3 weight loss rate
WDOTPS(7)	MCTAW6	▼	lbs./hr.	APS weight loss rate for short maneuver test

*Type
 H - Fixed-point half word
 F - Fixed-point full word
 E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal



Table PMMRKJ-5. System Parameters (Cont'd.)

Symbol in Program	Symbol in Master Tape	Type*	Units	Comments
WDOTPS(8)	MCTAW5	D	lbs./hr.	APS tail-off weight loss rate
DTSPAN(2)	MCTAD2		hrs.	APS phase 2 burn time
DTSPAN(3)	MCTAD3		hrs.	APS phase 3 burn time
DTSPAN(4)	MCTAD4		hrs.	APS burn time for short maneuver test
DTSPAN(6)	MCTAD6		hrs.	APS elapsed time from engine on to initial steering
DTSPAN(7)	MCTAD7		hrs.	APS minimum impulse time
DTSPAN(8)	MCTAD5		hrs.	APS tail-off burn time
HUA	MCAUAC		Er	Altitude of entry interface
AOBSQD	MCEABS			Geodetic Earth model A^2/B^2
OMEGA	MCERTS		radians/hr.	Angular velocity of Earth
G2	MCMShP			Second harmonic coefficient
G	MCEACG		Er/hr^2	Gravitational acceleration
XMU	MCEMUU		Er^3/hr^2	Gravitational constant
CDRAG	MCADRG			Coefficient of drag

*Type
 H - Fixed-point half word
 F - Fixed-point full word
 E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMRKJ

DATE 11/10/67 (205)

REV

PAGE 17 of 47

Table PMMRKJ-5. System Parameters (Cont'd.)

Symbol in Program	Symbol In Master Tape	Type *	Units	Comments
WTLIM	MCVCWT	D	lbs.	CSM limiting weight
WTLIM	MCVLWT	D	lbs.	LM limiting weight
THPS(1)	MCTJT7		lbs.	S-IVB blowdown vent thrust level
WDOTPS(1)	MCTJW7		lbs./hr.	S-IVB blowdown vent weight loss rate
THPS(1)	MCTLT1		lbs.	LM RCS (+ X, 2 quad.) thrust level
WDOTPS(1)	MCTLW1		lbs./hr.	LM RCS (+ X, 2 quad.) weight loss rate
THPS(1)	MCTLT2		lbs.	LM RCS (+ X, 2 quad.) thrust level
WDOTPS(1)	MCTLW2		lbs./hr.	LM RCS (+ X, 2 quad.) weight loss rate
THPS(1)	MCTLT3		lbs.	LM RCS (- X, 2 quad.) thrust level
WDOTPS(1)	MCTLW3		lbs./hrs.	LM RCS (- X, 2 quad.) weight loss rate
THPS(1)	MCTLT4		lbs.	LM RCS (- X, 4 quad.) thrust level
WDOTPS(1)	MCTLW4		lbs./hrs.	LM RCS (- X, 4 quad.) weight loss rate
THPS(1)	MCTCT1		lbs.	CSM RCS (+ X, 2 quad.) thrust level
WDOTPS(1)	MCTCW1	▶	lbs./hr.	CSM RCS (+ X, 2 quad.) weight loss rate

*Type
 H — Fixed-point half word
 F — Fixed-point full word
 E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal



Table PMMRKJ-5. System Parameters (Cont'd.)

Symbol in Program	Symbol In Master Tape	Type *	Units	Comments
THPS(1)	MCTCT2	D	lbs.	CSM RCS (+ X, 4 quad.) thrust level
WDOTPS(1)	MCTCW2		lbs./hr.	CSM RCS (+ X, 4 quad.) weight loss rate
THPS(1)	MCTCT3		lbs.	CSM RCS (- X, 2 quad.) thrust level
WDOTPS(1)	MCTCW3		lbs./hr.	CSM RCS (- X, 2 quad.) weight loss rate
THPS(1)	MCTCT4		lbs.	CSM RCS (- X, 4 quad.) thrust level
WDOTPS(1)	MCTCW4		lbs./hr.	CSM RCS (- X, 4 quad.) weight loss rate
PITCHG	MCTSPG		radians	SPS trim angles
YAWG	MCTSYG		radians	SPS trim angles
PITCHG	MCTAPG		radians	APS cant angles
YAWG	MCTARG		radians	APS cant angles
PITCHG	MCTDPG		radians	DPS trim angles
YAWG	MCTDRG		radians	DPS trim angles
THPS(2)	MCTST2		lbs.	SPS phase 2 thrust level
THPS(3)	MCTST3	▼	lbs.	SPS phase 3 thrust level

*Type
H - Fixed-point half word
F - Fixed-point full word
E - Single-precision (floating-point)
D - Double-precision (floating-point)
X - Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMRKJ
 DATE 11/10/67 (205)
 REV
 PAGE 19 of 47

Table PMMRKJ-5. System Parameters (Cont'd.)

Symbol in Program	Symbol in Master Tape	Type *	Units	Comments
THPS(7)	MCTST6	D	lbs.	SPS thrust level for short maneuver test
(8)	MCTST5		lbs.	SPS tail-off thrust level
THPS(10)	MCTST4		lbs.	SPS full load thrust level
WDOTPS(2)	MCTSW2		lbs./hr.	SPS phase 2 weight loss rate
WDOTPS(3)	MCTSW3		lbs./hr.	SPS phase 3 weight loss rate
WDOTPS(7)	MCTSW6		lbs./hr.	SPS weight loss rate for short maneuver test
WDOTPS(8)	MCTSW5		lbs./hr.	SPS tail-off weight loss rate
WDOTPS(10)	MCTSW4		lbs./hr.	SPS full load weight loss rate
DTSPAN(2)	MCTSD2		hours	SPS phase 2 burn time
DTSPAN(3)	MCTSD3		hours	SPS phase 3 burn time
DTSPAN(4)	MCTSD4		hours	SPS burn time for short maneuver test
DTSPAN(6)	MCTSD6		hours	SPS elapsed time from engine on to initial steering
DTSPAN(7)	MCTSD7		hours	SPS minimum impulse time
DTSPAN(8)	MCTSD5	▲	hours	SPS tail-off burn time

*Type
 H - Fixed-point half word
 F - Fixed-point full word
 E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal



Table PMMRKJ-6. Description of Terms

Definition of Term	Symbol In Program	Symbol In Model	Med I/O	* Type	Units	Limits (Min/Max)	Comments
Maneuver initiation time	TBM	TBM		2	hours		
Kickout altitude	HIMPCT	H_{Ro}		2	Er		
Current weight	WT	wt.		2	lbs.		
Input weight	CAPWT	Wt_{in}		2	lbs.		
Working position vector	RP	r_p		2	Er		
Current position vector	R	r		2	Er		
Working velocity vector	VP	v_p		2	Er/hr.		
Current velocity vector	V	v		2	Er/hr.		
Previous time	TPREV	T_{prev}		2	hours		
Previous step size	DTPREV	ΔT^1		2	hours		

*Type Code:

X = integer (fixed point)

1 = single precision (floating point)

2 = double precision (floating point)

C = command

L = logical

H = binary coded decimal (Hollerith)

O = octal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMRKJ

DATE 11/10/67 (205)

REV

PAGE 21 of 47

Table PMMRKJ-6. Description of Terms (Cont'd.)

Definition of Term	Symbol In Program	Symbol In Model	Med I/O	* Type	Units	Limits (Min/Max)	Comments
Previous step through guidance	DTP	ΔT_p		D	hours		
Current step through guidance	DTN	ΔT_n			hours		
Potential acceleration	RDDP	g			Er/hr. ²		
Current integration step	STEP	Step			hours		
Free-flight step size	DFFF	ΔT_{ff}			hours		
Time of next output	TNEXT	T_{next}			hours		
Indicator	TLARGE	T_{large}			hours		
Time of next event	TI	T_i			hours		
End of current step	TE	T_e		→	hours		
Ephemeris Counter	JT	n		H			

*Type Code:

C = command

X = integer (fixed point)

L = logical

1 = single precision (floating point)

H = binary coded decimal (Hollerith)

2 = double precision (floating point)

O = octal



Table PMMRKJ-6. Description of Terms (Cont'd.)

Definition of Term	Symbol In Program	Symbol In Model	Med I/O	Type*	Units	Limits (Min/Max)	Comments
Time to begin guidance	TGUID	T _g		D	hours		
Sign	SIGN	S _{gn}		D			
Weight table counter	KWT	KWT		D			
Current altitude comparison	HH	HH		D			
Previous altitude	ALTPR	ALT'		D	Er		
Thruster switch	KTHSWT	KTHSWT		H			
Error indicator	IERR	IERR		H			
Left integration limit	TLEFT	T _{left}		D	hours		
Anchor time	TA	TA		D	hours		
Maneuver type	MANOP	MANOP		H			

*Type Code:

X = integer (fixed point)

1 = single precision (floating point)

2 = double precision (floating point)

C = command

L = logical

H = binary coded decimal (Hollerith)

O = octal

Table PMMRKJ-6. Description of Terms (Cont'd.)

Definition of Term	Symbol In Program	Symbol In Model	Med I/O	* Type	Units	Limits (Min/Max)	Comments
Weight loss rate multiplier	WDMULT	$\dot{W}_{mult}$		D			
Thrust along X body axis	TH	Th_x		D	lbs.		
Current altitude	ALT	ALT		D	Er		
End of current integration step	TAU	T_{au}		D	hours		
Current time	T	T		D	hours		
Right integration limit	TRIGHT	T_{right}		D	hours		
Ullage time	DTUL	ΔT_{UL}		D	hours		
Pure ullage time	DTU	ΔT_u		D	hours		
Ullage plus buildup time	DTUB	ΔT_{ub}		D	hours		
Ullage plus overlap time	DTUSPS	ΔT_{usps}		D	hours		

*Type Code:

C = command

X = integer (fixed point)

L = logical

1 = single precision (floating point)

H = binary coded decimal (Hollerith)

2 = double precision (floating point)

O = octal

Table PMMRKJ-6. Description of Terms (Cont'd.)

[illegible]

Type Code:

X = integer (fixed point)

1 = single precision (floating point)

2 = double precision (floating point)

C = command

L = logical

H = binary coded decimal (Hollerith)

Q - total

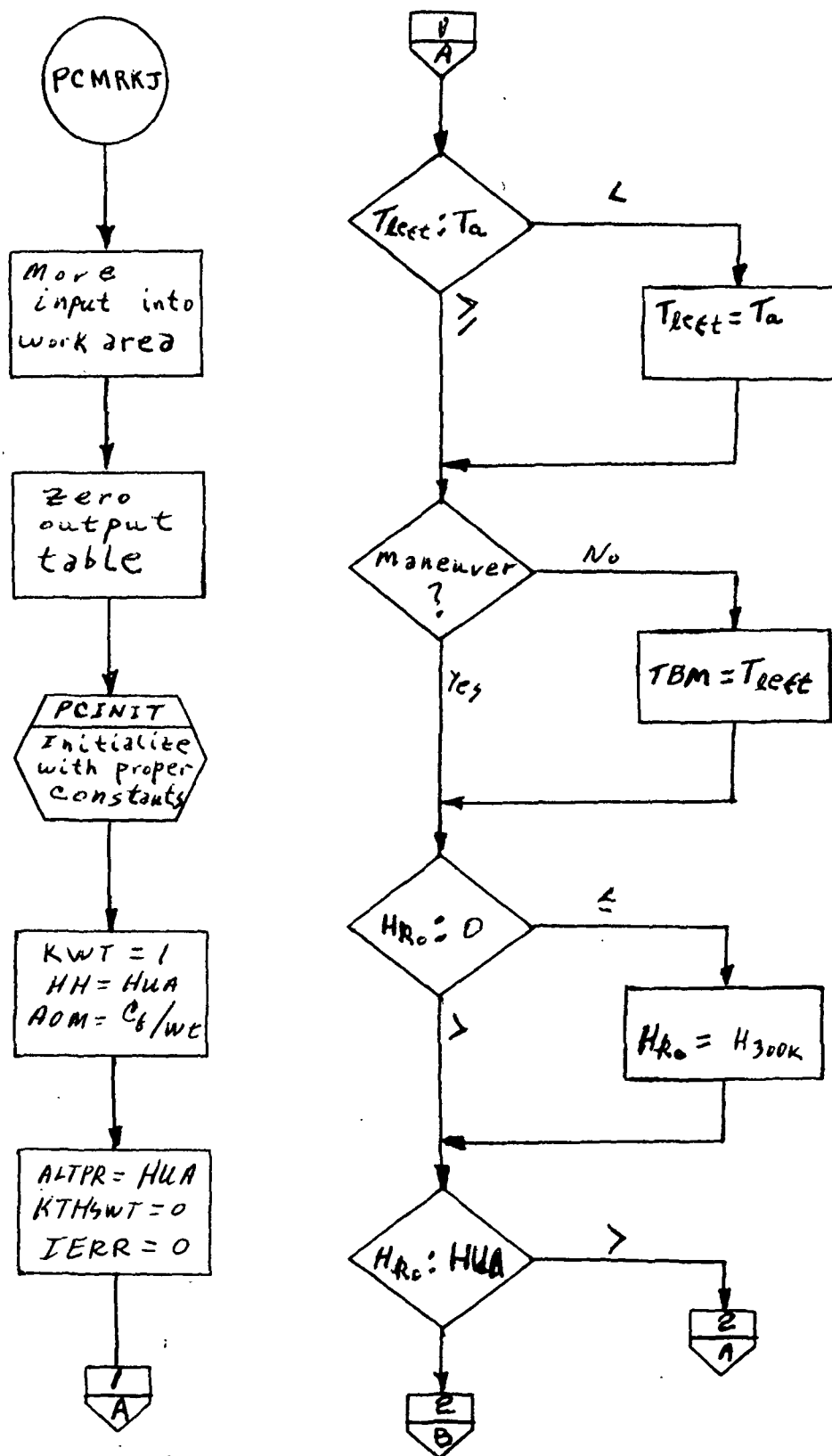


Figure PMMRKJ-1. Flowchart

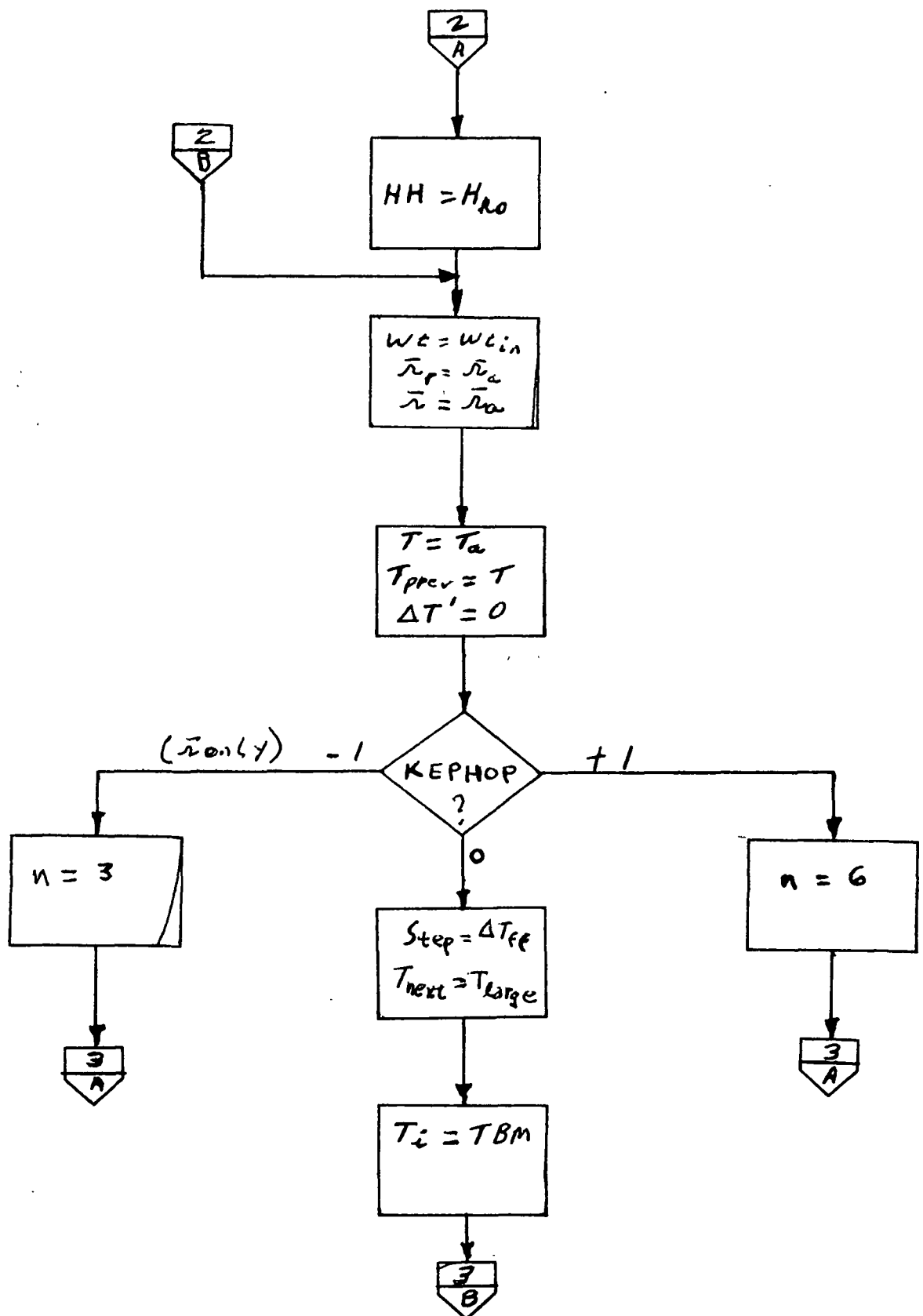


Figure PMMRKJ-1. Flowchart

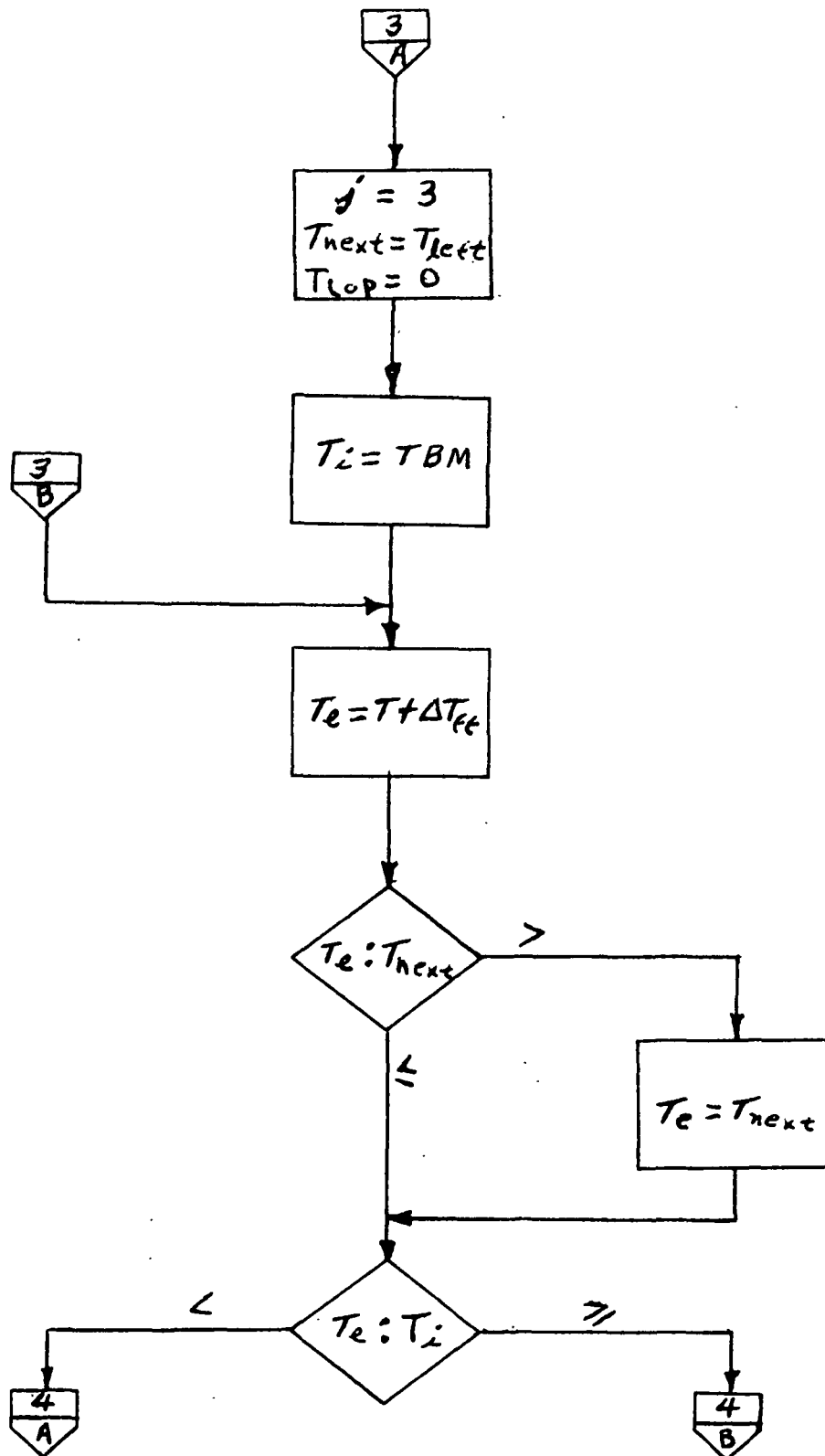


Figure PMMRKJ-1. Flowchart

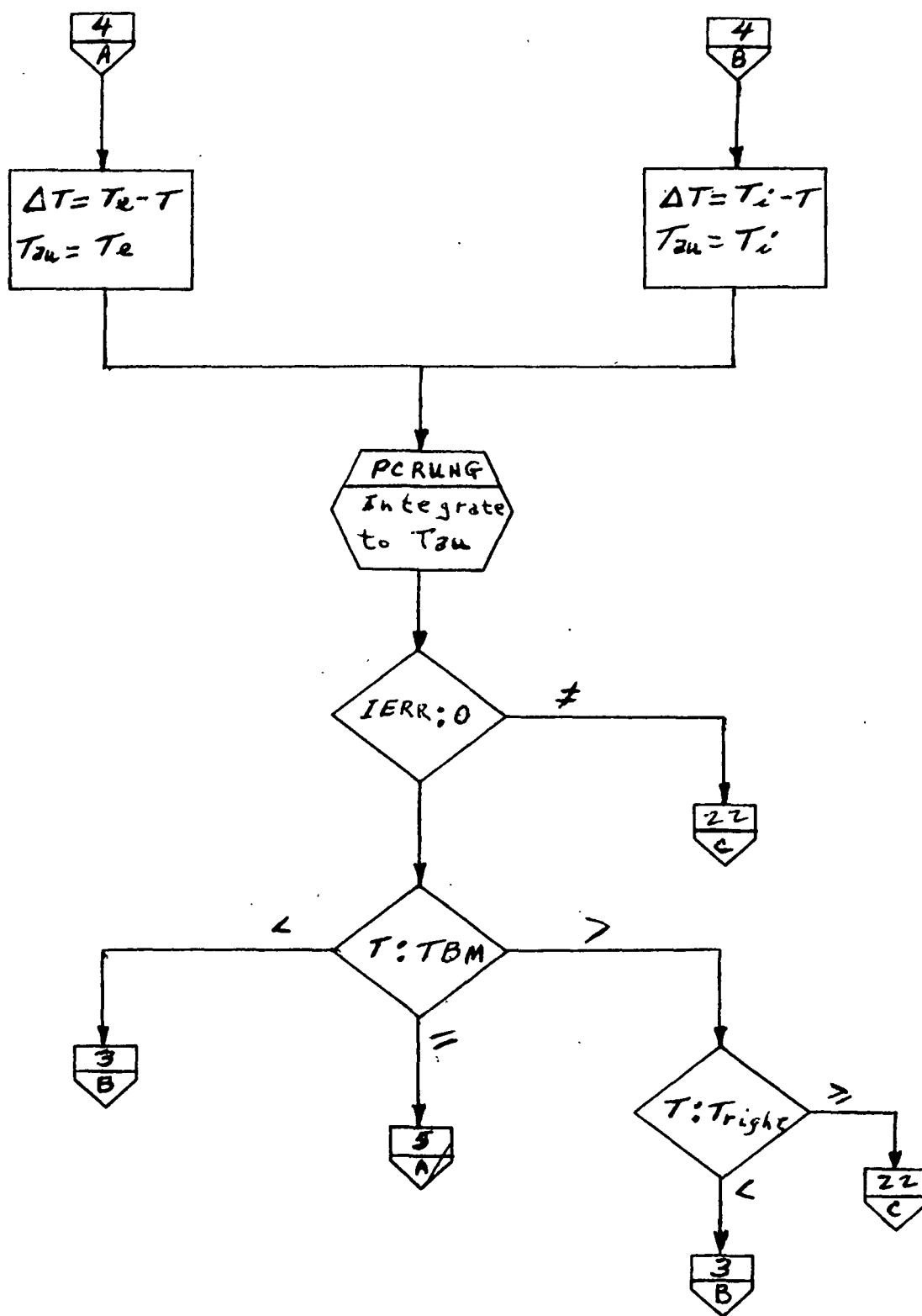


Figure PMMRKJ-1. Flowchart

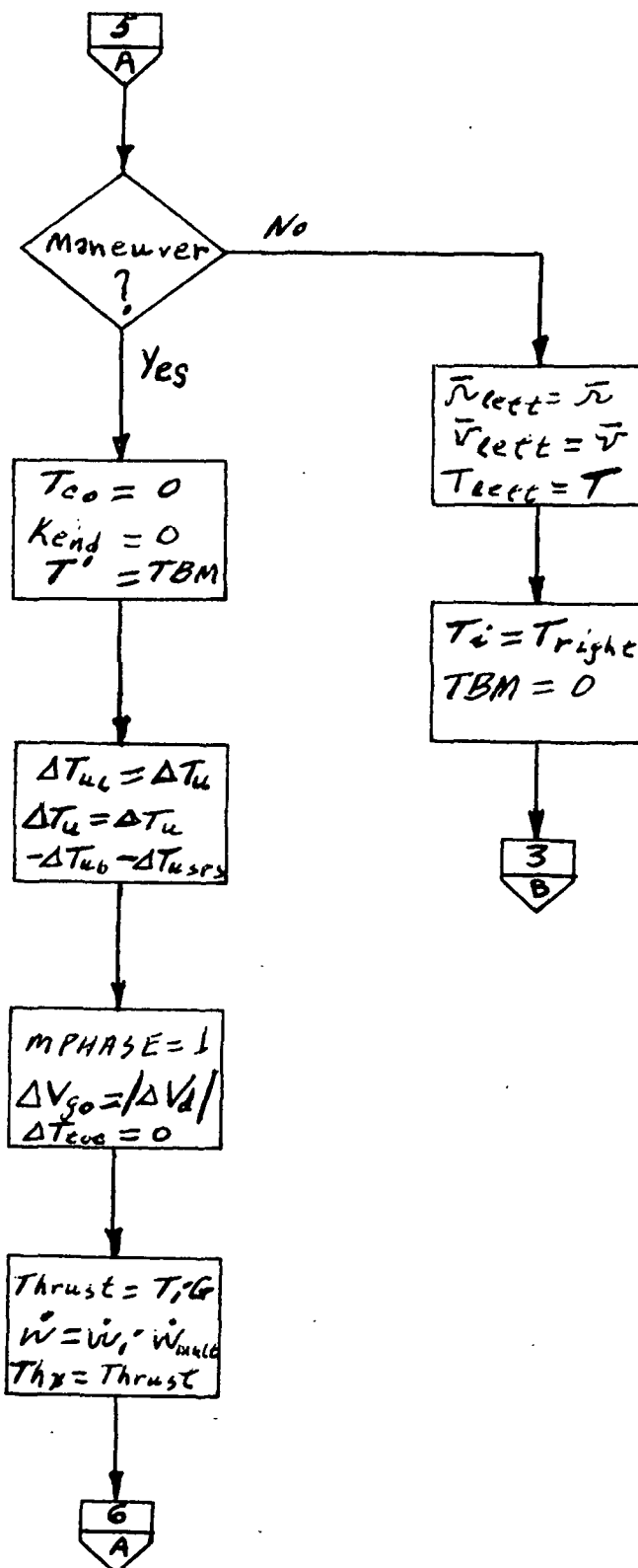


Figure PMMRKJ-1. Flowchart

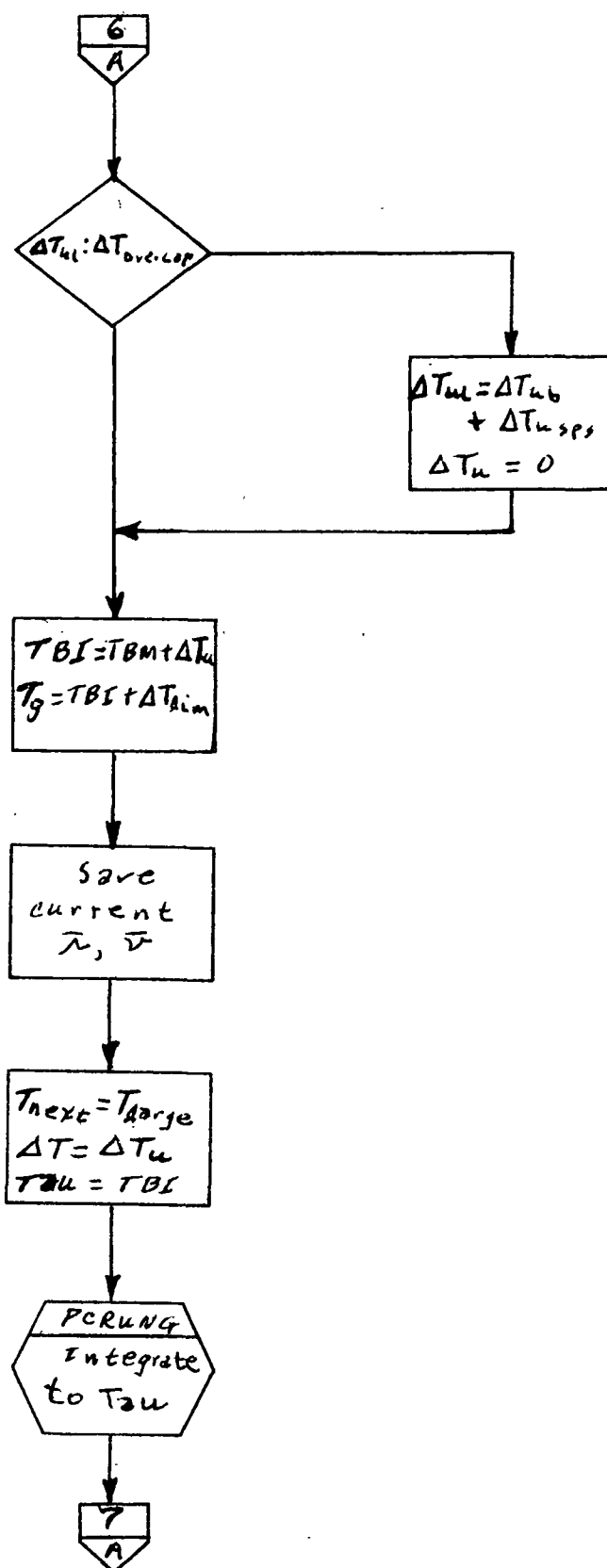


Figure PMMRKJ-1. Flowchart

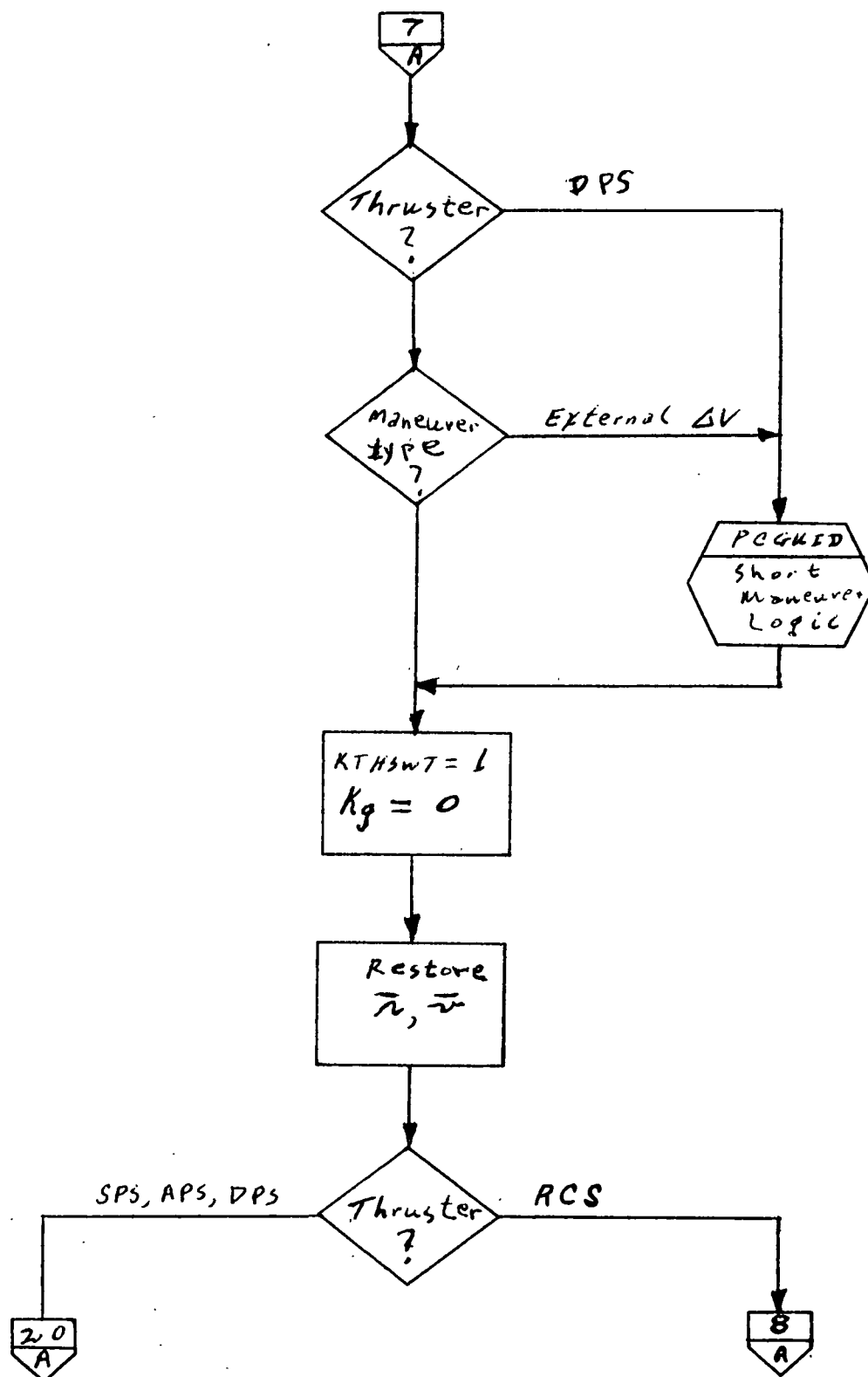


Figure PMMRKJ-1. Flowchart

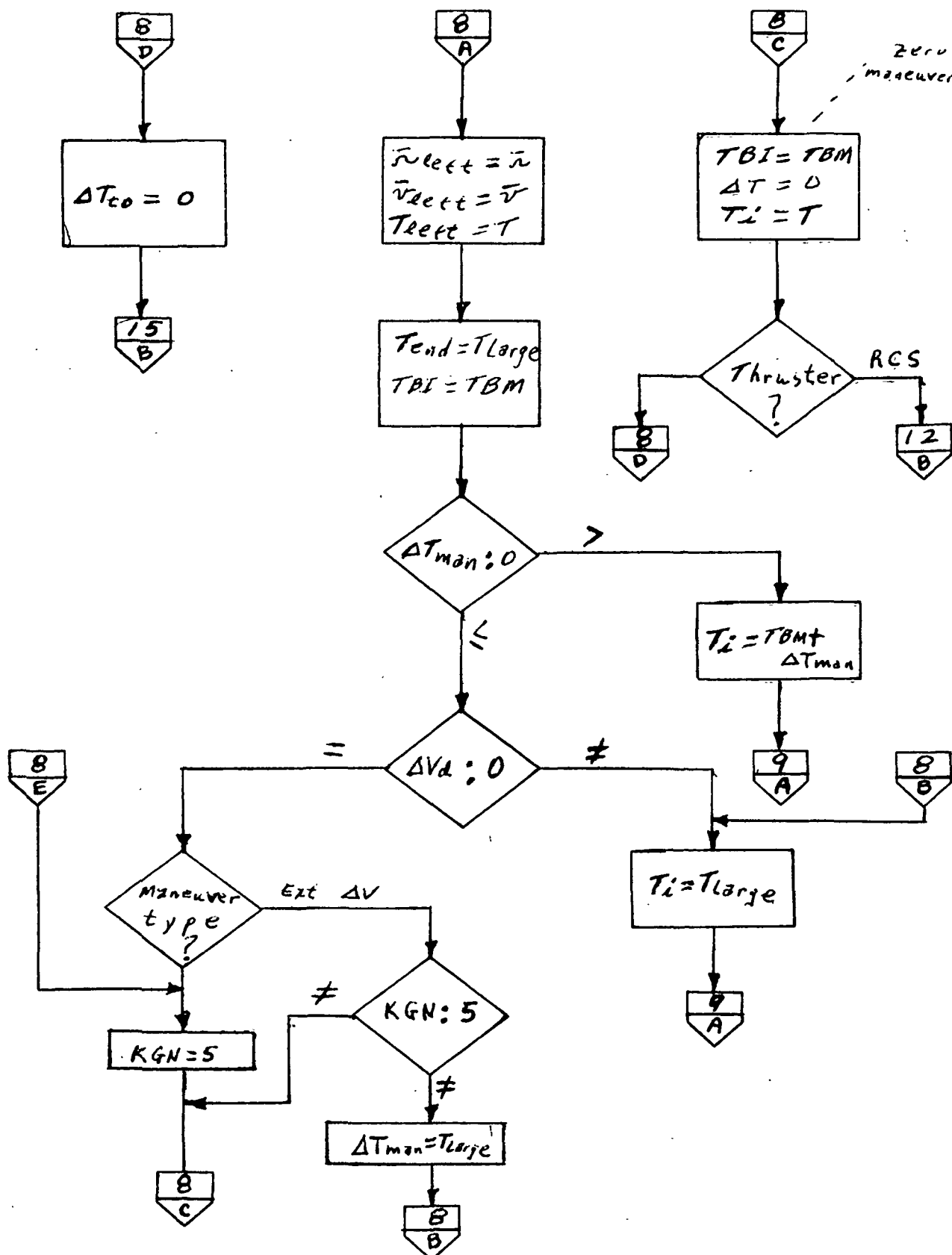


Figure PMMRKJ-1. Flowchart

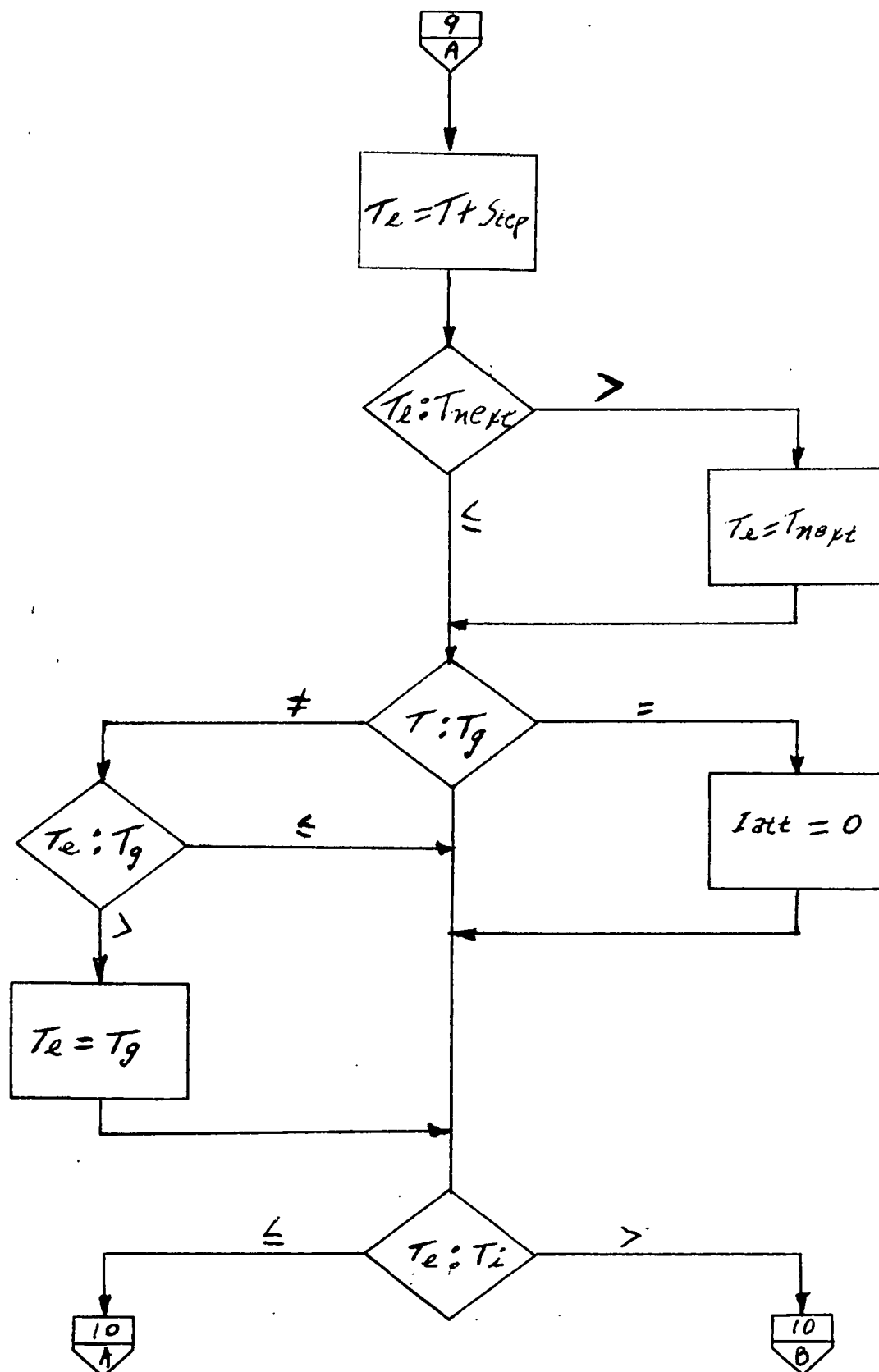


Figure PMMRKJ-1. Flowchart

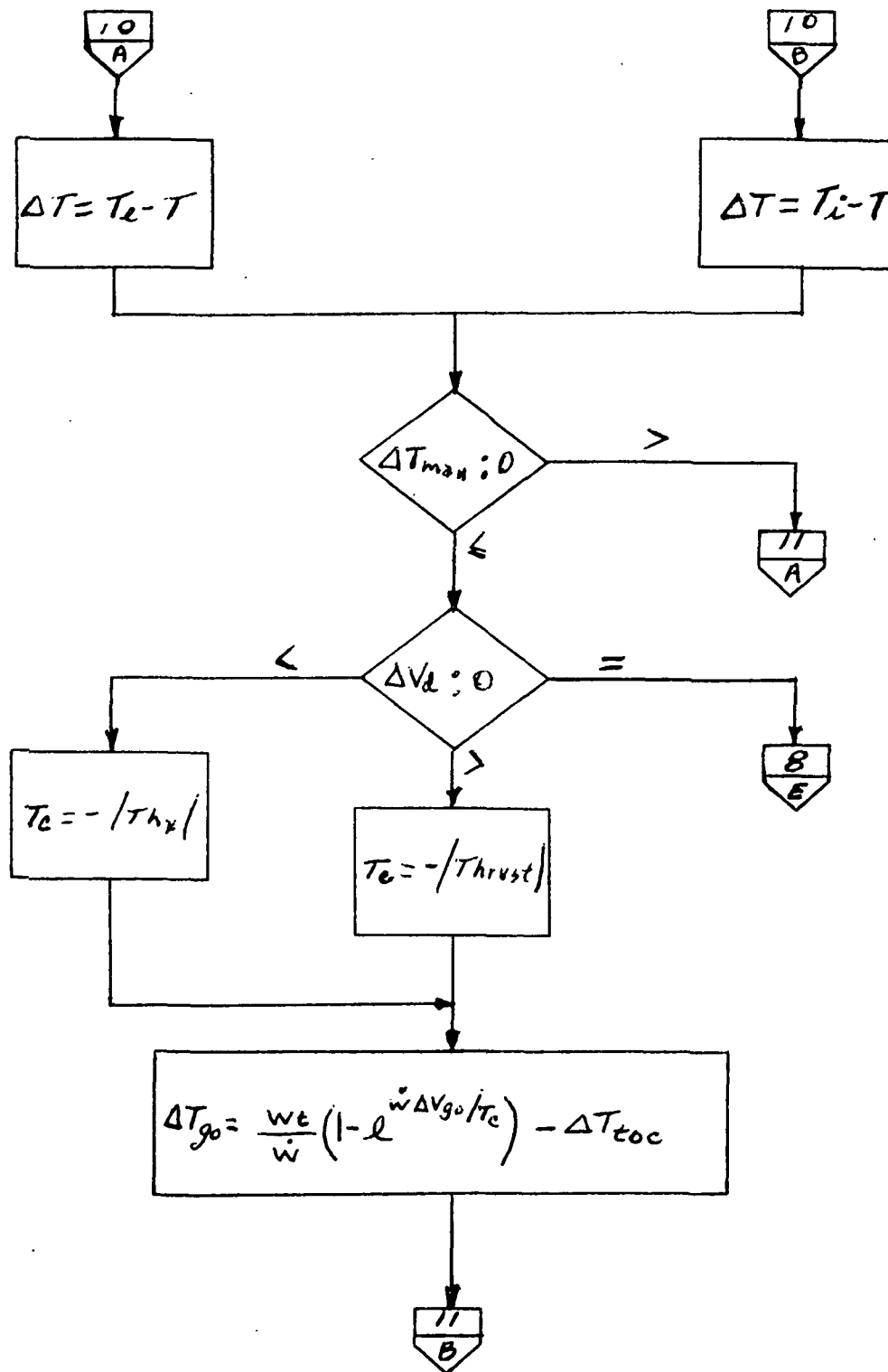


Figure PMMRKJ-1. Flowchart

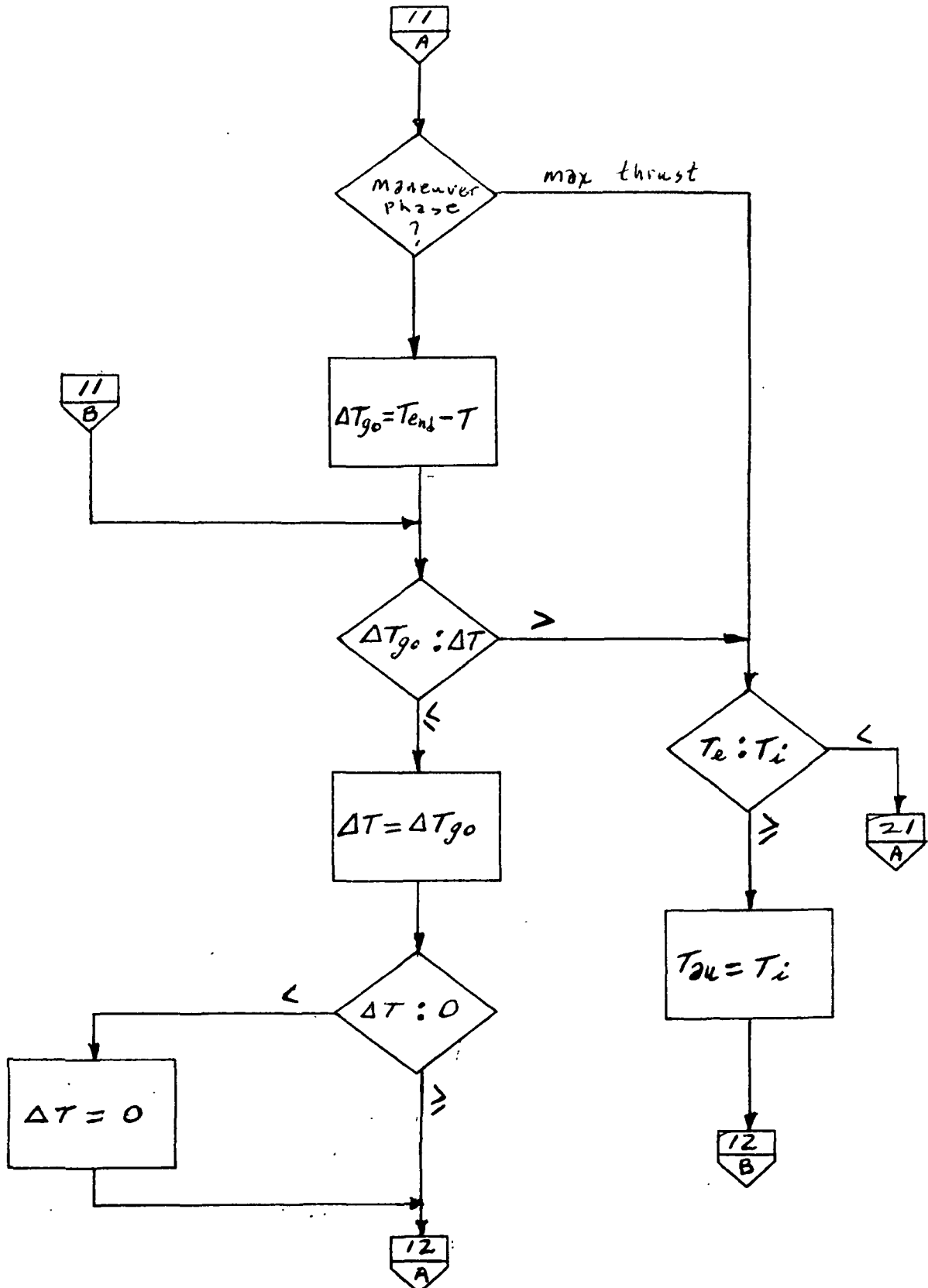


Figure PMMRKJ-1. Flowchart

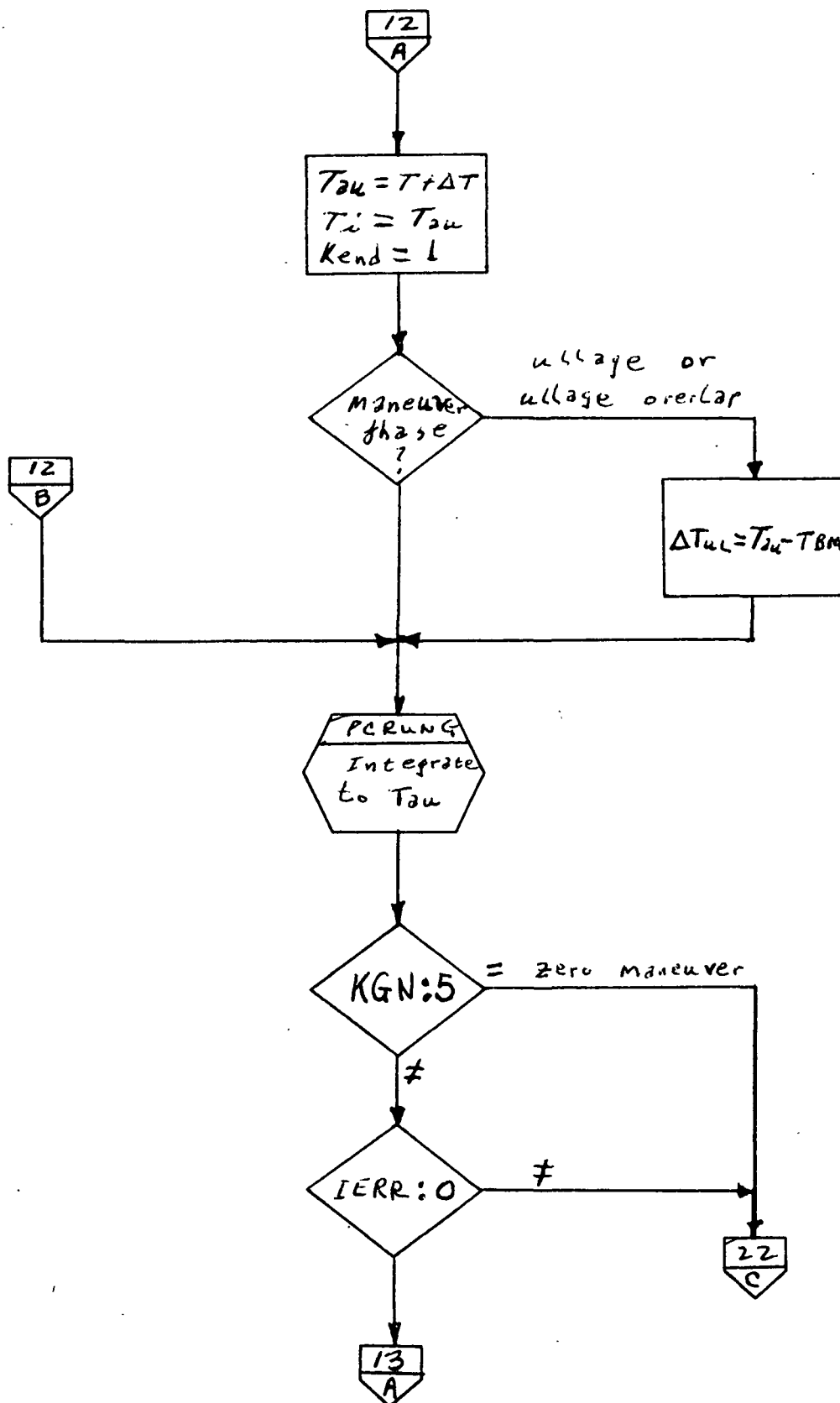


Figure PMMRKJ-1. Flowchart

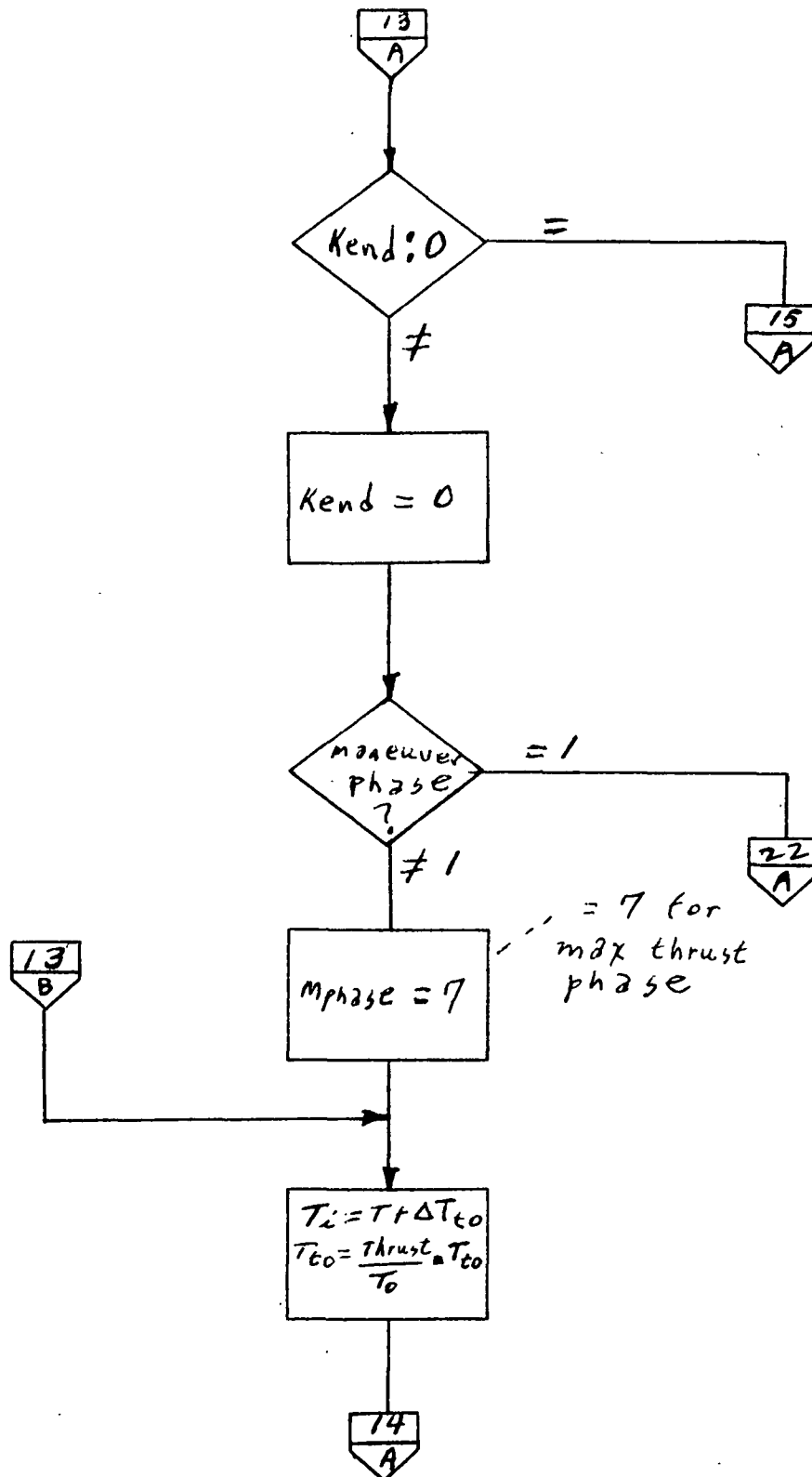


Figure PMMRKJ-1. Flowchart

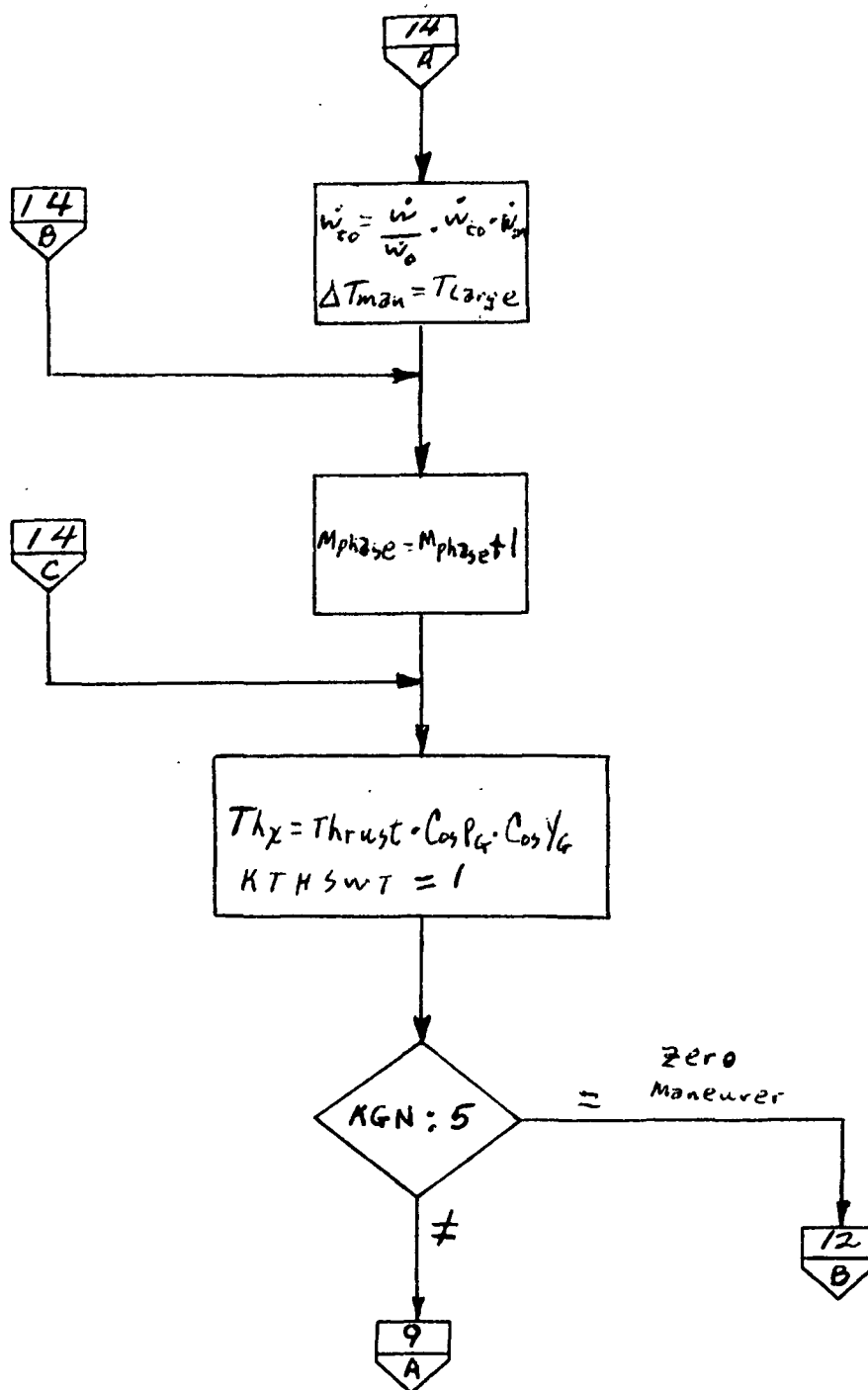


Figure PMMRKJ-1. Flowchart

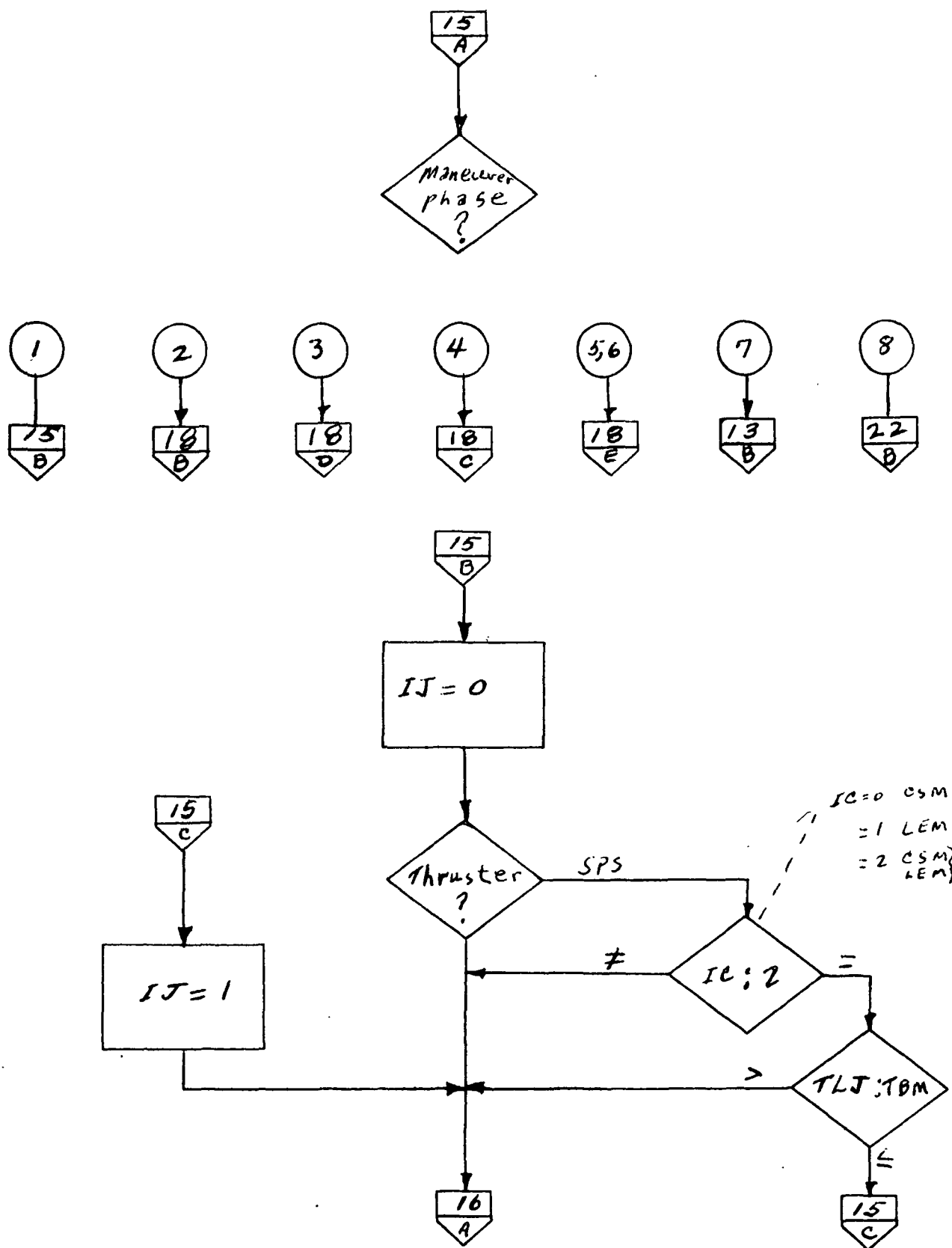


Figure PMMRKJ-1. Flowchart

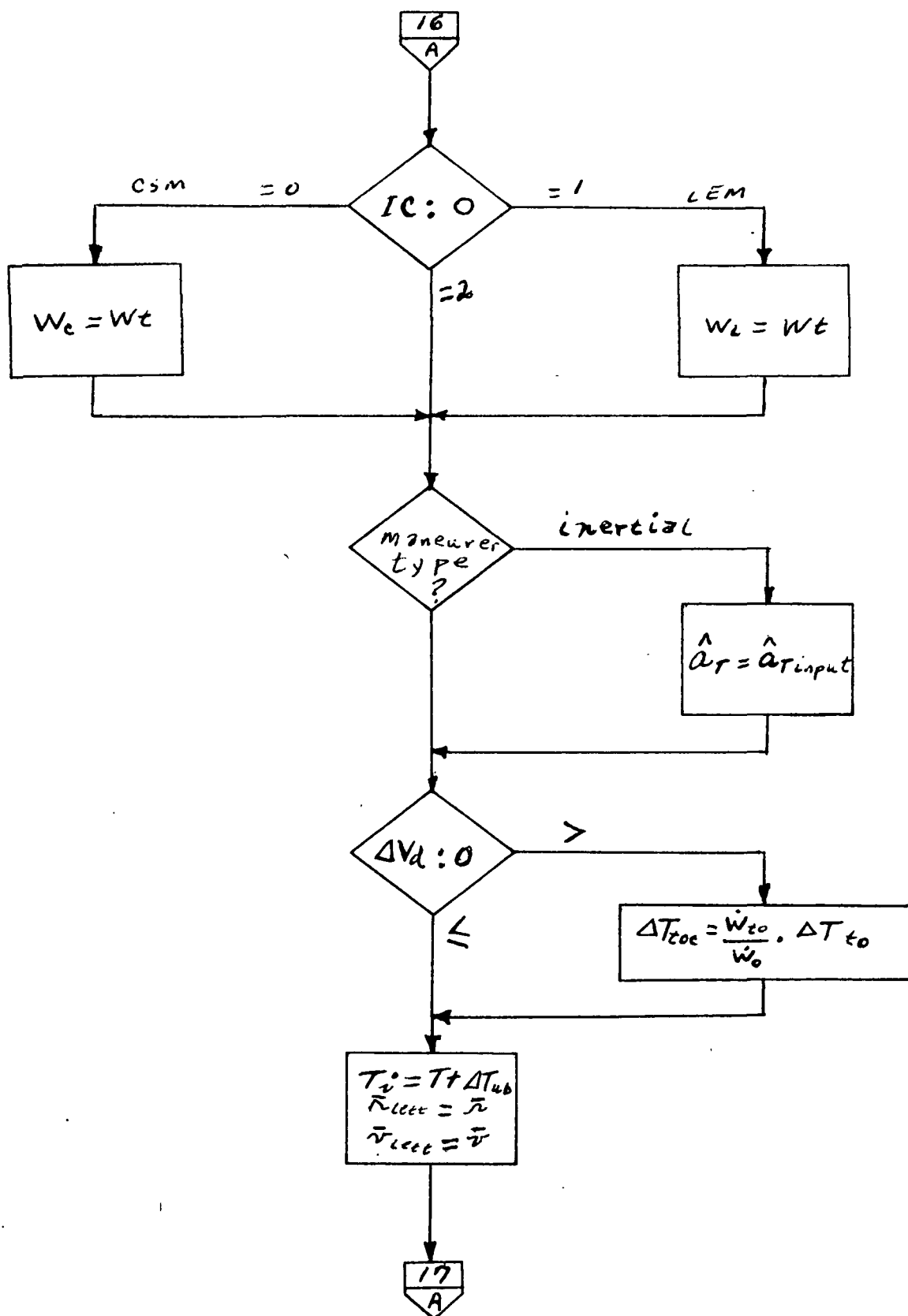


Figure PMMRKJ-1. Flowchart

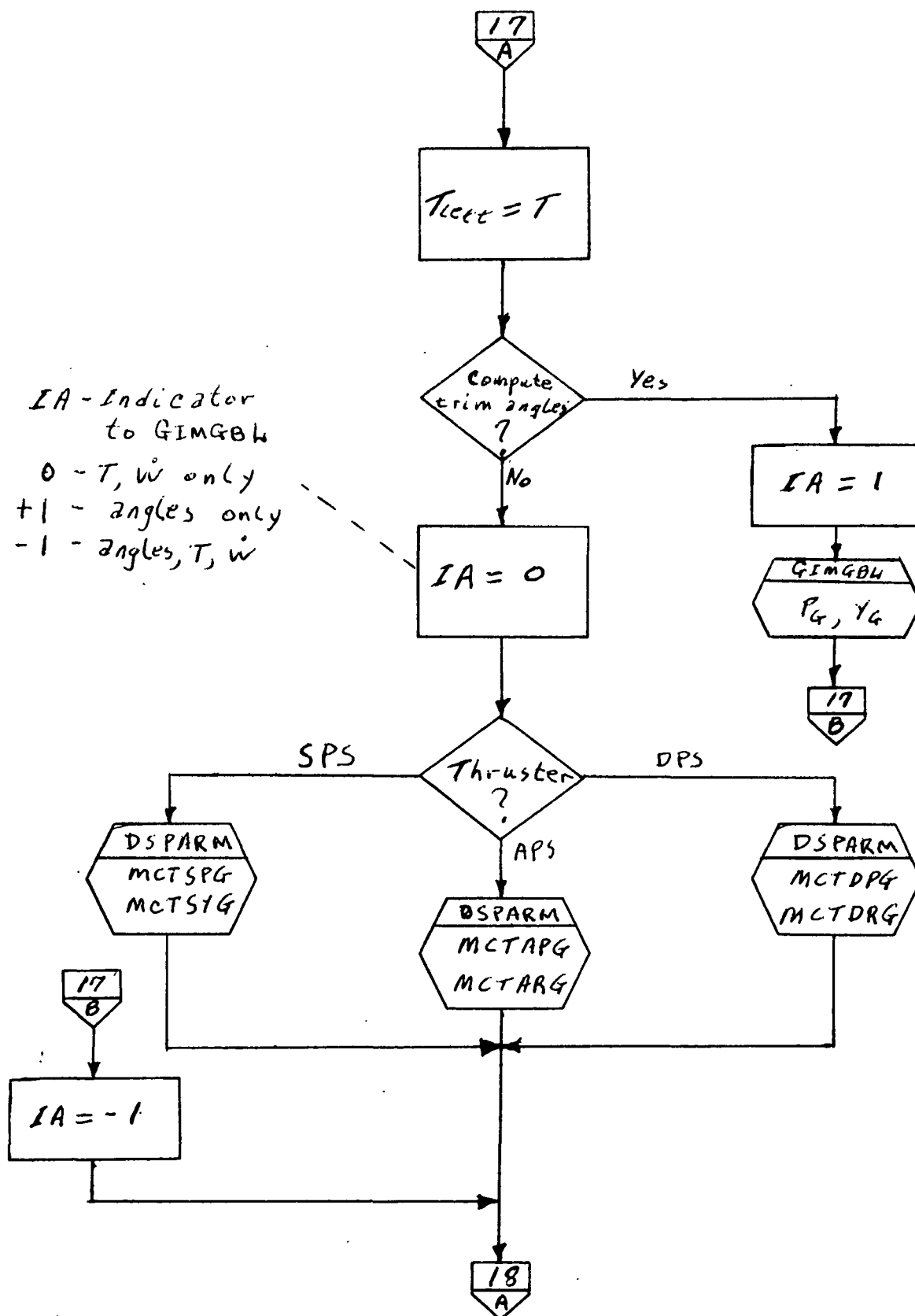


Figure PMMRKJ-1. Flowchart

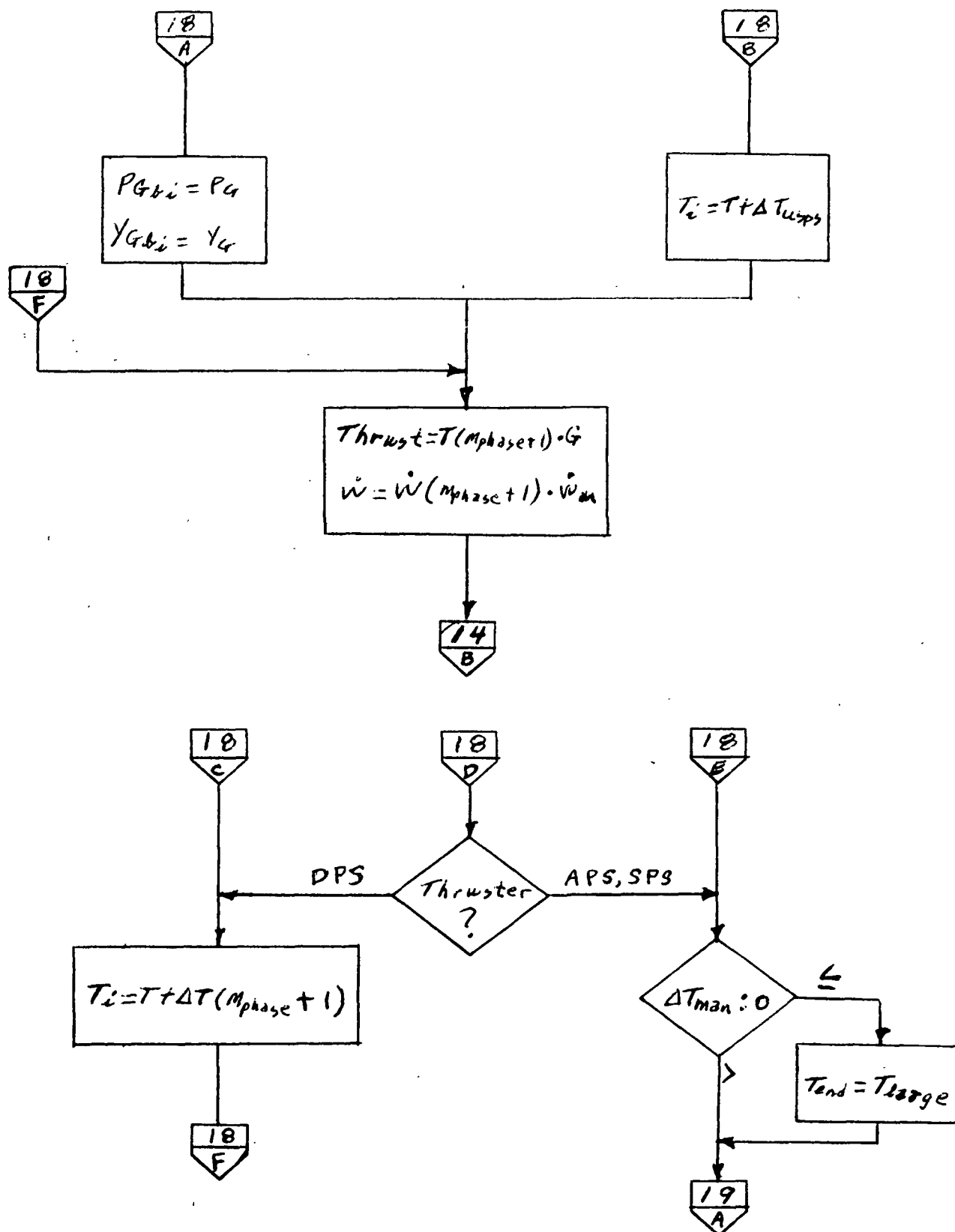


Figure PMMRKJ-1. Flowchart

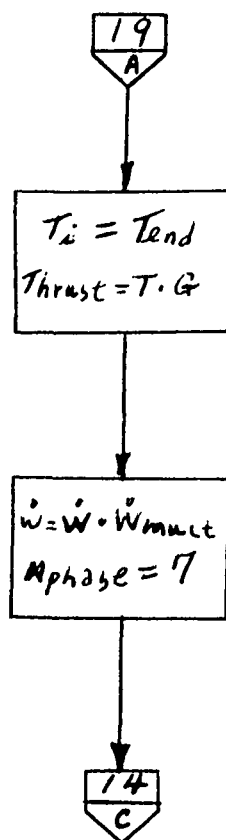


Figure PMMRKJ-1. Flowchart

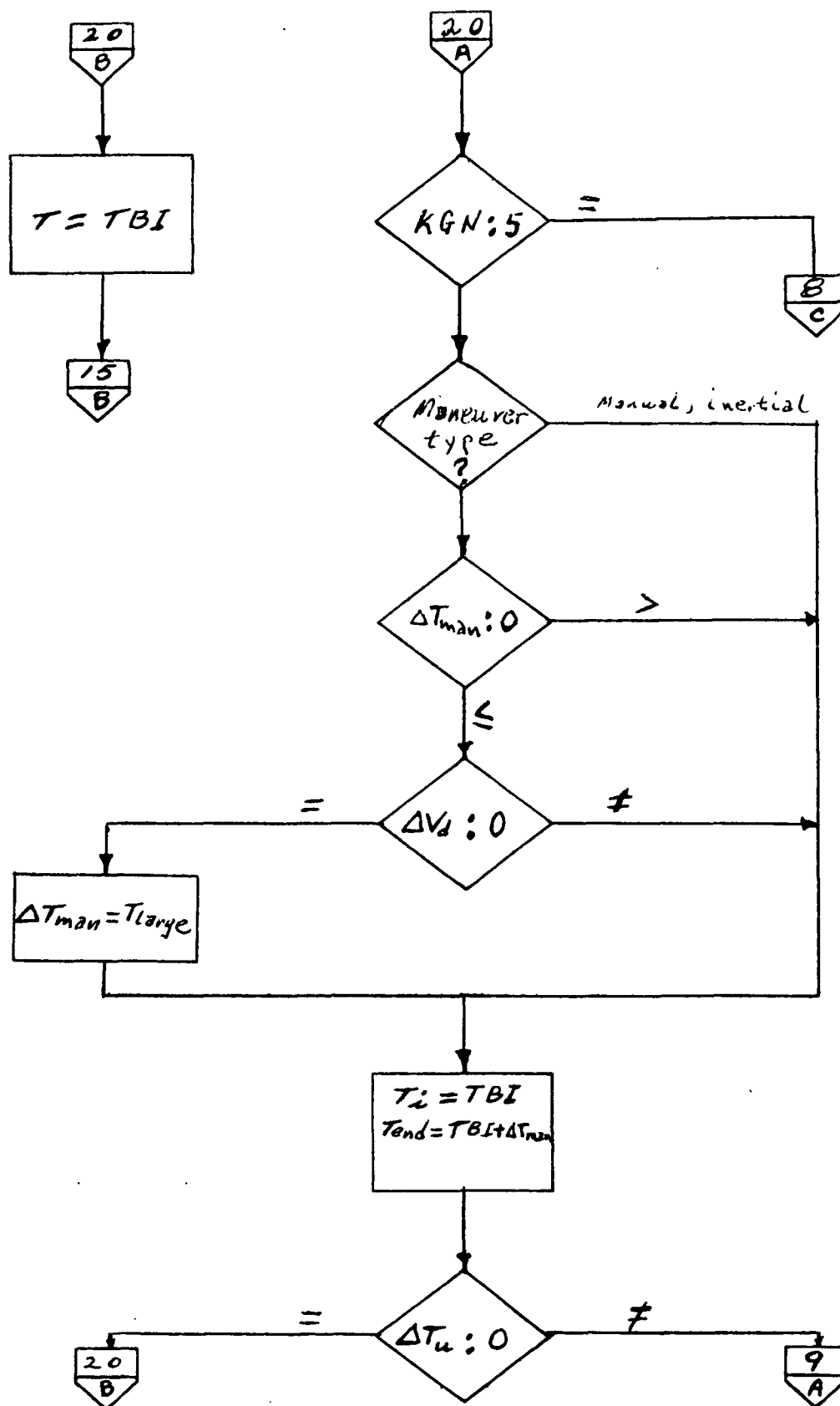


Figure PMMRKJ-1. Flowchart

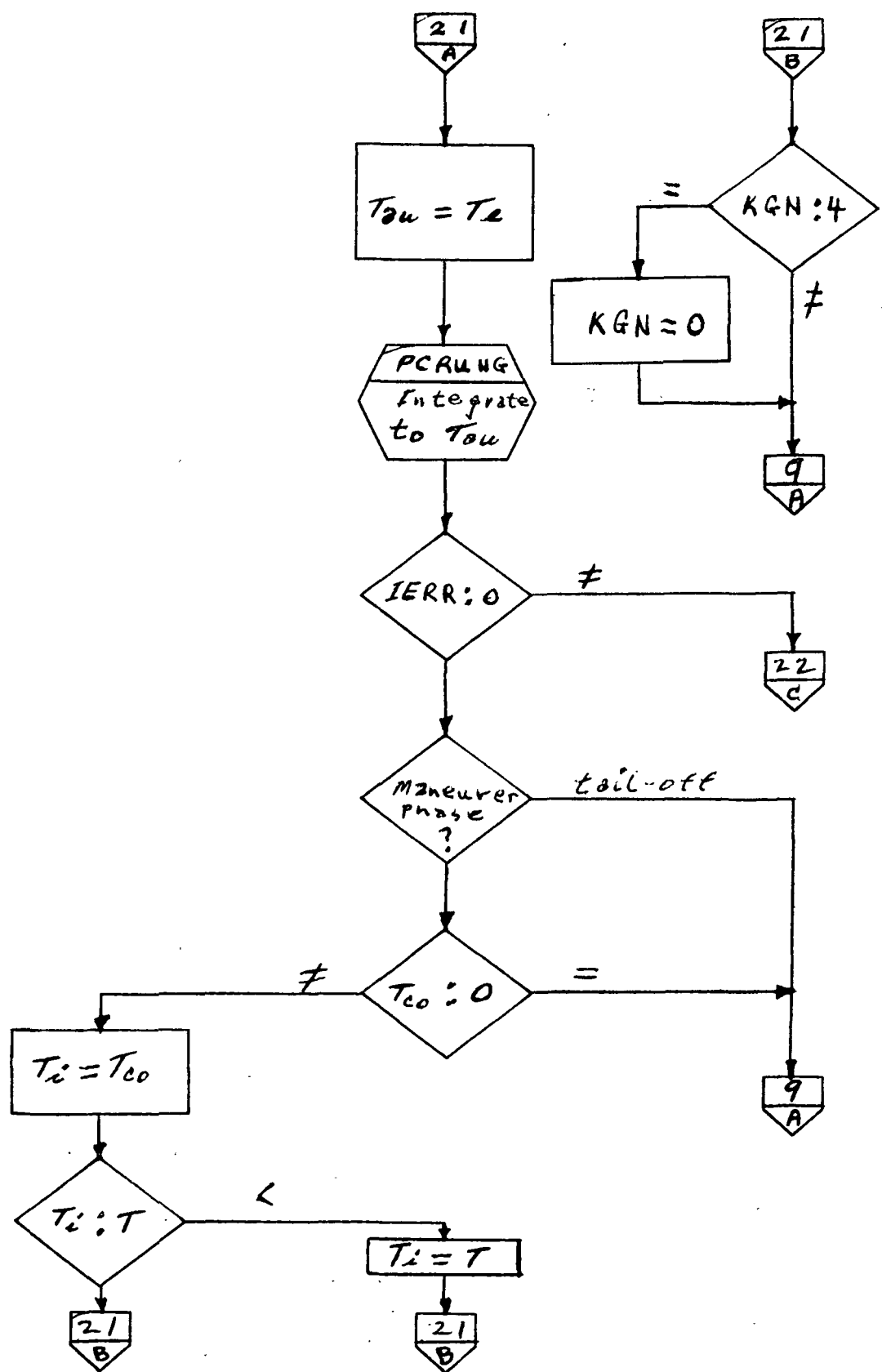


Figure PMMRKJ-1. Flowchart

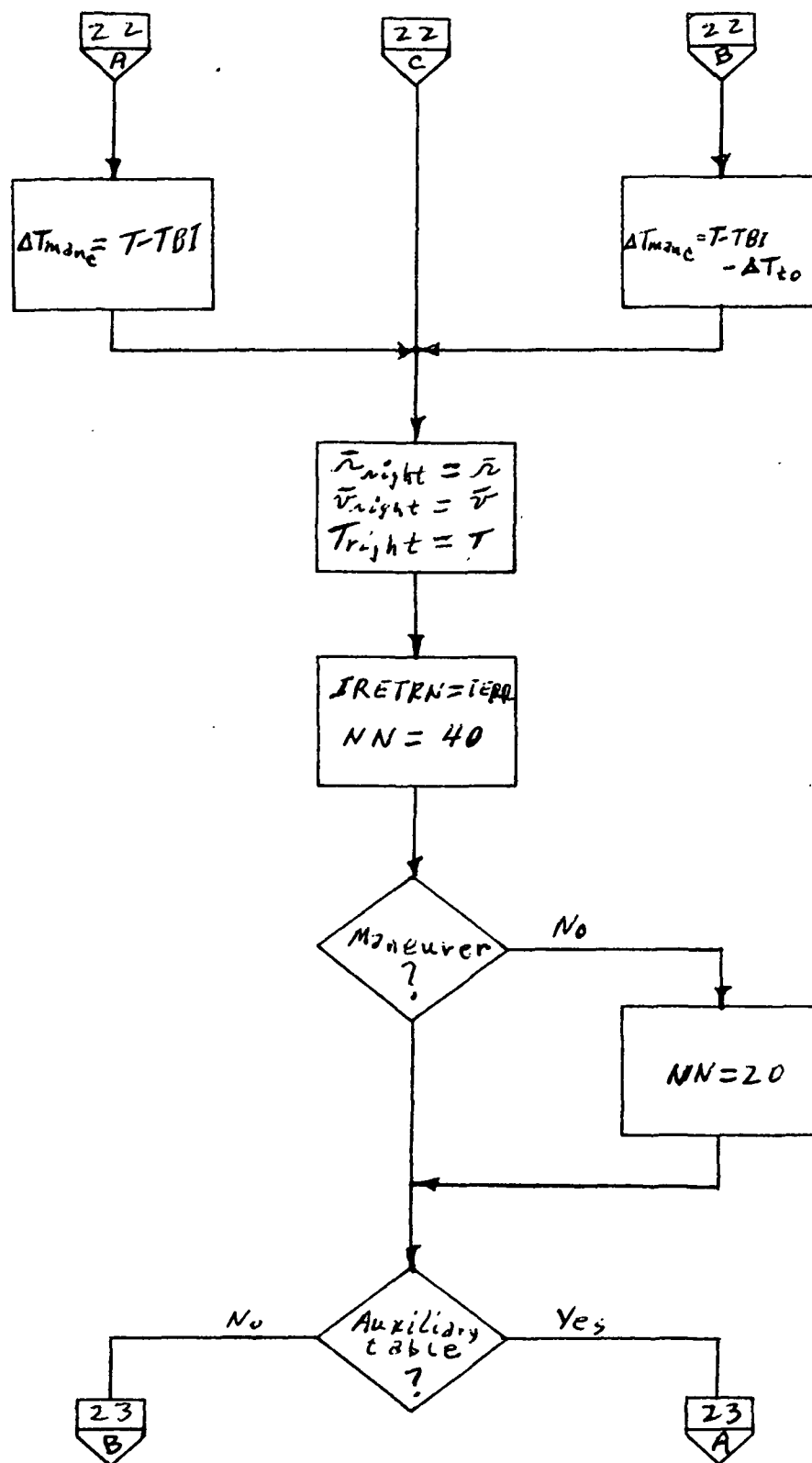


Figure PMMRKJ-1. Flowchart

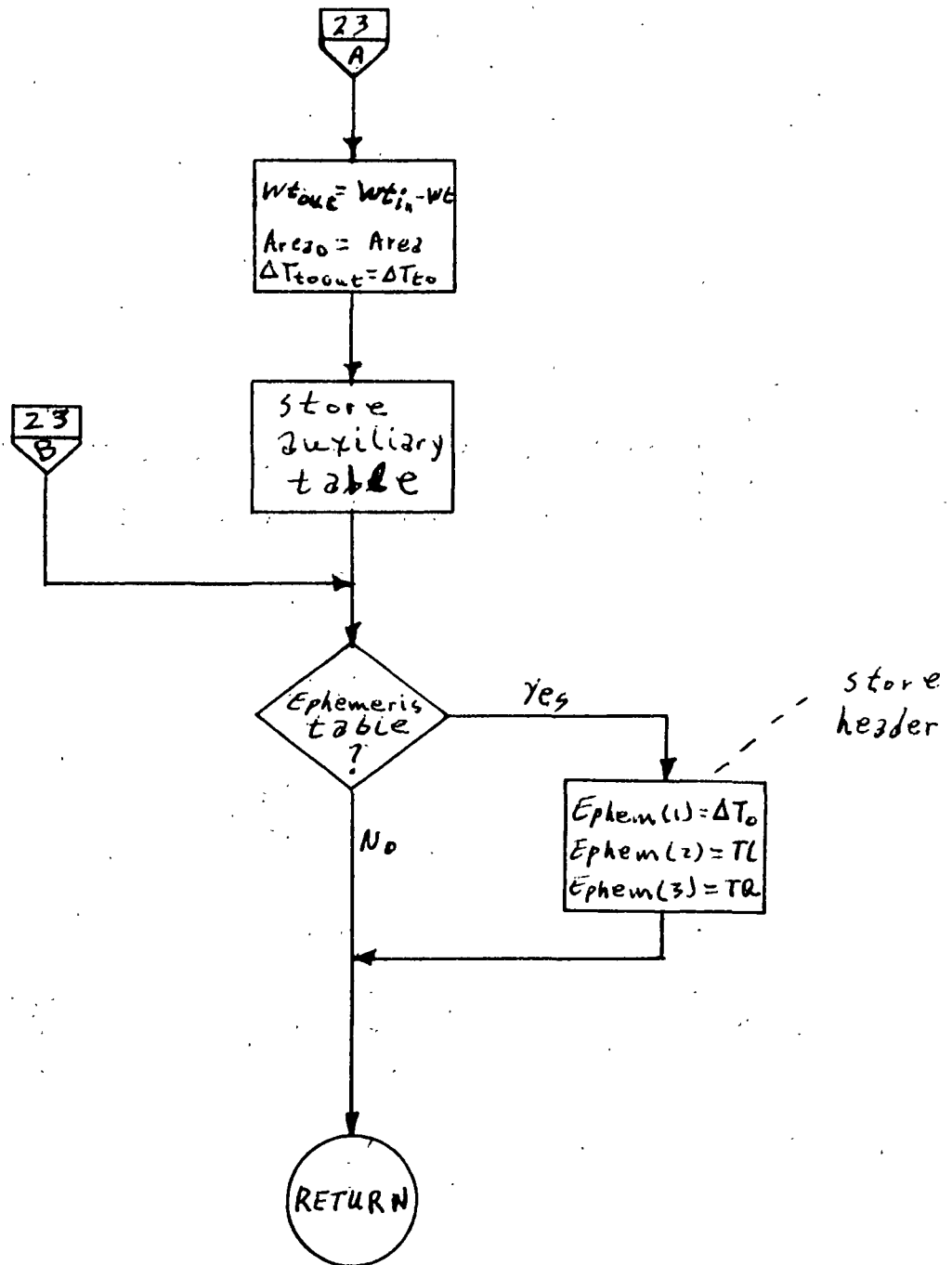


Figure PMMRKJ-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMRKJ

DATE 11/27/67 (EORP)

REV

PAGE 1 of 47

- a. Symbol — PMMRKJ
- b. Program Name — CSM/LM Powered Flight Numerical Integration and Guidance Module
- c. Programming Group — Maneuver Control
- d. Programmer — J. Gardner
- e. Language — FORTRAN
- f. Data Tables Used — None
- g. MED's Used — None
- h. Writeup Completion Date — 10/13/67
- i. Input — Table PMMRKJ-1
- j. Auxiliary Output Table — Table PMMRKJ-2
- k. Ephemeris Table — Table PMMRKJ-3
- l. Table of Weights — Table PMMRKJ-4
- m. System Parameters — Table PMMRKJ-5
- n. Description of Terms — Table PMMRKJ-6
- o. Flowchart — Figure PMMRKJ-1

FUNCTION

Load Module PMMRKJ is a program designed to integrate through periods of powered flight and short spans of free flight. A fourth order Runge-Kutta, Nystrom, numerical integration technique is used to generate an ephemeris table of position and velocity vectors, an auxiliary output table, and a table of weights.

USAGE

- a. Calling Sequence — In assembly language:
LINK EP = PMMRKJ, PARAM=(T, I, E, A, W)
In FORTRAN:
CALL LINK (PMMRKJ, T, I, E, A, W)
where:
T = 60 double-word input table
I = error indicator (half word)
E = output ephemeris table
A = 40-word auxiliary output table for maneuver
A = 20-word auxiliary output table for free flight
W = output table of weights
- b. Inputs — Table PMMRKJ-1
- c. Outputs
 - 1. Table PMMRKJ-2 auxiliary output table
 - 2. Table PMMRKJ-3 ephemeris table
 - 3. Table PMMRKJ-4 table of weights

Approval	Approval
<i>J. Gardner</i>	<i>S.E. James</i>

- d. Other Programs Used — GIMGBL, GLFSAT, GLFDEN, GLMRTM, DEXP, DCOS, DSQRT, DLOG, PCRUNG, PCRDD, PCGUID, PCINIT, DSPARM
- e. Data Restrictions and Limits
 1. All times input must be positive and conform to the following restrictions: anchor time $\leq$ left limit $\leq$ right limit
anchor time $\leq$ time to begin maneuver
 2. The auxiliary output table is initialized to zero; thus, conditional type entries in this table will be zero unless the condition is met.
 3. For RCS maneuvers ΔV desired must be ≥ 0 . That is, a desired ΔV along the X-body axis is not allowed.
- f. Error — Upon exit the error indicator will be set as follows:
 - 0 for no errors
 - 1 for impacted before reaching end conditions
 - 2 for exceeded available fuel
 - 3 for error in guided maneuver definition

If an error condition is met, the $\bar{r}$, $\bar{v}$, and t at the time the error occurred will be stored in words 8-14 of the auxiliary output table.

Table PMMRKJ-1. Input

Word	Byte	Type	Units	Description and Range
1-3	1-8	D	Er	Starting position vector
4-6	1-8	D	Er/hr.	Starting velocity vector
7	1-8	D	hr.	Starting time
8	1-8	D	hr.	Ephemeris output time cycle
9	1-8	D	hr.	Left integration limit
10	1-8	D	hr.	Right integration limit
11	1-8	D	hr.	Kickout altitude (if ≤ 0 , the system parameter value will be used)
12	1-8	D	-	Weight loss rate multiplier (should normally be 1.0)
13	1-8	D	hr.	Time of LM jettison of descent stage
14	1-8	D		Open for expansion
15	1-8	D		Density multiplier (should normally be 1.0)
16	1-8	D	ft ²	Vehicle cross sectional area
17	1-8	D	lbs.	Weight of specified configuration
18	1,2	H		Ephemeris output option: -1 for $\bar{r}$ only 0 for none +1 for $\bar{r}$ and $\bar{v}$ +2 for $\bar{r}$ and $\bar{v}$ and table of weights

Table PMMRKJ-1. Input (Cont'd.)

<u>Word</u>	<u>Byte</u>	<u>Type</u>	<u>Units</u>	<u>Description and Range</u>
18	3, 4	H		Auxiliary output option: 1 for table desired 0 for table not desired
18	7, 8			Open for expansion
19	1-8			CSM WT
20	1-8			LM WT
21	1, 2	H		Attitude code: 0 - no maneuver ±1 - fixed inertial ±2 - manual holding body orientation invariant ±3 - Lambert Guidance* ±4 - external ΔV (primary) ±5 - external ΔV (LM, AGS) negative sign indicates initial attitudes are siren.
21	3, 4	H		Thruster code: 1 - CSMRCS+X (2 thrusters) 2 - CSMRCS+X (4 thrusters) 3 - CSMRCS-X (2 thrusters) 4 - CSMRCS-X (4 thrusters) 16 - S-IVB blowdown vent 17 - LEMRCS+X (2 thrusters) 18 - LEMRCS+X (4 thrusters) 19 - LEMRCS-X (2 thrusters) 20 - LEMRCS-X (4 thrusters) 33 - CSMSPS 34 - LEMAPS 35 - LEMDPS
21	5, 6	H		Ullage option: 0 - two thrusters 1 - four thrusters
21	7, 8			Open for expansion
22	1, 2	H		Configuration code at maneuver initiation: 0 for CSM 1 for LM 2 for CSM and LM
23	1-8	D	hr.	Time to begin maneuver (first phase)
24	1-8	D	hr.	Δt of ullage (=0 for RCS)
25	1-8	D	hr.	Δt of maneuver (no pure ullage or tailoff) (=0 if ΔV is given)

*Lambert Guidance not a requirement for EORP.

Table PMMRKJ-1. Input (Cont'd.)

Word	Byte	Type	Units	Description and Range
26	1-8	D	Er/hr.	ΔV desired (if < 0 it is ΔV along X-body excluding tailoff)
*27				$\left. \begin{matrix} X \\ Y \\ Z \end{matrix} \right\} \begin{matrix} A \\ T \end{matrix}$ RTCC inertial coordinates of unit thrust vector at burn initiate
28				
29				
**30	1-8	D		$\left. \begin{matrix} X \\ Y \\ Z \end{matrix} \right\} \begin{matrix} A \\ B \end{matrix}$ RTCC inertial coordinates of the unit vectors in the direction of the vehicle body X, Y, Z axes, respectively
31	1-8	D		
32	1-8	D		
33	1-8	D		$\left. \begin{matrix} X \\ Y \\ Z \end{matrix} \right\} \begin{matrix} A \\ B \end{matrix}$ RTCC inertial coordinates of the unit vectors in the direction of the vehicle body X, Y, Z axes, respectively
34	1-8	D		
35	1-8	D		
36	1-8	D		$\left. \begin{matrix} X \\ Y \\ Z \end{matrix} \right\} \begin{matrix} A \\ B \end{matrix}$ axes, respectively
37	1-8	D		
38	1-8	D		
39	1-8	D	hr.	Δt at 10% thrust level for DPS thruster: Set negative if DPS. Short burn test is to be ignored.
40	1-2	H		Maneuvering vehicle code: 1 - for CSM 2 - for S-IVB 3 - for LM
40	3-4			Thruster trim angle indicator: -1 if angles to be computed +1 if system parameters are to be used
40	5-8			Open for expansion
41	1-8			Docking angle
42	1-8			Open for expansion
43	1-8			Open for expansion
44	1-8	D	Er/hr.	$\left. \begin{matrix} V_X \\ V_Y \\ V_Z \end{matrix} \right\}$ External $\overline{\Delta V}$ in RTCC inertial coordinate system
45	1-8	D	Er/hr.	
46	1-8	D	Er/hr.	
47-60	1-8	D		Open for expansion

*Words 27, 28, 29 need be set only for inertial maneuvers.

**Words 30, 31, 32 must be set for all multi-phase maneuvers.

Table PMMRKJ-2. Auxiliary Output

Word	Byte	Units	Description and Range
1-3	1-8	Er	For no maneuvers: $\bar{r}$, $\bar{v}$, t at t_L
4-6	1-8	Er/hr.	For maneuvers: $\bar{r}$, $\bar{v}$, t at burn initiate
7	1-8	hr.	
8-10	1-8	Er	For no maneuvers: $\bar{r}$, $\bar{v}$, t at t_R
11-13	1-8	Er/hr.	
14	1-8	hr.	For maneuvers: $\bar{r}$, $\bar{v}$, t at end of maneuver
15	1-8	hr.	Time at 400K feet
16	1-8	rad.	Geodetic latitude at 400K feet
17	1-8	rad.	Longitude at 400K feet
18	1-8	hr.	Time at 25K feet
19	1-8	rad.	Geodetic latitude at 25K feet
20	1-8	rad.	Longitude at 25K feet
21	1-8	Er/hr.	V_{gx} } Velocity to be gained at G and N
22	1-8	Er/hr.	V_{gy} } initiate in inertial coordinates
23	1-8	Er/hr.	V_{gz} }
24	1-8	Er/hr.	ΔV of maneuver along X-body axis excluding tailoff
25	1-8	Er/hr.	Total ΔV of tailoff
26	1-8	Er/hr.	Total ΔV of maneuver
27	1-8	hr.	Δt of maneuver excluding tailoff
28	1-8	lbs.	Weight at end of maneuver
29	1-8		Weight loss during maneuver
30	1-8		Area
31	1-8		$\hat{a}_T$ } inertial coordinates of unit
32	1-8		thrust vector at burn initiate
33	1-8		
34	1-8	Er/hr.	ΔV of tailoff along X-body axis (does not apply for RCS maneuvers)
35	1-8	rad.	Pitch gimbal at burn initiate
36	1-8	rad.	Yaw gimbal at burn initiate
37	1-8		ΔV of pure ullage
38	1-8	hrs.	Δt of tailoff
39	1-8		Weight at main engine on
40	1-8		Open for expansion

Table PMMRKJ-3. Ephemeris

<u>Word</u>	<u>Byte</u>	<u>Units</u>	<u>Description and Range</u>
1	1-8	hr.	Timer interval at which ephemeris is stored
2	1-8	hr.	Time associated with first entry in ephemeris
3	1-8	hr.	Time associated with last entry in ephemeris
4-6	1-8	Er	Position vector - 1st entry
7-9	1-8	Er/hr.	Velocity vector - 1st entry
10-12	1-8	Er	Position vector - 2nd entry
13-15	1-8	Er/hr.	Velocity vector - 2nd entry
			Etc.

Table PMMRKJ-4. Table of Weights

<u>Word</u>	<u>Byte</u>	<u>Units</u>	<u>Description and Range</u>
1	1-8	lbs.	Weight - 1st entry
2	1-8	lbs.	Weight - 2nd entry
			Etc.

METHOD

This program integrates the Newtonian equation of motion,

$$\ddot{\vec{r}} = - \frac{\partial V}{\partial \vec{r}} + \vec{F},$$

where

$$\frac{-\partial V}{\partial \vec{r}} = \frac{-\mu \vec{r}}{r^3} \left[1 + J_2 \left(\frac{a}{r} \right)^2 \left(1 - 5 \left(\frac{z}{i} \right)^2 \right) \right] - \frac{2\mu Z}{r^5} J_2 a^2 \hat{Z}$$

where $\hat{Z} = (0, 0, 1)$

$$-\frac{\partial V}{\partial X} = \frac{-\mu X}{r^3} \left[1 + J_2 \left(\frac{a}{r} \right)^2 \left(1 - 5 \left(\frac{Z}{r} \right)^2 \right) \right]$$

$$-\frac{\partial V}{\partial Y} = \frac{-\mu Y}{r^3} \left[1 + J_2 (a/r)^2 (1 - 5 (Z/r)^2) \right]$$

$$-\frac{\partial V}{\partial Z} = \frac{-\mu Z}{r^3} \left[1 + J_2 (a/r)^2 (3 - 5 (Z/r)^2) \right]$$

where $J_2 = (1.62345) (10)^{-3}$ = second harmonic coefficient,
 a = Earth's equatorial radius
 μ = gravitational constant times mass of Earth.

For free-flight integration, $\bar{F}$ is given by

$$\bar{F} = \bar{D}$$

where D = atmospheric drag acceleration.

The three components of drag acceleration are given by:

$$D_x = -1/2 \rho \frac{A}{M} C_D V_r \dot{X}_r,$$

$$D_y = -1/2 \rho \frac{A}{M} C_D V_r \dot{Y}_r,$$

$$D_z = -1/2 \rho \frac{A}{M} C_D V_r \dot{Z}_r,$$

where $\dot{X}_r = \dot{X} + Yw$

$$\dot{Y}_r = \dot{Y} - Xw$$

$$\dot{Z}_r = \dot{Z}$$

$$V_r = (\dot{X}_r^2 + \dot{Y}_r^2 + \dot{Z}_r^2)^{1/2},$$

w = Earth's rotational velocity,

A = cross sectional area,

M = mass

C_d = coefficient of atmospheric drag,

ρ = atmospheric density.

The thrust acceleration vector is given by:

$$\overline{T} = \frac{T_o G}{W} \hat{A}_T$$

where T_o = thrust level,

W = weight,

G = acceleration constant,

$\hat{A}_T$ = unit thrust vector

The maneuver thrust acceleration vector is a function of the engine, maneuver phase, and maneuver type. For the LM RCS, and CSM RCS engines the thrust level is constant throughout the burn. The unit thrust vector for RCS maneuvers is determined as follows:

Fixed Inertial

The unit thrust vector is input and is held throughout the maneuver.

Manual

The unit thrust vector is given by $\hat{A}_T = \hat{X}_b$ where $\hat{X}_b$ = inertial coordinates of X body axis

External $\Delta \bar{V}$

When the initial attitudes are given, the initial unit thrust vector is given by

$$\hat{A}_T = \hat{X}_b$$

Steering is suppressed for two computational cycles (4 seconds). At this time the desired unit thrust vectors for the upcoming cycle are given by

$$\hat{A}_{TR} = \text{Unit}(\bar{V}_g)$$

where

$$\bar{V}_g = \text{input external } \Delta \bar{V}$$



$\hat{A}_{TR}$ = unit thrust vector at the end of current computational cycle

$$\hat{A}_{TL} = \hat{A}_T$$

$$\hat{A}_{TM} = \text{unit} (\hat{A}_{TL} + \hat{A}_{TR})$$

The unit thrust vector for the upcoming computational cycle is then given by:

$$\hat{A}_T = \hat{A}_{TL} \text{ for } k_1,$$

$$\hat{A}_T = \hat{A}_{TM} \text{ for } k_2 \text{ and } k_3,$$

$$\hat{A}_T = \hat{A}_{TR} \text{ for } k_4$$

When the initial attitudes are not given, the initial unit thrust vector is given by $\hat{A}_T = \text{unit} (\bar{v}_g)$. The above procedure is repeated each computational cycle with $\bar{v}_g$ being updated by

$$\bar{v}_g = \bar{v}_g - (\bar{v}_n - \bar{v}_n') - \frac{1}{2} \Delta T_p (\bar{g} + \bar{g}')$$

where

$\bar{v}_n$ = current velocity vector,

$\bar{v}_n'$ = previous velocity vector,

$\bar{g}$ = current acceleration due to drag and Earth's potential

$\bar{g}'$ = previous $\bar{g}$

ΔT_p = elapsed time from $\bar{v}_n'$ to $\bar{v}_n$

The APS and SPS maneuvers are five phase:

Phase 1 - ullage

Phase 2 - ullage plus buildup

Phase 3 - ullage plus main engine

Phase 4 - main engine

Phase 5 - tailoff

The DPS maneuver is six phase:

- Phase 1 - ullage
- Phase 2 - ullage plus ten percent
- Phase 3 - ten percent
- Phase 4 - buildup of main engine
- Phase 5 - main engine
- Phase 6 - tailoff

The thrust level for each phase, except tailoff and main engine phase, is a system parameter. The thrust level during main engine phase is a function of the weight and is computed by subroutine GIMGBL. The thrust level for tailoff is

$$T = \frac{T_c}{T_o} T_{to}$$

where

T_c = thrust at engine off,

T_o = full load thrust,

T_{to} = tailoff thrust corresponding to full load thrust

During the ullage phase of APS, DPS, and SPS maneuvers the unit thrust vector is given by

$$\hat{A}_T = \hat{X}_b.$$

At main engine on the unit thrust vector is determined as follows:

Fixed Inertial

When the initial attitudes are given the unit thrust vector is determined in the same manner as for a manual maneuver. When the initial attitudes are not given, the unit thrust vector is input. In both cases the initial direction is held throughout the burn.

Manual

The unit thrust vector is determined from a sequence of rotations:

$$\begin{bmatrix} \hat{A}_T \\ \hat{Y}_T \\ \hat{Z}_T \end{bmatrix} = R_Z(\theta_Z) R_Y(\theta_Y) \begin{bmatrix} \hat{X}_b \\ \hat{Y}_b \\ \hat{Z}_b \end{bmatrix}$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMRKJ
 DATE 11/27/67(EORP)
 REV
 PAGE 11 of 47

where

- $\hat{X}_b$ = inertial coordinates of X body axis,
 $\hat{Y}_b$ = inertial coordinates of Y body axis,
 $\hat{Z}_b$ = inertial coordinates of Z body axis,
 R_Y = rotation about Y body axis,
 R_Z = rotation about Z body axis,
 θ_Z = cant angles for APS thruster,
 θ_Y = cant angles for APS thruster,
 θ_Z = gimbal angles so as to fire through the
 θ_Y = center of gravity for SPS and DPS
 thrusters.

External $\Delta\bar{V}$

When the initial attitudes are given, the initial unit thrust vector is determined in the same manner as for a manual maneuver. Steering is suppressed for two computational cycles after main engine on. After this time, the unit thrust vector is determined as described for RCS external $\Delta\bar{V}$ maneuvers. When the initial attitudes are not given, the initial unit thrust vector is determined as described for RCS external $\Delta\bar{V}$ maneuver.

The numerical integration technique used to integrate the equations of motion is the Nystrom modification of a fourth-order Runge-Kutta scheme given by:

$$r_{n+1,i} = r_{n,i} + h \left[r_{n,i} + \frac{1}{6} (K_{1,i} + K_{2,i} + K_{3,i}) \right],$$

$$r_{n+1,i} = r_{n,i} + \frac{1}{6} (K_{1,i} + 2K_{2,i} + 2K_{3,i} + K_{4,i}),$$

where:

$$K_{1,i} = h \ddot{r}(t_n, r_{n,i}, \dot{r}_{n,i}),$$

$$K_{2,i} = h \ddot{r} \left[\left(t_n + \frac{h}{2} \right), \left(r_{n,i} + \frac{h}{2} \dot{r}_{n,i} + \frac{h}{8} K_{1,i} \right), \left(\dot{r}_{n,i} + \frac{1}{2} K_{1,i} \right) \right],$$

$$K_{3,i} = h \ddot{r} \left[\left(t_n + \frac{h}{2} \right), \left(r_{n,i} + \frac{h}{2} \dot{r}_{n,i} + \frac{h}{8} K_{1,i} \right), \left(\dot{r}_{n,i} + \frac{1}{2} K_{2,i} \right) \right],$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMRKJ
DATE 11/27/67 (EO
REV
PAGE 12 of 47

$$K_{4,i} = h \ddot{r} \left[(t_n + h), \left(r_{n,i} + h \dot{r}_{n,i} + \frac{h}{2} K_{3,i} \right), \left(\dot{r}_{n,i} + K_{3,i} \right) \right],$$

where:

h = integration time step,

r = position vector,

 $\dot{r}$ = velocity vector, $\ddot{r}$ = acceleration vector,

for i = x, y, z, and n = 1, 2, 3, -----



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMRKJ

DATE 11/27/67 (EORP)

REV

PAGE 13 of 47

Table PMMRKJ-5. System Parameters

Symbol in Program	Symbol In Master Tape	Type *	Units	Comments
DTSPAN(2)	MCTDD2		hrs.	DPS phase 2 burn time
DTSPAN(4)	MCTDD5		hrs.	DPS burn time for short maneuver time
DTSPAN(5)	MCTDD4		hrs.	DPS build-up to maximum burn time
DTSPAN(6)	MCTAD6		hrs.	DPS elapsed time from engine on to initial steering
DTSPAN(7)	MCTDD7		hrs.	DPS time for ten percent test
DTSPAN(8)	MCTDD6		hrs.	DPS tail-off burn time
XK1	MCTSK1	D	lb-sec.	Impulse in one secone SPS burn
XK2VAL	MCTSK2	D	lb-sec.	Initial value of SPS minimum impulse curve
XK3VAL	MCTSK3	D	lb.	Slope of SPS curve
XK4	MCTSK4	D	lb-sec.	CSM RCS 7-second 2-jet ullage impulse
XK1	MCTAK1	D	lb-sec.	Impulse in one second APS burn
XK2VAL	MCTAK2	D	lb-sec.	Initial value of APS minimum impulse curve
XK3VAL	MCTAK3	D	lb.	Slope of APS curve
XK4	MCTAK4	D	lb-sec.	LM RCS 7-second 2-jet ullage impulse

*Type
 H - Fixed-point half word
 F - Fixed-point full word
 E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal



Table PMMRKJ-5. System Parameters (Cont'd.)

Symbol in Program	Symbol In Master Tape	Type *	Units	Comments
THPS(1)				
(2)	MCTDT2	D	lbs.	DPS phase 2 -thrust level
(3)				
(4)	MCTDT3		lbs.	DPS 10% thrust level
(5)	MCTDT4		lbs.	DPS buildup to maximum thrust level
(6)				
(7)				
(8)	MCTPT6		lbs.	DPS tail-off thrust level
(9)				
(10)	MCTDT5		lbs.	DPS full load thrust level
WDOTPS(1)				
(2)	MCTDW2		lbs./hr.	DPS phase 2 weight loss rate
(3)				
(4)	MCTDW3	▼	lbs./hr.	DPS weight loss rate at 10% thrust

*Type
 H - Fixed-point half word
 F - Fixed-point full word
 S - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMRKJ
 DATE 11/27/67 (EORP)
 REV
 PAGE 15 of 47

Table PMMRKJ-5. System Parameters (Cont'd.)

Symbol in Program	Symbol In Master Tape	Type*	Units	Comments
WDOTPS(5)	MCTDW4	D	lbs./hr.	DPS buildup to maximum weight loss rate
(6)				
(7)				
(8)	MCTDW6		lbs./hrs.	DPS tail-off weight loss rate
(9)				
(10)	MCTDW5		lbs./hr.	DPS full load weight loss rate
THPS(2)	MCTAT2		lbs.	APS phase 2 thrust level
THPS(3)	MCTAT3		lbs.	APS phase 3 thrust level
THPS(7)	MCTAT6		lbs.	APS thrust level for short maneuver test
THPS(8)	MCTAT5		lbs.	APS tail-off thrust level
THPS(10)	MCTAT4		lbs.	APS full load thrust level
WDOTPS(2)	MCTAW2		lbs./hr.	APS phase 2 weight loss rate
WDOTPS(3)	MCTAW3		lbs./hr.	APS phase 3 weight loss rate
WDOTPS(7)	MCTAW6	▼	lbs./hr.	APS weight loss rate for short maneuver test

*Type
 H — Fixed-point half word
 F — Fixed-point full word
 E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

Table PMMRKJ-5. System Parameters (Cont'd.)

Symbol in Program	Symbol In Master Tape	* Type	Units	Comments
WDOTPS(8)	MCTAW5	D	lbs./hr.	APS tail-off weight loss rate
DTSPAN(2)	MCTAD2		hrs.	APS phase 2 burn time
DTSPAN(3)	MCTAD3		hrs.	APS phase 3 burn time
DTSPAN(4)	MCTAD4		hrs.	APS burn time for short maneuver test
DTSPAN(6)	MCTAD6		hrs.	APS elapsed time from engine on to initial steering
DTSPAN(7)	MCTAD7		hrs.	APS minimum impulse time
DTSPAN(8)	MCTAD5		hrs.	APS tail-off burn time
HUA	MCAUAC		Er	Altitude of entry interface
AOBSQD	MCEABS			Geodetic Earth model A^2/B^2
OMEGA	MCERTS		radians/hr.	Angular velocity of Earth
G2	MCMShP			Second harmonic coefficient
G	MCEACG		Er/hr^2	Gravitational acceleration
XMU	MCEMUU		Er^3/hr^2	Gravitational constant
CDRAG	MCADRG	▼		Coefficient of drag

* Type
 E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 F - Fixed-point full word
 H - Fixed-point half word
 X - Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMRKJ

DATE 11/27/67 (EORP)

REV

PAGE 17 of 47

Table PMMRKJ-5. System Parameters (Cont'd.)

Symbol in Program	Symbol in Master Tape	Type*	Units	Comments
WTLM	MCVCWT	D	lbs.	CSM limiting weight
WTLM	MCVLWT	D	lbs.	LM limiting weight
THPS(1)	MCTJT7		lbs.	S-IVB blowdown vent thrust level
WDOTPS(1)	MCTJW7		lbs./hr.	S-IVB blowdown vent weight loss rate
THPS(1)	MCTLT1		lbs.	LM RCS (+ X, 2 quad.) thrust level
WDOTPS(1)	MCTLW1		lbs./hr.	LM RCS (+ X, 2 quad.) weight loss rate
THPS(1)	MCTLT2		lbs.	LM RCS (+ X, 2 quad.) thrust level
WDOTPS(1)	MCTLW2		lbs./hr.	LM RCS (+ X, 2 quad.) weight loss rate
THPS(1)	MCTLT3		lbs.	LM RCS (- X, 2 quad.) thrust level
WDOTPS(1)	MCTLW3		lbs./hrs.	LM RCS (- X, 2 quad.) weight loss rate
THPS(1)	MCTLT4		lbs.	LM RCS (- X, 4 quad.) thrust level
WDOTPS(1)	MCTLW4		lbs./hrs.	LM RCS (- X, 4 quad.) weight loss rate
THPS(1)	MCTCT1		lbs.	CSM RCS (+ X, 2 quad.) thrust level
WDOTPS(1)	MCTCW1	▼	lbs./hr.	CSM RCS (+ X, 2 quad.) weight loss rate

*Type
 H - Fixed-point half word
 F - Fixed-point full word
 S - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMRKJ

DATE 11/27/67 (EORP)

REV

PAGE 18 of 47

Table PMMRKJ-5. System Parameters (Cont'd.)

Symbol in Program	Symbol In Master Tape	Type*	Units	Comments
THPS(1)	MCTCT2	D	lbs.	CSM RCS (+ X, 4 quad.) thrust level
WDOTPS(1)	MCTCW2		lbs./hr.	CSM RCS (+ X, 4 quad.) weight loss rate
THPS(1)	MCTCT3		lbs.	CSM RCS (- X, 2 quad.) thrust level
WDOTPS(1)	MCTCW3		lbs./hr.	CSM RCS (- X, 2 quad.) weight loss rate
THPS(1)	MCTCT4		lbs.	CSM RCS (- X, 4 quad.) thrust level
WDOTPS(1)	MCTCW4		lbs./hr.	CSM RCS (- X, 4 quad.) weight loss rate
PITCHG	MCTSPG		radians	SPS trim angles
YAWG	MCTSYG		radians	SPS trim angles
PITCHG	MCTAPG		radians	APS cant angles
YAWG	MCTARG		radians	APS cant angles
PITCHG	MCTDPG		radians	DPS trim angles
YAWG	MCTDRG		radians	DPS trim angles
THPS(2)	MCTST2		lbs.	SPS phase 2 thrust level
THPS(3)	MCTST3	▼	lbs.	SPS phase 3 thrust level

*Type
 H - Fixed-point half word
 F - Fixed-point full word
 E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal

Table PMMRKJ-5. System Parameters (Cont'd.)

Symbol in Program	Symbol In Master Tape	Type *	Units	Comments
THPS(7)	MCTST6	D	lbs.	SPS thrust level for short maneuver test
(8)	MCTST5		lbs.	SPS tail-off thrust level
THPS(10)	MCTST4		lbs.	SPS full load thrust level
WDOTPS(2)	MCTSW2		lbs./hr.	SPS phase 2 weight loss rate
WDOTPS(3)	MCTSW3		lbs./hr.	SPS phase 3 weight loss rate
WDOTPS(7)	MCTSW6		lbs./hr.	SPS weight loss rate for short maneuver test
WDOTPS(8)	MCTSW5		lbs./hr.	SPS tail-off weight loss rate
WDOTPS(10)	MCTSW4		lbs./hr.	SPS full load weight loss rate
DTSPAN(2)	MCTSD2		hours	SPS phase 2 burn time
DTSPAN(3)	MCTSD3		hours	SPS phase 3 burn time
DTSPAN(4)	MCTSD4		hours	SPS burn time for short maneuver test
DTSPAN(6)	MCTSD6		hours	SPS elapsed time from engine on to initial steering
DTSPAN(7)	MCTSD7		hours	SPS minimum impulse time
DTSPAN(8)	MCTSD5	▼	hours	SPS tail-off burn time

*Type
H - Fixed-point half word
F - Fixed-point full word
E - Single-precision (floating-point)
D - Double-precision (floating-point)
X - Hexadecimal



Table PMMRKJ-6. Description of Terms

Definition of Term	Symbol In Program	Symbol In Model	Med I/O	* Type	Units	Limits (Min/Max)	Comments
Maneuver initiation time	TBM	TBM		2	hours		
Kickout altitude	HIMPCT	H_{Ro}		2	Er		
Current weight	WT	wt.		2	lbs.		
Input weight	CAPWT	Wt_{in}		2	lbs.		
Working position vector	RP	r_p		2	Er		
Current position vector	R	r		2	Er		
Working velocity vector	VP	v_p		2	Er/hr.		
Current velocity vector	V	v		2	Er/hr.		
Previous time	TPREV	T_{prev}		2	hours		
Previous step size	DTPREV	ΔT^1		2	hours		

*Type Code:

C = command

X = integer (fixed point)

L = logical

1 = single precision (floating point)

H = binary coded decimal (Hollerith)

2 = double precision (floating point)

O = octal



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMRKJ

DATE 11/27/67 (EORP)

REV

PAGE 21 of 47

Table PMMRKJ-6. Description of Terms (Cont'd.)

Definition of Term	Symbol In Program	Symbol In Model	Med I/O	* Type	Units	Limits (Min/Max)	Comments
Previous step through guidance	DTP	ΔT_p		D	hours		
Current step through guidance	DTN	ΔT_n			hours		
Potential acceleration	RDDP	g			Er/hr. ²		
Current integration step	STEP	Step			hours		
Free-flight step size	DFFF	ΔT_{ff}			hours		
Time of next output	TNEXT	T_{next}			hours		
Indicator	TLARGE	T_{large}			hours		
Time of next event	TI	T_i			hours		
End of current step	TE	T_e		▼	hours		
Ephemeris Counter	JT	n		H			

*Type Code:

X = integer (fixed point)

1 = single precision (floating point)

2 = double precision (floating point)

C = command

L = logical

H = binary coded decimal (Hollerith)

O = octal



Table PMMRKJ-6. Description of Terms (Cont'd.)

Definition of Term	Symbol In Program	Symbol In Model	Med I/O	* Type	Units	Limits (Min/Max)	Comments
Time to begin guidance	TGUID	T _g		D	hours		
Sign	SIGN	S _{gn}		D			
Weight table counter	KWT	KWT		D			
Current altitude comparison	HH	HH		D			
Previous altitude	ALTPR	ALT'		D	Er		
Thruster switch	KTHSWT	KTHSWT		H			
Error indicator	IERR	IERR		H			
Left integration limit	TLEFT	T _{left}		D	hours		
Anchor time	TA	TA		D	hours		
Maneuver type	MANOP	MANOP		H			

*Type Code:

X = integer (fixed point)

I = single precision (floating point)

2 = double precision (floating point)

C = command

L = logical

H = binary coded decimal (Hollerith)

O = octal



Table PMMRKJ-6. Description of Terms (Cont'd.)

Definition of Term	Symbol In Program	Symbol In Model	Med I/O	* Type	Units	Limits (Min/Max)	Comments
Weight loss rate multiplier	WDMULT	$\dot{w}_{mult}$		D			
Thrust along X body axis	TH	Th_x		D	lbs.		
Current altitude	ALT	ALT		D	Er		
End of current integration step	TAU	T_{au}		D	hours		
Current time	T	T		D	hours		
Right integration limit	TRIGHT	T_{right}		D	hours		
Ullage time	DTUL	ΔT_{UL}		D	hours		
Pure ullage time	DTU	ΔT_u		D	hours		
Ullage plus buildup time	DTUB	ΔT_{ub}		D	hours		
Ullage plus overlap time	DTUSPS	ΔT_{usps}		D	hours		

*Type Code:

X = integer (fixed point)

1 = single precision (floating point)

2 = double precision (floating point)

C = command

L = logical

H = binary coded decimal (Hollerith)

O = octal



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission planning

2. PMMRKJ

DATE 11/27/67 (EORI

REV

PAGE 24 of 47

Table PMMRKJ-6. Description of Terms (Cont'd.)

Definition of Term	Symbol In Program	Symbol In Model	Med I/O	* Type	Units	Limits (Min/Max)	Comments
Maneuver phase	MPHASE	MPHASE		D			
Thrust level	THRUST	T _{hrust}		D	lbs.		
Wt. loss rate	WTLRT	$\dot{W}$		D	lbs./hr.		

*Type Code:

X = integer (fixed point)

1 = single precision (floating point)

2 = double precision (floating point)

C = command

L = logical

H = binary coded decimal (Hollerith)

O = octal

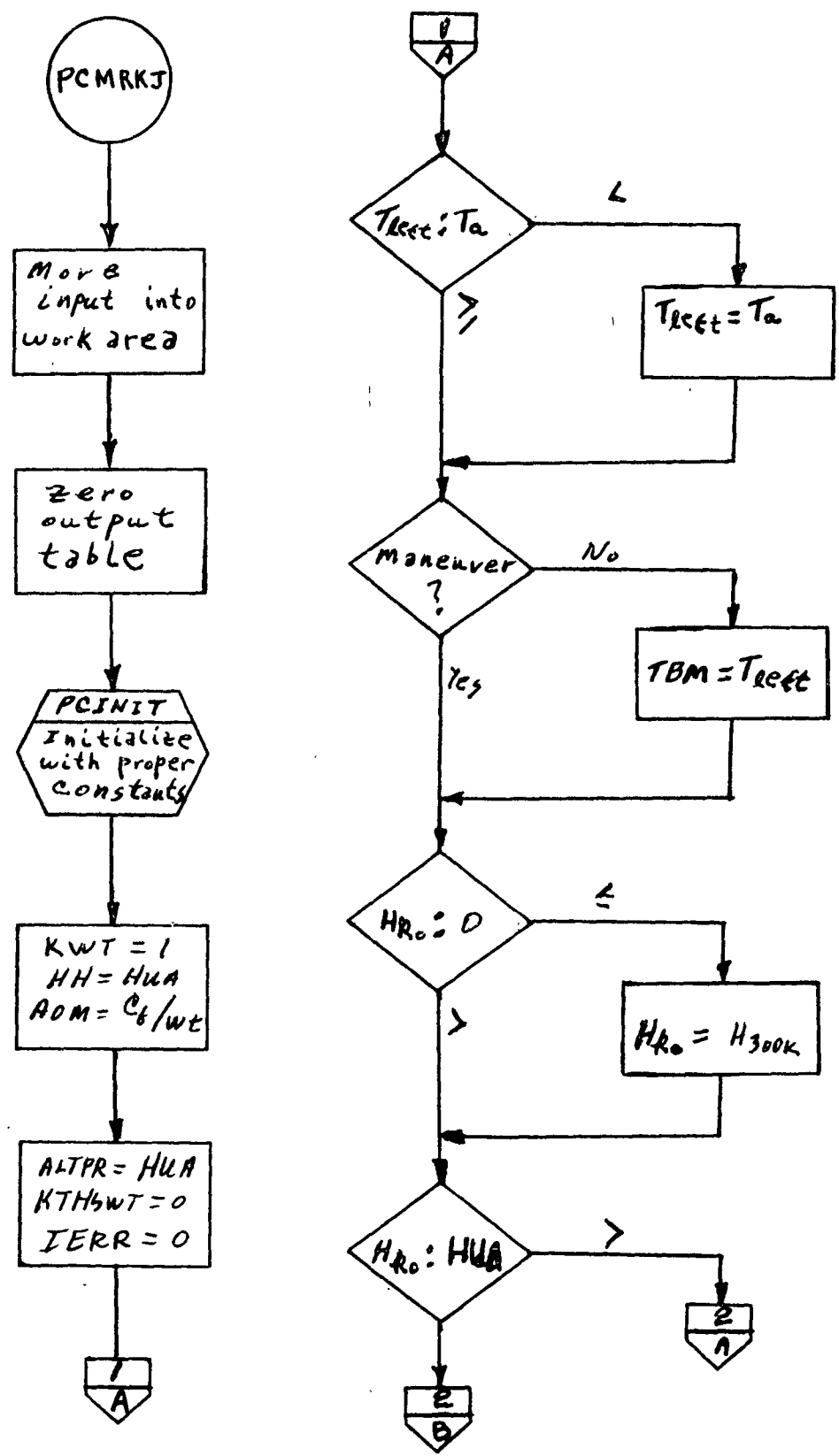


Figure PMMRKJ-1. Flowchart

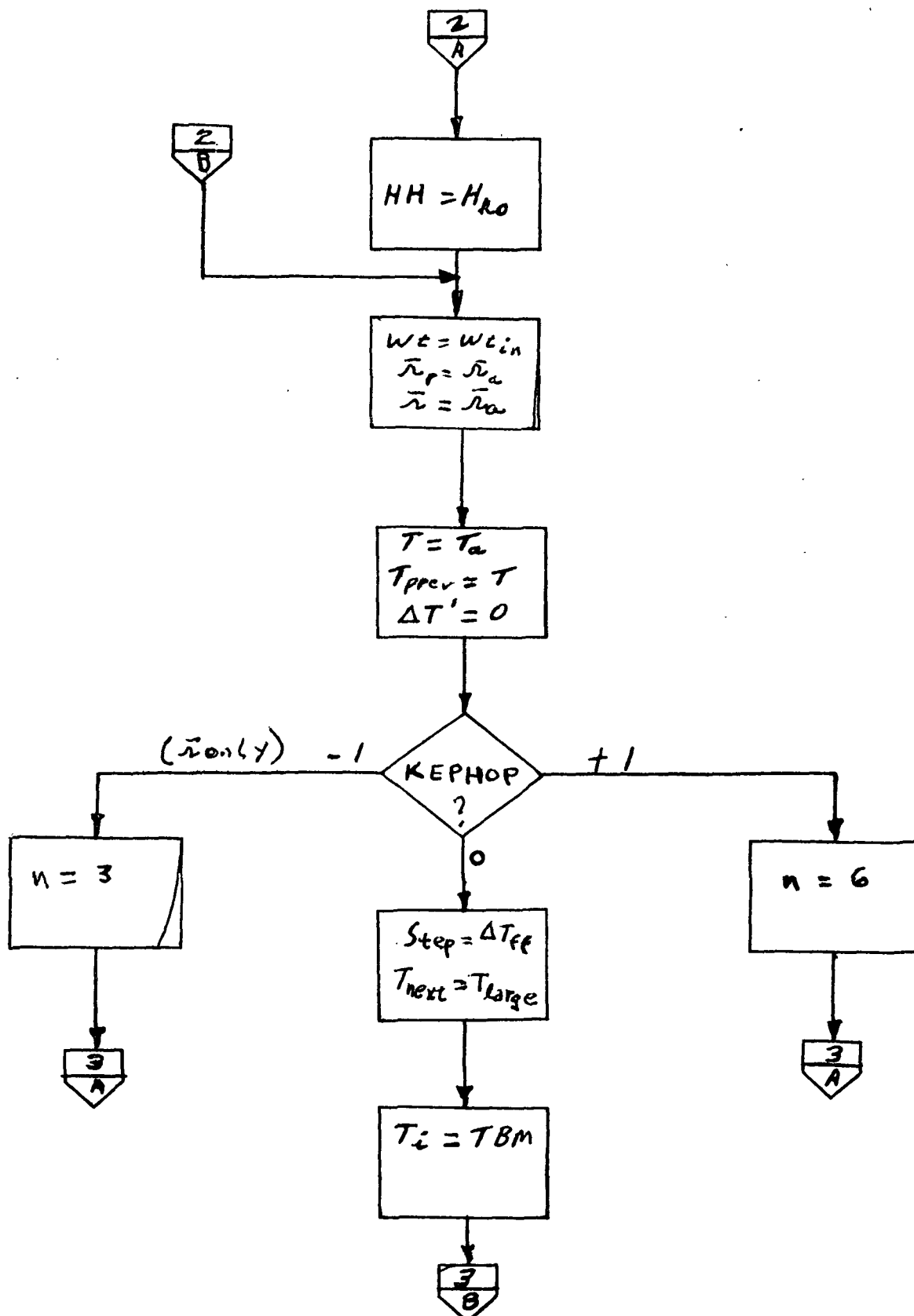


Figure PMMRKJ-1. Flowchart

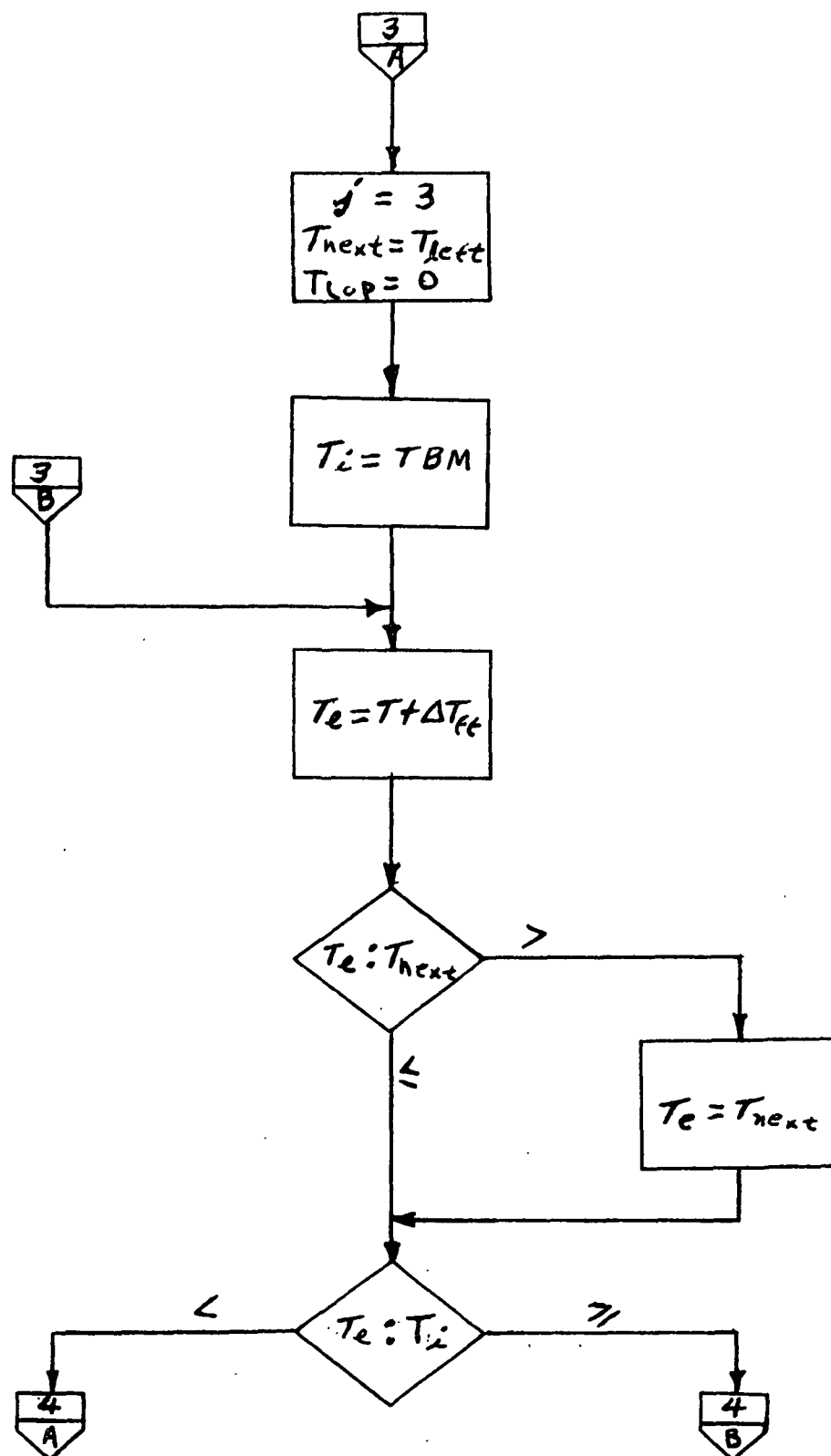


Figure PMMRKJ-1. Flowchart

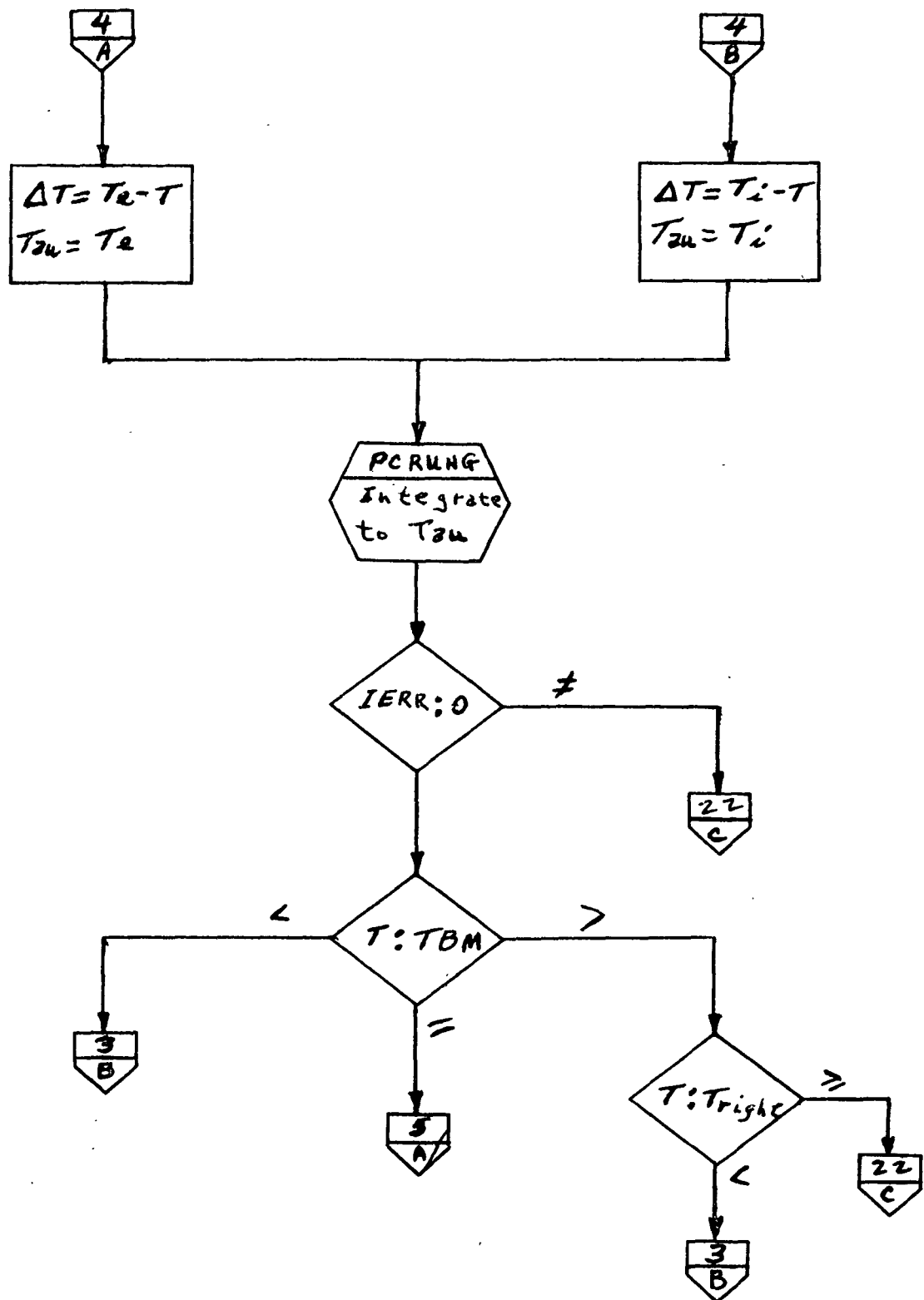


Figure PMMRKJ-1. Flowchart

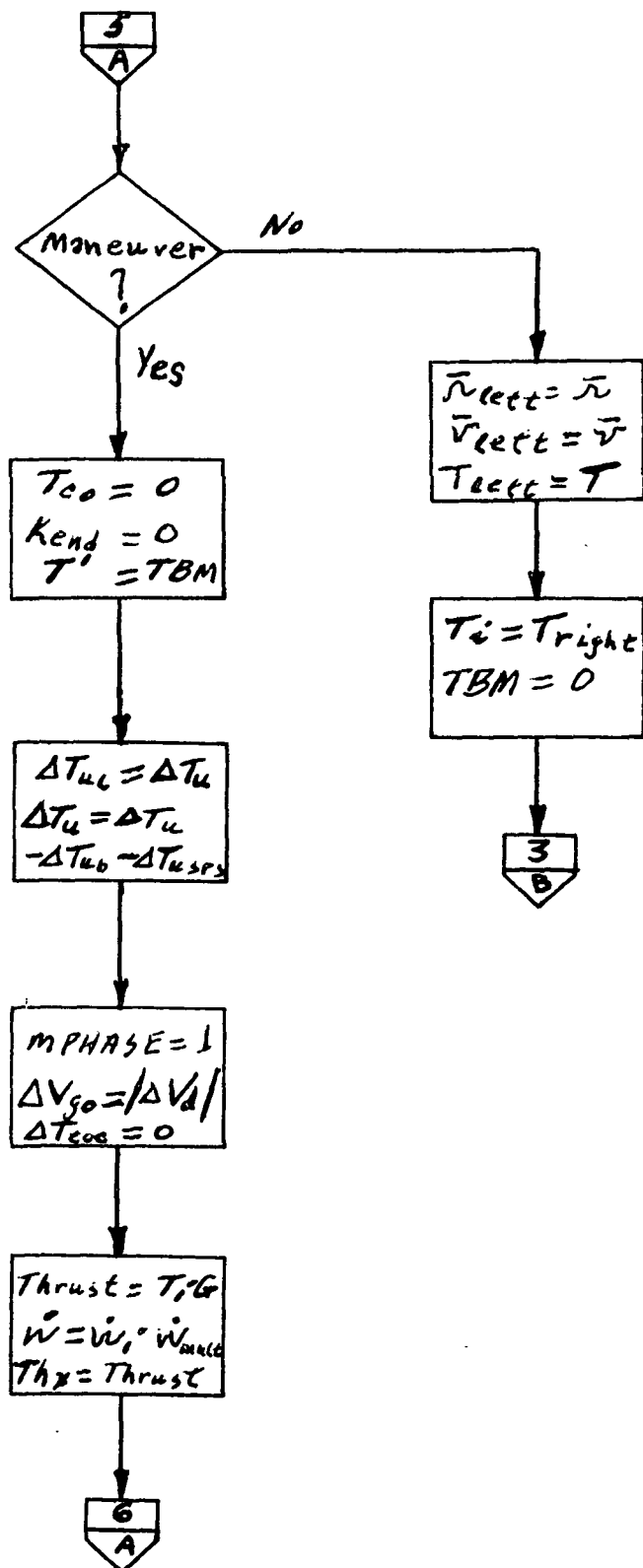


Figure PMMRKJ-1. Flowchart

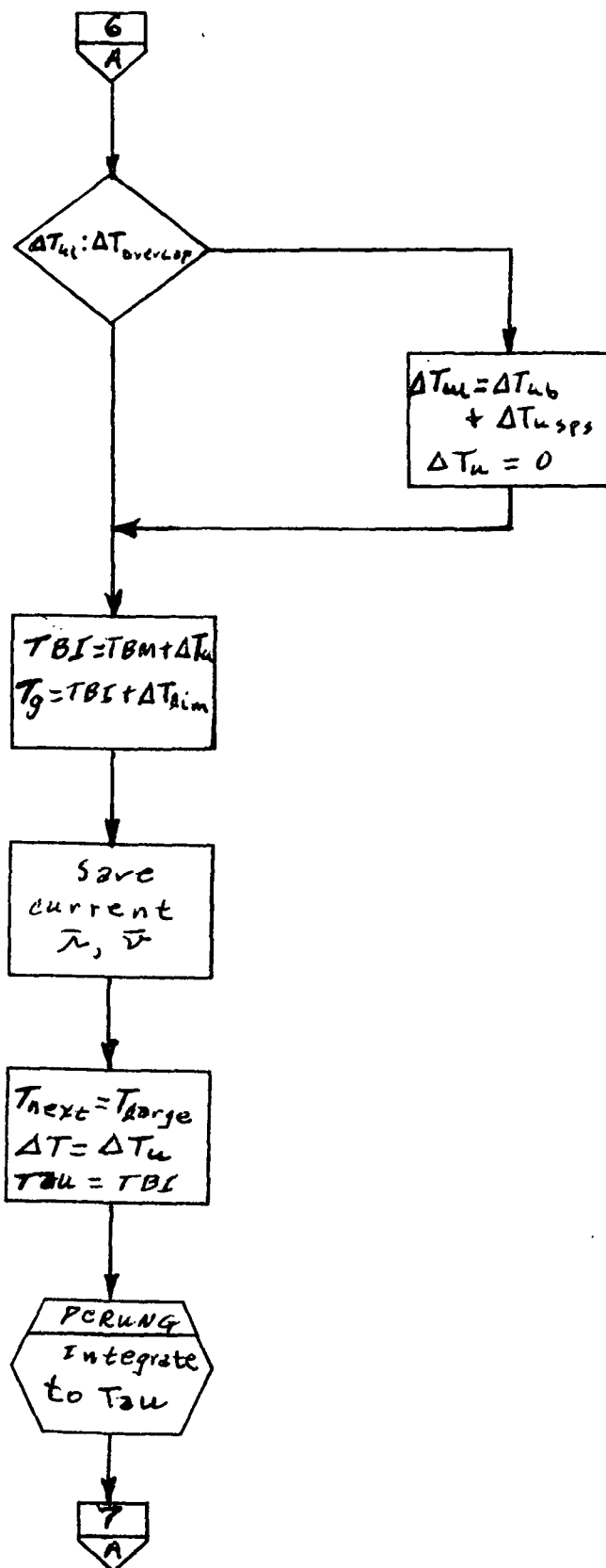


Figure PMMRKJ-1. Flowchart

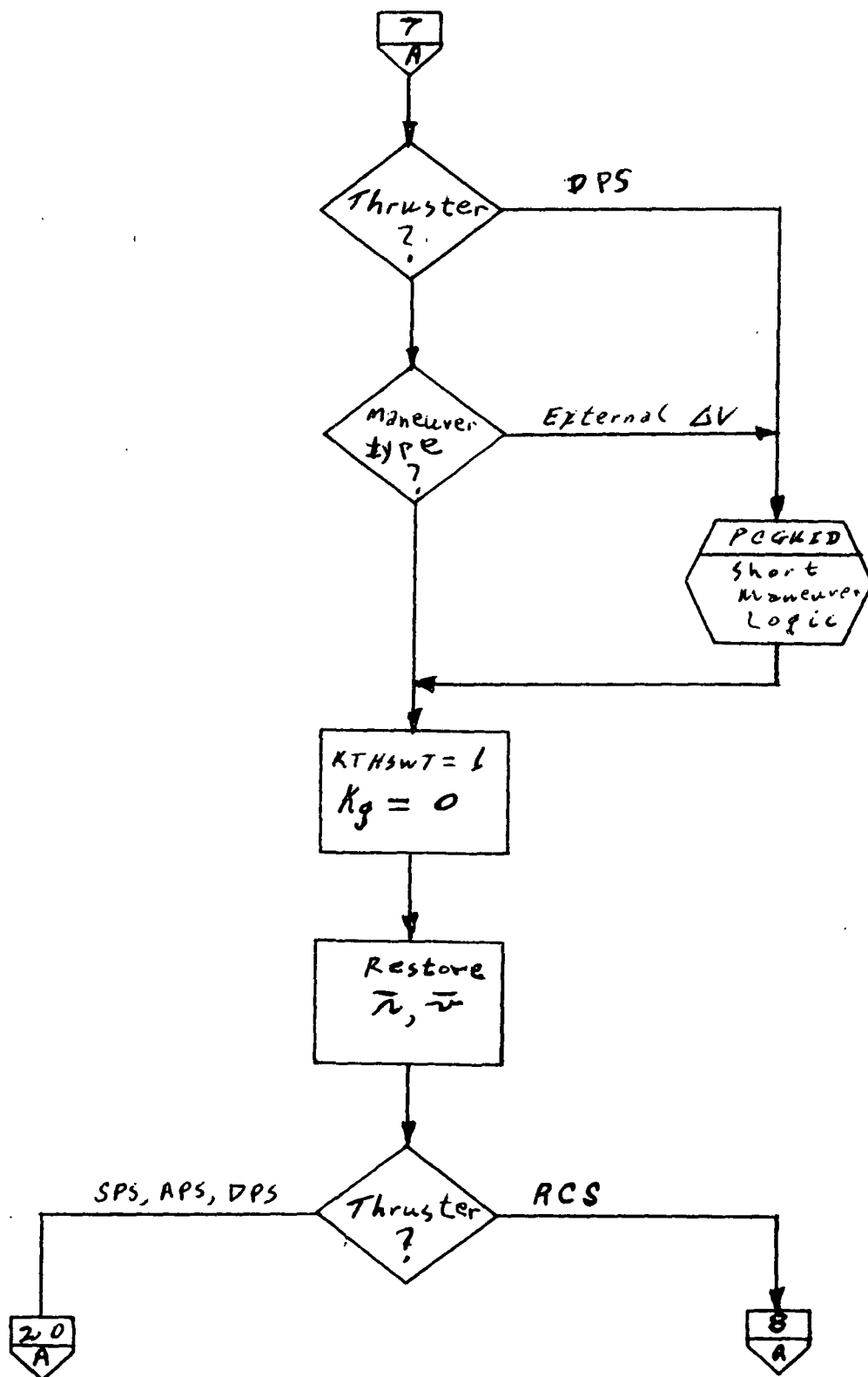


Figure PMMRKJ-1. Flowchart

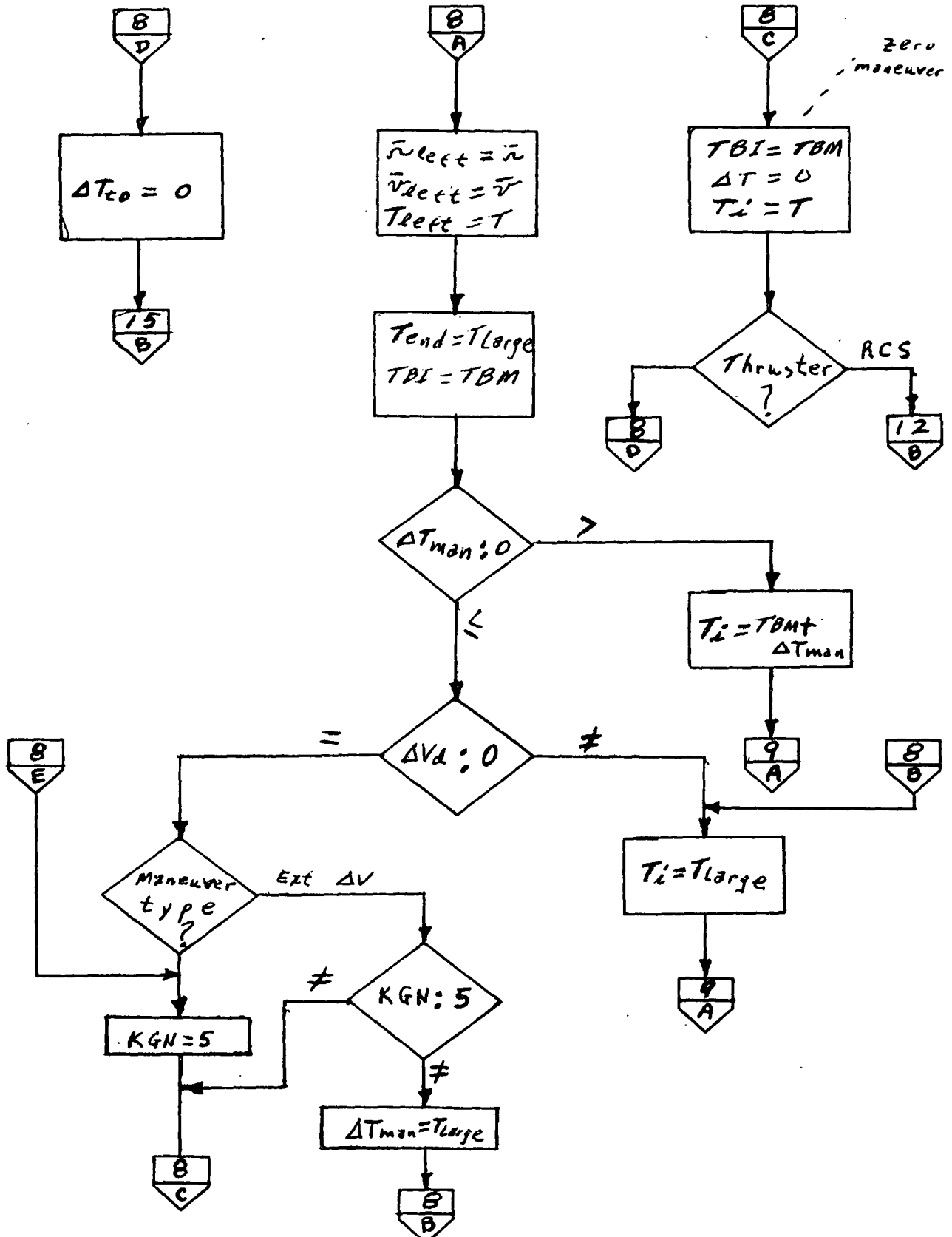


Figure PMMRKJ-1. Flowchart

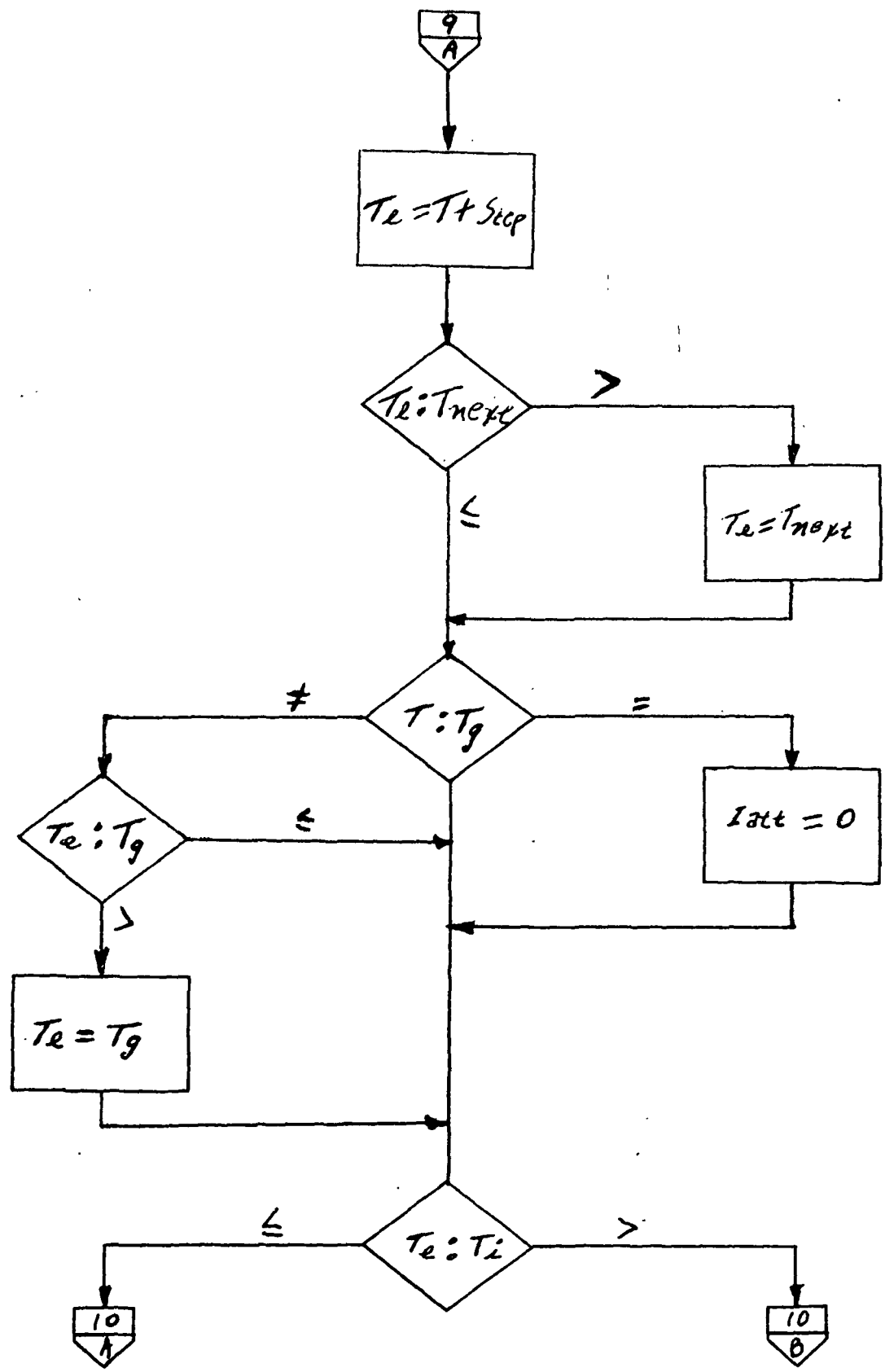


Figure PMMRKJ-1. Flowchart

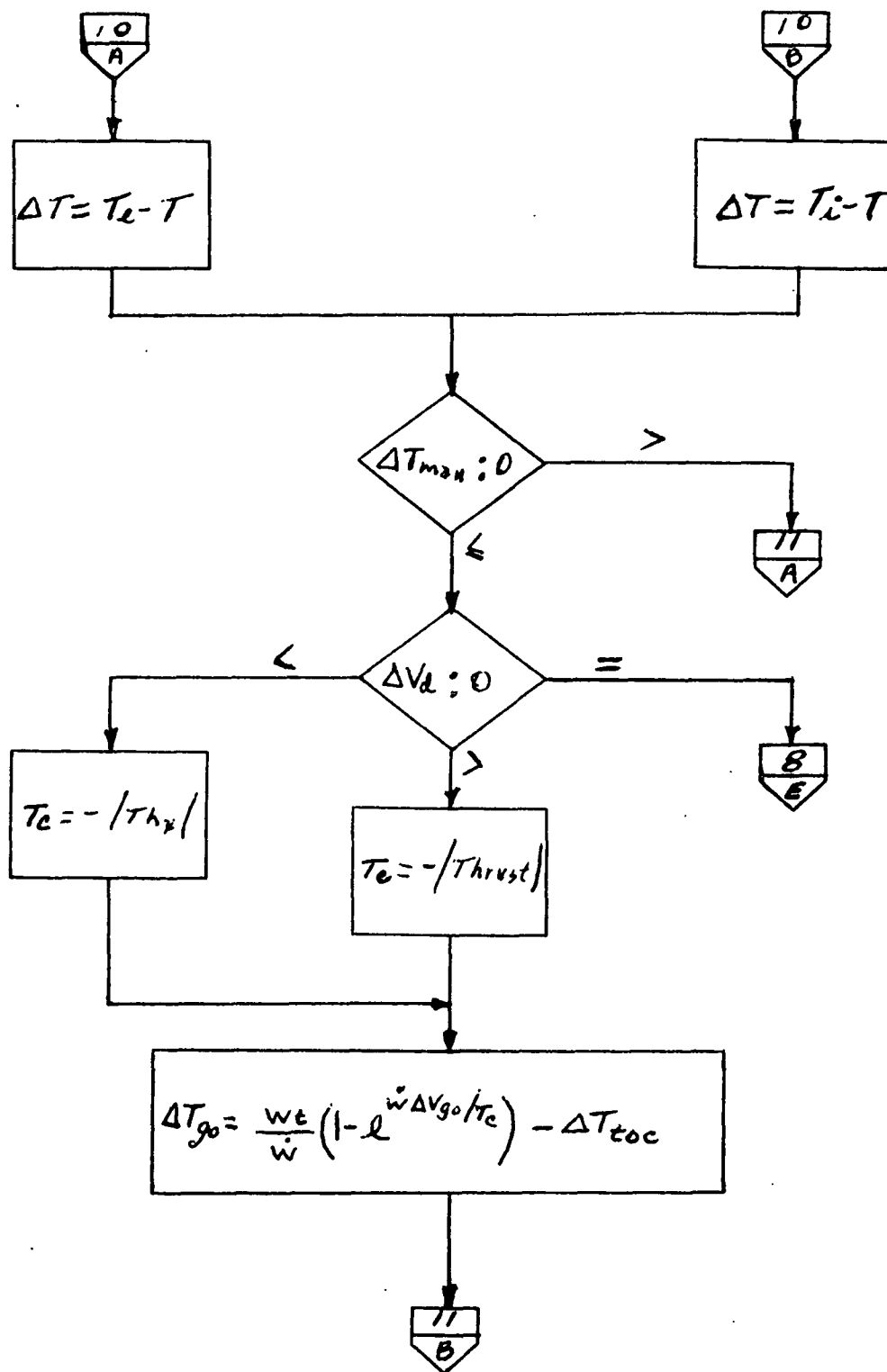


Figure PMMRKJ-1. Flowchart

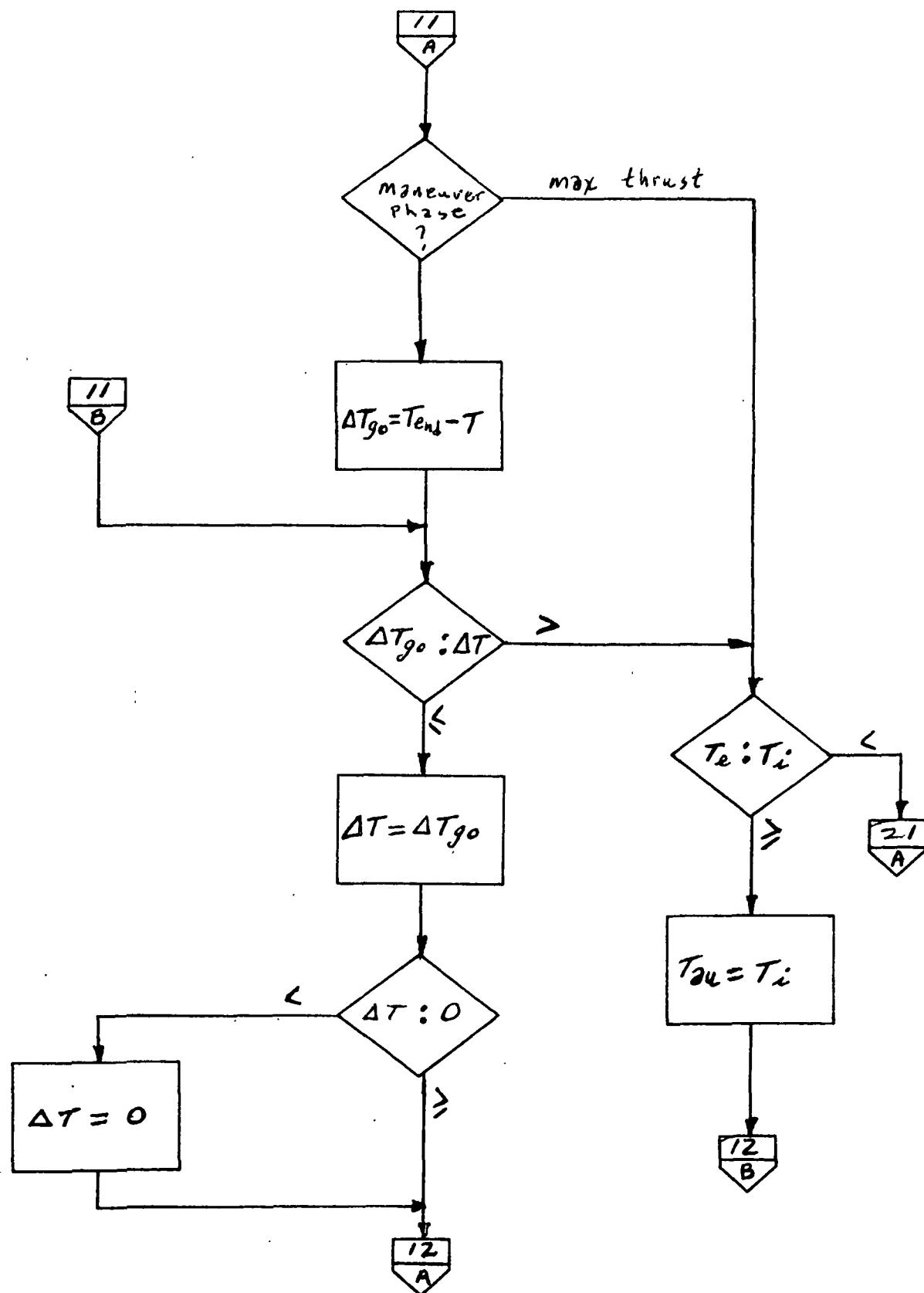


Figure PMMRKJ-1. Flowchart

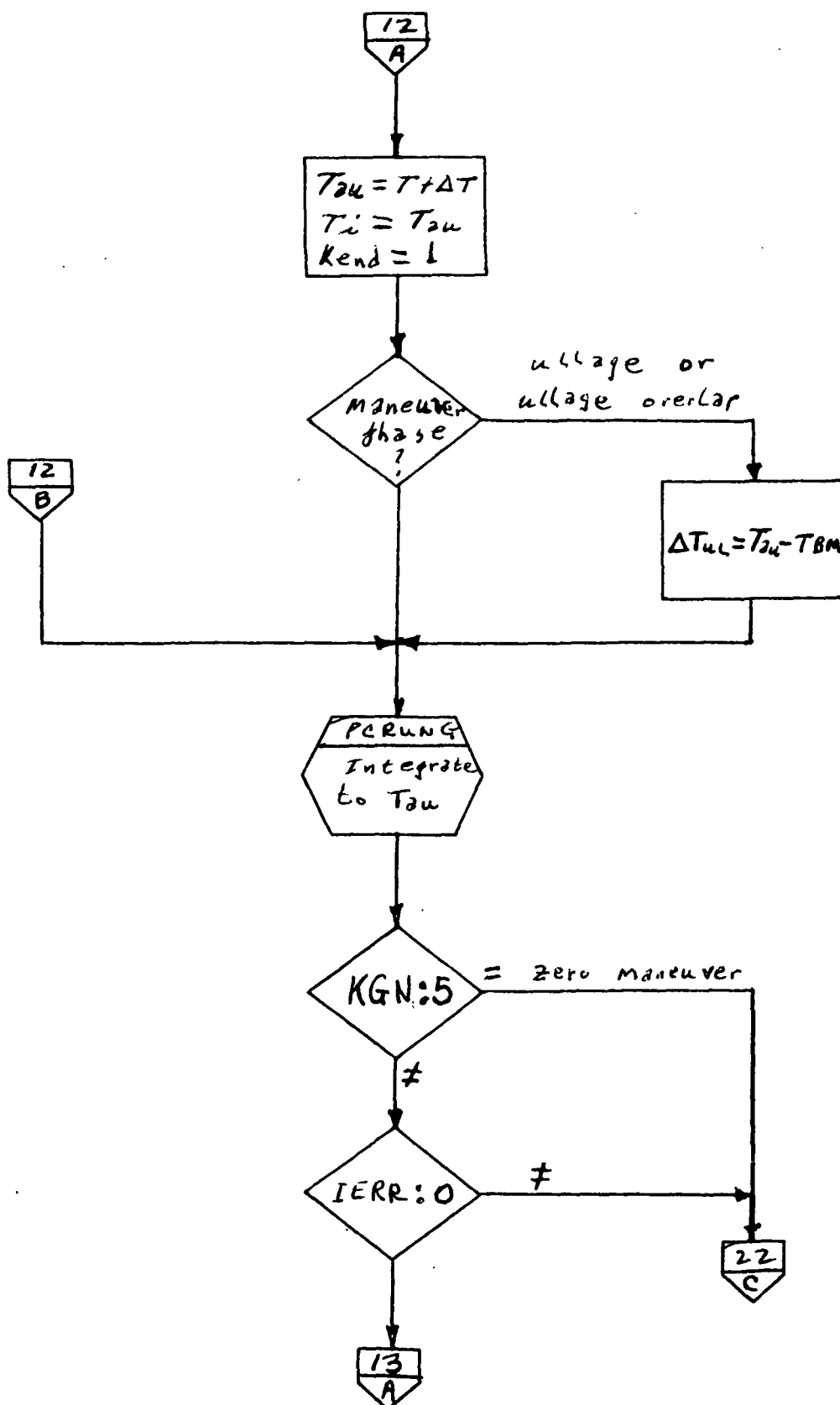


Figure PMMRKJ-1. Flowchart

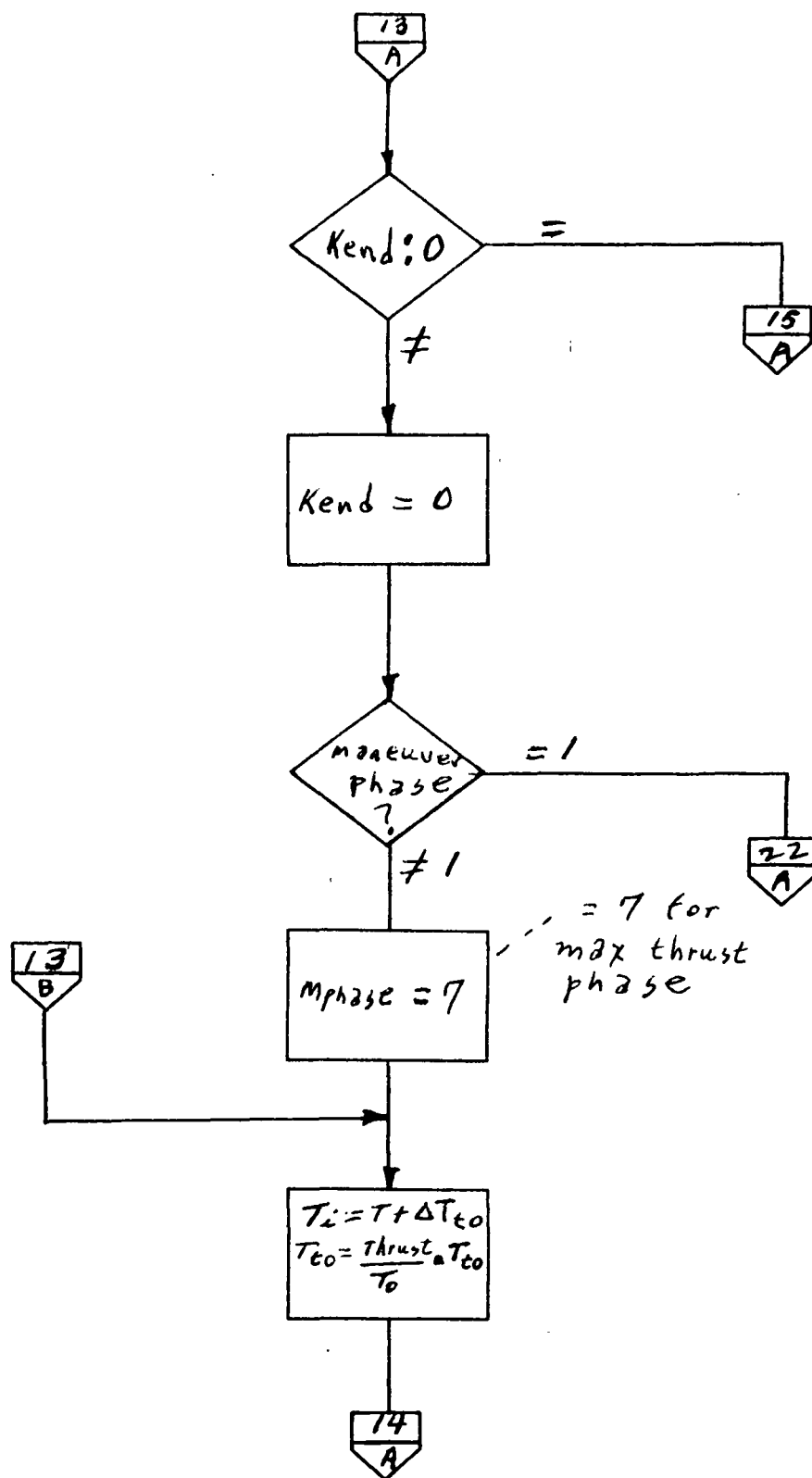


Figure PMMRKJ-1. Flowchart

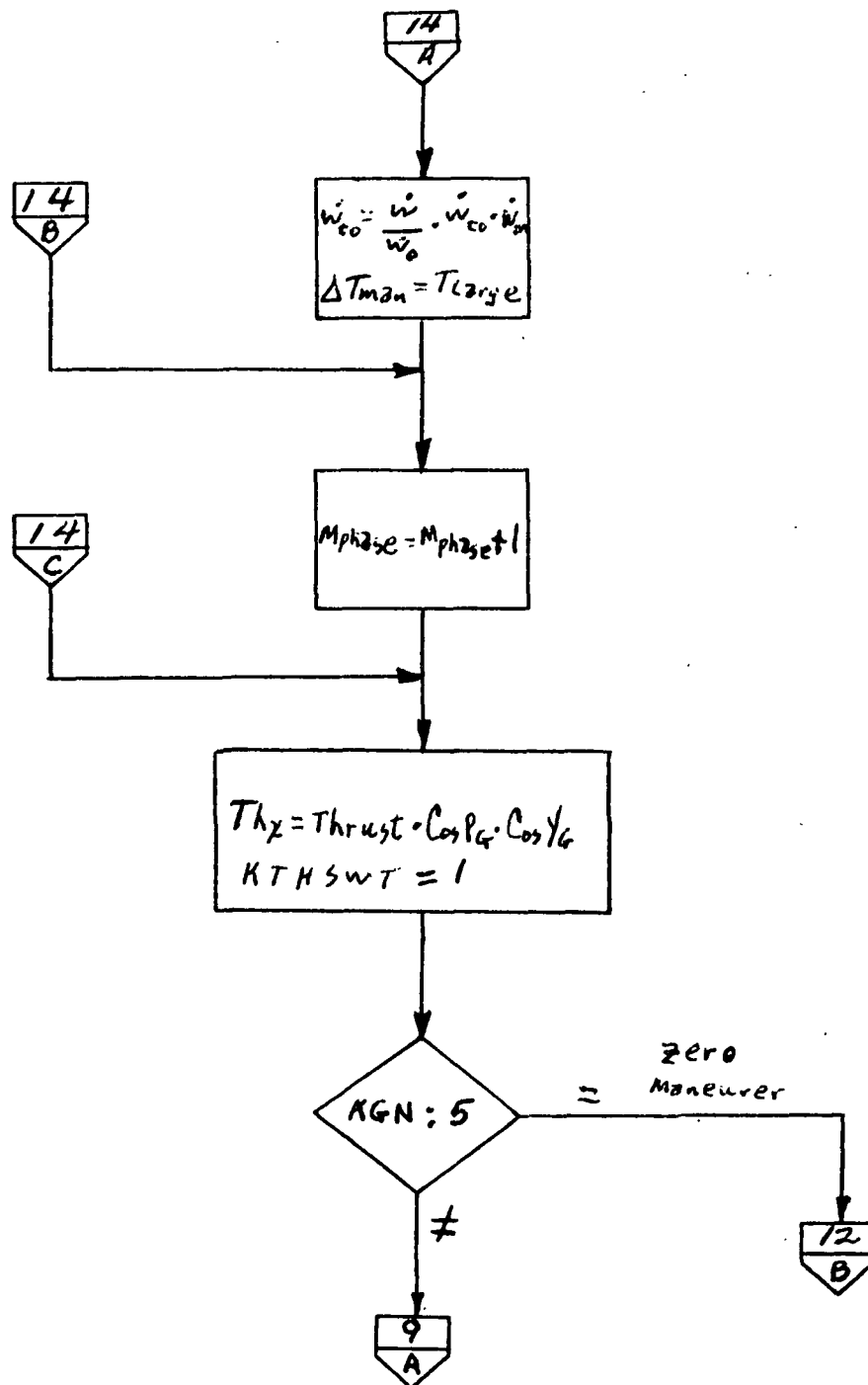


Figure PMMRKJ-1. Flowchart

NAS 9-996

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMRKJ

DATE 11/27/67 (EORP)

REV

PAGE 39 of 47

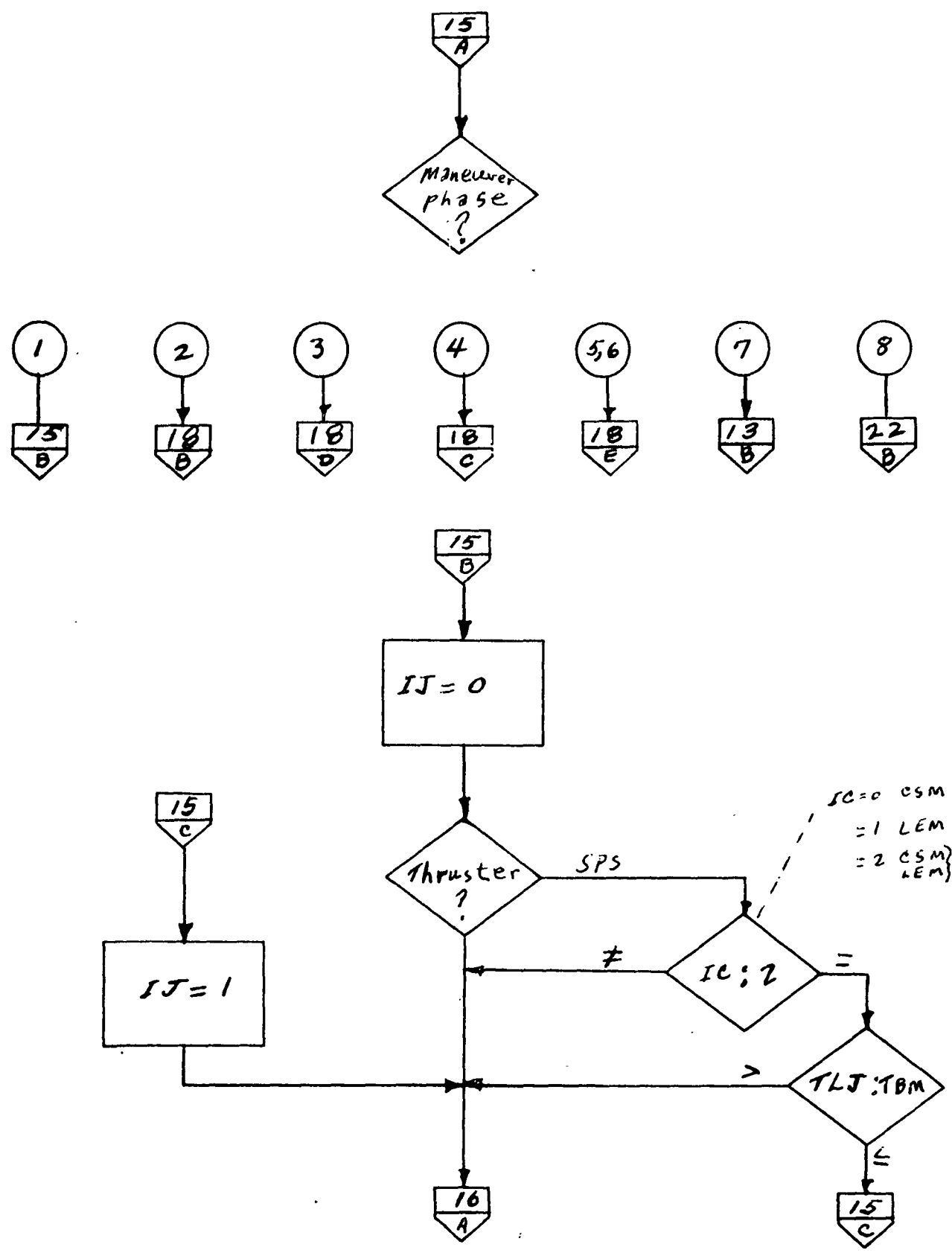


Figure PMMRKJ-1. Flowchart

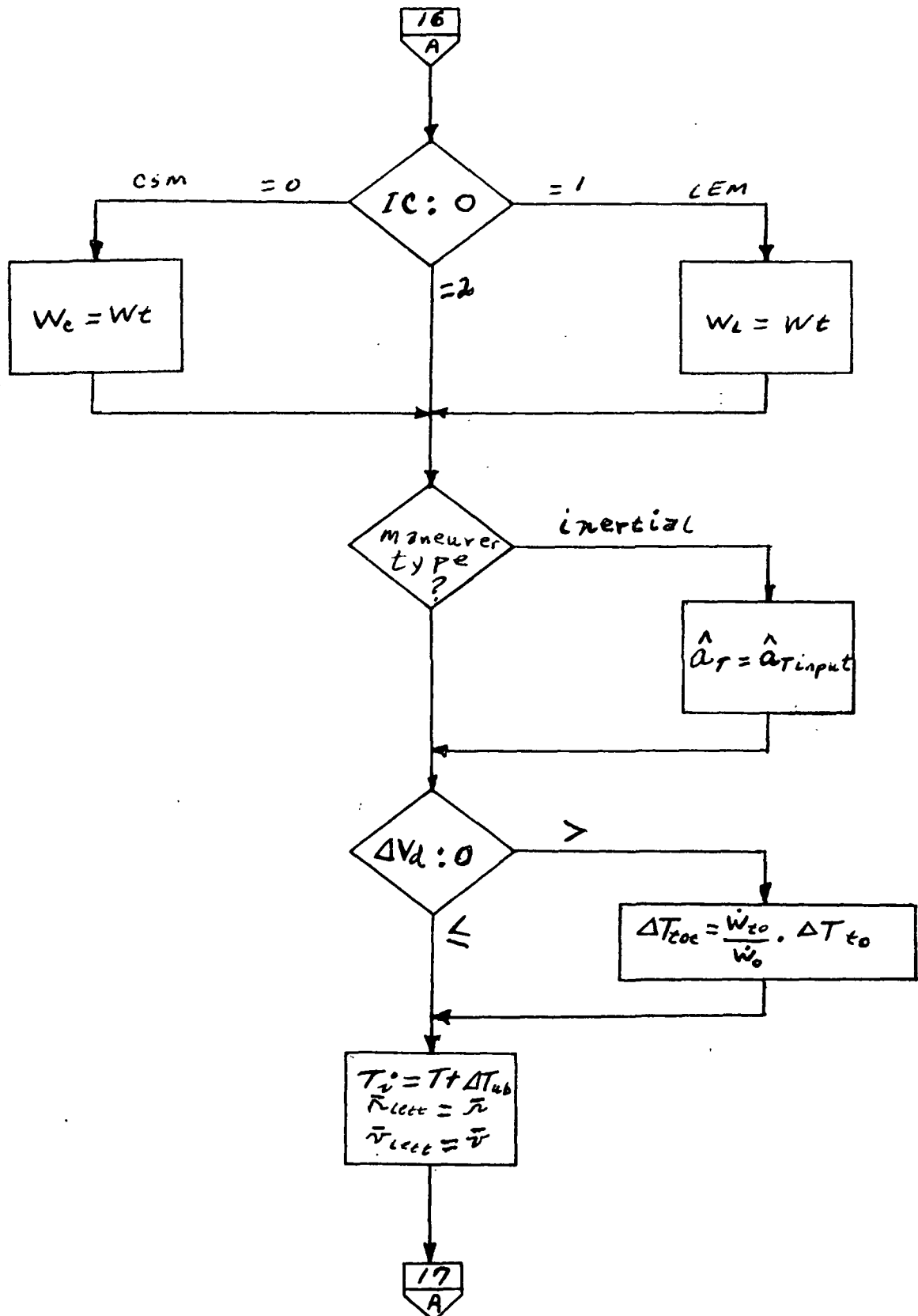


Figure PMMRKJ-1. Flowchart

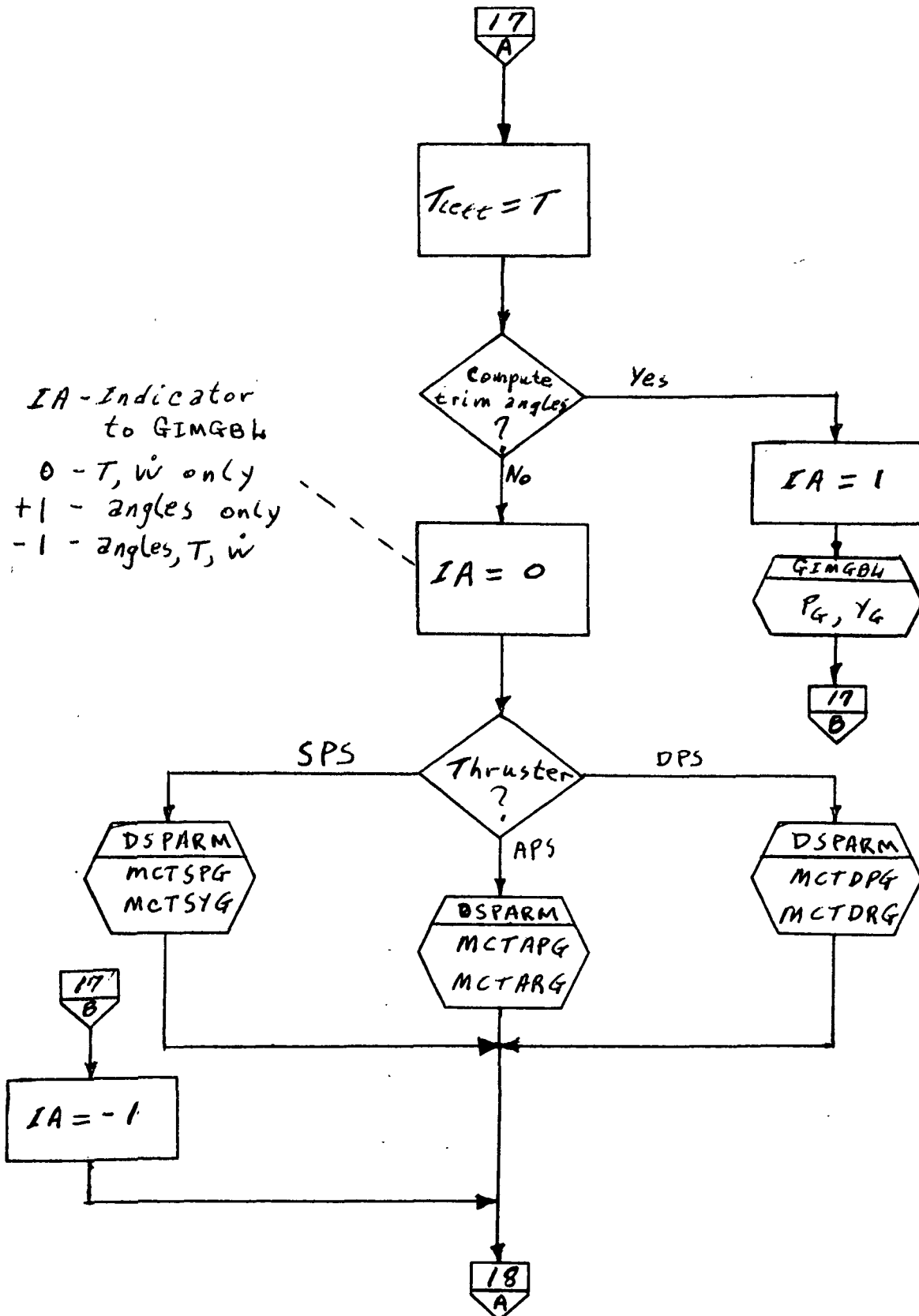


Figure PMMRKJ-1. Flowchart

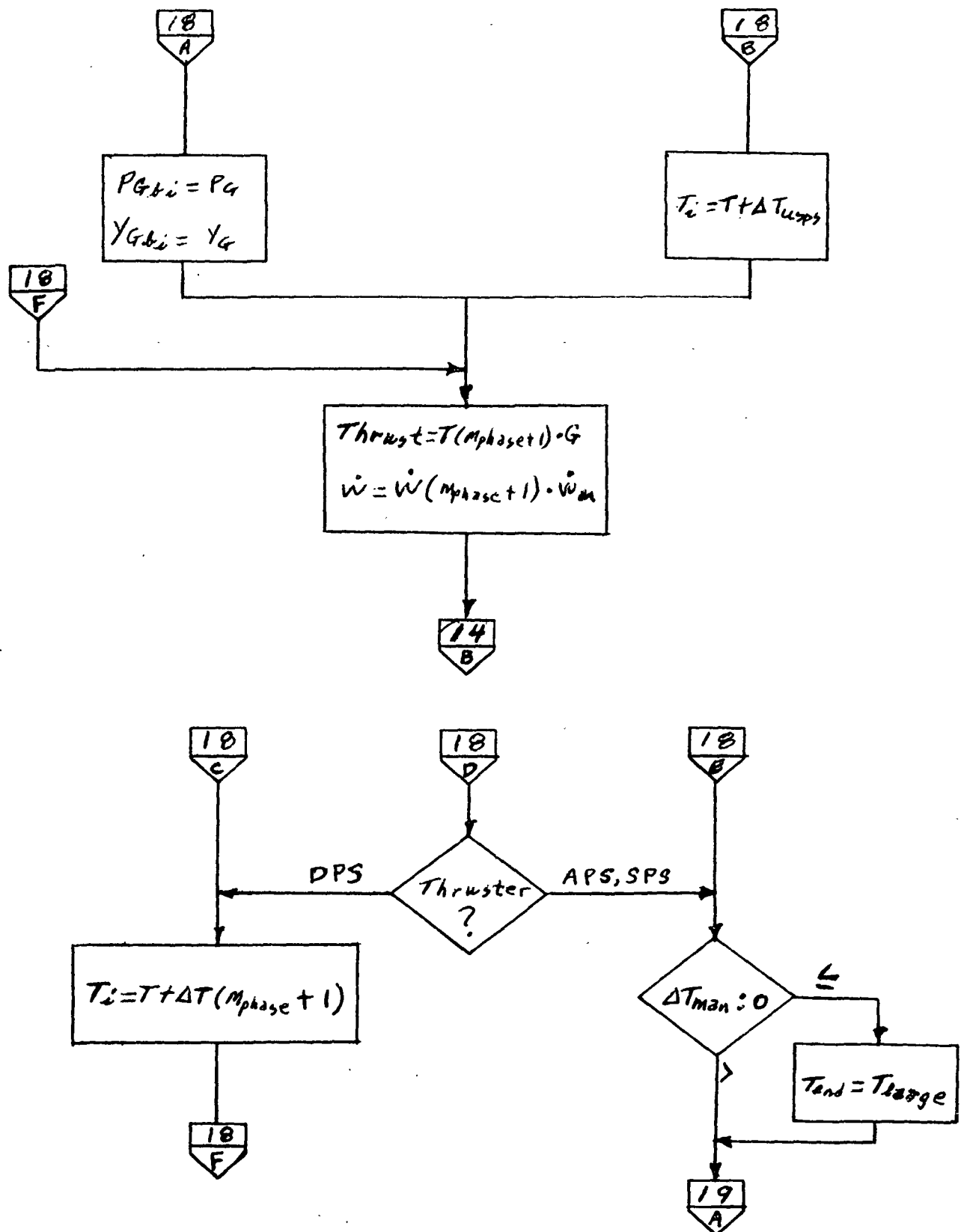


Figure PMMRKJ-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMRKJ

DATE 11/27/67 (EORP)

REV

PAGE 43 of 47

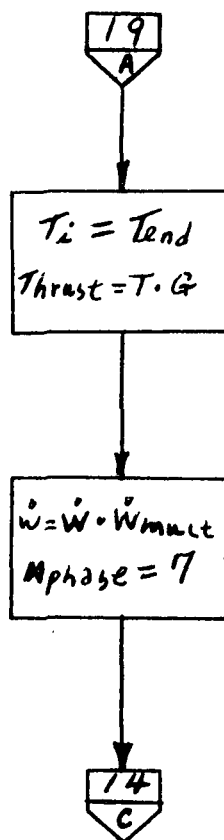


Figure PMMRKJ-1. Flowchart

(Sheet 19 of 23)

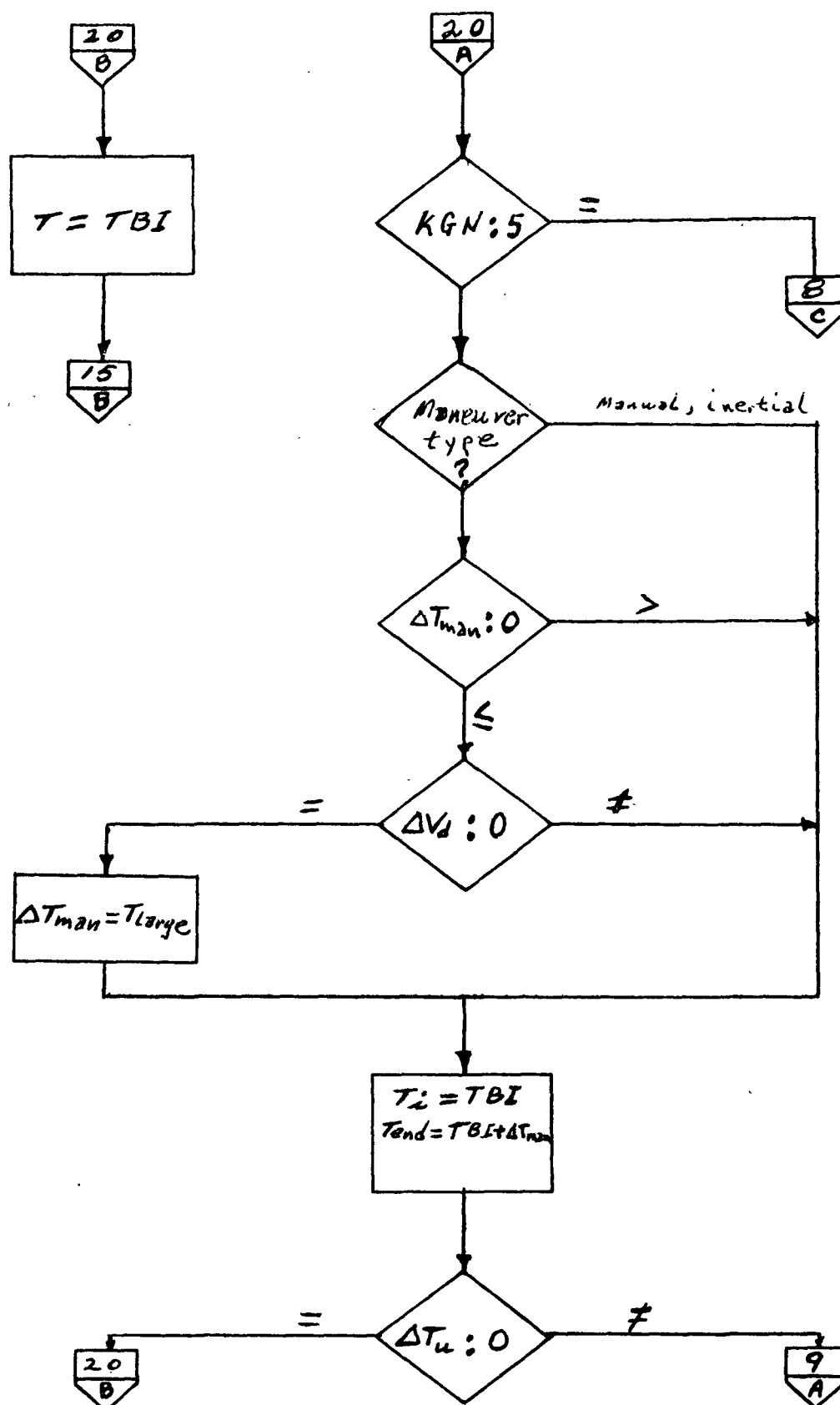


Figure PMMRKJ-1. Flowchart

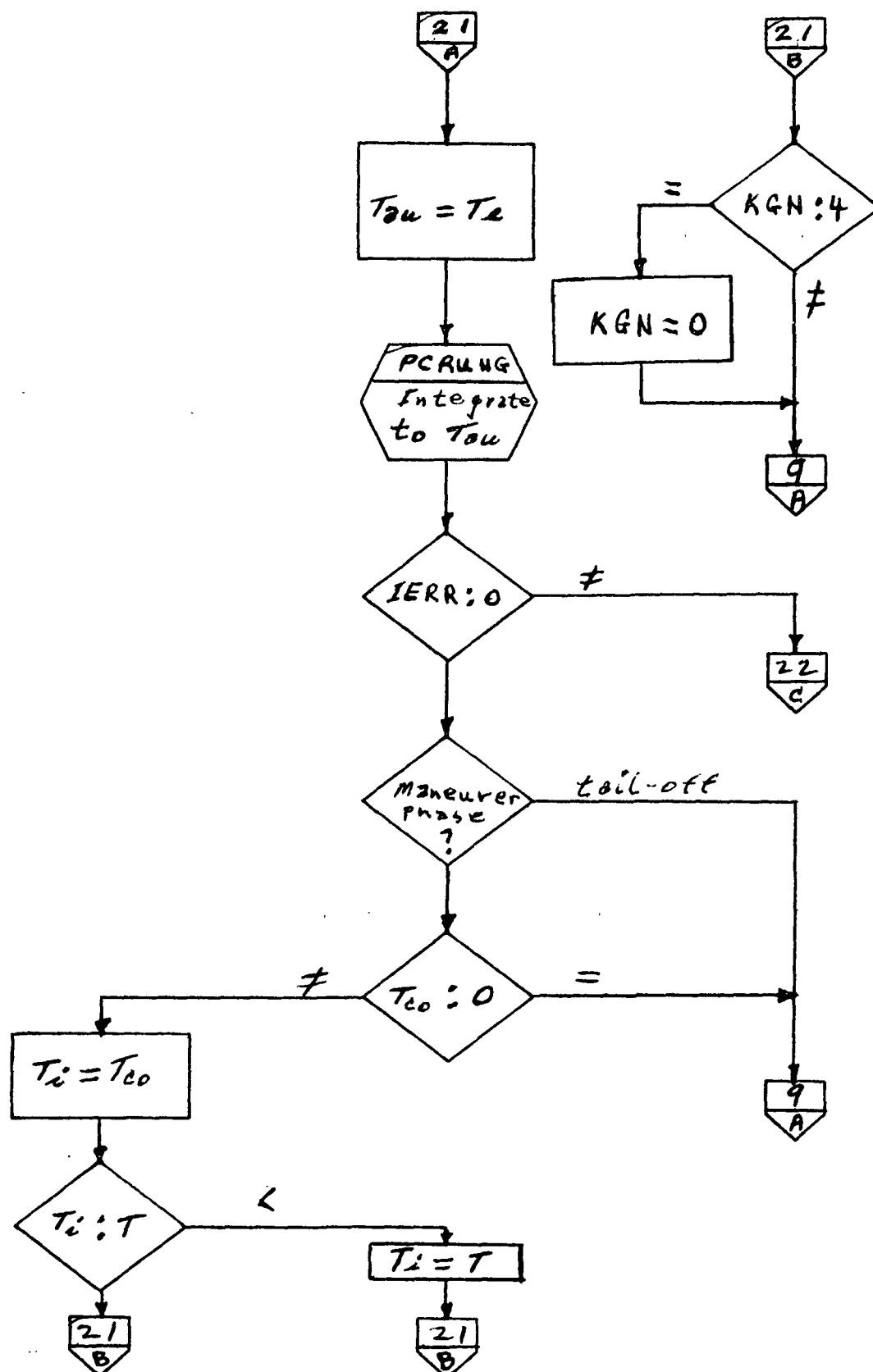


Figure PMMRKJ-1. Flowchart

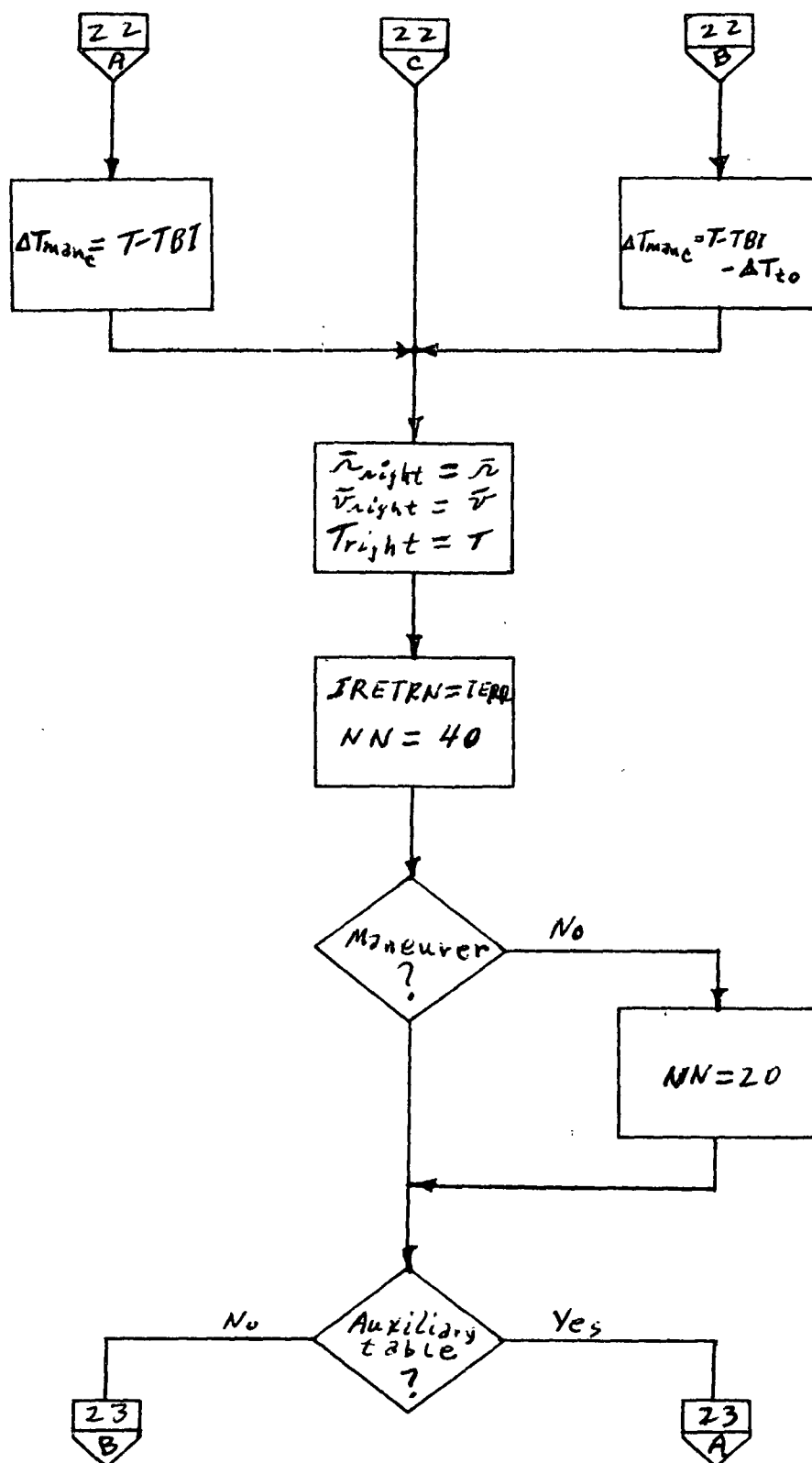


Figure PMMRKJ-1. Flowchart

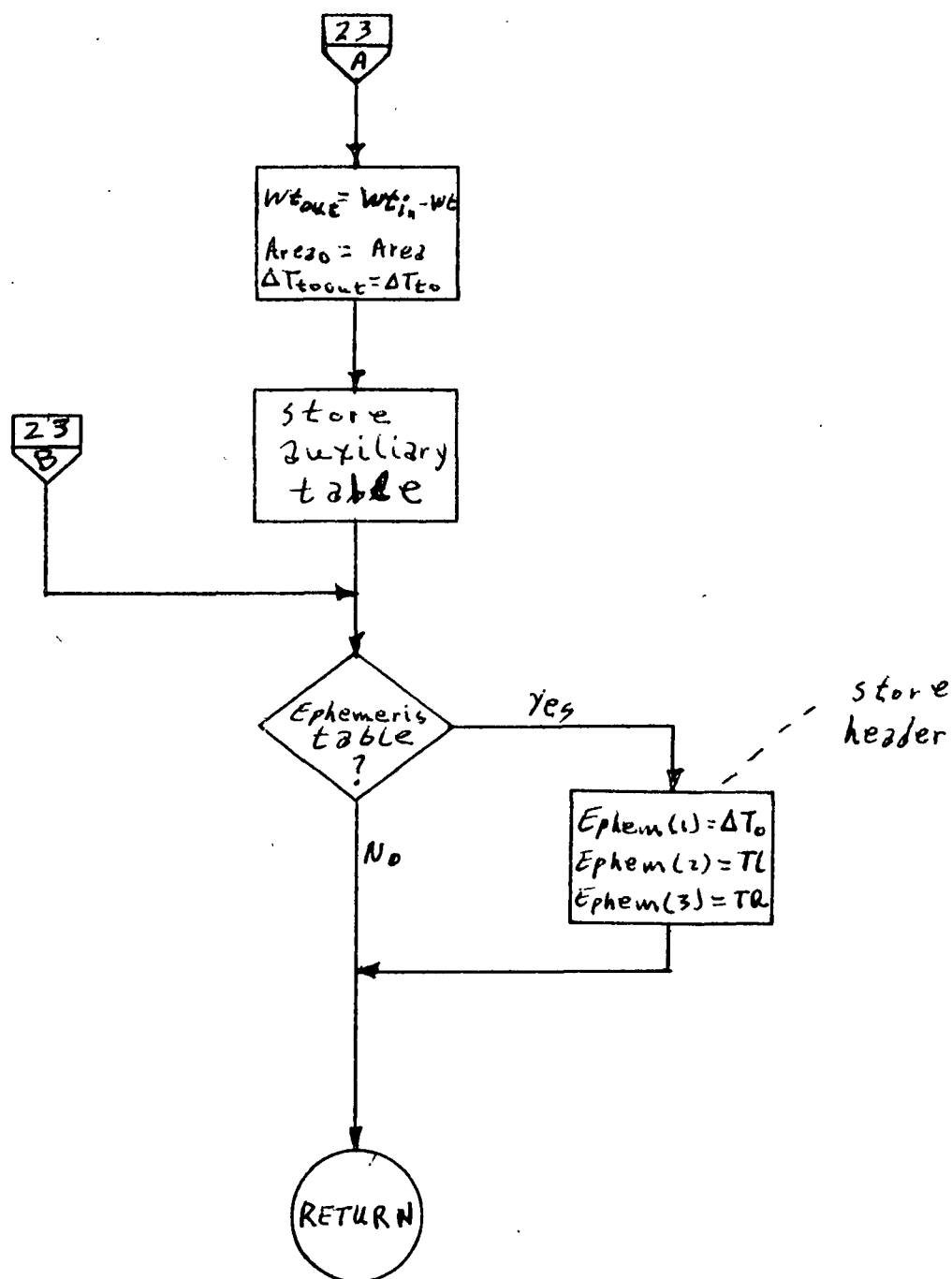


Figure PMMRKJ-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEM-AS-200 Mission Planning

2. PMMSCH
DATE 4/21/67 (205)
REV
PAGE 1 of 13

2. PMMSCH

- a. Symbol — PMMSCH
- b. Program Name — DKI Scheduling
- c. Programming Group — Rendezvous Planning
- d. Programmer — R. J. Barrett
- e. Language — FORTRAN IV
- f. Data Tables Used — None
- g. MED's Used — None
- h. Writeup Completion Date — 11/16/66
- i. Program Symbols — Table PMMSCH-1
- j. Input Parameters — Table PMMSCH-2
- k. Output Parameters — Table PMMSCH-3
- l. Flowchart — Figure PMMSCH-1.

FUNCTION

PMMSCH schedules the input in-plane maneuver points: height, phase and circularization, according to ground rules. It also sequences these points, including an input plane change point in ascending order while keeping their respective identification codes in the same order. In addition, the target vehicle lift-off delay time is computed if a phase maneuver is desired.

USAGE

- a. Calling Sequence — CALL PMMSCH (DKIBLK)
- b. Inputs — Refer to Table PMMSCH-1
- c. Outputs and Exit — Refer to Table PMMSCH-2
- d. Other Programs Used — PMMAEG, PCHAPE
- e. Data Restrictions and Limits — None
- f. Error — ICODE = 0, normal return
= -1, working counter line past circularization point
= +1, go to next mode
= +2, error return from PMMAEG
= +3, incorrect mode input

Approval	Approval
<i>R. J. Barrett</i>	<i>Arthur Orvetti</i>



METHOD

The program takes the desired controller input points (counter line maneuver values) and performs the following operations:

- a. Checks to determine if the circularization point (NSR) is past the working counter line (NA); if so, no plan is possible and the proper indicator is set.
- b. Computes maneuver points from input values according to the mode input, (phase at apogee for mode 1; phase at perigee for mode 2).
- c. Determines if NSR is within half an orbit of NA; if so, NSR is only maneuver that can be scheduled. Otherwise a height maneuver (NH) and a phase maneuver (NC1) are scheduled provided the controller wanted these maneuvers and there is adequate time to perform them.

When a phase maneuver is scheduled, the phase angle at NSR is computed and compared with the desired angle to determine if the chaser vehicle will be able to catch the target vehicle using the scheduled maneuver points. If not, the scheduled points are adjusted in an attempt to produce values that will enable the chaser to catch the target. If the adjustments fail to produce the proper catch-up rate, an error code is set and the routine returns to the calling program.

However, if the chaser was able to catch the target, whether using the scheduled points or corrected ones, the program goes on to compute the delay lift-off time, DTL. DTL is the maximum amount lift-off time (GMTLO) can be slipped and still achieve rendezvous at the input M-line.

Before returning the maneuver counter line values, the program arranges them in ascending order in an output area. In addition, the identification codes associated with each desired point are kept in a 1-1 correspondence with its scheduled value in another output area.

The plane change maneuver, NPC, if scheduled by the calling program, is included in the sequenced output array of maneuver counter line points.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEM - AS-200 Mission Planning

2. PMMSCH
DATE 4/21/67 (205)
REV
PAGE 3 of 13

Table PMMSCH-1. Program Symbols

Symbol in Program	Symbol in Flowchart	Type *	Units	Comments
M	M	E		Rendezvous counter line number
A2	A2	D	CUL	Mean semimajor axes of chaser
NA	NA	E		Working counter line number
NH	NH	E		Height maneuver line number
DTL	DTL	D	hours	Lift-off delay time
K12	K12	F		Switch indicator for phase correction
NAT	NAT	E		NA truncated
NC1	NC1	E		Phase maneuver line number
NPC	NPC	E		Plane change maneuver line number
NOM	NOM	H		Number of maneuvers scheduled
NSR	NSR	E		Circularization maneuver line number
RA2	RA2	D	CUL	Chaser radius at apogee
RP2	RP2	D	CUL	Chaser radius at perigee
SEQ	SEQ	E		4 word array for sequenced maneuver points
DELH	ΔH	D	CUL	Height difference between vehicle at NSR
DELN	ΔN	D		Intermediate value for phase computation
RAMAX	RAMAX	D	CUL	Chaser maximum apogee radius
RPMAX	RPMAX	D	CUL	Chaser maximum perigee radius
RPMIN	RPMIN	D	CUL	Chaser minimum perigee radius
DELR	ΔR	D	CUL	RP2 - RPMIN
IMIN	IMIN	F		Intermediate value for sequencing
MODE	MODE	H		Mode of DKI plan
NHPM	NH	E		Input value for NH
UNSEQ	UNSEQ	E		4 word array for unsequenced maneuver points
SEQTP	SEQTP	E		Intermediate value for sequencing

*Type
H - Fixed-point half word
F - Fixed-point full word
E - Single-precision (floating-point)
D - Double-precision (floating-point)
X - Hexadecimal



Table PMMSCH-1. Program Symbols (Cont'd.)

Symbol in Program	Symbol in Flowchart	Type *	Units	Comments
QUANI	QUANI	E		$= 2NA - [NA]_T + .5$
NCIPM	NCI'	E		Input value for NCI
ICODE	ICODE	H		Error return code
IDSEQ	IDSEQ	H		4 word array with maneuver point indicators
DELNH	ΔNH	D		Intermediate value for phase angle computation
DELNP	ΔNP	D		Intermediate value for phase angle computation
DOTLI	LI	D	radians / hour	Mean anomaly secular rate for target
THETAR	θR	D	radians	Phase angle at NSR returned from PMMAEG
THETRI	θRI	D	radians	θR - θWSR
THEWSR	θWSR	D	radians	Desired phase angle at NSR
OMEGAE	we	D	radians / hour	Rotational rate of the Earth radians/hour
DTHETH	Δθh	D		Intermediate value
DELHSR	ΔHSR	D	CUL	Desired height difference at NSR
DELNSR	ΔNSR	D		Difference between NSR and M
DEL RPM	ΔRPM	D		Intermediate value
IDTEMP	IDTEMP	F		Temporary storage
MLDATA		D		Data area for PCHAPE
THETSR	θSR	D	radians	Actual phase angle at NSR
IDUNSQ	IDUNSQ	H		4 word array with maneuver identification codes

*Type
 E — Single-precision (floating-point)
 H — Fixed-point half word
 D — Double-precision (floating-point)
 F — Fixed-point full word
 X — Hexadecimal

Table PMMSCH-1. Program Symbols (Cont'd.)

Symbol in Program	Symbol In Flowchart	Type *	Units	Comments
PI	π	D	radians	
TWOPI	2π	D	radians	
DKIBLK		Mixed	mixed	Input DKI data table
DKI		Mixed	mixed	DKI data table for PMMSCH from DKIBLK
AEGOUT		D		54 words for temporary AEG output
INHALF		H		} Equivalued arrays for DKI
SINGLE		E		
INFULL		F		
MHEAD		H		Main header input to AEG

*Type
H — Fixed-point half word
F — Fixed-point full word
E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

Table PMMSCH-2. Input Parameters through DKI Table

[illegible]

Table PMMSCH-3. Output Parameters through DKI Table

[illegible]

^aType

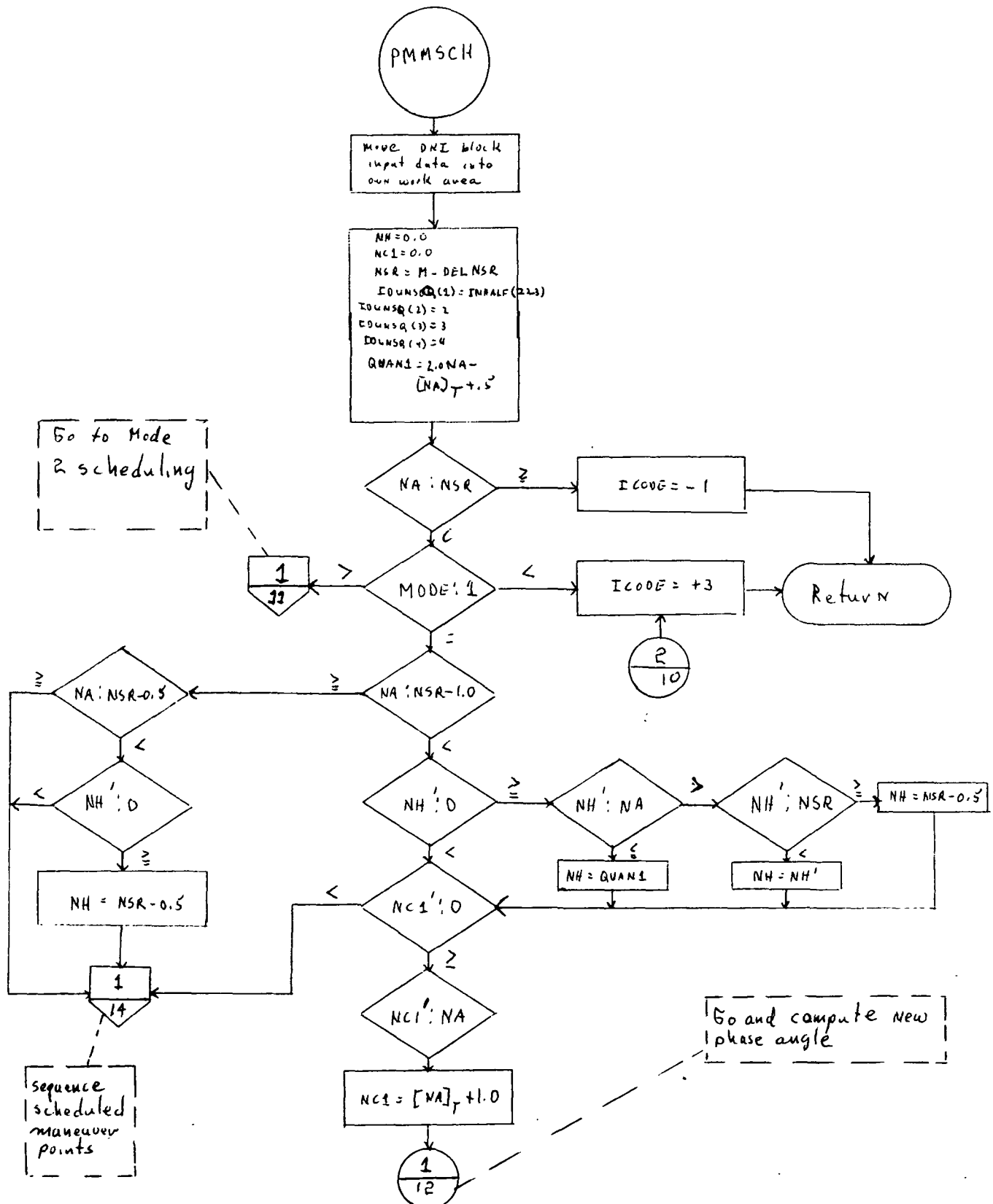


Figure PMMSCH-1. Flowchart

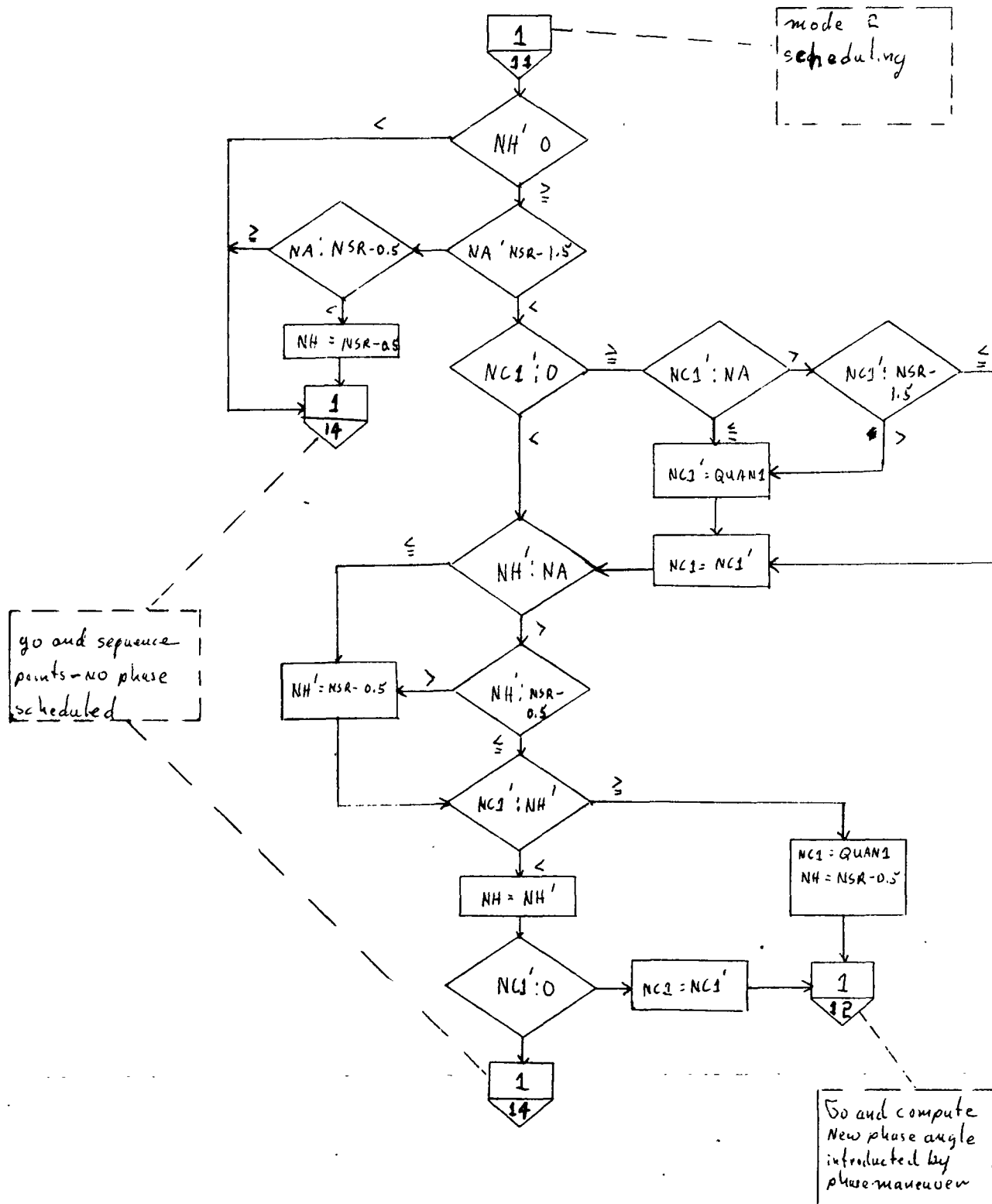


Figure PMMSCH-1. Flowchart

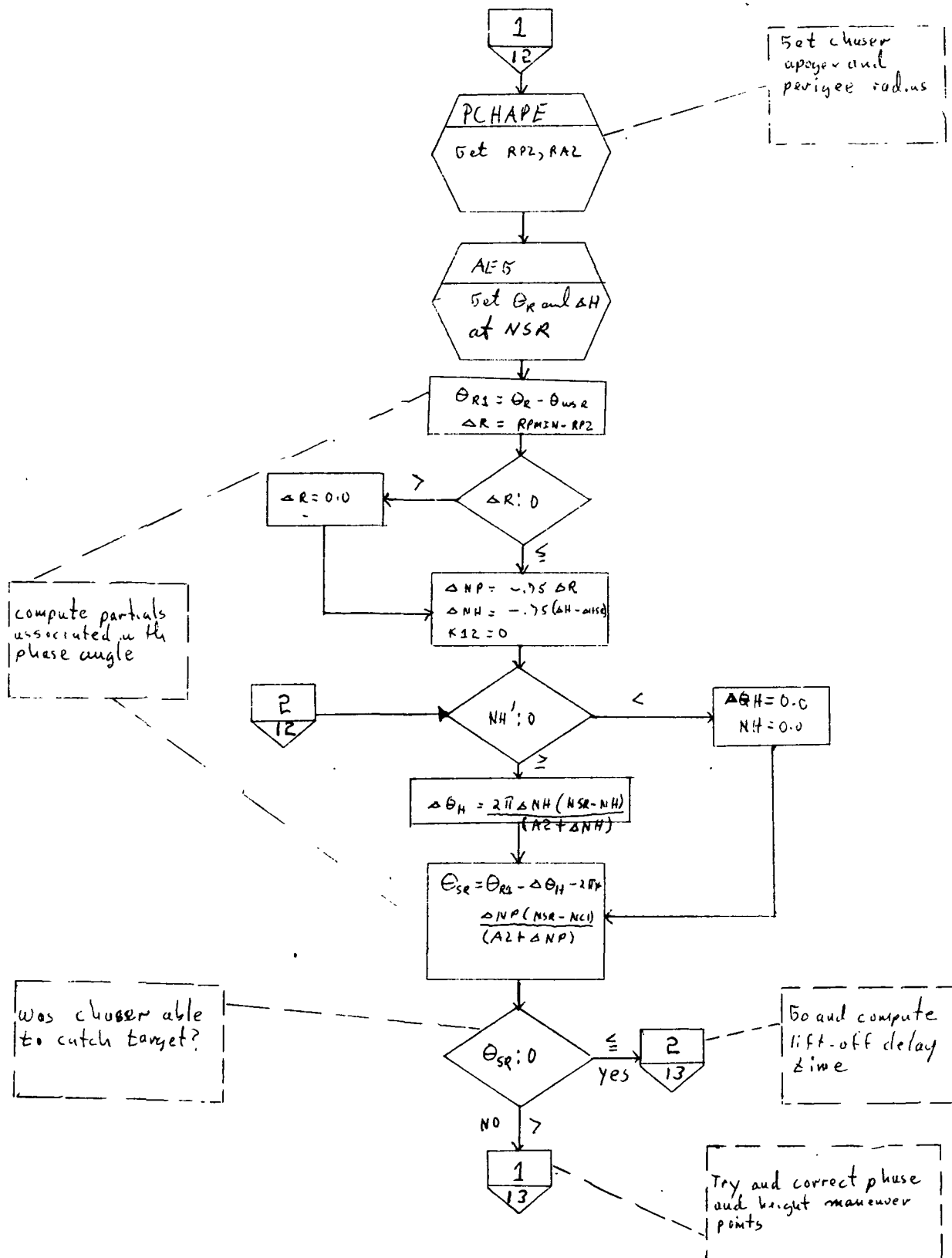


Figure PMMSCH-1. Flowchart

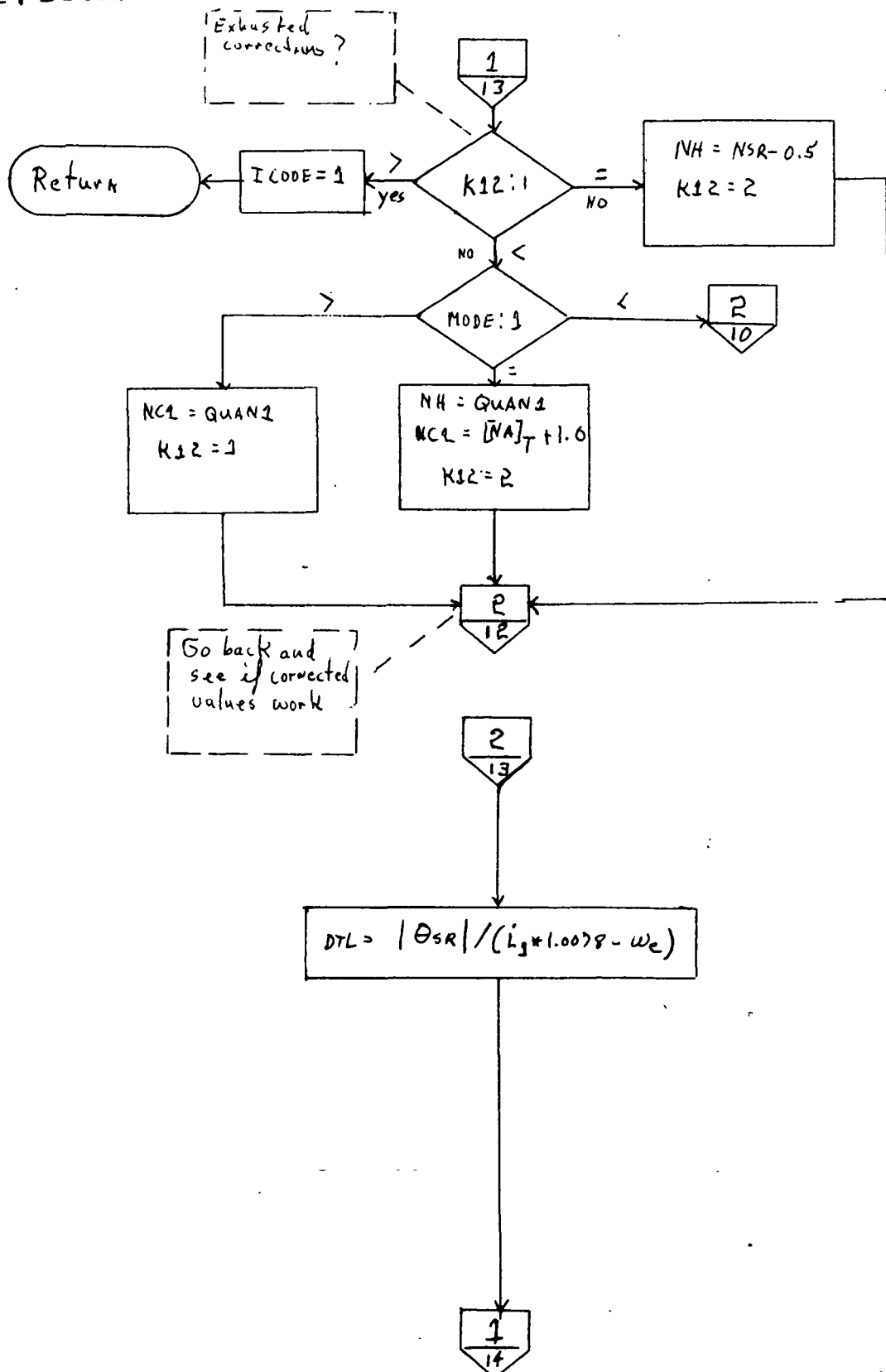


Figure PMMSCH-1. Flowchart

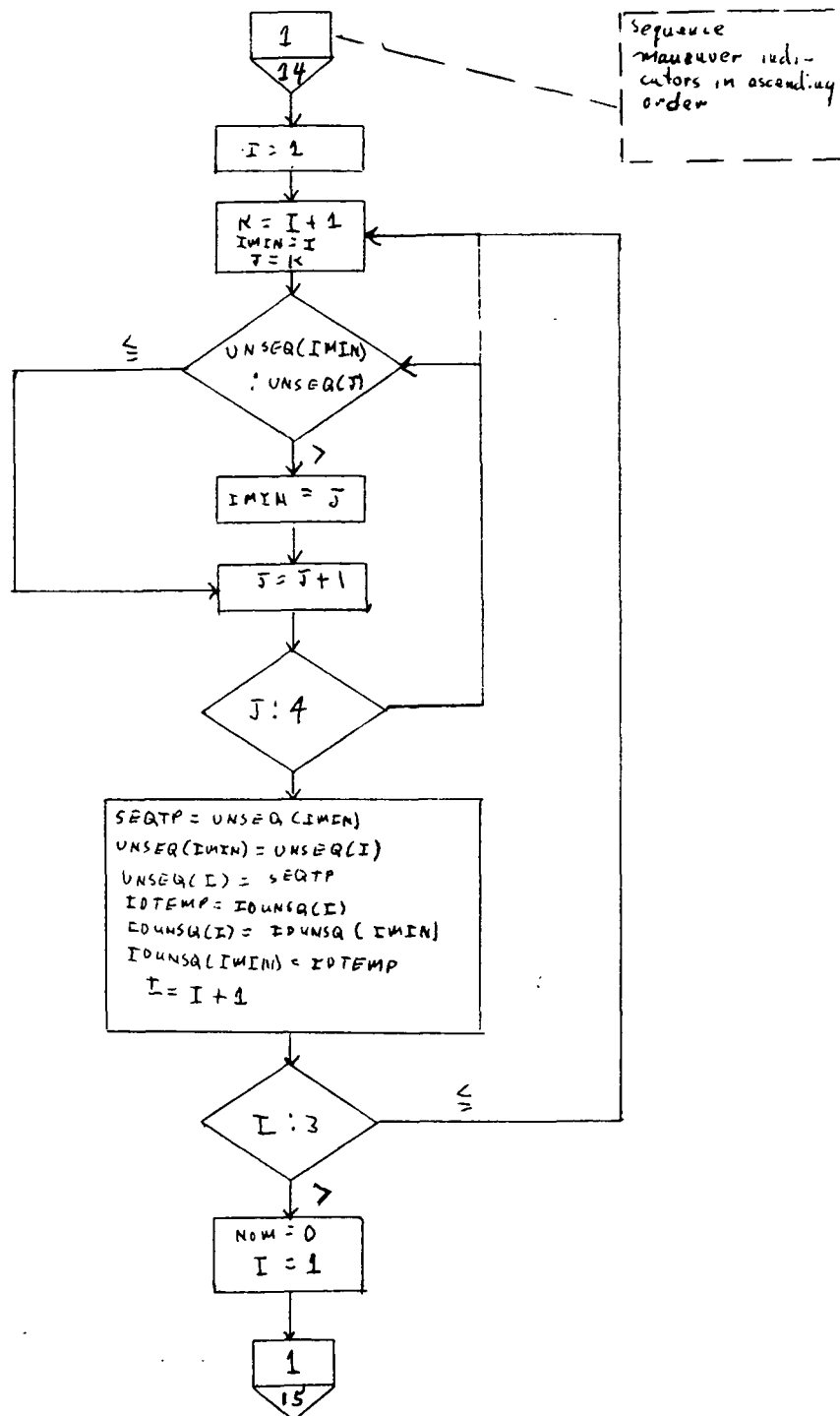


Figure PMMSCH-1. Flowchart

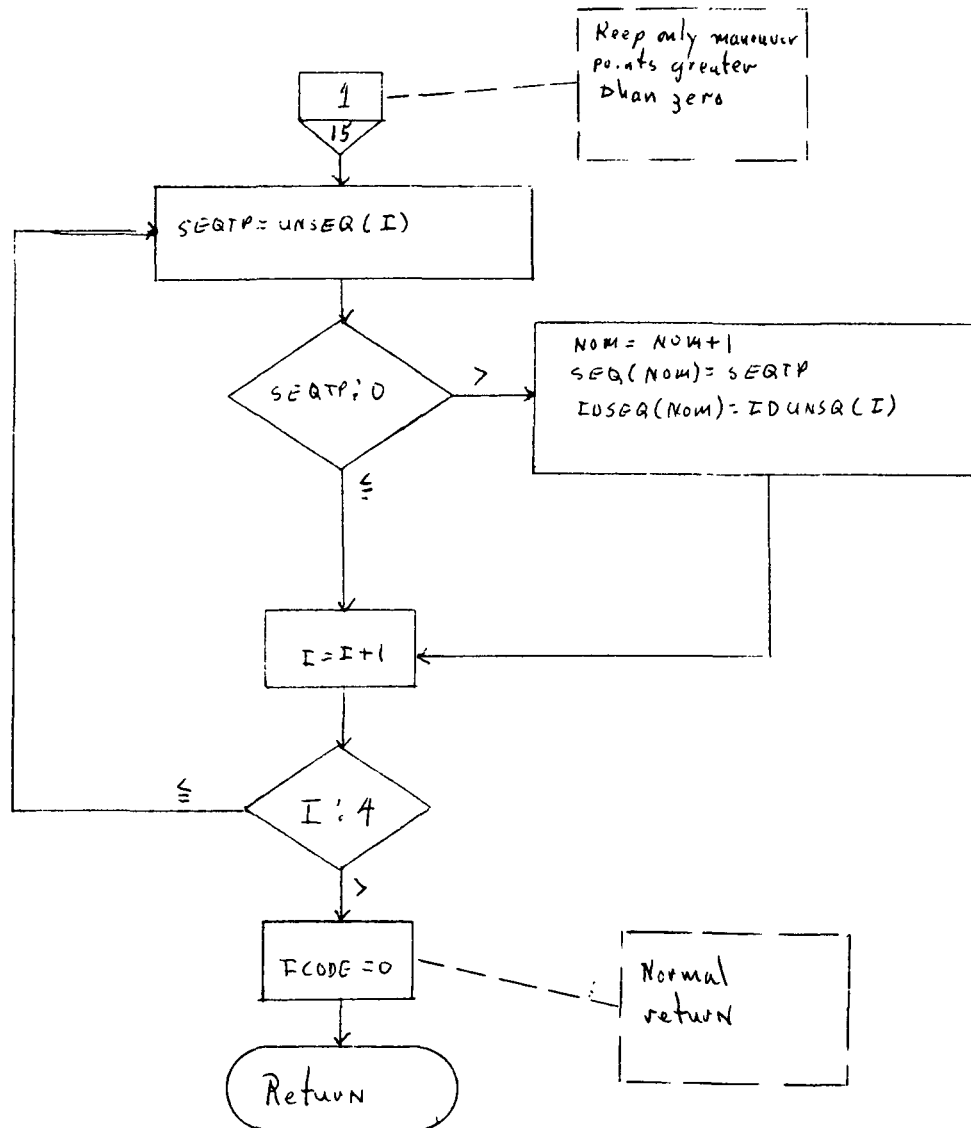


Figure PMMSCH-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS
 2. PMMSPQ
 DATE 2/28/68 (EORP)
 REV
 PAGE 1 of 31

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

- a. Symbol — PMMSPQ
- b. Program Name — Coelliptic Sequence
- c. Programming Group — Rendezvous Planning (Mission Planning)
- d. Programmer(s) — M. R. Secondo
- e. Language — FORTRAN and Assembler
- f. Data Tables Used — None
- g. MED's Used — P80, K05, K01
- h. Writeup Completion Date — 11/16/67
- i. System Parameters — Table PMMSPQ-1.
- j. Flowchart — Figure PMMSPQ-1.

FUNCTION

The Coelliptic Sequence program is an iterative maneuver scheme which computes the incremental velocities needed at the scheduled midcourse maneuvers to satisfy some relative vehicle positioning at the Terminal Phase Initiation point (TPI). Maneuvers of the sequence include the Concentric Sequence Initiation maneuver (CSI), the Constant Height Difference maneuver—also termed the Coelliptic maneuver (CDH), and a Plane Change Maneuver (PC).

USAGE

- a. Calling Sequence — LINK EP = PMMSPQ, PARAM = (DKI, ELEM)
- b. Inputs

Input to the Coelliptic Sequence program is by MED request and is a function of the options requested for a plan.

The initial phase angle may be defined in one of three ways: (1) the phase angle having the smallest absolute value, or (2) as the positive phase angle, or (3) as the negative phase angle. Different solutions will result from the iterative procedure dependent upon this definition.

At the CSI maneuver point, if it is scheduled, the time of CSI is always input. No radial incremental velocity component is computed and the CSI maneuver is the means of adjusting the height difference at CDH or the line-of-sight at TPI.

In defining the CDH maneuver point there are three options: (1) it may occur at a desired apsis arrival after CSI, or (2) after some desired travel angle from CSI, or (3) it may be scheduled at an input time. If the CSI maneuver is not scheduled, CDH has to be scheduled at a time. The CDH maneuver may not be deleted unless the CSI is also deleted.

Approval	Approval
<i>M. R. Secondo</i>	<i>A. Amicetti</i>

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMSPQ

DATE 2/28/68

REV

PAGE 2 of 31

If no plane change maneuver is scheduled, an option exists to apply the CSI and CDH maneuvers in-plane or parallel to the target plane.

A plane change maneuver may be scheduled any time prior to TPI by inputting a threshold option and a subsequent common node. The options for the threshold include inputting a time, or it may be designated to be either CSI or CDH time. Either the active or target vehicle may do the plane change maneuver.

There are two options for defining the desired relative vehicle positioning at TPI: (1) so that the active vehicle have a desired line-of-sight angle to the target at the desired time of TPI or (2) so that the active vehicle have a desired differential altitude at CDH with respect to the target vehicle. The first option does not use an input desired differential altitude; that is, the iteration is based on the line-of-sight at TPI and therefore the height difference at CDH is a function of that angle and the TPI time. When iterating on a differential altitude at CDH, the program will ignore the option which defines TPI and will find TPI as a function of the desired height difference and line-of-sight angle.

With the option to iterate on a line-of-sight, the desired TPI time can be input or it may be computed as the time of target arrival at a desired sunlighting condition or at a desired earth longitude.

Each of the input options from the MED request is stored in the DKI common area (the writeup of PMMDKI defines the input variables).

- c. Outputs and Exit — The elements before, and after, each maneuver are stored in the DKI area as well as any error which may have been detected.
- d. Other Programs Used — PMMAEG, PMMDAN, PMMPNE, GIMIKC, GIMKIC, DSQRT, DARCOS, DARSIN, DATAN, DTAN, DSIN, DCOS, DATAN2
- e. Data Restrictions and Limits — None
- f. Errors — INERR = 0, if no errors exist
INERR = 1, if the Analytic Ephemeris Generator (AEG) input is unacceptable
INERR = 2, if there is a failure to converge and the plan is retained

INERR = 3, if there is a time violation
INERR = 5, if there is a ΔV violation
INERR = 8, if there is a failure to converge and the plan is deleted.
INERR < 0, AEG RETURN other than -1 or -2
INERR = 94, For failure to obtain the input elevation

METHOD

Upon entry to the coelliptic program, the initial phase angle is adjusted according to the input option and TPI is determined. Then the adjusted elements are restored and the iteration begins with the first maneuver in the sequence. As the time of the plane change may occur any time prior to TPI, a check must be made prior to each maneuver to ensure that if the plane change is scheduled, it will be performed in the proper sequence.

In the iteration, ΔV_{CSI} is a function of the Coelliptic height difference at CDH or the line-of-sight at TPI (a function of the phase angle). The method of evaluating ΔV_{CSI} involves the solution of several differentials.

Considering V_X to be the horizontal CSI velocity and V_Y to be the radial CSI velocity, the total velocity can be expressed as

$$V^2 = V_X^2 + V_Y^2$$

Differentiating,

$$v dv = v_x dv_x + v_y dv_y$$

Since CSI is a horizontal maneuver, $dv_y = 0$. Therefore:

$$v dv = v_x dv_x$$

at the CSI point,

$$V^2 = \mu \left(\frac{2}{R} - \frac{1}{a} \right)$$

Differentiating, and with $dR = 0$,

$$da = \frac{2a^2}{\mu} v_x dv_x = k_a dv_x \quad \text{where } k_a = \frac{da}{dv_x}$$

IBM**RTCC APOLLO PROGRAMMING SYSTEMS**

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMSPQ
 DATE 2/28/68 (EOR)
 REV
 PAGE 4 of 31

For the eccentricity, e ,

$$e^2 = 1 - \frac{R^2 V_x^2}{\mu a}$$

and the differential

$$de = \frac{R^2 V_x \left(k_{a x} - 2a \right)}{2\mu a^2 e} dv_x = k_e dv_x \quad \text{where } k_e = \frac{de}{dv_x}$$

$$\text{With } P = \frac{R^2 V_x^2}{\mu},$$

$$dp = \frac{2R^2 V_x}{\mu} dv_x = k_p dv_x$$

To find the differential of the true anomaly with respect to V_x it is necessary to differentiate the equation

$$\cos f = \frac{P - R}{eR} \quad \text{where } f \text{ is evaluated at CSI,}$$

$$df = \frac{R(P - R) k_e - eRk_p}{e^2 R^2 \sin f} dv_x = k_f dv_x$$

An alternate equation is used if f is near zero; that is,

$$\dot{R} = e \sqrt{\frac{\mu}{P}} \sin f$$

Differentiating,

$$df = \frac{\tan f}{2ep} (ek_p - 2pk_e) dv_x = k_f dv_x$$

The cosine of the eccentric anomaly is

$$\cos E = \frac{a - R}{ea}$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

and the differential is

$$d_E = \frac{a(a-R)k_e - eRka}{e^2 a^2 \sin E} dv_x = k_e dv_x$$

Once again, if E is near zero, the following equation is evaluated for E

$$\tan \gamma = \frac{\dot{R}}{V_x} = \frac{e \sin E}{(1-e)^{2/2}}$$

$$\text{and } dE = \frac{R e^3 - (e - v_x k_e)}{(1-e)^{2/2} e^2 v_x^2 \cos E} dv_x = k_E dv_x$$

For the mean anomaly

$$M = E - e \sin E$$

the differential is

$$d_M = [(1 - e \cos E) k_E - k_e \sin E] dv_x = k_M dv_x$$

Mean motion expressed as a function of a is

$$n = \sqrt{\mu/a^3}$$

and this gives

$$dn = (-3/2 \mu^{1/2} a^{-5/2} k_a) dv_x = k_n dv_x$$

Letting the subscript "T" denote target vehicle parameters, the catchup rate can be expressed as

$$\Delta n = n - n_T$$

and

$$d(\Delta n) = dn = k_n dv_x$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMSPQ
DATE 2/28/68 (EOR)
REV
PAGE 6 of 31

For CDH at an apsis and if N' is the desired CDH apsis number from CSI, N is set:

- (1) for $M < \pi$ then $N = N'$
- (2) for $M \geq \pi$ then $N = N' + 1$

The time interval between CSI and CDH can then be expressed as

$$t_1 = \frac{N\pi - M}{n}$$

and

$$dt_1 = \left(\frac{-nk_M - (N\pi - M)k_n}{n^2} \right) dv_x$$

Denoting CDH parameters by primes and considering f' as CDH true anomaly

then

$$df' = 0 = k'_f dv_x \quad \text{for CDH at an apsis.}$$

For CDH at an input time

$$t_1 = T_{CDH} - T_{CSI}$$

$$dt_1 = 0 = k_t dv_x$$

so that

$$M' = M + t_1 n$$

$$dM' = (k_m + t_1 k_n) dv_x = k'_m dv_x$$

Since

$$f' = M' + 2e \sin M' + \dots$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMSPQ
DATE 2/28/68 (EORP)
REV
PAGE 7 of 31

df' may be approximated as

$$df' = \left[(1 + 2e \cos m' + 5/2 e^2 \cos 2m') k_{m'} + (2 \sin m' + 5/2 e \sin 2m') k_e \right] dv_x$$

The case where CDH is desired at an input travel angle from CSI, ΔU_D ,

then

$$f' = f + \Delta U_D$$

and

$$df' = df = k_f dv_x = k_f' dv_x$$

The mean anomaly can be expressed in terms of the true anomaly as

$$\Delta M = \Delta U_D - 2e (\sin f' - \sin f) + 3/4 e^2 (\sin 2f' - \sin 2f) + \dots$$

$$d(\Delta M) = \left\{ \left[3/2 e (\sin 2f' - \sin 2f) - 2(\sin f' - \sin f) \right] k_e + \left[3/2 e^2 (\cos 2f' - \cos 2f) - 2e(\cos f' - \cos f) \right] k_f \right\} dv_x$$

$$d(\Delta M) = k_{\Delta M} dv_x$$

Now,

$$t_1 = \frac{\Delta M}{n}$$

so that

$$dt_1 = \frac{nk_{\Delta M} - \Delta M k_n}{n^2} dv_x = k_t dv_x$$

If R' is the CDH radius and

$$R' = \frac{P}{1 + e \cos f'}$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMSPQ
DATE 2/28/68 (EOR)
REV.
PAGE 8 of 31

then

$$dR' = \frac{R'}{P} [k_p - R'(k_e \cos f' - e k_f \sin f')] dv_x = k_{R'} dv_x$$

The final differential dR' is used in the computation of a first guess at ΔV_{CSI} if a request is made to iterate on a height at CDH; that is,

$$\Delta V_{CSI} = \frac{\Delta R}{k_{R'}} \text{ where } \Delta R \text{ is the difference between the input desired}$$

height difference and the actual height difference at CDH. Subsequent estimates of ΔV_{CSI} are linear interpolations between previous estimates and the corresponding CDH height miss differences.

If a request was made to iterate on a line-of-sight angle at TPI, then $K_{R'}$ is used as follows: The mean motion of the chaser in the Coelliptic orbit is n' and the catchup rate is

$$\Delta n' = n' - n_T$$

$$d(\Delta n') = dn'$$

where

$$dn' = (-3/2 \mu^{1/2} a'^{-5/2} K_{R'}) dv_x = K_{n'} dv_x$$

The phase angles at CDH and TPI are

$$\theta_{CDH} = \theta_{CSI} - \Delta n t_1$$

$$\theta_{TPI} = \theta_{CDH} - \Delta n' t_2$$

where

$$t_2 = T_{TPI} - T_{CDH}$$

With CSI and TPI as fixed points, then

$$dt_2 = -dt_1$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

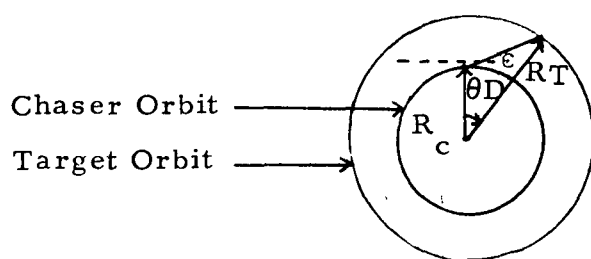
2. PMMSPQ
DATE 2/28/68 (EORP)
REV.
PAGE 9 of 31

and

$$d\theta_{TPI} = [(\Delta_n' - \Delta n) k_t - t_1 k_n - t_2 k_n'] dv_x = k_{\theta R} dv_x$$

The desired phase angle at TPI is a function of the Coelliptic height difference and the desired line-of-sight angle, ϵ .

From Figure 1, the formula for the desired phase angle can be developed.



- ϵ - Desired Line-of-Sight Angle
- R_c - Chaser Radius at TPI
- R_T - Target Radius at TPI
- θ_D - Desired Phase Angle at TPI

Using the Law of Sines,

$$\frac{\sin(\theta_D + \epsilon - \frac{\pi}{2})}{R_c} = \frac{\sin(90 + \epsilon)}{R_T}$$

$$= \frac{\cos \epsilon}{R_T}$$

$$\theta_D = \frac{\pi}{2} - \epsilon - \sin^{-1} \left(\frac{R_c \cos \epsilon}{R_T} \right)$$

Taking the differential

$$d\theta_D = \frac{-K' R_c \cos \epsilon}{R_T \sqrt{1 - \frac{R_c^2 \cos^2 \epsilon}{R_T^2}}} dv_x = k_{\theta D} dv_x$$

Considering θ_A and θ_w to be, respectively, the actual phase angle and the desired phase angle at TPI due to changing CSI horizontal velocity by

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMSPQ
DATE 2/28/68 (EORI)
REV
PAGE 10 of 31

V_X , then these quantities can be expressed as

$$\theta_A = \theta_{TPI} + d\theta_{TPI} = \theta_{TPI} + K_{\theta R} dv_x$$

$$\theta_w = \theta_D + d\theta_D = \theta_D + K_{\theta D} dv_x$$

Equating these two expressions and solving for dv_x

$$dv_x = \frac{\theta_{TPI} - \theta_D}{K_{\theta D} - K_{\theta R}} = \Delta V_{CSI}$$

and this becomes the first guess at the required CSI velocity increment to achieve the desired line-of-sight angle at TPI.

As in the case of iterating on a height difference at CDH, subsequent estimates of ΔV_{CSI} are linear interpolations between previous estimates and the corresponding TPI phase angle discrepancies.

When a solution is found, the sequence of maneuvers is checked for violations of constraints. Subsequent maneuvers should be spaced by some minimum time difference and the total ΔV expenditure should not exceed the initial quantity.

The final portion of the program stores the AEG elements before and after each maneuver in the elements table which was passed in the parameter list.



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMSPO
DATE 2/28/68 (EORP)
REV
PAGE 11 of 31

Table PMMSPO-1. System Parameters

Symbol in Program	Symbol in Master Tape	Type *	Units	Comments
$\pi/2$	MCMHFI	D	rad	
π	MCMPPIR	D	rad	
2π	MCMTPI	D	rad	
MU	MCEMUU	D	Er^3/hr^2	Earth Gravitation Constant
$\sqrt{\text{MU}}$	MCERMU	D	Er^3/hr^2	Square Root of Mu
RCS	MCTSPX	D	hr	RCS Specific Impulse
SPS	MCTSIP	D	hr	SPS Specific Impulse

* Type
H - Fixed-point half word
F - Fixed-point full word
E - Single-precision (floating-point)
D - Double-precision (floating-point)
X - Hexadecimal

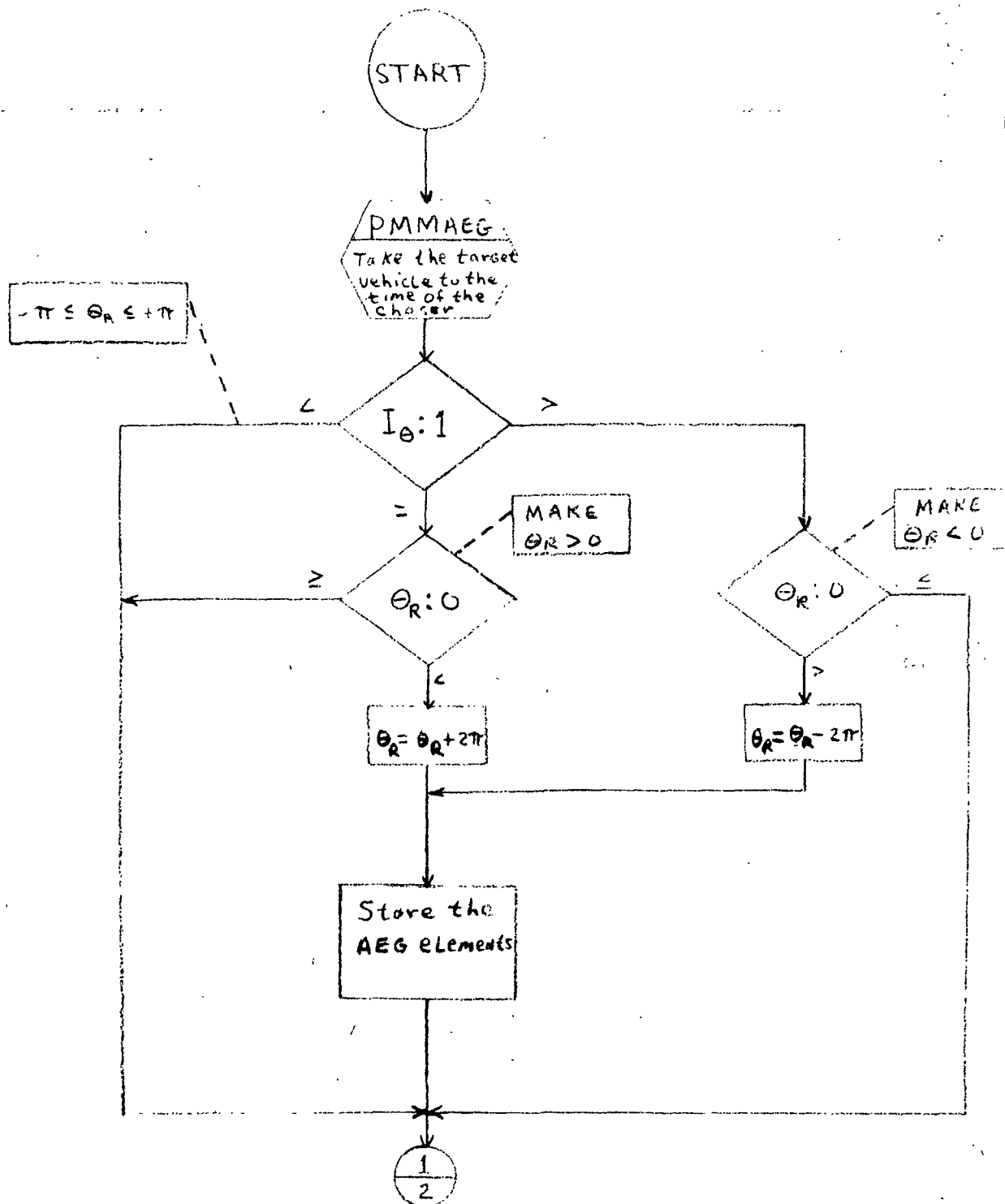


Figure PMMSPO-1. Flowchart

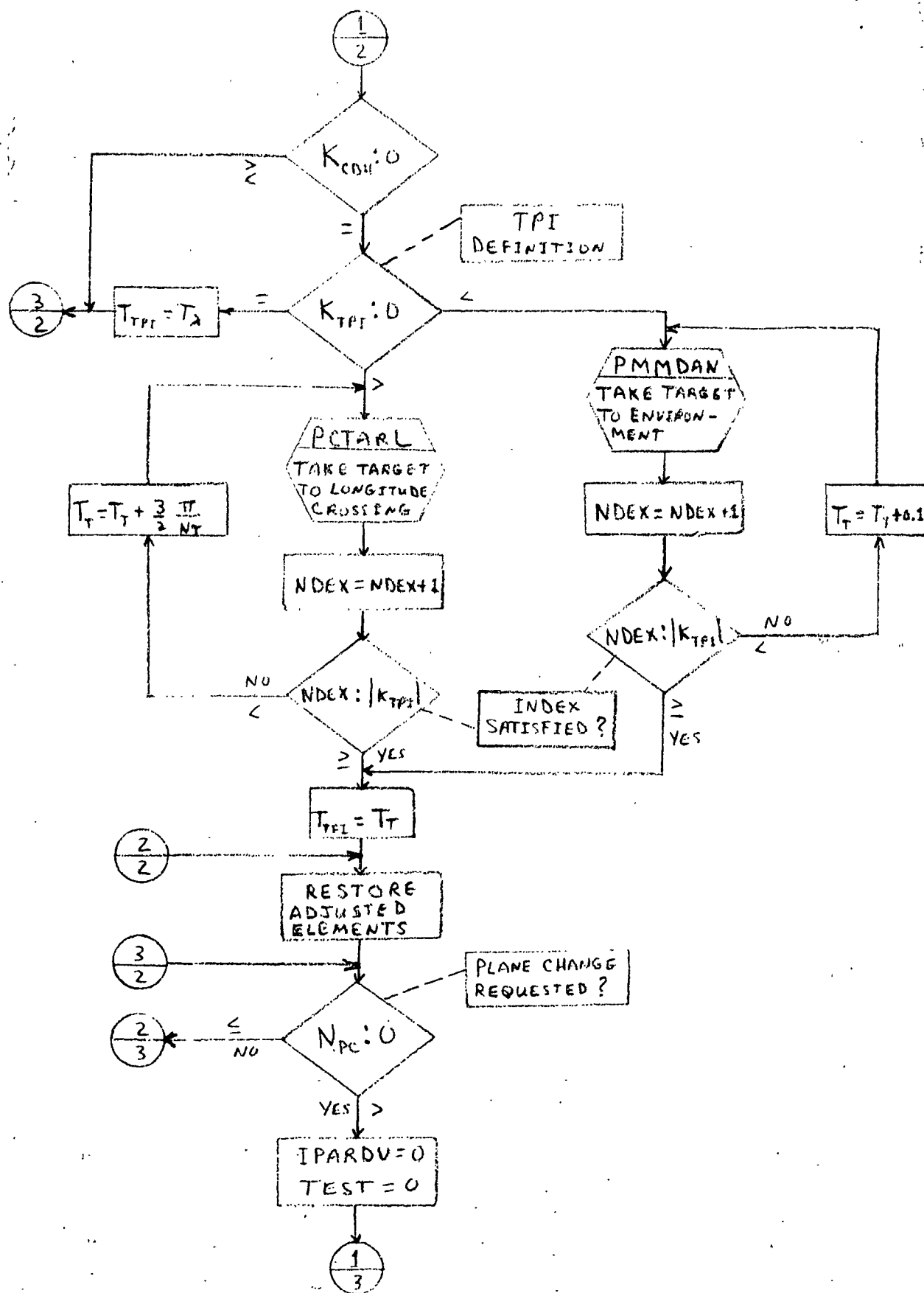


Figure PMMSPO-1. Flowchart

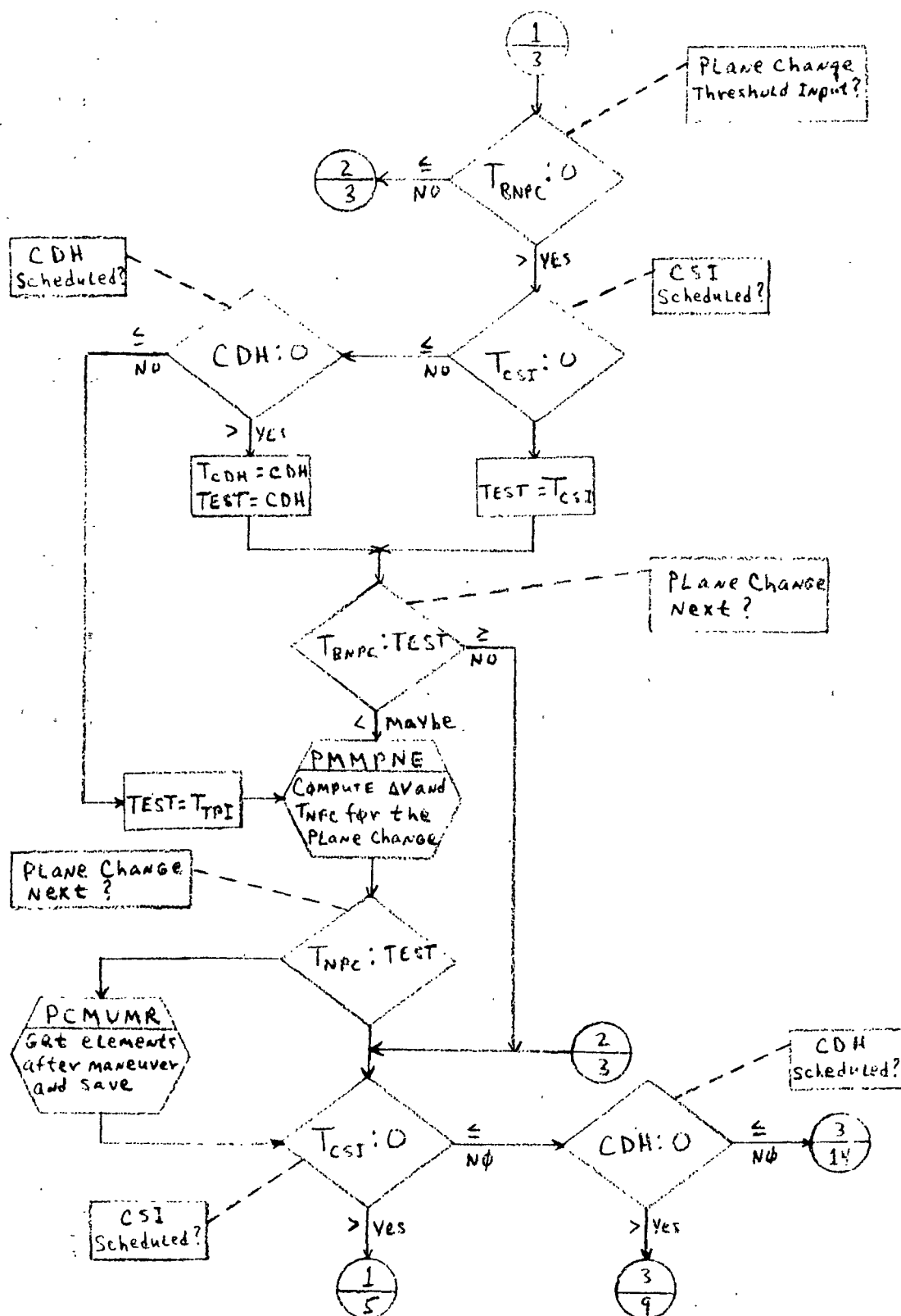


Figure PMMSPO-1. Flowchart

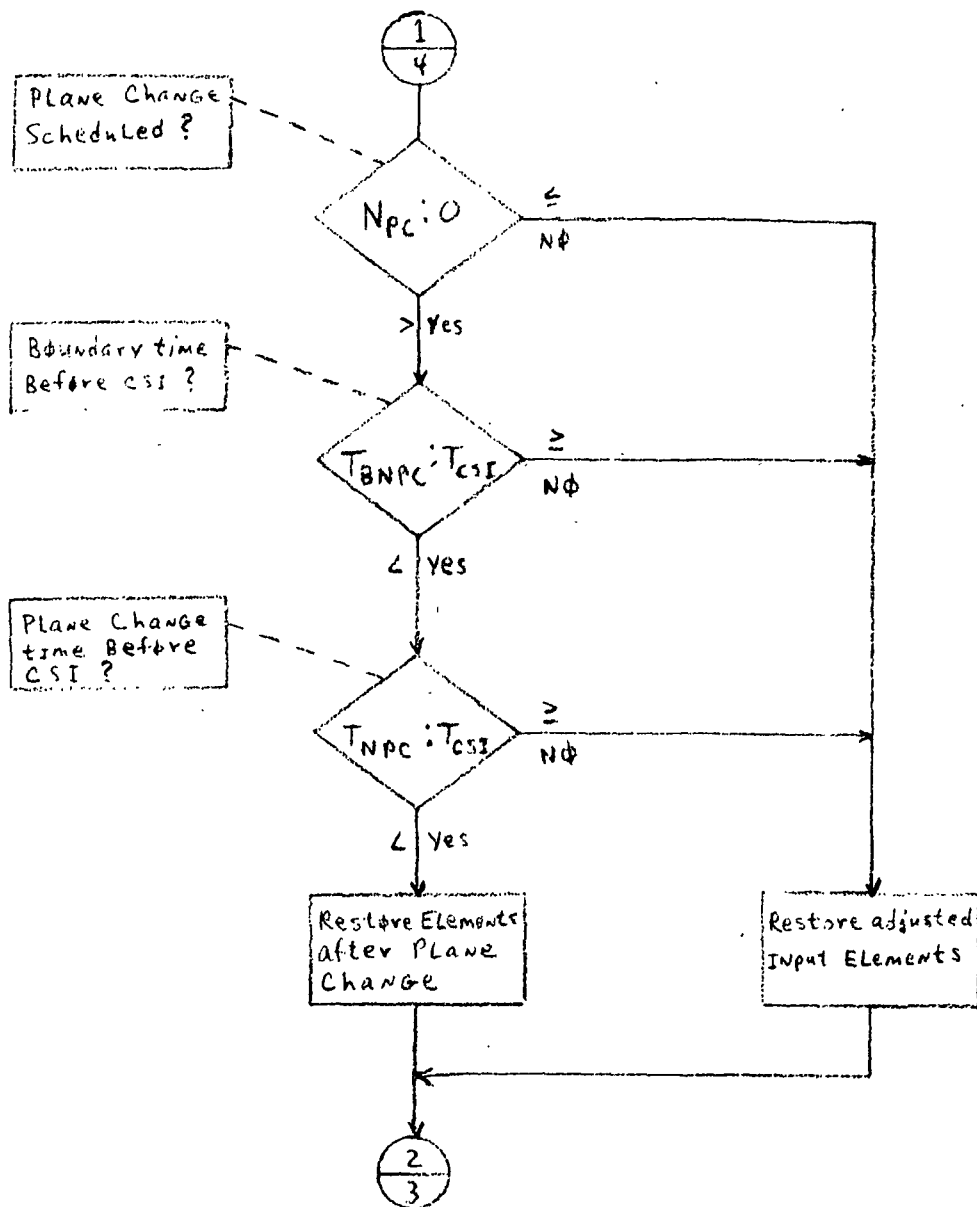


Figure PMMSPO-1. Flowchart

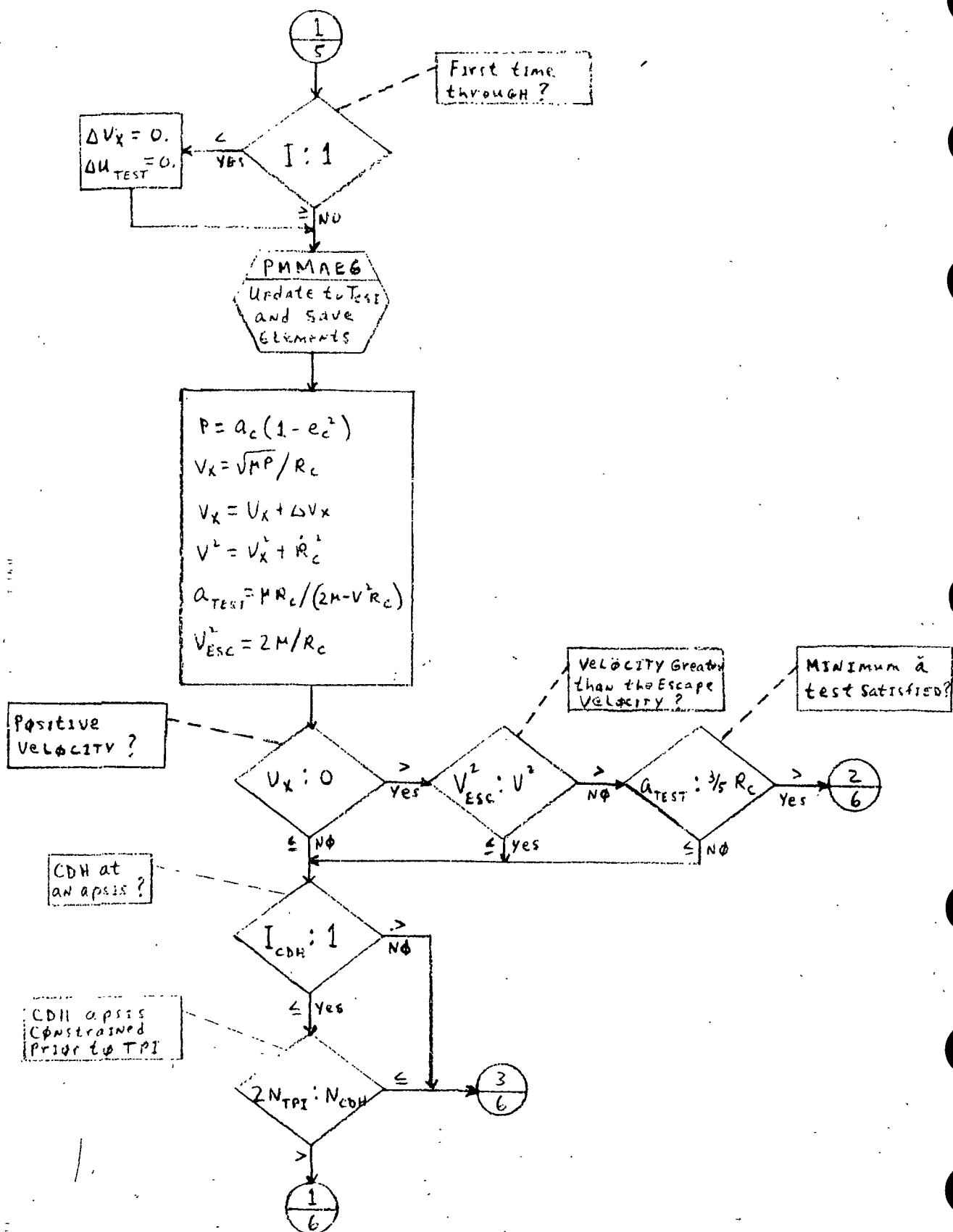


Figure PMMSPO-1. Flowchart

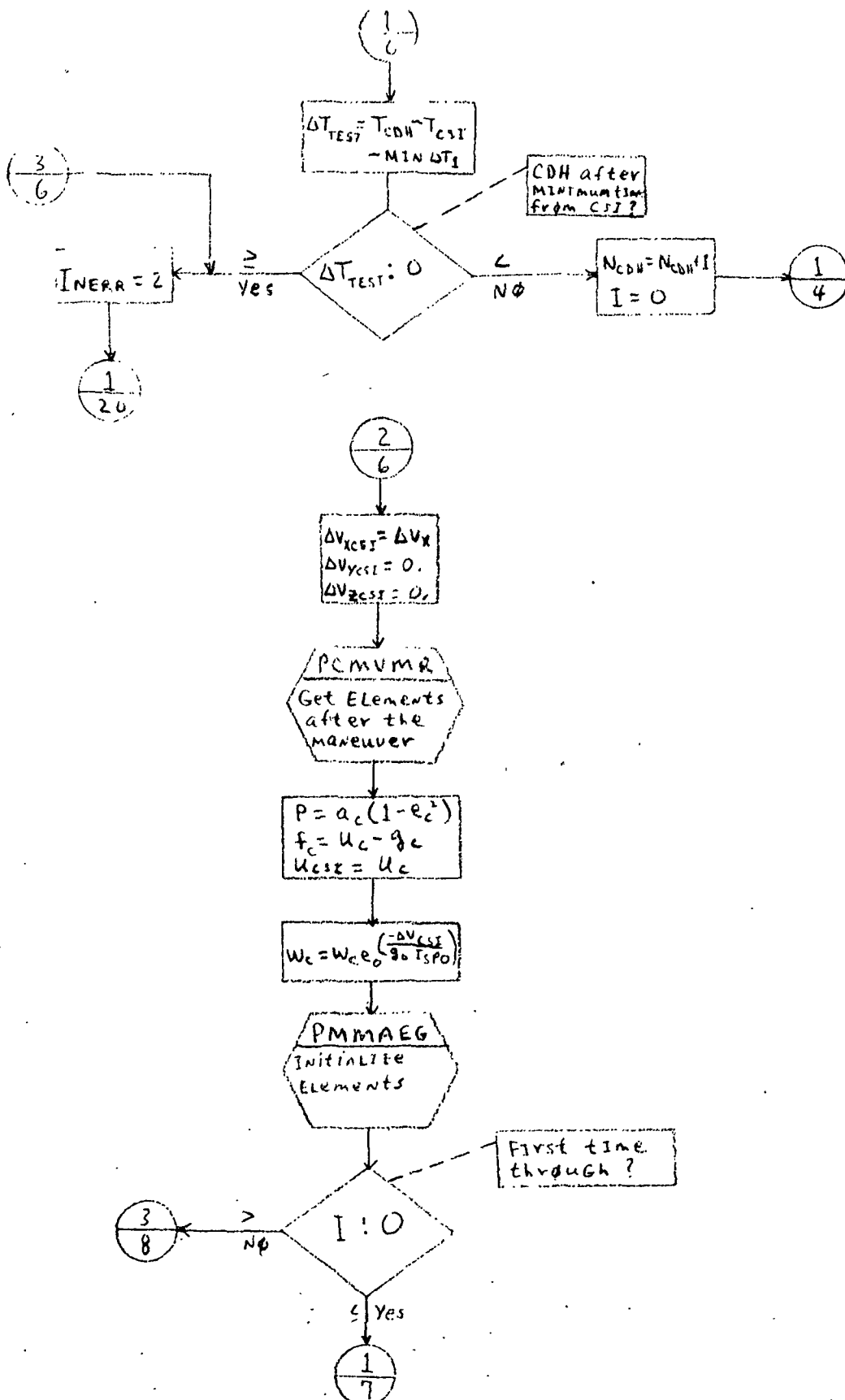


Figure PMMSPO-1. Flowchart

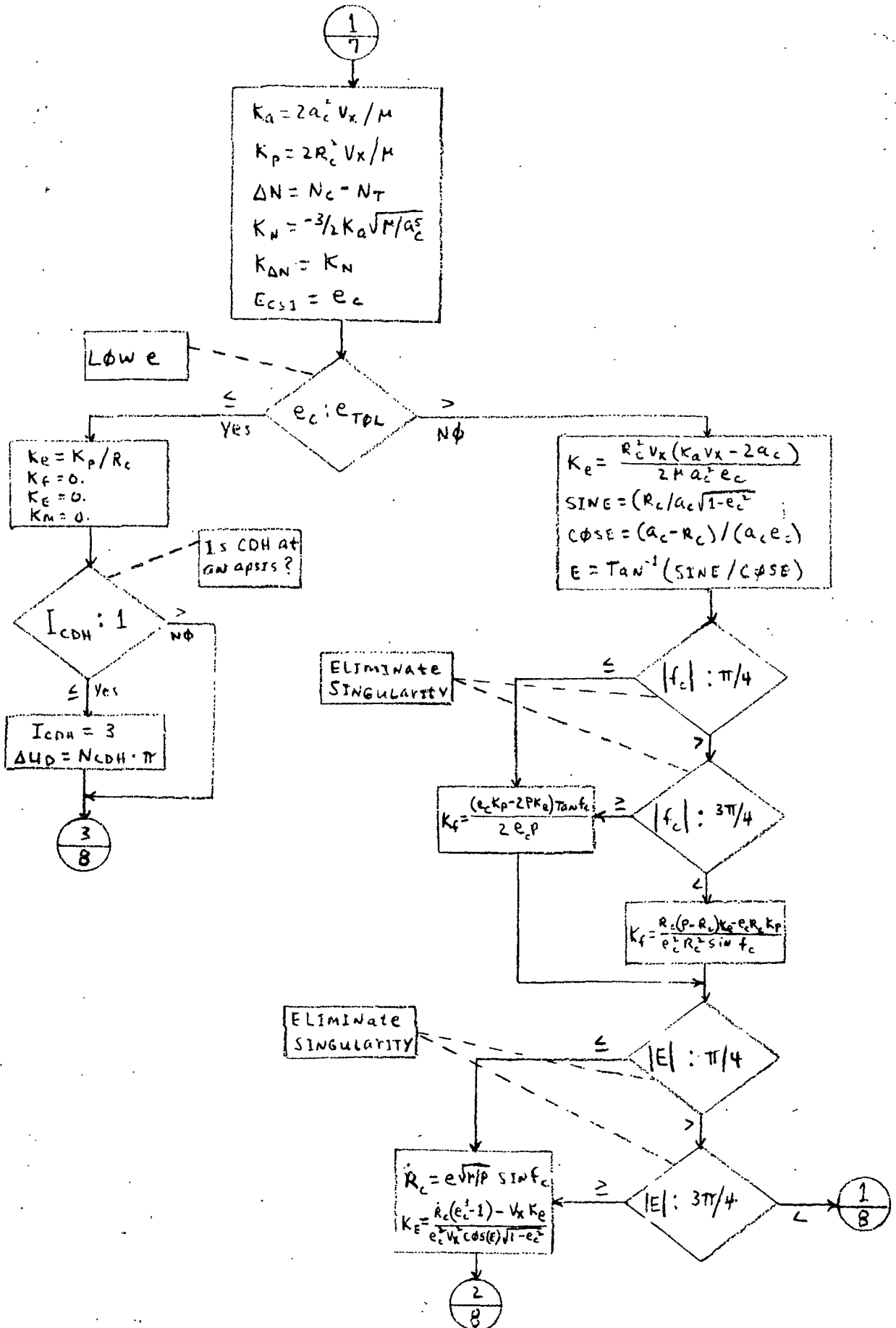


Figure PMMSPO-1. Flowchart (Sheet 7 of 20)

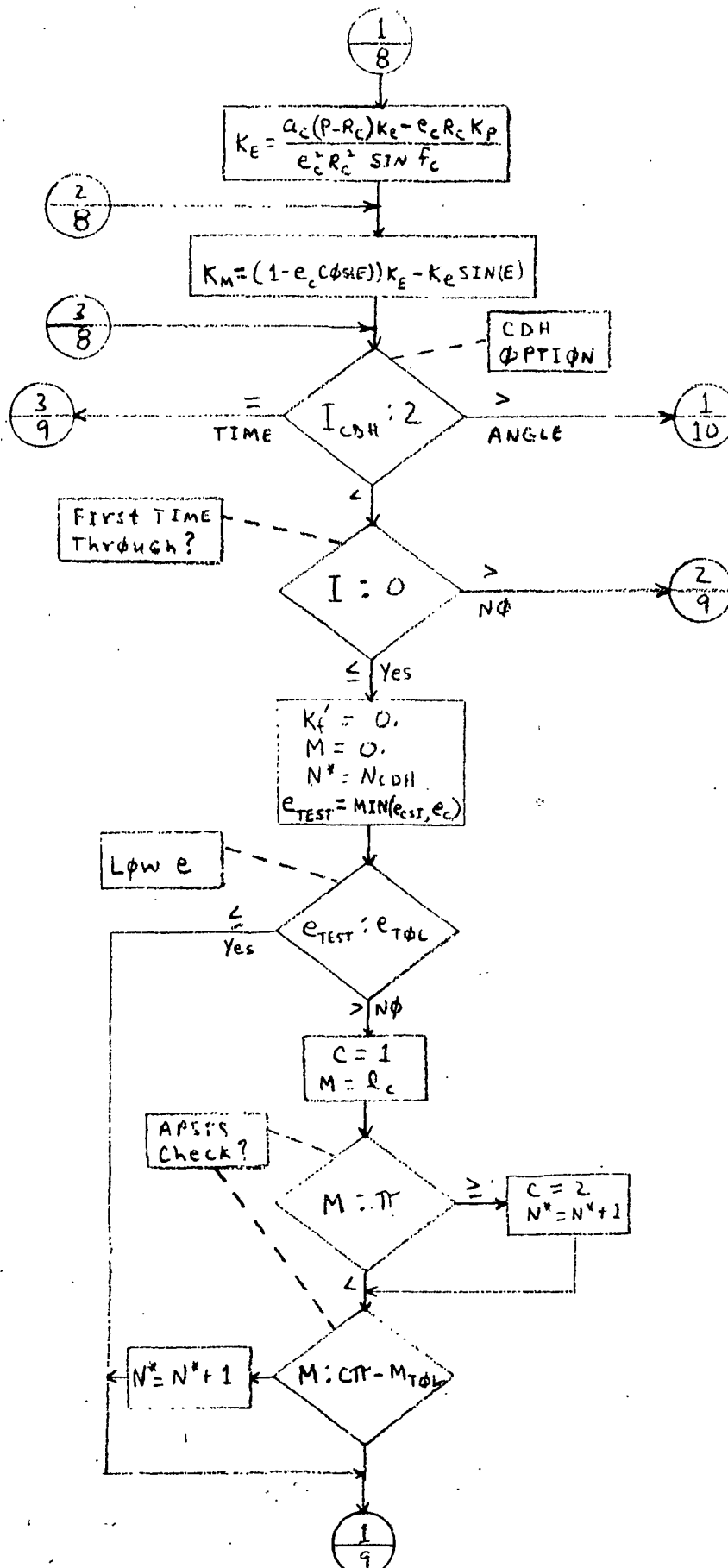


Figure PMMSPO-1. Flowchart (Sheet 8 of 20)

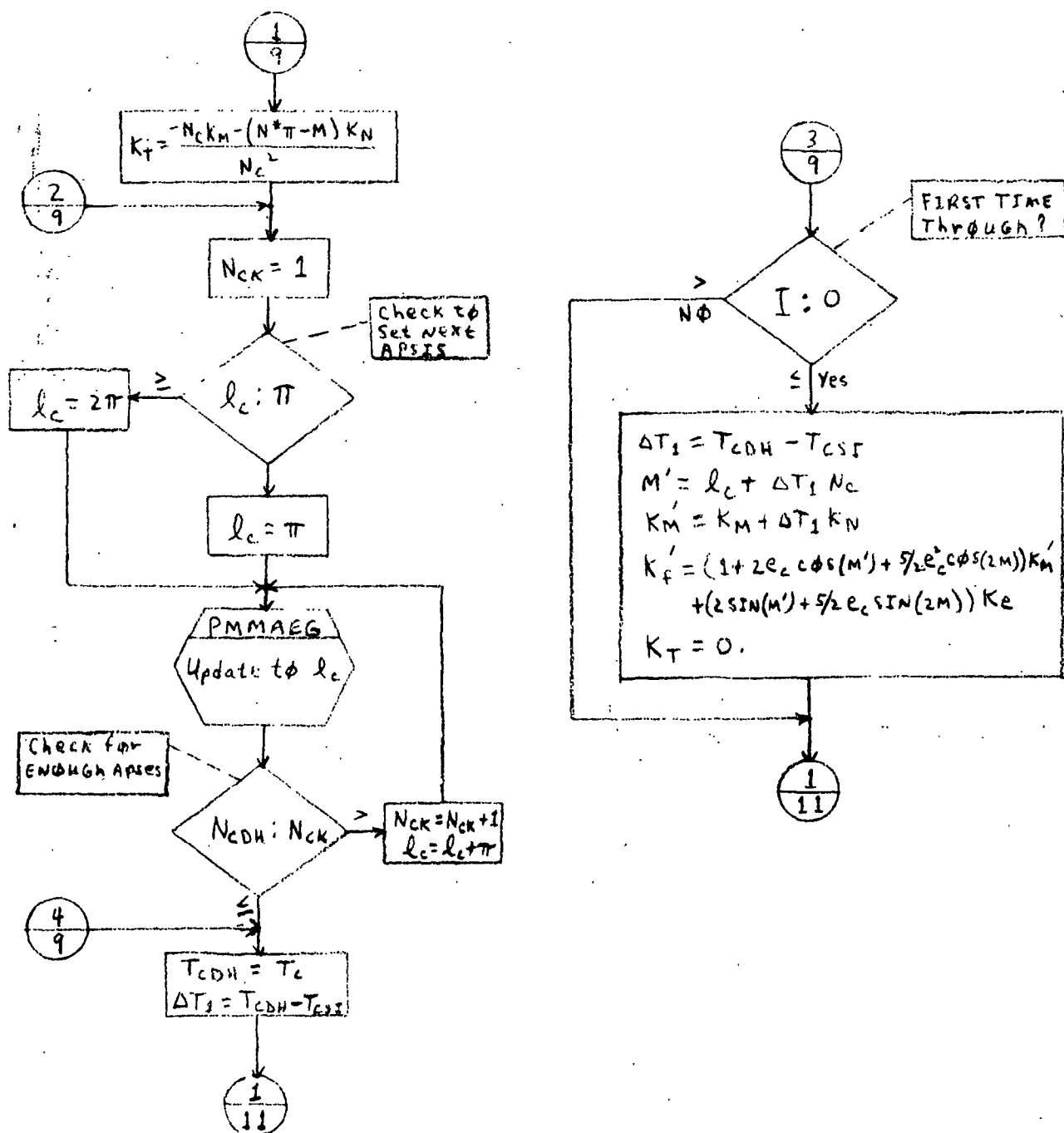


Figure PMMSPO-1. Flowchart

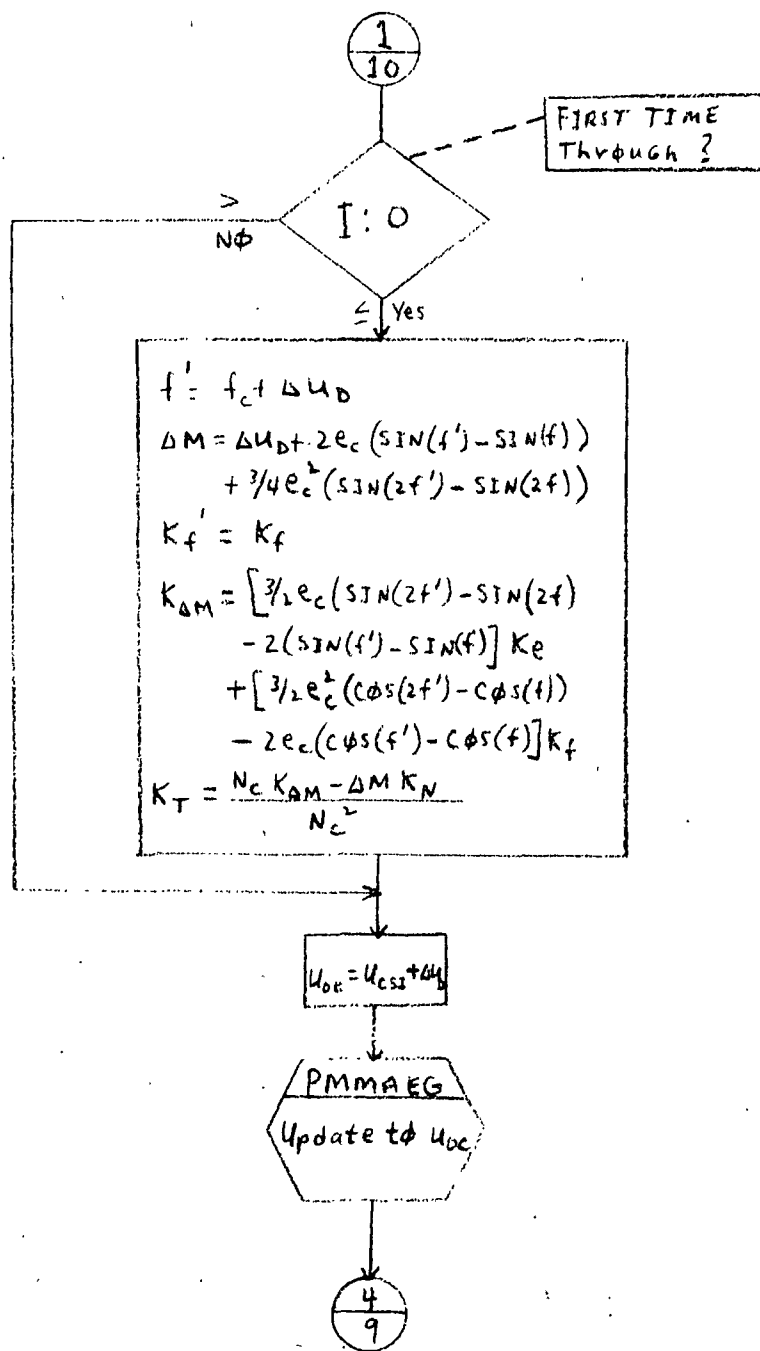


Figure PMMSPO-1. Flowchart

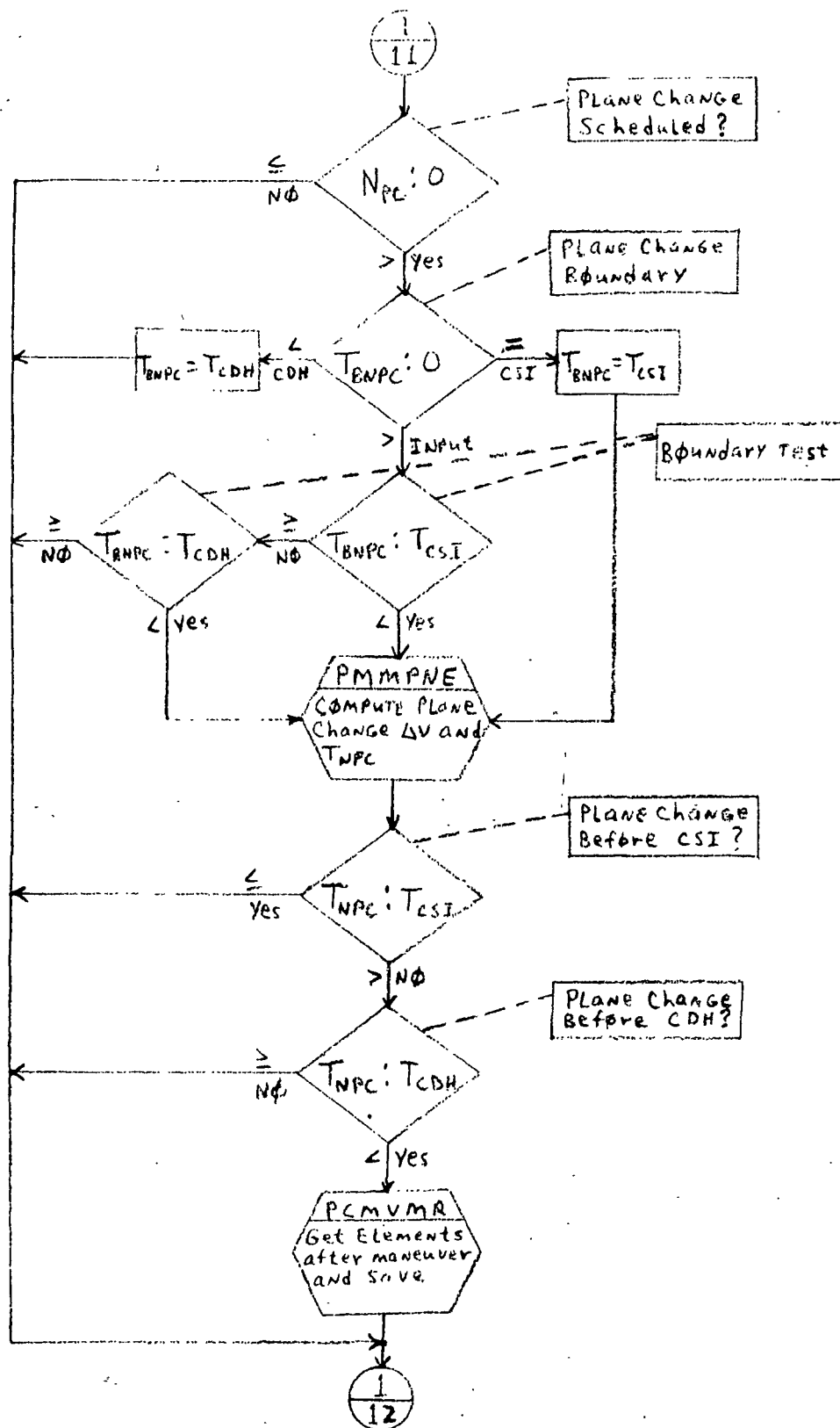


Figure PMMSPO-1. Flowchart

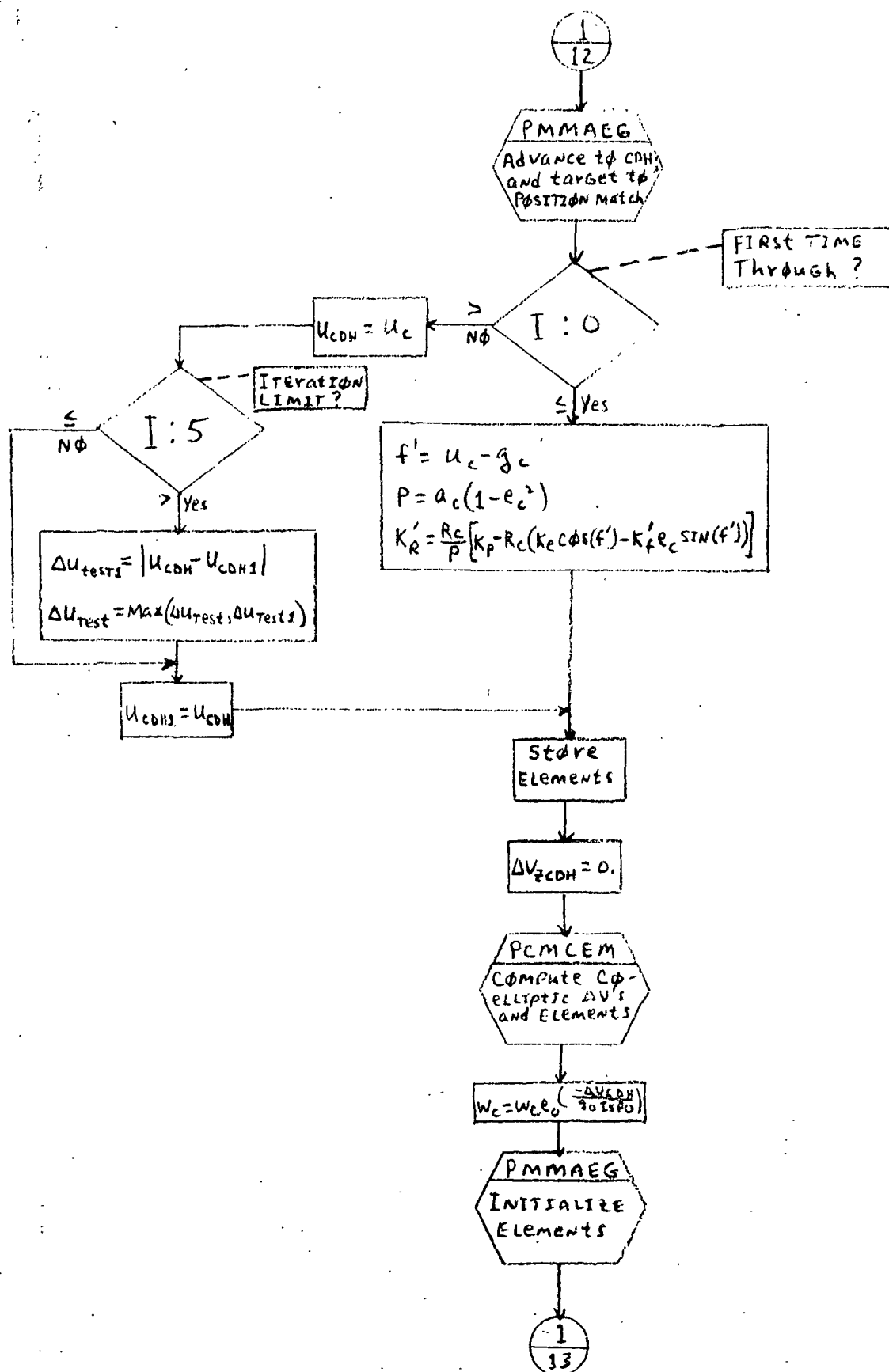


Figure PMMSPO-1. Flowchart

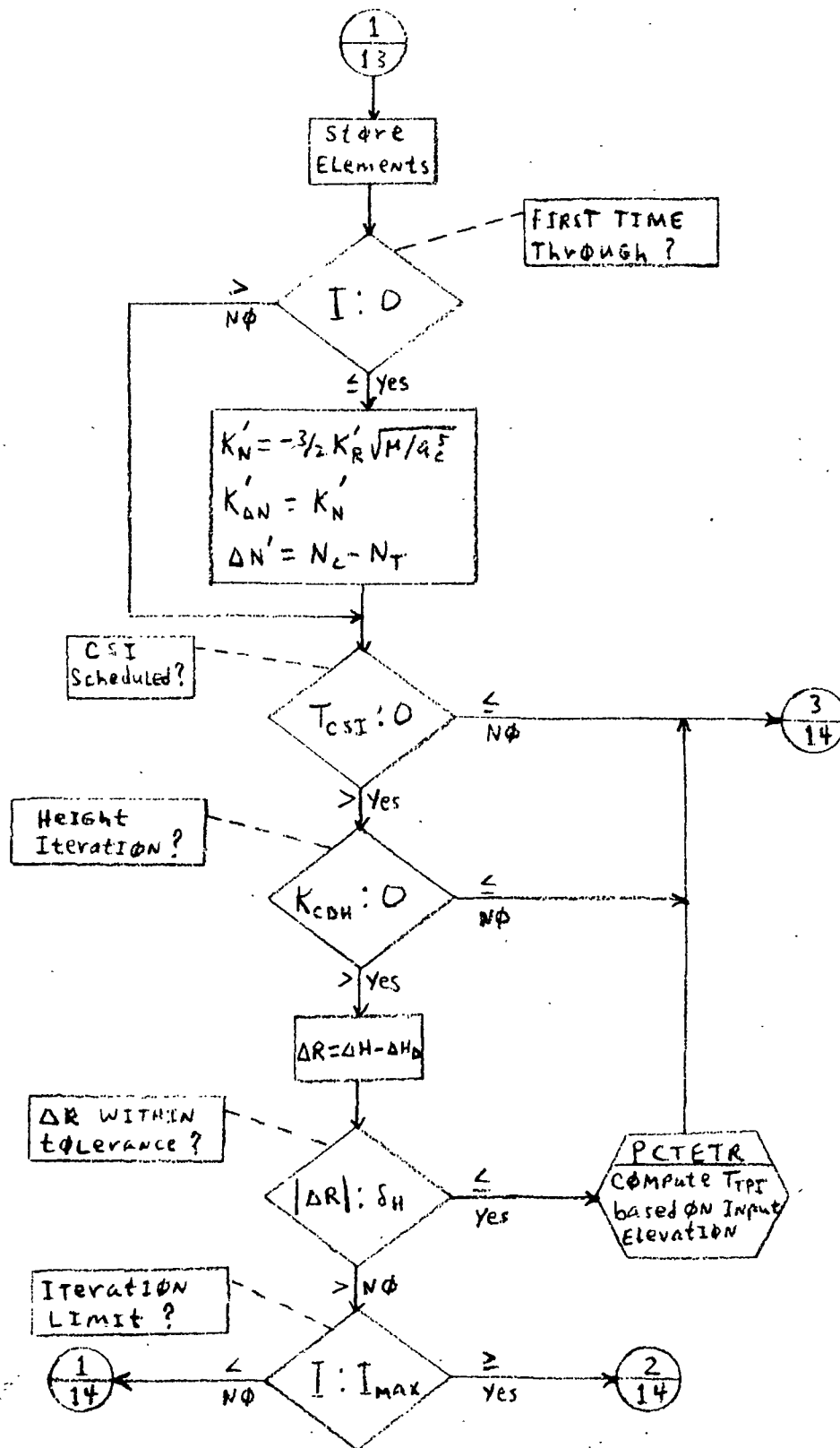


Figure PMMSPO-1. Flowchart

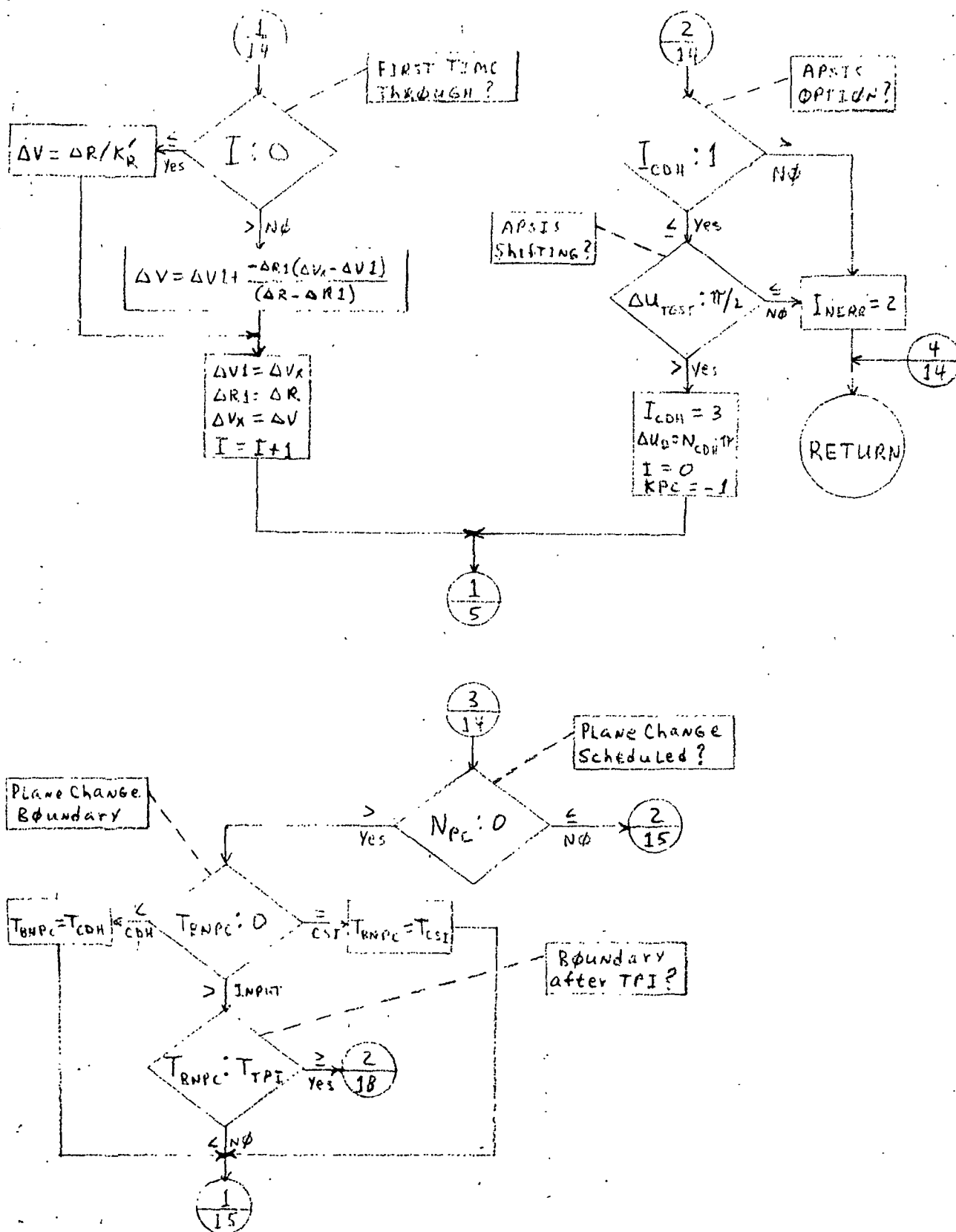


Figure PMMSPQ-1. Flowchart

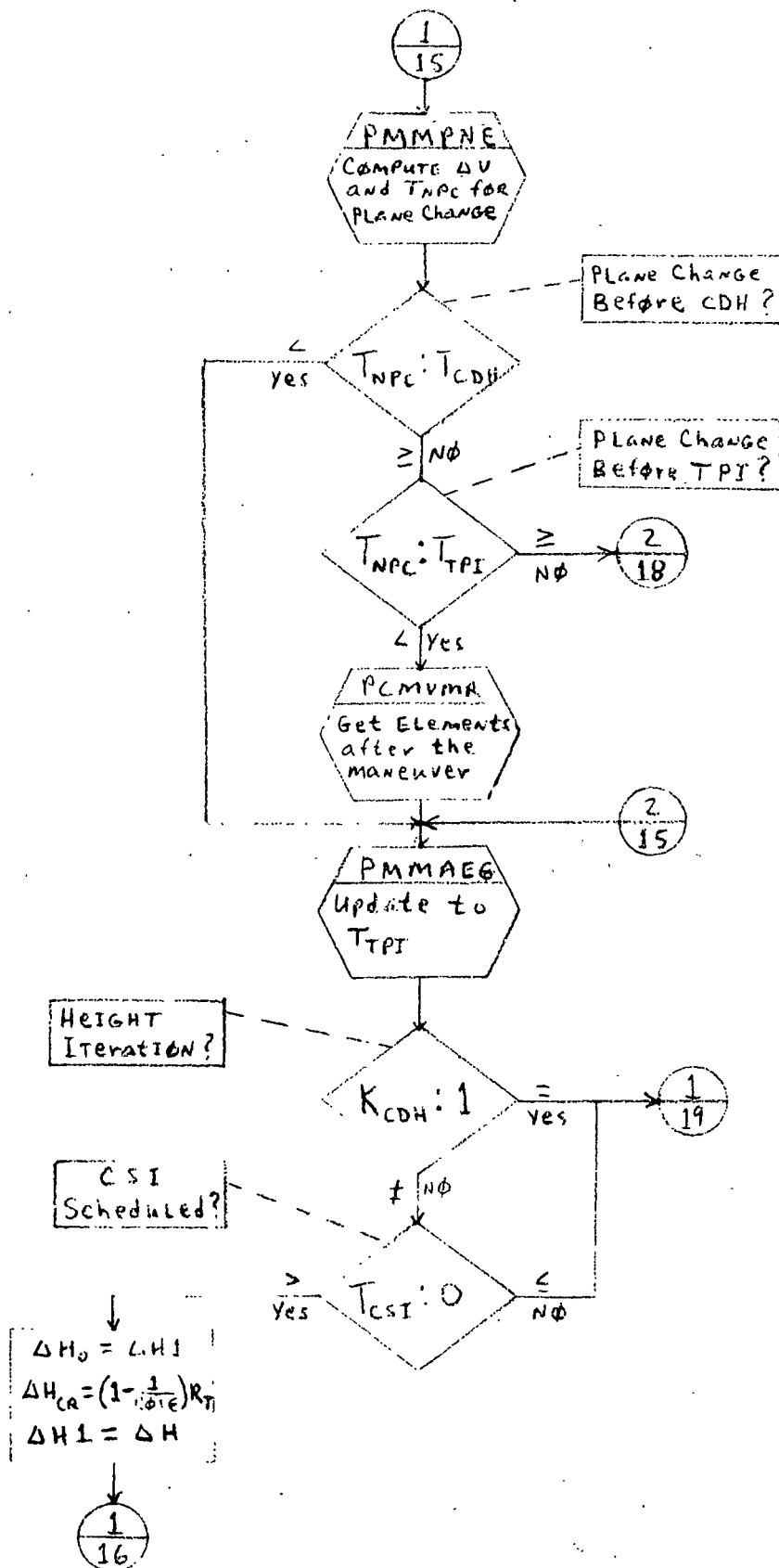


Figure PMMSPO-1. Flowchart

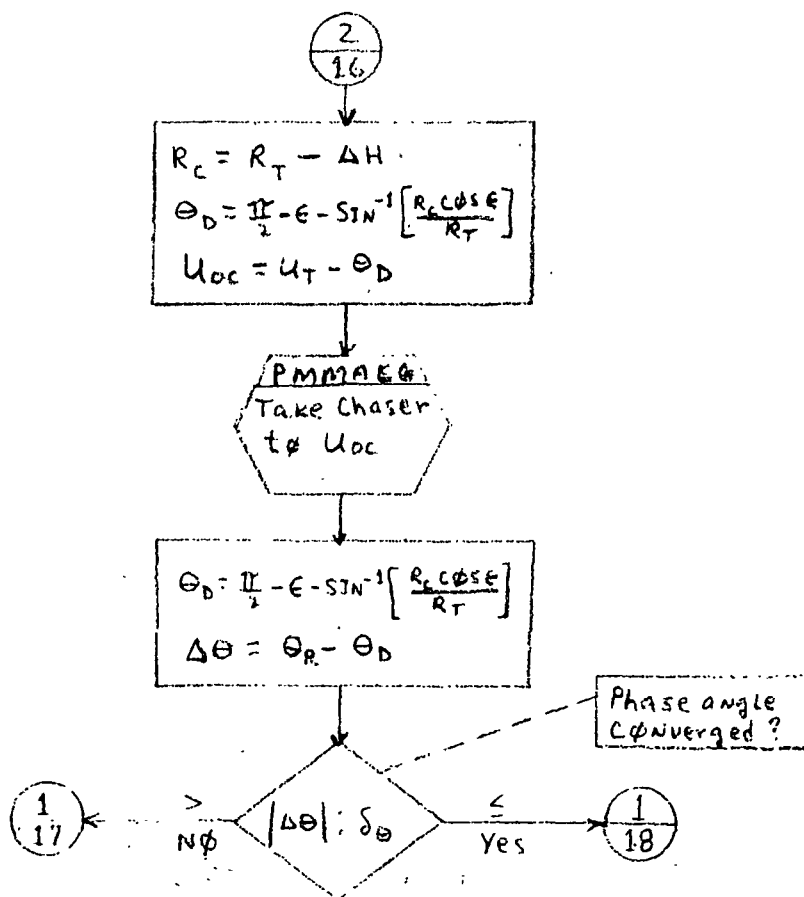
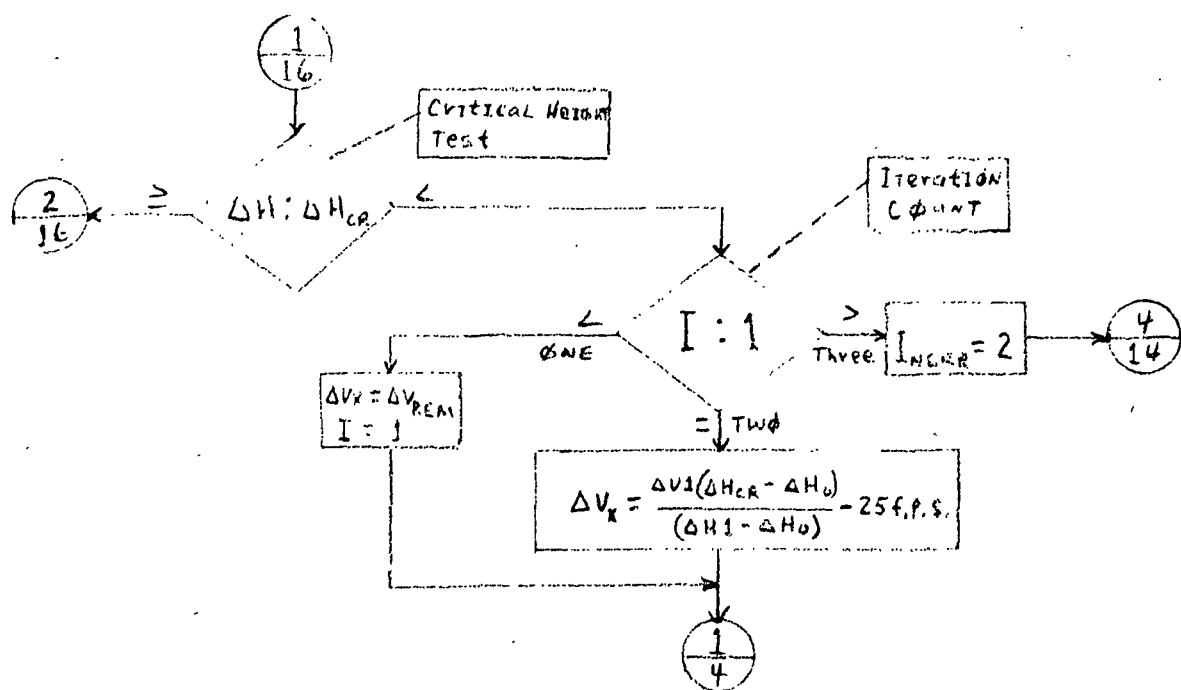


Figure PMMSPO-1. Flowchart

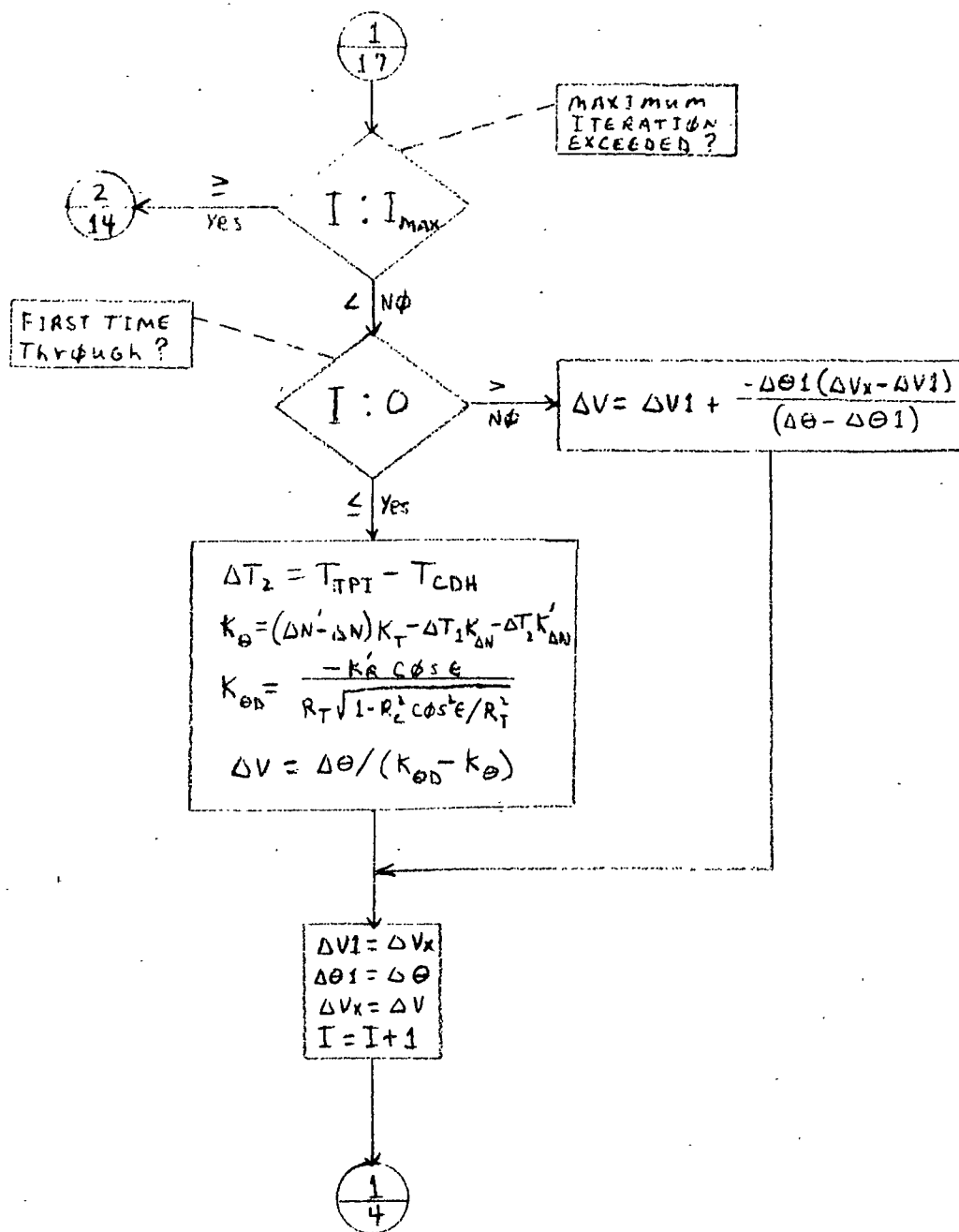


Figure PMMSPQ-1. Flowchart

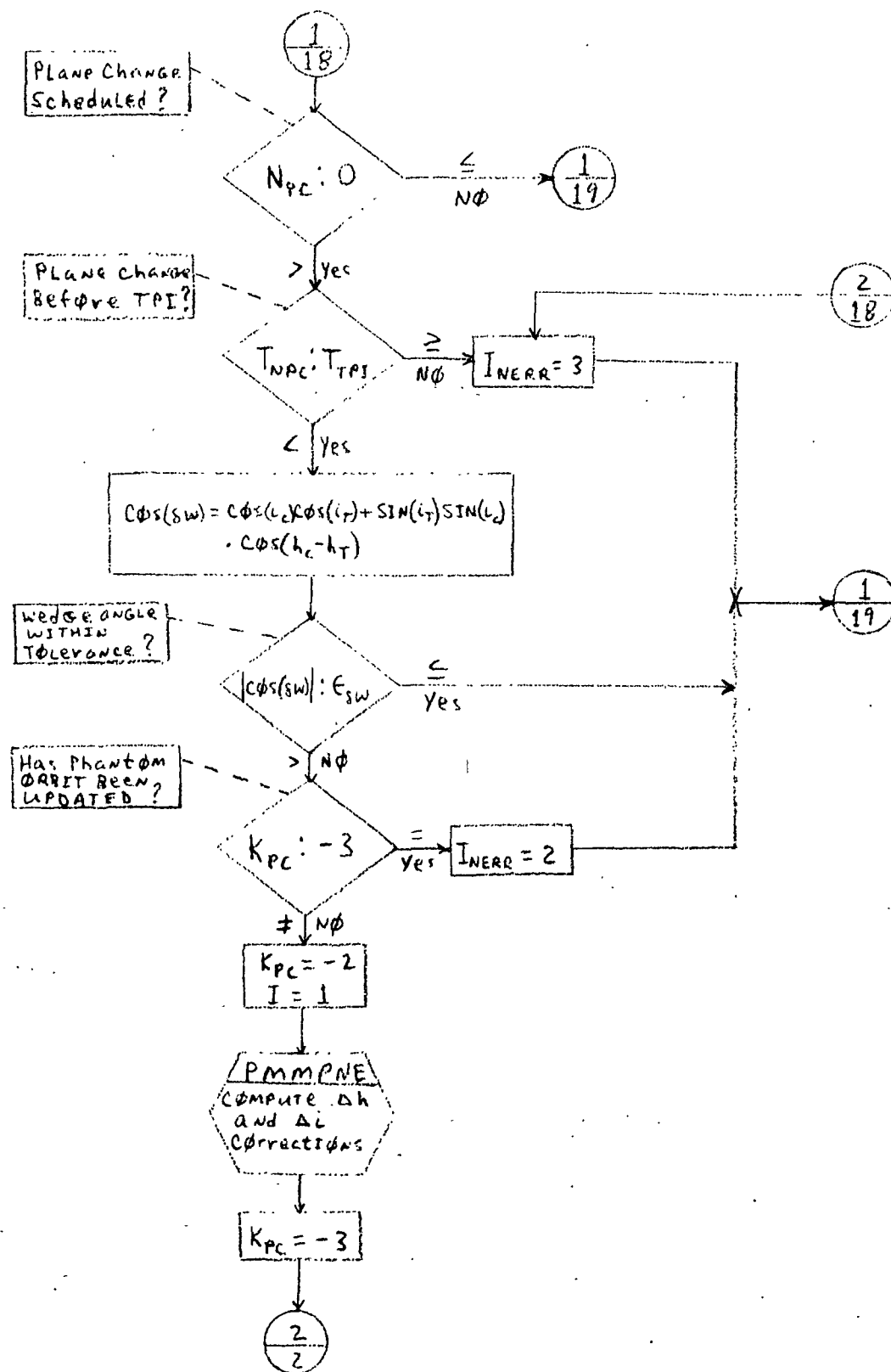


Figure PMMSPQ-1. Flowchart

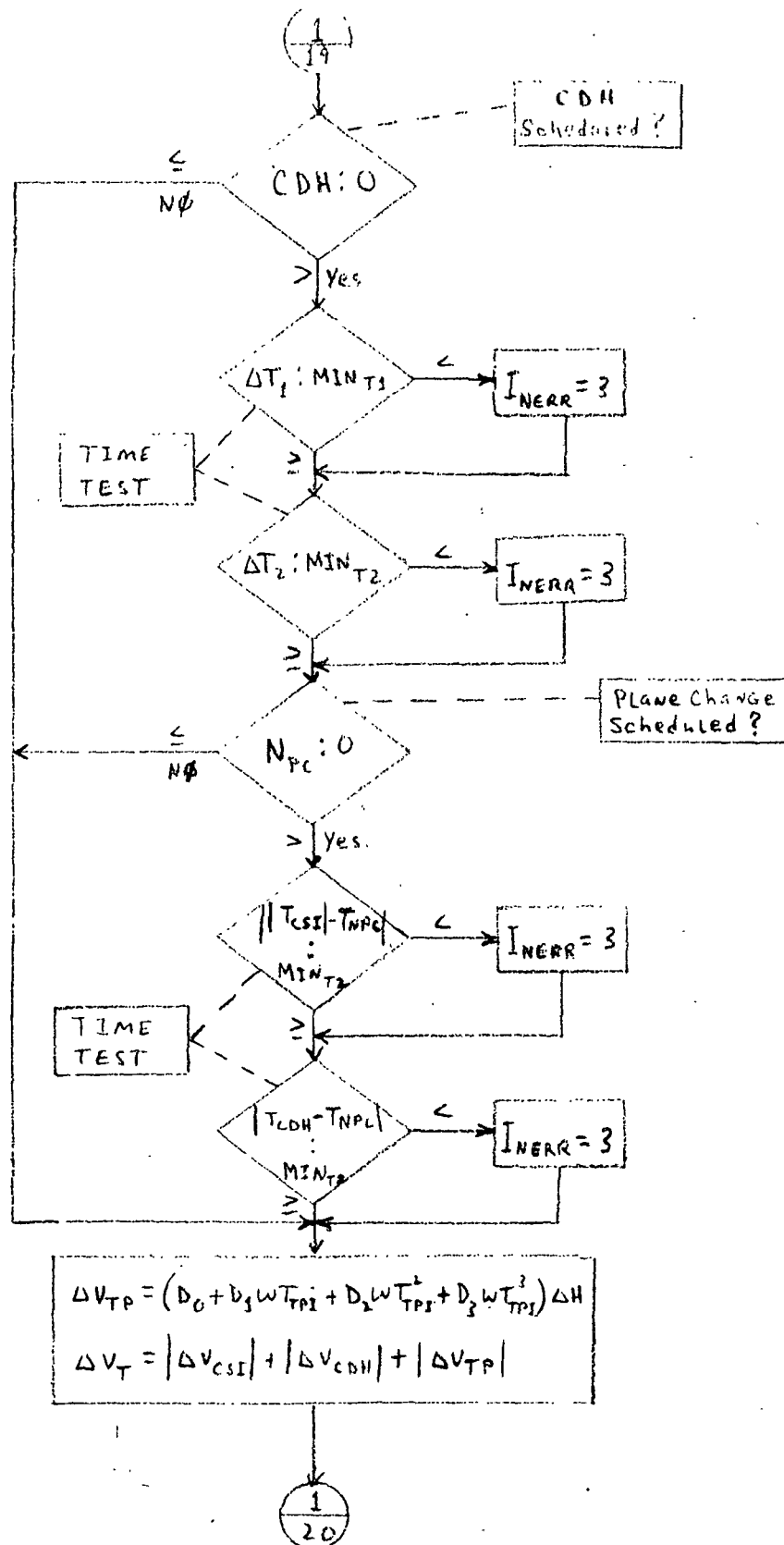
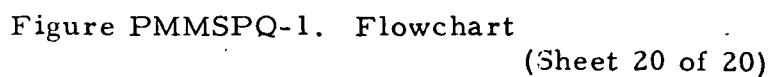


Figure PMMSPO-1. Flowchart



IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMTISS-PCMTISS
 DATE 1/8/68 (EORP)
 REV
 PAGE 1 of 20

- a. Symbol — PMMTISS-PCMTISS
- b. Program Name — Two-impulse Single Solution
- c. Programming Group — Rendezvous Planning
- d. Programmer — T. Hickey
- e. Language — Assembler
- f. Data Tables Used — PZMYSAVE, PZTIPSS
- g. MED's Used — K31
- h. Writeup Completion Date — 11/30/67
- i. System Parameters — Table PMMTISS-PCMTISS-1
- j. Parameter Lists — Tables PMMTISS-PCMTISS-2 through -4
- k. Flowchart — Figure PMMTISS-PCMTISS-1

FUNCTION

The function of the Two-impulse Single Solution load module is to build a data table (PZTIPSS) containing the parameters required for the Two-impulse Single Solution display.

USAGE

- a. Calling Sequence — LINK EP = PMMTISS, PARAM = (S.S. TABLE ADDR., AEG BLOCKS ADDR., WORK AREA ADDR.)
- b. Inputs — See parameter lists, Tables PMMTISS-PCMTISS-2 through -4.
- c. Outputs and Exit — Output is data table PZTIPSS; Exit is by "RETURN" macro to calling load module.
- d. Other Programs Used — CSECT PCVMR, PMMAEG, GIMKIC, GIMAPD
- e. Data Restrictions and Limits — None

METHOD

The additional quantities of the single solution table required for the single solution display are those associated with the line-of-sight (LOS) option, the apogee-perigee heights following each maneuver, the approach data prior to each maneuver, the actual offsets of the chaser from the target following the second maneuver, and the actual target terminal phase transfer angle between the two maneuvers.

The formulation for the LOS parameters and the approach data quantities are found in NASA-MSD Internal Note 67-FM-21. The apogee and perigee heights after each maneuver are computed by the library routine GIMAPD, which is documented separately. The Analytic Ephemeris Generator (AEG) is used to compute the

Approval	Approval
<i>Thomas M. Hickey</i>	<i>A. Aricetti</i>

IBM**RTCC APOLLO PROGRAMMING SYSTEMS**

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMTISS-PCMT
DATE 1/8/68 (EO)
REV
PAGE 2 of 20

actual offsets following the second maneuver. The actual target terminal phase transfer angle is computed as: $\omega_T = N_T (t_2 - t_1)$, where N_T is the mean motion of the target, and t_1 and t_2 are the maneuver times.

*Type
H — Fixed-point half word
F — Fixed-point full word
E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMTISS-PCMT

DATE 1/8/68 (E)

REV.

PAGE 4 of 20

Table PMMTISS-PCMTISS-2. Parameter List

QUEID — NA

QNAME — NA

From — PMSTICN

To-Task — PTMANC

Purpose — Compute single solution
parameters

EP — PMMTISS

Byte No.	Type*	Units	Value / Description / Comments
1-4	F		Address of area in which to build single solution table (see Table PMMTISS-PCMTISS-3.)
5-8	F		Address of AEG blocks and maneuver information (see Table PMMTISS-PCMTISS-4)
9-12	F		Address of work area for PMMTISS 972 bytes

*Type

H — Fixed-point half word

F — Fixed-point full word

E — Single-precision (floating-point)

D — Double-precision (floating-point)

X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMTISS-PCMTISS
 DATE 1/8/68 (EORP)
 REV
 PAGE 5 of 20

Table PMMTISS-PCMTISS-3. Parameter List

QUEID — NA

QNAME — NA

From — PMSTICN

To-Task — NA

Purpose — Area to build single solution
table

EP — NA

Byte No.	Type*	Units	Value / Description / Comments
1-8	X		LM station ID, pre-set by PMSTICN
9-16	X		CSM station ID, pre-set by PMSTICN
17-24	D	hrs.	LM threshold time, pre-set by PMSTICN
25-32	D	hrs.	CSM threshold time, pre-set by PMSTICN
33-34	H		Vehicle code, pre-set by PMSTICN
35-36	H		Mode for pointing chaser vehicle, pre-set by PMSTICN
37-38	H		Plan no., pre-set by PMSTICN
39-40	H		Table indicator, pre-set by PMSTICN
41-48	D	rad.	Actual phase after second maneuver
49-56	D	Er	Actual height offset after second maneuver
57-64	D	rad.	Actual target terminal phase transfer angle between the two maneuvers (no. of firing quads is pre-set in this location)
65-72	D	rad.	Delta pitch angle, pre-set by PMSTICN
73-80	D	hrs.	Time of first maneuver (approach time increment is pre-set in this location)
81-88	D	rad.	E_{HOR} of first maneuver
89-96	D	Er/hr.	ΔV of first maneuver
97-104	D	rad.	Pitch angle of first maneuver
105-112	D	rad.	Yaw angle of first maneuver
112-136	3D	Er/hr.	VX, VY, and VZ components of first maneuver, pre-set by PMSTICN
137-144	D	rad.	LOS pitch - first maneuver
145-152	D	rad.	LOS yaw - first maneuver
153-176	3D	Er/hr.	V80-V82 components - first maneuver
177-200	3D	hrs.	Burn times for each thruster for V80-V82 increments - first maneuver
201-208	D	min.	Minutes until next environment change, pre-set by PMSTICN
209-216	D	Er	Apogee height after first maneuver
217-224	D	Er	Perigee height after first maneuver

 *Type
 H — Fixed-point half word
 F — Fixed-point full word

 E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

2. PMMTISS-PCMTISS

DATE 1/8/68 (EORI)

REV

PAGE 6 of 20

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

Table PMMTISS-PCMTISS-3. Parameter List (Cont'd.)

QUEID —

QNAME —

From —

To-Task —

Purpose —

EP —

Byte No.	Type*	Units	Value / Description / Comments
225-232	D	hrs.	Time of approach data
233-240	D	rad.	Azimuth
241-248	D	rad.	Elevation
249-256	D	Er	Range
257-264	D	Er/hr.	Range rate
265-272	D	Er	X component of curvilinear system
273-280	D	Er	Y component of curvilinear system
281-288	D	Er	Z component of curvilinear system
289-416			Two duplicates of bytes 225-228
417-824			Duplicate of bytes 73-416 plus an additional duplicate of bytes 225-288 --- all quantities are for the second maneuver

*Type

H — Fixed-point half word

F — Fixed-point full word

E — Single-precision (floating-point)

D — Double-precision (floating-point)

C — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMTIS-PCMTISS
DATE 1/8/68 (EORP)
REV
PAGE 7 of 20

Table PMMTISS-PCMTISS-4. Parameter List

QUEID — NA

QNAME — NA

From — PMSTICN

To-Task — NA

Purpose — Area to interface with PMMAEG
and miscellaneous quantities

EP — NA

Byte No.	Type*	Units	Value / Description / Comments
1-440			Standard header and two blocks for PMMAEG interface
441-448	D	hrs.	Time of second maneuver
449-584			Unused
585-592	D	hrs.	Time of first maneuver
593-608	D		Unused
609-616	D	lbs.	Weight prior to first maneuver
617-1000			Unused
1001-1008	D	Er/hr.	Delta V of first maneuver
1009-1016	D	Er/hr.	Delta V of second maneuver

*Type

H — Fixed-point half word

F — Fixed-point full word

E — Single-precision (floating-point)

D — Double-precision (floating-point)

X — Hexadecimal

TWO - IMPULSE
SINGLE
SOLUTION

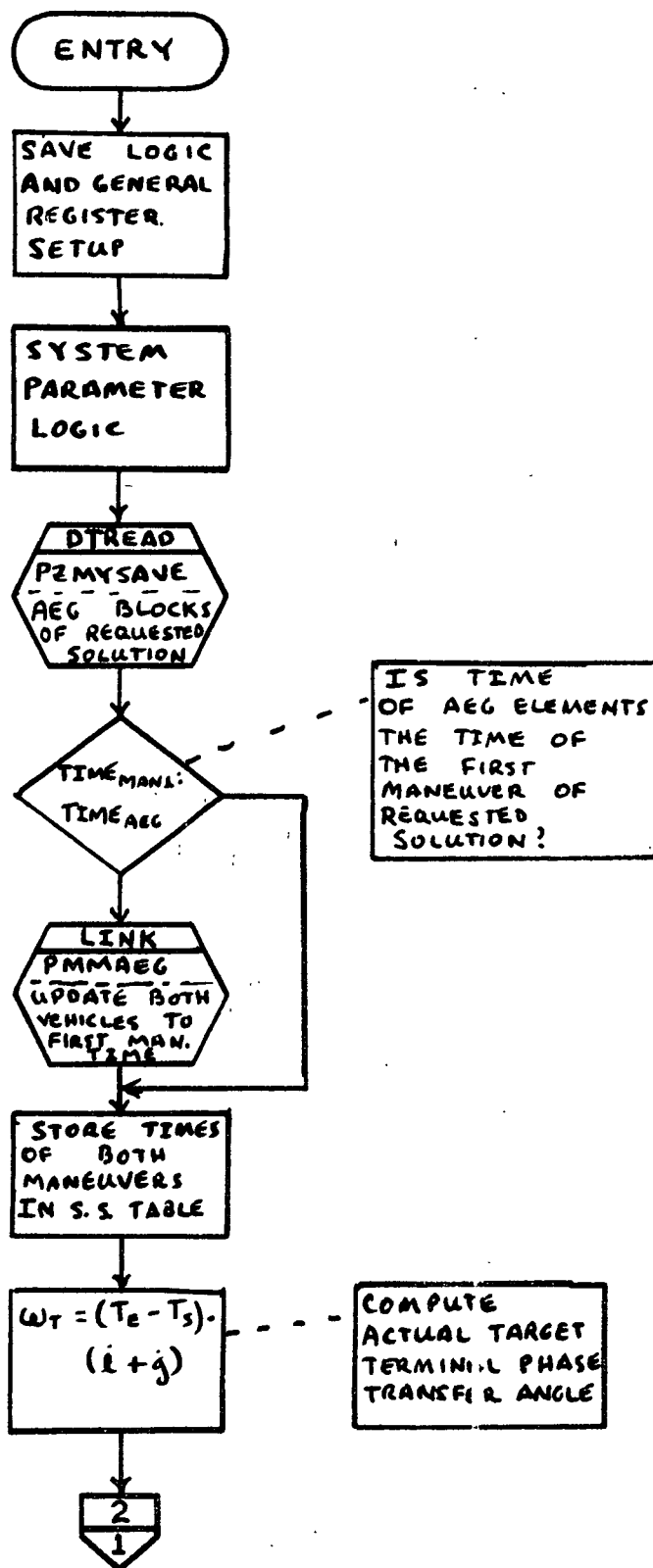


Figure PMMTISS-PCMTISS-1. Flowchart

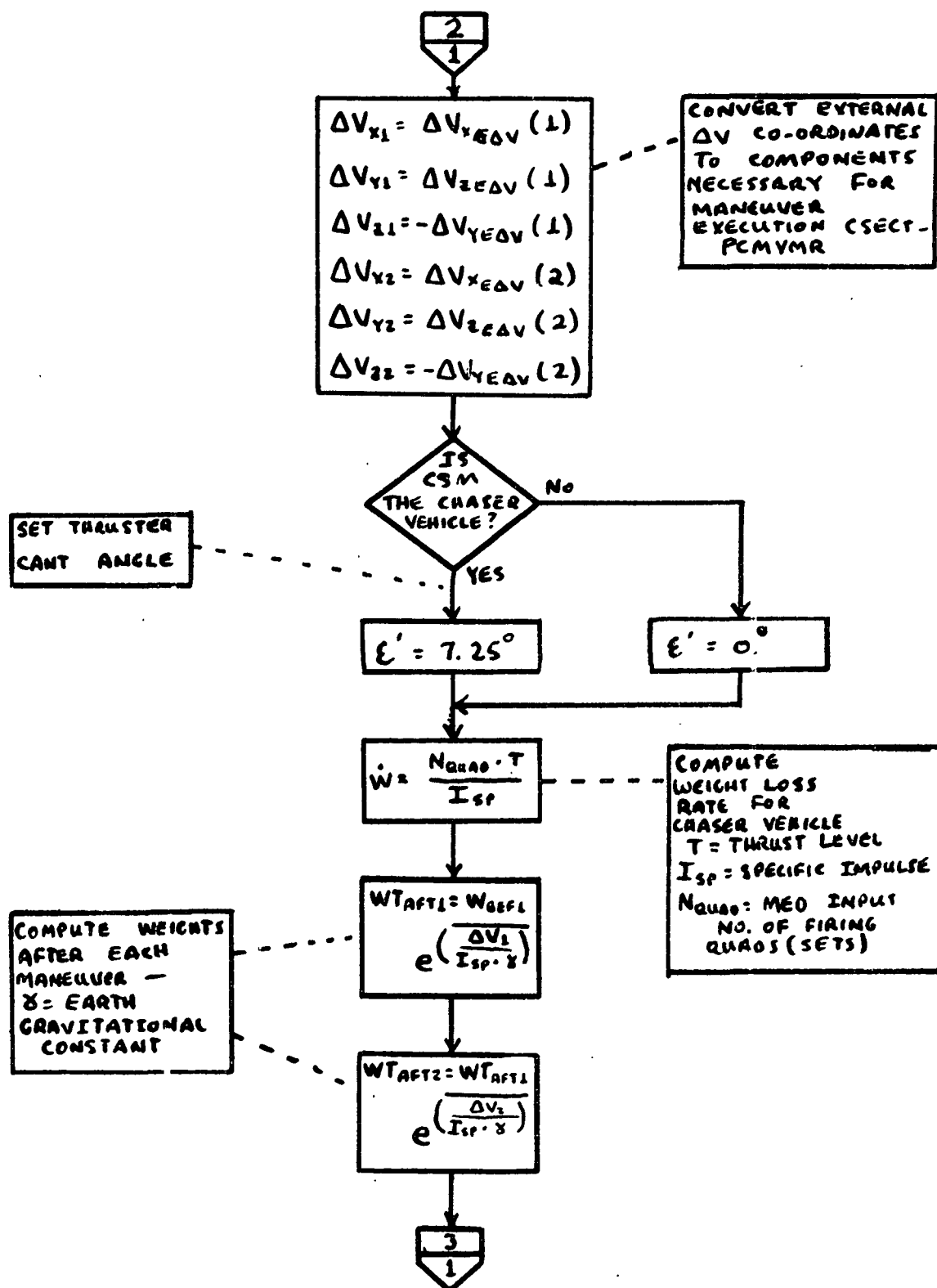


Figure PMMTISS-PCMTISS-1. Flowchart

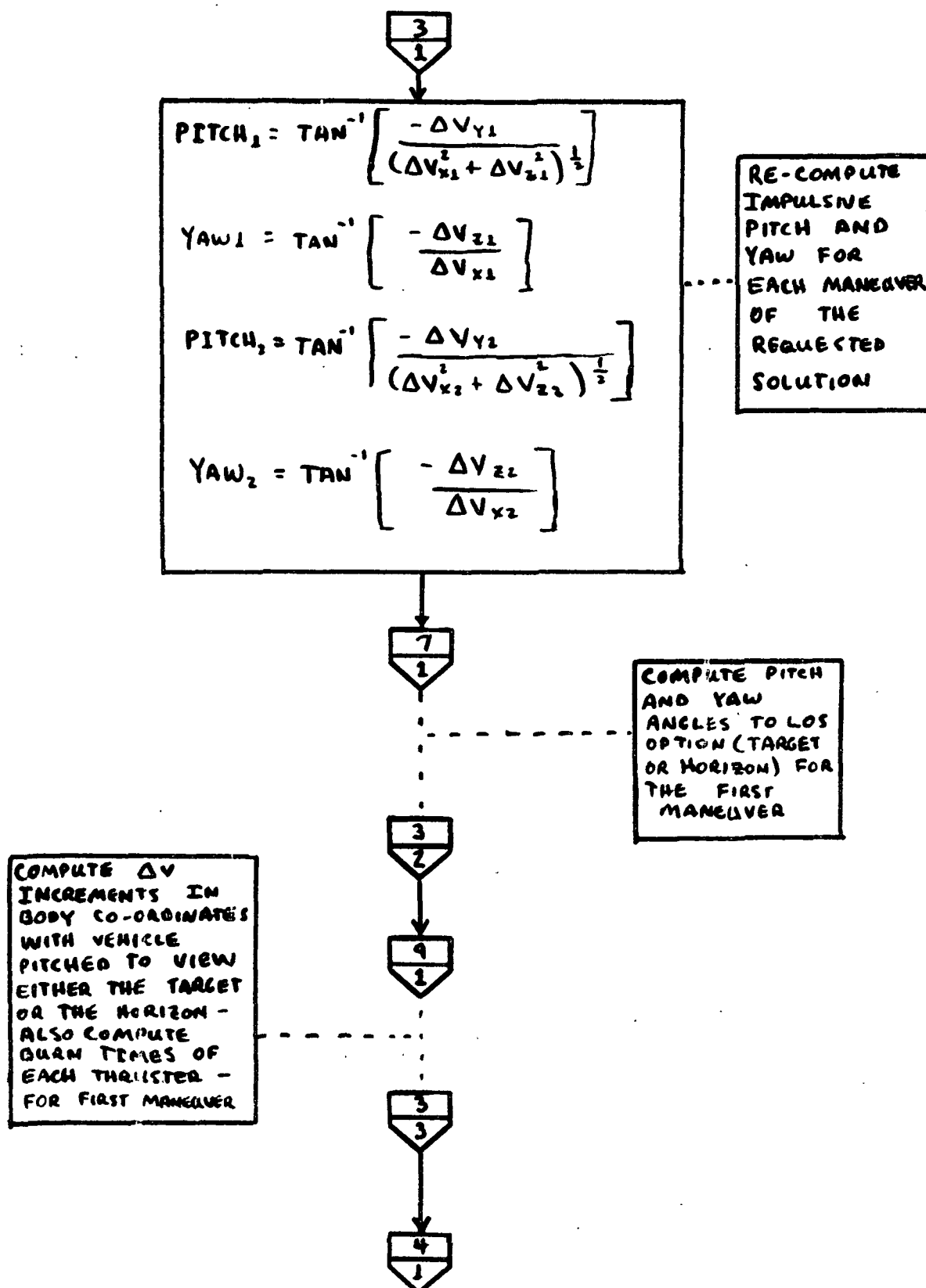


Figure PMMTISS-PCMTISS-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMTISS-PCMTISS

DATE 1/8/68 (EORP)

REV

PAGE 11 of 20

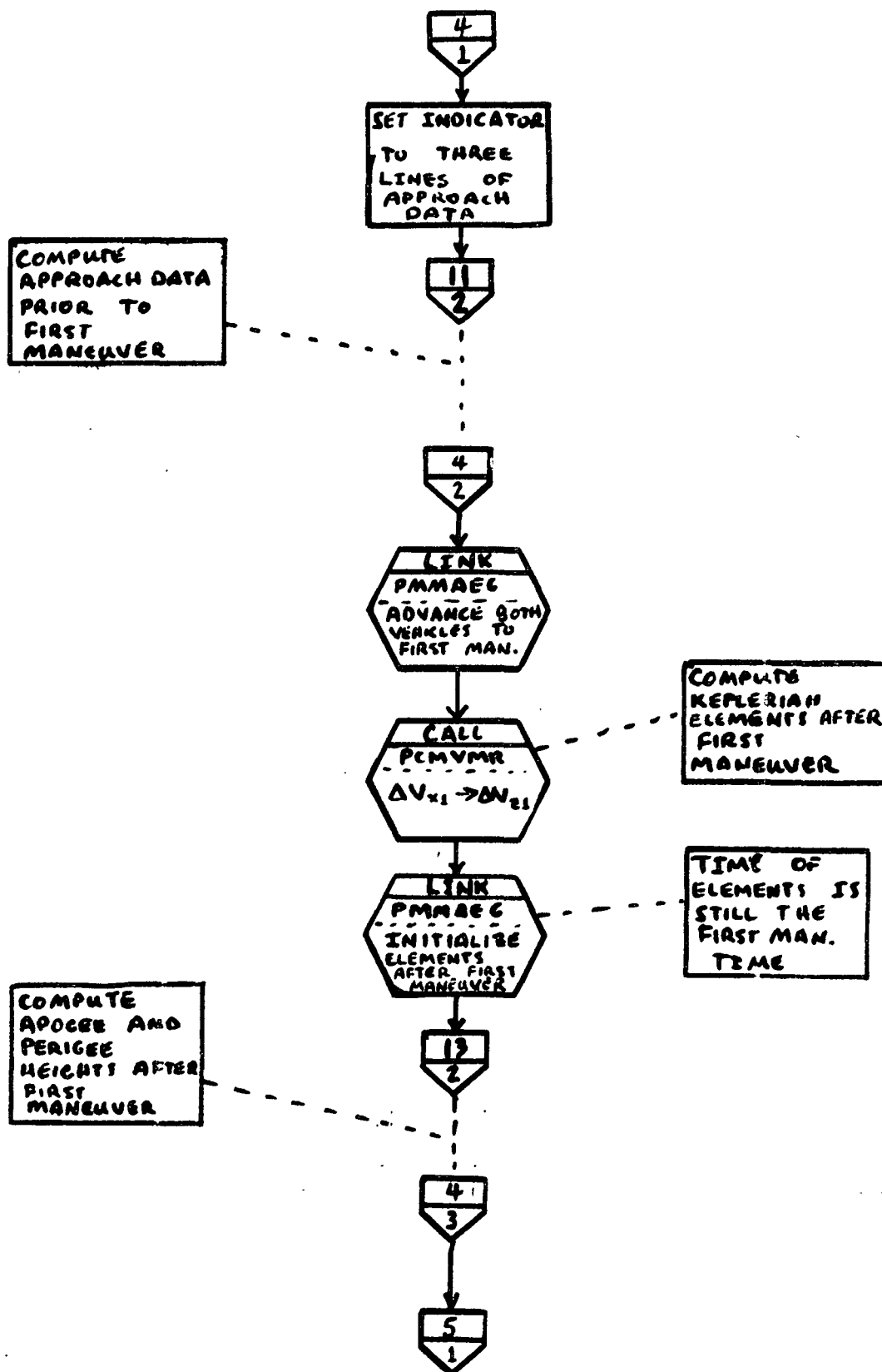


Figure PM MTISS-PCMTISS-1. Flowchart

(Sheet 4 of 13)

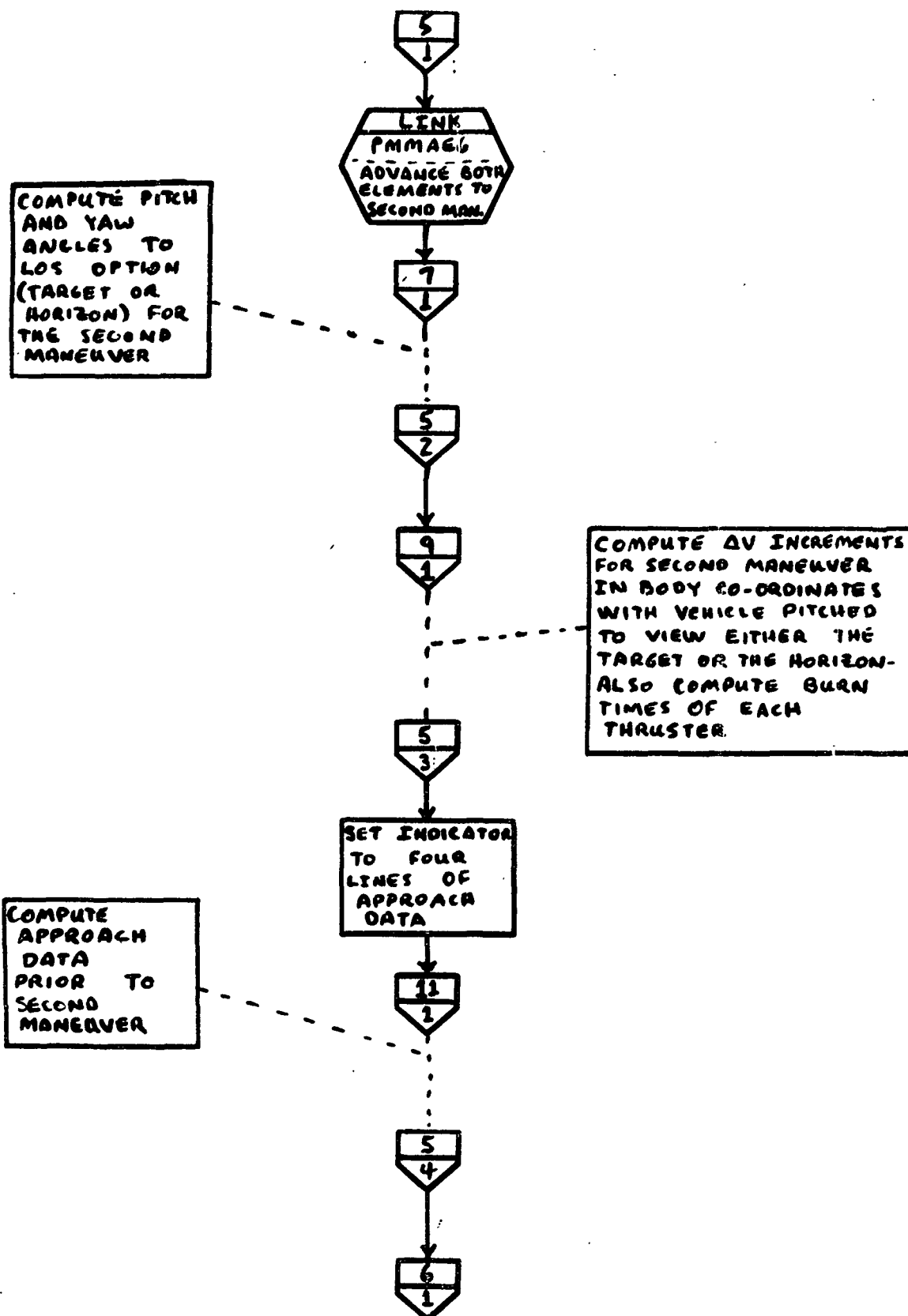


Figure PMMTISS-PCMTISS-1. Flowchart

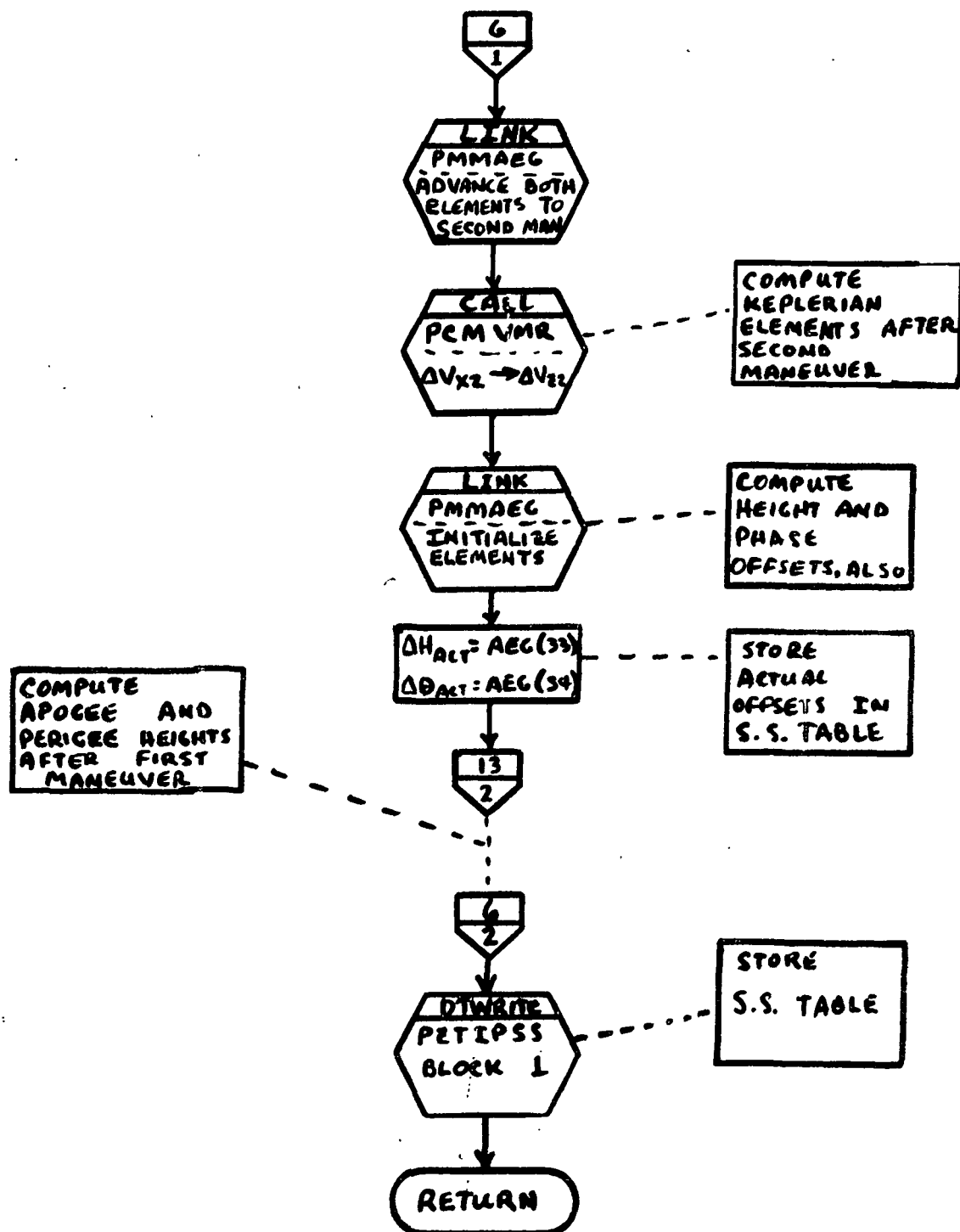


Figure PMMTISS-PCMTISS-1. Flowchart

LOOK ANGLE COMPUTATION INTERNAL SUBROUTINE

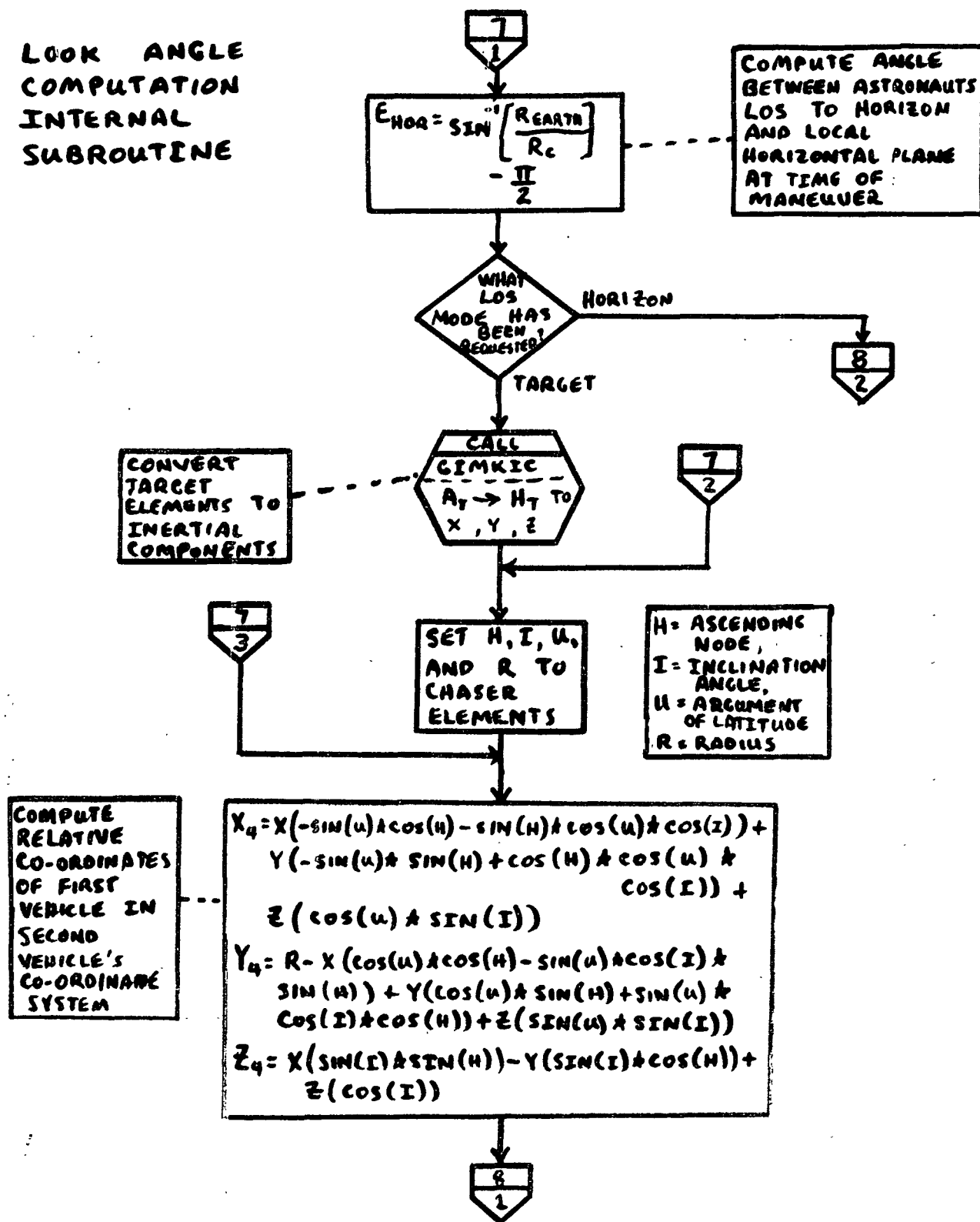


Figure PMMTISS-PCMTISS-1. Flowchart

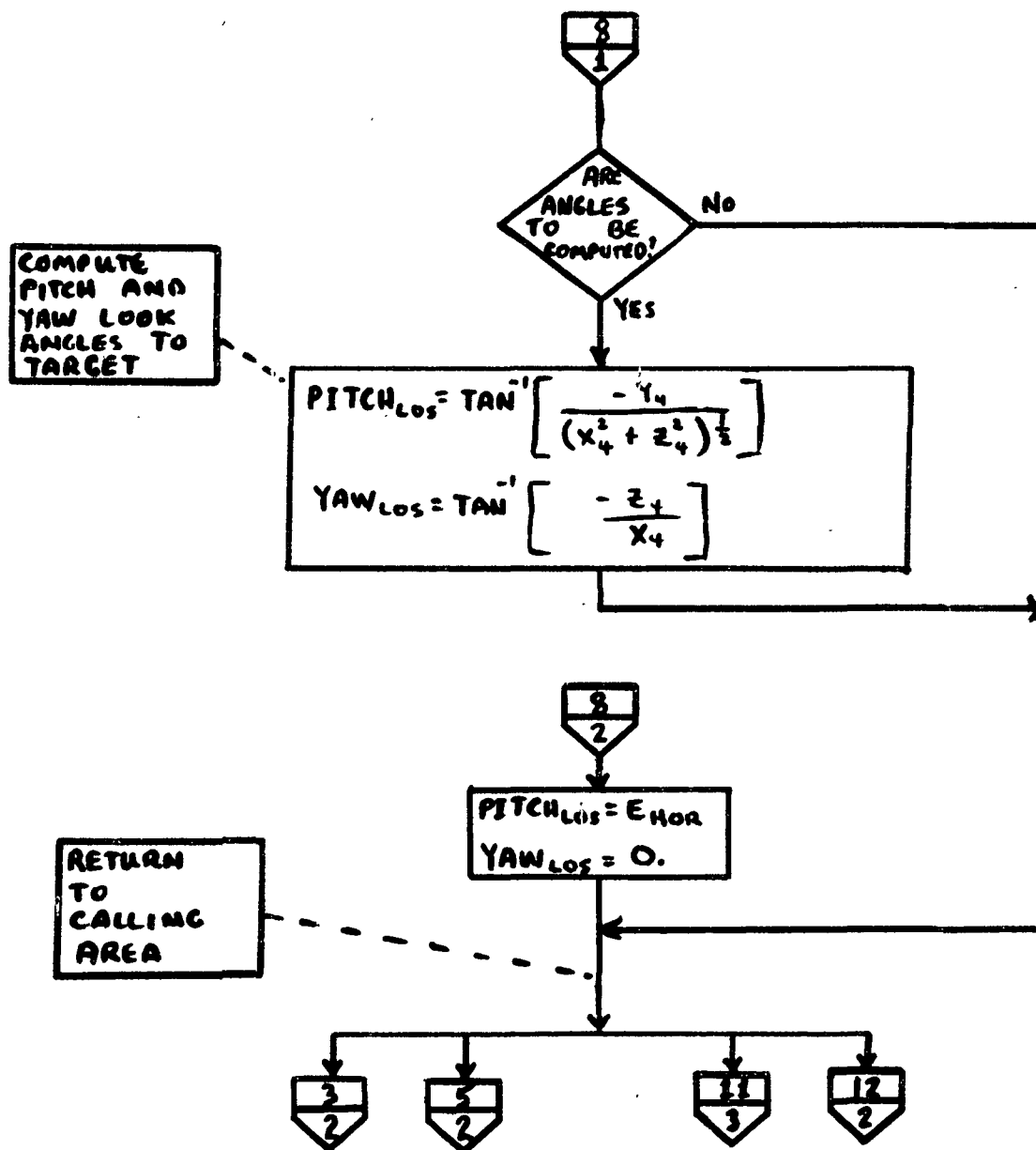


Figure PMMTISS-PCMTISS-1. Flowchart

ΔV INCREMENT INTERNAL SUBROUTINE

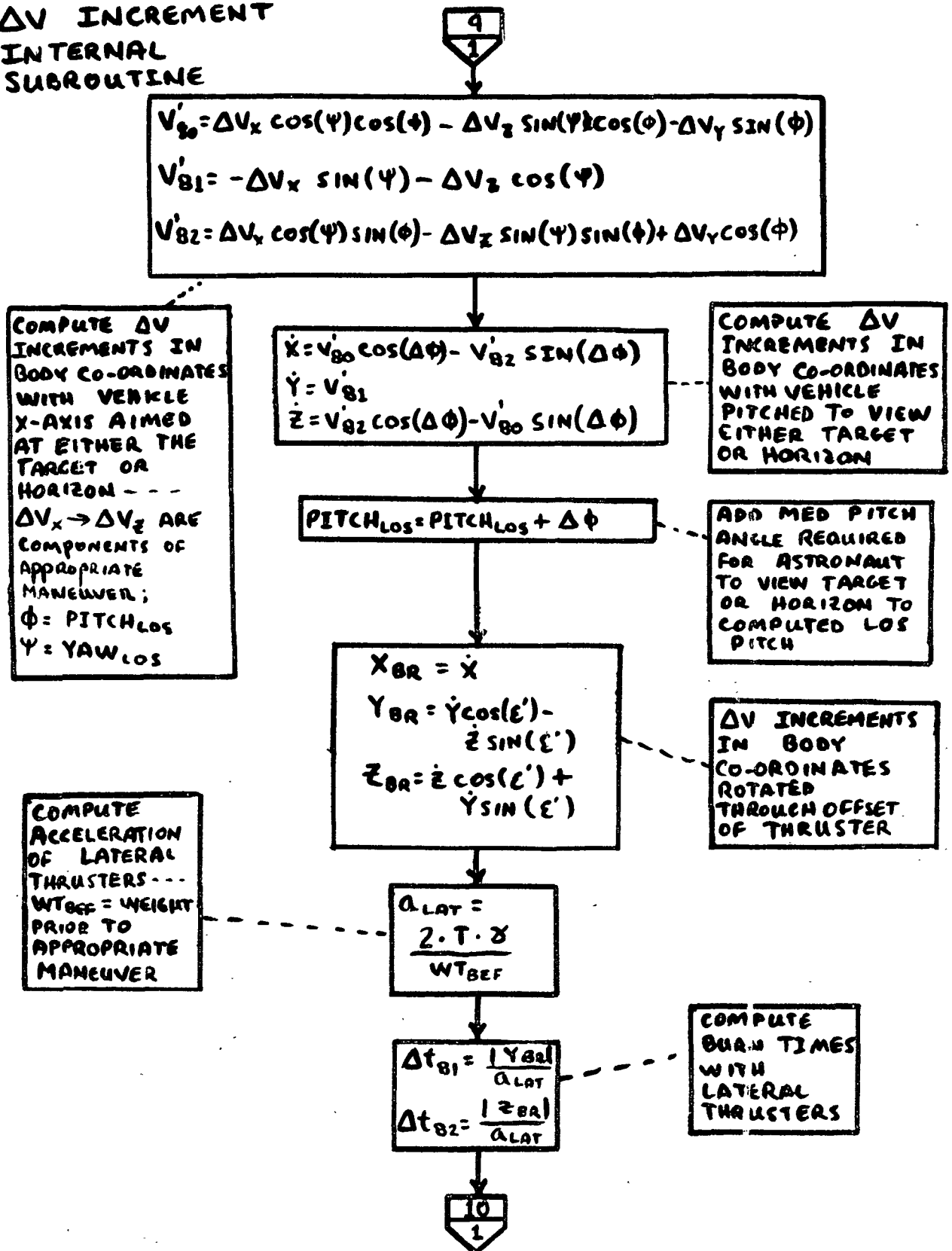


Figure PMMTISS-PCMTISS-1. Flowchart

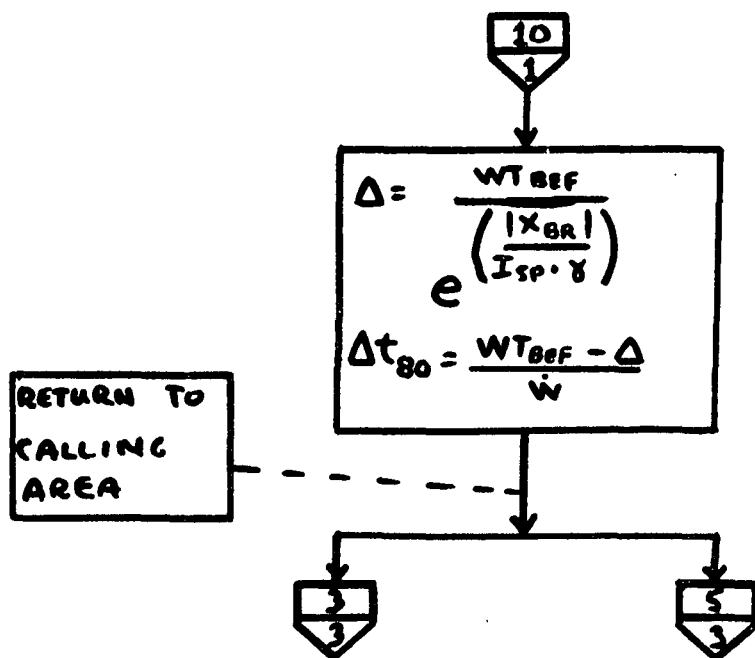


Figure PMMTISS-PCMTISS-1. Flowchart

APPROACH DATA INTERNAL SUBROUTINE

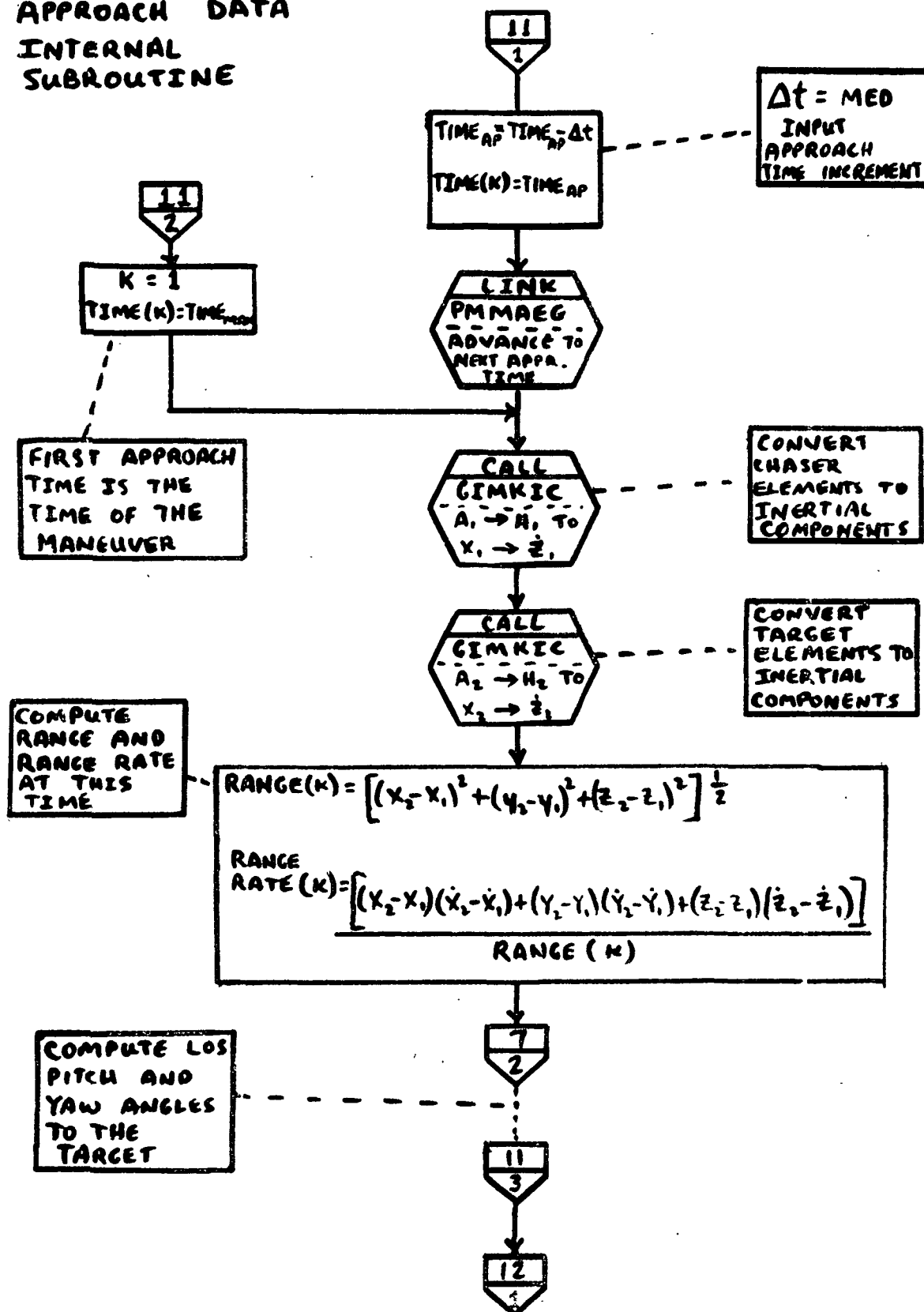


Figure PMMTISS-PCMTISS-1. Flowchart

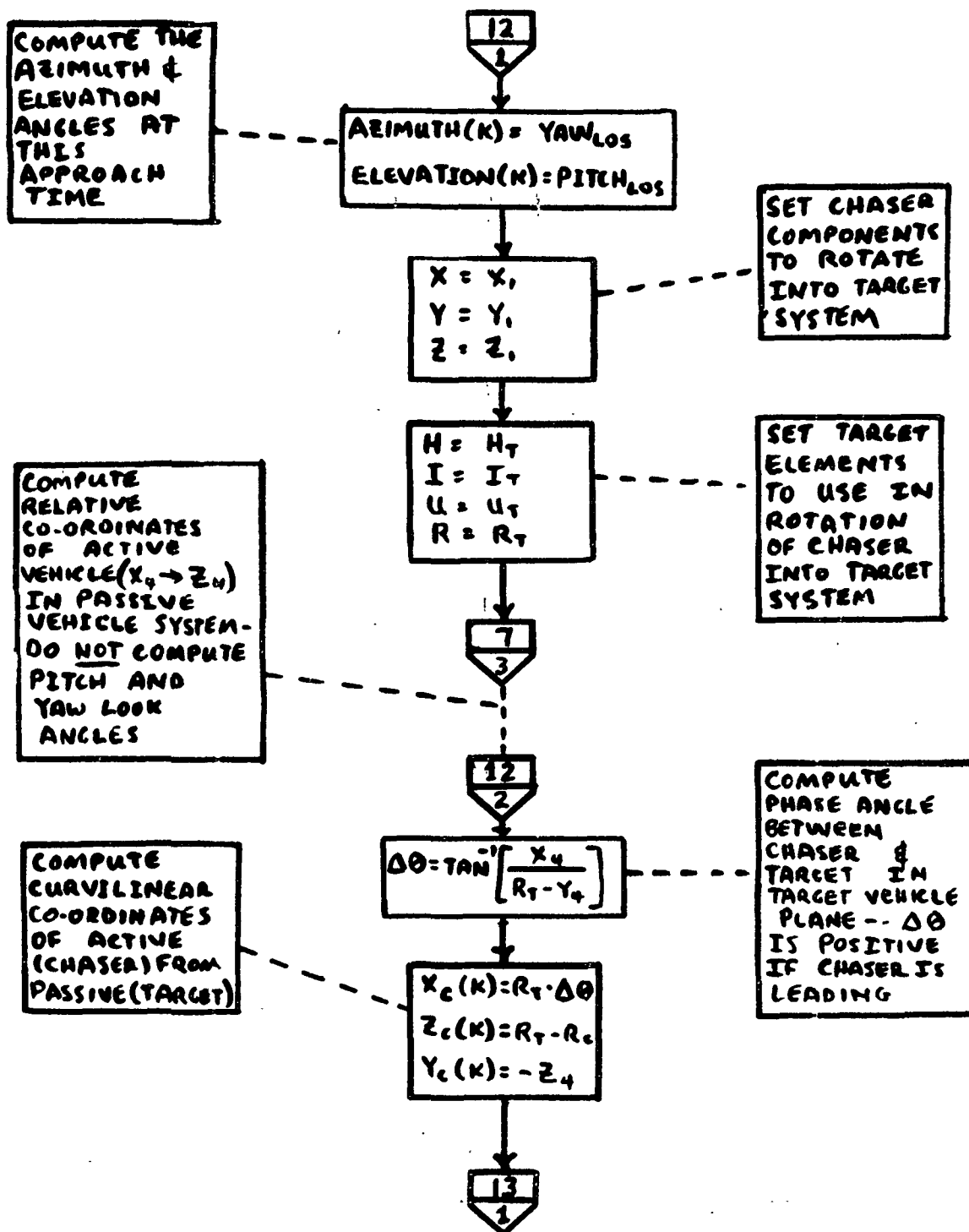


Figure PMMTISS-PCMTISS-1. Flowchart

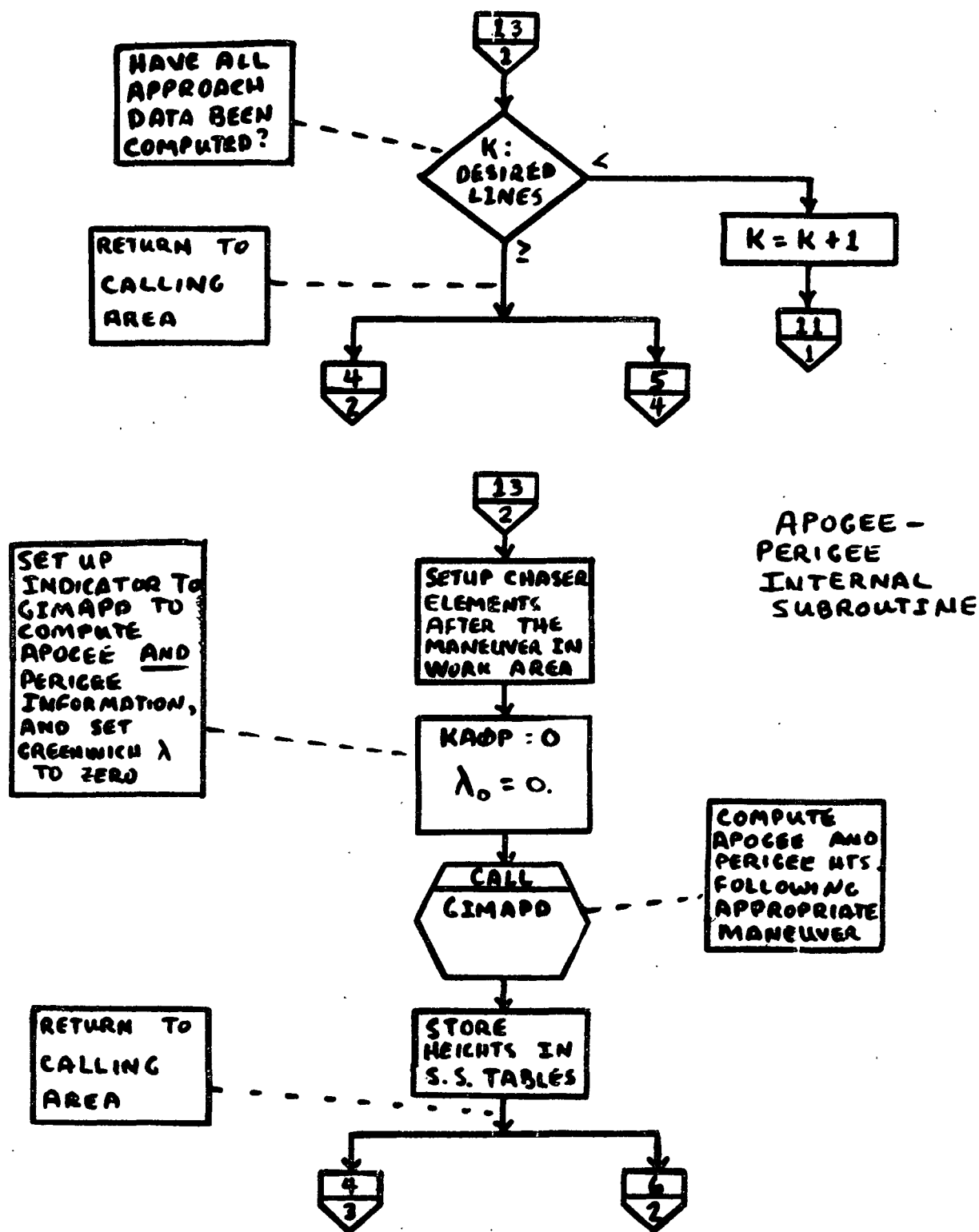


Figure PMMTISS-PCMTISS-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK:MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMTISS-PCMVMR
 DATE 1/8/68 (EORP)
 REV
 PAGE 1 of 6

- a. Symbol — PMMTISS-PCMVMR
- b. Program Name — Velocity Maneuver Performer
- c. Programming Group — Rendezvous Planning
- d. Programmers — R. Barrett, T. Hickey
- e. Language — FORTRAN IV
- f. Data Tables Used — None
- g. MED's Used — None
- h. Writeup Completion Date — 11/30/67
- i. Parameter Lists — Table PMMTISS-PCMVMR-1
- j. Flowchart — Figure PMMTISS-PCMVMR-1

FUNCTION

Given radial and horizontal velocity changes for a maneuver, plus the chaser and target AEG blocks, this routine will do one of three options as requested by an input indicator. It will either compute a thrust parallel to the target plane (out-of-plane for chaser) and execute the maneuver, or compute just the out-of-plane component, or execute the maneuver using the input components. Pitch and yaw altitude angles are also computed for the first two options.

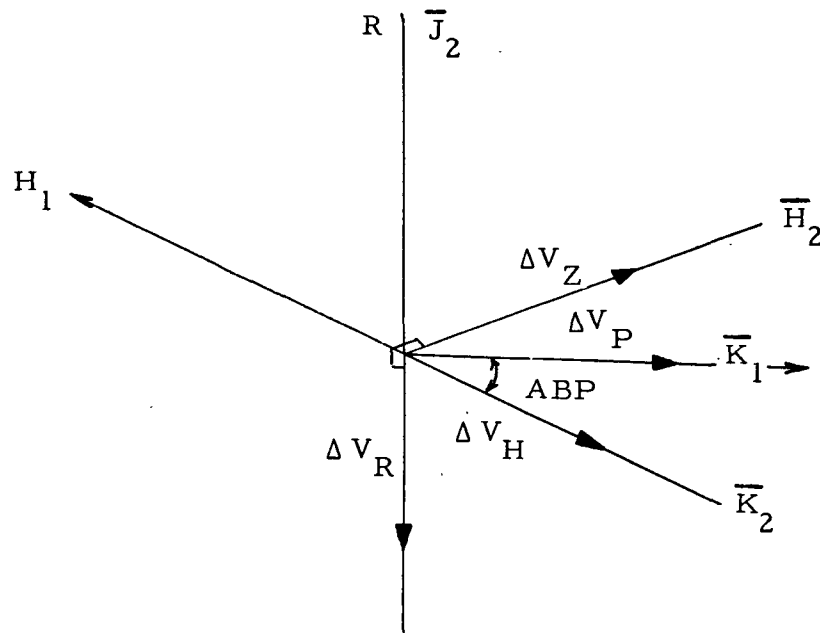
USAGE

- a. Calling Sequence — CALL PCMVMR (CHASER, TARGET, DELVX, DELVY, DELVZ, MU, PITCH, YAW, IND)
- b. Inputs — See Table PMMTISS-PCMVMR-1
- c. Outputs and Exit — Osculating elements after the maneuver, pitch, and yaw of the maneuver. Exit is by "RETURN".
- d. Other Programs Used — GIMIKC, GIMKIC, DCOS, DSQRT, PINRML, DATAN, DTAN, DARCOS

METHOD

Using the chaser as a point of reference, a coordinate system is found so that one component is normal to the chaser orbital plane ($\bar{H}_2$); the second component is along the chaser radius vector ($\bar{J}_2$), and the third makes a right-hand system ($\bar{K}_2$). Note that $\bar{K}_2$ is also in the direction of chaser horizontal velocity. See figure below.

Approval	Approval
<i>E. Barrett</i>	<i>A. Hickey</i>



If the target normal ($\bar{H}_1$) is crossed with $\bar{J}_2$, the resulting normal vector ($\bar{K}_1$) is in the plane formed by $\bar{H}_2$ and $\bar{K}_2$. The angle between $\bar{K}_1$ and $\bar{K}_2$ (ABP) is found by the following expression.

$$ABP = \frac{\bar{H}_2 \cdot \bar{K}_1}{|\bar{H}_2 \cdot \bar{K}_1|} \cos^{-1} \left(\frac{\bar{K}_1 \cdot \bar{K}_2}{|\bar{K}_2| |\bar{K}_1|} \right)$$

hence $180^\circ \leq ABP \leq 180^\circ$; that if $\bar{K}_1$ is on the same side of the plane formed by $\bar{J}_2$ and $\bar{K}_2$ as $\bar{H}_2$, then $ABP \geq 0$; otherwise, $ABP < 0$.

Yaw is measured in the $\bar{H}_2, \bar{K}_2$ plane clockwise; hence yaw = -ABP.

The out-of-plane velocity $\Delta V_Z = \Delta V_H \tan (ABP)$ and the velocity in the $\bar{H}_2, \bar{K}_2$ plane are found by this equation $\Delta V_P = \frac{\Delta V_H}{(\cos(ABP))}$

Therefore the total change in velocity, ΔV_T is found by:

$$\Delta V_T = \sqrt{\Delta V_P^2 + \Delta V_R^2}$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMTISS-PCMVMR
DATE 1/8/68 (EORP)
REV
PAGE 3 of 6

The pitch $P = \tan^{-1} \left[\frac{\Delta V_R}{\Delta V_P} \right]$; the minus sign is because the pitch is measured in the negative $\bar{J}_2$ direction.

The new velocity vector may be computed from the following expression:

$$\mathbf{V} = \mathbf{V} + (\Delta V_H) \bar{K}_2 - (\Delta V_R) \bar{J}_2 + (\Delta V_Z) \bar{H}_2$$

Because the maneuver does not change the radius vector at the time of the maneuver, the new velocity vector and input radius vector can now be converted to osculating Keplerian elements.



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMTISS-PCMV1

DATE 1/8/68 (EO

REV

PAGE 4 of 6

Table PMMTISS-PCMVMR-1. Parameter List

QUEID — NA

QNAME — NA

From — PMMTISS

To-Task — NA

Purpose — Execute maneuver

EP — NA

Byte No.	Type*	Units	Value / Description / Comments
1-4	F		Addr. of chaser AEG block
5-8	F		Addr. of target AEG block
9-12	F		Addr. of horizontal velocity component
13-16	F		Addr. of radial velocity component
17-20	F		Addr. of out-of-plane velocity component
21-24	F		Addr. of $\sqrt{\mu}$
25-28	F		Addr. of pitch result
29-32	F		Addr. of yaw result
33-36	F		Addr. of request indicator

*Type

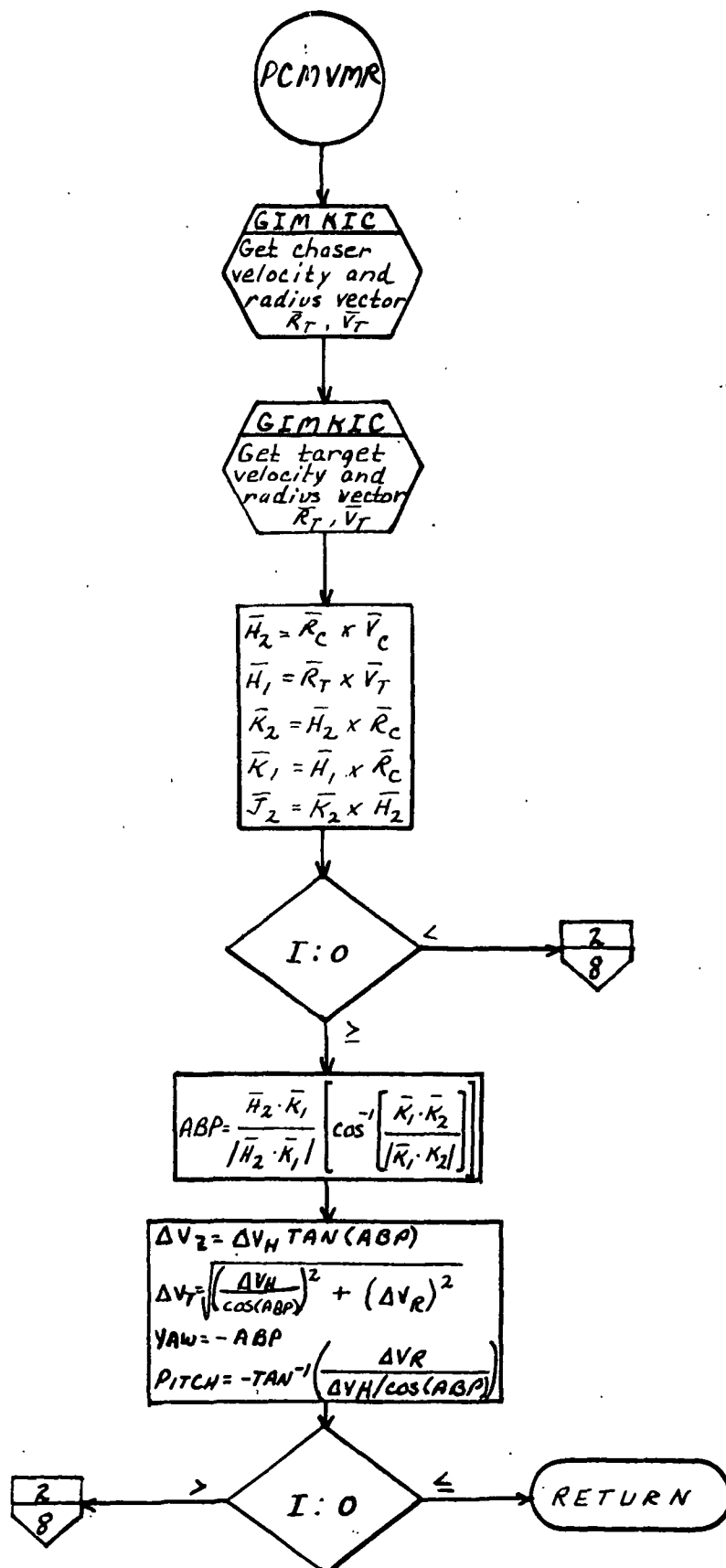
H — Fixed-point half word

F — Fixed-point full word

E — Single-precision (floating-point)

D — Double-precision (floating-point)

X — Hexadecimal

Figure PMMTISS-PCMVMR-1. Flowchart
(Sheet 1 of 2)

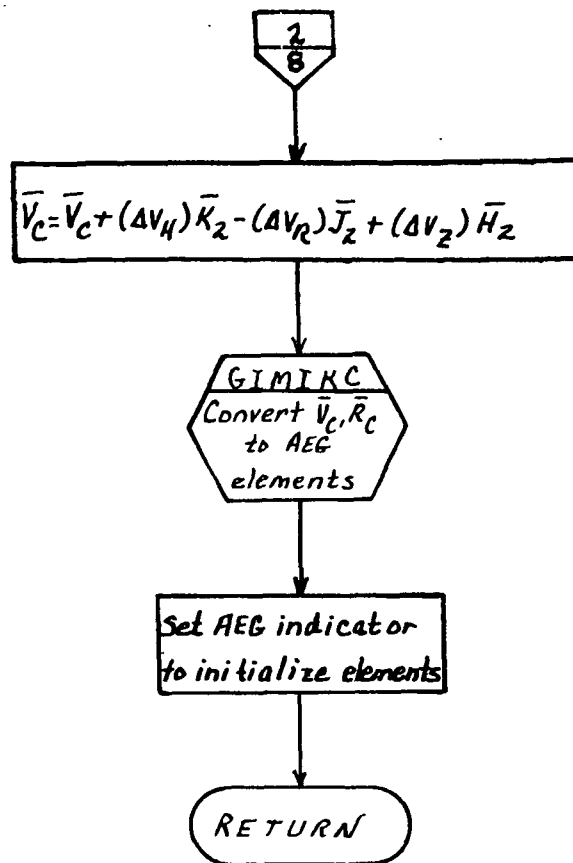


Figure PMMTISS-PCMVMR-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMUDT
 DATE 10/3/67 (EORP)
 REV
 PAGE 1 of 11

- a. Symbol — PMMUDT
- b. Program Name — Vehicle Orientation Change Processor
- c. Programming Group — Maneuver Transfer
- d. Programmer — W. Stone
- e. Language — Assembler
- f. Data Tables Used — PZMPTLEM, PZMPTCSM
- g. MED's Used — M08
- h. Writeup Completion Date — 9/8/67
- i. System Parameters — Table PMMUDT-1
- j. Parameter List and Diagram — Figure PMMUDT-1
- k. Flowchart — Figure PMMUDT-2

FUNCTION

PMMUDT changes vehicle orientation to "heads up" or "heads down". It also changes to the use of system parameters for thruster trim angles instead of GIMGBL or vice versa. One of the above items may be omitted.

USAGE

- a. Calling Sequence — LINK EP=PMMMPT (NOTE: Reg 1 points to MED input area.)
- b. Inputs — Reg 1 points to message block which points to MED input area which contains addresses of input items (vehicle code, maneuver number, orientation GIMBLE flag). (See Figure PMMUDT-1)
- c. Outputs and Exit — Changes a block of the mission plan table. Exits via "RETURN".
- d. Other Programs Used — GIMGBL, ELGETE, GIMIKC, GIMKIC and FORTRAN Library Routines.
- e. Data Restrictions and Limits — Maneuver must be in mission plan table.
- f. Error — 53 - Cannot obtain vector prior to maneuver
 50 - AEG error; i.e., cannot use vector obtained
 48 - Maneuver not in mission plan table.
 Above error codes set into Reg 15.

METHOD

The first item in this program after error checking is to "CALL ELGETE" and get a vector from the ephemeris. If this vector is in a maneuver or if a vector cannot be obtained, the burnout vector from the previous maneuver is used.

Approval	Approval
<i>Wesley Stone</i>	<i>S.E. James</i>

The vector obtained is advanced using PMMAEG to the time of the specified thruster on.

$$\hat{Y}_t = \frac{\hat{a}_t \times \bar{r}}{|\hat{a}_t \times \bar{r}|} \quad \text{by definition}$$

If the "heads-up, heads-down" indicator was not input, then $\hat{Y}_t$ is dotted with $\hat{Y}_B$ (Y body axis); if positive, a heads-up indicator is set. Otherwise a heads-down indicator is set.

If the thruster is one of the RCS thrusters, the following holds true.

For heads up with RCS + X thruster

$$\left\{ \begin{array}{l} \hat{X}_B = +\hat{a}_t \\ \hat{Y}_B = \hat{Y}_t \end{array} \right\}$$

For heads up with RCS - X thruster

$$\left\{ \begin{array}{l} \hat{X}_B = -\hat{a}_t \\ \hat{Y}_B = -\hat{Y}_t \end{array} \right\}$$

For heads down with RCS + X thruster

$$\left\{ \begin{array}{l} \hat{X}_B = \hat{a}_t \\ \hat{Y}_B = -\hat{Y}_t \end{array} \right\}$$

For heads down with RCS - X thruster

$$\left\{ \begin{array}{l} \hat{X}_B = -\hat{a}_t \\ \hat{Y}_B = \hat{Y}_t \end{array} \right\}$$

For all of the above cases, the "Z" body axis is as follows:

$$\hat{Z}_B = \hat{X}_B \times \hat{Y}_B.$$

If the thruster is not RCS, the gimble angles are necessary. If the gimble angle indicator was not input, use what was used in computation of this maneuver found in the Mission Plan Table. In case of heads-down, define $\hat{Y}_T = -\hat{Y}_T$. Now both cases, $\hat{Z}_T = \hat{X}_T \times \hat{Y}_T$. Depending upon the gimble angle indicator, either compute the gimble angles or pick up the system parameter values for them.

Then

$$\hat{X}_B = \cos P_G \cos Y_G \hat{a}_t - \cos P_G \sin P_G \hat{Y}_t + \sin P_G \hat{Z}_t$$

$$\hat{Y}_B = \sin Y_G \hat{a}_t + \cos Y_G \hat{Y}_t$$

$$\hat{Z}_B = \hat{X}_B \times \hat{Y}_B.$$

The only remaining quantities to be computed are the local-vertical local-horizontal Pitch, Yaw and Roll angles. First define the following platform axes.

$$\hat{Y}_P = \frac{\bar{V}_{IG} \times \bar{r}_{IG}}{\bar{V}_{IG} \times \bar{r}_{IG}} \quad \text{out of plane of motion}$$

$$\hat{Z}_P = \frac{-\bar{r}_{IG}}{|\bar{r}_{IG}|} \quad \text{along position vector toward center of Earth}$$

$$\hat{X}_P = \hat{Y}_P \times \hat{Z}_P \quad \text{completes right hand system.}$$

$$\text{now Pitch} = \sin^{-1} \left\{ \frac{\hat{Z}_P \cdot \hat{X}_B}{\|\hat{Z}_P\| \|\hat{X}_B\|} \right\} \quad -\frac{\pi}{2} < P < \frac{\pi}{2}$$

$$\text{Roll} = \tan^{-1} \left\{ \frac{\hat{Y}_P \cdot \hat{X}_B}{\hat{X}_P \cdot \hat{X}_B} \right\} \quad 0 \leq R \leq 2\pi$$

$$\text{Yaw} = \tan^{-1} \left\{ \frac{\hat{Z}_P \cdot \hat{Y}_B}{\hat{Z}_P \cdot \hat{Z}_B} \right\} \quad 0 \leq Y \leq 2\pi$$

The above is not true if Pitch is near $\frac{\pi}{2}$ or $-\frac{\pi}{2}$.

For these cases

$$\text{Roll} = 0.$$

$$\text{Yaw} = \tan^{-1} \left\{ \frac{-\hat{X}_P \cdot \hat{Y}_B}{\hat{Y}_B \cdot \hat{Y}_P} \right\}.$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMUDT

DATE 10/3/67 (EORP)

REV

PAGE 5 of 11

Table PMMUDT-1. System Parameters

Symbol in Program	Symbol In Master Tape	Type*	Units	Comments
SRMU	MCERMU	D	$\sqrt[3]{\text{CUL/hr.}^2}$	Gravitation Constant
	MCTDRG	D	rad.	System Parameter DPS Roll Gimble
	MCTDPG	D	rad.	System Parameter DPS Pitch Gimble
	MCTSYG	D	rad.	System Parameter SPS Yaw Gimble
	MCTSPG	D	rad.	System Parameter SPS Pitch Gimble
	MCTARG	D	rad.	System Parameter APS Roll Gimble
	MCTAPG	D	rad.	System Parameter APS Pitch Gimble

*Type
 H — Fixed-point half word
 F — Fixed-point full word
 E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

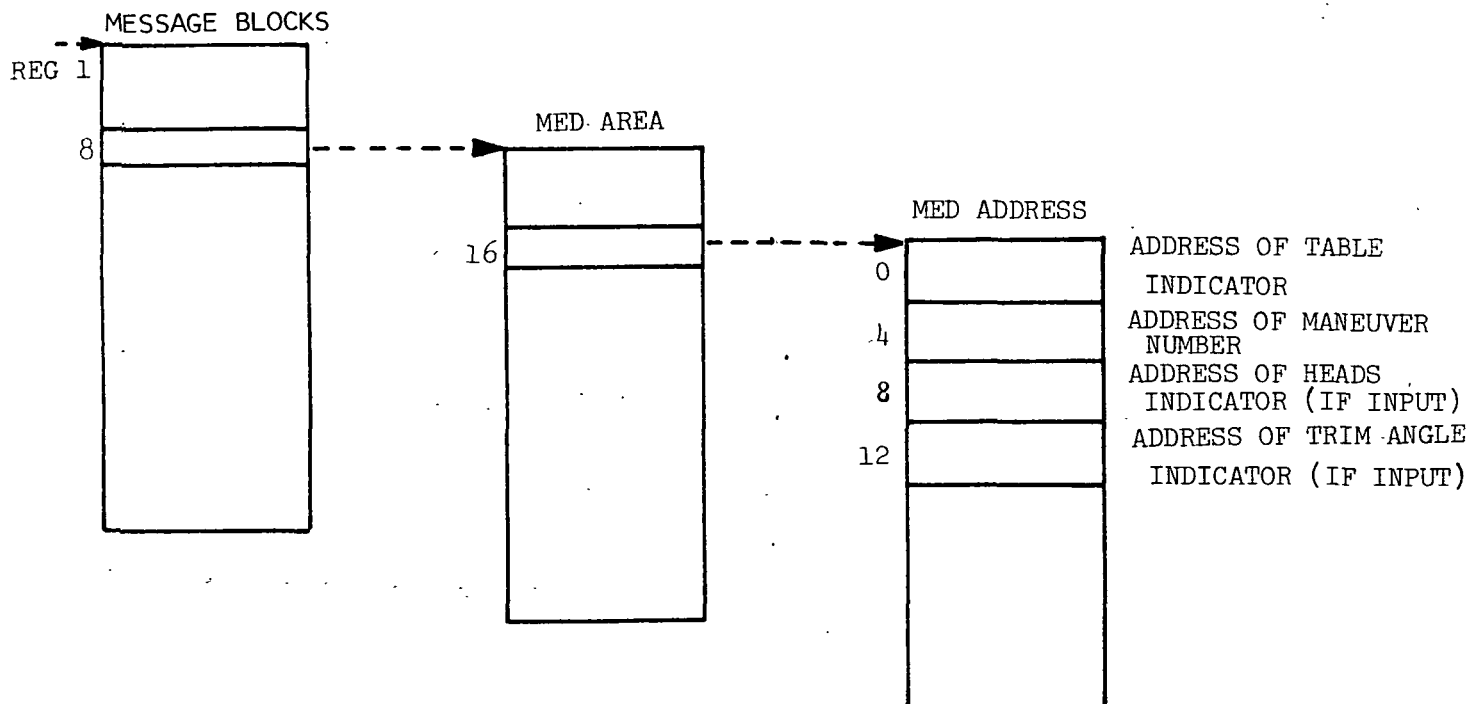
Figure PMMUDT-1. Parameter List and Diagram

Upon entry to this program Reg 1 must point to a message area. Bytes eight through eleven (8-11) of the message area must point to the MED input area. Bytes sixteen through nineteen (16-19) of the MED input area must point to the MED address area.

- a. Bytes zero through three (0-3) of MED address area will contain the address of the table number (1=CSM; 3=LM).
- b. Bytes four through seven (4-7) of the MED address area will contain the address of the maneuver number (1-15).
- c. Bytes eight through eleven (8-11) of the MED address area will contain the address of the heads indicator $\begin{pmatrix} 0 = \text{down} \\ 1 = \text{up} \end{pmatrix}$, if the address is "zero", the item is not input.
- d. Bytes twelve through fifteen (12-15) of the MED address area will contain the address of the Trim angle indicator $\begin{pmatrix} 0 = \text{compute} \\ 2 = \text{system parameter} \end{pmatrix}$ if the address is "zero", the item is not input.

NOTE: The above indicators are the same for all MED input formats.

The following diagram illustrates the information above.



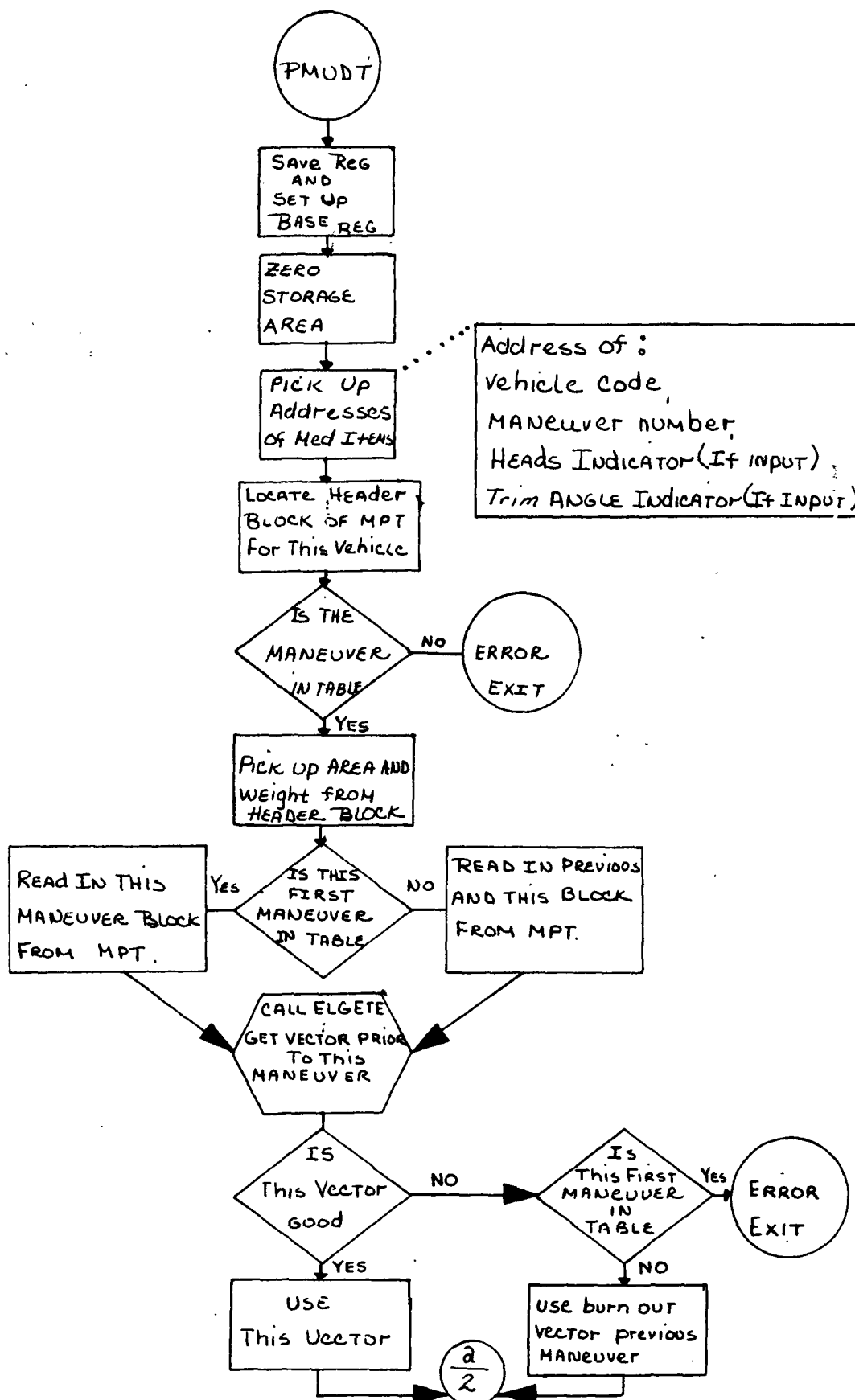


Figure PMMUDT-2. Flowchart

(Sheet 1 of 5)

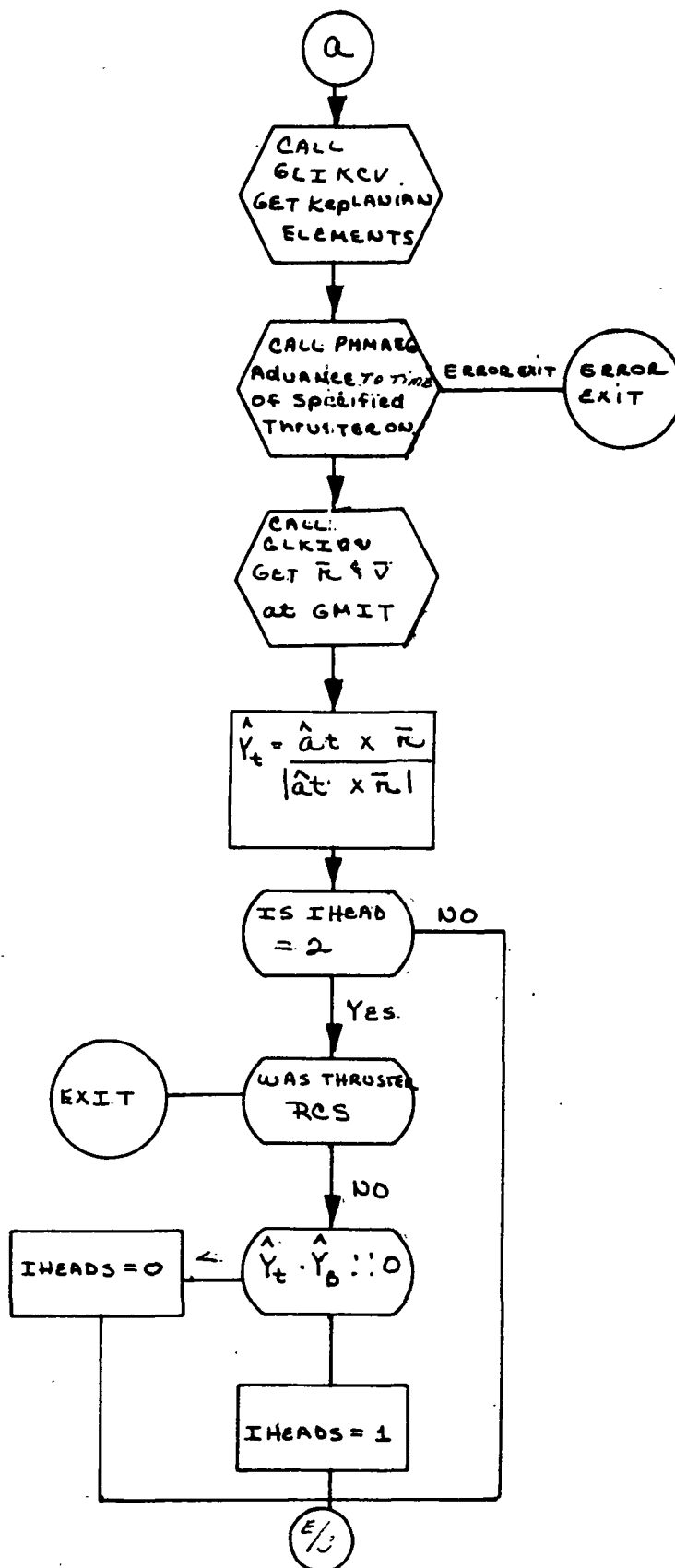


Figure PMMUDT-2. Flowchart

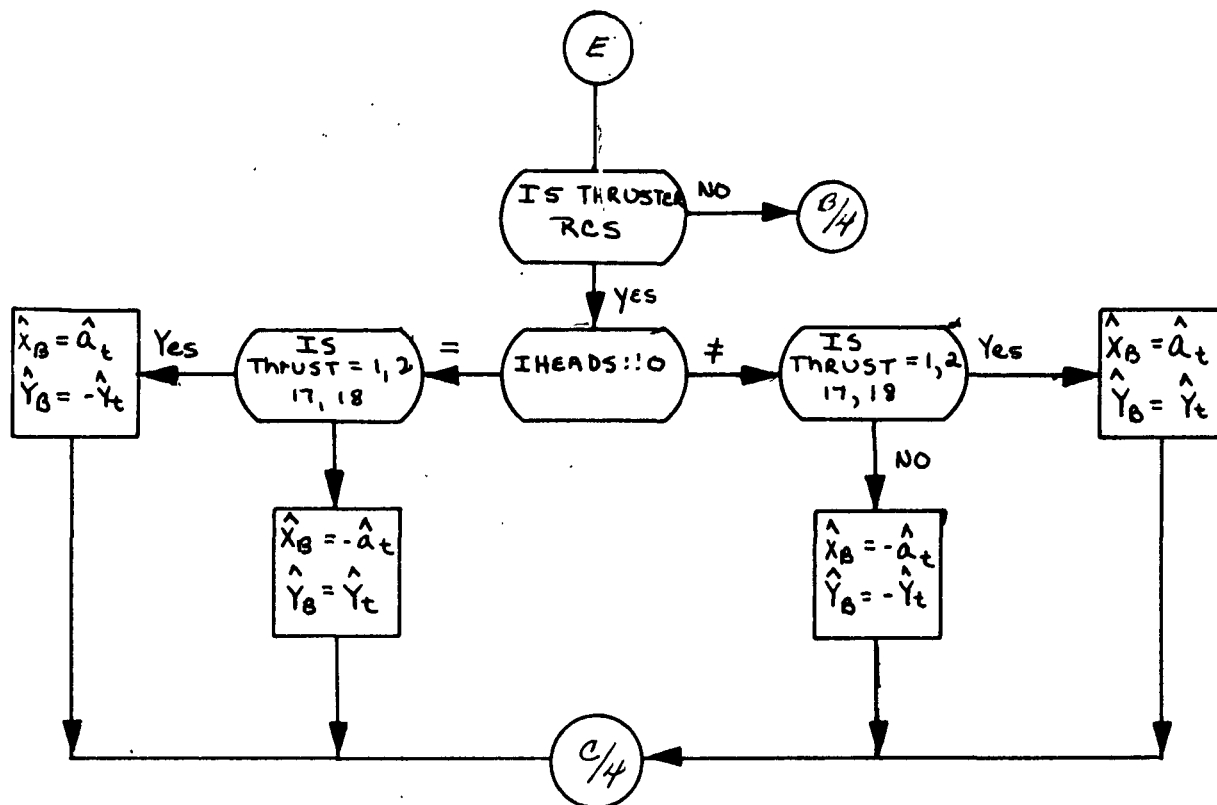


Figure PMMUDT-2. Flowchart

(Sheet 3 of 5)

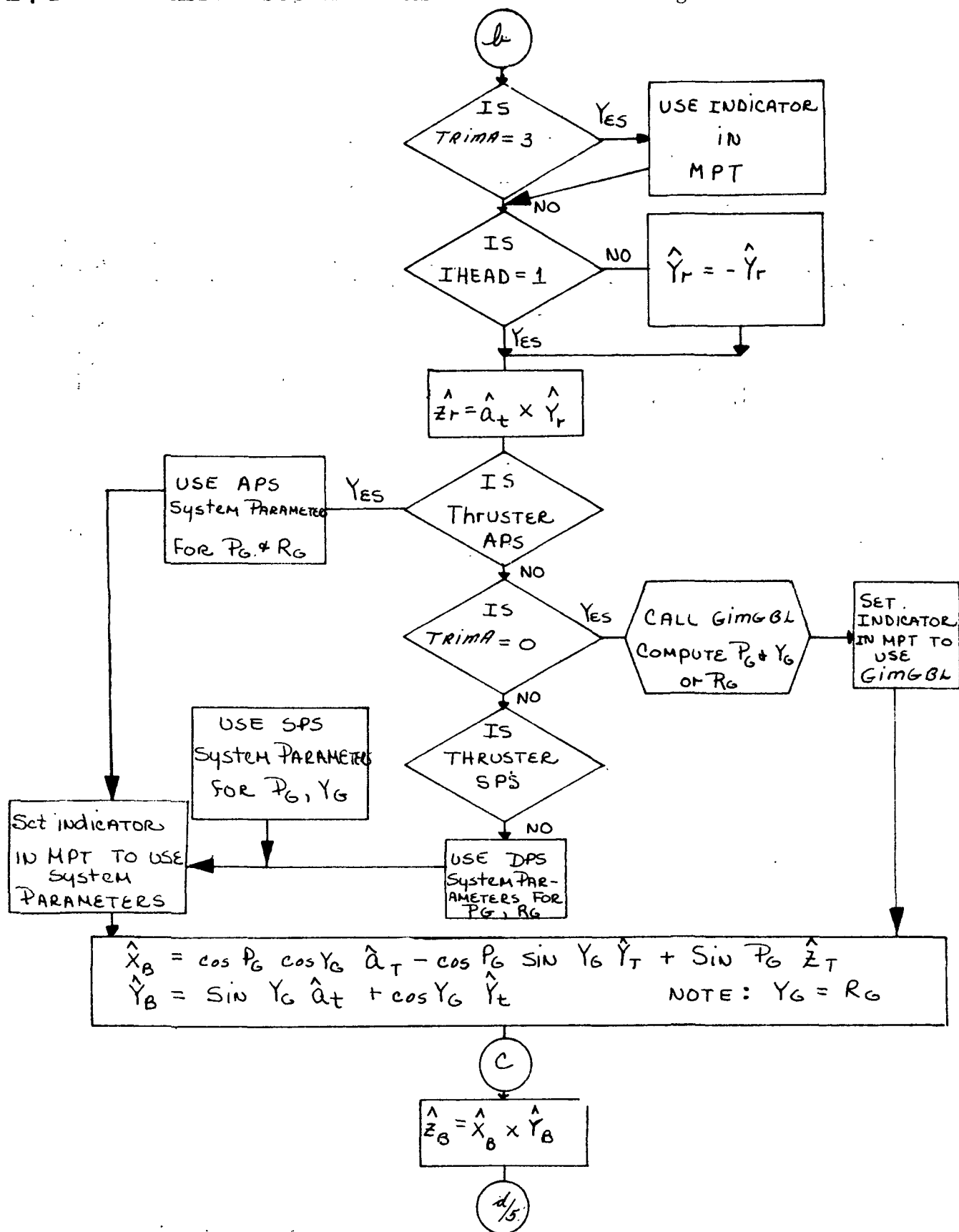


Figure PMMUDT-2. Flowchart

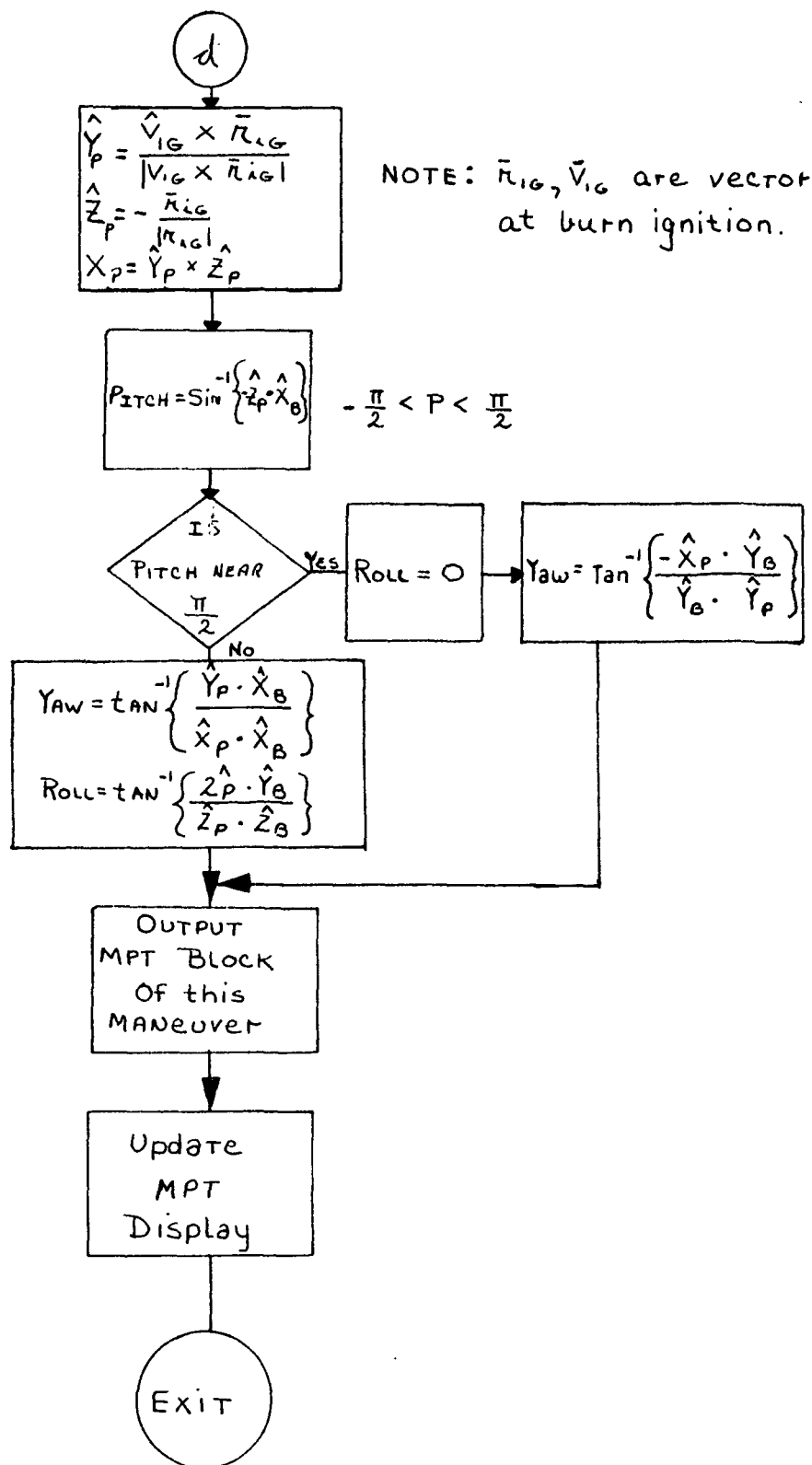


Figure PMMUDT-2. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMXFR
DATE 1/31/68 (206)
REV
PAGE 1 of 32PMMXFR
MANEUVER TRANSFER LOAD MODULEREFERENCES

1. Programming Group — Maneuver Control (Mission Planning)
2. Programmer — B. Oliver
3. Language — Assembler
4. Writeup Completion Date — 1/5/68
5. MED's Used — M01, M65, M66
6. Data Tables Used — PZMPTLEM, GZLTRA, PZGPMELM, PZGPMDIS
7. System Parameters — Table 1
8. Parameter Lists — Tables -2 through -10
9. Flowchart — Figure 1

FUNCTION

As a result of MED action, PMMXFR receives queues to control the processing necessary to place designated maneuvers in the Mission Plan Table (MPT), which are subsequently reflected in the vehicle trajectory.

USAGE

1. Calling Sequence — RTATTACH TASK=PTMANC, EP=PMMXFR, GSPV=1, ID=n, PARAM=(SUBPOOL, 4)
2. Inputs — MED parameters described in Tables PMMXFR-2 through -4; maneuver defining parameters from data tables GZLTRA, PZGPMELM, and PZGPMDIS.
3. Outputs and Exit — The output is a maneuver description in the MPT. Exit is via RETURN (14, 12) or XCTL with control passed to PMSVCT.
4. Other Programs Used — DSQRT, DARCOS, ELGETE, EMGVECST, GMSCTRL, PMMMPT, PMSVCT, PMXSPT
5. Data Restrictions and Limits — None
6. Error — The following on-line prints are a result of recognizing error conditions.
 - a. MANEUVER PRIOR TO PRESENT TIME
 - b. MANEUVER XX TO BE REPLACED NOT IN MPT
 - c. CODE XX TO BE REPLACED OVERLAPS ANOTHER MANEUVER
 - d. MANEUVER PRIOR TO FROZEN MANEUVER
 - e. REQUESTED MANEUVER TRANSFER REJECTED — MPT IS FULL

Approval	Approval
<i>Robert E. Oliver</i>	<i>S. E. James</i>

- f. REQUESTED MANEUVER TRANSFER OVERLAPS ANOTHER MANEUVER
- g. REQUESTED THRUSTER AND/OR ATTITUDE ILLEGAL FOR THIS MANEUVER TRANSFER
- h. ILLEGAL ENTRY TO PMMXFR
- i. UNUSABLE MANEUVER FOR TRANSFER
- j. UNABLE TO FETCH VECTOR FOR DIRECT INPUT MANEUVER — MPT UNCHANGED
- k. UNABLE TO FETCH VECTOR FOR LAUNCH/ORBIT TRANSFER — MPT UNCHANGED

METHOD

The Maneuver Transfer Load Module solves the ground rules and controls the processing necessary to place the following classes of maneuvers into the MPT.

Contingency insertion maneuver
Direct input maneuver
General purpose maneuver (GPM).

The parameters defining the contingency insertion maneuver are made available by processing during the launch phase. Upon manual entry of the MED M01 indicating that this maneuver is desired in the MPT, PMMXFR obtains the maneuver parameters along with the launch computed maneuver vector and simulates a direct input to PMMMPT. The transfer of this maneuver automatically deletes all other maneuvers from the MPT. The contingency insertion maneuver is zero delta velocity and is performed using the RCS thruster in the fixed inertial mode. Spacecraft attitudes at burn initiate are such that the initial thrust is in the orbital plane and in the direction of the Earth's horizon. The computations of the parameters necessary to define the maneuver to the powered flight integrator are performed by PMMMPT. An ephemeris anchored on the maneuver vector is then generated.

A future maneuver may be placed in the MPT by direct input of the parameters that describe the maneuver or by transfer of the impulsive maneuver parameters from the GPM table. PMMXFR determines the validity of the input parameters and sets up inputs to the Mission Plan Table Processor, PMMMPT, which performs the necessary computations to describe the maneuver to the powered flight integrator. Control is then passed to the Vector Routing Module, PMSVCT, and a trajectory update is performed.

A more detailed description of the processing performed may be found in the flow-chart, Figure PMMXFR-1.

DTROUTE, RTQCNTRL, RTATTACH Macros

Macro — RTATTACH TASK=GTMSCTRL,EP=GMSCCTRL,QNAME=GQ1CTL1,
ID=3,QRANK=QRANK20,DPRIOR=MKDGVF,PARAM=(CODE,72)

Purpose — Route insertion vector for trajectory update as a result of a
source change

Macro — DTROUTE PRMVREXE,RMODE=ENABLE,RTYPE=TIMK,QNAME=
GQ1CTL,TASK=GTMSCTRL,EP=GMSCCTRL,DPRIOR=MKDGMP,
MSGCNT=1,QRANK=QRANK12,START=(L,DTRTIME),ID=27 (Vehicle
code stored in ID during execution)

Purpose — Time out queue for maneuver execution.

Macro — RTATTACH TASK=GTMSCTRL,EP=GMSCCTRL,QNAME=GQ3CTL,ID=12,
QRANK=QRANK10,DPRIOR=MKDGMP

Purpose — Indicate PTMANC task completion.

Table 1. System Parameters

[illegible]

*Type
H — Fixed-point half word
F — Fixed-point full word
E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS
 2. PMMXFR
 DATE 1/31/68 (206)
 REV
 PAGE 5 of 32

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

Table 2. Parameter List
RTATTACH

QUEID — 31

QNAME — Common

From — GMSCTRL

To-Task — PTMANC

Purpose — Manual override of normal
insertion

EP — PMMXFR

Byte No.	Type*	Units	Value / Description / Comments
1-4	F		Pointer to the following MED parameter in Subpool 1.
n	F		Vector selection: 0 $\Rightarrow$ Preburn vector is selected, and maneuver is placed in the MPT. Nonzero $\Rightarrow$ Postburn vector selected.

 *Type
 H — Fixed-point half word
 F — Fixed-point full word

 E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

Table 3. Parameter List
 RTATTACH

QUEID — 29 QNAME — Common
 From — GM\$CTRL To-Task — PTMANC
 Purpose — General Purpose Maneuver EP — PMMXFR
 Transfer

Byte No.	Type*	Units	Value / Description / Comments
1-4	F		Pointer to the following MED parameters in Subpool 1:
n	F		Vehicle
n+4	F		Maneuver sequence number
n+8	F		Thruster code
n+12	F		Attitude code
n+16	D	hr.	Δt_u - Duration of ullage
n+24	D	rad.	Yaw
n+32	F		Number of thrusters for ullage

*Type
 H — Fixed-point half word
 F — Fixed-point full word
 E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMXFR
DATE 1/31/68 (206)
REV
PAGE 7 of 32Table 4. Parameter List
RTATTACH

QUEID — 33

QNAME — Common

From — GMSCTRL

To-Task — PTMANC

Purpose — Direct input of a maneuver to
the MPT

EP — PMMXFR

Byte No.	Type*	Units	Value / Description / Comments
1-4	F		Pointer to the following MED parameters in Subpool 1
n	F		Vehicle code
n+4	F		Maneuver sequence
n+8	D	hrs.	GMTBI
n+16	F		Thruster code
n+20	F		Attitude code
n+24	D	Er/hr.	Delta velocity of maneuver
n+32	D	hrs.	Duration of maneuver
n+40	D	hrs.	Duration of ullage
n+48	F		Coordinate indicator
n+52	F		Not used
n+56	D	rad.	Pitch
n+64	D	rad.	Yaw
n+72	D	rad.	Roll
n+80	F		Separation indicator
n+84	F		Ullage thrusters
n+88	D		Target parameter
n+96	D		Target parameter
n+104	D		Target parameter
n+112	D		Target parameter
n+120	D		Target parameter

*Type

H — Fixed-point half word

F — Fixed-point full word

E — Single-precision (floating-point)

D — Double-precision (floating-point)

X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMXFR
DATE 1/31/68 (206)
REV
PAGE 8 of 32

Table 5. Parameter List
LINK

QUEID — NA

QNAME — NA

From — PMMXFR

To-Task — NA

Purpose — Request launch vector for
manual override

EP — EMGVECST

Byte No.	Type*	Units	Value / Description / Comments
1-4			Address of the parameters below
n	F		3-ID for source override
n+4	F		3-Vehicle index for the LM

*Type
H — Fixed-point half word
F — Fixed-point full word
E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMXFR
 DATE 1/31/68 (206)
 REV
 PAGE 9 of 32

 Table 6. Parameter List
 RETURN FROM LINK

QUEID — NA

QNAME — NA

From — EMGVECST

To-Task — NA

 Purpose — Return launch vector for
 manual override

EP — PMMXFR

Byte No.	Type*	Units	Value / Description / Comments
1	F		Not used
	F		Vehicle code
2	D	hrs.	Time of vector
3	D	Er	X
4	D	Er	Y
5	D	Er	Z
6	D	Er/hr.	$\dot{X}$
7	D	Er/hr.	$\dot{Y}$
8	D	Er/hr.	$\dot{Z}$
9	D	ft. ²	Vehicle area
10	D	lbs.	Vehicle weight

 *Type
 H — Fixed-point half word
 F — Fixed-point full word

 E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMXFR
 DATE 1/31/68 (206)
 REV
 PAGE 10 of 32
Table 7. Parameter List
LINK

QUEID — NA

QNAME — NA

From — PMMXFR

To-Task — NA

Purpose — Compute maneuver finite burn
parameters

EP — PMMMPT

Byte No.	Type*	Units	Value / Description / Comments
1-4	F		Address of the following parameters in Subpool 1
n			First block of LM Mission Plan Table
n+960	F		Vehicle code
n+964	F		Maneuver sequence
n+968	D	hrs.	GMTBI
n+976	F		Thruster code
n+980	F		Attitude code
n+984	D	Er/hr.	Delta velocity of maneuver
n+992	D	hrs.	Duration of maneuver
n+1000	D	hrs.	Duration of ullage
n+1008	F		Coordinate indicator
n+1012	F		Not used
n+1016	D	rad.	Pitch
n+1024	D	rad.	Yaw
n+1032	D	rad.	Roll
n+1040	F		Separation indicator
n+1044	F		Ullage thrusters
n+1048	D		Target parameter
n+1056	D		Target parameter
n+1064	D		Target parameter
n+1072	D		Target parameter
n+1080	D		Target parameter

 *Type
 H — Fixed point half word
 F — Fixed-point full word

 E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMXFR
DATE 1/31/68 (206)
REV
PAGE 11 of 32

Table 7. Parameter List (Continued)

QUEID —

QNAME —

From —

To-Task —

Purpose —

EP —

Byte No.	Type*	Units	Value / Description / Comments
n+1088	F		Error flag (output)
n+1092	F		MPT block number
n+1096	F		Direct input flag
n+1100	F		Not used
n+1104	D	hrs.	Time of vector
n+1112	D	Er	X
n+1120	D	Er	Y
n+1128	D	Er	Z
n+1136	D	Er/hr.	$\dot{X}$
n+1144	D	Er/hr.	$\dot{Y}$
n+1152	D	Er/hr.	$\dot{Z}$
n+1160	D	ft. ²	Vehicle area
n+1168	D	lbs.	Vehicle weight
n+1176	D	hrs.	Start limit
n+1184	D	hrs.	End limit
n+1192	D	Er	A before
n+1200	D		E before
n+1208	D	rad.	I before
n+1216	D	rad.	L before
n+1224	D	rad.	G before
n+1232	D	rad.	H before
n+1240	D	Er	A after

*Type
H — Fixed point half word
F — Fixed point full wordE — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

IBM RTCC AFULLD PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMXFR
 DATE 1/31/68 (206)
 REV
 PAGE 12 of 32

Table 7. Parameter List (Continued)

QUEID —

QNAME —

From —

To-Task —

Purpose —

EP —

Byte No.	Type*	Units	Value / Description / Comments
n+1248	D		E after
n+1256	D	rad.	I after
n+1264	D	rad.	L after
n+1272	D	rad.	G after
n+1280	D	rad.	H after
n+1288	D	hrs.	Time impulsive
n+1296	D	rad.	F before
n+1304	D	rad.	F after
n+1312			Not used
n+1336			Maneuver code
n+1344	H		Not used
n+1346	H		Maneuver purpose code
n+1348	F		Not used
n+1352	F		Location of EBCDIC MPT name
n+1356	F		Not used
n+1360	D		Not used
n+1368	D	ft. ²	Undocking vehicle area
n+1376	D	lbs.	Undocking vehicle weight

 *Type
 H — Fixed-point half word
 F — Fixed-point full word

 E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMXFR
 DATE 1/31/68 (206)
 REV
 PAGE 13 of 32

 Table 8. Parameter List
 LINK

QUEID — NA

QNAME — NA

From — PMMXFR

To-Task — NA

Purpose — Print on-line error messages

EP — PMXSPT

Byte No.	Type*	Units	Value / Description / Comments
1-2	H		Error print code 1 — "Maneuver prior to present time." 2 — "Maneuver XX to be replaced not in MPT." 3 — "Code XX to be replaced overlaps another maneuver." 4 — "Maneuver prior to frozen maneuver." 5 — "Requested maneuver transfer rejected — MPT is full." 6 — "Requested maneuver transfer overlaps another maneuver." 9 — "Requested thruster and/or attitude illegal for this maneuver transfer." 37 — "Illegal entry to PMMXFR." 38 — "Unusable maneuver for transfer." 39 — "Unable to fetch vector for direct input maneuver — MPT unchanged." 54 — "Unable to fetch vector for launch/orbit transfer — MPT unchanged."

*Type

 H — Fixed-point half word
 F — Fixed-point full word

 E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMXFR
 DATE 1/31/68 (206)
 REV
 PAGE 14 of 32

 Table 9. Parameter List
 RTATTACH

QUEID — 3

QNAME — GQ1CTL1

From — PMMXFR

To-Task — GTMSCTRL

 Purpose — Route insertion vector for trajectory
 update as a result of a source change

EP — GMSCTRL

Byte No.	Type*	Units	Value / Description / Comments
1	F		Not used
	F		Vehicle code
2	D	hrs.	Time of vector
3	D	Er	X
4	D	Er	Y
5	D	Er	Z
6	D	Er/hr.	$\dot{X}$
7	D	Er/hr.	$\dot{Y}$
8	D	Er/hr.	$\dot{Z}$
9	D		EBCDIC source ID

 *Type
 H — Fixed-point half word
 F — Fixed-point full word

 S — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS
 2. PMMXFR
 DATE 1/31/68 (206)
 REV
 PAGE 15 of 32

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 Table 10. Parameter List
 XCTL

QUEID — 8

QNAME — NA

From — PMMXFR

To-Task — NA

Purpose — Causes a trajectory update

EP — PMSVCT

Byte No.	Type*	Units	Value / Description / Comments
0			Not used
4	F		Vehicle for update
8	F		Zero

 *Type
 H — Fixed-point half word
 F — Fixed-point full word

 E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

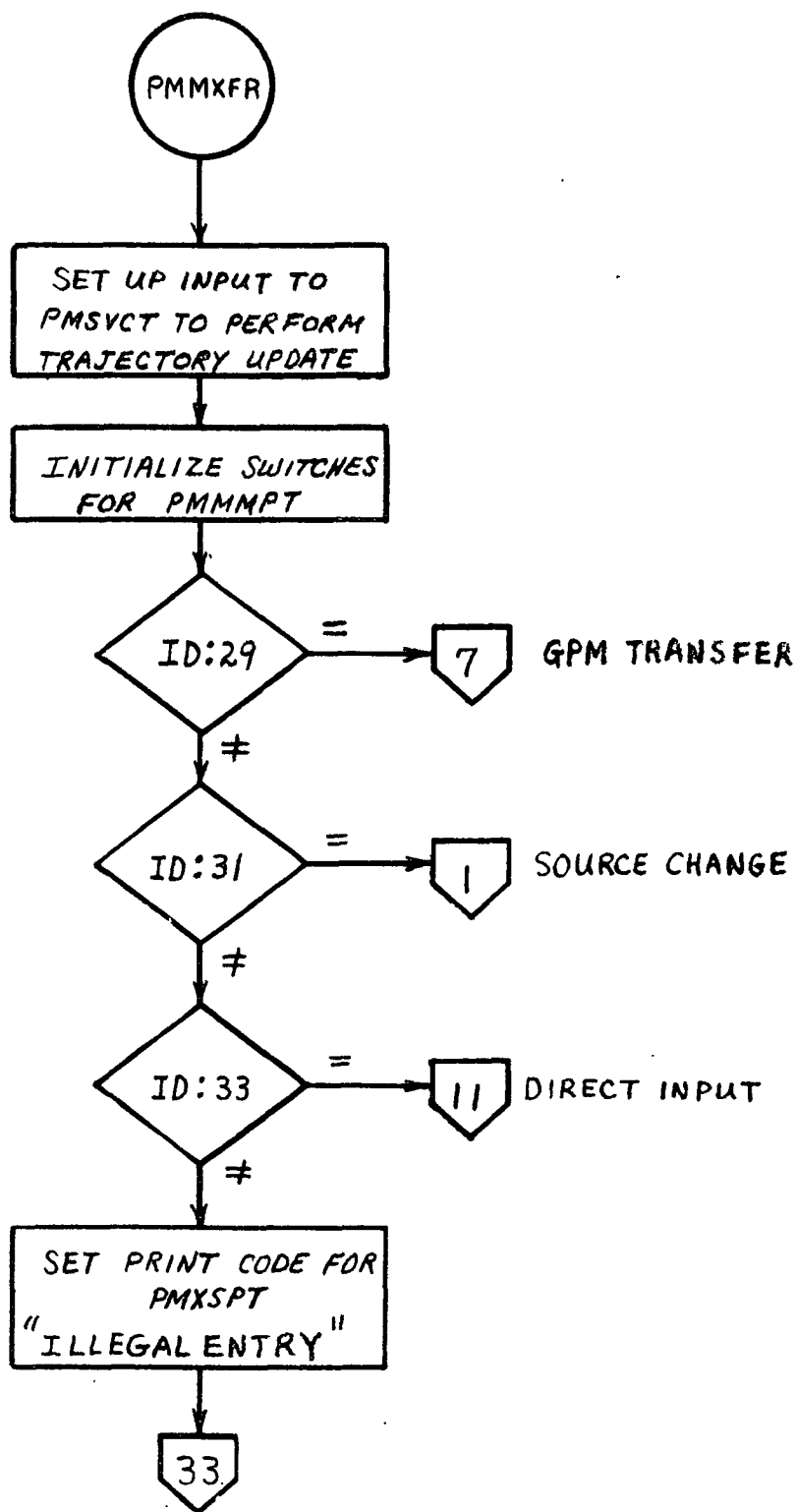


Figure 1

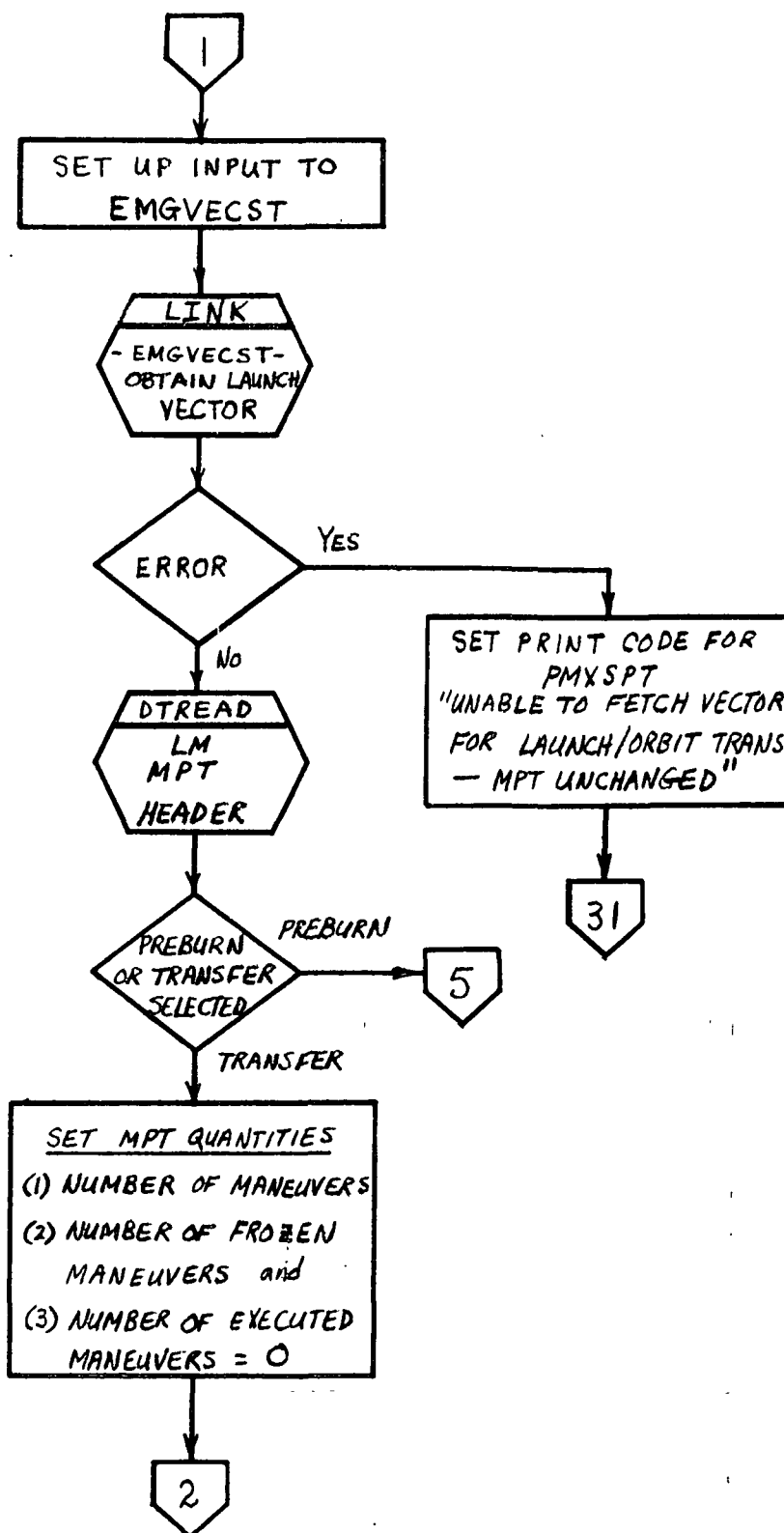


Figure 1

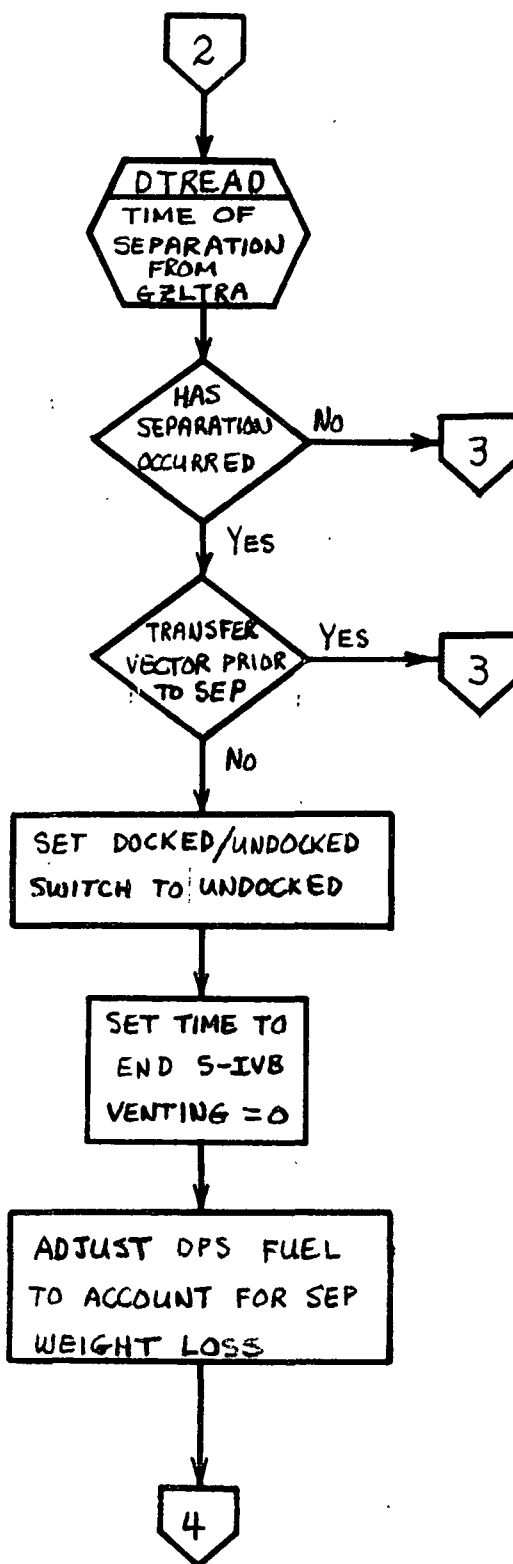


Figure 1

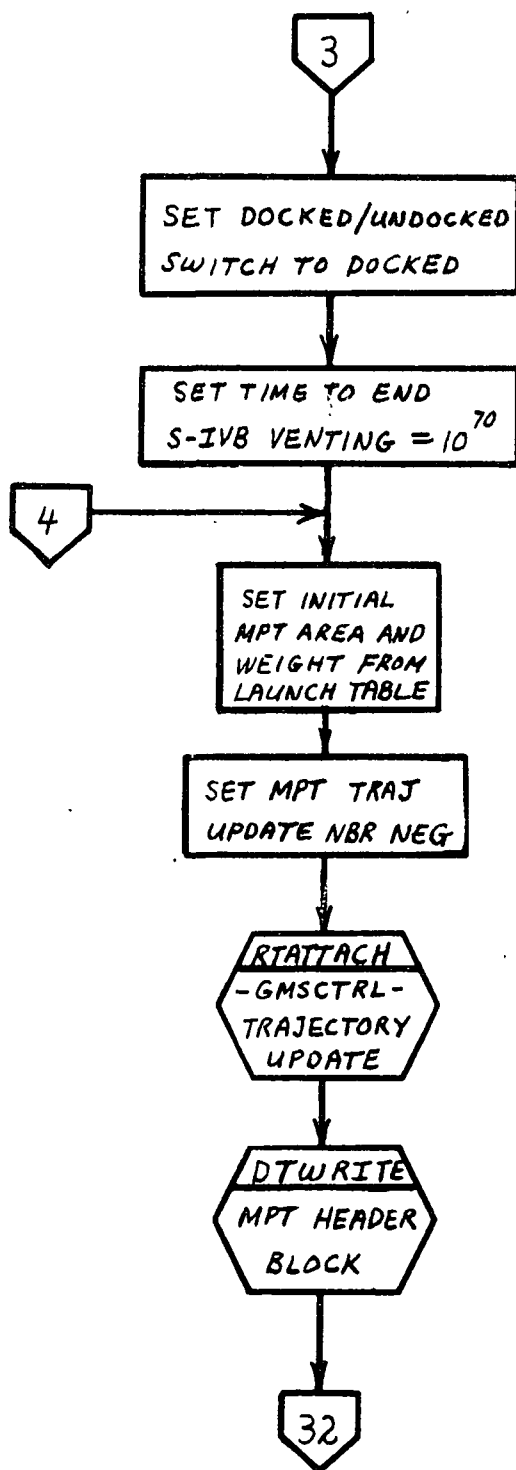


Figure 1

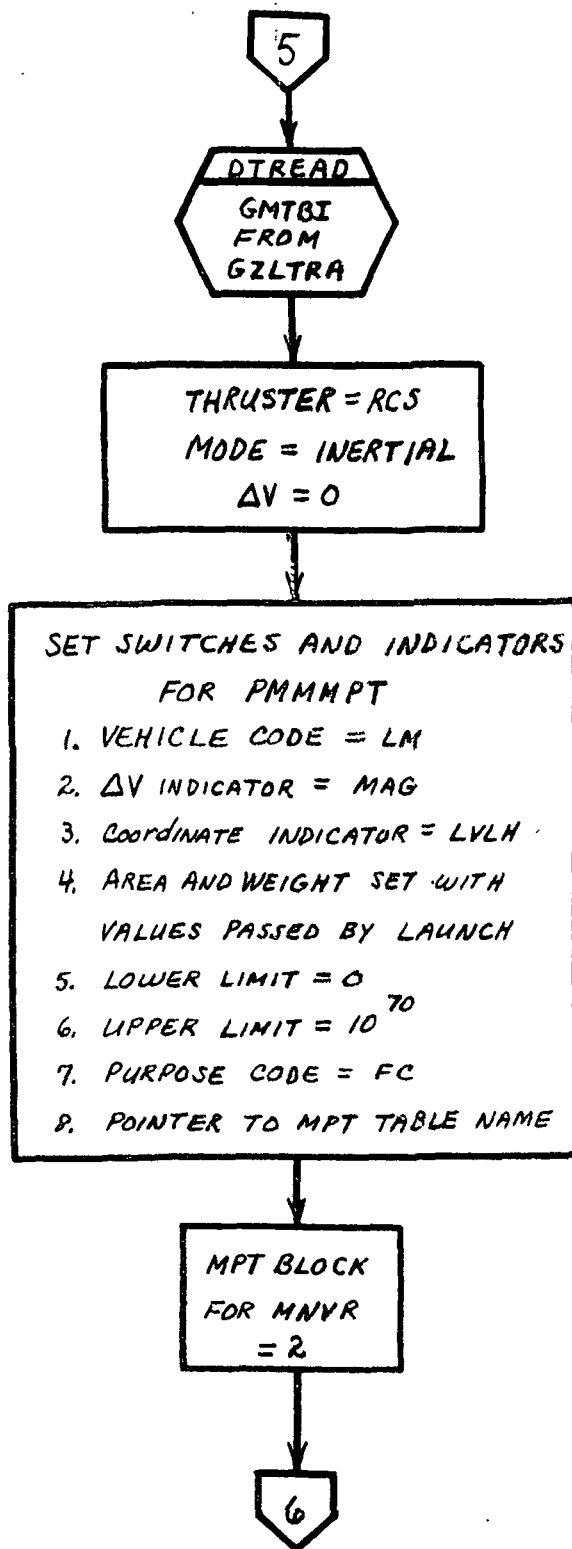


Figure 1

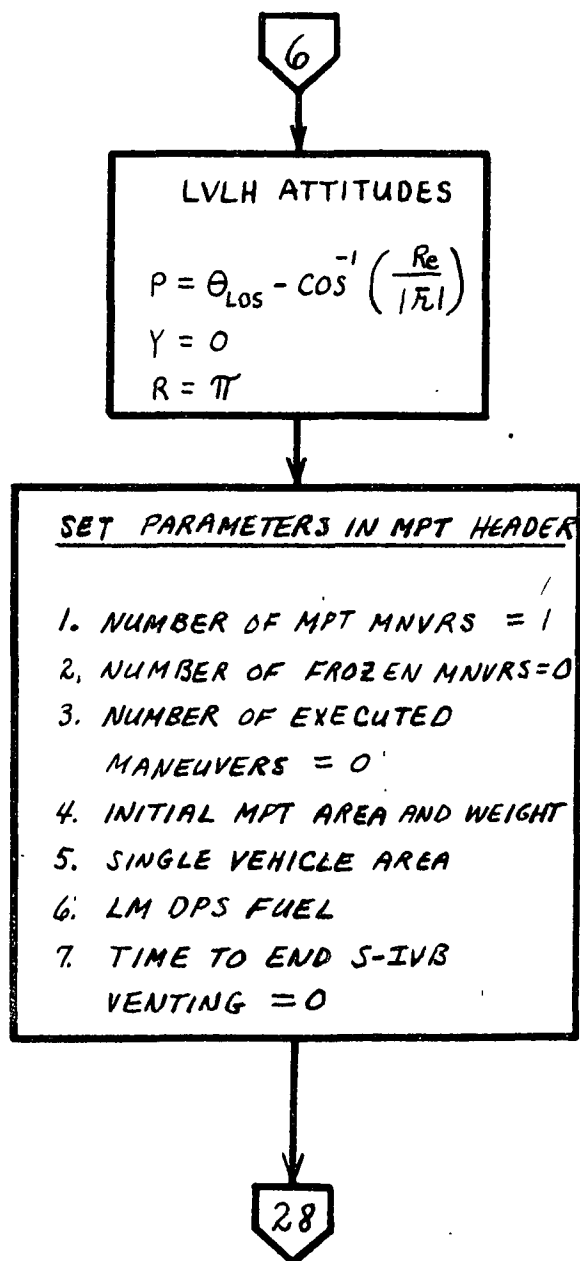


Figure 1

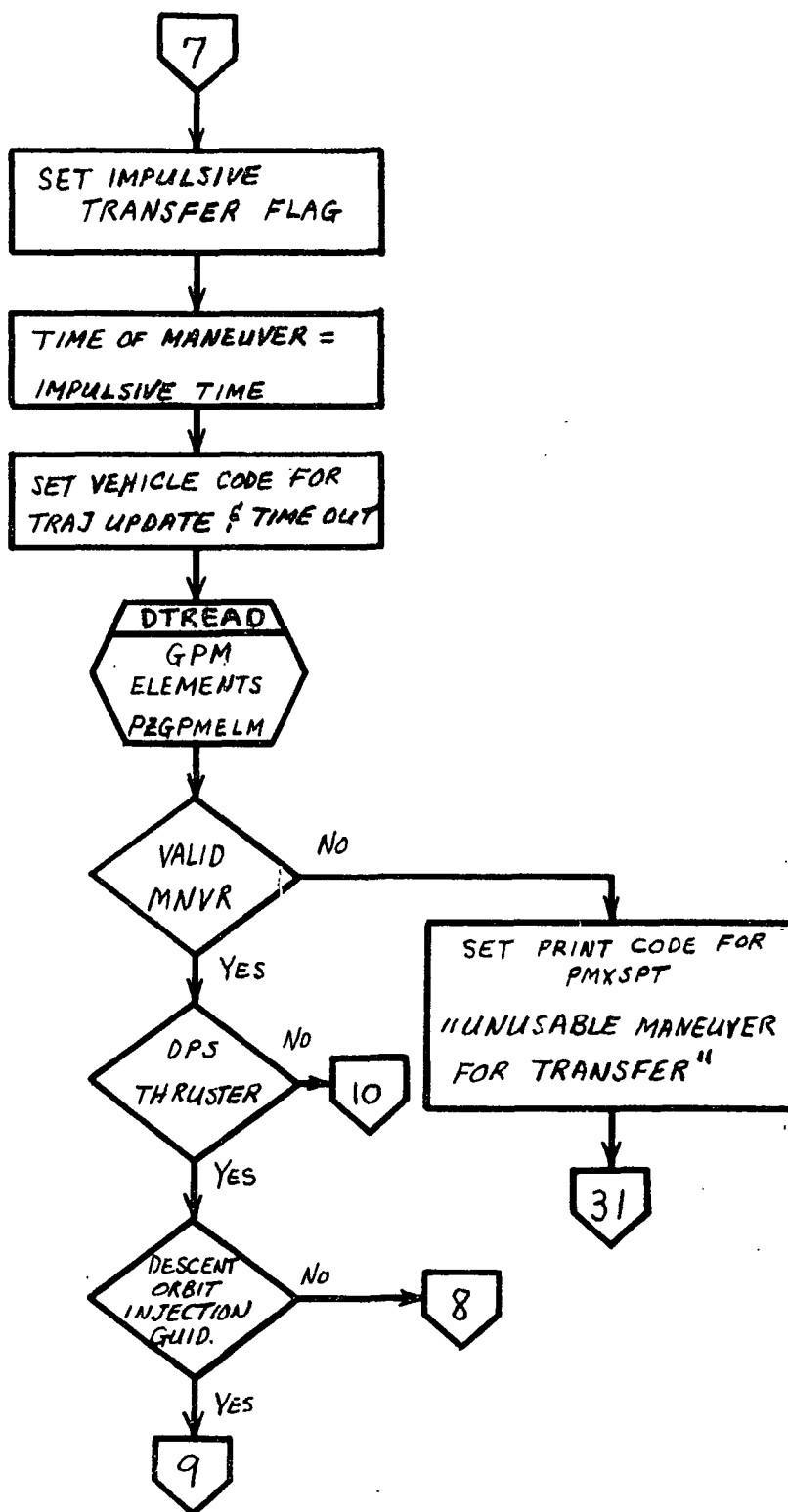


Figure 1

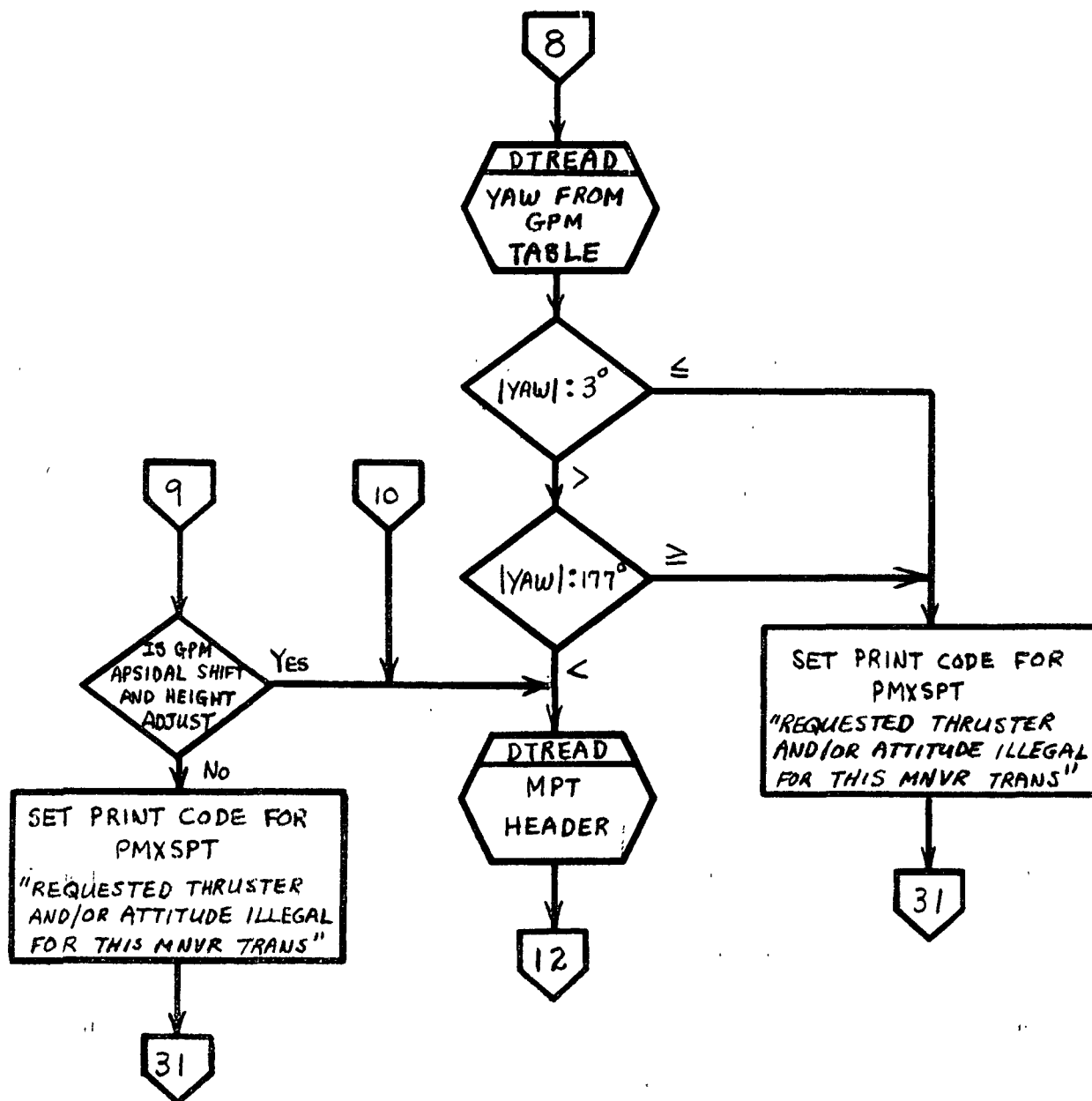


Figure 1

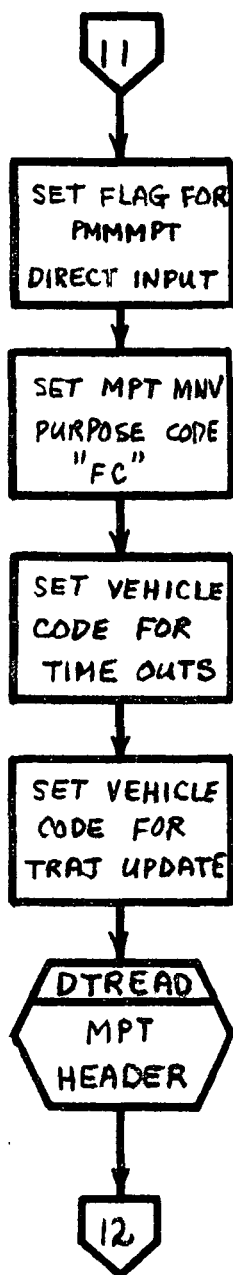


Figure 1

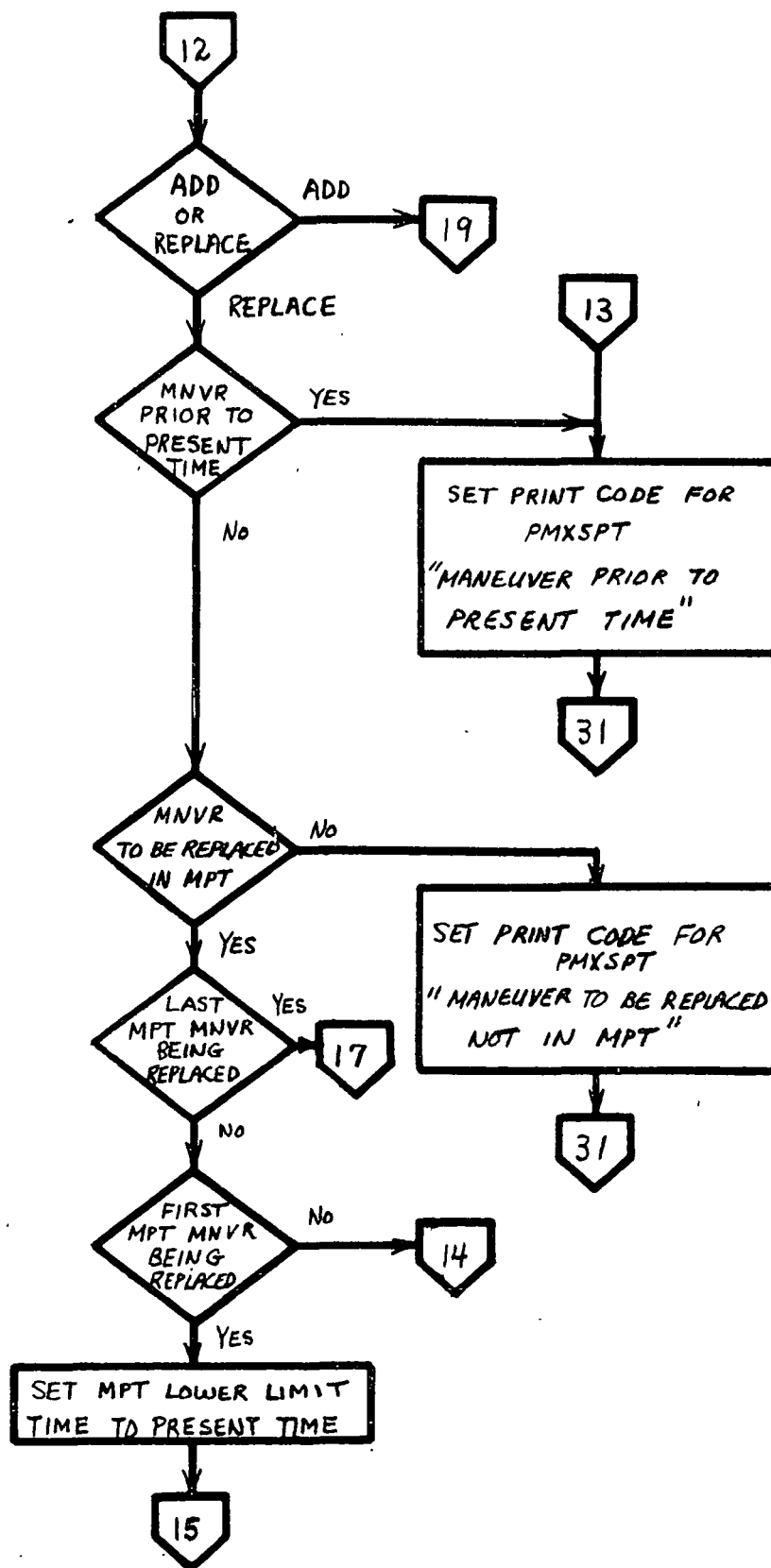


Figure 1

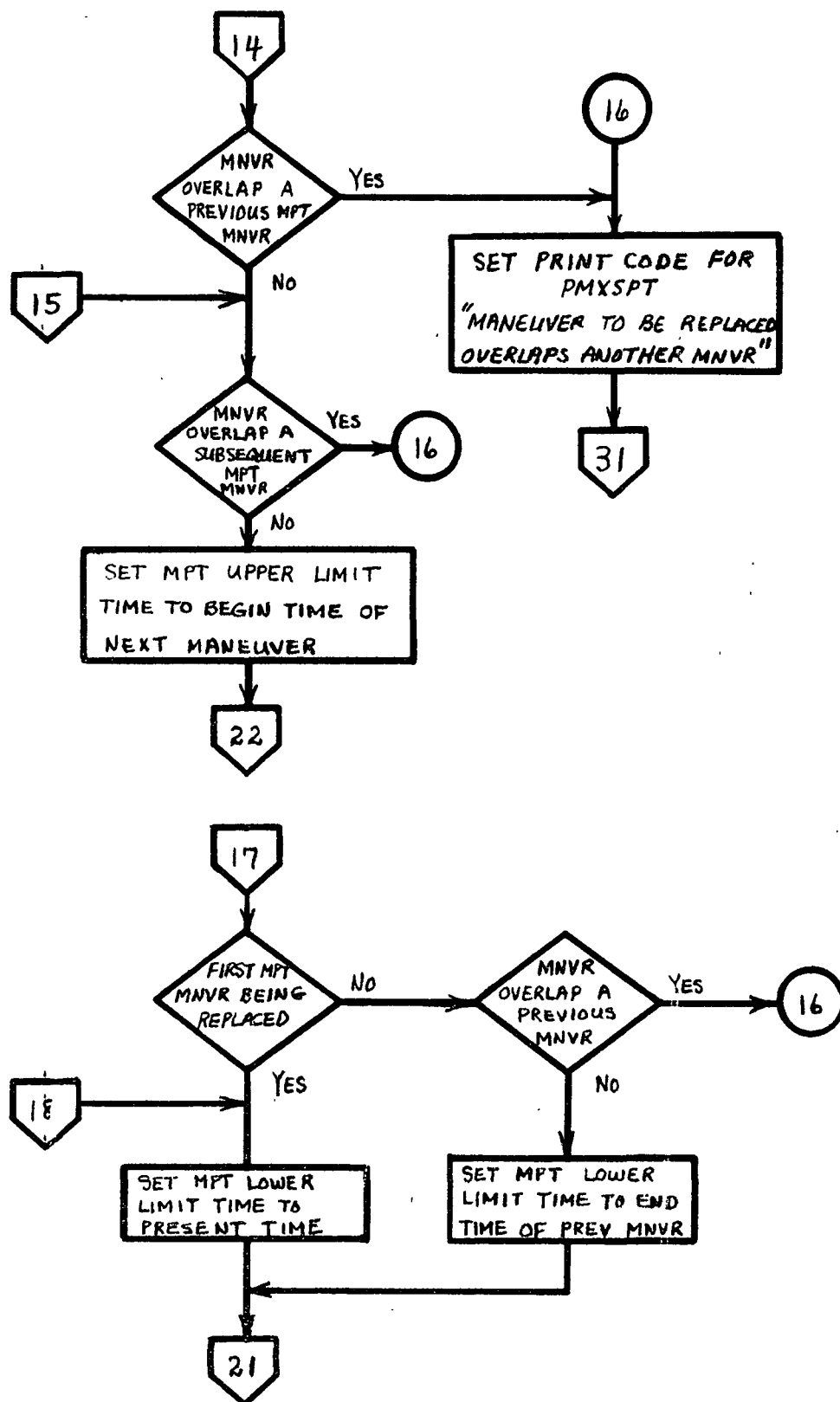


Figure 1

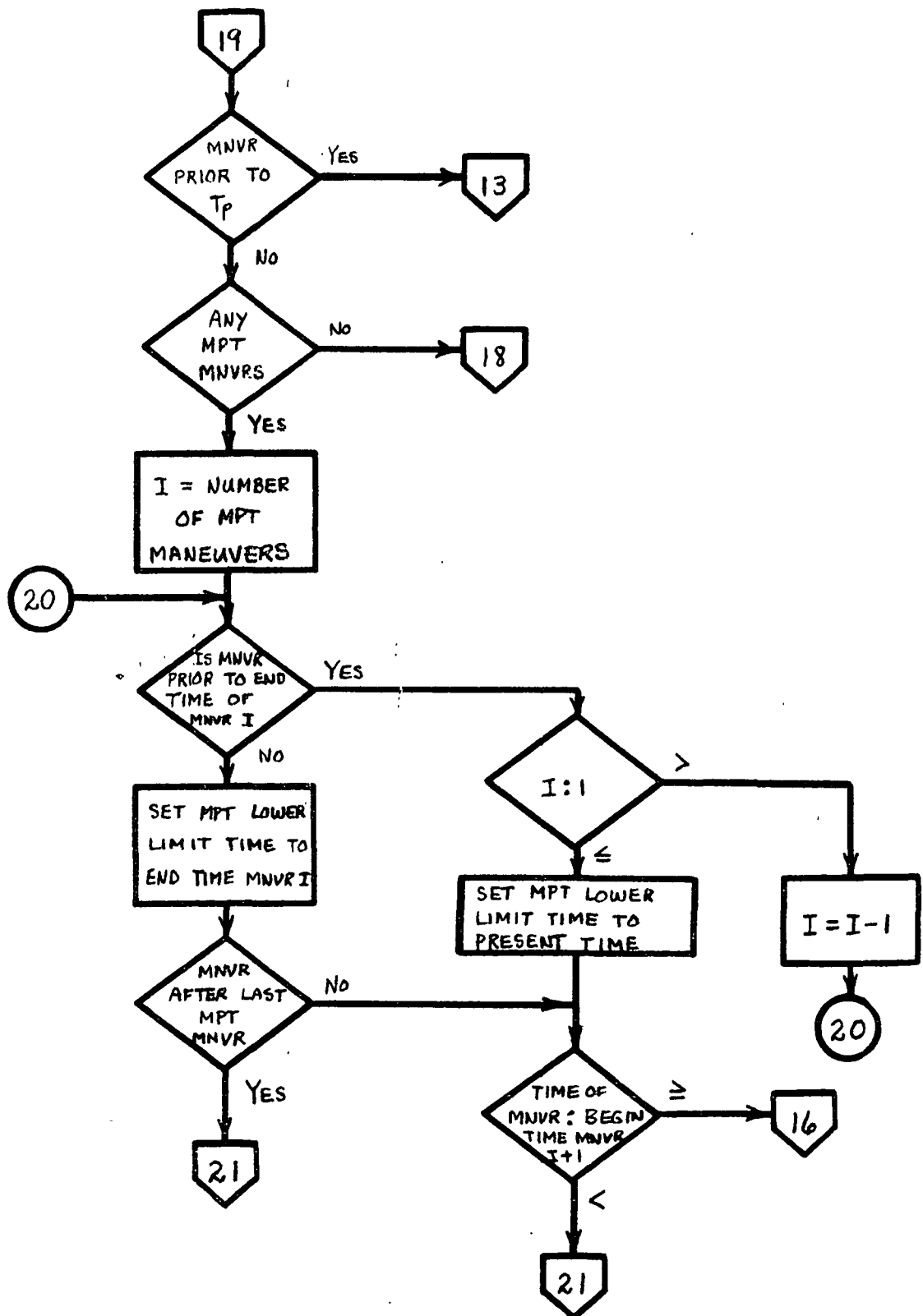


Figure 1

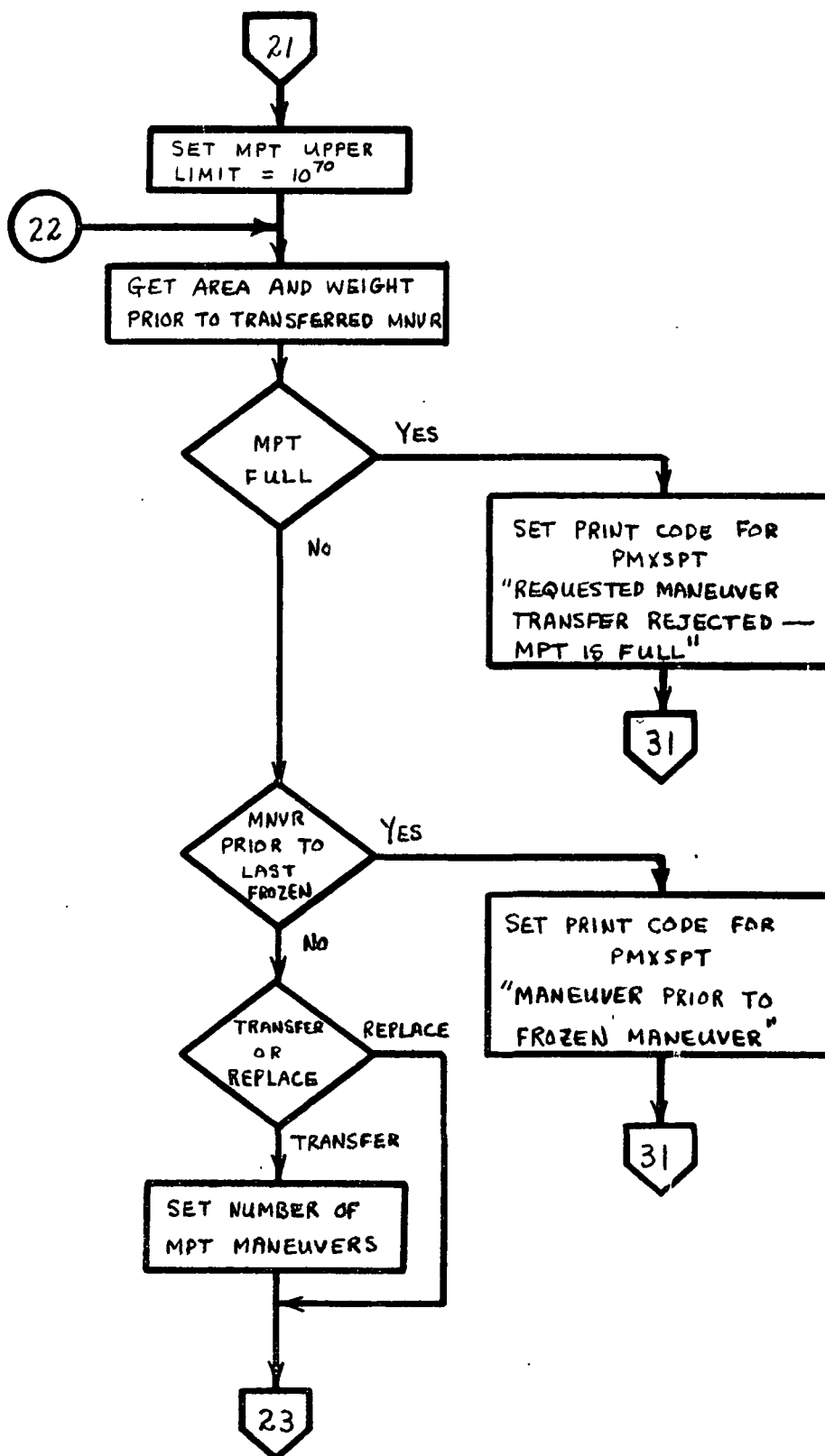


Figure 1

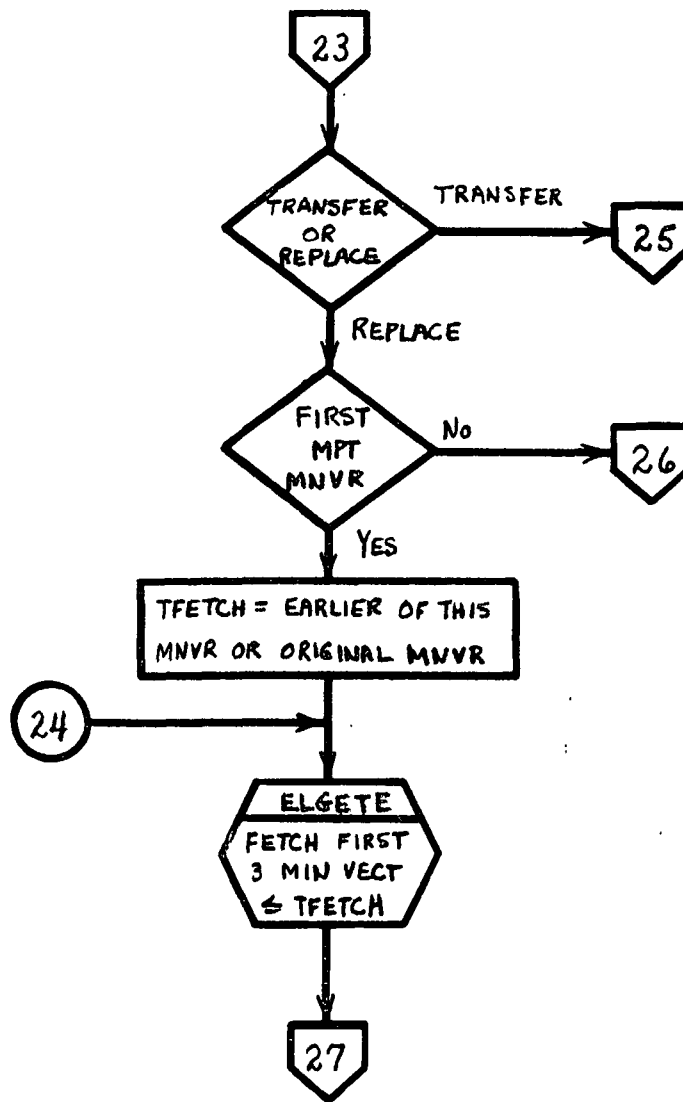


Figure 1

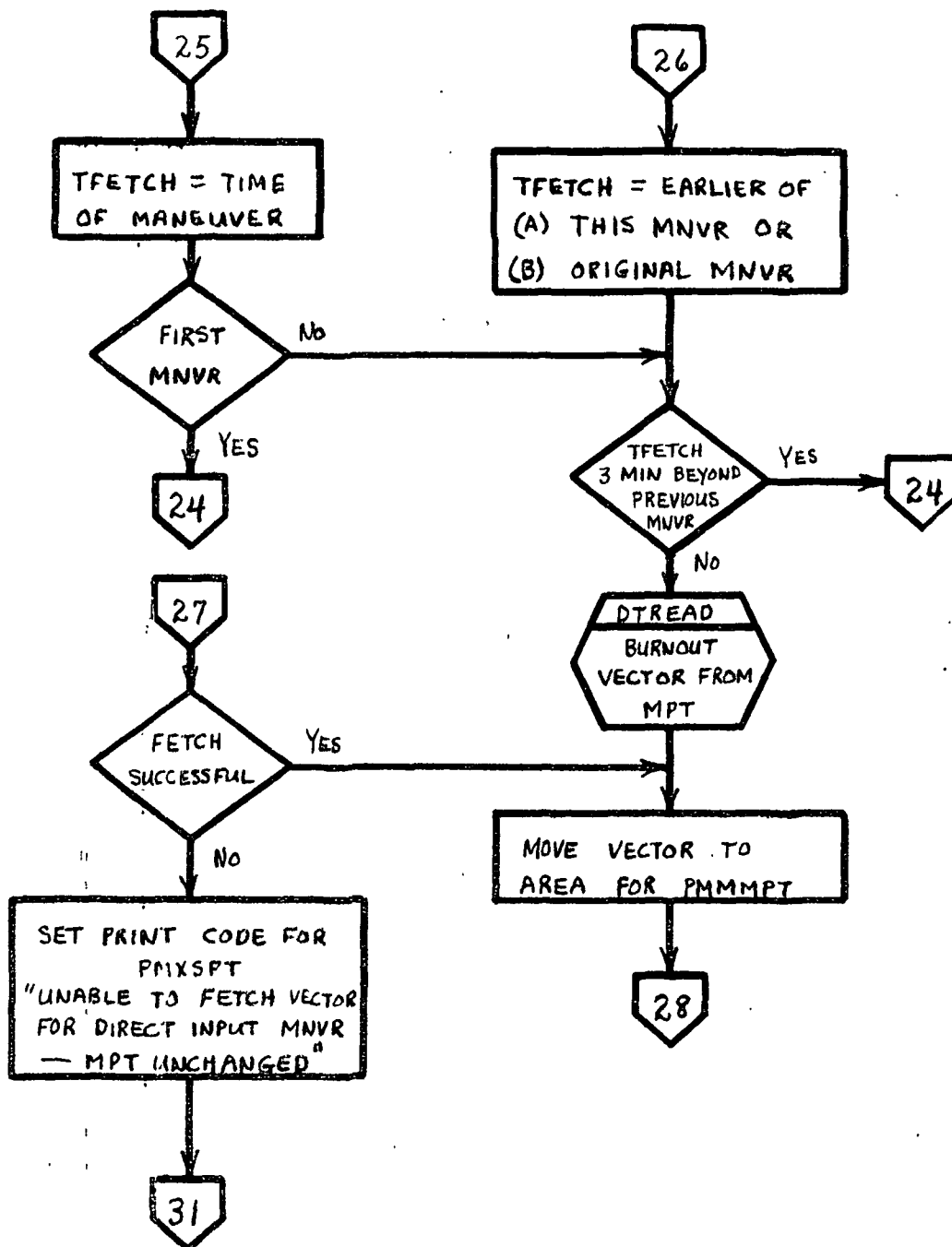


Figure 1

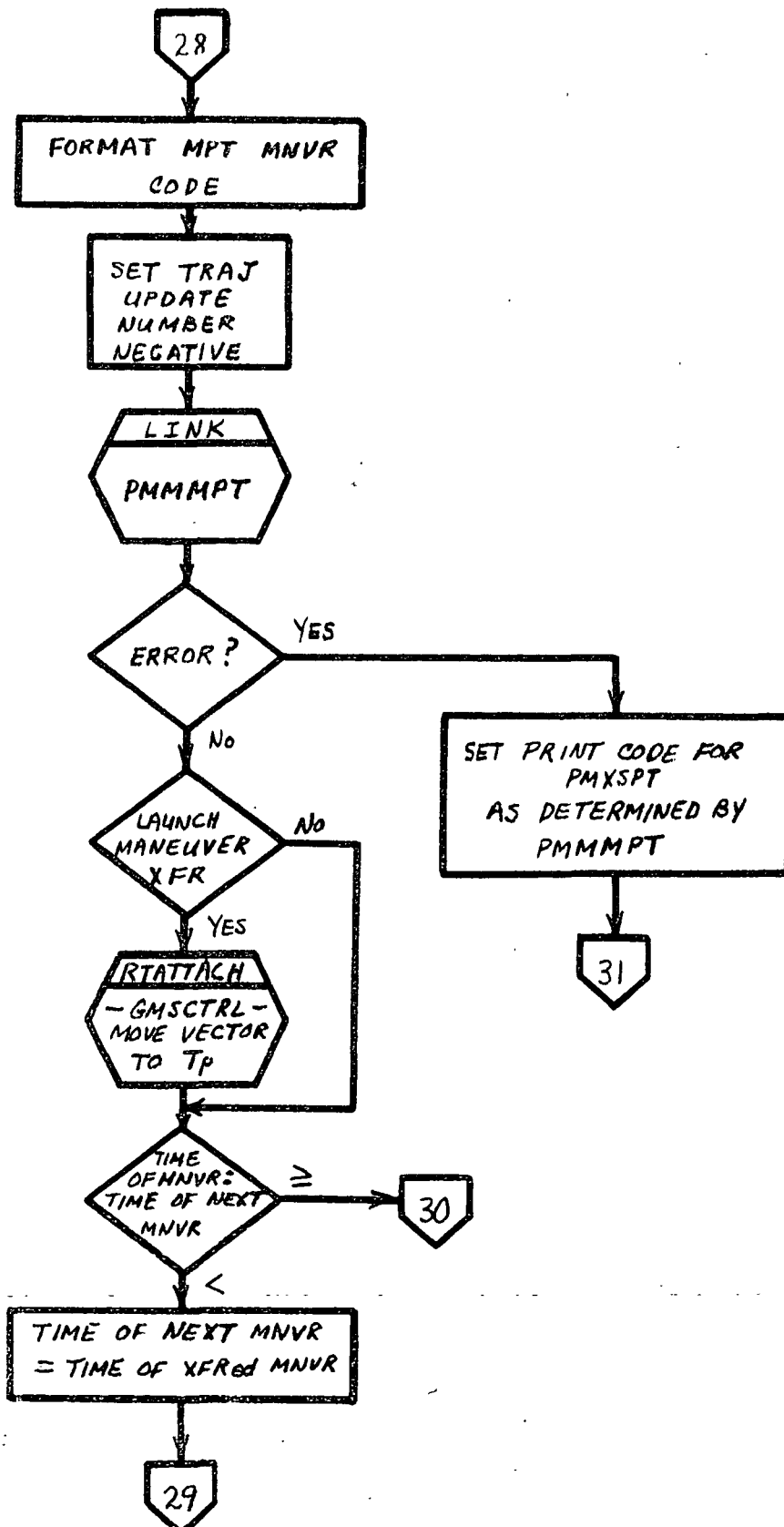


Figure 1

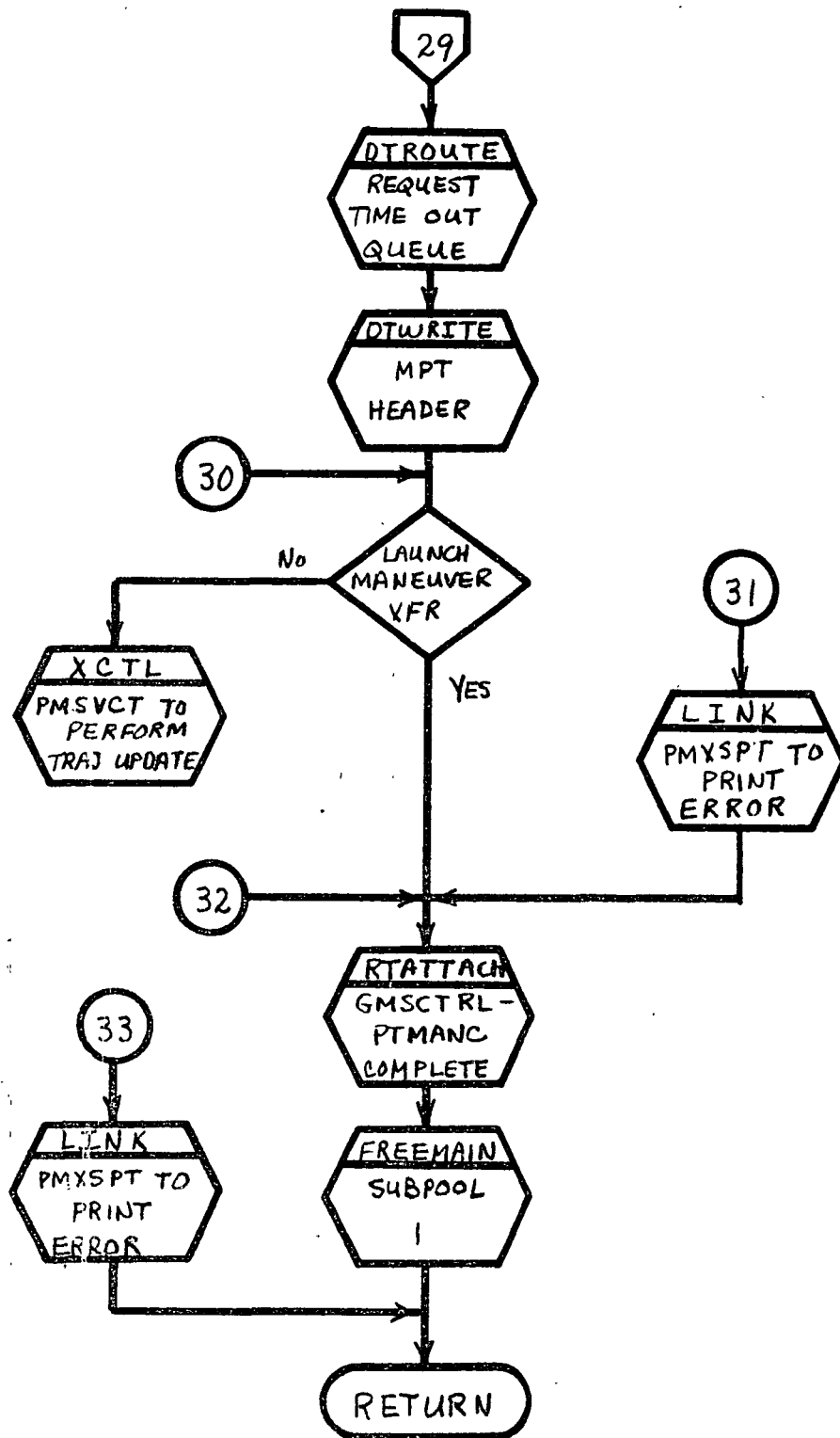


Figure 1

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMXFR
 DATE 11/8/67 (EORP)
 REV
 PAGE 1 of 50

- a. Symbol — PMMXFR
- b. Program Name — Maneuver Transfer Module
- c. Programming Group — Mission Planning
- d. Programmer — E. Kramer
- e. Language — Assembler
- f. Data Tables Used — PZMPTCSM, PZMPTLEM, PZGPMELM, PZDKIELM, PZDKIT, PZPADDIS, GZLTRA, GZGENCSN
- g. MED's Used — m01, m60, m64, m65, m66, m70, m72
- h. Writeup Completion Date — 8/30/67
- i. System Parameters — Table PMMXFR-1
- j. Parameter List — Table PMMXFR-2 through 8
- k. Subpool List — Table PMMXFR-9 through 15
- l. Flowchart — Figure PMMXFR-1

FUNCTION

The program solves the ground rules and controls the processing necessary to place the following types of maneuvers into the Mission Plan Table (MPT):

- a. Contingency Insertion
- b. Perigee Adjust
- c. General Purpose
- d. Direct Input
- e. Two-Impulse Plan
- f. DKI Plan
- g. Co-elliptic Sequence Plan

PMMXFR also solves ground rules and controls the processing necessary to confirm executed maneuvers which exist in the Mission Plan Table.

USAGE

- a. Calling Sequence — RTATTACH EP = PMMXFR, TASK = PTMANC, ID = X, PARAM = (ADDSP) where $29 \leq X \leq 36$ and ADDSP is the address of subpool 1 in which one of the MED's is located.
- b. Inputs — MED quantities
- c. Outputs and Exit — See Method
- d. Other Programs Used — ELGETE, PMMMCD, PMMMPT, PMXSPT, EMGVECST
- e. Data Restrictions and Limits — See Method
- f. Error — See Method

Approval	Approval
<i>E. Kramer</i>	<i>J. E. James</i>

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMXFR
DATE 11/8/67 (EORP)
REV
PAGE 2 of 50METHOD

The load module is divided into four main sections of logic; 1) Launch to Orbit Override, 2) Transfer of impulsive maneuvers, 3) Direct Input Maneuvers, and 4) Confirmations.

Launch to Orbit Override is used if the nominal transfer is not desired. The two options are to generate the ephemeris from a preburn or a postburn (or no burn) vector. If a preburn vector is desired, any existing maneuvers in the MPT are first deleted, and the maneuver quantities in the launch transfer table are sent to PMMMCD for processing. The trajectory update is then queued through GMSCTRL. If the postburn or "TRA" option is used, and the configuration is unchanged from launch, the vector is merely routed to GMSCTRL for the update. If, however, the configuration differs, all maneuvers are deleted before routing the vector.

Under impulsive transfers, there are three divisions: 1) GPM, 2) Two-Impulse, and 3) DKI-SPQ. For each of these divisions there is a separate initial logic after which these queues are treated the same. First of all, enough main storage is obtained to store all the pertinent data, and then the necessary data tables are read. Ground rules are then checked to see if this is a valid transfer. On a replace option, these checks include ensuring that the desired maneuver does exist, is unfrozen, and the new maneuver time is greater than the end time of the preceding maneuver and less than the start time of the following one if such maneuvers are presently in the MPT. In case of a normal transfer, all current maneuvers after the time of the incoming maneuver are deleted if they are unfrozen. If frozen, the transfer is rejected, and the MPT remains unchanged. Also a check is made to reject the transfer if the maximum number of allowable maneuvers is exceeded. For both standard transfers and replace, the time of the maneuver must be greater than present time and time of insertion. Having made these tests successfully, the maneuver inputs are set-up for module PMMMPT and control given to it. If no error code is returned by PMMMPT, checking on the next maneuver is started if there is a multiple transfer. If all maneuvers have been processed, the needed tables are written out, and a trajectory update is queued by XCTLing to PMSVCT.

For direct inputs, which also include Perigee Adjust transfers, the same sort of ground rules are checked as for impulsive maneuvers, but in addition, configuration checking is done. The input configuration change indicator and configuration code at maneuver end are tested against the current MPT's to ensure this change, along with the designated thruster, is physically possible. If not, the transfer is rejected. If all the inputs appear correct, they are routed to PMMMCD along with

a vector from the current ephemeris at an appropriate time. If PMMMCD finds no errors, it is responsible for writing out the tables. Upon regaining control, PMMXFR will XCTL to PMSVCT to queue the ephemeris update.

Confirmations are handled similar to the replace option of a direct input except the maneuver time must be prior to present time, and vectors are obtained from the anchor vector table through EMGVECST.

DTROUTE, RTQCNTRL, RTATTACH Macros

Macro — RTATTACH TASK = GTMSCTRL, EP = GMSCTRL, QNAME = GQ3CTL, QRANK = A10, ID = (4), DPRIOR = MKDGMP, PARM = (INPUT, 72)

Purpose — To initiate a trajectory update using an anchor vector in case of confirmation or contingency insertion

Macro — DTROUTE PRMVREX1, RMODE = ENABLE, RTYPE = TIMK, QNAME = GQ1CTL, TASK = GTMSCTRL, EP = GMSCTRL, DPRIOR = MKDGMP, MSGCNT = 1, QRANK = QRANK12, START = (L, TIME), ID = 283

Purpose — To initiate queue at maneuver execution time for each CSM maneuver

Macro — DTROUTE PRMVREX3, RMODE = ENABLE, RTYPE = TIMK, QNAME = GQ1CTL, TASK = GTMSCTRL, EP = GMSCTRL, DPRIOR = MKDGMP, MSGCNT = 1, QRANK = QRANK12, START = (L, TIME), ID = 795

Purpose — To initiate queue at maneuver execution time for each LM maneuver

Table PMXFR-1. System Parameters

[illegible]

*Type
H — Fixed-point half word
F — Fixed-point full word
E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

2. PMMXFR
 DATE 11/8/67 (EORP)
 REV
 PAGE 5 of 50

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

Table PMMXFR-2. Parameter List

QUEID — 29

QNAME — GQ1CTL

From — GMSCTRL

To-Task — PTMANC

Purpose — GPM Transfer

EP — PMMXFR

Byte No.	Type*	Units	Value / Description / Comments
1-4	F		Address subpool 1

Table PMMXFR-3. Parameter List

QUEID — 30

QNAME — GQ1CTL

From — GMSCTRL

To-Task — PTMANC

Purpose — Perigee Adjust Transfer

EP — PMMXFR

Byte No.	Type*	Units	Value / Description / Comments
1-4	F		Address subpool 1

*Type

H — Fixed-point half word
F — Fixed-point full word

E — Single-precision (floating-point)

D — Double-precision (floating-point)

X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

2. PMMXFR
 DATE 11/8/67 (EORP)
 REV
 PAGE 6 of 50

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

Table PMMXFR-4. Parameter List

QUEID — 31

QNAME — GQ1CTL

From — GMSCTRL

To-Task — PTMANC

Purpose — Launch to Orbit Transfer

EP — PMMXFR

Byte No.	Type*	Units	Value / Description / Comments
1-4	F		Address subpool 1

Table PMMXFR-5. Parameter List

QUEID — 32

QNAME — GQ1CTL

From — GMSCTRL

To-Task — PTMANC

Purpose — DKI plan transfer

EP — PMMXFR

Byte No.	Type*	Units	Value / Description / Comments
1-4	F		Address subpool 1

*Type
 H — Fixed-point half word
 F — Fixed-point full word

E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

Table PMMXFR-6. Parameter List2. PMMXFR
DATE 11/8/67 (EORP)
REV
PAGE 7 of 50

QUEID — 33

QNAME — GQ1CTL

From — GMSCTRL

To-Task — PTMANC

Purpose — Direct Input of a Maneuver

EP — PMMXFR

Byte No.	Type*	Units	Value / Description / Comments
1-4	F		Address subpool 1

Table PMMXFR-7. Parameter List

QUEID — 34

QNAME — GQ1CTL

From — GMSCTRL

To-Task — PTMANC

Purpose — Two-Impulse Transfer

EP — PMMXFR

Byte No.	Type*	Units	Value / Description / Comments
1-4	F		Address subpool 1

*Type
H — Fixed-point half word
F — Fixed-point full word

E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMXFR
DATE 11/8/67 (EORP)
REV
PAGE 8 of 50Table PMMXFR-8. Parameter List

QUEID — 36

QNAME — GQ1CTL

From — GMSCTRL

To-Task — PTMANC

Purpose — Maneuver Confirmation

EP — PMMXFR

Byte No.	Type*	Units	Value / Description / Comments
1-4	F		Address subpool 1

*Type
H — Fixed-point half word
F — Fixed-point full wordE — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMXFR
 DATE 11/8/67 (EORP)
 REV
 PAGE 9 of 50
Table PMMXFR-9. Subpool List

Subpool Number 1

For ID = 29 (GPM Transfer)

Symb. Ref.	Byte No.	Type*	Units	Value / Description / Comments
	1-28	F		Pointers to Items 1-7 of M65 MED.
	29-32	F		Open
	33-36	F		Table Code: 1 = PZMPTCSM 3 = PZMPTLEM
	37-40	F		Replace Code
	41-44	F		Thruster Code: 1 = CSM RCS + X (2 thrusters) 2 = CSM RCS + X (4 thrusters) 3 = CSM RCS - X (2 thrusters) 4 = CSM RCS - X (4 thrusters) 16 = SIVB Blowdown Vent 17 = LEM RCS + X (2 thrusters) 18 = LEM RCS + X (4 thrusters) 19 = LEM RCS - X (2 thrusters) 20 = LEM RCS - X (4 thrusters) 33 = CSM SPS 34 = LEM APS 35 = LEM DPS
	45-48	F		Open
	49-52	F		Attitude Code: 1 = fixed inertial 2 = manual holding body orientation invariant 4 = external ΔV (primary) 5 = external ΔV (LM, AGS)
	53-56	F		Ullage Thruster Option: 0 = two thrusters 1 = four thrusters
	57-64	D	Hr.	Δt of ullage
	65-73	D	Hr.	Δt at 10% thrust level for DPS thruster set negative if DPS short burn test is to be ignored.

 *Type
 H — Fixed-point half word
 F — Fixed-point full word

 E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

Table PMMXFR-10. Subpool List

Subpool Number 1

For ID = 30 (Perigee Adjust Transfer)

Symb. Ref.	Byte No.	Type*	Units	Value / Description / Comments
	1-20	F		Pointers to Items 1-5 of M64 MED
	21-24	F		Open
	25-28	F		Table Code: 1 = PZMPTCSM
	29-32	F		Replace Code
	23-36	F		Perigee Adjust Maneuver number
	37-40	F		Open
	41-48	D	hr.	Δt of ullage
	49-52	F		Ullage thruster Option: 0 = two thrusters 1 = four thrusters

Table PMMXFR-11. Subpool List

Subpool Number 1

For ID = 31 (Launch to Orbit Transfer)

Symb. Ref.	Byte No.	Type*	Units	Value / Description / Comments
	1-4	F		Pointer to item 1 of M01 MED
	5-8	F		Open
	9-12	F		Transfer code: 0 = Preburn 1 = Postburn or no burn

* Type

H — Fixed-point half word
F — Fixed-point full wordE — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMXFR
 DATE 11/8/67 (EORP)
 REV
 PAGE 11 of 50
Table PMMXFR-12. Subpool List

Subpool Number 1

For ID = 32 (DKI or SPQ Transfer)

Symb. Ref.	Byte No.	Type*	Units	Value / Description / Comments
	1-32	F		Pointers to items 1-8 of M70 MED
	33-36	F		Plan Number to be transferred - if zero means SPQ transfer
	37-40	F		Open
	41-48	D	hr.	Delete time - if zero, no delete
	49-52	F		Thruster Code
	53-56	F		Open
	57-60	F		Attitude Mode Code
	61-64	F		Ullage Thruster Option
	65-72	D	hr.	Δt ullage
	73-80	D	hr.	Δt 107 DPS thrust
	81-100	F		Pointers to Subparameters of item 3
	101-104	F		Open
	105-136			Same as bytes 49-80; for manv. #2
	137-156	F		Pointers to Subparameters of item 4
	157-160	F		Open
	161-192			Same as bytes 49-80; for manv. #3
	193-212	F		Pointers to Subparameters of item 5
	213-216	F		Open
	217-248			Same as bytes 49-80; for manv. #4
	249-268	F		Pointers to Subparameters of item 6
	269-272	F		Open
	273-304			Same as bytes 49-80; for manv. #5
	305-324	F		Pointers to Subparameters of item #7
	325-328	F		Open
	329-360			Same as bytes 49-80; for manv. #6

*Type
 H - Fixed-point half word
 F - Fixed-point full word

E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMXFR
 DATE 11/8/67 (EORP)
 REV
 PAGE 12 of 50

Table PMMXFR-13. Subpool List

Subpool Number 1

For ID = 33 (Direct Input)

Symb. Ref.	Byte No.	Type*	Units	Value / Description / Comments
	1-72	F		Pointers to items 1-18 of M66 MED
	73-76	F		Table code: 1 = PZMPTCSM 3 = PZMPTLEM
	77-80	F		Replace code
	81-84	F		Thruster code (See ID = 29)
	85-88	F		Attitude mode code (See ID = 29)
Burn Parameter = 1	89-96	D	Er. /hr.	ΔV (if neg., use Δt)
	97-100	F		ΔV Indicator; 0 = MAG, 1 = ΔV_c
	101-104	F		Open
	105-112	D	hr.	Δt
Burn Parameter = 2	89-96	D	Er. /hr.	V_x components of external ΔV vector
	97-104	D	$\downarrow$	V_y in the display and command
	105-112	D	$\downarrow$	V_z system
Burn Parameter = 3	89-96	D	Er. /hr.	V_{gx} components of ΔV vector
	97-104	D	$\downarrow$	V_{gy} in the IMLI coordinate
	105-112	D	$\downarrow$	V_{gz} system
Burn Parameter = 4	89-96	D	Er. /hr.	V_F components of ΔV vector
	97-104	D	$\downarrow$	V_S in the LVLH coordinate
	105-112	D	$\downarrow$	V_D system
	113	B		Burn parameter
	114-124	F		Pointers to subparameter of item 5 of MED
	125-128	F		Coordinate Indicator; 0 = LVLH, 1 = IMU
	129-136	D	rad.	Pitch

 *Type
 H - Fixed-point half word
 F - Fixed-point full word

 E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMMXFR
 DATE 11/8/67 (EORP)
 REV
 PAGE 13 of 50
Table PMMXFR-13. Subpool List (Cont'd)

Subpool Number 1

For ID = 33 (Direct Input)

Symb Ref.	Byte No.	Type*	Units	Value / Description / Comments
	137-144	D	rad.	Yaw
	145-152	D	rad.	Roll
	153-160	D	hr.	Δt of ullage
	161-164	F		Ullage thruster option; 0 = 2, 1 = 4
	165-168	F		Heads Up/Down Indicator; 0 = Down, 1 = Up
	169-176	D	hr.	Δt 10% Thrust level for DPS thruster
	177-180	F		Refsmat Indicator
	181-184	F		Open
	185-188	F		Configuration Change Indicator 0 = No Change 1 = Undocking 2 = Docking 3 = Low Ascent/Descent Separation
	189-192	F		Configuration at End of Maneuver 0 = CSM 1 = LM 2 = CSM/LM 3 = CSM/S-IVB 4 = LM/S-IVB 5 = CSM/LM/S-IVB
	193-200	D	rad.	Docking Angle

 *Type
 H - Fixed-point half word
 F - Fixed-point full word

 E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMXFR
DATE 11/8/67 (EORP)
REV
PAGE 14 of 50

For ID = 34 (Two-Impulse Transfer)

Table PMMXFR-14. Subpool List

Subpool Number 1

Symb. Ref.	Byte No.	Type*	Units	Value / Description / Comments
	1-20	F		Pointers to Items 1-5 of M72 MED
	21-24	F		Open
	25-28	F		Table Code: 1 = PZMPTCSM 2 = PZMPTLEM
	29-32	F		Plan number
	33-40	D	hr.	Delete Time: of zero, no delete
	41-44	F		Thruster Code
	45-48			Open
	49-52			Attitude Mode Code
	52-56			Ullage Thruster Option
	57-64			Δt ullage
	65-72			Δt 10% DPS thrust
	73-92			Pointer to Subparameters of Item 4
	93-96			Open
	97-128			Same as bytes 41-72; manv. #2

See ID = 29
for description* Type
H — Fixed-point half word
F — Fixed-point full wordE — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMXFR
DATE 11/8/67 (EORP)
REV
PAGE 15 of 50

For ID = 36 (Maneuver Configuration)

Table PMMXFR-15. Task Subpool List

Subpool Number 1

Symb. Ref.	Byte No.	Type*	Units	Value / Description / Comments	
Burn Parameter = 1	1-76	F		Pointers to Items 1-18 of M60 MED	
	77-80	F		Open	
	81-84	F		Table Code	
	85-88	F		Maneuver Number to be Confirmed	
	89-92	F		Thrust Code (See ID = 29)	
	93-96	F		Attitude Mode Code (See ID = 29)	
	97-104	D	Er. /hr.	ΔV	(if negative, use Δt)
	105-108	F		ΔV Indicator: 0 = MAG 1 = ΔV_c	
	109-112	F		Open	
	113-120	D	hr.	Δt	
Burn Parameter = 2	97-104	D	Er. /hr.	V_x	components of external ΔV vector in the display and command system
	105-112	↓	↓	V_y	
	113-120	↓	↓	V_z	
Burn Parameter = 3	97-104	D	Er. /hr.	V_{gx}	components of ΔV vector in the IMU coordinate system
	105-112	↓	↓	V_{gy}	
	113-120	↓	↓	V_{gz}	
	97-104	D	Er. /hr.	V_F	components of ΔV vector in the LVLH coordinate system
	105-112	↓	↓	V_S	
	113-120	↓	↓	V_P	
	97-104	D	Er. /hr.	$\Delta V_x''$	maneuver residuals in the RCS control axes system
	105-112	↓	↓	$\Delta V_y''$	
	113-120	↓	↓	$\Delta V_z''$	
	121	B		Burn Parameters	
	122-132	F		Pointers to subparameters of Item 5 of MED	

*Type
H - Fixed-point half word
F - Fixed-point full wordE - Single-precision (floating-point)
D - Double-precision (floating-point)
X - Hexadecimal



Table PMMXFR-15. Task Subpool List (Cont'd.)

Subpool Number 1

Symb. Ref.	Byte No.	Type*	Units	Value / Description / Comments
	133-136	F		Coordinate Indicator: 0 = LVLH, 1 = IMU
	137-144	D	rad.	Pitch
	145-152	D		Yaw
	153-160	D		Roll
	161-168	D	hr.	Δt of ullage
	169-172	F		Ullage thruster option 0 = 2, 1 = 4
	173-176	F		Heads Up/Down Indicator 0 = Down, 1 = Up
	177-184	D	hr.	Δt 10% Thrust Level for DPS thruster
	185-188	F		Refsmat Indicator
	189-192	F		Open
	193-196	F		Configuration Change Indicator (see ID = 33)
	197-200	F		Configuration End of Maneuver (see ID = 33)
	201-208	D	rad.	Docking Angle
	209-212	F		Trim Angle Indicator
				0 = Compute Angles
				2 = Use System Parameters

*Type

H - Fixed-point half word
F - Fixed-point full wordE - Single-precision (floating-point)
D - Double-precision (floating-point)
X - Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMXFR

DATE 11/8/67 (EORP)

REV

PAGE 17 of 50

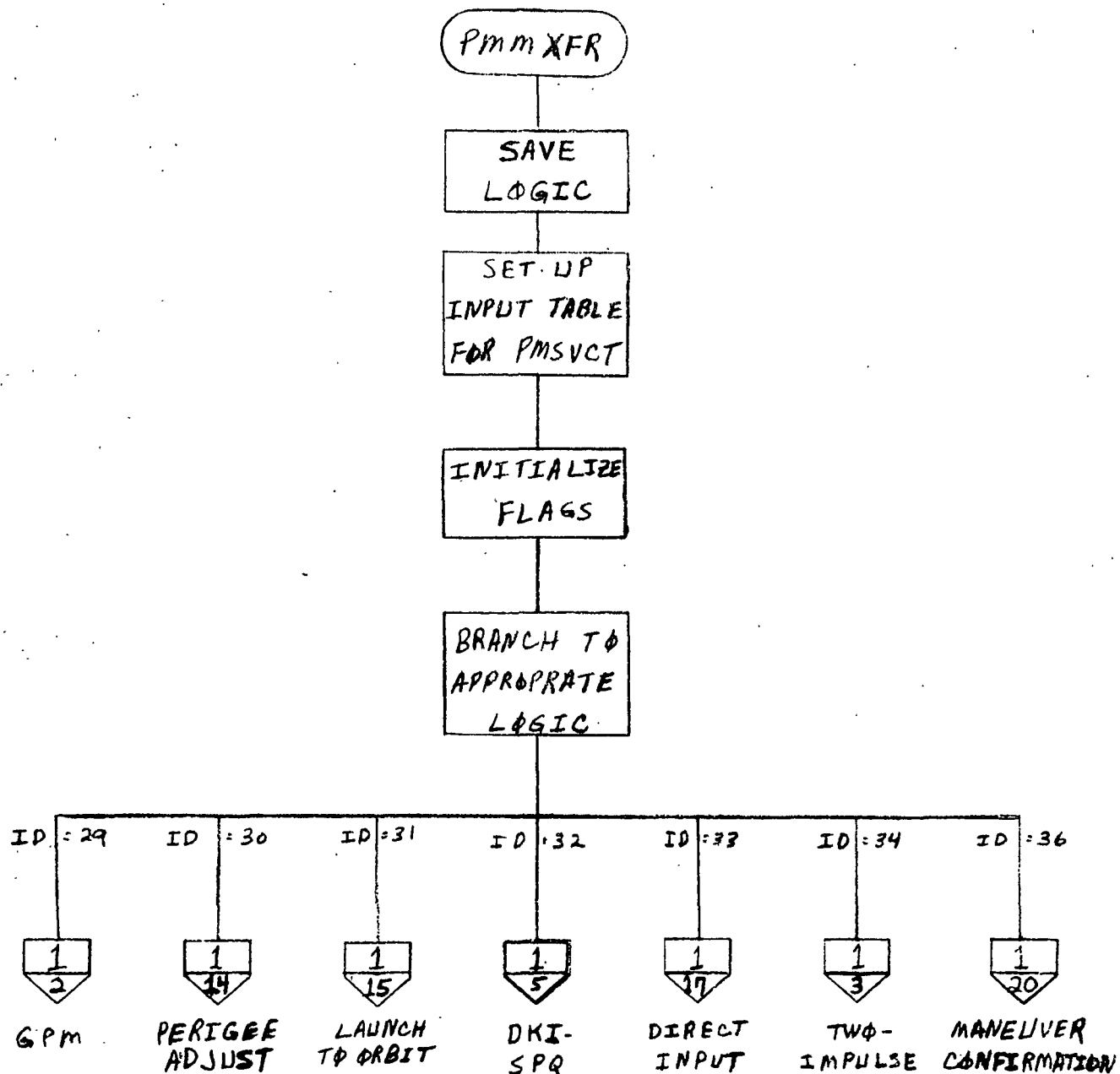


Figure PMMXFR-1. Flowchart

(Sheet 1 of 34)

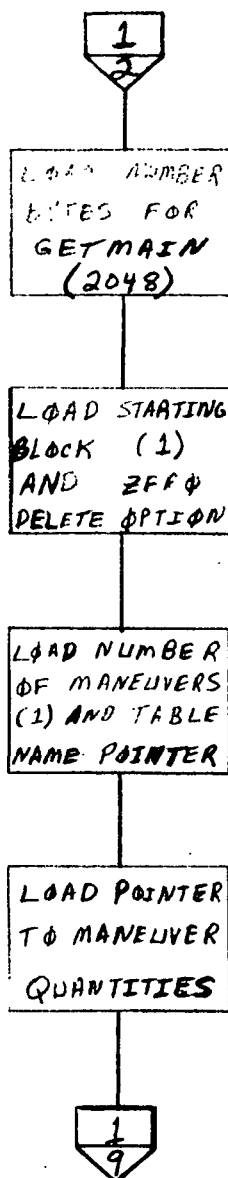


Figure PMMXFR-1. Flowchart

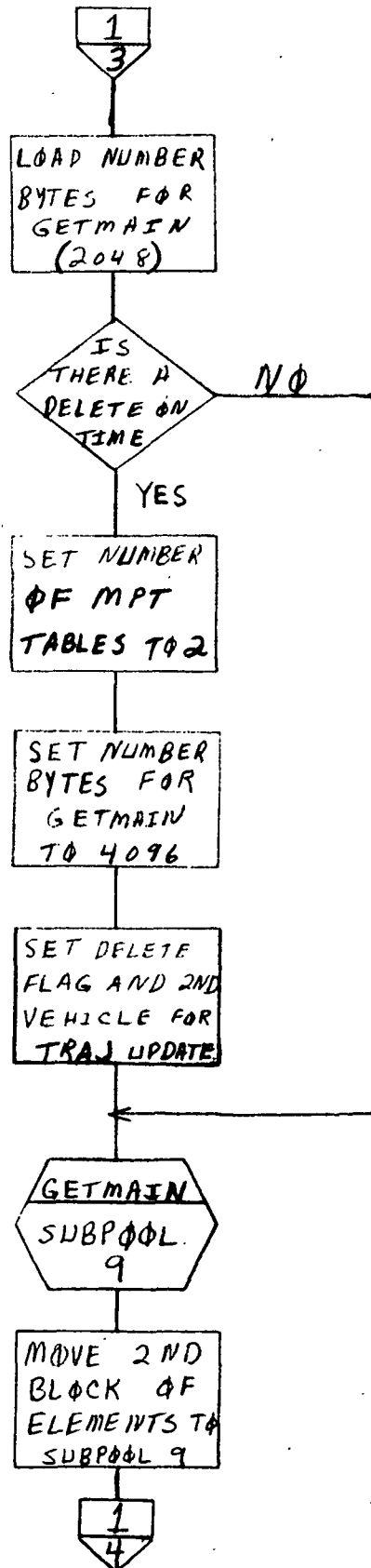


Figure PMMXFR-1. Flowchart

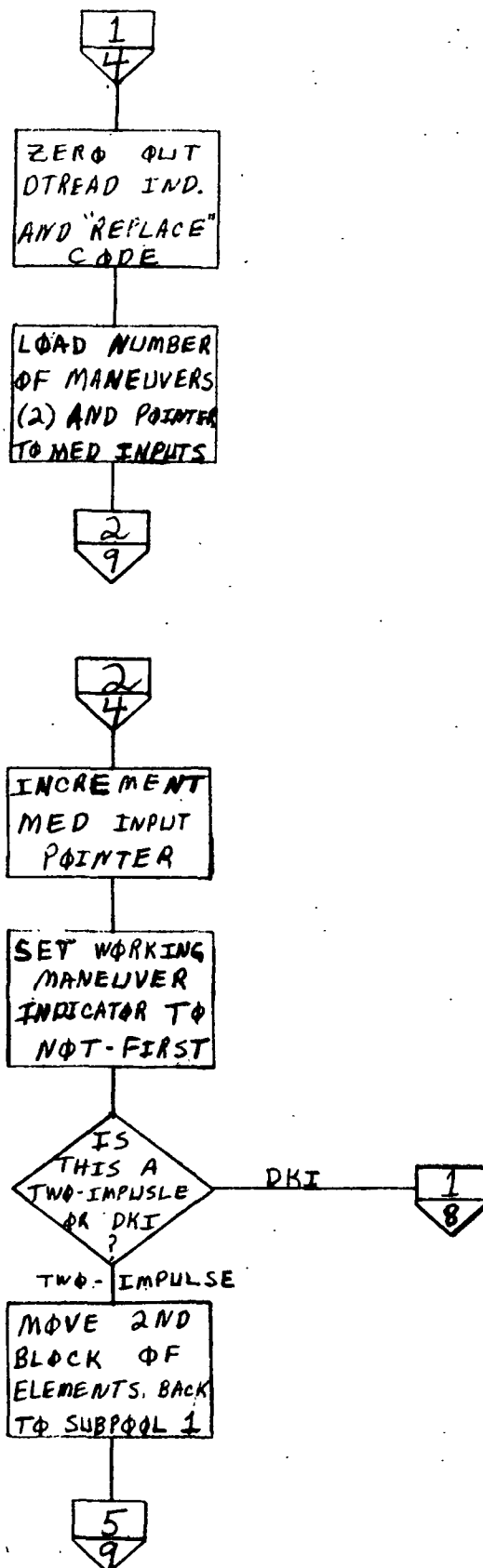


Figure PMMXFR-1. Flowchart

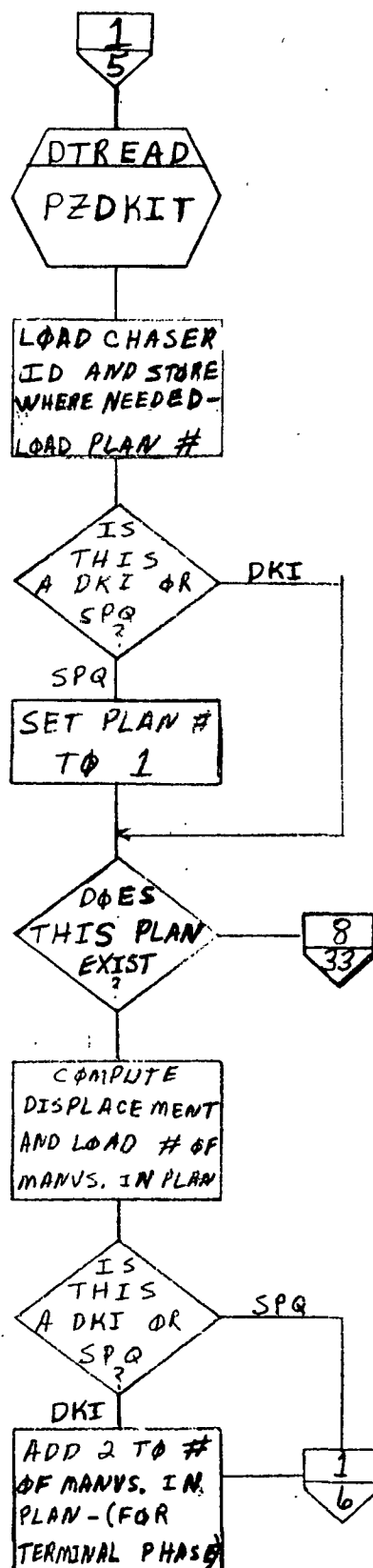


Figure PMMXFR-1: Flowchart

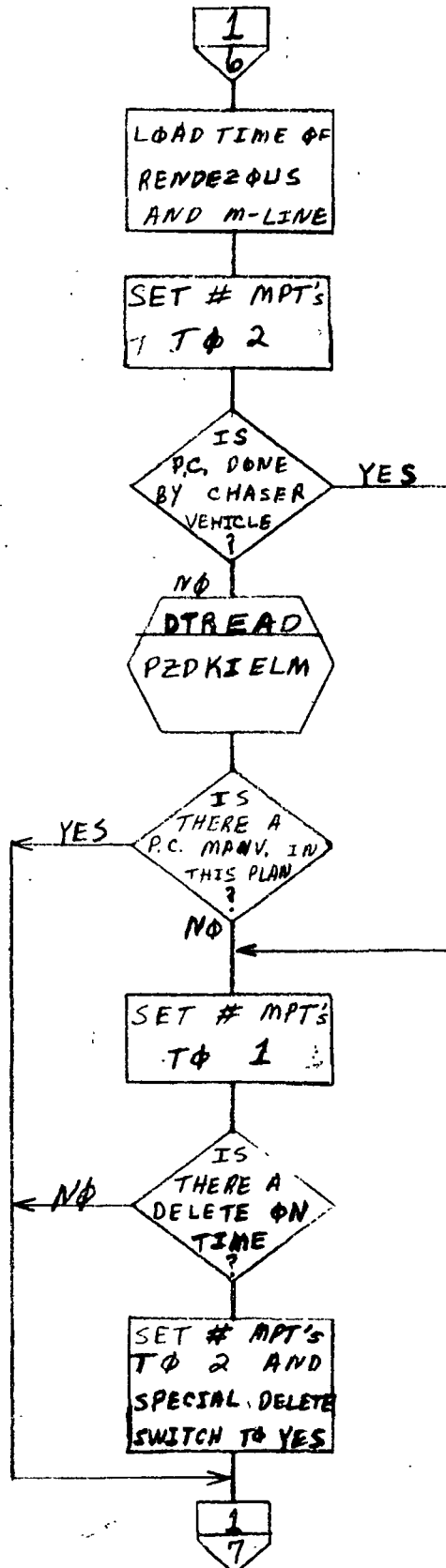


Figure PMMXFR-1. Flowchart

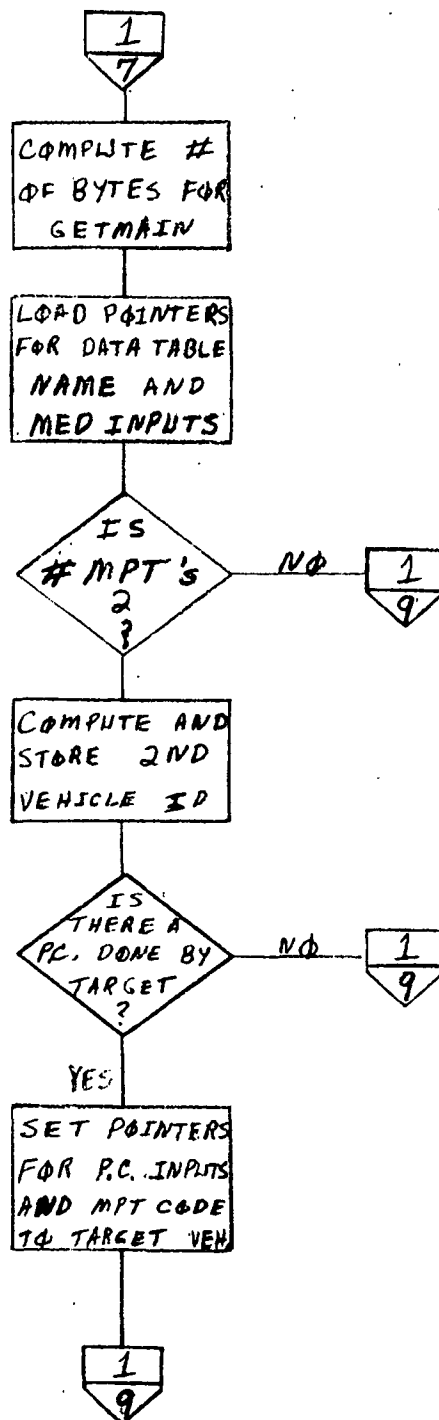


Figure PMMXFR-1. Flowchart

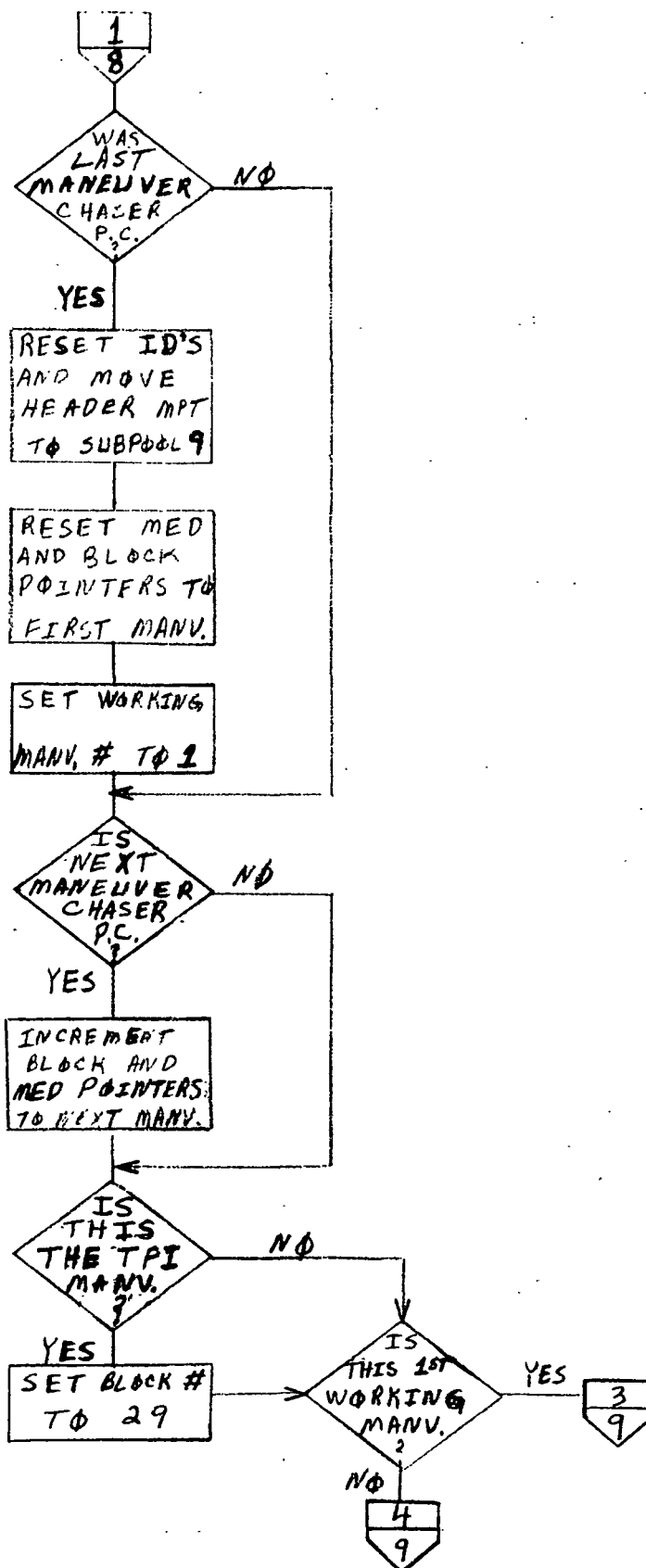


Figure PMMXFR-1. Flowchart

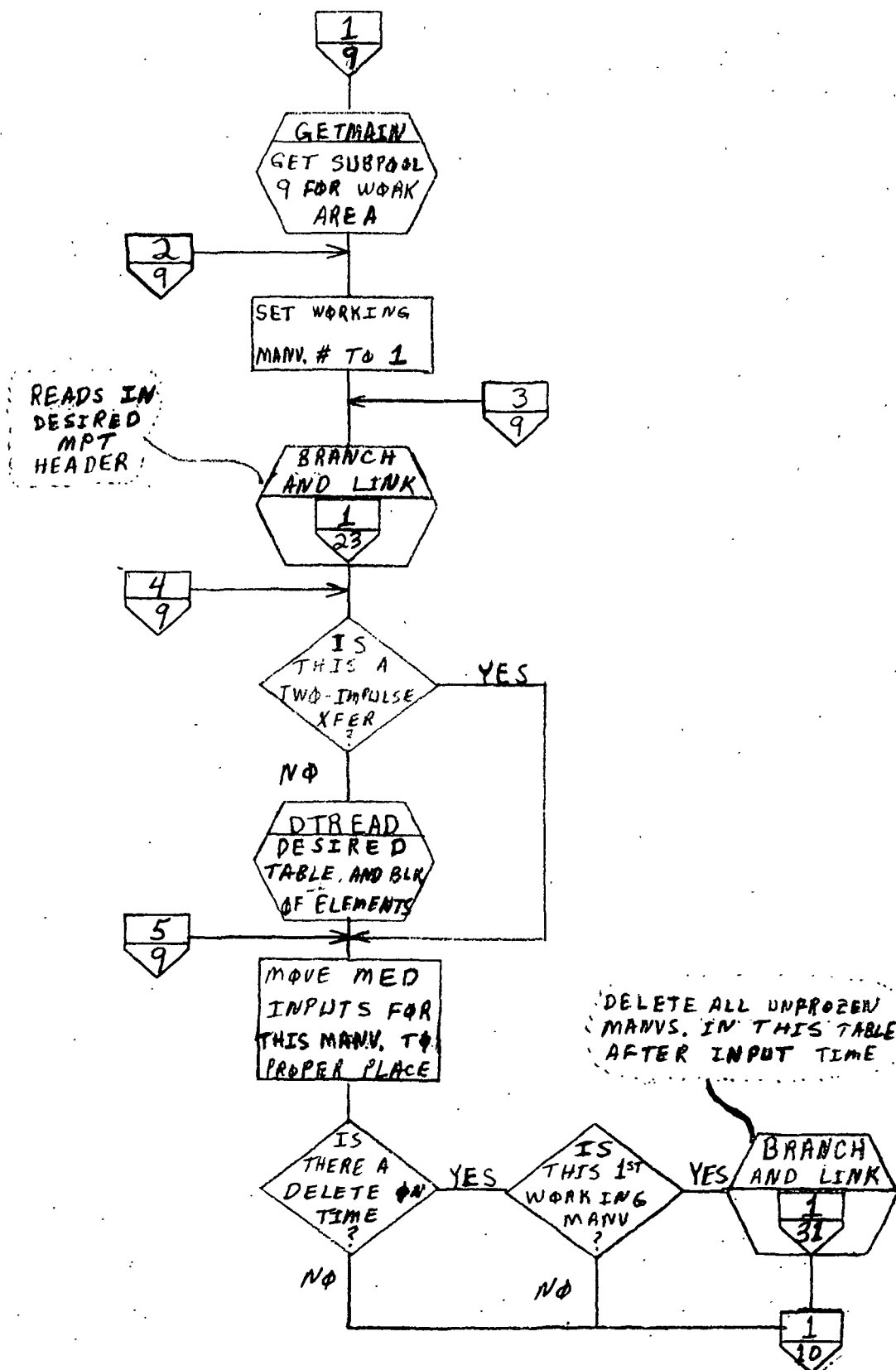


Figure PMMXFR-1. Flowchart

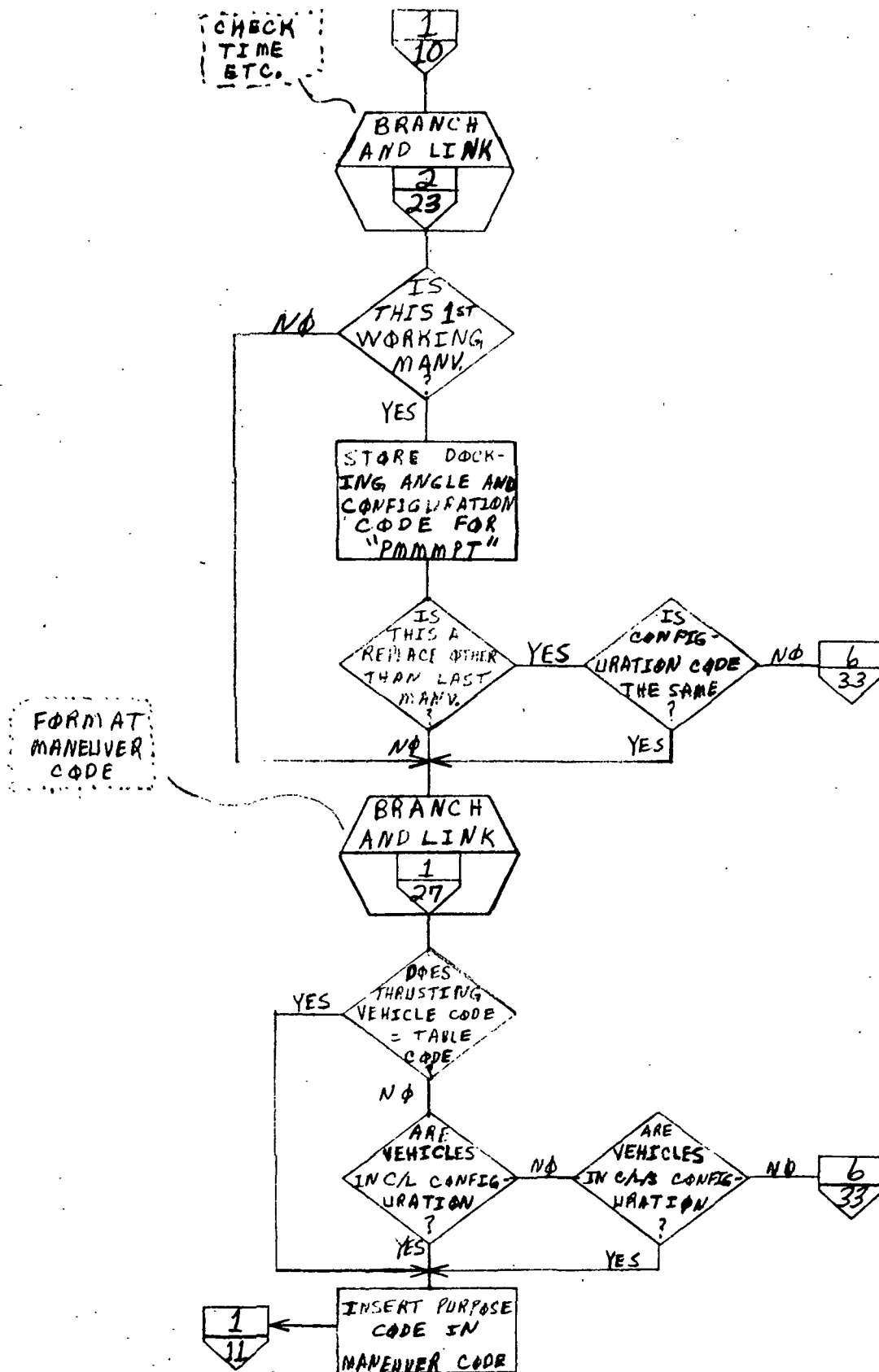


Figure PMMXFR-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMXFR

DATE 11/8/67

(EORP)

REV

PAGE 27 of 50

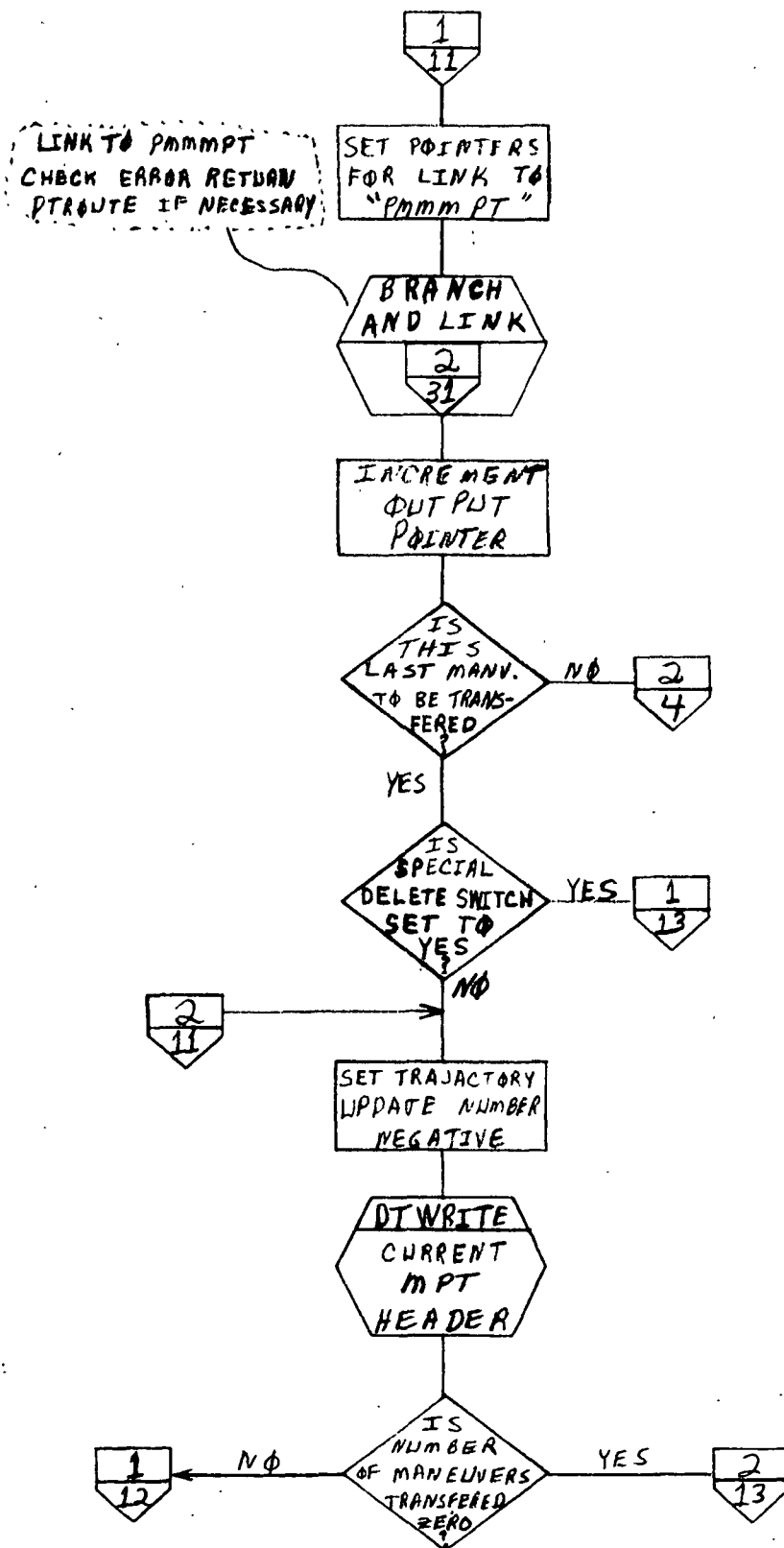


Figure PMMXFR-1. Flowchart

(Sheet 11 of 34)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMXFR

DATE 11/8/67

REV

PAGE 28 of 50

(EORP)

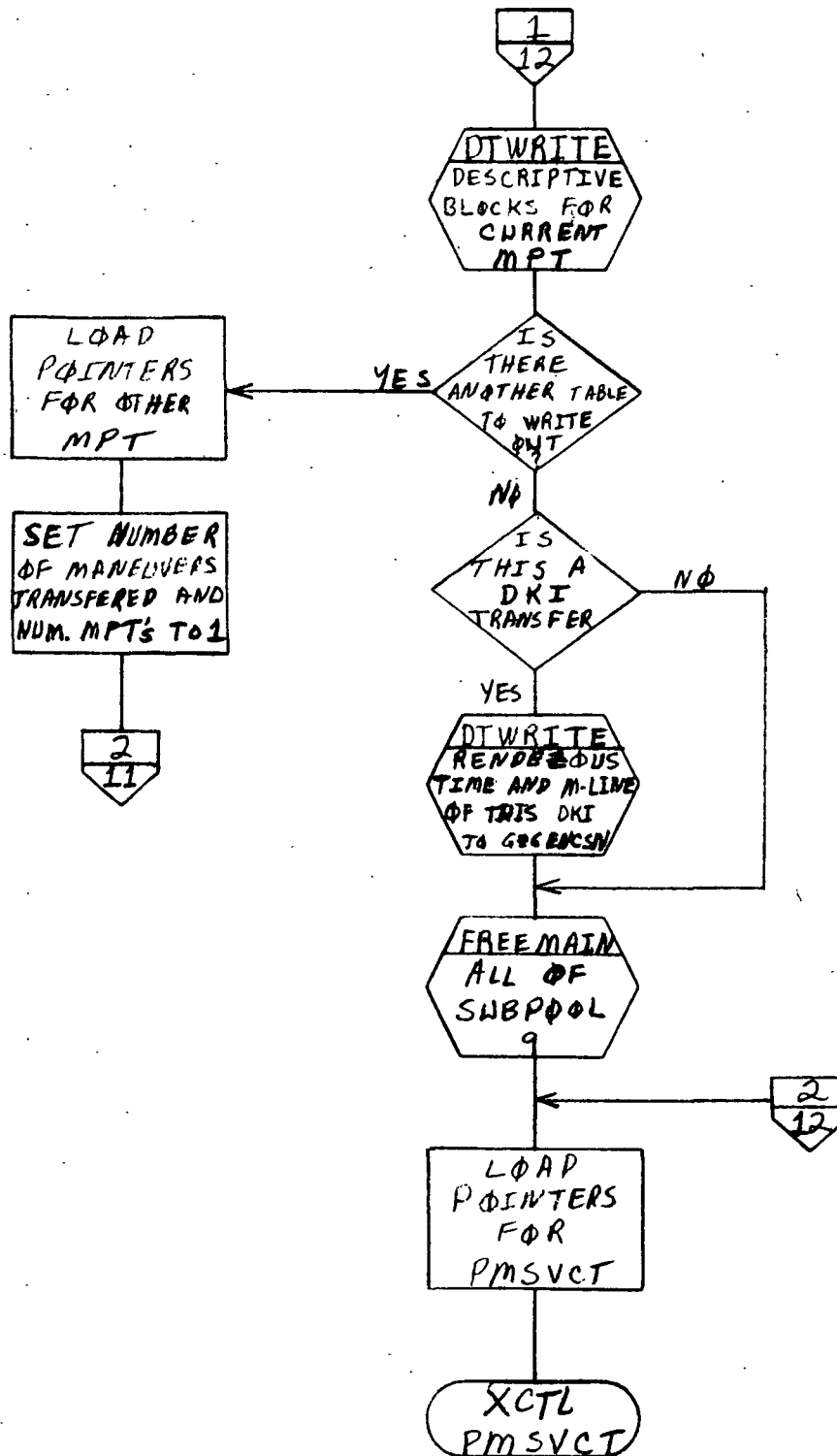


Figure PMMXFR-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMXFR

DATE 11/8/67 (EORP)

REV

PAGE 29 of 50

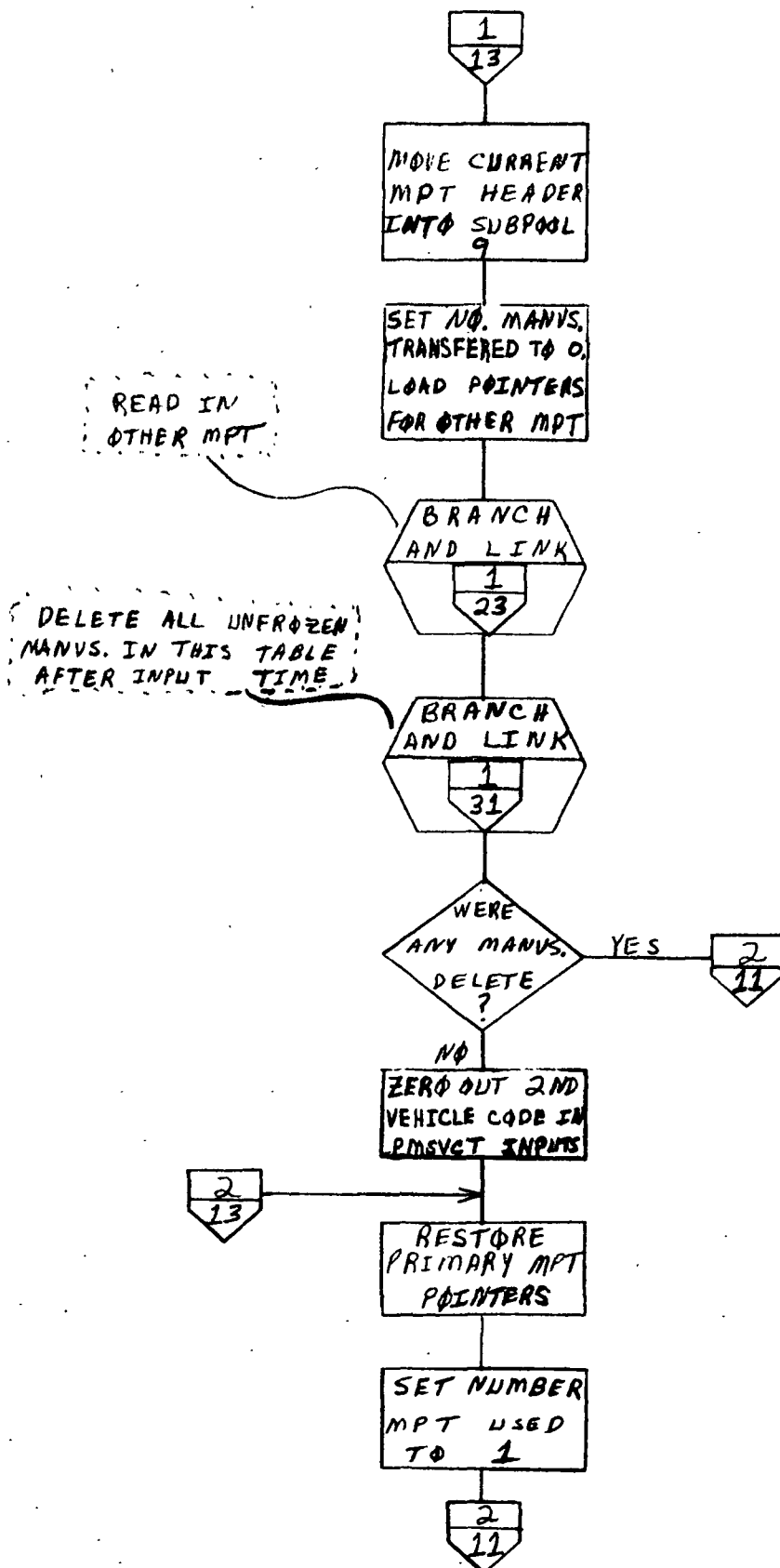


Figure PMMXFR-1. Flowchart

(Sheet 13 of 34)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

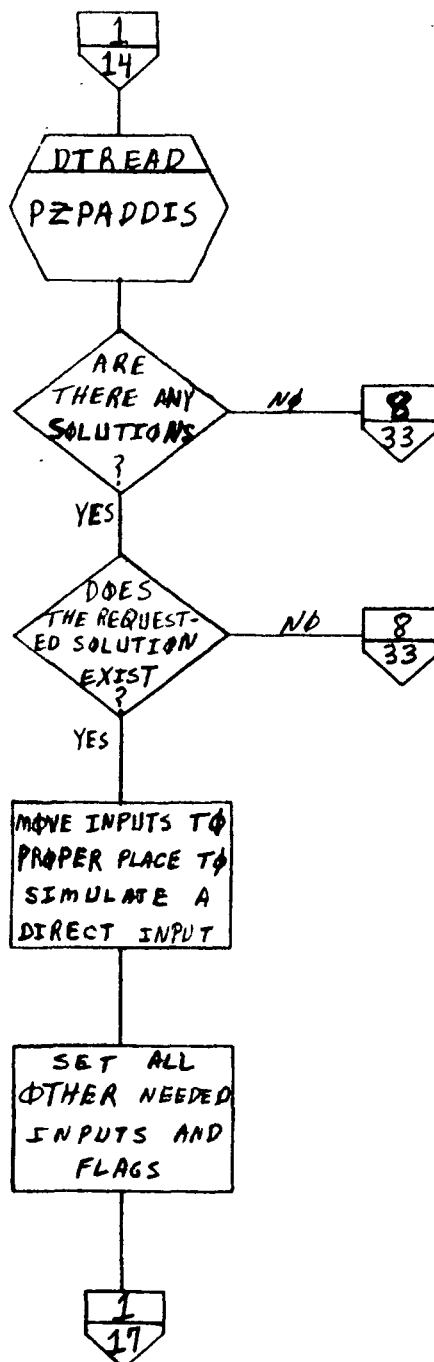
2. PMMXFR
DATE 11/8/67 (EORP)
REV
PAGE 30 of 50

Figure PMMXFR-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMXFR
DATE 11/8/67 (EORP)
REV
PAGE 31 of 50

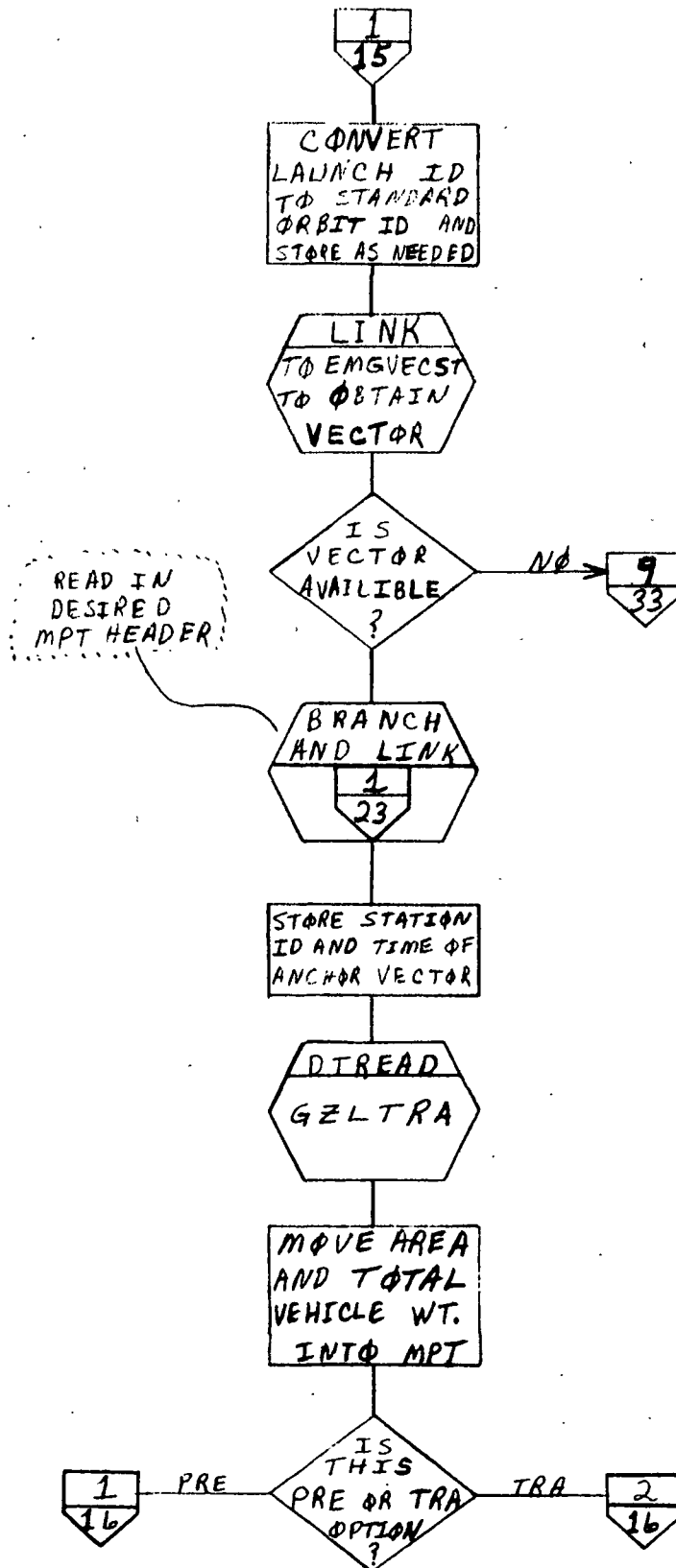


Figure PMMXFR-1. Flowchart

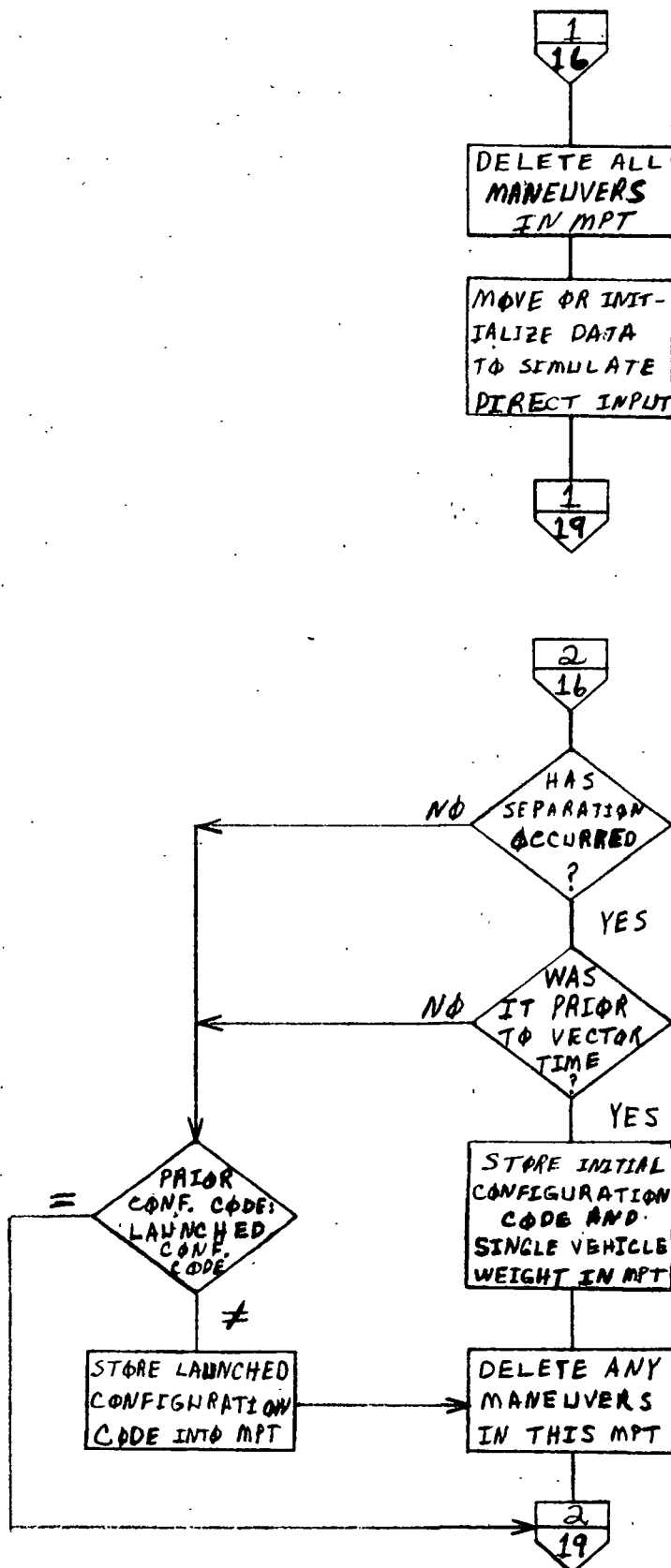


Figure PMMXFR-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMXFR

DATE 11/8/67 (EORP)

REV

PAGE 33 of 50

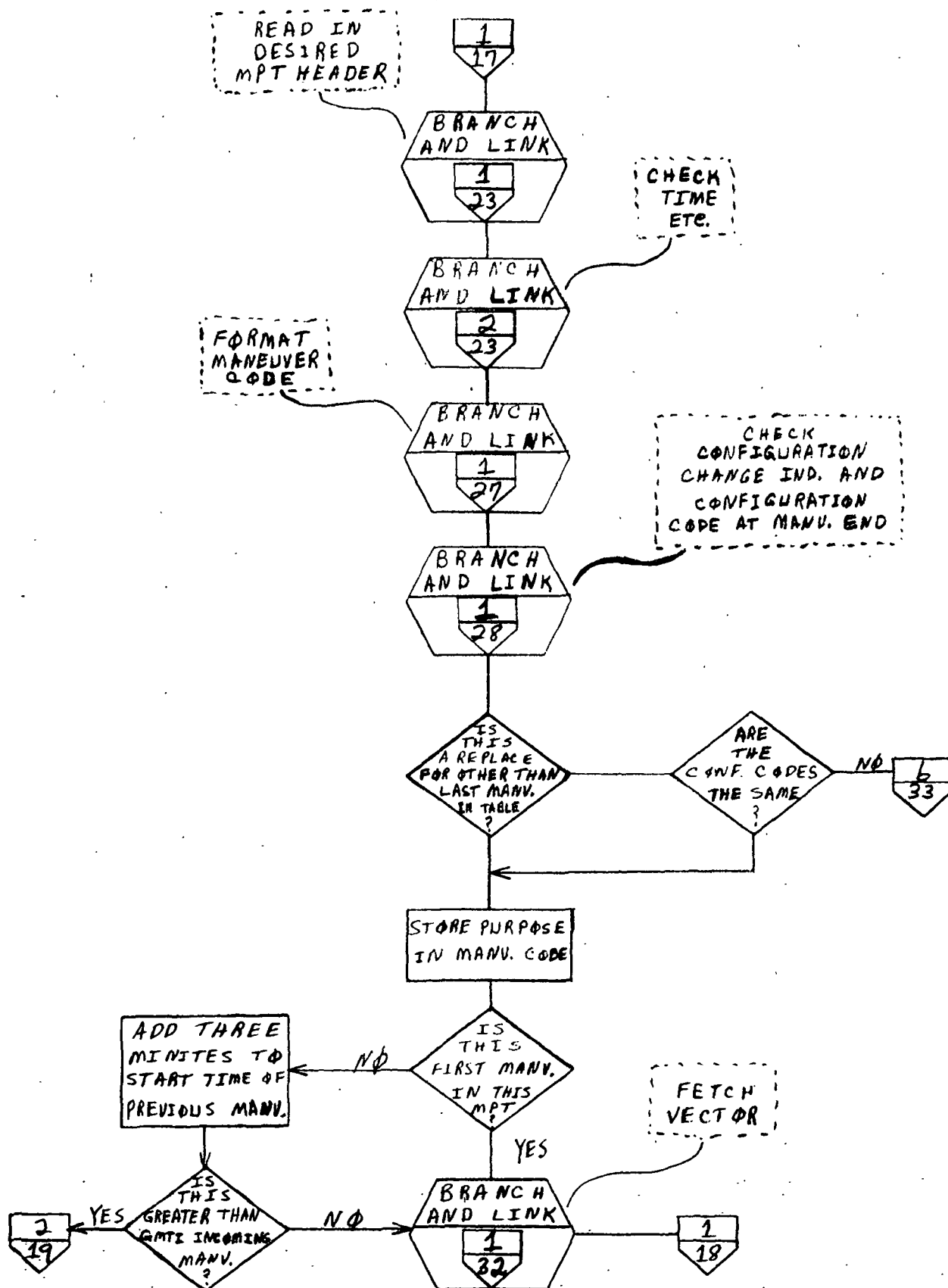


Figure PMMXFR-1. Flowchart

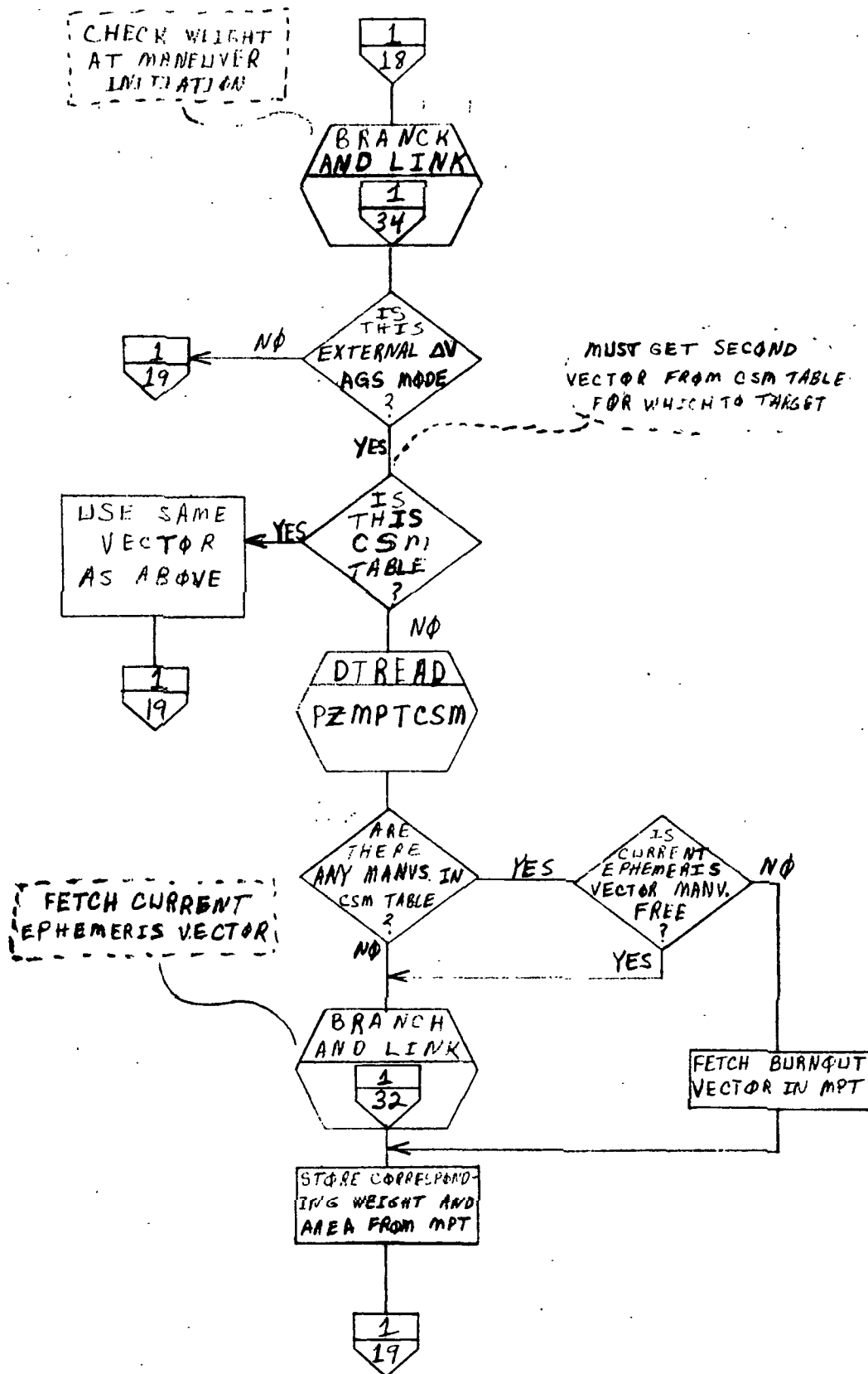


Figure PMMXFR-1. Flowchart

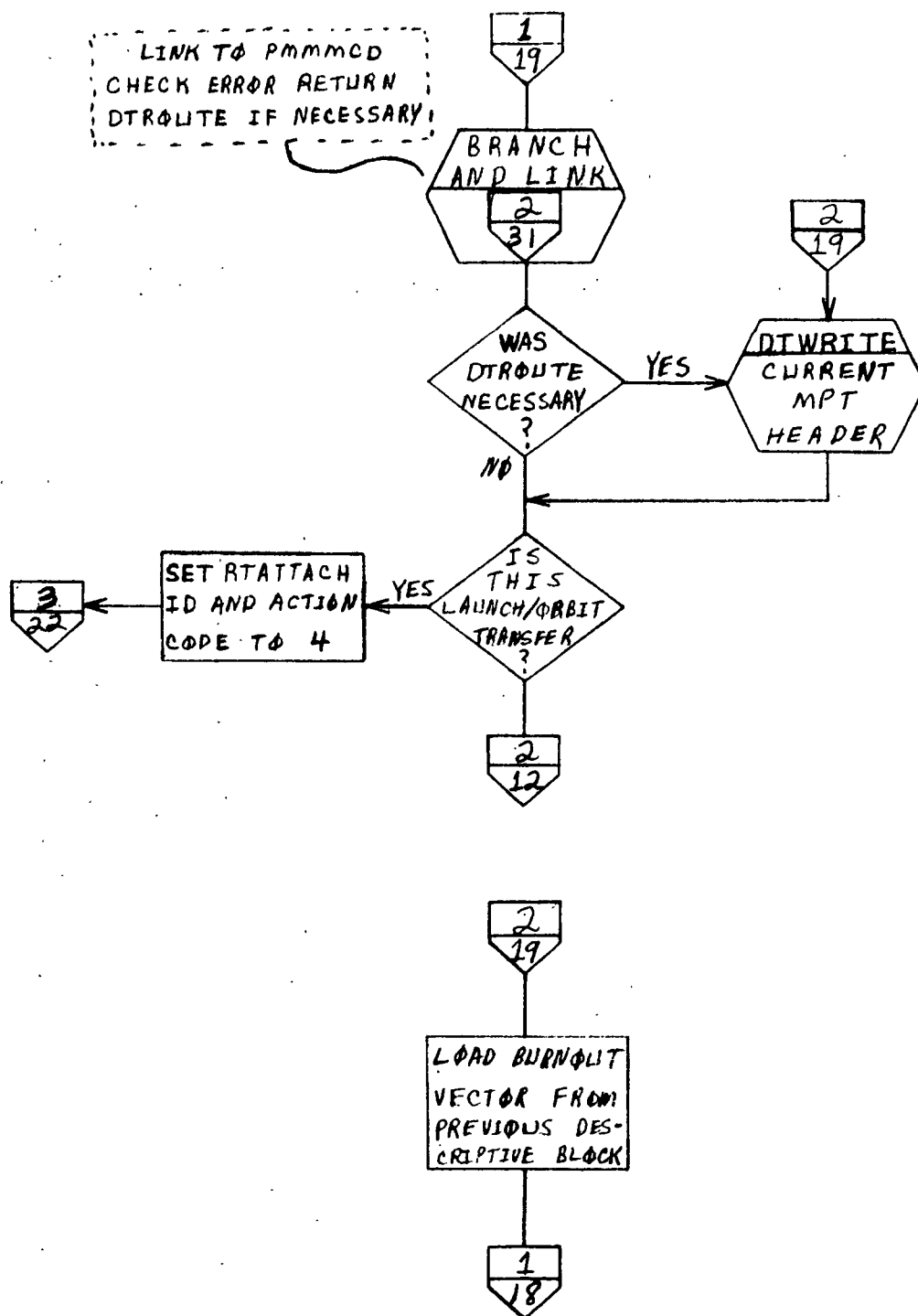


Figure PMMXFR-1. Flowchart

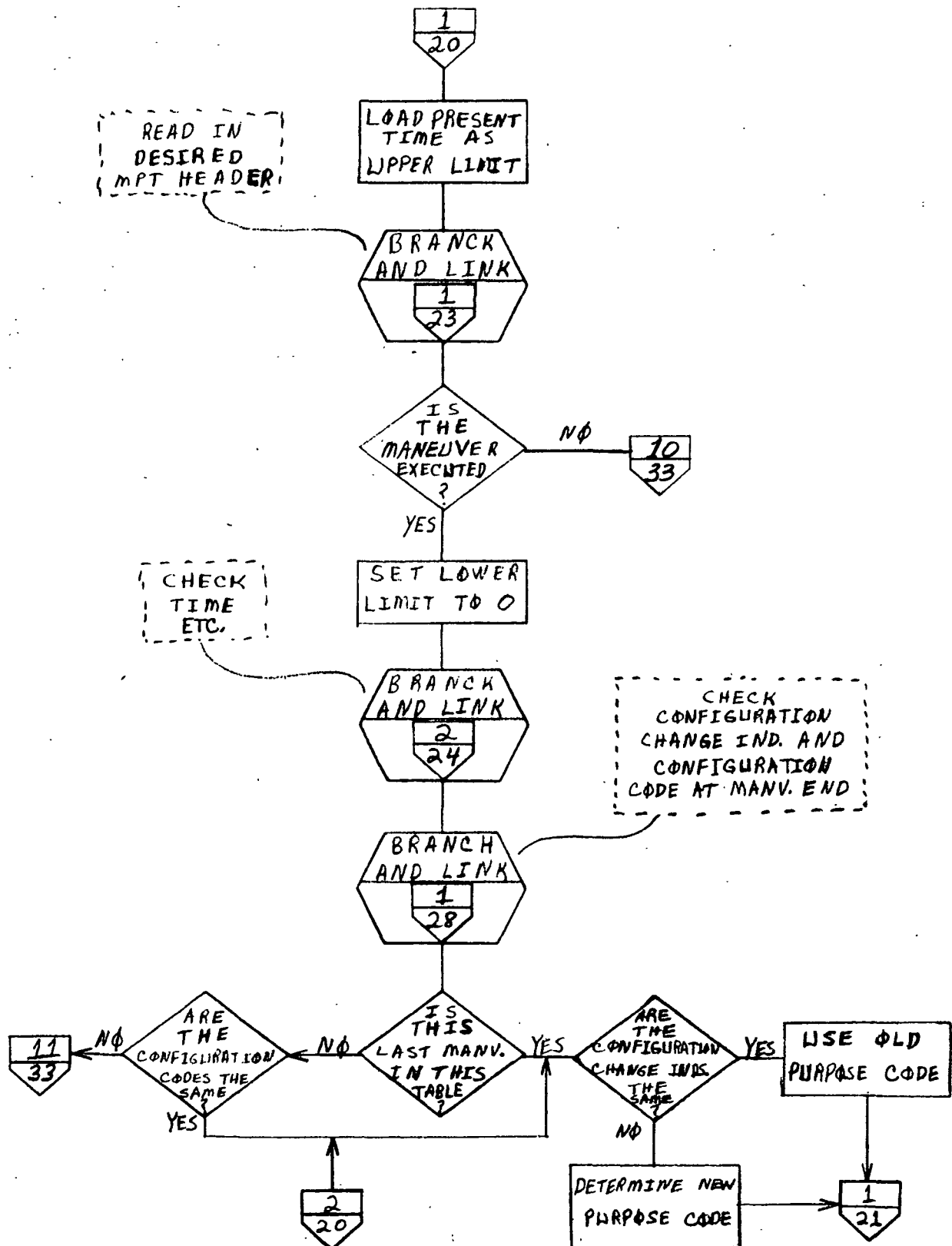


Figure PMMXFR-1. Flowchart

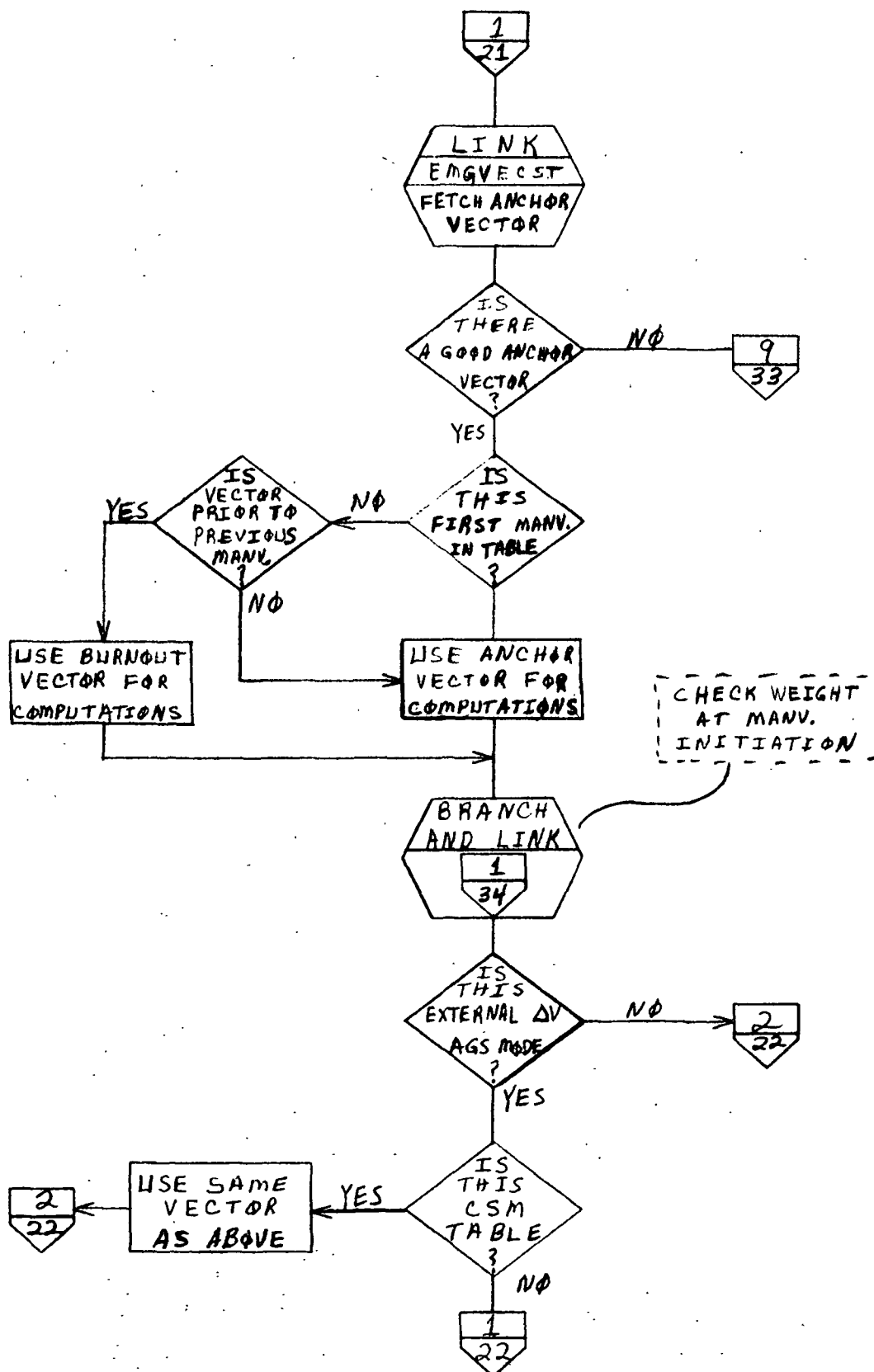


Figure PMMXFR-1. Flowchart.

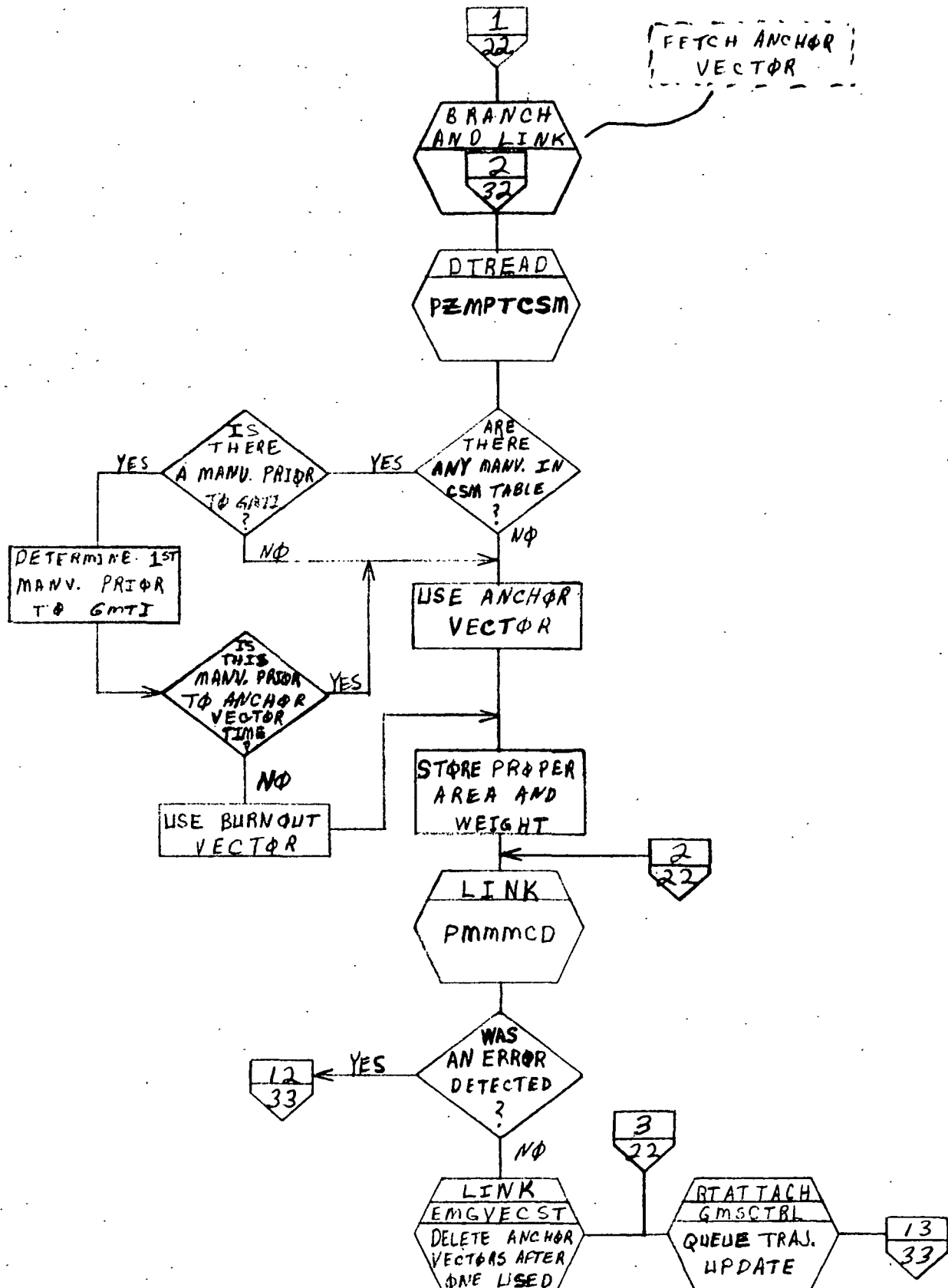


Figure PMMXFR-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMXFR

DATE 11/8/67

(EORP)

REV

PAGE 39 of 50

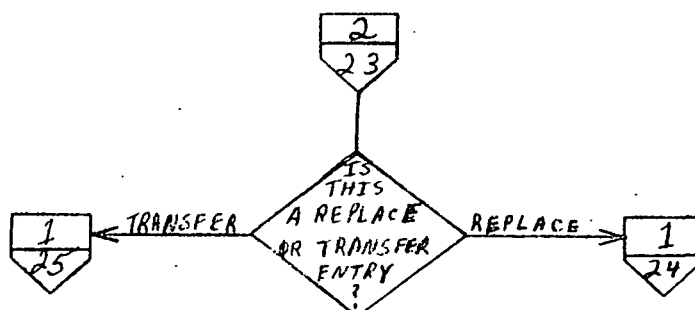
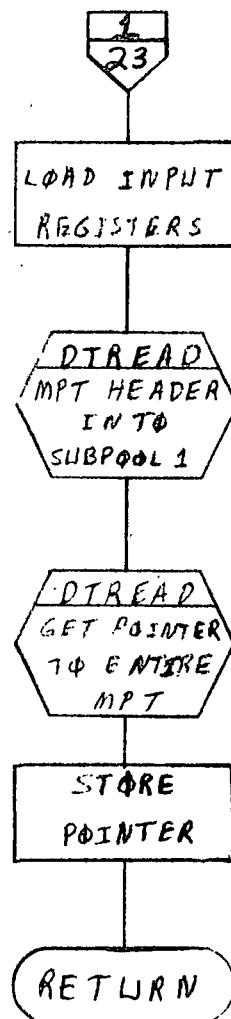


Figure PMMXFR-1. Flowchart

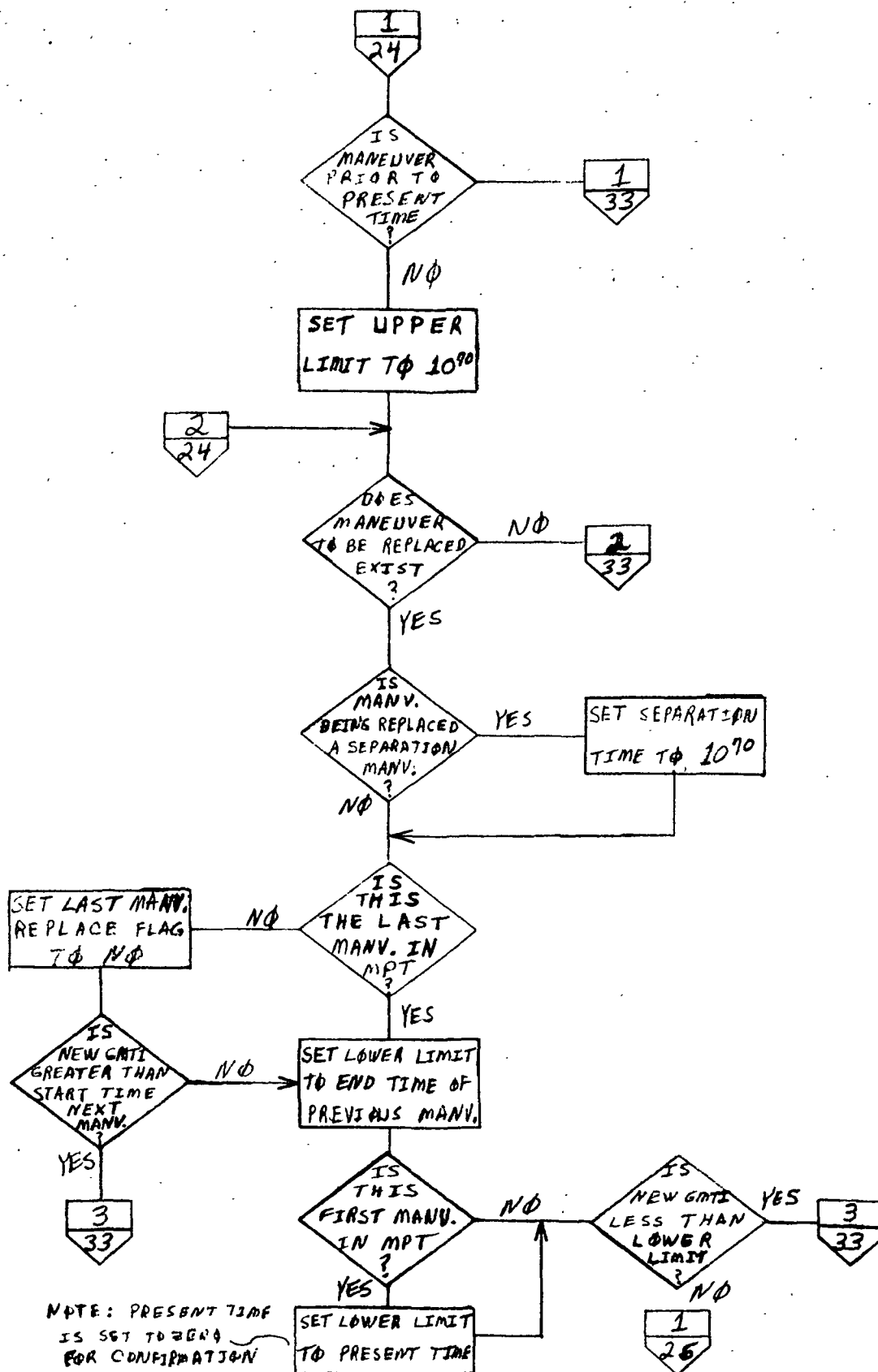


Figure PMMXFR-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMXFR

DATE 11/8/67 (EORF

REV

PAGE 41 of 50

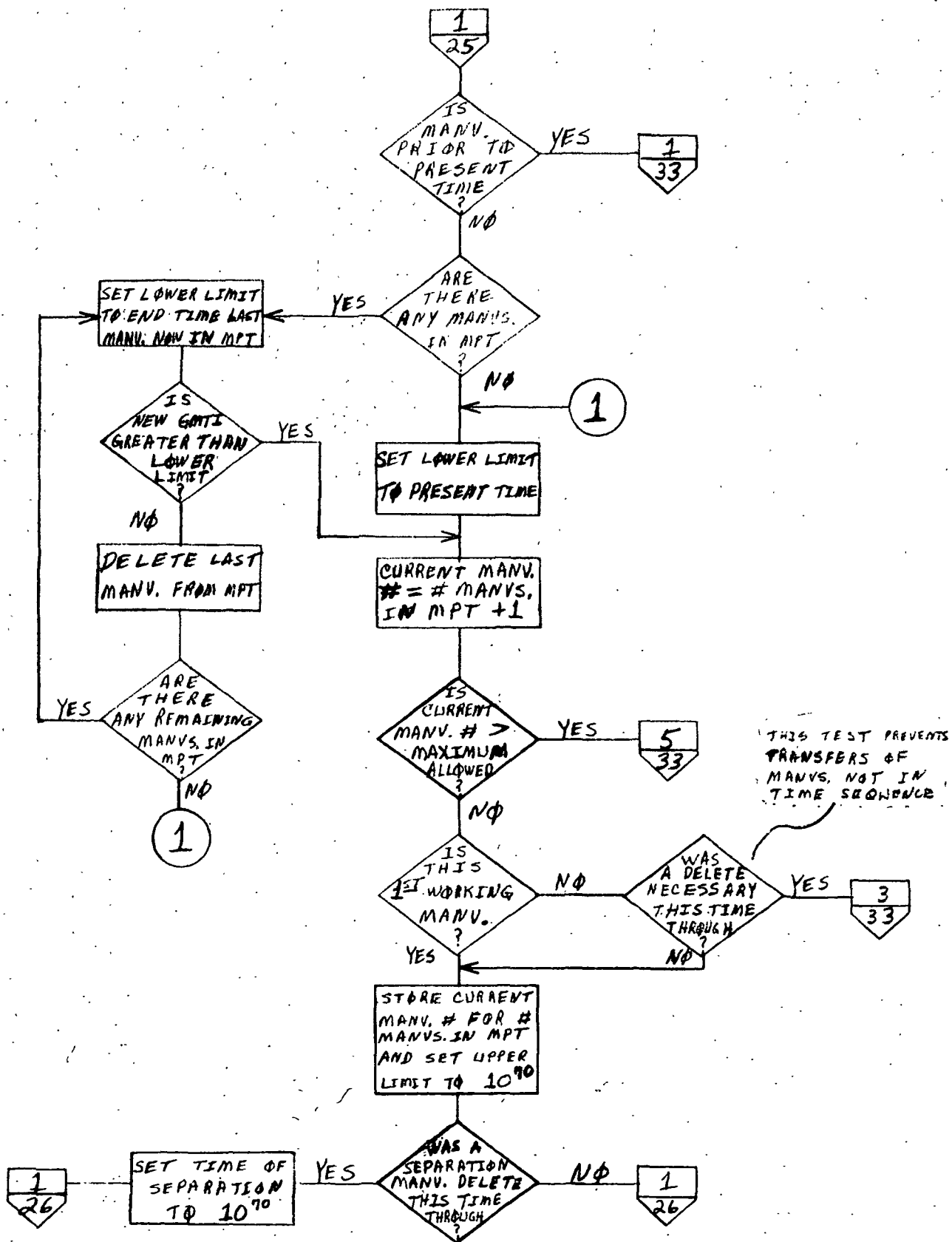


Figure PMMXFR-1. Flowchart

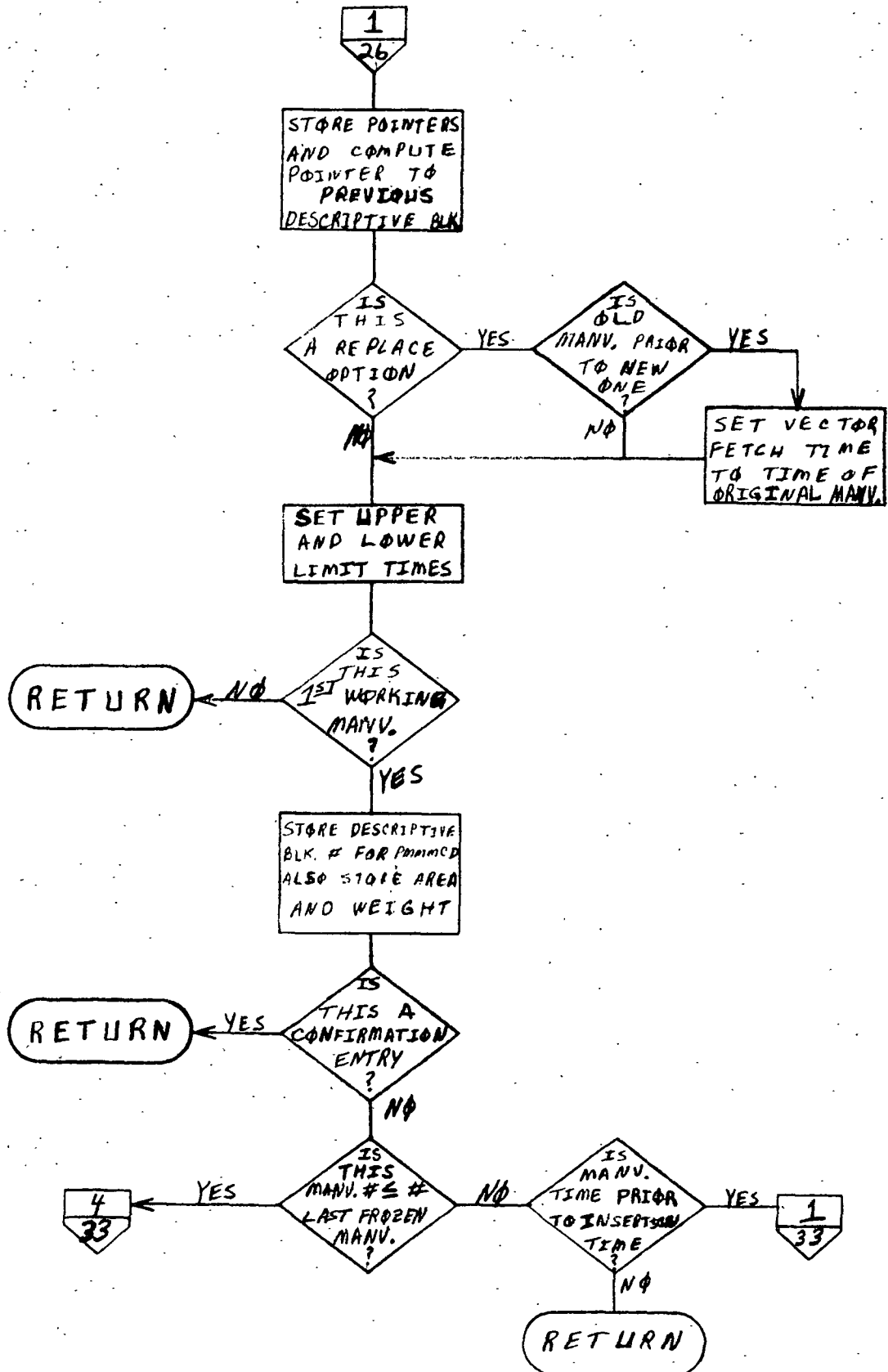


Figure PMMXFR-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMXFR

DATE 11/8/67

(EORP)

REV

PAGE 43 of 50

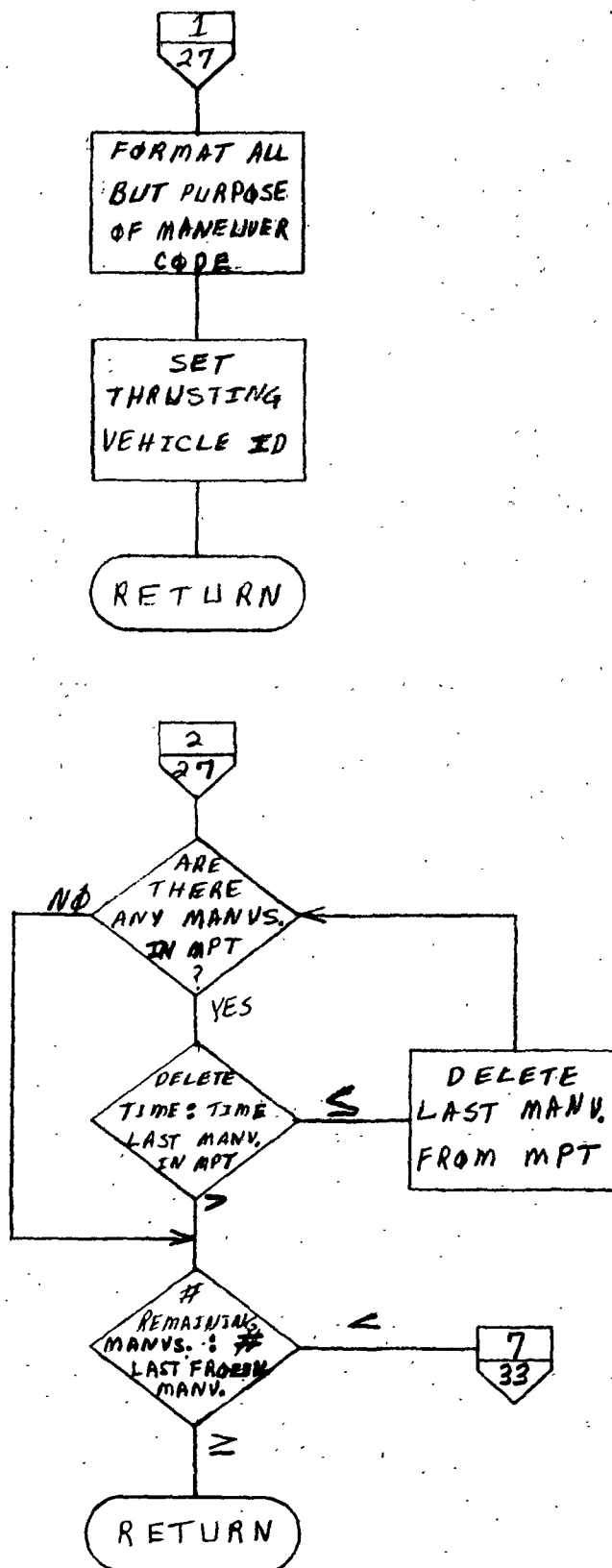


Figure PMMXFR-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMXFR
DATE 11/8/67 (EORF)
REV
PAGE 44 of 50

CCI = Configuration Change Indicator
CC_p = Configuration Code Prior to Manr.
CC = Configuration Code at end of Manr.
CC_p = Configuration Code from other table
CCMI = Configuration Code at Manr. Start
T.V. = Thrusting Vehicle

No Change = 0
Undocking = 1
Docking = 2
A/D Separation = 3

C = 0
L = 1
CL = 2
CS = 3
LS = 4
CLS = 5

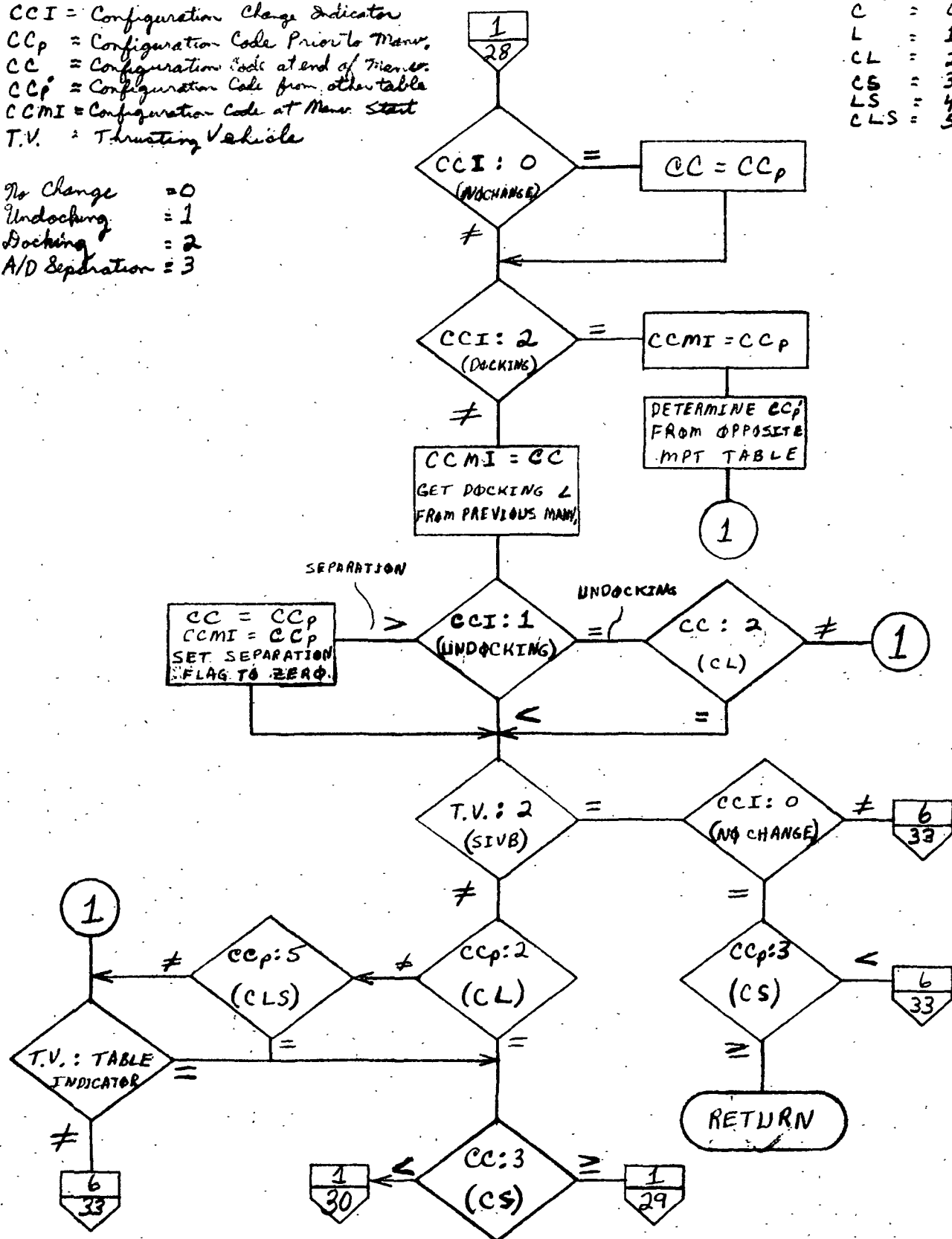


Figure PMMXFR-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMXFR

DATE 11/8/67

(EORP)

REV

PAGE 45 of 50

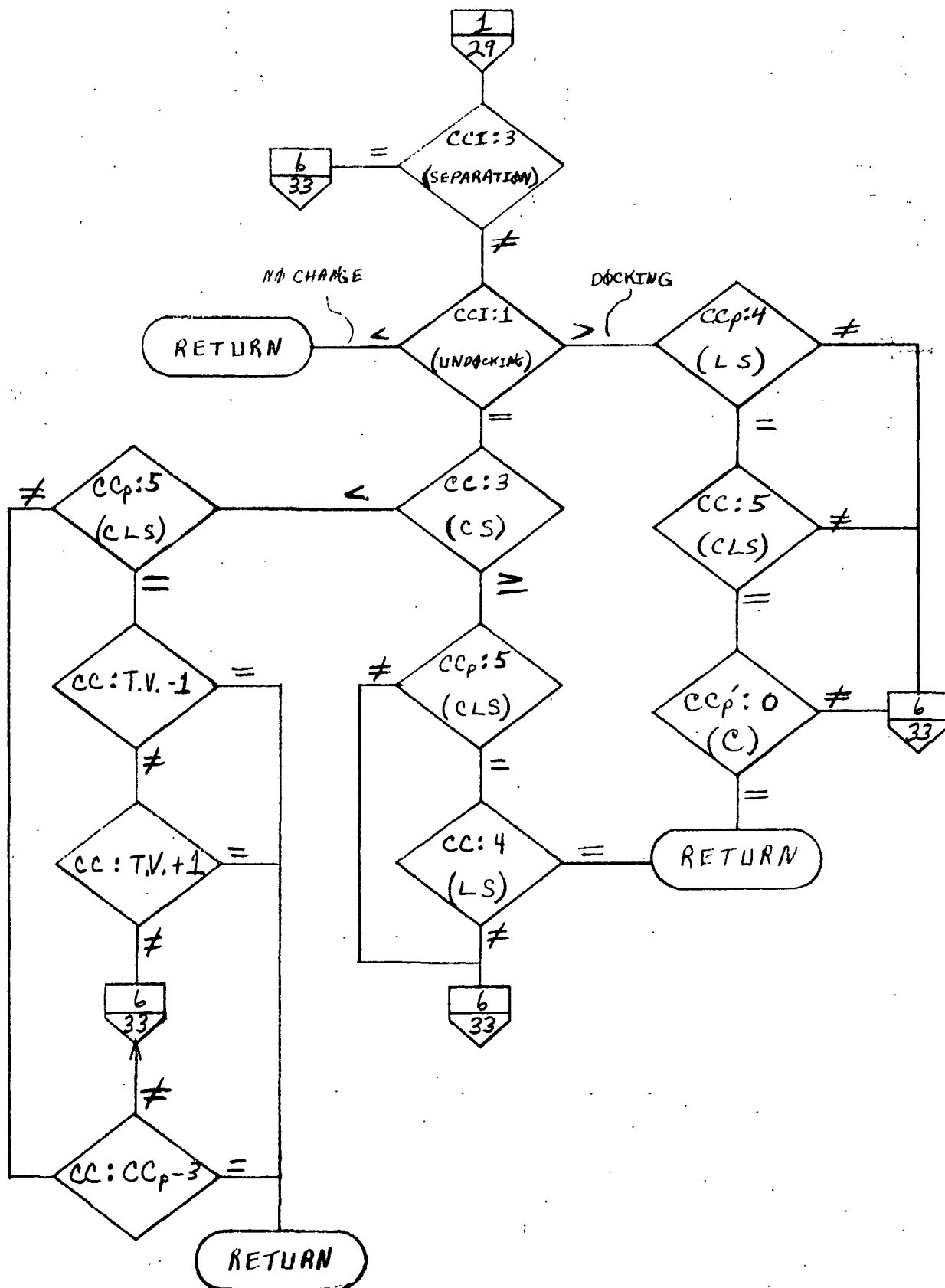


Figure PMMXFR-1. Flowchart

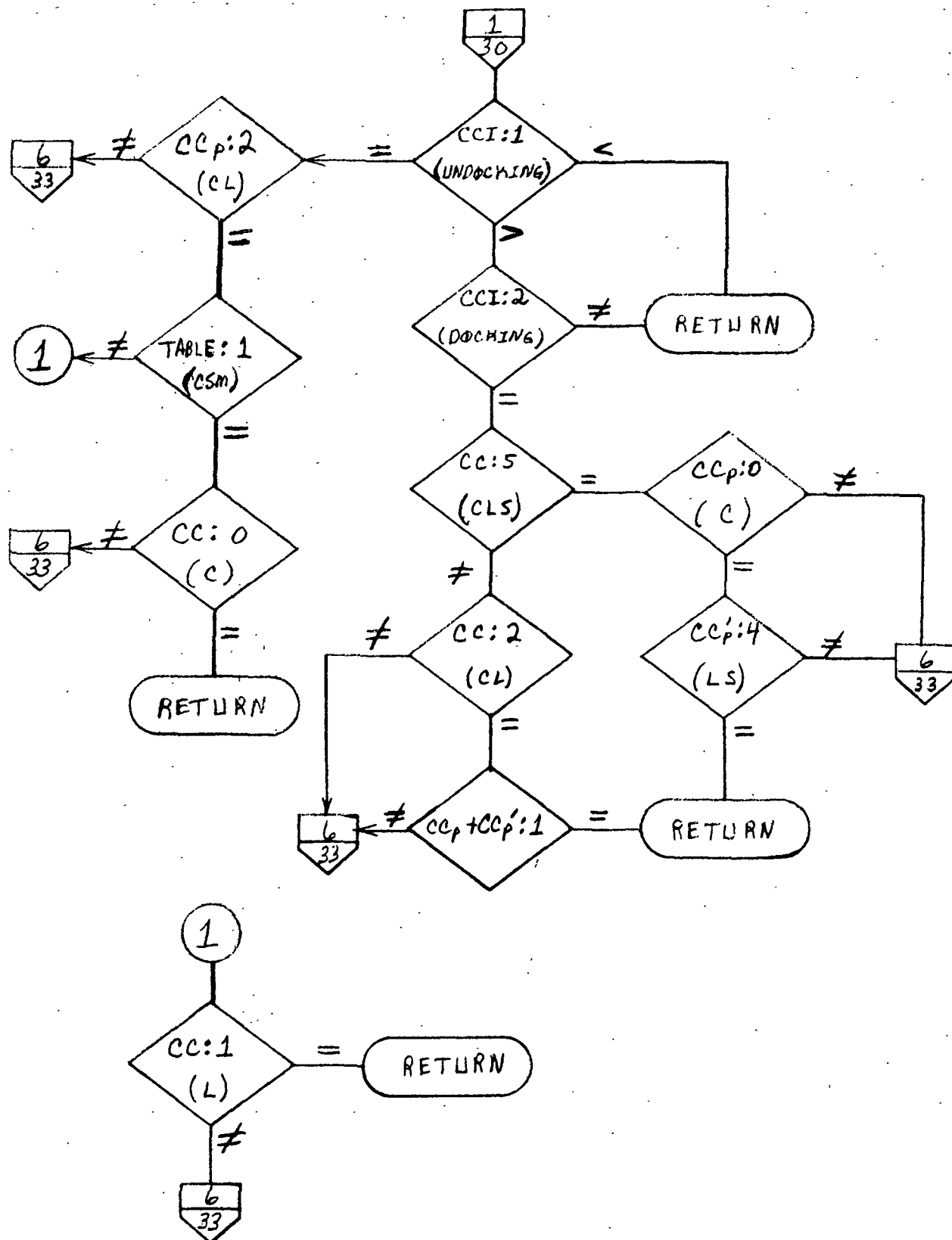


Figure PMMXFR-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMXFR

DATE 11/8/67 (EORP)

REV

PAGE 47 of 50

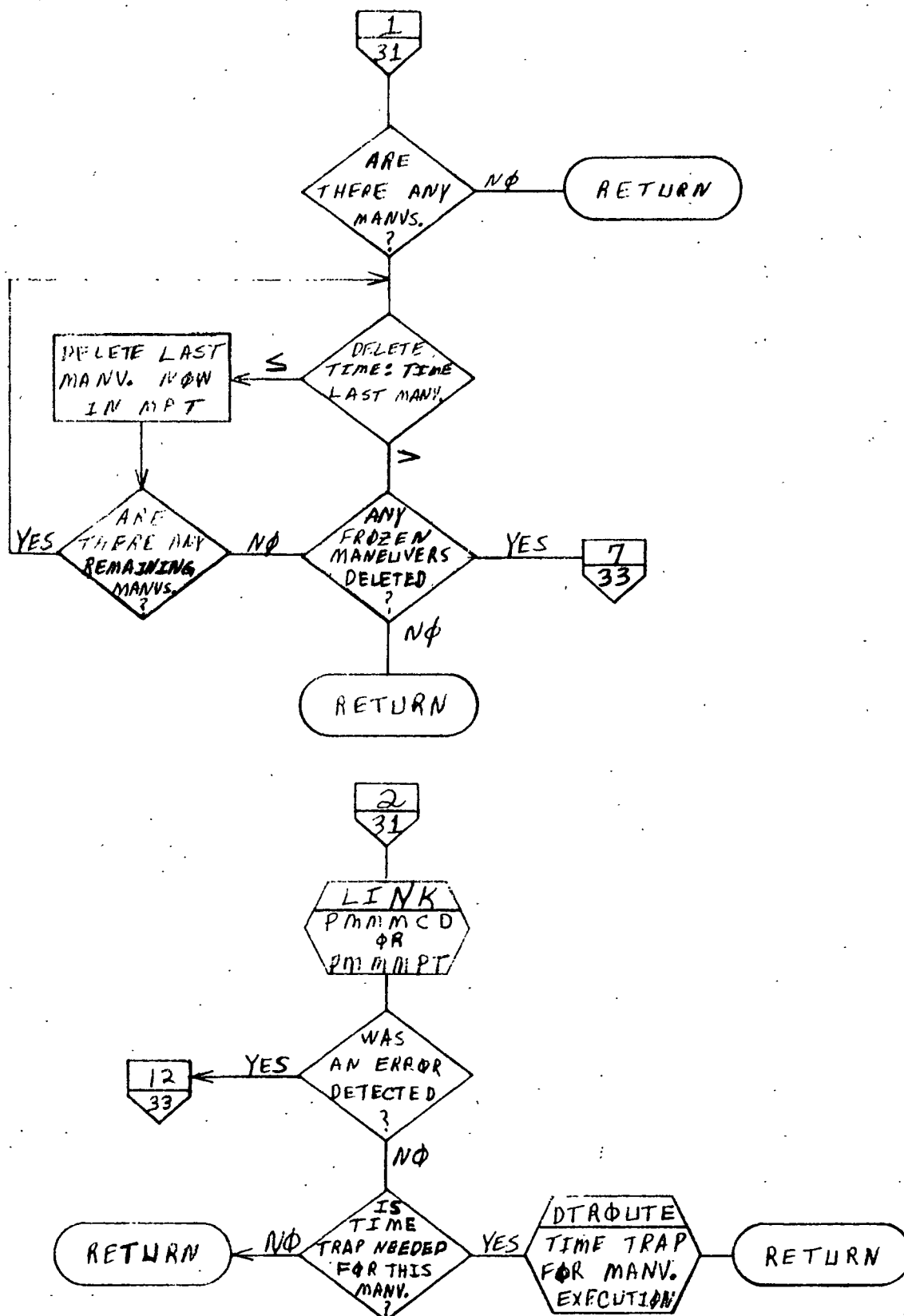


Figure PMMXFR-1. Flowchart

(Sheet 31 of 34)

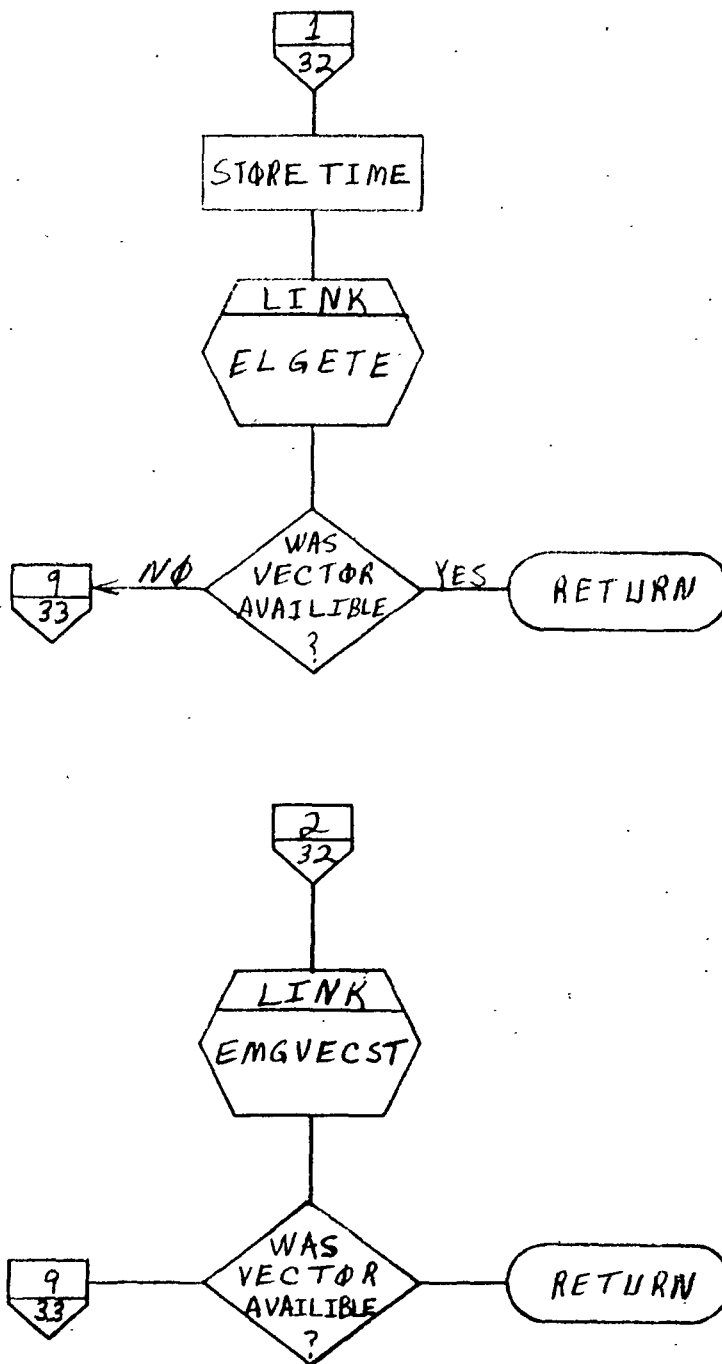


Figure PMMXFR-1. Flowchart

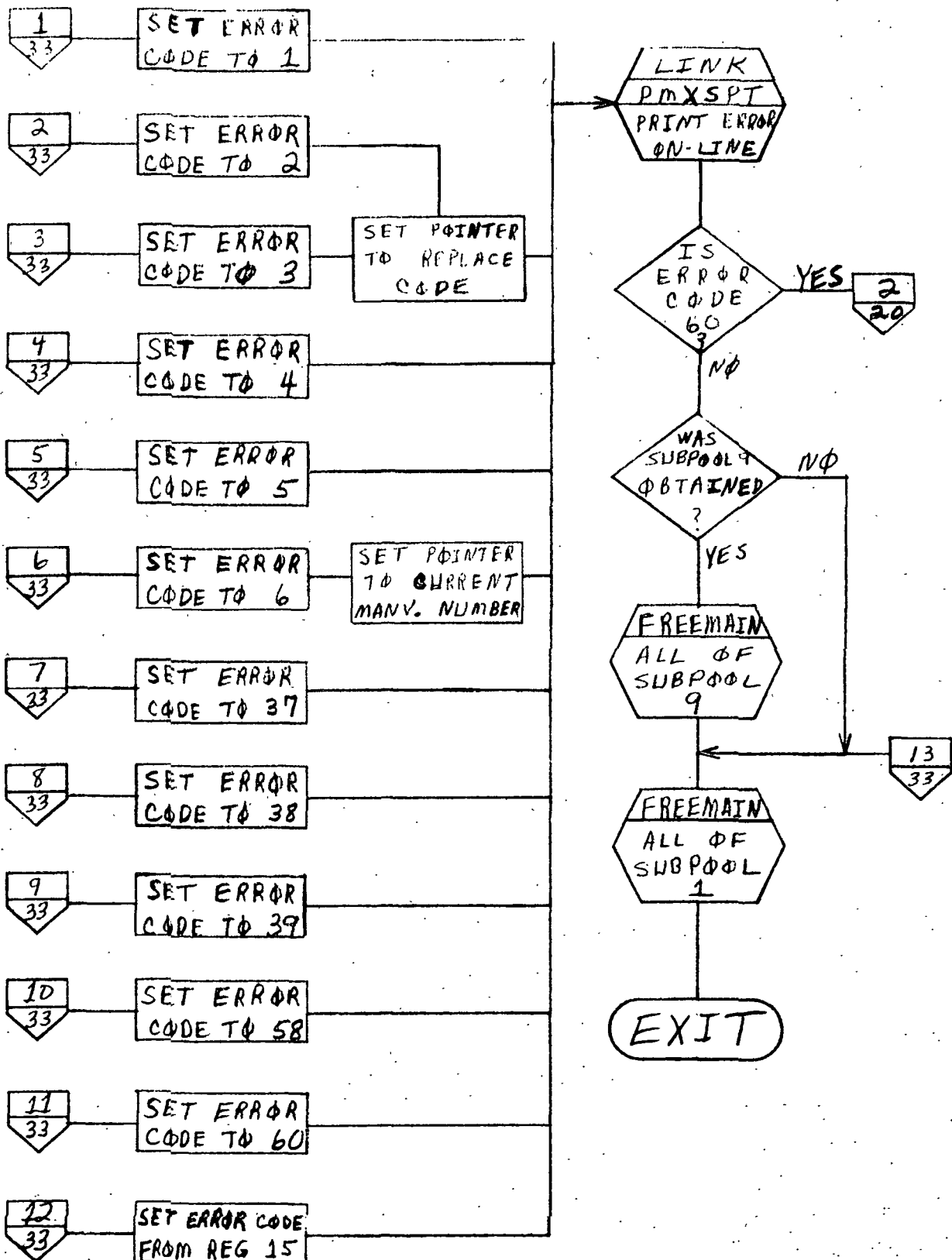


Figure PMMXFR-1. Flowchart

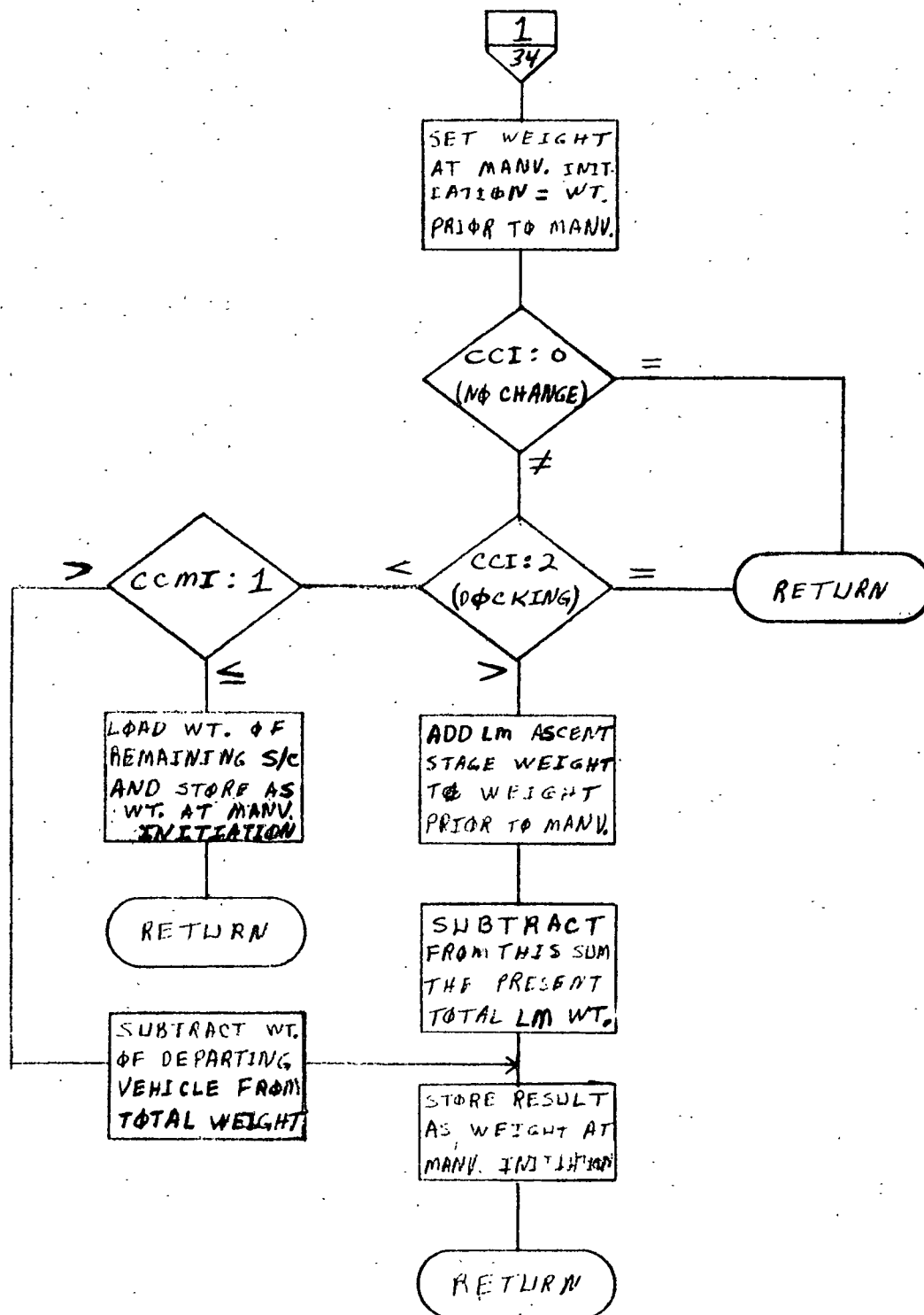


Figure PMMXFR-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

3. PMMWTC
DATE 11/22/66 (206)
REV
PAGE 1 of 12

3. PMMWTC

- a. Symbol — PMMWTC
- b. Program Name — Weight Area, Fuel Remaining and K-Factor Update
- c. Programming Group — Program Mission Planning
- d. Programmer (s) — W E Sherman
- e. Language — Assembler
- f. Data Tables Used — PZMPTCSM, PZMPTLEM and PZMPTS4B
- g. MED's Used — M50
- h. Writeup Completion Date — 10/14/66
- i. Parameter List — Table PMMWTC - 1 and - 2.
- j. Flowchart -- Figure PMMWTC - 1.

FUNCTION

PMMWTC provides the ability to update for future maneuvers, the area, weight, fuel remaining, and K-factor for a given vehicle.

USAGE

- a. Calling Sequence — RTATTACH TASK = PTMANC, EP = PMMWTC, ID = 28, GSPV = 1.
- b. Inputs — All inputs are stored in subpool 1 built by the MED decoder.
- c. Outputs and Exit — Updates quantities in the appropriate mission plan table.
- d. Other Programs Used — PMSVCT
- e. Data Restrictions and Limits — All data is checked by the MED.
- f. Error — None

METHOD

This program uses the MED input to update the specified quantities in the appropriate mission plan table and then transfers control to PMSVCT for a trajectory update.

Approval	Approval
W ⁿ E Sherman	J. J. Stein



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 3. PMMWTC
 DATE 11/22/66 (206)
 REV
 PAGE 2 of 12

Table PMMWTC - 1. Parameter List

ID — 28

QNAME — Common

From — GMSCTRL

To-Task — PTMANC

 Purpose — Update weight, area, fuel remaining
 or K-factor for a given vehicle

EP — PMMWTC

Byte No.	Type*	Units	Value / Description / Comments
			Standard MED pointers to the following data
n + 0	F	--	Vehicle number
n + 8	D	sq. ft.	Vehicle area
n + 16	D	lbs.	Vehicle weight
n + 24	D	lbs.	Vehicle SPS fuel weight
n + 32	D	lbs.	Vehicle RCS fuel weight
n + 40	D	--	Vehicle K-factor

*Type

 H — Fixed-point half word
 F — Fixed-point full word

 E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

3. PMMWTC

DATE 11/22/66 (206)

REV

PAGE 3 of 12

Table PMMWTC - 2. Parameter List

ID — 8

QNAME — Common

From — PMMWTC

To-Task — PTMANC

Purpose — Causes a trajectory update

EP — PMSVCT

Byte No.	Type*	Units	Value / Description / Comments
0			Open
4	F	--	Vehicle number for update
8	F	--	Zero

*Type
H — Fixed-point half word
F — Fixed-point full word

E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

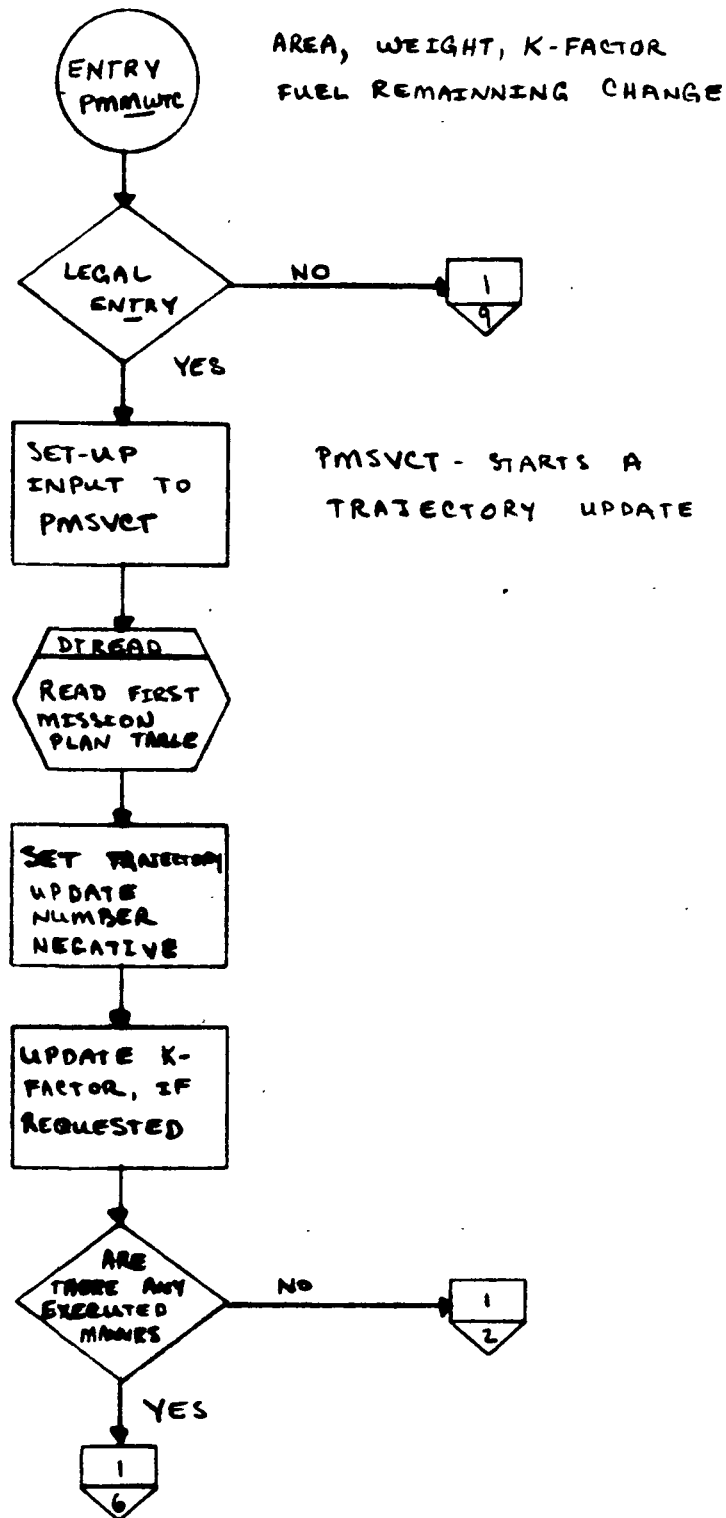


Figure PMMWTC - 1. Flowchart

(Sheet 1 of 9)

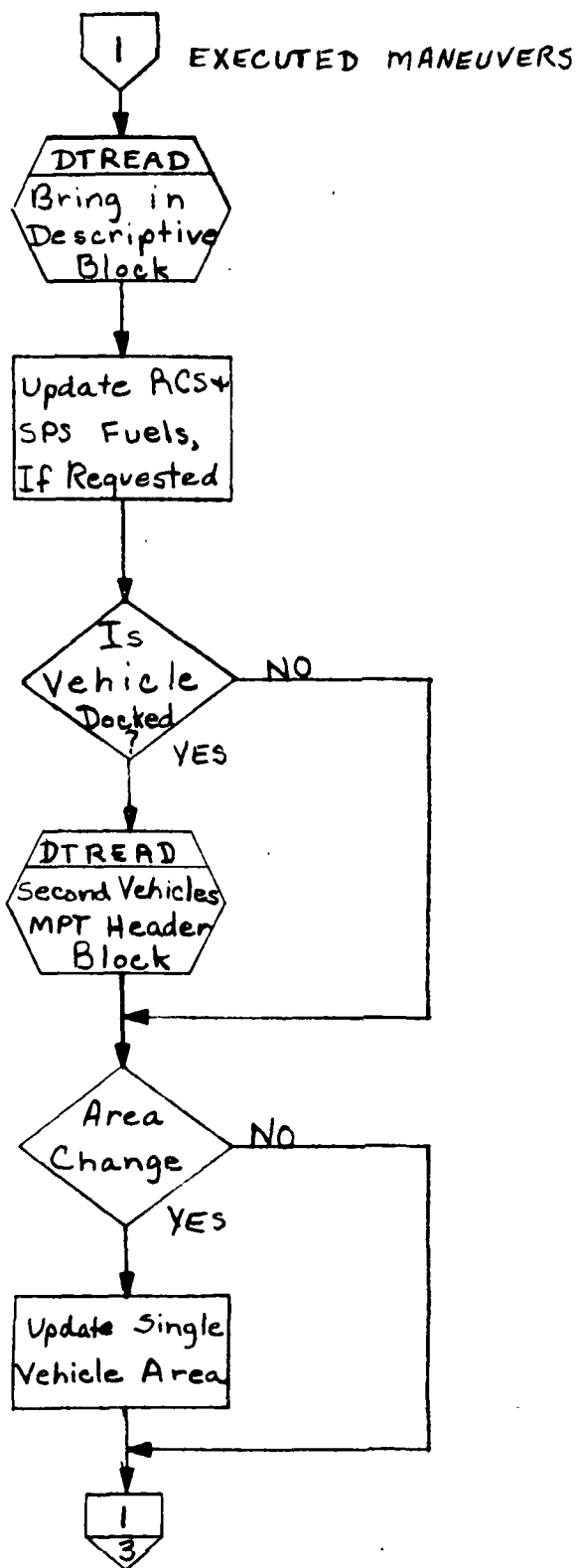


Figure PMMWTC - 1. Flowchart

(Sheet 2 of 9)

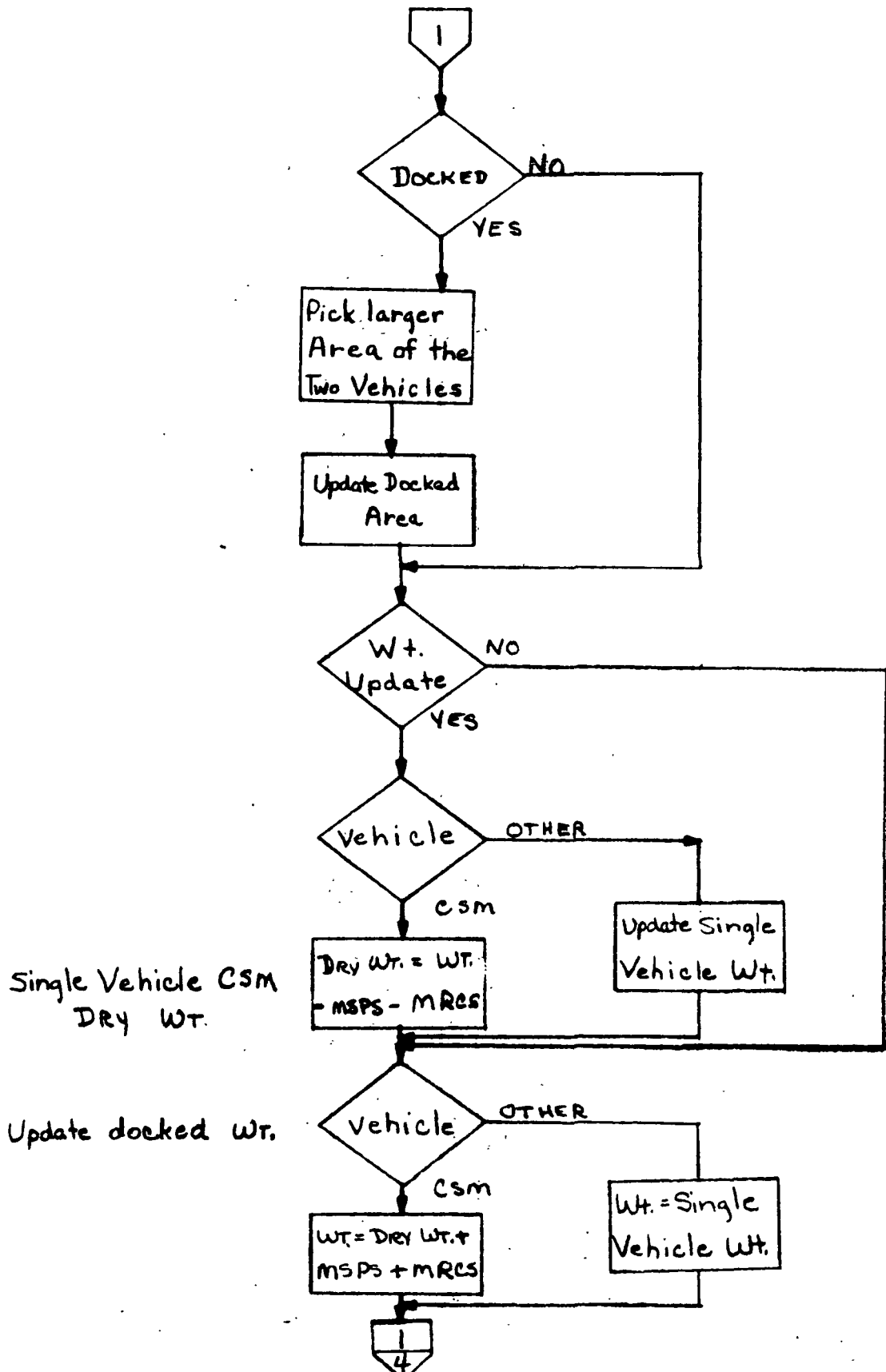


Figure PMMWTC - 1. Flowchart

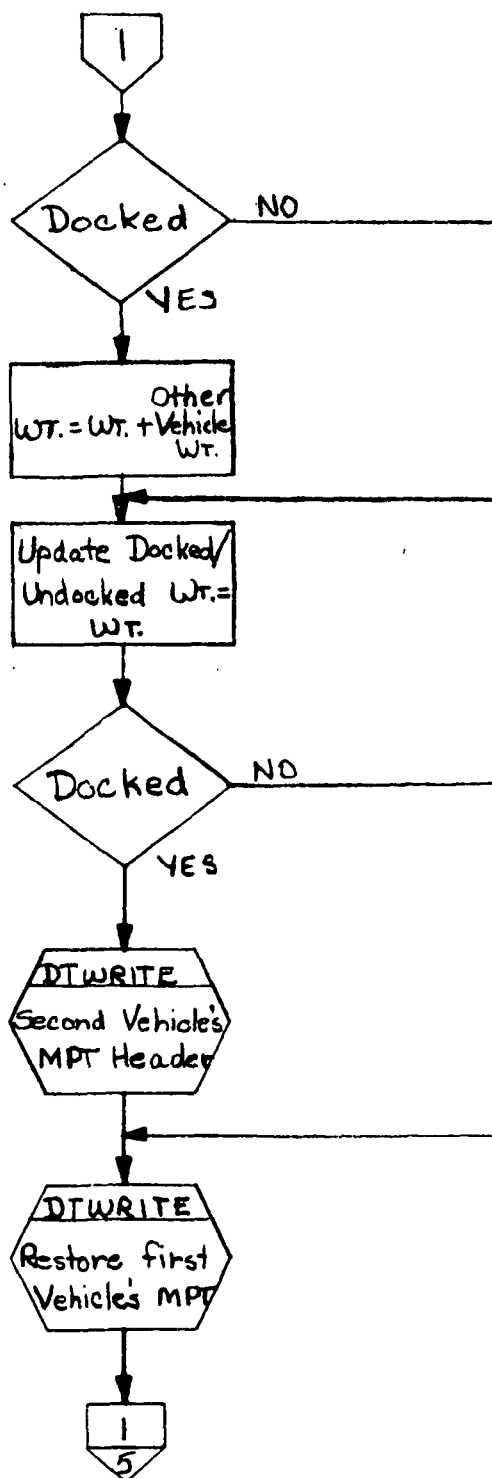


Figure PMMWTC - 1. Flowchart

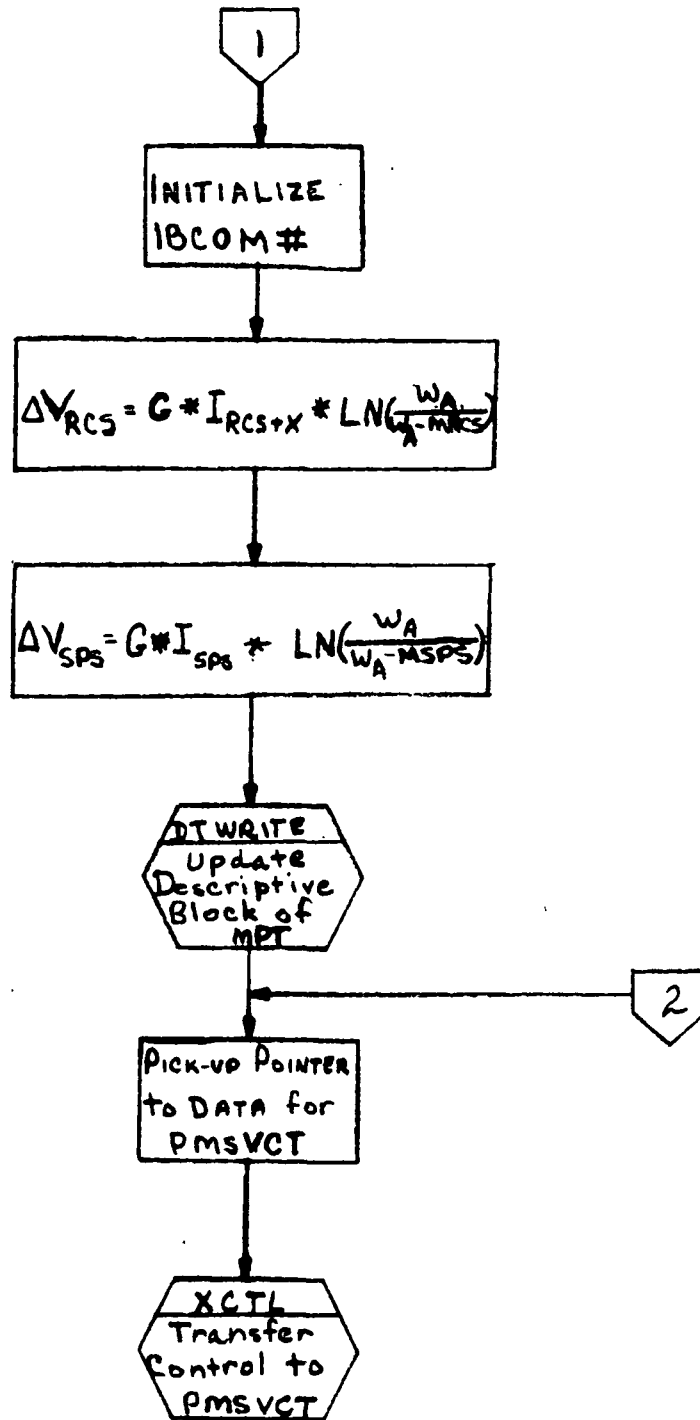


Figure PMMWTC - 1. Flowchart

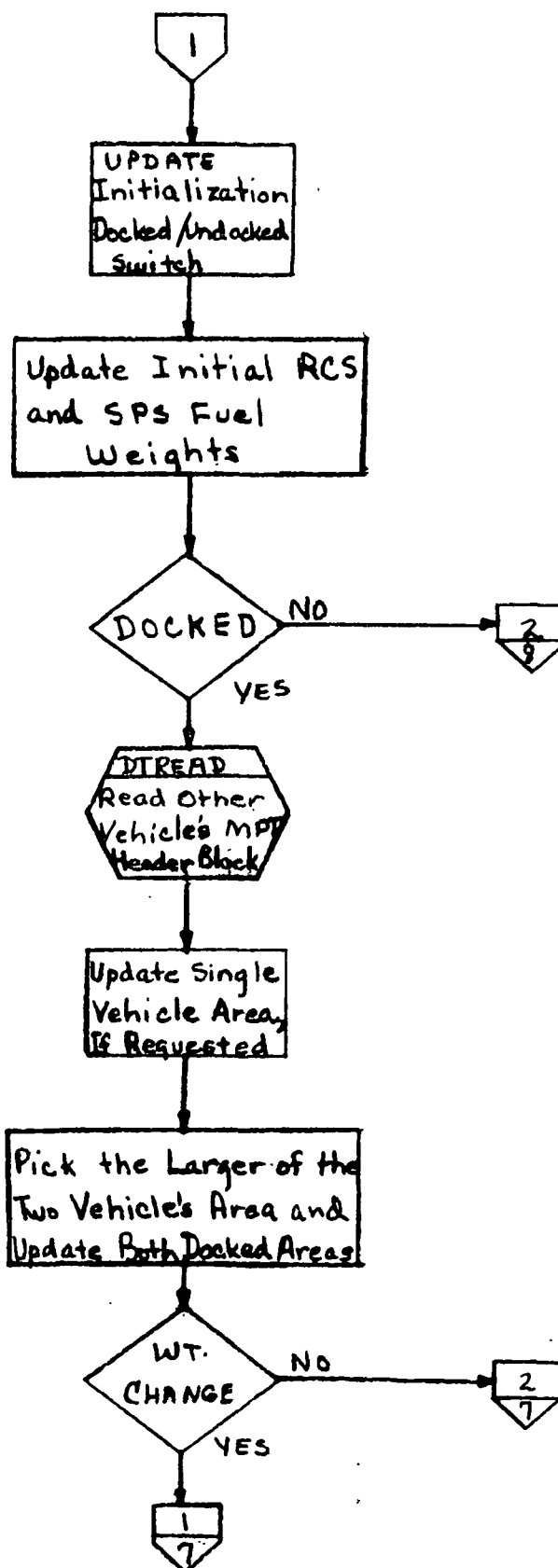


Figure PMMWTC - 1. Flowchart

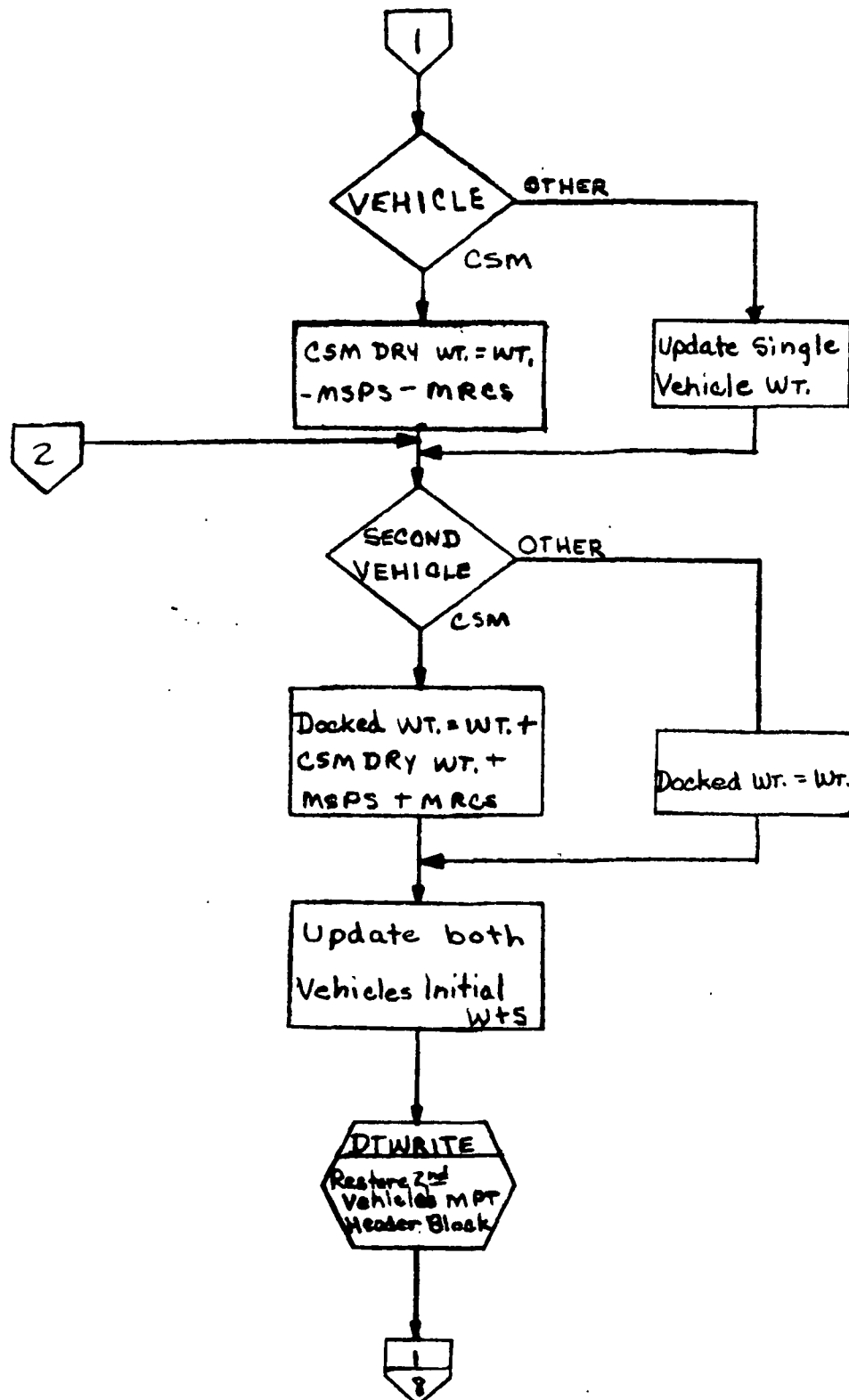


Figure PMMWTC - 1. Flowchart

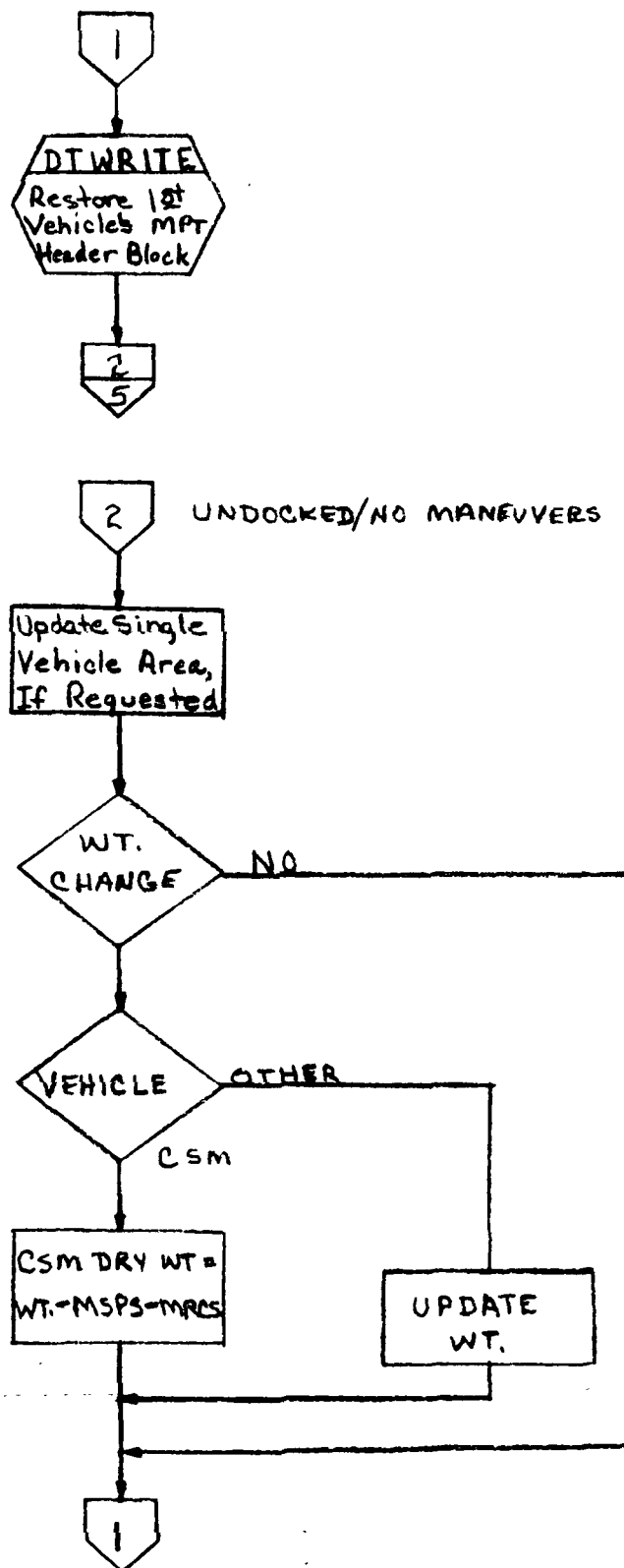


Figure PMMWTC - 1. Flowchart

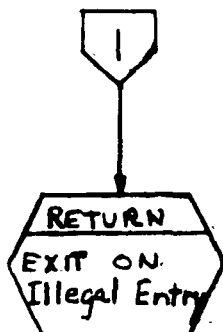


Figure PMMWTC - 1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMWTC

DATE 10/10/67 (EORP)

REV

PAGE 1 of 15

- a. Symbol — PMMWTC
- b. Program Name — Vehicle Weight Change Module
- c. Programming Group — Maneuver Control (Mission Planning)
- d. Programmer — B. Oliver
- e. Language — Assembler
- f. Data Tables Used — PZMPTCSM and PZMPTLEM
- g. MED's Used — M50, M55
- h. Writeup Completion Date — 9/14/67
- i. Parameter List — Tables PMMWTC - 1 through -3
- j. Flowchart — Figure PMMWTC-1

FUNCTION

This program provides the capability of changing the CSM, LM, and S-IVB vehicle area and weight in the CSM or LM Mission Plan Table. It also provides the capability of changing the following MPT parameters: (1) the orbit density multiplier (K-factor), (2) the initial configuration, (3) the CSM/LM docking angle, and (4) the time to begin S-IVB continuous venting.

USAGE

- a. Calling Sequence — RTATTACH TASK = PTMANC, EP = PMMWTC, GSPV = 1, ID = n, PARAM = (SUBPOOL, 4)
- b. Inputs — 1. MED quantities described in parameter lists
2. Data tables PZMPTCSM and PZMPTLEM
- c. Outputs and Exit — Output is to data tables PZMPTCSM and PZMPTLEM. Exit is via XCTL with control passed to PMSVCT.
- d. Other Programs Used — PMXSPT, PMSVCT
- e. Data Restrictions and Limits — None except limit restrictions placed on the MED.
- f. Error — The following error messages are printed on line:
 - 1. PMMWTC. ILLEGAL ENTRY
 - 2. PMMWTC. DELETE MPT MANEUVERS PRIOR TO ENTERING M55
 - 3. PMMWTC. INVALID MNVR SPECIFIED ON M50

METHOD

The method is described with an explanation of the processing of each input.

1. Vehicle Cross-Sectional Area or Weight (M50)

The CSM, LM, or S-IVB cross-sectional area or weight may be changed at present time or at the end of any planned MPT maneuver.

Approval	Approval
<i>R.E. Oliver</i>	<i>S.E. James</i>

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMWTC

DATE 10/10/67 (EC

REV

PAGE 2 of 15

The input values represent an undocked (single vehicle) configuration and are stored in words 13-18 of the MPT header if

- (a) there are no executed maneuvers in the MPT
- (b) a future maneuver was not specified on the M50 MED.

Otherwise, the input values will be stored in words 45-50 of the appropriate maneuver block. The composite vehicle area (the maximum area of any single vehicle of the configuration) and weight (the sum of the weights of the single vehicles of the confirmation) is then determined and stored in the MPT header after the appropriate maneuver.

2. K-Factor (M50)

The input value is stored in word 4 of the MPT header and is used as the orbit density multiplier for trajectory processing.

3. Initial MPT Configuration (M55)

The initial MPT configuration is stored in word 12 of the MPT header. The single vehicle cross-sectional areas and weights in words 13-18 of the MPT header are then used to determine the composite vehicle area and weight. This is done in the same manner as for (1) above.

The input of any of the above quantities (1, 2, or 3) will result in a queue to PMSVCT requesting that a trajectory update be performed.

4. Time to Begin S-IVB Continuous Venting (M55)

The time to begin S-IVB continuous venting is stored in word 11 of the MPT header. This value is the time associated with the initial S-IVB weight.

5. Docking Angle (M55)

The docking angle is stored in word 21 of the MPT header.

IBM**RTCC APOLLO PROGRAMMING SYSTEMS**

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

Table PMMWTC-1. Parameter List

2. PMMWTC

DATE 10/10/67 (EORP)

REV

PAGE 3 of 15

QUEID — 28

QNAME — NA

From — GMSCTRL

To-Task — PTMANC

Purpose — Manual entry of vehicle area,
RTATTACH weight, or K-factor (M50)

EP — PMMWTC

Byte No.	Type*	Units	Value / Description / Comments
1-4	F		Pointer to the following MED parameters in subpool 1:
n	F		Vehicle
n+4	F		Maneuver number
n+8	D	ft. ²	CSM area
n+16	D	lbs.	CSM weight
n+24	D	ft. ²	S-IVB area
n+32	D	lbs.	S-IVB weight
n+40	D	ft. ²	LM area
n+48	D	lbs.	LM weight
n+56	D		K-factor

*Type

H — Fixed-point half word

F — Fixed-point full word

E — Single-precision (floating-point)

D — Double-precision (floating-point)

X — Hexadecimal



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

Table PMMWTC-2. Parameter List

2. PMMWTC

DATE 10/10/67 (EOI)

REV

PAGE 4 of 15

QUEID — 38

QNAME — NA

From — GMSCTRL

To-Task — PTMANC

Purpose — Manual entry of MPT configuration, EP — PMMWTC
 RTATTACH time to begin S-IVB venting, or CSM/LM docking angle (M)

Byte No.	Type*	Units	Value / Description / Comments
1-4	F		Pointer to the following MED parameters in subpool 1:
n	F		Vehicle
	F		Not used
n+8	F		Configuration Code:
			0 - CSM
			1 - LM
			2 - CSM and LM
			3 - CSM and S-IVB
			4 - LM and S-IVB
			5 - CSM, LM, and S-IVB
			6 - LM ascent stage
			7 - CSM and LM ascent stage
	F		Not used
n+16	D	STU	GMT to begin S-IVB venting
n+24	D	rad.	Docking angle

*Type

H — Fixed-point half word

F — Fixed-point full word

E — Single-precision (floating-point)

D — Double-precision (floating-point)

X — Hexadecimal



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMMWTC

DATE 10/10/67(EORP)

REV

PAGE 5 of 15

Table PMMWTC-3. Parameter List

.XCTL

QUEID — 8

QNAME — NA

From — PMMWTC

To-Task — NA

Purpose — Select and route vector for vehicle
trajectory update

EP — PMSVCT

Byte No.	Type*	Units	Value / Description / Comments
0			Open
4	F		Vehicle for update
8	F		Zero

*Type
H — Fixed-point half word
F — Fixed-point full word

E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

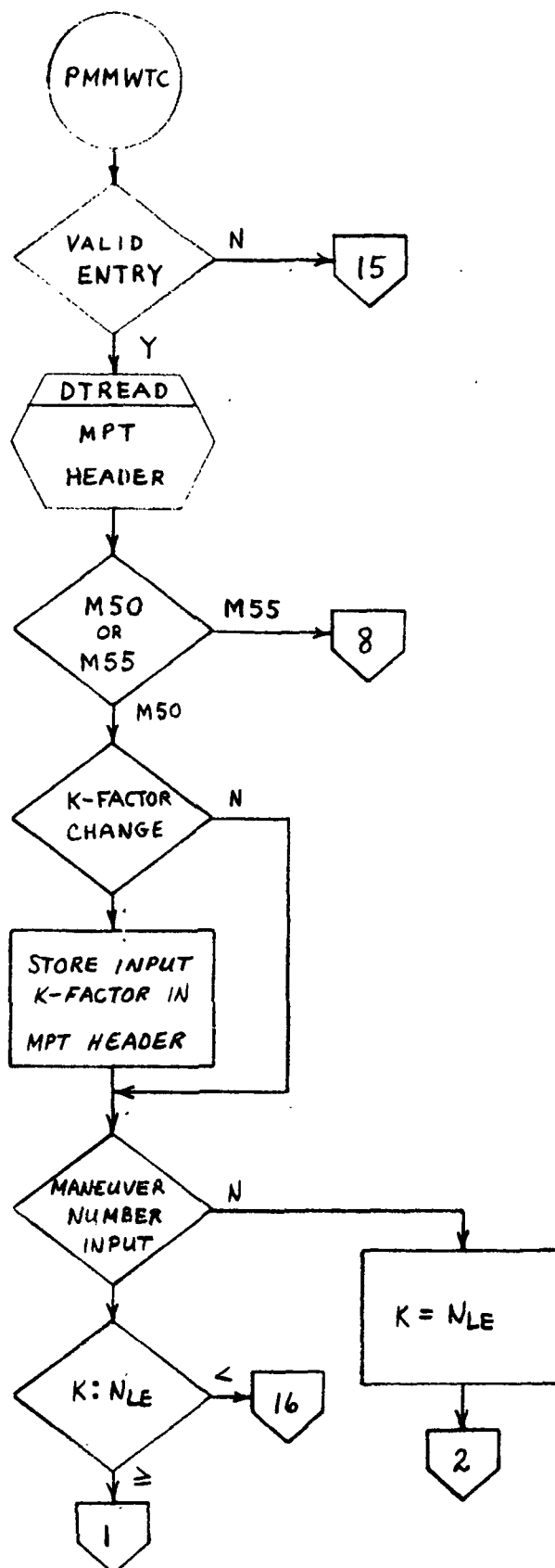


Figure PMMWTC-1. Flowchart

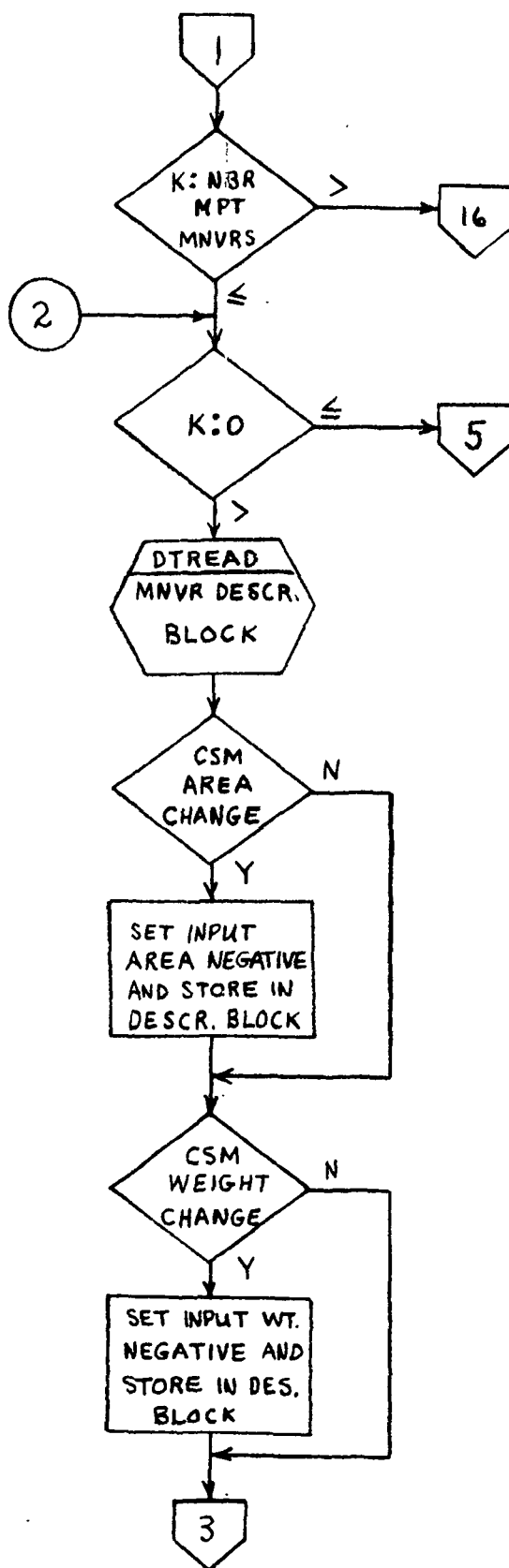


Figure PMMWTC-1. Flowchart

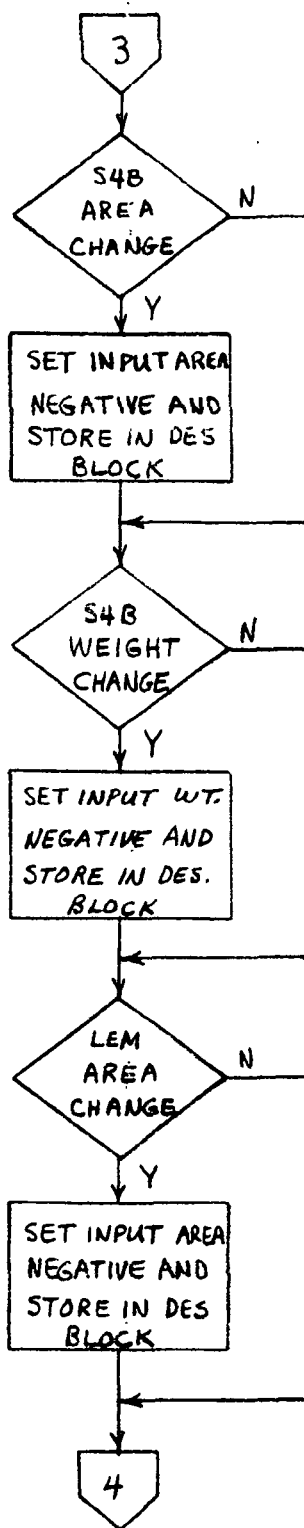


Figure PMMWTC-1. Flowchart

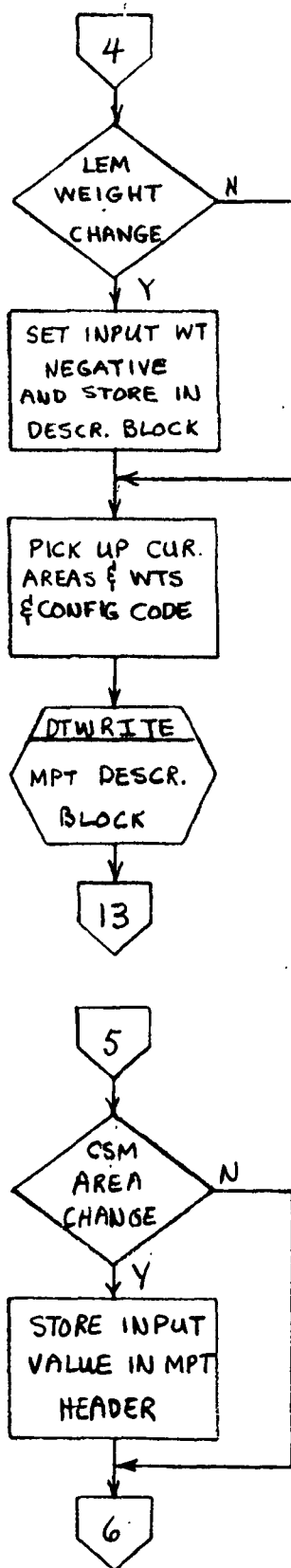


Figure PMMWTC-1. Flowchart

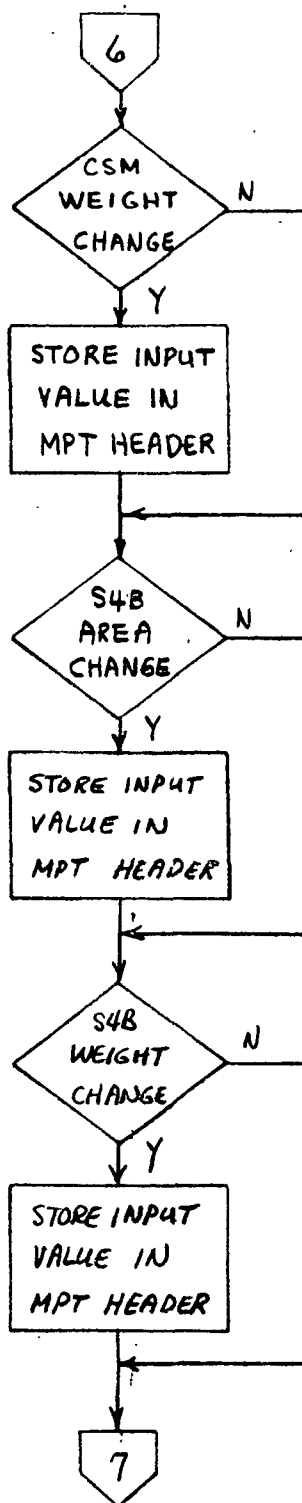


Figure PMMWTC-1. Flowchart

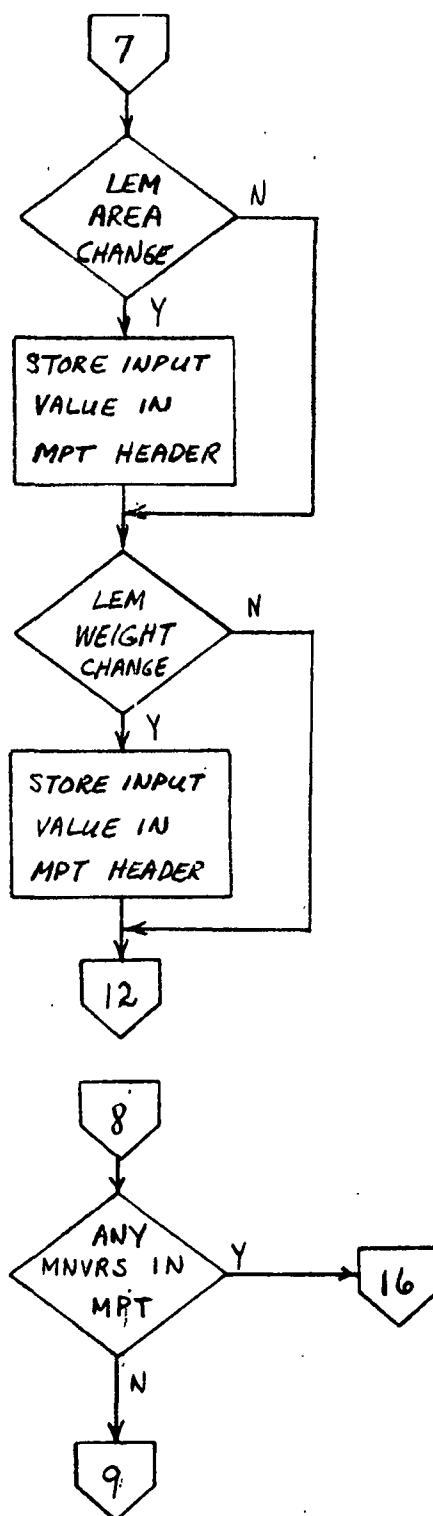


Figure PMMWTC-1. Flowchart

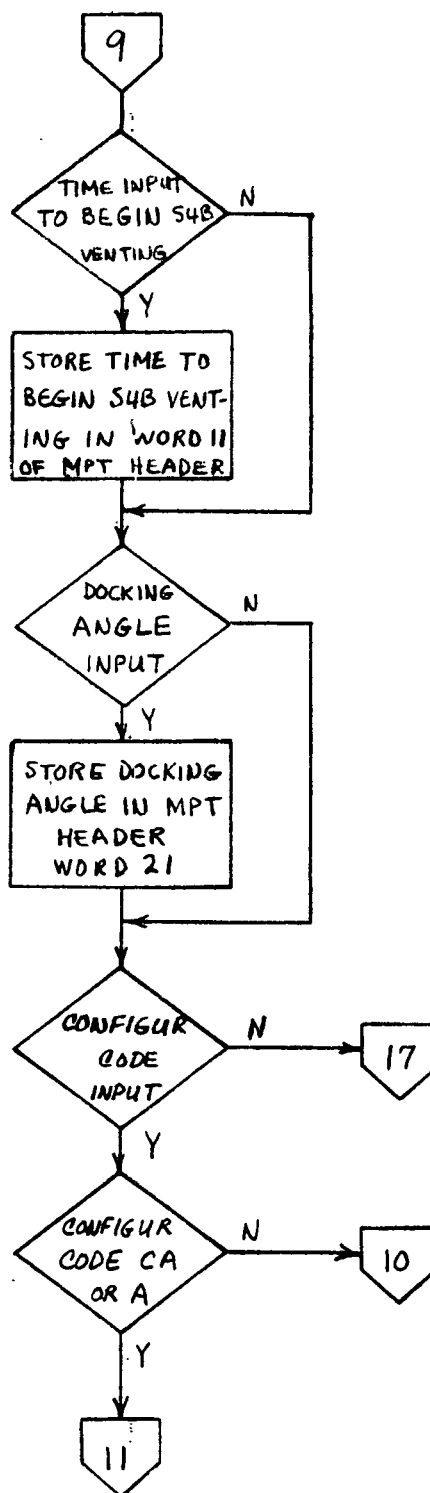


Figure PMMWTC-1. Flowchart

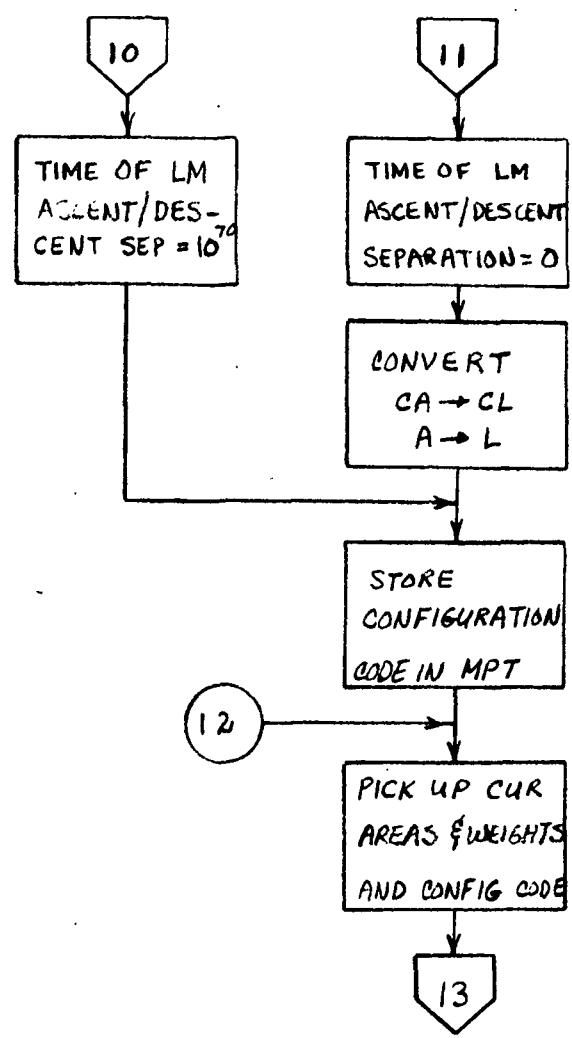


Figure PMMWTC-1. Flowchart

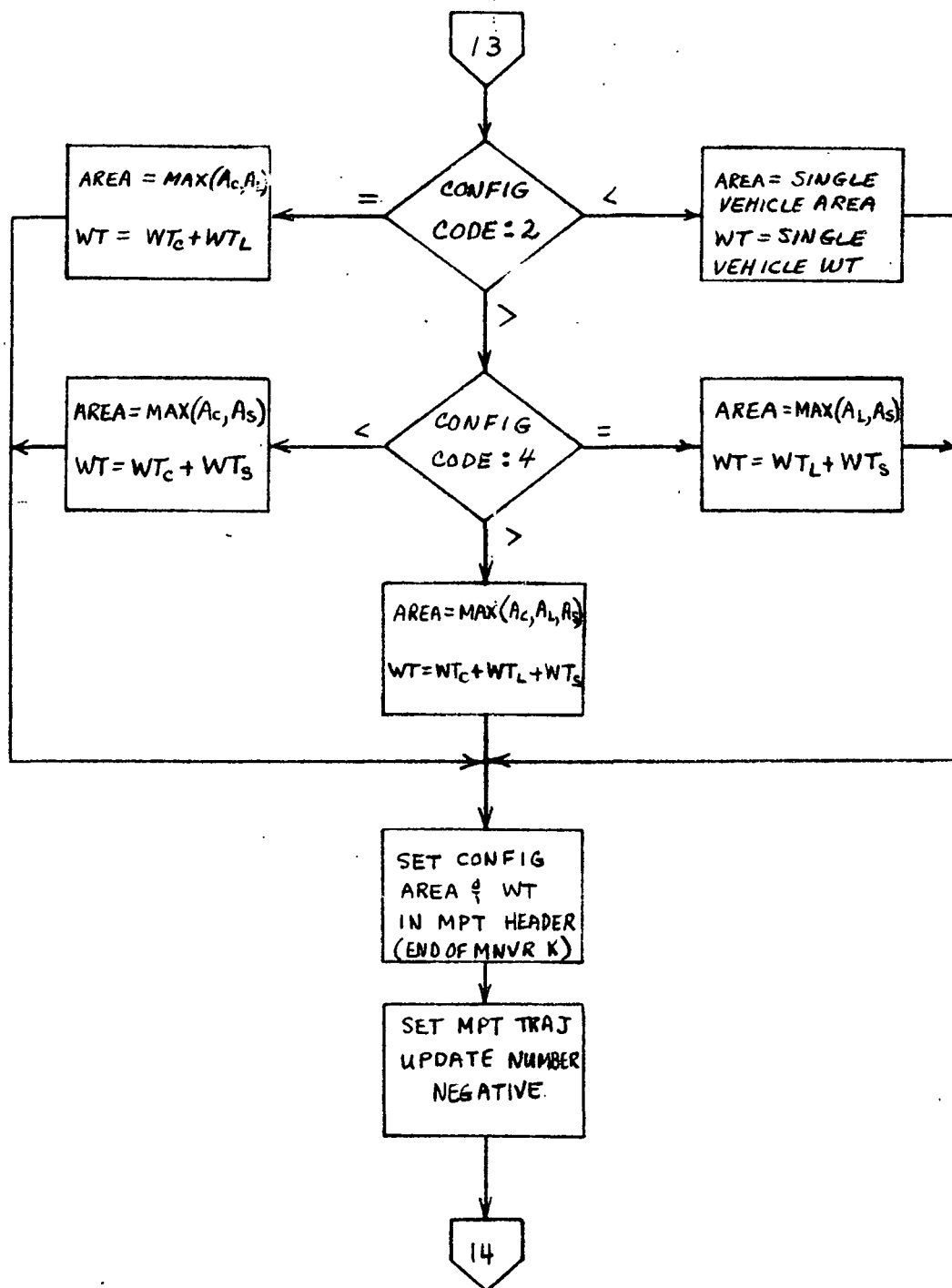


Figure PMMWTC-1. Flowchart

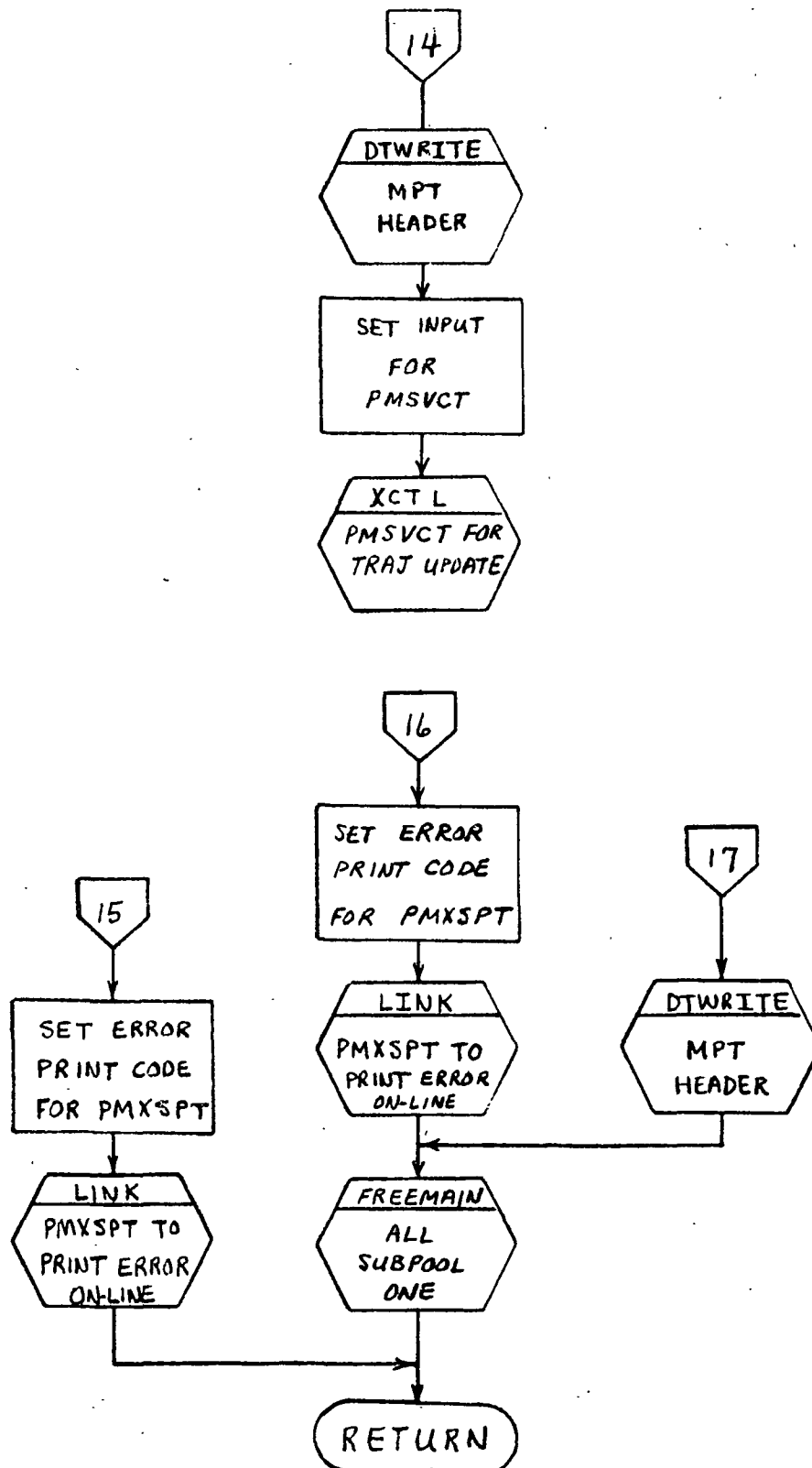


Figure PMMWTC-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS-AS-200 Mission Planning

2. PMQMED
DATE 8/24/67 (206)
REV
PAGE 1 of 4

- a. Symbol — PMQMED
- b. Program Name — MED decoder subprogram for K & M MEDs
- c. Programming Group — Mission Planning
- d. Programmer — E. Kramer
- e. Language — Assembler
- f. Data Tables Used — None
- g. MED's Used — All K & M codes
- h. Writeup Completion Date — 8/4/67
- i. System Parameters — Table PMQMED-1
- j. Flowchart — Figure PMQMED-1

FUNCTION

PMQMED checks the validity of all K & M MED's and sets up data for routing to proper destination

USAGE

- a. Calling Sequence — LINKed to by GMGMED
- b. Inputs — All items input on MED
- c. Outputs and Exit — Properly formatted data for routing
- d. Other Programs Used — GMGMED
- e. Data Restrictions and Limits — See MED formats.
- f. Error — See MED formats.

METHOD

See write-up for GMGMED for subprogram methods.

Approval	Approval
<i>E. Kramer</i>	<i>S.E. James</i>

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS-AS-200 Mission Planning

 2. PMQMED
 DATE 8/24/67 (206)
 REV
 PAGE 2 of 4

Table PMQMED -1. System Parameters

Symbol in Program	Symbol In Master Tape	Type *	Units	Comments
TWOPI	MCMTPPI	D	-	2π
CID60	MCCHPM	D	hr./min.	Hours per minute
CID3600	MCCHPS	D	hr./sec.	Hours per second
CARDEG	MCCRPD	D	rad./deg.	Radians per Degree
CERHRFS	MCCEHF	D	Er/hr. ft/sec.	Earth radii per hour/feet per second
CCULNM	MCCCUL	D	Cul/nm	Canatical unit of length per nautical mile
GMTLO	MCGMTL	D	hrs.	Greenwich mean time of lift-off
HALFPI	MCMHPI	D	-	$\pi/2$
PI	MCMPIR	D	-	π

*Type
 H - Fixed-point half word
 F - Fixed-point full word
 E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal

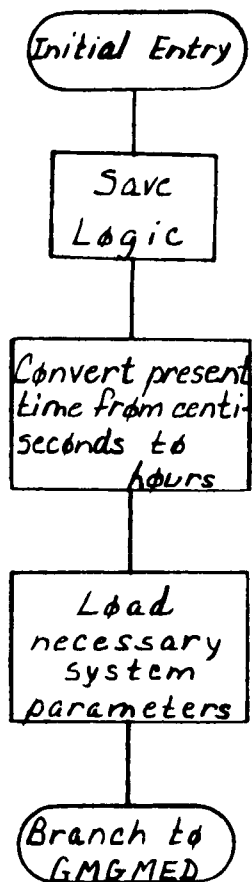


Figure PMQMED - 1 Flowchart

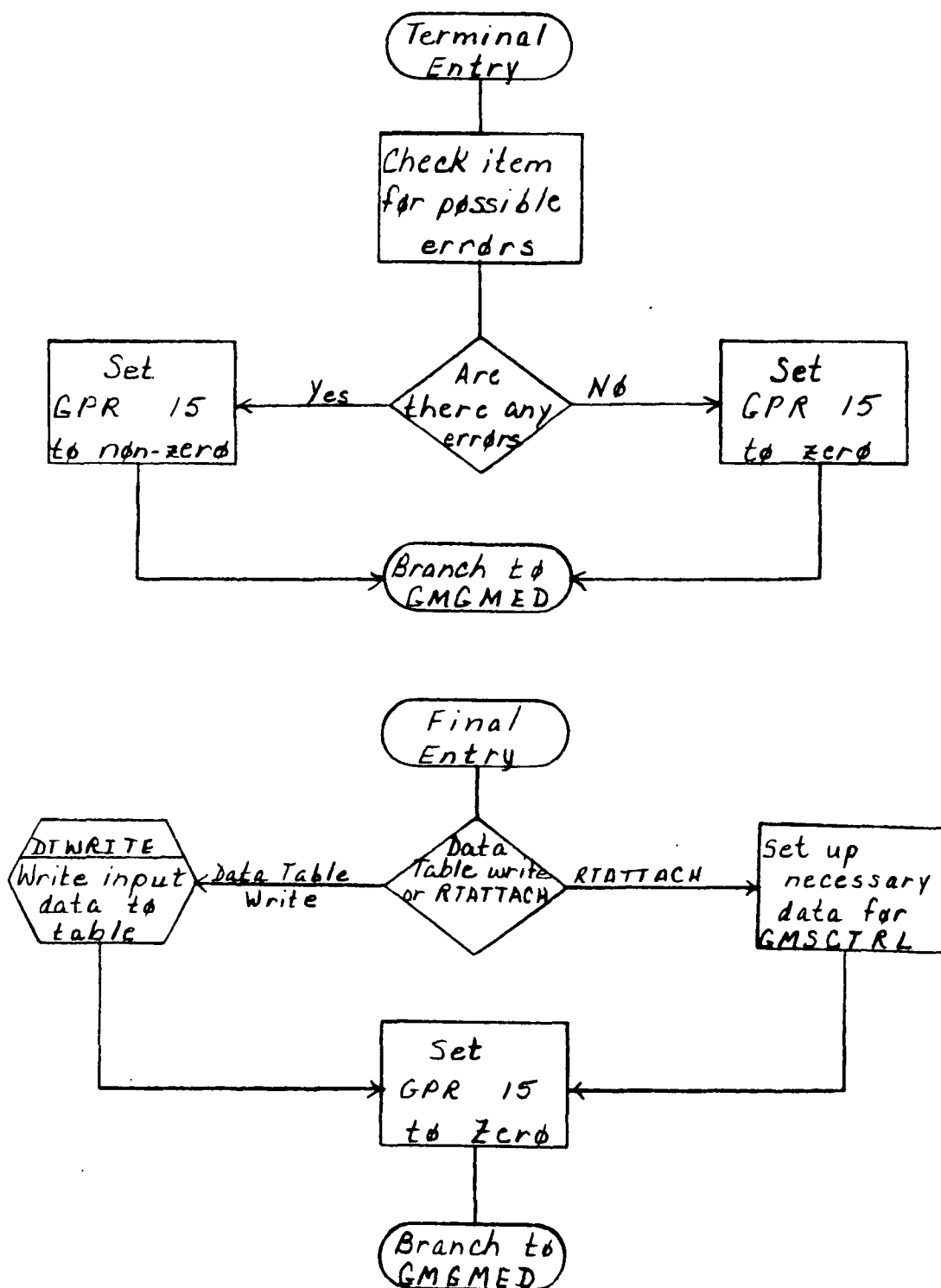


Figure PMQMED - 1 Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

3. PMSEXE
DATE 11/22/66 (206)
REV
PAGE 1 of 7

3. PMSEXE

- a. Symbol — PMSEXE
- b. Program Name — Maneuver Execution Program
- c. Programming Group — Mission Planning
- d. Programmer(s) — W. E. Sherman
- e. Language — Assembler
- f. Data Tables Used — PZMPTCSM, PZMPTLEM, PZMPTS4B
- g. MED's Used — None
- h. Writeup Completion Date — 10/14/66
- i. Parameter List — Tables PMSEXE - 1 through - 4.
- j. Flowchart — Figure PMSEXE-1.

FUNCTION

PMSEXE, after the execution of each maneuver in the Mission Plan Table, prints the maneuver execution vector on line and updates the last executed maneuver counter in the Mission Plan.

USAGE

- a. Calling Sequence — RTATTACH TASK = PTMANC, EP = PMSEXE, ID = 27
- b. Inputs — Present time from the RTATTACH
- c. Outputs and Exit — Maneuver execution vector on line and updates quantities in Mission Plan Table.
- d. Other Programs Used — GMSCTRL and PMXSPT
- e. Data Restrictions and Limits — None
- f. Error — Illegal entry - causes on-line print.

METHOD

PMSEXE request time traps at the beginning of each maneuver in the Mission Plan Table by the execution of the DTRROUTE macro. Upon the receipt of these time traps, the appropriate maneuver execution vector will be printed on line.

Approval	Approval
W ^m E. Sherman	<i>[Signature]</i>

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 3. PMSEXE
 DATE 11/22/66 (206)
 REV
 PAGE 2 of 7

Table PMSEXE - 1. Parameter List

ID — 12

QNAME — GQ3CTL

From — PMSEXE

To-Task — GTMSCTRL

Purpose — Release GMSCTRL

EP — GMSCTRL

Byte No.	Type*	Units	Value / Description / Comments

Table PMSEXE - 2. Parameter List

ID — 35

QNAME —

From — PMSEXE

To-Task — PTMANC

Purpose — Illegal entry print

EP — PMXSPT

Byte No.	Type*	Units	Value / Description / Comments

*Type

H — Fixed-point half word
F — Fixed-point full wordE — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

3. PMSEXE

DATE 11/22/66 (206)

REV

PAGE 3 of 7

Table PMSEXE - 3. Parameter List

ID — $27 + 2^8 * n^*$

QNAME —

From — GMSCTRL

To-Task — PTMANC

Purpose — Maneuver execution timeout for
vehicle nEP — PMSEXE
* vehicle number

Byte No.	Type*	Units	Value / Description / Comments

Parameter List

ID — 36

QNAME —

From — PMSEXE

To-Task — PTMANC

Purpose — Print maneuver execution vector
on line

EP — PMXSPT

Byte No.	Type*	Units	Value / Description / Comments
0-3	X	--	Vehicle name
4-11	D	hr	Time of vector
12-19	D	Er	X-Component of position
20-27	D	Er	Y-Component of position
28-35	D	Er	Z-Component of position
36-43	D	Er/hr	X-Component of velocity
44-51	D	Er/hr	Y-Component of velocity
52-59	D	Er/hr	Z-Component of velocity
60-67	D	lbs.	Vehicle weight

*Type
H — Fixed-point half word
F — Fixed-point full word

E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

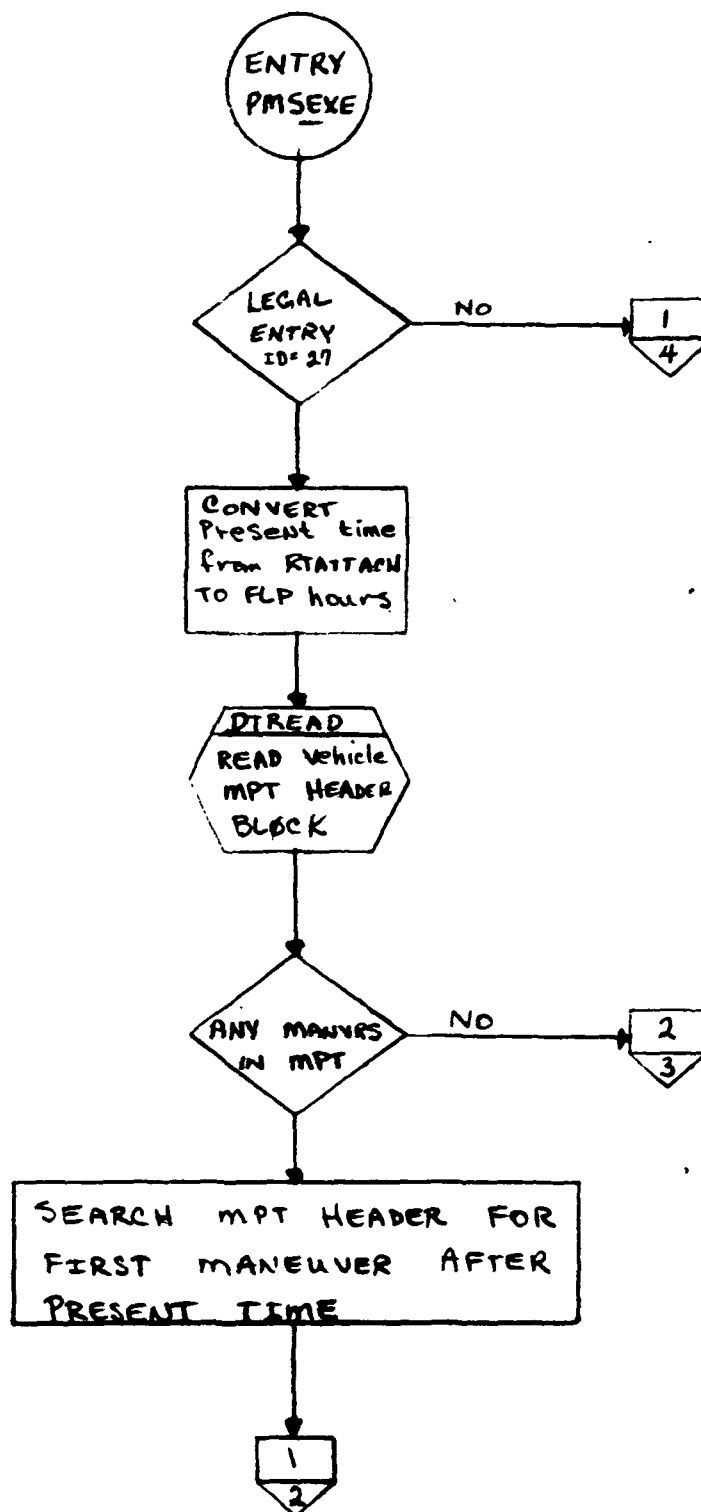


Figure PMSEXE - 1. Flowchart

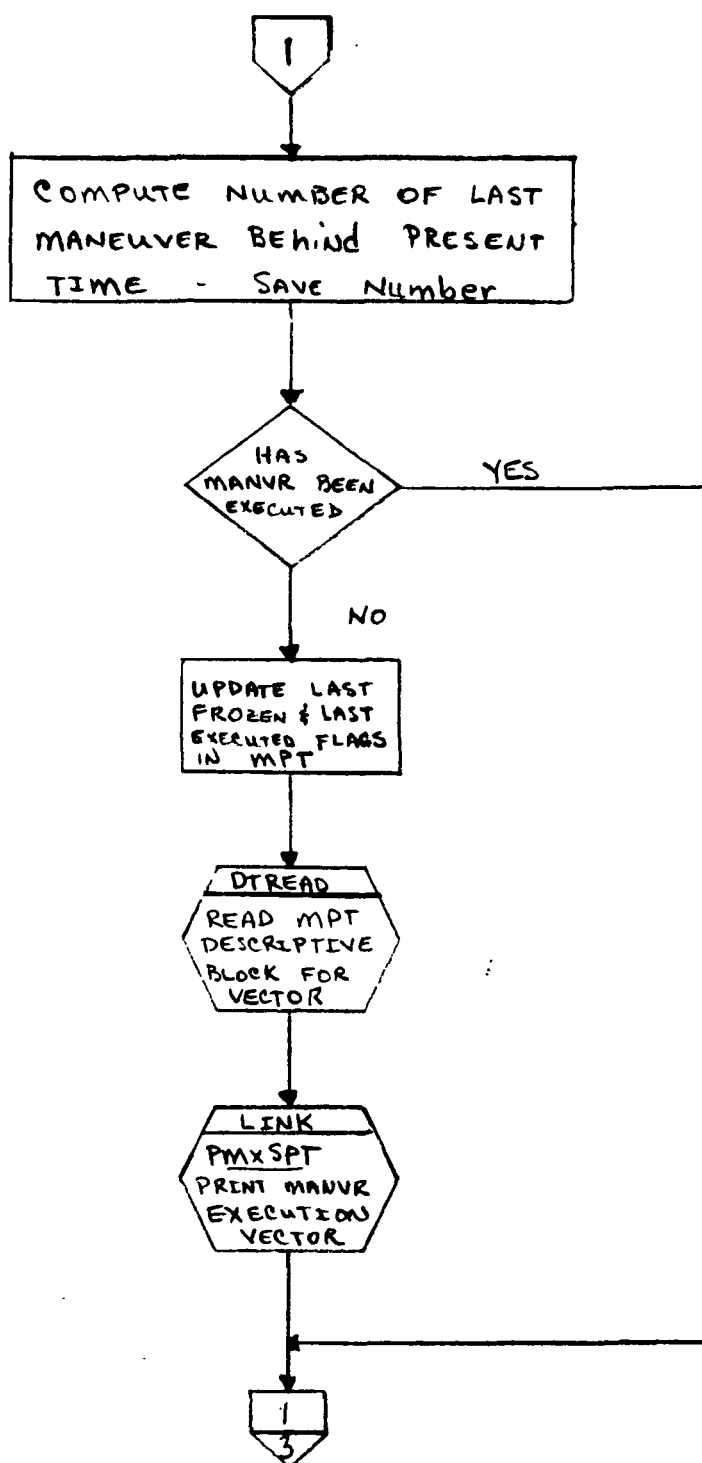


Figure PMSEXE - 1. Flowchart

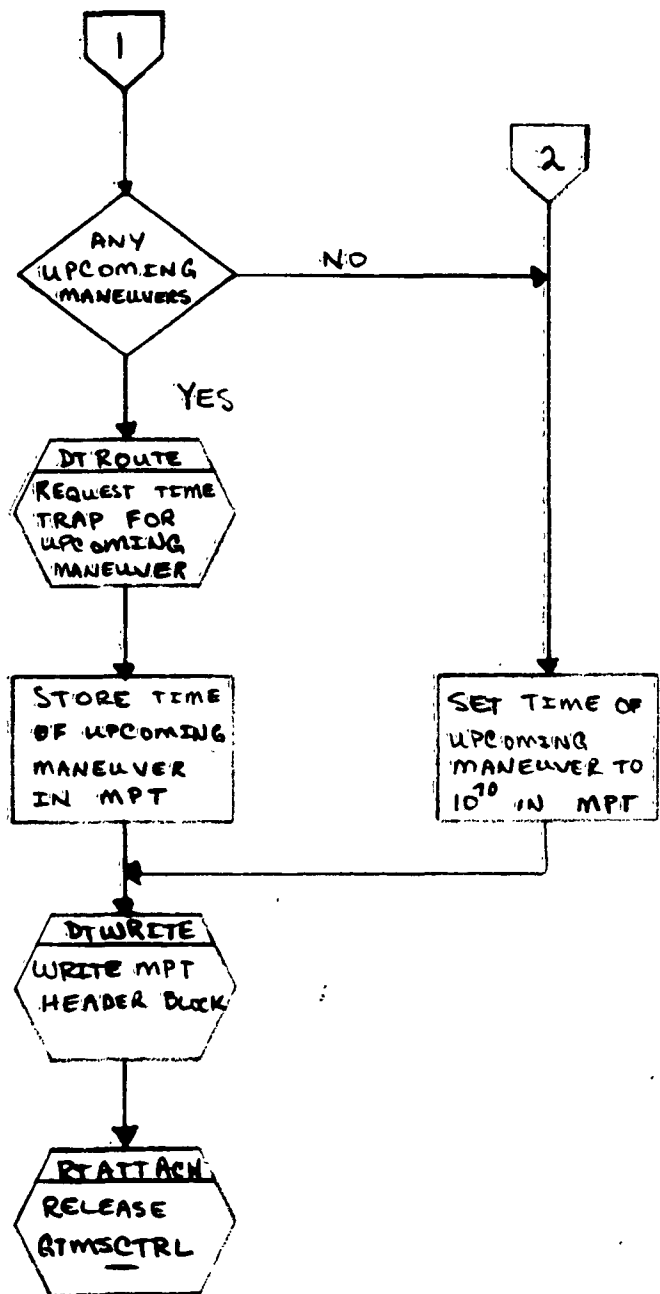


Figure PMSEXE - 1. Flowchart

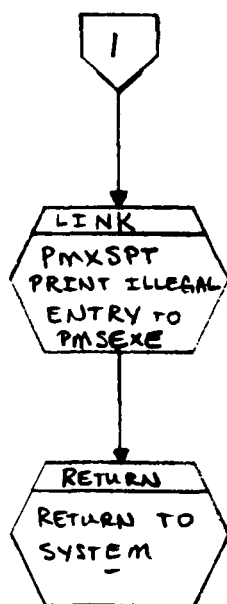


Figure PMSEXE - 1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMSTICN -
PCMTIS
DATE 7/28/67 (EORP)
REV
PAGE 1 of 9

- a. Symbol — PMSTICN - PCMTIS
- b. Program Name — Two-Impulse Impulsive Maneuver Csect
- c. Programming Group — Mission Planning
- d. Programmer (s) — G. Deaton, T. Hickey
- e. Language — FORTRAN IV
- f. Data Tables Used — None
- g. MED's Used — None
- h. Writeup Completion Date — 10/31/66
- i. Flowchart — Figure PMSTICN - PCMTIS-1.

FUNCTION

When given a chaser transfer initiation point and a transfer completion point, PCMTIS computes chaser impulsive osculating elements before and after the transfer initiation point, and outputs them in standard elements table format. Associated data that belongs in the elements table is also computed.

The transfer initiation point is defined in terms of a time. The transfer completion point is defined by a time and phase angle and height difference offsets of the chaser with the target at the time.

USAGE

- a. Calling Sequence — CALL PCMTIS
COMMON/TIS/AEGTAB (55), TCT, Δh_f , θ_f , DUMMY (2), SQRMU, ELETAB (40). All are double words.
- b. Inputs —
 - 1. AEGTAB - Analytic Ephemeris Generator header and initialized chaser and target blocks in standard format. The chaser should be at transfer initiation time. All indicator settings will remain unchanged by PCMTIS.
 - 2. TCT - Transfer completion time in hours
 - 3. Δh_f - Height difference offset in Earth radii.
 $\Delta h_f > 0$ if the chaser is below the target.
 - 4. θ_f - phase angle offset in radians.
 $\theta_f > 0$ if the chaser is behind the target.
 - 5. DUMMY - Not used by PCMTIS
 - 6. SQRMU - $\sqrt{\mu}$

Approval	Approval
<i>G. Deaton</i> 30 JUN	<i>T. Hickey</i>

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMSTICN -
PCMTIS
DATE 7/28/67 (EORP)
REV
PAGE 2 of 9

c. Outputs and Exit —

1. ELETAB - Two-block elements table, in standard format, for transfer initiation and transfer completion maneuvers. Exit is by RETURN statement. The actual phase angle and height difference at TCT are returned in the target AEG block.

d. Other Programs Used — PCZYCF, PCPASVEL (POSVEL), and standard library subroutines.

- e. Error — ELETAB (1) = 0, No solution was obtained.

METHOD

The two-impulse processor uses a conic fit solution of Lambert's equations subroutine in an iterative procedure which produces transfer initiation Keplerian elements. These elements put the chaser vehicle into a trajectory that will intercept a desired point in space at the given transfer completion time. Chaser Keplerian coelliptic elements are produced at the transfer completion time. Specified phase angle and height difference offsets, if any, are considered.

The iterative procedure uses a fake target point that is refined to produce the desired target point. A first guess fake target point is computed using a Keplerian approximation. The actual target point is obtained by advancing the target vehicle to the transfer completion time and computing the target inertial position vector, which takes into account any given phase angle and height difference offsets.

Using the data associated with the first fake target point, the conic fit subroutine computes a semimajor axis (a), eccentricity (e), time of transfer beyond perigee (TPTCU), and true anomaly (f), which are used to compute the inclination (i), mean anomaly (l), argument of perigee (g), and longitude of ascending node (h) for the chaser vehicle after the transfer initiation maneuver. These impulsive elements are input to the Analytic Ephemeris Generator (AEG), which initializes them and advances them with drag and harmonics to the transfer completion time.

If the inertial position of the chaser vehicle obtained at transfer completion time is not within the convergence criterion specified, the fake target point is refined using the position difference between the actual transfer completion point and the desired one.

The chaser elements before the transfer initiation maneuver are restored, a new a, e, TPTCU, and f are obtained from the conic fit subroutine, from which are computed the new i, l, g, and h after the maneuver.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMSTICN -
PCMTIS

DATE 7/28/67 (EORP)

REV

PAGE 3 of 9

The newelements are again advanced to transfer completion time, convergence check made, and iterations continued until convergence is obtained or the iteration counter is exceeded, in which case the actual phase angle and height difference are computed by the AEG and compared with the rendezvous tolerances of .04 degrees phase difference and 3000 feet height difference. If these tolerances are met, the logic follows the same path as if the iterative process had converged. If the tolerances are not met, an error indicator is output. If the iterative process converges, Keplerian elements before and after each of the two maneuvers are output.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMSTICN-
PCMTIS
DATE 7/28/67 (EORP)
REV
PAGE 4 of 9

INPUTS:

- ① T_i - TRANSFER INITIATION TIME
- ② CHASER ELEMENTS AT T_i AND TARGET ELEMENTS
- ③ T_F - TRANSFER COMPLETION TIME
- ④ Δh_F - HEIGHT DIFFERENCE WANTED AT T_F
- ⑤ θ_F - PHASE ANGLE WANTED AT T_F

SUBSCRIPT "C" REFERS TO CHASER, "T" TO TARGET

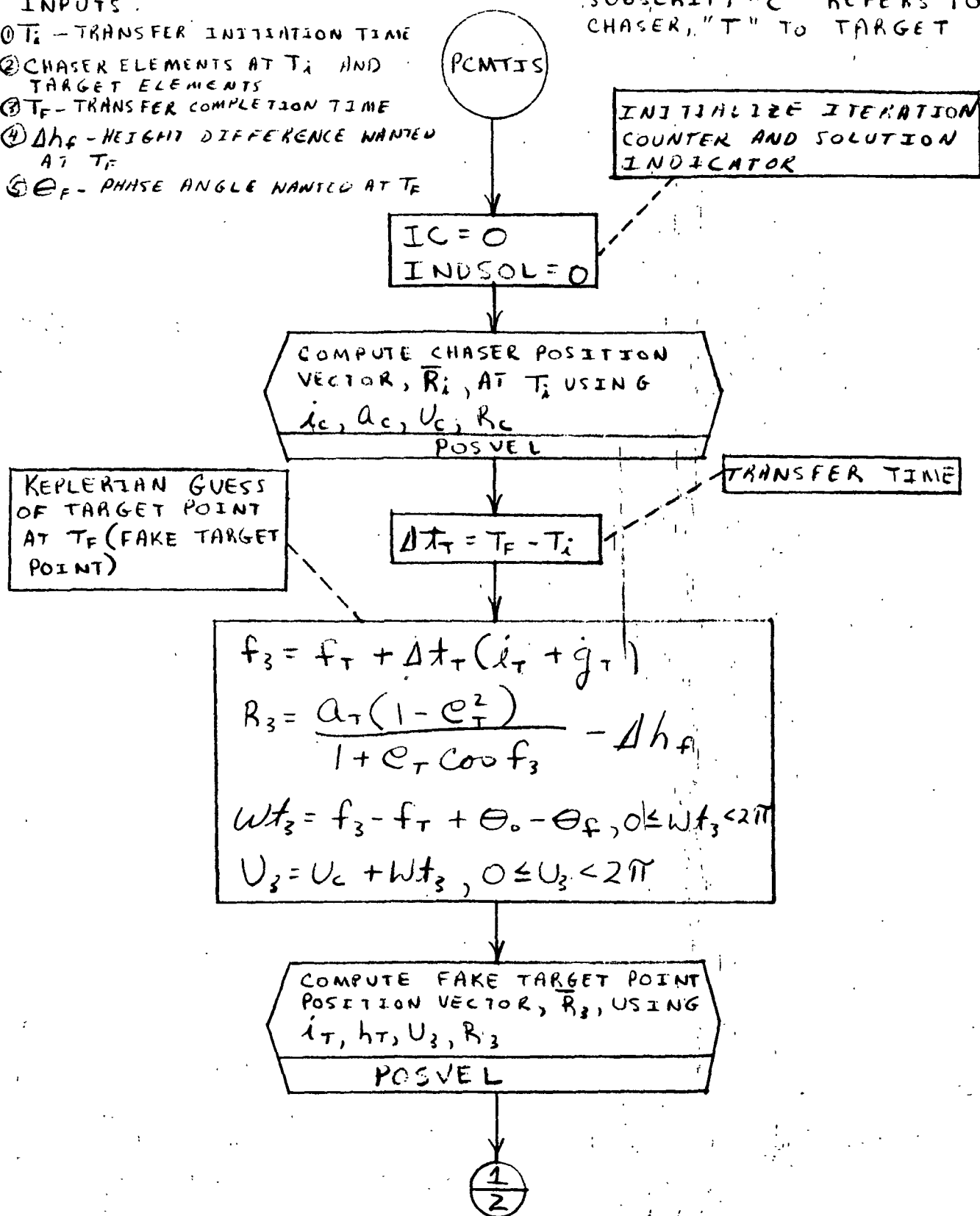
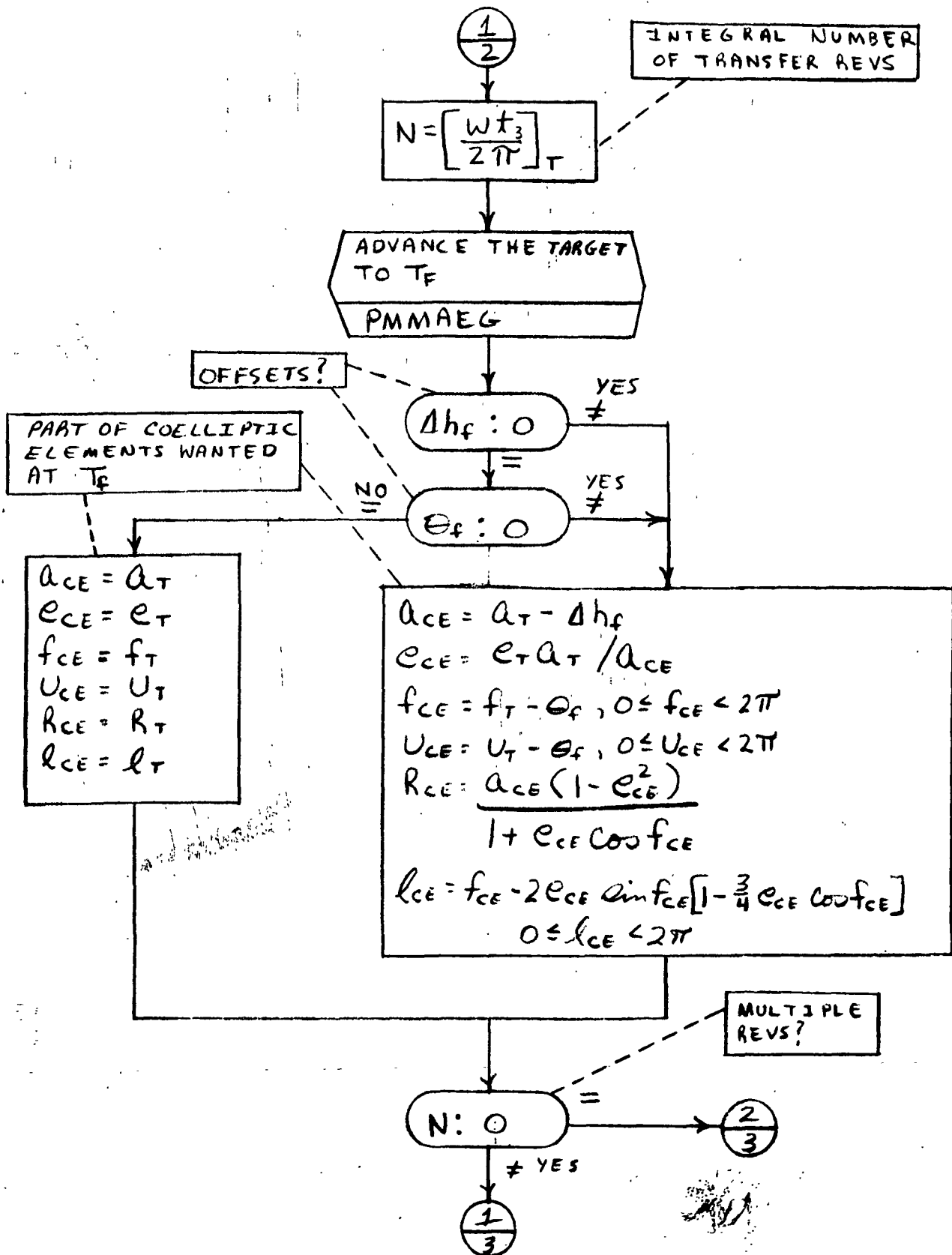


Figure PMSTICN - PCMTIS-1. Flowchart.
(Sheet 1 of 6)

Figure PMSTICN - PCMTIS-1. Flowchart
(Sheet 2 of 6)

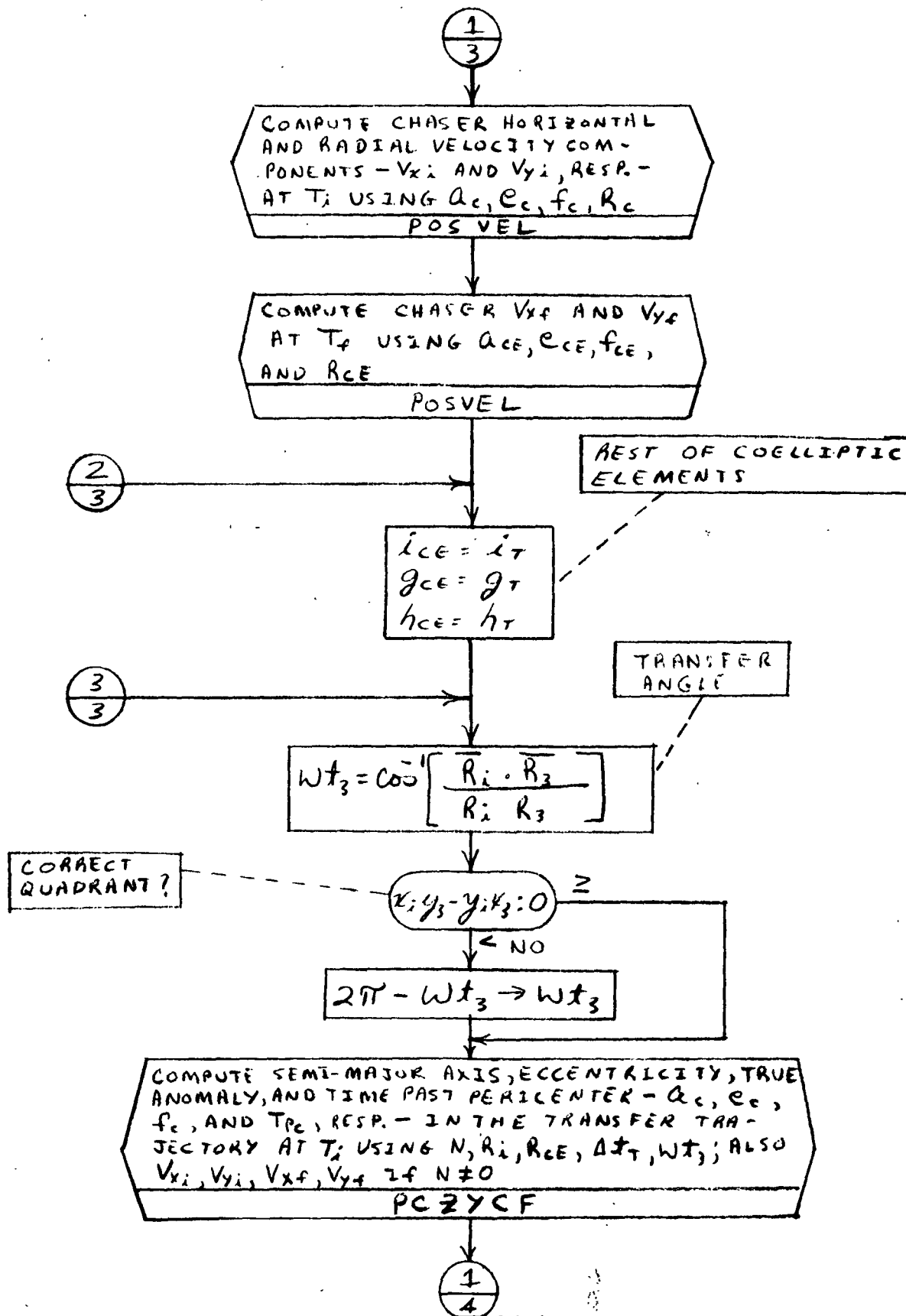


Figure PMSTICN - PCMTIS-1. Flowchart

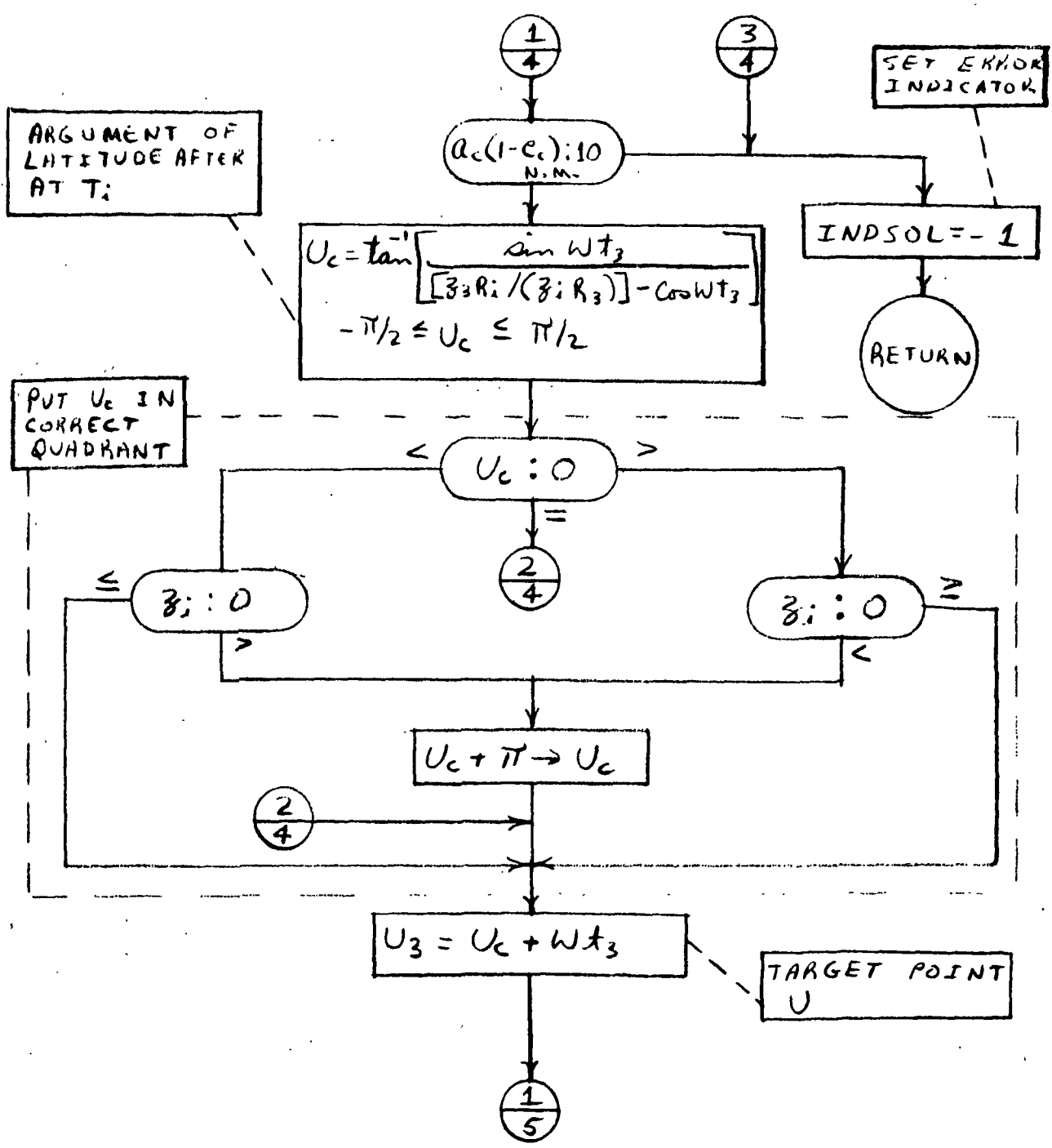


Figure PMSTICN - PCMTIS-1. Flowchart
(Sheet 4 of 6)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMSTICN -
PCMTIS

DATE 7/28/67 (EORP)

REV

PAGE 8 of 9

CHASER
TRANSFER
ELEMENTS
AT T_i

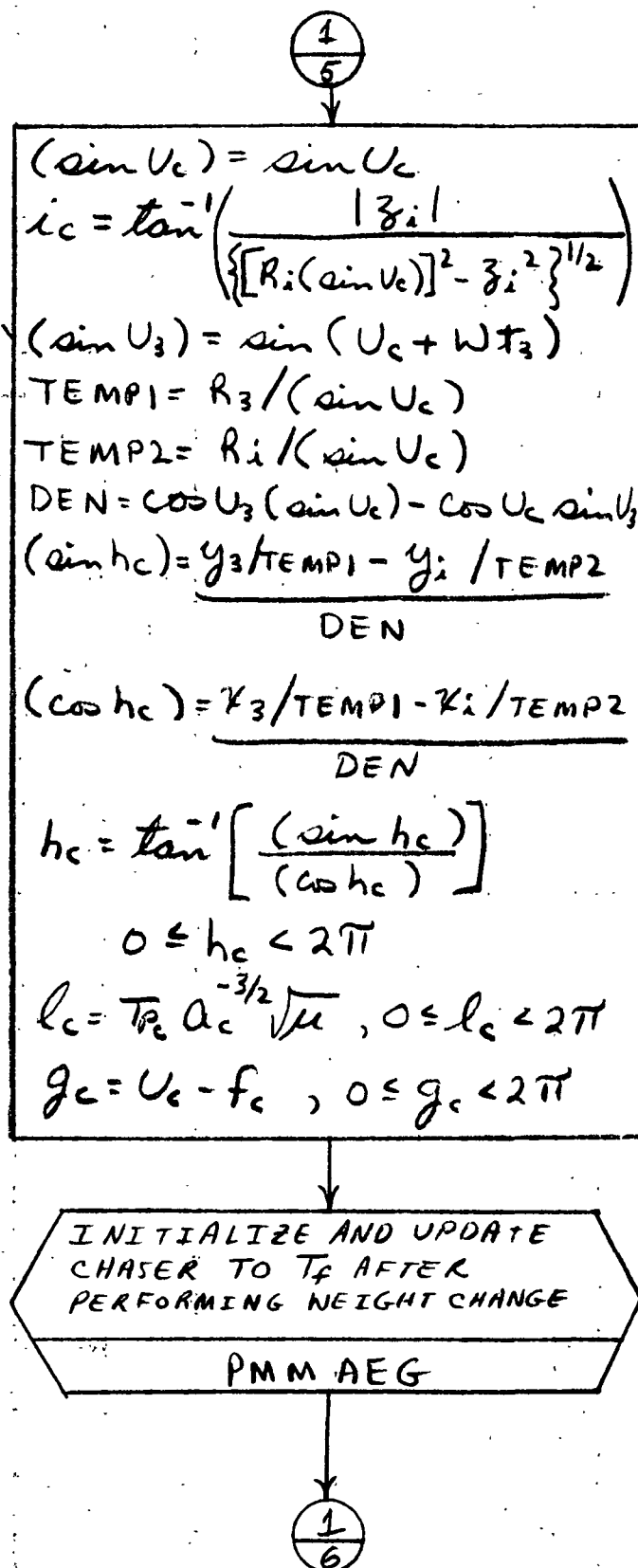


Figure PMSTICN - PCMTIS-1. Flowchart

(Sheet 5 of 6)

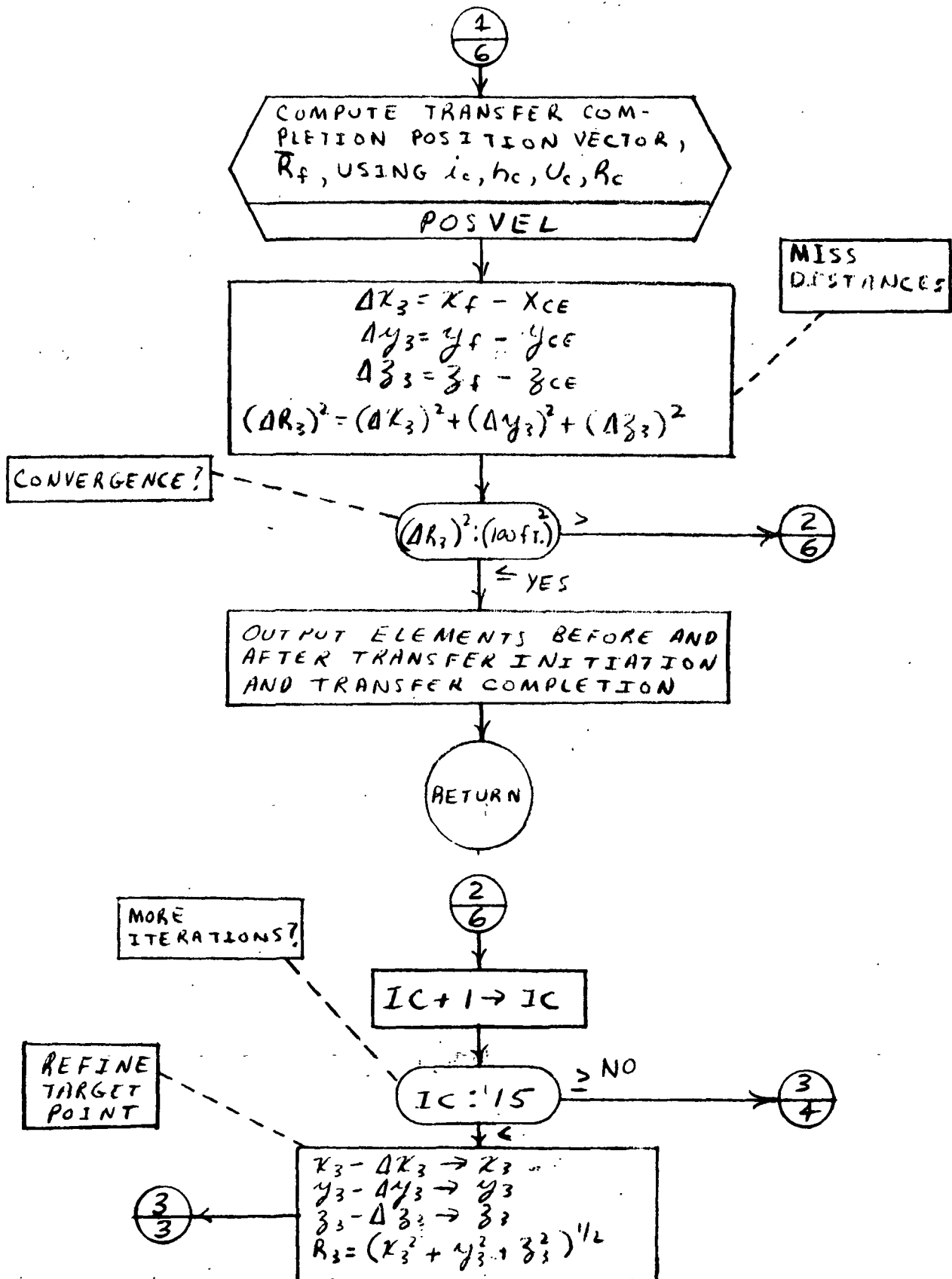


Figure PMSTICN - PCMTIS-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

DATE 12/13/67(EORP)

REV

PAGE 1 of 33

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

- a. Symbol — PMSTICN-PCSTICN
- b. Program Name — Two-Impulse Control Logic
- c. Programming Group — Rendezvous Planning
- d. Programmer — T. Hickey
- e. Language — Assembler
- f. Data Tables Used — PZTIPREG, PZTIPCCD, PZMYSAVE, GZGENCSN
- g. MED's Used — K30, K31, K32, P51, P52
- h. Writeup Completion Date — 11/17/67
- i. System Parameters — Table PMSTICN-PCSTICN-1
- j. Parameter List — Table PMSTICN-PCSTICN-2 through 6
- k. Flowchart — Figure PMSTICN-PCSTICN-1

FUNCTION

This section of the Two-Impulse Load Module has the control logic responsibilities for the entire unit. All MED requests are routed to the CSECT for control of all calculations using the math CSECT's PCMTIS and PCZYCF. In addition to the control logic, the math formulation for the elevation angle computation, the impulsive pitch and yaw computations, and several corrective combination computations are included in this CSECT.

USAGE

- a. Calling Sequence — All MED requests = XCTL EP = PMSTICN
PMMDKI, a standard LINK EP = PMSTICN is used
- b. Inputs — MED parameters, chaser and target vehicle AEG blocks; times of both maneuvers, and desired offsets
- c. Outputs and Exit — Data tables PZTIPREG and PZTIPCCD for MED requests, and two impulsive maneuver blocks for other requests —
Exit for MED M72 = XCTL to PMMXFR; all other exits by a 'RETURN' macro
- d. Other Programs Used — Day-night subroutine (PMMDAN), GIMKIC, PCMTIS, AEG (PMMAEG), print logic (PMXSPT), single solution computations (PMMTISS), display supervisor (GMSNAP), system subroutines ATAN2, DSIN, DCOS, DSQRT, DARSIN, DATAN, DATAN2, DTAN, FDXPD=, GLQATN, and DARCOS
- e. Data Restrictions and Limits — None other than restrictions controlled by MED logic

Approval	Approval
<i>Thomas M. Hickey</i>	<i>A. Amietti</i>



- f. Error — The following error messages may be printed on line:
- "PMSTICN. NO NOMINAL OFFSETS FOR CORRECTIVE COMBINATION"
 - "PMSTICN. SOLUTION NUMBER NOT IN TABLE"
 - "PMSTICN. FAILED TO CONVERGE ON ELEVATION ANGLE"
 - "PMSTICN. AEG ERROR"
- All other error conditions have internal indicators to the appropriate logic.

METHOD

The method is divided into five parts as listed below. The following paragraphs provide an explanation for each of these parts.

- Multiple Solution Request
- Corrective Combination Request
- Single Solution Request
- Transfer Option
- DKI or SPQ Generate

a. Multiple Solution Request

This MED request has the ability to vary the time of either the first or second maneuver through some input time range by some input increment and compute a two-impulse solution for each case. The same desired offsets are used for all cases. This request also has the capability of letting the two-impulse unit determine the time of the first maneuver by use of an elevation angle search routine. The following paragraphs explain the method used in determining the time of arrival at the desired elevation angle.

It is assumed that the orbits between the chaser and target are co-elliptical with a small phase lag. It is further assumed that the elevation angle desired is reached at some point in the future. For example, if the chaser is below the target, it is not expected that the chaser would be required to pitch down to acquire the target; hence, if an elevation angle of $0 > \phi > -180^\circ$ is requested, no solution is obtainable. Assuming co-ellipticity and a small phase lag, the assumption is made that the relative displacements X and Y of the target with respect to the chaser, vary linearly with time.

Therefore,

$$\begin{aligned} X &= At + B \\ Y &= Ct + D \end{aligned}$$



It should be pointed out that X is measured in the direction of motion, Y is measured radially toward the Earth, and the elevation angle is measured posi-time up.

Thus:

$$-\frac{Y}{X} = \tan \varphi, \text{ where } \varphi \text{ is the elevation angle}$$

assuming

$$\varphi = \varphi_{REQ}$$

$$\frac{-Ct - D}{At + B} = \tan \varphi_{REQ} = \frac{\sin \varphi_{REQ}}{\cos \varphi_{REQ}}$$

$$(A \sin \varphi_{REQ} + C \cos \varphi_{REQ})t = -B \sin \varphi_{REQ} - D \cos \varphi_{REQ}$$

$$t = \frac{-B \sin \varphi_{REQ} - D \cos \varphi_{REQ}}{A \sin \varphi_{REQ} + C \cos \varphi_{REQ}}$$

The time "t" should be the time when the required look angle is reached; however, perturbations, and non-linearity with time in the relative positions do require an iterative technique to be employed.

After all solutions have been attempted, and the best 13 solutions of the requested range are found, the unit queues the Multiple Solution Display Display Load Module, PMDTIMP.

b. Corrective Combination Request

The corrective combination technique involves two different requests. The first is varying the time of the second maneuver at a particular height through an input range and selecting the lowest fuel expenditure at that height. A range of heights is also input. The phase angle computation for a given time at a given height is as follows:

Terms:

- | | | |
|------------------|---|--------------|
| K | = ratio of catch-up rate to height difference | |
| | 5.9853114 rad/hr/Er | |
| θ_{WSR} | = nominal NSR phase angle | } MED inputs |
| Δh_{WSR} | = nominal NSR height offset | |
| T_{WSR} | = nominal time of NSR maneuver | |

θ_{TMIN} = nominal phase angle for start time of time range
 $H\theta R$ = height to theta ratio
 θ_i = phase angle for particular solution
 T_{MIN} = minimum time of second maneuver

Computations:

$$\theta_{TMIN} = \theta_{WSR} = (T_{MIN} - T_{WSR}) K \cdot \Delta h_{WSR}$$

$$H\theta R = \frac{\Delta h_{WSR}}{\theta_{TMIN}}$$

The phase angle for the minimum time at each height is computed as:

$$\theta_i = \frac{\Delta h}{H\theta R}$$

The phase computation for the next time at the same height is:

$$\theta_{i+1} = \theta_i - \Delta \dot{u} \cdot \Delta \Delta t$$

where

$\Delta \dot{u}$ = $K \cdot \Delta h$ and $\Delta \Delta t$ is the time increment between each of the times of the second maneuver

The second option varies only the phase angle through as input range. A slip time limit on each side of terminal phase initiation time in input as well as the time increment between the positive and negative limit. The time of the second maneuver is fixed. The computation of the range of phase angles for the same time at each height is as follows:

Terms:

Δt = limit of slip time on each side of terminal phase
 $\Delta \Delta t$ = time increment between positive and negative limit

Computations:

$$\theta_{TMIN} = \theta_{WSR} - (T_{MIN} - T_{WSR}) K \cdot \Delta h_{WSR}$$

$$H\theta R = \frac{\Delta h_{WSR}}{\theta_{TMIN}}$$

$$\theta_i = \frac{\Delta h}{H\theta R}$$



Therefore:

$$\theta_{\text{MIN}} = \theta_i - \Delta t \cdot \dot{\Delta u} \quad \text{and}$$

$$\theta_{\text{MAX}} = \theta_i + \Delta t \cdot \dot{\Delta u}$$

The increment phase angle is then computed the same as the first option:

$$\theta_{i+1} = \theta_i - \dot{\Delta u} \cdot \Delta \Delta t$$

In both options, the lowest fuel expenditure at each height is saved, and a queue is sent to the Corrective Combination Display Load Module, PMDDTVCC.

c. Single Solution Request

This pass through the control logic recomputes the requested corrective combination or multiple solution case, and queues the load module PMMTIS, the single-solution load module, for further computations.

d. Transfer Option

The logic here is exactly the same as the single-solution, except that after the regeneration of the requested case, control is given to PMMXFR, the maneuver transfer logic area of the maneuver control unit.

e. DKI or SPQ Generate

For this option, all inputs to the unit are passed from the DKI control load module PMMDKI, and a two impulse solution is computed using zero offsets. The results of the computation are then passed back to PMMDKI for further use.

DTRROUTE, RTQCNTRL, RTATTACH Macros

Macro — RTATTACH TASK=GTISPLAY, EP=GMSNAP, ID=3,

PARAM = (Display, 4, FAKEOUT, 4)

Purpose — Update appropriate display

Table PMSTICN-PCSTICN-1. System Parameters

[illegible]

* Type
H — Fixed-point half word
F — Fixed-point full word
E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

2. PMSTICN-PCSTICN

DATE 12/13/67 (EORP)

REV

PAGE 7 of 33

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

Table PMSTICN-PCSTICN-2. Parameter List

QUEID — 100

QNAME —

From — PMSVEC

To-Task —

Purpose — Corrective Combination Computation
- K32 MED Request

EP — PCSTICN

Byte No.	Type*	Units	Value / Description / Comments
1	F		Code = 1-3 - Maneuvering (chaser) vehicle code
5	F		0 or 1 - Request Indicator
9	D	hrs.	Vector time for maneuvering vehicle
17	D	hrs.	Vector time for target vehicle
25	D	hrs.	Time of first (NCC) maneuver
33	D	Er	Minimum height offset
41	D	Er	Maximum height offset
49	D	Er	Height increment
57	D	hrs.	Minimum time of second (NSR) maneuver
65	D	hrs.	Maximum time of second (NSR) maneuver
73	D	hrs.	Time increment of second maneuver
81	D	hrs.	Limit of terminal phase slip time
401	D	Er	Semimajor axis of chaser vehicle
409	D		Eccentricity of chaser vehicle
417	D	rad.	Inclination angle of chaser vehicle
425	D	rad.	Mean anomaly of chaser vehicle
433	D	rad.	Argument of perigee of chaser vehicle
441	D	rad.	Ascending node of chaser vehicle
449	D	hrs.	Time of chaser elements
457	D		K-Factor from chaser MPT
465	D	sq. ft.	Area of chaser vehicle
473	D	lbs.	Weight of chaser vehicle
481	D		Station ID from chaser MPT
497-584			Same as bytes 401-488 but for target vehicle

*Type
H — Fixed-point half word
F — Fixed-point full word

E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

2. PMSTICN-PCSTICN

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

DATE 12/13/67 (EORP)

REV

PAGE 8 of 33

Table PMSTICN-PCSTICN-3. Parameter List

QUEID — 34

QNAME —

From — GMSCTRL

To-Task —

Purpose — Transfer of Multiple or Corrective EP — PCSTICN

Combination Solution M72 MED Request

Byte No.	Type*	Units	Value / Description / Comments
1	F		IND = 1 or 2 - Two-Impulse Table Indicator
5	F		NO. = 1-13 - Plan number for transfer

Table PMSTICN-PCSTICN-4. Parameter List

QUEID — 102

QNAME —

From — PMSVEL

To-Task —

Purpose — Multiple Solution Computation - EP — PCSTICN
K30 MED Request

Byte No.	Type*	Units	Value / Description / Comments
1	F		Code = 1-3 - Maneuvering (chaser) vehicle code
5	F		IND. = 0-2 - Type of Request Indicator
9	D	hrs.	Vector time for maneuvering vehicle
17	D	hrs.	Vector time for target vehicle
25	D	hrs.	Time of first maneuver
33	D	hrs.	Time of second maneuver
41	D	hrs.	Time increment of variable maneuver
49	D	hrs.	Time range of variable maneuver
401-584			Same input as QUEID 100

*Type

H — Fixed-point half word
F — Fixed-point full wordE — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal



RTCC APOLLO PROGRAMMING SYSTEMS

DATE 12/13/67 (EORP)

REV

PAGE 9 of 33

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

Table PMSTICN-PCSTICN-5. Parameter List

QUEID — 103

QNAME —

From — PMMDKI - DKI Control Logic

To-Task —

Compute terminal phase maneuvers

Purpose — for DKI or co-elliptic sequence plan EP — PCSTICN

Byte No.	Type*	Units	Value / Description / Comments
5	F		Address of AEG blocks for computations
9	F		Address of output from two-impulse

Table PMSTICN-PCSTICN-6. Parameter List

QUEID — 104

QNAME —

From — GMSCTRL

To-Task —

Compute single solution parameters

Purpose — K31 MED request

EP — PCSTICN

Byte No.	Type*	Units	Value / Description / Comments
1	F		Ind. = 1 or 2 - Two-Impulse Table Indicator
5	F		No. = 1 - 13 - Plan No. for transfer
9	F		No. = 2 or 4 - No. of firings quads used
13	F		Mode = 1 or 2 - Mode Indicator for single solution
17	D	rad.	Pitch angle
25	D	hrs.	Approach data time increment

*Type
H — Fixed-point half word
F — Fixed-point full word

E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

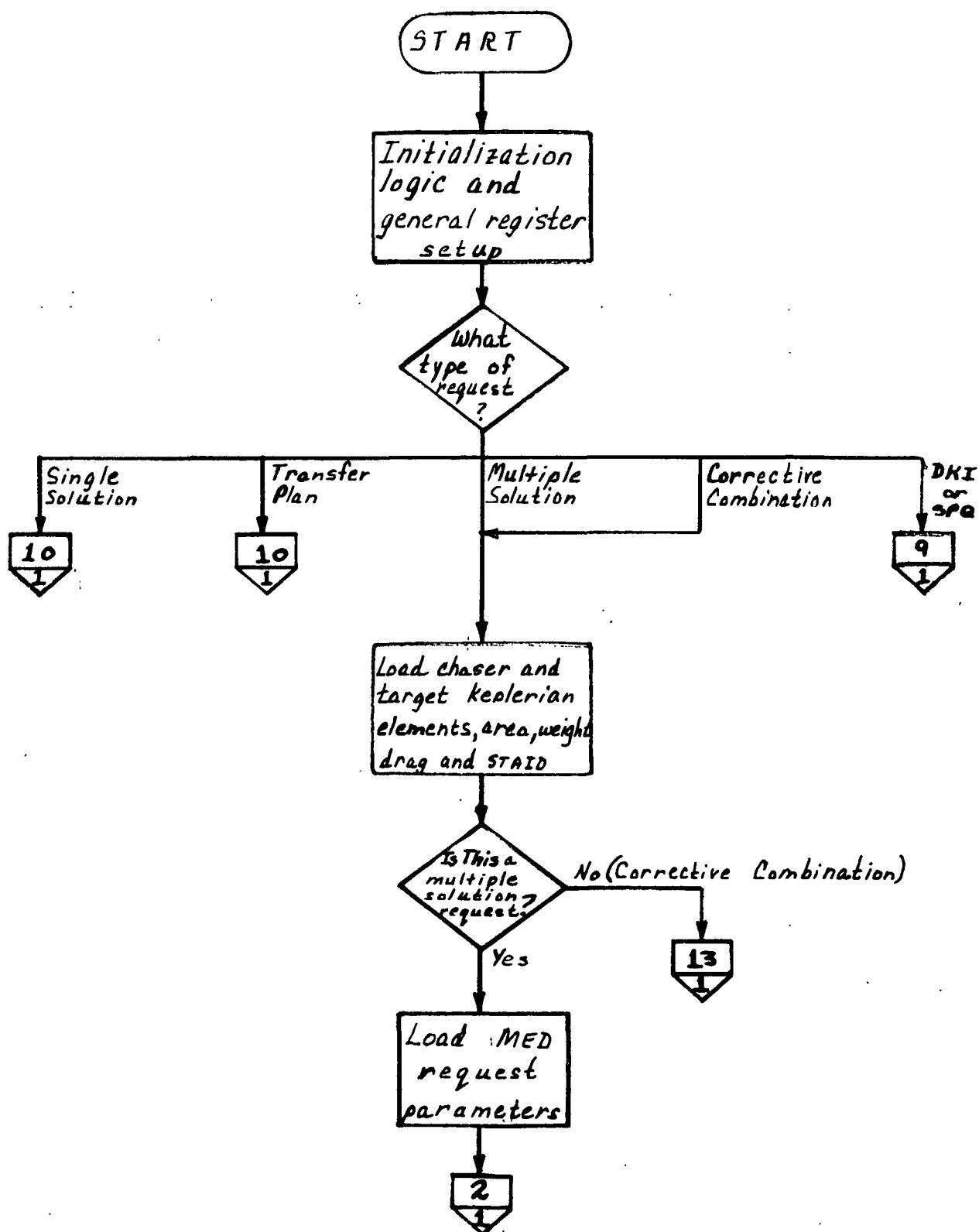


Figure PMSTICN-PCSTICN-1. Flowchart

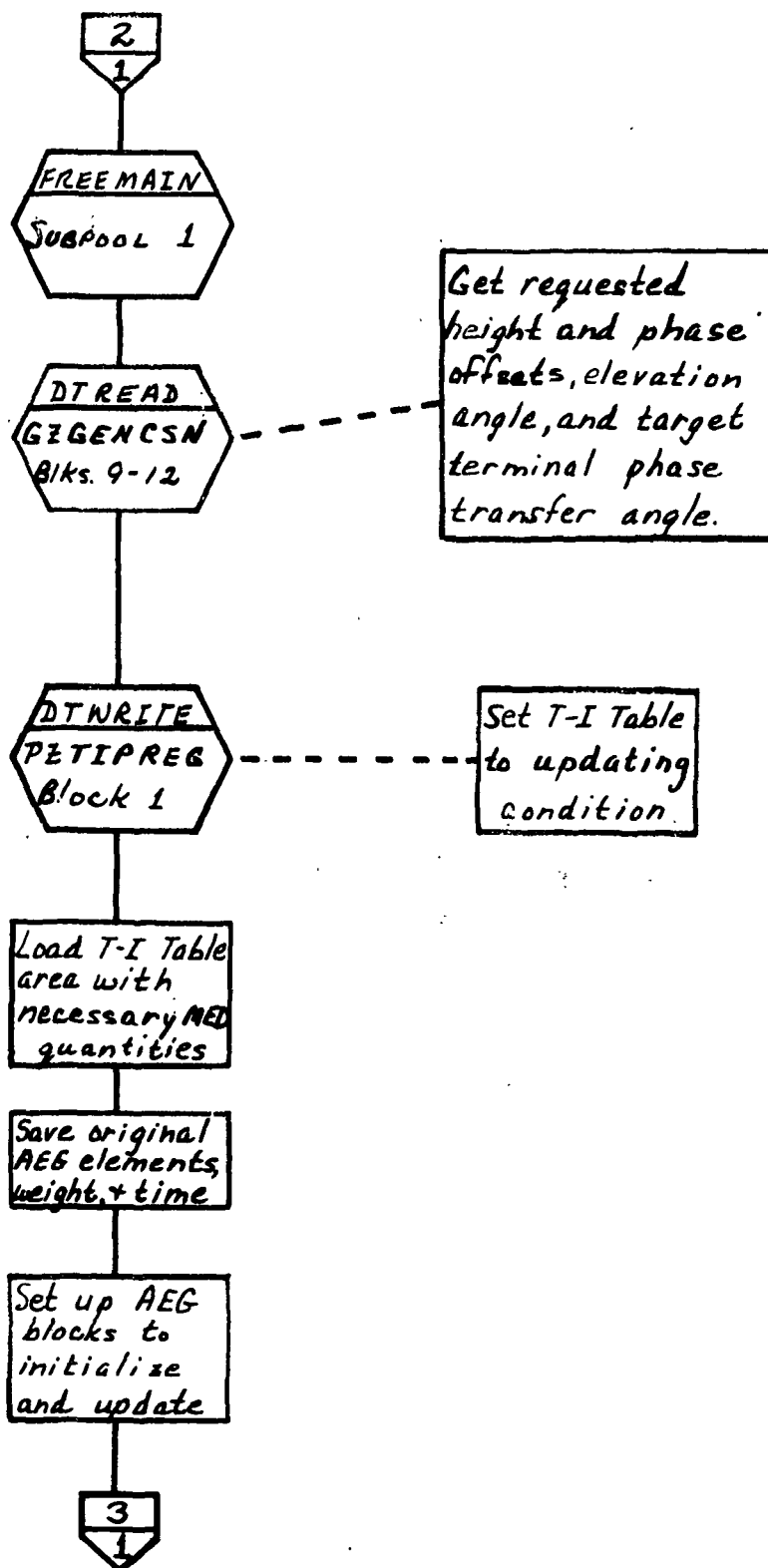


Figure PMSTICN-PCSTICN-1. Flowchart

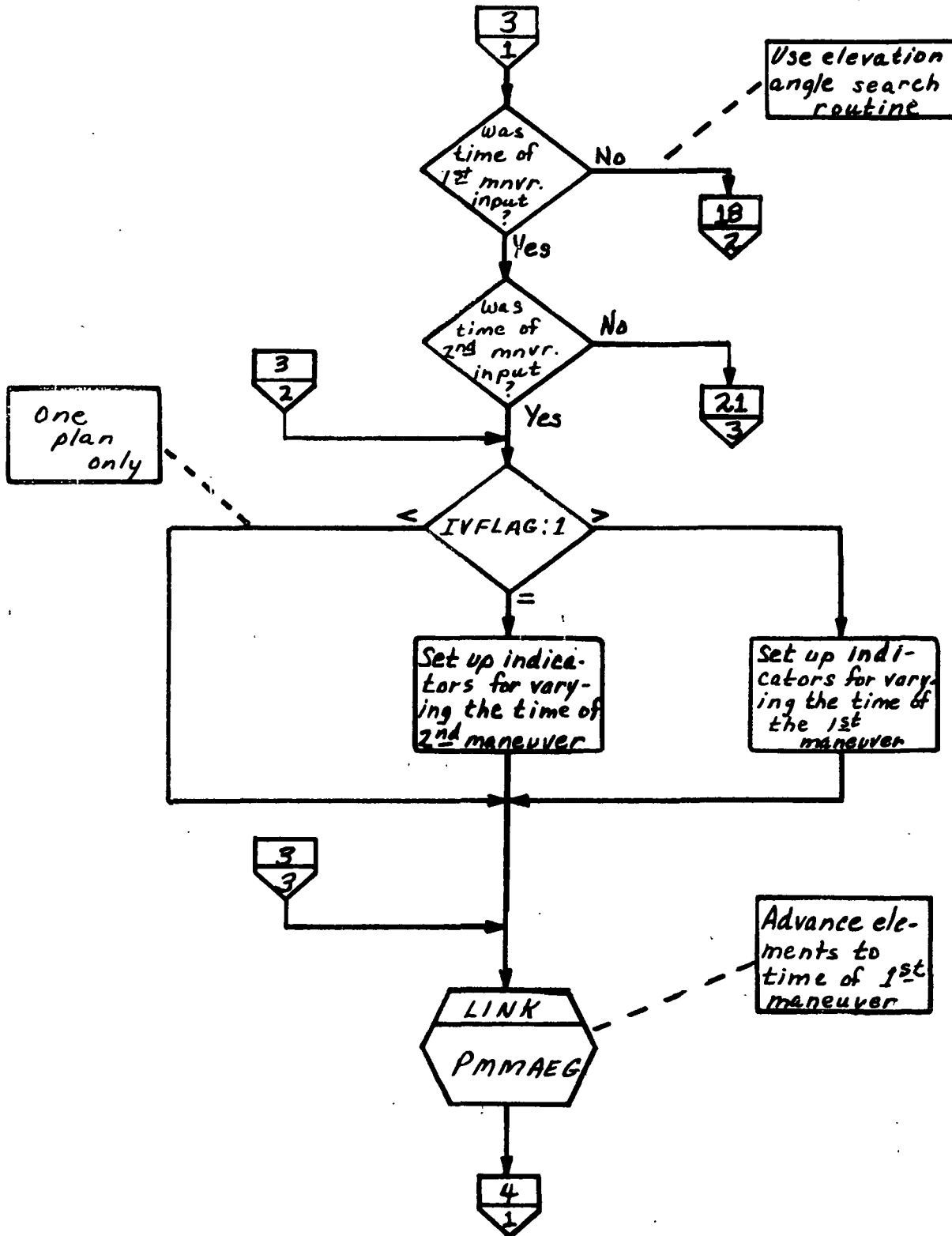


Figure PMSTICN-PCSTICN-1. Flowchart

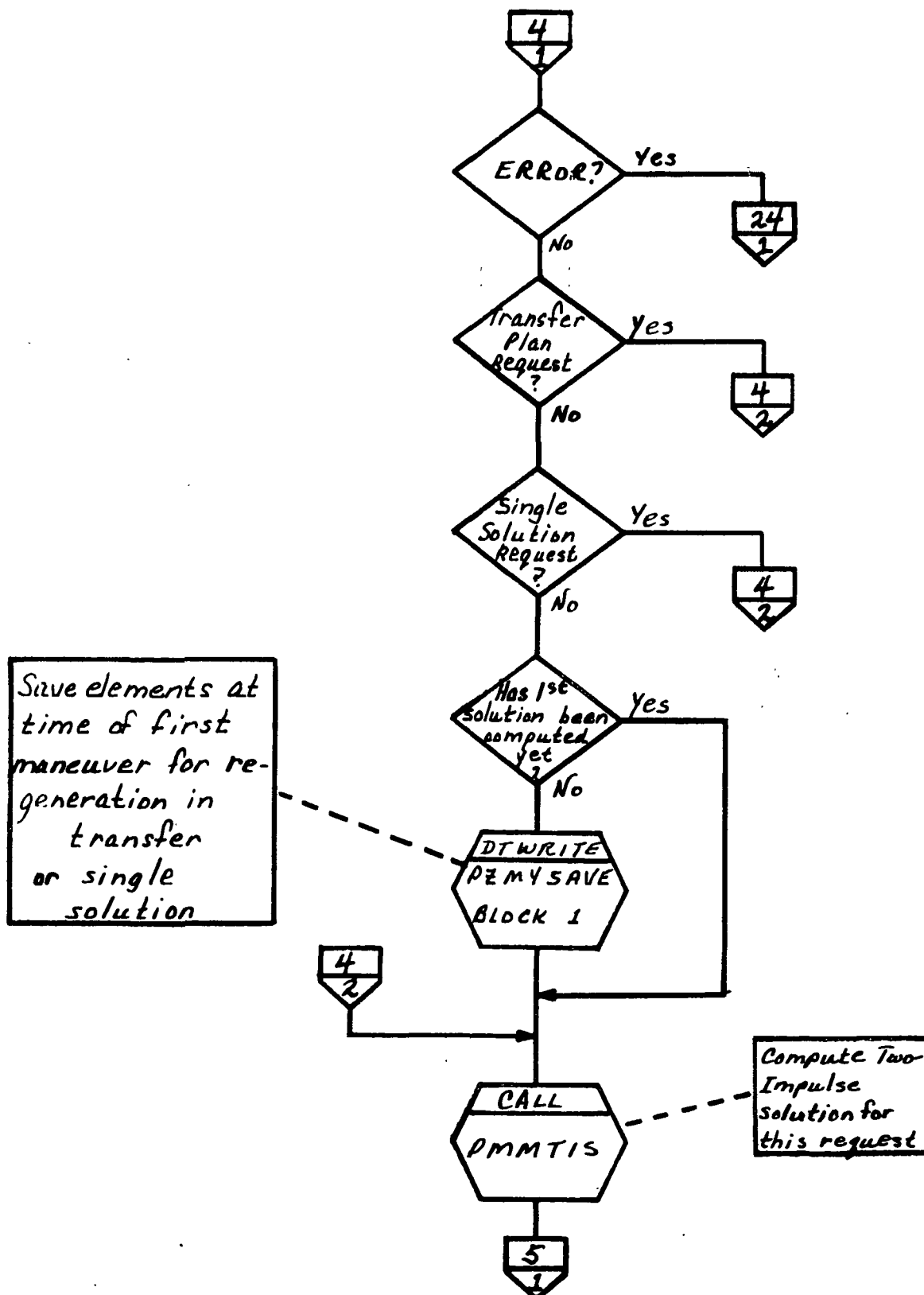


Figure PMSTICN-PCSTICN-1. Flowchart

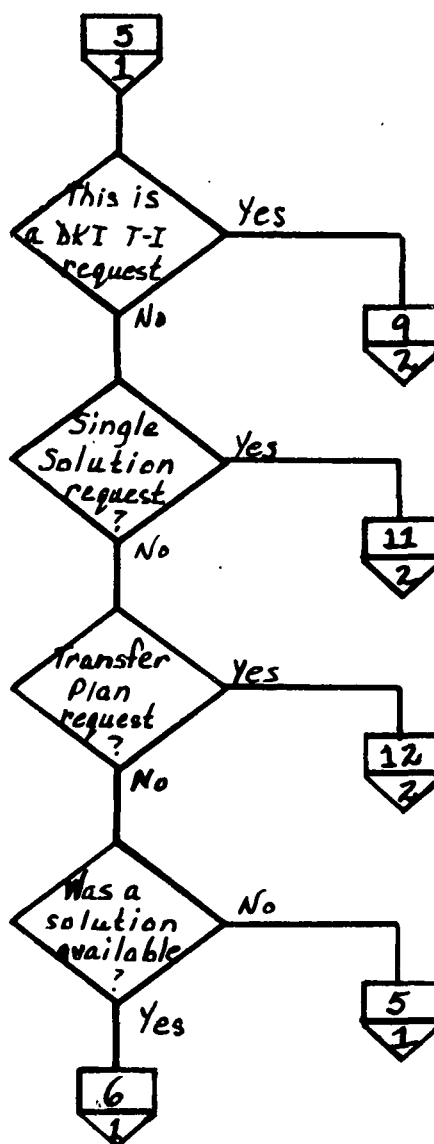


Figure PMSTICN-PCSTICN-1. Flowchart

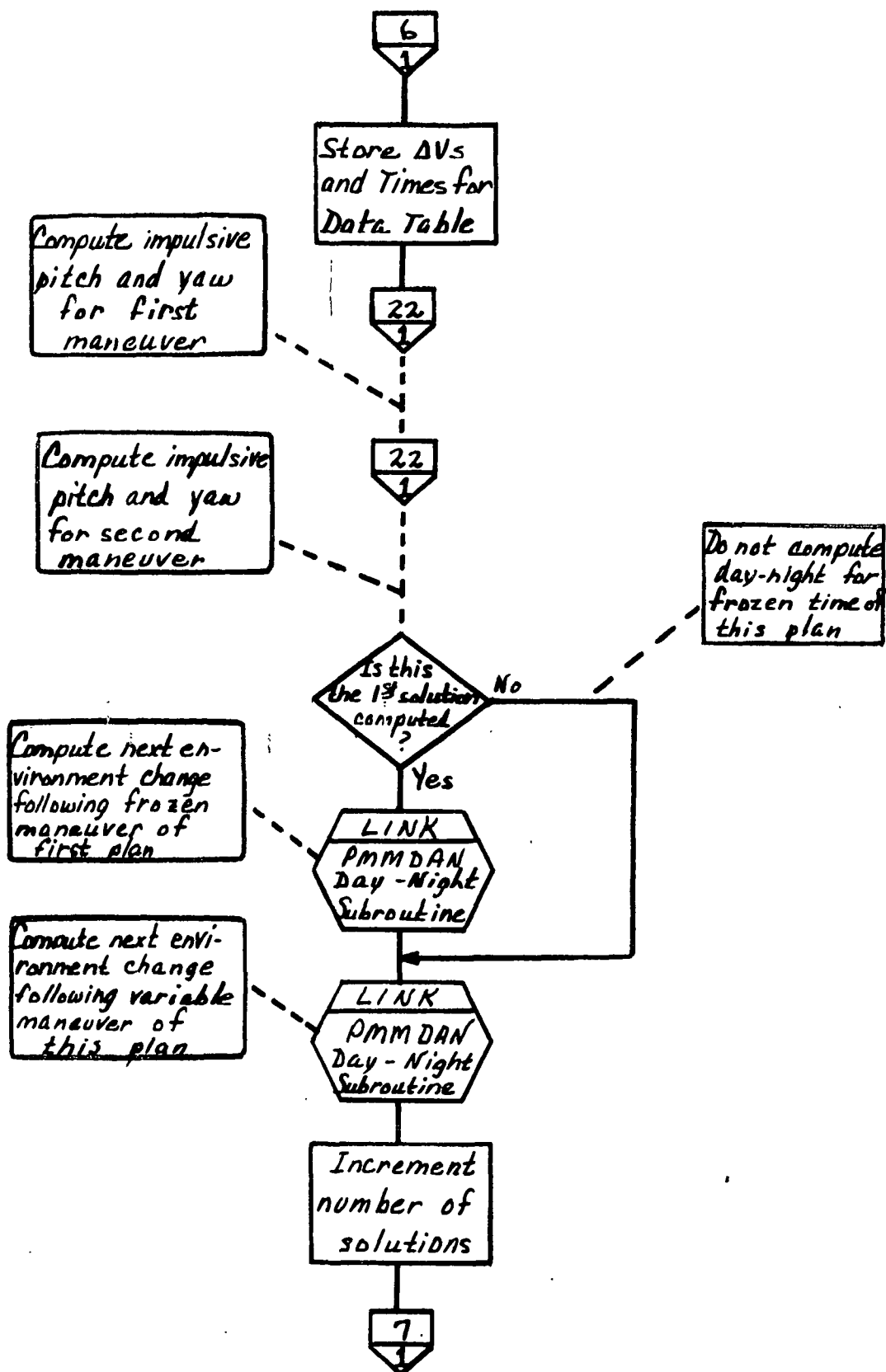


Figure PMSTICN-PCSTICN-1. Flowchart

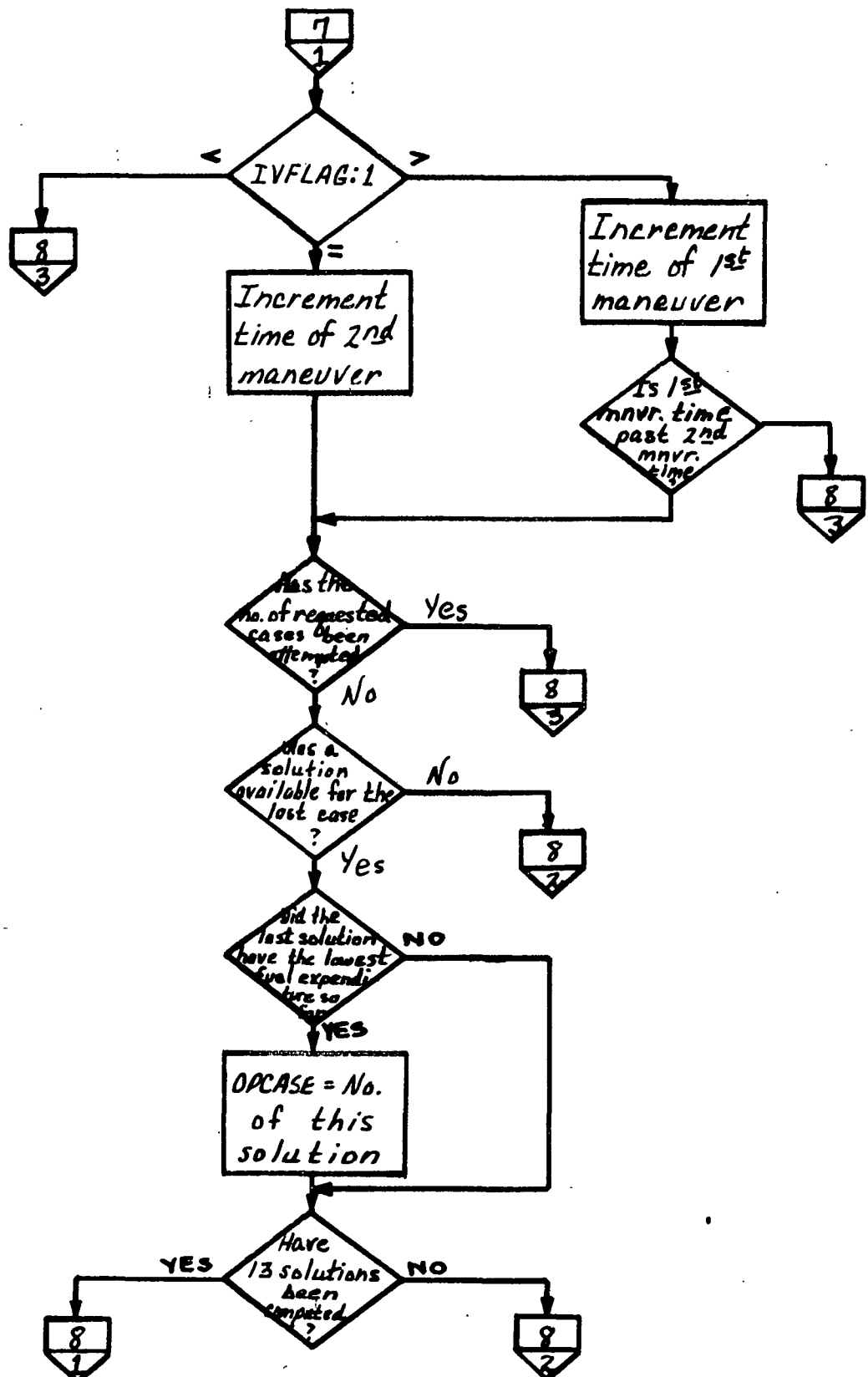


Figure PMSTICN-PCSTICN-1. Flowchart

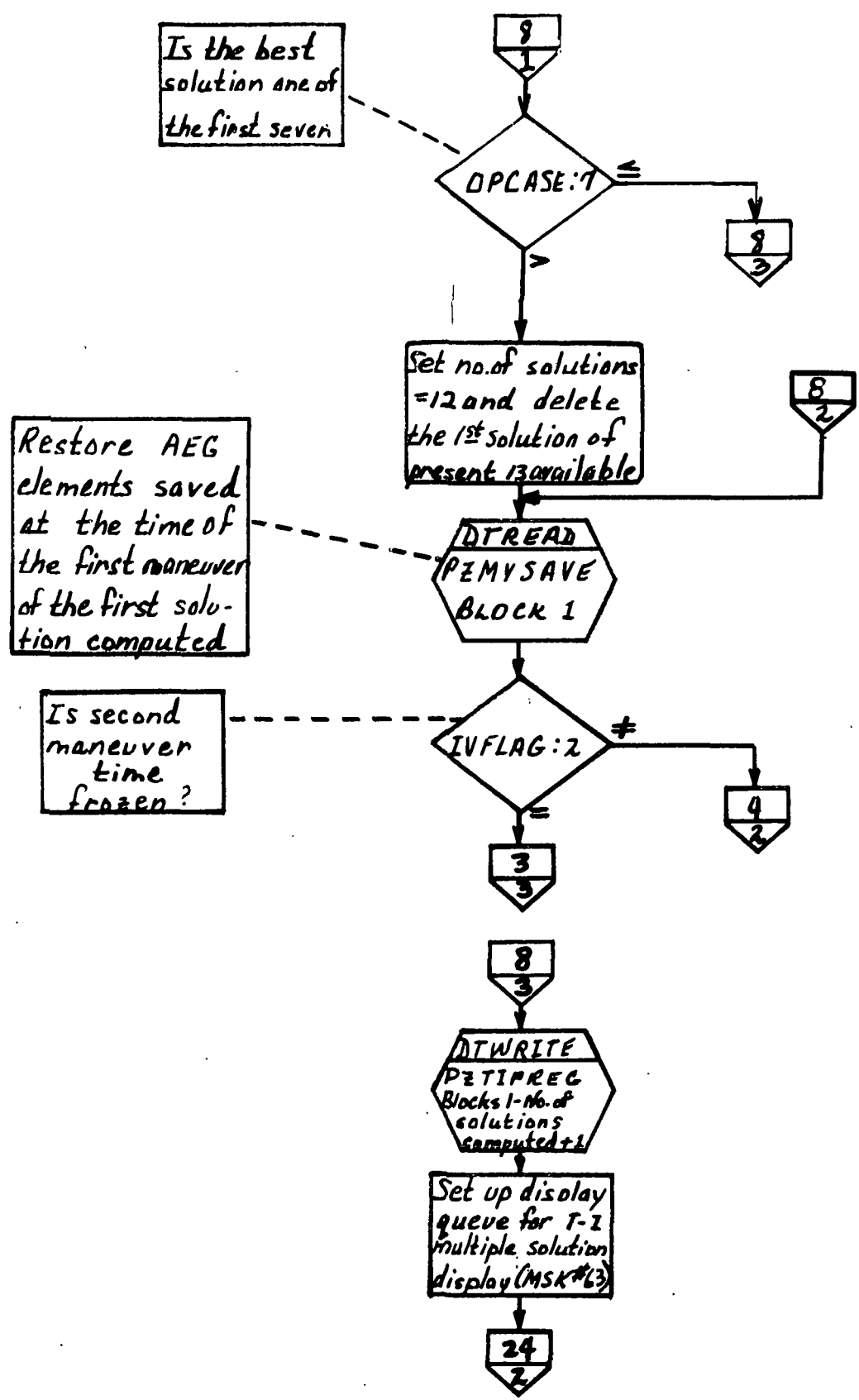


Figure PMSTICN-PCSTICN-1. Flowchart

DKI OR
SPA REQUEST

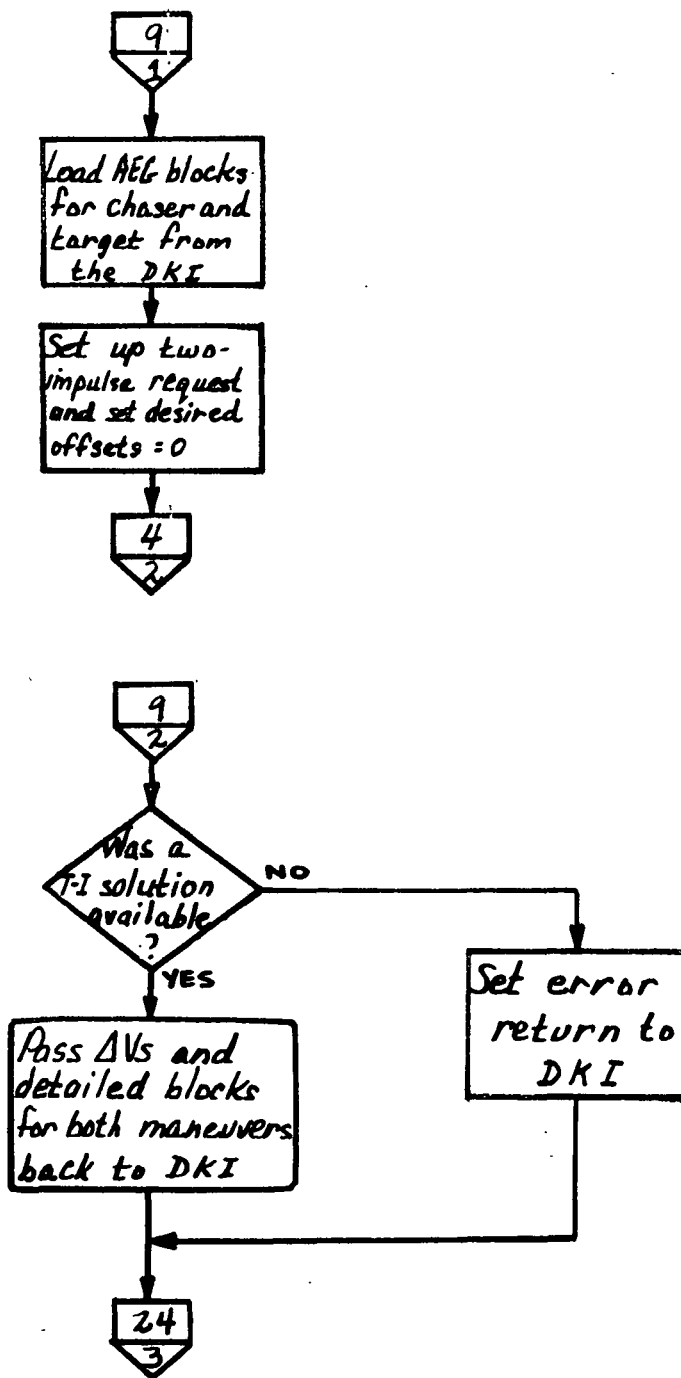


Figure PMSTICN-PCSTICN-1. Flowchart

SINGLE SOLUTION
and
TRANSFER PLAN

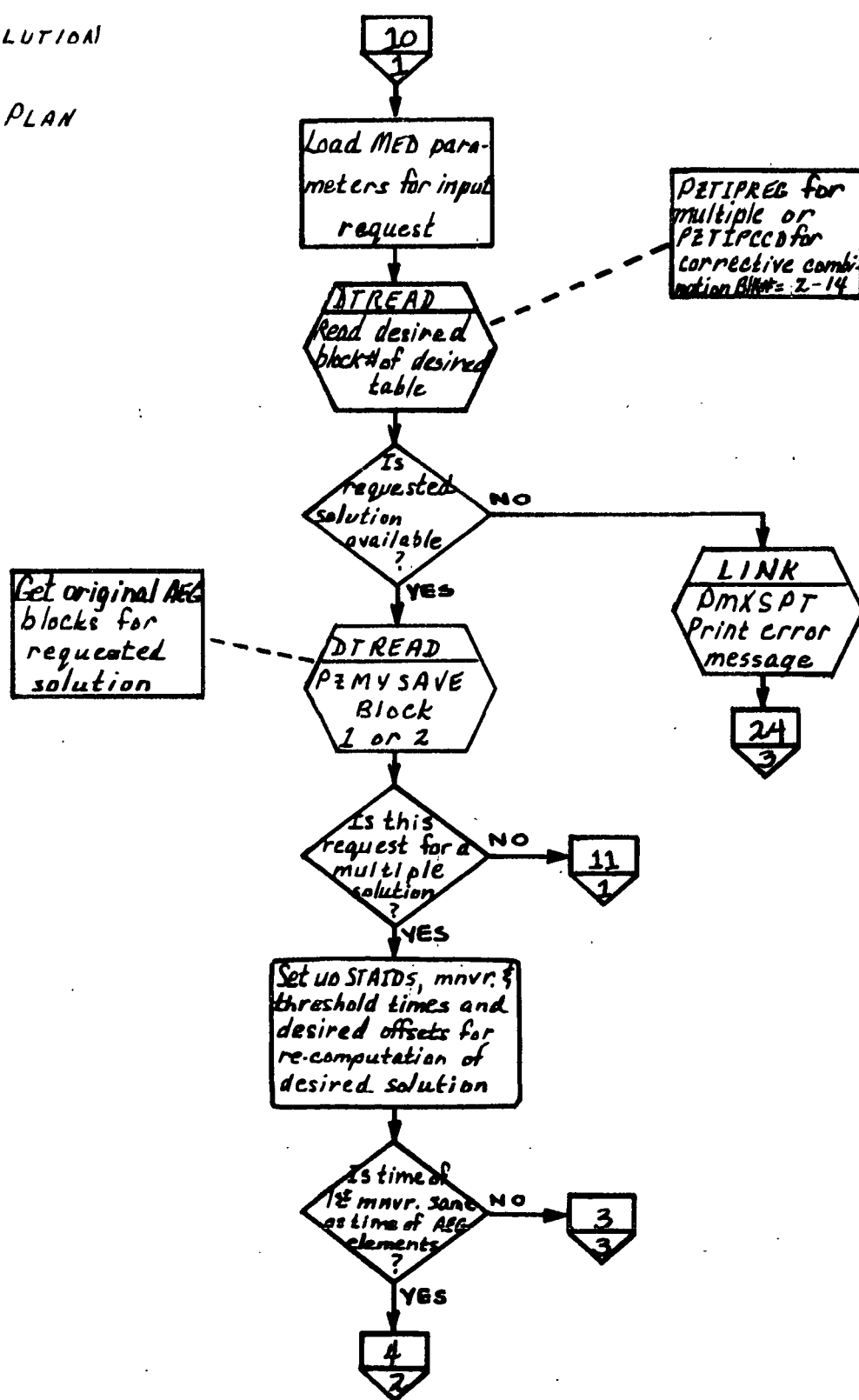


Figure PMSTICN-PCSTICN-1. Flowchart

(Sheet 10 of 24)

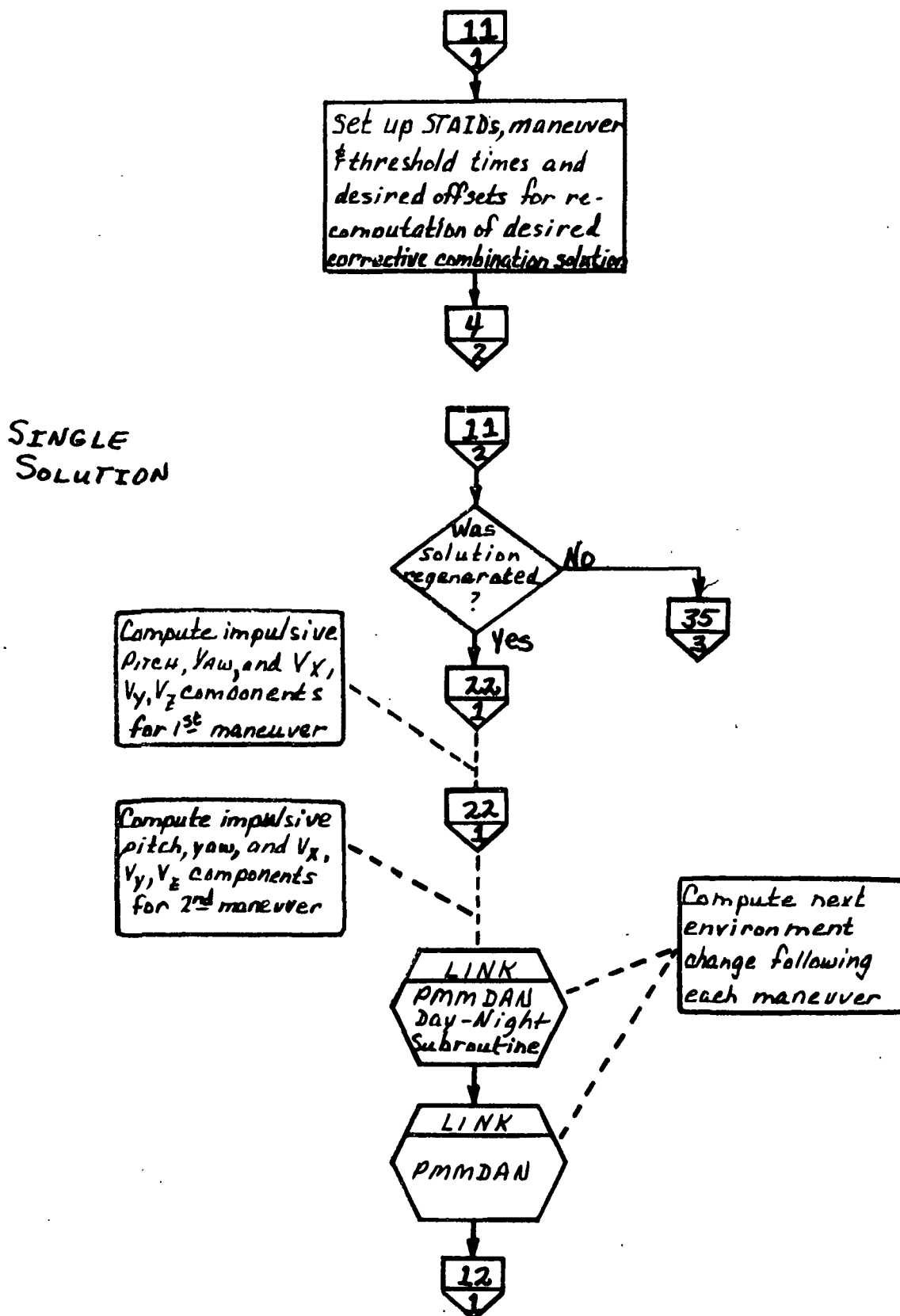


Figure PMSTICN-PCSTICN-1. Flowchart

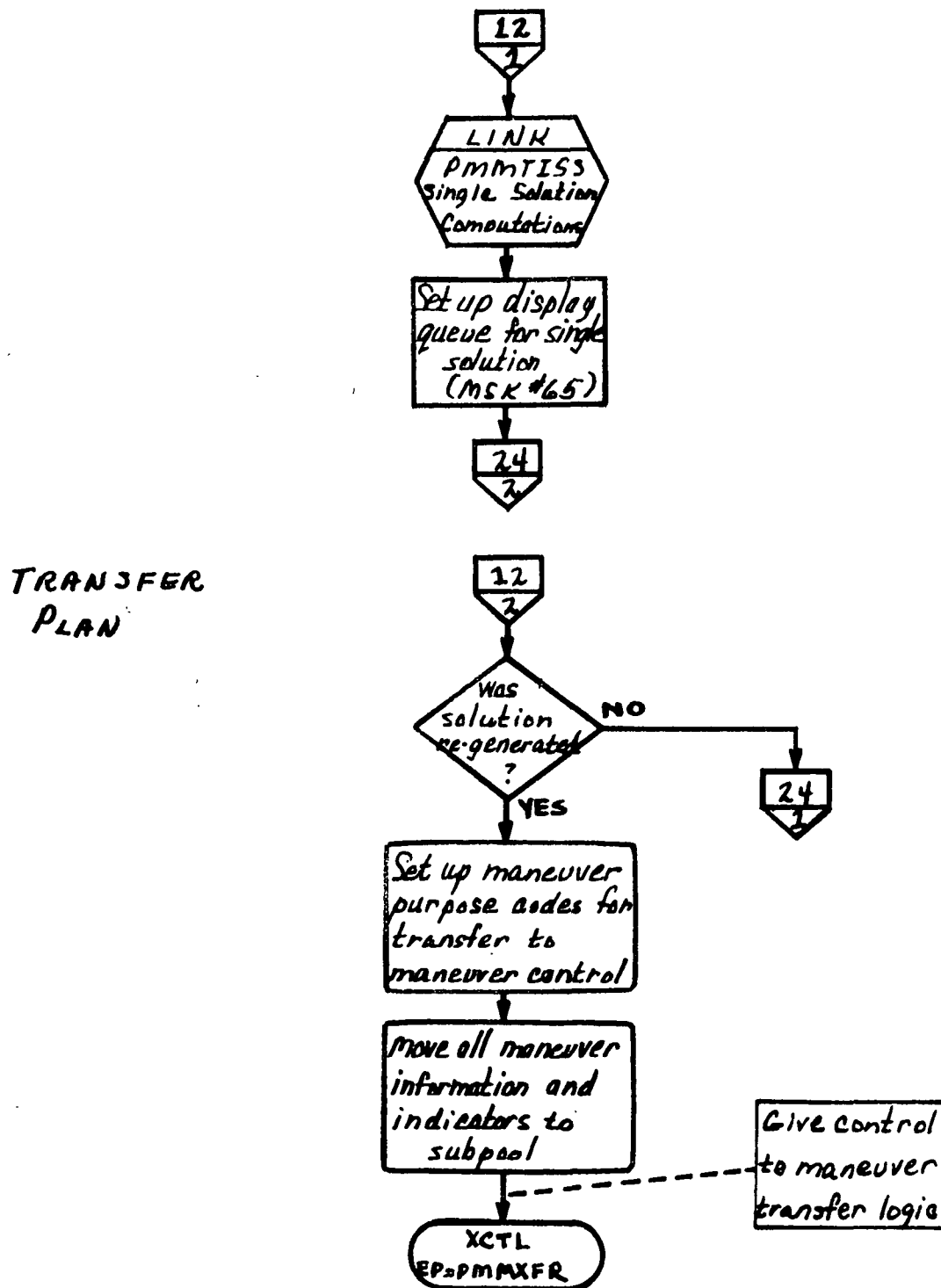


Figure PMSTICN-PCSTICN-1. Flowchart

CORRECTIVE COMBINATION

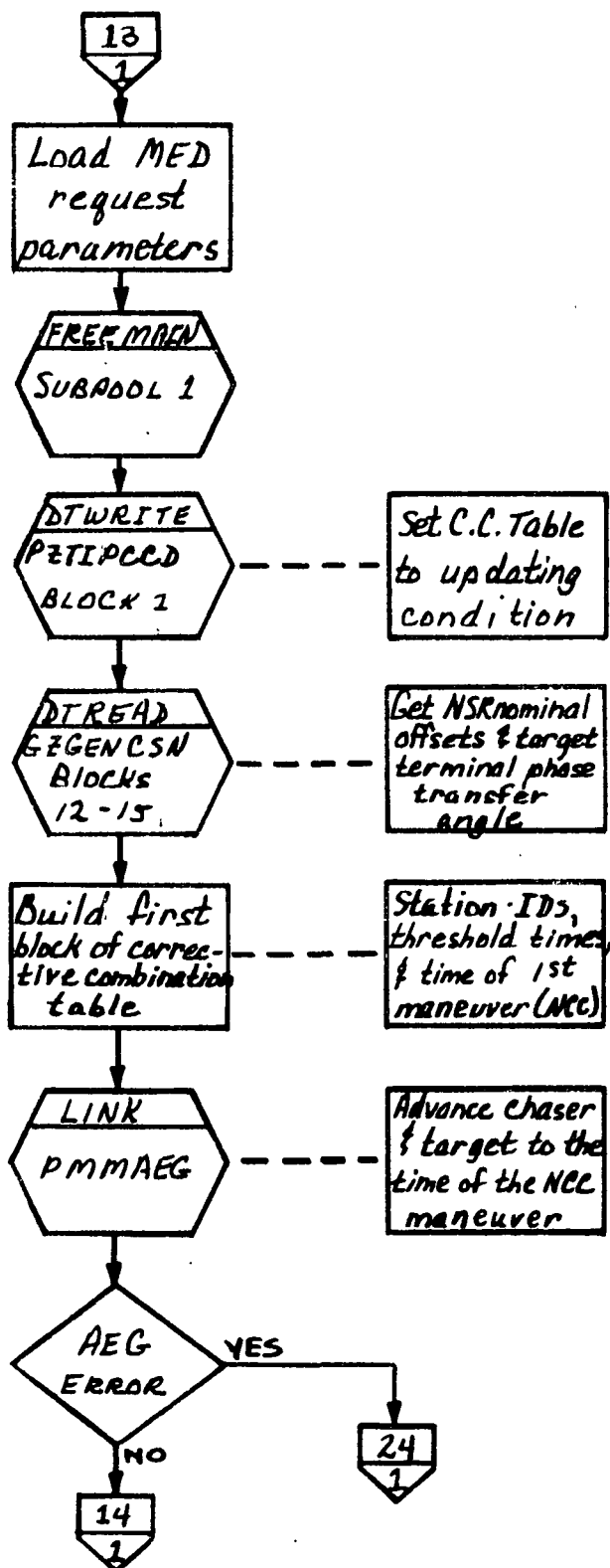


Figure PMSTICN-PCSTICN-1. Flowchart

(Sheet 13 of 24)

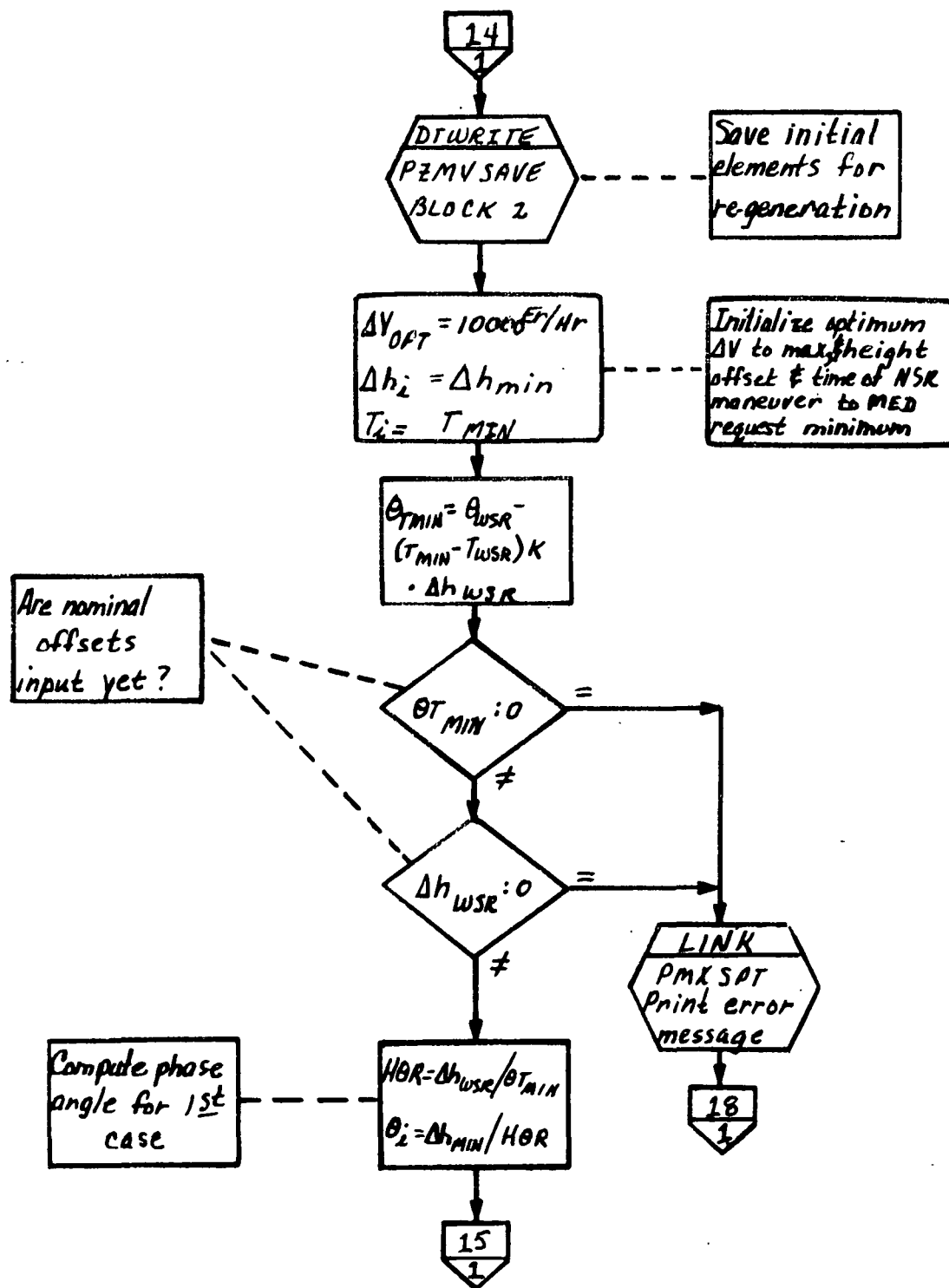


Figure PMSTICN-PCSTICN-1. Flowchart

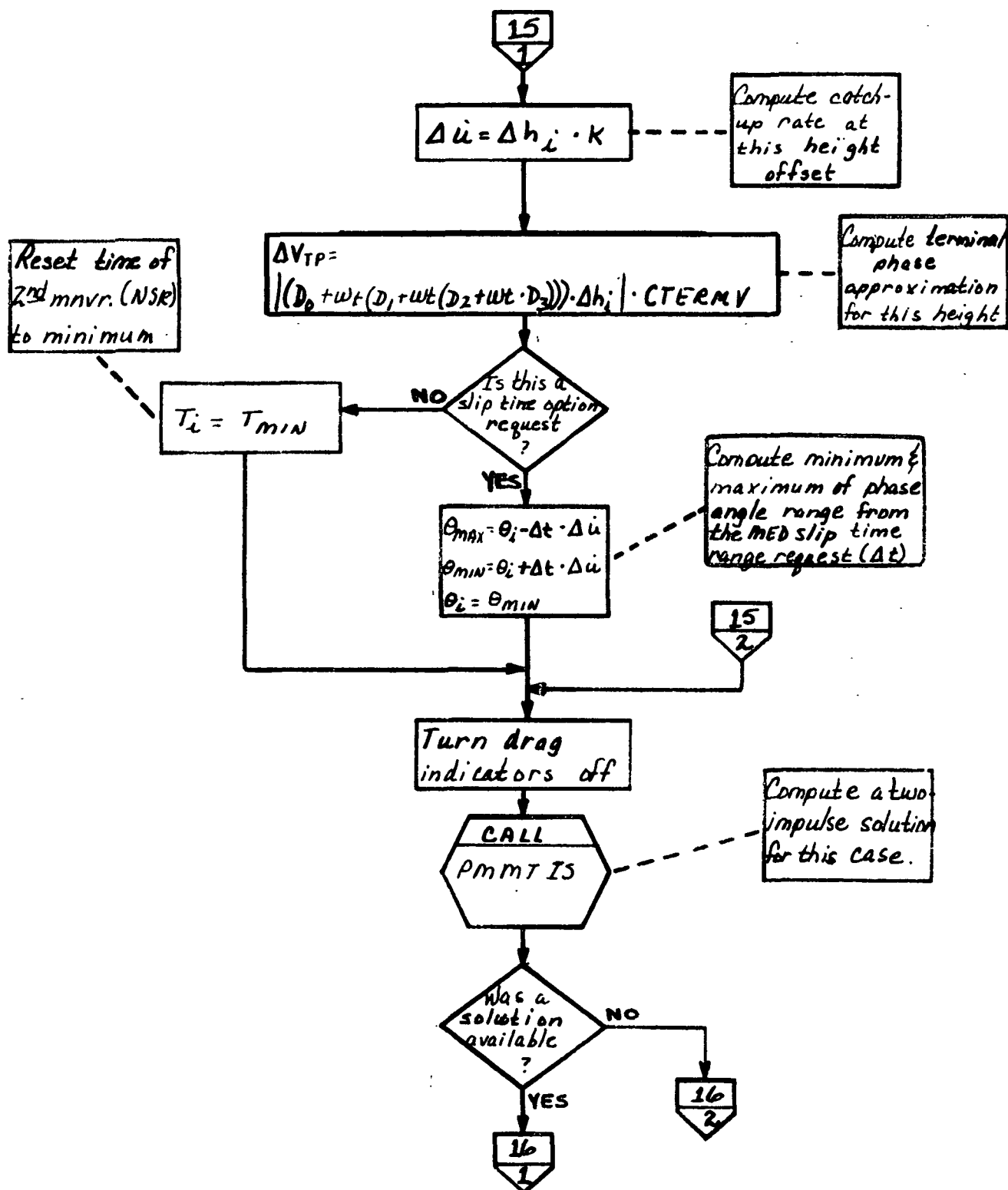


Figure PMSTICN-PCSTICN-1. Flowchart

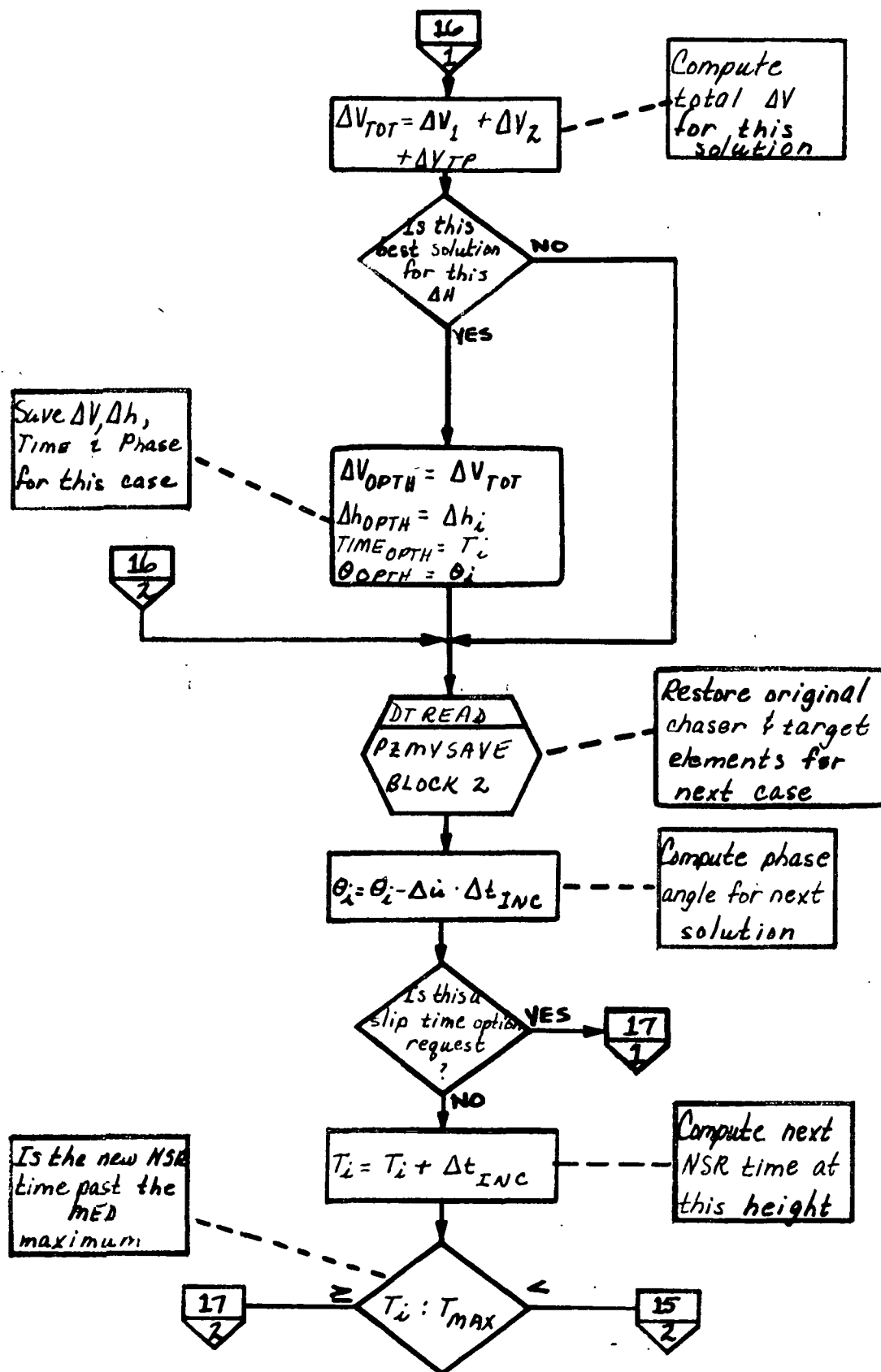


Figure PMSTICN-PCSTICN-1. Flowchart

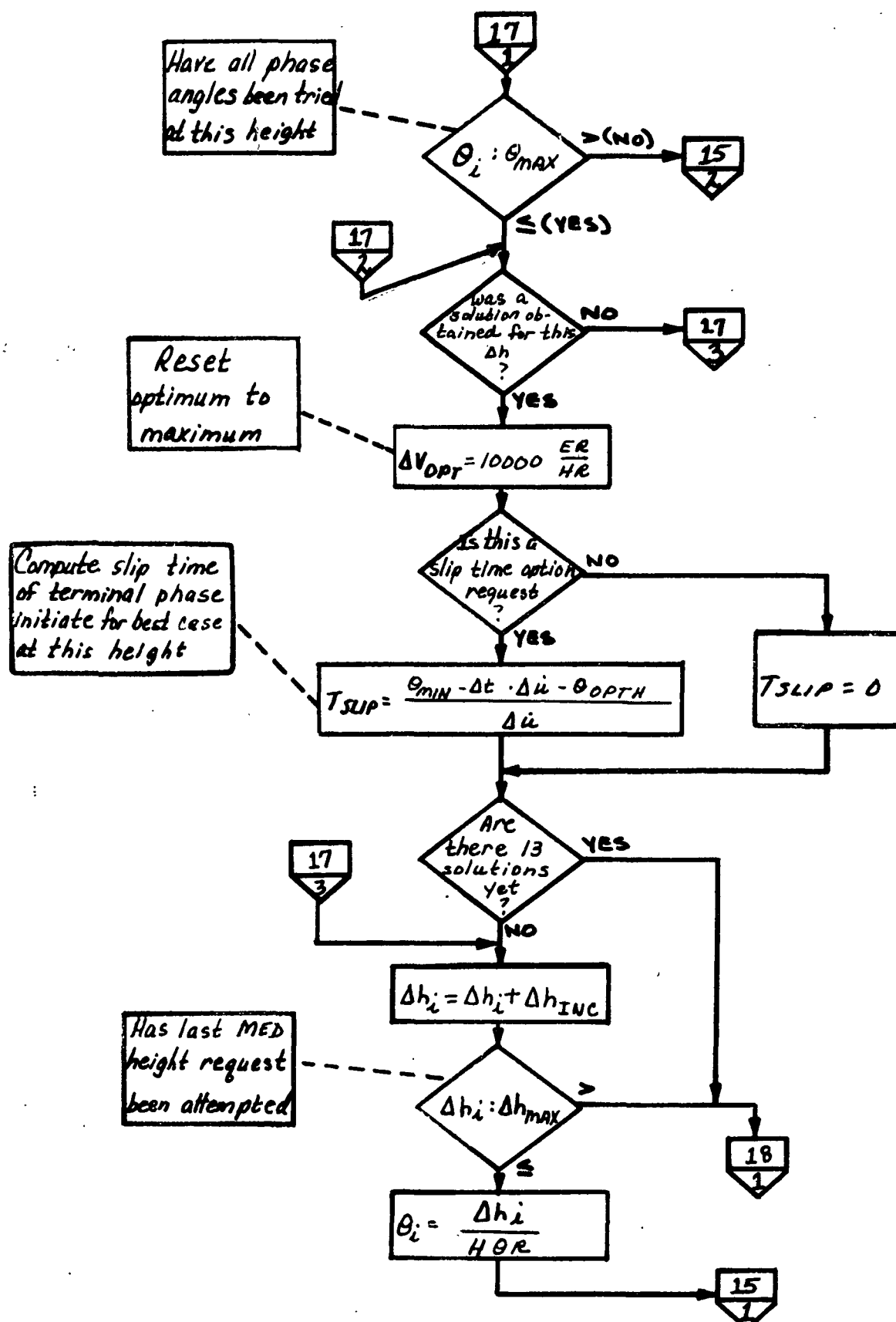


Figure PMSTICN-PCSTICN-1. Flowchart

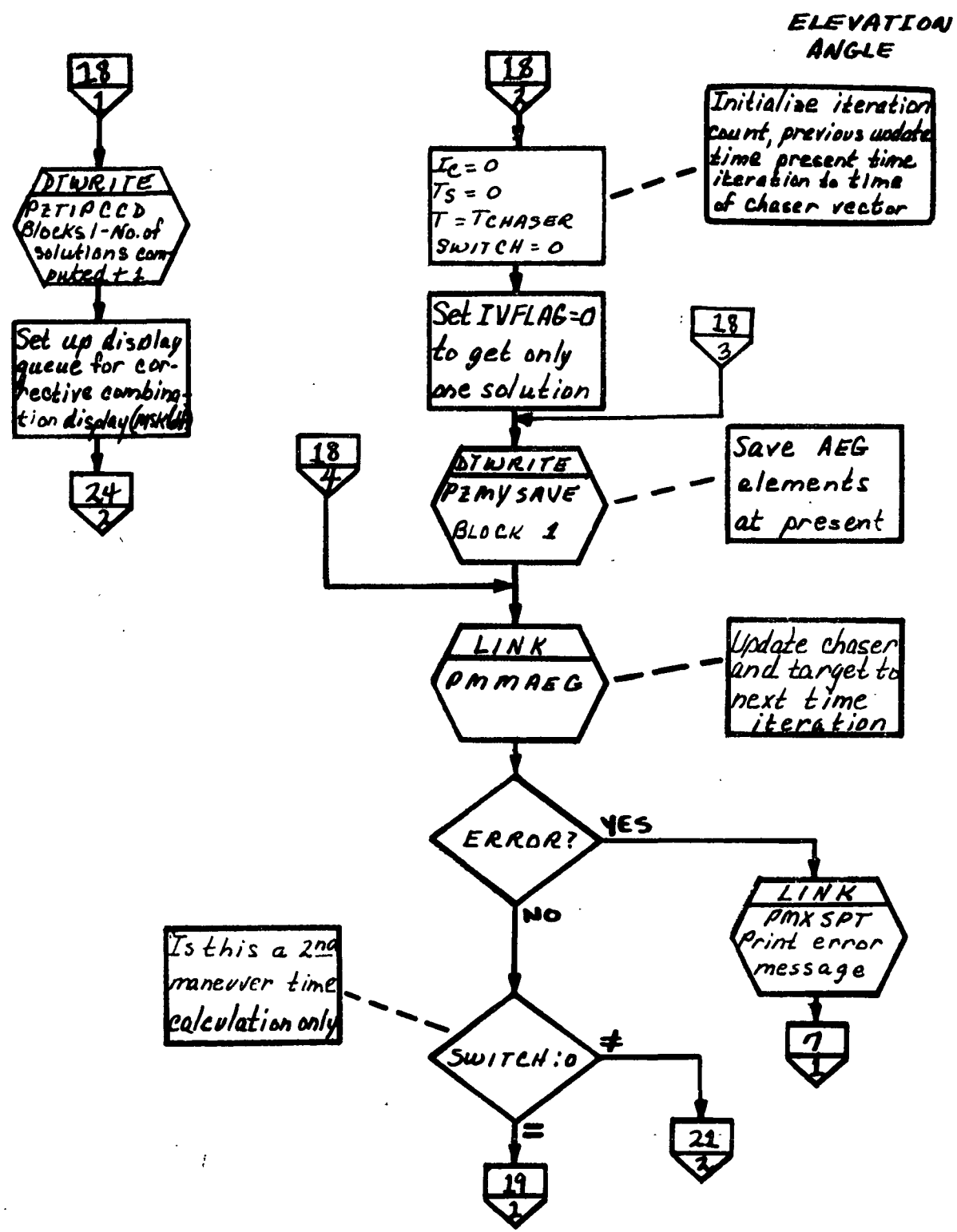


Figure PMSTICN-PCSTICN-1. Flowchart

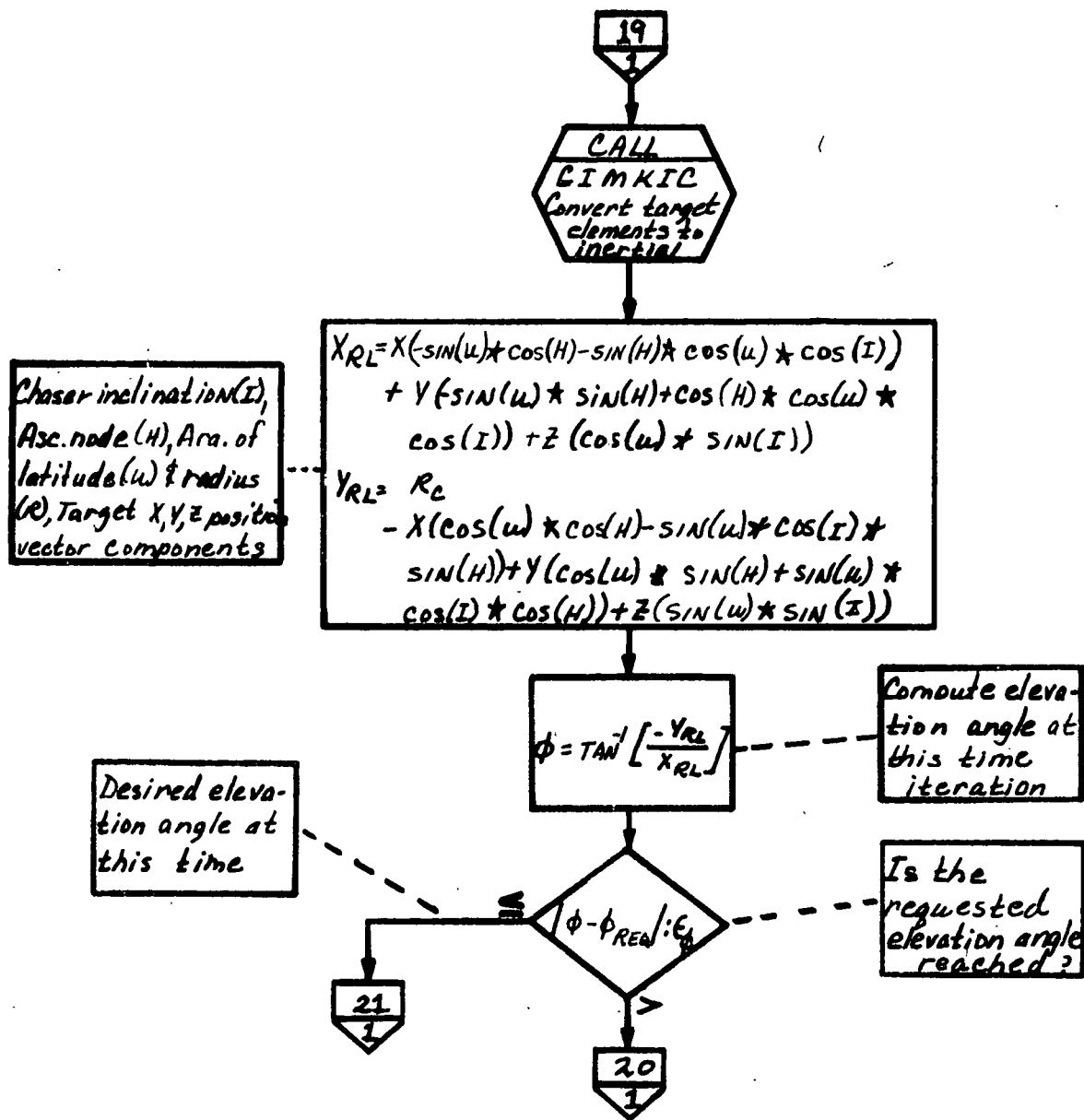


Figure PMSTICN-PCSTICN-1. Flowchart

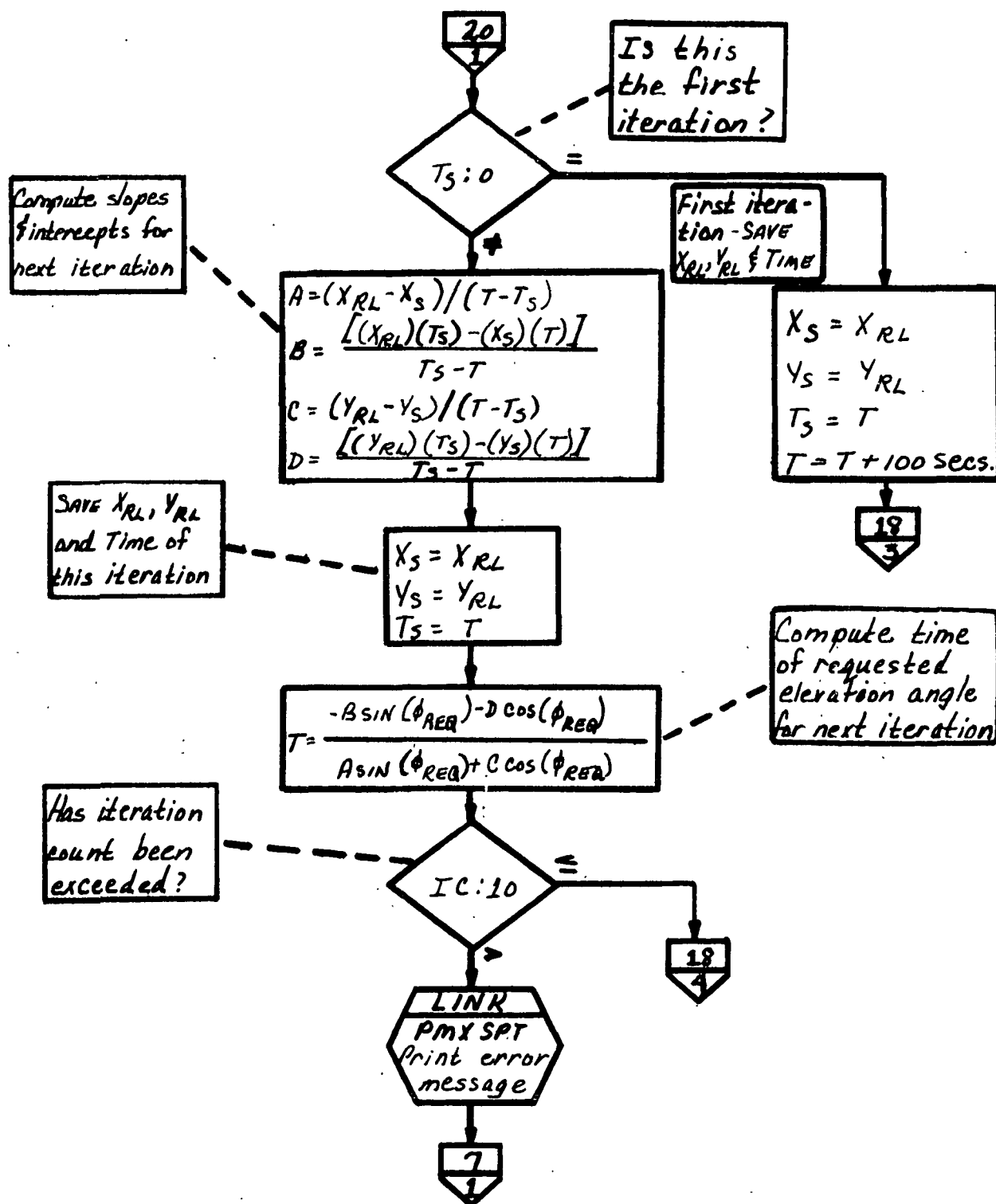


Figure PMSTICN-PCSTICN-1. Flowchart

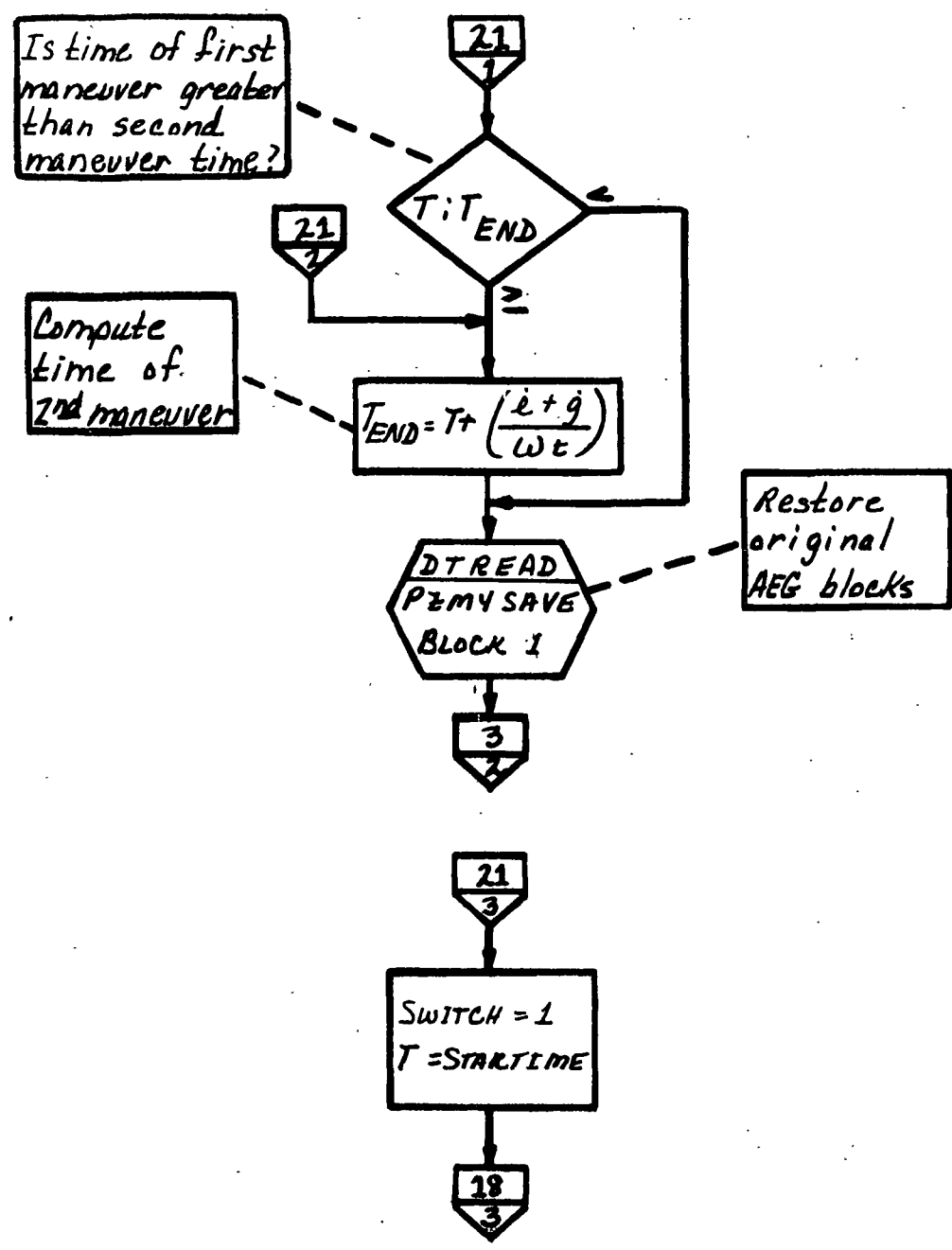


Figure PMSTICN-PCSTICN-1. Flowchart

Inputs are position
& velocity components
before and after a
maneuver -
 $\vec{R}_B, \vec{V}_B, \vec{R}_A, \vec{V}_A$

IMPULSIVE PITCH AND YAW INTERNAL SUBROUTINE

The symbols "X"
and "." represent
cross product and
dot product when
used between vec-
tor notation.

$$\begin{aligned}
 \vec{H}_A &= \vec{R}_A \times \vec{V}_A; h_A = \sqrt{\vec{H}_A \cdot \vec{H}_A} \\
 \vec{H}_B &= \vec{R}_B \times \vec{V}_B; h_B = \sqrt{\vec{H}_B \cdot \vec{H}_B} \\
 n_B &= \sqrt{\vec{R}_B \cdot \vec{R}_B} \\
 \sin \delta &= \frac{(\vec{H}_A \times \vec{H}_B) \cdot \vec{R}_B}{h_A \cdot h_B \cdot n_B} \\
 \dot{n}_B &= \frac{\vec{R}_B \cdot \vec{V}_B}{n_B}; \dot{n}_A = \frac{\vec{R}_A \cdot \vec{V}_A}{n_A} \\
 \Delta \dot{n} &= \dot{n}_B - \dot{n}_A \\
 \Delta V &= \sqrt{(\vec{V}_B - \vec{V}_A) \cdot (\vec{V}_B - \vec{V}_A)} \\
 \text{PITCH} &= \sin^{-1} \left(\frac{\Delta \dot{n}}{\Delta V} \right) \\
 V_{H_B} &= \frac{h_B}{n_B}; V_{H_A} = \frac{h_A}{n_A} \\
 \Delta V_H &= \Delta V \cos(\text{PITCH}) \\
 \text{YAW} &= \sin^{-1} \left(\frac{V_{H_A} \times \sin \delta}{\Delta V_H} \right)
 \end{aligned}$$

$$V_{H_A} = \sqrt{1 - (\sin \delta)^2} \cdot V_{H_B}$$

 $\geq$

$$\text{YAW} = \pi - \text{YAW}$$

$$\text{YAW} = \pi$$

$$\text{YAW} = \text{YAW} - 2\pi$$

23

Figure PMSTICN-PCSTICN-1: Flowchart

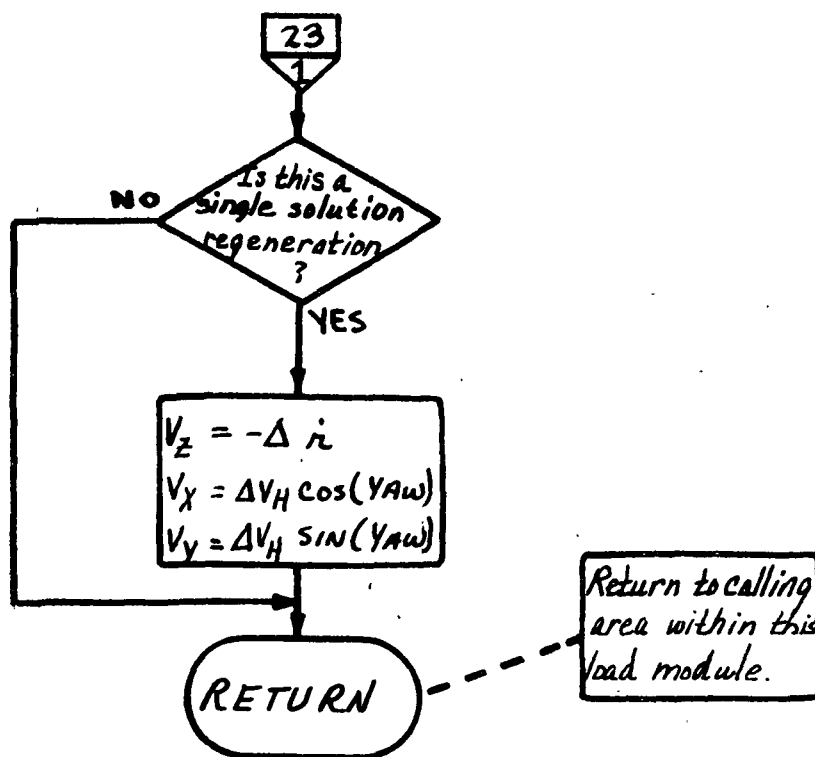


Figure PMSTICN-PCSTICN-1. Flowchart

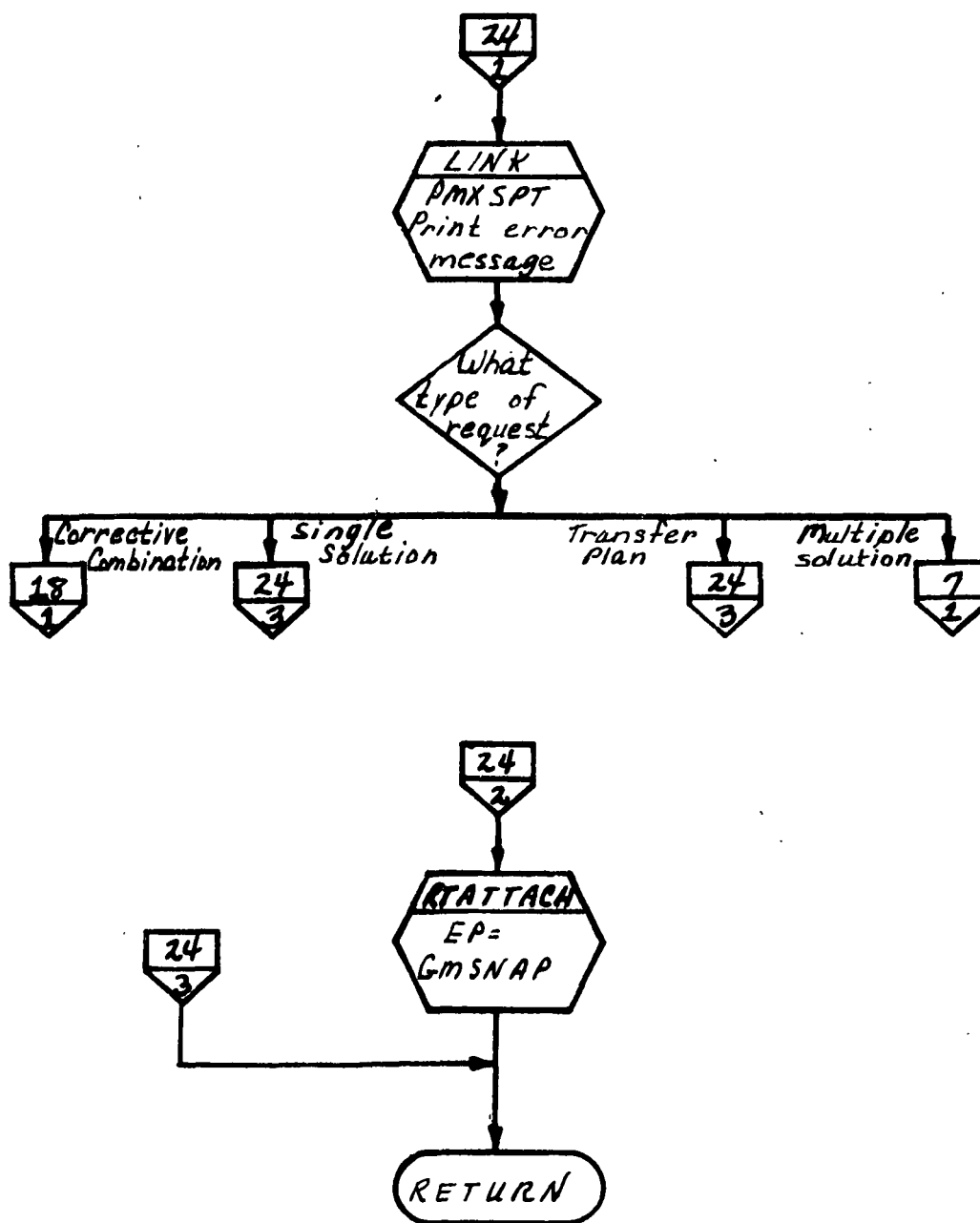


Figure PMSTICN-PCSTICN-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMSTICN -
PCZYCF

DATE 7/28/67 (EORP)

REV

PAGE 1 of 16

- a. Symbol — PMSTICN-PCZYCF
- b. Program Name — Conic Fit Csect
- c. Programming Group — Mission Planning
- d. Programmer(s) — G. Deaton, T. Hickey
- e. Language — FORTRAN IV
- f. Data Tables Used — None
- g. MED's Used — None
- h. Writeup Completion Date — 10/24/66
- i. Flowchart — Figure PMSTICN-PCZYCF-1.

FUNCTION

When given two radius vectors, the transfer angle separating the transfer time, and the integral number of revolutions required for the transfer, PCZYCF computes the two-body semimajor axis, eccentricity, true anomaly at the transfer initiation point, and time past pericenter at the transfer initiation point that are required to satisfy the given parameters.

USAGE

- a. Calling Sequence — CALL PCZYCF COMMON//R1, R2, PHIT, DELT, VXI 2, VYI 2, VXF 1, VYF 1, A, E, FT, SQRMU, INDSOL, NREVS
- b. Inputs —
 - 1. R1 - Transfer completion radius (REAL * 8)
 - 2. R2 - Transfer initiation radius (REAL * 8)
 - 3. PHIT - Transfer angle in radians modulo 2π (REAL * 8)
 - 4. DELT - Transfer Time (REAL * 8)
 - 5. VXI 2* - Horizontal velocity before transfer initiation (REAL * 8)
 - 6. VYI 2* - Radial velocity before transfer initiation (REAL * 8)
 - 7. VXF 1* - Horizontal velocity after transfer completion (REAL * 8)
 - 8. VYF 1* - Radial velocity after transfer completion (REAL * 8)
 - 9. SQRMU - $\sqrt{\mu}$ in units consistent with R1, R2, DELT (REAL * 8)
 - 10. NREVS - Integral number of transfer revolutions (INTEGER * 4)

NREVS = 0 for transfers of less than 360 degrees.
- c. Outputs and Exit —
 - 1. A - Semimajor axis (REAL * 8)
 - 2. E - Eccentricity (REAL * 8)
 - 3. FT - True anomaly after transfer initiation (REAL * 8)

Approval	Approval
<i>G. Deaton</i>	<i>T. Hickey</i>

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMSTICN -
 PCZYCF
 DATE 7/28/67 (EORP)
 REV
 PAGE 2 of 16

4. TPT - Time past pericenter after transfer initiation (REAL * 8)
5. INDSOL - Solution Indicator (INTEGER * 4). Exit is by RETURN statement.
- d. Other Programs Used — Standard library subroutine
- e. Data Restrictions and Limits — The smallest semimajor axis that can be returned is 1.02 Earth radii.
- f. Error — If INDSOL > 0, a solution was generated. If INDSOL < 0, no solution was obtained.

*This parameter need be given only if NREVS > 0.

METHOD*

The conic fit technique is designed to compute two-body semimajor axis, a , and eccentricity, e , of an ellipse which satisfies two given radii, Γ_1 , Γ_2 and Γ_1 are separated by the central angle, ϕ_T . The fourth quantity needed to make the ellipse unique is the transfer elapsed time, Δt , required for passage through the travel angle, ϕ_T . This is all the information needed if $\phi_T < 2\pi$.

Two other parameters (which are computed for output) are the true anomaly, f_T , and time past pericenter, t_{PT} , of the maneuvering vehicle in the transfer

* Reference MSC Internal Note No. 65-FM-28, by J. D. Alexander and "The Determination of an Orbit From Two Positions", by P. E. El'Yasberg, Planet. Space Science, 1963, Vol. 11, pages 395 to 415.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMSTICN -
PCZYCF

DATE 7/28/67 (EORP)

REV

PAGE 3 of 16

orbit at transfer initiation. Transfer initiation is assumed to be at Γ_2 and transfer completion is assumed to be at Γ_1 .

If $\Phi_T > 2\pi$, two solutions may exist. In this case, the integral number of revolutions, N , must also be given to the program as input. However, the value of Φ_T should be input modulo 2π . In addition, so that the solution which requires the smaller fuel expenditure can be chosen, the radial and horizontal velocity components before the transfer initiation and after the transfer completion maneuvers must be input.

First, consider the case $\Phi_T < 2\pi$ ($N = 0$). It can be determined if an elliptic solution exists by computing the parabolic transfer time

$$\Delta t_{\text{PAR}} = \frac{1}{6\sqrt{\mu}} \left[(R_1 + R_2 + S)^{3/2} \pm (R_1 + R_2 - S)^{3/2} \right],$$

where the cord length $S = \left[R_1^2 + R_2^2 - 2 R_1 R_2 \cos \Phi_T \right]^{1/2}$. The plus sign is chosen in the equation for Δt_{PAR} if $\Phi_T > \pi$, the minus sign if $\Phi_T < \pi$. This convention also holds for other formulae mentioned below. If $\Delta t < \Delta t_{\text{PAR}}$, no elliptic solution exists. An error indicator is set and the program is exited.

Next, it is necessary to determine to which family, first or second, the solution ellipse belongs. The conic fit equations which are used to determine this and the this and that are used in the semimajor axis iterative scheme are

$$\text{First Family: } \Delta t_1 = \frac{a^{3/2}}{\sqrt{\mu}} \left[2\pi N + \epsilon - \sin \epsilon \pm (\delta - \sin \delta) \right]$$

$$\text{Second Family: } \Delta t_2 = \frac{a^{3/2}}{\sqrt{\mu}} \left[2\pi (N + 1) - \epsilon + \sin \epsilon \pm (\delta - \sin \delta) \right]$$

$$\text{Limiting Case: } \Delta t_{\text{lim}} = \frac{a_{\text{lim}}^{3/2}}{\sqrt{\mu}} \left[\pi (2N + 1) \pm (\delta_{\text{lim}} - \sin \delta_{\text{lim}}) \right]$$



In the equations above,

$$\epsilon = 2 \tan^{-1} \left[\frac{R_1 + R_2 + S}{4a - (R_1 + R_2 + S)} \right]^{1/2}$$

$$\delta = 2 \tan^{-1} \left[\frac{R_1 + R_2 - S}{4a - (R_1 + R_2 - S)} \right]^{1/2}$$

$$\alpha_{\text{lim}} = \frac{R_1 + R_2 + S}{4}$$

$$\delta_{\text{lim}} = 2 \tan^{-1} \left[\frac{R_1 + R_2 + S}{2S} \right]$$

α_{lim} is the shortest possible semimajor axis that can satisfy R_1 , R_2 , and Φ_T .

The family to which the solution belongs is determined by computing the boundary transfer time, Δt_{lim} , between the first and second family solutions. The correct conic fit equation is chosen on the following bases:

- (1) If $\Delta t_{\text{PAR}} < \Delta t < \Delta t_{\text{lim}}$, use first family equation.
- (2) If $\Delta t = \Delta t_{\text{lim}}$, use the limiting case equation.
- (3) If $\Delta t > \Delta t_{\text{lim}}$, use second family equation.

Once the conic fit equation to be used is known, the correct equation for the Newton-Raphson iteration can be chosen. The general equation for the Newton-Raphson iteration is

$$\alpha_{i+1} = \alpha_i - \frac{f(\alpha_i)}{f'(\alpha_i)}$$

For instance, if the desired solution is of the first family, then

$$f(\alpha) = \Delta t - \frac{\alpha^{3/2}}{\sqrt{\mu}} \left[2\pi N + \epsilon - \sin \epsilon + (\delta - \sin \delta) \right]$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMSTICN -
PCZYCF

DATE 7/28/67 (EORP)

REV

PAGE 5 of 16

and

$$f^1(\alpha) = \frac{\alpha^{1/2}}{\sqrt{\mu}} \left\{ \left[\sqrt{\frac{R_1 + R_2 + S}{4a - (R_1 + R_2 + S)}} (1 - \cos \epsilon) \pm \sqrt{\frac{R_1 + R_2 - S}{4a - (R_1 + R_2 - S)}} (1 - \cos \delta) \right] - \frac{3}{2} \left[2\pi N + \epsilon - \sin \epsilon \pm (\delta - \sin \delta) \right] \right\}$$

The iteration is started by letting the first guess be $\alpha_{\text{lim}} + 10,000$ feet. J. D. Alexander of NASA points out in his internal note that using this first guess is as good or better than using other methods that compute a first guess.

The iteration is continued until $|\alpha_{i+1} - \alpha_i| \leq 50$ feet. If convergence is not obtained, an error indicator is set and the best semimajor axis — i.e., the α_{i+1} for which $|\alpha_{i+1} - \alpha_i|$ was smallest — is output and used for computing the output parameters e , f_T , and t_{PT} .

After the semimajor axis is obtained, the eccentricity is computed using the equation

$$e = \left\{ \left[\frac{R_1 - \alpha - (R_2 - \alpha) \cos(\epsilon - \delta)}{\sin(\epsilon - \delta)} \right]^2 + (\alpha - R_2)^2 \right\}^{1/2} / \alpha$$

The true anomaly, f_T , on the transfer trajectory at transfer initiation is computed as follows:

$$\text{Semi-latus Rectum} = P_T = \alpha (1 - e^2)$$

$$(\cos f)_T = \frac{P_T - R_2}{e \cdot R_2}$$

$$(\sin f)_T = \frac{(\cos f)_T \cos \phi_T - \frac{P_T - R_1}{e \cdot R_1}}{\sin \phi_T}$$

$$f_T = \tan^{-1} \left[\frac{(\sin f)_T}{(\cos f)_T} \right]$$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMSTICN -
PCZYCF

DATE 7/28/67 (EOR)

REV

PAGE 6 of 16

The time past pericenter corresponding to f_T is

$$t_{PT} = \alpha P_T^{1/2} \left[\frac{2}{\sqrt{1-e^2}} \tan^{-1} \left(\sqrt{\frac{1-e}{1+e}} \tan \frac{f_T}{2} \right) - \frac{e(\sin f)_T}{1+e(\cos f)_T} \right]$$

Orientation of the plane of the trajectory must be done by the calling program.

Next, consider the case $\Phi_T > 2\pi$ ($N > 0$). First a Newton-Raphson iteration is performed to compute the semimajor axis, α_N , which correspond to the minimum multiple revolution transfer time, Δt_N , which can exist for elliptic motion. The equation used for this iteration is on sheet 4 of 9 of the flowchart and is obtained from $\frac{d(\Delta t_1)}{d\alpha} = 0$.

As before, the best value is used if convergence is not reached before the iteration counter is exhausted.

The minimum multiple revolution transfer time, Δt_N , is computed so that it can be determined if a solution exists. The minimum time is

$$\Delta t_N = \frac{\alpha_N^{3/2}}{\sqrt{\mu}} \left[2\pi N + \epsilon_N - \sin \epsilon_N \pm (\delta_N - \sin \delta_N) \right]$$

where $\epsilon_N = 2 \tan^{-1} \left[\frac{R_1 + R_2 + S}{4a_N - (R_1 + R_2 + S)} \right]^{1/2}$

and $\delta_N = 2 \tan^{-1} \left[\frac{R_1 + R_2 - S}{4a_N - (R_1 + R_2 - S)} \right]^{1/2}$

The sign of the last term in the Δt_N computation is chosen according to the modulo 2π value of the transfer angle, Φ_T . If $\Phi_T > \pi$, use the plus sign.

If $\Delta t_N > \Delta t$, an error indicator is set, and the program is exited. If $\Delta t_N < \Delta t$, a solution exists and the boundary transfer time between the first and second family solutions is computed using the limiting case formula for Δt .

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMSTICN -
PCZYCF

DATE 7/28/67 (EORP)

REV

PAGE 7 of 16

Knowing t_{lim} and t_N , one can determine the types of solutions.

If $t_N < t_{lim}$, both solutions are first family.

If $t = t_{lim}$, one is first family, the other limiting case.

If $t > t_{lim}$, one is second family, the other first family.

If $t = t_N$, there is only one solution. It is first family.

The two solutions are computed using the same iterative procedure as is used for $N = 0$. First family solutions are computed using for a first guess $\alpha_i = \alpha_N + 500$ feet ($i = 0$). Second family solutions are computed using for a first guess $\alpha_i = \alpha_{lim} + 10,000$ feet. The solution that requires the smaller amount of fuel expenditure is output.

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMSTICN -

PCZYCF

DATE 7/28/67 (EORP)

REV

PAGE 8 of 16

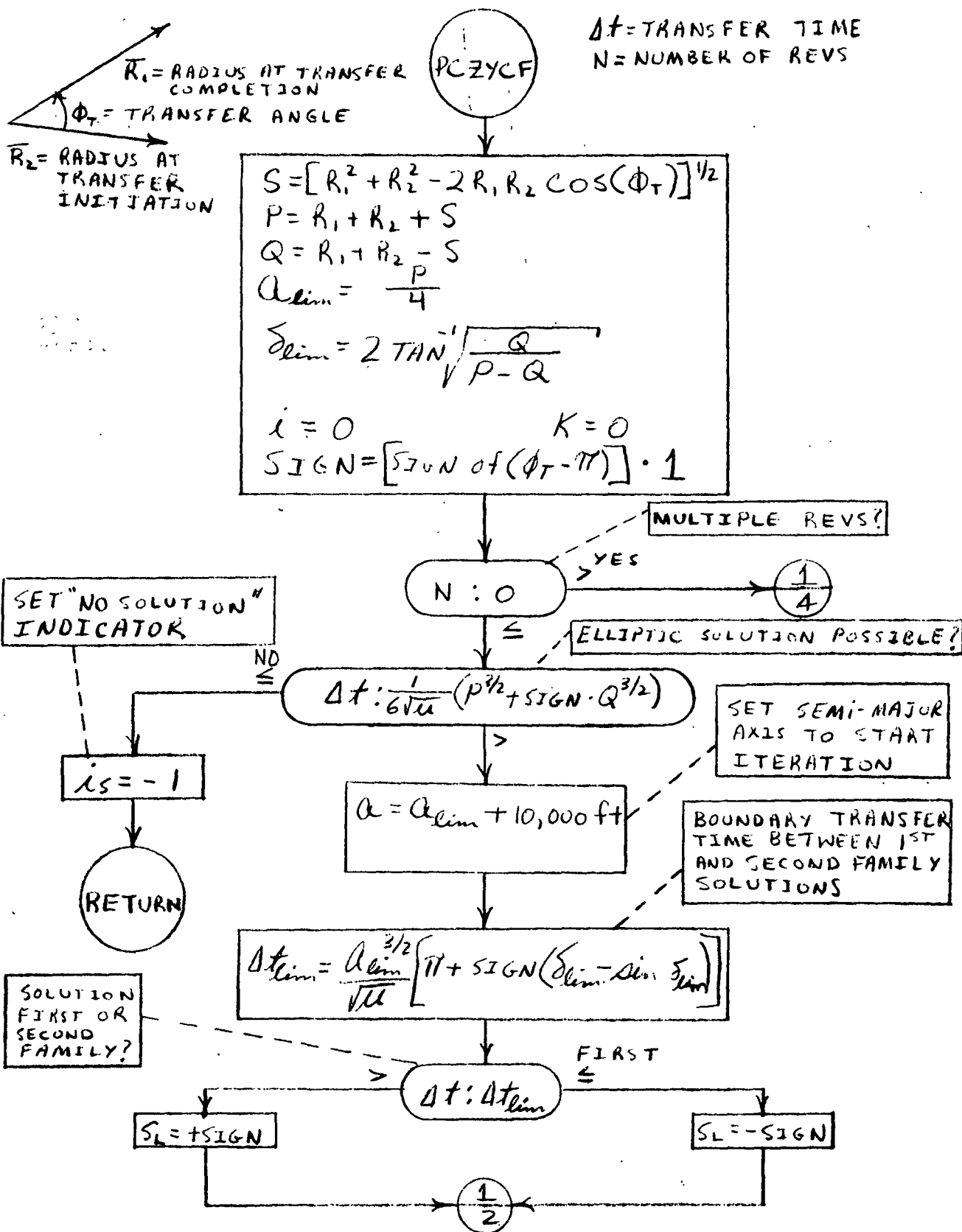


Figure PMSTICN - PCZYCF-1. Flowchart

(Sheet 1 of 9)

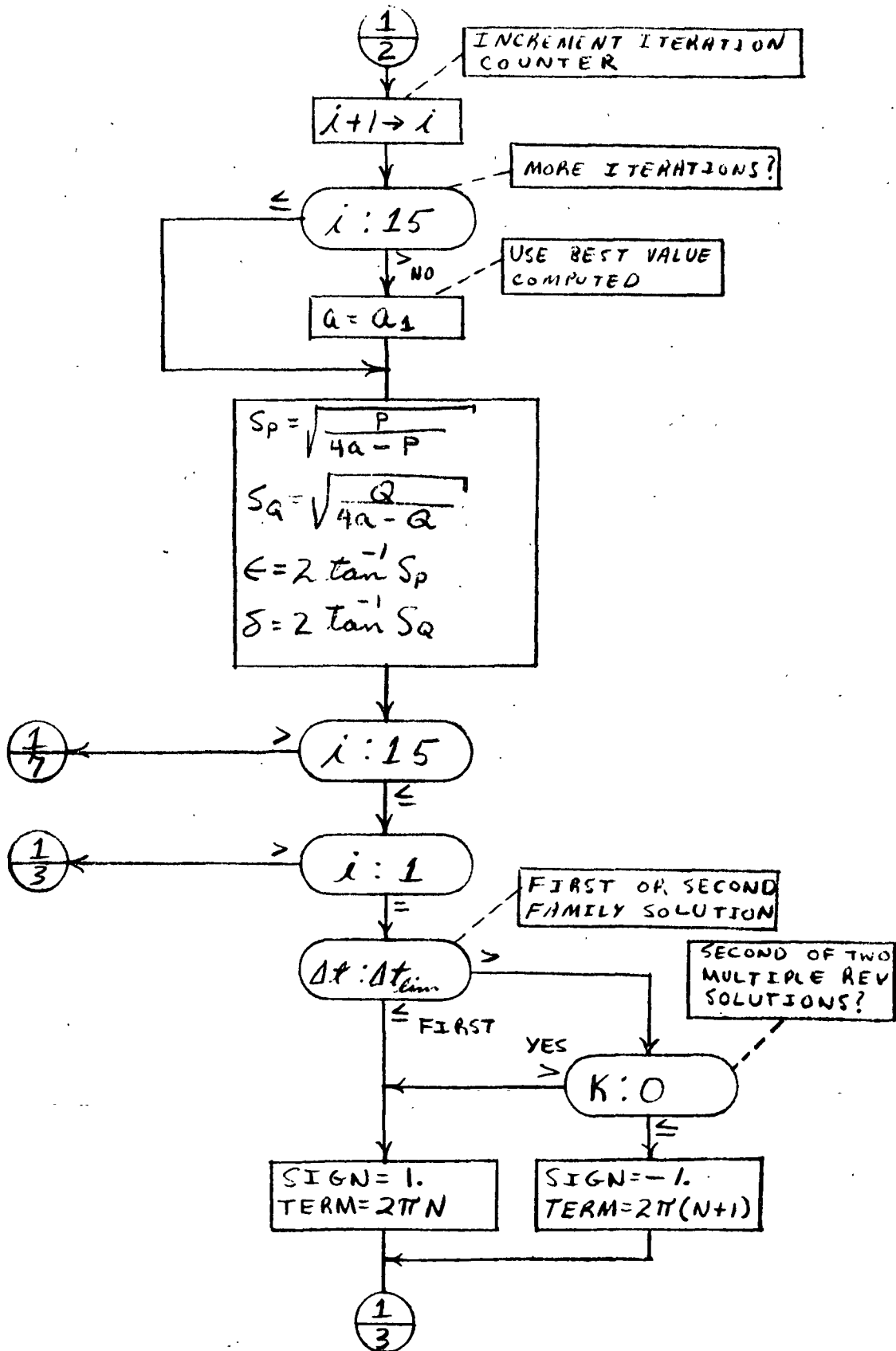


Figure PMSTICN - PCZYCF-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMSTICN -
PCZYCF

DATE 7/28/67 (EORP)

REV

PAGE 10 of 16

COMPUTE SEMI-MAJOR AXIS
USING NEWTON'S METHOD

$$P_p = \text{TERM} + \text{SIGN}[E - \sin E - S_L(\delta - \sin \delta)]$$

$$a_i = a - \frac{\sqrt{\mu} \Delta t - a^{3/2} P_p}{a^{1/2} \{ \text{SIGN}[(1 - \cos E) S_p - S_L(1 - \cos \delta) S_q] - \frac{3}{2} P_p \}}$$

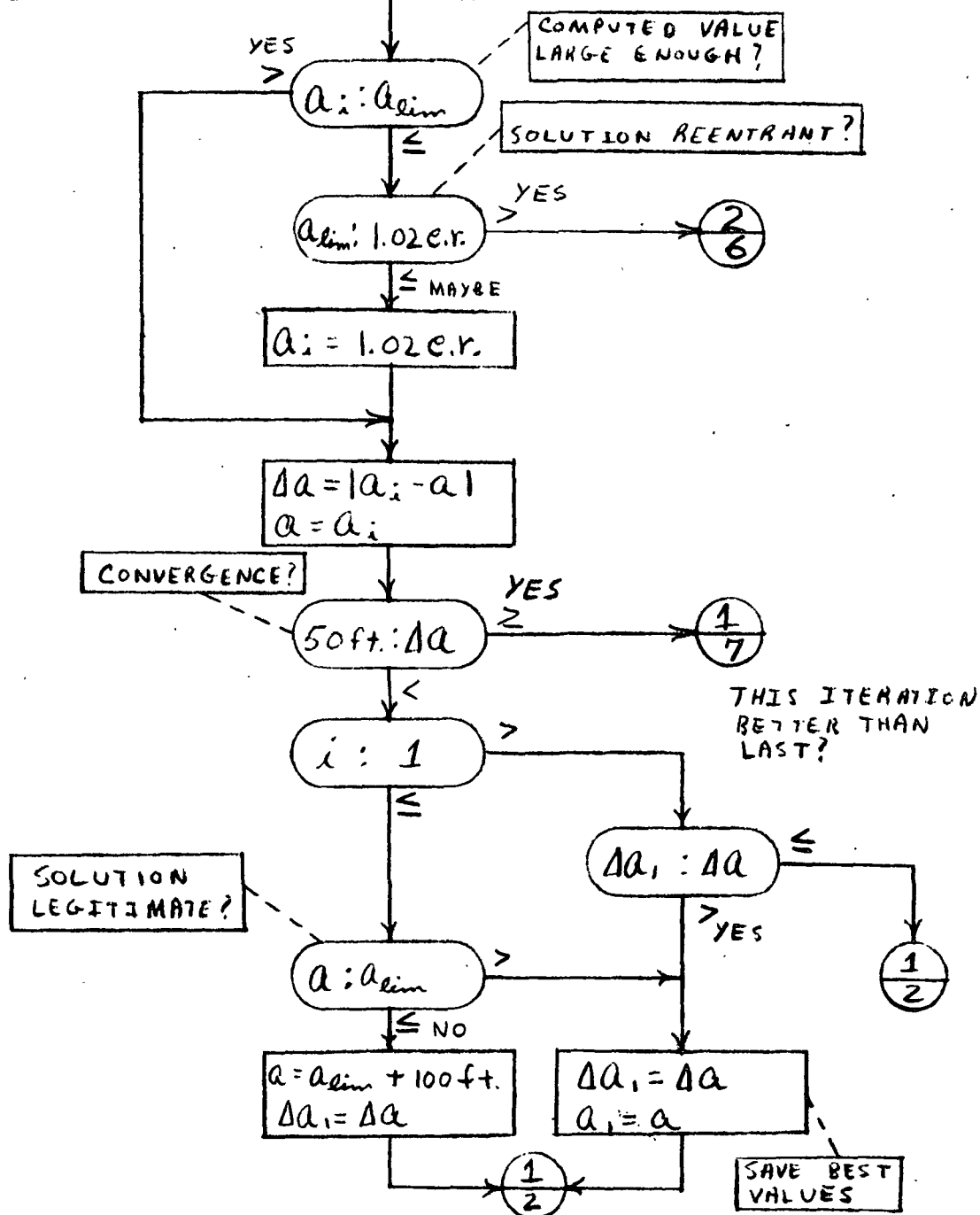


Figure PMSTICN - PCZYCF-1. Flowchart
(Sheet 3 of 9)

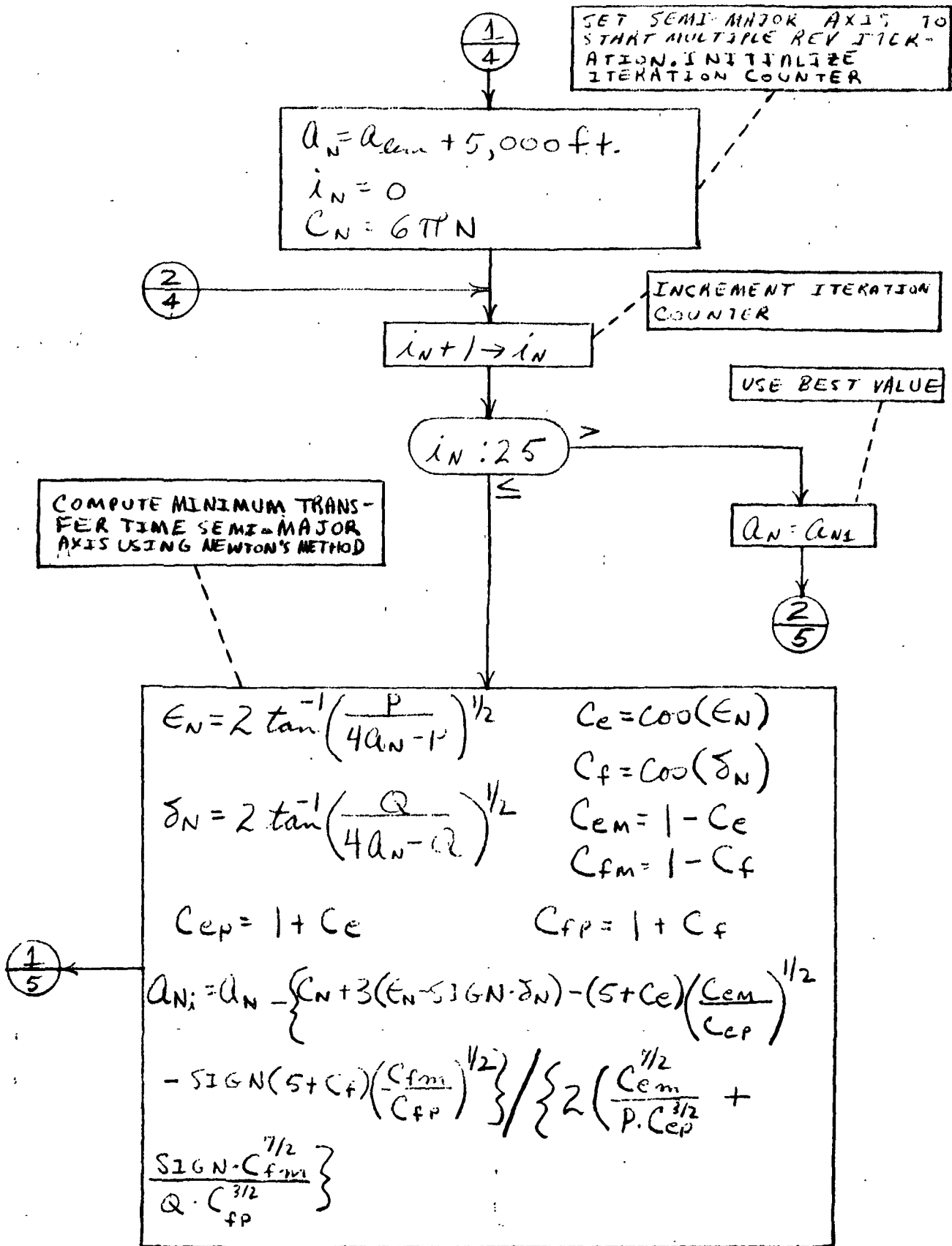
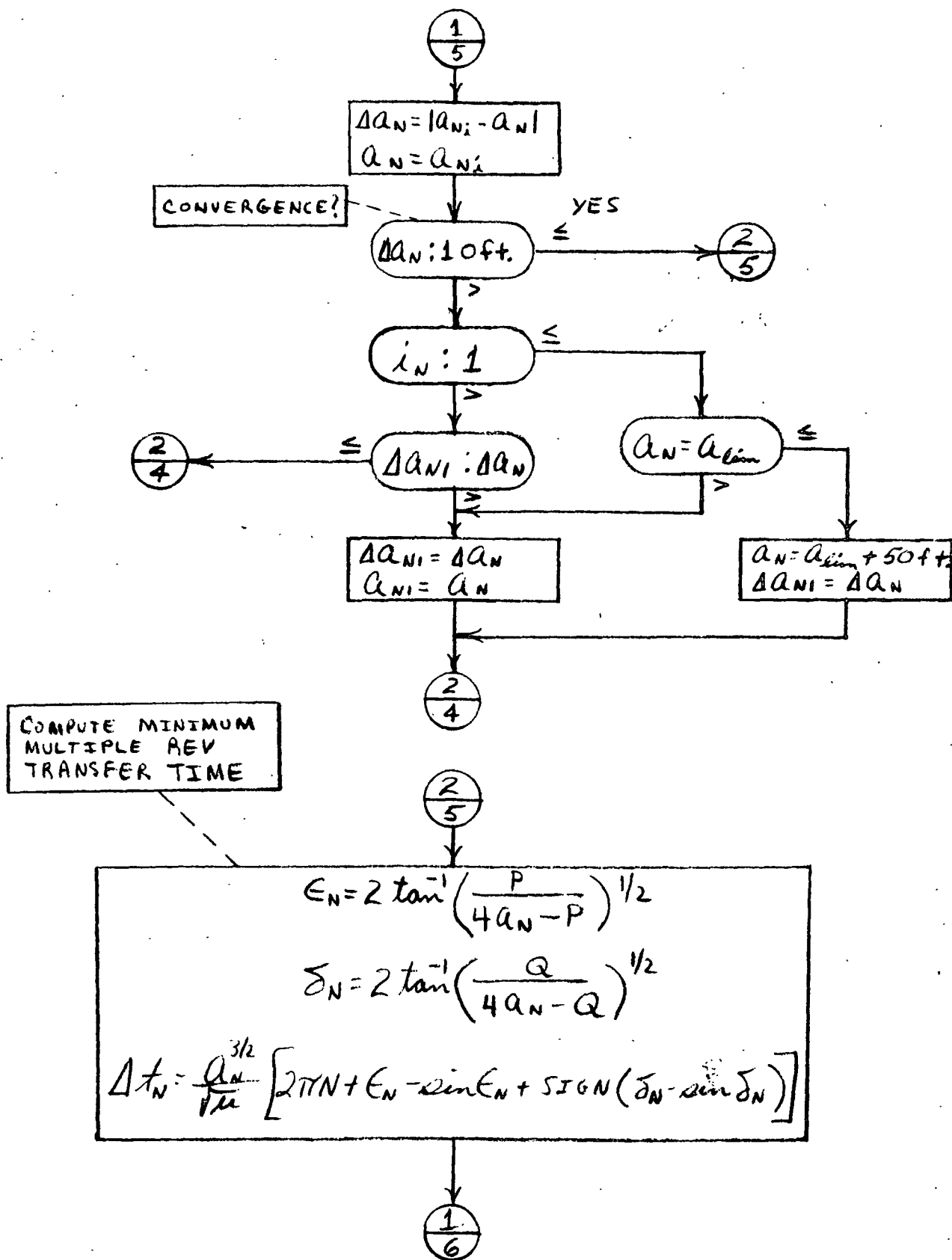


Figure PMSTICN - PCZYCF-1. Flowchart
(Sheet 4 of 9)


 Figure PMSTICN - PCZYCF-1. Flowchart
(Sheet 5 of 9)



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMSTICN -

PCZYCF

DATE 7/28/67 (EORP)

REV

PAGE 13 of 16

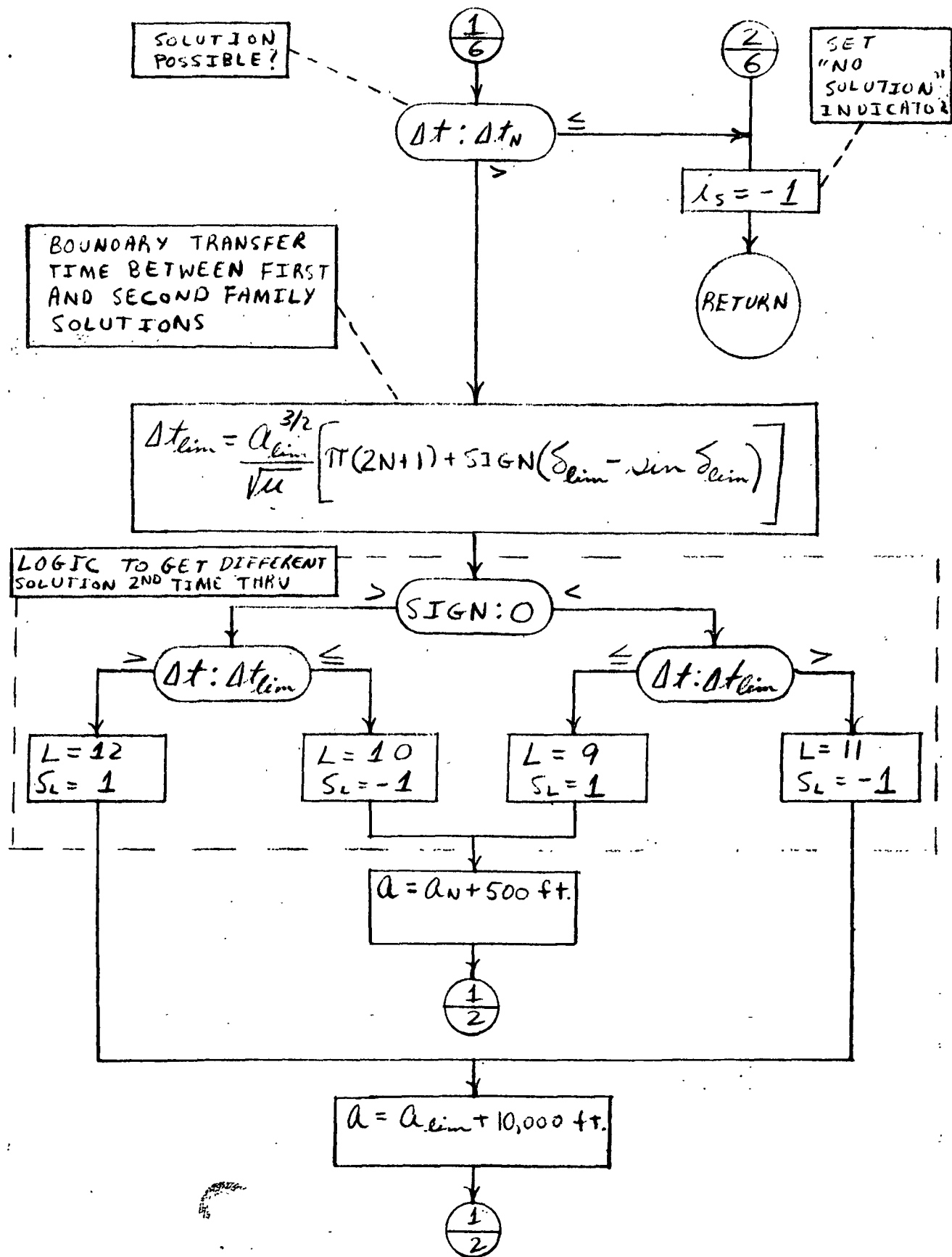


Figure PMSTICN - PCZYCF-1. Flowchart (Sheet 6 of 9)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMSTICN -

PCZYCF

DATE 7/28/67 (CORP)

REV

PAGE 14 of 16

ECCENTRICITY AND TIME
PAST PERICENTER AFTER
TRANSFER INITIATION

1
7

$$e = \frac{\left\{ \left[\frac{R_1 - a - (R_2 - a) \cos(\epsilon - \delta)}{\sin(\epsilon - \delta)} \right]^2 + (a - R_2)^2 \right\}^{1/2}}{a}$$

$$P_T = a(1 - e^2)$$

$$(\cos f)_T = \frac{P_T - R_2}{e \cdot R_2}$$

$$(\sin f)_T = \frac{(\cos f)_T \cos \phi_T - \frac{P_T - R_1}{e \cdot R_1}}{\sin \phi_T}$$

$$f_T = \tan^{-1} \left[\frac{(\sin \phi)_T}{(\cos \phi)_T} \right], 0 \leq f_T < 2\pi$$

$$t_{PT} = \frac{a P_T^{1/2}}{\sqrt{\mu}} \left[\frac{2}{1 - e^2} \tan^{-1} \left(\sqrt{\frac{1 - e}{1 + e}} \tan \frac{f_T}{2} \right) - \frac{e(\sin f)_T}{1 + e(\cos f)_T} \right]$$

"SOLUTION
INDICATOR"

$i_s = 1$

Mod time by
ONE PERIOD

$t_{PT} : 0$

$\geq$

$<$

$$\frac{2\pi a^{3/2}}{\sqrt{\mu}} + t_{PT} \rightarrow t_{PT}$$

1
8

Figure PMSTICN - PCZYCF-1. Flowchart
(Sheet 7 of 9)

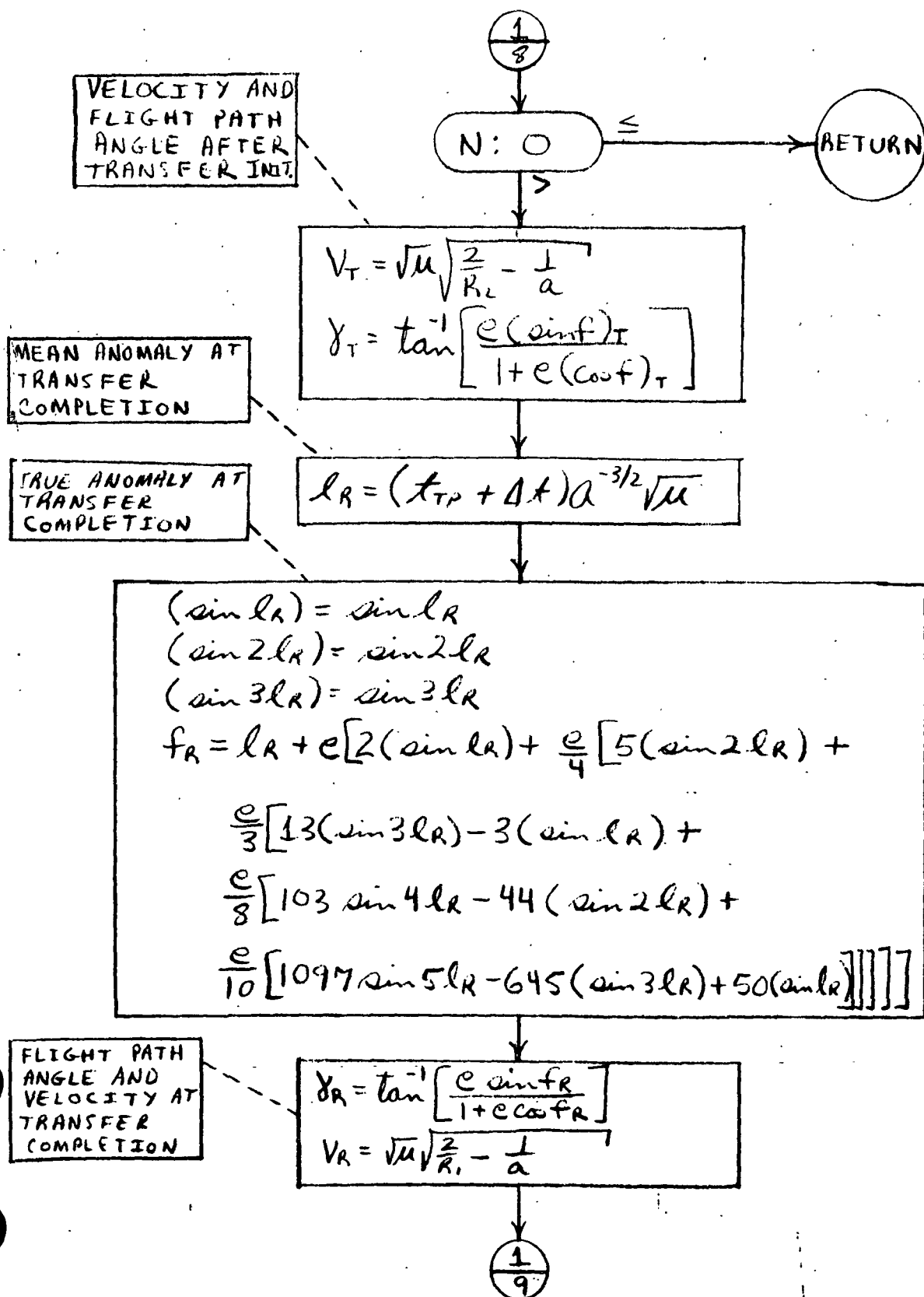


Figure PMSTICN - PCZYCF-1. Flowchart
(Sheet 8 of 9)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMSTICN -
PCZYCF

DATE 7/28/67 (EORP)

REV

PAGE 16 of 16

TOTAL VELOCITY REQUIRED
FOR TRANSFER
$$\frac{1}{9}$$

$$\Delta V_t = \left[(V_T \cos \gamma_T - V_{xT})^2 + (V_T \sin \gamma_T - V_{yT})^2 \right]^{1/2} + \left[(V_{xR} - V_R \cos \gamma_R)^2 + (V_{yR} - V_R \sin \gamma_R)^2 \right]^{1/2}$$

WHERE V_{xT} = INPUT HORIZONTAL VELOCITY
BEFORE TRANSFER INITIATION

V_{yT} = INPUT RADIAL VELOCITY
BEFORE TRANSFER INITIATION

V_{xR} = INPUT HORIZONTAL VELOCITY
AFTER TRANSFER COMPLETION
MANEUVER

V_{yR} = INPUT RADIAL VELOCITY
AFTER TRANSFER COMPLETION
MANEUVER

SECOND SOLUTION?

K:0

YES

OUTPUT
LOWER ΔV
SOLUTION $i_s = 1$

$a' = a$
 $e' = e$
 $f_{T1}' = f_T$
 $t_{PT1}' = t_{PT}$
 $\Delta V_{T1}' = \Delta V_T$
 $K=1, i=0$

SAVE FIRST
SOLUTIONSET INDICATOR
TO GET OTHER
SOLUTION $\Delta V_{T1} : \Delta V_T$

$a = a'$
 $e = e'$
 $f_T = f_{T1}'$
 $t_{PT} = t_{PT1}'$

RETURN

L:10

L:12

 $S_L = -1$ $S_L = 1$ $a = a_n + 10,000 \text{ ft.}$

$$\frac{1}{2}$$
Figure PMSTICN - PCZYCF-1. Flowchart
(Sheet 9 of 9)

IBM RTCC APOLLO PROGRAMMING SYSTEMS
 2. PMSVCT
 DATE 10/23/67 (EORP)
 REV
 PAGE 1 of 19

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

- a. Symbol — PMSVCT
- b. Program Name — Vector Routing Load Module
- c. Programming Group — Maneuver Control (Mission Planning)
- d. Programmer — B. Oliver
- e. Language — Assembler
- f. Data Tables Used — PZMPTCSM, PZMPTLEM, GZTSCNS1, GZTSCNS3
- g. MED's Used — None
- h. Writeup Completion Date — 9/29/67
- i. Parameter Lists — Tables PMSVCT-1 through -5
- j. Flowchart — Figure PMSVCT-1

FUNCTION

The primary function of the vector routing processing is to select vectors from the current vehicle ephemeris for the purpose of performing an ephemeris update and to initiate trajectory updates as the result of the selection of new vehicle anchor vectors.

USAGE

- a. Calling Sequence — RTATTACH TASK = PTMANC, EP = PMSVCT, ID = n, PARAM = (ADDR, LENGTH)
- b. Inputs — Parameter lists (Tables PMSVCT-1 through -3)
- c. Outputs and Exit — Routes vector for trajectory update via XCTL to EMSTRAJ
- d. Other Programs Used — ELGETE, EMGVECST, PMXSPT, EMSTRAJ
- e. Data Restrictions and Limits — None
- f. Error — If any error conditions are encountered, an appropriate comment is printed on line, and the program exists.

METHOD

The Vector Routing Load Module will be entered as the result of various maneuver control functions. These functions result in a change to the Mission Plan Table (MPT) data that affects the generation of the current CSM or LM ephemeris. These functions include the following:

- a. The insertion, deletion, or change of an MPT maneuver
- b. The change of the current orbit density multiplier (K-factor)
- c. The change of a vehicle weight or cross-sectional area.

In addition, PMSVCT will be entered to perform an "ephemeris extension" when the length of the future ephemeris fails to meet the requirement as specified in TR165.

Approval	Approval
R.E. Oliver	S.C. James



Since these functions do not affect the anchor vector for the ephemeris, a present position vector based on the current anchor vector is selected from the vehicle ephemeris. This vector will be the latter of : (1) the burnout vector of the last executed maneuver, (2) the first free-flight three-minute vector prior to present time, or (3) the present anchor vector. The vector selected is then routed to EMSTRAJ, and a trajectory update is performed.

When a new anchor vector is selected for the orbit ephemeris, it is routed to PMSVCT for the purpose of initiating a trajectory update. If the anchor vector is within the range of a maneuver in the Mission Plan Table, the following action is taken:

- a. The end time of the maneuver in the MPT header is set equal to the time of the anchor vector
- b. The burnout vector in the maneuver description block of the MPT is set equal to the anchor vector
- c. An indicator in the maneuver description block of the MPT is set.

In all cases, the time of the anchor vector is set in the MPT, and the vector is routed to EMSTRAJ to complete the trajectory update.

IBM RTCC APOLLO PROGRAMMING SYSTEMS
 2. PMSVCT
 DATE 10/23/67 (EORP)
 REV
 PAGE 3 of 19

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 Table PMSVCT-1. Parameter List
 RTATTACH

QUEID — * (0-7)

QNAME — NA

From — GMSCTRL

To-Task — PTMANC

 Purpose — Initiate a trajectory update
 with the vector transmitted

EP — PMSVCT

Byte No.	Type*	Units	Value / Description / Comments
1-4	F		* (0-7)
5-8	F		Vehicle index
9-16	D	hours	Time of vector
17-24	↓	Er	X
25-32		Er	Y
33-40		Er	Z
41-48		Er/hr	$\dot{X}$
49-56		Er/hr	$\dot{Y}$
57-64		Er/hr	$\dot{Z}$
65-72	↓		Source ID
			* 0 Nominal insertion
			1 Actual insertion
			2 Restart
			3 Launch override
			4 DC
			5 High-speed radar
			6 Telemetry
			7 Maneuver confirmation history vector

 *Type
 H — Fixed-point half word
 F — Fixed-point full word

 E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMSVCT
DATE 10/23/67 (EORF)
REV
PAGE 4 of 19Table PMSVCT-2. Parameter List

XCTL

QUEID — 8

QNAME — common

From — PMMXFR, PMMWTC, PMMFUD

To-Task — PTMANC

Purpose — Initiate a trajectory update(s) as a
result of maneuver control action

EP — PMSVCT

Byte No.	Type*	Units	Value / Description / Comments
1-4			Open
5-8	F		Vehicle for first update
9-12	F		Vehicle for second update (= zero if none)

Table PMSVCT-3. Parameter List

RTATTACH

QUEID — 17

QNAME — GQ3CTL

From — GTMSCTRL

To-Task — PTMANC

Purpose — Ephemeris extension request

EP — PMSVCT

Byte No.	Type*	Units	Value / Description / Comments
1-4			Open
5-8	F		Vehicle index

*Type
H — Fixed-point half word
F — Fixed-point full word

E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

Table PMSVCT-4. Parameter List

LINK

QU EID —

QNAME —

From - PMSVCT

To-Task —

Purpose — Obtain the orbit ephemeris
anchor vector

EP - EMGVECST

Byte No.	Type*	Units	Value / Description / Comments
1-4	F	hr	7 - ID
5-8	F		Vehicle code
9-16	D		Time of anchor vector

*Type
H - Fixed-point half word
F - Fixed-point full word

E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS
 2. PMSVCT
 DATE 10/23/67 (EORE)
 REV
 PAGE 6 of 19

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

Table PMSVCT-5. Parameter List

XCTL

QUEID — NA

QNAME — NA

From — PMSVCT

To-Task — PTMANC

Purpose — Perform orbit trajectory update

EP — EMSTRAJ

Byte No.	Type*	Units	Value / Description / Comments
1-4	F		Address of work area in subpool 1
5-8	F		Not used
9-12	F		Action code for trajectory update
			0 Nominal insertion
			1 Actual insertion
			2 Restart
			3 Launch override
			4 DC
			5 High-speed radar
			6 Telemetry
			7 Maneuver confirmation
			8 Ephemeris extension/Maneuver control action
13-16	F		VEH1 - Vehicle 1 ID
17-24	D		Time of vector
25-32	D	Er	X
33-40	D	Er	Y
41-48	D	Er	Z
49-56	D	Er/hr	$\dot{X}$
57-64	D	Er/hr	$\dot{Y}$
65-72	D	Er/hr	$\dot{Z}$
73-80	D		Station ID
81-152			Same as 9-80 for Vehicle 2
153-160			Not used
161-168			Zero

*Type

H — Fixed-point half word
F — Fixed-point full wordE — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

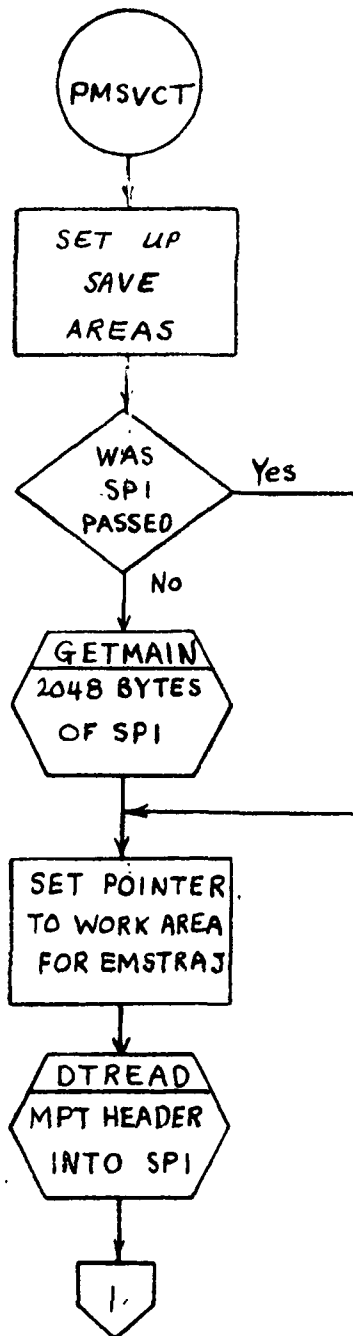


Figure PMSVCT-1. Flowchart

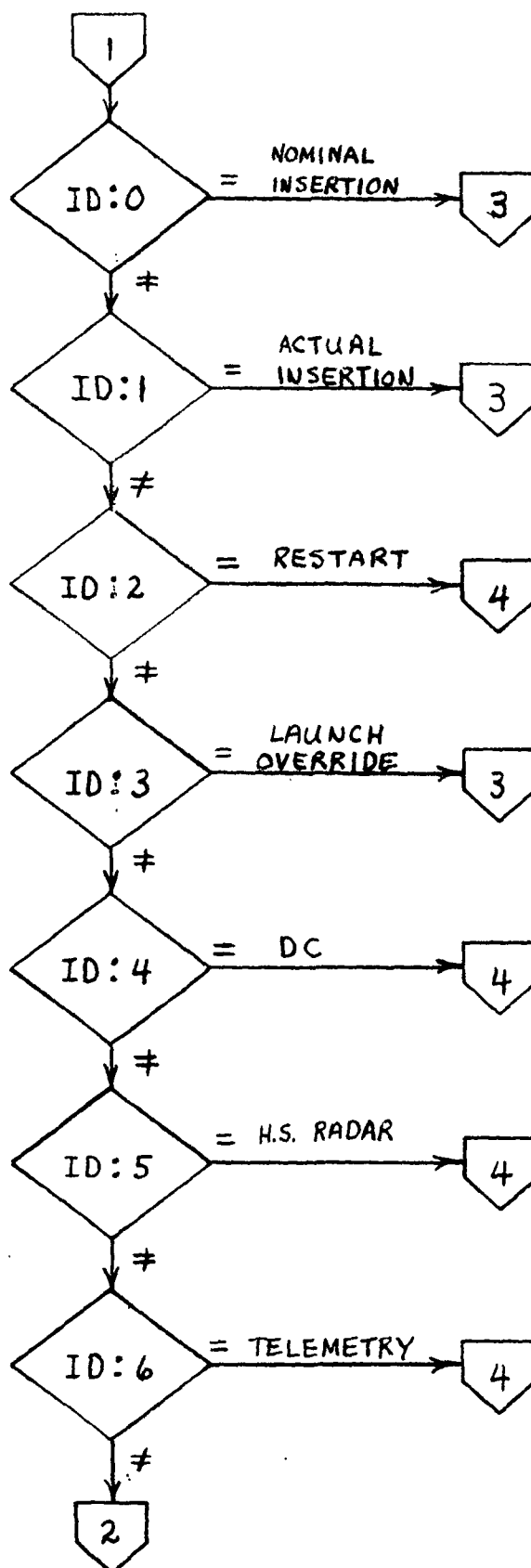


Figure PMSVCT-1. Flowchart

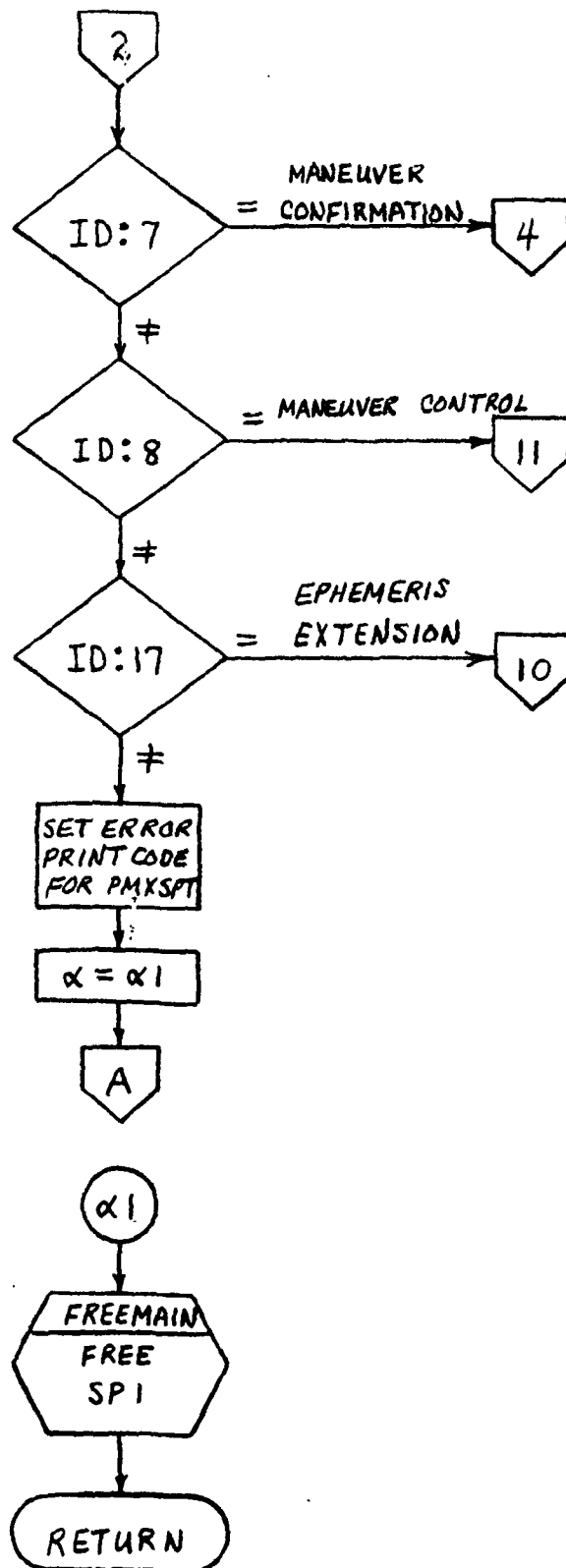


Figure PMSVCT-1. Flowchart

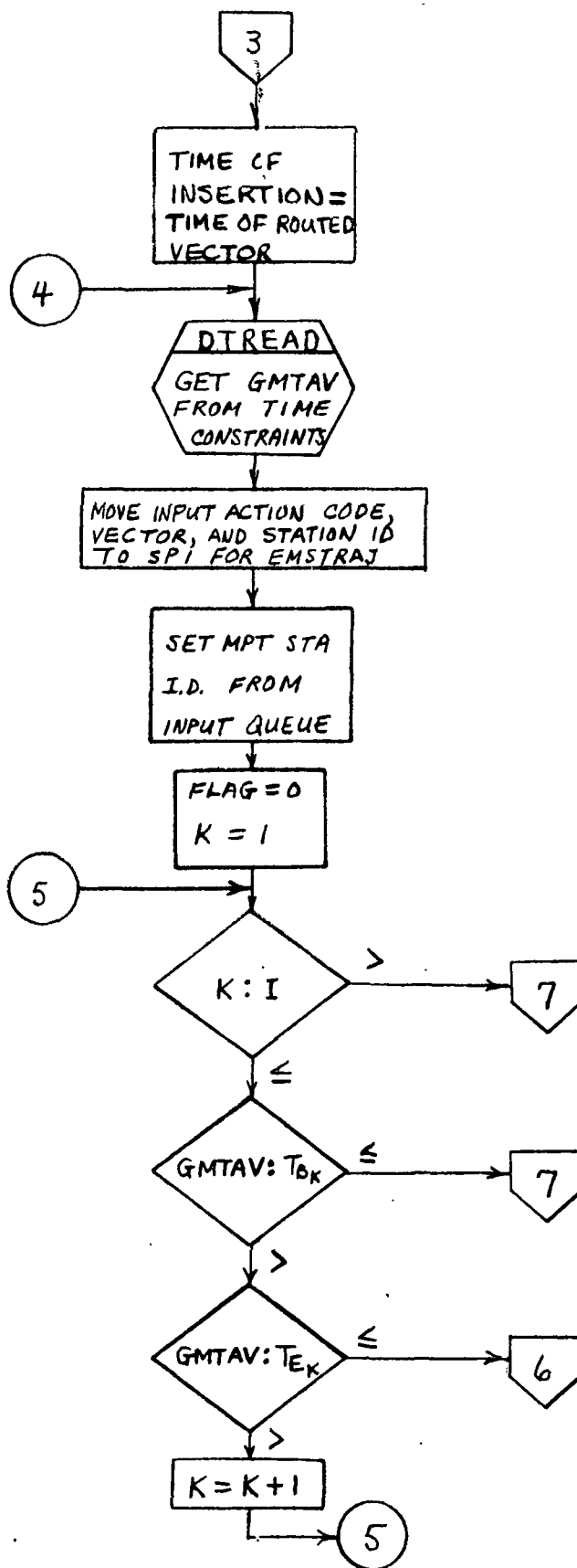


Figure PMSVCT-1. Flowchart

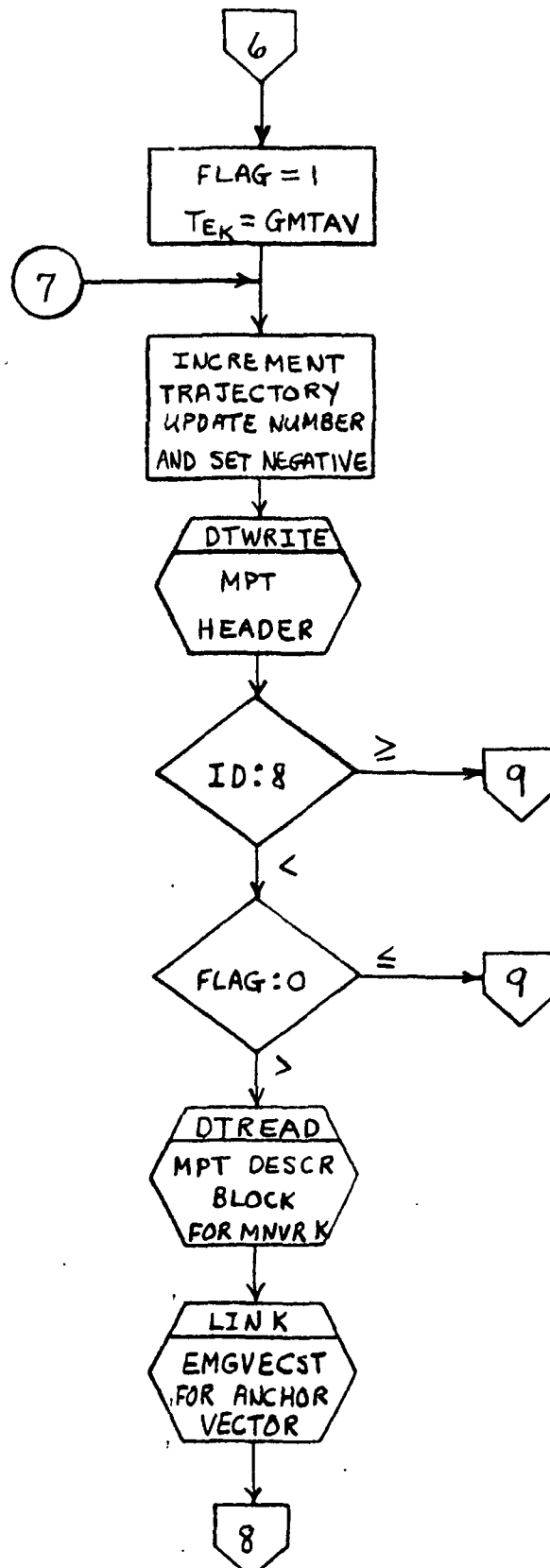


Figure PMSVCT-1. Flowchart

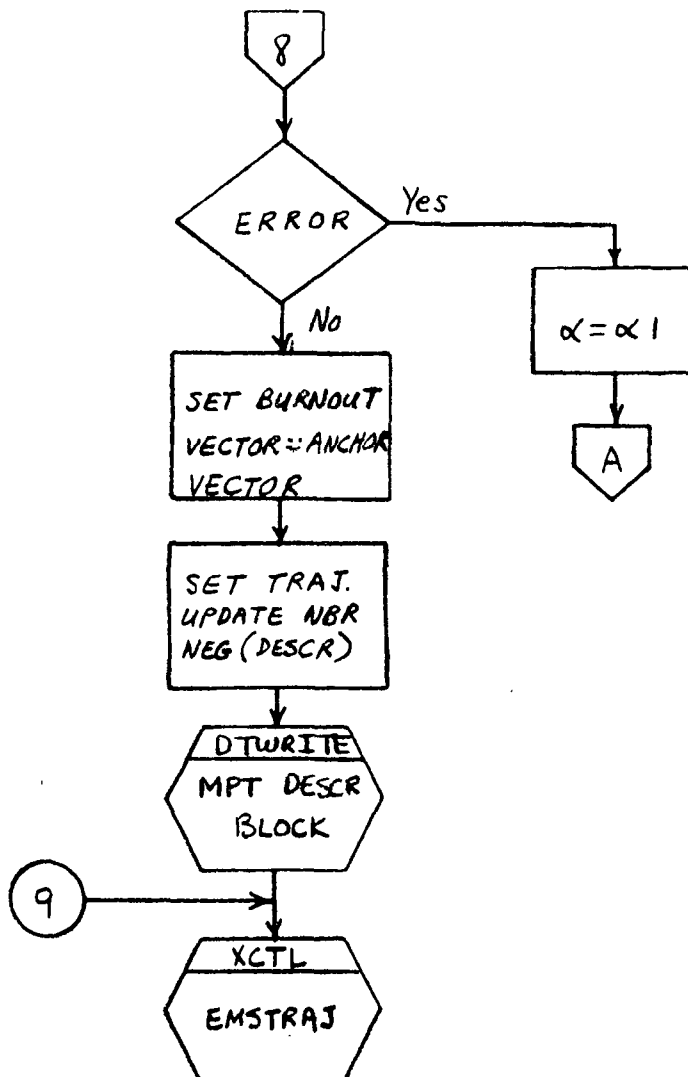


Figure PMSVCT-1. Flowchart

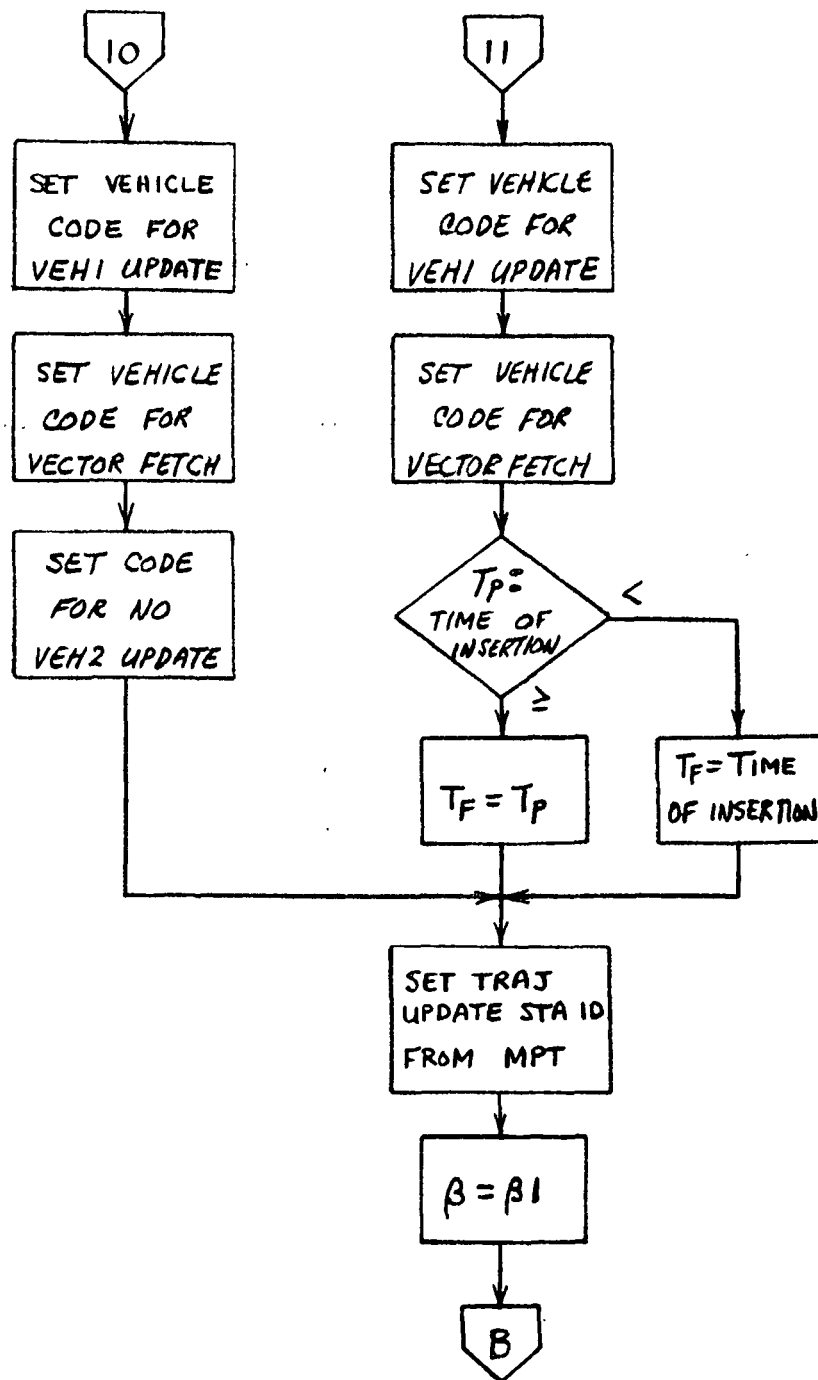


Figure PMSVCT-1. Flowchart

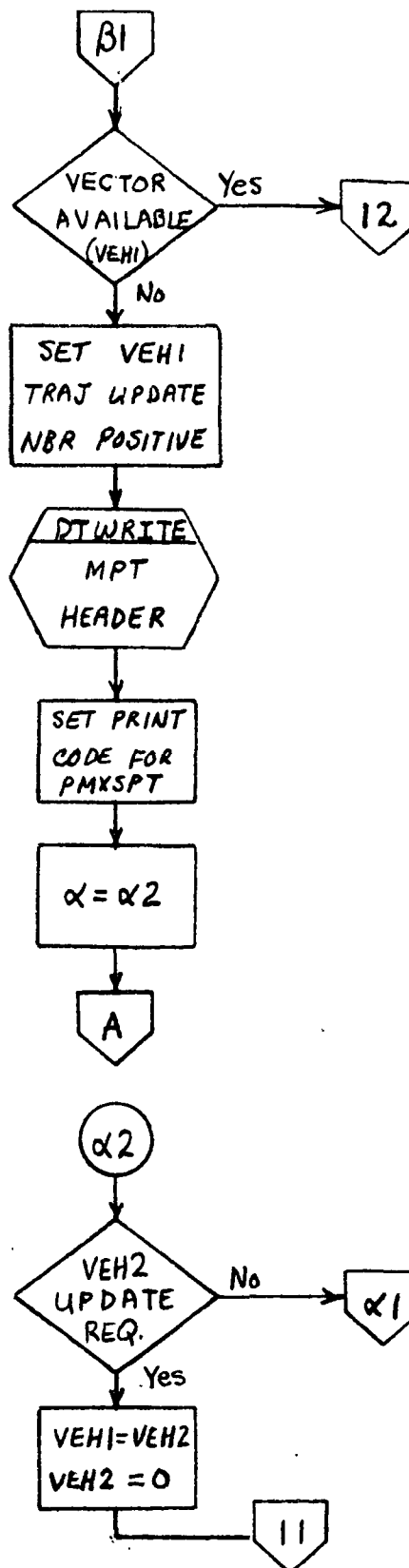


Figure PMSVCT-1. Flowchart

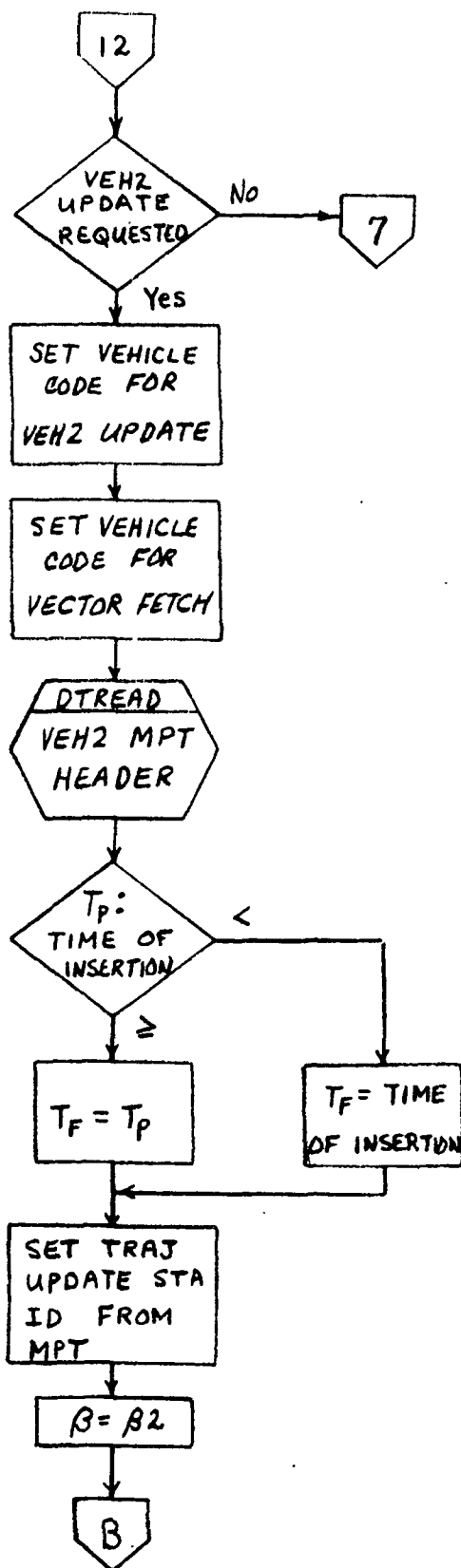


Figure PMSVCT-1. Flowchart

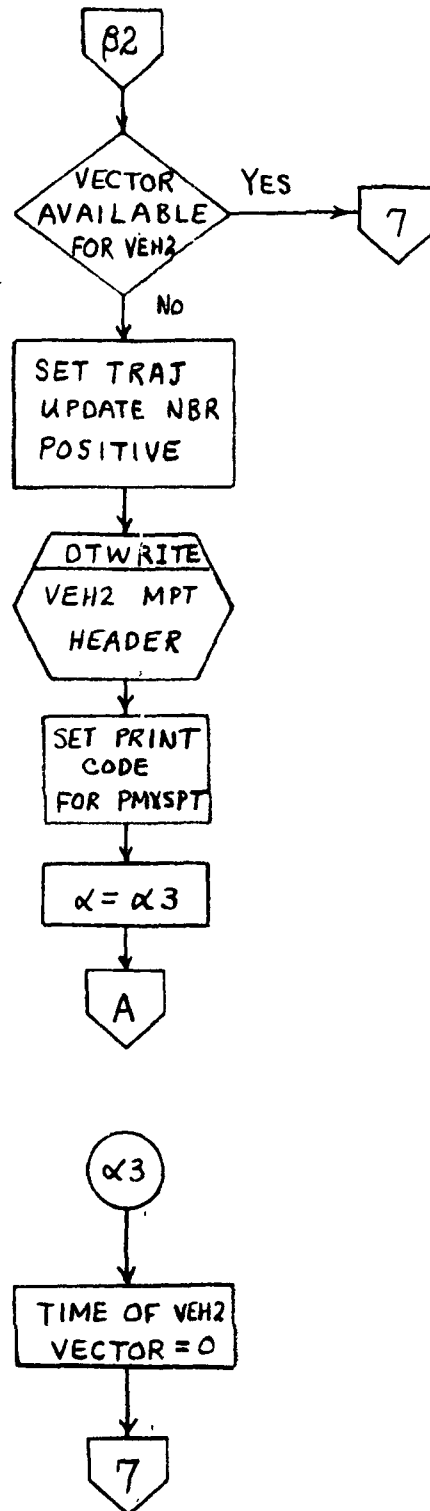


Figure PMSVCT-1, Flowchart

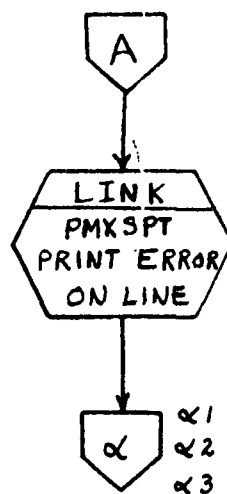


Figure PMSVCT-1. Flowchart

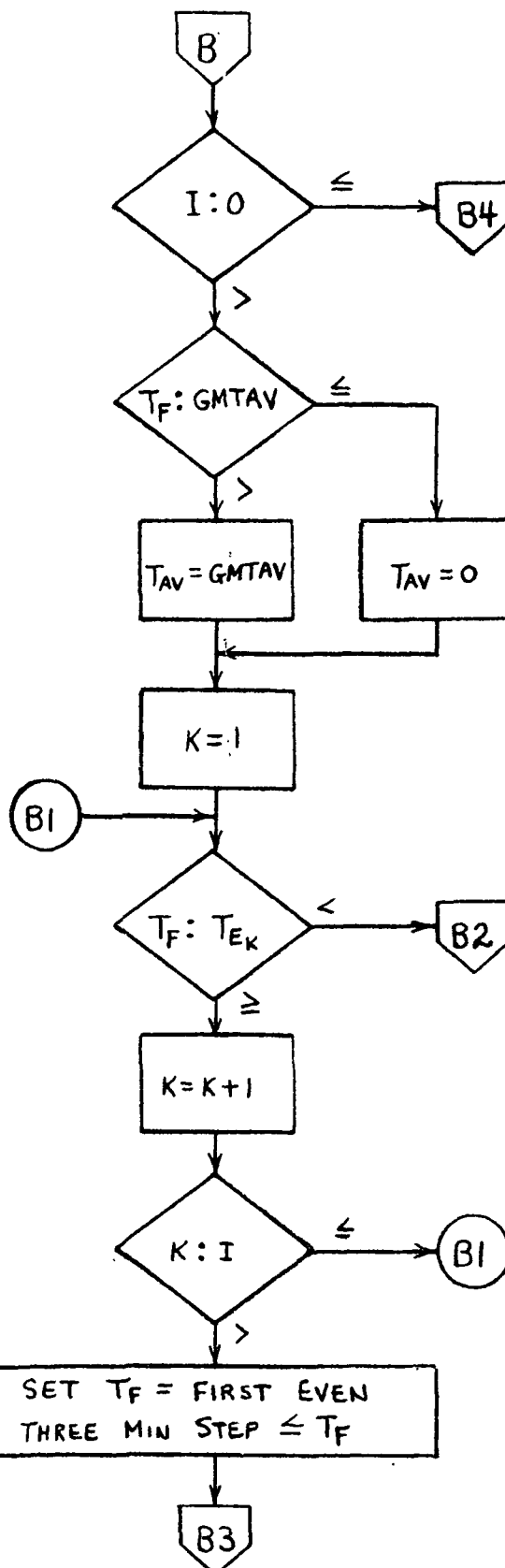


Figure PMSVCT-1. Flowchart

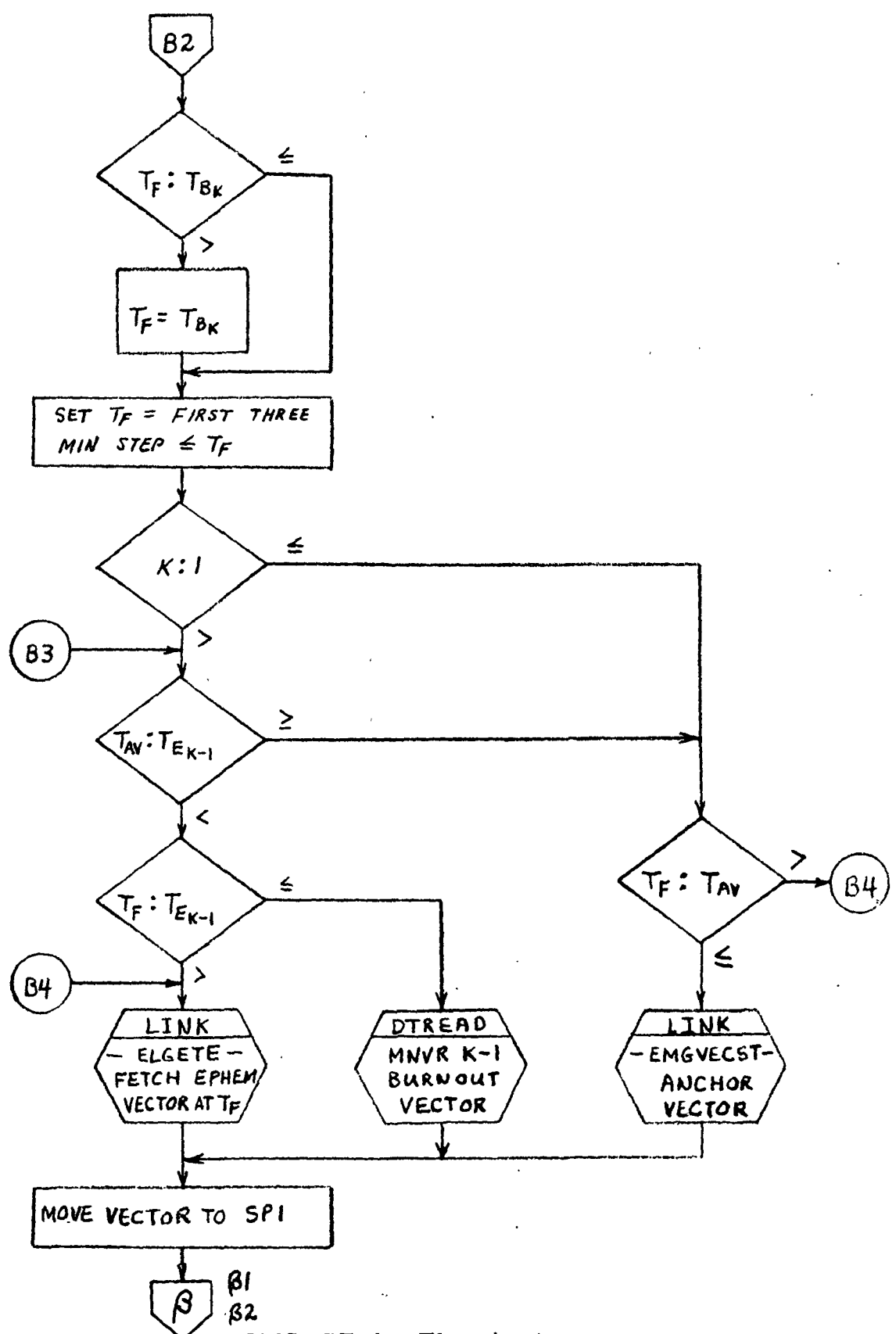


Figure PMSVCT-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMSVEC
 DATE 12/13/67 (EORP)
 REV
 PAGE 1 of 8

- a. Symbol — PMSVEC
- b. Program Name — Vector Fetch Load Module
- c. Programming Group — Rendezvous Planning
- d. Programmer(s) — T. Hickey
- e. Language — Assembler
- f. Data Tables Used — PZMPTCSM, PZMPTLEM, PZMPTS4B
- g. MED's Used — K00, K01, K20-K24, K28, K30, K32, K45
- h. Writeup Completion Date — 11/17/67
- i. Parameter List — Table PMSVEC-1 and 2
- j. System Parameter — Table PMSVEC-3
- k. Flowchart — Figure PMSVEC-1

FUNCTION

The function of this load module is to get one or two sets of Keplerian elements, with the associated time, area, weight, K-factor, and station ID, ensuring that the time of the elements meets all ground rules set up for maneuvers by the particular vehicle involved.

USAGE

- a. Calling Sequence — RTATTACH EP=PMSVEC for K20-K24, K28, K30, K32 or K45; LINK EP=PMSVEC for K00 and K01.
- b. Inputs — Vehicle code(s), Queue ID, TIME(s) to get vector for appropriate vehicle(s) - See input block Table PMSVEC 1.
- c. Outputs and Exit — See output block Table PMSVEC-2. Exit by "XCTL" = (LM Name) for K20-K24, K28, K30, K32 or K45; "RETURN" for all other queues.
- d. Other Programs Used — ELGETE, GIMIKC, PMXSPT;
- e. Data Restrictions and Limits — Time(s) of requested vectors must fall within range of corresponding ephemeris.
- f. Error — Error message printed when no vector is available, and error indicator set for K00 and K01 MED's; no queue (XCTL) issued for any other MED's - RETURN is issued.

METHOD

Since this load module does no computations, but is strictly a logic and utility type program, no method is presented here, and the reader is directed to the flow-chart for further information.

Approval	Approval
<i>Thomas M. Hickey</i>	<i>A. Amietti</i>



RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMSVEC
 DATE 12/13/67 (EORE)
 REV
 PAGE 2 of 8

Table PMSVEC-1. Parameter List

QUEID —

QNAME —

From — All Load Modules

To-Task —

Purpose — Input Block

EP —

Byte No.	Type*	Units	Value / Description / Comments
1-4	F	--	Code = 1, 2, or 3 - Maneuvering Vehicle Code
5-8			Spare
9-16	D	hrs.	Requested vector time of maneuvering vehicle
17-24	D	hrs.	Requested vector time of target vehicle (for K00, K01, K45, K30, and K32 MEDs)

*Type
 H — Fixed-point half word
 F — Fixed-point full word

E — Single-precision (floating-point)
 D — Double-precision (floating-point)
 X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMSVEC

DATE 12/13/67 (EORP)

REV

PAGE 3 of 8

Table PMSVEC-2. Parameter List

QUEID —

QNAME —

From — PMSVEC

To-Task —

Purpose — Output block to all
user load modules

EP —

Byte No.	Type*	Units	Value / Description / Comments
1-8	D	Er.	Chaser vehicle semimajor axis
9-16	D	--	Chaser vehicle eccentricity
17-24	D	rad.	Chaser vehicle inclination angle
25-32	D	rad.	Chaser vehicle mean anomaly
33-40	D	rad.	Chaser vehicle argument of perigee
41-48	D	rad.	Chaser vehicle ascending node
49-56	D	hrs.	Time of these Keplerian elements
57-64	D	--	K-Factor from MPT at this time
65-72	D	sq. ft.	Area of vehicle
73-80	D	lbs.	Weight of vehicle
81-88	D	--	Station ID in MPT
89-93	F	--	QUEID for this request
93-96	F	--	Spare (used by PMSVEC)
97-192			Same as above block, but for Target vehicle - used by MEDs K00, K01, K30, K32, and K45

*Type
H - Fixed point half word
F - Fixed-point full word

E - Single-precision (floating-point)
D - Double-precision (floating-point)
X - Hexadecimal

Table PMSVEC-3. System Parameters

[illegible]

*Type
H — Fixed-point half word
F — Fixed-point full word
S — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexidecimal

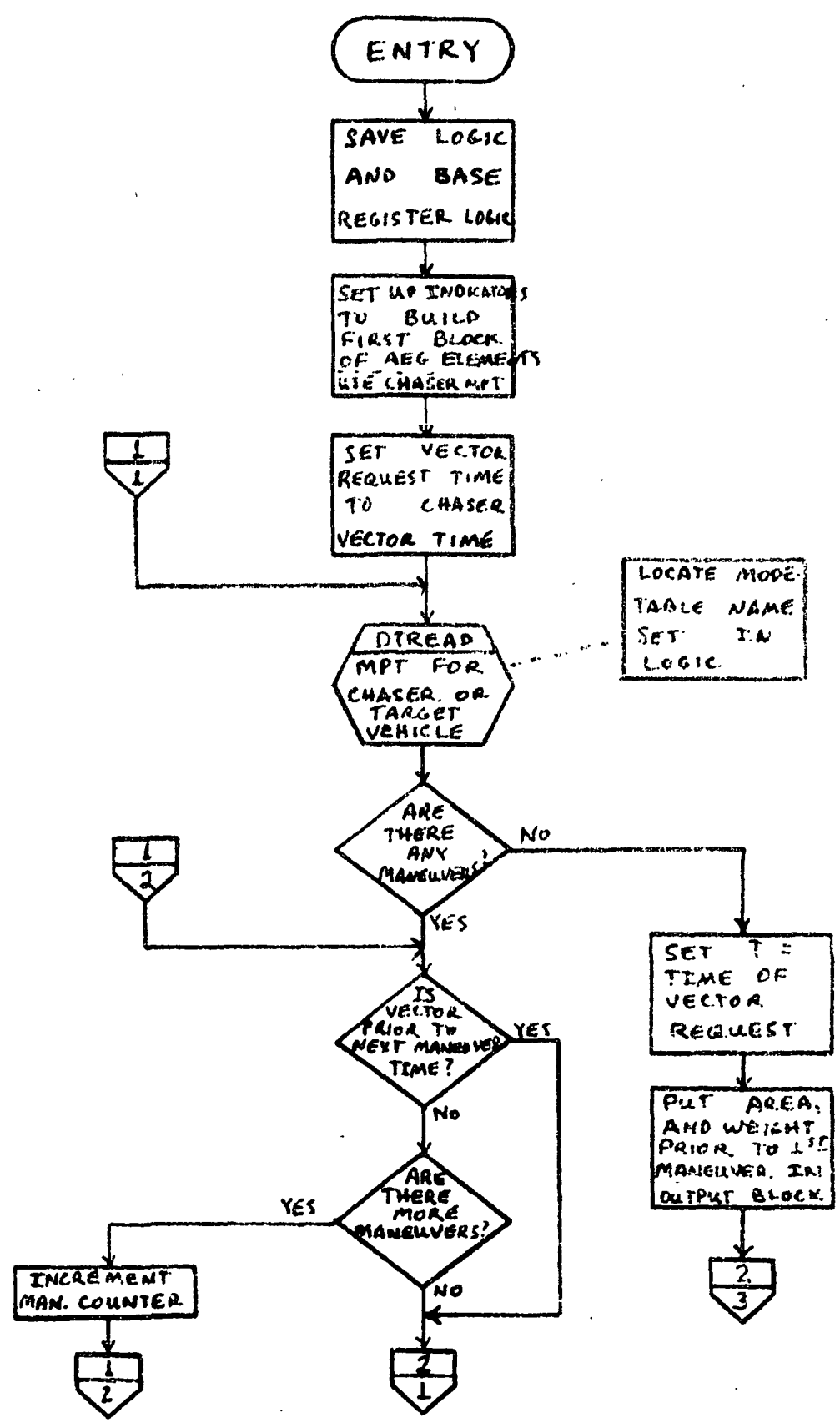


Figure PMSVEC-1. Flowchart

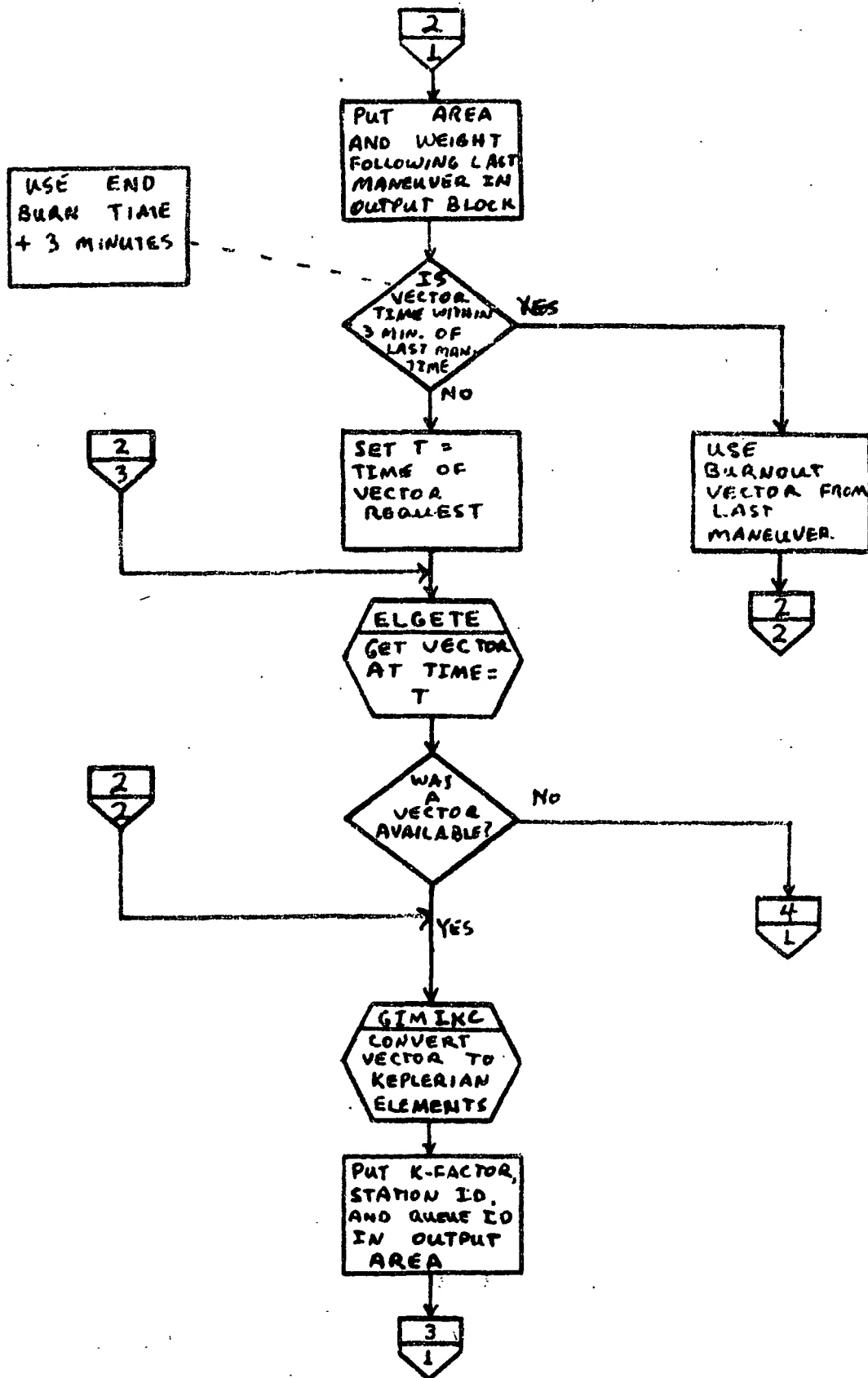


Figure PMSVEC-1. Flowchart

(Sheet 2 of 4)

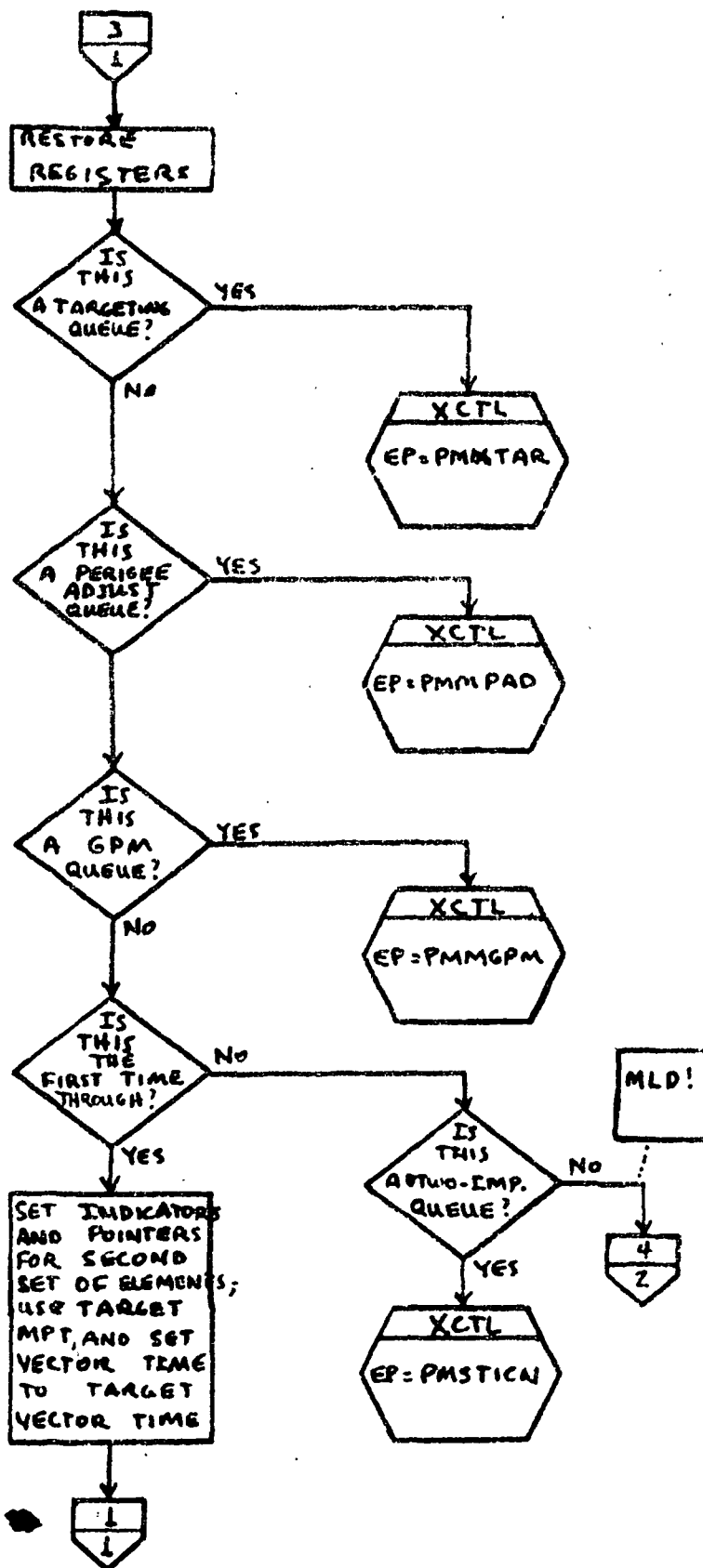


Figure PMSVEC-1. Flowchart

(Sheet 3 of 4)

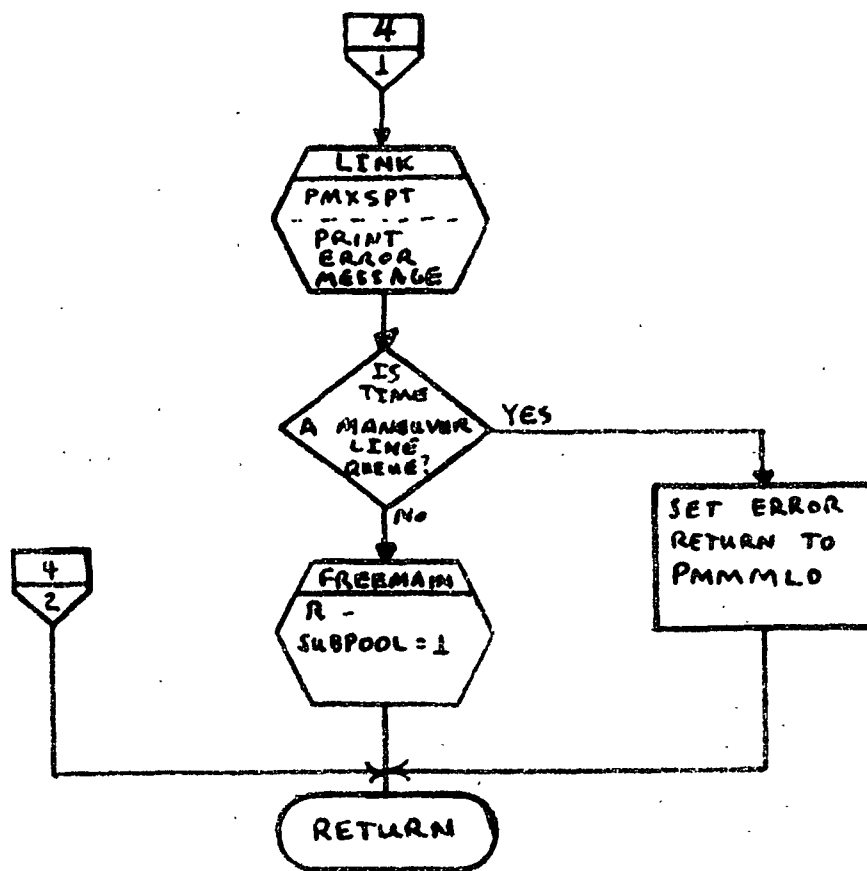


Figure PMSVEC-1. Flowchart

(Sheet 4 of 4)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMXSPT
DATE 10/23/67 (206)
REV
PAGE 1 of 3

- a. Symbol — PMXSPT
- b. Program Name — Mission Planning Print Load Module
- c. Programming Group — Maneuver Control
- d. Programmer — B. Oliver
- e. Language — Assembler
- f. Data Tables Used — None
- g. MED's Used — None
- h. Writeup Completion Date — 3/28/67
- i. Parameter List — Table PMXSPT-1
- j. Flowchart — Figure PMXSPT-1

FUNCTION

The function of this program is to print all Mission Planning related on-line messages.

USAGE

- a. Calling Sequence — LINK EP=PMXSPT, PARAM=(PRNT CODE, DATA)
- b. Inputs — Parameter list
- c. Outputs and Exit — The appropriate message is printed on line and an exit is made via RETURN (14, 12).
- d. Other Programs Used —; None
- e. Data Restrictions and Limits — None
- f. Error — If no message exists for the requested print code, XX, the message
"PMXSPT. ILLEGAL ENTRY=XX"
will be printed.
- g. Macros Used — RTCNV, RTPUT

METHOD

Messages are formatted for the on-line printer using display formatting language (DFL). The messages are then output to the printer using RTPUT access.

Approval	Approval
R.E. Oliver	S.C. James

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

2. PMXSPT
DATE 10/23/67 (206)
REV
PAGE 2 of 3Table PMXSPT-1. Parameter List

QUEID —

QNAME —

From — User

To-Task —

Purpose — Print on-line message

EP — PMXSPT

Byte No.	Type*	Units	Value / Description / Comments
1-2	H		Print code
5-8	F		Pointer to data

*Type
H — Fixed-point half word
F — Fixed-point full word

E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

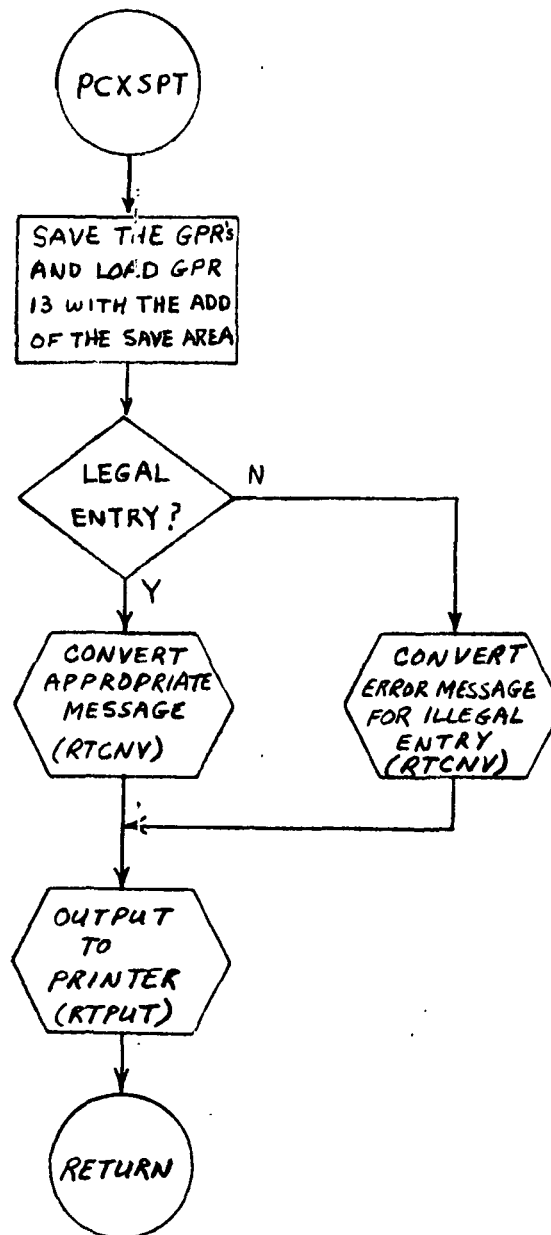


Figure PMXSPT-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMXTAR
 DATE 11/14/67 (EORP)
 REV
 PAGE 1 of 8

- a. Symbol — PMXTAR
- b. Program Name — Targeting Transmission Load Module
- c. Programming Group — Rendezvous Planning
- d. Programmer — T. Copper
- e. Language — Assembler
- f. Data Tables Used — PZSLVTAR, EZCCSM, EZCS4B
- g. MED's Used — K45
- h. Writeup Completion Date — 10/23/67
- i. System Parameters — Table PMXTAR-1
- j. Data Sent to Huntsville, Alabama — Table PMXTAR-2
- k. Flowchart — Figure PMXTAR-1

FUNCTION

PMXTAR generates a teletype format of the Saturn Launch Vehicle (SLV) targeting parameters and the current ephemeris data for the vehicle requested. It then transmits the format to Huntsville, Alabama.

USAGE

- a. Calling Sequence — XCNTL EP=PMXTAR
- b. Inputs — PZSLVTAR, EZCCSM, EZCS4B, SUBPOOL 1
- c. Outputs and Exit — Output is a teletype message sent to Huntsville, Alabama. Control is returned to the calling program.
- d. Other Programs Used — DTREAD, RTCNV, RTPUT, PMMAEG, GIMKI, DSQRT, DARSIN, DATANZ, DSIN, DCOS
- e. Data Restrictions and Limits — None
- f. Error — On-line print - if error is detected in Analytic Ephemeris Generator (AEG), processing will be terminated.

METHOD

The current SLV targeting parameters to be formatted are gathered from a data table (PZSLVTAR) generated by the SLV targeting load module (PMMPAR). Using the time input by the flight controller, a vector is fetched off the desired vehicle ephemeris. Using the geocentric inertial position and velocity vector, convert it to a spherical coordinate as shown below.

Given $x, y, z, \dot{x}, \dot{y}, \dot{z}$
 Compute $r, \varphi_c, \lambda, V, \gamma, \psi$

Approval	Approval
<i>T. Copper</i>	<i>A. Amlett</i>

where x, y, z - are the geocentric inertial position vector components

$\dot{x}, \dot{y}, \dot{z}$ - are the geocentric inertial velocity vector components

r - position magnitude

φ_c - geocentric latitude

λ - Earth fixed longitude

V - velocity magnitude

γ - flight path angle

ψ - azimuth

T - time of vector

W_e - Earth's rotation rate

Solution:

$$r = (x^2 + y^2 + z^2)^{1/2}$$

$$\varphi_c = \sin^{-1}(z/r)$$

$$\lambda = \tan^{-1}(y/x) - W_e T$$

$$V = (\dot{x}^2 + \dot{y}^2 + \dot{z}^2)^{1/2}$$

$$\gamma = \sin^{-1}\left(\frac{x\dot{x} + y\dot{y} + z\dot{z}}{rv}\right)$$

$$\psi = \tan^{-1}\left(\frac{x\dot{y} - y\dot{x}}{r\dot{z} - z\left(\frac{x\dot{x} + y\dot{y} + z\dot{z}}{r}\right)}\right)$$

The targeting and spherical coordinate data is formatted in a teletype format using the data format language and sent to Marshall Space Flight Center, Huntsville, Alabama.

Table PMXTAR-1. System Parameters

[illegible]

*Type
H — Fixed-point half word
F — Fixed-point full word
E — Single-precision (floating-point)
D — Double-precision (floating-point)
X — Hexadecimal

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

 2. PMXTAR
 DATE 11/14/67 (EORI)
 REV
 PAGE 4 of 8

Table PMXTAR-2. Data Sent to Huntsville, Alabama

Symb. Ref.	Byte No.	Type*	Units	Value / Description / Comments
TIME			STU	Time of Vector
GMTLO			STU	Time of First Vehicle Liftoff
REV				Revolution of Time
X			CUL	Position Component
Y			CUL	
Z			CUL	
X			CUV	Velocity Component
Y			CUV	
Z			CUV	
h_T			deg./sec.	Inertial Nodal Regression Rate
V MAG			kilometer	Inertial Velocity
			sec.	
GAMMA			deg.	Inertial Flight Path Angle
AZ			deg.	Heading Azimuth
RANGE			kilometer	Magnitude of Radius Vector
CALAT			deg.	Geocentric Latitude
CALOW			deg.	Earth Fixed Longitude
VIGM1			meters	Velocity at Orbital Insertion
			sec.	
VIGM2			meters	Cut-off Velocity
			sec.	
RIGM1			meters	Radius Magnitude at S-IVB Cutoff
YIGM1			degrees	Flight Path Angle at S-IVB Cutoff
IIGM1			degrees	Inclination of Targeted Orbit Plane
TIGM1			degrees	Angle Between Launch Pad Meridian and the Target Plan Orbit
TDIGM1			deg./sec.	Earth-fixed Nodal Regression Rate
AZL1			deg.	Flight Azimuth
TGRR1			STU	Predicted Time of Guidance Release

 *Type
 H - Fixed-point half word
 F - Fixed-point full word

 E - Single-precision (floating-point)
 D - Double-precision (floating-point)
 X - Hexadecimal

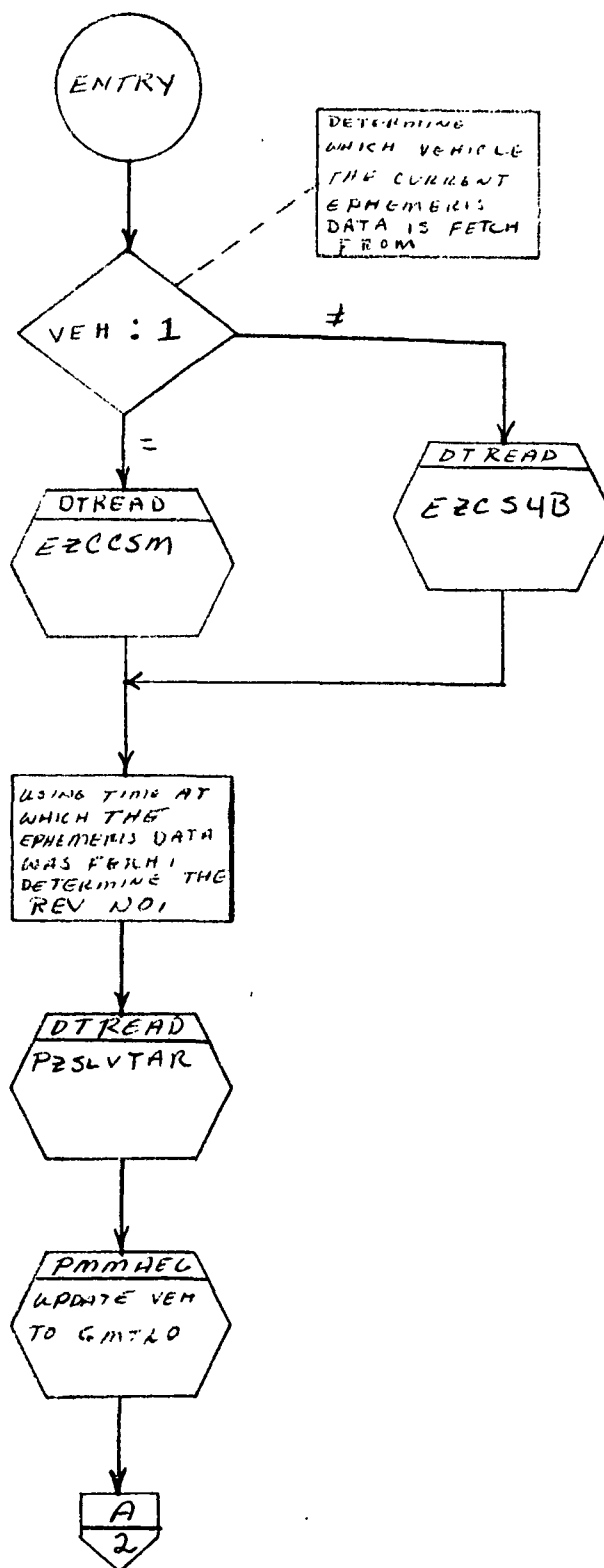


Figure PMXTAR-1. Flowchart

(Sheet 1 of 4)

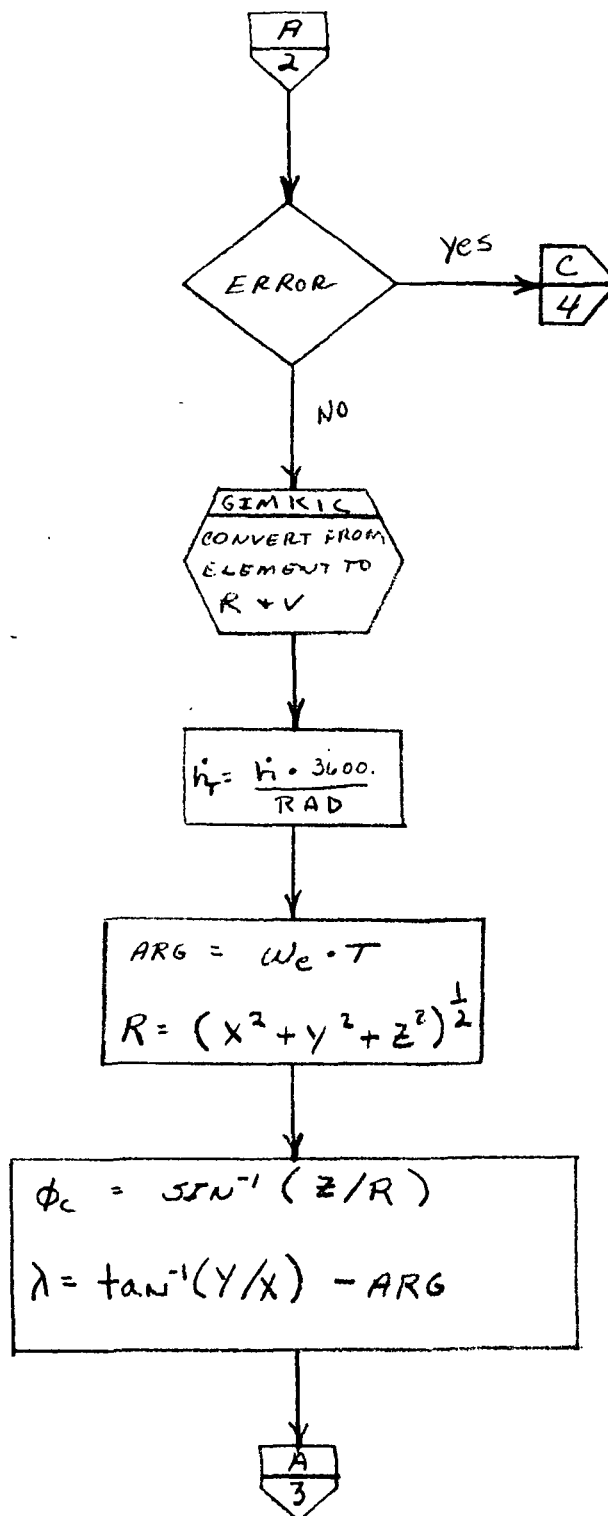


Figure PMXTAR-1. Flowchart

(Sheet 2 of 4)

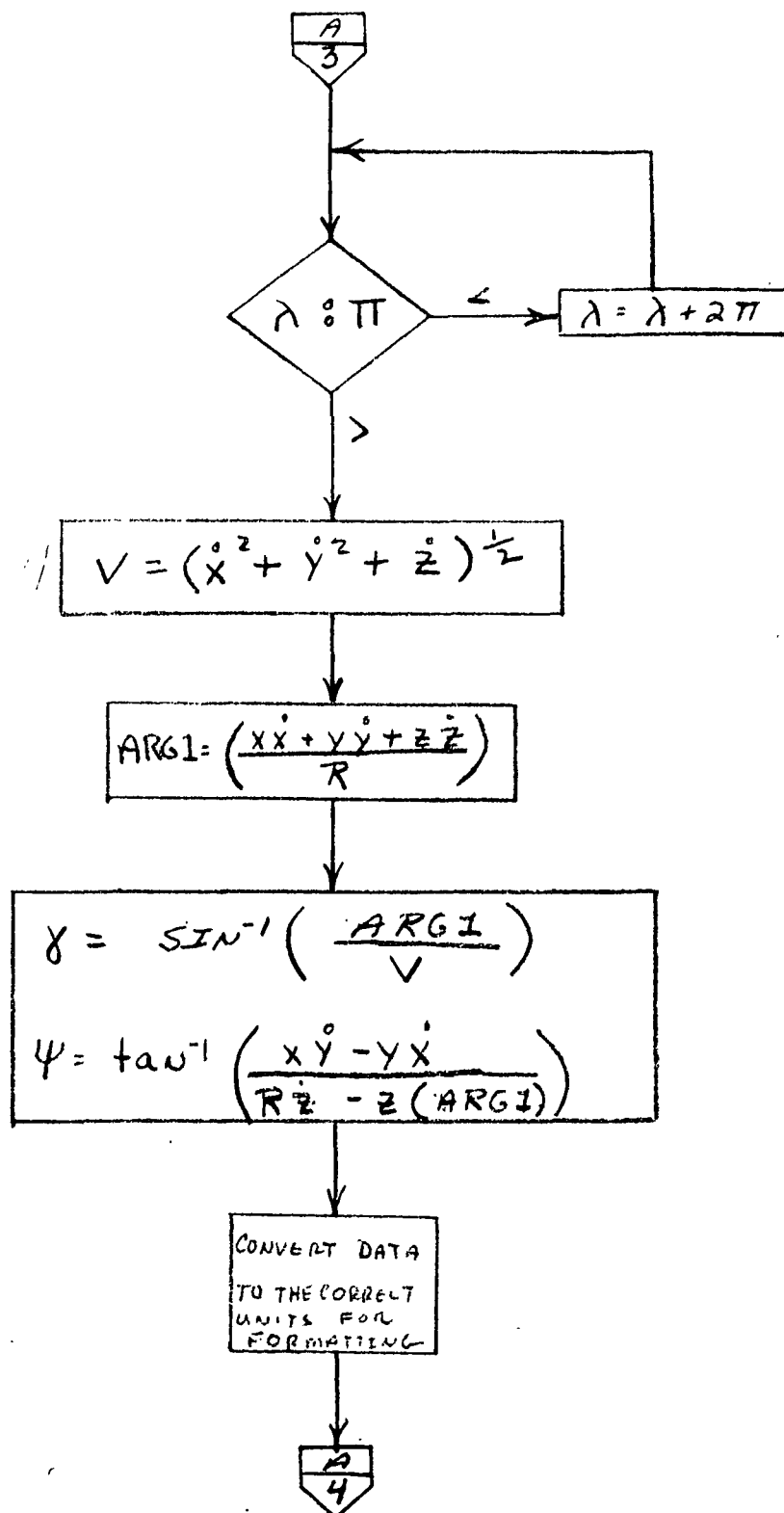


Figure PMXTAR-1. Flowchart

(Sheet 3 of 4)

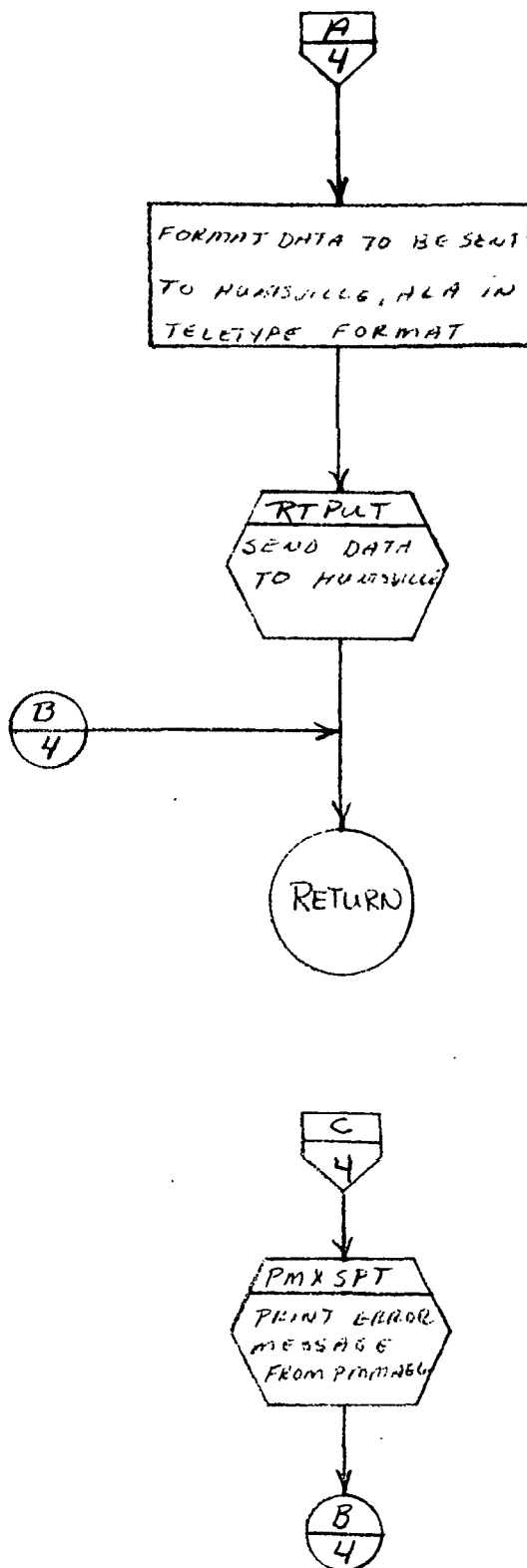


Figure PMXTAR-1. Flowchart

(Sheet 4 of 4)

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: Mission System - AS-200 Mission Planning

Section APP. PCHAPE

DATE 11/22/66 (207)

REV

PAGE 1 of 4

APP. PCHAPE

- a. Symbol — PCHAPE
- b. Program Name — Apogee And Perigee Radius Magnitude
- c. Programming Group — Mission Planning
- d. Programmer(s) — R. J. Barrett
- e. Language — Assembler
- f. Data Tables Used — None
- g. MED's Used — None
- h. Writeup Completion Date — 11/1/66
- i. Parameter List — Table PCHAPE-1.
- j. Flowchart — Figure PCHAPE-1.

FUNCTION

When given the radius and argument of latitude of three points spaced at 15-minute intervals, PCHAPE computes the apogee and perigee radius magnitudes of the trajectory.

USAGE

- a. Calling Sequence — CALL PCHAPE (DATA)
- b. Inputs — DATA - Twenty double words (see parameter list).
- c. Outputs and Exit — Radius at apogee and perigee (see parameter list)
- d. Other Programs Used — DSIN, DCOS, DATAN2
- e. Data Restrictions and Limits — None
- f. Error — None

METHOD

The general equation for harmonic oscillatory motion about a point is:

$$A = X \sin(\omega_0 t + B) \quad (1)$$

where X is the amplitude, $\omega_0 t$ is the phase angle, B is the initial phase and A is the instantaneous amplitude at time t.

Considering the elliptic motion of a satellite to be harmonic oscillatory motion about $\bar{R}$, where $\bar{R}$ is the mean radius, the radius at any point is defined by the equation:

$$R = \bar{R} + X_R \sin(U - U_0)$$

Approval	Approval
<i>R. J. Barrett</i>	<i>J. Stein</i>

where U_o is the argument of latitude for $\bar{R}$, X_R is the amplitude of the harmonic and U is the argument of latitude for R .

Three points (R_1, U_1) , (R_2, U_2) and (R_3, U_3) are needed to solve for $\bar{R}$, X_R and U_o . (These three points should be at 15-minute intervals to provide random points on the orbit).

Upon solving three simultaneous equations, the unknowns may be expressed as follows:

$$U_o = \tan^{-1} \frac{\sin U_1 - \sin U_2 - \alpha (\sin U_1 - \sin U_3)}{\alpha (\cos U_3 - \cos U_1) - (\cos U_2 - \cos U_1)}, \text{ where}$$

$$\alpha = \frac{R_1 - R_2}{R_1 - R_3}$$

$$X_R = (R_1 - R_3) / \left(\sin(U_1 - U_o) - \sin(U_3 - U_o) \right)$$

$$R = (R_1 + R_3) - X_R \left(\sin(U_1 - U_o) + \sin(U_3 - U_o) \right) / 2$$

Therefore, since $\bar{R}$ is the mean radius and X_R is the amplitude, the apogee and perigee radii may be defined by the following equations:

$$R_{\text{apogee}} = \bar{R} + |X_R|$$

$$R_{\text{perigee}} = \bar{R} - |X_R|$$

Note: X_R may be in the plus or minus amplitude direction hence $|X_R|$

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: Mission System - AS-200 Mission Planning

Section APP. PCHAPE

DATE 11/22/66 (207)

REV

PAGE 3 of 4

Table PCHAPE-1. Parameter List

QUEID —

QNAME —

From —

To-Task —

Purpose —

EP —

Byte No.	Type*	Units	Value / Description / Comments
1-8	D	CUL	R_1 radius at point 1 - input
9-16	D	CUL	R_2 radius at point 2 - input
17-24	D	CUL	R_3 radius at point 3 - input
25-32	D	radians	U_1 argument of latitude at point 1 - input
33-40	D	radians	U_2 argument of latitude at point 2 - input
41-48	D	radians	U_3 argument of latitude at point 3 - input
49-56	D	CUL	RAP - radius at apogee - output
57-64	D	CUL	RPE - radius at perigee
65-136	F		Save area for PCHAPE
137-160	D		Temporary Storage

*Type

H — Fixed-point half word

F — Fixed-point full word

E — Single-precision (floating-point)

D — Double-precision (floating-point)

X — Hexadecimal

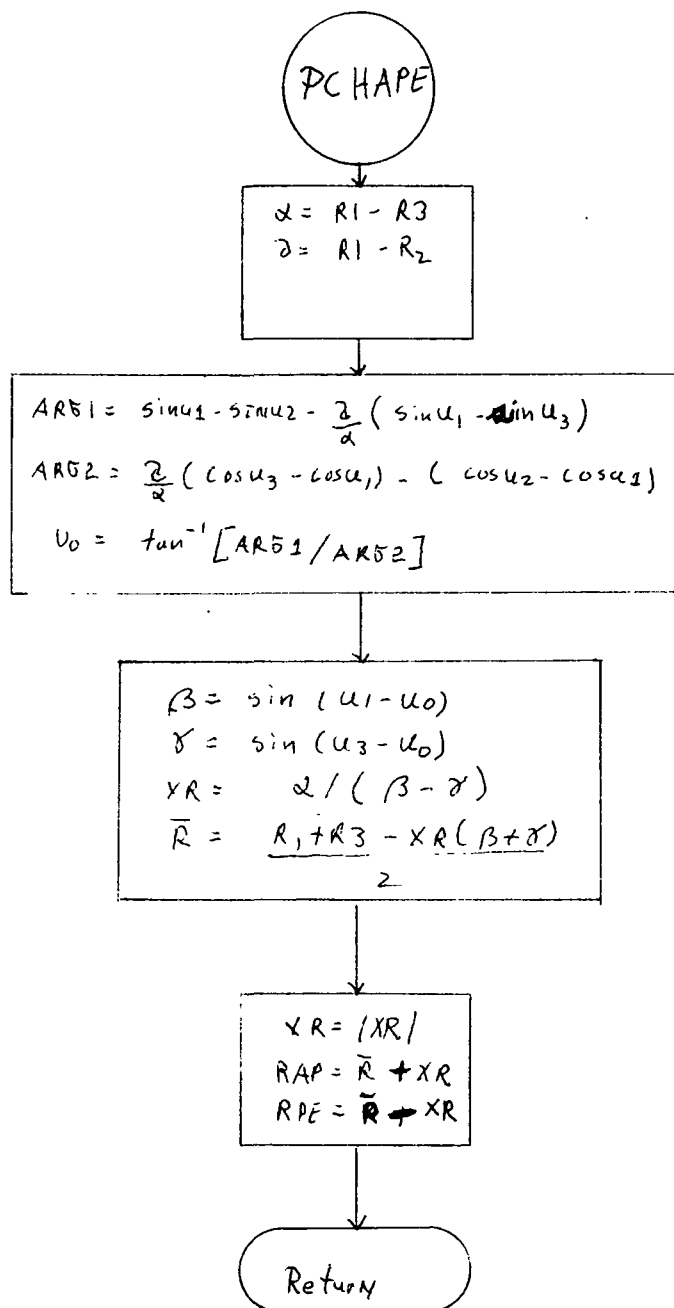


Figure PCHAPE-1. Flowchart

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

APP. PIPASVEL
DATE 11/22/66 (207)
REV
PAGE 1 of 3

APP. PIPASVEL

- a. Symbol — PIPASVEL
- b. Program Name — Position or Velocity Computation Subroutine
- c. Programming Group — Mission Planning
- d. Programmer (s) — G. Deaton
- e. Language — FORTRAN IV
- f. Data Tables Used — None
- g. MED's Used — None
- h. Writeup Completion Date — 10/31/66
- i. Flowchart — Figure PIPASVEL - 1.

FUNCTION

PIPASVEL. computes an inertial position vector or horizontal and radial velocity components based on input Keplerian osculating elements.

USAGE

- a. Calling Sequence — CALL POSVEL (A, B, C, D, E, SQRMU)
DIMENSION E (3). All are double words.
- b. Inputs —
 - 1. E (1) - indicator to compute position or velocity
E (1) = 0, compute position vector.
E (1) $\neq$ 0, compute velocities.
 - 2. SQRMU $-\sqrt{\mu}$
 - 3. If E (1) = 0, (A, B, C, D) = (i, h, u, R).
If E (1) $\neq$ 0, (A, B, C, D) = (a, e, f, R).
 - 4. i - inclination in radians.
 - 5. h - longitude of ascending node in radians.
 - 6. U - argument of latitude in radians.
 - 7. R - position magnitude in Earth radii (e. r.)
 - 8. a - semimajor axis in e. r.
 - 9. e - eccentricity
 - 10. f - true anomaly in radians.
- c. Outputs and Exit — E contains position vector components or velocity components (E_1 and E_2 only) according to input option.
Exit is by RETURN statement.
- d. Other Programs Used — Standard library subroutines
- e. Data Restrictions and Limits — None
- f. Error — None

Approval	Approval
<i>George Deaton</i>	<i>[Signature]</i>

IBM RTCC APOLLO PROGRAMMING SYSTEMS

BOOK: MISSION SYSTEMS - AS-200 Mission Planning

APP. PIPASVEL
DATE 11/22/66 (207
REV
PAGE 2 of 3METHOD

The position vector components are computed using the following formulae:

$$x = R (\cos U \cos h - \sin U \cos i \sin h)$$

$$y = R (\cos U \sin h + \sin U \cos i \cos h)$$

$$z = R \sin U \sin i$$

The horizontal and radial velocity components are, respectively:

$$V_x = V \cos \gamma$$

$$V_y = V \sin \gamma$$

where the flight-path angle $\gamma = \tan^{-1} \frac{e \sin f}{1 + e \cos f}$

and the velocity magnitude $V = \sqrt{\mu \left[\frac{2}{R} - \frac{1}{a} \right]^{1/2}}$

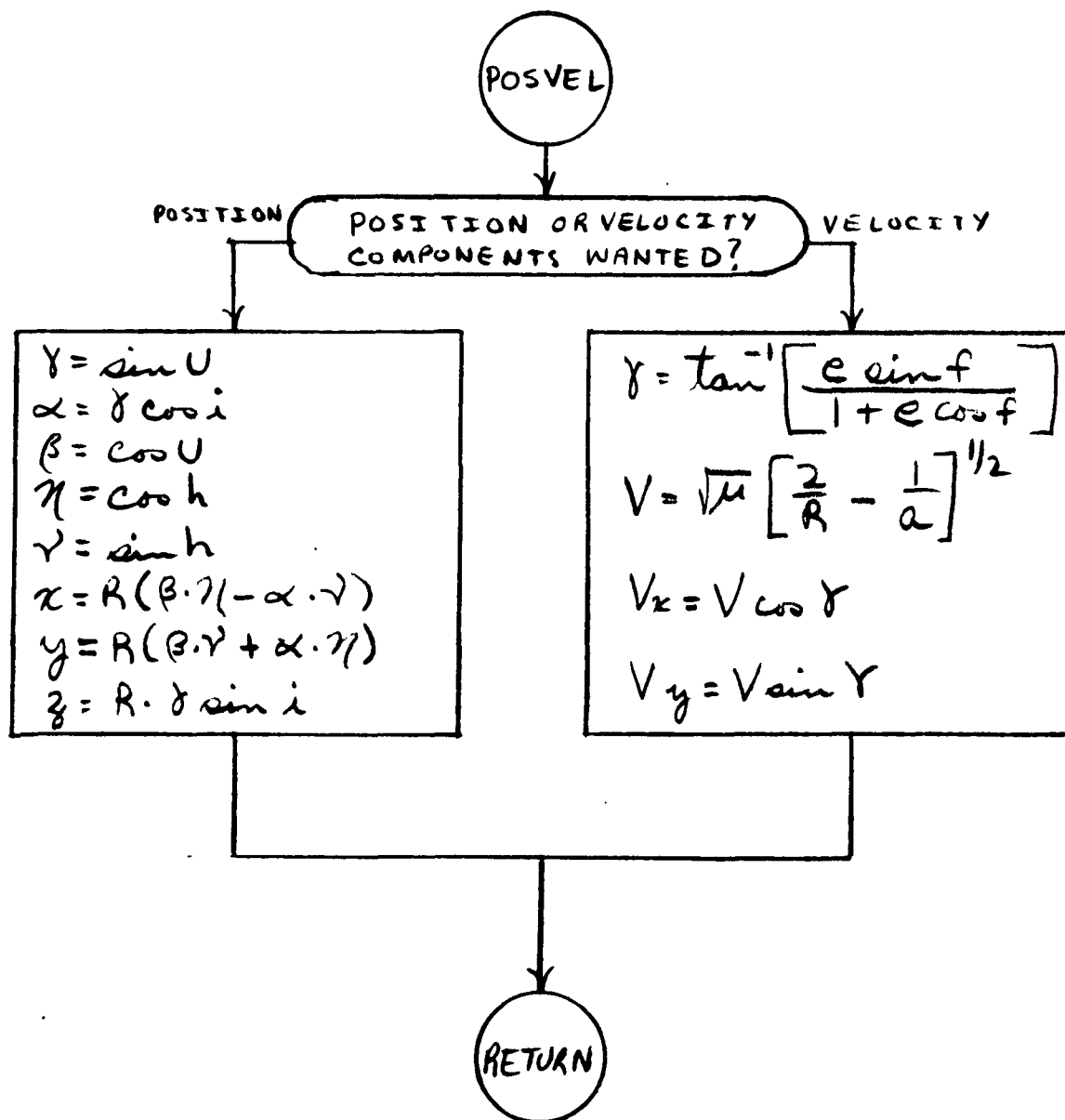


Figure PIPASVEL - 1.