79-FM-10
Rev. 2

JSC-14713
SEP 16 1980

Shuttle OFT Level C Navigation Requirements

Onorbit

Supersedes T80-10530

(NASA-TM-8221) SHUTTLE OFT LEVEL C
NAVIGATION REQUIREMENTS (NASA) 682 p
HC A99/MF A01
CSCL 17G
N80-33457
Unclas
G3/16 33863

Mission Planning and Analysis Division

September 1980

National Aeronautics and
Space Administration

Lyndon B. Johnson Space Center
Houston, Texas

79FM10

79-FM-10
Rev. 2

JSC-14713

SHUTTLE PROGRAM

SHUTTLE OFT
LEVEL C

NAVIGATION REQUIREMENTS
ONORBIT

By Mathematical Physics Branch, JSC;
and McDonnell Douglas Technical Services Co.

Approved: ______________________
Emil R. Schiesser, Chief
Mathematical Physics Branch

Approved: ______________________
Ronald L. Berry, Chief
Mission Planning and Analysis Division

Mission Planning and Analysis Division

National Aeronautics and Space Administration

Lyndon B. Johnson Space Center

Houston, Texas

September 1980

CONTENTS

TABLES

FIGURES

79FM10

SHUTTLE OFT LEVEL C
NAVIGATION REQUIREMENTS

ONORBIT

1.0 INTRODUCTION

1.1 PURPOSE

This document provides Level C detailed requirements for the orbital operations computer loads, OPS 2 and OPS 8. These requirements represent the total Onorbit/Rendezvous Navigation baseline requirements for the following principal functions.

A. Onorbit/Rendezvous Navigation Sequencer

B. Onorbit/Rendezvous UPP Sequencer

C. Onorbit/Rendezvous Navigation

D. Onorbit Prediction

E. Onorbit User Parameter Processing

F. Landing Site Update

The Onorbit/Rendezvous Navigation Baseline was last issued on May 1, 1979 as FSSR Version C; i.e., as SD 76-SH-0006C. That publication absorbed Version B (published November, 1977) and also the January 19, 1979 and May 1, 1979 block updates.

This publication reflects Version C of the Onorbit/Rendezvous Navigation FSSR plus the following page change numbers (PCN's) written against Version C:

A. PCN 1 - an August 15, 1979 update.

B. PCN 2 - a January 15, 1980 update.

C. PCN 3 - a May 15, 1980 update.

The following is a list of CR's that were incorporated into this document with Version C, PCN 1, PCN 2, and PCN 3.

CR 2599B
CR 12164
CR 19167B
CR 12266A
CR 12285
CR 12286
CR 12326
CR 12661
CR 12374
CR 12419
CR 12442A
CR 12456A
CR 12478
CR 12545
CR 12623
CR 12803
CR 19373B
CR 19445A
CR 19446A
CR 19472A
CR 19571A
CR 19655A
CR 19361A

CR 12681
CR 12765A
CR 12817
CR 12818A
CR 12819A
CR 12823
CR 12825B
CR 12676
CR 12812A
CR 12813A
CR 12820
CR 12821
CR 12822
CR 12214A
CR 12019
CR 29188C
CR 19810B
CR 19874A
CR 19873A
CR 19785B
CR 19900
CR 29300
CR 12600
CR 19452B
CR 12024B
CR 12411B
CR 19484A
CR 19575
CR 19246B
CR 12809A
CR 29500

1.2 ORGANIZATION AND STRUCTURE

The organization of the November 1977 baseline was extensively revised with its Version C in order to structure the text closely to the suggested implementation given in the flowcharts of appendices B, C, D, E, and F. In order to accomplish this task the following guidelines were developed for the organization and text structure of this document.

Guideline 1. - Each flowchart in the Appendices B, C, D, E, and F will be classified either as a subroutine, function, or code. The subroutine or function classification will be given for the following reasons:

(1) to avoid duplication of code,
and
(2) to clarify the function of the flowchart with text and Input/Output (I/O) tables.

Guideline 2. - The requirements for each subroutine and function will be described in one and only one text section of chapter 4.

A flowchart labeled code is to be regarded as in-line code and will be explained in the text section belonging to the subroutine in which the code is located. If a subroutine calls another subroutine or executes code, the CALL or EXECUTE statement will appear in the associated text. Text section descriptions will closely follow the associated flowchart.

Guideline 3. - Those text sections describing subroutines and functions will have Input/Output (I/O) tables.

The determination of Input and Output variables to a given subroutine can be quite subtle so the following simplistic definitions are proposed.

Input Variables

(1) Those variables needed to be input to the routine for the proper execution of the subroutine or function, e.g. variables used in computations, variables used in logical tests both implicit and explicit, etc.

(2) Variables in the INLIST to the subroutine.

Output Variables

(1) Those computed variables needed by other routines.

(2) Variables in the OUTLIST of the subroutine.

(3) Variables computed for the DOWNLIST.

Guideline 4. - Each I/O table will have the format given in Figure 1.2-1. The variables in the INLIST and OUTLIST as arguments to the subroutine have only local names so a special arrangement should be made for these variables by listing them under a special section in the Input/Output (I/O) tables with a slightly different format.

INLIST/OUTLIST*		INPUT SOURCE	OUTPUT DESTINATION
INTERNAL NAME	EXTERNAL NAME		
R	R_FILT	Subroutine A	
	R_TV	Subroutine B	
H	ALT		Subroutine A Subroutine B
VARIABLE NAME			
V_FILT		Subroutine A	Subroutine B
R_TV		Subroutine C	

Figure 1.2-1.- I/O table format.

*NOTE: If the subroutine has no INLIST or OUTLIST then the INLIST/OUTLIST portion of the I/O table will be omitted.

Input variables may be obtained by looking at the input source column and output variables may be obtained by looking at the output destination column.

Guideline 5. - Text sections describing Principal Functions will also have Input/Output Tables.

The Principal Function (PF) I/O will show Input/Output flow between principal functions. The Principal Function I/O variable definitions will be consistent with those used to define I/O variables for subroutines. The Principal Function I/O tables will have the format given by Figure 1.2-2. Local destination refers to the local subroutine needing the PF input variable. Local source refers to the local subroutine outputting the PF output variable.

VARIABLE NAME	PRINCIPAL FUNCTION SOURCE	LOCAL DESTINATION	PRINCIPAL FUNCTION DESTINATION	LOCAL SOURCE

Figure 1.2-2.- Principal function I/O table format.

Since one of the more important interfaces between Principal Functions is involved with the "snapping" of data, it is proposed that this type of interface be indicated in a special way by the Principal Function I/O tables. Variables involved in the IMU and Attitude Data Snap (section 4.2.2.1) or the Rendezvous Sensor Data Snap (section 4.2.2.2) will be given the variable name used by the appropriate SOP when that SOP supplies the variable for the Data Snap. The local destination will be labeled DATA SNAP and the reader will be referred to the text section explaining the Data Snap. This section will have tables showing the correspondence between the external SOP variable names and the variable names used by the Principal Function for the "snapped" variables.

Guideline 6. - In the November 1977 baseline document there was quite a bit of redundancy in the information supplied by various tables. For example, precision requirements were frequently given whenever a variable was listed in an I/O table. While such a practice can be defended on the grounds of clarity and convenience, it can lead to inconsistencies between tables. Subsequent versions of the FSSR have employed central tables which collate tabular information in one and only one place. It is intended that the variable list in Appendix A as well as the variable lists of Appendices C, D, E and F now include all of the following information.

1. Variable Name - The HAL variable name used in the flowcharts.

2. Precision and Type - This column will contain the following symbols:

   DF: double precision floating point scalar

   DF(n): double precision floating point n-dimensional array

   DF(n,m): double precision floating point n by m matrix

   SF: single precision floating point scalar

   SF(n): single precision floating point n-dimensional array

SF(n,m): single precision floating point n by m matrix

I: integer (unless otherwise specified all integer quantities are assumed to be single precision)

Bit: variable having only the values 0 or 1

Char: character string

3. Initialization category - This column will categorize the OPS-2 navigation parameters for initialization purposes.

   Mission Dependent (I-LOAD)

   Design Dependent

   Level A constants

   Hard Codeable

   OPS Transition parameters

   Other required initial values

4. COMPOOL or Local - This column will specify whether or not the variable is local or is required to be in COMPOOL.

5. Description - This column will provide a definition of the parameter.

6. Initial Value - This column will supply the numerical value for the Design Dependent and Hard Codeable parameters as well as other required initial values.

7. M/S ID - This column provides the M/S ID numbers for the variables in the PF I/O Tables.

8. Uplink/Downlist - This column will state which variables are uplinked or are to be downlisted.

Guideline 7. - Information that can be found in the Central Tables will not be duplicated in any other table unless necessary.

Corollary to guideline 7 is that there will be no collated Mission Dependent Parameter List (I-LOAD) and no collated Downlist. Rather, these variable lists can be gleaned from the information contained in the Central Tables in the Appendices.

1.3 THE CHANGE LOG

The changes made to the May 1979 baseline through PCN 2 to Version C of this document are tabulated in the following change log (table 1.3).

TABLE 1.3.- ONORBIT/RENDEZVOUS CHANGE LOG

Section	Section title	Description of change
1.1	Purpose	CR's added to list of CR's incorporated into this document.
1.3	Change Log	Change log for Nov. 1977 baseline document deleted. New change log begun for the May 1979 baseline.
3.3.4.1	Requirements Overview	Added description for new requirements on predictor.
3.3.4.2.1	Onorbit/Rendezvous Assumptions	Added and modified descriptions of nav's usage of o predictor o crew-controlled flags o state vector propagation scheme
3.3.4.2.2	Rendezvous Only Assumptions	Modified description of the covariance propagation rate.
4.1	Onorbit/Rendezvous Navigation Sequencer Principal Function	Sequencer task completely redesigned to take advantage of change to schedule NAV at only one state vector propagation rate -- DT_NAV_STATE_PROP.
		Interface with REL NAV display for the filter rate switch deleted.
		USE_IMU_DATA flag deleted as output to NAV and ORB_UPP.
		DOING_REND_NAV flag added as output to NAV and ORB_UPP.
		An IMU_THRESHOLD -- DA_THRESHOLD -- deleted as output to NAV.
		Significant changes made to principal function I/O table.
4.1.1	Sequencing Operations and Major Mode Transitions	Logic redesigned to account for NAV needing to be scheduled at only 1 state vector propagation rate.
		Logic deleted for setting USE_IMU_DATA, DA_THRESHOLD, and N_CYCLE.

TABLE 1.3.- ONORBIT/RENDEZVOUS CHANGE LOG

Section	Section title	Description of change
		Logic deleted for responding to DO_FLTR_SLOW_RATE switch and setting the DOING_FLTR_SLOW_RATE flag.
4.1.1.1	REND_NAV_EXIT	Equations added to turn off rendezvous flags.
4.1.1.2	NAV_EXIT	New section -- added as part of the sequencer redesign effort.
4.1.2	OPS_2_OR_8_INITIALIZE	Changed predictor step size for this task.
		Deleted setting powered flight navigation flags and the USE_IMU_DATA flag.
4.1.2.1	Orbiter State Vector Reset	Changes made to I/O table.
4.1.2.2	Rendezvous Navigation Initialization	Logic modified to delete directly calling the predictor and to schedule instead the state vector prediction task.
		Equations added for turning on two rendezvous navigation flags.
		Call to COV_LAST_RESET deleted (the call was moved to REND_COV_INIT).
		Changes made to I/O tables.
4.1.2.2.1	Covariance Matrix Initialization	Added call to COV_LAST_RESET.
4.1.2.2.1.3	Covariance Matrix Parameters Reset	New section number for existing module.
		Changes made to I/O tables.
4.1.2.2.2	Target State Vector Reset	Changes made to I/O tables.
4.1.2.2.3	Covariance Matrix Parameters Reset	Section number deleted. Module moved to section 4.1.2.2.1.3.

TABLE 1.3.- ONORBIT/RENDEZVOUS CHANGE LOG

Section	Section title	Description of change
4.2	Onorbit Rendezvous Navigation Principal Function	Logic added to compute value for N_CYCLE rather than depending upon obtaining the value from the sequencer. Changed names for the star tracker angles output by the S.T. SOP. Change made to interrogate the DOING_REND_NAV flag from the sequencer to determine when to exercise rendezvous navigation. Interface established with IMU_ALIGN_DISPLAY to access crew-input IMU threshold. Interface deleted with sequencer for obtaining an IMU threshold value. Interface deleted with sequencer and ORB_UPP for the USE_IMU_DATA flag. Internal logic modified to always set this flag. Significant changes made to principal function I/O tables.
4.2.1	Navigation Control	Sequence changed to execute state and covariance update functions prior to the state vector propagation function. Change made to key on the DOING_REND_NAV flag to determine when to exercise the rendezvous navigation logic. Logic added to compute N_CYCLE based on the crew-set DO_FLTR_SLOW_RATE switch. Changes made to I/O tables.
4.2.2.2	Rendezvous Sensor Data Snap	Names changed for the S.T. angles output by the S.T. SOP.
4.2.3	Position and Velocity State Propagation	Section moved to section number 4.2.4.

TABLE 1.3.- ONORBIT/RENDEZVOUS CHANGE LOG

Section	Section title	Description of change
		Change made to delete the call to the free flight propagator -- ONORBIT_PRECISE_PROP -- and call the SUPER_G propagator for both Orbiter and target state propagation for both free and powered flight.
		Logic added to make use of the IMU_NAV_ACCEL_THRESH set by the crew.
		Logic modified to set the USE_IMU_DATA flag on each navigation cycle.
		Logic modified to key on the DOING_REND_NAV flag to determine when target vehicle propagation is required.
		Logic modified to use the powered flight potential model only when the computed Orbiter acceleration exceeds MEAS_THRESHOLD.
		Changes made to I/O table.
4.2.3.1	Precision Propagation	Section deleted.
4.2.3.1.1	Integration of the Equations of Motion	Section moved to section number 4.3.1.
		Changes made to I/O table.
4.2.3.1.1	STATE_VECTOR_PREDICT_TASK	New section which has assumed the old section number of RK_GILL.
4.2.3.1.2	Equations of Motion	Section moved to section number 4.3.2.
		Changes made to I/O table.
4.2.3.1.3	Conic Solution	Section moved to section number 4.2.5.2.1.
4.2.3.1.4	Acceleration Models	Section moved to section number 4.2.4.1.1.
		Changes made to I/O table.

TABLE 1.3.- ONORBIT/RENDEZVOUS CHANGE LOG

Section	Section title	Description of change
4.2.3.1.4.1	Altitude Above the Reference Ellipsoid	Section moved to section number 4.2.4.1.1.1.
4.2.3.1.4.2	Earth Relative Velocity Computation	Section moved to section number 4.2.4.1.1.2.
4.2.3.2	Powered Flight Propagation	Section moved to section number 4.2.4.1. Section title changed to Position and Velocity Propagation. Logic made into a subroutine instead of a code since it is executed for both the Orbiter and target vehicles. Changes made to I/O table.
4.2.4	Unmodeled Acceleration State and Covariance Matrix Propagation	Section moved to section number 4.2.5. Changes made to I/O table.
4.2.4.1	Unmodeled Acceleration Bias and Covariance Initialization	Section moved to section number 4.2.5.1.
4.2.4.2	Position-Velocity Submatrix of the State Transition Matrix	Section moved to section number 4.2.5.2.
4.2.5	State and Covariance Matrix Updates	Section moved to section number 4.2.3
4.2.5.1	Auto Inflight Updates	Section moved to section number 4.2.3.1. Interface established to schedule the STATE_VECTOR_PREDICT_TASK as needed rather than making an in-line call to the predictor.

TABLE 1.3.- ONORBIT/RENDEZVOUS CHANGE LOG

Section	Section title	Description of change
		Significant changes made to interface with the state vector predict task.
		Local flag, SV_UPDATE, created for use in determining when to call the covariance matrix initialization module.
		Changes made to the I/O table.
4.2.5.2	REL_NAV Display Updates	Section moved to section number 4.2.3.2.
		Changes made to the I/O table.
4.2.5.2.1	Conic Solution	New section.
4.2.8.1.1	Measurement Interpolation	Changes made to the I/O table.
4.2.8.1.2	Kalman Filter Updates	Changes made to the I/O table.
4.2.8.3.1	Angle Measurements	Equations changed to account for S.T. SOP angles being output in radians, not degrees.
4.3	Onorbit Precision State Prediction Principal Function	Added 2 new sections — RK_GILL and PINES.
		Iterative prediction process modified to iterate on time, not number of steps.
		Changes made to principal function I/O table and to ONORBIT_PREDICT I/O table.
4.3.1	Integration of the Equations of Motion	New section number for an existing module.
4.3.2	Equations of Motion	New section number for an existing module.
4.5	User Parameter Processing Principal Function	Change made to interrogate the DOING_REND_NAV flag (set by the sequencer) to determine when to exercise rendezvous functions.

TABLE 1.3.- ONORBIT/RENDEZVOUS CHANGE LOG

Section	Section title	Description of change
4.5.1	User Parameter State Propagation	Interface established with the IMU_ALIGN_DISPLAY to access crew-input IMU threshold.
		Logic added to enable the UPP to make its own acceleration threshold test to determine when to use IMU data to propagate the Orbiter position and velocity.
		The PWRD_FLT_NAV flag was added as an input for determining when to interrogate the IMU data.
		Changes made to the principal function I/O table.
		Test on the PWRD_FLT_NAV flag added to determine whether or not to look at the IMU data.
		Orbiter acceleration threshold test added to determine whether or not to use the IMU data in propagating the Orbiter position and velocity.
		Rendezvous phase test modified to interrogate the DOING_REND_NAV flag rather than the REND_NAV_FLAG.
		Changes made to the I/O table.
4.5.1.1	Integration	Calling argument changed in calls to ACCEL_ONORBIT.
		Changes made to the I/O table.
4.5.2	Onorbit User Parameter Calculations	Rendezvous phase test modified to interrogate the DOING_REND_NAV flag rather than the REND_NAV_FLAG. Appropriate change made to I/O table.
4.6	Landing Site Update Principal Function	Updated text and tables per CR 19373B.
	Appendixes	Changes made corresponding to the changes in the text.

TABLE 4.3-3.- PREDICTOR SUGGESTED FLAG SETTINGS

Vehicle	Prediction type	GMDP*	GMOP*	DMP	VMP	ATMP	PRED_STEP	Comments
Orbiter	Precision	4	4	1	0	1	User selects	Full fourth degree potential model. Drag with constant drag coefficient, area.
Orbiter	Rapid precision	2	0	1	0	1	User selects	J_2 only potential model with constant drag coefficient, area.
Orbiter	Rapid two-body	0	0	0	0	0	0	Single-step two-body F and G series solution.
Target	Precision	4	4	1	0	2	User selects	Full fourth degree potential model drag with constant area, drag coefficient.
Target	Rapid precision	2	0	1	0	2	User selects	J_2 only potential model with constant drag coefficient, area.
Target	Rapid two-body	0	0	0	0	0	0	single-step two-body F and G series solution.

*When prediction is being performed for both vehicles (Orbiter and target) over a similar trajectory, the same degree and order potential model should be used for each prediction so that potential model errors will be avoided.

2.0 APPLICABLE DOCUMENTS

Document no.	Title
SS-P-0002-140	Computer Program Development Specification, Volume I, Book 4 Downlist/Uplink Software Requirements
SS-P-0002-170	OFT System Level Requirements, Software
SS-P-0002-510	OFT Functional Level Requirements, GN&C
SS-P-0002-520	OFT Detail Level Requirements, GN&C
SS-P-0002-530	OFT Functional Level Requirements, Systems Management
SS-P-0002-540	OFT Detail Level Requirements, Systems Management
SS-P-0002-550	OFT Functional Level Requirements, Vehicle Utility-02
SS-P-0002-560	OFT Detail Level Requirements, Vehicle Utility-02
SS-P-0002-570	OFT Functional Level Requirements, Payloads
SS-P-0002-580	OFT Detail Level Requirements, Payloads
SD76-SH-0020	OFT Flight Software System Requirements, Displays and Controls
SS-P-0002-195	OFT I-LOADS Flight Software System Requirements
SS-P-0002-110	OFT System Level Requirements (Level A) Constants

3.0 OVERVIEW

3.1 OPERATIONAL NAVIGATION PROGRAM

For the Orbiter flight tests, the requirements for each Operational Navigation Program (ONP) are specified at three levels: system level (A), functional level (B), and detailed level (C). In addition, the Level B and C requirements are specified in separate documents for guidance, navigation, and control (GN&C); system management (SM); vehicle utility (VU) and payloads (PL). This document is the Onorbit and Rendezvous navigation part of the GN&C Level C OFT ONP requirements.

3.2 GN&C SOFTWARE MAJOR FUNCTIONS

The Orbiter general-purpose computer (GPC) provides the following GN&C major functions:

- Guidance (GUID)
- Navigation (NAV)
- Flight Control (FC)
- Redundancy management/moding, sequencing, and control (RM/MSC)
- Subsystem operation programs (SOP)
- Displays and controls (D&C)
- Other

This document specifies the software functional requirements for the GN&C major function, onorbit and rendezvous navigation.

3.3 NAVIGATION SYSTEM OVERVIEW

The basic function of the navigation system is to provide an accurate estimate of the Orbiter and target state using orbital dynamics (for coasting flight), IMU data (for powered flight) and NAVAID data which may be used with or without IMU data. This document provides detailed navigation software requirements for the onorbit operational sequences:

- Onorbit Operational Sequence (OPS 2)
- Flight Control System Checkout Operational Sequence (OPS 8)

3.3.1 Navigation Functions

In general, the navigation software requirements can be divided into the following major functions:

Navigation Control. Performs the initialization of navigation function parameters and sets up the sequencing of functions to accomplish navigation requirements.

Measurement Scheduler. Selects the appropriate sensor measurements in accordance with selection criteria.

Data Handler. Prepares data for sensor measurement processing.

Navigation Reconfiguration. Initializes state vector and covariance matrix for sensor measurement processing, for manual updates, for ground updates or for changes in sensor type.

State and Covariance Propagation. Propagates the state and the covariance matrix for sensor measurement processing. Also propagates state for user parameter calculations.

State and Covariance Update. Determines and performs the state and the covariance matrix updates.

User Parameter Processing. Computes state-related parameters for guidance and display and control, and provides high rate propagation of the state vector.

In this document, the details of the major functions are presented under the following Level B principal functions:

(1) Onorbit/Rendezvous Navigation Sequencer

(2) Onorbit/Rendezvous Navigation

(3) Onorbit User Parameter Processing Sequencer

(4) Onorbit User Parameter Processing

The details of each function are shown in Appendices B, C, D and F in the form of flowcharts of the modules comprising each function. The interconnection of the functions and component modules is shown by means of a 'block' diagram in Appendix G.

The navigation software requirements as presented make assumptions about how non-navigation functions will be performed. Completion of the software requirements will require adequate definition of the timing and data time-tagging mechanization and the IOP mechanization. Changes in the navigation software requirements may be necessary, depending on actual implementation of the non-navigation functions.

3.3.2 Navigation-Related Functions

The navigation-related function is the Landing Site Update principal function. This function provides the capability to reconfigure the dynamic parameters pertaining to the runway and TACAN sites, which are to carry over into OPS 3. The Landing Site Update function is itself a Level B principal function. Flowcharts for this function are given in Appendix E.

3.3.3 General Requirements

This section discusses software requirements in the categories of service, single-use, or multiple-use that are not uniquely related to the navigation function.

The general requirements include, but are not limited to the following: Coordinate transformations (this publication contains transformations required for navigation).

Onorbit Prediction - this function provides the capability to propagate a state vector forward or backward in time over possibly large time intervals when requested by a user. Onorbit prediction is a Level B principal function.

Flowcharts for the general requirements functions are contained in Appendix C.

3.3.4 Requirements and Assumptions Overview

The following two sections present an overview of the requirements and assumptions of the OFT Onorbit/Rendezvous GN&C navigation-related software.

3.3.4.1 Requirements Overview

The following statements are navigation requirements that must not be violated. To illustrate the requirements, references are made to the flowcharts in Appendix B.

1. During the time that a state vector is being read by non-navigation application software, that state vector or portions thereof will not be updated by the navigation software. The reset equations in the user parameter propagator shall be so protected.

2. Statements that snap data upon entry into navigation shall be made so that a timewise consistent set of data is obtained before the set of data is used.

3. The navigation sensor read functions must not be interrupted by other programs until they are completed.

4. Since the acceleration models are to be used by both a navigation processing principal function (onorbit or rendezvous navigation) and the predictor principal function, execution of the model (for a given cycle) shall not be interrupted (ACCEL_ONORBIT).

5. The specific display interface flags (PWRD_FLT_NAV, REND_NAV_FLAG, MEAS_ENABLE, ANGLES_AIF, RANGE_AIF, RDOT_AIF) shall not have their values changed by any source external to navigation during the execution of a navigation cycle.

6. The Onorbit/Rendezvous sequencer principal function shall have the capability to execute at a maximum rate of one cycle each 1.92 seconds.

7 During any navigation phase, execution of the Onorbit/Rendezvous Sequencer and the Onorbit/Rendezvous Navigation principal function shall take precedence over the execution of the Onorbit Precision State Prediction principal function. This requirement is needed to assure that both the sequencer and navigation principal function executions will not be delayed by execution of the predictor principal function. This requirement shall not apply to the initialization procedures which take place upon entering OPS 2 or OPS 8.

3.3.4.2 Assumptions Overview

The following sections present the assumptions that were used in the development of the Level C Onorbit/Rendezvous requirements. The first section consists of those assumptions made that affect onorbit requirements as well as assumptions that affect both onorbit and rendezvous requirements. Section 3.3.4.2.2 contains only rendezvous related assumptions.

3.3.4.2.1 Onorbit/rendezvous assumptions

1. Upon entering OPS 2 or OPS 8 the Orbiter position and velocity vectors, obtained from the previous OPS, must be predicted to current time. Since this prediction is essential to the navigation task, the sequencer principal function execution must be dependent on the successful prediction of the Orbiter state vectors. In order to minimize the delay due to the execution of the predictor principal function, the initial step size (PRED_STEP_OPS_INIT), is set to ensure that only one prediction step will be taken by the predictor.

2. No onboard external data are processed during the non-rendezvous portion of operational sequence (OPS) 2 (i.e., onorbit coast and onorbit powered flight). One-way Doppler tracking and data relay satellite system (TDRSS) measurement incorporation is not currently planned for the orbital flight test (OFT) program.

3. A six-dimensional state vector is maintained during nonrendezvous portions of OPS 2 and during OPS 8 (three position and three velocity).

4. Prestored values for a nominal body contact force (due to venting and/or uncoupled thrusting, etc.) and vehicle/payload area configuration are required for acceleration models.

5. The inertial measurement unit subsystem operating program (IMU SOF) provides an estimate of the total accumulated IMU velocity at the time of a data snap in the presence of commfaults.

6. OPS 2 and OPS 8 computational accuracy is assumed to be of AP101 double precision accuracy where mixed mode arithmetic is employed.

7. Incorporation of sensed velocity into the state propagation integrator is controlled by setting the PWRD_FLT_NAV flag in conjunction with an acceleration threshold. This flag is set by either moding, sequencing, and control (MSC), or by the crew via item entry on the REL_NAV display (in OPS 2) or the FCS_DIS_C/O display (OPS 8). MSC will activate the PWRD_FLT_NAV flag upon entrance into Major Mode 202 and deactivate it upon transition from Major Mode 202 to Major Mode 201. The crew can activate/deactivate the PWRD FLT_NAV flag in Major Mode 201 specifically to handle major manual translation maneuvers and auxiliary power unit (APU) venting (during OPS 8) using the REL_NAV display (OPS 2) or the FCS DIS C/O display (OPS 8).

8. Use of sensed velocity in the navigation state propagator is triggered by entrance into the onorbit or rendezvous powered flight navigation phase (Event 67 OPS 2, or via crew control on the REL_NAV or FCS_DIS_C/O display in Major Mode 201 or 801) and by testing accelerometer output versus prestored threshold levels. The threshold level, in micro g's, can be changed by crew input

on the IMU ALIGN DISPLAY. The crew inputted threshold level will be preserved during OPS 2 and OPS 8 transitions.

9. Backward and forward integration capability is provided for state prediction and propagation.

10. Current attitude is used for propagation.

11. The precision state prediction function has accuracy comparable to that of the precision state propagation function and has the option of being executed in a faster (but less accurate) conic mode.

12. Acceleration models include both attitude-dependent and constant ballistic coefficient drag, a venting/RCS body contact force, and Earth gravity effects.

13. Only one Orbiter and/or target state vector shall be maintained during all navigation phases in OPS 2 and OPS 8.

14. The acceleration due to lift force is assumed to be negligible in the atmospheric drag acceleration model.

15. An automatic in-flight update capability will be provided by which the ground can uplink either an Orbiter or a target state vector (M50) and associated time tag during any navigation phase (rendezvous or nonrendezvous). The following additional assumptions apply to this capability:

    a. The ground shall uplink one vehicle state (three position, three velocity, associated time tag, and OPCODE) at a time.

    b. The onboard software receiving these data (ground uplink high-rate special processing S/W processor) will set the DO_OV_UPLINK or DO_TV_UPLINK flag to ON to specify whether the uplinked data pertains to Orbiter or target, and set up one of the following two variable sets depending on the results of this test:

Orbiter uplink		target uplink
$\underline{R}$ _GND	or	$\underline{R}$ _TV_GND
$\underline{V}$ _GND		$\underline{V}$ _TV_GND
T_GND		T_TV_GND
DO_OV_UPLINK = ON		DO_TV_UPLINK = ON

    c. The navigation software has the capability of reinitializing the Orbiter and/or target state vectors (and associated covariance matrix during rendezvous) in a single navigation cycle.

d. If a target vector is uplinked during a nonrendezvous navigation phase, it is stored for eventual use in a rendezvous phase.

16. Propagation of Orbiter and target position and velocity vectors will be performed by use of the super-G integration scheme during all navigation phases.

17. If the sensor (including IMU) SOP's are not in the same general-purpose computer (GPC) as the navigation filter software, then:

    a. Data and time tag must be preserved as a pair.

    b. ICC transmission rate must be fast enough such that the data time tag and current time (in navigation GPC) difference shall not adversely affect navigation software performance.

    If the sensor SOP's and navigation filter reside in the same GPC, then:

    a. Data must be time tagged.

    b. Data must be the latest available at the time of the data snap.

18. The onorbit/rendezvous navigation sequencer will always snap IMU data and predict the last available state vector to current time when a memory transition has been performed:

    OPS 1 to 2
    OPS 2 to 8
    OPS 3 to 2
    OPS 8 to 2
    OPS 0 to 2

19. The onorbit/rendezvous navigation sequencer is capable of responding to the following crew-controlled functions on the REL_NAV and FCS_DIS_C/O displays:

    a. REL_NAV

        (1) Enable/disable onorbit and rendezvous navigation phases.

        (2) Set flags for powered flight and coasting flight during Major Mode 201 (refer to Assumption 6).

        (3) Allow measurement processing during rendezvous navigation, Major Mode 202.

    b. FCS_DIS_C/O display (OPS 8): Set positive feedback flags to display for indicating powered flight or coasting flight conditions.

3.3.4.2.2 Rendezvous-only assumptions

1. Entrance into the rendezvous navigation phase shall require target position and velocity vectors that have successfully been predicted to within a design dependent tolerance of current navigation time. A flag, DOING_REND_NAV, shall be set to indicate to the crew as well as other software that the rendezvous navigation phase is active.

2. The onboard navigation software will store premission data for only one target vehicle.

3. It is assumed that one of the vehicles (either target or Shuttle) will have an accurately known position and velocity vector throughout the rendezvous navigation phase. The onboard navigation will carry a flag (SHUTTLE_FILTER_FLAG) which will determine whether the Shuttle or target state vector will be updated by the Kalman filter. The present design assumes that the SHUTTLE_FILTER_FLAG will be a mission dependent (I-LOAD) parameter; however, it is anticipated that future requirements may dictate that this flag be changed via UPLINK or crew display. Accordingly, any implementation should have the flexibility to allow for such a change with minimum impact.

4. The state vector maintained for rendezvous navigation will consist of 13 components.

Component no.	Description
1-3	Vehicle position (Shuttle or target)
4-6	Vehicle velocity (Shuttle or target)
7-9	Unmodeled acceleration estimates
10-13	Rendezvous tracker bias estimates

5. A capability shall be available to make the unmodeled acceleration states consider parameters only. In this mode, the unmodeled acceleration states are not updated by Kalman processing but the statistics are carried in the covariance matrix to act as state noise. An I-LOAD parameter will determine if this capability is in operation or not and this parameter will not be changed throughout the mission.

6. There is no capability to uplink vents or thrusts for the target vehicle.

7. A 13 by 13 covariance matrix of Aries mean of 1950 position and velocity, unmodeled acceleration bias errors, and of (at most) four rendezvous tracker (instrument) biases is propagated during rendezvous coast and rendezvous powered flight navigation phases.

8. A valid or appropriate target vector shall always be I-LOADED or UPLINKED prior to entry into the rendezvous navigation phase.

9. Upon entering rendezvous navigation, the covariance matrix is initialized to values stored in certain memory locations. These values are initialized

through I-LOAD and the position-velocity submatrix may be redefined with a ground uplink at any time.

10. The capability to load a premission determined set of known sensor biases (determined by calibration) into the bias slots of the navigated state vector shall be provided.

11. Both target and Orbiter states shall be propagated but only one state will be updated by the Kalman filter.

12. The covariance propagation shall be done at a multiple (NCYCLE) of the state propagation rate to allow for slower measurement processing. The following subfunctions shall be done at the covariance propagation rate.

    a. Sensor measurement selection

    b. Measurement reconfiguration

    c. Covariance matrix propagation

    d. All measurement incorporation subfunctions

    e. Measurement processing statistics

13. All rendezvous tracker bias variances are propagated as exponentially correlated random variables in the error covariance matrix propagation.

14. The unmodeled acceleration bias states are propagated as exponentially correlated random variables.

15. The covariance matrix can be reinitialized in any of four ways:

    a. Execution of Orbiter to target state transfer on the REL_NAV display,

    b. Execution of target to Orbiter state transfer on the REL_NAV display,

    c. Execution of covariance matrix reinitialization on the REL_NAV display, or

    d. Auto inflight update of either the target or the Orbiter.

    In each of these cases the covariance matrix is set to the values contained in a specified set of memory locations. The position and velocity portions of the covariance matrix are computed using parameters that can be changed by ground uplink.

16. The following external data will be processed during the rendezvous coast navigation or during rendezvous powered flight when measurement processing is enabled and the acceleration level is below a predetermined threshold.

a. Rendezvous radar shaft angle, trunnion angle, range, and range rate

b. Star tracker horizontal and vertical angles

c. Crew optical alinement sight (COAS) horizontal and vertical angles.

17. External measurement angle data are selected and processed mutually exclusively on an instrument basis. The rendezvous radar range and range rate may be processed with COAS, star tracker, or rendezvous radar angles. The display interface processor (DIP) will ensure this by activating the navigation sensor selection ENABLE flag for only the most recently crew-selected instrument.

18. External measurement data processing is inhibited for display and state vector updates whenever IMU sensed delta velocities are in excess of a design dependent amount (MEAS_THRESHOLD).

19. During Major Mode 202, when the MEAS_ENABLE switch on the REL_NAV display is set to OFF, the measurement statistics will be computed for display (assuming the data are valid), but the measurements will not be incorporated into the state vector update.

20. If a sensor AUTO/INHIBIT/FORCE (AIF) flag is switched to FORCE by the crew, this FORCE will be acknowledged by navigation for one navigation cycle only. After one cycle the flag must be reset to FORCE in order to force data. After the AIF flag has been processed, a copy of the AIF flag is communicated to the REL_NAV display by navigation.

21. There shall be a bilevel edit criterion in the navigation filter. If the crew attempts to force data, the allowable residual ratio threshold shall be formulated as the larger of either the sum of the last residual ratio for the measurement type being processed and a design dependent amount (DELTA_RESID_RATIO) or the number 1. For angle data the last residual ratio is equal to the greater of the two angle residual ratios from the previous cycle.

22. There shall be a data valid flag for rendezvous radar range and a data valid flag for rendezvous radar range rate. There will be a single data valid flag for each of the following angle sets: Rendezvous radar angles, star tracker angles, and COAS angles.

23. COAS processing is formulated such that data which is stale by more than a design dependent amount (DELTAT_COAS_MAX) is considered invalid data and is not processed. Furthermore, the same data will not be incorporated twice by the filter.

24. The star tracker SOP shall supply a flag indicating that the star tracker is in the target tracking mode.

If the star tracker is not in the target tracking mode then the angle data are considered invalid and it is not processed by the filter.

25. The Rendezvous Radar SOP shall supply a flag indicating if the rendezvous radar is in the self test mode and, if it is, none of the rendezvous radar measurements (range, range rate, or angles) are processed by the filter.

26. Kalman filter statistics (residuals and residual ratios) will be calculated and displayed for each sensor type (range, range rate, angles) whenever rendezvous navigation is enabled and the sensor data are valid.

3.3.4.2.3 General requirements

1. Figure 3.3-1 presents the assumed verified OFT transitions for OPS 2 and OPS 8 and a summary of the events and major mode (MM) transitions that the Onorbit/Rendezvous Navigation Sequencer principal function must account for. The following list of assumptions applies to these verified transitions:

2. There are four navigation phases: onorbit coast, rendezvous coast, onorbit powered flight, and rendezvous powered flight.

3. OPS 2 can be entered from OPS 1 and can only begin with the onorbit coast navigation phase of the Onorbit/Rendezvous Navigation principal function.

4. A flag, REND_NAV_FLAG, shall be set (and reset) external to the onorbit/ rendezvous navigation sequencer and onorbit/rendezvous navigation principal function. It is assumed that the crew will control this flag setting via the REL_NAV display. MSC may also control the flag for automatic transitions.

5. Once the onorbit navigation phase or the rendezvous navigation phase has been activated (while in Major Mode 201), the transition from one to the other can be controlled via crew resetting of the REND_NAV_FLAG via the REL_NAV display. This option is also available during Major Mode 202.

6. Entry into OPS 2 from OPS 3 can only occur when onorbit navigation is used (i.e., cannot begin OPS 2 with rendezvous navigation when coming from OPS 3).

7. Whenever reentering OPS 2 from OPS 00, entry is assumed to be made into Major Mode 201 with onorbit navigation active.

8. Transition to OPS 3 from OPS 2 (Major Mode 201) can occur while either onorbit or rendezvous navigation is active.

9. Transitions into OPS 2 and OPS 8 can only begin with onorbit navigation active in the coast flight phase.

10. It is assumed that OPS 2 and OPS 8 are separate memory loads in which all navigation software for the latter is contained within the former. A memory transition shall take place to transfer those OPS 2 principal functions needed for OPS 8, namely

a. Onorbit/Rendezvous Navigation Sequencer

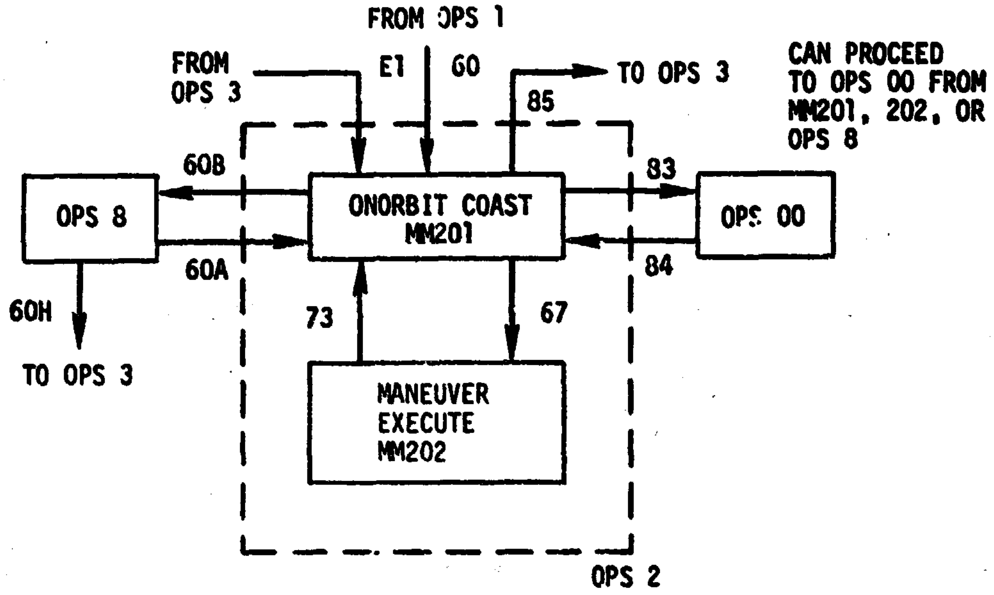
b. Onorbit/Rendezvous Navigation

c. Onorbit/Rendezvous User Parameter Processing Sequencer

d. Onorbit User Parameter Processing

e. Onorbit Predictor

The navigation sequencer will currently indicate a cancellation and rescheduling of ONORBIT_RENDEZVOUS_NAVIGATION during the transition from OPS 8 to OPS 2 or OPS 2 to OPS 8, because of this assumption.



EVENT SUMMARY

EVENT NO.	DESCRIPTION	CRITERIA	SOURCE
60	106 → 201	OPS 201 PRO	CREW
60A	OPS 8 → 201	OPS 201 PRO	CREW
60B	201 → OPS 8	OPS 801 PRO	CREW
67	201 → 202	PRO OR OPS 202 PRO	CREW
73	202 → 201	PRO OR OPS 201 PRO	CREW
84	OPS 00 → 201	OPS 201 PRO	CREW
85	201 → OPS 3	OPS 301 PRO	CREW
E1	OPS 3 → 201	OPS 201 PRO	CREW
60H	OPS 8 → OPS 3		CREW
83	OPS 2 → OPS 00		CREW/MSC

Figure 3.3-1.- Verified OFT transitions.

3.4 TRACEABILITY

Table 3.4 shows the traceability of the software segments in FSSR Level C with respect to the CPDS Level B.

TABLE 3.4.- SOFTWARE REQUIREMENT TRACEABILITY

CPDS Level B Section No.	Software Segment	FSSR Level C Section No.
4.148	Onorbit/Rendezvous Navigation Sequencer	4.1
4.198	Onorbit User Parameter Processing Sequencer	4.4
4.126	Onorbit/Rendezvous Navigation	4.2
4.246	Landing Site Update	4.6
4.22	Onorbit User Parameter Processing	4.5
4.224	Onorbit Predictor	4.3

3.5 OPERATIONAL SEQUENCES AND MAJOR MODES

Table 3.5 identifies the OPS 2 and OPS 8 operation sequence major modes as referenced in Section 4, Detailed Requirements, and Appendix B. Detailed descriptions may be found in the Level B CPDS.

TABLE 3.5.- OPERATION SEQUENCE MAJOR MODES

Major Mode No.	Onorbit/Rendezvous Phases
201	Onorbit powered flight Onorbit coast Rendezvous powered flight Rendezvous coast
202	Onorbit powered flight Rendezvous powered flight
OPS 8	Onorbit powered flight Onorbit coast

3.6 IMPLEMENTATION CONSTRAINTS

The navigation subsystem design is capable of recovering from the effects of many transient-type errors. Accordingly, the software design shall not preclude this capability by causing a program halt, permanent discontinuance of navigation processing, or loss of protected data as a result of program check or error interrupts.

Check or error interrupts shall be prevented by software safeguards, or standard fixup and error returns shall be provided which are appropriate in the navigation function as described below.

1. Floating Point Overflow. The operation in process should be terminated and the previous stored value of the variable retained.

2. Floating Point Underflow. This may be treated in the same manner as floating point overflow, or the variable may be set to zero and processing continued.

3. Divide by Zero. This should be treated in the same manner as floating point overflow.

4. Square Root. The square root of a negative number should be treated in the same manner as floating point overflow.

5. Arc Sine. If the argument is greater than +1, a value of $\pi/2$ should be returned. If the argument is less than -1, a value of $-\pi/2$ should be returned. In either case, processing shall not be terminated.

6. Arc Cosine. If the argument is greater than +1, a value of zero should be returned. If the argument is less than -1, a value of π should be returned. In either case, processing shall not be terminated.

7. Arc Tangent. If both arguments are zero, a value of zero should be returned.

8. Logarithm. If the argument of the logarithm function is negative or zero, the operation in process should be terminated, and the previous value of the result retained.

9. Sine and Cosine. The returned value of sine and cosine functions should be limited to the interval $[-1,1]$ and the functions shall be capable of accepting any valid floating point numbers as arguments.

4.0 DETAILED REQUIREMENTS

The various subsections of this section specify the detailed requirements for the Shuttle navigation system flight software package. This document contains orbital flight test (OFT) detailed requirements for Navigation and User Parameter Processing principal functions for the orbit operations computer load (on orbit and rendezvous), Operational Sequence (OPS) 2. In addition, requirements dealing with other navigation software functions during OPS 8 and OPS 00 are also addressed.

When viewed in the larger context of the total Shuttle flight software, the navigation software package documented herein is a modular system whose function is to supply various parameters required by other major modular systems, such as guidance, displays, flight control, etc. The requirements placed upon the navigation system by these various users often play a large role in determining the design structure and cyclic rate structure of the navigation system. The required interfaces between the navigation system and the other major software systems that use navigation system data are presented in the Level B CPDS document that controls all the interfaces between principal functions.

4.1 ONORBIT/RENDEZVOUS NAVIGATION SEQUENCER PRINCIPAL FUNCTION

The Onorbit/Rendezvous Navigation Sequencer principal function must initialize and sequence the Onorbit/Rendezvous navigation principal function during OPS 2 while the following major modes are active:

Major Mode 201 - Onorbit coast

Major Mode 202 - Maneuver execute

The Onorbit/Rendezvous Navigation Sequencer principal function must also initialize and sequence the Onorbit/Rendezvous Navigation principal function OPS 8 (orbital operation checkout).

The Onorbit/Rendezvous Navigation Sequencer principal function must also interface with certain crew controls on the REL_NAV and FDS DIS C/O displays relating to selection of rendezvous navigation (REND_NAV_FLAG), to selection of powered flight versus coasting flight navigation phases, and to selection of measurement processing in Major Mode 202 when in rendezvous navigation.

Detailed requirements for Onorbit/Rendezvous Navigation processing principal function are identified in the specific principal function description section 4.2. Cues (events) for performing the proper navigation initialization and sequencing during OPS 2 and OPS 8 are defined in the Level B GN&C CPDS. The particular events and resulting navigation software actions pertaining to the Onorbit/Rendezvous Navigation Sequencer principal function are shown in table 4.1-1. Dynamic parameter input/output data flow between the Onorbit/Rendezvous Navigation Sequencer principal function and other principal functions are shown in table 4.1-2.

TABLE 4.1-1.- ONORBIT/RENDEZVOUS NAVIGATION SEQUENCER EVENTS

Event no.	Description	Navigation Criteria	Navigation Action
THE FOLLOWING PERTAIN TO SEQUENCER FUNCTIONS DURING OPS 2.			
Begin coasting flight navigation phases (onorbit or rendezvous).			
60	Transition from OPS 1 to MM201 (begin OPS 2)	OPS 201 PRO	Entering into an onorbit coast navigation phase: SET: CURR_ORB_MASS = WT_DISP/G_2_FPS2 NAV_CURR_ORB_MASS = CURR_ORB_MASS CALL: OPS_2_OR_8_INITIALIZE SCHEDULE: NAV_ONORBIT_RENDEZVOUS, repeat every: DT_NAV_STATE_PROP
60A	Transition from OPS 8 to MM201 (begin OPS 2)		The REND-NAV-FLAG is assumed to be in the OFF state for this event: Same action as EVENT_60 except current Orbiter mass is calculated as: CURR_ORB_MASS=MASS_INIT
73	Transition from MM202 to MM201	OPS 201 PRO	SET: MEAS_ENABLE = OFF DOING_MEAS_ENABLE = OFF PWRD_FLT_NAV = OFF DOING_PWRD_FLT_NAV = OFF

TABLE 4.1-1.- ONORBIT/RENDEZVOUS NAVIGATION SEQUENCER EVENTS.- Continued

Event no.	Description	Navigation Criteria	Navigation Action
61	Transition from OPS 3 to MM201 (begin OPS 2)	OPS 201 PRO	The REND_NAV_FLAG is assumed to be in the OFF state. Same action as for Event 60.
	Change in the REND_NAV_FLAG setting (i.e., REND_NAV_FLAG ≠ REND_NAV_FLAG_LAST)	Crew activates REL_NAV display control (RNDZ_NAV_ENA)	If the REND_NAV_FLAG = OFF (i.e., entering an onorbit coasting navigation phase): CALL: REND_NAV_EXIT If REND_NAV_FLAG = ON (i.e., entering a rendezvous coast navigation phase): CALL: REND_NAV_INIT
Begin powered flight navigation phases (onorbit or rendezvous)			
67	Transition from MM201 to MM202	OPS 202 PRO	SET: PWRD_FLT_NAV = ON DOING_PWRD_FLT_NAV = ON
Perform special tasks upon termination of OPS 2.			
608	Transition from MM201 to OPS 8 (terminate OPS 2)	OPS 801 PRO (refer to VU, Level B CPDS)	SET: MASS_INIT = CURR_ORB_MASS R_FILT_INIT = R_FILT V_FILT_INIT = V_FILT T_FILT_INIT = T_LAST_FILT CANCEL: NAV_ONORBIT_RENDEZVOUS

79FM10

TABLE 4.1-1.- ONORBIT/RENDEZVOUS NAVIGATION SEQUENCER EVENTS.- Continued

Event no.	Description	Navigation Criteria	Navigation Action
85	Transition from MM201 to OPS 3 (terminate OPS 2)	OPS 301 PRO	Same action as for Event 60B except the current Orbiter mass(CURR_ ORB_MASS) is not saved for the memory transition as MASS_INIT
83	Transition from OPS 2 to OPS 00		Same action as for event 60B
THE FOLLOWING PERTAIN TO SEQUENCER FUNCTIONS DURING OPS 8. Begin coasting flight navigation phase (OPS 8).			SET: NAV_CURR_ORB_MASS = MASS_INIT
60B	Transition from MM201 to OPS 8 (terminate OPS 2)	OPS 801 PRO (refer to VU, Level B CPDS)	CALL: OPS_2_OR _8_INITIALIZE SCHEDULE: NAV_ONORBIT_RENDEZVOUS, repeat every: DT_NAV_STATE_PROP
Perform special tasks upon termination of OPS 8			
60A	Transition from OPS 8 to MM201	OPS 201 PRO	SET: MASS_INIT = CURR_ORB_MASS $\overline{R}$_FILT_INIT = $\overline{R}$_FILT $\overline{V}$_FILT_INIT = $\overline{V}$_FILT T_FILT_INIT = T_LAST_FILT CANCEL: NAV_ONORBIT_RENDEZVOUS

TABLE 4.1-1.- ONORBIT/RENDEZVOUS NAVIGATION SEQUENCER EVENTS.- Concluded

Event no.	Description	Navigation Criteria	Navigation Action
60H	Transition from OPS 8 to OPS 3	MSC	Same as EVENT_60A except the current Orbiter mass value (CURR_ORB_MASS) is not saved for the memory transition.

TABLE 4.1-2.- ONORBIT/RENDEZVOUS NAVIGATION SEQUENCER PRINCIPAL FUNCTION INPUT/OUTPUT

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
ATMP			Onorbit Predictor, TLM	OPS_2_OR_8_INITIALIZE, REND_NAV INIT
COV_ACCEL_UVW_INIT			Onorbit/Rend Nav	REND_COV_INIT
COV_COR_UPDATE	Ground Uplink	COVINIT_UVW		
CURR_ORB_MASS	Onorbit Guidance	NAV_EXIT	Onorbit/Rend Nav, Onorbit Guidance	ONORBIT_REND_NAV_SEQUENCER
COV_PWRD_FLT			Onorbit/Rend Nav	REND_NAV_INIT
COV_PWRD_FLT_LAST			Onorbit/Rend Nav	REND_NAV_INIT
DMP			Onorbit Predictor	OPS_2_OR_8_INITIALIZE, REND_NAV_INIT
DO_COAS_ANGLES_NAV_LAST			Onorbit/Rend Nav	REND_COV_INIT
DOING_MEAS_ENABLE			REL_NAV display, TLM	ONORBIT_REND_NAV_SEQUENCER
DOING_PWRD_FLT_NAV			REL_NAV display, TLM,FCS_DIS_C/O display	ONORBIT_REND_NAV_SEQUENCER
DOING_REND_NAV			REL_NAV display, TLM, ONORBIT/REND NAV, ONORBIT/REND USER PARAMETER PROCESSING	REND_NAV_INIT, REND_NAV_EXIT
DO_RR_ANGLES_NAV LAST			Onorbit/Rend Nav	REND_COV_INIT

TABLE 4.1-2.- ONORBIT/RENDEZVOUS NAVIGATION SEQUENCER PRINCIPAL FUNCTION INPUT/OUTPUT.- Continued

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
DO_RRDOT_NAV_LAST			Onorbit/Rend Nav	REND_COV_INIT
DO_ST_ANGLES_NAV_LAST			Onorbit/Rend Nav	REND_COV_INIT
DT_FILT	Onorbit/Rend Nav	REND_COV_INIT		
DV_COV			Onorbit/Rend Navigation, TLM	COV_LAST_RESET
DV_FILT	Onorbit/Rend Nav	REND_COV_INIT		
E			Onorbit/Rend Nav, TLM	COVINIT_UVW, UA_BIAS_AND_COVINIT
EVENT_E1	Moding, Sequencing and Control	ONORBIT_REND_NAV_SEQUENCER	TLM	
EVENT_60	Moding, Sequencing and Control	ONORBIT_REND_NAV_SEQUENCER	TLM	
EVENT_60A	Moding, Sequencing and Control	ONORBIT_REND_NAV_SEQUENCER, NAV_EXIT	TLM	
EVENT_60B	Moding, Sequencing and Control	ONORBIT_REND_NAV_SEQUENCER, NAV_EXIT	TLM	

TABLE 4.1-2.- ONORBIT/RENDEZVOUS NAVIGATION SEQUENCER PRINCIPAL FUNCTION INPUT/OUTPUT.- Continued

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
EVENT_60H	Moding, Sequencing and Control	ONORBIT_ REND_NAV_ SEQUENCER	TLM	
EVENT_67	Moding, Sequencing and Control	ONORBIT_ REND_NAV_ SEQUENCER	TLM	
EVENT_73	Moding, Sequencing and Control	ONORBIT_ REND_NAV_ SEQUENCER	TLM	
EVENT_83	Moding, Sequencing and Control	ONORBIT_ REND_NAV_ SEQUENCER, NAV_EXIT	TLM	
EVENT_84	Moding, Sequencing and Control	ONORBIT_ REND_NAV_ SEQUENCER	TLM	
EVENT_85	Moding, Sequencing and Control	ONORBIT_ REND_NAV_ SEQUENCER	TLM	
FILT_UPDATE			Onorbit/Rendezvous User Parameter Processing	SHUTTLE_RESET
GMDP			Onorbit Predictor	OPS_2_OR_8_ INITIALIZE, REND_NAV_INIT
GMOP			Onorbit Predictor	OPS_2_OR_8_ INITIALIZE, REND_NAV_INIT
G _TV_LAST			Onorbit/Rend Nav	REND_COV_INIT

TABLE 4.1-2.- ONORBIT/RENDEZVOUS NAVIGATION SEQUENCER PRINCIPAL FUNCTION INPUT/OUTPUT.- Continued

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
I_CYCLE			Onorbit/Rend Nav, TLM	REND_COV_INIT
IDRAG	Onorbit/Rend Nav	REND_COV_INIT		
IGD	Onorbit/Rend Nav	REND_COV_INIT		
IGO	Onorbit/Rend Nav	REND_COV_INIT		
IVENT	Onorbit/Rend Nav	REND_COV_INIT		
MASS_INIT	Onorbit/Rend Nav. Seq.	ONORBIT_REND_NAV_SEQUENCER	Onorbit/Rend Nav Seq	NAV_EXIT
MEAS_ENABLE			Onorbit/Rend Nav	ONORBIT_REND_NAV_SEQUENCER
NAV_CURR_ORB_MASS			TLM	ONORBIT_REND_NAV_SEQUENCER
N _ACCEPT			Onorbit/Rend Nav, TLM	DISPLAY_COUNT_INIT
NOISY_NAV_MEAS			Onorbit/Rend Nav	REND_NAV_INIT
N _REJECT			Onorbit/Rend Nav, TLM	DISPLAY_COUNT_INIT
OPS_2_OR_8_INITIALIZE_COMPLETE			Onorbit/Rend User Parameter Processing Seq, MSC	OPS_2_OR_8_INITIALIZE
PRED_ORB_AREA			Onorbit Predictor, TLM	OPS_2_OR_8_INITIALIZE
PRED_ORB_CD			Onorbit Predictor, TLM	OPS_2_OR_8_INITIALIZE

TABLE 4.1-2.- ONORBIT/RENDEZVOUS NAVIGATION SEQUENCER PRINCIPAL FUNCTION INPUT/OUTPUT.- Continued

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
PRED_ORB_MASS			Onorbit Predictor, TLM	OPS_2_OR_8_INITIALIZE
PRED_STEP			Onorbit Predictor, TLM	OPS_2_OR_8_INITIALIZE, REND_NAV_INIT
PRED_USE	Onorbit/Rend Nav	REND_NAV_INIT	Onorbit/Rend Nav, TLM	REND_NAV_INIT, STATE_VECTOR_PREDICT_TASK
PWRD_FLT_NAV			TLM, Onorbit/Rend Nav, ORB UPP	ONORBIT_REND_NAV_SEQUENCER, OPS_2_OR_8_INITIALIZE
REND_NAV_FLAG	REL_NAV display	ONORBIT_REND_NAV_SEQUENCER, REND_NAV_INIT	TLM	REND_NAV_INIT
R _FILT	Onorbit/Rend Nav	REND_COV_INIT, NAV_EXIT, COV_LAST_RESET	Onorbit/Rend Nav	OPS_2_OR_8_INITIALIZE
R _FILT_INIT	ASC NAV SEQ., D/L NAV SEQ., Onorbit/Rend Nav Seq	OPS_2_OR_8_INITIALIZE	D/L Nav Seq, Onorbit/Rend Nav Seq	NAV_EXIT
R _LAST			Onorbit/Rend Nav	COV_LAST_RESET

TABLE 4.1-2.- ONORBIT/RENDEZVOUS NAVIGATION SEQUENCER PRINCIPAL FUNCTION INPUT/OUTPUT.- Continued

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
R _PRED_FINAL	Onorbit Predictor	OPS_2_OR_8_INITIALIZE, REND_NAV_INIT, STATE_VECTOR_PREDICT_TASK	TLM	
R _PRED_INIT			Onorbit Predictor, TLM	OPS_2_OR_8_INITIALIZE, REND_NAV_INIT, STATE_VECTOR PREDICT_TASK
R _RESET			Onorbit/Rend User Parameter Processing	SHUTTLE_RESET
R _TV	Onorbit/Rend Nav	REND_NAV_INIT, COV_LAST_RESET	Onorbit/Rend Nav	REND_NAV_INIT
R _TV_LAST			Onorbit/Rend Nav	COV_LAST_RESET
R _TV_RESET			Onorbit/Rend User Parameter Processing	TARGET_RESET
SEQ_ACCEPT			Onorbit/Rend Nav, TLM	DISPLAY_COUNT_INIT
SEQ_REJECT			Onorbit/Rend Nav, TLM	DISPLAY_COUNT_INIT
SIG_UPDATE	Ground Uplink	COVINIT_UVW		
SQR_EMU			Onorbit/Rend Nav, Onorbit Predictor	OPS_2_OR_8_INITIALIZE

TABLE 4.1-2.- ONORBIT/RENDEZVOUS NAVIGATION SEQUENCER PRINCIPAL FUNCTION INPUT/OUTPUT.- Continued

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
TAU_UNMOD_ACC_COV			Onorbit/Rend Nav	REND_NAV_INIT
T_COV_LAST			Onorbit/Rend Nav	COV_LAST_RESET
T_CURRENT_FILT			TLM	OPS_2_OR_8_INITIALIZE
T_FILT_INIT	Deorbit/Landing Nav Seq, Ascent Nav Seq, Onorbit/Rend Nav Seq	OPS_2_OR_8_INITIALIZE	D/L Nav Seq, Onorbit/Rend Nav Seq	NAV_EXIT
T_IMUS_GA	IMU INT PROC	IMU DATA SNAP		
T_LAST_FILT	Onorbit/Rend Nav	NAV_EXIT, REND_COV_INIT, COV_LAST_RESET	Onorbit/Rend Nav	OPS_2_OR_8_INITIALIZE
TOT_ACC_LAST			Onorbit/Rend Nav	REND_COV_INIT
T_PRED_FINAL			Onorbit Predictor, TLM	OPS_2_OR_8_INITIALIZE, STATE_VECTOR_PREDICT_TASK
T_PRED_INIT			Onorbit Predictor, TLM	OPS_2_OR_8_INITIALIZE, REND_NAV_INIT, STATE_VECTOR_PREDICT_TASK
T_RESET	Onorbit/Rend Nav	STATE_VECTOR_PREDICT_TASK	Onorbit/Rend User Parameter Processing, Onorbit/Rend Nav	SHUTTLE_RESET

TABLE 4.1-2.- ONORBIT/RENDEZVOUS NAVIGATION SEQUENCER PRINCIPAL FUNCTION INPUT/OUTPUT.- Continued

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
T_TV	Onorbit/Rend Nav	REND_NAV_INIT	TLM, Onorbit/Rend Nav	REND_NAV_INIT
TV_PREDICT_FAIL			TLM	REND_NAV_INIT
UNMOD_ACC_BIAS			Onorbit/Rend Nav	REND_NAV_EXIT, OPS_2_OR_8_INITIALIZE, U_A_BIAS_AND_COV_INIT
VAR_UNMOD_ACC			Onorbit/Rend Nav	REND_NAV_INIT
V _CURRENT_FILT			TLM	OPS_2_OR_8_INITIALIZE
V _FILT	Onorbit/Rend Nav	REND_COV_INIT, NAV_EXIT, COV_LAST_RESET	Onorbit/Rend Nav	OPS_2_OR_8_INITIALIZE
V _FILT_INIT	Asc. Nav Seq, D/L Nav Seq, Onorbit/Rend Nav Sequencer	OPS_2_OR_8_INITIALIZE	D/L Nav Seq, Onorbit/Rend Nav Sequencer	NAV_EXIT
V _IMU_CURRENT	IMU RM	IMU DATA SNAP		
V _IMU_RESET			Onorbit/Rend User Parameter Processing, TLM	SHUTTLE_RESET
V _LAST			Onorbit/Rend Nav	COV_LAST_RESET
V _LAST_FILT			Onorbit/Rend Nav	OPS_2_OR_8_INITIALIZE

TABLE 4.1-2.- ONORBIT/RENDEZVOUS NAVIGATION SEQUENCER PRINCIPAL FUNCTION INPUT/OUTPUT.- Concluded

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
VMP			Onorbit Predictor	OPS_2_OR_8_INITIALIZE, REND_NAV_INIT
V _PRED_FINAL	Onorbit Predictor	REND_NAV_INIT, OPS_2_OR_8_INITIALIZE, STATE_VECTOR_PREDICT_TASK	TLM	
V _PRED_INIT			Onorbit Predictor, TLM	OPS_2_OR_8_INITIALIZE, REND_NAV_INIT, STATE_VECTOR_PREDICT_TASK
V _RESET			Onorbit/Rend User Parameter Processing	SHUTTLE_RESET
V _TV	Onorbit/Rend Nav	REND_NAV_INIT, COV_LAST_RESET	Onorbit/Rend Nav	REND_NAV_INIT
V _TV_LAST			Onorbit/Rend Nav	COV_LAST_RESET
V _TV_RESET			Onorbit/Rend User Parameter Processing	TARGET_RESET
WT_DISP	Deorbit maneuver dip; Ascent maneuver dip	ONORBIT_REND_NAV_SEQUENCER		

4.1.1 Sequencing Operations and Major Mode Transitions (ONORBIT_REND_NAV_SEQUENCER)

The Onorbit/Rendezvous Navigation Sequencer principal function performs specific actions which control the execution of the Onorbit/Rendezvous Navigation principal function based on navigation phase and major mode transitions. These tasks are performed by the Sequencing Operations and Major Mode Transitions subfunction and include the following tasks for OPS 2 control:

- response to events for entering OPS 2
- response to events for exiting OPS 2
- response to events for major mode transitions within OPS 2
- response to crew entry on the REL NAV DISPLAY regarding selection of powered flight phase (both onorbit and rendezvous)
- response to crew entry (REL NAV DISPLAY) regarding selection of onorbit or rendezvous navigation phase (powered or coasting flight)

For OPS 8 control the following tasks are performed:

- response to events for entering OPS 8
- response to events for exiting OPS 8
- response to crew entry on the FCS/DED DISP C/O DISPLAY regarding selection of powered flight phase

The events for OPS 2 and OPS 8 navigation control are summarized below. Detailed requirements are specified separately below for OPS 2 and OPS 8.

- Events to enter OPS-2
    - 60 OPS-1 → OPS-2
    - 60A OPS-8 → OPS-2
    - E1 OPS-3 → OPS-2
    - 84 OPS-00 → OPS-2
- Events to exit OPS-2
    - 60B OPS-2 → OPS-8
    - 83 OPS-2 → OPS-00
    - 85 OPS-2 → OPS-3
- Events to enter OPS-8
    - 60B OPS-2 → OPS-8

- Events to exit OPS-8
  - 60A OPS-8 → OPS-2
  - 60H OPS-8 → OPS-3
- Events during OPS-2 & 8
  - 73 MM202 → 201
  - 67 MM201 → 202

A. Detailed requirements.

Part I.- OPS 2 Requirements

The detailed sequencing and major mode transition requirements for the OPS 2 portion of the Onorbit/Rendezvous Navigation Sequencer principal function are described as follows:

1. If an onorbit coast navigation phase is begun by entry into Major Mode 201 from OPS 1, OPS 3, OPS 8 or OPS 00 (EVENTS E1, 60, 60A, or 84), the Onorbit/Rendezvous Navigation Sequencer principal function shall provide the capability to initialize the Orbiter state vector, mass, and other required navigation parameters on the basis of prestored computer locations unaffected by the computer program memory load reconfiguration. The following sequence should be followed:

   a. Compute the value for Orbiter mass as follows:

   If OPS 2 is to be entered from OPS 1 or OPS 3 the Orbiter mass shall be calculated as:

   CURR_ORB_MASS = WT_DISP/G_2_FPS2 F3

   Otherwise, Orbiter mass shall be reset to a saved value:

   CURR_ORB_MASS = MASS_INIT

   b. Store the computed mass value into a NAV slot for use by navigation.

   NAV_CURR_ORB_MASS = CURR_ORB_MASS

   c. An initialization operation shall now be performed to obtain current IMU data, predict the last saved OPS sequence Orbiter position and velocity vectors to current time, reset parameters for user parameter state propagation, and initialize flags to OPS 2 or OPS 8 initial values (refer to section 4.1.2.1 for detailed requirements):

F3 This equation shall be protected against division by zero (reference 3.6-3).

CALL: OPS_2_OR_8_INITIALIZE

d. After completion of initialization, the capability shall be provided for sequencing the Onorbit/Rendezvous Navigation principal function at the designated repetition rate, DT_NAV_STATE_PROP.

2. If OPS 2 is not being entered, tests for detecting leaving OPS 2 to go to OPS 8 (EVENT 60B), OPS 00 (EVENT 83), or OPS 3 (EVENT 85) are made. If any of these 3 transitions is detected,

   CALL: NAV_EXIT

   (section 4.1.1.2) to store the necessary information into protected locations for the new OPS sequence.

3. If transitions into or out of OPS 2 have not been detected, tests for transitions in major mode in OPS 2 are made.

   a. The test for transition from MM202 to MM201 (EVENT_73) is made. If EVENT_73 is ON, the powered flight flag, MM202 measurement enable flag, and the measurement enable positive feedback flag (for REL_NAV_DISPLAY) must be set to OFF.

      PWRD_FLT_NAV = OFF

      MEAS_ENABLE = OFF

      DOING_MEAS_ENABLE = OFF

   b. If EVENT 73 is OFF, the test for transition from MM201 to MM202 (EVENT 67) is made. If EVENT_67 is ON, the powered flight flag is set to ON.

      PWRD_FLT_NAV = ON

   In any event, the positive feedback flag for powered flight is set for the REL NAV display to the value of the powered flight navigation flag.

   DOING_PWRD_FLT_NAV = PWRD_FLT_NAV

4. A test is made to detect the following:

   - A transition into or out of the rendezvous navigation phase, or
   - the rendezvous navigation initialization target prediction task has been initiated but not completed.

   a. If REND_NAV_FLAG ≠ REND_NAV_FLAG_LAST, a transition has occurred; or if REND_NAV_INIT_PRED = ON, then the target prediction task has not been completed. Thus the REND_NAV_FLAG and REND_NAV_INIT_FLAG are interrogated to determine the nature of the transition and the state of the target prediction.

(1) If REND_NAV_FLAG = ON or REND_NAV_INIT_PRED = ON then the rendezvous navigation initialization subfunction is invoked

CALL: REND_NAV_INIT

(See section 4.1.2.2)

(2) If REND_NAV_FLAG = OFF and REND_NAV_INIT_PRED = OFF, the rendezvous navigation phase will be exited.

CALL: REND_NAV_EXIT

(See section 4.1.1.1)

b. If REND_NAV_FLAG = REND_NAV_FLAG_LAST and REND_NAV_INIT_PRED = OFF, then a transition has not occurred; and the rendezvous navigation initialization prediction task is not in progress. Thus the sequencer control logic has been completed for this navigation cycle.

Part II.- OPS 8 Requirements:

The detailed sequencing and major mode transition requirements for the OPS 8 portion of the Onorbit/Rendezvous Navigation Sequencer principal function are described as follows:

1. If the onorbit coast navigation phase is begun by entry into GN&C OPS 8 from OPS 2 (Event 60B), the Onorbit/Rendezvous Navigation Sequencer principal function shall provide the capability to initialize the Orbiter state vector and other required navigation parameters on the basis of prestored data and OPS 2 data obtained from protected computer locations unaffected by the computer program memory load reconfiguration. A flag, REND_NAV_FLAG, will be maintained in the OFF state (by MSC) throughout OPS 8 as rendezvous navigation phase is not available in this OPS sequence. This following initialization sequence shall be performed:

   a. Initialize the current Orbiter mass as saved from OPS 2:

      NAV_CURR_ORB_MASS = MASS_INIT

   b. Perform an initialization operation to obtain current IMU data, predict the last saved OPS sequence Orbiter position and velocity vectors to current time, reset parameters for user parameter state propagation, and initialize flags to OPS 2 or OPS 8 initial values (refer to section 4.1.2.1 for detailed requirements):

      CALL: OPS_2_OR_8_INITIALIZE

c. After completion of this initialization, the capability shall be provided for sequencing the Onorbit/Rendezvous Navigation principal function at the designated repetition rate (DT_NAV_STATE_PROP) during OPS 8.

2. If a transition for entering OPS 8 was not detected, tests are made to detect leaving OPS 8 to enter OPS 2 (EVENT 60A) or OPS 3 (EVENT 60H). If either event is "ON",

   CALL: NAV_EXIT

   to store necessary information into protected locations for the next OPS sequence.

3. If transitions into and out of OPS 8 have not been detected, the positive feedback flag for powered flight navigation is set (for the FCS/DED DISP C/O DISPLAY) to the value of the powered flight navigation flag.

   DOING_PWRD_FLT_NAV = PWRD_FLT_NAV

B. Interface Requirements. Input and output parameters are given in Table 4.1.1.

C. Processing Requirements. None

D. Constraints. The following additional constraints apply to the requirements presented in Section A:

1. The following flags are set by either REL_NAV (via crew input) or FCS/DED DISP C/O DISPLAY (crew input):

   REND_NAV_FLAG

   PWRD_FLT_NAV

   MEAS_ENABLE

   and should not be changed (in value) by these external functions during any given cycle of the Onorbit/Rendezvous Navigation principal function during any navigation phase since a timewise consistent set of navigation data is required (i.e., a completion of a navigation cycle) to perform the various navigation functions.

2. Memory transitions shall only be performed following the completion of a navigation cycle.

3. The maximum repetition rate for the Onorbit/Rendezvous Navigation Sequencer principal function shall be 1.92 seconds.

E. <u>Supplemental Information</u>. A suggested implementation of these requirements is illustrated in the Appendix B flow diagrams:

ONORBIT_REND_NAV_SEQUENCER (OPS 2)

ONORBIT_REND_NAV_SEQUENCER (OPS 8)

NAV_EXIT

TABLE 4.1.1.- ONORBIT_REND_NAV_SEQUENCER INPUT/OUTPUT

Variable Name	Input Source	Output Destination
CURR_ORB_MASS		*
DOING_MEAS_ENABLE	**	*
DOING_PWRD_FLT_NAV	**	*
DOING_REND_NAV	**	
DT_NAV_STATE_PROP	**	
EVENT_E1	*	*
EVENT_60	*	*
EVENT_60A	*	*
EVENT_60B	*	*
EVENT_60H	*	*
EVENT_67	*	*
EVENT_73	*	*
EVENT_83	*	*
EVENT_84	*	*
EVENT_85	*	*
G_2_FPS2	**	
MASS_INIT	*	
MEAS_ENABLE		*
NAV_CURR_ORB_MASS		*, ACCEL_ONORBIT, OPS_2_OR_8_INITIALIZE
PWRD_FLT_NAV	OPS_2_OR_8_INITIALIZE	*
REND_NAV_FLAG	*, REND_NAV_INIT	*
REND_NAV_FLAG_LAST	**, REND_NAV_INIT, REND_NAV_EXIT	
REND_NAV_INIT_PRED	**, REND_NAV_INIT	
WT_DISP	*	

*See principal function I/O table for the Onorbit/Rendezvous Navigation Sequencer (table 4.1-2)

**See initialization parameters, section 4.7

4.1.1.1 Rendezvous Navigation Phase Termination (REND_NAV_EXIT)

The purpose of the rendezvous navigation phase termination subfunction is to perform the necessary operations required when a rendezvous navigation phase is terminated in preparation for a transition to an onorbit navigation phase. This action occurs during OPS 2 when the crew or MSC changes the value of the REND_NAV_FLAG from a value of ON to OFF (crew control of the REND_NAV_FLAG is via the REL_NAV display control, RNDZ_NAV_ENA). This subfunction is invoked by the sequencing operations and major mode transitions subfunction (section 4.1.1, ONORBIT_REND_NAV_SEQUENCER).

A. Detailed Requirements. When called, this subfunction shall first set the unmodeled acceleration biases to zero. These bias terms were solved for by the rendezvous navigation filter and need to be set to zero for non-rendezvous navigation phases.

  UNMOD_ACC_BIAS = 0.

  The display parameters initialization subfunction shall now be executed to zero the ACCEPT/REJECT counters for display and measurement processing requirements:

  CALL: DISPLAY_COUNT_INIT

  Detailed requirements for the above subfunction are stated in section 4.1.2.2.1.2.

  The "last" flag is turned OFF to indicate the rendezvous navigation is no longer active.

  REND_NAV_FLAG_LAST = OFF

  The positive feedback flag for the Rel Nav Display is set to OFF to indicate that rendezvous navigation is no longer active.

  DOING_REND_NAV = OFF

B. Interface Requirements. Input and output parameters for this subfunction may be found in table 4.1.1.1.

C. Processing Requirements. This subfunction shall be called by the sequencing operations and major mode transitions subfunction (ONORBIT_REND_NAV_SEQUENCER) when it is required to transition from rendezvous to onorbit navigation phases during OPS 2.

D. Constraints. None

E. Supplemental Information. A suggested implementation of this subfunction may be found in Appendix B flow diagrams:

  REND_NAV_EXIT

TABLE 4.1.1.1.- REND_NAV_EXIT INPUT/OUTPUT

Variable Name	Input Source	Output Destination
UNMOD_ACC_BIAS		*
REND_NAV_FLAG_LAST		ONORBIT_REND_NAV_ SEQUENCER
DOING_REND_NAV		*

*See principal function Input/Output table 4.1-2 for the Onorbit/Rendezvous Navigation Sequencer

4.1.1.2 Operation Sequencer Termination (NAV_EXIT)

Specific actions are required by the onboard navigation software when either the OPS-2 or OPS-8 operational sequence is terminated in preparation for transition to another OPS sequence. These actions are primarily concerned with protecting special memory locations to preserve data required for the next OPS sequence.

A. Detailed Requirements. The Operations Sequence Termination subfunction shall provide the capability to save off (in protected memory locations) certain navigation-related data sets for transmission across a memory transition from one operational sequence to another and, once the data has been saved, terminate the Onorbit/Rendezvous Navigation principal function. This subfunction shall perform these functions for the termination of OPS-2 and OPS-8 as follows:

1. If OPS-2 or OPS-8 is being terminated for a transition to either OPS-8 (from OPS-2, Event 60B), or to OPS-2 (from OPS-8, Event 60A), or to OPS-00 (from OPS-2, Event 83) the current Orbiter mass must be saved prior to termination:

   MASS_INIT = CURR_ORB_MASS

2. In all cases where OPS-2 or OPS-8 is terminated, the following parameters shall be saved off prior to termination:

   $\underline{R}$ _FILT_INIT = $\underline{R}$ _FILT

   $\underline{V}$ _FILT_INIT = $\underline{V}$ _FILT

   T_FILT_INIT = T_LAST_FILT

   Although the variable names with the _INIT have been designated as unique variables, this may not be required if the same physical core location can be used for $\underline{R}$ _FILT (for example) in each memory load. The _INIT notation has been used for visibility purposes only.

3. During OPS-2 and OPS-8 once the above data have been stored, execution of the Onorbit/Rendezvous Navigation principal function shall be cancelled.

   CANCEL: NAV_ONORBIT_RENDEZVOUS

B. Interface Requirements. Input and output requirements for this subfunction are defined in table 4.1.1.2.

C. Processing Requirements. This subfunction is called by the following subfunction:

   ONORBIT_REND_NAV_SEQUENCER (section 4.1.1)

D. Constraints. None

E. Suggested Implementation. A suggested implementation in the form of a detailed flowchart may be found in Appendix B under the name NAV_EXIT.

TABLE 4.1.1.2 NAV_EXIT INPUT/OUTPUT

Variable Name	Input Source	Output Destination
EVENT_60B	*	
EVENT_60A	*	
EVENT_83	*	
CURR_ORB_MASS	*	
R _FILT	*	
V _FILT	*	
T_LAST_FILT	*	
MASS_INIT		*
R _FILT_INIT		*
V _FILT_INIT		*
T_FILT_INIT		*

*See principal function I/O table for ONORBIT_REND_NAV_SEQUENCER.

4.1.2 Operation Sequence Initialization (OPS_2_OR_8_INITIALIZE)

The purpose of the operation sequence initialization subfunction is to perform specific functions required when either operations sequence 2 (OPS 2) or 8 (OPS 8) is entered from another OPS sequence (i.e., OPS 1, OPS 3, OPS 00, OPS 2 or OPS 8).

A. Detailed Requirements.

1. Snap current accumulated inertial measurement unit (IMU)-sensed velocity data and associated time tag (section 4.2.2.1), SNAP (V _CURRENT FILT, T_CURRENT_FILT) and immediately store the snapped IMU and time data in local variable locations.

   V _LAST_FILT = V _CURRENT_FILT

   T_LAST_FILT = T_CURRENT_FILT

2. Initialize the unmodeled acceleration biases solved for in the filter to zero. These bias terms are solved for by the rendezvous navigation filter and need to be zeroed for non-rendezvous navigation phases:

   UNMOD_ACC_BIAS = 0.

   - Compute the square root of MU of the Earth for the precise predictor and precision state propagation:

   SQR_EMU = SQRT(EARTH_MU)

3. Set up the proper parameters to predict the stored Orbiter state vector to current time (refer to section 4.3 for predictor requirements; the following are unique COMPOOL locations for use by the predictor):

   PRED_ORB_AREA = REF_ORB_AREA

   PRED_ORB_MASS = NAV_CURR_ORB_MASS

   PRED_ORB_CD = REF_ORB_CD

   GMDP = GM_DEG

   GMOP = GM_ORD

   DMP = DFL

   VMP = VFLOV_PRED

   ATMP = 1

   PRED_STEP = PRED_STEP_OPS_INIT

   T_PRED_INIT = T_FILT_INIT

R _PRED_INIT = R _FILT_INIT

V _PRED_INIT = V _FILT_INIT

T_PRED_FINAL = T_LAST_FILT

Then call the onorbit precise prediction principal function

CALL: ONORBIT_PREDICT

4. Finally, reset the following onorbit navigation parameters:

R _FILT = R _PRED_FINAL

V _FILT = V _PRED_FINAL

5. Initialize those parameters required by the user parameter state propagation subfunction (section 4.5.1) through the use of the Orbiter state vector reset task (section 4.1.2.1) for the Orbiter vehicle only:

CALL: SHUTTLE_RESET

6. Set a flag that indicates use of the coasting flight state propagation algorithm.

PWRD_FLT_NAV = OFF

7. Signal that the proper initialization has been accomplished to allow the Onorbit/Rendezvous User Parameter Processing Sequencer principal function to begin scheduling.

SET: OPS_2_OR_8_INITIALIZE COMPLETE = ON

B. Interface Requirements. The input and output requirements for this subfunction are described in table 4.1.2.

C. Processing Requirements. This subfunction is called by the sequencing operations and major mode transitions subfunction, section 4.1.1; ONORBIT_REND_NAV_SEQUENCER.

D. Constraints. None

E. Supplemental Information. A suggested implementation of this module can be found in Appendix B; OPS_2_OR_8_INITIALIZE.

TABLE 4.1.2.- OPS_2_OR_8_INITIALIZE INPUT/OUTPUT

Variable Name	Input Source	Output Destination
V _CURRENT_FILT	IMU DATA SNAP	*
T_CURRENT_FILT	IMU DATA SNAP	*
V _LAST_FILT		*,SHUTTLE_RESET
T_LAST_FILT		*,SHUTTLE_RESET
EARTH_MU	* *	
SQR_EMU		*
REF_ORB_AREA	* *	
REF_ORB_CD	* *	
NAV_CURR_ORB_MASS	ONORBIT_REND_NAV_ SEQUENCER	
GM_DEG	* *	
GM_ORD	* *	
DFL	* *	
VFLOV_PRED	* *	
PRED_STEP_OPS_INIT	* *	
R _FILT_INIT	*	
V _FILT_INIT	*	
T_FILT_INIT	*	
R _FILT		*,SHUTTLE_RESET
V _FILT		*,SHUTTLE_RESET
T_PRED_INIT		*

*See principal function Input/Output table 4.1-2 for the Onorbit/Rendezvous Navigation Sequencer

**See Initialization parameters, section 4.7

TABLE 4.1.2.- OPS_2_OR_8_INITIALIZE INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
GMDP		*
GMOP		*
DMP		*
VMP		*
ATMP		*
PRED_STEP		*
R _PRED_INIT		*
V _PRED_INIT		*
PWRD_FLT_NAV		*,ONORBIT_REND_NAV_ SEQUENCER
OPS_2_OR_8_ INITIALIZE_COMPLETE		*
R _PRED_FINAL	*	
V _PRED_FINAL	*	
UNMOD_ACC_BIAS		*
PRED_ORB_AREA		*
PRED_ORB_CD		*
PRED_ORB_MASS		*
T_PRED_FINAL		*

*See principal function Input/Output table 4.1-2 for the Onorbit/Rendezvous Navigation Sequencer

4.1.2.1 Orbiter State Vector Reset (SHUTTLE_RESET)

The purpose of the Orbiter state vector reset subfunction is to provide updated state vector associated parameters to the user parameter state propagation subfunction (section 4.5.1) at the beginning of OPS sequence 2 or OPS 8 and at the end of every navigation cycle through the use of navigated state parameters.

A. Detailed Requirements. At the completion of each OPS sequence 2 or 8 initialization procedure and at the completion of a navigation cycle, the Orbiter state vector reset subfunction shall be called:

CALL: SHUTTLE_RESET

1. This subfunction shall then initialize the reset Orbiter state vector and associated IMU sensed velocity reading as follows:

$\underline{R}$ _RESET = $\underline{R}$ _FILT

$\underline{V}$ _RESET = $\underline{V}$ _FILT

$\underline{V}$ _IMU_RESET = $\underline{V}$ _LAST_FILT

T_RESET = T_LAST_FILT

2. Additionally, a flag shall be set to indicate to the user parameter state propagation subfunction that a navigated state update has occurred.

FILT_UPDATE = ON

These parameters are required by the user parameter propagator to reinitialize the user parameter state following the completion of each navigation cycle.

B. Interface Requirements. The input and output requirements for this subfunction are shown in table 4.1.2.1.

C. Processing Requirements. This subfunction is called by the following modules:

OPS_2_OR_8_INITIALIZE

NAV_ONORBIT_RENDEZVOUS

D. Constraints. None

E. Supplemental Information. A suggested implementation of this subfunction can be found in Appendix B in the form of a flow diagram:

SHUTTLE_RESET

TABLE 4.1.2.1.- SHUTTLE_RESET INPUT/OUTPUT

Variable Name	Input Source	Output Destination
$\underline{R}$ _FILT	OPS_2_OR_8_INITIALIZE, ONORBIT_REND_R_V_STATE_PROP, REND_NAV_FILTER	
$\underline{V}$ _FILT	OPS_2_OR_8_INITIALIZE, ONORBIT_REND_R_V_STATE_PROP, REND_NAV_FILTER	
T_LAST_FILT	OPS_2_OR_8_INITIALIZE, ONORBIT_REND_R_V_STATE_PROP	
$\underline{V}$ _LAST_FILT	OPS_2_OR_8_INITIALIZE, ONORBIT_REND_R_V_STATE_PROP	
$\underline{R}$ _RESET		*
$\underline{V}$ _RESET		*
$\underline{V}$ _IMU_RESET		*
T_RESET		*, STATE_VECTOR_PREDICT_TASK
FILT_UPDATE		*

*See principal function Input/Output table 4.1-2 for the Onorbit/Rendezvous Navigation Sequencer outputs and table 4.2 for the Onorbit/Rendezvous Navigation outputs

4.1.2.2 Rendezvous Navigation Initialization (REND_NAV_INIT)

The Rendezvous Navigation Initialization subfunction is responsible for the proper initialization of selected rendezvous related parameters whenever the Onorbit/Rendezvous Navigation Sequencer principal function has detected the following:

(a) There is a request to initalize the rendezvous navigation phase, or

(b) The prediction of the target vehicle has been initiated by the rendezvous navigation initialization subfunction but the prediction has not been completed.

This subfunction shall be responsible for the following tasks.

- Schedule the state vector prediction task in order to predict the target position and velocity vectors to current time.
- Set selected flags to OFF to insure the proper functioning of the Onorbit/Rendezvous Navigation principal function should the rendezvous navigation phase be canceled and then re-entered.
- Set certain parameters for the unmodeled acceleration bias propagation to values used for coasting flight.
- Initialize the covariance matrix.
- Store the current target position and velocity vectors for use by the User Parameter Processing principal function.

A. Detailed Requirements. This subfunction shall perform the following steps in the order indicated:

1. The prediction task indicator flag, PRED_USE, is interrogated for a zero value to determine if the state vector prediction task is available for scheduling. If PRED_USE = 0 then the following parameters are defined preceding the scheduling of the state vector prediction task.

REND_NAV_INIT_PRED = ON
PRED_USE = 4
TV_PREDICT_FAIL = OFF
GMDP = GM_DEG
GMOP = GM_ORD
DMP = DFL
VMP = VFLTV_PRED
ATMP = ATFL_TV
PRED_STEP = PREC_STEP_PRED
$\underline{R}$_PRED_INIT = $\underline{R}$_TV
$\underline{V}$_PRED_INIT = $\underline{V}$_TV
T_PRED_INIT = T_TV

The state vector prediction task is then scheduled.

SCHEDULE: STATE_VECTOR_PREDICT_TASK

2. If the PRED_USE flag is nonzero, it is tested for a value of 6 which indicates that the state vector prediction task has determined that the prediction interval is too large; hence no prediction will take place.

If PRED_USE = 6 then the following flags are set.

PRED_USE = 0
TV_PREDICT_FAIL = ON
REND_NAV_INIT_PRED = OFF
REND_NAV_FLAG = OFF

The rendezvous navigation flag is set to OFF so that the rendezvous navigation initialization subfunction will not be invoked again until the REND_NAV_FLAG is reset to ON by the crew.

3. If the PRED_USE flag is not set to 6, then it is tested for a value of 5 which indicates that the state vector prediction task has successfully predicted the target state vectors.

If PRED_USE = 5 then the following actions are taken.

(a) The predictor outputs are stored, and flags are set to free the predictor for other navigation users.

PRED_USE = 0
T _TV = T_PRED_FINAL
R _TV = R _PRED_FINAL
V _TV = V _PRED_FINAL
REND_NAV_INIT_PRED = OFF

(b) Next, the REND_NAV_FLAG is tested for an ON value to determine if the crew request for activating the rendezvous navigation phase is still valid. If the REND_NAV_FLAG is on, the following actions are taken:

(1) Flags are set to indicate that the unmodeled acceleration bias state statistics are to be initialized to their coast values.

NOISY_NAV_MEAS = OFF
COV_PWRD_FLT = OFF
COV_PWRD_FLT_LAST = OFF

(2) Unmodeled acceleration bias state statistical parameters are initialized to their coast values.

TAU_UNMOD_ACC_COV = TAU_U_A_COAST
VAR_UNMOD_ACC = VAR_U_A_COAST
COV_ACCEL_UVW_INIT = COV_U_A_COAST

(3) The rendezvous covariance initialization function is called to initialize the covariance matrix as described in section 4.1.2.2.1.

CALL: REND_COV_INIT

(4) The target state vector reset subfunction resets the user parameter processing state vectors as described in section 4.1.2.2.2.

CALL: TARGET_RESET

(5) The flags REND_NAV_FLAG_LAST and DOING_REND_NAV are set to ON to indicate that the rendezvous navigation phase is now active.

REND_NAV_FLAG_LAST = ON
DOING_REND_NAV = ON

B. Interface Requirements. Input and output parameters are specified in table 4.1.2.2.

C. Processing Requirements. The subfunction is called by ONORBIT_REND_NAV_SEQUENCER (section 4.1.1).

D. Constraints. None

E. Supplemental Information. A suggested implementation of these requirements is illustrated in Appendix B with REND_NAV_INIT.

TABLE 4.1.2.2.- REND_NAV_INIT INPUT/OUTPUT

Variable Name	Input Source	Output Destination
ATFL_TV	**	
ATMP		*
COV_ACCEL_UVW_INIT		U_A_BIAS_AND_COVINIT,*
COV_PWRD_FLT		*
COV_PWRD_FLT_LAST		*
COV_U_A_COAST	**	
DFL	**	
DMP		*
DOING_REND_NAV		*
GM_DEG	**	
GMDP		*
GMOP		*
GM_ORD	**	
NOISY_NAV_MEAS		*
PREC_STEP_PRED	**	
PRED_STEP		*
PRED_USE	**,*, STATE_VECTOR_ PREDICT_TASK	*, STATE_VECTOR_ PREDICT_TASK
R _PRED_FINAL	*	
R _PRED_INIT		*

*Onorbit/Rendezvous Navigation Sequencer principal function, see section 4.1
**Initialization parameters, see section 4.7

TABLE 4.1.2.2.- REND_NAV_INIT INPUT/OUTPUT.- Continued

Variable Name	Input Source	Output Destination
R _TV	**,*	*,REND_COV_INIT, TARGET_RESET, COV_LAST_RESET
REND_NAV_FLAG	*	*,ONORBIT_REND_NAV_SEQUENCER
REND_NAV_FLAG_LAST		ONORBIT_REND_NAV_SEQUENCER
REND_NAV_INIT_PRED		ONORBIT_REND_NAV_SEQUENCER
TAU_U_A_COAST	**	
TAU_UNMOD_ACC_COV		*
T_PRED_INIT		*, STATE_VECTOR_PREDICT_TASK
T_PRED_FINAL	STATE_VECTOR_PREDICT_TASK	
T_TV	**,*	*, REND_COV_INIT
TV_PREDICT_FAIL		*
VAR_U_A_COAST	**	
VAR_UNMOD_ACC		*
VFLTV_PRED	**	
VMP		*
V _PRED_FINAL	*	
V _PRED_INIT		*

*Onorbit/Rendezvous Navigation Sequencer principal function, see section 4.1
**Initialization parameters, see section 4.7

TABLE 4.1.2.2.- REND_NAV_INIT INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
V_TV	*,**	*,REND_COV_INIT, TARGET_RESET, COV_LAST_RESET

*Onorbit/Rendezvous Navigation Sequencer principal function, see section 4.1

**Initialization parameters, see section 4.7

4.1.2.2.1 Covariance matrix initialization (REND_COV_INIT)

The covariance matrix initialization subfunction shall perform the following tasks upon entering the rendezvous phase, whenever there is an automatic inflight update during the rendezvous phase, or whenever the crew requests a covariance matrix reinitialization or a state vector transfer.

1. Initialize the covariance matrix.

2. Zero the counter (I-CYCLE) for the asynchronous covariance propagation.

3. Set certain flags to OFF so that the rendezvous sensor initialization subfunction will properly execute.

4. Zero measurement Accept/Reject counters used for display purposes.

A. Detailed Requirements. In circumstances in which the filter vehicle position and velocity elements of the on-board filter covariance matrix are to be initialized to UVW values, the following steps shall be performed (in the order indicated):

1. Zero the entire 13 by 13 dimensional covariance matrix

$$E_{1 \text{ to } 13,\ 1 \text{ to } 13} = 0.0$$

2. Test the SHUTTLE_FILTER_FLAG to determine if the Shuttle vehicle or the target vehicle state is being used by the Kalman filter.

   a. If the Shuttle vehicle is the filter vehicle

      (SHUTTLE_FILTER_FLAG = ON)

   (1) Call the UVW parameters initialization subfunction to initialize the position-velocity portion of the covariance matrix with the Shuttle state vector statistics (see section 4.1.2.2.1.1)

      CALL: COVINIT_UVW

      IN LIST: $\underline{R}$_FILT, $\underline{V}$_FILT

   (2) Calculate the last acceleration vector for the Shuttle vehicle for use in the covariance matrix propagation subfunction.

$$\underline{\text{T}}\text{OT\_ACC\_LAST} = \underline{\text{A}}\text{CCEL\_ONORBIT (IGD, IGO, IDRAG, IVENT, ATFL\_CV, }\underline{\text{R}}\text{\_FILT, }\underline{\text{V}}\text{\_FILT, T\_LAST\_FILT)} + \frac{\underline{\text{D}}\text{V\_FILT}}{\text{DT\_FILT}} \qquad \text{F3}$$

F3-This equation shall be protected against division by zero(reference 3.6-3).

where IGD is the degree of the gravitational potential model, IGO is the order of the gravitational potential model, IDRAG is the drag model flag, IVENT is the vent model flag, all set by the state propagation, and ATFL_OV is from I-Load.

(For detailed requirements, see section 4.2.4.1.1)

b. If the target vehicle is the filter vehicle

(SHUTTLE_FILTER_FLAG = OFF)

(1) Call the UVW parameters initialization subfunction to initialize the position-velocity portion of the covariance matrix with the target state vector statistics (see section 4.1.2.2.1.1)

CALL: COVINIT_UVW

IN LIST: $\underline{R}$ _TV, $\underline{V}$ _TV

(2) Calculate the last acceleration vector for the target vehicle for use in the covariance matrix propagation subfunction,

$\underline{G}$ _TV_LAST = $\underline{A}$CCEL_ONORBIT (GM_DEG, GM_ORD, DFL, VFL_TV, ATFL_TV, $\underline{R}$ _TV, $\underline{V}$ _TV, T_TV)

where GM_DEG is the degree of the gravitational potential model, GM_ORD is the order of the gravitational potential model, DFL is the drag model flag, VFL_TV is the vent model flag, and ATFL_ TV is the attitude flag, all for the target vehicle.

(For detailed requirements, see section 4.2.4.1.1)

3. The covariance matrix propagation cycle counter shall be reset

I_CYCLE = 0

4. The covariance reset subfunction is called to store copies of the Shuttle state vector, the target state vector and their time tag, as well as to zero the covariance accumulated delta velocity.

CALL: COV_LAST_RESET

5. Reset the DO_NAV_LAST flags for all sensors so that the sensor bias portions of the covariance matrix will be reconfigured for active sensors

EXECUTE: DO_NAV_LAST_SETUP CODE

DO_COAS_ANGLES_NAV_LAST = OFF

DO_RR_ANGLES_NAV_LAST = OFF

DO_RRDOT_NAV_LAST = OFF

DO_ST_ANGLES_NAV_LAST = OFF

6. Reset the measurement ACCEPT/REJECT counters

(Refer to section 4.1.2.2.1.2)

CALL: DISPLAY_COUNT_INIT

B. Interface Requirements. The input and output variables for this subfunction are defined in table 4.1.2.2.1.

C. Processing Requirements. This subfunction is called by the following subfunctions:

ONORBIT_REND_AUTO_INFLIGHT_UPDATE (section 4.2.3.1)

REL_NAV_DISPLAY_UPDATES (section 4.2.3.2)

REND_NAV_INIT (section 4.1.2.2)

D. Constraints. None

E. Supplementary Information. A suggested implementation in the form of a detailed flowchart can be found in Appendix B under the name:

REND_COV_INIT

TABLE 4.1.2.2.1.- REND_COV_INIT INPUT/OUTPUT

Variable Name	Input Source	Output Destination
ATFL_OV	**	ACCEL_ONORBIT
ATFL_TV	**	ACCEL_ONORBIT
DFL	**	ACCEL_ONORBIT
DT_FILT	ONORBIT_REND_R_V_STATE_PROP, ***	
DV_FILT	ONORBIT_REND_R_V_STATE_PROP, ***	
GM_DEG	**	ACCEL_ONORBIT
GM_ORD	**	ACCEL_ONORBIT
IDRAG	ONORBIT_REND_R_V_STATE_PROP, ***	ACCEL_ONORBIT
IGD	ONORBIT_REND_R_V_STATE_PROP, ***	ACCEL_ONORBIT
IGO	ONORBIT_REND_R_V_STATE_PROP, ***	ACCEL_ONORBIT
IVENT	ONORBIT_REND_R_V_STATE_PROP, ***	ACCEL_ONORBIT
R _FILT	ONORBIT_REND_AUTO_INFLIGHT_UPDATE,ONORBIT_REND_R_V_STATE_PROP,REL_NAV_DISPLAY_UPDATES, ***,REND_NAV_FILTER,*	ACCEL_ONORBIT, COVINIT_UVW
R _TV	ONORBIT_REND_AUTO_INFLIGHT_UPDATE,REND_NAV_INIT,ONORBIT_REND_R_V_STATE_PROP,REL_NAV_DISPLAY_UPDATES,REND_NAV_FILTER,*	ACCEL_ONORBIT, COVINIT_UVW

*Onorbit/Rendezvous principal function, see section 4.2

**Initialization parameters, see section 4.7

***Onorbit/Rendezvous Navigation Sequencer principal function, see section 4.1

TABLE 4.1.2.2.1.- REND_COV_INIT INPUT/OUTPUT (Continued)

Variable Name	Input Source	Output Destination
SHUTTLE_FILTER_FLAG	**	
T_LAST_FILT	ONORBIT_REND_AUTO_INFLIGHT_UPDATE,ONORBIT_REND_R_V_STATE_PROP,***	
V _FILT	ONORBIT_REND_R_V_STATE_PROP,ONORBIT_REND_AUTO_INFLIGHT_UPDATE,REL_NAV_DISPLAY_UPDATES,***, REND_NAV_FILTER,*	ACCEL_ONORBIT, COVINIT_UVW
VFL_TV	**	ACCEL_ONORBIT
V _TV	ONORBIT_REND_R_V_STATE_PROP,REND_NAV_INIT, ONORBIT_REND_AUTO_INFLIGHT_UPDATE,REL_NAV_DISPLAY_UPDATES,REND_NAV_FILTER,*	ACCEL_ONORBIT, COVINIT_UVW
†	ACCEL_ONORBIT	
DO_COAS_ANGLES_NAV_LAST		REND_NAV_SENSOR_INIT, ***
DO_RR_ANGLES_NAV_LAST		REND_NAV_SENSOR_INIT, ***
DO_RRDOT_NAV_LAST		REND_NAV_SENSOR_INIT, ***
DO_ST_ANGLES_NAV_LAST		REND_NAV_SENSOR_INIT, ***
E		REND_BIAS_AND_COV_PROP, REND_NAV_FILTER,*,***

*Onorbit/Rendezvous principal function, see section 4.2
**Initialization parameters, see section 4.7
***Onorbit/Rendezvous Navigation Sequencer principal function, see section 4.1
†Value returned from the function

TABLE 4.1.2.2.1.- REND_COV_INIT_INPUT/OUTPUT (Concluded)

Variable Name	Input Source	Output Destination
G _TV_LAST		REND_BIAS_AND_COV_PROP, ***
I_CYCLE		NAV_ONORBIT_RENDEZVOUS, ***,*
TOT_ACC_LAST		REND_BIAS_AND_COV_PROP, ***
T_TV	ONORBIT_REND_R_V_STATE_PROP,ONORBIT_REND_AUTO_INFLIGHT_UPDATE,REND_NAV_INIT,*	ACCEL_ONORBIT

***Onorbit/Rendezvous Navigation Sequencer principal function, see section 4.1

4.1.2.2.1.1 UVW parameters initialization (COVINIT_UVW)

The UVW parameters initialization subfunction initializes the covariance matrix when rendezvous navigation is initialized, or when an auto-inflight update or REL_NAV display update to the state vectors or covariance matrix is performed.

A. Detailed Requirements. This subfunction has an inlist with the following internal names

INLIST: $\underline{R}$, $\underline{V}$

where $\underline{R}$ is the filter vehicle position vector and $\underline{V}$ is the filter vehicle velocity vector.

This subfunction performs the following steps.

1. Initialize the first six diagonal elements of the covariance with the square of prestored standard deviations.

$$E_{I,I} = \text{SIG\_UPDATE}_I \ \text{SIG\_UPDATE}_I \quad \text{for } I = 1 \text{ to } 6$$

2. Initialize the off-diagonal correlation terms of the covariance matrix in the upper left 6 by 6 portion using prestored correlation coefficients along with the prestored sigmas.

$$E_{1,2} = \text{COV\_COR\_UPDATE}_1 \ \text{SIG\_UPDATE}_1 \ \text{SIG\_UPDATE}_2$$

$$E_{1,4} = \text{COV\_COR\_UPDATE}_2 \ \text{SIG\_UPDATE}_1 \ \text{SIG\_UPDATE}_4$$

$$E_{1,5} = \text{COV\_COR\_UPDATE}_3 \ \text{SIG\_UPDATE}_1 \ \text{SIG\_UPDATE}_5$$

$$E_{2,4} = \text{COV\_COR\_UPDATE}_4 \ \text{SIG\_UPDATE}_2 \ \text{SIG\_UPDATE}_4$$

$$E_{2,5} = \text{COV\_COR\_UPDATE}_5 \ \text{SIG\_UPDATE}_2 \ \text{SIG\_UPDATE}_5$$

$$E_{3,6} = \text{COV\_COR\_UPDATE}_6 \ \text{SIG\_UPDATE}_3 \ \text{SIG\_UPDATE}_6$$

$$E_{4,5} = \text{COV\_COR\_UPDATE}_7 \ \text{SIG\_UPDATE}_4 \ \text{SIG\_UPDATE}_5$$

$$E_{2,1} = E_{1,2}$$

$$E_{5,4} = E_{4,5}$$

3. Call the unmodeled acceleration bias and covariance initialization subfunction (section 4.2.4.1) to initialize the unmodeled acceleration bias states and the unmodeled acceleration bias slots of the covariance matrix.

CALL: U_A_BIAS_AND_COVINIT

INLIST: $\underline{R}$, $\underline{V}$

4. Convert the upper left 6 by 6 portion of the covariance matrix from UVW to mean of '50 coordinates with the following equations

$$E_{1to3,\ 1to3} = \text{M\_UVW\_M50}\ E_{1to3,\ 1to3}\ \text{M\_UVW\_M50}^T$$

$$E_{4to6,\ 4to6} = \text{M\_UVW\_M50}\ E_{4to6,\ 4to6}\ \text{M\_UVW\_M50}^T$$

$$E_{1to3,\ 4to6} = \text{M\_UVW\_M50}\ E_{1to3,\ 4to6}\ \text{M\_UVW\_M50}^T$$

$$E_{4to6,\ 1to3} = (E_{1to3,\ 4to6})^T$$

where the M_UVW_M50 transformation matrix was computed by U_A_BIAS_AND COVINIT.

B. Interface Requirements. The inputs and outputs for this subfunction are given in table 4.1.2.2.1.1.

C. Processing Requirements. This subfunction is called by

REND_COV_INIT (section 4.1.2.2.1)

D. Constraints. None

E. Supplementary Information. A suggested implementation of this subfunction is given by the flowchart COVINIT_UVW in Appendix B.

TABLE 4.1.2.2.1.1.- COVINIT_UVW INPUT/OUTPUT

In list/Outlist		Input Source	Output Destination
Internal Name	External Name		
R	R _FILT	REND_COV_INIT	
V	V _FILT	REND_COV_INIT	
R	R _TV	REND_COV_INIT	
V	V _TV	REND_COV_INIT	

*Onorbit/Rendezvous principal function, see section 4.2
**Initialization parameters, see section 4.7

TABLE 4.1.2.2.1.1.- COVINIT_UVW INPUT/OUTPUT (Concluded)

Variable Name	Input Source	Output Destination
COV_COR_UPDATE	**,***,*	
E		*,***,REND_BIAS_AND_COV_PROP, REND_NAV_FILTER
M_UVW_M50	U_A_BIAS_AND_COVINIT	
R		U_A_BIAS_AND_COVINIT
SIG_UPDATE	**,***,*	
V		U_A_BIAS_AND_COVINIT

*Onorbit/Rendezvous principal function, see section 4.2
**Initialization parameters, see section 4.7
***Onorbit/Rendezvous Navigation Sequencer principal function, see section 4.1

4.1.2.2.1.2 Display parameter initialization (DISPLAY_COUNT_INIT)

The display initialization subfunction shall reset measurement ACCEPT/REJECT counters whenever the covariance initialization subfunction is invoked, whenever exiting a rendezvous navigation phase, or whenever the magnitude of the IMU sensed accelerations exceed a design dependent threshold for measurement incorporation.

A. Detailed Requirements. The following steps shall be performed (in the order indicated):

Zero the ACCEPT/REJECT counters. N_ACCEPT (N_REJECT) is a 4x1 array which counts the total number of accepted (rejected) measurements during the current mark sequence. SEQ_ACCEPT (SEQ_REJECT) counts the number of sequential accepted (rejected) measurements during the current mark sequence.

N_ACCEPT = 0
N_REJECT = 0 (all are 4 x 1
SEQ_ACCEPT = 0 dimensioned arrays)
SEQ_REJECT = 0

B. Interface Requirements. The input and output data are shown in table 4.1.2.2.1.2.

C. Processing Requirements. This subfunction is called by the following subfunctions:

REND_COV_INIT	(section 4.1.2.2.1)
REND_NAV_EXIT	(section 4.1.1.1)
REND_BIAS_AND_COV_PROP	(section 4.2.5)

D. Constraints. None

E. Supplementary Information. A suggested implementation in the form of a detailed flowchart may be found in Appendix B under the name of DISPLAY_COUNT_INIT.

TABLE 4.1.2.2.1.2.- DISPLAY_COUNT_INIT INPUT/OUTPUT

Variable Name	Input Source	Output Destination
N_ACCEPT		MEAS_PROCESSING_STATISTICS_REND,*,***
N_REJECT		MEAS_PROCESSING_STATISTICS_REND,*,***
SEQ_ACCEPT		MEAS_PROCESSING_STATISTICS_REND,*,***
SEQ_REJECT		MEAS_PROCESSING_STATISTICS_REND,*,***

*Onorbit/Rendezvous principal function, see section 4.2
***Onorbit/Rendezvous Navigation Sequencer principal function, see section 4.1

c-2

4.1.2.2.1.3 Covariance matrix parameters reset (COV_LAST_RESET)

The covariance matrix parameters reset subfunction saves certain navigation parameters at the end of a covariance matrix propagation subcycle for use in the next subcycle. These parameters include the Orbiter and target position and velocity vectors as well as the current time. Also, a variable used for summing the IMU sensed delta velocities over the subcycle interval is zeroed for use in the next subcycle.

A. Detailed Requirements. The following steps shall be performed:

Store the Orbiter position and velocity vectors (R _FILT, V _FILT), the target position and velocity vectors (R _TV, V _TV), last filter time (T_LAST_FILT), and zero the delta velocity accumulator (DV_COV).

R _LAST = R _FILT

V _LAST = V _FILT

R _TV_LAST = R _TV

V _TV_LAST = V _TV

DV_COV = 0.

T_COV_LAST = T_LAST_FILT

B. Interface Requirements. The input and output data are shown in table 4.1.2.2.1.3.

C. Processing Requirements. This subfunction is called by the following subfunctions:

NAV_ONORBIT_RENDEZVOUS	(section 4.2.1)
REND_COV_INIT	(section 4.1.2.2.1)

D. Constraints. None

E. Supplementary Information. A suggested implementation in the form of a detailed flowchart may be found in Appendix B under the name COV_LAST_RESET.

TABLE 4.1.2.2.1.3.- COV_LAST_RESET INPUT/OUTPUT

Variable Name	Input Source	Output Destination
R _FILT	ONORBIT_REND_AUTO_INFLIGHT_UPDATE, REND_NAV_FILTER, ONORBIT_REND_R_V_STATE_PROP, REL_NAV_DISPLAY_UPDATES,*,***	
R _TV	ONORBIT_REND_AUTO_INFLIGHT_UPDATE, REND_NAV_FILTER, ONORBIT_REND_R_V_STATE_PROP, REL_NAV_DISPLAY_UPDATES, REND_NAV_INIT	
V _FILT	ONORBIT_REND_AUTO_INFLIGHT_UPDATE, REND_NAV_FILTER, ONORBIT_REND_R_V_STATE_PROP, REL_NAV_DISPLAY_UPDATES,*,***	
V _TV	ONORBIT_REND_AUTO_INFLIGHT_UPDATE, REND_NAV_FILTER, ONORBIT_REND_R_V_STATE_PROP, REL_NAV_DISPLAY_UPDATES, REND_NAV_INIT	
DV_COV		NAV_ONORBIT_RENDEZVOUS, REND_BIAS_AND_COV_PROP,*,***
R _LAST		REND_BIAS_AND_COV_PROP, REND_NAV_INTERP,*
R _TV_LAST		REND_BIAS_AND_COV_PROP, REND_NAV_INTERP,*
T_LAST_FILT	ONORBIT_REND_R_V_STATE_PROP,*	

*Onorbit/Rendezvous Navigation Sequencer principal function, see section 4.1

**Initialization parameters, see section 4.7

***Onorbit/Rendezvous Navigation principal function, see section 4.2

TABLE 4.1.2.2.1.3.- COV_LAST_RESET INPUT/OUTPUT (Concluded)

Variable Name	Input Source	Output Destination
T_COV_LAST		REND_BIAS_AND_COV_PROP,*
$\underline{V}$ _LAST		REND_BIAS_AND_COV_PROP, REND_NAV_INTERP,*
$\underline{V}$ _TV_LAST		REND_BIAS_AND_COV_PROP, REND_NAV_INTERP,*

*Onorbit/Rendezvous Navigation Sequencer principal function, see section 4.1

4.1.2.2.2 Target state vector reset (TARGET_RESET)

Upon entering the rendezvous navigation phase this subfunction is exercised to store the current target state vector (position and velocity) for user parameter processing. Thereafter, while still in the rendezvous navigation phase this subfunction is exercised at the end of each navigation cycle.

A. Detailed Requirements. This subfunction shall reset the user parameter processing (section 4.5) copy of navigated target position and velocity:

$\underline{R}$ _TV_RESET = $\underline{R}$ _TV

$\underline{V}$ _TV_RESET = $\underline{V}$ _TV

B. Interface Requirements. The input and output parameters for this subfunction are found in table 4.1.2.2.2.

C. Processing Requirements. This subfunction is called by the following subfunctions:

NAV_ONORBIT_RENDEZVOUS	(section 4.2.1)
REND_NAV_INIT	(section 4.1.2.2)

D. Constraints. None

E. Supplementary Information. A suggested implementation of this subfunction is found in Appendix B, TARGET_RESET.

TABLE 4.1.2.2.2.- TARGET_RESET INPUT/OUTPUT

Variable Name	Input Source	Output Destination
R _TV	ONORBIT_REND_R_V_ STATE_PROP,REND_NAV_ FILTER,REND_NAV_INIT	
V _TV	**,ONORBIT_REND_R_V_ STATE_PROP,REND_NAV_ FILTER,REND_NAV_INIT	
R _TV_RESET		*,***
V _TV_RESET		*,***

*Onorbit/Rendezvous principal function, see section 4.2
**Initialization parameters, see section 4.7
***Onorbit/Rendezvous Navigation Sequencer principal function, see section 4.1

4.2 ONORBIT/RENDEZVOUS NAVIGATION PRINCIPAL FUNCTION

The Onorbit/Rendezvous Navigation principal function is the name given to the collection of all subfunctions whose major task is to supply users of this principal function with accurate estimates of the Orbiter and target state vectors. This section presents an overview of this principal function.

The Onorbit/Rendezvous Navigation principal function contains nine primary subfunctions.

- Navigation Control (section 4.2.1)
- Navigation Data Snap (section 4.2.2)
- State and Covariance Matrix Updates (section 4.2.3)
- Position and Velocity State Propagation (section 4.2.4)
- Unmodeled Acceleration State and Covariance Matrix Propagation (section 4.2.5)
- Sensor Measurement Selection (section 4.2.6)
- Sensor Measurement Initialization (section 4.2.7)
- State and Covariance Measurement Incorporation (section 4.2.8)
- Measurement Processing Statistics (section 4.2.9)

This Onorbit/Rendezvous Navigation design makes use of a Kalman filter with a 13 component state vector in order to produce an accurate estimate of position and velocity for the Orbiter and target. The state vector is composed of the following components.

- Orbiter or target position and velocity in the mean of 1950 coordinate system - six elements
- Orbiter or target unmodeled acceleration biases in mean of 1950 coordinates - three elements
- Systematic sensor biases - four elements

A flag (SHUTTLE_FILTER_FLAG) with a premission specified value determines whether the Orbiter or the target position and velocity are to be updated with the Kalman filter. This same flag also indicates which vehicle is to be associated with the unmodeled acceleration bias states.

The Navigation Control subfunction supplies the navigation trunk logic, which sequences the other eight primary subfunctions in the proper order and at the proper rates. The covariance matrix propagation, sensor selection, sensor initialization, measurement incorporation and measurement processing statistics subfunctions can be performed at a rate slower than the data snap, state propagation and state and covariance matrix update subfunctions. This subrate is determined by the setting of the DO_FLTR_SLOW_RATE switch on the onboard REL NAV display. The switch causes the value of a cycle count, N_CYCLE, to be set to 1 of 2 I-Load values. When the cycle counter, I_CYCLE, reaches the value N_CYCLE, the slow rate subfunctions are executed. This subrate capability is basically intended to be used when it is undesirable to perform the computationally complex covariance matrix propagation and measurement incorporation at the state vector propagation rate. The Navigation Control subfunction is scheduled by the Onorbit/Rendezvous Navigation Sequencer principal function at the proper rate for the active navigation phase and operational sequence.

All external flags input to NAV are snapped at the same time and stored into locations for use by NAV. The Navigation Data Snap subfunction copies frequently changing measurement data buffers into static memory locations local to the navigation processing. During the rendezvous navigation phase, the IMU, the rendezvous radar, the star tracker, and the Crew Optical Alinement Sight (COAS) sensors are available to the navigation. During non-rendezvous, only IMU data is available.

The State and Covariance Matrix Updates subfunction implements ground updates to the Orbiter or target state vector via uplink and performs state vector transfers and covariance matrix reinitialization when requested by the crew on the REL NAV display. All of these types of updates require reinitialization of the covariance matrix. For ground updates the prediction task must be invoked to bring the Shuttle or target state vector to current time. Due to the priority of the Precise Predictor principal function in the onboard computer, the prediction can proceed only after navigation has completed a cycle. The prediction can possibly require more than one navigation interval to complete the prediction. The update to the navigation state vectors will not take place until the prediction has completed.

The Position and Velocity State Propagation subfunction must maintain a current estimate of the Orbiter position and velocity during rendezvous and non-rendezvous navigation phases and a current estimate of the target position and velocity during the rendezvous navigation phase only. The subfunction will provide a unit vector from the center of the Earth in the direction of the Sun for use by the Universal Pointing Processor principal function at a frequency of once every navigation cycle. When the powered flight navigation phase is active, IMU sensed delta velocities and a model of the Earth's gravitational acceleration are used to propagate the Orbiter position and velocity to current time. When in the coasting flight navigation phase, models of the Earth's gravitational acceleration, aerodynamic drag, and venting or uncoupled RCS thrusting acceleration are used to propagate the Orbiter. The target propagation uses models of the Earth's gravitational acceleration and aerodynamic drag during either powered flight or coasting flight.

The task of propagating the statistics of the state vector to current time as well as the propagation of the unmodeled acceleration bias states is performed by the Unmodeled Acceleration State and Covariance Matrix Propagation subfunction. The unmodeled acceleration bias state elements are propagated exponentially (the sensor bias state elements are assumed constant over the propagation interval) so this subfunction will determine the appropriate time constant and variances used for propagation dependent on whether or not IMU sensed delta velocities were used to propagate the position and velocity vectors for the Orbiter.

The unmodeled acceleration bias states actually represent different quantities depending on whether the IMU sensed delta velocities were used to propagate the state or not. In the former case the unmodeled acceleration biases represent the errors in the IMU accelerometers. In the latter case they represent unmodeled body forces such as drag, venting and RCS uncoupled thrusting. In each case, a different time constant and variance are used for the exponentially correlated random variable statistics.

A 13 by 13 covariance matrix represents the statistics of the 13 element state vector. The position and velocity statistics are propagated using an analytic partial called the mean conic partial transition matrix. The unmodeled acceleration bias statistics and the sensor bias statistics are propagated as exponentially correlated random variables. A model of platform drift supplies state noise whenever the IMU sensed delta velocities are used to propagate the state.

The Sensor Measurement Selection subfunction must select a measurement set for processing by the Kalman filter. The following measurements are available:

- Rendezvous radar range and range rate
- Rendezvous radar shaft and trunnion angles
- Startracker horizontal and vertical angles
- Crew Optical Alinement Sight (COAS) horizontal and vertical angles

Up to four measurements can be chosen for processing - rendezvous radar range and range rate along with one pair of angles selected by the crew on the REL_NAV display. When a measurement type is selected for the first time, statistical parameters associated with the newly selected measurement type must be initialized for use in the Kalman estimation of systematic sensor biases. This task is performed by the Navigation Sensor Initialization subfunction.

The State and Covariance Measurement Incorporation subfunction performs the final step in the Kalman processing, namely, the incorporation of the selected measurements. The covariance matrix is updated. The position and velocity vectors of the vehicle state selected for Kalman processing (determined by SHUTTLE_FILTER_FLAG) are updated. The sensor biases are updated. A flag from the ILOAD (UNMOD_ACC_UPDATE_FLAG) determines whether the unmodeled acceleration states are to be updated or to be left zero, leaving only their statistical influence on the Kalman processing. Measurements can be incorporated into the state vector only when all the following criteria are satisfied.

- Rendezvous navigation phase is active (REND_NAV_FLAG = ON)
- The measurement type is selected

- The measurement data is valid
- The IMU sensed acceleration is below a premission specified threshold (MEAS_THRESHOLD)
- The covariance matrix has been propagated on the current navigation cycle
- The estimated distance between the Orbiter and the target is greater than a premission specified distance
- The measurement is not in the INHIBIT mode as selected by the crew on the REL NAV display
- The measurement does not fail to pass the edit criteria which rejects measurements which are not commensurate with the estimated measurement
- When in Major Mode 202 the MEAS_ENABLE switch is set to ON by the crew

After Kalman processing has been completed, the Measurement Processing Statistics subfunction shall compute parameters to be displayed on the REL NAV display. The parameters calculated are the measurement residuals and residual ratios, the mark histories, and the edit status.

The only direct user of the onorbit/rendezvous navigated state vector is the User Parameter Processing principal function. This principal function integrates the state vector using a high rate, less precise propagator (average-g) then that used by the navigation. The user parameter processing state vector is updated with the navigation state vector at the navigation rate. Other principal functions which require state vectors get them from the User Parameter Processing principal function at the higher rate.

In table 4.2 (the Onorbit/Rendezvous Navigation Principal Function Inputs/Outputs), there are some parameters which are being output to telemetry (TLM) which were not set by NAV. For these situations, the local source is listed as "NONE".

TABLE 4.2.- ONCRBIT/RENDEZVOUS NAVIGATION PRINCIPAL FUNCTION INPUT/OUTPUT

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
ALT_SS			TLM	ACCEL_ONORBIT
ANGLES_AIF	REL_NAV display	NAV_ ONORBIT_ RENDEZVOUS		
ANGLES_AIF_ DISPLAY			REL_NAV display, TLM	REND_SENSOR_ SELECT
ANGLES_ENABLE_ DISPLAY			REL_NAV display, TLM	REND_NAV_ SENSOR_INIT
ATMP			Onorbit Predictor, TLM	ONORBIT_REND_ AUTO_INFLIGHT _UPDATE
AXN	Star Tracker SOP	Sensor Data Snap		
COAS_DATA_GOOD			TLM	NAV_ONORBIT_ RENDEZVOUS
COAS_ENABLE			TLM	none
COAS_HORIZ	Star Tracker SOP	Sensor Data Snap		
COAS_ID			TLM	NAV_ONORBIT_ RENDEZVOUS
COAS_MARK_NUM			TLM	NAV_ONORBIT_ RENDEZVOUS
COAS_VERT	Star Tracker SOP	Sensor Data Snap		
CONT_ACC			TLM	ONORBIT_REND_ R_V_STATE_ PROP
COV_ACCEL_UVW_ INIT	Onorbit/Rendezvous Navigation Sequencer	U_A_BIAS_ AND_COVINIT		

TABLE 4.2.- ONORBIT/RENDEZVOUS NAVIGATION PRINCIPAL FUNCTION INPUT/OUTPUT (Continued)

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
COV_COR_UPDATE	Uplink Processor	COVINIT_UVW		
COV_PWRD_FLT	Onorbit/Rendezvous Navigation Sequencer	REND_BIAS_AND_COV_PROP		
COV_PWRD_FLT_LAST	Onorbit/Rendezvous Navigation Sequencer	REND_BIAS_AND_COV_PROP		
CURR_ORB_MASS	Onorbit/Rendezvous Navigation Sequencer, Onorbit Guidance	NAV_ONORBIT_RENDEZVOUS		
DATA_GOOD	Startracker SOP	Sensor Data Snap		
DID_COVAR_REINIT			REL_NAV display, TLM	REL_NAV_DISPLAY_UPDATES
DID_ORB_TO_TGT			REL_NAV display, TLM	REL_NAV_DISPLAY_UPDATES
DID_TGT_TO_ORB			REL_NAV display, TLM	REL_NAV_DISPLAY_UPDATES
DISP_DELQ			REL_NAV display, TLM	MEAS_PROCESSING_STATISTICS_REND
DISP_EDIT			REL_NAV display, GN&C Communication Interface, TLM	MEAS_PROCESSING_STATISTICS_REND

TABLE 4.2.- ONORBIT/RENDEZVOUS NAVIGATION PRINCIPAL FUNCTION INPUT/OUTPUT (Continued)

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
DISP_SIG			REL_NAV display, TLM	MEAS_PROCESSING_STATISTICS_REND
DMP			Onorbit Predictor	ONORBIT_REND_AUTO_INFLIGHT_UPDATE
DO_COAS_ANGLES_NAV			TLM	REND_SENSOR_SELECT
DO_COAS_ANGLES_NAV_LAST	Onorbit/Rendezvous Navigation Sequencer	REND_NAV_SENSOR_INIT		
DO_COVAR_REINIT	REL_NAV display	NAV_ONORBIT_RENDEZVOUS		
DO_FLTR_SLOW_RATE	REL_NAV display	NAV_ONORBIT_RENDEZVOUS		
DOING_FLTR_SLOW_RATE			REL NAV display, TLM	NAV_ONORBIT_RENDEZVOUS
DOING_REND_NAV	Onorbit/Rendezvous Navigation Sequencer	ONORBIT_REND_R_V_STATE_PROP, ONORBIT_REND_AUTO_INFLIGHT_UPDATE, NAV_ONORBIT_RENDEZVOUS		
DOING_MEAS_ENABLE			REL_NAV display, TLM	REND_SENSOR_SELECT

TABLE 4.2.- ONORBIT/RENDEZVOUS NAVIGATION PRINCIPAL FUNCTION INPUT/OUTPUT (Continued)

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
DO_ORB_TO_TGT	REL_NAV display	NAV_ONORBIT_RENDEZVOUS		
DO_OV_UPLINK	Uplink Processor	NAV_ONORBIT_RENDEZVOUS	Uplink Processor, TLM	ONORBIT_REND_AUTO_INFLIGHT_UPDATE
DO_RR_ANGLES_NAV			TLM	REND_SENSOR_SELECT
DO_RR_ANGLES_NAV_LAST	Onorbit/Rendezvous Navigation Sequencer	REND_NAV_SENSOR_INIT		
DO_RRDOT_NAV_LAST	Onorbit/Rendezvous Navigation Sequencer	REND_NAV_SENSOR_INIT		
DO_ST_ANGLES_NAV			TLM	REND_SENSOR_SELECT
DO_ST_ANGLES_NAV_LAST	Onorbit/Rendezvous Navigation Sequencer	REND_NAV_SENSOR_INIT		
DO_TV_UPLINK	Uplink Processor	NAV_ONORBIT_RENDEZVOUS	Uplink Processor, TLM	ONORBIT_REND_AUTO_INFLIGHT_UPDATE
DO_TGT_TO_ORB	REL_NAV display	NAV_ONORBIT_RENDEZVOUS		
D_ _SS			TLM	ACCEL_ONORBIT
DT_COV			TLM	REND_BIAS_AND_COV_PROP

TABLE 4.2.- ONORBIT/RENDEZVOUS NAVIGATION PRINCIPAL FUNCTION INPUT/OUTPUT (Continued)

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
DV_COV	Onorbit/Rendezvous Navigation Sequencer	NAV_ONORBIT_RENDEZVOUS, REND_BIAS_AND_COV_PROP, REND_NAV_INTERP	TLM	NAV_ONORBIT_RENDEZVOUS, COV_LAST_RESET
DT_FILT			Onorbit/Rendezvous Navigation Sequencer	ONORBIT_REND_R_V_STATE_PROP
DV_FILT			Onorbit/Rendezvous Navigation Sequencer	ONORBIT_REND_R_V_STATE_PROP
E	Onorbit/Rendezvous Navigation Sequencer	REND_BIAS_AND_COV_PROP	TLM	REND_BIAS_AND_COV_PROP, REND_NAV_FILTER, COVINIT_UVW, U_A_BIAS_AND_COVINIT, SETUP, REND_COV_INIT
FILT_UPDATE			User Parameter Processing	SHUTTLE_RESET
GMDP			Onorbit Predictor	ONORBIT_REND_AUTO_INFLIGHT_UPDATE
GMOP			Onorbit Predictor	ONORBIT_REND_AUTO_INFLIGHT_UPDATE
G _NEW			TLM	ONORBIT_REND_R_V_STATE_PROP
G _TV			TLM	ONORBIT_REND_R_V_STATE_PROP

TABLE 4.2.- ONORBIT/RENDEZVOUS NAVIGATION PRINCIPAL FUNCTION INPUT/OUTPUT (Continued)

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
G_TV_LAST	Onorbit/Rendezvous Navigation Sequencer	REND_BIAS_AND_COV_PROP		
H_NAV	Startracker SOP	Sensor Data Snap		
I_CYCLE	Onorbit/Rendezvous Navigation Sequencer	NAV_ONORBIT_RENDEZVOUS	TLM	NAV_ONORBIT_RENDEZVOUS, REND_COV_INIT
IDRAG			Onorbit/Rendezvous Navigation Sequencer	ONORBIT_REND_R_V_STATE_PROP
IMU_NAV_ACCEL_THRESH	IMU align display	ONORBIT_REND_R_V_STATE_PROP	TLM	NONE
IVENT			Onorbit/Rendezvous Navigation Sequencer	ONORBIT_REND_R_V_STATE_PROP
IGD			Onorbit/Rendezvous Navigation Sequencer	ONORBIT_REND_R_V_STATE_PROP
IGO			Onorbit/Rendezvous Navigation Sequencer	ONORBIT_REND_R_V_STATE_PROP
KFACTOR	Uplink Processor	ACCEL_ONORBIT		
MEAS_ENABLE	REL_NAV display, Onorbit/Rendezvous Navigation Sequencer	NAV_ONORBIT_RENDEZVOUS		
M_M50_TO_BODY_COAS			TLM	NAV_ONORBIT_RENDEZVOUS

TABLE 4.2.- ONORBIT/RENDEZVOUS NAVIGATION PRINCIPAL FUNCTION INPUT/OUTPUT (Continued)

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
M_M50_TO_ST			TLM	NAV_ONORBIT_ RENDEZVOUS
MM_202	MSC	NAV_ONORBIT _RENDEZVOUS		
N _ACCEPT	Onorbit/Rendezvous Navigation Sequencer	MEAS_ PROCESSING_ STATISTICS_ REND	TLM, REL_NAV display	MEAS_ PROCESSING_ STATISTICS_ REND, DISPLAY _COUNT_INIT, SETUP
NAV_MEAS			TLM	NAV_ONORBIT_ RENDEZVOUS
NAV_ANGLES_AIF			TLM	NAV_ONORBIT_ RENDEZVOUS
NAV_CURR_ORB_ MASS			TLM	NAV_ONORBIT_ RENDEZVOUS
NAV_DO_COVAR_ REINIT			TLM	NAV_ONORBIT_ RENDEZVOUS
NAV_DO_FLTR_ SLOW_RATE			TLM	NAV_ONORBIT_ RENDEZVOUS
NAV_DO_ORB_TO_ TGT			TLM	NAV_ONORBIT_ RENDEZVOUS
NAV_DO_OV_ UPLINK			TLM	NAV_ONORBIT_ RENDEZVOUS
NAV_DO_TGT_TO_ ORB			TLM	NAV_ONORBIT_ RENDEZVOUS
NAV_DO_TV_ UPLINK			TLM	NAV_ONORBIT_ RENDEZVOUS
NAV_MEAS_ ENABLE			TLM	NAV_ONORBIT_ RENDEZVOUS

TABLE 4.2.- ONORBIT/RENDEZVOUS NAVIGATION PRINCIPAL FUNCTION INPUT/OUTPUT (Continued)

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
NAV_MM_202			TLM	NAV_ONORBIT_ RENDEZVOUS
NAV_PWRD_FLT_ NAV			TLM	NAV_ONORBIT_ RENDEZVOUS
NAV_RANGE_AIF			TLM	NAV_ONORBIT_ RENDEZVOUS
NAV_RDOT_AIF			TLM	NAV_ONORBIT_ RENDEZVOUS
NAV_RR_ANGLES_ ENABLE			TLM	NAV_ONORBIT_ RENDEZVOUS
NAV_ST_ENABLE			TLM	NAV_ONORBIT_ RENDEZVOUS
NAV_SIGHT	Startracker SOP	Sensor Data Snap		
NAV_TARGET	Startracker SOP	Sensor Data Snap		
NOISY_NAV_MEAS	Onorbit/Rendezvous Navigation Sequencer	REND_BIAS_ AND_COV_ PROP		
N _REJECT	Onorbit/Rendezvous Navigation Sequencer	MEAS_ PROCESSING_ STATISTICS_ REND	TLM, REL_NAV display	MEAS_ PROCESSING_ STATISTICS_ REND, DISPLAY_ COUNT_INIT, SETUP
OP_CODE			TLM	none
OV_PREDICT_ FAIL			TLM	ONORBIT_REND_ AUTO_INFLIGHT _UPDATE

TABLE 4.2.- ONORBIT/RENDEZVOUS NAVIGATION PRINCIPAL FUNCTION INPUT/OUTPUT (Continued)

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
PRED_ORB_AREA			Onorbit Predictor, TLM	ONORBIT_REND_AUTO_INFLIGHT_UPDATE
PRED_ORB_CD			Onorbit Predictor, TLM	ONORBIT_REND_AUTO_INFLIGHT_UPDATE
PRED_ORB_MASS			Onorbit Predictor, TLM	ONORBIT_REND_AUTO_INFLIGHT_UPDATE
PRED_STEP			Onorbit Predictor, TLM	ONORBIT_REND_AUTO_INFLIGHT_UPDATE
PRED_USE	Onorbit/Rendezvous Navigation Sequencer	ONORBIT_REND_AUTO_INFLIGHT_UPDATE	Onorbit/Rendezvous Navigation Sequencer, TLM	ONORBIT_REND_AUTO_INFLIGHT_UPDATE, STATE_VECTOR_PREDICT_TASK
PWRD_FLT_NAV	REL_NAV display, Onorbit/Rendezvous Navigation Sequencer, FCS/DED DISP C/O Display	NAV_ONORBIT_RENDEZVOUS	TLM	none
Q_BOD_M50	Onorbit Attitude Processing	IMU Data Snap		
Q_COAS_HORIZ			TLM	NAV_ONORBIT_RENDEZVOUS
Q_COAS_VERT			TLM	NAV_ONORBIT_RENDEZVOUS
Q_M50BODY_IMU			TLM	NAV_ONORBIT_RENDEZVOUS
Q_M50BODY_RR	Rendezvous Radar SOP	Sensor Data Snap	TLM	NAV_ONORBIT_RENDEZVOUS

TABLE 4.2.- ONORBIT/RENDEZVOUS NAVIGATION PRINCIPAL FUNCTION INPUT/OUTPUT (Continued)

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
Q_ST_HORIZ			TLM	NAV_ONORBIT_RENDEZVOUS
Q_ST_VERT			TLM	NAV_ONORBIT_RENDEZVOUS
Q_RR_RNG			TLM	NAV_ONORBIT_RENDEZVOUS
Q_RR_RNG_DOT			TLM	NAV_ONORBIT_RENDEZVOUS
Q_RR_SHFT			TLM	NAV_ONORBIT_RENDEZVOUS
Q_RR_TRUN			TLM	NAV_ONORBIT_RENDEZVOUS
RANGE_AIF	REL_NAV display	NAV_ONORBIT_RENDEZVOUS		
RANGE_AIF_DISPLAY			REL_NAV display, TLM	REND_SENSOR_SELECT
RDOT_AIF	REL_NAV display	NAV_ONORBIT_RENDEZVOUS		
RDOT_AIF_DISPLAY			REL_NAV display, TLM	REND_SENSOR_SELECT
RDOT_DATA_GOOD			TLM	NAV_ONORBIT_RENDEZVOUS
R _FILT	Onorbit/Rendezvous Navigation Sequencer	ONORBIT_REND_R_V_STATE_PROP, COV_LAST_RESET, REND_COV_INIT, NAV_ONORBIT_RENDEZVOUS	Onorbit/Rendezvous Navigation Sequencer	ONORBIT_REND_R_V_STATE_PROP, REND_NAV_FILTER
R _FILT_TLM			TLM	NAV_ONORBIT_RENDEZVOUS

TABLE 4.2.- ONORBIT/RENDEZVOUS NAVIGATION PRINCIPAL FUNCTION INPUT/OUTPUT (Continued)

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
R _GND	Uplink Processor	ONORBIT_ REND_AUTO_ INFLIGHT_ UPDATE		
R _LAST	Onorbit/Rendezvous Navigation Sequencer	REND_BIAS_ AND_COV_ PROP, REND_NAV_ INTERP		
RNG_DATA_GOOD			TLM	NAV_ONORBIT_ RENDEZVOUS
R _PRED_FINAL	Onorbit Predictor	ONORBIT_ REND_AUTO_ INFLIGHT_ UPDATE,STATE _VECTOR_ PREDICT_TASK	TLM	none
R _PRED_INIT			Onorbit Predictor, TLM	ONORBIT_REND_ AUTO_INFLIGHT _UPDATE,STATE _VECTOR_ PREDICT_TASK
RR_ANG_DG	Rendezvous Radar SOP	Sensor Data Snap		
RR_ANGLE_DATA_ GOOD			TLM	NAV_ONORBIT_ RENDEZVOUS
RR_ANGLE_MARK_ NUM			TLM	NAV_ONORBIT_ RENDEZVOUS
RR_ANGLES_ ENABLE	REL_NAV display	NAV_ONORBIT_ RENDEZVOUS		
RRDOT_MARK_NUM			TLM	NAV_ONORBIT_ RENDEZVOUS

TABLE 4.2.- ONORBIT/RENDEZVOUS NAVIGATION PRINCIPAL FUNCTION INPUT/OUTPUT (Continued)

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
R _RESET			User Parameter Processing	SHUTTLE_RESET
RR_PITCHO	Rendezvous Radar SOP	Sensor Data Snap		
RR_RANGEO	Rendezvous Radar SOP	Sensor Data Snap		
RR_RNG_DG	Rendezvous Radar SOP	Sensor Data Snap		
RR_RNGR_DG	Rendezvous Radar SOP	Sensor Data Snap		
RR_RNGRO	Rendezvous Radar SOP	Sensor Data Snap		
RR_ROLLO	Rendezvous Radar SOP	Sensor Data Snap		
RR_SELF_TEST	Rendezvous Radar SOP	Sensor Data Snap		
RR_TIM	Rendezvous Radar SOP	Sensor Data Snap		
R _TV	Onorbit/Rendezvous Navigation Sequencer	ONORBIT_REND_R_V_STATE_PROP, REL_NAV_DISPLAY_UPDATES, COV_LAST_RESET,REND_COV_INIT, NAV_ONORBIT_RENDEZVOUS	Onorbit/Rendezvous Navigation Sequencer	ONORBIT_REND_R_V_STATE_PROP,REND_NAV_FILTER
R _TV_CND	Uplink Processor	ONORBIT_REND_AUTO_INFLIGHT_UPDATE		

TABLE 4.2.- ONORBIT/RENDEZVOUS NAVIGATION PRINCIPAL FUNCTION INPUT/OUTPUT (Continued)

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
R _TV_LAST	Onorbit/Rendezvous Navigation Sequencer	REND_BIAS_AND_COV_PROP, REND_NAV_INTERP		
R _TV_RESET			User Parameter Processing	TARGET_RESET
R _TV_TLM			TLM	NAV_ONORBIT_RENDEZVOUS
SENSOR_BIAS_TLM			TLM	NAV_ONORBIT_RENDEZVOUS
SENSOR_EDIT			TLM	REND_SENSOR_SELECT, RRDOT_NAV, RR_ANGLE_NAV, ANGLE_NAV
SELF_TEST_FLAG			TLM	NAV_ONORBIT_RENDEZVOUS
SEQ_ACCEPT	Onorbit/Rendezvous Navigation Sequencer	MEAS_PROCESSING_STATISTICS_REND	TLM	SETUP, MEAS_PROCESSING_STATISTICS_REND, DISPLAY_COUNT_INIT
SEQ_REJECT	Onorbit/Rendezvous Navigation Sequencer	MEAS_PROCESSING_STATISTICS_REND	TLM	SETUP, MEAS_PROCESSING_STATISTICS_REND, DISPLAY_COUNT_INIT

TABLE 4.2.- ONORBIT/RENDEZVOUS NAVIGATION PRINCIPAL FUNCTION INPUT/OUTPUT (Continued)

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
SHUTTLE_FILTER_FLAG			TLM	none
SIG_UPDATE	Uplink Processor	COVINIT_UVW		
SQR_EMU	Onorbit/Rendezvous Navigation Sequencer	MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6, ONORBIT_SV_INTERP		
ST_DATA_GOOD			TLM	NAV_ONORBIT_RENDEZVOUS
ST_ENABLE	REL_NAV display	NAV_ONORBIT_RENDEZVOUS		
ST_MARK_NUM			TLM	NAV_ONORBIT_RENDEZVOUS
TAU_UNMOD_ACC_COV	Onorbit/Rendezvous Navigation Sequencer	REND_BIAS_AND_COV_PROP		
T_COAS			TLM	NAV_ONORBIT_RENDEZVOUS
T_COV_LAST	Onorbit/Rendezvous Navigation Sequencer	REND_BIAS_AND_COV_PROP		
T_CURRENT_FILT			TLM	NAV_ONORBIT_RENDEZVOUS
TFOFF	Uplink Processor	ACCEL_ONORBIT	TLM	none
TFON	Uplink Processor	ACCEL_ONORBIT	TLM	none

TABLE 4.2.- ONORBIT/RENDEZVOUS NAVIGATION PRINCIPAL FUNCTION INPUT/OUTPUT (Continued)

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
T_GND	Uplink Processor	ONORBIT_ REND_AUTO_ INFLIGHT_ UPDATE		
TIME	Startracker SOP	Sensor Data Snap		
T_IMUS_GA	IMU Inertial Processing	IMU Data Snap		
T_LAST_FILT	Onorbit/Rendezvous Navigation Sequencer	ONORBIT_ REND_R_V_ STATE_PROP, NAV_ONORBIT _RENDEZVOUS	Onorbit/Rendezvous Navigation Sequencer	ONORBIT_REND_ R_V_STATE_ PROP
T_LAST_FILT_ TLM			TLM	NAV_ONORBIT_ RENDEZVOUS
T_M50_BODY	Startracker SOP	Sensor Data Snap		
T_M50_ST	Startracker SOP	Sensor Data Snap		
T_ORB_STATE_ UPDATE			TLM	ONORBIT_REND_ AUTO_INFLIGHT _UPDATE
TOT_ACC			TLM	REND_BIAS_AND _COV_PROP
TOT_ACC_LAST	Onorbit/Rendezvous Navigation Sequencer	REND_BIAS_ AND_COV_ PROP		
T_PRED_FINAL			Onorbit Predictor, TLM	STATE_VECTOR_ PREDICT_TASK

TABLE 4.2.- ONORBIT/RENDEZVOUS NAVIGATION PRINCIPAL FUNCTION INPUT/OUTPUT (Continued)

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
T_PRED_INIT			Onorbit Predictor, TLM	ONORBIT_REND_AUTO_INFLIGHT_UPDATE,STATE_VECTOR_PREDICT_TASK
T_REND_RADAR			TLM	NAV_ONORBIT_RENDEZVOUS
T_RESET	Onorbit/Rendezvous Navigation Sequencer	STATE_VECTOR_PREDICT_TASK	User Parameter Processing, Onorbit/Rendezvous Navigation Sequencer	SHUTTLE_RESET
TRG_TRK_MODE			TLM	NAV_ONORBIT_RENDEZVOUS
T_STAR_TRACKER			TLM	NAV_ONORBIT_RENDEZVOUS
T_TARLOS	Startracker SOP	Sensor Data Snap		
T_TV	Onorbit/Rendezvous Navigation Sequencer	REND_COV_INIT	TLM, Onorbit/Rendezvous Navigation Sequencer	ONORBIT_REND_R_V_STATE_PROP, ONORBIT_REND_AUTO_INFLIGHT_UPDATE (TLM only)
T_TV_GND	Uplink Processor	ONORBIT_REND_AUTO_INFLIGHT_UPDATE		
T_TV_STATE_UPDATE			TLM	ONORBIT_REND_AUTO_INFLIGHT_UPDATE

TABLE 4.2.- ONORBIT/RENDEZVOUS NAVIGATION PRINCIPAL FUNCTION INPUT/OUTPUT (Continued)

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
TV_PREDICT_FAIL			TLM	ONORBIT_REND_AUTO_INFLIGHT_UPDATE
UNMOD_ACC_BIAS	Onorbit/Rendezvous Navigation Sequencer	ACCEL_ONORBIT, REND_BIAS_AND_COV_PROP, NAV_ONORBIT_RENDEZVOUS		
UNMOD_ACC_BIAS_TLM			TLM	NAV_ONORBIT_RENDEZVOUS
UR_SUN			Universal Pointing Processing	ONORBIT_REND_R_V_STATE_PROP
USE_IMU_DATA			TLM	ONORBIT_REND_R_V_STATE_PROP
VAR_UNMOD_ACC	Onorbit/Rendezvous Navigation Sequencer	REND_BIAS_AND_COV_PROP		
V _CURRENT_FILT			TLM	NAV_ONORBIT_RENDEZVOUS
VENT_SS			TLM	ACCEL_ONORBIT
V _FILT	Onorbit/Rendezvous Navigation Sequencer	ONORBIT_REND_R_V_STATE_PROP, REND_COV_INIT, NAV_ONORBIT_RENDEZVOUS	Onorbit/Rendezvous Navigation Sequencer	ONORBIT_REND_R_V_STATE_PROP, REND_NAV_FILTER
V _FILT_TLM			TLM	NAV_ONORBIT_RENDEZVOUS

TABLE 4.2.- ONORBIT/RENDEZVOUS NAVIGATION PRINCIPAL FUNCTION INPUT/OUTPUT (Continued)

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
VFORCE	Uplink Processor	ACCEL_ONORBIT	TLM	none
V _GND	Uplink Processor	ONORBIT_REND_AUTO_INFLIGHT_UPDATE		
V _IMU_CURRENT	IMU Redundancy Management	IMU Data Snap		
V _IMU_RESET			User Parameter Processing, TLM	SHUTTLE_RESET
V _LAST	Onorbit/Rendezvous Navigation Sequencer	REND_BIAS_AND_COV_PROP, REND_NAV_INTERP		
V _LAST_FILT	Onorbit/Rendezvous Navigation Sequencer	ONORBIT_REND_R_V_STATE_PROP		
VMP			Onorbit Predictor	ONORBIT_REND_AUTO_INFLIGHT_UPDATE
V_NAV	Startracker SOP	Sensor Data Snap		
V _PRED_FINAL	Onorbit Predictor	ONORBIT_REND_AUTO_INFLIGHT_UPDATE, STATE_VECTOR_PREDICT_TASK	TLM	none

TABLE 4.2.- ONORBIT/RENDEZVOUS NAVIGATION PRINCIPAL FUNCTION INPUT/OUTPUT (Concluded)

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
V _PRED_INIT			TLM, Onorbit Predictor	ONORBIT_REND_AUTO_INFLIGHT_UPDATE,STATE_VECTOR_PREDICT_TASK
V _RESET			User Parameter Processing	SHUTTLE_RESET
V _TV	Onorbit/Rendezvous Navigation Sequencer	ONORBIT_REND_R_V_STATE_PROP, REL_NAV_DISPLAY_UPDATES, COV_LAST_RESET, REND_COV_INIT, NAV_ONORBIT_RENDEZVOUS	Onorbit/Rendezvous Navigation Sequencer	ONORBIT_REND_R_V_STATE_PROP, REND_NAV_FILTER
V _TV_GND	Uplink Processor	ONORBIT_REND_AUTO_INFLIGHT_UPDATE		
V _TV_LAST	Onorbit/Rendezvous Navigation Sequencer	REND_BIAS_AND_COV_PROP, REND_NAV_INTERP		
V _TV_RESET			User Parameter Processing	TARGET_RESET
V _TV_TLM			TLM	NAV_ONORBIT_RENDEZVOUS

4.2.1 Navigation Control (NAV_ONORBIT_RENDEZVOUS)

The navigation control subfunction is responsible for providing the executive logic for the proper execution of the Onorbit/Rendezvous navigation principal function. This subfunction shall perform the following tasks.

- Store the Orbiter and target state vectors from the previous navigation cycle into protected locations for downlist
- Snap all external flags input to NAV, the IMU data, and rendezvous sensor data
- Provide for possible automatic inflight updates of either the Orbiter or target position and velocity vectors
- When in rendezvous phase, the navigation control subfunction will provide the capability to respond to crew requests, via the REL_NAV display, to reinitialize the covariance matrix or to do state vector transfers between the Orbiter position and velocity vectors and the target position and velocity vectors
- Provide for the propagation of the position and velocity vectors for the Orbiter and, when in rendezvous phase, propagate the target position and velocity vectors also
- When in rendezvous phase, this subfunction shall perform the scheduling of the asynchronous covariance propagation and will invoke the Rendezvous Bias and Covariance propagation subfunction when scheduled
- When in rendezvous phase, the navigation control subfunction will provide for the proper processing of the rendezvous sensor data. If the IMU sensed acceleration magnitude falls below a design dependent threshold, then the sensor data will be processed
- Finally, the navigation control subfunction shall invoke the Shuttle reset subfunction to store the updated Shuttle position and velocity vectors for the User Parameter Processing principal function. When in rendezvous phase, the target reset subfunction is called to store the target position and velocity vectors.

A. Detailed Requirements. This subfunction shall perform the following steps in the order indicated.

1. Execute the SNAP_INPUTS code which consists of the following tasks:

   a. Store the Orbiter and target state vectors and time tag from the previous navigation cycle into protected locations for downlist.

      T_LAST_FILT_TLM = T_LAST_FILT
      $\underline{R}$_FILT_TLM = $\underline{R}$_FILT
      $\underline{V}$_FILT_TLM = $\underline{V}$_FILT
      $\underline{R}$_TV_TLM = $\underline{R}$_TV

```
V_TV_TLM = V_TV
SENSOR_BIAS_TLM = SENSOR_BIAS
UNMOD_ACC_BIAS_TLM = UNMOD_ACC_BIAS
```

b. Store the external flags input to NAV and the Orbiter mass into protected locations for NAV.

```
NAV_ANGLES_AIF = ANGLES_AIF
NAV_RANGE_AIF = RANGE_AIF
NAV_RDOT_AIF = RDOT_AIF
NAV_CURR_ORB_MASS = CURR_ORB_MASS
NAV_DO_COVAR_REINIT = DO_COVAR_REINIT
NAV_DO_ORB_TO_TGT = DO_ORB_TO_TGT
NAV_DO_TGT_TO_ORB = DO_TGT_TO_ORB
NAV_DO_OV_UPLINK = DO_OV_UPLINK
NAV_DO_TV_UPLINK = DO_TV_UPLINK
NAV_DO_FLTR_SLOW_RATE = DO_FLTR_SLOW_RATE
NAV_MEAS_ENABLE = MEAS_ENABLE
NAV_MM_202 = MM_202
NAV_PWRD_FLT_NAV = PWRD_FLT_NAV
NAV_RR_ANGLES_ENABLE = RR_ANGLES_ENABLE
NAV_ST_ENABLE = ST_ENABLE
```

c. Snap the measurement data input to NAV, which consist of the following steps:

(1) Snap the IMU accumulated velocity count, time tag, and mean of 50 body quaternion as described in section 4.2.2.1.

```
SNAP(V_CURRENT_FILT, T_CURRENT_FILT, Q_M50BODY_IMU)
```

(2) Snap the sensor measurement data as described in section 4.2.2.2. For the rendezvous radar this includes the shaft, trunnion, range and range rate measurements; the range, range rate and angle data validity flags; the mean of 50 to body additude quaternion at the time of the snap; the time tag for the measurements; and finally a self-test indicator.

```
SNAP REND_RADAR (Q_RR_SHFT, Q_RR_TRUN, Q_RR_RNG,
Q_RR_RNG_DOT, RNG_DATA_GOOD,
RDOT_DATA_GOOD, RR_ANGLE_DATA_GOOD,
Q_M50BODY_RR, T_REND_RADAR, SELF_TEST_FLAG)
```

For startracker, snap the horizontal and vertical measurements; angle data validity flag; the mean of 50 to startracker transformation matrix; the time tag for the startracker measurements, and the target tracking mode indicator.

```
SNAP STAR_TRACKER (Q_ST_HORIZ, Q_ST_VERT,
ST_DATA_GOOD, M_M50_TO_ST, T_STAR_TRACKER,
TRG_TRK_MODE)
```

For COAS, snap the horizontal and vertical angle measurements, the data validity indicator, mean of 50 to body transformation matrix at the time of COAS data snap, the COAS identification, and the COAS time tag.

```
SNAP COAS (Q_COAS_HORIZ, Q_COAS_VERT,
COAS_DATA_GOOD, M_M50_TO_BODY_COAS,
COAS_ID, T_COAS)
```

2. Convert the mean of 50 to body quaternion from the IMU into a mean of 50 to body transformation matrix by calling the quaternion to matrix function described in section 4.10.8.

   $$M\_BODYM50 = \left(QUAT\_TO\_MAT(Q\_M50BODY\_IMU)\right)^T$$

3. Test the value of the DOING_REND_NAV flag to determine if rendezvous phase is active.

   a. If the rendezvous phase is active (DOING_REND_NAV = ON), perform the following steps:

      (1) Subtract any known time bias from the measurement time for each sensor.

      ```
      T_REND_RADAR = T_REND_RADAR - T_BIAS_REND_RADAR
      T_STAR_TRACKER = T_STAR_TRACKER-T_BIAS_ST
      T_COAS = T_COAS-T_BIAS_COAS
      ```

      (2) Call the onorbit rendezvous auto inflight updates subfunction to perform any required ground updates to the Shuttle or target state vectors as described in section 4.2.3.1.

      ```
      CALL: ONORBIT_REND_AUTO_INFLIGHT_UPDATE
      ```

      (3) Call the REL_NAV display updates subfunction to perform any required updates to the state vectors or to the covariance matrix as selected by the crew as described in section 4.2.3.2.

      ```
      CALL: REL_NAV_DISPLAY_UPDATES
      ```

      (4) Invoke the onorbit rendezvous state propagation function described in section 4.2.4 to propagate the position and velocity vectors to current time.

      ```
      CALL: ONORBIT_REND_R_V_STATE_PROP
      ```

      (5) Test the value of the NAV_DO_FLTR_SLOW_RATE switch.

      - If the switch is on, set

        ```
        N_CYCLE = N_CYCLE_SLOW
        ```

so that the slower of the 2 rates for processing the rendezvous measurements will be used.

- If the switch is off, set

  N_CYCLE = N_CYCLE_FAST

  so that the faster rate will be used.

(6) Set the positive feedback flag to inform the REL NAV display which measurement processing rate is being used in navigation.

DOING_FLTR_SLOW_RATE =
NAV_DO_FLTR_SLOW_RATE

(7) Increment the covariance matrix propagation counter.

I_CYCLE = I_CYCLE+1

Then execute the covariance propagation setup code,

EXECUTE: COV_PROP_SETUP CODE,

which consists of the following tasks:

- If IMU sensed delta velocity was used by the state propagation subfunction on the present navigation cycle (USE_IMU DATA = ON) then accumulate the delta velocity into the total delta velocity for the covariance propagation interval and set the covariance powered flight indicator to ON.

  DV_COV = DV_COV+DV_FILT

  COV_PWRD_FLT = ON

- If the contact acceleration (CONT_ACC) for the Shuttle is greater than a design dependent threshold for measurement processing (CONT_ ACC MEAS_THRESHOLD) then a flag is set to indicate that measurements will not be processed on the current covariance propagation subcycle.

  NOISY_NAV_MEAS = ON

(8) If the covariance matrix propagation cycle counter is greater than or equal to the covariance propagation subcycle count (I CYCLE _ N CYCLE) then perform the following tasks:

- Zero the cycle counter

  I_CYCLE = 0

- Call the rendezvous bias and covariance matrix propagation subfunction described in section 4.2.5.

  CALL: REND_BIAS_AND_COV_PROP

  This subfunctions propagates the covariance matrix and propagates the ECRV bias statistics and sets a flag (NAV_MEAS) which determines whether sensor measurements will be incorporated into the state vector on the current covariance propagation cycle.

(9) If the unmodeled acceleration bias and covariance propagation sub-function has set NAV_MEAS = ON, i.e., NOISY_NAV_MEAS = OFF and we are on a covariance propagation cycle, the measurements shall be processed on this cycle by performing the following tasks.

- Call the sensor measurement selection subfunction to select a measurement set for processing as described in section 4.2.6.

  CALL: REND_SENSOR_SELECT

- Call the sensor measurement initialization subfunction to initialize filter statistics and sensor bias states for newly selected measurement types as described in section 4.2.7.

  CALL: REND_NAV_SENSOR_INIT

- Call the rendezvous radar range and range rate measurement subfunction as described in section 4.2.8.1.

  CALL: RRDOT_NAV

  This subfunction incorporates the rendezvous radar range and range rate measurements using the Kalman filter update equations. Increment the mark counter for downlist.

  RRDOT_MARK_NUM = RRDOT_MARK_NUM+1

- If rendezvous radar angles have been selected for processing (DO_RR_ANGLES_NAV = ON) then call the rendezvous radar angles measurement subfunction described in section 4.2.8.2.

  CALL: RR_ANGLE_NAV

  This subfunction incorporates the rendezvous radar shaft and trunnion measurements using the Kalman update.

Increment the mark counter for this measurement

RR_ANGLE_MARK_NUM = RR_ANGLE_MARK_NUM+1

Otherwise (DO_RR_ANGLES_NAV = OFF), if startracker angles have been selected (DO_ST_ANGLES_NAV = ON) then call the startracker angles measurement subfunction described in section 4.2.8.3.

CALL: STAR_TRACKER_NAV

This subfunction incorporates the horizontal and vertical startracker measurements by using the Kalman filter update equations.

Increment the mark counter for this measurement

ST_MARK_NUM = ST_MARK_NUM+1

Otherwise (DO_RR_ANGLES_NAV = OFF and DO_ST_ANGLES_NAV = OFF), call the COAS angles measurement subfunction described in section 4.2.8.4.

CALL: COAS_NAV

This subfunction incorporates the horizontal and vertical COAS measurements using the Kalman update equations.

Increment the mark counter for this measurement

COAS_MARK_NUM = COAS_MARK_NUM+1

- Call measurement processing statistics to calculate display parameters as described in section 4.2.9.

  CALL: MEAS_PROCESSING_STATISTICS_REND

- Set the measurement processing indicator to OFF

  NAV_MEAS = OFF

(10) If the covariance matrix was propagated on this navigation cycle (I__CYCLE = 0) then call the covariance matrix parameters reset subfunction.

CALL: COV_LAST_RESET

(11) Call the target state vector reset subfunction to reset the user parameter processing state vector as described in section 4.1.2.2.2.

CALL: TARGET_RESET

(12) Call the Orbiter state vector reset subfunction to reset the user parameter processing state vector as described in section 4.1.2.1.

CALL: SHUTTLE_RESET

b. If the rendezvous phase is not active (DOING_REND_NAV = OFF) then the following steps shall be performed.

(1) Repeat step (2) under Item a, i.e., call the onorbit rendezvous auto inflight updates to perform any required ground updates to the Shuttle or target position and velocity vectors.

CALL: ONORBIT_REND_AUTO_INFLIGHT_UPDATE

(2) Invoke the onorbit rendezvous state propagation subfunction (see section 4.2.4) to propagate the Orbiter position and velocity vectors to current time.

CALL: ONORBIT_REND_R_V_STATE_PROP

(3) Repeat step (12) under Item a, i.e., call the Orbiter State Vector Reset subfunction to reset the user parameter processing Orbiter state vector as described in section 4.1.2.1.

CALL: SHUTTLE_RESET

B. Interface Requirements. The inputs and outputs for this subfunction are listed in table 4.2.1.

C. Processing Requirements. This subfunction is called by ONORBIT_REND_NAV SEQUENCER (section 4.1.1).

D. Constraints. None

E. Supplementary Information. During OPS 8 rendezvous navigation cannot be active, hence it is not necessary to implement the DOING_REND_NAV = ON branch of the logic for the OPS 8 computer load.

A suggested implementation of this subfunction can be found in Appendix B.

NAV_ONORBIT_RENDEZVOUS

SNAP_INPUTS CODE

COV_PROP_SETUP CODE

TABLE 4.2.1.- NAV_ONORBIT_RENDEZVOUS INPUT/OUTPUT

Variable Name	Input Source	Output Destination
ANGLES_AIF	*	
COAS_DATA_GOOD	Sensor Data Snap	COAS_NAV,*
COAS_ID	Sensor Data Snap	COAS_NAV,*
COAS_MARK_NUM	**	*
CONT_ACC	ONORBIT_REND_R_V_STATE_PROP	
COV_PWRD_FLT		REND_BIAS_AND_COV_PROP
CURR_ORB_MASS	*	
DO_COVAR_REINIT	*	
DO_FLTR_SLOW_RATE	*	
DO_ORB_TO_TGT	*	
DO_OV_UPLINK	*,**	
DO_TGT_TO_ORB	*	
DO_TV_UPLINK	*,**	
DOING_FLTR_SLOW_RATE		*
DO_RR_ANGLES_NAV	REND_SENSOR_SELECT	
DO_ST_ANGLES_NAV	REND_SENSOR_SELECT	
DOING_REND_NAV	*, **	
DV_COV	*,COV_LAST_RESET	REND_BIAS_AND_COV_PROP, *,REND_NAV_INTERP
DV_FILT	ONORBIT_REND_R_V_STATE_PROP	

*Onorbit/Rendezvous principal function, see section 4.2
**Initialization parameters, see section 4.7

TABLE 4.2.1.- NAV_ONORBIT_RENDEZVOUS INPUT/OUTPUT (Continued)

Variable Name	Input Source	Output Destination
I_CYCLE	*,REND_COV_INIT	*
M_BODYM50		ACCEL_ONORBIT
MEAS_ENABLE	*	
MEAS_THRESHOLD	**	
M_M50_TO_BODY_COAS	Sensor Data Snap	*,COAS_NAV
M_M50_TO_ST	Sensor Data Snap	*,STAR_TRACKER_NAV
MM_202	*	
NAV_MEAS	**,REND_BIAS_AND_COV_PROP	*
NAV_ANGLES_AIF		*,REND_SENSOR_SELECT
NAV_CURR_ORB_MASS		*,ACCEL_ONORBIT, ONORBIT_REND_AUTO_INFLIGHT_UPDATE
NAV_DO_COVAR_REINIT		*,REL_NAV_DISPLAY_UPDATE
NAV_DO_FLTR_SLOW_RATE		*
NAV_DO_ORB_TO_TGT		*,REL_NAV_DISPLAY_UPDATE
NAV_DO_OV_UPLINK		*,ONORBIT_REND_AUTO_INFLIGHT_UPDATE
NAV_DO_TGT_TO_ORB		*,REL_NAV_DISPLAY_UPDATE
NAV_DO_TV_UPLINK		*,ONORBIT_REND_AUTO_INFLIGHT_UPDATE
NAV_MM_202		*,REND_SENSOR_SELECT
NAV_MEAS_ENABLE		*,REND_SENSOR_SELECT

*Onorbit/Rendezvous principal function, see section 4.2
**Initialization parameters, see section 4.7

TABLE 4.2.1.- NAV_ONORBIT_RENDEZVOUS INPUT/OUTPUT (Continued)

Variable Name	Input Source	Output Destination
NAV_PWRD_FLT_NAV		*,ONORBIT_REND_R_V_STATE_PROP
NAV_RR_ANGLES_ENABLE		*,REND_SENSOR_SELECT
NAV_RANGE_AIF		*,REND_SENSOR_SELECT
NAV_RDOT_AIF		*,REND_SENSOR_SELECT
NAV_ST_ENABLE		*,REND_SENSOR_SELECT
N_CYCLE	*	
N_CYCLE_SLOW	**	
N_CYCLE_FAST	**	
NOISY_NAV_MEAS	**	REND_BIAS_AND_COV_PROP
PWRD_FLT_NAV	*	
Q_COAS_HORIZ	Sensor Data Snap	COAS_NAV,*
Q_COAS_VERT	Sensor Data Snap	COAS_NAV,*
Q _M50BODY_IMU	IMU Data Snap	QUAT_TO_MAT,*
Q _M50BODY_RR	Sensor Data Snap	RR_ANGLE_NAV,*
Q_RR_RNG	Sensor Data Snap	RRDOT_NAV,*
Q_RR_RNG_DOT	Sensor Data Snap	RRDOT_NAV,*
Q_RR_SHFT	Sensor Data Snap	RR_ANGLE_NAV,*
Q_RR_TRUN	Sensor Data Snap	RR_ANGLE_NAV,*
Q_ST_HORIZ	Sensor Data Snap	STAR_TRACKER_NAV,*
Q_ST_VERT	Sensor Data Snap	STAR_TRACKER_NAV,*
†	QUAT_TO_MAT	
R _FILT	*,ONORBIT_REND_R_V_STATE_PROP,REND_NAV_FILTER	
R _TV	**,*,ONORBIT_REND_R_V_STATE_PROP,REND_NAV_FILTER	

*Onorbit/Rendezvous principal function, see section 4.2
**Initialization parameters, see section 4.7

TABLE 4.2.1.- NAV_ONORBIT_RENDEZVOUS INPUT/OUTPUT (Continued)

Variable Name	Input Source	Output Destination
RDOT_AIF	*	
R _FILT_TLM		*
R _TV_TLM		*
RANGE_AIF	*	
RDOT_DATA_GOOD	Sensor Data Snap	RRDOT_NAV,*
RNG_DATA_GOOD	Sensor Data Snap	RRDOT_NAV,*
RR_ANGLE_DATA_GOOD	Sensor Data Snap	RR_ANGLE_NAV,*
RR_ANGLE_MARK_NUM	**	*
RR_ANGLES_ENABLE	*	
RRDOT_MARK_NUM	**	*
SELF_TEST_FLAG	Sensor Data Snap	RRDOT_NAV, RR_ANGLE_NAV,*
SENSOR_BIAS	SETUP,REND_NAV_FILTER,**	
SENSOR_BIAS_TLM		*
ST_DATA_GOOD	Sensor Data Snap	STAR_TRACKER_NAV,*
ST_ENABLE	*	
ST_MARK_NUM	**	*
T_BIAS_COAS	**	
T_BIAS_REND_RADAR	**	
T_BIAS_ST	**	
T_COAS	Sensor Data Snap	COAS_NAV,*
T_CURRENT_FILT	IMU Data Snap	*,ONORBIT_REND_R_V_STATE_PROP,ONORBIT_SV_INTERP, REND_BIAS_AND_COV_PROP, RRDOT_NAV,RR_ANGLE_NAV, STAR_TRACKER_NAV, COAS_NAV
T_LAST_FILT	ONORBIT_REND_R_V_STATE_PROP,*	

*Onorbit/Rendezvous principal function, see section 4.2
**Initialization parameters, see section 4.7
†Value returned from the function

TABLE 4.2.1.- NAV_ONORBIT_RENDEZVOUS INPUT/OUTPUT (Concluded)

Variable Name	Input Source	Output Destination
T_LAST_FILT_TLM		*
T_REND_RADAR	Sensor Data Snap	RRDOT_NAV, RR_ANGLE_NAV,*
TRG_TRK_MODE	Sensor Data Snap	STAR_TRACKER_NAV,*
T_STAR_TRACKER	Sensor Data Snap	STAR_TRACKER_NAV,*
UNMOD_ACC_BIAS	*,U_A_BIAS_AND_COVINIT, REND_NAV_FILTER,REND_BIAS_AND_COV_PROP	
UNMOD_ACC_BIAS_TLM		*
USE_IMU_DATA	ONORBIT_REND_R_V_STATE_PROP	
$\underline{V}$ _CURRENT_FILT	IMU Data Snap	ONORBIT_REND_R_V_STATE_PROP,*
$\underline{V}$ _FILT	*,ONORBIT_REND_R_V_STATE_PROP,REND_NAV_FILTER	
$\underline{V}$ _TV	**,*,ONORBIT_REND_R_V_STATE_PROP,REND_NAV_FILTER	
$\underline{V}$ _FILT_TLM		*
$\underline{V}$ _TV_TLM		*

*Onorbit/Rendezvous principal function, see section 4.2

**Initialization parameters, see section 4.7

4.2.2 Navigation Data Snap

The Onorbit/Rendezvous navigation principal function shall accomplish the task of snapping navigation data to obtain Orbiter IMU-sensed accumulated velocities, the current attitude quaternion, and sensor data from the rendezvous radar, the star tracker, and the crew optical alinement sight (COAS). The total accumulated sensed velocity is required to account for nongravitional accelerations during integration of Orbiter equations of motion. The sensor data are collected and stored for use in navigation processing during the rendezvous navigation phase.

4.2.2.1 IMU and Attitude Data Snap

The IMU and attitude data snap task will provide the capability to obtain Orbiter IMU-sensed accumulated velocities (expressed in M50 coordinates) and the current attitude quaternion, along with their associated GMT time tag. These data will be obtained through IMU RM/SOP and stored for use in the navigation and user parameter propagation subfunctions.

A. Detailed Requirements. Data from at least one good IMU are required as indicated in the following example:

```
SNAP (V _IMU_SNAP, T_IMU)
```

where V _IMU_SNAP and T_IMU are respective copies of IMU-sensed accumulated velocities and their associated time tag in the user parameter state propagation.

If a consistent set of IMU and attitude data are required, this is indicated by the following second example:

```
SNAP (V _CURRENT_FILT, T_CURRENT_FILT, Q_M50BODY_IMU)
```

where V _CURRENT_FILT and T_CURRENT_FILT are respective copies of IMU-sensed accumulated velocities, their associated time tag, and mean of 50 to body quaternion in the navigation control subfunction.

B. Interface Requirements. The parameter crossreference table between IMU RM/SOP names and their copies of Onorbit/Rendezvous navigation variables is shown in table 4.2.2.1.

C. Processing Requirements. The data from IMU RM/SOP (time tag, accumulated velocities, and attitude quaternions) must be made available for the collection and storage process. The collection rate is indicated by the onorbit/rendezvous navigation sequencer. However, this rate assumes that the available data are fresh. This implies that SOP's processing must be at a rate equal to or greater than the collection rate. This data snap is called by

NAV_ONORBIT_RENDEZVOUS	(section 4.2.1)
ONORBIT_REND_USER_PARAM_STATE_PROP	(section 4.5.1)
OPS_2_OR_8_INITIALIZE	(section 4.1.2)

D. Constraints. The data collections should occur after a complete current set is available and just prior to use in navigation in order to supply current data.

E. Supplementary Information. The snap statement above implies the assignment of current IMU RM/SOP values to the variable names shown in parentheses. The suggested implementation of this subfunction is not described in this document because the discussion involves the processor level interface design concept, which is beyond the scope of this document. However, extreme care must be exercised to provide sequentially time homogeneous data set to the appropriate subfunctions.

ONORBIT/REND NAVIGATION VARIABLES, AND USER PARAMETER PROCESSING VARIABLES

IMU RM & IMU SOP	ONORBIT/REND NAV	USER PARAMETER PROCESSING
T_IMUS_GA	T_CURRENT_FILT	T_IMU
V _IMU_CURRENT	V _CURRENT_FILT	V _IMU_SNAP
Q _BOD_M50	Q _M50BODY_IMU	

4.2.2.2 Rendezvous Sensor Data Snap

During the rendezvous navigation phase, this subfunction collects and stores sensor data from the rendezvous radar, the star tracker, and the crew optical alinement sight (COAS).

The purpose of the rendezvous sensor data snap is to properly save the data sets used in navigation processing for use in the appropriate rendezvous sensor navigation subfunction (section 4.2.8) whereas the actual data may continue to be refreshed by hardware sensor reading and sensor SOP processing.

A. <u>Detailed Requirements</u>.- During the rendezvous navigation phase, data from the external sensors, together with the corresponding data good flags, associated time tags, and the appropriate attitude information valid at those times shall be obtained. A premission-loaded time bias shall then be subtracted from the time tag for each sensor. The equations are:

1. For the rendezvous radar:

```
SNAP REND_RADAR (Q_RR_SHFT, _Q_RR_TRUN, Q_RR_RNG, Q_RR_RNG_DOT,
RNG_DATA_GOOD, RDOT_DATA_GOOD, RR_ANGLE_DATA_GOOD, Q_M50BODY_RR,
T_REND_RADAR, SELF_TEST_FLAG)
```

where Q_RR_SHFT is the shaft angle measurement

Q_RR_TRUN is the trunnion angle measurement

RR_ANGLE_DATA_GOOD is the validity flag of the above measurements

Q_RR_RNG is the radar range measurement

RNG_DATA_GOOD is the respective data good flag

Q_RR_RNG_DOT is the radar range rate reading

RDOT_DATA_GOOD is the respective validity indicator

T_REND_RADAR is the time at which these measurements are considered to have been effected

Q_M50BODY_RR is the gimbal angle quaternion array

SELF_TEST_FLAG is the flag indicating whether the rendezvous radar is operating in the self-test mode

```
T_REND_RADAR = T_REND_RADAR - T_BIAS_REND_RADAR
```

2. For the star tracker,

```
SNAP STAR_TRACKER (Q_ST_HORIZ, Q_ST_VERT, ST_DATA_GOOD,
M_M50_TO_ST, T_STAR_TRACKER, TKG_TRK_MODE)
```

where:

Q_ST_HORIZ is the horizontal angle measurement

Q_ST_VERT is the vertical angle measurement

ST_DATA_GOOD is the data good flag relative to these angle measurements

M_M50_TO_ST is the M50-to-star tracker sensor coordinate system rotation matrix at the time T_STAR_TRACKER

T_STAR_TRACKER is the time tag

TRG_TRK_MODE Is the flag indicating whether the star tracker is in the target tracking mode

T_STAR_TRACKER = T_STAR_TRACKER - T_BIAS_ST

3. For the COAS,

SNAP COAS (Q_COAS_HORIZ, Q_COAS_VERT, COAS_DATA_GOOD, M_M50_TO_BODY_COAS, COAS_ID, T_COAS)

where:

Q_COAS_HORIZ is the horizontal angle measurement

Q_COAS_VERT is the vertical angle measurement

COAS_DATA_GOOD is the data good flag relative to these angle measurements

COAS_ID is the COAS select indicator

T_COAS is the time of the measurement

M_M50_TO_BODY_COAS is the M50-to-COAS sensor coordinate system transformation matrix at the time T_COAS

T_COAS = T_COAS - T_BIAS_COAS

B. Interface Requirements.- The parameter name cross reference between rendezvous radar SOP and Onorbit/Rendezvous Navigation variables is shown in table 4.2.2.2-1. The cross reference table between star tracker SOP and Onorbit/Rendezvous Navigation variables is given in table 4.2.2.2-2.

C. Processing Requirements.- The data from the sensors (measurements, ID's, validity flags, rotation matrices, and time tags) must be made available for the collection and storage process. The collection rate (not necessarily sensor interrogations) is indicated by the Onorbit/Rendezvous Navigation Sequencer. However, this rate assumes that the available data are fresh.

This implies that SOP's processing must be at a rate equal to or greater than the collection rate. This data is called by NAV_ONORBIT_RENDEZVOUS (section 4.2.1).

D. Constraints.- The data collections should occur after a complete current set is available and just prior to use in navigation in order to supply current data.

E. Supplementary Information.- The snap statement above implies the assignment of current SOP values to the variable names shown in parentheses. The suggested implementation of this subfunction is not described in this document because the discussion involves the processor level interface design concept which is beyond the scope of this document. However, extreme care must be exercised to provide sequentially time homogeneous data sets to the appropriate subfunctions.

TABLE 4.2.2.2-1.- PARAMETER CROSS REFERENCE TABLE BETWEEN
RR SOP AND ONORBIT/REND NAV VARIABLES

RR SOP	ONORBIT/REND NAV
RR_ROLLO	Q_RR_SHFT
RR_PITCHO	Q_RR_TRUN
RR_RANGEO	Q_RR_RNG
RR_RNGRO	Q_RR_RNG_DOT
RR_RNG_DG	RNG_DATA_GOOD
RR_RNGR_DG	RDOT_DATA_GOOD
RR_ANG_DG	RR_ANGLE_DATA_GOOD
Q_M50BODY_RR	Q_M50BODY_RR
RR_TIM	T_REND_RADAR
RR_SELF_TEST	SELF_TEST_FLAG

TABLE 4.2.2.2-2.- PARAMETER NAME CROSS REFERENCE TABLE BETWEEN STAR TRACK SOP AND ONORBIT/REND NAV VARIABLES

STAR TRCK SOP	ONORBIT/REND NAV
H_NAV	Q_ST_HORIZ
V_NAV	Q_ST_VERT
DATA_GOOD	ST_DATA_GOOD
T_M50_ST	M_M50_TO_ST
TIME	T_STAR_TRACKER
NAV_TARGET	TRG_TRK_MODE
COAS_HORIZ	Q_COAS_HORIZ
COAS_VERT	Q_COAS_VERT
NAV_SIGHT	COAS_DATA_GOOD
T_M50_BODY	M_M50_TO_BODY_COAS
AXN	COAS_ID
T_TARLOS	T_COAS

4.2.3 State and Covariance Matrix Updates

The Onorbit/Rendezvous Navigation principal function shall provide for the capability to perform automatic inflight updates of the Orbiter and/or target position and velocities, crew requested covariance matrix reinitialization, and crew requested state vector transfers. The capability to perform the position-velocity update of either the Orbiter or target will be handled by the automatic inflight updates subfunction while the capability to handle crew requests for state vector transfers and covariance matrix reinitialization will be done by the REL NAV display updates subfunction.

4.2.3.1 Auto Inflight Updates (ONORBIT_REND_AUTO_INFLIGHT UPDATE)

The automatic inflight updates subfunction shall determine when an update to the shuttle or target state vector is required by the ground and then schedule the state vector prediction task to predict the uplinked state vectors to the time tag T_RESET, established by the Orbiter state vector reset subfunction (see section 4.1.2.1). If a state vector update has occurred during rendezvous navigation phase, the covariance matrix shall be reinitialized.

A. Detailed Requirements.

1. First, a local flag SV_UPDATE, which indicates whether a state vector update has occurred, is set to OFF.

2. If the ground has indicated that the Orbiter vector is uplinked (NAV_DO_OV_UPLINK = ON) then the following tasks will be performed in the order indicated.

    a. The prediction task indicator flag, PRED_USE, is interrogated for a zero value to determine if the state vector prediction task is available for scheduling. If the state vector prediction task is available then the following tasks are performed.

        (1) The following parameters are defined preceding the scheduling of the state vector prediction task:

PRED_USE = 1
OV_PREDICT_FAIL = OFF
PRED_ ORB_MASS = NAV_CURR_ORB_MASS
PRED_ORB_AREA = REF_ORB_AREA
PRED_ORB_CD = REF_ORB_CD
GMDP = GM_DEG
GMOP = GM_ORD
DMP = DFL
VMP = VFLOV_PRED
ATMP = 1
PRED_STEP = PREC_STEP_PRED
R _PRED_INIT = R _GND
V _PRED_INIT = V _GND
T_PRED_INIT = T_GND

(2) Next, the state vector prediction task is scheduled.

SCHEDULE: STATE_VECTOR_PREDICT_TASK

b. If the PRED_USE flag is nonzero, it is tested for a value of 3 which indicates that the state vector prediction task has determined that the prediction interval is too large; hence no prediction will take place. If PRED_USE = 3 then the following flags are set.

PRED_USE = 0, indicating the state vector prediction task is available.
DO_OV_UPLINK = OFF
OV_PREDICT_FAIL = ON, indicating that no prediction will take place for the Orbiter state vectors.

c. If the PRED_USE flag is not set to 3 then it is tested for a value of 2 which indicates that the state vector prediction task has successfully predicted the Orbiter state vector. If PRED_USE = 2 then the following actions are taken.

PRED_USE = 0, indicating that the state vector prediction task is available.
T_LAST_FILT = T_PRED_FINAL   predicted Orbiter
$\underline{R}$_FILT = $\underline{R}$_PRED_FINAL   vectors and time tag
$\underline{V}$_FILT = $\underline{V}$_PRED_FINAL   are stored.
SV_UPDATE = ON, indicating that an inflight update has taken place.
T_ORB_STATE_UPDATE
= T_LAST_FILT, the time tag of the inflight update is saved for downlist.
DO_OV_UPLINK = OFF

3. If the ground has indicated that the target vector is uplinked (NAV_DO_TV_UPLINK = ON) then the DOING_REND_NAV flag and the PRED_USE flags are interrogated to determine if the uplinked target state vectors are to be stored or if the target state prediction logic should be exercised.

a. If rendezvous navigation is active, DOING_REND_NAV = ON, or the target state prediction logic is in progress, PRED_USE $\geq$ 7, then the PRED_USE flag is tested to determine the appropriate action to follow.

(1) If PRED_USE = 0, target prediction is to be scheduled. The following actions are taken:

- The prediction parameters are initialized.

PRED_USE = 7
TV_PREDICT_FAIL = OFF
GMDP = GM_DEG

GMOP = GM_ORD
DMP = DFL
VMP = VFLTV_PRED
ATMP = ATFL_TV
PRED_STEP = PREC_STEP_PRED
R _PRED_INIT = R _TV_GND
V _PRED_INIT = V _TV_GND
T_PRED_INIT = T_TV_GND

- The state vector prediction task is scheduled.

SCHEDULE: STATE_VECTOR_PREDICT_TASK

(2) If the PRED_USE flag is equal to 9, the state vector prediction task has determined that the prediction interval is too large; hence no prediction will take place. The following flags are set.

PRED_USE = 0, indicating the state vector prediction task is available.
DO_TV_UPLINK = OFF
TV_PREDICT_FAIL = ON, indicating that no prediction will take place for the target state vectors.

(3) If the PRED_USE flag is equal to 8, the state vector prediction task has successfully predicted the target state vectors. The following actions are taken.

PRED_USE = 0, indicating that the state vector prediction task is available.
T_TV = T_PRED_FINAL   predicted target vectors and
R _TV = R _PRED_FINAL   time tag are stored.
V _TV = V _PRED_FINAL
SV_UPDATE = ON, indicating that an inflight update has taken place.
T_TV_STATE_UPDATE
= T_TV, the time tag of the inflight update is saved for downlist.
DO_TV_UPLINK = OFF

b. If rendezvous navigation phase is not active, DOING_REND_NAV = OFF, and the target state prediction task is not in progress for an inflight update, PRED_USE < 7, then the uplinked state vector and associated time tag are stored, the time of the uplink is stored, and the DO_TV_UPLINK flag is turned OFF.

T_TV = T_TV_GND
R _TV = R _TV_GND
V _TV = V_TV_GND
T_TV_STATE_UPDATE = T_TV
DO_TV_UPLINK = OFF

4. Finally, the covariance initialization subfunction is invoked if the SV_ UPDATE flag is ON and DOING_REND_NAV = ON, indicating that the rendezvous navigation phase is active.

CALL: REND_COV_INIT

B. Interface Requirements. The inputs and outputs for this subfunction are given in table 4.2.3.1.

C. Processing Requirements. This subfunction is called by

NAV_ONORBIT_RENDEZVOUS (section 4.2.1)

D. Constraints. None

E. Supplementary Information. A suggested implementation of this subfunction is given in Appendix B.

ONORBIT_REND_AUTO_INFLIGHT_UPDATE

TABLE 4.2.3.1.- ONORBIT_REND_AUTO_INFLIGHT_UPDATE INPUT/OUTPUT

Variable Name	Input Source	Output Destination
ATFL_TV	**	
ATMP		*
DFL	**	
DMP		*
DO_OV_UPLINK		*
DO_TV_UPLINK		*
DOING_REND_NAV	*,**	
GM_DEG	**	
GMDP		*
GMOP		*
GM_ORD	**	
NAV_CURR_ORB_MASS	NAV_ONORBIT_RENDEZVOUS	
NAV_DO_OV_UPLINK	NAV_ONORBIT_RENDEZVOUS	*
NAV_DO_TV_UPLINK	NAV_ONORBIT_RENDEZVOUS	*
OV_PREDICT_FAIL		*
PRED_ORB_AREA		*
PRED_ORB_CD		*
PRED_ORB_MASS		*
PRED_STEP		*
PRED_USE	*,**,STATE_VECTOR_PREDICT_TASK	*,STATE_VECTOR_PREDICT_TASK

*Onorbit/Rendezvous principal function, see section 4.2
**Initialization parameters, see section 4.7

TABLE 4.2.3.1.- ONORBIT_REND_AUTO_INFLIGHT_UPDATE INPUT/OUTPUT.- Continued

Variable Name	Input Source	Output Destination
PREC_STEP_PRED	**	
REF_ORB_AREA	**	
PREC_STEP_PRED	**	
REF_ORB_AREA	**	
REF_ORB_CD	**	
R _FILT		REL_NAV_DISPLAY_UPDATES, REND_COV_INIT, ONORBIT_REND_R_V_STATE_PROP, COV_LAST_RESET
R _GND	*	
R _PRED_FINAL	*	
R _PRED_INIT		*
R _TV		REL_NAV_DISPLAY_UPDATES, REND_COV_INIT, ONORBIT_REND_R_V_STATE_PROP, COV_LAST_RESET
R _TV_GND	*	
T_GND	*	
T_ORB_STATE_UPDATE		*
T_LAST_FILT		ONORBIT_REND_R_V_STATE_PROP, REND_COV_INIT
T_PRED_FINAL	STATE_VECTOR_PREDICT_TASK	

*Onorbit/Rendezvous principal function, see section 4.2
**Initialization parameters, see section 4.7

TABLE 4.2.3.1.- ONORBIT_REND_AUTO_INFLIGHT_UPDATE INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
T_PRED_INIT		*,STATE_VECTOR_PREDICT_TASK
T_TV		*,REND_COV_INIT
T_TV_GND	*	
T_TV_STATE_UPDATE		*
TV_PREDICT_FAIL		*
V _FILT		REL_NAV_DISPLAY_UPDATES, REND_COV_INIT, ONORBIT_REND_R_V_STATE_PROP, COV_LAST_RESET
VFLOV_PRED	**	
VFLTV_PRED	**	
V _GND	*	
VMP		*
V _PRED_FINAL	*	
V _PRED_INIT		*
V _TV		REL_NAV_DISPLAY_UPDATES, REND_COV_INIT, ONORBIT_REND_R_V_STATE_PROP, COV_LAST_RESET
V_TV_GND	*	

*Onorbit/Rendezvous principal function, see section 4.2

**Initialization parameters, see section 4.7

4.2.3.1.1 State vector prediction task (STATE_VECTOR_PREDICT_TASK).- It is a requirement that the execution of either the Onorbit/Rendezvous Sequencer or the Onorbit/Rendezvous Navigation principal function has priority over the execution of the Onorbit Precision State Prediction principal function. The purpose of the state vector prediction task is to provide logic, to be executed at the same priority as the predictor, which will allow the navigation and sequencer principal functions to execute in a nominal manner despite delays in the predictor due to its low priority execution.

This task shall be scheduled by the Rendezvous Navigation Initialization subfunction and the Auto Inflight Updates subfunction to provide for the prediction of either the Orbiter and/or target state vectors to the time tag specified by the Orbiter State Vector Reset subfunction. The state vector prediction task shall also test the prediction interval size against a gross reasonability design dependent threshold to prevent the Onorbit Precision State Prediction principal function from predicting over inordinately large time intervals. The state vector prediction task shall also set the value of the PRED_USE flag which indicates the status of prediction to the scheduling subfunction. The PRED_USE flag has the following values.

Value of PRED USE	Definition
0	Prediction task is available for scheduling
1	Prediction task scheduled for Orbiter inflight update
2	Prediction task completed for Orbiter inflight update
3	Prediction task failed for Orbiter inflight update
4	Prediction task scheduled for rendezvous navigation initialize
5	Prediction task completed for rendezvous navigation initialize
6	Prediction task failed for rendezvous navigation initialize
7	Prediction task scheduled for target inflight update
8	Prediction task completed for target inflight update
9	Prediction task failed for target inflight update

The scheduling subfunction will set PRED_USE to the values of 0, 1, 4, or 7. All other values are set by the prediction task.

A. Detailed Requirements.

1. A local flag, PRED_TASK_COMPLETE, which signals when the prediction task is completed, is initialized to OFF upon entering the state vector prediction task.

   PRED_TASK_COMPLETE = OFF

2. Next, the absolute value of the difference between the initial prediction time tag, T_PREDICT_INIT, and the time tag T_RESET is tested against a gross reasonability (design dependent) threshold in order to prevent the Onorbit Precision State Prediction principal function from predicting over excessively large time intervals.

$$|\text{T\_PRED\_INIT} - \text{T\_RESET}| > \text{MAX\_TIME\_TOL}$$

a. If the time threshold is exceeded then the flag PRED_USE is incremented by 2 in order to communicate to the invoking subfunction that the prediction task has failed.

$$\text{PRED\_USE} = \text{PRED\_USE} + 2$$

b. If the maximum time tolerance MAX_TIME_TOL is not exceeded, then the following logic (steps (1), (2), (3), and (4)) is executed in a cyclic fashion until the PREDICT_TASK_COMPLETE flag is turned ON.

```
DO UNTIL
PREDICT_TASK_COMPLETE = ON
```

(1) First, the time tag for the final predicted state vectors is set to the current value of T_RESET as defined by the navigation task.

$$\text{T\_PRED\_FINAL} = \text{T\_RESET}$$

(2) Next, the Onorbit Precision State Prediction principal function is invoked.

```
CALL: ONORBIT_PREDICT
```

(3) The predicted state vectors and time tag are saved for the next state vector prediction task cycle.

$$\underline{R}\text{\_PRED\_INIT} = \underline{R}\text{\_PRED\_FINAL}$$
$$\underline{V}\text{\_PRED\_INIT} = \underline{V}\text{\_PRED\_FINAL}$$
$$\text{T\_PRED\_INIT} = \text{T\_PRED\_FINAL}$$

(4) Using NAV's current value for T_RESET, the prediction interval magnitude is tested against a hard coded time tolerance SV_TIME_TAG_DIFF to determine if the predicted vectors are close to the time tag T_RESET.

$$|\text{T\_PRED\_INIT} - \text{T\_RESET}| < \text{SV\_TIME\_TAG\_DIFF}$$

(It should be noted that any value for SV_TIME_TAG_DIFF which is smaller than the length of a single navigation cycle will cause the predictor task to predict to current time (T_RESET).)

If the predicted state vector time tag is close to the current reset time tag (i.e., less than or equal to SV_TIME_TAG_DIFF), then the prediction complete flag is set to ON and the PRED_USE flag is incremented by one to indicate to the proper navigation subfunction that the prediction task is completed and predicted state vectors are available.

PRED_TASK_COMPLETE = ON

PRED_USE = PRED_USE + 1

Steps (1), (2), (3), and (4) are repeated until PRED_TASK_COMPLETE is ON as specified in step b.

B. Interface Requirements. Input and output parameters for the state vector prediction task are specified in table 4.2.3.1.1. Since the STATE_VECTOR_PREDICT_TASK (SVPT) executes in parallel to the navigation task, the value of T_RESET can update in NAV while the SVPT is in process. The SVPT makes cyclic use of T_RESET in two places in its logic and should be provided NAV's current value of T_RESET each time it uses the parameter.

C. Processing Requirements. The following subfunctions schedule the state vector prediction task.

Rendezvous Navigation Initialization
(REND_NAV_INIT, Section 4.1.2.2)
Auto Inflight Updates
(ONORBIT_REND_AUTO_INFLIGHT_UPDATE, Section 4.2.3.1)

D. Constraints. The STATE_VECTOR_PREDICT_TASK is one of several users of the onorbit predictor. Since the same compool locations are used by all users of this principal function for setup and output and since the use of the predictor by the STATE_VECTOR_PREDICT_TASK may be interrupted by other users, it is required that its predictor parameters be protected from alteration by other users during execution of this principal function.

E. Supplemental Information. A suggested implementation of the state vector prediction task may be found in the Appendix B flowchart STATE_VECTOR_PREDICT_TASK.

TABLE 4.2.3.1.1.- STATE_VECTOR_PREDICT_TASK INPUT/OUTPUT

Variable Name	Input Source	Output Destination
MAX_TIME_TOL	**	
PRED_USE	ONORBIT_REND_AUTO_ INFLIGHT_UPDATE, REND_NAV_INIT	ONORBIT_REND_AUTO_ INFLIGHT_UPDATE, REND_NAV_INIT,*,***
R_PRED_INIT		*,***
R_PRED_FINAL	*,***	
SV_TIME_TAG_DIFF	**	
T_PRED_INIT	ONORBIT_REND_AUTO_ INFLIGHT_UPDATE, REND_NAV_INIT	*,***
T_PRED_FINAL		*,***,REND_NAV_INIT, ONORBIT_REND_AUTO_ INFLIGHT_UPDATE
T_RESET	SHUTTLE_RESET, *,***	
V_PRED_INIT		*,***
V_PRED_FINAL	*,***	

*Onorbit/Rendezvous principal function, see section 4.2

**Initialization parameters, see section 4.7

***Onorbit/Rendezvous Navigation Sequencer Principal Function, see section 4.1

4.2.3.2 REL_NAV Display Updates (REL_NAV_DISPLAY_UPDATES)

This subfunction is required to respond to crew request, made via item entries on the REL_NAV display, to reinitialize the covariance matrix or to perform a state vector transfer from the Orbiter state vector to the target state vector, or vice versa. This subfunction is exercised on each state vector propagation cycle by the NAV control logic.

A. Detailed Requirements. The following steps shall be performed (in the order indicated) in response to crew request while rendezvous navigation is active:

1. Turn off the positive feedback flags to the REL_NAV display.

   DID_COVAR_REINIT = OFF

   DID_ORB_TO_TGT = OFF

   DID_TGT_TO_ORB = OFF

2. Test the NAV_DO_COVAR_REINIT flag to determine if the crew has requested that the covariance matrix be reinitialized. If NAV_DO_COVAR_REINIT = ON, perform the following steps:

   a. Call the covariance matrix initialization subfunction to reinitialize the covariance matrix, to zero the unmodeled acceleration states, and compute the total filter vehicle acceleration vector (see section 4.1.2.2.1):

      CALL: REND_COV_INIT

   b. Reset the covariance matrix initialization indicator flag and the positive feedback flag (to the REL_NAV display) as follows:

      DO_COVAR_REINIT = OFF

      DID_COVAR_REINIT = ON

3. Test the NAV_DO_ORB_TO_TGT flag to determine if the crew has requested a state vector transfer (Orbiter to target).

   a. If NAV_DO_ORB_TO_TGT = ON, the following steps are taken:

      (1) The target state vectors are set equal to the Orbiter state vectors:

         $\underline{R}$_TV = $\underline{R}$_FILT

         $\underline{V}$_TV = $\underline{V}$_FILT

(2) The covariance matrix initialization subfunction is called to reinitialize the covariance matrix, to zero the unmodeled acceleration states, and compute the total filter vehicle acceleration vector (see section 4.1.2.2.1):

CALL: REND_COV_INIT

(3) The flags indicating the Orbiter-to-target state vector transfer and the positive feedback (to the REL_NAV display) shall be set:

DO_ORB_TO_TGT = OFF

DID_ORB_TO_TGT = ON

b. If the NAV_DO_ORB_TO_TGT flag is off, test the NAV_DO_TGT_TO_ORB flag to determine if the crew has requested a state vector transfer (target to Orbiter). If NAV_DO_TGT_TO_ORB = ON, the following steps are taken:

(1) The Orbiter state vector shall be set equal to the target state vector:

$\underline{R}$ _FILT = $\underline{R}$ _TV

$\underline{V}$ _FILT = $\underline{V}$ _TV

(2) The covariance matrix initialization subfunction is called to reinitialize the covariance matrix, to zero the unmodeled acceleration state, and compute the total filter vehicle acceleration vector (see section 4.1.2.2.1):

CALL: REND_COV_INIT

(3) The flags indicating target-to-Orbiter state vector transfer and the positive feedback (to the REL_NAV display) shall be set:

DO_TGT_TO_ORB = OFF

DID_TGT_TO_ORB = ON

B. Interface Requirements. Input and output parameters are listed in table 4.2.3.2.

C. Processing Requirements. This subfunction is called by

NAV_ONORBIT_RENDEZVOUS (section 4.2.1)

D. Constraints. Covariance matrix reinitialization as well as the state vector transfer must be activated via crew input to the REL_NAV display and only while the Rendezvous Navigation principal function is active.

E. Supplementary Information. A suggested implementation in the form of a detailed flowchart can be found in Appendix B under the name:

REL_NAV_DISPLAY_UPDATES

TABLE 4.2.3.2.- REL_NAV_DISPLAY_UPDATES INPUT/OUTPUT

Variable Name	Input Source	Output Destination
DID_COVAR_REINIT		*
DID_ORB_TO_TGT		*
DID_TGT_TO_ORB		*
NAV_DO_COVAR_REINIT	NAV_ONORBIT_RENDEZVOUS	
NAV_DO_ORB_TO_TGT	NAV_ONORBIT_RENDEZVOUS	
NAV_DO_TGT_TO_ORB	NAV_ONORBIT_RENDEZVOUS	
R _FILT	ONORBIT_REND_R_V_STATE_ PROP,ONORBIT_REND_AUTO_ INFLIGHT_UPDATE, REND_NAV_FILTER	REND_COV_INIT, COV_LAST_RESET, ONORBIT_REND_R_V_STATE_ PROP
R _TV	ONORBIT_REND_R_V_STATE_ PROP,ONORBIT_REND_AUTO_ INFLIGHT_UPDATE,*, REND_NAV_FILTER	REND_COV_INIT, COV_LAST_RESET, ONORBIT_REND_R_V_STATE_ PROP
V _FILT	ONORBIT_REND_R_V_STATE_ PROP,ONORBIT_REND_AUTO_ INFLIGHT_UPDATE, REND_NAV_FILTER	REND_COV_INIT, COV_LAST_RESET, ONORBIT_REND_R_V_STATE_ PROP
V _TV	ONORBIT_REND_R_V_STATE_ PROP,ONORBIT_REND_AUTO_ INFLIGHT_UPDATE,*, REND_NAV_FILTER	REND_COV_INIT, COV_LAST_RESET, ONORBIT_REND_R_V_STATE_ PROP

*Onorbit/Rendezvous principal function, see section 4.2

4.2.4 Position and Velocity State Propagation (ONORBIT_REND_R_V_STATE_PROP)

The position and velocity state propagation subfunction is contained within the Onorbit/Rendezvous Navigation principal function and is used to perform a number of tasks related to the propagation of the Orbiter and target vehicle state vectors. The subfunction will be employed to propagate the Orbiter and target vehicle state vectors from the time of the previous navigation cycle to the current navigation cycle time.

Prior to state vector propagation, the task of snapping IMU and attitude data shall be performed to obtain the current time (T_CURRENT_FILT), accumulated IMU sensed velocity ($\underline{V}$ _CURRENT_FILT), and an attitude quaternion (Q_M50BODY_IMU). For detailed requirements of these data snaps, see IMU and Attitude Data Snap, section 4.2.2.1.

During vehicle state propagation, various acceleration models are available for use in the determination of perturbing acceleration values and include gravitational accelerations (always used) and nongravitational accelerations (drag, and a limited venting and uncoupled RCS thrusting model). The nongravitational acceleration models shall be used only when the vehicle sensed acceleration ($\underline{D}$V FILT), obtained from the IMU accumulated sensed velocity ($\underline{V}$ CURRENT_FILT), is judged to be insignificant; that is, below a predetermined value (DA_THRESHOLD TEST). A detailed description of the acceleration models may be found in section 4.2.4.1.1. State vector propagation will employ only 1 scheme for integration of the equations of motion, the SUPER_G algorithm (see section 4.2.4.1). This algorithm will be used for propagating both the Orbiter and target state vectors in both coasting and powered flight.

A. Detailed Requirements.- The computations that shall be performed for propagation of the position and velocity vectors are initiated by a call to the Onorbit/Rendezvous position and velocity state propagation subfunction (ONORBIT_REND_R_V_STATE_PROP) in the following form:

CALL: ONORBIT_REND_R_V_STATE_PROP

The following will be performed in the order indicated.

1. The IMU navigation acceleration threshold value initialized by ILOAD and updated by the crew on the IMU ALIGN DISPLAY will be loaded into the DA_THRESHOLD parameter:

   DA_THRESHOLD = IMU_NAV_ACCEL_THRESH

2. The acceleration model flags shall be set up for Orbiter coasting flight propagation:

   IGD = GM_DEG
   IGO = GM_ORD
   IDRAG = 1
   IVENT = 1

3. Next the change in velocity (DV_FILT) and the corresponding time interval (DT_FILT) used in the advancement of both Orbiter and target state vectors shall be calculated by subtraction of the previous cycle values (V _LAST_FILT and T_LAST_FILT) from the current IMU snapped values.

   DV_FILT = V _CURRENT_FILT - V _LAST_FILT
   DT_FILT = T_CURRENT_FILT- T_LAST_FILT

4. An acceleration averaged over the desired interval is next determined by:

   CONT_ACC = |DV_FILT| / DT_FILT F3

5. Next the powered flight navigation flag, NAV_PWRD_FLT_NAV, shall be tested, and flags specifying the acceleration models to be used to propagate the Orbiter state vector on this navigation cycle shall be defined.

   a. If the NAV_PWRD_FLT_NAV flag is found to be ON, acceleration threshold tests must be made to determine the acceleration models to be used in propagating the Orbiter state vector. The procedure is as follows:

      (1) Convert the DA_THRESHOLD value from micro g's to feet/sec^2, normalize according to the propagation time interval (DT_FILT), and store the new value into DA_THRESHOLD_TEST.

      DA_THRESHOLD_TEST = DA_THRESHOLD (GO) (10^{-6})/DT_FILT F3

      ("GO" is the micro G's to feet/sec^2 conversion factor.)

      (2) Test the acceleration value (CONT_ACC) against the converted threshold value, DA_THRESHOLD_TEST.

         - If CONT_ACC > DA_THRESHOLD_TEST, turn the IMU data-use flag ON and set the flags for modeling drag and vent forces to 0 (OFF).

           USE_IMU_DATA = ON
           IDRAG = 0
           IVENT = 0

         - Otherwise, set the IMU data-use flag to OFF, and force the accumulated velocity for the propagation interval to zero.

           USE_IMU_DATA = OFF
           DV_FILT = 0.

F3 This equation shall be protected against division by zero (Reference 3.6-3).

(3) Test the acceleration value (CONT_ACC) against the ILOADED MEAS_THRESHOLD value to determine if the powered flight gravitational potential model should be used.

- If CONT_ACC > MEAS_THRESHOLD, use the powered flight values for the potential models

  IGD = GM_DEG_LOW
  IGO = GM_ORD_LOW

- Otherwise, the gravitational parameters will remain set to the free flight values.

b. In the situation where the NAV_PWRD_FLT_NAV is found to be OFF, the Orbiter sensed acceleration is not used in propagating the Orbiter state vector. Therefore, set

DV_FILT = 0.
USE_IMU_DATA = OFF

6. Next the SUPER_G algorithm is called to propagate the Orbiter state vector, using the acceleration models computed above.

CALL: SUPER_G

IN LIST: IGD,IGO,IDRAG,IVENT,ATFL_OV,R_FILT,V_FILT,T_
LAST_FILT,T_CURRENT_FILT,DT_FILT,DV_FILT

OUT LIST: R_FILT,V_FILT,G_NEW

The values of R_FILT and V_FILT output by SUPER_G are the required propagated position and velocity vectors of the Orbiter during a powered or coasting flight mode. The vector G_NEW is a modeled total acceleration vector obtained according to the specified flag settings and corresponding to R_FILT, V_FILT and T_CURRENT_FILT.

7. Once the Orbiter state vector has been propagated, the DOING_REND_NAV flag will be tested to determine if propagation of the target vehicle state is required.

```
 _______________
!
!DOING_REND_NAV
!_______________
```

If the DOING_REND_NAV flag is ON, the target vehicle state vector will be propagated to time T_CURRENT_FILT from time T_TV using the SUPER_G algorithm.

CALL: SUPER_G

IN LIST: GM_DEG, GM_ORD, DFL, VFL_TV, ATFL_TV, R _TV, V _TV, T_TV, T_CURRENT_FILT, DT_FILT, TDV_FILT

OUT LIST: R _TV, V _TV, G _TV

The out list variables represent the required advanced target vehicle state vector and total target acceleration vector. Here R _TV is the target position vector, V _TV the target velocity vector, and G _TV the corresponding acceleration vector, all determined for the time T_CURRENT_FILT.

The time T_CURRENT_FILT is saved as T_TV.

T_TV = T_CURRENT_FILT

8. After satisfaction of the operations caused by the DOING_REND_NAV flag value, T_LAST_FILT and V _LAST_FILT will be updated to T_CURRENT_FILT and V _CURRENT_FILT in preparation for the next cycle through ONORBIT_REND_R_V_STATE_PROP.

T_LAST_FILT = T_CURRENT_FILT
V _LAST_FILT = V _CURRENT_FILT

9. A call to the solar ephemeris subfunction will provide sine and cosine functions of the sun's current position preparatory to calculation of the Earth-sun unit vector. The unit vector will be constructed as follows:

(a) CALL: SOLAR_EPHEM

IN LIST: T_CURRENT_FILT

OUT LIST: SDEC, CDEC1, COS_SOL_RA, SIN_SOL_RA

(b) UR_SUN$_1$ = COS_SOL_RA CDEC1
UR_SUN$_2$ = SIN_SOL_RA CDEC1
UR_SUN$_3$ = SDEC

The solar unit vector is required by the Universal pointing processing principal function.

B. Interface Requirements.- Input and output parameters for the position and velocity state propagation subfunction trunk logic are given in table 4.2.4.

C. Processing Requirements.- This subfunction shall be called by the Onorbit/Rendezvous Navigation principal function (NAV_ONORBIT_RENDEZVOUS).

D. Constraints.- The following constraints apply:

1. The acceleration models task is needed not only by the navigation state propagation subfunction, but also by the Onorbit Precision State Prediction principal function and by the User Parameter state propagation subfunction. Each user of acceleration models shall set its own flags and therefore requires a different calculation. To protect against interference in the acceleration computations, it is important that these computations not be interrupted.

2. The current Orbiter mass (CURR_ORB_MASS) will be initialized for use by NAV by the Onorbit/Rendezvous Navigation Sequencer principal function and will be maintained for NAV by the Onorbit Guidance principal function.

E. Supplementary Information.- A suggested implementation of this subfunction in the form of detailed flow diagrams may be found in Appendix D under the following:

ONORBIT_REND_R_V_STATE_PROP

TABLE 4.2.4.- ONORBIT_REND_R_V_STATE_PROP INPUT/OUTPUT

Variable Name	Input Source	Output Destination
ATFL_OV	**	SUPER_G
ATFL_TV	**	SUPER_G
CDEC1	SOLAR_EPHEM	
COS_SOL_RA	SOLAR_EPHEM	
CONT_ACC		NAV_ONORBIT_RENDEZVOUS, *
DFL	**	SUPER_G
DOING_REND_NAV	*,**	
DT_FILT		REND_COV_INIT,*,SUPER_G
DV_FILT		REND_COV_INIT, SUPER_G, NAV_ONORBIT_RENDEZVOUS, *
GM_DEG	**	SUPER_G
GM_DEG_LOW	**	
GM_ORD	**	SUPER_G
GM_ORD_LOW	**	
G _NEW	SUPER_G	REND_BIAS_AND_COV_PROP, *
G _TV	SUPER_G	REND_BIAS_AND_COV_PROP, REND_NAV_INTERP,*
IDRAG		SUPER_G, REND_COV_INIT, REND_NAV_INTERP,*
IGD		SUPER_G,REND_COV_INIT, REND_NAV_INTERP,*
IGO		SUPER_G,REND_COV_INIT, REND_NAV_INTERP,*
IVENT		SUPER_G,REND_COV_INIT, REND_NAV_INTERP,*

*See Onorbit/Rendezvous Nav P.F. I/O

**Initialization parameters, see section 4.7

TABLE 4.2.4.- ONORBIT_REND_R_V_STATE_PROP INPUT/OUTPUT.- Continued

Variable Name	Input Source	Output Destination
MEAS_THRESHOLD	**	
NAV_PWRD_FLT_NAV	NAV_ONORBIT_RENDEZVOUS	
R _FILT	SUPER_G, ONORBIT_REND_AUTO_INFLIGHT_UPDATE, REL_NAV_DISPLAY_UPDATES, REND_NAV_FILTER,*	SUPER_G, REND_COV_INIT, COV_LAST_RESET, REND_NAV_FILTER, REND_NAV_INTERP, REL_NAV_DISPLAY_UPDATES, REND_BIAS_AND_COV_PROP,*, SHUTTLE_RESET,NAV_ONORBIT_RENDEZVOUS
R _TV	ONORBIT_REND_AUTO_INFLIGHT_UPDATE, REL_NAV_DISPLAY_UPDATES, REND_NAV_FILTER,*, SUPER_G	COV_LAST_RESET, REND_COV_INIT, TARGET_RESET, REND_NAV_FILTER, REND_BIAS_AND_COV_PROP, REL_NAV_DISPLAY_UPDATES, REND_NAV_INTERP,*, SUPER_G,NAV_ONORBIT_RENDEZVOUS
SDEC	SOLAR_EPHEM	
SIN_SOL_RA	SOLAR_EPHEM	
T_CURRENT_FILT	NAV_ONORBIT_RENDEZVOUS	SUPER_G
T_LAST_FILT	*,ONORBIT_REND_AUTO_INFLIGHT_UPDATE	SUPER_G,*,SHUTTLE_RESET, COV_LAST_RESET, REND_COV_INIT,NAV_ONORBIT_RENDEZVOUS
T_TV		*, REND_COV_INIT, SUPER_G
UR_SUN		*
TDV_FILT	**	SUPER_G
USE_IMU_DATA		NAV_ONORBIT_RENDEZVOUS,*

*Onorbit/Rendezvous principal function, see section 4.2
**Initialization parameters, see section 4.7

TABLE 4.2.4.- ONORBIT_REND_R_V_STATE_PROP INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
V _CURRENT_FILT	NAV_ONORBIT_RENDEZVOUS	
V _FILT	SUPER_G, REND_NAV_FILTER, ONORBIT_REND_AUTO_INFLIGHT_UPDATE,REL_NAV_DISPLAY_UPDATES,*	SUPER_G, REND_COV_INIT, COV_LAST_RESET, SHUTTLE_RESET, REND_BIAS_AND_COV_PROP, REND_NAV_FILTER, REND_NAV_INTERP, REL_NAV_DISPLAY_UPDATES,*,NAV_ONORBIT_RENDEZVOUS
VFL_TV	**	SUPER_G
V _LAST_FILT	*	SHUTTLE_RESET
V _TV	ONORBIT_REND_AUTO_INFLIGHT_UPDATE, REL_NAV_DISPLAY_UPDATES, REND_NAV_FILTER,*, SUPER_G	COV_LAST_RESET, REL_NAV_DISPLAY_UPDATES, REND_COV_INIT,TARGET_RESET, REND_NAV_INTERP, REND_NAV_FILTER,*, SUPER_G, REND_BIAS_AND_COV_PROP,NAV_ONORBIT_RENDEZVOUS

*Onorbit/Rendezvous principal function, see section 4.2

**Initialization parameters, see section 4.7

4.2.4.1 Position and Velocity Propagation (SUPER_G)

During powered flight navigation phases, the equations of motion used for Orbiter state propagation have the form of a Taylor series truncated at the term in h^3, where h is the step size. The integration scheme used, called SUPER_G, is an improved version of the AVERAGE_G method, containing a correction cycle.

During Orbiter powered flight conditions, non-gravitational accelerations sensed by the IMU's are incorporated into the state vector in the SUPER_G state propagation algorithm when above a pre-set threshold. For Orbiter coasting flight or target state vector propagation, the accelerations are always modeled in the SUPER_G algorithm.

A. Detailed Requirements.-

The SUPER_G subfunction shall be invoked whenever the following statement is encountered:

CALL: SUPER_G

IN LIST: GMD,GMO,DM,VM,ATM,$\underline{R}$ _IN, $\underline{V}$ _IN, T_IN, T_FIN, DT_FILT, $\underline{DV}$_IN

OUT LIST: $\underline{R}$ _FIN,$\underline{V}$ _FIN, $\underline{G}$ _OUT

1. Upon initiation of SUPER_G, an initial total acceleration, $\underline{G}$ _INT, will be calculated by the $\underline{A}$CCEL_ONORBIT function as:

$$\underline{G}\text{\_INT} = \underline{A}\text{CCEL\_ONORBIT (GMD,GMO,DM,VM,ATM,}\underline{R}\text{\_IN,}\underline{V}\text{\_IN,T\_IN)}$$

where the function arguments are the appropriate flag settings (that is, the degree and order of the gravitational potential model, drag mode, vent mode, and attitude mode flag settings), current position vector, current velocity vector, and previous time.

2. Next the position vector is advanced using the current position and velocity vectors, the time interval DT_FILT, initial acceleration vector $\underline{G}$ _INT calculated in the previous step, and the value of $\underline{DV}$_FILT:

$$\underline{R}\text{\_FIN} = \underline{R}\text{\_IN} + \text{DT\_FILT}\left(\underline{V}\text{\_IN} + .5\left(\underline{DV}\text{\_IN} + \text{DT\_FILT}\ \underline{G}\text{\_INT}\right)\right)$$

3. Find a new value of the total acceleration vector, based on the advanced position vector, velocity vector, and time tag:

$$\underline{G}\text{\_OUT} = \underline{A}\text{CCEL\_ONORBIT (GMD,GMO,DM,VM,ATM,}\underline{R}\text{\_FIN,}\underline{V}\text{\_IN,T\_FIN)}$$

4. The vehicle velocity vector is corrected using an average total acceleration difference.

$$\underline{V}\text{\_FIN} = \underline{V}\text{\_IN} + \underline{DV}\text{\_IN} + .5\ \text{DT\_FILT}\left(\underline{G}\text{\_INT} + \underline{G}\text{\_OUT}\right)$$

5. Finally, the vehicle position vector is corrected by:

$$\underline{R}\_FIN = \underline{R}\_FIN + ((\underline{G}\_OUT - \underline{G}\_INT)\ DT\_FILT^2)/6.$$

The velocity and position vectors calculated in steps 4 and 5 constitute the required propagated state.

B. Interface Requirements.- The input and output required for the SUPER_G subfunction are listed in table 4.2.4.1. Required inputs for use of the ACCEL_ONORBIT function are given in section 4.2.4.1.1.

C. Processing Requirements.- The SUPER_G subfunction is called by the following:

ONORBIT_REND_R_V_STATE_PROP

D. Constraints.- The SUPER_G subroutine will be called to integrate the Orbiter and target state vectors during coasting and powered flight.

E. Supplementary Information.- A suggested implementation of the SUPER_G subroutine, in the form of a detailed flowchart, is presented in Appendix B under

SUPER_G

TABLE 4.2.4.1.- SUPER_G INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
GMD	IGD	ONORBIT_REND_R_V_ STATE_PROP	
GMO	IGO	ONORBIT_REND_R_V_ STATE_PROP	
DM	IDRAG	ONORBIT_REND_R_V_ STATE_PROP	
VM	IVENT	ONORBIT_REND_R_V_ STATE_PROP	
ATM	ATFL_OV	ONORBIT_REND_R_V_ STATE_PROP	
R _IN	R _FILT	ONORBIT_REND_R_V_ STATE_PROP	
V _IN	V _FILT	ONORBIT_REND_R_V_ STATE_PROP	
T_IN	T_LAST_FILT	ONORBIT_REND_R_V_ STATE_PROP	
T_FIN	T_CURRENT_ FILT	ONORBIT_REND_R_V_ STATE_PROP	
DT_FILT	DT_FILT	ONORBIT_REND_R_V_ STATE_PROP	
DV_IN	DV_FILT	ONORBIT_REND_R_V_ STATE_PROP	
GMD	GM_DEG	ONORBIT_REND_R_V_ STATE_PROP	
GMO	GM_ORD	ONORBIT_REND_R_V_ STATE_PROP	

TABLE 4.2.4.1.- SUPER_G INPUT/OUTPUT.- Continued

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
DM	DFL	ONORBIT_REND_R_V_ STATE_PROP	
VM	VFL_TV	ONORBIT_REND_R_V_ STATE_PROP	
ATM	ATFL_TV	ONORBIT_REND_R_V_ STATE_PROP	
R _IN	R _TV	ONORBIT_REND_R_V_ STATE_PROP	
V _IN	V _TV	ONORBIT_REND_R_V_ STATE_PROP	
T_IN	T_TV	ONORBIT_REND_R_V_ STATE_PROP	
T_FIN	T_CURRENT_ FILT	ONORBIT_REND_R_V_ STATE_PROP	
DT_FILT	DT_FILT	ONORBIT_REND_R_V_ STATE_PROP	
DV_IN	TDV_FILT	ONORBIT_REND_R_V_ STATE_PROP	
R _FIN	R _FILT		ONORBIT_REND_R_V_ STATE_PROP
V _FIN	V _FILT		ONORBIT_REND_R_V_ STATE_PROP
G _OUT	G _NEW		ONORBIT_REND_R_V_ STATE_PROP
R _FIN	R _TV		ONORBIT_REND_R_V_ STATE_PROP

TABLE 4.2.4.1.- SUPER_G INPUT/OUTPUT.- Continued

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
$\underline{V}$ _FIN	$\underline{V}$ _TV		ONORBIT_REND_R_V_ STATE_PROP
$\underline{G}$ _OUT	$\underline{G}$ _TV		ONORBIT_REND_R_V_ STATE_PROP

TABLE 4.2.4.1.- SUPER_G INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
GMD		ACCEL_ONORBIT
GMO		ACCEL_ONORBIT
DM		ACCEL_ONORBIT
VM		ACCEL_ONORBIT
ATM		ACCEL_ONORBIT
R_IN		ACCEL_ONORBIT
R_FIN		ACCEL_ONORBIT
V_IN		ACCEL_ONORBIT
T_IN		ACCEL_ONORBIT
T_FIN		ACCEL_ONORBIT
†	ACCEL_ONORBIT	

†Only the value of ACCEL_ONORBIT is passed

4.2.4.1.1 Acceleration models (ACCEL_ONORBIT).- During orbital operations, models to account for gravitational, vent and thrust, and vehicle aerodynamic drag accelerations shall be available. These models are to be used in the Orbiter state vector propagation whenever the IMU sensed acceleration magnitude is below a given threshold level. When the sensed acceleration magnitude is above the threshold level, the gravitational acceleration model only will be used. In the Orbiter state prediction mode, only the gravitational and drag acceleration models may be used. During propagation or prediction of a target vehicle state, the gravitational and drag acceleration models may be used. Additionally, during the target state propagation mode, capability exists to incorporate the unmodeled acceleration biases as determined by the Kalman filter in REND_NAV_FILTER.

A. Detailed Requirements.- This function is activated whenever the statement ACCEL_ONORBIT (GMD,GMO,DM,VM,ATM,R,V,T) is encountered,

where:

GMD	input degree of Earth gravitational potential model (ACCEL_EARTH_GRAV)
GMO	input order of Earth gravitational potential model (ACCEL_EARTH_GRAV)
DM	flag indicating use (1) or non-use (0) of vehicle drag acceleration model (ACCEL_ONORBIT_DRAG)
VM	flag indicating use (1) or non-use (0) of vent and thrust model (ACCEL_ONORBIT_VENT_AND_THRUST)
ATM	attitude mode flag (used when DM and/or VM are set to 1.)
R	position vector of vehicle in M50 coordinates
V	velocity vector of vehicle in M50 coordinates
T	position and velocity vectors time tag

The following steps will be performed (in the order indicated) whenever the ACCEL_ONORBIT function is activated.

1. The values of G, the gravitational acceleration due to the Earth's nonspherical shape, D, the drag model acceleration vector, and VENT, the acceleration due to vent and thrusting shall be initially nulled.

$\underline{G} = 0.$
$\underline{D} = 0.$
$\underline{VENT} = 0.$

The current Earth fixed to M50 transformation FIFTY will be constructed.

$$FIFTY = EARTH\_FIXED\_TO\_M50\_COORD(T)$$

Here, the EARTH_FIXED_TO_M50_COORD function is defined in section 4.10.2.

Next the input M50 position vector will be transformed to Earth fixed coordinates.

$$\underline{R}\ \_EF = FIFTY^T\ \underline{R}$$

Components of the Earth fixed position unit vector will be determined by the following:

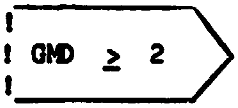
$$R\_INV = 1./\ |\ \underline{R}\ | \qquad \text{F3}$$

$$\underline{UR} = R\_INV\ \underline{R}\ \_EF$$

The acceleration vector due to the Earth's gravitational attraction as a point mass will be determined by:

$$\underline{G}\ \_CENTRAL = -\ EARTH\_MU\ R\_INV^3\ \underline{R}$$

2. Next the value of GMD shall be tested to determine if the gravitational acceleration vector due to the Earth's non-sphericity ($\underline{G}$) shall be determined.



a. If GMD is equal to or greater than 2, the ACCEL_EARTH_GRAV code will be executed - which is a model formulated using S. Pines' spherical harmonics development.

(1) The following variables are to be set up to serve as starting values for recursive relations used in the Pines formulation:

$$RO\_ZERO = EARTH\_RADIUS\_GRAV\ R\_INV$$

$$RO\_N = RO\_ZERO\ EARTH\_MU\ R\_INV^2$$

$$A_{1,2} = 3.\ UR_3$$

$$A_{2,2} = 3.$$

$$L = 1$$

F3 This equation shall be protected against division by zero (Reference 3.6-3).

$\mathrm{AUXILIARY} = 0.$

$\mathrm{ZETA\_REAL}_1 = 1.$

$\mathrm{ZETA\_IMAG}_1 = 0.$

A is a two-column array used for temporary storage of Legendre polynomials and derived Legendre functions (which are latitude-dependent terms), and RO_N is the distance-related term. AUXILIARY is an intermediate scalar variable.

(2) Recursive calculations shall then proceed, using as many components of the one-column arrays ZETA_REAL and ZETA_IMAG as required to account for the effects of tesseral harmonics. ZETA_REAL and ZETA_IMAG are the only terms that depend on the vehicle's longitude.

```
Do for I = 1 to GMO:

    $\mathrm{ZETA\_REAL}_{I+1} = \mathrm{UR}_1\ \mathrm{ZETA\_REAL}_I - \mathrm{UR}_2\ \mathrm{ZETA\_IMAG}_I$

    $\mathrm{ZETA\_IMAG}_{I+1} = \mathrm{UR}_1\ \mathrm{ZETA\_IMAG}_I + \mathrm{UR}_2\ \mathrm{ZETA\_REAL}_I$
```

(3) The derived Legendre functions shall then be obtained by means of recursion formulas, multiplied by appropriate combinations of tesseral harmonics (Legendre polynomials shall be multiplied by zonal harmonics coefficients), and stored as certain auxiliary variables F1, F2, F3, and F4.

```
Do for N = 2 to GMD the following three steps:

-   $A_{N+1,1} = 0.$
    $A_{N+1,2} = (2.\ N + 1.)\ A_{N,2}$
    $A_{N,1} = A_{N,2}$
    $A_{N,2} = \mathrm{UR}_3\ A_{N+1,2}$

-   Do for J = 2 to N:

        $A_{N-J+1,1} = A_{N-J+1,2}$
        $A_{N-J+1,2} = (\mathrm{UR}_3\ A_{N-J+2,2} - A_{N-J+2,1})/J$

-   $F1 = 0.$
    $F2 = 0.$
    $F3 = -A_{1,1}\ \mathrm{ZONAL}_N$
    $F4 = -A_{1,2}\ \mathrm{ZONAL}_N$
```

(These account for the zonal harmonics contributions.)

(4) If the maximum order of tesserals wanted has not been attained (i.e., if N $\leq$ GMO), do for N1 = 1 to N:

$$F1 = F1 + N1\ A_{N1,1}\ (C_L\ \mathrm{ZETA\_REAL}_{N1} + S_L\ \mathrm{ZETA\_IMAG}_{N1})$$

$F2 = F2 + N1\ A_{N1,1}\ (S_L\ \text{ZETA\_REAL}_{N1} - C_L\ \text{ZETA\_IMAG}_{N1})$
$DNM = C_L\ \text{ZETA\_REAL}_{N1+1} + S_L\ \text{ZETA\_IMAG}_{N1+1}$
$F3 = F3 + DNM\ A_{N1+1,1}$
$F4 = F4 + DNM\ A_{N1+1,2}$
$L = L + 1$

(These take into account the contributions of the tesseral and sectorial harmonics.)

(5) $\text{RO\_N} = \text{RO\_N}\ \text{RO\_ZERO}$
$G_1 = G_1 + \text{RO\_N}\ F1$
$G_2 = G_2 + \text{RO\_N}\ F2$
$G_3 = G_3 + \text{RO\_N}\ F3$
$\text{AUXILIARY} = \text{AUXILIARY} + \text{RO\_N}\ F4$

(These equations multiply the sum of zonal and tesseral effects by appropriate distance-related factors, store the results as components of the acceleration vector $\underline{G}$, and prepare for final computation by obtaining the intermediate scalar variable AUXILIARY, which accounts for an additional effect proportional to the unit radius vector $\underline{UR}$).

b. Once these calculations have been completed (N = GMD) and stored, the Earth-fixed acceleration vector shall be obtained and rotated to the M50 coordinate system.

$$\underline{G} = \underline{G} - \text{AUXILIARY}\ \underline{UR}$$

$$\underline{G} = \text{FIFTY}\ \underline{G}$$

This is the gravitational acceleration vector needed for equations of motion of the Shuttle. Values of GMD and of GMO may be set by the user independently. However, it is necessary that $GMO \leq GMD$. A maximum value of 4 for GMD shall be used, which will make the array $\underline{\text{ZONAL}}$ have 4 components, the arrays $\underline{C}$ and $\underline{S}$ have 9 components each, $\underline{\text{ZETA\_REAL}}$ and $\underline{\text{ZETA\_IMAG}}$ have 5 each, and $\underline{A}$ has a maximum dimension of 5 by 2.

Terms shown in Earth's gravity calculations as C_L and S_L are usually represented by $C_{n,m}$ and $S_{n,m}$, respectively, but were renumbered for single subscript utilization; the terms called ZONAL_N correspond to $J_N = -C_{N,0}$.

The S. Pines formulation of gravitational potential may be found, in condensed form, in the paper "Uniform Representation of the Gravitational Potential and its Derivatives," AIAA Journal, Vol. 11, No. 11, November 1973. In expanded form, and with an earlier draft of the computer program herein presented, it is contained in MDC Report W0013, NASA CR 14/478, of 9 February 1976, Pines' Nonsingular Gravitational Potential: Derivation, Description and Implementation.

3. Next a test will be performed to determine if vent and thrust accelerations are to be modeled. It should be noted that the vent and thrust acceleration model is used only during state propagation (i.e., this model is not used for Orbiter or target vehicle state prediction).

VM = 1

a. If VM is equal to 1, vent and thrust accelerations are to be modeled for use during Orbiter or target vehicle propagations. The ACCEL_ONORBIT_VENT_AND_THRUST CODE will be executed as follows:

(1) A check of the ATM flag will be made to determine if the modeled vent and thrust acceleration will apply to the Orbiter state propagation (ATM = 0), or target vehicle propagation (ATM = 2).

ATM = 2

- If ATM is equal to 2, the vent and thrust acceleration is for the target vehicle. A check of the SHUTTLE_FILTER_FLAG will be made to determine if the unmodeled acceleration bias (UNMOD_ACC_BIAS) applies to the Orbiter (SHUTTLE_FILTER_FLAG = ON), or the target vehicle (SHUTTLE_FILTER_FLAG = OFF).

SHUTTLE_FILTER_FLAG = OFF

If the Shuttle filter flag is OFF, the target vehicle vent and thrust vector (VENT) will be set equal to the unmodeled acceleration bias; otherwise the value of VENT will remain at its initialized value. The value of UNMOD_ACC_BIAS is determined in the Onorbit/Rendezvous Navigation principal function by the Kalman filter. The bias acceleration is applicable to either the Shuttle or target vehicle as indicated by setting of the SHUTTLE_FILTER_FLAG as mentioned above.

- If the value of ATM does not equal 2 in step 3a(1), the vent and thrust modeled acceleration will be for the Orbiter. The value of the SHUTTLE_FILTER_FLAG will next be checked

SHUTTLE_FILTER_FLAG = ON

and if on, the vent and thrust acceleration vector VENT is set equal to UNMOD_ACC_BIAS.

VENT=UNMOD_ACC_BIAS

(2) The value of time (T) is tested to determine if it lies within the vent and thrust model action time span as specified by the values of TFON and TFOFF,

$T \geq$ TFON
and
T < TFOFF

and if so, a body contact force vector (VFORCE) and the current Orbiter mass (NAV_CURR_ORB_MASS) will be used to construct the Orbiter venting and acceleration vector. A multiplication by M_BODYM50 transforms the vector from body to M50 coordinates. This acceleration vector is then added to VENT.

VENT = VENT + M_BODYM50 (VFORCE/NAV_CURR_ORB_MASS) F3

- The value VENT, the Orbiter vent and thrust modeled acceleration vector, is stored as VENT_SS to be available for downlist.

4. After the ACCEL_ONORBIT_VENT_AND_THRUST logic has been satisfied, the drag model flag (DM) shall be tested

DM = 1

and if true (DM = 1), the vehicle drag acceleration vector shall be determined.

a. The first step to be performed in the vehicle drag computational flow will be to exercise the SOLAR_EPHEM subfunction by the following:

CALL: SOLAR_EPHEM

INLIST: T

OUTLIST: SDEC,CDEC1,COS_SOL_RA,SIN_SOL_RA

F3 This equation shall be protected against division by zero (Reference 3.6-3).

Where:

CDEC1 = cosine of the solar declination
COS_SOL_RA = cosine of the solar right ascension
SDEC = sine of the solar declination
SIN_SOL_RA = sine of the solar right ascension
T = time of desired computation

b. Next the ONORBIT_DENSITY code will be exercised to provide the atmospheric density value associated with the vehicle position. This section of code is initiated with determination of the vehicle altitude (ALT) above the reference ellipsoid through use of the H_ELLIPSOID function (see section 4.2.4.1.1.1).

$$\text{ALT} = \text{H\_ELLIPSOID}(\underline{\text{R}})$$

where $\underline{\text{R}}$ is the vehicle M50 position vector.

(1) The next series of expressions will be performed to determine GDI, one of the Babb-Mueller atmospheric density diurnal factors.

$\text{CDEC1} = \text{CDEC1 R\_INV}$
$\text{SDEC} = \text{SDEC R\_INV R}_3$
$\text{CSFST} = \text{R}_1\ \text{COS\_SOL\_RA COS\_LAG}$
$\text{CSSND} = \text{R}_1\ \text{SIN\_SOL\_RA SIN\_LAG}$
$\text{SIFST} = \text{R}_2\ \text{SIN\_SOL\_RA COS\_LAG}$
$\text{SSND} = \text{R}_2\ \text{COS\_SOL\_RA SIN\_LAG}$
$\text{COS\_PSI} = \text{SDEC} + \text{CDEC1 (CSFST-CSSND + SIFST + SSND)}$
$\text{GDI} = [(1.0 + \text{COS\_PSI})/2.0)^{\text{GDIE}}$

where COS_LAG, SIN_LAG and GDIE are design dependent parameters (see section 4.7).

(2) A check of the vehicle altitude above the reference ellipsoid (ALT) will be made to see if it is greater than ALT_L (the Babb-Mueller density layer altitude).

ALT > ALT_L

If the statement is true, the layering index integer K will be set to 2; otherwise K will be set to 1.

(3) The night-time vertical density profile factor (AFH) will be determined

$$\text{AFH} = \text{ABM}_{1,K} + \text{ABM}_{2,K}\ \text{ALT} + \text{ABM}_{3,K}\ /\text{ALT} \qquad \text{F3}$$

where $ABM_{1\ TO\ 3,K}$ are mission dependent calibration coefficients (see section 4.7). The diurnal density effect will be determined next.

$$BFH = (BM_1 + BM_2\ ALT + BM_3/ALT)\ GDI$$ F3

The $BM_{1\ TO\ 3}$ are mission dependent calibration coefficients (see section 4.7). A seasonal-latitudinal term will be determined as follows,

$$CBM1 = ALT - C\_DENSEA$$
$$CBM1 = (CBD1(R\_INV)^2\ ABS\ (R_3)R_3)CBM1\ EXP\ (CBD2\ CBM1)$$

C_DENSEA, CBD1 and CBD2 are design dependent parameters (see section 4.7).

(4) The atmospheric density will now be determined as

$$RHO = RREF\ EXP\ (AFH + BFH + CBM1\ CBM2)$$

Here RREF is a design dependent parameter and CBM2 is a seasonal-latitudinal mission dependent parameter (see section 4.7).

c. Next the ACCEL_ONORBIT_DRAG code will be executed using the atmospheric density value of (RHO) previously determined. The code will begin with calculation of the vehicle velocity vector in Earth-relative M50 coordinates as determined by

$$\underline{V}\_R = \underline{V}\_REL\ (\underline{V},\underline{R})$$

where $\underline{V}$ _REL ($\underline{V}$,$\underline{R}$) is the Earth relative velocity function (see section 4.2.4.1.1.2), and $\underline{V}$,$\underline{R}$ are the vehicle velocity and position vectors in M50 coordinates.

(1) A test to determine if ATM is greater than 0 is performed:

```
 ____________
!            \
! ATM   >  0  >
!____________/
```

- If ATM is greater than 0, a second test will be performed to determine if ATM is equal to 1.

```
 __________
!          \
! ATM = 1   >
!__________/
```

F3 This equation shall be protected against division by zero (Reference 3.6-3).

If ATM is equal to 1, the drag computation will be for Orbiter state prediction and will use the following configuration parameters:

VEH_MASS = PRED_ORB_MASS
CD = PRED_ORB_CD
AREA = PRED_ORB_AREA

The values of PRED_ORB_MASS, PRED_ORB_CD and PRED_ORB_AREA to be used will be determined by that principal function which initiates a call to the Onorbit predictor.

If in the test of ATM, ATM does not equal 1, the ensuing drag computation will be used for a target vehicle state propagation or prediction mode. The following configuration parameters will be set:

VEH_MASS = TARGET_MASS
CD = TARGET_CD
AREA = TARGET_AREA

Here TARGET_MASS, TARGET_CD and TARGET_AREA are mission dependent parameters (I-LOAD parameters - section 4.7).

(2) If in the test of ATM, ATM is not greater than 0, the computation of drag will be used for Orbiter state propagation.

In the above event, VEH_MASS and AREA will be designated by:

VEH_MASS = NAV_CURR_ORB MASS
AREA = REF_ORB_AREA

where NAV_CURR_ORB_MASS was set by NAV_ONORBIT_RENDEZVOUS and REF_ORB_AREA is a design dependent parameter (see section 4.7).

The coefficient of drag, CD, shall be determined as follows for Orbiter state propagation.

- First, $\underline{V}$ _REL_BODY, the vehicle velocity vector relative to the atmosphere, but expressed in body coordinates, is determined as:

$$\underline{V}\text{\_REL\_BODY} = (\text{M\_BODYM50})^{T}\,\underline{V}\text{\_R}$$

where M_BODYM50 is the transformation matrix of body to M50 coordinates and $\underline{V}$ _R has been determined in c.

- Next the square of the sine of the vehicle angle of attack (SA) will be determined as:

$$SA = (V\_REL\_BODY_3)^2/(V\_REL\_BODY_1{}^2 + V\_REL\_BODY_3{}^2) \qquad F3$$

The sine of the vehicle sideslip angle, SB, will be determined by the following:

$$SB = |V\_REL\_BODY_2|/|\underline{V}\_R| \qquad F3$$

The sine of twice the sideslip angle S2B will be determined by:

$$S2B = 2.\ SB\ SQRT\ (1.-SB^2) \qquad F4$$

- The Orbiter coefficient of drag CD may now be determined as:

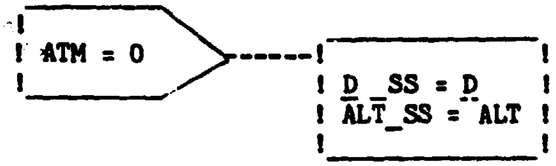
$$CD = \left[(CDF + CDN\ SA^{EXP\_SHAPE\_FACTOR})\ (1.-SB) + CDS\ SB + CDA\ S2B\ SA\ \right] KFACTOR$$

The vehicle configuration constants CDF,CDN,EXP_SHAPE_FACTOR, CDS and CDA are design dependent parameters (see section 4.7). KFACTOR is the drag coefficient adjustment factor and is mission dependent (see section 4.7) and included as an uplink parameter (see section 4.9). The parameter KFACTOR is included to allow adjustments to the vehicle drag coefficient due to configuration parameter uncertainties, atmospheric density model deviations or other causes during Orbiter state propagation.

(3) After vehicle parameters VEH_MASS, AREA and CD have been determined by one of the logic paths of 4c(1) or of 4c(2) as dictated by the value of the ATM flag, a vehicle drag acceleration vector $\underline{D}$ in M50 coordinates will be determined by:

$$\underline{D} = -0.5\ CD\ RHO\ AREA\ |\ \underline{V}\_R\ |\ \underline{V}\_R/VEH\_MASS \qquad F3$$

(4) If the drag acceleration $\underline{D}$ is being computed for the Orbiter during state propagation, ATM=0, it shall be stored in $\underline{D}$_SS for downlist. ALT, the vehicle altitude above the Earth's reference ellipsoid, will be also saved in ALT_SS for downlist.



F3 This equation shall be protected against division by zero (Reference 3.6-3).

F4 This equation shall be protected against square roots of a negative number (Reference 3.6-4).

5. Finally, the acceleration values G, D and VENT determined by the non-spherical Earth gravitational, vehicle drag, and vent and thrust models, respectively, will be combined with the value of point mass Earth gravitational acceleration (G _CENTRAL) to result in the final ACCEL_ONORBIT function M50 output vector.

ACCEL_ONORBIT = G _CENTRAL + G + D + VENT

B. Interface Requirements.- The input and output data are shown in table 4.2.4.1.1. It should be noted that ACCEL_ONORBIT is treated as a function subprogram; that is, the computed value of the acceleration vector will be returned and occupy the position of the function name.

C. Processing Requirements.- This function subprogram shall be performed each time the function name is encountered with suitable expressions for arguments such as:

ACCEL_ONORBIT (GMD,GMO,DM,VM,ATM,R,V,T)

The ACCEL_ONORBIT function is used by the following:

SUPER_G
ONORBIT_SV_INTERP
REND_COV_INIT
PINES_METHOD
AVERAGE_G_INTEGRATOR

D. Constraints.- The currently functioning propagator and a predictor may need different acceleration models at the same time. It is therefore necessary that execution of the acceleration calculations be protected from interruption by other users.

E. Supplementary Information.- A suggested implementation of the ACCEL_ONORBIT in the form of detailed flowcharts may be found in Appendix B under the names:

ACCEL_ONORBIT FUNCTION
ACCEL_EARTH_GRAV CODE
ACCEL_ONORBIT_VENT_AND_THRUST CODE
SOLAR_EPHEM
ONORBIT_DENSITY CODE
H_ELLIPSOID FUNCTION
ACCEL_ONORBIT_DRAG CODE
V _REL FUNCTION

and the following from Appendix C:

EARTH_FIXED_TO_M50_COORD FUNCTION

TABLE 4.2.4.1.1.- ACCEL_ONORBIT INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
ATM	ATM	SUPER_G	
DM	DM	SUPER_G	
GMD	GMD	SUPER_G	
GMO	GMO	SUPER_G	
VM	VM	SUPER_G	
R	R_IN	SUPER_G	
T	T_IN	SUPER_G	
V	V_IN	SUPER_G	
ATM	ATM	SUPER_G	
DM	DM	SUPER_G	
GMO	GMO	SUPER_G	
GMD	GMD	SUPER_G	
VM	VM	SUPER_G	
R	R_FIN	SUPER_G	
T	T_FIN	SUPER_G	
		SUPER_G	
V	V_IN	SUPER_G	
ATM	IATM	ONORBIT_SV_INTERP	
DM	IDM	ONORBIT_SV_INTERP	
GMD	IGD	ONORBIT_SV_INTERP	
GMO	IGO	ONORBIT_SV_INTERP	
VM	IVM	ONORBIT_SV_INTERP	
R	R_RESID	ONORBIT_SV_INTERP	
T	T_RESID	ONORBIT_SV_INTERP	
V	V_RESID	ONORBIT_SV_INTERP	
ATM	ATFL_OV	REND_COV_INIT	
DM	IDRAG	REND_COV_INIT	
IGD	IGD	REND_COV_INIT	
IGO	IGO	REND_COV_INIT	
VM	IVENT	REND_COV_INIT	
R	R_FILT	REND_COV_INIT	
T	T_LAST_FILT	REND_COV_INIT	
V	V_FILT	REND_COV_INIT	

TABLE 4.2.4.1.1.- ACCEL_ONORBIT INPUT/OUTPUT.- Continued

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
ATM	ATFL_TV	REND_COV_INIT	
DM	DFL	REND_COV_INIT	
IGD	GM_DEG	REND_COV_INIT	
IGO	GM_ORD	REND_COV_INIT	
$\underline{R}$	R_TV	REND_COV_INIT	
$\underline{\underline{T}}$	T_TV	REND_COV_INIT	
VM	VFL_TV	REND_COV_INIT	
$\underline{V}$	V_TV	REND_COV_INIT	
ATM	ATM	PINES_METHOD	
DM	DM	PINES_METHOD	
GMD	GMD	PINES_METHOD	
GMO	GMO	PINES_METHOD	
T	T_ACCEL	PINES_METHOD	
VM	VM	PINES_METHOD	
$\underline{R}$	X_{1to3}	PINES_METHOD	
$\underline{V}$	X_{4to6}	PINES_METHOD	
ATM	ATFL_OV	AVERAGE_G_INTEGRATOR	
DM	DFL_AVG	AVERAGE_G_INTEGRATOR	
IGD	GM_DEG_LOW	AVERAGE_G_INTEGRATOR	
IGO	GM_ORD_LOW	AVERAGE_G_INTEGRATOR	
$\underline{R}$	R_AV	AVERAGE_G_INTEGRATOR	
$\underline{\underline{T}}$	T_STATE	AVERAGE_G_INTEGRATOR	
$\underline{V}$	V_AV	AVERAGE_G_INTEGRATOR	
VM	VFLTV_PRED	AVERAGE_G_INTEGRATOR	
ATM	ATFL_OV	AVERAGE_G_INTEGRATOR	
DM	DFL_AVG	AVERAGE_G_INTEGRATOR	
IGD	GM_DEG_LOW	AVERAGE_G_INTEGRATOR	
IGO	GM_ORD_LOW	AVERAGE_G_INTEGRATOR	
$\underline{R}$	R_AV	AVERAGE_G_INTEGRATOR	
$\underline{\underline{T}}$	T_IMU	AVERAGE_G_INTEGRATOR	
$\underline{V}$	V_AV	AVERAGE_G_INTEGRATOR	
VM	VFLTV_PRED	AVERAGE_G_INTEGRATOR	

TABLE 4.2.4.1.1.- ACCEL_ONORBIT INPUT/OUTPUT.- Continued

Variable Name	Input Source	Output Destination
ABM(K = 1 to 2)	**	
ALT_L	**	
BM	**	
C	**	
CBD1	**	
CBD2	**	
CBM2	**	
CDA	**	
CDEC1	SOLAR_EPHEM	
C_DENSEA	**	
CDF	**	
CDN	**	
CDS	**	
COS_LAG	**	
COS_SOL_RA	SOLAR_EPHEM	
†	EARTH_FIXED_TO_M50 COORD(T)	
EARTH_MU	**	
EARTH_RADIUS_GRAV	**	
EXP_SHAPE_FACTOR	**	
GDIE	**	

*See P.F. I/O tables of section 4.2

**Initialization parameters, see section 4.7

†The function value of EARTH_FIXED_TO_M50_COORD(T) is used

TABLE 4.2.4.1.1.- ACCEL_ONORBIT INPUT/OUTPUT.- Continued

Variable Name	Input Source	Output Destination
KFACTOR	**,*	
M_BODYM50	NAV_ONORBIT_RENDEZVOUS	
NAV_CURR_ORB_MASS	ONORBIT_REND_NAV_ SEQUENCER,NAV_ONORBIT_ RENDEZVOUS	
PRED_ORB_AREA	***	
PRED_ORB_CD	***	
PRED_ORB_MASS	***	
REF_ORB_AREA	**	
RREF	**	
S	**	
SDEC	SOLAR_EPHEM	
SHUTTLE_FILTER_FLAG	**	
SIN_LAG	**	
SIN_SOL_RA	SOLAR_EPHEM	
TARGET_AREA	**	
TARGET_CD	**	
TARGET_MASS	**	
TFOFF	**,*	

*See P.F. I/O table of sections 4.1, 4.2, 4.3, 4.4 and 4.5
**Initialization parameters, see section 4.7
***See Onorbit precision state prediction P.F. I/O

TABLE 4.2.4.1.1.- ACCEL_ONORBIT INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
TFON	**,*	
UNMOD_ACC_BIAS	OPS_2_OR_8_INITIALIZE, REND_NAV_EXIT, U_A_BIAS_AND_COVINIT, REND_BIAS_AND_COV_PROP, REND_NAV_FILTER,*	
VFORCE	**,*	
ZONAL	**	
†		AVERAGE_G_INTEGRATOR, ONORBIT_SV_INTERP, PINES_METHOD, REND_COV_INIT, SUPER_G
ALT_SS		***
D _SS		***
G _CENTRAL		PINES_METHOD
VENT_SS		***
R		H_ELLIPSOID,V _REL
V		V _REL
T		EARTH_FIXED_TO_M50_ COORD (T),SOLAR_EPHEM
†	V _REL	
†	H_ELLIPSOID	

*See P.F. I/O tables of sections 4.1, 4.2, 4.3, 4.4 and 4.5
**Initialization parameters, see section 4.7
***See P.F. I/O tables of section 4.2
†Only the value of the function is passed

4.2.4.1.1.1. Altitude above the reference ellipsoid (H_ELLIPSOID): Altitude above the Earth's reference ellipsoid in M50 coordinates will be determined using the H_ELLIPSOID function.

A. Detailed Requirements.- Altitude above the Earth's reference ellipsoid will be calculated each time the statement H_ELLIPSOID (R) is encountered. Here R is the vehicle M50 position vector. The value of altitude above the Earth's reference ellipsoid will be determined using the following:

$$H\_ELLIPSOID(\underline{R}) = |\underline{R}| - \frac{(1 - ELLIPT)\ EARTH\_RADIUS\_EQUATOR}{\sqrt{1 + ((1-ELLIPT)^2 - 1)\ (1 - (UNIT(R) \cdot EARTH\ POLE)^2)}} \qquad \text{F3 F4}$$

where ELLIPT, EARTH_RADIUS_EQUATOR, and EARTH_POLE are constants defined in section 4.7 (Initialization parameters).

B. Interface Requirements.- The input and output data are shown in table 4.2.4.1.1.1. It should be noted that only the value of H_ELLIPSOID is passed.

C. Processing Requirements.- The H_ELLIPSOID function will be processed each time the function name is encountered with a suitable expression for the argument such as: H_ELLIPSOID (R). The H_ELLIPSOID function is used by the following.

ACCEL_ONORBIT

D. Constraints.- None

E. Supplementary Information.- A suggested implementation of the H_ELLIPSOID function in the form of a detailed flowchart may be found in Appendix B under the name:

H_ELLIPSOID FUNCTION

F3 This equation shall be protected against division by zero (Reference 3.6-3).

F4 This equation shall be protected against square roots of a negative number (Reference 3.6-4).

TABLE 4.2.4.1.1.1.- H_ELLIPSOID INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
R	R	ACCEL_ONORBIT (ONORBIT_DENSITY CODE)	

TABLE 4.2.4.1.1.1.- H_ELLIPSOID INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
EARTH_POLE	**	
EARTH_RADIUS_EQUATOR	**	
ELLIPT	**	
†		ACCEL_ONORBIT

**Initialization parameters, see section 4.7

†The value of H_ELLIPSOID only is passed to output destination

4.2.4.1.1.2 Earth relative velocity computation (V_REL): The velocity vector relative to the Earth but expressed in M50 coordinates will be determined using the V_REL function.

A. Detailed Requirements.- The function is activated whenever the statement V_REL(V,R) is encountered. The function arguments are the M50 velocity vector (V) and M50 position vector (R). The vector V_REL is determined by the following:

V_REL(V,R) = V - EARTH_RATE (EARTH_POLE x R)

where EARTH_RATE and EARTH_POLE are constants defined in section 4.7 (Initialization parameters).

B. Interface Requirements.- The input and output data are shown in table 4.2.4.1.1.2. It should be noted that only the value of V_REL is passed.

C. Processing Requirements.- The V_REL function will be processed each time the function name is encountered with a suitable expression for the arguments such as V_REL(V,R). The V_REL function is used by ACCEL_ONORBIT

D. Constraints.- None

E. Supplementary Information.- A suggested implementation of the V_REL function in the form of a detailed flowchart may be found in Appendix B under the name:

V_REL FUNCTION

TABLE 4.2.4.1.1.2.- V_REL INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
R	R	ACCEL_ONORBIT (ACCEL_ONORBIT_DRAG CODE)	
V	V	ACCEL_ONORBIT (ACCEL_ONORBIT_DRAG CODE)	

TABLE 4.2.4.1.1.2.- V_REL INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
EARTH_POLE	**	
EARTH_RATE	**	
†		ACCEL_ONORBIT (ACCEL_ONORBIT_DRAG CODE)

**Initialization parameters, see section 4.7
†V_REL value only is passed to output destinations

4.2.4.1.1.3 Solar ephemeris model (SOLAR_EPHEM)

Output from the solar ephemeris model will provide sine and cosine functions of the solar right ascension and declination. These outputs are to be used in the acceleration models (ACCEL_ONORBIT) for determination of atmospheric density and in position and velocity state propagation (ONORBIT_REND_R_V_ STATE_PROP) for computation of a unit vector from earth center to the center of the sun in M50 coordinates to be available for the universal pointing processing principal function.

A. Detailed Requirements.

The solar ephemeris subfunction shall be invoked whenever the following statement is encountered:

CALL: SOLAR_EPHEM

IN LIST: T

OUT LIST: SDEC,CDEC1,COS_SOL_RA,SIN_SOL_RA

where:

CDEC1 = cosine of the solar declination
COS_SOL_RA = cosine of the solar right ascension
SDEC = sine of the solar declination
SIN_SOL_RA = sine of the solar right ascension
T = time of desired computation

The SOLAR_EPHEM subfunction will initiate the following calculations in the indicated order:

1. The longitude of the Sun in M50 coordinates, LOS, will be determined as

   LOS = LOS_ZERO + T LOS_R - LOC SIN(T OMEG_C + PHASE_C). F9

   Here, LOS_ZERO,LOS_R,LOS,OMEG_C and PHASE_C are design dependent parameters (see section 4.7).

2. The sine and cosine of the solar declination SDEC and CDEC1 are calculated by

   SDEC = LOSK3 SIN(LOS) F9
   CDEC1 = SQRT(1. - SDEC^2) F4

F4 This operation shall be protected against square roots of a negative number (Reference 3.6-4).

F9 This equation shall be protected against return value of sine or cosine with magnitude greater than unity (Reference 3.6-9).

3. The cosine and sine of the solar right ascension are determined by

COS_SOL_RA = COS(LOS)/CDEC1 F9,F3
SIN_SOL_RA = LOSK1 SIN(LOS)/CDEC1 F3

LOSK1 and LOSK3 are design dependent parameters (see section 4.7).

B. Interface Requirements. The input and output required of the SOLAR_EPHEM subfunction are listed in table 4.2.4.1.1.3.

C. Processing Requirements. The SOLAR_EPHEM subfunction shall be called by the acceleration subfunction program ACCEL_ONORBIT and the ONORBIT_REND_R_V_STATE_PROP.

D. Constraints. None

E. Supplementary Information. A suggested implementation of the SOLAR_EPHEM function in the form of a detailed flowchart may be found in Appendix B under the name:

SOLAR_EPHEM

F3 This equation shall be protected against division by zero Reference 3.6-3).

F9 This equation shall be protected against return value of sine or cosine with magnitude greater than unity (Reference 3.6-9).

TABLE 4.2.4.1.1.3- SOLAR_EPHEM INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
T	T	ACCEL_ONORBIT	
T	T_CURRENT_FILT	ONORBIT_REND_R_V_STATE_PROP	
CDEC1	CDEC1		ACCEL_ONORBIT
COS_SOL_RA	COS_SOL_RA		ACCEL_ONORBIT
SDEC	SDEC		ACCEL_ONORBIT
SIN_SOL_RA	SIN_SOL_RA		ACCEL_ONORBIT
CDEC1	CDEC1		ONORBIT_REND_R_V_STATE_PROP
COS_SOL_RA	COS_SOL_RA		ONORBIT_REND_R_V_STATE_PROP
SDEC	SDEC		ONORBIT_REND_R_V_STATE_PROP
SIN_SOL_RA	SIN_SOL_RA		ONORBIT_REND_R_V_STATE_PROP

TABLE 4.2.4.1.1.3.(Concluded)- SOLAR_EPHEM INPUT/OUTPUT

Variable Name	Input Source	Output Destination
LOC	**	
LOSK1	**	
LOSK3	**	
LOS_R	**	
LOS_ZERO	**	
OMEG_C	**	
PHASE_C	**	

** Initialization parameters, see section 4.7

4.2.5 Unmodeled Acceleration State and Covariance Matrix Propagation (REND_BIAS_AND_COV_PROP)

The unmodeled acceleration state and covariance matrix propagation subfunction will be executed only during the rendezvous navigation phase and will be scheduled by the Onorbit/Rendezvous control subfunction, ONORBIT_RENDEZVOUS_NAV, at an N_CYCLE multiple of the state propagation rate. The rate at which this subfunction is executed will be called the filter subcycle rate. The following tasks will be performed during each execution.

- If IMU sensed accelerations were used at any time during the previous covariance propagation interval but are not used at all during the current propagation interval, or if IMU sensed accelerations were not used at all during the previous covariance propagation interval but are used at some time during the current propagation interval, then the unmodeled acceleration bias vector as well as the unmodeled acceleration portions of the covariance matrix are reinitialized.

- The unmodeled acceleration bias vector will be propagated to current time as an exponentially correlated random variable (ECRV).

- A 13-by-13 covariance matrix, E, shall be propagated to current time by using a state transition matrix.

- Additive process noise will be incorporated into the covariance matrix to account for unmodeled state and dynamic errors.

- Finally, a flag (NAV_MEAS) is set to ON, to allow measurement incorporation, only if the magnitude of the IMU sensed acceleration vector is below a design dependent threshold.

The 13-by-13 state transition matrix ϕ is mathematically defined as the partial of the current state with respect to the previous state. For propagation of the filter vehicle covariance matrix, ϕ will be formulated as shown below.

$$\phi = \left[\begin{array}{cc|c} \multicolumn{2}{c|}{} & 0_{3x4} \\ \cline{3-3} \multicolumn{2}{c|}{\text{PHI}_{6x9}} & \\ \multicolumn{2}{c|}{} & 0_{3x4} \\ \hline 0_{3x6} & \multicolumn{1}{|c|}{\text{PHI\_UNMOD\_ACC}} & 0_{3x4} \\ \hline 0_{4x6} & \multicolumn{1}{|c|}{0_{4x3}} & \text{PHI\_BIAS} \end{array}\right]$$

State Transition Matrix Composition

The submatrices $\Phi_{1 \text{ to } 9,\ 10 \text{ to } 13}$, $\Phi_{7 \text{ to } 9,\ 1 \text{ to } 6}$, and $\Phi_{10 \text{ to } 13,\ 1 \text{ to } 9}$ are null matrices.

The state noise matrix, NOISE, shall be formulated as shown below. The matrix is to be used to account for unmodeled state errors, unmodeled acceleration errors, and sensor bias errors.

$$\text{NOISE} = \left[\begin{array}{c|c|c} S_{6x6} & 0_{6x3} & 0_{6x4} \\ \hline 0_{3x6} & S\_UNMOD\_ACC & 0_{3x4} \\ \hline 0_{4x6} & 0_{4x3} & \underline{S}\_BIAS \end{array}\right]$$

State Noise Matrix Composition

The submatrices $\text{NOISE}_{1 \text{ to } 6,\ 7 \text{ to } 13}$, $\text{NOISE}_{7 \text{ to } 9,\ 10 \text{ to } 13}$, $\text{NOISE}_{7 \text{ to } 9,1 \text{ to } 6}$, and $\text{NOISE}_{10 \text{ to } 13,\ 1 \text{ to } 9}$ are null matrices.

The above formulations for both the state transition and the state noise transition matrices were made for clarity and code efficiency.

The unmodeled acceleration state and covariance matrix propagation subfunction will also propagate the unmodeled acceleration bias states as ECRV's.

A. Detailed Requirements.- The following steps shall be performed:

1. If there has been a change in the value of the COV_PWRD_FLT flag (COV_PWRD_FLT = ON indicates that the sensed delta velocity has exceeded a threshold value since the previous covariance propagation cycle) since the last filter cycle, then reinitialize the unmodeled acceleration states.

   a. EXECUTE: UNMOD_ACC_REINIT CODE

      (1) Zero the unmodeled acceleration state portions of the covariance matrix.

      $$E_{7 \text{ to } 9,\ 1 \text{ to } 13} = 0.0$$

      $$E_{1 \text{ to } 13,\ 7 \text{ to } 9} = 0.0$$

      (2) Set up the time constants and variances as a function of the COV_PWRD_FLT flag.

      - If the sensed delta velocity is above the threshold

(COV_PWRD_FLT = ON), then
TAU_UNMOD_ACC_COV = TAU_U_A_PWRD_FLT
VAR_UNMOD_ACC = VAR_U_A_PWRD_FLT
COV_ACCEL_UVW_INIT = COV_U_A_PWRD_FLT

- If the sensed delta velocity is less than or equal to the threshold (COV_PWRD_FLT = OFF), then

TAU_UNMOD_ACC_COV = TAU_U_A_COAST
VAR_UNMOD_ACC = VAR_U_A_COAST
COV_ACCEL_UVW_INIT = COV_U_A_COAST

(3) Reinitialize the unmodeled acceleration biases and the unmodeled acceleration slots of the covariance matrix.

- If the Shuttle is the filter vehicle (SHUTTLE_FILTER_FLAG = ON), then

CALL: U_A_BIAS_AND_COVINIT
IN LIST: R _FILT, V _FILT

(Refer to section 4.2.5.1 for detailed requirements.)

If the target is the filter vehicle (SHUTTLE_FILTER_FLAG = OFF), then

CALL: U_A_BIAS_AND_COVINIT
IN LIST: R _TV, V _TV

(Refer to section 4.2.5.1 for detailed requirements.)

(4) Save the last value of COV_PWRD_FLT.
COV_PWRD_FLT_LAST = COV_PWRD_FLT

2. Compute the change in time since the last covariance propagation subcycle

DT_COV = T_CURRENT_FILT - T_COV_LAST

where T_COV_LAST is the time of the last covariance matrix propagation subcycle. Also TOT_ACC, the total acceleration vector for the Shuttle, shall be calculated for use in the state vector interpolation routine

TOT_ACC = G _NEW + DV_COV/DT_COV F3

where DV_COV is the total accumulated sensed delta velocity since the time of the last covariance matrix propagation subcycle and G _NEW is the Shuttle acceleration vector.

F3 This equation shall be protected against division by zero (Reference 3.6-3).

3. The SHUTTLE_FILTER_FLAG is tested to determine which vehicle is used to calculate the 6-by-6 submatrix, PHI$_{1 \text{ to } 6,\ 1 \text{ to } 6}$ ($\phi_{1 \text{ to } 6,\ 1 \text{ to } 6}$), composed of the filter vehicle position and velocity portion of the total state transition matrix. This submatrix is calculated by calling the mean conic partials subfunction (described in section 4.2.5.2).

   a. If the Shuttle is the filter vehicle

      (SHUTTLE_FILTER_FLAG = ON)

      (1) CALL: MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6x6

          IN LIST: R _LAST, V _LAST, TOT_ACC_LAST, R _FILT, V _FILT, TOT_ACC, DT_COV

          OUT LIST: PHI$_{1 \text{ to } 6,\ 1 \text{ to } 6}$

      (2) Compute a UVW to mean of 1950 transformation matrix for the Shuttle for use in the state noise formulation.

          MAT = UVW_TO_M50 (R _FILT,V _FILT)

          (Refer to section 4.10.10 for a detailed description.)

          The current inertial Orbiter vehicle acceleration shall be saved for the next filter subcycle

          TOT_ACC_LAST = TOT_ACC

   b. If the target is the filter vehicle

      (SHUTTLE_FILTER_FLAG = OFF)

      (1) CALL: MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6x6

          IN LIST: R _TV_LAST, V _TV_LAST, G _TV_LAST, R _TV, V _TV, G _TV, DT_COV

          OUT LIST: PHI$_{1 \text{ to } 6,\ 1 \text{ to } 6}$

      (2) Compute a UVW to mean of 1950 transformation matrix for the target vehicle for use in the state noise formulation.

          MAT = UVW_TO_M50 (R _TV,V _TV)

          (Refer to section 4.10.10 for a detailed description.)

          The current inertial target vehicle acceleration vector shall be saved for the next filter subcycle.

          G _TV_LAST = G _TV

4. Calculate a scalar, PHI_UNMOD_ACC, which will be stored into the diagonal elements of the unmodeled acceleration bias portion of the state transition matrix ($\phi_{7 \text{ to } 9,\ 7 \text{ to } 9}$)

   PHI_UNMOD_ACC = exp(-DT_COV/TAU_UNMOD_ACC_COV) F2

   A value used to compute the remainder of the PHI matrix is also calculated.

   DIAG = TAU_UNMOD_ACC_COV (1. - PHI_UNMOD_ACC)

   where DIAG is in mean of 1950 coordinates.

5. The unmodeled acceleration bias states shall be propagated as ECRV's as follows:

   UNMOD_ACC_BIAS = exp(-DT_COV/TAU_UNMOD_ACC_STATE · UNMOD_ACC_BIAS) F2

6. Calculate a 3-by-3 diagonal matrix S_UNMOD_ACC (NOISE$_{7 \text{ to } 9,7 \text{ to } 9}$) composed of the unmodeled acceleration errors where

   S_UNMOD_ACC$_{I,I}$ = VAR_UNMOD_ACC$_I$(1. - PHI_UNMOD_ACC^2) (for I = 1 to 3)
   S_UNMOD ACC = MAT S_UNMOD_ACC MAT^T

   and MAT is the UVW to mean of 1950 coordinate transformation matrix for the filter vehicle.

7. Calculate a 6-by-3 submatrix, PHI$_{1 \text{ to } 6,7 \text{ to } 9}$ ($\phi_{1 \text{ to } 6,7 \text{ to } 9}$), composed of two 3-by-3 diagonal submatrices that correlate the position and velocity with the estimated unmodeled accelerations where

   PHI$_{I+3,I+6}$ = DIAG
   PHI$_{I,I+6}$ = TAU_UNMOD_ACC_COV (DT_COV-DIAG)

   (for I = 1 to 3)

8. Calculate a vector, PHI_BIAS, which will be stored into the diagonal elements of the sensor bias portion of the total state transition matrix ($\phi_{10 \text{ to } 13,10 \text{ to } 13}$).

   PHI_BIAS$_I$ = exp (-DT_COV/TAU_SENS$_I$)

   (for I = 1 to 4) F2

F2 This equation shall be protected against floating point underflow (Ref. 3.6-2).

9. Calculate a vector, S_BIAS, which will be stored into the diagonal elements of the sensor bias portion of the total state noise matrix ($NOISE_{10\text{ to }13,10\text{ to }13}$)

   $S\_BIAS_I = VAR\_SENS_I\,(1. - PHI\_BIAS_I^2)$

   (for I = 1 to 4)

10. Propagate the covariance matrix

    EXECUTE: PHI_E_PHI_TRANSPOSE CODE

    a.
    $$E\,\phi^T \begin{cases} TEMP_{1\text{ to }13,1\text{ to }6} = E_{1\text{ to }13,1\text{ to }9}\,PHI^T \\ TEMP_{J,I+6} = E_{J,I+6}\ \ PHI\_UNMOD\_ACC \text{ (for I=1 to 3,J=1 to 13)} \\ TEMP_{J,K+9} = E_{J,K+9}\ PHI\_BIAS_K \text{ (for J=1 to 13,K=1 to 4)} \end{cases}$$

    b.
    $$\phi\,E\,\phi^T \begin{cases} E_{1\text{ to }6,1\text{ to }13} = PHI\ TEMP_{1\text{ to }9,1\text{ to }13} \\ E_{I+6,J} = PHI\_UNMOD\_ACC\ TEMP_{I+6,J} \text{ (for I=1 to 3,J=1 to 13)} \\ E_{K+9,J} = PHI\_BIAS_K\ TEMP_{K+9,J} \text{ (for J=1 to 13,K=1 to 4)} \end{cases}$$

    c.
    $$\begin{array}{c}\phi\,E\,\phi^T \\ +\ NOISE\end{array} \begin{cases} E_{7\text{ to }9,7\text{ to }9} = E_{7\text{ to }9,7\text{ to }9} + S\_UNMOD\_ACC \\ E_{K+9,K+9} = E_{K+9,K+9} + S\_BIAS_K \text{ (for K=1 to 4)} \end{cases}$$

    d. If IMU sensed acceleration data have been used (COV_PWRD_FLT=ON) since the last covariance matrix propagation subcycle, then

       (1) Reset the COV_PWRD_FLT flag

       COV_PWRD_FLT = OFF and

       EXECUTE: PWRD_FLT_NOISE CODE

       - Calculate a 6-by-6 matrix S ($NOISE_{1\text{ to }6,1\text{ to }6}$) composed of misalinement errors:

         $S_{4,4} = DV\_COV_3^2 + DV\_COV_2^2$

         $S_{5,5} = DV\_COV_1^2 + DV\_COV_3^2$

         $S_{6,6} = DV\_COV_1^2 + DV\_COV_2^2$

         $S_{4,5} = -DV\_COV_1\ DV\_COV_2$

         $S_{4,6} = -DV\_COV_1\ DV\_COV_3$

$S_{5,6} = -DV\_COV_2\ DV\_COV_3$

$S_{6,5} = S_{5,6}$

$S_{6,4} = S_{4,6}$

$S_{5,4} = S_{4,5}$

$S_{4\ to\ 6,4\ to\ 6} = (VAR\_IMU\_ALIGN)\ S_{4\ to\ 6,4\ to\ 6}$

$S_{1\ to\ 3,4\ to\ 6} = 0.5\ (DT\_COV)\ S_{4\ to\ 6,4\ to\ 6}$

$S_{1\ to\ 3,1\ to\ 3} = 0.5\ (DT\_COV)\ S_{1\ to\ 3,4\ to\ 6}$

(2) Add the powered flight noise to the covariance matrix

$E_{1to6,1to6} = E_{1to6,1to6} + S$

e. Test the flag NOISY_NAV_MEAS to see if the sensed contact acceleration has exceeded the measurement threshold

(1) If NOISY_NAV_MEAS = ON

- Reset NOISY_NAV_MEAS to OFF for the next covariance matrix propagation subcycle.
- Reset the measurement ACCEPT/REJECT counters. (See section 4.1.2.2.1.2)

CALL: DISPLAY_COUNT_INIT

(2) If NOISY_NAV_MEAS = OFF

- Set the flag NAV_MEAS to ON so that the sensor measurements may be incorporated on the current filter subcycle.

11. As a final step in the covariance propagation, the entire 13-by-13 covariance matrix shall be made symmetric:

$E_{J,I} = E_{I,J}$ (for I = 1to12, J = I+1to13)

B. Interface Requirements.- The input and output data are shown in table 4.2.5.

C. Processing Requirements.- The subfunction is called by NAV_ONORBIT_RENDEZVOUS (section 4.2.1).

D. Constraints.- Either uplinked or mission dependent data are to be used for initialization. The propagated matrix must remain symmetric.

E. Supplementary Information.- A suggested inplementation in the form of detailed flowcharts can be found in Appendix B under the following names:

PHI_E_PHI_TRANSPOSE CODE
PWRD_FLT_NOISE CODE
REND_BIAS_AND_COV_PROP
UNMOD_ACC_REINIT CODE

TABLE 4.2.5.- REND_BIAS_AND_COV_PROP INPUT/OUTPUT

Variable Name	Input Source	Output Destination
COV_ACCEL_UVW_INIT		U_A_BIAS_AND_COVINIT
COV_PWRD_FLT	NAV_ONORBIT_RENDEZVOUS, *	
COV_PWRD_FLT_LAST	*	
COV_U_A_COAST	**	
COV_U_A_PWRD_FLT	**	
DV_COV	NAV_ONORBIT_RENDEZVOUS, COV_LAST_RESET,*	
E	COVINIT_UVW,SETUP, REND_NAV_FILTER, REND_COV_INIT,U_A_BIAS_AND_COVINIT,*	REND_NAV_FILTER,*
G _NEW	ONORBIT_REND_R_V_STATE_PROP,SUPER_G	
G _TV	ONORBIT_REND_R_V_STATE_PROP	MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6x6
G _TV_LAST	REND_COV_INIT,*	MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6x6
NOISY_NAV_MEAS	NAV_ONORBIT_RENDEZVOUS, *	
$PHI_{1to6,1to6}$	MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6x6	
R _FILT	ONORBIT_REND_R_V_STATE_PROP	MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6x6, UVW_TO_M50,U_A_BIAS_AND_COVINIT

*Onorbit/Rendezvous principal function, see section 4.2

**Initialization parameters, see section 4.7

TABLE 4.2.5.- REND_BIAS_AND_COV_PROP INPUT/OUTPUT.- Continued

Variable Name	Input Source	Output Destination
R _LAST	COV_LAST_RESET,*	MEAN_CONIC_PARTIAL_ TRANSITION_MATRIX_6x6
R _TV	ONORBIT_REND_R_V_STATE_ PROP	MEAN_CONIC_PARTIAL_ TRANSITION_MATRIX_6x6, UVW_TO_M50,U_A_BIAS_ AND_COVINIT
R _TV_LAST	COV_LAST_RESET,*	MEAN_CONIC_PARTIAL_ TRANSITION_MATRIX_6x6
SHUTTLE_FILTER_FLAG	**	
TAU_SENS	SETUP,**	
TAU_U_A_COAST	**	
TAU_U_A_PWRD_FLT	**	
TAU_UNMOD_ACC_COV	*	
TAU_UNMOD_ACC_STATE	**	
T_COV_LAST	COV_LAST_RESET,*	
T_CURRENT_FILT	NAV_ONORBIT_RENDEZVOUS	
TOT_ACC_LAST	REND_COV_INIT,*	MEAN_CONIC_PARTIAL_ TRANSITION_MATRIX_6x6
UNMOD_ACC_BIAS	U_A_BIAS_AND_COVINIT, REND_NAV_FILTER,*	REND_NAV_FILTER,ACCEL_ ONORBIT,NAV_ONORBIT_ RENDEZVOUS

*Onorbit/Rendezvous principal function, see section 4.2
**Initialization parameters, see section 4.7

TABLE 4.2.5.- REND_BIAS_AND_COV_PROP INPUT/OUTPUT.- Continued

Variable Name	Input Source	Output Destination
VAR_IMU_ALIGN	**	
VAR_SENS	SETUP	
VAR_U_A_COAST	**	
VAR_U_A_PWRD_FLT	**	
VAR_UNMOD_ACC	*	
V_FILT	ONORBIT_REND_R_V_STATE_PROP	MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6x6, UVW_TO_M50,U_A_BIAS_AND_COVINIT
V_LAST	COV_LAST_RESET,*	MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6x6
V_TV	ONORBIT_REND_R_V_STATE_PROP	MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6x6, UVW_TO_M50,U_A_BIAS_AND_COVINIT
V_TV_LAST	COV_LAST_RESET,*	MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6x6
†	UVW_TO_M50	
DT_COV		MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6x6, ONORBIT_SV_INTERP,*

*Onorbit/Rendezvous principal function, see section 4.2

**Initialization parameters, see section 4.7

†Value returned from the function

TABLE 4.2.5.- REND_BIAS_AND_COV_PROP INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
NAV_MEAS		NAV_ONORBIT_RENDEZVOUS
TOT_ACC		MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6x6,REND_NAV_INTERP,*

*Onorbit/Rendezvous principal function, see section 4.2

4.2.5.1 Unmodeled Acceleration Bias and Covariance Initialization (U_A_BIAS_AND_COVINIT)

The initialization subfunction initializes the unmodeled acceleration bias states and the corresponding elements of the covariance matrix whenever the covariance matrix is reinitialized or whenever the unmodeled acceleration bias and covariance propagation subfunction has detected a change in the value of the flag COV_PWRD_FLT, which indicates whether the magnitude of the IMU sensed acceleration has exceeded the threshold value DA_THRESHOLD on the current covariance propagation subcycle.

A. Detailed Requirements. This subfunction is called with the following internal variables in the IN LIST:

IN LIST: $\underline{R}$,$\underline{V}$

where $\underline{R}$ and $\underline{V}$ are the input position and velocity vectors of the filter vehicle in mean of 1950 coordinates.

The following steps shall be performed (in the order indicated):

1. Zero the unmodeled acceleration bias states

   $\underline{\text{UNMOD\_ACC\_BIAS}} = \underline{0}$.

2. Compute the UVW to mean of 1950 coordinate transformation matrix

   M_UVW_M50 = UVW_TO_M50 ($\underline{R}$,$\underline{V}$)

   (refer to section 4.10.10 for a detailed description).

3. Initialize the unmodeled acceleration bias slots of the covariance matrix to initial UVW values and transform to mean of 1950 coordinates,

   $E_{I+6,I+6} = \text{COV\_ACCEL\_UVW\_INIT}_I$ (for I = 1 to 3)

   $E_{7to9,7to9} = \text{M\_UVW\_M50}\ E_{7to9,7to9}\ \text{M\_UVW\_M50}^T$

B. Interface Requirements. The input and output data are shown in table 4.2.5.1.

C. Processing Requirements. This subfunction is called by

REND_BIAS_AND_COV_PROP (section 4.2.5) and by
COVINIT_UVW (section 4.1.2.2.1.1)

D. Constraints. Mission dependent (ILOAD) data are used for initialization.

E. Supplementary Information. A suggested implementation in the form of a detailed flowchart may be found in Appendix B under the name of U_A_BIAS_AND_COVINIT.

TABLE 4.2.5.1.- U_A_BIAS_AND_COVINIT INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
R	R	COVINIT_UVW	
V	V	COVINIT_UVW	
R	R_FILT	REND_BIAS_AND_COV_PROP	
V	V_FILT	REND_BIAS_AND_COV_PROP	
R	R_TV	REND_BIAS_AND_COV_PROP	
V	V_TV	REND_BIAS_AND_COV_PROP	

TABLE 4.2.5.1.- U_A_BIAS_AND_COVINIT INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
COV_ACCEL_UVW_INIT	REND_NAV_INIT, REND_BIAS_AND_COV_PROP,*	
†	UVW_TO_M50	
$E_{7to9,7to9}$		REND_BIAS_AND_COV_PROP, REND_NAV_FILTER,*,***
M_UVW_M50		COVINIT_UVW
$\underline{R}$		UVW_TO_M50
UNMOD_ACC_BIAS		REND_BIAS_AND_COV_PROP, ACCEL_ONORBIT, REND_NAV_FILTER,***, NAV_ONORBIT_RENDEZVOUS
$\underline{V}$		UVW_TO_M50

†Value returned from the function

*Onorbit/Rendezvous principal function, see section 4.2

***Onorbit/Rendezvous Nav Sequencer principal function, see section 4.1

4.2.5.2 Position-Velocity Submatrix of the State Transition Matrix (MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6)

This subfunction computes a 6 by 6 submatrix (PHI_MC) of a larger state transition matrix for use in propagating the covariance matrix. This subfunction is also invoked by the rendezvous navigation interpolation subfunction for use in calculating the partial derivative vector.

A. Detailed Requirements. This subfunction is called with the following internal variables in the INLIST and OUTLIST:

INLIST: $\underline{R}$_ONE, $\underline{V}$_ONE, $\underline{G}$_ONE, $\underline{R}$_TWO, $\underline{V}$_TWO, $\underline{G}$_TWO, DELTIM

OUTLIST: PHI_MC

where

$\underline{R}$_ONE	initial position vector
$\underline{V}$_ONE	initial velocity vector
$\underline{G}$_ONE	initial total acceleration vector
$\underline{R}$_TWO	final position vector
$\underline{V}$_TWO	final velocity vector
$\underline{G}$_TWO	final total acceleration vector
DELTIM	time step

The following steps shall be performed (in the order indicated):

1. A formulation is used that assumes that a mean conic section may be used to describe the path taken between the initial position and velocity ($\underline{R}$_ONE and $\underline{V}$_ONE) at initial total acceleration ($\underline{G}$_ONE) and the final position and velocity ($\underline{R}$_TWO and $\underline{V}$_TWO) at final total acceleration ($\underline{G}$_TWO) over a time step DELTIM.

   The following local variables are computed:

$$\text{R\_ONE\_INV} = 1./|\underline{R}\text{\_ONE}| \qquad \text{F3}$$
$$\text{R\_TWO\_INV} = 1./|\underline{R}\text{\_TWO}| \qquad \text{F3}$$
$$\text{D\_ONE} = \underline{R}\text{\_ONE} \cdot \underline{V}\text{\_ONE}$$
$$\text{D\_TWO} = \underline{R}\text{\_TWO} \cdot \underline{V}\text{\_TWO}$$

2. The ergodic semi-major axis SMA, is computed by using the average of the reciprocal of the semi-major axis derived from combination of the respective vis-viva computations at the initial and final orbital states, and is given by:

$$\text{SMA} = 1./\left(\text{R\_ONE\_INV} + \text{R\_TWO\_INV} - (\underline{V}\text{\_ONE} \cdot \underline{V}\text{\_ONE} + \underline{V}\text{\_TWO} \cdot \underline{V}\text{\_TWO})/(2.\ \text{EARTH\_MU})\right) \qquad \text{F3}$$

F3 This equation shall be protected against division by zero (Reference 3.6-3).

where EARTH_MU is the Earth's gravitational constant. The Stumpff constant, C1, predicated on the mean conic semi-major axis, is computed by:

C1 = SQRT (SMA)/SQR_EMU F3,F4

3. The F_AND_G subfunction is then invoked to compute the F and G functions, the derivatives of F and G, as well as the auxiliary variables S0, S1, S2 and S3. These quantities are fundamental to the computation of the mean conic transition matrix.

CALL: F_AND_G

IN LIST: SMA,DELTIM,C1,$\underline{R}$ _ONE,R_ONE_INV,R_TWO_INV,$\underline{V}$ _ONE,D_ONE,D_TWO

OUT LIST: F,G,FDOT,GDOT,S0,S1,S2,S3,$\underline{R}$ _TWO,R_TWO_INV,THETA

For this case, $\underline{R}$ _TWO and R_TWO_INV are not updated since R_TWO_INV is supplied as a non-zero quantity. However, THETA, the eccentric anomaly angle, is generated as an output in any case.

4. Compute auxiliary constants:

FM1 = F-1

GDM1 = GDOT-1

S1 = C1 S1

C2 = $C1^2$

CONST = C2 (EARTH_MU C2 (3. S3 C1-S1 S2) + G S2)

S2 = C2 S2

which represent common functionals and Stumpff series summations for circular or elliptical orbits.

5. The partial derivatives are now calculated. The equation for the partial derivatives are written algebraically for clarity, with R_0 representing $\underline{R}$ _ONE, R representing $\underline{R}$ _TWO, $\dot{R}_0$ representing $\underline{V}$ _ONE, $\dot{R}$ representing $\underline{V}$ _TWO, $\ddot{R}_0$ representing $\underline{G}$ _ONE, $\ddot{R}$ representing $\underline{G}$ _TWO, f representing F, g representing G, $\dot{f}$ representing FDOT, $\dot{g}$ representing GDOT and U representing CONST, as well as having lower case letters

---

F3 This equation shall be protected against division by zero (Reference 3.6-3).

F4 This equation shall be protected against square roots of a negative number (Reference 3.6-4).

representing the scalar magnitude of the respective upper case letter vectors.

$$\text{PHI\_MC}_{1\text{ to }3,\ 1\text{ to }3} = \frac{\partial R}{\partial R_o} = -\left[\frac{\dot{f}\ S_1 + (f-1)/r_o}{r_o}\right] R\ R_o{}^T$$

$$-\ \dot{f}\ S_2\ R\ \dot{R}_o{}^T \qquad \text{F3}$$

$$+\ \frac{(f-1)S_1}{r_o}\dot{R}\ R_o{}^T + (f-1)\ S_2\ \dot{R}\ \dot{R}_o{}^T$$

$$+\ U\ \dot{R}\ \ddot{R}_o{}^T + f\ I$$

$$\text{PHI\_MC}_{1\text{ to }3,\ 4\text{ to }6} = \frac{\partial R}{\partial \dot{R}_o} = -\ \dot{f}\ S_2\ R\ R_o{}^T - (\dot{g}-1)\ S_2\ R\ \dot{R}_o{}^T$$

$$+\ (f-1)\ S_2\ \dot{R}\ R_o{}^T + g\ S_2\ \dot{R}\ \dot{R}_o{}^T$$

$$+\ g\ I - U\ \dot{R}\ \dot{R}_o{}^T$$

$$\text{PHI\_MC}_{4\text{ to }6,\ 1\text{ to }3} = \frac{\partial \dot{R}}{\partial R_o} = -\ \dot{f}\ \left(\frac{S_0}{r\ r_o} + \frac{1}{r^2} + \frac{1}{r_o{}^2}\right)\ R\ R_o{}^T \qquad \text{F3}$$

$$-\ \left[\frac{\dot{f}\ S_1 + (\dot{g}-1)/r}{r}\right]\ R\ \dot{R}_o{}^T$$

$$+\ \left[\frac{\dot{f}\ S_1 + (f-1)/r_o}{r_o}\right]\ \dot{R}\ R_o{}^T + \dot{f}\ S_2\ \dot{R}\ \dot{R}_o{}^T$$

$$+\ \dot{f}\ I + U\ \ddot{R}\ \ddot{R}_o{}^T$$

F3 This equation shall be protected against division by zero (Reference 3.6-3).

$$\text{PHI\_MC}_{4 \text{ to } 6,\ 4 \text{ to } 6} = \frac{\partial \dot{R}}{\partial \dot{R}_o} = -\left[\frac{\dot{f}\ S_1 + (\dot{g} - 1)/r}{r}\right] R\ R_o{}^T \qquad \text{F3}$$

$$-\left[\frac{(\dot{g} - 1)\ S_1}{r}\right] R\ \dot{R}_o{}^T + \dot{f}\ S_2\ \dot{R}\ R_o{}^T$$

$$+ (\dot{g} - 1)\ S_2\ \dot{R}\ \dot{R}_o{}^T + \dot{g}\ I - U\ \ddot{R}\ \dot{R}_o{}^T$$

Certain recurring groups of symbols may be collected to facilitate ease of coding and minimization of error. Each 3-by-3 submatrix of the 6-by-6 matrix PHI_MC results from the summation of 3-by-3 matrices generated by the dyadic product of groups of vectors of length three.

B. Interface Requirements. The input and output data are shown in table 4.2.5.2.

C. Processing Requirements. This subfunction is called by the following subfunctions:

REND_BIAS_AND_COV_PROP	(section 4.2.5)
REND_NAV_INTERP	(section 4.2.8.1.1)

D. Constraints. None

E. Supplementary Information. A suggested implementation in the form of a detailed flowchart may be found in Appendix B under the name of MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6.

F3 This equation shall be protected against division by zero (Reference 3.6-3).

TABLE 4.2.5.2.- MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6 INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
DELTIM	DT_COV	REND_BIAS_AND_COV_PROP	
G_ONE	TOT_ACC_LAST	REND_BIAS_AND_COV_PROP	
G_TWO	TOT_ACC	REND_BIAS_AND_COV_PROP	
R_ONE	R_LAST	REND_BIAS_AND_COV_PROP	
R_TWO	R_FILT	REND_BIAS_AND_COV_PROP	
V_ONE	V_LAST	REND_BIAS_AND_COV_PROP	
V_TWO	V_FILT	REND_BIAS_AND_COV_PROP	
DELTIM	DT_COV	REND_BIAS_AND_COV_PROP	
G_ONE	G_TV_LAST	REND_BIAS_AND_COV_PROP	
G_TWO	G_TV	REND_BIAS_AND_COV_PROP	
R_ONE	R_TV_LAST	REND_BIAS_AND_COV_RPOP	
R_TWO	R_TV	REND_BIAS_AND_COV_PROP	
V_ONE	V_TV_LAST	REND_BIAS_AND_COV_PROP	
V_TWO	V_TV	REND_BIAS_AND_COV_PROP	
DELTIM	DELTAT_GO	REND_NAV_INTERP	
G_ONE	TOT_ACC	REND_NAV_INTERP	
G_TWO	A_RESID	REND_NAV_INTERP	
R_ONE	R_FILT	REND_NAV_INTERP	
R_TWO	R_RESID	REND_NAV_INTERP	
V_ONE	V_FILT	REND_NAV_INTERP	
V_TWO	V_RESID	REND_NAV_INTERP	
DELTIM	DELTAT_GO	REND_NAV_INTERP	
G_ONE	G_TV	REND_NAV_INTERP	
G_TWO	A_TV_RESID	REND_NAV_INTERP	
R_ONE	R_TV	REND_NAV_INTERP	
R_TWO	R_TV_RESID	REND_NAV_INTERP	
V_ONE	V_TV	REND_NAV_INTERP	
V_TWO	V_TV_RESID	REND_NAV_INTERP	

TABLE 4.2.5.2.- MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6 INPUT/OUTPUT.- Continued

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
PHI_MC	PHI$_{1to6,\ 1to6}$		REND_BIAS_AND_COV_PROP
PHI_MC	PHI$_{1to6,\ 1to6}$		REND_BIAS_AND_COV_PROP
PHI_MC	PHI_PATCH		REND_NAV_INTERP
PHI_MC	PHI_PATCH		REND_NAV_INTERP

TABLE 4.2.5.2.- MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6 INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
EARTH_MU	**	
F	F_AND_G	
FDOT	F_AND_G	
G	F_AND_G	
GDOT	F_AND_G	
ID_MATRIX_3X3	**	
R_TWO	F_AND_G	
R_TWO_INV	F_AND_G	F_AND_G
SQR_EMU	*	
S0	F_AND_G	
S1	F_AND_G	
S2	F_AND_G	
S3	F_AND_G	
THETA	F_AND_G	
C1		F_AND_G
DELTIM		F_AND_G
D_ONE		F_AND_G
D_TWO		F_AND_G
R_ONE		F_AND_G
R_ONE_INV		F_AND_G
SMA		F_AND_G
V_ONE		F_AND_G

*Onorbit/Rendezvous principal function, see section 4.2

**Initialization parameters, see section 4.7

4.2.5.2.1 Conic solution (F_AND_G).- The conic solution subfunction utilized by the state vector interpolation, position-velocity submatrix of state transition matrix, and precision integration subfunctions shall provide the capability to trace the progress of a point along its orbit assuming pure Keplerian motion, by means of the F and G series algorithm in terms of the eccentric anomaly.

The variables F and G and $\dot{F}$ and $\dot{G}$ shall be calculated as functions of the difference in eccentric anomaly between an initial time at which a position and a velocity vector are known and a final time at which they are required.

If the final position and velocity are known, the difference in eccentric anomaly can be easily calculated and the F, G, $\dot{F}$ and $\dot{G}$ expressions can be obtained with the use of certain auxiliary variables called here S0, S1, S2, and S3.

If the final position and velocity are not known but only the transfer time, it is necessary to solve a form of Kepler's equation to obtain the difference in eccentric anomaly.

A. Detailed Requirements.- The conic solution (F_AND_G) subfunction will be initiated by the call statement of the following form:

CALL: F_AND_G

IN LIST: SMA,DELTAT,C1,$\underline{R}$ _IN,R_IN_INV,R_FIN_INV,$\underline{V}$ _IN,D_IN,D_FIN

OUT LIST: F,G,FDOT,GDOT,S0,S1,S2,S3,$\underline{R}$ _FIN,R_FIN_INV,THETA

where:

SMA = semi-major axis of the conic

DELTAT = transfer time

C1 = an auxiliary constant, equal to the square root of SMA divided by the square root of Earth's gravitational constant

$\underline{R}$ _IN = the initial position vector (M50)

R_IN_INV = the reciprocal of the magnitude of $\underline{R}$ _IN

R_FIN_INV = the reciprocal of the magnitude of R_FIN (if unknown, a zero shall be input)

$\underline{V}$ _IN = the initial velocity vector (M50)

D_IN = the dot product of the initial position and velocity vectors

D_FIN = the dot product of the final position and velocity vectors (if unknown, a zero shall be input)

The conic solution subfunction shall perform the following in the order indicated:

1. Check the value of R_FIN_INV to see if Kepler's equation is to be solved.

```
IF R_FIN_INV = 0
```

a. If R_FIN_INV ≠ 0., which indicates that the final position vector is already known, the difference in eccentric anomaly shall be obtained from the expression

THETA = [C1(D_FIN-D_IN) + DELTAT/C1]/SMA F3

b. If R_FIN_INV = 0., the final position vector is to be calculated. This requires solving a modified form of Kepler's equation, which shall be accomplished by an iterative process that consists of the following steps:

(1) Two auxiliary quantities shall be obtained from the input data:

ONEMRIN = (SMA - 1./R_IN_INV)/SMA F3

D_MN_AN = DELTAT/(C1 SMA) F3

D_MN_AN is the difference in mean anomaly, which shall be taken as a first approximation to the difference in eccentric anomaly, THETA. An iteration counter shall also be started.

THETA = D_MN_AN

I = 1

(2) Then THETA and THETA_COR shall be recalculated until the number of iterations attains a predetermined maximum.

```
DO UNTIL
I = NUM_KEP_ITER
```

F3 This equation shall be protected against division by zero (Reference 3.6-3).

by repeatedly evaluating the equations

SO = COS (THETA) F9

S1 = SIN (THETA) F9

S2 = 1. -SO

ERR = D_MN_AN-THETA-D_IN C1 S2/SMA + ONEMRIN S1 F3

THETA_COR = ERR/(1. + D_IN C1 S1/SMA - ONEMRIN SO) F3

THETA = THETA + THETA_COR

I = I + 1

2. When the difference in eccentric anomaly is determined, certain auxiliary variables shall be calculated.

SO = COS (THETA) F9

S1 = SIN (THETA) F9

S2 = 1. -SO

S3 = THETA - S1

3. The values of F and G shall then be determined.

F = 1. - SMA S2 R_IN_INV

G = DELTAT - C1 SMA S3

4. If the final position vector and the reciprocal of its magnitude were not known, they shall be calculated:

IF R_FIN_INV = 0., then set

$\underline{R}$ _FIN = F $\underline{R}$ _IN + G $\underline{V}$ _IN

R_FIN_INV = 1./ | $\underline{R}$ _FIN | F3

5. The functions $\dot{F}$ and $\dot{G}$, required for the calculation of the final velocity vector, shall be evaluated:

F3 This equation shall be protected against division by zero (Reference 3.6-3).

F9 This equation shall be protected against return value of sine or cosine with magnitude greater than unity (Reference 3.6-9).

FDOT = - EARTH_MU C1 S1 R_IN_INV R_FIN_INV

GDOT = 1. - SMA S2 R_FIN_INV

Finally, the out list of the conic solution subfunction shall contain the following quantities (different users require different sets of these):

F, G, FDOT, GDOT, S0, S1, S2, S3, R _FIN, R_FIN_INV,THETA

B. Interface Requirements.- Input and output parameters for the conic solution (F_AND_G) are given in table 4.2.5.2.1.

C. Processing Requirements.- The following are the code names of those subfunctions which call the F_AND_G conic solution subfunction:

PINES_METHOD

MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6x6

ONORBIT_SV_INTERP

D. Constraints.- None

E. Supplementary Information.- A suggested implementation of this subfunction in the form of a detailed flow diagram may be found in Appendix B under the following:

F_AND_G

TABLE 4.2.5.2.1.- F_AND_G INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
C1	C1	PINES_METHOD	
DELTAT	DELTAT	PINES_METHOD	
D_FIN	D_FIN_TEMP	PINES_METHOD	
D_IN	D_IN	PINES_METHOD	
R_FIN_INV	R_FIN_TEMP_INV	PINES_METHOD	
R _IN	$XN_{1 \text{ to } 3}$	PINES_METHOD	
R_IN_INV	R_IN_INV	PINES_METHOD	
SMA	SMA	PINES_METHOD	
V _IN	XN_{4} to 6	PINES_METHOD	
C1	C1	MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6	
DELTAT	DELTIM	MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6	
D_FIN	D_TWO	MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6	
D_IN	D_ONE	MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6	
R_FIN_INV	R_TWO_INV	MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6	
R _IN	R _ONE	MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6	
R_IN_INV	R_ONE_INV	MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6	
SMA	SMA	MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6	
V _IN	V _ONE	MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6	
C1	C1	ONORBIT_SV_INTERP	
DELTAT	-DELTAT_GO	ONORBIT_SV_INTERP	
D_FIN	D_FIN_TEMP	ONORBIT_SV_INTERP	

TABLE 4.2.5.2.1.- F_AND_G INPUT/OUTPUT.- Continued

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
D_IN	D_TWO	ONORBIT_SV_INTERP	
R_FIN_INV	R_FIN_TEMP_INV	ONORBIT_SV_INTERP	
R _IN	R _TWO	ONORBIT_SV_INTERP	
R_IN_INV	R_TWO_INV	ONORBIT_SV_INTERP	
SMA	SMA	ONORBIT_SV_INTERP	
V _IN	V _TWO	ONORBIT_SV_INTERP	
F	F		PINES_METHOD
FDOT	FDOT		PINES_METHOD
G	G		PINES_METHOD
GDOT	GDOT		PINES_METHOD
R _FIN	$X_{1 to 3}$		PINES_METHOD
R_FIN_INV	R_FIN_INV		PINES_METHOD
S0	S0		PINES_METHOD
S1	S1		PINES_METHOD
S2	S2		PINES_METHOD
S3	S3		PINES_METHOD
THETA	THETA		PINES_METHOD
F	F		MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6
FDOT	FDOT		MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6
G	G		MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6
GDOT	GDOT		MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6
R _FIN	R _TWO		MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6
R_FIN_INV	R_TWO_INV		MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6
S0	S0		MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6

TABLE 4.2.5.2.1.- F_AND_G INPUT/OUTPUT.- Continued

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
S1	S1		MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6
S2	S2		MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6
S3	S3		MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6
THETA	THETA		MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6
F	F		ONORBIT_SV_INTERP
FDOT	FDOT		ONORBIT_SV_INTERP
G	G		ONORBIT_SV_INTERP
GDOT	GDOT		ONORBIT_SV_INTERP
R _FIN	R _RESID		ONORBIT_SV_INTERP
R_FIN_INV	R_FIN_INV		ONORBIT_SV_INTERP
S0	S0		ONORBIT_SV_INTERP
S1	S1		ONORBIT_SV_INTERP
S2	S2		ONORBIT_SV_INTERP
S3	S3		ONORBIT_SV_INTERP
THETA	THETA		ONORBIT_SV_INTERP

TABLE 4.2.5.2.1.- F_AND_G INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
EARTH_MU	**	
NUM_KEP_ITER	**	

**Initialization parameters, see section 4.7

4.2.6 Rendezvous Sensor Measurement Selection (REND_SENSOR_SELECT)

The rendezvous sensor measurement selection subfunction shall perform the following tasks when in the rendezvous navigation phase, when the magnitude of the IMU sensed accelerations are below a predetermined threshold, and when on a covariance propagation cycle.

- Supply certain positive feedback flags to the REL_NAV display
- Select the proper rendezvous sensor navigation measurement incorporation subfunctions (section 4.2.8) to process the sensor angle data
- Select the type of data processing by the rendezvous navigation Kalman filter, REND_NAV_FILTER (sec. 4.2.8.1.2). Data are used to compute statistical parameters for display only, or data are used to update the state and covariance matrix as well as for the statistical computations
- Select the proper type of edit criteria to be used by the rendezvous navigation filter, REND_NAV_FILTER (section 4.2.8.1.2), in the bilevel edit test which determines if data are to be edited
- Reset the AUTO/INHIBIT/FORCE (AIF) flags so that a FORCE will be in effect for only one navigation filter cycle without being reset by the crew

A. Detailed Requirements.

1. The rendezvous sensor measurement selection subfunction initially sets certain positive feedback display parameters and then sets other appropriate flags to OFF so that the selection tasks are correctly accomplished. The following flags are required to be initialized upon each execution of the rendezvous sensor measurement selection subfunction.

   a. Certain positive feedback flags are defined for the REL_NAV display.

      RDOT_AIF_DISPLAY = NAV_RDOT_AIF

      RANGE_AIF_DISPLAY = NAV_RANGE_AIF

      ANGLES_AIF_DISPLAY = NAV_ANGLES_AIF

      DOING_MEAS_ENABLE = NAV_MEAS_ENABLE

   b. In order for this subfunction to properly select the appropriate rendezvous sensor navigation measurement incorporation subfunction for the proper angle set, the following flags (DO NAV flags) are initially set to OFF.

      DO_RR_ANGLES_NAV = OFF
      DO_ST_ANGLES_NAV = OFF
      DO_COAS_ANGLES_NAV = OFF

c. In order to properly process and record the type of processing of the data, the following flags are initially set to OFF.

(1) Statistical processing flags

RDOT_STAT = OFF
ST_ANGLES_STAT = OFF
COAS_ANGLES_STAT = OFF
RANGE_STAT = OFF
RR_ANGLES_STAT = OFF

(2) EDIT OVERRIDE flags for the bilevel edit test

RANGE_EDIT_OVERRIDE = OFF
RDOT_EDIT_OVERRIDE = OFF
RR_ANGLES_EDIT_OVERRIDE = OFF
ST_ANGLES_EDIT_OVERRIDE = OFF
COAS_ANGLES_EDIT_OVERRIDE = OFF

(3) A 4-dimensional array to record the type of processing of the data by the rendezvous sensor navigation measurement incorporation subfunction

SENSOR_EDIT = OFF

2. Next, a local flag (MEAS_STAT) is set to control measurement processing for statistical display purposes when in Major Mode 202.

a. MEAS_STAT = ON when NAV_MEAS_ENABLE = OFF and NAV_MM_202 = ON. This will result in measurement processing proceeding as if all the measurement AIF flags are set to INHIBIT, i.e., measurements are only used to compute statistical parameters for display purposes.

b. MEAS_STAT = OFF when NAV_MEAS_ENABLE = ON or NAV_MM_202 = OFF. This will result in the measurement data being processed in the nominal manner.

3. The NAV_RANGE_AIF switch and the MEAS_STAT flag are tested to appropriately set the RANGE_STAT flag and the RANGE_EDIT_OVERRIDE flag.

a. If either NAV_RANGE_AIF = INHIBIT or MEAS_STAT = ON then

RANGE_STAT = ON

resulting in the range measurement being processed for statistical display only.

b. If NAV_RANGE_AIF is not set to INHIBIT and MEAS_STAT = OFF then the NAV_RANGE_AIF flag is tested for a FORCE value.

(1) If NAV_RANGE_AIF = FORCE then

RANGE_EDIT_OVERRIDE = ON

so that the edit criteria will be relaxed for the bilevel residual edit test.

4. Next, the NAV_RDOT_AIF switch and the MEAS_STAT flag are tested to appropriately set the RDOT_STAT flag and the RDOT_EDIT_OVERRIDE flag.

   a. If either NAV_RDOT_AIF = INHIBIT or MEAS_STAT = ON then

      RDOT_STAT = ON,

      resulting in the range rate measurement being processed for statistical display only.

   b. If NAV_RDOT_AIF is not set to INHIBIT and MEAS_STAT = OFF then the NAV_RDOT_ AIF flag is tested for a FORCE value.

      (1) If NAV_RDOT_AIF = FORCE then

      RDOT_EDIT_OVERRIDE = ON

      so that the edit criteria will be relaxed for the bilevel residual edit test.

5. The NAV_RR_ANGLES_ENABLE switch is tested to see if the rendezvous radar angle set has been selected by the crew.

   a. If NAV_RR_ANGLES_ENABLE = ON then set

      DO_RR_ANGLES_NAV = ON

      so that the proper initialization procedure will take place for the angle data, and the rendezvous radar angles measurement incorporation subfunction will be executed.

      (1) Next, the NAV_ANGLES_AIF switch and the MEAS_STAT flags are tested to set the RR_ANGLES_STAT flag and the RR_ANGLES_EDIT_OVERRIDE flag.

         - If either NAV_ANGLES_AIF = INHIBIT or MEAS_STAT = ON then

           RR_ANGLES_STAT = ON,

           resulting in the rendezvous radar angle measurements being processed for statistical display only.

         - If NAV_ANGLES_AIF is not set to INHIBIT and MEAS_STAT = OFF the NAV_ANGLES_AIF flag is tested for a FORCE value.

If NAV_ANGLES_AIF = FORCE then

RR_ANGLES_EDIT_OVERRIDE = ON

so that the edit criteria will be relaxed for the bilevel residual edit test.

b. If NAV_RR_ANGLES_ENABLE is not ON then the NAV_ST_ENABLE switch is tested to see if the star tracker angles set has been selected by the crew.

(1) If NAV_ST_ENABLE = ON then set

DO_ST_ANGLES_NAV = ON

so that the proper initialization procedure will take place for the angle data and the star tracker measurement incorporation subfunction will be executed.

- Next, the NAV_ANGLES_AIF switch and the MEAS_STAT flags are tested to set the ST_ANGLES_STAT flag and the ST_ANGLES_EDIT_OVERRIDE flag.

If either NAV_ANGLES_AIF = INHIBIT or MEAS_STAT = ON then

ST_ANGLES_STAT = ON,

resulting in the star tracker angle measurements being processed for statistical display only.

If NAV_ANGLES_AIF is not set to INHIBIT and MEAS_STAT = OFF the NAV_ANGLES_AIF flag is tested for a FORCE value.

If NAV_ANGLES_AIF = FORCE then

ST_ANGLES_EDIT_OVERRIDE = ON

so that the edit criteria will be relaxed for the bilevel residual edit test.

c. If NAV_RR_ANGLES_ENABLE = OFF and NAV_ST_ENABLE = OFF then the COAS angle data set is assumed to be enabled, thus

DO_COAS_ANGLES_NAV = ON

so that the proper initialization procedure will take place for the angle data and the COAS measurement incorporation subfunction will be executed.

(1) Next, the NAV_ANGLES_AIF switch and the MEAS_STAT flags are tested to set the COAS_ANGLES_STAT flag and the COAS_ANGLES_EDIT_ OVERRIDE flag.

- If either NAV_ANGLES_AIF = INHIBIT or MEAS_STAT = ON then

  COAS_ANGLES_STAT = ON,

  resulting in the COAS angle measurements being processed for statistical display only.

- If NAV_ANGLES_AIF is not set to INHIBIT and MEAS_STAT = OFF the NAV_ANGLES_AIF flag is tested for a FORCE value.

  If NAV_ANGLES_AIF = FORCE then

  COAS_ANGLES_EDIT_OVERRIDE = ON

  so that the edit criteria will be relaxed for the bilevel residual edit test.

6. The NAV_RANGE_AIF flag is tested so that a FORCE value will be acknowledged for one navigation cycle without being reset by the crew.

   a. If NAV_RANGE_AIF = FORCE then

      RANGE_AIF = RANGE_AIF_LAST.

      where the RANGE_AIF_LAST flag is a local flag which records the last non force value of the RANGE_AIF flag.

   b. If NAV_RANGE_AIF is not equal to FORCE then

      RANGE_AIF_LAST = RANGE_AIF.

7. Next, the NAV_RDOT_AIF flag is tested for the same reasons as given in 6.

   a. If NAV_RDOT_AIF is equal to FORCE then

      RDOT_AIF = RDOT_AIF_LAST.

   b. If NAV_RDOT_AIF is not equal to FORCE then

      RDOT_AIF_LAST = RDOT_AIF.

8. Finally, the NAV_ANGLES_AIF flag is tested similarly to 6 and 7.

   a. If NAV_ANGLES_AIF = FORCE then

      ANGLES_AIF = ANGLES_AIF_LAST.

b. If NAV_ANGLES_AIF is not equal to FORCE then

ANGLES_AIF_LAST = ANGLES_AIF.

B. Interface Requirements. The input and output parameters for this subfunction are indicated in table 4.2.6.

C. Processing Requirements. The sensor measurement selection subfunction shall be performed only when on a covariance propagation cycle and when the IMU sensed accelerations are below a design dependent threshold (MEAS_THRESHOLD), i.e., NAV_MEAS = ON. This subfunction is called by the onorbit/rendezvous trunk logic, ONORBIT_RENDEZVOUS_NAV, and shall be executed before the sensor measurement initialization subfunction, REND_NAV_SENSOR_INIT.

D. Constraints. The RANGE_AIF_LAST, RDOT_AIF_LAST and ANGLES_AIF_LAST flags need to be initialized to INHIBIT.

E. Supplementary Information. A suggested implementation of this subfunction may be found in the Appendix B flowchart

REND_SENSOR_SELECT.

TABLE 4.2.6.- REND_SENSOR_SELECT INPUT/OUTPUT

Variable Name	Input Source	Output Destination
NAV_ANGLES_AIF	NAV_ONORBIT_RENDEZVOUS	
ANGLES_AIF_LAST	**	
NAV_MEAS_ENABLE	NAV_ONORBIT_RENDEZVOUS	
NAV_MM_202	NAV_ONORBIT_RENDEZVOUS	
NAV_RANGE_AIF	NAV_ONORBIT_RENDEZVOUS	
RANGE_AIF_LAST	**	
NAV_RDOT_AIF	NAV_ONORBIT_RENDEZVOUS	
RDOT_AIF_LAST	**	
NAV_RR_ANGLES_ENABLE	NAV_ONORBIT_RENDEZVOUS	
NAV_ST_ENABLE	NAV_ONORBIT_RENDEZVOUS	
ANGLES_AIF_DISPLAY		*
COAS_ANGLES_EDIT_OVERRIDE		COAS_NAV
COAS_ANGLES_STAT		COAS_NAV
DO_COAS_ANGLES_NAV		REND_NAV_SENSOR_INIT,*
DOING_MEAS_ENABLE		*

*Onorbit/Rendezvous principal function I/O Table, table 4.2
**Initialization parameters, see section 4.7

TABLE 4.2.6.- REND_SENSOR_SELECT INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
DO_RR_ANGLES_NAV		REND_NAV_SENSOR_INIT, NAV_ONORBIT_RENDEZVOUS,*
DO_ST_ANGLES_NAV		REND_NAV_SENSOR_INIT, NAV_ONORBIT_RENDEZVOUS,*
RANGE_AIF_DISPLAY		*
RANGE_EDIT_OVERRIDE		RRDOT_NAV
RANGE_STAT		RRDOT_NAV
RDOT_AIF_DISPLAY		*
RDOT_EDIT_OVERRIDE		RRDOT_NAV
RDOT_STAT		RRDOT_NAV
RR_ANGLES_EDIT_OVERRIDE		RR_ANGLE_NAV
RR_ANGLES_STAT		RR_ANGLE_NAV
SENSOR_EDIT		MEAS_PROCESSING_STATISTICS_REND,*
ST_ANGLES_EDIT_OVERRIDE		STAR_TRACKER_NAV
ST_ANGLES_STAT		STAR_TRACKER_NAV

*Onorbit/Rendezvous principal function I/O Table, table 4.2

4.2.7 Sensor Measurement Initialization (REND_NAV_SENSOR_INIT)

This sensor measurement initialization subfunction shall be invoked on each covariance propagation subcycle when the IMU sensed accelerations fall below a design dependent threshold (MEAS_THRESHOLD). The reinitialization of the sensor bias portion of the state vector and covariance matrix shall be performed upon entering rendezvous navigation, whenever the measurement type configuration has changed to include new measurements, or when an automatic in-flight update or a REL_NAV display update occurs while the Onorbit/Rendezvous Navigation principal function is active. The sensor covariance and bias setup subfunction (section 4.2.7.1) shall be invoked to account for the change in the measurement status.

A. Detailed Requirements. The following steps shall be performed (in the order indicated):

1. If the rendezvous radar range and range rate data set was off (DO_RRDOT_NAV_LAST = OFF) on the previous covariance propagation subcycle:

   Call the sensor covariance and bias setup subfunction to reconfigure the bias portions of the state vector and the covariance matrix for this data set (see section 4.2.7.1)

   CALL: SETUP

   IN LIST: 2, TAU_RRDOT, BIAS_VAR_RRDOT, VAR_RRDOT, RRDOT_BIAS_INIT

2. Test to determine if a new angle sensor has been selected by the crew.

   a. If the rendezvous radar angle data set has been made available since the previous covariance propagation subcycle (DO_RR_ANGLES_NAV = ON and DO_RR_ANGLES_NAV_LAST = OFF):

      Call the sensor covariance and bias setup subfunction to reconfigure the bias portions of the state vector and the covariance matrix for this data set (see section 4.2.7.1).

      CALL: SETUP

      IN LIST: 0, TAU_RR_ANGLES, BIAS_VAR_RR_ANGLES, VAR_RR_ANGLES, RR_ANGLES_BIAS_INIT

      Also set the positive feedback flag to the display for rendezvous radar angles:

      ANGLES_ENABLE_DISPLAY = 2

   b. If the star tracker angle data set has been made available since the previous covariance propagation subcycle (DO_ST_ANGLES_NAV = ON and DO_ST_ANGLES_NAV_LAST = OFF):

Call the sensor covariance and bias setup subfunction to reconfigure the bias portions of the state vector and the covariance matrix for this data set (see section 4.2.7.1).

CALL: SETUP

IN LIST: 0, TAU_ST_ANGLES, BIAS_VAR_ST_ANGLES,
VAR_ST_ANGLES, ST_ANGLES_BIAS_INIT

Also set the positive feedback flag to the display for star tracker angles:

ANGLES_ENABLE_DISPLAY = 0

c. If the COAS angle data set has been made available since the previous covariance propagation subcycle (DO_COAS_ANGLES_NAV = ON and DO_COAS_ANGLES_NAV_LAST = OFF):

Call the sensor covariance and bias setup subfunction to reconfigure the bias portions of the state vector and the covariance matrix for this data set (see section 4.2.7.1).

CALL: SETUP

IN LIST: 0, TAU_COAS_ANGLES, BIAS_VAR_COAS_ANGLES,
VAR_COAS_ANGLES, COAS_ANGLES_BIAS_INIT

Also set the positive feedback flag to the display for COAS angles:

ANGLES_ENABLE_DISPLAY = 1

3. Save the current values of the "DO_NAV" flags for the next execution of this subfunction.

DO_RRDOT_NAV_LAST = ON

DO_RR_ANGLES_NAV_LAST = DO_RR_ANGLES_NAV

DO_ST_ANGLES_NAV_LAST = DO_ST_ANGLES_NAV

DO_COAS_ANGLES_NAV_LAST = DO_COAS_ANGLES_NAV

B. Interface Requirements. The input and output variables for this subfunction are described in table 4.2.7.

C. Processing Requirements. This subfunction is called by

NAV_ONORBIT_RENDEZVOUS (section 4.2.1)

D. Constraints. None

E. Supplementary Information. A suggested implementation in the form of a detailed flowchart can be found in Appendix B under the name:

REND_NAV_SENSOR_INIT

TABLE 4.2.7.- REND_NAV_SENSOR_INIT INPUT/OUTPUT

Variable Name	Input Source	Output Destination
BIAS_VAR_COAS_ANGLES	* *	SETUP
BIAS_VAR_RR_ANGLES	* *	SETUP
BIAS_VAR_RRDOT	* *	SETUP
BIAS_VAR_ST_ANGLES	* *	SETUP
COAS_ANGLES_BIAS_INIT	* *	SETUP
DO_COAS_ANGLES_NAV	REND_SENSOR_SELECT	
DO_COAS_ANGLES_NAV_LAST	REND_COV_INIT,*	
DO_RR_ANGLES_NAV	REND_SENSOR_SELECT	
DO_RR_ANGLES_NAV_LAST	REND_COV_INIT,*	
DO_RRDOT_NAV_LAST	REND_COV_INIT,*	
DO_ST_ANGLES_NAV	REND_SENSOR_SELECT	
DO_ST_ANGLES_NAV_LAST	REND_COV_INIT,*	
RR_ANGLES_BIAS_INIT	* *	SETUP
RRDOT_BIAS_INIT	* *	SETUP
ST_ANGLES_BIAS_INIT	* *	SETUP
TAU_COAS_ANGLES	* *	SETUP
TAU_RR_ANGLES	* *	SETUP
TAU_RRDOT	* *	SETUP
TAU_ST_ANGLES	* *	SETUP
VAR_COAS_ANGLES	* *	SETUP

*Onorbit/Rendezvous principal function, see section 4.2

**Initialization parameters, see section 4.7

TABLE 4.2.7.- REND_NAV_SENSOR_INIT INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
VAR_RR_ANGLES	* *	SETUP
VAR_RRDOT	* *	SETUP
VAR_ST_ANGLES	* *	SETUP
ANGLES_ENABLE_DISPLAY		*

*Onorbit/Rendezvous principal function, see section 4.2
**Initialization parameters, see section 4.7

4.2.7.1 Sensor Covariance and Bias Setup (SETUP)

The sensor measurement initialization subfunction (section 4.2.7) shall invoke the sensor covariance and bias setup subfunction to reconfigure the bias portions of the state vector and the covariance matrix whenever the covariance matrix initialization subfunction (REND_COV_INIT) has been exercised since the last covariance propagation subcycle, or when a new measurement set has been selected since the previous covariance propagation subcycle. New exponentially correlated time constants and process noise variances are also selected from premission values for use in the computation of the state transition matrix and in the addition of process noise.

A. Detailed Requirements. This subfunction is called with the following internal variables in the IN LIST:

INLIST: I, TAU, BIAS_VAR, BIAS_COV_VAR, BIAS_INIT

where

I is a pointer for the desired measurement set

= 0 for all angle data

= 2 for rendezvous range and range rate data

TAU sensor time constants

BIAS_VAR sensor bias variances

BIAS_COV_VAR sensor bias variance terms for the covariance matrix

BIAS_INIT initial sensor bias

The following steps shall be performed for J = 1 to 2:

1. K = I + J

2. The state vector is to be reconfigured by setting its bias slots associated with the new measurement types to premission values. Bias values of measurement types no longer needed do not have to be zeroed.

$\text{TAU\_SENS}_K = \text{TAU}_J$

$\text{SENSOR\_BIAS}_K = \text{BIAS\_INIT}_J$

$\text{VAR\_SENS}_K = \text{BIAS\_VAR}_J$

3. The covariance matrix is to be reconfigured by zeroing the off-diagonal terms associated with the new measurement type. The diagonal terms are then set equal to premission variance values of the new measurement type. The rows and columns associated with the discontinued measurement types do not have to be zeroed unless they are used by a new measurement type.

$$E_{9+K,1to13} = 0.0$$

$$E_{1to13,9+K} = 0.0$$

$$E_{9+K,9+K} = BIAS\_COV\_VAR_J$$

4. The accept/reject counters and the edit ratio of the residual edit test for each measurement group must be reset to zero for use by the measurement processing statistics subfunction (section 4.2.9).

$$N\_ACCEPT_K = 0$$

$$N\_REJECT_K = 0$$

$$SEQ\_ACCEPT_K = 0$$

$$SEQ\_REJECT_K = 0$$

$$NAV\_SIG_K = 0.0$$

B. Interface Requirements. The input and output variables for this subfunction are defined in table 4.2.7.1.

C. Processing Requirements. This subfunction is called by

REND_NAV_SENSOR_INIT (section 4.2.7)

D. Constraints. None

E. Supplementary Information. A suggested implementation in the form of a detailed flowchart can be found in Appendix B under the following name:

TABLE 4.2.7.1.- SETUP INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
BIAS_COV_VAR	VAR_RRDOT	REND_NAV_SENSOR_INIT	
BIAS_INIT	RRDOT_BIAS_INIT	REND_NAV_SENSOR_INIT	
BIAS_VAR	BIAS_VAR_RRDOT	REND_NAV_SENSOR_INIT	
I	2	REND_NAV_SENSOR_INIT	
TAU	TAU_RRDOT	REND_NAV_SENSOR_INIT	
BIAS_COV_VAR	VAR_RR_ANGLES	REND_NAV_SENSOR_INIT	
BIAS_INIT	RR_ANGLES_BIAS_INIT	REND_NAV_SENSOR_INIT	
BIAS_VAR	BIAS_VAR_RR_ANGLES	REND_NAV_SENSOR_INIT	
I	0	REND_NAV_SENSOR_INIT	
TAU	TAU_RR_ANGLES	REND_NAV_SENSOR_INIT	
BIAS_COV_VAR	VAR_ST_ANGLES	REND_NAV_SENSOR_INIT	
BIAS_INIT	ST_ANGLES_BIAS_INIT	REND_NAV_SENSOR_INIT	
BIAS_VAR	BIAS_VAR_ST_ANGLES	REND_NAV_SENSOR_INIT	
I	0	REND_NAV_SENSOR_INIT	
TAU	TAU_ST_ANGLES	REND_NAV_SENSOR_INIT	
BIAS_COV_VAR	VAR_COAS_ANGLES	REND_NAV_SENSOR_INIT	
BIAS_INIT	COAS_ANGLES_BIAS_INIT	REND_NAV_SENSOR_INIT	

TABLE 4.2.7.1.- SETUP INPUT/OUTPUT.- Continued

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
BIAS_VAR	BIAS_VAR_COAS_ANGLES	REND_NAV_SENSOR_INIT	
I	0	REND_NAV_SENSOR_INIT	
TAU	TAU_COAS_ANGLES	REND_NAV_SENSOR_INIT	

TABLE 4.2.7.1.- SETUP INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
E		REND_NAV_FILTER, REND_BIAS_AND_COV_PROP,*
N _ACCEPT		MEAS_PROCESSING_STATISTICS_REND,*
NAV_SIG		RRDOT_NAV, RR_ANGLE_NAV, ANGLE_NAV, MEAS_PROCESSING_STATISTICS_REND
N _REJECT		MEAS_PROCESSING_STATISTICS_REND,*
SENSOR_BIAS		REND_NAV_FILTER, RRDOT_NAV, RR_ANGLE_NAV, ANGLE_NAV, NAV_ONORBIT_RENDEZVOUS
SEQ_ACCEPT		MEAS_PROCESSING_STATISTICS_REND,*
SEQ_REJECT		MEAS_PROCESSING_STATISTICS_REND,*
TAU_SENS		REND_BIAS_AND_COV_PROP
VAR_SENS		REND_BIAS_AND_COV_PROP

*Onorbit/Rendezvous principal function, see section 4.2

4.2.8 State and Covariance Measurement Incorporation

This section documents the requirements for the following four rendezvous sensor navigation measurement incorporation subfunctions:

- Rendezvous radar range and range rate measurements
- Rendezvous radar shaft and trunnion angle measurements
- Star tracker horizontal and vertical angle measurements
- COAS horizontal and vertical angle measurements

Each measurement incorporation subfunction updates the state vector and covariance matrix with the corresponding rendezvous sensor data using a 13-state process noise Kalman filter.

4.2.8.1 Rendezvous Radar Range and Range Rate Measurements (RRDOT_NAV)

This subfunction is responsible for the proper processing of the rendezvous radar range and range rate measurements. This subfunction shall perform the following tasks only when the rendezvous radar is not in the self test mode and the measurement data is labeled valid.

- Calculate the partial derivative of the measurement with respect to the estimated state at measurement time
- Compute the estimated measurement and the measurement residual
- Select the proper variances to model the uncorrelated measurement errors
- Store the old residual ratio, the EDIT OVERRIDE flag, and the STAT flags into temporary locations used by the Kalman Filter Updates subfunction (section 4.2.8.1.2)
- Schedule the Kalman Filter Updates subfunction to process the data
- Store the current residual ratio, the EDIT flag, and the measurement residual for display purposes

A. Detailed Requirements. The following steps shall be performed (in the order indicated):

1. If the rendezvous radar is not in the self test mode (SELF_TEST_FLAG = OFF), then continue with the measurement processing; otherwise exit this subfunction.

   a. The time interval between the current filter time and the time of the radar measurement shall be computed:

DELTAT_GO = T_CURRENT_FILT - T_REND_RADAR

b. Test the range data good flag. If the data is good, perform the following steps:

(1) Call the measurement interpolation subfunction to interpolate the Orbiter state vector and the target state vector to the time of the range measurement (see section 4.2.8.1.1):

CALL: REND_NAV_INTERP

REND_NAV_INTERP subfunction shall also compute an Orbiter or target patch transition matrix PHI_PATCH, the normalized line of sight vector $\underline{I}$ _RHO, and the magnitude of the relative position vector $\underline{R}$ _RHO used in this subfunction.

(2) The rendezvous radar range measurement partial vector is computed with the following equations:

$$B_{1\text{ to }6} = -(\text{PHI\_PATCH}_{1\text{ to }3,\ 1\text{ to }6})^T\ \underline{I}\text{\_RHO}$$

$$B_{12} = 1.0$$

(3) The estimated range measurement and the range measurement residual are then calculated:

$$\text{RNG} = \text{R\_RHO\_MAG} + \text{SENSOR\_BIAS}_3$$

$$\text{DELQ} = \text{Q\_RR\_RNG-RNG}$$

(4) The variance of the uncorrelated range measurement noise is computed:

$$\text{VAR} = (\text{SIG\_RR\_RNG} + \text{SLOPE\_SIG\_RR\_RNG R\_RHO\_MAG})^2$$

$$\text{VAR} = \text{MAXIMUM VAR, VAR\_RR\_RNG\_MIN}$$

(5) The residual test ratio from the previous filter cycle and the measurement processing control flags shall be set as follows:

where RANGE_EDIT_OVERRIDE and RANGE_STAT come from the sensor measurement selection subfunction (section 4.2.6) and the NAV_ SIG_3 comes from the previous navigation cycle's execution of this subfunction as given in the forthcoming step (7).

(6) The Kalman filter update subfunction (section 4.2.8.1.2) shall then be called to update the state and the covariance matrix:

CALL: REND_NAV_FILTER

(7) The measurement edit flag, the residual test ratio, and the range measurement residual shall then be stored for subsequent computation of measurement processing statistics, as described in section 4.2.9:

$$\mathrm{SENSOR\_EDIT}_3 = \mathrm{EDIT\_FLAG}$$

$$\mathrm{NAV\_SIG}_3 = \mathrm{RESID\_TEST\_RATIO}$$

$$\mathrm{SENSOR\_DELQ}_3 = \mathrm{DELQ}$$

c. Test the range rate data good flag. If the data is good, perform the following steps:

(1) Call the measurement interpolation subfunction (section 4.2.8.1.1) to interpolate the Orbiter state vector and the target state vector to the time of the range rate measurement:

CALL: REND_NAV_INTERP.

REND_NAV_INTERP subfunction shall also compute an Orbiter or target patch transition matrix PHI_PATCH, the normalized line of sight vector $\underline{I}\_\mathrm{RHO}$, and the magnitude of the relative position vector R_RHO used in this subfunction.

(2) The rendezvous radar range rate measurement partial vector is computed with the following equations:

$$\underline{U}\_\mathrm{RDOT} = \underline{V}\_\mathrm{RHO/R\_RHO\_MAG} \qquad \text{F3}$$

$$B_{1\text{ to }3} = \underline{I}\_\mathrm{RHO}\ \mathrm{X}\ (\underline{I}\_\mathrm{RHO}\ \mathrm{X}\ \underline{U}\_\mathrm{RDOT})$$

$$B_{4\text{ to }6} = -\underline{I}\_\mathrm{RHO}$$

$$B_{1\text{ to }6} = (\mathrm{PHI\_PATCH})^T B_{1\mathrm{to}6}$$

$$B_{13} = 1.0$$

(3) The estimated range rate measurement and the measurement residual are then calculated:

$$\mathrm{RNG\_DOT} = \underline{R}\_\mathrm{RHO} \cdot \underline{U}\_\mathrm{RDOT} + \mathrm{SENSOR\_BIAS}_4$$

$$\mathrm{DELQ} = \mathrm{Q\_RR\_RNG\_DOT} - \mathrm{RNG\_DOT}$$

where $\underline{R}\_\mathrm{RHO}$ comes from the interpolation process.

F3 This equation shall be protected against division by zero (Reference 3.6-3).

(4) The variance of the uncorrelated range rate measurement error is defined:

VAR = VAR_RANGE_DOT

(5) The residual test ratio from the previous filter cycle and the filter processing control flags shall be set as follows:

RESID_RATIO_OLD = NAV_SIG$_4$

MANUAL_EDIT_OVERRIDE = RDOT_EDIT_OVERRIDE

STAT_FLAG = RDOT_STAT

where RDOT_EDIT_OVERRIDE and RDOT_STAT come from the sensor measurement selection subfunction (section 4.2.6) and the NAV_SIG$_4$ comes from the previous navigation cycle's execution of this subfunction as given in the forthcoming step (7).

(6) The Kalman filter subfunction (section 4.2.8.1.2) shall then be called to update the state and the covariance matrix:

CALL: REND_NAV_FILTER

(7) The measurement edit flag, the residual test ratio, and the range rate measurement residual shall then be stored for subsequent computation of measurement processing statistics, as described in section 4.2.9:

SENSOR_EDIT$_4$ = EDIT_FLAG

NAV_SIG$_4$ = RESID_TEST_RATIO

SENSOR_DELQ$_4$ = DELQ

B. Interface Requirements. The input and output variables for the rendezvous radar range and range rate measurement subfunction are given in table 4.2.8.1.

C. Processing Requirements. This subfunction is called by

NAV_ONORBIT_RENDEZVOUS (section 4.2.1)

D. Constraints. None

E. Supplementary Information. A suggested implementation in the form of a detailed flowchart can be found in Appendix B under the name:

RRDOT_NAV

TABLE 4.2.8.1.- RRDOT_NAV INPUT/OUTPUT

Variable Name	Input Source	Output Destination
B		REND_NAV_FILTER
DELQ		REND_NAV_FILTER
DELTAT_GO		REND_NAV_INTERP, ONORBIT_SV_INTERP
EDIT_FLAG	REND_NAV_FILTER	
I _RHO	REND_NAV_INTERP	
MANUAL_EDIT_OVERRIDE		REND_NAV_FILTER
NAV_SIG	SETUP	MEAS_PROCESSING_ STATISTICS_REND
PHI_PATCH	REND_NAV_INTERP	
Q_RR_RNG	NAV_ONORBIT_RENDEZVOUS	
Q_RR_RNG_DOT	NAV_ONORBIT_RENDEZVOUS	
RANGE_EDIT_OVERRIDE	REND_SENSOR_SELECT	
RANGE_STAT	REND_SENSOR_SELECT	
RDOT_DATA_GOOD	NAV_ONORBIT_RENDEZVOUS	
RDOT_EDIT_OVERRIDE	REND_SENSOR_SELECT	
RDOT_STAT	REND_SENSOR_SELECT	
RESID_RATIO_OLD		REND_NAV_FILTER
RESID_TEST_RATIO	REND_NAV_FILTER	
RNG_DATA_GOOD	NAV_ONORBIT_RENDEZVOUS	
R _RHO	REND_NAV_INTERP	
R_RHO_MAG	REND_NAV_INTERP	

*Onorbit/Rendezvous principal function, see section 4.2

**Initialization parameters, see section 4.7

TABLE 4.2.8.1.- RRDOT_NAV INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
SELF_TEST_FLAG	NAV_ONORBIT_RENDEZVOUS	
SENSOR_BIAS	SETUP, REND_NAV_FILTER	
SENSOR_DELQ		MEAS_PROCESSING_ STATISTICS_REND
SENSOR_EDIT		MEAS_PROCESSING_ STATISTICS_REND, *
SIG_RR_RNG	**	
SLOPE_SIG_RR_RNG	**	
STAT_FLAG		REND_NAV_FILTER
T_CURRENT_FILT	NAV_ONORBIT_RENDEZVOUS	
T_REND_RADAR	NAV_ONORBIT_RENDEZVOUS	
VAR		REND_NAV_FILTER
VAR_RANGE_DOT	**	
VAR_RR_RNG_MIN	**	
V_RHO	REND_NAV_INTERP	

*Onorbit/Rendezvous principal function, see section 4.2
**Initialization parameters, see section 4.7

4.2.8.1.1 Measurement interpolation (REND_NAV_INTERP).- The rendezvous navigation interpolation subfunction is invoked by the rendezvous sensor measurement incorporation subfunctions and is responsible for providing parameters that have been interpolated from current filter time back to the appropriate sensor measurement time. Specifically, this subfunction is charged with the following tasks.

- Invoke the onorbit state vector interpolation subfunction to interpolate both the Shuttle and target position and velocity vectors from current filter time to the time of the measurement
- Invoke the mean conic partial transition matrix subfunction to calculate a patch transition matrix from current filter time to the measurement time for either the Shuttle or target vehicle, depending on which vehicle state is to be updated by the Kalman filter
- Calculate the relative velocity, relative position vector, range, and the line of sight vector between the Shuttle and target. Each of these parameters is computed using the interpolated positions and velocities

A. Detailed Requirements. The following steps shall be performed (in the order indicated):

1. The Orbiter state vector shall be interpolated to the time of the measurement with the use of the state vector interpolation subfunction as described in section 4.2.8.1.1.1.

   CALL: ONORBIT_SV_INTERP

   IN LIST: R _LAST, V _LAST, R _FILT, V _FILT, DV_COV,
   IGD, IGO, IDRAG, IVENT, ATFL_OV

   OUT LIST: R _RESID, V _RESID, A _RESID

2. The target state vector shall be interpolated to the time of the measurement with the use of the state vector interpolation subfunction as described in section 4.2.8.1.1.1.

   CALL: ONORBIT_SV_INTERP

   IN LIST: R _TV_LAST, V _TV_LAST, R _TV, V _TV, O.
   GM_DEG, GM_ORD, DFL, VFL_TV, ATFL_TV

   OUT LIST: R _TV_RESID, V _TV_RESID, A _TV_RESID

3. Next, the SHUTTLE_FILTER_FLAG is tested to see if the Shuttle state or the target state is to be included in the Kalman filter.

   a. If the Shuttle vehicle is the filter vehicle, then the position-velocity state transition submatrix subfunction is used to construct an Orbiter patch transition matrix as described in section 4.2.5.2 for use in the measurement partial calculations.

CALL: MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6

IN LIST: $\underline{R}$_FILT, $\underline{V}$_FILT, TOT_ACC, $\underline{R}$_RESID, $\underline{V}$_RESID, $\underline{A}$_RESID, -DELTAT_GO

OUT LIST: PHI_PATCH

b. If the target vehicle is the filter vehicle, then the position-velocity state transition submatrix subfunction is used to construct a target patch transition matrix as described in section 4.2.5.2 for use in the measurement partials calculation.

CALL: MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6X6

IN LIST: $\underline{R}$_TV, $\underline{V}$_TV, $\underline{G}$_TV, $\underline{R}$_TV_RESID, $\underline{V}$_TV_RESID, $\underline{A}$_TV_RESID, -DELTAT_GO

OUT LIST: PHI_PATCH

4. The following auxiliary parameters will be calculated for use by the measurement subfunctions.

$\underline{V}$_RHO = $\underline{V}$_TV_RESID - $\underline{V}$_RESID

$\underline{R}$_RHO = $\underline{R}$_TV_RESID - $\underline{R}$_RESID

R_RHO_MAG = $\underline{R}$_RHO.

If R_RHO_MAG is less than a premission determined constant, SENSOR_EPS, then

I_RHO_I = SENSOR_DELTA FOR I = 1 to 3

R_RHO_MAG = SENSOR_DELTA

Otherwise,

$\underline{I}$_RHO = $\underline{R}$_RHO/R_RHO_MAG F3

B. Interface Requirements. The input and output data are shown in table 4.2.8.1.1.

C. Processing Requirements. This subfunction is called by the following subfunctions:

F3 This equation shall be protected against division by zero (Reference 3.6-3).

ANGLE_NAV	(section 4.2.8.3.1)
RR_ANGLE_NAV	(section 4.2.8.2)
RRDOT_NAV	(section 4.2.8.1)

D. Constraints. None

E. Supplementary Information. A suggested implementation in the form of a detailed flowchart may be found in Appendix B under the name REND_NAV_INTERP.

TABLE 4.2.8.1.1.- REND_NAV_INTERP INPUT/OUTPUT

Variable Name	Input Source	Output Destination
A _RESID	ONORBIT_SV_INTERP	MEAN_CONIC_PARTIAL_ TRANSITION_MATRIX_6X6
ATFL_OV	**	ONORBIT_SV_INTERP
ATFL_TV	**	ONORBIT_SV_INTERP
A _TV_RESID	ONORBIT_SV_INTERP	MEAN_CONIC_PARTIAL_ TRANSITION_MATRIX_6X6
DELTAT_GO	COAS_NAV, RR_ANGLE_NAV, RRDOT_NAV, STAR_TRACKER_ NAV	MEAN_CONIC_PARTIAL_ TRANSITION_MATRIX_6X6
DFL	**	ONORBIT_SV_INTERP
DV_COV	NAV_ONORBIT_RENDEZVOUS,*	ONORBIT_SV_INTERP
GM_DEG	**	ONORBIT_SV_INTERP
GM_ORD	**	ONORBIT_SV_INTERP
G _TV	ONORBIT_REND_R_V_STATE_ PROP	MEAN_CONIC_PARTIAL_ TRANSITION_MATRIX_6X6
IDRAG	ONORBIT_REND_R_V_STATE_ PROP	ONORBIT_SV_INTERP
IGD	ONORBIT_REND_R_V_STATE_ PROP	ONORBIT_SV_INTERP
IGO	ONORBIT_REND_R_V_STATE_ PROP	ONORBIT_SV_INTERP
IVENT	ONORBIT_REND_R_V_STATE_ PROP	ONORBIT_SV_INTERP
PHI_PATCH	MEAN_CONIC_PARTIAL_ TRANSITION_MATRIX_6X6	REND_ANGLE_PARTIALS, RRDOT_NAV

*Onorbit/Rendezvous principal function, see section 4.2

**Initialization parameters, see section 4.7

TABLE 4.2.8.1.1.- REND_NAV_INTERP INPUT/OUTPUT.- Continued

Variable Name	Input Source	Output Destination
R _FILT	ONORBIT_REND_R_V_STATE_ PROP	ONORBIT_SV_INTERP, MEAN_CONIC_PARTIAL_ TRANSITION_MATRIX_6X6
R _LAST	COV_LAST_RESET,*	ONORBIT_SV_INTERP
R _RESID	ONORBIT_SV_INTERP	MEAN_CONIC_PARTIAL_ TRANSITION_MATRIX_6X6
R _TV	ONORBIT_REND_R_V_STATE_ PROP	ONORBIT_SV_INTERP, MEAN_CONIC_PARTIAL_ TRANSITION_MATRIX_6X6
R _TV_LAST	COV_LAST_RESET,*	ONORBIT_SV_INTERP
R _TV_RESID	ONORBIT_SV_INTERP	MEAN_CONIC_PARTIAL_ TRANSITION_MATRIX_6X6
SHUTTLE_FILTER_FLAG	**	
TOT_ACC	REND_BIAS_AND_COV_PROP	MEAN_CONIC_PARTIAL_ TRANSITION_MATRIX_6X6
V _FILT	ONORBIT_REND_R_V_STATE_ PROP	ONORBIT_SV_INTERP, MEAN_CONIC_PARTIAL_ TRANSITION_MATRIX_6X6
VFL_TV	**	ONORBIT_SV_INTERP
V _LAST	COV_LAST_RESET,*	ONORBIT_SV_INTERP
V _RESID	ONORBIT_SV_INTERP	MEAN_CONIC_PARTIAL_ TRANSITION_MATRIX_6X6
V _TV	ONORBIT_REND_R_V_STATE_ PROP	ONORBIT_SV_INTERP, MEAN_CONIC_PARTIAL_ TRANSITION_MATRIX_6X6
V _TV_LAST	COV_LAST_RESET,*	ONORBIT_SV_INTERP

*Onorbit/Rendezvous principal function, see section 4.2
**Initialization parameters, see section 4.7

TABLE 4.2.8.[illegible].1.- REND_NAV_INTERP INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
$\underline{V}$ _TV_RESID	ONORBIT_SV_INTERP	MEAN_CONIC_PARTIAL_ TRANSITION_MATRIX_6X6
$\underline{I}$ _RHO		REND_ANGLE_PARTIALS, RRDOT_NAV, RR_ANGLE_NAV
$\underline{R}$ _RHO		REND_ANGLE_PARTIALS, RRDOT_NAV
R_RHO_MAG		REND_NAV_FILTER, RRDOT_NAV
$\underline{V}$ _RHO		RRDOT_NAV
SENSOR_EPS	**	
SENSOR_DELTA	**	

** Initialization parameters, see section 4.7.

4.2.8.1.1.1. State vector interpolation (ONORBIT_SV_INTERP): The state vector interpolation subfunction shall provide the approximate position, velocity, and acceleration of either the Orbiter or target at a specified time within a given propagation interval given that the position, velocity and acceleration vectors are known at both ends of the interval.

The time at which vectors are desired is the time of an external sensor measurement, and the purpose of the interpolation is to enable the navigation filter to calculate measurement residuals at that time.

The method utilized for interpolation shall consist of defining a mean conic on the basis of positions and velocities of the vehicle in question at both ends of the propagation interval, and obtaining the desired vectors as if the vehicle moved along this mean conic. That is, a calculation shall be made to determine the point on the mean conic corresponding to the time of the measurement, and the velocity and position of such a point shall be taken as the state of the vehicle.

A. <u>Detailed Requirements</u>. This subfunction is called with the following internal variables in the IN LIST and the OUT LIST:

IN LIST: R _ONE, V _ONE, R _TWO, V _TWO, V _IMU_DIF, IGD, IGO, IDM, IVM, IATM

OUT LIST: R _RESID, V _RESID, A _RESID

where

R _ONE V _ONE	position and velocity of the vehicle at the previous propagation step;
R _TWO V _TWO	current position and velocity
V _IMU_DIF	difference between IMU accumulated sensed velocities at the current time and the previous time
IGD IGO IDM IVM IATM	flags for the call to the acceleration function ACCEL_ONORBIT (refer to section 4.2.3.1.4 for details of these flags)

The following steps shall be performed (in the order indicated):

1. A check of the absolute value of DELTAT_GO (where DELTAT_GO is the current filter time minus the time of the sensor measurement) against a premission-specified tolerance level will be performed:

$$|\text{DELTAT\_GO}| \le \text{EPS\_TIME}$$

a. If it is found that DELTAT_GO in absolute value is less than or equal to the tolerance, the values of the position and velocity of the vehicle at the current time shall be used as the state at the measurement time; the time tag at the measurement instant shall also be set equal to the current time:

$$\underline{R}\text{\_RESID} = \underline{R}\text{\_TWO}$$

$$\underline{V}\text{\_RESID} = \underline{V}\text{\_TWO}$$

$$\text{T\_RESID} = \text{T\_CURRENT\_FILT}$$

b. If, on the other hand, the difference between the time of the measurement and the current time exceeds the tolerance, perform the following:

(1) Certain parameters associated with the mean conic shall be obtained

$$\text{R\_TWO\_INV} = 1./|\underline{R}\text{\_TWO}| \qquad \text{F3}$$

$$\begin{aligned}\text{SMA} = 1./\big(&1./|\underline{R}\text{\_ONE}| + \text{R\_TWO\_INV} \\ &- (\underline{V}\text{\_ONE} \cdot \underline{V}\text{\_ONE} \\ &+ \underline{V}\text{\_TWO} \cdot \underline{V}\text{\_TWO})/(2.\ \text{EARTH\_MU})\big)\end{aligned} \qquad \text{F3}$$

$$\text{C1} = \text{SQRT (SMA)/SQR\_EMU} \qquad \text{F3,F4}$$

$$\text{D\_TWO} = \underline{R}\text{\_TWO} \cdot \underline{V}\text{\_TWO}$$

and the time tag of the state vector at measurement time shall be set:

$$\text{T\_RESID} = \text{T\_CURRENT\_FILT-DELTAT\_GO}$$

Additionally, set R_FIN_TEMP_INV = 0.

(2) The F and G series subfunction shall then be called (refer to section 4.2.5.2.1 for the description of this subfunction)

CALL: F_AND_G

F3 This equation shall be protected against division by zero (Reference 3.6-3).

F4 This equation shall be protected against square roots of a negative number (Reference 3.6-4).

IN LIST: SMA, - DELTAT_GO, C1, R _TWO, R_TWO_INV,
R_FIN_TEMP_INV, V _TWO, D_TWO, D_FIN_TEMP

OUT LIST: F, G, FDOT, GDOT, S0, S1, S2, S3,
R _RESID, R_FIN_INV, THETA

The position vector (R _RESID) comes out of this call; the velocity vector (V _RESID) does not, but it can be calculated on the basis of FDOT and GDOT, which are also obtained from the F and G series call:

V _RESID = FDOT R _TWO + GDOT V _TWO

2. Finally, the modeled acceleration shall be obtained by invoking the acceleration function with the position, velocity, and time (determined by the above process) in the calling arguments and adding the central force term. The sensed acceleration shall be found by dividing the difference in accumulated sensed velocities at both ends of the propagation interval by the duration of the interval. Total acceleration will be the sum of these two.

A _RESID = ACCEL_ONORBIT (IGD, IGO, IDM, IVM, IATM, R
_RESID, V _RESID, T_RESID) + V _IMU_DIF/DT_COV F3

(refer to section 4.2.4.1.1 for the description of this subfunction)

B. Interface Requirements. The input and output data are shown in table 4.2.8.1.1.1.

C. Processing Requirements. This subfunction is called by

REND_NAV_INTERP (section 4.2.8.1.1)

D. Constraints. None

E. Supplementary Information. A suggested implementation in the form of a detailed flowchart may be found in Appendix B under the name of ONORBIT_SV_INTERP.

F3 This equation shall be protected against division by zero (Reference 3.6-3).

TABLE 4.2.8.1.1.1.- ONORBIT_SV_INTERP INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
IATM	ATFL_OV	REND_NAV_INTERP	
IDM	IDRAG	REND_NAV_INTERP	
IGD	IGD	REND_NAV_INTERP	
IGO	IGO	REND_NAV_INTERP	
IVM	IVENT	REND_NAV_INTERP	
R_ONE	R_LAST	REND_NAV_INTERP	
R_TWO	R_FILT	REND_NAV_INTERP	
V_IMU_DIF	DV_COV	REND_NAV_INTERP	
V_ONE	V_LAST	REND_NAV_INTERP	
V_TWO	V_FILT	REND_NAV_INTERP	
IATM	ATFL_TV	REND_NAV_INTERP	
IDM	DFL	REND_NAV_INTERP	
IGD	GM_DEG	REND_NAV_INTERP	
IGO	GM_ORD	REND_NAV_INTERP	
IVM	VFL_TV	REND_NAV_INTERP	
R_ONE	R_TV_LAST	REND_NAV_INTERP	
R_TWO	R_TV	REND_NAV_INTERP	
V_IMU_DIF	0.	REND_NAV_INTERP	
V_ONE	V_TV_LAST	REND_NAV_INTERP	
V_TWO	V_TV	REND_NAV_INTERP	
A_RESID	A_RESID		REND_NAV_INTERP
R_RESID	R_RESID		REND_NAV_INTERP
V_RESID	V_RESID		REND_NAV_INTERP
A_RESID	A_TV_RESID		REND_NAV_INTERP
R_RESID	R_TV_RESID		REND_NAV_INTERP
V_RESID	V_TV_RESID		REND_NAV_INTERP

TABLE 4.2.8.1.1.1.- ONORBIT_SV_INTERP INPUT/OUTPUT.- Continued

Variable Name	Input Source	Output Destination
DELTAT_GO	COAS_NAV, RR_ANGLE_NAV, RRDOT_NAV, STAR_TRACKER_NAV	F_AND_G
DT_COV	REND_BIAS_AND_COV_PROP	
EARTH_MU	**	
EPS_TIME	**	
F	F_AND_G	
FDOT	F_AND_G	
G	F_AND_G	
GDOT	F_AND_G	
R_FIN_INV	F_AND_G	
R _RESID	F_AND_G	ACCEL_ONORBIT
SQR_EMU	*	
S0	F_AND_G	
S1	F_AND_G	
S2	F_AND_G	
S3	F_AND_G	
T_CURRENT_FILT	NAV_ONORBIT_RENDEZVOUS	
THETA	F_AND_G	
†	ACCEL_ONORBIT	
C1		F_AND_G

*Onorbit/Rendezvous principal function, see section 4.2
**Initialization parameters, see section 4.7
†Value returned from the function

TABLE 4.2.8.1.1.1.- ONORBIT_SV_INTERP INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
D_FIN_TEMP	**	F_AND_G
D_TWO		F_AND_G
IATM		ACCEL_ONORBIT
IDM		ACCEL_ONORBIT
IGD		ACCEL_ONORBIT
IGO		ACCEL_ONORBIT
IVM		ACCEL_ONORBIT
R_FIN_TEMP_INV		F_AND_G
R _TWO		F_AND_G
R_TWO_INV		F_AND_G
SMA		F_AND_G
T_RESID		ACCEL_ONORBIT
V _RESID		ACCEL_ONORBIT
V _TWO		F_AND_G

4.2.8.1.2 Kalman filter updates (REND_NAV_FILTER).- The Kalman Filter Updates subfunction shall be responsible for the processing of the rendezvous sensor measurement data. If the estimated range between the Orbiter and target vehicles is greater than or equal to a design dependent minimum range threshold, then the Kalman Filter Updates subfunction will perform the following major tasks.

- A bilevel residual edit test will be performed to determine whether the Kalman filter scheme will be used to update the state vector. This subfunction shall be able to respond to crew requests to relax the edit criteria thereby increasing the likelihood of incorporating the measurement data into the state vector.
- This subfunction shall be able to update the covariance matrix as well as the state vector by means of the Kalman update equations. The Kalman filter scheme shall be modified to allow for underweighting of the estimated sensor variance and the selective updating of the unmodeled acceleration bias states.
- Finally, the Kalman Filter Updates subfunction shall record the type of data processing that has occurred for crew display.

A. Detailed Requirements. For the measurement type to be processed on this cycle, test the magnitude of the relative position vector (R_RHO_MAG) against the minimum separation distance (RNG_MIN) to determine whether or not to exercise the Kalman filter update equations.

If R_RHO_MAG $<$ RNG_MIN, the EDIT_FLAG is to be set to OFF in order to blank the display. (Note: The measurement subfunction generates the partial vector, the residual, the magnitude of the relative position vector, and the a priori variance.) The logic then exits REND_NAV_FILTER without exercising the Kalman filter equations.

If R_RHO_MAG $\geq$ RNG_MIN, exercise the Kalman filter update equations, as follows:

1. Test the SHUTTLE_FILTER_FLAG. If the target is the filter vehicle (SHUTTLE_FILTER_FLAG = OFF), the sign is changed on the measurement partials vector.

$$B_{1 \text{ to } 6} = -B_{1 \text{ to } 6}$$

2. The scalar quantity BT_E_B is to be calculated from the covariance matrix E and vector measurement partials $\underline{B}$,

$$\underline{\text{EB\_COPY}} = E\ \underline{B}$$

$$\text{BT\_E\_B} = \underline{B} \cdot \underline{\text{EB\_COPY}},$$

where the second equation requires a dot product. The partials vector $\underline{B}$ shall then be set equal to zero so that subsequent measurement subroutines will only be required to calculate nonzero elements.

$\underline{B} = 0.$

The quantity MS_DELQ, which represents the expected variance in the residual, is then to be computed by

MS_DELQ = BT_E_B + VAR.

3. If the trace of the filter vehicle position portion of the covariance matrix ($E_{1,1} + E_{2,2} + E_{3,3}$) is greater than a threshold, MS_POS_UND_WGT, then the denominator of the Kalman gains is underweighted in order to improve the transient response of the filter.

   MS_DELQ = MS_DELQ + K_UND_WGT(BT_E_B)

4. The residual test quantity (RESID_TEST) shall be computed for the residual edit test and for display purposes,

   RESID_TEST = (K_RES_EDIT) MS_DELQ

   where K_RES_EDIT is a premission constant.

5. The residual edit test ratio (RESID_TEST_RATIO) is to be computed.

   RESID_TEST_RATIO = ABS(DELQ)/$\sqrt{\text{RESID\_TEST}}$. F3 F4

6. Test STAT_FLAG to determine if the residual and residual ratio have been computed on this cycle for display purposes only. If STAT_FLAG = ON, set

   EDIT_FLAG = STAT

   for display (section 4.2.9) and exit REND_NAV_FILTER. If STAT_FLAG is OFF, proceed to the next step in the logic.

7. Preparations for a residual edit test shall be performed. If the manual edit override is inactive, the test quantity (TEST_VALUE) shall be set [illegible]

   OLD is the value of the residual ratio for the [illegible] being processed as calculated by the state and covariance measurement incorporation subfuncton during the previous navigation cycle.

F3 This equation shall be protected against division by zero (Reference 3.6-3).
F4 This equation shall be protected against square roots of a negative number (Reference 3.6-4).

8. The edit test (RESID_TEST_RATIO $\leq$ TEST_VALUE) is performed.

   a. If the test fails, the edit flag is set to ON for crew information.

      EDIT_FLAG = ON

   b. If the measurements are to be incorporated, then update the state and the covariance matrix.

      EXECUTE: REND_STATE_AND_COV_UPDATE CODE

      (1) Compute the Kalman filter gain and update and symmetrize the covariance matrix.

$$\underline{\text{OMEGA}} = \underline{\text{EB}}\text{\_COPY}/\text{MS\_DELQ} \qquad \text{F3}$$

$$E_{I,I} = E_{I,I} - \text{OMEGA}_I \ \text{EB\_COPY}_I$$

$$E_{I,J} = E_{I,J} - \text{OMEGA}_I \ \text{EB\_COPY}_J \qquad \text{for } I = 1,12 \text{ and } J = I + 1,13$$

$$E_{J,I} = E_{I,J}$$

      Finally, $E_{13,13} = E_{13,13} - \text{OMEGA}_{13} \ \text{EB\_COPY}_{13}$.

      (2) Test the SHUTTLE_FILTER_FLAG.

      - If the Shuttle vehicle is the filter vehicle (SHUTTLE_FILTER_FLAG = ON), then this subfunction shall update the shuttle state vector by application of the following equations:

$$\underline{R}\text{\_FILT} = \underline{R}\text{\_FILT} + \text{OMEGA}_{1 \text{ to } 3} \ \text{DELQ}$$

$$\underline{V}\text{\_FILT} = \underline{V}\text{\_FILT} + \text{OMEGA}_{4 \text{ to } 6} \ \text{DELQ}$$

      - If the target vehicle is the filter vehicle (SHUTTLE_FILTER_FLAG = OFF), then this subfunction shall update the target state vector by application of the following equations:

$$\underline{R}\text{\_TV} = \underline{R}\text{\_TV} + \text{OMEGA}_{1 \text{ to } 3} \ \text{DELQ}$$

$$\underline{V}\text{\_TV} = \underline{V}\text{\_TV} + \text{OMEGA}_{4 \text{ to } 6} \ \text{DELQ}$$

      (3) If the unmodeled acceleration bias states are to be updated (UNMOD_ACC_BIAS_UPDATE_FLAG = ON), then this subfunction shall update the unmodeled acceleration bias states by application of the following equation:

F3 The equation shall be protected against division by zero (Reference 3.6-3).

$$\underline{\text{U}}\text{NMOD\_ACC\_BIAS} = \underline{\text{U}}\text{NMOD\_ACC\_BIAS} + \text{OMEGA}_{7 \text{ to } 9}\ \text{DELQ}$$

(4) The sensor bias states shall be updated by application of the following equation:

$$\underline{\text{S}}\text{ENSOR\_BIAS} = \underline{\text{S}}\text{ENSOR\_BIAS} + \text{OMEGA}_{10 \text{ to } 13}\ \text{DELQ}$$

where DELQ corresponds to the appropriate measurement residual.

After executing REND_STATE_AND_COV_UPDATE code, if the manual edit override is active (MANUAL_EDIT_OVERRIDE = ON), the edit flag is set to FORCED.

EDIT_FLAG = FORCED

Otherwise, set the edit flag to PROCESSED.

EDIT_FLAG = PROCESSED

Both the FORCED and the PROCESSED conditions result from measurement incorporation.

It is required that the residual, the residual test quantity (RESID_TEST_RATIO), and the residual edit flag corresponding to each measurement subfunction be saved for display purposes.

B. Interface Requirements. The inputs and outputs for this subfunction are given in table 4.2.8.1.2.

C. Processing Requirements. This subfunction is called by the following subfunctions:

ANGLE_NAV	(section 4.2.8.3.1)
RR_ANGLE_NAV	(section 4.2.8.2)
RRDOT_NAV	(section 4.2.8.1)

D. Constraints. None

E. Supplementary Information. A suggested implementation in the form of detailed flowcharts can be found in Appendix B under the following names:

REND_NAV_FILTER

REND_STATE_AND_COV_UPDATE CODE

TABLE 4.2.8.1.2.- REND_NAV_FILTER INPUT/OUTPUT

Variable Name	Input Source	Output Destination
B	RRDOT_NAV, RR_ANGLE_NAV, REND_ANGLE_PARTIALS, ANGLE_NAV	
DELTA_RESID_RATIO	**	
DELQ	RRDOT_NAV, RR_ANGLE_NAV, ANGLE_NAV	
E	REND_BIAS_AND_COV_PROP, REND_COV_INIT, COVINIT_UVW, U_A_BIAS_AND_COVINIT, SETUP	REND_BIAS_AND_COV_PROP,*
EDIT_FLAG		RRDOT_NAV, RR_ANGLE_NAV, ANGLE_NAV
K_RESID_EDIT	**	
K_UND_WGT	**	
MANUAL_EDIT_OVERRIDE	RRDOT_NAV, RR_ANGLE_NAV, ANGLE_NAV	
MS_POS_UND_WGT	**	
RESID_RATIO_OLD	RRDOT_NAV, RR_ANGLE_NAV, ANGLE_NAV	
RESID_TEST_RATIO		RRDOT_NAV, RR_ANGLE_NAV, ANGLE_NAV
R _FILT	ONORBIT_REND_R_V_STATE_PROP	COV_LAST_RESET, SHUTTLE_RESET, ONORBIT_REND_R_V_STATE_PROP,*,REL_NAV_DISPLAY_UPDATES,REND_COV_INIT,NAV_ONORBIT_RENDEZVOUS

*Onorbit/Rendezvous principal function, see section 4.2

**Initialization parameters, see section 4.7

TABLE 4.2.8.1.2.- REND_NAV_FILTER INPUT/OUTPUT.- Continued

Variable Name	Input Source	Output Destination
RNG_MIN	**	
R_RHO_MAG	REND_NAV_INTERP	
R _TV	ONORBIT_REND_R_V_STATE_PROP	COV_LAST_RESET, TARGET_RESET, ONORBIT_REND_R_V_STATE_PROP,*,REL_NAV_DISPLAY_UPDATES,REND_COV_INIT,NAV_ONORBIT_RENDEZVOUS
SENSOR_BIAS	SETUP	RR_ANGLE_NAV, RRDOT_NAV, ANGLE_NAV,NAV_ONORBIT_RENDEZVOUS
SHUTTLE_FILTER_FLAG	**	
STAT_FLAG	RRDOT_NAV, RR_ANGLE_NAV, ANGLE_NAV	
UNMOD_ACC_BIAS	REND_BIAS_AND_COV_PROP, U_A_BIAS_AND_COVINIT	ACCEL_ONORBIT, REND_BIAS_AND_COV_PROP, NAV_ONORBIT_RENDEZVOUS
UNMOD_ACC_UPDATE_FLAG	**	
VAR	RRDOT_NAV, RR_ANGLE_NAV, ANGLE_NAV	
V _FILT	ONORBIT_REND_R_V_STATE_PROP	COV_LAST_RESET,SHUTTLE_RESET, ONORBIT_REND_R_V_STATE_PROP,*,REL_NAV_DISPLAY_UPDATES,REND_COV_INIT,NAV_ONORBIT_RENDEZVOUS

*Onorbit/Rendezvous principal function, see section 4.2
**Initialization parameters, see section 4.7

TABLE 4.2.8.1.2.- REND_NAV_FILTER INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
V _TV	ONORBIT_REND_R_V_STATE_PROP	COV_LAST_RESET, TARGET_RESET, ONORBIT_REND_R_V_STATE_PROP,*,REL_NAV_DISPLAY_UPDATES, REND_COV_INIT,NAV_ONORBIT_RENDEZVOUS

*Onorbit/Rendezvous principal function, see section 4.2.

4.2.8.2 Rendezvous Radar Shaft and Trunnion Angle Measurements (RR_ANGLE_NAV)

This subfunction is responsible for the proper processing of the rendezvous radar shaft and trunnion angle measurements. This subfunction shall perform the following tasks only when the rendezvous radar is not in the self test mode and the measurement data is labeled valid.

- Calculate the partial derivative of the measurement with respect to the estimated state at measurement time.
- Compute the estimated measurement and the measurement residual.
- Select the proper variances to model the uncorrelated measurement errors.
- Store the old residual ratio, the EDIT OVERRIDE flag, and the STAT flags into temporary locations used by the Kalman Filter Updates subfunction (section 4.2.8.1.2).
- Schedule the Kalman Filter Updates subfunction to process the data.
- Store the current residual ratio, the EDIT flag and the measurement residual for display purposes.

A. Detailed Requirements.- Process the rendezvous radar angle data only if the data are valid and the rendezvous radar is not in the self test mode (SELF_TEST_FLAG = OFF). The following steps shall be performed in the order indicated.

1. Compute the mean of 1950 to sensor coordinate transformation matrix.

   M_M50_TO_SENSOR = M_BODY_TO_RR QUAT_TO_MAT(Q_M50BODY_RR)

2. Compute the time difference between current filter time and the time of the rendezvous radar angle data measurements.

   DELTAT_GO = T_CURRENT_FILT - T_REND_RADAR

3. Call the measurement interpolation subfunction and interpolate the Orbiter and target state vectors to the time of the shaft angle measurement (see section 4.2.8.1.1)

   CALL: REND_NAV_INTERP

4. Call the angle partials subfunction to compute the partial vector (see section 4.2.8.2.1)

   CALL: REND_ANGLE_PARTIALS

   IN LIST: $-M\_M50\_TO\_SENSOR_{3,1 \text{ to } 3}$

5. Calculation of the partial vector is completed by setting the appropriate value in the bias slot of that vector.

$B_{10} = 1.0$

6. Compute the estimated shaft angle measurement and the shaft angle measurement residual.

   SHAFT = ARCTAN2 (-U_M$_2$, U_M$_1$) + SENSOR_BIAS$_1$ [F7]

   DELQ = Q_RR_SHFT RAD_PER_DEG - SHAFT

7. If measurement residual (DELQ) falls outside the range $-\pi$ to π, adjust DELQ such that it falls inside the range.

   If DELQ > 0. DELQ = DELQ - 2 π

   DELQ $\leq$ 0. DELQ = DELQ + 2 π

8. Assign the appropriate variance for the rendezvous radar shaft angle.

   VAR = VAR_SHAFT

9. The residual test ratio from the previous filter cycle and the measurement processing control flags shall be set as follows:

   RESID_RATIO_OLD = MAX (NAV_SIG$_1$, NAV_SIG$_2$)
   MANUAL_EDIT_OVERRIDE = RR_ANGLES_EDIT_OVERRIDE
   STAT_FLAG = RR_ANGLES_STAT

   (where RR_ANGLES_EDIT_OVERRIDE and RR_ANGLES_STAT come from the sensor measurement selection subfunction (section 4.2.6) and NAV_SIG comes from the previous execution of this subfunction as given in step 11).

10. Call the Kalman filter subfunction to process the rendezvous radar shaft angle measurement (see section 4.2.8.1.2)

    CALL: REND_NAV_FILTER

11. Store the output data from the Kalman filter subfunction in the appropriate variables for use by the measurement processing statistics subfunction.

    SENSOR_EDIT$_1$ = EDIT_FLAG

    NAV_SIG$_1$ = RESID_TEST_RATIO

    SENSOR_DELQ$_1$ = DELQ

[F7] This equation shall be protected against arc tangents with both arguments equal to zero (Reference 3.6-7).

12. Call the measurement interpolation subfunction to interpolate the Orbiter and target state vectors to the time of the trunnion angle measurement (see section 4.2.8.1.1)

    CALL: REND_NAV_INTERP

13. Call the angle partials subfunction to compute the partial vector (see section 4.2.8.2.1)

    CALL: REND_ANGLE_PARTIALS

    IN LIST: $\text{UNIT}(\underline{\text{I}}\text{\_RHO} \times \text{M\_M50\_TO\_SENSOR}_{3,1\text{to}3})$

14. Calculation of the partial vector is completed by setting the appropriate value in the bias slot of that vector.

    $\text{B}_{11} = 1.0$

15. Compute the estimated trunnion angle measurement and the trunnion angle measurement residual.

    $\text{TRUN} = \text{ARCSIN}(\text{U\_M}_3) + \text{SENSOR\_BIAS}_2$ F5

    $\text{DELQ} = \text{Q\_RR\_TRUN RAD\_PER\_DEG} - \text{TRUN}$

16. Assign the appropriate variance for the rendezvous radar trunnion angle

    VAR = VAR_TRUN

17. Call the Kalman filter to process the rendezvous radar trunnion angle measurement (see section 4.2.8.1.2)

    CALL: REND_NAV_FILTER

B. Interface Requirements.- The input and output variables for this subfunction are defined in table 4.2.8.2.

F5 This equation shall be protected against arc sine of arguments with magnitudes greater than unity (Reference 3.6-5).

C. Processing Requirements.- This subfunction is called by NAV_ONORBIT_RENDEZVOUS (section 4.2.1).

D. Constraints.- None

E. Supplementary Information.- A suggested implementation in the form of a detailed flowchart can be found in Appendix B under the name RR_ANGLE_NAV.

TABLE 4.2.8.2.- RR_ANGLE_NAV INPUT/OUTPUT

Variable Name	Input Source	Output Destination
EDIT_FLAG	REND_NAV_FILTER	
I _RHO	REND_NAV_INTERP	
M_BODY_TO_RR	**	
NAV_SIG	SETUP	MEAS_PROCESSING_ STATISTICS_REND
PI	**	
Q _M50BODY_RR	NAV_ONORBIT_RENDEZVOUS	QUAT_TO_MAT
Q_RR_SHFT	NAV_ONORBIT_RENDEZVOUS	
Q_RR_TRUN	NAV_ONORBIT_RENDEZVOUS	
RAD_PER_DEG	**	
RESID_TEST_RATIO	REND_NAV_FILTER	
RR_ANGLE_DATA_GOOD	NAV_ONORBIT_RENDEZVOUS	
RR_ANGLES_EDIT_OVERRIDE	REND_SENSOR_SELECT	
RR_ANGLES_STAT	REND_SENSOR_SELECT	
SELF_TEST_FLAG	NAV_ONORBIT_RENDEZVOUS	
SENSOR_BIAS	SETUP,REND_NAV_FILTER	
T_CURRENT_FILT	NAV_ONORBIT_RENDEZVOUS	

**Initialization parameters, see section 4.7

TABLE 4.2.8.2- RR_ANGLE_NAV INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
T_REND_RADAR	NAV_ONORBIT_RENDEZVOUS	
U_M	REND_ANGLE_PARTIALS	
VAR_SHAFT	**	
VAR_TRUN	**	
B		REND_NAV_FILTER
DELQ		REND_NAV_FILTER
DELTAT_GO		REND_NAV_INTERP, ONORBIT_SV_INTERP
MANUAL_EDIT_OVERRIDE		REND_NAV_FILTER
M_M50_TO_SENSOR		REND_ANGLE_PARTIALS
RESID_RATIO_OLD		REND_NAV_FILTER
SENSOR_DELQ		MEAS_PROCESSING_ STATISTICS_REND
SENSOR_EDIT		MEAS_PROCESSING_ STATISTICS_REND,*
STAT_FLAG		REND_NAV_FILTER
VAR		REND_NAV_FILTER
†	QUAT_TO_MAT	
UNIT(I_RHO x M_M50_TO_SENSOR$_{3,1to3}$)		REND_ANGLE_PARTIALS

---

*Onorbit/Rendezvous principal function, see section 4.2

**Initialization parameters, see section 4.7

†Value returned from the function

4.2.8.2.1 Angle partials (REND_ANGLE_PARTIALS)

The angle partials subfunction (REND_ANGLES_PARTIALS) is a utility subfunction whose purpose is to compute the angle measurement partial derivatives, with respect to the Orbiter position and velocity, for all sensor angle measurements. The partials vector is used by the Kalman filter update subfunction, REND_NAV_FILTER (section 4.2.8.1.2).

A. Detailed Requirements.-

1. First, we compute $\underline{\text{R}}\text{HO\_PLANE}$, which is the projection of the Shuttle/target relative position vector, $\underline{\text{R}}\text{\_RHO}$, into the orthogonal complement plane of the axis of rotation of the angle measurement,

$$\underline{\text{R}}\text{HO\_PLANE} = \underline{\text{R}}\text{\_RHO} - (\underline{\text{R}}\text{\_RHO} \cdot \underline{\text{I}}\text{\_N})\underline{\text{I}}\text{\_N},$$

where $\underline{\text{I}}\text{\_N}$ is a unit vector along the axis of rotation.

2. Next, the partial derivative of the angle measurement with respect to the Shuttle position and velocity is computed.

$$B_{1\ \text{to}\ 6} = (\text{PHI\_PATCH}_{1\ \text{to}\ 3,\ 1\ \text{to}\ 6})^T \cdot \Big(\text{UNIT}(\underline{\text{R}}\text{HO\_PLANE} \times \underline{\text{I}}\text{\_N}) / |\underline{\text{R}}\text{HO\_PLANE}|$$ F3

PHI_PATCH is the position-velocity part of the state transition matrix calculated in the measurement interpolation subfunction, REND_NAV_INTERP (section 4.2.8.1.1).

3. Finally, the unit vector in the line of sight direction, $\underline{\text{I}}\text{\_RHO}$, is rotated into sensor coordinates

$$\underline{\text{U}}\text{\_M} = \text{M\_M50\_TO\_SENSOR}\ \underline{\text{I}}\text{\_RHO}$$

B. Interface Requirements.- Input and output parameters are listed in table 4.2.8.2.1.

C. Processing Requirements.- This subfunction is called by the following subfunctions:

ANGLE_NAV (section 4.2.8.3.1)
RR_ANGLE_NAV (section 4.2.8.2)

D. Constraints.- None

E. Supplemental Information.- A suggested implementation of the angle partials subfunction may be found in the Appendix B flowchart REND_ANGLE_PARTIALS.

F3 This equation shall be protected against division by zero (Reference 3.6-3).

TABLE 4.2.8.2.1.- REND_ANGLE_PARTIALS INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
I _N	-M_M50_TO_SENSOR$_{3,1to3}$	RR_ANGLE_NAV	
I _N	UNIT(I _RHO x M_M50_TO_SENSOR$_{3,1to3}$)	RR_ANGLE_NAV	
I _N	-M_M50_TO_SENSOR$_{2,1to3}$	ANGLE_NAV	
I _N	-M_M50_TO_SENSOR$_{1,1to3}$	ANGLE_NAV	

---

[*] Onorbit/Rendezvous principal function, see section 4.2

[**] Initialization parameters, see section 4.7

TABLE 4.2.8.2.1.- REND_ANGLE_PARTIALS INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
$\underline{R}$_RHO	REND_NAV_INTERP	
PHI_PATCH	REND_NAV_INTERP	
M_M50_TO_SENSOR	RR_ANGLE_NAV, STAR_TRACKER_NAV, COAS_NAV	
$\underline{U}$_M		RR_ANGLE_NAV, ANGLE_NAV
$\underline{B}$		REND_NAV_FILTER
$\underline{I}$_RHO	REND_NAV_INTERP	

4.2.8.3 Star Tracker Horizontal and Vertical Measurements (STAR_TRACKER_NAV)

This subfunction is responsible for the proper processing of the star tracker vertical and horizontal measurement data. This subfunction shall perform the following tasks provided the star tracker is in the target track mode and the angle data are valid.

- Store the mean of '50 to star tracker transformation matrix for use in the angle measurements subfunction.
- Compute the time difference between current filter time and measurement time.
- Invoke the angle measurement subfunction for the processing of the angle data with the proper IN LIST arguments.

If the star tracker time tag is too close in value to the star tracker time tag on the last filter cycle the angle measurement subfunction is not invoked in order to avoid the processing of the same measurement twice.

A. <u>Detailed Requirements</u>.- If the star tracker data is good (ST_DATA_GOOD = ON) and the star tracker is in the target track mode (TRG_TRK_MODE = ON), then the following steps shall be performed in the order indicated.

1. Store the mean of 1950 to star tracker transformation matrix into the mean of 1950 to sensor transformation matrix for use in the angle measurements subfunction (section 4.2.8.3.1)

   M_M50_TO_SENSOR = M_M50_TO_ST

2. Compute both the time difference between the current time and the star tracker measurement time and the time difference between the current star tracker measurement time and the time of the last processed measurement.

   DELTAT_GO = T_CURRENT_FILT-T_STAR_TRACKER
   DELTAT_ST = T_STAR_TRACKER-T_ST_LAST

3. Test the time difference between the current star tracker measurement time and the time of the last processed measurement to avoid the redundant processing of the star tracker data.

   If DELTAT_ST is larger than a premission time difference, then

   a. Invoke the angle measurement subfunction to incorporate the star tracker angle data into the filter vehicle state.

      CALL: ANGLE_NAV

IN LIST: VAR_ST_HORIZ, Q_ST_HORIZ, VAR_ST_VERT,
Q_ST_VERT, ST_ANGLES_EDIT_OVERRIDE,
ST_ANGLES_STAT

b. Set the time of the last Star tracker measurement.

T_ST_LAST = T_STAR_TRACKER

B. Interface Requirements.- The input and output variables for this subfunction are defined in table 4.2.8.3.

C. Processing Requirements.- This subfunction is called by NAV_ONORBIT_RENDEZVOUS (section 4.2.1).

D. Constraints.- None

E. Supplementary Information.- A suggested implementation in the form of a detailed flowchart can be found in Appendix B under the name STAR_TRACKER_NAV.

TABLE 4.2.8.3.- STAR_TRACKER_NAV INPUT/OUTPUT

Variable Name	Input Source	Output Destination
M_M50_TO_ST	NAV_ONORBIT_RENDEZVOUS	
Q_ST_HORIZ	NAV_ONORBIT_RENDEZVOUS	ANGLE_NAV
Q_ST_VERT	NAV_ONORBIT_RENDEZVOUS	ANGLE_NAV
ST_ANGLES_EDIT_OVERRIDE	REND_SENSOR_SELECT	ANGLE_NAV
ST_ANGLES_STAT	REND_SENSOR_SELECT	ANGLE_NAV
ST_DATA_GOOD	NAV_ONORBIT_RENDEZVOUS	
T_CURRENT_FILT	NAV_ONORBIT_RENDEZVOUS	
TRG_TRK_MODE	NAV_ONORBIT_RENDEZVOUS	
T_STAR_TRACKER	NAV_ONORBIT_RENDEZVOUS	
VAR_ST_HORIZ	**	ANGLE_NAV
VAR_ST_VERT	**	ANGLE_NAV
DELTAT_GO		REND_NAV_INTERP, ONORBIT_SV_INTERP
M_M50_TO_SENSOR		ANGLE_NAV, REND_ANGLE_PARTIALS
DELTAT_ST_MIN	**	

**Initialization of parameters, see section 4.7

4.2.8.3.1 Angle measurements (ANGLE_NAV)

This utility subfunction is responsible for the proper processing of vertical and horizontal angle measurements taken by either the COAS or star tracker. This subfunction shall perform the following tasks when called by either the COAS or star tracker vertical and horizontal measurements subfunction (see sections 4.2.8.4 and 4.2.8.3, respectively):

- Calculate the partial derivative of the measurement with respect to the estimated state at measurement time.
- Compute the estimated measurement and the measurement residual.
- Select the proper variances to model the uncorrelated measurement errors.
- Store the old residual ratio, the EDIT OVERRIDE flag, and the STAT flags into temporary locations used by the Kalman filter Updates subfunction (section 4.2.8.1.2).
- Schedule the Kalman filter Updates subfunction to process the data.
- Store the current residual ratio, the EDIT flag, and the measurement residual for display purposes.

A. Detailed Requirements.- This subfunction is called with the following internal variables in the IN LIST:

IN LIST: VAR_HORIZ, Q_HORIZ, VAR_VERT, Q_VERT, ANGLES_MANUAL_EDIT_OVERRIDE, ANGLES_STAT_FLAG

where

VAR_HORIZ	variance of the horizontal measurement
Q_HORIZ	horizontal angle measurement
VAR_VERT	variance of the vertical measurement
Q_VERT	vertical angle measurement
ANGLES_MANUAL_EDIT_OVERRIDE	manual edit override flag
ANGLES_STAT_FLAG	stat flag

The following steps shall be performed (in the order indicated):

1. Call the measurement interpolation subfunction to interpolate the Orbiter and target state vectors to the time of the vertical angle measurement (see section 4.2.8.1.1).

   CALL: REND_NAV_INTERP

2. Call the angle partials subfunction to compute the partial vector (see section 4.2.8.2.1)

   CALL: REND_ANGLE_PARTIALS

   IN LIST: $-M\_M50\_TO\_SENSOR_{2,1 \text{ to } 3}$

3. Calculation of the partial vector is completed by setting the appropriate value in the bias slot of that vector:

   $B_{11} = 1.0$

4. Compute the estimated vertical angle measurement and the vertical angle measurement residual.

   $VERT = ARCTAN2(-U\_M_1, U\_M_3) + SENSOR\_BIAS_2$ F7

   $DELQ = Q\_VERT - VERT$

5. Assign the appropriate variance for the vertical angle

   VAR = VAR_VERT

6. Set up the required inputs to the Kalman filter subfunction:

   $RESID\_RATIO\_OLD = MAX\ (NAV\_SIG_1, NAV\_SIG_2)$

   MANUAL_EDIT_OVERRIDE = ANGLES_MANUAL_EDIT_OVERRIDE

   STAT_FLAG = ANGLES_STAT_FLAG

7. Call the Kalman filter subfunction to process the vertical angle measurement (see section 4.2.8.1.2).

   CALL: REND_NAV_FILTER

8. Store the output data from the Kalman filter subfunction in the appropriate variables for use by the measurement processing statistics subfunction.

   $SENSOR\_DELQ_2 = DELQ$

   $NAV\_SIG_2 = RESID\_TEST\_RATIO$

   $SENSOR\_EDIT_2 = EDIT\_FLAG$

F7 This equation shall be protected against arc tangents with both arguments equal to zero (Reference 3.6-7).

9. Call the measurement interpolation subfunction to interpolate the Orbiter and target state vectors to the time of the horizontal angle measurement (see section 4.2.8.1.1).

   CALL: REND_NAV_INTERP

10. Call the angle partials subfunction to compute the partial vector (see section 4.2.8.2.1).

    CALL: REND_ANGLE_PARTIALS

    IN LIST: -M_M50_TO_SENSOR$_{1}$, 1 to 3

11. Calculation of the partial vector is completed by setting the appropriate value in the bias slot of that vector:

    $B_{10} = 1.0$

12. Compute the estimated horizontal angle measurement and the horizontal angle measurement residual.

    HORIZ = ARCTAN2 (U_M$_2$,U_M$_3$) + SENSOR_BIAS$_1$ F7

    DELQ = Q_HORIZ - HORIZ

13. Assign the appropriate variance for the horizontal angle

    VAR = VAR_HORIZ

14. Call the Kalman filter subfunction to process the horizontal angle measurement (see section 4.2.8.1.2).

    CALL: REND_NAV_FILTER

15. Store the output data from the Kalman filter subfunction in the appropriate variables for use by the measurement processing statistics subfunction.

    SENSOR_DELQ$_1$ = DELQ

    NAV_SIG$_1$ = RESID_TEST_RATIO

    SENSOR_EDIT$_1$ = EDIT_FLAG

B. Interface Requirements.- The input and output variables for this subfunction are defined in table 4.2.8.3.1.

F7 This equation shall be protected against arc tangents with both arguments equal to zero (Reference 3.6-7).

C. Processing Requirements.- This subfunction is called by the following subfunctions:

COAS_NAV	(section 4.2.8.4)
STAR_TRACKER_NAV	(section 4.2.8.3)

D. Constraints.- None

E. Supplementary Information.- A suggested implementation of this subfunction in the form of a detailed flowchart can be found in Appendix B under the name

ANGLE_NAV.

TABLE 4.2.8.3.1.- ANGLE_NAV INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
ANGLES_MANUAL_EDIT_OVERRIDE	COAS_ANGLES_EDIT_OVERRIDE	COAS_NAV	
ANGLES_STAT_FLAG	COAS_ANGLES_STAT	COAS_NAV	
Q_HORIZ	Q_COAS_HORIZ	COAS_NAV	
Q_VERT	Q_COAS_VERT	COAS_NAV	
VAR_HORIZ	VAR_COAS_HORIZ	COAS_NAV	
VAR_VERT	VAR_COAS_VERT	COAS_NAV	
ANGLES_MANUAL_EDIT OVERRIDE	ST_ANGLES_EDIT_OVERRIDE	STAR_TRACKER_NAV	
ANGLES_STAT_FLAG	ST_ANGLES_STAT	STAR_TRACKER_NAV	
Q_HORIZ	Q_ST_HORIZ	STAR_TRACKER_NAV	
Q_VERT	Q_ST_VERT	STAR_TRACKER_NAV	
VAR_HORIZ	VAR_ST_HORIZ	STAR_TRACKER_NAV	
VAR_VERT	VAR_ST_VERT	STAR_TRACKER_NAV	

TABLE 4.2.8.3.1.- ANGLE_NAV INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
EDIT_FLAG	REND_NAV_FILTER	
M_M50_TO_SENSOR	COAS_NAV, STAR_TRACKER_NAV	REND_ANGLE_PARTIALS
NAV_SIG	SETUP	MEAS_PROCESSING_ STATISTICS_REND
RESID_TEST_RATIO	REND_NAV_FILTER	
SENSOR_BIAS	SETUP,REND_NAV_FILTER	
U _M	REND_ANGLE_PARTIALS	
B		REND_NAV_FILTER
DELQ		REND_NAV_FILTER
MANUAL_EDIT_OVERRIDE		REND_NAV_FILTER
RESID_RATIO_OLD		REND_NAV_FILTER
SENSOR_DELQ		MEAS_PROCESSING_ STATISTICS_REND
SENSOR_EDIT		MEAS_PROCESSING _STATISTICS_REND,*
STAT_FLAG		REND_NAV_FILTER
VAR		REND_NAV_FILTER

---

*Onorbit/Rendezvous principal function, see section 4.2

C-4

4.2.8.4 COAS Horizontal And Vertical Angle Measurements (COAS_NAV)

This subfunction is responsible for the proper processing of the COAS horizontal and vertical angle measurements. This subfunction shall perform the following task if the COAS data are labeled valid.

- Compute the time difference between current filter time and measurement time. Also we will compute the time difference (DELTAT_COAS) between the current COAS measurement and the measurement last used by this subfunction.
- If DELTAT_COAS satisfies criteria for staleness and is not greater than a design dependent threshold, the angle measurement subfunction shall be invoked to process the vertical and horizontal angle data.
- If the COAS horizontal and vertical angle measurement subfunction is not to process the angle data for statistical display purposes only, then the time of the last COAS measurement is reset.

A. Detailed Requirements.- If the COAS data is good (COAS_DATA_GOOD = ON), then the following steps shall be performed (in the order indicated):

1. Determine the delta time between the current time and the time of the COAS measurements and also determine the delta time since the last processing of COAS data.

   DELTAT_GO = T_CURRENT_FILT - T_COAS
   DELTAT_COAS = T_COAS - T_COAS_LAST

2. This subfunction will only process COAS data if the COAS data have not been previously processed and the time elapsed since the COAS data snap is smaller than a design dependent time delta.

   i.e., if DELTAT_COAS > DELTAT_COAS_MIN

   AND

   DELTAT_GO < DELTAT_COAS_MAX

   a. Compute the mean of 1950 to sensor coordinate transformation matrix.

      $M\_M50\_TO\_SENSOR = M\_BODY\_TO\_COAS_{COAS\_ID}\ M\_M50\_TO\_BODY\_COAS$

   b. Call the angle measurements subfunction to incorporate the COAS angle data (see section 4.2.8.3.1).

      CALL: ANGLE_NAV

      IN LIST: VAR_COAS_HORIZ, Q_COAS_HORIZ,
      VAR_COAS_VERT, Q_COAS_VERT,
      COAS_ANGLES_EDIT_OVERRIDE,
      COAS_ANGLES_STAT

c. If the COAS data were processed (COAS_ANGLES_STAT = OFF), save the time of the COAS data for use on the next filter subcycle.

T_COAS_LAST = T_COAS

B. Interface Requirements.- The input and output variables for this subfunction are defined in table 4.2.8.4.

C. Processing Requirements.- This subfunction is called by NAV_ONORBIT_RENDEZVOUS (section 4.2.1).

D. Constraints.- None

E. Supplementary Information.- A suggested implementation in the form of a detailed flowchart can be found in Appendix B under the name COAS_NAV.

TABLE 4.2.8.4.- COAS_NAV INPUT/OUTPUT

Variable Name	Input Source	Output Destination
COAS_ANGLES_EDIT_OVERRIDE	REND_SENSOR_SELECT	ANGLE_NAV
COAS_ANGLES_STAT	REND_SENSOR_SELECT	ANGLE_NAV
COAS_DATA_GOOD	NAV_ONORBIT_RENDEZVOUS	
COAS_ID	NAV_ONORBIT_RENDEZVOUS	
DELTAT_COAS_MAX	**	
DELTAT_COAS_MIN	**	
M_BODY_TO_COAS	**	
M_M50_TO_BODY_COAS	NAV_ONORBIT_RENDEZVOUS	
Q_COAS_HORIZ	NAV_ONORBIT_RENDEZVOUS	ANGLE_NAV
Q_COAS_VERT	NAV_ONORBIT_RENDEZVOUS	ANGLE_NAV
T_COAS	NAV_ONORBIT_RENDZVOUS	
T_COAS_LAST	**	
T_CURRENT_FILT	NAV_ONORBIT_RENDEZVOUS	
VAR_COAS_HORIZ	**	ANGLE_NAV
VAR_COAS_VERT	**	ANGLE_NAV

**Initialization parameters, see section 4.7

TABLE 4.2.8.4.- COAS_NAV INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
DELTAT_GO		REND_NAV_INTERP, ONORBIT_SV_INTERP
M_M50_TO_SENSOR		ANGLE_NAV, REND_ANGLE_PARTIALS

4.2.9 Measurement Processing Statistics (MEAS_PROCESSING_STATISTICS_REND)

During rendezvous navigation phases that utilize external measurements, the measurement processing statistics subfunction will compute for display certain parameters that are indicative of the condition of the navigation filter and the external sensor measurements that it utilizes. These display parameters serve as the basis for the crew decision as to how external measurement data are to be processed by the nav filter.

The measurement processing statistics subfunction will be performed after the state and covariance measurement incorporation subfunctions have been performed. Filter edit indicators, which will have been initialized to a default value during the sensor measurement selection subfunction, will be redefined during performance of the state and covariance measurement incorporation subfunctions. This will indicate to the measurement processing statistics subfunction, for each measurement type being utilized, which of the following five cases has occurred:

- Edit indicator = OFF - The filter was not configured for the measurement type or the data were bad and the filter did not attempt to process data of that type.
- Edit indicator = ON - The filter did attempt to process the measurement type but automatically edited the data.
- Edit indicator = PROCESSED - The filter processed the measurement type as a result of the data satisfying the edit criterion.
- Edit indicator = STAT - The filter was used solely for producing the residual and ratio parameters for display.
- Edit indicator = FORCED - The filter processed the data as a result of a crew edit override.

Moreover, the state and covariance measurement incorporation subfunction will provide the measurement processing statistics subfunction with the value of each measurement residual and the corresponding residual edit ratio value. The data supplied to the measurement processing statistics subfunction are used to compute statistics for the sensor measurement type selected.

For each measurement type, the following parameters are to be computed for display to show how well the navigation filter is processing external measurements of that particular type:

$DISP\_DELQ_I$ - The actual measurement residual computed by the nav filter for the Ith measurement type.

$DISP\_SIG_I$ - The residual edit ratio for the Ith measurement type. The ratio is the absolute value of the measurement residual divided by the square root of the scaled value of the residual variance for the measurement type. (See section 4.2.8.1.2 for the definition of RESID_TEST_RATIO.)

N_ACCEPT$_I$ - The number of data marks for the Ith measurement type which have been used to update the nav state vector.

N_REJECT$_I$ - The number of data marks for the Ith measurement type which have been automatically rejected as a result of failing the nav filter edit criterion.

DISP_EDIT$_I$ - The status indicator which shall be displayed as a BLANK unless the nav filter has edited a predetermined number of sequential data marks for the Ith type. In this case, the status indicator shall be displayed as the symbol, ↓. Once set, the down arrow symbol shall continue to be displayed until a predetermined number of sequential data marks have been processed by the nav filter, or until a data mark has been processed by the nav filter as a result of the crew setting the appropriate AUTO/INHIBIT/FORCE flag to FORCE.

If valid data were not presented to the filter or if the estimated distance between vehicles is less than a specified value (RANGE_MIN), then these parameters will maintain the values defined during the previous filter subcycle.

The ACCEPT/REJECT counters are initialized to zero whenever the covariance matrix is reinitialized, when leaving rendezvous navigation, when the corresponding sensor type is changed, or when the IMU sensed accelerations exceed a premission specified amount (MEAS_THRESHOLD).

Sensor data will consist of two types: angular data and range data. The angular data will consist of a pair of angles from one of three mutually exclusive sources - COAS, star tracker (ST), or rendezvous radar (RR). The range data will consist of range and range rate from the rendezvous radar. Angular data, from whichever source has been chosen, can be utilized in conjunction with range data.

A. Detailed Requirements.- The correspondence between the measurement type and the subscript, I, shall be as follows:

I = 1 - COAS horizontal angle, ST horizontal angle, or RR shaft

For each value of the integer I in the interval [1,4], the following steps shall be performed (in the order indicated) when SENSOR_EDIT$_I$ ≠ OFF:

1. DISP_DELQ$_I$ and DISP_SIG$_I$ are given the values:

$DISP\_DELQ_I = SENSOR\_DELQ_I$

$DISP\_SIG_I = NAV\_SIG_I$

where $SENSOR\_DELQ_I$ and NAV_SIG_I were computed in the state and covariance measurement incorporation subfunctions (sections 4.2.8.1 through 4.2.8.4).

2. For the range measurement,

   $I = 3,$

   the residual is converted to kilofeet:

   $DISP\_DELQ_I = DISP\_DELQ_I/1000.0$

3. For the angle measurements,

   $I = 1$ or $2,$

   the residuals are converted to degrees:

   $DISP\_DELQ_I = DISP\_DELQ_I\ DEG\_PER\_RAD$

4. Test the S̲ENSOR_EDIT value for the Ith measurement type

   a. If $SENSOR\_EDIT_I = STAT$, $DISP\_EDIT_I$ shall be given the value BLANK:

      $DISP\_EDIT_I = BLANK$

      The logic then exits from MEAS_PROCESSING_STATISTICS_REND.

   b. If $SENSOR\_EDIT_I \neq STAT$, then test S̲ENSOR_EDIT again.

      (1) If $\underline{S}ENSOR\_EDIT_I = ON$, the sequential accept counter shall be set to zero:

          $SEQ\_ACCEPT_I = 0,$

          the counter for the number of marks rejected by the nav filter shall be incremented by one:

          $N\_REJECT_I = N\_REJECT_I + 1,$

          and the sequential reject counter shall be incremented by one:

          $SEQ\_REJECT_I = SEQ\_REJECT_I + 1$

If SEQ_REJECT_I is found to exceed a predetermined number, REJ_ MAX, DISP_EDIT_I shall be set to ↓ to indicate that a number of sequential data marks have been edited.

$$\text{DISP\_EDIT}_I = \downarrow$$

(2) If $\text{SENSOR\_EDIT}_I \neq \text{ON}$, then the sequential reject counter shall be set to zero:

$$\text{SEQ\_REJECT}_I = 0,$$

the counter for the number of marks processed by the nav filter shall be incremented:

$$\text{N\_ACCEPT}_I = \text{N\_ACCEPT}_I + 1,$$

and the sequential accept counter shall be also incremented:

$$\text{SEQ\_ACCEPT}_I = \text{SEQ\_ACCEPT}_I + 1$$

If SEQ_ACCEPT_I exceeds a pre-determined number ACC_MIN, or SENSOR_EDIT_I has a value of FORCED, DISP_EDIT_I is given the value BLANK:

$$\text{DISP\_EDIT}_I = \text{BLANK}$$

If the indicator SENSOR_EDIT_I is found to have the value OFF on the initial check, both DISP_DELQ_I and DISP_SIG_I shall maintain the previous values from the prior filter subcycle.

B. Interface Requirements.- Input and output parameters are listed in table 4.2.9.

C. Processing Requirements.- This subfunction is called by NAV_ONORBIT_ RENDEZVOUS (section 4.2.1).

D. Constraints.- None

E. Supplementary Information.- A suggested implementation for this subfunction may be found in the detailed flow chart of Appendix B entitled MEAS_ PROCESSING_STATISTICS_REND.

TABLE 4.2.9.- MEAS_PROCESSING_STATISTICS_REND INPUT/OUTPUT

Variable Name	Input Source	Output Destination
ACC_MIN	**	
DEG_PER_RAD	**	
DISP_DELQ		*
DISP_EDIT	**	*
DISP_SIG		*
N_ACCEPT	*, SETUP, DISPLAY_COUNT_INIT	*
NAV_SIG	RRDOT_NAV, RR_ANGLE_ NAV, ANGLE_NAV, SETUP	
N_REJECT	*, SETUP, DISPLAY_COUNT_INIT	*
REJ_MAX	**	
SENSOR_DELQ	RRDOT_NAV, RR_ANGLE_NAV, ANGLE_NAV	
SENSOR_EDIT	REND_SENSOR_SELECT, RRDOT_NAV, RR_ANGLE_NAV, ANGLE_NAV	

*Onorbit/Rendezvous principal function, see section 4.2

**Initialization parameters, see section 4.7

TABLE 4.2.9.- MEAS_PROCESSING_STATISTICS_REND
INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
SEQ_ACCEPT	*, SETUP, DISPLAY_COUNT_INIT	*
SEQ_REJECT	*, SETUP, DISPLAY_COUNT_INIT	*

---

*Onorbit/Rendezvous principal function, see section 4.2

**Initialization parameters, see section 4.7

4.3 ONORBIT PRECISION STATE PREDICTION PRINCIPAL FUNCTION

A capability shall be provided for predicting the position and velocity of the Orbiter or target at some final time in the future or past, when an initial state and time are given.

The Onorbit Precision State Prediction principal function shall make no use of the IMU accumulated sensed velocities and therefore is a free-flight prediction process even though it may be performed during periods of flight in which navigation is using accumulated sensed velocities.

Since this principal function shall be used for different purposes having different environmental requirements in various navigation phases, the user shall, by setting the control flags to the appropriate values and by choosing the prediction method or integration step size, have the option to trade off the accuracy of the integration and the fidelity of the mathematical models in favor of the shorter execution time. This is accomplished with parameters specified prior to the invoking of the Onorbit Precision State Prediction principal function.

Table 4.3-1 is the principal function input and output list and shows data flow between the Onorbit Precision State Prediction principal function and other principal functions.

This principal function, which provides for Onorbit precision state prediction and rapid state prediction of the Orbiter or target position and velocity states, shall use either a fourth-order Runge-Kutta numerical integration technique, modified with Gill's coefficients, or a single-step two-body method (rapid state prediction). The S. Pines formulation of the equations of motion shall be used with each technique. Detailed requirements for the Runge-Kutta-Gill integration technique and the Pines formulation are provided in sections 4.3.1 and 4.3.2. Noncentral body accelerations shall be generated by the user-selected acceleration models (section 4.2.4.1.1) to account for perturbations due to drag, venting and uncoupled thrusting, and variations in the Earth's gravitational potential.

A. <u>Detailed Requirements</u>.- The Onorbit Precision State Prediction principal function computational scheme shall be performed as follows:

1. The desired gravity (GMDP and GMOP), drag (DMP), venting and uncoupled thrusting (VMP), and vehicle-attitude (ATMP) mode flags shall be obtained from the user, together with the prediction integration step size (PRED_STEP), initial state and time ($\underline{R}$ _PRED_INIT, $\underline{V}$ _PRED_INIT, and T_PRED_INIT), and final time at the end of the prediction interval (T_PRED_FINAL). If prediction is being performed for the Orbiter (i.e., ATMP = 1), the Orbiter mass to be used during prediction shall be user specified (PRED_ORB_MASS) along with an Orbiter reference area (PRED_ORB_AREA) and drag coefficient (PRED_ORB_CD).

The initial state vector shall then be renamed for use in the Pines equations-of-motion formulation and the seventh variable of integration (XN_7) initialized to zero:

$XN_{1 \text{ to } 3} = \underline{R}$ _PRED_INIT

$XN_{4 \text{ to } 6} = \underline{V}$ _PRED_INIT

$XN_7 = 0.$

In the above equations, the seventh variable of integration (XN_7, required by the Pines technique), is the integrated initial time T_PRED_INIT.

2. A check shall now be made on the gravity mode flag (GMDP) to determine if prediction is to be accomplished through the use of a simple two-body solution of a more precise integration technique.

GMDP = 0

a. If a two-body solution is required, (i.e., GMDP = 0) the prediction interval is computed as

i_CUR = T_PRED_FINAL - T_PRED_INIT

b. If a more precise integration technique is required (i.e., GMDP ≠ 0), several steps shall be performed to set up parameters required for the integration process.

(1) The input prediction step (PRED_STEP) is checked against a permission loaded maximum stepsize (DT_MAX) to verify its reasonability:

PRED_STEP > DT_MAX

If the input step is greater than the allowable maximum, then the value is reset to the maximum stepsize pre-mission loaded:

PRED_STEP = DT_MAX

Otherwise, the prediction stepsize input is left unchanged.

(2) Next, the total prediction time interval is calculated from the input initial and final times, and the current integrator time is set to zero:

TIME_DEL = T_PRED_FINAL - T_PRED_INIT

T_CUR = 0.

(3) If the total prediction time interval, TIME_DEL, is less than zero, a backward prediction has been requested and the internal integration step (DT_STEP) is set to a negative value of the input stepsize:

```
If:    ! TIME_DEL < 0.

Then:  DT_STEP = - PRED_STEP
```

Otherwise, TIME_DEL ≥ 0., the internal prediction step is set to the input prediction step:

DT_STEP = PRED_STEP

(4) The actual integration of the Orbiter or target state equations (formulated according to the Pines technique) shall now be performed by proceeding as follows for each step in the integration interval. Note that, in the Pines equations-of-motion formulation, it is the initial conditions (R_PRED_INIT, V_PRED_INIT, and T_PRED_INIT) that are integrated and then used in the closed-form solution of a two-body, unperturbed orbital problem using an F and G series.

The fourth-order Runge-Kutta-Gill integration technique shall be invoked in conjunction with the Pines equation-of-motion formulation for each predictor step (as discussed in section 4.3.1) until the prediction interval has been covered as follows (i.e. until T_CUR = TIME_DEL):

```
! DO UNTIL
! |T_CUR - TIME_DEL|
! ≤ PRED_TIME_TOL
```

- Check, on each step, to determine if the absolute value of the prediction step is greater than the absolute value of of the prediction interval remaining;

If, |DT_STEP| > |TIME_DEL - T_CUR|

then: DT_STEP = TIME_DEL - T_CUR

- The Runge-Kutta-Gill integrator shall then be invoked, with the input or adjusted value of DT_STEP;

CALL: RK_GILL

IN LIST: XN, DT_STEP, T_CUR, GMOP, GMDP, DMP,
VMP, ATMP, T_PRED_INIT

OUT LIST: XN, T_CUR

The output vector (XN), are the adjusted initial conditions to be used in the Pines equations of motion for a precision prediction conic solution.

3. After the calculations, as dictated by the testing of step 2, have been performed, the Pines equations of motion will be invoked to solve for the position and velocity vectors corresponding to T_PRED FINAL:

   CALL: PINES_METHOD

   IN LIST: XN, T_CUR, GMOP, GMDP, DMP, VMP, ATMP, T_PRED_INIT

   OUT LIST: DERIV, X

4. Upon being calculated (whether by a precise technique or a single step two-body solution), the final position and velocity are renamed for output:

   R _PRED_FINAL = X_1 to 3

   V _PRED_FINAL = X_4 to 6

B. Interface Requirements.- Input and output requirements are contained in table 4.3-2.

C. Processing Requirements.- This principal function requires user-supplied values of gravity (GMOP and GMDP), drag (DMP), venting and uncoupled thrusting (VMP), and vehicle-attitude (ATMP) mode flags, in conjunction with the initial state and time (R _PRED_INIT, V _PRED_INIT, T_PRED_INIT) and the final time (T_PRED_FINAL). Appropriate acceleration models may be found in section 4.2.4.1.1. When using this function for Orbiter or target vehicle state prediction, the venting and uncoupled thrusting flag (VMP) shall be set to zero. Additionally, if drag modeling is desired for Orbiter or target state prediction, the drag mode flag (DMP) should be set to one and the attitude mode flag (ATMP) set equal to one for the Orbiter or equal to two for the target. For prediction of the Orbiter's state, an Orbiter reference area (PRED_ORB_AREA), an Orbiter drag coefficient (PRED_ORB_CD), and an Orbiter mass (PRED_ORB_MASS) are also to be supplied. The Onorbit Precision State Prediction principal function is called by the following modules in the Onorbit/Rendezvous Navigation Sequencer principal function:

OPS_2_OR_8_INITIALIZE
STATE_VECTOR_PREDICT_TASK

and by the following module in the Onorbit/Rendezvous Navigation principal function:

STATE_VECTOR_PREDICT_TASK

In addition it may be called by other users outside of navigation.

D. Constraints.- Vent thrust is not to be modeled in prediction. Hence, the vent thrust flag (VMP) is to be set to zero. Atmospheric drag is to be modeled with constant coefficients. Hence, whenever drag is to be modeled in a prediction, the vehicle attitude flag (ATMP) is not to be set to zero (see section 4.2.4.1.1).

Since the same compool locations are used by all users of this principal function for setup and output, it is required that these parameters be protected from alteration by other users during execution of this principal function. Variables to be protected are listed as follows: GMOP, GMDP, DMP, VMP, ATMP, PRED_STEP, R PRED_INIT, V _PRED_INIT, T_PRED_INIT, R _PRED FINAL, V _PRED_FINAL, T_PRED_FINAL, PRED_ORB_AREA, PRED_ORB_CD, PRED_ORB_MASS.

E. Supplementary Information.- The Onorbit Precision State Prediction principal function shall be used for both precision and rapid state prediction. Rapid state prediction consists of a less accurate, single-step two-body F and G series solution involving no numerical integration. Table 4.3-3 lists several examples of input variable list combinations for the various types of prediction performed. A suggested implementation of this principal function may be found in Appendix C under the following:

ONORBIT_PREDICT

TABLE 4.3-1.- ONORBIT PRECISION STATE PREDICTION PRINCIPAL FUNCTION INPUT/OUTPUT

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
ATMP	Onorbit/Rendezvous Nav, Onorbit Guidance, Onorbit/Rendezvous Nav Seq, ORB MNVR DIP	PINES_METHOD,RK_GILL	TLM	NONE
DMP	Onorbit/Rendezvous Nav, Onorbit Guidance, Onorbit/Rendezvous Nav Seq, ORB MNVR DIP	PINES_METHOD,RK_GILL		
GMDP	Onorbit/Rendezvous Nav, Onorbit Guidance, Onorbit/Rendezvous Nav Seq, ORB MNVR DIP	PINES_METHOD,RK_GILL		
GMOP	Onorbit/Rendezvous Nav, Onorbit Guidance, Onorbit/Rendezvous Nav Seq, ORB MNVR DIP	PINES_METHOD,RK_GILL		
PRED_ORB_AREA	Onorbit/Rendezvous Nav, Onorbit Guidance, Onorbit/Rendezvous Nav Seq, ORB MNVR DIP	ACCEL_ONORBIT		
PRED_ORB_CD	Onorbit/Rendezvous Nav, Onorbit Guidance, Onorbit/Rendezvous Nav Seq, ORB MNVR DIP	ACCEL_ONORBIT		

TABLE 4.3-1.- ONORBIT PRECISION STATE PREDICTION PRINCIPAL FUNCTION INPUT/OUTPUT.- Continued

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
PRED_ORB_ MASS	Onorbit/Rendezvous Nav, Onorbit Guidance, Onorbit/ Rendezvous Nav Seq, ORB MNVR DIP	ACCEL_ ONORBIT		
PRED_STEP	Onorbit/Rendezvous Nav, Onorbit Guidance, Onorbit/ Rendezvous Nav Seq, ORB MNVR DIP	ONORBIT_ PREDICT	TLM	NONE
R _PRED_ FINAL			Onorbit/ Rendezvous Nav, Onorbit Guidance, Onorbit/ Rendezvous Nav Seq, TLM, ORB MNVR DIP	ONORBIT_ PREDICT
R _PRED_INIT	Onorbit/Rendezvous Nav, Onorbit Guidance, Onorbit/ Rendezvous Nav Seq, ORB MNVR DIP	ONORBIT_ PREDICT	TLM	NONE
SQR_EMU	Onorbit/Rendezvous Nav Seq	PINES_ METHOD		
T_PRED_FINAL	Onorbit/Rendezvous Nav, Onorbit Guidance, Onorbit/ Rendezvous Nav Seq, ORB MNVR DIP	ONORBIT_ PREDICT	TLM	NONE

TABLE 4.3-1.- ONORBIT PRECISION STATE PREDICTION PRINCIPAL FUNCTION INPUT/OUTPUT.- Concluded

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
T_PRED_INIT	Onorbit/Rendezvous Nav, Onorbit Guidance, Onorbit/Rendezvous Nav Seq, ORB MNVR DIP	PINES_METHOD, RK_GILL	TLM	NONE
VMP	Onorbit/Rendezvous Nav, Onorbit Guidance, Onorbit/Rendezvous Nav Seq, ORB MNVR DIP	PINES_METHOD, RK_GILL		
V _PRED_FINAL			Onorbit/Rendezvous Nav, Onorbit Guidance, Onorbit/Rendezvous Nav Seq, TLM, ORB MNVR DIP	ONORBIT_PREDICT
V _PRED_INIT	Onorbit/Rendezvous Nav, Onorbit Guidance, Onorbit/Rendezvous Nav Seq, ORB MNVR DIP	ONORBIT_PREDICT	TLM	NONE

TABLE 4.3-2.- ONORBIT_PREDICT INPUT/OUTPUT

Variable Name	Input Source	Output Destination
ATMP	*	PINES_METHOD,RK_GILL
DERIV	PINES_METHOD	
DMP	*	PINES_METHOD,RK_GILL
DT_MAX	**	
DT_STEP		RK_GILL
GMDP	*	PINES_METHOD,RK_GILL
GMOP	*	PINES_METHOD,RK_GILL
PRED_STEP	*	
PRED_TIME_TOL	**	
R _PRED_FINAL		*
R _PRED_INIT	*	
T_CUR	RK_GILL	PINES_METHOD,RK_GILL
T_PRED_FINAL	*	
T_PRED_INIT	*	PINES_METHOD,RK_GILL
VMP	*	PINES_METHOD,RK_GILL
V _PRED_FINAL		*
V _PRED_INIT	*	
X	PINES_METHOD	
XN	RK_GILL	PINES_METHOD,RK_GILL

---

*P.F. I/O for onorbit precision state prediction principal function, see section 4.3

**Initialization parameters, see section 4.7

TABLE 4.3-3.- PREDICTOR SUGGESTED FLAG SETTINGS

Vehicle	Prediction type	GMDP*	GMOP*	DMP	VMP	ATMP	PRED_STEP	Comments
Orbiter	Precision	4	4	1	0	1	User selects	Full fourth degree potential model. Drag with constant drag coefficient, area.
Orbiter	Rapid precision	2	0	1	0	1	User selects	J_2 only potential model with constant drag coefficient, area.
Orbiter	Rapid two-body	0	0	0	0	0	0	Single-step two-body F and G series solution.
Target	Precision	4	4	1	0	2	User selects	Full fourth degree potential model drag with constant area, drag coefficient.
Target	Rapid precision	2	0	1	0	2	User selects	J_2 only potential model with constant drag coefficient, area.
Target	Rapid two-body	0	0	0	0	0	0	Single-step two-body F and G series solution.

*When prediction is being performed for both vehicles (Orbiter and target) over a similar trajectory, the same degree and order potential model should be used for each prediction so that potential model errors will be avoided.

4.3.1 Integration of the Equations of Motion (RK_GILL)

A fourth-order Runge-Kutta-Gill integration technique is used for prediction of the Orbiter and target state vectors. The technique is actually a fourth-order Runge-Kutta numerical integration technique, modified with Gill's coefficients, used in conjunction with S. Pines' formulation for the equations of motion (see section 4.3.2).

A. Detailed Requirements.- The Runge-Kutta-Gill (RK_GILL) integration subfunction will be activated each time a call statement of the following form is encountered:

CALL: RK_GILL

IN LIST: $\underline{XN}$,DT_STEP, T_CUR,GMO,GMD,DM,VM,ATM,T_IN

OUT LIST: $\underline{XN}$,T_CUR

where:

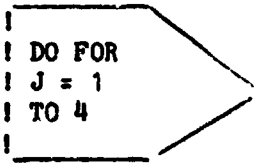
$\underline{XN}$ = an array containing the seven variables of integration (i.e., integrated initial conditions)
DT_STEP = the integration step size
T_CUR = the RK_GILL step size subinterval time (i.e., there are four cycles of RK_GILL per integration step size)
GMO = the Earth gravitational potential model order
GMD = the Earth gravitational potential model degree
DM = the drag acceleration model flag
VM = the vent and uncoupled thrust acceleration model flag
ATM = the vehicle attitude mode flag
T_IN = the initial state time

The computations initiated by call to the Runge-Kutta-Gill subfunction will be the following and in the order indicated.

1. The initial time of the current integration step, T_CUR, shall be saved in T_STOR.

T_STOR = T_CUR

2. Next a counter, J, shall be tested to ensure that four evaluations of Runge-Kutta-Gill are determined.



a. If the evaluation cycle is 4 or less, a new value of T_CUR (the integration step time) shall be determined as:

$$T\_CUR = T\_STOR + AA_1 \, BB_J \quad DT\_STEP$$

b. Next the Pines method will be called to calculate derivatives (DERIV), of the initial conditions.

CALL: PINES_METHOD

IN LIST: XN,T_CUR,GMO,GMD,DM,VM,ATM,T_IN

OUT LIST: DERIV,X

The call arguments are as previously described, and details of the Pines method are given in section 4.3.2.

c. The Runge-Kutta-Gill integration continues with the numerical integration of derivatives of the initial conditions (XN_L) in the following manner:

```
DO FOR L = 1,7
    $P = DT\_STEP \; DERIV_L$
    $XN_L = XN_L + AA_J (P - BB_J Q_L)$
    $Q_L = CC_J P + DD_J Q_L$
```

where:

AA, BB, CC, DD = premission-loaded arrays (J = 1 to 4) containing coefficients required for this formulation of the Runge-Kutta-Gill integration technique

XN = an array containing the seven variables of integration (i.e., integrated initial conditions)

DERIV = an array containing the total derivatives of the initial conditions at the current time

P = integration variable used in Runge-Kutta-Gill technique

Q_L = integration variable used in the Runge-Kutta-Gill technique

After the seven variables of integration have been obtained, the RK GILL will return to repeat step 2, and cycle through until J is greater than 4.

B. Interface Requirements.- Input and output parameters for the Runge-Kutta-Gill integration subfunction are given in table 4.3.1.

C. Processing Requirements.- The Runge-Kutta-Gill subfunction is called by

Onorbit precision state prediction (ONORBIT_PREDICT)

D. Constraints.- Because the Runge-Kutta-Gill subfunction is used by precision prediction subfunctions which may be executed by multiple users at the same time, it should be protected against interruption.

E. Supplementary Information.- A suggested implementation of this subfunction in the form of detailed flow diagrams may be found in Appendix C under the following:

RK_GILL

TABLE 4.3.1.- RK_GILL INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
ATM	ATMP	ONORBIT_PREDICT	
DM	DMP	ONORBIT_PREDICT	
DT_STEP	DT_STEP	ONORBIT_PREDICT	
GMD	GMDP	ONORBIT_PREDICT	
GMO	GMOP	ONORBIT_PREDICT	
T_CUR	T_CUR	ONORBIT_PREDICT	
T_IN	T_PRED_INIT	ONORBIT_PREDICT	
VM	VMP	ONORBIT_PREDICT	
XN	XN	ONORBIT_PREDICT	
T_CUR	T_CUR		ONORBIT_PREDICT
XN	XN		ONORBIT_PREDICT

TABLE 4.3.1.- RK_GILL INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
AA	**	
ATM		PINES_METHOD
BB	**	
CC	**	
DD	**	
DM		PINES_METHOD
DERIV	PINES_METHOD	
GMD		PINES_METHOD
GMO		PINES_METHOD
T_CUR		PINES_METHOD
T_IN		PINES_METHOD
VM		PINES_METHOD
X	PINES_METHOD	
XN		PINES_METHOD

**Initialization parameters, see section 4.7

4.3.2 Equations of motion (PINES_METHOD)

For predicting the Orbiter and target state vectors, the equations of motion are to take the form of a variation-of-parameters method devised by S. Pines, where parameters to be varied are the Cartesian initial conditions of the motion. The integration scheme to be used in connection with these equations is the Gill modification of the Runge-Kutta technique (see section 4.3.1).

A. Detailed Requirements.- The Pines equations of motion subfunction will be invoked whenever a call statement of the following form is encountered:

CALL: PINES_METHOD

IN LIST: $\underline{XN}$,T_CUR,GMO,GMD,DM,VM,ATM,T_IN

OUT LIST: $\underline{DERIV}$,$\underline{X}$

where:

$\underline{XN}$ = the seven variables of integration

T_CUR = the initial integration time of the current step

GMO = Earth's gravitational potential model order

GMD = Earth's gravitational potential model degree

DM = drag model acceleration computation flag

VM = vent and thrusting acceleration model flag

ATM = attitude mode flag

T_IN = initial time

and:

$\underline{DERIV}$ = the output total derivatives of integration

$\underline{X}$ = output two body position and velocity vectors of conic

The PINES_METHOD subfunction will cause the following calculations to be made in the order given:

1. Several terms used in the F and G series calculations for the closed-form two-body equations are computed.

$R\_IN = | XN_{1 \text{ to } 3} |$

$R\_IN\_INV = 1./R\_IN$ F3

$SMA = 1./(2.\ R\_IN\_INV - (XN_{4 \text{ to } 6} \cdot XN_{4 \text{ to } 6})/EARTH\_MU)$ F3

$C1 = SQRT\ (SMA)/SQR\_EMU$ F3 F4

$DELTAT = T\_CUR - XN_7$

$D\_IN = XN_{1 \text{ to } 3} \cdot XN_{4 \text{ to } 6}$

$R\_FIN\_TEMP\_INV = 0.$

2. The conic solution subfunction (F_AND_G) shall then be invoked to calculate several terms used in computation of the conic velocity vector ($X_{4 \text{ to } 6}$) and initial condition derivatives and compute the two-body conic position vector ($X_{1 \text{ to } 3}$) as follows (see section 4.2.5.2.1).

   CALL: F_AND_G

   IN LIST: SMA, DELTAT, C1, $XN_{1 \text{ to } 3}$, R_IN_INV, R_FIN_TEMP_INV, $XN_{4 \text{ to } 6}$, D_IN, D_FIN_TEMP

   OUT LIST: F, G, FDOT, GDOT, S0, S1, S2, S3, $X_{1 \text{ to } 3}$, R_FIN_INV, THETA

3. The two-body velocity vector shall then be computed:

   $X_{4 \text{ to } 6} = FDOT\ XN_{1 \text{ to } 3} + GDOT\ XN_{4 \text{ to } 6}$

4. Perturbation accelerations shall now be calculated and several computations shall then be performed to compute perturbation derivatives for F and G series terms used in calculating total derivatives of the seven variables of integration.

   $T\_ACCEL = T\_IN + T\_CUR$

   $\underline{P} = \underline{ACCEL\_ONORBIT}\ (GMD, GMO, DM, VM, ATM, X_{1 \text{ to } 3}, X_{4 \text{ to } 6}, T\_ACCEL)$

   $\underline{P} = \underline{P} - \underline{G}\ \_CENTRAL$

   $D\_TAU = X_{1 \text{ to } 3} \cdot \underline{P}$

   $D\_AUX = X_{4 \text{ to } 6} \cdot \underline{P}$

   $C2 = C1^2$

F3 This equation shall be protected against division by zero (Reference 3.6-3).
F4 This equation shall be protected against square roots of a negative number (Reference 3.6-4).

S1 = C1 S1

S2 = C2 S2

C3 = 1./C2 F3

S3 = SMA S2

S4 = 2. S3 D_AUX

C4 = C2 D_AUX

C5 = C4 S1

S5 = S2 D_TAU

DD = S1 C3 R_IN (SMA R_IN_INV-1.) + S0 D_IN

S6 = 2. S2 C4 DD + S5

R_IN_TAU = S4 - C2 S1 D_AUX DD - S1 D_TAU

R_IN_AUX = R_IN_INV R_IN_TAU

F_TAU = (S3 C3 R_IN_AUX - S4) R_IN_INV

G_TAU = C5/R_FIN_INV - S6 F3

FD_TAU = FDOT (C4 - R_IN_AUX)

GD_TAU = - S4 R_FIN_INV

5. Finally, the total derivatives of the variables of integration are to be computed as follows:

$\text{DERIV}_{1\ \text{to}\ 3} = \text{GD\_TAU}\ X_{1\ \text{to}\ 3} - \text{G\_TAU}\ X_{4\ \text{to}6} - \text{G}\ \underline{\text{P}}$

$\text{DERIV}_{4\ \text{to}\ 6} = - \text{FD\_TAU}\ X_{1\ \text{to}\ 3} + \text{F\_TAU}\ X_{4\ \text{to}\ 6} + \text{F}\ \underline{\text{P}}$

$\text{DERIV}_{7} = \text{S6} - 3.\ \text{C4 SMA (C1 THETA-S1)} - \text{C5/R\_FIN\_INV}$ F3

B. Interface Requirements.- Input and output for the Pines method subfunction are given in table 4.3.2.

C. Processing Requirements.- The following is a list of subfunctions and principal functions that call the Pines method.

RK_GILL
ONORBIT_PREDICT

F3 This equation shall be protected against division by zero (Reference 3.6-3).

D. Constraints.- The ACCEL_ONORBIT function shall not be invoked by any other software module during the computation of the perturbation acceleration vector, P, to maintain the correct value of the central force term, G_CENTRAL.

E. Supplementary Information.- A suggested implementation of the Pines method of the equations of motion in the form of detailed flowcharts may be found in Appendix C under the following:

PINES_METHOD

TABLE 4.3.2.- PINES_METHOD INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
ATM	ATM	RK_GILL	
DM	DM	RK_GILL	
GMD	GMD	RK_GILL	
GMO	GMO	RK_GILL	
T_CUR	T_CUR	RK_GILL	
T_IN	T_IN	RK_GILL	
VM	VM	RK_GILL	
XN	XN	RK_GILL	
ATM	ATMP	ONORBIT_PREDICT	
DM	DMP	ONORBIT_PREDICT	
GMD	GMDP	ONORBIT_PREDICT	
GMO	GMOP	ONORBIT_PREDICT	
T_CUR	T_CUR	ONORBIT_PREDICT	
T_IN	T_PRED_INIT	ONORBIT_PREDICT	
VM	VMP	ONORBIT_PREDICT	
XN	XN	ONORBIT_PREDICT	
DERIV	DERIV		RK_GILL
X	X		RK_GILL
DERIV	DERIV		ONORBIT_PREDICT
X	X		ONORBIT_PREDICT

TABLE 4.3.2.- PINES_METHOD INPUT/OUTPUT.- Continued

Variable Name	Input Source	Output Destination
ATM †	ACCEL_ONORBIT	ACCEL_ONORBIT
C1		F_AND_G
DELTAT		F_AND_G
D_FIN_TEMP	**	F_AND_G
D_IN		F_AND_G
DM		ACCEL_ONORBIT
EARTH_MU	**	
F	F_AND_G	
FDOT	F_AND_G	
G	F_AND_G	
G _CENTRAL	ACCEL_ONORBIT	
GDOT	F_AND_G	
GMD		ACCEL_ONORBIT
GMO		ACCEL_ONORBIT
R_FIN_INV	F_AND_G	
R_FIN_TEMP_INV		F_AND_G
R_IN_INV		F_AND_G
SO	F_AND_G	
S1	F_AND_G	

**Initializations parameters, see section 4.7

†Only the value of ACCEL_ONORBIT is used

TABLE 4.3.2.- PINES_METHOD INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
S2	F_AND_G	
S3	F_AND_G	
SMA		F_AND_G
SQR_EMU	*	
T_ACCEL		ACCEL_ONORBIT
THETA	F_AND_G	
VM		ACCEL_ONORBIT
X_1 to 3	F_AND_G	ACCEL_ONORBIT
X_4 to 6		ACCEL_ONORBIT
XN		F_AND_G

*Precision State Prediction principal function, see section 4.3

4.4 ONORBIT/RENDEZVOUS USER PARAMETER PROCESSING SEQUENCER PRINCIPAL FUNCTION (ONORBIT_REND_UPP_SEQ)

This principal function will provide a capability for initialization and control of the principal functions and subfunctions associated with the computations of user parameters during the onorbit/rendezvous operational sequence. This sequencer will provide initialization and control of the onorbit user parameter state propagation subfunction and those user parameter processing principal functions used for this operational sequence.

Events to be used as cues by the sequencer for performing the required initialization and sequencing are defined in the Level B GN&C CPDS. The particular events and a summary of the associated user parameter actions pertaining to this user parameter sequencer are given in table 4.4-1.

A. Detailed Requirements.- The Onorbit/Rendezvous User Parameter Processing Sequencer will be initiated upon the occurrence of any of the following events:

- Major mode transition from 106 to 201
- Major mode transition from 301 to 201
- Transition from OPS 2 to OPS 8
- Transition from OPS 00 to Major Mode 201
- Transition from OPS 8 to Major Mode 201

This sequencer shall be terminated upon the transition from OPS 2 or OPS 8 to OPS 3 or OPS 00.

The following paragraphs specify the detailed requirements that are summarized in table 4.4-1. These requirements specify, for each of the event cues to be utilized by the sequencer, the actions that the sequencer is to initiate.

- EVENT_60A; transition from OPS 8 to OPS 2
- EVENT_60B; transition from OPS 2 to OPS 8

a. If any of the above event flags has been set to "ON" by Moding, Sequencing, and Control (MSC), or crew input and detected "ON" by this sequencer, a check shall be made on a flag set by the Onorbit/ Rendezvous Navigation Sequencer:

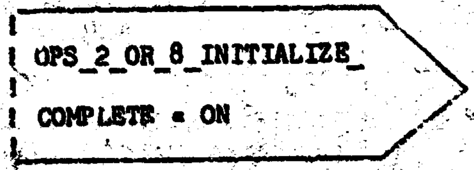


TABLE 4.4-1.- ONORBIT/RENDEZVOUS USER PARAMETER PROCESSING SEQUENCER EVENTS

Event No.	Event Name/Description	Action Taken by Sequencer in Response to Event
60	Transition from OPS 1 to OPS 2	Initiate cyclic execution of onorbit user parameter state propagation and onorbit user parameter calculations at a repetition rate of 0.52 Hertz
E1	Transition from 301 to 201	Same as Event 60 action
84	Transition from OPS 00 to 201	Same as Event 60 action
60A	Transition from OPS 8 to 201	Same as Event 60 action
73	Transition from 202 to 201	Cancel and reschedule the onorbit user parameter state propagation to change the cyclic repetition rate of onorbit user parameter state propagation to repetition rate of 0.52 Hertz.
69	Initiate guidance	Cancel onorbit user parameter state propagation. Reschedule cyclic processing of onorbit user parameter state propagation at a repetition rate of 1.04 Hertz
60B	Transition from 201 to OPS 8	Same as Event 60 action

This signal (=ON) indicates that the necessary initialization of certain state parameters has been accomplished within the Onorbit/ Rendezvous Navigation Sequencer (section 4.1) and that the onorbit/ rendezvous user parameter state propagation and onorbit user parameter calculations subfunctions shall commence at a repetition rate of 0.52 Hertz.

2. Next, a check shall be made to detect a transition from Major Mode 202 to 201; EVENT_73. Based upon this cue, execution of the onorbit user parameter state propagator shall be cancelled and rescheduled at a rate of 0.52 Hertz.

3. A check shall now be made on the EVENT_69 cue (initiate guidance). Based upon this cue, the current scheduling of the user parameter state propagator is to be cancelled. Cyclic execution of this task is to be rescheduled at a repetition rate of 1.04 Hertz beginning with this event. The purpose of cancelling and rescheduling the onorbit user parameter state propagator upon the initiate guidance signal is to not only change the execution rate, but to also get the execution of this module in synchronization with the execution of onorbit guidance, which is to be initiated at this time. This cancelling and rescheduling is to be done "y" seconds prior to OMS ignition, such that a subsequent user state update will occur, as nearly as possible, at the time of ignition.

B. Interface Requirements.- Input and output requirements for this principal function are presented in table 4.4-2.

C. Processing Requirements.- None

D. Constraints.- None

E. Supplemental Information.- A suggested implementation of the onorbit/ rendezvous UPP sequencer in the form of detailed flow charts is shown in Appendix D under ONORBIT_REND_UPP_SEQ.

TABLE 4.4-2.- ONORBIT/RENDEZVOUS USER PARAMETER PROCESSING SEQUENCER PRINCIPAL FUNCTION INPUT/OUTPUT

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
EVENT_60	MSC	ONORBIT_ REND_UPP_ SEQ	TLM	
EVENT_60A	MSC	ONORBIT_ REND_UPP_ SEQ	TLM	
EVENT_E1	MSC	ONORBIT_ REND_UPP_ SEQ	TLM	
EVENT_60B	MSC	ONORBIT_ REND_UPP_ SEQ	TLM	
EVENT_69	MSC	ONORBIT_ REND_UPP_ SEQ	TLM	
EVENT_73	MSC	ONORBIT_ REND_UPP_ SEQ	TLM	
EVENT_84	MSC	ONORBIT_ REND_UPP_ SEQ	TLM	
OPS_2_OR_8_ INITIALIZE_ COMPLETE	Onorbit/Rendezvous Nav Seq	ONORBIT_ REND_UPP_ SEQ		

4.5 USER PARAMETER PROCESSING PRINCIPAL FUNCTION

This principal function shall serve as the interface between navigation and users of navigation-related data during the onorbit operational sequence. This function shall maintain the vehicle state within the user parameter state propagation subfunction and shall:

- Provide this state to users who require vehicle state parameters in M50 coordinates (see User Parameter State Propagation, section 4.5.1 and Onorbit User Parameter Calculations, section 4.5.2).
- Provide the software to transform this state for users who require nav state-related parameters (see User Parameter Calculations, section 4.5.2).

Interface parameters between this principal function and other GN&C principal functions are presented in table 4.5.

TABLE 4.5.- USER PARAMETER PROCESSING PRINCIPAL FUNCTION INPUT/OUTPUT

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
A _SENSED			TLM	ONORBIT_REND_USER_PARAM_STATE_PROP
DEL_R_TARG			Universal Pointing, TLM	ONORBIT_USER_PARAMETER_CALCULATIONS
DEL_V_TARG			Universal Pointing, TLM	ONORBIT_USER_PARAMETER_CALCULATIONS
DOING_REND_NAV	ONORBIT_REND_NAV_SEQUENCER	ONORBIT_REND_USER_PARAM_STATE_PROP, ONORBIT_USER_PARAMETER_CALCULATIONS		
FILT_UPDATE	Onorbit/Rendezvous Nav, Onorbit/Rendezvous Nav Seq	ONORBIT_REND_USER_PARAM_STATE_PROP		
IMU_NAV_ACCEL_THRESH	IMU_ALIGN_DISPLAY	ONORBIT_REND_USER_PARAM_STATE_PROP		
PWRD_FLT_NAV	REL NAV DISPLAY, FCS/DED DISP c/o DISPLAY, Onorbit Rendezvous Navigation Sequencer	ONORBIT_REND_USER_PARAM_STATE_PROP		

TABLE 4.5.- USER PARAMETER PROCESSING PRINCIPAL FUNCTION INPUT/OUTPUT.- Continued

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
$\underline{R}$ _AVGG			Onorbit Guidance, Startracker SOP, REL NAV SPEC FUNC, Rendezous Targeting, ATT PROC, Orbit Maneuver DIP, TLM	ONORBIT_REND_USER_PARAM_STATE_PROP
$\underline{R}$ _EF			GN&C/SM-PL	ONORBIT_USER_PARAMETER_CALCULATIONS
$\underline{R}$ _RESET	Onorbit/Rendezvous Nav, Onorbit/Rendezvous Nav Seq	ONORBIT_REND_USER_PARAM_STATE_PROP		
$\underline{R}$ _TARGET			Star Tracker SOP, Rend Target, REL NAV SPEC FUNC, TLM	ONORBIT_REND_USER_PARAM_STATE_PROP
$\underline{R}$ _TV_RESET	Onorbit/Rendezvous Nav, Onorbit/Rendezvous Nav Seq	ONORBIT_REND_USER_PARAM_STATE_PROP		
T_IMU			TLM	ONORBIT_REND_USER_PARAM_STATE_PROP
T_IMUS_GA	IMU INT PROC	IMU DATA SNAP		

TABLE 4.5.- USER PARAMETER PROCESSING PRINCIPAL FUNCTION INPUT/OUTPUT.- Continued

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
T_RESET	Onorbit/Rendezvous Nav, Onorbit/ Rendezvous Nav Seq	ONORBIT_REND_USER_PARAM_STATE_PROP		
T_SEC_CMT			GN&C/SM-PL	ONORBIT_USER_PARAMETER_CALCULATIONS
T_STATE			Rendezvous Targeting, Onorbit Guidance, Orbit Maneuver DIP, TLM	ONORBIT_PARAM STATE_PROP
UPP_USE_IMU_DATA			TLM	ONORBIT_REND_USER_PARAM_STATE_PROP
V _AVGG			Attitude Proc, Orbit Maneuver DIP,TLM,Star Tracker SOP,REL NAV SPEC FUNC, Rendezvous Targeting, Onorbit Guidance	ONORBIT_REND USER_PARAM_STATE_PROP
V _IMU_CURRENT	IMU_RM	IMU DATA SNAP		
V _IMU_OLD			Onorbit Guidance	ONORBIT_REND_USER_PARAM_STATE_PROP

TABLE 4.5.- USER PARAMETER PROCESSING PRINCIPAL FUNCTION INPUT/OUTPUT.- Concluded

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
V _IMU_RESET	Onorbit/Rendezvous Nav, Onorbit/Rendezvous Nav Seq	ONORBIT_REND_USER_PARAM_STATE_PROP		
V _RESET	Onorbit/Rendezvous Nav, Onorbit/Rendezvous Nav Seq	ONORBIT_REND_USER_PARAM_STATE_PROP		
V _RHO_EF			GN&C/SM-PL	ONORBIT_USER_PARAMETER_CALCULATIONS
V _TARGET			Rendezvous Targeting, REL NAV SPEC FUNC, TLM	ONORBIT_REND_USER_PARAM_STATE_PROP
V _TV_RESET	Onorbit/Rendezvous Nav, Onorbit/Rendezvous Nav Seq	ONORBIT_REND_USER_PARAM_STATE_PROP		
V _IMU_SNAP			TLM	ONORBIT_REND_USER_PARAM_STATE_PROP

4.5.1 User Parameter State Propagation (ONORBIT_REND_USER_PARAM_STATE_PROP)

The onorbit and rendezvous navigation state propagation subfunctions advance the navigation state vector at relatively large intervals, at the end of which external measurement data processed by the filter are incorporated when appropriate. Users, such as guidance and displays, require a knowledge of the state vector at shorter intervals. The onorbit and rendezvous user parameter state propagation subfunction will satisfy the requirements of such users by integrating the equations of motion within the intervals of the navigation propagation with use of a simplified computation of the gravitational acceleration in conjunction with a small step size.

The integration of the Orbiter and target state vectors is to be performed by an average-g process, using a modeled acceleration that contains a simplified Earth gravitational force model. In the case of the Orbiter, if the acceleration derived in the UPP state propagator from the IMU sensed velocities is above a certain threshold level, this acceleration is to be used in the integration process. If the sensed acceleration falls below the threshold level, only the modeled acceleration is to be utilized in the integration.

In the rendezvous phases it is also necessary to propagate the target vehicle state. There being no IMU's in this vehicle, only the modeled acceleration is to be used in the integration.

This process will be restarted after each filter update with the filter states. The values of the filter updated position and velocity vectors, together with their time tag and the total accumulated IMU velocity, are stored (at each navigation cycle) in special locations for use by the user parameter state propagation subfunction. This prevents the errors resulting from use of a less accurate integrating scheme from becoming too large and, at the same time, provides a synchronization between the propagation tasks.

A. Detailed Requirements.- A capability shall be provided for a fast computation of the position and velocity of the Orbiter during all phases of OPS 2 and OPS 8, and of the position and velocity of the target vehicle during all rendezvous phases OPS 2. This computation shall provide the required state vectors in an M50 coordinate system by the integration of the equations of motion that include gravity accelerations and, for the Orbiter, the IMU sensed velocities, if they give a significant contribution.

The following steps shall be performed (in the order indicated):

1. Snap the IMU accumulated sensed velocity and time tag:

SNAP($\underline{V}$_IMU_SNAP ,T_IMU)

(see section 4.2.2.1)

2. In the case of the Orbiter, the value of the state that is to be advanced (integrated forward in time) may be from one of two sources (the one used depends on the tested value of the flag (FILT_UPDATE), which indicates the availability of a filter updated state):

   a. If an update from the filter is available (FILT_UPDATE = ON), the navigation filter updated state, together with its time tag and associated IMU accumulated sensed velocity, is to replace the previous propagated state, time tag, and accumulated velocity. The filter updated values are $\underline{R}$_RESET, $\underline{V}$_RESET, T_RESET and $\underline{V}$_IMU_RESET; the vectors maintained by the user parameter state propagator are $\underline{R}$_AVGG and $\underline{V}$_AVGG. The time tag is T_STATE. Thus, if FILT_UPDATE = ON, the following will be done:

      $\underline{R}$_AVGG = $\underline{R}$_RESET

      $\underline{V}$_AVGG = $\underline{V}$_RESET

      $\underline{V}$_IMU_OLD = $\underline{V}$_IMU_RESET

      T_STATE = T_RESET

   b. If an update from the filter is not available (FILT_UPDATE = OFF), the propagated state, saved from the previous cycle, is to be advanced. The value of the IMU-accumulated sensed velocity saved from the previous cycle is available for state advancement purposes.

3. Compute the interval over which advancement is required:

   DT_IMU = T_IMU - T_STATE

[illegible]

interval (DT_IMU).

DA_THRESHOLD_IMU = IMU_NAV_ACCEL_THRESH

(GO 10^{-6}) / DT_IMU F3

F3 This equation shall be protected against division by zero (Reference 3.6-3).

(3) Test the magnitude of the sensed acceleration ($|\underline{A}$ _SENSED$|$) for this UPP cycle against the computed IMU threshold.

$|\underline{A}$ _ SENSED$|$ > DA_THRESHOLD_IMU

If $|\underline{A}$ _SENSED$|$ > DA_THRESHOLD_IMU, set a flag to ON for telemetry

UPP_USE_IMU_DATA = ON

Otherwise, turn the telemetry flag OFF, and set the acceleration value to zero.

UPP_USE_IMU_DATA = OFF

$\underline{A}$ _ SENSED = $\underline{0}$.

b. If the PWRD_FLT_NAV flag is OFF, turn the telemetry flag to OFF, and set the acceleration value to zero.

UPP_USE_IMU_DATA = OFF

$\underline{A}$ _SENSED = $\underline{0}$.

5. The position and velocity vectors of the Orbiter shall then be obtained by a call to the user parameter state integrator (see section 4.5.1.1).

CALL: AVERAGE_G_INTEGRATOR

IN LIST: $\underline{R}$ _AVGG, $\underline{V}$ _AVGG, DT_IMU, $\underline{A}$ _SENSED, T_STATE, T_IMU

OUT LIST: $\underline{R}$ _AVGG, $\underline{V}$ _AVGG

The calculations performed up to this point refer to the Orbiter's state. Propagation of the target state is required only during the rendezvous phases. A flag (DOING_REND_NAV), which has the value ON only during these phases, shall then be consulted by the user parameters state propagator.

6. Test the DOING_REND_NAV flag. If it is found to be ON, perform the following steps:

a. Test the FILT_UPDATE flag to determine if a target state is available from NAV.

If FILT_UPDATE = ON, set

$\underline{R}$ _TARGET = $\underline{R}$ _TV_RESET

$\underline{V}$ _TARGET = $\underline{V}$ _TV_RESET

where $\underline{R}$ _TARGET and $\underline{V}$ _TARGET represent the position and velocity vectors of the target vehicle advanced by the user parameter state propagator, and $\underline{R}$ _TV_RESET and $\underline{V}$ _TV_RESET represent the target state vectors from the navigation filter.

b. Advance the target state by a call to the integrator. In this call, the vector that contains the sensed acceleration shall be set to zero.

$\underline{A}$ _SENSED = 0

CALL: AVERAGE_G_INTEGRATOR

IN LIST: $\underline{R}$ _TARGET, $\underline{V}$ _TARGET, DT_IMU, $\underline{A}$ _SENSED, T_STATE, T_IMU

OUT LIST: $\underline{R}$ _TARGET, $\underline{V}$ _TARGET
(see section 4.5.1.1)

After the state vector updates have been completed, the following steps are to be executed:

7. Save the time tag output for use in the next cycle:

T_STATE = T_IMU

8. Save the latest IMU accumulated sensed velocity:

$\underline{V}$ _IMU_OLD = $\underline{V}$ _IMU_SNAP

9. Set the FILT_UPDATE flag to OFF.

This completes the sequence of calculations of a user parameter state propagation cycle.

B. Interface Requirements.- The input and output variables are listed in table 4.5.1.

C. Processing Requirements.- None

D. Constraints.- None

E. Supplementary Information.- A suggested implementation in the form of a detailed flowchart is found in Appendix $\underline{D}$ under the name ONORBIT_REND_ USER_PARAM_STATE_PROP.

TABLE 4.5.1.- ONORBIT_REND_USER_PARAM_STATE_PROP INPUT/OUTPUT

Variable Name	Input Source	Output Destination
DOING_REND_NAV	**	
FILT_UPDATE	**	
R _RESET	**	
R _TV_RESET	**	
T_IMU		**, AVERAGE_G_ INTEGRATOR
T_RESET	**	
UPP_USE_IMU_DATA		**
V _IMU_SNAP		**
V _IMU_RESET	**	
V _RESET	**	
V _TV_RESET	**	
A _SENSED		**, AVERAGE_G_ INTEGRATOR
DT_IMU		AVERAGE_G_INTEGRATOR
R _AVGG	AVERAGE_G_INTEGRATOR	**, AVERAGE_G_ INTEGRATOR,ONORBIT_ USER_PARAMETER_ CALCULATIONS
R _TARGET	AVERAGE_G_INTEGRATOR	**, AVERAGE_G_ INTEGRATOR,ONORBIT_ USER_PARAMETER_ CALCULATIONS
GO	***	
PWRD_FLT_NAV	**	

**User parameter processing principal function, see section 4.5

***Initialization parameter, see section 4.7

TABLE 4.5.1.- ONORBIT_REND_USER_PARAM_STATE_PROP INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
T_STATE		**, AVERAGE_G_ INTEGRATOR, ONORBIT_ USER_PARAMETER_ CALCULATIONS
$\underline{V}$ _AVGG	AVERAGE_G_ INTEGRATOR	**, AVERAGE_G_ INTEGRATOR, ONORBIT_ USER_PARAMETER_ CALCULATIONS
$\underline{V}$ _IMU_OLD		**
$\underline{V}$ _TARGET	AVERAGE_G_ INTEGRATOR	**, AVERAGE_G_ INTEGRATOR, ONORBIT_ USER_PARAMETER_ CALCULATIONS

[**] User parameter processing principal function, see section 4.5

4.5.1.1 Integration (AVERAGE_G_INTEGRATOR)

The integration subfunction is called by the user parameter state propagation subfunction to propagate the Shuttle and target state vectors to current time.

A. Detailed Requirements.- This subfunction is called with the following internal variables in the IN LIST and OUT LIST:

IN LIST: $\underline{R}$ _AV, $\underline{V}$ _AV, DTIME, $\underline{AC}$, T_STATE, T_IMU

OUT LIST: $\underline{R}$ _AV, $\underline{V}$ _AV

where

$\underline{R}$ _AV $\underline{V}$ _AV	user parameter state vector
DTIME	interval to propagate state vector over
$\underline{AC}$	acceleration vector
T_STATE	user parameter time tag
T_IMU	current time

The following steps shall be performed (in the order indicated);

1. By means of a call to the acceleration function, compute the gravitational acceleration for the input state vector and corresponding time tag (see section 4.2.4.1.1):

   $$\underline{GR} = \text{ACCEL\_ONORBIT}\ (\text{GM\_DEG\_LOW, GM\_ORD\_LOW, DFL\_AVG, VFLTV\_PRED, ATFL\_OV}, \underline{R}\text{\_AV}, \underline{V}\text{\_AV, T\_STATE})$$

2. Advance the position vector by the average-g method:

   $$\underline{R}\text{\_AV} = \underline{R}\text{\_AV} + \text{DTIME}\left(\underline{V}\text{\_AV} + .5\ \text{DTIME}\ (\underline{AC} + \underline{GR})\right)$$

3. Use this updated position vector and the current time to find a new value of the gravitational acceleration (see section 4.2.4.1.1):

   $$\underline{GR1} = \text{ACCEL\_ONORBIT}\ (\text{GM\_DEG\_LOW, GM\_ORD\_LOW, DFL\_AVG, VFLTV\_PRED, ATFL\_OV}, \underline{R}\text{\_AV}, \underline{V}\text{\_AV, T\_IMU})$$

4. Advance the velocity vector by the average-g method:

   $$\underline{V}\text{\_AV} = \underline{V}\text{\_AV} + \text{DTIME}\left(\underline{AC} + .5\ (\underline{GR} + \underline{GR1})\right)$$

   The quantities GM_DEG_LOW, GM_ORD_LOW, VFLTV_PRED, DFL_AVG, and ATFL_OV are pad-loaded values of the various acceleration constants, designed for high usage rates (see section 4.7 for values).

B. Interface Requirements.- The input and output variables are listed in table 4.5.1.1.

C. Processing Requirements.- None

D. Constraints.- None

E. Supplementary Information.- A suggested implementation in the form of a detailed flow chart is found in Appendix D under the name

AVERAGE_G_INTEGRATOR.

TABLE 4.5.1.1.- AVERAGE_G_INTEGRATOR INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
AC	A _SENSED	ONORBIT_REND_USER_ PARAM_STATE_PROP	
DTIME	DT_IMU	ONORBIT_REND_USER_ PARAM_STATE_PROP	
R _AV	R _AVGG	ONORBIT_REND_USER_ PARAM_STATE_PROP	
T_IMU	T_IMU	ONORBIT_REND_USER_ PARAM_STATE_PROP	
T_STATE	T_STATE	ONORBIT_REND_USER_ PARAM_STATE_PROP	
V _AV	V _AVGG	ONORBIT_REND_USER_ PARAM_STATE_PROP	
AC	A _SENSED	ONORBIT_REND_USER_ PARAM_STATE_PROP	
DTIME	DT_IMU	ONORBIT_REND_USER_ PARAM_STATE_PROP	
R _AV	R _TARGET	ONORBIT_REND_USER_ PARAM_STATE_PROP	
T_IMU	T_IMU	ONORBIT_REND_USER_ PARAM_STATE_PROP	
T_STATE	T_STATE	ONORBIT_REND_USER_ PARAM_STATE_PROP	
V _AV	V _TARGET	ONORBIT_REND_USER_ PARAM_STATE_PROP	
R _AV	R _AVGG		ONORBIT_REND_USER_ PARAM_STATE_PROP

TABLE 4.5.1.1.- AVERAGE_G_INTEGRATOR INPUT/OUTPUT.- Continued

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
V _AV	V _AVGG		ONORBIT_REND_USER_PARAM_STATE_PROP
R _AV	R _TARGET		ONORBIT_REND_USER_PARAM_STATE_PROP
V _AV	V _TARGET		ONORBIT_REND_USER_PARAM_STATE_PROP

TABLE 4.5.1.1.- AVERAGE_G_INTEGRATOR INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
ATFL_OV	**	ACCEL_ONORBIT
DFL_AVG	**	ACCEL_ONORBIT
GM_DEG_LOW	**	ACCEL_ONORBIT
GM_ORD_LOW	**	ACCEL_ONORBIT
VFLTV_PRED	**	ACCEL_ONORBIT
†	ACCEL_ONORBIT	
R _AV		ACCEL_ONORBIT
V _AV		ACCEL_ONORBIT
T_STATE		ACCEL_ONORBIT
T_IMU		ACCEL_ONORBIT

**Initialization parameters, see section 4.7
†Value returned from the function

4.5.2 Onorbit User Parameter Calculations (ONORBIT_USER_PARAMETER_CALCULATIONS)

This subfunction contains software to compute quantities required by the GN&C/SM-PL interface and the universal pointing principal function. These calculations are required during OPS 2.

A. Detailed Requirements. - The following steps shall be performed (in the order indicated):

1. If the rendezvous nav flag is set on (DOING_REND_NAV = ON), the delta position and velocity vectors in the M50 system are computed. These vectors represent position and velocity vectors from the Shuttle to the rendezvous target.

$$\mathrm{\underline{DEL}\_R\_TARG = \underline{R}\_TARGET - \underline{R}\_AVGG}$$

$$\mathrm{\underline{DEL}\_V\_TARG = \underline{V}\_TARGET - \underline{V}\_AVGG}$$

2. The M50 to Greenwich-true-of-date transformation matrix is computed at the time of the state vectors, T_STATE.

$$\mathrm{M50\_TO\_EF = EARTH\_FIXED\_TO\_M50\_COORD(T\_STATE)^T}$$

3. The Shuttle state vector ($\mathrm{\underline{R}\_AVGG}$, $\mathrm{\underline{V}\_AVGG}$) is transformed into the Greenwich-true-of-date, and the Earth's rotational effects are added to the X and Y components to yield the Earth-relative velocity vector.

$$\mathrm{\underline{R}\_EF = M50\_TO\_EF\ \underline{R}\_AVGG}$$

$$\mathrm{\underline{V}\_EF = M50\_TO\_EF\ \underline{V}\_AVGG}$$

$$\mathrm{V\_RHO\_EF_1 = V\_EF_1 + EARTH\_RATE\ R\_EF_2}$$

$$\mathrm{V\_RHO\_EF_2 = V\_EF_2 - EARTH\_RATE\ R\_EF_1}$$

$$\mathrm{V\_RHO\_EF_3 = V\_EF_3}$$

4. The Greenwich mean time associated with the position and velocity vectors is then saved:

$$\mathrm{T\_SEC\_GMT = T\_STATE}$$

B. Interface Requirements.- Input and output parameters for this subfunction are shown in table 4.5.2.

C. Processing Requirements.- Processing requirements for this subfunction are specified in the ONORBIT_REND_UPP_SEQUENCER.

D. Constraints.- None

E. Supplemental Information.- A suggested implementation in the form of a detailed flowchart can be found in Appendix D under the name:

ONORBIT_USER_PARAMETER_CALCULATIONS

TABLE 4.5.2.- ONORBIT_USER_PARAMETER_CALCULATIONS INPUT/OUTPUT

Variable Name	Input Source	Output Destination
EARTH_RATE	**	
$\underline{R}$ _AVGG	ONORBIT_REND_USER_PARAM_STATE_	
DOING_REND_NAV	***	
$\underline{R}$ _TARGET	ONORBIT_REND_USER_PARAM_STATE_PROP	
T_STATE	ONORBIT_REND_USER_PARAM_STATE_PROP	
$\underline{V}$ _AVGG	ONORBIT_REND_USER_PARAM_STATE_PROP	
$\underline{V}$ _TARGET	ONORBIT_REND_USER_PARAM_STATE_PROP	
†	EARTH_FIXED_TO_M50_COORD	
$\underline{D}$EL_R_TARG		***
$\underline{D}$EL_V_TARG		***
T_SEC_GMT		***
$\underline{V}$ _RHO_EF		***
$\underline{R}$ _EF		***

**Initialization parameters, see section 4.7

***User parameter processing principal function, see section 4.5

†Value returned from the function

4.6 LANDING SITE UPDATE PRINCIPAL FUNCTION (LANDING_SITE_UPDATE_PRINCIPAL_FUNCTION)

This function provides the capability to reconfigure the dynamic parameters pertaining to the runway and TACAN sites, which are to carry over into OPS 3 for use during the deorbit through landing operational sequence. This section describes the software processing for data transfer from maxi to mini tables for runway and TACAN parameters. Output from this function is via the OPS 2 data list. The requirements of this function are to give the crew specific control via keyboard input to:

- Effect a maxi to mini table transfer of runway data (including MSBLS where available) and the TACAN's appropriate for the primary runway selection.

- View the current (latest) choice of runway identifying parameters on the orbit OPS display.

- Provide the current mini table data at the end of OPS 2 to be available for use in OPS 3. If this function is not executed in OPS 2, the I-Loaded mini tables contain default values appropriate for nominal end of mission use in OPS 3.

The guidance targeting uplink OPS formats (PEG 4 and PEG 7) also contain parameters identifying a request for maxi to mini data transfer. Thus the appropriate uplink processors may also invoke this process and provide the necessary input data.

A. Detailed Requirements.- The landing site update principal function may be invoked during OPS 2 by the universal pointing display function. This function may be performed whenever the OPS display is active and will respond to crew input from the keyboard. This function may also be invoked by the uplink processors for the PEG 4 and PEG 7 command uplink.

The principal function requires two tables in memory for both runways and TACAN:

- Runway and TACAN maxi tables

- Runway and TACAN mini tables

The runway maxi table consists of data sets for 18 runways and 4 MSBLS to support contingency landing sites which are grouped by geographic location.

Runway Maxi Table

Runways	MSBLS[*]	Geographic Locations
1-3	1-2	A
4-6	3-4	B
7-8		C
9-10		D
11-12		E
13-14		F
15-16		G
17-18		H

The runway mini table contains data for three runways and two MSBLS installations. The data in this table is normally I-Loaded to support the nominal end of mission (OPS-3); however, these data may be overwritten by invoking the principal function to transfer contingency site data. The mini table is considered to consist of runways 19, 20 and 21 initially, that is:

Runway Mini Table

Runway No	Runway Designation
19[**]	PRIMARY (Runway and associated MSBLS site data)
20	SECONDARY (Runway and associated MSBLS site data)
21	ALTERNATE (Runway site data)

The parameters PRI_MAXI_CURRENT, SEC_MAXI_CURRENT, and ALT_MAXI_CURRENT are used to specify which of the maxi table runways have been transferred to the respective mini table locations. Prior to any such transfer, the parameters should be initialized to the values 19, 20, and 21, respectively, to indicate that the I-Loaded mini table values are current and have not been overwritten. Following any transfer of data, the appropriate parameters will be updated to identify the current contents in the runway mini table. These three parameters are output to the universal pointing display.

The runway tables consist of data sets of the following parameters:

Geodetic latitude
Longitude
Runway azimuth
Altitude above reference ellipsoid
Δh for MSL at runway site (from reference ellipsoid)
Magnetic variation
Runway name (six character string; example: EDW17)
MLS_AVAIL (primary and secondary runways only)

---

[*] MSBLS installations are available for runways 1, 2, 4 and 5.

[**] For any of these initial contents which are overwritten, it may thereafter consist only of runway number 1 to 18.

The following parameters are associated with available MSBLS SITES; i.e., for runways in maxi table slots 1, 2, 4, and 5 and for the I-Loaded mini table runways designated PRIMARY and SECONDARY.

LAT_MLS_R_AZ
LONG_MLS_R_AZ
ALT_MLS_R_AZ
R_AZ_SCANNER_BEARING
LAT_MLSEL
LONG_MLSEL
ALT_MLSEL
EL_SCANNER_BEARING
BIAS_MLS_RANGE
BIAS_AZMLS
BIAS_ELMLS
X_DMEAZ_RW
X_EL_RW
Y_DMEAZ_RW

The TACAN maxi table consists of 50 TACAN's grouped according to proximity to the approach path to landing at a particular runway site; they are grouped as follows:

TACAN	Runway Geographic Location
1-10	A
11-20	B
21-25	C
26-30	D
31-35	E
36-40	F
41-45	G
46-50	H

The TACAN mini table is I-Loaded to support the initial (I-Load) mini table runways, but it shall be reconstructed automatically by a transfer of a maxi table runway to the mini table primary runway. For example, if any maxi table runway in geographic area B is transferred to the mini table primary runway, then the TACAN's for area B (# 11-20) will also be transferred to the TACAN mini table.

The TACAN tables contain the following data for each TACAN site:

TACAN channel number {Positive for X-mode / Negative for Y-mode}

Geodetic latitude
Longitude
Height above reference ellipsoid
Δh for MSL correction at TACAN site
Magnetic variation

TACAN bearing bias *
TACAN range bias *

The runway maxi table to mini table transfer is performed on demand and expects one or more of the discretes, ITEM1_IN, ITEM2_IN, or ITEM3_IN to be ON. If ITEM1_IN (primary runway) or ITEM2_IN (secondary runway) is set to ON, both runway site data and the corresponding MSBLS site data (if available, that is, for choice of runway = 1, 2, 4, or 5) from the maxi table is to be loaded into the primary or secondary data locations, respectively, within the mini table. If ITEM3_IN (alternate runway) is set to ON, only runway site data are to be loaded into the alternate runway data locations within the mini table. The integers PRI_MAXI_SELECT, SEC_MAXI_SELECT, and ALT_MAXI_SELECT indicate which data set within the maxi table is to be transferred to the corresponding data set within the mini table. After performing the data transfer, the appropriate ITEM1_IN, ITEM2_IN, or ITEM3_IN is set to ON for output.

Whenever the primary runway data has been altered by a maxi to mini data transfer, TACAN data are to be automatically transferred from the maxi TACAN table to the mini TACAN table according to the primary runway's geographic location; specifically, if ITEM1_IN is "ON" and PRI_MAXI_SELECT has a value of 1, 2, or 3, then TACAN data sets 1-10 from the maxi TACAN table are to be transferred to the mini table. Likewise, if PRI_MAXI_SELECT has a value of 4, 5, or 6, then TACAN data sets 11-20 from the maxi TACAN table are to be transferred to the mini table. Finally, whenever PRI_MAXI_ SELECT is 7 through 10, or 11 through 14, or 15 through 18, TACAN data sets 21-30 or 31-40 or 41-50, respectively, from the maxi table are to be transferred to the mini table. Note that transfer into secondary or alternate mini table runway slots does not cause additional TACAN transfers; this is operationally redundant.

The ORBIT OPS display also provides additional identification of the mini table contents. That is, it displays RUNWAY_NAME_PSL, RUNWAY_NAME_SSL, and RUNWAY_NAME_ASL, which are contained in the runway mini table. These parameters consists of 6 characters - three letters and two numbers and a blank - e.g: KSC15 or KSC33 for the RTLS runways at KSC.

B. Interface Requirements.- The initial values of data for the landing site and TACAN minitables are defined in the I-load by the parameter names given in table 4.6-1 (left-most column); normally these data are provided to support the nominal end of mission for the entry through landing in OPS 3. The parameter names for the minitables configured in this OPS for use in OPS 3 are given also in table 4.6-1 (right-most column). It is only these latter minitables (right-most column) which may be updated by the maxi to mini transfers, or uplink, for use in OPS 3, while the I-load parameters (left-most column) are preserved for subsequent transitions into OPS 2. Therefore, for certain transitions into OPS 2, it may be necessary to ensure that those

---

*Contained only in the mini table; assumed equal to zero for maxi table TACANs.

parameters in the right-hand column are initialized with the I-load values, while for other transitions into OPS 2 reinitialization must be avoided. Specific conditions for these initializations are given in paragraph C, Processing Requirements.

The maxitables are I-loaded with the parameter names which are used by the software; therefore, they are always initialized by the I-load values whenever OPS 2 is entered.

The inputs and outputs for this function are shown in table 4.6-2. The maxitables (4.6-2, Inputs) are initilized directly by I-loads, but may be updated by uplink. The minitables (4.6-2, Outputs) may (depending on previous OPS) be initialized by I-load, may be updated by uplink, and may be updated by this spec function.

The maxitable locations of the current primary, secondary, and alternate runways selected are initialized to 19, 20, and 21 (parameters PRI_MAXI_CURRENT, SEC_MAXI_CURRENT, and ALT_MAXI_CURRENT, respectively). The I-load parameters interface is shown in table 4.6-1 and table 4.6-3.

C. Processing Requirements.- As stated in paragraph B, it was necessary to determine whether the minitables which are reconfigured in this OPS (OPS 2) are to be initialized to the values provided by the I-load, or whether values from the previous OPS should be retained. If the transition into OPS 2 is from either OPS 1 or OPS 3, then the I-load values are to be used; but if the transition into OPS 2 is from any OPS other than OPS 1 or OPS 3, the values from that preceding OPS should be retained.

It is not intended that this spec function be invoked for a nominal mission; therefore, the orbit OPS display should show current values prior to execution of this spec. The minitable carried across the OPS 2 to OPS 3 transition should contain the I-loaded data if the spec is not invoked and an uplink has not occurred; it should contain the current reconfiguration data if the spec had been invoked or an uplink had occurred.

D. Constraints.- Any initializations that may be required by the implementation of paragraph C to provide the appropriate data for the interface tables of paragraph B must precede the execution of this spec function, precede uplink to the minitables, precede the use of outputs from this function, and precede the transition out of OPS 2. Because outputs support the orbit OPS display, it is recommended that any special processing required be accomplished at the beginning of the OPS 2.

E. Supplementary Information.- The flight software I-load contains three minitables of Runway and TACAN site data:

One table appropriate for RTLS (used in OPS 1/6).

One table appropriate for AOA (used in OPS 3, but only when OPS 3 is entered from OPS 1).

One table appropriate for End of Mission (used in OPS 3, but initialized in OPS 2 for nominal End of Mission). The End of Mission table may be reconfigured during OPS 2 via maxi mini transfers and/or uplink with several alternate data sets; the reconfigured tables should be used in OPS 3 <u>except</u> for AOA, which should use data provided by I-load for an AOA.

The tabular I-loaded RTLS data are identified by "use table" parameter names which have a specification of memory configuration 01. The AOA tables have unique I-load parameter names ending with "-E"; and a specification of MC03.

The nominal End-of-Mission tables have unique I-load parameter names ending with "-O" and a specification of MC02.

A suggested implementation of these requirements is illustrated by a detailed flow diagram shown in appendix E under the following name:

LANDING_SITE_UPDATE

TABLE 4.6-1.- CORRESPONDENCE OF NOMINAL END OF MISSION I-LOAD NAMES TO OPS-2 "USE TABLE" NAMES

OPS-2 Nominal End-of-Mission Parameters	OPS-2 Use Table Parameters
RW_LAT(1)_0	RW_LAT_PSL
RW_LON(1)_0	RW_LON_PSL
RW_AZIMUTH(1)_0	RW_AZIMUTH_PSL
RUNWAY_ALT(1)_0	RUNWAY_ALT_PSL
RW_DELH(1)_0	RW_DELH_PSL
RW_MAG_VAR(1)_0	RW_MAG_VAR_PSL
LAT_MLS_R_AZ(1)_0	LAT_MLS_R_AZ_PSL
LONG_MLS_R_AZ(1)_0	LONG_MLS_R_AZ_PSL
R_AZ_RADAR_BEARING(1)_0	R_AZ_RADAR_BEARING_PSL
ALT_MLS_R_AZ(1)_0	ALT_MLS_R_AZ_PSL
LAT_MLSEL(1)_0	LAT_MLSEL_PSL
LONG_MLSEL(1)_0	LONG_MLSEL_PSL
ALT_MLSEL(1)_0	ALT_MLSEL_PSL
EL_SCANNER_BEARING(1)_0	EL_SCANNER_BEARING_PSL
BIAS_MLSRANGE(1)_0	BIAS_MLSRANGE_PSL
BIAS_AZMLS(1)_0	BIAS_AZMLS_PSL
BIAS_ELMLS(1)_0	BIAS_ELMLS_PSL
X_DMEAZ_RW(1)_0	X_DMEAZ_RW_PSL
X_EL_RW(1)_0	X_EL_RW_PSL

TABLE 4.6-2.- LANDING_SITE_UPDATE PRINCIPAL FUNCTION INPUT/OUTPUT

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
ALT_MAXI_SELECT	Univ Point(6.27), Uplink PROC.	LANDING_SITE_UPDATE		
ITEM1_IN	Univ Point(6.27), Uplink PROC.	LANDING_SITE_UPDATE		
ITEM2_IN	Univ Point(6.27), Uplink PROC.	LANDING_SITE_UPDATE		
ITEM3_IN	Univ Point(6.27), Uplink PROC.	LANDING_SITE_UPDATE		
PRI_MAXI_SELECT	Univ Point(6.27), Uplink PROC.	LANDING_SITE_UPDATE		
SEC_MAXI_SELECT	Univ Point(6.27), Uplink PROC.	LANDING_SITE_UPDATE		
ALT_MAXI_CURRENT			Univ Point (6.27)	LANDING_SITE_UPDATE
PRI_MAXI_CURRENT			Univ Point (6.27)	LANDING_SITE_UPDATE
RUNWAY_NAME_ASL			Univ Point (6.27)	MAXI_RWY_TRANSFER CODE
RUNWAY_NAME_PSL			Univ Point (6.27)	MAXI_RWY_TRANSFER CODE

TABLE 4.6-2.- LANDING_SITE_UPDATE PRINCIPAL FUNCTION INPUT/OUTPUT.- Concluded

Variable Name	Principal Function Source	Local Destination	Principal Function Destination	Local Source
RUNWAY_NAME_SSL			Univ Point (6.27)	MAXI_RWY_TRANSFER_CODE
SEC_MAXI_CURRENT			Univ Point (6.27)	LANDING_SITE_UPDATE
RUNWAY MAXI TABLE PARAMETERS (SEE TABLE 4.6-3)	I-LOAD			
TACAN MAXI TABLE PARAMETERS (SEE TABLE 4.6-3)	I-LOAD			
RUNWAY MINI TABLE PARAMETERS (SEE TABLE 4.6-3)			I-LOAD	
TACAN MINI TABLE PARAMETER (SEE TABLES 4.6-3)			I-LOAD	

TABLE 4.6-3.- LANDING_SITE_UPDATE PRINCIPAL FUNCTION I-LOAD PARAMETERS

Level C Symbol	Input Source
RUNWAY MINI TABLE	
RW_LAT_PSL	**
RW_LON_PSL	**
RW_AZIMUTH_PSL	**
RUNWAY_ALT_PSL	**
RW_DELH_PSL	**
RW_MAG_VAR_PSL	**
LAT_MLS_R_AZ_PSL	**
LONG_MLS_R_AZ_PSL	**
R_AZ_RADAR_BEARING_PSL	**
ALT_MLS_R_AZ_PSL	**
LAT_MLSEL_PSL	**
LONG_MLSEL_PSL	**
ALT_MLSEL_PSL	**
EL_SCANNER_BEARING_PSL	**
BIAS_MLSRANGE_PSL	**
BIAS_AZMLS_PSL	**
BIAS_ELMLS_PSL	**
X_DMEAZ_RW_PSL	**
X_EL_RW_PSL	**
Y_DMEAZ_RW_PSL	**

**Initialization parameters, see section 4.7 (These parameters may also be dynamically updated.)

TABLE 4.6-3.- LANDING_SITE_UPDATE PRINCIPAL FUNCTION I-LOAD PARAMETERS.- Continued

Level C Symbol	Input Source
MLS_AVAIL_PSL	**
RUNWAY_NAME_PSL	**
RW_LAT_SSL	**
RW_LON_SSL	**
RW_AZIMUTH_SSL	**
RUNWAY_ALT_SSL	**
RW_DELH_SSL	**
RW_MAG_VAR_SSL	**
LAT_MLS_R_AZ_SSL	**
LONG_MLS_R_AZ_SSL	**
R_AZ_RADAR_BEARING_SSL	**
ALT_MLS_R_AZ_SSL	**
LAT_MLSEL_SSL	**
LONG_MLSEL_SSL	**
ALT_MLSEL_SSL	**
EL_SCANNER_BEARING_SSL	**
BIAS_MLSRANGE_SSL	**
BIAS_AZMLS_SSL	**
BIAS_ELMLS_SSL	**
X_DMEAZ RW_SSL	**

**Initialization parameters, see section 4.7 (These parameters may also be dynamically updated.)

TABLE 4.6-3.- LANDING_SITE_UPDATE PRINCIPAL FUNCTION I-LOAD PARAMETERS.- Continued

Level C Symbol	Input Source
X_EL_RW_SSL	**
Y_DMFAZ_RW_SSL	**
MLS_AVAIL_SSL	**
RUNWAY_NAME_SSL	**
RW_LAT_ASL	**
RW_LON_ASL	**
RW_AZIMUTH_ASL	**
RUNWAY_ALT_ASL	**
RW_DELH_ASL	**
RW_MAG_VAR_ASL	**
RUNWAY_NAME_ASL	**
TACAN MINI TABLE	
LATITUDE_GEODETIC	**
LONGITUDE_EAST	**
ALT_ABOVE_ELLIPSOID	**
MSL_ABOVE_ELLIPSOID	**
MAGNETIC_VARIATION	**
TAC_ID	**
RUNWAY MAXI TABLE	
RW_LAT	**

** Initialization parameters, see section 4.7 (These parameters may also be dynamically updated.)

TABLE 4.6-3.- LANDING_SITE_UPDATE PRINCIPAL FUNCTION I-LOAD PARAMETERS.- Continued

Level C Symbol	Input Source
RW_LON	**
RW_AZIMUTH	**
RUNWAY_ALT	**
RW_DELH	**
RW_MAG_VAR	**
RUNWAY_NAME	**
LAT_MLS_R_AZ	**
LONG_MLS_R_AZ	**
R _AZ_RADAR_BEARING	**
ALT_MLS_R_AZ	**
LAT_MLSEL	**
LONG_MLSEL	**
ALT_MLSEL	**
EL_SCANNER_BEARING	**
BIAS_MLS_RANGE	**
BIAS_AZML	**
BIAS_ELMLS	**
X _DMEAZ_RW	**
X _EL_RW	**
Y _DMEAZ_RW	**

** Initialization parameters, see section 4.7 (These parameters may also be dynamically updated.)

TABLE 4.6-3.- LANDING_SITE_UPDATE PRINCIPAL FUNCTION I-LOAD PARAMETERS.- Concluded

Level C Symbol	Input Source
TACAN MAXI TABLE	
TACAN_ID_MAXI	**
LATITUDE_GEODETIC_MAXI	**
LONGITUDE_EAST_MAXI	**
ALT_ABOVE_ELLIPSOID_MAXI	**
MSL_ABOVE_ELLIPSOID_MAXI	**
MAGNETIC_VARIATION_MAXI	**

**Initialization parameters, see section 4.7 (These parameters may also be dynamically updated.)

4.7 INITIALIZATION OF SPECIFIC OPS 2 NAVIGATION PARAMETERS

It is required that initial values for certain specific parameters be made available for use by the navigation flight software. These parameters can be divided into six categories:

1. Level A Constants - Those parameters with values that are not mission related nor design dependent. These parameters and values are defined in Level A CPDS (SS-P-0002-170D).

2. Mission Dependent Parameters - Those parameters with values that may change from mission to mission. These parameters and their values will be defined in an I-LOAD FSSR that has not been written as of the date of this document.

3. Design Dependent Parameters - Those parameters whose values are considered to be a part of the software design. These parameters and values are defined in this book.

4. Hard Codeable Parameters - Those parameters whose values must be defined but are neither mission dependent nor design dependent and are not defined by the Level A CPDS.

5. Other required initial values - Those parameters whose values do not fit in any of the four above categories.

6. OPS Transition Parameters - Those parameters whose values are needed by the OPS 2 or OPS 8 navigation flight software, or calculated by OPS 2 for an OPS transition to OPS 8, OPS 3, or OPS 00, or calculated by OPS 8 for an OPS transition to OPS 2, OPS 3, or OPS 00.

These six categories of initialization parameters are denoted in the INITIALIZATION CATEGORY column of the variable lists in the appendices by the following abbreviations:

C - Level A constants

MD - Mission dependent parameters

DD - Design dependent parameters

HC - Hard codeable parameters

IV - Other required initial values

OPS - OPS transition parameters

4.8 DOWNLIST REQUIREMENTS

Downlist requirements for the Onorbit/Rendezvous (OPS-2) operations computer load are given in the Computer Program Development Specification (CPDS) Vol 1, Book 4, Downlist/Uplink Software Requirements (SS-P-0002-140F). Data that will be available for downlist may be found in the principal function input/output tables marked with TLM as output destination. They may also be found in the variable lists of appendices A, C, D, E and F indicated by the word "downlist" in the Uplink/Downlist column of the variable lists.

4.9 UPLINK REQUIREMENTS

The following subsections identify uplink requirements to support the orbital operations computer load onboard navigation software. Each subsection identifies a specific set of parameters which are candidates for uplink and details the structure of the formatted load, specifies any OP-CODE checking and/or special processing required as a result of the occurrence of the particular uplinked data, and describes any special constraints involved. All navigation related uplink parameter requirements are grouped into the following categories:

- Orbiter state vector uplink parameters (section 4.9.1)
- Rendezvous vehicle state vector uplink parameters (section 4.9.2)
- Vent/RCS thrust body force vector uplink parameters (section 4.9.3)
- Drag model correction factor uplink parameter (section 4.9.4)
- Covariance matrix uplink parameters (section 4.9.5)
- Landing/TACAN table uplink processing (section 4.9.6)
- Landing site selection parameters (section 4.9.7)

4.9.1 Orbiter State Vector Uplink Parameters (ORBITER_STATE_VECTOR_UPLINK)

The parameters to be candidates for change via command uplink associated with the Crbiter state vector update include the Orbiter M50 position and velocity vectors, and the associated time tag.

A. Detailed Requirements.- The Orbiter M50 position and velocity vectors with the associated time tag (GMT) can be changed via a formatted load structured as described in table 4.9.1-1. The following high-rate special processing is required upon receipt of an Orbiter state vector command uplink load:

1. A bit-string (OP-CODE) shall have been set to the "Orbiter state vector uplink" OP-CODE value (0001001) by the ground uplink processing software when data are received from the ground.

2. The bit-string (OP-CODE) shall first be tested by the special processing software to determine if it equals the value specified for an Orbiter state uplink:

```
!
! OP-CODE = 0001001
!
```

3. If the above is true, a flag (DO_OV_UPLINK) shall be set to ON, and the uplinked Orbiter state vector with associated time tag shall be stored in special locations:

   DO_OV_UPLINK = ON
   R_GND = BUFFER_R
   V_GND = BUFFER_V
   T_GND = BUFFER_T

   and the OP-CODE shall be nulled.

   OP-CODE = 0000000

   Once the special high-rate special processing has been performed by the ground uplink software, additional low-rate special processing is required to be performed at a particular point during execution of the next navigation cycle. Detailed requirements for the low-rate special processing requirements are presented in section 4.2.5.1.

B. Interface Requirements.- The input/output required for the high-rate special processing s/w are listed in table 4.9.1-2. Required input and output for low-rate special processing s/w are listed in table 4.2.5.1.

C. Processing Requirements.- The high rate special processing detailed in this section shall be performed at a fast enough rate so that if two command uplinks (Orbiter and target) are transmitted between navigation cycles, both sets of data (including the DO_OV_UPLINK and DO_TV_UPLINK flags) will be available to the low - (nav) - rate special processing software. The low-rate special processing (section 4.2.5.1) shall be performed during a navigation cycle immediately following the normal state vector propagation and during rendezvous navigation phases covariance matrix propagation subfunctions.

D. Constraints.- Because of a system software requirement to clear uplink data buffers in a relatively short amount of time, and because of the relatively slow rate at which the onorbit/rendezvous navigation principal function will be operating, a fast-rate special processing function is required to buffer off the data and set flags. There is also a requirement to be able to update both the Orbiter and the target states in a single navigation cycle, if such data has been uplinked.

E. Supplementary Information.- A suggested implementation of the Orbiter state vector formatted load special processing requirements in the form of detailed flowcharts, are presented in Appendix F under:

ONORBIT_REND_AUTO_INFLIGHT_UPDATE (low-rate special processing)
ORBITER_STATE_VECTOR_UPLINK (high-rate special processing)

TABLE 4.9.1-1.- ORBITER STATE VECTOR UPLINK DATA FORMAT

Level B Mnemonic	UPLINK BUFFER DATA		Command Word	Data Type	Units
	Name	Description			
	BUFFER_R_1	X-Position in M50 coord	CW3, CW4	DF	ft
	BUFFER_R_2	Y-Position in M50 coord	CW5, CW6	DF	ft
	BUFFER_R_3	Z-Position in M50 coord	CW7, CW8	DF	ft
	BUFFER_V_1	X-Velocity in M50 coord	CW9	SF	ft/sec
	BUFFER_V_2	Y-Velocity in M50 coord	CW10	SF	ft/sec
	BUFFER_V_3	Z-Velocity in M50 coord	CW11	SF	ft/sec
	BUFFER_T	Time Tag in M50	CW1, CW2	DF	sec

TABLE 4.9.1-2.- ORBITER STATE VECTOR UPLINK INPUT/OUTPUT

Variable Name	Input Source	Output Destination
BUFFER_R	GROUND UPLINK PROCESSOR	
BUFFER_V	GROUND UPLINK PROCESSOR	
BUFFER_T	GROUND UPLINK PROCESSOR	
DO_OV_UPLINK		NAV_ONORBIT_RENDEZVOUS
R _GND		ONORBIT_REND_AUTO_INFLIGHT_UPDATE
T_GND		ONORBIT_REND_AUTO_INFLIGHT_UPDATE
V _GND		ONORBIT_REND_AUTO_INFLIGHT_UPDATE

4.9.2 Rendezvous Vehicle State Vector Uplink Parameters (RENDEZVOUS_STATE_VECTOR_UPLINK)

The parameters to be candidates for change via command uplink associated with the rendezvous vehicle state vector update include the rendezvous vehicle M50 position and velocity vectors, and the associated time tag.

A. Detailed Requirements.- The rendezvous vehicle M50 position and velocity vectors can be changed via a formatted load structured as described in table 4.9.2-1. Each load contains a time tag (GMT) associated with the state vector. The following high-rate special processing is required upon receipt of a rendezvous vehicle state vector command uplink load.

1. A bit-string (OP_CODE) shall have been set to the "rendezvous vehicle state vector" OP_CODE value (0001010) by the ground uplink processing software when data are received from the ground.

2. The bit string (OP_CODE) shall first be tested by the special processing software to determine if it equals the value specified for the rendezvous vehicle state uplink,

   OP_CODE = 0001010

3. If the above is true, a flag (DO_TV_UPLINK), shall be set to ON, and the rendezvous vehicle state vector and associated time tag shall be stored in special locations:

   DO_TV_UPLINK = ON
   R_TV_GND = BUFFER_RT
   V_TV_GND = BUFFER_VT
   T_TV_GND = BUFFER_TT

   and the OP_CODE shall be nulled signifying that the state vector has been transferred to the locations:

   OP_CODE = 0000000

   Once the above high-rate special processing has been performed by the ground uplink software, additional low-rate special processing is required to be performed at a particular point during execution of the next navigation cycle. This latter class of special processing will actually re-initialize the rendezvous vehicle state vector using the uplinked data. Detailed requirements for this (low-rate) portion of the special processing requirements are presented in section 4.2.5.1.

B. Interface Requirements.- The input and output required for the high-rate special processing software are listed in table 4.9.2-2. Input and output for the low-rate special processing software are listed in table 4.2.5.1.

C. Processing Requirements.- The high-rate special processing detailed in this section shall be performed at a fast enough rate so that if two command uplinks (Orbiter and/or target) are transmitted between navigation cycles, both data sets (including the DO_OV_UPLINK and DO_TV_UPLINK flags) will be available to the low -(nav)- rate special processing software. The low-rate special processing (section 4.2.5.1) shall be performed during a navigation cycle immediately following normal state vector propagation and, during rendezvous, covariance matrix propagation subfunctions.

D. Constraints.- Because of a system software requirement to clear uplink data buffers in a relatively short amount of time, and because of the relatively slow rate at which the onorbit/rendezvous navigation principal function will be operating, a fast-rate special processing function is required to buffer off the data and set flags. There is also a requirement to be able to update both the Orbiter and rendezvous vehicle state vectors on a single navigation cycle if such data have been uplinked.

E. Supplementary Information.- A suggested implementation of the formatted rendezvous vehicle state vector load special processing requirements, in the form of detailed flowcharts are presented in Appendix F under:

ONORBIT_REND_AUTO_INFLIGHT_UPDATE (low-rate special processing)
RENDEZVOUS_STATE_VECTOR_UPLINK (high-rate special processing)

TABLE 4.9.2-1- RENDEZVOUS VEHICLE STATE VECTOR UPLINK DATA FORMAT

Level B Mnemonic	UPLINK BUFFER DATA		Command Word	Data Type	Units
	Name	Description			
	$BUFFER\_RT_1$	X-Position in M50 coord	CW3, CW4	DF	ft
	$BUFFER\_RT_2$	Y-Position in M50 coord	CW5, CW6	DF	ft
	$BUFFER\_RT_3$	Z-Position in M50 coord	CW7, CW8	DF	ft
	$BUFFER\_VT_1$	X-Velocity in M50 coord	CW9	SF	ft/sec
	$BUFFER\_VT_2$	Y-Velocity in M50 coord	CW10	SF	ft/sec
	$BUFFER\_VT_3$	Z-Velocity in M50 coord	CW11	SF	ft/sec
	BUFFER_TT	Time Tag in M50	CW1, CW2	DF	sec

TABLE 4.9.2-2.- RENDEZVOUS VEHICLE STATE VECTOR UPLINK INPUT/OUTPUT

Variable Name	Input Source	Output Destination
BUFFER_RT	GROUND UPLINK PROCESSOR	
BUFFER_VT	GROUND UPLINK PROCESSOR	
BUFFER_TT	GROUND UPLINK PROCESSOR	
DO_TV_UPLINK		NAV_ONORBIT_RENDEZVOUS
R _TV_GND		ONORBIT_REND_AUTO_INFLIGHT_UPDATE
V _TV_GND		ONORBIT_REND_AUTO_INFLIGHT_UPDATE
T_TV_GND		ONORBIT_REND_AUTO_INFLIGHT_UPDATE

4.9.3 Vent/RCS Body Force Vector Uplink Parameters (VENT_RCS_MODEL_PARAM_UPLINK)

The parameters to be candidates for change via the Vent/RCS thrusting uplink command load include a body contact force $\underline{\text{V}}$FORCE, to account for venting and uncoupled RCS thrusting and other body contact forces if desired, and the associated action ON/OFF times (TFON and TFOFF). These data are specified by premission I-load values (see section 4.7) or uplinked to be used by the Onorbit/Rendezvous Navigation principal function in both OPS 2 and OPS 8.

ONORBIT_RENDEZVOUS_NAV

A. Detailed Requirements.- The values of the body contact force ($\underline{\text{V}}$FORCE) and associated on-off times (TFON and TFOFF) may be changed via a formatted command load structured as described in table 4.9.3-1. The following processing will be initiated upon receipt of a Vent/RCS uplink from the ground.

1. A bit-string (OP_CODE) shall have been set to the "Vent/RCS Body Force Vector uplink" OP_CODE value (0101001) by the ground uplink processing software when data are received from the ground.

2. The bit-string (OP_CODE) shall first be tested by the processing software to determine if it equals the value specified for a Vent/RCS uplink:

OP_CODE = 0101001

3. If the above is true, the Vent/RCS Body Contact Force Vector $\underline{\text{V}}$FORCE and ON/OFF action times (TFON/TFOFF) shall be stored in special locations:

$\underline{\text{V}}$FORCE = BUFFER_B
TFON = BUFFER_O_1
TFOFF = BUFFER_O_2

and the OP_CODE shall be nulled.

OP_CODE = 0000000

B. Interface Requirements.- The input and output required for a Vent/RCS uplink are shown in table 4.9.3-2.

C. Processing Requirements.- The uplink Vent/RCS command load will cause revision of all entries; i.e.,

$\underline{\text{V}}$FORCE
TFON
TFOFF

For this reason, all entries of the Vent/RCS command load shall be filled with their desired values (those values desired immediately after the completion of the uplink) even if some of the entries are to remain the same.

D. Constraints.- None

E. Supplementary Information.- A suggested implementation of the Vent/RCS body contact force parameter uplink in the form of a detailed flowchart (VENT_RCS_MODEL_PARAM_UPLINK) is presented in Appendix F.

TABLE 4.9.3-1.- VENT/RCS BODY FORCE VECTOR UPLINK DATA FORMAT

Level B Mnemonic	UPLINK BUFFER DATA		Command Word	Data Type	Units
	Name	Description			
	BUFFER_B$_1$	Vent/RCS X-Body force in	CW5	SF	lb
	BUFFER_B$_2$	Vent/RCS Y-Body force body	CW6	SF	lb
	BUFFER_B$_3$	Vent/RCS Z-Body force coord	CW7	SF	lb
	BUFFER_O$_1$	Vent/RCS ON time	CW1, CW2	DF	sec
	BUFFER_O$_2$	Vent/RCS OFF time	CW3, CW4	DF	sec

TABLE 4.9.3-2.- VENT/RCS BODY FORCE VECTOR UPLINK INPUT/OUTPUT

Variable name	Input Source	Output Destination
BUFFER_B	GROUND UPLINK PROCESSOR	
BUFFER_O_1	GROUND UPLINK PROCESSOR	
BUFFER_O_2	GROUND UPLINK PROCESSOR	
V FORCE		ACCEL_ONORBIT
TFON		ACCEL_ONORBIT
TFOFF		ACCEL_ONORBIT

4.9.4 Drag Model Correction Factor Uplink Parameters (DRAG_MODEL_PARAM_UPLINK)

The drag model parameter which is a candidate for change via uplink is KFACTOR. Command load uplink capability of this parameter ensures that there exists a means whereby the Orbiter drag coefficient may be adjusted if necessary during an Orbiter propagation mode. The drag model is used by the onorbit/rendezvous navigation principal function during OPS 2 and OPS 8:

ONORBIT_RENDEZVOUS_NAV

A. Detailed Requirements.- The value of the drag model correction factor (KFACTOR) can be changed via a formatted command load structured as described in table 4.9.4-1. The following processing will be initiated upon receipt of a drag model correction factor uplink from the ground.

1. A bit-string (OP_CODE) shall have been set to the "Drag Model Correction Factor Uplink" OP_CODE (0101010) by the ground uplink processing software when data are received from the ground.

2. The bit-string (OP_CODE) shall first be tested by the processing software to determine if it equals the value specified for a drag model correction factor uplink.

   OP_CODE = 0101010

3. If the above is true, the drag model correction factor (KFACTOR) shall be stored in a special location:

   KFACTOR = BUFFER_K

   and the OP_CODE shall be nulled.

   OP_CODE = 0000000

B. Interface Requirements.- The input and output required for a drag model correction factor uplink are shown in table 4.9.4-2.

C. Processing Requirements.- None

D. Constraints.- None

E. Supplementary Information.- A suggested implementation of the KFACTOR update capability is presented in Appendix F in the form of a detailed flowchart, DRAG_MODEL_PARAM_UPLINK.

TABLE 4.9.4-1.- DRAG MODEL CORRECTION FACTOR UPLINK DATA FORMAT

Level B Mnemonic	UPLINK BUFFER DATA		Command Word	Data Type	Units
	Name	Description			
	BUFFER_K	Drag Model Correction Facter	CW1, CW2	DF	n.d.

TABLE 4.9.4-2.- DRAG MODEL CORRECTION FACTOR UPLINK INPUT/OUTPUT

Variable Name	Input Source	Output Destination
BUFFER_K	GROUND UPLINK PROCESSOR	
KFACTOR		ACCEL_ONORBIT

4.9.5 Covariance Matrix Uplink Parameters (COV_MATRIX_PARAM_UPLINK)

The parameters to be candidates for change via uplink associated with the covariance matrix update include diagonal elements (standard deviations) of the filter state covariance matrix in UVW coordinates and selected correlation coefficients.

A. Detailed Requirements.- Values of the pre-stored (I-Load) UVW covariance matrix can be changed via a formatted command load structured as described in table 4.9.5-1. Each load will contain data entries to replace the values of the UVW standard deviations (SIG_UPDATE). Detailed descriptions of these parameters can be found in section 4.2.5 and section 4.7. The following processing is required upon receipt of the covariance matrix uplink command load.

1. A bit-string (OP_CODE) shall have been set to the "Covariance Matrix Uplink" OP_CODE (0101011) by the ground uplink processing software when data are received from the ground.

2. The bit-string (OP_CODE) shall first be tested by the processing software to determine if it equals the value specified for a covariance matrix uplink:

   OP_CODE = 0101011

3. If the above is true, the standard deviations (SIG_UPDATE) and correlation coefficients (COV_COR_UPDATE) shall be stored in special locations:

   SIG_UPDATE = BUFFER_S

   COV_COR_UPDATE = BUFFER_C

   and the OP_CODE shall be nulled.

   OP_CODE = 0000000

B. Interface Requirements.- The input and output required for covariance matrix parameters uplink can be found in table 4.9.5-2.

C. Processing Requirements.- Changes to SIG_UPDATE and COV_COR_UPDATE will be made each time the covariance matrix parameters uplink is performed.

D. Constraints.- The user shall perform the covariance matrix uplink prior to the state vector uplink when rendezvous navigation is active and it is desired that the uplinked covariance matrix parameters be incorporated along with an uplinked state vector.

E. Supplementary Information.- A suggested implementation of the covariance matrix parameters uplink in the form of a detailed flowchart (COV_MATRIX_PARAM_UPLINK) is presented in Appendix F.

TABLE 4.9.5-1.- COVARIANCE MATRIX UPLINK DATA FORMAT

Level B Mnemonic	UPLINK BUFFER DATA		Command Word	Data Type	Units
	Name	Description			
	BUFFER_S$_1$	U position standard deviation	CW1	SF	ft
	BUFFER_S$_2$	V position standard deviation	CW2	SF	ft
	BUFFER_S$_3$	W position standard deviation	CW3	SF	ft
	BUFFER_S$_4$	U velocity standard deviation	CW4	SF	ft/sec
	BUFFER_S$_5$	V velocity standard deviation	CW5	SF	ft/sec
	BUFFER_S$_6$	W velocity standard deviation	CW6	SF	ft/sec
	BUFFER_C$_1$	U-V correlation coeff	CW7	SF	-
	BUFFER_C$_2$	U-$\dot{U}$ correlation coeff	CW8	SF	-
	BUFFER_C$_3$	U-$\dot{V}$ correlation coeff	CW9	SF	-
	BUFFER_C$_4$	V-$\dot{U}$ correlation coeff	CW10	SF	-
	BUFFER_C$_5$	V-$\dot{V}$ correlation coeff	CW11	SF	-
	BUFFER_C$_6$	W-$\dot{W}$ correlation coeff	CW12	SF	-
	BUFFER_C$_7$	$\dot{U}$-$\dot{V}$ correlation coeff	CW13	SF	-

TABLE 4.9.5-2.- COVARIANCE MATRIX UPLINK INPUT/OUTPUT

Variable Name	Input Source	Output Destination
BUFFER_S	GROUND UPLINK PROCESSOR	
BUFFER_C	GROUND UPLINK PROCESSOR	
SIG_UPDATE		COVINIT_UVW
COV_COR_UPDATE		COVINIT_UVW

4.9.6 Landing/TACAN Site Table Uplink Processing

This subsection provides the software requirements associated with data uplink to the landing site maxi-table, the landing site mini-table, the TACAN site maxi-table, and the TACAN site mini-table.

A. Detailed Requirements.- The data set associated with the landing site maxi- and mini-tables and the uplink is as follows.

Geodetic latitude
Longitude
Runway azimuth with regard to true north
Magnetic variation
Runway altitude above ellipsoid
MSL altitude above ellipsoid

Runway name identifier (one six character string word in maxi/mini tables-two four character string words required for uplink). In addition, the uplink data set contains a landing site location number which identifies which slot in the landing site maxi-or mini-table is to be changed.

The following special processing is required upon receipt of a landing site uplink as indicated by a flag, DO_LND_SITE_UPLINK, having been set to ON by the ground uplink processing software. The integer, LND_SITE_NO, shall be tested and, if found to lie in the range 1 to 18, the uplink data set (see table 4.9.6-1) shall be loaded into the corresponding slot in the landing site maxi-table. If LND_SITE_NO has a value of 19, 20 or 21, then the uplink data set shall be loaded into the primary, secondary, or alternate table slot, respectively, in the landing site mini-table. Whenever landing site data are loaded into the maxi-or mini-table, the MSBLS availability flag associated with that runway shall be reset to OFF. After the uplinked data have been transferred to the appropriate table slot, the DO_LND_SITE_UPLINK flag shall be reset to OFF.

When LND_SITE_NO lies outside the range 1 to 21, no data transfer shall occur.

The uplink data set associated with the TACAN site maxi- and mini-tables is as follows:

Geodetic latitude
Longitude
Magnetic variation
TACAN altitude above ellipsoid
MSL altitude above ellipsoid
TACAN channel/mode identifier
TACAN bearing bias (not stored in maxi table)
TACAN range bias (not stored in maxi table)

In addition, the TACAN uplink data set contains a TACAN site location number which identifies which slot in the TACAN site maxi- or mini-table is to be changed.

The following special processing is required upon receipt of a TACAN site table uplink as indicated by a flag, DO_TACAN_SITE_UPLINK, having been set to ON by the ground uplink processing software. The integer, TAC_SITE_NO, shall be tested and, if found to lie in the range 1 to 50, the uplink data set (see table 4.9.6-2) shall be loaded into the corresponding slot in the TACAN site maxi-table. If TAC_SITE_NO lies in the range 51 to 60, the uplink data shall be loaded into the TACAN site mini-table slot corresponding to TAC_SITE_NO minus 50. Finally the DO_TACAN_SITE_UPLINK flag shall be reset to OFF.

If TAC_SITE_NO lies outside the range 1 to 60, no data transfer shall occur.

B. Interface Requirements.- The input and output requirements for this subfunction are listed in tables 4.9.6-3 and 4.9.6-4, respectively.

C. Processing Requirements.- To be executed on demand and provided with required input data. If the uplink data words are in units other than radians for angles and feet for lengths, onboard conversion to these units are required before storing into maxi or mini tables.

D. Constraints.- The task of setting up the uplink buffer data sets and the uplink data availability flags (DO_LND_SITE_UPLINK, DO_TAC_SITE_UPLINK) is assumed to have been performed by some uplink interface software prior to execution of the processing identified in this section. The data availability flags are to be reset to OFF to prevent re-execution of this software.

E. Supplementary Information.- A suggested implementation, in the form of detailed flowcharts, can be found in Appendix F.

LANDING_SITE_MAXI_MINI_UPLINK

TACAN_SITE_MAXI_MINI_UPLINK

TABLE 4.9.6-1.- LANDING SITE UPLINK DATA FORMAT

Name	Description	Command Word	Data Type	Units **
LND_SITE_NO	Landing site table slot (1 to 18 - maxi, 19 to 21 - mini)	CW1	I	
RUNWAY_NAME_UL	Runway name identifier	CW2*	C	
SPARE_UL	Runway name identifier-plus fill	CW3*	C	-
RW_LAT_UL	Runway geodetic latitude	CW4	SF	rad
RW_LON_UL	Runway longitude	CW5	SF	rad
RW_AZIMUTH_UL	Runway azimuth measured from true north	CW6	SF	rad
RW_MAG_VAR_UL	Runway magnetic variation	CW7	SF	rad
RUNWAY_ALT_UL	Runway altitude above reference ellipsoid	CW8	SF	ft
RW_DELH_UL	Altitude of mean sea level above ellipsoid at runway	CW9	SF	ft

*CW2 contains a four character string word.
CW3 contains one additional character, left justified.

These are constructed to form a five string character word for storing (i.e., four characters in CW2 and the left most character in CW3 are combined).

```
             CW-2         CW-3
Example:   ! EDW1    !    7bbb    !
           !    stored   !
```

, where bbb are fill characters which are ignored.

**If the uplinked data words are in units other than those shown, onboard conversion to these units is required before storing into the maxi or mini tables.

TABLE 4.9.6-2.- TACAN SITE UPLINK DATA FORMAT

Uplink Buffer Data		Command Word	Data Type	Units[**]
Name	Description			
MAGNETIC_VARIATION_UL	Magnetic variation at TACAN site	CW1	SF	rad
BIAS_TAC_BRG_UL	TACAN bearing bias[*]	CW2	SF	rad
BIAS_TAC_R_UL	TACAN range bias[*]	CW3	SF	ft
TAC_SITE_NO	TACAN site table slot (1 to 50 - maxi, 51 to 60 - mini)	CW4	I	
TACAN_ID_UL	TACAN channel/mode identifier	CW5	I	
LATITUDE_GEODETIC_UL	TACAN site geodetic latitude	CW6	SF	rad
LONGITUDE_EAST_UL	TACAN site longitude	CW7	SF	rad
ALT_ABOVE_ELLIPSOID_UL	TACAN altitude above reference ellipsoid	CW8	SF	ft
MSL_ABOVE_ELLIPSOID_UL	Altitude of mean sea level above ellipsoid at TACAN site	CW9	SF	ft

[*] Biases are stored only in mini tables; thus CW-2 and CW-3 are ignored if maxi table is addressed.

[**] If the uplink data words are in units other than those shown, onboard conversion to these units is required before storing into the maxi or mini tables.

TABLE 4.9.6-3.- LANDING/TACAN SITE TABLE UPLINK INPUT

Symbol	Input Source
DO_LND_SITE_UPLINK	GROUND UPLINK PROCESSER
LND_SITE_NO	UPLINK
RUNWAY_NAME_UL	UPLINK
SPARE_UL	UPLINK
RW_LAT_UL	UPLINK
RW_LON_UL	UPLINK
RW_AZIMUTH_UL	UPLINK
RW_MAG_VAR_UL	UPLINK
RUNWAY_ALT_UL	UPLINK
RW_DELH_UL	UPLINK
DO_TACAN_SITE_UPDATE	GROUND UPLINK PROCESSOR
TAC_SITE_NO	UPLINK
TACAN_ID_UL	UPLINK
LATITUDE_GEODETIC_UL	UPLINK
LONGITUDE_EAST_UL	UPLINK
ALT_ABOVE_ELLIPSOID_UL	UPLINK
MSL_ABOVE_ELLIPSOID_UL	UPLINK
MAGNETIC_VARIATION_UL	UPLINK
BIAS_TAC_BRG_UL	UPLINK
BIAS_TAC_R_UL	UPLINK

TABLE 4.9.6-4.- LANDING/TACAN SITE TABLE UPLINK OUTPUT

Symbol	Output Destination
DO_LND_SITE_ UPLINK	LANDING SITE TABLE SPECIAL GROUND UPLINK PROCESSING SOFTWARE
RW_LAT PSL	*
RW_LON_PSL	*
RW_AZIMUTH_PSL	*
RUNWAY_ALT_PSL	*
RW_DELH_PSL	*
RW_MAG_VAR_PSL	*
MLS_AVAIL_PSL	*
RUNWAY_NAME_PSL	*
RW_LAT_SSL	*
RW_LON_SSL	*
RW_AZIMUTH_SSL	*
RUNWAY_ALT_SSL	*
RW_DELH_SSL	*
RW_MAG_VAR_SSL	*
MLS_AVAIL_SSL	*
RUNWAY_NAME_SSL	*
RW_LAT_ASL	*
RW_LON_ASL	*

*Update I-Load parameters in Landing Site Update principal function

C-5

TABLE 4.9.6-4.- LANDING/TACAN SITE TABLE UPLINK OUTPUT.- Continued

Symbol	Output Destination
RW_AZIMUTH_ASL	*
RUNWAY_ALT_ASL	*
RW_DELH_ASL	*
RW_MAG_VAR_ASL	*
RUNWAY_NAME_ASL	*
RUNWAY_NAME	*
RW_LAT	*
RW_LON	*
RW_AZIMUTH	*
RUNWAY_ALT	*
RW_DELH	*
RW_MAG_VAR	*
MLS_AVAIL	*
DO_TACAN_SITE_UPLINK	TACAN SITE TABLE UPLINK SPECIAL GROUND UPLINK PROCESSING SOFTWARE
MINI - TABLE	
LATITUDE_GEODETIC	*
LONGITUDE_EAST	*
ALT_ABOVE_ELLIPSOID	*

*Update I-Loaded parameters in Landing Site Update principal function

TABLE 4.9.6-4.- LANDING/TACAN SITE TABLE UPLINK OUTPUT.- Concluded

Symbol	Output Destination
MSL_ABOVE_ELLIPSOID	*
MAGNETIC_VARIATION	*
TAC_ID	*
BIAS_TAC_BRG	*
BIAS_TAC_R	*
MAXI-TABLE TACAN_ID_MAXI	*
LATITUDE_GEODETIC_MAXI	*
LONGITUDE_EAST_MAXI	*
ALT_ABOVE_ELLIPSOID_MAXI	*
MSL_ABOVE_ELLIPSOID_MAXI	*
MAGNETIC_VARIATION_MAXI	*

---

*Update I-Loaded parameters in Landing Site Update principal function

4.9.7 Landing Site Selection Parameters

This subsection provides the requirements associated with the landing site selection uplink which is contained within the Deorbit Guidance PEG-4 and the Deorbit Guidance PEG-7 command uplink loads.

A. Detailed Requirements. The Deorbit Guidance PEG-4 and PEG-7 command uplinks each contain three integer parameters which represent the maxi table slot locations from which landing site data is to be transferred to the landing site mini table primary, secondary and alternate runway slots. The uplink of these parameters is to cause the activation of the Landing Site Update principal function (see section 4.6) which in turn will accomplish the data transfer. Since the Landing Site Update principal function is designed to also respond to crew inputs via the CRT keyboard, certain special processing is required to provide compatibility with these requirements.

In response to a deorbit guidance uplink, as indicated by the flag, DO_GUID_UPLINK, having been set to ON by the ground uplink processing software, the following is to occur. The uplinked parameters PRI_MAXI_UL, SEC_MAXI_UL, and ALT_MAXI_UL (see tables 4.9.7-1 and 4.9.7-2) are integer values which specify which of the maxi table runways are to be transferred to the respective mini table locations. Each of these parameters is to be tested and the corresponding Landing Site Update input parameters set as follows:

```
If PRI_MAXI_UL > 0 and PRI_MAXI_UL < 19, then
set ITEM1_IN = ON
    PRI_MAXI_SELECT = PRI_MAXI_UL
If SEC_MAXI_UL > 0 and SEC_MAXI_UL < 19, then
set ITEM2_IN = ON
    SEC_MAXI_SELECT = SEC_MAXI_UL
If     ALT_MAXI_UL > 0 and ALT_MAXI_UL   < 19, then
set    ITEM3_IN = ON
       ALT_MAXI_SELECT = ALT_MAXI_UL
```

Thus, if it is desired to uplink a deorbit guidance target data set without changing part (or all) of the mini table slots, the respective maxi/mini indicators are uplinked with values outside the 1 to 18 range.

If any of the "ITEM_IN" discretes has been set to ON by the above processing, the Landing Site Update specialist function software is invoked to accomplish the data transfer. Note that the "ITEM_IN" discretes are reset to OFF by the Landing Site Update software after completing the data transfer.

Finally the flag DO_GUID_UPLINK is reset to OFF to indicate that the appropriate processing has occurred.

B. Interface Requirements. The input and output requirements for this subfunction are listed in tables 4.9.7-3 and 4.9.7-4, respectively.

C. Processing Requirements. To be executed on demand and provided with the required input data.

D. Constraints. The task of setting up the uplink buffer data sets and the uplink data availability flag is assumed to have been performed by uplink interface software prior to the execution of the processing identified in this section.

E. Supplemental Information. A suggested implementation, in the form of a detailed flowchart can be found in Appendix E,

SITE_SELECTION_UPLINK

TABLE 4.9.7-1.- LANDING SITE SELECTION UPLINK FORMAT (PEG 4 LOAD)

Name	Description	Command Word	Data Type	Units
PRI_MAXI_UL	Runway selection index for maxi table to primary runway data transfer	CW-13 (Bits 17-32)	I	--
SEC_MAXI_UL	Runway selection index for maxi table to secondary runway data transfer	CW-13 (Bits 33-48)	I	--
ALT_MAXI_UL	Runway selection index for maxi table to alternate runway data transfer	CW-14 (Bits 17-32)	I	--

TABLE 4.9.7-2.- LANDING SITE SELECTION UPLINK FORMAT (PEG 7 LOAD)

Name	Description	Command Word	Data Type	Units
PRI_MAXI_UL	Runway selection index for maxi table to primary runway data transfer	CW-11 (Bits 33-48)	I	--
SEC_MAXI_UL	Runway selection index for maxi table to secondary runway data transfer	CW-12 (Bits 17-32)	I	--
ALT_MAXI_UL	Runway selection index for maxi table to alternate runway data transfer	CW-12 (Bits 33-48)	I	--

TABLE 4.9.7-3.- LANDING SITE SELECTION UPLINK PROCESSING INPUT

Symbol	Input Source
PRI_MAXI_UL	UPLINK
SEC_MAXI_UL	UPLINK
ALT_MAXI_UL	UPLINK
DO_GUID_UPLINK	GROUND UPLINK PROCESSOR

TABLE 4.9.7-4.- LANDING SITE SELECTION UPLINK PROCESSING OUTPUT

Symbol	Output Destination
PRI_MAXI_SELECT	Landing Site Update (4.246)
SEC_MAXI_SELECT	Landing Site Update (4.246)
ALT_MAXI_SELECT	Landing Site Update (4.246)
ITEM1_IN	Landing Site Update (4.246)
ITEM2_IN	Landing Site Update (4.246)
ITEM3_IN	Landing Site Update (4.246)
DO_GUID_UPLINK	Ground Uplink Processor (4.246)

4.10 COORDINATE TRANSFORMATIONS

These ten subfunctions provide the capability for transforming parameters specified (or computed) in one coordinate system to another coordinate system. Coordinate system definitions are provided in Appendix C.

Each of the ten subfunctions described in sections 4.10.1 through 4.10.10 may be executed separately. With two exceptions, these subfunctions do not actually perform the coordinate transformation; only the rotation matrix is computed. The two exceptions convert between Earth-fixed coordinates and geodetic parameters.

For consistency, it is assumed that all coordinate systems located on the surface of the Earth are specified in terms of geodetic parameters (i.e., geodetic latitude, longitude, and altitude above the reference ellipsoid) and that all azimuth angles are referenced to true (not magnetic) north.

4.10.1 Transformation From Aries Mean of 1950 to Earth-Fixed

The purpose of this subfunction is to provide a transformation matrix (M_M50TOEF AT_EPOCH) from Aries M50 coordinates to Earth-fixed coordinates that account for the rotation, nutation, and precession motion of the Earth at a specified time, T_EPOCH.

This subfunction is not part of the actual onorbit Navigation Software. Its output parameters, M_M50TOEF_AT_EPOCH and their corresponding time, T_EPOCH, are to be supplied to the onorbit Navigation Software via I-Load as Mission-Dependent Parameters (see section 4.7 and Appendix C Variable List).

The software will not be designed to preclude any particular value of T_EPOCH, whether future or past except that, if the time desired is in a year other than the one of the mission's commencement, the general time variable is to be continuous over year-end/year-beginning transitions.

4.).2 Earth-Fixed to M50 (EARTH_FIXED_TO_M50_COORD)

The purpose of this subfunction is to propagate the matrix M_M50TOEF_AT_EPOCH from the last epoch time to the time of interest. This propagation accounts for the Earth's daily rotation effects only. (The nutation and precession computations are lengthy.)

The Earth-fixed coordinate frame of the time of interest is rotated about its Z-axis to obtain the Earth-fixed coordinate frame of the time of the M_M50TOEF_AT_EPOCH matrix. The angle of rotation is the product of the Earth's mean rotational rate relative to the mean of data system and the difference between the matrix's time tag and the time of interest. A matrix that represents this Z-axis rotation is calculated. The matrix that represents the transformation from the Earth-fixed frame of the last epoch time to the mean of 1950 frame is the M_M50TOEF_AT EPOCH matrix's transpose. Then the desired Earth-fixed to mean of 1950 matrix is the product of the transpose of the M_M50TOEF AT EPOCH matrix postmultiplied by the Z-axis rotation matrix.

A. Detailed Requirements. This function is designated in calling routines as

EARTH_FIXED_TO_M50_COORD(TIME)

where TIME -- time of interest measured from the beginning of the particular mission year

EARTH_FIXED_TO_M50_COORD -- Earth fixed to mean of 1950 coordinate transformation matrix of the time of interest

The following steps shall be performed (in the order indicated):

1. The time difference from the epoch time to the current time is calculated:

   DELT = TIME - T_EPOCH

   (T_EPOCH is the epoch time).

2. The angle of rotation about the Earth-fixed Z-axis is obtained.

   LAM = EARTH_RATE DELT

3. The Earth-fixed Z-axis rotation matrix is then defined:

$$M = \begin{pmatrix} \cos(\mathrm{LAM}) & -\sin(\mathrm{LAM}) & 0 \\ \sin(\mathrm{LAM}) & \cos(\mathrm{LAM}) & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathrm{F9}$$

F9 This equation shall be protected against return value of sine or cosine with magnitude greater then unity (Reference 3.6-9).

4. The Earth-fixed to mean of 1950 coordinate transformation matrix is then computed as follows:

$$\mathrm{EARTH\_FIXED\_TO\_M50\_COORD} = \mathrm{M\_M50TOEF\_AT\_EPOCH}^{T} \; \mathrm{M}$$

Because of the sparseness of the matrix M, both core and execution time will be conserved by use of the following formulation: For notational convenience, let N denote the matrix M_M50TOEF_AT_EPOCH. The sine and cosine of LAM will be computed once and denoted by CLAM = COS (LAM) and SLAM = SIN (LAM). Then EARTH_FIXED_TO_M50_COORD will be computed as follows:

$$\mathrm{EARTH\_FIXED\_TO\_M50\_COORD}_{1,1} = N_{1,1}\ \mathrm{CLAM} + N_{2,1}\ \mathrm{SLAM}$$

$$\mathrm{EARTH\_FIXED\_TO\_M50\_COORD}_{1,2} = -N_{1,1}\ \mathrm{SLAM} + N_{2,1}\ \mathrm{CLAM}$$

$$\mathrm{EARTH\_FIXED\_TO\_M50\_COORD}_{1,3} = N_{3,1}$$

$$\mathrm{EARTH\_FIXED\_TO\_M50\_COORD}_{2,1} = N_{1,2}\ \mathrm{CLAM} + N_{2,2}\ \mathrm{SLAM}$$

$$\mathrm{EARTH\_FIXED\_TO\_M50\_COORD}_{2,2} = -N_{1,2}\ \mathrm{SLAM} + N_{2,2}\ \mathrm{CLAM}$$

$$\mathrm{EARTH\_FIXED\_TO\_M50\_COORD}_{2,3} = N_{3,2}$$

$$\mathrm{EARTH\_FIXED\_TO\_M50\_COORD}_{3,1} = N_{1,3}\ \mathrm{CLAM} + N_{2,3}\ \mathrm{SLAM}$$

$$\mathrm{EARTH\_FIXED\_TO\_M50\_COORD}_{3,2} = -N_{1,3}\ \mathrm{SLAM} + N_{2,3}\ \mathrm{CLAM}$$

$$\mathrm{EARTH\_FIXED\_TO\_M50\_COORD}_{3,3} = N_{3,3}$$

B. Interface Requirements. The input and output data are shown in table 4.10.2.

C. Processing Requirements. This function may be executed as needed. The time (TIME) at which the output transformation matrix (EARTH_FIXED_TO_M50_COORD) is desired must be supplied by the user in elapsed seconds from the beginning of the year of mission commencement.

A valid M_M50TOEF_AT_EPOCH must be available before this subfunction can be executed.

D. Constraints. None

E. Supplementary Information. The Aries mean of 1950 and the Earth-fixed coordinate systems are shown in Appendix C. A suggested implementation in the form of a detailed flowchart may be found in Appendix C under the name:

EARTH_FIXED_TO_M50_COORD FUNCTION

TABLE 4.10.2.- EARTH_FIXED_TO_M50_COORD FUNCTION INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
TIME	T	ACCEL_ONORBIT FUNCTION	
TIME	T_STATE	ONORBIT_USER_PARAMETER_CALCULATIONS	

TABLE 4.10.2.- EARTH_FIXED_TO_M50_COORD FUNCTION
INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
EARTH_RATE	**	
M_M50TOEF_AT_EPOCH	**	
T_EPOCH	**	
†		ACCEL_ONORBIT FUNCTION, ONORBIT_USER_PARAMETER_CALCULATIONS

---

†Only the values of EARTH_FIXED_TO_M50_COORD's elements are passed to output destination.
**Initialization parameters, see section 4.7

4.10.3 Geodetic to Earth-Fixed (GEODETIC_TO_EF)

This submodule accepts the geodetic parameters of a point and computes the Earth-fixed Cartesian coordinates of that point.

A. Detailed Requirements. This function is addressed in calling modules as

GEODETIC_TO_EF(LAT_GEOD,LON,ALT)

This function is referred to internally as

GEODETIC_TO_EF(LAT_GEOD,LON,ALT) = R _EF

where ALT -- altitude above reference ellipsoid of point of interest
LAT_GEOD -- geodetic latitude of point of interest
LON -- longitude of point of interest
R _EF -- Earth-fixed position vector of point of interest

These equations shall transform the geodetic parameters to Earth-fixed coordinates:

$$R\_EF_1 = \left[\frac{\text{EARTH\_RADIUS\_EQUATOR}}{(\cos^2(\text{LAT\_GEOD}) + (1 - \text{ELLIPT})^2 \sin^2(\text{LAT\_GEOD}))^{1/2}} + \text{ALT}\right] \cdot \cos(\text{LAT\_GEOD}) \cos(\text{LON})$$

F3, F9, F4

$$R\_EF_2 = \left[\frac{\text{EARTH\_RADIUS\_EQUATOR}}{(\cos^2(\text{LAT\_GEOD}) + (1 - \text{ELLIPT})^2 \sin^2(\text{LAT\_GEOD}))^{1/2}} + \text{ALT}\right] \cdot \cos(\text{LAT\_GEOD}) \sin(\text{LON})$$

F3, F9, F4

$$R\_EF_3 = \left[\frac{(1 - \text{ELLIPT})^2\text{EARTH\_RADIUS\_EQUATOR}}{(\cos^2(\text{LAT\_GEOD}) + (1 - \text{ELLIPT})^2 \sin^2(\text{LAT\_GEOD}))^{1/2}} + \text{ALT}\right] \cdot \sin(\text{LAT\_GEOD})$$

F3, F9, F4

F3 This equation shall be protected against division by zero (Reference 3.6-3).

F9 This equation shall be protected against return value of sine or cosine with magnitude greater than unity (Reference 3.6-9).

F4 This equation shall be protected against square roots of a negative number (Reference 3.6-4).

B. Interface Requirements. The input and output data are shown in table 4.10.3.

C. Processing Requirements. The input angles must be in radians.

D. Constraints. None

E. Supplementary Information. The geodetic parameter set and the Earth-fixed coordinate system are illustrated in Appendix C. A suggested implementation in the form of a detailed flowchart may be found in Appendix C under the name:

GEODETIC_TO_EF FUNCTION

TABLE 4.10.3.- GEODETIC_TO_EF FUNCTION INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
ALT		Various Users	
LAT_GEOD		Various Users	
LON		Various Users	
$\underline{R}$_EF	†		Various Users

†Only the values of $\underline{R}$_EF's components are passed to output destination.

TABLE 4.10.3.- GEODETIC_TO_EF FUNCTION INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
EARTH_RADIUS_EQUATOR	**	
ELLIPT	**	

**Initialization parameters, see section 4.7

4.10.4 Earth-Fixed to Topodetic (EF_TO_TOPDET)

This subfunction accepts the geodetic latitude and longtitude of a point in radians and computes the rotation matrix from Earth-fixed coordinates to a topodetic coordinate system for the input location.

Formulation: This subfunction creates the rotation matrix as an Euler Z, Y sequence through the longitude angle (LON) and the geodetic latitude angle plus 90 degrees (LAT_GEOD + $\pi/2$), respectively.

A. Detailed Requirements. Calling modules address this function as

EF_TO_TOPDET(LAT_GEOD, LON)

This function is referred to internally as

EF_TO_TOPDET(LAT_GEOD, LON) = M

where
LAT_GEOD -- geodetic latitude of point of interest
LON -- longitude of point of interest
M -- desired Earth-fixed to topodetic coordinate transformation matrix

The transformation matrix shall be calculated as shown here:

$$M = \begin{bmatrix} (-\cos \text{LON} \sin \text{LAT\_GEOD}) & (-\sin \text{LON} \sin \text{LAT\_GEOD}) & (\cos \text{LAT\_GEOD}) \\ (-\sin \text{LON}) & (\cos \text{LON}) & 0 \\ (-\cos \text{LON} \cos \text{LAT\_GEOD}) & (-\sin \text{LON} \cos \text{LAT\_GEOD}) & (-\sin \text{LAT\_GEOD}) \end{bmatrix} \text{F9}$$

B. Interface Requirements. The input and output data are shown in table 4.10.4.

C. Processing Requirements. The input variables must be in radians.

D. Constraints. None

E. Supplementary Information. The Earth-fixed and topodetic coordinate systems, and a suggested implementation of this module are provided in Appendix C. A suggested implementation in the form of a detailed flowchart may be found in Appendix C under the name:

EF_TO_TOPDET FUNCTION

F9 This equation shall be protected against return value of sine or cosine with magnitude greater than unity (Reference 3.6-9).

TABLE 4.10.4.- EF_TO_TOPDET FUNCTION INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
LAT_GEOD	LAT_GEOD	EF_TO_RUNWAY FUNCTION	
LON	LON	EF_TO_RUNWAY FUNCTION	
M	†		EF_TO_RUNWAY FUNCTION

---

†Only the values of M's elements are passed to output destination.

4.10.5 Earth-Fixed to Runway (EF_TO_RUNWAY)

This subfunction accepts the geodetic latitude, longitude, and azimuth of the runway and computes the rotation matrix from the Earth-fixed coordinate system to the runway coordinate system.

Formulation: The Earth-fixed to topodetic subfunction is used to obtain an Earth-fixed to topodetic rotation matrix. Then the Earth-fixed to topodetic matrix is multiplied by an Euler Z rotation matrix through the runway azimuth angle (AZ), measured from true north to the +X-axis of the runway.

A. Detailed Requirements. Other routines designate this function as

EF_TO_RUNWAY(LAT_GEOD,LON,AZ)

in which LAT_GEOD -- runway's geodetic latitude
LON -- runway's longitude
AZ -- runway's azimuth
EF_TO_RUNWAY -- desired Earth-fixed to runway coordinate transformation matrix

The following steps shall be performed (in the order indicated):

1. The Euler Z rotation matrix is calculated.

$$M = \begin{bmatrix} (\cos AZ) & (\sin AZ) & 0 \\ (-\sin AZ) & (\cos AZ) & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{F9}$$

2. The Earth-Fixed to Topodetic Function is called and its rotation matrix is premultiplied by the Euler Z rotation matrix to produce the Earth-fixed to runway rotation matrix.

EF_TO_RUNWAY = M EF_TO_TOPDET(LAT_GEOD, LON)

B. Interface Requirements. The input and output data are shown in table 4.10.5.

C. Processing Requirements. Input angles must be in radians.

D. Constraints. None

F9 This equation shall be protected against return value of sine or cosine with magnitude greater than unity (Reference 3.6-9).

E. Supplementary Information. The Earth-fixed, runway, and topodetic coordinate systems are shown in Appendix C. A suggested implementation in the form of a detailed flowchart may be found in Appendix C under the name:

EF_TO_RUNWAY FUNCTION

TABLE 4.10.5.- EF_TO_RUNWAY FUNCTION INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
AZ	AZ	EF_TO_SCANNER FUNCTION	
LAT_GEOD	LAT_GEOD	EF_TO_SCANNER FUNCTION	
LON	LON	EF_TO_SCANNER FUNCTION	

TABLE 4.10.5.- EF_TO_RUNWAY FUNCTION INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
†	EF_TO_TOPDET(LAT_GEOD, LON)	
†		EF_TO_SCANNER FUNCTION

---

†Only EF_TO_TOPDET's element values are passed from input source, and only values of EF_TO_RUNWAY's elements are passed to output destination.

4.10.6 Earth-Fixed to Scanner (EF-TO-SCANNER)

The purpose of this subfunction is to compute the rotation matrix from Earth-fixed to scanner coordinates.

Formulation: The Earth-fixed to runway subfunction is executed, with use of the geodetic latitude and longitude of the scanner and the azimuth of the scanner boresight (AZ) from the true north to obtain a rotation matrix.

The rotation matrix is then multiplied by a rotation matrix representing a 180-degree rotation about the X-axis.

A. Detailed Requirements. This function is designated

EF_TO_SCANNER(LAT_GEOD,LON,AZ)

in calling routines.

where

LAT_GEOD -- geodetic latitude of scanner
LON -- longitude of scanner
AZ -- azimuth of scanner boresight
EF_TO_SCANNER -- desired Earth-fixed to scanner coordinate transformation matrix

This step shall be performed:

The evaluation of the Earth-Fixed to Runway Function is premultiplied by the 180 degree X-axis rotation matrix

$$M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

to obtain the Earth-fixed to scanner rotation matrix.

EF_TO_SCANNER = M EF_TO_RUNWAY(LAT_GEOD,LON.AZ)

B. Interface Requirements. The input and output data are given in table 4.10.6.

C. Processing Requirements. Input angles must be in radians.

D. Constraints. None

E. Supplementary Information. The scanner, runway, topodetic, and Earth-fixed coordinate systems are shown in Appendix C. A suggested implementation in the form of a detailed flowchart may be found in Appendix C under the name:

EF_TO_SCANNER FUNCTION

TABLE 4.10.6.- EF_TO_SCANNER FUNCTION INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
AZ		Various Users	
LAT_GEOD		Various Users	
LON		Various Users	

TABLE 4.10.6.- EF_TO_SCANNER FUNCTION INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
†	EF_TO_RUNWAY (LAT_GEOD,	
†	LON, AZ)	
		Various Users

---

†Only the values of EF_TO_RUNWAY's elements are passed from input source; and only the values of EF_TO_SCANNER's elements are passed to output destination.

4.10.7 Transformation From Structural Body to M50 Coordinates (SBODY_TO_M50)

The transformation matrix from structural body to M50 coordinates (MAT) is computed by postmultiplying the transformation matrix from body to M50 coordinates by a matrix representing a 180-degree rotation about the Y-axis.

The transformation from body to M50 coordinates is the transpose of the matrix derived from the quaternion Q_FIFTY_BODY by employing the special purpose matrix function QUAT_TO_MAT.

A. Detailed Requirements. This subfunction is called with the following internal variables in the IN LIST and the OUT LIST:

IN LIST: Q_FIFTY_BODY

OUT LIST: MAT

where

Q_FIFTY_BODY	mean of 1950 to body rotation quaternion
MAT	desired structural body to mean of 1950 coordinate transformation matrix

The following steps shall be performed (in the order indicated):

1. The special purpose matrix function QUAT_TO_MAT is called to obtain a mean of 1950 to body rotation matrix.

   MATRIX = QUAT_TO_MAT (Q_FIFTY_BODY)

2. Rotate the transpose of the output matrix 180° about the Y-axis.

$$
\begin{aligned}
MAT_{1,1} &= -MATRIX_{1,1} \\
MAT_{1,2} &= MATRIX_{2,1} \\
MAT_{1,3} &= -MATRIX_{3,1} \\
MAT_{2,1} &= -MATRIX_{1,2} \\
MAT_{2,2} &= MATRIX_{2,2} \\
MAT_{2,3} &= -MATRIX_{3,2} \\
MAT_{3,1} &= -MATRIX_{1,3} \\
MAT_{3,2} &= MATRIX_{2,3} \\
MAT_{3,3} &= -MATRIX_{3,3}
\end{aligned}
$$

B. Interface Requirements. Input and output parameters are listed in table 4.10.7.

C. Processing Requirements. This subfunction may be executed as needed.

D. Constraints. None

E. Supplementary Information. A suggested implementation in the form of a detailed flowchart can be found in Appendix C under the name:

SBODY_TO_M50

TABLE 4.10.7.- SBODY_TO_M50 INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
Q_FIFTY_BODY		Various Users	
MAT			Various Users

TABLE 4.10.7.- SBODY_TO_M50 INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
†	QUAT_TO_MAT (Q_FIFTY_ BODY)	

---

†Only the values of QUAT_TO_MAT's elements are passed from input source.

4.10.8 Conversion of a Quaternion to a Matrix (QUAT_TO_MAT)

The special purpose matrix function QUAT_TO_MAT computes the transformation matrix, A, associated with a quaternion, Q, as QUAT_TO_MAT(Q) = A.

A. **Detailed Requirements**. Calling modules designate this function

QUAT_TO_MAT(Q)

This function is referred to internally as

QUAT_TO_MAT(Q) = A

where Q - quaternion

A - corresponding transformation matrix

The following steps shall be performed (in the order indicated).

$P2 = Q_2 + Q_2$

$P3 = Q_3 + Q_3$

$P4 = Q_4 + Q_4$

$P5 = P2\ Q_2$

$P6 = P4\ Q_4$

$TEMP = 1.0 - P3\ Q_3$

$A_{1,1} = TEMP - P6$

$A_{2,2} = 1.0 - P5 - P6$

$A_{3,3} = TEMP - P5$

$P5 = P2\ Q_3$

$P6 = P4\ Q_1$

$A_{1,2} = P5 - P6$

$A_{2,1} = P5 + P6$

$P5 = P2\ Q_4$

$P6 = P3\ Q_1$

$A_{1,3} = P5 + P6$

$A_{3,1} = P5 - P6$

$P5 = P3\ Q_4$

$P6 = P2\ Q_1$

$A_{2,3} = P5 - P6$

$A_{3,2} = P5 + P6$

B. Interface Requirements. Input and output parameters are listed in table 4.10.8.

C. Processing Requirements. None

D. Constraints. None

E. Supplementary Information. A suggested implementation in the form of a detailed flowchart can be found in Appendix C under the name:

QUAT_TO_MAT FUNCTION

TABLE 4.10.8.- QUAT_TO_MAT FUNCTION INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
$\underline{Q}$	$\underline{Q}$_M50BODY_IMU	NAV_ONORBIT_RENDEZVOUS	
$\underline{Q}$	$\underline{Q}$_M50BODY_RR	RR_ANGLE_NAV	
$\underline{Q}$	$\underline{Q}$_FIFTY_BODY	SBODY_TO_M50	
A	†		NAV_ONORBIT_RENDEZVOUS, RR_ANGLE_NAV, SBODY_TO_M50

†Only values of A's components are passed to output destination.

4.10.9 Earth-Fixed to Geodetic (EF_TO_GEODETIC)

This function is required to transform a Cartesian position vector in the Earth-fixed (Greenwich) coordinate system to the geodetic parameters: geodetic altitude, longitude, and geodetic latitude.

A. Detailed Requirements. This subfunction is called with the following internal variables in the IN LIST and the OUT LIST:

IN LIST: $\underline{R}$_EF

OUT LIST: LAT_GEOD, LON, ALT

where

$\underline{R}$_EF	Cartesian position vector
LAT_GEOD	geodetic latitude
LON	geodetic longitude
ALT	geodetic altitude

The following steps shall be performed (in the order indicated):

1. The computation for longitude is:

$$LON = ATAN \left(\frac{R\_EF_2}{R\_EF_1} \right), \quad 0 \leq LON < 2\pi \qquad \text{F3,F7}$$

2. Computations for geodetic latitude, LAT_GEOD, and height above the reference ellipsoid, ALT, are as follows:

$$R\_XY = R\_EF_1^2 + R\_EF_2^2$$

$$R = (R\_XY + R\_EF_3^2)^{1/2} \qquad \text{F4}$$

$$FLATCON = 1.0 - (1.0-ELLIPT)^2, \text{ where } ELLIPT = \text{flattening of the reference ellipsoid}$$

F3 This equation shall be protected against division by zero (Reference 3.6-3).

F7 This equation shall be protected against arc tangents with both arguments equal to zero (Reference 3.6-7).

F4 This equation shall be protected against square roots of a negative number (Reference 3.6-4).

$$\text{SIN\_P} = \text{R\_EF}_3/\text{R} \qquad \text{F3}$$

$$\text{COS\_P} = (\text{R\_XY})^{1/2}/\text{R} \qquad \text{F3, F4}$$

$$\text{RAD\_P} = \frac{\text{EARTH\_RADIUS\_EQUATOR}}{\sqrt{1.0 + \text{FLATCON SIN P}^2/(1.0 - \text{FLATCON})}} \qquad \text{F3, F4}$$

$$\text{DEL} = \frac{\text{FLATCON SIN\_P COS\_P}}{1.0 - \text{FLATCON COS\_P}^2} \qquad \text{F3}$$

$$\text{DEL\_LAT} = \frac{\text{RAD\_P DEL}}{\text{R}} \qquad \text{F3}$$

$$\text{PHI} = \text{ATAN (SIN\_P/COS\_P)},\ -\pi/2 \le \text{PHI} \le \pi/2 \qquad \text{F3, F7}$$

$$\text{LAT\_GEOD} = \text{PHI} + \text{DEL\_LAT (in radians)}$$

and

$$\text{ALT} = (\text{R-RAD\_P})\ (1.0 - 0.50\ \text{DEL DEL\_LAT})\ \text{(in feet)}$$

B. Interface Requirements. The input and output data are shown in table 4.10.9.

C. Processing Requirements. This subfunction may be executed on demand.

D. Constraints. None

E. Supplementary Information. The Earth-fixed coordinate system and geodetic parameters are defined in Appendix C. A suggested implementation in the form of a detailed flowchart may be found in Appendix C under the name:

EF_TO_GEODETIC

F3 This equation shall be protected against division by zero (Reference 3.6-3).

F4 This equation shall be protected against square roots of a negative number (Reference 3.6-4).

F7 This equation shall be protected against arc tangents with both arguments equal to zero (Reference 3.6-7).

TABLE 4.10.9.- EF_TO_GEODETIC INPUT/OUTPUT

Inlist/Outlist			
Internal Name	External Name	Input Source	Output Destination
R_EF		Various Users	
ALT			Various Users
LAT_GEOD			Various Users
LON			Various Users

TABLE 4.10.9.- EF_TO_GEODETIC INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
EARTH_RADIUS_EQUATOR	**	
ELLIPT	**	

**Initialization parameters, see section 4.7

4.10.10 UVW to M50 (UVW_TO_M50)

The purpose of this subfunction is to compute the transformation matrix from UVW coordinates to Aries M50 coordinates, given the position and velocity of the vehicle.

The orientation of the U, V, W system is determined by the Orbiter inertial position and velocity vectors ($\underline{R}$, $\underline{V}$) at the point (or time) of interest.

A. Detailed Requirements. Users of this function designate it:

UVW_TO_M50($\underline{R}$,$\underline{V}$)

where $\underline{R}$ -- position vector (mean of 1950)
$\underline{V}$ -- velocity vector (mean of 1950)
$\overline{\text{UVW\_TO\_M50}}$ -- UVW to mean of 1950 coordinate transformation matrix

The following steps shall be performed (in the order indicated):

1. The UVW coordinate frame axes unit vectors in mean of 1950 coordinates are determined. ($\underline{Z}$ represents the V-axis unit vector to avoid confusion with the velocity vector.)

$$\underline{U} = \underline{R}/|\underline{R}|$$ [F3]

$$\underline{W} = (\underline{R} \times \underline{V})/|\underline{R} \times \underline{V}|$$ [F3]

$$\underline{Z} = \underline{W} \times \underline{U}$$

2. The transformation matrix from UVW coordinates to Aries mean of 1950 is then given by

$$\text{UVW\_TO\_M50} = \underline{U}|\underline{Z}|\underline{W}$$

B. Interface Requirements. The input and output data are given in table 4.10.10.

C. Processing Requirements. All computations are to be performed in double precision.

D. Constraints. None

E. Supplementary Information. The UVW coordinate system and the Aries M50 system are defined in Appendix C. A suggested implementation in the form of a detailed flowchart may be found in Appendix C under the name:

UVW_TO_M50 FUNCTION

[F3] This equation shall be protected against division by zero (Reference 3.6-3).

TABLE 4.10.10.- UVW_TO_M50 FUNCTION INPUT/OUTPUT

Inlist/Outlist		Input Source	Output Destination
Internal Name	External Name		
R	R _FILT	REND_BIAS_AND_COV_PROP	
R	R _TV	REND_BIAS_AND_COV_PROP	
R	R	U_A_BIAS_AND_COVINIT	
V	V _FILT	REND_BIAS_AND_COV_PROP	
V	V _TV	REND_BIAS_AND_COV_PROP	
V	V	U_A_BIAS_AND_COVINIT	

TABLE 4.10.10.- UVW_TO_M50 FUNCTION INPUT/OUTPUT.- Concluded

Variable Name	Input Source	Output Destination
†		REND_BIAS_AND_COV_PROP, U_A_BIAS_AND_COVINIT

---

†Only the values of UVW_TO_M50's components are passed to output destination.

APPENDIX A

GENERAL REQUIREMENTS FUNCTIONS

CONTENTS

Subject | Page

PRECEDING PAGE BLANK NOT FILMED

VARIABLES LIST DEFINITIONS

CODE USED FOR VARIABLE DATA TYPE

F: floating point quantity. An n-dimensional floating point vector will be denoted F(n). Similarly, a nxm floating point matrix will be denoted by F(n,m).

I: integer quantity; I(n) will denote an n-dimensional integer vector

B: bit, i.e., data having only values of 0 or 1

C: character; C(n) will denote an n-dimensional character string

CODE USED FOR VARIABLE PRECISION

D: double precision

S: single precision; integer quantities are assumed single precision unless otherwise specified

VARIABLE LOCATION

Compool: Variable value located in common storage, accessible by all functions

Local : Variable is used by one function only, and usable to other functions through call argument only

VARIABLE INITIALIZATION CATEGORY

blank: display is vacant

C: constant (unchanging)

DD: design dependent

HC: hard coded

MD: mission dependent (I-LOAD)

OPS: OPS transition parameter

IV: other required initial values

VARIABLE INITIAL VALUE

initial operation sequence computer inputs

VARIABLE UPLINK AND DOWNLIST STATUS

UPLINK:	variable is an uplink item
DOWNLIST:	variable is a downlist item

UNITS DEFINITIONS

deg:	angular measurement, degrees
ft:	feet
lb:	pounds
n.d.:	non-dimensional
rad:	radian
sec:	time measurement, seconds
slugs:	mass measurement, slugs
vary:	units have different values which depend on variable use

VARIABLES LIST

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
A	DF(5,2)	local	--	—	—	--	Legendre functions array in gravitational acceleration calculation.
ABM (V96U9061C-66C)	SF(3,2)	local	MD	—	—	n.d. 1/ft ft	Variable used in ACCEL_ONORBIT atmospheric density mode.
ACC_MIN (V96U9067C)	I	local	DD	2	—	--	Minimum number of sequential marks accepted by the residual edit test to blank the display edit status indicator (DISP_EDIT).
AFH	SF	local	—	--	—	--	Density model night time vertical coefficient.
ALT_SS (V90H4945C)	SF	compool	—	—	downlist	ft	Altitude of the Shuttle for downlist.
ALT	SF	local	--	—	--	ft	Altitude above reference ellipsoid used in ACCEL_ONORBIT.
ALT_L (V96U9068C)	SF	local	MD	—	--	ft	Density model upper bound of lower layer used in ACCEL_ONORBIT drag model.
ANGLES_AIF (V93J6237C)	C	compool	--	—	—	--	Auto/Inhibit/Force switch associated with the currently enabled angles data set.
ANGLES_AIF_DISPLAY (V90J1361C)	C	compool	—	—	downlist	—	Flag indicating to display that the angles auto/inhibit/force flag (ANGLES_AIF) has been acknowledged by navigation.
ANGLES_AIF_LAST	C	local	IV	inhibit	—	--	The last nonforce value of the ANGLES_AIF flag.
ANGLES_ENABLE_DISPLAY (V90J1416C)	I	compool	--	—	downlist	--	Positive feedback flag to the display indicating the angle set currently enabled.

VARIABLES LIST

Variable name (M/S ID)	Preci- sion & type	Compool or local	Initial- ization category	Initial value	Uplink/ downlist	Units	Description
ANGLES_MANUAL_EDIT_ OVERRIDE	B	local	--	--	--	--	Local value of the manual edit override flag.
ANGLES_STAT_FLAG	B	local	--	--	--	--	Local value of the stat flag.
AREA	SF	local	--	--	--	ft^2	Area used in ACCEL_ONORBIT drag model.
$\underline{A}$_RESID	DF(3)	local	--	--	--	ft/sec^2	Acceleration of the Orbiter interpolated to a specified measurement time (M50).
ATFL_OV	I	compool	HC	0	--	--	Flag controlling use or nonuse of current attitude of Orbiter.
ATFL_TV	I	compool	HC	2	--	--	Flag controlling use of mass, area, and drag coefficient of target vehicle.
ATM	I	local	--	--	--	--	Flag indicating if current or prestored configuration constants are to be used in drag computations within ACCEL_ ONORBIT.
ATMP (V94J3999C)	I	compool	--	--	downlist	--	Flag indicating target or Orbiter CD, mass and area for prediction.
$\underline{A}$_TV_RESID	DF(3)	local	--	--	--	ft/sec^2	Acceleration of the target vehicle interpolated to a specified measurement time (M50).
AUXILIARY	DP	local	--	--	--	$1/sec^2$	Scalar variable used in gravitational acceleration model in ACCEL_ONORBIT.
A1 A2 A3 A4 A5 A6 A7 A8 A9	DP	local	--	--	--	ft^{-2} sec/ft^2 sec/ft^2 sec^2/ft^2 sec^2/ft^2 sec^3/ft^2 $sec^{-1} ft^{-2}$ ft^{-2} sec/ft^2	Temporary variables in mean conic partials transition matrix computations.

VARIABLES LIST

Variable name (M/S ID)	Preci- sion & type	Compool or local	Initial- ization category	Initial value	Uplink/ downlist	Units	Description
B	DP(13)	compool	—	—	—	vary	Partial derivative of the measurement with respect to the filter vehicle state.
BFH	SP	local	—	—	—	--	Atmospheric density model diurnal coefficient used in ACCEL_ONORBIT.
BIAS_COV_VAR	SP(2)	local	—	—	—	vary	Copy of the bias variance terms for the covariance matrix of the sensor data currently being processed.
BIAS_INIT	SP(2)	local	—	—	—	vary	Copy of the initial sensor bias of the sensor data currently being processed.
BIAS_VAR	SP(2)	local	—	—	—	vary	Copy of the sensor bias variances for the sensor data currently being processed.
BIAS_VAR_COAS_ANGLES (V96U9069C-70C)	SP(2)	local	DD	5.E-6 5.E-6	—	rad^2	Variance of the COAS angle biases.
BIAS_VAR_RR_ANGLES (V96U9071C-2C)	SP(2)	local	DD	3.0E-4 3.0E-4	—	rad^2	Variance of the rendezvous radar angle biases.
BIAS_VAR_RRDOT (V96U9073C-4C)	SP(2)	local	DD	711.0 1.0	—	ft^2, ft^2/sec^2	Variance of the rendezvous radar range and range rate biases.
BIAS_VAR_ST_ANGLES (V96U9075C-6C)	SP(2)	local	DD	1.2E-6 1.2E-6	—	rad^2	Variance of the star tracker angle biases.
BM (V96U9077C-9C)	SP(3)	local	MD	—	—	nd 1/ft ft	Atmospheric density model diurnal coefficients used in ACCEL_ONORBIT.
BT_E_B	DP	local	—	—	—	vary	Variable used to store the dot product of B and EB
C	DP(9)	local	C	—	—	--	Tesseral harmonics coefficients used in ACCEL_ONORBIT Earth gravity code.

VARIABLES LIST

Variable name (M/S ID)	Preci-sion & type	Compool or local	Initial-ization category	Initial value	Uplink/ downlist	Units	Description
CBD1 (V96U9087C)	SF	local	DD	1.4036E-5	—	1/ft	Coefficient used in ACCEL_ONORBIT atmospheric density seasonal latitudinal effect.
CBD2 (V96U9088C)	SF	local	DD	-1.3716E -5	—	1/ft	Coefficient used in ACCEL_ONORBIT atmospheric density seasonal latitudinal effect.
CBM1	SF	local	--	—	—	--	Coefficient used in ACCEL_ONORBIT atmospheric density seasonal latitudinal effect.
CBM2 (V96U9089C)	SF	local	MD	--	—	--	Coefficient used in ACCEL_ONORBIT atmospheric density seasonal latitudinal effect.
CD	SF	local	--	—	—	--	Vehicle drag coefficient used in ACCEL_ONORBIT drag acceleration calculations.
CDA (V97U5281C)	SF	local	DD	0.64592	—	--	Constant used to model drag coefficient used in ACCEL_ONORBIT.
CDEC1	DF	compool	--	—	—	--	Cosine of solar declination used in ACCEL_ONORBIT density calculations.
C_DENSEA (V96U9094C)	SF	local	DD	295274.7	—	ft	Constant used in ACCEL_ONORBIT density model.
CDF (V97U6002C)	SF	local	DD	0.78590	—	--	Constant used in ACCEL_ONORBIT drag coefficient calculation.
CDN (V97U5282C)	SF	local	DD	2.41256	—	--	Constant used in ACCEL_ONORBIT drag coefficient calculation.
CDS (V97U6010C)	SF	local	DD	1.92837	—	--	Constant used in ACCEL_ONORBIT drag coefficient calculation.
COAS_ANGLES_BIAS_INIT (V96U9084C-5C)	SF(2)	local	MD	—	—	rad	Initial value of the COAS angle bias slots in the state vector.
COAS_ANGLES_EDIT_OVERRIDE	B	local	--	—	—	--	Flag used to override (ON) the residual edit test for the COAS angles data.

VARIABLES LIST

Variable name (M/S ID)	Presi- sion & type	Compool or local	Initial- ization category	Initial value	Uplink/ downlist	Units	Description
COAS_ANGLES_STAT	B	local	—	—	—	--	Flag indicating that COAS angles data are to be processed for statistical display purpose only (OM).
COAS_DATA_GOOD (V̄90X4849X)	B	compool	—	—	downlist	--	Snapped COAS flag indicating the use of COAS to track a target. This is used by navigation as a DATA_GOOD flag. (STAR TRCK SOP HAL name NAV_SIGHT,M/SID # V̄95X4779X).
COAS_ENABLE (V̄93X6225X)	B	compool	—	—	downlist	--	COAS angle ENABLE flag not used in the software but is included here for downlist notification.
COAS_ID (V̄90J4854C)	I	compool	—	—	downlist	--	Snapped COAS selection flag (STAR TRCK SOP HAL name AXM, M/S ID# V̄95X4972X).
COAS_MARK_NUM (V̄90J4827C)	I	compool	IV	0	downlist	--	COAS measurement mark counter.
CONST	DF	local	--	—	—	sec^3/ft^2	Temporary variable used in transition matrix computation.
CONT_ACC (V̄90A4944C)	SP	compool	--	—	downlist	ft/sec^2	Contact acceleration (magnitude of the IMU sensed acceleration) used to determine whether IMU data and/or measurement data are to be used.
COS_LAG (V̄96U9086C)	SP	local	DD	0.798 63551	—	--	Cosine of atmospheric bulge lag angle used in ACCEL_ONORBIT density model.
COS_PSI	SF	local	—	—	—	--	Variable used in ACCEL_ONORBIT atmospheric density model.
COS_SOL_RA	DF	compool	—	—	—	--	Cosine of solar right ascension used in ACCEL_ONORBIT density model.
COV_ACCEL_UVW_INIT (V̄90H4930C-32C)	SF(3)	compool	--	—	—	ft^2/sec^4	Vector of unmodeled acceleration bias error variances in UVW coordinates.
COV_COR_UPDATE (V̄96H1290C-6C)	SF(7)	compool	MD	—	uplink	--	Vector of correlation coefficients associated with UVW standard deviations (SIG_UPDATE) used for Orbiter/target covariance initialization (ground update).

VARIABLES LIST

Variable name (M/S ID)	Preci- sion & type	Compool or local	Initial- ization category	Initial value	Uplink/ downlist	Units	Description
COV_PWRD_FLT (V90X4936X)	B	compool	—	—	—	—	Flag indicating (ON) that the IMU use flag has been turned on at least once since the last COV prop.
COV_PWRD_FLT_LAST (V90X4845X)	B	compool	—	—	—	--	The last value of COV_PWRD_FLT.
COV_U_A_COAST (V96U9095C-7C)	SF(3)	compool	DD	3* 1.E-8	—	ft^2/sec^4	Unmodeled acceleration bias variances for coasting flight.
COV_U_A_PWRD_FLT (V96U9098C-100C)	SF(3)	local	DD	3*5.36E-2	—	ft^2/sec^4	Unmodeled acceleration bias variances for powered flight.
CSPST	SF	local	—	—	—	ft	Variable used in ACCEL_ONORBIT atmospheric density model.
CSSMD	SF	local	—	—	—	ft	Variable used in ACCEL_ONORBIT atmospheric density model.
CURR_ORB_MASS (V90U1961C)	SF	compool	—	—	—	slugs	Current Orbiter mass
C1	DF	local	—	—	—	sec/ft	Scratch variable used in mean conic partials calculations.
C1	DF	local	—	—	—	—	Auxiliary variable used in F and G series computations and in Pines method.
C2	SF	local	—	—	—	sec^2/ft^2	Scratch variable used in mean conic partials calculations.
C2	DF	local	—	—	—	—	Auxiliary variable used in F and G series computations and in Pines method.
D	SF(3)	local	—	—	—	ft/sec^2	Variable used in ACCEL_ONORBIT to define vehicle drag acceleration.
DA_THRESHOLD (V90A4747C)	SF	local	—	—		micro G's	Threshold value for magnitude of sensed acceleration.
DA_THRESHOLD_TEST	SF	local	—	—	—	ft/sec^2	Threshold test value for magnitude of sensed acceleration.
DEG_PER_RAD	SF	compool	C	—	—	deg/rad	Conversion factor from radians to degrees.

VARIABLES LIST

Variable name (M/S ID)	Preci- sion & type	Compool or local	Initial- ization category	Initial value	Uplink/ downlist	Units	Description
DELQ	DP	compool	—	—	—	vary	Measurement residuals.
DELTA_RESID_RATIO (V96U9107C)	SP	local	DD	.2	—	—	Increment on the residual edit test ratio when data is forced.
DELTAT	DP	local	—	—	—	sec	Time interval between two positions in a conic (F and G series).
DELTAT_COAS	SP	local	—	—	—	sec	Time since last COAS measurement processing.
DELTAT_COAS_MAX (V96U9108C)	SP	local	DD	120.	—	sec	Maximum age of COAS measurement before navigation will not process it.
DELTAT_COAS_MIN (V96U9109C)	SP	local	DD	1.	—	sec	Minimum change in COAS time tag before it is considered by navigation to be a new measurement.
DELTAT_GO	DP	compool	—	—	—	sec	Time interval between two positions in a conic.
DELTAT_ST	SP	local	—	—	—	sec	Time difference between current star tracker measurement time and last star tracker measurement time.
DELTAT_ST_MIN (V96U9110C)	SP	local	DD	1.	—	sec	Minimum change in star tracker time tag needed to verify that the star tracker data has changed.
DELTIM	DP	local	—	—	—	sec	Time interval for the mean conic partial transition matrix calculation.
D_PIN	DP	local	—	—	—	—	Temporary storage variable used in Pines formulation (call argument to F and G).
D_PIN_TEMP	DP	compool	HC	0.	—	ft^2/sec	Temporary storage for D_PIN (Pines method).
DPL	B	compool	HC	1	—	—	Flag indicating activation (1) or deactivation of drag model.
DIAG	DP	local	—	—	—	sec	Scratch variable.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
DID_COVAR_REINIT (V90X1415X)	B	compool	IV	OFF	downlist	--	Positive feedback flag indicating (ON) that a covariance reinitialization has been performed.
DID_ONB_TO_TGT (V90X1413X)	B	compool	IV	OFF	downlist	--	Positive feedback flag indicating (ON) that a state vector transfer has been performed (TGT = ONB).
DID_TGT_TO_ONB (V90X1414X)	B	compool	IV	OFF	downlist	--	Positive feedback flag indicating (ON) that a state vector transfer has been performed (ONB = TGT).
D_IN	DP	local	--	--	--	ft^2/sec	Dot product of position and velocity used in Pines method in call to F and G series.
DISP_DELQ (V90H0212C,14C,21C, V90A0711C)	SP(4)	compool	--	--	downlist	vary	Display measurement residual.
DISP_EDIT (V90X0213X,15X,23X, 0712X)	C(4)	compool	IV	"blank"	downlist	--	Display edit status indicator.
DISP_SIG (V90U0216C,17C,22C, 713C)	SP(4)	compool	--	--	downlist	--	Display residual edit ratio.
DM	B	local	IV	OFF	--	--	Flag indicating use of or nonuse of drag model in ACCEL_ONORBIT.
D_MN_AN	DP	local	--	--	--	rad	Difference in mean anomalies used to solve Kepler's equations, used in F and G series.
DMP (V94X3969X)	B	compool	--	--	--	--	Flag indicating use or nonuse of drag acceleration in prediction.
DMN	DP	local	--	--	--	--	Local variable used in ACCEL_ONORBIT gravitational acceleration computation.
DO_COAS_ANGLES_NAV (V90X4880X)	B	local	--	--	downlist	--	On-off switch indicating (ON) that COAS angles data have been selected for processing.

VARIABLES LIST

Variable name (M/3 ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
DO_COAS_ANGLES_NAV_LAST (V90X4725X)	B	compool	--	—	—	--	On-off switch indicating (ON) that COAS angles data were selected for processing in the last filter cycle.
DO_COVAR_REINIT (V93X6233X)	B	compool	--	—	—	--	Flag indicating (ON) the performance of covariance matrix reinitialization.
DO_FLTR_SLOW_RATE (V93X6235X)	B	compool	IV	ON	—	--	Flag indicating rendezvous filter rate: ON = slow rate OFF = fast rate
DOING_FLTR_SLOW_RATE (V90X1365X)	B	compool	IV	ON	downlist	--	Flag indicating (ON) rendezvous filter rate is slow, and (OFF) rendezvous filter rate is fast. Provides positive feedback to REL_NAV DISPLAY.
DOING_MEAS_ENABLE (V90X1417X)	B	compool	IV	OFF	downlist	--	Flag indicating (ON) measurement data are being processed and (OFF) measurement data are not being processed. Provides positive feedback to REL_NAV DISPLAY.
DOING_PWRD_FLT_NAV (V90X1289X)	B	compool	IV	OFF	downlist	--	Flag indicating (ON) that a powered flight navigation phase is currently active and (OFF) a nonpowered flight nav phase is active. Provides positive feedback to REL_NAV DISPLAY.
DOING_REND_NAV (V90X4765X)	B	compool	IV	OFF	downlist	--	Flag indicating (ON) a rendezvous navigation phase is active and (OFF) a non-rendezvous navigation phase is active. Provides positive feedback to REL_NAV DISPLAY.
D_ONE	DP	local	—	—	—	ft^2/sec	Dot product of position and velocity vectors for transition matrix computation and F and G series.
DO_ORB_TO_TGT (V93X6215X)	B	compool	--	—	—	--	Flag indicating (ON) state vector transfer (TGT = ORB).

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
DO_OV_UPLINK (V90X4754X)	B	compool	IV	OFF	—	--	Flag indicating (ON) that data have been uplinked for Orbiter state vector update.
DO_RR_ANGLES_NAV (V90X4866X)	B	local	—	—	downlist	--	Flag indicating (ON) that rendezvous radar angles data have been selected for processing.
DO_RR_ANGLES_NAV_LAST (V90X4726X)	B	compool	--	—	—	--	Flag indicating (ON) that rendezvous radar angles data were selected for processing in the last filter cycle.
DO_RDOT_NAV_LAST (V90X4727X)	B	compool	—	—	—	--	Flag indicating (ON) that rendezvous radar range and range rate data were selected for processing in the last filter cycle.
DO_ST_ANGLES_NAV (V90X4868X)	B	compool	—	—	downlist	--	Switch indicating (ON) that star tracker angles data have been selected for processing.
DO_ST_ANGLES_NAV_LAST (V90X4728X)	B	compool	--	—	—	--	Flag indicating (ON) that star tracker angles data were selected for processing in the last filter cycle.
DO_TGT_TO_ORB (V93X6216X)	B	compool	—	—	—	--	Flag indicating (ON) state vector transfer (ORB = TGT).
DO_TV_UPLINK (V90X4757X)	B	compool	IV	OFF	—	--	Flag indicating (ON) that data have been uplinked for target state vector update.
D_SS (V90V4842C-44C)	SF(3)	compool	—	—	downlist	ft/sec^2	Drag of Shuttle for downlist.
DT_COV (V90W1366C)	DF	compool	—	—	downlist	sec	Change in time since the last COV propagation.
DT_FILT (V90W4946C)	DF	compool	—	—	—	sec	Interval over which to propagate the state vector.

VARIABLES LIST

Variable name (M/S ID)	Preci- sion & type	Compool or local	Initial- ization category	Initial value	Uplink/ downlist	Units	Description
DT_NAV_STATE_PROP (V95U8683C)	SP	compool	DD	4	--	sec	Onorbit/rendezvous nav state vector propagation rate.
DT_STEP	DP	local	--	--	--	sec	Integration step size for prediction or propagation.
D_TWO	DP	local	--	--	--	ft^2/sec	Dot product of the position and velocity vectors for the transition matrix computation and F and G series.
DV_COV (V90L4772C, 891C, 892C)	SP(3)	compool	--	--	downlist	ft/sec	Change in velocity since the last COV propagation.
DV_FILT (V90L4953C -55C)	SP(3)	compool	--	--	--	ft/sec	Differences between accumulated sensed IMU readings on the present cycle and the previous cycle (M50).
DV_IN	SP(3)	local	--	--	--	ft/sec	Input change in velocity used in SUPER_G integrator.
E (V90U4006C -09C V90U4011C -19C V90U4026C -38C V90U4045C -57C V90U4064C -76C V90U4083C -95C V90U4103C -15C V90U4122C -34C V90U4141C -53C V90U4160C -72C V90U4179C -91C V90U4198C -200C V90U4202C -11C V90U4218C -30C V90U4237C -49C)	DP(13,13)	compool	--	--	downlist	vary	Filter covariance matrix.
EARTH_MU	DP	compool	C	--	--	ft^3/sec^2	Gravitational constant of the Earth.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
EARTH_POLE	DF(3)	compool	C	—	—	--	Unit vector in direction of Earth's axis of rotation.
EARTH_RADIUS_EQUATOR	DF	compool	C	--	—	ft	Earth's equatorial radius.
EARTH_RADIUS_GRAV	DF	local	C	—	—	ft	Earth's radius used for gravitational accelerations in ACCEL_ONORBIT.
EARTH_RATE	DF	compool	C	—	—	rad/sec	Earth's angular rotation rate.
EB_COPY	DF(13)	local	—	—	—	vary	Covariance matrix times the partials vector.
EDIT_FLAG	C	compool	—	—	—	--	Five valued flag used to indicate the status of the sensor measurement processing: OFF - not suitable for processing ON - edited by the residual edit test STAT - processed for statistical display only PROCESSED - processed normally FORCED _ processed through force command
ELLIPT	DF	compool	C	--	—	--	Earth's ellipticity constant.
EPS_TIME (V96U9115C)	SF	local	DD	.01	—	sec	Minimum time separation for state vector interpolation.
ERR	DF	local	--	—	—	rad	Auxiliary variable used in F and G series (conic solution) calculations.
EVENT_E1 (V̄90X8164X)	B	compool	IV	OFF	downlist	--	OPS 3 to MM201
EVENT_60 (V̄90X8189X)	B	compool	IV	OFF	downlist	--	MM106 to MM201

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
EVENT_60A (V90X8160X)	B	compool	IV	OFF	downlist	--	OPS 8 to MM201
EVENT_60B V90X8645X)	B	compool	IV	OFF	downlist	--	MM201 to OPS 8
EVENT_60H (V90X8644X)	B	compool	IV	OFF	downlist	--	OPS 8 to OPS 3
EVENT_67 (V90X8646X)	B	compool	IV	OFF	downlist	--	MM201 to MM202
EVENT_73 (V90X8200X)	B	compool	IV	OFF	downlist	--	MM202 to MM201
EVENT_83 (V90X8655X)	B	compool	IV	OFF	downlist	--	MM201 to OPS 00
EVENT_84 (V90X8148X)	B	compool	IV	OFF	downlist	--	OPS 00 to MM201
EVENT_85 (V90X8180X)	B	compool	IV	OFF	downlist	--	MM201 to OPS 3
EXP_SHAPE_FACTOR (V97U5334C)	SP	local	DD	0.65793	—	--	Constant used in drag model drag coefficient calculation.
F	DP	local	—	—	—	--	Closed form version of the F time series.
FDOT	DP	local	—	—	—	sec^{-1}	Closed form version of the time derivative of F series.
FIFTY	DP(3,3)	local	—	—	—	--	Transformation matrix Earth fixed to M50 used in ACCEL_ ONORBIT.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
FILT_UPDATE (V90X0224X)	B	compool	IV	OFF	—	--	Flag indicating (ON) that current navigation cycle is complete.
FM1	DF	local	—	—	—	--	Auxiliary variable for F minus 1.
F1 F2 F3 F4	DF	local	--	—	—	--	Auxiliary variables in gravitational acceleration calculations used in ACCEL_ONORBIT.
G	DF	local	--	—	—	sec	Closed form version of the G time series.
G	DF(3)	local	--	—	—	ft/sec^2	Gravitational acceleration in ACCEL_ONORBIT.
G _CENTRAL	DF(3)	compool	--	—	—	ft/sec^2	Gravitational acceleration due to Earth as a point mass.
GDI	SF	local	--	—	—	--	Atmospheric density diurnal coefficient used in ACCEL_ONORBIT.
GDIE (V96U9116C)	SF	local	DD	1.375	—	--	Exponent of atmospheric density diurnal coefficient used in ACCEL_ONORBIT.
GDM1	DF	local	--	--	—	--	Auxiliary variable for GDOT minus 1.
GDOT	DF	local	--	—	—	--	Closed form version of the time derivative of the G series.
G _INT	DF(3)	local	--	—	—	ft/sec^2	Intermediate value of acceleration used in SUPER-G integration.
GMD	I	local	—	--	--	--	Flag indicating degree of gravitional potential model in ACCEL_ONORBIT.
GM_DEG (V96U9117C)	I	compool	DD	4	—	--	Flag indicating the degree of gravitational potential model.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
GM_DEG_LOW (V96U9118C)	I	compool	DD	2	—	--	Lowest degree of potential model in calls to the acceleration function; used for sensed velocity, powered flight integration.
GMDP (V94J3975C)	I	compool	—	—	—	--	Flag indicating the degree of gravitional potential model for prediction.
GMO	I	local	--	—	—	--	Flag indicating order of gravitional potential model to be used in ACCEL_ONORBIT.
GMOP (V94J3963C)	I	compool	—	--	--	--	Flag indicating the order of the gravitional potential model to be used for prediction.
GM_ORD (V96U9119C)	I	compool	DD	2	—	--	Flag indicating the order of the gravitational potential model.
GM_ORD_LOW (V96U9120C)	I	compool	DD	0	—	--	Lowest order of potential model in calls to the acceleration function; used for sensed velocity powered flight integration.
G_NEW (V90U4940C-42C)	DP(3)	compool	—	—	downlist	ft/sec^2	Orbiter acceleration vector (M50).
GO	SP	compool	C	—	—	ft/sec^2/ g	Earth gravity acceleration constant.
G_ONE	DP(3)	local	—	--	—	ft/sec^2	Acceleration vector in transition matrix calculation.
G_OUT	DP(3)	local	—	—	—	ft/sec^2	Acceleration vector used in SUPER_G.
G_TV (V90L4961C-63C)	DP(3)	compool	—	—	downlist	ft/sec^2	Target vehicle acceleration vector (M50).

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
G _TV LAST (V90A4796C,97C,88C)	DF(3)	compool	—	—	—	ft/sec^2	Target vehicle acceleration vector, last value (M50).
G _TWO	DF(3)	local	—	—	—	fc/sec^2	Acceleration vector in transition matrix calculation.
G_2_FPS2	SF	compool	C	—	—	lb/slug	Mass to weight conversion.
HORIZ	SF	local	—	—	—	rad	Filter estimate of the horizontal angle measurement.
I	I	local	—	—	—	—	Counter
IATM	I	local	—		—	—	Attitude mode flag that controls the use or nonuse of prestored average area, mass, and drag coefficient of Orbiter or target vehicle.
1_CYCLE (V90J4889C)	I	compool	—	—	downlist	—	Counter for the navigation cycle.
IDM	B	local	—	—	—	—	Flag indicating the activation (1) or deactivation (0) of the drag model.
ID_MATRIX_3X3	SF(3,3)	local	HC	$\begin{bmatrix}1. & 0. & 0.\\0. & 1. & 0.\\0. & 0. & 1.\end{bmatrix}$	—	—	3 by 3 identity matrix.
IDRAG (V90X4950X)	B	compool	—	—	—	—	Drag model flag used by state prop.
IGD (V90J4947C)	I	compool	—	—	—	—	Temporary storage for the potential model degree used by state prop.
IGO (V90J4948C)	I	compool	—	—	—	—	Temporary storage for the potential model order used by state prop.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
I _N	SF(3)	local	--	—	—	--	Raw vector of M50 to sensor transformation matrix used in angles partials calculation.
I _RHO	SF(3)	compool	—	—	—	ft	Unit line-of-sight vector.
IMU_NAV_ACCEL_THRESH (V93A6710C)	SF	compool	MD	—	downlist	micro-g's	Crew selected acceleration threshold value for incorporating IMU data into nav.
IVENT (V90X4951X)	B	compool	--	—	—	--	Temporary value of venting model flag used in state propagator.
IVM	B	local	—	--	--	--	Flag indicating activation (1) or deactivation (0) of the venting and RCS uncoupled thrusting model.
J	I	local	--	—	—	--	Counter.
K	I	local	--	—	—	--	Counter.
KFACTOR (V96U8173C)	SF	compool	MD	—	uplink	--	Drag model correction factor used in ACCEL_ONORBIT drag model.
K_RESID_EDIT (V96U9121C)	SF	local	DD	225.	—	--	Residual edit test scale factor.
K_UND_WGT (V97U5350C)	SF	local	DD	0.2	—	--	Filter underweighing factor.
L	I	local	—	—	—	--	Counter.
LOC (V96U9122C)	DF	local	MD	--	--	--	Coefficient used in ACCEL_ONORBIT solar ephemeris model.
LOS	DF	local	--	—	—	rad	Longitude of the Sun in M50 coordinates.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
LOSK1 (V96U9123C)	DF	local	MD	—	—	--	Coefficient used in ACCEL_ONORBIT solar ephemeris model.
LOSK3 (V96U9124C)	DF	local	MD	—	—	--	Coefficient used in ACCEL_ONORBIT solar ephemeris model.
LOS_R (V96U9125C)	DF	local	DD	1.9909865 94E-7	—	rad/sec	Coefficient used in ACCEL_ONORBIT solar ephemeris model.
LOS_ZERO (V96U9126C)	DF	local	MD	—	—	rad	Coefficient used in ACCEL_ONORBIT solar ephemeris model.
MANUAL_EDIT_OVERRIDE	B	compool	--	—	—	--	Copy of the manual edit override flag of sensor data type currently being processed that is sent to the filter.
MASS_INIT (V90U4938C)	SF	compool	OPS	—	—	slugs	Used for OPS 2, 8, 0 initialization of current Orbiter mass.
MAT	SF(3,3)	local	--	—	—	--	UVW to M50 transformation matrix for the filtered vehicle in COV prop.
MAX_TIME_TOL (V96U8685C)	SF	local	DD	54000.	—	sec	Maximum time threshold for the state vector prediction task. If the absolute value of the difference between current filter time and the state vector exceeds this threshold, no state vector prediction will take place.
M_BODYM50	SF(3,3)	compool	--	—	—	--	Body to M50 transformation matrix.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
M_BODY_TO_COAS (V96U9127C-44C)	SP(3,3,2)	local	DD	$\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}$ $\begin{bmatrix}0 & 0 & -1\\0 & 1 & 0\\1 & 0 & 0\end{bmatrix}$	—	--	Body to COAS coordinates transformation matrix.
M_BODY_TO_RR (V93U9145C-53C)	SP(3,3)	compool	DD	$\begin{bmatrix}0 & 0 & -1\\0 & 1 & 0\\1 & 0 & 0\end{bmatrix}$	—	--	Body to RR coordinates transformation matrix.
MEAS_ENABLE (V93X6232X)	B	compool	IV	OFF	—	--	Flag indicating that measurement processing is enabled during MM202.
MEAS_STAT	B	local	—	--	—	--	Temporary flag used in measurement selection.
MEAS_THRESHOLD (V96U9154C)	SP	compool	DD	0.00016	—	ft/sec^2	Threshold value for the magnitude of sensed accelerations for nonuse of measurement data.
MM_202 (V90X8656X)	B	compool	--	—	—	--	Major mode 202 indicator.
M_M50_TO_BODY_COAS (V90U4857C -65C)	SP(3,3)	compool	—	—	downlist	--	Snapped variable for the M50 to body coordinates transformation matrix at the time of COAS data snap (STAR TRCK SOP HAL name T_M50_BODY, M/S ID #V95U4809-17C).
M_M50_TO_SENSOR	SP(3,3)	compool	—	—	—	--	M50 to sensor coordinates transformation matrix for the sensor type being processed.
M_M50_TO_ST (V90U4920C -28C)	SP(3,3)	compool	—	—	downlist	--	Snapped variable for M50 to star tracker coordinate transformation matrix (STAR TRCK SOP HAL name T_M50_ST, M/S ID #V95U4800 -8C).

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
MS_DELQ	DP	local	—	—	—	vary	Estimated variance of the computed sensor measurement.
MS_POS_UND_WGT (V97U5471C)	SP	local	DD	5.0E 4	—	ft^2	Threshold of the Shuttle position variance for filter underweighing.
M_UVW_M50	DP(3,3)	compool	—	—	—	--	UVW to M50 transformation matrix.
N	I	local	—	—	—	--	Counter.
N_ACCEPT (V90J1402C,03C,00C, 01C)	I(4)	compool	IV	0	downlist	--	Counter for the number of data marks that have been used to update the navigation state vector.
NAV_ANGLES_AIF (V90J4253C)	C	compool	—	—	downlist	--	Auto/Inhibit/Force switch used in nav for the currently enabled angles data set.
NAV_CURR_ORB_MASS (V90U4254C)	SP	compool	—	—	downlist	slugs	Current Orbiter mass used in nav.
NAV_DO_COVAR_REINIT (V90X4256X)	B	compool	—	—	downlist	--	Nav flag tested to determine when to reinitialize the the covariance matrix
NAV_DO_FLTR_SLOW_RATE (V90X4257X)	B	compool	—	—	downlist	--	Nav flag indicating rendezvous filter rate: ON = slow rate OFF = fast rate
NAV_DO_ORB_TO_TGT (V90X4258X)	B	compool	—	—	downlist	--	Nav flag indicating (ON) crew - requested state vector transfer (Orbiter to target)
NAV_DO_OV_UPLINK (V90X4260X)	B	compool	—	—	downlist		Nav flag indicating (ON) Orbiter state vector uplink has occurred.
NAV_DO_TGT_TO_ORB (V90X4259X)	B	compool	—	—	downlist		Nav flag indicating (ON) a crew requested state vector transfer (target to Orbiter)

VARIABLES LIST

Variable name (M/S ID)	Preci- sion & type	Compool or local	Initial- ization category	Initial value	Uplink/ downlist	Units	Description
NAV_DO_TV_UPLINK (V90X4261X)	B	compool	—	—	downlist		Nav flag indicating (ON) target state vector uplink has occurred.
NAV_MEAS (V90X4967X)	B	compool	IV	OFF	downlist	--	Flag indicating (ON) that sensor measurements are to be processed.
NAV_MEAS_ENABLE (V90X4262X)	B	compool	—	—	downlist	--	Nav flag indicating (ON) that measurement processing is enabled in MM 202.
NAV_MM_202 (V90X4263X)	B	compool	—	—	downlist	--	Major mode 202 indicator for NAV.
NAV_PWRD_FLT_NAV (V90X4264X)	B	compool	—	—	downlist	--	Flag indicating (ON) to NAV that powered flight navigation should be exercised.
NAV_RANGE_AIF (V90J4268C)	C	compool	—	—	downlist	--	Auto/Inhibit/Force switch used in NAV in processing the rendezvous radar range measurements.
NAV_RDOT_AIF (V90J4269C)	C	compool	—	—	downlist	--	Auto/Inhibit/Force switch used in NAV in processing the rendezvous radar range rate measurements.
NAV_RR_ANGLES_ENABLE (V90X4273X)	B	compool	—	—	downlist	--	Rendezvous radar angles enable flag for NAV.
NAV_SIG	SP(4)	compool	—	—	—	--	Copy of the residual edit test ratios for the sensor measurements processed.
NAV_ST_ENABLE (V90X4274X)	B	compool			downlist		Star tracker angles enable flag for NAV.
N_CYCLE (V90M4773C)	I	compool	—	—	—	--	Frequency at which measurements are to be processed and the covariance matrix propagated.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
N_CYCLE_SLOW (V96U8682C)	I	compool	DD	4	—	—	Multiplier for frequency of covariance matrix propagation.
N_CYCLE_FAST (V96U8681C)	I	compool	DD	2	—	—	Multiplier for frequency of covariance matrix propagation.
NOISY_NAV_MEAS (V90X4937X)	B	—	IV	OFF	—	—	Flag indicating (ON) that the sensed acceleration is above MEAS_THRESHOLD and measurement data are not to be used.
N_REJECT (V90J1407C,08C,05C, 06C)	I(4)	compool	IV	0	downlist	—	Counter for the number of data marks that have been edited by the navigation filter.
N_STEPS	I	local	—	—	—	—	Number of integration steps in the prediction or propagation interval.
NUM_KEP_ITER	I	local	BC	5	—	—	Maximum number of iterations in the solution of Kepler's equation (F and G).
N1	I	local	—	—	—	—	Variable used in ACCEL_ONORBIT Earth gravitational code.
OMEG_C (V96U9157C)	DF	local	DD	1.990968 7158-7	—	rad/sec	Coefficient used in ACCEL_ONORBIT solar ephemeris model.
OMEGA	DF(13)	local	—	—	—	vary	Kalman gains vector.
OMEMRIN	DF	local	—	—	—	—	Auxiliary variable used in solving Kepler's equation (F and G series).
OPS_2_OR_8_INITIALIZE_COMPLETE (V90X1242X)	B	compool	—	—	—	—	Signal to MSC indicating that initialization of user parameter state propagation quantities is complete.

VARIABLES LIST

Variable name (M/S ID)	Preci- sion & type	Compool or local	Initial- ization category	Initial value	Uplink/ downlist	Units	Description
OV_PREDICT_FAIL (V90X4770X)	B	compool	--	—	downlist	--	Flag which indicates that the time threshold, MAX_TIME_TOL, has been exceeded and consequently no Orbiter state vector prediction will take place.
PHASE_C (V96U9158C)	DF	local	MD	—	—	rad	Coefficient used in ACCEL_ONORBIT solar ephemeris model.
PHI	DF(6,9)	local	IV	0.	—	vary	State transition matrix from previous filter time to current time.
PHI_BIAS	DF(4)	local	--	—	—	--	Diagonal elements of the sensor bias portion of the state transition matrix.
PHI_MC	DF(6,6)	local	--	--	—	vary	Patch transition matrix computed by mean conic partials.
PHI_PATCH	DF(6,6)	compool	—	--	--	vary	Transition matrix for converting from time of measurement to current time.
PHI_UNMOD_ACC	DF	local	--	—	—	--	Diagonal elements of the unmodeled acceleration portion of the state transition matrix.
PI	DF	local	C	—	—	--	Transcendental constant which is the ratio of the circumference of a circle [illegible] its diameter.
PREC_STEP_PRED (V98U8726C)	SF	compool	DD	300.0	—	sec	Integration step size for precise prediction.
PRED_ORB_AREA (V93U6955C)	SF	compool	—	—	downlist	ft^2	Orbiter's cross-sectional area for use in prediction.
PRED_ORB_CD (V93U6954C)	SF	compool	—	—	downlist	--	Orbiter's drag coefficient for use in prediction.

VARIABLES LIST

Variable name (M/S ID)	Preci-sion & type	Compool or local	Initial-ization category	Initial value	Uplink/ downlist	Units	Description
PRED_ORB_MASS (V93U6953C)	SF	compool	—	—	downlist	slugs	Orbiter mass used in prediction.
PRED_STEP (V94W3964C)	SF	compool	—	—	downlist	sec	Prediction step size.
PRED_STEP_OPS_INIT (V96U8686C)	DF	local	DD	610.	—	sec	Initial predictor step size used in OPS_2_CR_8_INITIALIZE.
PRED_TASK_COMPLETE	B	local	—	—	—	--	A flag, local to the state vector prediction task, indicating that the prediction task is complete.
PRED_USE (V90J4768CA)	I	compool	IV	0	downlist	--	A flag, having values 0 thru 9, which indicates to the user of the state vector prediction task the status of the prediction task.
PWRD_FLT_NAV (V93X5408X)	B	compool	—	—	downlist	--	Flag indicating use of powered flight propagation (ON) or coasting flight propagation (OFF).
Q_COAS_HORIZ (V90U4847C)	SF	compool	—	—	downlist	rad	Snapped variable for the COAS horizontal angle measurement (STAR TRCK SOP HAL name COAS_HORIZ, M/S ID #V95U4969C).
Q_COAS_VERT (V90U4848C)	SF	compool	—	—	downlist	rad	Snapped variable for the COAS vertical angle measurement (STAR TRCK SOP HAL name COAS_VERT, M/S ID #V95U4992C).
Q_HORIZ	SF	local	—	—	—	rad	Measurement from horizontal measurement sensor.
Q_M50BODY_IMU (V95U0873C -6C)	SF(4)	compool	—	—	downlist	--	Snapped variable for the Shuttle attitude quaternion at the time of the IMU data snap (ORB ATT PROC HAL name Q_BOD_M50, M/S ID# V90U2240-3C).
Q_M50BODY_RR (V90U4829C -32C)	SF(4)	compool	—	—	downlist	--	Snapped Shuttle attitude quaternion at the time of the rendezvous radar data snap (RR SOP HAL name Q_M50BODY_RR, M/S ID# V95U4312-15C).

VARIABLES LIST

Variable name (M/S ID)	Preci- sion & type	Compool or local	Initial- ization category	Initial value	Uplink/ downlist	Units	Description
Q_RR_RNG (V̄90U4895C)	SF	compool	--	—	downlist	ft	Snapped variable for the rendezvous radar range measurement (RR SOP HAL name RR_RANGEO, M/S ID# V95H4287C).
Q_RR_RNG_DOT (V̄90U4896C)	SF	compool	—	—	downlist	ft/sec	Snapped variable for the rendezvous radar range rate measurement (RR SOP HAL name RR_RNGRO, M/S ID# V95R4288C).
Q_RR_SHFT (V̄90U4893C)	SF	compool	--	—	downlist	deg	Snapped variable for the rendezvous radar shaft angle measurement (RR SOP HAL name RR_ROLLO, M/S ID # V95H4284C).
Q_RR_TRUN (V̄90U4894C)	SF	compool	—	—	downlist	deg	Snapped variable for the rendezvous radar trunnion angle measurement (RR SOP HAL name RR_PITCHO, M/S ID # V95H4281C).
Q_ST_HORIZ (V̄90U4833C)	SF	compool	--	—	downlist	rad	Snapped variable for the star tracker horizontal angle measurement (STAR TRCK SOP HAL name H_NAV, M/S ID # V95H4753C).
Q_ST_VERT (V̄90U4834C)	SF	compool	--	—	downlist	rad	Snapped variable for the star tracker vertical angle measurement (STAR TRCK SOP HAL name V_NAV, M/S ID # V95H4754C).
Q_VERT	SF	local	--	—	—	rad	Measurement from vertical measurement sensor.
R	DF(3)	local	--	—	—	ft	Temporary M50 position vector used in ACCEL_ONORBIT.
RAD_PER_DEG	SF	compool	C	—	—	rad/deg	Conversion factor from degrees to radians.
RANGE_AIF (V93J6241C)	C	compool	--	—	—	--	Auto/inhibit/force switch associated with the rendezvous radar range measurement.
RANGE_AIF_DISPLAY (V90J1362C)	C	compool	—	—	downlist	--	Flag indicating to displays that the range auto/inhibit/ force flag has been processed.
RANGE_AIF_LAST	C	local	IV	inhibit	—	—	Last value of RANGE_AIF.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/downlist	Units	Description
RANGE_EDIT_OVERRIDE	B	local	--	--	--	--	Flag indicating (ON) that the crew wishes to force rendezvous radar range measurement.
RANGE_STAT	B	local	--	--	--	--	Flag indicating (ON) that rendezvous radar range data is to be processed for statistical display only.
RDOT_AIF ($\overline{V}$93J6247C)	C	compool	--	--	--	--	Auto/inhibit/force switch associated with the rendezvous radar range rate measurement.
RDOT_AIF_DISPLAY ($\overline{V}$90$\overline{J}$1368C)	C	compool	--	--	downlist	--	Flag indicating to displays that the range rate auto/inhibit/force flag has been processed.
RDOT_AIF_LAST	C	local	IV	inhibit	--	--	Last value of RDOT_AIF.
RDOT_DATA_GOOD ($\overline{V}$90X4$\overline{9}$00X)	B	compool	--	--	downlist	--	Snapped data validity flag for the rendezvous radar range rate measurement (RR SOP HAL name RR_RNGR_DG, M/S ID # V95X4294X).
RDOT_EDIT_OVERRIDE	B	local	--	--	--	--	Flag indicating (ON) that the crew wishes to force rendezvous radar range rate measurement.
RDOT_STAT	B	local	--	--	--	--	Flag indicating (ON) that rendezvous radar range rate data is to be processed for statistical display only.
$\underline{R}$_EF	DF(3)	local	--	--	--	ft	Earth fixed position vector used in ACCEL_ONORBIT.
REF_ORB_AREA ($\overline{V}$96$\overline{U}$9160C)	SF	compool	DD	2690.	--	ft^2	Orbiter reference cross sectional area used in propagation drag calculations.
REF_ORB_CD ($\overline{V}$96$\overline{U}$9161C)	SF	compool	DD	2.0	--	--	Orbiter reference drag coefficient.
REND_NAV_INIT_PRED	B	compool	IV	OFF	--	--	Flag indicating prediction task in progress as scheduled by REND_NAV_INIT.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
REJ_MAX (V96U9162C)	I	local	DD	3	—	--	Maximum number of sequential marks; rejected by the residual edit test, before DISP_EDIT for the corresponding measurment is set to [illegible].
REND_NAV_FLAG (V93X6220X)	B	compool	IV	OFF	downlist	--	Flag indicating whether rendezvous navigation (ON) or on-orbit navigation (OFF) is in operation.
REND_NAV_FLAG_LAST	B	local	IV	OFF	--	--	Last value of REND_NAV_FLAG.
RESID_RATIO_OLD	SP	compool	—	--	--	--	Value of residual ratio from the previous cycle for the measurement being processed.
RESID_TEST	SP	local	--	—	—	vary	Scaled value of the estimated measurement variance for comparision with the measurement deviation squared in the residual edit test.
RESID_TEST_RATIO	SP	compool	--	--	—	--	Ratio of the magnitude of the measurement residual and the scale value of the estimated measurement standard deviation to be used in the residual edit test.
R _FILT (V90H0584C -6C)	DP(3)	compool	--	—	—	ft	Orbiter position vector (M50).
R _FILT_INIT (V90H1276C -8C)	DP(3)	compool	OPS	--	--	ft	Orbiter position vector saved across memory reconfiguration and used for navigation initialization.
R _FILT_TLM (V90H4277C-79C)	DP(3)	compool	--	—	downlist	ft	Orbiter position vector for downlist (M50).
R _FIN	DP(3)	local	—	--	--	ft	Position vector at the end of a time interval in M50. Used in F & G series and in SUPER_G.
R_FIN_INV	DP	local	--	—	—	ft^{-1}	Reciprocal of the magnitude of R _FIN.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
R_FIN_TEMP_INV	DF	local	—	—	—	ft^{-1}	Dummy variable used in the call to F_AND_G.
R _GND (V96H3500C-02C)	DF(3)	compool	—	—	uplink	ft	Uplink Orbiter position vector in M50.
RHO	SF	local	—	—	—	$slugs/ft^3$	Atmospheric density used in ACCEL_ONORBIT drag model.
RHO_PLANE	SF(3)	local	—	—	—	ft	In-plane component of the line of sight.
R _IN	DF(3)	local	—	—	—	ft	Position vector at the beginning of a time interval in M50. Used in F & G series and in SUPER_G.
R_IN_INV	DF	local	—	—	—	ft^{-1}	Reciprocal of the magnitude of R _IN.
R_INV	DF	local	—	—	—	1/ft	Inverse of M50 position used in ACCEL_ONORBIT.
R _LAST (V90H4782C -4C)	DF(3)	compool	—	—	—	ft	Position vector of the Orbiter at the end of the last filter cycle (M50).
RNG	SF	local	—	—	—	ft	Computed range measurement.
RNG_DATA_GOOD (V90X4899X)	B	compool	—	—	downlist	--	Snapped data validity flag for range measurement (RR SOP HAL name RR_RNG_DG, M/S ID # V95X4293X).
RNG_DOT	SF	local	—	—	—	ft/sec	Computed range rate measurement.
RNG_MIN (V96U9163C)	SF	local	DD	1.0	—	ft	Minimum separation between the Orbiter and the target.
RO_N	DF	local	—	—	—	ft/sec^2	Distance term used in ACCEL_ONORBIT gravitational acceleration model.
R _ONE	DF(3)	local	—	—	—	ft	Position vector at the beginning of an interpolation interval (M50).

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
R_ONE_INV	DF	compool	—	—	—	ft^{-1}	Reciprocal of the magnitude of R _ONE.
RO_ZERO	DF	local	—	—	—	—	Distance term in ACCEL_ONORBIT gravitational acceleration model.
R _PRED_FINAL (V95H0811C -3C)	DF(3)	compool	—	—	downlist	ft	Position vector as output from precise state predictor (M50).
R _PRED_INIT (V94H4002C -4C)	DF(3)	compool	—	—	downlist	ft	Position vector input for prediction (M50).
RR_ANGLE_DATA_GOOD (V90X4901X)	B	compool	—	—	downlist	—	Snapped data validity flag for the rendezvous radar angles measurement (RR SOP HAL name RR_ANG_DG, M/S ID #V95X4295X).
RR_ANGLE_MARK_NUM (V90J4824C)	I	compool	IV	0	downlist	—	Rendezvous radar angle (shaft plus trunnion) mark counter.
RR_ANGLES_BIAS_INIT (V96U9164C-5C)	SF(2)	local	MD	—	—	rad	Initial value for the rendezvous radar angles bias slot of the state vector.
RR_ANGLES_EDIT_ OVERRIDE	B	local	—	—	—	—	Flag indicating (ON) that the crew wishes to force rendezvous radar angles data.
RR_ANGLES_ENABLE (V93X6230X)	B	compool	—	—	—	—	Rendezvous radar angles enable flag.
RR_ANGLES_STAT	B	local	—	—	—	—	Flag indicating (ON) that rendezvous radar angles data is to be processed for statistical display only.
RRDOT_BIAS_INIT (V96U9166C-7C)	SF(2)	local	MD	—	—	ft, ft/sec	Initial value for the rendezvous radar range and range rate bias slots of the state vector.
RRDOT_MARK_NUM (V90J4825C)	I	compool	IV	0	downlist	—	Range and range rate data mark counter.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
RREF (V96U9168C)	DF	local	DD	2.3769016 E-3	—	slugs/ft^3	Constant used to define units of density in ACCEL_ONORBIT drag model.
R _RESET (V90H0235C -7C)	DF(3)	compool	—	—	—	ft	Orbiter position vector reserved for the reset of user parameter state propagation position vector (M50).
R _RESID	DF(3)	local	—	—	—	ft	Orbiter position vector interpolated to measurement time (M50).
R _RHO	SF(3)	compool	—	—	—	ft	Line of sight (M50).
R_RHO_MAG	SF	compool	—	—	—	ft	Magnitude of the relative position vector.
R _TV (V90H0307C -9C)	DF(3)	compool	MD	—	—	ft	Target vehicle M50 position vector.
R _TV_GND (V96H1277C -9C)	DF(3)	compool	—	—	uplink	ft	Uplink target position vector in M50.
R _TV_LAST (V90H4789C -91C)	DF(3)	compool	—	—	—	ft	Target position vector at the end of last filter cycle (M50).
R _TV_RESET (V90H1363C -5C)	DF(3)	compool	—	—	—	ft	Target position vector reserved for the reset of user parameter state propagation position vector (M50).
R _TV_RESID	DF(3)	local	—	—	—	ft	Target position vector interpolated to measurement time (M50).
R _TV_TLM (V90H4287C-89C)	DF(3)	compool	—	—	downlist	ft	Target position vector for downlist (M50).
R _TWO	DF(3)	local	—	—	—	ft	Position vector at the end of the interpolation interval (M50).

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
R_TWO_INV	DF	local	--	--	--	ft^{-1}	Reciprocal of the magnitude of $\underline{R}$ _TWO.
$\underline{S}$	DF(9)	local	C	--	--	--	Tesseral harmonics coefficients used in ACCEL_ONORBIT gravitational acceleration.
S	DF(6,6)	local	IV	0.	--	vary	State noise matrix for the covariance propagation.
SA	SF	local	--	--	--	--	Square of the sine of the angle of attack used in ACCEL_ONORBIT drag model.
SB	SF	local	--	--	--	--	Absolute value of the sine of side slip angle used in ACCEL_ONORBIT drag model.
$\underline{S}$ _BIAS	DF(4)	local	--	--	--	(rad^2, rad^2, ft^2, ft^2/ sec^2)	State noise terms for the sensor bias slots.
SDEC	DF	compool	--	--	--	--	Sine of solar declination used in ACCEL_ONORBIT atmospheric density model.
SELF_TEST_FLAG (V90X4888X)	B	compool	--	--	downlist	--	Snapped flag indicating that rendezvous radar is in the self-test mode. (RR SOP HAL name RR_SELF_TEST, M/S ID # V95X4309X.)
SENSOR_BIAS	SF(4)	compool	IV	0,0,0,0	--	rad,rad, ft,ft/sec	Sensor biases, part of state vector (M50).
SENSOR_BIAS_TLM (V90U4295C-98C)	SF(4)	compool	--	--	downlist	rad,rad, ft,ft/sec	Sensor biases (part of state vector) for TLM (M50).
SENSOR_DELQ	SF(4)	compool	--	--	--	vary	Measurement residuals.
SENSOR_DELTA	DF	local	DD	$1/\sqrt{3}$	--	--	An increment used to adjust the relative unit vector for measurement processing.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
SENSOR_EDIT (V90U8541C, V90J8880C-82C)	C(4)	compool	—	—	downlist	--	Five-valued parameter defining use of measurement data by the navigation filter. ON - rejected by the residual edit test OFF - no processing attempted PROCESSED - accepted by residual edit test and used to update the state vector STAT - used to generate display parameters FORCED - used to update state vector as a result of manual edit override
SENSOR_EPS	DF	local	DD	—	—	ft	Minimum allowable distance between target and shuttle for measurement processing.
SEQ_ACCEPT (V90J4732C,33C,30C, 31C)	I(4)	compool	IV	0	downlist	--	Number of sequential sensor marks accepted by the residual edit test for the various sensor measurements.
SEQ_REJECT (V90J4737C,38C,35C, 36C)	I(4)	compool	IV	0	downlist	--	Number of sequential sensor marks rejected by the residual edit test for the various sensor measurements.
SHAFT	SF	local	—	—	—	rad	Estimate of the rendezvous radar shaft measurement.
SHUTTLE_FILTER_FLAG (V90X4952X)	B	compool	MD	—	downlist	--	Flag indicating which vehicle state is to be used by the filter; (ON) Shuttle state (OFF) target state.
SIFST	SF	local	--	--	—	ft	Variable used in atmospheric density model in ACCEL_ONORBIT.
SIG_RR_RNG (V96U9168C)	SF	local	DD	0.	--	feet	One sigma statistic of rendezvous radar range measurement.
SIG_UPDATE (V96H1284C-9C)	SF(6)	compool	MD	—	uplink	ft, ft/sec	Standard deviations (UVW) for the filter vehicle position/ velocity covariance initialization.
SIN_LAG (V96U9169C)	SF	local	DD	.601815023	—	--	Sine of atmospheric density bulge lag angle used in ACCEL_ONORBIT.

VARIABLES LIST

Variable name (M/S ID)	Preci- sion & type	Compool or local	Initial- ization category	Initial value	Uplink/ downlist	Units	Description
SIN_SOL_RA	DF	compool	—	—	—	--	Sine of solar right ascension used in ACCEL_ONORBIT density model.
SLOPE_SIG_RR_RNG (V96U9170C)	SF	local	DD	3.E-3	—	--	Rate of change of rendezvous radar range statistics with respect to range.
SMA	DF	compool	—	—	—	ft	Semimajor axis of conic.
SQR_EMU (V90U1241C)	DF	compool	—	—	—	$ft^{3/2}/sec$	Square root of EARTH_MU used in onorbit prediction/ propagation equations of motion (Pines method).
SSND	SF	local	—	—	—	ft	Variable used in ACCEL_ONORBIT atmospheric density calculation.
ST_ANGLES_BIAS_INIT (V96U9171C-2C)	SF(2)	local	MD	—	—	rad	Initial value for the star tracker angles bias slots of the state vector.
ST_ANGLES_EDIT_ OVERRIDE	B	local	—	—	—	--	Flag used (ON) to override the residual edit test for star tracker angles data.
ST_ANGLES_STAT	B	local	--	—	—	--	Flag indicating (ON) that star tracker angles data are to be processed for statistical display only.
STAT_FLAG	C	compool	—	—	—	--	Copy of the stat flag associated with the measurement type currently being processed.
ST_DATA_GOOD (V90X4835X)	B	compool	—	—	downlist	--	Snapped data validity flag for Star tracker data (STAR TRCK SOP HAL name DATA_GOOD, M/S ID # V95X4771X).
ST_ENABLE (V93X6223X)	B	compool	—	—	—	--	Star tracker angle ENABLE flag.
ST_MARK_NUM (V90J4826C)	I	compool	IV	0	downlist	--	Star tracker measurement mark counter.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
S_UNMOD_ACC	DF(3,3)	local	--	—	—	ft^2/sec^4	Unmodeled acceleration noise terms.
SV_TIME_TAG_DIFF	SF	local	HC	0.5		sec	Time tolerance used to test the time tag of the predicted state against the reset time tag of navigation.
SV_UPDATE	B	local	—	--	—	--	A flag, local to the auto-inflight update subfunction, which indicates (ON) that a state vector update has occurred.
S0	DF	local	—	—	—	--	Auxiliary variables used in F & G series and Pines computations, S1 & S2 also used as auxiliary variables in mean conic partials transition matrix computations.
S1						vary	
S2						vary	
S3						vary	
S2B	SF	local	—	—	—	--	Absolute value of sine of twice the side slip angle used in ACCEL_ONORBIT drag model.
T	DF	local	--	—	—	sec	Time variable (GMT) used in ACCEL_ONORBIT.
TARGET_AREA (V96U9173C)	SF	compool	MD	—	—	ft^2	Target vehicle reference cross-sectional area.
TARGET_CD (V96U9174C)	SF	compool	MD	—	—	--	Target vehicle reference drag coefficient.
TARGET_MASS (V96U9175C)	SF	compool	MD	—	—	slugs	Target vehicle reference mass.
TAU	SF(2)	local	--	—	--	sec	Local variable for sensor time constants.
TAU_COAS_ANGLES (V96U9176C-7C)	SF(2)	local	DD	1300., 1300.	--	sec	Time constant for the COAS angles sensor.
TAU_RR_ANGLES (V96U9178C-9C)	SF(2)	local	DD	1.E6,1.E6	—	sec	Correlation time constant for the rendezvous radar angle measurement.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
TAU_RRDOT (V96U9180C-1C)	SF(2)	local	DD	1.E6,1.E9	--	sec	ECRV correlation time vector for rendezvous radar range and range rate.
TAU_SENS	SF(4)	compool	IV	1.,1.,1., 1.	—	sec	General correlation time constants for sensors.
TAU_ST_ANGLES (V96U9182C-3C)	SF(2)	local	DD	1300., 1300.	—	sec	Correlation time constant for star tracker measurements.
TAU_U_A_COAST (V96U9184C)	SF	compool	DD	300.	—	sec	Correlation time constant for unmodeled accelerations during coasting flight.
TAU_U_A_PWRD_FLT (V96U9185C)	SF	local	DD	300.	—	sec	Correlation time constant for unmodeled accelerations during powered flight.
TAU_UNMOD_ACC_COV (V90W4929C)	SF	compool	DD	300.	—	sec	Correlation time constant for unmodeled accelerations used in covariance propagation.
TAU_UNMOD_ACC_STATE (V96U9186C)	SF	local	DD	300.	—	sec	Correlation time constant for unmodeled accelerations used in state propagation.
T_BIAS_COAS (V96U9187C)	SF	local	DD	0.	—	sec	Time tag bias associated with COAS measurements.
T_BIAS_REND_RADAR (V96U9188C)	SF	local	DD	0.	—	sec	Time tag bias associated with rendezvous radar measurements.
T_BIAS_ST (V96U9189C)	SF	local	DD	0.	—	sec	Time tag bias associated with star tracker measurements.
T_COAS (V90W4853C)	DF	compool	—	—	downlist	sec	Snapped variable for the COAS measurement time tag (STAR TRCK SOP HAL name T_TARLOS, M/S ID # V95W4985C).
T_COAS_LAST	DF	local	IV	0.	—	sec	Time of last COAS measurement update.
T_COV_LAST (V90W4771C)	DF	compool	--	—	—	sec	Time tag of last covariance propagation.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
T_CUR	DF	local	--	--	--	sec	Current integration time within the predictor or propagator.
T_CURRENT_FILT (V90W4749C)	DF	compool	--	--	downlist	sec	Snapped time tag of the current IMU/ATT data (IMU SOP HAL name T_IMUS_GA, M/S ID # V95W0002C).
TDV_FILT	SF(3)	local	HC	0,0,0	--	ft/sec	Dummy vector for target accumulated velocity.
TEMP	DF(13,13)	local	--	--	--	vary	Temporary matrix used during covariance matrix propagation to store the intermediate calculation of E_ϕ^T during the $\phi E\phi^T + S$ computation.
TEMP	DF(3)	local	--	--	--	sec/feet	Temporary vector used by the mean conic partial.
TEST_VALUE	SF	local	--	--	--	--	Value against which the residual ratio is tested. If the residual ratio is greater than TEST_VALUE, the measurement is edited.
T_FILT_INIT (V90W1287C)	DF	compool	GPS	--	--	sec	Time tag of navigation initialization data carried across memory transition.
T_FIN	DF	local	--	--	--	sec	Final time at end of prediction or propagation, used in SUPER_G.
TFOFF (V96W8178C) (V90W4960C)	DF	compool	MD	--	uplink, downlist	sec	Vent/RCS thrust force off time used in ACCEL_ONORBIT.
TFON (V96W8177C) (V90W4959C)	DF	compool	MD	--	uplink, downlist	sec	Vent/RCS thrust force on time used in ACCEL_ONORBIT.
T_GND (V96W3520C)	DF	compool	--	--	uplink	sec	Uplinked time tag of Orbiter state vector.
THETA	DF	local	--	--	--	rad	Difference of eccentric anomaly.

VARIABLES LIST

Variable name (M/S ID)	Preci- sion & type	Compool or local	Initial- ization category	Initial value	Uplink/ downlist	Units	Description
THETA_COR	DF	local	--	--	--	rad	Correction to THETA in the solution of Kepler's equation, used in F and G.
T_IN	DF	local	--	--	--	sec	Initial time input for state propagation, used in SUPER_G.
T_LAST_FILT (V90W1252C)	DF	compool	--	--	--	sec	Time tag of V _LAST_FILT and of filter state at last navigation cycle.
T_LAST_FILT_TLM (V90W4285C)	DF	compool	--	--	downlist	sec	Time tag of Orbiter and target state vector at last navigation cycle (for TLM)
T_ORB_STATE_UPDATE (V94W3727C)	DF	compool	MD	--	downlist	sec	Time tag of most recent Orbiter state vector update (for TLM).
TCT_ACC (V90A4874C-76C)	DF(3)	compool	--	--	downlist	ft/sec^2	Orbiter M1950 total acceleration vector.
TOT_ACC_LAST (V90A1[illegible]2C -4C)	DF(3)	compool	--	--	--	ft/sec^2	Value of TOT_ACC at the end of the previous cycle.
T_PRED_FINAL (V94W3979C)	DF	compool	--	--	downlist	sec	Time tag for prediction (associated with R _PRED_FINAL and V _PRED_FINAL).
T_PRED_INIT (V94W3970C)	DF	compool	--	--	downlist	sec	Time tag for prediction (associated with R _PRED_INIT and V _PRED_INIT).
T_REND_RADAR (V90W4841C)	DF	compool	--	--	downlist	sec	Snapped time tag of the rendezvous radar measurement (RR SOP HAL name RR_TIM, M/S ID # V95W4292C).
T_RESET (V90W0225C)	DF	compool	--	--	--	sec	Time associated with reserved reset state.
T_RESID	DF	local	--	--	--	sec	Time tag of interpolated state vector.
TRG_TRK_MODE (V90U4836C)	B	compool	--	--	downlist	--	Snapped flag indicating (ON) that the star tracker is in the target tracking mode (STAR TRCK SOP HAL name NAV_TARGET, M/S ID # V95X4778X).

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
TRUN	SF	local	--	--	--	rad	Estimate of the rendezvous radar trunnion angle measurement.
T_STAR_TRACKER (V90U4837C)	DF	compool	--	--	downlist	sec	Snapped variable for the star tracker measurement time tag (STAR TRCK SOP HAL name TIME, M/S ID # V95W4760C).
T_ST_LAST	DF	local	IV	0.	--	sec	Time of last star tracker measurement.
T_TV (V90W0315C)	DF	compool	MD	--	downlist	sec	Target vehicle state vector time tag.
T_TV_GND (V96W1283C)	DF	compool	--	--	uplink	sec	Uplinked target state time tag.
T_TV_STATE_UPDATE (V90W4939C)	DF	compool	MD	--	downlist	sec	Time tag of most recent target state vector update (for TLM).
TV_PREDICT_FAIL (V90X4769XA)	B	compool	--	--	downlist	--	Flag which indicates that the time tolerance MAX_TIME_TOL has been exceeded and consequently no target state vector prediction will take place.
U _M	SF(3)	compool	--	--	--	feet	Line of sight vector in sensor system.
UNMOD_ACC_BIAS (V90U1265C -7C)	SF(3)	compool	--	--	--	ft/sec^2	Unmodeled acceleration bias.
UNMOD_ACC_BIAS_TLM (V90U4301C-03C)	SF(3)	compool	--	--	downlist	ft/sec^2	Unmodelled acceleration bias for TLM.
UNMOD_ACC_UPDATE_FLAG (V96U9190C)	B	local	MD	--	--	--	Flag indicating (ON) that the unmodeled acceleration bias states are to be updated by the filter.
UR	DF(3)	local	--	--	--	--	Earth fixed unit vector used in ACCEL_ONORBIT drag model.
U _RDOT	SF(3)	local	--	--	--	1/sec	Local variable used in the calculation of the estimated range rate measurement.
UR_SUN	DF(3)	compool	--	--	--	--	Sun unit vector in M50 coordinates.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
USE_IMU_DATA (V90X0538X)	B	compool	IV	OFF	downlist	--	Flag indicating (ON) the usage of IMU data by the powered flight propagator.
V	DF(3)	local	—	--	—	ft/sec	Vehicle M50 velocity used in ACCEL_ONORBIT.
VAR	SF	compool	—	--	—	vary	Copy of variance associated with measurement currently being processed.
VAR_COAS_ANGLES (V96U9191C-2C)	SF(2)	local	DD	4.E-6, 4.E-6	—	rad^2	COAS angles measurement bias variables used to initialize the covariance matrix.
VAR_CCAS_HORIZ (V96U9193C)	SF	local	DD	4.E-8	—	rad^2	Variance of the COAS horizontal measurement.
VAR_COAS_VERT (V96U9194C)	SF	local	DD	4.E-8	—	rad^2	Variance of the COAS vertical measurement.
VAR_HORIZ	SF	local	—	—	—	rad^2	Variance of the horizontal measurement sensor.
VAR_IMU_ALIGN (V96U9195C)	SF	local	DD	.1	--	rad^2	Variance of platform misalinement added as process noise in the covariance matrix.
VAR_RANGE_DOT (V96U9196C)	SF	local	DD	1.0	—	ft^2/sec^2	Variance of the rendezvous radar range rate sensor measurement.
VAR_RR_ANGLES (V96U9197C-8C)	SF(2)	local	DD	3.E-4, 3.E-4	--	rad^2	Rendezvous radar angles measurement bias variances used to initialize the covariance matrix.
VAR_RRDOT (V96U9199C-200C)	SF(2)	local	DD	1.0,1.0	—	feet^2, feet^2/sec^2	Rendezvous radar range and range rate measurement bias variance used to initialize the covariance matrix.
VAR_RR_RNG_MIN (V96U9201C)	SF	local	DD	711.	—	feet^2	Minimum value for computation of rendezvous radar range variance.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
VAR_SENS	SF(4)	compool	--	--	--	vary	General measurement bias filter variance used in propagation of sensor biases and in adding process noise.
VAR_SHAFT (V96U9202C)	SF	local	DD	1.1 E-5	--	rad^2	Variance of the rendezvous radar shaft angle measurement.
VAR_ST_ANGLES (V96U9203C-4C)	SF(2)	local	DD	1.2E-6, 1.2E-6	--	rad^2	Star tracker angles measurement bias variances used to initialize the covariance matrix.
VAR_ST_HORIZ (V96U9205C)	SF	local	DD	1.0 E-8	--	rad^2	Variance of star tracker horizontal angle measurement.
VAR_ST_VERT (V96U9206C)	SF	local	DD	1.0 E-8	--	rad^2	Variance of star tracker vertical angle measurement.
VAR_TRUN (V96U9207C)	SF	local	DD	1.1 E-5	--	rad^2	Variance of the rendezvous radar trunnion angle measurement.
VAR_U_A_COAST (V96U9208C-10C)	SF(3)	compool	DD	3*3.E-10	--	ft^2/sec^4	Variance of unmodeled accelerations during coasting flight.
VAR_U_A_PWRD_FLT (V96U9211C-13C)	SF(3)	local	DD	3*E-2	--	ft^2/sec^4	Variance of unmodeled accelerations during powered flight.
VAR_UNMOD_ACC (V90A4933C-35C)	SF(3)	compool	--	--	--	ft^2/sec^4	Variance of unmodeled accelerations.
VAR_VERT	SF	local	--	--	--	rad^2	Variance of the vertical measurement sensor.
V_CURRENT_FILT (V90L8927C -9C)	DF(3)	compool	--	--	downlist	ft/sec	Snapped variable for the total accumulated IMU sensed velocity (IMU RM HAL name V_IMU_CURRENT, M/S ID # V90L2557C-9C).
VEH_MASS	SF	local	--	--	--	slugs	Vehicle mass used in ACCEL_ONORBIT drag model.
VENT	SF(3)	local	--	--	--	ft/sec^2	Vent/RCS thrust acceleration used in ACCEL_ONORBIT.
VENT_SS (V90U4964C-66C)	SF(3)	compool	--	--	downlist	ft/sec^2	Orbiter vent vector for downlist.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
VERT	SF	local	--	—	—	rad	Filter estimated vertical angle measurement.
V _FILT (V90L0624C -6C)	DP(3)	compool	--	—	—	ft/sec	Orbiter M1950 velocity vector.
V _FILT_INIT (V90L1279C -81C)	DP(3)	compool	OPS	--	--	ft/sec	Orbiter velocity vector saved across memory reconfiguration and used for navigation initialization.
V _FILT_TLM (V90L4281C-83C)	DP(3)	compool	--	--	downlist	ft/sec	Orbiter velocity vector for TLM (M50).
V _FIN	DP(3)	local	—	--	--	ft/sec	Orbiter or target velocity vector at T_FIN, used in SUPER_G.
VFL_OV	B	compool	HC	1	—	--	Flag indicating activation (1) or deactivation (0) of Vent/RCS thrust acceleration models for Orbiter propagation.
VFLOV_PRED	B	compool	HC	0	—	--	Flag indicating activation (1) or deactivation (0) of Vent/ RCS thrust acceleration model for Orbiter prediction.
VFL_TV	B	compool	HC	1	—	--	Flag setting that specifies use of unmodeled acceleration bias for target vehicle propagation.
VFLTV_PRED	B	compool	HC	0	--	--	Flag setting that specifies use of unmodeled acceleration bias for target vehicle prediction.
VFORCE (V96U8174C-76C) (V90U4956C-58C)	SF(3)	compool	MD	--	Uplink, downlist	lbs.	Vent/RCS thrust force vector in body coordinates used in ACCEL_ONORBIT.
V _GND (V96L3510C-12C)	DP(3)	compool	--	—	uplink	ft/sec	Uplinked Orbiter velocity vector.
V _IMU_DIF	DP(3)	local	—	--	--	ft/sec	Difference in current and past accumulated sensed IMU velocities used in state vector interpolation.
V _IMU_RESET (V90L0247C -9C)	DP(3)	compool	--	—	downlist	ft/sec	Copy of V_LAST_FILT.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
V _IN	DF(3)	local	--	--	--	ft/sec	Orbiter or target velocity vector at T_IN used in F & G or SUPER_G.
V _LAST (V90L4785C -7C)	DF(3)	compool	--	--	--	ft/sec	Velocity vector of the Shuttle at the end of the last filter subcycle.
V _LAST_FILT (V90L0621C -3C)	DF(3)	compool	--	--	--	ft/sec	Total accumulated IMU sensed velocity.
VM	B	local	--	--	--	--	Flag used in ACCEL_ONORBIT to indicate if Vent/RCS model is to be used.
VMP (V94X3971X)	B	compool	--	--	--	--	Flag indicating use or nonuse of the vent model in prediction.
V _ONE	DF(3)	local	--	--	--	ft/sec	Velocity vector at the beginning of a time interval used to generate a transition matrix.
V _PRED_FINAL (V95L0814C -6C)	DF(3)	compool	--	--	downlist	ft/sec	Velocity vector as output from precision state predictor.
V _PRED_INIT (V94L4006C -8C)	DF(3)	compool	--	--	downlist	ft/sec	Initial velocity vector for prediction.
V _R	SF(3)	local	--	--	--	ft/sec	Velocity of vehicle relative to atmosphere used in ACCEL_ ONORBIT drag model.
V _REL_BODY	SF(3)	local	--	--	--	ft/sec	Orbiter velocity relative to the atmosphere.
V _RESET (V90L0239C -41C)	DF(3)	compool	--	--	--	ft/sec	Orbiter vehicle velocity vector after all navigation updates reserved for reset of user parameter state propagator velocity vector.
V _RESID	DF(3)	local	--	--	--	ft/sec	M1950 velocity vector interpolated to the measurement time.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
V _RHO	SF(3)	compool	—	—	—	ft/sec	Relative M1950 velocity vector.
V _TV (V90L0310C -12C)	DF(3)	compool	MD	—	—	ft/sec	Target vehicle M1950 velocity vector.
V _TV_GND (V96L1280C -2C)	DF(3)	compool	—	—	uplink	ft/sec	Uplinked target velocity vector.
V _TV_LAST (V90L4792C -4C)	DF(3)	compool	—	—	—	ft/sec	Last value of the target vehicle velocity vector.
V _TV_RESET (V90L1380C -2C)	DF(3)	compool	—	—	—	ft/sec	Target vehicle velocity vector after all navigation updates, reserved for reset of user parameter state propagator velocity vector.
V _TV_RESID	DF(3)	local	—	—	—	ft/sec	Velocity vector of the target vehicle at measurement time.
V _TV_TLM (V90L4291C -93C)	DF(3)	compool	—	—	downlist	ft/sec	Target velocity vector for TLM (M50).
V _TWO	DF(3)	local	—	—	—	ft/sec	Velocity vector at the end of a time interval used to generate a transition matrix.
WT_DISP (V93G6940C)	SF	compool	OPS	—	—	lbs	Displayed weight transferred from OPS 1 to OPS 2, or OPS 3 to OPS 2.
X	DF(6)	local	—	—	—	ft,ft,ft, ft/sec, ft/sec, ft/sec	Temporary array for the Orbiter or target state vector, used in Pines method.
XN	DF(7)	local	—	—	—	ft,ft,ft, ft/sec, ft/sec, ft/sec, sec	Array of integrated initial conditions for onorbit prediction and propagation.

VARIABLES LIST

Variable name (M/S ID)	Preci- sion & type	Compool or local	Initial- ization category	Initial value	Uplink/ downlist	Units	Description
ZETA_IMAG	DF(5)	local	--	--	--	--	Coefficients used in ACCEL_ONORBIT gravitational acceleration which are longitude dependent.
ZETA_REAL	DF(5)	local	--	--	--	--	Coefficients used in ACCEL_ONORBIT gravitational acceleration which are longitude dependent.
ZONAL	DF(4)	local	C	--	--	--	Zonal harmonics coefficients used in ACCEL_ONORBIT gravitational acceleration.

APPENDIX B

NAVIGATION SEQUENCER PRINCIPAL FUNCTION
AND NAVIGATION PROCESSING PRINCIPAL FUNCTION
FLOWCHARTS

CONTENTS

Figure Page

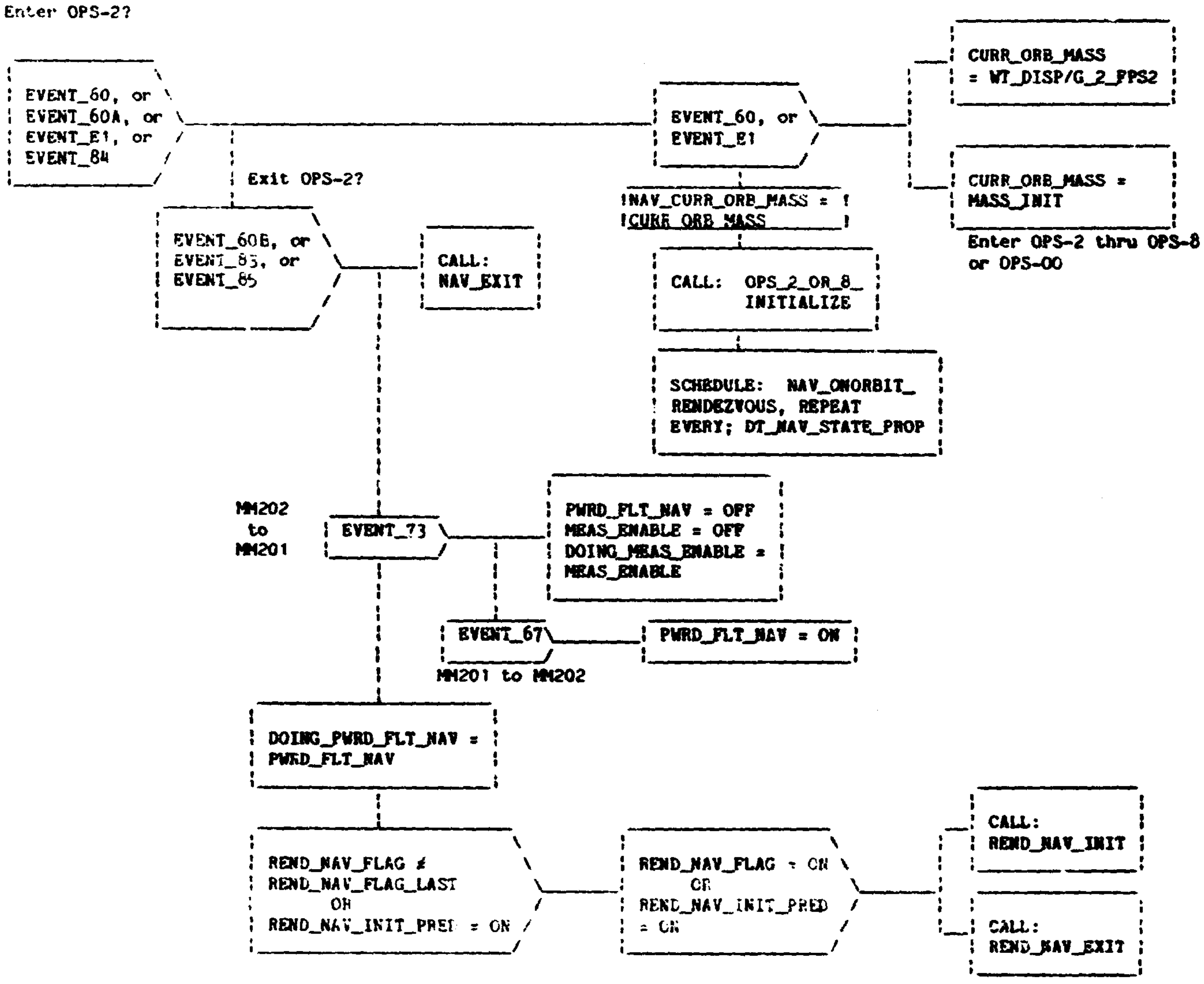


Figure B-1.- ONORBIT_REND_NAV_SEQUENCER (OPS-2).

ORIGINAL PAGE IS
OF POOR QUALITY

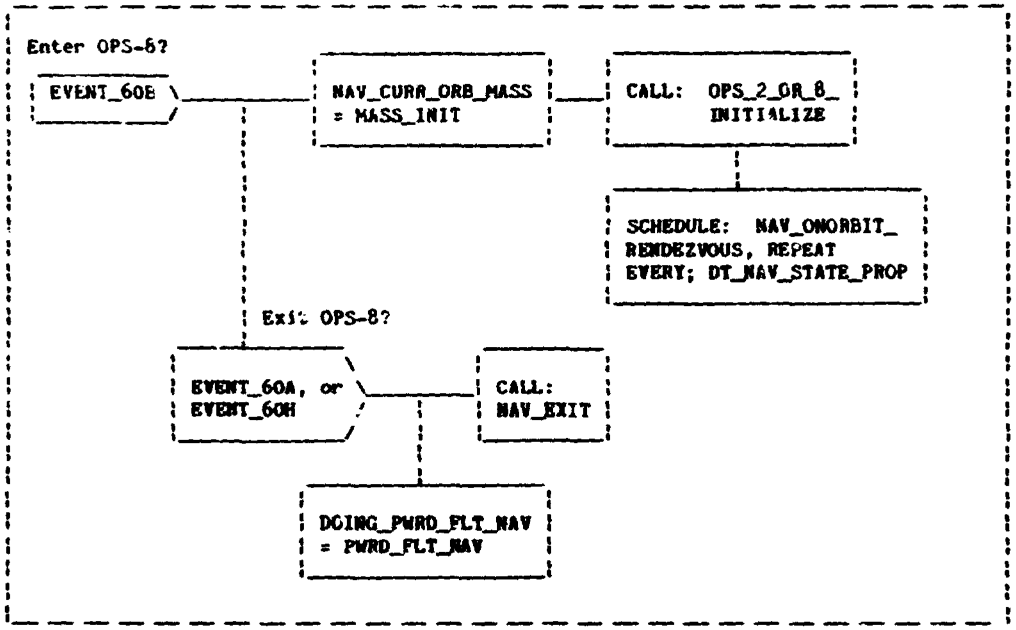


Figure B-2.- ONORBIT_REND_NAV_SEQUENCER (OPS-8).

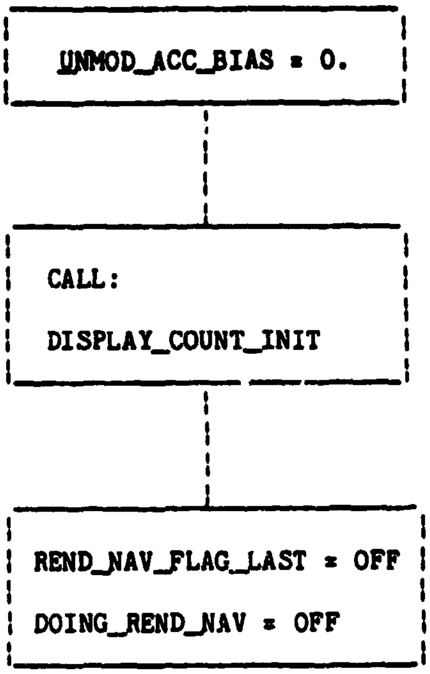
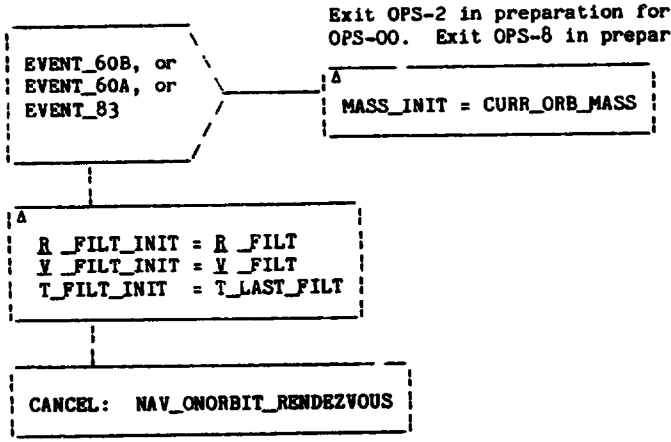


Figure B-3.- REND_NAV_EXIT.

Exit OPS-2 or OPS-8?



Exit OPS-2 in preparation for transition to OPS-8, or OPS-00. Exit OPS-8 in preparation to transition to OPS-2

ΔThese parameters, in addition to landing site and tacan table data, must be saved across the memory transitions.

Figure B-4.- NAV_EXIT

```
SNAP (V_CURRENT_FILT, T_CURRENT_FILT)

V_LAST_FILT = V_CURRENT_FILT
T_LAST_FILT = T_CURRENT_FILT

UNMOD_ACC_BIAS = 0.
SQR_EMU = SQRT(EARTH_MU)

PRED_ORB_CD = REF_ORB_CD
PRED_ORB_AREA = REF_ORB_AREA
PRED_ORB_MASS = NAV_CURR_ORB_MASS
GMOP = GM_ORD
GMDP = GM_DEG
DMP = DFL
VMP = VFLOV_PRED
ATMP = 1
PRED_STEP = PRED_STEP_OPS_INIT
R_PRED_INIT = R_FILT_INIT
V_PRED_INIT = V_FILT_INIT
T_PRED_INIT = T_FILT_INIT
T_PRED_FINAL = T_LAST_FILT

CALL: ONORBIT_PREDICT

R_FILT = R_PRED_FINAL
V_FILT = V_PRED_FINAL

(CONT'D)
```

Figure B-5.- OPS_2_OR_8_INITIALIZE (Sheet 1 of 2).

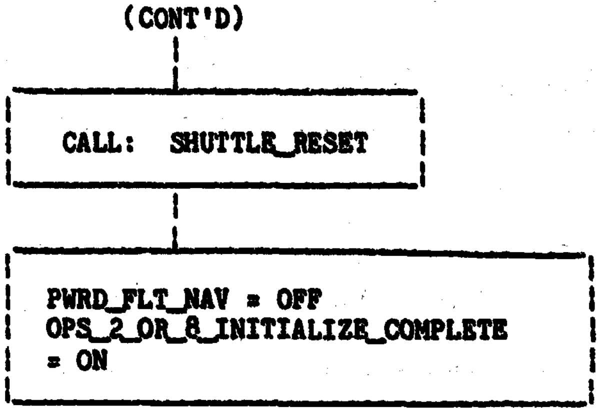


Figure B-5.- OPS_2_OR_8_INITIALIZE (Sheet 2 of 2).

```
R_RESET = R_FILT
V_RESET = V_FILT
T_RESET = T_LAST_FILT
V_IMU_RESET = V_LAST_FILT
FILT_UPDATE = ON
```

Figure B-6.- SHUTTLE_RESET.

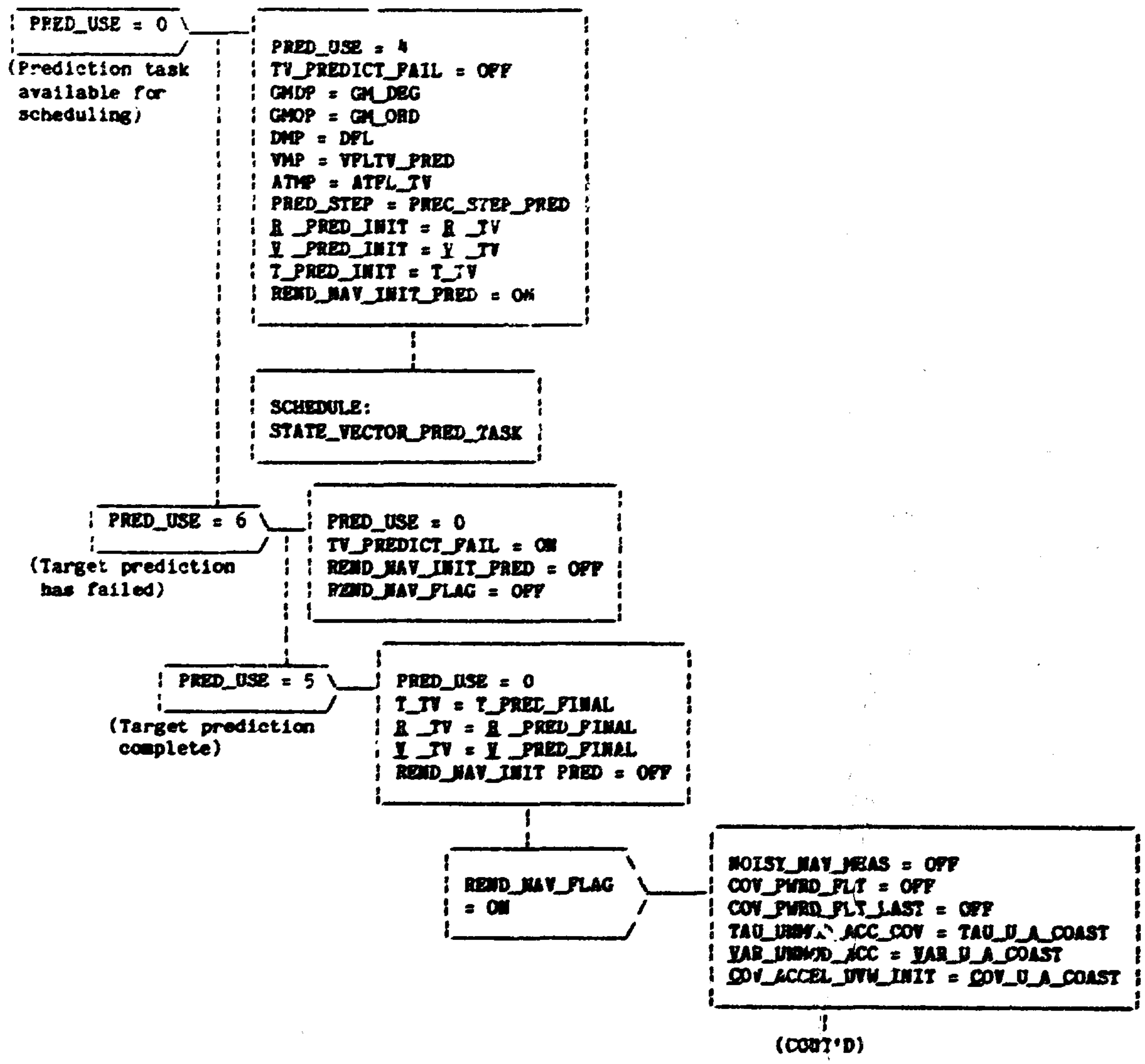


Figure B-7.- REND_NAV_INIT (Sheet 1 of 2).

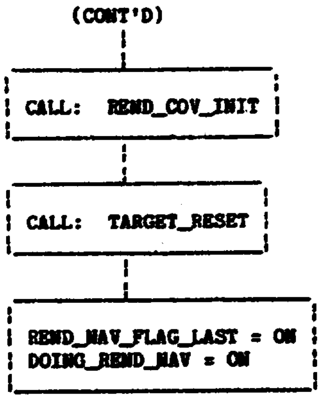


Figure B-7.- REND_NAV_INIT (Sheet 2 of 2).

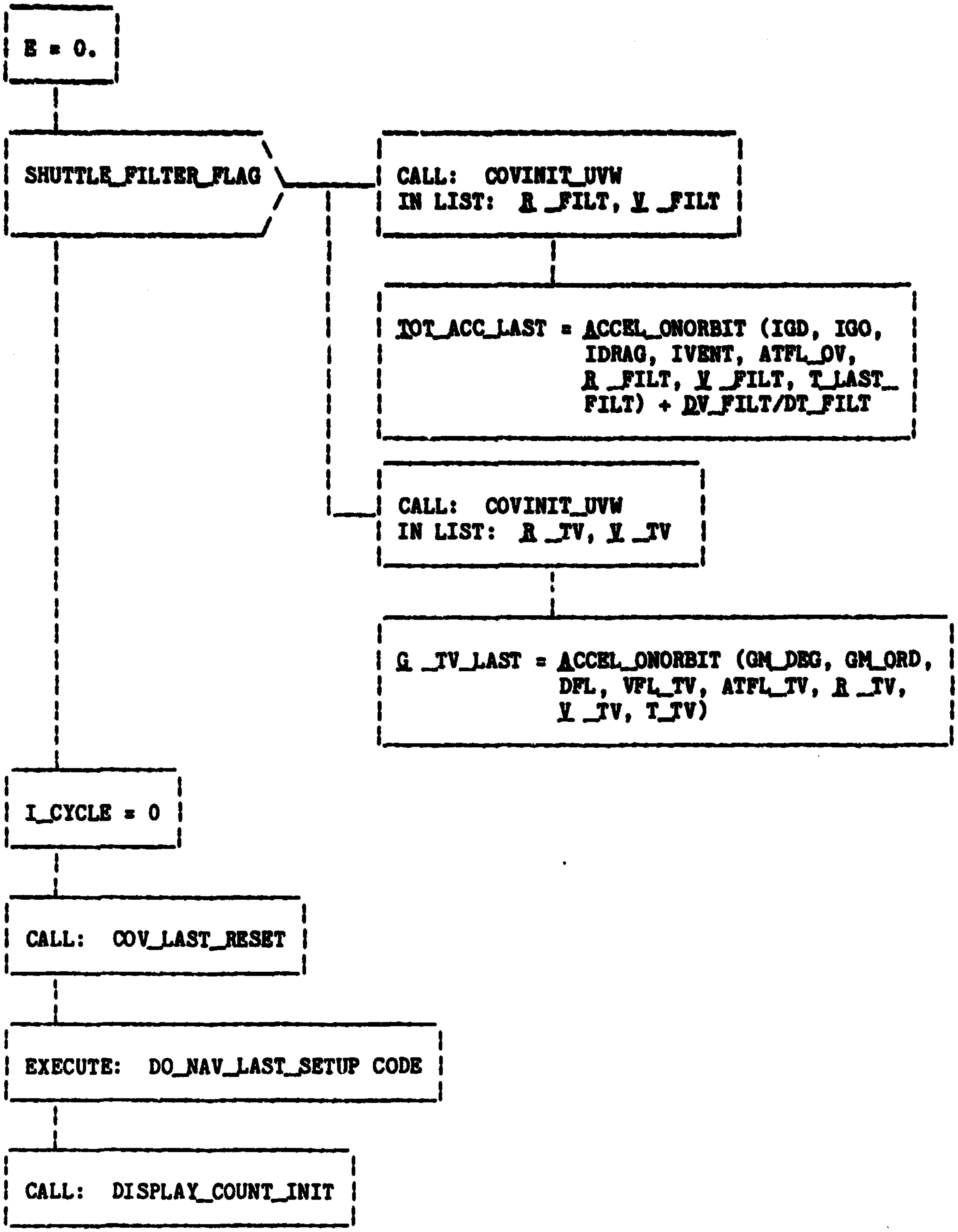


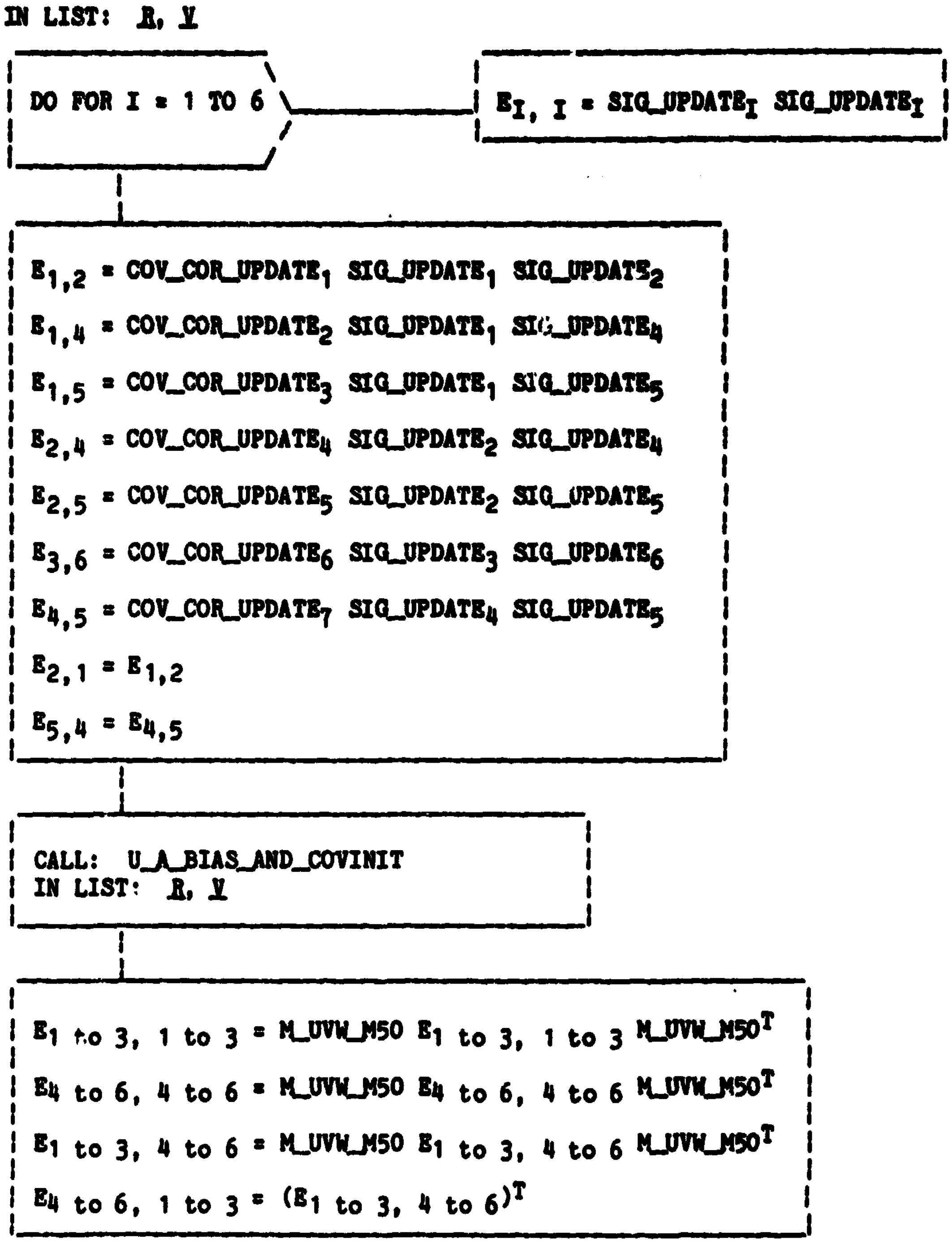
Figure B-8.- REND_COV_INIT.

```
R_LAST = R_FILT
V_LAST = V_FILT
R_TV_LAST = R_TV
V_TV_LAST = V_TV
DV_COV = 0.
T_COV_LAST = T_LAST_FILT
```

Figure B-9.- COV_LAST_RESET.

```
DO_RR_ANGLES_NAV_LAST = OFF

DO_RRDOT_NAV_LAST = OFF

DO_ST_ANGLES_NAV_LAST = OFF

DO_COAS_ANGLES_NAV_LAST = OFF
```

Figure B-10.- DO_NAV_LAST_SETUP CODE.



IN LIST: $\underline{R}$, $\underline{V}$

DO FOR I = 1 TO 6

$E_{I,\,I} = \text{SIG\_UPDATE}_I\ \text{SIG\_UPDATE}_I$

$E_{1,2} = \text{COV\_COR\_UPDATE}_1\ \text{SIG\_UPDATE}_1\ \text{SIG\_UPDATE}_2$

$E_{1,4} = \text{COV\_COR\_UPDATE}_2\ \text{SIG\_UPDATE}_1\ \text{SIG\_UPDATE}_4$

$E_{1,5} = \text{COV\_COR\_UPDATE}_3\ \text{SIG\_UPDATE}_1\ \text{SIG\_UPDATE}_5$

$E_{2,4} = \text{COV\_COR\_UPDATE}_4\ \text{SIG\_UPDATE}_2\ \text{SIG\_UPDATE}_4$

$E_{2,5} = \text{COV\_COR\_UPDATE}_5\ \text{SIG\_UPDATE}_2\ \text{SIG\_UPDATE}_5$

$E_{3,6} = \text{COV\_COR\_UPDATE}_6\ \text{SIG\_UPDATE}_3\ \text{SIG\_UPDATE}_6$

$E_{4,5} = \text{COV\_COR\_UPDATE}_7\ \text{SIG\_UPDATE}_4\ \text{SIG\_UPDATE}_5$

$E_{2,1} = E_{1,2}$

$E_{5,4} = E_{4,5}$

CALL: U_A_BIAS_AND_COVINIT
IN LIST: $\underline{R}$, $\underline{V}$

$E_{1\text{ to }3,\ 1\text{ to }3} = \text{M\_UVW\_M50}\ E_{1\text{ to }3,\ 1\text{ to }3}\ \text{M\_UVW\_M50}^T$

$E_{4\text{ to }6,\ 4\text{ to }6} = \text{M\_UVW\_M50}\ E_{4\text{ to }6,\ 4\text{ to }6}\ \text{M\_UVW\_M50}^T$

$E_{1\text{ to }3,\ 4\text{ to }6} = \text{M\_UVW\_M50}\ E_{1\text{ to }3,\ 4\text{ to }6}\ \text{M\_UVW\_M50}^T$

$E_{4\text{ to }6,\ 1\text{ to }3} = (E_{1\text{ to }3,\ 4\text{ to }6})^T$

Figure B-11.- COVINIT_UVW.

```
N _ACCEPT = 0
N _REJECT = 0
SEQ_ACCEPT = 0
SEQ_REJECT = 0
```

Figure B-12.- DISPLAY_COUNT_INIT.

```
R _TV_RESET = R _TV
V _TV_RESET = V _TV
```

Figure B-13.- TARGET_RESET.

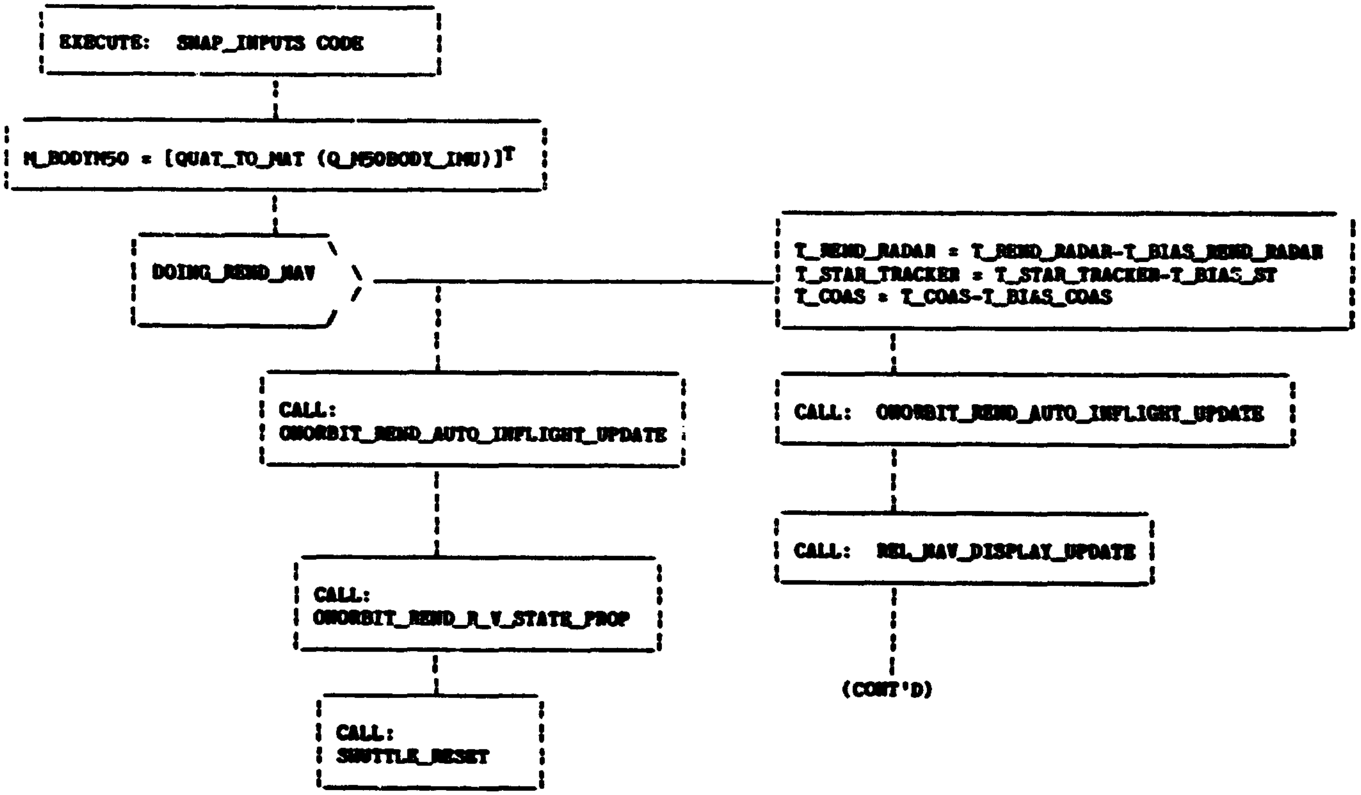


Figure B-14.- NAV_ONORBIT_RENDEZVOUS (Sheet 1 of 4).

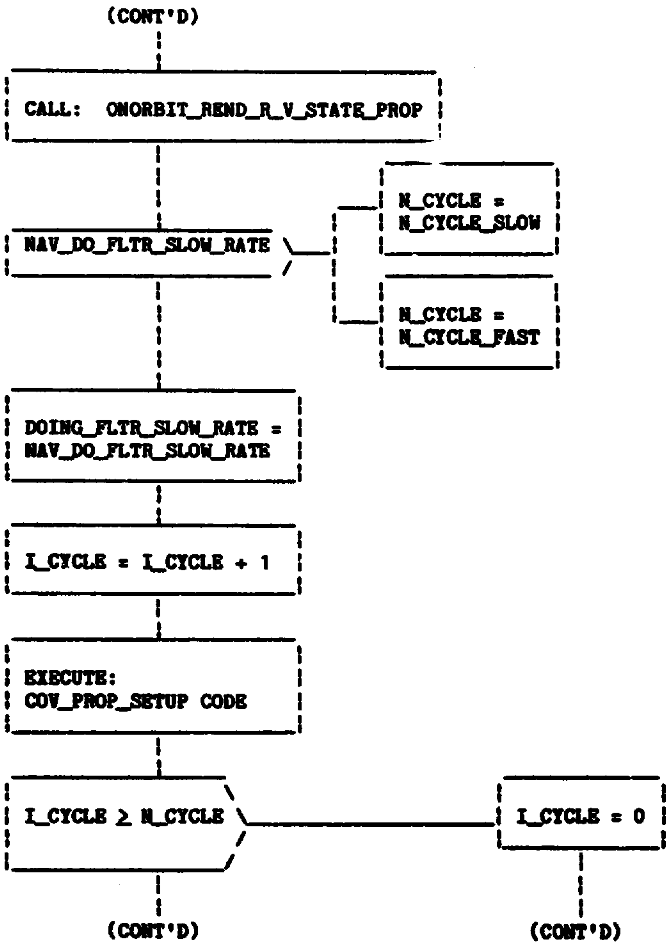


Figure B-14.- NAV_ONORBIT_RENDEZVOUS (Sheet 2 of 4).

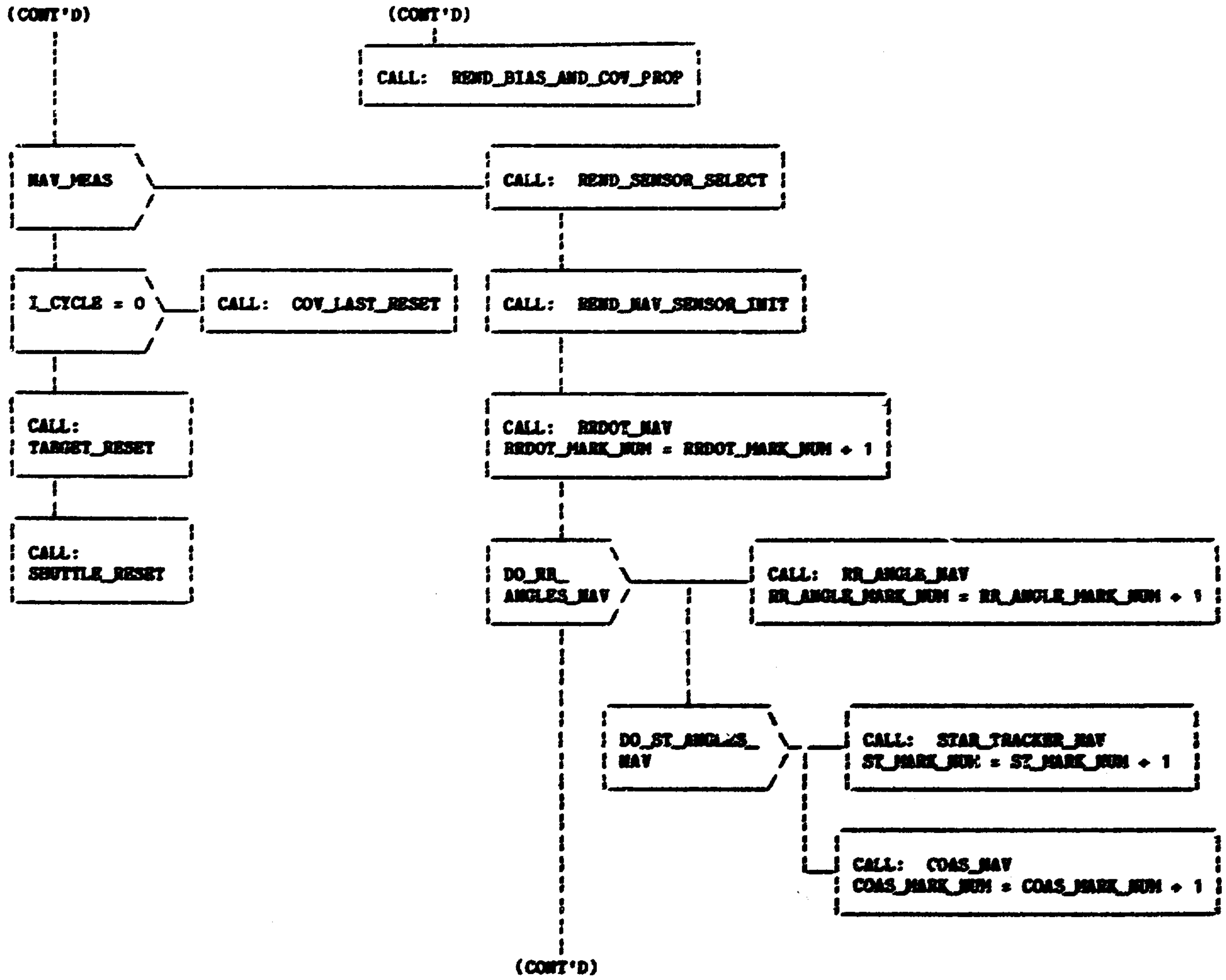


Figure B-14.- NAV_ONORBIT_RENDEZVOUS (Sheet 3 of 4)

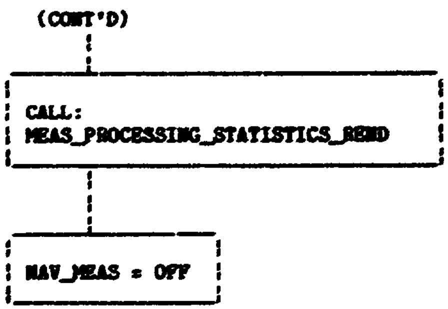
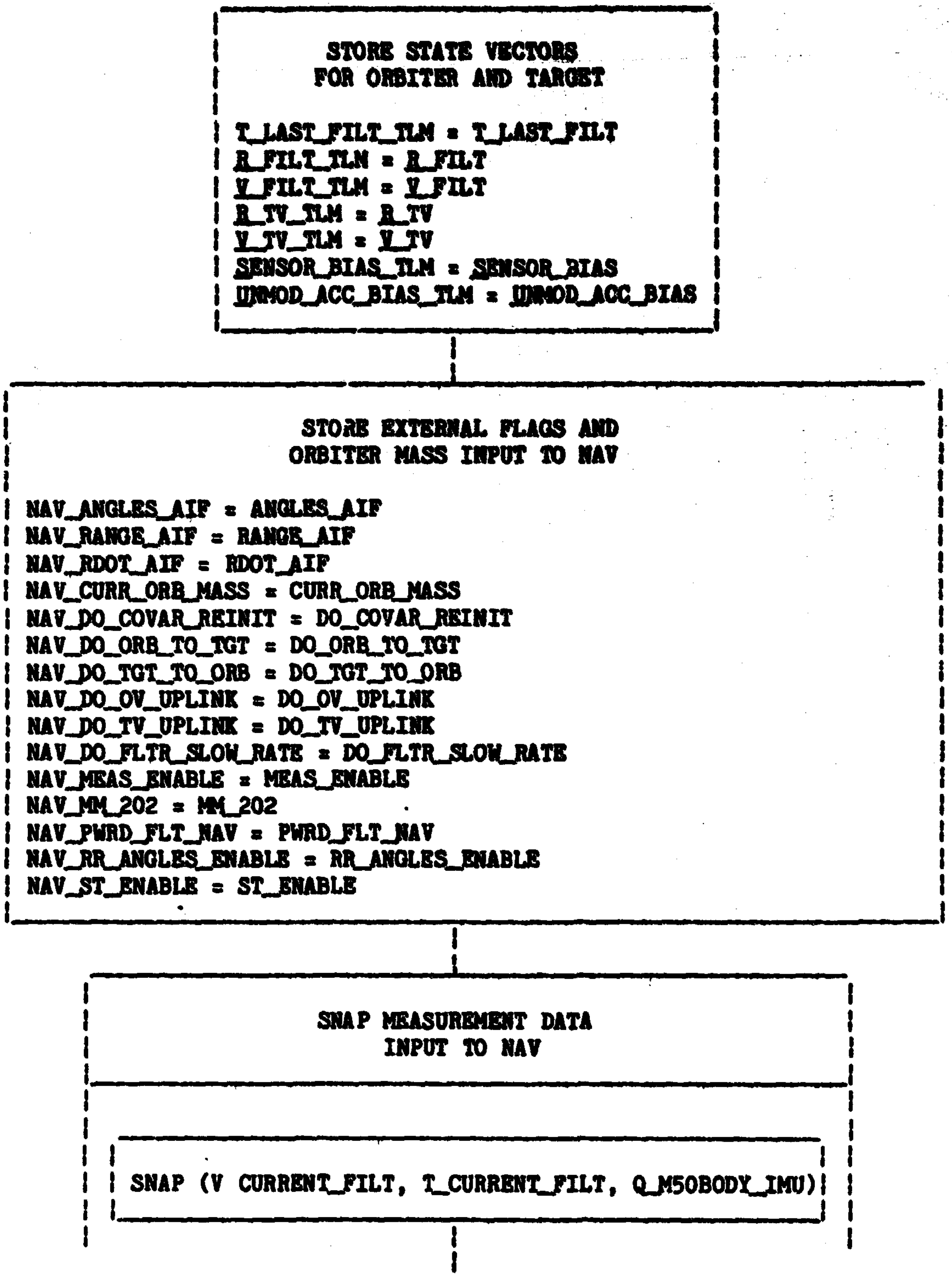


Figure B-14.- NAV_ONORBIT_RENDEZVOUS (Sheet 4 of 4).

SNAP_INPUTS CODE



STORE STATE VECTORS
FOR ORBITER AND TARGET

T_LAST_FILT_TLM = T_LAST_FILT
R_FILT_TLM = R_FILT
V_FILT_TLM = V_FILT
R_TV_TLM = R_TV
V_TV_TLM = V_TV
SENSOR_BIAS_TLM = SENSOR_BIAS
UNMOD_ACC_BIAS_TLM = UNMOD_ACC_BIAS

STORE EXTERNAL FLAGS AND
ORBITER MASS INPUT TO NAV

NAV_ANGLES_AIF = ANGLES_AIF
NAV_RANGE_AIF = RANGE_AIF
NAV_RDOT_AIF = RDOT_AIF
NAV_CURR_ORB_MASS = CURR_ORB_MASS
NAV_DO_COVAR_REINIT = DO_COVAR_REINIT
NAV_DO_ORB_TO_TGT = DO_ORB_TO_TGT
NAV_DO_TGT_TO_ORB = DO_TGT_TO_ORB
NAV_DO_OV_UPLINK = DO_OV_UPLINK
NAV_DO_TV_UPLINK = DO_TV_UPLINK
NAV_DO_FLTR_SLOW_RATE = DO_FLTR_SLOW_RATE
NAV_MEAS_ENABLE = MEAS_ENABLE
NAV_MM_202 = MM_202
NAV_PWRD_FLT_NAV = PWRD_FLT_NAV
NAV_RR_ANGLES_ENABLE = RR_ANGLES_ENABLE
NAV_ST_ENABLE = ST_ENABLE

SNAP MEASUREMENT DATA
INPUT TO NAV

SNAP (V CURRENT_FILT, T_CURRENT_FILT, Q_M50BODY_IMU)

(CONT'D)

Figure B-14A.-SNAP_INPUTS CODE (Sheet 1 of 2).

(CONT'D)

```
SNAP REND_RADAR (Q_RR_SHFT, Q_RR_TRUN, Q_RR_RNG,
Q_RR_RNG_DOT, RNG_DATA_GOOD, RDOT_DATA_GOOD,
RR_ANGLE_DATA_GOOD;
Q_M50BODY__RR, T_REND_RADAR,
SELF_TEST_FLAG)
```

```
SNAP STAR_TRACKER (Q_ST_HORIZ, Q_ST_VERT,
ST_DATA_GOOD, M_M50_TO_ST,
T_STAR_TRACKER, TRG_TRK_MODE)
```

```
SNAP COAS (Q_COAS_HORIZ, Q_COAS_VERT,
COAS_DATA_GOOD, M_M50_TO_BODY_COAS,
COAS_ID, T_COAS)
```

Figure B-14A.- SNAP_INPUTS CODE (Sheet 2 of 2).

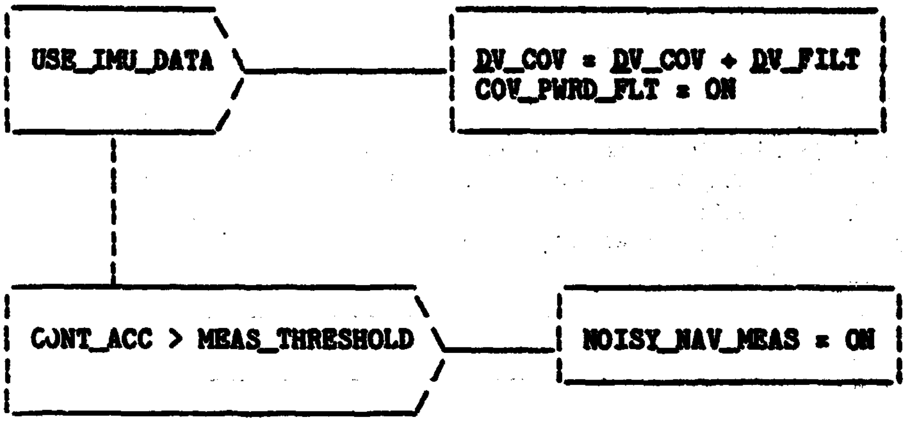


Figure B-15.- COV_PROP_SETUP CODE.

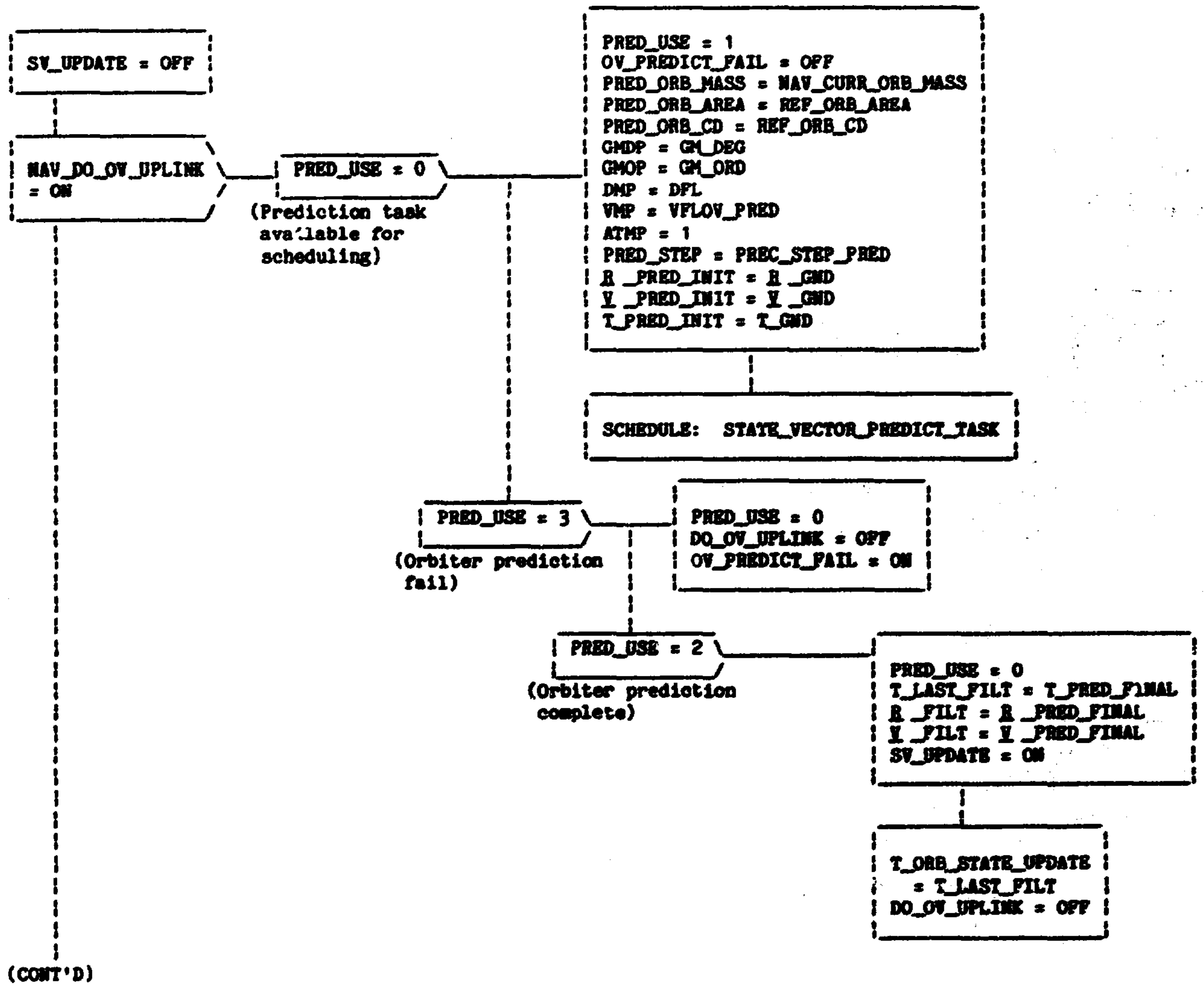


Figure B-16.- ONORBIT_REND_AUTO_INFLIGHT_UPDATE (Sheet 1 of 2).

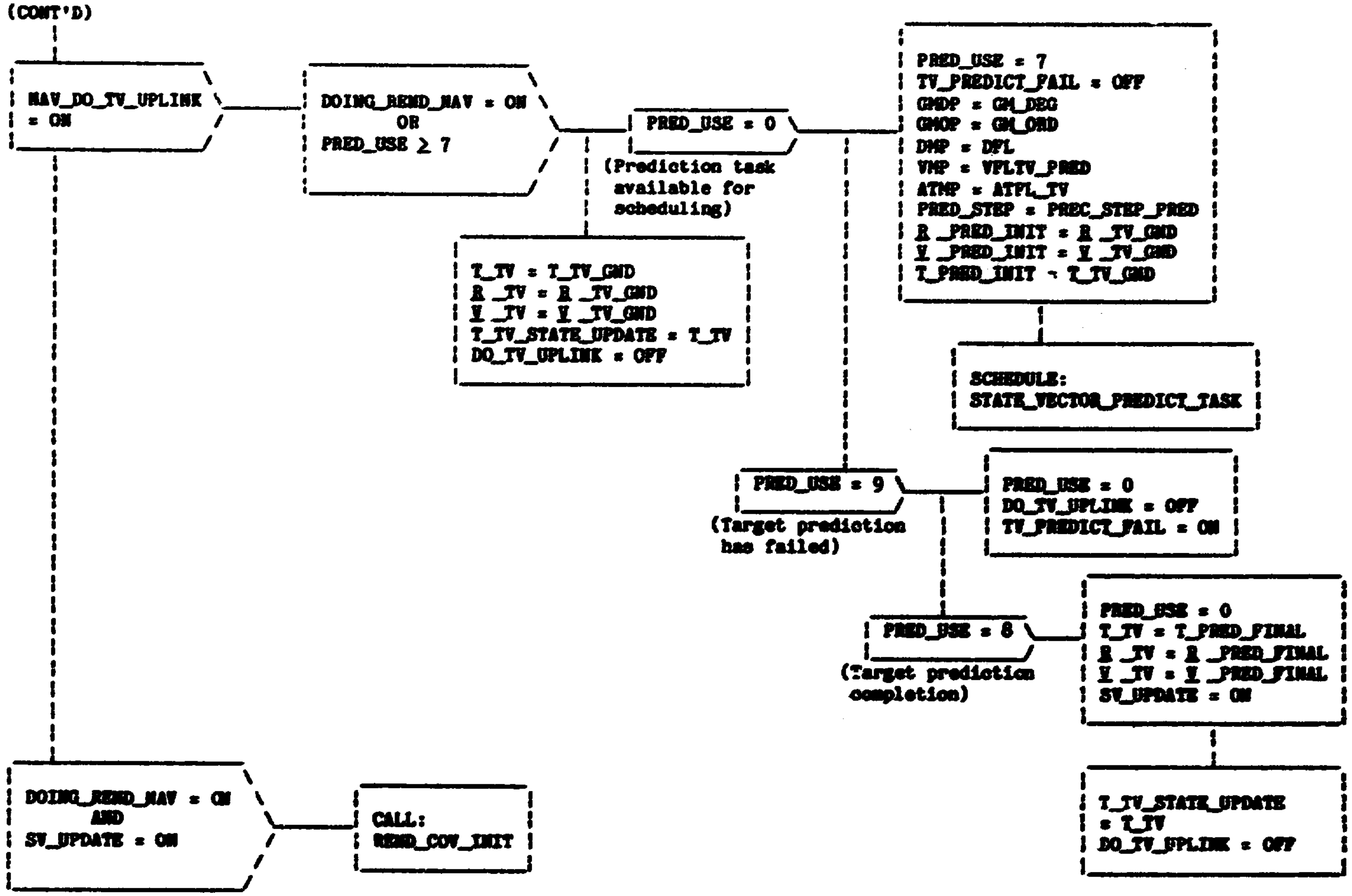


Figure B-16.- ONORBIT_REND_AUTO_INFLIGHT_UPDATE (Sheet 2 of 2).

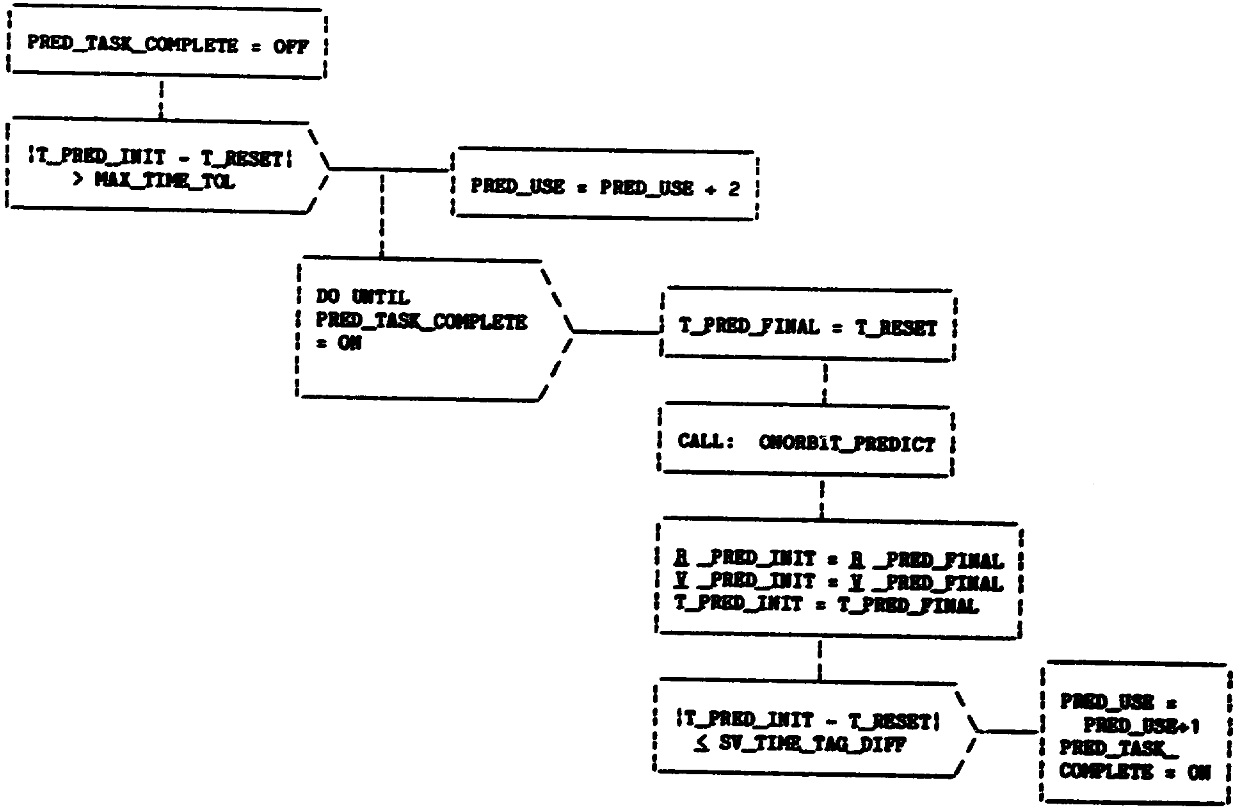
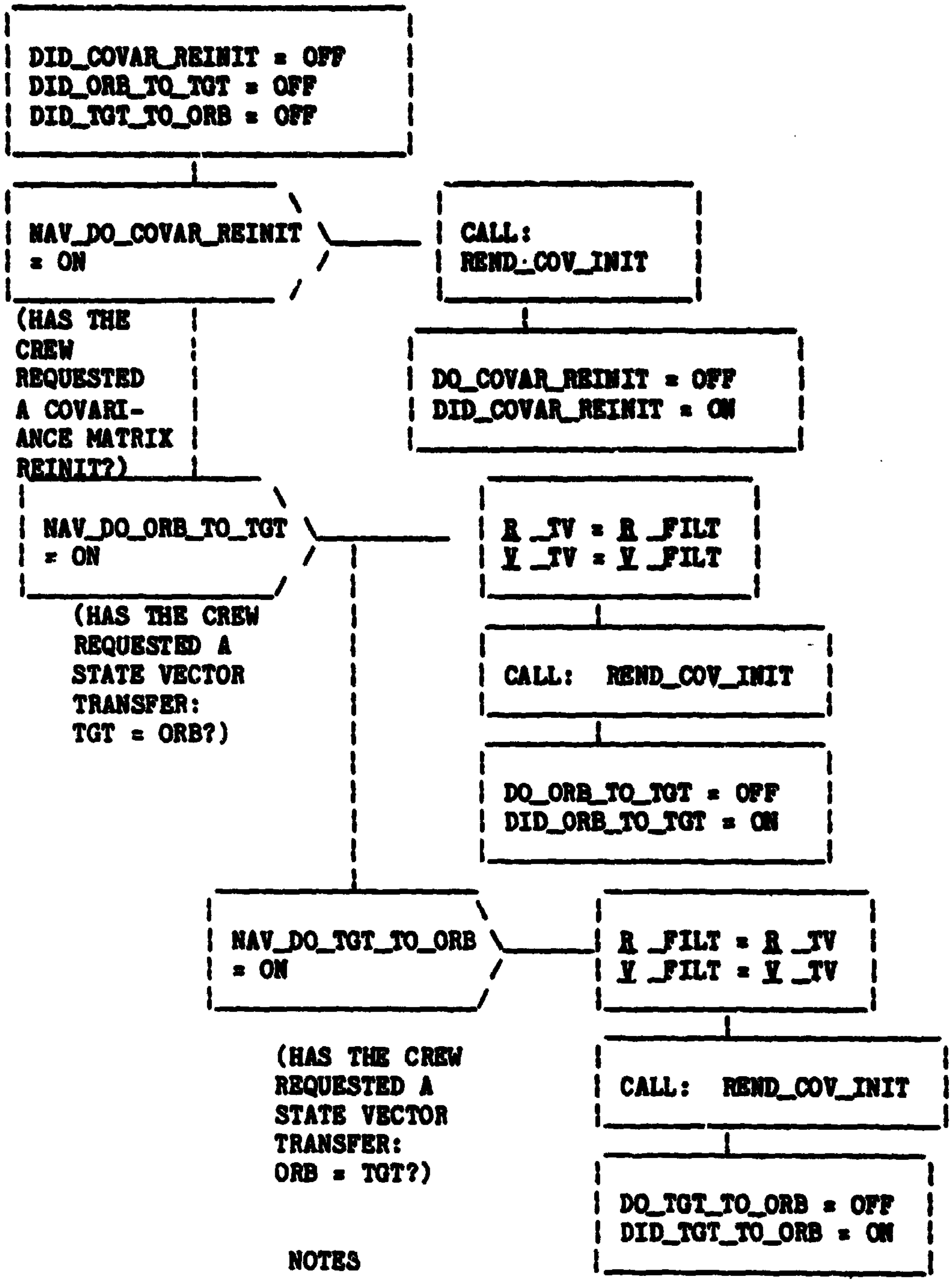


Figure B-17.- STATE_VECTOR_PREDICT_TASK



NOTES

1. THE DO_COVAR_REINIT FLAG MUST BE ACTIVATED ONLY DURING THE RENDEZVOUS NAVIGATION PHASE.

2. STATE VECTOR TRANSFER MUST BE ACTIVATED VIA CREW INPUT TO THE REL_NAV DISPLAY AND ONLY DURING THE RENDEZVOUS NAVIGATION PHASE.

Figure B-18.- REL_NAV_DISPLAY_UPDATES.

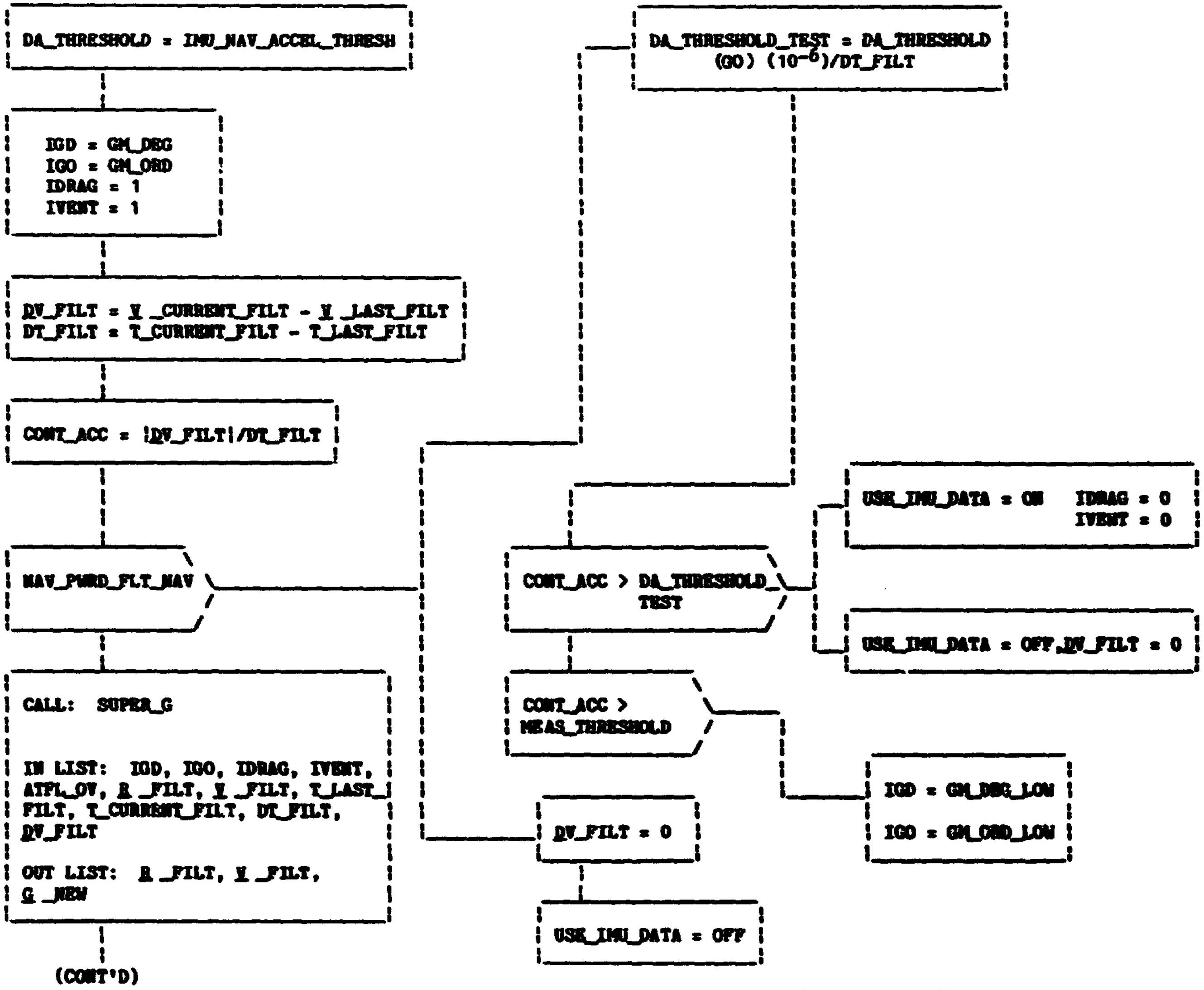
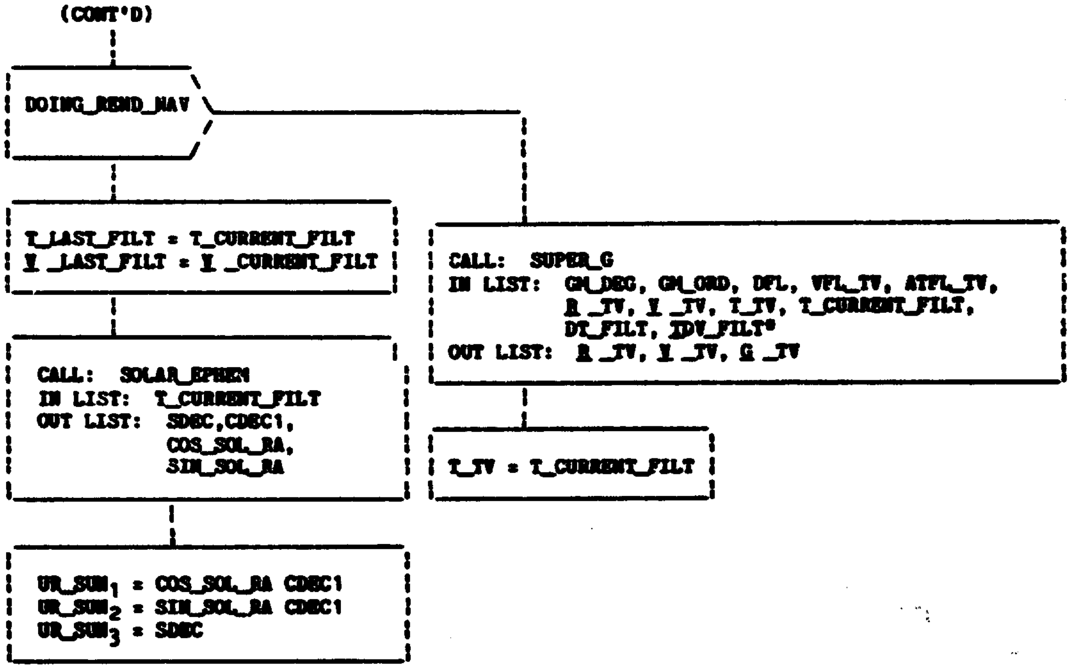


Figure B-19.- ONORBIT_REND_R_V_STATE_PROP (Sheet 1 of 2).



*IDV_FILT is a HC vector (3) 0, 0, 0.

Figure B-19.- ONORBIT_REND_R_V_STATE_PROP (Sheet 2 of 2).

IN LIST: GMD, GMO, DM, VM, ATM, $\underline{R}$_IN, $\underline{V}$_IN, T_IN, T_FIN, DT_FILT, $\underline{DV}$_IN

OUT LIST: $\underline{R}$_FIN, $\underline{V}$_FIN, $\underline{G}$_OUT

$$\underline{G}\text{\_INT} = \underline{\mathrm{A}}\text{CCEL\_ONORBIT (GMD, GMO, DM, VM, ATM, }\underline{R}\text{\_IN, }\underline{V}\text{\_IN, T\_IN)}$$

$$\underline{R}\text{\_FIN} = \underline{R}\text{\_IN} + \text{DT\_FILT}\,[\underline{V}\text{\_IN} + 0.5\,(\underline{DV}\text{\_IN} + \text{DT\_FILT}\ \underline{G}\text{\_INT})]$$

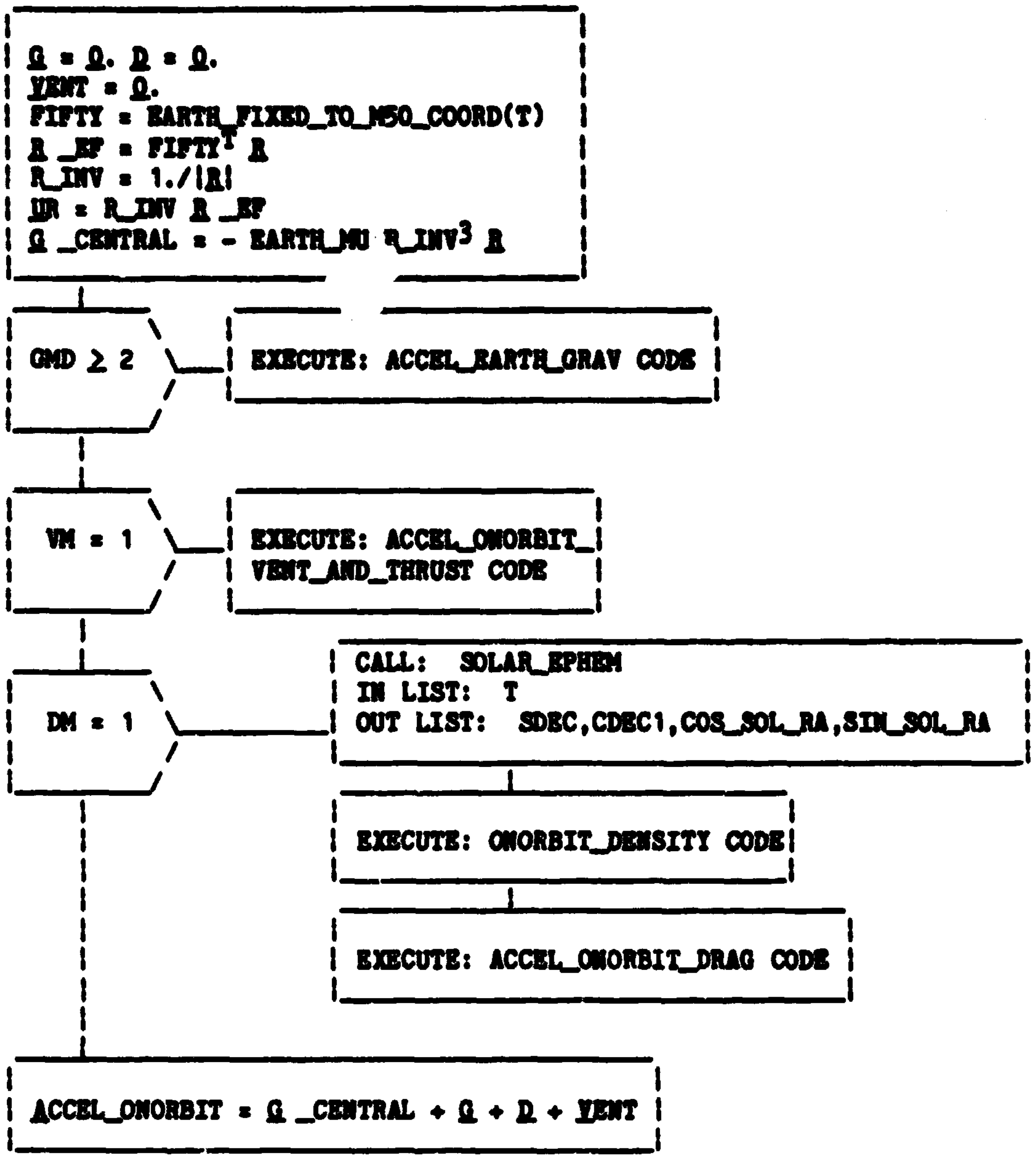
$$\underline{G}\text{\_OUT} = \underline{\mathrm{A}}\text{CCEL\_ONORBIT (GMD, GMO, DM, VM, ATM, }\underline{R}\text{\_FIN, }\underline{V}\text{\_IN, T\_FIN)}$$

$$\underline{V}\text{\_FIN} = \underline{V}\text{\_IN} + \underline{DV}\text{\_IN} + 0.5\ \text{DT\_FILT}\,(\underline{G}\text{\_INT} + \underline{G}\text{\_OUT})$$

$$\underline{R}\text{\_FIN} = \underline{R}\text{\_FIN} + [(\underline{G}\text{\_OUT} - \underline{G}\text{\_INT})\ \text{DT\_FILT}^2]/6.$$

Figure B-20.- SUPER_G.

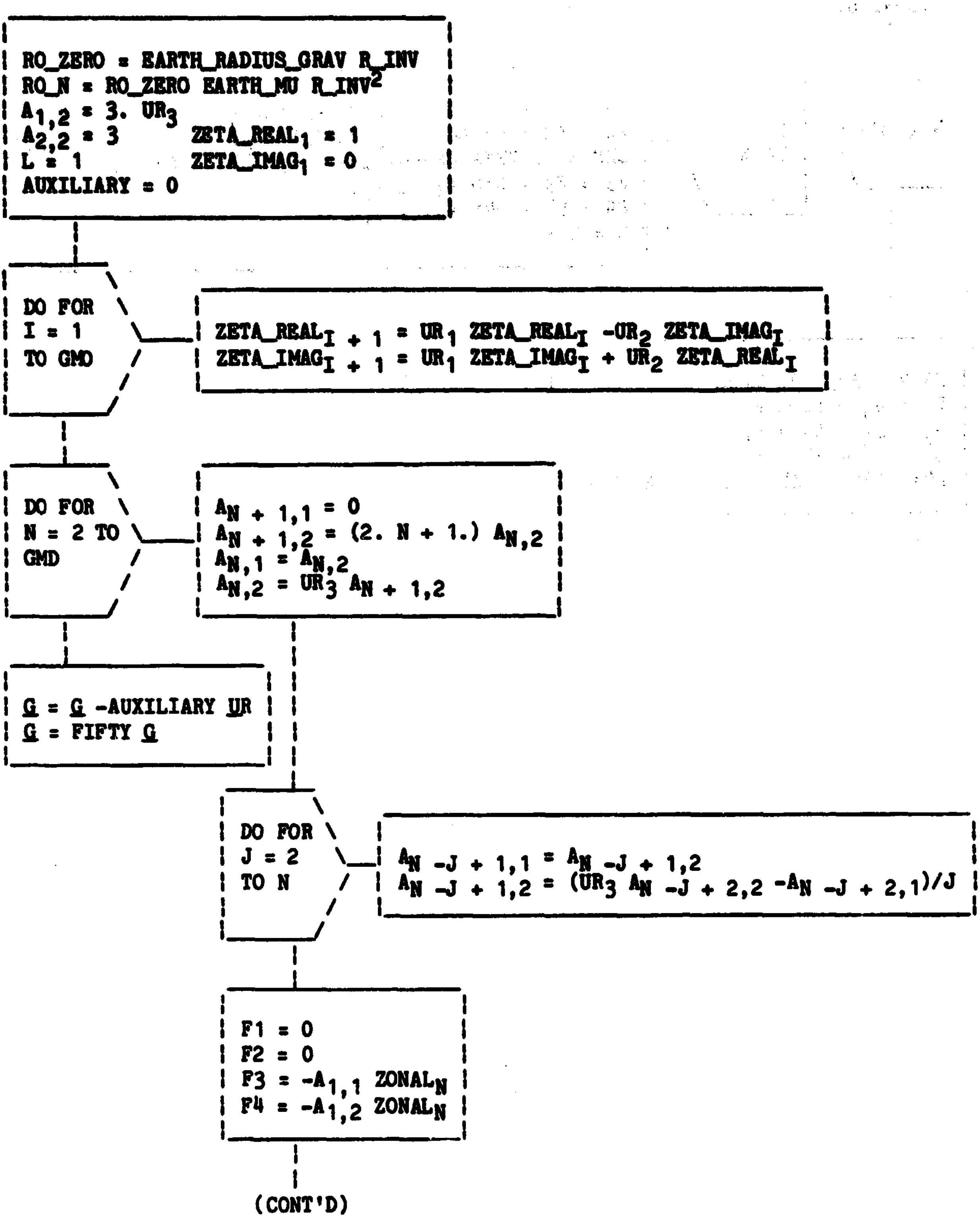
IN LIST: GMD, GMO, DM, VM, ATM, R, V, T



```
G = 0. D = 0.
VENT = 0.
FIFTY = EARTH_FIXED_TO_M50_COORD(T)
R _EF = FIFTY^T R
R_INV = 1./|R|
UR = R_INV R _EF
G _CENTRAL = - EARTH_MU R_INV^3 R

GMD ≥ 2
    EXECUTE: ACCEL_EARTH_GRAV CODE

VM = 1
    EXECUTE: ACCEL_ONORBIT_VENT_AND_THRUST CODE

DM = 1
    CALL:  SOLAR_EPHEM
    IN LIST:  T
    OUT LIST:  SDEC,CDEC1,COS_SOL_RA,SIN_SOL_RA
    EXECUTE: ONORBIT_DENSITY CODE
    EXECUTE: ACCEL_ONORBIT_DRAG CODE

ACCEL_ONORBIT = G _CENTRAL + G + D + VENT
```

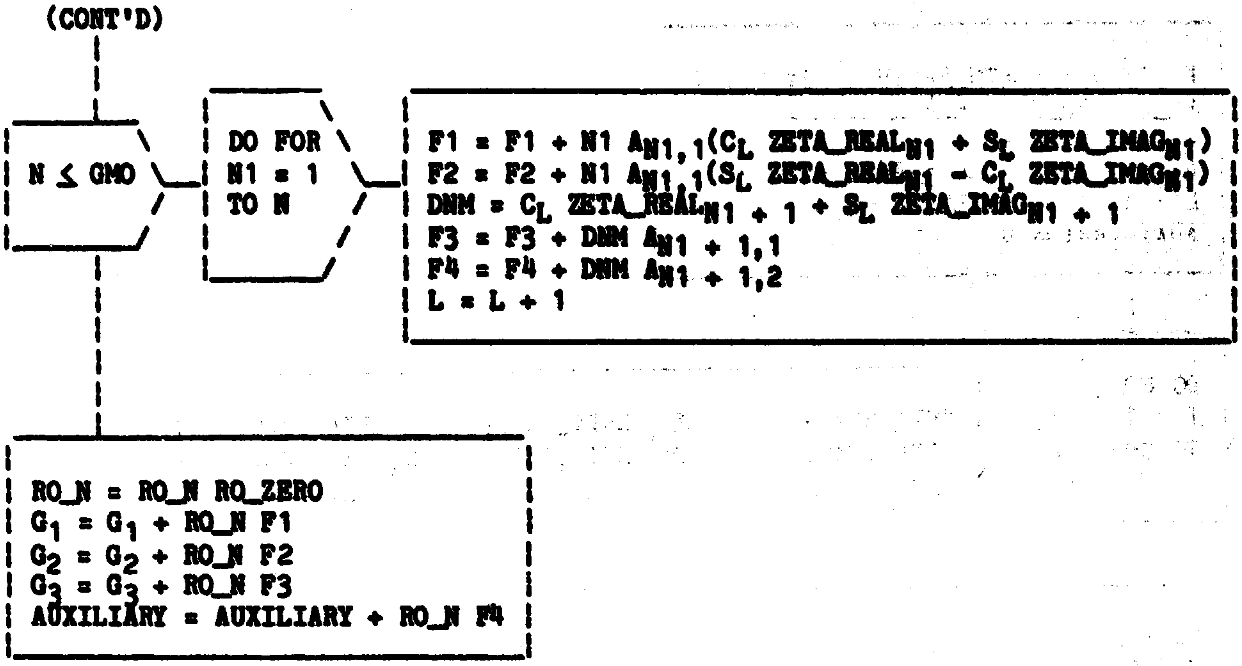
GMD CONTROLS THE DEGREE OF THE GRAVITY MODEL.
GMO CONTROLS THE ORDER OF THE GRAVITY MODEL.
DM CONTROLS THE USE OR NONUSE OF DRAG ACCELERATION MODEL.
VM CONTROLS THE USE OR NONUSE OF VENTING AND UNCOUPLED THRUSTING MODEL.
ATM CONTROLS THE INCORPORATION OF UNMODELED ACCELERATIONS (IN CONJUNCTION WITH THE SHUTTLE_FILTER_FLAG) AND WHEN COMPUTING DRAG, THE VARIOUS DRAG MODEL OPTIONS USED (i.e., CURRENT ATTITUDE, AREA, DRAG COEFFICIENT, ETC.).
R , V ARE THE POSITION AND VELOCITY VECTORS OF THE VEHICLE IN M50 COORDINATES.
T IS THE TIME IN GMT SECONDS.

Figure B-21.- ACCEL_ONORBIT FUNCTION.



```
RO_ZERO = EARTH_RADIUS_GRAV R_INV
RO_N = RO_ZERO EARTH_MU R_INV^2
$A_{1,2} = 3.\ UR_3$
$A_{2,2} = 3$          $ZETA\_REAL_1 = 1$
L = 1                  $ZETA\_IMAG_1 = 0$
AUXILIARY = 0
DO FOR I = 1 TO GMO
    $ZETA\_REAL_{I+1} = UR_1\ ZETA\_REAL_I - UR_2\ ZETA\_IMAG_I$
    $ZETA\_IMAG_{I+1} = UR_1\ ZETA\_IMAG_I + UR_2\ ZETA\_REAL_I$
DO FOR N = 2 TO GMD
    $A_{N+1,1} = 0$
    $A_{N+1,2} = (2.\ N + 1.)\ A_{N,2}$
    $A_{N,1} = A_{N,2}$
    $A_{N,2} = UR_3\ A_{N+1,2}$
    DO FOR J = 2 TO N
        $A_{N-J+1,1} = A_{N-J+1,2}$
        $A_{N-J+1,2} = (UR_3\ A_{N-J+2,2} - A_{N-J+2,1})/J$
    F1 = 0
    F2 = 0
    $F3 = -A_{1,1}\ ZONAL_N$
    $F4 = -A_{1,2}\ ZONAL_N$
    (CONT'D)
$\underline{G} = \underline{G} - AUXILIARY\ \underline{UR}$
$\underline{G} = FIFTY\ \underline{G}$
```

Figure B-22.- ACCEL_EARTH_GRAV CODE (Sheet 1 of 2).



```
(CONT'D)
$N \leq GMO$
    DO FOR $N1 = 1$ TO $N$
        $F1 = F1 + N1\ A_{N1,1}(C_L\ ZETA\_REAL_{N1} + S_L\ ZETA\_IMAG_{N1})$
        $F2 = F2 + N1\ A_{N1,1}(S_L\ ZETA\_REAL_{N1} - C_L\ ZETA\_IMAG_{N1})$
        $DNM = C_L\ ZETA\_REAL_{N1+1} + S_L\ ZETA\_IMAG_{N1+1}$
        $F3 = F3 + DNM\ A_{N1+1,1}$
        $F4 = F4 + DNM\ A_{N1+1,2}$
        $L = L + 1$
    $RO\_N = RO\_N\ RO\_ZERO$
    $G_1 = G_1 + RO\_N\ F1$
    $G_2 = G_2 + RO\_N\ F2$
    $G_3 = G_3 + RO\_N\ F3$
    $AUXILIARY = AUXILIARY + RO\_N\ F4$
```

Figure B-22.- ACCEL_EARTH_GRAV CODE (Sheet 2 of 2).

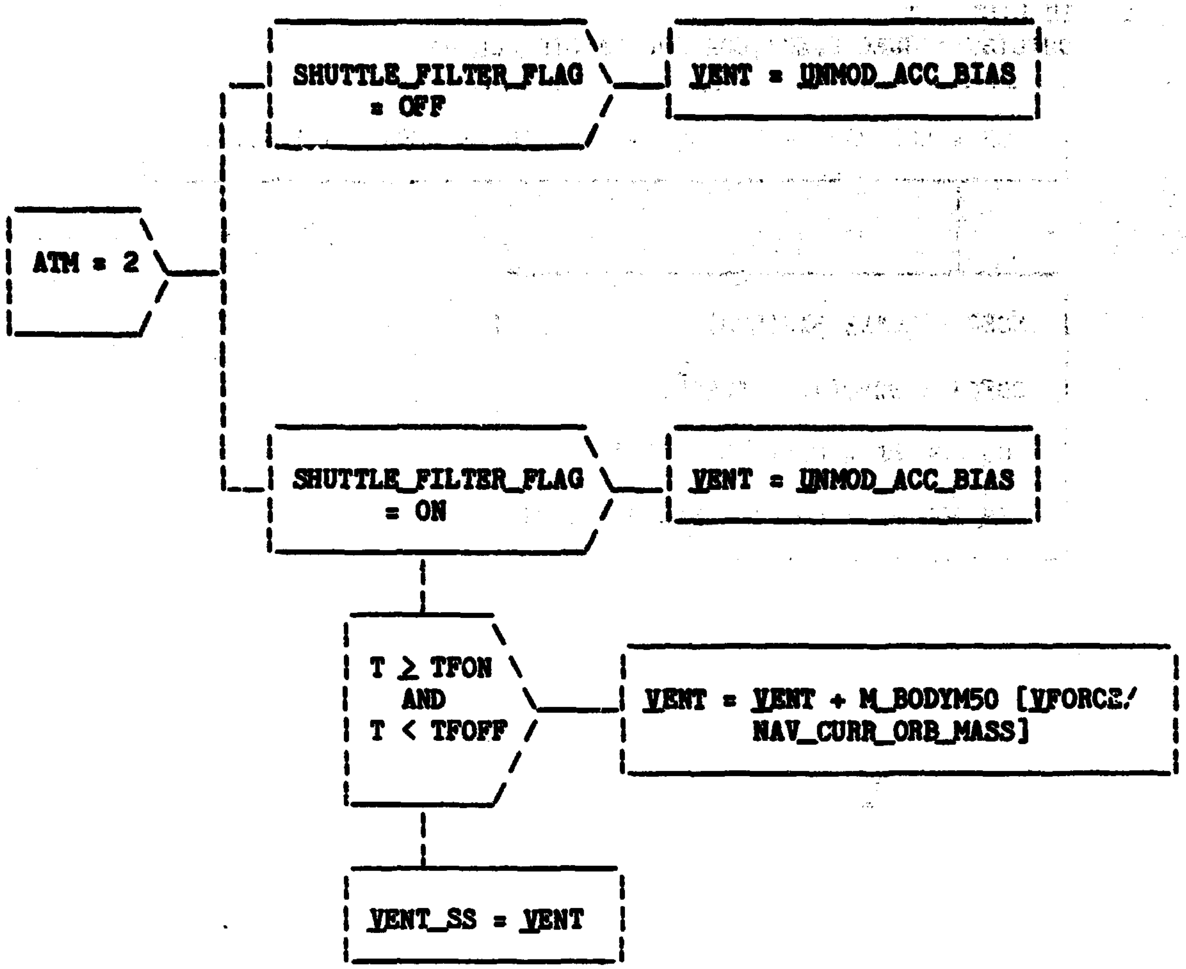


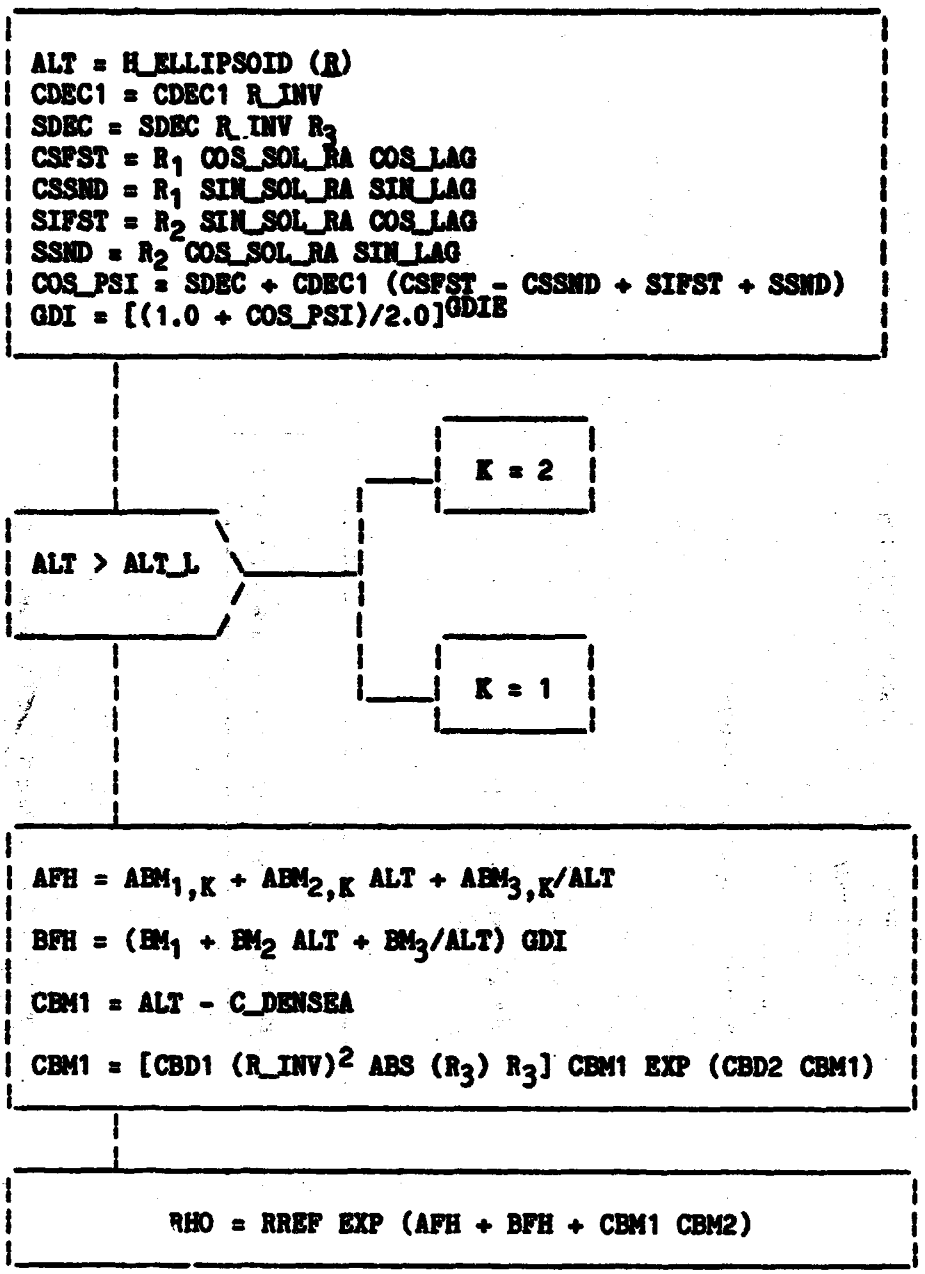
Figure B-23.- ACCEL_ONORBIT_VENT_AND_THRUST CODE.

IN LIST: T

OUTLIST: SDEC,CDEC1,COS_SOL_RA,SIN_SOL_RA

```
LOS = LOS_ZERO + T LOS_R - LOC SIN (T OMEG_C + PHASE_C)

SDEC = LOSK3 SIN(LOS)
CDEC1 = SQRT(1. - SDEC^2)
COS_SOL_RA = COS(LOS)/CDEC1
SIN_SOL_RA = LOSK1 SIN(LOS)/CDEC1
```

Figure B-24.- SOLAR_EPHEM.



ALT = H_ELLIPSOID ($\underline{R}$)
CDEC1 = CDEC1 R_INV
SDEC = SDEC R_INV R_3
CSFST = R_1 COS_SOL_RA COS_LAG
CSSND = R_1 SIN_SOL_RA SIN_LAG
SIFST = R_2 SIN_SOL_RA COS_LAG
SSND = R_2 COS_SOL_RA SIN_LAG
COS_PSI = SDEC + CDEC1 (CSFST - CSSND + SIFST + SSND)
GDI = $[(1.0 + \text{COS\_PSI})/2.0]^{\text{GDIE}}$

ALT > ALT_L

K = 2

K = 1

AFH = $\text{ABM}_{1,K}$ + $\text{ABM}_{2,K}$ ALT + $\text{ABM}_{3,K}$/ALT

BFH = (BM_1 + BM_2 ALT + BM_3/ALT) GDI

CBM1 = ALT - C_DENSEA

CBM1 = [CBD1 $(\text{R\_INV})^2$ ABS (R_3) R_3] CBM1 EXP (CBD2 CBM1)

RHO = RREF EXP (AFH + BFH + CBM1 CBM2)

Figure B-25.- ONORBIT_DENSITY CODE.

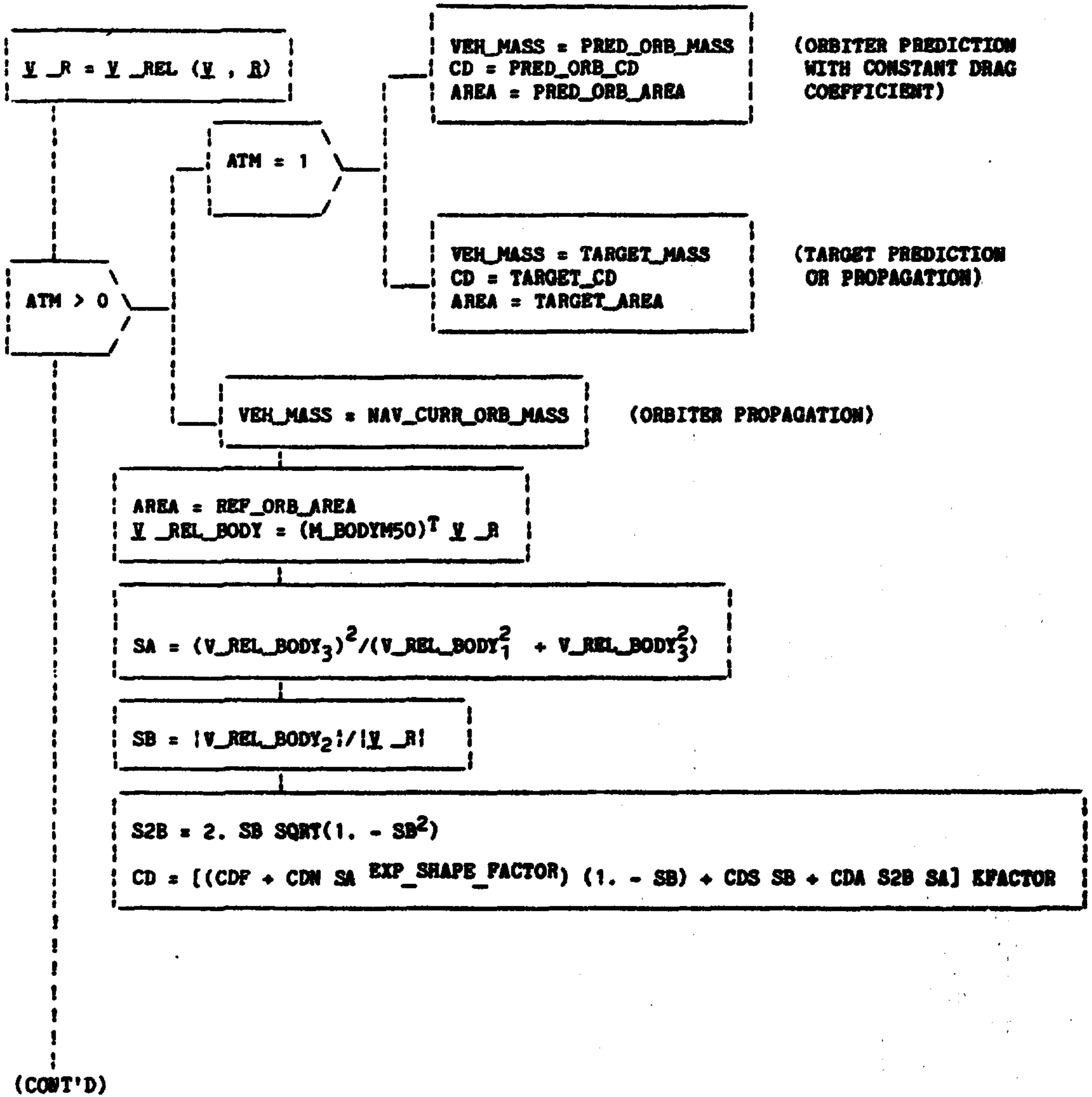


Figure B-26.- ACCEL_ONORBIT_DRAG CODE (Sheet 1 of 2).

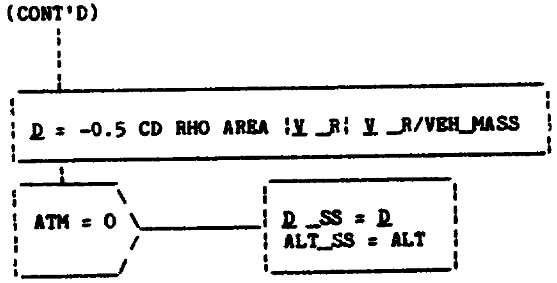


Figure B-26.- ACCEL_ONORBIT_DRAG CODE (Sheet 2 of 2).

$$\mathrm{H\_ELLIPSOID}(\underline{R}) = |\underline{R}| - \frac{(1 - \mathrm{ELLIPT})\ \mathrm{EARTH\_RADIUS\_EQUATOR}}{\sqrt{1 + ((1 - \mathrm{ELLIPT})^2 - 1)\ (1 - (\mathrm{UNIT}(\underline{R}) \cdot \underline{\mathrm{E}}\mathrm{ARTH\_POLE})^2)}}$$

Figure B-27.- H_ELLIPSOID FUNCTION.

$\underline{V}$ _REL($\underline{V}$, $\underline{R}$) = $\underline{V}$ - EARTH_RATE ($\underline{E}$ARTH_POLE X $\underline{R}$)

Figure B-28.- $\underline{V}$ _REL FUNCTION.

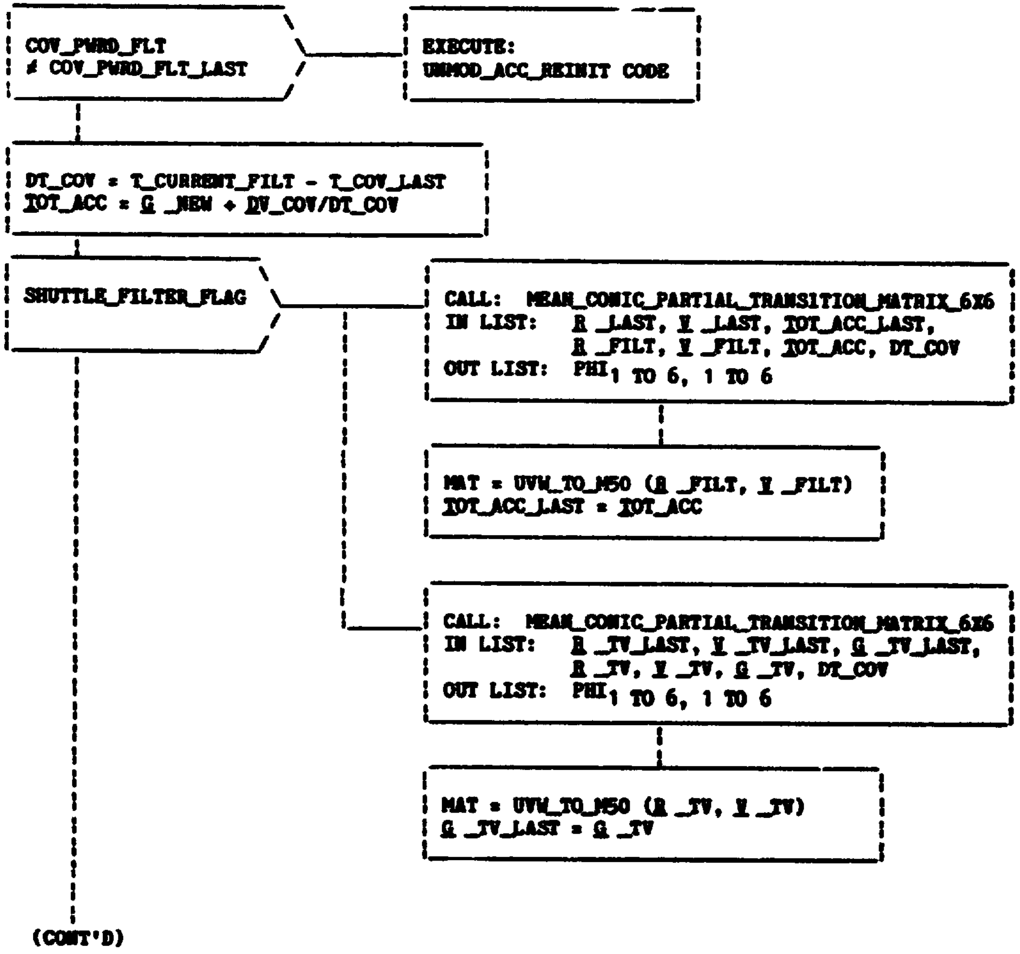
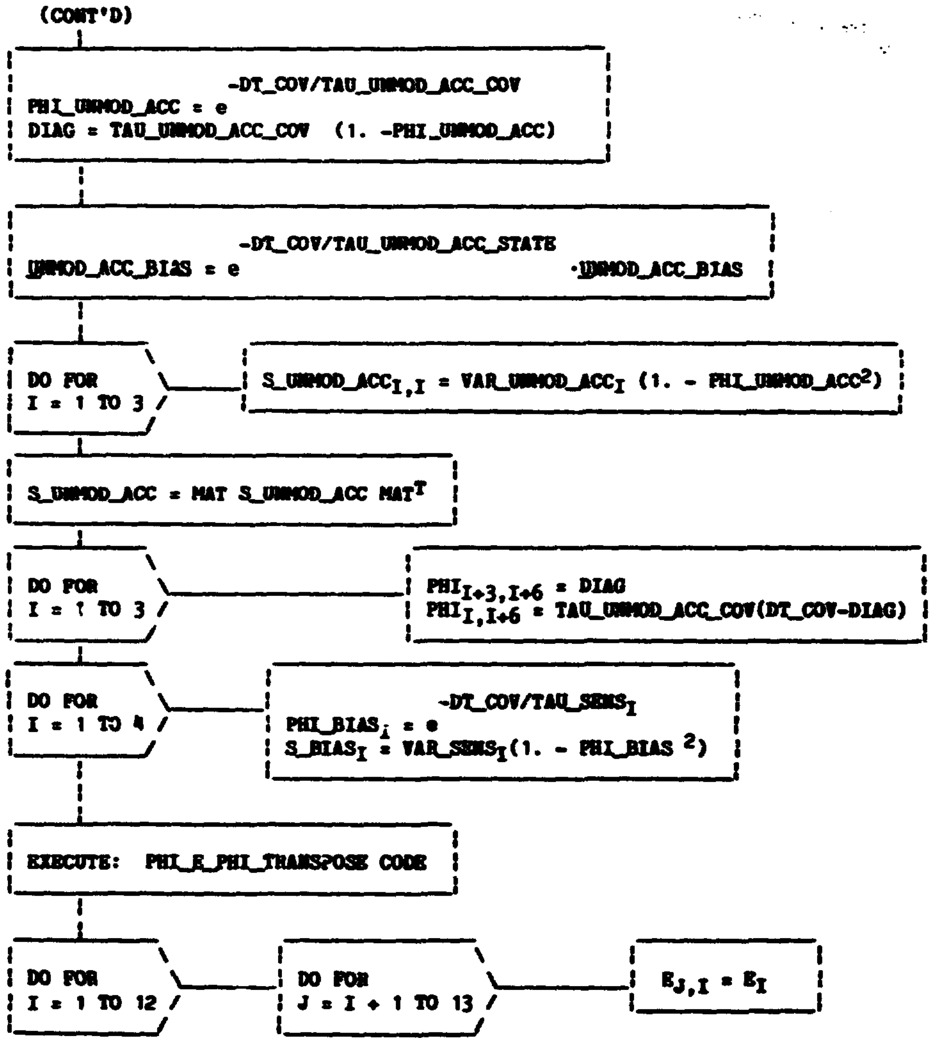


Figure B-29.- REND_BIAS_AND_COV_PROP (Sheet 1 of 2).



```
(CONT'D)
$PHI\_UNMOD\_ACC = e^{-DT\_COV/TAU\_UNMOD\_ACC\_COV}$
$DIAG = TAU\_UNMOD\_ACC\_COV \ (1. - PHI\_UNMOD\_ACC)$
$UNMOD\_ACC\_BIAS = e^{-DT\_COV/TAU\_UNMOD\_ACC\_STATE} \cdot UNMOD\_ACC\_BIAS$
DO FOR I = 1 TO 3
    $S\_UNMOD\_ACC_{I,I} = VAR\_UNMOD\_ACC_I \ (1. - PHI\_UNMOD\_ACC^2)$
$S\_UNMOD\_ACC = MAT \ S\_UNMOD\_ACC \ MAT^T$
DO FOR I = 1 TO 3
    $PHI_{I+3,I+6} = DIAG$
    $PHI_{I,I+6} = TAU\_UNMOD\_ACC\_COV(DT\_COV-DIAG)$
DO FOR I = 1 TO 4
    $PHI\_BIAS_i = e^{-DT\_COV/TAU\_SENS_I}$
    $S\_BIAS_I = VAR\_SENS_I(1. - PHI\_BIAS^2)$
EXECUTE:  PHI_E_PHI_TRANSPOSE CODE
DO FOR I = 1 TO 12
    DO FOR J = I + 1 TO 13
        $E_{J,I} = E_I$
```

Figure B-29.- REND_BIAS_AND_COV_PROP (Sheet 2 of 2).

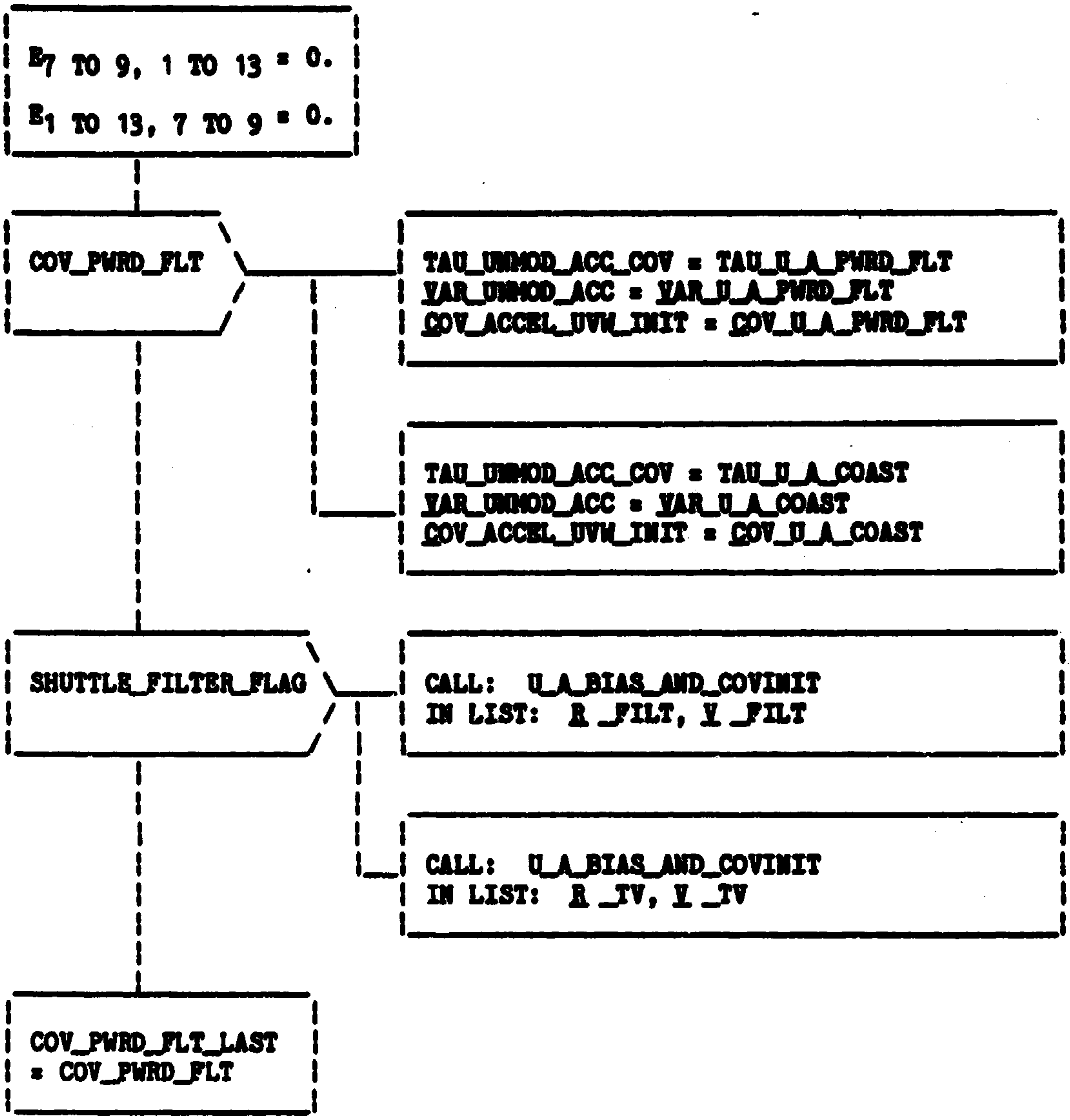
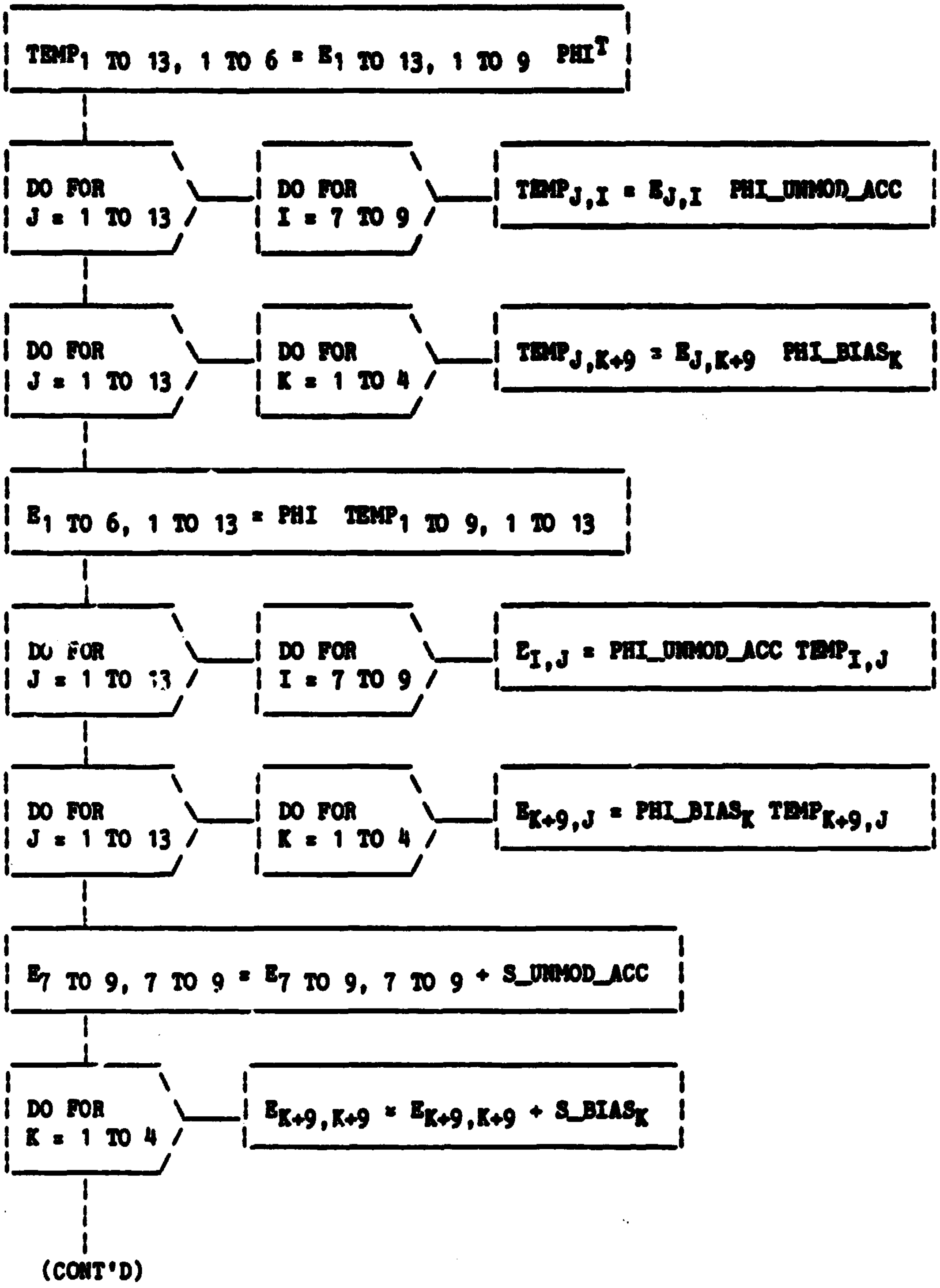


Figure B-30.- UNMOD_ACC_REINIT CODE.



```
$TEMP_{1\ TO\ 13,\ 1\ TO\ 6} = E_{1\ TO\ 13,\ 1\ TO\ 9}\ PHI^T$
DO FOR J = 1 TO 13
    DO FOR I = 7 TO 9
        $TEMP_{J,I} = E_{J,I}$  PHI_UNMOD_ACC
DO FOR J = 1 TO 13
    DO FOR K = 1 TO 4
        $TEMP_{J,K+9} = E_{J,K+9}$  $PHI\_BIAS_K$
$E_{1\ TO\ 6,\ 1\ TO\ 13} = PHI\ \ TEMP_{1\ TO\ 9,\ 1\ TO\ 13}$
DO FOR J = 1 TO 13
    DO FOR I = 7 TO 9
        $E_{I,J}$ = PHI_UNMOD_ACC $TEMP_{I,J}$
DO FOR J = 1 TO 13
    DO FOR K = 1 TO 4
        $E_{K+9,J} = PHI\_BIAS_K\ TEMP_{K+9,J}$
$E_{7\ TO\ 9,\ 7\ TO\ 9} = E_{7\ TO\ 9,\ 7\ TO\ 9}$ + S_UNMOD_ACC
DO FOR K = 1 TO 4
    $E_{K+9,K+9} = E_{K+9,K+9} + S\_BIAS_K$
(CONT'D)
```

Figure B-31.- PHI_E_PHI_TRANSPOSE CODE (Sheet 1 of 2).

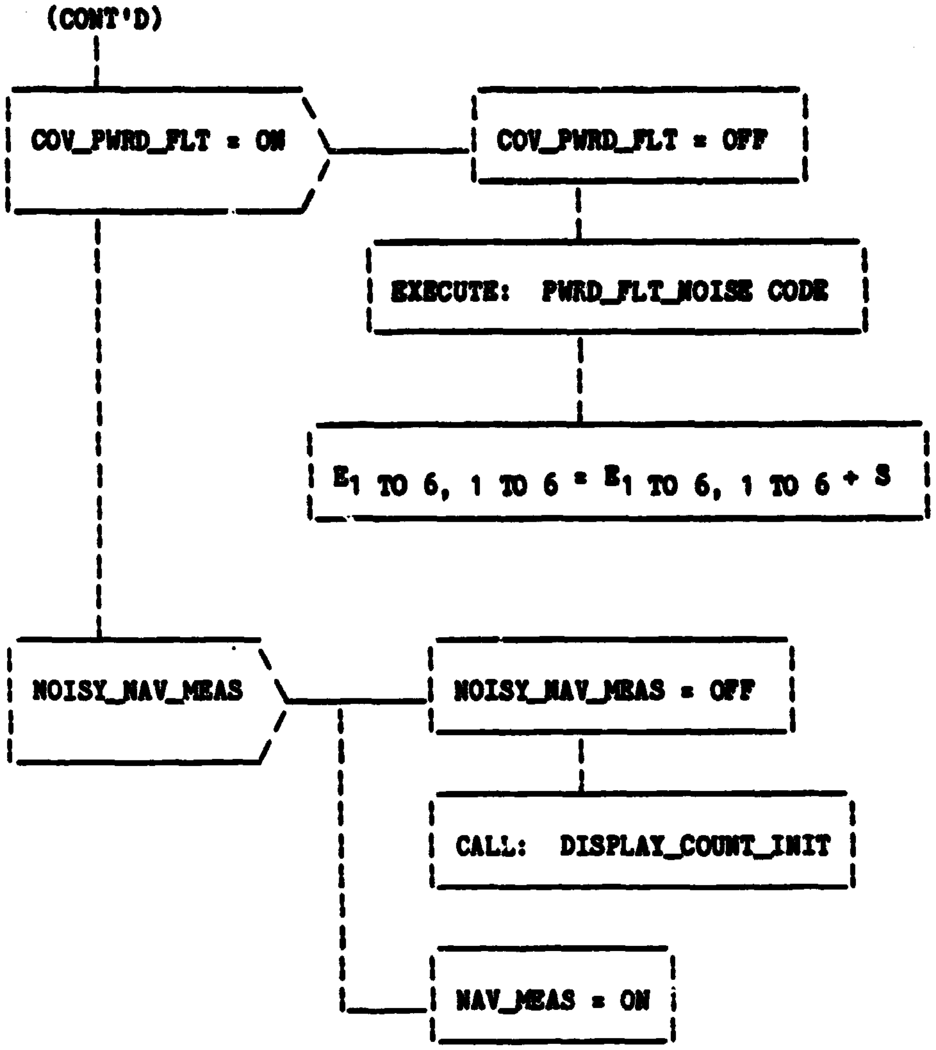


Figure B-31.- PHI_E_PHI_TRANSPOSE CODE (Sheet 2 of 2).

$S_{4,4} = DV\_COV_3{}^2 + DV\_COV_2{}^2$

$S_{5,5} = DV\_COV_1{}^2 + DV\_COV_3{}^2$

$S_{6,6} = DV\_COV_1{}^2 + DV\_COV_2{}^2$

$S_{4,5} = -DV\_COV_1 \; DV\_COV_2$

$S_{4,6} = -DV\_COV_1 \; DV\_COV_3$

$S_{5,6} = -DV\_COV_2 \; DV\_COV_3$

$S_{6,5} = S_{5,6}$

$S_{6,4} = S_{4,6}$

$S_{5,4} = S_{4,5}$

$S_{4 \text{ TO } 6,4 \text{ TO } 6} = (VAR\_IMU\_ALIGN)S_{4 \text{ TO } 6,4 \text{ TO } 6}$

$S_{1 \text{ TO } 3,4 \text{ TO } 6} = 0.5(DT\_COV)S_{4 \text{ TO } 6,4 \text{ TO } 6}$

$S_{1 \text{ TO } 3,1 \text{ TO } 3} = 0.5(DT\_COV)S_{1 \text{ TO } 3,4 \text{ TO } 6}$

Figure B-32.- PWRD_FLT_NOISE CODE.

IN LIST: R , V

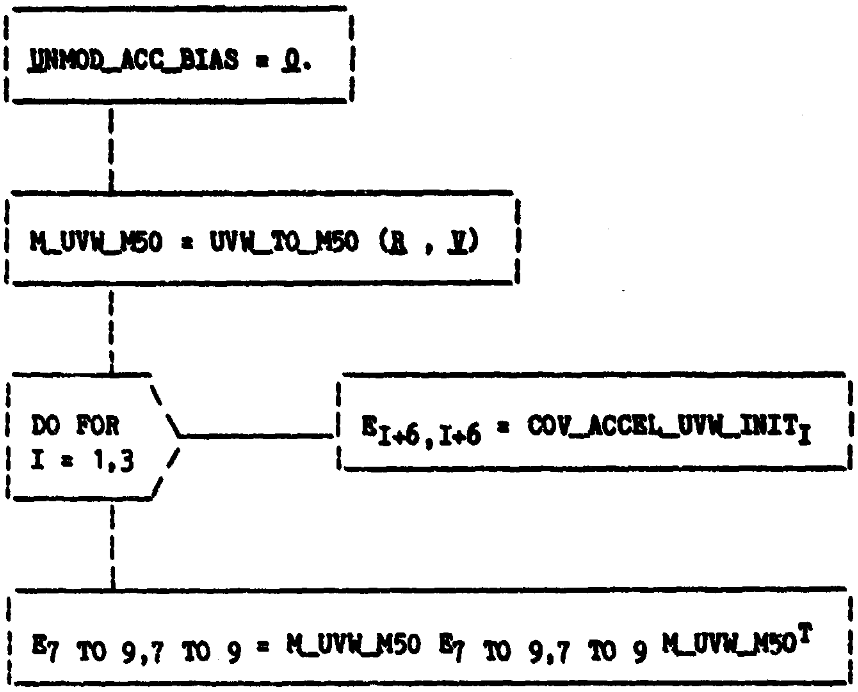
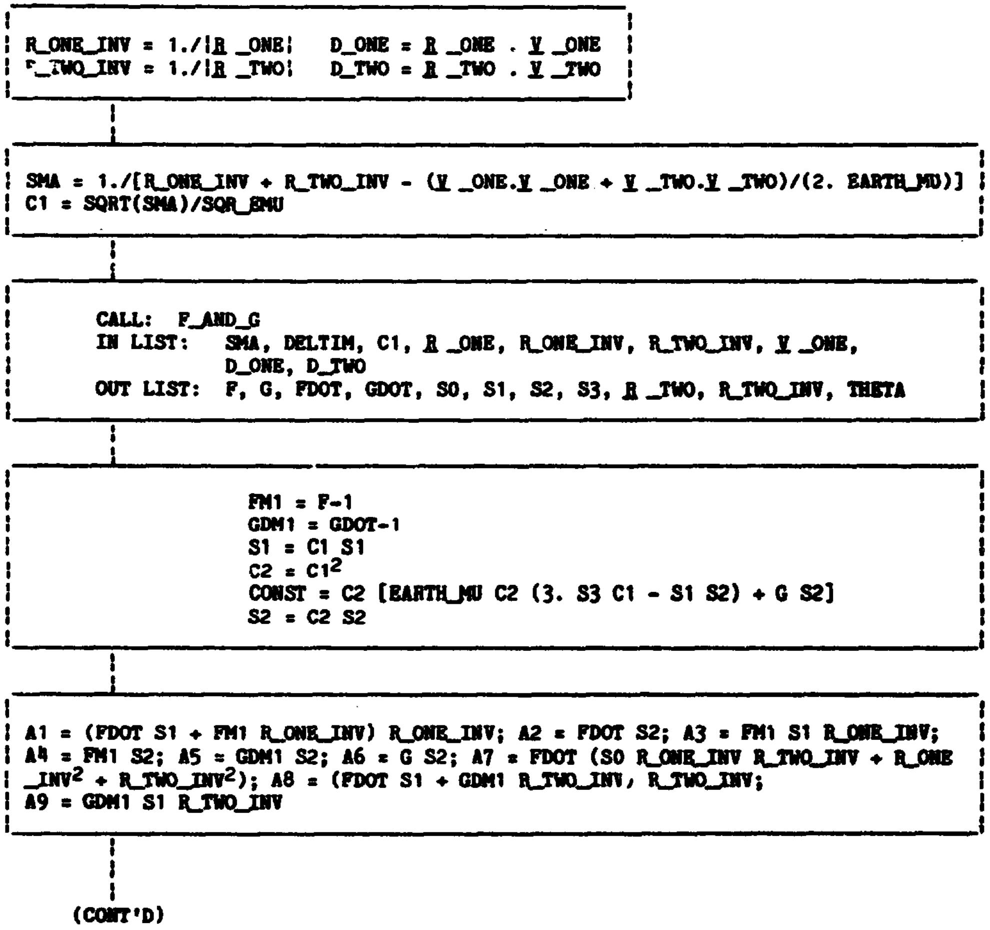


Figure B-33.- U_A_BIAS_AND_COVINIT.

79FM10

IN LIST: R _ONE, V _ONE, G _ONE, R _TWO, V _TWO, G _TWO, DELTIM
OUT LIST: PHI_MC



```
R_ONE_INV = 1./|R _ONE|    D_ONE = R _ONE . V _ONE
R_TWO_INV = 1./|R _TWO|    D_TWO = R _TWO . V _TWO

SMA = 1./[R_ONE_INV + R_TWO_INV - (V _ONE.V _ONE + V _TWO.V _TWO)/(2. EARTH_MU)]
C1 = SQRT(SMA)/SQR_EMU

      CALL:  F_AND_G
      IN LIST:   SMA, DELTIM, C1, R _ONE, R_ONE_INV, R_TWO_INV, V _ONE,
                 D_ONE, D_TWO
      OUT LIST:  F, G, FDOT, GDOT, S0, S1, S2, S3, R _TWO, R_TWO_INV, THETA

                 FM1 = F-1
                 GDM1 = GDOT-1
                 S1 = C1 S1
                 C2 = $C1^2$
                 CONST = C2 [EARTH_MU C2 (3. S3 C1 - S1 S2) + G S2]
                 S2 = C2 S2

A1 = (FDOT S1 + FM1 R_ONE_INV) R_ONE_INV; A2 = FDOT S2; A3 = FM1 S1 R_ONE_INV;
A4 = FM1 S2; A5 = GDM1 S2; A6 = G S2; A7 = FDOT (S0 R_ONE_INV R_TWO_INV + R_ONE
_$INV^2$ + R_TWO_$INV^2$); A8 = (FDOT S1 + GDM1 R_TWO_INV) R_TWO_INV;
A9 = GDM1 S1 R_TWO_INV
```

(CONT'D)

Figure B-34.- MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6 X 6 (Sheet 1 of 2).

(CONT'D)

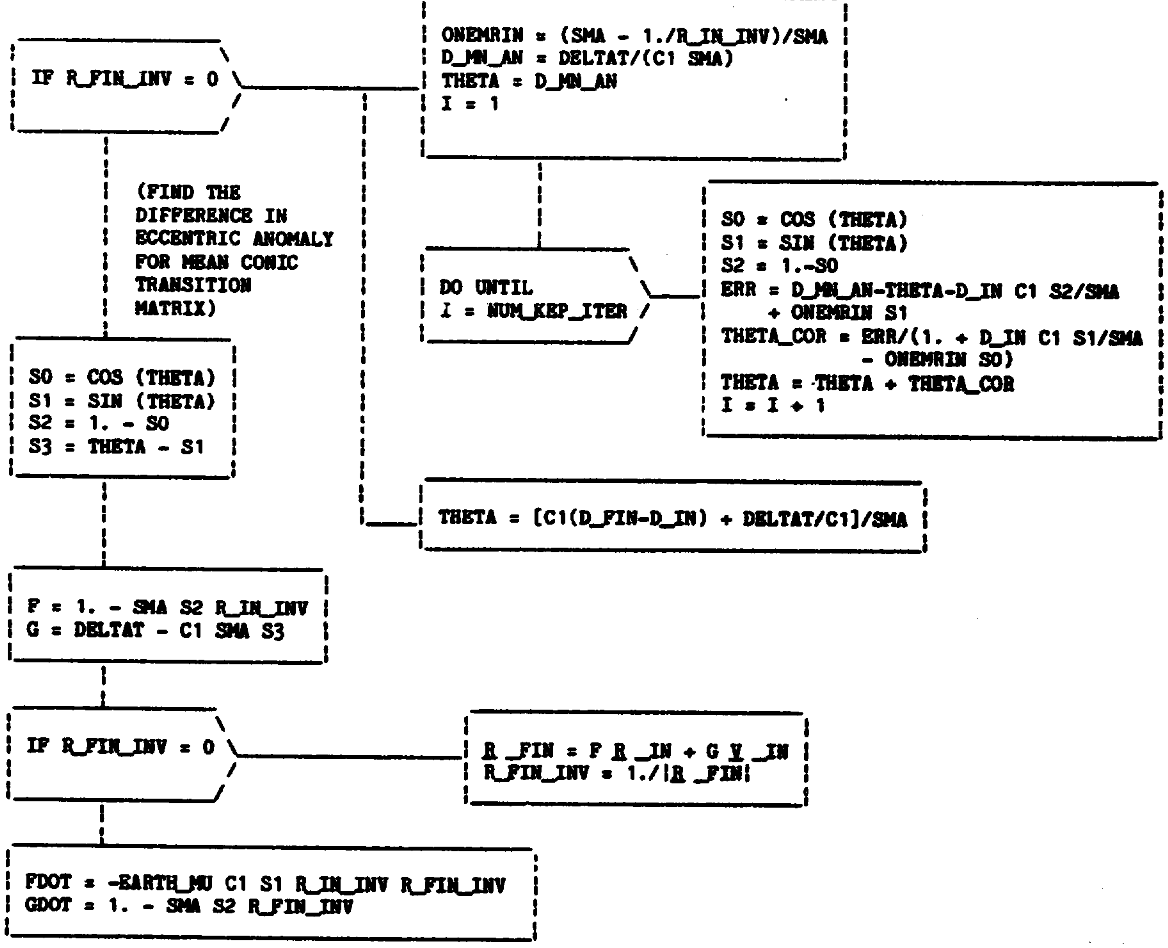
```
$\text{TEMP} = \text{A4}\ \underline{V}\text{\_TWO} - \text{A2}\ \underline{R}\text{\_TWO}$
$\text{PHI\_MC}_{1\ \text{TO}\ 3,\ 1\ \text{TO}\ 3} = \text{F ID\_MATRIX\_3 X 3} + \text{CONST}\ \underline{V}\text{\_TWO}\ \underline{G}\text{\_ONE} +$
        $(\text{A3}\ \underline{V}\text{\_TWO} - \text{A1}\ \underline{R}\text{\_TWO})\ \underline{R}\text{\_ONE} + \text{TEMP}\ \underline{V}\text{\_ONE}$
$\text{PHI\_MC}_{1\ \text{TO}\ 3,\ 4\ \text{TO}\ 6} = \text{G ID\_MATRIX\_3 X 3} - \text{CONST}\ \underline{V}\text{\_TWO}\ \underline{V}\text{\_ONE} +$
        $\text{TEMP}\ \underline{R}\text{\_ONE} + (\text{A6}\ \underline{V}\text{\_TWO} - \text{A5}\ \underline{R}\text{\_TWO})\ \underline{V}\text{\_ONE}$
$\text{TEMP} = \text{A2}\ \underline{V}\text{\_TWO} - \text{A8}\ \underline{R}\text{\_TWO}$
$\text{PHI\_MC}_{4\ \text{TO}\ 6,\ 1\ \text{TO}\ 3} = \text{FDOT ID\_MATRIX\_3 X 3} + \text{CONST}\ \underline{G}\text{\_TWO}\ \underline{G}\text{\_ONE} +$
        $(\text{A1}\ \underline{V}\text{\_TWO} - \text{A7}\ \underline{R}\text{\_TWO})\ \underline{R}\text{\_ONE} + \text{TEMP}\ \underline{V}\text{\_ONE}$
$\text{PHI\_MC}_{4\ \text{TO}\ 6,\ 4\ \text{TO}\ 6} = \text{GDOT ID\_MATRIX\_3 X 3} - \text{CONST}\ \underline{G}\text{\_TWO}\ \underline{V}\text{\_ONE}$
        $+ \text{TEMP}\ \underline{R}\text{\_ONE} + (\text{A5}\ \underline{V}\text{\_TWO} - \text{A9}\ \underline{R}\text{\_TWO})\ \underline{V}\text{\_ONE}$
```

Figure B-34.- MEAN_CONIC_PARTIAL_TRANSITION_MATRIX_6 X 6 (Sheet 2 of 2).

IN LIST: SMA, DELTAT, C1, R _IN, R_IN_INV, R_FIN_INV, V _IN, D_IN, D_FIN

OUT LIST: F, G, FDOT, GDOT, S0, S1, S2, S3, R _FIN, R_FIN_INV, THETA

(SOLVE KEPLER'S EQUATION FOR PINES METHOD OR SV_INTERP)



```
IF R_FIN_INV = 0
    ONEMRIN = (SMA - 1./R_IN_INV)/SMA
    D_MN_AN = DELTAT/(C1 SMA)
    THETA = D_MN_AN
    I = 1
    DO UNTIL I = NUM_KEP_ITER
        S0 = COS (THETA)
        S1 = SIN (THETA)
        S2 = 1.-S0
        ERR = D_MN_AN-THETA-D_IN C1 S2/SMA
              + ONEMRIN S1
        THETA_COR = ERR/(1. + D_IN C1 S1/SMA
                         - ONEMRIN S0)
        THETA = THETA + THETA_COR
        I = I + 1
ELSE
    THETA = [C1(D_FIN-D_IN) + DELTAT/C1]/SMA
(FIND THE
DIFFERENCE IN
ECCENTRIC ANOMALY
FOR MEAN CONIC
TRANSITION
MATRIX)
S0 = COS (THETA)
S1 = SIN (THETA)
S2 = 1. - S0
S3 = THETA - S1
F = 1. - SMA S2 R_IN_INV
G = DELTAT - C1 SMA S3
IF R_FIN_INV = 0
    R _FIN = F R _IN + G V _IN
    R_FIN_INV = 1./|R _FIN|
FDOT = -EARTH_MU C1 S1 R_IN_INV R_FIN_INV
GDOT = 1. - SMA S2 R_FIN_INV
```

Figure B-35.- F_AND_G.

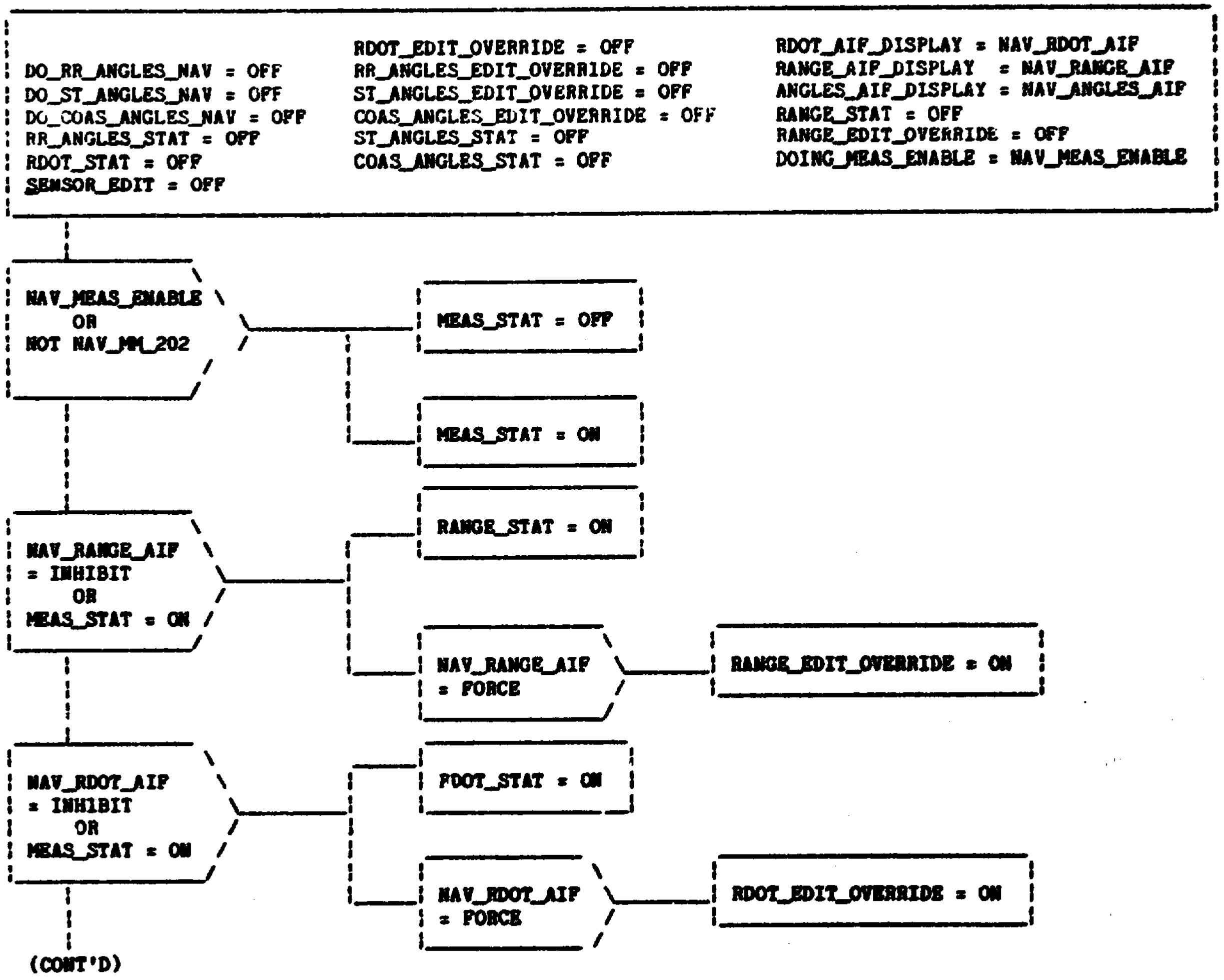


Figure B-36.- REND_SENSOR_SELECT (Sheet 1 of 3).

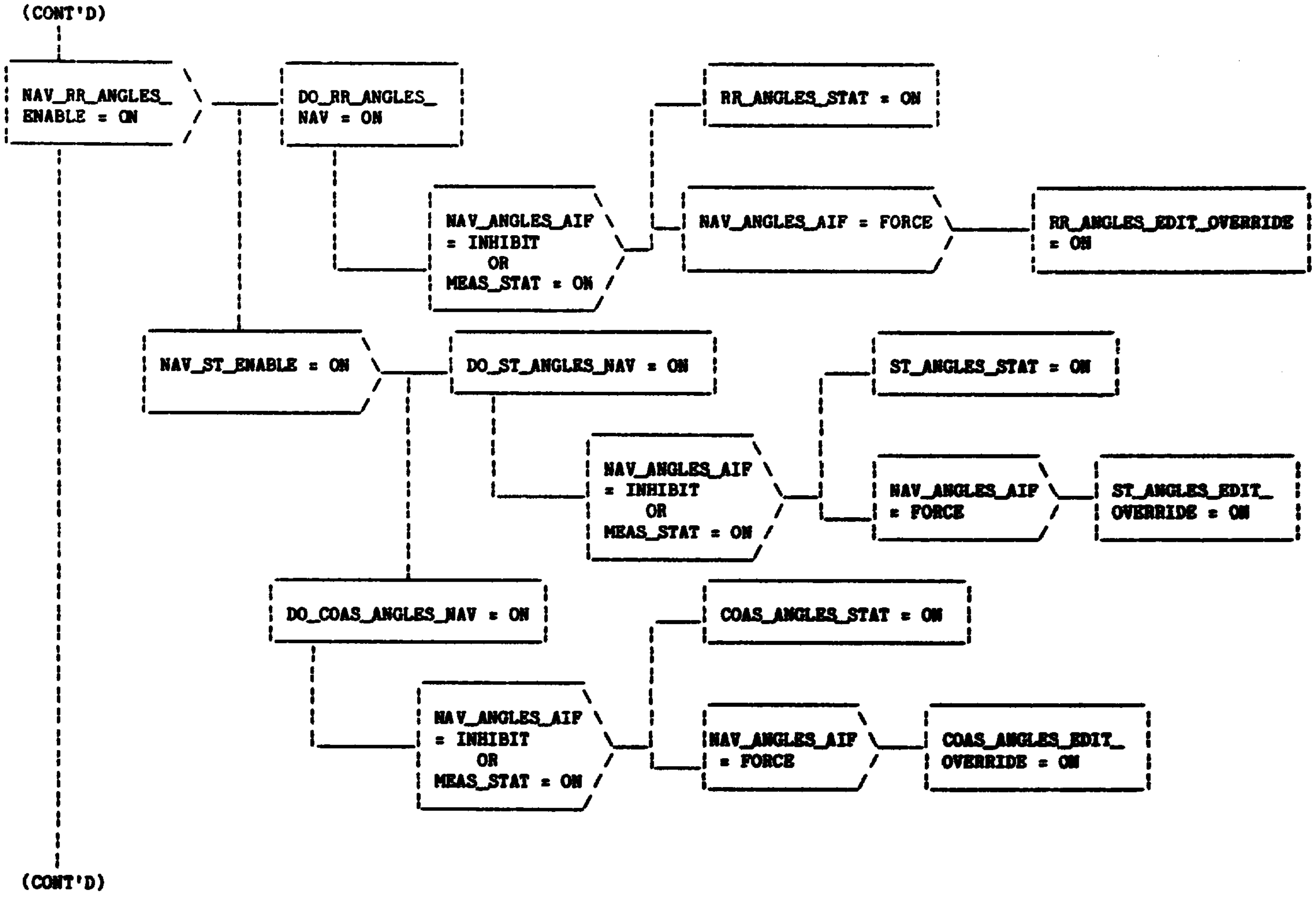


Figure B-36.- REND_SENSOR_SELECT (Sheet 2 of 3).

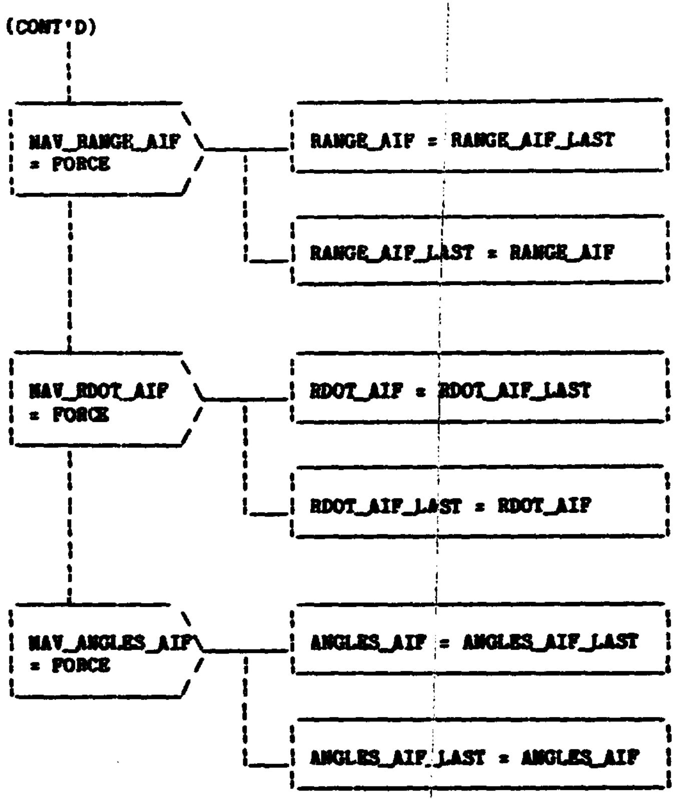


Figure B-36.- REND_SENSOR_SELECT (Sheet 3 of 3).

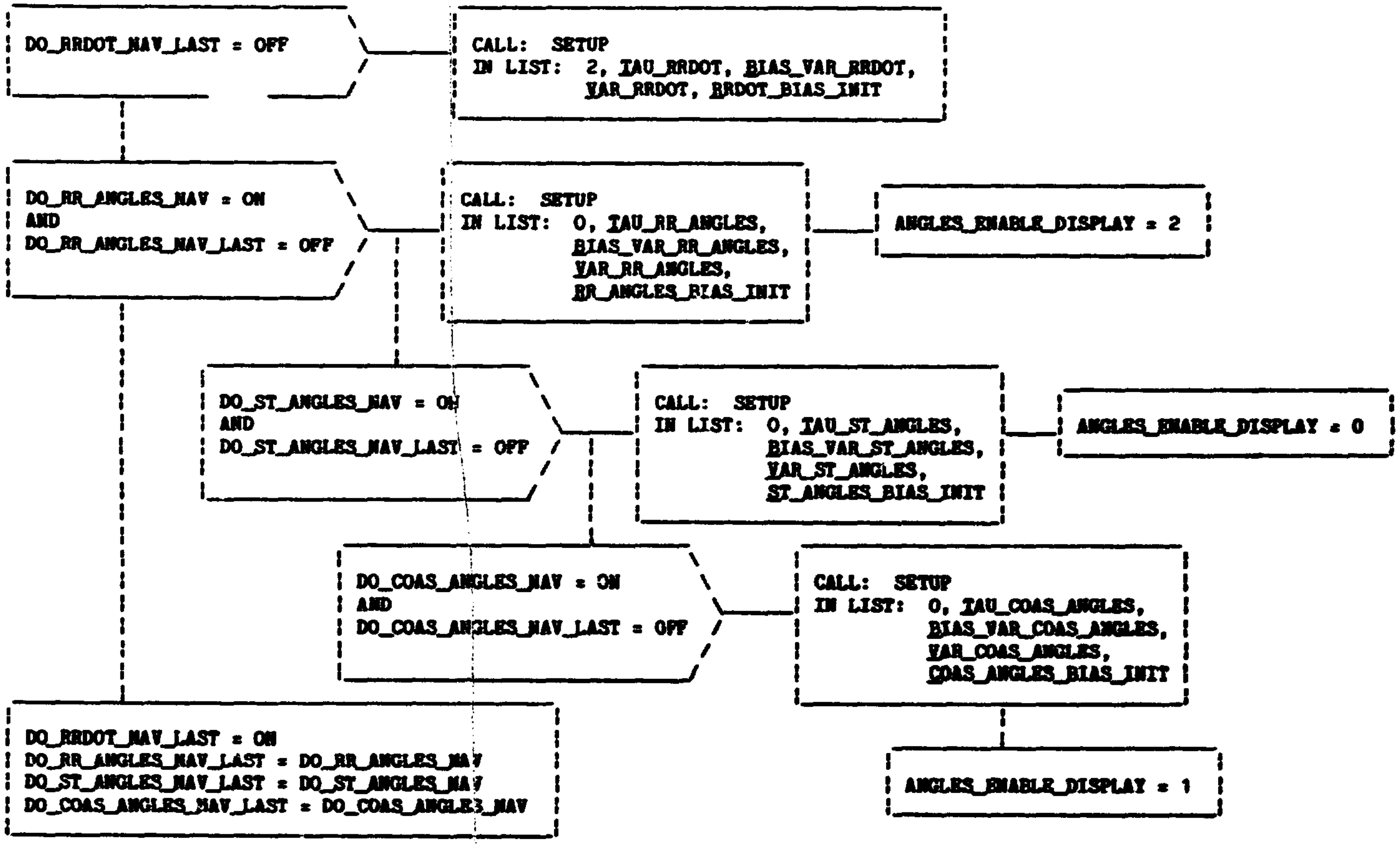


Figure B-37.- REND_NAV_SENSOR_INIT.

IN LIST: I, TAU, BIAS_VAR, BIAS_COV_VAR, BIAS_INIT

```
DO FOR J = 1 TO 2
    $K = I + J$
    $TAU\_SENS_K = TAU_J$
    $SENSOR\_BIAS_K = BIAS\_INIT_J$
    $VAR\_SENS_K = BIAS\_VAR_J$
    $E_{9 + K,\ 1 \text{ TO } 13} = 0.0$
    $E_{1 \text{ TO } 13,\ 9 + K} = 0.0$
    $E_{9 + K,\ 9 + K} = BIAS\_COV\_VAR_J$
    $N\_ACCEPT_K = 0$
    $N\_REJECT_K = 0$
    $SEQ\_ACCEPT_K = 0$
    $SEQ\_REJECT_K = 0$
    $NAV\_SIG_K = 0.0$
```

Figure B-38.- SETUP.

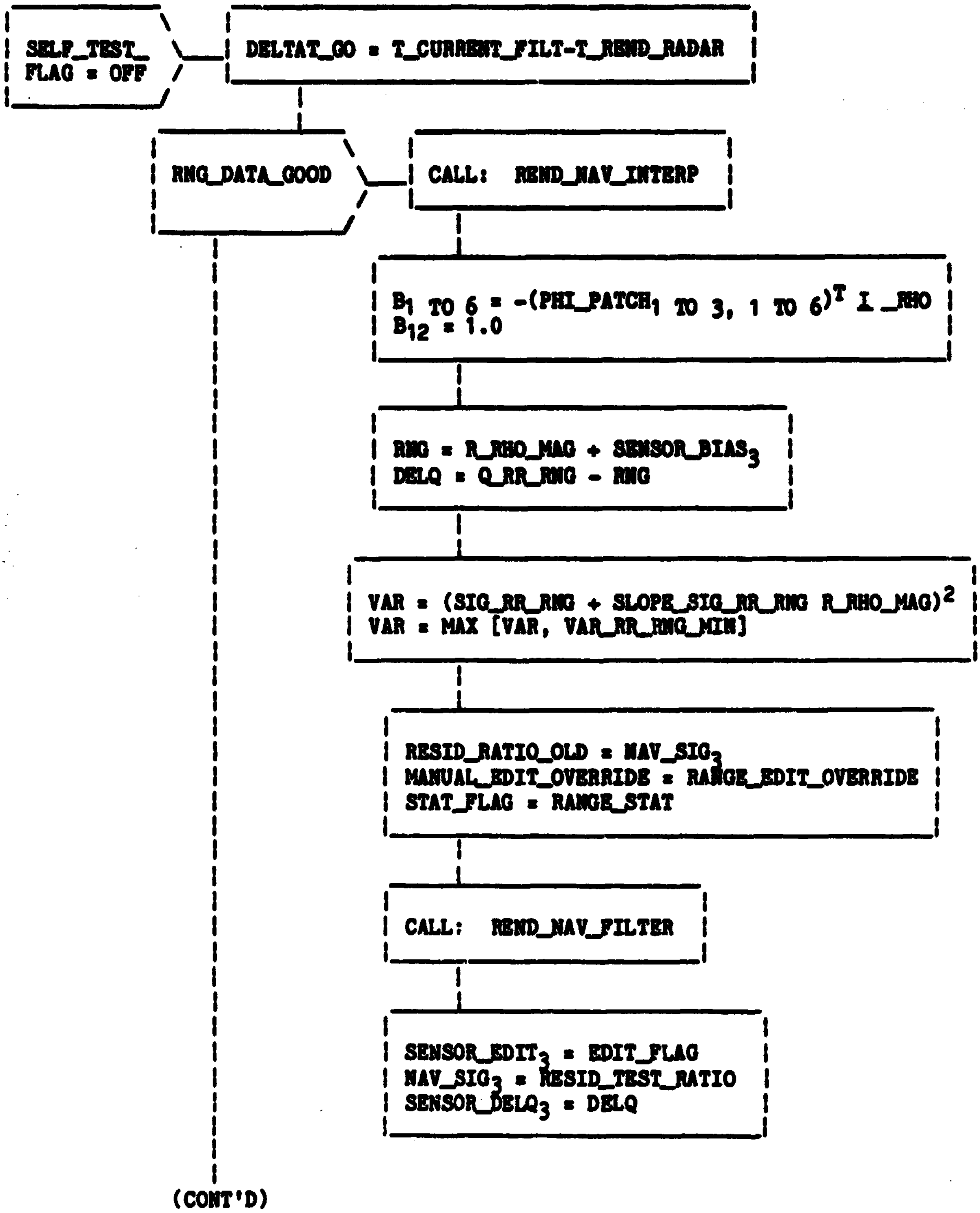


Figure B-39.- RRDOT_NAV (Sheet 1 of 2).

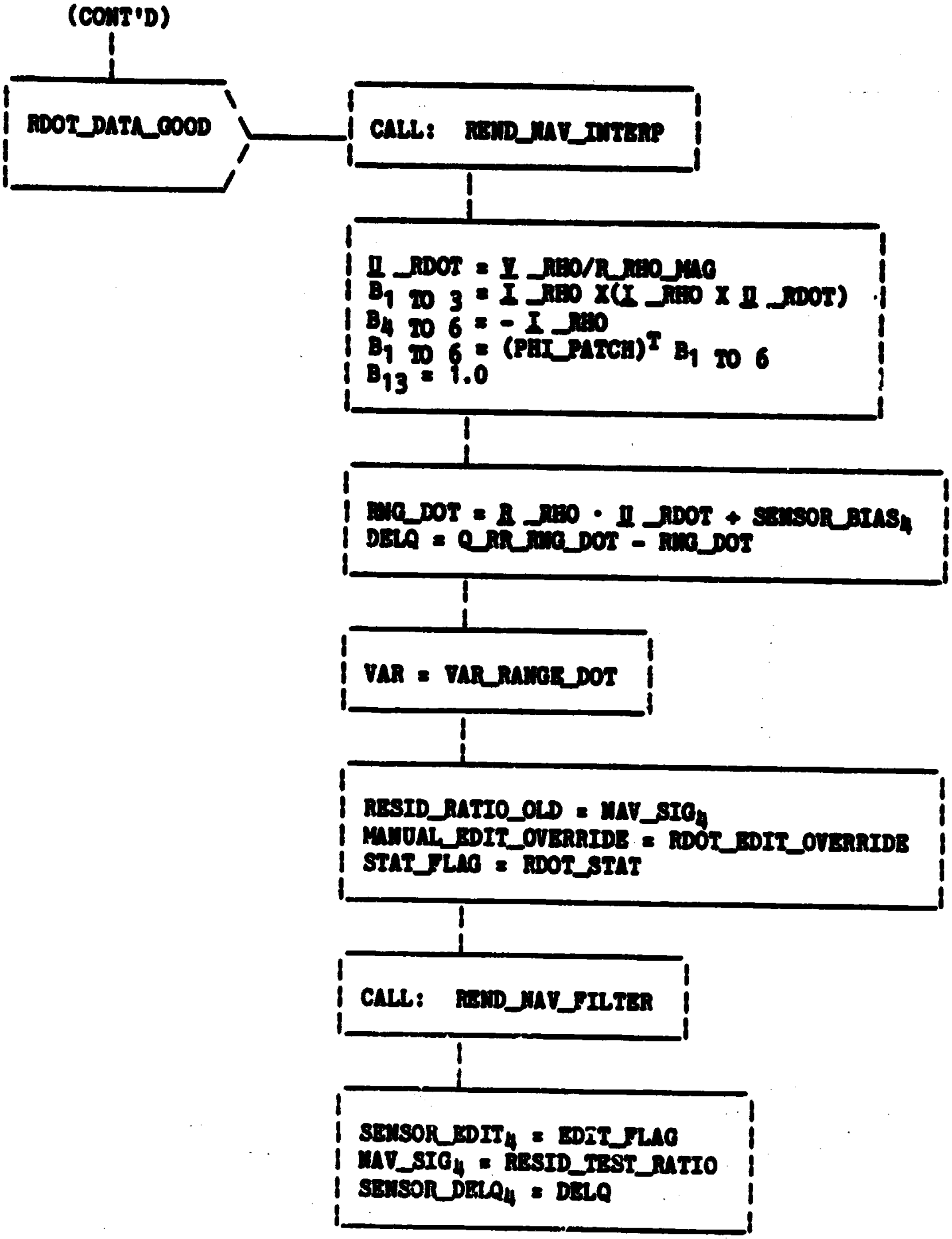


Figure B-39.- RRDOT_NAV (Sheet 2 of 2).

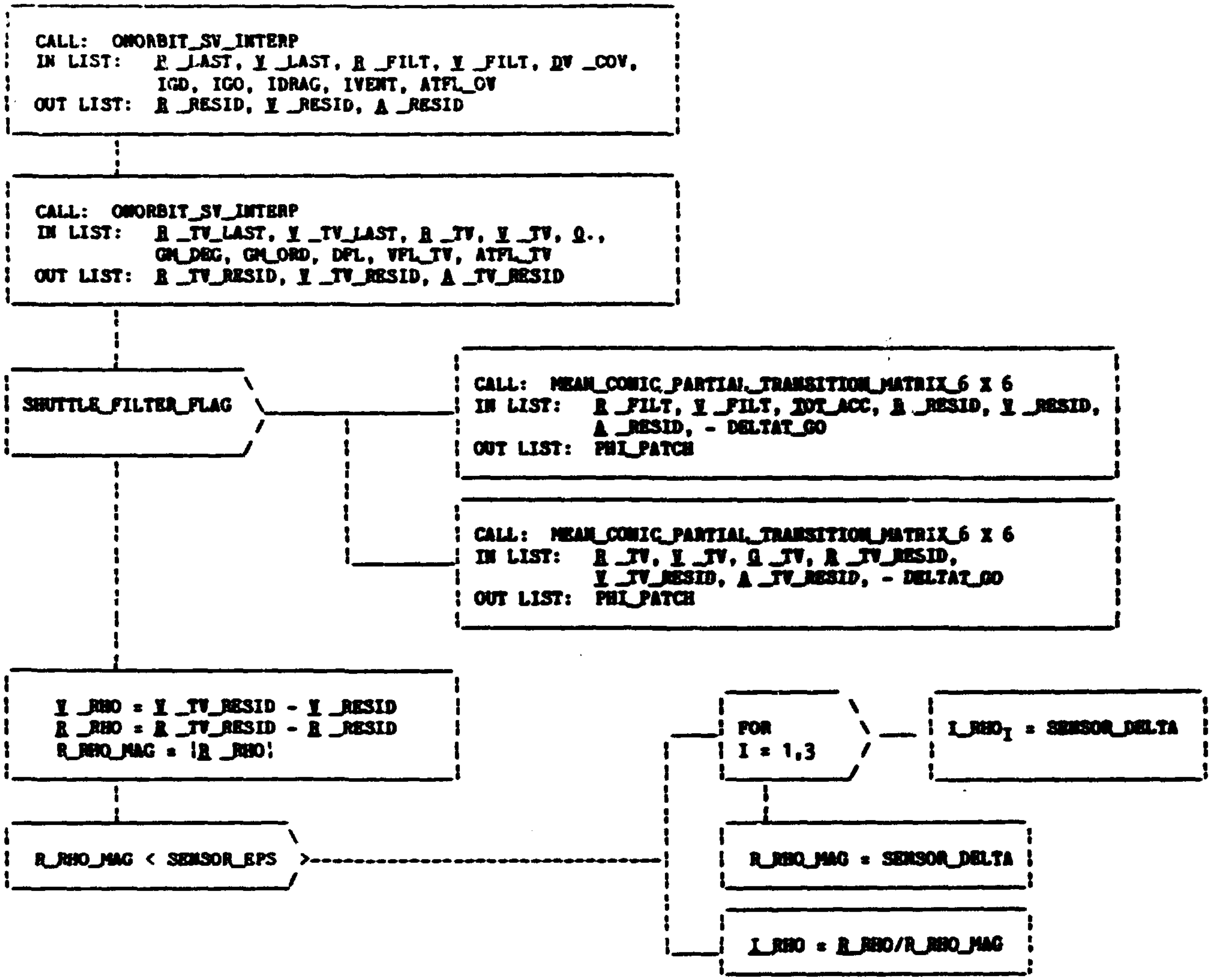


Figure B-40.- REND_NAV_INTERP.

IN LIST: R _ONE, V _ONE, R _TWO, V _TWO, V _IMU_DIF,
IGD, IGO, IDM, IVM, IATM
OUT LIST: R _RESID, V _RESID, A _RESID

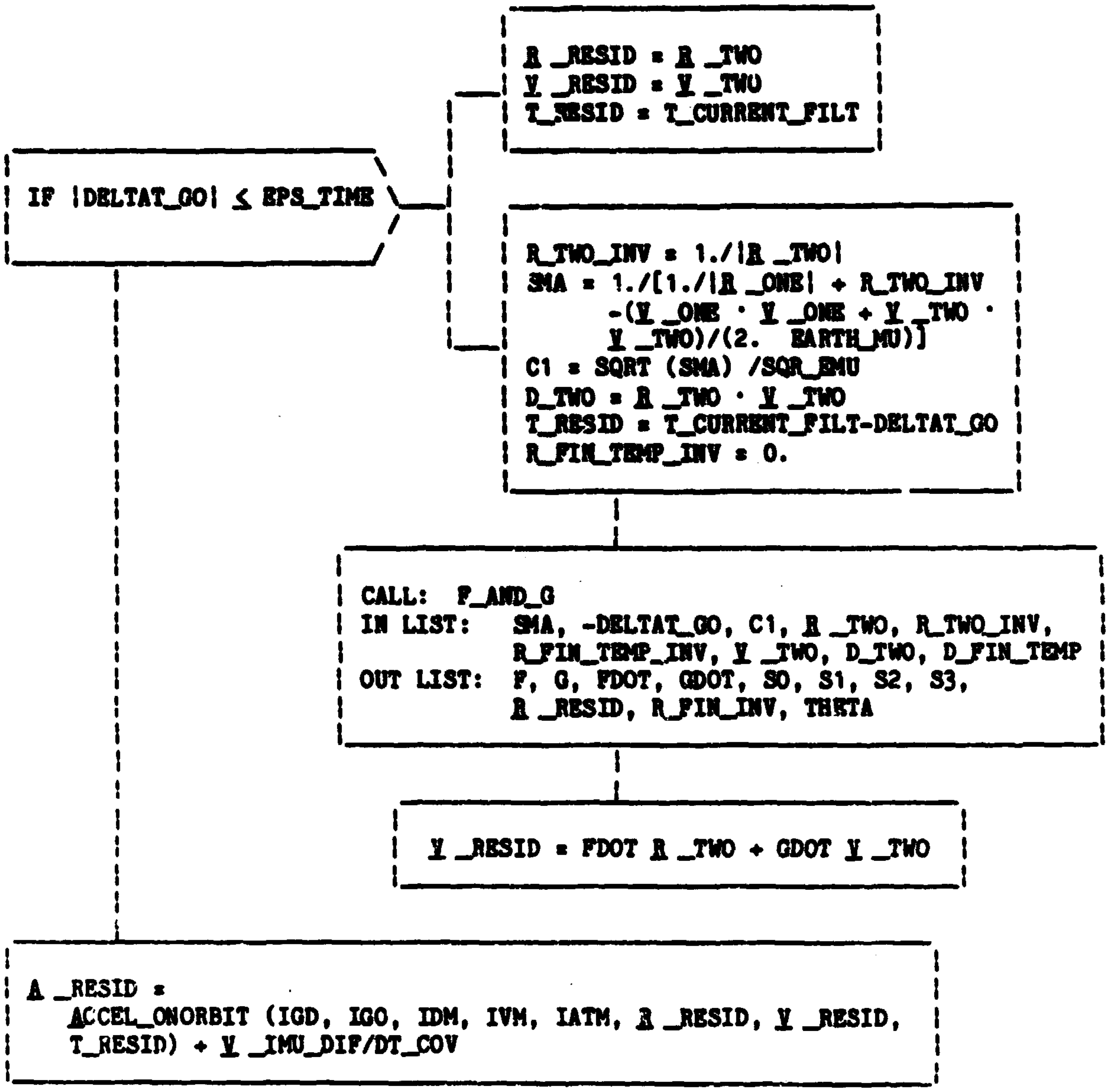


Figure B-41.- ONORBIT_SV_INTERP.

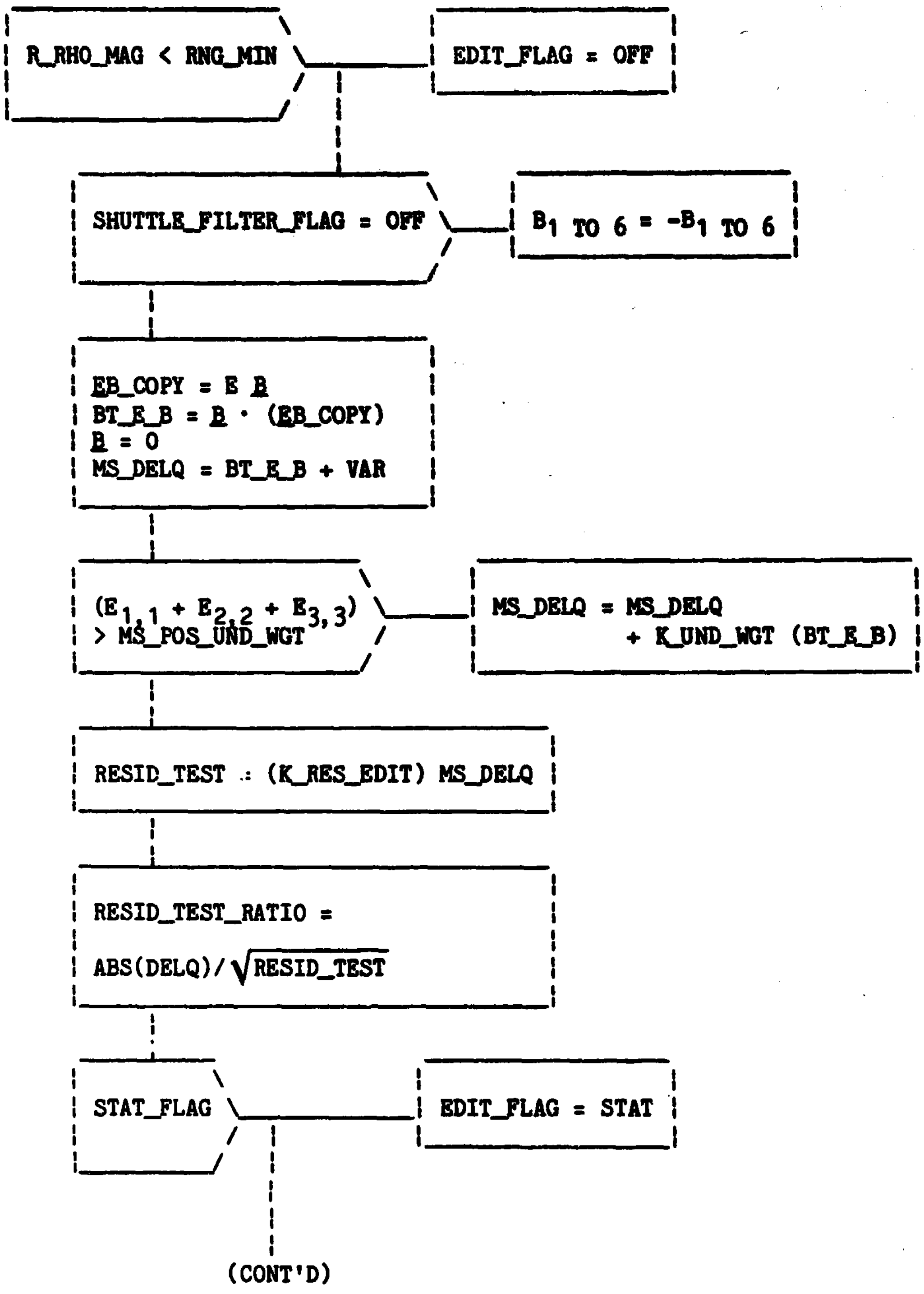


Figure B-42.- REND_NAV_FILTER (Sheet 1 of 2).

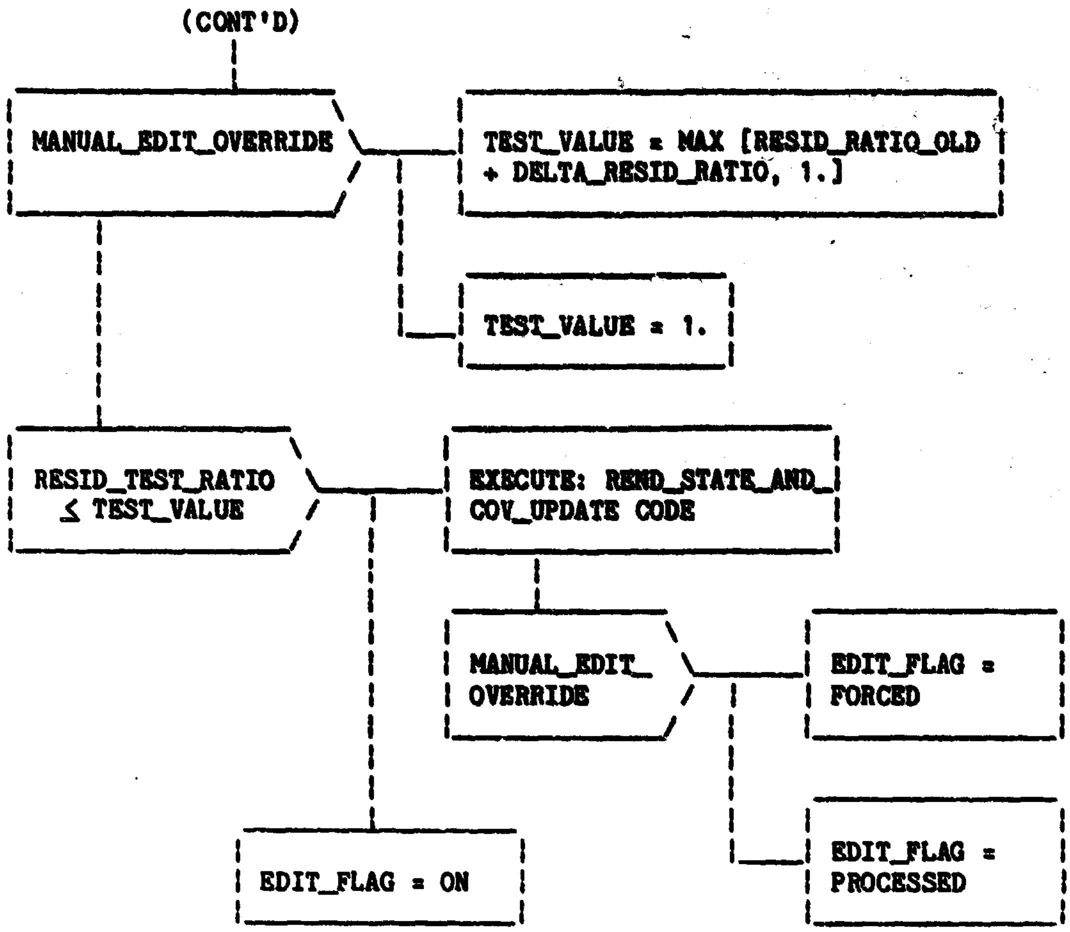


Figure B-42.- REND_NAV_FILTER (Sheet 2 of 2).

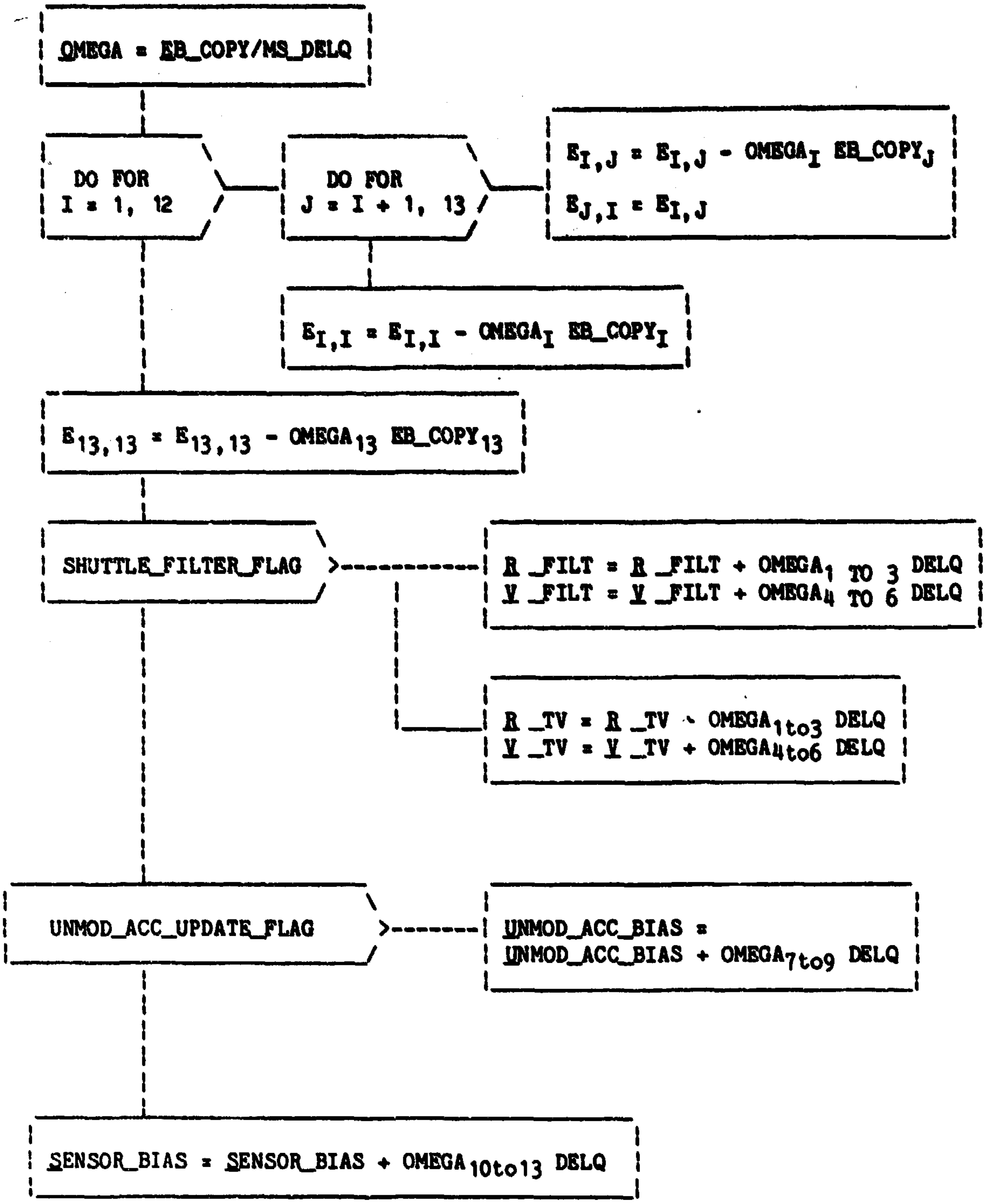


Figure B-43.- REND_STATE_AND_COV_UPDATE CODE.

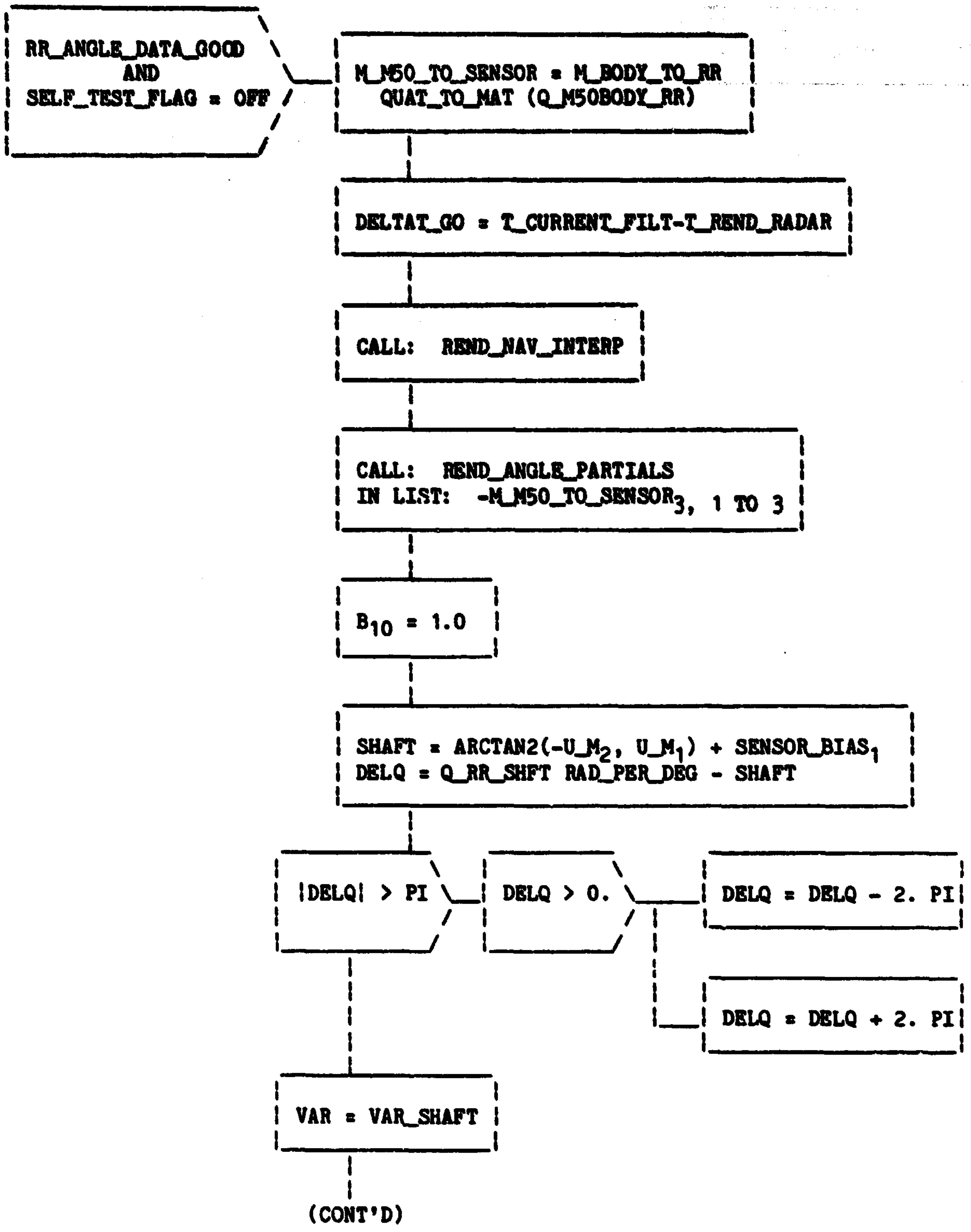


Figure B-44.- RR_ANGLE_NAV (Sheet 1 of 2).

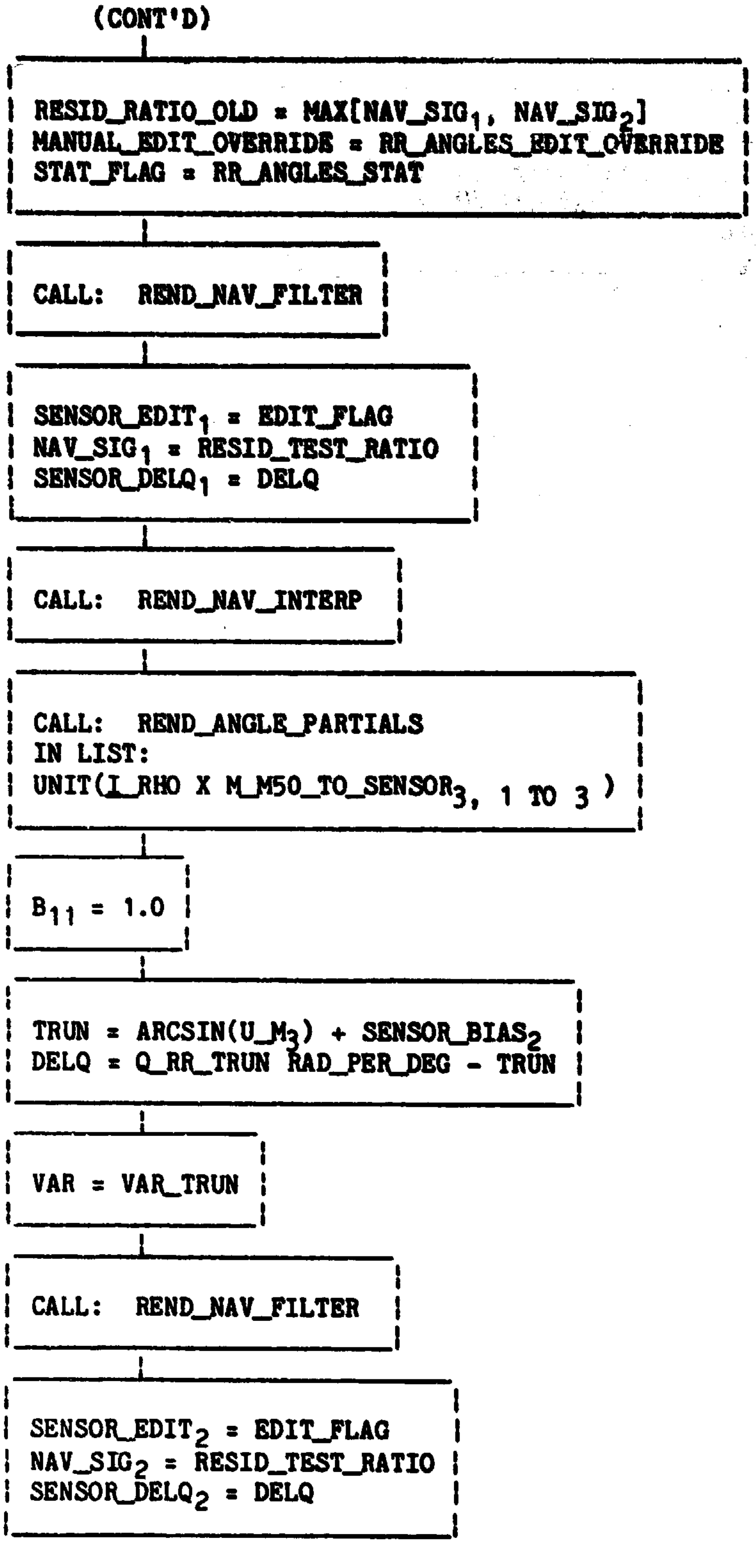


Figure B-44.- RR_ANGLE_NAV (Sheet 2 of 2).

IN LIST: $\underline{I}\_N$

$$\underline{RHO\_PLANE} = \underline{R}\_RHO - (\underline{R}\_RHO \cdot \underline{I}\_N)\,\underline{I}\_N$$

$$B_{1\ TO\ 6} = (PHI\_PATCH_{1\ TO\ 3,\ 1\ TO\ 6})^{T}[UNIT(\underline{RHO\_PLANE}\ X\ \underline{I}\_N) / |\underline{RHO\_PLANE}|]$$

$$\underline{U}\_M = M\_M50\_TO\_SENSOR\ \underline{I}\_RHO$$

Figure B-45.- REND_ANGLE_PARTIALS.

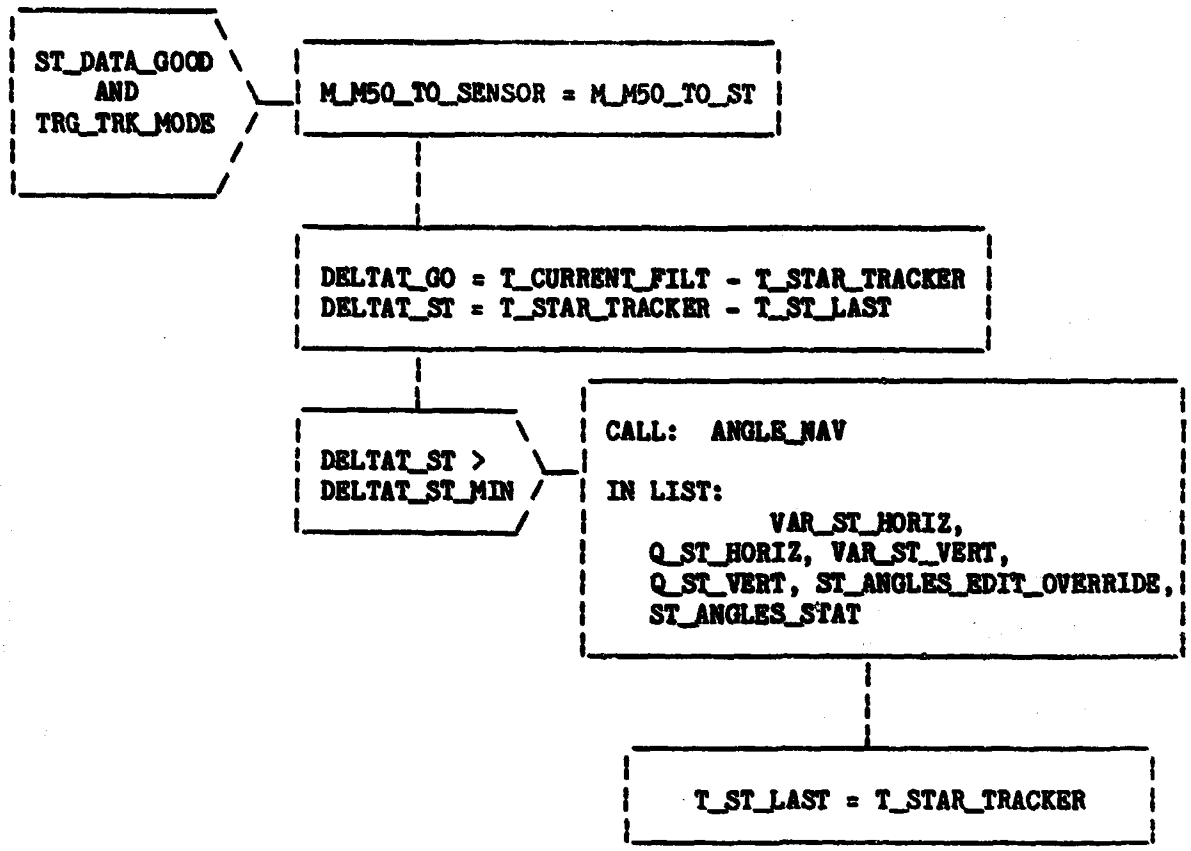
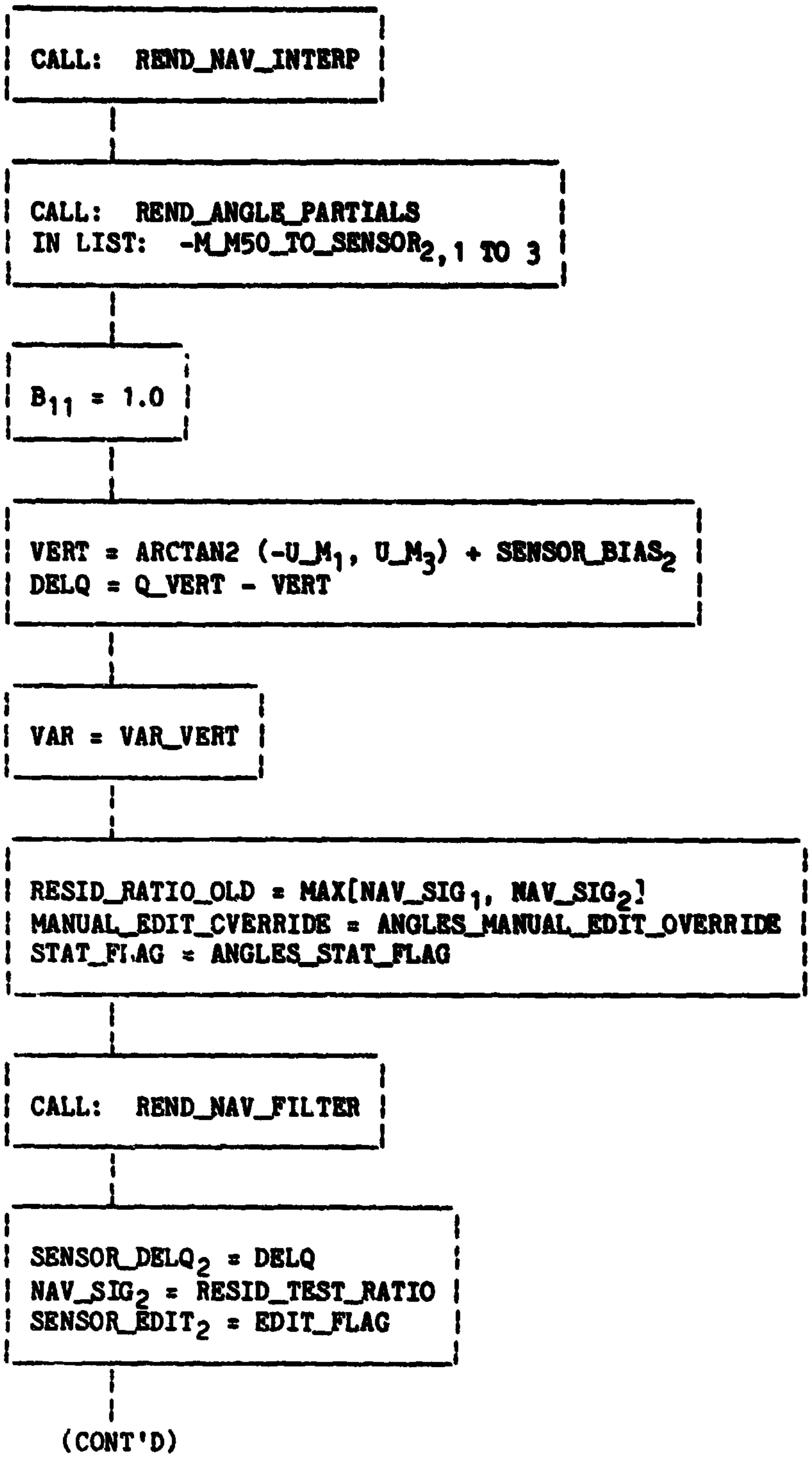


Figure B-46.- STAR_TRACKER_NAV.

IN LIST: VAR_HORIZ, Q_HORIZ, VAR_VERT, Q_VERT, ANGLES_MANUAL_EDIT_OVERRIDE, ANGLES_STAT_FLAG



```
CALL:  REND_NAV_INTERP

CALL:  REND_ANGLE_PARTIALS
IN LIST:  -M_M50_TO_SENSOR$_{2,1 TO 3}$

B$_{11}$ = 1.0

VERT = ARCTAN2 (-U_M$_1$, U_M$_3$) + SENSOR_BIAS$_2$
DELQ = Q_VERT - VERT

VAR = VAR_VERT

RESID_RATIO_OLD = MAX[NAV_SIG$_1$, NAV_SIG$_2$]
MANUAL_EDIT_CVERRIDE = ANGLES_MANUAL_EDIT_OVERRIDE
STAT_FLAG = ANGLES_STAT_FLAG

CALL:  REND_NAV_FILTER

SENSOR_DELQ$_2$ = DELQ
NAV_SIG$_2$ = RESID_TEST_RATIO
SENSOR_EDIT$_2$ = EDIT_FLAG

(CONT'D)
```

Figure B-47.- ANGLE_NAV (Sheet 1 of 2).

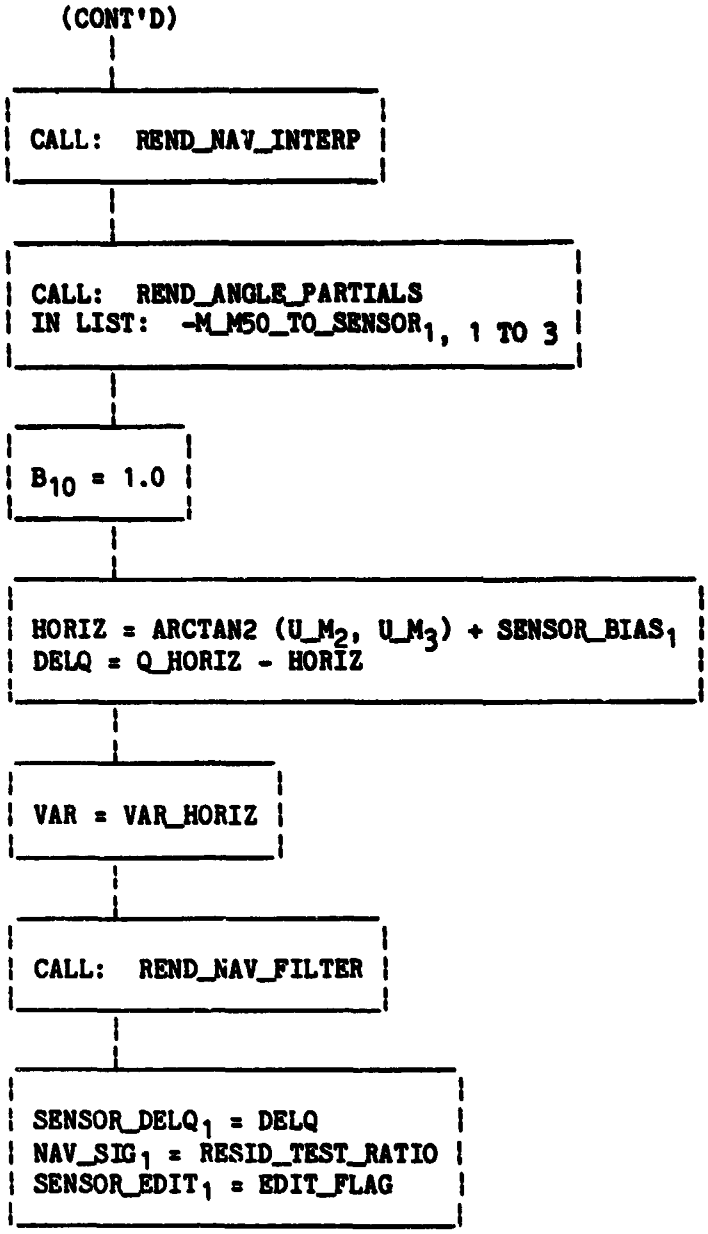


Figure B-47.- ANGLE_NAV (Sheet 2 of 2).

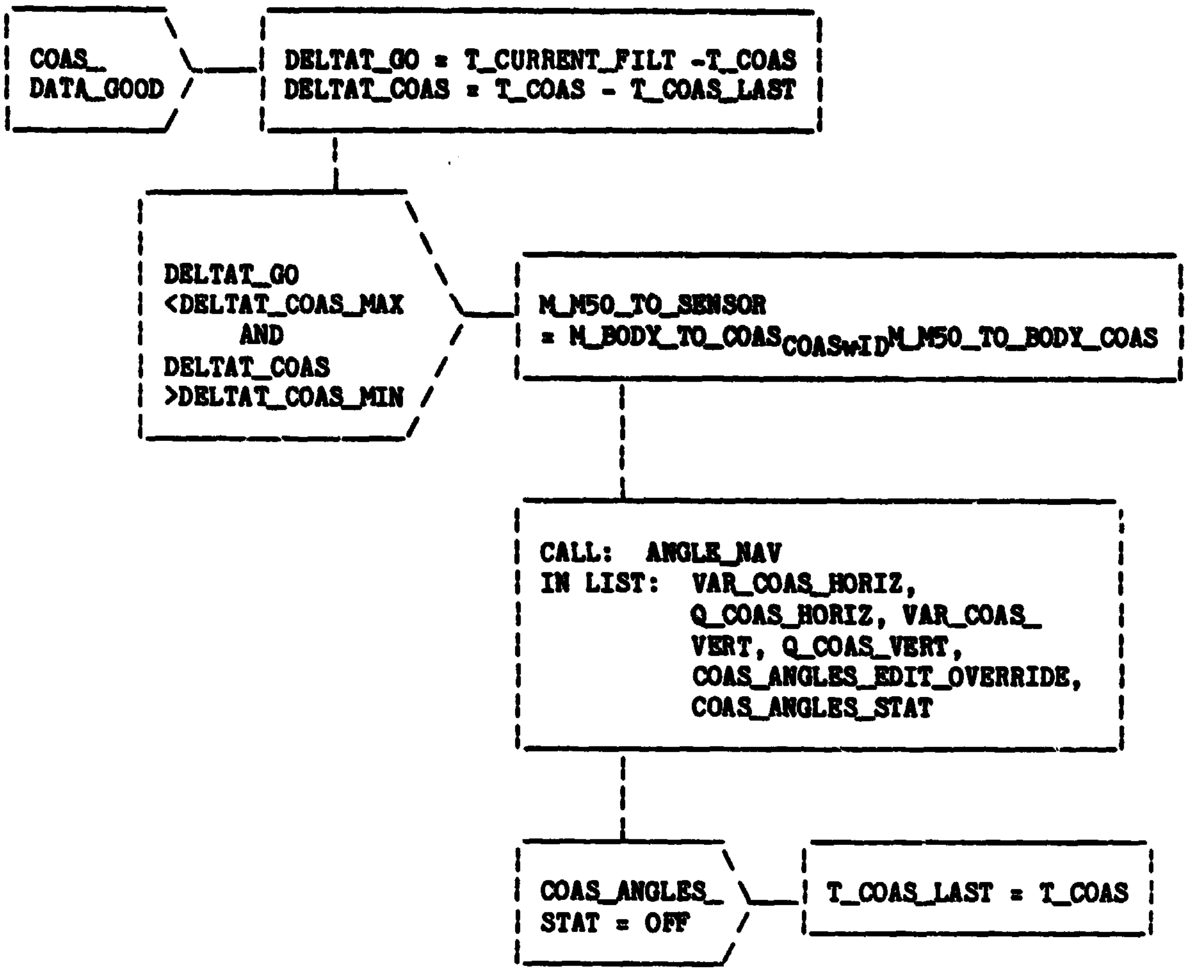


Figure B-48.- COAS_NAV.

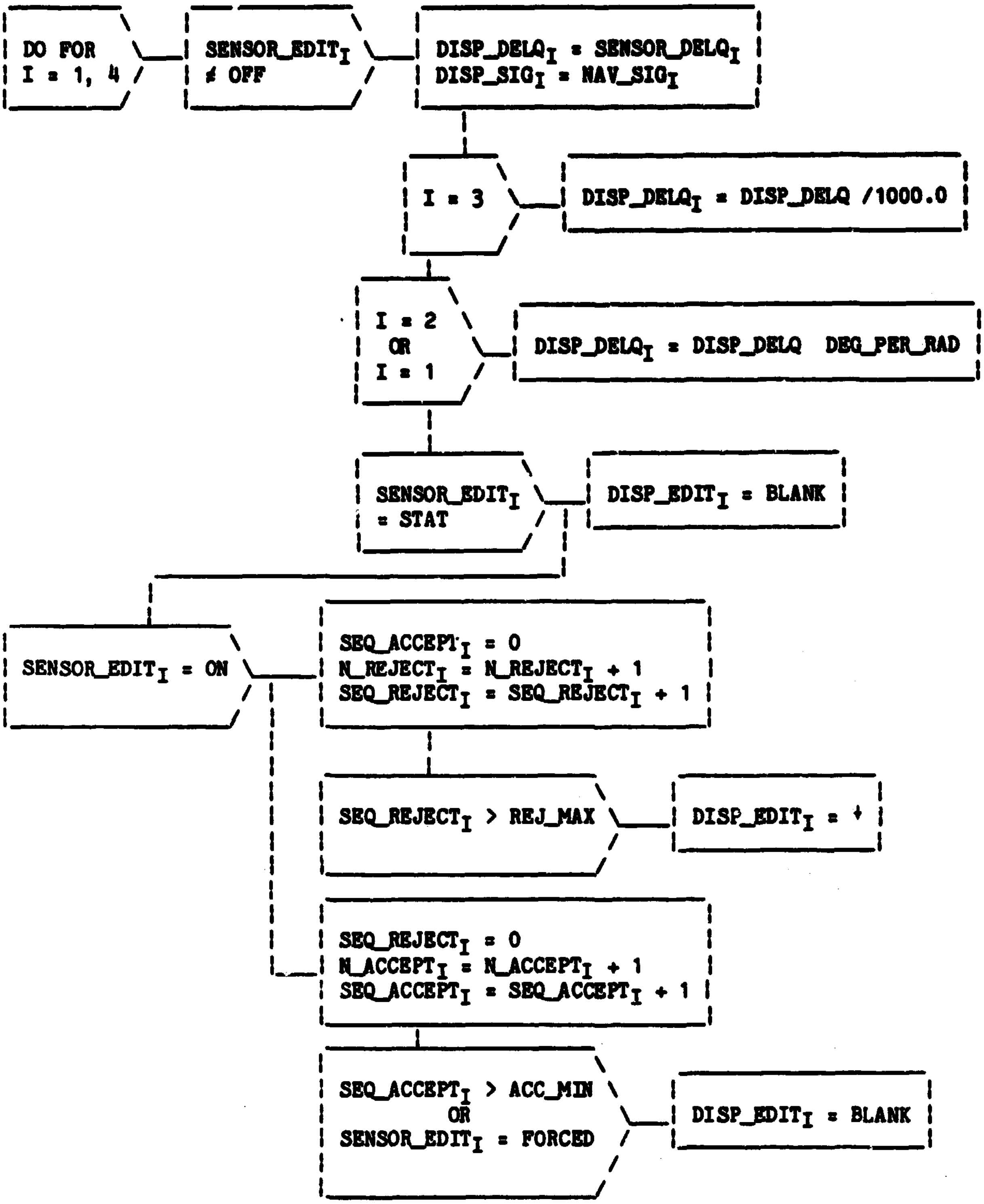


Figure B-49.- MEAS_PROCESSING_STATISTICS_REND.

APPENDIX C

GENERAL REQUIREMENT FUNCTIONS FLOWCHARTS, VARIABLE NAMES, AND DESCRIPTIONS

CONTENTS

PRECEDING PAGE BLANK NOT FILMED

VARIABLE LIST

PRECEDING PAGE BLANK NOT FILMED

VARIABLES LIST DEFINITIONS

CODE USED FOR VARIABLE DATA TYPE

F: floating point quantity; an n-dimensional floating point vector will be denoted F(n); similarly, an n x m floating point matrix will be denoted by F(n,m)

I: integer quantity; I(n) will denote an n-dimensional integer vector

B: bit, i.e., data having only values of 0 or 1

C: character; C(n) will denote an n-dimensional character string

CODE USED FOR VARIABLE PRECISION

D: double precision

S: single precision; integer quantities are assumed single precision unless otherwise specified

VARIABLE LOCATION

COMPOOL: Variable value located in common storage, accessible by all functions

LOCAL : Variable is used by one function only, and usable to other functions through call argument only

VARIABLE INITIALIZATION CATEGORY

blank: display is vacant

C: constant (unchanging)

DD: design dependent

HC: hard coded

MD: mission dependent (I-LOAD)

[illegible]

VARIABLE INITIAL VALUE

initial operation sequence computer inputs

VARIABLE UPLINK AND DOWNLIST STATUS

UPLINK: variable is an uplink item

DOWNLIST: variable is a downlist item

UNITS DEFINITIONS

deg: angular measurement, degrees

ft: feet

lb: pounds

n.d. (or -) non-dimensional

rad: radian

sec: time measurement, seconds

slugs: mass measurement, slugs

vary: units have different values which depend on variable use

PRECEDING PAGE BLANK NOT FILMED

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial-ization category	Initial value	Uplink/ downlist	Units	Description
A	SF(3,3)	local	-	-	-	-	Temporary matrix used in QUAT_TO_MAT
AA (V96U9057C-60C)	DF(4)	local	DD	1/2 $1 - \sqrt{1/2}$ $1 + \sqrt{1/2}$ 1/6	-	-	Array of coefficients required by the RK_GILL integrator
ALT	DF	local	-	-	-	ft	Current Orbiter altitude above reference ellipsoid
A_0	DF	local	-	-	-	-	One minus eccentricity squared of Earth (ellipsoid), 1.0-FLATCON, initialized only
ATM	I	local	-	-	-	-	Flag indicating if current or prestored configuration constants are to be used in drag computations within ACCEL_ONORBIT
ATMP (V94J3999C)	I	compool	-	-	downlist	-	Flag indicating vehicle attitude mode to be used for prediction
AZ	SF	local	-	-	-	rad	Bearing from true north (EF_TO_RUNWAY, EF_TO_SCANNER)
BB (V96U9080C-3C	DF(4)	local	DD	0. 1. 1. 2.	-	-	Array of coefficients used by the RK_GILL integrator
CAZ	SF	local	-	-	-	-	Cosine of AZ
CC (V96U9090C-3C)	DF(4)	local	DD	1. $2(1-\sqrt{1/2})$ $2(1+\sqrt{1/2})$ 0.	-	-	Array of coefficients used by the RK_GILL integrator
CLAT	SF	local	-	-	-	-	Cosine of LAT_GEOD (EF_TO_TOPDET)
CLON	SF	local	-	-	-	-	Cosine of LON (EF_TO_TOPDET)
COS_P	SF	local	-	-	-	-	Cosine of geocentric latitude to Shuttle

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial-ization category	Initial value	Uplink/ downlist	Units	Description
C1	DF	local	-	-	-	-	Auxiliary variable used in F and G series computations and in Pines method
C2	DF	local	-	-	-	-	Auxiliary variable used in F and G series computations and in Pines method
C3	DF	local	-	-	-	-	Auxiliary variable used in Pines variation of parameters method
C4	DF	local	-	-	-	-	Auxiliary variable used in Pines variation of parameters method
C5	DF	local	-	-	-	-	Auxiliary variable used in Pines variation of parameters method
D_AUX	DF	local	-	-	-	ft^2/sec	Dot protect of velocity vector and perturbing acceleration, used in Pines method
D_TAU	DF	local	-	-	-	ft^2/sec^2	Variable used in Pines method
DD	DF	local	-	-	-	ft^2/sec	Temporary variable used in Pines method
$\overline{DD}$ (V96U9103C-6C)	DF(4)	local	DD	0. $-2 + 3\sqrt{\frac{1}{2}}$ $-2 - 3\sqrt{\frac{1}{2}}$ 0.	-	-	Array of coefficients used by the RK_GILL integrator
DEL	SF	local	-	-	-	ft	Radius correction factor at geocentric radius of Earth
DEL_LAT	SF	local	-	-	-	rad	Correction angle from geocentric latitude of radius vector to Shuttle to geodetic subvehicle latitude
DELTAT	DF	local	-	-	-	sec	Time interval between two positions in a conic (F and G series)
DERIV	DF(7)	local	-	-	-	ft/sec ft/sec^2 n.d.	Temporary storage for derivatives output from Pines Method
D_FIN_TEMP	DF	compool	HC	0.	-	ft^2/sec	Temporary storage for D_FIN (Pines method)
D_IN	DF	local	-	-	-	ft^2/sec	Dot product of position and velocity used in Pines method in call to F and G series

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial-ization category	Initial value	Uplink/ downlist	Units	Description
DMP (V94X3969X)	B	compool	-	-	-	-	Flag indicating if model for acceleration due to drag is to be used for prediction
DT_MAX (V96U9227C)	DF	local	DD	180.	-	sec	Maximum integration step size used for prediction
DT_STEP	DF	local	-	-	-	sec	Integration step size for prediction or propagation
DUM	DF	local	-	-	-	-	Dummy variable (GEODETIC_TO_EF)
DUM1	DF	local	-	-	-	ft	Dummy variable (GEODETIC_TO_EF)
EARTH_RADIUS_EQUATOR	DF	compool	C	-	-	ft	Earth's equatorial radius
EARTH_RATE	DF	compool	C	-	-	rad/sec	Earth's rotation rate
ELLIPT	DF	compool	C	-	-	-	Earth's ellipticity constant
F	DF	local	-	-	-	-	Closed form version of the F time series
FDOT	DF	local	-	-	-	sec^{-1}	Closed form version of the time derivative of F series
FD_TAU	DF	local	-	-	-	-	Variable used in Pines method
FLATCON (V96U9216C)	DF	local	DD	.006693 42162	-	-	Eccentricity squared of reference ellipsoid (FISCHER), initialized only
F_TAU	DF	local	-	-	-	-	Auxiliary variable used in Pines method
G	DF	local	-	-	-	sec	Closed form version of the G time series
G _CENTRAL	DF(3)	compool	-	-	-	ft/sec^2	Gravitational acceleration due to Earth as a point mass
GDOT	DF	local	-	-	-	-	Closed form version of the time derivative of the G series
GD_TAU	DF	local	-	-	-	-	Auxiliary variable used in Pines method
GMDP (V94J3975C)	I	compool	-	-	-	-	Flag indicating degree of gravitational potential model used for prediction

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial-ization category	Initial value	Uplink/ downlist	Units	Description
GMOP (V94J3963C)	I	compool	-	-	-	-	Flag indicating order of gravitational potential model used for prediction
G_TAU	DF	local	-	-	-	-	Perturbation derivative of GDOT in Pines method
I	I	local	-	-	-	-	Counter
LAM	SP	local	-	-	-	rad	Earth's rotation angle from epoch
LAT_GEOD	SP	local	-	-	-	rad	Geodetic latitude of R _EF
LON	SP	local	-	-	-	rad	Longitude of R _EF
M	SP(3,3)	local	-	-	-	-	General transformation matrix
MAT	SP(3,3)	local	-	-	-	-	Transformation matrix (SBODY_TO_M50)
MATRIX	SP(3,3)	local	-	-	-	-	Temporary matrix
M_M50TOEF_AT_EPOCH (V97U5738C-46C)	DP(3,3)	compool	MD	-	-	-	Transformation matrix from M50 to Earth-fixed at T_EPOCH
N_STEPS	I	local	-	-	-	-	Number of integration steps in prediction or propagation interval
P	DP	local	-	-	-	-	Variable used in the RK_GILL integrator
P	DP(3)	local	-	-	-	ft/sec	Variable used in Pines method as perturbing acceleration
PHI	SP	local	-	-	-	rad	Geocentric latitude of position vector to Shuttle
PRED_ORB_AREA (V93U6955C)	SP	compool	-	-	downlist	ft^2	Orbiter cross-sectional area
PRED_ORB_CD (V93U6954C)	SP	compool	-	-	downlist	--	Orbiter coefficient of drag
PRED_ORB_MASS (V93U6953C)	SP	compool	-	-	downlist	slugs	Orbiter mass

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial- ization category	Initial value	Uplink/ downlist	Units	Description
PRED_STEP (V94W3964C)	SF	compool	-	-	downlist	sec	Input integration step size for prediction
P1 P2 P3 P4 P5 P6	SF	local	-	-	-	-	Temporary variables
Q	SF(4)	local	-	-	-	-	Input quaternion to QUAT_TO_MAT
Q	DF(7)	local	-	0.	-	-	Local array used in RK_GILL integration
Q_FIFTY_BODY	SF(4)	local	-	-	-	-	Quaternion representing current Orbiter attitude
R	DF	local	-	-	-	ft	Radius vector to Shuttle in Greenwich true-of-date system
R	DF(3)	local	-	-	-	ft	Shuttle position vector
PRED_TIME_TOL	SF	local	HC	10^{-8}	-	sec	Time tolerance for predictor
RAD_P	DF	local	-	-	-	ft	Geocentric radius of Earth at geocentric latitude of Shuttle
R _EF	DF(3)	local	-	-	-	ft	Position vector of Shuttle in Earth-fixed coordinates
R _EF_EQUAT	DF	local	-	-	-	ft	Position of vehicle position vector in the equatorial plane
R_FIN_INV	DF	local	-	-	-	ft^{-1}	Reciprocal of the magnitude of R _FIN
R_FIN_TEMP_INV	DF	local	-	-	-	ft^{-1}	Dummy variable used in the call to F_AND_G
R _IN	DF(3)	local	-	-	-	ft	Position vector at the beginning of a time interval in M50. Used in F and G series and in SUPER_G
R_IN_AUX	DF	local	-	-	-	ft^{-1}	Reciprocal of the magnitude of the position vector
R_IN_INV	DF	local	-	-	-	ft^{-1}	Reciprocal of the magnitude of R _IN

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
R_IN_TAU	DF	local	-	-	-	-	Auxiliary variable used in the Pines method
R _PRED_FINAL (V95H0811C -3C)	DF(3)	compool	-	-	downlist	ft	Orbiter or target position vector at T_PRED_FINAL
R _PRED_INIT (V94H4002C-4C)	DF(3)	compool	-	-	downlist	ft	Orbiter or target position vector at T_PRED_INIT
R_XY	DF	local	-	-	-	ft^2	Projection squared of radius vector in X-Y plane of Greenwich true-of-date system
SAZ	SF	local	-	-	-	-	Sine of AZ
SIN_P	SF	local	-	-	-	-	Sine of geocentric latitude to Shuttle
SLAT	SF	local	-	-	-	-	Sine of geocentric latitude
SLON	SF	local	-	-	-	-	Sine of longitude
SMA	DF	compool	-	-	-	ft	Semimajor axis of conic
SQR_EMU (V90U1241C)	DF	compool	-	-	-	$\frac{ft^{3/2}}{sec}$	Square root of the Earth's gravitational constant
S0 S1 S2 S3	DF	local	-	-	-	- vary vary vary	Auxiliary variables used in F and G series and Pines computations, S1 and S2 also used as auxiliary variables in mean conic partials transition matrix computations
S4 S5 S6	DF	local	-	-	-	vary	Auxiliary variables used in Pines method
T_ACCEL	DF	local	-	-	-	sec	Time of acceleration function evaluation in Pines method
T_CUR	DF	local	-	-	-	sec	Current integration time within the predictor or propagator
TEMP	DF	local	-	-	-	-	Temporary variable

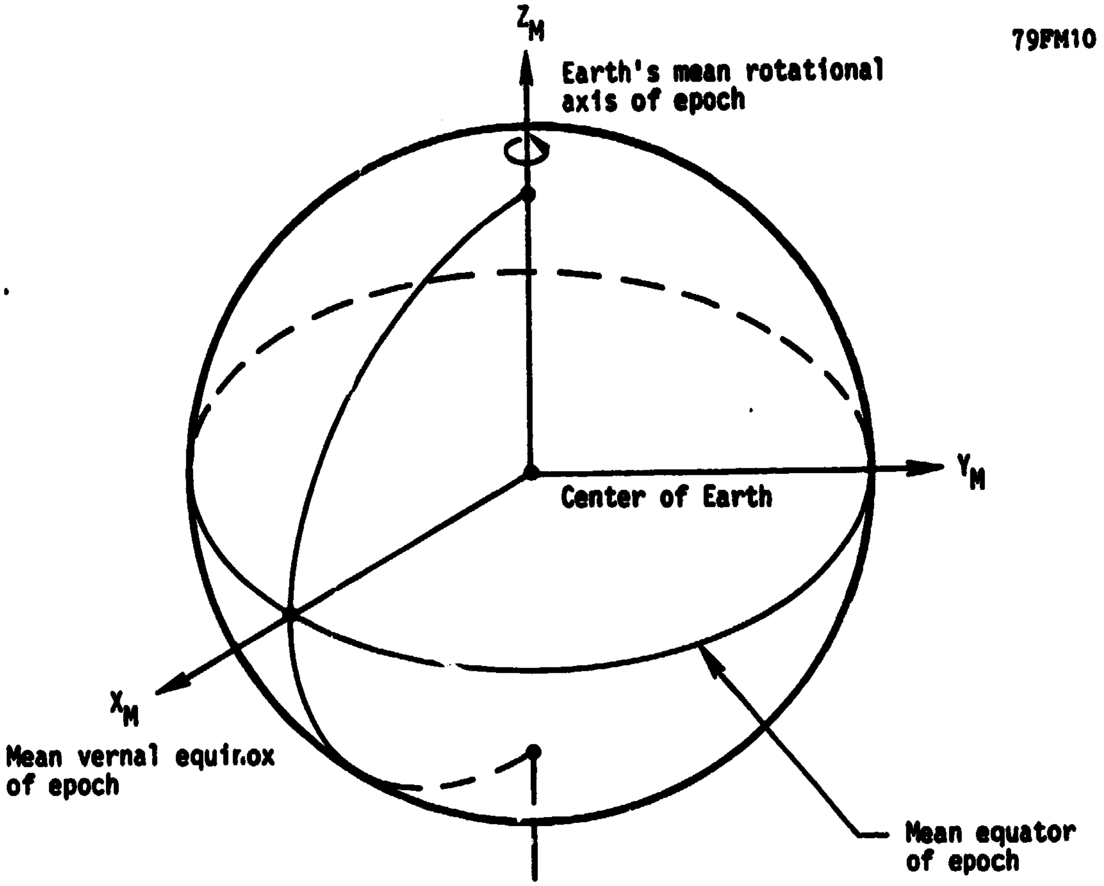
VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial-ization category	Initial value	Uplink/ downlist	Units	Description
T_EPOCH (V97U5528C)	DF	local	MD	-	-	sec	Time of M_M50TOEF_AT_EPOCH
THETA	DF	local	-	-	-	rad	Difference of eccentric anomaly
TIME	DF	local	-	-	-	sec	General time variable
TIME_DEL	DF	local	-	-	-	sec	Difference between initial and final predictor times
T_IN	DF	local	-	-	-	sec	Initial time input for state propagation, used in SUPER_G
T_PRED_FINAL (V94W3979C)	DF	compool	-	-	downlist	sec	Final time input for onorbit prediction
T_PRED_INIT (V94W3970C)	DF	compool	-	-	downlist	sec	Initial time input for onorbit prediction
T_STOR	DF	local	-	-	-	sec	Initial time of each Runge-Kutta integration step
U,Z,W	DF(3)	local	-	-	-	-	Columns of the UVW_TO_M50 temporary matrix
V	DF(3)	local	-	-	-	ft/sec	Shuttle velocity vector (local)
VM	B	local	-	-	-	-	Flag used in ACCEL_ONORBIT to indicate if Vent/RCS model is to be used
VMP (V94X3971X)	B	compool	-	-	-	-	Flag indicating whether venting accelerations are to be modeled for prediction
V_PRED_FINAL (V95L0814C -6C)	DF(3)	compool	-	-	downlist	ft/sec	Orbiter or target velocity vector at T_PRED_FINAL
V_PRED_INIT (V94L4006C-8C)	DF(3)	compool	-	-	downlist	ft/sec	Orbiter or target velocity vector at T_PRED_INIT
X	DF(6)	local	-	-	-	ft,ft,ft, ft/sec, ft/sec' ft/sec	Temporary array for Shuttle or target state vector used in Pines method

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
XN	DP(7)	local	-	-	-	ft,ft,ft, ft/sec, ft/sec, ft/sec, sec	Array of integrated initial conditions for onorbit prediction and propagation

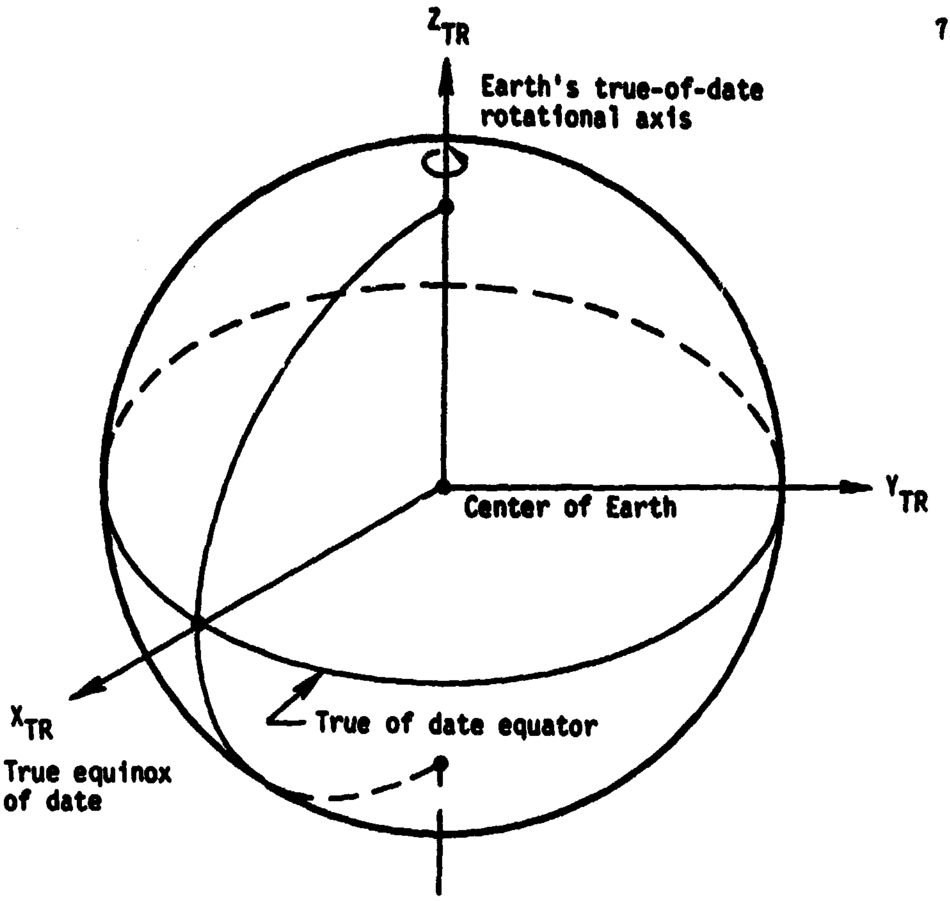
COORDINATE SYSTEMS



NAME:	Aries mean of 1950, Cartesian, coordinate system
ORIGIN:	The center of the Earth
ORIENTATION:	The epoch is the beginning of Besselian Year 1950 or Julian ephemeris date 2433282.423357. The X_M-Y_M plane is the mean Earth's equator of epoch. The X_M axis is directed toward the mean vernal equinox of epoch. The Z_M axis is directed along the Earth's mean rotational axis of epoch and is positive north. The Y_M axis completes a right-handed system.
CHARACTERISTICS:	Inertial, right-handed Cartesian system

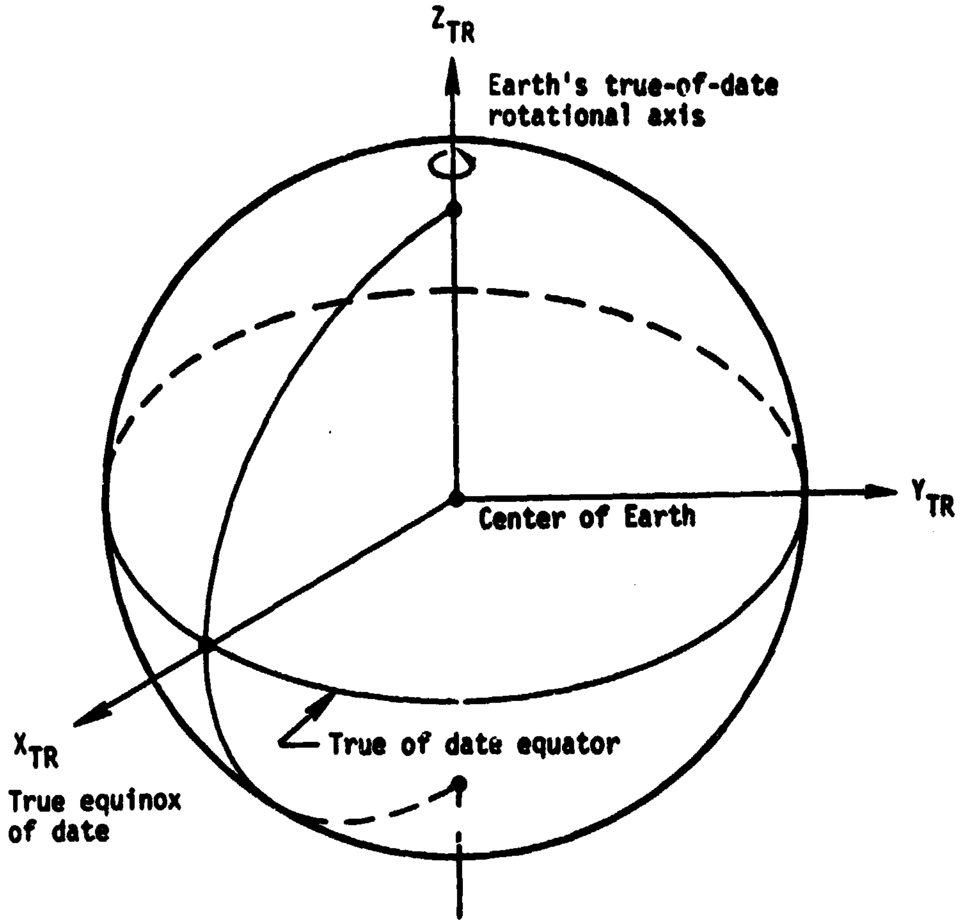
Figure C-1.- Aries Mean of 1950, Cartesian.

79FM10



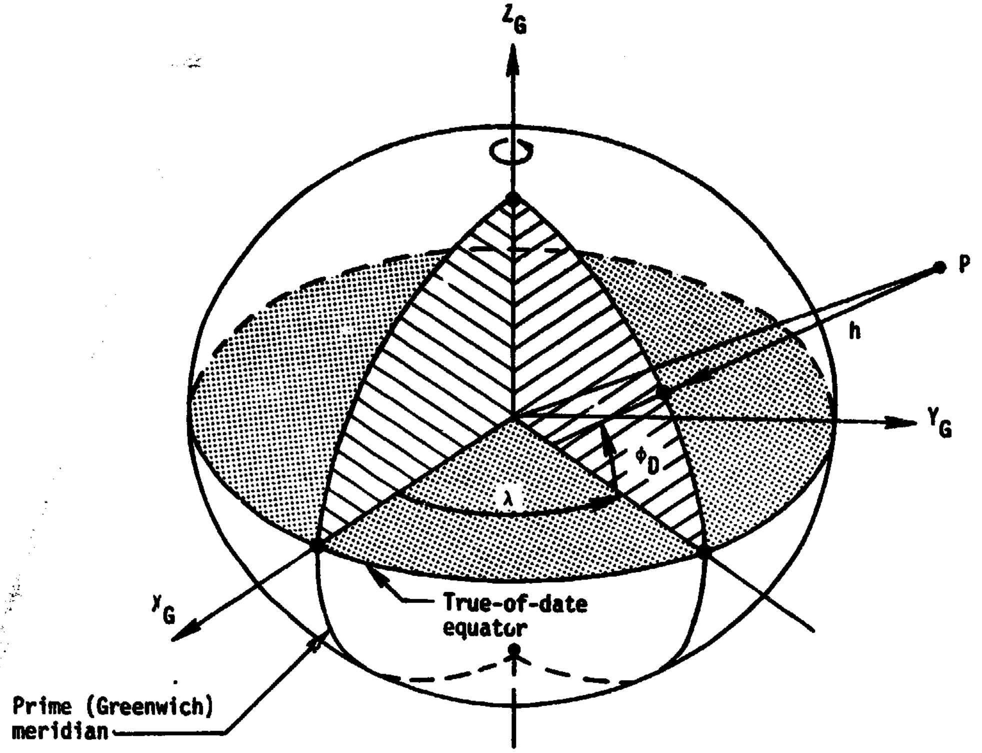
NAME:	Earth fixed (Greenwich true of date) coordinate system
ORIGIN:	The center of the Earth
ORIENTATION:	The X_G-Y_G plane is the Earth's true-of-date equator. The Z_G axis is directed along the Earth's true-of-date rotational axis and is positive north. The $+X_G$ axis is directed toward the prime meridian. The Y_G axis completes a right-handed system.
CHARACTERISTICS:	Rotating, right-handed Cartesian. Velocity vectors expressed in this system are relative to a rotating reference frame fixed to the Earth, whose rotation rates are expressed relative to the Aries mean of 1950 system.

Figure C-2.- Earth-Fixed Greenwich True of Date.



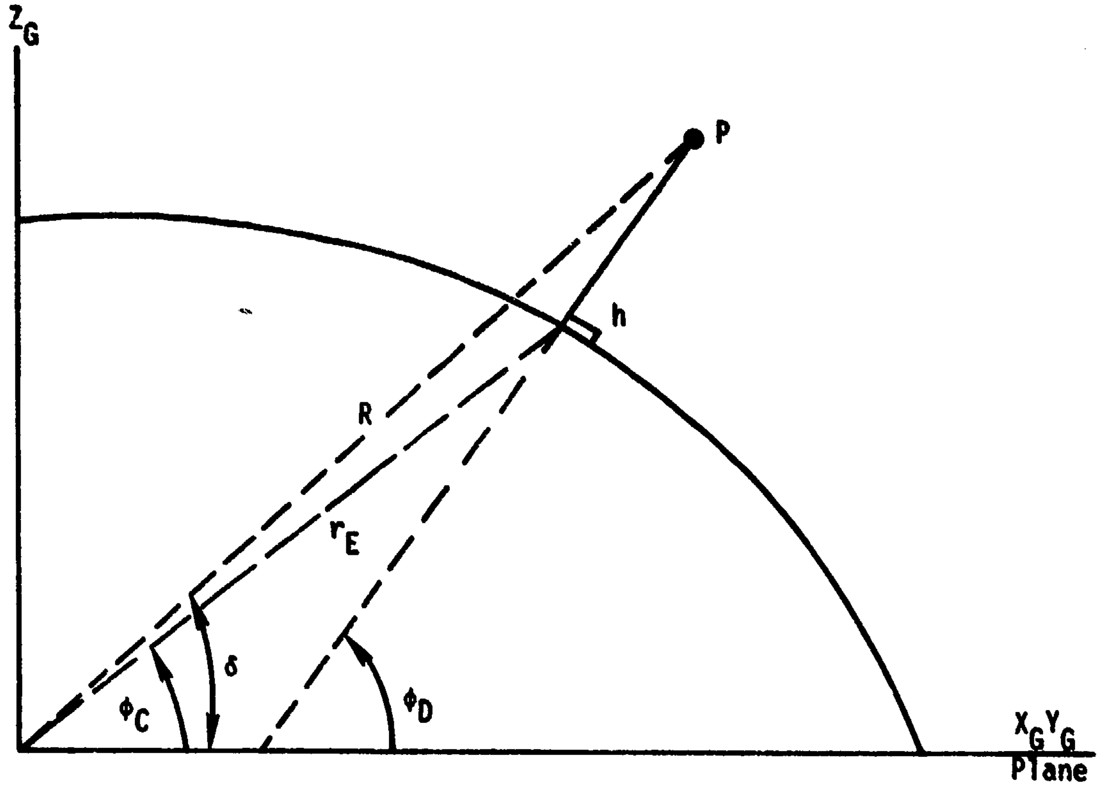
NAME:	Aries true of date, Cartesian, coordinate system
ORIGIN:	The center of the Earth
ORIENTATION:	The epoch is the current time of interest. The X_{TR}-Y_{TR} plane is the Earth's true equator of epoch. The X_{TR} axis is directed toward the true vernal equinox of epoch. The Z_{TR} axis is directed along the Earth's true rotational axis of epoch and is positive north. The Y_{TR} axis completes the right-handed system.
CHARACTERISTICS:	Quasi-inertial, right-handed Cartesian

Figure C-3.- Aries True of Date, Cartesian.



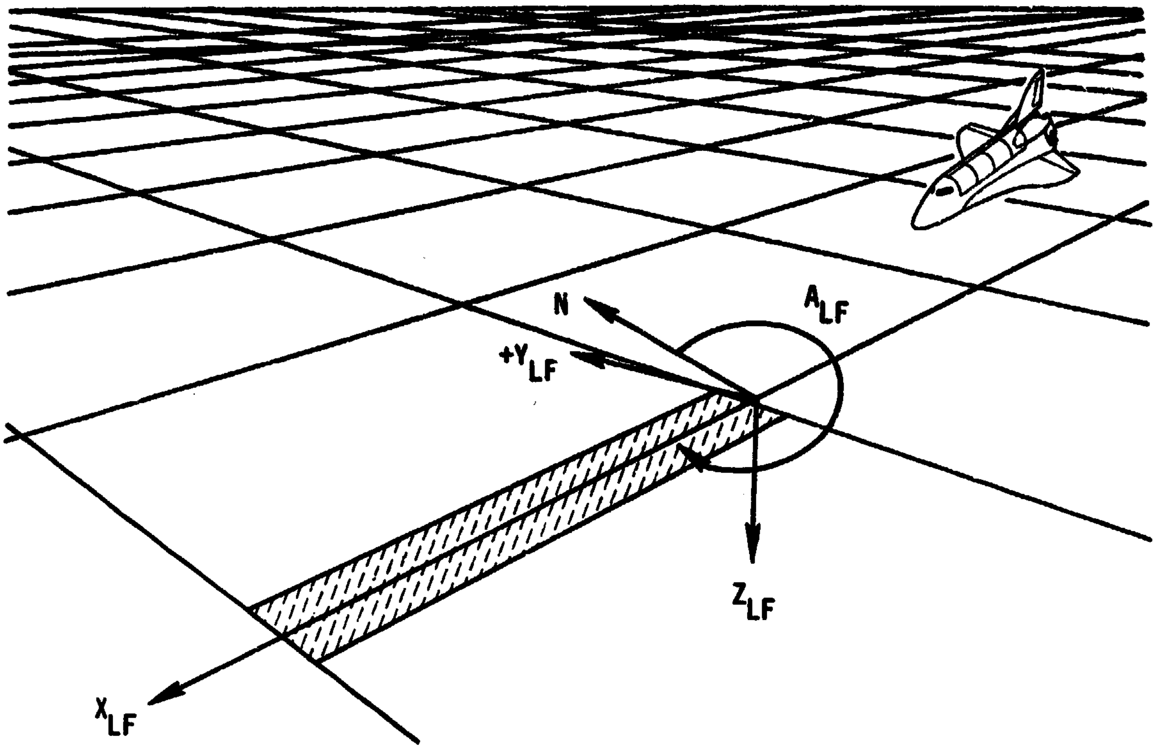
NAME:	Geodetic parameters
ORIGIN:	This system consists of a set of parameters rather than a coordinate system; therefore, no origin is specified.
ORIENTATION:	This system of parameters is based on an ellipsoidal model of the Earth (e.g., the Fischer ellipse of 1960). For any point of interest we define a line known as the geodetic local vertical which is perpendicular to the ellipsoid and which contains the point of interest. h, geodetic altitude, is the distance from the point of interest to the reference ellipsoid, measured along the geodetic local vertical, and is positive for points outside the ellipsoid. λ is the longitude measured in the plane of the Earth's true equator from the Prime (Greenwich) Meridian to the local meridian, measured positive eastward. ϕ_D is the geodetic latitude, measured in the plane of the local meridian from the Earth's true equator to the geodetic local vertical, measured positive north from the equator.
CHARACTERISTICS:	Rotating polar coordinate parameters. Only position vectors are expressed in this coordinate system. Velocity vectors should be expressed in the Aries mean of 1950 or the Aries true-of-date polar for inertial or quasi-inertial representations, respectively.

Figure C-4.- Geodetic Parameters (Basic Definitions) (Sheet 1 of 2)



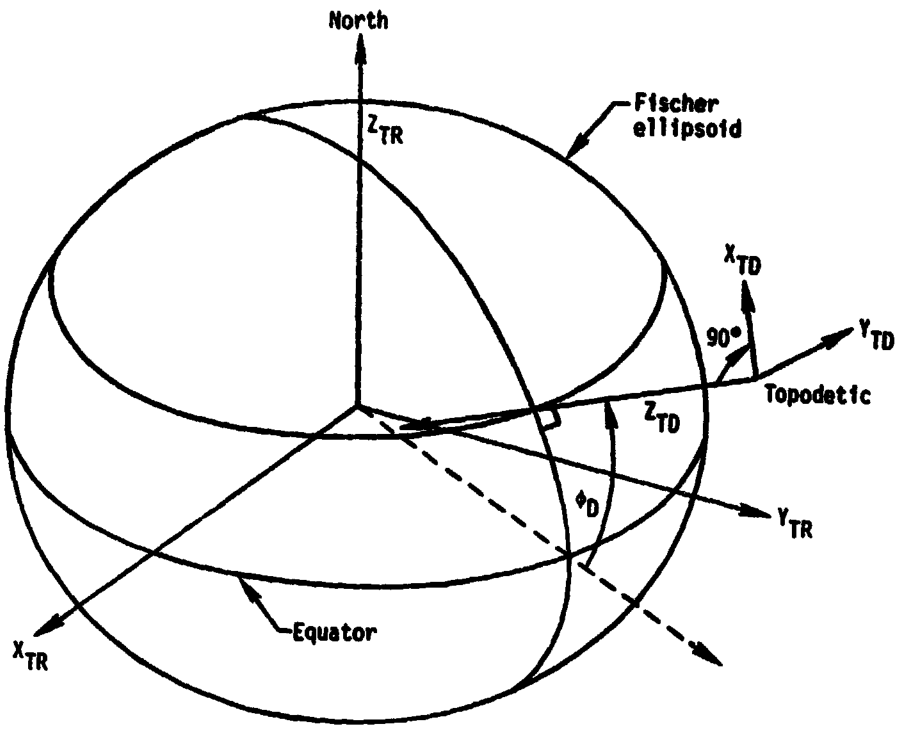
NAME:	Geodetic parameters
DEFINITIONS:	h is the altitude of Point P, measured perpendicular from the surface of the referenced ellipsoid. ϕ_D is the geodetic latitude of Point P. ϕ_C is the geocentric latitude of Point P. δ is the angle between radius vector and equatorial plane (declination). λ is the longitude of Point P. Angle (+ east) between plane of the figure and the plane formed by the Greenwich Meridian.

Figure C-4.- Geodetic Parameters (Detailed Explanation) (Sheet 2 of 2).



NAME:	Runway coordinate system
ORIGIN:	Runway center at approach threshold
ORIENTATION AND DEFINITIONS:	Z_{LF} axis is normal to the ellipsoid model through the runway centerline at the approach threshold and positive toward the center of the Earth. X_{LF} axis is perpendicular to the Z_{LF} axis and lies in a plane containing the Z_{LF} axis and the runway centerline (positive in the direction of landing). Y_{LF} axis completes the right-handed system. A_{LF} is the runway azimuth measured in the $X_{LF}Y_{LF}$ plane from true north to the $+X_{LF}$ axis (positive clockwise).
CHARACTERISTICS:	Rotating, Earth-referenced

Figure C-5.- Runway Coordinate System.



NAME:	Topodetic coordinate system
ORIGIN:	Orbiter center of mass*
ORIENTATION:	Z_{TD} is normal to a geodetic local tangent plane and is positive toward the Earth's center. X_{TD} is perpendicular to Z_{TD} axis and is positive northward along the meridian plane containing the Orbiter. Y_{TD} completes the right-handed orthogonal system.
CHARACTERISTICS:	Rotating, right-handed Cartesian system. Velocity vectors are expressible in this system for the Orbiter, given relative velocity V_{TD} in this system.

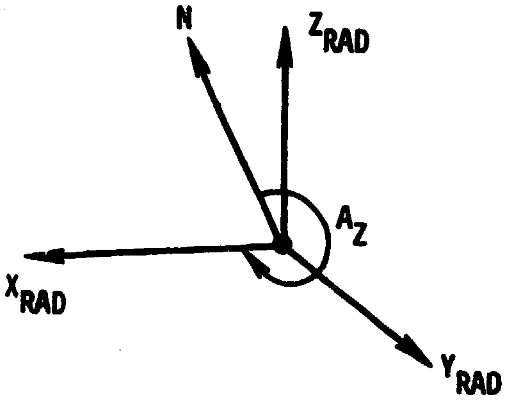
$$\gamma_{TD} = \sin^{-1}\left(\frac{-\dot{Z}_{TD}}{V_{TD}}\right)$$

$$\psi_{TD} = \tan^{-1}\left(\frac{\dot{Y}_{TD}}{\dot{X}_{TD}}\right)$$

ϕ_D = geodetic latitude

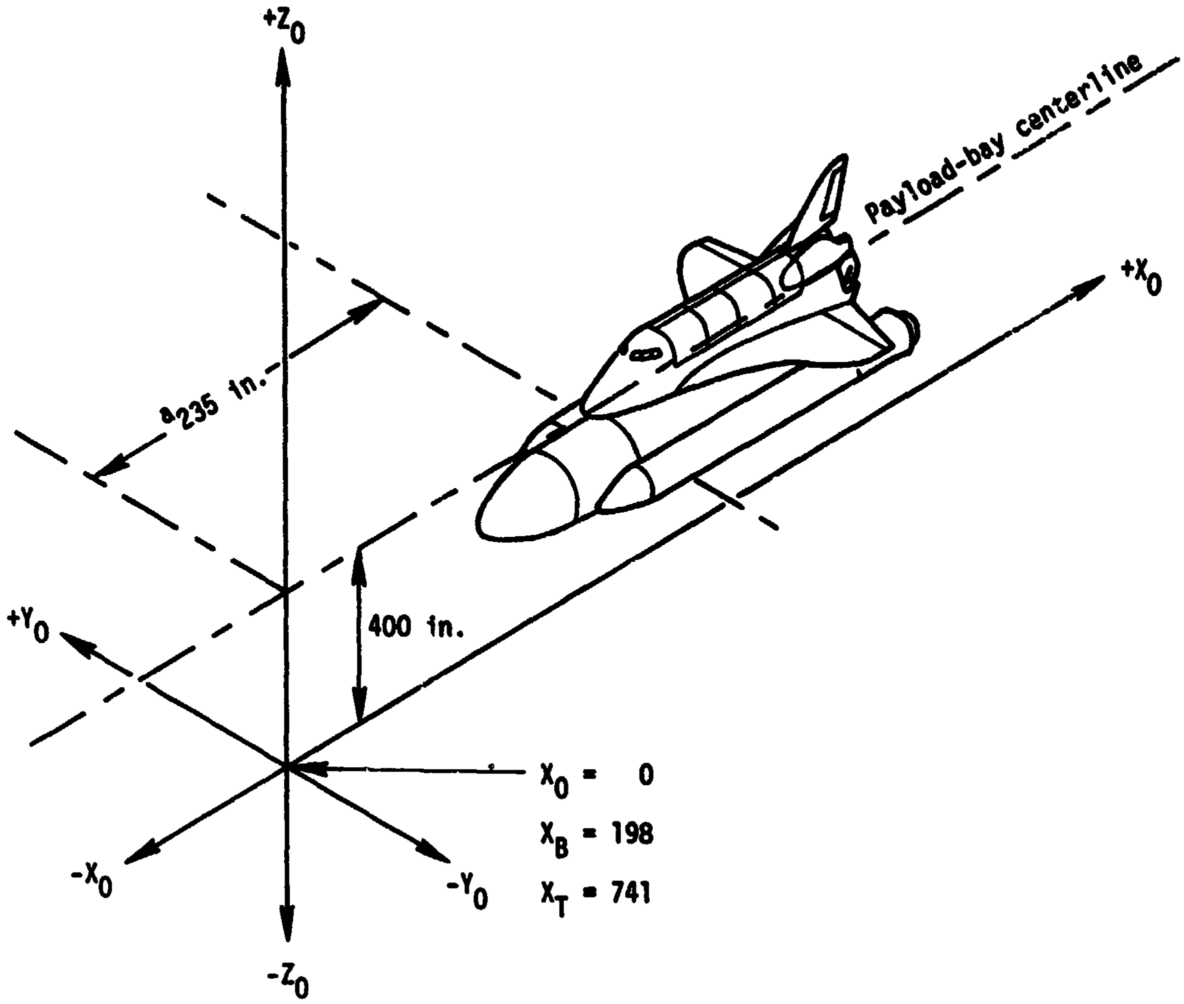
*A similar system may be defined for any point of interest.

Figure C-6.- Topodetic Coordinate System.



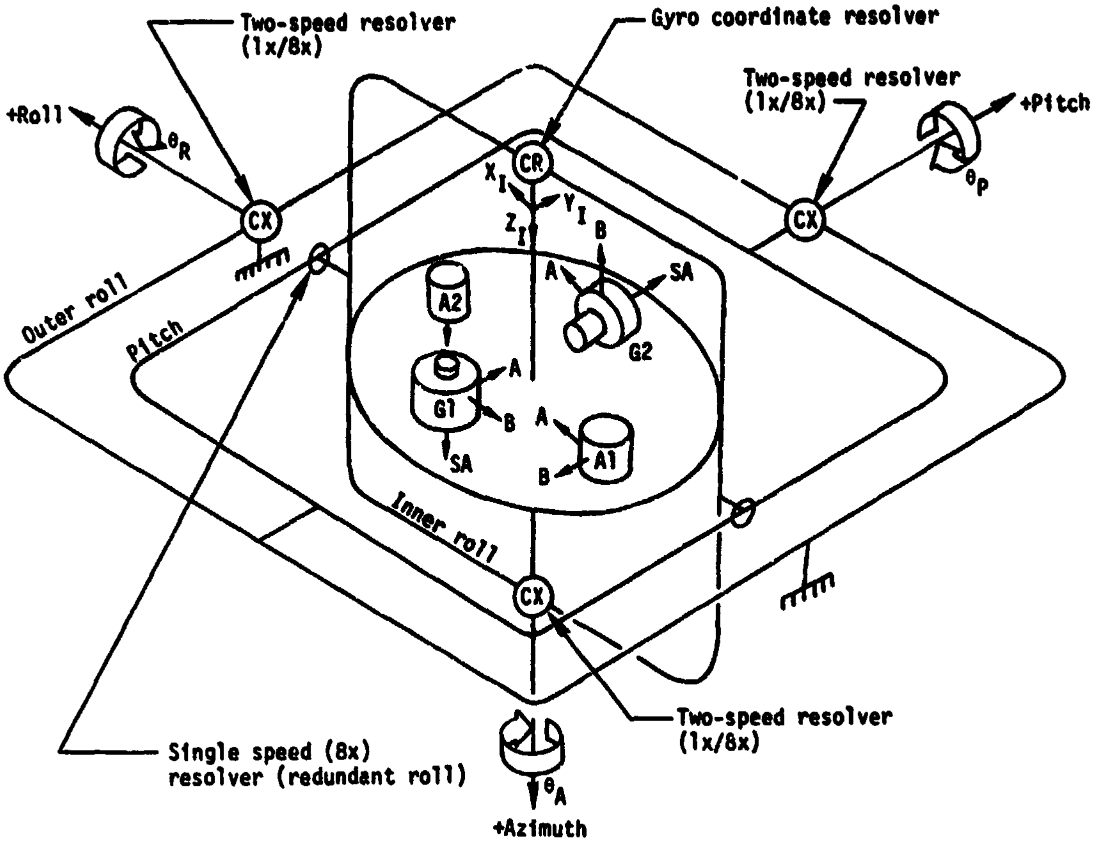
NAME:	MSBLS radar coordinate system
ORIGIN:	MSBLS radar position
ORIENTATION AND DEFINITIONS:	Z_{RAD} is normal to the ellipsoidal model, positive away from the center of the Earth. X_{RAD} is in the local horizontal plane of the radar in the boresight direction of the radar antenna. The angle A_Z is measured positively clockwise from true north to X_{RAD}.
CHARACTERISTICS:	Rotating, Earth-referenced

Figure C-7.- MSBLS Radar Coordinate System.



NAME:	Body coordinate system (structural)
ORIGIN:	In the Orbiter plane of symmetry, 400 inches below the centerline of the payload bay and at Orbiter X-station = 0
ORIENTATION:	The X_0 axis is in the vehicle plane of symmetry, parallel to and 400 inches below the payload bay centerline. Positive sense is from the nose of the vehicle toward the tail. The Z_0 axis is in the vehicle plane of symmetry, perpendicular to the X_0 axis and positive upward in landing attitude. The Y_0 axis completes a right-handed system.
CHARACTERISTICS:	Rotating, right-handed Cartesian

Figure C-8.- Body Coordinate System (Structural).



NAME:	Stable member (IMU)
ORIGIN:	The intersection of the innermost gimbal axis and the measurement plane of the XY two-axis accelerometer
ORIENTATION:	The Z_I axis is coincident with the innermost gimbal axis. The X_I axis is determined by the projection of the X accelerometer input axis (IA) onto a plane orthogonal to Z_I. Y_I completes a right-handed triad. In a perfect IMU, with all misalinements zero, these relationships hold: The X accelerometer and X gyro IA's are parallel to the X_I axis. The Y accelerometer and Y gyro IA's are parallel to the Y_I axis. The Z accelerometer and Z gyro IA's are parallel to the Z_I axis.

Figure C-9.- Stable Member (IMU) (Sheet 1 of 2).

CHARACTERISTICS: Nonrotating, right-handed, Cartesian system

The reference alinement for the gimbal case is defined with the four gimbal angles at zero and with the vehicle in a horizontal position. In a perfect IMU, with all misalinements zero and with all gimbal angles at zero, the following relationships hold.

X_{NB}, Y_{NB}, Z_{NB} are Cartesian components of the navigation base coordinate system.

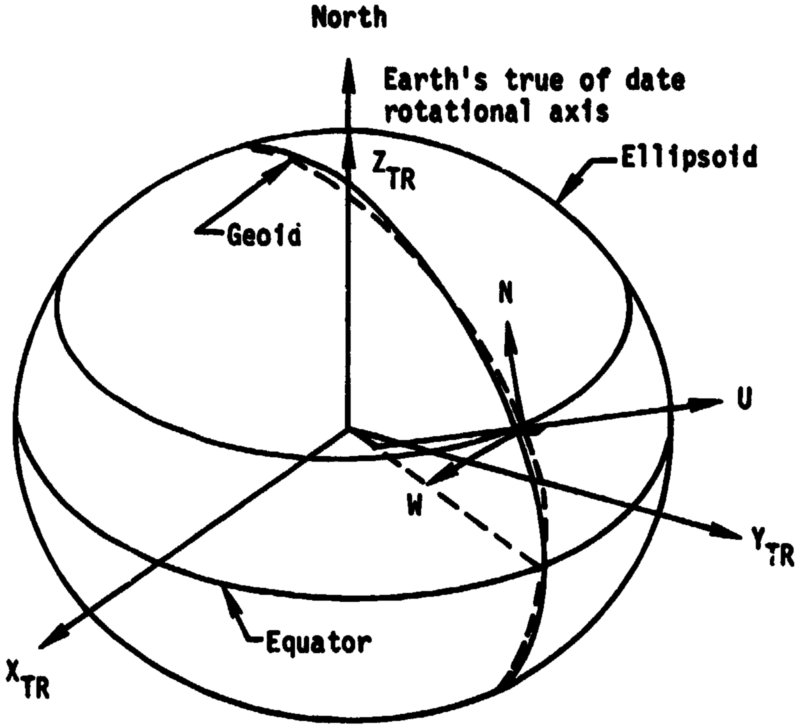
The outer roll axis and the X_I-axis will be parallel to X_{NB}. Positive X_I will be in the forward direction. Positive roll gimbal angles will be in the sense of a right-handed rotation of the gimbal case relative to the platform about the plus outer roll axis.

The pitch axis and Y_I will be parallel to Y_{NB}. Positive Y_I will be to the right of an observer looking forward in the vehicle. Positive pitch gimbal angles will be in the sense of a right-handed rotation of the gimbal case relative to the platform about the plus pitch axis.

The inner roll axis will be parallel to the outer roll axis, with the sense of rotation the same as for the outer roll axis.

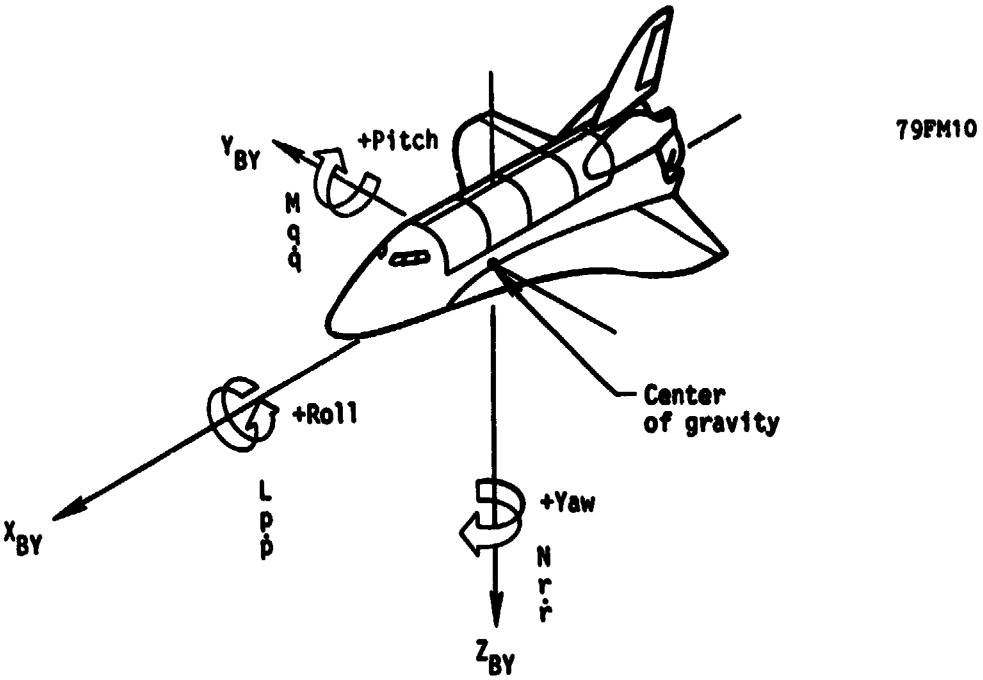
The azimuth axis and Z_I will be parallel to Z_{NB}. Positive Z_I will be down, relative to an observer standing in the vehicle. Positive azimuth gimbal angles will be in the sense of a right-handed rotation of the gimbal case relative to the platform about the plus azimuth axis.

Figure C-9.- Stable Member (IMU) (Sheet 2 of 2).



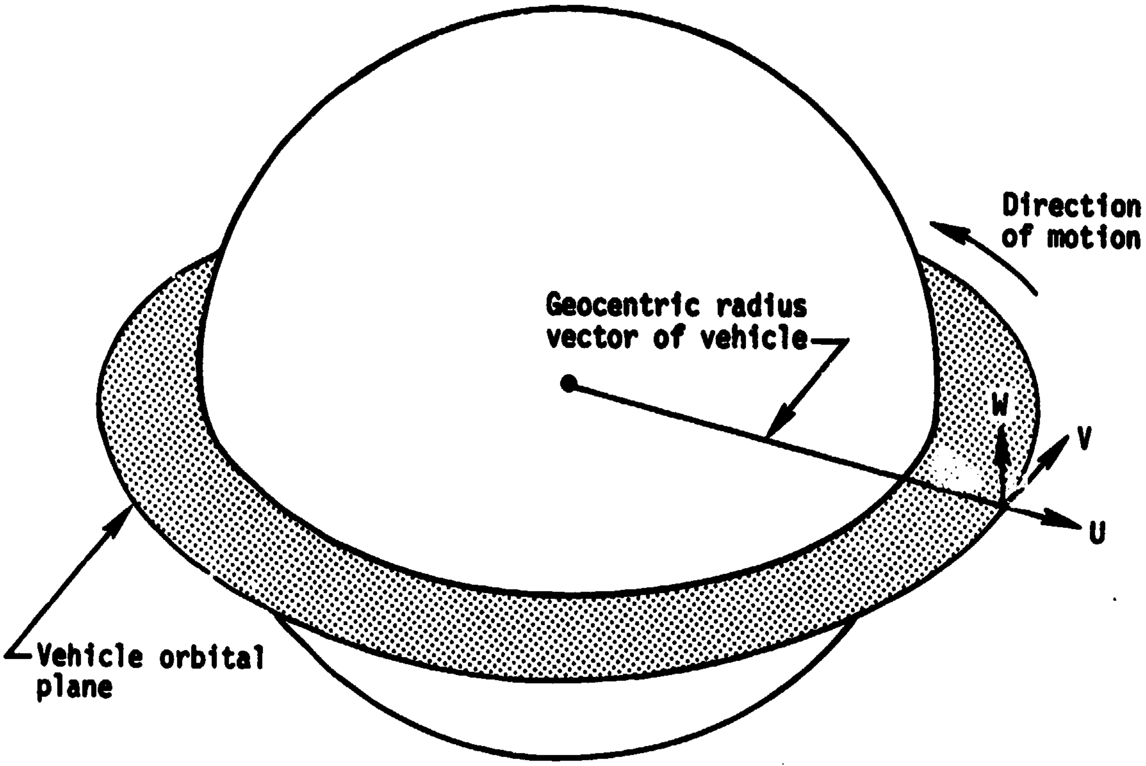
NAME:	NWU geographic coordinate system
ORIGIN:	Point of interest
ORIENTATION:	U is positive up along the Earth's astronomic or plumb-bob vertical. W is positive along the cross product of U and the Earth's spin axis. N completes the right-handed orthogonal system ($N = W \times U$).
CHARACTERISTICS:	Rotating, right-handed, Earth-referenced coordinate system

Figure C-10.- NWU Geographic Coordinate System.



NAME:	Body axis coordinate system
ORIGIN:	Center of mass
ORIENTATION:	X_{BY} axis is parallel to the Orbiter structural body X_O axis, positive toward the nose. Z_{BY} axis is parallel to the Orbiter plane of symmetry and is perpendicular to X_{BY}, positive down with respect to the Orbiter fuselage. Y_{BY} axis completes the right-handed orthogonal system.
CHARACTERISTICS:	Rotating, right-handed, Cartesian system L, M, N: Moments about X_{BY}, Y_{BY}, and Z_{BY} axes, respectively p, q, r: Body rates about X_{BY}, Y_{BY}, and Z_{BY} axes, respectively $\dot{p}$, $\dot{q}$, $\dot{r}$: Angular body acceleration about X_{BY}, Y_{BY}, and Z_{BY} axes, respectively The Euler sequence that is commonly associated with this system is a yaw, pitch, roll sequence, where ψ = yaw, θ = pitch, and ϕ = roll or bank. This attitude sequence is yaw, pitch, and roll around the Z_{BY}, Y_{BY}, and X_{BY} axes, respectively.

Figure C-11.- Body Axis.



NAME:	UVW coordinate system
ORIGIN:	Point of interest
ORIENTATION:	The U-V plane is the instantaneous orbit plane at epoch. The U axis lies along the geocentric radius vector to the vehicle and is positive radially outward. The W axis lies along the instantaneous orbital angular momentum vector at epoch and is positive in the direction of the angular momentum vector. The V axis completes a right-handed system (V = W x U).
CHARACTERISTICS:	Quasi-inertial, right-handed Cartesian coordinate system. This system is quasi-inertial in the sense that it is treated as an inertial coordinate system, but it is redefined at each point of interest.

Figure C-12.- UVW Coordinate System

FLOWCHARTS

$$\boxed{\text{EARTH\_FIXED\_TO\_M50\_COORD(TIME)} = \text{M\_M50TOEF\_AT\_EPOCH}^{T}\text{(M)}}$$

WHERE:

M_M50TOEF_AT_EPOCH IS THE TRANSFORMATION FROM MEAN OF 1950 TO EARTH-FIXED COORDINATES AT TIME = T_EPOCH. THIS MATRIX, T_EPOCH, AND EARTH_RATE INITIAL VALUES ARE DISCUSSED IN SECTION 4.10.2.

M IS COMPUTED WITHIN THE FUNCTION (EARTH_FIXED_TO_M50_COORD) AS

$$\text{LAM} = \text{EARTH\_RATE(TIME} - \text{T\_EPOCH)}$$

$$M = \begin{pmatrix} \text{COS(LAM)} & -\text{SIN(LAM)} & 0 \\ \text{SIN(LAM)} & \text{COS(LAM)} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Figure C-13.- EARTH_FIXED_TO_M50_COORD FUNCTION.

```
EF_TO_TOPDET(LAT_GEOD, LON) = M
```

WHERE

M IS CALCULATED WITHIN THE FUNCTION AS

CLON = COS(LON), SLON = SIN(LON)

CLAT = COS(LAT_GEOD), SLAT = SIN(LAT_GEOD)

$$M = \begin{pmatrix} -SLAT\ CLON & -SLAT\ SLON & CLAT \\ -SLON & CLON & 0 \\ -CLAT\ CLON & -CLAT\ SLON & -SLAT \end{pmatrix}$$

Figure C-14.- EF_TO_TOPDET FUNCTION.

EF_TO_RUNWAY(LAT_GEOD, LON, AZ) = (M) EF_TO_TOPDET(LAT_GEOD, LON)

WHERE

M IS CALCULATED WITHIN THE FUNCTION AS

CAZ = COS(AZ), SAZ = SIN(AZ)

$$M = \begin{pmatrix} CAZ & SAZ & 0 \\ -SAZ & CAZ & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Figure C-15.- EF_TO_RUNWAY FUNCTION.

```
EF_TO_SCANNER(LAT_GEOD, LON, AZ) = (M) EF_TO_RUNWAY(LAT_GEOD, LON, AZ)
```

WHERE

M IS CALCULATED WITHIN THE FUNCTION AS

$$M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

Figure C-16.- EF_TO_SCANNER FUNCTION.

IN LIST: R_EF
OUT LIST: LAT_GEOD, LON, ALT

```
$\text{FLATCON} = 1.0 - (1.0-\text{ELLIPT})^2$
$\text{A\_0} = 1.0 - \text{FLATCON}$
$\text{R\_XY} = \text{R\_EF}_1^{\,2} + \text{R\_EF}_2^{\,2}$
$\text{R} = (\text{R\_XY} + \text{R\_EF}_3^{\,2})^{1/2}$
$\text{SIN\_P} = \text{R\_EF}_3/\text{R}$
$\text{COS\_P} = (\text{R\_XY})^{1/2}/\text{R}$
$\text{RAD\_P} = \dfrac{\text{EARTH\_RADIUS\_EQUATOR}}{(1.0 + \text{FLATCON SIN\_P}^2/\text{A\_0})^{1/2}}$
$\text{DEL} = (\text{FLATCON SIN\_P COS\_P})/(1.0-\text{FLATCON COS\_P}^2)$
$\text{DEL\_LAT} = \text{RAD\_P DEL}/\text{R}$
$\text{PHI} = \text{TAN}^{-1}\,(\text{SIN\_P}/\text{COS\_P})$
$\text{LAT\_GEOD} = \text{PHI} + \text{DEL\_LAT}$
$\text{LON} = \text{ARCTAN2}(\text{R\_EF}_2,\ \text{R\_EF}_1)$
$\text{ALT} = (\text{R-RAD\_P})\ (1.0-0.50\ \text{DEL DEL\_LAT})$
```

Figure C-17.- EF_TO_GEODETIC.

GEODETIC_TO_EF(LAT_GEOD, LON, ALT) = $\underline{\text{R}}$ _EF

WHERE

$\underline{\text{R}}$ _EF IS CALCULATED WITHIN THE FUNCTION AS

CLAT = COS(LAT_GEOD), SLAT = SIN(LAT_GEOD)

DUM = $(1 - \text{ELLIPT})^2$

DUM1 = EARTH_RADIUS_EQUATOR/ $\sqrt{\text{CLAT}^2 + \text{SLAT}^2 \ \text{DUM}}$

R_EF_EQUAT = (DUM1 + ALT)CLAT

$\underline{\text{R}}$ _EF = (R_EF_EQUAT COS(LON), R_EF_EQUAT SIN(LON),
(DUM DUM1 + ALT)SLAT)

Figure C-18.- GEODETIC_TO_EF FUNCTION.

IN LIST: $\underline{R}$, $\underline{V}$

$$\underline{U} = \underline{R}/|\underline{R}|$$

$$\underline{W} = (\underline{R} \times \underline{V})/|\underline{R} \times \underline{V}|$$

$$\underline{Z} = (\underline{W} \times \underline{U})$$

$$\mathrm{UVW\_TO\_M50}_{1 \text{ TO } 3,1} = \underline{U}$$

$$\mathrm{UVW\_TO\_M50}_{1 \text{ TO } 3,2} = \underline{Z}$$

$$\mathrm{UVW\_TO\_M50}_{1 \text{ TO } 3,3} = \underline{W}$$

Figure C-19.- UVW_TO_M50 FUNCTION.

IN LIST: Q_FIFTY_BODY
OUT LIST: MAT

MATRIX = QUAT_TO_MAT (Q_FIFTY_BODY)

$MAT_{1,1} = -MATRIX_{1,1}$

$MAT_{1,2} = MATRIX_{2,1}$

$MAT_{1,3} = -MATRIX_{3,1}$

$MAT_{2,1} = -MATRIX_{1,2}$

$MAT_{2,2} = MATRIX_{2,2}$

$MAT_{2,3} = -MATRIX_{3,2}$

$MAT_{3,1} = -MATRIX_{1,3}$

$MAT_{3,2} = MATRIX_{2,3}$

$MAT_{3,3} = -MATRIX_{3,3}$

Figure C-20.- SBODY_TO_M50.

```
QUAT_TO_MAT(Q) = A
```

WHERE A IS CALCULATED WITHIN THE FUNCTION AS

```
$P2 = Q_2 + Q_2$
$P3 = Q_3 + Q_3$
$P4 = Q_4 + Q_4$
$P5 = P2\ Q_2$
$P6 = P4\ Q_4$
$TEMP = 1.0 - P3\ Q_3$
$A_{1,1} = TEMP-P6$
$A_{2,2} = 1.0-P5-P6$
$A_{3,3} = TEMP-P5$
$P5 = P2\ Q_3$
$P6 = P4\ Q_1$
$A_1 = P5-P6$
$A_{2,1} = P5 + P6$
$P5 = P2\ Q_4$
$P6 = P3\ Q_1$
$A_{1,3} = P5 + P6$
$A_{3,1} = P5-P6$
$P5 = P3\ Q_4$
$P6 = P2\ Q_1$
$A_{2,3} = P5-P6$
$A_{3,2} = P5 + P6$
```

Figure C-21.- QUAT_TO_MAT FUNCTION.

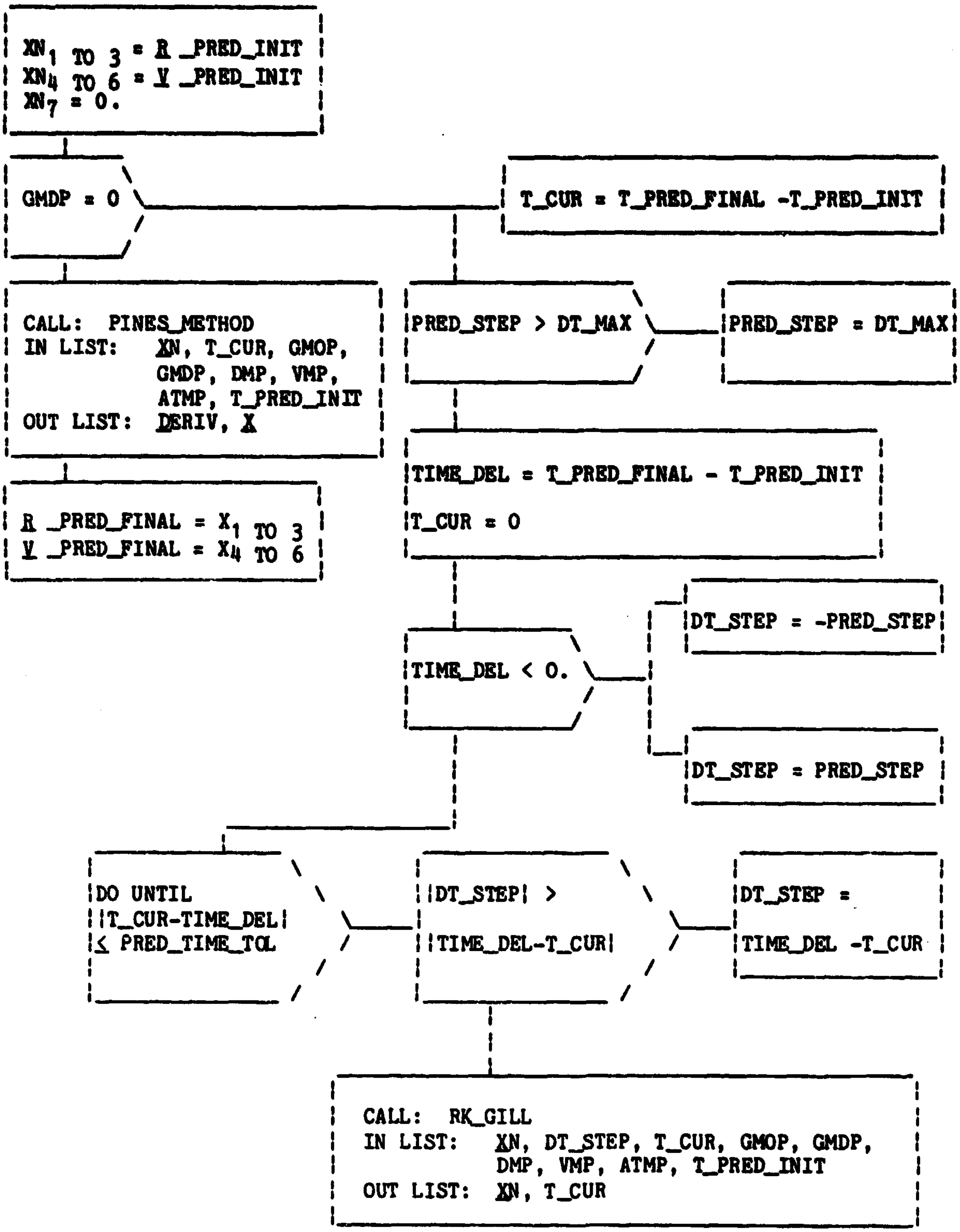


Figure C-22.- ONORBIT_PREDICT.

IN LIST: XN, DT_STEP, T_CUR, GMO, GMD, DM, VM, ATM, T_IN
OUT LIST: XN, T_CUR

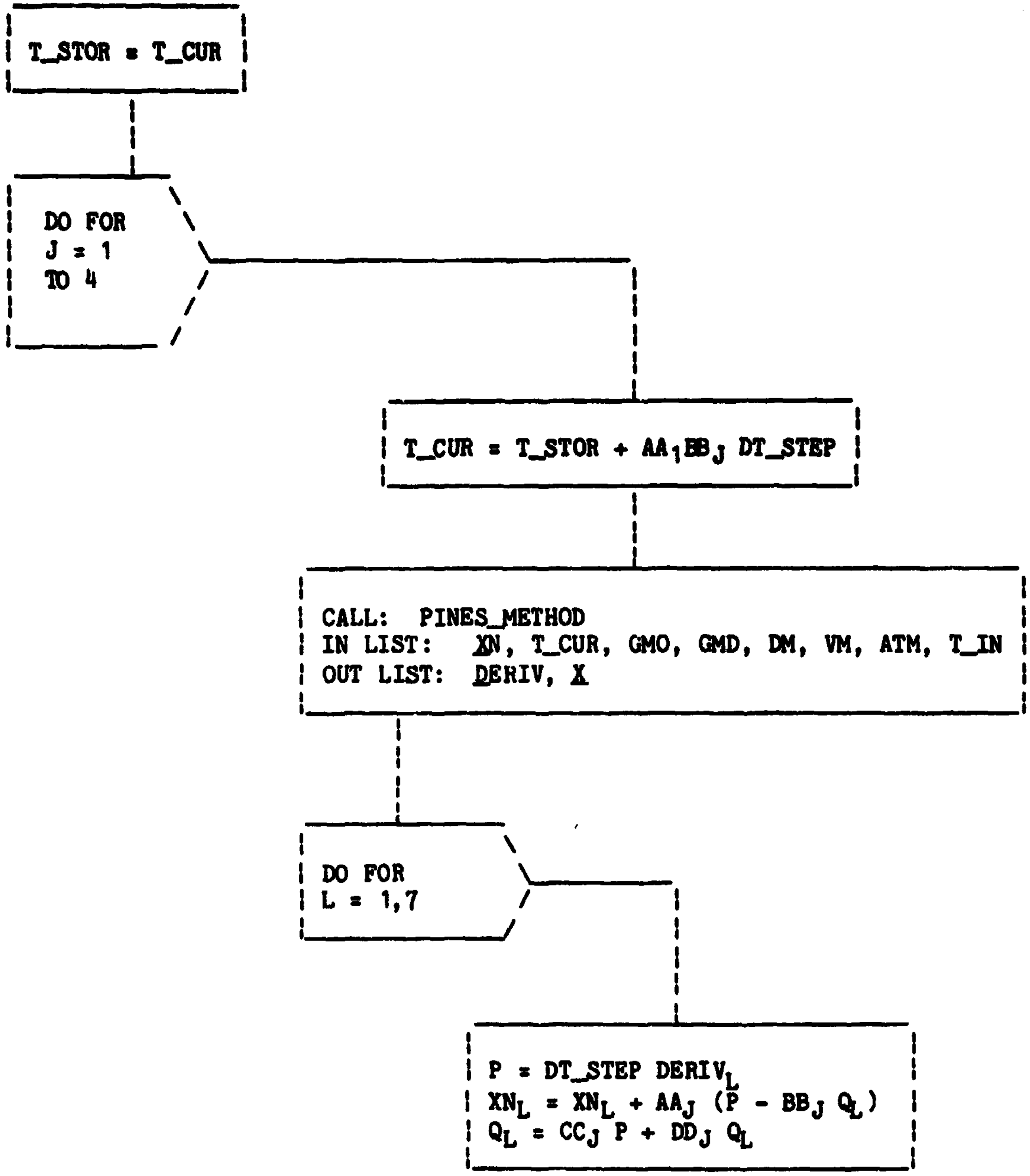


Figure C-23.- RK_GILL.

IN LIST: $\underline{XN}$, T_CUR, GMO, GMD, DM, VM, ATM, T_IN
OUT LIST: $\underline{DERIV}$, $\underline{X}$

```
R_IN = |$XN_{1\ TO\ 3}$|
R_IN_INV = 1./R_IN
SMA = 1./[2.R_IN_INV - ($XN_{4\ TO\ 6} \cdot XN_{4\ TO\ 6}$)/EARTH_MU]
C1 = SQRT(SMA)/SQR_EMU
DELTAT = T_CUR-$XN_7$
D_IN = $XN_{1\ TO\ 3} \cdot XN_{4\ TO\ 6}$
R_FIN_TEMP_INV = 0.

CALL:  F_AND_G
IN LIST:   SMA, DELTAT, C1, $XN_{1\ TO\ 3}$, R_IN_INV,
           R_FIN_TEMP_INV, $XN_{4\ TO\ 6}$, D_IN, D_FIN_TEMP
OUT LIST:  F, G, FDOT, GDOT, S0, S1, S2, S3, $X_{1\ TO\ 3}$, R_FIN_INV, THETA

$X_{4\ TO\ 6}$ = FDOT $XN_{1\ TO\ 3}$ + GDOT $XN_{4\ TO\ 6}$

T_ACCEL = T_IN + T_CUR
$\underline{P}$ = ACCEL_ONORBIT (GMD, GMO, DM, VM, ATM, $X_{1\ TO\ 3}$, $X_{4\ TO\ 6}$, T_ACCEL)
$\underline{P}$ = $\underline{P}$ - $\underline{G}$_CENTRAL
D_TAU = $X_{1\ TO\ 3} \cdot \underline{P}$
D_AUX = $X_{4\ TO\ 6} \cdot \underline{P}$

(CONT'D)
```

Figure C-24.- PINES_METHOD (Sheet 1 of 2).

(CONT'D)

```
$C2 = C1^{2}$
S1 = C1 S1
S2 = C2 S2
C3 = 1./C2
S3 = SMA S2
S4 = 2. S3 D_AUX
C4 = C2 D_AUX
C5 = C4 S1
S5 = S2 D_TAU

DD = S1 C3 R_IN(SMA R_IN_INV-1.) + S0 D_IN
S6 = 2. S2 C4 DD + S5
R_IN_TAU = S4 - C2 S1 D_AUX DD - S1 D_TAU
R_IN_AUX = R_IN_INV R_IN_TAU
F_TAU = (S3 C3 R_IN_AUX - S4) R_IN_INV
G_TAU = C5/R_FIN_INV - S6
FD_TAU = FDOT (C4-R_IN_AUX)
GD_TAU = -S4 R_FIN_INV

$DERIV_{1\ TO\ 3}$ = GD_TAU $X_{1\ TO\ 3}$ - G_TAU $X_{4\ TO\ 6}$ - G $\underline{P}$
$DERIV_{4\ TO\ 6}$ = -FD_TAU $X_{1\ TO\ 3}$ + F_TAU $X_{4\ TO\ 6}$ + F $\underline{P}$
$DERIV_{7}$ = S6 - 3. C4 SMA (C1 THETA - S1) - C5/R_FIN_INV
```

Figure C-24.- PINES_METHOD (Sheet 2 of 2).

APPENDIX D

USER PARAMETER FLOWCHARTS, VARIABLE NAMES, AND DESCRIPTIONS

CONTENTS

VARIABLES LIST DEFINITIONS

CODE USED FOR VARIABLE DATA TYPE

F: floating point quantity; an n-dimensional floating point vector will be denoted F(n); similarly, an n x m floating point matrix will be denoted by F(n,m)

I: integer quantity; I(n) will denote an n-dimensional integer vector

B: bit, i.e., data having only values of 0 or 1

C: character. C(n) will denote an n-dimensional character string

CODE USED FOR VARIABLE PRECISION

D: double precision

S: single precision; integer quantities are assumed single precision unless otherwise stated

VARIABLE LOCATION

COMPOOL: Variable value located in common storage, accessible by all functions

LOCAL: Variable is used by one function only, and usable to other functions through call argument only

VARIABLE INITIALIZATION CATEGORY

blank: display is vacant

C: constant (unchanging)

DD: design dependent

HC: hard coded

MD: mission dependent (I-LOAD)

VARIABLE INITIAL VALUE

initial operation sequence computer inputs

VARIABLE UPLINK AND DOWNLIST STATUS

UPLINK: variable is an uplink item

DOWNLIST: variable is a downlist item

UNITS DEFINITIONS

deg: angular measurement, degrees

ft: feet

lb: pounds

n.d.: non-dimensional

rad: radian

sec: time measurement, seconds

slugs: mass measurement, slugs

vary: units have different values which depend on variable use

VARIABLES LIST

Variable name (M/3 ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
AC	SP (3)	local	—	—	--	ft/sec^2	Sensed acceleration (local variable used in AVERAGE_G_ INTEGRATOR).
A _SENSED (V95A0902C -AC)	SP (3)	compool	—	—	downlist	ft/sec^2	Ratio of difference of selected accelerometer readings to difference of their time tags.
ATFL_OV	I	compool	HC	0	—	--	Attitude flag for Orbiter and for UPP.
DA_THRESHOLD_IMU	SP	local	—	—	—	ft/sec^2	Threshold value for IMU acceleration in UPP.
DEL_R_TARG (V95H0855C -7C)	DP (3)	compool	—	—	downlist	ft	M50 vector from Shuttle to target.
DEL_V_TARG (V95L0858C -60C)	DP (3)	compool	—	—	downlist	ft/sec	M50 delta velocity vector between Shuttle and target.
DFL_AVG	B	local	HC	0	—	--	Drag flag used in average-G integration.
DOING_REND_NAV (V90X4765X)	B	compool	—	—	downlist	--	Flag indicating whether rendezvous navigation is active (ON) or onorbit navigation is active (OFF).
DTIME	SP	local	—	—	--	sec	Step size for state vector advancement (local variable used in AVERAGE_G_INTEGRATOR).
DT_IMU	SP	local	—	—	--	sec	State vector average-G integration time step.
EARTH_RATE	DP	compool	C	--	—	rad/sec	Rotation rate of Earth in radians per second.
EVENT_60H (V90X8644X)	B	compool	—	OFF	downlist	--	Transition from OPS 8 to OPS 3.
EVENT_E1 (V90X8164X)	B	compool	—	OFF	downlist	--	Transition to MM201 from MM301 event flag.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial-ization category	Initial value	Uplink/ downlist	Units	Description
EVENT_60 (V̄90X8189X)	B	compool	—	OFF	downlist	--	Transition to MM201 from MM106 event flag.
EVENT_60A (V̄90X3160X)	B	compool	—	OFF	downlist	--	Transition to MM201 from OPS 8 event flag.
EVENT_60B (V̄90X864[illegible]X)	B	compool	—	OFF	downlist	--	Transition to OPS 8 from MM201.
EVENT_67 (V̄90X8646X)	B	compool	—	OFF	downlist	--	Transition from MM201 to MM202.
EVENT_69 (V̄90X8191X)	B	compool	—	OFF	downlist	--	Guidance initiate event flag.
EVENT_73 (V̄90X8200X)	B	compool	—	OFF	downlist	--	Transition to MM201 from MM202 event flag.
EVENT_83 (T̄BD)	B	compool	—	OFF	downlist	--	Transition from MM201 to OPS 00.
EVENT_84 (V̄90X8148X)	B	compool	—	OFF	downlist	--	Transition to MM201 from OPS 00 event.
FILT_UPDATE (V̄90X0224X)	B	compool	—	OFF	downlist	--	Flag indicating the availability of a filter updated state.
GM_DEG_LOW (V̄96U9118C)	I	compool	DD	2	—	--	Prestored value of potential model degree indicator.
GM_ORD_LOW (V̄96U9120C)	I	compool	DD	0	—	--	Prestored value of potential model order indicator.
G̲R	DP (3)	local	—	—	—	ft/sec^2	Gravitational acceleration (local variable used in AVERAGE_G_INTEGRATOR).

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial- ization category	Initial value	Uplink/ downlist	Units	Description
GR1	DF (3)	local	--	--	--	ft/sec^2	Gravitational acceleration (local variable used in AVERAGE_G_INTEGRATOR).
IMU_NAV_ACCEL_THRESH (V93A6710C)	SF	compool	MD	592	downlist	μg	IMU Navigation acceleration threshold value in μg's.
M50_TO_EF	DF (3,3)	local	--	--	--	--	Mean of 1950 to Earth-fixed Greenwich coordinate transformation matrix.
OPS_2_OR_8_INITIALIZE_COMPLETE (V90X1242X)	B	compool	--	--	--	--	Signal to MSC indicating that initialization of user parameter state propagation is complete.
R _AV	DF (3)	local	--	--	--	ft	Position vector (local variable used in AVERAGE_G_INTEGRATOR).
R _AVGG (V95H0185C -7C)	DF (3)	compool	--	--	downlist	ft	Current Orbiter position vector updated by user parameter propagator.
R _EF (V95H0155C -7C)	DF (3)	compool	--	--	--	ft	Orbiter position vector in Earth-fixed coordinates.
R _RESET (V90H0235C -7C)	DF (3)	compool	--	--	--	ft	Copy of filter updated Orbiter position vector for user parameter propagator reset.
R _TARGET (V95H0862C -4C)	DF (3)	compool	--	--	downlist	ft	Position vector of the target vehicle updated by the user parameter propagator.
R _TV_RESET (V90H1383C -5C)	DF (3)	compool	--	--	--	ft	Copy of filter updated target position vector for user parameter propagator reset.
T_IMU (V95W0901C)	DF	compool	--	--	downlist	sec	Current time tag snapped from the IMU SOP (IMU SOP name T_IMUS_GA M/SID V95W0002C).

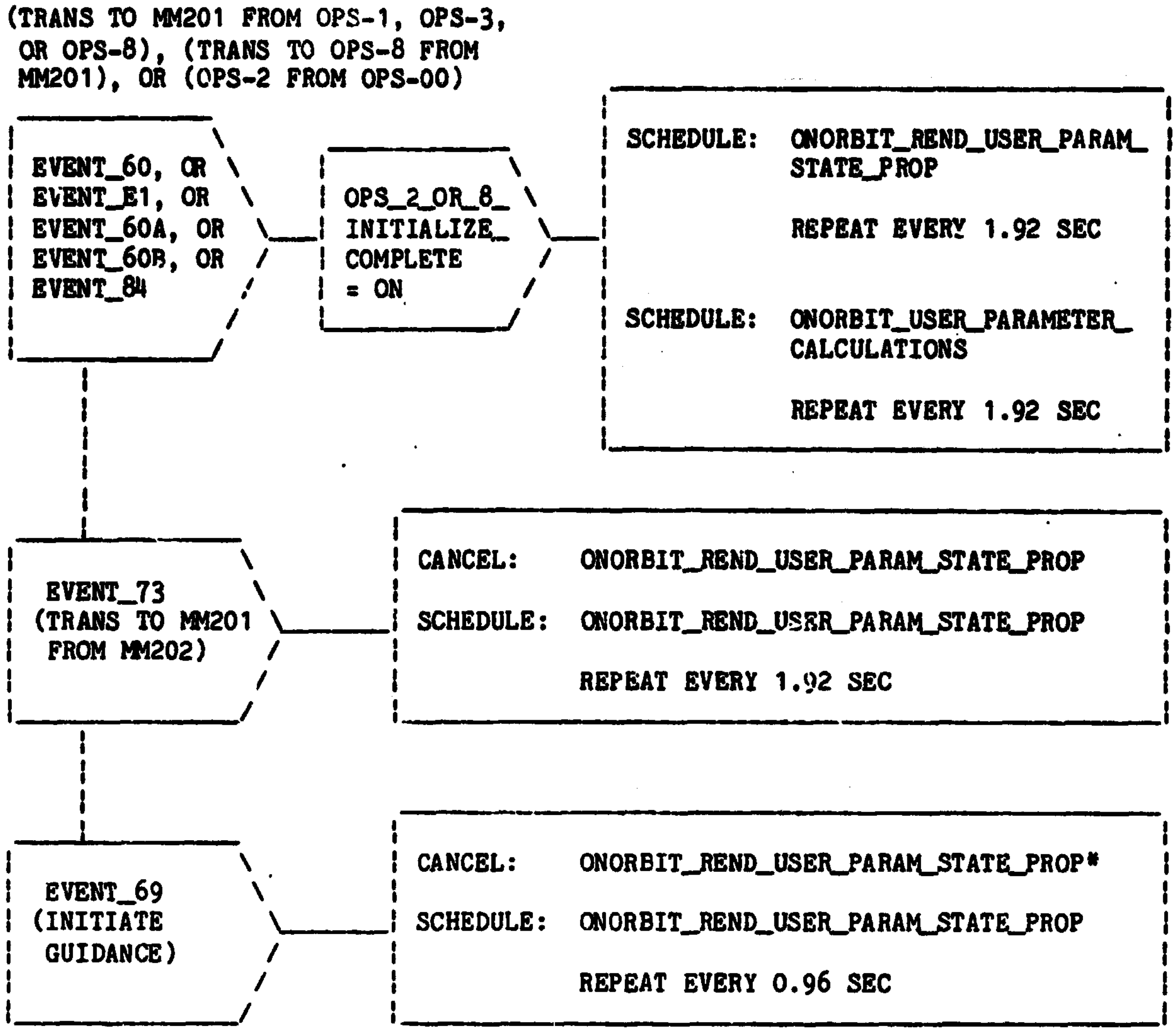
VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial-ization category	Initial value	Uplink/ downlist	Units	Description
T_RESET (V90W1386C)	DF	compool	--	—	--	sec	Copy of time tag of filter update of state vectors for user parameter propagator reset.
T_SEC_GMT (V95W3890C)	DF	compool	—	—	—	sec	Time in seconds since Greenwich midnight of the reference day.
T_STATE (V95W0200C)	DF	compool	—	—	downlist	sec	Time tag for current user parameter state vector.
UPP_USE_IMU_DATA (V95X0889X)	B	compool	IV	OFF	downlist	--	Flag indicating IMU data are to be used in integration (ON) in UPP.
V _AV	DF (3)	local	—	—	—	ft/sec	Velocity vector (local variable used in AVERAGE_G_ INTEGRATOR).
V _AVGG (V95L0190C -2C)	DF (3)	compool	—	—	downlist	ft/sec	Velocity vector of Orbiter updated by the user parameter propagator.
V _EF	SF (3)	local	—	—	—	ft/sec	Orbiter velocity vector in Earth-fixed coordinates.
VFLTV_PRED	B	compool	HC	0	—	--	Venting flag for target vehicle prediction and for UPP state prop.
V _IMU_SNAP (V95L0898C -900C)	DF (3)	compool	—	—	downlist	ft/sec	Current snapped accumulated IMU velocity obtained from the IMU SOP (IMU SOP name V _IMU_CURRENT M/SID V90L2557 -9C).
V _IMU_OLD (V95L0210C -2C)	DF (3)	compool	—	—	—	ft/sec	Previous accumulated IMU velocity.
V _IMU_RESET (V90L0247C -9C)	DF (3)	compool	—	—	downlist	ft/sec	Copy of IMU accumulated sensed velocity for user parameter propagator reset.
V _RESET (V90L0239C -41C)	DF (3)	compool	—	—	—	ft/sec	Copy of filter updated Orbiter velocity vector for user parameter propagator reset.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
$\underline{V}$ _RHO_EF ($\bar{V}$95L3884C -6C)	DP (3)	compool	—	—	—	ft/sec	Earth-fixed velocity vector for Shuttle.
$\underline{V}$ _TARGET ($\bar{V}$95L0867C -9C)	DP (3)	compool	—	—	downlist	ft/sec	Velocity vector of target vehicle updated by the user parameter propagator.
$\underline{V}$ _TV_RESET ($\bar{V}$90L1380C -2C)	DP (3)	compool	—	—	—	ft/sec	Copy of filter updated target velocity vector for user parameter propagator reset.

Flowcharts



*THE PURPOSE OF THIS CANCELLATION AND RESCHEDULING IS TO SYNCHRONIZE THIS MODULE WITH THE EXECUTION OF ONORBIT GUIDANCE, WHICH IS TO BEGIN COMPUTATIONS AT THIS TIME.

Figure D-1.- ONORBIT_REND_UPP_SEQ.

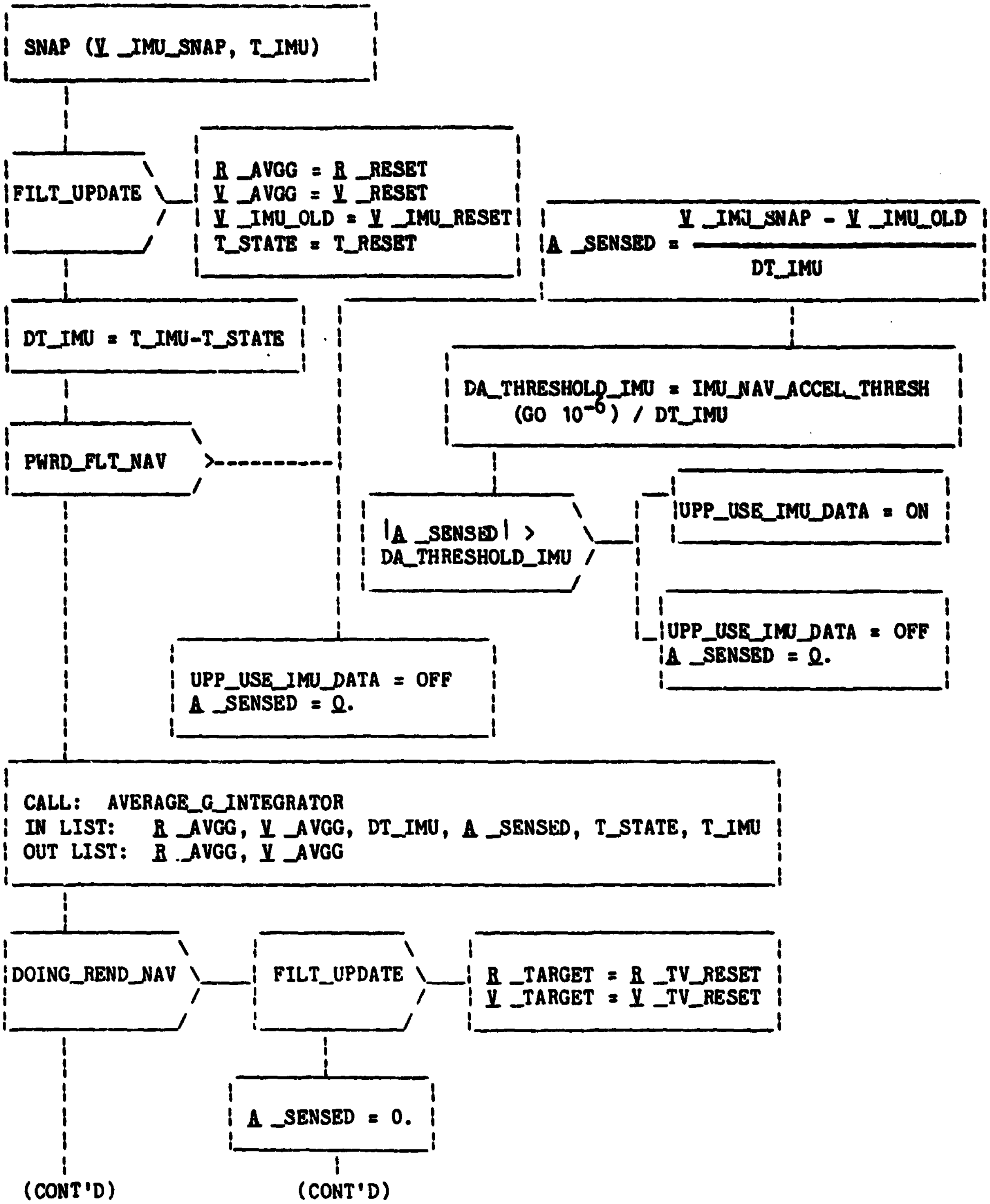


Figure D-2.- ONORBIT_REND_USER_PARAM_STATE_PROP (Sheet 1 of 2).

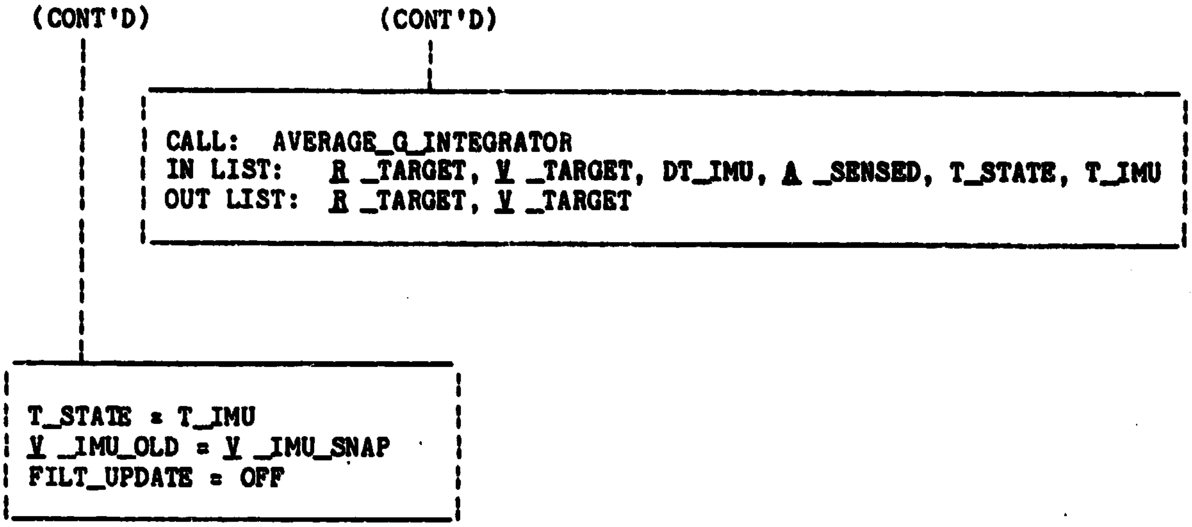


Figure D-2.- ONORBIT_REND_USER_PARAM_STATE_PROP (Sheet 2 of 2).

IN LIST: R _AV, V _AV, DTIME, AC, T_STATE, T_IMU
OUT LIST: R _AV, V _AV

```
GR = ACCEL_ONORBIT (GM_DEG_LOW, GM_ORD_LOW, DFL_AVG, VFLTV_PRED, ATFL_OV,
     R _AV, V _AV, T_STATE)
R _AV = R _AV + DTIME [V _AV + 0.5 DTIME (AC + GR)]
GR1 = ACCEL_ONORBIT (GM_DEG_LOW, GM_ORD_LOW, DFL_AVG, VFLTV_PRED, ATFL_OV,
      R _AV, V _AV, T_IMU)
V _AV = V _AV + DTIME [AC + 0.5 (GR + GR1)]
```

Figure D-3.- AVERAGE_G_INTEGRATOR.

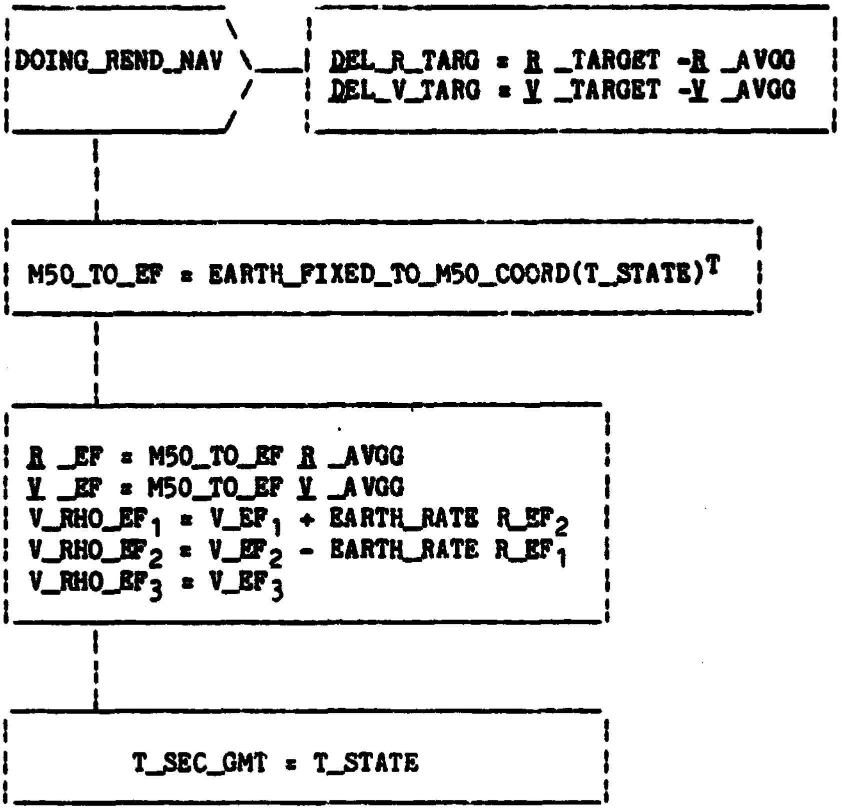


Figure D-4.- ONORBIT_USER_PARAMETER_CALCULATIONS.

APPENDIX E

LANDING SITE UPDATE PRINCIPAL FUNCTION FLOWCHARTS AND NAME DESCRIPTIONS

CONTENTS

PRECEDING PAGE BLANK NOT FILMED

VARIABLES LIST DEFINITIONS

CODE USED FOR VARIABLE DATA TYPE

F: Floating point quantity. An n-dimensional floating point vector will be denoted F(n). Similarly, an n x m floating point matrix will be denoted by F(n,m).

I: Integer quantity; I(n) will denote an n-dimensional integer vector

B. Bit, i.e., data having only values of 0 or 1

C: Character; C(n) will denote an n-dimensional character string

CODE USED FOR VARIABLE PRECISION

D: Double precision

S: Single precision; integer quantities are assumed single precision unless otherwise stated

VARIABLE LOCATION

COMPOOL: Variable value located in common storage, accessible by all functions.

LOCAL: Variable is used by one function only, and usable to other functions through call argument only

VARIABLE INITIALIZATION CATEGORY

blank: Display is vacant

C: Constant (unchanging)

DD: Design dependent

HC: Hard coded

MD: Mission dependent (I-LOAD)

VARIABLE INITIAL VALUE

Initial operation sequence computer inputs

PRECEDING PAGE BLANK NOT FILMED

VARIABLE UPLINK AND DOWNLIST STATUS

UPLINK: Variable is an uplink item

DOWNLIST: Variable is a downlist item

UNITS DEFINITIONS

deg: angular measurement, degrees

ft: feet

lb: pounds

n.d.: non-dimensional

rad: radian

sec: time measurement, seconds

Slugs: mass measurement, slugs

Vary: Units have different values which depend on variable use

VARIABLES LIST

PRECEDING PAGE BLANK NOT FILMED

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial-ization category	Initial value	Uplink/ downlist	Units	Description
ALT_ABOVE_ELLIPSOID(1)_O	SF(10)	compool	MD	-	-	ft	Height of TACAN site above reference ellipsoid in maxi and mini tables.
ALT_ABOVE_ELLIPSOID_MAXI	SF(50)	compool	MD	-	-	ft	
ALT_MLSEL	SF(4)	compool	MD	-	-	ft	Geodetic altitude of MSBLS elevation antenna location for primary, secondary, and maxi table landing sites.
ALT_MLSEL(1)_O	SF	compool	MD	-	-	ft	
ALT_MLSEL(2)_O	SF	compool	MD	-	-	ft	
ALT_MLS_R_AZ	SF(4)	compool	MD	-	-	ft	Geodetic altitude of the MSBLS range/azimuth antenna location for primary,secondary, and maxi table landing sites.
ALT_MLS_R_AZ(1)_O	SF	compool	MD	-	-	ft	
ALT_MLS_R_AZ(2)_O	SF	compool	MD	-	-	ft	
ALT_RWY_MAXI_SELECT	SI	compool	MD	-	-	ft	Runway selection index from maxi table for alternate runway.
BIAS_AZMLS	SF(4)	compool	MD	-	-	rad	MSBLS azimuth data bias for the primary, secondary, and maxi table landing sites.
BIAS_AZMLS(1)_O	SF	compool	MD	-	-	rad	
BIAS_AZMLS(2)_O	SF	compool	MD	-	-	rad	
BIAS_ELMLS	SF(4)	compool	MD	-	-	rad	MSBLS elevation data bias for the primary, secondary, and maxi table runway sites.
BIAS_ELMLS(1)_O	SF	compool	MD	-	-	rad	

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial- ization category	Initial value	Uplink/ downlist	Units	Description
BIAS_ELMLS(2)_0	SF	compool	MD	-	-	rad	
BIAS_MLSRANGE	SF(4)	compool	MD	-	-	ft	MSBLS range data bias for the primary and secondary maxi table landing sites.
BIAS_MLSRANGE(1)_0	SF	compool	MD	-	-	ft	
BIAS_MLSRANGE(2)_0	SF	compool	MD	-	-	ft	
BIAS_TAC_BRG	SF(10)	compool	MD	-	-	rad	Bias in TACAN bearing in maxi and mini tables.
BIAS_TAC_R	SF(10)	compool	MD	-	-	ft	Bias in TACAN range in maxi and mini tables.
EL_SCANNER_BEARING	SF(4)	compool	MD	-	-	rad	Bearing from true north of MLS elevation scanner boresight axis from maxi tables.
EL_SCANNER_BEARING(1)_0	SF	compool	MD	-	-	rad	MSBLS elevation antenna boresight for primary landing site.
EL_SCANNER_BEARING(2)_0	SF	compool	MD	-	-	rad	MSBLS elevation antenna boresight for secondary landing site.
ITEM1_IN	B	compool		OFF	-	-	Maxi to primary runway transfer discrete.
ITEM2_IN	B	compool		OFF	-	-	Maxi table to secondary runway data transfer discrete.
ITEM3_IN	B	compool		OFF	-	-	Maxi table to alternate runway data transfer discrete.
LATITUDE_GEODETIC(1)_0	SF(10)	compool	MD	-	-	rad	Geodetic latitude at TACAN site from maxi and mini tables.
LATITUDE_GEODETIC_MAXI	SF(50)	compool	MD	-	-	rad	
LAT_MLSEL	SF(4)	compool	MD	-	-	rad	Geodetic latitude of MSBLS elevation antenna from maxi table.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
LAT_MLSEL(1)_0	SP	compool	MD	-	-	rad	Geodetic latitude of the MSBLS elevation antenna location for the primary and secondary landing sites.
LAT_MLSEL(2)_0	SP	compool	MD	-	-	rad	
LAT_MLS_R_AZ	SP(4)	compool	MD	-	-	rad	Latitude of MSBLS range/azimuth scanner site from maxi table.
LAT_MLS_R_AZ(1)_0	SP	compool	MD	-	-	rad	Geodetic latitude of range/azimuth MSBLS antenna location for the primary and secondary landing sites.
LAT_MLS_R_AZ(2)_0	SP	compool	MD	-	-	rad	
LONGITUDE_EAST(1)_0	SP(10)	compool	MD	-	-	rad	Geodetic longitude at TACAN site in maxi and mini tables.
LONGITUDE_EAST_MAXI	SP(50)	compool	MD	-	-	rad	
LONG_MLSEL	SP(4)	compool	MD	-	-	rad	Geoetic longitude of the MSBLS elevation antenna from maxi table.
LONG_MLSEL(1)_0	SP	compool	MD	-	-	rad	Longitude of the MSBLS elevation antenna location for the primary and secondary landing sites.
LONG_MLSEL(2)_0	SP	compool	MD	-	-	rad	
LONG_MLS_R_AZ	SP(4)	compool	MD	-	-	rad	Longitude of MSBLS range/azimuth scanner site from maxi table.
LONG_MLS_R_AZ(1)_0	SP	compool	MD	-	-	rad	Longitude of range/azimuth MSBLS antenna location for primary and secondary landing sites.
LONG_MLS_R_AZ(2)_0	SP	compool	MD	-	-	rad	
MAGNETIC_VARIATION(1)_0	SP(10)	compool	MD	-	-	rad	Magnetic variation of TACAN site in maxi and mini tables.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
MAGNETIC_VARIATION_MAXI	SF(50)	compool	MD	-	-	rad	
MLS_AVAIL(1)_0	B	compool	MD	-	-	-	MSBLS available discrete for primary and secondary runways.
MLS_AVAIL(2)_0	B	compool	MD	-	-	-	
MSL_ABOVE_ELLIPSOID(1)_0	SF(10)	compool	MD	-	-	ft	MSL correction to reference ellipsoid at the TACAN site in maxi and mini tables.
MSL_ABOVE_ELLIPSOID_MAXI	SF(50)	compool	MD	-	-	ft	
R _AZ_RADAR_BEARING	SF(4)	compool	MD	-	-	rad	
R_AZ_RADAR_BEARING(1)_0	SF	compool	MD	-	-	rad	MSBLS azimuth boresight relative to true north for the primary, secondary, and maxi table runways.
R_AZ_RADAR_BEARING(2)_0	SF	compool	MD	-	-	rad	
RUNWAY_ALT	SF(18)	compool	MD	-	-	ft	Altitude above reference ellipsoid of runway reference point for maxi table runways.
RUNWAY_NAME	C(18)	compool	MD	-	-	-	Name of landing site from maxi table.
RUNWAY_NAME(3)_0	C	compool	MD	-	-	-	Name of landing site from mini table for alternate runway.
RUNWAY_NAME(1)_0	C	compool	MD	-	-	-	
RUNWAY_NAME(2)_0	C	compool	MD	-	-	-	Name of landing site from mini table for primary and secondary runways.
RUNWAY_ALT(3)_0	SF	compool	MD	-	-	ft	

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial-ization category	Initial value	Uplink/ downlist	Units	Description
RUNWAY_ALT(1)_0	SP	compool	MD	-	-	ft	Altitude above reference ellipsoid of runway reference point for primary, secondary, and alternate runways.
RUNWAY_ALT(2)_0	SP	compool	MD	-	-	ft	
RW_AZIMUTH	SP(18)	compool	MD	-	-	rad	Bearing from true north of runway coordinate system +X axis for maxi table.
RW_AZIMUTH(3)_0	SP	compool	MD	-	-	rad	Bearing from true north of runway coordinate system +X for primary, secondary, and alternate runways.
RW_AZIMUTH(1)_0	SP	compool	MD	-	-	rad	
RW_AZIMUTH(2)_0	SP	compool	MD	-	-	rad	
RW_DELH	SP(18)	compool	MD	-	-	ft	MSL correction to ellipsoid at runway site for maxi table runways.
RW_DELH(3)_0	SP	compool	MD	-	-	ft	MSL correction to ellipsoid at runway site for primary, secondary, and alternate runways.
RW_DELH(1)_0	SP	compool	MD	-	-	ft	
RW_DELH(2)_0	SP	compool	MD	-	-	ft	
RW_LAT$_I$	SP(18)	compool	MD	-	-	rad	Geodetic latitude of Ith runway in maxi table.
RW_LAT(3)_0	SP	compool	MD	-	-	rad	Geodetic latitude of primary, secondary, and alternate runways.
RW_LAT(1)_0	SP	compool	MD	-	-	rad	
RW_LAT(2)_0	SP	compool	MD	-	-	rad	
RW_LON$_I$	SP(18)	compool	MD	-	-	rad	Geodetic longitude of Ith runway in maxi table.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial-ization category	Initial value	Uplink/ downlist	Units	Description
RW_LON(3)_0	SF	compool	MD	-	-	rad	Geodetic longitude of primary, secondary, and alternate runways.
RW_LON(1)_0	SF	compool	MD	-	-	rad	
RW_LON(2)_0	SF	compool	MD	-	-	rad	
RW_MAG_VAR	SF(18)	compool	MD	-	-	rad	Angle of magnetic variation from true north at maxi table runways.
RW_MAG_VAR(3)_0	SF	compool	MD	-	-	rad	Angle of magnetic variation from true north at primary, secondary, and alternate mini table runways.
RW_MAG_VAR(1)_0	SF	compool	MD	-	-	rad	
RW_MAG_VAR(2)_0	SF	compool	MD	-	-	rad	
TACAN_ID	SI(10)	compool	MD	-	-	-	TACAN channel number from maxi and mini tables. Positive for x-mode
TACAN_ID_MAXI	SI(50)	compool	MD	-	-	-	Negative for y-mode
X_DMEAZ_RW	SF(4)	compool	MD	-	-	ft	X-component of range/azimuth antenna in runway coordinates from maxi tables.
X_DMEAZ_RW(1)_0	SF	compool	MD	-	-	ft	
X_DMEAZ_RW(2)_0	SF	compool	MD	-	-	ft	X-component of MSBLS range/azimuth antenna in runway coordinates for primary and secondary landing sites.
X_EL_RW	SF(4)	compool	MD	-	-	ft	X-component of elevation antenna-runway coordinates from maxi tables.
X_EL_RW(1)_0	SF	compool	MD	-	-	ft	

VARIABLES LIST

Variable name (H/S ID)	Precision & type	Compool or local	Initial-ization category	Initial value	Uplink/ downlist	Units	Description
X_EL_RW(2)_O	SP	compool	MD	-	-	ft	X-component of MSBLS elevation antenna in runway coordinates for primary and secondary landing sites.
Y_DMEAZ_RW_I	SP(4)	compool	MD	-	-	ft	Y-component of range/azimuth antenna in runway coordinates of the Ith runway in maxi tables.
Y_DMEAZ_RW(1)_O	SP	compool	MD	-	-	ft	
Y_DMEAZ_RW(2)_O	SP	compool	MD	-	-	ft	Y-component of MSBLS range/azimuth antenna in runway coordinates for primary and secondary landing sites.
PRI_MAXI_SELECT	SI	compool		19	-	-	Runway selection index maxi table to mini table primary, secondary, and alternate runway data transfer.
SEC_MAXI_SELECT	SI	compool		20	-	-	
ALT_MAXI_SELECT	SI	compool		21	-	-	
PRI_MAXI_CURRENT	SI	compool		19	-	-	Indicators of which maxi table runway has been transferred to the mini table primary, secondary and alternate slots.
SEC_MAXI_CURRENT	SI	compool		20	-	-	
ALT_MAXI_CURRENT	SI	compool		21	-	-	

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial- ization category	Initial value	Uplink/ downlist	Units	Description
ALT_ABOVE_ELLIPSOID	SP(10)	compool	OPS	-	-	ft	Height of TACAN site above reference ellipsoid in mini table.
ALT_MLSEL_PSL	SP	compool	OPS	-	-	ft	Geodetic altitude of MSBLS elevation antenna location for primary and secondary mini table landing sites.
ALT_MLSEL_SSL	SP	compool	OPS	-	-	ft	
ALT_MLS_R_AZ_PSL	SP	compool	OPS	-	-	ft	Geodetic altitude of the MSBLS range/azimuth antenna location for primary and secondary mini table landing sites.
ALT_MLS_R_AZ_SSL	SP	compool	OPS	-	-	ft	
BIAS_AZMLS_PSL	SP	compool	OPS	-	-	rad	MSBLS azimuth data bias for the primary and secondary mini table landing sites.
BIAS_AZMLS_SSL	SP	compool	OPS	-	-	rad	
BIAS_ELMLS_PSL	SP	compool	OPS	-	-	rad	MSBLS elevation data bias for the primary and secondary mini table runway sites.
BIAS_ELMLS_SSL	SP	compool	OPS	-	-	rad	
BIAS_MLSRANGE_PSL	SP	compool	OPS	-	-	ft	MSBLS range data bias for the primary and secondary mini table landing sites.
BIAS_MLSRANGE_SSL	SP	compool	OPS	-	-	ft	
EL_SCANNER_BEARING_ PSL	SP	compool	OPS	-	-	rad	MSBLS elevation antenna boresight for primary landing site.
EL_SCANNER_BEARING_ SSL	SP	compool	OPS	-	-	rad	MSBLS elevation antenna boresight for secondary landing site.
LATITUDE_GEODETIC(1)	SP(10)	compool	OPS	-	-	rad	Geodetic latitude at TACAN site from mini table.
LAT_MLSEL_PSL	SP	compool	OPS	-	-	rad	Geodetic latitude of the MSBLS elevation antenna location for the primary and secondary landing sites.
LAT_MLSEL_SSL	SP	compool	OPS	-	-	rad	

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
LAT_MLS_R_AZ_PSL	SP	compool	OPS	-	-	rad	Geodetic latitude of range/azimuth MSBLS antenna location for the primary and secondary landing sites.
LAT_MLS_R_AZ_SSL	SP	compool	OPS	-	-	rad	
LONGITUDE_EAST(1)	SP(10)	compool	OPS	-	-	rad	Geodetic longitude at TACAN site in mini table.
LONG_MLSEL_PSL	SP	compool	OPS	-	-	rad	Longitude of the MSBLS elevation antenna location for the primary and secondary landing sites.
LONG_MLSEL_SSL	SP	compool	OPS	-	-	rad	
LONG_MLS_R_AZ_PSL	SP	compool	OPS	-	-	rad	Longitude of range/azimuth MSBLS antenna location for primary and secondary landing sites.
LONG_MLS_R_AZ_SSL	SP	compool	OPS	-	-	rad	
MAGNETIC_VARIATION	SP(10)	compool	OPS	-	-	rad	Magnetic variation of TACAN site in mini table.
MLS_AVAIL_PSL	B	compool	OPS	-	-	-	MSBLS available discrete for primary and secondary runways.
MLS_AVAIL_SSL	B	compool	OPS	-	-	-	
MSL_ABOVE_ELLIPSOID	SP(10)	compool	OPS	-	-	ft	MSL correction to reference ellipsoid at the TACAN site in mini table.
R_AZ_RADAR_BEARING_PSL	SP	compool	OPS	-	-	rad	MSBLS azimuth boresight relative to true north for the primary and secondary mini table runways.
R_AZ_RADAR_BEARING_SSL	SP	compool	OPS	-	-	rad	
RUNWAY_NAME_ASL	C	compool	OPS	-	-	-	Name of landing site from mini table for alternate runway.
RUNWAY_NAME_PSL	C	compool	OPS	-	-	-	

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial-ization category	Initial value	Uplink/ downlist	Units	Description
RUNWAY_NAME_SSL	C	compool	OPS	-	-	-	Name of landing site from mini table for primary and secondary runways.
RUNWAY_ALT_ASL	SP	compool	OPS	-	-	ft	Altitude above reference ellipsoid of runway reference point for primary, secondary, and alternate runways.
RUNWAY_ALT_PSL	SP	compool	OPS	-	-	ft	
RUNWAY_ALT_SSL	SP	compool	OPS	-	-	ft	
RW_AZIMUTH_ASL	SP	compool	OPS	-	-	rad	Bearing from true north of runway coordinate system +X for primary, secondary, and alternate runways.
RW_AZIMUTH_PSL	SP	compool	OPS	-	-	rad	
RW_AZIMUTH_SSL	SP	compool	OPS	-	-	rad	
RW_DELH_ASL	SP	compool	OPS	-	-	ft	MSL correction to ellipsoid at runway site for primary, secondary, and alternate runways.
RW_DELH_PSL	SP	compool	OPS	-	-	ft	
RW_DELH_SSL	SP	compool	OPS	-	-	ft	
RW_LAT_ASL	SP	compool	OPS	-	-	rad	Geodetic latitude of primary, secondary, and alternate runways.
RW_LAT_PSL	SP	compool	OPS	-	-	rad	
RW_LAT_SSL	SP	compool	OPS	-	-	rad	
RW_LON_ASL	SP	compool	OPS	-	-	rad	Geodetic longitude of primary, secondary, and alternate runways.
RW_LON_PSL	SP	compool	OPS	-	-	rad	
RW_LON_SSL	SP	compool	OPS	-	-	rad	

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial-ization category	Initial value	Uplink/ downlist	Units	Description
RW_MAG_VAR_ASL	SF	compool	OPS	-	-	rad	Angle of magnetic variation from true north at primary, secondary, and alternate mini table runways.
RW_MAG_VAR_PSL	SF	compool	OPS	-	-	rad	
RW_MAG_VAR_SSL	SF	compool	OPS	-	-	rad	
TACAN_ID(1)	SI(10)	compool	OPS	-	-	-	TACAN channel number from mini table Positive for x-mode Negative for y-mode
X_DMEAZ_RW_PSL	SF	compool	OPS	-	-	ft	
X_DMEAZ_RW_SSL	SF	compool	OPS	-	-	ft	X-component of MSBLS range/azimuth antenna in runway coordinates for primary and secondary landing sites.
X_EL_RW_PSL	SF	compool	OPS	-	-	ft	
X_EL_RW_SSL	SF	compool	OPS	-	-	ft	X-component of MSBLS elevation antenna in runway coordinates for primary and secondary landing sites.
Y_DMEAZ_RW_PSL	SF	compool	OPS	-	-	ft	
Y_DMEAZ_RW_SSL	SF	compool	OPS	-	-	ft	Y-component of MSBLS range/azimuth antenna in runway coordinates for primary and secondary landing sites.

Flowcharts

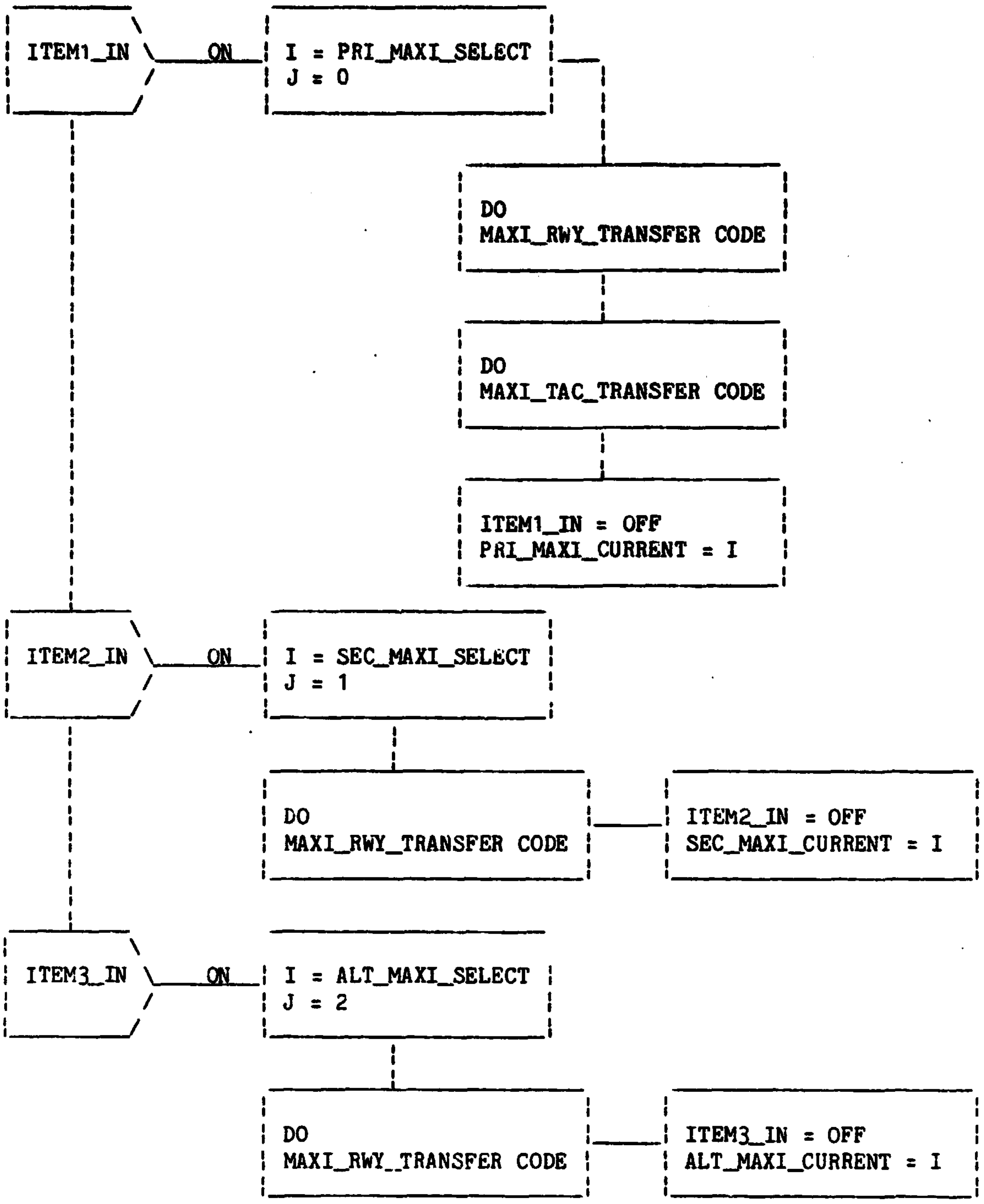


Figure E-1.- Landing Site Update Principal Function.

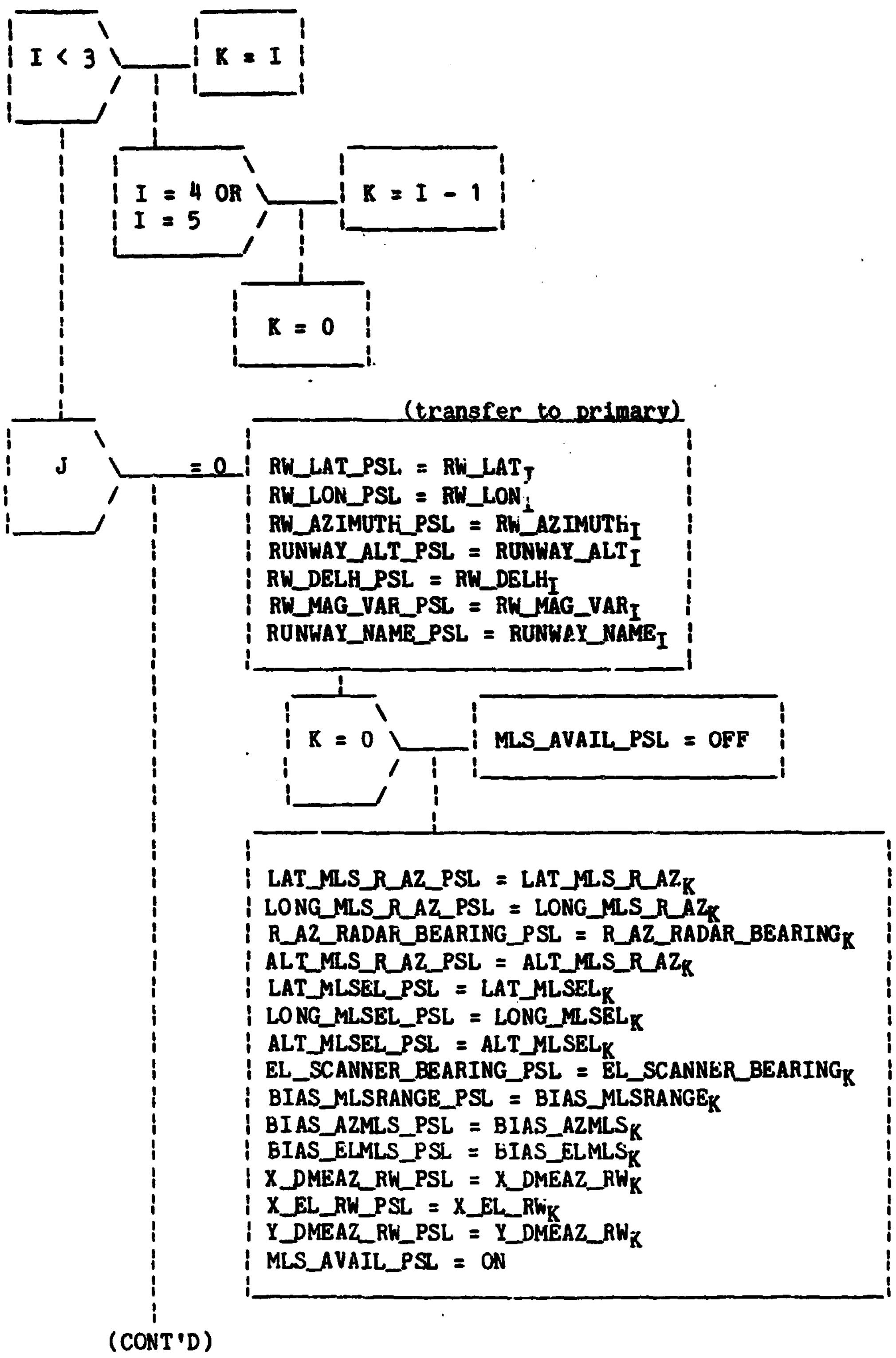
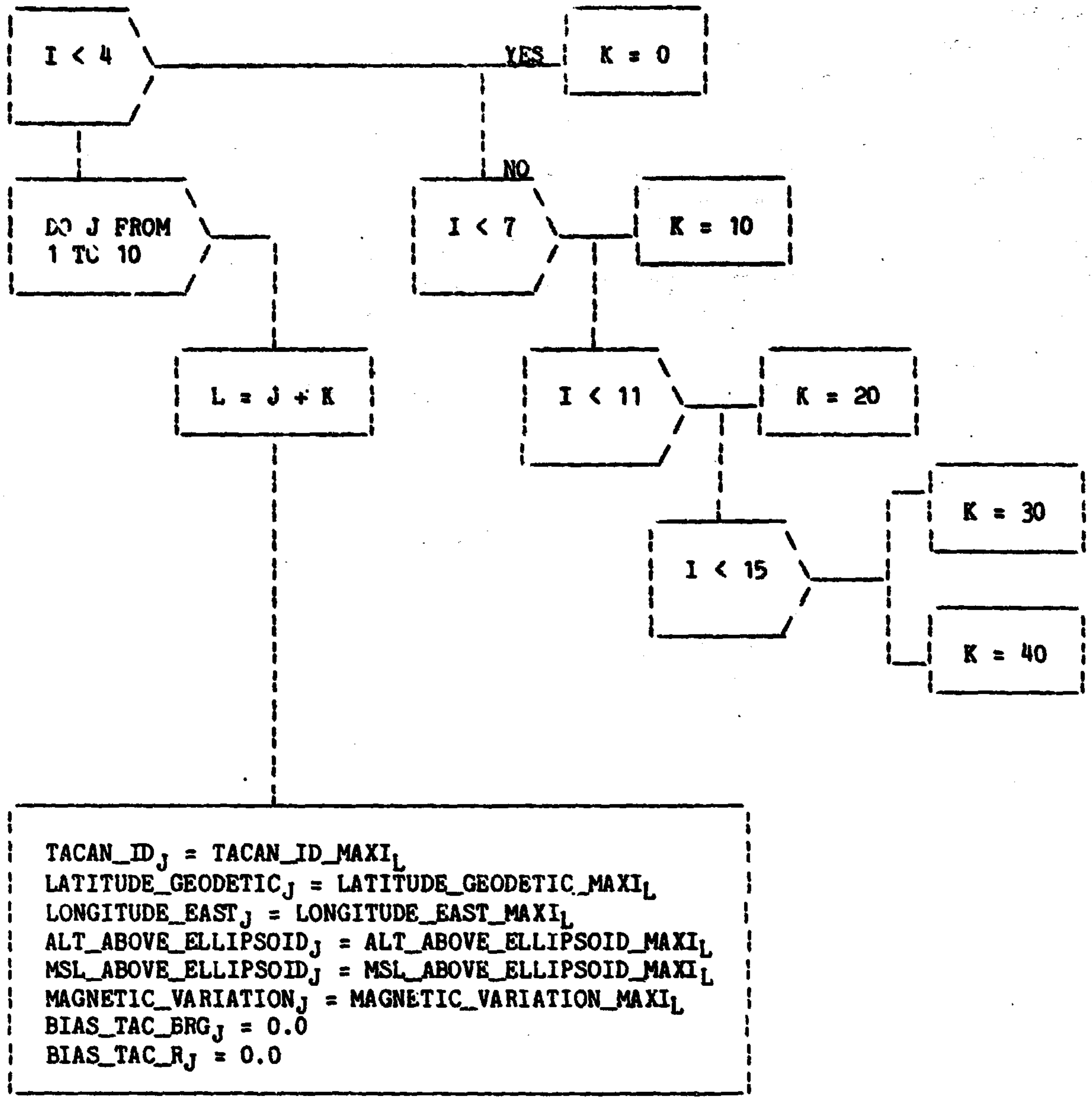


Figure E-2.- MAXI_RWY_TRANSFER CODE (Sheet 1 of 2).

```
(CONT'D)
    |
    |           ______(transfer to secondary)
    |          |                                        |
    |___= 1____| RW_LAT_SSL = RW_LAT_I                  |
    |          | RW_LON_SSL = RW_LON_I                  |
    |          | RW_AZIMUTH_SSL = RW_AZIMUTH_I          |
    |          | RUNWAY_ALT_SSL = RUNWAY_ALT_I          |
    |          | RW_DELH_SSL = RW_DELH_I                |
    |          | RW_MAG_VAR_SSL = RW_MAG_VAR_I          |
    |          | RUNWAY_NAME_SSL = RUNWAY_NAME_I        |
    |          |________________________________________|
    |                    |
    |             _______|__        ______________________________
    |            |         \       |                              |
    |            | K = 0    \______| MLS_AVAIL_SSL = OFF          |
    |            |          /   |  |______________________________|
    |            |_________/    |
    |                           |
    |          _________________|__________________________________________
    |         |                                                            |
    |         | LAT_MLS_R_AZ_SSL = LAT_MLS_R_AZ_K                          |
    |         | LONG_MLS_R_AZ_SSL = LONG_MLS_R_AZ_K                        |
    |         | R_AZ_RADAR_BEARING_SSL = R_AZ_RADAR_BEARING_K              |
    |         | ALT_MLS_R_AZ_SSL = ALT_MLS_R_AZ_K                          |
    |         | LAT_MLSEL_SSL = LAT_MLSEL_K                                |
    |         | LONG_MLSEL_SSL = LONG_MLSEL_K                              |
    |         | ALT_MLSEL_SSL = ALT_MLSEL_K                                |
    |         | EL_SCANNER_BEARING_SSL = EL_SCANNER_BEARING_K              |
    |         | BIAS_MLSRANGE_SSL = BIAS_MLSRANGE_K                        |
    |         | BIAS_AZMLS_SSL = BIAS_AZMLS_K                              |
    |         | BIAS_ELMLS_SSL = BIAS_ELMLS_K                              |
    |         | X_DMEAZ_RW_SSL = X_DMEAZ_RW_K                              |
    |         | X_EL_RW_SSL = X_EL_RW_K                                    |
    |         | Y_DMEAZ_RW_SSL = Y_DMEAZ_RW_K                              |
    |         | MLS_AVAIL_SSL = ON                                         |
    |         |____________________________________________________________|
    |
    |           ______(transfer to alternate)
    |          |                                        |
    |___= 2____| RW_LAT_ASL = RW_LAT_I                  |
               | RW_LON_ASL = RW_LON_I                  |
               | RW_AZIMUTH_ASL = RW_AZIMUTH_I          |
               | RUNWAY_ALT_ASL = RUNWAY_ALT_I          |
               | RW_DELH_ASL = RW_DELH_I                |
               | RW_MAG_VAR_ASL = RW_MAG_VAR_I          |
               | RUNWAY_NAME_ASL = RUNWAY_NAME_I        |
               |________________________________________|
```

Figure E-2.- MAXI_RWY_TRANSFER CODE (Sheet 2 of 2).



TACAN_ID$_J$ = TACAN_ID_MAXI$_L$
LATITUDE_GEODETIC$_J$ = LATITUDE_GEODETIC_MAXI$_L$
LONGITUDE_EAST$_J$ = LONGITUDE_EAST_MAXI$_L$
ALT_ABOVE_ELLIPSOID$_J$ = ALT_ABOVE_ELLIPSOID_MAXI$_L$
MSL_ABOVE_ELLIPSOID$_J$ = MSL_ABOVE_ELLIPSOID_MAXI$_L$
MAGNETIC_VARIATION$_J$ = MAGNETIC_VARIATION_MAXI$_L$
BIAS_TAC_BRG$_J$ = 0.0
BIAS_TAC_R$_J$ = 0.0

Figure E-3.- MAXI_TAC_TRANSFER CODE.

APPENDIX F

NAVIGATION UPLINK PROCESSING FUNCTIONS: FLOWCHARTS, VARIABLE NAMES, AND DESCRIPTIONS

CONTENTS

Subject **Page**

Flowcharts

VARIABLES LIST DEFINITIONS

CODE USED FOR VARIABLE DATA TYPE

F: floating point quantity; an n-dimensional floating point vector will be denoted F(n); similarly, an n x m floating point matrix will be denoted by F(n,m)

I: integer quantity; I(n) will denote an n-dimensional integer vector

B: bit, i.e., data having only values of 0 or 1

C: character; C(n) will denote an n-dimensional character string

CODE USED FOR VARIABLE PRECISION

D: double precision

S: single precision. Integer quantities are assumed single precision unless otherwise stated.

VARIABLE LOCATION

COMPOOL: Variable value located in common storage, accessible by all functions

LOCAL: Variable is used by one function only, and usable to other functions through call argument only

VARIABLE INITIALIZATION CATEGORY

blank: display is vacant

C: constant (unchanging), Level A

DD: design dependent

HC: hard coded

MD: mission dependent (I-Load)

VARIABLE INITIAL VALUE

Initial operation sequence computer inputs

VARIABLE UPLINK AND DOWNLIST STATUS

UPLINK:	variable is an uplink item
DOWNLIST:	variable is a downlist item

UNITS DEFINITIONS

deg:	angular measurement, degrees
ft:	feet
lb:	pounds
n.d.:	non-dimensional
rad:	radian
sec:	time measurement, seconds
slugs:	mass measurement, slugs
vary:	units have different values which depend on variable use

VARIABLES LIST

PRECEDING PAGE BLANK NOT FILMED

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial-ization category	Initial value	Uplink/ downlist	Units	Description
COV_COR_UPDATE (V96H1290C -6C)	SF (7)	compool	MD	—	uplink	vary	Vector of correlation coefficients associated with UVW standard deviations (SIG_UPDATE) used for Orbiter position/ velocity covariance initialization (ground update).
DO_CV_UPLINK (V90X4754X)	B	compool	--	OFF	—	--	Flag indicating (ON) that data has been uplinked for Orbiter state vector update.
DO_TV_UPLINK (V90X4757X)	B	compool	—	OFF	—	--	Flag indicating (ON) that data has been uplinked for target state vector update.
R _GND (V96H3500C -02C)	DF (3)	compool	--	--	uplink	ft	Uplink Orbiter position vector (M50).
R _TV_GND (V96H1277C -79C)	DF (3)	compool	—	—	uplink	ft	Uplinked target vehicle position vector at T_TV_GND.
SIG_UPDATE (V96H1284C -9C)	SF (6)	compool	MD	--	uplink	--	Vector of standard deviations for Orbiter position/velocity covariance initialization (ground update).
T_GND (V96W3520C)	DF	compool	—	--	uplink	sec	Uplinked time tag of Orbiter state vector (R _GND, V _GND).
T_TV_GND (V96W1283C)	DF	compool	--	—	uplink	sec	Uplinked target state time tag.
V _GND (V96L3510C -12C)	DF (3)	compool	—	--	uplink	ft/sec	Uplinked Orbiter velocity vector.
V _TV_GND (V96L1280C -82C)	DF (3)	compool	--	--	uplink	ft/sec	Uplinked M1950 target vehicle velocity vector at T_TV_GND.
MLS_AVAIL_PSL	B	compool	MD	--	—	--	MSBLS available discrete for primary and secondary runways.
MLS_AVAIL_SSL	B	compool	MD	--	—	--	

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial- ization category	Initial value	Uplink/ downlist	Units	Description
RUNWAY_ALT	SF (18)	compool	MD	--	--	ft	Altitude above reference ellipsoid of runway reference point for maxi table runways.
RUNWAY_ALT_ASL	SF	compool	MD	--	--	ft	Altitude above reference ellipsoid of runway reference point for primary, secondary, and alternate runways.
RUNWAY_ALT_PSL	SF	compool	MD	--	--	ft	
RUNWAY_ALT_SSL	SF	compool	MD	--	--	ft	
RUNWAY_NAME	C (18)	compool	MD	--	--	--	Name of landing site in maxi table.
RUNWAY_NAME_PSL	C	compool	MD	--	--	--	Name of landing site in mini table for primary, secondary, and alternate runways.
RUNWAY_NAME_SSL	C	compool	MD	--	--	--	
RUNWAY_NAME_ASL	C	compool	MD	--	--	--	
RW_AZIMUTH	SF (18)	compool	MD	--	--	rad	Bearing from true north of runway coordinate system +X axis for maxi table runways.
RW_AZIMUTH_ASL	SF	compool	MD	--	--	rad	Bearing from true north of runway coordinate system +X axis for primary, secondary, and alternate mini table runways.
RW_AZIMUTH_PSL	SF	compool	MD	--	--	rad	
RW_AZIMUTH_SSL	SF	compool	MD	--	--	rad	
LND_SITE_NO (V96U1207C)	SI	compool	--	--	uplink	--	Landing site table slot (1 to 18 - maxi, 19 to 21 - mini).
ALT_ABOVE_ELLIPSOID	SF (10)	compool	MD	--	--	ft	Height of TACAN site above reference ellipsoid in mini table.
ALT_ABOVE_ELLIPSOID_MAXI	SF (50)	compool	MD	--	--	ft	Height of TACAN site above reference ellipsoid in maxi table.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial-ization category	Initial value	Uplink/ downlist	Units	Description
BIAS_TAC_BRG	SF (10)	compool	MD	--	--	rad	Bias in TACAN bearing in mini table.
BIAS_TAC_R	SF (10)	compool	MD	--	--	ft	Bias in TACAN range in mini table.
LATITUDE_GEODETIC	SF (10)	compool	MD	--	--	rad	Geodetic latitude at TACAN site in mini table.
LATITUDE_GEODETIC_MAXI	SF (50)	compool	MD	--	--	rad	Geodetic latitude at TACAN site in maxi table.
LONGITUDE_EAST	SF (10)	compool	MD	--	--	rad	Longitude at TACAN site in mini table.
LONGITUDE_EAST_MAXI	SF (50)	compool	MD	--	--	rad	Longitude at TACAN site in maxi table.
MAGNETIC_VARIATION	SF (10)	compool	MD	--	--	rad	Magnetic variation of TACAN site in mini table.
MAGNETIC_VARIATION_ MAXI	SF (50)	compool	MD	--	--	rad	Magnetic variation at TACAN site in maxi table.
MSL_ABOVE_ELLIPSOID	SF (10)	compool	MD	--	--	ft	MSL correction to reference ellipsoid at the TACAN site in mini table.
MSL_ABOVE_ELLIPSOID_ MAXI	SF (50)	compool	MD	--	--	ft	MSL correction to reference ellipsoid at the TACAN site in maxi table.
TAC_ID	SI (10)	compool	MD	--	--	--	TACAN channel number in mini table. Positive for x-mode. Negative for y-mode.
TACAN_ID_MAXI	SI (50)	compool	MD	--	--	--	TACAN channel number in maxi table. Positive for x-mode. Negative for y-mode.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial-ization category	Initial value	Uplink/ downlist	Units	Description
ALT_ABOVE_ELLIPSOID_UL (V96H1220C)	SF	compool	—	—	uplink	ft	Height of uplinked TACAN site above reference ellipsoid.
BIAS_TAC_BRG_UL (V96H1223C)	SF	compool	—	—	uplink	rad	Uplinked TACAN bearing bias.
BIAS_TAC_R_UL (V96H1224C)	SF	compool	—	—	uplink	ft	Uplinked TACAN range bias.
LATITUDE_GEODETIC_UL (V96H1218C)	SF	compool	—	—	uplink	rad	Geodetic latitude at uplinked TACAN site.
LONGITUDE_EAST_UL (V96H1219C)	SF	compool	—	—	uplink	rad	Longitude of uplinked TACAN site.
MAGNITUDE_VARIATION_UL (V96H1222C)	SF	compool	—	—	uplink/ downlist	rad	Magnetic variation at uplinked TACAN site.
MSL_ABOVE_ELLIPSOID_UL (V96H1221C)	SF	compool	—	—	uplink/ downlist	ft	Altitude of mean sea level above reference ellipsoid at uplinked TACAN site.
DO_TACAN_SITE_UPLINK (V95X1275X)	B	compool	—	—	uplink/ downlist	—	TACAN site uplink command.
TAC_SITE_NO (V96J1216C)	SI	compool	—	—	uplink/ downlist	—	TACAN site table slot number.
TACAN_ID_UL (V96J1217C)	SI	compool	—	—	uplink/ downlist	—	Channel/mode identifier for uplinked TACAN site.
BUFFER_B	SF (3)	local	—	—	—	lbs	Buffer area associated with VENT/RCS vector uplink.
BUFFER_C	SF (7)	local	—	—	—	—	Buffer area associated with covariance matrix uplink.

VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial-ization category	Initial value	Uplink/ downlist	Units	Description
BUFFER_K	SP	local	—	—	—	--	Buffer location associated with drag model correction factor.
BUFFER_O	SP (2)	local	—	—	—	sec	Buffer area associated with Vent/RCS ON-OFF times uplink.
BUFFER_R	DP (3)	local	—	—	—	ft	Buffer area associated with Orbiter state position vector uplink.
BUFFER_RT	DP (3)	local	—	—	—	ft	Buffer area associated with target vehicle position vector uplink.
BUFFER_S	SP (6)	local	—	—	—	ft, ft/sec	Buffer designations for UVW standard deviations associated with covariance matrix uplink.
BUFFER_T	DP	local	—	—	—	sec	Buffer location associated with Orbiter state vector uplink (time tag).
BUFFER_TT	DP	local	—	—	—	sec	Buffer location associated with rendezvous vehicle state vector uplink (time tag).
BUFFER_V	SP (3)	local	—	—	—	ft/sec	Buffer area associated with Orbiter state vector uplink (velocity).
BUFFER_VT	SP (3)	local	—	—	—	ft/sec	Buffer area associated with target vehicle state vector uplink (velocity).
KFACTOR (V96U8173C)	SP	compool	MD	--	uplink	--	Drag model correction coefficient.
OP_CODE	SI	compool	—	—	uplink/ downlist	--	7-BIT string code to define command load uplink type.
TFOFF (V96W8178C)	DP	compool	MD	-	uplink/ downlist	sec	Vent/RCS off time.

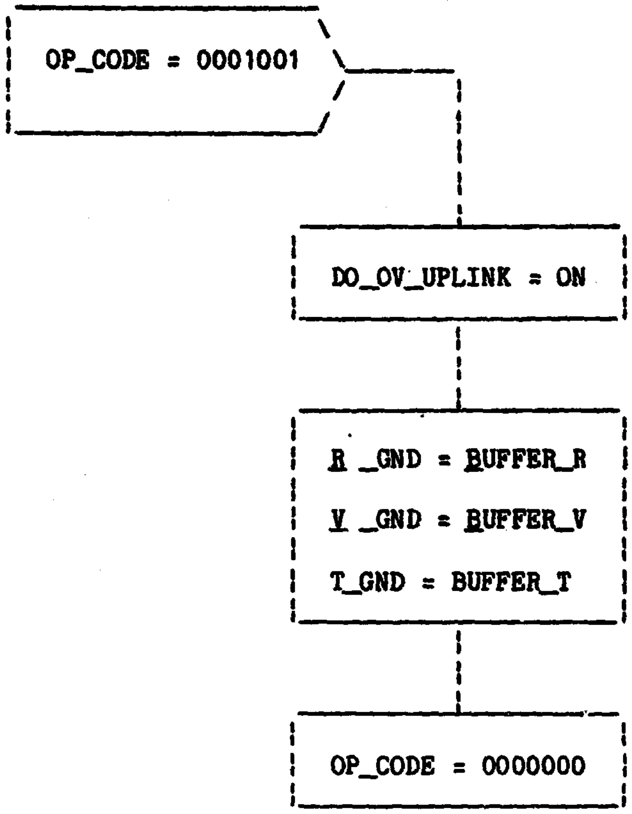
VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initial-ization category	Initial value	Uplink/ downlist	Units	Description
TPON(V96W8177C)	DP	compool	MD	--	uplink downlist	sec	Vent/RCS on time.
VFORCE(V96U8174C -76C)	SP (3)	compool	MD	--	uplink downlist	lbs	Vent/RCS body force vector.
PRI_MAXI_UL (V96U8281C)	SI	compool	--	--	uplink/ downlist	--	Runway selection index for maxi table to primary runway data transfer.
SEC_MAXI_UL (V96U8282C)	SI	compool	--	--	uplist/ downlist	--	Runway selection index for maxi table to secondary runway data transfer.
ALT_MAXI_UL (V96U8283C)	SI	compool	--	--	uplink/ downlist	--	Runway selection index for maxi table to alternate runway data transfer.
DO_GUID_UPLINK (V96U8280CB)	B	compool	--	--	uplink/ downlist	--	Flag indicating (ON) that landing site selection parameters have been uplinked.
PRI_MAXI_SELECT	SI	compool	--	19	--	--	Runway selection index for maxi table to primary runway data transfer.
SEC_MAXI_SELECT	SI	compool	--	20	--	--	Runway selection index for maxi table to secondary runway data transfer.
ALT_MAXI_SELECT	SI	compool	--	21	--	--	Runway selection index for maxi table to alternate runway date transfer.
ITEM1_IN	B	compool	--	OFF	--	--	Maxi table to primary runway transfer discrete.
ITEM2_IN	B	compool	--	OFF	--	--	Maxi table to secondary runway transfer discrete.
ITEM3_IN	B	compool	--	OFF	--	--	Maxi table to alternate runway transfer discrete.

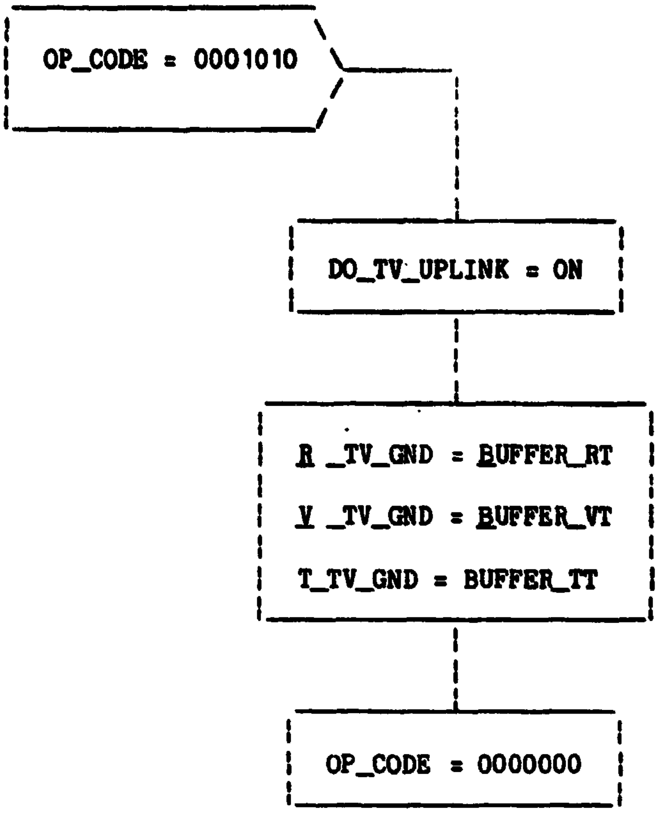
VARIABLES LIST

Variable name (M/S ID)	Precision & type	Compool or local	Initialization category	Initial value	Uplink/ downlist	Units	Description
DO_LND_SITE_UPLINK (V96X1274X)	B	compool			uplink/ downlist		Landing site uplink command.
RUNWAY_NAME_UL (V96J1208C)	SF	compool		uplink	uplink	--	Name of runway at uplinked landing site.
RUNWAY_ALT_UL (V96H1214C)	SF	compool		uplink	uplink	ft	Height of uplinked landing site above reference ellipsoid.
RW_LAT_UL (V96H1210C)	SF	compool		uplink	uplink	rad	Geodetic latitude at uplinked landing site.
RW_LON_UL (V96H1211C)	SF	compool		uplink	uplink	rad	Longitude at uplinked landing site.
RW_AZIMUTH_UL (V96H1212C)	SF	compool		uplink	uplink	rad	Runway heading for uplinked landing site.
RW_DELH_UL (V96H1215C)	SF	compool		uplink	uplink	ft	Altitude at mean sea level above reference ellipsoid at uplinked landing site.
RW_MAG_VAR_UL (V96H1213C)	SF	compool		uplink	uplink	rad	Magnetic variation at uplinked landing site.

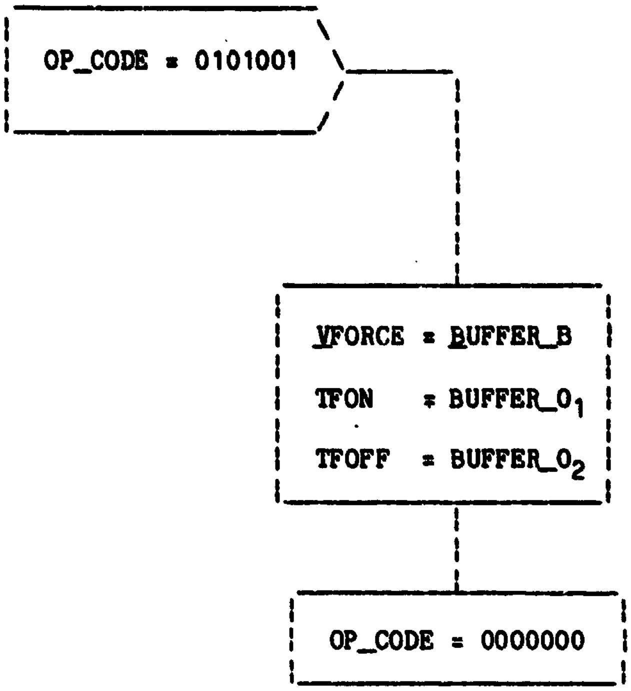
Flowcharts



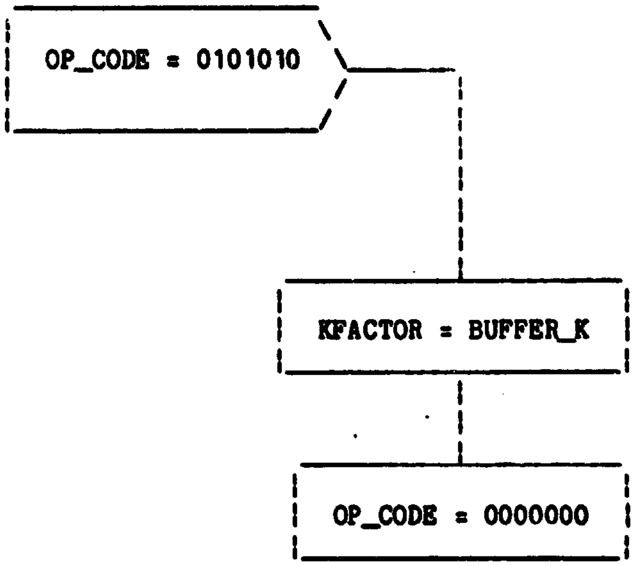
ORBITER_STATE_VECTOR_UPLINK



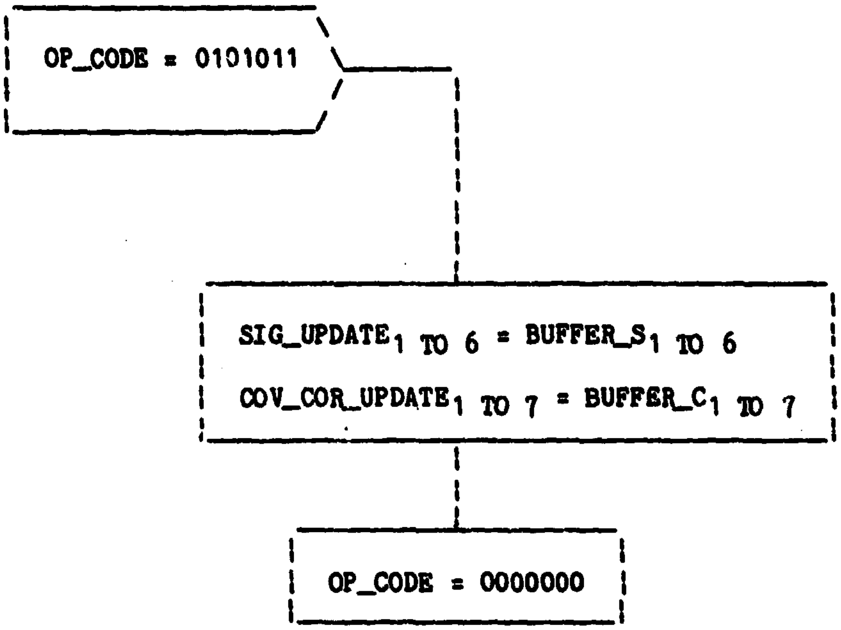
RENDEZVOUS_STATE_VECTOR_UPLINK



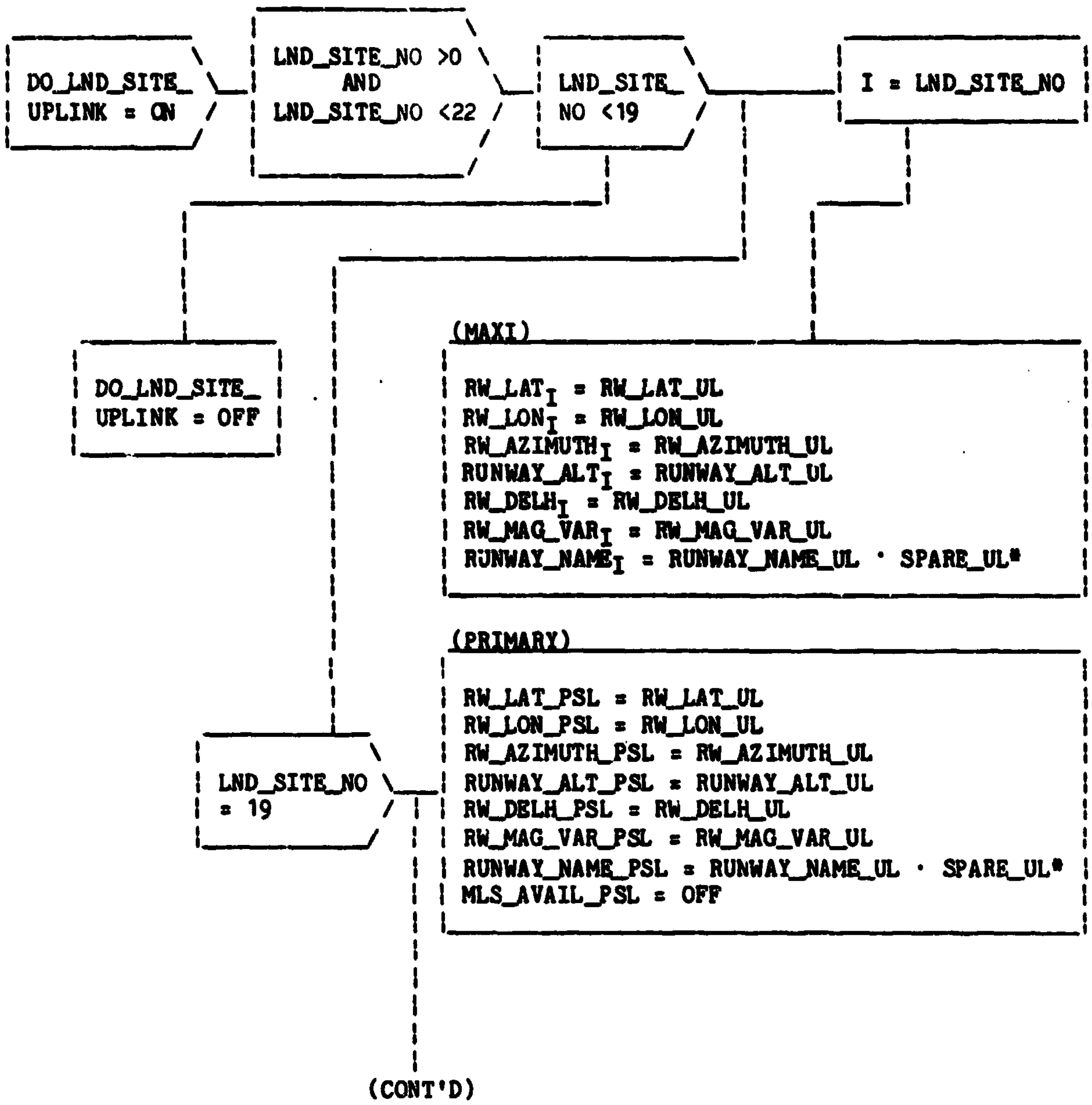
VENT_RCS_MODEL_PARAM_UPLINK



DRAG_MODEL_PARAM_UPLINK

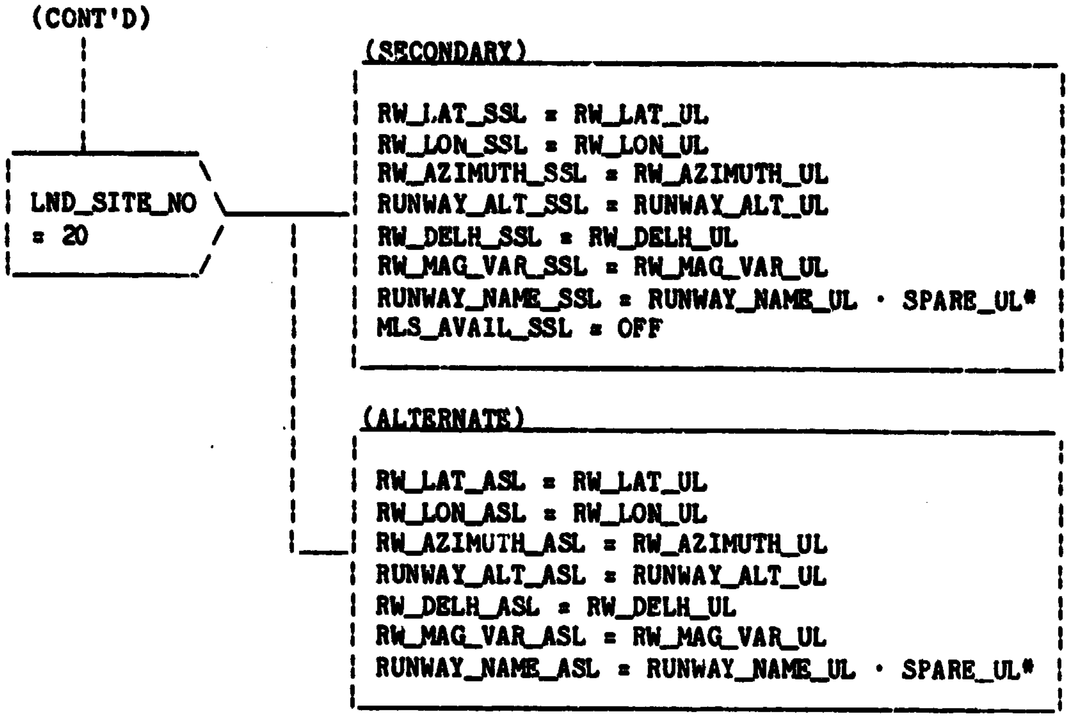


COV_MATRIX_PARAM_UPLINK



*Implies catenation of first character string with first character of second character string.

LANDING_SITE_MAXI_MINI_UPLINK (Sheet 1 of 2).



(SECONDARY)

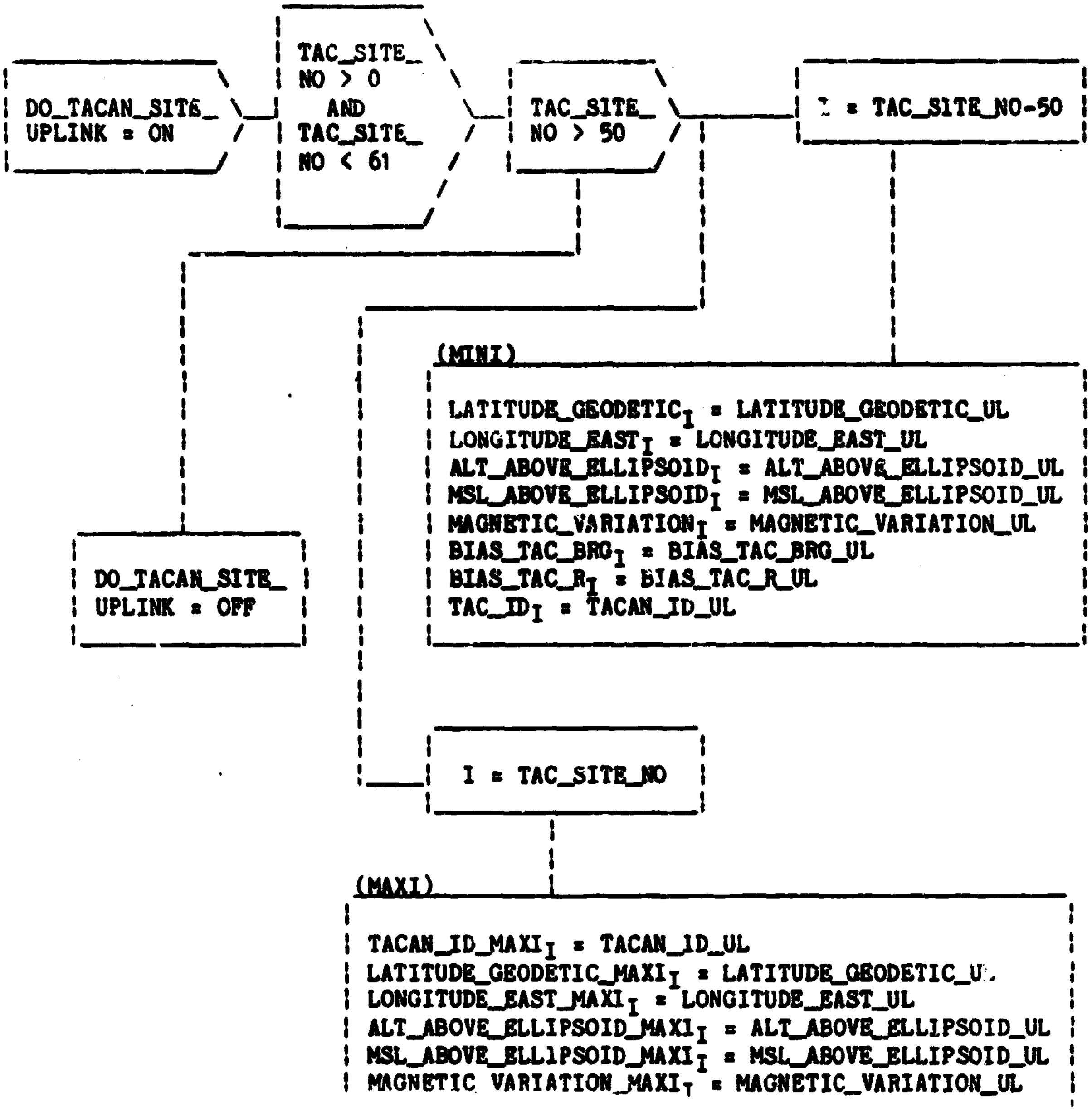
RW_LAT_SSL = RW_LAT_UL
RW_LON_SSL = RW_LON_UL
RW_AZIMUTH_SSL = RW_AZIMUTH_UL
RUNWAY_ALT_SSL = RUNWAY_ALT_UL
RW_DELH_SSL = RW_DELH_UL
RW_MAG_VAR_SSL = RW_MAG_VAR_UL
RUNWAY_NAME_SSL = RUNWAY_NAME_UL · SPARE_UL*
MLS_AVAIL_SSL = OFF

(ALTERNATE)

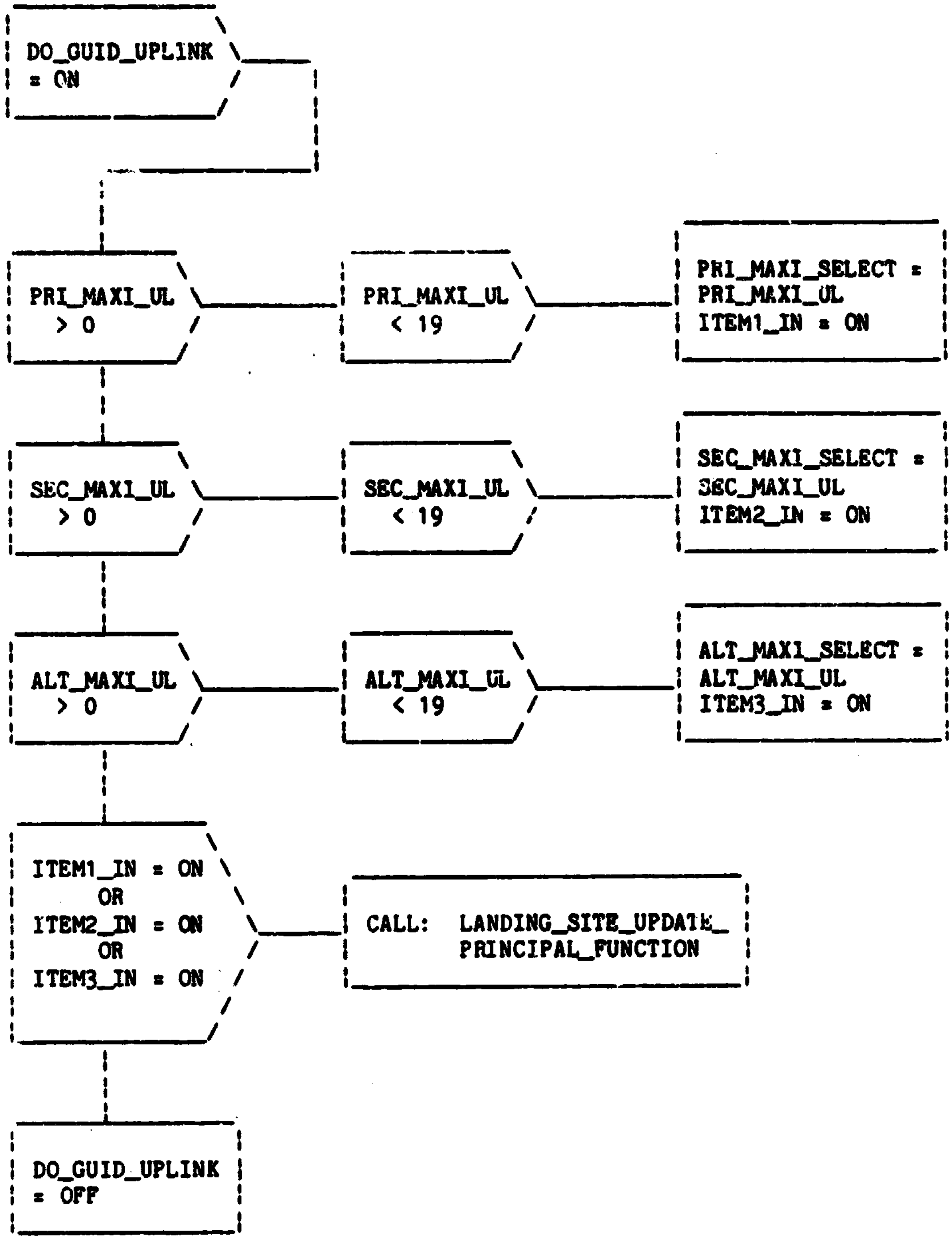
RW_LAT_ASL = RW_LAT_UL
RW_LON_ASL = RW_LON_UL
RW_AZIMUTH_ASL = RW_AZIMUTH_UL
RUNWAY_ALT_ASL = RUNWAY_ALT_UL
RW_DELH_ASL = RW_DELH_UL
RW_MAG_VAR_ASL = RW_MAG_VAR_UL
RUNWAY_NAME_ASL = RUNWAY_NAME_UL · SPARE_UL*

*Implies catenation of first character string with first character of second character string.

LANDING_SITE_MAXI_MINI_UPLINK (Sheet 2 of 2).



TACAN_SITE_MAXI_MINI_UPLINK.



SITE_SELECTION_UPLINK.

APPENDIX G

INTERFACE DIAGRAMS FOR THE ONORBIT/RENDEZVOUS NAVIGATION SEQUENCER PRINCIPAL FUNCTION, THE ONORBIT/RENDEZVOUS NAVIGATION PRINCIPAL FUNCTION AND THE ONORBIT PRECISION STATE PREDICTION PRINCIPAL FUNCTION

This appendix provides supplementary interface information concerning the following principal functions:

- Onorbit/Rendezvous Navigation Sequencer
- Onorbit/Rendezvous Navigation
- Onorbit Precision State Prediction

The intent of this appendix is to provide block diagrams for each principal function in order to supply the following information:

a. The constituent subfunctions of each principal function.

b. A calling diagram for each subfunction in each principal function.

c. The interrelationship of the principal functions.

The information contained in this appendix is intended to be supplementary only.

CONTENTS

PRECEDING PAGE BLANK NOT FILMED

PRECEDING PAGE BLANK NOT FILMED

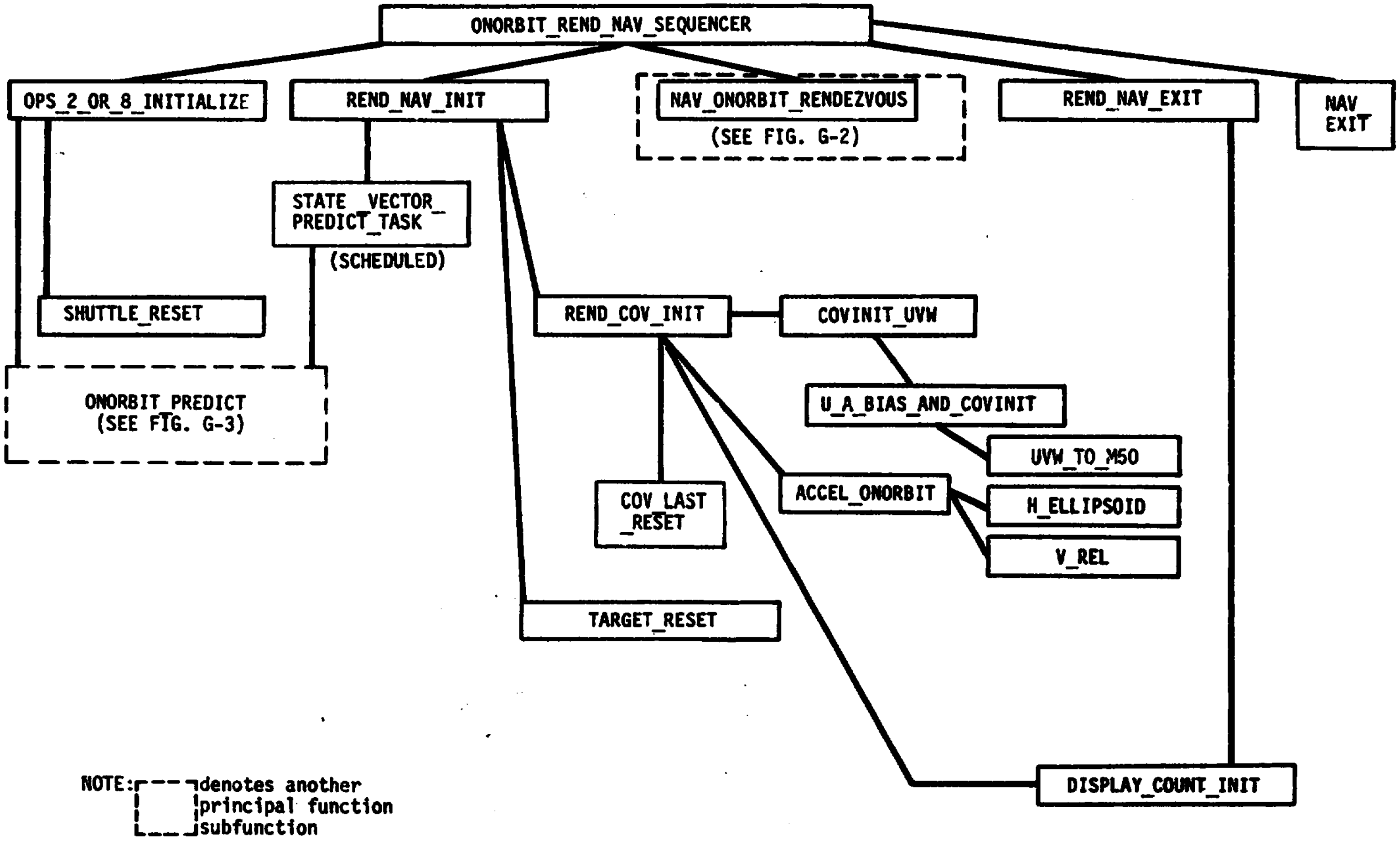


Figure G-1.- The onorbit/rendezvous navigation sequencer principal function.

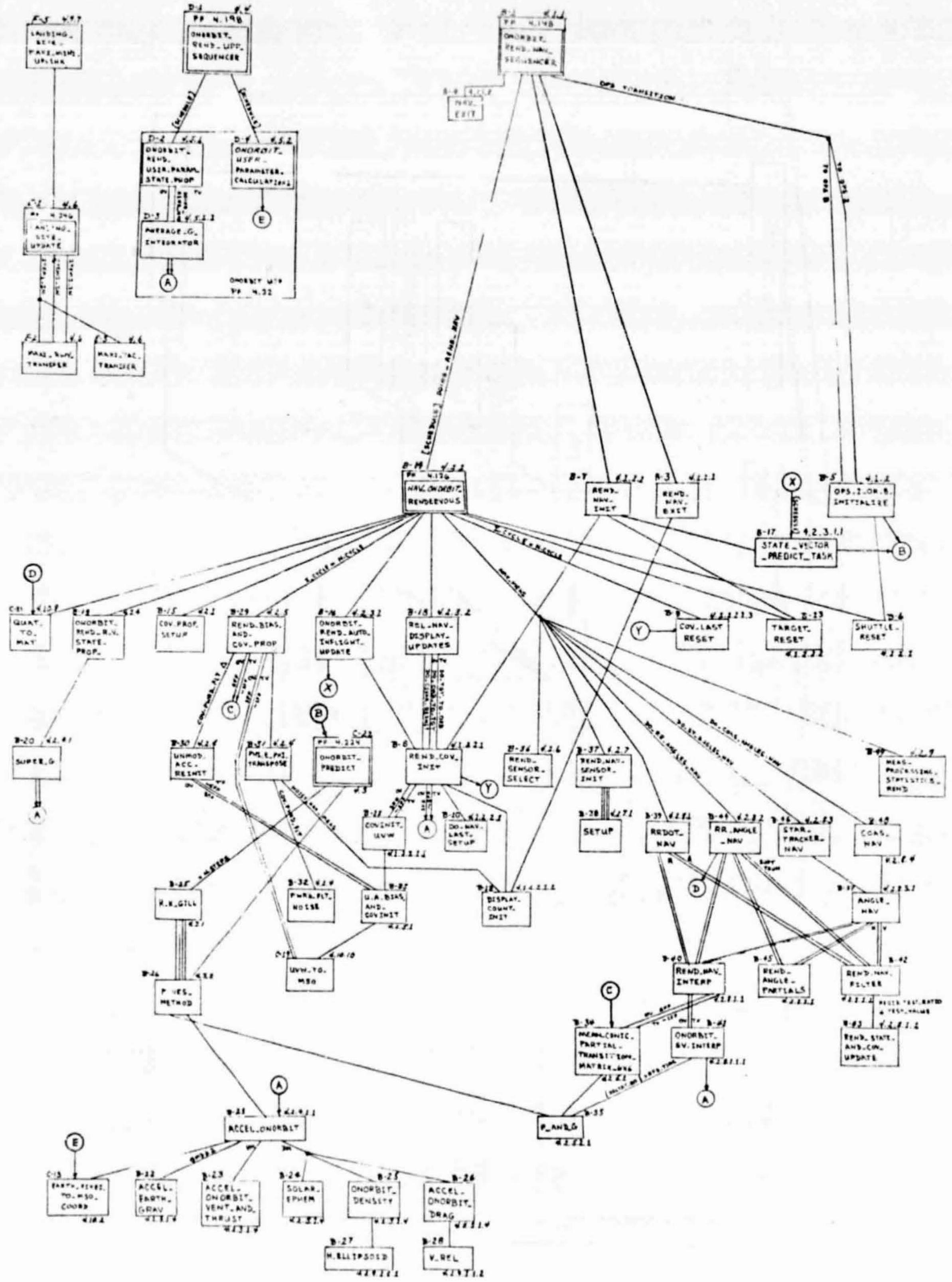


FIGURE G-2.- MODULE INTERFACE FOR ONORBIT/RENDEZVOUS NAVIGATION.

ORIGINAL PAGE IS
OF POOR QUALITY

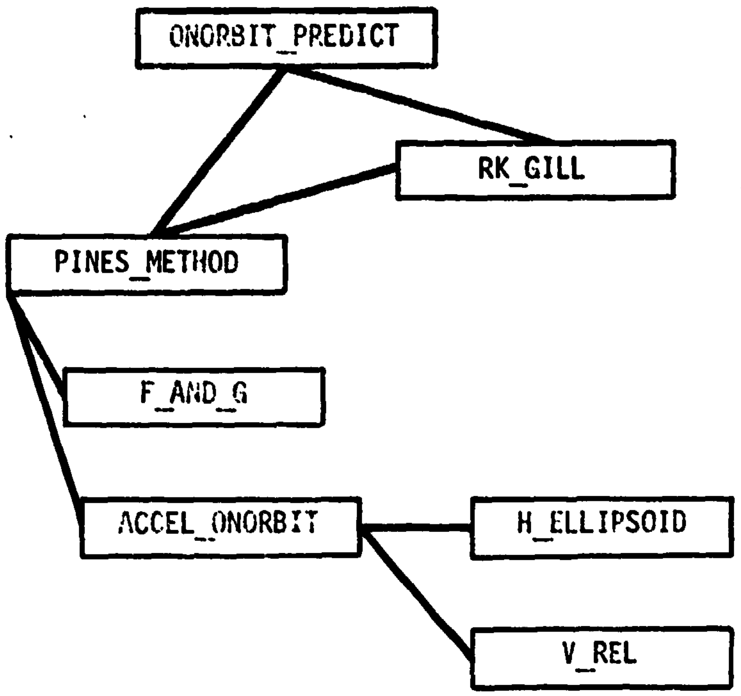


Figure G-3.- The onorbit precision state prediction principal function.

U.S. GOVERNMENT PRINTING OFFICE: 1980—671-099 23 78