78-FM-43
(Supersedes 78-FM-27)

JSC-14393

Orbit Targeting Specialist Function

Level C Formulation Requirements

Mission Planning and Analysis Division

August 1978

(NASA-TM-79755) ORBIT TARGETING SPECIALIST FUNCTION: LEVEL C FORMULATION REQUIREMENTS (NASA) 166 p HC A08/MF A01 CSCL 22A

N79-10077

Unclas
G3/13 35937

NASA

National Aeronautics and
Space Administration

Lyndon B. Johnson Space Center
Houston, Texas

78-FM-43
(Supersedes 78-FM-27)

JSC-14393

SHUTTLE PROGRAM

ORBIT TARGETING SPECIALIST FUNCTION

LEVEL C FORMULATION REQUIREMENTS

By A. DuPont, S. McAdoo, and H. Jones, Flight Planning Branch;
and A. K. Jones and D. Pearson, McDonnell Douglas Technical Services Co.

JSC Task Monitor: A. DuPont, Flight Planning Branch

Approved: [signature]
Edgar C. Lineberry, Chief
Flight Planning Branch

Approved: [signature]
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

August 1978

CONTENTS

TABLES

FIGURES

ORBIT TARGETING SPECIALIST FUNCTION

LEVEL C FORMULATION SPECIFICATION

By A. DuPont, S. McAdoo, and H. Jones, Flight Planning Branch;
and A. K. Jones and D. Pearson, McDonnell Douglas Technical
Services Co.

1.0 SUMMARY

This document contains a definition of the Mission Planning and Analysis Division (MPAD) level C requirements for onboard maneuver targeting software. Included are revisions of the level C software requirements delineated in JSC IN 78-FM-27, Proximity Operations Software; Level C Requirements, dated May 1978. The software will support the terminal phase midcourse (TPM) maneuver, braking and close-in operations as well as supporting computation of the rendezvous corrective combination maneuver (NCC) and the terminal phase initiation (TPI). The second corrective combination maneuver (NSR), which is designed to create an Orbiter orbit coelliptic with the target orbit, is not available. Another NCC to an offset will be used as a substitute. Specific formulation is contained here for the orbit targeting specialist function including the processing logic, linkage, and data base definitions for all modules. The crew interface with the software is through the keyboard and the ORBIT_TGT display.

2.0 INTRODUCTION

The requirements that are provided define software to support maneuver targeting for rendezvous and close-in operations. A primary input is an accurate navigated state vector. To maintain a relative navigation state for maneuver targeting and vehicle control, relative navigation and the processing of sensor tracking data from the star tracker, crew optical alinement sight, or the Ku-band rendezvous radar are required. Prior to the TPI maneuver, sensor updates are incorporated only during the coasting major mode. After TPI, the sensor update will be continuously processed except during periods of Orbiter accelerations above a TBD threshold. Access and control of the software will be through the keyboard and the ORBIT_TGT display. Definition is provided for input, output, and control. The maneuver targeting software consists of two-impulse Lambert software and two-impulse Clohessy-Wiltshire (CW) software, each capable of targeting to an offset position. The maneuvers are displayed to the crew and transferred to guidance for manual execution. Implementation of these requirements will provide the crew with software support to reduce crew workload, to reduce reaction control system (RCS) propellant usage, and to fly a special trajectory.

3.0 ACRONYMS

CW	Clohessy-Wiltshire
CRT	cathode-ray tube
delta-t	delta-time
delta-v	delta-velocity
GN&C	guidance, navigation, and control
I/O	input/output
LV	local vertical
LVC	local vertical rotating curvilinear
LVLH	local vertical local horizontal
LVIR	local vertical inertial rectangular
NCC	corrective combination maneuvers
RCS	reaction control system
TBD	to be determined
TPI	terminal phase initiation
TPM	terminal phase midcourse

4.0 ORBIT TARGETING SPECIALIST FUNCTION

The orbit targeting specialist function gives the Orbiter crew the capability to generate targeted maneuvers so as to generate vehicle motion from a given position to a desired target referenced relative position. Profile examples that could utilize this targeting function include NCC, TPI, TPM, terminal phase finalization (TPF), braking, stationkeeping, transition, approach, and separation. The targeted maneuver outputs of this specialist function will be available for display to the crew and as inputs to the maneuver execution software functions, including the maneuver guidance function.

The targeting algorithm computes either maneuver of a two-maneuver set. The initial maneuver of the set is targeted to achieve a given position relative to the target vehicle in a given delta-time (delta-t). The second maneuver is targeted to null the Orbiter's velocity relative to a target-centered rotating reference frame.

The targeting algorithm has as inputs a set of constraints (denoted as a target set) that are selected by crew action via the orbit targeting specialist function display (fig. 1). The selected target set can be one of 40 sets available in the I-load or from ground uplink. Lambert maneuvers will be computed for the first NLAMB target sets. Closed-form CW equations will be used to compute maneuvers for the rest of the target sets. The crew has the capability via display input to modify a selected target set and have the appropriate I-load changed accordingly.

In general, a sequence of rendezvous maneuvers, braking maneuvers, stationkeeping maneuvers, or transition maneuvers can be performed by selecting the appropriate target set, computing the solution, executing the indicated maneuver, and then repeating the process for each remaining maneuver in the sequence.

The following sections describe the detailed logic and equations necessary to support orbit operations targeting.

The specifications for the orbit targeting specialist function are presented as a set of modular tasks. These tasks are listed as follows and are numbered with corresponding section numbers of this document in which the detailed requirements are presented.

4.1 Proximity operations targeting executive task (PROX_EXEC)

4.2 Proximity operations targeting status task (PROX_STAT)

4.3 Proximity operations targeting target set select task (PROX_TGT_SEL)

4.4 Proximity operations targeting initialization task (PROX_INIT)

4.5 Proximity operations targeting guidance quantity transfer task (PROX_TRANS)

4.6 Proximity operations targeting supervisory logic task (PROX_TGT_SUP)

4.7 Proximity operations targeting supervisory-Lambert logic task (PROX_TGT_SUP_LAMB)

4.8 Proximity operations targeting start timer task (PROX_STIME)

4.9 Maneuver to offset targeting task (OFFSET_TGT)

4.10 Relative state predictor task (REL_PRED)

4.11 Relative state compute task (REL_COMP)

4.12 Proximity operations targeting output display load task (PROX_DISP_LOAD)

4.13 Time conversion task (TIME_CONVRT)

4.14 Delta-t compute task (DT_COMP)

4.15 Omega-dt compute task (OMEGA_DT_COMP)

4.16 Elevation angle search task (TELEV)

4.17 Precision velocity required task (PREVR)

4.18 Elevation angle computation task (COMELE)

4.19 Elevation angle iteration task (ELITER)

4.20 Lambert conic-velocity-required task (LAMBERT)

4.21 State vector update task (UPDATVP)

4.22 Newton-Raphson iteration task (ITERV)

4.23 Orbiter LVLH transformation task (ORBLV)

The organization of the tasks is shown in figure 2. The main control is in the executive task (PROX_EXEC). The executive calls other tasks that monitor the status and provide for display and guidance interfacing. It also calls one of two targeting supervisory tasks to compute either a maneuver using CW equations or a Lambert maneuver (PROX_TGT_SUP and PROX_TGT_SUP_LAMB, respectively). A general data flow of the orbit targeting specialist function is shown in figure 3.

To facilitate understanding of the data flow interfaces, all orbit targeting specialist function variables are placed in one of the five common packages:

a. PROX_WORLD_COMMON contains all variables that are passed between the specialist function and other guidance, navigation, and control (GN&C) functions.

b. PROX_DIP_COMMON contains all variables that are passed between the specialist function and the ORBIT_TGT_DIP.

c. PROX_ILOAD_COMMON contains all variables that are input data to the specialist function.

d. PROX_VARIABLES_COMMON contains all variables internal to the specialist function.

e. PROX_LAMVAR_COMMON contains variables internal to the Lambert routines of the specialist function.

The specific contents of each of these common packages are listed in tables 1 through 5. The common packages available to each of the orbit targeting specialist function modules are shown in figure 4.

In the following sections each function module is explained, and specific input/output (I/O) variables for each module are listed.

4.1 PROXIMITY OPERATIONS TARGETING EXECUTIVE TASK (PROX_EXEC)

This cyclically executed module (fig. 5) is the top-level module of the orbit targeting specialist function. A functional flow of the executive task is shown in figure 5. It performs as follows.

a. Initializes data

b. Responds to the crew changes in the display area

c. Sets status discretes for display items to be flashed and for display items to carry stars

d. Transfers target set data between the I-load buffers and the display and computational buffers

e. Calls the task to compute the relative state

f. Calls the appropriate targeting supervisory logic task

g. Calls the task to load the output display data buffers

h. Calls the task to load the guidance buffers with data for the upcoming maneuver

i. Calls the task to load the time management buffer with the time to be counted down to the next maneuver.

This module design assumes that a higher-level software calls or schedules PROX_EXEC whenever the specialist function is called by a crew request. It is assumed that the higher-level software will cycle PROX_EXEC at a TBD frequency until the crew terminates the specialist function.

4.1.1 Detailed Requirements

The following steps are required to perform the orbit targeting function.

a. Perform a logic test to determine if this is the first pass through the executive.

   If first pass (PROX_FIRST_PASS_STATUS = ON)

   (1) Set the prox base time to the ILOAD values by setting up the inputs and calling the time conversion task (TIME_CONVRT). The TIME_CONVRT detailed requirements are presented in section 4.13.

       TIME_CONVERT_FLAG = 1

       DAY = BASE_START_DAY

       HR = BASE_START_HOUR

       MIN = BASE_START_MIN

       SEC = BASE_START_SEC

       Call TIME_CONVRT; inputs: TIME_CONVERT_FLAG, DAY, HR, MIN, SEC

       output: TIME_SEC

       PROX_BASE_TIME = TIME_SEC

   (2) Put the base time into the display.

       PROX_BASE_DAY = BASE_START_DAY

       PROX_BASE_HR = BASE_START_HR

       PROX_BASE_MIN = BASE_START_MIN

       PROX_BASE_SEC = BASE_START_SEC

   (3) Initialize T1_TIG and T2_TIG.

       T1_TIG = 0

       T2_TIG = 0

   (4) Initialize the compute in progress status flags.

       PROX_T1_STAR_STATUS = OFF

       PROX_T2_STAR_STATUS = OFF

(5) Set the first pass status off.

PROX_FIRST_PASS_STATUS = OFF

b. Perform a logic test to determine if the crew made an entry to items 21, 22, 23, or 24.

If an entry was made (PROX_ITEM_21TO24_STATUS = ON), set the LOAD to flash (PROX_LOAD_FLASH = ON).

c. Call the proximity operations targeting status task (sec. 4.1) to set the maneuver status flag to on or off position if the maneuver TIG time is prior to or past current time. The inputs and outputs are listed in tables 6 and 7.

d. Perform a logic test to determine if the crew has made an entry to item 1. If so (PROX_ITEM_1_STATUS = ON):

(1) Call the proximity operations target set select task (sec. 4.3) to load input display buffer and computation buffer from the ILOAD buffer.

(2) Set PROX_ITEM_1_STATUS = OFF.

e. Perform a logic test to determine if the crew executed item 25. If an execution was made (PROX_ITEM_25_STATUS = ON):

(1) Transfer the T2 maneuver time data in the computation buffer to the T1 maneuver time computational buffer slots.

T1_TIG = T2_TIG

(2) Transfer the T2 maneuver time data in the computation buffer to the T1 maneuver time display buffer slots.

TIME_CONVERT_FLAG = 0

TIME_SEC_ = T1_TIG

Call TIME_CONVRT; inputs: TIME_CONVERT_FLAG, TIME_SEC

outputs: DAY, HR, MIN, SEC

DISP_T1_DAY = DAY

DISP_T1_HR = HR

DISP_T1_MIN = MIN

DISP_T1_SEC = SEC

(3) Set the status flag to off.

PROX_ITEM_25_STATUS = OFF

f. Perform a logic test to determine if the crew made any entries to items 2 through 20.

(1) If entry was made (PROX_ITEM_2TO20_STATUS = ON), set flag to flash "LOAD".

PROX_LOAD_FLASH = ON

g. Perform a logic test to determine if the crew executed item 26. If execution was made (PROX_ITEM_26_STATUS = ON):

(1) Convert the input data display buffer items and store in the computational data buffers and the I-load data buffers. The conversion is performed by the proximity operations targeting initialization task (sec. 4.4). The inputs and outputs are presented in tables 8 and 9.

(2) Set status flags to the off condition.

PROX_ITEM_2TO20_STATUS = OFF

PROX_ITEM_21TO24_STATUS = OFF

PROX_LOAD_FLASH = OFF

PROX_ITEM_26_STATUS = OFF

h. Perform a logic test to determine if the crew executed item 27.

If so (PROX_ITEM_27_STATUS = ON):

(1) Set the compute T1 star status flag to on.

PROX_T1_STAR_STATUS = ON

(2) Perform a logic test to see if any of the display buffers for T1 relative state are blank (items 7, 8, 9, 10, 11, 12), or if the T1 maneuver time is in the past (T1_TIG < PROX_T_CURRENT).

(a) If so, set USE_DISP_REL_STATE = OFF

(b) If not, set USE_DISP_REL_STATE = ON

i. Perform a logical test to determine if the crew executed item 28.

If so (PROX_ITEM_28_STATUS = ON):

(1) Set the compute T2 star status flag to on.

PROX_T2_STAR_STATUS = ON

(2) Set the use relative display to off.

USE_DISP_REL_STATE = OFF

(3) Set the item 28 status to off.

PROX_ITEM_28_STATUS = OFF

j. Perform a logic test to determine if the compute T1 or compute T2 solutions were requested.

If a request was made (PROX_T1_STAR_STATUS = ON or PROX T2_STAR_STATUS = ON):

(1) Retrieve the current Orbiter and target state vectors from ON_ORB_UPP.

RS_M50_PROX = R_AVGG

VS_M50_PROX = V_AVGG

RT_M50_PROX = R_TARGET

VT_M50_PROX = V_TARGET

(2) Retrieve the ON_ORB_UPP nav state time tag (T_STATE), assumed GMT, if the use display flag is off or a Lambert solution is requested.

If USE_DISP_REL_STATE = OFF or PROX_TGT_SET_NO $\leq$ NLAMB,

set TIME_PROX = T_STATE-BASE_MET

(3) Convert the orbital angular rate from the target inertial vector.

RT_MAG = MAG (RT_M50_PROX)

VTAN = VT_M50_PROX - UNIT (RT_M50_PROX) (RT_M50_PROX·VT_M50_PROX)/RT_MAG

OMEGA_PROX = MAG (VTAN)/RT_MAG

(4) Determine whether a Lambert or a CW calculation will be performed and call the appropriate targeting supervisory logic.

If PROX_TGT_SET_NO$\leq$NLAMB, call PROX_TGT_SUP_LAMB.
Otherwise, call PROX_TGT_SUP

(5) Call the prox ops targeting output display load (sec. 4.12) to transfer the computed output data to the display buffers. The inputs and outputs are listed in tables 10 and 11.

(6) Set the status flags to off.

PROX_T1_STAR_STATUS = OFF

PROX_T2_STAR_STATUS = OFF

(7) Call the prox ops targeting start timer task (sec. 4.8) and compute the time to the upcoming maneuver and place it in the time management buffer. The inputs and outputs are listed in tables 12 and 13.

(8) Transfer the manuever execution data to the guidance buffer or array. This transfer is performed by the proximity operations guidance quantity transfer task (sec. 4.5). The inputs and outputs for the task are listed in tables 14 and 15.

4.1.2 Interface Requirements

The input and output requirements for the proximity operations targeting executive task are given in tables 16 and 17.

4.1.3 Processing Requirements

The specialist function will be called on crew demand and will be cycled thereafter at a TBD rate until the crew terminates the specialist function. The display data will be in the main memory so that when the specialist function is recalled the display will contain the data present when the display was last deactivated.

4.1.4 Initialization

The keyboard status flags will be initially set by the orbit targeting DIP I-load values. The output data in the upper portion of the display will be blank until an initial compute is performed. The input data (items 2 through 20) will be blank initially. Initialization of items 2 through 20 will occur from execution of items 26 and 27 or 28. The initialization of items 21 through 24 will be by I-load values.

4.1.5 Supplemental Information

None.

4.2 PROXIMITY OPERATIONS TARGETING STATUS TASK (PROX_STAT)

This task sets the maneuver status flag if the maneuver ignition time is prior to current time.

4.2.1 Detailed Requirements

The following steps are required to perform the proximity operations targeting status task.

a. Perform a logic test to see if current time (T_CURRENT) from FCOS is GMT or MET.

   (1) If GMT (TM_IND = OFF), compute a MET current time.

   PROX_T_CURRENT = T_CURRENT-BASE_MET

   (2) If T_CURRENT is MET (TM_IND = ON), compute a MET current time.

   PROX_T_CURRENT = T_CURRENT

b. Perform a logical test to determine if the maneuver exists and if it is in the past.

   If it exists and if it is in the past (T_MAN > 0 and PROX_T_CURRENT > T MAN), set the past status flag to on.

   PROX_PAST_STATUS = ON

   Otherwise, set it to off.

   PROX_PAST_STATUS = OFF

4.2.2 Interface Requirements

The input and output parameters for this task are listed in tables 6 and 7.

4.2.3 Processing Requirements

Perform once on call.

4.2.4 Initialization

None.

ORIGINAL PAGE IS
OF POOR QUALITY

4.2.5 Supplemental Information

None.

4.3 PROXIMITY OPERATIONS TARGETING TARGET SET SELECT TASK (PROX_TGT_SEL)

This task loads the inputs into the display and computational buffers from the selected set of I-load values.

4.3.1 Detailed Requirements

The following steps are required to perform the proximity operations targeting target set select task. Perform a logic test to retrieve the selected target set data from the I-load arrays.

If the array index equals the target set number (I_INDEX = PROX_TGT_SET_NO), load that value of the array to define the desired set.

T1_ILOAD_ARRAY (I_INDEX)
DT_ILOAD_ARRAY (I_INDEX)
EL_ILOAD_ARRAY (I_INDEX)
XOFF_ILOAD_ARRAY (I_INDEX)
YOFF_ILOAD_ARRAY (I_INDEX)
ZOFF_ILOAD_ARRAY (I_INDEX)

} = the desired set in the I-load arrays

a. Load the computational data buffers where:

   I_INDEX = PROX_TGT_SET_NO

   T1_TIG = PROX_BASE_TIME + 60 T1_ILOAD_ARRAY (I_INDEX)

   COMP_PROX_DT = DT_ILOAD_ARRAY (I_INDEX)

   EL_ANG = EL_ILOAD_ARRAY (I_INDEX)

   COMP_T2_XOFF = XOFF_ILOAD_ARRAY (I_INDEX)

   COMP_T2_YOFF = YOFF_ILOAD_ARRAY (I_INDEX)

   COMP_T2_ZOFF = ZOFF_ILOAD_ARRAY (I_INDEX)

b. Load the display buffers for T1 relative position and T2 time with blanks.

   DISP_T2_DAY = blank

DISP_T2_HR = blank

DISP_T2_MIN = blank

DISP_T2_SEC = blank

DISP_T1_X = blanks

DISP_T1_XD = blanks

c. Convert the computational T1 maneuver time (which is in MET) to days, hours, minutes, and seconds by calling the time conversion task (TIME_CONVRT). TIME_CONVRT detailed requirements are presented in section 13.

   TIME_CONVERT_FLAG = 0

   TIME_SEC = T1_TIG

   Call TIME_CONVRT; inputs: TIME_CONVERT_FLAG, TIME_SEC

   outputs: DAY, HR, MIN, SEC

d. Load the display data buffers where I_INDEX = PROX_TGT_SET_NO.

   DISP_T1_DAY = DAY

   DISP_T1_HR = HR

   DISP_T1_MIN = MIN

   DISP_T1_SEC = SEC

   DISP_PROX_DT = DT_ILOAD_ARRAY (I_INDEX)

   DISP_EL_ANG = (180/PI) EL_ILOAD_ARRAY (I_INDEX)

   DISP_T2_XOFF = XOFF_ILOAD_ARRAY (I_INDEX)

   DISP_T2_YOFF = YOFF_ILOAD_ARRAY (I_INDEX)

   DISP_T2_ZOFF = ZOFF_ILOAD_ARRAY (I_INDEX)

4.3.2 Interface Requirements

The input and output parameters for the proximity operations targeting target set select task are listed in tables 18 and 19.

4.3.3 Processing Requirements

Perform once on call.

4.3.4 Initialization

None.

4.3.5 Supplemental Information

None.

4.4 PROXIMITY OPERATIONS TARGETING INITIALIZATION TASK (PROX_INIT)

This task performs conversions on the input data display buffer items and stores the results in computational data buffers and I-load data buffers.

4.4.1 Detailed Requirements

The following steps are required to peform the proximity operations targeting initialization tasks.

a. Did crew request a change to the base time (PROX_ITEM_21TO24_STATUS = ON) ? If so, put the displayed base time into the computational buffer:

   If PROX_ITEM_21TO24_STATUS = ON and PROX_ITEM_2TO20_STATUS = OFF, then set:

   TIME_CONVERT_FLAG = 1

   DAY = PROX_BASE_DAY

   HR = PROX_BASE_HR

   MIN = PROX_BASE_MIN

   SEC = PROX_BASE_SEC

   Call TIME_CONVRT; inputs: TIME_CONVERT_FLAG

   DAY, HR, MIN, SEC

   output: TIME_SEC

   PROX_BASE_TIME = TIME_SEC.

b. If PROX_ITEM_2TO20_STATUS is ON and PROX_ITEM_21TO24_STATUS is OFF:

   (1) Load the offset position and delta-t computation buffers and I-load buffers. The I-load buffers index value is determined by the selected target set (I_INDEX = PROX_TGT_SET_NO).

COMP_T2_XOFF = DISP_T2_XOFF

COMP_T2_YOFF = DISP_T2_YOFF

COMP_T2_ZOFF = DISP_T2_ZOFF

COMP_PROX_DT_ = DISP_PROX_DT

EL_ANG = DISP_EL_ANG PI/180

XOFF_ILOAD_ARRAY (I_INDEX) = DISP_T2_XOFF

YOFF_ILOAD_ARRAY (I_INDEX) = DISP_T2_YOFF

ZOFF_ILOAD_ARRAY (I_INDEX) = DISP_T2_ZOFF

DT_ILOAD_ARRAY (I_INDEX) = DISP_PROX_DT

EL_ILOAD_ARRAY (I_INDEX) = EL_ANG

(2) Load the computation relative position and velocity buffers.

COMP_$\overline{X}$ = DISP_T1_$\overline{X}$

COMP_$\overline{XD}$ = DISP_T1_$\overline{XD}$

(3) Convert the display T1 time and display T2 time and load the computation and I-load buffers.

(a) Call the time conversion task (TIME_CONVRT) to convert display T1 time to total seconds. TIME_CONVRT detailed requirements are presented in section 4.13.

TIME_CONVERT_FLAG = 1

DAY = DISP_T1_DAY

HR = DISP_T1_HR

MIN = DISP_T1_MIN

SEC = DISP_T1_SEC

Call TIME_CONVRT; inputs: TIME_CONVERT_FLAG,DAY, HR, MIN, SEC

output: TIME_SEC

(b) Load the T1 time in the computation and I-load buffers. Set TIME_ PROX.

T1_TIG = TIME_SEC

T1_1LOAD_ARRAY (I_INDEX) = (TIME_SEC-PROX_BASE_TIME)/60.

TIME_PROX = T1_TIG

I_INDEX = PROX_TGT_SET_NO

(c) Call the time conversion task (TIME_CONVRT) to convert display T2 time to total seconds.

DAY = DISP_T2_DAY

HR = DISP_T2_HR

MIN = DISP_T2_MIN

SEC = DISP_T2_SEC

Call TIME_CONVRT; inputs: TIME_CONVERT_FLAG, DAY, HR, MIN, SEC

output: TIME_SEC

(d) Load the T2 time in the computation array.

T2_TIG = TIME_SEC

4.4.2 Interface Requirements

The input and output parameters for the proximity operations targeting initialization tasks are listed in tables 8 and 9.

4.4.3 Processing Requirements

Perform once on call.

4.4.4 Initialization

None.

4.4.5 Supplemental Information

None.

4.5 PROXIMITY OPERATIONS TARGETING GUIDANCE QUANTITY TRANSFER TASK (PROX_TRANS)

This task sets the quantities required to perform the maneuver.

The delta-v maneuver (called DELTA_V_LVLH_GUID) is defined in terms of the Orbiter-centered local vertical local horizontal (LVLH) coordinate system for Lambert maneuvers (which have GUID_FLAG = 1) and is defined in terms of the target-centered LVLH curvilinear coordinate system, otherwise.

4.5.1 Detailed Requirements

The following steps are required to perform the proximity operations targeting guidance quantity transfer task.

a. Perform a logical test to see if the maneuver has been executed. If it has not been executed (PROX_PAST_STATUS = OFF), then:

   (1) Set TIG_GUID_DAY = DISP_TMAN_DAY

   TIG_GUID_HR = DISP_TMAN_HR

   TIG_GUID_MIN = DISP_TMAN_MIN

   TIG_GUID_SEC = DISP_TMAN_SEC

   (2) Set DELTA_V_LVLH_GUID = DV_LVLH

   (3) Test to see if the maneuver is a Lambert maneuver. If so (GUID_FLAG = 1), set:

   S_ROTATE_GUID = S_ROTATE

   R_OFFSET_GUID = R_OFFSET

   T_OFFSET_GUID = T_OFFSET

4.5.2 Interface Requirements

The input and output parameters for the proximity operations targeting guidance quantity transfer task are given in tables 14 and 15.

4.5.3 Processing Requirements

Perform once on call.

4.5.4 Initialization

None.

4.5.5 Supplemental Information

None.

4.6 PROXIMITY OPERATIONS TARGETING SUPERVISORY LOGIC TASK (PROX_TGT_SUP)

This task is scheduled and executed following a crew execution of item 27 or item 28 for a non-Lambert targeted maneuver. This task is the top-level supervisory module for performing the subtask required to compute the maneuver delta-velocity (delta-v) vector using closed form CW equations. A functional flow of PROX_TGT_SUP is shown in figure 6.

4.6.1 Detailed Requirements

The following steps are required to perform the targeting supervisory logic.

a. Is the USE_DISP_REL_STATE flag off? If so, use navigated target and Orbiter states and compute the LVLH curvilinear relative state:

   R_T_INER = RT_M50_PROX

   V_T_INER = VT_M50_PROX

   R_S_INER = RS_M50_PROX

   V_S_INER = VS_M50_PROX

   INER_TO_LVC = ON

   Call REL_COMP; inputs: R_T_INER, V_T_INER
   R_S_INER, V_S_INER
   INER_TO_LVC, OMEGA_PROX
   outputs: R_REL, V_REL

   COMP_X = R_REL

   COMP_XD = V_REL

b. Compute the minumum time to ignition:

   TIG_MIN = PROX_T_CURRENT + PROX_DTMIN

c. Did the crew request the compute T1 function (PROX_T1_STAR_STATUS = ON)?

(1) If not, compute the T2 maneuver:

(a) If TIG_MIN > T2_TIG, set T2_TIG = TIG_MIN

(b) Update the present relative state to T2_TIG:

$\vec{X}$ = COMP_$\vec{X}$

$\vec{XD}$ = COMP_$\vec{XD}$

DTIME = T2_TIG-TIME_PROX

Call REL_PRED; inputs: $\vec{X}$, $\vec{XD}$, DTIME

outputs: $\vec{X2}$, $\vec{XD2}$

COMP_$\vec{X}$ = $\vec{X2}$

COMP_$\vec{XD}$ = $\vec{XD2}$

TIME_PROX = TIME_PROX + DTIME

(c) Null the relative rate.

$\vec{DV}$_LVLH = -COMP_$\vec{XD}$

T_MAN = T2_TIG

(2) Otherwise, compute the T1 maneuver:

(a) Compute the transfer time and T2_TIG.

Call DT_COMP; inputs: T1_TIG, T2_TIG, COMP_PROX_DT

outputs: T2_TIG, COMP_PROX_DT

USE_OMEGA_DT

(b) If TIG_MIN > T1_TIG, then:

(i) Set T1_TIG = TIG_MIN

(ii) If USE_OMEGA_DT is off, set

COMP_PROX_DT = (T2_TIG -T1_TIG)/60.

(iii) Set USE_DISP_REL_STATE = OFF

(c) If USE_DISP_REL_STATE is off, then:

(i) Update the present relative state to T1_TIG:

$\overline{X}$ = COMP_$\overline{X}$

$\overline{XD}$ = COMP_$\overline{XD}$

DTIME = T1_TIG_TIME_PROX

Call REL_PRED; inputs: $\overline{X}$, $\overline{XD}$, DTIME

output: $\overline{X2}$ = $\overline{XD2}$

COMP_$\overline{X}$ = $\overline{X2}$

COMP_$\overline{XD}$ = $\overline{XD2}$

(ii) Display the expected state at T1_TIG.

DISP_T1_$\overline{X}$ = COMP_$\overline{X}$

DISP_T1_$\overline{XD}$ = COMP_$\overline{XD}$

(d) If USE_OMEGA_DT is on:

(i) Use the OMEGA_DT_COMP routine to compute the transfer time.

Call OMEGA_DT_COMP;

inputs:
- COMP_$\overline{X}$
- COMP_T2_XOFF
- COMP_T2_YOFF
- COMP_T2_ZOFF
- XOFF_ILOAD_ARRAY
- YOFF_ILOAD_ARRAY
- ZOFF_ILOAD_ARRAY
- DT_ILOAD_ARRAY
- OMEGA_PROX

output: COMP_PROX_DT

(ii) Compute T2_TIG.

T2_TIG = T1_TIG + 60 COMP_PROX_DT

(e) Compute the T1 maneuver.

DT_OFFTGT = COMP_PROX_DT

$\overline{X}$_OFFTGT = COMP_$\overline{X}$

$\overline{XD}$_OFFTGT = COMP_$\overline{XD}$

$$\overline{X2}\text{\_OFFTGT} = \begin{cases} \text{COMP\_T2\_XOFF} \\ \text{COMP\_T2\_YOFF} \\ \text{COMP\_T2\_ZOFF} \end{cases}$$

Call OFFSET_TGT inputs: DT_OFFTGT

$\overline{X}$_OFFTGT

$\overline{XD}$_OFFTGT

$\overline{X2}$_OFFTGT

output: $\overline{DV}$

$\overline{DV}$_LVLH = $\overline{DV}$

T_MAN = T1_TIG

d. Set the guidance flag (GUID_FLAG) to zero.

4.6.2 Interface Requirements

The input and output parameters for the proximity operations targeting supervisory logic task are listed in tables 20 and 21.

4.6.3 Processing Requirements

Perform once on call.

4.6.4 Initialization

None.

4.6.5 Supplemental Information

None.

4.7 PROXIMITY OPERATIONS TARGETING SUPERVISORY LAMBERT LOGIC TASK (PROX_TGT_SUP_LAMB)

This task is scheduled and executed following a crew execution of item 27 or item 28 if Lambert targeting is requested (PROX_TGT_SET_NO $\leq$ NLAMB). This task is the top level supervisory module for performing the subtask required to compute the maneuver delta-v vector with Lambert equations. A functional flow of PROX_TGT_SUP_LAMB is shown in figure 7.

4.7.1 Detailed Requirements

The following steps are required to compute maneuvers.

a. Compute minimum time to ignition.

TIG_MIN = PROX_T_CURRENT + PROX_DTMIN_LAMB

b. Check compute T1 flag:

(1) If PROX_T1_STAR_STATUS = OFF, compute the T2 maneuver.

(a) If T2_TIG < TIG_MIN, set T2_TIG = TIG_MIN

(b) Update Shuttle and target inertial states from the present time to T2_TIG.

S_OPTION = 1

$\overline{\text{R\_IN}}$ = $\overline{\text{RS\_M50\_PROX}}$

$\overline{\text{V\_IN}}$ = $\overline{\text{VS\_M50\_PROX}}$

T_IN = TIME_PROX + BASE_MET

T_OUT = T2_TIG + BASE_MET

Call UPDATVP; inputs: S_OPTION, $\overline{\text{R\_IN}}$, $\overline{\text{V\_IN}}$, T_IN, T_OUT

outputs: $\overline{\text{R\_OUT}}$, $\overline{\text{V\_OUT}}$

$\overline{\text{R\_S\_INER}}$ = $\overline{\text{R\_OUT}}$; $\overline{\text{V\_S\_INER}}$ = $\overline{\text{V\_OUT}}$

S_OPTION = 2

$\overline{\text{R\_IN}}$ = $\overline{\text{RT\_M50\_PROX}}$

$\overline{\text{V\_IN}}$ = $\overline{\text{VT\_M50\_PROX}}$

Call UPDATVP:

$\overline{R}$_T_INER = $\overline{R}$_OUT

$\overline{V}$_T_INER = $\overline{V}$_OUT

(c) Convert to a target-centered curvilinear state.

INER_TO_LVC = ON

Call REL_COMP; inputs: $\overline{R}$_T_INER, $\overline{V}$_T_INER

$\overline{R}$_S_INER, $\overline{V}$_S_INER

INER_TO_LVC

outputs: $\overline{R}$_REL, $\overline{V}$_REL

(d) Null relative velocity. Determine maneuver time. Set guidance flag for external DV.

$\overline{DV}$_LVLH = -$\overline{V}$_REL

T_MAN = T2_TIG

GUID_FLAG = 0.

(2) If PROX_T1_STAR_STATUS = ON, compute the T1 maneuver.

(a) If elevation angle (EL_ANG) is ≠ 0, then determine T1 time from elevation angle search:

(i) Compute time of elevation angle.

Call TELEV; inputs: $\overline{RS}$_M50_PROX

$\overline{VS}$_M50_PROX

$\overline{RT}$_M50_PROX

$\overline{VT}$_M50_PROX

TIME_PROX,

EL_ANG,

PI

EL_TOL, EL_DH_TOL

outputs: TTPI,
$\overline{\text{RS\_OUT}}$, $\overline{\text{VS\_OUT}}$
$\overline{\text{RT\_OUT}}$, $\overline{\text{VT\_OUT}}$
TS_OUT, TT_OUT

T1_TIG = TTPI

(ii) Find relative state.

$\overline{\text{R\_S\_INER}}$ = $\overline{\text{RS\_OUT}}$

$\overline{\text{V\_S\_INER}}$ = $\overline{\text{VS\_OUT}}$

$\overline{\text{R\_T\_INER}}$ = $\overline{\text{RT\_OUT}}$

$\overline{\text{V\_T\_INER}}$ = $\overline{\text{VT\_OUT}}$

INER_TO_LVC = ON

Call REL_COMP;

COMP_$\overline{\text{X}}$ = $\overline{\text{R}}$_REL

COMP_$\overline{\text{XD}}$ = $\overline{\text{V}}$_REL

(iii) Display relative state.

DISP_T1_$\overline{\text{X}}$ = $\overline{\text{R}}$_REL

DISP_T1_$\overline{\text{XD}}$ = $\overline{\text{V}}$_REL

(iv) Change base time in display and computational buffer; change I-load.

T1_ILOAD_ARRAY (PROX_TGT_SET_NO) = 0.

PROX_BASE_TIME = T1_TIG

TIME_CONVERT_FLAG = 0

TIME_SEC = T1_TIG

Call TIME_CONVRT; inputs: TIME_CONVERT_FLAG
TIME_SEC

outputs: DAY, HR, MIN, SEC

PROX_BASE_DAY = DAY

PROX_BASE_HR = HR

PROX_BASE_MIN = MIN

PROX_BASE_SEC = SEC

(b) If elevation angle (EL_ANG) = 0, use the T1 time given for the maneuver time.

(i) Given T1_TIG, compute T2_TIG and/or COMP_PROX_DT.

Call DT_COMP; inputs: T1_TIG, T2_TIG, COMP_PROX_DT

outputs: T2_TIG, COMP_PROX_DT

USE_OMEGA_DT

(ii) Compare to the minimum time of ignition (TIG_MIN). If T1_TIG < TIG_MIN:

(aa) Set T1_TIG = TIG_MIN.

(bb) Check the USE_OMEGA_DT. If it is off, then the maneuver transfer time will be adjusted so that the next maneuver will be on time.

COMP_PROX_DT = (T2_TIG-T1_TIG)/60

(cc) Set the USE_DISP_REL_STATE flag to off.

(iii) Update the target state to T1_TIG:

S_OPTION = 2

R_IN = RT_M50_PROX

V_IN = VT_M50_PROX

T_IN = TIME_PROX + BASE_MET

T_OUT = T1_TIG + BASE_MET

Call UPDATVP; inputs: R_IN, V_IN, T_IN, T_OUT,

S_OPTION

outputs: R_OUT, V_OUT

ORIGINAL PAGE IS
OF POOR QUALITY

R_T_INER = R_OUT

V_T_INER = V_OUT

RT_T1TIG = R_OUT

VT_T1TIG = V_OUT

(iv) Check the USE_DISP_REL_STATE flag.

(aa) If USE_DISP_REL_STATE is on, compute the Shuttle inertial state at T1.

INER_TO_LVC = OFF

R_REL = COMP_X

V_REL = COMP_XD

Call REL_COMP inputs: R_REL, V_REL, INER_TO_LVC

R_T_INER, V_T_INER

outputs: R_S_INER

V_S_INER

RS_T1TIG = R_S_INER

VS_T1TIG = V_S_INER

(bb) If USE_DISP_REL_STATE is off, update the present Shuttle states to T1_TIG, convert to curvilinear coordinates and display:

(aaa) Update the Shuttle inertial state.

S_OPTION = 1

R_IN = RS_M50_PROX

V_IN = VS_M50_PROX

Call UPDATVP

RS_T1TIG = R_OUT

VS_T1TIG = V_OUT

(bbb) Convert to LVLH curvilinear coodinates.

$\dot{\text{I}}$NER_TO_LVC = ON

Call REL_COMP;

inputs: $\bar{\text{R}}$_T_INER, $\bar{\text{V}}$_T_INER

$\bar{\text{R}}$_S_INER, $\bar{\text{V}}$_S_INER

I$\dot{\text{N}}$ER_TO_LVC

outputs: $\bar{\text{R}}$_REL, $\bar{\text{V}}$_REL

(ccc) Load the computation buffers at T1_TIG for possible use in OMEGA_DT_COMP later.

COMP_$\bar{\text{X}}$ = $\bar{\text{R}}$_REL

COMP_$\bar{\text{X}}$D = $\bar{\text{V}}$_REL

(ddd) Load the display buffers at T1_TIG.

DISP_T1_$\bar{\text{X}}$ = $\bar{\text{R}}$_REL

DISP_T1_$\bar{\text{X}}$D = $\bar{\text{V}}$_REL

(c) Compute T2_TIG and COMP_PROX_DT from the T1_TIG being used.

Call DT_COMP; inputs: T1_TIG, T2_TIG, COMP_PROX_DT

outputs: T2_TIG, COMP_PROX_DT,

USE_$\dot{\text{O}}$MEGA_DT

(d) Check the USE_$\dot{\text{O}}$MEGA_DT flag. If it is on, there is insufficient information to determine T2_TIG and COMP_PROX_DT. Compute this information instead by use of the "$_\omega$t-calculation."

(i) Compute the transfer time.

Call OMEGA_DT_COMP.

inputs: COMP_$\bar{\text{X}}$

COMP_T2_XOFF

COMP_T2_YOFF

COMP_T2_ZOFF

XOFF_ILOAD_ARRAY (PROX_TGT_SET_NO + 1)

YOFF_ILOAD_ARRAY (PROX_TGT_SET_NO + 1)

ZOFF_ILOAD_ARRAY (PROX_TGT_SET_NO + 1)

DT_ILOAD_ARRAY (PROX_TGT_SET_NO + 1)

OMEGA_PROX

output: COMP_PROX_DT

(ii) Compute T2_TIG:

T2_TIG = T1_TIG + 60 COMP_PROX_DT

(e) Update the target inertial state from T1_TIG.

S_OPTION = 2

$\overline{\text{R\_IN}}$ = $\overline{\text{R\_T\_INER}}$

$\overline{\text{V\_IN}}$ = $\overline{\text{V\_T\_INER}}$

T_IN = T1_TIG + BASE_MET

T_OUT = T2_TIG + BASE_MET

Call UPDATVP; inputs: $\overline{\text{R\_IN}}$, $\overline{\text{V\_IN}}$, T_IN, T_OUT

outputs: $\overline{\text{R\_OUT}}$, $\overline{\text{V\_OUT}}$

$\overline{\text{R\_T\_INER}}$ = $\overline{\text{R\_OUT}}$

$\overline{\text{V\_T\_INER}}$ = $\overline{\text{V\_OUT}}$

(f) Compute the Shuttle inertial state at T2.

$$\overline{\text{R\_REL}} = \begin{cases} \text{COMP\_T2\_XOFF} \\ \text{COMP\_T2\_YOFF} \\ \text{COMP\_T2\_ZOFF} \end{cases}$$

$\dot{\text{I}}$NER_TO_LVC = OFF

Call REL_COMP; inputs: $\overline{\text{R\_T\_INER}}$, $\overline{\text{V\_T\_INER}}$, $\overline{\text{R\_REL}}$

$\dot{\text{I}}$NER_TO_LVC

output: $\overline{\text{R\_S\_INER}}$

$\overline{RS\_T2TIG} = \overline{R\_S\_INER}$

(g) Do the Lambert problem using the precision velocity required routine.

Call PREVR; inputs: T1_TIG, T2_TIG, $\overline{RS\_T1TIG}$, $\overline{VS\_T1TIG}$, $\overline{RS\_T2TIG}$

output: $\overline{VS\_REQUIRED}$

$\overline{RS\_IPO}$

$\overline{R\_OFFSET}$, T_OFFSET

S_ROTATE

(h) Compute the LVLH Shuttle-centered relative state at T1 before the maneuver.

$\overline{RS} = \overline{RS\_T1TIG}$

$\overline{VS} = \overline{VS\_T1TIG}$

$\overline{RT} = \overline{RT\_T1TIG}$

$\overline{VT} = \overline{VT\_T1TIG}$

Call ORBLV; inputs: $\overline{RS}$, $\overline{VS}$, $\overline{RT}$, $\overline{VT}$

output: $\overline{VSLV}$

$\overline{V\_LVLH} = \overline{VSLV}$

(i) Compute the LVLH Shuttle-centered relative state at T1 after the impulsive maneuver.

$\overline{VS} = \overline{VS\_REQUIRED}$

Call ORBLV; inputs: $\overline{RS}$, $\overline{VS}$, $\overline{RT}$, $\overline{VT}$

output: $\overline{VSLV}$

(j) Compute the LVLH Shuttle-centered maneuver required at T1.

$\overline{DV\_LVLH} = \overline{VSLV} - \overline{V\_LVLH}$

T_MAN = T1_TIG

GUID_FLAG = 1

4.7.2 Interface Requirements

The input and output parameters for the proximity operations targeting supervisory Lambert logic task are listed in tables 22 and 23.

4.7.3 Processing Requirements

Perform once per call.

4.7.4 Initialization

None.

4.7.5 Supplemental Information

None.

4.8 PROXIMITY OPERATIONS TARGETING START TIMER TASK (PROX_STIME)

This task sets the display time to count down to.

4.8.1 Detailed Requirements

The following steps are required to perform this task.

a. Perform a test to see if the maneuver has been executed (PROX_PAST_STATUS = ON).

   (1) If PROX_PAST_STATUS = ON, set CRT_TIME = OFF.

   (2) If PROX_PAST_STATUS = OFF, calculate the delta-t between the current time and the maneuver time and set the cathode-ray tube (CRT) timer status flags:

      (a) If current time is GMT (TM_IND = OFF), set

         DT_ST_TIMER = T_MAN + BASE_MET-T_CURRENT

      (b) If current time is MET (TM_IND = ON), set

         DT_ST_TIMER = T_MAN-T_CURRENT

      (c) Set the CRT timer status flags:

         CRT_TIME = ON

         CRT_UPDWN = ON

4.8.2 Interface Requirements

The input and output parameters for the proximity operations targeting start time task are listed in tables 12 and 13.

4.8.3 Processing Requirements

Perform once on call.

4.8.4 Initialization

None.

4.8.5 Supplemental Information

None.

4.9 MANEUVER TO OFFSET TARGETING TASK (OFFSET_TGT)

This task computes a LVLH maneuver delta-v vector to achieve the desired offset position in the specified delta-t.

4.9.1 Detailed Requirements

The following steps are required to perform the maneuver to offset targeting task.

a. Compute the T1 maneuver delta-v vector:

W = OMEGA_PROX

T = 60 DT_OFFTGT

S = SIN (W T)

C = COS (W T)

C1 = 1-C

K = W/(8-8C-3S W T)

$$\mathrm{MAT} = \begin{bmatrix} -KS & 0 & K(14C1-6WTS) & KS & 0 & -2KC1 \\ 0 & -CW/S & 0 & 0 & W/S & 0 \\ -2KC1 & 0 & K(3CWT-4S) & 2KC1 & 0 & K(4S-3WT) \end{bmatrix}$$

ORIGINAL PAGE IS OF POOR QUALITY

$\overline{STATE}_{1,2,3} = \overline{X\_OFFTGT}$

$\overline{STATE}_{4,5,6} = \overline{X2\_OFFTGT}$

$\overline{V\_T1\_NEED} = MAT\ \overline{STATE}$

$\overline{DV} = \overline{V\_T1\_NEED} - \overline{XD\_OFFTGT}$

4.9.2 Interface Requirements

The input and output parameters for the maneuver to offset targeting task are listed in tables 24 and 25.

4.9.3 Processing Requirements

Perform once on call.

4.9.4 Initialization

None.

4.9.5 Supplemental Information

None.

4.10 RELATIVE STATE PREDICTOR TASK (REL_PRED)

This task propagates a target-centered LVLH relative rotating curvilinear vector. The algorithm is closed form and utilizes several simplifying assumptions.

4.10.1 Detailed Requirements

The following are the steps required to perform the relative state predictor tasks.

a. Find transformation matrix to predict the new state at time t_2 from the state at t_1.

W = OMEGA_PROX

T = DTIME

S = SIN (W T)

$C = COS\ (W\ T)$

$C1 = 1-C$

$$MAT\_PRED = \begin{bmatrix} 1 & 0 & 6WT-6S & 4S/W-3T & 0 & 2\ C1/W \\ 0 & C & 0 & 0 & S/W & 0 \\ 0 & 0 & 4-3C & -2\ C1/W & 0 & 2/W \\ 0 & 0 & 6W\ C1 & 4C-3 & 0 & 2S \\ 0 & -WS & 0 & 0 & C & 0 \\ 0 & 0 & 3\ WS & -2S & 0 & C \end{bmatrix}$$

b. Use predictor matrix to get new relative state at T_2.

$\overline{STATE1}_{1,2,3} = \overline{X}$

$\overline{STATE1}_{4,5,6} = \overline{XD}$

$\overline{STATE2} = MAT\_PRED\ \overline{STATE1}$

$\overline{X2} = \overline{STATE2}_{1,2,3}$

$\overline{XD2} = \overline{STATE2}_{4,5,6}$

4.10.2 Interface Requirements

The input and output parameters for the relative state predictor task are listed in tables 26 and 27.

4.10.3 Processing Requirements

Perform once on call.

4.10.4 Initialization

None.

4.10.5 Supplemental Information

None.

4.11 RELATIVE STATE COMPUTE TASK (REL_COMP)

This task computes the state of the Orbiter in a target-centered local vertical rotating curvilinear (LVC) coordinate system using the Earth-centered inertial (M50) states of the Orbiter and the target, or performs the reverse transformation, depending on an input flag.

4.11.1 Detailed Requirements

The following steps are required to perform this task.

a. Find RT_MAG: RT_MAG = mag($\overline{\text{R\_T\_INER}}$)

b. Compute the transformation matrix from M50 inertial frame to the local vertical inertial rectangular (LVIR) coordinate frame:

$$\overline{\text{MAT\_M50\_LVIR}} = \begin{bmatrix} \text{unit } ((\overline{\text{R\_T\_INER}} \text{ X } \text{V\_T\_INER}) \text{ X } \text{R\_T\_INER})^T \\ -\text{unit } (\overline{\text{R\_T\_INER}} \text{ X } \overline{\text{V\_T\_INER}})^T \\ -\text{unit } (\overline{\text{R\_T\_INER}})^T \end{bmatrix}$$

c. Determine the orbital angular rate vector in LVLH coodinates:

$$\overline{\text{OMEGA\_LV\_PROX}} = \text{OMEGA\_PROX} \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}$$

d. Determine if the transformation to be done is from inertial M50 coordinates to the LV curvilinear target-centered coordinates (LVC), i.e. $\dot{\text{I}}$NER_TO_LVC = ON.

(1) If $\dot{\text{I}}$NER_TO_LVC = ON:

(a) Compute the relative M50 state of the Orbiter with respect to the target.

$$\overline{\text{RTS\_M50}} = \overline{\text{R\_S\_INER}} - \overline{\text{R\_T\_INER}}$$

$$\overline{\text{VTS\_M50}} = \overline{\text{V\_S\_INER}} - \overline{\text{V\_T\_INER}}$$

(b) Convert these states to a target-centered LVIR coordinate frame:

$$\overline{\text{RTS\_LVIR}} = \text{MAT\_M50\_LVIR} \ \ \overline{\text{RTS\_M50}}$$

$$\overline{\text{VTS\_LVIR}} = \text{MAT\_M50\_LVIR} \ \ \overline{\text{VTS\_M50}}$$

(c) Convert the LVIR relative states to a target-centered LV rotating rectangular coordinate frame.

$$\overline{\text{RTS\_LV}} = \overline{\text{RTS\_LVIR}}$$

$$\overline{\text{VTS\_LV}} = \overline{\text{VTS\_LVIR}} - (\overline{\text{OMEGA\_LV\_PROX\_}} \text{ X } \overline{\text{RTS\_LVIR}})$$

(d) Convert the LV relative states to a target-centered local vertical rotating curvilinear coordinate frame.

$$\text{ZCON} = \overline{\text{RT\_MAG}} - \text{RTS\_LV}_3$$

$$THETA = ATAN\ (\overline{RTS\_LV}_1/ZCON)$$

$$\overline{R\_REL} = \begin{bmatrix} RT\_MAG\ THETA \\ \overline{RTS\_LV}_2 \\ RT\_MAG-ZCON/COS(THETA) \end{bmatrix}$$

$$\overline{THETA\_DOT} = COS^2\ (THETA)(\overline{VTS\_LV}_1\ ZCON + \overline{RTS\_LV}_1\ \overline{VTS\_LV}_3)/(ZCON)^2$$

$$\overline{V\_REL} = \begin{bmatrix} RT\_MAG\ THETA\_DOT \\ \overline{VTS\_LV}_2 \\ (\overline{VTS\_LV}_3 - ZCON\ THETA\_DOT\ TAN(THETA))\ /COS(THETA) \end{bmatrix}$$

(2) If INER_TO_LVC = OFF, convert from target-centered curvilinear coordinates to M50 inertial coordinates.

(a) Compute internal variables.

$$THETA = \overline{R\_REL}_1/\ RT\_MAG$$

$$THETA\_DOT = V\_REL_1\ /\ RT\_MAG$$

$$ZCON = RT\_MAG - R\_REL_3$$

(b) Convert to LV target-centered rotating rectangular coordinates.

$$\overline{RTS\_LV} = \begin{pmatrix} ZCON\ \ SIN(THETA) \\ \overline{R}\ REL_2 \\ \overline{RT}\_MAG - ZCON\ COS\ (THETA) \end{pmatrix}$$

$$\overline{VTS\_LV} = \begin{pmatrix} \begin{bmatrix} ZCON\ THETA\_DOT\ \ COS\ (THETA) \\ -\overline{V}\ REL_3\ \ SIN\ (THETA) \end{bmatrix} \\ \overline{V}\ REL_2 \\ \overline{RTS\_LV}_1\ THETA\_DOT + \overline{V}\_REL_3\ COS\ (THETA) \end{pmatrix}$$

(c) Convert from the LV frame to a target-centered LVIR frame.

$$\overline{RTS\_LVIR} = \overline{RTS\_LV}$$

$$\overline{VTS\_LVIR} = \overline{VTS\_LV} + \overline{OMEGA\_LV\_PROX}\ X\ \overline{RTS\_LVIR}$$

(d) Convert from LVIR coordinates to an inertial frame relative to the target.

$$\vec{RTS\_M50} = [MAT\_M50\_LVIR]^T \; \vec{RTS\_LVIR}$$

$$\vec{VTS\_M50} = [MAT\_M50\_LVIR]^T \; \vec{VTS\_LVIR}$$

(e) Compute the Shuttle inertial M50 coordinate.

$$\vec{R\_S\_INER} = \vec{R\_T\_INER} + \vec{RTS\_M50}$$

$$\vec{V\_S\_INER} = \vec{V\_T\_INER} + \vec{VTS\_M50}$$

4.11.2 Interface Requirements

The input and output parameters for the relative state compute task are given in tables 28 and 29.

4.11.3 Processing Requirements

Perform once on call.

4.11.4 Initialization

None.

4.11.5 Supplemental Information

None.

4.12 PROXIMITY OPERATIONS TARGETING OUTPUT DISPLAY LOAD TASK (PROX_DISP_LOAD)

This task loads the output display buffers with data from the computational buffers once a compute item 27 or item 28 is complete.

4.12.1 Detailed Requirements

The following steps are required to perform the proximity operations targeting output display load task.

a. Display the maneuver.

MAN_TGT = PROX_TGT_SET_NO

$$DISP\_\vec{DV} = \vec{DV\_LVLH}$$

DISP_DV_MAG = MAG (DV_LVLH)

b. Convert the T1_TIG and T2_TIG times to days, hours, minutes, and seconds using the time conversion task.

(1) TIME_SEC = T1_TIG

TIME_CONVERT_FLAG = 0

Call TIME_CONVRT; inputs: TIME_CONVERT_FLAG, TIME_SEC

outputs: DAY, HR, MIN, SEC

DISP_T1_DAY = DAY

DISP_T1_HR = HR

DISP_T1_MIN = MIN

DISP_T1_SEC = SEC

(2) TIME_SEC = T2_TIG

Call TIME_CONVRT; inputs: TIME_CONVERT_FLAG, TIME_SEC

outputs: DAY, HR, MIN, SEC

DISP_T2_DAY = DAY

DISP_T2_HR = HR

DISP_T2_MIN = MIN

DISP_T2_SEC = SEC

c. Display the transfer time.

DISP_PROX_DT = COMP_PROX_DT

d. Display the maneuver time.

(1) If T_MAN = T1_TIG, set:

DISP_TMAN_TIME = T1_TIG

DISP_TMAN_DAY = DISP_T1_DAY

DISP_TMAN_HR = DISP_T1_HR

DISP_TMAN_MIN = DISP_T1_MIN

DISP_TMAN_SEC = DISP_T1_SEC

(2) If T_MAN = T2_TIG, set:

DISP_TMAN_TIME = T2_TIG

DISP_TMAN_DAY = DISP_T2_DAY

DISP_TMAN_HR = DISP_T2_HR

DISP_TMAN_MIN = DISP_T2_MIN

DISP_TMAN_SEC = DISP_T2_SEC

4.12.2 Interface Requirements

The input and output parameters for the proximity operations targeting output display load task are listed in tables 10 and 11.

4.12.3 Processing Requirements

Perform once on call.

4.12.4 Initialization

None.

4.12.5 Supplemental Information

4.13 TIME CONVERSION TASK (TIME_CONVRT)

This task computes total time in seconds (given days, hours, minutes, and seconds and computes days, hours, minutes, and seconds (given total time in seconds).

4.13.1 Detailed Requirements

The following steps are required to perform the time conversion tasks.

a. Perform a logical test to determine if the time is to be computed in seconds or days, hours, minutes, and seconds.

(1) If total time is to be computed in seconds

(TIME_CONVERT_FLAG = 1), set

TIME_SEC = 86400 DAY + 3600 HR + 60 MIN + SEC

(2) If time is to be computed in days, hours, minutes, and seconds (TIME_CONVERT_FLAG = 0), set

DAY = truncate (TIME_SEC/86400)

HR = truncate (TIME_SEC - 86400 DAY)/3600

MIN = truncate (TIME_SEC - 86400 DAY - 3600 HR)/60

SEC = TIME_SEC - 86400 DAY - 3600 HR-60 MIN

4.13.2 Interface Requirements

The input and output parameters for the time conversion task are listed in tables 30 and 31.

4.13.3 Processing Requirements

Perform once per call.

4.13.4 Initialization

None.

4.13.5 Supplemental Information

None.

4.14 DELTA-T COMPUTE TASK (DT_COMP)

The delta-t compute task is called to determine the transfer time and/or the maneuver time of the second of two impulse maneuvers.

4.14.1 Detailed Requirements

The following steps are required to perform this task.

a. Set the USE_OMEGA_DT flag to off.

b. Check the transfer time (COMP_PROX_DT):

$$\text{If COMP\_PROX\_DT} \begin{cases} >0, & \text{set T2\_TIG = T1\_TIG + 60 COMP\_PROX\_DT} \\ <0, & \text{set USE\_OMEGA\_DT = ON} \\ = 0, & \text{set COMP\_PROX\_DT = (T2\_TIG - T1\_TIG)/60} \end{cases}$$

4.14.2 Interface Requirements

The input and output parameters for the delta-t compute task are given in tables 32 and 33.

4.14.3 Processing Requirements

Perform once on call.

4.14.4 Initialization

None.

4.14.5 Supplemental Information

None.

4.15 OMEGA-DT CALCULATION TASK (OMEGA_DT_COMP)

This task is called when the given transfer time and T2 TIG are zero. This task calculates the transfer time needed to go from the T1_TIG state to the T2_TIG state such that the T2_TIG maneuver is perpendicular to the line of sight and intercepts the next target point at the appropriate time.

4.15.1 Detailed Requirements

The following steps are required to perform the omega-dt calculation task.

a. Find the velocity required at the displayed T2 time to intercept the next target set.

DT_OFFTGT = DT_ILOAD_ARRAY (PROX_TGT_SET_NO + 1)

$$\overline{X}\_OFFTGT = \begin{pmatrix} COMP\_T2\_XOFF \\ COMP\_T2\_YOFF \\ COMP\_T2\_ZOFF \end{pmatrix}$$

$\overline{XD}$_OFFTGT = (0,0,0)

$$X2\_\overline{OFFTGT} = \begin{pmatrix} XOFF\_ILOAD\_ARRAY\ (PROX\_TGT\_SET\_NO + 1) \\ YOFF\_ILOAD\_ARRAY\ (PROX\_TGT\_SET\_NO + 1) \\ ZOFF\_ILOAD\_ARRAY\ (PROX\_TGT\_SET\_NO + 1) \end{pmatrix}$$

Call OFFSET_TGT; inputs: OMEGA_PROX, DT_OFFTGT, $\overline{X}$_OFFTGT, $\overline{XD}$_OFFTGT, $\overline{X2}$_OFFTGT

output: $\overline{DV}$

b. Define internal variables.

$\overline{XD}1 = DV_1$

$\overline{YD}1 = DV_2$

$\overline{ZD}1 = DV_3$

W = OMEGA_PROX

$\overline{X}0 = COMP\_X_1$

$\overline{Y}0 = COMP\_X_2$

$\overline{Z}0 = COMP\_X_3$

X1 = COMP_T2_XOFF

Y1 = COMP_T2_YOFF

Z1 = COMP_T2_ZOFF

$X2 = \overline{X2}\_OFFTGT_1$

ORIGINAL PAGE IS
OF POOR QUALITY

Y2 = $\overline{X2}$_OFFTGT$_2$

Z2 = $\overline{X2}$_OFFTGT$_3$

ALPHA = (X1 XD1 + Y1 YD1 + Z1 ZD1)/W

A = 8 ALPHA -12 X1 Z1 -2XO Z1 + 2X1 ZO

B = -A

C = XO X1 - X1 X1 + 4ZO Z1 -4 Z1 Z1 + 8 YO Y1

D = 3(Y1**2 + Z1**2)

E = -3 ALPHA + 6 X1 Z1

F = 8 Y1(YO + Y1)

G = -8 YO Y1

H = -3 YO Y1

I = -3 Y1 Y1

J = -8 Y1 Y1

L = -3 ZO Z1

M = -8 YO Y1

c. Set up Newton-Raphson iteration.

IC = 0

ICMAX = 10

DEL_X_TOL = 0

X_IND = -60 COMP_PROX_DT

DEL_X_GUESS = 100

d. Do until IC > ICMAX or ABS (X_IND-X_IND_PRIME) < 0.5:

T = W X_IND

COS = COS (T)

SIN = SIN (T)

TAN = SIN/COS

X_DEP = A + B COS + C SIN + DT COS + ET SIN + F COS/TAN + G COS COS/TAN

+ HT COS COS + I T SIN SIN + J/TAN + LT + M COS SIN

Call ITERV; inputs: IC, X_DEP, X_IND

X_DEP_PRIME, X_IND_PRIME

DEL_X_GUESS, ICMAX, DEL_X_TOL

outputs: IC, X_IND, X_DEP_PRIME

X_IND_PRIME, SFAIL

e. Compute the transfer time (COMP_PROX_DT)

COMP_PROX_DT = X_IND/60

4.15.2 Interface Requirements

The input and output parameters for the omega-dt calculation task are given in tables 34 and 35.

4.15.3 Processing Requirements

Perform once on call.

4.15.4 Initialization

None.

4.15.5 Supplemental Information

None.

4.16 ELEVATION ANGLE SEARCH TASK (TELEV)

The elevation angle search task determines the time a desired elevation angle exists between the Shuttle and the target.

4.16.1 Detailed Requirements

The following steps are required in the iteration logic to determine the time a desired elevation angle exists between the Shuttle and the target.

a. Set the iteration counter to zero, and initialize the previous pass differential altitude indicator to indicate inconsistency:

IC = 0

NN = 0

b. Advance both the Shuttle and the target to first-guess time where the desired elevation angle should exist. The advancement is performed in two calls to the state vector update task (sec. 4.21), with inputs of vehicle position and velocity vectors, first-guess time. The outputs are the vehicle position and velocity vectors, and time at the first-guess time.

Update Shuttle state:

S_OPTION = 1

$\overline{\mathrm{R\_IN}} = \overline{\mathrm{RS\_M50\_PROX}}$

$\overline{\mathrm{V\_IN}} = \overline{\mathrm{VS\_M50\_PROX}}$

T_IN = TIME_PROX + BASE_MET

T_OUT = 60. T1_ILOAD_ARRAY (PROX_TGT_SET_NO) + BASE_MET

Call UPDATVP; inputs: $\overline{\mathrm{R\_IN}}$, $\overline{\mathrm{V\_IN}}$, T_IN, T_OUT

outputs: $\overline{\mathrm{R\_OUT}}$, $\overline{\mathrm{V\_OUT}}$

$\overline{\mathrm{RS\_OUT}} = \overline{\mathrm{R\_OUT}}$

$\overline{\mathrm{VS\_OUT}} = \overline{\mathrm{V\_OUT}}$

TS_OUT = T_OUT

Update target state:

S_OPTION = 2

$\overline{\mathrm{R\_IN}} = \overline{\mathrm{RT\_M50\_PROX}}$

$\overline{\mathrm{V\_IN}} = \overline{\mathrm{VT\_M50\_PROX}}$

Call UPDATVP; inputs: $\overline{\mathrm{R\_IN}}$, $\overline{\mathrm{V\_IN}}$, T_IN, T_OUT

outputs: $\overline{\mathrm{R\_OUT}}$, $\overline{\mathrm{V\_OUT}}$

$\overline{\mathrm{RT\_OUT}} = \overline{\mathrm{R\_OUT}}$

$\overline{\mathrm{VT\_OUT}} = \overline{\mathrm{V\_OUT}}$

TT_OUT = T_OUT

c. Perform an iteration to find a time for which the desired elevation angle and the delta-h at that line are consistent (- DELTA_H* (ELANG-PI) $\geq$ 0). Execute the following code while the maximum number of iterations has not been reached (IC $\leq$ ICMAX) and while the delta-h and elevation angle are inconsistent (NN = 0). (NOTE: Initially NN = 0, thus the loop will be entered at least once.)

(1) Calculate the differential altitude:

$$DELTA\_H = |\overline{RT\_OUT}| - |\overline{RS\_OUT}|$$

(2) Test for consistency between the calculated delta-h and the desired elevation angle.

(a) If the elevation angle and differential altitude are consistent, set the differential altitude consistency indicator to indicate consistency.

If -DELTA_H (EL_ANG - PI) $\geq$ 0, then

NN = 1

(b) If they are not consistent, compute a dependent variable (ERR_DH) with the differential altitude and incompatibility tolerance (EL_DH_TOL).

If -DELTA_H (EL_ANG - PI) < 0, then

ERR_DH = DELTA_H + EL_DH_TOL SIGN(DELTA_H)

Calculate the new guess at the time of consistency and advance the Shuttle and target to the time. This is performed in the elevation angle iterator task (sec. 4.19), with inputs of Shuttle position and velocity vectors, iteration counter, error in the elevation angle, current TPI time, previous error, and previous time. The outputs are the Shuttle and target position and velocity vectors, time, iteration counter, the new time, the new values of the previous error and time, and an error flag.

ERR = ERR_DH

ERR_PRIME = ERR_DH_PRIME

Inputs/outputs for the elevation angle iteration (ELITER) task are:

Inputs: $\overline{RS\_OUT}$, $\overline{VS\_OUT}$, $\overline{RT\_OUT}$, VT_OUT,

IC, ERR, TTPI, ERR_PRIME,

TTPI_PRIME

Outputs: $\vec{\text{RS\_OUT}}$, $\vec{\text{VS\_OUT}}$, TS_OUT, $\vec{\text{RT\_OUT}}$, $\vec{\text{VT\_OUT}}$

TT_OUT, IC, TTPI, ERR_PRIME, TTPI_PRIME

ALARM

ERR_DH_PRIME = ERR_PRIME

d. Determine the time that the desired elevation angle exists. Test to see if the delta-h iteration has failed (i.e., the consistency flag, NN, is still set to zero).

(1) If the delta-h iteration has failed (i.e., the consistency flag is set to OFF, NN = 0), then exit the routine, since there will be no time for which the desired elevation angle will exist.

(2) If the consistency flag is set to ON (NN = 1), then find the time of elevation angle.

(a) If iterations have occurred (IC $\neq$ 0), compute a TPI time in the direction of the direction of difference between the current and previous TPI time and set iteration counter (IC) to zero. This is done to ensure that the first two iterations are consistent.

TTPI = TTPI + 10 SIGN (TTPI - TTPI_PRIME)

IC = 0

Advance both the Shuttle and target to the time of TPI. This is performed by two calls to the state vector update task, with inputs of Shuttle and target positions and velocity vectors, time, and TPI time. The outputs are position and velocity vectors, and time.

Update Shuttle state:

S_OPTION = 1

$\vec{\text{R\_IN}}$ = $\vec{\text{RS\_OUT}}$

$\vec{\text{V\_IN}}$ = $\vec{\text{VS\_OUT}}$

T_IN = TTPI_PRIME + BASE_MET

T_OUT = TTPI + BASE_MET

Call UPDATVP; inputs: $\vec{\text{R\_IN}}$, $\vec{\text{V\_IN}}$, T_IN, T_OUT

outputs: $\vec{\text{R\_OUT}}$, $\vec{\text{V\_OUT}}$

$\vec{\text{RS\_OUT}}$ = $\vec{\text{R\_OUT}}$

$\overline{VS\_OUT} = \overline{V\_OUT}$

TS_OUT = T_OUT

Update target state:

S_OPTION = 2

$\overline{R\_IN} = \overline{RT\_OUT}$

$\overline{V\_IN} = \overline{VT\_OUT}$

Call UPDATVP; inputs: $\overline{R\_IN}$, $\overline{V\_IN}$, T_IN, T_OUT

outputs: $\overline{R\_OUT}$, $\overline{V\_OUT}$

$\overline{RT\_OUT} = \overline{R\_OUT}$

$\overline{VT\_OUT} = \overline{V\_OUT}$

TT_OUT = T_OUT

(b) Perform an iteration to find the time of elevation angle. Execute the following code as long as the maximum number of iterations has not been reached and the current error is larger than the tolerance, or while the number of iterations is equal to zero. (This condition forces at least one iteration.) The condition for iteration is

$$\left[(ERR\_EL \geq EL\_TOL) \text{ and } (IC \leq ICMAX)\right] \text{ or } (IC \neq 0)$$

(i) Calculate the elevation angle that currently exists between the two vehicles. The calculations are performed in the elevation angle computation task (sec. 4.18), with inputs of the Shuttle position and velocity vectors, and target position vector. The output is the elevation angle.

$\overline{RS\_COM} = \overline{RS\_OUT}$

$\overline{VS\_COM} = \overline{VS\_OUT}$

$\overline{RT\_COM} = \overline{RT\_OUT}$

Call COMELE; inputs: $\overline{RS\_COM}$, $\overline{VS\_COM}$, $\overline{RT\_COM}$

outputs: EL_ANG_COM

(ii) Calculate the difference between the desired and computed elevation angles:

ERR_EL = EL_ANG_EL_ANG_COM

Calculate the new guess at the time of TPI and advance the Shuttle and target to the time. This is performed in the elevation angle iterator task (sec. 4.19), with inputs of Shuttle position and velocity vectors, iteration counter, error in the elevation angle, current TPI time, previous error, and TPI time. The outputs are the Shuttle and target positions and velocity vectors, time, iteration counter, new TPI time, new values of the previous error and TPI time, and an error flag.

ERR = ERR_EL

ERR_PRIME = ERR_EL_PRIME

Inputs/outputs for the elevation angle iteration task (ELITER) are:

Inputs: $\overline{\text{RS\_OUT}}$, $\overline{\text{VS\_OUT}}$, $\overline{\text{RT\_OUT}}$, $\overline{\text{VT\_OUT}}$, IC,

ERR, TTPI, ERR_PRIME, TTPI_PRIME

Outputs: $\overline{\text{RS\_OUT}}$, $\overline{\text{VS\_OUT}}$, TS_OUT, $\overline{\text{RT\_OUT}}$, $\overline{\text{VT\_OUT}}$,

TT_OUT, IC, TTPI, ERR_PRIME, TTPI_PRIME,

ALARM

ERR_EL_PRIME = ERR_PRIME

4.16.2 Interface Requirements

The input and output parameters for the elevation angle search task are given in tables 36 and 37.

4.16.3 Processing Requirements

Perform once on call.

4.16.4 Initialization

None.

4.16.5 Supplemental Information

None.

4.17 PRECISION-REQUIRED VELOCITY TASK (PREVR)

The precision-required velocity task computes the precision velocity-required to satisfy both terminal position and time of flight constraints (Lambert problem).

4.17.1 Detailed Requirements

The following steps are required to compute the velocity that satisfies the Lambert problem constraints.

a. Set or initialize variables that are used in the iteration to find the required velocity:

$$\text{DEL\_T\_TRAN} = \text{T2\_TIG} - \text{T1\_TIG}$$

$$\overline{\text{R\_OFFSET}} = \overline{\text{RS\_T2TIG}}$$

$$\text{T\_OFFSET} = \text{T2\_TIG}$$

$$\text{N} = 0$$

$$\overline{\text{VG}} = \overline{0}$$

$$\text{VG\_MAG} = 0$$

$$\text{ALPHA} = 2/|\overline{\text{RS\_T1TIG}}| - (\overline{\text{VS\_T1TIG}} \cdot \overline{\text{VS\_T1TIG}})/\text{EARTH\_MU}$$

$$\text{ORB\_RATE} = \text{ALPHA (ALPHA EARTH\_MU)}^{.5}$$

$$\text{S\_ROTATE} = 1$$

$$\overline{\text{UN\_REF}} = \text{UNIT } (\overline{\text{RS\_T1TIG}} \text{ X } \overline{\text{VS\_T1TIG}})$$

$$\overline{\text{RS\_REF}} = \overline{\text{RS\_T1TIG}}$$

$$\text{ACC} = \text{THRUST G/WS}$$

b. Perform the iteration to compute the required velocity. Continue until the miss distance between the Shuttle position vector at the terminal time is within a tolerance ($|\text{R\_MISS}| \leq \text{R\_TOL}$), and a minimum number of iterations are achieved ($\text{N} > \text{N\_MIN}$), or the alarm flag is set ($\text{ALARM} > 0$).

(1) Increase the iteration counter by one and set the initial position offset vector to the Shuttle position vector:

$$\text{N} = \text{N} + 1$$

$$\overline{\text{RS\_IPO}} = \overline{\text{RS\_REF}}$$

(2) Perform a logical test to determine if the transfer angle is near 180°.

(a) If the transfer angle is not near 180° (S_ROTATE ≠0), calculate the transfer plane vector and the transfer angle before and after the burn:

$\overline{UN}$ = $\overline{RS\_IPO}$ X $\overline{R\_OFFSET}$

S_ROTATE = SIGN ($\overline{UN}\cdot\overline{UN\_REF}$)

SBETA = S_ROTATE $|\overline{UN}|$

CBETA = $\overline{RS\_IPO}$ · $\overline{R\_OFFSET}$

BBEF = PI + ARCTAN2 (-SBETA, -CBETA)

BAFT = BBEF - ORB_RATE VG_MAG/ACC

(b) Perform a logical test to determine if the transfer angle (before and after) the burn is not near 180°.

(i) If the transfer angle is not near 180° (BBEF < PI - CONE or BAFT > PI + CONE), calculate the transfer plane:

$\overline{UN}$ = $\overline{UN}$/SBETA

(ii) If the transfer angle is near 180° (BBEF > PI - CONE and BAFT < PI + CONE), set the iteration counter to zero, set the transfer plane equal to the Shuttle orbital plane, set the projection flag to zero, and project the offset position vector into the transfer plane:

N = 0

S_ROTATE = 0

$\overline{UN}$ = $\overline{UN\_REF}$

$\overline{R\_OFFSET}$ = $\overline{R\_OFFSET}$ - ($\overline{R\_OFFSET}$ · $\overline{UN}$) $\overline{UN}$

(3) Calculate the required velocity vector, which with the position vector will pass through the desired target vector at the desired time. The calculations for the required velocity vector are in the Lambert conic velocity-required task (sec. 4.20) with inputs of the initial position offset vector, the desired target offset vector, the transfer plane vector, and the transfer time. The outputs are the required velocity vector and an alarm or error flag.

Inputs/outputs for the Lambert conic-velocity-required task (LAMBERT) are:

Inputs: $\overline{RS\_IPO}$, $\overline{R\_OFFSET}$, $\overline{UN}$, DEL_T_TRAN

Outputs: $\overline{VS\_REQUIRED}$, ALARM

(4) Calculate the velocity-to-be-gained vector and magnitude:

$$\overline{VG} = \overline{VS\_REQUIRED} - \overline{VS}\ T1TIG$$

$$\overline{VG\_MAG} = |VG|$$

(5) Advance the Shuttle state vector to the terminal point by a call to the state vector update task (sec. 4.21), with inputs of the state vector time, update option flag, terminal point time. The output is the state vector at terminal point time.

$$S\_OPTION = 1$$

$$\overline{R\_IN} = \overline{RS\_IPO}$$

$$\overline{V\_IN} = \overline{VS\_REQUIRED}$$

$$T\_IN = T1\_TIG + BASE\_MET$$

$$T\_OUT = T2\_TIG + BASE\_MET$$

Call UPDATVP; inputs: $\overline{R\_IN}$, $\overline{V\_IN}$, T_IN, T_OUT, S_OPTION

outputs: $\overline{R\_OUT}$, $\overline{V\_OUT}$

$$\overline{RS\_TERMINAL} = \overline{R\_OUT}$$

$$\overline{VS\_TERMINAL} = \overline{V\_OUT}$$

(6) Compute the miss-distance vector between the Shuttle position vector and the desired position vector at the terminal point:

$$\overline{R\_MISS} = \overline{RS\_TERMINAL} - \overline{RS\_T2TIG}$$

(7) Perform a logic test to determine if the transfer angle is near 180°. If it is (S_ROTATE = 0), compute the miss as a projection into the actual transfer plane:

$$\overline{R\_MISS} = \overline{R\_MISS} - (\overline{R\_MISS} \cdot \overline{UN})\ \overline{UN}$$

(8) Calculate the offset position vector that the required velocity calculations are based on. Increment the iteration counter:

$$\overline{R\_OFFSET} = \overline{R\_OFFEST} - \overline{R\_MISS}$$

(9) Perform a logic test to determine if the maximum number of iterations were exceeded.

If $N > NMAX$, then ALARM = 6

ORIGINAL PAGE IS
OF POOR QUALITY

4.17.2 Interface Requirements

The input and output parameters for the precision-required velocity task are given in tables 38 and 39.

4.17.3 Processing Requirements

Perform once on call.

4.17.4 Initialization

None.

4.17.5 Supplemental Information

None.

4.18 ELEVATION ANGLE COMPUTATION TASK (COMELE)

The elevation angle computation task computes the elevation angle between one vehicle position vector and another vehicle position vector. The elevation is defined as the angle measured from the local horizontal plane of the Shuttle to the line of sight of the target.

4.18.1 Detailed Requirements

The following steps are required to calculate the elevation angle between one vehicle position vector and another vehicle position vector.

a. Calculate intermediate variables used in the elevation angle computation:

$$A = \overline{RS\_COM} \cdot \overline{RS\_COM}$$

$$B = \overline{RS\_COM} \cdot \overline{RT\_COM}$$

$$C = \overline{RT\_COM} \cdot \overline{RT\_COM}$$

$$D = \overline{RS\_COM} \cdot \overline{VS\_COM}$$

$$E = \overline{RT\_COM} \cdot \overline{VS\_COM}$$

b. Calculate the elevation angle between two position vectors ($\overline{RS\_COM}$ AND RT_COM):

If $(A\,C - B^2) < 0$, then $EL\_ANG\_COM = \pi + [SIGN\ (A-C)]\ \pi/2$, otherwise

$EL\_ANG\_COM = PI + ARCTAN2 (A - B, [SIGN (B D - A E)](A C - B^2)^{1/2})$

If EL_ANG_COM = 2·PI, reset EL_ANG_COM = 0

4.18.2 Interface Requirements

The input and output parameters for the elevation angle computation task are given in tables 40 and 41.

4.18.3 Processing Requirements

Perform once on call.

4.18.4 Initialization

None.

4.18.5 Supplemental Information

None.

4.19 ELEVATION ANGLE ITERATION TASK (ELITER)

The elevation angle iterator task computes the time of TPI and updates the Shuttle and target to this time.

4.19.1 Detailed Requirements

The following steps are required in the computation and the advancement of the Shuttle and target to the time of the TPI maneuver.

a. Calculate the new guess at the time of TPI. This is performed in the Newton-Raphson iteration task (sec. 4.22), with inputs of iteration counter, current error in the elevation angle, current TPI time, previous error, and TPI time. The outputs are the new TPI time, new previous error and TPI time, and an error flag that is set if the maximum number of iterations is exceeded.

   X_DEP = ERR

   X_IND = TTPI

   X_DEP_PRIME = ERR_PRIME

X_IND_PRIME = TTPI_PRIME

Call ITERV; inputs: IC, X_DEP, X_IND, X_DEP_PRIME, X_IND_PRIME

outputs: IC, X_IND, X_DEP_PRIME, X_IND_PRIME, SFAIL

TTPI = X_IND

ERR_PRIME = X_DEP_PRIME

TTPI_PRIME = X_IND_PRIME

b. Perform a logic test to determine if the maximum number of iterations has been exceeded.

(1) If the iteration maximum has been exceeded (SFAIL ≠ 0), set an ALARM flag and terminate the elevation angle iterator task:

If SFAIL ≠ 0, then ALARM = 7

(2) If the maximum number of iterations has not been exceeded (SFAIL = 0,), perform a logic test to determine if the current TPI time is greater than some delta-t from the previous TPI time. If it is, set the current TPI time to the previous time plus the delta-t:

If |TTPI - TTPI_PRIME| > DEL_T_MAX,

then TTPI = TTPI_PRIME + DEL_T_MAX sign (TTPI - TTPI_PRIME)

(3) Advance both Shuttle and target to the current TPI time. The state vector update task performs the advancement with two calls, one for the Shuttle and the other for the target. The inputs are position and velocity vectors and TPI time. The outputs are position and velocity vectors and time of TPI.

Update Shuttle state:

S_OPTION = 1

$\overline{\text{R\_IN}}$ = $\overline{\text{RS\_OUT}}$

$\overline{\text{V\_IN}}$ = $\overline{\text{VS\_OUT}}$

T_IN = TTPI_PRIME + BASE_MET

T_OUT = TTPI + BASE_MET

Call UPDATVP; inputs: $\overline{\text{R\_IN}}$, $\overline{\text{V\_IN}}$, T_IN, T_OUT

outputs: $\overline{\text{R\_OUT}}$, $\overline{\text{V\_OUT}}$

$\overline{\text{RS\_OUT}}$ = $\overline{\text{R\_OUT}}$

$\overline{\text{VS\_OUT}} = \overline{\text{V\_OUT}}$

TS_OUT = T_OUT

Update target state:

S_OPTION = 2

$\overline{\text{R\_IN}} = \overline{\text{RT\_OUT}}$

$\overline{\text{V\_IN}} = \overline{\text{VT\_OUT}}$

Call UPDATVP; inputs: $\overline{\text{R\_IN}}$, $\overline{\text{V\_IN}}$, T_IN, T_OUT

outputs: $\overline{\text{R\_OUT}}$, $\overline{\text{V\_OUT}}$

$\overline{\text{RT\_OUT}} = \overline{\text{R\_OUT}}$

$\overline{\text{VT\_OUT}} = \overline{\text{V\_OUT}}$

TT_OUT = T_OUT

4.19.2 Interface Requirements

The input and output parameters for the elevation angle iteration task are given in table 42 and 43.

4.19.3 Processing Requirements

Perform once on call.

4.19.4 Initialization

None.

4.19.5 Supplemental Information

None.

4.20 LAMBERT CONIC-VELOCITY-REQUIRED TASK (LAMBERT)

The Lambert conic-velocity-required task calculates the required orbital velocity vector that satisfies the following Lambert problem: given an initial position vector, a terminal position vector, and a specified transfer time between the initial and final vector-, determine the initial velocity required to transfer from the initial vector to the final vector with the required transfer time, assuming two-body conic orbital motion. The equations hold only for the elliptical case, and the task does not have multirevolution capability. This task is used both in targeting and in the execution of a Lambert-guided maneuver.

4.20.1 Detailed Requirements

The following steps are required to calculate the initial velocity vector to transfer from an initial position vector to a final position vector with a specified transfer time.

a. The alarm flag is set to the off position. The alarm flag is used to indicate when problems occur from input or from calculations:

   ALARM = 0

b. Calculate the magnitude of the initial vector and the final vector:

   $\overline{RO} = \overline{RS\_IPO}$

   $\overline{R1} = \overline{R\_OFFSET}$

   $RO\_MAG = |\overline{RO}|$

   $R1\_MAG = |\overline{R1}|$

c. Calculate the semiperimeter of the transfer triangle to be used as a normalizing constant:

   $R\_PARABOLA = (RO\_MAG + R1\_MAG + |R1 - RO|)/2$

d. Calculate the parabolic velocity at perigee for R_PARABOLA for use as a normalizing factor.

   $V\_PARABOLA = 2\ EARTH\_MU/R\_PARABOLA$

e. Calculate the parameter Z, where $Z = RO\_MAG\ R1\_MAG - \overline{RO} \cdot \overline{R1}$. Z is equal to $RO\_MAG\ R1\_MAG(1-\cos\theta)$, where θ is the transfer angle. Test to determine if θ is close to 0 or 360°. If $Z \leq EP\_TRANSFER\ RO\_MAG\ R1\_MAG$, set the alarm flag and exit:

   ALARM = 2

f. Calculate the cotangent of $\theta/2$ (also to be designated by the parameter Z). RO and R1 are assumed to be in the plane defined by UN:

   $Z = (\overline{RO} \times \overline{R1}) \cdot \overline{UN}/Z$

g. Calculate the parameter VH:

   $VH = \sqrt{RO\_MAG\ R1\_MAG/(1 + Z^2)}$

h. Set the initial value on the counter to count the number of iterations:

N = 0.

i. Set the upper and lower limits of the independent variable:

U_MAX = 1 - DU/2

U_MIN = -1.0

DU is an I-load parameter to prevent the orbit from becoming close to parabolic.

j. Calcuate the constant parameter, LAMBDA:

LAMBDA = (VH/R_PARABCLA) Z

k. Calculate the first guess for the independent variable. This first guess assumes a circular orbit:

$U = LAMBDA/\sqrt{1 + LAMBDA^2}$

l. Calculate the normalized transfer time:

T_TILDA_DESIRED = (V_PARABOLA/R_PARABOLA) DEL_T_TRAN

m. Determine a tranfer time that is slightly greater than the parabolic transfer time:

$T\_MIN = (2/3)\ (1 - LAMBDA^3) + (0.4)\ (1 - LAMBDA^5)\ DU$

If the input transfer time is negative or is less than or equal to T MIN, set the alarm flag and exit:

ALARM = 1

n. The initialization procedure has been completed. The iteration steps to determine the value of U to yield the desired transfer time from Kepler's equation follows. This iteration continues until the time error is within a given tolerance or until the number of iterations, N, is equal to the maximum number of iterations, NMAX. Set ALARM = 5 and exit if N = N_MAX.

(1) Increase the counter on the number of iterations by one:

N = N + 1

(2) The following parameters are calculated to obtain the transfer time corresponding to the current value of U:

$W = \sqrt{1 - U^2}$

X = LAMBDA W

$Y = \sqrt{1 - X^2}$

$F = W\ Y - U\ X$

$G = U\ Y + W\ X$

(3) The current value of the transfer time is given by

$T\_TILDA = \left[ARCTAN2\ (F,\ G) - (U\ W - X\ Y)\right]/W^3$

The ARCTAN2 function should produce angles between 0 degree and 2 pi radians.

(4) Determine the slope of T_TILDA with respect to the independent variable U for use with the Newton-Raphson iteration:

$S\_TILDA = (3\ U\ T\_TILDA - 2(1 - (U/Y)LAMBDA^3))/W^2$

(5) Determine the error in the current value of T_TILDA and the desired value T_TILDA_DESIRED:

T_TILDA_ERROR = T_TILDA_DESIRED - T_TILDA

(6) Determine if the error in T_TILDA_DESIRED is small enough to consider the current value of U as the correct answer. If the absolute value of T_TILDA_ERROR is less than T_TILDA_DESIRED times EPS_U, then the Newton-Raphson iteration is considered converged, and the required initial velocity can be determined in step o. Otherwise, continue calculations.

(7) Calculate a change in the current value of U:

U_STEP = T_TILDA_ERROR/S_TILDA

If U_STEP is positive, then set U_MIN equal to U. Set the new value of U equal to U plus U_STEP. However, if the value of U is greater than UMAX, set U equal to (U_MIN + U_MAX)/2.

If U_STEP is negative, set U_MAX equal to the current value of U. Set the new value of U equal to U plus U_STEP. However, if the new value of U is less than U_MIN, set U equal to (U_MIN + U_MAX)/2.

o. The required initial velocity can now be obtained by the following relationships:

VH = VH/RO_MAG

VR = (R_PARABOLA/RO_MAG) LAMBDA - G

COEF = V_PARABOLA/(Y - LAMBDA U)

ORIGINAL PAGE IS OF POOR QUALITY

$\overline{VS\_REQUIRED}$ = (COEF/RO_MAG)(VR $\overline{RO}$ + VH ($\overline{UN}$ x $\overline{RO}$))

where $\overline{UN}$ is a unit vector in the direction of the angular momentum vector.

4.20.2 Interface Requirements

The input and output parameters for the Lambert conic-velocity-required task are given in tables 44 and 45.

4.20.3 Processing Requirements

Perform once on call.

4.20.4 Initialization

None.

4.20.5 Supplemental Information

If the transfer angle is near 180°, the parameter LAMBDA will be near zero. When this occurs, raising LAMBDA to a power in the code may result in an underflow. This situation must be covered in the code.

4.21 STATE VECTOR UPDATE TASK (UPDATVP)

The state vector update task calls for the precision onorbit predictor to update a state vector to a time.

4.21.1 Detailed Requirements

The following steps are required to update a state vector to a specific point of interest.

a. Determine if the Shuttle state or the target state, is to be updated. (S_OPTION = 1, 2, respectiv ly). Also set the onorbit predictor parameters: gravity model degree and order flag, drag model, vent model, altitude mode flag, and minimum integration step size.

Set GMD_PRED = GMD_I

GMO_PRED = GMO_I

ATM = ATM_I

Set DMP = DMP_I (S_OPTION)

VMP = VMP_I (S_OPTION)

PRED_ORB_MASS = PRED_ORB_MASS_I (S_OPTION)

PRED_ORB_CD = PRED_ORB_CD_I (S_OPTION)

PRED_ORB_AREA = PRED_ORB_AREA_I (S_OPTION)

b. Call the onorbit predictor to update the state vector to the final time.

Call ON_ORB_PRED; inputs: GMD_PRED, GMO_PRED, DMP, VMP, ATM, PRED_ORB_MASS, PRED_ORB_CD, PRED_ORB_AREA, R_IN, V_IN, T_IN, T_OUT

outputs: R_OUT, V_OUT

4.21.2 Interface Requirements

The input and output parameters for the state vector update task are given in tables 46 and 47.

4.21.3 Processing Requirements

Perform once per call.

4.21.4 Initialization

None.

4.21.5 Supplemental Information

None.

4.22 NEWTON-RAPHSON ITERATION TASK (ITERV)

The Newton-Raphson task performs an iteration with a secant numerical derivative to drive the differences between two successive values of a dependent variable to zero.

4.22.1 Detailed Requirements

The following are the steps required to drive the successive difference of the dependent variable to zero.

a. Perform a logic test to determine if it is the first pass through the function.

   (1) If it is the first pass (IC = 0), set the delta independent variable to a constant first guess:

   DEL_X_IND = DEL_X_GUESS

   (2) If it is not the first pass (IC ≠ 0), calculate the delta dependent variable from the current and previous values of the dependent variable.

   DEL_X_DEP = X_DEP - X_DEP_PRIME

   Perform a logic test of DEL_X_DEP to see if it is less than a small tolerance:

   If it is (DEL_X_TOL), then

   DEL_X_IND = DEL_X_GUESS

   If DEL_X_DEP is not less than DEL_X_TOL, then

$$\text{SLOPE} = \frac{\text{DEL\_X\_DEP}}{(\text{X\_IND} - \text{X\_IND\_PRIME})}$$

$$\text{DEL\_X\_IND} = \frac{\text{X\_DEP}}{\text{SLOPE}}$$

b. Set the previous values of the dependent and independent variable to the current values. Calculate the new current values of the independent variable. Add one to the iteration counter, and set an iteration failure flag to zero:

   SFAIL = 0

   IC = IC + 1

   X_DEP_PRIME = X_DEP

   X_IND_PRIME = X_IND

   X_IND = X_IND - DEL_X_IND

c. Perform a logic test to determine if the maximum number of iterations has been exceeded. If the iteration exceeds the maximum (IC > ICMAX), set the failure flag:

   SFAIL = 1

4.22.2 Interface Requirements

The input and output parameters for the Newton-Raphson iteration task are given in tables 48 and 49.

4.22.3 Processing Requirements

Perform once on call.

4.22.4 Initialization

None.

4.22.5 Supplemental Information

None.

4.23 ORBITER LVLH TRANSFORMATION TASK (ORBLV)

The Orbiter LVLH transformation tasks determines the position and velocity of the target as seen from an Orbiter-centered LVLH frame starting with Orbiter and target M50 inertial vectors.

4.23.1 Detailed Requirements

The following steps are required to perform this task.

a. Find the magnitude of the Shuttle inertial vector.

$$\text{RS\_MAG} = \text{MAG}\ (\overline{\text{RS}})$$

b. Compute the transformation matrix from M50 inertial frame to the local vertical inertial Shuttle-centered rectangular coordinate frame:

$$\text{MAT\_M50\_LVIR} = \begin{bmatrix} \text{unit}\ ((\overline{\text{RS}}\ \text{X}\ \overline{\text{VS}})\text{X}\ \overline{\text{RS}})^T \\ -\ \text{unit}\ (\overline{\text{RS}}\ \text{X}\ \overline{\text{VS}})^T \\ -\ \text{unit}\ (\overline{\text{RS}})^T \end{bmatrix}$$

c. Determine the orbital angular rate vector.

$$\overline{\text{VTAN}} = \overline{\text{VS}}\text{-UNIT}(\overline{\text{RS}})\ (\overline{\text{RS}} \cdot \overline{\text{VS}})/\text{RS\_MAG}$$

$$\overline{\text{OMEGA\_LV\_PROX}} = (|\overline{\text{VTAN}}|/\text{RS\_MAG}) \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}$$

d. Compute the relative M50 state of the target with respect to the Shuttle.

$$\vec{RST\_M50} = \vec{RT} - \vec{RS}$$

$$\vec{VST\_M50} = \vec{VT} - \vec{VS}$$

e. Convert to a Shuttle-centered LV inertial rectangular frame.

$$\vec{RST\_LVIR} = \left[MAT\_M50\_LVIR \right] \vec{RST\_M50}$$

$$\vec{VST\_LVIR} = \left[MAT\_M50\_LVIR \right] \vec{VST\_M50}$$

f. Convert to a Shuttle-centered LVLH frame.

$$\vec{RSLV} = \vec{RST\_LVIR}$$

$$\vec{VSLV} = \vec{VST\_LVIR} - (OMEGA\_LV\_PROX \times \vec{RST\_LVIR}).$$

4.23.2 Interface Requirements

The input and output parameters for the Orbiter LVLH transformation task are given in table 50 and 51.

4.23.3 Processing Requirements

Perform once on call.

4.23.4 Initialization

None.

4.23.5 Supplemental Information

None.

TABLE 1.- PROX_WORLD COMMON

SYMBOL	PRECISION	DEFINITION	TYPE	UNITS
TM_IND	TBD	Time GMT/MET select indicator	D	--
T_CURRENT	TBD	Current time	F	sec
BASE_MET	TBD	GMT/MET reference time	F	sec
DELTA_$\bar{V}$_LVLH_GUID	TBD	Delta-v vector of current maneuver	F(3)	ft/sec
GUID_FLAG	TBD	Current maneuver guidance option flag	D	--
DT_ST_TIMER	TBD	Time to be counted down	F	sec
CRT_TIME	TBD	CRT timer start/stop discrete	D	--
CRT_UPDWN	TBD	CRT timer up/down discrete	D	--
T_STATE	TBD	Time tag for M50 vectors	F	sec
$\bar{R}$_AVGG	TBD	Orbiter position M50 vectors	F(3)	ft
$\bar{V}$_AVGG	TBD	Orbiter velocity M50 vectors	F(3)	ft/sec
$\bar{R}$_TARGET	TBD	Target position M50 vector	F(3)	ft
$\bar{V}$_TARGET	TBD	Target velocity M50 vector	F(3)	ft/sec
S_ROTATE_GUID	TBD	Rotation flag for near 180° transfers	D	--
$\bar{R}$_OFFSET_GUID	TBD	Offset position vector	F(3)	ft
T_OFFSET_GUID	TBD	Time of offset position vector (MET)	F	sec
TIG_GUID_DAY	TBD	Ignition time of current maneuver (MET)	I	day
TIG_GUID_HR	TBD		I	hr
TIG_GUID_MIN	TBD		I	min
TIG_GUID_SEC	TBD		F	sec

TABLE 1.– Concluded

SYMBOL	PRECISION	DEFINITION	TYPE	UNITS
GMD_PRED	TBD	Gravity model degree flag	D	--
GMO_PRED	TBD	Gravity model order flag	D	--
ATM	TBD	Attitude mode flag	D	--
DMP	TBD	Drag model flag	D	--
VMP	TBD	Vent model flag	D	--
PRED_ORB_MASS	TBD	Onorbit predictor drag parameters	F	?
PRED_ORB_CD	TBD		F	?
PRED_ORB_AREA	TBD		F	ft^2
$\bar{R}$_IN	TBD	Input inertial position vector to UPDATVP	F(3)	ft
$\bar{V}$_IN	TBD	Input inertial velocity vector to UPDATVP	F(3)	ft/sec
$\bar{R}$_OUT	TBD	Output inertial position vector to UPDATVP	F(3)	ft
$\bar{V}$_OUT	TBD	Output inertial velocity vector to UPDATVP	F(3)	ft/sec
T_IN	TBD	Time of the input state to UPDATVP (GMT)	F	sec
T_OUT	TBD	Time of the output state to UPDATVP (GMT)	F	sec
WS	TBD	Orbiter mass for PREVR	F	lbs

TABLE 2.- PROX_DIP COMMON

SYMBOL	PRECISION	DEFINITION	TYPE	UNITS
PROX_T1_STAR_STATUS	TBD	T1 maneuver compute star flag	D	--
PROX_T2_STAR_STATUS	TBD	T2 maneuver compute star flag	D	--
PROX_PAST_STATUS	TBD	Maneuver past time status flag	D	--
PROX_TGT_SET_NO	TBD	ILOAD set number selected for targeting	I	--
DISP_PROX_DT	TBD	Displayed delta-t between T1 and T2	F	min
DISP_EL_ANG	TBD	Displayed target elevation angle at T1	F	deg
DISP_T1_X	TBD	Orbiter relative position at T1	F(3)	ft
DISP_T1_XD	TBD	Orbiter relative velocity at T1	F(3)	ft/sec
DISP_T1_DAY	TBD		I	day
DISP_T1_HR	TBD	T1 time	I	hr
DISP_T1_MIN	TBD		I	min
DISP_T1_SEC	TBD		F	sec
DISP_T2_DAY	TBD		I	day
DISP_T2_HR	TBD	T2 time	I	hr
DISP_T2_MIN	TBD		I	min
DISP_T2_SEC	TBD		F	sec
DISP_T2_XOFF	TBD		F	ft
DISP_T2_YOFF	TBD	Orbiter desired relative position at T2	F	ft
DISP_T2_ZOFF	TBD		F	ft

TABLE 2.- Concluded

SYMBOL	PRECISION	DEFINITION	TYPE	UNITS
DISP_DV	TBD	Maneuver LVLH velocity vector	F(3)	ft/sec
DISP_DV_MAG	TBD	Maneuver velocity magnitude	F	ft/sec
PROX_ITEM_21to24_STATUS	TBD	Items 21 through 24 data entry status flag	D	--
PROX_ITEM_1_STATUS	TBD	Item 1 data entry status flag	D	--
PROX_ITEM_2to20 STATUS	TBD	Items 2 through 20 data entry status flag	D	--
PROX_ITEM_25_STATUS	TBD	Item 25 data entry status flag	D	--
PROX_ITEM_26_STATUS	TBD	Item 26 data entry status flag	D	--
PROX_ITEM_27_STATUS	TBD	Item 27 data entry status flag	D	--
PROX_ITEM_28_STATUS	TBD	Item 28 data entry status flag	D	--
PROX_LOAD_FLASH	TBD	"LOAD" flash status flag	D	--
PROX_BASE_DAY	TBD	Targeting base time	I	day
PROX_BASE_HR	TBD		I	hr
PROX_BASE_MIN	TBD		I	min
PROX_BASE_SEC	TBD		F	sec
DISP_TMAN_TIME	TBD	Displayed maneuver time	F	sec
MAN_TGT	TBD	Target set for which maneuver was computed	I	--

TABLE 3.- PROX_ILOAD_COMMON

SYMBOL	PRECISION	DEFINITION	TYPE	UNITS
T1_ILOAD_ARRAY	TBD	T1 time for target sets (times are relative to prox ops base time.)	F(40)	min
DT_ILOAD_ARRAY	TBD	Delta times from T1 maneuver to T2 maneuver for targets sets	F(40)	min
EL_ILOAD_ARRAY	TBD	Maneuver elevation angle	F(40)	rad
XOFF_ILOAD_ARRAY	TBD		F(40)	ft
YOFF_ILOAD_ARRAY	TBD	ILOAD LVLH offset position for target sets	F(40)	ft
ZOFF_ILOAD_ARRAY	TBD		F(40)	ft
PROX_DT_MIN	TBD	Tolerance on minimum time in the future to compute a prox ops maneuver solution	F	sec
PROX_DTMIN_LAMB	TBD	Tolerance on minimum time in the future to compute a Lambert maneuver solution	F	sec
BASE_START_DAY	TBD		I	day
BASE_START_HR	TBD	Reference base time for T1_ILOAD_ARRAY	I	hr
BASE_START_MIN	TBD		I	min
BASE_START_SEC	TBD		F	sec
NLAMB	TBD	Number of Lambert targeted target sets	I	--
PI	TBD	Mathematical constant	F	--
EL_TOL	TBD	Tolerance between computed and desired elevation angle	F	rad
EL_DH_TOL	TBD	Elevation angle-differential altitude in-compatability, tolerance	F	?

TABLE 3.- Continued

SYMBOL	PRECISION	DEFINITION	TYPE	UNITS
R_TOL	TBD	Convergence tolerance in terminal point offset iteration	F	ft
EARTH_MU	TBD	Earth gravitational constant	F	ft^3/sec^2
N_MAX	TBD	Maximum allowed number of iterations	I	--
CONE	TBD	Angular tolerance used to determine if the transfer angle is near 180°	F	rad
DEL_T_MAX	TBD	Maximum step size used during any given iteration	I	--
DU	TBD	Small deviation to prevent orbit from being almost parabolic	F	--
EP_TRANSFER	TBD	Parameter to test if transfer angle is close to 0°	F	?
EPS_U	TBD	Parameter to test convergence of the Newton-Raphson iteration	F	?
NMIN	TBD	Minimum allowed number of iterations	I	--
DEL_X_GUESS	TBD	ΔX guess to be used in iterator if no prediction is possible	F	?
IC_MAX	TBD	Maximum allowed number of iterations	I	--
DEL_X_TOL	TBD	Tolerence of dependent variable to ensure that a slope exists	F	?
GMD_I	TBD	Gravity model degree flag	D	--
GMO_I	TBD	Gravity model order flag	D	--
ATM_I	TBD	Attitude mode flag	D	--
DMP_I	TBD	Drag model flag	D(2)	--
VMP_I	TBD	Vent model flag	D(2)	--

TABLE 3.- Concluded

SYMBOL	PRECISION	DEFINITION	TYPE	UNITS
PRED_ORB_MASS_I	TBD	Onorbit predictor drag parameters	F(2)	?
PRED_ORB_CD_I	TBD	Onorbit predictor drag parameters	F(2)	?
PRED_ORB_AREA_I	TBD	Onorbit predictor drag parameters	F(2)	ft^2
THRUST	TBD	Thrust of the propulsion system for PREVR	F	lbs
G	TBD	Gravitational constant for PREVR	F	ft/sec^2

TABLE 4.- PROX_VARIABLES_COMMON

SYMBOL	PRECISION	DEFINITION	TYPE	UNITS
T1_TIG	TBD	T1 maneuver time (MET)	F	sec
T2_TIG	TBD	T2 maneuver time (MET)	F	sec
PROX_T_CURRENT	TBD	Current MET time	F	sec
PROX_BASE_TIME	TBD	Prox ops base time	F	sec
DAY	TBD	Converted time for TIMECV	I	day
HR	TBD		I	hr
MIN	TBD		I	min
SEC	TBD		F	sec
COMP_PROX_DT	TBD	Delta-t in the computation buffer	F	min
COMP_T2_XOFF	TBD	Computational desired position at T2	F	ft
COMP_T2_YOFF	TBD		F	ft
COMP_T2_ZOFF	TBD		F	ft
TIME_CONVERT_FLAG	TBD	Time conversion flag	D	--
TIME_SEC	TBD	Conversion time	F	sec
$\overline{DV}$_LVLH	TBD	Impulsive maneuver in LVLH reference frame	F(3)	ft/sec
TIME_PROX	TBD	Time tag for M50 state (assumed MET)	F	sec
$\overline{X}$2	TBD	Predicted Orbiter relative state position	F(3)	ft
$\overline{X}$D2	TBD	Predicted Orbiter relative state velocity	F(3)	ft/sec
USE_OMEGA_DT	TBD	Use the ωt calculation flag	D	--
USE_DISP_REL_STATE	TBD	Use displayed relative state flag	D	--
COMP_$\overline{X}$	TBD	Computation relative position	F(3)	ft
COMP_$\overline{X}$D	TBD	Computation relative velocity	F(3)	ft/sec

TABLE 4.- Continued

SYMBOL	PRECISION	DEFINITION	TYPE	UNITS
DTIME	TBD	Time interval for prediction computation	F	sec
OMEGA_PROX	TBD	Orbital angular rate of target	F	rad/sec
$\bar{R}$S_M50_PROX	TBD	Prox ops Orbiter M50 position vector	F(3)	ft
$\bar{R}$T_M50_PROX	TBD	Prox ops target M50 position vector	F(3)	ft
$\bar{V}$S_M50_PROX	TBD	Prox ops Orbiter M50 velocity vector	F(3)	ft/sec
$\bar{V}$T_M50_PROX	TBD	Prox ops target M50 velocity vector	F(3)	ft/sec
$\bar{X}$	TBD	Input relative position for the relative predict task	F(3)	ft
$\bar{X}$D	TBD	Input relative velocity for the relative predict task	F(3)	ft/sec
EL_ANG	TBD	Desired elevation angle at TPI	F	rad
S_ROTATE	TBD	Rotation flag for near 180° transfers	D	--
$\bar{R}$_OFFSET	TBD	Offset position vector	F(3)	ft
T_OFFSET	TBD	Time of offset position vector (MET)	F	sec
INER_TO_LVC	TBD	Inertial to curvilinear conversion flag	D	--
$\bar{R}$_REL	TBD	LVLH curvilinear position vector from REL_COMP	F(3)	ft
$\bar{V}$_REL	TBD	LVLH curvilinear velocity vector from REL_COMP	F(3)	ft/sec
$\bar{D}$V	TBD	Delta-v vector from OFFSET_TGT	F(3)	ft/sec
DT_OFFTGT	TBD	Transfer time for OFFSET_TGT	F	sec
$\bar{X}$_OFFTGT	TBD	T1 relative position vector for OFFSET_TGT	F(3)	ft
$\bar{X}$D_OFFTGT	TBD	T1 relative velocity vector for OFFSET_TGT	F(3)	ft/sec
$\bar{X}$2_OFFTGT	TBD	T2 relative position vector for OFFSET_TGT	F(3)	ft
$\bar{R}$_S_INER	TBD	Input Shuttle inertial position for REL_COMP	F(3)	ft

TABLE 4.- Concluded

SYMBOL	PRECISION	DEFINITION	TYPE	UNITS
$\bar{V}$_S_INER	TBD	Input Shuttle inertial velocity for REL_COMP	F(3)	ft/sec
$\bar{R}$_T_INER	TBD	Input target inertial position for REC_COMP	F(3)	ft
$\bar{V}$_T_INER	TBD	Input target inertial velocity for REL_COMP	F(3)	ft/sec
T_MAN	TBD	Time of the computed maneuver (MET)	F	sec

TABLE 5.- PROX_LAMVAR_COMMON

SYMBOL	PRECISION	DEFINITION	TYPE	UNITS
$\overline{RS}$_T1TIG	TBD	Inertial position vector of Shuttle at T1_TIG for PREVR	F(3)	ft
$\overline{VS}$_T1TIG	TBD	Inertial velocity vector of Shuttle at T1_TIG for PREVR	F(3)	ft/sec
$\overline{RS}$_T2TIG	TBD	Inertial position vector of Shuttle at T2_TIG for PREVR	F(3)	ft
S_OPTION	TBD	Spacecraft flag for UPDATVP	D	--
TTPI	TBD	Predicted time of TPI (MET)	F	sec
ALARM	TBD	Alarm flag to show error	I	--
$\overline{RS}$_OUT	TBD	Shuttle position vector	F(3)	ft
$\overline{VS}$_OUT	TBD	Shuttle velocity vector	F(3)	ft/sec
$\overline{RT}$_OUT	TBD	Target position vector	F(3)	ft
$\overline{VT}$_OUT	TBD	Target velocity vector	F(3)	ft/sec
TS_OUT	TBD	Time of the Shuttle position vector (GMT)	F	sec
TT_OUT	TBD	Time of the target position vector (GMT)	F	sec
$\overline{VS}$_REQUIRED	TBD	Velocity required at the maneuver time	F(3)	ft/sec
$\overline{RS}$_IPO	TBD	Initial position offset of Shuttle from PREVR	F(3)	ft
$\overline{RS}$_COM	TBD	Shuttle inertial position for COMELE	F(3)	ft
$\overline{VS}$_COM	TBD	Shuttle inertial velocity for COMELE	F(3)	ft
$\overline{RT}$_COM	TBD	Target inertial position for COMELE	F(3)	ft

TABLE 5.- Concluded

SYMBOL	PRECISION	DEFINITION	TYPE	UNITS
EL_ANG_COM	TBD	Computed elevation angle from COMELE	F	rad
$\overline{UN}$	TBD	Unit normal along the Shuttle angular momentum vector	F(3)	--
DEL_T_TRAN	TBD	Transfer time from PREVR	F	sec
IC	TBD	Iteration counter to ITERV	I	--
X_DEP	TBD	Dependent variable to ITERV	F	--
X_IND	TBD	Independent variable to ITERV	F	--
X_DEP_PRIME	TBD	Previous value of the dependent variable to ITERV	F	--
X_IND_PRIME	TBD	Previous value of the independent variable to ITERV	F	--
SFAIL	TBD	Flag set if maximum number of iterations have occurred	I	--
ERR	TBD	Dependent variable in search for elevation angle	F	--
ERR_PRIME	TBD	Previous value of the dependent variable in search for elevation angle	F	--
TTPI_PRIME	TBD	Previous value of the independent variable in search for elevation angle (MET)	F	sec
N	TBD	Iteration counter from ELITER	I	--
$\overline{RT}$	TBD	Input inertial position of target to ORBLV	F(3)	ft
$\overline{VT}$	TBD	Input inertial velocity of target to ORBLV	F(3)	ft/sec
$\overline{RS}$	TBD	Input inertial position of Shuttle to ORBLV	F(3)	ft
$\overline{VS}$	TBD	Input inertial velocity of Shuttle to ORBLV	F(3)	ft/sec
$\overline{VSLV}$	TBD	Shuttle-centered LVLH relative velocity of target	F(3)	ft/sec

TABLE 6.- INPUT PARAMETERS FOR THE PROX_STAT MODULE

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
TM_IND	TBD	Time GMT/MET select indicator		D	--
T_CURRENT	TBD	Current time	FCOS	F	sec
BASE_MET	TBD	GMT/MET reference time		F	sec
T_MAN	TBD	Time of the computed maneuver (MET)	PROX_TGT_SUP PROX_TGT_SUP_LAMB	F	sec
PROX_BASE_TIME	TBD	Prox ops base time	PROX_EXEC PROX_INIT PROX_TGT_SUP_LAMB	F	sec

TABLE 7.- OUTPUT PARAMETERS FOR THE PROX_STAT MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
PROX_PAST_STATUS	TBD	Maneuver past time status flag	PROX_TRANS PROX_STIME	D	--
PROX_T_CURRENT	TBD	Current MET time	PROX_EXEC PROX_TGT_SUP PROX_TGT_SUP_LAMB	F	sec

TABLE 8.- INPUT PARAMETERS FOR THE PROX_INIT MODULE

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
PROX_TGT_SET_NO	TBD	ILOAD set number selected for targeting	ORBIT_TGT_DIP (6.14)	I	--
DISP_PROX_DT	TBD	Displayed delta-t between T1 and T2	PROX_TGT_SEL PROX_DISP_LOAD ORBIT_TGT_DIP (6.14)	F	min
DISP_T1_X	TBD	Orbiter relative position at T1	PROX_TGT_SEL PROX_TGT_SUP PROX_TGT_SUP_LAMB	F(3)	ft
DISP_T1_XD	TBD	Orbiter relative velocity at T1	ORBIT_TGT_DIP (6.14)	F(3)	ft/sec
DISP_T1_DAY	TBD	T1 time	PROX_EXEC	I	day
DISP_T1_HR	TBD	T1 time	PROX_TGT_SEL	I	hr
DISP_T1_MIN	TBD	T1 time	PROX_DISP_LOAD	I	min
DISP_T1_SEC	TBD	T1 time	ORBIT_TGT_DIP (6.14)	F	sec
DISP_T2_DAY	TBD	T2 time		I	day
DISP_T2_HR	TBD	T2 time		I	hr
DISP_T2_MIN	TBD	T2 time	PROX_TGT_SEL PROX_DISP_LOAD ORBIT_TGT_DIP (6.14)	I	min
DISP_T2_SEC	TBD	T2 time		F	sec
DISP_T2_XOFF	TBD	Orbiter desired relative position at T2		F	ft
DISP_T2_YOFF	TBD	Orbiter desired relative position at T2	PROX_TGT_SEL ORBIT_TGT_DIP (6.14)	F	ft
DISP_T2_ZOFF	TBD	Orbiter desired relative position at T2		F	ft
DISP_EL_ANG	TBD	Displayed target elevation angle at T1		F	deg

TABLE 8.- Concluded

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
PROX_ITEM_21TO24_STATUS	TBD	Items 21 through 24 data entry status flag	PROX_EXEC ORBIT_TGT_DIP (6.14)	D	--
PROX_ITEM_2TO20_STATUS	TBD	Items 2 through 20 data entry status flag		D	--
PROX_BASE_DAY	TBD	Targeting base time		I	day
PROX_BASE_HR	TBD	Targeting base time	PROX_TGT_SUP_LAMB	I	hr
PROX_BASE_MIN	TBD	Targeting base time	ORBIT_TGT_DIP (6.14)	I	min
PROX_BASE_SEC	TBD	Targeting base time		F	sec
PROX_BASE_TIME	TBD	Prox ops base time	PROX_EXEC PROX_TGT_SUP_LAMB	F	sec
TIME_SEC	TBD	Conversion time	TIME_CONVRT	F	sec
PI	TBD	Mathematical constant	ILOAD	F	--

TABLE 9.- OUTPUT PARAMETERS FOR THE PROX_INIT MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
T1_ILOAD_ARRAY	TBD	T1 time for target sets (times are relative to prox ops base time)	TELEV PROX_TGT_SEL PROX_TGT_SUP_LAMB ILOAD	F(40)	min
EL_ILOAD_ARRAY	TBD	Maneuver elevation angle	PROX_TGT_SEL ILOAD	F(40)	rad
DT_ILOAD_ARRAY	TBD	Delta-t's from T1 maneuver to T2 maneuver for target sets	PROX_TGT_SEL OMEGA_DT_COMP ILOAD	F(40)	min
XOFF_ILOAD_ARRAY	TBD	ILOAD LVLH offset position for target sets	PROX_TGT_SEL OMEGA_DT_COMP ILOAD	F(40)	ft
YOFF_ILOAD_ARRAY	TBD	ILOAD LVLH offset position for target sets	PROX_TGT_SEL OMEGA_DT_COMP ILOAD	F(40)	ft
ZOFF_ILOAD_ARRAY	TBD	ILOAD LVLH offset position for target sets	PROX_TGT_SEL OMEGA_DT_COMP ILOAD	F(40)	ft
T1_TIG	TBD	T1 maneuver time (MET)	PROX_TGT_SUP PROX_TGT_SUP_LAMB PROX_DISP_LOAD DT_COMP PREVR	F	sec
T2_TIG	TBD	T2 maneuver time (MET)	PROX_TGT_SUP PROX_TGT_SUP_LAMB PROX_DISP_LOAD DT_COMP PREVR	F	sec
PROX_BASE_TIME	TBD	Prox Ops base time	PROX_STAT	F	sec
DAY	TBD	Converted time for TIMECV	TIME_CONVRT	I	day
HR	TBD	Converted time for TIMECV	TIME_CONVRT	I	hr
MIN	TBD	Converted time for TIMECV	TIME_CONVRT	I	min
SEC	TBD	Converted time for TIMECV	TIME_CONVRT	F	sec

TABLE 9.- Concluded

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
COMP_PROX_DT	TBD	Delta-t in the computation buffer	PROX_TGT_SUP PROX_TGT_SUP_LAMB PROX_DISP_LOAD DT_COMP	F	min
COMP_T2_XOFF	TBD	Computational desired position at T2		F	ft
COMP_T2_YOFF	TBD	Computational desired position at T2	PROX_TGT_SUP PROX_TGT_SUP_LAMB OMEGA_DT_COMP	F	ft
COMP_T2_ZOFF	TBD	Computational desired position at T2		F	ft
TIME_CONVERT_FLAG	TBD	Time conversion flag	TIME_CONVRT	D	--
TIME_PROX	TBD	Time tag for M50 state (assumed MET)	PROX_TGT_SUP PROX_TGT_SUP_LAMB TELEV	F	sec
EL_ANG	TBD	Desired elevation angle at TPI	PROX_TGT_SUP_LAMB TELEV	F	rad
COMP_$\bar{X}$	TBD	Computation relative position	PROX_TGT_SUP PROX_TGT_SUP_LAMB OMEGA_DT_COMP	F(3)	ft
COMP_$\bar{X}$D	TBD	Computation relative velocity	PROX_TGT_SUP PROX_TGT_SUP_LAMB	F(3)	ft/sec

TABLE 10.- INPUT PARAMETERS FOR THE PROX_DISP_LOAD MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
T_MAN	TBD	Time of the computed maneuver (MET)	PROX_TGT_SUP PROX_TGT_SUP_LAMB	F	sec
PROX_TGT_SET_NO	TBD	ILOAD set number selected for targeting	ORBIT_TGT_DIP (6.14)	I	--
T1_TIG	TBD	T1 maneuver time (MET)	PROX_EXEC PROX_TGT_SEL PROX_INIT PROX_TGT_SUP PROX_TGT_SUP_LAMB	F	sec
T2_TIG	TBD	T2 maneuver time (MET)	PROX_EXEC PROX_INIT PROX_TGT_SUP PROX_TGT_SUP_LAMB DT_COMP	F	sec
DAY	TBD	Converted time	TIME_CONVRT	I	day
HR	TBD	Converted time	TIME_CONVRT	I	hr
MIN	TBD	Converted time	TIME_CONVRT	I	min
SEC	TBD	Converted time	TIME_CONVRT	F	sec
DV_LVLH	TBD	Inpulsive maneuver in LVLH reference frame	PROX_TGT_SUP PROX_TGT_SUP_LAMB	F(3)	ft/sec

TABLE 11.- OUTPUT PARAMETERS FOR THE PROX_DISP_LOAD MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
DISP_PROX_DT	TBD	Displayed delta-t between T1 and T2	ORBIT_TGT_DIP (6.14)	F	min
DISP_T1_DAY	TBD	Displayed T1 time		I	day
DISP_T1_HR	TBD	Displayed T1 time		I	hr
DISP_T1_MIN	TBD	Displayed T1 time		I	min
DISP_T1_SEC	TBD	Displayed T1 time		F	sec
DISP_T2_DAY	TBD	Displayed T2 time		I	day
DISP_T2_HR	TBD	Displayed T2 time		I	hr
DISP_T2_MIN	TBD	Displayed T2 time		I	min
DISP_T2_SEC	TBD	Displayed T2 time		F	sec
DISP_$\vec{D}$V	TBD	Maneuver LVLH velocity vector		F(3)	ft/sec
DISP_DV_MAG	TBD	Maneuver velocity magnitude		F	ft/sec
DISP_TMAN_TIME	TBD	Maneuver time (MET)		F	sec
DISP_TMAN_DAY	TBD	Displayed maneuver time	PROX_TRANS	I	day
DISP_TMAN_HR	TBD	Displayed maneuver time		I	hr
DISP_TMAN_MIN	TBD	Displayed maneuver time		I	min
DISP_TMAN_SEC	TBD	Displayed maneuver time		F	sec
TIME_CONVERT_FLAG	TBD	Time conversion flag	TIME_CONVRT	I	--
TIME_SEC	TBD	Converted time		F	sec

TABLE 12.- INPUT PARAMETERS FOR THE PROX_STIME MODULE

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
TM_IND	TBD	Time GMT/MET select indicator	FCOS	D	--
BASE_MET	TBD	GMT/MET reference time	FCOS	F	sec
T_MAN	TBD	Time of the computed maneuver (MET)	PROX_TGT_SUP PROX_TGT_SUP_LAMB	F	sec
PROX_PAST_STATUS	TBD	Maneuver past time status flag	PROX_STAT	D	--

TABLE 13.- OUTPUT PARAMETERS FOR THE PROX_STIME MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
DT_ST_TIMER	TBD	Time to be counted down		F	sec
CRT_TIME	TBD	CRT timer start/stop discrete	ORBIT_TGT_DIP (6.14)	D	--
CRT_UPDWN	TBD	CRT timer up/down discrete		D	--

TABLE 14.- INPUT PARAMETERS FOR THE PROX_TRANS MODULE

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
GUID_FLAG	TBD	Current maneuver guidance option flag	PROX_TGT_SUP PROX_TGT_SUP_LAMB	D	--
PROX_PAST_STATUS	TBD	Maneuver past time status flag	PROX_STAT	D	--
$\overline{\text{DV}}$_LVLH	TBD	Impulsive maneuver in LVLH reference frame	PROX_TGT_SUP PROX_TGT_SUP_LAMB	F(3)	ft/sec
S_ROTATE	TBD	Rotation flag for near 180° transfers		D	--
$\overline{\text{R}}$_OFFSET	TBD	Offset position vector	PREVR	F(3)	ft
T_OFFSET	TBD	Time of offset position vector (MET)		F	sec
DISP_TMAN_DAY	TBD			I	day
DISP_TMAN_HR	TBD	Maneuver time for display	PROX_DISP_LOAD	I	hr
DISP_TMAN_MIN	TBD			I	min
DISP_TMAN_SEC	TBD			F	sec

TABLE 15.- OUTPUT PARAMETERS FOR THE PROX_TRANS MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
DELTA_$\bar{V}$_LVLH_GUID	TBD	Delta-v vector of current maneuver	ORBIT_MNVR_DISP (6.35)	F(3)	ft/sec
S_ROTATE_GUID	TBD	Rotation flag for near-180° transfers	ORBIT_MNVR_DIP (4.158)	D	--
$\bar{R}$_OFFSET_GUID	TBD	Offset position vector	ORBIT_MNVR_DIP (4.158)	F(3)	ft
T_OFFSET_GUID	TBD	Time of offset position vector (MET)	ORBIT_MNVR_DIP (4.158)	F	sec
TIG_GUID_DAY	TBD	Ignition time of current maneuver	ORBIT_MNVR_DISP (6.35)	I	day
TIG_GUID_HR	TBD	Ignition time of current maneuver	ORBIT_MNVR_DISP (6.35)	I	hr
TIG_GUID_MIN	TBD	Ignition time of current maneuver	ORBIT_MNVR_DISP (6.35)	I	min
TIG_GUID_SEC	TBD	Ignition time of current maneuver	ORBIT_MNVR_DISP (6.35)	F	sec

TABLE 16.- INPUT PARAMETERS FOR THE PROX_EXEC MODULE

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
BASE_MET	TBD	GMT/MET reference time	FCOS	F	sec
T_STATE	TBD	Time tag for M50 vectors	ON_ORB_UPP (4.22)	F	sec
$\vec{R}$_AVGG	TBD	Orbiter position M50 vectors	ON_ORB_UPP (4.22)	F(3)	ft
$\vec{V}$_AVGG	TBD	Orbiter veloctiy M50 vectors	ON_ORB_UPP (4.22)	F(3)	ft/sec
$\vec{R}$_TARGET	TBD	Target position M50 vectors	ON_ORB_UPP (4.22)	F(3)	ft
$\vec{V}$_TARGET	TBD	Target velocity M50 vectors	ON_ORB_UPP (4.22)	F(3)	ft/sec
PROX_TGT_SET_NO	TBD	ILOAD set number selected for trajectory	ORBIT_TGT_DIP (6.14)	I	--
PROX_ITEM_21TO24_STATUS	TBD	Items 21 through 24 data entry status flag	PROX_INIT ORBIT_TGT_DIP (6.14)	D	--
PROX_ITEM_2TO20_STATUS	TBD	Items 2 through 20 data entry status flag		D	--
PROX_ITEM_1_STATUS	TBD	Item 1 data entry status flag		D	--
PROX_ITEM_25_STATUS	TBD	Item 25 data entry status flag		D	--
PROX_ITEM_26_STATUS	TBD	Item 26 data entry status flag	ORBIT_TGT_DIP (6.14)	D	--
PROX_ITEM_27_STATUS	TBD	Item 27 data entry status flag		D	--
PROX_ITEM_28_STATUS	TBD	Item 28 data entry status flag		D	--

TABLE 16.- Concluded

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
PROX_FIRST_PASS_STATUS	TBD	Orbit tgt exec first pass status flag		D	--
BASE_START_DAY	TBD	Referenced base time for T1_ILOAD_ARRAY		I	day
BASE_START_HR	TBD	Referenced base time for T1_ILOAD_ARRAY	ILOAD	I	hr
BASE_START_MIN	TBD	Referenced base time for T1_ILOAD_ARRAY		I	min
BASE_START_SEC	TBD	Referenced base time for T1_ILOAD_ARRAY		F	sec
NLAMB	TBD	Number of Lambert targeted target sets		I	--
PROX_T_CURRENT	TBD	Current MET time	PROX_STAT	F	sec
DAY	TBD	Converted time for TIMECV		I	day
HR	TBD	Converted time for TIMECV	TIME_CONVRT	I	hr
MIN	TBD	Converted time for TIMECV		I	min
SEC	TBD	Converted time for TIMECV		F	sec
TIME_SEC	TBD	Conversion time	TIME_CONVRT	F	sec

TABLE 17.- OUTPUT PARAMETERS FOR THE PROX_EXEC MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
PROX_T1_STAR_STATUS	TBD	T1 maneuver compute star flag	PROX_TGT_SUP PROX_TGT_SUP_LAMB ORBIT_TGT_DIP (6.14)	D	--
PROX_T2_STAR_STATUS	TBD	T2 maneuver compute star flag		D	--
DISP_T1_DAY	TBD	T1 time	PROX_INIT ORBIT_TGT_DIP (6.14)	I	day
DISP_T1_HR	TBD	T1 time		I	hr
DISP_T1_MIN	TBD	T1 time		I	min
DISP_T1_SEC	TBD	T1 time		F	sec
PROX_ITEM_21TO24_STATUS	TBD	Items 21 through 24 data entry status flag	PROX_INIT	D	--
PROX_ITEM_2TO20_STATUS	TBD	Items 2 through 20 data entry status flag		D	--
PROX_LOAD_FLASH	TBD	"LOAD" flash status flag	ORBIT_TGT_DIP (6.14)	D	--
PROX_FIRST_PASS_STATUS	TBD	Orbit tgt exec first pass status flag	ORBIT_TGT_DIP (6.14)	D	--

TABLE 17.- Concluded

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
T1_TIG	TBD	T1 maneuver time (MET)	PROX_TGT_SUP PROX_TGT_SUP_LAMB PROX_DISP_LOAD DT_COMP PREVR	F	sec
T2_TIG	TBD	T2 maneuver time (MET)		F	sec
PROX_BASE_TIME	TBD	Prox ops base time	PROX_STAT PROX_INIT	F	sec
DAY	TBD	Converted time for TIMECV	TIME_CONVRT	I	day
HR	TBD	Converted time for TIMECV		I	hr
MIN	TBD	Converted time for TIMECV		I	min
SEC	TBD	Converted time for TIMECV		F	sec
TIME_CONVERT_FLAG	TBD	Time conversion flag		D	--
TIME_SEC	TBD	Conversion time		F	sec
TIME_PROX	TBD	Time tag for M50 state (assumed MET)	PROX_TGT_SUP PROX_TGT_SUP_LAMB TELEV	F	sec
USE_DISP_REL_STATE	TBD	Use displayed relative state flag	PROX_TGT_SUP PROX_TGT_SUP_LAMB	D	--
OMEGA_PROX	TBD	Orbital angular rate of target	OFFSET_TGT REL_PRED REL_COMP OMEGA_DT_COMP	F	rad/sec
$\bar{R}$S_M50_PROX	TBD	Prox ops Orbiter M50 position vector	PROX_TGT_SUP PROX_TGT_SUP_LAMB TELEV	F(3)	ft
$\bar{V}$S_M50_PROX	TBD	Prox ops Orbiter M50 velocity vector		F(3)	ft/sec
$\bar{R}$T_M50_PROX	TBD	Prox ops target M50 position vector		F(3)	ft
$\bar{V}$T_M50_PROX	TBD	Prox ops target M50 velocity vector		F(3)	ft/sec

TABLE 18.- INPUT PARAMETERS FOR THE PROX_TGT_SEL MODULE

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
PROX_TGT_SET_NO	TBD	ILOAD set number selected for targeting	ORBIT_TGT_DIP (6.14)	I	--
T1_ILOAD_ARRAY	TBD	T1 time for target sets		F(40)	min
DT_ILOAD_ARRAY	TBD	Delta-t from T1 maneuver to T2 maneuver for target sets		F(40)	min
EL_ILOAD_ARRAY	TBD	Maneuver elevation angle	PROX_INIT ILOAD	F(40)	rad
XOFF_ILOAD_ARRAY	TBD	ILOAD LVLH offset position for target sets		F(40)	ft
YOFF_ILOAD_ARRAY	TBD	ILOAD LVLH offset position for target sets		F(40)	ft
ZOFF_ILOAD_ARRAY	TBD	ILOAD LVLH offset position for target sets		F(40)	ft
DAY	TBD	Converted time for TIMECV		I	day
HR	TBD	Converted time for TIMECV	TIME_CONVRT	I	hr
MIN	TBD	Converted time for TIMECV		I	min
SEC	TBD	Converted time for TIMECV		F	sec
PI	TBD	Mathematical constant	ILOAD	F	--

TABLE 19.- OUTPUT PARAMETERS FOR THE PROX_TGT_SEL MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
DISP_PROX_DT	TBD	Displayed delta-t between T1 and T2		F	min
DISP_T1_$\bar{X}$	TBD	Orbiter relative position at T1		F(3)	ft
DISP_T1_$\bar{X}$D	TBD	Orbiter relative velocity at T1		F(3)	ft/sec
DISP_T1_DAY	TBD	T1 time		I	day
DISP_T1_HR	TBD	T1 time		I	hr
DISP_T1_MIN	TBD	T1 time		I	min
DISP_T1_SEC	TBD	T1 time		F	sec
DISP_T2_DAY	TBD	T2 time	PROX_INIT ORBIT_TGT_DIP (6.14)	I	day
DISP_T2_HR	TBD	T2 time		I	hr
DISP_T2_MIN	TBD	T2 time		I	min
DISP_T2_SEC	TBD	T2 time		F	sec
DISP_T2_XOFF	TBD	Orbiter desired relative position at T2		F	ft
DISP_T2_YOFF	TBD	Orbiter desired relative position at T2		F	ft
DISP_T2_ZOFF	TBD	Orbiter desired relative position at T2		F	ft
DISP_EL_ANG	TBD	Displayed target elevation angle at T1		F	deg
T1_TIG	TBD	T1 maneuver time (MET)	PROX_TGT_SUP PROX_TGT_SUP_LAMB PROX_DISP_LOAD DT_COMP PREVR	F	sec

TABLE 19.- Concluded

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
COMP_PROX_DT	TBD	Delta-t in the computation buffer	PROX_TGT_SUP PROX_TGT_SUP_LAMB PROX_DISP_LOAD DT_COMP	F	min
COMP_T2_XOFF	TBD	Computational desired position at T2		F	ft
COMP_T2_YOFF	TBD	Computational desired position at T2	PROX_TGT_SUP PROX_TGT_SUP_LAMB OMEGA_DT_COMP	F	ft
COMT_T2_ZOFF	TBD	Computational desired position at T2		F	ft
TIME_CONVERT_FLAG	TBD	Time conversion flag	TIME_CONVRT	D	--
TIME_SEC	TBD	Conversion time		F	sec
EL_ANG	TBD	Desired elevation angle at TPI	PROX_TGT_SUP_LAMB TELEV	F	rad

TABLE 20.- INPUT PARAMETERS FOR THE PROX_TGT_SUP MODULE

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
PROX_T1_STAR_STATUS	TBD	T1 maneuver compute star flag	PROX_EXEC ORBIT_TGT_DIP (6.14)	D	--
PROX_DTMIN	TBD	Tolerance on minimum time in the future to compute a prox ops maneuver solution	ILOAD	F	sec
PROX_DTMIN_LAMB	TBD	Tolerance on minimum time in the future to compute a Lambert maneuver solution	ILOAD	F	sec
T1_TIG	TBD	T1 maneuver time (MET)	PROX_EXEC PROX_TGT_SEL PROX_INIT PROX_TGT_SUP_LAMB	F	sec
T2_TIG	TBD	T2 maneuver time (MET)	PROX_EXEC PROX_INIT PROX_TGT_SUP_LAMB DT_COMP	F	sec
PROX_T_CURRENT	TBD	Current MET time	PROX_STAT	F	sec
COMP_PROX_DT	TBD	Delta-t in the computation buffer	PROX_TGT_SEL PROX_INIT PROX_TGT_SUP_LAMB DT_COMP OMEGA_DT_COMP	F	min
COMP_T2_XOFF	TBD	Computational desired position at T2		F	ft
COMP_T2_YOFF	TBD	Computational desired position at T2	PROX_TGT_SEL PROX_INIT PROX_TGT_SUP_LAMB	F	ft
COMP_T2_ZOFF	TBD	Computational desired position at T2		F	ft
TIME_PROX	TBD	Time tag for M50 state (assume MET)	PROX_EXEC PROX_INIT	F	sec

TABLE 20.- Concluded

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
$\bar{X}$2	TBD	Predicted Orbiter relative state position	REL_PRED	F(3)	ft
$\bar{X}$D2	TBD	Predicted Orbiter relative state velocity	REL_PRED	F(3)	ft/sec
USE_OMEGA_DT	TBD	Use the WT calculation flag	PROX_TGT_SUP_LAMB DT_COMP	D	--
USE_DISP_REL_STATE	TBD	Use displayed relative state flag	PROX_EXEC PROX_TGT_SUP_LAMB	D	--
COMP_$\bar{X}$	TBD	Computation relative position	PROX_INIT PROX_TGT_SUP_LAMB	F(3)	ft
COMP_$\bar{X}$D	TBD	Computation relative velocity	PROX_INIT PROX_TGT_SUP_LAMB	F(3)	ft/sec
$\bar{R}$S_M50_PROX	TBD	Prox ops Orbiter M50 position vector	PROX_EXEC	F(3)	ft
$\bar{R}$T_M50_PROX	TBD	Prox ops target M50 position vector	PROX_EXEC	F(3)	ft
$\bar{V}$S_M50_PROX	TBD	Prox ops Orbiter M50 velocity vector	PROX_EXEC	F(3)	ft/sec
$\bar{V}$T_M50_PROX	TBD	Prox ops target M50 velocity vector	PROX_EXEC	F(3)	ft/sec
$\bar{R}$_REL	TBD	LVLH curvilinear position vector from REL_COMP	REL_COMP	F(3)	ft
$\bar{V}$_REL	TBD	LVLH curvilinear velocity vector from REL_COMP	REL_COMP	F(3)	ft/sec
$\bar{D}$V	TBD	Delta-v vector from OFFSET_TGT	OFFSET_TGT	F(3)	ft/sec

TABLE 21.- OUTPUT PARAMETERS FOR THE PROX_TGT_SUP MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
GUID_FLAG	TBD	Current maneuver guidance option flag	ORBIT_MNVR_DIP (4.158) PROX_TRANS	D	--
T_MAN	TBD	Time of the computed maneuver (MET)	PROX_STAT PROX_TRANS PROX_STIME PROX_DISP_LOAD	F	sec
DISP_T1_$\overline{X}$	TBD	Orbiter relative position at T1	PROX_INIT ORB_TGT_DIP (6.14)	F(3)	ft
DISP_T1_$\overline{XD}$	TBD	Orbiter relative velocity at T1		F(3)	ft/sec
T1_TIG	TBD	T1 maneuver time (MET)	PROX_DISP_LOAD DT_COMP	F	sec
T2_TIG	TBD	T2 maneuver time (MET)	PREVR	F	sec
COMP_PROX_DT	TBD	Delta-t in the computation buffer	PROX_TGT_SUP_LAMB PROX_DISP_LOAD DT_COMP	F	min
COMP_T2_XOFF	TBD	Computational desired position at T2		F	ft
COMP_T2_YOFF	TBD	Computational desired position at T2	PROX_TGT_SUP_LAMB OMEGA_DT_COMP	F	ft
COMP_T2_ZOFF	TBD	Computational desired position at T2		F	ft
DV_LVLH	TBD	Impulsive maneuver in LVLH reference frame	PROX_TRANS PROX_DISP_LOAD	F(3)	ft/sec
USE_DISP_REL_STATE	TBD	Use displayed relative state flag	PROX_TGT_SUP_LAMB	D	--
COMP_$\overline{X}$	TBD	Computation relative position	PROX_TGT_SUP_LAMB OMEGA_DT_COMP	F(3)	ft

TABLE 21.- Concluded

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
COMP_X̄D	TBD	Computation relative velocity	PROX_TGT_SUP_LAMB	F(3)	ft/sec
DTIME	TBD	Time internal for prediction computation	REL_PRED	F	min
X̄	TBD	Input relative position for the relative predict task	REL_PRED	F(3)	ft
X̄D	TBD	Input relative velocity for the relative predict task	REL_PRED	F(3)	ft/sec
INER_TO_LVC	TBD	Inertial to curvilinear conversion flag	REL_COMP	D	--
DT_OFFTGT	TBD	Transfer time for OFFSET_TGT		F	sec
X̄_OFFTGT	TBD	T1 relative position vector for OFFSET_TGT	OFFSET_TGT	F(3)	ft
X̄D_OFFTGT	TBD	T1 relative velocity vector for OFFSET_TGT		F(3)	ft/sec
X̄2_OFFTGT	TBD	T2 relative position vector for OFFSET_TGT		F(3)	ft
R̄_S_INER	TBD	Input Shuttle inertial position for REL_COMP		F(3)	ft
V̄_S_INER	TBD	Input Shuttle inertial velocity for REL_COMP	REL_COMP	F(3)	ft/sec
R̄_T_INER	TBD	Input target inertial position for REL_COMP		F(3)	ft
V̄_T_INER	TBD	Input target inertial velocity for REL_COMP		F(3)	ft/sec

TABLE 22.- INPUT PARAMETERS FOR THE PROX_TGT_SUP_LAMB MODULE

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
BASE_MET	TBD	GMT/MET reference time	FCOS	F	sec
PROX_T1_STAR_STATUS	TBD	T1 maneuver compute star flag	PROX_EXEC	D	--
PROX_TGT_SET_NO	TBD	ILOAD set number selected for targeting	ORBIT_TGT_DIP (6.14)	I	--
T1_TIG	TBD	T1 maneuver time (MET)	PROX_EXEC PROX_TGT_SEL PROX_INIT PROX_TGT_SUP	F	sec
T2_TIG	TBD	T2 maneuver time (MET)	PROX_EXEC PROX_INIT PROX_TGT_SUP DT_COMP	F	sec
PROX_T_CURRENT	TBD	Current MET time	PROX_STAT	F	sec
DAY	TBD	Converted time for TIMECV		I	day
HR	TBD	Converted time for TIMECV	TIME_CONVRT	I	hr
MIN	TBD	Converted time for TIMECV		I	min
SEC	TBD	Converted time for TIMECV		F	sec
COMP_PROX_DT	TBD	Delta-t in the computation buffer	PROX_TGT_SEL PROX_INIT PROX_TGT_SUP DT_COMP OMEGA_DT_COMP	F	min

TABLE 22.- Continued

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
COMP_T2_XOFF	TBD	Computational desired position at T2	PROX_TGT_SEL PROX_INIT PROX_TGT_SUP	F	ft
COMP_T2_YOFF	TBD	Computational desired position at T2	PROX_TGT_SEL PROX_INIT PROX_TGT_SUP	F	ft
COMP_T2_ZOFF	TBD	Computational desired position at T2	PROX_TGT_SEL PROX_INIT PROX_TGT_SUP	F	ft
TIME_PROX	TBD	Time tag for the M50 state (assumed MET)	PROX_EXEC PROX_INIT	F	sec
EL_ANG	TBD	Desired elevation angle at TPI	PROX_TGT_SEL PROX_INIT	F	rad
USE_OMEGA_DT	TBD	Use the WT calculation flag	DT_COMP	D	--
USE_DISP_REL_STATE	TBD	Use displayed relative flag	PROX_EXEC PROX_TGT_SUP	D	--
COMP_$\overline{X}$	TBD	Computational relative position	PROX_INIT PROX_TGT_SUP	F(3)	ft
COMP_$\overline{XD}$	TBD	Computational relative velocity	PROX_INIT PROX_TGT_SUP	F(3)	ft/sec
$\overline{RS}$_M50_PROX	TBD	Prox ops Orbiter M50 position vector	PROX_EXEC	F(3)	ft
$\overline{VS}$_M50_PROX	TBD	Prox ops Orbiter M50 velocity vector	PROX_EXEC	F(3)	ft/sec
$\overline{RT}$_M50_PROX	TBD	Prox ops target M50 position vector	PROX_EXEC	F(3)	ft
$\overline{VT}$_M50_PROX	TBD	Prox ops target M50 velocity vector	PROX_EXEC	F(3)	ft/sec

TABLE 22.- Concluded

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
$\bar{R}$_OUT	TBD	Output inertial position vector	UPDATVP	F(3)	ft
$\bar{V}$_OUT	TBD	Output inertial veclocity vector	UPDATVP	F(3)	ft/sec
TTPI	TBD	Predicted time of TPI (MET)	TELEV ELITER	F	sec
$\bar{RS}$_OUT	TBD	Shuttle position vector	TELEV ELITER	F(3)	ft
$\bar{VS}$_OUT	TBD	Shuttle velocity vector	TELEV ELITER	F(3)	ft/sec
$\bar{RT}$_OUT	TBD	Target position vector	TELEV ELITER	F(3)	ft
$\bar{VT}$_OUT	TBD	Target velocity vector	TELEV ELITER	F(3)	ft/sec
TS_OUT	TBD	Time of the Shuttle position vector (GMT)	TELEV ELITER	F	sec
TT_OUT	TBD	Time of the target position vector (GMT)	TELEV ELITER	F	sec
$\bar{VS}$_REQUIRED	TBD	Velocity required at the maneuver time	PREVR LAMBERT	F(3)	ft/sec
$\bar{RS}$_IPO	TBD	Initial position offset of Shuttle	PREVR	F(3)	ft
$\bar{R}$_REL	TBD	LVLH curvilinear position vector	REL_COMP	F(3)	ft
$\bar{V}$_REL	TBD	LVLH curvilinear velocity vector	REL_COMP	F(3)	ft/sec
$\bar{R}$_S_INER	TBD	Input Shuttle inertial position vector	REL_COMP	F(3)	ft
$\bar{V}$_S_INER	TBD	Input Shuttle inertial velocity vector	REL_COMP	F(3)	ft/sec
$\bar{VSLV}$	TBD	Shuttle-centered LVLH relative velocity of the target	ORBLV	F(3)	ft/sec

TABLE 23.- OUTPUT PARAMETERS FOR THE PROX_TGT_SUP_LAMB MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
GUID_FLAG	TBD	Current maneuver guidance option flag	PROX_TRANS ORBIT_MNVR_DIP (4.158)	I	--
T_MAN	TBD	Ignition time of maneuver (MET)	PROX_STAT PROX_TRANS PROX_STIME PROX_DISP_LOAD	F	sec
DISP_T1_$\bar{X}$	TBD	Orbiter relative position at T1	PROX_INIT ORB_TGT_DIP (6.14)	F(3)	ft
DISP_T1_$\bar{XD}$	TBD	Orbiter relative velocity at T1		F(3)	ft/sec
PROX_BASE_DAY	TBD	Targeting base time	PROX_INIT ORB_TGT_DIP (6.14)	I	day
PROX_BASE_HR	TBD	Targeting base time		I	hr
PROX_BASE_MIN	TBD	Targeting base time		I	min
PROX_BASE_SEC	TBD	Targeting base time		F	sec
T1_TIG	TBD	T1 maneuver time (MET)	PROX_TGT_SUP PROX_DISP_LOAD DT_COMP PREVR	F	sec
T2_TIG	TBD	T2 maneuver time (MET)		F	sec
PROX_BASE_TIME	TBD	Prox ops base time	PROX_STAT PROX_INIT	F	sec
COMP_PROX_DT	TBD	Delta-t in the computation buffer	PROX_DISP_LOAD DT_COMP	F	min
COMP_T2_XOFF	TBD	Computational desired position at T2	OMEGA_DT_COMP	F	ft
COMP_T2_YOFF	TBD	Computational desired position at T2		F	ft
COMP_T2_ZOFF	TBD	Computational desired position at T2		F	ft

TABLE 23.- Continued

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
TIME_CONVERT_FLAG	TBD	Time conversion flag	TIME_CONVRT	D	--
TIME_SEC	TBD	Conversion time	TIME_CONVRT	F	sec
DV_LVLH	TBD	Inpulsive maneuver in LVLH reference frame	PROX_TRANS PROX_DISP_LOAD	F(3)	ft/sec
EL_ANG	TBD	Desired elevation angle	TELEV	F	rad
USE_DISP_REL_STATE	TBD	Use display relative state flag	PROX_TGT_SUP	D	--
COMP_X	TBD	Computation relative position	PROX_TGT_SUP OMEGA_DT_COMP	F(3)	ft
COMP_XD	TBD	Computation relative velocity	PROX_TGT_SUP	F(3)	ft/sec
INER_TO_LVC	TBD	Inertial to curvilinear conversion flag	REL_COMP	D	--
R_S_INER	TBD	Input Shuttle inertial position	REL_COMP	F(3)	ft
V_S_INER	TBD	Input Shuttle inertial velocity	REL_COMP	F(3)	ft/sec
R_T_INER	TBD	Input target inertial position	REL_COMP	F(3)	ft
V_T_INER	TBD	Input target inertial velocity	REL_COMP	F(3)	ft/sec

TABLE 23.- Concluded

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
$\bar{R}$S_T1TIG	TBD	Inertial position vector of Shuttle at T1		F(3)	ft
$\bar{V}$S_T1TIG	TBD	Inertial velocity vector of Shuttle at T1	PREVR	F(3)	ft/sec
$\bar{R}$S_T2TIG	TBD	Inertial position vector of Shuttle at T2		F(3)	ft
$\bar{R}$_IN	TBD	Input inertial position vector		F(3)	ft
$\bar{V}$_IN	TBD	Input inertial velocity	UPDATVP	F(3)	ft/sec
S_OPTION	TBD	Spacecraft flag in UPDATVP		D	--
T_IN	TBD	Time of input state in GMT	UPDATVP	F	sec
T_OUT	TBD	Time of output state in GMT	UPDATVP	F	sec
$\bar{R}$_REL	TBD	LVLH curvilinear position vector	REL_COMP	F(3)	ft
$\bar{V}$_REL	TBD	LVLH curvilinear velocity vector		F(3)	ft/sec
T1_ILOAD_ARRAY	TBD	Time for the target set relative to base time (MET)	PROX_TGT_SEL TELEV ILOAD	F(40)	min
$\bar{R}$T	TBD	Input inertial position of target	ORBLV	F(3)	ft
$\bar{V}$T	TBD	Input inertial velocity of target	ORBLV	F(3)	ft/sec
$\bar{R}$S	TBD	Input inertial position of Shuttle	ORBLV	F(3)	ft
$\bar{V}$S	TBD	Input inertial velocity of Shuttle	ORBLV	F(3)	ft/sec

TABLE 24.- INPUT PARAMETERS FOR THE OFFSET_TGT MODULE

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
OMEGA_PROX	TBD	Orbital angular rate of target	PROX_EXEC	F	rad/sec
DT_OFFTGT	TBD	Transfer time		F	min
$\bar{X}$_OFFTGT	TBD	T1 relative position vector		F(3)	ft
$\bar{X}$D_OFFTGT	TBD	T1 relative velocity vector	PROX_TGT_SUP OMEGA_DT_COMP	F(3)	ft/sec
$\bar{X}$2_OFFTGT	TBD	T2 relative position vector		F(3)	ft

TABLE 25.- OUTPUT PARAMETERS FOR THE OFFSET_TGT MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
$\bar{D}$V	TBD	Delta-v vector	PROX_TGT_SUP OMEGA_DT_COMP	F(3)	ft/sec

TABLE 26.- INPUT PARAMETERS FOR THE REL_PRED MODULE

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
DTIME	TBD	Time interval for prediction computation	PROX_TGT_SUP	F	min
OMEGA_PROX	TBD	Orbital angular rate of target	PROX_EXEC	F	rad/sec
$\overline{X}$	TBD	Input relative position	PROX_TGT_SUP	F(3)	ft
$\overline{X}D$	TBD	Input relative velocity	PROX_TGT_SUP	F(3)	ft/sec

TABLE 27.- OUTPUT PARAMETERS FOR THE REL_PRED MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
$\bar{X}2$	TBD	Predicted Orbiter relative state position	PROX_TGT_SUP	F(3)	ft
$\bar{X}D2$	TBD	Predicted Orbiter relative state velocity	PROX_TGT_SUP	F(3)	ft/sec

TABLE 28.- INPUT PARAMETERS FOR THE REL_COMP MODULE

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
OMEGA_PROX	TBD	Orbital angular rate of target	PROX_EXEC	F	rad/sec
INER_TO_LVC	TBD	Inertial to curvilinear conversion flag	PROX_TGT_SUP PROX_SUP_LAMB	D	--
$\overline{R}$_REL	TBD	LVLH curvilinear position vector	PROX_TGT_SUP_LAMB	F(3)	ft/sec
$\overline{V}$_REL	TBD	LVLH curvilinear velocity vector		F(3)	ft/sec
$\overline{R}$_S_INER	TBD	Input Shuttle inertial position vector	PROX_TGT_SUP_LAMB	F(3)	ft
$\overline{V}$_S_INER	TBD	Input Shuttle inertial velocity vector		F(3)	ft/sec
$\overline{R}$_T_INER	TBD	Input target inertial position vector	PROX_TGT_SUP PROX_TGT_SUP_LAMB	F(3)	ft
$\overline{V}$_T_INER	TBD	Input target inertial velocity vector		F(3)	ft/sec

TABLE 29.- OUTPUT PARAMETERS FOR THE REL_COMP MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
$\bar{R}$_REL	TBD	LVLH curvilinear position vector	PROX_TGT_SUP PROX_TGT_SUP_LAMB	F(3)	ft
$\bar{V}$_REL	TBD	LVLH curvilinear velocity vector	PROX_TGT_SUP PROX_TGT_SUP_LAMB	F(3)	ft/sec
$\bar{R}$_S_INER	TBD	Input Shuttle inertial position	PROX_TGT_SUP_LAMB	F(3)	ft
$\bar{V}$_S_INER	TBD	Input Shuttle inertial velocity	PROX_TGT_SUP_LAMB	F(3)	ft/sec

TABLE 30.- INPUT PARAMETERS FOR THE TIME_CONVRT MODULE

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
DAY	TBD	Converted time for TIME_CONVRT		I	day
HR	TBD	Converted time for TIME_CONVRT	PROX_EXEC PROX_TGT_SEL	I	hr
MIN	TBD	Converted time for TIME_CONVRT		I	min
SEC	TBD	Converted time for TIME_CONVRT		F	sec
TIME_CONVERT_FLAG	TBD	Time conversion flag	PROX_EXEC PROX_TGT_SEL PROX_INIT PROX_TGT_SUP_LAMB PROX_DISP_LOAD	I	--
TIME_SEC	TBD	Converted time	PROX_EXEC PROX_TGT_SEL PROX_TGT_SUP_LAMB PROX_DISP_LOAD	F	sec

TABLE 31.- OUTPUT PARAMETERS FOR THE TIME_CONVRT MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
DAY	TBD	Converted time for TIME_CONVRT		I	day
HR	TBD	Converted time for TIME_CONVRT	PROX_EXEC PROX_TGT_SEL PROX_TGT_SUP_LAMB	I	nr
MIN	TBD	Converted time for TIME_CONVRT	PROX_DISP_LOAD	I	min
SEC	TBD	Converted time for TIME_CONVRT		F	sec
TIME_SEC	TBD	Conversion time	PROX_EXEC PROX_INIT	F	sec

TABLE 32.- INPUT PARAMETERS FOR THE DT_COMP MODULE

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
T1_TIG	TBD	T1 maneuver time (MET)	PROX_TGT_SUP PROX_TGT_SUP_LAMB	F	sec
T2_TIG	TBD	T2 maneuver time (MET)	PROX_TGT_SUP PROX_TGT_SUP_LAMB	F	sec
COMP_PROX_DT	TBD	Delta-t in the computation buffer	PROX_TGT_SUP PROX_TGT_SUP_LAMB	F	min

TABLE 33.- OUTPUT PARAMETERS FOR THE DT_COMP MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
T2_TIG	TBD	T2 maneuver time (MET)	PROX_TGT_SUP PROX_TGT_SUP_LAMB	F	sec
COMP_PROX_DT	TBD	Delta-t in the computation buffer		F	min

TABLE 34.- INPUT PARAMETERS FOR THE OMEGA_DT_COMP MODULE

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
PROX_TGT_SET_NO	TBD	ILOAD set number selected for targeting	ORBIT_TGT_DIP (6.14)	I	--
DT_ILOAD_ARRAY	TBD	Delta-t's from T1 to T2 for target sets		F(40)	min
EL_ILOAD_ARRAY	TBD	Maneuver elevation angle		F(40)	rad
XOFF_ILOAD_ARRAY	TBD	ILOAD LVLH offset position for target sets	PROX_INIT ILOAD	F(40)	ft
YOFF_ILOAD_ARRAY	TBD	ILOAD LVLH offset position for target sets		F(40)	ft
ZOFF_ILOAD_ARRAY	TBD	ILOAD LVLH offset position for target sets		F(40)	ft
COMP_T2_XOFF	TBD	Computational desired at T2		F	ft
COMP_T2_YOFF	TBD	Computational desired at T2	PROX_TGT_SUP PROX_TGT_SUP_LAMB	F	ft
COMP_T2_ZOFF	TBD	Computational desired at T2		F	ft
COMP_$\overline{X}$	TBD	Computation relative position		F(3)	ft
OMEGA_PROX	TBD	Orbital angular rate of target	PROX_EXEC	F	rad/sec
$\overline{DV}$	TBD	Delta-v vector	OFFSET_TGT	F(3)	ft/sec

TABLE 35.- OUTPUT PARAMETERS FOR THE OMEGA_DT_COMP MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
COMP_PROX_DT	TBD	Delta-t in the computation buffer	PROX_TGT_SUP PROX_TGT_SUP_LAMB	F	min
DT_OFFTGT	TBD	Transfer time		F	sec
$\bar{X}$_OFFTGT	TBD	T1 relative position vector		F(3)	ft
$\bar{X}$D_OFFTGT	TBD	T1 relative position vector	OFFSET_TGT	F(3)	ft/sec
$\bar{X}$2_OFFTGT	TBD	T2 relative position vector		F(3)	ft

TABLE 36.- INPUT PARAMETERS FOR THE TELEV MODULE

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
BASE_MET	TBD	GMT/MET reference time	FCOS	F	sec
PI	TBD	Mathematical constant			
T1_ILOAD_ARRAY	TBD	Nominal time of maneuver for targets		F(40)	min
EL_TOL	TBD	Tolerance between computed and desired elevation angles	ILOAD	F	rad
EL_DH_TOL	TBD	Elevation angle-differential altitude incompatability tolerance		F	--
TIME_PROX	TBD	Time tag for M50 state (assumed MET)	PROX_TGT_SUP_LAMB	F	sec
EL_ANG	TBD	Desired elevation angle at TPI		F	rad
$\overline{RS}$_M50_PROX	TBD	Prox ops Orbiter M50 position vector		F(3)	ft
$\overline{RT}$_M50_PROX	TBD	Prox ops target M50 position vector		F(3)	ft
$\overline{VS}$_M50_PROX	TBD	Prox ops Orbiter M50 velocity vector	PROX_EXEC	F(3)	ft/sec
$\overline{VT}$_M50_PROX	TBD	Prox ops target M50 velocity vector		F(3)	ft/sec
$\overline{R}$_OUT	TBD	Output inertial position vector		F(3)	ft
$\overline{V}$_OUT	TBD	Output inertial velocity vector	UPDATVP	F(3)	ft/sec

TABLE 36.- Concluded

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
TTPI	TBD	Predicted time of TPI in MET	ELITER	F	sec
EL_ANG_COM	TBD	Computed elevation angle	COMELE	F	rad
ERR_PRIME	TBD	Previous value of the elevation angle guess		F	rad
TTPI_PRIME	TBD	Previous value of the TPI time guess	ELITER	F	sec
N	TBD	Iteration counter		I	--

TABLE 37.- OUTPUT PARAMETERS FOR THE TELEV MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
S_OPTION	TBD	Spacecraft flag for UPDATVP		D	--
$\bar{R}$_IN	TBD	Input inertial position vector		F(3)	ft
$\bar{V}$_IN	TBD	Input inertial velocity vector	UPDATVP	F(3)	ft/sec
T_IN	TBD	Input time in GMT		F	sec
T_OUT	TBD	Output time in GMT		F	sec
$\bar{RS}$_OUT	TBD	Shuttle position vector		F(3)	ft
$\bar{VS}$_OUT	TBD	Shuttle velocity vector	ELITER	F(3)	ft/sec
$\bar{RT}$_OUT	TBD	Target position vector		F(3)	ft
$\bar{VT}$_OUT	TBD	Target velocity vector		F(3)	ft/sec
$\bar{RS}$_COM	TBD	Shuttle inertial position		F(3)	ft
$\bar{VS}$_COM	TBD	Shuttle inertial velocity	COMELE	F(3)	ft/sec
$\bar{RT}$_COM	TBD	Target inertial position		F(3)	ft
ERR_PRIME	TBD	Previous value of the dependent variable in search of elevation angle	ELITER	F	rad
TIPI_PRIME	TBD	Previous value of the independent variable in search of elevation angle		F	sec
TTPI	TBD	Predicted time of TPI (MET)	ELITER PROX_TGT_SUP_LAMB	F	sec

TABLE 38.- INPUT PARAMETERS FOR THE PREVR MODULE

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
BASE_MET	TBD	GMT/MET reference time	FCOS	F	sec
PI	TBD	Mathematical constant	ILOAD	F	--
WS	TBD	Weight of the Orbiter	ON_ORB_UPP (4.22)	F	ft/sec^2
R_TOL	TBD	Convergence tolerance in the terminal point offset iteration	ILOAD	F	ft
EARTH_MU	TBD	Earth gravitational constant	ILOAD	F	ft^3/sec^2
N_MAX	TBD	Maximum allowed number of iterations	ILOAD	I	--
CONE	TBD	Angular tolerance used to determine if the transfer angle ~ 180°	ILOAD	F	rad
N_MIN	TBD	Minimum allowed number of iterations	ILOAD	I	--
T1_TIG	TBD	T1 maneuver time (MET)	PROX_TGT_SUP_LAMB	F	sec
T2_TIG	TBD	T2 maneuver time (MET)	PROX_TGT_SUP_LAMB	F	sec
$\overline{\mathrm{RS}}$_T1TIG	TBD	Inertial position vector of Shuttle at T1 TIG	PROX_TGT_SUP_LAMB	F(3)	ft
$\overline{\mathrm{VS}}$_T1TIG	TBD	Inertial velocity vector of Shuttle at T1 TIG	PROX_TGT_SUP_LAMB	F(3)	ft/sec
$\overline{\mathrm{RS}}$_T2TIG	TBD	Inertial position vector of Shuttle at T2 TIG	PROX_TGT_SUP_LAMB	F(3)	ft
$\overline{\mathrm{R}}$_OUT	TBD	Output inertial position vector	UPDATVP	F(3)	ft
$\overline{\mathrm{V}}$_OUT	TBD	Output inertial velocity vector	UPDATVP	F(3)	ft/sec
$\overline{\mathrm{VS}}$_REQUIRED	TBD	Velocity required at the maneuver time	LAMBERT	F(3)	ft/sec

TABLE 38.- Concluded

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
G	TBD	Gravitational constant	ILOAD	F	ft/sec^2
THRUST	TBD	Thrust of the propulsion system	ILOAD	F	lbs

TABLE 39.- OUTPUT PARAMETERS FOR THE PREVR MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
S_ROTATE	TBD	Rotation flag for near-180° transfers	PROX_TRANS	D	--
$\overline{\text{R}}$_OFFSET	TBD	Offset position vector (inertial)	PROX_TRANS	F(3)	ft
T_OFFSET	TBD	Time of offset position vector in MET	UPDATVP	F	sec
$\overline{\text{R}}$_IN	TBD	Input inertial position vector	UPDATVP	F(3)	ft
$\overline{\text{V}}$_IN	TBD	Input inertial velocity vector	UPDATVP	F(3)	ft/sec
T_IN	TBD	Time of input state in GMT	UPDATVP	F	sec
T_OUT	TBD	Time of output state in GMT	UPDATVP	F	sec
ALARM	TBD	Alarm flag indicating error	--	I	--
$\overline{\text{RS}}$_IPO	TBD	Initial position offset in GMT	LAMBERT	F(3)	ft
$\overline{\text{UN}}$	TBD	Unit normal along the Shuttle angular momentum vector	LAMBERT	F(3)	--
DEL_T_TRAN	TBD	Transfer time	LAMBERT	F	sec
$\overline{\text{VS}}$_REQUIRED	TBD	Velocity required at the maneuver time	PROX_TGT_SUP_LAMB	F(3)	ft/sec

TABLE 40.- INPUT PARAMETERS FOR THE COMELE MODULE

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
PI	TBD	Mathematical constant	ILOAD	F	--
$\bar{R}$S_COM	TBD	Shuttle inertial position		F(3)	ft
$\bar{V}$S_COM	TBD	Shuttle inertial velocity	TELEV	F(3)	ft/sec
$\bar{R}$T_COM	TBD	Target inertial position		F(3)	ft

TABLE 41.- OUTPUT PARAMETERS FOR THE COMELE MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
EL_ANG_COM	TBD	Computed elevation angle	TELEV	F	rad

TABLE 42.- INPUT PARAMETERS FOR THE ELITER MODULE

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
BASE_MET	TBD	GMT/MET reference time	FCOS	F	sec
DEL_T_MAX	TBD	Maximum step size used during an iteration	ILOAD	I	--
$\bar{R}$_OUT	TBD	Output inertial position vector	UPDATVP	F(3)	ft
$\bar{V}$_OUT	TBD	Output inertial velocity vector		F(3)	ft/sec
TTPI	TBD	Predicted time of TPI (MET)	TELEV	F	sec
$\bar{R}$S_OUT	TBD	Shuttle position vector		F(3)	ft
$\bar{V}$S_OUT	TBD	Shuttle velocity vector	TELEV	F(3)	ft/sec
$\bar{R}$T_OUT	TBD	Target position vector		F(3)	ft
$\bar{V}$T_OUT	TBD	Target velocity vector		F(3)	ft/sec
X_IND	TBD	Independent variable		F	--
X_DEP_PRIME	TBD	Previous value of the dependent variable	ITERV	F	--
X_IND_PRIME	TBD	Previous value of the independent variable		F	--
ERR	TBD	Guess of the elevation angle		F	rad
ERR_PRIME	TBD	Previous guess of the elevation angle	TELEV	F	rad
TTPI_PRIME	TBD	Previous guess of the TPI time		F	sec
N	TBD	Iteration counter		I	--

TABLE 43.- OUTPUT PARAMETERS FOR THE ELITER MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
S_OPTION	TBD	Spacecraft flag for UPDATVP	UPDATVP	I	--
$\bar{R}$_IN	TBD	Input inertial position vector for UPDATVP	UPDATVP	F(3)	ft
$\bar{V}$_IN	TBD	Input inertial velocity for UPDATVP	UPDATVP	F(3)	ft/sec
TTPI	TBD	Predicted time of TPI (MET)	TELEV	F	sec
T_IN	TBD	Time of input state to UPDATVP in GMT	UPDATVP	F	sec
T_OUT	TBD	Time of output state to UPDATVP in GMT	UPDATVP	F	sec
$\bar{R}$S_OUT	TBD	Shuttle position vector	PROX_TGT_SUP_LAMB	F(3)	ft
$\bar{V}$S_OUT	TBD	Shuttle velocity vector	PROX_TGT_SUP_LAMB	F(3)	ft/sec
$\bar{R}$T_OUT	TBD	Target position vector	PROX_TGT_SUP_LAMB	F(3)	ft
$\bar{V}$T_OUT	TBD	Target velocity vector	PROX_TGT_SUP_LAMB	F(3)	ft/sec
TS_OUT	TBD	Time of the Shuttle position vector (GMT)	PROX_TGT_SUP_LAMB	F	sec
TT_OUT	TBD	Time of the Target position vector (GMT)		F	sec
IC	TBD	Iteration counter to ITERV	ITERV	I	--
X_DEP	TBD	Dependent variable to ITERV	ITERV	F	--
X_IND	TBD	Independent variable to ITERV	ITERV	F	--
X_DEP_PRIME	TBD	Previous value of the dependent variable to ITERV	ITERV	F	--
X_IND_PRIME	TBD	Previous value of the independent variable to ITERV	ITERV	F	--

TABLE 43.- Concluded

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
ERR_PRIME	TBD	Previous guess of the elevation angle		F	rad
TTPI_PRIME	TBD	Previous guess of TPI in MET	TELEV	F	sec
N	TBD	Iteration counter for ELITER		I	--

TABLE 44.- INPUT PARAMETERS FOR THE LAMBERT MODULE

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
EARTH_MU	TBD	Earth gravitational constant		F	ft^3/sec^2
N_MAX	TBD	Maximum allowed number of iterations		I	--
DU	TBD	Small deviation to prevent orbit from being almost parabolic	ILOAD	I	--
EP_TRANSFER	TBD	Parameter to test if transfer angle is close to 0°		F	--
EPS_U	TBD	Parameter to test convergence of the Newton-Raphson iteration		F	--
$\overline{R}$_OFFSET	TBD	Offset position vector		F(3)	ft
$\overline{RS}$_IPO	TBD	Initial position offset of Shuttle	PREVR	F(3)	ft
$\overline{UN}$	TBD	Unit normal along the Shuttle angular momentum vector		F(3)	--
DEL_T_TRAN	TBD	Transfer time		F	sec

TABLE 45.- OUTPUT PARAMETERS FOR THE LAMBERT MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
ALARM	TBD	Alarm flag to show error	--	I	--
$\bar{V}$S_REQUIRED	TBD	Velocity required at the maneuver time	PROX_TGT_SUP_LAMB PREVR	F(3)	ft/sec

TABLE 46.- INPUT PARAMETERS FOR THE UPDATVP MODULE

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
GMD_PRED_I	TBD	Gravity Model Degree Flag		D	--
GMO_PRED_I	TBD	Gravity Model Order Flag		D	--
ATM_I	TBD	Attitude Mode Flag		D	--
DMP_I	TBD	Drag Model Flag	ILOAD	D (2)	--
VMP_I	TBD	Vent Model Flag		D (2)	--
PRED_ORB_MASS_I	TBD			F (2)	--
PRED_ORB_CD_I	TBD	Onorbit Predictor Drag		F (2)	--
PRED_ORB_AREA_I	TBD			F (2)	--
S_OPTION	TBD	Spacecraft Flag for UPDATVP		I	--
$\vec{R}$_IN	TBD	Input Inertial Position for Vector for UPDATVP	PROX_TGT_SUP_LAMB TELEV PREVR ELITER	F (3)	ft
$\vec{V}$_IN	TBD	Input Inertial Velocity		F (3)	ft/sec
T_IN	TBD	Time of the Input State to UPDATVP in GMT		F	sec
T_OUT	TBD	Time of the Output State to UPDATVP in GMT		F	sec

TABLE 47.- OUTPUT PARAMETERS FOR THE UPDATVP MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
$\bar{R}$_OUT	TBD	Output inertial position vector	PROX_TGT_SUP_LAMB TELEV PREVR ELITER	F(3)	ft
$\bar{V}$_OUT	TBD	Output inertial velocity vector		F(3)	ft/sec

TABLE 48.- INPUT PARAMETERS FOR THE ITERV MODULE

SYMBOL	PRECISION	DEFINITION	SOURCE	TYPE	UNITS
DEL_X_GUESS	TBD	ΔX guess for iterator if no prediction is possible		F	--
IC_MAX	TBD	Maximum allowed number of iterations	ILOAD	I	--
DEL_X_TOL	TBD	Tolerance of dependent variable to ensure that a slope exists		F	--
IC	TBD	Iteration counter in ITERV		I	--
X_DEP	TBD	Dependent variable to ITERV		F	--
X_IND	TBD	Independent variable to ITERV	ITERV	F	--
X_DEP_PRIME	TBD	Previous value of the dependent variable to ITERV		F	--
X_IND_PRIME	TBD	Previous value of the independent variable to ITERV		F	--

TABLE 49.- OUTPUT PARAMETERS FOR THE ITERV MODULE

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
X_IND	TBD	Independent variable to ITERV	ELITER	F	--
X_DEP_PRIME	TBD	Previous value of the dependent variable	ELITER	F	--
X_IND_PRIME	TBD	Previous value of the independent variable	ELITER	F	--
SFAIL	TBD	Flag set if maximum number of iterations occurred	--	D	--

TABLE 50.- INPUT PARAMETERS FOR THE ORBLV TASK

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
$\bar{RS}$	TBD	Input inertial position of Shuttle		F(3)	ft
$\bar{VS}$	TBD	Input inertial velocity of Shuttle	PROX_TGT_SUP_LAMB	F(3)	ft/sec
$\bar{RT}$	TBD	Input inertial position of target		F(3)	ft
$\bar{VT}$	TBD	Input inertial velocity of target		F(3)	ft/sec

TABLE 51.- OUTPUT PARAMETERS FOR THE ORBLV TASK

SYMBOL	PRECISION	DEFINITION	DESTINATION	TYPE	UNITS
$\bar{V}SLV$	TBD	Shuttle-centered LVLH relative velocity of target	PROX_TGT_SUP_LAMB	F(3)	ft/sec

```
XXXXXXXX/XXX      ORBIT TGT       XX X DDD/HH:MM:SS
                                       DDD/HH:MM:SS
MNVR    TIG        ΔVX     ΔVY      ΔVZ      ΔVT
XX X X/XX:XX:XX  ±XXX.X  ±XX.X   ±XX.X   ±XXX.X

INPUTS                                    CONTROLS
 1 TGT NO             XX                  T2 TO T1    25
 2 TI TIG  X/XX:XX:XX                     XXXX        26
 6   EL         [±]XXX.XX                 COMPUTE T1  27X
 7   ΔX/DNRNG   [±]XXX.XX                 COMPUTE T2  28X
 8   ΔY         [±]XXX.XX
 9   ΔZ/ΔH      [±]XXX.XX
10   ΔẊ         [±]XXX.XX
11   ΔẎ         [±]XXX.XX
12   ΔŻ         [±]XXX.XX
13 T2 TIG  X/XX:XX:XX
17   ΔT            XXX.X
18   ΔX         [±]XXX.XX
19   ΔY         [±]XXX.XX
20   ΔZ         [±]XXX.XX
21 BASE TIME X/XX:XX:XX

                                                    (XX)
```

Figure 1.- Orbit targeting generic display.

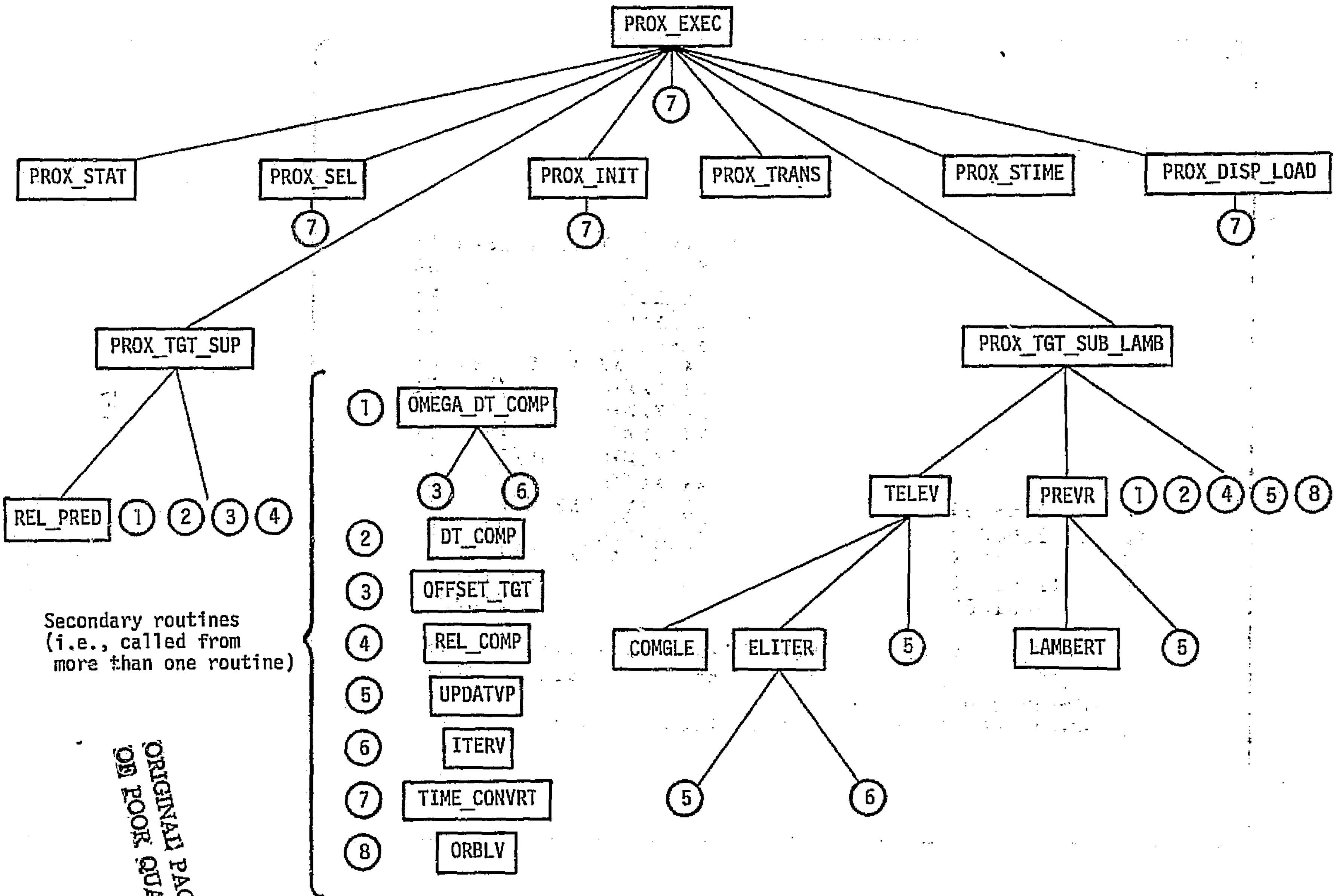


Figure 2.- Orbit targeting specialist function task organization.

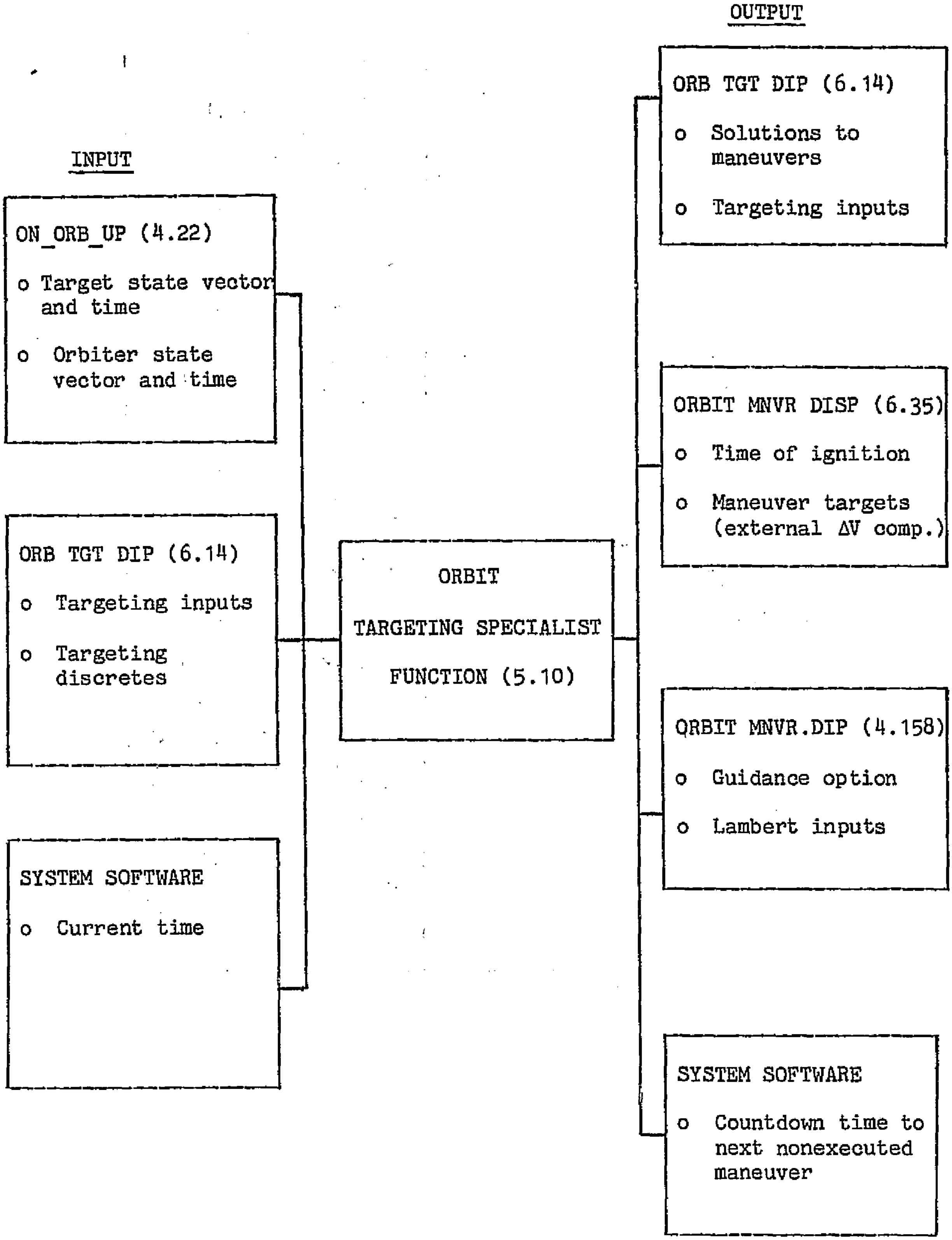
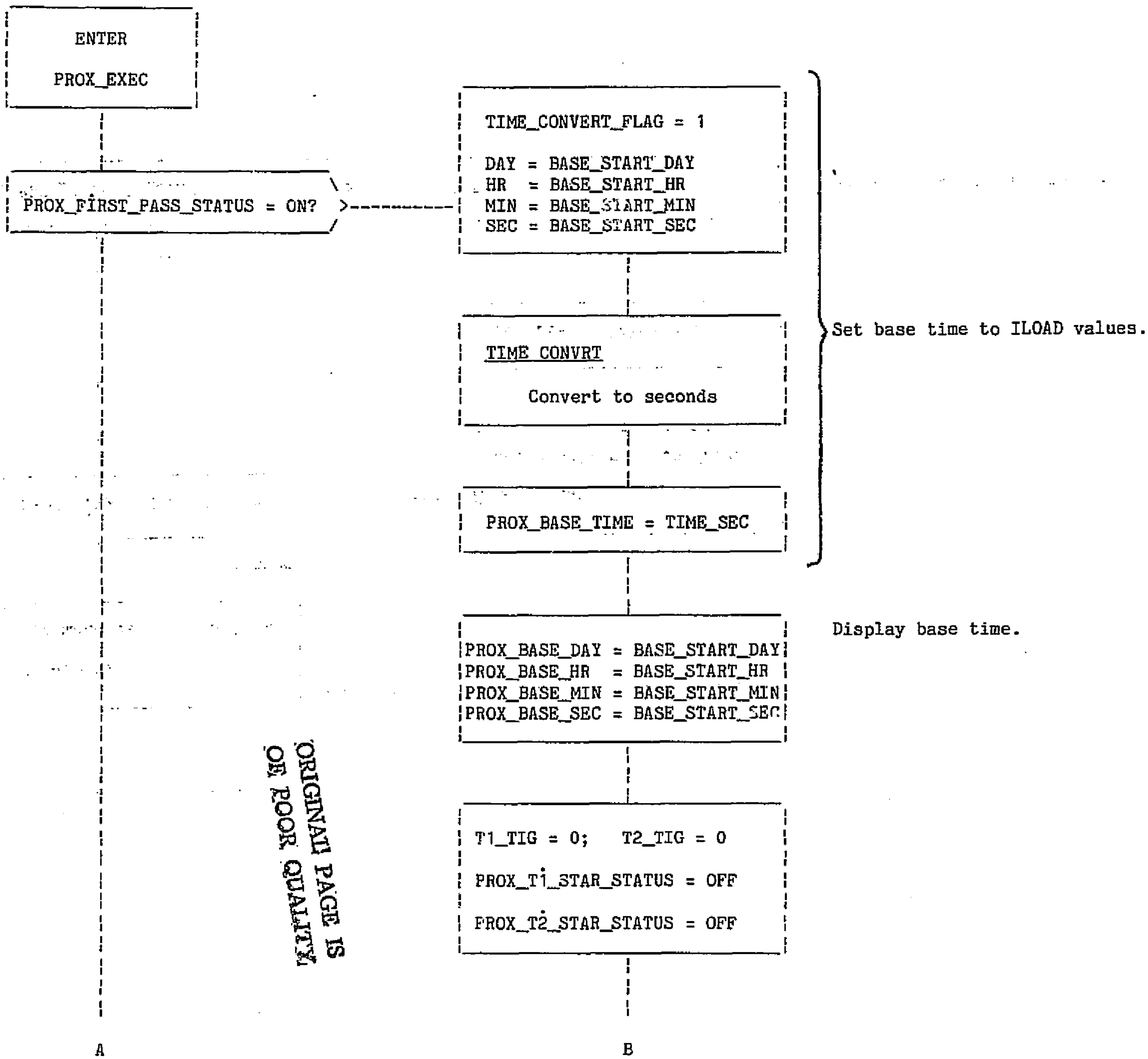


Figure 3.- Orbit targeting specialist function data flow.

		PROX_WORLD_COMMON	PROX_DIP_COMMON	PROX_ILOAD_COMMON	PROX_VARIABLES_COMMON	PROX_LAMVAR_COMMON
1	PROX_EXEC	✓	✓	✓	✓	
2	PROX_STAT	✓	✓		✓	
3	PROX_TGT_SEL		✓	✓	✓	
4	PROX_INIT		✓	✓	✓	
5	PROX_TRANS	✓	✓		✓	
6	PROX_TGT_SUP	✓	✓	✓	✓	
7	PROX_TGT_SUP_LAMB	✓	✓	✓	✓	✓
8	PROX_STIME	✓	✓			
9	OFFSET_TGT				✓	
10	REL_PRED				✓	
11	REL_COMP				✓	
12	PROX_DISP_LOAD		✓		✓	
13	TIME_CONVRT				✓	
14	DT_COMP				✓	
15	OMEGA_DT_COMP		✓	✓	✓	✓
16	TELEV	✓	✓	✓	✓	✓
17	PREVR	✓		✓	✓	✓
18	COMELE			✓		✓
19	ELITER	✓		✓		✓
20	LAMBERT			✓	✓	✓
21	UPDATVP	✓		✓		✓
22	ITERV			✓		✓
23	ORBLV					✓

ORIGINAL PAGE IS
OF POOR QUALITY

Figure 4.- Common package allocations for function modules.



Page 1 of 6

Figure 5.- Orbit targeting executive task functional flow.

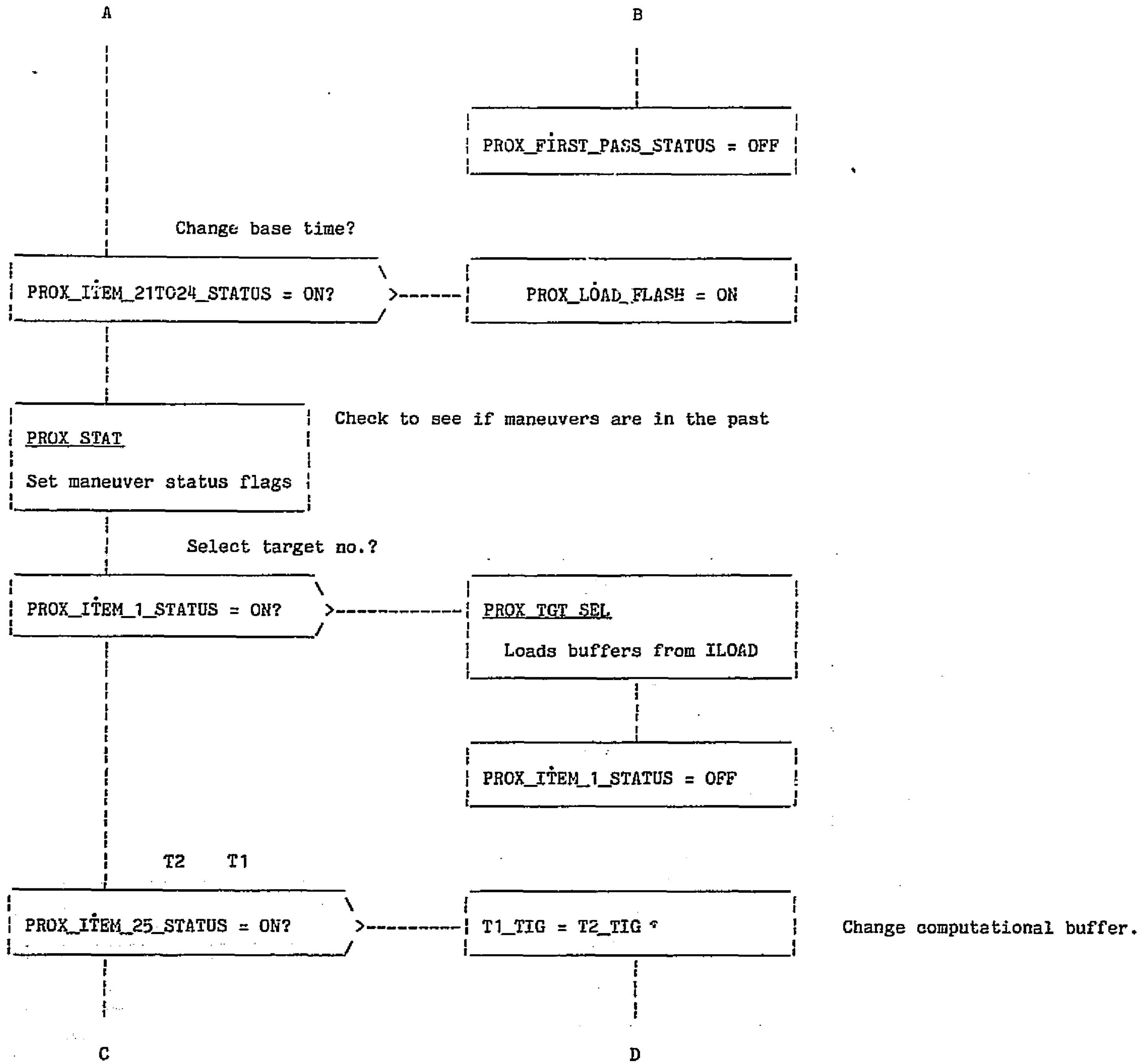


Figure 5.- Continued.

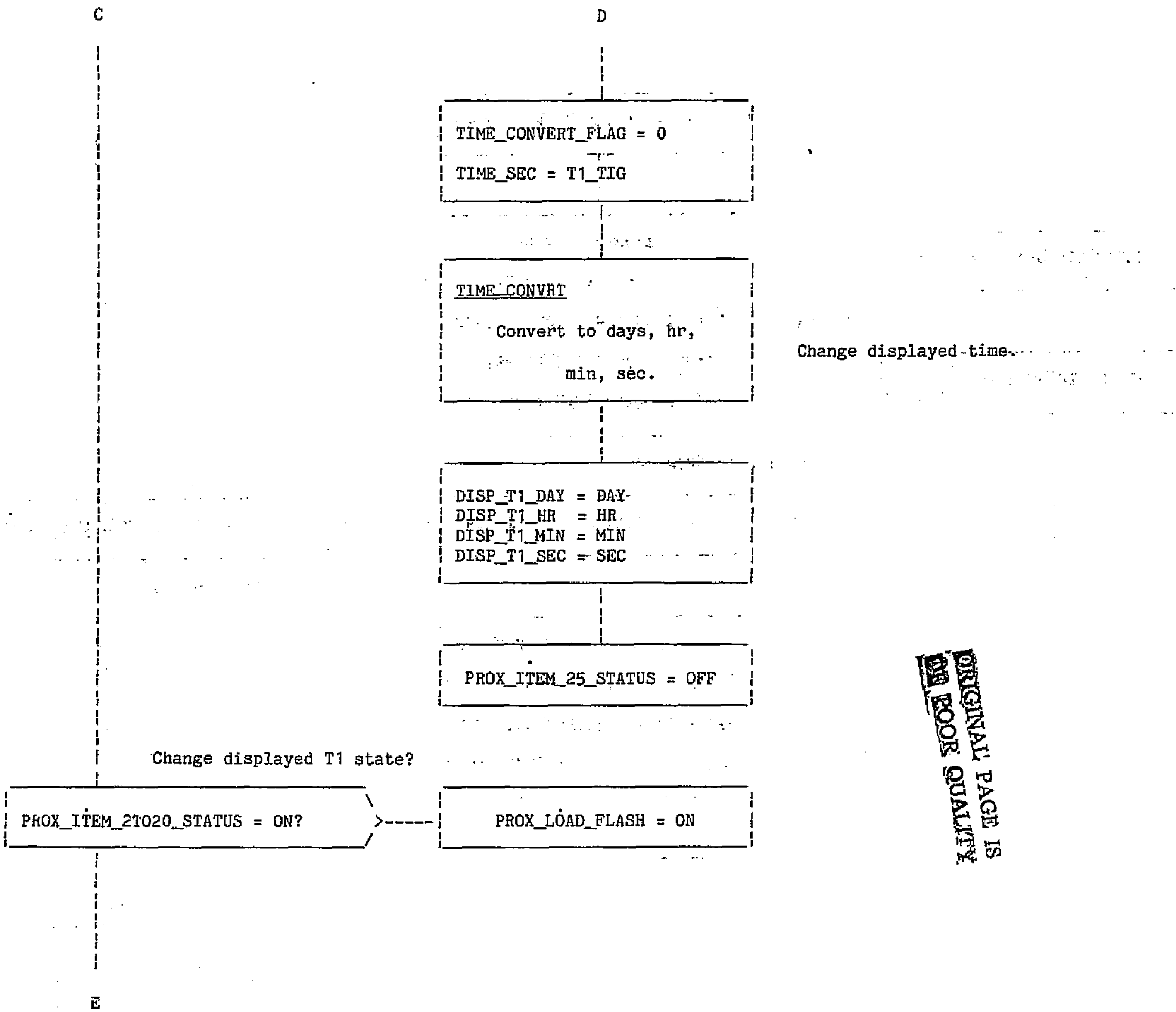


Figure 5.- Continued

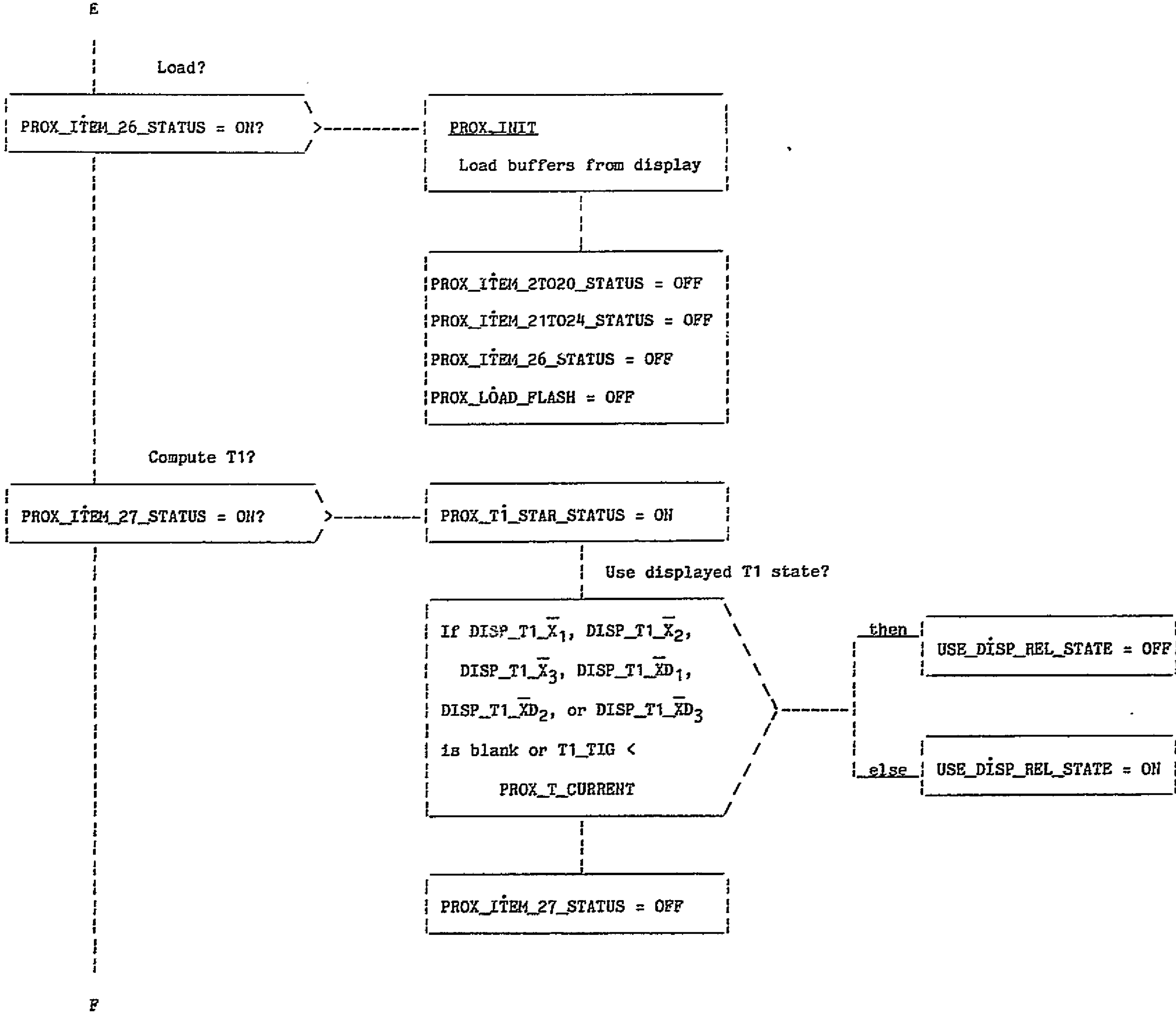


Figure 5.- Continued

145

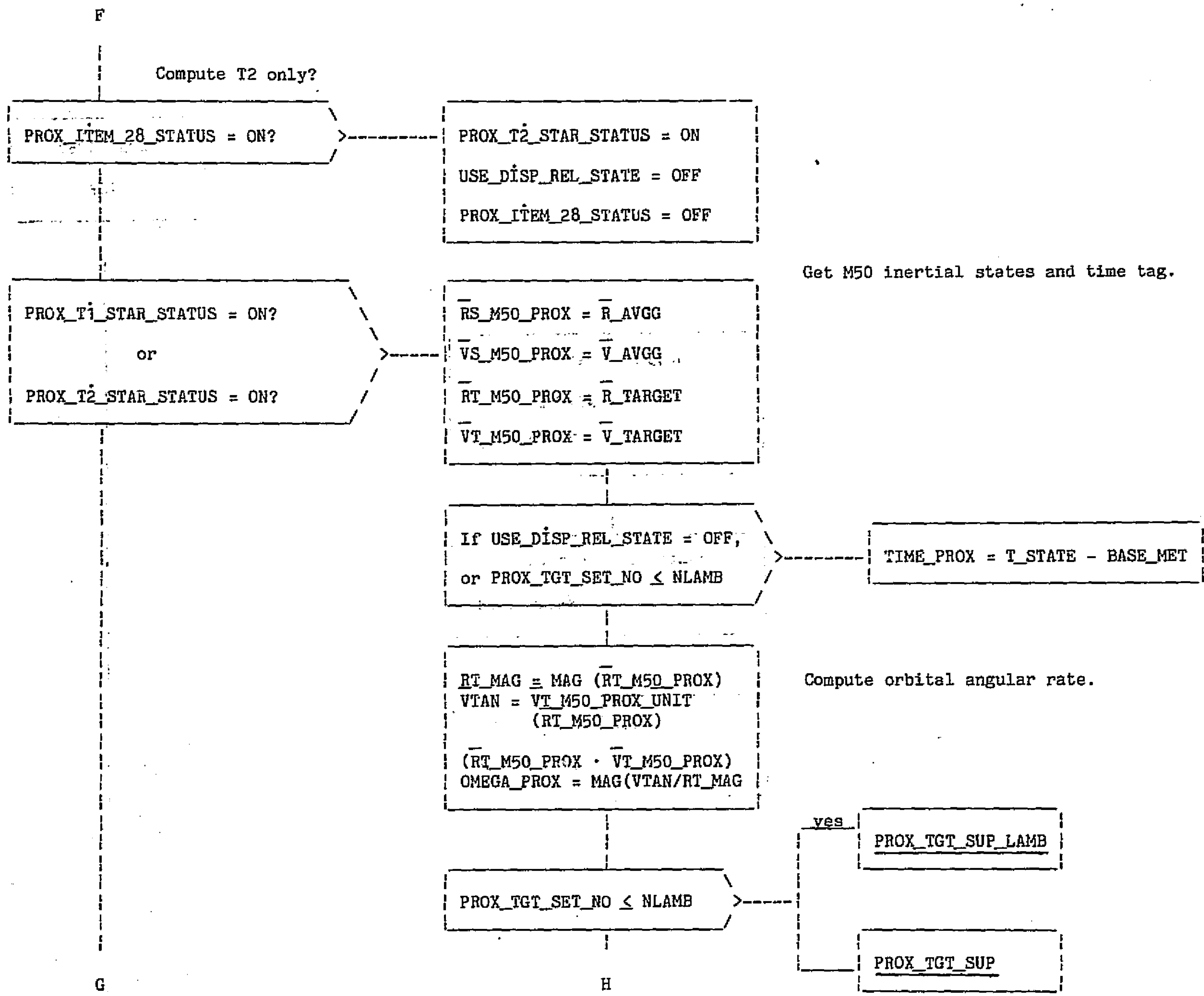


Figure 5.- Continued

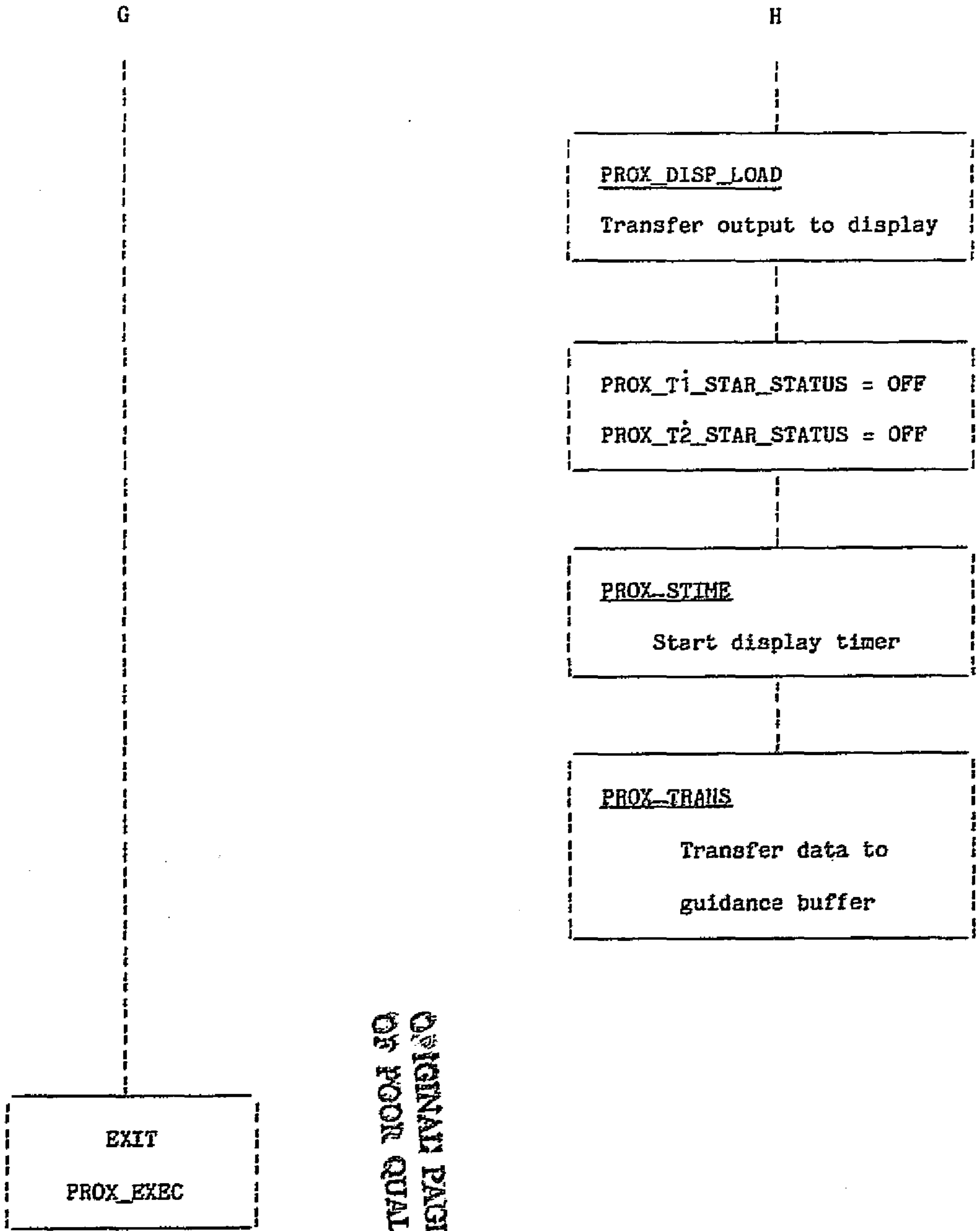


Figure 5 — Concluded.

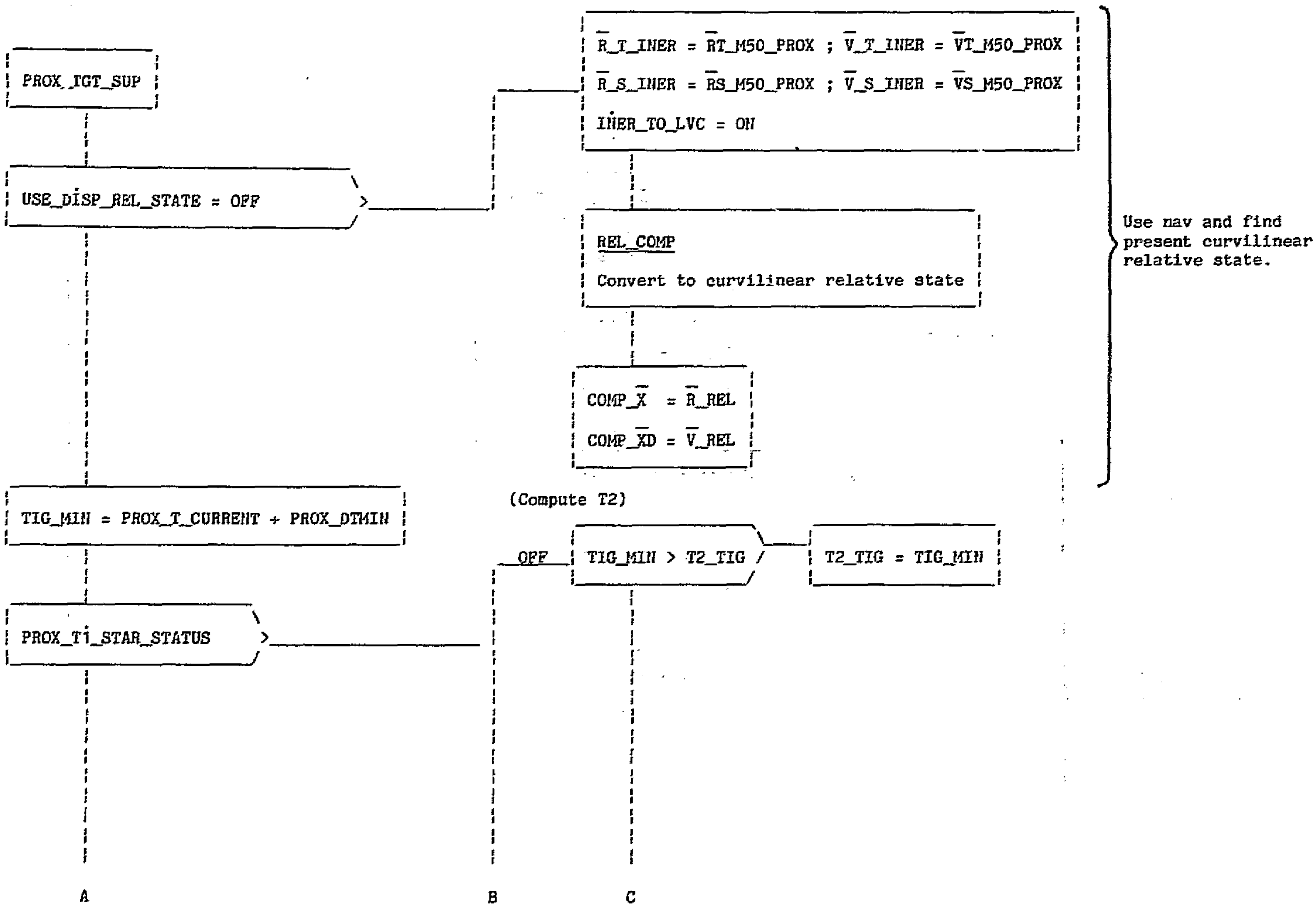


Figure 6.- Proximity operations targeting supervisory logic task functional flow.

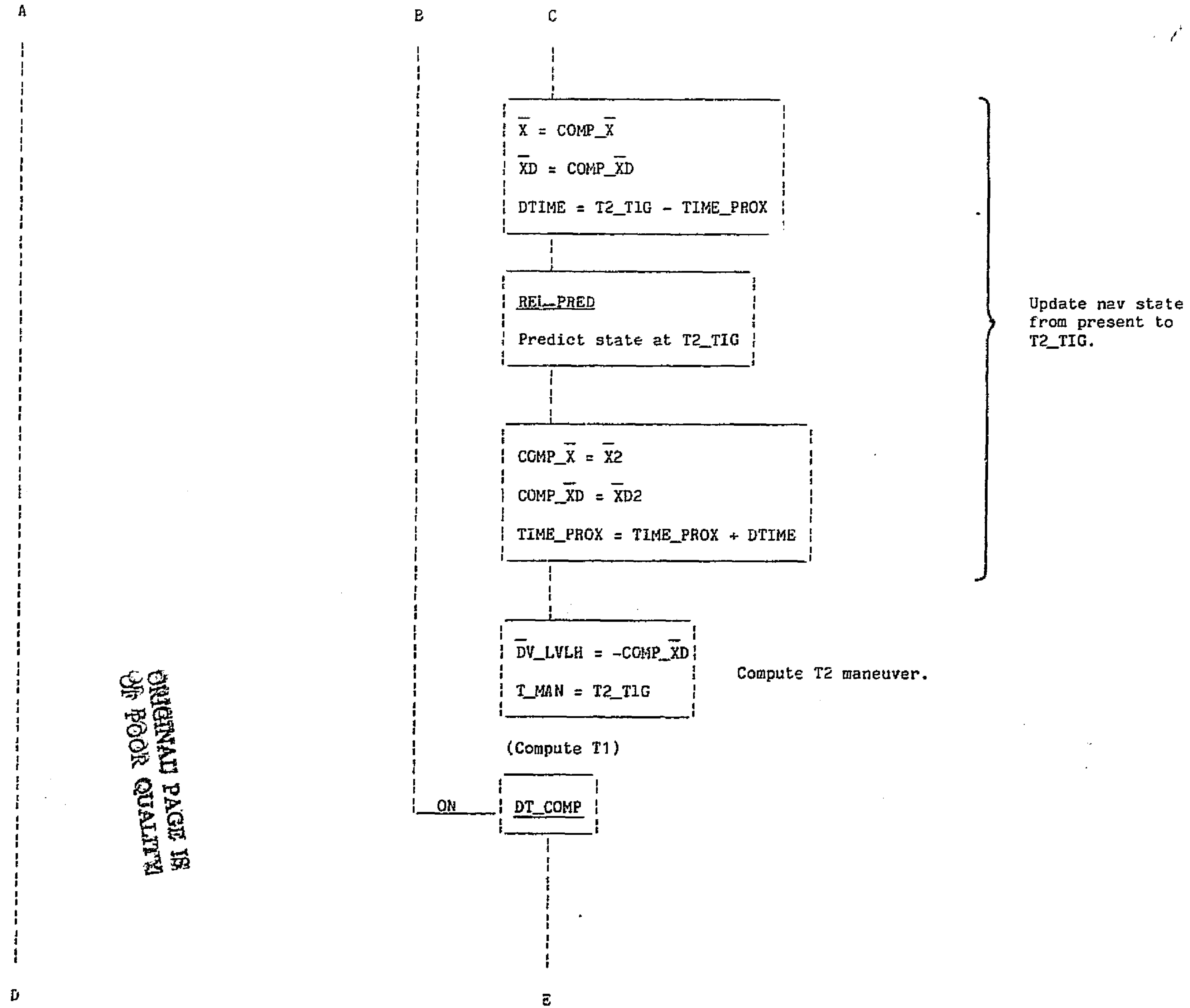


Figure 6.- Continued.

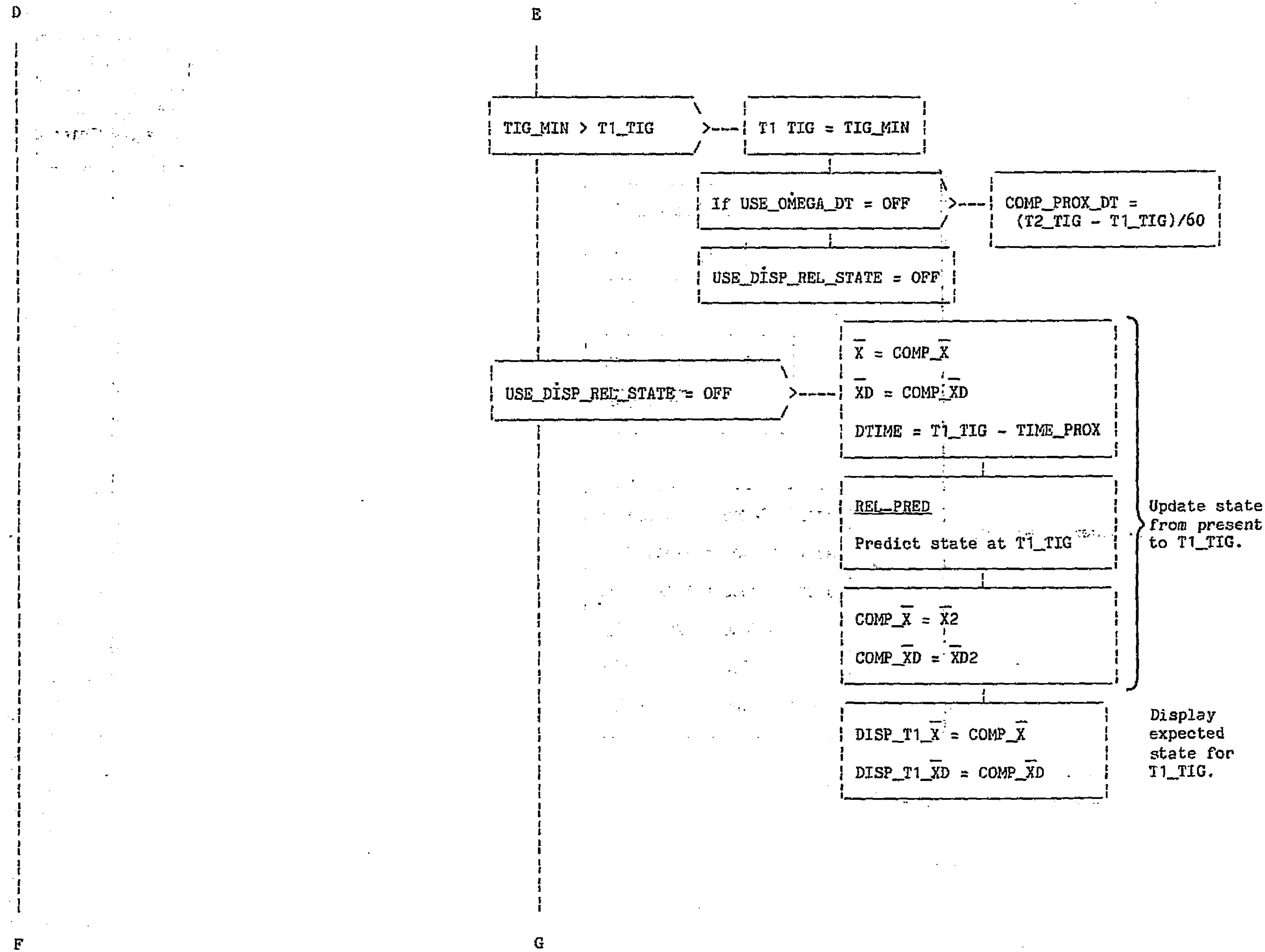


Figure 6.- Continued

F

GUID_FLAG = 0

EXIT

G

USE_OMEGA_DT = ON > --- OMEGA_DT_COMP

T2_TIG = T1_TIG + 60 COMP_PROX_DT

DT_OFFTGT = COMP_PROX_DT

$\overline{X}$_OFFTGT = COMP_$\overline{X}$

$\overline{XD}$_OFFTGT = COMP_$\overline{XD}$

$\overline{X2}$_OFFTGT$_1$ = COMP_T2_XOFF

$\overline{X2}$_OFFTGT$_2$ = COMP_T2_YOFF

$\overline{X2}$_OFFTGT$_3$ = COMP_T2_ZOFF

OFFSET_TGT

Compute T1 maneuver

$\overline{DV}$_LVLH = $\overline{DV}$

T_MAN = T1_TIG

Compute T1 maneuver.

ORIGINAL PAGE IS OF POOR QUALITY

Figure 6.- Concluded

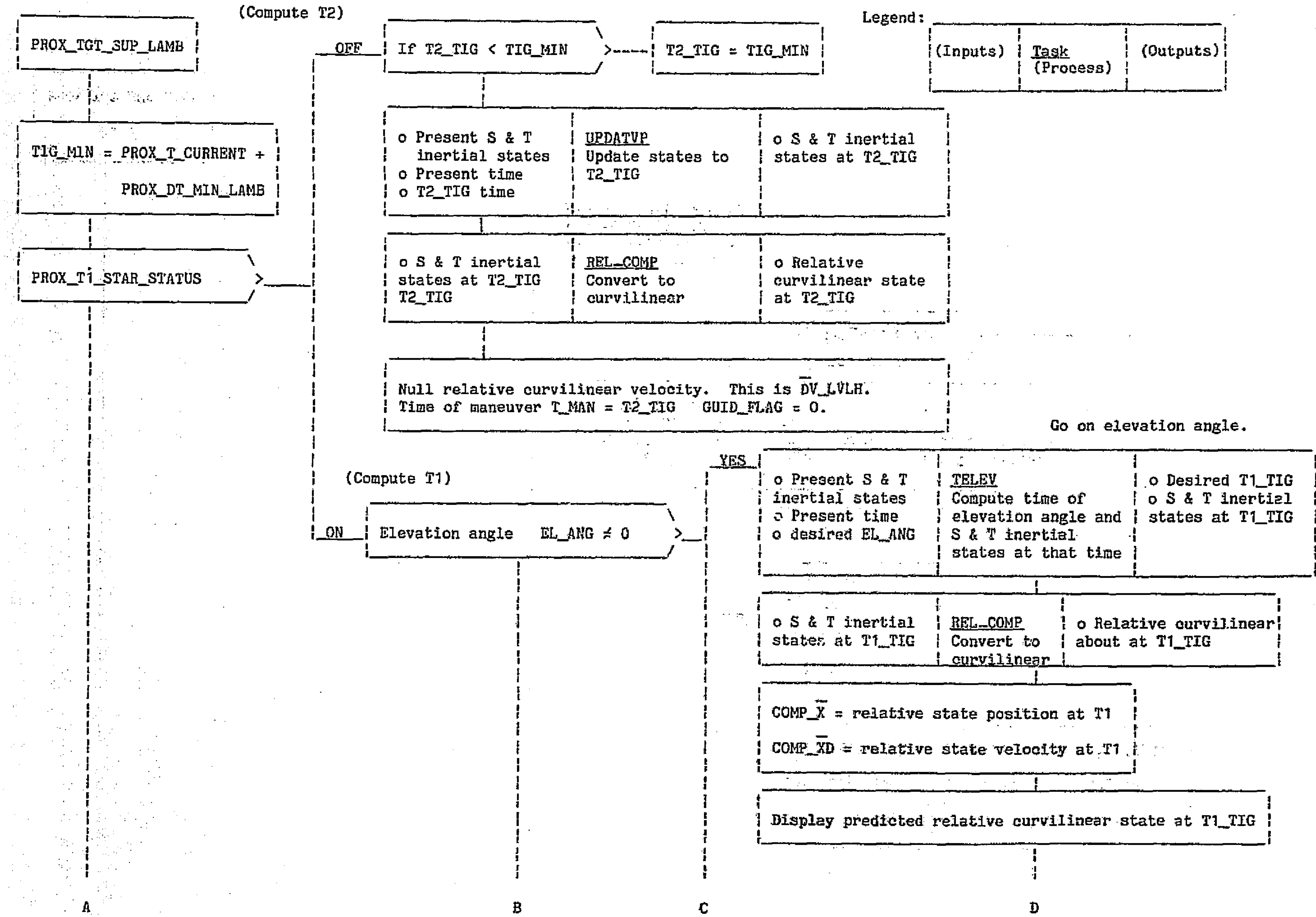


Figure 7.- Proximity operations targeting supervisory Lambert logic task functional flow.

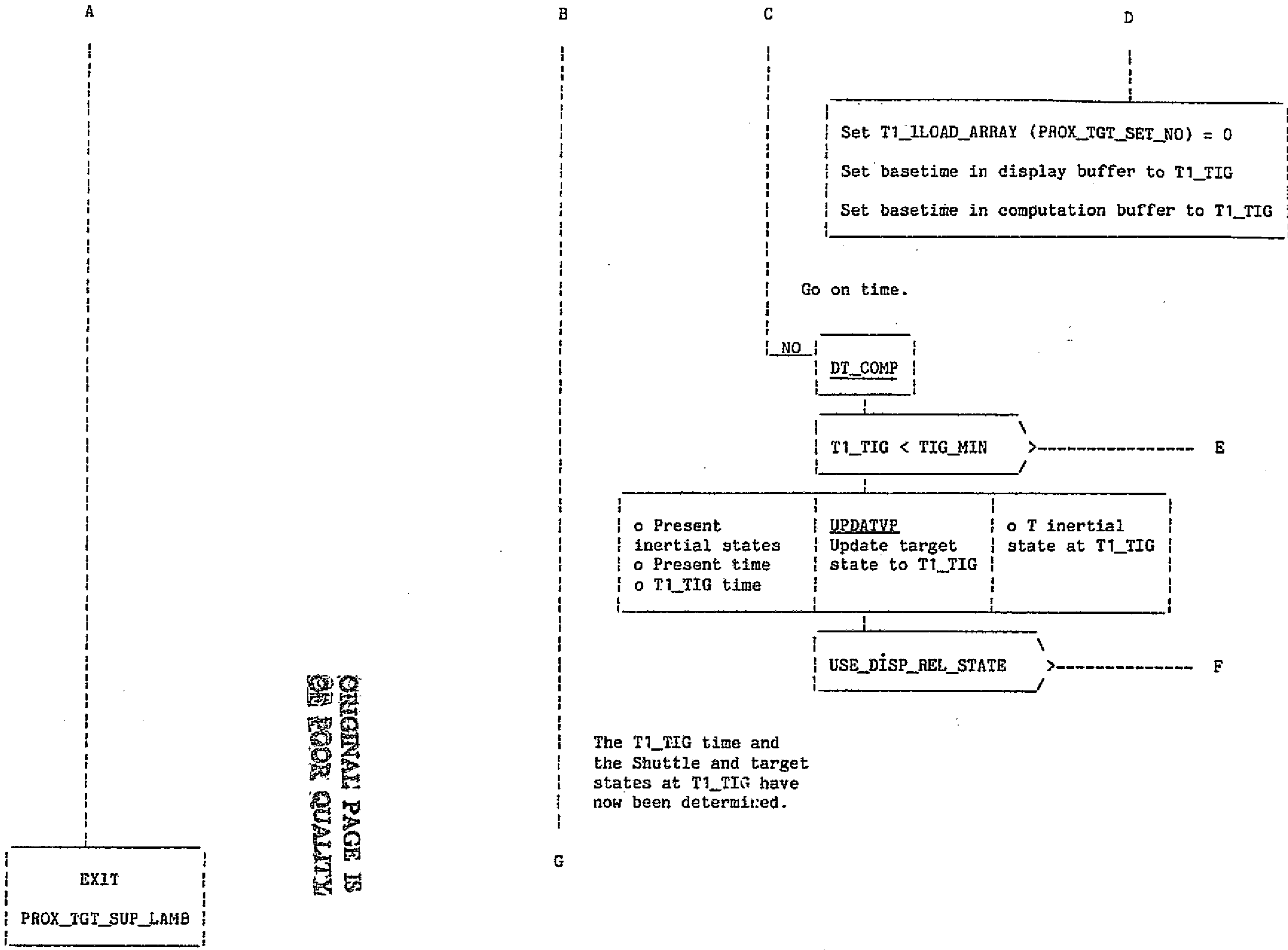


Figure 7.- Continued.

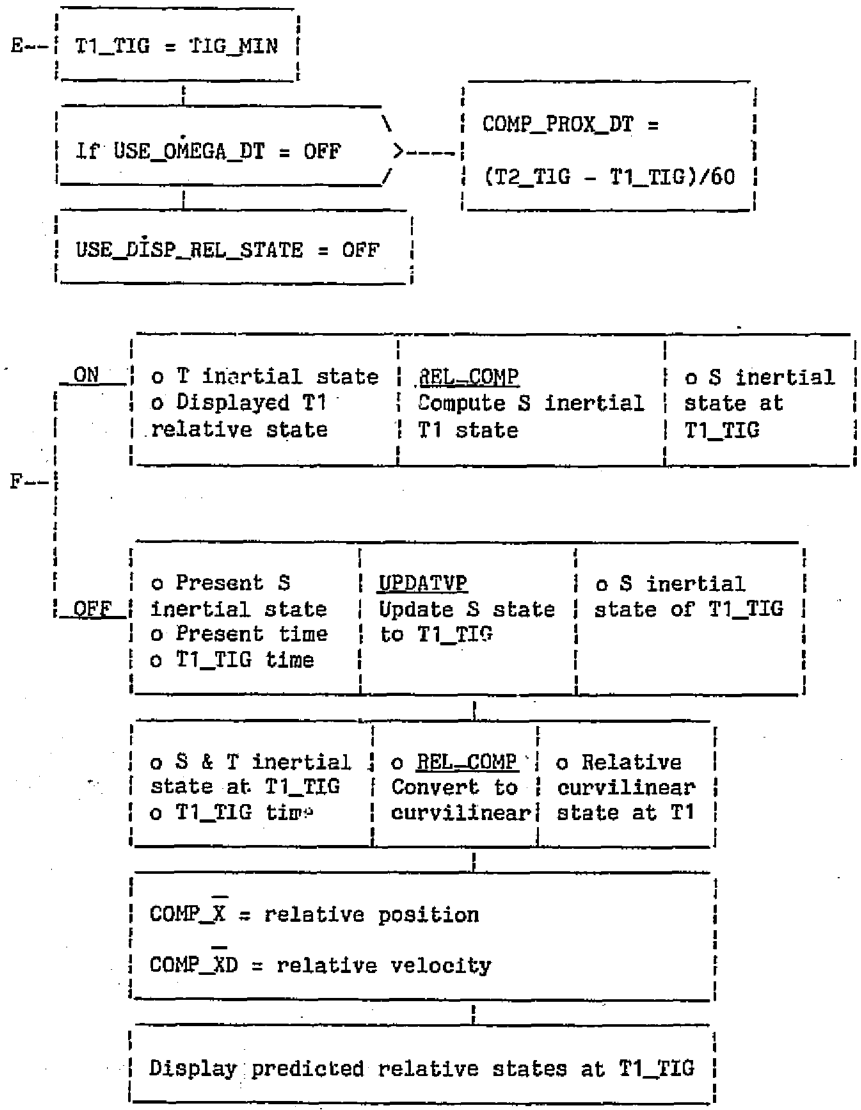
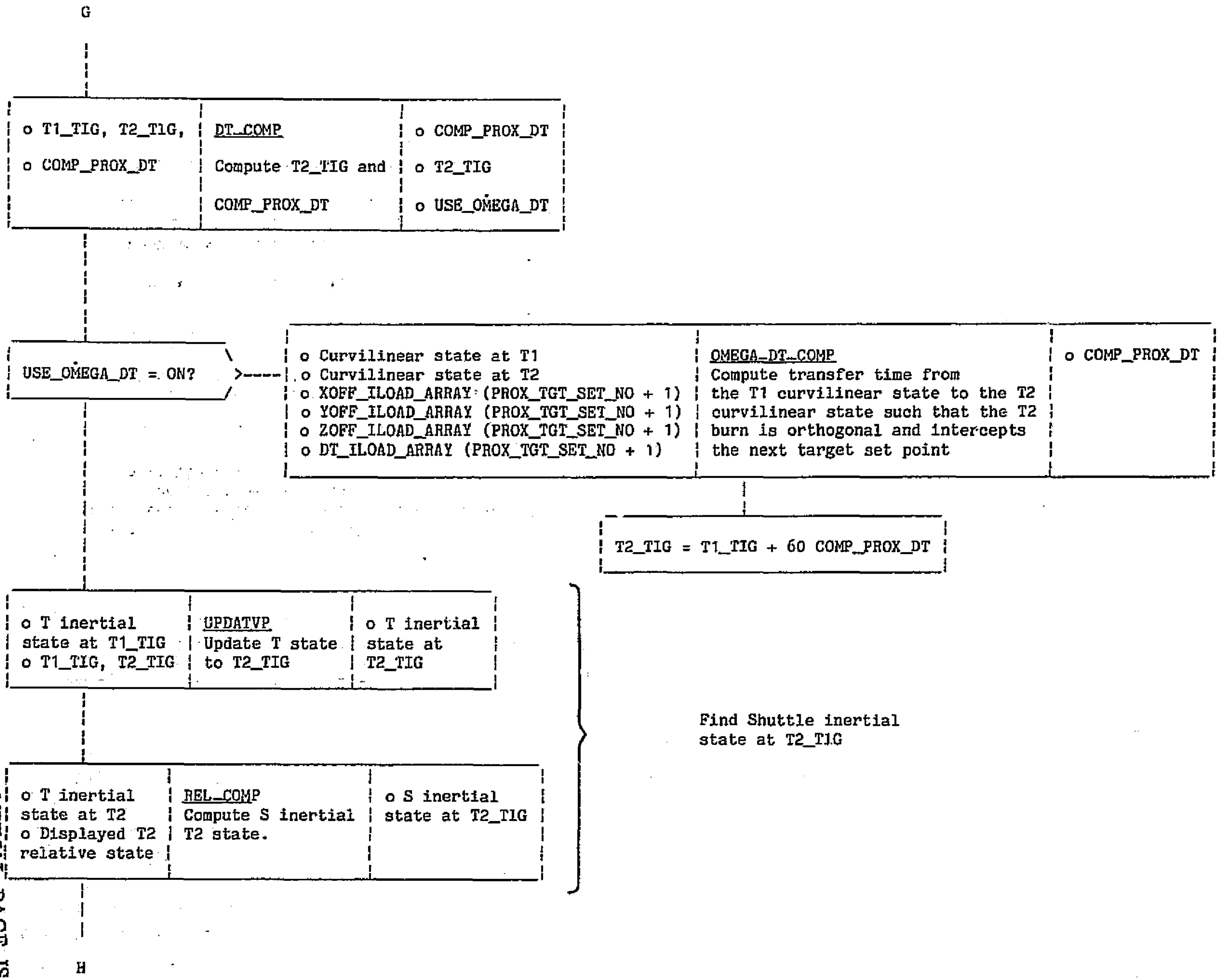


Figure 7.- Continued.



154

ORIGINAL PAGE IS OF POOR QUALITY

Figure 7.- Continued.

H

o COMP_PROX_DT o S inertial states at T1_TIG and T2_TIG.	PREVR Do Lambert problem	o $\overline{\text{VS}}$_REQUIRED, S inertial velocity required at T1_TIG o S inertial at T1

o Shuttle and target inertial states before the T1 maneuver	ORBLV Compute Shuttle-centered LVLH relative state	o $\overline{\text{V}}$_LVLH

o Shuttle and target inertial states after the T1 maneuver	ORBLV Compute Shuttle-centered LVLH relative state	o $\overline{\text{V}}$_LVLH_DESIRED

Determine T1 maneuver in LVLH coordinates.

$\overline{\text{DV}}$_LVLH = $\overline{\text{V}}$_LVLH - $\overline{\text{V}}$_LVLH_DESIRED

T_MAN = T1_TIG

GUID_FLAG = 1

Figure 7.- Concluded.