SPACE SHUTTLE GN & C EQUATION DOCUMENT

No. 7
(Revision 3)

Rendezvous Targeting

by

Wayne H. Tempelman

Charles Stark Draper Laboratory
July 1973
NAS9-10268

for

National Aeronautics and Space Administration
Guidance and Control Systems Branch
Avionics Systems Engineering Division
Lyndon B. Johnson Space Center, Houston, Texas
ACKNOWLEDGEMENT

This report was prepared under DSR Project 55-40800, sponsored by the Manned Spacecraft Center of the National Aeronautics and Space Administration through Contract NAS9-10268.

The publication of this report does not constitute approval by the National Aeronautics and Space Administration of the findings or the conclusions contained therein. It is published only for the exchange and stimulation of ideas.
REVISION 3 CHANGES

The following is a summary of the major technical changes included in this revision:

1. A plane change maneuver has been incorporated in the program's logic.

2. In the search for an apsidal crossing, program checks have been included to update through intervals of $\pi$ in the case of near circular orbits or close proximity to an apsidal point.

3. Following an astronaut overwrite of a Lambert maneuver, the state vector is updated to establish a new target vector for use in the powered flight routine.

4. Following the computation of each maneuver, the position vector of the primary vehicle is offset to help compensate for the effects of the finite maneuver prior to the update of the state vector to the next maneuver point.
FOREWORD


Gerald M. Levine
Division Leader, Guidance Analysis
NASA Programs Department
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section 1</td>
<td>Introduction</td>
</tr>
<tr>
<td>Section 2</td>
<td>Functional Flow Diagram</td>
</tr>
<tr>
<td>Section 3</td>
<td>Input and Output Variables</td>
</tr>
<tr>
<td>Section 4</td>
<td>Description of Equations</td>
</tr>
<tr>
<td>Section 5</td>
<td>Detailed Flow Diagrams</td>
</tr>
<tr>
<td>Section 6</td>
<td>Supplementary Information</td>
</tr>
</tbody>
</table>
NOMENCLATURE

a
Semi-major axis of a conic

a_i
Alarm code i

\(a_T\)
Estimated magnitude of thrust acceleration

a_1
Failure in fuel optimization loop

a_2
Failure in height loop

a_3
Failure in phasing loop

a_4
Failure in obtaining Lambert solution in General Maneuver Routine

a_5
Failure to find perigee/apogee in Search Routine

a_6
Incompatible altitudes and elevation angle

a_7
Failure to find time corresponding to elevation angle in Search Routine

a_8
Failure to find desired position vector in Desired Position Routine

a_9
Failure to update through \(\theta\) in Update Routine

c
Iteration counter

c_h
Height iteration counter

c_p
Phase iteration counter

c_1, c_2, c_3
Intermediate variables

\(\Delta h\)
Delta altitude

\(\Delta r_{proj}\)
Delta projected position
\( \Delta t \)  
Delta time

\( \Delta t_{\text{max}} \)  
Maximum time step allowed in Search Routine iteration

\( \Delta v \)  
Maneuver velocity

\( \Delta v_{\text{LOS}} \)  
Maneuver in line-of-sight coordinates

\( \Delta v_{\text{LV}} \)  
Maneuver in local vertical coordinates

\( \Delta v_h \)  
\( \Delta v \) used during height maneuver

\( \Delta v_p \)  
\( \Delta v \) used during phasing maneuver

\( \Delta v_T \)  
Delta velocity used in fuel minimization loop

\( \Delta x \)  
Delta independent variable

\( \Delta \theta \)  
Delta central angle

\( e \)  
Error

\( e_c \)  
Eccentricity

\( e_h \)  
Height error

\( e_p \)  
Phasing error

\( e_L \)  
Elevation angle (defined in Figure 2)

\( g_0 \)  
Reference value for the acceleration of gravity

\( i \)  
Unit vector

\( i_N \)  
Unit normal to the plane used in powered flight guidance

\( i \)  
Number of the maneuver

\( i_{\text{max}} \)  
Number of the last maneuver in the maneuver sequence

\( m \)  
Estimated vehicle mass

\( M \)  
Rotational matrix

\( n \)  
Vector normal to the orbital plane

\( n_r \)  
Number of revolutions
p Partial used in Newton Raphson iteration
r Distance ratio
r Position vector
\( \mathbf{r_D} \) Desired position vector
\( \mathbf{r_{1c}} \) Target vector used in powered flight guidance
s Switch used in Desired Position Routine
\( s_{\text{astro}} \) Astronaut overwrite switch
\( s_{\text{coplan}} \) Coplanar switch
\( s_{\text{direct}} \) Maneuver direction switch
\( s_{\text{eng}} \) Engine select switch
\( s_{\text{exit}} \) Program exit switch
\( s_{\text{fail}} \) Failure switch
\( s_{\text{man}} \) Maneuver switch
\( s_{\text{opt}} \) Maneuver optimizing switch
\( s_{\text{outp}} \) Out-of-plane switch
\( s_{\text{pert}} \) Perturbation switch
\( s_{\text{phase}} \) Phase match switch
\( s_{\text{proj}} \) Projection switch
\( s_{\text{rdes}} \) Desired position switch
\( s_{\text{soln}} \) Solution switch
\( s_{\text{search}} \) Search switch
\( s_{\text{tar}} \) Target offset switch
\( s_{\text{term}} \) Terminal constraint switch
\( s_{\text{update}} \) Update switch
\( t \)  
Time

\( t_F \)  
Final time

\( \mathbf{v} \)  
Velocity vector

\( \mathbf{v}_v \)  
Vertical component of velocity

\( \mathbf{v}_c \)  
Circular velocity

\( x \)  
Independent variable in Iteration Routine

\( y_p, \dot{y}_p \)  
Out-of-plane parameters (see Figure 6a)

\( \dot{y}_T \)  

\( \alpha \)  
Radial component of velocity divided by \( v_C \)

\( \beta \)  
Horizontal component of velocity divided by \( v_C \)

\( \epsilon_1 \)  
Tolerance on time in fuel optimizing loop

\( \epsilon_2 \)  
Tolerance on height in height loop

\( \epsilon_3 \)  
Tolerance on central angle in phasing loop

\( \epsilon_4 \)  
Tolerance on central angle in Search Routine

\( \epsilon_5 \)  
Tolerance on elevation angle in Search Routine

\( \epsilon_6 \)  
Altitude increment in Search Routine

\( \epsilon_7 \)  
Tolerance on central angle in Desired Position Routine

\( \epsilon_8 \)  
Tolerance on central angle in Update Routine

\( \gamma \)  
Flight path angle

\( \mu \)  
Gravitational constant

\( \theta \)  
Central angle

\( \theta_p \)  
Perigee angle
<table>
<thead>
<tr>
<th>Subscripts</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>Final</td>
</tr>
<tr>
<td>i</td>
<td>Number of the maneuver</td>
</tr>
<tr>
<td>I</td>
<td>Initial</td>
</tr>
<tr>
<td>LOS</td>
<td>Line-of-sight</td>
</tr>
<tr>
<td>LV</td>
<td>Local vertical</td>
</tr>
<tr>
<td>N</td>
<td>New</td>
</tr>
<tr>
<td>o</td>
<td>Old</td>
</tr>
<tr>
<td>P</td>
<td>Primary</td>
</tr>
<tr>
<td>S</td>
<td>Stored</td>
</tr>
<tr>
<td>T</td>
<td>Target</td>
</tr>
<tr>
<td>TA</td>
<td>Target for primary vehicle</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

The rendezvous of the Shuttle (primary vehicle) with a target vehicle (e.g., the Space Station) is accomplished by maneuvering the Shuttle into a trajectory that intercepts the target vehicle orbit at a time that results in the rendezvous of the two vehicles. The function of rendezvous targeting is to determine the targeting parameters for the powered flight guidance for each of the maneuvers made by the Shuttle during the rendezvous sequence.

In order to construct the multimaneuver rendezvous trajectory, sufficient constraints must be imposed to determine the desired trajectory. Constraints associated with the Shuttle mission will involve such considerations as fuel, lighting, navigation, communication, time, and altitude. The function of premission analysis is to convert these—which are generally qualitative constraints—into a set of secondary quantitative constraints that can be used by the onboard targeting program. By judicious selection of the secondary constraints, it should be possible to determine off-nominal trajectories that come close to satisfying the primary constraints.

The proposed onboard rendezvous targeting program consists primarily of a main program and a generalized multiple-option maneuver subroutine. The driving program automatically and sequentially calls the maneuver subroutine to construct the rendezvous configuration from a series of maneuver segments. The main program is capable of handling rendezvous sequences involving any given number of maneuvers. Enough different types of maneuver constraints are incorporated into the subroutine to provide the flexibility required to select the best set of secondary constraints during premission planning. In addition, the astronaut has a large, well defined list of maneuver options if he chooses to modify the selected nominal rendezvous scheme.

As the new approach represents, in essence, just one targeting program, there is considerable savings in computer-storage requirements compared to former approaches in which each maneuver used in the rendezvous scheme had a separate targeting program. The programming and verification processes of this unified approach will also result in implementation efficiencies.

1.1 Number of Independent Constraints Involved in a Rendezvous Sequence

During the Gemini and Apollo flights and in the design of the Skylab rendezvous scheme various numbers of maneuvers were utilized in the rendezvous sequence. The range went from two (Apollo 14 and 15) to six (Skylab).
The number of independent constraints (i.e., the number of explicitly satisfied constraints) in each rendezvous sequence must equal the number of degrees of freedom implicitly contained in the sequence. To establish this number, a rendezvous configuration can be constructed by imposing arbitrary constraints until the configuration is uniquely defined. For example, a four maneuver coplanar sequence is shown in Figure 1, followed by a coast to a terminal point. Using the constraints $v_i$ (velocity magnitude), $r_i$ and $\theta_i$, it is easy to establish that the total number involved is 12, assuming the time of the first maneuver has been established. Removing one maneuver will reduce the number of degrees of freedom by three. Hence, the number of independent constraints necessary to uniquely determine the maneuver sequences are

<table>
<thead>
<tr>
<th>Number of maneuvers in sequence</th>
<th>Number of independent constraints required</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>etc</td>
<td></td>
</tr>
</tbody>
</table>

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

In some cases the number of primary constraints may be insufficient to uniquely determine a rendezvous trajectory for the desired number of maneuvers. One way of overcoming this deficiency in constraints is by introducing sufficient variables to complete the determination of the rendezvous trajectory and then determining values for these variables by minimizing the fuel used.

In order to take advantage of updated state vectors due to navigation or ground updates, the rendezvous targeting program is called prior to each maneuver to compute the upcoming maneuver. In general, each maneuver computation will involve a multimaneuver sequence as the nature of the targeting constraints do not allow the maneuvers to be independently computed. The relationship between the rendezvous sequence involving n maneuvers and the maneuver sequences is shown below.

```
Rendezvous Sequence

1 2 3 4 max1 ... n-1

Maneuver Sequences

{ 1 2 i max1 2 3 i max2 ...
   (maneuver points)   ...
   ...  ...
   n-1 }
```
Figure 1. A Possible Set of Constraints Involved in a Four Maneuver Rendezvous Sequence
Each maneuver sequence is composed of a number of maneuver segments and is basically independent from the other maneuver sequences. These sequences must have the same number of independent constraints as tabulated above.

1.2 The Construction of a Maneuver Segment

Each n-maneuver rendezvous sequence can be divided into n-1 maneuver segments. Each segment involves, basically, the addition of a maneuver to the primary vehicle's velocity vector and an update of both vehicle's state vectors to the next maneuver point.

A maneuver segment is herein generated in one of three ways:

**Forward generation**

A maneuver $\Delta v$ is computed and added to the velocity vector in a specified direction. The state vector of the primary vehicle is then updated through a specified amount to arrive at the next maneuver position.

**Target generation**

The target vehicle is updated through a specified amount and then offset to establish a target vector. An option is available at this point to compute a coelliptic velocity vector and update through $\Delta t$ to establish a new target vector as shown below.

The maneuver is then computed by uniquely specifying the nature of the traverse between the primary vehicle's position and the target vector.
Integrated generation

In this case, the maneuver segment is computed as an integral part of a maneuver sequence involving more than one maneuver segment. The nature of the constraints are such that the maneuver sequence cannot be subdivided into uniquely defined maneuver segments. The maneuver segment will usually have one degree of freedom, which will generally be assumed to be the magnitude of the maneuver.

Each of the above methods is defined by specifying trajectory constraints by setting certain switches and parameter values. Specifying a trajectory constraint is equivalent to specifying one or more independent constraints. On the other hand, specifying an independent constraint can also be equivalent to specifying one or more trajectory variables. (See Ref. 9) A trajectory constraint common to all three of the above methods is the state vector update switch $s_{\text{update}}$. The options associated with this switch are:

$$ s_{\text{update}} = \begin{cases} 
1 & \text{Update from time } t \text{ to time } t_F \\
2 & \text{Update through time interval } \Delta t \\
3 & \text{Update through } n \text{ revolutions} \\
4 & \text{Update through } \theta \text{ radians} \\
5 & \text{Update to be colinear with a specified position vector} 
\end{cases} $$

In the next three sections, the trajectory constraints associated with each of the above methods will be listed.
1.2.1 Maneuver Options in Forward Generation of Maneuver Segment

The forward generation of a maneuver segment is accomplished in one of two ways. Either the maneuver magnitude is uniquely determined in terms of the state vector at the maneuver time or the maneuver is determined by an iterative search to satisfy a terminal constraint.

The maneuver magnitude $\Delta v$ is either calculated or assumed depending on the maneuver switch $s_{man}$, and it is applied in a direction controlled by the direction switch $s_{direct}$. The options associated with the maneuver switch are:

$$s_{man} = \begin{cases} 
1 & \Delta v \text{ is assumed specified} \\ 
2 & \Delta v \text{ is computed based on a post maneuver velocity vector being "coelliptic" with the state vector of the target vehicle} \\ 
3 & \Delta v \text{ is computed from the conic circular velocity constraint} \\ 
4 & \Delta v \text{ is computed based on a Hohmann type transfer resulting in a $\Delta h$ change in altitude} 
\end{cases}$$

The options associated with the maneuver direction switch are:

$$s_{direct} = \begin{cases} 
1 & \text{Apply } \Delta v \text{ is horizontal direction in plane of primary vehicle} \\ 
1 & \text{Apply } \Delta v \text{ in horizontal direction parallel to orbital plane of the target vehicle} \\ 
2 & \text{Apply } \Delta v \text{ along velocity vector in plane of primary vehicle} \\ 
2 & \text{Apply } \Delta v \text{ along velocity vector parallel to orbital plane of the target vehicle} 
\end{cases}$$

The selection of the update switch $s_{update}$ determines the update of the primary vehicle's trajectory following the maneuver to the position of the next maneuver. A terminal constraint can be imposed at this point by setting the terminal switch $s_{term}$.
The target generation of a maneuver segment starts with the selection of the update switch for the target vehicle. If this switch equals four, $\theta$ will be augmented by the central angle between the primary and target vehicles before being used. The position of the target vehicle is then offset through either $(e_L, \Delta h)$ or $(\Delta \theta, \Delta h)$, depending on whether $s_{\text{tar}}$ is negative or positive, to obtain a target vector. The "TPI offsets" $(e_L, \Delta h)$ are discussed in Section 5 (see Figure 2 for definition of $e_L$). If $|s_{\text{tar}}|$ equals two, a coelliptic velocity vector is computed based on the target vector, and a new target vector is defined by updating the coelliptic state vector through $\Delta t$. The options associated with $s_{\text{tar}}$ are:

$$s_{\text{tar}} = \begin{cases} 
-2 & \text{Offset target} (e_L, \Delta h). \text{ Compute coelliptic velocity and update through (negative) } \Delta t. \\
-1 & \text{Offset target} (e_L, \Delta h) \\
0 & \text{No target offset} \\
1 & \text{Offset target} (\Delta \theta, \Delta h) \\
2 & \text{Offset target} (\Delta \theta, \Delta h). \text{ Compute coelliptic velocity and update through (negative) } \Delta t
\end{cases}$$

The nature of the traverse between the primary vehicle's initial state vector and the target vector is controlled by the maneuver switch $s_{\text{man}}$:

$$s_{\text{man}} = \begin{cases} 
5 & \text{Lambert - the trajectory is time constrained} \\
6 & \text{Horizontal - the maneuver is constrained to be in the horizontal direction} \\
7 & \text{Tangential - the maneuver is constrained to be in the direction of the velocity vector} \\
8 & \text{Apogee/Perigee - the trajectory has an apogee/perigee occurring at the target point}
\end{cases}$$

There is a minimum $\Delta v$ option associated with the above maneuvers which is controlled with the optimization switch $s_{\text{opt}}$.
Position Vector of Primary Vehicle

\[ \mathbf{i} = \text{Unit horizontal in forward direction for primary vehicle} \]

\[ \text{LOS} = \text{Line of Sight} \]

1. If the LOS projection on \( \mathbf{i} \) is positive:
   a. When the LOS is above the horizontal plane, \( 0 < e_L < \frac{\pi}{2} \)
   b. When the LOS is below the horizontal plane, \( \frac{3\pi}{2} < e_L < 2\pi \)

2. If the LOS projection on \( \mathbf{i} \) is negative:
   a. When the LOS is above the horizontal plane, \( \frac{\pi}{2} < e_L < \pi \)
   b. When the LOS is below the horizontal plane, \( \pi < e_L < \frac{3\pi}{2} \)

Figure 2. Definition of the Elevation Angle \( e_L \)
-2 Minimize the sum of the magnitude of the first and the next maneuver (based on a coelliptic parting velocity) by varying $\Delta t$, the time of update of the target vehicle.

-1 Minimize the magnitude of the first maneuver by varying $\Delta t$, the time of update of the target vehicle.

1 Minimize the magnitude of the first maneuver by varying $\Delta t$, the time between the next maneuver and the initial offset position. (See sketch on page 1-4)

2 Minimize the sum of the magnitudes of the first and the next maneuver (based on a coelliptic parting velocity) by varying $\Delta t$, the time between the next maneuver and the initial offset position (see sketch on page 1-4).

This minimization is accomplished by driving the slope ($\Delta v$/independent variable) to zero using a Newton-Raphson iteration scheme.

1.2.3 Maneuver Options in Integrated Generation of Maneuver Segment

The integrated generation of a maneuver segment involves an iterative solution to determine a maneuver sequence which cannot be sequentially solved for its maneuver segment components. The maneuver is computed by guessing its magnitude, assigning a direction and plane through selection of the direction switch $s_{\text{direct}}$, updating the primary vehicle's state vector after selecting switch $s_{\text{update}}$, and then calling additional maneuver segments until reaching the point at which the terminal constraint is to be attained. The maneuver is then iteratively determined by satisfying the terminal constraint. The number of additional maneuver segments and the nature of the terminal constraint are controlled by the terminal constraint switch $s_{\text{term}}$. 

1-9
The terminal constraint is a phasing constraint and it occurs at the \( s_{\text{term}} \) maneuver point from the start of the maneuver segment.

The terminal constraint is a height constraint and it occurs at the \( s_{\text{term}} \) maneuver point from the start of the maneuver segment.

Both a height and phasing constraint occur at the same maneuver point. The first digit \( n_1 \) of \( s_{\text{term}} \) represents a phasing constraint that occurs at the \( n_1 \) maneuver point from the start of the phasing maneuver segment. The last digit \( n_2 \) of \( s_{\text{term}} \) represents a height constraint that occurs at the \( n_2 \) maneuver point from the start of the height maneuver segment.

1.2.4 Summary of the Maneuver Segment Constraints

The maneuver and trajectory constraints that can be imposed on a maneuver segment can be divided into the following categories (see Figure 3).

- Primary vehicle update constraints
- Target vehicle update constraints
- Initial velocity constraints
- Offset constraints
- Terminal constraints
- Traverse constraints

Table 1 contains a detailed listing of the constraints. The three independent constraints (four in the case of noncoplanar traverses) which govern a maneuver segment cannot be chosen arbitrarily from this list for two reasons:

1. There is not a one-to-one correspondence between the trajectory constraints and the independent constraints.
Figure 3. Constraint Categories on a Maneuver Segment
Selecting some constraints negates the need for some others (e.g. selecting a Lambert constraint negates the need for a maneuver direction constraint).

In the case of a straight forward rendezvous profile, a basic understanding of the nature of the constraints should allow the constructor of the rendezvous sequence to choose a set of trajectory constraints which determine the required number of independent constraints. For a complex rendezvous profile, such as Skylab a more formal approach such as presented in Reference 11 should be used. One of the justifications for presenting the three methods of generating a maneuver segment was to aid the constructor of the rendezvous sequence in choosing compatible sets of constraints.
TABLE 1

DETAILED LISTING OF CONSTRAINTS
(Sheet 1 of 2)

Primary and Target Vehicle Update Constraints
- Delta time
- Initial and final time
- Central angle
- Number of revolutions
- Terminal position vector

Initial Velocity Constraints
- Plane
  - Parallel to target orbit
  - Parallel to primary orbit
- Direction
  - Horizontal
  - Along velocity vector
- Magnitude
  - Circular
  - Coelliptic
  - Altitude change
  - Specified

Offset Constraints
- Angle
- Altitude
- Elevation angle

Terminal Constraints
- Height
- Phase
**TABLE 1**

**DETAILED LISTING OF CONSTRAINTS**

(Sheet 2 of 2)

**Traverse Constraints**

- Minimum Fuel
  - One maneuver optimization
  - Two maneuver optimization
- Apogee/Perigee designation
- Horizontal maneuver
- Tangential maneuver
- Lambert (time)
2. FUNCTIONAL FLOW DIAGRAMS

The rendezvous targeting program consists of two major parts—a generalized maneuver subroutine which basically computes a maneuver and updates the state vectors of both vehicles to the time of the next maneuver and a main program which sequentially calls the subroutine to assemble a rendezvous sequence. These programs call a number of subroutines which are briefly described below and in detail in Section 5.

**Search** - To update the state vectors to either a specified apsidal crossing, a time, or an elevation angle.

**Phase Match** - To phase match the target vehicle's state vector to the primary vehicle's position vector.

**Desired Position** - To compute an offset target vector or a desired position to be used in a phasing constraint.

**Update** - To update a state vector through a specified interval.

**Coelliptic Maneuver** - To compute a coelliptic velocity vector.

**Iteration** - To determine a new estimate of the independent variable in a Newton-Raphson iteration scheme.

The functional flow diagram for the main program is shown in Figure 4. The main function of this program is to sequentially call the General Maneuver Routine to compute each maneuver segment for maneuvers numbered from $i$ to $i_{\text{max}}$. There are three major options that can be exercised prior to the calculation of the first maneuver segment:

1. A search for the time of the first maneuver.

   This time can be specified by:

   a. An elevation angle, which is to be attained at the maneuver time.
ENTER

Call Search Routine obtaining state vectors and time at next maneuver point.

Exit Switch =1 EXIT
=0

Phase Match Switch ≠ 0 Call Phase Match Routine obtaining modified target vehicle state vector
=0

Coplanar Switch =1 Rotate primary vehicle's state vector into orbital plane of target vehicle
=0

Desired Position Switch =1 Call Desired Position Routine obtaining desired position vector
=0

Optimize Switch ≠ 0 Iterate using GM Routine to find maneuver which minimizes fuel usage.
=0

Height Switch

Terminal Switch ≠ 0 Iterate using GM Routine to find maneuver which satisfies height constraint
=0

Phasing Switch Iterate using GM Routine to find maneuver which satisfies phasing constraint

Call General Maneuver (GM) Routine to obtain maneuver and time and state vectors at next maneuver point

```
\[ \text{i} = \text{i} + 1 \]
```

\[ \text{i} > \text{i}_{\text{max}} \]

Yes

Astronaut Overwrite

Yes Overwrite \( \Delta v \) in local-vertical or line-of-sight coordinates

No

Compute displays EXIT

Figure 4. Main Program - Function Flow Diagram
Whether the next maneuver should occur at the next apsidal crossing, the next perigee crossing or the nth apsidal crossing.

(2) A phase matching of the state vector.

(3) A rotation of the primary vehicle's state vector into the plane of the target vehicle.

There are three separate iterative loops built around the call to the general maneuver routine. One loop serves to minimize the fuel used during a maneuver segment with the options determined by the optimizing switch.

The other two iterative loops involve maneuver segments which contain constraints that do not allow the explicit calculation of the maneuver. These constraints are height and phasing constraints imposed at the end of a maneuver segment and controlled with the terminal switch. The iterative loop will involve several maneuver segments if sufficient constraints are not imposed to solve each segment uniquely.

The functional flow diagram for the general maneuver routine is shown in Figure 5. This routine generates the departure velocity at the initial point in one of two ways:

(1) As an explicit function of the initial state vectors.

(2) By defining a target vector and then computing an intercept trajectory based on a specified constraint (as indicated by the setting of $s_{\text{man}}$). The target vector is determined by offsetting the updated position vector of the target vehicle. Depending on the setting of the switch $s_{\text{tar}}$, a coelliptic velocity vector is computed at the offset point and the coelliptic state vector is updated through $\Delta t$ to obtain a target vector.

Following an update of both vehicle's state vectors to the time of the next maneuver, the $\Delta v$ used or the terminal height/phase errors are calculated as required.
ENTER

Direction Switch

\( \neq 0 \)

Determine direction maneuver is to be applied

\( = 0 \)

\(< 5 \)

Determine final velocity based on present state

\( > 5 \)

Call Update Routine to update passive state vector

Call Desired Position Routine to obtain offset target vector

\( \neq 2 \)

Terminal Offset Switch

\( = 2 \)

Call Coelliptic Maneuver Routine to obtain primary's velocity vector

Call Update Routine to obtain new target vector

Determine departure velocity based on offset target and setting of \( s_{\text{man}} \)

Call Update Routine to update active and passive state vectors

Compute the maneuver

Optimize Switch

\( \neq 0 \)

Compute required velocity

\( = 0 \)

Termain Switch

\( \neq 0 \)

Compute error in height/phase as required

\( = 0 \)

RETURN

Figure 5. General Maneuver Routine - Functional Flow Diagram
3. **INPUT AND OUTPUT VARIABLES**

The inputs to the orbiter rendezvous targeting program can be divided into five categories.

**Pre-Maneuver Switches**

Upon selecting a maneuver from the rendezvous sequence, these switches (specified for each maneuver) serve in determining the state vectors at the maneuver point, the out-of-plane parameters and the calculation of a desired position vector. These inputs can also be used in determining the time of a specified apsidal crossing or the time at which a specified elevation angle is to be attained.

\[
\begin{align*}
\text{Coplanar switch} \\
{s}_{\text{coplan}} = & \\
& \begin{cases}
0 & \text{Bypass} \\
1 & \text{Rotate primary state vector into plane of target vehicle's orbit}
\end{cases} \\
\text{Exit switch} \\
{s}_{\text{exit}} = & \\
& \begin{cases}
0 & \text{Bypass} \\
1 & \text{Exit from routine}
\end{cases} \\
\text{Out-of-plane switch} \\
{s}_{\text{outp}} = & \\
& \begin{cases}
0 & \text{Bypass} \\
1 & \text{Compute out-of-plane parameters} \\
2 & \text{Compute out-of-plane parameters and modify maneuver by } -\dot{y}_P
\end{cases} \\
\text{Perturbation switch} \\
{s}_{\text{pert}} = & \\
& \begin{cases}
0 & \text{Do conic state vector updates} \\
1 & \text{Include oblateness based on } J_2 \\
\ldots & \text{Other perturbations as required}
\end{cases} \\
\text{Phase match switch} \\
{s}_{\text{phase}} = & \\
& \begin{cases}
0 & \text{Bypass} \\
1 & \text{Phase match state vectors (target leading primary)} \\
2 & \text{Phase match state vectors based on target leading primary by more than } 360^\circ \\
-1 & \text{Phase match state vectors (primary leading target)} \\
-2 & \text{Phase match state vectors based on primary leading target by more than } 360^\circ
\end{cases}
\end{align*}
\]
Desired position switch

\[ s_{\text{rdes}} = \begin{cases} 
0 & \text{Bypass} \\
1 & \text{Compute desired position vector based on } (e_L, \Delta h) \\
1 & \text{Compute desired position vector based on } (\Delta \theta, \Delta h) 
\end{cases} \]

Search switch

\[ s_{\text{search}} = \begin{cases} 
-4 & \text{Compute elevation angle} \\
-3 & \text{Search for elevation angle} \\
-2 & \text{Update to time } t_i \\
-1 & \text{Search for next perigee crossing} \\
0 & \text{Bypass} \\
 n & \text{Search for the } n\text{th apsidal crossing } (n>0) 
\end{cases} \]

Maneuver Switches

These switches (specified for each maneuver) set the constraints employed in determining the maneuver segments.

Direction switch

\[ s_{\text{direct}} = \begin{cases} 
-2 & \Delta v \text{ in direction of primary's velocity vector, parallel to primary's orbital plane} \\
-1 & \Delta v \text{ in horizontal direction, parallel to primary's orbital plane} \\
0 & \text{Bypass} \\
1 & \Delta v \text{ in horizontal direction, parallel to target's orbital plane} \\
2 & \Delta v \text{ in direction of primary's velocity vector, parallel to target's orbital plane} 
\end{cases} \]

Maneuver switch

\[ s_{\text{man}} = \begin{cases} 
1 & \Delta v \text{ is specified} \\
2 & \Delta v \text{ is based on coelliptic velocity} \\
3 & \Delta v \text{ is based on circular velocity} \\
4 & \Delta v \text{ is based on altitude change} \\
5 & \text{Lambert maneuver to offset target vector} \\
6 & \text{Horizontal maneuver to offset target vector} \\
7 & \text{Tangential maneuver to offset target vector} \\
8 & \text{Perigee/apogee insertion at offset target vector} \\
9 & \text{Plane change} 
\end{cases} \]
Maneuver optimizing switch

\[ s_{opt} = \begin{cases} 
0 & \text{Bypass} \\
1 & \text{Minimize } \Delta v_i \\
2 & \text{Minimize } \Delta v_i + \Delta v_{i+1} 
\end{cases} \]

Multi-revolution solution switch

\[ s_{soln} = \begin{cases} 
-1 & \text{Solution with smallest initial flight path angle (measured from local vertical)} \\
1 & \text{Solution with largest initial flight path angle} 
\end{cases} \]

\[ s_{tar} = \begin{cases} 
-2 & \text{Offset target } (e_L, \Delta h). \text{ Compute coelliptic velocity and update through (negative) } \Delta t \\
-1 & \text{Offset target } (e_L, \Delta h) \\
0 & \text{No target offset} \\
1 & \text{Offset target } (\Delta \theta, \Delta h) \\
2 & \text{Offset target } (\Delta \theta, \Delta h). \text{ Compute coelliptic velocity and update through (negative) } \Delta t 
\end{cases} \]

Terminal constraint switch

\[ s_{term} = \begin{cases} 
\text{for } n < 0 \text{ Compute phasing error and back up} \\
\text{for } n = 0 \text{ Bypass} \\
\text{for } 0 < n < 10 \text{ Compute height error and back up} \\
\text{for } 10 < n < 100 \text{ Phase and height loop terminate on same maneuver. For phase loop back up} \\
\text{for } n \text{ (x-1 where x is first digit of n) maneuvers for start of phase loop. For height loop back up} \\
\text{for } n \text{ (y-1 where y is last digit of n) maneuvers for start of height loop} 
\end{cases} \]

Update switch

\[ s_{update} = \begin{cases} 
1 & \text{Update through } t_F - t \\
2 & \text{Update through } \Delta t \\
3 & \text{Update through } n_T \\
4 & \text{Update through } \theta \\
5 & \text{Update to be colinear with } \mathbf{r}_D 
\end{cases} \]

Parameter Values

The parameter values (specified for each maneuver) are values for the constrained parameters.
\[ \Delta h \] Delta altitude
\[ \Delta h_F \] Delta altitude, final
\[ \Delta \theta \] Delta central angle
\[ \Delta t \] Delta time
\[ \Delta v \] Maneuver magnitude
\[ n_r \] Number of revolutions
\[ t_F \] Final time
\[ e_L \] Elevation angle

Post-Maneuver Switch

This switch determines the options available following the calculation of the maneuver.

\[
\text{Astronaut overwrite switch}
\begin{cases}
0 & \text{Bypass} \\ 
1 & \text{Overwrite maneuver in local vertical coordinates} \\ 
2 & \text{Overwrite maneuver in line-of-sight coordinates}
\end{cases}
\]

Maneuver Call Variables

The maneuver call variables have to be specified for each call to the maneuver sequence.

\[ \Sigma P, \nu_P \] State vector of the primary vehicle
\[ \Sigma T, \nu_T \] State vector of the target vehicle
\[ i \] Maneuver number
\[ t \] Current time
\[ t_i \] Time of the \( i \)th maneuver (if \( s_{\text{search}} \neq -2 \), then \( t_i \) is not specified)
\[ s_{\text{eng}} \] Engine select switch
\[ m \] Estimated vehicle mass

Depending on the rendezvous sequence, there may also be some switches that have to be modified as a function of the maneuver number.

Excluding the maneuver call variables, all the input variables can be set prior to the flight.

The output parameters for the initial maneuver in the sequence are more complete than for the succeeding maneuvers.
Output Parameters for the Initial Maneuver

\[ \Delta v_i \] Maneuver magnitude

\[ \Delta v_{\text{LOS} i} \] Maneuver in line of sight coordinates

\[ \Delta v_{\text{LV} i} \] Maneuver in local vertical coordinates

\[ \mathbf{r}_{1c} \] Target vector used in Powered Flight Guidance Routine (See Ref. 5)

\[ \mathbf{n}_N \] Unit normal to plane used in same routine

\[ t_T \] Time associated with \( \mathbf{r}_{1c} \)

Other parameters such as delta altitude, phasing angle, elevation angle and perigee altitude can be computed as required.

Output Parameters for the Other Maneuvers in the Sequence

\[ t \] Time of the maneuver

\[ \Delta v \] Maneuver magnitude

Illustration of Inputs

Table 2 contains a set of inputs for the Shuttle targeting program based on the five maneuver Skylab rendezvous configuration. The following switches and parameters are not used as inputs to the Orbiter program:

\[ s_{\text{astro}} , s_{\text{exit}} , s_{\text{opt}} , s_{\text{outp}} , s_{\text{soln}} , \Delta \theta \]

The inputs in Table 2 are set prior to the mission so they will not have to be inserted by the astronaut. The astronaut will have to modify the following quantities upon resetting the maneuver number as well as inserting the time of the next maneuver.

\[ i = 2: \ s_{\text{term}_2} = 0, \ s_{\text{term}_4} = 32 \]

\[ i = 3: \ s_{\text{man}_3} = 5, \ \Delta t_3 = -\Delta t_{\text{NSR-TPI}} \]

\[ i = 4: \ s_{\text{term}_4} = 0 \]
### TABLE 2

**INPUT VARIABLES FOR SKYLAB RENDEZVOUS CONFIGURATION**

<table>
<thead>
<tr>
<th>Maneuver</th>
<th>1 (NC1)</th>
<th>2 (NC2)</th>
<th>3 (NCC)</th>
<th>4 (NSR)</th>
<th>5 (TPI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_coplan</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>s_direct</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>s_man</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>s_pert</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>s_phase</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>s_rdes</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>s_search</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
</tr>
<tr>
<td>s_star</td>
<td></td>
<td>0</td>
<td>-2</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>s_term</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>42</td>
<td>0</td>
</tr>
<tr>
<td>s_update_p</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>s_update_T</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>e_L</td>
<td>e_L</td>
<td>e_L</td>
<td>e_L</td>
<td>e_L</td>
<td></td>
</tr>
<tr>
<td>i_max</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>n_r</td>
<td>n_rNC1-NC2</td>
<td>n_rNC2-NCC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t_F</td>
<td>t_TPI</td>
<td>t_TPI</td>
<td>t_TPI</td>
<td>t_TPI</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>9_TPF-TPI</td>
<td></td>
</tr>
<tr>
<td>Δt</td>
<td></td>
<td>Δt_NS-R-NCC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Δh</td>
<td>Δh_NCC</td>
<td>Δh_TPI</td>
<td>Δh_TPI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Δh_F</td>
<td>Δh_TPI</td>
<td>Δh_TPI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Δv</td>
<td>Δv_NC1</td>
<td>Δv_NC2</td>
<td>Δv_NCC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3-6
4. DESCRIPTION OF EQUATIONS

The only equations contained in this document which are not trivial are those involved in computing the traverse between two specified position vectors. The required equations can be derived from the equation of the conic expressed in the form

\[ r = r_F / r_I = \beta_I^2 / \left[ 1 + \frac{e_c \cos (\Theta + \theta_p)}{1 + e_c \cos (\Theta + \theta_p)} \right] \]

where

- \( e_c = \left[ \alpha_I^2 \beta_I^2 + (\beta_I^2 - 1)^2 \right]^{1/2} \)
- \( \theta_p = \cos^{-1}\left( (\beta_I^2 - 1)/e_c \right) \) (perigee angle)
- \( v_c = (\mu / r_I)^{1/2} \)

\( \alpha_I \) and \( \beta_I \) are the normalized (with respect to \( v_c \)) radial and horizontal components of velocity.

The above equation can be expressed

\[ \frac{p_s}{r_I} = \beta_I^2 = \frac{(\alpha_I \beta_I \sin \Theta - c_2)}{c_1} \]  \( (1) \)

where

- \( c_1 = \cos \Theta - 1/r \)
- \( c_2 = 1 - \cos \Theta \)
- \( p_s = \text{semilatus rectum} \)

For a maneuver that is constrained to be in a horizontal direction, Eq. (1) can be solved for \( \beta_I \) as a function of the specified \( \alpha_I \).

\[ \beta_I = \left[ \alpha_I \sin \Theta \pm (\alpha_I^2 \sin^2 \Theta - 4 c_1 c_2)^{1/2} \right] / 2 c_1 \]

As there has to be both a positive and negative \( \beta_I \) solution to this equation (one trajectory in each rotational direction), the sign choice is resolved in favor of plus \( \beta_I \).

For a maneuver that is applied along the velocity vector, the flight path angle \( \gamma_I \) is to be held fixed. Using Eq. (1)
\[ \tan \gamma_1 \beta_1 = \alpha_1 \beta_1 = (c_1 \beta_1^2 + c_2)/\beta_1^2 \sin \theta \]

Therefore,

\[ \beta_1 = \left[ c_2/(\sin \theta \tan \gamma_1 - c_1) \right]^{1/2} \]

By interchanging the I and F subscripts, Eq. (1) can be expressed

\[ p_s = r_F (\alpha_F \beta_F \sin \theta - c_2)/(\cos \theta - r) \]

Combining with Eq. (1) using the apogee/perigee constraint \( \alpha_F = 0 \) results in the required horizontal and radial components of velocity for apogee/perigee designation maneuvers.

\[ \beta_1 = [r c_2/(r - \cos \theta)]^{1/2} \]
\[ \alpha_1 = \beta_1 (1 - 1/r)/\tan (\theta/2) \]

The derivation of the equation

\[ \theta = \cos^{-1} \left( r_P \cos (e_{L'})/r_T \right) - e_{L'} \]

where

\[ e_{L'} = \begin{cases} e_L & \text{if } e_L \leq \pi \\ e_L - \pi & \text{if } e_L > \pi \end{cases} \]

for computing the desired central angle \( \theta \) between two positions \( (r_P, e_T) \) which satisfies the TPI constraints \((e_L, \Delta h)\) is discussed in Ref. 12. This equation is used in the Desired Position Routine.
5. DETAILED FLOW DIAGRAMS

Figures 6 and 7 contain the detailed flow diagrams of the main Shuttle rendezvous targeting program and the general maneuver routine, respectively. The following six routines are called by these two programs.

Iteration Routine

This routine contains a Newton Raphson iterative driver based on numerically computed partials. The routine computes a new estimate of the dependent variable \( x \) and returns the old values of the error \( e \) and \( x \). If the iteration counter \( c \) exceeds 15, a convergence switch \( s_{\text{fail}} \) is set equal to one.

Coelliptic Maneuver Routine

This routine computes a coelliptic velocity vector \( \mathbf{v}_N \) based on a target vehicle's state vector and a delta altitude.

Phase Match

This routine phase matches the target state vector to the primary state vector. The controlling switch \( s_{\text{phase}} \) equals two if the leading vehicle leads the other vehicle by more than one revolution; otherwise the switch equals one. If the primary vehicle leads to target vehicle, the switch is negative.

Desired Position Routine

This routine updates a specified state vector to the time \( t_F \) and then offsets the updated state vector through either \( (\Delta \theta, \Delta h) \) or \( (e_L, \Delta h) \), depending on the setting of the switch \( s \), to obtain \( r_D \). The routine contains an iterative search to solve the \( (e_L, \Delta h) \) offset problem, where \( e_L \) is defined in Figure 2 and \( \Delta h \) (positive when the target orbit is above the primary) is defined as shown below. (This represents the TPI geometry used in Apollo and Skylab.)

![TPI point on target orbit diagram](image-url)
**Update Routine**

This routine updates a state vector based on the update switch $s_{\text{update}}$

- $s_{\text{update}} = 1$ Updates through the time $t_F - t$
- $s_{\text{update}} = 2$ Updates through the time $\Delta t$
- $s_{\text{update}} = 3$ Updates through $n_F$ revolutions
- $s_{\text{update}} = 4$ Updates through the angle $\theta$
- $s_{\text{update}} = 5$ Updates to where the orbit intersects the line defined by $r_D$

**Search Routine**

This routine makes the following computations depending on the setting of the search switch $s_{\text{search}}$

- $s_{\text{search}} = n$ Finds the time of the $n$th apsidal crossing $(>0)$ and updates the state vector to that time
- $s_{\text{search}} = -1$ Finds the time of the next perigee crossing and updates the state vector to that time
- $s_{\text{search}} = -2$ Updates the state vector through the time $t_F - t$ and computes the elevation angle
- $s_{\text{search}} = -3$ Finds the time associated with a specified elevation angle and updates the state vector to that time

The detailed flow charts for these routines are shown in Figures 8 to 13. The iterative algorithm used to determine the time associated with the elevation angle is described in Ref. 8.

Each input and output variable in the routine and subroutine call statements can be followed by a symbol in brackets. This symbol identifies the notation for the corresponding variable in the desired description and flow diagrams of the called routine. When identical notation is used, the bracketed symbol is omitted.
INPUT VARIABLES

**UNIVERSAL PROGRAM**

For each $i$ ($1 \leq i \leq$ total number of maneuvers):

- $s_{\text{astro}}$, $s_{\text{coplan}}$, $s_{\text{direct}}$, $s_{\text{exit}}$, $s_{\text{man}}$
- $s_{\text{opt}}$, $s_{\text{outp}}$, $s_{\text{pert}}$, $s_{\text{phase}}$, $s_{\text{search}}$, $s_{\text{soln}}$, $s_{\text{rdes}}$, $s_{\text{star}}$
- $t_{F}$, $\theta$, $\Delta h$, $\Delta \theta$, $\Delta t$, $\Delta v$

Set $f$ according to $s_{\text{eng}}$: $a_{T} = f g_{0}/m$

**Call Search Routine**

Input: $x_{P}$, $y_{P}$, $x_{T}$, $y_{T}$, $t$, $t_{i}$, $s_{\text{pert}}$, $s_{\text{search}}$, $e_{L_{i}}$

Output: $x_{Pi}$, $y_{Pi}$, $x_{Ti}$, $y_{Ti}$, $t_{i}$, $e_{LN}$

**Call Phase Match Routine**

Input: $x_{Pi}$, $y_{Pi}$, $x_{Ti}$, $y_{Ti}$, $t_{i}$, $s_{\text{pert}}$, $s_{\text{phase}}$

Output: $x_{Ti}$, $y_{Ti}$

**Call Desired Position Routine**

Input: $x_{Ti}$, $y_{Ti}$, $t_{i}$, $s_{\text{pert}}$, $s_{\text{rdes}}$

Output: $x_{D}$
Call Iteration Routine

Input: $c, e, t_s(x), e_0, t_{S0}(x_0), \Delta t_i$
Output: $c, e, t_s(x), e_0, t_{S0}(x_0), s_{fail}$

Set Alarm
Code $a_1$

Figure 6b. Main Program - Detailed Flow Diagram
Figure 6c. Main Program - Detailed Flow Diagram
\[ i_z = - \text{unit}(r_{Pi}); i_y = \text{unit}(y_{Ti} \times r_{Ti}); i_x = i_y \times i_z \]

\[ M_{LV} = \begin{bmatrix} i_x & i_y & i_z \end{bmatrix}; \Delta y_{LV_i} = M_{LV}^T \Delta v_i + (s_{outp}^{-1})(0, -y_P, 0) \]

\[ i_x = \text{unit}(r_{Ti} \times r_{Pi}); i_y = \text{unit}\left(\left[(r_{Pi} \times y_{Pi}) \times i_x\right] \times i_x\right); i_z = i_x \times i_y \]

\[ M_{LOS} = \begin{bmatrix} i_x & i_y & i_z \end{bmatrix}; \Delta y_{i} = M_{LV} \Delta y_{LV_i}; \Delta y_{LOS_i} = M_{LOS}^T \Delta v_i \]

\[ s_{astro} = 1 \]

Input: \( \Delta y_{LV_i} \)

\[ s_{astro} = 2 \]

Input: \( \Delta y_{LOS_i} \)

\[ \Delta y_{i} = M_{LV} \Delta y_{LV_i}; \Delta y_{LOS_i} = M_{LOS}^T \Delta v_i \]

\[ s_{man} = 5 \]

Call Conic State Extrapolation (Kepler) Routine (Ref. 6)

Input: \( r_{Pi}[r_0] \), \( y_{Pi} + \Delta y_{i} \) \( \{y_0\} \), \( (t_{i+1} - t_i) \) \( \{\Delta t\} \)

Output: \( r_{1c}[r] \)

Compute Desired Displays

EXIT

Figure 6d. Main Program - Detailed Flow Diagram
Figure 7a. General Maneuver Routine - Detailed Flow Diagram
Figure 7b. General Maneuver Routine - Detailed Flow Diagram
\[ \theta = \text{sign}\left(\mathbf{x}_p \times \mathbf{x}_{TA}\right) \cdot \left(\mathbf{r}_p \times \mathbf{v}_p\right) \cdot \cos^{-1}\left(\mathbf{x}_p \cdot \mathbf{x}_{TA}/r_p r_{TA}\right) - \pi + \pi \]

\[ r = r_{TA}/r_p; \quad v_c = (\mu/r_p)^{1/2}; \quad c_1 = \cos \theta - 1/r; \quad c_2 = 1 - \cos \theta \]

\[ \alpha = \beta (1-1/r)/\tan(\theta/2) \]

\[ \beta = [r c_2/(r - \cos \theta)]^{1/2} \]

\[ \alpha = \beta c_3 \]

\[ \beta = (\alpha \sin \theta - c_3)/2 c_1 \]

\[ \Delta v = v_{PF} - v_F \]

\[ r_p = r_p - (\Delta v/2 a_t) \Delta v \]

\[ s_{\text{update}} = 2 \]

\[ c_3 = \mathbf{x}_p \cdot \mathbf{v}_p/r_p (\mathbf{v}_p \cdot \mathbf{i}) \]

\[ \beta = [c_2/(c_3 \sin \theta - c_1)]^{1/2} \]

\[ s_{\text{update}} = 5 \]

\[ \mathbf{i} = \text{unit}\left(\mathbf{v}_{TN} \times \mathbf{r}_{TN}\right); \quad \Delta v = -\mathbf{v}_{PN} \cdot \mathbf{i}; \quad \Delta v = \Delta v \mathbf{i} \]

Figure 7c. General Maneuver Routine - Detailed Flow Diagram
Call Update Routine
Input: \( r_{TN}[x] \), \( x_{TN}[y] \), \( x_{PN}[x] \), \( t_N[t] \), \( s_{pert} \), \( s_{update} = 5 \)
Output: \( x_{TF}[x_F] \), \( x_{TF}[y_F] \)

Call Coelliptic Maneuver
Input: \( x_{TF}[x] \), \( x_{TF}[y] \), \( (r_{TF} - r_{PN})[\Delta h] \)
Output: \( x_{PNF}[x_N] \)

\[ \Delta v = \Delta v \cdot \text{sign} (\Delta v \cdot j) \]

\( s_{opt} = 0 \)
\( s_{opt} = 1 \)
\( s_{opt} = 2 \)

\[ \Delta v_T = |\Delta v| \]

\[ \Delta v_T = |\Delta v| + |x_{PNF} - x_{PN}| \]

\[ e_h = r_{TNF} - r_{PN} - \Delta h \]

\[ e_p = \text{sign} \left( (x_{PN} \times x_D) \cdot (x_{PN} \times x_{PN}) \right) \cos^{-1} \left( x_{PN} \cdot x_D / r_{PN} \cdot r_D \right) \]

Output Variables:
\( x_{PN'}, y_{PN'}, x_{TN'}, y_{TN'}, t_{1c'}, \Delta y, t_N', \Delta v_T, e_h, e_p, s_{proj}, \Delta r_{proj} \)

Figure 7d. General Maneuver Routine - Detailed Flow Diagram

5-10
INPUT VARIABLES

\[ c, e, x, e_0, x_0, s_{\text{fail}} = 0, \Delta x \]

\[ \neq 0 \]
\[ c = 0 \]
\[ e - e_0 \]
\[ \neq 0 \]
\[ p = (e - e_0)/(x - x_0) \]
\[ \Delta x = e/p \]
\[ c = c + 1; \quad e_0 = e; \quad x_0 = x, \quad x = x - \Delta x \]
\[ \text{if } c > 15 \]
\[ s_{\text{fail}} = 1 \]
\[ \text{if } c < 15 \]

OUTPUT VARIABLES

\[ c, x, e_0, x_0, s_{\text{fail}} \]

Figure 8. Iteration Routine - Detailed Flow Diagram

UNIVERSAL CONSTANTS

\[ \mu \]

\[ \sum_\var_ \nu_ \Delta h \]

\[ a = 1/(2/r - \nu \cdot \nu / \mu); \quad c_1 = a - \Delta h \]

\[ \nu_\var = \nu \cdot \var (a / c_1)^{1.5} / \var; \quad \var_N = \var - \Delta h \text{ unit } (\var) \]

\[ \var_N = \left[ \mu (2/\var_N - 1/c_1) - \nu_\var^2 \right]^{1/2} \text{ unit } ((\var \times \var) \times \var) + \nu_\var \var_N / \var_N \]

OUTPUT VARIABLES

\[ \var_N \]

Figure 9. Coelliptic Maneuver Routine - Detailed Flow Diagram

5-11
Figure 10. Phase Match Routine - Detailed Flow Diagram
Figure 11. Desired Position Routine - Detailed Flow Diagram
Figure 12a. Update Routine - Detailed Flow Diagram
\[
\theta = \text{sign}[(r_S \times r) \cdot (r \times v)] \left[ \cos^{-1}\frac{r \cdot r_S}{|r_S|} - \pi \right] + \pi
\]

\[
\Delta t = t_F - t_S
\]

\[
\tau_F = \tau, \quad v_F = v
\]

**OUTPUT VARIABLES**

\[
\tau_F, \quad v_F, \quad \Delta t, \quad t_F
\]

**Figure 12b. Update Routine - Detailed Flow Diagram**
UNIVERSAL PROGRAM

CONSTANTS

C04' E5' 66A maxj rpp'IT

INPUT VARIABLES

VT' t, ti'

spert ' Ssearch' eL

tS = t; c = c1 = 0

v = p

p = Pa = 1/(2/rp - vp)

search = -lor>O

c1 = rpr 2/P; c2 = (1 - c1/a)1/2

C2 No

IV I es IE = 500

Yes

C2 < .0005

0.0005

No

2 m/sec

Yes

\( \epsilon_4 = 500 \)

\( \theta = |s_{search}|'\pi \)

\( c = 0 \)

\( \theta = - \theta \) sign \( v_v \)

\( \theta = (\pi - \theta) \) sign \( v_v \)

\( v_v > 0 \)

\( v_v < 0 \)

\( v_v = 0 \)

\( \theta = \theta + \pi \)

\( \theta = \theta + \pi (s_{search} - 1) \)

\( \theta = \pi - \theta \)

Call Update Routine

Input: \( r_p [r], v_p [v], \theta, s_{pert} = 0, s_{update} = 4 \)

Output: \( r_{FS}[r], v_{PS}[v], \Delta t \)

\( \frac{r_p - r_{PS}}{v_p} = v_{PS} \)

\( s_{pert} = 0 \)

\( t_1 = t + \Delta t \)

\( e_L \)

Figure 13a. Search Routine - Detailed Flow Diagram
Call Update Routine

Input: $r_P \{x\}, y_P \{y\}, t, \Delta t, s_{\text{pert}}, S_{\text{update}} = 2$

Output: $r_P \{x_F\}, y_P \{y_F\}, t[t_F]$}

| $<\epsilon_4$ | $c = c + 1$ | $|\beta| < \epsilon_4$ | No | $c > 10$ | Set Alarm Code $a_5$ |
| $<10$ | | | Yes | $c < 10$ |

---

Call Update Routine

Input: $r_P \{x\}, y_P \{y\}, t, t_i \{t_F\}, s_{\text{pert}}, S_{\text{update}} = 1$

Output: $r_P \{x_P\}, y_P \{y_P\}$

Input: $r_T \{x\}, y_T \{y\}, t, t_i \{t_F\}, s_{\text{pert}}, S_{\text{update}} = 1$

Output: $r_T \{x_F\}, y_T \{y_F\}$

---

$\text{iLOS} = \text{unit} (r_T - r_P); i = \text{unit} \left[ \text{iLOS} \cdot \text{iLOS} \cdot r_P/r_P^2 \right]$

$e_{\text{LN}} = \cos^{-1} \left[ \text{iLOS} \cdot \text{i sign} \left( \text{i} \cdot \left( r_P \times y_P \right) \times r_F \right) \right]$

---

Figure 13b. Search Routine - Detailed Flow Diagram

5-17
Call Iteration Routine
Input: c, e, t₁[x], e₀, t[x₀], ⁴
Output: c, t₁[x], e₀, t[x₀], sfail

|e| < ε₅

|t₁ - t| > Δtₘₐₓ

Call Update Routine
Input: Eₜ[E], Yₜ[V], tₛ[t], t₁[tₚ], sₚᵉʳᵗ, s_update = 1
Output: Eₜ[Eₚ], Yₜ[Vₚ]

OUTPUT VARIABLES
Eₚ, Yₚ, Eₜ, Yₜ, t₁, e_LN

Set Alarm
Code a₇
EXIT

Figure 13c. Search Routine - Detailed Flow Diagram
6. SUPPLEMENTARY INFORMATION

The Shuttle rendezvous targeting program proposed herein uses the basic targeting philosophy employed in Apollo and Skylab. (See Ref. 3 and 4.) The Shuttle program represents, to some degree, a general solution to the rendezvous targeting problem. The program has the capability of solving the Apollo and Skylab rendezvous configurations as well as many other configurations that can be determined by specifying sets of secondary constraints. The constraints contained in this document may require modification as experience is gained in generating sets of secondary constraints based on a variety of Shuttle missions. Reference 9 contains a more detailed discussion of the nature and number of the secondary constraints employed in the maneuver sequences. Reference 11 is a users' guide to the Shuttle targeting program.

The targeting program logic has been verified by a computer simulation that disclosed no iterative convergence problems for the range of trajectories considered. References 10 and 13 contain simulations of the Skylab five-maneuver rendezvous configuration and the six maneuver (through TPI) "Mission 2" shuttle rendezvous, respectively. This program is currently being integrated with the Orbiter navigation programs to verify their combined effectiveness on a number of Orbiter missions.

Some of the maneuvers described in this document were based on conic orbit assumptions although the conic/precision switch was set to indicate that oblateness should be considered. One example is the computation of the "circular orbit" maneuver based on the conic circular velocity vector. This calculation can be modified to provide a maneuver that would result in a minimum altitude change orbit in an oblate gravity field. (See Ref. 2.)

As TPI is generally desired at the midpoint of darkness, a stored value for the sun's position vector could be used to generate a display giving the angular distance between TPI and the midpoint of darkness. This vector could also be used to directly define the location of the TPI point (using the Update Routine option of updating to be colinear with a specified vector).
A decision has not been reached on a final listing of the desired astronaut displays. As no program alarms are set for low perigees (following Skylab practice), the perigee altitudes for each maneuver segment contained in the maneuver sequence can be displayed. (If the orbits and maneuver magnitudes are approximately nominal, the astronaut can probably assume that the perigee altitudes will be acceptable.)

Some standard orbit shaping maneuvers (e.g., circularization, altitude change) are included as rendezvous targeting maneuvers in this document. Hence, the rendezvous targeting program as herein contained has a limited capability to perform orbit shaping maneuvers assuming a dummy target vehicle state vector is used. The program can be expanded to allow the computation of the orbit shaping maneuvers for the shuttle mission by eliminating the need for the dummy state vector and adding additional options for orbit shaping maneuvers. However, the additional program complexity and the large number of inputs that have to be set nominally equal to zero do not seem to justify a unified approach to the calculation of the rendezvous targeting and orbit shaping maneuvers. Once all the orbit shaping maneuvers are defined, a separate Space Shuttle GN&C Equation Document will be issued for orbit shaping targeting. This document might utilize some of the routines herein contained.
REFERENCES


