RTCC REQUIREMENTS FOR
APOLLO 14 (MISSION H-3):
LOI TARGETING
REVISION 1

Lunar Mission Analysis Branch
MISSION PLANNING AND ANALYSIS DIVISION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS
Addressees:
Bellcomm/K. Martersteck
IBM Library
IBM/J. Hoskins (5)
S. E. James (3)
G. E. Myers
M. E. Jones
TRW Library
TRW/D. P. Johnson (3)
D. M. Gafford (2)
F. A. Evans (10)
M463/Technical Library (2)
CP/W. J. North
EA2/R. A. Gardiner
EG/R. G. Chilton
EG2/D. C. Cheatham
KA/K. S. Kleinknecht
KM/W. B. Evans
PA/J. A. McDivitt
PD/K. Thomas
FD/R. H. Kohrs
FD8/T. P. Loftus, Jr.
PD4/R. W. Kubicki
PF/A. Cohen
FA/S. A. Sjoberg
R. G. Rose
C. C. Critzos
FA4/C. R. Hicks
FC/E. F. Kranz (5)
M. P. Frank
J. C. Bostick
E. L. Pavelka, Jr.
H. D. Reed
W. M. Stoval
W. J. Boone, III
 FS/L. C. Dunseith
FS5/J. C. Stokes, Jr.
L. J. Dungan
R. Arndt
K. Leach
F. Williams
FM/J. P. Mayer
C. R. Huss
Branch Chiefs
FM4/J. C. McPherson
FM6/R. R. Regelbrugge
W. Sullivan
FM13/R. P. Parten
K. Henley

FM15/Report Control Files (25)
FM5/J. D. Yencharis
Q. A. Holmes
K. T. Zeiler
R. S. Davis
R. H. Sanders

FM54:RFWiley:jrh
MEMORANDUM TO: See attached list

FROM: FM5/Chief, Lunar Mission Analysis Branch

SUBJECT: RTCC requirements for Apollo 14 (H-3): LOI targeting

The enclosed MSC Internal Note 70-FM-145 updates and supercedes MSC Internal Notes 69-FM-11, 69-FM-327, and 70-FM-81. The enclosed internal note specifies the RTCC LOI targeting processor logic for Apollo 14. The enclosed internal note corrects mistakes and documents necessary formulation changes found in programming and using the LOI targeting processor, and reflects the RTCC LOI processor as it currently exists.

Major formulation changes were made to subroutine DELV2 which computes the LOI-2 ΔV. Minor formulation changes were made to subroutines DELTAT, MINTHT, CANS, and INTER.

Ronald L. Berry

APPROVED BY:

John P. Mayer
Chief, Mission Planning and Analysis Division

The Flight Software Branch concurs with the above recommendations.

Enclosure
TO: Division Mission Planning and Analysis Branch Lunar Mission Analysis Section
FROM: Robert F. Wiley

Subject: RTCC REQUIREMENTS FOR APOLLO 14 (MISSION H-3) LOI TARGETING, REVISION 1
Date of Paper: September 25, 1970

Distribution:
- See attached memo

This is a change to distribution on Release Approval dated, 
This is an addition to distribution on Release Approval dated,

MSC Form 199 (Rev Dec 63)
PROJECT APOLLO

RTCC REQUIREMENTS FOR APOLLO 14 (MISSION H-3):
LOI TARGETING
REVISION 1

By Robert F. Wiley
Lunar Mission Analysis Branch

September 25, 1970

MISSION PLANNING AND ANALYSIS DIVISION
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS

Approved: 
Ronald L. Berry, Chief
Lunar Mission Analysis Branch

Approved: 
for John P. Mayer, Chief
Mission Planning and Analysis Division
RTCC REQUIREMENTS FOR APOLLO 14 (MISSION H-3):

LOI TARGETING

By Robert F. Wiley

1.0 SUMMARY AND INTRODUCTION

Because of experience gained from previous lunar missions, new logic for the RTCC lunar orbit insertion targeting processor was specified for Apollo 14 (Mission H-3). Several things which were being done preflight or in real time through use of graphs needed to be automated. Some of this occurred because the emphasis had been changed from automatic computation of solutions for a range of azimuths at the landing site to computation of solutions to achieve a specified orbit at some point past LOI.

This new logic was specified in reference 1. Because of RTCC program priorities, this logic was slipped to Apollo 15 (Mission H-4), and an update to the existing RTCC LOI targeting processor was made for Apollo 14 (Mission H-3). Reference 2 was the updated version of reference 1 for Apollo 15. However, because the Apollo 14 launch has been slipped, this Apollo 15 logic has been incorporated in the RTCC for Apollo 14. This internal note updates reference 2 to correct the errors detected in programing the inhouse and RTCC versions of the program and to improve some formulation.

2.0 GENERAL DISCUSSION

The LOI targeting logic for H-2 and previous missions computed several types of maneuvers (which are referred to as solutions to the LOI problem or just as solutions in this internal note) corresponding to a range of azimuths at the landing site. That logic also propagated in only a circular orbit, propagated with the AEG, and depended on some preflight data for certain calculations. This pre-H-3 LOI targeting logic rotated for the LOI ellipse in only one direction which can sometimes make the DOI maneuver farther from nominal than is necessary. However, both lunar orbit rotations are provided for in the program logic specified in this internal note; also, propagation in a high ellipse (e.g., the one that would be used for a DPS LOI) is considered; the integrator is used to predict the lunar orbit; preflight data are eliminated; and a solution to intersect with another specified lunar orbit is computed.
Some capabilities have been added to the program but some also have been given up. No fuel reserves after TEI are calculated for two reasons. First, the lunar orbit profiles are very uncertain; thus, much preflight work or much programming would have to be done to retain this function. Second, there are ways in the RTCC to estimate the reserves associated with various LOI solutions to sufficient accuracy if the nominal is known and if there is little or no landing site azimuth change. The capability to compute various solutions within a specified azimuth range over the site also has been given up because the work to program this capability cannot be justified for three reasons. First, the desire to return to the nominal is very strong. Second, an azimuth change very seldom would be required. Third, if it is ever necessary that the azimuth be changed, the flight controller should be able to iterate on azimuth and easily find an acceptable solution.

The same philosophy that was used for the present program is used in this new program (ref. 3). This philosophy consists of three things. First, the functions now known are programmed, but enough information is computed and displayed and enough degrees of freedom to the problem are specified as input to permit the accomplishment of other functions, even though it may require much real-time effort. Second, the problem of the sensitive node between the hyperbola and the lunar orbit caused by very small plane differences is solved in the same manner. Third, because it is impossible to obtain all of the desired end conditions of the LOI burn in some cases (e.g., if the nodal altitude is less than the desired lunar orbit perilune altitude, the desired lunar orbit peak altitude cannot be obtained), certain end conditions are relaxed one at a time to give various solutions.

3.0 LOI SOLUTIONS

Some terminology must be defined. A positive solution is one whose perilune is rotated ahead (i.e., in the direction of motion); a negative solution is one whose perilune is rotated behind (i.e., opposite to the direction of motion). Eight solutions are computed. There are four solution types, each with a positive-negative solution. The four types are as follows.

a. Plane solutions: obtain the desired azimuth at the landing site, giving up the lunar orbit perilune altitude if necessary.

b. Coplanar solutions: obtain the desired lunar orbit shape in the plane of the approach hyperbola with a pre-hyperbolic perilune impulsive point for the positive solution and a post-hyperbolic perilune impulsive point for the negative solution.

c. Minimum $\delta$ solutions: obtain the desired lunar orbit shape and
minimize the wedge angle between the actual and desired lunar orbit within an input maximum allowable ΔV.

d. Intersection solutions: adjust the first lunar orbit perilune altitude to obtain a specified altitude difference between it and the altitude on the post-DOI lunar orbit.

Note that this logic is like the one-azimuth logic of the pre-H-3 programs with intersection and positive-negative solutions added and with a more sophisticated method of lunar orbit propagation.

4.0 DISCUSSION OF THE TARGETING LOGIC

The LOI-1 maneuver is targeted based on the following assumed trajectory profile to the landing site. All plane change is accomplished with the first burn. A second burn adjusts the inplane orbital elements so that a specified orbit occurs at the landing site. It is not always possible to meet all desired end conditions; thus, the various solutions mentioned previously are computed.

The time at the landing site is estimated from the two parking orbits (LPO-1 and LPO-2) and the number of revolutions in each, all of which are MED inputs. Because perturbations are dependent on the orientation of the orbit relative to the moon, a true anomaly at which to transfer from the hyperbola to the first LPO is entered. (This input affects only the estimated time at the landing site and the orientation of LPO-1 when it is moved back to LOI.) Two-body equations are used; therefore, the error in time estimate will be caused by the perturbations of the lunar orbit perilune position in inertial space.

The backward integration from the landing site is done with the integrator to obtain the inplane elements of LPO-2 used to compute intersection solutions and the plane used to determine the node between the hyperbola and LPO-1. This integration is done twice. The first time, an estimate of the error in the time at the landing site is made and the time at the landing site is corrected. The second backward integration starts at this corrected time.

A maximum allowable LOI ΔV for both positive and negative solutions is entered (ΔV_{max+} and ΔV_{max-}) to calculate minimum θ solutions. Unlike the old program, solutions will be computed here even if ΔV is less than ΔV_{max}. 
The logic is divided into two major parts after the orbit has been backed up from the landing site and the coplanar solutions have been calculated. On one path, the LPO-1 plane and hyperbola plane are within some specified small angle of each other; on the other path, they are farther apart.

To compute solutions for small plane changes, the capability is provided to compute intersection solutions in the minimum θ solutions' planes. This capability is used when the planes of the minimum θ solutions are so close to the desired lunar orbit plane that the difference is negligible.

For the intersection solutions, the amount that perilune of the first lunar orbit should be above or below the second lunar orbit at perilune of the first is specified by a MED. The angle between the node and perilune of the first lunar orbit is minimized. If the perilune altitude is less than the altitude on the second LPO minus the Δh_BIAS when the lunar orbit perilune altitude is equal to the nodal altitude, perilune is placed at the node and no further computations are done.

The computation of the intersection solutions requires solving a fourth-order equation for the LPO-1 perilune radius. Because numerical problems were encountered in solving for the radius, a change was necessary to the routine that solves up to fourth-order equations, PCAQRE. That change was to remove the epsilon test on the resolvent cubic equation, as is discussed in reference 4. The RTCC LOI program has its own version of PCAQRE with this change.

The backward integration of LPO-1 from the landing site provides an estimate of the amount that the apolune and perilune altitudes and perilune position will change. These differences are used in computing intersection solutions so that Δh_BIAS is achieved at the time of the transfer from LPO-1 to LPO-2.

The ΔV for the maneuver after LOI-1 is computed for each solution. If there is no intersection or LPO-1 is the same as LPO-2, no ΔV will be computed. The nodal altitude and true anomaly for the Ψ_MX and Ψ_MN lunar orbits are computed to provide information about which way to change the landing site azimuth to improve the solution and to give an estimate of the size of the change.

The flow charts at the end of this internal note give the detailed computations discussed above. There is a functional flow chart (flow chart 1) followed by detailed flow charts (flow charts 2 through 12) which give the equations and logic routes.
5.0 MED INPUTS

LOI targeting will require an initialization MED to enter the quantities that normally would not change and a computational MED to tell the program to execute. All data the program needs to run are specified by MED for flexibility. The initialization MED should have the following information.

a. The lunar orbit parameters at the landing site (except for azimuth): $\phi$, $\lambda$, $R$ of surface; $h_a$, $h_p$, angle of perilune from the site, $\Delta\omega$ ($\Delta\omega$ is negative if the site is post-perilune.)

b. Number of revolutions in the first (revs 1) and second lunar orbits (revs 2). (Revs 1 may have a fractional part.)

c. True anomaly on LPO-1 for transferring from the hyperbola to LPO-1, $\eta_1$

d. Altitude constraint of the intersection solutions, $\Delta h_{BIAS}$. The $\Delta h_{BIAS}$ is negative if LPO-2 is to be below the LPO-1 perilune. The $\Delta h_{BIAS}$ and $\eta_1$ should revert to zero.

e. A flag to specify if plane or minimum 6 nodes should be used to compute intersection solutions. This should revert to plane solutions.

The computational MED should have the following information.

a. Apolune and perilune altitudes of the first LPO

b. A maximum allowable $\Delta V$ for the positive solutions and a maximum allowable $\Delta V$ for the negative solutions

c. The maximum, desired, and minimum azimuths at the landing site

d. From where and at what time to obtain the pre-LOI state vector

6.0 FLOW CHART NOTATION

a

semimajor axis (except when used in INTER and DELV2); value redefined at several places in the program

$a_h$

semimajor axis of the hyperbola at perilune
e

eccentricity (except when used in INTER and DELV2); values redefined at several places in the program

e^H

eccentricity of the hyperbola at perilune

H_ALPO, H_PLPO

input apolune and perilune altitudes of the first LPO

LLS

lunar landing site

LPO

lunar parking orbit

R, V, R, R, V

state vector positions and velocities; values redefined at several places in the program

REVS 1, REVS 2

revs in the first and second LPO's

R_ALPO, R_PLPO

apolune and perilune radii of the first LPO

R_N, R_NODE, R_N, R_NODE

nodal radius magnitude and unit vector to the node

SOLN

denotes a positive solution (value is +1) or a negative solution (value is -1)

S_PH

hyperbolic perilune state vector

U_H

unit angular momentum vector to S_PH

U_L

unit inertial vector to LPO-2 perilune used in INTER. Equal to \( \hat{U}_{PL} \) or the projection of \( \hat{U}_{PL} \) into the minimum \( \psi \) solutions' planes

U_LLS

unit inertial position vector to the LLS at the estimated time at the LLS

U_MX, U_DS, U_MN

unit angular momentum vectors to the \( \psi_{MX}, \psi_{DS}, \psi_{MN} \) LPO's, respectively

U_N

unit \( (\hat{U}_{DS} \times \hat{U}_{XTRA}) \)

U_PC

unit position vector to hyperbolic perilune

U_PL

unit inertial vector to LPO-2 perilune at LOI-2 time

U_S

unit angular momentum vector to the plane of the solution
unit angular momentum vector to an LPO at the LLS made from $\psi_{DS} - 5^\circ$ and backed up to LOI

estimate of the change of apolune altitude, perilune altitude, and perilune position due to propagation in LPO-1

difference between the perilune of the first LPO and the corresponding position on the second LPO

time error in the estimated time at the LLS

maximum allowable LOI $\Delta V$ for the positive solution

maximum allowable LOI $\Delta V$ for the negative solution

$\Delta V$'s for the positive and negative selections

maximum, desired, and minimum azimuths at the LLS

ture anomaly at which to get on LPO-1. Used only for the $T_{LLS}$ estimate and in backing up the lunar orbit to LOI

unit vector; value redefined at several places in the program

multiply $y$ by $\hat{1}$, where the sign of the $\hat{1}$ is the sign of $x$

denotes the positive solution

denotes the negative solution

denotes a unit vector
PRELIMINARY CALCULATIONS AND ERROR CHECKS

ESTIMATE TIME AT THE LLS

PROPAGATE THE LPO FROM LLS TO LOI

PROPAGATE THE SAME LPO BUT WITH A DIFFERENT $\psi_{lls}$ FROM LLS TO LOI

COMMENT: TO COMPUTE SOME DISPLAY INFORMATION

COMPUTE $\theta_{mx}$ AND $\theta_{mn}$

COMPUTE $\pm$ COPLANAR SOLNS

COMPUTE NODE < 90° FROM HYPERBOLA PERILUNE

$R_{node} > R_{alpo}$

NO

COMPUTE $\pm$ PLANE SOLNS AT NODE

COMMENT: $h_p = h_n$ IF $h_n < h_{LPO}$

YES

LPO AND HYPERBOLA COPLANAR

STORE $\pm$ COPLANAR SOLNS AT $\pm$ PLANE SOLNS

COMMENT: NO $\theta$ SOLNS ARE NECESSARY

COMPUTE $\pm$ MIN $\theta$ SOLNS

COMMENT: CAN NOT BURN TO AN LPO WITH $R_{alpo}$ APOLUNE RADIUS AND THE DESIRED PLANE

COMPUTE $\pm$ INTERSECTION SOLNS

USING $\pm$ PLANE SOLNS

Flow chart 1.- Functional flow.
MUST MIN \( \theta \) SOLNS BE COMPUTED

**NO**

**YES**

COMMENT: ONE OR BOTH PLANE SOLNS HAS EXCEEDED \( \Delta V_{\text{MAX}} \) OR \( h_n \) WAS LESS THAN \( h_{\text{PLPO}} \)

**COMPUTE NECESSARY MIN \( \theta \) SOLNS**

**PLANE**

**OR MIN \( \theta \) SOLNS TO BE USED FOR INTERSECTION SOLNS**

**A**

**MIN \( \theta \)**

COMMENT: THIS PATH SHOULD BE USED TO KEEP INTERSECTION SOLNS ALMOST THE SAME FOR SUCCESSIVE TRAJECTORY UPDATES WITH SMALL LOI PLANE CHANGES

**COMPUTE \( \pm \) INTERSECTION SOLNS USING \( \pm \) MIN \( \theta \) SOLNS**

COMMENT: IF A \( + \) OR \( - \) MIN \( \theta \) SOLN IS NOT AVAILABLE, THE CORRESPONDING \( + \) OR \( - \) INTERSECTION SOLN WILL NOT BE COMPUTED

**C**

**COMPUTE ADDITIONAL DISPLAY INFORMATION**

**COMPUTE ELEMENTS BEFORE AND AFTER LOI FOR TRANSFER TO THE MPT**

**EXIT AND DISPLAY**

Flow chart 1. – Functional flow – Concluded.
ENTER WITH $H_{\text{ALPO}}', H_{\text{PLPO}}', \psi_{\text{MX}}, \psi_{\text{DS}}$, $\psi_{\text{MN}}, \Delta V_{\text{MAX}}', \Delta V_{\text{MAX}}$, LPO-2 AT LLS, $R_{\text{LLS}}$, REV 1, REV 2, $\Delta h_{\text{BIAS}}'\gamma_{\text{L}}$, SPH

$R_{\text{ALPO}} = H_{\text{ALPO}} + R_{\text{LLS}}$
$R_{\text{PLPO}} = H_{\text{PLPO}} + R_{\text{LLS}}$
$R_{\text{PHYP}} = |R_{\text{PHYP}}|$
$\dot{u}_H = \text{UNIT}(R \times \vec{v})_{\text{SPH}}$
$\dot{u}_\text{PC} = \text{UNIT}(R_{\text{SPH}})$

$R_{\text{LLS}} \leq R_{\text{PHYP}} \leq R_{\text{ALPO}}$ NO

$\psi_{\text{MX}} \geq \psi_{\text{DS}} \geq \psi_{\text{MN}}$ NO ERROR EXIT

$\psi_{\text{XTRA}} = \psi_{\text{DS}} - 5^\circ$

$\psi_{\text{MX}} - \psi_{\text{MN}} < 170^\circ$ NO

$R_{\text{ALPO}} \geq R_{\text{PLPO}}$ NO ERROR EXIT

$\psi_{\text{XTRA}} = \psi_{\text{DS}} - 5^\circ$

Flow chart 2.- Detailed flow.
COMPUTE LPO AT LLS:

\[ a_{LLS} = \frac{h_a + h_p}{2} + R_{LLS} \]

\[ e_{LLS} = \frac{h_a + R_{LLS}}{a_{LLS}} - 1 \]

\[ P_{LLS} = a_{LLS} \left(1 - e_{LLS}^2\right) \]

\[ R = P \left(1 + e_{LLS} \cos \Delta \omega\right) \]

\[ V = \left[\frac{\mu}{R} - \frac{1}{a_{LLS}}\right]^{1/2} \]

\[ \cos \gamma = \sqrt{\mu a_{LLS} \left(1 - e_{LLS}^2\right)/RV} \]

CHECK \(|\cos \gamma|\) FOR > 1 AND RESET IF NECESSARY

\[ \gamma = -\text{SIGN (}\Delta \omega\text{)} \cos^{-1} (\cos \gamma) \]

CALL TIME FOR \(T_{LLS}\)

Flow chart 2 - Detailed flow - Continued.
Flow chart 2.- Detailed flow - Continued.
CALL COPLNR FOR ± COPLANAR SOLNS

\( \hat{\theta} = \text{UNIT} (\hat{0}_N \times \hat{0}_{DS}) \)

\[
\begin{align*}
\hat{0}_{MN} &= \cos(\psi_{DS} - \psi_{MN}) \hat{0}_{DS} + \sin(\psi_{DS} - \psi_{MN}) \hat{\theta} \\
\hat{0}_{MX} &= \cos(\psi_{MX} - \psi_{DS}) \hat{0}_{DS} - \sin(\psi_{MX} - \psi_{DS}) \hat{\theta}
\end{align*}
\]

\( \hat{0} = \hat{0}_{DS} \)

\( \theta_{\text{COP}} = \cos^{-1} (\hat{0} \cdot \hat{0}_H) \)

Flow chart 2. - Detailed flow - Continued.
Flow chart 2. - Detailed flow - Continued.
Flow chart 2.- Detailed flow - Continued.
Flow chart 2.- Detailed flow - Continued.
WHICH NODES FOR INTERSECTION SOLNS

CALL INTER WITH + PLANE SOLN NODE STATE AND \( \Phi_{DS} \) LPO AT LOI-2 TIME (FROM BACKUP) FOR + INTERSECTION SOLN

CALL INTER WITH - PLANE SOLN NODE STATE AND \( \Phi_{DS} \) LPO AT LOI-2 TIME FOR - INTERSECTION SOLN

DISPLAY 10

Flow chart 2. – Detailed flow – Continued.
COMPUTE DISPLAY PARAMETERS AT THE NODE FOR THE \( \pm \) PLANE SOLNS: \( r_N \) AND \( h_N \)

CALL MINTHT FOR THE \( \pm \) MIN \( \theta \) SOLNS

\( \mathbf{\dot{0}}_{S+} = \) UNIT NORMAL TO + MIN \( \theta \) SOLN PLANE
\( \mathbf{\dot{0}}_{S-} = \) UNIT NORMAL TO - MIN \( \theta \) SOLN PLANE

\( \mathbf{\dot{0}}_{L} = \) UNIT vector to perihelion of LPO, backed up from LLS, at LOI-2 time projected into \( \mathbf{\dot{0}}_{S+} \) plane
\( a, e \) TO BE USED IN INTER ARE THOSE COMPUTED FROM THE \( \psi_{DS} \) CALL TO BACKUP

CALL INTER WITH + MIN \( \theta \) SOLN NODE STATE AND \( a, e \) AND \( \mathbf{\dot{0}}_{L} \) FOR + INTERSECTION SOLN

NO

YES

CALL INTER WITH - MIN \( \theta \) SOLN NODE STATE AND \( a, e \) AND \( \mathbf{\dot{0}}_{L} \) FOR - INTERSECTION SOLN

Flow chart 2.- Detailed flow - Continued.
\[ h_{MX} = \frac{a_H (1 - e_H^2)}{1 + e_H (v_{MX} \cdot u_{PC})} - R_{LLS} \]
\[ h_{MN} = \frac{a_H (1 - e_H^2)}{1 + e_H (v_{MN} \cdot o_{PC})} - R_{LLS} \]

**COMPUTE** \( R_p \) **TO GIVE** \( \eta_1 \) **AT THE NODE FOR EACH INTERSECTION SOLN** (FOR POSSIBLE DISPLAY)

\[
e = \frac{(R_{ALPO} - R_{NODE})}{R_{NODE} \cos \eta_1 + R_{ALPO}}
\]

\[
R_p = \frac{2R_{ALPO}}{e + 1} - R_{ALPO}
\]

**CALL DELV2 FOR EACH SOLN TO COMPUTE THE LOI-2 \( \Delta V \) FOR DISPLAY**

Flow chart 2. - Detailed flow - Continued.
SELECT FIRST SOLN FOR COMPUTATION OF STATE AFTER LOI

INTEGRATE ON HYPERBOLA FROM SPH IN THE DIRECTION OF
SIGN \( \hat{\left( 0_{PC} \times \hat{R} \right)} \) NODE OF THE SOLN \( \hat{R}_H \) TO

\[ |R| = |\hat{R}_N| \] NODE OF THE SOLN \( R \), RETURN \( \vec{R}_H \), \( \vec{V}_H \), \( T_H \), \( \gamma_H \), \( \Psi_H \)

COMPUTE STATE AFTER LOI

\[ \hat{R}_A = \hat{R}_H \]
\[ |\vec{V}_A| = |\vec{V}_H| + \Delta V \]
\[ \Psi_A = \Psi_H + \Delta \Psi \]
\[ \gamma_A = \gamma_H + \Delta \gamma \]
\[ T_A = T_H \]

CONVERT \( |\vec{V}_A| \), \( \Psi_A \), \( \gamma_A \) TO \( \dot{x}, \dot{v}, \dot{z} \)

STORE FOR TRANSFER TO MPT:
STATE BEFORE LOI:
\[ x_H \quad \dot{x}_H \]
\[ y_H \quad \dot{y}_H \]
\[ z_H \quad \dot{z}_H \]
STATE AFTER LOI:
\[ x_A \quad \dot{x}_A \]
\[ y_A \quad \dot{y}_A \]
\[ z_A \quad \dot{z}_A \]
TIME OF MANEUVER
\[ T_H \]

FLOW CHART 2 - DETAIL FLOW - CONCLUDED.
START

R1 = INTEGER PART OF REVS 1
DR1 = (FRACTIONAL PART OF REVS 1)(2π)
R2 = REVS 2

\[ a_1 = \left( \frac{R_{ALPO} + R_{PLPO}}{2} \right) \]
\[ e_1 = \frac{R_{ALPO}}{a_1} - 1 \]

CALL DELTAT WITH \( a_1, e_1 \), \( \eta_1 \), AND DR1 FOR \( \Delta t_2 \)

\[ \eta_2 = \eta_1 + DR1 \]

\[ \Delta t_1 = 2\pi R1 \frac{a_1^3}{\mu} + 2\pi R2 \frac{a_{LLS}^3}{\mu} \]

\[ T_{LLS} = T_{PCYN} + \Delta t_1 + \Delta t_2 \]

MAKE A UNIT INERTIAL POSITION VECTOR \( \hat{\phi}_{LLS} \) FROM \( \phi_{LLS}, \lambda_{LLS} \), AND \( T_{LLS} \)

\[ \Delta \phi = \cos^{-1} \left( \hat{\phi}_{LLS} \cdot \hat{\phi}_{PC} \right) \]

PURPOSE: TO ESTIMATE THE TIME AT THE LLS.
COMMENT: CONIC ESTIMATE.

Flow chart 3, Subroutine TIME.
SAVE \( \eta_2 \) AND \( \Delta t_3 \) AND \( \Delta a \) AND \( DA \) FOR USE IN BACKUP

\[
U_{LLS} \times U_{PC}, U_H > 0 \quad \text{YES}
\]

\[
\Delta a = 2\pi - \Delta a \quad \text{NO}
\]

\[
\Delta a > DR1 \quad \text{YES}
\]

\[
DA = 2\pi - (DR1 - \Delta a) \quad \text{NO}
\]

\[
DA = \Delta a - DR1 \quad \text{YES}
\]

CALL DELTAT WITH \( a_{LLS}, e_{LLS}, \Delta A, -\Delta \omega \) IF \( \Delta \omega < 0 \) AND \( (2\pi - \Delta \omega) \) IF \( \Delta \omega > 0 \)
FOR \( \Delta t_3 \)

\[
T_{LLS} = T_{PCYN} + \Delta t_1 + \Delta t_2 - \Delta t_3
\]

SECOND PASS

NO

YES

\[
A = 1
\]

RETURN

Flow chart 3. - Subroutine TIME - Concluded.
PURPOSE: TO BACKUP AN INERTIAL LPO AT THE LLS THROUGH REVS 1 AND REVS 2 TO LOI, PROPAGATING IN AN APPROXIMATION TO THE INITIAL LPO AND USING THE INTEGRATOR.

Flow chart 4.- Subroutine BACKUP.
CALL DELTAT WITH $\eta, a, e, \Delta \eta$, FOR $\Delta t$

INTEGRATE FROM PRESENT STATE TO $T + \Delta t = \vec{R}_2, \vec{V}_2, \tau_2$

COMPUTE $a, e = a_L, e_L$, AND CONIC UNIT SELENOCENTRIC VECTOR TO PERILUNE $\vec{U}_{PL}$ AND SAVE FOR USE IN COMPUTING INTERSECTION SOLUTIONS

COMPUTE $R_P$ OF DOI ORBIT

IF $R_P < R_{LLS}$ WRITE ON-LINE MESSAGE AND $R_P$

INTEGRATE BACK ONE PERIOD FROM THIS POINT TO COMPUTE AND LPO FOR DISPLAY OR ON-LINE PRINT

COMPUTE FIRST LPO

PLANE = PLANE OF $\vec{R}_2$ AND $\vec{V}_2$

$a = \frac{R_{ALPO} + R_{PLPO}}{2}$

$e = \frac{R_{ALPO}}{a} - 1$

$P = a \left( 1 - e^2 \right)$

$R = P / \left( 1 + e \cos \eta_2 \right)$

$V = \left[ \mu \left( \frac{2}{R} - \frac{1}{a} \right) \right]^{1/2}$

$\cos \gamma = \sqrt{\frac{\mu a (1 - e^2)}{RV}}$

$T = \tau_2$

Flow chart 4. - Subroutine BACKUP - Continued.
Flow chart 4. - Subroutine BACKUP - Concluded.
START

a, e, η, Δη SENT

CONVERT η TO 0° TO 360°

\[ K_e = \sqrt{\frac{1-e}{1+e}} \]

\[ E = \tan^{-1} \left( \frac{\sqrt{1-e^2} \sin \eta}{\cos \eta + e} \right) \]

SET E 0° TO 360°

\[ η' = η + Δη \]

\( 8E = 0 \)

YES

\[ 0 \leq η' \leq 2\pi \]

NO

\[ η' < 0 \]

YES

\[ η' = η' LESS MULTIPLES OF 2π \]

\[ 8E = -\text{(MULTIPLES OF } 2\pi + 2\pi) \text{ (NEGATIVE NUMBER)} \]

\[ a \]

FLOW CHART 5.- Subroutine DELTAT.
Flow chart 5.- Subroutine DELTAT - Concluded.
PURPOSE: TO COMPUTE A MANEUVER ΔV OR THE COSINE OF THE ALLOWABLE PLANE CHANGE WITHIN A GIVEN ΔV

Flow chart 6.- Subroutine DVDA.
\[
\cos \gamma_L = \frac{H_L}{R_N V_L}
\]

CHECK \(|\cos \gamma_L| > 1\)
AND RESET IF NECESSARY

\[
\sin \gamma_L = \pm \sqrt{1 - \cos^2 \gamma_L}
\]

\[
\Delta V = \sqrt{V_H^2 + V_L^2 - 2V_H V_L \left(\cos |\gamma_H| \cos \gamma_L \cos \Delta \phi_N + \text{sgn} \sin \gamma_H \sin \gamma_L\right)}
\]

\[
\cos \Delta \phi_N = \frac{\left[\frac{V_H^2 + V_L^2 - \Delta V^2}{2V_L V_H} - \text{sgn} \sin \gamma_H \sin \gamma_L\right]}{\cos \gamma_L \cos \gamma_H}
\]

RETURN WITH \(\Delta V\)
RETURN WITH \(\cos \Delta \phi_N\)

Flow chart 6.- Subroutine DVDA - Concluded.
START

\[ V_H = |\vec{V}| \text{ ON HYPERBOLA AT IMPULSE POINT} \]
\[ \gamma_H = \gamma \text{ ON HYPERBOLA AT IMPULSE POINT} \]
\[ \psi_H = \psi \text{ ON HYPERBOLA AT IMPULSE POINT} \]
\[ R_N = |\vec{R}| \text{ ON HYPERBOLA AT IMPULSE POINT} \]

\[ V_{LPO} = \left[ \frac{2}{R_N} \left( \frac{2}{R_{ALPO} + R_P} \right) \right]^{1/2} \]

\[ \cos \gamma_{LPO} = \frac{\sqrt{\mu \left( \frac{R_{ALPO} + R_P}{2R_{ALPO} + 2R_P} \right)^2 - 1}}{2 \frac{R_N V_{LPO}}{R_{LPO}}} \]

CHECK \( \cos \gamma_L \) FOR >1 AND RESET IF NECESSARY

COMPUTE AZIMUTH ON LPO FROM \( \hat{u}_S \)

\[ \Delta V = V_{LPO} - V_H \]
\[ \Delta \psi = \psi_{LPO} - \psi_H \]
\[ \Delta \gamma = -(\text{SIGN OF SOLN}) \gamma_{LPO} - \gamma_H \]

PURPOSE: TO COMPUTE \( \Delta |V|, \Delta \gamma, \Delta \psi \) OF THE SOLUTION AND CERTAIN DISPLAY QUANTITIES

Flow chart 7. - Subroutine CANS.
COMPUTE DISPLAY QUANTITIES:

\[ h_N = R_N - R_{LLS} \]

\[ \eta_N = \text{SIGN} \left[ (\vec{0}_{PC} \times \vec{R}) \cdot \vec{0}_H \right] \cos^{-1} \left( \vec{0}_{PC} \cdot \vec{R} \right) \]

\[ \phi_{OP} = \left| \cos^{-1} \left( \vec{0}_S \cdot \vec{0}_N \right) - \pi/2 \right| \]

\[ T_N = \text{TIME OF VECTOR AT THE NODE} \]

\[ a = \left( \frac{R_{ALPO} + R_P}{2} \right) \]

\[ e = \frac{R_{ALPO}}{a} - 1 \]

\[ \begin{cases} \text{NO} & \left[ R_{ALPO} - R_P \right] > .01 \text{ N. MI.} \\ \text{YES} & \left[ R_{ALPO} - R_P \right] \leq .01 \text{ N. MI.} \end{cases} \]

\[ \cos W_p = 1 \]

\[ \cos W_p = \left[ \frac{a(1 - e^2)}{R_N} - 1 \right] / e \]

CHECK [\cos W_p] > 1 AND

RESET IF NECESSARY

\[ W_p = - (\text{SIGN OF SOLN}) \cos^{-1} \left( \cos W_p \right) \]

STORE: STATE AT NODE, \( \Delta v, \Delta y, \Delta \phi, h_N, \eta_N, \phi_{OP}, \theta, h_p, W_p \)

SAVE \( \vec{0}_S \) FOR USE IN COMPUTING INTERSECTION SOLNS AND IN DELV2

RETURN
AVoops. = 10 000 FPS

\[ R_{\text{PHY}} \geq R_{\text{PLPO}} \]

\[ R = R_{\text{PLPO}} - 0.01 \text{ N. MI.} \]

\[ R = R_{\text{PHY}} - 0.01 \text{ N. MI.} \]

\[ \theta_{\text{COP}} = \cos^{-1}(\hat{u}_S \cdot \hat{u}_H) \]

\[ \hat{u}_S = \hat{u}_H \]

\[ \cos \Delta \varphi_N = 1.0 \]

\[ R = R + 0.01 \text{ N. MI.} \]

\[ R > R_{\text{ALPO}} \]

CALL DVDA FOR \( \Delta V_+ \) OF + COPLANAR SOLN
SGN IN DVDA IS +

\[ \Delta V_{\text{COP}+} = \Delta V_+ \]

\[ \Delta V_+ < \Delta V_{\text{COP}+} \]

CALL CANS WITH PREHYPERBOLIC PERILUNE
STATE AT R - 0.01 N. MI. TO STORE + SOLN

CALL CANS WITH POSTHYPERBOLIC PERILUNE STATE
AT R - 0.01 N. MI. TO STORE - SOLN
\( \Delta V_{\text{COP}} = \Delta V_{\text{COP}+} \)

PURPOSE: TO COMPUTE \( \pm \) COPLANAR SOLNS

Flow chart B. - Subroutine COPLNR.
START

INITIALIZE ± MIN θ SOLNS WITH ± COPLANAR SOLNS

PROPAGATE ON THE HYPERBOLA CONIC TO PREPERILUNE R = R_ALPO

FIRST PASS

YES

R < R_PLPO

YES

PROPAGATE TO R = R_PLPO' POST-PCYN

NO

SET INDICATOR FOR + SOLN

CALL DVDA WITH ΔV_MAX OF SOLN BEING COMPUTED, R_ALPO' AND R_PLPO FOR COS Δψ_Na

Θ_P = UNIT[Θ_DS - (Θ_DS · ˆ R)]

COS ω = Θ_P · ˆ H

COS Δψ_Na > 1

YES

NO

PURPOSE: TO COMPUTE + AND - MINIMUM θ SOLNS

Flow chart 9. - Subroutine MINTHT.
Flow chart 9. - Subroutine MINTHT - Concluded.
\[
\Delta W_a = \cos^{-1}(\hat{r}_N \cdot \hat{u}_L)
\]
\[
R_P = R_N
\]
\[
DW = \Delta W_a + \text{SIGN} \left[ (\hat{u}_L \times \hat{r}_N) \cdot \hat{u}_S \right] \Delta W_P
\]
\[
\Delta h = \frac{a_L (1 - e_L^2)}{1 + e_L \cos \omega} - R_P - \Delta h_P
\]

**PURPOSE:** TO COMPUTE + OR - INTERSECTION SOLUTIONS

Flow chart 10. - Subroutine INTER.
COMPUTE EQUATION COEFFICIENTS

\[
\begin{align*}
    a &= R_N^2 \left(1 + ECA^2\right) + e_L^2 \left(K_1^2 - R_N^2\right) - 2K_1 R_N ECA \\
    b &= 2R_N^2 \left[ECA^2 \left(\Delta h - R_a\right) - R_a - c_1\right] + 2e_L^2 \left[K_1^2 \Delta h - K_1 R_N R_a\right] \\
    & \quad - R_N^2 \left(\Delta h - R_a\right) + 2ECA R_N \left[K_1 (R_a - \Delta h + c_1) + R_a R_N\right] \\
    c &= R_N^2 \left[R_a^2 + c_1^2 + 4c_1 R_a + ECA^2 \left(R_a^2 + \Delta h^2 - 4R_a \Delta h\right)\right] \\
    & \quad + e_L^2 \left(\Delta h^2 K_1 - 4K_1 R_a R_N \Delta h + R_N^2 \left(4R_a \Delta h - \Delta h^2\right)\right) \\
    & \quad - 2ECA \left[R_a R_N \left(K_1 c_1 - K_1 \Delta h - R_N \Delta h + R_N R_a\right) + R_N^2 c_1 R_a - c_1 \Delta h R_N K_1\right] \\
    d &= 2e_L^2 \Delta h^2 R_a R_N \left(R_N - K_1\right) - 2R_N^2 R_a \left[c_1^2 + ECA^2 \left(\Delta h^2 - R_a \Delta h\right) + c_1 R_a\right] \\
    & \quad + 2ECA \left[R_N^2 R_a^2 \left(c_1 - \Delta h\right) - c_1 \Delta h R_a R_N \left(K_1 + R_N\right)\right] \\
    e &= R_N^2 R_a^2 \left[c_1^2 + 2ECA c_1 \Delta h + ECA^2 \Delta h^2\right]
\end{align*}
\]

SOLVE \( aR_p^4 + bR_p^3 + cR_p^2 + dR_p + e = 0 \)
FOR \( R_p \) WITH PCAQRE

REAL SOLNS

\( \text{NO} \) \( \rightarrow \) ERROR

\( \text{YES} \)

SELECT LARGEST REAL \( R_p \)

CALL DELTAH FOR \( \Delta h \)

Flow chart 10. - Subroutine INTER - Continued.
Flow chart 10,- Subroutine INTER - Concluded.
START

\[
a = \frac{R_{ALPO} + R_p}{2}
\]

\[
e = \frac{R_{ALPO}}{a} - 1
\]

\[
p = a(1 - e^2)
\]

\[
\cos W_p = \frac{p - R_N}{R_N e}
\]

CHECK \( \cos W_p \) FOR \( >1 \)
AND RESET IF NECESSARY

\[
W_p = \cos^{-1}(\cos W_p)
\]

\[
DW = \Delta w + \left[ \text{SIGN} \left( (\hat{\theta}_L \times \hat{\theta}_N) \cdot \hat{\theta}_S \right) \right] \left[ \Delta w_p + \text{SIGN OF SOLN} \right] W_p
\]

\[
R_{NL} = \frac{a_L(1 - e_L^2)}{1 + e_L \cos(DW)}
\]

\[
\Delta h = R_{NL} - R_p - \Delta h_p
\]

RETURN

PURPOSE: TO COMPUTE THE \( \Delta h \) BETWEEN \( R_p \) OF THE FIRST LPO
AT DOI TIME AND THE \( R \) ON THE SECOND LPO

Flow chart 11. - Subroutine DELTAH.
START

\[ \hat{S} = \text{UNIT} \left( \mathbf{R} \times \mathbf{V} \right) \text{OF STATE AFTER LOI} \]
\[ R = \text{APOLUNE RADIUS OF STATE AFTER LOI} + \Delta a \]
\[ \hat{R} = \text{PERILUNE RADIUS OF STATE AFTER LOI} + \Delta p \]

\[ W_p = W_p \text{ COMPUTED IN CANS} \]
\[ \hat{\mathbf{p}}' = \cos W_p \hat{R}_{\text{NODE}} - \sin W_p \text{UNIT} \left( \hat{S} \times \hat{R}_{\text{NODE}} \right) \]

\[ \hat{\mathbf{p}} = \cos \Delta W_p \hat{\mathbf{p}}' - \sin \Delta W_p \left[ \text{UNIT} \left( \hat{R}_{\text{PL}} \times \hat{S} \right) \right] \]
\[ a = \left( \frac{R_a + R_p}{2} \right) \]
\[ e = \frac{a - 1}{a} \]

\[ e = e_L \quad \text{YES} \quad \quad e > e_L \quad \text{NO} \]

\[ \hat{\mathbf{p}}_1 = \hat{\mathbf{p}} \]
\[ \hat{\mathbf{p}}_2 = \hat{\mathbf{p}}_{\text{PL}} \]

\[ P_2 = a_2 \left( 1 - e_2^2 \right) \]
\[ P_1 = a_1 \left( 1 - e_1^2 \right) \]
\[ \cos (\Delta W) = \hat{\mathbf{p}}_1 \cdot \hat{\mathbf{p}}_2 \]

CHECK \[ \cos \Delta W \] FOR \( \leq 1 \) AND RESET IF NECESSARY

\[ \sin \Delta W = \sqrt{1 - \cos^2 \Delta W} \]

\[ \Delta W = \text{SIGN} \left( \left( \hat{\mathbf{p}}_1 \times \hat{\mathbf{p}}_2 \right) \cdot \hat{S} \right) \cos^{-1} \left( \cos \Delta W \right) \]

PURPOSE: TO COMPUTE THE \( \Delta V \) FOR THE MANEUVER AFTER LOI

Flow chart 12.- Subroutine DELV2.
\[ K_p = P_1 - P_2 \]
\[ K_c = P_1 e_2 - (\cos DW) P_2 e_1 \]
\[ K_s = P_2 e_1 \sin DW \]

\[ a = K_c^2 + K_s^2 \]
\[ b = 2K_pK_c \]
\[ c = K_p^2 - K_s^2 \]

\[ \text{DISC} = b^2 - 4ac \]

- DISC < 0
  - \( SGN_1 = +1 \)
  - \( \cos \eta_2 = \frac{-b + SGN_1 \sqrt{\text{DISC}}}{2a} \)
  - CHECK \(|\cos \eta_2|\) FOR > 0 AND RESET IF NECESSARY
  - \( \eta_2 = \cos^{-1}(\cos \eta_2) \)

- DISC = 0

Flow chart 12. - Subroutine DELV2 - Continued.
Flow chart 12 - Subroutine DELV2 - Continued.

$\eta_1 = DW + \eta_2$

$R_1 = \frac{p_1}{(1 + e_1 \cos \eta_1)}$

$R_2 = \frac{p_2}{(1 + e_2 \cos \eta_2)}$

$|R_1 - R_2| \leq \epsilon_2$

$|\eta_1| \geq \pi$

$\eta_1 = \eta_1 - \text{SIGN}(\eta_1) \times 2\pi$

$\eta_2 = -\eta_2$

$\Delta V = 0$

Page 3 of 4
\[ V_1 = \left[ \frac{2}{R_1 - \frac{1}{a_1}} \right]^{1/2} \]
\[ V_2 = \left[ \frac{2}{R_2 - \frac{1}{a_2}} \right]^{1/2} \]
\[ \cos \gamma_1 = \frac{\mu P_1}{R_1 V_1} \]
\[ \cos \gamma_2 = \frac{\mu P_2}{R_2 V_2} \]

CHECK |\cos \gamma| FOR > 1 AND RESET IF NECESSARY

\[ \gamma_1 = \text{SIGN}(\eta_1) \cos^{-1}(\cos \gamma_1) \]
\[ \gamma_2 = \text{SIGN}(\eta_2) \cos^{-1}(\cos \gamma_2) \]

\[ \Delta \gamma = |\gamma_1 - \gamma_2| \]

\[ \Delta V = \sqrt{V_1^2 + V_2^2 - 2V_1 V_2 \cos \Delta \gamma} \]

SELECT LESSER OF TWO \( \Delta V \)'S FOR DISPLAY, SAVE \( \eta_1, \eta_2, \Delta \gamma, V_1, V_2, R_1, R_2 \) FOR POSSIBLE OUTPUT

RETURN

Flow chart 12 - Subroutine DELV2 - Concluded.
REFERENCES


4. Notes on the Quartic Problem Encountered While Computing the Intersection Solutions in the LOI Processor. TRW IOC 70-5521.4-59, Sept. 8, 1970.