BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W. WASHINGTON, D.C. 20024

B69 09056

SUBJECT: A Method for Determining Minimum ΔV Two Impulse Transfer Trajectories Between Arbitrary State Vectors and Its Application to Three Impulse LOI Optimization - Case 310

DATE: September 24, 1969

FROM: R. A. Bass
E. A. McGinness

ABSTRACT

The need for a method of determination of an optimum transfer trajectory between two state vectors arises frequently in multiple impulse optimization techniques. An analytical solution formulated by Mr. F. T. Sun in a recent AIAA Journal provides such a method for the two-impulse transfer that minimizes the sum of the ΔV requirements at the terminal states. Mr. Sun's technique requires significantly less computation than normal iterative routines.

The method is presently being used in a three-impulse lunar orbit insertion optimization program to determine the minimum total ΔV given the initial and final state vectors and the position vector for the second maneuver.

(NASA-CR-109073) A METHOD FOR DETERMINING MINIMUM DELTA 5 TWO IMPULSE TRANSFER TRAJECTORIES BETWEEN ARBITRARY STATE VECTORS AND ITS APPLICATION TO THREE IMPULSE LOI OPTIMIZATION (Bellcomm, Inc.) 26 p

N79-71888

Unclas
00/13 11706

(PAGES) (CODE)

CR-109073

(NASA CR OR TMX OR AD NUMBER) (CATEGORY)

FF No. 602/A

[redacted]

BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W. WASHINGTON, D.C. 20024

B69 09056

SUBJECT: A Method for Determining Minimum ΔV Two Impulse Transfer Trajectories Between Arbitrary State Vectors and Its Application to Three Impulse LOI Optimization - Case 310

DATE: September 24, 1969

FROM: R. A. Bass
E. A. McGinness

MEMORANDUM FOR FILE

Introduction

A common problem in trajectory analysis involves determination of the transfer trajectory that minimizes the total ΔV required between an initial position and velocity and a desired final position and velocity. F. T. Sun has formulated in Reference 1 an analytical solution for this optimum two impulse transfer trajectory. A FORTRAN subroutine TOHSUN has been written implementing his formulation.

Summary of the Analytical Solution

The problem is formulated in Reference 1 using a pair of oblique velocity coordinates (V_C, V_R), the components parallel to the chordal and local radial directions respectively (see Figure 1B). If the transfer angle (ψ) is not equal to 180 degrees all the possible transfer trajectories will lie in a unique plane defined by the two position vectors. The transfer velocities at each terminal are related as follows

$$\bar{V}_{C1} = \bar{V}_{C2} \text{ and } V_{C2}\, V_{R2} = V_{C1}\, V_{R1} = \frac{\mu}{d} \tan(\psi/2) \qquad (1)$$

where μ is the gravitational parameter and d is defined as shown in Figure 1B. These relations are derived in Appendix D.

Using these relations and the geometrical properties as shown in Figure 1, an analytical expression for the ΔV requirement at each terminal can be developed by use of Figure 1 and the cosine law.

$$\Delta V_i = \left(V_C^2 + V_R^2 - 2N_{Oi}\, V_C - 2M_{Oi}\, V_R + V_{Oi}^2 - 2\frac{\mu}{d} \tan \psi/2 \cos \phi_i\right)^{1/2} \qquad (2)$$

with

$$M_{Oi} = \bar{V}_{Oi} \cdot \bar{e}_{Ri},\ N_{Oi} = \bar{V}_{Oi} \cdot \bar{e}_{Ci} \text{ and for } i = 1, 2$$

where $\bar{e}$ represents a unit vector with the direction denoted by the subscript.

Upon summing the ΔV's, differentiating, and equating to zero, the following octic[1] equation results.

$$\sum_{n=0}^{8} C_n V_C^n = 0 \tag{3}$$

where the coefficients are:

$$C_8 = N_{01}^2 - N_{02}^2 - P_{01} + P_{02}$$

$$C_7 = 4K(M_{01} - M_{02}) + 2N_{01}N_{02}(N_{02} - N_{01}) + 2(N_{02}P_{01} - N_{01}P_{02})$$

$$C_6 = 2K(N_{02}M_{02} - N_{01}M_{01} + 4N_{01}M_{02} - 4N_{02}M_{01}) + (N_{01}^2P_{02} - N_{02}^2P_{01})$$

$$C_5 = 4K^2(N_{02} - N_{01}) + 4KN_{01}N_{02}(M_{01} - M_{02}) + 2K(M_{01}P_{02} - M_{02}P_{01}) + 2K(M_{01}N_{02}^2 - M_{02}N_{01})^2$$

$$C_4 = K^2(M_{01}^2 - M_{02}^2 + N_{01}^2 - N_{02}^2 + 2P_{01} - 2P_{02}) + 2K(N_{02}M_{02}P_{01} - N_{01}M_{01}P_{02})$$

$$C_3 = 4K^3(M_{02} - M_{01}) + 4K^2M_{01}M_{02}(N_{01} - N_{02}) + 2K^2(N_{01}P_{02} - N_{02}P_{01} + N_{01}M_{02}^2 - N_{02}M_{01}^2)$$

---

[1]The coefficients C_3, and C_5 are incorrect in the reference but have been corrected here. An errata has been submitted to the AIAA Jorunal by Mr. Sun.

$$C_2 = 2K^3 (N_{O2}M_{O2} - N_{O1}M_{O1} + 4N_{O2}M_{O1} - 4N_{O1}M_{O2}) +$$

$$K^2 (M_{O1}{}^2P_{O2} - M_{O2}{}^2P_{O1})$$

$$C_1 = 4K^4(N_{O1} - N_{O2}) + 2K^3M_{O1}M_{O2}(M_{O2} - M_{O1}) + 2K^3(M_{O2}P_{O1} - M_{O1}P_{O2})$$

$$C_0 = K^4(M_{O1}{}^2 - M_{O2}{}^2 - P_{O1} + P_{O2})$$

with

$$M_{Oi} = V_{ROi} - V_{COi} \cos \phi_i$$

$$N_{Oi} = V_{COi} - V_{ROi} \cos \phi_i$$

these relations are equivalent to the previous definitions of M_{Oi} and N_{Oi}.

$$P_{Oi} = V_{Oi}{}^2 - 2K \cos \phi_i$$

and (i = 1,2)

$$K = (\mu/d) \tan (\psi/2)$$

The correct root of the octic defining the optimum two impulse transfer trajectory is obtained by first determining two other unique pairs of transfer trajectories between the two terminal conditions. One pair of trajectories represents the "single impulse optimum" posigrade and retrograde transfer associated with the initial state vector. This single impulse optimum transfer is that member of the family of trajectories passing through the initial and final position vectors that requires the least ΔV at the initial maneuver. The second pair is the posigrade and retrograde transfers associated with a minimum ΔV at the final maneuver.

These single impulse optimized cases are defined by differentiating Equation 2 and setting the result equal to zero. This yields the quartic equations

$$V_{Ci*}{}^4 - N_{Oi}V_{C*}{}^3 + KM_{Oi}V_{C*} - K^2 = 0 \quad (i = 1,2)$$

The two pairs of single impulse optimum solutions arise as the real roots of these two quartic equations. While the positive pair represent roots for one direction around the central body, the negative set are the roots for the opposite direction. With some minor exceptions representing extreme situations (see Reference (1)) only two real roots exist for each quartic equation.

For a given type of transfer trajectory (e.g. posigrade) we have the following inequalities:

$$|\Delta V_{1*}| < |\Delta V_{1**}| < |\Delta V_1|$$

$$|\Delta V_{2*}| < |\Delta V_{2**}| < |\Delta V_2|$$

where a single asterisk indicates that the velocity impulse has been minimized only at the specified terminal point, a double asterisk denotes the optimum two impulse transfer ΔV at the given terminal, and no asterisk denotes the velocity impulse required at the specified terminal when the ΔV has been minimized at the other terminal point.

TOHSUN uses this fact to define the regions in which to search for the proper roots of the octic (Equation 3). This method avoids calculating or using superfluous roots of the octic equation.

After determining the magnitude of V_{C1} and V_{C2} from the roots of the octic, the values of V_{R1} and V_{R2} are determined by the geometry as shown in Figure 1 and the relationship in Equation 1.

This method is repeated for retrograde transfers. The posigrade and retrograde two-impulse optimums are then compared and the one with the lowest total ΔV is selected as the best. A description and listing of FORTRAN subroutine TOHSUN is included in Appendix A.

Under the present structure of the program the angle of separation between the two terminal position vectors must not equal 180 degrees. The 180 degree case also has an analytic optimum but requires a unique formulation and is not included in TOHSUN.

Applications

TOHSUN is presently being used to determine the best transfer trajectory from a given pre-maneuver state point to a given post-maneuver state point as one link of a three impulse lunar orbit insertion optimization program. A simple iterative technique is used with TOHSUN to determine optimum three impulse transfers between two terminal conditions and a given intermediate position vector.

A transfer trajectory between the initial condition and the intermediate position vector is found with TOHSUN by assuming a reasonable velocity at the intermediate point. Using the pre-maneuver velocity at the intermediate point determined by TOHSUN, an optimum transfer to the final conditions is determined by a second use of TOHSUN. The post-maneuver velocity determined here for the intermediate state becomes the input for the next call to TOHSUN from the initial condition point. This flow is repeated until the pre and post-maneuver velocities at the intermediate position vector stabilize.

The total optimum 3 burn lunar orbit insertion requires selection of the best set of terminal conditions and intermediate position vector. The initial conditions are determined by the energy and momentum associated with the lunar approach hyperbola as well as the true anomaly at which the first maneuver is made. Final conditions are defined by specifying the position on the desired lunar orbit. The magnitude of the intermediate position vector is constrained by the maximum flight time allowed for the maneuver. Selection of all these parameters must be made external to the above iterative scheme involving TOHSUN.

R. A. Bass

E. A. McGinness

2013-$\frac{\text{RAB}}{\text{EAM}}$-srb

Attachments:
References
Appendices
Figures 1, D1, D2

References

1. Sun, Fang Toh, "Analysis of the Optimum Two-Impulse Orbital Transfer under Arbitrary Conditions", AIAA Journal, Vol. 6, No. 11, November 1968, pp. 2145-53.

2. McCormick, John M. and Salvadori, Mario G., Numerical Methods in Fortran, Prentice-Hall, Inc., 1964, pps. 61-63.

Appendix A

TOHSUN

Calling Sequence: CALL TOHSUN (VO1X, RO1X, VO2X, RO2X, V1BST, V2BST, BSTDV1, BSTDV2)

Input:

VO1X(3)	Initial velocity at first position vector
RO1X(3)	First position vector
VO2X(3)	Terminal velocity at second position vector
RO2X(3)	Second position vector

Output:

V1BST(3)	Transfer velocity at first position vector
V2BST(3)	Transfer velocity at second position vector
BSTDV1	Magnitude of the velocity increment vector at first position vector |V1BST-VO1X|
BSTDV2	Magnitude of the velocity increment vector at second position vector |V2BST-VO2X|

Description:

TOHSUN determines the minimum ΔV two-impulse transfer trajectory between two position vectors* whose angle of separation is not equal to 180°. The method used is to find two specific roots of an eighth degree polynomial in the variable V_C - the chordal component of the transfer velocity. These roots are constrained to lie in two intervals - one corresponding to a positive chordal component and the other corresponding to a negative chordal component.

---

*The program listing given here is designed for lunar trajectories. The user need only modify GMM to a suitable value for whatever central body is desired.

In order to establish these intervals the single impulse transfer trajectories that yield the minimum ΔV at each terminal are found. The endpoints of the intervals correspond to the chordal components of the minimum single impulse transfer velocity vectors. These numbers are the solutions of two quartic equations and are found by the subroutine QUART.

The octic equation is then solved for the desired roots within the intervals by subroutine OCTIC. The method used is a variation of the Newton-Raphson technique. In the event that OCTIC cannot find a root in a specified interval, the value of the variable V_C is selected which corresponds to the smallest absolute value of the octic equation that has been computed. If this occurs an error message is printed and the program continues. The coefficients and roots of the octic equation as well as those of the two quartic equations are computed in double precision.

The transfer velocities corresponding to the two roots of the octic equation are then computed. These two solutions are the posigrade and retrograde trajectories connecting states 1 and 2. The set requiring the lowest total ΔV is selected as the best and returned as V1BST and V2BST together with the corresponding magnitudes of the velocity increments BSTDV1 and BSTDV2.

TOHSUN calls the following subroutines:

a) OCTIC - see Appendix B

b) VALUE, DOT, CROSS - BCMASP vector subroutines

c) QUART - see Appendix C.

```
      SUBROUTINE TOHSUN(V01X,R01X,V02X,R02X,V1BST,V2BST,BSTDV1,BSTDV2)
      DOUBLE PRECISION COEF1(5),COEF2(5),XR(4),XI(4)
      DOUBLE PRECISION RDBL(2)
      DOUBLE PRECISION C(9)
      DOUBLE PRECISION M01,M02,N01,N02,P01,P02,K
      DOUBLE PRECISION AA,BB,AAA,BBB,POS1,POS2,XNEG1,XNEG2
      DIMENSION V1BST(3),V2BST(3)
      DIMENSION V1(3),V2(3),SWR02X(3)
      DIMENSION V01X(3), R01X(3), V02X(3), R02X(3), CHRD1X(3)
     1 ,F1(8), VR(8), VC(8), F2(8)
      DIMENSION SUMF(2)
C
C     SET CONSTANTS AND ESTABLISH VARIABLE VALUES
C
      FTMTR = 1. / .3048
      PI = 3.14159265
      GMM = 4.902778E3 * (1000. * FTMTR) ** 3
      DLVBST=1.0E+10
      R02=VALUE(R02X)
      R01=VALUE(R01X)
      DO 100 I=1,3
  100 CHRD1X(I)=R02X(I)-R01X(I)
      CHRD1=VALUE(CHRD1X)
      CHORD = CHRD1
      COSPSI = (DOT(R01X, R02X))/R01/R02
      PSI = ACOS(COSPSI)
      COSPH1 = DOT(CHRD1X, R01X) / CHRD1 / R01
      PH1 = PI - ACOS(COSPH1)
      COSPH2 = DOT(CHRD1X, R02X) / CHRD1 / R02
      PH2 = ACOS(COSPH2)
      D = SIN(PH1) * R01
      K = GMM / D * TAN(PSI/2.)
      V01 = VALUE(V01X)
      V02 = VALUE(V02X)
      M01 = DOT(V01X, R01X) / R01
      DO 142 KKJ = 1, 3
  142 SWR02X(KKJ) = -R02X(KKJ)
      M02 = DOT (V02X, SWR02X) / R02
      N01 = DOT (V01X, CHRD1X) / CHRD1
      N02 = DOT (V02X, CHRD1X) / CHRD1
      P01 = V01 ** 2 - 2. * K * COS(PH1)
      P02 = V02 ** 2 - 2. * K * COS(PH2)
      REDUCE = 1.E5
      M01=M01/REDUCE
      M02=M02/REDUCE
      N01=N01/REDUCE
      N02=N02/REDUCE
      P01=P01/REDUCE**2.0
      P02=P02/REDUCE**2.0
      K=K/REDUCE**2.0
C
C     DEFINE COEFFICIENTS OF THE QUARTIC EQUATIONS
C
      COEF1(1)=1.0
```

```
      COEF2(1)=1.0
      COEF1(2)=-N01
      COEF2(2)=-N02
      COEF1(3)=0.0
      COEF2(3)=0.0
      COEF1(4)=K*M01
      COEF2(4)=K*M02
      COEF1(5)=-K**2
      COEF2(5)=-K**2
C
C     SOLVE FOR REAL ROOTS OF QUARTIC EQUATIONS
C
      CALL QUART(COEF1,XR,XI)
      DO 165 I=1,4
      IF(XI(I))   144,145,144
  145 IF(XR(I))   146,147,147
  147 POS1=XR(I)
      GO TO 144
  146 XNEG1=XR(I)
  144 CONTINUE
      Y=K/SNGL(XR(I))
165   CONTINUE
      CALL QUART(COEF2,XR,XI)
      DO 166 I=1,4
      IF(XI(I)) 184,185,184
  185 IF(XR(I))  186,187,187
  187 POS2=XR(I)
      GO TO 184
  186 XNEG2=XR(I)
  184 CONTINUE
166   CONTINUE
C
C     DEFINE COEFFICIENTS OF THE OCTIC EQUATION
C
      C(9)=N01**2-N02**2-P01+P02
      C(8)=4*K*(M01-M02)+2*N01*N02*(N02-N01)+2*(N02*P01-N01*P02)
      C(7)=2*K*(N02*M02-N01*M01+4*N01*M02-4*N02*M01)+(N01**2*P02-N02**2
     .      *P01)
      C(6) = 4. * K ** 2 * (N02 - N01) + 4. * K*N01 * N02 * (M01 -
     1 M02) + 2. * K * (M01 * P02 - M02 * P01) + 2. * K * (M01 * N02**2
     2 - M02 * N01 ** 2)
      C(5)=K**2*(M01**2-M02**2+N01**2-N02**2+2*P01-2*P02)+2*K*(N02*M02
     .      *P01-N01*M01*P02)
      C(4) = 4. * K ** 3 * (M02 - M01) + 4. * K ** 2 * M01 * M02 *
     1 (N01 - N02) + 2. * K ** 2 * (N01 * P02 - N02 * P01 + N01 *
     2 M02 ** 2 - M01 ** 2 * N02)
      C(3)=2*K**3*(N02*M02-N01*M01+4*N02*M01-4*N01*M02)+K**2*(M01**2
     .      *P02-M02**2*P01)
      C(2)=4*K**4*(N01-N02)+2*K**3*M01*M02*(M02-M01)+2*K**3*(M02*P01
     .      -M01*P02)
      C(1)=K**4*(M01**2-M02**2-P01+P02)
      IF (POS1 .GT. POS2) GO TO 8000
      AA = POS1
      BB = POS2
      GO TO 8001
8000  AA = POS2
      BB = POS1
```

```
8001    IF (XNEG1 .GT. XNEG2) GO TO 8005
        AAA = XNEG1
        BBB = XNEG2
        GO TO 8010
8005    AAA = XNEG2
        BBB = XNEG1
C
C       SOLVE OCTIC EQUATION FOR THE DESIRED TWO REAL ROOTS
C
8010    CALL OCTIC(C,AA,BB,AAA,BBB,RDBL)
C
C       COMPUTE THE RESULTANT VELOCITY VECTORS AND RETURN THE PAIR WITH
C       THE LOWEST DELTA V
C
        DO 200 L = 1, 2
        I = L
        VC(I) = RDBL(I)
        VR(I)=K/VC(I)
        F1(I) =  VC(I) ** 2 + VR(I) ** 2 - 2.*NO1 * VC(I) - 2. * MO1 *
      1 VR(I) + PO1
        F2(I) =  VC(I) ** 2 + VR(I) ** 2 - 2.*NO2 * VC(I) - 2. * MO2 *
      1 VR(I) + PO2
        IF (F1(I) .LE. 0.0) WRITE(6,999) I, F1(I)
        F1(I) = SQRT(ABS(F1(I))) * REDUCE
999     FORMAT(' I =', I2, '      F1(I) =', G30.20)
        IF (F2(I) .LE. 0.0) WRITE(6,998) I, F2(I)
        F2(I) = SQRT(ABS(F2(I))) * REDUCE
998     FORMAT(' I =', I2, '      F2(I) =', G30.20)
        SUMF(I) = F1(I) + F2(I)
        DO 202 J=1,3
        V1(J)=VC(I)*CHRD1X(J)/CHRD1+VR(I)* RO1X(J)/ RO1
        V2(J) = VC(I) * CHRD1X(J) / CHRD1 + VR(I) * SWRO2X(J) / RO2
202     CONTINUE
        SUMDLV = SUMF(I)
        DLVT1 = F1(I)
        DLVT2 = F2(I)
        IF(SUMDLV-DLVBST) 3001,3001,3002
3001    DO 3003 J=1,3
        V1BST(J)=V1(J)*REDUCE
3003    V2BST(J)=V2(J)*REDUCE
        BSTDV1 = DLVT1
        BSTDV2 = DLVT2
        DLVBST=SUMDLV
3002    CONTINUE
200     CONTINUE
        RETURN
        END
```

Appendix B

OCTIC

Calling Sequence: CALL OCTIC (C, A, B, Q, R, RR)

Input:

C	Nine element array of the coefficients of the octic polyniminal - $$\sum_{I=1}^{9} C(I)X^{I-1} = 0$$
A,B	Lower and upper endpoints of the first interval
Q,R	Lower and upper endpoints of the second interval

Output:

RR	Two element array of the desired roots, or, in the event OCTIC failed to converge, the numbers corresponding to the smallest absolute error.

Parameters:

N	Maximum number of iterations, N = 200
E	Error tolerance, $E = 1.0 \times 10^{-15}$

Description:

OCTIC is designed to find two real roots of an eighth degree polynominal, each of which lies in a specified interval. The program uses the Newton-Raphson method of tangents to find the roots (Reference 2). An initial guess at the value of the root is necessary to start the iteration process. The midpoint

of the interval is this starting value. If a root is found which does not lie in the desired interval, or if no root is found, a new iteration process is begun with the initial value set equal to $R_o = G+(\frac{H-G}{4})$ where G is the lower endpoint of the interval and H is the upper endpoint. If an acceptable root is again not found within 200 iterations, a third attempt is made with the initial value $R_o = H-(\frac{H-G}{4})$. If OCTIC fails to converge an error message is printed out and the value of the variable corresponding to the smallest absolute error computed during the iterations is returned as the root. If OCTIC converges the roots are returned as follows:

RR(1) = root in the first interval

RR(2) = root in the second interval.

```
      SUBROUTINE OCTIC(C,A,B,Q,R,RR)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION C(9), RR(2)
C
C
C     INPUTS
C     C        ARRAY OF NINE COEFFICIENTS
C     A,Q      LOWER ENDPOINTS OF THE TWO INTERVALS
C     B,R      UPPER ENDPOINTS OF THE TWO INTERVALS
C
C     OUTPUTS
C     RR       ARRAY OF THE DESIRED TWO ROOTS, OR, IN THE EVENT OCTIC
C              FAILED TO CONVERGE, THE NUMBERS CORRESPONDING TO THE
C              SMALLEST ABSOLUTE ERROR
C
C     PARAMETERS
C     E        ERROR TOLERANCE, CURRENTLY E = 1.D-15
C     N        NUMBER OF ITERATIONS, CURRENTLY N = 200
C
C
      E = .1D-14
      K = 0
      N = 200
      Y1 = (B-A) / 2.
      X1 = A + Y1
      X=X1
      G=A
      H=B
      M=1
      II = 1
      JK = 1
19    CONTINUE
20    FX =C(1)+C(2)*X+C(3)*(X**2)+C(4)*(X**3)+C(5)*(X**4)+C(6)*(X**5)
     1 +C(7)*(X**6)+C(8)*(X**7)+C(9)*(X**8)
      FXP =C(2)+2.*C(3)*X+3.*C(4)*(X**2)+4.*C(5)*(X**3)+5.*C(6)*(X**4)
     1 +6.*C(7)*(X**5)+7.*C(8)*(X**6)+8.*C(9)*(X**7)
      IF (K .EQ. 0) GO TO 90
      IF (ABS(FX) .GT. ABS(FBEST)) GO TO 91
      IF (X .LT. G .OR. X .GT. H) GO TO 91
90    FBEST = FX
      XBEST = X
C
C     NEWTON - RAPHSON ITERATION FOR REAL ROOT OF F(X) = 0
C
91    IF (K .GT. 0) GO TO 1
      K=1
      NMAX=N
      N=0
1     T=FX
      IF (.NOT.ABS(T) .GT. E) GO TO 4
3     IF (N .EQ. NMAX) GO TO 5
      IF (.NOT. ABS(FXP) .GT. 0.) GO TO 7
      T=X
      X=X-(FX/FXP)
```

```
      N=N+1
      IF (.NOT. ABS(T-X) .GT. 0.) GO TO 6
      GO TO 22
4     K=2
      IF (X .LT. G .OR. X .GT. H) GO TO 86
      GO TO 22
5     K=3
      GO TO 22
6     K=4
      GO TO 22
7     K=5
      GO TO 22
9     K = 6
22    CONTINUE
      GO TO (20, 30, 40, 50, 60, 80), K
30    IF (M .EQ. 2) GO TO 35
      RR(1) = X
      GO TO 10
35    RR(2) = X
      RETURN
C     START COMPUTATION ON THE INTERVAL (Q,R)
10    I1 = 1
      Y1 = (R-Q) / 2.
      X1 = Q + Y1
      N = NMAX
      K=0
      X = X1
      G = Q
      H = R
      M=2
      GO TO 20
40    WRITE (6,41)
41    FORMAT (' ITERATION FOR ROOT HAS NOT CONVERGED IN N ITERATIONS')
      IF (M .EQ.2) GO TO 99
      RR(1) = XBEST
      GO TO 10
50    WRITE (6,51)X,FX,N,K
51    FORMAT (' UNDERFLOW IN ITERATION '/ ' X=', G25.8, '      FX =',
     1 G25.8, '      N=', I3, '      K =', I2)
      IF (M .EQ.2) GO TO 99
      RR(1) = XBEST
      GO TO 10
60    WRITE (6,61)
61    FORMAT (' DERIVATIVE EQUALS 0, Q CANNOT BE COMPUTED')
      IF (M .EQ.2) GO TO 99
      RR(1) = XBEST
      GO TO 10
      IF (M .EQ.2) GO TO 99
      RR(1) = XBEST
      GO TO 10
80    WRITE (6,81)
81    FORMAT (' ITERATION CANNOT FIND ROOT IN SPECIFIED INTERVAL')
      IF (M .EQ.2) GO TO 99
      RR(1) = XBEST
      GO TO 10
86    IF (II .GT. 1) GO TO 82
      Y1 = Y1 / 2.
```

```
      X1 = G + Y1
      X = X1
      II = II + 1
      N = NMAX
      K = 0
      GO TO 19
82    IF (II .EQ. 3) GO TO 9
      X1 = H - Y1
      X = X1
      II = II + 1
      N = NMAX
      K = 0
      GO TO 19
99    RR(2) = XBEST
      RETURN
      END
```

Appendix C

```
C                                QUART
C
C          SOLUTION OF THE QUARTIC EQUATION
C          C(1)*X**4 + C(2)*X**3 + C(3)*X**2 + C(4)*X + C(5) = 0.
C
      SUBROUTINE QUART(C,XR,XI)
      DIMENSION C(5),  XR(4),  XI(4),  AC(4),  AQ(3),  BQ(3),  RT(3)
      EQUIVALENCE (AQ,BQ)
      IMPLICIT DOUBLE PRECISION (A-Z)
      A3=C(2)/C(1)
      A2=C(3)/C(1)
      A1=C(4)/C(1)
      A0=C(5)/C(1)
      A=A3/2.
      AC(1)=1.
      AC(2)=-A2
      AC(3)=A1*A3-4.*A0
      AC(4)=A0*(4.*A2-A3*A3)-A1*A1
      CALL CUBIC(AC,RT,RTI)
      IF(RTI) 20,10,20
 10   IF(RT(1)-RT(2)) 11,12,12
 11   RT(1)=RT(2)
 12   IF(RT(1)-RT(3)) 13,20,20
 13   RT(1)=RT(3)
 20   B=RT(1)/2.
      IF(B*B-A0) 22,22,24
 22   D=0.
      CA=SQRT(A*A+2.*B-A2)
      GO TO 25
 24   D=SQRT(B*B-A0)
      CA=-(A1/2.-A*B)/D
 25   AQ(1)=1.
      AQ(2)=A-CA
      AQ(3)=B-D
      CALL QUAD(AQ,XR(1),XR(2),XI(1))
      BQ(2)=A+CA
      BQ(3)=B+D
      CALL QUAD(BQ,XR(3),XR(4),XI(3))
      XI(2)=-XI(1)
      XI(4)=-XI(3)
      RETURN
      END
```

NOTE: Discussion of all routines in Appendix C can be found in Reference 2.

```
C                                CUBIC
C
C          SOLUTION OF THE CUBIC EQUATION
C          A(1)*X**3 + A(2)*X + A(3)*X + A(4) = 0.
C
      SUBROUTINE CUBIC(A,XR,XI)
      DIMENSION A(4),  XR(3),  AQ(3)
      IMPLICIT DOUBLE PRECISION (A-H,J-Z)
      IPATH=2
      EX=1./3.
      IF(A(4)) 1006,1004,1006
 1004 XR(1)=0.
      GO TO 1034
 1006 A2=A(1)*A(1)
      Q=(27.*A2*A(4)-9.*A(1)*A(2)*A(3)+2.*A(2)**3)/(54.*A2*A(1))
      IF (Q) 1010,1008,1014
 1008 Z=0.
      GO TO 1032
 1010 Q=-Q
      IPATH=1
 1014 P=(3.*A(1)*A(3)-A(2)*A(2))/(9.*A2)
      ARG=P**3+Q*Q
      IF(ARG) 1016,1018,1020
 1016 Z=-2.*SQRT(-P)*COS(ATAN(SQRT(-ARG)/Q)/3.)
      GO TO 1028
 1018 Z=-2.*Q**EX
      GO TO 1028
 1020 SARG=SQRT(ARG)
      IF(P) 1022,1024,1026
 1022 Z=-(Q+SARG)**EX-(Q-SARG)**EX
      GO TO 1028
 1024 Z=-(2.*Q)**EX
      GO TO 1028
 1026 Z=(SARG-Q)**EX-(SARG+Q)**EX
 1028 GO TO (1030,1032),IPATH
 1030 Z=-Z
 1032 XR(1)=(3.*A(1)*Z-A(2))/(3.*A(1))
 1034 AQ(1)=A(1)
      AQ(2)=A(2)+XR(1)*A(1)
      AQ(3)=A(3)+XR(1)*AQ(2)
      CALL QUAD(AQ,XR(2),XR(3),XI)
      RETURN
      END
```

```
C                                QUAD
C
C          SOLUTION OF THE QUADRATIC EQUATION
C          A(1)*X**2 + A(2)*X + A(3) = 0.
C
      SUBROUTINE QUAD(A,XR1,XR2,XI)
      DIMENSION A(3)
      IMPLICIT DOUBLE PRECISION (A-Z)
      X1=-A(2)/(2.*A(1))
      DISC=X1*X1-A(3)/A(1)
      IF(DISC) 10,20,20
   10 X2=SQRT(-DISC)
      XR1=X1
      XR2=X1
      XI=X2
      GO TO 30
   20 X2=SQRT(DISC)
      XR1=X1+X2
      XR2=X1-X2
      XI=0.
   30 RETURN
      END
```

Appendix D

Derivation of Equation 1

By conservation of angular momentum

$$\bar{R}_1 \times \bar{V}_1 = \bar{R}_2 \times \bar{V}_2 = \bar{H}$$

but using Figure D1 it is apparent

$$|\bar{R}_1 \times \bar{V}_1| = (|\bar{V}_{C1}|) \cdot (d) \text{ and}$$

$$|\bar{R}_2 \times \bar{V}_2| = (|\bar{V}_{C2}|) \cdot (d)$$

leading to $|\bar{V}_{C1}| = |\bar{V}_{C2}|$

and since by definition the chordal components are along the chordal direction

$$\bar{V}_{C1} = \bar{V}_{C2} \tag{D1}$$

Also

$$|\bar{H}| = r^2 \frac{d\theta}{dt} = h \text{ where } \theta \text{ is defined in Figure D1.}$$

The equation of motion

$$\frac{d\bar{V}}{dt} = \frac{-\mu \bar{r}}{r^3}$$

can now be written

$$\frac{d\bar{V}}{dt} = \frac{-\mu}{h} \frac{\bar{r}}{r} \frac{d\theta}{dt}$$

Integrating

$$\int_{\bar{V}_1}^{\bar{V}_2} d\bar{V} = \frac{-\mu}{h} \int_{\theta_1}^{\theta_2} \frac{\bar{r}}{r} \, d\theta =$$

$$= \frac{-\mu}{h} \int_{\theta_1}^{\theta_1} (\bar{i} \cos \theta + \bar{j} \sin \theta) d\theta$$

$$\bar{V}_2 - \bar{V}_1 = \frac{-\mu}{h} (\bar{i} \sin \theta - \bar{j} \cos \theta) \Big|_{\theta_1}^{\theta_2}$$

$$(\bar{V}_{2R} + \bar{V}_{2C}) - (\bar{V}_{1R} + \bar{V}_{1C}) = \frac{-\mu}{h} (\bar{i} (\sin \theta_2 - \sin \theta_1) -$$

$$\bar{j} (\cos \theta_2 - \cos \theta_1))$$

Using Equation D1 and the identities

$$\sin \theta_2 - \sin \theta_1 = 2 \cos \left(\frac{\theta_2+\theta_1}{2}\right) \sin \left(\frac{\theta_2-\theta_1}{2}\right)$$

$$\cos \theta_2 - \cos \theta_1 = 2 \sin \left(\frac{\theta_2+\theta_1}{2}\right) \sin \left(\frac{\theta_2-\theta_1}{2}\right)$$

$$\bar{V}_{2R} - \bar{V}_{1R} = \frac{-\mu}{h} \left(2 \sin \left(\frac{\theta_2-\theta_1}{2}\right)\right) \left(\bar{i} \cos \left(\frac{\theta_2+\theta_1}{2}\right) + \bar{j} \sin \left(\frac{\theta_2+\theta_1}{2}\right)\right)$$

and

$$\bar{V}_{2R} - \bar{V}_{1R} = \frac{-\mu}{h} \left(2 \sin \left(\frac{\theta_2+\theta_1}{2}\right)\right) \bar{B}$$

and $\overline{B}$ is along the bisector of $\overline{R}_1$ and $\overline{R}_2$.

Since $\overline{B}$ is that bisector, λ_1 and λ_2 are equal in Figure D2. The triangle formed by $\overline{V}_{R1}$, $\overline{V}_{R2}$, and $\overline{B}$ is isosceles and

$$|\overline{V}_{R1}| = |\overline{V}_{R2}|. \tag{D2}$$

From the conservation of energy,

$$\frac{V_1^2}{2} - \frac{\mu}{R_1} = \frac{V_2^2}{2} - \frac{\mu}{R_2}, \tag{D3}$$

and from the cosine law,

$$V_i^2 = V_{iR}^2 + V_{iC}^2 - 2\, V_{iR}\, V_{iC} \cos \phi_i. \tag{D4}$$

Noting that $r_i = d/\sin \phi_i$ and substituting D4 and D2 into D3

$$V_{C1}\, V_{R1} \cos \phi_1 - \frac{\mu}{d} \sin \phi_1 = V_{C1}\, V_{R1} \cos \phi_2 - \frac{\mu}{d} \sin \phi_2$$

$$V_{C1}\, V_{R1} (\cos \phi_1 - \cos \phi_2) = \frac{\mu}{d} (\sin \phi_1 - \sin \phi_2)$$

$$V_{C1}\, V_{R1} = \frac{\mu}{d} \frac{\sin \phi_1 - \sin \phi_2}{\cos \phi_1 - \cos \phi_2} = \frac{-\mu}{d} \cot \left(\frac{\phi_1 + \phi_2}{2}\right)$$

but

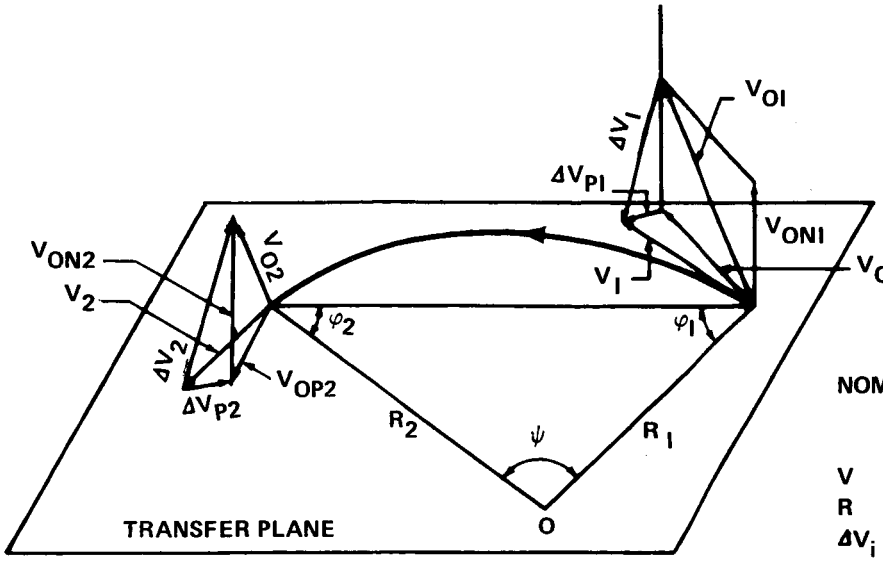
$$\phi_1 + \phi_2 + \psi = 180$$

so

$$- \cot \left(\frac{\phi_1 + \phi_2}{2}\right) = \tan \frac{\psi}{2}$$

and finally

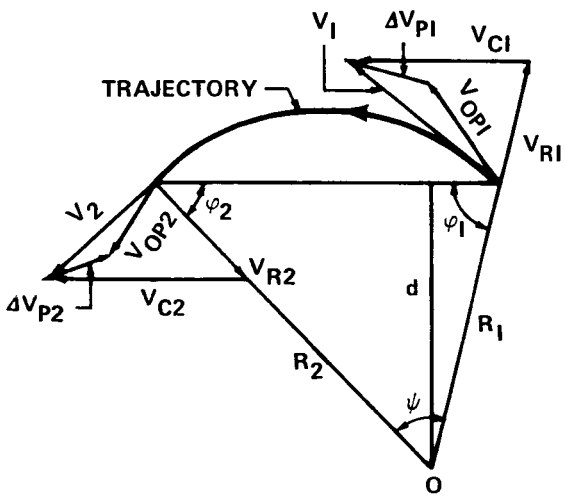
$$V_{R1} V_{C1} = V_{R2} V_{C2} = \frac{\mu}{d} \tan \left(\frac{\psi}{2}\right)$$



A) THREE-DIMENSIONAL VIEW

NOMENCLATURE

V = VELOCITY VECTOR
R = RADIUS VECTOR
ΔV_i = VELOCITY CHANGE $V_i - V_{Oi}$



B) TRANSFER TRAJECTORY

SUBSCRIPTS

i = INDEX (= 1, 2)
1 = AT THE INITIAL POSITION
2 = AT THE FINAL POSITION
0 = GIVEN VELOCITY AT 1 OR 2
P,N = IN PLANE AND OUT OF PLANE COMPONENTS
R = RADIAL
C = CHORDAL

FIGURE 1- TRANSFER GEOMETRY

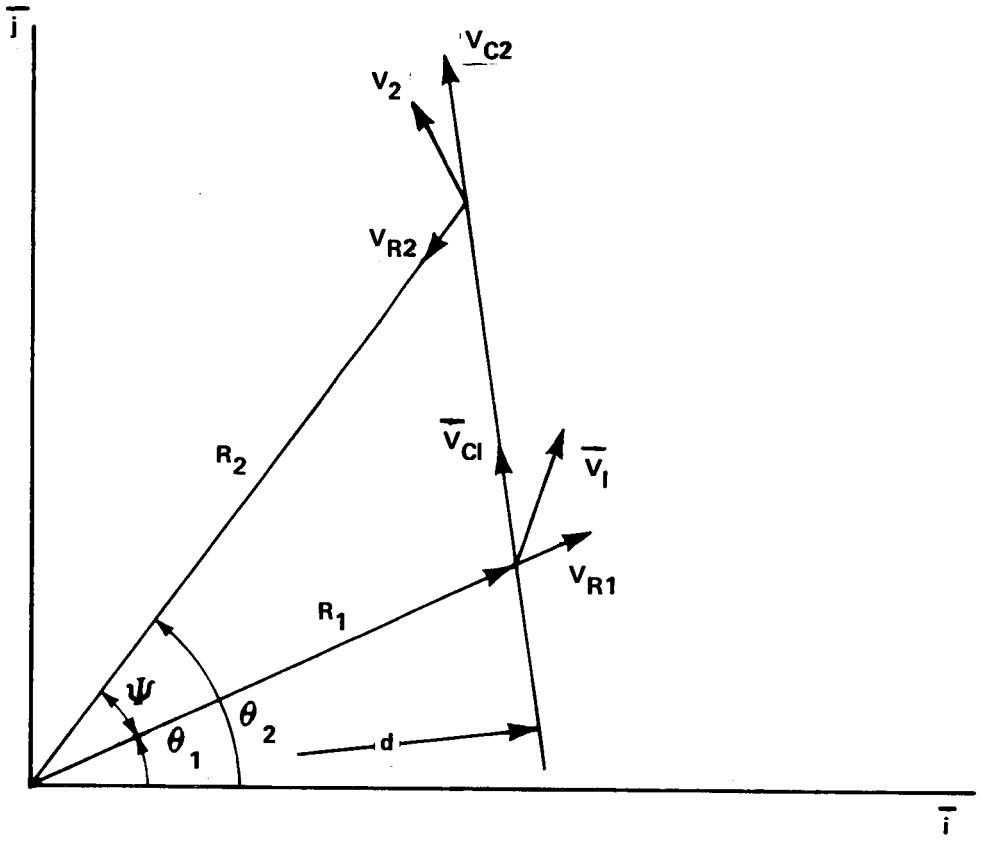


FIGURE D1

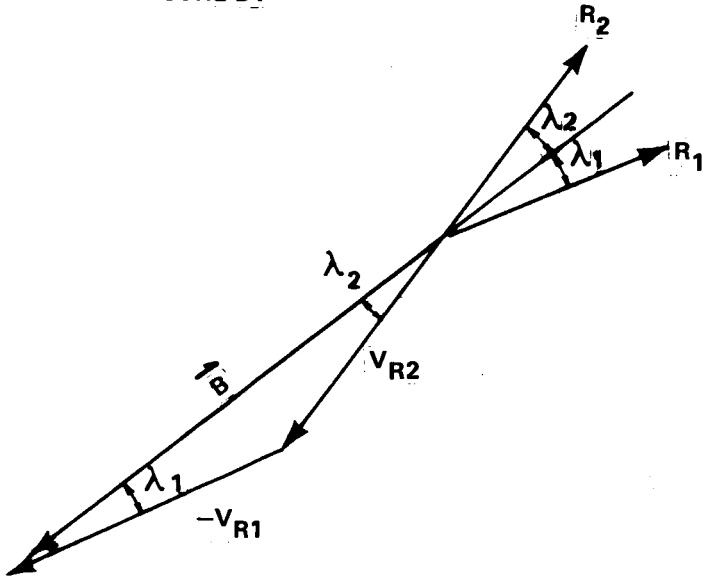


FIGURE D2

BELLCOMM, INC.

Subject: A Method for Determining Minimum ΔV Two Impulse Transfer Trajectories Between Arbitrary State Vectors and Its Application to Three Impulse LOI Optimization Case 310

From: R. A. Bass
E. A. McGinness

Distribution List

NASA Headquarters

T. A. Keegan/MA-2

MSC

H. D. Beck/FM5
J. R. Elk/FM5
J. Funk/FM8

NASA Electronics Research Center

F. T. Sun

Bellcomm, Inc.

D. R. Anselmo
A. P. Boysen, Jr.
J. O. Cappellari, Jr.
D. A. Corey
D. R. Hagner
W. G. Heffron
T. B. Hoekstra
D. B. James
S. L. Levie
H. S. London
J. L. Marshall, Jr.
J. Z. Menard
V. S. Mummert
P. E. Reynolds
R. L. Wagner
M. P. Wilson
Central Files
Department 1024 File
Library

Abstract Only to

Bellcomm, Inc.

B. T. Howard
I. M. Ross
J. W. Timko