Optimal Impulsive Orbital Transfer – SNOPT version

This document describes an algorithm and MATLAB script called `oota_snopt` that can be used to determine optimum one and two impulse orbital transfers between *non-coplanar* circular and elliptical orbits. The method is general and the initial and final orbits need not be coapsidal.

The algorithm implemented in this script is based on the classic orbit transfer and rendezvous work of Gary McCue, Gentry Lee and David Bender, described in "Numerical Investigation of Minimum Impulse Orbital Transfer", *AIAA Journal*, **3**, 2328-2334 (1963); and "An Analysis of Two-Impulse Orbital Transfer", *AIAA Journal*, **2**, 1767-1773 (1964).

The numerical solution of this classic astrodynamics problem involves a combination of one-dimensional, derivative-free root-finding using Richard Brent's method and multi-dimensional unconstrained minimization using the SNOPT algorithm. Information about MATLAB versions of SNOPT for several computer platforms can be found at Professor Philip Gill's web site which is located at http://scicomp.ucsd.edu/~peg/. Professor Gill's web site also includes a PDF version of the SNOPT software user's guide.

For a two-impulse transfer, This script also provides a graphical display of the primer vector characteristics. This information can be used to verify the optimality of the transfer.

Data file format

The script reads a simple ASCII data file that defines the initial and final orbits along with the algorithm search characteristics. The following is a typical data file named `leo2gto.in` for this application.

This example solves the problem of two impulse, non-coplanar orbital transfer from a typical low altitude Earth orbit (LEO) to a geosynchronous transfer orbit (GTO). The annotation text in this file can be modified but should not be deleted because the routine that reads this data expects to find exactly 81 lines of text and numeric information.

The first two data items define the gravitational constant and radius of the central body. Please note the units and valid range for each user input highlighted in bold text.

```
************************************************
* input data file for oota_snopt.m MATLAB script
* impulsive LEO-to-GTO orbital transfer
* filename ==> leo2gto.in
************************************************

central body gravitational constant (kilometers^3/seconds^2)
398600.5

central body radius (kilometers)
6378.14

*************
initial orbit
*************

semimajor axis (kilometers)
(semimajor axis > 0)
6563.14
```

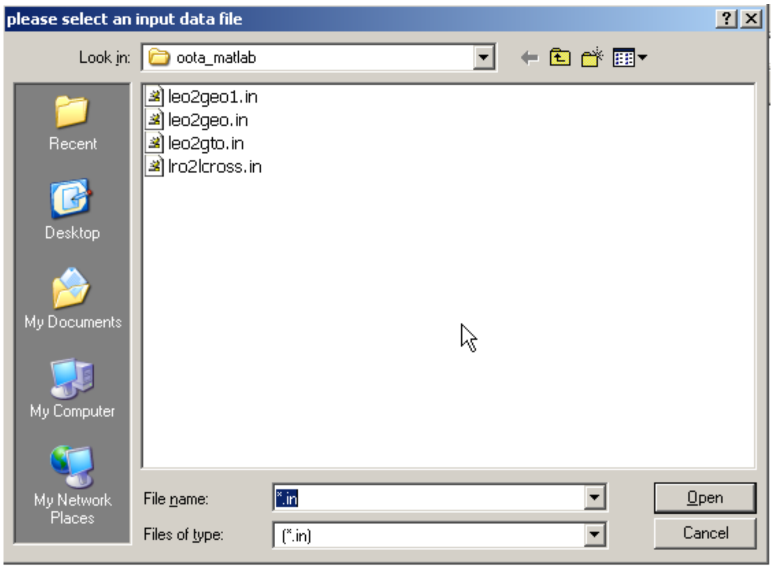
```
orbital eccentricity (non-dimensional)
(0 <= eccentricity < 1)
0.015

orbital inclination (degrees)
(0 <= inclination <= 180)
28.5

argument of perigee (degrees)
(0 <= argument of perigee <= 360)
270

right ascension of the ascending node (degrees)
(0 <= raan <= 360)
60

***********
final orbit
***********

semimajor axis (kilometers)
(semimajor axis > 0)
24364.8

orbital eccentricity (non-dimensional)
(0 <= eccentricity < 1)
0.73062206

orbital inclination (degrees)
(0 <= inclination <= 180)
26.3355

argument of perigee (degrees)
(0 <= argument of perigee <= 360)
270

right ascension of the ascending node (degrees)
(0 <= raan <= 360)
60

***************************
algorithm search parameters
***************************

initial orbit true anomaly at which to begin search (degrees)
0

final orbit true anomaly at which to begin search (degrees)
0

initial orbit true anomaly search increment (degrees)
60

final orbit true anomaly search increment (degrees)
60

number of initial orbit true anomaly search intervals
6

number of final orbit true anomaly search intervals
6
```

The last section of this data file defines the algorithm grid search parameters to use during the optimization. These numbers define the initial true anomaly for the initial and final orbits, the true anomaly search increment for each orbit, and the total number of intervals to analyze.

Notice that the combination of true anomaly search increments and number of search intervals in this example will encompass the entire true anomaly range for both the initial and final orbits.

Running the oota_snopt script

When the `oota_snopt` script is started, the software will display a screen similar to the following which allows the user to select a data file for processing.



please select an input data file

Look in: oota_matlab

leo2geo1.in
leo2geo.in
leo2gto.in
lro2lcross.in

File name: *.in

Files of type: (*.in)

The file type defaults to names with a `*.in` filename extension. However, you can select any `oota_snopt` compatible ASCII data file by selecting the Files of type: field or by typing the name of the file directly in the File name: field.

Optimal solution

The following is the output created by the `oota_snopt` script for a LEO-to-GTO orbit transfer example. The pitch and yaw angles for each impulsive maneuver are computed and displayed in the local-vertical-local horizontal (LVLH) coordinate system.

```
 program oota_snopt - SNOPT version

 < optimal orbital transfer analysis >

 initial orbit - prior to the first impulse
 ------------------------------------------

        sma  (km)              eccentricity          inclination (deg)         argper (deg)
  +6.56314000000000e+03   +1.49999999999998e-02   +2.84999999999999e+01   +2.70000000000000e+02

        raan (deg)           true anomaly (deg)         arglat (deg)            period (min)
  +6.00000000000000e+01   +3.59106563416917e+02   +2.69106563416917e+02   +8.81915957810343e+01

        rx  (km)                 ry  (km)                 rz  (km)                rmag  (km)
  +4.86914450945849e+03   -2.92759882870748e+03   -3.08431537131980e+03   +6.46470451494875e+03

        vx  (kps)                vy  (kps)                vz  (kps)               vmag  (kps)
  +4.04748953403655e+00   +6.79685126966771e+00   -5.79893644770649e-02   +7.91092418599189e+00

 transfer orbit - after the first impulse
 ----------------------------------------
```

```
        sma (km)              eccentricity          inclination (deg)         argper (deg)
 +2.31089169970822e+04   +7.20262221535054e-01    +2.85356428985632e+01   +2.66311610206117e+02

       raan (deg)          true anomaly (deg)         arglat (deg)            period (min)
 +6.22642200655780e+01   +8.05394903210770e-01    +2.67117005109327e+02   +5.82678625956222e+02

        rx (km)                 ry (km)                rz (km)                  rmag (km)
 +4.86914450945849e+03   -2.92759882870749e+03    -3.08431537131980e+03   +6.46470451494875e+03

        vx (kps)                vy (kps)               vz (kps)                 vmag (kps)
 +5.23533351528738e+00   +8.86465188982699e+00    -2.76365033735933e-01   +1.02988906089917e+01

transfer orbit - prior to the second impulse
--------------------------------------------

        sma (km)              eccentricity          inclination (deg)         argper (deg)
 +2.31089169970823e+04   +7.20262221535055e-01    +2.85356428985632e+01   +2.66311610206117e+02

       raan (deg)          true anomaly (deg)         arglat (deg)            period (min)
 +6.22642200655780e+01   +1.18056542754014e+02    +2.43681529601304e+01   +5.82678625956225e+02

        rx (km)                 ry (km)                rz (km)                  rmag (km)
 +1.73402775326040e+03   +1.63965935980509e+04    +3.31482252917502e+03   +1.68179422721285e+04

        vx (kps)                vy (kps)               vz (kps)                 vmag (kps)
 -3.17182402302713e+00   +3.73876246361085e+00    +2.47269665113150e+00   +5.49117848197219e+00

final orbit - after the second impulse
--------------------------------------

        sma (km)              eccentricity          inclination (deg)         argper (deg)
 +2.43648000000000e+04   +7.30622060000000e-01    +2.63355000000000e+01   +2.70000000000000e+02

       raan (deg)          true anomaly (deg)         arglat (deg)            period (min)
 +6.00000000000000e+01   +1.16378140978541e+02    +2.63781409785408e+01   +6.30817851038059e+02

        rx (km)                 ry (km)                rz (km)                  rmag (km)
 +1.73402775326040e+03   +1.63965935980509e+04    +3.31482252917503e+03   +1.68179422721285e+04

        vx (kps)                vy (kps)               vz (kps)                 vmag (kps)
 -3.27096389398481e+00   +3.84706378023522e+00    +2.35436281135017e+00   +5.57154635378892e+00

ECI delta-v vectors, magnitudes and LVLH angles
-----------------------------------------------

delta-v1x            1187.8440 meters/second
delta-v1y            2067.8006 meters/second
delta-v1z            -218.3757 meters/second

delta-v1             2394.6734 meters/second

LVLH pitch angle        1.4940 degrees
LVLH yaw angle          4.6578 degrees

delta-v2x             -99.1399 meters/second
delta-v2y             108.3013 meters/second
delta-v2z            -118.3338 meters/second

delta-v2              188.5757 meters/second

LVLH pitch angle       22.4598 degrees
LVLH yaw angle         77.2200 degrees

total delta-v        2583.2491 meters/second

transfer time       2864.3401 seconds
                      47.7390 minutes
                       0.7957 hours
```

The following is a brief description of the information provided in the screen display created by the `oota_snopt` script.

sma (km) = semimajor axis in kilometers

eccentricity = orbital eccentricity (non-dimensional)

inclination (deg) = orbital inclination in degrees

argper (deg) = argument of perigee in degrees

raan (deg) = right ascension of the ascending node in degrees

true anomaly (deg) = true anomaly in degrees

arglat (deg) = argument of latitude in degrees. The argument of latitude is the sum of true anomaly and argument of perigee.

period (mins) = orbital period in minutes

rx (km) = x-component of the position vector in kilometers

ry (km) = y-component of the position vector in kilometers

rz (km) = z-component of the position vector in kilometers

rmag (km) = scalar magnitude of the position vector in kilometers

vx (kps) = x-component of the velocity vector in kilometers per second

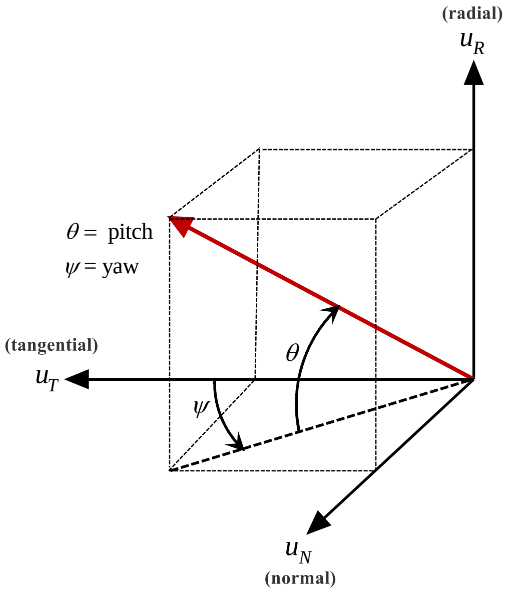
vy (kps) = y-component of the velocity vector in kilometers per second

vz (ksp) = z-component of the velocity vector in kilometers per second

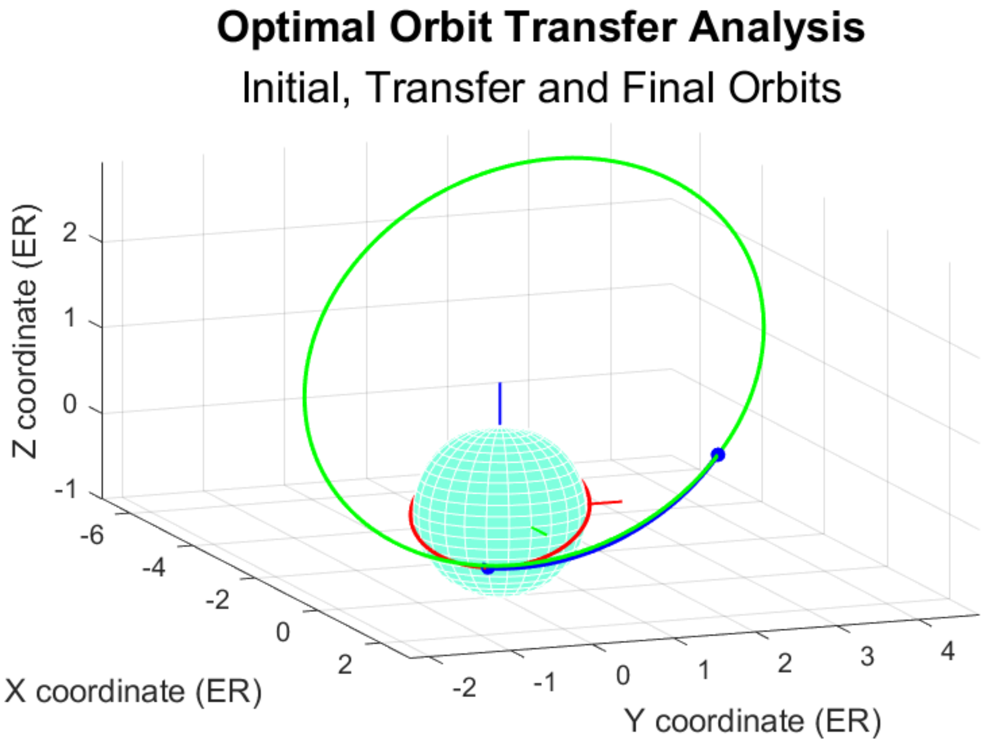
vmag (kps) = scalar magnitude of the velocity vector in kilometers per second

transfer time = flight time between the two impulses in seconds, minutes and hours

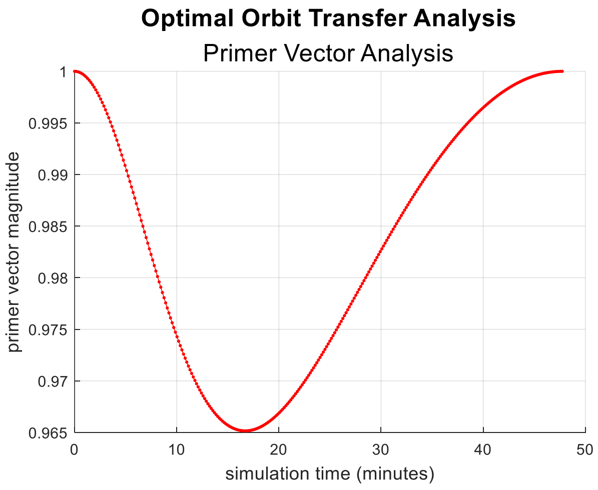
The pitch and yaw angles for each impulsive maneuver are computed and displayed in a local-vertical-local horizontal coordinate system. The following diagram illustrates the geometry of the pitch and yaw angles in this system. In this figure, the radial direction is along the geocentric radius vector directed away from the Earth, the tangential direction is tangent to the orbit in the direction of the orbital motion, and the normal direction is along the angular momentum vector of the orbit. The pitch angle is positive above the local horizontal plane formed by the tangential and normal directions, and the yaw angle is positive in the direction of the angular momentum vector which is perpendicular to the orbit plane.

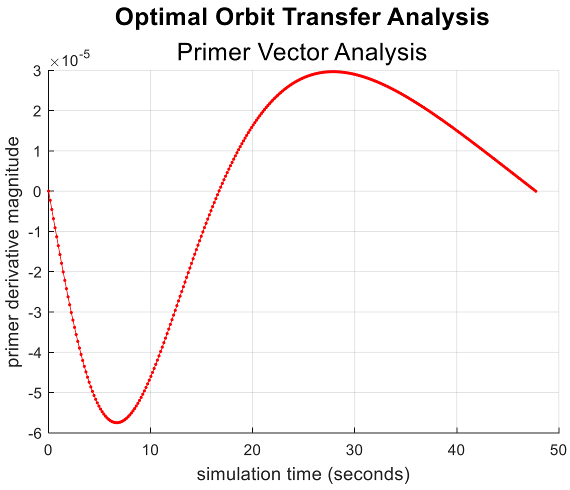


The `oota_snopt` script will also create a graphics display of the initial, transfer and final orbits. The following is the graphics display for this example. The initial orbit trace is red, the transfer orbit is blue and the final mission orbit is green. The dimensions are Earth radii (ER) and the plot is labeled with an ECI coordinate system where green is the x-axis, red is the y-axis and blue the z-axis. The location of each impulse is marked with a small blue dot. The interactive features of MATLAB graphics allow the user to interactively find the best viewpoint as well as verify basic three-dimensional geometry of the orbital maneuver.



The graphical primer vector analysis for this example is shown below. These plots illustrate the behavior of the scalar magnitudes of the primer vector and its derivative as a function of the simulation or elapsed time. From the properties of the primer vector and its derivative, we can see that this is the optimal solution to this orbit transfer problem.

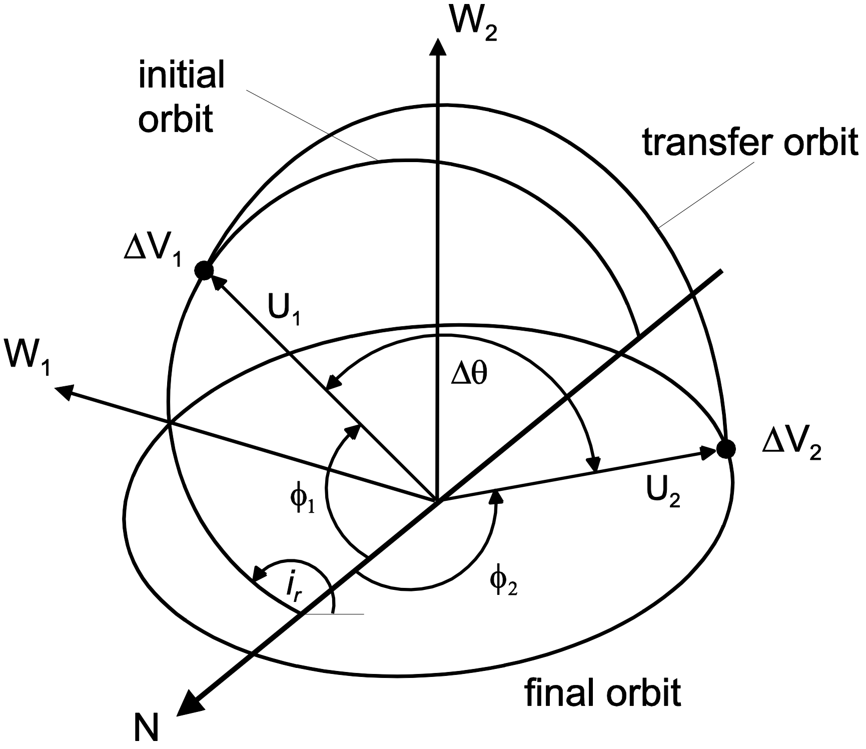




Technical Discussion

The solution to this important astrodynamics problem is formulated in a *reference coordinate* system. The fundamental reference plane of this coordinate system is the final orbit plane and the x-axis is aligned with the intersection of the planes of the initial and final orbits. The z-axis of this system is aligned with the angular momentum vector of the final orbit and the y-axis completes this orthogonal coordinate system. In the equations which follow, elements of the initial orbit have a subscript of *1* and elements of the final orbit a subscript of *2*. Elements of the transfer orbit will have a subscript of *t*.

The following diagram illustrates the geometry of two impulse orbital transfer. The relative inclination between the initial and final orbit planes is i_r and $\Delta\theta$ is the transfer angle which is the angle from the first and second impulse measured in the plane of the transfer orbit. **N** corresponds to the x-axis, $\mathbf{W}_1$ is in the direction of the initial orbit angular momentum vector, and $\mathbf{W}_2$ is in the direction of the angular momentum vector of the final orbit.



The independent variables for this problem are ϕ_1, ϕ_2 and p_t, where ϕ_1 is the angle from the **N** axis to the first impulse as measured in the initial orbit plane, ϕ_2 is the angle from the **N** axis to the second impulse as measured in the final orbit plane, and p_t is the semiparameter of the transfer orbit. The expression for **N** is as follows:

$$\mathbf{N} = \frac{\mathbf{W}_2 \times \mathbf{W}_1}{\left|\mathbf{W}_2 \times \mathbf{W}_1\right|}$$

where $\mathbf{W}_1$ can be calculated with $\mathbf{W}_1 = \begin{bmatrix} 0 & -\sin i_r & \cos i_r \end{bmatrix}^T$ and $\mathbf{W}_2$ is determined from $\mathbf{W}_2 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$.

The relative inclination between the initial and final orbit planes is determined from

$$i_r = \cos^{-1}\left(\mathbf{w}_1 \bullet \mathbf{w}_2\right)$$

where $\mathbf{w}_1$ is the ECI unit angular momentum vector of the initial orbit given by

$$\mathbf{w}_1 = \begin{bmatrix} \sin\Omega_1 \sin i_1 \\ -\cos\Omega_1 \sin i_1 \\ \cos i_1 \end{bmatrix}$$

and $\mathbf{w}_2$ is the ECI unit angular momentum vector of the final orbit given by

$$\mathbf{w}_2 = \begin{bmatrix} \sin\Omega_2 \sin i_2 \\ -\cos\Omega_2 \sin i_2 \\ \cos i_2 \end{bmatrix}$$

The unit position vector at the first impulse in the reference coordinate system is

$$\mathbf{U}_1 = \begin{bmatrix} \cos\phi_1 \\ \sin\phi_1 \cos i_r \\ \sin\phi_1 \sin i_r \end{bmatrix}$$

and the unit position vector of the second impulse, also in the reference coordinate system, is determined from

$$\mathbf{U}_2 = \begin{bmatrix} \cos\phi_2 \\ \sin\phi_2 \\ 0 \end{bmatrix}$$

The transfer angle can be computed from the following dot product:

$$\Delta\theta = \cos^{-1}\left(\mathbf{U}_1 \bullet \mathbf{U}_2\right)$$

The minimum and maximum bounds on the semiparameter of the transfer orbit can be determined from the following two expressions:

$$p_{\min} = \frac{r_1 r_2 - \mathbf{r}_1 \bullet \mathbf{r}_2}{r_1 + r_2 + \sqrt{2\left(r_1 r_2 + \mathbf{r}_1 \bullet \mathbf{r}_2\right)}} \qquad p_{\max} = \frac{r_1 r_2 - \mathbf{r}_1 \bullet \mathbf{r}_2}{r_1 + r_2 - \sqrt{2\left(r_1 r_2 + \mathbf{r}_1 \bullet \mathbf{r}_2\right)}}$$

The partial derivative of the total required ΔV with respect to the semiparameter of the transfer orbit is as follows:

$$\frac{\partial V_t}{\partial p_t} = \frac{1}{2 p_t}\left(\frac{\Delta\mathbf{V}_1 \bullet \left(\mathbf{V} - z\mathbf{U}_1\right)}{\left|\Delta\mathbf{V}_1\right|} - \frac{\Delta\mathbf{V}_2 \bullet \left(\mathbf{V} + z\mathbf{U}_2\right)}{\left|\Delta\mathbf{V}_2\right|}\right)$$

Part of the optimal orbital transfer solution involves finding the value of p_t which lies between $p_{\min}$ and $p_{\max}$ and makes this partial derivative expression equal to zero.

The $\Delta\mathbf{V}$ vectors in the reference coordinate system are given by the following two expressions

$$\Delta\mathbf{V}_1 = \pm\left(\mathbf{V} + z\mathbf{U}_1\right) - \mathbf{V}_1$$

$$\Delta\mathbf{V}_2 = \mathbf{V}_2 \mp \left(\mathbf{V} - z\mathbf{U}_2\right)$$

where the upper sign in these two equations corresponds to the short transfer and

$$z = \sqrt{\frac{\mu}{p}}\tan\frac{\Delta\theta}{2}$$

with

$$\mathbf{V} = \sqrt{\mu p_t}\,\frac{\left(\mathbf{r}_2 - \mathbf{r}_1\right)}{\left|\mathbf{r}_1 \times \mathbf{r}_2\right|}$$

The velocity vector of the satellite prior to the first impulse with respect to the reference coordinate system is calculated from

$$\mathbf{V}_1 = \sqrt{\frac{\mu}{p_1}}\mathbf{W}_1 \times \left(\mathbf{e}_1 + \mathbf{U}_1\right)$$

and prior to the second impulse it is given by:

$$\mathbf{V}_2 = \sqrt{\frac{\mu}{p_2}}\mathbf{W}_2 \times \left(\mathbf{e}_2 + \mathbf{U}_2\right)$$

In these expressions $\mathbf{e}_1$ is the reference coordinate system eccentricity vector of the initial orbit is given by

$$\mathbf{e}_1 = e_1\begin{bmatrix}\cos\omega_1 \\ \sin\omega_1\cos i_r \\ \sin\omega_1\sin i_r\end{bmatrix}$$

and $\mathbf{e}_2$ is the eccentricity of the final orbit defined by

$$\mathbf{e}_2 = e_2\begin{bmatrix}\cos\omega_2 \\ \sin\omega_2 \\ 0\end{bmatrix}$$

where e_1 and e_2 are the scalar eccentricity of the initial and final orbits, respectively.

The total delta-v required for the orbit transfer is given by

$$\Delta V = \left|\Delta \mathbf{V}_1\right| + \left|\Delta \mathbf{V}_2\right|$$

In terms of the scalar components of the two $\Delta V's$, the total ΔV required is

$$\Delta V = \sqrt{\Delta V_{1x}^2 + \Delta V_{1y}^2 + \Delta V_{1x}^2} + \sqrt{\Delta V_{2x}^2 + \Delta V_{2y}^2 + \Delta V_{2x}^2}$$

This is the scalar quantity we want to minimize.

Eventually, we want to convert the reference coordinate system solution to ECI vectors and then to classical orbital elements. The transformation of an ECI position or velocity vector $\mathbf{X}_{eci}$ to its corresponding reference coordinate system companion $\mathbf{X}_{rcs}$ is given by the following matrix-vector multiplication:

$$\mathbf{X}_{eci} = \left[\mathbf{T}\right]\mathbf{X}_{rcs}$$

The conversion of a vector in the reference coordinate system to its corresponding ECI vector involves the transpose of this matrix as follows:

$$\mathbf{X}_{rcs} = \left[\mathbf{T}\right]\mathbf{X}_{eci}$$

The elements of the reference coordinate system-to-ECI transformation matrix $\left[\mathbf{T}\right]$ are given by the following nine expressions:

$$\begin{aligned}
T_{11} &= \cos\Omega_2 \cos\phi - \sin\Omega_2 \cos i_2 \sin\phi \\
T_{12} &= -\cos\Omega_2 \sin\phi - \sin\Omega_2 \cos i_2 \cos\phi \\
T_{13} &= \sin\Omega_2 \cos i_2 \\
T_{21} &= \sin\Omega_2 \cos\phi + \cos\Omega_2 \cos i_2 \sin\phi \\
T_{22} &= -\sin\Omega_2 \sin\phi + \cos\Omega_2 \cos i_2 \cos\phi \\
T_{23} &= -\cos\Omega_2 \sin i_2 \\
T_{31} &= \sin i_2 \sin\phi \\
T_{32} &= \sin i_2 \cos\phi \\
T_{33} &= \cos i_2
\end{aligned}$$

where

$$\phi = -\cos^{-1}\left(\mathbf{N} \bullet \mathbf{U}\right)\operatorname{sign}\left(N_z\right)$$

and

$$\mathbf{U} = \begin{bmatrix} \cos\Omega_2 \\ \sin\Omega_2 \\ 0 \end{bmatrix}$$

The position vector of the initial and transfer orbits at the first impulse in the reference coordinate system is

$$\mathbf{r}_1 = \left(\frac{p_1}{1 + e_1 \cos\left(\phi_1 - \omega_1\right)} \right) \mathbf{U}_1$$

and the position vector of the transfer and final orbit at the second impulse is

$$\mathbf{r}_2 = \left(\frac{p_2}{1 + e_2 \cos\left(\phi_2 - \omega_2\right)} \right) \mathbf{U}_2$$

In these equations the arguments of perigee ω_1 and ω_2 are with respect to the reference coordinate system. They can be determined with the following three equations:

$$\omega_{rcs} = \left[\mathbf{T}\right]^T \omega_{eci}$$

$$\omega_1 = \cos^{-1}\left(\omega_x\right)$$

$$\omega_2 = \tan^{-1}\left(\omega_y, \omega_z\right)$$

where the inverse tangent calculation here is a four-quadrant operation.

The ECI argument of perigee vectors at each impulse are given by

$$\omega_{eci_1} = \begin{bmatrix} \cos\omega_1 \cos\Omega_1 - \sin\omega_1 \sin\Omega_1 \cos i_1 \\ \cos\omega_1 \sin\Omega_1 + \sin\omega_1 \cos\Omega_1 \cos i_1 \\ \sin\omega_1 \sin i_1 \end{bmatrix}$$

and

$$\omega_{eci_2} = \begin{bmatrix} \cos\omega_2 \cos\Omega_2 - \sin\omega_2 \sin\Omega_2 \cos i_2 \\ \cos\omega_2 \sin\Omega_2 + \sin\omega_2 \cos\Omega_2 \cos i_2 \\ \sin\omega_2 \sin i_2 \end{bmatrix}$$

where all the orbital elements in these two equations are with respect to the ECI coordinate system.

The semiparameter of the initial orbit can be determined from

$$p_1 = a_1\left(1 - e_1^2\right)$$

and the semiparameter of the final orbit is given by

$$p_2 = a_2\left(1 - e_2^2\right)$$

where a_1 and a_2 are the semimajor axes of the initial and final orbits, respectively.

The transfer orbit velocity vectors prior to the first and second impulses in the reference coordinate system are calculated from the next two equations:

$$\mathbf{V}_{T_1} = \mathbf{V} + z\mathbf{U}_1$$

$$\mathbf{V}_{T_2} = \mathbf{V} - z\mathbf{U}_2$$

The transfer orbit position and velocity vectors can be transformed into the ECI coordinate system using the transpose of the $[\mathbf{T}]$ matrix as described above, and then converted to classical orbital elements.

The following is the MATLAB source code that performs the two-dimensional grid search and minimization.

```
% number of control variables

ncv = 2;

xg = zeros(ncv, 1);

xlwr = zeros(ncv, 1);

xupr = zeros(ncv, 1);

% number of mission constraints

nmc = 0;

flow = zeros(nmc, 1);
fupp = zeros(nmc, 1);

% bounds on objective function

flow(1) = 0.0e0;

fupp(1) = +Inf;

% solve the orbital TPBVP using SNOPT

xmul = zeros(ncv, 1);

xstate = zeros(ncv, 1);

fmul = zeros(nmc, 1);

fstate = zeros(nmc, 1);

tol = 1.0e-4;

for i = 1:1:nphi1
```

```
    for j = 1:1:nphi2

        % current initial guesses and bounds

        xg(1) = phi01 + dphi1 * (i - 1);

        xg(2) = phi02 + dphi2 * (j - 1);

        xlwr(1) = xg(1) - 2.0 * pi;

        xupr(2) = xg(2) + 2.0 * pi;

        % save current values

        xs1 = xg(1);

        xs2 = xg(2);

        dx = abs(xg(2) - xg(1));

        if ((abs(dx - pi) > tol) && (dx > tol) && (abs(dx - 2.0 * pi) > tol))

            % perform 2-dimensional minimization

            [xmin, ~, inform, xmul, fmul] = snopt(xg, xlwr, xupr, xmul, xstate,
                flow, fupp, fmul, fstate, 'oota_func1');

            % evaluate the current solution

            f = oota_func1(xmin);

            % check for global minimum

            if (f < gdvmin)

                % save current solution as the global minimum

                fs = f;

                gdvmin = fs;

                xsav(1) = xmin(1);

                xsav(2) = xmin(2);

            end

        end

    end

end
```

The 'oota_func1' computes the current value of the objective function which for this problem is the sum of the scalar values of the two impulsive maneuvers.

Primer Vector Analysis

This section summarizes the primer vector analysis included with this MATLAB script. The term primer vector was invented by Derek F. Lawden and represents the adjoint vector for velocity. A technical discussion about primer theory can be found in Lawden's classic text, *Optimal Trajectories for Space Navigation*, Butterworths, London, 1963. Another excellent resource is "Primer Vector Theory and Applications", Donald J. Jezewski, NASA TR R-454, November 1975, along with "Optimal, Multi-burn, Space Trajectories", also by Jezewski.

As shown by Lawden, the following four necessary conditions must be satisfied in order for an impulsive orbital transfer to be *locally optimal*:

(1) the primer vector and its first derivative are everywhere continuous

(2) whenever a velocity impulse occurs, the primer is a unit vector aligned with the impulse and has unit magnitude $\left(\mathbf{p} = \hat{\mathbf{p}} = \hat{\mathbf{u}}_T \text{ and } \|\mathbf{p}\| = 1\right)$

(3) the magnitude of the primer vector may not exceed unity on a coasting arc $\left(\|\mathbf{p}\| = p \le 1\right)$

(4) at all interior impulses (not at the initial or final times) $\mathbf{p} \cdot \dot{\mathbf{p}} = 0$; therefore, $d\|\mathbf{p}\|/dt = 0$ at the intermediate impulses

Furthermore, the scalar magnitudes of the primer vector derivative at the initial and final impulses provide information about how to improve the nominal transfer trajectory by changing the endpoint times and/or moving the impulse times. These four cases for non-zero slopes are summarized as follows;

- If $\dot{p}_0 > 0$ and $\dot{p}_f < 0 \rightarrow$ perform an initial coast before the first impulse and add a final coast after the second impulse
- If $\dot{p}_0 > 0$ and $\dot{p}_f > 0 \rightarrow$ perform an initial coast before the first impulse and move the second impulse to a later time
- If $\dot{p}_0 < 0$ and $\dot{p}_f < 0 \rightarrow$ perform the first impulse at an earlier time and add a final coast after the second impulse
- If $\dot{p}_0 < 0$ and $\dot{p}_f > 0 \rightarrow$ perform the first impulse at an earlier time and move the second impulse to a later time

The primer vector analysis of a two impulse orbital transfer involves the following steps.

First partition the two-body state transition matrix as follows:

$$\Phi\left(t,t_0\right)=\begin{bmatrix}\dfrac{\partial\mathbf{r}}{\partial\mathbf{r}_0} & \dfrac{\partial\mathbf{r}}{\partial\mathbf{v}_0}\\ \dfrac{\partial\mathbf{v}}{\partial\mathbf{r}_0} & \dfrac{\partial\mathbf{v}}{\partial\mathbf{v}_0}\end{bmatrix}=\begin{bmatrix}\Phi_{11} & \Phi_{12}\\ \Phi_{21} & \Phi_{22}\end{bmatrix}=\begin{bmatrix}\Phi_{rr} & \Phi_{rv}\\ \Phi_{vr} & \Phi_{vv}\end{bmatrix}$$

where

$$\Phi_{11}=\left[\frac{\partial\mathbf{r}}{\partial\mathbf{r}_0}\right]=\begin{bmatrix}\partial x/\partial x_0 & \partial x/\partial y_0 & \partial x/\partial z_0\\ \partial y/\partial x_0 & \partial y/\partial y_0 & \partial y/\partial z_0\\ \partial z/\partial x_0 & \partial z/\partial y_0 & \partial z/\partial z_0\end{bmatrix}$$

and so forth.

The value of the primer vector at any time *t* along a two-body trajectory is given by

$$\mathbf{p}\left(t\right)=\Phi_{11}\left(t,t_0\right)\mathbf{p}_0+\Phi_{12}\left(t,t_0\right)\dot{\mathbf{p}}_0$$

and the value of the primer vector derivative is

$$\dot{\mathbf{p}}\left(t\right)=\Phi_{21}\left(t,t_0\right)\mathbf{p}_0+\Phi_{22}\left(t,t_0\right)\dot{\mathbf{p}}_0$$

which can also be expressed as

$$\begin{Bmatrix}\mathbf{p}\\ \dot{\mathbf{p}}\end{Bmatrix}=\Phi\left(t,t_0\right)\begin{Bmatrix}\mathbf{p}_0\\ \dot{\mathbf{p}}_0\end{Bmatrix}$$

The primer vector boundary conditions at the initial and final impulses are as follows:

$$\mathbf{p}\left(t_0\right)=\mathbf{p}_0=\frac{\Delta\mathbf{V}_0}{\left|\Delta\mathbf{V}_0\right|}\qquad\qquad\mathbf{p}\left(t_f\right)=\mathbf{p}_f=\frac{\Delta\mathbf{V}_f}{\left|\Delta\mathbf{V}_f\right|}$$

These two conditions illustrate at the locations of velocity impulses, the primer vector is a unit vector in the direction of the impulses.

The value of the primer vector derivative at the initial time is

$$\dot{\mathbf{p}}\left(t_0\right)=\dot{\mathbf{p}}_0=\Phi_{12}^{-1}\left(t_f,t_0\right)\left\{\mathbf{p}_f-\Phi_{11}\left(t_f,t_0\right)\mathbf{p}_0\right\}$$

provided the Φ_{12} sub-matrix is non-singular.

Finally, the scalar magnitude of the derivative of the primer vector can be determined from

$$\frac{d\left\|\mathbf{p}\right\|}{dt}=\frac{d}{dt}\left(\mathbf{p}\cdot\mathbf{p}\right)^2=\frac{\dot{\mathbf{p}}\cdot\mathbf{p}}{\left\|\mathbf{p}\right\|}$$

The state transition matrix implemented in this MATLAB script is based on the algorithm described in "*Universal Keplerian state transition matrix*" by Stanley W. Shepperd, *Celestial Mechanics*, volume 35, February 1985, pages 129-144.

The calling syntax for this numerical method is as follows:

```
function [rf, vf, stm] = stm2(mu, tau, ri, vi)

% two body state transition matrix

% Shepperd's method

% input

%  tau = propagation time interval (seconds)
%  ri  = initial eci position vector (kilometers)
%  vi  = initial eci velocity vector (kilometers/second)

% output

%  rf  = final eci position vector (kilometers)
%  vf  = final eci velocity vector (kilometers/second)
%  stm = state transition matrix
```