Impulsive Hyperbolic Injection from a Circular Park Orbit – NLP Method

This document is the user's guide for a MATLAB script named `hyper2.m` which can be used to determine the characteristics of the single impulsive maneuver required to transfer a spacecraft from an initial circular park orbit to a departure hyperbola. The algorithm implemented in this script is based on the equations derived in Chapter 4 of Richard Battin's classic text, *Astronautical Guidance*, and Chapter 11 of *An Introduction to the Mathematics and Methods of Astrodynamics*, also written by Professor Battin and published by the American Institute of Aeronautics and Astronautics (AIAA).

The Earth departure trajectory for interplanetary missions is usually defined by a "targeting specification" which consists of twice the specific (per unit mass) orbital energy C_3, and the right ascension α_∞ (RLA) and declination δ_∞ (DLA) of the outgoing asymptote. These targets may be supplied by a spacecraft customer or determined with a patched-conic or more sophisticated trajectory analysis computer program that solves Lambert's problem for an interplanetary mission.

The `hyper2.m` MATLAB script determines the orbital elements and state vectors of the park orbit and departure hyperbola at injection, and the injection delta-v vector and magnitude. This information can be used as initial guesses for other trajectory simulations. This script uses the SNOPT nonlinear programming (NLP) method to find the optimal coplanar or non-coplanar transfer maneuver.

This computer program assumes that the hyperbolic targets and orbital characteristics are in the same Earth-centered-inertial (ECI) coordinate system. For example, targeting specs are often provided or computed in an Earth mean equator and equinox of J2000 coordinate system (EME2000). For this situation, the state vectors and orbital elements computed by this MATLAB script will also be with respect to the EME2000 coordinate system.

Running the Script

The script will interactively prompt the user for the park orbit altitude and orbital inclination, and the departure hyperbola characteristics. These prompts appear as follows;

```
please input the altitude of the circular park orbit (kilometers)
? 185.2

please input the orbital inclination of the park orbit (degrees)
(0 <= inclination <= 180)
? 28.5

please input the C3 of the departure hyperbola (km^2/sec^2)
(C3 > 0)
? 9.28

please input the right ascension of the outgoing asymptote (degrees)
(0 <= right ascension <= 360)
? 352.59

please input the declination of the outgoing asymptote (degrees)
? 38.5
```

Please note the proper units and valid data range for each input.

Script Output

The following is the script output for this non-coplanar example.

```
----------------------------------------------------------------
Interplanetary Injection from a Circular Park Orbit - NLP Method
----------------------------------------------------------------

departure hyperbola characteristics
-----------------------------------

c3                               9.280000  km^2/sec^2

asymptote right ascension      352.590000  degrees

asymptote declination           38.500000  degrees

orbital elements and state vector of park orbit at injection
--------------------------------------------------------------

        sma (km)              eccentricity          inclination (deg)         argper (deg)
 +6.56346000000000e+003  +0.00000000000000e+000  +2.85000000000000e+001  +0.00000000000000e+000

       raan (deg)          true anomaly (deg)         arglat (deg)            period (min)
 +2.62590000391829e+002  +3.12544908830941e+002  +3.12544908830941e+002  +8.81980522880484e+001

        rx (km)                 ry (km)                 rz (km)                rmag (km)
 -4.78649275664045e+003  -3.85286985447614e+003  -2.30735522767897e+003  +6.56346000000000e+003

        vx (kps)                vy (kps)                vz (kps)               vmag (kps)
 +3.85165475038245e+000  -6.29072335152133e+000  +2.51431627455823e+000  +7.79296034444086e+000

orbital elements and state vector of hyperbola at injection
-----------------------------------------------------------

        sma (km)              eccentricity          inclination (deg)         argper (deg)
 -4.29526337823274e+004  +1.15042824853112e+000  +3.97389482427168e+001  +3.12784284153612e+002

       raan (deg)          true anomaly (deg)         arglat (deg)
 +2.45683986407545e+002  +1.38556047267724e+001  +3.26639888880384e+002

        rx (km)                 ry (km)                 rz (km)                rmag (km)
 -4.78649275664045e+003  -3.85286985447614e+003  -2.30735522767897e+003  +6.56346000000000e+003

        vx (kps)                vy (kps)                vz (kps)               vmag (kps)
 +2.99280761783281e+000  -9.54679394718429e+000  +5.53554782769536e+000  +1.14341795446834e+001

injection delta-v vector and magnitude
--------------------------------------

x-component of delta-v       -858.847133  meters/second
y-component of delta-v      -3256.070596  meters/second
z-component of delta-v       3021.231553  meters/second

delta-v magnitude            4524.097061  meters/second
```

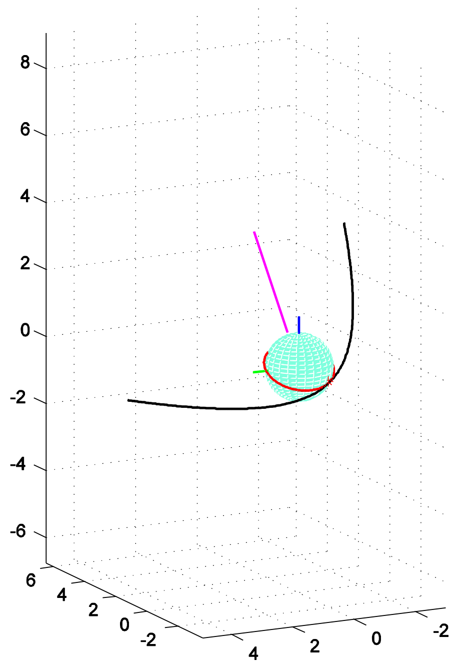
Trajectory Graphics

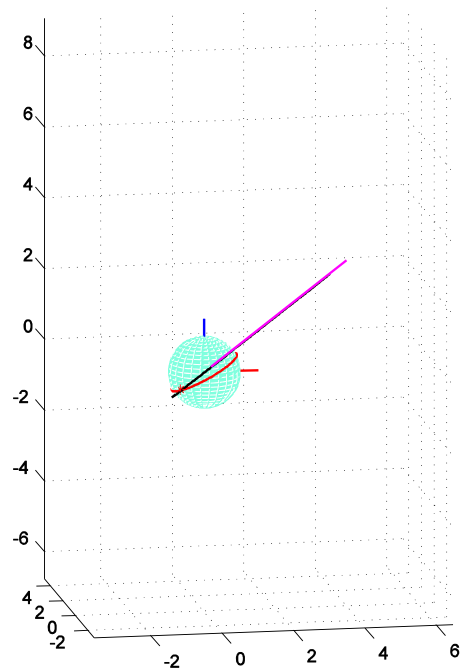
The `hyper2.m` MATLAB script will also create a graphics display of the park orbit and departure hyperbola for the injection opportunity. The interactive graphic features of MATLAB permit the user to rotate and zoom this display. These capabilities allow the user to interactively find the best viewpoint as well as verify basic orbital geometry of the park orbit and departure trajectory.

This script will also create a disk file of this graphics display called `hyper2.eps`. This file is a color Postscript image with a TIFF preview. The name and other characteristics of this file can be edited in the following line of MATLAB source code;

```
print -depsc -tiff -r300 hyper2.eps
```

The following are two typical displays for this example. Each display is labeled with an Earth centered, inertial coordinate system. The x-axis of this system is red, the y-axis is green and the z-axis is blue. The outgoing asymptote is colored magenta, the park orbit traces are red, and the hyperbolic trajectories are black. Please note the units for each coordinate axis are Earth radii (ER).





Technical Discussion

This section describes the numerical algorithms implemented in this MATLAB script. The script assumes an impulsive injection maneuver which implies an instantaneous change in velocity only. In the following paragraph, *i* is the orbital inclination of the initial circular Earth park orbit and δ_∞ is the declination of the outgoing or launch hyperbola.

Whenever $i > |\delta_\infty|$, there will be two coplanar opportunities to establish a departure hyperbola that will satisfy the energy and orientation of the outgoing asymptote. One injection opportunity will occur while the spacecraft is ascending and the other while the spacecraft is descending along the park orbit. For the case where $i = |\delta_\infty|$, there will be a single coplanar injection opportunity. Finally, for the case where $|\delta_\infty| > i$, there will be a single *non-coplanar* injection opportunity.

Please note that the `hyper2` script will solve for a single injection opportunity. Also note that for $i \geq |\delta_\infty|$, the impulsive maneuver occurs at perigee of the departure hyperbola.

A geocentric unit vector is the direction of the departure asymptote is given by

$$\hat{\mathbf{i}}_\infty = \begin{Bmatrix} \cos\delta_\infty \cos\alpha_\infty \\ \cos\delta_\infty \sin\alpha_\infty \\ \sin\delta_\infty \end{Bmatrix}$$

where

$\alpha_\infty =$ right ascension of departure asymptote

$\delta_\infty =$ declination of departure asymptote

Departure delta-V

The velocity vector of the spacecraft on the initial circular orbit is given by

$$\mathbf{v}_0 = \sqrt{\frac{\mu}{r}}\,\hat{\mathbf{i}}_\theta$$

The velocity vector at any geocentric position vector $\mathbf{r}$ required to achieve a launch hyperbola defined by v_∞, α_∞ and δ_∞ is given by

$$\mathbf{v}_1 = \frac{1}{2} v_\infty \left[(D+1)\hat{\mathbf{i}}_\infty + (D-1)\hat{\mathbf{i}}_r \right]$$

where

$$D = \sqrt{1 + \frac{4\mu}{r v_\infty^2 \left(1 + \hat{\mathbf{i}}_\infty \bullet \hat{\mathbf{i}}_r\right)}}$$

and

$$\hat{\mathbf{i}}_r = \begin{bmatrix} \cos\Omega\cos\theta - \sin\Omega\sin\theta\cos i \\ \sin\Omega\cos\theta + \cos\Omega\sin\theta\cos i \\ \sin\theta\sin i \end{bmatrix}$$

$$\hat{\mathbf{i}}_\theta = \begin{bmatrix} -\cos\Omega\sin\theta - \sin\Omega\cos\theta\cos i \\ -\sin\Omega\sin\theta + \cos\Omega\cos\theta\cos i \\ \cos\theta\sin i \end{bmatrix}$$

In these equations, Ω is the right ascension of the ascending node, i is the orbital inclination, θ is the true anomaly at injection, r is the geocentric radius of the park orbit and $v_\infty = \sqrt{C_3}$.

The injection $\Delta\mathbf{v}$ vector can be determined from the following expression

$$\Delta\mathbf{v} = \mathbf{v}_1 - \mathbf{v}_0$$

Finally, the scalar injection delta-v is $\Delta v = |\Delta\mathbf{v}|$ which is the value that this script attempts to minimize.

Orientation of the park orbit and departure hyperbola

The orientation of the park orbit and departure hyperbola at injection is computed using a two-dimensional grid search involving the park orbit right ascension of the ascending node (RAAN) and the true anomaly of the impulsive maneuver on the park orbit.

The following is the MATLAB source code that performs this grid search. In this code snippet, `nphi1` and `nphi2` are the number of RAAN and true anomaly increments. Also, `dphi1` and `dphi2` are the RAAN and true anomaly increments in radians, and `phi01` and `phi02` are the initial values for the RAAN and park orbit true anomaly grid search.

```
% perform grid search

for i = 1:1:nphi1

    for j = 1:1:nphi2

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

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

        if (firstpass == 1)

            xg = xg';

            % bounds on control variables

            xlwr(1) = - 2.0 * pi;
            xupr(1) = + 2.0 * pi;

            xlwr(2) = - 2.0 * pi;
            xupr(2) = + 2.0 * pi;

            xlwr = xlwr';
            xupr = xupr';

            % bounds on objective function

            flow(1) = 0.0d0;

            fupp(1) = +Inf;

            firstpass = 0;
        end

        % compute current optimal solution

        [x, f, inform, xmul, fmul] = snopt(xg, xlwr, xupr, flow, fupp, 'hyperfunc');

        % check for global minimum

        if (f < gdvmin)

            % save current solution as the global minimum

            fs = f;

            gdvmin = fs;

            xsav(1) = x(1);

            xsav(2) = x(2);
```

```
        end
    end
end
% evaluate global solution
[f, g] = hyperfunc (xsav);
```

This code illustrates how the initial guess vector `xg` is calculated, the setting of the lower and upper bounds on the control variables and objective function, and the actual call to the SNOPT function. Note that `'hyperfunc'` is the name of the MATLAB function that computes the objective function.

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.

Algorithm Resources

"Design of Lunar and Interplanetary Ascent Trajectories", Victor C. Clarke, Jr., JPL Technical Report No. 32-30, March 15, 1962.

An Introduction to the Mathematics and Methods of Astrodynamics, Richard H. Battin, AIAA Education Series, 1987.

Analytical Mechanics of Space Systems, Hanspeter Schaub and John L. Junkins, AIAA Education Series, 2003.

Spacecraft Mission Design, Charles D. Brown, AIAA Education Series, 1992.

Orbital Mechanics, Vladimir A. Chobotov, AIAA Education Series, 2002.

"A Computer Simulation of the Orbital Launch Window Problem", Archie C. Young and Pat R. Odom, AIAA 67-615, 1967.

"Launch Parameters for Interplanetary Flights", W. C. Riddell, *American Rocket Society Journal*, December 1960.