A MATLAB Script for Optimal Single Impulse De-orbit from Earth Orbits

This document describes a MATLAB script named `deorbit_snopt` that can be used to compute the optimal impulsive maneuver required to de-orbit a spacecraft in a circular or elliptical Earth orbit. The user provides the classical orbital elements of the initial orbit along with geodetic altitude and relative flight path angle *targets* at the entry interface (EI).

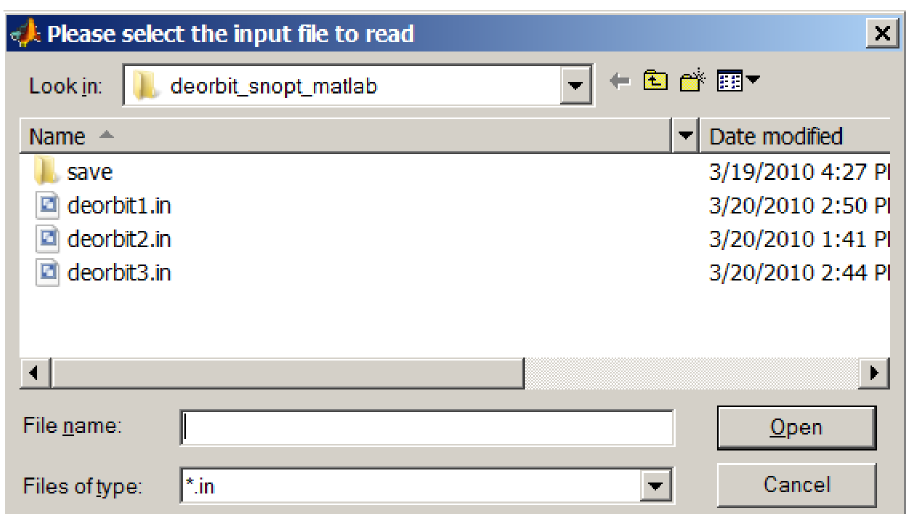
This script solves this maneuver optimization problem using a *simple shooting* method. During the solution process, the script numerically integrates the spacecraft equations of motion subject to the Earth's J_2 gravity coefficient. The numerical integration is performed using MATLAB's `ode45` function. The entry interface targets are computed with respect to an oblate, rotating Earth.

In this classic maneuver optimization problem, the maneuver true anomaly, the ECI components of the maneuver delta-v vector and the flight time from the maneuver to the entry interface are the *control variables*. The scalar magnitude of the de-orbit ΔV is the *objective function* or *performance index*, and the geodetic altitude and relative flight path angle at the entry interface are treated as *nonlinear equality constraints*. The algorithm uses an initial guess determined from the analytic de-orbit solution relative to a spherical, non-rotating Earth.

The `deorbit_snopt` script uses the SNOPT nonlinear programming algorithm to solve this orbital mechanics problem. MATLAB versions of SNOPT for several computer platforms can be requested or purchased 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 software user's guide.

Interacting with the script

This MATLAB script is "data driven" by a text file created by the user. When the `deorbit_snopt` script is started, the software will display the following screen which allows the user to select a data file for processing.



The file type defaults to names with a `*.in` filename extension. However, you can select any `deorbit_snopt` compatible ASCII data file. The next section describes the format and typical contents of compatible input files.

Input data file

This section describes a typical input data file for the software. In the following discussion the actual input file contents are in `courier` font and all explanations are in *times italic* font. Typical user-provided values are in bold font.

Each data item within an input file is preceded by one or more lines of *annotation* text. Do not delete any of these annotation lines or increase or decrease the number of lines reserved for each comment. However, you may change them to reflect your own explanation. The annotation line also includes the correct units and when appropriate, the valid range of the input.

The first five lines of any input file are reserved for user comments. These lines are ignored by the software. However the input file must begin with five and only five initial text lines.

```
*****************************************************
** impulsive de-orbit delta-v trajectory optimization
** de-orbit from initial circular orbit
** file ==> deorbit3.in   April 18, 2020
*****************************************************
```

The first input is the calendar date of the impulsive maneuver. Be sure to include all four digits of the calendar year.

```
calendar date at time of impulsive maneuver (month, day, year)
3, 18, 2010
```

The next input is the UTC time of the de-orbit maneuver.

```
UTC at time of impulsive maneuver (hours, minutes, seconds)
12, 30, 45.875
```

The next series of inputs define the classical orbital elements of the initial Earth orbit. Notice that the true anomaly is an initial guess for the location of the maneuver. The true anomaly initial guess for elliptical Earth orbits should be 180 degrees.

```
**********************************************
orbital elements at time of impulsive maneuver
**********************************************

semimajor axis (kilometers)7378.14
6878.14

orbital eccentricity (non-dimensional)
0.0

orbital inclination (degrees)
28.5

argument of perigee (degrees)
100.0

right ascension of the ascending node (degrees)
220.0

initial guess for true anomaly (degrees)
180.0
```

The software allows the user to specify lower and upper bounds for the optimal true anomaly of the maneuver. The algorithm enforces an inequality constraint on the true anomaly according to

$$\theta_L \le \theta \le \theta_U$$

where θ_L and θ_U *are the user-defined lower and upper bounds, respectively.*

The numerical values of these bounds are defined in the next two data items.

```
lower bound for true anomaly (degrees)
170.0

upper bound for true anomaly (degrees)
190.0
```

The final two items in the simulation file define the geodetic altitude and relative flight path angle targets at the entry interface.

```
***********************************
entry interface mission constraints
***********************************

geodetic altitude (kilometers)
121.92

relative flight path angle (degrees)
-2.0
```

Script examples

The following is the `deorbit_snopt` numerical solution for this example.

```
***************************************
single impulse deorbit from Earth orbits
***************************************

time and conditions prior to deorbit maneuver
--------------------------------------------

calendar date      18-Mar-2010

UTC time           12:30:45.875

        sma (km)              eccentricity          inclination (deg)         argper (deg)
 +6.87814000000000e+03  +0.00000000000000e+00  +2.85000000000000e+01  +1.00000000000000e+02

       raan (deg)          true anomaly (deg)         arglat (deg)            period (min)
 +2.20000000000000e+02  +1.90000000000000e+02  +2.90000000000000e+02  +9.46163624134673e+01

        rx (km)                 ry (km)                 rz (km)                 rmag (km)
 -5.45318321844679e+03  +2.83906883966968e+03  -3.08403806222734e+03  +6.87814000000000e+03

        vx (kps)                vy (kps)                vz (kps)                vmag (kps)
 -4.00911535387506e+00  -6.35100857948859e+00  +1.24236145363939e+00  +7.61260651018449e+00

deorbit delta-v vector and magnitude
-----------------------------------

x-component of delta-v         80.301516  meters/second
y-component of delta-v        117.688815  meters/second
z-component of delta-v        -21.001409  meters/second
total delta-v                 144.014061  meters/second
```

```
deorbit delta-v pointing angles
-------------------------------

pitch angle                    -2.256618  degrees

yaw angle                    -179.973071  degrees

time and conditions after deorbit maneuver
------------------------------------------

calendar date      18-Mar-2010

UTC time           12:30:45.875

        sma (km)              eccentricity          inclination (deg)         argper (deg)
 +6.62986196765162e+03  +3.74561317348360e-02  +2.84998225494152e+01  +1.08881122818562e+02

       raan (deg)         true anomaly (deg)         arglat (deg)             period (min)
 +2.20001021787282e+02  +1.81117979216534e+02  +2.89999102035096e+02  +8.95398701301721e+01

        rx (km)                 ry (km)                 rz (km)                 rmag (km)
 -5.45318321844679e+03  +2.83906883966968e+03  -3.08403806222734e+03  +6.87814000000000e+03

        vx (kps)                vy (kps)                vz (kps)                vmag (kps)
 -3.92881383778096e+00  -6.23331976439964e+00  +1.22136004444855e+00  +7.46870630131950e+00

time and conditions at entry interface
--------------------------------------

calendar date      18-Mar-2010

UTC time           13:00:49.638

        sma (km)              eccentricity          inclination (deg)         argper (deg)
 +6.63194303419306e+03  +3.87915541331080e-02  +2.85109989908368e+01  +1.07810489094078e+02

       raan (deg)         true anomaly (deg)         arglat (deg)             period (min)
 +2.19909866252810e+02  +2.99617139359573e+02  +4.74276284536510e+01  +8.95820323314157e+01

        rx (km)                 ry (km)                 rz (km)                 rmag (km)
 -5.45318321844679e+03  +2.83906883966968e+03  -3.08403806222734e+03  +6.87814000000000e+03

        vx (kps)                vy (kps)                vz (kps)                vmag (kps)
 +7.50958358577267e+00  +3.73745768234497e-01  +2.46143688225353e+00  +7.91152343460458e+00

relative flight path coordinates at entry interface
---------------------------------------------------

east longitude              252.44489639  degrees

geocentric declination       20.58001385  degrees

flight path angle            -2.00000000  degrees

relative azimuth             68.65207490  degrees

position magnitude         6497.40258326  kilometers

velocity magnitude            7.49698673  kilometers/second

geodetic coordinates at entry interface
---------------------------------------

geodetic latitude            20.70458617  degrees

geodetic altitude           121.92000000  kilometers


flight time from maneuver to EI     30.06271094  minutes
```

The following is a brief description of the information provided in the script output.

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

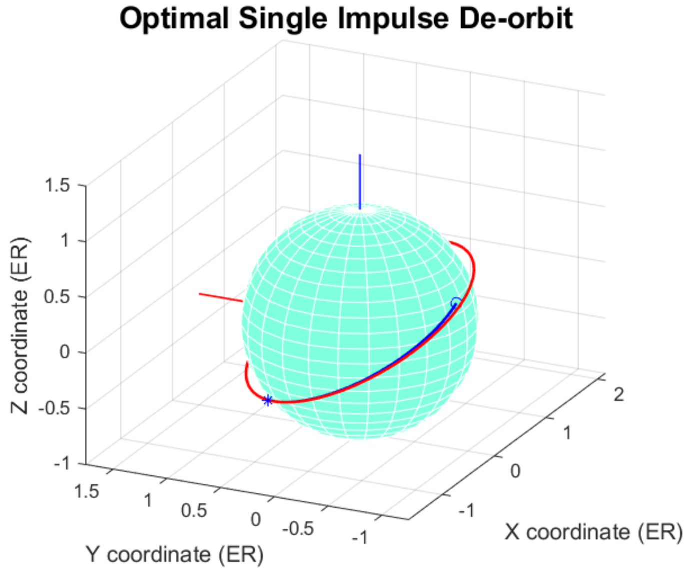
vz (ksp) = z-component of the velocity vector in kilometers per second

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

The components of the de-orbit delta-v vector are displayed in the ECI coordinate system. The relative flight path coordinates are with respect to a rotating Earth. The UTC time is given in hours, minutes and seconds.

The `deorbit_snopt` script will also create a three-dimensional graphics display of the initial orbit and re-entry trajectory. The following is the graphic image for this example. The initial orbit trace is red and the re-entry trajectory is blue. The dimensions are Earth radii (ER) and the plot is labeled with an Earth-centered-inertial (ECI) coordinate system where green is the x-axis, red is the y-axis and blue is the z-axis. The impulse location is marked with a blue asterisk and entry interface is marked with a small blue circle.

Trajectory image files are saved to disk in both `tif` format and MATLAB `fig` format with a file name indicating the solution number. The disk file names are `deorbit_snopt.tif` and `deorbit_snopt.fig`. The interactive features of MATLAB graphics allow the user to manipulate the `fig` version of the trajectory display. These capabilities allow the user to interactively find the best viewpoint as well as verify basic three-dimensional geometry of the orbital maneuver and entry.

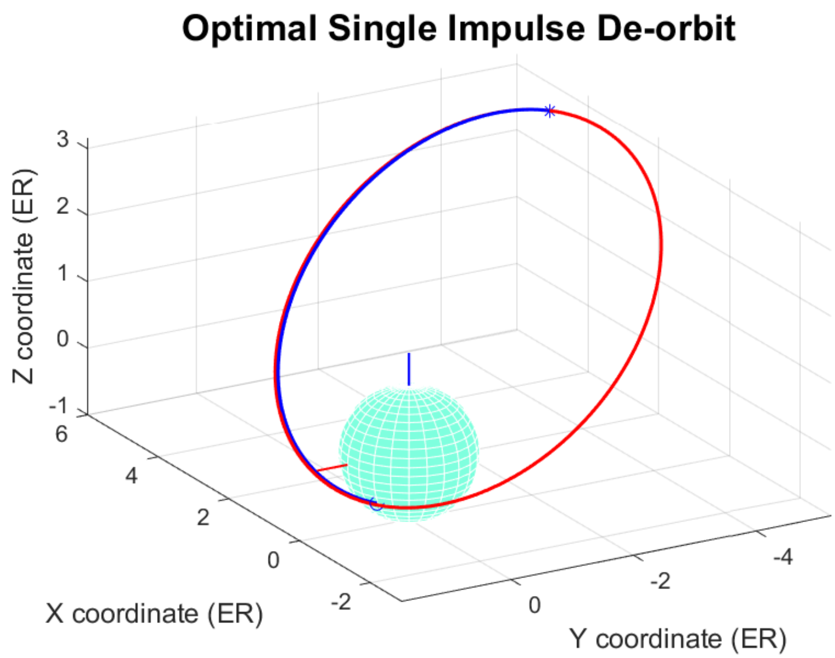


The following is the output created by this MATLAB script for the optimal de-orbit from a typical highly elliptical orbit (HEO).

```
****************************************
single impulse deorbit from Earth orbits
****************************************

time and conditions prior to deorbit maneuver
---------------------------------------------

calendar date      18-Mar-2010

UTC time           12:30:45.875

        sma (km)               eccentricity          inclination (deg)          argper (deg)
 +2.44140000000000e+04  +7.27044000000000e-01  +2.85000000000000e+01  +2.70000000000000e+02

       raan (deg)           true anomaly (deg)         arglat (deg)              period (min)
 +2.20000000000000e+02  +1.80010875055299e+02  +9.00108750552986e+01  +6.32729583646570e+02

        rx (km)                 ry (km)                  rz (km)                  rmag (km)
 +2.38242965164960e+04  -2.83802405542600e+04  +2.01189455552070e+04  +4.21640501929899e+04

        vx (kps)                vy (kps)                 vz (kps)                 vmag (kps)
 +1.22991535564376e+00  +1.03330298046443e+00  -5.32994999257885e-04  +1.60636456496297e+00

deorbit delta-v vector and magnitude
------------------------------------

x-component of delta-v        -16.099990  meters/second
y-component of delta-v        -13.510509  meters/second
z-component of delta-v          0.004894  meters/second

total delta-v                  21.017696  meters/second

deorbit delta-v pointing angles
-------------------------------
pitch angle                    -0.002649  degrees

yaw angle                     179.989285  degrees

time and conditions after deorbit maneuver
------------------------------------------

calendar date      18-Mar-2010
```

```
UTC time           12:30:45.875

        sma (km)              eccentricity          inclination (deg)         argper (deg)
 +2.43140994191278e+04  +7.34139993195612e-01  +2.84999999733610e+01  +2.69999971672997e+02

       raan (deg)          true anomaly (deg)         arglat (deg)           period (min)
 +2.20000297715537e+02  +1.80010641744793e+02  +9.00106134177898e+01  +6.28849923664669e+02

        rx (km)                 ry (km)                 rz (km)                rmag (km)
 +2.38242965164960e+04  -2.83802405542600e+04  +2.01189455552070e+04  +4.21640501929899e+04

        vx (kps)                vy (kps)                vz (kps)               vmag (kps)
 +1.21381536549559e+00  +1.01979247133070e+00  -5.28100846447572e-04  +1.58534687213445e+00

time and conditions at entry interface
--------------------------------------

calendar date      18-Mar-2010

UTC time           17:43:27.044

        sma (km)              eccentricity          inclination (deg)         argper (deg)
 +2.43457651117266e+04  +7.34618508714193e-01  +2.84881736895298e+01  +2.70116511603440e+02

       raan (deg)          true anomaly (deg)         arglat (deg)           period (min)
 +2.19931376435734e+02  +3.50932531329709e+02  +2.61049042933149e+02  +6.30078806344008e+02

        rx (km)                 ry (km)                 rz (km)                rmag (km)
 +2.38242965164960e+04  -2.83802405542600e+04  +2.01189455552070e+04  +4.21640501929899e+04

        vx (kps)                vy (kps)                vz (kps)               vmag (kps)
 -8.39531350465642e+00  -5.97405295708906e+00  -4.38335404240751e-01  +1.03132310893426e+01

relative flight path coordinates at entry interface
---------------------------------------------------

east longitude               37.72990551  degrees

geocentric declination      -28.11018407  degrees

flight path angle            -4.00000004  degrees

relative azimuth             95.03036771  degrees

position magnitude         6495.29077143  kilometers

velocity magnitude            9.89797352  kilometers/second

geodetic coordinates at entry interface
---------------------------------------

geodetic latitude           -28.26740378  degrees

geodetic altitude           121.92000146  kilometers


flight time from maneuver to EI    312.68615417  minutes
```

Here’s the trajectory graphics display for this example.



Technical discussion

This section is a brief explanantion of the algorithms implemented in this MATLAB script.

Nonlinear programming problem

A trajectory optimization problem can be described by a system of *dynamic variables*

$$\mathbf{z} = \begin{bmatrix} \mathbf{y}(t) \\ \mathbf{u}(t) \end{bmatrix}$$

consisting of the *state variables* $\mathbf{y}$ and the *control variables* $\mathbf{u}$ for any time *t*. In this discussion vectors are denoted in bold.

The system dynamics are defined by a vector system of ordinary differential equations called the *state equations* that can be represented as follows

$$\dot{\mathbf{y}} = \frac{d\mathbf{y}}{dt} = \mathbf{f}\left[\mathbf{y}(t), \mathbf{u}(t), \mathbf{p}, t\right]$$

where $\mathbf{p}$ is a vector of problem *parameters* that is not time dependent.

The initial dynamic variables at time t_0 are defined by $\boldsymbol{\psi}_0 \equiv \boldsymbol{\psi}\left[\mathbf{y}(t_0), \mathbf{u}(t_0), t_0\right]$ and the terminal conditions at the final time t_f are defined by $\boldsymbol{\psi}_f \equiv \boldsymbol{\psi}\left[\mathbf{y}(t_f), \mathbf{u}(t_f), t_f\right]$. These conditions are called the *boundary values* of the trajectory problem. The problem may also be subject to *path constraints* of the form $\mathbf{g}\left[\mathbf{y}(t), \mathbf{u}(t), t\right] = 0$.

The basic nonlinear programming problem (NLP) is to determine the control vector history and problem parameters that minimize the scalar performance index or objective function given by

$$J = \phi\left[\mathbf{y}(t_0), t_0, \mathbf{y}(t_f), t_f, \mathbf{p}\right]$$

while satisfying all the user-defined mission constraints.

In this classic maneuver optimization problem, the maneuver true anomaly, the ECI components of the maneuver delta-v vector and the flight time from the maneuver to the entry interface are the *control variables*. The scalar magnitude of the de-orbit ΔV is the *objective function*, and the geodetic altitude and relative flight path angle at the entry interface are the *nonlinear equality constraints*.

Initial guess

An initial guess for the scalar magnitude of the de-orbit delta-v and time-of-flight from the maneuver location to the entry interface is determined using analytic solutions for these values relative to a non-rotating, spherical Earth model and two-body or Keplerian motion. The analytic solution for circular orbits in discussed in Appendix B and Appendix C contains the equations for elliptical orbits. Please note the elliptical orbit analytic solution assumes the de-orbit maneuver occurs at apogee (true anomaly = 180°). This is the typical true anomaly initial guess for de-orbit from elliptical orbits.

The initial guess for the de-orbit delta-v vector is aligned opposite (retrograde) to the unit velocity vector on the initial Earth orbit at the maneuver location. This creates an impulsive velocity increment in the Earth-centered-inertial (ECI) Cartesian coordinate system.

Spacecraft equations of motion

During the solution process, the `deorbit_snopt` script numerically propagates the spacecraft trajectory from the maneuver time to the current estimate of the time at the entry interface. The system of six first-order differential equations subject to Earth gravity, defined in the ECI coordinate system (x, y, z), is given by the following expressions

$$\dot{y}_1 = v_x = y_4 \qquad \dot{y}_2 = v_y = y_5 \qquad \dot{y}_2 = v_z = y_6$$

$$\dot{y}_4 = -\mu \frac{r_x}{r^3}\left\{1 + \frac{3}{2}\frac{J_2 r_{eq}^2}{r^2}\left(1 - \frac{5r_z^2}{r^2}\right)\right\} \quad \dot{y}_5 = -\mu \frac{r_y}{r^3}\left\{1 + \frac{3}{2}\frac{J_2 r_{eq}^2}{r^2}\left(1 - \frac{5r_z^2}{r^2}\right)\right\} \quad \dot{y}_6 = -\mu \frac{r_z}{r^3}\left\{1 + \frac{3}{2}\frac{J_2 r_{eq}^2}{r^2}\left(3 - \frac{5r_z^2}{r^2}\right)\right\}$$

where $r = \sqrt{r_x^2 + r_y^2 + r_z^2} = \sqrt{y_1^2 + y_2^2 + y_3^2}$. In these equations μ and r_{eq} are the gravitational constant and equatorial radius of the Earth, and J_2 is the first order oblateness gravity coefficient.

At the entry interface, the algorithm computes the errors in the target constraints according to

$$\varepsilon_h = h_p - h_t$$
$$\varepsilon_\gamma = \gamma_p - \gamma_t$$

where h is the geodetic altitude and γ is the flight path angle relative to a rotating Earth. In these equations, the p subscript indicates the value *predicted* by the software, and the t subscript is the *target* value provided by the user. During the solution process, the SNOPT algorithm attempts to drive these two errors to zero.

The equations for calculating the relative flight path coordinates from an ECI position and velocity vectors is summarized in Appendix D. The algorithm used to calculate geodetic coordinates can be found in Appendix E and Appendix F discusses the coordinate system used to define the pitch and yaw orientation angles of the maneuver.

SNOPT algorithm implementation

This section provides details about the MATLAB source code that solve this nonlinear programming (NLP) problem using the SNOPT algorithm. MATLAB versions of SNOPT for several computer platforms can be found at Professor Philip Gill's University of California, San Diego 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.

The SNOPT algorithm requires an initial guess for the control variables. For this problem they are given by

```
xg(1) = oevpo(6);

xg(2) = dvg(1);

xg(3) = dvg(2);

xg(4) = dvg(3);

xg(5) = dtof;
```

where `xg(1)` is the user's initial guess for the true anomaly on the initial orbit at the time of the impulsive maneuver. `xg(2)`, `xg(2)` and `xg(3)` are the initial guesses for the ECI components of the de-orbit impulse, and `dtof` is the initial guess for the transfer time from the de-orbit maneuver to the entry interface.

The algorithm also requires lower and upper bounds for the control variables. These are determined from the initial guesses and user-defined true anomaly boundaries as follows:

```
% lower and upper bounds for deorbit true anomaly (radians)

xlwr(1) = ta_lower;

xupr(1) = ta_upper;

% lower and upper bounds for components of
% deorbit delta-v vector (kilometers/second)

dvm = norm(dvg);

xlwr(2:4) = -(dvm + 0.1 * dvm);

xupr(2:4) = +(dvm + 0.1 * dvm);

% lower and upper bounds for flight time
% from maneuver to entry interface (seconds)
```

```
xlwr(5) = dtof - 30.0;

xupr(5) = dtof + 30.0;
```

The algorithm also requires lower and upper bounds on the objective function. For this problem these bounds are given by

```
% bounds on objective function

flow(1) = 0.0;

fupp(1) = +Inf;
```

The following MATLAB code sets the lower and upper bounds for the two equality constraints (geodetic altitude and relative flight path angle) at the entry interface.

```
% geodetic altitude at entry interface equality constraint

flow(2) = 0.0;
fupp(2) = 0.0;

% relative flight path angle at entry interface equality constraint

flow(3) = 0.0;
fupp(3) = 0.0;
```

The actual call to the SNOPT MATLAB interface function is as follows

```
[x, f, inform, xmul, fmul] = snopt(xg, xlwr, xupr, xmul, xstate, ...
             flow, fupp, fmul, fstate, 'deorbit_shoot');
```

where `deorbit_shoot` is the name of the MATLAB function that integrates the spacecraft equations of motion and computes the current value of the objective function and the equality constraints at the entry interface. The solution for the control variables is returned in the `x` vector and `f` is the converged value of the objective function. Please consult the SNOPT documentation for additional information about the syntax of this function.

Algorithm Resources

"User's Guide for SNOPT Version 7, A Fortran Package for Large-Scale Nonlinear Programming", Philip E. Gill, Walter Murray and Michael A. Saunders, March 20, 2006.

"Optimum Deboost Altitude for Specified Atmospheric Entry Angle", Jerome M. Baker, Bruce E. Baxter, and Paul D. Arthur, *AIAA Journal*, Vol. 1, No. 7, July 1963.

"Deboost from Circular Orbits", A. H. Milstead, *The Journal of the Astronautical Sciences*, Vol. XIII, No. 4, pp. 170-171, Jul-Aug., 1966.

Hypersonic and Planetary Entry Flight Mechanics, Vinh, Busemann and Culp, The University of Michigan Press, 1980.

"On Autonomous Optimal Deorbit Guidance", Morgan C. Baldwin, Binfeng Pan and Ping Lu, AIAA 2009-5667, AIAA Guidance, Navigation, and Control Conference, August 10-13, 2009.

"Autonomous Optimal Deorbit Targeting", Donald J. Jezewski, AAS 91-136, AAS/AIAA Spaceflight Mechanics Meeting, February 11-13, 1991.

"Analysis of the Accuracy of Ballistic Descent from a Circular Circumterrestrial Orbit", Yu. G. Sikharulidze and A. N. Korchagin, *Cosmic Research*, Vol. 40, No. 1, 2002, pp.75-87.

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

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

APPENDIX A

Optimization Toolbox Implementation

There is a version of this MATLAB script named `deorbit_otb` that uses the Mathworks Optimization Toolbox to solve this orbital mechanics problem. This appendix describes the source code implementation using the `fmincon/interior-point` algorithm. Unlike SNOPT, this version requires the mission constraints and objective algorithms reside in two different MATLAB functions.

The following MATLAB source code solves the deorbit trajectory optimization problem.

```
% load initial guesses for control variables

xg(1) = oevpo(6);

xg(2) = dvg(1);

xg(3) = dvg(2);

xg(4) = dvg(3);

xg(5) = dtof;

% lower and upper bounds for deorbit true anomaly (radians)

xlwr(1) = ta_lower;

xupr(1) = ta_upper;

% lower and upper bounds for components of
% deorbit delta-v vector (kilometers/second)

dvm = norm(dvg);

xlwr(2:4) = -(dvm + 0.1 * dvm);

xupr(2:4) = +(dvm + 0.1 * dvm);

% lower and upper bounds for flight time
% from maneuver to entry interface (seconds)

xlwr(5) = dtof - 30.0;

xupr(5) = dtof + 30.0;

% solve trajectory optimization problem

options = optimoptions('fmincon', 'Display', 'iter', 'Algorithm', 'interior-point', ...
    'MaxFunctionEvaluations', 5000, 'FiniteDifferenceType', 'forward');

% optimize with user-defined mission constraints

[x, fval] = fmincon('deorbit_objective', xg, [], [], [], [], xlwr, xupr,
'deorbit_constraints', options);
```

The MATLAB function that evaluates the objective function is named `deorbit_objective` and `deorbit_constraints` calculates the current mission constraints.

Feel free to experiment with other `fmincon` non-linear programming algorithms such as *sqp*, etc.

APPENDIX B

De-orbit from a Circular Earth Orbit

The scalar magnitude of the single impulsive maneuver required to de-orbit a spacecraft from an initial circular orbit can be determined from the following expression

$$\Delta V = V_{c_e}\sqrt{\frac{1}{\tilde{r}}}\left\{1-\sqrt{\frac{2\left(\tilde{r}-1\right)}{\left(\dfrac{\tilde{r}}{\cos\gamma_e}\right)^2-1}}\right\} = V_{c_i}\left\{1-\sqrt{\frac{2\left(\tilde{r}-1\right)}{\left(\dfrac{\tilde{r}}{\cos\gamma_e}\right)^2-1}}\right\}$$

where

$$\tilde{r} = \frac{h_i + r_{eq}}{h_e + r_{eq}} = \frac{r_i}{r_e} = \text{radius ratio}$$

$$V_{c_e} = \sqrt{\frac{\mu}{\left(h_e + r_{eq}\right)}} = \sqrt{\frac{\mu}{r_e}} = \text{ local circular velocity at entry interface}$$

$$V_{c_i} = \sqrt{\frac{\mu}{\left(h_i + r_{eq}\right)}} = \sqrt{\frac{\mu}{r_i}} = \text{ local circular velocity of initial circular orbit}$$

$\gamma_e =$ flight path angle at entry interface

$h_i =$ altitude of initial circular orbit

$h_e =$ altitude at entry interface

$r_i =$ radius of initial circular orbit

$r_e =$ radius at entry interface

$r_{eq} =$ Earth equatorial radius

$\mu =$ Earth gravitational constant

This algorithm is described in the technical article, "Deboost from Circular Orbits", A. H. Milstead, *The Journal of the Astronautical Sciences*, Vol. XIII, No. 4, pp. 170-171, Jul-Aug., 1966. Additional information can be found in Chapter 5 of *Hypersonic and Planetary Entry Flight Mechanics* by Vinh, Busemann and Culp, The University of Michigan Press.

The true anomaly on the de-orbit trajectory at the entry interface θ_e can be determined from the following two equations

$$\sin\theta_e = \frac{\dot{r}}{e_d}\sqrt{\frac{a_d\left(1-e_d^2\right)}{\mu}} \qquad \cos\theta_e = \frac{a_d\left(1-e_d^2\right)}{e_d r_e} - \frac{1}{e_d}$$

and the following four quadrant inverse tangent operation

$$\theta_e = \tan^{-1}\left(\sin\theta_e, \cos\theta_e\right)$$

where

$e_d =$ eccentricity of the de-orbit trajectory

$a_d =$ semimajor axis of the de-orbit trajectory

$$\dot{r} = -\sqrt{\frac{\mu\left[2a_d r_e - r_e^2 - a_d^2\left(1-e_d^2\right)\right]}{a_d r_e^2}}$$

The elapsed time-of-flight between perigee of the de-orbit trajectory and the entry true anomaly θ_e is given by

$$t\left(\theta_e\right) = \frac{\tau}{2\pi}\left[2\tan^{-1}\left\{\sqrt{\frac{1-e_d}{1+e_d}}\tan\frac{\theta_e}{2}\right\} - \frac{e_d\sqrt{1-e_d^2}\sin\theta_e}{1+e_d\cos\theta_e}\right]$$

In this equation τ is the Keplerian orbital period of the de-orbit trajectory and is equal to $2\pi\sqrt{a_d^3/\mu}$.

Therefore, the flight time between the de-orbit impulse and entry interface is given by

$$\Delta t = t\left(\theta_e\right) - t\left(180^\circ\right) = t\left(\theta_e\right) - \frac{\tau}{2}$$

Finally, the orbital speed at the entry interface V_e can be determined from

$$V_e = \sqrt{\frac{2\mu}{r_e} - \frac{\mu}{a_d}}$$

APPENDIX C

De-orbit from an Elliptical Earth Orbit

The scalar magnitude of the impulsive delta-v for de-orbit from an initial elliptical orbit is given by

$$\Delta V = \sqrt{\frac{\mu}{r_e}}\left(\sqrt{\frac{2\tilde{r}_p}{\tilde{r}_a\left(\tilde{r}_a+\tilde{r}_p\right)}}-\sqrt{\frac{2\left(\tilde{r}_a-1\right)}{\tilde{r}_a\left(\tilde{r}_a^2-\cos^2\gamma_e\right)}}\cos\gamma_e\right)$$

where

$r_e =$ geocentric radius at the entry altitude

$\tilde{r}_a = r_a/r_e$

$\tilde{r}_p = r_p/r_e$

$\gamma_e =$ flight path angle at entry

$r_a =$ apogee radius of the initial elliptical orbit

$r_p =$ perigee radius of the initial elliptical orbit

$\mu =$ gravitational constant of the Earth

The true anomaly at entry can be determined from the following series of equations.

$$\sin\theta_e = \frac{\dot{r}}{e_d}\sqrt{\frac{a_d\left(1-e_d^2\right)}{\mu}} \qquad \cos\theta_e = \frac{a_d\left(1-e_d^2\right)}{e_d r_e}-\frac{1}{e_d} \qquad \theta_e = \tan^{-1}\left(\sin\theta_e,\cos\theta_e\right)$$

where

$e_d =$ eccentricity of the de-orbit trajectory

$a_d =$ semimajor axis of the de-orbit trajectory

$$\dot{r} = -\sqrt{\frac{\mu\left[2a_d r_e - r_e^2 - a_d^2\left(1-e_d^2\right)\right]}{a_d r_e^2}}$$

The time-of-flight between perigee and the entry true anomaly θ_e is given by

$$tof\left(\theta_e\right) = \frac{\tau}{2\pi}\left[2\tan^{-1}\left\{\sqrt{\frac{1-e_d}{1+e_d}}\tan\frac{\theta_e}{2}\right\}-\frac{e_d\sqrt{1-e_d^2}\sin\theta_e}{1+e_d\cos\theta_e}\right]$$

In this equation, τ is the orbital period of the de-orbit trajectory.

Therefore, the flight time between the de-orbit impulse time and entry is given by

$$\Delta t = tof\left(\theta_e\right) - tof\left(180^\circ\right) = tof\left(\theta_e\right) - \frac{\tau}{2}$$

Finally, the speed at reentry V_e can be determined from

$$V_e = \sqrt{\frac{2\mu}{r_e} - \frac{\mu}{a_d}}$$

APPENDIX D

Flight Path Coordinates

Relative flight path coordinates are defined with respect to a rotating Earth. This set of coordinates consists of the following trajectory elements

$$
\begin{aligned}
r &= \text{ geocentric radius} \\
V &= \text{ speed} \\
\gamma &= \text{ flight path angle} \\
\delta &= \text{ geocentric declination} \\
\lambda &= \text{ geographic longitude } (+\text{ east}) \\
\psi &= \text{ flight azimuth } (+\text{ clockwise from north})
\end{aligned}
$$

Please note the sign and direction convention.

The following are several useful equations that summarize the relationships between inertial and relative flight path coordinates.

$$
\begin{aligned}
& v_r \sin\gamma_r = v_i \sin\gamma_i \\
& v_r \cos\gamma_r \cos\psi_r = v_i \cos\gamma_i \cos\psi_i \\
& v_r \cos\gamma_r \sin\psi_r + \omega_e r \cos\delta = v_i \cos\gamma_i \sin\psi_i
\end{aligned}
$$

where the r subscript denotes relative coordinates and the i subscript inertial coordinates.

The inertial speed can also be computed from the following expression

$$v_i = \sqrt{v^2 + 2v\, r\omega \cos\gamma \sin\psi \cos\delta + r^2\omega^2 \cos^2\delta}$$

The inertial flight path angle can be computed from

$$\cos\gamma_i = \sqrt{\frac{v^2\cos^2\gamma + 2vr\omega\cos\gamma\cos\psi\cos\delta + r^2\omega^2\cos^2\delta}{v^2 + 2vr\omega\cos\gamma\cos\psi\cos\delta + r^2\omega^2\cos^2\delta}}$$

The inertial azimuth can be computed from

$$\cos\psi_i = \frac{v\cos\gamma\cos\psi + r\omega\cos\delta}{\sqrt{v^2\cos^2\gamma + 2vr\omega\cos\gamma\cos\psi\cos\delta + r^2\omega^2\cos^2\delta}}$$

where all coordinates on the right-hand-side of these equations are relative to a rotating Earth.

The transformation of an Earth-centered inertial (ECI) position vector $\mathbf{r}_{ECI}$ to an Earth-centered fixed (ECF) position vector $\mathbf{r}_{ECF}$ is given by the following vector-matrix operation

$$\mathbf{r}_{ECF} = [\mathbf{T}]\mathbf{r}_{ECI}$$

where the elements of the transformation matrix $[\mathbf{T}]$ are given by

$$[\mathbf{T}]=\begin{bmatrix}\cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}$$

and θ is the Greenwich apparent sidereal time at the moment of interest. Greenwich sidereal time is given by the following expression:

$$\theta = \theta_{g0} + \omega_e t$$

where θ_{g0} is the Greenwich sidereal time at 0 hours UTC, ω_e is the inertial rotation rate of the Earth, and t is the elapsed time since 0 hours UTC.

Finally, the flight path coordinates are determined from the following set of equations

$$r = \sqrt{r_{ECF}^2 + r_{ECF_y}^2 + r_{ECF}^2} \qquad v = \sqrt{v_{ECF}^2 + v_{ECF_y}^2 + v_{ECF_z}^2}$$

$$\lambda = \tan^{-1}\left(r_{ECF_y}, r_{ECF_x}\right) \qquad \delta = \sin^{-1}\left(\frac{r_{ECF_z}}{\left|\mathbf{r}_{ECF}\right|}\right)$$

$$\gamma = \sin^{-1}\left(-\frac{v_{R_z}}{\left|\mathbf{v}_R\right|}\right) \qquad \psi = \tan^{-1}\left[v_{R_y}, v_{R_x}\right]$$

where

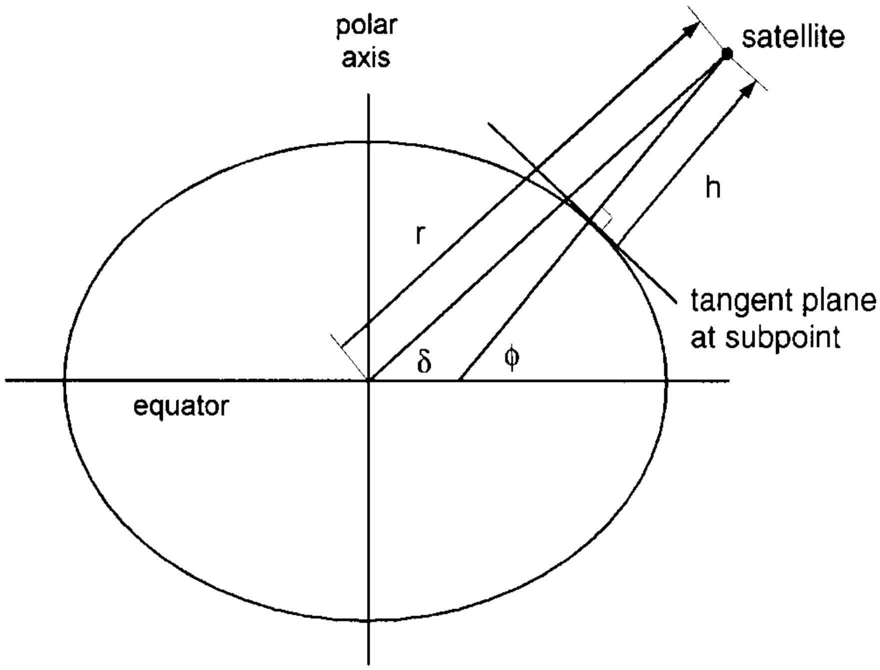
$$\mathbf{v}_R = \begin{bmatrix}-\sin\delta\cos\lambda & -\sin\delta\sin\lambda & \cos\delta\\ -\sin\lambda & \cos\lambda & 0\\ -\cos\delta\cos\lambda & -\cos\delta\sin\lambda & -\sin\delta\end{bmatrix}\mathbf{v}_{ECF}$$

Please note the two-argument inverse tangent calculation for λ and ψ is a four-quadrant operation.

APPENDIX E

Geodetic Coordinates

The following diagram illustrates the geometric relationship between geocentric and geodetic coordinates.



In this diagram, δ is the geocentric declination, ϕ is the geodetic latitude, r is the geocentric distance, and h is the geodetic altitude. The exact mathematical relationship between geocentric and geodetic coordinates is given by the following system of two nonlinear equations

$$(c+h)\cos\phi - r\cos\delta = 0$$
$$(s+h)\sin\phi - r\sin\delta = 0$$

where the geodetic constants c and s are given by

$$c = \frac{r_{eq}}{\sqrt{1-\left(2f - f^2\right)\sin^2\phi}} \qquad s = c\left(1-f\right)^2$$

and r_{eq} is the Earth equatorial radius (6378.14 kilometers) and f is the flattening factor for the Earth (1/298.257).

In this MATLAB script, the geodetic latitude is determined using the following expression

$$\phi = \delta + \left(\frac{\sin 2\delta}{\rho}\right) f + \left[\left(\frac{1}{\rho^2} - \frac{1}{4\rho}\right)\sin 4\delta\right] f^2$$

which is a series expansion in flattening factor (NASA TN D-7522).

The geodetic altitude is calculated from

$$\hat{h} = \left(\hat{r} - 1\right) + \left\{ \left(\frac{1 - \cos 2\delta}{2} \right) f + \left[\left(\frac{1}{4\rho} - \frac{1}{16} \right) \left(1 - \cos 4\delta\right) \right] f^2 \right\}$$

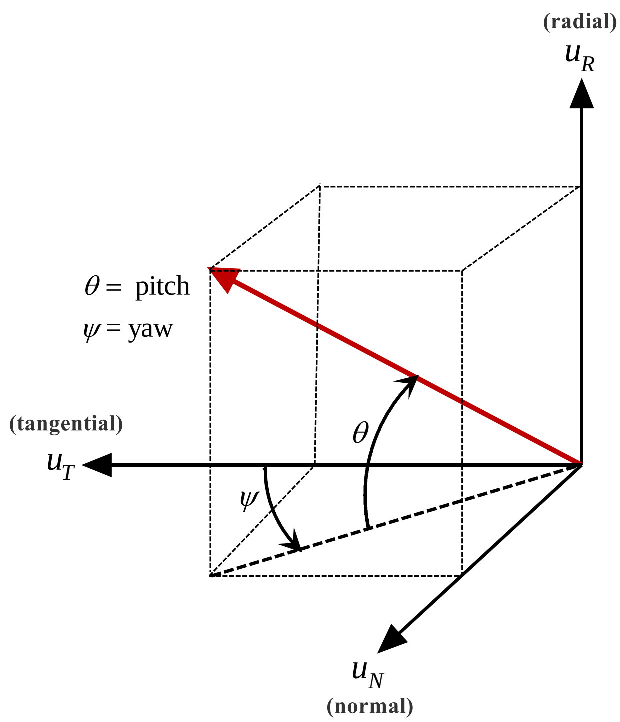
In these equations, ρ is the geocentric distance of the satellite, $\hat{h} = h / r_{eq}$ and $\hat{r} = \rho / r_{eq}$.

APPENDIX F

Pitch and Yaw Angles

The pitch and yaw angles for the de-orbit impulsive maneuver are computed and displayed in the local-vertical-local horizontal (LVLH; also called the radial-tangential-normal RTN) 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 pitch angle varies between ± 90 degrees and the yaw angle can have a value between ± 180 degrees.



The following is the MATLAB source code for the function that computes the orientation angles using the Earth-centered-inertial (ECI) position and velocity vectors (`reci`, `veci`) at the impulse location and the unit pointing vector of the impulsive delta-v (`ueci`).

```
function [pitch, yaw] = ueci2angles(reci, veci, ueci)

% convect eci unit vector to rtn angles

% input

%  reci = eci position vector (kilometers)
%  veci = eci velocity vector (kilometers/second)
%  ueci = eci unit vector

% output
```

```
%  pitch = pitch angle (radians)
%          positive above the local horizon
%  yaw   = yaw angle (radians)
%          positive in the direction of the angular momentum vector

% Orbital Mechanics with MATLAB

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% compute radial frame unit vectors

rmag = norm(reci);

xrdl = reci / rmag;

zrdl = cross(reci, veci);

hmag = norm(zrdl);

zrdl = zrdl / hmag;

yrdl = cross(zrdl, xrdl);

% unit vector in radial-tangential-normal frame

umee(1) = dot(ueci, xrdl);

umee(2) = dot(ueci, yrdl);

umee(3) = dot(ueci, zrdl);

% pitch angle (radians)

pitch = asin(umee(1));

% yaw angle (radians)

yaw = atan2(umee(3), umee(2));
```