TO: Prof. John Keesee, Prof. David Miller, and Ms. Marilyn Good

FROM: Namiko Yamamoto, Seung Chung

SUBJECT: Problem Set #2 (Orbit and Propulsion)

DATE: 12/11/2003

Problem Set #2
(Orbit and Propulsion Subsystems)

Motivation

In a mission design in which an orbit transfer is necessary, the orbit transfer and propulsion system types may drive of the design. Many missions desire minimal mass to reduce the mission cost, which may be achieved in part by choosing the appropriate propulsion system. Some military missions and manned missions, however, may prefer minimal orbit transfer time over minimal cost associated with minimal mass. In many missions, the choices for the orbit transfer and propulsion system may be obvious. For some missions, the appropriate design is not obvious, given that designing a propulsion system depends on the transfer orbit chosen, and vice versa. For such design problem, a tool that designs the orbit transfer and propulsion system may alleviate the difficulty in solving the coupled problem.

Problem Statement

The objective is to design a tool that compares a set of feasible orbit transfer and propulsion system combinations for a given mission requirement. The mission requirement is in terms of the desired orbit transfers. For simplicity, we will only consider an orbit transfer from a lower circular orbit to a higher circular orbit. The output of the design tool should be a set of feasible transfer orbit and propulsion system combinations. The comparison metric is the mass of the propulsion system and the time required for orbit transfer.

Note that the set of possible transfer orbits depends on the propulsion system type, and the total ΔV_{total} required depends on the transfer orbit type. While this problem can become very complex depending on the accuracy of the computation and analysis, a right set of assumptions should be applied to minimize the computational complexity while maintaining the fidelity for comparison among various designs (e.g. assume fixed dry mass, but excluding the propulsion system mass). The satellite mission will have the following guidelines:

1. Orbit transfer takes place purely by provision of additional energy by the propulsion system. No plane change is considered.
2. Shapes of initial and final orbits are circular.
3. Both initial and final orbits are assumed to satisfy all components which affect possibility, duration, and efficacy of satellite mission, such as launch windows, radiation effects, earth convergence and others.

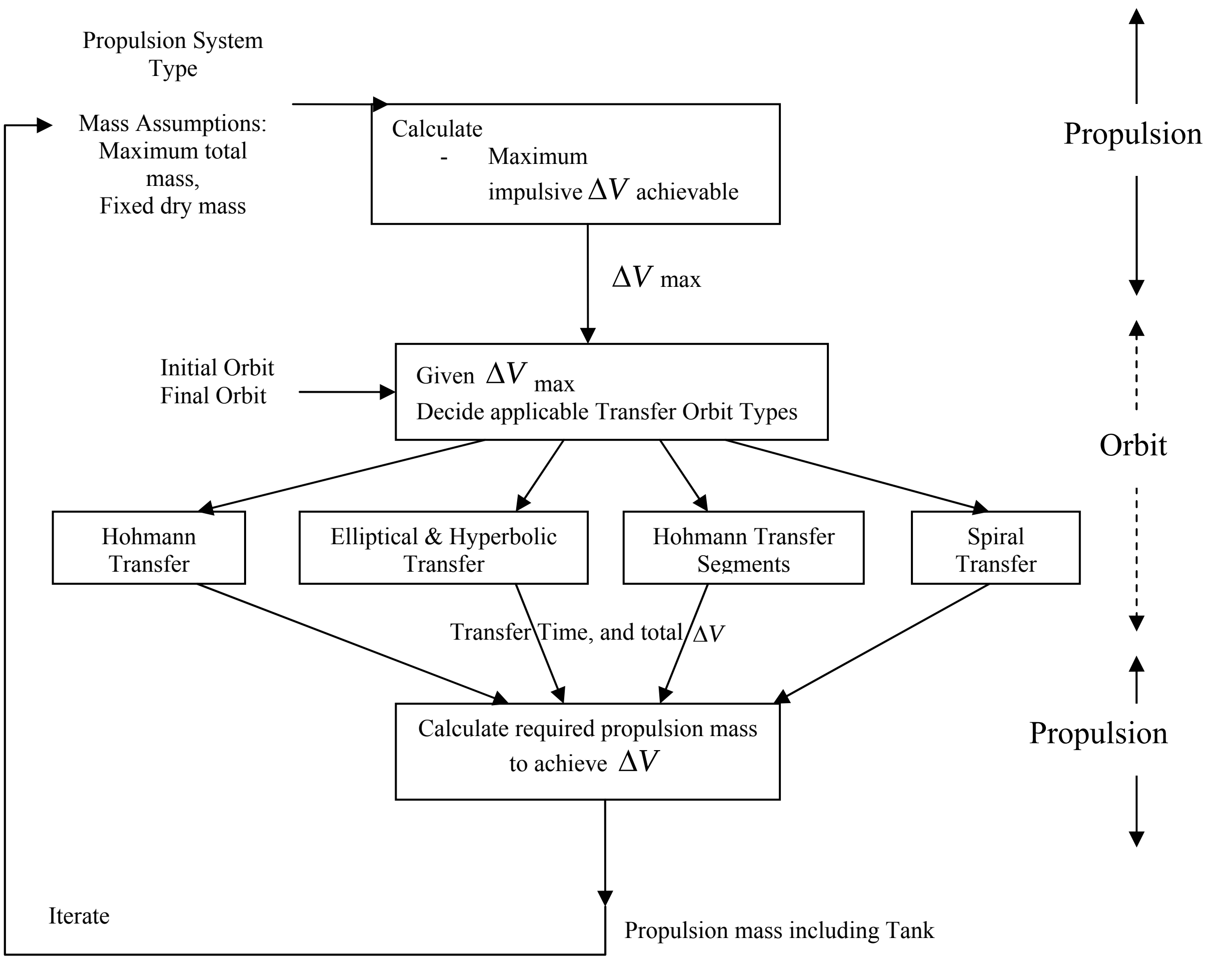
4. The dry mass, m_{dry}, (excluding the propulsion system mass) is fixed. The power subsystem mass have a fixed mass, although its crucial influence on comparison between electric propulsion systems are noted.
5. The maximum initial mass, m_{max}, is fixed.
6. Transfer orbit types to consider are:
   a. Elliptical & Hyperbolic Transfer
   b. Hohmann Transfer
   c. Hohmann Transfer Segments
   d. Spiral Transfer
7. Types of propulsion system to consider are:
   a. Solid Motor
   b. Chemical
      i. Bipropellant
      ii. Monopropellant
   c. Electric

Propulsion systems can be broken down further with different propellants and associated I_{sp}s. cold gas systems are excluded because of the combination of low thrust and low I_{sp}. In general, cold gas systems are not applicable to orbit transfers even for satellites that are extreme sensitivity to contamination and/or for which the complexity of the propulsion system is of an issue.

Approach

Flow chart below shows the outline of out approach procedure.

The correlation between orbit and propulsion system is complementary for each other. The velocity change calculated from initial and final orbits selects propulsion systems such that achieve that value. On the other hand, propulsion system with a certain velocity change budget constrains the types of orbital maneuvers. Calculation starting with certain assumptions and releasing those assumptions as variable later is required to get out of this loop relationship. In this MATLAB code, we commenced the calculation with propulsion system and associated achievable velocity change limit as inputs, and checked its validity with velocity change obtained from transfer orbits later.



STEP 1: Compute the maximum ΔV available.

To calculate the maximum velocity change achievable and the maximum firing time by a propulsion system, Impulsive Maneuvers for Orbit Transfer is considered. For most orbit transfer calculations, each ΔV is assumed impulsive given the thrust is "relatively" high. That is, with high thrust capability, an infinitesimal time is necessary to achieve the necessary ΔV, and as such, we can approximate the burn as an impulse. If the thrust is relatively low, however, the maneuver will require a finite amount of time.

Figure 1 illustrates the case in which the burn time is finite. In such case, the thrust vector is not along the flight path and results in gravity loss as shown by the second term on the right hand side of Eq. (1). If the thrust vector were aligned along the flight path, the gravity loss would be eliminated. The resulting orbit transfer, however, will not be the intended one. Thus, for a spacecraft to achieve the intended orbit transfer, the burn time must be short enough such that the gravity loss is negligible compared to the thrust.

$$\Delta V = gI_{sp} \ln \frac{m_i}{m_f} - \int_{t_i}^{t_f} g \sin \gamma dt \tag{1}$$

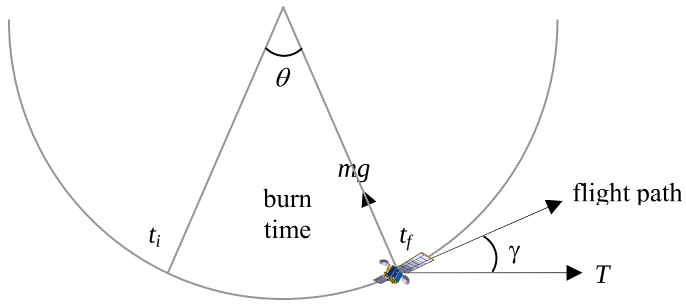


Figure 1. Forces during finite burn.

We use a small angle approximation, and assume that for $\gamma \leq \pi/12$, i.e. $\theta \leq \pi/6$, the gravity loss effect is minimal. Then, given the gravitational constant μ of the central body, current circular orbital radius r, mass of the spacecraft m_i, and available propellant mass m_{prop}, we can compute the maximum achievable ΔV_{max}:

$$\Delta V_{max} = gI_{sp} \ln\left(\frac{m_i}{m_i - \Delta m_{\max}}\right) \tag{2}$$

where

$$\Delta m_{\max} = \min\left[m_p, \frac{\theta T r^{3/2}}{gI_{sp}\sqrt{\mu}}\right]. \tag{3}$$

Note that m_o and m_p are initially unknown. Thus, we assume $m_i = m_{max}$ and $m_p = m_{max} - m_{dry}$. These values must be update once the actual values are known, and the remaining design processes must be repeated.

This is implemented in `compute_impulsive_Delta_V_max.m`.

Input

m_i	initial total mass, i.e. $m_{dry} + m_{wet}$
m_{prop}	available propellant mass
μ	gravitational constant of the body orbiting
r	radius of the initial orbit
T	thrust
I_{sp}	specific impulse

Output

ΔV_{max}	maximum impulsive ΔV available

STEP2: Compute the ΔV required for the orbit transfer.

Calculate the velocity change required and transfer time for each orbit transfer from orbit elements of given initial and final orbit elements.

Verify that velocity change for each firing is smaller than the maximum velocity achievable by a propulsion system.

Orbit element values provided by given initial and final orbits are follows:

- r_A: Radius of initial orbit
- r_B; Radius of final orbit
- ΔV_{max}; Maximum velocity change according to propellant

The satellite velocities on initial, transfer, and final orbits are calculated using these values mentioned above and the gravitational constant of a body the spacecraft is orbiting.

Hohmann Transfer

Hohmann Transfer orbit's ellipse is tangent to both the initial and final orbits at the transfer orbit's perigee and apogee respectively. This is one of the most fuel-efficient transfers between two circular coplanar orbits and the orbit transfer time is relatively small. The disadvantage is that a failure is unrecoverable during the orbit transfer.

Atx: Semimajor axis of transfer ellipse $$a_{tx} = \frac{r_A + r_B}{2}$$

Via: Velocity in the initial orbit $$v_{iA} = \sqrt{\frac{\mu}{r_A}}$$

Vfb: Velocity in the final orbit $$v_{fB} = \sqrt{\frac{\mu}{r_B}}$$

Vtxa: Velocity at perigee point on transfer orbit $$v_{txA} = \sqrt{\mu\left(\frac{2}{r_A} - \frac{1}{a_{tx}}\right)}$$

Vfxb: Velocity at apogee point on transfer orbit $$v_{txB} = \sqrt{\mu\left(\frac{2}{r_B} - \frac{1}{a_{tx}}\right)}$$

The velocity changes required at each staging is computed as follows.

ΔV_A: Velocity change at the perigee of transfer orbit (1st burn) $$\Delta V_A = \left|V_{txA} - V_{iA}\right|$$

ΔV_B: Velocity change at the apogee of transfer orbit (2nd burn) $$\Delta V_B = \left|V_{fB} - V_{txB}\right|$$

After verifying that both ΔV_A and ΔV_B are smaller than $\Delta V\,\mathrm{max}$ from propellant, total ΔV and transfer time can be determined by:

$$\Delta V_{total} = \Delta V_A + \Delta V_B$$

$$\Delta t = \pi \sqrt{\frac{a_{tx}^{\ 3}}{\mu}}$$

This is implemented in `compute_Hohmann_transfer.m`.

Input

r_A	inner orbit radius
r_B	outer orbit radius
μ	gravitational constant of the body orbiting

Output

ΔV	ΔV required for transfer
Δt	time required for transfer

Elliptical & Hyperbolic Transfer

Elliptical & Hyperbolic Transfer is a modified version of Hohmann to achieve rapid transfer. Extra energy was put at the first burn, and transfer orbit can be a bigger ellipse than Hohmann transfer orbit or a hyperbola.

First, decide maximum rc, apoapsis of transfer orbit, which should achieve the fastest transfer, according to ΔV max due to propellant.

atx; Semimajor axis of transfer eclipse $atx = \frac{ra + rc}{2}$

Via: Velocity in the initial orbit $Via = \sqrt{\frac{\mu}{ra}}$

Vfb: Velocity in the final orbit $Vfb = \sqrt{\frac{\mu}{rb}}$

Vtxa: Velocity at perigee point on transfer orbit $Vtxa = \sqrt{\mu\left(\frac{2}{ra} - \frac{1}{atx}\right)}$

Vfxb: Velocity at apogee point on transfer orbit $Vtxb = \sqrt{\mu\left(\frac{2}{rb} - \frac{1}{atx}\right)}$

ΔVa : Velocity change at the perigee of transfer orbit (1st burn) $\Delta Va = |Vtxa - Via|$

ΔVb : Velocity change at the apogee of transfer orbit (2nd burn)

$$\Delta Vb = \sqrt{Vfb^2 + Vtxb^2 - 2VfbVtxb\cos\phi}$$

where

$$\phi = \tan^{-1}\left[\frac{e\sin\nu}{1+e\cos\nu}\right]$$

$$\nu = \cos^{-1}\left[\frac{\left\{\frac{atx\left(1-e^2\right)}{rb}-1\right\}}{e}\right]$$

Considering that both ΔVa and ΔVb are smaller than $\Delta V\max$ from propellant, we can get maximum rc, accordingly. Also, e is also determined by:

$$e = 1 - \frac{ra}{atx}$$

1) $\Delta Va > \Delta Vb$

First, the code assumes that ΔVa is the limitation. By deploying the equation of $\Delta Va = \Delta V\max$, maximum atx and rc are calculated as follows:

$$a_{tx} = \frac{1}{\frac{2}{r_A} - \frac{\left(\sqrt{\frac{\mu}{r_A}} + \Delta V_{\max}\right)^2}{\mu}}$$

$$rc = 2atx - ra$$

Verify that ΔVb corresponding to this rc is smaller than $\Delta V\max$.

2) $\Delta Vb > \Delta Va$

Try out this case only if $\Delta Vb > \Delta V\max$.

Start from the decent number of rc (>rb), and increase the number until ΔVb value becomes bigger than $\Delta V\max$. The maximum value of rc which still achieve $\Delta Vb < \Delta V\max$ determines the fastest transfer.

In either case, plugging into rc value obtained into the equations above leads to ΔV total and transfer time required.

$$\Delta Vtotal = \Delta Va + \Delta Vb$$

$$t = 0.001583913 atx^{\frac{3}{2}}\left(E - e\sin E\right)$$

where

$$E = \cos^{-1}\left[\frac{e + \cos\nu}{1 + e\cos\nu}\right]$$

This is implemented in `compute_high_energy_transfer.m`.

Input

r_A	inner orbit radius
r_B	outer orbit radius
μ	gravitational constant of the body orbiting
ΔV_{max}	maximum impulsive ΔV available

Output

ΔV	ΔV required for transfer
Δt	time required for transfer

Hohmann Transfer Segments

This transfer orbit is often chosen for a low-thrust transfer, using low thrust chemical or electrical propulsion.

First, calculate the value of atk (semimajor axis of k^{th} transfer eclipse) achievable using $\Delta V\max$ from propellant by deploying $\Delta Vak = \Delta V\max$:

atk: Semimajor axis of k^{th} transfer

$$atk = \frac{1}{\dfrac{2}{ra} - \dfrac{\left(\sqrt{\mu\left(\dfrac{2}{ra} - \dfrac{1}{at(k-1)}\right)} + \Delta V\max\right)^2}{\mu}}$$

rck: Apoapsis of k^{th} transfer orbit $\quad rck = 2atx - ra$

where
at0 = ra (k=0)
atK = rb (k=K, final orbit)

Via: Velocity in the initial orbit $\quad Via = \sqrt{\dfrac{\mu}{ra}}$

Vfb: Velocity in the final orbit $\quad Vfb = \sqrt{\dfrac{\mu}{rb}}$

Vtka: Velocity at perigee point on k^{th} transfer orbit $\quad Vtka = \sqrt{\mu\left(\dfrac{2}{ra} - \dfrac{1}{atkx}\right)}$

ΔVi Velocity change at the perigee of transfer orbit on k^{th} burn $\quad \Delta Vak = \left| Vtka - Vt(k-1)a \right|$

Total velocity change and total transfer time required are accumulated in each transfer orbit segments, and are determined by:

$$\Delta Vtotal = \sum_{i=1}^{k} \Delta Vi + \mu\left(\frac{2}{rb} - \frac{1}{atk}\right)$$

$$t = \sum_{i=1}^{k-1} 2\Pi\sqrt{\frac{(ra+rk)^3}{8\mu}} + \Pi\sqrt{\frac{(ra+rk^3)}{8\mu}}$$

Spiral Orbit Transfer

Given a propulsion system with a very low acceleration capability (i.e.T/W < 10^{-3}), we cannot approximate each maneuver as impulsive due to high gravity loss (see Approach, STEP1). This implies that the aforementioned orbit transfers cannot be used. In such case, we can use a spiral orbit transfer (see Figure 2). During a spiral orbit transfer, the spacecraft thrusts continuously until it reaches the desired orbit.

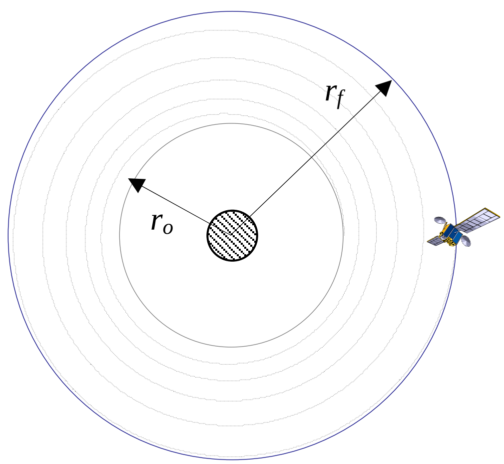


Figure 2. Low thrust spiral orbit.

Edelbaum [2] has introduced a method for optimal low-thrust transfer between inclined circular orbits, which is also discussed in [1]. For coplanar orbit transfer between two circular orbits, however, we can derive the equations for the necessary ΔV and the time for orbit transfer using basic knowledge of the orbital mechanics and few key assumptions.

Assuming that the spacecraft thrusts in the direction of it's velocity vector, the change in the specific mechanical energy, ε, of the spacecraft with mass m is

$$\frac{d\varepsilon}{dt} = \frac{T}{m} \cdot v = a \cdot v, \tag{4}$$

where T is thrust (assumed constant), a is the acceleration of the spacecraft, and v is velocity. Given that the thrust is very small, we can assume that the orbit remains relatively circular. That is,

$$\frac{d\varepsilon}{dt} = \frac{\mu}{2r^2}\frac{dr}{dt}, \quad (5)$$

where μ is the gravitational constant and r is the radius at which the spacecraft is in. Equating Eqs. (4) and (5), and assuming constant acceleration, we can compute the amount of time the spacecraft takes, Δt, to transfer to a circular orbit at r_B radius:

$$\Delta t = \frac{1}{a}\left(\sqrt{\frac{\mu}{r_A}} - \sqrt{\frac{\mu}{r_B}}\right) \quad (6)$$

The total ΔV necessary to transfer to a circular orbit with radius r_B from r_A is then given by:

$$\Delta V_{total} = a \cdot \Delta t \quad (7)$$

This is implemented in `compute_spriral_transfer.m`.

Input

r_A	inner orbit radius
r_B	outer orbit radius
μ	gravitational constant of the body orbiting
m_i	initial total mass, i.e. $m_{dry} + m_{wet}$
T	thrust

Output

ΔV	ΔV required for transfer
Δt	time required for transfer

STEP 3: Compute the propulsion system mass.

Once propulsion system type has been chosen and the necessary ΔV budget is determined, we can compute the necessary propellant mass m_{prop} from the rocket equation:

$$m_{prop} = m_{dry} e^{\frac{\Delta V}{g_0 I_{sp}}} - m_{dry} \quad (8)$$

where I_{sp} is the specific impulse of the propulsion system chosen, g_0 is the acceleration due to Earth's gravitational force at the sea level. Note that in general, the true dry mass m_{dry} is unknown since the propulsion system mass is unknown among others. Thus, this number has to be estimated in the beginning and must be updated as better estimate of the dry mass is generated.

Now with the propellant mass, we can calculate the mass of the propulsion system. Once the propulsion system mass is calculated, the dry mass must be updated and the whole process must be iterated until it converges to a desired level of confidence.

Chemical

We could go through a detailed analysis to determine the mass of a chemical propulsion system, such as monopropellant and bipropellant systems. The analysis, however, become very complex and tedious. Furthermore, there are many detailed design choices required to compute the propulsion system mass. While such tool would be great for future use, we take a very simple approach to approximating the chemical propulsion system mass. That is, we assume that approximately 85 ~ 90% of the system mass is the propellant mass [5]. Pressurant, tanks, lines, fittings, components, etc. contributes to the remaining 15%.

$$m_{Total} = \frac{m_{prop}}{0.85} \tag{9}$$

Solid

Similar to the chemical propulsion system, we take a crude approximation of the solid rocket propulsion system mass as a mass fraction. Typically, 82 ~ 94% of a solid rocket prolusion system mass account for the propellant mass. We take the conservative 82%.

$$m_{Total} = \frac{m_{prop}}{0.82} \tag{10}$$

Electric Propulsion

For electric propulsion, we will use a simple empirical model [3] to estimate the mass of the propulsion system. This is will more than suffice for our desired level of fidelity. This tool could be replaced with higher fidelity model as desired or necessary in the future.

First, we compute the efficiency of the propulsion system η as follows:

$$\eta = \eta_{thruster}\eta_{PowerProcessor} = A + B\ln\left(I_{sp}\right) \tag{11}$$

where the constants A and B are defined in Table 1.

Table 1. Empirical data for modeling the efficiency and the specific mass of an electric propulsion system.

Propulsion System	Constants for Models			
	A	B	C	D
H_2 Arcjet			5.0	0
NH_3 Arcjet			1.8	0
Ar Ion	-2.024	0.307	4490	-0.781

Xe Ion	-1.776	0.307	123100	-1.198
Hg Ion	-0.765	0.181	82870	-1.136
Ar MPD	-0.591	0.126	7	0
Ar PIT	-1.99	0.32	7	0

Given the flow rate of the propellant as a function of the propellant mass and the duration for with the thruster is turned on Δt_{burn},

$$\dot{m} = \frac{m_{prop}}{\Delta t_{burn}} \tag{12}$$

we can compute the necessary power:

$$P = \frac{\dot{m}\left(I_{sp} g_0\right)^2}{2\eta} \tag{13}$$

Then, the mass of the thruster and the power processor is computed from an empirical formula using Table 1 where:

$$m_{Thruster} + m_{PowerProcessor} = C I_{sp}{}^{D} P \tag{14}$$

Finally the tank mass is calculated using one of the equation listed in Table 2.

Table 2. Electric propulsion system propellant tank mass.

Propellant Type	Propellant Mass Range (kg)	Tank Mass (kg)
NH_3	5000 ~ 18300	$120 + 0.173\ m_{prop} + 2.28\ m_{prop}{}^{2/3}$
NH_3	18300 ~ 2200	$1020 + 0.198\ m_{prop}$
He	5000 ~ 13000	$610 + 0.493\ m_{prop}$
Xe	5000 ~ 2200	$52 + 0.075\ m_{prop} + 0.154\ m_{prop}{}^{2/3}$

The total electrical propulsion system ass is:

$$m_{Total} = m_{Thruster} + m_{\text{Prower Processr}} + m_{Tank} + m_{prop} \tag{15}$$

While the mass of the necessary power source for the propulsion system may be significant, it isn't included as it overlaps with the power subsystem's requirements.

Implementation Source Code

compute_impulsive_Delta_V_max.m

```
function Delta_V_max =
compute_impulsive_Delta_V_max(m_o,m_prop,mu,r,thruster)
% DELTA_V_MAX = COMPUTE_IMPULSIVE_DELTA_V_MAX(M_O,M_PROP)
%
% Inputs:
%   M_O             Initial total mass, i.e. m_dry + m_wet
%   M_PROP          Available propellant mass
%   MU              Gravitational constant of the body orbiting
%   R               radius of the initial orbit
%   TRUSTER         Thruster info
%
% Outputs:
%   DELTA_V_MAX     Maximum impulsive Delta V available

% reject any invalid inputs
if (m_o <= m_prop)
    error('Error! Propellant mass must be less than the initial spacecraft
mass.');
end

theta = 2*pi/12;    % Angle of the orbit for wich the Delta V occurs. [rad]
                    % Assume pi/12 ~ 0, i.e. small angle approximation.

g = 98.1;           % Earth gravity at sea level [m/s^2]

% Maximum propellant available during an impulsive burn. [kg]
Delta_m_max = min(m_prop, theta * thruster.T * r^(3/2) /...
                       (g*thruster.I_sp * sqrt(mu)));

% Maximum Delta_V available during an impulsive burn. [m/s]
Delta_V_max = g * thruster.I_sp * log(m_o/(m_o - Delta_m_max));
```

compute_high_energy_transfer.m

```
function [Delta_V,Delta_t] =
compute_high_energy_transfer(mu,r_A,r_B,Delta_V_max)
% [DELTA_V,DELTA_T] = COMPUTE_HIGH_ENERGY_TRANSFER(MU,R_A,R_B,Delta_V_max)
%
% Inputs:
%   MU              Gravitational constant of the body orbiting
%   R_A             Radius of the inner circular orbit
%   R_B             Radius of the outer circular orbit
%   Delta_V_max     Maximum impulsive Delta_V available
%
% Outputs:
%   DELTA_V         Total Delta_V necessary for Hohmann transfer
%   Delta_T         Orbit transfer time

% Constrain so that the first burn is no greater than the max allowed.
Delta_V_A = Delta_V_max;

v_iA = sqrt(mu/r_A);                % Velocity for inner circular orbit [m/s]
v_fB = sqrt(mu/r_B);                % Velocity for outer circular orbit [m/s]

% Compute the transfer orbit assuming that the maximum Delta_V_max is used
% at point A.
```

```
v_txA = Delta_V_max + v_iA;         % Velocity for transfer orbit at first
burn [m/s]
a_tx = 1/(2/r_A - v_txA^2/mu);      % Semi-major axis of the transfer orbit
[m]
e = 1 - r_A/a_tx;                   % Eccentricity of the transfer orbit
nu = acos((a_tx*(1-e^2)/r_B - 1)/e);% True anomaly at the second burn [rad]
phi = atan(e*sin(nu)/(1+e*cos(nu)));% Flight path angle at second burn [rad]
v_txB = sqrt(mu*(2/r_B - 1/a_tx));  % Velocity for transfer orbit at second
burn [m/s]
% Second burn Delta_V necessary [m/s]
Delta_V_B = sqrt(v_fB^2 + v_txB^2 - 2*v_fB*v_txB*cos(phi));

% Check to see if the second burn is less than the maximum allowed.
if (Delta_V_B <= Delta_V_max)
    % The first burn is the limiting factor.
    Delta_V = Delta_V_A + Delta_V_B;            % Total Delta_V [m/s]
    E = acos((e + cos(nu))/(1+e*cos(nu)));      % Eccentricy anomaly at
second burn [rad]
    Delta_t = sqrt(a_tx^3/mu)*(E-e*sin(E));     % Orbit transfer time [sec]
    return
else
    % The second burn is the liminting factor.
    Delta_V_B = Delta_V_max;                    % Delta_V at second burn
[m/s]
    a_tx = search_a_tx(mu,r_A,r_B,Delta_V_B);   % Semi-major axis of transfer
oribt [m]
    v_txA = sqrt(mu*(2/r_A - 1/a_tx));          % Velocity for tranfer orbit
at first burn [m/s]
    Delta_V_A = v_txA -v_iA;                    % Delta_V at first burn [m/s]
    Delta_V = Delta_V_A + Delta_V_B;            % Total Delta_V [m/s]
    e = 1 - r_A/a_tx;                           % Eccentricity of the
transfer orbit
    nu = acos((a_tx*(1-e^2)/r_B - 1)/e);        % True anomaly at the second
burn [rad]
    E = acos((e + cos(nu))/(1+e*cos(nu)));      % Eccentricy anomaly at
second burn [rad]
    Delta_t = sqrt(a_tx^3/mu)*(E-e*sin(E));     % Orbit transfer time [sec]
end

function Delta_V_B = compute_Delta_V_B(a_tx,mu,r_A,r_B)
e = 1 - r_A/a_tx                                % Eccentricity of the
transfer orbit
nu = acos((a_tx*(1-e^2)/r_B - 1)/e);            % True anomaly at the second
burn [rad]
phi = atan(e*sin(nu)/(1+e*cos(nu)));            % Flight path angle at second
burn [rad]
v_txB = sqrt(mu*(2/r_B - 1/a_tx));              % Velocity for transfer orbit
at second burn [m/s]
v_fB = sqrt(mu/r_B);                            % Velocity for outer circular
orbit [m/s]
Delta_V_B = sqrt(v_fB^2 + v_txB^2 - 2*v_fB*v_txB*cos(phi))

function a_tx = search_a_tx(mu,r_A,r_B,Delta_V_B)
allowable_percent_error = 0.000001;
a_tx_low = r_B/2;
a_tx = (r_A + r_B)/2;
a_tx_high = (r_A + r_B)*10000;
```

```
quit = 0;
while (not(quit))
    difference = Delta_V_B - compute_Delta_V_B(a_tx,mu,r_A,r_B); % This
should be zero
    if ((a_tx_high - a_tx)/a_tx < allowable_percent_error | ...
            (a_tx - a_tx_low)/a_tx < allowable_percenT_error);
        quit = 1;
    elseif (difference > 0)
        a_tx_low = a_tx;
        a_tx = (a_tx_high - a_tx)/2;
    elseif (difference < 0)
        a_tx_high = a_tx;
        a_tx = (a_tx - a_tx_low)/2;
    end
end
```

compute_Hohmann_transfer

```
function [Delta_V,Delta_t] = compute_Hohmann_transfer(mu,r_A,r_B)
% [DELTA_V,DELTA_T] = COMPUTE_HOHMANN_TRANSFER(MU,R_A,R_B)
%
% Inputs:
%   MU              Gravitational constant of the body orbiting
%   R_A             Radius of the inner circular orbit
%   R_B             Radius of the outer circular orbit
%
% Outputs:
%   DELTA_V         Total Delta_V necessary for Hohmann transfer
%   Delta_T         Hohman transfer time

% reject any invalid inputs
if (r_A > r_B)
    error('Error! Inner orbit radius is larger than the outer.');
end

v_iA = sqrt(mu/r_A);                 % Velocity for inner orbit [m/s]
v_fB = sqrt(mu/r_B);                 % Velocity for outer orbit [m/s]
a_tx = (r_A + r_B)/2;                % Semi-major axis of transfer orbit [m/s]
v_txA = sqrt(mu*(2/r_A - 1/a_tx));  % Velocity at periapsis for m/s]
v_txB = sqrt(mu*(2/r_B - 1/a_tx));  % Velocity at apoapsis for [m/s]
Delta_V_A = v_txA - v_iA;           % Delta_V at periapsis of [m/s]
Delta_V_B = v_fB - v_txB;           % Delta_V at apoapsis of transfer [m/s]
Delta_V = Delta_V_A + Delta_V_B;    % Total Delta V required transfer [m/s]
Delta_t = pi*sqrt(a_tx^3/mu);       % Hohmann transfer time [s]
```

compute_spiral_transfer

```
function [Delta_V,Delta_t] = compute_spiral_transfer(mu,r_A,r_B,m_i,T)
% [DELTA_V,DELTA_T] = COMPUTE_SPIRAL_TRANSFER(MU,R_A,R_B,M_I,T)
%
% Inputs:
%   MU              Gravitational constant of the body orbiting
%   R_A             Radius of the inner circular orbit
%   R_B             Radius of the outer circular orbit
%   M_I             Initial Total Mass
%   T               Thrust
```

```
%
% Outputs:
%   DELTA_V         Total Delta_V necessary for Hohmann transfer
%   Delta_T         Hohman transfer time

% reject any invalid inputs
if (r_A > r_B)
    error('Error! Inner orbit radius is larger than the outer.');
end
if (T/m_i*g > 10e3)
    error('Error! Too much thrust for spiral transfer assumptions.');
end

a = T/m_i;                                  % Acceleration [m/s^2]
Delta_t = (sqrt(mu/r_A) - sqrt(mu/r_B))/a;  % Orbit transfer time [s]
Delta_V = a*Delta_t;        % Total Delta V required for orbit transfer [m/s]
```

References

1. Chobotov V., *Orbital Mechanics*, 2nd ed. Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 1996.
2. Edelbaum, T., “Propulsion Requirements for Controllable Satellites,” *ARS Journal*, Vol. 31, August 1961, pp. 1079-1089.
3. Humble R., Henry G., Larson W., *Space Propulsion Analysis and Design*., New York: The McGraw-Hill Companies, Inc. Primis custom Publishing, 1995.
4. Sutton G. and Biblarz O., *Rocket Propulsion Elements*, New York: John Wiley & Sons, Inc., 2001.
5. Wertz, J., and Larson, W., *Space Mission Analysis and Design,* 3rd ed. Torrance, CA: Microcosm Press, 1999.