Nicholas Luis (PSU ID: 930843181)
Aersp 450 - HW2

Problem 1)

given: $\mu = 39{,}860 \frac{km^3}{s^2}$

| | | | |
|---|---|---|---|
| Departure (near-circular LEO) | $a_1 = 8000$ km | $e_1 = 0.01$ | $f_1 = 30° = \frac{\pi}{6}$ rad |
| Arrival (HEO) | $a_2 = 27000$ km | $e_2 = 0.6$ | $f_2 = 210° = \frac{7}{6}\pi$ rad |

no plane change
($\Delta I = 0, \Delta\Omega = 0$)

1) $a_T = ?$ (transfer orbit semi-major axis)

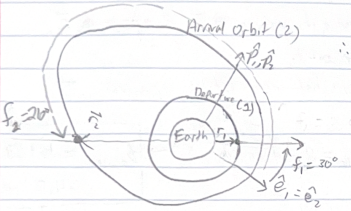
$a_{min_T} = \frac{1}{4}(r_1 + r_2 + \sqrt{r_1^2 + r_2^2 - 2r_1 r_2 \cos(\Delta f)})$, $r_1 = ?$, $r_2 = ?$, $\Delta f = ?$

* $f_2 - f_1$, because it is in different frames

$r = \frac{a(1-e^2)}{1+e\cos f}$

$r_1 = \frac{a_1(1-e_1^2)}{1+e_1\cos f_1} \rightarrow r_1 = \frac{(8000)(1-0.01^2)}{1+(0.01)\cos(30°)}$ km $\rightarrow r_1 = 7930.519685$ km

$r_2 = \frac{a_2(1-e_2^2)}{1+e_2\cos f_2} \rightarrow r_2 = \frac{(27000)(1-0.6^2)}{1+(0.6)\cos(210°)}$ km $\rightarrow r_2 = 35971.16628$ km




The departure orbit is near-circular. ∴ We can approximate that it really is circular. For circular orbits, we can choose the orientation of the $\hat{e}, \hat{p}$ frame. I chose to align it with that of the high eccentricity orbit.

Because $\hat{e}_1 = \hat{e}_2$, we are able to simply subtract $f_2 - f_1$ to find $\Delta f$

$\Delta f = f_2 - f_1 \rightarrow \Delta f = 210° - 30° \rightarrow \Delta f = 180°$

$a_{min,T} = \frac{1}{4}(r_1 + r_2 + \sqrt{r_1^2 + r_2^2 - 2r_1 r_2 \cos(\Delta f)})$

$a_{min,T} = \frac{1}{4}((7930) + (35971) + \sqrt{(7930)^2 + (35971)^2 - 2(7930)(35971)\cdot\cos(180°)})$

$a_{min,T} = 21950.5$ km

* note that the decimal places were omitted in writing, but used in the calculations

2) $p = ?$ given $p = \frac{km - 2a\ell}{4am - 2a\ell^2}$,

$k = r_1 r_2(1 - \cos\Delta f) \rightarrow k = (7930)(35971)(1 - \cos(180°)) \rightarrow k = 5.70954 \times 10^8\ km^2$

$m = r_1 r_2(1 + \cos\Delta f) \rightarrow m = (7930)(35971)(1 + \cos(180°)) \rightarrow m = 0\ km^2$

$\ell = r_1 + r_2 \rightarrow \ell = 7930 + 35971 \rightarrow \ell = 43902$ km

Plugging in $k, m, \ell, a_{min,T} \rightarrow$ $p_T = 12995.8582$ km

3) $\Delta V_1$ (Departure orbit → transfer orbit)

- using the energy equation, we can calculate the velocity at each point
- because $\Delta f = 180$, the velocity impulses will be parallel to the direction it's moving in already.

∴ $\Delta V = |V_{orbit} - V_{transfer,orbit}|$ (the vectors are already aligned)

↓next page

Nicholas Lui
930941391

Problem 1.3 (continued)

$\frac{1}{2}v^2 = \frac{\mu}{r} = -\frac{\mu}{2a} \rightarrow \frac{1}{2}v^2 = \frac{\mu}{r} - \frac{\mu}{2a} \rightarrow v = \sqrt{2\mu\left(\frac{1}{r} - \frac{1}{2a}\right)}$

$\rightarrow V_1 = 2\mu\left(\frac{1}{r_1} - \frac{1}{2a_1}\right) = 7.1203$ km/s

$\rightarrow V_{1T} = 2\mu\left(\frac{1}{r_1} - \frac{1}{2a_T}\right) = 9.0755$ km/s

Note that when moving from the departure orbit to transfer orbit, radius = constant for impulse

$\Delta V_1 = V_{1T} - V_1 \rightarrow \boxed{\Delta V_1 = 1.9552 \text{ km/s}}$

4) The same energy equation can be used to find the $\Delta V$ to go from the transfer orbit to the arrival orbit.

This process yields: $V_{2T} = 2.0009$ km/s

$V_2 = 2.7202$ km/s

$\rightarrow \boxed{\Delta V_2 = 0.7193 \text{ km/s}}$

5) Because $\Delta f = 180°$ (for a minimum energy transfer with $\Delta f = 180°$), the transfer orbit will have its perigee @ $\vec{r}_1$ & apogee @ $\vec{r}_2$.

Going from perigee → apogee is exactly half an orbit.

Therefore the time it takes is equal to one half the orbital period, $T_p$.

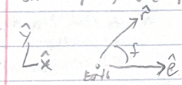
$T_p = 2\pi\sqrt{\frac{a^3}{\mu}} \rightarrow t_{transfer} = \pi\sqrt{\frac{a_T^3}{\mu}} \rightarrow t_{transfer} = 2\pi\sqrt{\frac{21951^3}{398600}}$ s $\rightarrow \boxed{t_{transfer} = 16183 \text{ seconds} \approx 4.5 \text{ hours}}$

6) Plotting:

1. get the radius at each point of the orbit by using this equation and iterating through f values:

$r = \frac{a(1-e^2)}{1+e\cos f}$ ← (This can be used for all orbits given that we know the eccentricity)

2. To plot, we need x and y coordinates. So, convert r & f to x & y:




$\vec{r} = r\cos f\,\hat{x} + r\sin f\,\hat{y}$

Note we still need e of the transfer orbit:

$r = \frac{p}{1+e\cos f} \rightarrow r_1 = \frac{p_T}{1+e_T} \rightarrow e_T = \frac{p_T}{r_1} - 1 \rightarrow e_T = \frac{12996}{7930} - 1 \rightarrow e_T = 0.6393$

($r = r_1$ @ $f = 0$)

This also does not account for the orientation of the transfer orbit. However, we know that perigee is @ $r_1$ and apogee @ $r_2$ so we can just subtract 30° (of f) in the argument for the angle.

7) $\hat{e}, \hat{p}$ does not align between the departure and transfer orbit. This is evident on the figure. The angle between $\hat{e}_1$ and $\hat{e}_{transfer} = 30°$ because it is the same as $\Delta f$

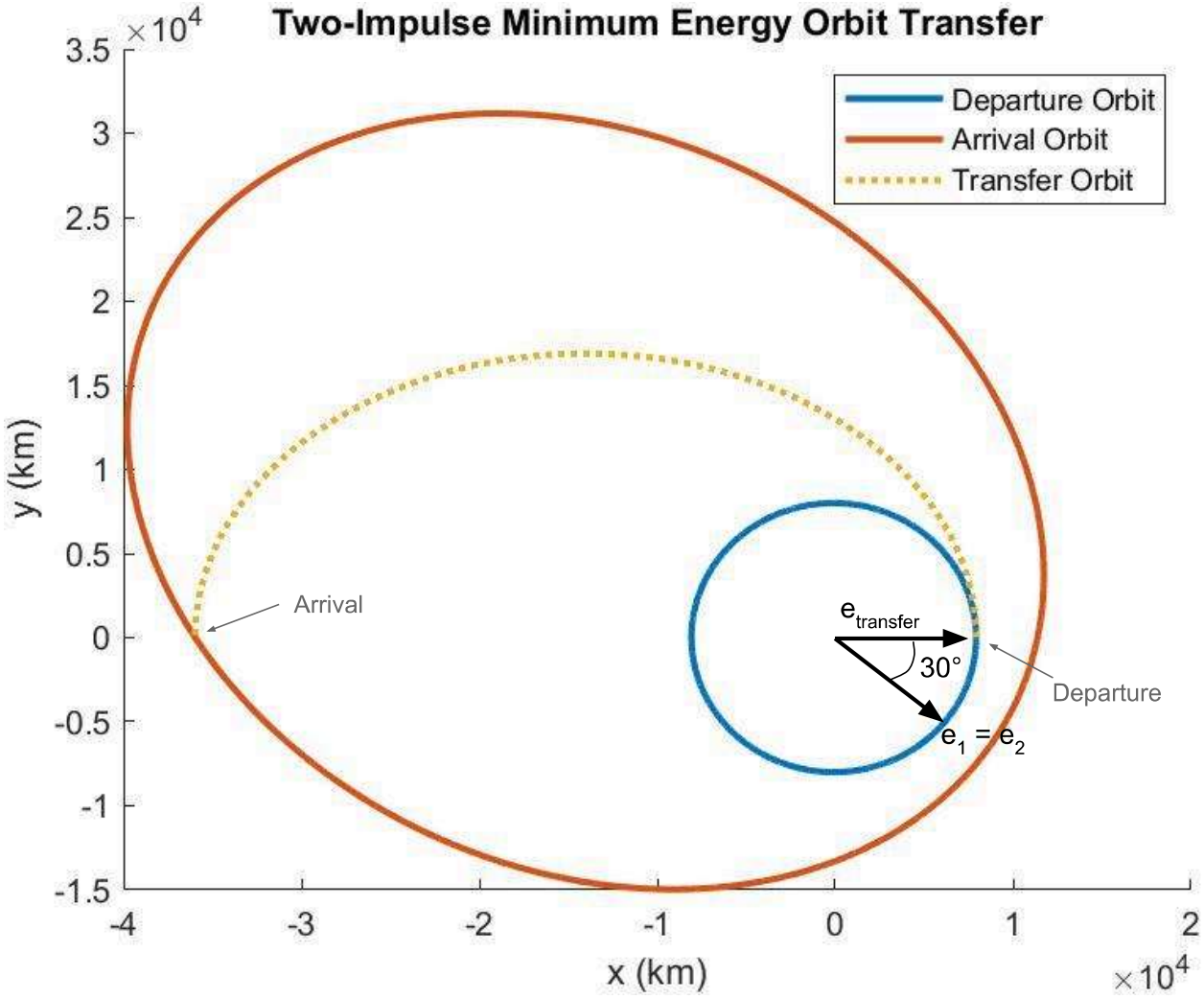



Figure 1. Labeled Plot of Problem 1's Orbit Transfer

```
% This is the code for HW2 - Problem 1
% By Nicholas Luis (PSU ID 930841391)

clear; clc; close all;

% Constants
MU = 398600; % km^3 / s^2
PI = 3.141592654;

% Departure Orbit
a1 = 8000; % km
e1 = 0.01;
```

```
f1 = 30; % degrees
r1 = getRadius(a1, e1, f1);
v1 = getVelo(r1, a1); % velocity at point 1 (on departure orbit)

% Arrival Orbit
a2 = 27000; % km
e2 = 0.6;
f2 = 210; % degrees
r2 = getRadius(a2, e2, f2);
v2 = getVelo(r2, a2); % velocity at point 2 (on arrival orbit)

% Transfer Orbit
delta_f = f2-f1;
a_T = get_aMin(r1, r2, delta_f);
p_T = getPT(a_T, r1, r2, delta_f);
e_T = (p_T / r1) -1;
v1T = getVelo(r1, a_T); % velocity at point 1 (on transfer orbit)
v2T = getVelo(r2, a_T); % velocity at point 2 (on transfer orbit)
t = PI * sqrt((a_T^3) / MU); % Transfer time

% Delta V
dV1 = abs(v1T - v1);
dV2 = abs(v2T - v2);

%% Plotting Values
fvec = 0:0.1:360; % Creates a list of true anomaly values to iterate through (in degrees)

% X Y values. Note that (0,0) will be located on the Earth
r1vec = (a1*(1-e1^2)) ./ (1+e1*cosd(fvec)); % List of the radii of the departure orbit
xvec1 = r1vec.*cosd(fvec); % List of the x coordinates of the departure orbit
yvec1 = r1vec.*sind(fvec); % List of the y coordinates of the departure orbit

rTvec = (a_T*(1-e_T^2)) ./ (1+e_T*cosd(fvec)); % List of the radii of the transfer orbit
xvecT = rTvec.*cosd(fvec); % List of the x coordinates of the transfer orbit
yvecT = rTvec.*sind(fvec); % List of the y coordinates of the transfer orbit

r2vec = (a2*(1-e2^2)) ./ (1+e2*cosd(fvec)); % List of the radii of the arrival orbit
xvec2 = r2vec.*cosd(fvec-30); % List of the x coordinates of the arrival orbit
yvec2 = r2vec.*sind(fvec-30); % List of the y coordinates of the arrival orbit
```

```
figure(1)
hold on
plot(xvec1, yvec1, LineWidth=2)
plot(xvec2, yvec2, LineWidth=2)
plot(xvecT(1:1801), yvecT(1:1801), ':', LineWidth=2) % Only plotting half of the transfer
orbit
title('Two-Impulse Minimum Energy Orbit Transfer')
xlabel("x (km)")
ylabel("y (km)")
legend('Departure Orbit', 'Arrival Orbit', 'Transfer Orbit')
hold off

exportgraphics(gca,"HW2_Problem1_Figure.jpg");
%% Functions
function radius = getRadius(a_input, e_input, f_input)
    p = a_input * (1 - (e_input^2));
    radius = p / (1+e_input*cosd(f_input));
end

function aMinT = get_aMin(r1_in, r2_in, df)
    % This function gets the semimajor axis of minimum energy transfer
    % orbit given r1, r2, and the change in f
    sqrtTerm = sqrt( r1_in^2 + r2_in^2 - 2*r1_in*r2_in*cosd(df) ) ;
    aMinT = 0.25 * (r1_in + r2_in + sqrtTerm);
end

function P_output = getPT(amin, r1_in, r2_in, df)
    % This funcion gets the semilatus rectum of a transfer orbit given r1,
    % r2, and the change in f
    k = r1_in*r2_in*(1-cosd(df));
    m = r1_in*r2_in*(1+cosd(df));
    l = r1_in + r2_in;

    P_output = (k*m - 2*amin*k*l)  / (4*amin*m - 2*amin*l*l);

end

function velo = getVelo(r_input, a_input)
    % This function gets the velocity using energy given MU, r, and a
    MU = 398600; % km^3 / s^2
```

```
    velo = sqrt(2 * MU * ( (1/r_input)  - (1 / (2*a_input)) ) );
end
```