Nicholas Luis (PSU ID: 930841381)
Aersp 450 - HW2

Problem 1)

given: $\mu = 398600 \frac{km^3}{s^2}$

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)

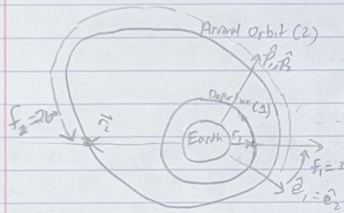
$a_{min,T} = \frac{1}{4}(r_1 + r_2 + \sqrt{r_1^2 + r_2^2 - 2r_1 r_2 \cos(\Delta f)})$, $r_1 = ?$, $r_2 = ?$, $\Delta f = ?$

* $f_2 - f_1$ 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 $\rightarrow r_1 = 7930.519695$ 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 Δf

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

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

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

$a_{min,T} = 21950.5$ km

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

2) $p = ?$ given $p = \frac{km - 2ak\ell}{4am - 2k\ell}$, $k = r_1 r_2(1 - \cos\Delta f) \rightarrow k = (7930)(35971)(1 - \cos(180°)) \rightarrow k = 5.70549 \times 10^8\ km^2$

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

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

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

3) ΔV_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 Lins
9309941391

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

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

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

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 ΔV to go from the transfer orbit to the arrival orbit.

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

$\cdot V_2 = 2.7202 \text{ km/s}$

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

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.

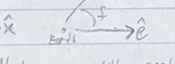
$T_p = 2\pi\sqrt{\frac{a^3}{\mu}} \rightarrow t_{transfer} = \pi\sqrt{\frac{a_T^3}{\mu}} \rightarrow t_{transfer} = 2\pi\sqrt{\frac{21451^3}{398600}} \text{ s} \rightarrow \boxed{t_{transfer} = 16183 \text{ seconds} \approx 4.5 \text{ hours}}$

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:

$r = \frac{p}{1+e\cos f} \rightleftarrows r_1 = \frac{p_T}{1+e_T} \rightarrow e_T = \frac{p_T}{r_1} - 1 \rightarrow e_T = \frac{12996}{7930} - 1 \rightarrow e_T = 0.6397$

$r = r_1 @ f = 0$

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° 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 $f_2 - \Delta f$.

This was noticed based on the geometry of the figure in part 2

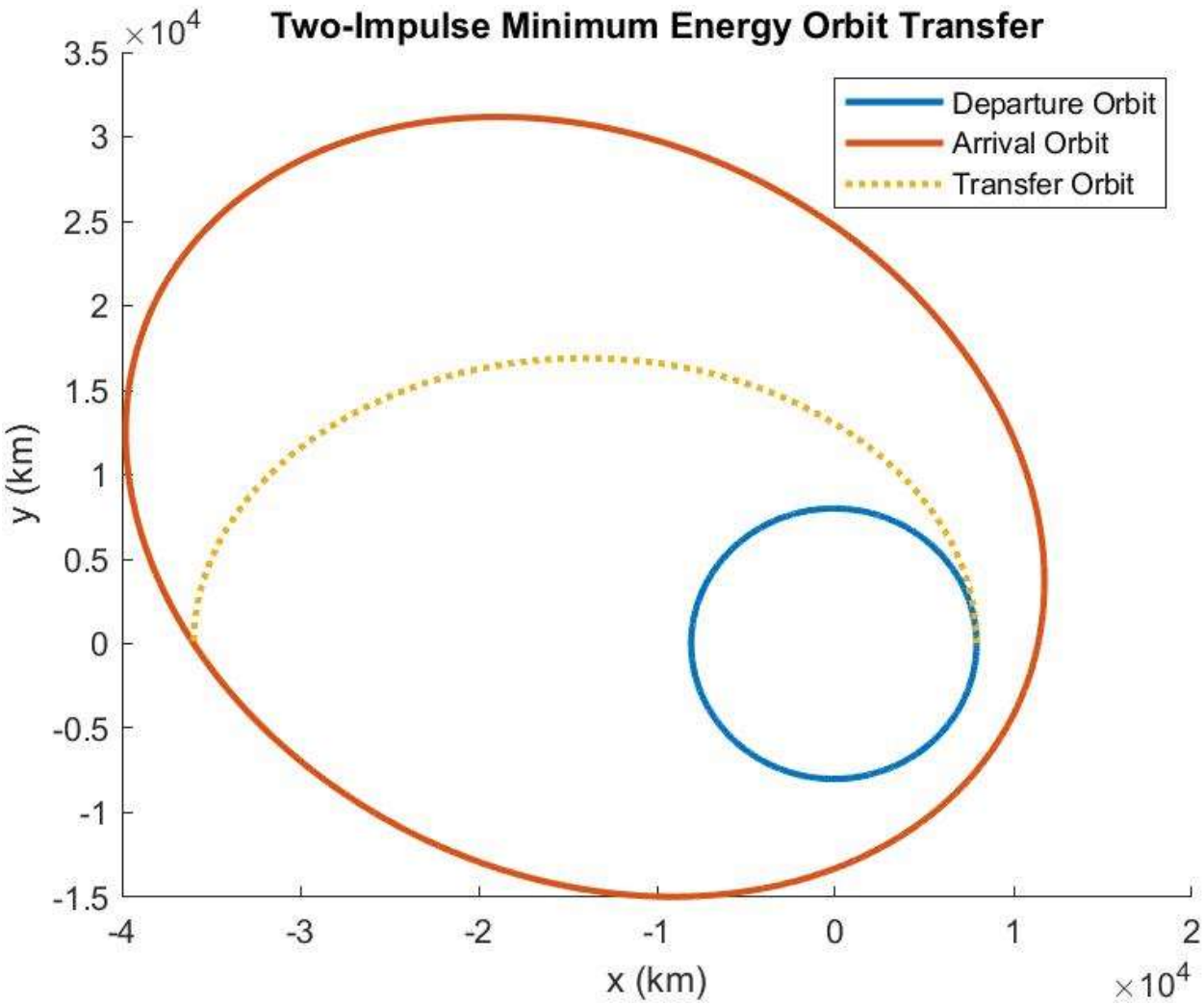


Figure 1 Labeled Plot of Problem 1's Orbit Transfer

```
% 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
```

Nicholas Luis
PSU ID 930941391

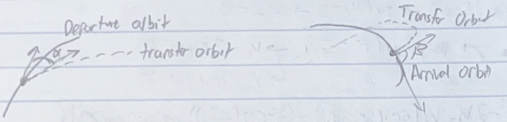
Problem 2)

given: Plane change maneuver:

Departure	$a_1 = 7000$ km	$e_1 = 0$	$I_1 = 60°$
Arrival	$a_2 = 70000$ km	$e_2 = 0$	$I_2 = 20°$

Assume: $\Omega = \omega = \text{constant } 0°$, Hohmann-like transfer (departure @ perigee, arrival @ apogee of T.O.)
Also means a_T & e_T can be calculated as if it were co-planar

Observations from given figure: ΔV_1 & ΔV_2 apply plane change and shape change simultaneously



A1) $a_T = ?$, $e_T = ?$

- Because both orbits are circular ($e_1 = e_2 = 0$), $a_1 = r_1 = 7000$km and $a_2 = r_2 = 70000$ km
- Assuming departure @ perigee & arrival @ apogee of the transfer orbit: $\Delta f = 180°$ because perigee and apogee are on opposite ends of the transfer orbit.

(assume elliptic T.O.)

$$a_T = \frac{1}{4}(r_1 + r_2 + \sqrt{r_1^2 + r_2^2 - 2r_1 r_2 \cos(\Delta f)}) \rightarrow \boxed{a_T = 38{,}500 \text{ km}}$$

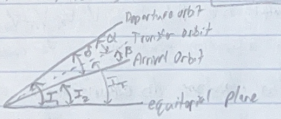
(same process as problem 1.6)

$$r = \frac{p}{1 + e\cos f} \rightarrow e_T = \frac{P_T}{r_1} - 1, \quad P_T = \frac{\text{km}^2\text{-dale}}{\text{dam}^2\text{-dale}},$$

$$\begin{cases} k = r_1 r_2 (1 - \cos\Delta f) \\ m = r_1 r_2 (1 + \cos\Delta f) \\ l = r_1 r_2 \end{cases} \Rightarrow \begin{cases} k = 9.8 \times 10^8 \text{ km}^2 \\ m = 0 \text{ km}^2 \\ l = 7.7 \times 10^5 \text{ km}^2 \end{cases}$$

$$\rightarrow P_T = 1.2727 \times 10^4 \text{ km} \rightarrow e_T = \frac{1.2727 \times 10^4 \text{ km}}{7 \times 10^3 \text{ km}} - 1 \rightarrow \boxed{e_T = 0.812}$$

A2) $I_T = ?$ given $\alpha = 20° = \Delta I_1$,



(Side view of orbit planes)

Observations:

- $\delta = \alpha + \beta = I_1 - I_2$
- $I_T = I_2 + \beta$
- $I_T = I_1 - \alpha \rightarrow I_T = (60°) - (20°) \rightarrow \boxed{I_T = 40°}$

- Calculating other angles for future use:

$\delta = I_1 - I_2 \rightarrow \delta = 60° - 20° \rightarrow \delta = 40°$

$\delta = \alpha + \beta \rightarrow \beta = \delta - \alpha \rightarrow \beta = 40° - 20° \rightarrow \beta = 20° = \Delta I_2$

A3) $\Delta V_1 = ?$, $\Delta V_2 = ?$

General equation: $\Delta V^2 = V_f^2 + V_i^2 - 2V_f V_i [\cos(\Delta\gamma) - (1 - \cos(\Delta I))\cos(\gamma_f)\cos(\gamma_i)]$,

↓ next page

Nicholas Luis
PSN ID: 9003941391

Problem 2 - A3 (continued)

- From Departure → Transfer Orbit: $\Delta I = \Delta I_1 = \alpha = 20°$
  - $V_i = V_1$
  - $V_f = V_{1T}$

  } can determine using energy equations

- $\gamma_1 = ?$ $\tan\gamma_1 = \frac{e_1 \sin f_1}{1 + e_1 \cos f_1}$. We can state that $\gamma_1 = 0$ because it is a function of the true anomaly. We can arbitrarily set the true anomaly to anything such as 0° because on a circular orbit the direction of $\hat{e}$ doesn't matter. $\therefore \sin(0°) = 0 \rightarrow \gamma_1 = \tan^{-1}(0) = 0°$
- $\gamma_{1T} = ?$ Departure @ perigee ($f = 0°$) $\rightarrow \sin(0°) = 0 \rightarrow \gamma_{1T} = 0$
- $V_1 = ?$ $\frac{1}{2}v^2 - \frac{\mu}{r} = -\frac{\mu}{2a} \rightarrow V = \sqrt{2\mu\left(\frac{1}{r} - \frac{1}{2a}\right)}$ ($a = r$ for a circular orbit) $\rightarrow V_1 = 7.5460 \frac{Km}{s}$, $V_{1T} = 10.1751 \frac{Km}{s}$

$\sqrt{\Delta V_1^2} = \text{sqrt}\{V_{1T}^2 + V_1^2 - 2V_{1T}V_1[\cos(\gamma_{1T} - \gamma_1) - (1 - \cos(\Delta I_1))\cos(\gamma_{1T})\cos(\gamma_1)]\}$ → (α, 1)

→ $\Delta V_1 = \sqrt{V_{1T}^2 + V_1^2 - 2V_{1T}V_1\cos(\alpha)}$ → $\boxed{\Delta V_1 = 4.0216 \frac{Km}{s}}$ (V_1, V_1)

- From Transfer Orbit → Arrival Orbit: $\Delta I_2 = \beta = -20°$ (from part A2)
- → using the same thought process as above: $\gamma_2 = 0°$, $\gamma_{2T} = 0°$, $V_2 = 2.3863 \frac{Km}{s}$, $V_{2T} = 1.0175 \frac{Km}{s}$
- → Plugging in these values into $\Delta V_2 = \sqrt{V_2^2 + V_{2T}^2 - 2V_2V_{2T}\cos(\beta)}$ → $\boxed{\Delta V_2 = 1.4719 \frac{Km}{s}}$

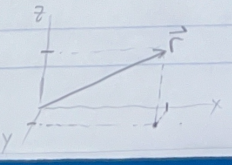
Part B

1) Steps:
   1. Iterate through a list of α values from −50 → +50 in increments of 3°
   2. At each iteration, calculate ΔV_i. Note that $V_1, V_2, V_{1T}, V_{2T}, \beta, \gamma_1, \gamma_{1T}, \gamma_2, \gamma_{2T}$ are all constant (I do this step using a function that does all the calculations.)
   3. At the end of all the iterations do element-wise vector addition and store the result for plotting.

2) - Plot α on x-axis & the total ΔV on the y-axis
   - Have Matlab find and print the minimum ΔV and the corresponding α based on its index

The result was a minimum ΔV of $4.105 \frac{Km}{s}$ @ $\alpha = 1°$. (should be 0° but it's skipped because we iterate in steps of 3°)

3) The process for plotting the orbits is the same as in problem 1.6 except we need to calculate the z-coordinates: $r_z = r \cdot \sin I$



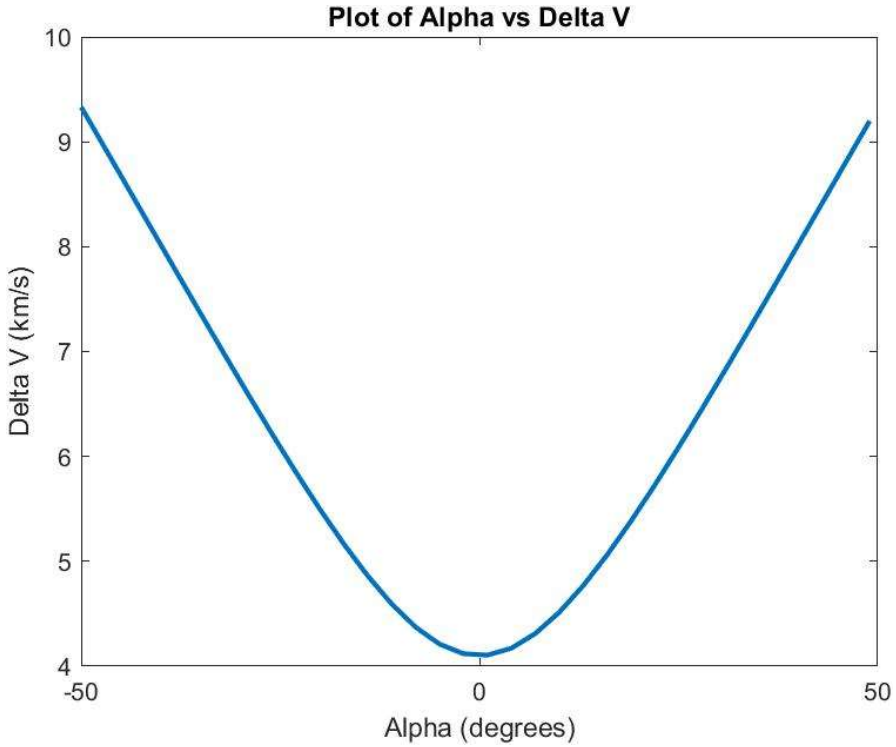


Figure 1 Determining minimum Delta V based on Alpha angle

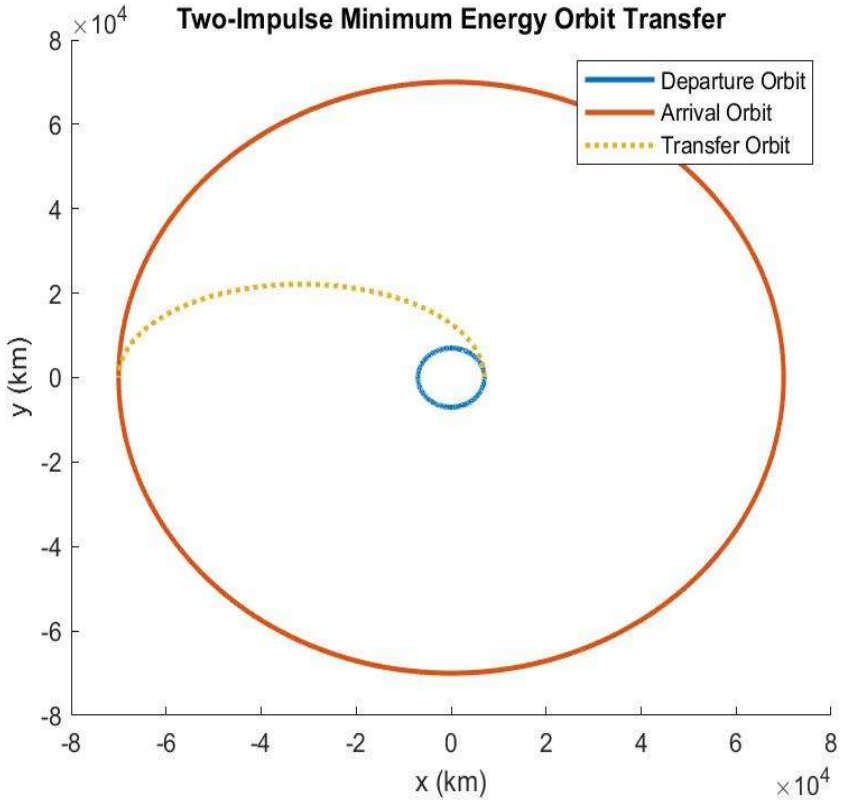


Figure 2 Plot of the Orbital Transfer using the Alpha for minimum Delta V

```
% This is the code for HW2 - Problem 2
% By Nicholas Luis (PSU ID 930841391)

clear; clc; close all;

%% Part A (used matlab as a calculator)
% Constants
MU = 398600; % km^3 / s^2
PI = 3.141592654;

% Departure Orbit
r1 = 7000; % km
e1 = 0;
a1 = r1;
v1 = getVelo(r1, a1);
I1 = 60;

% Arrival Orbit
r2 = 70000; % km
e2 = 0;
a2 = r2;
v2 = getVelo(r2, a2);
I2 = 20;

% Transfer Orbit
delta_f = 180;
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);
v2T = getVelo(r2, a_T);
IT = 40;

% Plane change angles
delta_I1 = -20; % AKA alpha
delta_I2 = -20; % AKA beta
gamma1 = 0;
gamma1T = 0;
gamma2T = 0;
gamma2 = 0;

% Delta V
deltaV1 = getDeltaV(v1, v1T, delta_I1, gamma1, gamma1T);
deltaV2 = getDeltaV(v2, v2T, delta_I2, gamma2, gamma2T);

%% Part B
% Question 1
alphaVec = -50:3:50; % Creates a list of alpha values to iterate through (in degrees)
deltaV1List = zeros(size(alphaVec)); % vector to store the delta V1's as we iterate
deltaV2List = zeros(size(alphaVec)); % vector to store the delta V2's as we iterate
for i=1 : length(alphaVec) % Iterates through each index
    deltaV1List(i) = getDeltaV(v1, v1T, alphaVec(i), gamma1, gamma1T);
```

```
    deltaV2List(i) = getDeltaV(v2, v2T, delta_I2, gamma2, gamma2T);
end
deltaVTotalList = deltaV1List + deltaV2List; % vector that stores the total delta V's

% Question 2
figure(1)
plot(alphaVec, deltaVTotalList, LineWidth=2)
title('Plot of Alpha vs Delta V')
xlabel("Alpha (degrees)")
ylabel("Delta V (km/s)")
exportgraphics(gca,"HW2_Problem2_Figure1.jpg");

[minVal, minIndex] = min(deltaVTotalList);
fprintf("The alpha for minimum delta V is: \t\t%d degrees\n", alphaVec(minIndex))
fprintf("This results in a minimum delta V of: \t%.3f km/s\n", minVal)

% Question 3
fvec = 0:0.1:360; % Creates a list of true anomaly values to iterate through (in
degrees)

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
%zvec1  = r1vec.*sind(I1);

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
%zvecT  = r1vec.*sind(IT);

r2vec = (a2*(1-e2^2)) ./ (1+e2*cosd(fvec)); % List of the radii of the arrival orbit
xvec2 = r2vec.*cosd(fvec); % List of the x coordinates of the arrival orbit
yvec2 = r2vec.*sind(fvec); % List of the y coordinates of the arrival orbit
%zvec2  = r1vec.*sind(I2);

figure(2)
hold on
%plot(xvec1, yvec1, zvec1, LineWidth=2)
%plot(xvec2, yvec2, zvec2, LineWidth=2)
%plot(xvecT(1:1801), yvecT(1:1801), zvecT(1:1801), ':', LineWidth=2) % Only plotting
half of the transfer orbit
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)")
%ylabel("z (km)")
legend('Departure Orbit', 'Arrival Orbit', 'Transfer Orbit')
hold off

exportgraphics(gca,"HW2_Problem2_Figure2.jpg");
```

```
%% Functions
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

function dvelo = getDeltaV(vi, vf, dI, gami, gamf)
    dvelo = sqrt( vf^2 + vi^2 - 2*vi*vf*( cosd(gamf-gami) - (1-
cosd(dI))*cos(gamf)*cos(gami) ) );
end
```

Problem 3: Lambert's Problem

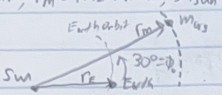
a) Given:
Need to get to mars on a T.O. who's T.O.F $\leq$ 183 days (1.58112×10^6 s)

$1 A.U. = 149.6 \times 10^6$ km

Assumptions: • circular, co-planar orbits ✓

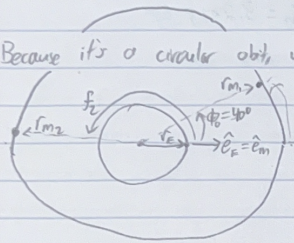
• $r_E = 1 A.U.$, $r_m = 1.524 A.U.$, $\mu_{Sun} = 1.327 \times 10^{11} \frac{km^3}{s^2}$

• $\phi_0 = 40°$



1) b) To find: Δf
c) Solution Process:
For circular orbits, $a = r$ and $e = 0$ → $a_E = r_E = 1 A.U.$, $e_E = 0$
$a_m = r_m = 1.524 A.U.$, $e_M = 0$

Because it's a circular obit, we can define $f = 0°$ in any direction. Will choose to align it w/ $\vec{r}_E$



$f_1 = 40°$
$f_2 = ?$
$f_2 = \omega_{orbit} \times \Delta t$, $\omega_{orbit} = \frac{2\pi}{T_p}$ and $\Delta t = T.O.F$
for circular orbit: $T_p = 2\pi\sqrt{\frac{r^3}{\mu}}$

→ $\Delta f = \frac{2\pi}{2\pi\sqrt{\frac{r_E^3}{\mu_S}}}(T.O.F) \rightarrow \Delta f = \sqrt{\frac{1.327\times10^{11}\ km^3/s^2}{(1.524\times149.6\times10^6\ km)^3}} \times (1.58112 \times 10^6\ sec) \rightarrow \boxed{\Delta f = 2.3712\ rad = 135.8624°}$

2) a) given: everything above
b) To find: a_T & e_T
c) Solution Process:

$a_{min_T} = a_T = \frac{1}{4}(r_E + r_m + \sqrt{r_E^2 + r_m^2 - 2 r_E r_m \cos\Delta f}) \rightarrow \boxed{a_T = 1.8219 \times 10^8\ km}$

e_T can be found using the same process as problem 1.6) & also problem 2A1)

$r = \frac{p}{1+e\cos f} \rightarrow e_T = \frac{p_T}{r_1} - 1$, $p_T = \frac{km - 2akl}{4am - 2akl^2}$, $\begin{cases} k = r_1 r_2 (1 - \cos\Delta f) \\ m = r_1 r_2 (1 + \cos\Delta f) \\ l = r_1 + r_2 \end{cases} \Rightarrow \begin{cases} k = 5.0585 \times 10^{16}\ km^2 \\ m = 9.6296 \times 10^{16}\ km^2 \\ l = 3.7739 \times 10^8\ km \end{cases}$

→ Plugging in those values → $p_T = 1.6683 \times 10^8$ km
∴ $e_T = \frac{1.6683\times10^8\ km}{1.496\times10^8\ km} - 1 \rightarrow \boxed{e_T = 0.1152}$

3) a) given: everything above
b) To find: ΔV_{Total}
c) Solution Process: $\Delta V_{Total} = |\Delta\vec{V}_1 + \Delta\vec{V}_2|$, $\Delta V_1 = ?$, $\Delta V_2 = ?$

• Finding ΔV_1 $\frac{1}{2}V^2 - \frac{\mu}{r} = -\frac{\mu}{2a} \rightarrow V = \sqrt{2\mu(\frac{1}{r} - \frac{1}{2a})}$ → $V_E = \sqrt{2\mu_S(\frac{1}{r_1} - \frac{1}{2a_1})} = 29.7831$ km/s
$V_{ET} = \sqrt{2\mu_S(\frac{1}{r_1} - \frac{1}{2a_T})} = 32.3371$ km/s

$\Delta V_1 = \sqrt{V_{ET}^2 + V_E^2 - 2V_{ET}V_E\cos(\gamma_{ET} - \gamma_E)}$, $\gamma_{ET} = ?$, $\gamma_E = ?$

↓ next page

Recall that for a circular orbit we can choose anywhere for $\hat{e}$ to point because there is no perigee to point to. Therefore, we can make $f=0$. Thus, $\gamma_E = 0°$ and $\gamma_M = 0°$.

~~$\gamma_{ET} = \tan^{-1}\left(\frac{e_T \sin f_{ET}}{1 + e_T \cos f_{ET}}\right)$~~

~~$r = \frac{h^2/\mu}{1 + e\cos f} \rightarrow f = \cos^{-1}\left(\frac{\frac{h^2}{\mu r} - 1}{e}\right) \rightarrow f_{ET} =$~~

$$\therefore \Delta V_E = \sqrt{V_{ET}^2 + V_E^2 - 2V_{ET}V_E} \rightarrow \Delta V_E = 2.5542 \ \frac{km}{s}$$

- The same process can be repeated for getting ΔV_2 (ΔV_M)

$$V = \sqrt{2\mu\left(\frac{1}{r} - \frac{1}{2a}\right)} \rightarrow \begin{cases} V_{MT} = 20.8738 \ m/s \\ V_M = 24.1255 \ Km/s \end{cases} \rightarrow \Delta V_M = 3.2517 \ \frac{Km}{s}$$

$$\Delta V = \sqrt{V_f^2 + V_i^2 - 2V_f V_i}$$

$$\Delta V_{Total} = \Delta V_E + \Delta V_M \rightarrow \boxed{\Delta V_{Total} = 5.8059 \ \frac{km}{s}}$$

4) Plotting Orbit

a) Given: everything above

b) To find: Plot

c) Solution Process:

- Convert radii and true anomaly to x & y coordinates for plotting. Same process as problem 1.6 or problem 2.B3:

$X_E = r_E \cos F$, $Y_E = r_E \sin F$ ← r_E, F are both known (and constant)

$X_m = r_m \cos F$, $Y_m = r_m \sin F$ ← r_m, F are both known (and constant)

$X_T = r_T \cos F$, $Y_T = r_T \sin F$ ← use energy to get r_T @ each f value

- These calculations are easily vectorized and plotted

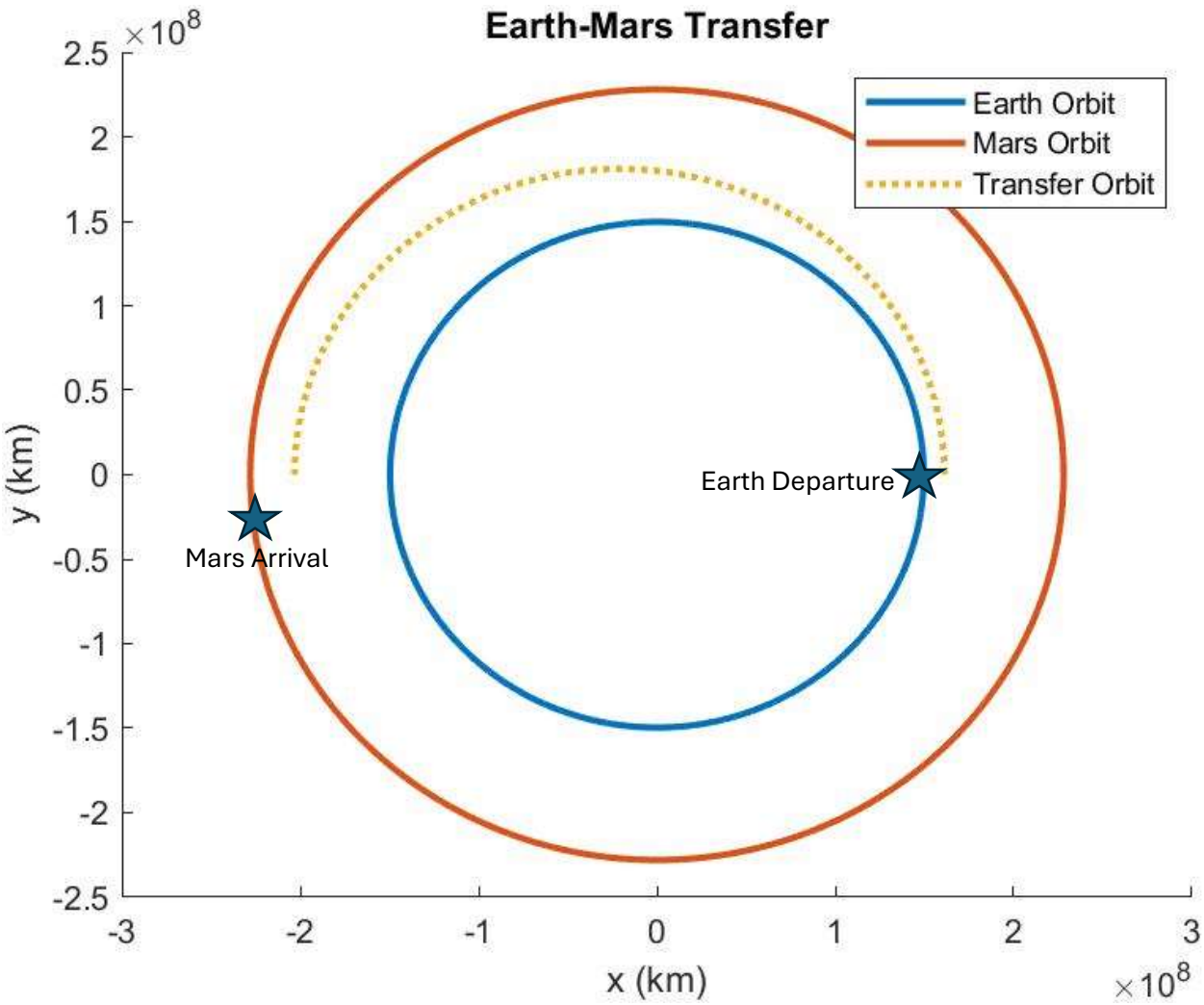


Figure 1 Transfer Between Earth and Mars

```
% This is the code for HW3 - Problem 3
% By Nicholas Luis (PSU ID 930841391)

clear; clc; close all;

% Constants
MU = 1.327*(10^11); % km^3 / s^2
PI = 3.141592654;
AU = 149.6*(10^6); % km

% Earth Orbit
rE = 1*AU;
aE = rE; % True for circular orbits
vE = getVelo(rE, aE);

% Mars Orbit
rM = 1.524*AU;
```

```
aM = rM; % True for circular orbits
vM = getVelo(rM, aM);

% Transfer Orbit
TOF = 1.58112*(10^6); % Time Of Flight in seconds
delta_f = 135.8624; % Degrees
a_T = get_aMin(rE, rM, delta_f);
p_T = getPT(a_T, rE, rM, delta_f);
e_T = (p_T / rE) - 1;
vET = getVelo(rE, a_T); % Velocity on the TO at inersection w/ Earth orbit
vMT = getVelo(rM, a_T); % Velocity on the TO at inersection w/ Mars orbit

% Delta V
delta_VE = getDeltaV(vE, vET, 0);
delta_VM = getDeltaV(vM, vMT, 0);

%% Plotting
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 Sun

xvec1 = rE.*cosd(fvec); % List of the x coordinates of the departure orbit
yvec1 = rE.*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

xvec2 = rM.*cosd(fvec-30); % List of the x coordinates of the arrival orbit
yvec2 = rM.*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('Earth-Mars Transfer')
xlabel("x (km)")
ylabel("y (km)")
legend('Earth Orbit', 'Mars Orbit', 'Transfer Orbit')
hold off

exportgraphics(gca,"HW2_Problem3_Figure1.jpg");

%% Functions

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 = 1.327*(10^11); % km^3 / s^2

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

function dvelo = getDeltaV(vi, vf, gamma)
    dvelo = sqrt( vf^2 + vi^2 - 2*vi*vf*cosd(gamma) );
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 fOut = getF(e, p, r)
    fOut = acosd((1/e)*((p/r)-1));
end

function gammaOut = getGamma(e, f, df)
    gammaOut = atand((e*sind(f)) / (1+e*cosd(f)));

    if (df > 180)
        gammaOut = abs(gammaOut);
    else
        gammaOut = - abs(gammaOut);
    end
end
```