Personal Project

Lundon Salley

EGR 115

Reaching the moon with a rocket launched from cape canaveral to Low Earth Orbit, and then executing a hohmann transfer to the moon.

refrences

https://web.stanford.edu/~cantwell/AA284A_Course_Material/Karabeyoglu%20AA%20284A%20Lectures/AA284a_Lecture7.pdf

http://www.nssc.ac.cn/wxzygx/weixin/201607/P020160718380095698873.pdf

https://dspace.mit.edu/bitstream/handle/1721.1/60691/16-07-fall-2004/contents/lecture-notes/d30.pdf

```
clear;
clc;
close all;
```

Define constants

```
G = 6.67*10^(-11);
g = 9.81;
M_earth = 5.972*10^(24);
VeE = 483.8; % velocity of earth
r_earth = 6.37814 * 10^(6);
mu = M_earth * G;
altitude_of_launch = 3; % cape canaveral has an altitude of 3m
omega_moon = 2*pi/2360250;
```

Polar launch, orbital alt of 400km, circular orbit

```
altitude = 400 * 10^3;
v_orbit = sqrt(G * M_earth / (r_earth+altitude));
Vtrans = sqrt(mu * ((2/(r_earth+altitude_of_launch)) -(1/(r_earth+altitude))));
Vco = sqrt(mu/(r_earth + altitude));
deltaVpoten = Vtrans-Vco;
deltaVgravloss = 2200;
deltaVdrag = 100; % negligable
```

delta v required to reach LEO from a polar launch (earth's rotation has no effect on launch deltaV)

```
deltaVtotal = v_orbit + deltaVpoten + deltaVgravloss + deltaVdrag
```

```
deltaVtotal =
1.0433e+04
```

Launch from cape canaveral

```
Latitude = 28.4740; % Cape canaveral
Az = 90; % due east flight
cosinclination = cosd(Latitude)*sind(Az);
deltaVearthRot = VeE * cosinclination; % this is deltaV GIVEN into the rocket by
earth's rotation
```

total deltaV required to achive LEO from cape canaveral

```
deltaVreq = deltaVtotal - deltaVearthRot
```

```
deltaVreq =
1.0007e+04
```

Begin Hohmann transfer to moon

```
orbit_leo = altitude + r_earth; % circular
orbit_moon = 384400 * 1000; % avg, assumed circular because .16% difference in alt
in reality
r1 = orbit_leo;
v_c1 = sqrt(mu/r1);
r2 = orbit_moon;
v_c2 = sqrt(mu/r2);
two_a = r1+r2;
E = -mu/(two_a);
vpi = sqrt(g*(r_earth^2)*((2/r1) - (2/(r1+r2))));
valpha = sqrt(g*(r_earth^2)*((2/r2) - (2/(r1+r2))));
h = r1*vpi;
e = sqrt(1+ ((2*E*h^2)/(mu^2)));
deltaVpi = vpi - v_c1;
deltaValpha = v_c2 - valpha;
```

Total deltaV required to execute a hohmann transfer from LEO to lunar orbit

```
deltaVTransfer = deltaVpi + deltaValpha
```

```
deltaVTransfer =
3.9193e+03
```

Total time to reach lunar orbit from LEO

```
TOF = pi*sqrt(((r1+r2)^3)/(8*mu)); % not optimal time of flight, 5 days, but fuel
and deltaV efficient
Time_of_flight_in_days = TOF/(60^2 * 24)
```

```
Time_of_flight_in_days =
4.9835
```

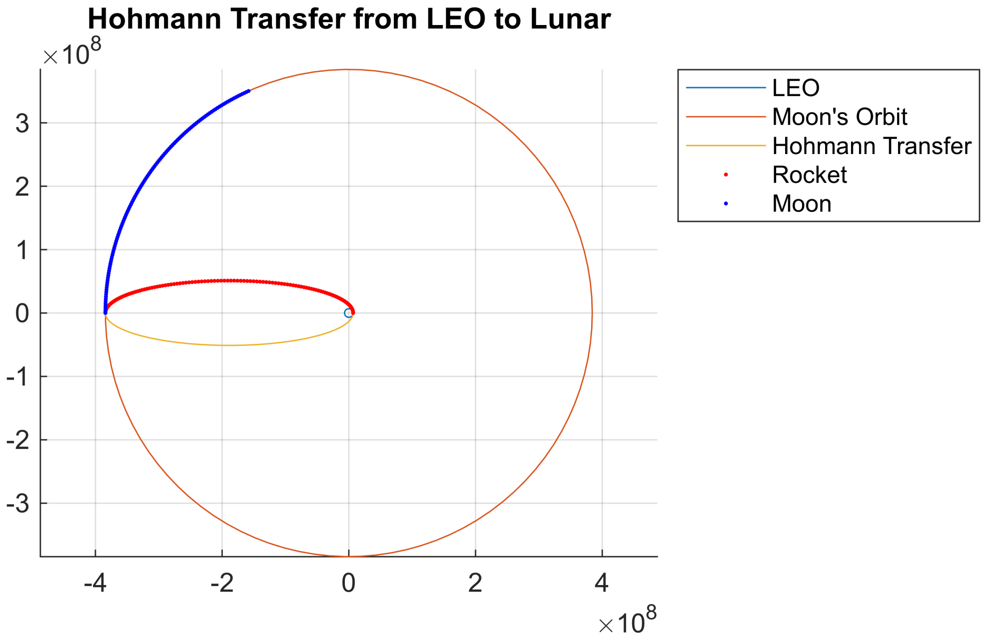
angular displacement of moon to allow for rocket to intersect with the moon

```
alpha = pi - omega_moon*TOF;
displacement_of_moon_in_degrees = rad2deg(alpha)
```

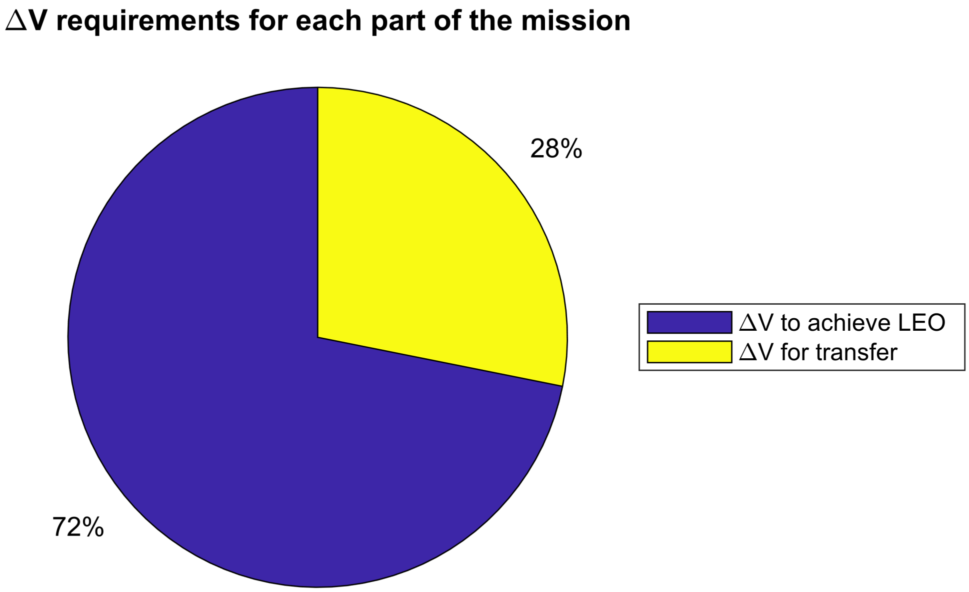
```
displacement_of_moon_in_degrees =
114.3268
```

Display the data

```
hold on
a = (r1+r2)/2;
b = a*sqrt(1-e^2);
theta = linspace(0,2*pi,100);
plot(r1*cos(theta),r1*sin(theta),'-')
plot(r2*cos(theta),r2*sin(theta),'-')
plot(a*cos(theta) - (a-r1),b*sin(theta),'-')

axis equal
grid on
legend('Location','northeastoutside');
title({"Hohmann Transfer from LEO to Lunar",""})

points = round(TOF/(60^2)); % plot a dot for ever hour of flight
theta = linspace(0,pi,points);
theta2 = linspace(alpha,pi,points);
x = a*cos(theta) - (a-r1);
x2 = r2 * cos(theta2);
y = b*sin(theta);
y2 = r2 * sin(theta2);
line_handle = plot(x(1), y(1), 'r.', 'MarkerSize', 5);
line_handle2 = plot(x2(1), y2(1), 'b.', 'MarkerSize', 5);

legend("LEO", "Moon's Orbit", "Hohmann Transfer", "Rocket", "Moon")

run = true;
if run
    for i = 1:points
        set(line_handle, 'XData', x(1:i), 'YData', y(1:i))
        set(line_handle2, 'XData', x2(1:i), 'YData', y2(1:i))
        drawnow;
        pause(0.01);
    end
end
hold off
```

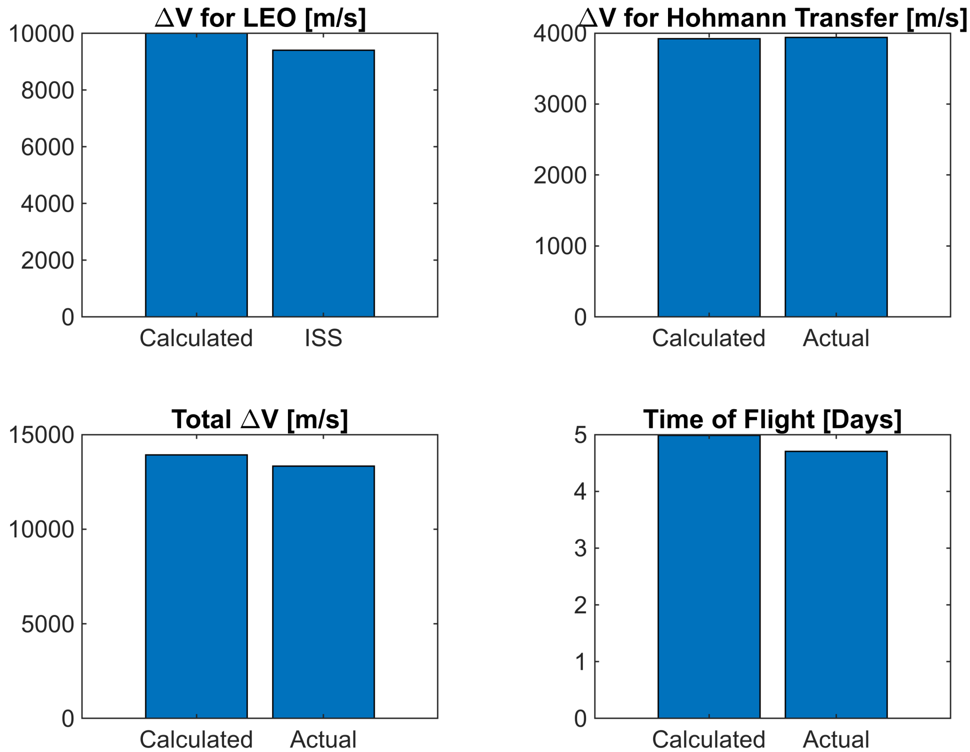



```
delV = [deltaVreq, deltaVTransfer];
labels = {'\DeltaV to achieve LEO','\DeltaV for transfer' };
pie(delV)
title("\DeltaV requirements for each part of the mission");
legend(labels)
legend('Location','eastoutside');

str = ["\DeltaV for LEO \approx "+deltaVreq/1000+" [km/s]","\DeltaV for Transfer
\approx "+deltaVTransfer/1000+" [km/s]"];
dim = [.5 .05 .5 .05];
annotation('textbox',dim,'String',str,'FitBoxToText','on');
t = tiledlayout(2,2);
```



ΔV for LEO ≈ 10.0073 [km/s]
ΔV for Transfer ≈ 3.9193 [km/s]

```
nexttile
bar(["Calculated","ISS"],[deltaVreq, 9.4e3])
title("\DeltaV for LEO [m/s]")
nexttile
bar(["Calculated","Actual"],[deltaVTransfer, 3.94e3])
title("\DeltaV for Hohmann Transfer [m/s]")
nexttile
bar(["Calculated","Actual"],[deltaVreq + deltaVTransfer, 9.4e3 + 3.94e3])
title("Total \DeltaV [m/s]")
nexttile
bar(["Calculated","Actual"],[Time_of_flight_in_days, 406340.8/(24*60^2)])
title("Time of Flight [Days]")
```



∆V for LEO [m/s]
10000
8000
6000
4000
2000
0
Calculated
ISS
∆V for Hohmann Transfer [m/s]
4000
3000
2000
1000
0
Calculated
Actual
Total ∆V [m/s]
15000
10000
5000
0
Calculated
Actual
Time of Flight [Days]
5
4
3
2
1
0
Calculated
Actual