

---

# Mission Design HW

```
peri_alt = 600; %km
apo_alt = 800; %km
i = 28.5*pi/180; %rad
Re = 6378; %km
mu = 3.986e5; %km3/s2
J2 = 0.00108263; %J2 of Earth

a = (peri_alt + 2*Re + apo_alt)/2; %sma
c = a - peri_alt - Re; % from center to focus
e = c/a; %eccentricity
vp = sqrt(2*mu/(peri_alt+Re) - mu/a); %km/s, velocity at perigee
% Apsidal rotation
n = sqrt(mu/(a*a*a));
dw_dt = 3*n*J2*Re*Re*(4-5*sin(i)*sin(i))/(4*a*a*(1-e*e)*(1-e*e));
% Nodal regression
dW_dt = -3*n*J2*Re*Re*cos(i)/(2*a*a*(1-e*e)*(1-e*e));

E_10_min = t_to_E(600, e, n);
f_10_min = acos((cos(E_10_min)-e)/(1-e*cos(E_10_min)));

% circularization
% One instantaneous burn
circ_alt = 700;
v_circ = sqrt(mu/(circ_alt+Re));
v_at_circ_alt = sqrt(2*mu/(circ_alt+Re) - mu/a);
delta_v_circ = v_at_circ_alt - v_circ;
% this is zero! meaning dV can't be in the velocity vec direction

f = acos((peri_alt + Re)*(1+e)/(circ_alt+Re)/e - 1/e); % true anom @ burn

fpa_burn = atan2(e*sin(f),1+e*cos(f));
delta_v_circ = 2*v_circ*tan(fpa_burn/2);
burn_direction = pi - (pi-fpa_burn)/2;

fprintf('Problem 1\n')
fprintf('Eccentricity: %f\n', e)
fprintf('Semi-major axis: %f km\n', a)
fprintf('Vp: %f km/s\n', vp)
fprintf('Apsidal rotation rate: %f deg/day\n', dw_dt*180/pi*86400)
fprintf('Nodal regression rate: %f deg/day\n', dW_dt*180/pi*86400)
fprintf('True anomaly at 10 minutes: %f deg\n', f_10_min * 180/pi)
fprintf('Circularization dV: %f km/s\n', delta_v_circ)
fprintf('Circularization dV direction: %f deg\n', burn_direction*180/pi)
fprintf('Circularization occurs at true anom: %f deg\n', f*180/pi)
fprintf('\n')
%TODO real dv?

% Problem 2
clearvars -except Re mu J2
```

---

```
alt = 270; %km
geo_alt = 35786; %km, Brown p. 102
% first dV, made at the Hohmann xfer's perigee
v_orig = sqrt(mu/(alt+Re));
a_hohmann = (alt + 2*Re + geo_alt)/2;
v_hoh_peri = sqrt(2*mu/(alt+Re)-mu/a_hohmann);
dv1 = v_hoh_peri - v_orig; %km/s, in direction of motion (being positive)

% Second dV, made at the Hohmann xfer's apogee
v_hoh_apo = sqrt(2*mu/(geo_alt+Re)-mu/a_hohmann);
v_final = sqrt(mu/(geo_alt + Re));
dv2 = v_final - v_hoh_apo; %km/s, in direction of motion.

fprintf('Problem 2\n')
fprintf('dV 1: %f km/s in velocity vector direction\n', dv1)
fprintf('dV 2: %f km/s in velocity vector direction\n', dv2)
fprintf('\n')

% Problem 3
clearvars -except Re mu J2

AU = 149.597870e6; %km, Brown p. 110
Earth_R = 1*AU;
Neptune_R = 30.011 * AU; %km, Brown p. 111
Earth_V = 29.77; %km/s
mu_sun = 132712440018; %km3/s2

a_hoh = (Earth_R + Neptune_R)/2;
v_hoh_peri = sqrt(2*mu_sun/(Earth_R)-mu_sun/a_hoh);
dV = v_hoh_peri-Earth_V;
t_xfer = 2*pi/sqrt(mu_sun/(a_hoh*a_hoh*a_hoh)) /2;

fprintf('Problem 3\n')
fprintf('dV: %f km/s \n', dV)
fprintf('Transfer Time: %f years\n', t_xfer/3600/24/365)
fprintf('\n')

% Problem 4
clearvars -except Re mu J2
alt_p = 41756; %km
e = 0.0061;
geo_alt = 35786; %km, Brown p. 102
% we are shrinking the orbit, so dv's will be in direction opp. to motion
a_hoh = (alt_p+2*Re+geo_alt)/2;
a_orig = (alt_p + Re)/(1-e);
v_p_orig = sqrt(2*mu/(alt_p + Re) - mu/a_orig);
v_hoh_apo = sqrt(2*mu/(alt_p + Re) - mu/a_hoh);
dv1 = v_hoh_apo-v_p_orig;

v_hoh_peri = sqrt(2*mu/(geo_alt + Re) - mu/a_hoh);
v_final = sqrt(mu/(geo_alt + Re));
dv2 = v_final - v_hoh_peri;
% TODOtry the other way?
```

```
fprintf('Problem 4\n')
fprintf('dV 1: %f km/s (anti-velocity-vector direction)\n', dv1)
fprintf('dV 2: %f km/s (anti-velocity-vector direction)\n', dv2)
fprintf('\n')

ra = a_orig*(1+e);
a_hoh = (ra+Re+geo_alt)/2;
v_a_orig = sqrt(2*mu/ra - mu/a_orig);
v_hoh_apo = sqrt(2*mu/(ra) - mu/a_hoh);
dv1 = v_hoh_apo-v_a_orig

v_hoh_peri = sqrt(2*mu/(geo_alt + Re) - mu/a_hoh);

% Problem 5
clearvars -except Re mu J2
v_esc = sqrt(2*mu/Re); %km/s, from the surface of the earth.
fprintf('Problem 5\n')
fprintf('V_esc: %f km/s\n', v_esc)
fprintf('\n')

% Problem 6
clearvars -except Re mu J2
% (using Laplace)
M_Mars = 639e21; %kg
M_Sun = 1.989e30; %kg
R_Mars = 227939100;
Rs = R_Mars*(M_Mars/M_Sun)^(2/5);
fprintf('Problem 6\n')
fprintf('Sphere of Influence: %f km (Using Laplace eqn)\n', Rs)
fprintf('\n')

% Problem 7
clearvars -except Re mu J2
a = 12500; %km
e = 0.472;
n = sqrt(mu/a/a/a);
f = 205*pi/180; %rad
E = acos((e+cos(f))/(1+e*cos(f)));
if f > pi
    E = 2*pi - E;
end
t = (E-e*sin(E))/n;
fprintf('Problem 7\n')
fprintf('Time of periapsis passage @ 205 deg: %f s\n', t)
fprintf('\n')

% Problem 8
clearvars -except Re mu J2
alt = 400; %km
i = 28.5*pi/180; %rad
v_i = sqrt(mu/(alt+Re));
dV = 2*v_i*sin(i/2);
fprintf('Problem 8\n')
```

```
fprintf('dV applied at node: %f km/s\n', dV)
fprintf('If descending at node, dV is toward the North Pole.\n')
fprintf('If ascending at node, dV is toward the South Pole.\n')
fprintf('\n')

% Problem 10
clearvars -except Re mu J2

% Orbit 1 conditions
rp = 700 + Re; % km
vp = 7.8; % km/s
a1 = mu*rp/(2*mu-vp*vp*rp);
e1 = 1-rp/a1;
n1 = sqrt(mu/a1/a1/a1);

% burn 1 conditions, new orbit
t_12 = 12*60;
E_12 = t_to_E(t_12, e1, n1);
f_12 = E_to_f(E_12, e1);
fpa_burn1 = atan2(e1*sin(f_12), 1+e1*cos(f_12));
r_burn1 = a1*(1-e1*e1)/(1+e1*cos(f_12));
v_burn1_pre = sqrt(2*mu/r_burn1 - mu/a1);
v_burn1_post = v_burn1_pre + 0.75;
a2 = mu*r_burn1/(2*mu-v_burn1_post*v_burn1_post*r_burn1);
H2 = r_burn1*v_burn1_post*cos(fpa_burn1);
e2 = sqrt(1-H2*H2/mu/a2);
n2 = sqrt(mu/a2/a2/a2);
f2_burn1_post = fpa_to_f(fpa_burn1, e2);
E2_burn1_post = f_to_E(f2_burn1_post, e2);
tpp_burn1_post = (E2_burn1_post-e2*sin(E2_burn1_post))/n2;

% burn 2 conditions, new orbit
tpp_burn2 = 30*60+tpp_burn1_post;
f2_burn2 = E_to_f(t_to_E(tpp_burn2, e2, n2), e2);
fpa_burn2 = atan2(e2*sin(f2_burn2), 1+e2*cos(f2_burn2));
r_burn2 = a2*(1-e2*e2)/(1+e2*cos(f2_burn2));
v_burn2_pre = sqrt(2*mu/r_burn2 - mu/a2);
v_burn2_post = v_burn2_pre + .5;
a3 = mu*r_burn2/(2*mu-v_burn2_post*v_burn2_post*r_burn2);
H3 = r_burn2*v_burn2_post*cos(fpa_burn2);
e3 = sqrt(1-H3*H3/mu/a3);
n3 = sqrt(mu/a3/a3/a3);
P3 = 2*pi/n3;
ra3 = a3*(1+e3);
rp3 = a3*(1-e3);
fprintf('Problem 10\n')
fprintf('Final semimajor axis: %f km\n', a3)
fprintf('Final eccentricity: %f \n', e3)
fprintf('Final perigee altitude: %f km\n', rp3-Re)
fprintf('Final apogee altitude: %f km\n', ra3-Re)
fprintf('Final period: %f s\n', P3)
```

*Problem 1*  
*Eccentricity: 0.014128*

Semi-major axis: 7078.000000 km  
Vp: 7.611138 km/s  
Apsidal rotation rate: 9.906275 deg/day  
Nodal regression rate: -6.084576 deg/day  
True anomaly at 10 minutes: 37.423888 deg  
Circularization dV: 0.106029 km/s  
Circularization dV direction: 90.404759 deg  
Circularization occurs at true anom: 90.809518 deg

*Problem 2*

dV 1: 2.434361 km/s in velocity vector direction  
dV 2: 1.469961 km/s in velocity vector direction

*Problem 3*

dV: 11.667205 km/s  
Transfer Time: 30.549508 years

*Problem 4*

dV 1: -0.105518 km/s (anti-velocity-vector direction)  
dV 2: -0.100013 km/s (anti-velocity-vector direction)

dv1 =

-0.0964

*Problem 5*

V\_esc: 11.179989 km/s

*Problem 6*

Sphere of Influence: 576189.393044 km (Using Laplace eqn)

*Problem 7*

Time of periapsis passage @ 205 deg: 9204.005153 s

*Problem 8*

dV applied at node: 3.775318 km/s  
If descending at node, dV is toward the North Pole.  
If ascending at node, dV is toward the South Pole.

*Problem 10*

Final semimajor axis: 12082.486372 km  
Final eccentricity: 0.339653  
Final perigee altitude: 1600.638305 km  
Final apogee altitude: 9808.334440 km  
Final period: 13217.389617 s

*Published with MATLAB® R2013b*