Table of Contents


```
%-------- HW 5 MATLAB code --------%
% Romeo Perlstein, section 0101 %

% IT'S A NEW WEEK - HOPEFULLY I CAN KEEP UP THE WORK AND NOT FALL BEHIND!
```

Q1

Find deltaV for Mercury to Jupiter transfer using conic sections

```
% Givens:
mew_mercury = 22031.868551; % km^3/s^2 - FROM JPL
mew_saturn = 126712764.1; % km^3/s^2 - FROM JPL
radius_planet_mercury = 2440.5; % km - from NASA fact sheet
r_craft_mercury = 400+radius_planet_mercury; % km
r_mercury = 57.909*10^6; % semi-major axis - from NASA fact sheet

radius_planet_saturn = 60268; % km - from NASA fact sheet
r_craft_saturn = 10000+radius_planet_saturn; % km
r_saturn = 1432.041*10^6; % semi-major axis - from NASA fact sheet

% since we are assuming circular orbits, we need to find the orbit
% velocity of both planets!
mew_sun = 132712*10^6; % from NASA fact sheet
v_mercury = sqrt(mew_sun/r_mercury);
v_saturn = sqrt(mew_sun/r_saturn);

% find velocities of orbits of spacecraft
v_initial_craft_mercury = sqrt(mew_mercury/r_craft_mercury);
v_final_craft_saturn = sqrt(mew_saturn/r_craft_saturn);

% Find the velocity to transfer from mercury to saturn
v_transfer_peri = sqrt(2*((mew_sun/r_mercury) - (mew_sun/(r_mercury
+r_saturn))));
v_transfer_apo = sqrt(2*((mew_sun/r_saturn) - (mew_sun/(r_mercury
+r_saturn))));

% Now, lets get the escape velocity from mercury
v_escape_mercury = v_transfer_peri - v_mercury;
v_escape_hyperbola_mercury = sqrt(2*((mew_mercury/r_craft_mercury) +
 ((v_escape_mercury^2)/2)));

% Now we can find the delta V to get to Saturn
deltaV1 = v_escape_hyperbola_mercury - v_initial_craft_mercury;
```

```
% Now do saturn
v_escape_saturn = v_saturn - v_transfer_apo;

% MAKING ASSUMPTION THAT HYPERBOLA PERIAPSIS IS SAME AS PARKING ORBIT
% PERIAPSIS, A. BECAUSE THE PROBLEM DOESN'T SAY WE CAN'T AND B. BECAUSE I
% WOULD NOT BE ABLE TO SUBMIT THE HW ON TIME
v_escape_hyperbola_saturn = sqrt(2*((mew_saturn/r_craft_saturn) +
 ((v_escape_saturn^2)/2)));
deltaV2 = v_escape_hyperbola_saturn - v_final_craft_saturn;

% now get the supplementary info
mass_mercury = .3301*10^24; % kg
mass_saturn = 568.32*10^24; % kg
mass_sun = 1998500*10^24; % kg
SOI_mercury = r_mercury*(mass_mercury/mass_sun)^(2/5); % km - Matches with
 Wikipedia!
SOI_saturn = r_saturn*(mass_saturn/mass_sun)^(2/5); % km - Matches with
 Wikipedia!
deltaV_total = deltaV1+deltaV2;

% Find TOF, assuming ellipse:
a = (r_mercury+r_saturn)/2;
e = 1-(r_mercury/a);
E = pi;
t = sqrt((a^3)/mew_sun)*(pi-0.9223*sin(pi)); % seconds!!!
```

Q2

do a bunch of stuff I don't have time to finish :/ given:

```
e2 = 1.2;
rp2 = 5380;
a2 = 1-(rp2/e2);
mew_mars = 0.042828 *10^6; % km^3/s^2 - from NASA fact sheet
% assuming circular orbit
r_mars = 227.956 * 10^6;
v_mars = sqrt(mew_sun/r_mars);
```

a

assuming velocity of planet is same direction as flyby

Published with MATLAB® R2022b