Homework #5 – ASEN 5050
Due: Thursday, 10/8/2015

Name: John Clouse

Note: Use Appendix D of the book for all constants not given in the problem.

Please put your answers in this Answer Sheet (or something similar), and then show all work as usual below.

1. **(30 pts):**

Planet	Orbital Radius (km)	ΔV_1 (km/s)	ΔV_2 (km/s)	Total ΔV (km/s)	Transfer Duration (years)
Earth	149598023	----	----	----	----
Venus	108208601	-2.495	-2.707	5.202	0.400
Mars	227939186	2.945	2.649	5.594	0.709
Jupiter	778298361	8.793	5.643	14.436	2.731

2. **(20 pts):** Earth–Sun–Mars phasing angle: -44.34 deg

   Synodic period: 2.14 years

3. **(20 pts):** ΔV_1: 0.146 km/s ΔV_2: -0.146 km/s

   Total ΔV: 0.291 km/s Transfer Duration: 14847 sec

4. **(30 pts):**

Transfer Option	ΔV_1 (km/s)	ΔV_2 (km/s)	Total ΔV (km/s)
5 minutes	1.292e-3	1.205e-3	2.497e-3
30 minutes	0.450e3	0.109e-3	0.558e-3

Table of Contents

HW5 Problem 1

```
fprintf('\n');
clearvars -except function_list hw_pub toolsPath
close all
CelestialConstants; % import useful constants

% quick function to compute velocity on the fly:
visviva = @(r,a) sqrt(2*Sun.mu/(r) - Sun.mu/a);
```

Earth->Venus

```
fprintf('Earth->Venus\n');
earth_v = visviva(Earth.a, Earth.a);
Venus_v = visviva(Venus.a, Venus.a);
a_xfer = (Earth.a + Venus.a)/2;
dv1 = visviva(Earth.a, a_xfer) - earth_v
dv2 = Venus_v - visviva(Venus.a, a_xfer)
dv_tot = abs(dv1) + abs(dv2)
T = pi*sqrt(a_xfer^3/Sun.mu)/day2sec/365.25
```

```
Earth->Venus

dv1 =

   -2.4954


dv2 =

   -2.7066


dv_tot =

    5.2020


T =

    0.3999
```

Earth->Mars

```
fprintf('Earth->Mars\n');
earth_v = visviva(Earth.a, Earth.a);
Mars_v = visviva(Mars.a, Mars.a);
a_xfer = (Earth.a + Mars.a)/2;
dv1 = visviva(Earth.a, a_xfer) - earth_v
dv2 = Mars_v - visviva(Mars.a, a_xfer)
dv_tot = abs(dv1) + abs(dv2)
T = pi*sqrt(a_xfer^3/Sun.mu)/day2sec/365.25
```

```
Earth->Mars

dv1 =

    2.9447


dv2 =

    2.6489


dv_tot =

    5.5936


T =

    0.7087
```

Earth->Jupiter

```
fprintf('Earth->Jupiter\n');
earth_v = visviva(Earth.a, Earth.a);
Jupiter_v = visviva(Jupiter.a, Jupiter.a);
a_xfer = (Earth.a + Jupiter.a)/2;
dv1 = visviva(Earth.a, a_xfer) - earth_v
dv2 = Jupiter_v - visviva(Jupiter.a, a_xfer)
dv_tot = abs(dv1) + abs(dv2)
T = pi*sqrt(a_xfer^3/Sun.mu)/day2sec/365.25
```

```
Earth->Jupiter

dv1 =

    8.7926
```

```
dv2 =

    5.6432


dv_tot =

   14.4358


T =

    2.7308
```

Published with MATLAB® R2013b

HW5 Problem 2

```
fprintf('\n');
clearvars -except function_list hw_pub toolsPath
close all
CelestialConstants; % import useful constants

a_xfer = (Earth.a + Mars.a)/2;
% Find the Lead Angle first:
n_mars = sqrt(Sun.mu/Mars.a^3);
T_xfer = pi*sqrt(a_xfer^3/Sun.mu);
lead_angle = n_mars*T_xfer;
phase_angle = lead_angle - pi;
phase_angle*180/pi

% Synodic period
n_earth = sqrt(Sun.mu/Earth.a^3);
syn_period = 2*pi/(n_earth-n_mars)/day2sec/365.25
```

```
ans =

  -44.3441


syn_period =

    2.1354
```

Published with MATLAB® R2013b

HW5 Problem 3

```
fprintf('\n');
clearvars -except function_list hw_pub toolsPath
close all
CelestialConstants; % import useful constants

phase_angle = 30*pi/180;
h = 6e3; %km
a = Earth.R + h;
n = sqrt(Earth.mu/a^3);
t_phase = (2*pi + phase_angle)/n
t_phase/3600;
a_phase = ((t_phase/(2*pi))^2*Earth.mu)^(1/3);
dv1 = sqrt(2*Earth.mu/a-Earth.mu/a_phase) - sqrt(Earth.mu/a)
dv2 = sqrt(Earth.mu/a) - sqrt(2*Earth.mu/a-Earth.mu/a_phase)
dv_tot = abs(dv1) + abs(dv2)
```

```
t_phase =

   1.4847e+04


dv1 =

    0.1456


dv2 =

   -0.1456


dv_tot =

    0.2911
```

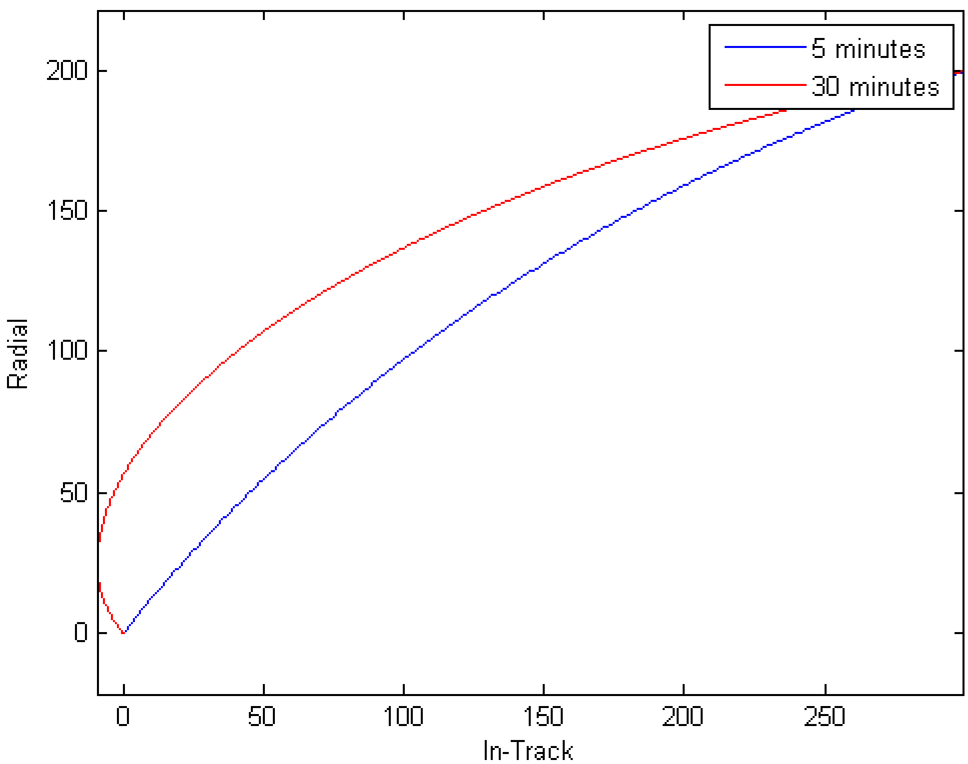
Published with MATLAB® R2013b

HW5 Problem 4

```
fprintf('\n');
clearvars -except function_list hw_pub toolsPath
close all
CelestialConstants; % import useful constants

t = 300; % sec
n = sqrt(Earth.mu/(400+Earth.R)^3);
rel_state = [200;300;50;0.1;0;-0.1]; %m
final_rel_state = [0;0;0;0;0;0]; %m
figure
color = 'b';

for t = [300, 30*60]
% Can take the state transition matrix, divide it into sub-matrices
% and solve for the required initial velocities.
% Subtract the contribution from initial pos to the final pos, from the
% final pos.
% Multiply by the inverse of the contribution of the initial vel to the
% final pos.
fprintf('t = %d sec:\n',t);
STM = CWHillSTM(n,t);
req_vel_init = inv(STM(1:3, 4:6))*...
    (final_rel_state(1:3)-STM(1:3, 1:3)*rel_state(1:3));
dv1 = norm(req_vel_init - rel_state(4:6))
final_state_preburn = STM*[rel_state(1:3);req_vel_init];
dv2 = norm(-final_state_preburn(4:6))
dv_tot = dv1 + dv2
for ii = 1:t
    plot_state = CWHillSTM(n,ii)*[rel_state(1:3);req_vel_init];
    x(ii) = plot_state(1);
    y(ii) = plot_state(2);
end
plot(y,x,color)
axis equal
hold on
color = 'r';
end
xlabel('In-Track')
ylabel('Radial')
legend('5 minutes', '30 minutes')
```

```
t = 300 sec:

dv1 =

    1.2918


dv2 =
```

```
    1.2054


dv_tot =

    2.4972

t = 1800 sec:

dv1 =

    0.4499


dv2 =

    0.1085


dv_tot =

    0.5584
```



Published with MATLAB® R2013b

CelestialConstants

Table of Contents

Description

All sorts of constants for orbital mechanics purposes

```
fcnPrintQueue(mfilename('fullpath')) % Add this code to code app
```

Earth

```
Earth.mu = 3.986e5; %km3/s2
Earth.R = 6378; %km
Earth.a = 149598023; %km

%%Sun
Sun.mu = 1.32712428e11; %km3/s2
```

Venus

```
Venus.a = 108208601; %km
```

Mars

```
Mars.a = 227939186; %km
```

Jupiter

```
Jupiter.a = 778298361; %km
```

Celestial units

```
au2km = 149597870.7;
```

Physical constants

```
day2sec = 86400; % sec/day
```

Published with MATLAB® R2013b

```
function STM = CWHillSTM(w,t)
fcnPrintQueue(mfilename('fullpath')) % Add this code to code app

s = sin(w*t);
c = cos(w*t);
x_t = [4-3*c, 0, 0, s/w, -2/w*c+2/w, 0];
y_t = [6*s-6*w*t, 1, 0, 2/w*c-2/w, 4/w*s-3*t, 0];
z_t = [0, 0, c, 0, 0, s/w];
xd_t = [3*w*s, 0, 0, c, 2*s, 0];
yd_t = [6*w*c-6*w, 0, 0, -2*s, 4*c-3, 0];
zd_t = [0, 0, -w*s, 0, 0, c];

STM = [x_t; y_t; z_t; xd_t; yd_t; zd_t];
```

Published with MATLAB® R2013b

```
function fcnPrintQueue( filename )
global function_list;
if exist('function_list', 'var')
    file_in_list = 0;
    for idx = 1:length(function_list)
        if strcmp(function_list(idx), filename);
            file_in_list = 1;
            break
        end
    end
    if ~file_in_list
        function_list = [function_list, filename];
    end
end
end
```

Published with MATLAB® R2013b