A picture containing fireworks, dark, water, flying

Description automatically generated

College of Engineering

School of Aeronautics and Astronautics

AAE 532

Orbital Mechanics

PS 6

In-Plane Maneuver

*Author:*

Tomoki Koike

*Supervisor:*

K. C. Howell

October 16th, 2020 Friday

Purdue University

West Lafayette, Indiana

**Problem 1**: Assume a relative two-body model and a space vehicle that is currently in a Mars orbit with and . The spacecraft is currently located at at time . A single in-plane adjustment will be employed to circularize the orbit.

1. At what true anomaly values does ? (Note that two locations exist!) Select the location that is the earliest opportunity after to reach . let this time be . Determine at this location.

From the radius of periapsis and apoapsis we can find the orbital parameters to define the orbit.

Since, , the semi-major axis becomes

Next, from the periapsis and apoapsis we can calculate the eccentricity

Then, the semi-latus rectum, becomes

When the gravitational parameter, , the specific angular momentum becomes

Then, at we can find the true anomaly by

The earliest to is the positive value. Thus,

The velocity at this point is

The flight path angle is positive because at time it is ascending,

Using the equation below the eccentric anomaly becomes,

Using, the equation on notes G7 we can find the elapsed time as the following

We check if this is an elliptical orbit

1. Sketch the orbit. Mark the usual quantities at the time ; also add appropriate unit vectors .

Chart, diagram

Description automatically generated

1. What is the “wait time” till the maneuver ?

Similar to , we can find using .

Then, like in the previous part (a), where we found the time difference for point 1, we can find,

Thus, the wait time becomes

1. Determine after the maneuver. Compute the required maneuver . Recall that . [Include VECTOR diagrams!!!!]

Since after the maneuver the spacecraft is going to enter a circular orbit, we know that the radius of this circle is going to be equivalent with the distance at the point at time . Hence,

And the velocity after the maneuver is going to be equal to the circular velocity,

Now, since the new velocity is tangent to a circle the direction of it is parallel to , its vector form is

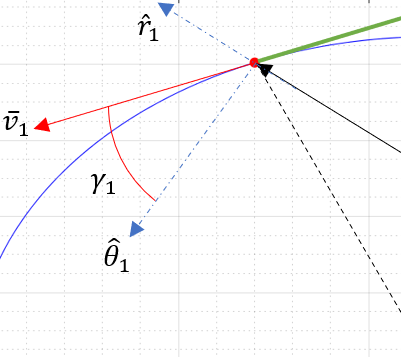
And because can be expressed as

Thus, becomes

Thus,

Then the angle can be found by the dot product rule and geometry (negative sign since it is moving away from the center of orbit),

The angle alpha is equal to the flight path angle since the flight path angle is the angle between and .

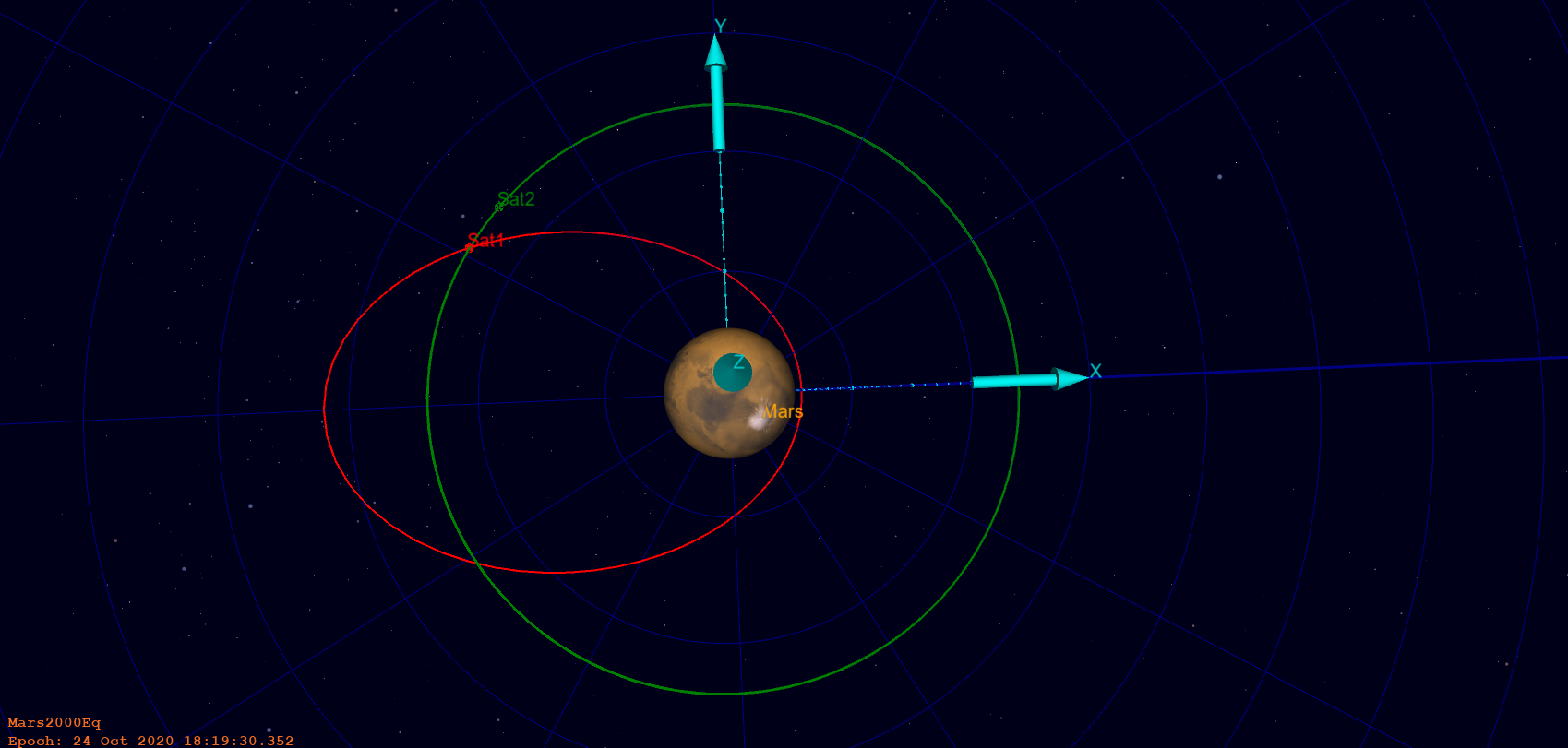


1. Plot the old and new orbits on the same figure using your MATLAB script. On the plot, mark .

A picture containing chart

Description automatically generated

1. Plot the two orbits in GMAT using Mars as the central body. At the maneuver time, use a report to list in each orbit at the maneuver time. Choose a convenient set of unit vectors (coordinate frame). Subtracting the velocity vectors should yield your =. Does it?



From the GMAT report we have the following results for the position and velocity vectors at the maneuver point for the two orbits. The raw data is the following



Elliptical orbit:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | -1.3e+4 km | 8044 km |
|  | -1.37 km/s | -0.4173 km/s |

Circular orbit:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | -1.3e+4 km | 8044 km |
|  | -0.8808 km/s | -1.4230 km/s |

Thus, the velocity difference from the GMAT results become

Now since

where , therefore

And,

This agrees with our result in part (d). Thus, we have verified our result.

**Problem 2**: given a two-body model, vehicle is successfully launched into an Earth orbit with and . A single in-plane maneuver will be implemented when . Let the maneuver be defined as .

1. Determine at the moment of maneuver point.

Firstly, the semi latus rectum is

Then, we can find the magnitude of

Since, , the magnitude of the velocity vector is

The specific angular momentum is

The flight path angle is

The position vector can be expressed by its magnitude and true anomaly

or

Then the velocity vector becomes

or in the frame

1. Express the maneuver in both and unit vectors. Also determine the maneuver in the VNB set of coordinates.
2. Prepare any VECTOR diagrams!!!! Determine in the new orbit immediately after the maneuver.

The vector diagram at the maneuver point is

From the cosine law

The corresponding flight path angle is

The vector of becomes,

Thus, the velocity vector of the difference of the velocity vectors become

In unit vectors the two become

In the VNB coordinate system,

and is normal to on the same plane depicted in the vector diagram on the previous page.

And finally,

1. To determine the impact that such a maneuver creates on the orbital characteristics, compute the following characteristics of the new orbit: .

Now, the semi major axis is

The eccentricity becomes from notes JS3

Then the semi latus rectum becomes

Then the specific angular momentum becomes

Then the period becomes

The specific energy becomes

We know from the previous problem that the flight path angle is

The true anomaly is positive since the flight path angle is positive, and therefore,

The eccentric anomaly is computed by

Then, the time elapsed from periapsis is

The radius of periapsis is

Finally,

1. Plot the new and the old orbits in Matlab on the same figure using your Matlab script. On the plot, mark . Also indicate the new and old lines of apsides and the shift, i.e., . Is it positive or negative? Why?

Chart

Description automatically generated

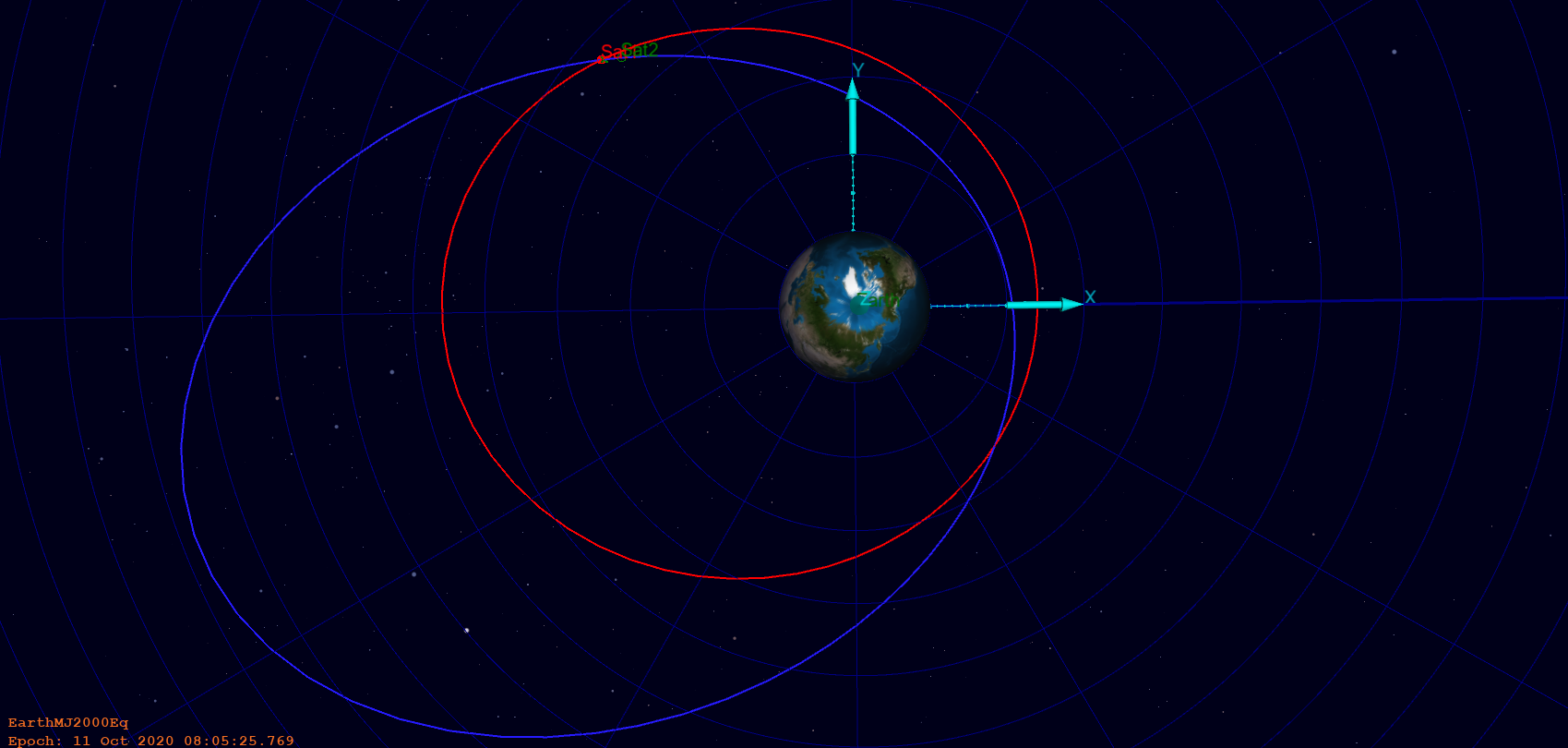
New line of apsides

Old line of apsides

is a positive value because this value is the difference between the true anomaly of the new orbit and the old orbit, and that calculation yields a positive value. Also, this value indicates the angle in which the line of apsides rotates and from the old orbit to the new one the line of apsides rotates in a positive angle with respect to the vector which is coming out of the page.

1. Bonus 🡪 You can use GMAT in two ways to check the maneuver and the new orbit. Use either method to check your results:

Use a start date October 10, 2020 12:00:00 to propagate the satellites. Put in the old and new orbits with 2 satellites and compare the velocities at the intersection point to assess if the difference equals the required ;



Sat1 in the figure orbits the old orbit and Sat2 orbits the new one.

The summary of Sat1 and Sat2 at the maneuver point is the following

Propagate Command: Propagate1

Spacecraft : Sat1

Coordinate System: EarthMJ2000Eq

Time System Gregorian Modified Julian

----------------------------------------------------------------------

UTC Epoch: 10 Oct 2020 23:15:55.660 29133.4693942075

TAI Epoch: 10 Oct 2020 23:16:32.660 29133.4698224483

TT Epoch: 10 Oct 2020 23:17:04.844 29133.4701949483

TDB Epoch: 10 Oct 2020 23:17:04.842 29133.4701949292

Cartesian State Keplerian State

--------------------------- --------------------------------

X = -21130.064944958 km SMA = 25513.000000033 km

Y = 21130.802534376 km ECC = 0.4000000000002

Z = 0.0000000000000 km INC = 0.0000000000000 deg

**VX = -3.0495872416288 km/sec**  RAAN = 0.0000000000000 deg

**VY = -1.3244038463119 km/sec**  AOP = 360.00000000000 deg

**VZ = 0.0000000000000 km/sec**  TA = 134.99900000255 deg

MA = 94.643047549128 deg

EA = 115.35387465803 deg

Spherical State Other Orbit Data

--------------------------- --------------------------------

RMAG = 29882.945978014 km Mean Motion = 1.549268155e-04 deg/sec

RA = 134.99900000253 deg Orbit Energy = -7.8117124897009 km^2/s^2

DEC = 0.0000000000000 deg C3 = -15.623424979402 km^2/s^2

VMAG = 3.3247598247740 km/s Semilatus Rectum = 21430.920000024 km

AZI = 90.000000000000 deg Angular Momentum = 92424.965100133 km^2/s

VFPA = 68.475802105204 deg Beta Angle = -6.9917202653511 deg

RAV = -156.52519789227 deg Periapsis Altitude = 8929.6637000150 km

DECV = 0.0000000000000 deg VelPeriapsis = 6.0377693136860 km/s

VelApoapsis = 2.5876154201499 km/s

Orbit Period = 40555.828160079 s

Planetodetic Properties

---------------------------

LST = 135.26514928876 deg

MHA = 8.9964759666473 deg

Latitude = -0.0802677349499 deg

Longitude = 126.26867332211 deg

Altitude = 23504.809719853 km

Spacecraft Properties

------------------------------

Cd = 2.200000

Drag area = 15.00000 m^2

Cr = 1.800000

Reflective (SRP) area = 1.000000 m^2

Dry mass = 850.00000000000 kg

Total mass = 850.00000000000 kg

SPADDragScaleFactor = 1.000000

SPADSRPScaleFactor = 1.000000

========================================================================

Spacecraft : Sat2

Coordinate System: EarthMJ2000Eq

Time System Gregorian Modified Julian

----------------------------------------------------------------------

UTC Epoch: 10 Oct 2020 12:00:00.000 29133.0000000000

TAI Epoch: 10 Oct 2020 12:00:37.000 29133.0004282407

TT Epoch: 10 Oct 2020 12:01:09.184 29133.0008007407

TDB Epoch: 10 Oct 2020 12:01:09.182 29133.0008007216

Cartesian State Keplerian State

--------------------------- --------------------------------

X = -21131.423961286 km SMA = 37668.000000000 km

Y = 21129.948761085 km ECC = 0.6565600000000

Z = 0.0000000000000 km INC = 0.0000000000000 deg

**VX = -3.9938238313617 km/sec** RAAN = 0.0000000000000 deg

**VY = -0.3802198116453 km/sec** AOP = 19.482000000000 deg

**VZ = 0.0000000000000 km/sec** TA = 115.52000000000 deg

MA = 35.947110547546 deg

EA = 71.653010021515 deg

Spherical State Other Orbit Data

--------------------------- --------------------------------

RMAG = 29883.303252446 km Mean Motion = 8.635948526e-05 deg/sec

RA = 135.00200000000 deg Orbit Energy = -5.2909690121589 km^2/s^2

DEC = 0.0000000000000 deg C3 = -10.581938024318 km^2/s^2

VMAG = 4.0118818403737 km/s Semilatus Rectum = 21430.416306355 km

AZI = 90.000000000000 deg Angular Momentum = 92423.878955830 km^2/s

VFPA = 50.436279597976 deg Beta Angle = -6.8144880605859 deg

RAV = -174.56172040202 deg Periapsis Altitude = 6558.5616200000 km

DECV = 0.0000000000000 deg VelPeriapsis = 7.1443176247429 km/s

VelApoapsis = 1.4811684726431 km/s

Orbit Period = 72756.169034291 s

Planetodetic Properties

---------------------------

LST = 135.26813282482 deg

MHA = 199.55190411210 deg

Latitude = -0.0804990060739 deg

Longitude = -64.283771287277 deg

Altitude = 23505.166994526 km

Spacecraft Properties

------------------------------

Cd = 2.200000

Drag area = 15.00000 m^2

Cr = 1.800000

Reflective (SRP) area = 1.000000 m^2

Dry mass = 850.00000000000 kg

Total mass = 850.00000000000 kg

SPADDragScaleFactor = 1.000000

SPADSRPScaleFactor = 1.000000

The red bolded values are the velocities for Sat1 and Sat2 at the maneuver point.

For Sat1

|  |  |
| --- | --- |
|  | -3.0495872416288 km/sec |
|  | -1.3244038463119 km/sec |
|  | 0.0000000000000 km/sec |

For Sat2

|  |  |
| --- | --- |
|  | -3.9938238313617 km/sec |
|  | -0.3802198116453 km/sec |
|  | 0.0000000000000 km/sec |

Then the difference of these two vectors become

Which is approximately

This agrees with out results on page 12. Thus, we have verified our results using GMAT.

**Problem 3**: Assume a relative two-body model and a space vehicle that is currently located in an Earth orbit with and . A single in-plane adjustment is to be implemented for a perigee-raise maneuver. The goal is a new periapsis distance such the new . At the same time, it is desired to produce an orbit that is less eccentric, i.e., the new eccentricity is and both goals are accomplished using the same maneuver. The maneuver will take place when the spacecraft is located at the end of the minor axis and descending.

1. Determine at the maneuver point. Sketch or plot the orbit in MATLAB. Mark the usual quantities at the maneuver location prior to the maneuver: ; also add appropriate unit vectors .

Plugging in the radius of the Earth we have

And we are given that

Using the following relationship

We find that the true anomaly is

the semi latus rectum is

Then, we can find the magnitude of

Since, , the magnitude of the velocity vector is

The specific angular momentum is

The flight path angle is

The mean anomaly becomes

The time elapse from periapsis is

Using the radial distance, we can express the r vector in the orbital frame

In the other frame this is

Then, the velocity vector becomes

The velocity vector is also represented as

The orbit is plotted on the next page

Diagram

Description automatically generated

1. Determine at the maneuver point. [Include the vector diagrams!!!!] Compute the required maneuver (). Express that maneuver in terms of sets of unit vectors.

From the requirements of the new orbit we can compute the new semi major axis

Then, the new semi latus rectum becomes

Then distance from Earth is the same as

Thus, we can get the true anomaly for the new orbit

The new velocity becomes

Then the new flight path angle is going to become

If ,

Similarly, if

If we choose a positive flight path angle,

If we choose a negative flight path angle,

We want to choose a that is smaller so we define

Thus, the new orbit is descending as well. Hence the position and velocity vectors will become

The maneuver becomes

and the angle becomes

The angle alpha is negative since the Earth is oriented above the local horizon and the is moving downwards (moving away from the Earth).

In the VNB coordinate system, we can define the velocity change as

1. Determine the characteristics of the new orbit: . How long till the spacecraft reaches perigee in the new orbit?

From the calculations of the previous parts of this problem we know that

The radius of perigee and apogee become

The period is

The specific energy is

From the relationship

the eccentric anomaly becomes,

The elapsed time from the perigee is

Finally,

1. Plot the old and new orbits on the same figure using your MATLAB script. On the plot, mark .

Chart

Description automatically generated

Appendix

MATLAB Code

Problem 1

% AAE 532 HW 6 Problem 1

% Tomoki Koike

close all; clear all; clc;

fdir = 'C:\Users\Tomo\Desktop\studies\2020-Fall\AAE532\matlab\outputs\ps6';

set(groot, 'defaulttextinterpreter','latex');

set(groot, 'defaultAxesTickLabelInterpreter','latex');

set(groot, 'defaultLegendInterpreter','latex');

format shortG;

% Set constants

planet\_consts = setup\_planetary\_constants(); % Function that sets up all the constants in the table

sun = planet\_consts.sun; % structure of sun

earth = planet\_consts.earth; % structure of earth

mars = planet\_consts.mars; % structure of mars

G = 6.6743015e-20; % Gravitational constant [km^3/kg/s^2]

% (a)

Rm = mars.mer;

mu = mars.gp;

rp = 1.1 \* Rm; % radius of periapsis

ra = 6.0 \* Rm; % radius of apoapsis

a = 0.5 \* (rp + ra); % semi major axis

e = (ra - rp) / (ra + rp); % eccentricity

p = a \* (1 - e^2); % semi laturs rectum

h = sqrt(mu \* p); % specific angular momentum

% At time t1

TA\_c = 90; % at time tc

r\_1 = 4.5 \* Rm;

TA\_1 = acos\_dbval(1/e \* (p/r\_1 - 1), "deg");

TA\_1 = TA\_1(find(min(TA\_1 - TA\_c)));

v\_1 = vis\_viva(r\_1, a, mu);

FPA\_1 = acos\_dbval(h / r\_1 / v\_1, "deg");

FPA\_1 = FPA\_1(FPA\_1 > 0);

EA\_1 = T2E\_anomaly(e, TA\_1, "deg");

EA\_c = T2E\_anomaly(e, TA\_c, "deg");

dt\_p1 = sqrt(a^3 / mu) \* (EA\_1 - e \* sind(EA\_1));

dt\_p1\_day = dt\_p1 / 60 / 60 / 24;

% (b)

% Ellipse

theta = 0:0.01:2\*pi;

R = p ./ (1 + e \* cos(theta));

X = R .\* cos(theta); Y = R .\* sin(theta);

% Mars

Xm = Rm \* cos(theta); Ym = Rm \* sin(theta);

% At time t1

X1 = r\_1 \* cosd(TA\_1); Y1 = r\_1 \* sind(TA\_1);

% Plot

fig = figure("Renderer","painters","Position",[10 10 900 700]);

plot(X, Y, '-b')

title('Elliptical Orbit of $e=0.6901$ Around Mars - T. Koike')

hold on; grid on; grid minor; box on; axis equal;

plot(X1, Y1, '.r', 'MarkerSize', 18)

% Mars

plot(Xm, Ym, '-m')

plot(0, 0, '.k', 'MarkerSize', 15)

% Axes

nanikore = -ra:rp; korenani = -a\*sqrt(1-e^2):a\*sqrt(1-e^2);

plot(nanikore, zeros(size(nanikore)), '--k')

plot(-a\*e\*ones(size(korenani)), korenani, '--k')

hold off

xlabel('$\hat{e}$ [km]')

ylabel('$\hat{p}$ [km]')

saveas(fig, fullfile(fdir, 'p1\_orbit1.png'));

% (c)

EA\_c = T2E\_anomaly(e, TA\_c, "deg");

dt\_pc = sqrt(a^3 / mu) \* (EA\_c - e \* sind(EA\_c));

dt\_pc\_day = dt\_pc / 60 / 60 / 24;

dt\_c1 = dt\_p1 - dt\_pc;

dt\_c1\_day = dt\_c1 / 60 / 60 / 24;

% (d)

r\_new = r\_1;

v\_new = sqrt(mu / r\_1);

v\_1\_vec = v\_1 \* [sind(FPA\_1), cosd(FPA\_1)];

v\_new\_vec = v\_new \* [0, 1];

dv\_vec = v\_new\_vec - v\_1\_vec;

dv = norm(dv\_vec);

syms alpha

eqn = v\_new / sind(180-alpha) == dv / sind(FPA\_1);

alpha = solve(eqn, alpha)

temp\_ang = acosd(dot(v\_new\_vec, dv\_vec)/ dv / v\_new) + FPA\_1

% alpha = acosd(dot(v\_new\_vec, v\_1\_vec) / v\_new / v\_1);

% (e)

% Ellipse

theta = 0:0.01:2\*pi;

R = p ./ (1 + e \* cos(theta));

X = R .\* cos(theta); Y = R .\* sin(theta);

% Mars

Xm = Rm \* cos(theta); Ym = Rm \* sin(theta);

% At time t1

X1 = r\_1 \* cosd(TA\_1); Y1 = r\_1 \* sind(TA\_1);

% New circular orbit

Xnew = r\_1 \* cos(theta); Ynew = r\_1 \* sin(theta);

% Plot

fig = figure("Renderer","painters","Position",[10 10 900 700]);

plot(X, Y, '-b')

title('Elliptical Orbit of $e=0.6901$ Around Mars with Maneuver to Circular Orbit - T. Koike')

hold on; grid on; grid minor; box on; axis equal;

plot(X1, Y1, '.r', 'MarkerSize', 18)

% Mars

plot(Xm, Ym, '-m')

plot(0, 0, '.k', 'MarkerSize', 15)

% Axes

nanikore = -ra:rp; korenani = -a\*sqrt(1-e^2):a\*sqrt(1-e^2);

plot(nanikore, zeros(size(nanikore)), '--k')

plot(-a\*e\*ones(size(korenani)), korenani, '--k')

% New orbit

plot(Xnew, Ynew, '-g')

hold off

xlabel('$\hat{e}$ [km]')

ylabel('$\hat{p}$ [km]')

saveas(fig, fullfile(fdir, 'p1\_orbit2.png'));

% (f)

vvec1 = [-1.37, -0.4173];

vvec2 = [-0.8808, -1.4230];

Dv = vvec2 - vvec1;

Dv\_orbital = Dv \* [cosd(TA\_1), -sind(TA\_1); sind(TA\_1), cosd(TA\_1)];

Problem 2

% AAE 532 HW 6 Problem 2

% Tomoki Koike

close all; clear all; clc;

fdir = 'C:\Users\Tomo\Desktop\studies\2020-Fall\AAE532\matlab\outputs\ps6';

set(groot, 'defaulttextinterpreter','latex');

set(groot, 'defaultAxesTickLabelInterpreter','latex');

set(groot, 'defaultLegendInterpreter','latex');

format shortG;

% Set constants

planet\_consts = setup\_planetary\_constants(); % Function that sets up all the constants in the table

sun = planet\_consts.sun; % structure of sun

earth = planet\_consts.earth; % structure of earth

mars = planet\_consts.mars; % structure of mars

G = 6.6743015e-20; % Gravitational constant [km^3/kg/s^2]

% (a)

Re = earth.mer; % Earth radius

mu = earth.gp; % gravitational parameter

a = 4\*Re; % semi major axis

e = 0.4; % eccentricity

p = a \* (1 - e^2); % semi latus rectum

TA = 135; % true anomaly at maneuver point

r1 = p / (1 + e \* cosd(TA));

v1 = vis\_viva(r1, a, mu);

h = sqrt(mu \* p); % specific angular momentum

FPA1 = acosd(h / r1 / v1); % flight path angle

r1vec\_i = r1 \* [cosd(TA), sind(TA)];

v1vec\_o = v1 \* [sind(FPA1), cosd(FPA1)]

v1vec\_i = v1vec\_o \* [cosd(TA), sind(TA); -sind(TA), cosd(TA)]

% Maneuver

dv = 0.90; % km/s

alpha = 45; % degrees

v2 = sqrt(dv^2 + v1^2 - 2\*dv\*v1\*cosd(180 - alpha));

FPA2 = acosd(h / r1 / v2)

dFPA = FPA2 - FPA1

v2vec\_o = v2 \* [sind(FPA2), cosd(FPA2)]

dvvec\_o = v2vec\_o - v1vec\_o

v2vec\_i = v2vec\_o \* [cosd(TA), sind(TA); -sind(TA), cosd(TA)]

dvvec\_i = dvvec\_o \* [cosd(TA), sind(TA); -sind(TA), cosd(TA)]

v1vec\_vbn = v1 \* [1, 0];

v2vec\_vbn = v2 \* [cosd(dFPA), sind(dFPA)];

dvvec\_vbn = v2vec\_vbn - v1vec\_vbn;

% (d)

r2 = r1;

a\_new = (-mu / 2) / (v2^2/2 - mu/r2);

e\_new = calc\_e\_posVel(r2, v2, FPA2, mu, "deg")

p\_new = a\_new\*(1 - e\_new^2)

h = sqrt(mu \* p\_new)

IP = 2\*pi\*sqrt(a\_new^3 / mu)

IP\_day = IP / 60 /60 / 24

En = -mu / 2 / a\_new

TA\_new = acos\_dbval(1/e\_new \*(p\_new/r2 - 1), "deg")

TA\_new = TA\_new(TA\_new >0)

E\_new = T2E\_anomaly(e\_new, TA\_new, "deg")

dt = (deg2rad(E\_new) - sind(E\_new)) \* sqrt(a\_new^3 / mu)

rp\_new = a\_new\*(1 - e\_new)

Domega = TA - TA\_new

ra\_new = a\_new\*(1 + e\_new)

rp = a\*(1-e);

ra = a\*(1+e);

% Plotting for visualization

% old orbit

angles = 0:0.01:2\*pi;

RR = p ./ (1 + e\*cos(angles)); XX = RR.\*cos(angles); YY = RR.\*sin(angles);

% new orbit

RR\_new = p\_new ./ (1 + e\_new\*cos(angles - deg2rad(Domega)));

XX\_new = RR\_new.\*cos(angles);

YY\_new = RR\_new.\*sin(angles);

rp\_vec = rp\_new\*[cosd(Domega), sind(Domega)];

ra\_vec = ra\_new\*[cosd(Domega+180), sind(Domega+180)];

Xearth = Re\*cos(angles); Yearth = Re\*sin(angles);

fig = figure("Renderer","painters","Position",[10, 10, 900, 700]);

plot(XX,YY)

hold on; grid on; grid minor; box on; axis equal;

plot(r1vec\_i(1), r1vec\_i(2), '.r', 'MarkerSize', 20)

plot(0,0,'.k', 'MarkerSize', 15)

plot(Xearth, Yearth, '-g')

plot(XX\_new, YY\_new, '-m')

plot([-ra, rp], [0, 0], '--k')

plot([ra\_vec(1), rp\_vec(1)],[ra\_vec(2), rp\_vec(2)], '--r')

hold off

title('PS6 P2 Maneuver Example Orbit - T. Koike')

xlabel('$\hat{e}$ [km]')

ylabel('$\hat{p}$ [km]')

saveas(fig, fullfile(fdir, "p2-orbit.png"))

function e = calc\_e\_posVel(r, v, gamma, mu, unit)

A = (r\*v^2 / mu - 1)^2;

if unit == "rad"

B = cos(gamma)^2;

C = sin(gamma)^2;

else

B = cosd(gamma)^2;

C = sind(gamma)^2;

end

e = sqrt(A\*B+C);

end

Problem 3

% AAE 532 HW 6 Problem 2

% Tomoki Koike

close all; clear all; clc;

fdir = 'C:\Users\Tomo\Desktop\studies\2020-Fall\AAE532\matlab\outputs\ps6';

set(groot, 'defaulttextinterpreter','latex');

set(groot, 'defaultAxesTickLabelInterpreter','latex');

set(groot, 'defaultLegendInterpreter','latex');

format shortG;

% Set constants

planet\_consts = setup\_planetary\_constants(); % Function that sets up all the constants in the table

sun = planet\_consts.sun; % structure of sun

earth = planet\_consts.earth; % structure of earth

mars = planet\_consts.mars; % structure of mars

G = 6.6743015e-20; % Gravitational constant [km^3/kg/s^2]

% (a)

Re = earth.mer; % Earth radius

mu = earth.gp; % gravitational parameter

a = 3\*Re; % semi major axis

e = 0.6; % eccentricity

p = a \* (1 - e^2); % semi latus rectum

E = -90; % true anomaly at maneuver point

M = rad2deg(deg2rad(E) - sind(E))

TA = E2T\_anomaly(e, E, "deg")

r1 = p / (1 + e \* cosd(TA));

v1 = vis\_viva(r1, a, mu);

h = sqrt(mu \* p); % specific angular momentum

FPA1 = -acosd(h / r1 / v1) % flight path angle

dt\_1p = deg2rad(360+M)\*sqrt(a^3 / mu)

dt\_1p\_day = dt\_1p / 60 / 60 / 24

ra = a\*(1+e); rp = a\*(1-e); b = a\*sqrt(1-e^2);

r1vec\_o = r1 \* [1, 0];

r1vec\_i = r1 \* [cosd(TA), sind(TA)];

v1vec\_o = v1 \* [sind(FPA1), cosd(FPA1)];

v1vec\_i = v1vec\_o \* [cosd(TA), sind(TA); -sind(TA), cosd(TA)];

% Plotting for visualization

% old orbit

angles = 0:0.01:2\*pi;

RR = p ./ (1 + e\*cos(angles)); XX = RR.\*cos(angles); YY = RR.\*sin(angles);

% Reference circle

Xref = a\*cos(angles)-a\*e; Yref = a\*sin(angles);

% new orbit

% RR\_new = p\_new ./ (1 + e\_new\*cos(angles - deg2rad(Domega)));

% XX\_new = RR\_new.\*cos(angles);

% YY\_new = RR\_new.\*sin(angles);

% rp\_vec = rp\_new\*[cosd(Domega), sind(Domega)];

% ra\_vec = ra\_new\*[cosd(Domega+180), sind(Domega+180)];

% Earth

Xearth = Re\*cos(angles); Yearth = Re\*sin(angles);

fig = figure("Renderer","painters","Position",[10, 10, 900, 700]);

plot(XX,YY)

hold on; grid on; grid minor; box on; axis equal;

% Maneuver point

plot(r1vec\_i(1), r1vec\_i(2), '.r', 'MarkerSize', 20)

% Center of orbit

plot(0,0,'.k', 'MarkerSize', 15)

% Reference circle

plot(Xref, Yref, '-m')

% Semi minor axis

plot([-a\*e, -a\*e], [-b, b], '--k')

% Earth

plot(Xearth, Yearth, '-g')

% plot(XX\_new, YY\_new, '-m')

plot([-ra, rp], [0, 0], '--k')

% plot([ra\_vec(1), rp\_vec(1)],[ra\_vec(2), rp\_vec(2)], '--r')

hold off

title('PS6 P3 Orbit without Maneuver - T. Koike')

xlabel('$\hat{e}$ [km]')

ylabel('$\hat{p}$ [km]')

saveas(fig, fullfile(fdir, "p3-orbit-noManeuver.png"))

% New orbit after maneuver

rp2 = 2\*Re

e2 = 0.4

a2 = rp2 / (1 - e2)

p2 = a2 \* (1 - e2^2)

r2 = r1

TA2 = acos\_dbval(1/e2 \* (p2/r2 - 1), "deg")

v2 = vis\_viva(r2, a2, mu)

FPA2 = acos\_dbval(sqrt(mu\*p2)/r2/v2, "deg")

TA2\_eval1 = TAfromRVGamma(r2, v2, FPA2(1), mu, "deg")

TA2\_eval2 = TAfromRVGamma(r2, v2, FPA2(2), mu, "deg")

FPA2 = FPA2(FPA2 < 0)

TA2 = TA2(TA2 < 0)

r2vec\_o = r2 \* [1, 0];

v2vec\_o = v2 \* [sind(FPA2), cosd(FPA2)]

dvvec\_o = v2vec\_o - v1vec\_o

dv = norm(dvvec\_o)

alpha = acos\_dbval(dot(v1vec\_o, dvvec\_o)/v1/norm(dvvec\_o), "deg")

alpha = alpha(alpha > 0)

dvvec\_VBN = dv \* [cosd(alpha), sind(alpha)]

rp2 = a2\*(1-e2)

ra2 = a2\*(1+e2)

IP = 2\*pi\*sqrt(a2^3/mu)

IP\_day = IP / 60 / 60 / 24

En = -mu/2/a2

E2 = T2E\_anomaly(e2, TA2, "deg")

dt\_2p = (deg2rad(360 + E2) - sind(360 + E2))\*sqrt(a2^3/mu)

dt\_2p\_day = dt\_2p / 60/60/24

Domega = (TA-360) - TA2

% Plotting for visualization

% old orbit

angles = 0:0.01:2\*pi;

RR = p ./ (1 + e\*cos(angles)); XX = RR.\*cos(angles); YY = RR.\*sin(angles);

% Reference circle

% Xref = a\*cos(angles)-a\*e; Yref = a\*sin(angles);

% new orbit

RR\_new = p2 ./ (1 + e2\*cos(angles - deg2rad(Domega)));

XX\_new = RR\_new.\*cos(angles);

YY\_new = RR\_new.\*sin(angles);

rp\_vec = rp2\*[cosd(Domega), sind(Domega)];

ra\_vec = ra2\*[cosd(Domega+180), sind(Domega+180)];

% Earth

Xearth = Re\*cos(angles); Yearth = Re\*sin(angles);

fig = figure("Renderer","painters","Position",[10, 10, 900, 700]);

plot(XX,YY)

hold on; grid on; grid minor; box on; axis equal;

% Maneuver point

plot(r1vec\_i(1), r1vec\_i(2), '.r', 'MarkerSize', 20)

% Center of orbit

plot(0,0,'.k', 'MarkerSize', 15)

% Reference circle

% plot(Xref, Yref, '-m')

% Semi minor axis

% plot([-a\*e, -a\*e], [-b, b], '--k')

% Earth

plot(Xearth, Yearth, '-g')

plot(XX\_new, YY\_new, '-m')

plot([-ra, rp], [0, 0], '--k')

plot([ra\_vec(1), rp\_vec(1)],[ra\_vec(2), rp\_vec(2)], '--r')

hold off

title('PS6 P3 Orbit without Maneuver - T. Koike')

xlabel('$\hat{e}$ [km]')

ylabel('$\hat{p}$ [km]')

saveas(fig, fullfile(fdir, "p3-orbit-withManeuver.png"))

function TA = TAfromRVGamma(r, v, gamma, mu, unit)

A = r\*v^2/mu;

if unit == "rad"

gamma = rad2deg(gamma);

end

num = A\*cosd(gamma)\*sind(gamma);

den = A\*cosd(gamma)^2 - 1;

TA = atan\_dbval(num/ den, unit);

end

Functions

function v = vis\_viva(r, a, mu)

%{

NAME: VIS\_VIVA

AUTHOR: TOMOKI KOIKE

INPUTS: (1) r: POSITION (LENGTH) ON ORBIT

(2) a: SEMI MAJOR AXIS

(3) mu: GRAVITATIONAL PARAMETER

OUTPUTS: (1) v: VELOCITY AT THE POSITION

DESCRIPTION: CALCULATES THE VELOCITY FOR A CERTAIN POSITION ON A

CONIC ORBIT.

%}

v = sqrt(mu \* (2/r - 1/a));

end

function E = T2E\_anomaly(e, theta\_star, unit)

%{

NAME: T2E\_anomaly

AUTHOR: TOMOKI KOIKE

INPUTS: (1) e: ECCENTRICITY

(2) theta\_star: TRUE ANOMALY

(3) unit: DEGREES OR RADIANS

OUTPUTS: (1) E: ECCENTRIC ANOMALY

DESCRIPTION: CALCULATES THE ECCENTRIC ANOMALY FROM THE TRUE ANOMALY.

%}

ee = sqrt((1 - e) / (1 + e));

if unit == "deg"

E = 2\*atand(ee \* tand(theta\_star / 2));

else

E = 2\*atan(ee \* tan(theta\_star / 2));

end

end

function theta\_star = E2T\_anomaly(e, E, unit)

%{

NAME: E2T\_anomaly

AUTHOR: TOMOKI KOIKE

INPUTS: (1) e: ECCENTRICITY

(2) E: ECCENTRIC ANOMALY

(3) unit: DEGREES OR RADIANS

OUTPUTS: (1) theta\_star: TRUE ANOMALY

DESCRIPTION: CALCULATES THE TRUE ANOMALY FROM THE ECCENTRIC ANOMALY.

%}

ee = sqrt((1 + e) / (1 - e));

if unit == "deg"

theta\_star = 2\*atand(ee \* tand(E / 2));

if theta\_star < 0

theta\_star = 360 + theta\_star;

end

else

theta\_star = 2\*atan(ee \* tan(E/ 2));

if theta\_star < 0

theta\_star = 2\*pi + theta\_star;

end

end

end

function res = atan\_dbval(x, unit)

if unit == "deg"

ang1 = atand(x); % -90 to 90

ang2 = 180 - (-ang1);

if ang2 > 180

ang2 = ang2 - 360;

end

else

ang1 = atan(x);

ang2 = pi - (-ang1);

if ang2 > pi

ang2 = ang2 - 2\*pi;

end

end

res = [ang1, ang2];

end

function res = asin\_dbval(x, unit)

if unit == "deg"

ang1 = asind(x);

if (0<=ang1 && ang1<=180)

ang2 = 180 - ang1;

elseif -90<=ang1 && ang1<0

ang2 = -ang1 - 180;

else

ang2 = 540 - ang1;

end

else

ang1 = asin(x);

if (0<=ang1 && ang1<=pi)

ang2 = pi - ang1;

elseif -pi/2<=ang1 && ang1<0

ang2 = -ang1 - pi;

else

ang2 = 3\*pi - ang1;

end

end

res = [ang1, ang2];

end

function res = acos\_dbval(x, unit)

if unit == "deg"

ang1 = acosd(x);

if (0<=ang1 && ang1<=180)

ang2 = -ang1;

else

ang2 = 360 - ang1;

end

else

ang1 = asin(x);

if (0<=ang1 && ang1<pi)

ang2 = -ang1;

else

ang2 = 2\*pi - ang1;

end

end

res = [ang1, ang2];

end

Auxiliary MATLAB Setup File

%% Table of Constants

function planets = setup\_planetary\_constants()

%{

arp : Axial Rotational Period (Rev / Day)

mer : Mean Equatorial Radius (km)

gp : Gravitational Parameter, mu (km^3 / s^2)

smao : Semi-Major Axis of Orbit (km)

op : Orbital Period (s)

eo : Eccentricity of Orbit

ioe : Inclination of Orbit to Ecliptic (deg)

%}

% Sun

sun.arp = 0.0394011;

sun.mer = 695990;

sun.gp = 132712440017.99;

sun.smao = NaN;

sun.op = NaN;

sun.eo = NaN;

sun.ioe = NaN;

% Moon

moon.arp = 0.0366004;

moon.mer = 1738.2;

moon.gp = 4902.8005821478;

moon.smao = 384400;

moon.op = 2360592;

moon.eo = 0.0554;

moon.ioe = 5.16;

% Mercury

mercury.arp = 0.0170514;

mercury.mer = 2439.7;

mercury.gp = 22032.080486418;

mercury.smao = 57909101;

mercury.op = 7600537;

mercury.eo = 0.20563661;

mercury.ioe = 7.00497902;

% Venus

venus.arp = 0.0041149; % retrograde

venus.mer = 6051.9;

venus.gp = 324858.59882646;

venus.smao = 108207284;

venus.op = 19413722;

venus.eo = 0.00676399;

venus.ioe = 3.39465605;

% Earth

earth.arp = 1.0027378;

earth.mer = 6378.1363;

earth.gp = 398600.4415;

earth.smao = 149597898;

earth.op = 31558205;

earth.eo = 0.01673163;

earth.ioe = 0.00001531;

% Mars

mars.arp = 0.9747000;

mars.mer = 3397;

mars.gp = 42828.314258067;

mars.smao = 227944135;

mars.op = 59356281;

mars.eo = 0.09336511;

mars.ioe = 1.84969142;

% Jupiter

jupiter.arp = 2.4181573;

jupiter.mer = 71492;

jupiter.gp = 126712767.8578;

jupiter.smao = 778279959;

jupiter.op = 374479305;

jupiter.eo = 0.04853590;

jupiter.ioe = 1.30439695;

% Saturn

saturn.arp = 2.2522053;

saturn.mer = 60268;

saturn.gp = 37940626.061137;

saturn.smao = 1427387908;

saturn.op = 930115906;

saturn.eo = 0.05550825;

saturn.ioe = 2.48599187;

% Uranus

uranus.arp = 1.3921114; % retrograde

uranus.mer = 25559;

uranus.gp = 5794549.0070719;

uranus.smao = 2870480873;

uranus.op = 2652503938;

uranus.eo = 0.04685740;

uranus.ioe = 0.77263783;

% Neptune

neptune.arp = 1.4897579;

neptune.mer = 25269;

neptune.gp = 6836534.0638793;

neptune.smao = 4498337290;

neptune.op = 5203578080;

neptune.eo = 0.00895439;

neptune.ioe = 1.77004347;

% Pluto

pluto.arp = -0.1565620; % retrograde

pluto.mer = 1162;

pluto.gp = 981.600887707;

pluto.smao = 5907150229;

pluto.op = 7830528509;

pluto.eo = 0.24885238;

pluto.ioe = 17.14001206;

% Return

planets.sun = sun;

planets.moon = moon;

planets.mercury = mercury;

planets.venus = venus;

planets.earth = earth;

planets.mars = mars;

planets.jupiter = jupiter;

planets.saturn = saturn;

planets.uranus = uranus;

planets.neptune = neptune;

planets.pluto = pluto;

end