Homework 3

Tycho Bogdanowitsch
AA 279A - Space Mechanics

February 5, 2025

Problem 1.

The given parameters for the orbit are: $e = 0.89$, $r_a = 3800$ km, and $\nu_0 = 110°$. Apoapsis occurs at $\nu = 180°$, so Kepler's equation can be used to solve the time of flight problem.
First, the semi-major axis a is computed using:

$$a = \frac{r_a}{1+e}$$

Then, the mean motion n is calculated using:

$$n = \sqrt{\frac{\mu_{\text{Mars}}}{a^3}}$$

where μ_{Mars} is the standard gravitational parameter of Mars.
Next, the eccentric anomaly E is related to the true anomaly ν by:

$$E = \cos^{-1}\left(\frac{e + \cos\nu}{1 + e\cos\nu}\right)$$

This expression is evaluated at the initial and final conditions to find E_0 and E.
The time required to move from ν_0 to ν is given by Kepler's equation (assuming $t_0 = 0$):

$$t = \frac{1}{n}\left[(E - e\sin E) - (E_0 - e\sin E_0)\right]$$

Converting this to minutes, the remaining time before Starship reaches apoapsis is $\mathbf{t = 21.971}$ minutes.

Problem 2.

Part (a)

The given parameters for Halley's Comet are:

- Semi-major axis:
$$a = 17.834 \times AU$$
where $AU = 1.496 \times 10^8$ km (astronomical unit).

- Eccentricity:
$$e = 0.967$$

- Standard gravitational parameter of the Sun:
$$\mu_{\text{Sun}} = 1.327 \times 10^{11} \text{ km}^3/\text{s}^2$$

The orbital period T is computed using Kepler's Third Law:

$$T = 2\pi\sqrt{\frac{a^3}{\mu_{\text{Sun}}}}$$

Solving and converting to Earth years, we find that the orbital period of Halley's Comet is **75.362** Earth years.
The **mean motion** n is computed using:

$$n = \frac{2\pi}{T}$$

Evaluating this, we find the mean motion of Halley's Comet is **2.64e-9 rad/s**.

Part (b)

See the MATLAB code in the appendix.

Part (c)

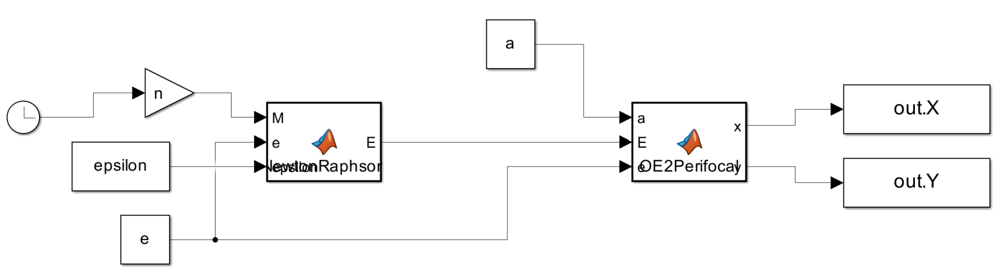

Figure 1: Simulink Block Diagram

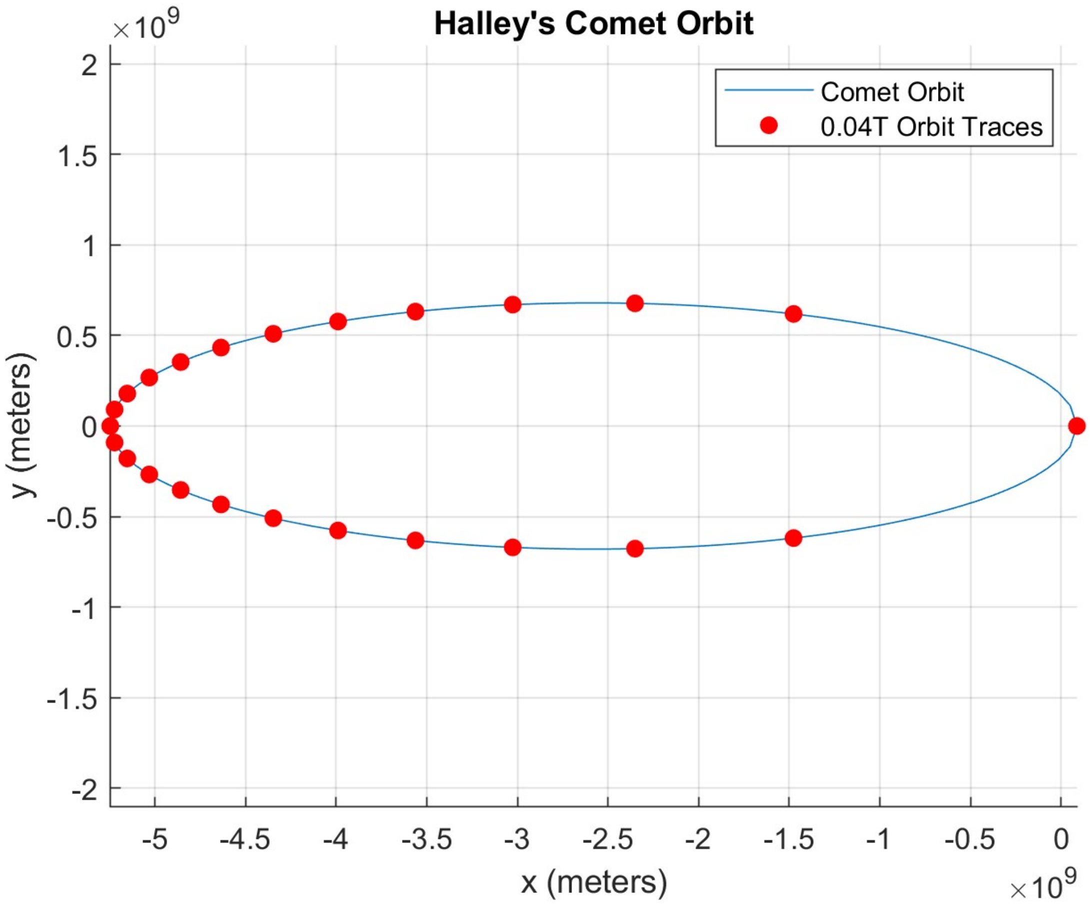

Figure 2: Halley's Comet Orbit with Orbital Traces

Part (d)

Part (e)

The spacing of the tick marks near the periapsis is much larger than near the apoapsis. This is because the comet's velocity is maximum at its periapsis and minimum at its apoapsis. A higher velocity correlates to larger spacing since the comet travels more distance in the same amount of time.

Problem 3.

Part (a)

The distance unit (DU) and time unit (TU) are based on Earth's radius and defined as:

$$DU_{\text{Earth}} = R_E = 6378 \text{ km}$$
$$TU_{\text{Earth}} = \sqrt{\frac{DU_{\text{Earth}}^3}{\mu_{\text{Earth}}}}$$

where μ_{Earth} is the standard gravitational parameter of Earth.
The satellite's initial position vector $\mathbf{r}$ in the inertial frame is given by:

$$\mathbf{r} = \begin{bmatrix} 8 \\ 0 \\ 0 \end{bmatrix} DU_{\text{Earth}}$$

The velocity vector $\mathbf{v}$ is given by:

$$\mathbf{v} = \begin{bmatrix} -\frac{1}{\sqrt{8}} \\ \frac{1}{\sqrt{8}} \\ 0 \end{bmatrix} \frac{DU_{\text{Earth}}}{TU_{\text{Earth}}}$$

First, the magnitudes of the position and velocity vectors are calculated. Then the specific mechanical energy ϵ is given by:

$$\epsilon = \frac{v_{\text{norm}}^2}{2} - \frac{\mu_{\text{Earth}}}{r_{\text{norm}}}$$

Evaluating this, we find that the specific mechanical energy is 0, which means that the orbit is in a parabolic orbit. Specifically, since there is no k component of position or velocity (i.e. the angular momentum vector is only in k, the debris is in an **equatorial parabolic orbit**.

Part (b)

To see if it possible for the debris to collide with the satellite at some time in the future, we need to find the periapsis of the debris orbit, compare it to the radius of the circular satellite orbit, and then if necessary check the current position of the debris relative to any potential collision point.
The specific angular momentum $\mathbf{h}$ is computed using the cross product of the position $\mathbf{r}$ and velocity $\mathbf{v}$:

$$\mathbf{h} = \mathbf{r} \times \mathbf{v}$$

The semi-parameter p of the debris orbit is given by:

$$p_{\text{debris}} = \frac{h_{\text{norm}}^2}{\mu_{\text{Earth}}}$$

where μ_{Earth} is the standard gravitational parameter of Earth.

The periapsis radius r_p is computed as:

$$r_{p_{\text{debris}}} = \frac{p_{\text{debris}}}{2}$$

This represents the closest approach of the debris to Earth.
The radius of the orbit of the satellite is computed as:

$$r_{\text{threshold}} = DU_{\text{Earth}} + 19134$$

where: - DU_{Earth} is the Earth's reference radius.
Evaluating this, we find that the periapsis of the debris is at **25,512 km**. The radius of the satellite's orbit is also **25,512 km**.
Next, we need to figure out where the periapsis of the debris orbit occurs. So, the eccentricity vector $\mathbf{e}$ is computed using:

$$\mathbf{e} = \frac{\mathbf{v} \times \mathbf{h}}{\mu_{\text{Earth}}} - \frac{\mathbf{r}}{r_{\text{norm}}}$$

Evaluating this, we find the eccentricity vector is entirely in the j-direction. Comparing this to the current position of the debris, we find that the debris is still approaching its periapsis. This means that it **is possible for the orbital debris to collide with the satellite**.

Part (c)

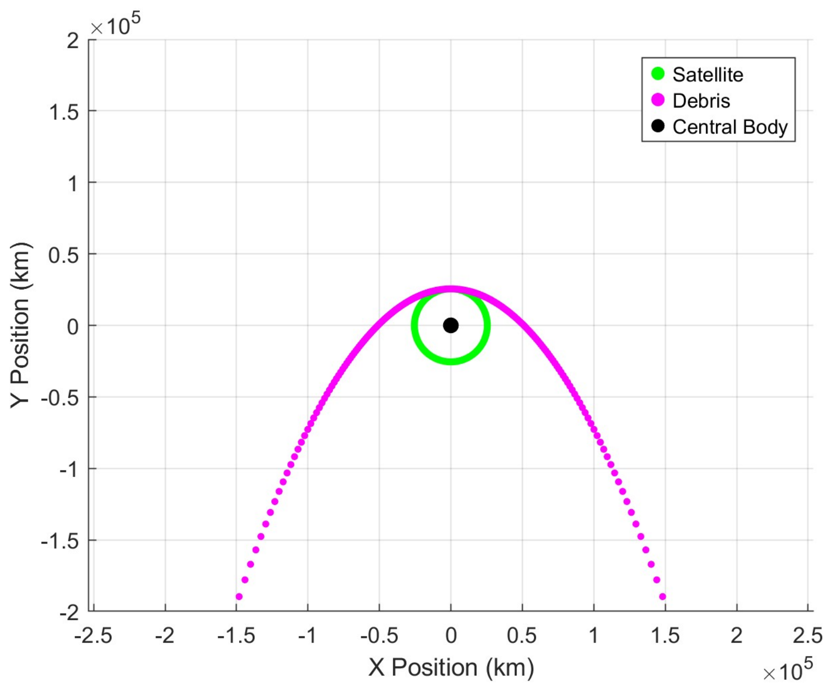

Figure 3: Orbits of the debris and the satellite

Part (d)

The velocity of the debris at periapsis or collision point is computed using the vis-viva equation:

$$v_{\text{debris},p} = \sqrt{\mu_{\text{Earth}}\left(\frac{2}{r_{p_{\text{debris}}}}\right)}$$

Since:

$$r_{p_{\text{debris}}} = \frac{p_{\text{debris}}}{2}$$

we can substitute:

$$v_{\text{debris},p} = \sqrt{\mu_{\text{Earth}}\left(\frac{2}{\frac{p_{\text{debris}}}{2}}\right)}$$

The velocity of the satellite at the collision point is computed as:

$$v_{\text{sat}} = \sqrt{\mu_{\text{Earth}}\left(\frac{2}{r_{p_{\text{sat}}}} - \frac{1}{2r_{p_{\text{sat}}}}\right)}$$

Since the satellite is in prograde orbit, the debris and the satellite are moving in the same direction. Thus, the relative velocity between the debris and satellite is calculated as:

$$v_{\text{rel}} = v_{\text{debris},p} - v_{\text{sat}}$$

Evaluating this, we find that the relative velocity is **0.749 km/s**.
The true anomaly ν of the debris at its initial position is computed using:

$$\nu = \cos^{-1}\left(\frac{p_{\text{debris}}}{r_{\text{norm}}} - 1\right)$$

Evaluating this, we find that the true anomaly of the debris at the initial position is **-90 degrees**.

Problem 4.

The position vector $\mathbf{r}_{IJK}$ in the ECI frame of the Envisat is given by:

$$\mathbf{r}_{IJK} = \begin{bmatrix} 3105.4128 \\ -880.8531 \\ 6368.9408 \end{bmatrix} \text{ km}$$

The velocity vector $\mathbf{v}_{IJK}$ in the ECI frame of the Envisat is:

$$\mathbf{v}_{IJK} = \begin{bmatrix} -6.7276 \\ -0.5612 \\ 3.2023 \end{bmatrix} \text{ km/s}$$

To calculate the Keplerian orbital elements from position and velocity in the ECI frame, first the magnitudes (norms) of the position and velocity vectors are computed.
The specific angular momentum vector is computed as:

$$\mathbf{h} = \mathbf{r}_{IJK} \times \mathbf{v}_{IJK}$$

and its magnitude is:

$$h_{\text{norm}} = \|\mathbf{h}\|$$

The unit vector normal to the orbital plane is:

$$\mathbf{W} = \frac{\mathbf{h}}{\|\mathbf{h}\|}$$

with components:

$$W_x, W_y, W_z$$

The inclination i is computed using:

$$i = \tan^{-1}\left(\frac{\sqrt{W_x^2 + W_y^2}}{W_z}\right)$$

The Right Ascension of the Ascending Node (RAAN) is given by:

$$\Omega = \tan^{-1}\left(\frac{W_x}{-W_y}\right)$$

The semi-major axis is computed using the vis-viva equation:

$$a = \left(\frac{2}{r_{\text{norm}}} - \frac{v_{\text{norm}}^2}{\mu_{\text{Earth}}}\right)^{-1}$$

The eccentricity is determined as:

$$e = \sqrt{1 - \frac{p}{a}}$$

where p is the semi-parameter:

$$p = \frac{h_{\text{norm}}^2}{\mu_{\text{Earth}}}$$

The eccentric anomaly E is computed as:

$$E = \tan^{-1}\left(\frac{\mathbf{r}_{IJK} \cdot \mathbf{v}_{IJK}}{a^2 n}, \frac{1 - r_{\text{norm}}}{a}\right)$$

where n is the mean motion:

$$n = \sqrt{\frac{\mu_{\text{Earth}}}{a^3}}$$

The mean anomaly is:

$$M = E - e \sin E$$

The true anomaly ν is computed as:

$$\nu = 2 \tan^{-1}\left(\sqrt{\frac{1+e}{1-e}} \tan \frac{E}{2}\right)$$

If $\nu < 0$, we adjust:

$$\nu = 360^\circ + \nu$$

The argument of perigee is computed as:

$$\omega = u - \nu$$

where:

$$u = \tan^{-1}\left(\frac{r_z}{\sin i}, \frac{r_x \cos \Omega + r_y \sin \Omega}{\cos i}\right)$$

The results are converted to degrees:

$$i = \deg(i), \quad \Omega = \deg(\Omega), \quad \omega = \deg(\omega), \quad \nu = \deg(\nu)$$

Thus, the computed orbital elements for Envisat are:

- **Semi-major axis**: 7140.97 km
- **Eccentricity**: 0.00011
- **Inclination**: 98.26 degrees
- **RAAN**: 0.82 degrees
- **Argument of perigee**: 87.78 degrees
- **True anomaly**: 336.6 degrees

Appendix 1

```
%% Space Mechanics Homework 3
% Tycho Bogdanowitsch
clc; clear;
% Constants
R_earth = 6371; % km
DU_earth = R_earth; % km
AU = 1.496e8; % km
mu_earth = 3.986e5; % km^3/s^2
mu_mars = 4.283e4; % km^3/s^2
mu_sun = 1.327e11; % km^3/s^2
R_sun = 6.963e5; % km

%% Problem 1
close all;
e = 0.89;
r_a = 3800; % kmco
nu_0 = deg2rad(110); % rad

nu = deg2rad(180); % rad
a = r_a/(1+e); % km
n = sqrt(mu_mars/a^3);the re

E_0 = acos((e+cos(nu_0))/(1+e*cos(nu_0)));
E = acos((e+cos(nu))/(1+e*cos(nu)));

t = (1/n)*((E-e*sin(E))-(E_0-e*sin(E_0))) / 60;

fprintf('The time remaining is: %.3f minutes\n', t);

%% Problem 2
close all;
a = 17.834 * AU; % km
e = 0.967;

% Part i
T = 2*pi*sqrt(a^3/mu_sun); % seconds
T_converted = T / 3.154e7; % seconds --> Earth years
n = 2*pi/T; % rad/s

fprintf('The orbital period of Halleys Comet is: %.3f Earth
   years\n', T_converted);
```

```
fprintf('The mean motion of Halleys Comet is: %.3e rad/s\n', n)
    ;

%% Problem 2, Part ii
epsilon = 10e-10; % rad

function E = NewtonRaphson(M,e,epsilon)
    E = M;

    while true
        fE = E - e*sin(E) - M;
        fprimeE = 1 - e*cos(E);
        delta = -fE/fprimeE;
        E_next = E+delta;
        if abs(delta)<epsilon
            break;
        end

        E = E_next;
    end
end

%% Problem 2, Part iii
a = 17.834 * AU; % km
e = 0.967;
epsilon = 1e-10;

%% Problem 2, Part iv
close all;
a = 17.834 * AU; % km
e = 0.967;
epsilon = 1e-10;
DU_sun = R_sun;
TU_sun = sqrt(DU_sun^3/mu_sun); %

T = 2*pi*sqrt(a^3/mu_sun); % seconds
n = 2*pi/T; % rad/s
dt = T/1000;
time = 0:dt:T;

x = out.X;
y = out.Y;

n = length(x);
num_points = 25;
```

```
points = round(linspace(1,n,num_points));
x_points = x(points);
y_points = y(points);

%assignin('base', 'dt', dt);
%assignin('base', 'a', a);
%assignin('base', 'e', e);
%assignin('base', 'n', n);
%assignin('base', 'epsilon', epsilon);

%open_system('HalleysCometModel');

%simOut = sim('HalleysCometModel');

%outputData = simOut.get('yout');

%t = outputData.time;
%x_timeseries = outputData.get('x').Values;        % Extract 'x'
    data
%y_timeseries = outputData.get('y').Values;
%x_data = x_timeseries.Data;
%y_data = y_timeseries.Data;
figure(1);
hold on;
plot(x, y,'DisplayName','Comet Orbit');
scatter(x_points,y_points,'r','filled','DisplayName', '0.04T
   Orbit Traces');
axis equal;
xlabel('x (meters)');
ylabel('y (meters)');
title('Halley''s Comet Orbit');
grid on;
legend show;
hold off;
% Extract 'y' data
%y = outputData.signals.values;
%plot(t, y);
%xlabel('Time (s)');
%ylabel('Orbital Position');
%title('Halleys Comet Orbit Simulation');

%% Problem 3
close all;
% Part i
```

```
DU_earth = 6378; % km (spherical assumption)
TU_earth = sqrt(DU_earth^3/mu_earth); %
r = [8;0;0]*DU_earth; % km
v = [-1/sqrt(8); 1/sqrt(8); 0]* DU_earth/TU_earth; % km/s
r_norm = norm(r);
v_norm = norm(v);

epsilon = (v_norm^2/2) - (mu_earth/r_norm);
fprintf('The specific mechanical energy is: %.3f km^2/s^2\n',
   epsilon);

% Part ii
h = cross(r,v);
h_norm = norm(h);

p_debris = h_norm^2/mu_earth;
r_p_debris = p_debris/2;

e = (cross(v,h)/mu_earth) - r/r_norm;

fprintf('The position of periapsis for the debris is: %.3f km\n
   ', r_p_debris);
fprintf('The threshold position is: %.3f km\n', DU_earth+19134)
   ;

% Part iii
nu = 0:1:360; % degrees, true anomaly
e_debris = 1;

r_ECI = p_debris ./(1+e_debris*cosd(nu));
x_parabola = -r_ECI.*sind(nu); % shift to get parabola in right
    orientation
y_parabola = r_ECI.*cosd(nu);

e_sat = 0;
r_p_sat = 19134+DU_earth; % km

a_sat = r_p_sat/(1-e_sat);
p_sat = a_sat*(1-e_sat^2);
r_ECI = p_sat ./(1+e_sat*cosd(nu));
x_circle = r_ECI.*cosd(nu);
y_circle = r_ECI.*sind(nu);

figure(1)
hold on
```

```
scatter(x_circle, y_circle, 10, 'g', 'filled', 'DisplayName', '
    Satellite');
scatter(x_parabola, y_parabola, 10, 'm', 'filled', 'DisplayName
    ', 'Debris');
scatter(0, 0, 50, 'k', 'filled', 'DisplayName', 'Central Body')
    ;
xlabel('X Position (km)')
ylabel('Y Position (km)')
grid on
legend show;
ylim([-2e5 2e5])
axis equal
hold off

% Part iv
r_p_debris = p_debris/2;
v_debris_p = sqrt(mu_earth * (2 / r_p_debris));

v_sat = sqrt(mu_earth * (2 / r_p_sat - 1 / (2*r_p_sat)));

v_rel = v_debris_p - v_sat;

fprintf('The relative velocity is: %.3f km/s in the -i
    direction\n', v_rel);

nu = acosd((p_debris/r_norm) -1);
fprintf('The true anomaly of the debris at the initial position
     is: -%.3f degrees\n', nu);

%% Problem 4
clear;
r_IJK = [3105.4128; -880.8531; 6368.9408]; % km
v_IJK = [-6.7276;-0.5612;3.2023]; % km/s

[a,e,i,RAAN,omega,nu] = ECI2OE(r_IJK,v_IJK);

fprintf('Envisat semi-major axis: %.3f km\n', a);
fprintf('Envisat eccentricity: %.3e. \n', e);
fprintf('Envisat inclination: %.3f degrees\n', i);
fprintf('Envisat RAAN: %.3f degrees\n', RAAN);
fprintf('Envisat argument of perigee: %.3f degrees\n', omega);
fprintf('Envisat true anomaly: %.3f degrees\n', nu);

function [a,e,i,RAAN,omega,nu] = ECI2OE(r_IJK,v_IJK)
    mu_earth = 3.986e5; % km^3/s^2
```

```
    r_norm = norm(r_IJK);
    r_i = r_IJK(1);
    r_j = r_IJK(2);
    r_k = r_IJK(3);
    v_norm = norm(v_IJK);
    h = cross(r_IJK,v_IJK);
    h_norm = norm(h);
    W = h/norm(h);
    W_i = W(1);
    W_j = W(2);
    W_k = W(3);
    i = atan2(sqrt(W_i^2 + W_j^2),W_k);
    RAAN = atan2(W_i,-W_j);
    p = h_norm^2/mu_earth;
    a = ((2/r_norm)-(v_norm^2/mu_earth))^(-1);
    n = sqrt(mu_earth/a^3);
    e = sqrt(1-p/a);
    E = atan2(dot(r_IJK,v_IJK)/(a^2*n),(1-r_norm/a));
    M = E - e*sin(E);
    nu = 2*atan2(sqrt(1+e)*tan(E/2),sqrt(1-e));
    u = atan2(r_k/sin(i),r_i*cos(RAAN)+r_j*sin(RAAN));
    omega = u - nu;
    i = rad2deg(i);
    RAAN = rad2deg(RAAN);
    omega = rad2deg(omega);
    nu = rad2deg(nu);
    if nu<0
        nu = 360 + nu;
    end
end
```