AAE 440: Spacecraft Attitude Dynamics

PS7

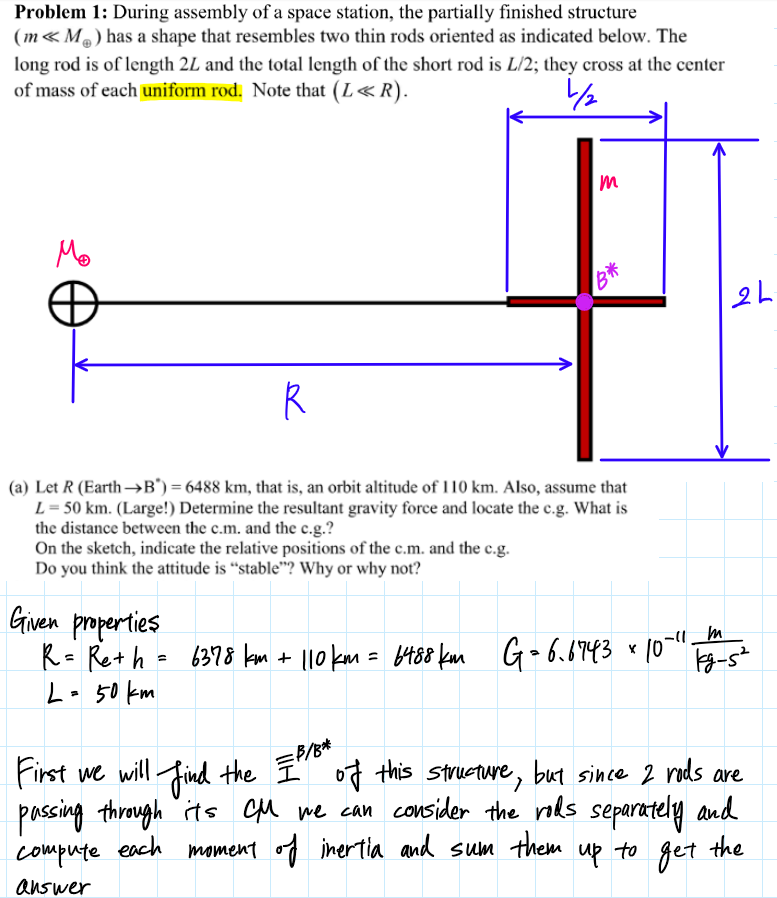
Dr. Howell

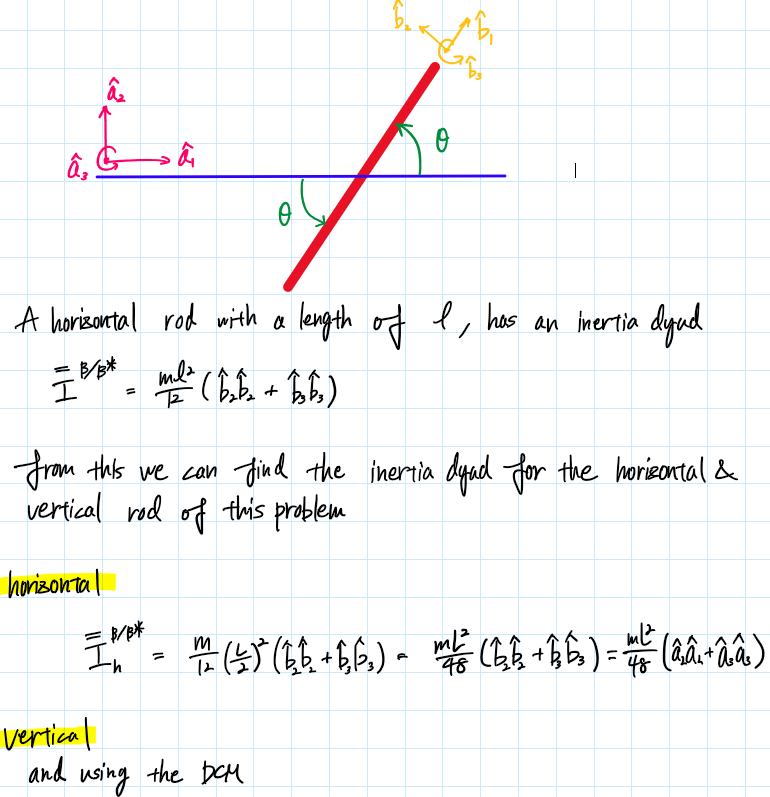
School of Aeronautical and Astronautical

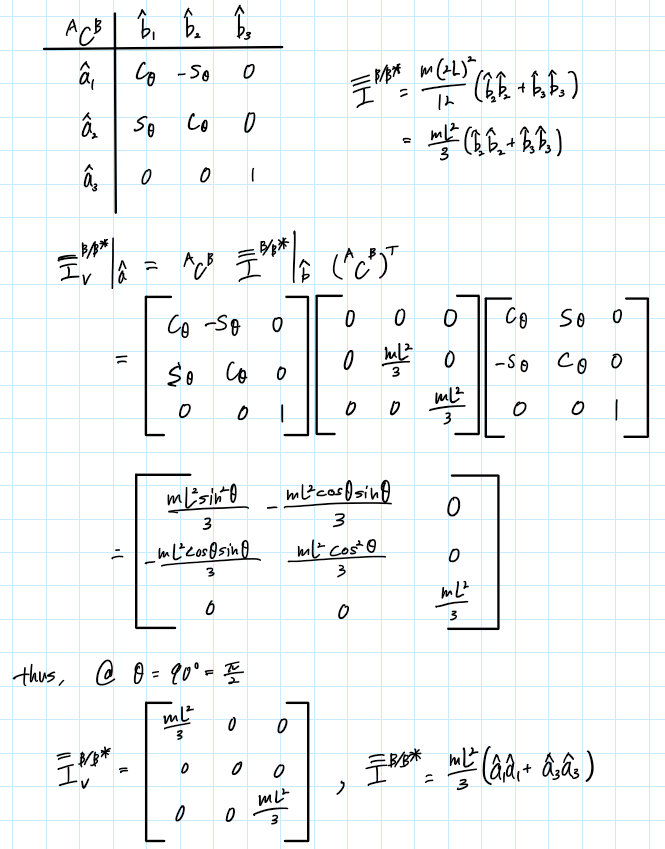
Purdue University

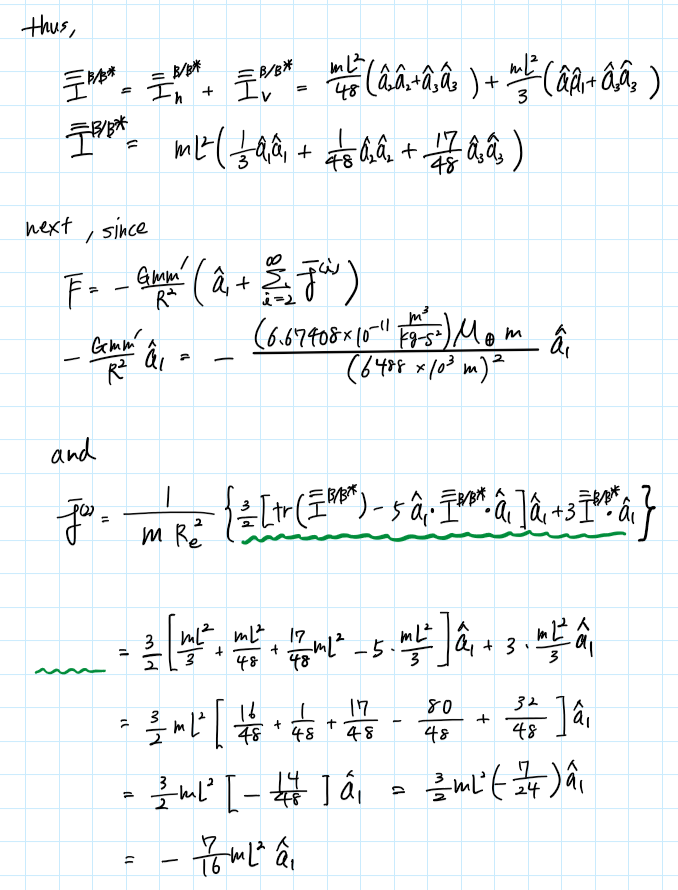
Tomoki Koike

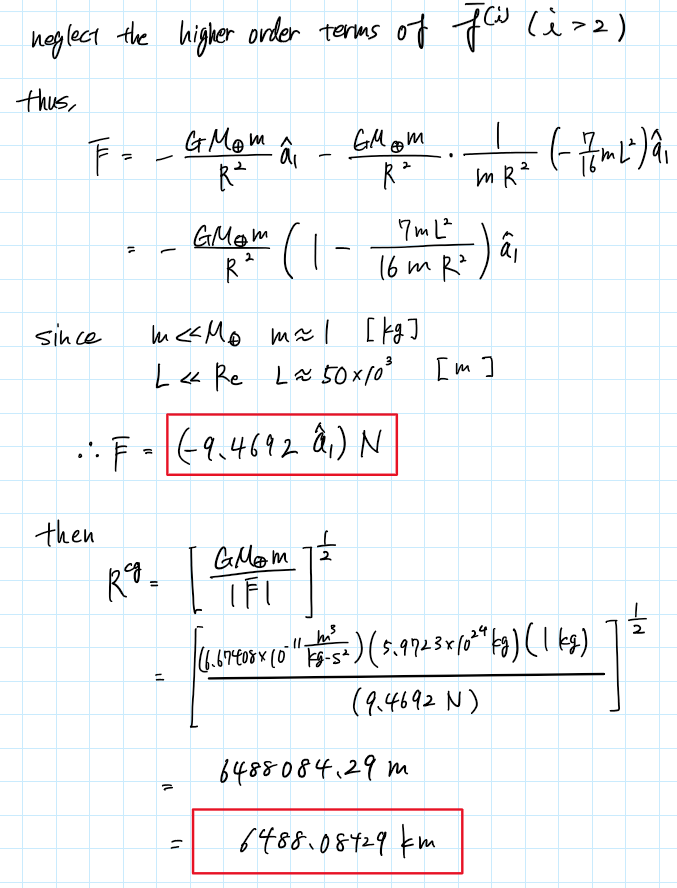
March 13, 2020

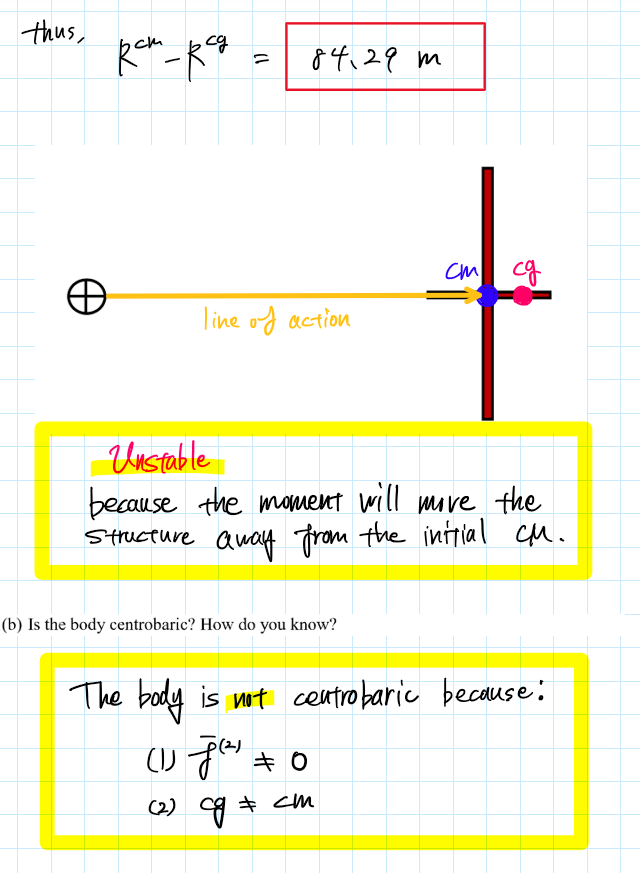


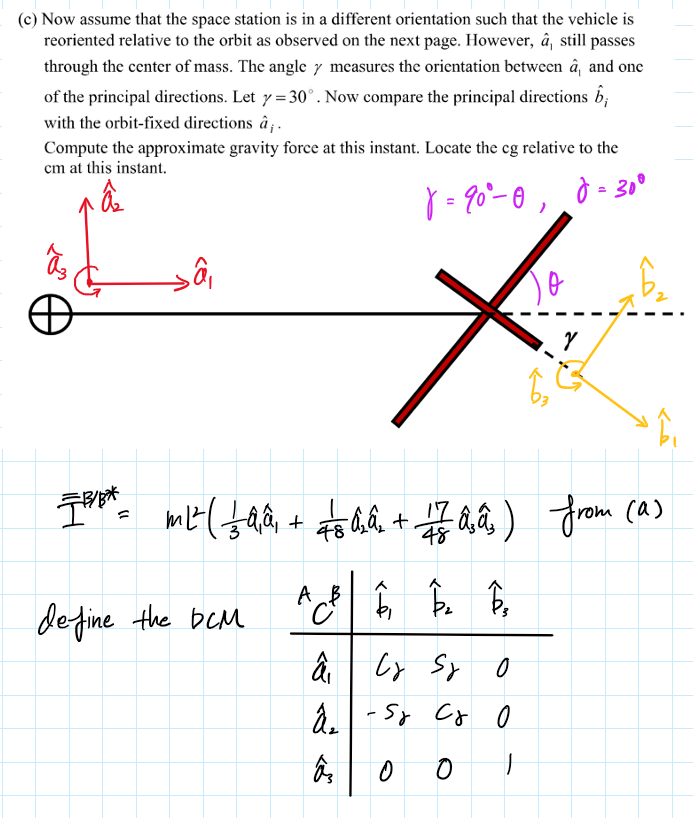


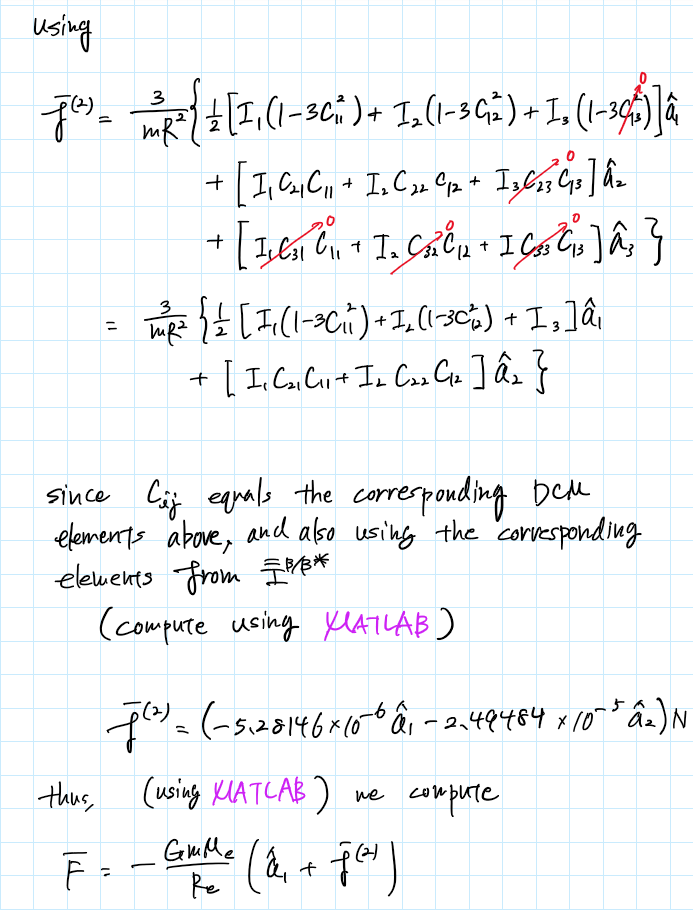


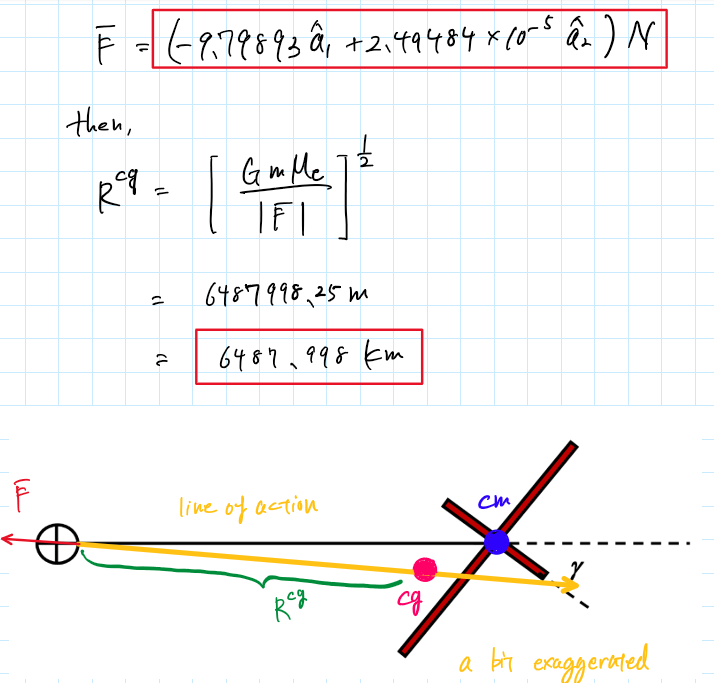


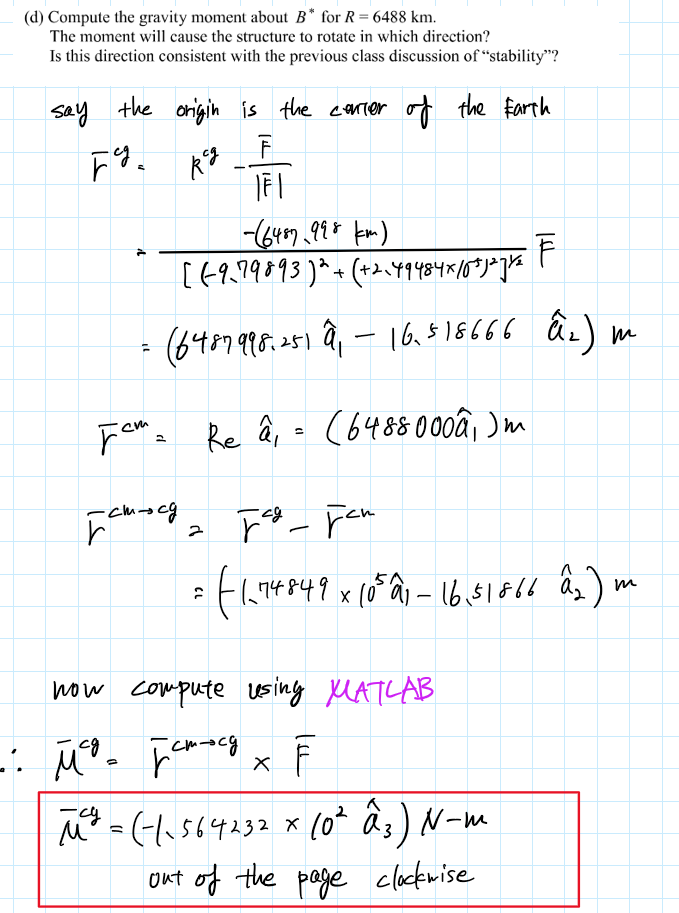


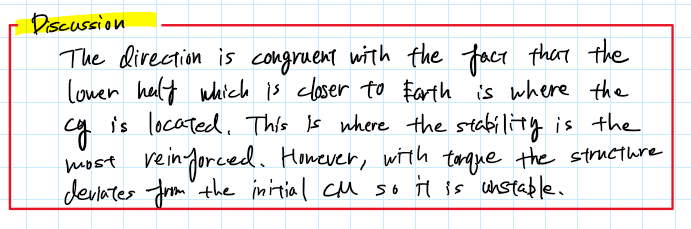


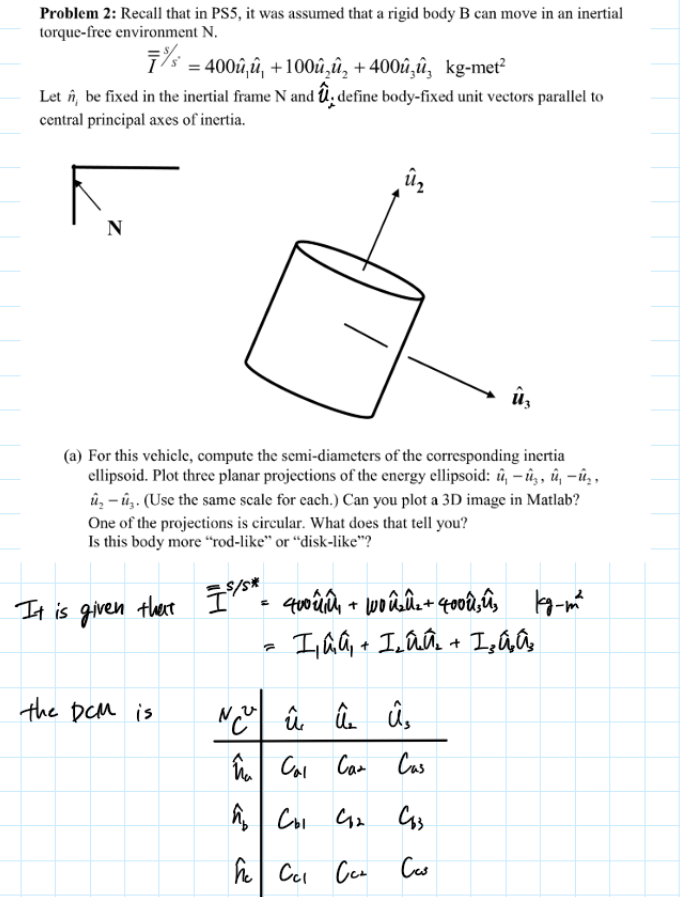


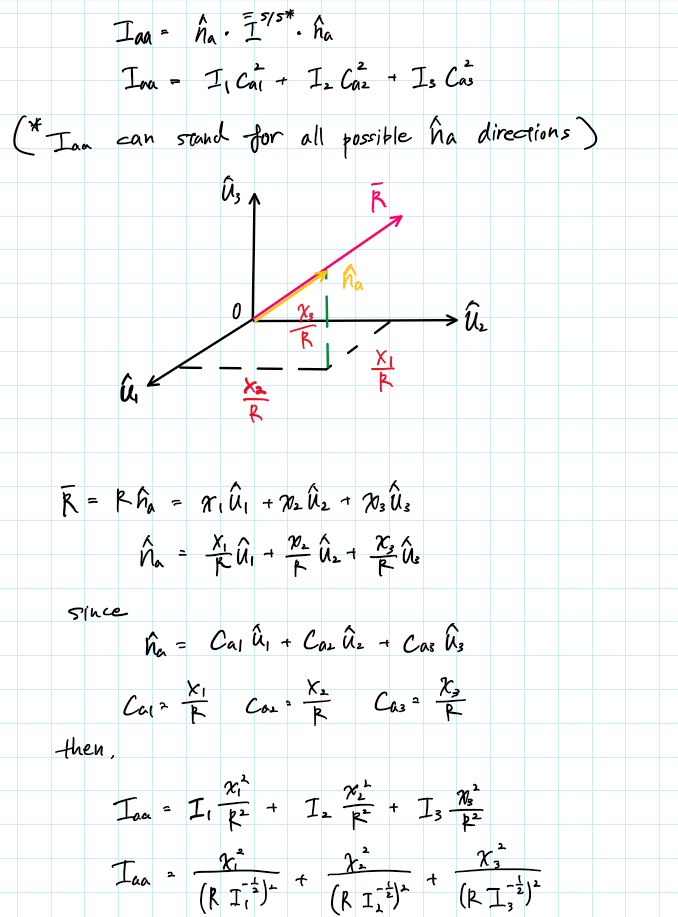


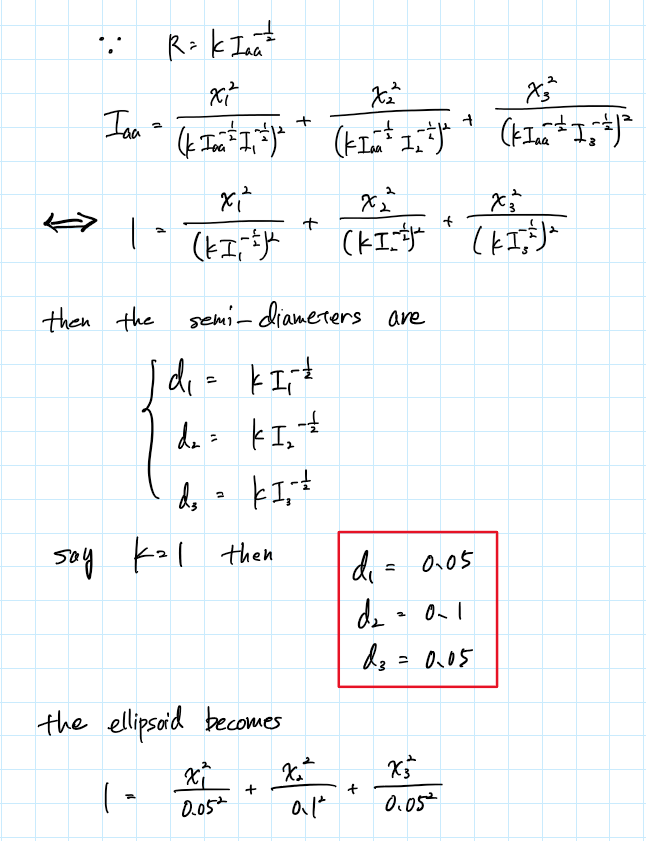


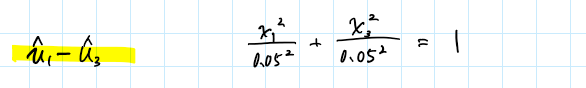


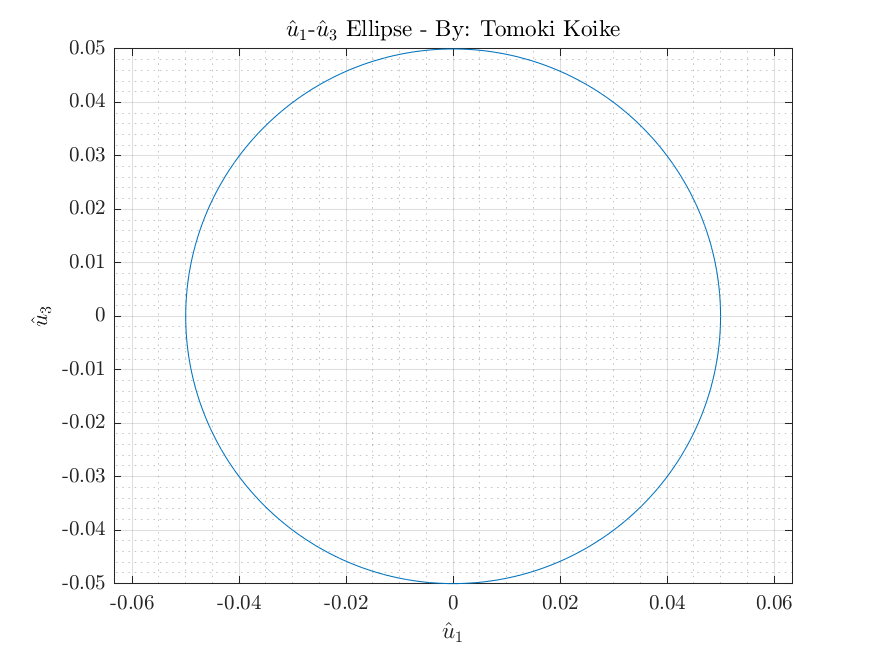


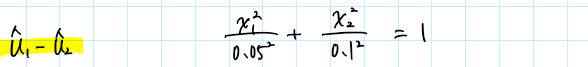


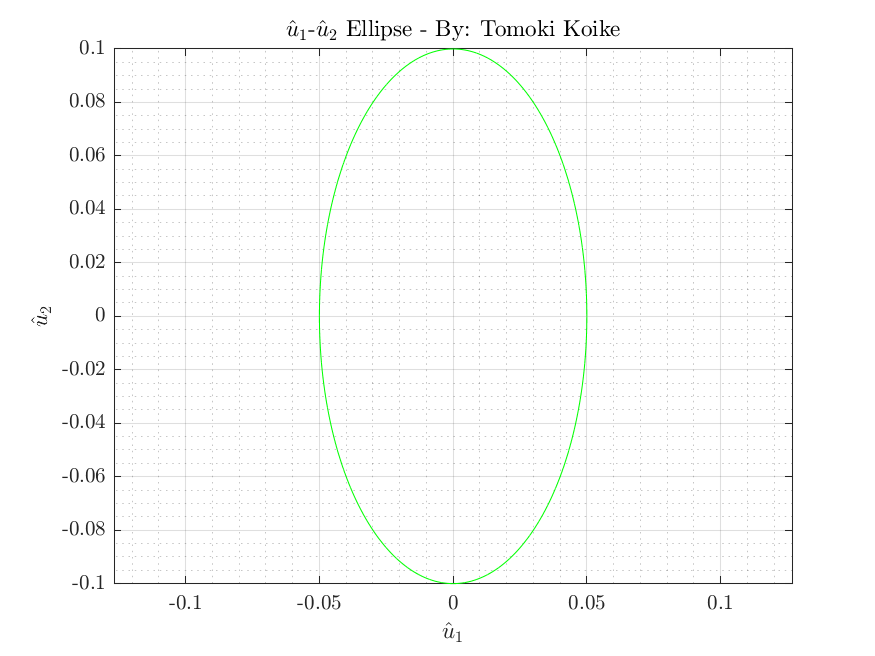


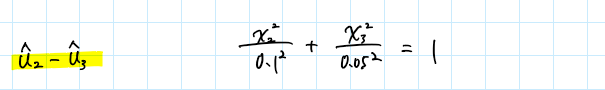


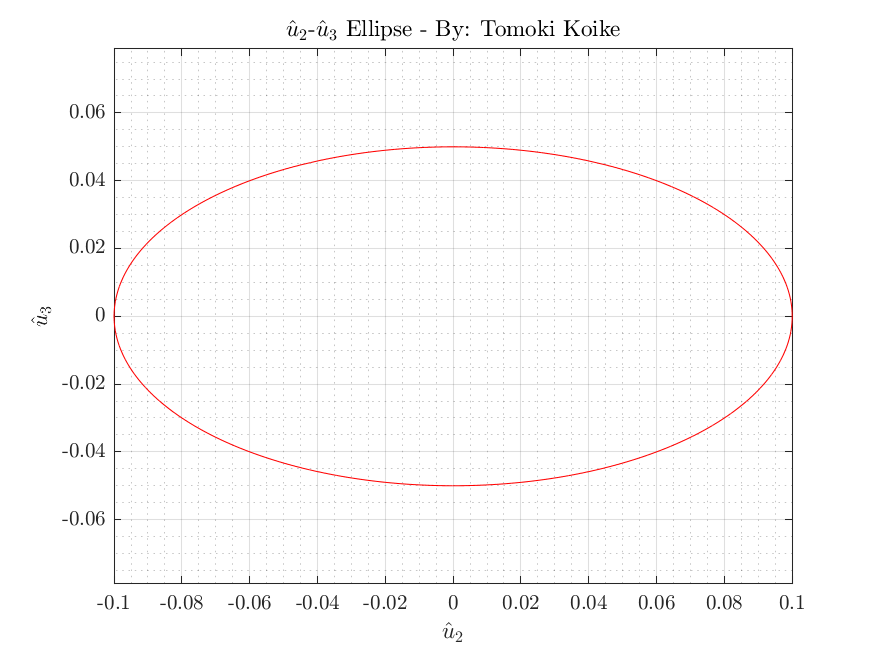




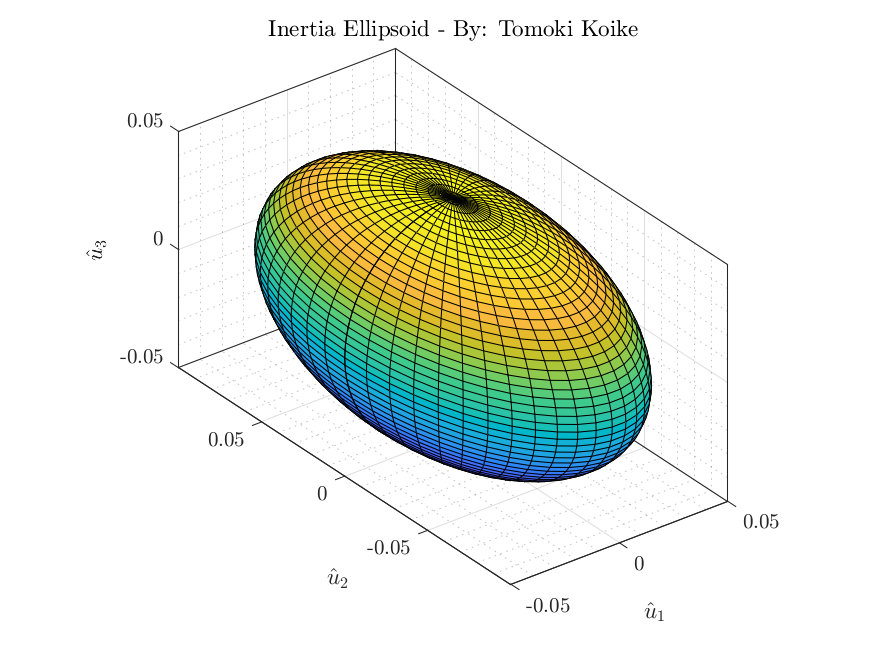


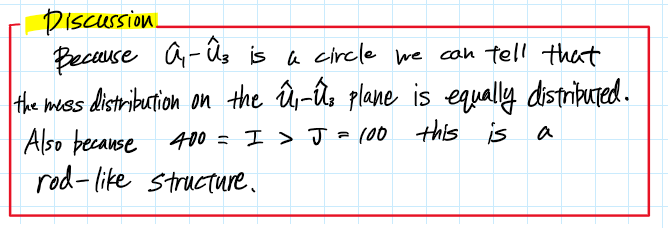


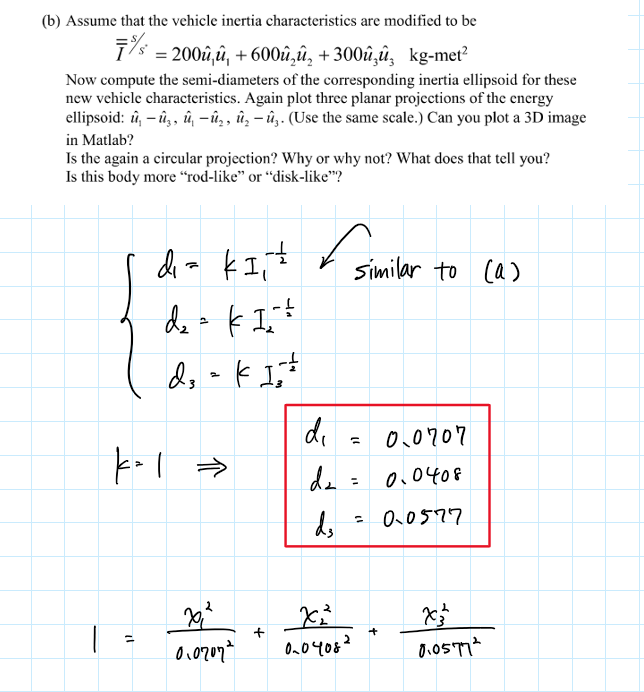




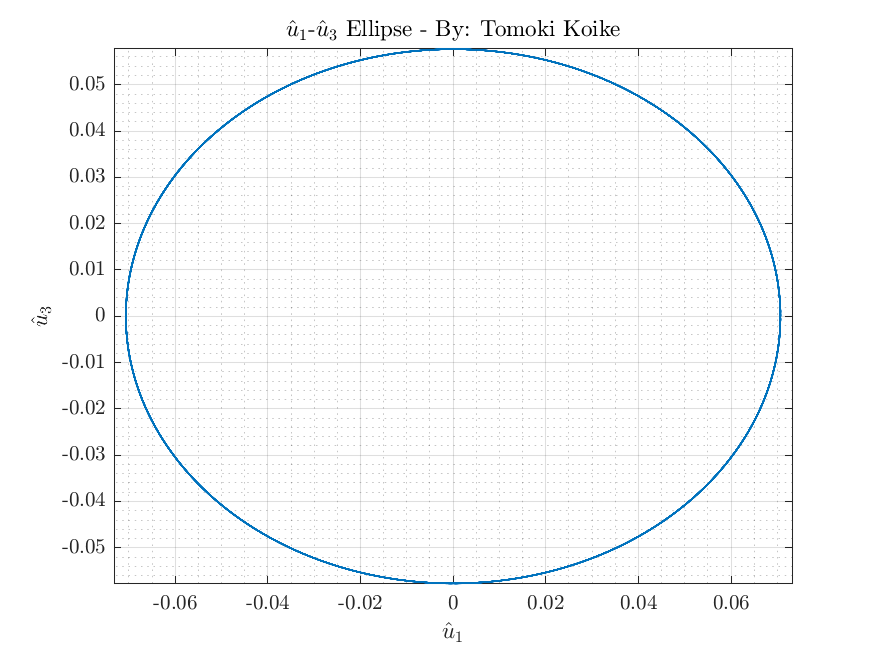




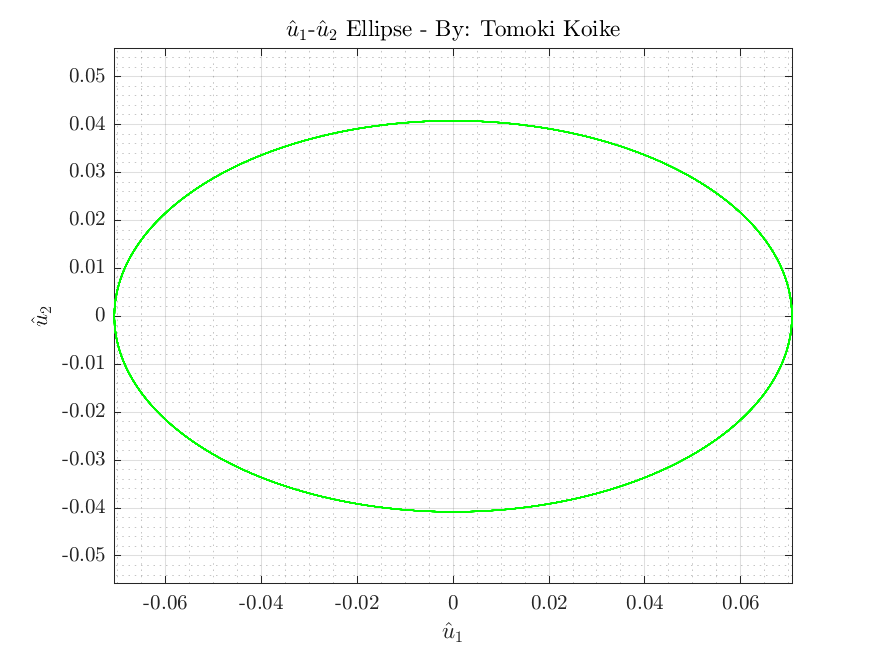




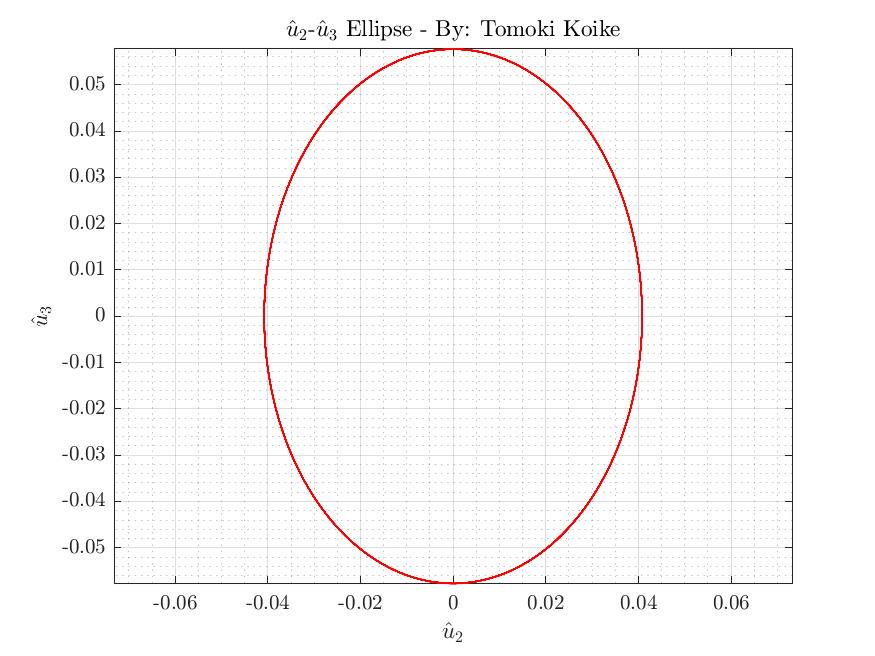




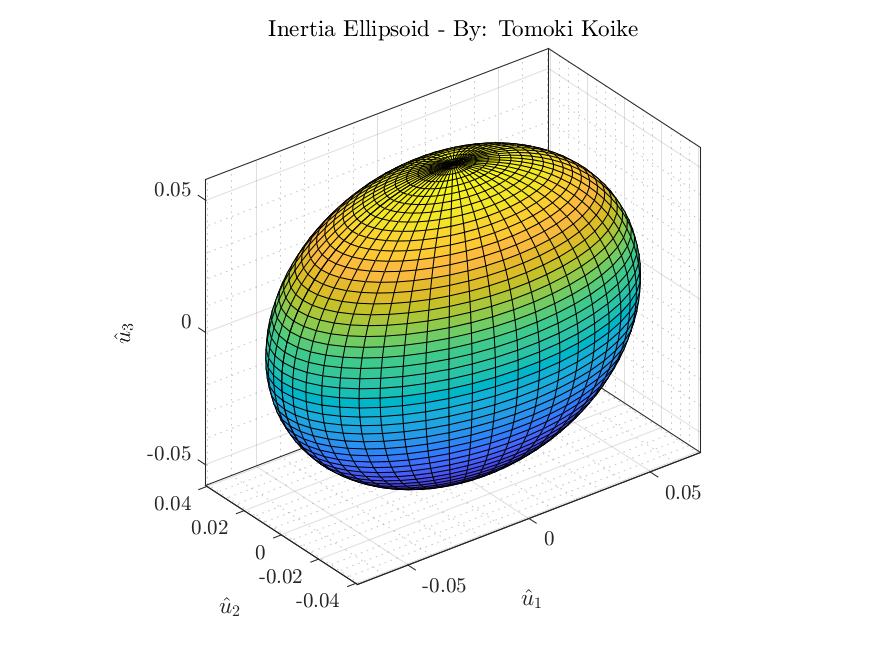


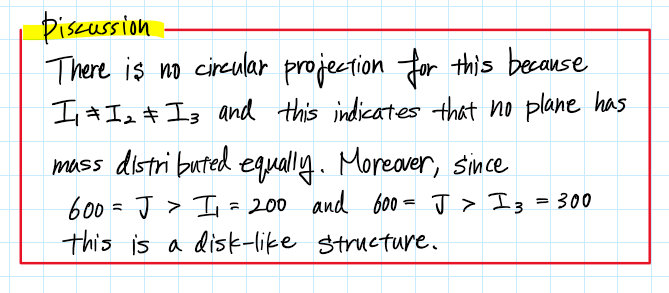


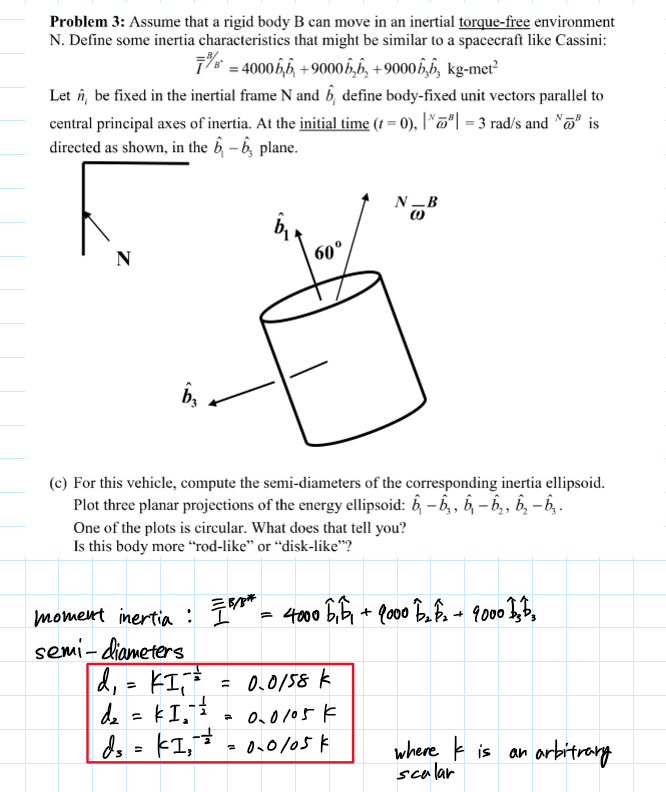




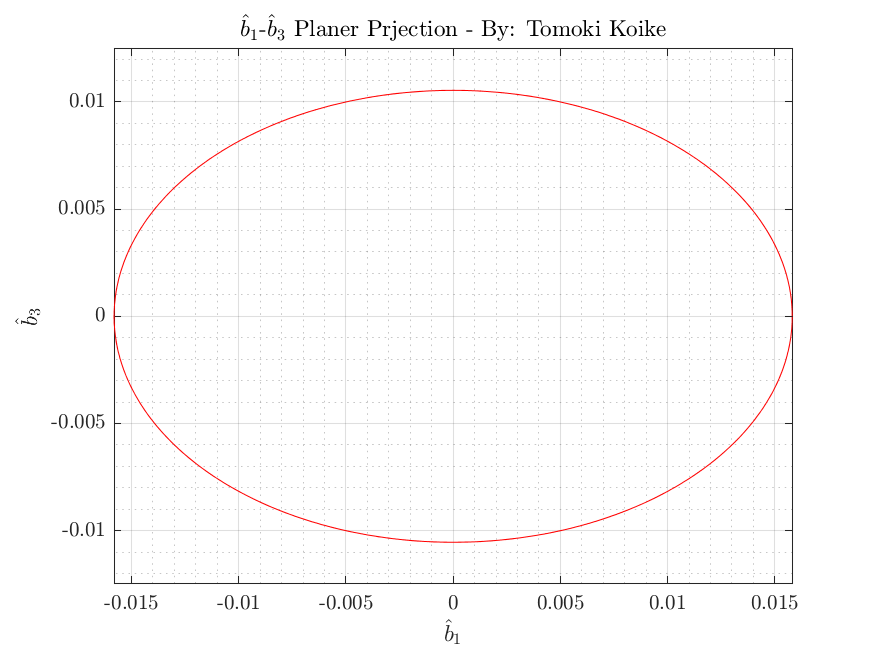














A close up of a map

Description automatically generated



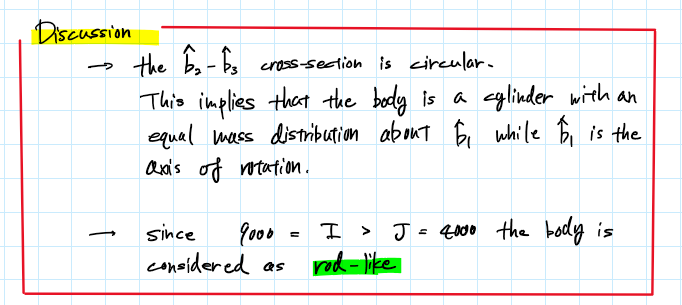
A close up of a device

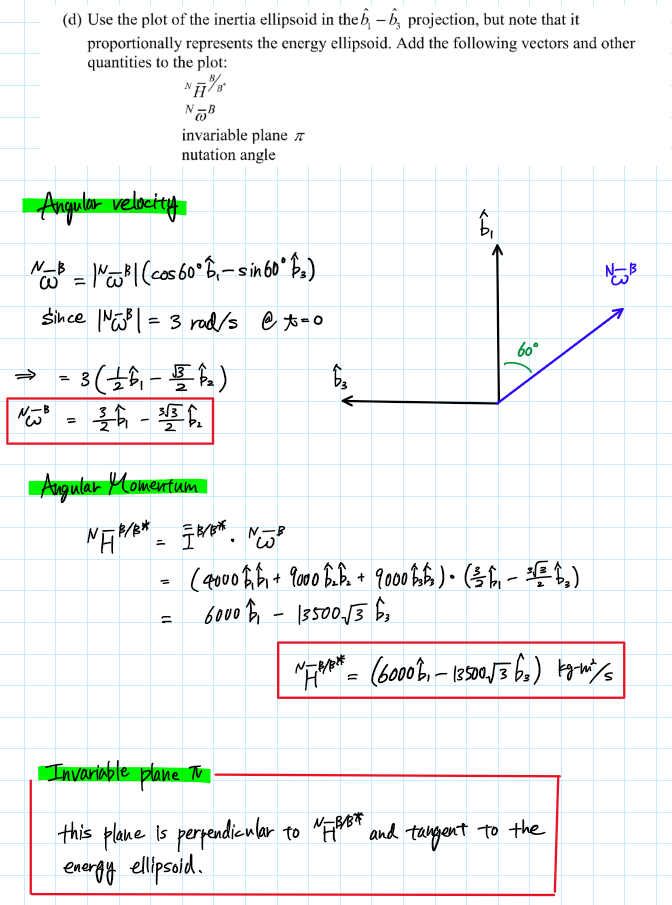
Description automatically generated

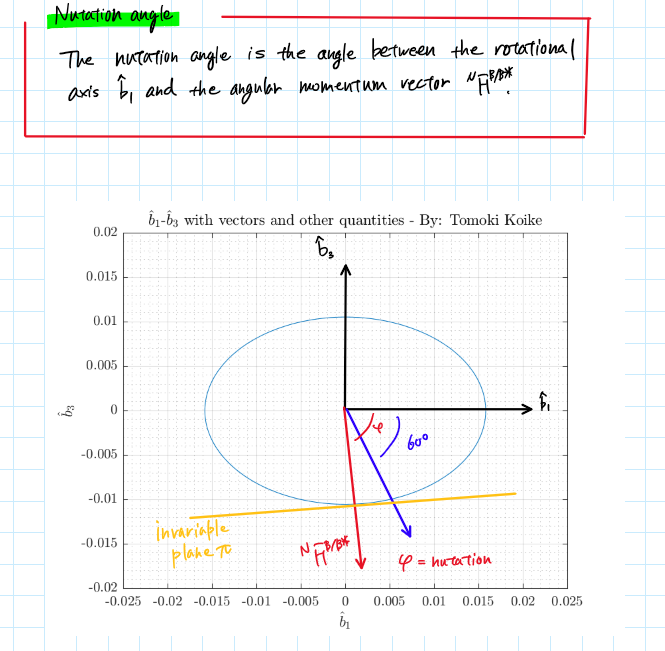


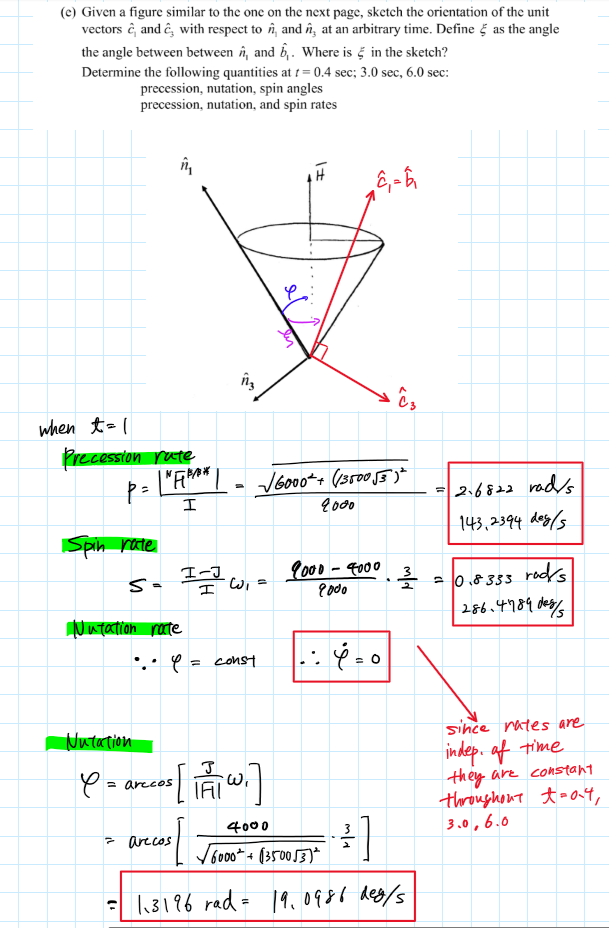
A close up of a logo

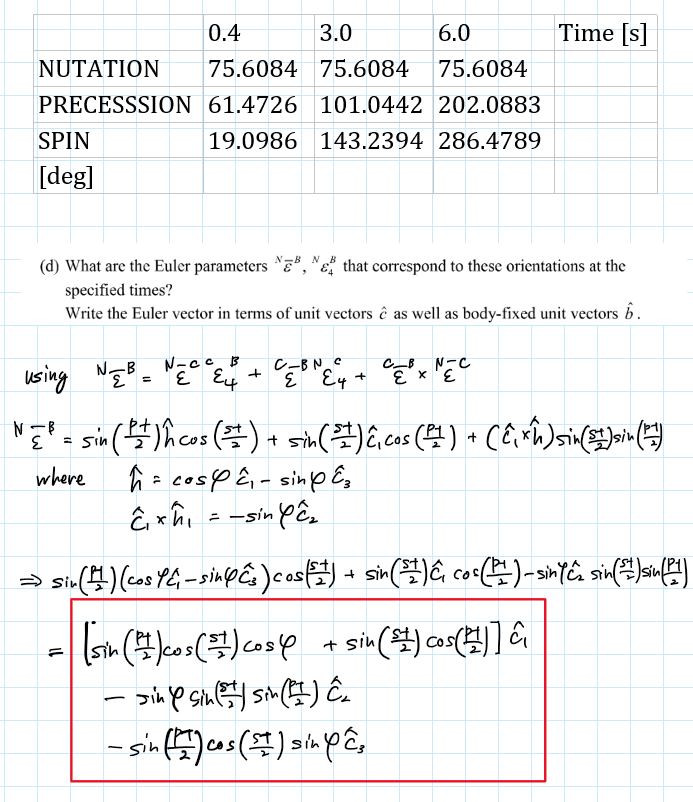
Description automatically generated

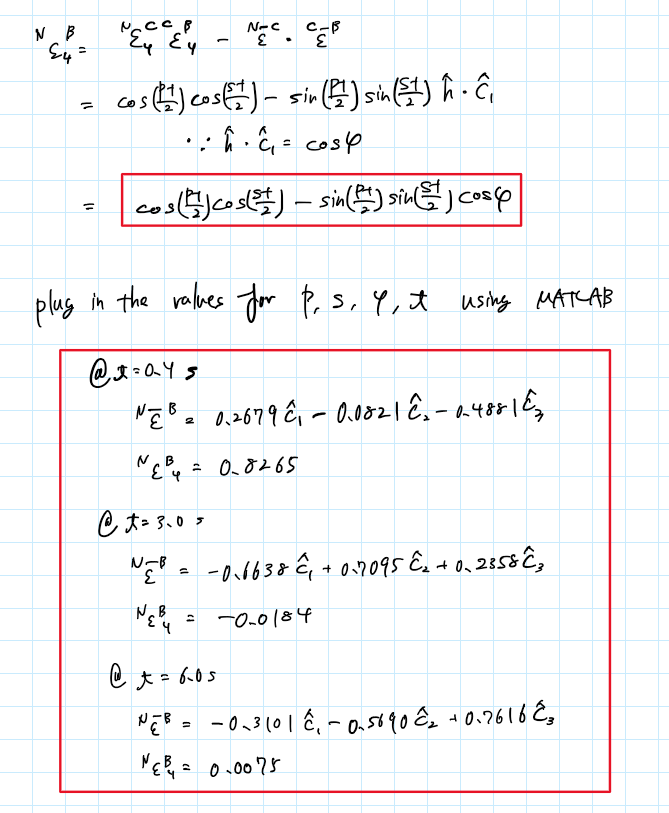


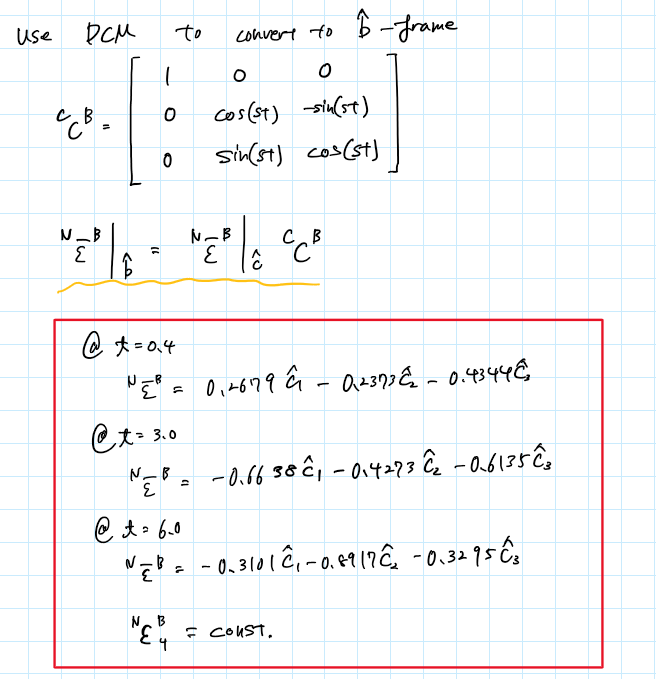


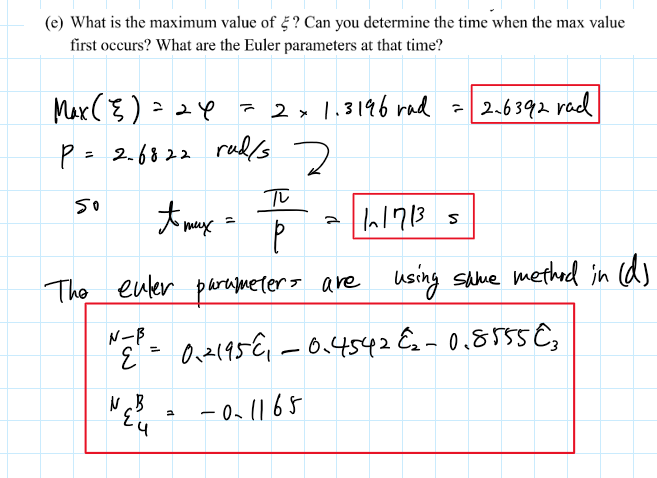


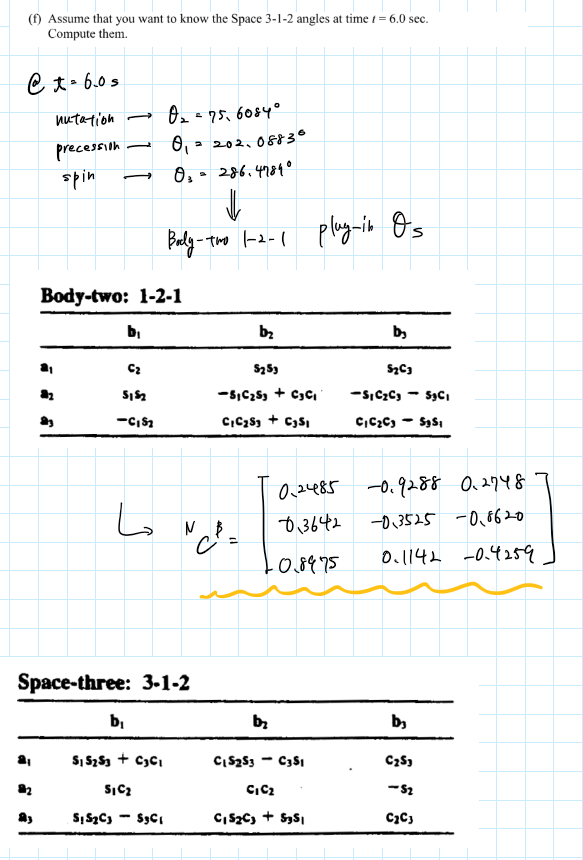


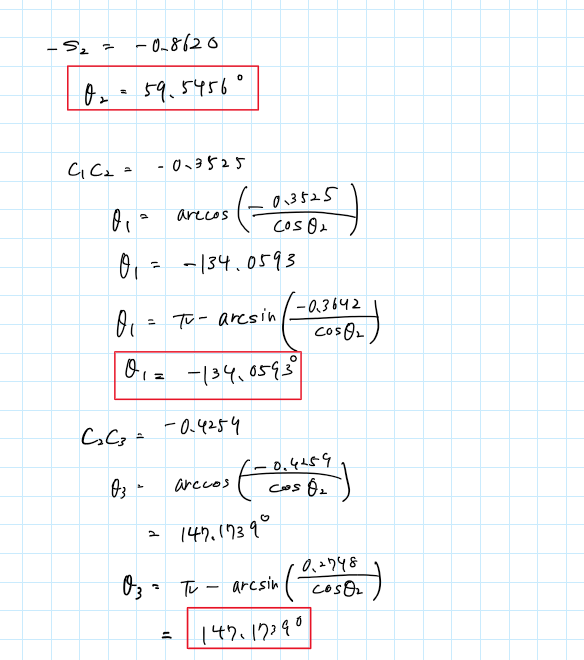












Appendix

**AAE440 PS7 Problem 1**

clear all; close all; clc;

format long e

digits(100)

syms m L theta

I\_rod = [0 0 0; 0 m\*L^2/3 0; 0 0 m\*L^2/3];

C\_AB = [cos(theta) -sin(theta) 0; sin(theta) cos(theta) 0; 0 0 1];

I\_h = C\_AB\*I\_rod\*transpose(C\_AB)

I\_h = subs(I\_h, theta, pi/2)

I\_v = subs(I\_rod, theta, pi/2)

clear all;

**(a) & (b)**

% Define given properties

M\_e = 5.9723e24; % [kg]

R\_e = 6378e3; % [m]

R = R\_e + 110e3

G = 6.6743015e-11; % [m3kg-1s-2]

m = 1;

L = 50e3;

% Resultant gravity force

m\_horz = m;

m\_vert = m;

l\_horz = L/2;

l\_vert = 2\*L;

I\_horz = m\_horz\*l\_horz^2/12\*[0 0 0; 0 1 0; 0 0 1];

I\_vert = m\_vert\*l\_vert^2/12\*[1 0 0; 0 0 0; 0 0 1];

I\_body = (I\_horz + I\_vert);

F\_g = resultant\_Gforce(M\_e,m,R,I\_body)

R\_cg = distance2cg(M\_e,m,F\_g)

delta\_cm\_cg = R - R\_cg

**(c)**

gamma = -30; % [deg]

C\_AB = [ cosd(gamma) sind(gamma) 0;

-sind(gamma) cosd(gamma) 0;

0 0 1];

F\_g\_new = resultant\_Gforce\_with\_DCM(M\_e,m,R,I\_body,C\_AB)

R\_cg\_new = distance2cg(M\_e,m,F\_g\_new)

**(d)**

r\_cg = -F\_g\_new/norm(F\_g\_new)\*R\_cg\_new

r\_cm = [R 0 0]

r\_cm\_cg = r\_cg - r\_cm

Mmt\_cg = cross(r\_cm\_cg,F\_g\_new)

**AAE440 PS7 Problem 2**

clear all; close all; clc;

fdir = 'C:\Users\Tomo\Desktop\studies\2020-Spring\AAE440\MATLAB\outputs\HW7';

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

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

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

% ellipsoid

I\_body = [400 0 0; 0 100 0; 0 0 400];

k = 1;

d1 = k\*I\_body(1,1)^(-0.5)

d2 = k\*I\_body(2,2)^(-0.5)

d3 = k\*I\_body(3,3)^(-0.5)

% Plotting

theta = 0:0.1:360;

theta = deg2rad(theta);

% u1-u3

fig1 = figure("Renderer","painters");

plot(d1\*cos(theta), d3\*sin(theta))

title('$\hat{u}\_1$-$\hat{u}\_3$ Ellipse - By: Tomoki Koike')

xlabel('$\hat{u}\_1$')

ylabel('$\hat{u}\_3$')

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

saveas(fig1, fullfile(fdir,'u1\_u3\_ellipse.png'));

% u1-u2

fig2 = figure("Renderer","painters");

plot(d1\*cos(theta), d2\*sin(theta), 'g')

title('$\hat{u}\_1$-$\hat{u}\_2$ Ellipse - By: Tomoki Koike')

xlabel('$\hat{u}\_1$')

ylabel('$\hat{u}\_2$')

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

saveas(fig2, fullfile(fdir,'u1\_u2\_ellipse.png'));

% u1-u2

fig3 = figure("Renderer","painters");

plot(d2\*cos(theta), d3\*sin(theta), 'r')

title('$\hat{u}\_2$-$\hat{u}\_3$ Ellipse - By: Tomoki Koike')

xlabel('$\hat{u}\_2$')

ylabel('$\hat{u}\_3$')

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

saveas(fig3, fullfile(fdir,'u2\_u3\_ellipse.png'));

% Ellipsoid

fig4 = figure("Renderer","painters");

ellipsoid(0,0,0,d1,d2,d3,50)

title('Inertia Ellipsoid - By: Tomoki Koike')

xlabel('$\hat{u}\_1$')

ylabel('$\hat{u}\_2$')

zlabel('$\hat{u}\_3$')

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

saveas(fig4, fullfile(fdir,'ellipsoid.png'));

clear all; close all;

fdir = 'C:\Users\Tomo\Desktop\studies\2020-Spring\AAE440\MATLAB\outputs\HW7';

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

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

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

% ellipsoid

I\_body = [200 0 0; 0 600 0; 0 0 300];

k = 1;

d1 = k\*I\_body(1,1)^(-0.5)

d2 = k\*I\_body(2,2)^(-0.5)

d3 = k\*I\_body(3,3)^(-0.5)

% Plotting

theta = 0:0.1:360;

% u1-u3

fig5 = figure("Renderer","painters");

plot(d1\*cos(theta), d3\*sin(theta))

title('$\hat{u}\_1$-$\hat{u}\_3$ Ellipse - By: Tomoki Koike')

xlabel('$\hat{u}\_1$')

ylabel('$\hat{u}\_3$')

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

saveas(fig5, fullfile(fdir,'u1\_u3\_ellipse2.png'));

% u1-u2

fig6 = figure("Renderer","painters");

plot(d1\*cos(theta), d2\*sin(theta), 'g')

title('$\hat{u}\_1$-$\hat{u}\_2$ Ellipse - By: Tomoki Koike')

xlabel('$\hat{u}\_1$')

ylabel('$\hat{u}\_2$')

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

saveas(fig6, fullfile(fdir,'u1\_u2\_ellipse2.png'));

% u1-u2

fig7 = figure("Renderer","painters");

plot(d2\*cos(theta), d3\*sin(theta), 'r')

title('$\hat{u}\_2$-$\hat{u}\_3$ Ellipse - By: Tomoki Koike')

xlabel('$\hat{u}\_2$')

ylabel('$\hat{u}\_3$')

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

saveas(fig7, fullfile(fdir,'u2\_u3\_ellipse2.png'));

% Ellipsoid

fig8 = figure("Renderer","painters");

ellipsoid(0,0,0,d1,d2,d3,50)

title('Inertia Ellipsoid - By: Tomoki Koike')

xlabel('$\hat{u}\_1$')

ylabel('$\hat{u}\_2$')

zlabel('$\hat{u}\_3$')

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

saveas(fig8, fullfile(fdir,'ellipsoid2.png'));

**AAE 440 PS7 p3**

clear all; close all; clc;

fdir = 'C:\Users\Tomo\Desktop\studies\2020-Spring\AAE440\MATLAB\outputs\HW7';

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

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

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

% Draw Arrow Function

drawArrow = @(x,y,varargin) quiver( x(1),y(1),x(2)-x(1),y(2)-y(1),0, varargin{:});

% Setting given properties

I\_body = [4000 0 0; 0 9000 0; 0 0 9000];

k = 1;

d\_1 = k\*I\_body(1,1)^(-0.5);

d\_2 = k\*I\_body(2,2)^(-0.5);

d\_3 = k\*I\_body(3,3)^(-0.5);

theta = 0:0.01:2\*pi;

**<a>**

% Plotting

% b1-b3

fig1 = figure("Renderer","painters");

plot(d\_1\*cos(theta), d\_3\*sin(theta), 'r')

title('$\hat{b}\_1$-$\hat{b}\_3$ Planer Prjection - By: Tomoki Koike')

xlabel('$\hat{b}\_1$')

ylabel('$\hat{b}\_3$')

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

saveas(fig1, fullfile(fdir,'b1\_b3\_ellipse3.png'));

% b1-b2

fig2 = figure("Renderer","painters");

plot(d\_1\*cos(theta), d\_2\*sin(theta), 'b')

title('$\hat{b}\_1$-$\hat{b}\_2$ Planer Prjection - By: Tomoki Koike')

xlabel('$\hat{b}\_1$')

ylabel('$\hat{b}\_2$')

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

saveas(fig2, fullfile(fdir,'b1\_b2\_ellipse3.png'));

% b2-b3

fig3 = figure("Renderer","painters");

plot(d\_2\*cos(theta), d\_3\*sin(theta), 'g')

title('$\hat{b}\_2$-$\hat{b}\_3$ Planer Prjection - By: Tomoki Koike')

xlabel('$\hat{b}\_2$')

ylabel('$\hat{b}\_3$')

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

saveas(fig3, fullfile(fdir,'b2\_b3\_ellipse3.png'));

% Ellipsoid

fig4 = figure("Renderer","painters");

ellipsoid(0,0,0,d\_1,d\_2,d\_3,100)

title('Inertia Ellipsoid - By: Tomoki Koike')

xlabel('$\hat{b}\_1$')

ylabel('$\hat{b}\_2$')

zlabel('$\hat{b}\_3$')

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

saveas(fig4, fullfile(fdir,'ellipsoid3.png'));

**<b>**

% Defining properties

T = 0; % Torque [N-m]

I\_cm = [4000 0 0; 0 9000 0; 0 0 9000]; % Inertia Dyadic [kg-m2]

I = 9000;

J = 4000;

% Defining the angular velocity

w\_mag = 3; % Magnitude of angular velocity

ang = deg2rad(60); % [rad]

w\_NB = [w\_mag\*cos(ang) 0 -w\_mag\*sin(ang)];

% Angular Momentum

H\_NB = I\_cm\*w\_NB';

% Replotting b1-b3 and adding vectors and quantities

fig5 = figure("Renderer","painters");

plot(d\_1\*cos(theta), d\_3\*sin(theta))

title('$\hat{b}\_1$-$\hat{b}\_3$ with vectors and other quantities - By: Tomoki Koike')

xlabel('$\hat{b}\_1$')

ylabel('$\hat{b}\_3$')

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

xlim([-0.025, 0.025]); ylim([-0.020, 0.020]);

saveas(fig5, fullfile(fdir,'b1\_b3\_ellipse\_new.png'));

**<c>**

% angles @ t = 1

p = norm(H\_NB)/I; % precession rate

s = (I-J)/I\*w\_NB(1); % spin rate

phi = acos(J/norm(H\_NB)\*w\_NB(1)); % nutation angle = constant

% t = 0.4, 3.0, 6.0

t = [0.4 3.0 6.0];

p\_rad = p.\*t;

s\_rad = s.\*t;

phi\_deg = rad2deg(phi)

p\_deg = rad2deg(p\_rad)

s\_deg = rad2deg(s\_rad)

**<d>**

% Euler Parameters

e\_04\_c = EulerPara\_from\_OrientAngs(0.4,p,s,phi);

e\_30\_c = EulerPara\_from\_OrientAngs(3.0,p,s,phi);

e\_60\_c = EulerPara\_from\_OrientAngs(6.0,p,s,phi);

% t = 0.4s

t = 0.4;

e\_04\_c = [e\_04\_c(1) e\_04\_c(2) e\_04\_c(3)];

C\_CB\_04 = [1 0 0; 0 cos(s\*t) -sin(s\*t); 0 sin(s\*t) cos(s\*t)];

e\_04\_b = e\_04\_c\*C\_CB\_04

% t = 3.0s

t = 3.0;

e\_30\_c = [e\_30\_c(1) e\_30\_c(2) e\_30\_c(3)];

C\_CB\_30 = [1 0 0; 0 cos(s\*t) -sin(s\*t); 0 sin(s\*t) cos(s\*t)];

e\_30\_b = e\_30\_c\*C\_CB\_30

% t = 6.0s

t = 6.0;

e\_60\_c = [e\_60\_c(1) e\_60\_c(2) e\_60\_c(3)];

C\_CB\_60 = [1 0 0; 0 cos(s\*t) -sin(s\*t); 0 sin(s\*t) cos(s\*t)];

e\_60\_b = e\_60\_c\*C\_CB\_60

**<e>**

% maximum of zeta

zeta = 2\*phi;

% time corresponding to the maximum zeta

t\_maxzeta = pi/p;

% Euler parameters corresponding

e\_maxzeta = EulerPara\_from\_OrientAngs(t\_maxzeta,p,s,phi)

**<f>**

% t = 6.0 s

C = DCM\_Body(1, 2, 1, p\_rad(3), phi, s\_rad(3));

theta2 = asin(-C(2,3))

theta1 = -acos(C(2,2)/cos(theta2))

theta1 = -pi-asin(C(2,1)/cos(theta2))

theta3 = pi-asin(C(1,3)/cos(theta2))

theta3 = acos(C(3,3)/cos(theta2))

theta1 = rad2deg(theta1)

theta2 = rad2deg(theta2)

theta3 = rad2deg(theta3)

function epsilons = EulerPara\_from\_OrientAngs(t,p,s,phi)

e1 = cos(phi)\*sin(p\*t/2)\*cos(s\*t/2)+cos(p\*t/2)\*sin(s\*t/2);

e2 = -sin(phi)\*sin(s\*t/2)\*sin(p\*t/2);

e3 = -sin(phi)\*sin(p\*t/2)\*cos(s\*t/2);

e4 = cos(p\*t/2)\*cos(s\*t/2)-cos(phi)\*sin(p\*t/2)\*sin(s\*t/2);

epsilons = [e1 e2 e3 e4];

end