AAE 440: Spacecraft Attitude Dynamics

PS5*: Dynamics and Kinematic DE Simulation

Dr. Howell

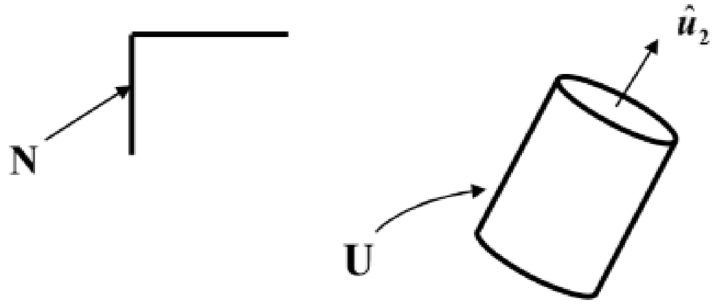
Tomoki Koike

Monday March 2, 2020

Problem 1: Recall the class discussion concerning Notes H. In this problem set, we will develop an algorithm to complete the analysis in Notes H.

An axisymmetric rigid body U (spacecraft) can move in an inertial reference frame N. Let $\hat{n}_i$ and $\hat{u}_i$ be unit vectors fixed in N and U, respectively. Assume that the body is axisymmetric such that the inertia dyadic is

$$\bar{\bar{I}}^{S/S^*} = 400\hat{u}_1\hat{u}_1 + 100\hat{u}_2\hat{u}_2 + 400\hat{u}_3\hat{u}_3 \ \text{kg-m}^2$$



(a) Assume that U is subject to a constant torque $\bar{T}$ but it now acts parallel to the transverse axis $\hat{u}_1$. Let Euler parameters be defined as the kinematic variables and write the kinematic equations of motion. Also derive the dynamic equations of motion. Carefully compare the angular velocity measure numbers that appear in the dynamic and kinematic differential equations. Which angular velocity is incorporated, i.e., between which two frames? Which unit vectors are used to define the measure numbers? Are they the same in both sets of equations?

The Kinematic Equations of Motion is

if ${}^{N}\omega_i^{U} = \omega_i$, ${}^{N}\varepsilon_i^{U} = \varepsilon_i$, and ${}^{N}\dot{\varepsilon}^{U} = \dot{\varepsilon}_i$

$$\boxed{\begin{aligned} \dot{\varepsilon}_1 &= 0.5(\omega_1\varepsilon_4 - \omega_2\varepsilon_3 + \omega_3\varepsilon_2) \\ \dot{\varepsilon}_2 &= 0.5(\omega_1\varepsilon_3 + \omega_2\varepsilon_4 - \omega_3\varepsilon_1) \\ \dot{\varepsilon}_3 &= 0.5(-\omega_1\varepsilon_2 + \omega_2\varepsilon_1 + \omega_3\varepsilon_4) \\ \dot{\varepsilon}_4 &= -0.5(\omega_1\varepsilon_1 + \omega_2\varepsilon_2 + \omega_3\varepsilon_3) \end{aligned}}$$

Next we shall derive the Dynamic Equations of Motion.
First, we know that the angular momentum is by def.

$$\begin{aligned} {}^{N}\bar{H}^{U^*} &= \bar{\bar{I}}^{U/U^*} \cdot {}^{N}\bar{\omega}^{U} \\ &= (I\,\hat{u}_1\hat{u}_1 + J\,\hat{u}_2\hat{u}_2 + I\,\hat{u}_3\hat{u}_3) \cdot (\omega_1\hat{u}_1 + \omega_2\hat{u}_2 + \omega_3\hat{u}_3) \\ &= I\omega_1\hat{u}_1 + J\omega_2\hat{u}_2 + I\omega_3\hat{u}_3 \end{aligned}$$

and using BKE

$$\begin{aligned} \frac{{}^{N}d\,{}^{N}\bar{H}^{U^*}}{dt} &= \frac{{}^{U}d\,{}^{N}\bar{H}^{U^*}}{dt} + {}^{N}\bar{\omega}^{U} \times {}^{N}\bar{H}^{U^*} \\ &= I\dot{\omega}_1\hat{u}_1 + J\dot{\omega}_2\hat{u}_2 + I\dot{\omega}_3\hat{u}_3 \\ &\qquad (\omega_1\hat{u}_1 + \omega_2\hat{u}_2 + \omega_3\hat{u}_3) \times (I\omega_1\hat{u}_1 + J\omega_2\hat{u}_2 + I\omega_3\hat{u}_3) \end{aligned}$$

$$= I\dot{\omega}_1 \hat{u}_1 + J\dot{\omega}_2 \hat{u}_2 + I\dot{\omega}_3 \hat{u}_3$$
$$+ J\omega_1\omega_2 \hat{u}_3 - \cancel{I\omega_1\omega_3\hat{u}_2}$$
$$- I\omega_1\omega_2 \hat{u}_3 + I\omega_2\omega_3 \hat{u}_1$$
$$+ \cancel{I\omega_1\omega_3\hat{u}_2} - J\omega_2\omega_3 \hat{u}_1$$

$$= (I\dot{\omega}_1 + I\omega_2\omega_3 - J\omega_2\omega_3)\hat{u}_1$$
$$+ J\dot{\omega}_2 \hat{u}_2$$
$$+ (I\dot{\omega}_3 - I\omega_1\omega_2 + J\omega_1\omega_2)\hat{u}_3 \quad \cdots ①$$

Also we know from the problem that

$$\bar{T} = T\hat{u}_1 \iff \bar{T}^{u^*} = \frac{{}^N d \bar{H}^{N-u^*}}{dt} \quad \cdots ②$$

thus, ① = ②

$$\begin{cases} I\dot{\omega}_1 + (I-J)\omega_2\omega_3 = T \\ J\dot{\omega}_2 = 0 \\ I\dot{\omega}_3 - (I-J)\omega_1\omega_2 = 0 \end{cases}$$

$$\Longleftrightarrow \quad \boxed{\begin{aligned} \dot{\omega}_1 &= \frac{T}{I} - \frac{I-J}{I}\omega_2\omega_3 \\ \dot{\omega}_2 &= 0 \\ \dot{\omega}_3 &= \frac{I-J}{I}\omega_1\omega_2 \end{aligned}}$$

Comparing the 2 answers

(1) The angular velocity, ${}^N\bar{\omega}^u$ implies that it is expressed in body u-frame relative to the inertial N-frame.

(2) $\hat{u}_i$ is used to define the measure numbers

(3) The angular velocities are the same in both sets.

(b) Recall that $\hat{u}_k = \hat{n}_k$ at $t = 0$ and $\omega_1(0) = +1.0$ rad/s, $\omega_2(0) = +2.0$ rad/s $\omega_3(0) = +1.0$ rad/s, and $T = 40$ N-met. What are the initial values of Euler parameters? Why?
Numerically integrate the dynamic and kinematic differential equations simultaneously for at least 3 cycles of the motion.
During the simulation, check the constraint equation on ε_i's. What is this constant (call it K) and is it, in fact, constant? (Plot $K - K_o$ over the simulation where K_o the value at the initial time.) How does this plot compare with the integration tolerance that you used in your simulation?

@ initial condition $\hat{u}_k = \hat{n}_k$

thus, the DCM will be a identity matrix

$$ {}^{N}C^{U}\Big|_{t=0} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} $$

thus,

$$ \varepsilon_4\Big|_{t=0} = \frac{1}{2}\sqrt{1 + C_{11} + C_{22} + C_{33}} = 1 $$

$$ \varepsilon_1\Big|_{t=0} = \frac{C_{32} - C_{23}}{4\varepsilon_4} = 0 $$

$$ \varepsilon_2\Big|_{t=0} = \frac{C_{13} - C_{31}}{4\varepsilon_4} = 0 $$

$$ \varepsilon_3\Big|_{t=0} = \frac{C_{21} - C_{12}}{4\varepsilon_4} = 0 $$

$$ \boxed{t=0 \Rightarrow {}^{N}\bar{\varepsilon}^{U} = [0, 0, 0],\ {}^{N}\varepsilon_4^{U} = 1} $$

Numerical integration (Matlab code is in appendix)

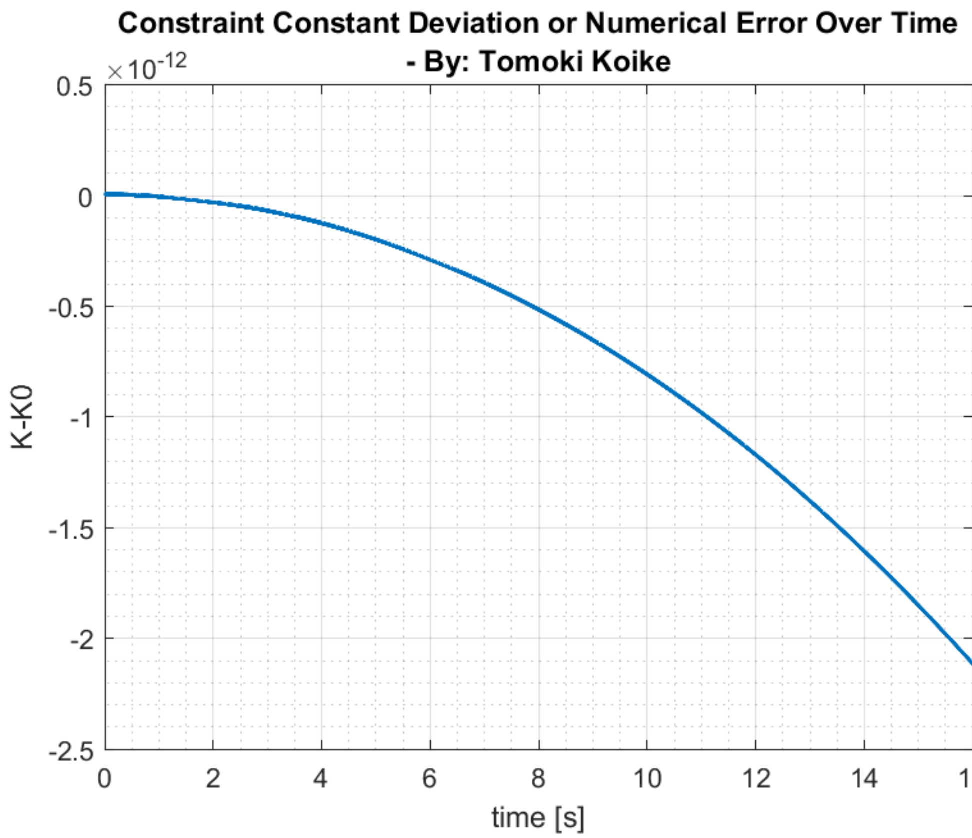
→ Constraints

$$ \varepsilon_1^2 + \varepsilon_2^2 + \varepsilon_3^2 + \varepsilon_4^2 = 1 $$

$$\Longleftrightarrow \varepsilon_1^2 + \varepsilon_2^2 + \varepsilon_3^2 + \varepsilon_4^2 = K$$

$$K\big|_{t=0} = K_0 = \varepsilon_1^2 + \varepsilon_2^2 + \varepsilon_3^2 + \varepsilon_4^2 \Big|_{t=0}$$

$$K_0 = 0 + 0 + 0 + 1 = 1$$



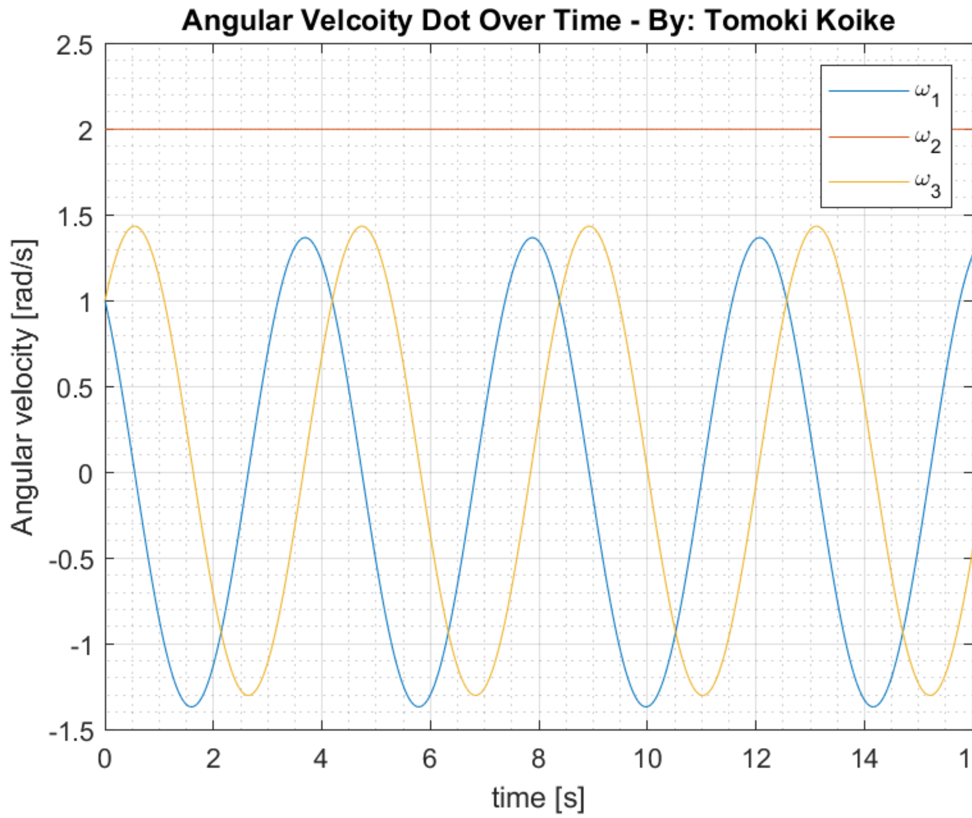
Discussion

(1) $k - k_0$ is decreasing over time which means the constant k is decreasing and deviating from 1. The rate of decrease is also increasing through time.

(2) The error shows the same order of magnitude with the tolerance of 1e13.

(c) Plot all three angular velocity measure numbers on the same plot. One should be constant…is it? To what level of accuracy? Compare its variations with the integration tolerance.

using MATLAB we can plot the Angular velocities as the following.



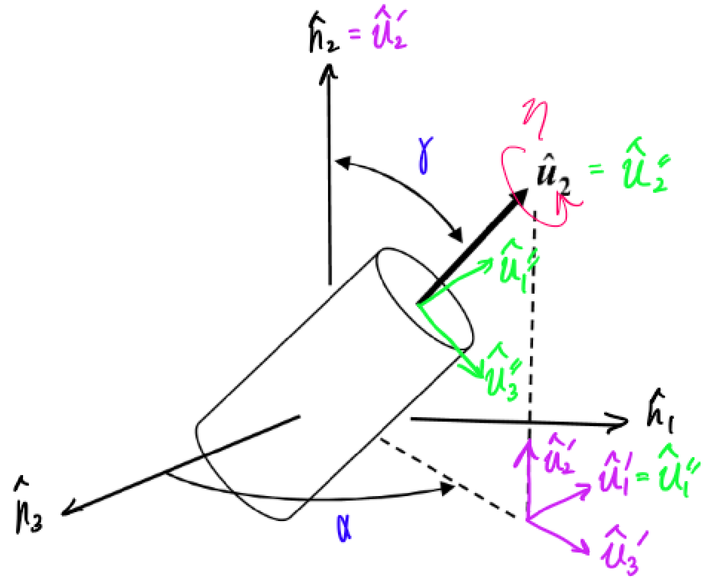
Discussion

(1) ω_2 is a constant value of 2.

(2) ω_1 and ω_2 is a sinusoidal plot with phase changes

(3) for ω_2, since $\dot{\omega}_2 = 0$ we know that ω_2 has no change from the initial condition of $\omega_2 = 0$. Thus, we can observe high accuracy which almost does not showcase the error from integration tolerance.

(d) The desired output information includes precession (α) and nutation (γ) angles. On the sketch below, identify the appropriate angles that reflects body-two 2-1-2 angles and include spin (η) and add the unit vectors that are associated with the intermediate frames to clarify the angle sequence.



Body-two: 2-1-2

	$\mathbf{b}_1$	$\mathbf{b}_2$	$\mathbf{b}_3$
$\mathbf{a}_1$	$-s_1c_2s_3 + c_3c_1$	s_1s_2	$s_1c_2c_3 + s_3c_1$
$\mathbf{a}_2$	s_2s_3	c_2	$-s_2c_3$
$\mathbf{a}_3$	$-c_1c_2s_3 - c_3s_1$	c_1s_2	$c_1c_2c_3 - s_3s_1$

$$
\begin{cases}
\theta_1 = \alpha \\
\theta_2 = \gamma \\
\theta_3 = \eta
\end{cases}
$$

(e) Produce a list of the output at 2 sec intervals. Include the following quantities: t, ω_i, C_{12}, C_{22}, C_{32}, α, γ K. Note that all angles are listed and plotted in <u>degrees</u>. Explain how you determine the correct quadrant for the angles.

nutation, γ can be computed (from the body 2-1-2 table)

$$\cos\theta_2 = \cos\gamma = C_{22}$$

$$\gamma = \arccos(C_{22}) \rightarrow 0 \leq \gamma \leq 180^\circ$$

and, precession, α can be expressed as

$$\cos\theta_1 \sin\theta_2 = C_{32}$$

$$\cos\theta_1 = \frac{C_{32}}{\sin\theta_2} = \frac{C_{32}}{\sin\gamma}$$

$$\cos\alpha = \frac{C_{32}}{\sin\gamma}$$

$$\alpha_1 = \arccos\left(\frac{C_{32}}{\sin\gamma}\right)$$

also

$$\sin\theta_1 \sin\theta_2 = C_{12}$$

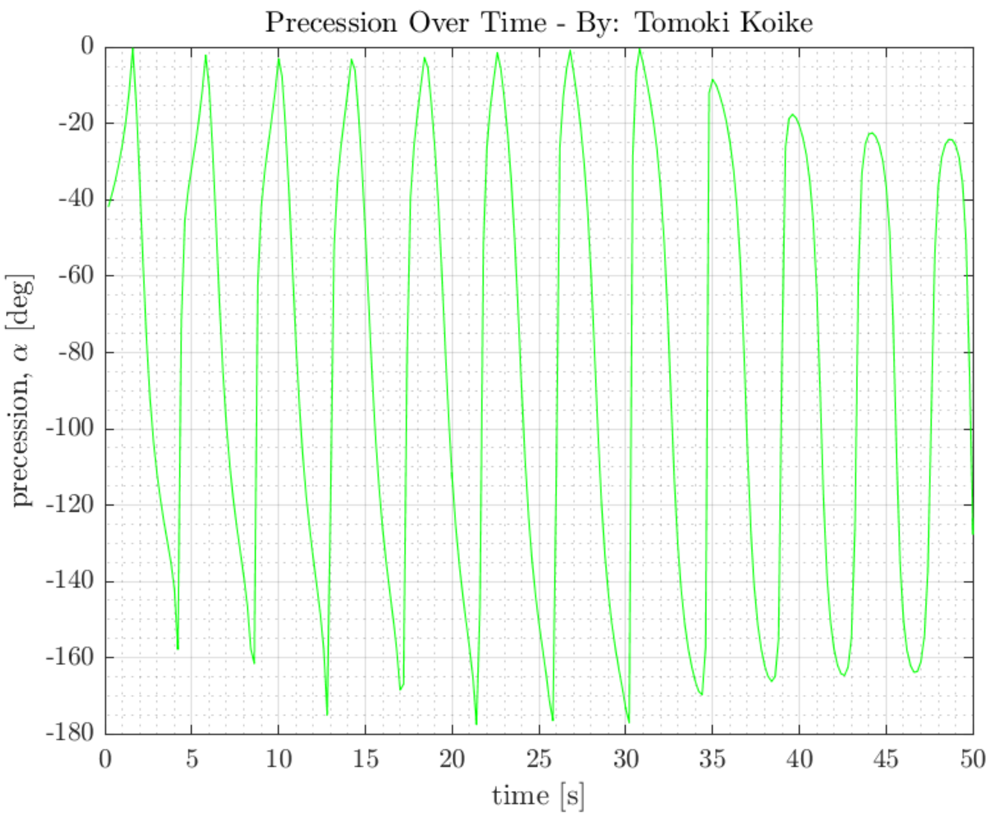
$$\alpha_2 = \arcsin\left(\frac{C_{12}}{\sin\gamma}\right)$$

chose the proper alpha from α_1 & α_2

(Matlab code in appendix)

Problem 2: Assuming that you have successfully completed Prob 1, plot and discuss some of the results:

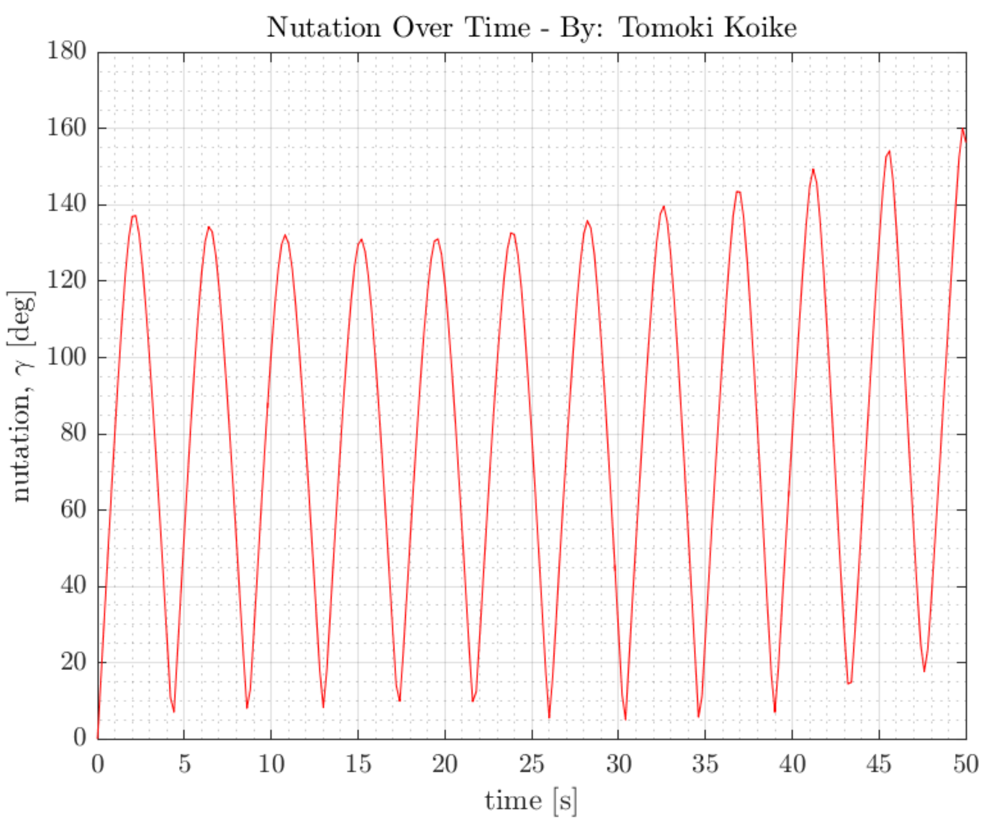
(a) Plot the precession and nutation angles separately as functions of time. Initially, $\gamma = 0^{\circ}$. How does it change over time? Does it ever return to zero? Why or why not?



precession, α

The precession illustrates a triangular wave ; but, with varying peaks and all values being a negative value. The highest peaks seem to approach 0 but does not become equivalent.

Because of the moment T the orientation is changed for each cycle (you can see this on $C_{32}-C_{12}$ plot), and therefore will not return to 0.



Nutation, γ

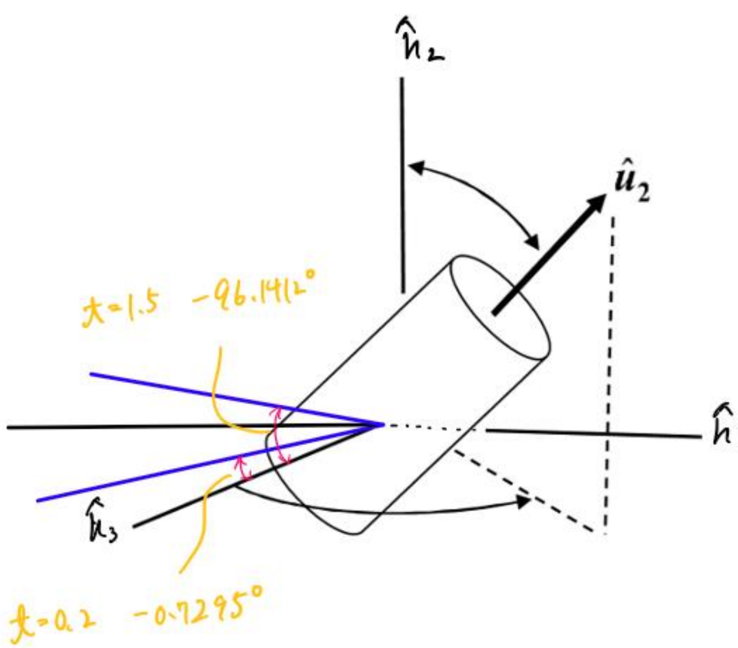
The nutation plot is also a triangle wave but the wave amplitude recedes as time goes by. The plot shows that nutation is converging to a certain value which is above 0 and does return to 0.

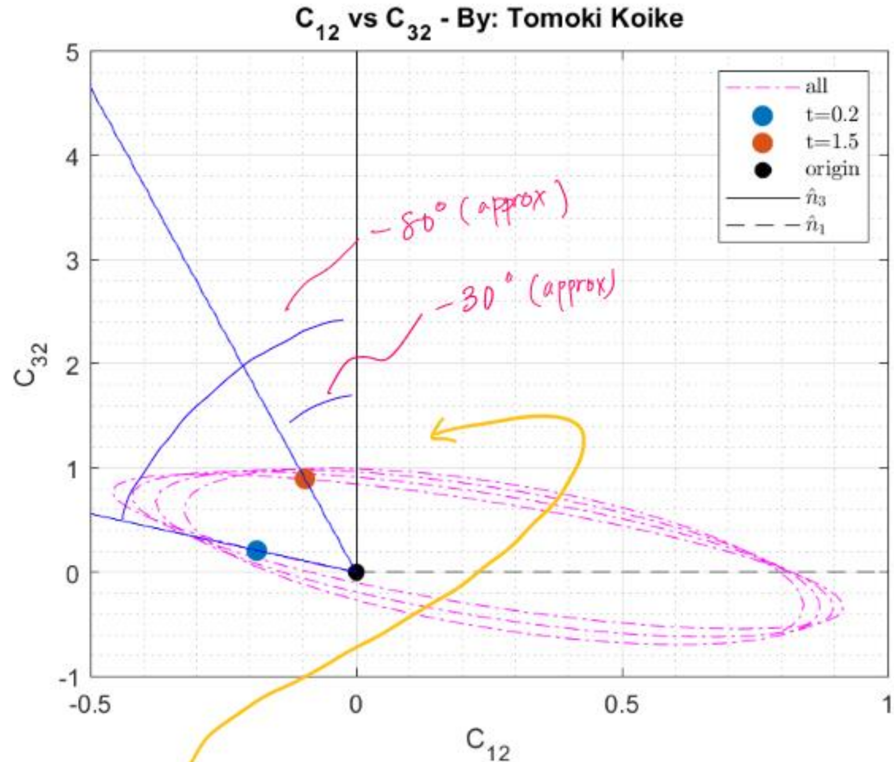
The nutation angle does not go back to 0 with the same reason as the precession angle, that is the torque changes the orientation for each cycle.

(b) Plot C_{32} as a function of C_{12}; this results in a view down the $\hat{n}_2$ axis. So, add unit vectors $\hat{n}_1$ and $\hat{n}_3$ to the sketch. On the plot, mark the time $t = 0.2$ sec and 1.5 sec. At these times, sketch the precession angle. If you measure the angle, does it match the value for precession angle that you computed corresponding to this time?
From the sketch in Prob 1, you should also be able to add $\hat{u}_1'$ and $\hat{u}_3'$ to the $C_{12} - C_{32}$ plot at $t = 0.2$ sec and 1.5 sec. (Maybe plot twice for clarity?) How are these unit vectors related to the precession angle? At this time, evaluate the quantity $h^2 = C_{12}^2 + C_{32}^2$; what does it tell you?
How is the value h related to the nutation angle at this time?

Compute γ from the value of h. Do the values match the nutation angles that you computed from the Euler paramters? Should they? What does it mena if $\gamma < 90^{\circ}$, $\gamma > 90^{\circ}$? Should γ ever be negative? Why or why not?

redo the ode45 computations with tspan of 0:0.1:16
increments of 0.1 and identify the points for $t=0.2$ and $t=1.5$





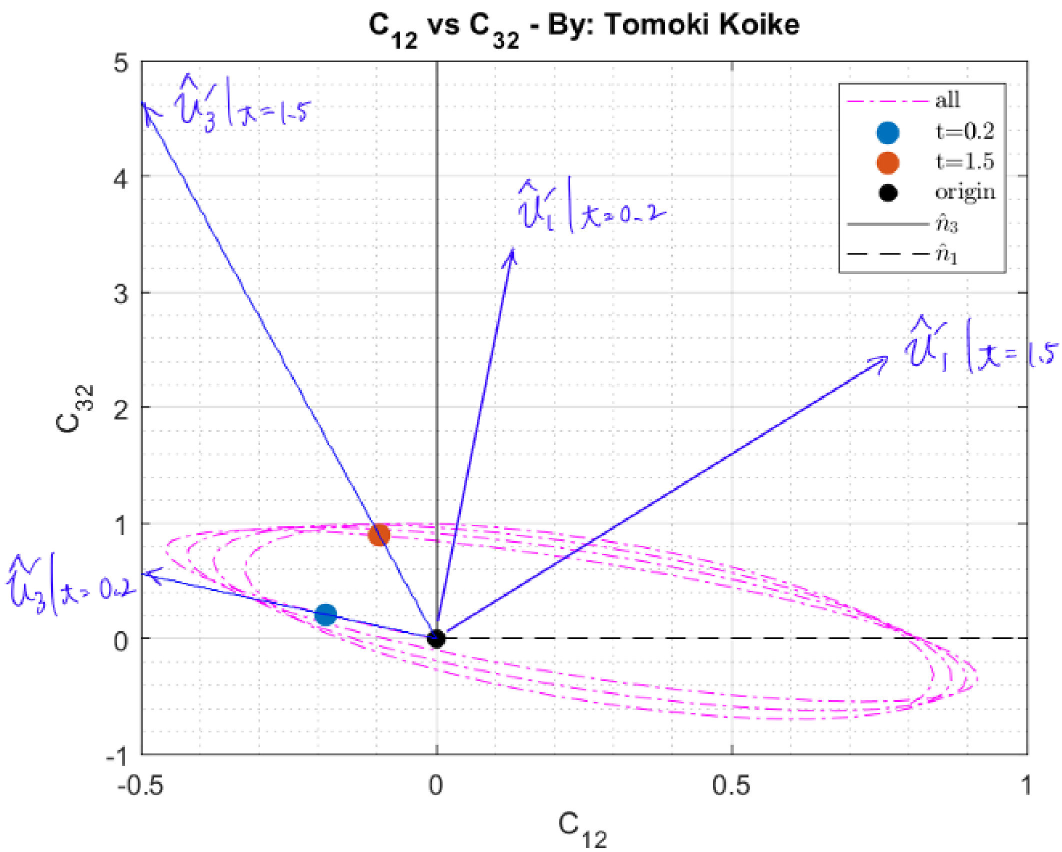
physically measuring the angles
@ $t = 0.2$
@ $t = 1.5$ (computed with MATLAB)

The actual angles (precession) are

@ $t = 0.2$ $\quad \alpha|_{t=0.2} = -41.7981°$

@ $t = 1.5$ $\quad \alpha|_{t=1.5} = -6.1412°$ (from MATLAB)

Due to the plots scaling the angle measured roughly is different from the actual angles, but they have relatively close values



Using MATLAB we compute

$$\boxed{\begin{aligned} h^2\big|_{x=0.2} &= C_{12}^2\big|_{x=0.2} + C_{32}^2\big|_{x=0.2} = 0.0765 \\ h^2\big|_{x=1.5} &= C_{12}^2\big|_{x=1.5} + C_{32}^2\big|_{x=1.5} = 0.0186 \end{aligned}}$$

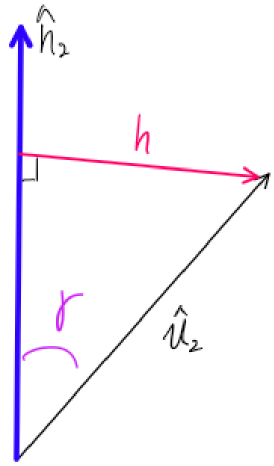
since

${}^N C^u$	$\hat{u}_1$	$\hat{u}_2$	$\hat{u}_3$
$\hat{n}_1$	$\hat{n}_1 \cdot \hat{u}_1$	$\hat{n}_1 \cdot \hat{u}_2$	$\hat{n}_1 \cdot \hat{u}_3$
$\hat{n}_2$	$\hat{n}_2 \cdot \hat{u}_1$	$\hat{n}_2 \cdot \hat{u}_2$	$\hat{n}_2 \cdot \hat{u}_3$
$\hat{n}_3$	$\hat{n}_3 \cdot \hat{u}_1$	$\hat{n}_3 \cdot \hat{u}_2$	$\hat{n}_3 \cdot \hat{u}_3$

${}^N C^u_{21}$ is a projection of $\hat{u}_2$ onto $\hat{n}_1$ and

${}^N C^u_{32}$ is a projection of $\hat{u}_2$ onto $\hat{n}_3$

and thus, $h = \sqrt{C_{12}^2 + C_{32}^2}$ is length from $\hat{h}_2$ axis to $\hat{u}_2$ which is also perpendicular to $\hat{h}_2$



thus, $\sin\gamma = \dfrac{h}{\|\hat{u}_2\|}$

$$\Longleftrightarrow \sin\gamma = h \qquad \because \|\hat{u}_2\| = 1$$

Using MATLAB we calculate

@ $t = 0.2$ $\boxed{\gamma\big|^{A}_{t=0.2} = \arcsin\left(h\big|_{t=0.2}\right) = 0.2840}$

and using

$\boxed{\gamma\big|^{B}_{t=0.2} = \arccos\left(C_{22}\big|_{t=0.2}\right) = 0.2840}$

@ $t = 1.5$ $\boxed{\gamma\big|^{A}_{t=1.5} = \arcsin\left(h\big|_{t=1.5}\right) = 1.1204}$

and

$\boxed{\gamma\big|^{B}_{t=1.5} = \arccos\left(C_{22}\big|_{t=1.5}\right) = 2.0212}$

Analysis

here we can see the relation

$$\pi - 1.1204 = 2.0212$$

$$\Longleftrightarrow \quad \pi - \gamma\Big|^{A}_{t=1.5} = \gamma\Big|^{B}_{t=1.5}$$

since $C_{22}\big|_{t=1.5} < 0$

Thus nutation $\gamma = \arcsin(h)$ if $C_{22} > 0$

$\gamma = \pi - \arcsin(h)$ if $C_{22} < 0$

- This conclusion was deduced by creating and examining the following table

(from MATLAB)

$\pi - \gamma|^{A}$

table_temp_top40

	C22	gammaD	gammaA	piMinusGammaA
1	1.0000	0	0	3.1416
2	0.9900	0.1417	0.1417	2.9999
3	0.9600	0.2840	0.2840	2.8576
4	0.9104	0.4264	0.4264	2.7152
5	0.8425	0.5689	0.5689	2.5727
6	0.7577	0.7110	0.7110	2.4306
7	0.6581	0.8525	0.8525	2.2891
8	0.5461	0.9931	0.9931	2.1485
9	0.4248	1.1323	1.1323	2.0093
10	0.2985	1.2697	1.2697	1.8719
11	0.1651	1.4049	1.4049	1.7367
12	0.0336	1.5372	1.5372	1.6044
13	-0.0951	1.6660	1.4756	1.6660
14	-0.2178	1.7904	1.3512	1.7904
15	-0.3320	1.9092	1.2323	1.9092
16	-0.4353	2.0212	1.1204	2.0212
17	-0.5257	2.1244	1.0172	2.1244
18	-0.6016	2.2163	0.9253	2.2163
19	-0.6618	2.2940	0.8476	2.2940
20	-0.7054	2.3538	0.7878	2.3538
21	-0.7320	2.3920	0.7496	2.3920
22	-0.7414	2.4060	0.7356	2.4060
23	-0.7339	2.3949	0.7467	2.3949
24	-0.7099	2.3602	0.7814	2.3602
25	-0.6701	2.3052	0.8364	2.3052
26	-0.6154	2.2337	0.9079	2.2337
27	-0.5469	2.1495	0.9921	2.1495
28	-0.4658	2.0554	1.0862	2.0554
29	-0.3738	1.9537	1.1879	1.9537
30	-0.2718	1.8461	1.2955	1.8461
31	-0.1622	1.7337	1.4079	1.7337
32	-0.0466	1.6174	1.5242	1.6174
33	0.0730	1.4977	1.4977	1.6439
34	0.1944	1.3751	1.3751	1.7665
35	0.3154	1.2500	1.2500	1.8916
36	0.4336	1.1225	1.1225	2.0191
37	0.5463	0.9929	0.9929	2.1487
38	0.6514	0.8614	0.8614	2.2802
39	0.7464	0.7282	0.7282	2.4134
40	0.8289	0.5936	0.5936	2.5479

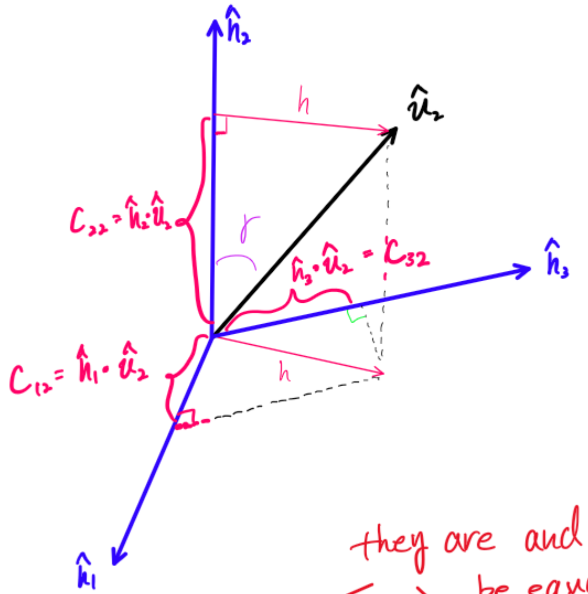
positive C_{22}

negative C_{22}

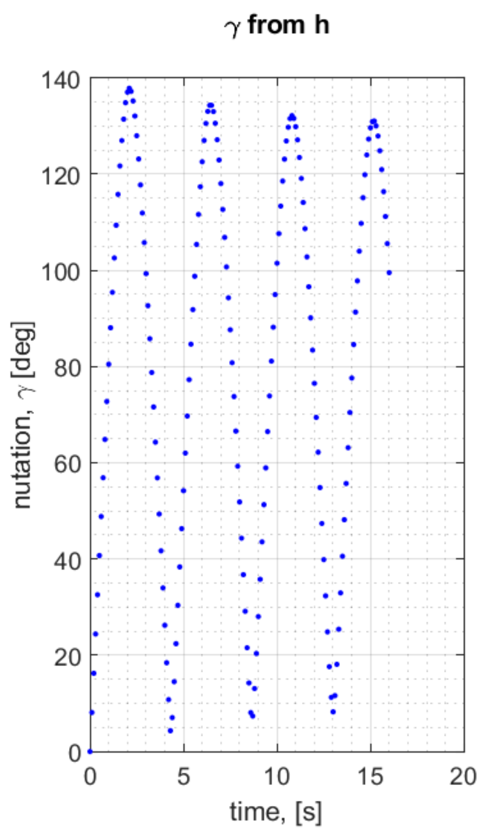
This table also implies that nutation angle, γ computed from Euler parameters equal $\arcsin(h)$ if $C_{22} > 0$ but if $C_{22} < 0$ $\gamma = \pi - \arcsin(h)$

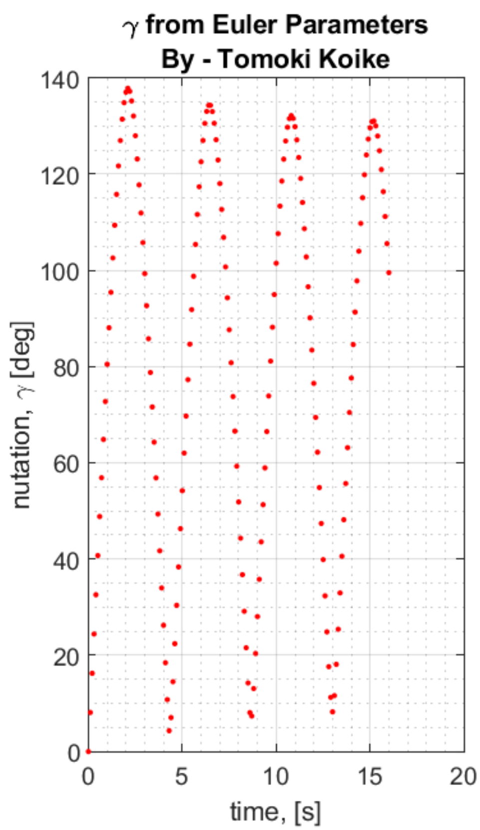
$$\boxed{\text{nutation} = \gamma = \begin{cases} \arcsin(h) & \text{if } C_{22} > 0 \\ \pi - \arcsin(h) & \text{if } C_{22} < 0 \end{cases}}$$

again from the angular relation below this should be true.



they are and should be equal





we have computed the gammas with the function we have defined above and by plotting it juxtaposed to the gamma results computed from the Euler Parameters we can vindicate our formula to be correct.

From these relations we can also tell that, since C_{22} is a projection of $\hat{u}_2$ onto $\hat{n}_2$

$$\gamma > 90^\circ \iff C_{22} < 0$$
$$\iff \gamma = \pi - \arcsin(h)$$

$$\gamma < 90^\circ \iff C_{22} > 0$$
$$\iff \gamma = \arcsin(h)$$

also

we can scrutinize that

@ $h < 1$

axis of symmetry is above or below

and

$h_{max} = 1 \Rightarrow$ where axis of symmetry is 90° w.r.t $\hat{n}_2$

and since $h > 0$ from the formula

$$h = \sqrt{C_{12}^2 + C_{32}^2} \leq 1$$

and

$$\gamma = \begin{cases} h > 0 \\ \pi - h > 0 \end{cases}$$

so

γ can never be negative.

(c) Plot the precession rate as a function of time. What does it tell you? Is it ever negative? What deos that mean?

Body-two: 2-1-2

$\omega_1 = \dot\theta_1 s_2 s_3 + \dot\theta_2 c_3$	$\dot\theta_1 = (\omega_1 s_3 - \omega_3 c_3)/s_2$
$\omega_2 = \dot\theta_1 c_2 + \dot\theta_3$	$\dot\theta_2 = \omega_1 c_3 + \omega_3 s_3$
$\omega_3 = -\dot\theta_1 s_2 c_3 + \dot\theta_2 s_3$	$\dot\theta_3 = (-\omega_1 s_3 + \omega_3 c_3)c_2/s_2 + \omega_2$

the above is from the supplemental document

precession rate

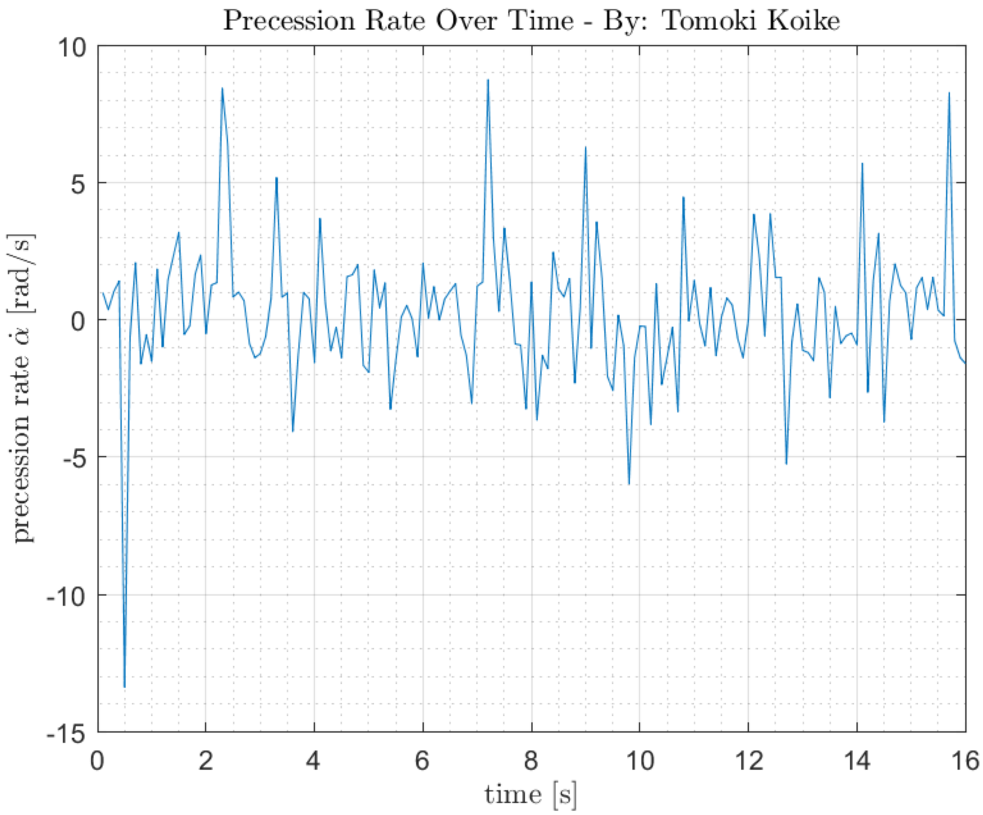
$$\dot\theta_1 = \dot\alpha_1 = \frac{\omega_1 s_3 - \omega_3 c_3}{s_2}$$

using ω_1, ω_2, ω_3 from the numerical integration.

also $\theta_3 = \eta$ and $\theta_2 = \gamma$

now calculate using the function above to find precession rate

plot this using MATLAB and we obtain the following



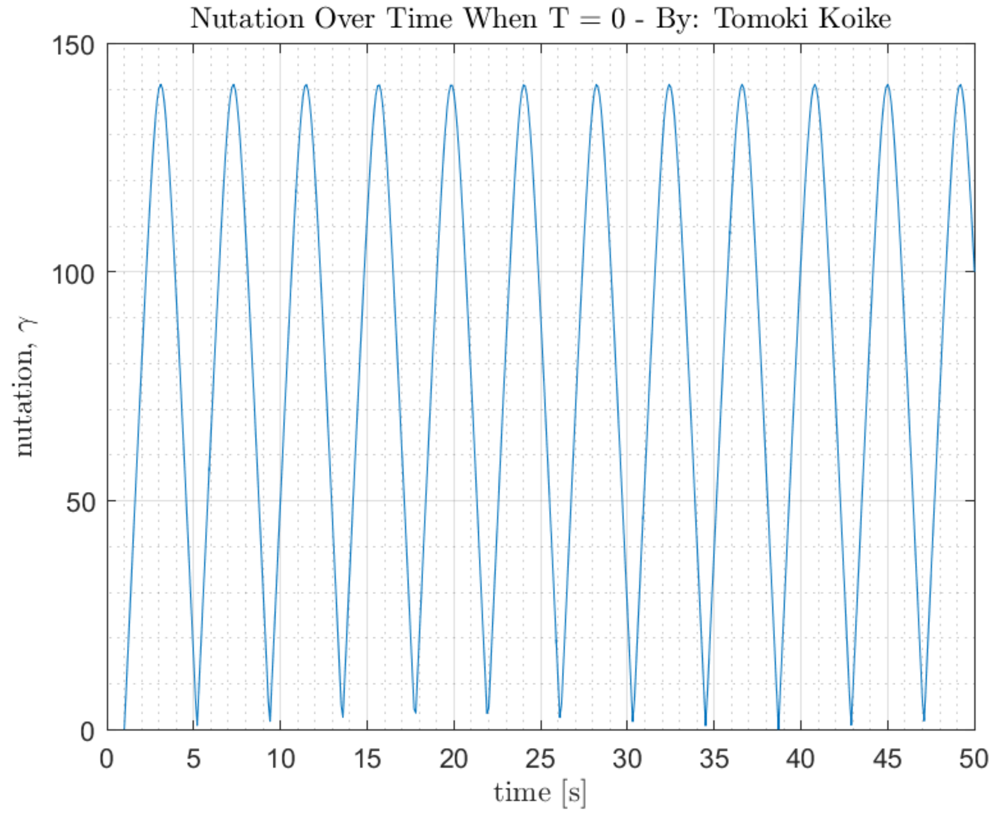
This shows the precession rate is fluctuating incessantly without any patterns.

It is sometimes negative.

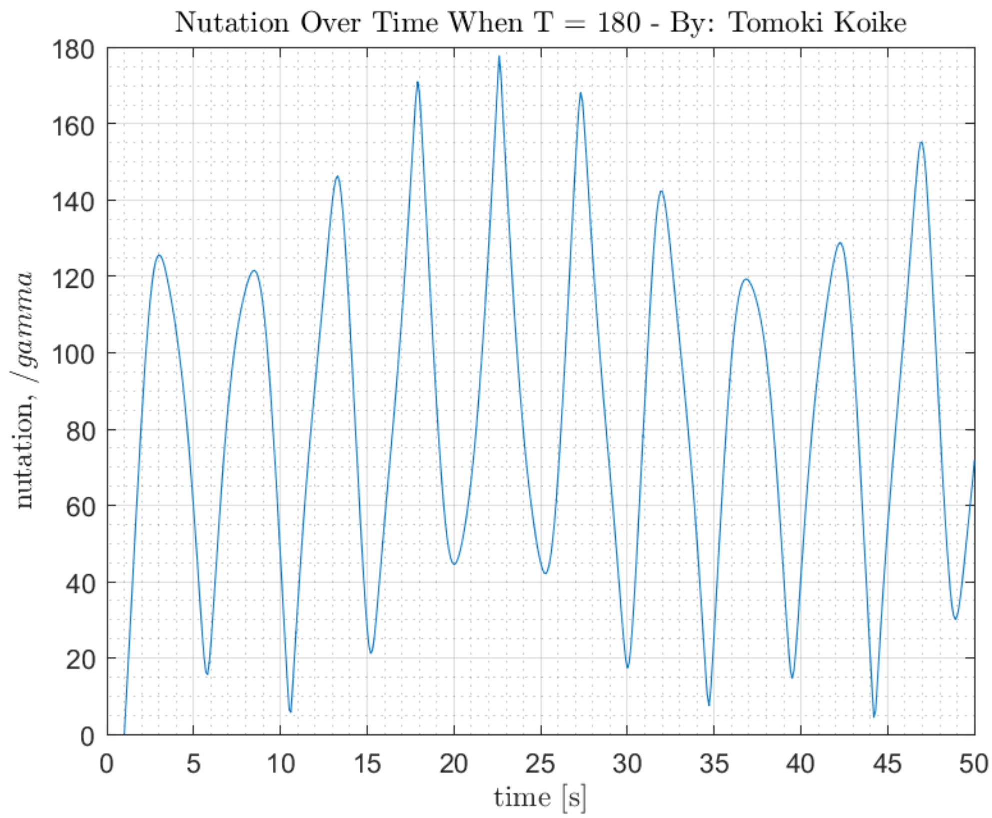
Because of the singularities existing in the process of computing the precession rate with euler parameters and DCM the points do not completely represent the motion of the body. But the logic of the precession decreasing agrees as to when the precession rate turns to negative. Whereby, the body effected by both precession and nutation will make it possible for the body to rotate in a way that the precession decreases.

(d) For the simulation, choose some additional values for the constant torque: $T = 0$, $T = 180$ N-met. Plot γ as a function of time for these additional sample torques. What is happening to the motion of the body? How does the torque affect the behavior? What happens to the $C_{12} - C_{32}$

Do you think this constant torque is 'stabilizing' or 'destabilizing'?



Nutation Over Time When T = 0 - By: Tomoki Koike
150
100
50
0
nutation, γ
0
5
10
15
20
25
30
35
40
45
50
time [s]



Nutation Over Time When T = 180 - By: Tomoki Koike
180
160
140
120
100
80
60
40
20
0
nutation, $/gamma$
0
5
10
15
20
25
30
35
40
45
50
time [s]

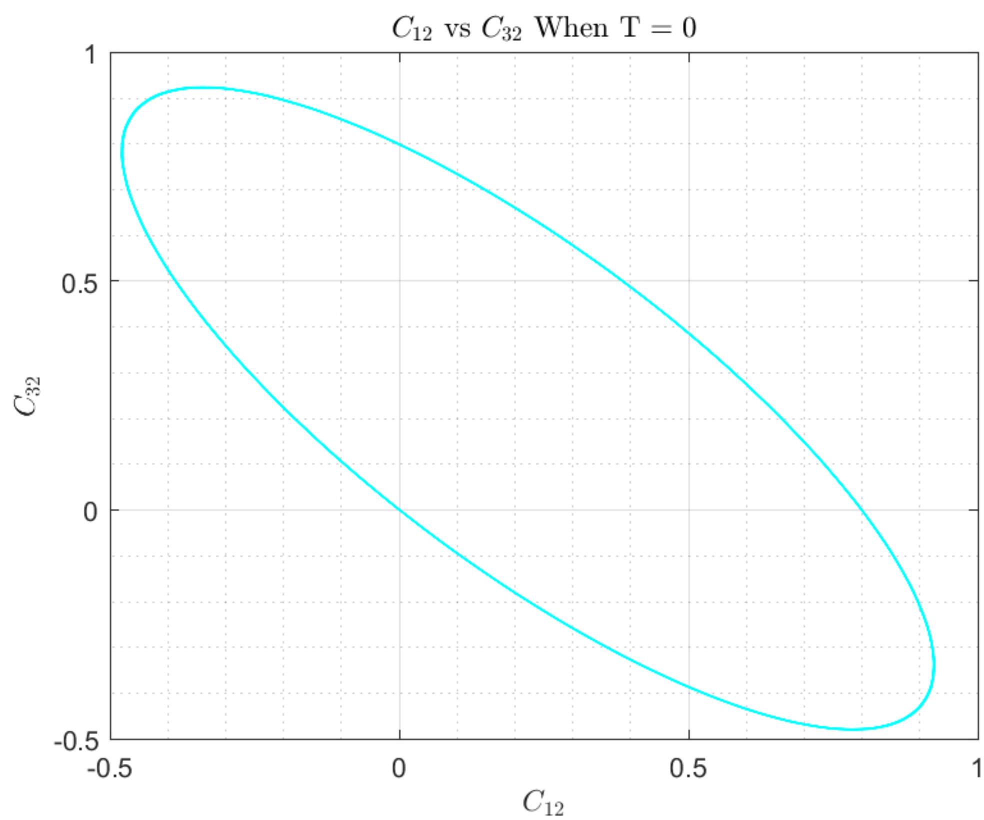
when $T = 0$

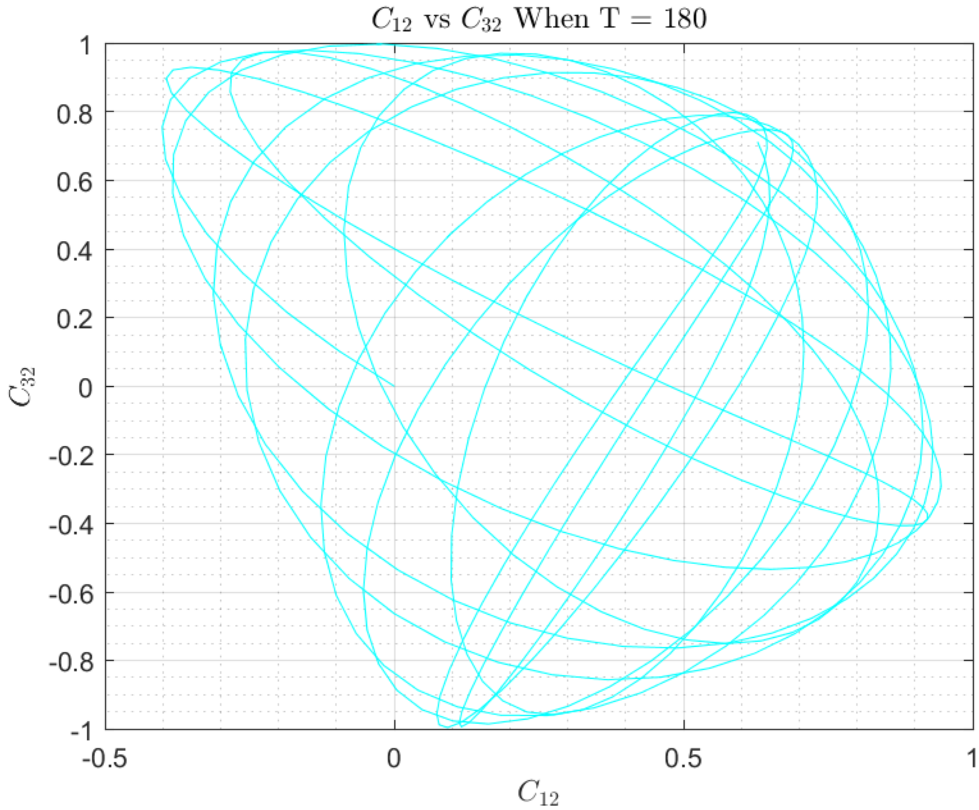
the nutation, γ transitions in a cyclic manner by rocking back and forth in angles $0°$ to approximately $140°$ w.r.t $\hat{n}_2$. That is, it is periodic.

when $T = 180$

the orientation is shifted constantly that the γ never goes to 0.

The torque adds disturbance to the motion of the body.





-

for T=0

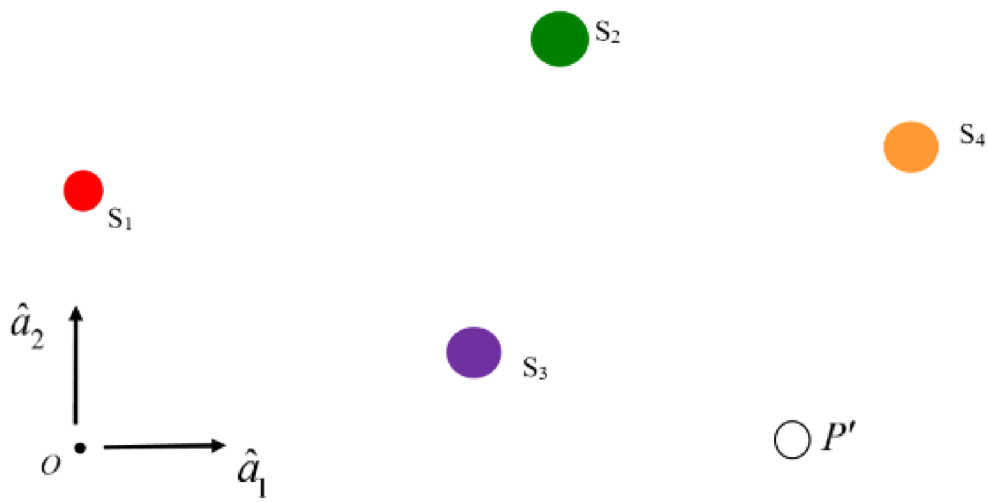
$C_{12}-C_{32}$ depicts a clean ellipse. That is, it is periodic.

for T=180

$C_{12}-C_{32}$ rotates in a radical manner without any pattern.

Thus, we can conclude that the torque is "destabilizing" the rotational motion.

Problem 3: In class, the discussion has concerned a more detailed understanding of the gravity force as we are developing an expression for the torque. So, consider the gravity force for a simple system comprised of a set of particles. Let P' (mass 5 m) be an attracting particle acting on the system S. The system S is comprised of the 4 colored particles:



Then, the system S possesses the following characteristics where the distance of each particle relative to P' is given:

S_1	mass = $3m$	$\bar{r}^{O\to S_1} = 3d\,\hat{a}_2$
S_2	mass = 4 m	$\bar{r}^{O\to S_2} = 4d\,\hat{a}_1 + 4d\,\hat{a}_2$
S_3	mass = 4 m	$\bar{r}^{O\to S_3} = 3d\,\hat{a}_1 + 1d\,\hat{a}_2$
S_4	mass = 3 m	$\bar{r}^{O\to S_4} = 7d\,\hat{a}_1 + 3d\,\hat{a}_2$
P'	mass = 5 m	$\bar{r}^{O\to P'} = +6d\,\hat{a}_1$

(a) Plot the system exactly to scale. Compute the location of the c.m.; add the center of mass to the figure.

(b) Compute the resultant gravity force; express it as a magnitude and unit vector in terms of $\hat{a}_1, \hat{a}_2$

action as it extends through the system S.

(c) Determine the distance R^{cg} between the attracting particle and the c.g. Add the c.g. to the figure. Do the c.m. and the c.g. coincide?

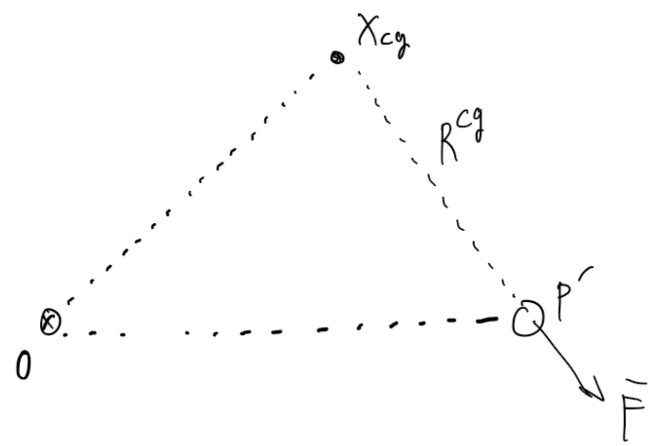
Brainstorming

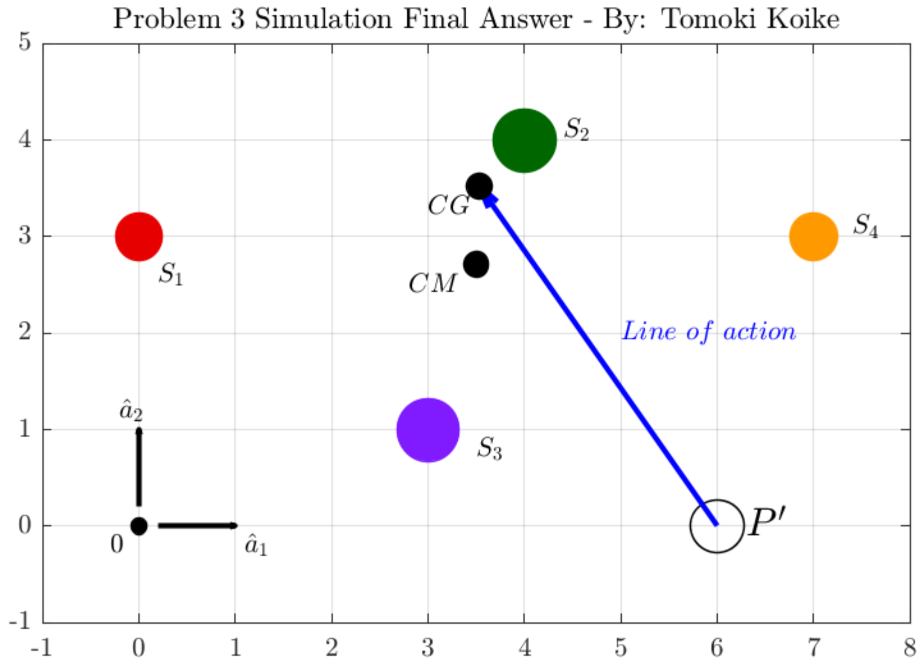
Finding $\overrightarrow{OX_{cg}}$

$$\overrightarrow{P'X_{cg}} = \frac{-\bar{F}}{|\bar{F}|} R^{cg}$$

$$\overrightarrow{OX_{cg}} = \overrightarrow{OP'} + \overrightarrow{P'X_{cg}}$$

$$= \overrightarrow{OP'} - \frac{\vec{F}}{|\tilde{F}|} R^{cg}$$





Problem 3 Simulation Final Answer - By: Tomoki Koike
S_1
S_2
S_3
S_4
CG
CM
Line of action
P'
$\hat{a}_1$
$\hat{a}_2$
0

(b)

From MATLAB $\qquad |\vec{F}| = G\dfrac{m^2}{r^2}$

$$\bar{F} = G\left(2.1684\, m^2 d^{-2}\,\hat{a}_1 - 3.0090\, m^2 d^{-2}\,\hat{a}_2\right)$$

$$|\bar{F}| = 3.7823\, G m^2 d^{-2}$$

$$\therefore\ \bar{F} = |\bar{F}|\left(0.5733\,\hat{a}_1 - 0.8193\,\hat{a}_2\right)$$

$$\boxed{\bar{F} = 3.7823\, G m^2 d^{-2}\left(0.5733\,\hat{a}_1 - 0.8193\,\hat{a}_2\right)}$$

where G = gravitational constant

$R^{cg} = 4.3020\, d$

The cm and cg do NOT match.

Appendix

AAE440 HW5 MATLAB CODE

Problem 1

<b>

```
fdir = 'C:\Users\Tomo\Desktop\studies\2020-Spring\AAE440\MATLAB\outputs\HW5';
set(groot, 'defaulttextinterpreter',"latex");
set(groot, 'defaultAxesTickLabelInterpreter',"latex");
set(groot, 'defaultLegendInterpreter',"latex");


% Constants
T = 40;  % Torque [N-m]
I_cm = [400 0 0; 0 100 0; 0 0 400];  % Inertia Dyadic [kg-m2]
I = 400;
J = 100;


% Initial Conditions
w0 = [1 2 1];  % Initial angular velocities [rad/s]
e0 = [0 0 0 1];  % Initial Euler Parameters
C0 = [1 0 0 0 1 0 0 0 1];  % Initial DCM


% Numerical integrations dynamic and kinematic EOMs
tspan = [0 16]; % Integration time
y0 = [w0 e0 0 C0]; % Initial conditions
option = odeset('RelTol', 1e-13, 'AbsTol', 1e-13); % Integration Tolerance
[t, res] = ode45(@(t,y) EOM(t,y,I,J,T), tspan, y0, option);
ws = res(:,1:3);  % angular velocity dot
es = res(:,4:7);  % Euler parameter dot
K_minus_K0 = res(:,8);  % K-K0
C_mats = res(:,9:end);  % DCM values


% Plotting K-K0 against time
fig1 = figure("Renderer","painters");
plot(t, K_minus_K0)
ylabel('$K-K0$', "Interpreter","latex")
xlabel('$time$ [s]', "Interpreter","latex")
title({'Constraint Constant Deviation or Numerical Error Over Time',['- By:' ...
    ' Tomoki Koike']},"Interpreter", "latex")
grid on
grid minor
box on
```

```
saveas(fig1, fullfile(fdir, 'constraint_constant.png'));
```

<c>

```
% Plotting the angular velocity
fig2 = figure("Renderer","painters");
plot(t, ws)
ylabel('Angular velocity [rad/s]')
xlabel('time [s]')
title({'Angular Velcoity Dot Over Time - By: Tomoki Koike'})
ylim([-1.5, 2.5])
legend('$\omega_1$', '$\omega_2$', '$\omega_3$')
grid on
grid minor
box on
saveas(fig2, fullfile(fdir, 'angular_velocity.png'))
```

<d>

```
% Define a new time span with a 2 second increment
tspan2 = 0:0.2:50;


% conduct ode45 for differential equation
[t2, res2] = ode45(@(t,y) EOM(t,y,I,J,T), tspan2, y0, option);


% Assign C12, C22, and C32
C12s = res2(:,10);
C22s = res2(:,13);
C32s = res2(:,16);
```

```


% Calculating the precession and nutation using loop
alphas = zeros([length(t2),1]);
gammas = zeros([length(t2),1]);
for n = 1:length(t2)
    % calculating and verfiying gamma
    gammas(n) = acos(C22s(n));
    if gammas(n) < 0 || gammas(n) > pi
        gammas(n) = -gammas(n);
    end
    % calculating and verfying the alpha
    alpha1 = asin(C12s(n)/sin(gammas(n)));
```

```
    alpha2 = acos(C32s(n)/sin(gammas(n)));
    if alpha1 == alpha2 || alpha1 == -alpha2
        alphas(n) = alpha1;
    elseif pi-alpha1 == alpha2 || -pi-alpha1 == alpha2
        alphas(n) = alpha2;
    else
        alphas(n) = -alpha2;
    end
    gammas(n) = rad2deg(gammas(n));
    alphas(n) = rad2deg(alphas(n));
end
% Define K
K = res2(:,8) + 1;
```

```
res_array = [t2 res2(:,1:3) C12s C22s C32s alphas gammas K];
res_table = array2table(res_array, "VariableNames",{'time','omega1',['' ...
    'omega2'],'omega3','C12','C22','C32','alpha', 'gamma','K'});
writetable(res_table, fullfile(fdir, 'output_table.xlsx'));
```

Problem 2

<a>

```
% Plotting the precession and nutation individually as a function of time
% Precession
fig3 = figure("Renderer","painters");
plot(t2, alphas,'g')
title({'Precession Over Time - By: Tomoki Koike'})
xlabel('time [s]')
ylabel('precession, $\alpha$ [deg]')
grid on
grid minor
box on
saveas(fig3, fullfile(fdir, 'precession_vs_time.png'));
% Nutation
fig4 = figure("Renderer","painters");
plot(t2, gammas,'r')
title({'Nutation Over Time - By: Tomoki Koike'})
xlabel('time [s]')
ylabel('nutation, $\gamma$ [deg]')
grid on
grid minor
box on
saveas(fig4, fullfile(fdir, 'nutation_vs_time.png'));
```

<b>

```
% Do the steps in the last half of problem 1 with smaller increments of
% time span
tspan3 = 0:0.1:16;
% conduct ode45 for differential equation
[t3, res3] = ode45(@(t,y) EOM(t,y,I,J,T), tspan3, y0, option);
C_new = res3(:,9:17);
% Assign C12, C22, and C32
C12s_new = res3(:,10);
C22s_new = res3(:,13);
C32s_new = res3(:,16);


% Finding the index when t=0.2 and t=1.5 and corresponding C12 and C32
idx_t0p2 = find(t3==0.2);
idx_t1p5 = find(t3==1.5);
C12_t0p2 = C12s_new(idx_t0p2);
C22_t0p2 = C22s_new(idx_t0p2);
C32_t0p2 = C32s_new(idx_t0p2);
C12_t1p5 = C12s_new(idx_t1p5);
C22_t1p5 = C22s_new(idx_t1p5);
C32_t1p5 = C32s_new(idx_t1p5);


% Assigning a temporary DCM with corresponding times
C_temp = res3([idx_t0p2 idx_t1p5], 9:end);
% calculating and verfiying gamma
[alphas_temp, gammas_temp, etas_temps] = ang_calc_body212(C_temp);
```

```
% Plots with the specific times t = 0.2 and 1.5
fig5 = figure("Renderer","painters");
plot(C12s_new, C32s_new,'-.m','MarkerSize',15)
title('$C_{12}$ vs $C_{32}$ - By: Tomoki Koike')
xlabel('$C_{12}$')
ylabel('$C_{32}$')
hold on
plot(C12_t0p2, C32_t0p2, '.','MarkerSize',26)
plot(C12_t1p5,C32_t1p5,'.','MarkerSize',26)
plot(0,0,'.k','MarkerSize',20)
plot([0 0],[0 5],'-k')
plot([0 1],[0 0],'--k')
d = linspace(0,-0.5,100);
plot(d,d.*(C32_t0p2/C12_t0p2),'-b')
plot(d,d.*(C32_t1p5/C12_t1p5),'-b')
hold off
legend('all','t=0.2','t=1.5','origin','$\hat{n}_3$','$\hat{n}_1$')
grid on
grid minor
```

```
box on
saveas(fig5, fullfile(fdir, 'C12_vs_C32.png'));
```

```
% Calcluating h
% @ t = 0.2
h_t0p2 = sqrt(C12_t0p2^2 + C32_t0p2^2);
gamma_est_t0p2 = asin(h_t0p2);
% @ t = 1.5
h_t1p5 = sqrt(C12_t1p5^2 + C32_t1p5^2);
gamma_est_t1p5 = asin(h_t1p5);


% Analysis
array_temp = [C22s_new, acos(C22s_new), asin(sqrt(C12s_new.^2+C32s_new.^2)),...
    pi-asin(sqrt(C12s_new.^2+C32s_new.^2))];
table_temp = array2table(array_temp, "VariableNames",{'C22', 'gammaB', 'gammaA',
'piMinusGammaA'});
table_temp_top40 = table_temp(1:40,:);
```

```
% Actually calculating gamma with h
gammas_h = calc_gamma_with_h(C12s_new, C22s_new, C32s_new);
gammas_h = rad2deg(gammas_h);
% The actual gammas from the Euler parameters
[alpha_eulers, gammas_eulers, etas_eulers] = ang_calc_body212(C_new);
fig6 = figure("Renderer","painters");
subplot(1,2,1)
plot(t3, gammas_h,'.b')
title({'$\gamma$ from h',''})
xlabel('time, [s]')
ylabel('nutation, $\gamma$ [deg]')
grid on
grid minor
box on
subplot(1,2,2)
plot(t3, gammas_eulers, '.r')
title({'$\gamma$ from Euler Parameters','By - Tomoki Koike'})
xlabel('time, [s]')
ylabel('nutation, $\gamma$ [deg]')
grid on
grid minor
box on
saveas(fig6, fullfile(fdir, 'gammas_with_h_and_euler.png'))
```

<c>

```
% Do the steps in the last half of problem 1 with smaller increments of
% time span
tspan_c = 0:0.1:16;
% conduct ode45 for differential equation
[t_c, res_c] = ode45(@(t,y) EOM(t,y,I,J,T), tspan_c, y0, option);
% Assign C12, C22, and C32
C_c = res_c(:,9:end);
w1 = res_c(:,1);
w2 = res_c(:,2);
w3 = res_c(:,3);
[alpha_c, gamma_c, eta_c] = ang_calc_body212(C_c);


% Calculating precession rate
alpha_dot = (w1.*sin(eta_c) - w3.*cos(eta_c)) ./ sin(gamma_c);


% Plotting
fig7 = figure('Renderer',"painters");
plot(t_c, alpha_dot,'-b')
title('Precession Rate Over Time - By: Tomoki Koike',"Interpreter","latex")
xlabel('time [s]',"Interpreter","latex")
ylabel('precession rate $\dot{\alpha}$ [rad/s]',"Interpreter","latex")
grid on
grid minor
box on
saveas(fig7, fullfile(fdir,'precession_rate.png'));
```

<d>

```
% Constants
tspan_d = 1:0.1:50;
T_d1 = 0;  % Torque [N-m]
[t_d1, res_d1] = ode45(@(t,y) EOM(t,y,I,J,T_d1), tspan_d, y0, option);
C_d1 = res_d1(:,9:end);
[alpha_d1, gamma_d1, eta_d1] = ang_calc_body212(C_d1);


T_d2 = 180;
[t_d2, res_d2] = ode45(@(t,y) EOM(t,y,I,J,T_d2), tspan_d, y0, option);
C_d2 = res_d2(:,9:end);
[alpha_d2, gamma_d2, eta_d2] = ang_calc_body212(C_d2);


% Plotting
fig8 = figure("Renderer","painters");
plot(t_d1, gamma_d1)
xlabel('time [s]', "Interpreter","latex")
ylabel('nutation, $\gamma$', "Interpreter","latex")
```

```
title('Nutation Over Time When T = 0 - By: Tomoki Koike',"Interpreter","latex")
grid on
grid minor
box on
saveas(fig8, fullfile(fdir,'gamma_t_equal_0.png'));


fig9 = figure("Renderer","painters");
plot(t_d2, gamma_d2)
xlabel('time [s]',"Interpreter","latex")
ylabel('nutation, $/gamma$',"Interpreter","latex")
title('Nutation Over Time When T = 180 - By: Tomoki Koike',"Interpreter","latex")
grid on
grid minor
box on
saveas(fig9, fullfile(fdir,'gamma_t_equal_180.png'));


fig10 = figure("Renderer","painters");
plot(C_d1(:,2), C_d1(:,8), '-c')
title('$C_{12}$ vs $C_{32}$ When T = 0',"Interpreter","latex")
xlabel('$C_{12}$',"Interpreter","latex")
ylabel('$C_{32}$',"Interpreter","latex")
grid on
grid minor
box on
saveas(fig10, fullfile(fdir,'C12_C32_T0.png'))


fig11 = figure('Renderer',"painters");
plot(C_d2(:,2), C_d2(:,8), '-c')
title('$C_{12}$ vs $C_{32}$ When T = 180',"Interpreter","latex")
xlabel('$C_{12}$',"Interpreter","latex")
ylabel('$C_{32}$',"Interpreter","latex")
grid on
grid minor
box on
saveas(fig11, fullfile(fdir,'C12_C32_T180.png'))
```

Functions

```
function [alphas, gammas, etas] = ang_calc_body212(DCM)
    % DCM is 1 by 9 matrix with each column being C_ij
    C12s = DCM(:,2);
    C21s = DCM(:,4);
    C22s = DCM(:,5);
    C23s = DCM(:,6);
    C32s = DCM(:,8);
```

```
    alphas = zeros([length(C12s),1]);
    gammas = zeros([length(C12s),1]);
    etas = zeros([length(C12s),1]);

    for n = 1:length(alphas)
        % calculating and verfiying gamma
        gammas(n) = acos(C22s(n));
        if gammas(n) < 0 || gammas(n) > pi
            gammas(n) = -gammas(n);
        end
        % calculating and verfying the alpha
        alpha1 = asin(C12s(n)/sin(gammas(n)));
        alpha2 = acos(C32s(n)/sin(gammas(n)));
        if alpha1 == alpha2 || alpha1 == -alpha2
            alphas(n) = alpha1;
        elseif pi-alpha1 == alpha2 || -pi-alpha1 == alpha2
            alphas(n) = alpha2;
        else
            alphas(n) = -alpha2;
        end
        eta1 = asin(C21s(n)/sin(gammas(n)));
        eta2 = acos(-C23s(n)/sin(gammas(n)));
        if eta1 == eta2 || eta1 == -eta2
            etas(n) = eta1;
        elseif pi-eta1 == eta2 || -pi-eta1 == eta2
            etas(n) = eta2;
        else
            etas(n) = -eta2;
        end
        gammas(n) = rad2deg(gammas(n));
        alphas(n) = rad2deg(alphas(n));
        etas(n) = rad2deg(etas(n));
    end
end


function ang = eval_cos(theta)
    if theta < 0 || theta > pi
        ang = -theta;
    else
        ang = theta;
    end
end


function ang = eval_sin(theta)
    if -pi/2 <= theta && theta <= pi/2
        ang = theta;
    else
        ang = -theta;
```

```
    end
end


function gamma = calc_gamma_with_h(C12, C22, C32)
    h = asin(sqrt(C12.^2 + C32.^2));
    gamma = zeros([length(h), 1]);
    for x = 1:length(h)
        if C22(x) > 0
            gamma(x) = h(x);
        elseif C22(x) < 0
            gamma(x) = pi - h(x);
        end
    end
end
```

Problem 3

```
clear all; close all; clc;


fdir = 'C:\Users\Tomo\Desktop\studies\2020-Spring\AAE440\MATLAB\outputs\HW5';
set(groot, 'defaulttextinterpreter',"latex");

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

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


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


%% (a)
% Plotting the system

% Postion vectors for each S(i) and P

d = 1;

origin = [0; 0];

S1 = [d*0 d*3];

S2 = [d*4 d*4];

S3 = [d*3 d*1];

S4 = [d*7 d*3];
```

```
P = [d*6 d*0];

S_x = [S1(1); S2(1); S3(1); S4(1)];  % all x positions

S_y = [S1(2); S2(2); S3(2); S4(2)];  % all x positions


name_str = ["$0$","$\hat{a}_1$","$\hat{a}_2$","$S_1$","$S_2$",...
    "$S_3$","$S_4$", "$P^\prime$"];


fig1 = figure("Renderer","painters");
hold on; grid on; box on; axis equal;

ylim([-1, 5]); xlim([-1, 8]);

plot(origin(1),origin(2), '.','MarkerSize', 20,'Color',[0 0 0])

plot(S1(1),S1(2), '.','MarkerSize', 60,'Color',[0.9   0 0]);

plot(S2(1),S2(2), '.','MarkerSize', 80,'Color',[  0 0.4 0]);

plot(S3(1),S3(2), '.','MarkerSize', 80,'Color',[0.5 0.1 1]);

plot(S4(1),S4(2), '.','MarkerSize', 60,'Color',[  1 0.6 0]);

plot(P(1),P(2), 'ko','MarkerSize',20);


text(-0.3, -0.2, name_str(1),"Interpreter","latex")
text(0.2, 2.6, name_str(4),"Interpreter","latex")

text(4.4, 4.1, name_str(5),"Interpreter","latex")

text(3.5, 0.8, name_str(6),"Interpreter","latex")

text(7.4, 3.1, name_str(7),"Interpreter","latex")

text(6.3, 0, name_str(8), "FontSize",15, "Interpreter","latex")
```

```
% a1_hat axis
x1 = [0.2 1];

y1 = [0 0];

drawArrow(x1,y1,'k', 'linewidth',2); text(1.1,-0.2,
name_str(2),"Interpreter","latex");

% a2_hat axis

x2 = [0 0];

y2 = [0.2 1];
```

```
drawArrow(x2,y2,'k', 'linewidth',2); text(-0.2,1.2,
name_str(3),"Interpreter","latex");
```

```
% Masses for each S(i) and P
m1 = 3;

m2 = 4;

m3 = 4;

m4 = 3;

mP = 5;

m_S = [m1 m2 m3 m4];

m_tot = sum(m_S);


% Computing the CM of the system
x_cm = dot(S_x,m_S)/m_tot;

y_cm = dot(S_y,m_S)/m_tot;


% Plotting the CM
plot(x_cm, y_cm, '.k','MarkerSize', 32); text(2.8, 2.5,
'$CM$',"Interpreter","latex");
```

```
% Re-defining positions of S(i) in terms of P' (attracting body)
S1_P = S1 - P;

S2_P = S2 - P;

S3_P = S3 - P;

S4_P = S4 - P;

S_P_all = [S1_P; S2_P; S3_P; S4_P];


% Computing the Line of action
% **Non-dimensionalized so disregard gravitational constant G

N = length(S_P_all(:,1));

dim = length(S_P_all(1,:));

F_i = zeros([N, dim]);

for i = 1:N
```

```
    F_i(i,:) = -mP.*m_S(i).*S_P_all(i,:).*norm(S_P_all(i,:)).^-3;
end
F = sum(F_i);
F_mag = norm(F);
F_unit = F/F_mag;
```

```
% Computing the CG
% **Non-dimensionalized so disregard gravitational constant G
R_cg = sqrt(mP*(m_tot)/norm(F));
```

```
% R_cg vector is essentially G'-P' (from CG to P') vector
% To make this into a postion vector wrt the origin we do the following vector
manipulation
P_Xcg = -F*R_cg/norm(F);
O_Xcg = P + P_Xcg;
```

```
% Plotting the Line of action
x3 = [6 O_Xcg(1)];
y3 = [0 O_Xcg(2)];
drawArrow(x3,y3, 'linewidth',2,'Color',[0 0 1]);
text(5, 2, '$Line$ $of$ $action$','Color','b',"Interpreter","latex");
```

```
% Plotting the CG
plot(O_Xcg(1), O_Xcg(2), 'k.','MarkerSize', 32);
text(3, 3.3, '$CG$', "Interpreter","latex");
title('Problem 3 Simulation Final Answer - By: Tomoki Koike')
saveas(fig1, fullfile(fdir,'hw5_p3_gravity_system.png'));
```

```
function dwdt = EOM(t,y,I,J,T)
    %{
      inputs:  1) t: time lapse
               2) y: angular velocities, euler parameters, initial
                     euler constraint constant, DCM
               3) I: moment of inertia about the non-rotating axis
               4) J: moment of inertia about the rotating axis
               5) T: torque
      outputs: 1) dwdt: differential y
    %}
    dwdt = zeros(17,1);
    % Dynamics EOMs
    dwdt(1) = T/I - (I-J)/I*y(3)*y(2);
    dwdt(2) = 0;
    dwdt(3) = (I-J)/I*y(1)*y(2);
    % Kinematic EOM of angular velocities and Euler parameters
    dedt1 =  0.5*( y(1)*y(7)-y(2)*y(6)+y(3)*y(5));
    dedt2 =  0.5*( y(1)*y(6)+y(2)*y(7)-y(3)*y(4));
    dedt3 =  0.5*(-y(1)*y(5)+y(2)*y(4)+y(3)*y(7));
    dedt4 = -0.5*( y(1)*y(4)+y(2)*y(5)+y(3)*y(6));

    dwdt(4) = dedt1;
    dwdt(5) = dedt2;
    dwdt(6) = dedt3;
    dwdt(7) = dedt4;

    dwdt(8) = y(4)^2 + y(5)^2 + y(6)^2 + y(7)^2 - 1; % Euler Constraint


    e = [y(4) y(5) y(6) y(7)];
    C = DCM_from_EulerPara(e); % DCM

    % Kinematic EOM of angular velocities and direction cosines
    dwdt(9) = C(1,2)*y(3)-C(1,3)*y(2);
    dwdt(10) = C(1,3)*y(1)-C(1,1)*y(3);
    dwdt(11) = C(1,1)*y(2)-C(1,2)*y(1);
    dwdt(12) = C(2,2)*y(3)-C(2,3)*y(2);
    dwdt(13) = C(2,3)*y(1)-C(2,1)*y(3);
    dwdt(14) = C(2,1)*y(2)-C(2,2)*y(1);
    dwdt(15) = C(3,2)*y(3)-C(3,3)*y(2);
    dwdt(16) = C(3,3)*y(1)-C(3,1)*y(3);
    dwdt(17) = C(3,1)*y(2)-C(3,2)*y(1);
end
```