1)

Given vector $x(t)$ & frame A

& $|x(t)|$ = constant

$\Rightarrow$ $\sqrt{x^T(t)\, x(t)}$ = constent

$x^T(t)\, x(t)$ = constant

differentiating bothsides w.r.t A

$\Rightarrow$ $\overset{A}{\dot{x}}{}^T(t)\, x(t) + x^T(t)\, \overset{A}{\dot{x}}(t) = 0$

$\overset{A}{\dot{x}}{}^T(t)$ if $p^T q$ = scalar

$\Rightarrow$ (scalar)^T = scalar $\Rightarrow$ $p^T q = q^T p$

$\Rightarrow$ $2\, \overset{A}{\dot{x}}{}^T(t)\, x(t) = 0$

$\Rightarrow$ Dot product of $\overset{A}{\dot{x}}(t)$ & $x(t)$ is

zero

$\Rightarrow$ Both $\overset{A}{\dot{\vec{x}}}(t)$ and $\vec{x}(t)$ are

mutually orthogonal

2)

From transport theorm

$$ {}^{A}\dot{\vec{r}} = {}^{B}\dot{\vec{r}} + \omega_{B/A} \times \vec{r} \qquad -(1)$$

where $\vec{r}$ is a physical vector & . A & B are frames

differentiating above eq^n

$$ {}^{A}\ddot{\vec{r}} = \overbrace{\left({}^{B}\dot{\vec{r}}\right)}^{A\cdot} + \overbrace{\left(\omega_{B/A} \times \vec{r}\right)}^{A\cdot} \qquad -(2)$$

$$ \to \overbrace{\left({}^{B}\dot{\vec{r}}\right)}^{A\cdot} = {}^{B}\ddot{\vec{r}} + \omega_{B/A} \times {}^{B}\dot{\vec{r}} \qquad -(3)$$

↳ from transport theorm

Substitue (3) in (2)

$$ {}^{A}\ddot{\vec{r}} = {}^{B}\ddot{\vec{r}} + \omega_{B/A} \times {}^{B}\dot{\vec{r}} + \overbrace{\left(\omega_{B/A} \times \vec{r}\right)}^{A\cdot}$$

$$ = {}^{B}\ddot{\vec{r}} + \omega_{B/A} \times {}^{B}\dot{\vec{r}} + {}^{A}\dot{\omega}_{B/A} \times \vec{r} + \omega_{B/A} \times {}^{A}\dot{\vec{r}}$$

using (1)

$$ = {}^{B}\ddot{\vec{r}} + \omega_{B/A} \times {}^{B}\dot{\vec{r}} + {}^{A}\dot{\omega}_{B/A} \times \vec{r} + \omega_{B/A} \times \left({}^{B}\dot{\vec{r}} + \omega_{B/A}\vec{r}\right)$$

$$ = {}^{B}\ddot{\vec{r}} + \omega_{B/A} \times {}^{B}\dot{\vec{r}} + {}^{A}\dot{\omega}_{B/A} \times \vec{r} + \vec{\omega}_{B/A} \times {}^{B}\dot{\vec{r}}$$

$$ + \omega_{B/A} \times \left(\omega_{B/A} \times \vec{r}\right)$$

$$\Rightarrow \quad {}^{A}\ddot{\vec{r}} = {}^{B}\ddot{\vec{r}} + 2\,\omega_{B|A} \times {}^{B}\dot{\vec{r}} + {}^{A}\dot{\vec{\omega}}_{B|A} \times \vec{r} + \vec{\omega}_{B|A} \times \left(\vec{\omega}_{B|A} \times \vec{r}\right)$$

$$\vec{r} = \vec{r}_{m|w}$$

$$\rightarrow \quad {}^{A}\cancel{\ddot{\vec{r}}} = {}^{B}\cancel{\ddot{\vec{r}}}$$

$${}^{A}\ddot{\vec{r}}_{m|w} = {}^{B}\ddot{\vec{r}}_{m|w} + 2\,\omega_{B|A} \times {}^{B}\dot{\vec{r}}_{m|w} + {}^{A}\dot{\vec{\omega}}_{B|A} \times \vec{r}_{m|w} + \vec{\omega}_{B|A} \times \left(\vec{\omega}_{B|A} \times \vec{r}_{m|w}\right)$$

$\therefore$

$$\vec{a}_{m|w|A} = \underbrace{\vec{a}_{m|w|B}}_{\text{relative acceleration}} + \underbrace{2\,\vec{\omega}_{B|A} \times \vec{V}_{m|w|B}}_{\text{coriolis acceleration}} + \underbrace{{}^{A}\dot{\vec{\omega}}_{B|A} \times \vec{r}_{m|w}}_{\text{angular acceleration}} + \underbrace{\vec{\omega}_{B|A} \times \left(\vec{\omega}_{B|A} \times \vec{r}_{m|w}\right)}_{\text{centripetal acceleration}}$$

3) i,

$$\omega = S(\phi, \theta)\,\dot{\theta}$$

$$S(\phi, \theta) = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix}$$

if $\theta = \pm\frac{\pi}{2}$

then

$$S = \begin{bmatrix} 1 & 0 & \pm 1 \\ 0 & \cos\phi & 0 \\ 0 & -\sin\phi & 0 \end{bmatrix}$$

This can also be seen as

$\det(S) = \cos\theta$ if $\theta = \pm\frac{\pi}{2}$

then $\det(S) = 0$

$\Rightarrow$ inverse of S can't be found

$\rightarrow$ there donot exist a function which can map all values of ω to $\dot{\theta}$ space

$\Rightarrow$ S is not bijective, So it means it can't Produce all angular velocities

3) (ii)

when $\theta = \pm \frac{\pi}{2}$

$$S = \begin{bmatrix} 1 & 0 & \pm 1 \\ 0 & \cos\phi & 0 \\ 0 & -\sin\phi & 0 \end{bmatrix}$$

$$\Rightarrow \omega_{D/A} = \begin{bmatrix} 1 & 0 & \pm 1 \\ 0 & \cos\phi & 0 \\ 0 & -\sin\phi & 0 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{\phi} \pm \dot{\psi} \\ \cos\phi\, \dot{\theta} \\ -\sin\phi\, \dot{\theta} \end{bmatrix}$$

if $\omega_{D/A} = \omega \hat{k}$ then $\dot{\phi} \pm \dot{\psi} = 0$

$\dot{\theta} \cos\phi = 0$ $\quad -\sin\phi\, \dot{\theta} \neq 0$

$\Rightarrow \cos\phi = 0 \Rightarrow \boxed{\phi = \pm \frac{\pi}{2}}$

& if $\phi = \pm\frac{\pi}{2}$ then $\omega_{D/A} = \begin{bmatrix} \dot{\phi} \pm \dot{\psi} \\ 0 \\ -\dot{\theta} \end{bmatrix}$ then $\omega_{D/A} = \omega \hat{k}$

Can be attainable when $\dot{\phi}$ & $\dot{\psi}$ are equal ~~opposi~~

4)

i) we need to find $\vec{\omega}_{Rotation/Star}$

given Earth rotates w.r.t itself in 24 hours

$\Rightarrow \quad \vec{\omega}_{Rotation|Earth} = \frac{2\pi}{24}$ rad/hrs

& Given Earth is under rotation around Sun and ~~for~~ completes rotation every 365.25 solar days

$\Rightarrow \quad \vec{\omega}_{Earth|Star} = \frac{2\pi}{(365.25 \times 24)}$ (rad/hrs)

$$\Rightarrow \vec{\omega}_{Rotation|Star} = \vec{\omega}_{Rotation|Earth} + \vec{\omega}_{Earth|Star}$$

$$= \frac{2\pi}{24} + \frac{2\pi}{365.25 \times 24}$$

$$= 2\pi\left(\frac{366.25}{365.25 \times 24}\right)$$

$\Rightarrow$ time for rotation $= \left(\frac{365.25}{366.25}\right) 24$ hrs

$= 23.9345$ hrs

$= 23$ hr 56 min 4.2 sec

4) ii)

now weneed to find $\vec{\omega}$ rotation | sun

$$\Rightarrow \vec{\omega}_{\text{Rotation} \mid \text{sun}} = \vec{\omega}_{\text{Rotation} \mid \text{star}} + \vec{\omega}_{\text{star} \mid \text{sun}}$$

$$= \vec{\omega}_{\text{Rotation} \mid \text{star}} - \vec{\omega}_{\text{sun} \mid \text{star}}$$

As sun is rotating about itself by, 27 solar days taking for rotation.

$$\vec{\omega}_{\text{Rotational} \mid \text{Sun}} = \frac{2\pi}{23.9345 \text{ hrs}} + \frac{2\pi}{27 \times 24 \text{ hrs}}$$

$$= \frac{2\pi}{24}\left(\frac{1}{0.9972} - \frac{1}{27}\right)\left(\frac{\text{rad}}{\text{hrs}}\right)$$

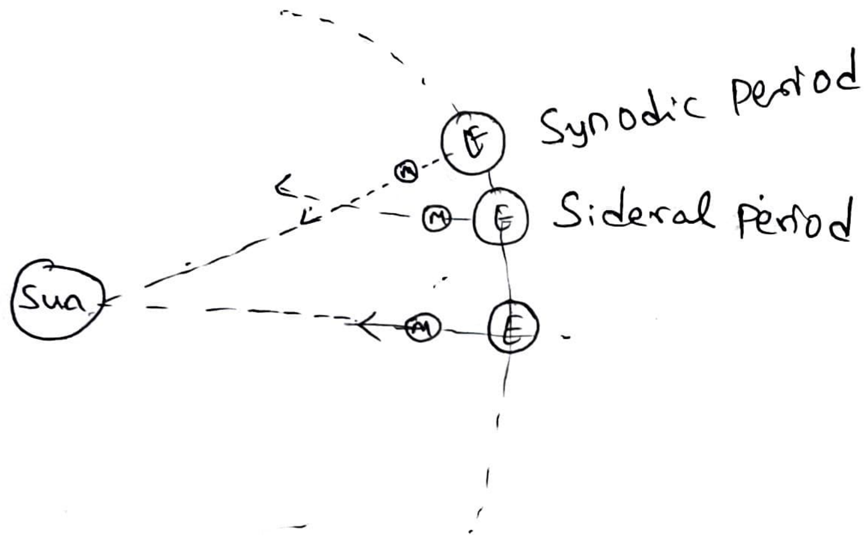
$\Rightarrow$ Time of rotation w.r.t sun body fixed frame is

is $24 \times \left(\dfrac{27 \times 0.9972}{27 - 0.9972}\right)$

$24 \times 1.0356 = 24.8524$

$=$ 24 hrs 51 minutes 8.64 sec

4) iii)



=) Synodic period can be obtained by finding $\omega_{\text{Moon}|\text{Earth}}$

$$\omega_{\text{moon}|\text{Earth}} = \omega_{\text{moon}|\text{Star}} - \omega_{\text{Earth}|\text{Star}}$$

$$= \left(\frac{2\pi}{27.3} - \frac{2\pi}{365.25}\right) \text{(rad/days)}$$

$$= 2\pi \times 0.0339$$

Time for Synodic period $= \frac{1}{0.0339}$

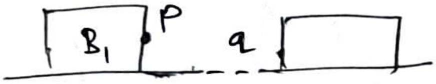
$= 29.5$ days

5)

Given mass of $B_1 = m_1$

mass of $B_2 = m_2$

lets take a point P on B_1 to its right & point q on B_2 to its left



$$\vec{r}_{P|q} = \vec{r}_{P|world} - \vec{r}_{q|world}$$

as Both are in contact throught $\vec{r}_{P|q}$ is constent

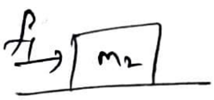
$$\Rightarrow \dot{\vec{r}}_{P|world|world} = \dot{\vec{r}}_{q|world|world}$$

$$\ddot{\vec{r}}_{P|world|world} = \ddot{\vec{r}}_{q|world|world}$$

$$\Rightarrow \ddot{\vec{r}}_{B_1|world|world} = \ddot{\vec{r}}_{B_2|world|world} \quad (\because \text{rigid body})$$

& from newton's 3rd law reaction force on Both is same

B_1

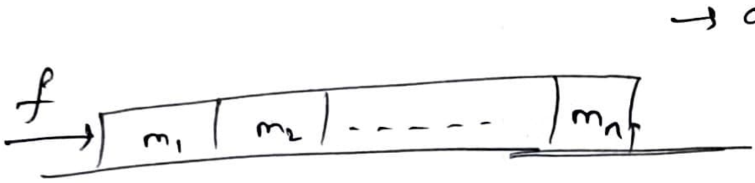


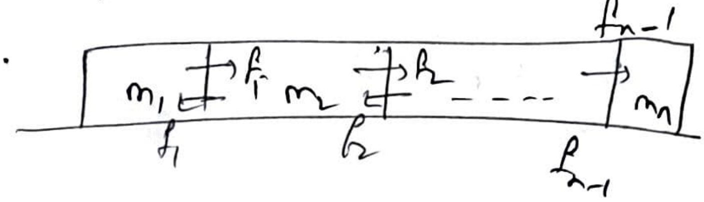
$$(m_1+m_2)a = f \Rightarrow \boxed{a = \frac{f}{m_1+m_2}}$$

& $f_1 = m_2 a_1$

$$\Rightarrow \boxed{f_1 = \frac{f m_2}{m_1+m_2}}$$

if Expanded to n body problem



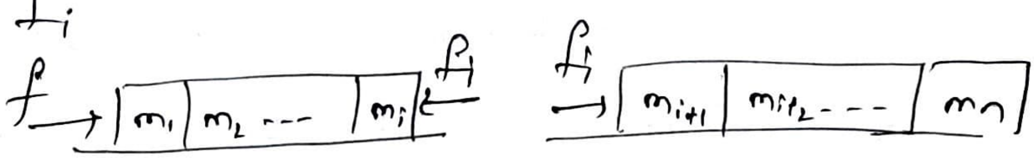


$$(m_1+m_2+\dots m_n)\,a = f$$

$$\boxed{a = \frac{f}{(m_1+m_2+\dots m_n)}}$$

where f_i is interaction force between m_i & m_{i+1}

if we take f_i



$$f_i = (m_{i+1}+m_{i+2}\dots m_n)\,a$$

$$\Rightarrow \boxed{f_i = \frac{\left(\sum_{j=i+1}^{n} m_j\right)}{\left(\sum_{k=1}^{n} m_k\right)}\, f}$$

6)

using Euler eq^n

$$\vec{M}_{B/\cancel{Z}} = \vec{J}_{B/c} \overset{B.}{\vec{\omega}}_{B/A} + \vec{\omega}_{B/A} \times \vec{J}_{B/c} \vec{\omega}_{B/A}$$

$$+ \cancel{r} \; r_{c/z} \times m_B \, a_{c/w/A}$$

as It said

$$\vec{M}_{B/Z} = \text{External moment} + \vec{r}_{c/z} \times m_B \vec{g}$$

as ~~there~~ are no External moments

External moments $= 0$

and as it is rotating about center of mass

$$\Rightarrow \vec{r}_{c/z} = 0.$$

& as it is rotating through principle axis

so. $\vec{\omega}_{B/A}$ & $\vec{J}_{B/c} \vec{\omega}_{B/A}$ i.e. ~~linear momentum~~ Angular velocity & momentum will be parallel

$$\Rightarrow \vec{\omega}_{B/A} \times \vec{J}_{B/c} \vec{\omega}_{B/A} = 0.$$

$$\Rightarrow \quad 0 = \vec{J}_{B/C} \, {}^{B}\dot{\omega}_{B/A} + 0 + 0$$

$$\Rightarrow \quad {}^{B}\dot{\omega}_{B/A} = 0$$

$$\Rightarrow \quad \omega_{B/A}\big|_{B} = \text{Constant}$$

$\Rightarrow$ for all time t the Body keeps Rotating

$\therefore$ it rotates indefinitely.

hw_1.m - top level code for the problem

Plots and helper codes attached after this code

```
%% Problem 7
%%
%% Initialise Parameters
t0=0; tf=10;
euler_initial=[0 0 0];
orientation_matrix_initial = [1, 0, 0; 0, 1, 0; 0, 0, 1];
%% Part 1 - Obtaining Euler_angles from Euler derivative
sol = ode45(@(t,y) euler_dot(t,y),[t0, tf],euler_initial); % Integrating
4.10.10 using ode45
t_1 = linspace(0,10,1000); % Time frame
euler_angles_o = deval(sol,t_1); % Obtaining Function for all reqired time of
evaluation
t_1 = t_1.';
omega_D_frame = [cos(2*t_1), cos(2*t_1), 0.025*t_1].'; % Defining Omega
%%% Orientation Matrix
O_1 = zeros(3,3,length(omega_D_frame));
for i = 1:length(omega_D_frame)
   % Orientation Matrix generated from Euler to Rotation Matrix
   O = orientation_matrix_euler(euler_angles_o(:,i));
   O_1(:,:,i) = O;
end
%% Part 2 - Orientation Matrix using Poission Integral
sol = ode45(@(t,O_linear) poisson_integral(t,O_linear),[t0,
tf],orientation_matrix_initial); %% Poission Integral using ode45
t_2 = linspace(0,10,1000); % Time frame
O_2_linear = deval(sol,t_2); % Obtaining Function for all reqired time of
evaluation
t_2 = t_2.';
%% Part_3 - Comparing Euler using both methods
euler_angles_set_1 = zeros(3,length(t_1));
euler_angles_set_2 = zeros(3,length(t_2));
for i = 1:length(t_1)
   % Euler angle using Euler_derivative Integrals
   euler_angles_set_1(:,i) = euler_from_rotation(O_1(:,:,i));

   % Euler Angle using Poission Integral
   O(1,1) = O_2_linear(1,i);
   O(1,2) = O_2_linear(2,i);
   O(1,3) = O_2_linear(3,i);
   O(2,1) = O_2_linear(4,i);
   O(2,2) = O_2_linear(5,i);
   O(2,3) = O_2_linear(6,i);
   O(3,1) = O_2_linear(7,i);
   O(3,2) = O_2_linear(8,i);
```

```
    O(3,3) = O_2_linear(9,i);
    euler_angles_set_2(:,i) = euler_from_rotation(O);
end
%% Plotting
%%% Plot for euler angles using part 1
fig1 = figure(1);
fig1.Position = [10 10 900 600];
ax1 = axes(fig1);
for i = 1:3
    subplot(3,1,i);
    plot(t_1, euler_angles_o(i,:));
    if i==1
        title('Euler from Euler Derivative');
        ax = gca;
        ax.TitleFontSizeMultiplier = 1;
    end
end
saveas(fig1,'./results/Euler_from_part1.png');
%%% Plot for Omega
fig2 = figure(2);
fig2.Position = [10 10 900 600];
ax2 = axes(fig2);
for i = 1:3
    subplot(3,1,i);
    plot(t_1, omega_D_frame(i,:));
    if i==1
        title('Omega Values');
        ax = gca;
        ax.TitleFontSizeMultiplier = 1;
    end
end
saveas(fig2,'./results/omega_from_part1.png');
%%% Plot for Orientation matrix using Poisson
fig3 = figure(1);
fig3.Position = [10 10 900 600];
ax3 = axes(fig3);
for i = 1:3
    for j = 1:3
        subplot(3,3,(i-1)*3+j);
        O = reshape(O_1(i,j,:), [1,length(O_1(i,j,:))]);
        plot(t_1, O);
        ith = string(i);
        jth = string(j);
        title(ith+jth);
        ax = gca;
        ax.TitleFontSizeMultiplier = 0.5;
    end
end
saveas(fig3,'./results/orientation_matrix_poission.png');
```

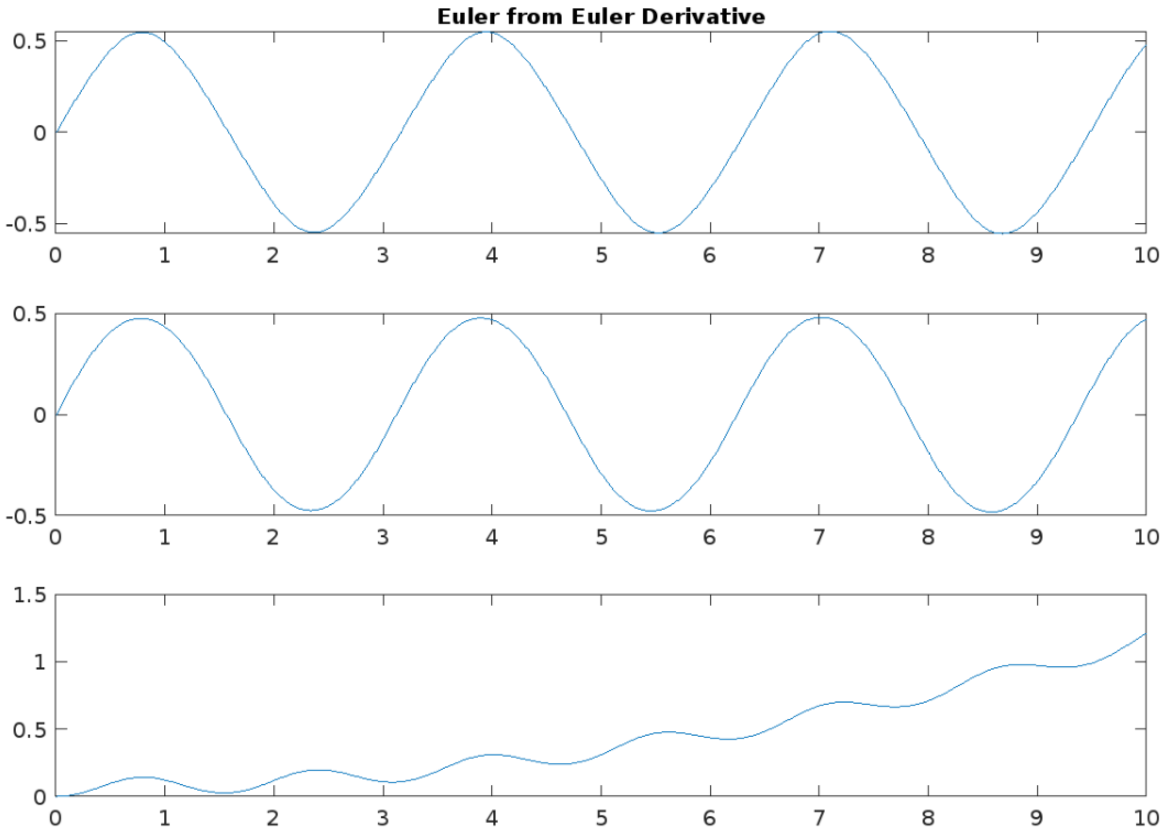
```
%%% Plot to compare Orientation: blue - from first part & Red from second
fig4 = figure(1);
fig4.Position = [10 10 900 600];
ax4 = axes(fig4);
for i = 1:3
   for j = 1:3
       subplot(3,3,(i-1)*3+j);
       O_a = reshape(O_1(i,j,:), [1,length(O_1(i,j,:))]);
       O_b = reshape(O_2_linear(((i-1)*3)+j,:),
[1,length(O_2_linear((i-1)*3+j,:))]);
       plot(t_1, O_a, '-b',t_2, O_b, '--r');
       ith = string(i);
       jth = string(j);
       title(ith+jth);
       ax = gca;
       ax.TitleFontSizeMultiplier = 0.5;
   end
end
saveas(fig4,'./results/orientation_matrix_comapre.png');
%%% Plot to compare Euler Angles: blue - from first part & Red from second
fig5 = figure(1);
fig5.Position = [10 10 900 600];
ax5 = axes(fig5);
for j = 1:3
   subplot(3,1,j);
   plot(t_1, euler_angles_set_1(j,:), '-b',t_2, euler_angles_set_2(j,:),
'--r');

   if j==1
       title('Euler angles: From Euler derivative - blue; From Poisson - red
');
       ax = gca;
       ax.TitleFontSizeMultiplier = 1;
   end
end
saveas(fig5,'./results/euler_compare.png');
```

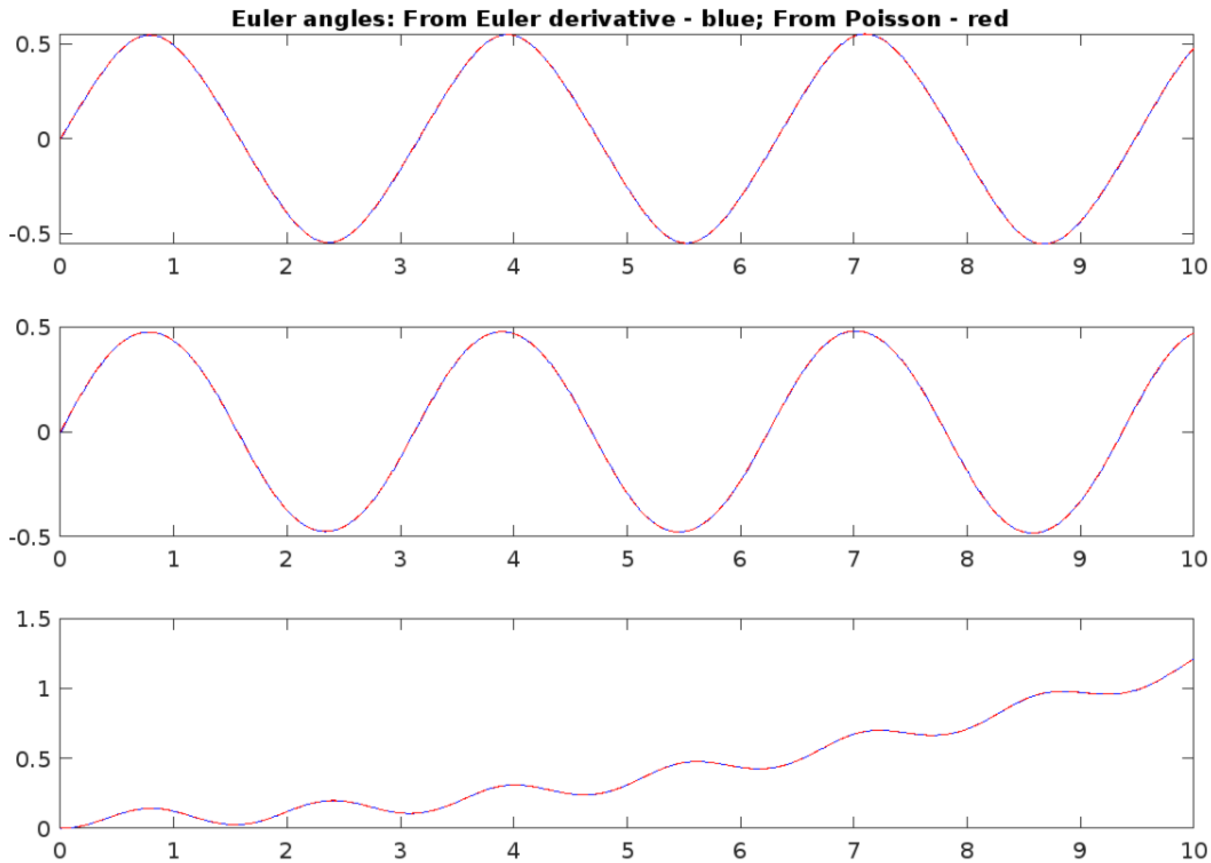
—------- Plots and Helper functions in next pages —-----------

Plots:

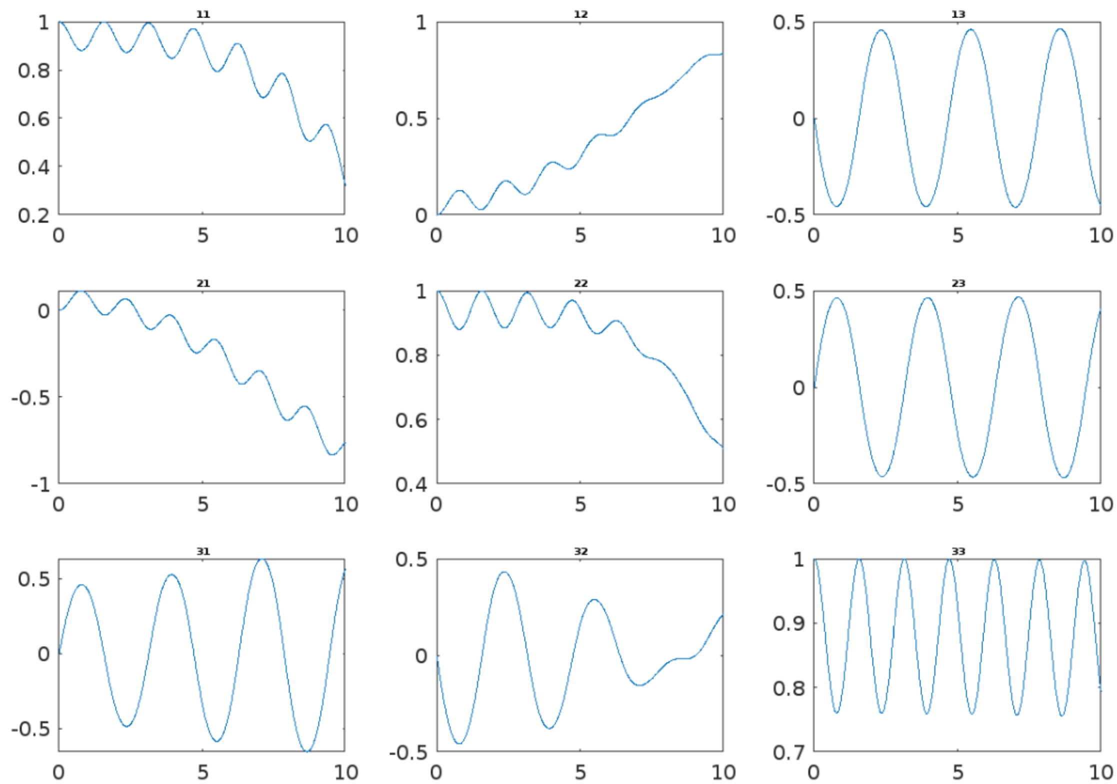
Euler from Euler derivative : In Psi theta Phi order



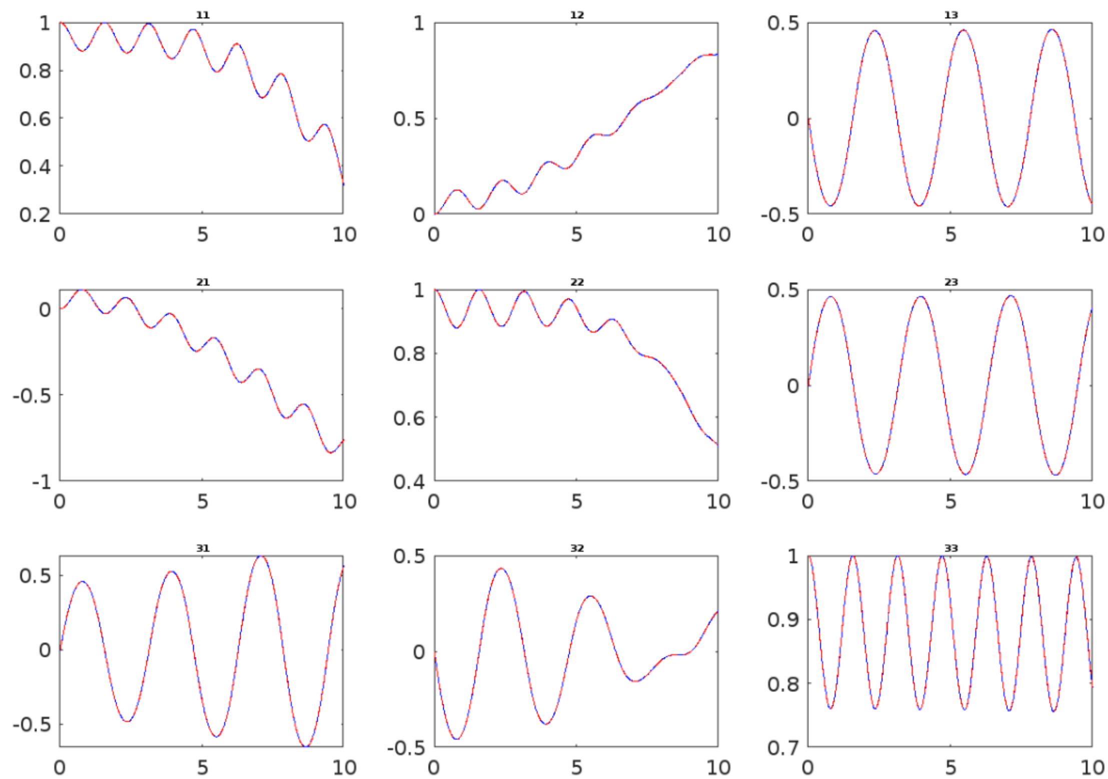
Euler Comparision



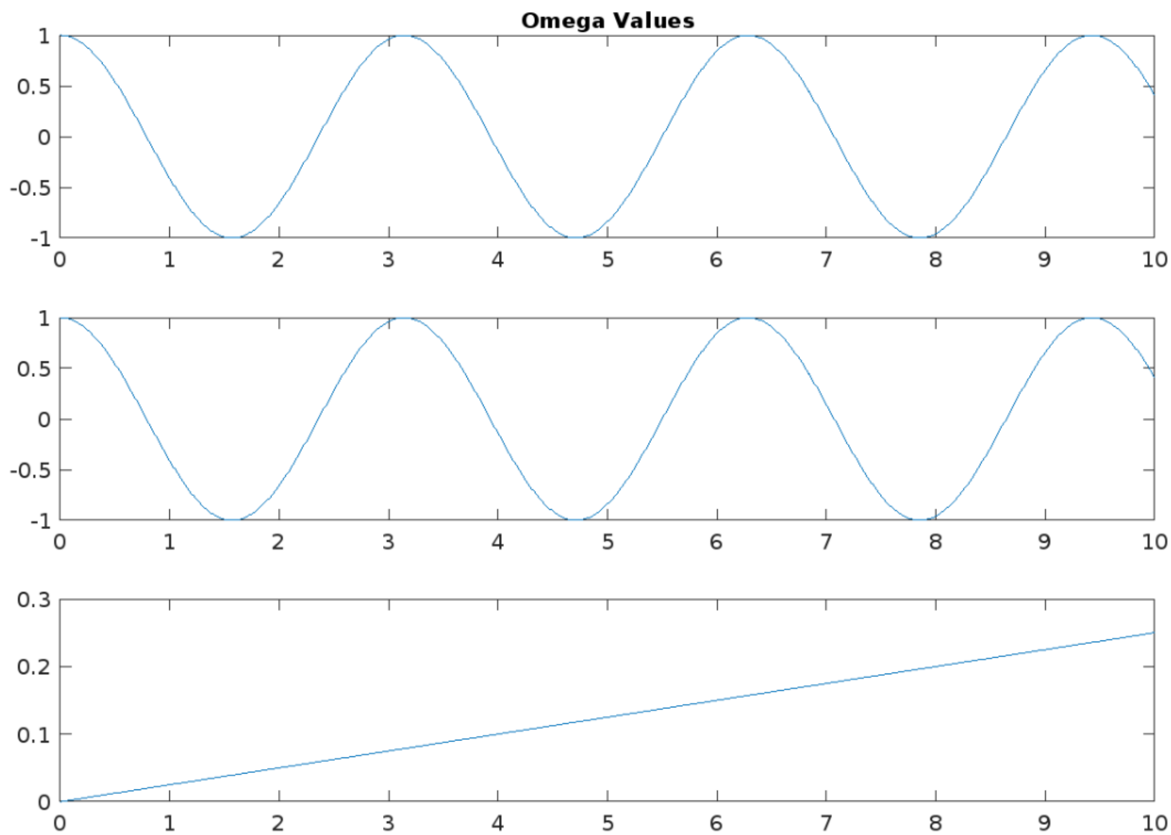
Orientation Matrix From Poisson:



Orientation Matrix Compare:
Part 7.b : Yes both Coincide



Omega Values:



euler_dot.m :

```
function xdot = euler_dot(t,euler_angles)
xdot = zeros(3,1);
phi = euler_angles(1);
theta = euler_angles(2);
psi = euler_angles(3);
omega = [cos(2*t), cos(2*t), 0.025*t];
s_inverse = [1,  sin(phi)*tan(theta), cos(phi)*tan(theta);
             0, cos(phi), -sin(phi);
             0, sin(phi)*sec(theta), cos(phi)*sec(theta)];
xdot(1) = omega(1)*s_inverse(1,1) + omega(2)*s_inverse(1,2) +
omega(3)*s_inverse(1,3);
xdot(2) = omega(1)*s_inverse(2,1) + omega(2)*s_inverse(2,2) +
omega(3)*s_inverse(2,3);
xdot(3) = omega(1)*s_inverse(3,1) + omega(2)*s_inverse(3,2) +
omega(3)*s_inverse(3,3);
end
```

—--

euler_from_rotation.m:

```
function euler_angles = euler_from_rotation(R)
euler_angles = zeros(3,1);
if abs(R(3,1)) ~= 1

   %% theta vals
   theta1 = -asin(R(1,3));
   theta2 = pi - theta1;
   %% psi vals
   psi1 = atan2((R(2,3)/cos(theta1)) , (R(3,3)/cos(theta1)));
   psi2 = atan2((R(2,3)/cos(theta2)) , (R(3,3)/cos(theta2)));
   %% phi vals
   phi1 = atan2((R(1,2)/cos(theta1)) , (R(1,1)/cos(theta1)));
   phi2 = atan2((R(1,2)/cos(theta2)) , (R(1,1)/cos(theta2)));
   euler_angles = [psi1, theta1, phi1];
else
   phi = 0;
   if R(1,3)==-1
       theta = pi/2;
       psi = phi + atan2(R(2,1), R(3,1));
   else
       theta = -pi/2;
       psi = -phi + atan2(-R(2,1), -R(3,1));
   end
```

```
   euler_angles = [ psi,theta, phi];
end
end
```

poisson_integral.m

```
%% Poisson's Integral Implementation
function O_dot_linear = poisson_integral(t,O_linear)
O = [O_linear(1), O_linear(2), O_linear(3);
    O_linear(4), O_linear(5), O_linear(6);
    O_linear(7), O_linear(8), O_linear(9);];
O_dot = zeros(3,3);
O_dot_linear = zeros(9,1);
omega = [cos(2*t),cos(2*t),0.025*t];
omega_cross = [0, -omega(3), omega(2);
               omega(3), 0, -omega(1);
               -omega(2), omega(1), 0];
omega_cross = omega_cross*-1;
%% O_dot elements
O_dot(1,1) =  omega_cross(1,1)*O(1,1) + omega_cross(1,2)*O(2,1) +
omega_cross(1,3)*O(3,1);
O_dot(1,2) =  omega_cross(1,1)*O(1,2) + omega_cross(1,2)*O(2,2) +
omega_cross(1,3)*O(3,2);
O_dot(1,3) =  omega_cross(1,1)*O(1,3) + omega_cross(1,2)*O(2,3) +
omega_cross(1,3)*O(3,3);
O_dot(2,1) =  omega_cross(2,1)*O(1,1) + omega_cross(2,2)*O(2,1) +
omega_cross(2,3)*O(3,1);
O_dot(2,2) =  omega_cross(2,1)*O(1,2) + omega_cross(2,2)*O(2,2) +
omega_cross(2,3)*O(3,2);
O_dot(2,3) =  omega_cross(2,1)*O(1,3) + omega_cross(2,2)*O(2,3) +
omega_cross(2,3)*O(3,3);
O_dot(3,1) =  omega_cross(3,1)*O(1,1) + omega_cross(3,2)*O(2,1) +
omega_cross(3,3)*O(3,1);
O_dot(3,2) =  omega_cross(3,1)*O(1,2) + omega_cross(3,2)*O(2,2) +
omega_cross(3,3)*O(3,2);
O_dot(3,3) =  omega_cross(3,1)*O(1,3) + omega_cross(3,2)*O(2,3) +
omega_cross(3,3)*O(3,3);
%%
O_dot_linear(1) = O_dot(1,1);
O_dot_linear(2) = O_dot(1,2);
O_dot_linear(3) = O_dot(1,3);
O_dot_linear(4) = O_dot(2,1);
O_dot_linear(5) = O_dot(2,2);
O_dot_linear(6) = O_dot(2,3);
O_dot_linear(7) = O_dot(3,1);
O_dot_linear(8) = O_dot(3,2);
O_dot_linear(9) = O_dot(3,3);
end
```

orientation_matrix_euler.m

```matlab
function matrix = orientation_matrix_euler(euler_angles)
matrix = zeros(3,3);
a = euler_angles(1);
b = euler_angles(2);
c = euler_angles(3);
%% Orientation matrix elements
matrix(1,1) = cos(b)*cos(c);
matrix(1,2) = cos(b)*sin(c);
matrix(1,3) = -sin(b);
matrix(2,1) = (cos(c)*sin(a)*sin(b))-(cos(a)*sin(c));
matrix(2,2) = (sin(c)*sin(a)*sin(b))+(cos(a)*cos(c));
matrix(2,3) = cos(b)*sin(a);
matrix(3,1) = (cos(c)*cos(a)*sin(b))+(sin(a)*sin(c));
matrix(3,2) = (cos(a)*sin(c)*sin(b))-(sin(a)*cos(c));
matrix(3,3) = cos(a)*cos(b);
%%
end
```