ENPM 662 – Homework 2 Report

1.1 Composition of Transforms

The 4 x 4 homogeneous matrix is of the form: $\begin{bmatrix} R_{3x3} & T_{3x1} \\ 0_{1x3} & 1_{1x1} \end{bmatrix}$

<u>Given sequence of rotations and translations:</u>

1. Rotate by ϕ about the world (fixed) z-axis.

The 4 x 4 homogeneous transformation matrix $R_{z,\phi}$ is given by:

$$R_{z,\phi} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Translate by y along the current y-axis.

The 4 x 4 homogeneous transformation matrix $T_{y,y}$ is given by:

$$T_{y,y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. Rotate by θ about the current z-axis.

The 4 x 4 homogeneous transformation matrix $R_{z,\theta}$ is given by:

$$R_{z,\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4. Rotate by ψ about the world (fixed) x-axis.

   The 4 x 4 homogeneous transformation matrix $R_{x,\psi}$ is given by:

$$R_{x,\psi} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi & 0 \\ 0 & \sin\psi & \cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The matrix production equation that will give the resulting pose of the frame is given by:

$R = [R_{x,\psi}] \cdot [R_{z,\phi}] \cdot [T_{y,y}] \cdot [R_{z,\theta}]$

The reason why I chose this order of multiplication:

According to the rule of composition of homogeneous transformation matrices,

- If the rotation/translation is performed relative to the **current frame**, then we must post multiply the homogeneous matrix (multiply the homogeneous matrix **in the order** of rotation/translation) to get the resultant homogeneous matrix.
- If the rotation/translation is performed relative to the **fixed frame**, then we must pre multiply the homogeneous matrix (multiply the homogeneous matrix in the **reverse order** of rotation/translation) to get the resultant homogeneous matrix.
- If the rotation/translation is performed **both** with respect to **fixed and current** frame, then the final resultant matrix is the product of the resultant matrix with respect to fixed frame and the resultant matrix with respect to current frame.
  i.e. $R = [R_{fixed}] \cdot [R_{current}]$

In the present case,

$R_{world(fixed)} = [R_{x,\psi}] \cdot [R_{z,\phi}]$

$R_{current} = [T_{y,y}] \cdot [R_{z,\theta}]$

Therefore, $R = [R_{world(fixed)}] \cdot [R_{current}]$

$$R = [R_{x,\psi}] \cdot [R_{z,\phi}] \cdot [T_{y,y}] \cdot [R_{z,\theta}]$$

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi & 0 \\ 0 & \sin\psi & \cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In [3]:

```
from sympy import *

theta = symbols("theta")
phi = symbols("phi")
psi = symbols("psi")
y = symbols("y")

R_z_phi = Matrix([[cos(phi),-sin(phi),0,0],[sin(phi),cos(phi),0,0],[0,0,1,0],[0,0,0,1]])

T_y_y = Matrix([[1,0,0,0],[0,1,0,y],[0,0,1,0],[0,0,0,1]])

R_z_theta = Matrix([[cos(theta),-sin(theta),0,0],[sin(theta),cos(theta),0,0],[0,0,1,0],[0,0,0,1]])

R_x_psi = Matrix([[1,0,0,0],[0,cos(psi),-sin(psi),0],[0,sin(psi),cos(psi),0],[0,0,0,1]])

R = R_x_psi * R_z_phi * T_y_y * R_z_theta
```

In [4]:

```
R
```

Out[4]:

$$\begin{bmatrix} -\sin(\phi)\sin(\theta) + \cos(\phi)\cos(\theta) & -\sin(\phi)\cos(\theta) - \sin(\theta)\cos(\phi) & 0 & -y\sin(\phi) \\ \sin(\phi)\cos(\psi)\cos(\theta) + \sin(\theta)\cos(\phi)\cos(\psi) & -\sin(\phi)\sin(\theta)\cos(\psi) + \cos(\phi)\cos(\psi)\cos(\theta) & -\sin(\psi) & y\cos(\phi)\cos(\psi) \\ \sin(\phi)\sin(\psi)\cos(\theta) + \sin(\psi)\sin(\theta)\cos(\phi) & -\sin(\phi)\sin(\psi)\sin(\theta) + \sin(\psi)\cos(\phi)\cos(\theta) & \cos(\psi) & y\sin(\psi)\cos(\phi) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig.1 Python(sympy) script for finding the value of R

1.2 Modelling beyond rigid transformations

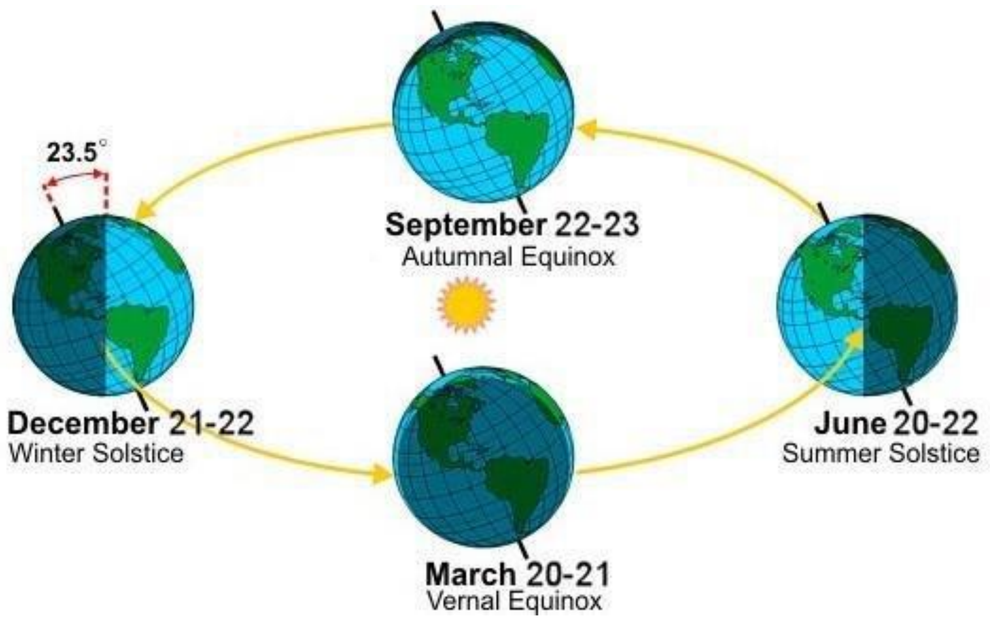


Fig. 2 - Given

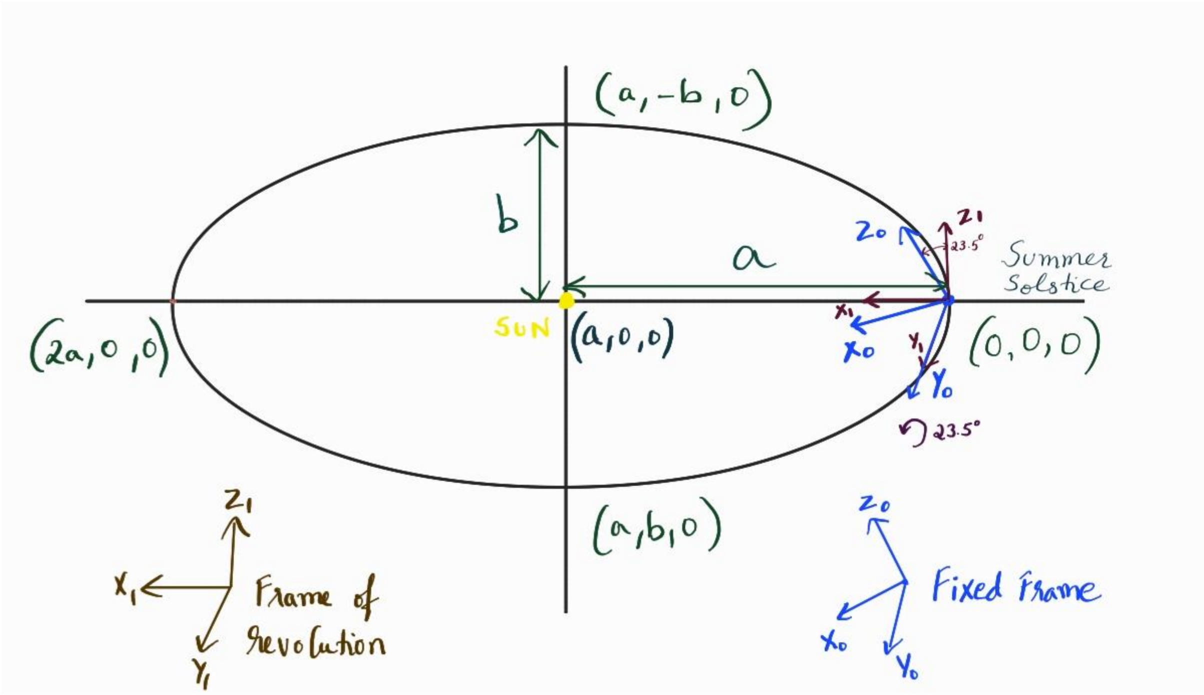


Fig. 3 – Ellipse and frames

- Let the frame of reference of earth be represented by: F_e = (x_e , y_e , z_e)^T.
- Let fixed frame (i.e. the frame of reference of earth at time t=0) be represented by: F_f = (x_0 , y_0 , z_0)^T {i.e. F_e at time t=0 is F_f}.
- Given that the orbit of earth is elliptical, and earth completes one rotation in 24 hours and 1 revolution in 365 days.

To find: Homogeneous matrix M that specifies earth's frame (F_e) w.r.t Fixed Frame (F_f).

To find the value of M, we must consider both the rotation and revolution of earth. It is assumed that the time variable 't' is measured in hours.
(i.e. Time increases in steps of 1 hour)

Let the angle of tilt of the earth's axis be represented by:

θ_1 = 23.5 degrees = $\frac{2*\pi*23.5}{360}$ radians

The homogeneous transformation matrix that represents the tilting of the earth (w.r.t current frame - about y axis) is given by:

$$H_{y,\theta 1} = \begin{bmatrix} R_{3x3} & T_{3x1} \\ 0_{1x3} & 1_{1x1} \end{bmatrix} = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_1 & 0 & \cos\theta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ (positive rotation)

$$H_{y,-\theta 1} = \begin{bmatrix} R_{3x3} & T_{3x1} \\ 0_{1x3} & 1_{1x1} \end{bmatrix} = \begin{bmatrix} \cos -\theta_1 & 0 & \sin -\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ -(\sin -\theta_1) & 0 & \cos -\theta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ (negative rotation)

Revolution:

- 1 year = 365 days = 365 * 24 hours = 8,760 hours
- The earth completes 1 revolution in 365 days. i.e., The earth revolute by 2π radians in 365 days.
- Hence, the movement of the earth around the sun (revolution) in 1 hour = $\frac{2 * \pi}{8760}$ radians.
- Therefore, the revolution of earth in 't' hours is: $\theta_2 = \frac{2 * \pi * t}{8760}$ radians

The orbit of the earth around the sun is elliptical.

Hence consider an ellipse with its centre at (h, k, 0), with semi-major axis length 'a' and semi-minor axis length 'b' as shown in figure 3. The parametric equation of this ellipse, with *respect to a point on its vertex*, is given by:

- x = h + (a * $\cos\theta$)
- y = k + (b * $\sin\theta$)
- z =0

Here, h=a and k=0 as the sun is at a distance of 'a' (along x axis) and 0 (along y axis) from revolution frame (x1 y1 z1)

i.e. the sun is at the coordinates (a,0,0) relative to the frame on its vertex (revolution frame as mentioned in fig.3).

Substituting:

- x = a + (a * $\cos\theta_2$)
- y = (b * $\sin\theta_2$)
- z = 0

✓ Here, the value of 'a' is the maximum distance between the sun and the earth and 'b' is the minimum distance between the sun and the earth. From astronomical data:

- a = 152.1 * $(10)^9$ meters
- b = 147.3 * $(10)^9$ meters

The homogeneous transformation matrix that represents the translation of the earth (w.r.t current frame – representing revolution) is given by:

$$H_{t,(x,y)} = \begin{bmatrix} R_{3x3} & T_{3x1} \\ 0_{1x3} & 1_{1x1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \mathrm{a} + (\mathrm{a} * \cos\theta_2) \\ 0 & 1 & 0 & (\mathrm{b} * \sin\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

<u>Rotation:</u>

- The earth completes 1 rotation in 24 hours. i.e., The earth rotates by 2π radians in 24 hours.
- Hence, the rotation of earth in 1 hour = $\frac{2 * \pi}{24}$ radians.
- Therefore, the rotation of earth in 't' hours is: $\theta_3 = \frac{2 * \pi * t}{24}$ radians

The homogeneous transformation matrix that represents the rotation of the earth (w.r.t current frame - about z axis) is given by:

$$H_{z,\theta_3} = \begin{bmatrix} R_{3x3} & T_{3x1} \\ 0_{1x3} & 1_{1x1} \end{bmatrix} = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & 0 \\ \sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Sequence of steps involved in finding the final homogeneous matrix:

i. Rotate the fixed frame (F_f) by -23.5 degrees about the y axis to get frame F_1(Frame of revolution).
ii. Translate the frame F_1 to a point specified by x and y parametric equations (derived above) with respect to time t, to get the frame F_2.
iii. Rotate the frame F_2 by +23.5 degrees about the y axis to get frame F_3.
iv. Rotate the frame F_3 by angle θ_3 about z axis to get the frame F_e.

i.e. The homogeneous transformation matrix is given by:

$$M = H_{y,-\theta 1} * H_{t,(x,y)} * H_{y,\theta 1} * H_{z,\theta 3}$$

Substituting the values,

$$M = \begin{bmatrix} \cos -\theta_1 & 0 & \sin -\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ -(\sin -\theta_1) & 0 & \cos -\theta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & (a + (a * \cos\theta_2)) \\ 0 & 1 & 0 & (b * \sin\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} *$$
$$\begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_1 & 0 & \cos\theta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & 0 \\ \sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
In [13]: from sympy import *

         theta_1 = symbols("theta_1")
         theta_2 = symbols("theta_2")
         theta_3 = symbols("theta_3")
         a = symbols("a")
         b = symbols("b")

         y_negt1 = Matrix([[cos(-(theta_1)),0,sin(-(theta_1)),0],[0,1,0,0],[-sin(-(theta_1)),0,cos(-(theta_1)),0],[0,0,0,1]])

         t_x_y = Matrix([[1,0,0,(a+(a*cos(theta_2)))],[0,1,0,b*sin(theta_2)],[0,0,1,0],[0,0,0,1]])

         y_post1 = Matrix([[cos(theta_1),0,sin(theta_1),0],[0,1,0,0],[-sin(theta_1),0,cos(theta_1),0],[0,0,0,1]])

         z_t3 = Matrix([[cos(theta_3),-sin(theta_3),0,0],[sin(theta_3),cos(theta_3),0,0],[0,0,1,0],[0,0,0,1]])

         M = y_negt1 * t_x_y * y_post1 * z_t3

In [14]: M
```

Out[14]:

$$\begin{bmatrix} (\sin^2(\theta_1) + \cos^2(\theta_1))\cos(\theta_3) & -(\sin^2(\theta_1) + \cos^2(\theta_1))\sin(\theta_3) & 0 & (a\cos(\theta_2) + a)\cos(\theta_1) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & b\sin(\theta_2) \\ 0 & 0 & \sin^2(\theta_1) + \cos^2(\theta_1) & (a\cos(\theta_2) + a)\sin(\theta_1) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig.4 Python(sympy) script for finding the value of M

$$M = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & (a + (a * \cos\theta_2)) * \cos\theta_1 \\ \sin\theta_3 & \cos\theta_3 & 0 & (b * \sin\theta_2) \\ 0 & 0 & 1 & (a + (a * \cos\theta_2)) * \sin\theta_1) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(as $\sin{\theta_1}^2 + \cos{\theta_1}^2 = 1$)

- $\theta_1 = 23.5$ degrees $= \frac{2 * \pi * 23.5}{360}$ rad, $\theta_2 = \frac{2 * \pi * t}{365 * 24}$ rad, $\theta_3 = \frac{2 * \pi * t}{24}$ rad
- $a = 152.1 * (10)^9$ meters, $b = 147.3 * (10)^9$ meters

Here, as the terms θ_2 and θ_3 are time variant, the homogeneous matrix M, changes with respect to time.

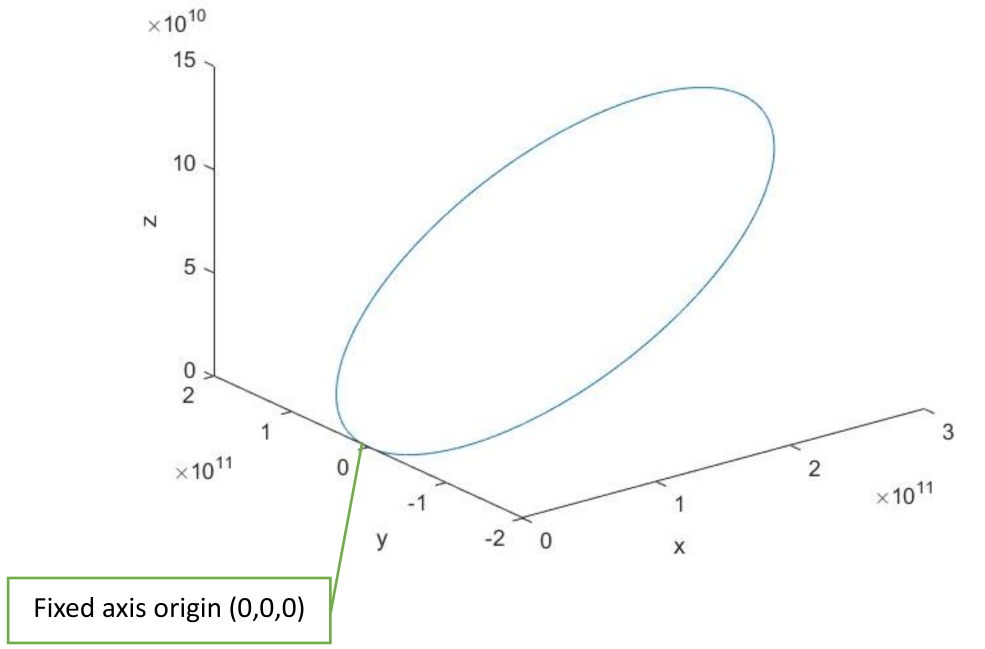
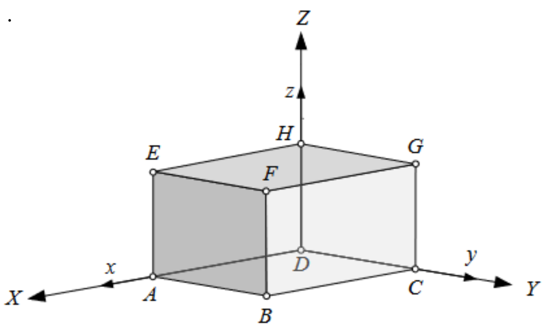
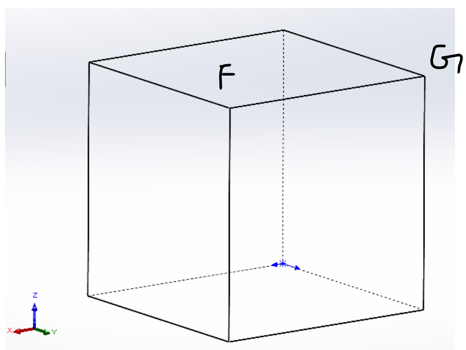


Fig.4 3d plot of the revolution w.r.t fixed axis (Matlab)

1.3 Transform Estimation

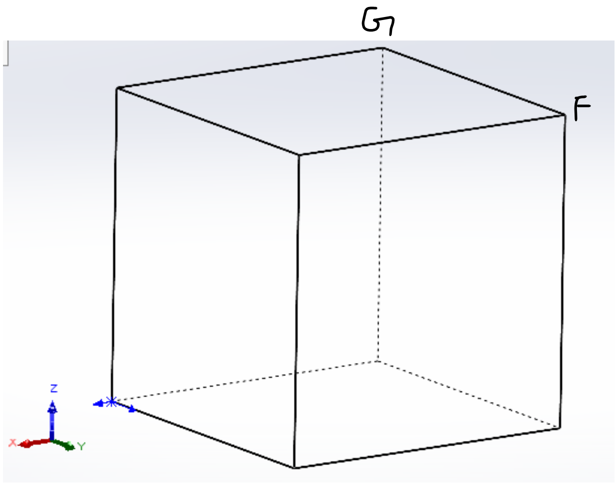




Given: Cube Side = 1m

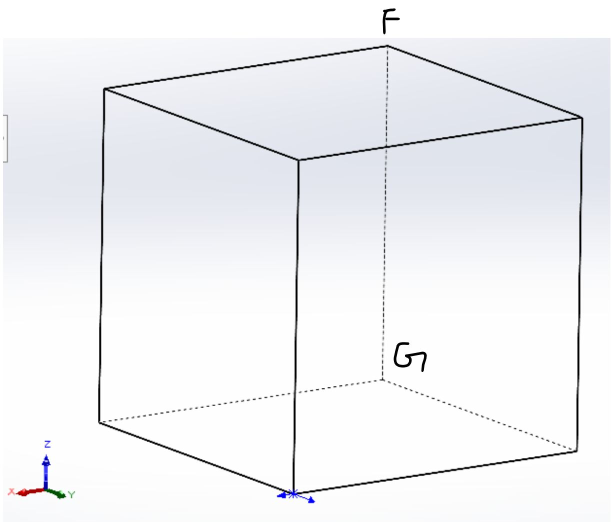
Sequence of translations and rotations that will result in a final position such that the EFGH plane on the cube coincides with the XZ plane, point F is on the Z axis and point G is on the X axis :

1. Rotate by 90 ^{0} about the Z-axis.



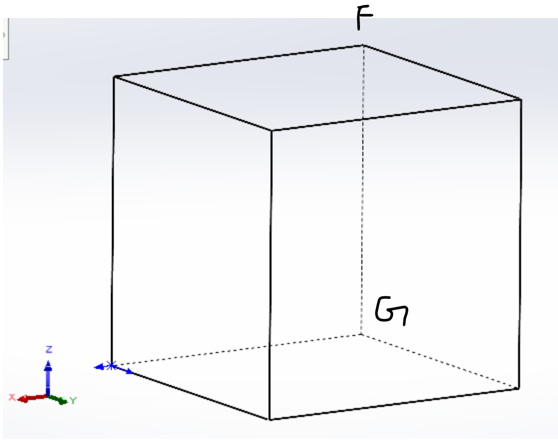
$$R_{z,90} = \begin{bmatrix} \cos 90 & -\sin 90 & 0 & 0 \\ \sin 90 & \cos 90 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Rotate by 90 ^{0} about the X-axis.



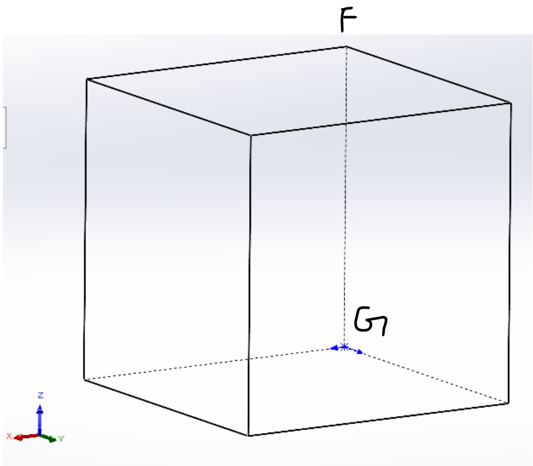
$$R_{x,90} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 90 & -\sin 90 & 0 \\ 0 & \sin 90 & \cos 90 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. Translate by 1m along the Y-axis.



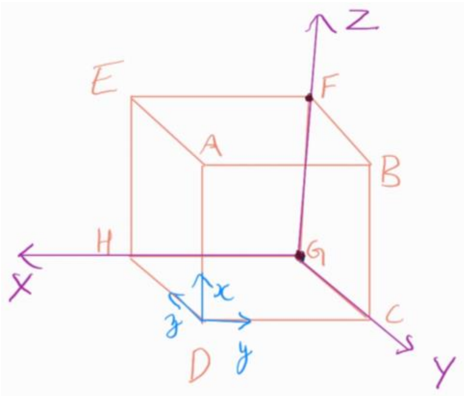
$$T_{y,1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4. Translate by 1m along the X-axis.



$$T_{x,1} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Final cube orientation:



The plane EFGH coincides with the X-Z plane and the point G is on the X axis and the point F is on the Z axis.

The homogeneous transformation matrix representing the above rotations and/or translations is given by:

H = $T_{x,1} * T_{y,1} * R_{x,90} * R_{z,90}$

H = $\begin{bmatrix} 0 & -1 & 0 & 1 \\ 0 & 1 & -1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ where R = $\begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & -1 \\ 1 & 0 & 0 \end{bmatrix}$ and T = $\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$

Finding Euler Angles XYZ from the Rotation matrix R = $\begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & -1 \\ 1 & 0 & 0 \end{bmatrix}$

R = $R_{x'y'z'} = R_x(\phi) * R_y(\theta) * R_z(\psi)$

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} \cos\psi * \cos\theta & -\cos\theta\sin\psi & \sin\theta \\ \sin\theta*\sin\phi*\cos\psi + \cos\phi*\sin\psi & -\sin\theta*\sin\phi*\sin\psi + \cos\psi*\cos\phi & -\sin\phi*\cos\theta \\ -\cos\phi*\sin\theta*\cos\psi + \sin\phi*\sin\psi & -\sin\psi*\cos\phi*\sin\theta + \sin\phi*\cos\psi & \cos\phi*\cos\theta \end{bmatrix}$$

$$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & -1 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \cos\psi * \cos\theta & -\cos\theta*\sin\psi & \sin\theta \\ \sin\theta*\sin\phi*\cos\psi + \cos\phi*\sin\psi & -\sin\theta*\sin\phi*\sin\psi + \cos\psi*\cos\phi & -\sin\phi*\cos\theta \\ -\cos\phi*\sin\theta*\cos\psi + \sin\phi*\sin\psi & -\sin\psi*\cos\phi*\sin\theta + \sin\phi*\cos\psi & \cos\phi*\cos\theta \end{bmatrix}$$

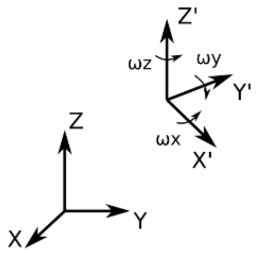
Equating,
Sin θ = 0
$\theta = \sin^{-1} 0 = 0^0$ or 180^0

Case 1: $\theta = 0^0$
$-(\cos\theta * \sin\psi) = -1$ and $-(\sin\phi * \cos\theta) = -1$
$\Psi = \sin^{-1} 1 = 90^0$ and $\phi = \sin^{-1} 1 = 90^0$

Case 2: $\theta = 180^0$
$-(\cos\theta * \sin\psi) = -1$ and $-(\sin\phi * \cos\theta) = -1$
$\Psi = \sin^{-1}(-1) = -90^0$ and $\phi = \sin^{-1}(-1) = -90^0$

Hence the Euler angles are $(\phi_x, \theta_y, \Psi_z) = (90^0, 0^0, 90^0)$ or $(-90^0, 180^0, -90^0)$
The translation along X, Y and Z axis: $(x,y,z)^T = T = (1, 1, 0)^T$
This solution is with respect to XYZ Euler angle representation.

2.1 Trajectory Optimization



<u>Given:</u>

Initial position --> [X , Y , Z]

Final position --> [X' , Y' , Z']

Final orientation: $\psi g = 45^\circ$,

$\theta g = 25^\circ$,

$\phi g = 16^\circ$

X-Y-Z Sequence

Maximum angular velocity = 2 deg/sec

<u>To compute</u>: Angular velocity trajectory (ψ, θ, ϕ, ωx, ωy, ωz w.r.t time)

<u>Solution:</u>

- It is assumed that the ψ, θ, ϕ represent the <u>consecutive rotations</u> with respect to the <u>global X,Y,Z axis</u> (fixed axis) to obtain the final orientation of the satellite. Hence, we can compute the final rotational matrix with this information.

$$R_{x,\psi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 45 & -\sin 45 \\ 0 & \sin 45 & \cos 45 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.70710 & -0.70710 \\ 0 & 0.70710 & 0.70710 \end{bmatrix}$$

$$R_{y,\psi} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} = \begin{bmatrix} \cos 25 & 0 & \sin 25 \\ 0 & 1 & 0 \\ -\sin 25 & 0 & \cos 25 \end{bmatrix} = \begin{bmatrix} 0.90630 & 0 & 0.42261 \\ 0 & 1 & 0 \\ -0.42261 & 0 & 0.90630 \end{bmatrix}$$

$$R_{z,\phi} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos 16 & -\sin 16 & 0 \\ \sin 16 & \cos 16 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.96126 & -0.27563 & 0 \\ 0.27563 & 0.96126 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

As the rotations are with respect to global/fixed axis XYZ, we must pre-multiply the rotational matrix to get the final rotational matrix.

$R = R_{z,\phi} * R_{y,\psi} * R_{x,\psi}$

$$R = \begin{bmatrix} 0.96126 & -0.27563 & 0 \\ 0.27563 & 0.96126 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 0.90630 & 0 & 0.42261 \\ 0 & 1 & 0 \\ -0.42261 & 0 & 0.90630 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.70710 & -0.70710 \\ 0 & 0.70710 & 0.70710 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.8712 & 0.0924 & 0.4821 \\ 0.2498 & 0.7621 & -0.5973 \\ 0.4226 & 0.6408 & 0.6408 \end{bmatrix}$$

Converting the Rotational matrix to axis angle representation:

Angle:

$\delta = \cos^{-1} (\frac{Trace(R)-1}{2})$ Where, $Trace(R) = r_{11} + r_{12} + r_{13}$

$$\delta = \cos^{-1} \left(\frac{\mathrm{r11 + r12 + r13} - 1}{2} \right) = \cos^{-1} \left(\frac{1.2741}{2} \right) = \cos^{-1} (0.63705)$$

$\delta = 50.4278^0$

Axis:

$$a = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \frac{1}{2 * \sin\delta} * \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} = \frac{1}{2 * \sin(50.4278)} * \begin{bmatrix} 0.6408 - (-0.5973) \\ 0.4821 - 0.4226 \\ 0.2498 - 0.0924 \end{bmatrix}$$

$$a = \frac{1}{2 * \sin(50.4278)} * \begin{bmatrix} 1.2381 \\ 0.9047 \\ 0.1574 \end{bmatrix}$$

$$a = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} 0.8031 \\ 0.5867 \\ 0.1028 \end{bmatrix}$$

and $\delta = 50.4278^0 = 0.8801$ radians

Finding the Angular velocity and shortest amount of time:

we know that, $\omega = \frac{angle}{time}$

It is given that, $\omega_x <= \omega_{max}$, $\omega_y <= \omega_{max}$ and $\omega_z <= \omega_{max}$ .... where ω_{max} = 2 degrees/second

Therefore, $\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \leq \omega_{max} \\ \leq \omega_{max} \\ \leq \omega_{max} \end{bmatrix} = \frac{\delta}{t} * \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$

Let us assume that ω_x , ω_y and ω_z are equal to ω_{max} to get the shortest time:

$\omega_x = 2 = \frac{\delta}{t_x} * a_x$

$t_x = \frac{\delta}{\omega_x} * a_x = \frac{50.42780}{2} * 0.8031 =$ 20.25 seconds

$t_y = \frac{\delta}{\omega_y} * a_y = \frac{50.42780}{2} * 0.5867 =$ 14.79 seconds

$t_z = \frac{\delta}{\omega_z} * a_z = \frac{50.42780}{2} * 0.1028 =$ 2.5919 seconds

max(t_x , t_y , t_z) = t_x = T = 20.25 seconds

Therefore, the shortest amount of time the satellite takes to reach its final orientation (considering angular velocity limit) is : **20.25 seconds**

$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \frac{\delta}{t} * \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \frac{50.42780}{20.25} * \begin{bmatrix} 0.8031 \\ 0.5867 \\ 0.1028 \end{bmatrix} = \begin{bmatrix} 2.000 \\ 1.4617 \\ 0.2561 \end{bmatrix}$ in degrees/second

ω_x , ω_y and ω_z are the angular velocities of the satellite about the axis X Y and Z

Rodrigues' Rotation Formula to get rotation matrix at time instant t:

- 't' is the time value that ranges from 0 to T (where T = 20.25 seconds).
- Time interval assumed for plotting the graph, dt = 0.05 seconds.
- $\delta_t = (\frac{\delta}{T} * t)$
- $c = \cos \delta_t$; $s = \sin \delta_t$; v = 1 - $\cos \delta_t$

- $\begin{matrix} a_x \\ a_y \\ a_z \end{matrix} = \begin{bmatrix} 0.8031 \\ 0.5867 \\ 0.1028 \end{bmatrix}$

- $R_t = \begin{bmatrix} a_x^2 v + c & a_x a_y v - a_z s & a_x a_z v + a_y s \\ a_x a_y v + a_z s & a_y^2 v + c & a_y a_k v - a_x s \\ a_x a_z v - a_y s & a_y a_z v + a_x s & a_z^2 v + c \end{bmatrix}$

Angles about X Y and Z axis:

- Angle about X axis, Ψ_t = arctan2 ($R_t(3,2)$, $R_t(3,3)$)
- Angle about Y axis, θ_t = arctan2 ($-R_t(3,1)$, $\sqrt{R_t^2(3,2) + R_t^2(3,3)}$)
- Angle about Z axis, ϕ_t = arctan2 ($R_t(2,1)$, $R_t(1,1)$)

Steps followed:

i. From the given values of ψ, θ and ϕ, rotational matrix R is calculated.

ii. Using the rotational matrix R, the satellite rotation is expressed in terms of a single axis and a single rotation (axis-angle representation)

iii. For the above step, the rotation (δ) about a newly defined axis 'a' is calculated along with the x y z components of the unit vector of the axis 'a'.

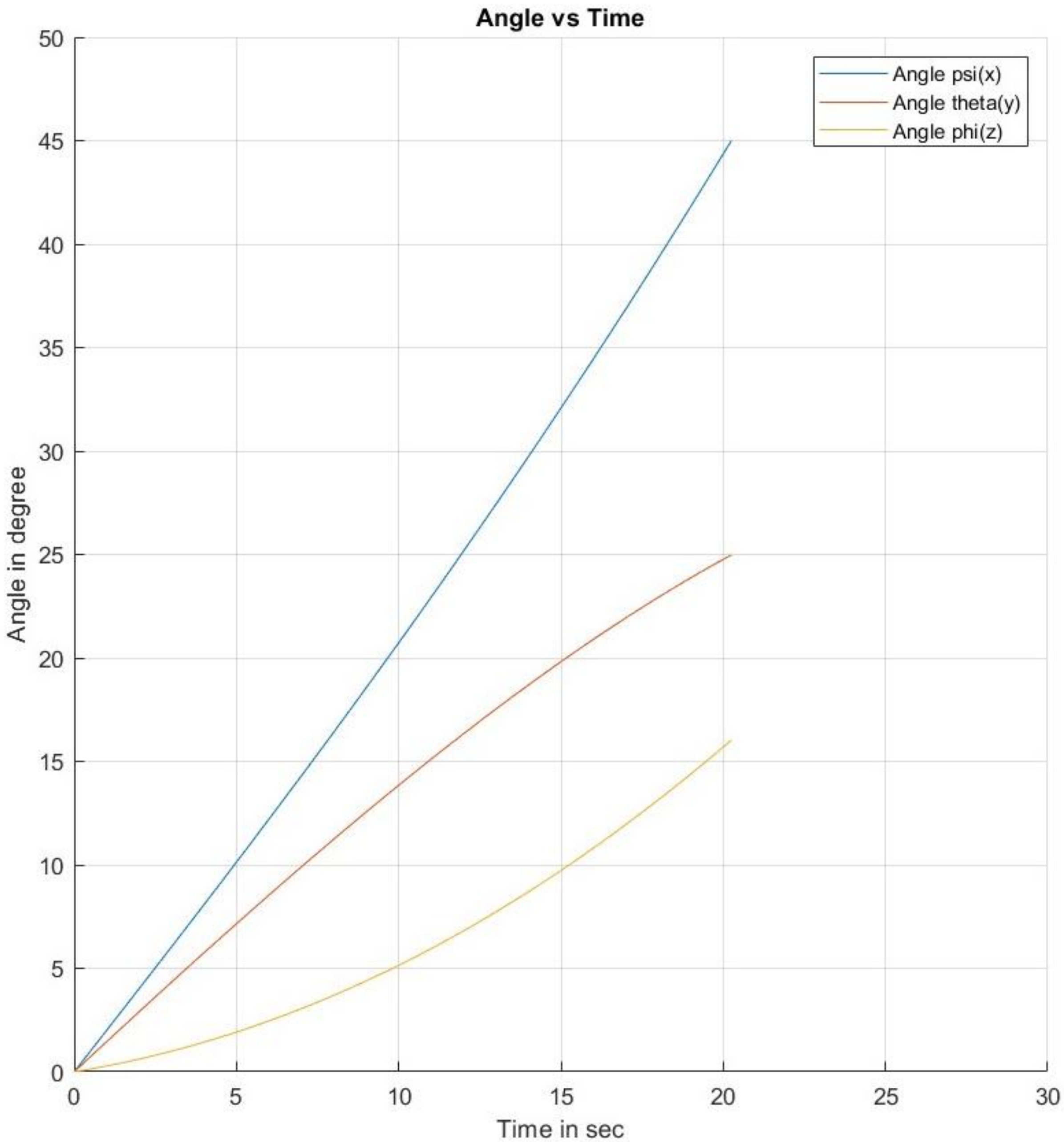
iv. Using the angular velocity limits (about x y and z axis) provided in the question, the shortest time taken by the satellite to reach its desired orientation is calculated.

v. The angular velocity profile, i.e. the angular velocity about the x y z axis (for δ rotation about 'a') is established for the problem.

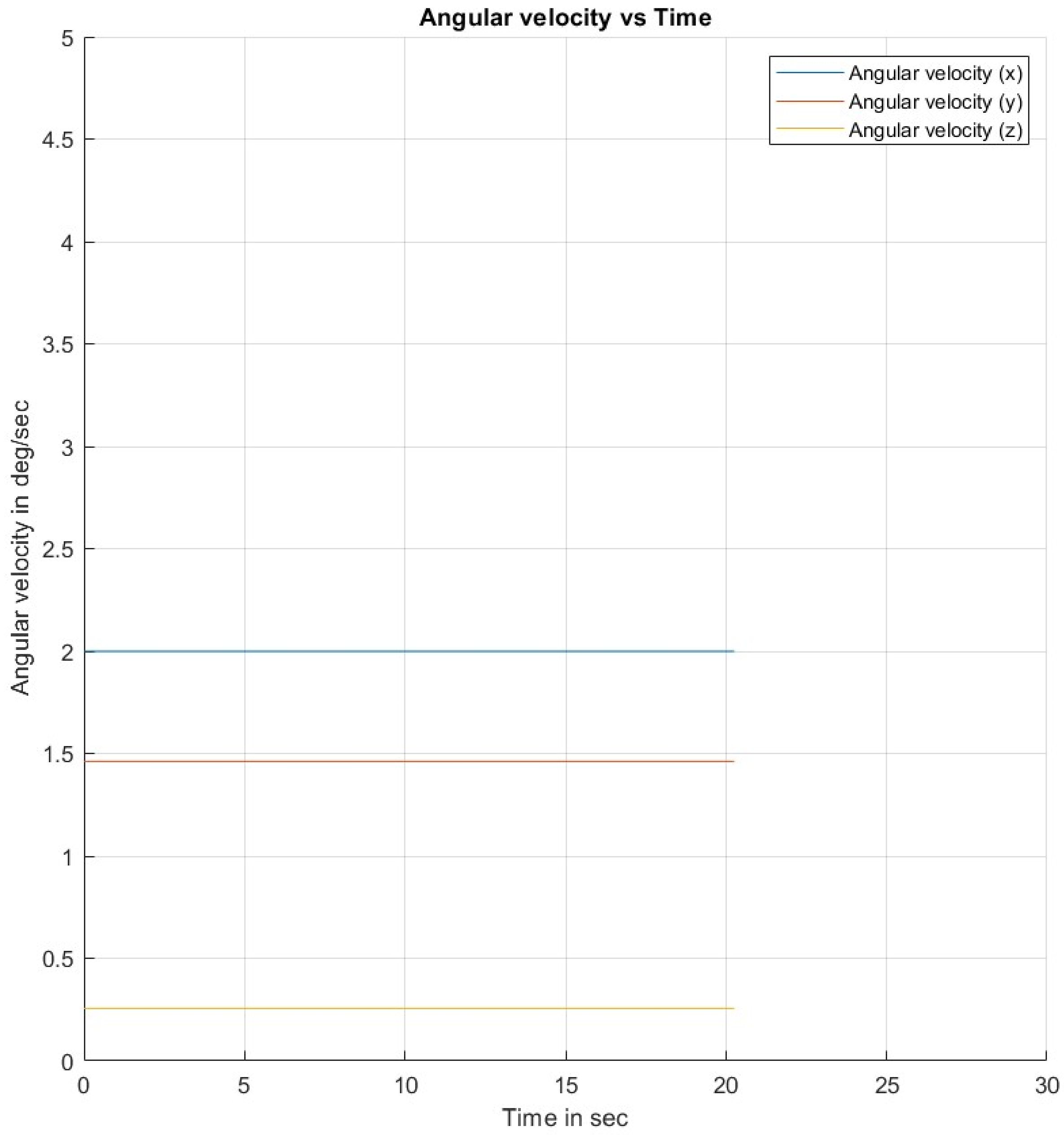
vi. Rodrigues' Rotation Formula is used to get a time dependent rotational matrix, required for plotting.

vii. From the rotational matrix, the angles rotated about x, y and z axis (ψ θ and ϕ respectively) at time instant 't' are calculated.

viii. The angles and the angular velocity are plotted against time t to get the trajectory.



Angle (ψ(x) , θ(y) and ϕ(z)) vs Time plot



Angular Velocity (ω_x, ω_y and ω_z) vs Time plot

```matlab

T=20.25;        % Shortest time
VT=[];          % Time vector
VPsi=[];        % Psi (x-angle) vector
Vwx=[];         % Angular velocity vector (about x-axis)
VTheta=[];      % Theta (y-angle) vector
Vwy=[];         % Angular velocity vector (about y-axis)
VPhi=[];        % Phi (z-angle) vector
Vwz=[];         % Angular velocity vector (about y-axis)
i=0;
dt=0.05;        % Time interval for plotting graph

for t=0:dt:T
    wx=2;          % Angular velocity of rotation about x axis
    wy=1.461;      % Angular velocity of rotation about x axis
    wz=0.2560;     % Angular velocity of rotation about x axis

    delta=deg2rad(2.49*t);  % Angular velocity of rotation about axis a
    ax=0.8031;     % x component of unit vector a
    ay=0.5867;     % y component of unit vector a
    az=0.1028;     % z component of unit vector a
    c=cos(delta);
    s=sin(delta);
    v=1-c;

    %  Rodrigues' Rotation Formula to get rotation matrix

    R1 = [[(((ax)^2)*v)+c,(ax*ay*v)-(az*s),(ax*az*v)+(ay*s)];[(ax*ay*v)+(az*s),(((ay)^2)*v)+c,(az*ay*v)-(ax*s)];[(ax*az*v)-(ay*s),(ay*az*v)+(ax*s),(((az)^2)

    psi_x=(atan2(R1(3,2),R1(3,3)));                          % Rotation about x-axis after time t
    theta_y=(atan2(-R1(3,1),(sqrt((R1(3,2)^2)+(R1(3,3)^2)))));     % Rotation about y-axis after time t
    phi_z=(atan2(R1(2,1),R1(1,1)));                          % Rotation about z-axis after time t


    VT(i+1)=t;
    VPsi(i+1)=rad2deg(psi_x);
    VTheta(i+1)=rad2deg(theta_y);                             % adding the values to the vector for plotting
    VPhi(i+1)=rad2deg(phi_z);
    Vwx(i+1)=wx;
    Vwy(i+1)=wy;
    Vwz(i+1)=wz;

    i=i+1;

end

a = tiledlayout(1,2);
nexttile
hold on

plot(VT,VPsi,'DisplayName','Angle ψ (x)')
plot(VT,VTheta,'DisplayName','Angle θ (y)')
plot(VT,VPhi,'DisplayName','Angle φ (z)')
title('Angle vs Time')
legend('Angle psi(x)', 'Angle theta(y)', 'Angle phi(z)')          % Angle vs Time plot
axis([0 30 0 50])
xlabel('Time in sec')
ylabel('Angle in degree')
grid on

hold off


nexttile

hold on

plot(VT,Vwx)
plot(VT,Vwy)
plot(VT,Vwz)
title('Angular velocity vs Time')
legend('Angular velocity (x)', 'Angular velocity (y)', 'Angular velocity (z)')          % Angular Velocity vs Time plot
axis([0 30 0 5])
xlabel('Time in sec')
ylabel('Angular velocity in deg/sec')
grid on

hold off

```

Matlab script for plotting the trajectory