AEROSPACE ENGINEERING
DEPT. FACULTY OF
ENGINEERING CAIRO
UNIVERSITY
GIZA, 12613, EGYPT

قسم هندسة الطيران و الفضاء
كلية الهندسة
جامعة القاهرة
الجيزة – جمهورية مصر العربية

AER408
Aerospace Guidance & Control Systems

Task (2)
Airplane Simulator Part I

Before you go, please note the following:

- You do not have to use the equations presented in this document, consult the reference you prefer, equations presented here are for illustration
- All angles are in radians through this course !!

Task statement:

"Write a code that solves the Rigid body dynamics (RBD) equations for given values of forces & moments, using the "Raunge-Kutta" 4th order method"

Develop your code from scratch, and use the following data and initial conditions to solve the equations (all variables are in SI units)

$$t_{final} = 15\ sec$$

$$Forces = [10\ 5\ 9]\ N$$

$$Moments = [10\ 20\ 5]\ Nm$$

$$mass = 15\ kg$$

$$I = \begin{bmatrix} 1 & -2 & -1 \\ -2 & 5 & -3 \\ -1 & -3 & 0.1 \end{bmatrix}$$

Initial conditions

$$(u, v, w, p, q, r, \emptyset, \theta, \Psi, x, y, z) = \left(10,\quad 2,\quad 0,\quad 2 * \frac{\pi}{180},\quad 1 * \frac{\pi}{180},\quad 0 * \frac{\pi}{180},\quad 20 * \frac{\pi}{180},\quad 15 * \frac{\pi}{180},\quad 30 * \frac{\pi}{180},\quad 2,\quad 4,\quad 7\right)$$

Notes & Hints

RBD Equations in vector form:

Kinetics

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m\left(\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix}\right)$$

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = I\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I\begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Kinematics

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = J\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & (\sin\phi\tan\theta) & (\cos\phi\tan\theta) \\ 0 & \cos\phi & -\sin\phi \\ 0 & \dfrac{\sin\phi}{\cos\theta} & \dfrac{\cos\phi}{\cos\theta} \end{bmatrix}\begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

$$\begin{bmatrix} \dot{x}_E \\ \dot{y}_E \\ \dot{z}_E \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix}\begin{bmatrix} u_b \\ v_b \\ w_b \end{bmatrix};$$

Where inertia matrix is defined as follows

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix}$$

RBD Equations in scalar form:

$$X - mg\sin\theta = m(\dot{u}^E + qw^E - rv^E) \qquad (a)$$

$$Y + mg\cos\theta\sin\phi = m(\dot{v}^E + ru^E - pw^E) \qquad (b)$$

$$Z + mg\cos\theta\cos\phi = m(\dot{w}^E + pv^E - qu^E) \qquad (c)$$

$$L = I_x\dot{p} - I_{zx}\dot{r} + qr(I_z - I_y) - I_{zx}pq + qh'_z - rh'_y \qquad (a)$$

$$M = I_y\dot{q} + rp(I_x - I_z) + I_{zx}(p^2 - r^2) + rh'_x - ph'_z. \qquad (b)$$

$$N = I_z\dot{r} - I_{zx}\dot{p} + pq(I_y - I_x) + I_{zx}qr + ph'_y - qh'_x \qquad (c)$$

$$p = \dot{\phi} - \dot{\psi}\sin\theta \qquad (a)$$

$$q = \dot{\theta}\cos\phi + \dot{\psi}\cos\theta\sin\phi \qquad (b)$$

$$r = \dot{\psi}\cos\theta\cos\phi - \dot{\theta}\sin\phi \qquad (c)$$

$$\dot{\phi} = p + (q\sin\phi + r\cos\phi)\tan\theta \qquad (d)$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \qquad (e)$$

$$\dot{\psi} = (q\sin\phi + r\cos\phi)\sec\theta \qquad (f)$$

$$\dot{x}_E = u^E\cos\theta\cos\psi + v^E(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi) + w^E(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) \qquad (a)$$

$$\dot{y}_E = u^E\cos\theta\sin\psi + v^E(\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi) + w^E(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi) \qquad (b)$$

$$\dot{z}_E = -u^E\sin\theta + v^E\sin\phi\cos\theta + w^E\cos\phi\cos\theta \qquad (c)$$

The "Runge-Kutta" method for simultaneous differential equations:

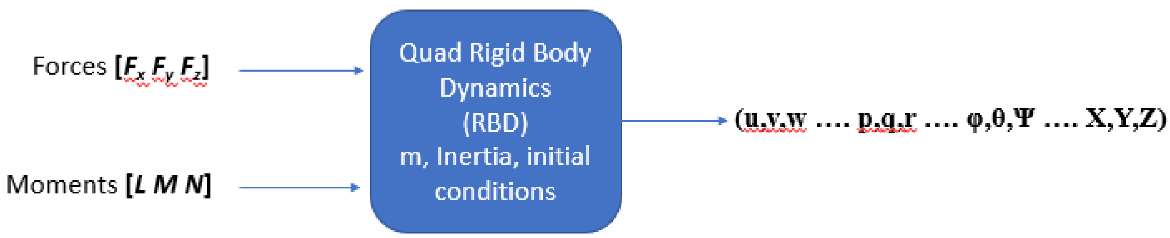
$$
\begin{aligned}
y_{n+1} &= y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\
u_{n-1} &= u_n + \frac{1}{6}(q_1 + 2q_2 + 2q_3 + q_4) \\
k_1 &= hg(x_n, y_n, u_n) \\
q_1 &= hf(x_n, y_n, u_n) \\
k_2 &= hg\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1, u_n + \frac{1}{2}q_1\right) \\
q_2 &= hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1, u_n + \frac{1}{2}q_1\right) \\
k_3 &= hg\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2, u_n + \frac{1}{2}q_2\right) \\
q_3 &= hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2, u_n + \frac{1}{2}q_2\right) \\
k_4 &= hg(x_n + h, y_n + k_3, u_n + q_3) \\
q_4 &= hf(x_n + h, y_n + k_3, u_n + q_3).
\end{aligned}
$$

Input to the RBD solver code:

- Forces and moments values [F_x, F_y, F_z, L, M, N]
- Initial conditions of all the 12 states (u_0, v_0, w_0, …)

Output from the RBD solver code:

- The new values of the 12 states (u, v, w, …) at each time step



Constants to be defined:

- Mass & Inertia matrix (m, *I*)
- Time interval parameters (t_{final}, time step)

Note: after building your code, you can validate your results using the "6 DOF (Euler angles)" function in Simulink (BONUS), or just check the results using the same inputs with your colleagues

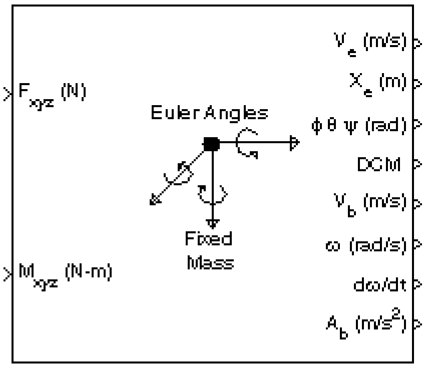
6DOF (Euler Angles)

Implement Euler angle representation of six-degrees-of-freedom equations of motion

Library

Equations of Motion/6DOF

Description



Hint: It is very useful to learn how to interact with a SIMULINK model from MATLAB code, you will need this to send parameters to the SIMULINK model and receive result from it for plotting and performing further analysis with a MATLAB code. Check this nice video to get this skill https://www.youtube.com/watch?v=sF_sjFqNFUk