Clase 3: Matrices de rotación y cuaterniones

Fundamentos de ingeniería de control de nanosatélites

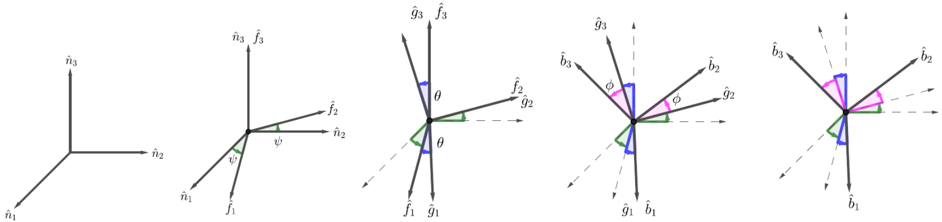
Ing. Franklin Ticona

Universidad Católica Boliviana San Pablo sede La Paz

Junio 2023

Matrices de rotación y ángulos de Euler

Orden de rotación 3-2-1 (Yaw-Pitch-Roll)



$$\{\hat{b}\} = \begin{bmatrix} \cos(\phi)\cos(\psi) - \cos(\theta)\sin(\phi)\sin(\psi) & \cos(\psi)\sin(\phi) + \cos(\theta)\cos(\phi)\sin(\psi) & \sin(\theta)\sin(\psi) \\ -\cos(\theta)\cos(\psi)\sin(\phi) - \cos(\phi)\sin(\psi) & \cos(\theta)\cos(\phi)\cos(\psi) - \sin(\phi)\sin(\psi) & \cos(\psi)\sin(\theta) \\ \sin(\theta)\sin(\phi) & -\cos(\phi)\sin(\theta) & \cos(\theta) \end{bmatrix} \{\hat{n}\}$$

Ecuación diferencial de attitude

Si empleamos el orden de rotación 3-2-1, obtenemos la siguiente ecuación diferencial para la posición angular del satélite (attitude):

$$\begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \frac{1}{\cos\theta} \begin{bmatrix} 0 & \sin(\phi) & \cos(\phi) \\ 0 & \cos(\theta)\cos(\phi) & -\cos(\theta)\sin(\phi) \\ \cos(\theta) & \sin(\theta)\sin(\phi) & \sin(\theta)\cos(\phi) \end{bmatrix}^{\mathcal{B}} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = [B(\psi, \theta, \phi)]^{\mathcal{B}} \omega_{B/N}$$

Si programamos un integrador, podemos hallar la attitude del satélite a medida que cambia $\omega_1, \omega_2, \omega_3$. Link de ejemplo en Python

Cuaterniones

Un cuaternión es un vector $\boldsymbol{q} \in \mathbf{R}^4$, que se encuentra sobre una hiperesfera unitaria. Por lo que satisface:

$$|\boldsymbol{q}| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1$$

O también:

$$|\boldsymbol{q}|^2 = q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$$

Un cuaternión no tiene singularidades

Ecuación diferencial empleando cuaterniones

$$\dot{\boldsymbol{q}} = \frac{1}{2}\begin{bmatrix} 0 & -\omega_1 & 0 & 0 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} \boldsymbol{q}$$

O también:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & -\omega_1 & 0 & 0 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

Notar que el cuaternión $\boldsymbol{q}$ está asociado a una rotación, por ejemplo $[BN]$ y en particular, el vector velocidad angular ${}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^T$.

Matrices de rotación y cuaterniones

Si el cuaternión está asociado a la rotación de $\mathcal{N}$ a $\mathcal{B}$:

$$[BN] = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0^2 + q_1^2 - q_2^2 - q_3^2 \end{bmatrix}$$

También, existe la operación de multiplicación de cuaterniones:

$$[FN(\boldsymbol{q})] = [FB(\boldsymbol{q}'')][BN(\boldsymbol{q}')]$$

Por dos dulces ¿qué significa **la inversa** de un cuaternión?.

Problemas

Problema 1.- Dado el programa base en simulink, identificar un orden de rotación adecuado, convertir el cuaternión a ángulos de rotación y con uno de esos ángulos, aplicar el control PID para lograr que el cubo rojo se balancee sobre su arista.

Problema 2.- Dado el código base en simulink, completar el sistema de control para que el cubo (en gravedad cero) apunte al set-point establecido.