Obiettivo: Calcolare il vettore **u** che rappresenta il controllo.

Input:

- Matrice A_BN "actual", cioè quella in uscita dalla attitude determination (**non** quella in uscita dalle Euler Equations!)
- Matrice A_BN "desidered"
- Vettore velocità angolare del satellite, dal giroscopio

Metodo 1: "quaternioni" (pagina 56 dispensa)

Procedimento:

1. **Trasformare le 2 matrici in quaternioni**, ottenendo quindi 2 vettori di quaternion (**q**$_c$ e **q**, dove il primo rappresenta il quaternione desiderato, mentre il secondo rappresenta il quaternione "actual").
2. **Costruire la seguente matrice**:

$$\begin{bmatrix} q_{4c} & q_{3c} & -q_{2c} & -q_{1c} \\ -q_{3c} & q_{4c} & q_{1c} & -q_{2c} \\ q_{2c} & -q_{1c} & q_{4c} & -q_{3c} \\ q_{1c} & q_{2c} & q_{3c} & q_{4c} \end{bmatrix}$$

3. **Costruire il quaternione errore:**

$$\begin{bmatrix} q_{1e} \\ q_{2e} \\ q_{3e} \\ q_{4e} \end{bmatrix} = \begin{bmatrix} q_{4c} & q_{3c} & -q_{2c} & -q_{1c} \\ -q_{3c} & q_{4c} & q_{1c} & -q_{2c} \\ q_{2c} & -q_{1c} & q_{4c} & -q_{3c} \\ q_{1c} & q_{2c} & q_{3c} & q_{4c} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}$$

4. **Calcolare il controllo come:**

$$\underline{u} = -k_1 \underline{\omega} + k_2 \frac{\partial H(q_{4e})}{\partial q_{4e}} \underline{q}_e$$

Come scegliere H(q_{4e})?

$$H(q_{4e}) = 1 - q_{4e}^2$$
$$H(q_{4e}) = 1 - sgn(q_{4e}(0))q_{4e}$$
$$H(q_{4e}) = arccos^2 q_{4e}$$

Metodo 2: "eigenaxis rotation" (pagina 58 dispensa)

I primi 3) passi sono uguali a sopra

4. **Calcolare il controllo come:**

$$\underset{\sim}{u} = \underset{\sim}{\omega} \times J\underset{\sim}{\omega} - k_1 J\underset{\sim}{\omega} + k_2 J \frac{\partial H(q_{4e})}{\partial q_{4e}} \underset{\sim}{q_e}$$

Dove J è la matrice di inerzia

5. **Ottimizzare la scelta delle costanti** ponendo:

$$k_1^2 = 2k_2$$

Metodo 3: "*reduced-attitude control problem*" (pagina 61 dispensa)

In some applications it may only be required to point in a prescribed direction. We define a desired pointing vector by $\Gamma_d$which is expressedIn such cases full 3-axis stabilization is not required and it may not be necessary to define the entire reference attitude on the pointing direction. Furthermore, in such cases it may not be necessary to determine the entire attitude but only a pointing vector. Define an orthonormal frame **N** such that the desired $\Gamma_d = [1 \quad 0 \quad 0]_N^T$. The current pointing vector can be expressed in the body frame as $\underset{\sim}{\Gamma}$then $\underset{\sim}{\Gamma} = A_{B/N}\Gamma_d$ then:

$$\dot{\underset{\sim}{\Gamma}} = \dot{A}_{B/N}\Gamma_d = -[\omega]^{\wedge} A_{B/N}\Gamma_d = -[\omega]^{\wedge}\underset{\sim}{\Gamma} = \underset{\sim}{\Gamma} \times \underset{\sim}{\omega}$$

Then define the error angle between the two vectors by:

$$\Gamma_d^T \Gamma = \cos\theta_e$$

We define the Lyapunov function

$$\boldsymbol{V} = \frac{\mathbf{1}}{\mathbf{2}}\langle \underset{\sim}{\boldsymbol{\omega}}, \boldsymbol{J}\underset{\sim}{\boldsymbol{\omega}}\rangle + \boldsymbol{k_2}(\mathbf{1} - \boldsymbol{\Gamma}_d^T\boldsymbol{\Gamma})$$

Which satisfies the first two requirements for a Lyapunov function. Then differentiating

$$\dot{\boldsymbol{V}} = \langle \underset{\sim}{\boldsymbol{\omega}}, \underset{\sim}{\boldsymbol{u}}\rangle - \boldsymbol{k_2}(\boldsymbol{\Gamma}_d^T\dot{\underset{\sim}{\boldsymbol{\Gamma}}}) = \langle \underset{\sim}{\boldsymbol{\omega}}, \underset{\sim}{\boldsymbol{u}}\rangle - \boldsymbol{k_2}\boldsymbol{\Gamma}_d^T(\underset{\sim}{\boldsymbol{\Gamma}} \times \underset{\sim}{\boldsymbol{\omega}})$$
$$\dot{\boldsymbol{V}} = \langle \underset{\sim}{\boldsymbol{\omega}}, \underset{\sim}{\boldsymbol{u}}\rangle - \boldsymbol{k_2}\underset{\sim}{\boldsymbol{\omega}}^T(\boldsymbol{\Gamma}_d \times \underset{\sim}{\boldsymbol{\Gamma}}) = \langle \underset{\sim}{\boldsymbol{\omega}}, \underset{\sim}{\boldsymbol{u}} + \boldsymbol{k_2}(\underset{\sim}{\boldsymbol{\Gamma}} \times \boldsymbol{\Gamma}_d)\rangle$$

Then the control

$$\underset{\sim}{u} = -k_1\underset{\sim}{\omega} - k_2(\Gamma \times \Gamma_d)$$

Asymptotically converges to the desired poiting direction Γ_d.