Task Space Inverse Dynamics

We have a 6-DOF Robot

let $p = \begin{bmatrix} \text{position} \\ \text{orientation} \end{bmatrix} = \begin{bmatrix} P_{os} \\ R \end{bmatrix}$ $\quad \dot{p} = \begin{bmatrix} V \\ \omega \end{bmatrix} = \begin{bmatrix} \dot{P}_{os} \\ \dot{R} \end{bmatrix}$

$$\tilde{p} = p - p_d \qquad \dot{\tilde{p}} = \dot{p} - \dot{p}_d = J\dot{q} - \dot{p}_d$$

$$\ddot{\tilde{p}} = \ddot{p} - \ddot{p}_d = J\ddot{q} + \dot{J}\dot{q}$$

$$\ddot{\tilde{p}} + K_d \dot{\tilde{p}} + K_p \tilde{p} = 0 \quad (1)$$

from Sponge: $\underset{6\times1}{a_q} = \underset{6\times6}{J^{-1}} \cdot \left\{ \underset{6\times1}{\begin{bmatrix} a_x \\ a_\omega \end{bmatrix}} - \underset{6\times6}{\dot{J}} \underset{6\times1}{\dot{q}} \right\}$ – outer-loop control

where $\begin{cases} a_x = \ddot{X} = \ddot{p}os \\ a_\omega = \dot{\omega} \\ \dot{R} = S(\omega) \cdot R \end{cases}$ $\qquad \tilde{S} = \log(R_d \cdot R^T)$ – orientation error

$$\ddot{X} = a_x \Rightarrow \text{equation } 1$$

$$\updownarrow$$

$$\ddot{\tilde{X}} + K_1 \dot{\tilde{X}} + K_0 \tilde{X} = 0$$ – task space tracking error satisfies this equation

$\underset{6\times1}{a_q} = \underset{6\times6}{J^{-1}} \cdot \begin{bmatrix} a_x \\ a_\omega \end{bmatrix} - \underset{6\times6}{\dot{J}} \underset{6\times1}{\dot{q}} \Big\}$ – for joints control.

~~obtained from trajectory~~

$$a_q = J^{-1}\{a_x - \dot{J}\dot{q}\}$$

- $a_x = \ddot{X}^d - K_0(X - X^d) - K_1(\dot{X} - \dot{X}^d)$ (2)

and we can track error: $Err = \ddot{\tilde{X}} + K_1\dot{\tilde{X}} + K_0\tilde{X}$

$X \in R^6$ it means $X - X^d$ is not a simple subtraction!

$$X = \begin{bmatrix} x \\ y \\ z \\ \varphi \end{bmatrix}_{3\times1} \text{ or } R \rightarrow X - X^d = \begin{bmatrix} x - x_d \\ y - y_d \\ z - z_d \\ \tilde{S} \text{ to vector} \end{bmatrix}$$

where $\tilde{S} = \log(R_d R^T)$

$$\dot{X} - \dot{X}^d = \begin{bmatrix} \dot{x} - \dot{x}_d \\ \dot{y} - \dot{y}_d \\ \dot{z} - \dot{z}_d \\ \dot{\tilde{S}} \end{bmatrix}$$

where $\dot{\tilde{S}} = \log(\dot{R}_d \dot{R}^T)$

$\dot{R}_d = S(\omega_d) \cdot R_d$

$\dot{R} = S(\omega) \cdot R$

$u = M a_q + C\dot{q} + g$ – substitute to code

I generated a straight line with direction as $q = [1 0 0 0]$, $\dot{X}_d = \vec{0}$; $\ddot{X}_d = \vec{0}$ (zero vectors for simplicity)

Everything done!

See the code and logs!