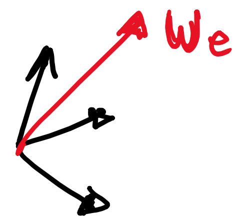
H1- Orientation Control in robot manipulators

- until now we focused on Cartesian control (3D)

we want now to incorporate tracking of the orientation of a frame (e.g. attached at the end-effector).

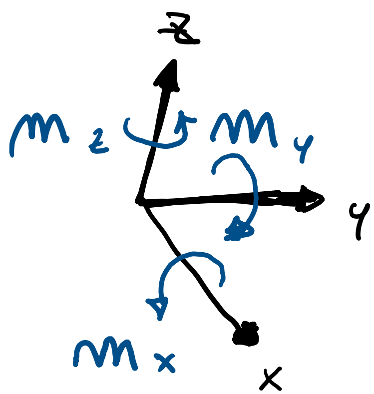
ORIENTATION JACOBIAN



End-effector FRAME

$$\omega_e \in \mathbb{R}^3 \longleftarrow \boxed{J_o} \longleftarrow \dot q \in \mathbb{R}^n$$

↳ NOTE: This is the angular part of the geometric jacobian



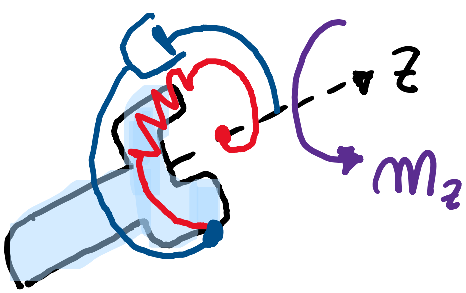
$$m = [m_x, m_y, m_z] \in \mathbb{R}^3 \longrightarrow \boxed{J_o^T} \longrightarrow \tau \in \mathbb{R}^n$$

PD FOR ORIENTATION CONTROL

$$\boxed{u = J_o^T \left(K_o \, e_o + D_o \underbrace{(\omega^d - J_o \dot{q})}_{\dot{e}_o \text{ already in euclidean space}} \right)}$$

e_o: orientation error

- This is equivalent to have torsional springs and dampers in the 3 directions



$$m_z = K_{o,zz} \, e_{o_z} \in \mathbb{R}$$

$$m = K_o \, e_o \in \mathbb{R}^3$$

$$\hookrightarrow K_o = \begin{bmatrix} K_{o\,xx} & 0 & 0 \\ 0 & K_{o\,yy} & 0 \\ 0 & 0 & K_{o\,zz} \end{bmatrix} \in \mathbb{R}^{3\times3}$$

- To control the orientation we need to define an error variable that indicates the deviation between the actual and the desired orientation

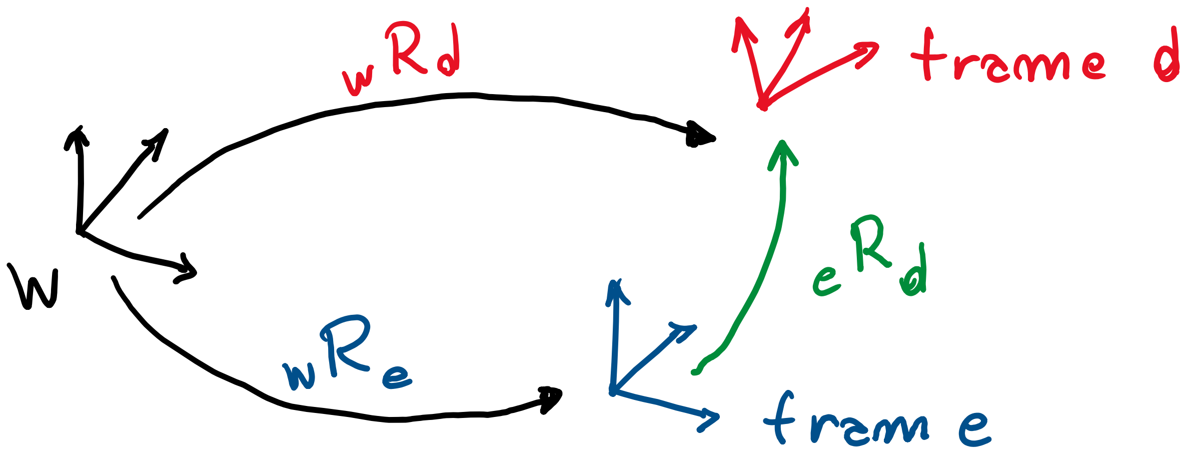
- The position error e_p is simply defined as

$$e_p = p^d - p$$

- for the orientation error e_o difficulty arise due to the fact that we need to consider the algebra of the rotation group

- its expression depends on the particular representation that we chose for the orientation
  - rotation matrix
  - angle - axis
  - euler angles
  - quaternions

- let us consider the rotation matrix representing the desired ${}_wR_d$ and the actual ${}_wR_e$ end-effector orientations



- ${}_wR_e$ is computed from q through direct kinematics

- The relative orientation can be obtained by composing subsequent rotations

$${}_{w}R_{d} = {}_{w}R_{e}\ {}_{e}R_{d}$$ left-multiply by ${}_{w}R_{e}^{T}$

$$\boxed{{}_{e}R_{d} = {}_{w}R_{e}^{T}\ {}_{w}R_{d}}$$ orientation of frame d with respect to frame e

- Note the orientation error can be defined in 2 ways according to which coordinate frame is used as reference (we use frame e)

⊖ using directly ${}_{e}R_{d}$ for orientation control is limited, due to the difficulty of handling the 9 elements of the matrix

⇒ better to express ${}_{e}R_{d}$ in terms of angle-axis

ORIENTATION ERROR WITH ANGLE-AXIS

- let's define the orientation error e_o as:

$$\boxed{{}_e e_o = \hat{r} \sin(\Delta\theta)}$$

[LUH 1980]

- using this definition it is possible to proof that we can link e_o to the elements of ${}_e R_d = R$

$$\boxed{{}_e e_o = \frac{1}{2} \begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix}}$$

- otherwise if we define it as $e_o = \hat{r}\Delta\theta$ we use the formules to compute separately $\hat{r}, \Delta\theta$ of Lecture E02

- note that e_o is expressed in the frame e and therefore it should be mapped in the same frame of J (i.e w) before using for the computation of the restoring moment m:

$$\boxed{{}_w e_o = {}_w R_e \, {}_e e_o}$$

ORIENTATION ERROR WITH EULER ANGLES

- we can define the orientation error analogously to the position error:

$$e_o = \Phi_d - \Phi_e(q)$$

- where ϕ_d and ϕ_e can be extracted from the rotation matrix ${}_wR_d$ and ${}_eR_d$
- if we compute the moment $m_\phi = K_o e_o$ will be defined in a frame with non-orthogonal axes
- let us use the analytic jacobian J_A instead of the geometric one to map to torques:

$$\dot{q} \rightarrow \boxed{J_A} \rightarrow \begin{bmatrix} \dot{p} \\ \dot{\Phi} \end{bmatrix} \qquad \dot{q} \rightarrow \boxed{J_{A,O}} \rightarrow \dot{\Phi} \qquad m \rightarrow \boxed{J_{A,O}^T} \rightarrow \tau$$

$$\boxed{u = J_{A,O}^T K_O e_O + J_O^T D_O (\omega^d - \omega)}$$

or if we want to use $\dot{\phi}$ for damping

$$\boxed{u = J_{A,O}^T \left[K_O e_O + D_O (\dot{\phi}^d - \dot{\phi}) \right]}$$

$\hookrightarrow J_{A,O} = T_\omega(\phi) J_O \rightarrow$ angular part of geometric jacobian

⊖ There are representation singularities in J_A due to $T_\omega(\phi)$

⊖ Need to derive ϕ_e from ${}_wR_e(q)$ with atan2 $\rightarrow$ (complex) because we have no closed form expression from q to ϕ_e

ORIENTATION ERROR WITH QUATERNIONS

① compute the quaternions Q_e and Q_d associated to ${}_wR_e$ and ${}_wR_d$

$$Q = \frac{1}{2}\begin{bmatrix} A \\ (R_{23} - R_{32})/A \\ (R_{31} - R_{13})/A \\ (R_{12} - R_{21})/A \end{bmatrix}, \quad A = (1 + R_{11} + R_{22} + R_{33})^{1/2}$$

② composition of quaternions is analogous to composition of rotation matrix and is used the quaternion multiplication $\otimes$ formula

$$R_3 = R_1 \cdot R_2$$

⇩

$$Q_3 = Q_1 \otimes Q_2 = \begin{bmatrix} \eta_1\eta_2 - \varepsilon_1^T \varepsilon_2 \\ \eta_1 \varepsilon_2 + \eta_2 \varepsilon_1 + \varepsilon_1 \times \varepsilon_2 \end{bmatrix}$$

in our case the quaternion that represents the relative orientation of frame d with respect to frame e is:

$${}_eR_d = {}_wR_e^T \, {}_wR_d$$

$$\Delta Q = \underset{\substack{\text{quaternion}\\\text{inverse}}}{\underbrace{\bar{Q}_e}} \otimes \underset{\text{desired}}{\underbrace{Q_d}} = \begin{bmatrix} \eta_e \eta_d + \varepsilon_e^T \varepsilon_d \\ \eta_e \varepsilon_d - \eta_d \varepsilon_e - \varepsilon_e \times \varepsilon_d \end{bmatrix}$$

$$\bar{Q}_e = \begin{bmatrix} \eta_e \\ -\varepsilon_e \end{bmatrix} \qquad Q_d = \begin{bmatrix} \eta_d \\ \varepsilon_d \end{bmatrix}$$

${}_e e_o$: vector part of ΔQ

- remember that e_o is expressed in e frame and need to be mapped into w frame

⊕ no singularity

⊖ requires intermediate computation of Q_e from ${}_wR_e(q)$ because there is no direct mapping from q to Q_e

References:

- Resolved-acceleration control of mechanical manipulators, J. Luh 1980.
- Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors, James Diebel, 2006.
- Operational Space Control: A Theoretical and Empirical Comparison, Jun Nakanishi, 2008.
- Closed-Loop Manipulator Control Using Quaternion Feedback, J. Yuan, 1988.
- Quaternion kinematics for the error-state Kalman filter, Joan Solà, 2017.