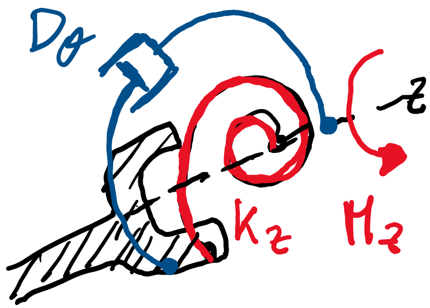
Orientation Control

Mapping to torques

Parametrizations of orientation

- **Angle-axis**
- **Rotation Matrix**
- **Euler-angles**
- **Unit Quaternions**



To control end-effector orientation we need torsional springs and dampers

	LINEAR	ROTATIONAL	
1 AXIS	$F_z = K_z \Delta x_z$	$M_z = K_z \Delta\theta$	$\in \mathbb{R}$
3 AXES ?	$\vec{F} = K \vec{\Delta X}$	$\vec{M} = K_\theta \vec{e}_o + D_\theta \dot{e}_o$	$\in \mathbb{R}^3$

MAP TO JOINT SPACE

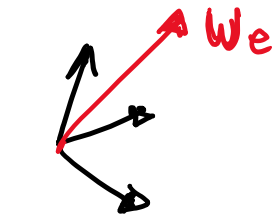
$$\tau = J_{LIN}^T \vec{F}$$

LINEAR JACOBIAN

$$\tau = J_o^T M$$

ORIENTATION JACOBIAN

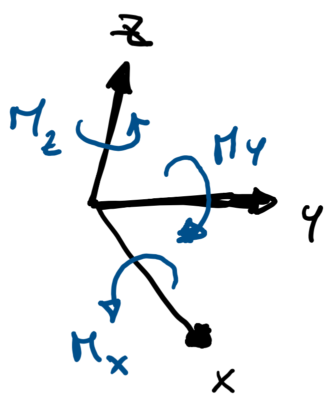
ORIENTATION JACOBIAN



End-effector FRAME

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

NOTE: This is the geometric Jacobian (analitic Jacobian is mapping to Euler angles rates)



$$M = [M_x, M_y, M_z] \in \mathbb{R}^3 \longrightarrow \boxed{J_o^T} \longrightarrow \tau \in \mathbb{R}^n$$

"TORSIONAL" PD

M = virtual moment at end-effector

$$\tau = J_o^T \overbrace{\left(-K_\theta\, e_o + D_\theta\, \underbrace{(\omega^d - \omega)}_{\dot{e}_o}\right)}^{M}$$

orientation error

$\dot{e}_o$ already in EUCLIDEAN space

drives orientation error to zero

- How to compute e_o? depends on the representation of the orientation

Orientation Control

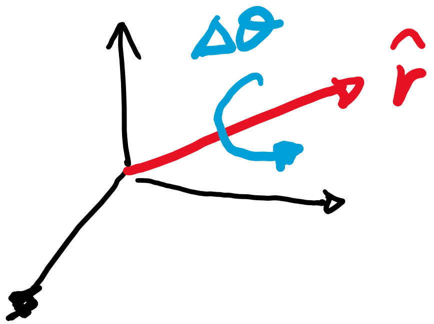
Mapping to torques

Parametrizations of orientation

- Angle-axis
- Rotation Matrix
- Euler-angles
- Unit Quaternions

ANGLE - AXIS REPRESENTATION

- 4 params - non minimal representation



$\hat{r}$: axis about which rotation is made

$\Delta\theta$: magnitude of rotation

$$\boxed{\text{ROTATION VECTOR: } \varphi = \Delta\theta\,\hat{r} \in R^3}$$

NOTE: $\vec{\varphi}_1 + \vec{\varphi}_2 \neq \vec{\varphi}_2 + \vec{\varphi}_1$ Rotations are not commutative!

TO MAP TO ROTATION MATRIX $\Rightarrow$ Rodriguez Formula

$$\boxed{R(\Delta\theta, \hat{r}) = \cos\Delta\theta\, I + (1-\cos\theta)\,\hat{r}\hat{r}^T - \sin\theta\,[\hat{r}]_x}$$

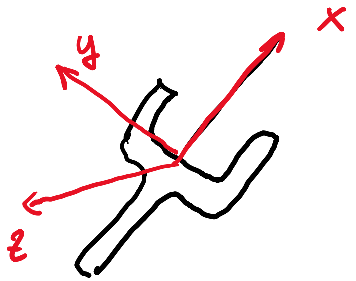
Orientation Control

Mapping to torques

Parametrizations of orientation

- Angle-axis
- Rotation Matrix
- Euler-angles
- Unit Quaternions

ROTATION MATRIX

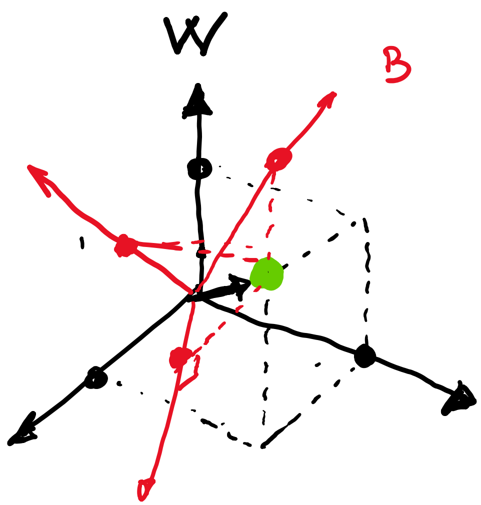


2 INTERPRETATIONS OF A ROTATION MATRIX

- Represents the orientation of a frame

$$R = [\, x \mid y \mid z \,]$$

- change of coordinates : ${}_B R_W$ is a linear operator that maps a vector from a fixed frame W to a rotated frame B

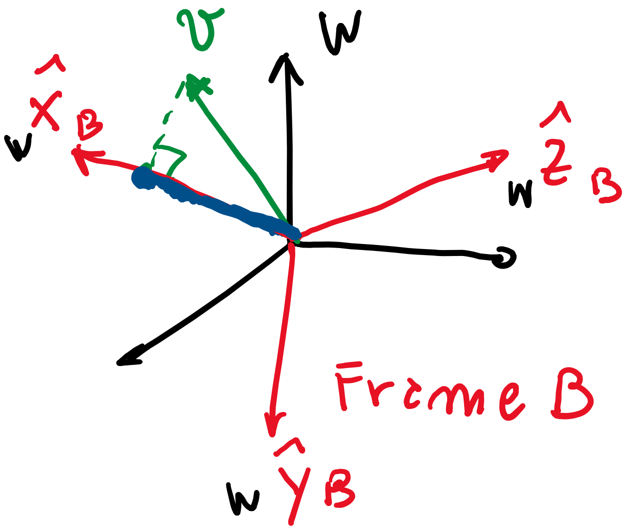


$$\begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} = {}_B R_W \begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

$${}_B v \qquad\qquad {}_W v$$

${}_B R_W$

① The rows are the axis of frame B expressed in the starting frame W.



projection of ${}_W v$ along axis ${}_W\hat{x}_B$

$${}_B v = {}_B R_W \; {}_W v = \begin{bmatrix} {}_W\hat{x}_B^{T} \\ \hline {}_W\hat{y}_B^{T} \\ \hline {}_W\hat{z}_B^{T} \end{bmatrix} \underset{{}_W v}{\begin{bmatrix} \quad \\ \quad \\ \quad \end{bmatrix}} = \begin{bmatrix} {}_W\hat{x}_B^{T} \; {}_W v \\ \quad \\ \quad \end{bmatrix}$$

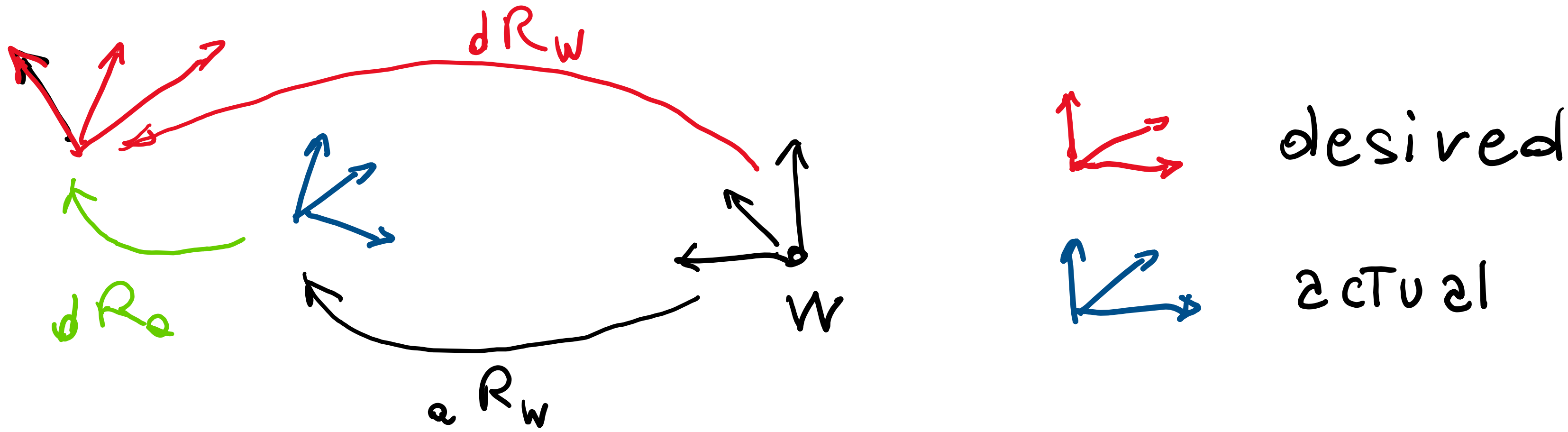
$_W R_B$ ② columns are the axes of frame B expressed in frame W

$$\left[\, {}_W\hat{x}_B \;\middle|\; {}_W\hat{y}_B \;\middle|\; {}_W\hat{z}_B \,\right]$$

$$ {}_W v = {}_W\hat{x}_B \cdot ({}_B v_x) + {}_W\hat{y}_B \cdot ({}_B v_y) + {}_W\hat{z}_B \cdot ({}_B v_z) $$

components of v in the B frame

ORIENTATION ERROR WITH ROT. MATRIX



$$_dR_w = {_dR_a}\, {_aR_w} \qquad \text{right mult for } {_aR_w}^T$$

$$_dR_w\, {_aR_w}^T = {_dR_a}\, \underbrace{{_aR_w}\, {_aR_w}^T}_{I} \Rightarrow \boxed{{_dR_a} = {_dR_w}\, {_aR_w}^T}$$

$\Rightarrow$ map to angle - axis

Rotation matrix are:

⊕ Non minimal representation for orientation (9 parameters)

⊖ orthonormal conditions (6) $R = [u \; v \; w]$

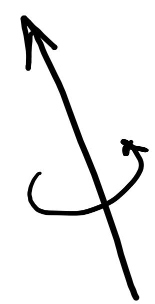
$$u^T u = 1 \quad v^T v = 1 \quad w^T w = 1$$
$$u^T v = 0 \quad v^T w = 0 \quad u^T w = 0$$
$$\approx \; R R^T = I \; , \; \det(R) = 1$$

u, v, w are vectors of orthonormal frame

ORIENTATION ERROR WITH ROT. MATRIX

$$dR_e = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \Longrightarrow$$

ANGLE AXIS



$\Delta\theta$

$\hat{r}$ (unit vector)

$$\begin{cases} \Delta\theta = \cos^{-1}\left(\dfrac{r_{11}+r_{22}+r_{33}-1}{2}\right) & \text{SCALAR} \\ \hat{r} = \dfrac{1}{2\sin\Delta\theta}\begin{bmatrix} r_{32}-r_{23} \\ r_{13}-r_{31} \\ r_{21}-r_{12} \end{bmatrix} \in \mathbb{R}^3 \end{cases}$$

$\hookrightarrow \Delta\theta \approx \emptyset$ $\hat{r}$ IS not DEFINED!

ORIENTATION ERROR

$$\boxed{e_o = \Delta\theta\, \hat{r}}$$

$\Rightarrow$ is defined in the end effector frame

$\Rightarrow$ the Jacobian is in the world frame

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

$\Rightarrow$ mapping before computing the moment

DERIVATIVE OF A ROTATION MATRIX

$${}_W R_B \; {}_W R_B^T = I$$ orthogonality

$$\downarrow d/dt$$

$$\underbrace{\dot{R} R^T}_{S} + R \dot{R}^T = 0$$

$S + S^T = 0 \Rightarrow S$ is skew sim

$$\dot{R} R^T = S \quad \Rightarrow \quad {}_W \dot{R}_B = S \; {}_W R_B \, , \quad S = ?$$

- consider constant vector P in a rotating frame B

$${}_W P = {}_W R_B \; {}_B P$$

$${}_W \dot{P} = {}_W \dot{R}_B \; {}_B P + {}_W R_B \; \overset{=0}{{}_B \dot{P}} = {}_W \dot{R}_B \; {}_B P$$

$$ {}_{W}\dot{P} = S \, {}_{W}R_{B} \cdot {}_{B}P = S \cdot {}_{W}P \qquad (1) $$

- from mechanics: velocity of point in a frame rotating at ω

$$ {}_{W}\dot{P} = {}_{W}\omega \times {}_{W}P \qquad (2) $$

$$ \boxed{ {}_{W}\dot{R}_{B} = S \, {}_{W}R_{B} = {}_{W}\omega \times {}_{W}R_{B} } $$

That is equivalent to:

$$ \boxed{ {}_{W}\dot{R}_{B} = [{}_{W}\omega]_{\times} \, {}_{W}R_{B} } $$

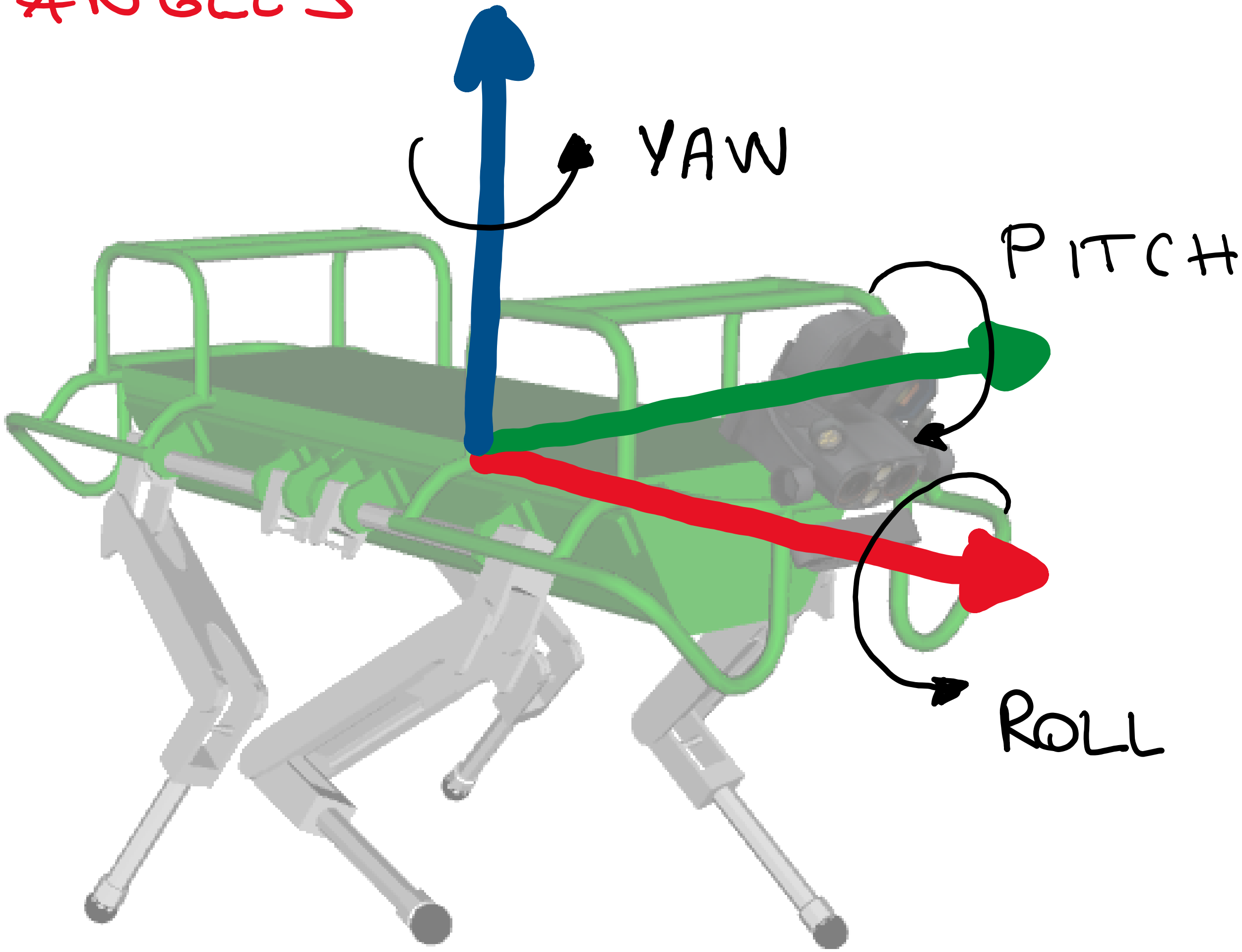
Orientation Control

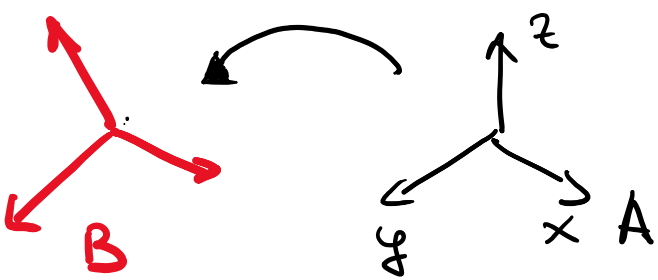
Mapping to torques

Parametrizations of orientation

- Angle-axis
- Rotation Matrix
- Euler-angles
- **Unit Quaternions**

EULER ANGLES





rotation about x : ROLL φ

rotation about y : PITCH θ

rotation about z : YAW ψ

EULER ANGLES

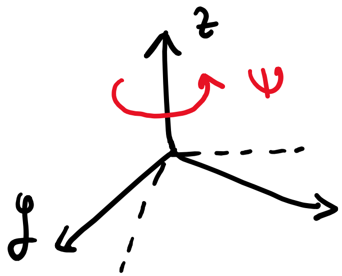
most famous : ZYX convention (TAIT BRYAN ANGLES)

⊖ minimal representation (3 params)
have singularity (GIMBAL LOCK)

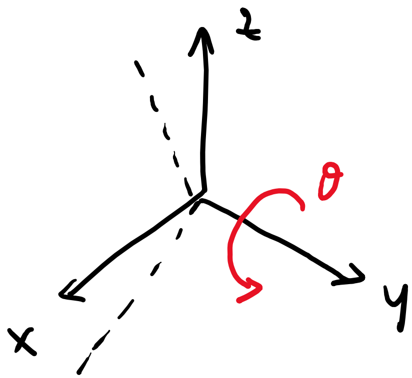
⊖ NOT orthonormal axes (subsequent rotations)

⊕ easy to be numerically integrated

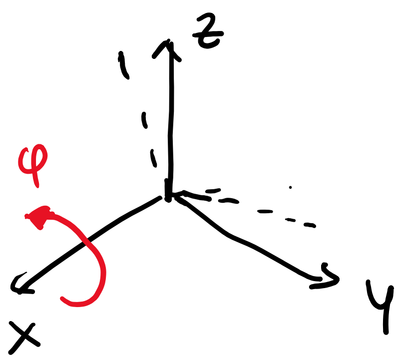
can be linked to rotation matrix composing BASIC ROTATIONS (CCW = ⊕ angle)



$$R_z = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$$R_y = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

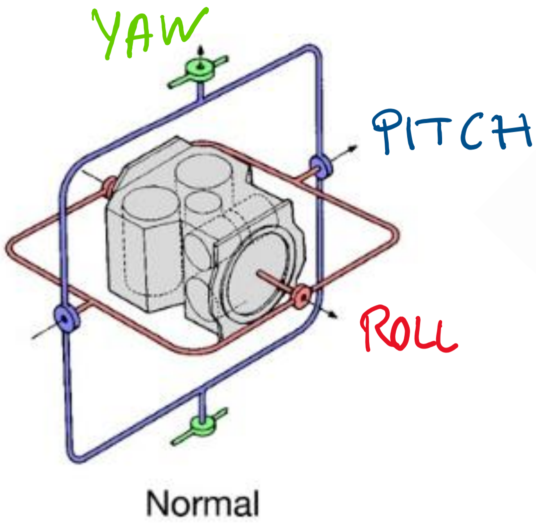


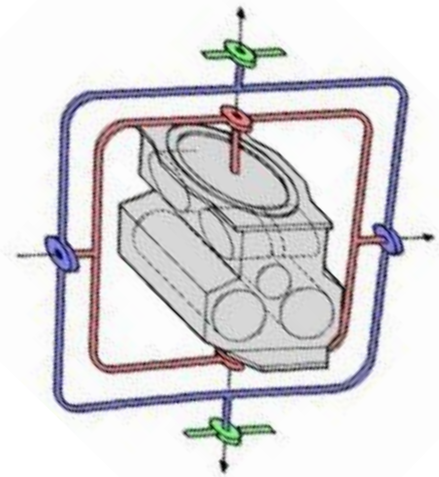
$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & \sin\varphi \\ 0 & -\sin\varphi & \cos\varphi \end{bmatrix}$$

$${}_B R_A = R_x(\theta) \cdot R_y(\theta) \cdot R_z(\psi)$$

subsequent rotations

GIMBAL LOCK





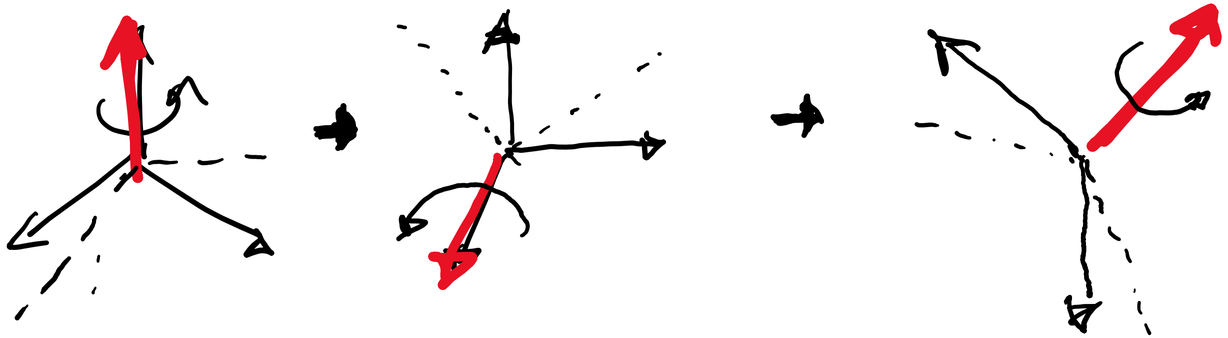
GIMBAL LOCK : 2 AXES ALIGNED

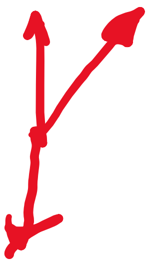
singularity : PITCH = 90° ⇒ ROLL ≈ YAW

↳ more sets of euler angles representing the same orientation

e.g. $R(-45°, 90°, 0) = R(0, 90°, 45°)$

NON ORTHOGONAL AXES





because they are SUBSEQUENT rotations!

$$\bar{\emptyset} = [\varphi, \theta, \psi]$$

We cannot compute $e_o = \emptyset^d - \emptyset$!

$\Rightarrow$ map to R or to angle / axis

EULER RATES AND OMEGA

$$\dot{\Phi} = [\dot{\varphi}, \dot{\theta}, \dot{\psi}]$$ EULER RATES

$\int \dot{\Phi}\, dt \neq \omega$ because ω lives in an euclidean space (90° axes)

- a mapping exists that depends on RPY

$$\boxed{{}_W\omega = T(\Phi)\,\dot{\Phi}}$$

$$T(\Phi) = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\varphi & \cos\theta \sin\varphi \\ 0 & -\sin\varphi & \cos\theta\cos\varphi \end{bmatrix}$$

Orientation Control

Mapping to torques

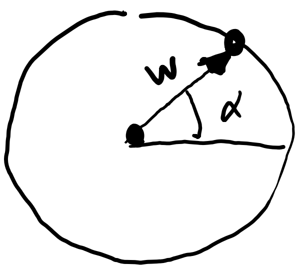
Parametrizations of orientation

- Angle-axis
- Rotation Matrix
- Euler-angles
- Unit Quaternions

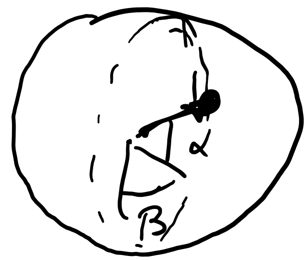
UNIT QUATERNION

(+) 4 params – non minimal
no singularity

(−) unit norm constraint



point on a circle
w, α



point on a sphere
w, α, β

?

quaternion is like a point on a 4D sphere
w, α, β, γ

$$q = \begin{bmatrix} w \\ \varepsilon \end{bmatrix} \begin{matrix} \in \mathbb{R} \\ \in \mathbb{R}^3 \end{matrix}$$

scalar part

$\varepsilon_x, \varepsilon_y, \varepsilon_z$ vectorial part

- unit norm constraint: $\|q\| = 1$

$$\Rightarrow w^2 + \varepsilon_x^2 + \varepsilon_y^2 + \varepsilon_z = 1 \quad \text{or} \quad w^2 + \varepsilon^T \varepsilon = 1$$

Rodriguez formula for quaternions

$$\boxed{R = (\eta^2 - \varepsilon^T \varepsilon) I + 2 \varepsilon \varepsilon^T - 2\eta [\varepsilon]_\times}$$

INVERSE OF QUATERNION

$$q^{-1} = \bar{q} = \begin{bmatrix} w \\ -\varepsilon \end{bmatrix}$$

$(\eta, \varepsilon) = (-\eta, -\varepsilon) \rightarrow$ are same orientation

ANGLE-AXIS TO QUATERNION

$$q \begin{cases} \eta = \cos\left(\frac{\Delta\theta}{2}\right) \\ \varepsilon = \hat{r} \sin\left(\frac{\Delta\theta}{2}\right) \end{cases}$$

for small $\Delta\theta \approx 0$

$$q = \begin{bmatrix} 1 \\ \frac{\hat{v}}{2} \end{bmatrix}$$

QUATERNION TO ANGLE-AXIS

$$\begin{cases} \theta = 2 \arccos(\eta) \\ r = \frac{\varepsilon}{\|\varepsilon\|} = \frac{\varepsilon}{\sqrt{1-\eta}} \end{cases} \qquad (1)$$

quaternions follow a special algebra ...

QUATERNION MULTIPLICATION

$$q_3 = q_2 \otimes q_1$$

$$= \begin{bmatrix} \eta_1 \eta_2 - \varepsilon_1^T \varepsilon_2 \\ \eta_1 \varepsilon_2 + \eta_2 \varepsilon_1 + \varepsilon_1 \times \varepsilon_2 \end{bmatrix} \qquad (2)$$

ORIENTATION ERROR WITH QUATERNIONS

like $R_a^d = R^d R_a^T \rightarrow \boxed{q_e = q_d \otimes \bar{q}_a}$

$$q_d = \begin{bmatrix} \eta^d \\ \varepsilon^d \end{bmatrix} \qquad \bar{q}_a = \begin{bmatrix} \eta_a \\ -\varepsilon_a \end{bmatrix}$$

(inverse)

apply (2):

$$q_e = \begin{bmatrix} \eta_a \eta_d + \varepsilon_a^T \varepsilon^d \\ \eta_a \varepsilon_d - \eta^d \varepsilon_a - \varepsilon_a \times \varepsilon_d \end{bmatrix}$$

$q_e \rightarrow (1) \rightarrow \Delta\vartheta, \hat{r} \rightarrow e_o$

References:

- 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.