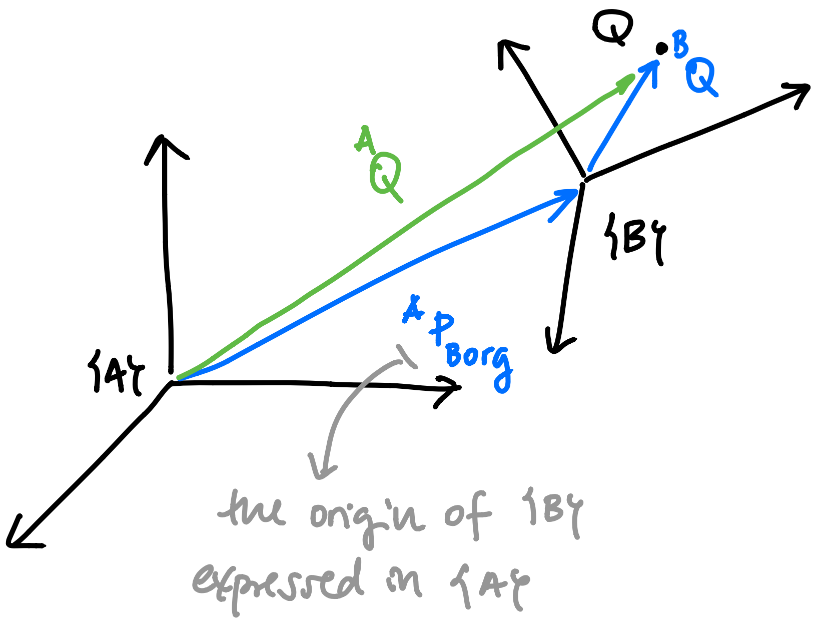
① Time varying position and orientation: linear & angular velocities



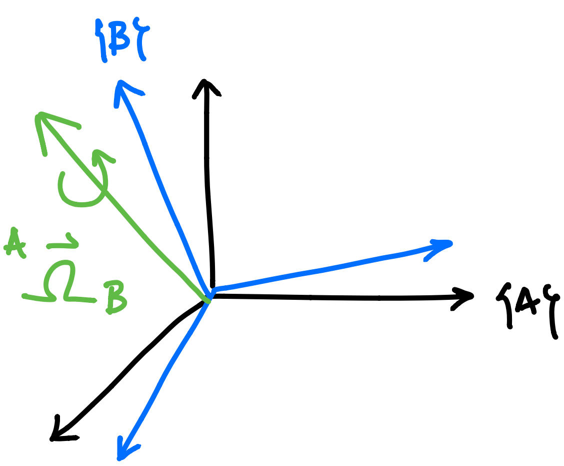
For velocities, it is important

1) the frame in which the differentiation of the position is done
2) the frame in which the velocity is expressed

$$\boxed{{}^BV_Q = \frac{d}{dt}{}^BQ = \lim_{\Delta t \to 0} \frac{{}^BQ(t+\Delta t) - {}^BQ(t)}{\Delta t}}$$: Velocity of Q computed in B and expressed in B

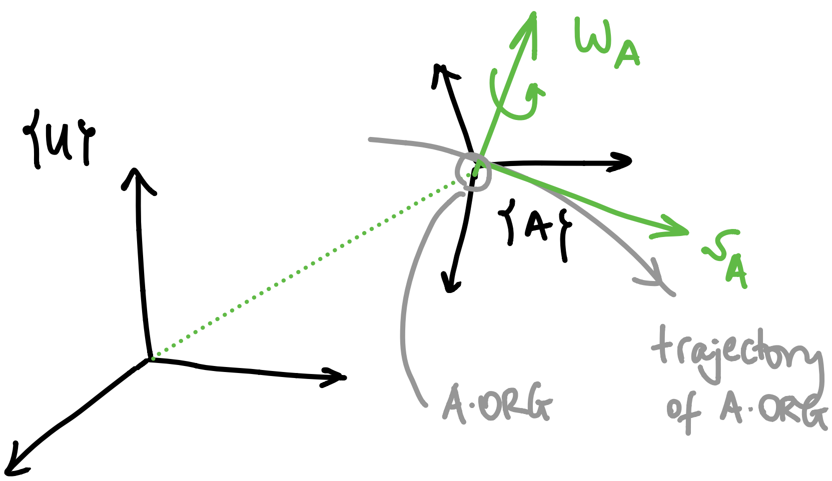
$$\boxed{{}^A({}^BV_Q) = {}^A_{B}R \, {}^BV_Q}$$: Velocity of Q computed in B and expressed in A

The linear velocity refers to a point, but the angular velocity refers to a body. The angular velocity ${}^A\Omega_B$ is the instantaneous axis of rotation of frame B relative to A.



It can be also expressed in another frame C : ${}^C({}^A\Omega_B)$

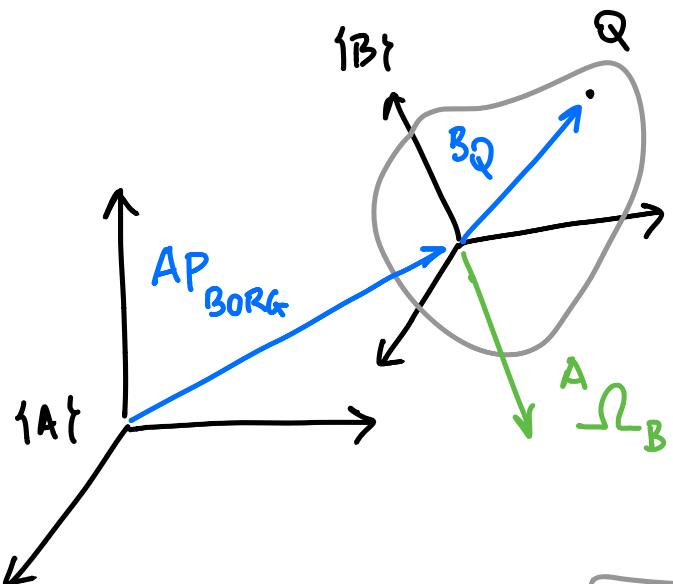
In general the velocity of a frame A refers to the linear velocity of its origin wrt. a fixed universal frame and its angular velocity:



$$\vec{\omega}_A = {}^U\Omega_A$$

$$\vec{v}_A = {}^U V_{AORG}$$

② Velocity of Rigid Bodies



Given a rigid body with frame B moving wrt a frame A, the velocity of its point Q is the following, considering simultaneous linear and rotational motion:

$$
{}^A V_Q = \underbrace{{}^A V_{BORG}}_{\text{velocity of the origin of B expressed in A}} + \overbrace{\overbrace{\underbrace{{}^A \underset{\sim}{R}_B\,{}^B V_Q}_{\text{this is } \emptyset \text{ if Q is fixed in B!}}}^{{}^A V_Q \text{ component if Q not fixed in B}} + \overbrace{\underbrace{\underbrace{{}^A \Omega_B}_{\vec{\omega}_B} \times \underbrace{{}^A R_B\,{}^B Q}_{{}^A Q}}_{\text{This component is } \emptyset \text{ if B is not rotating!}}}^{{}^A V_Q \text{ if Q were fixed in B}}}^{{}^A V_Q}
$$

3 Notes on the Angular Velocity

$$\underset{\sim}{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \longrightarrow \underset{\sim}{R} \cdot \underset{\sim}{R}^T = \underset{\sim}{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\frac{d}{dt}(\underset{\sim}{R} \cdot \underset{\sim}{R}^T) = \dot{\underset{\sim}{R}}\underset{\sim}{R}^T + \underset{\sim}{R}\dot{\underset{\sim}{R}}^T = \underset{\sim}{0}$$

$$= \dot{R}R^T + (\dot{R}R^T)^T = 0$$

$$= S + S^T = 0$$

$$\boxed{\underset{\sim}{S} = \dot{\underset{\sim}{R}}\underset{\sim}{R}^T} = \dot{\underset{\sim}{R}}\underset{\sim}{R}^{-1} = \begin{bmatrix} 0 & -\Omega_z & \Omega_y \\ \Omega_z & 0 & -\Omega_x \\ -\Omega_y & \Omega_x & 0 \end{bmatrix} = \underbrace{[\Omega]}_{\text{cross product operator}} = \Omega \wedge$$

$$\Omega = \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix}$$

skew-symmetric matrix which computes the angular velocity Ω given the rotation vector $\underset{\sim}{R}$

matrix elements are the elements of the angular velocity!

the skew matrix acts as a cross product of the angular velocity!

$$\boxed{\underset{\sim}{S} \cdot \vec{P} = [\Omega]\,\vec{P} = \vec{\Omega} \times \vec{P}}$$

Therefore, the skew symmetric matrix is aka the angular velocity matrix.

The physical meaning of the angular velocity $\vec{\Omega}$: instantaneous axis of rotation ($\vec{e}$, $\|\vec{e}\| = 1$) around which we rotate at a speed $\dot{\theta}$

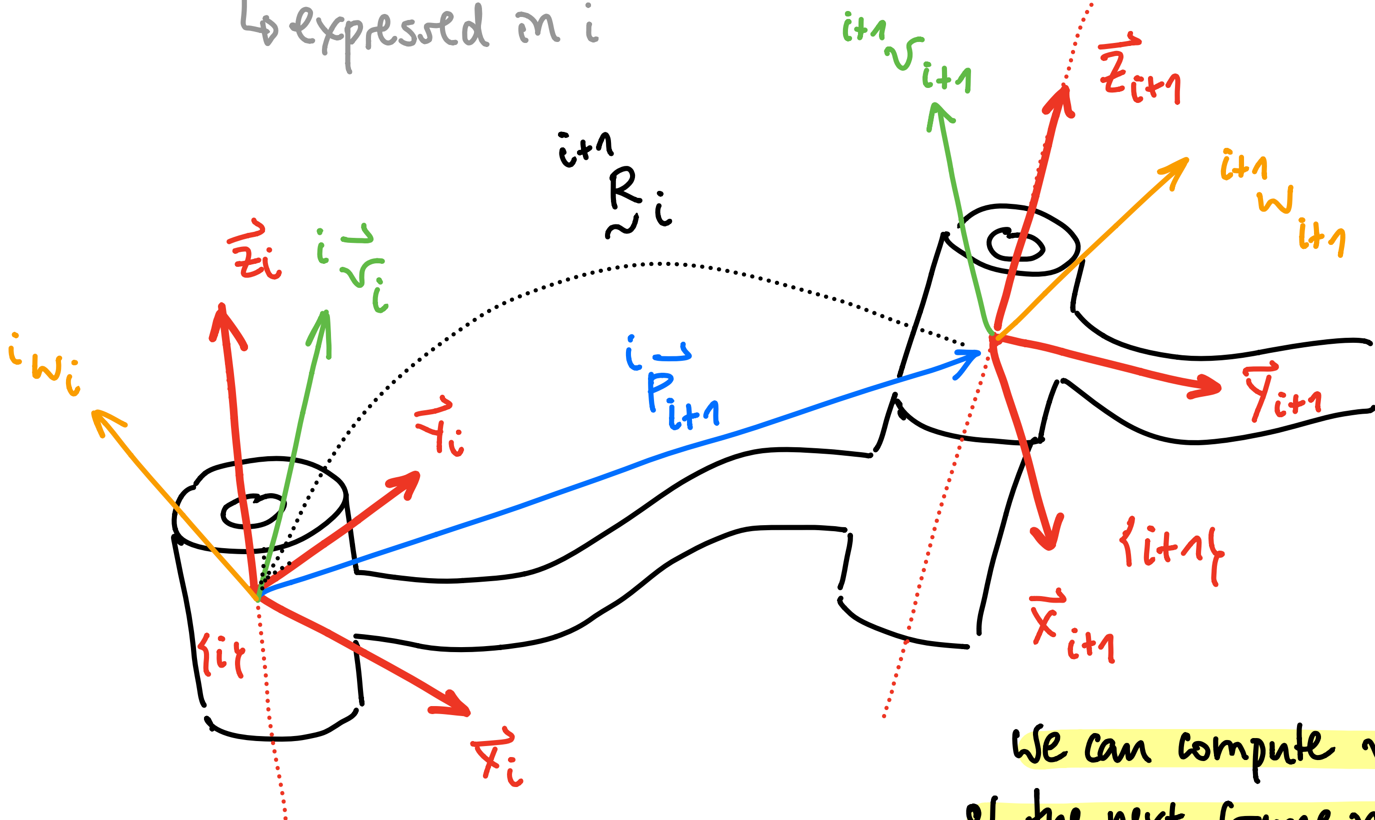
$$\boxed{\vec{\Omega} = \dot{\theta}\,\underbrace{\vec{e}}_{\text{unitary vector}} = \|\Omega\| \frac{\Omega}{\|\Omega\|}}$$

④ Velocity propagation from link to link

Using the formulas from previous sections we can compute the linear and angular velocities of each link i in frame $\{i\}$:

- $v_i = {}^{i}v_i$: velocity of the origin of frame $\{i\}$, expressed in i
- $w_i = {}^{i}w_i$: angular velocity of frame $\{i\}$, expressed in i

expressed in i



We can compute v & w of the next frame in terms of the previous!

For REVOLUTE (R) joints:

$$\boxed{{}^{i+1}\vec{w}_{i+1} = {}^{i+1}\underset{\sim}{R}_i \, {}^{i}\vec{w}_i + \overbrace{\dot{\theta}_{i+1} \, {}^{i+1}\vec{Z}_{i+1}}}$$

The local component of angular velocity is in Z and with the angle rate!

$$\boxed{\underbrace{{}^{i+1}\vec{v}_{i+1}} = {}^{i+1}\underset{\sim}{R}_i \left(\underbrace{{}^{i}\vec{v}_i + {}^{i}\vec{w}_i \times \overbrace{{}^{i}\vec{P}_{i+1}}} \right)}$$

Distance vector from i origin to $i+1$ origin expressed in i

expressed in, velocities of $i+1$

from PREVIOUS frame i

For PRISMATIC (P) joints:

$$\boxed{{}^{i+1}\vec{\omega}_{i+1} = {}^{i+1}\underset{\sim}{R}_i \, {}^{i}\vec{\omega}_i}$$

We don't have local rotations around joint axis, but linear motion along it!

$$\boxed{{}^{i+1}\vec{v}_{i+1} = {}^{i+1}\underset{\sim}{R}_i \left({}^{i}\vec{v}_i + {}^{i}\vec{\omega}_i \times {}^{i}\vec{P}_{i+1} \right) + \underbrace{\dot{d}_{i+1}}_{} \, {}^{i+1}\vec{Z}_{i+1}}$$

invert (transpose) for using here...

velocity of joint movement along Z

All v & ω are expressed in their link frames – if we want to express them in any other frame (e.g., base frame {0}), we premultiply them with ${}^{0}\underset{\sim}{R}_i$:

$$\boxed{\begin{aligned} {}^{0}\vec{v}_i &= {}^{0}\underset{\sim}{R}_i \, {}^{i}\vec{v}_i \\ {}^{0}\vec{\omega}_i &= {}^{0}\underset{\sim}{R}_i \, {}^{i}\vec{\omega}_i \end{aligned}}$$

Note that $\underset{\sim}{R}$ & $\vec{P}$ are obtained from the DH transformations!

$${}^{i}\underset{\sim}{H}_{i+1} = \left[\begin{array}{c|c} {}^{i}\underset{\sim}{R}_{i+1} & {}^{i}\vec{P}_{i+1} \\ \hline 0 \; 0 \; 0 & 1 \end{array}\right]$$

Note that the equations above can be unified and depending on P/R, $\dot{d}/\dot{\theta}$ cancel automatically.

The strategy for using the velocity propagation equations is:

1) Compute velocities in base frame or first joint: $\underbrace{{}^{0}v_0, {}^{0}\omega_0}_{\text{oftentimes } \emptyset}$ $\Rightarrow$ 2) ${}^{1}v_1, {}^{1}\omega_1$ ...

$\uparrow$

${}^{0}v_0, {}^{0}\omega_0$

⑤ The Jacobian Matrix

The Jacobian matrix can be used to

- find singularities
- determine forces & torques that act on a joint
- calculate joint velocities ← the Jacobian is actually built with the joint velocities

Consider the position functions y_i of a system:

$$\left.\begin{aligned} y_1 &= f_1(x_1, x_2, \dots, x_m) \\ y_2 &= f_2(x_1, x_2, \dots, x_m) \\ &\vdots \\ y_n &= f_n(x_1, x_2, \dots, x_m) \end{aligned}\right\}$$

Position: $\underset{\sim}{Y} = (y_1, y_2, \dots y_n)^T$

$\underset{\sim}{F} = (f_1, f_2, \dots f_n)^T$

$$\partial Y = \frac{\partial F}{\partial x} \cdot \partial x$$

$$\boxed{\dot{Y} = \frac{\partial F}{\partial x} \cdot \dot{x}}$$

Note that although we want to obtain a closed form expression of the Jacobian, its numeric values will change!

JACOBIAN MATRIX $\underset{\sim}{J}$

$$\underbrace{\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \vdots \\ \dot{y}_n \end{bmatrix}}_{n \times 1} = \underbrace{\begin{bmatrix} \dfrac{\partial F}{\partial x} \end{bmatrix}}_{n \times m} \cdot \underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_m \end{bmatrix}}_{m \times 1}$$

In general Jacobians can be rectangular

velocities: positions differentiated wrt time

parameters differentiated wrt time

The Jacobian can be used to relate joint velocities and cartesian velocities in a manipulator; typically the velocities of the tip of the arm are related with the joint velocities:

thus, linear velocities

Cartesian velocities at the tip

Joint velocities

$$\begin{bmatrix} {}^0 v \\ {}^0 \omega \end{bmatrix} = {}^0 \underset{\sim}{J}(\underset{\sim}{\theta}) \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_6 \end{bmatrix} \longleftrightarrow \begin{bmatrix} {}^0 v_x \\ {}^0 v_y \\ {}^0 v_z \\ {}^0 \omega_x \\ {}^0 \omega_y \\ {}^0 \omega_z \end{bmatrix} = {}^0 \underset{\sim}{J}(\underset{\sim}{\theta}) \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_6 \end{bmatrix}$$

frame in which velocities are expressed

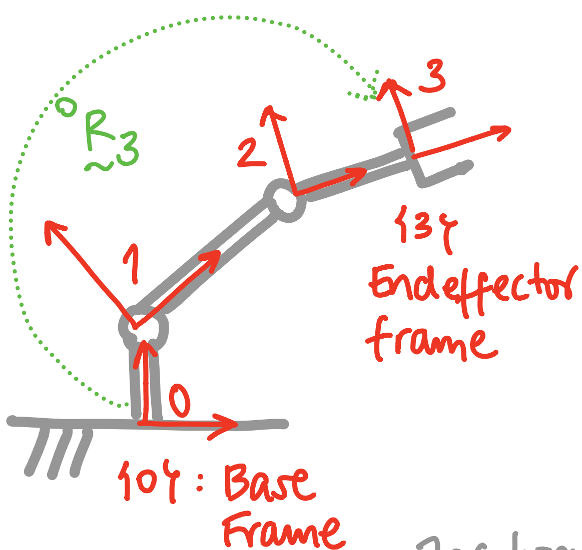
$$\boxed{{}^0 \vec{v} = {}^0 \underset{\sim}{J}(\underset{\sim}{\theta}) \, \vec{\dot{\theta}}}$$

angular velocity

Particular case in which we have 6 joints

Note, however, that for our purposes we consider only the linear velocities!

⑥ The Frame of the Jacobian



- The Jacobian (as well as poses and velocities) is always expressed in a frame!
- To transform the Jacobian from a frame to another we use $\underset{\sim}{R}$:

$$\boxed{{}^0 \underset{\sim}{J} = {}^0 \underset{\sim}{R}_3 \cdot {}^3 \underset{\sim}{J}}$$

Jacobian at base frame

See next note.

- In robotics, the Jacobian which relates the joint motion with the linear velocity is used. However, the general expression of the Jacobian can link the angular velocity too. For that case, the Jacobian is extended with angular mapping terms and its transformation too:

Incorrect transformation

$$\underbrace{{}^{0}\tilde{J}}_{6\times 6} = \underbrace{{}^{0}R_3}_{3\times 3} \cdot \underbrace{{}^{3}\tilde{J}}_{6\times 6}$$

$\neq$ $\neq$

Correct velocity transformation

$$\underbrace{\begin{bmatrix} {}^{0}v \\ {}^{0}\omega \end{bmatrix}}_{6\times 1} = \underbrace{\begin{bmatrix} {}^{0}R_3 & 0 \\ 0 & {}^{0}R_3 \end{bmatrix}}_{6\times 6} \underbrace{\begin{bmatrix} {}^{3}v \\ {}^{3}\omega \end{bmatrix}}_{6\times 1}$$

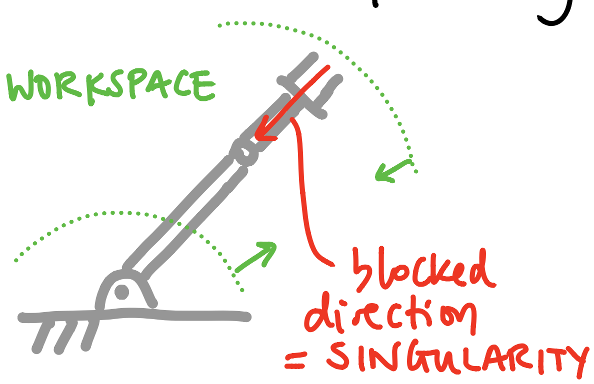
$$\boxed{{}^{0}\tilde{J} = \begin{bmatrix} {}^{0}R_3 & 0 \\ 0 & {}^{0}R_3 \end{bmatrix} {}^{3}\tilde{J}}$$

Applying the idea above, we build the correct transformation of the Jacobian from frame 3 (endeffector) to 0 (base)

GENERAL JACOBIAN TRANSFORM, considering angular velocity components, too

7 Singularities

A singularity is a point / configuration in which the robot loses one or more degrees of freedom (DoF) in a certain direction in Cartesian space, regardless of the joint rates.



It happens when joint axes align, thus, in all workspace boundaries, among others.

Singularities can be detected because we cannot obtain the necessary joint rates $\dot{\theta}$ from the velocities v using the Jacobian, because it has no inverse:

Jacobian with linear velocities in the base frame

$$\vec{v} = {}^{0}\underset{\sim}{J} \cdot \vec{\dot{\theta}} \longrightarrow \vec{\dot{\theta}} = {}^{0}\underset{\sim}{J}^{-1} \cdot \vec{v} \quad \exists \text{ iff } \det(\underset{\sim}{J}) \neq 0$$

Singularities appear when $\det({}^{0}\underset{\sim}{J}) = 0$

Two types of singularities are distinguished

1. Workspace-boundary singularities: when the robot is fully streched out or folded back on itself (see image above)
2. Workspace-interior singularities: tool is not at the boundary (as in 1), but 2 or more axis are aligned

We want to avoid singularities! So we compute the joint values that produce them and avoid reaching them:

$$\det({}^{0}J(\theta_i)) = 0 \Rightarrow f(\theta_1, \theta_2, \ldots, \theta_6) = 0 \Rightarrow \ldots$$

8 Calculating the Jacobian

Two methods introduced here:

1) Partial differentiation method
2) Velocity propagation method

8.1. Partial Differentiation for Calculating the Jacobian

- Recall: the Jacobian relates the joint velocities and the cartesian velocities; so it's about finding joint velocities. The Jacobian used to compute singularities.

$$\left.\begin{aligned} y_1 &= f_1(x_1, x_2, x_3, \dots) \\ y_2 &= f_2(x_1, x_2, x_3, \dots) \\ &\vdots \\ y_n &= f_n(x_1, x_2, x_3, \dots) \end{aligned}\right\} \xrightarrow{\text{diff.}} \left\{\begin{aligned} \partial y_1 &= \frac{\partial f_1}{\partial x_1} \cdot \partial x_1 + \frac{\partial f_1}{\partial x_2} \cdot \partial x_2 + \dots \\ \partial y_2 &= \frac{\partial f_2}{\partial x_1} \cdot \partial x_1 + \frac{\partial f_2}{\partial x_2} \cdot \partial x_2 + \dots \\ &\vdots \\ \partial y_n &= \frac{\partial f_n}{\partial x_1} \partial x_1 + \frac{\partial f_2}{\partial x_2} \partial x_2 + \dots \end{aligned}\right.$$

POSITION EQS OF JOINTS

$$\partial y = \frac{\partial F}{\partial x} \cdot \partial x$$

$$\boxed{\dot{y} = \underset{\sim}{J}\, \dot{x}} \longleftarrow \underbrace{\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \vdots \\ \dot{y}_n \end{bmatrix}}_{\text{velocities of joints}} = \underbrace{\begin{bmatrix} \dfrac{\partial F}{\partial x} \end{bmatrix}}_{\text{JACOBIAN MATRIX}} \cdot \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix}$$

- steps we follow:
  1) Create DH table & transformation matrices ${}^{i-1}\underset{\sim}{H}_i$: ${}^{0}\underset{\sim}{H}_1$, ${}^{1}\underset{\sim}{H}_2$, ${}^{2}\underset{\sim}{H}_3$ ...
  2) Compose the parametrized transformation matrix from the base to the endeffector using the DH matrices:

$$\underset{\sim}{{}^{0}H}_{E} = \underset{\sim}{{}^{0}H}_{1} \cdot \underset{\sim}{{}^{1}H}_{2} \cdot \underset{\sim}{{}^{2}H}_{3} \cdots = \left[\begin{array}{c|c} \underset{\sim}{R} & \boxed{\begin{matrix} P_x \\ P_y \\ P_z \end{matrix}} \\ \hline 0\,0\,0 & 1 \end{array}\right]$$

position of the tool/endeffector wrt the base

3) Differentiate $\vec{P} = (P_x, P_y, P_z)^T$ wrt the time and isolate $\dot{\vartheta}_i$ components to re-arrange the equations to have the Jacobian factorized:

$$\begin{bmatrix} \dot{P}_x \\ \dot{P}_y \\ \dot{P}_z \end{bmatrix} = \begin{bmatrix} \cdots \\ \cdots \\ \cdots \end{bmatrix} = \cdots = \underset{\sim}{J} \begin{bmatrix} \dot{\vartheta}_1 \\ \dot{\vartheta}_2 \\ \dot{\vartheta}_3 \end{bmatrix}$$

we take only the linear velocity / linear position

Example

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} r_1 \cos\vartheta \\ r_2 \sin\vartheta \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} \dot{P}_x \\ \dot{P}_y \\ \dot{P}_z \end{bmatrix} = \begin{bmatrix} -r_1 \sin\vartheta \cdot \dot{\vartheta} \\ r_2 \cos\vartheta \cdot \dot{\vartheta} \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \dot{P}_x \\ \dot{P}_y \\ \dot{P}_z \end{bmatrix} = \boxed{\begin{bmatrix} -r_1 \sin\vartheta \\ r_2 \cos\vartheta \\ 0 \end{bmatrix}} \cdot \left[\dot{\vartheta}\right]$$

$$\dot{Y} = \boxed{{}^{0}\underset{\sim}{J}} \cdot \dot{X}$$

I'm not sure if that method is properly described... I think in Craig the next method seems to be described more thoroughly...

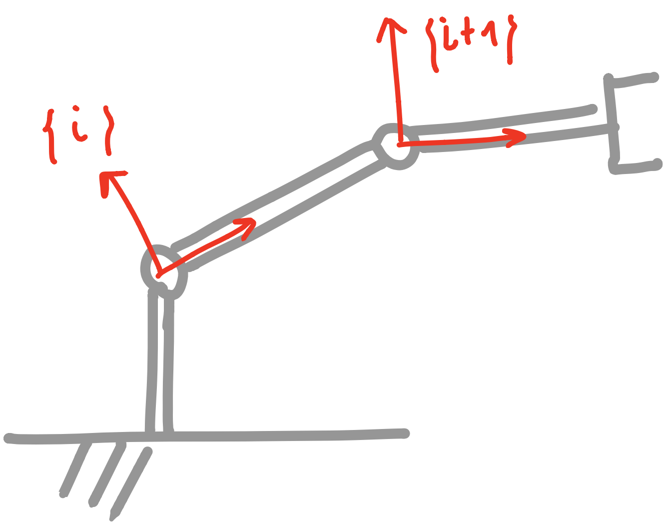
8.2. Velocity Propagation Method for Calculating the Jacobian

This method consists in applying the techniques in Section 4. The partial differentiation is sometimes difficult because very complex functions appear – thus, in that cases the velocity propagation is preferred.

According to section 4, we can propagate the velocities from links/joints as follows:

$$\boxed{\begin{aligned}
{}^{i+1}\vec{\omega}_{i+1} &= {}^{i+1}\underset{\sim}{R}_i \; {}^{i}\vec{\omega}_i + \overbrace{\dot{\theta}_{i+1} \; {}^{i+1}\vec{Z}_{i+1}}^{\varnothing \text{ for prismatic joints; else: } \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_{i+1} \end{bmatrix}} \\
{}^{i+1}\vec{v}_{i+1} &= {}^{i+1}\underset{\sim}{R}_i \left({}^{i}\vec{v}_i + {}^{i}\vec{\omega}_i \times {}^{i}\vec{P}_{i+1} \right) + \underbrace{\dot{d}_{i+1} \; {}^{i+1}\vec{Z}_{i+1}}_{\varnothing \text{ for revolute joints, else: } \begin{bmatrix} 0 \\ 0 \\ \dot{d}_{i+1} \end{bmatrix}}
\end{aligned}}$$

velocities of frame i+1 expressed in frame i



They are function of the velocities of previous frame i

Steps: for that we need the DH table and DH transformations!

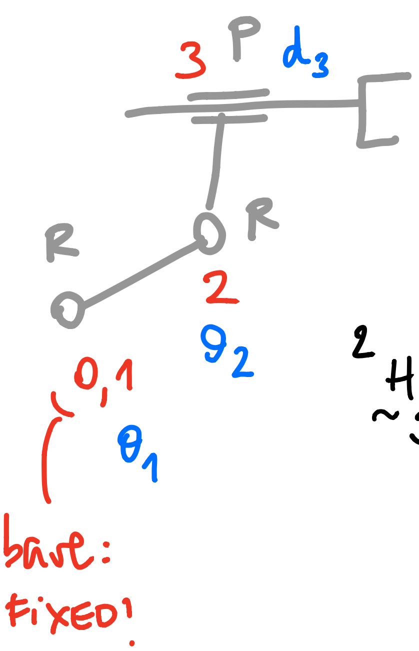
1) Find ${}^0\tilde{v}_0$ and ${}^0\tilde{\omega}_0$ and propagate them until the last frame

2) Find the Jacobian at the last frame

3) Transform the Jacobian to the base frame

4) Find the equation for the singularities: $\det({}^0\underset{\sim}{J}) = 0$

Example



$${}^0\underset{\sim}{H}_1 = \left[\begin{array}{ccc|c} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \qquad {}^0\underset{\sim}{R}_1,\ {}^0P_1$$

$${}^1\underset{\sim}{H}_2 = \left[\begin{array}{ccc|c} -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ c_2 & -s_2 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \qquad {}^1\underset{\sim}{R}_2,\ {}^1P_2$$

$${}^2\underset{\sim}{H}_3 = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -d_3 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \qquad {}^2\underset{\sim}{R}_3,\ {}^2P_3$$

$${}^0\underset{\sim}{H}_3 = \left[\begin{array}{ccc|c} -c_1 s_2 & s_1 & c_1 c_2 & c_1 c_2 d_3 \\ -s_1 s_2 & c_1 & s_1 c_2 & s_1 c_2 d_3 \\ c_2 & 0 & s_2 & s_2 d_3 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \qquad {}^0\underset{\sim}{R}_3$$

$${}^0H_3 = {}^0H_1\,{}^1H_2\,{}^2H_3$$

1) We start propagating the velocities:

* Base: 0: FIXED!

$${}^0\tilde{v}_0 = (0,0,0)^T, \quad {}^0\omega_0 = (0,0,0)^T$$

* **Frame 1 : θ_1**

$$^1\omega_1 = {}^1R_0(\cancel{^0\omega_0}_{0}) + \dot\theta_1 \, {}^1\hat{\vec z}_1 = \dot\theta_1 \, {}^1\hat{\vec z}_1 = \begin{bmatrix} 0 \\ 0 \\ \dot\theta_1 \end{bmatrix}$$

$$^1v_1 = {}^1R_0(\cancel{^0v_0}_{0} + \cancel{^0\omega_0}_{0} \times {}^0P_1) + \cancel{\dot d_1}_{0} \, {}^1\hat{\vec z}_1 = \vec 0 = (0,0,0)^T$$

* **Frame 2 : θ_2**

$$^2\omega_2 = {}^2R_1(^1\omega_1) + \dot\theta_2 \, {}^2\vec z_2 =$$

$$= \underbrace{\begin{bmatrix} -s_2 & 0 & c_2 \\ -c_2 & 0 & -s_2 \\ 0 & 1 & 0 \end{bmatrix}}_{^2R_1 = (^1R_2)^T} \begin{bmatrix} 0 \\ 0 \\ \dot\theta_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot\theta_2 \end{bmatrix} = \begin{bmatrix} c_2\dot\theta_1 \\ -s_2\dot\theta_1 \\ \dot\theta_2 \end{bmatrix}$$

$$^2v_2 = {}^2R_1(\cancel{^1v_1}_{0} + {}^1\omega_1 \times \cancel{^1P_2}_{0}) + \cancel{\dot d_2}_{0} \, {}^2\vec z_2 = \vec 0$$

look at $^1\underset{\sim}{H}_2$

* **Frame 3 : d_3**

$$^3\vec W_3 = {}^3R_2(^2\vec W_2) + \cancel{\dot\theta_3}_{0} \, {}^3\vec z_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} c_2\dot\theta_1 \\ -s_2\dot\theta_1 \\ \dot\theta_2 \end{bmatrix} = \begin{bmatrix} c_2\dot\theta_1 \\ \dot\theta_2 \\ s_2\dot\theta_1 \end{bmatrix}$$

$$ {}^3\vec{v}_3 = {}^3\underset{\sim}{R}_2 \left(\cancel{{}^2\vec{v}_2}^{\,0} + {}^2\vec{\omega}_2 \times {}^2\vec{P}_3 \right) + \dot{d}_3 \, {}^3\vec{z}_3 = $$

$$ = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \left(\vec{0} + \begin{bmatrix} c_2 \dot{\theta}_1 \\ -s_2 \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \times \begin{bmatrix} 0 \\ -d_3 \\ 0 \end{bmatrix} \right) + \dot{d}_3 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = $$

$$ = \begin{bmatrix} d_3 \dot{\theta}_2 \\ -d_3 c_2 \dot{\theta}_1 \\ \dot{d}_3 \end{bmatrix} $$

2) Now we find the Jacobian at the last frame: ${}^3\underset{\sim}{J}$.

In order to propagate the velocities we need both ω_i & v_i, but for finding the Jacobian we consider only the linear velocity. We factor v_N (N: last frame) in terms of joint rates

$$ \boxed{ {}^N\vec{v}_N = {}^N J \cdot {}^N\vec{\dot{\theta}} } $$

This is always the joint rates

this is the linear velocity of the last frame

$$ {}^3\vec{v}_3 = \underbrace{\begin{bmatrix} 0 & d_3 & 0 \\ -d_3 c_2 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{{}^3\underset{\sim}{J}} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{d}_3 \end{bmatrix} = \begin{bmatrix} d_3 \dot{\theta}_2 \\ -d_3 c_2 \dot{\theta}_1 \\ \dot{d}_3 \end{bmatrix} $$

This is the Jacobian!
We need to find its elements so that the eq. matches

3) Next, we transform ${}^{3}\underset{\sim}{J}$ to the base frame

$$ {}^{0}\underset{\sim}{J} = {}^{0}\underset{\sim}{R}_{3} \cdot {}^{3}\underset{\sim}{J} = \begin{bmatrix} -c_1 s_2 & s_1 & c_1 c_2 \\ -s_1 s_2 & c_1 & s_1 c_2 \\ c_2 & 0 & s_2 \end{bmatrix} \begin{bmatrix} 0 & d_3 & 0 \\ -d_3 c_2 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = $$

$$ = \begin{bmatrix} -s_1 d_3 c_2 & -c_1 s_2 d_3 & c_1 c_2 \\ -c_1 d_3 c_2 & -s_1 s_2 d_3 & s_1 c_2 \\ 0 & c_2 d_3 & s_2 \end{bmatrix} $$

4) Finally, the singularities are found by solving the joint params that yield a Jacobian in the base frame which has $\det({}^{0}\underset{\sim}{J}) = 0$.

↳ any other frame is not valid

$$ \det({}^{0}\underset{\sim}{J}) = f(\theta_1, \theta_2, d_3) = 0 $$

we solve θ_1, θ_2, d_3: These are the singularity parameters: we need to set limits before reaching them!

9 Static Forces in Manipulators

As with the velocity propagation, we can propagate the forces at the endeffector so that we obtain the torques & forces necessary at the joints to maintain static equilibrium while holding an object (self-weight of the links ignored):

$$\boxed{{}^{i}\vec{f}_i = {}^{i}\underset{\sim}{R}_{i+1} \cdot {}^{i+1}\vec{f}_{i+1}}$$

Force in joint i expressed in i

$$\boxed{{}^{i}\vec{t}_i = {}^{i}\underset{\sim}{R}_{i+1} \cdot {}^{i+1}\vec{t}_{i+1} + {}^{i}\vec{P}_{i+1} \times {}^{i}\vec{f}_i}$$

Vector from i to i+1

Torque in joint i expressed in i

The formulas are obtained applying static equilibrium ($\Sigma f, \Sigma t = 0$) on the links:

$$\text{Link } i: \quad {}^{i}\vec{f}_i - {}^{i}\vec{f}_{i+1} = 0 \Rightarrow {}^{i}\vec{f}_i = {}^{i}\underset{\sim}{R}_{i+1} \, {}^{i+1}\vec{f}_{i+1}$$

All forces acting on the robot are resisted by the rigid structure links except the force components around the joints, therefore we project them:

$$\boxed{\begin{aligned} \tau_i &= {}^{i}\vec{t}_i \cdot {}^{i}\vec{Z}_i \\ \tau_i &= {}^{i}\vec{f}_i \cdot {}^{i}\vec{Z}_i \end{aligned}}$$

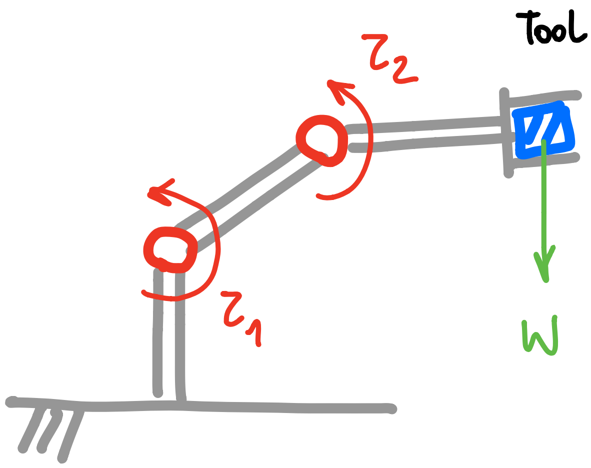
REVOLUTE

PRISMATIC

} joint forces/torques necessary to support a force

However, we can also use the Jacobian to compute them!

⑩ The Jacobian to Compute the Torques & Forces in the Joints necessary to apply static forces at the Endeffector



We consider the case in which we have a weight W at the tool. We want to compute the torques z_i necessary at the joints so that the robot is at static equilibrium. Self-weight of the robot is not considered.

The Jacobian at the base frame can be used for that:

$$\boxed{\vec{\tau} = \left({}^{0}\underset{\sim}{J}\right)^T \, {}^{0}\vec{F}}$$

Important interpretation: when J looses full rank (singularity), we cannot exert forces in some Cartesian directions and small joint torques lead to large forces!

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ f_1 \\ f_2 \\ \vdots \end{bmatrix} = \left({}^{0}\underset{\sim}{J}\right)^T \cdot \begin{bmatrix} {}^{0}F_1 \\ {}^{0}F_2 \\ {}^{0}F_3 \end{bmatrix}$$

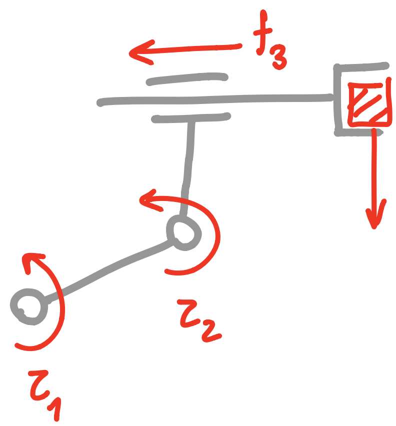
→ the static force at the endeffector / tool. For instance: the weight W of the object held, expressed in the base frame 0

→ forces and torques necessary at the joints

τ : torques, for REVOLUTE joints

f : forces, for PRISMATIC joints

Example



Jacobian of the tool/endeffector in the base frame, computed before

$$
{}^{0}\underset{\sim}{J} = \begin{bmatrix} -s_1 d_3 c_2 & -c_1 s_2 d_3 & c_1 c_2 \\ -c_1 d_3 c_2 & -s_1 s_2 d_3 & s_1 c_2 \\ 0 & c_2 d_3 & s_2 \end{bmatrix}
$$

Applying the formula:

$$
\vec{\tau} = ({}^{0}\underset{\sim}{J})^T \vec{F}
$$

$$
\begin{bmatrix} \tau_1 \\ \tau_2 \\ f_1 \end{bmatrix} = \begin{bmatrix} -s_1 d_3 c_2 & -c_1 d_3 c_2 & 0 \\ -c_1 s_2 d_3 & -s_1 s_2 d_3 & c_2 d_3 \\ c_1 c_2 & s_1 c_2 & s_2 \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \dots
$$

We compute that...

11 Cartesian Transformation of Velocities and Static Forces

We might want to have generalized 6x1 representations of forces and velocities:

$$\vec{\nu} = \begin{bmatrix} \vec{v} \\ \vec{\omega} \end{bmatrix} = (v_1, v_2, v_3, w_1, w_2, w_3)^T$$

$$\vec{F} = \begin{bmatrix} \vec{f} \\ \vec{t} \end{bmatrix} = (f_1, f_2, f_3, t_1, t_2, t_3)^T$$

The instantaneous velocity transformation which maps velocities in {A} to {B} is:

skew matrix operator

$$\boxed{{}^B\vec{\nu}_B = ({}^BT_A)_v \cdot {}^A\vec{\nu}_A} \quad : \quad \begin{bmatrix} {}^B\vec{v}_B \\ {}^B\vec{\omega}_B \end{bmatrix} = \begin{bmatrix} {}^B\underset{\sim}{R}_A & -{}^B\underset{\sim}{R}_A \left[{}^A\vec{P}_{BORG}\right] \\ 0 & {}^B\underset{\sim}{R}_A \end{bmatrix} \begin{bmatrix} {}^A\vec{v}_A \\ {}^A\vec{\omega}_A \end{bmatrix}$$

$$\left[{}^A\vec{P}_{BORG}\right] = \begin{bmatrix} 0 & -P_z & P_y \\ P_z & 0 & -P_x \\ -P_y & P_x & 0 \end{bmatrix}$$

Vector from A to B expressed in A

Similarly, the static force-moment transformation:

$$\boxed{{}^B\vec{f}_B = ({}^BT_A)_f \cdot {}^A\vec{f}_A}$$

$$({}^BT_A)_f = \underline{(({}^BT_A)_v)^T} = \begin{bmatrix} {}^B\underset{\sim}{R}_A & 0 \\ -{}^B\underset{\sim}{R}_A \left[{}^AP_{BORG}\right] & {}^B\underset{\sim}{R}_A \end{bmatrix}$$