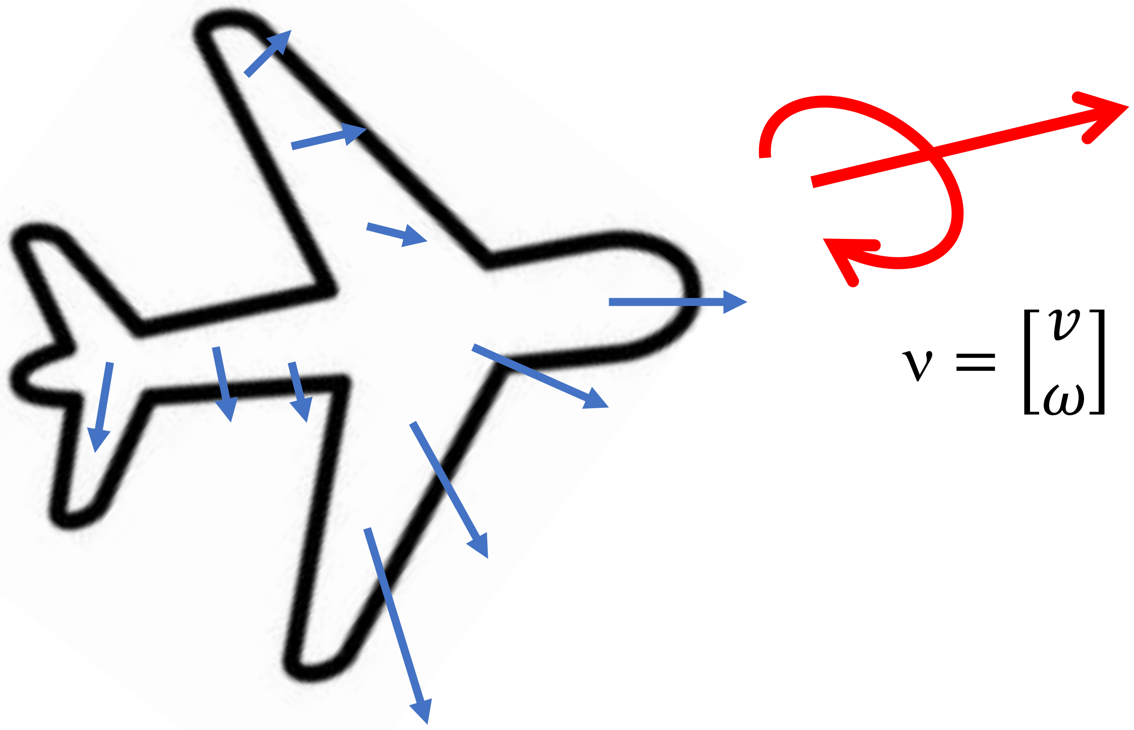
#2 Trajectory optimization

Nicolas Mansard (slide author)

Stéphane Caron, Justin Carpentier, Olivier Roussel, Guilhem Saurel

Velocity is a field

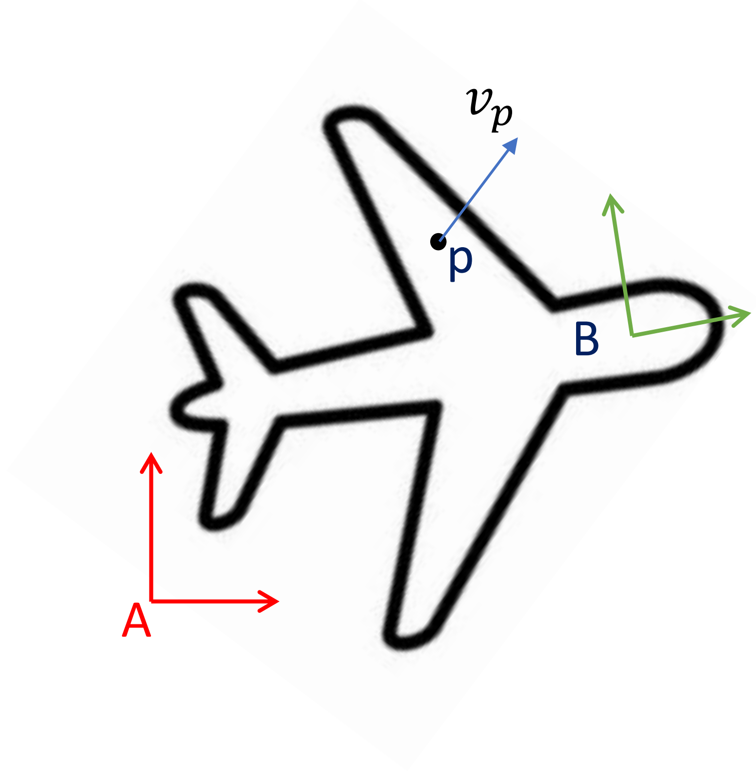


$$\nu = \begin{bmatrix} v \\ \omega \end{bmatrix}$$

- Vector field defined by

$$v_A = v_B + \overrightarrow{AB} \times \omega$$

Spatial velocity



- Following the derivations of angular velocity:

$$\dot{M} = \nu \times M$$

$$^{A}v_p = {}^{A}\nu \times {}^{A}p$$

The *spatial velocity* ν defines a vector field of linear velocities

- Spatial velocities are *transported* by SE(3)

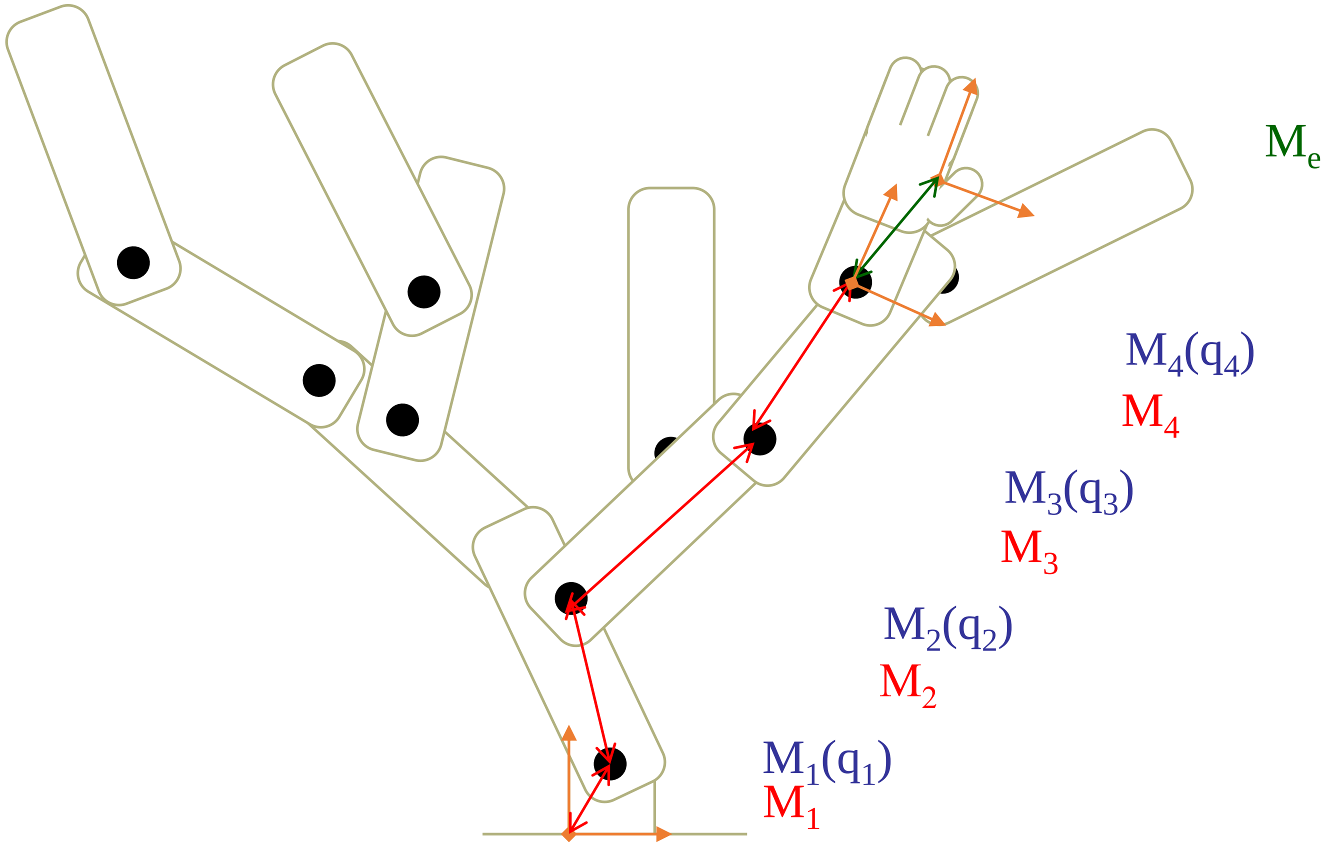
$$^{A}\nu = {}^{A}X_B \, {}^{B}\nu$$

$$^{A}X_B = \begin{pmatrix} {}^{A}R_B & {}^{A}R_B \, {}^{B}AB \times \\ 0 & {}^{A}R_B \end{pmatrix}$$

$$^{A}\nu_{A:C} = {}^{A}X_B \left({}^{B}\nu_{A:B} + {}^{B}X_C \, {}^{C}\nu_{B:C} \right)$$

Direct geometry

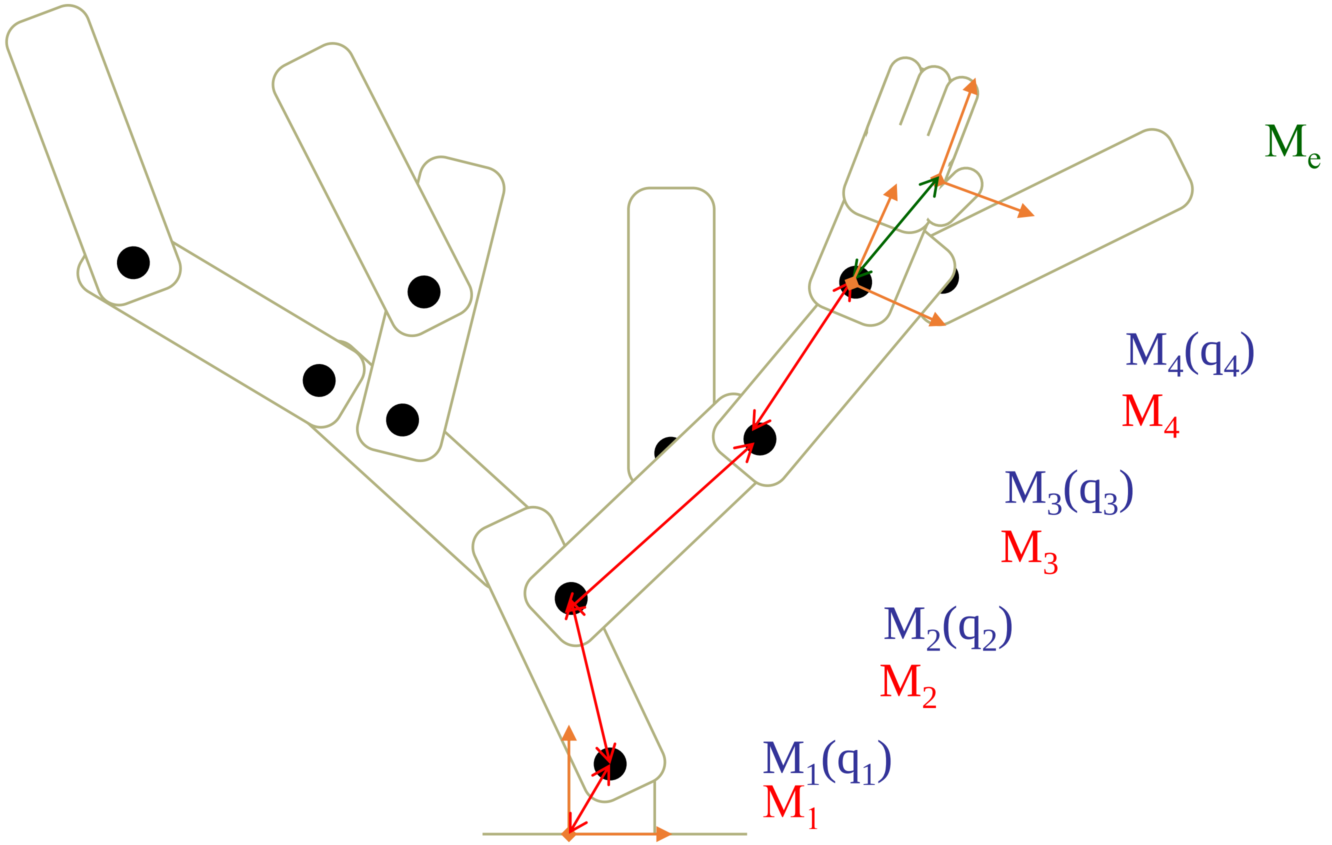
The geometric model is a tree of joints and bodies



$$M(q) = M_1 \oplus M_1(q_1) \oplus M_2 \oplus \ldots \oplus M_4 \oplus M_4(q_4) \oplus M_e$$

Direct (forward) kinematics

The geometric model is a tree of joints and bodies



$$\begin{aligned} \nu_E &= {}^0X_1\,{}^1\nu_1(q_1,v_{q1}) + {}^0X_2\,{}^2\nu_2(q_2,v_{q2}) + {}^0X_3\,{}^3\nu_3(q_3,v_{q3}) + \dots \\ &= J_1(q_1)\;v_{q1} + J_2(q_2)\;v_{q2} + J_3(q_3)\;v_{q3} + \dots \end{aligned}$$

Robot Jacobian

- Transform joint velocities into Cartesian velocities

$$\dot{p} = J_3\, v_q$$
$$\nu = J_6\, v_q$$

- If we know the reference velocity v^* we want to see…

Search v_q so that $v = J\, v_q = v^*$

$$\min_{v_q} \left\| J\, v_q - v^* \right\|^2$$

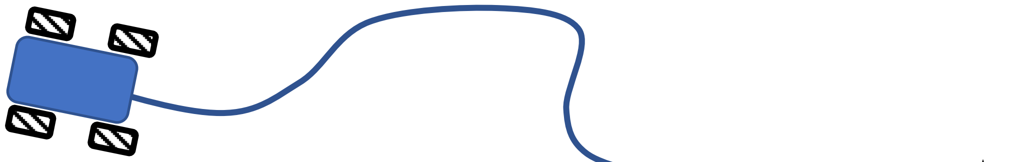
Recap

`pin.forwardKinematics(model,data,q,v,a)`	Evaluate the forward kinematics in position, velocity and acceleration
`data.v[i],data.a[i]`	Spatial velocities and accelerations of joints
`pin.LOCAL`	Representation in local (joint, or frame) coordinates
`pin.WORLD`	Representation in world coordinates
`pin.LWA`	Nonspatial representation "local world aligned"
`pin.getFrameVelocity`	Frame velocity from joint velocity data.v
`pin.getFrameAcceleration`	Frame **spatial** acceleration
`pin.getClassicalAcceleration`	Classical acceleration $[\ddot{p}, \dot{w}]$

Optimal control

$$\min_{X,U} \int_0^T l\big(x(t), u(t)\big)dt + l_T\big(x(T)\big)$$

$$\text{s.t.} \quad \dot{x}(t) = f(x(t),u(t))$$



- X and U are functions of t:

  $X: t \in \Re \rightarrow x(t) \in \Re^{\text{nx}}$

  $U: t \in \Re \rightarrow u(t) \in \Re^{\text{nu}}$

- The terminal time T is fixed

Problem definition

$$\min_{X,U} \sum_{t=0}^{T-1} l(x_t, u_t) + l_T(x_T)$$

$$\text{s.t.} \quad x_{t+1} = f(x_t, u_t)$$

- X and U are vectors of dimension $T.nx$ and $T.nu$ resp.

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \qquad U = \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}$$

- The information in X and U is somehow redundant