Indice

Formule Utili

Relazioni utili

$\dot{p} \rightleftarrows v$ $\quad \dot{p} = v$

$\dot{R} \rightleftarrows \omega$ $\quad \dot{R} = S(\omega)R \iff$ for each (unit) column r_i of R (a frame): $\dot{r}_i = \omega \times r_i$

$S(\omega) = \dot{R}R^T$

[in body frame ($\Omega = R^T\omega$): $\dot{R} = RS(\Omega)$, $S(\Omega) = R^T\dot{R} = R^T S(\omega) R$]

$\dot{\phi} \rightleftarrows \omega$

$$\omega = \omega_{\dot{\phi}_1} + \omega_{\dot{\phi}_2} + \omega_{\dot{\phi}_3} = a_1\dot{\phi}_1 + a_2(\phi_1)\dot{\phi}_2 + a_3(\phi_1, \phi_2)\dot{\phi}_3$$
$$= T(\phi)\,\dot{\phi}$$

(moving) axes of definition for the sequence of rotations ϕ_i, $i = 1,2,3$

---

special case: if the task vector r is

$$r = \begin{pmatrix} p \\ \phi \end{pmatrix} \Rightarrow J_r(q) = \begin{pmatrix} I & 0 \\ 0 & T^{-1}(\phi) \end{pmatrix} J(q) \iff J(q) = \begin{pmatrix} I & 0 \\ 0 & T(\phi) \end{pmatrix} J_r(q)$$

$J_r \rightleftarrows J$

$T(\phi)$ has always a singularity $\Leftrightarrow$ singularity of the specific minimal representation of orientation

P_dot e q_dot / P_dot_dot e q_dot_dot

velocity $\quad \dot{r} = \boxed{J_r(q)}\,\dot{q}$

matrix function $N_2(q, \dot{q})$

acceleration $\quad \ddot{r} = \boxed{J_r(q)}\,\ddot{q} + \dot{J}_r(q)\dot{q}$

matrix function $N_3(q, \dot{q}, \ddot{q})$

jerk $\quad \dddot{r} = \boxed{J_r(q)}\,\dddot{q} + 2\dot{J}_r(q)\ddot{q} + \ddot{J}_r(q)\dot{q}$

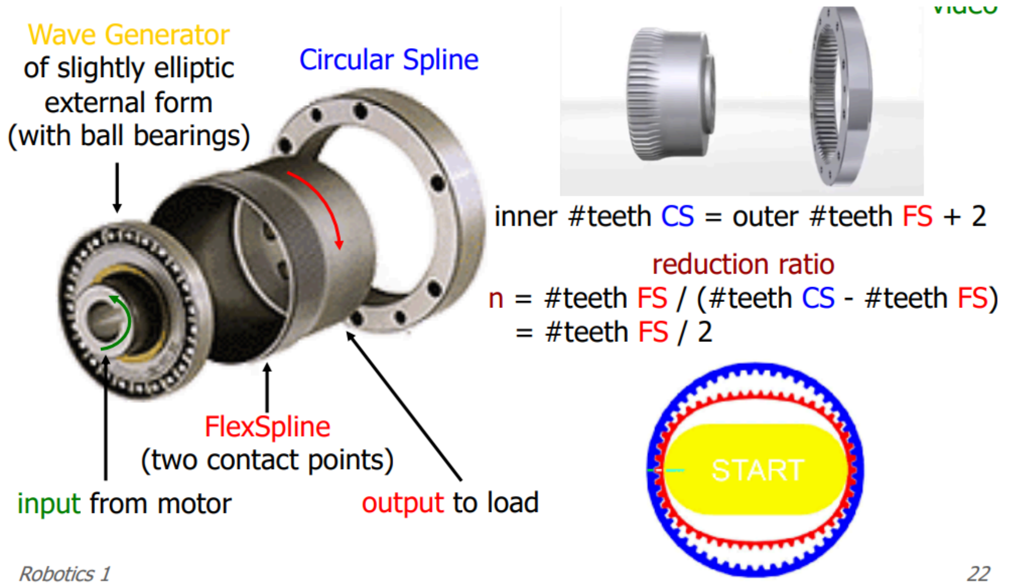
snap $\quad \ddddot{r} = \boxed{J_r(q)}\,\ddddot{q} + \cdots$

Elementi della DH

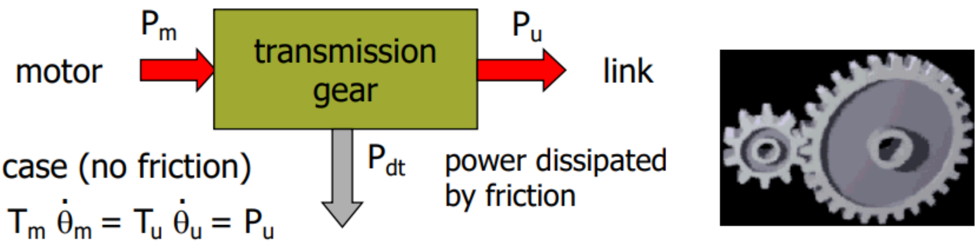
- a_i = distanza di z_{i-1} da z_i lungo x_i
- d_i = distanza di x_{i-1} da x_i lungo z_{i-1}
- α_i = angolo da z_{i-1} a z_i GUARDANDO ATTRAVERSO x_i
- θ_i = angolo da x_{i-1} a x_i GUARDANDO ATTRAVERSO z_{i-1}

Formule sensori

Harmonic Drives



Reduction Ratio



ideal case (no friction)

$P_m = T_m \dot{\theta}_m = T_u \dot{\theta}_u = P_u$

torque × angular speed

n = reduction ratio (≫1) $\qquad \dot{\theta}_m = n\,\dot{\theta}_u \Rightarrow T_u = n\,T_m$

to have $\ddot{\theta}_u = a$ (thus $\ddot{\theta}_m = n\,a$), the motor should provide a torque

$$T_m = J_m \ddot{\theta}_m + 1/n\,(J_u \ddot{\theta}_u) = (J_m\,n + J_u/n)\,a$$

inertia × angular acceleration

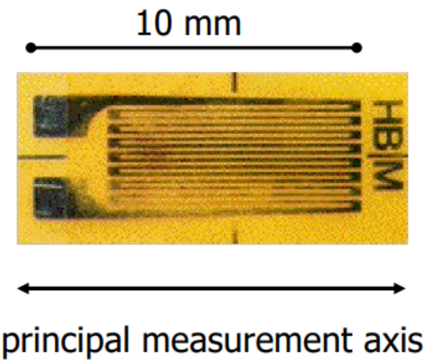
for minimizing T_m, we set: $\quad \dfrac{\partial T_m}{\partial n} = (J_m - J_u/n^2)\,a = 0$

$\Rightarrow \boxed{n = (J_u / J_m)^{1/2}}$ "matching" condition between inertias

Absolute encoders

$resolution = 360^{\circ} / 2^{N_t}$ -> negli esami al posto di 360 qualche volta ha dato lui il valore

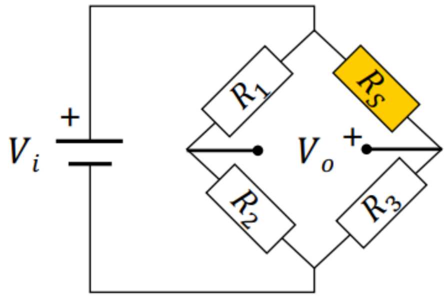
Strain gauges



Gauge-Factor = GF = $\dfrac{\Delta R/R}{\Delta L/L}$ ← strain ε

(typically GF ≈ 2, i.e., small sensitivity)

Wheatstone single-point quarter-bridge
(for accurately measuring resistance)

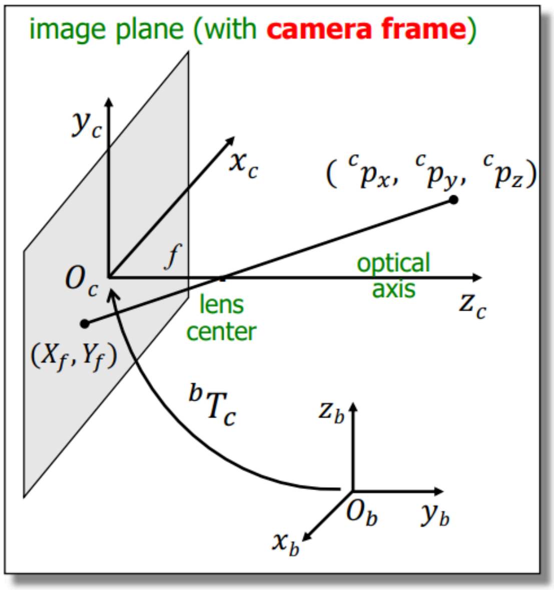


- R_1, R_2, R_3 very well matched ($\approx R$)
- $R_S \approx R$ at rest (no stress)
- two-point bridges have 2 strain gauges connected oppositely (↗ sensitivity)

if R_1 has the same dependence on T of R_S thermal variations are automatically compensated

$$V_0 = \left(\frac{R_3}{R_3 + R_S} - \frac{R_2}{R_1 + R_2}\right) V_i$$

Camera



1. in metric units $\quad X_f = \dfrac{f\,{}^c p_x}{f - {}^c p_z} \quad Y_f = \dfrac{f\,{}^c p_y}{f - {}^c p_z}$

2. in pixel

$$X_I = \frac{\alpha_x f\,{}^c p_x}{f - {}^c p_z} + X_0 \qquad Y_I = \frac{\alpha_y f\,{}^c p_y}{f - {}^c p_z} + Y_0$$

offsets of pixel coordinate system w.r.t. optical axis

pixel/metric scaling factor

3. LINEAR MAP in homogeneous coordinates

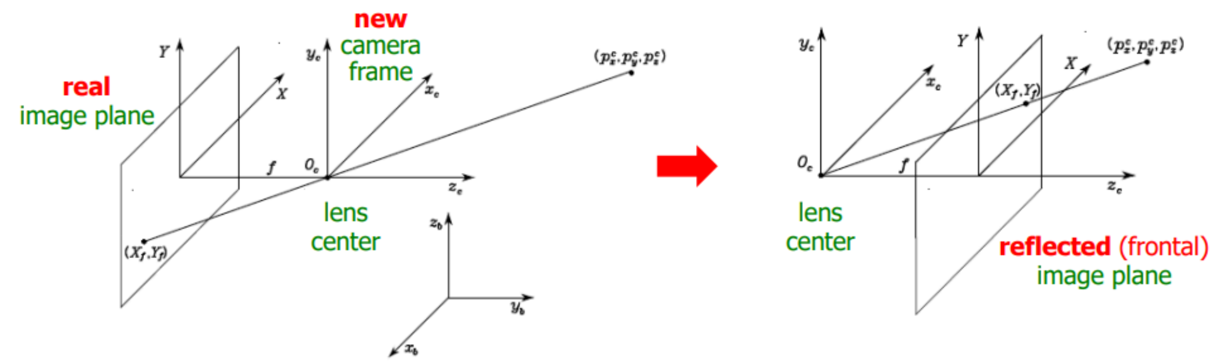
$$X_I = \frac{x_I}{z_I} \quad Y_I = \frac{y_I}{z_I} \quad \Rightarrow \quad \begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} = \Omega \begin{bmatrix} {}^c p_x \\ {}^c p_y \\ {}^c p_z \\ 1 \end{bmatrix}$$

$$\Omega = \begin{bmatrix} \alpha_x & 0 & X_0 & 0 \\ 0 & \alpha_y & Y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1/f & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

calibration matrix

$$H = \Omega \cdot {}^c T_b$$

intrinsic and extrinsic parameters



1. in metric units $X_f = -\dfrac{f\ {}^c p_x}{{}^c p_z} \quad Y_f = -\dfrac{f\ {}^c p_y}{{}^c p_z}$ ➡ $X_f = \dfrac{f\ {}^c p_x}{{}^c p_z} \quad Y_f = \dfrac{f\ {}^c p_y}{{}^c p_z}$

2. in pixel ... ➡ $X_I = \dfrac{\alpha_x f\ {}^c p_x}{{}^c p_z} + X_0 \quad Y_I = \dfrac{\alpha_y f\ {}^c p_y}{{}^c p_z} + Y_0$

3. LINEAR MAP in homogeneous coordinates ... ➡ $\begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} = \Omega \begin{bmatrix} {}^c p_x \\ {}^c p_y \\ {}^c p_z \\ 1 \end{bmatrix} \quad \Omega = \begin{bmatrix} \alpha_x f & 0 & X_0 & 0 \\ 0 & \alpha_y f & Y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

Orientation and Position

Skew-symmetric matrix

- canonical form of a 3×3 skew-symmetric matrix

also called vee map ∨

$v = S^{\vee}$

$$\boldsymbol{v} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow S(\boldsymbol{v}) = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \qquad S = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix} \Rightarrow \boldsymbol{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$

Axis/angle: Direct problem

$$\boxed{R(\theta, \boldsymbol{r}) = C\, R_z(\theta)\, C^T}$$

$$R(\theta, \boldsymbol{r}) = \begin{bmatrix} \boldsymbol{n} & \boldsymbol{s} & \boldsymbol{r} \end{bmatrix} \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{n}^T \\ \boldsymbol{s}^T \\ \boldsymbol{r}^T \end{bmatrix}$$

$$= \boldsymbol{r}\boldsymbol{r}^T + (\boldsymbol{n}\boldsymbol{n}^T + \boldsymbol{s}\boldsymbol{s}^T)\, c\theta + (\boldsymbol{s}\boldsymbol{n}^T - \boldsymbol{n}\boldsymbol{s}^T)\, s\theta$$

taking into account

$$CC^T = \boldsymbol{n}\boldsymbol{n}^T + \boldsymbol{s}\boldsymbol{s}^T + \boldsymbol{r}\boldsymbol{r}^T = I$$

$$\boldsymbol{s}\boldsymbol{n}^T - \boldsymbol{n}\boldsymbol{s}^T = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} = S(\boldsymbol{r})$$

depends only on $\boldsymbol{r}$ and θ ! →

$$\boxed{R(\theta, \boldsymbol{r}) = \boldsymbol{r}\boldsymbol{r}^T + (I - \boldsymbol{r}\boldsymbol{r}^T)\, c\theta + S(\boldsymbol{r})\, s\theta}$$

developing computations...

$$R(\theta,\boldsymbol{r}) =$$

$$\begin{bmatrix} r_x^2(1-\cos\theta)+\cos\theta & r_x r_y(1-\cos\theta)-r_z\sin\theta & r_x r_z(1-\cos\theta)+r_y\sin\theta \\ r_x r_y(1-\cos\theta)+r_z\sin\theta & r_y^2(1-\cos\theta)+\cos\theta & r_y r_z(1-\cos\theta)-r_x\sin\theta \\ r_x r_z(1-\cos\theta)-r_y\sin\theta & r_y r_z(1-\cos\theta)+r_x\sin\theta & r_z^2(1-\cos\theta)+\cos\theta \end{bmatrix}$$

note that

sum of the diagonal elements of a matrix

$$\text{trace}\, R(\theta,\boldsymbol{r}) = 1 + 2\cos\theta$$

$$R(\theta,\boldsymbol{r}) = R(-\theta,-\boldsymbol{r}) = R^T(-\theta,\boldsymbol{r})$$

Axis/angle: Inverse problem

from the data $\longleftrightarrow$ from $R(\theta,\boldsymbol{r})$

$$R - R^T = \begin{bmatrix} 0 & R_{12}-R_{21} & R_{13}-R_{31} \\ R_{21}-R_{12} & 0 & R_{23}-R_{32} \\ R_{31}-R_{13} & R_{32}-R_{23} & 0 \end{bmatrix} = 2\sin\theta \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}$$

it follows

$$\|\boldsymbol{r}\| = 1 \;\Rightarrow\; \sin\theta = \pm\frac{1}{2}\sqrt{(R_{12}-R_{21})^2+(R_{13}-R_{31})^2+(R_{23}-R_{32})^2} \quad (*)$$

thus

$$\boxed{\theta = \text{atan2}\left\{\pm\sqrt{(R_{12}-R_{21})^2+(R_{13}-R_{31})^2+(R_{23}-R_{32})^2},\, R_{11}+R_{22}+R_{33}-1\right\}} \quad (**)$$

see the slide with its definition!

$$\boxed{\boldsymbol{r} = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \frac{1}{2\sin\theta}\begin{bmatrix} R_{32}-R_{23} \\ R_{13}-R_{31} \\ R_{21}-R_{12} \end{bmatrix}} \longleftarrow$$

can be used only if $\boxed{\sin\theta \neq 0}$

test is made on (*) using the data $\{R_{ij}\}$

Gradient Method

see Matlab for code

- Gradient method (max descent)
  - minimize the error function

$$H(q) = \frac{1}{2}\|r_d - f_r(q)\|^2 = \frac{1}{2}(r_d - f_r(q))^T(r_d - f_r(q))$$

$$q^{k+1} = q^k - \alpha\,\nabla_q H(q^k)$$

from

$$\nabla_q H(q) = (\partial H(q)/\partial q)^T = -\left((r_d - f_r(q))^T(\partial f_r(q)/\partial q)\right)^T = -J_r^T(q)(r_d - f_r(q))$$

we get

$$\boxed{q^{k+1} = q^k + \alpha\, J_r^T(q^k)\big(r_d - f_r(q^k)\big)}$$

  - the scalar step size $\alpha > 0$ should be chosen so as to guarantee a decrease of the error function at each iteration: too large values for α may lead the method to "miss" the minimum
  - when the step size is too small, convergence is extremely slow

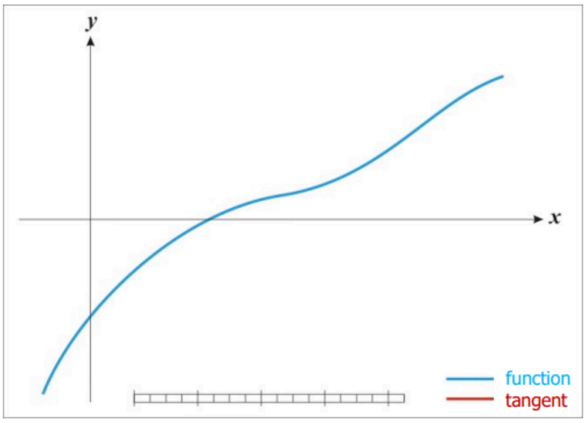
Newton method

- in the scalar case, also known as "method of the tangent"
- for a differentiable function $f(x)$, find a root x^* of $f(x^*) = 0$ by iterating as

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

an approximating sequence

$$\{x_1, x_2, x_3, x_4, x_5, \cdots\} \longrightarrow x^*$$



Time derivative of Rotation Matrix

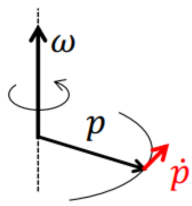
- let $R = R(t)$ be a rotation matrix, given as a function of time
- since $I = R(t)R^T(t)$, taking the time derivative of both sides yields

$$0 = d\big(R(t)R^T(t)\big)/dt = (dR(t)/dt)R^T(t) + R(t)(dR^T(t)/dt)$$
$$= (dR(t)/dt)R^T(t) + \big((dR(t)/dt)\, R^T(t)\big)^T$$

  thus $(dR(t)/dt)\, R^T(t) = S(t)$ is a skew-symmetric matrix
- let $p(t) = R(t)p'$ a vector (with constant norm) rotated over time
- comparing

$$\dot{p}(t) = (dR(t)/dt)p' = S(t)R(t)p' = S(t)p(t)$$
$$\dot{p}(t) = \omega(t) \times p(t) = S\big(\omega(t)\big)p(t)$$

  we get $S = S(\omega)$



$$\boxed{\dot{R} = S(\omega)R} \quad \Longleftrightarrow \quad \boxed{S(\omega) = \dot{R}\, R^T}$$

Squaring and Summing

direct kinematics

$$p_x = l_1 c_1 + l_2 c_{12}$$
$$p_y = l_1 s_1 + l_2 s_{12}$$

"squaring and summing" the equations of the direct kinematics

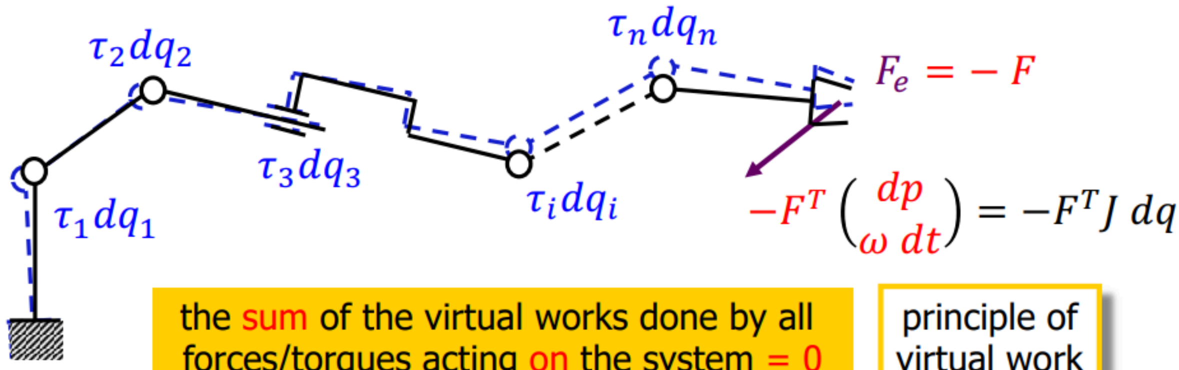
$$p_x^2 + p_y^2 - (l_1^2 + l_2^2) = 2 l_1 l_2 (c_1 c_{12} + s_1 s_{12}) = 2 l_1 l_2 c_2$$

and from this

$$c_2 = \big(p_x^2 + p_y^2 - (l_1^2 + l_2^2)\big)/2 l_1 l_2, \quad s_2 = \pm\sqrt{1 - c_2^2} \qquad q_2 = \mathrm{atan2}\{s_2, c_2\}$$

Statics Force

IMPORTANTE METTERE -F



the sum of the virtual works done by all forces/torques acting on the system = 0

principle of virtual work

$$\tau^T dq - F^T \begin{pmatrix} dp \\ \omega dt \end{pmatrix} = \tau^T dq - F^T J dq = 0 \qquad \boxed{\forall dq}$$

$$\boxed{\tau = J^T(q) F}$$

Trajectory

TRAJECTORY $\begin{cases} \text{of motion } p_d(t) \text{ (or } q_d(t)) \\ \text{of interaction } F_d(t) \end{cases}$

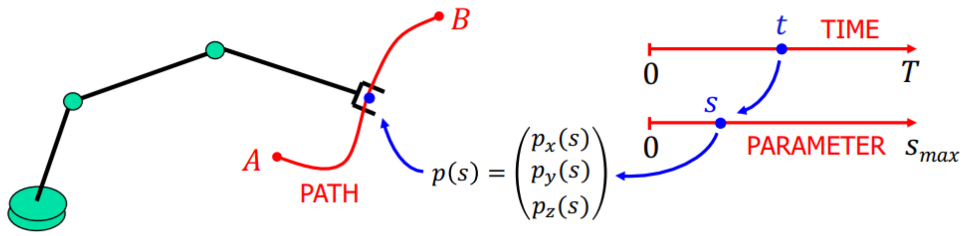
||

GEOMETRIC PATH parameterized by s: $p = p(s)$ (e.g., s is the arc length)

+

TIMING LAW describes the time evolution of $s = s(t)$

$\Big\}\ p(s(t))$



example: TASK planner provides A, B
TRAJECTORY planner generates $p(t)$

Robotics 1 8

Cubic Polynomial in space

$\boxed{q(0) = q_0}$ $\boxed{q(1) = q_1}$ $\boxed{q'(0) = v_0}$ $\boxed{q'(1) = v_1}$ ← 4 conditions

$$q(\lambda) = q_0 + \Delta q\underbrace{(a\lambda^3 + b\lambda^2 + c\lambda + d)}$$

$\Delta q = q_1 - q_0$

$\lambda \in [0,1]$

4 coefficients → "doubly normalized" polynomial $q_N(\lambda)$

$q_N(0) = 0 \Leftrightarrow d = 0$ $\quad q_N(1) = 1 \Leftrightarrow a + b + c = 1$

$q_N'(0) = dq_N/d\lambda|_{\lambda=0} = c = v_0/\Delta q$ $\quad q_N'(1) = dq_N/d\lambda|_{\lambda=1} = 3a + 2b + c = v_1/\Delta q$

special case: $v_0 = v_1 = 0$ (zero tangent)

$$q_N'(0) = 0 \Leftrightarrow c = 0$$

$$\left.\begin{aligned} q_N(1) = 1 &\Leftrightarrow a + b = 1 \\ q_N'(1) = 0 &\Leftrightarrow 3a + 2b = 0 \end{aligned}\right\} \Leftrightarrow \begin{aligned} a &= -2 \\ b &= 3 \end{aligned}$$

$\boxed{q(0) = q_{in}}$ $\boxed{q(T) = q_{fin}}$ $\boxed{\dot{q}(0) = v_{in}}$ $\boxed{\dot{q}(T) = v_{fin}}$ ← 4 conditions

$$q(\tau) = q_{in} + \Delta q\underbrace{(a\tau^3 + b\tau^2 + c\tau + d)}$$

$\Delta q = q_{fin} - q_{in}$

$\tau = t/T \in [0,1]$

4 coefficients → "doubly normalized" polynomial $q_N(\tau)$

$q_N(0) = 0 \Leftrightarrow d = 0$ $\quad q_N(1) = 1 \Leftrightarrow a + b + c = 1$

$q_N'(0) = dq_N/d\tau|_{\tau=0} = c = \dfrac{v_{in}T}{\Delta q}$ $\quad q_N'(1) = dq_N/d\tau|_{\tau=1} = 3a + 2b + c = \dfrac{v_{fin}T}{\Delta q}$

special case: $v_{in} = v_{fin} = 0$ (rest-to-rest)

$$q_N'(0) = 0 \Leftrightarrow c = 0$$

$$\left.\begin{aligned} q_N(1) = 1 &\Leftrightarrow a + b = 1 \\ q_N'(1) = 0 &\Leftrightarrow 3a + 2b = 0 \end{aligned}\right\} \Leftrightarrow \begin{aligned} a &= -2 \\ b &= 3 \end{aligned}$$

$\boxed{q(0) = q_{in}}$ $\boxed{q(T) = q_{fin}}$ $\boxed{\dot{q}(0) = 0}$ $\boxed{\dot{q}(T) = 0}$ ← boundary conditions (rest-to-rest)

$$q(\tau) = q_{in} + \Delta q \underbrace{\frac{1 - \cos \pi\tau}{2}}$$

$\Delta q = q_{fin} - q_{in}$

$\tau = t/T \in [0,1]$

doubly normalized

$$\dot{q}(\tau) = \frac{\Delta q}{T}\frac{\pi}{2}\sin \pi\tau \qquad \ddot{q}(\tau) = \frac{\Delta q}{T^2}\frac{\pi^2}{2}\cos \pi\tau$$

Quintic polynomial

$$q(\tau) = a\tau^5 + b\tau^4 + c\tau^3 + d\tau^2 + e\tau + f$$ 6 coefficients

$$\tau \in [0, 1]$$

allows to satisfy 6 conditions, for example (in normalized time $\tau = t/T$)

$$\boxed{q(0) = q_0} \quad \boxed{q(1) = q_1} \quad \boxed{q'(0) = v_0 T} \quad \boxed{q'(1) = v_1 T} \quad \boxed{q''(0) = a_0 T^2} \quad \boxed{q''(1) = a_1 T^2}$$

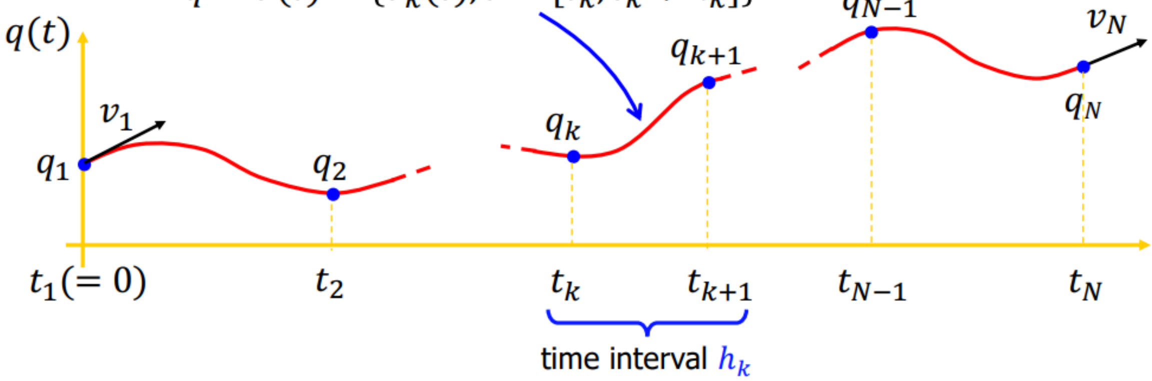
$$\begin{aligned} q(\tau) = {} & (1-\tau)^3 (q_0 + (3q_0 + v_0 T)\tau + (a_0 T^2 + 6 v_0 T + 12 q_0)\tau^2/2) \\ & + \tau^3 (q_1 + (3q_1 - v_1 T)(1-\tau) + (a_1 T^2 - 6 v_1 T + 12 q_1)(1-\tau)^2/2) \end{aligned}$$

special case: $v_0 = v_1 = a_0 = a_1 = 0$

$$q(\tau) = q_0 + \Delta q (6\tau^5 - 15\tau^4 + 10\tau^3) \qquad \Delta q = q_1 - q_0$$

Cubic Spline

$$q = \theta(t) = \{\theta_k(t), t \in [t_k, t_k + h_k]\}$$



$$\theta_k(\tau) = a_{k0} + a_{k1}\tau + a_{k2}\tau^2 + a_{k3}\tau^3 \qquad \tau = t - t_k \in [0, h_k]$$

$$(k = 1, \cdots, N-1)$$

continuity conditions for velocity and acceleration →

$$\boxed{\begin{aligned} \dot{\theta}_k(h_k) &= \dot{\theta}_{k+1}(0) \\ \ddot{\theta}_k(h_k) &= \ddot{\theta}_{k+1}(0) \end{aligned} \qquad k = 1, \cdots, N-2}$$

1. if all velocities v_k at internal knots were known, then each cubic in the spline would be uniquely determined by

$$\theta_k(0) = q_k = a_{k0} \qquad \dot{\theta}_k(0) = v_k = a_{k1} \qquad \begin{pmatrix} h_k^2 & h_k^3 \\ 2h_k & 3h_k^2 \end{pmatrix} \begin{pmatrix} a_{k2} \\ a_{k3} \end{pmatrix} = \begin{pmatrix} q_{k+1} - q_k - v_k h_k \\ v_{k+1} - v_k \end{pmatrix} \quad (1)$$

2. impose the continuity for accelerations ($N-2$ conditions)

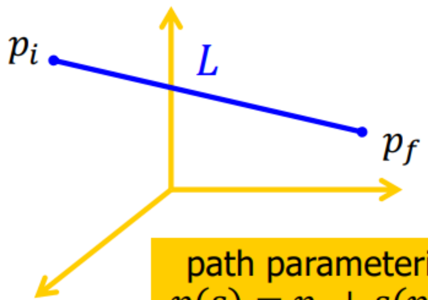
$$\ddot{\theta}_k(h_k) = 2a_{k2} + 6a_{k3}h_k = 2a_{k+1,2} = \ddot{\theta}_{k+1}(0)$$

3. expressing the coefficients $a_{k2}, a_{k3}, a_{k+1,2}$ in terms of the still unknown knot velocities (see step 1.) yields a linear system of equations that is always solvable

$$\underbrace{\Big(\; \boldsymbol{A}(h_1, \cdots, h_{N-1}) \; \Big)}_{\text{tri-diagonal matrix always invertible}} \underbrace{\begin{pmatrix} v_2 \\ v_3 \\ \vdots \\ \vdots \\ v_{N-1} \end{pmatrix}}_{\text{unknown}} = \underbrace{\begin{pmatrix} \vdots \\ \boldsymbol{b}(h_1, \cdots, h_{N-1}, q_1 \cdots, q_N, v_1, v_N) \\ \vdots \end{pmatrix}}_{\text{known vector}}$$

to be substituted then back in (1)

Planning a linear Cartesian path



GIVEN
$p_i, p_f \in \mathbb{R}^3$; $v_i, v_f \in \mathbb{R}$ (typically $= 0$);
bounds $v_{max}, a_{max} \in \mathbb{R}^+$

$$L = \|p_f - p_i\|$$

$$\frac{p_f - p_i}{\|p_f - p_i\|} = \text{unit vector of directional cosines of the line}$$

path parameterization
$$p(s) = p_i + s(p_f - p_i)$$

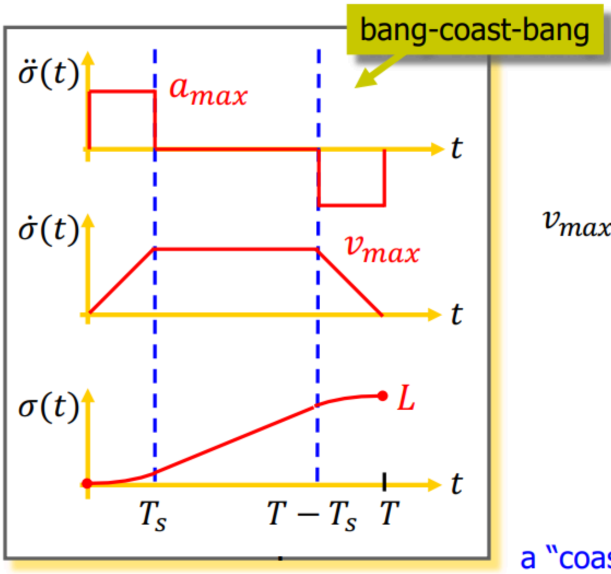
$s \in [0,1]$

may also use $s = \sigma/L$, where $\sigma \in [0, L]$ is the arc length (gives the current length of the path)

$$\dot{p}(s) = \frac{dp}{ds}\dot{s} = (p_f - p_i)\,\dot{s} = \frac{p_f - p_i}{L}\,\dot{\sigma}$$

$$\ddot{p}(s) = \cancel{\frac{d^2p}{ds^2}}\dot{s}^2 + \frac{dp}{ds}\ddot{s} = (p_f - p_i)\,\ddot{s} = \frac{p_f - p_i}{L}\,\ddot{\sigma}$$

Bang-Coast-Bang



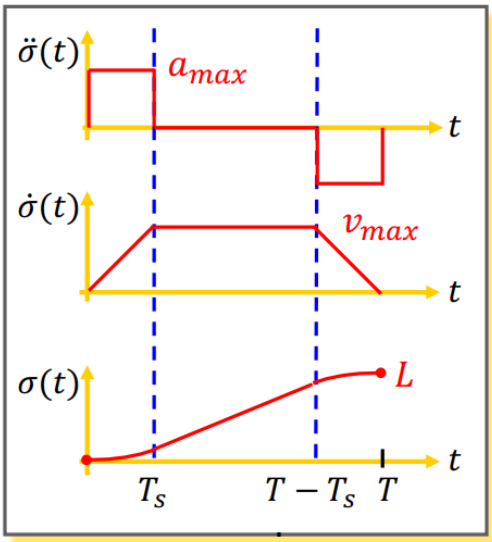
given*: L, v_{max}, a_{max}
find: T_s, T

$$v_{max}\,(T - T_s) = L \leftarrow$$ = area of the speed profile

$$T_s = \frac{v_{max}}{a_{max}}$$

$$T = \frac{L a_{max} + v_{max}^2}{a_{max} v_{max}}$$

a "coast" phase exists iff $L > v_{max}^2 / a_{max}$

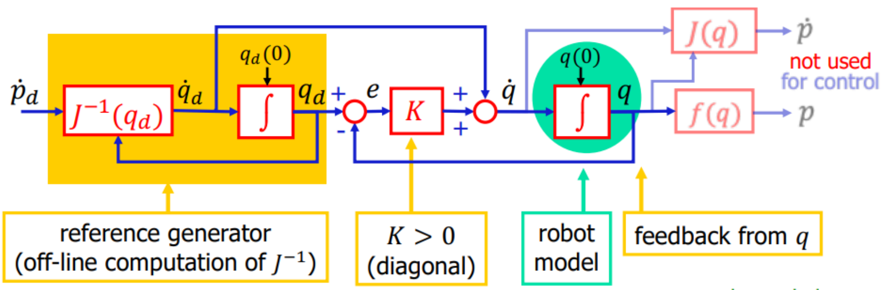


$$\sigma(t) = \begin{cases} \dfrac{a_{max} t^2}{2}, & t \in [0, T_s] \\ v_{max} t - \dfrac{v_{max}^2}{2\, a_{max}}, & t \in [T_s, T - T_s] \\ -\dfrac{a_{max}(t-T)^2}{2} + v_{max} T - \dfrac{v_{max}^2}{a_{max}}, & t \in [T - T_s, T] \end{cases}$$

discontinuous acceleration profile!
if needed, use for instance a
a rest-to-rest quintic polynomial timing

can be used also in the joint space!

Kinematic control of joint motion



$$e = q_d - q \Rightarrow \dot{e} = \dot{q}_d - \dot{q} = \dot{q}_d - (\dot{q}_d + K(q_d - q)) = -Ke$$

decoupled $e_i \to 0$
$(i = 1, \cdots, n)$
exponentially,
$\forall e(0)$

$$e_p = p_d - p \Rightarrow \dot{e}_p = \dot{p}_d - \dot{p} = J(q_d)\dot{q}_d - J(q)(\dot{q}_d + K(q_d - q))$$

$$q \approx q_d, \quad e_p \to J(q)e \Rightarrow \dot{e}_p \approx -J(q) K J^{-1}(q) e_p$$

coupled Cartesian error dynamics

Cartesian kinematic control

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q})\Big(\dot{\boldsymbol{p}}_d + \boldsymbol{K}_P\left(\boldsymbol{p}_d - \boldsymbol{p}(\boldsymbol{q})\right)\Big), \qquad \text{with } \boldsymbol{K}_P = k_P \cdot \boldsymbol{I}_{2\times 2} > 0,$$