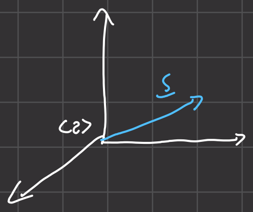
Quadratic programming for task priority

26/09/24

Time derivative of vectors

Time derivative of a projected vector on a given frame.



$$ {}^a\dot{s}(t) \triangleq \frac{d_a}{dt} s(t) \triangleq D_a\, s(t) $$

notations

The differentiation happens after the projection

$$ {}^a s(t) = \int_0^t {}^a\dot{s}(\tau)\, d\tau + {}^a s(0) $$

Higher order derivatives

$$ {}^a s(t+dt) = {}^a s(t) + {}^a\dot{s}(t)\, dt + O(dt) $$

Approximation of the state.

Note that you can't do the diff with an observer in b and then integrate on another frame.

Change of differentiation frame.

$$ {}^b\dot{s}(t) \neq {}^b_aR\; {}^a\dot{s}(t) \qquad \text{N.B.} $$

$$ {}^b\left({}^a\dot{s}(t)\right) = {}^b_aR\; {}^a\dot{s}(t) $$

The mother of all the formulas

$$ {}^a\dot{s}(t) = D\, {}^a s(t) = D\left({}^a_bR\; {}^b s(t)\right) = $$

$$ = D\left({}^a_bR\right) {}^b s(t) + {}^a_bR\, D\left({}^b s(t)\right) = $$

$$ = {}^a_b\dot{R}\; {}^b s(t) + {}^a_bR\; {}^b\dot{s}(t) = $$

$$ {}^a\dot{s}(t) = \begin{cases} {}^a_bR\; {}^b\dot{s}(t) + {}^a_bR \left[{}^b\omega_{b/a}\times\right] {}^a s(t) \\ {}^a_bR\; {}^b\dot{s}(t) + \left[{}^a\omega_{b/a}\times\right] {}^a_bR\; {}^b s(t) \end{cases} $$

Algebraic formula of diff frame

$$ \boxed{D_a\, s = D_b\, s + \omega_{b/a} \times s} $$

Time derivative of constant module vectors

$|s| = \sigma > 0$ and $\dot{\sigma} = \emptyset$

We want to differentiate w.r.t. $\langle a \rangle$

Consider another frame $\langle h \rangle$ such that:

$$^{h}\dot{s} = \emptyset$$

This means that I have attached a frame to the vector s which is rotating but not changing in modulus.

If now apply the mother of all the formulas:

$$\underbrace{D_a s = \omega_{h/a} \times s}_{\text{let's express this like } \omega_{h/a} \text{ where}} \quad \forall \langle h \rangle \text{ s.t. } D_h s = \emptyset$$

$$\omega_{h/a} = \omega_{s/a} + \hat{n} z \quad \forall z \in \mathbb{R}$$

↳ unity vector of s

The minimum norm solution is:

$D_a s = \omega_{s/a} \times s$ Note without nz which creates the span of other vector.

admittance / impedance control

Time derivative of generic vector

$s = n\sigma$ $\quad |\underline{n}| = 1$ (constant modulus vector) $\quad \sigma > \emptyset$

$$D_a s = D_a(n\sigma) = \underbrace{n\dot{\sigma}}_{\text{aligned to } s} + \underbrace{\sigma D_a(n)}_{\text{ortogonal to } s}$$

$$D_a \underline{s} = n\underbrace{(n\cdot)\, D_a s}_{\text{this scalar must be } \dot{\sigma}} + (1 - n(n\cdot))\, D_a s$$

↳ this states: $\sigma D_a n = (1 - n(n\cdot))\, D_a s$

$$\begin{cases} \dot{\sigma} = n \cdot D_a s \\ \sigma D_a n = (1 - n(n\cdot))\, D_a s \end{cases}$$

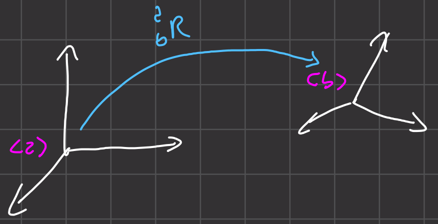
Recalling that:

$D_a(n) = \omega_{n/a} \times n$

then substituting

$$\boxed{D_a(\underline{\sigma}) = n\dot{\sigma} + \sigma\,\omega_{n/a} \times \underline{n} = n\dot{\sigma} + \omega_{s/a} \times \underline{n}}$$

Derivative of rotation vector



$\rho \triangleq \rho_{b/a}$ $\rho \triangleq \underline{n}\,\theta$

$\omega_{b/a}$

$D_a\,\rho = \underline{n}\,\dot{\theta} + \theta\, D_a\,\underline{n}$

write <a> and not <b> because the rotation vector is the axis of rotation, so common

$$\boxed{{}^aD_a\,\underline{\rho} = {}^bD_b\,\underline{\rho}}$$

$$D_a\,\rho = [\underbrace{n(n\cdot)}_{\text{aligned}} + \underbrace{N_a(\theta)}_{\text{orthogonal}}]\,\underline{\omega}_{b/a}$$

$$N_a(\theta) \triangleq \frac{\theta}{2}\left[\frac{1}{\tan(\frac{\theta}{2})} - [n\times]\right]\underline{[I - n(n\cdot)]}$$

↳ because of this necessarily orthogonal to n

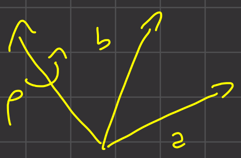
$\dot{\rho} = -\lambda\rho$ → desired behaviour to reach the goal

ω has to be aligned with ρ why (125:38)

If you take an ω with elements not aligned with n you also make the rotation vector rotate.

3/10/24

Time derivative of rot vec. between two vectors



$\rho = \underline{\hat{n}}\,\theta$

$D_a\,\rho = \underline{n}\,\dot{\theta} + \theta\, D_a(\underline{n})$

↳ observer

① $\underline{a} \times \underline{b} = \underline{n}\,\sin(\theta)$ and ② $a \cdot b = \underline{n}\,\cos(\theta)$ → remember this

Let's now derive ②

$$\underline{a} \cdot (\omega_{b/\alpha} \times b) + b \cdot (\omega_{a/\alpha} \times a) = -\sin(\theta)\,\dot{\theta}$$

Using the mixed product among vectors:

$$\underline{a} \cdot (\underline{b} \times \underline{c}) = \underline{b} \cdot (\underline{c} \times \underline{a}) = \underline{c} \cdot (\underline{a} \times \underline{b})$$

then we get:

$$\omega_{b/\alpha} \cdot (\underline{b} \times \underline{a}) + \omega_{a/\alpha} \cdot (\underline{a} \times \underline{b}) = -\sin(\theta)\,\dot{\theta}$$

$$-\omega_{b/\alpha} \cdot (a \times b) + \omega_{a/\alpha} \cdot (a \times b) = -\sin(\theta)\,\dot{\theta}$$

$$\left(\omega_{a/\alpha} - \omega_{b/\alpha}\right)(a \times b) = -\sin(\theta)\,\dot{\theta}$$

$$\omega_{b/a}\,(a \times b) = \sin(\theta)\,\dot{\theta}$$

$a \times \underline{b} = \underline{n}\,\sin(\theta)$ using this we get:

$$\dot{\theta} = \underline{\omega_{b/a}} \cdot \underline{n}$$

$$D_\alpha(n) = D_\alpha\left(\frac{1}{\sin(\theta)}\, a \times b\right) =$$

$$= N(\theta)\,\omega_{b/\alpha} + M(\theta)\,\omega_{a/\alpha}$$ M and N in the notes.

Hence

$$D_\alpha \rho = n\dot{\theta} + \theta\, D_\alpha(n) = n(n\cdot)\,\omega_{b/a} + \theta N(\theta)\,\omega_{b/\alpha} + \theta M(\theta)\,\omega_{a/\alpha}$$

If $\alpha = a$ So we choose an observer where a is constant

$$D_a \rho = \underbrace{n(n\cdot)\,\omega_{b/a}}_{\text{this makes } \rho \text{ smaller or larger}} + \underbrace{\theta N(\theta)\,\omega_{b/a}}_{\text{makes } \rho \text{ rotate.}}$$ (Same simplification if you sit on b)

We want ρ to grow smaller, we don't really need to make ρ rotate.

Time derivative of points

$P(t)$ (point)

$$\underline{v}_{P/a} \triangleq D_a\, \underline{r}_{P/a} \triangleq \frac{d_a}{dt}\, \underline{r}_{P/a}$$

$$\underline{r}_{P/a} \triangleq (P - O_a)$$

$${}^a v_{P/a} = {}^a\dot{P}$$

$${}^a P(t+dt) = {}^a P(t) + {}^a\dot{P}(t)\,dt + O(dt)$$

Composition of linear velocity vectors

$$\underline{r}_{P/a} = \underline{r}_{P/b} + \underline{r}_{b/a}$$

$$D_a(r_{P/a}) = D_a(r_{P/b} + r_{b/a})$$

$$v_{P/a} = D_b(r_{P/b}) + \omega_{b/a} \times r_{P/b} + v_{b/a}$$

$$v_{P/a} = v_{P/b} + v_{b/a} + \underline{\omega_{b/a} \times r_{P/b}}$$

If the frames do not rotate wrt to each other this is $\emptyset$

Points attached to a rigid space (tool/ee)

$P \in$ rigid space of <b>

$$v_{P/a} = v_{b/a} + \omega_{b/a} \times r_{P/b}$$

The only difference is that $v_{P/b} = \emptyset$

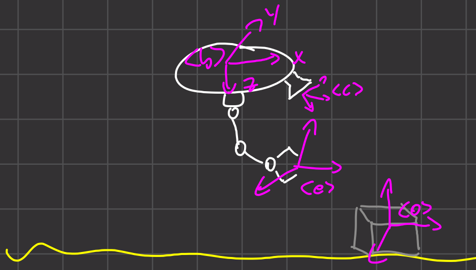
Time derivative of distance vector

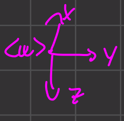
$$\underline{s} = (P - Q)$$

$$D_a \underline{s} \triangleq v_{s/a} = D_a P - D_a Q = v_{P/a} - v_{Q/a}$$

Single agent control through task priority approach

Consider an underwater vehicle with a manipulator





(NED frame)
↓
North East Down

The number of degrees of freedom of the robot are:

- 6 base
- (4-6) arm

Configuration vector: $c \triangleq \begin{bmatrix} q \\ \eta \end{bmatrix}$ $\quad q \in \mathbb{R}^{\ell}$, $\quad \eta \in \mathbb{R}^{6}$

$$\eta \triangleq \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} \in \mathbb{R}^6 \qquad \eta_1 \triangleq \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^3 \qquad \eta_2 \triangleq \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \in \mathbb{R}^3$$

$${}^{w}_{b}R = R_z(\psi)\, R_y(\theta)\, R_x(\phi)$$

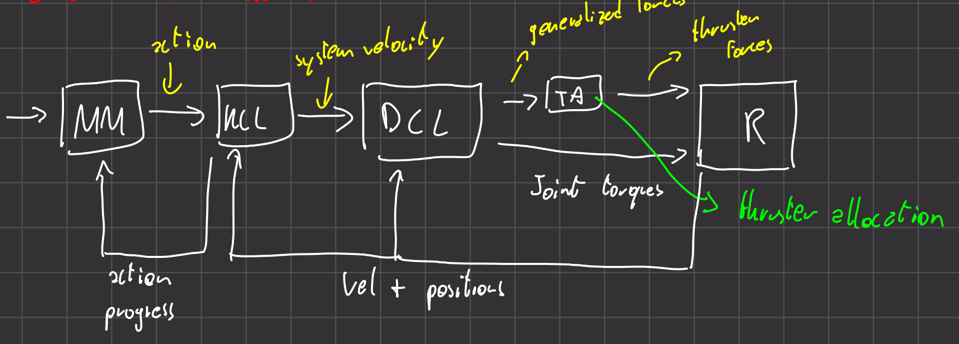
The q vector is measured with encoders.

For η_1 if you are over the water you can use GNSS, underwater you have to use a special kind of sonar.

Otherwise you can use beacons over or under the water to localize. This is called (LBL) or (USBL)

η_2 you measure with a (AHRS)

General control architecture



Control objectives

Mathematical description of what the robot needs to achieve.

→ general variable that depend on a configuration

$x(c) = x_0$

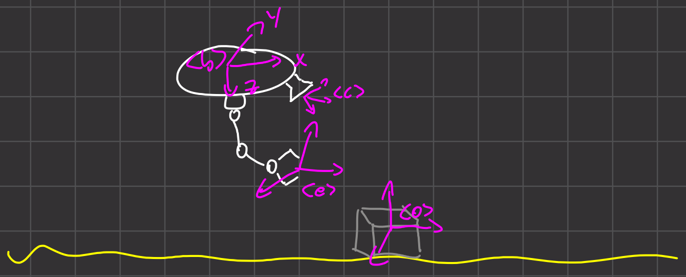
↳ I want to be the variable equal to x_0 (equality objective)

$$\left.\begin{aligned} x(c) &\le x_m \\ x(c) &\ge x_M \end{aligned}\right\} \text{ inequality objectives.}$$

Categories of objectives

- Constraint objective
- Safety objectives
- Prerequisite objectives
- Action defining objectives
- Optional objective

Example of control objectives



- end effector / tool-frame position and orientation control

$$\begin{cases} \rho_{e/g} = \phi \\ p_{e/g} = \phi \end{cases}$$

[E, AD]
equality action defining

- Collision avoidance [I, S]

$r_{e/g}^{(i)} \geq th_{(i)}$ $\quad i = 1,2,3$

$\| r_{e/g} \| \geq th$

- Joint limits [I, S]

$$\begin{cases} q_i \geq q_{i,min} \\ q_i \leq q_{i,max} \end{cases} \quad i = 1, \dots, l$$

- Camera centering [I, P]

$\| \rho_{c/g} \| \leq th$

- Arm fixed position [E, C] It could also be a [E, Optimization]

$q = q^*$ or $\dot{q} = \phi$

• Vehicle position and orientation control

$$\begin{cases} \sigma_{n/gb} = \phi \\ g_{n/gb} = \phi \end{cases}$$

[E, P]
[E, AD]
[E, O]

• Vehicle motion optimization

$v = \phi$ [E, C] (If on the seafloor)
[E, O]

• Horizontal attitude [I, P]

$$\|\rho_{k_b, k_w}\| \le th$$

To reach the goal you can either set roll and pitch to ϕ or $\rho = \phi$. The result is the same, the transient no.

• Vehicle Alignment [I, O]

$$\|\rho_{iv,d}\| \le th$$

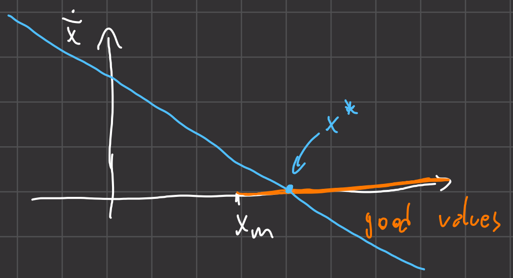
Control Task

Part of the task is the reference rate which is a vector built like this:

$$\dot{\bar{x}} = \underbrace{\lambda}_{gain}(x^* - x), \quad \lambda > 0$$

For equality objectives: $x^* = x_\phi$

For inequality objectives:



We will go towards x^*

Task Jacobian

$\dot{x} = J_x \dot{y}$ $\qquad J_x \in \mathbb{R}^{m \times n}$

$\qquad \dot{x}, x \in \mathbb{R}^m$

Reactive / non reactive tasks

Ideal operation and we stop if we find an obstacle.

Actions

Action Grasping (Fixed Base)

1) Joint limits

2) Manipulability

3) e.e position control

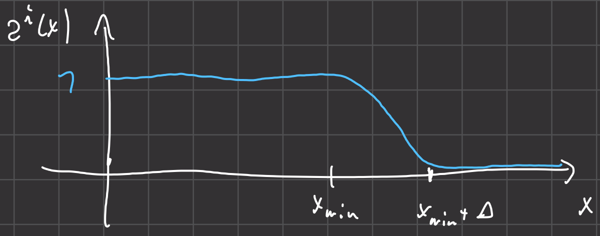
4) ee orientation control

5) arm preferred shape.

Task activation Function

$a^i(x) \in [0, 1]$ $\quad$ → inequality $\qquad x(c) \geq x_m$ → minimum

$$a^i(x) = \begin{cases} 1 & x < x_{min} \\ s(x) & x_{min} < x < x_{min} + \Delta \\ \phi & x > x_{min} + \Delta \end{cases}$$



In the generation of the reference rate we had:

↳ this is good because the velocity goes to ∅ when the deactivation of the task happens.

The problem with the left one is that you never deactivate because you stay in the transition. (chatter)

The problem with the right is the high speed on the right.

Task priorities

Once objectives and tasks have been individuated, a priority must be established.

For every priority level $k = 1 \dots N$

$$\dot{\bar{x}}_k \triangleq \begin{bmatrix} \dot{\bar{x}}_{1,k} \\ \vdots \\ \dot{\bar{x}}_{m,k} \end{bmatrix}$$

Stacked vector of $\overbrace{}^{\text{reference times of tasks}}$ at that priority level

Do not mix Jacobians stacking one on top of the other, otherwise you will lose the possibility of SVD and check for singularity!