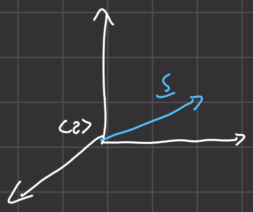
Quadratic programming for task priority

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:

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

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

↳ 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 vectors.

admittance / impedance control

Time derivative of generic vector.

$s = n\sigma \qquad |\underset{\text{constant modulus vector}}{n}| = 1 \qquad \sigma > \emptyset$

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

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

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

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

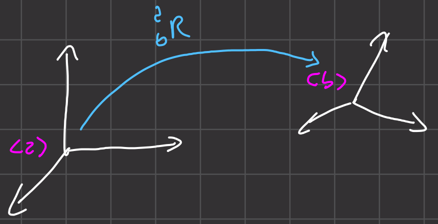
Recalling that:

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

then substituting

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

Derivative of rotation vector



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

$\omega_{b/a}$

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

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

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

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

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

↳ because of this necessarily orthogonal to n

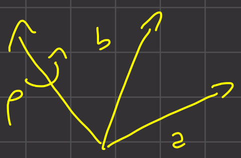
$\dot{\rho} = -\lambda\rho$ → desired behaviour to reach the goal

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



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

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

↳ 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

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

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:

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

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

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

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

$a \times \underline{b} = \underline{n}\,\sin(\theta)$ 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 $v_{P/b} = \emptyset$

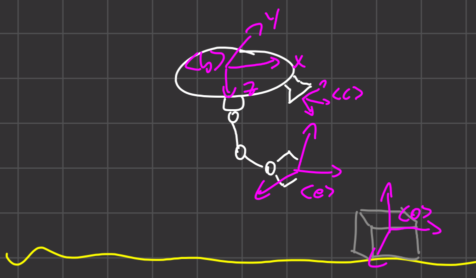
Time derivative of distance vector

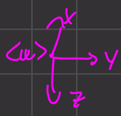
$$\underline{s} = (P - Q)$$

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

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

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

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

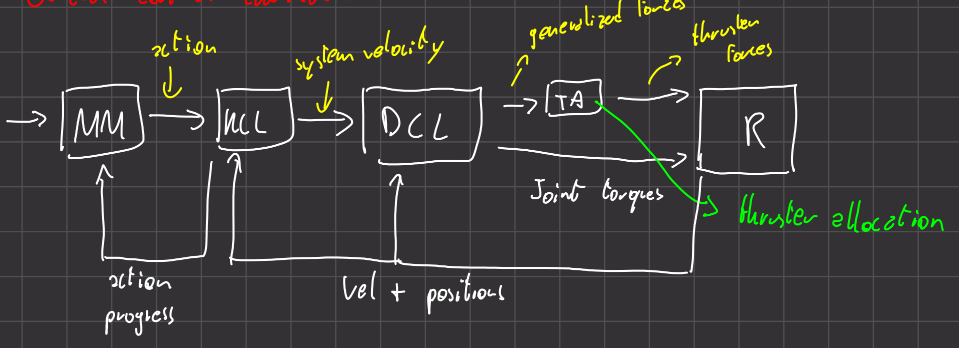
The q vector is measured with encoders.

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 b.c the variable equal to x_0 (equality objective)

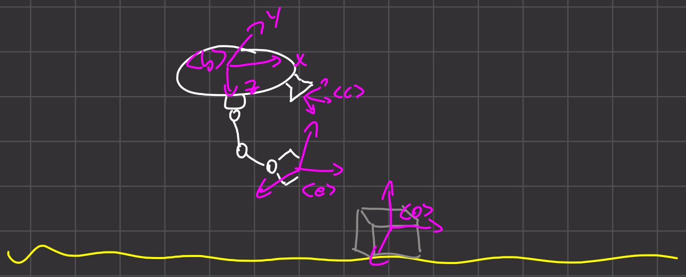
$x(c) \leq x_m$

$x(c) \geq x_M$ } 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

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

[E, AD]
equality   action defining

- Collision avoidance [I, S]

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

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

- Joint limits [I, S]

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

- Camera centering [I, P]

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

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

$$q = q^* \text{ or } \dot{q} = \emptyset$$

• Vehicle position and orientation control

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

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

• Vehicle motion optimization

$v = \phi$

[E, C] (If on the seafloor)
[E, O]

• Horizontal attitude [I, P]

$\| \rho_{k_b, n_e} \| \leq th$

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]

$\| \rho_{i_v, d} \| \leq th$

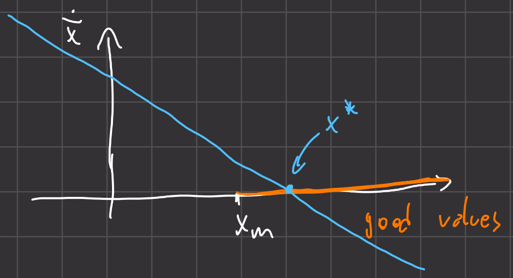
Control Task

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

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

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

For inequality objectives:



We will go towards x*

Task Jacobian

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

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

Reactive / non reactive tasks

Idea operation and we stop if we find an obstacle.

Actions

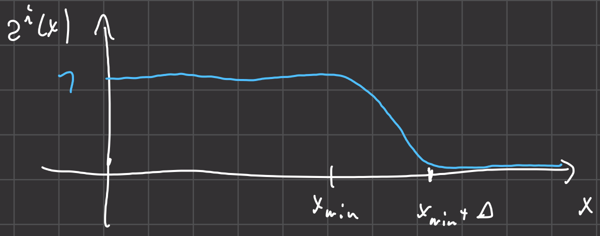
Action Grasping (Fixed Base)

1) Joint limits
2) Manipulability
3) e.e position control
4) ee orientation control
5) arm preferred shape.

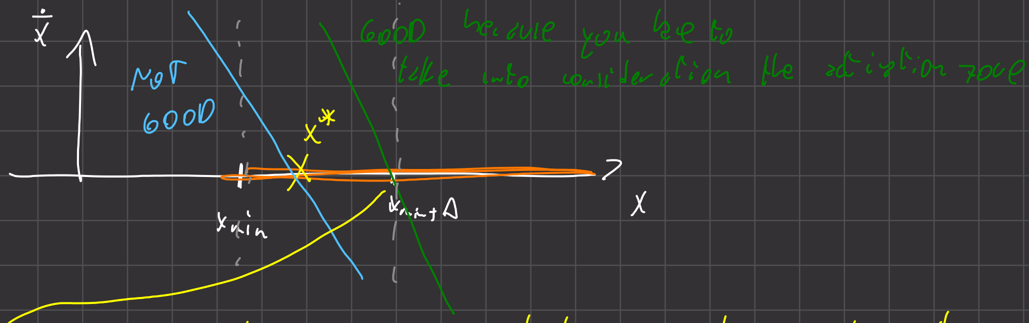
Task activation Function

$a^i(x) \in [0, 1]$ 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 $\emptyset$ 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 relevant 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!

29/10/24

Moore - Penrose Pseudo Inverse

Pseudo-inverse $A^{\#}$ of a matrix A is the generalization of the inverse.

$A \in \mathbb{R}^{m \times n}$

$A^\# \in \mathbb{R}^{n \times m}$ which satisfies the following 4 criteria.

1) $AA^\#A = A$

2) $A^\#AA^\# = A^\#$

3) $(A^\#A)^T = A^\#A$

4) $(AA^\#)^T = AA^\#$

Note that if you substitute the inverse to the pseudoinverse and the matrix is square and full-rank then it's the same

This is good for every matrix.

- When A is full rank and has linearly independent columns:

$$A^\# = \underbrace{(A^TA)^{-1}}_{\text{is full rank and invertible for this}} A^T \qquad A^\#A = \mathbb{1}$$

Left pseudo-inverse

- When A has linearly independent rows

$$A^\# = A^T(AA^T)^{-1} \qquad AA^\# = \mathbb{1}$$

Right pseudo inverse

Among other properties we recall the following ones:

- $A^\# = (A^TA)^\# A^T$
- $A^\# = A^T(AA^T)^\#$

Preliminaries on least square problems

$\dot{x} = J\dot{q}$ $\qquad J \in \mathbb{R}^{m \times n}$

$\dot{q} \in \mathbb{R}^n$

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

$\dot{\bar{x}}$ = desired velocity

What's $\dot{y}$ that best satisfies $\dot{\bar{x}} = J\dot{y}$

$$\min_{\dot{y}} \|\dot{\bar{x}} - J\dot{y}\|^2$$

$$\|\dot{x} - J\dot{y}\|^2 = (\dot{x} - J\dot{y})^T(\dot{x} - J\dot{y}) =$$

$$= \underbrace{\dot{x}^T\dot{x} + \dot{y}^T J^T J\dot{y} - 2\dot{y}^T J^T\dot{x}}_{\text{SCALAR}}$$

$$\frac{\partial}{\partial \dot{y}} = 2J^TJ\dot{y} - 2J^T\dot{x} = \emptyset$$

$$\cancel{2}\, J^TJ\dot{y} = \cancel{2}\, J^T\dot{x}$$

According to the definition of pseudo inverse the best that we can do is:

$$\dot{y} = (J^TJ)^\# J^T\dot{x} \qquad norm(y) \leq norm(y)$$

$$\dot{y} = \underbrace{(J^TJ)^\# J^T\dot{x}}_{\text{Min norm solution } *} + \underbrace{\left(\mathbb{1} - (J^TJ)^\# J^TJ\right)}_{Ker(J^TJ)\ *}\dot{z} \qquad \forall \dot{z}$$

this is the projector on the kernel J^TJ

this mean that if you multiply this matrix times J^TJ you get $\emptyset$.

$\dot{z}$ is arbitrary.

Using the pseudoinverse properties:

$$\dot{y} = J^\#\dot{x} + (\mathbb{1} - J^\# J)\dot{z} \qquad \forall \dot{z}$$

The manifold of solutions are * *

Performing a regularization means changing the original problem by adding a cost:

$\dot{y}^T R\dot{y}$ → This is added in order to prevent singularities.

$$\min_{\dot{y}}\left\{\|\dot{x} - J\dot{y}\| + \|\dot{y}\|_R^2\right\} \qquad R \geq 0 \qquad \|\dot{y}\|_R^2 = \dot{y}^T R\dot{y}$$

↳ this means that R acts as a weight.

By following steps equal to the previous we get:

$$(J^TJ + R)\dot{y} = J^T\dot{x}$$

minimizes

$$\dot{y} = (J^TJ + R)^\# J^T\dot{x}$$

the corresponding manifold is:

$$\dot{y} = (J^T J + R)^{\#} J^T \dot{\bar{x}} + \Big(\mathbb{1} - \underbrace{(J^T J + R)^{\#}}_{\text{If } R = \mathbb{1} \text{ then full rank} \Rightarrow \text{no kernel} \Rightarrow \text{one solution}} (J^T J + R)\Big) \dot{z} \quad \forall \dot{z}$$

This is bad because we want to find the other tasks in the manifold of the first one.

difference

Instead, what we do is:

$$J^{\#} \rightarrow (J J^T + R)^{\#}$$

$$\dot{y} = (\underbrace{J^T J + R}_{\text{changed } J^{\#} \text{ with its regularized form}})^{\#} J^T \dot{\bar{x}} + \big(\mathbb{1} - (J^T J + R)^{\#} J^T\big) \dot{z} \quad \forall \dot{z}.$$

this is the one that we use in software because less strict

How this choice influences the result depends on the regularization.

This is not a projector in the kernel but a non orthogonal projector.

Regularizations:

- Damped Least squares

$$\min_{\dot{y}} \| \dot{\bar{x}} - J \dot{y} \|^2 + \gamma^2 \| \dot{y} \|^2$$

$$R = \gamma^2 \mathbb{1}$$

there is no possibility of finding a kernel after this. If you try it will be $\emptyset$.

- Classic task priority Algorithm (80s - 90s)

$$\underset{\substack{\downarrow \\ \text{manifold of solutions}}}{S_1} = \{ \arg\min_{\dot{y}} \| \dot{\bar{x}} - J_1 \dot{y} \|^2 \}$$

S_1 corresponds to

$$\dot y = \underbrace{(J_1)^\# \dot x_1}_{\text{Min norm solution for } T_1} + (\mathbb{1} - J_1^\# J_1)\, \dot z \qquad \forall \dot z$$

The idea is finding:

$$S_2 \triangleq \left\{ \arg\min_{\dot y \in S_1} \| \dot x_2 - J_2 \dot y \|^2 \right\} =$$

What I have to do is substitute S_1 to $\dot y$

$$= \left\{ \arg\min_{\dot z} \| \dot x_2 - J_2 J_1^\# \dot x_1 - J_2 \underbrace{(\mathbb{1} - J_1^\# J_1)}_{Q_1} \dot z \|^2 \right\}$$

$\dot z$ ↳ free var

original mot for T_2

this is contribution of T_2 to T_1. We will call this $\hat{\dot x}_2$

Kernel of J_2 projected on J_1. How T_1 is helping J_2. If it's orthogonal then this becomes $\emptyset$.

$$S_2 = \left\{ \arg\min_{\dot z} \| \tilde{\dot x}_2 - J_2 Q_1 \dot z \|^2 \right\}$$

$$\dot z = (J_2 Q_1)^\# \tilde{\dot x}_2 + \mathbb{1} - (J_2 Q_1)^\# J_2 Q_1)\, \dot w, \quad \forall \dot w$$

After T_2 then:

Ploonestic, could be removed -> $Q_1 Q_1 = Q_1$

$$\dot y = J_1^\# \dot x_1 + \underbrace{Q_1 (J_2 Q_1)^\# \tilde{\dot x}_2}_{\text{min norm of } 2} + \underbrace{Q_1 \left(\mathbb{1} - (J_2 Q_1)^\# J_2 Q_1 \right) \dot w}_{\text{description of the kernel for both}}$$

Iterative task priority alg.

This can be rewritten iteratively:

$$P_\emptyset = \emptyset \qquad Q_\emptyset = \mathbb{1}$$

→ Kernel of task at level K

$$Q_K = Q_{K-1} \left(\mathbb{1} - (J_K Q_{K-1})^\# J_K Q_{K-1} \right)$$

$$P_K = P_{K-1} + Q_{K-1} (J_K Q_{K-1})^\# (\dot x_K - J_K P_{K-1})$$

↳ solution that I am building step by step. At the end it will be $\dot y$

N.B

The final solution is

$$\dot{y} = \rho_N \qquad N \text{ is the number of priority levels.}$$

Modern Task priority algorithm

$$X^{\#, A, Q} \triangleq \left(X^T A X + \eta (\mathbb{1} - Q)^T (\mathbb{1} - Q) + V^T P V \right)^{\#} X^T A A$$

Where V is the right orthonormal matrix of the SVD of

$$\underbrace{X^T A X + \eta (\mathbb{1} - Q)^T (\mathbb{1} - Q)}$$

this mean that some actions are shared with higher priority task.

$$W_k = J_k Q_{k-1} \left(J_k Q_{k-1} \right)^{\#, A_k, Q_{k-1}}$$

$$Q_k = Q_k \left(\mathbb{1} - (J_k Q_{k-1})^{\#, A_k, \mathbb{1}} J_k Q_{k-1} \right)$$

$$\rho_k = \rho_{k-1} + Q_{k-1} (J_k Q_{k-1})^{\#, A_k, \mathbb{1}} W_k \left(\dot{x}_k - J_k \rho_{k-1} \right)$$

everything starts with:

$$\min_{\dot{y}} \| A_1 \underbrace{(\dot{x}_1 - J_1 \dot{y})} \|^2$$

Weighted problems can be not affected by weights with least square error: It could happen that if this goes to $\emptyset$ A_1 (weights) is irrelevant. When it becomes $\emptyset$, then you have a discontinuity.

The second task will take all the dimensions.

Singular Value Decomposition

Theorem

Given any matrix $A \in \mathbb{R}^{m \times n}$, it can always be composed as

$$A = U \Sigma V^T$$

$U \in \mathbb{R}^{m \times m}$ → dimension of the task. orthonormal $\quad U^{-1} = U^T$

$V \in \mathbb{R}^{n \times n}$ $\quad$ // $\quad V^{-1} = V^T$

$\Sigma \in \mathbb{R}^{m \times n}$

$$\Sigma = \begin{cases} [\bar{\Sigma} \vdots \varnothing] & \text{if} \quad m < n \\ \bar{\Sigma} & \text{if} \quad m = n \\ \begin{bmatrix} \bar{\Sigma} \\ \cdots \\ \varnothing \end{bmatrix} & \text{if} \quad m > n \end{cases}$$

$$\bar{\Sigma} = \mathrm{diag}(\sigma_1, \dots, \sigma_k) \in \mathbb{R}^{k \times k}$$

$$k = \min(m, n)$$

$$\sigma_i = \begin{cases} \sqrt{\lambda_i (AA^T)} & m < n \\ \sqrt{\lambda_i (A^T A)} & m \geq n \end{cases}$$

If $m = n$ you can use weather.

$$\sigma_1 > \sigma_2 > \sigma_3 > \dots > \sigma_k$$

Geometric Interpretation

$$y = Ax$$

$$y = U \Sigma V^T x$$

$$\underbrace{U^T y}_{\xi} = \cancel{U^T U} \Sigma \underbrace{V^T x}_{\theta}$$

these can be considered as new variables

$\xi = \Sigma \theta \rightarrow$ Between these two spaces the transformation is diagonal which means that there is only one scalar between one direction in a space and in one other.

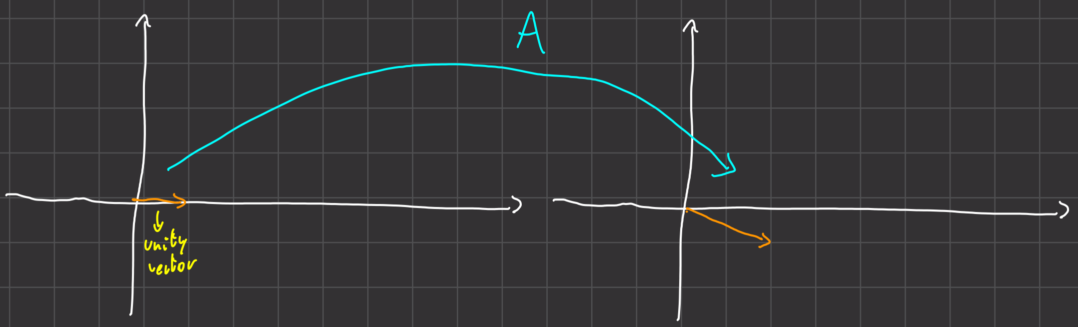
It's like you rotate both spaces to have a 1 to 1 match between them.

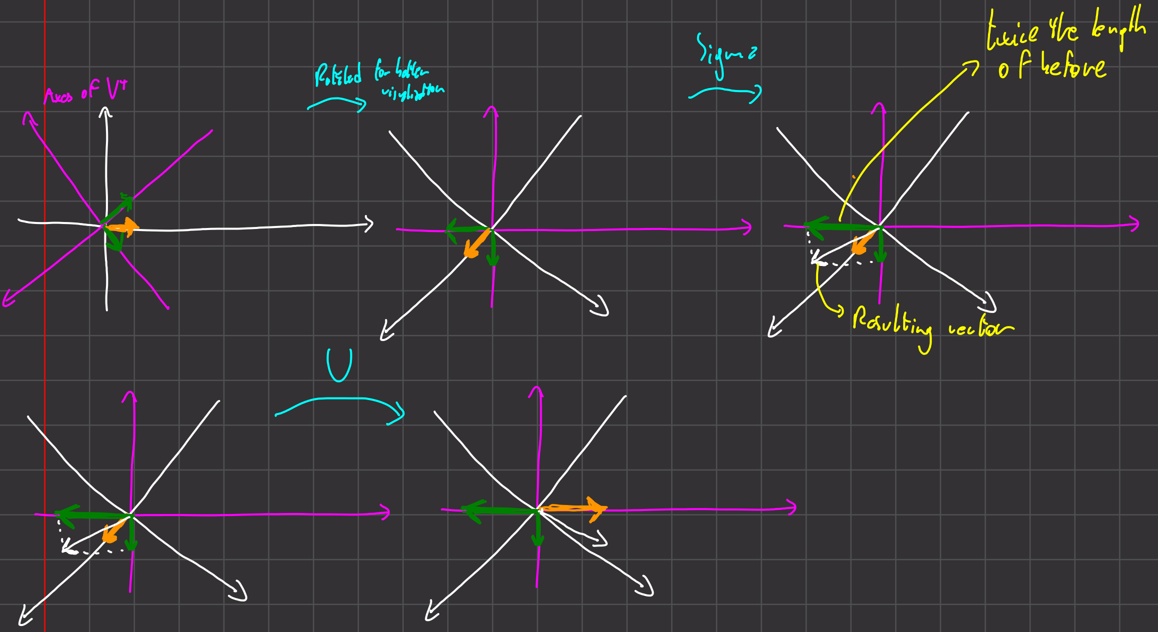
If the dimension is different. Example 3 and 6. the 3 direction of the one of 3 will map directly to the one of 6.

Example

$$A = \begin{bmatrix} 1.4142 & 1.4142 \\ -0.7071 & 0.7071 \end{bmatrix}$$

$$A = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -0.7071 & -0.7071 \\ -0.7071 & 0.7071 \end{bmatrix}$$





If you have a situation where one of the scalar is approaching ∅ then in the output goes to ∅.

In the opposite direction the vector becomes very large.

If we have three joints in the rotated space of the SVD we will see that the value of that joint will tend to ∅.

Pseudo inversion through SVD

Case $m < n$

$$J^{\#} = J^T (J J^T)^{-1} = (U \Sigma V^T)^T \left(U \Sigma V^T (U \Sigma V^T)^T \right)^{-1} =$$

$$= V \Sigma^T U^T \left(U \Sigma^T \cancel{V^T V} \Sigma^T U^T \right)^{-1} =$$

$$= V \Sigma^T U^T \left(U \Sigma \Sigma^T U^T \right)^{-1} =$$

$$= V \Sigma^T U^T \left(U \Sigma^2 U^T \right)^{-1} =$$

$$= V\Sigma^T U^T U \Sigma^{-2} U^T =$$

$$= V \Sigma^T \Sigma^{-2} U^T \qquad \Sigma^T\Sigma^{-2} = \begin{bmatrix} \Sigma \\ \hline \emptyset \end{bmatrix} \Sigma^{-2} = \begin{bmatrix} \Sigma^{-1} \\ \hline \emptyset \end{bmatrix}$$

$$J^{\#} = V \begin{bmatrix} \Sigma^{-1} \\ \hline \emptyset \end{bmatrix} U^T = V \begin{bmatrix} \frac{1}{\sigma_1} & & & \\ & \frac{1}{\sigma_2} & & \\ & & \ddots & \\ & & & \frac{1}{\sigma_n} \\ \hline & & \emptyset & \end{bmatrix} U^T$$

If $m>n$

$$J^{\#} = V \left[\Sigma^{-1} \,\vdots\, \emptyset \right] U^T = V \left[\begin{array}{cccc|c} \frac{1}{\sigma_1} & & & & \\ & \ddots & & & \emptyset \\ & & & \frac{1}{\sigma_n} & \end{array}\right] U^T$$

Now what if one of the σ approaches $\emptyset$.

Singular value oriented regularization

$$J^{\#} = (J^TJ + K)^{-1} J^T \qquad K \in \mathbb{R}^{n\times n}$$

How to choose K

$$K = V \Gamma V^T \qquad \Gamma = \operatorname{diag}(\gamma_1, \gamma_2 \dots \gamma_n)$$

$$J^{\#} = (J^TJ + K)^{-1} J^T = (V\Sigma^T U^T U \Sigma V^T + K)^{-1} V\Sigma^T U^T =$$

$$= (V\Sigma^T\Sigma V^T + V\Gamma V^T)^{-1} V\Sigma^T U^T =$$

$$= (V(\Sigma^T\Sigma + \Gamma)V^T)^{-1} V\Sigma^T U^T =$$

$$= V(\Sigma^T\Sigma + \Gamma)^{-1} V^T V \Sigma^T U^T =$$

$$= \underline{V(\Sigma^T\Sigma + \Gamma)^{-1} \Sigma^T U^T}$$

If $m<n$

$$(\Sigma^T\Sigma + \Gamma)^{-1}\Sigma^T = \left(\begin{bmatrix} \Sigma^2 & \emptyset \\ \emptyset & \emptyset \end{bmatrix} + \Gamma \right)^{-1} \begin{bmatrix} \Sigma \\ \emptyset \end{bmatrix}$$

$$\left(\begin{bmatrix} \sigma_1^2+\gamma_1 & & & & & & \\ & \sigma_2^2+\gamma_2 & & & & & \\ & & \ddots & & & & \\ & & & \sigma_m^2+\gamma_m & & & \\ & & & & \gamma_{m+1} & & \\ & & & & & \ddots & \\ & & & & & & \gamma_n \end{bmatrix}^{-1} \begin{bmatrix} \Sigma \\ \hline \emptyset \end{bmatrix}$$

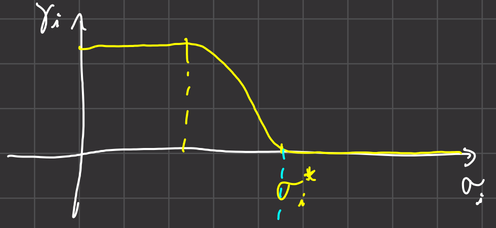
With this regularization the matrix is always invertible.

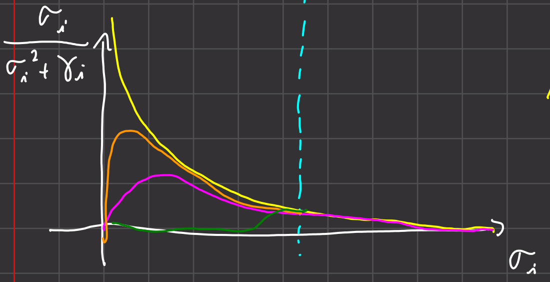
$$= \begin{bmatrix} \frac{\sigma_1}{\sigma_1^2+\gamma_1} & & \\ & \ddots & \\ & & \frac{\sigma_n}{\sigma_m^2+\gamma_m} \\ \hline & \emptyset & \end{bmatrix}$$

If sigma is sufficiently big then if gamma is $\emptyset$ then we have the original version.

If sigma is approaching $\emptyset$ we can use gamma because if $\gamma \neq \emptyset$ as σ goes to $\emptyset$ then we have $\frac{\emptyset}{\text{a number}}$.

A policy for choosing γ is:





Lambda smaller than this

Moving σ^* to the right does not change much.

If you choose a big γ and choosing a small σ^* then you would go down immediately.

From a practical point of view if we start too early σ^* high we regularize even when not necessary so for example a task that sets the gripper to go straight will not work.

In order to know if sigma is correct then I would need to look at the unit of measures.

Link between least squares and SVD regularization.

$$\min_{\dot{y}} \|\dot{x} - J\dot{y}\|^2 + \underbrace{\|V^T \dot{y}\|^2_{\Gamma}}_{}$$ → Cost that we introduced

$R = V \Gamma V^T$ → Regularization matrix

$$\dot{y} = V(\Sigma^T\Sigma + \Gamma)^{\#}\Sigma^T U^T \dot{x}$$

If one direction is going to singularity then I try to use that as few as possible.

Exercise one

The goal is to develop position control for the vehicle.

$x(c) = ?$

$$x(c) = O_{gv} - O_b$$

$$D_w\, x(c) = D_w(O_{gv} - O_b) = D_w(O_{gv}) - D_w(O_b) =$$

$$= v_{gv/w} - v_{b/w}$$ Actual kinematics

Desired kinematics

$$-\lambda x = v_{gv/w} - v^*_{b/w}$$ reference

$$\underbrace{-\lambda x - v_{gv/w}}_{\dot{x}_{gv}} = -v^*_{b/w}$$

$$\dot{x}_{gv} = J_{gv}\,\dot{y} \qquad \dot{y} = \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \\ {}^{v}v_{b/w} \\ {}^{b}\omega_{b/w} \end{bmatrix}$$

$${}^{b}\dot{x}_{gv} = \underbrace{\left[\begin{array}{c|c|c} 0_{3\times 7} & -1 & 0_{3\times 3} \end{array}\right]}_{{}^{b}J_{gv/w}} \dot{y}$$

$${}^{w}J_{gv/w} = {}^{w}_{b}R\; J_{gv/w}$$

$$= \left[\begin{array}{c|c|c} 0_{3\times 7} & -{}^{w}_{b}R & 0_{3\times 3} \end{array}\right]$$