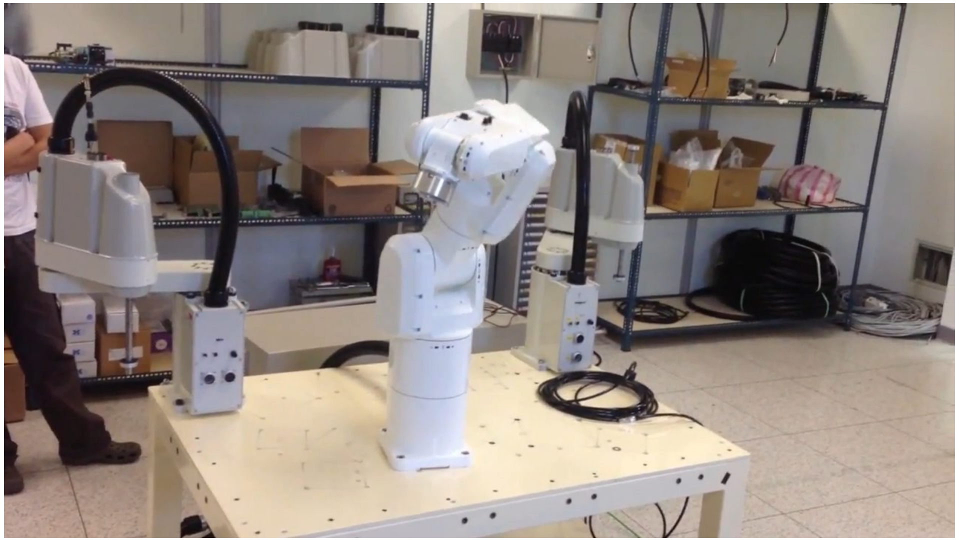
Slides have been created by Prof. Michele Focchi

webpage: https://mfocchi.github.io/Teaching/

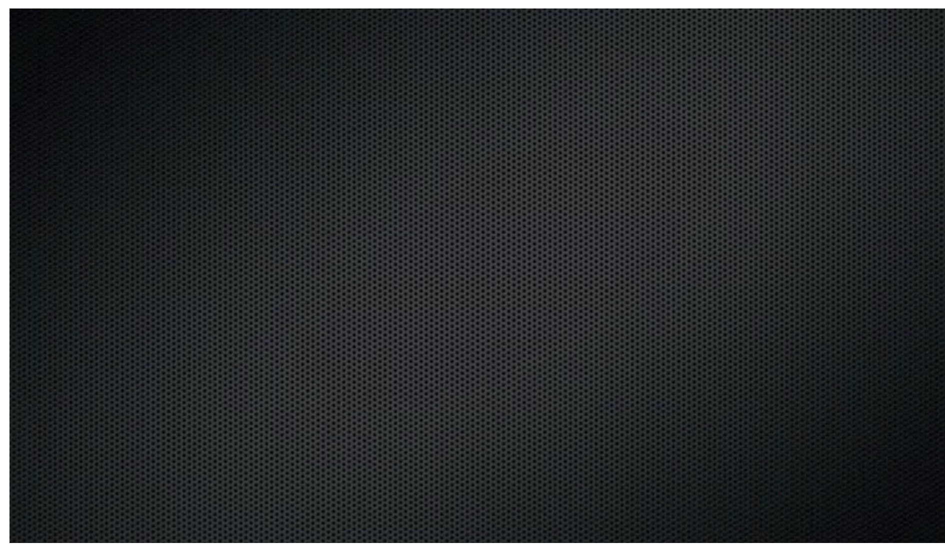
E5-Redundancy and singularities in robot manipulators

REDUNDANCY VS SINGULARITY

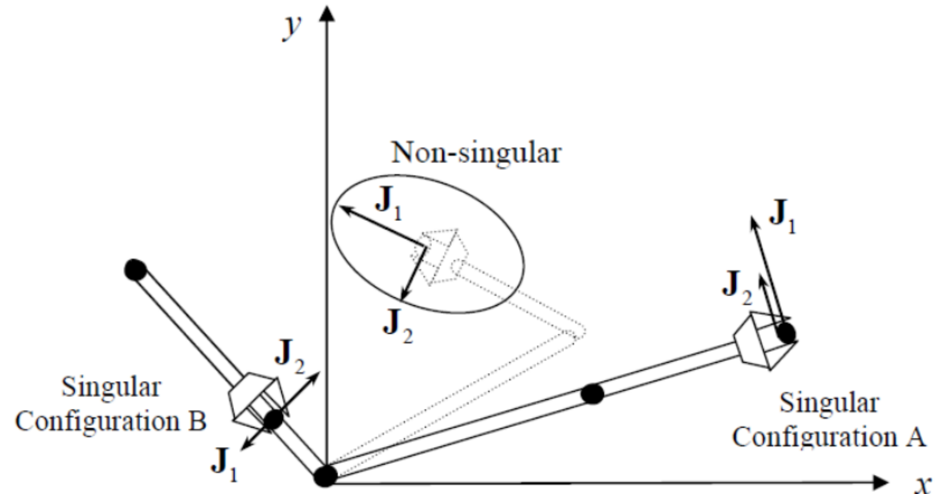


Redundancy

Singularity



REDUNDANCY VS SINGULARITY



Redundancy:

you have more mobility (i.e DoFs) than you need

Singularity:

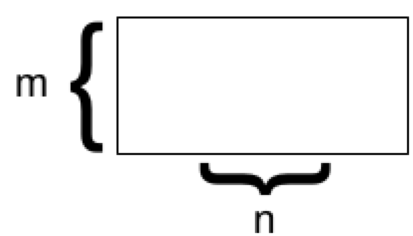
you have less mobility than you need

Redundant manipulators

1) Have more DoFs than the dimension of the task (m<n)

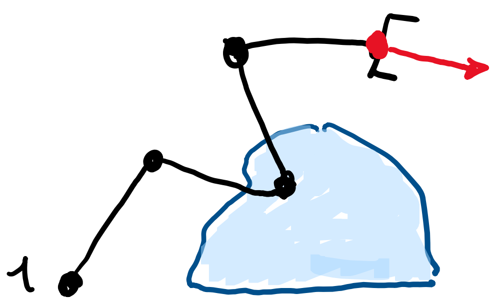
2) infinite solutions exist to the IK problem

3) The Jacobian matrix is rectangular (fat)



- The redundant DoFs can be used to increase dexterity (e.g. for obstacle avoidance)

PLANAR EXAMPLE



Robot : 4 DoFs (n)

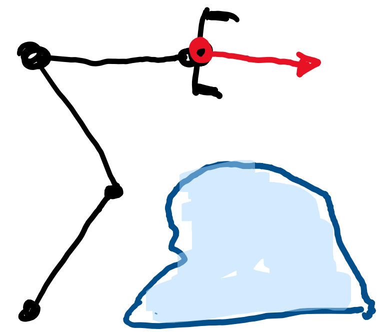
Task : 3 D (m)

1 orientations

2 positions

$\boxed{n > m}$

$\Rightarrow$



∞^{4-3} solutions

To achieve the task

- A redundant robot has self-motions, i.e. internal motions in the joint space which do not affect the task variables

Some Examples

- "redundancy" of a robot is thus a relative concept, i.e., it holds with respect to a given task…

- Some task dimensions (m):
  - position in the plane: 2
  - position in 3D space: 3
  - orientation in the plane: 1
  - pointing in 3D space: 2
  - position and orientation in 3D space: 6

- a planar robot with N=3 joints is redundant for the task of positioning its E-E in the plane (m=2), but NOT for the task of positioning AND orienting the E-E in the plane (m=3)

Usage of redundant robots

- avoid collision with obstacles (in Cartesian space) ...
- ... or kinematic singularities (in joint space)
- increase manipulability in specified directions
- stay within the admissible joint ranges
- uniformly distribute/limit joint velocities and/or accelerations
- minimize energy consumption or needed motion torques
- optimize execution time

Disadvantages of redundant robots

- a greater structural complexity of construction (more links, transmissions, ...)
- higher costs
- more complicated algorithms for motion control

Primer on linear algebra

Recall: v = J(q) q

The J matrix that changes with configuration, and as any matrix of numbers has some properties and has some geometric spaces associated to it

- rank $\rho(J)$ = max # of rows or columns that are linearly independent
  - $\rho(J) \leq \min(m, n)$ if $\rho(J) = \min(m, n) \Rightarrow$ $\boxed{\text{Full rank}}$
  - if $m = n$ and J is full rank $\Rightarrow$ J is not singular and $\exists\, J^{-1}$
  - $\rho(J)$ = dimension of the largest non singular square submatrix of J

Primer on linear algebra

- range (or column or Image) space $R(J)$: subspace of all the linear combinations of the columns of J

$$R(J) = \{ v \in \mathbb{R}^m : \text{for } \dot q \in \mathbb{R}^n ,\ v = J \dot q \} \qquad \boxed{\dim(R(J)) = \rho(J)}$$

$$v = J(:,1)\,\dot q_1 + J(:,2)\,\dot q_2 + J(:,3)\,\dot q_3 = \underbrace{\begin{array}{ccc} \begin{bmatrix} \ \\ \ \\ \ \end{bmatrix} & \begin{bmatrix} \ \\ \ \\ \ \end{bmatrix} & \begin{bmatrix} \ \\ \ \\ \ \end{bmatrix} \\ \text{col 1} & \text{col 2} & \text{col 3} \end{array}}_{J} \bullet \begin{array}{c} \begin{bmatrix} \dot q_1 \\ \dot q_2 \\ \dot q_3 \end{bmatrix} \\ \dot q \end{array}$$

$\llcorner \in \mathbb{R}(J)$

- null space (kernel) $N(J)$: subspace of all vectors that are zeroed by matrix J

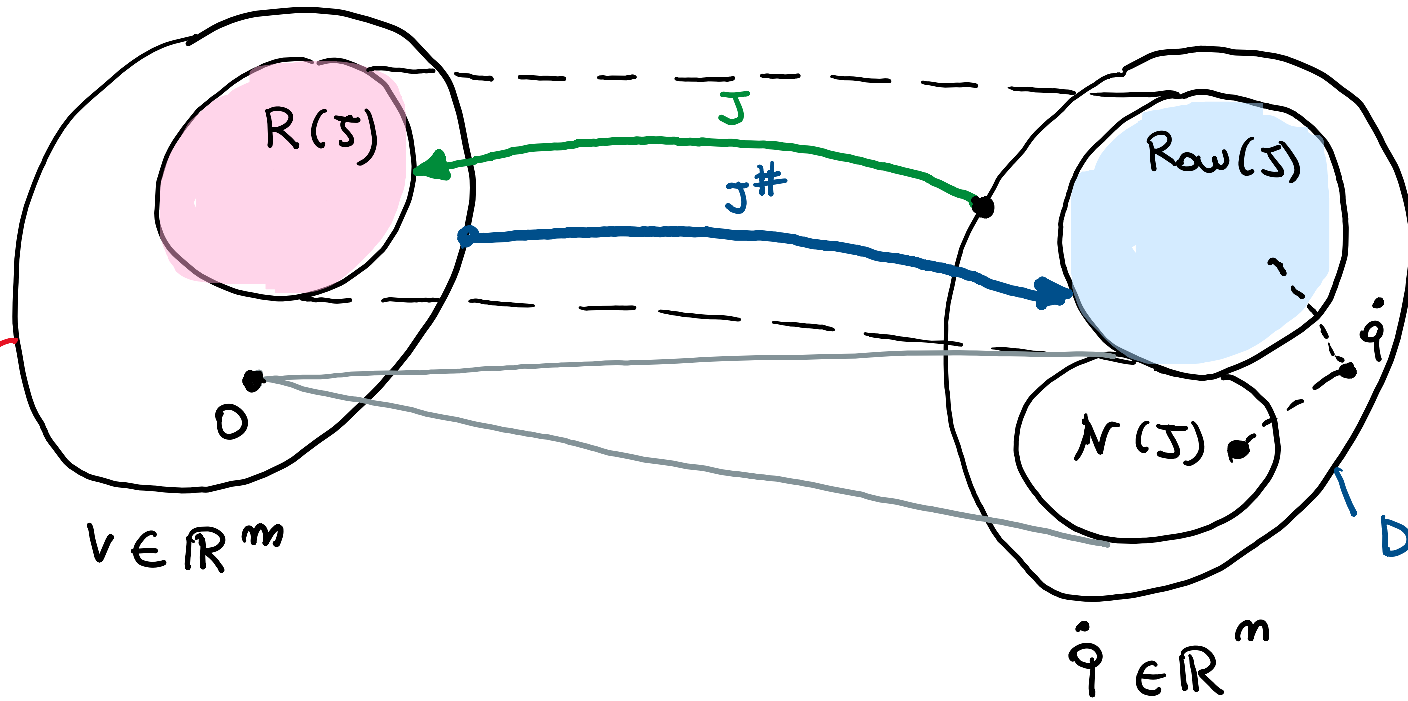
$$N(J) = \{ \dot q_0 \in \mathbb{R}^n : J \dot q_0 = 0 \in \mathbb{R}^m \} \qquad \boxed{\dim(N(J)) = n - \rho(J)}$$

- row space : subspace spanned by the rows of J or equivalently by the columns of J^T

$$Row(J) = R(J^T)$$

$$\boxed{\dim(Row(J)) = \rho(J)}$$

same as $R(J)$!



- $J^\#$ maps always onto the row space

- rank nullity Theorem

Ⓐ $\underbrace{\dim(Row(J))}_{\rho(J)} + \dim(N(J)) = n \rightarrow$ dimension of domain

$\Rightarrow Row(J) = R(J^{\#}) \perp N(J)$, $\boxed{R^n = N(J) + R(J^{\#})}$

$\Rightarrow$ any element $\dot{q} \in \mathbb{R}^n$ can be written as:

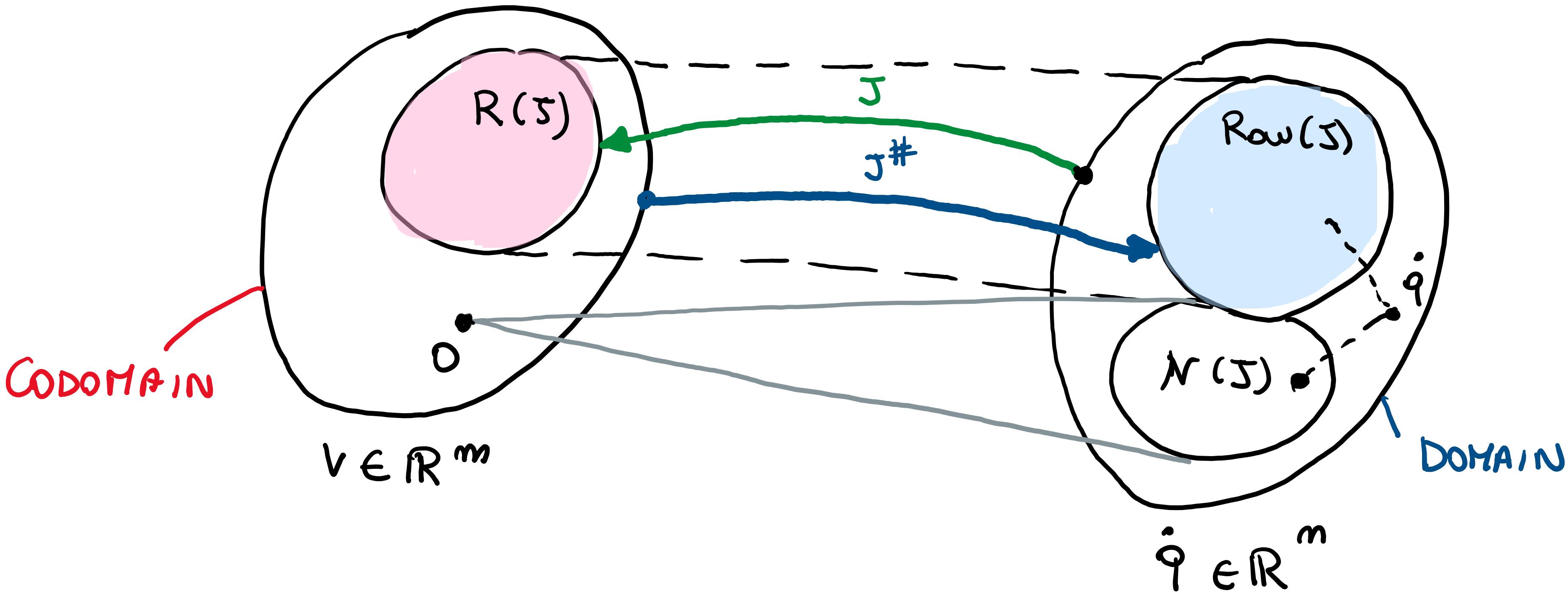
$q = q_{ROW} + q_{NULL}$, $\quad q_{ROW} \in R(J^T)$, $q_{NULL} \in N(J)$

Ⓑ $\dim(R(J)) + \dim(N(J^{\#})) = m \rightarrow$ dimension of codomain

$\Rightarrow R(J) \perp N(J^{\#})$, $\boxed{R^m = N(J^{\#}) + R(J)}$

$\Rightarrow$ any element $v \in \mathbb{R}^m$ can be written as

$v = v_{RANGE} + v_{NULL}$, $v_{RANGE} \in R(J)$, $v_{NULL} \in N(J^{\#})$



- J filters out whatever in the domain is in the null space and maps what is in the row space into the range space

- The null space component is nullified by the action of J

PHYSICAL MEANING OF SPACES

all spaces are locally defined: depend on current configuration q.

- $N(J(q)) \in \mathbb{R}^n$ set of joint velocities ($\neq \emptyset$) that do not produce any end-effector velocity

  $\Rightarrow$ always exists when $n > m$ $\Rightarrow$ the robot is redundant for the task

- $R(J(q)) \in \mathbb{R}^m$ set of end-effector velocities that can be instantaneously realized varying joint velocities $\dot{q}$ (in a given configuration q)

  - if $\rho(J) = m$ the end-effector can be moved in any direction of the task space.

  - if $\rho(J) < m$ there are directions in which the end-effector cannot move (singularity)

REDUNDANCY AND NULL SPACE

- A null space in the domain can exist only when $n > \text{rank}(J)$.
- In the case of a redundant robot with $n > m$ and full rank ($\text{rank}(J) = m$) the dimension of the null-space is $n - m$

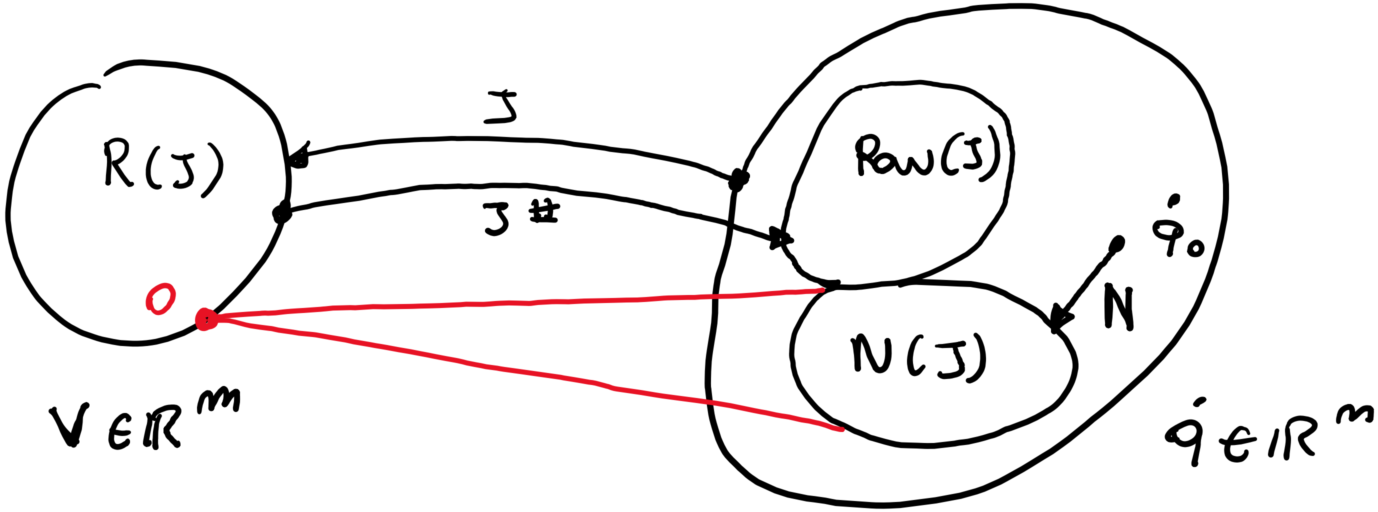
$n > m$ $\quad v = J(q)\,\dot{q} \quad$ $J = \begin{array}{cc} & n \\ m & \begin{bmatrix} \# & \# & \# & \# \\ \# & \# & \# & \# \\ \# & \# & \# & \# \end{bmatrix} \end{array}$ J rectangular (FAT)

- To compute the inverse diff. kinematics we need to invert a rectangular matrix using a pseudo-inverse

$$\dot{q} = \underbrace{J^{\#}}_{\text{pseudo-inverse}} v + \underbrace{[I - J^{\#} J]}_{\text{Null-space projector } N} \dot{q}_0$$

$\boxed{\text{NULL-SPACE METHOD}}$

$\Rightarrow$ I can define a secondary Task $\dot{q}_0$ (e.g. postural) that will not infuence the E-E one

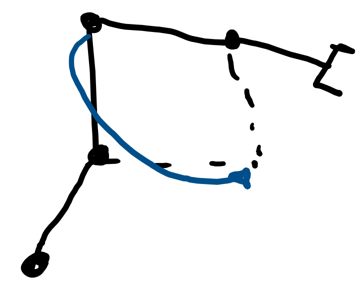


$$0 = J \boxed{\left[I - \underbrace{J^{\#} J}_{\text{row space projector}} \right] \dot{q}_0}$$

$\downarrow$

v

Joint motion that results in no end-effector motion



⊕ We can implement other tasks in $\dot{q}_0$ (increase dexterity / avoid obstacles)

e.g. postural task $\quad \dot{q}_0 = K(q^\circ - q) \quad q^\circ$ = default config

PSEUDO-INVERSE COMPUTATION

- necessary to invert a rectangular matrix
- depending on the nature of the matrix A we can have RIGHT or LEFT pseudo-inverses $A^{\#}$:

Ⓐ we assume A is FULL RANK

FAT CASE ($m < n$)

$$A = \begin{array}{c c} & n \\ m & [\quad\quad] \end{array}$$

$$A^{\#} = ? \begin{array}{c c} & m \\ n & [\quad] \end{array}$$

$$\boxed{A^{\#} = A^T \underbrace{(A A^T)}_{} {}^{-1}}$$

square & invertible

SKINNY CASE ($m > n$)

$$A = \begin{array}{c c} & n \\ m & [\quad] \end{array}$$

$$A = \begin{array}{c c} & m \\ n & [\quad\quad] \end{array} ?$$

$$\boxed{A^{\#} = (A^T A)^{-1} A^T}$$

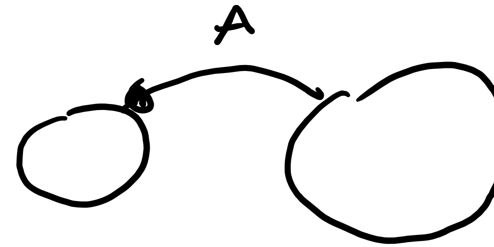
PSEUDO INVERSE COMPUTATION

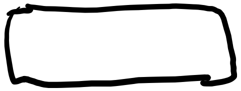
(B) assume rank is not full

- AA^T or A^TA are not invertible
- $A^\#$ still exists and can be computed numerically with singular value decomposition (SVD) of A

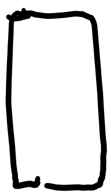
$\boxed{\text{MATLAB : pinv}}$

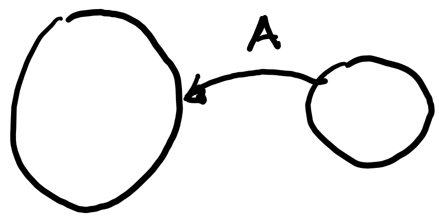
NOTES ON PSEUDO INVERSES

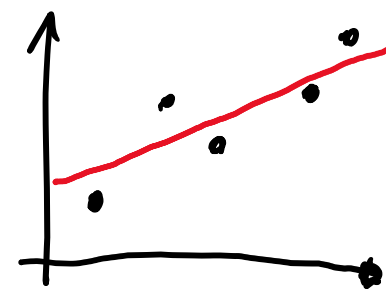


FAT = 

- more unknown than equations
- ∞ solutions exist
- returns the minimum norm solution to $Ax = b \longrightarrow x = A^{\#} b$ (that minimize $\|x\|^2$)

SKINNY = 





- more equations than unknowns
- no exact solution exists
- returns the "best fit" (least square) that minimizes $\|Ax - b\|^2$

RECALL OPTIMIZATION PROBLEM

- problem of finding the "best" solution from all the feasible solutions

- in standard form:

$$\min_{x} \; f(x) \rightarrow \text{cost}$$

$x \rightarrow$ decision variable(s)

s.t. $\rightarrow$ subject to

$$\left.\begin{aligned} g(x) &= 0 \\ h(x) &\leq 0 \end{aligned}\right\} \rightarrow \text{constraints}$$

- $\max \Rightarrow \min \; -f(x)$

- $x^* = \arg\min f(x) \rightarrow$ optimal solution

LEAST SQUARE PROGRAM - SKINNY

$x = A^{\#} b$ can be casted as an un constraint optimization problem

$\quad\to Ax - b = 0$

$$x^* = \underset{x}{\arg\min} \frac{1}{2} \| Ax - b \|^2 = \underbrace{\frac{1}{2} x^T A^T A x - b^T A + \frac{1}{2} b^T b}_{f(x)}$$

$$\nabla f = A^T A x - A^T b = 0 \Rightarrow A^T A x = A^T b$$

$$x^* = \underbrace{(A^T A)^{-1} A^T}_{A^{\#}_{\text{SKINNY}}} b$$

NORM MINIMIZATION – FAT

- if A is full-row rank ρ

$$x^* = \arg\min_{x} \|x\|^2 \quad \text{s.t. } Ax = b$$

$\Rightarrow x^*$ minimizes $\|x\|^2$ among the ∞ x that satisfy $Ax = b$

- if A is not full-row rank ρ

$$x^* = \arg\min_{x \in S} \|x\|^2$$

$$S = \{ x \in \mathbb{R}^n : \|Ax - b\| \text{ is minimum} \}$$

$\Rightarrow x^*$ minimizes $\|x\|^2$ among the ∞ x that minimize $\|Ax - b\|$

GEOMETRIC INTERPRETATION FOR $m < n$

$$\dot{q} = J^{\#} v + \underbrace{[I - J^{\#} J]}_{} q^0$$

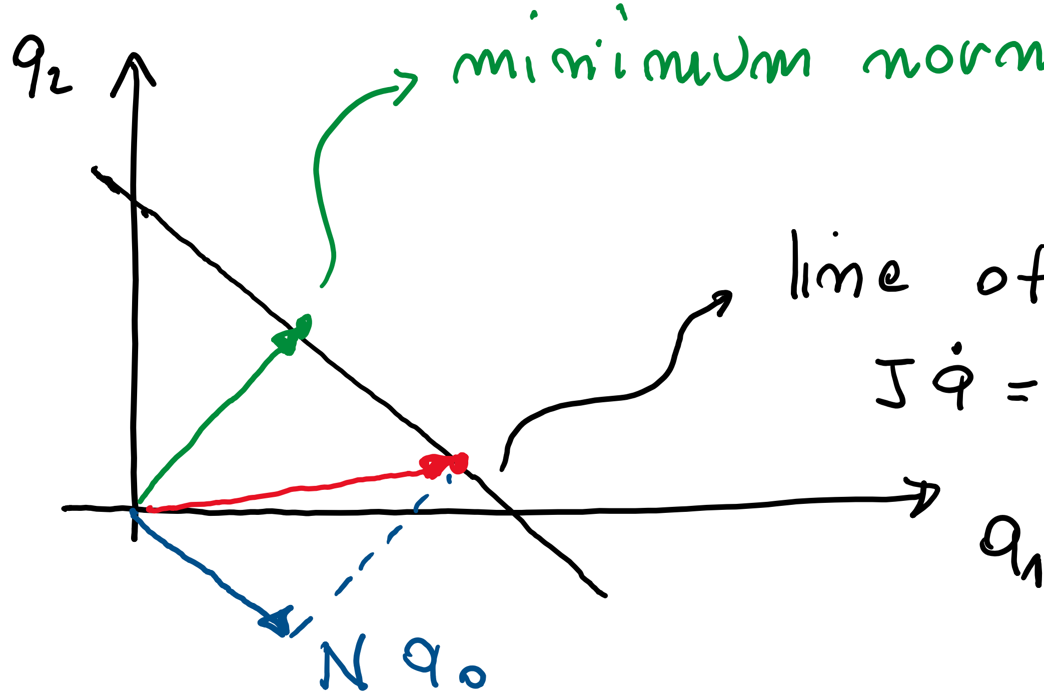
→ filters the row space component and projects in the null-space

Null-space projector N

simple example:

$n = 2$, $m = 1$

minimum norm solution ($J^{\#} v$)



line of ∞ solutions of

$J \dot{q} = v$

GENERALIZED PSEUDOINVERSE

- what we defined so far was a specific Type of pseudo-inverse called MOORE-PENROSE pseudo-inverse, we indicate with The symbol A^+
- ∞ pseudo-inverse exist setting a weight W

$$\boxed{A_W^\# = W A^T (A W A^T)^{-1}}$$

generalized pseudo-inverse (for a fat matrix)

check:

$$A A_W^\# = A W A^T (A W A^T)^{-1} = I_{m \times m} \quad \boxed{\text{OK}}$$

- The solution of $\dot{q} = J_W^\# v$ in This case minimizes The weighted norm

$$\|\dot{q}\|_W^2 = \dot{q}^T W \dot{q} \qquad W \text{ pos. definite}$$

SINGULAR VALUES

- extension of eigenvalues to rectangular matrix
- let $A \in \mathbb{R}^{m \times n}$
- The matrix $A^T A \in \mathbb{R}^{n \times n}$:

  ① is symmetric → n real eigenvalues λ_i

  ② $\lambda_i \geq 0 \qquad i \in [1, \ldots, n]$

  ③ The singular values of A are linked to the eigenvalues of $A^T A$

$$\boxed{\sigma_i = \sqrt{\lambda_i}} \qquad \sigma_1 > \sigma_2 > \cdots \sigma_n$$

- if $m < n$ then A has at most rank m and therefore at most m non-zero σ_i

- The number of non-zero singular values of A equals the rank $\rho(A)$ and Tell us the dimension of the range space $R(A)$

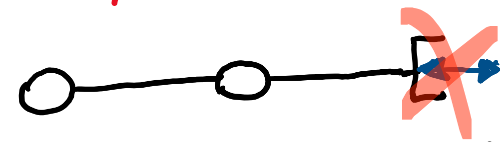
$$\sigma_1 > \sigma_2 > \cdots \sigma_\rho > 0$$

- The number of zero singular values Tell us The dimension of The null-space $N(A)$

$$\sigma_{\rho+1} = \cdots = \sigma_m = 0$$

KINEMATIC SINGULARITIES AT VELOCITY LEVEL

in a singular configuration $\bar{q}_s$:



① end-effector mobility loss (cannot move in some Cartesian directions of the task space)

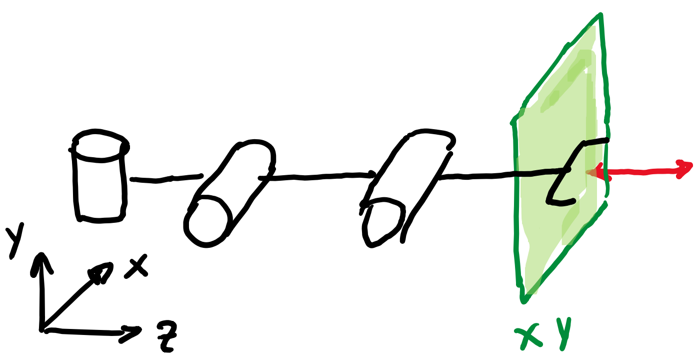
② possibility of ∞ solutions to the IK problem

③ The Jacobian loses rank $\boxed{\rho(J) < m}$

$$J = \begin{array}{c} \\ \delta x \\ \delta y \\ \delta z \end{array} \begin{array}{c} \begin{array}{ccc} \delta q_1 & \delta q_2 & \delta q_3 \end{array} \\ \left[\begin{array}{ccc} \; & \; & \; \\ \; & \; & \; \\ 0 & 0 & 0 \end{array}\right] \end{array}$$

it becomes a redundant manipulator in subspace orthogonal to singular direction

no joint motion can create movement along z, we get a row of zeros (only in this example we have a row of zeros because the robot is aligned with z axis otherwise we get a row dependent from others)

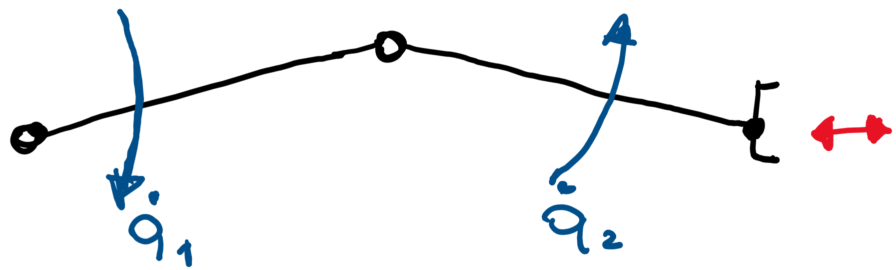


If the Jacobian degenerates at singularity the dimension of range space decreases (rank decreases) while the dimension of the nullspace increases: $\rho(J) + \dim(N(J)) = n$

① → ④:

cannot instantaneously change e-e velocity in some directions (but I can change joint configuration to escape from singularity)

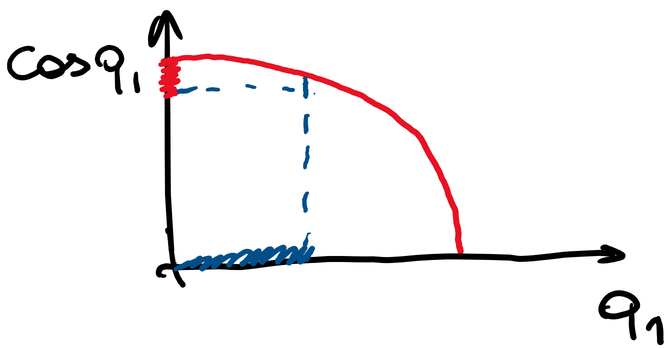
(5) close to singularity you can have big $\dot{q}$ to achieve small end-effector velocities



$$dq = J^{-1} dx$$

↳ ↑↑

acts as an AMPLIFIER



↳ Actuator saturation !

NOTE: finding all the singular configurations of a robot with big # joints is computationally complex

HOW TO COMPUTE THEM?

(A) case non-redundant robot $(m = n)$: solve the non-linear scalar equation for $\bar{q}_s$:

$$\boxed{\det(J(\bar{q}_s)) = 0} \quad \Rightarrow \bar{q}_s$$

(B) redundant robot $(m < n)$:

(B.1) check symbolically where $\boxed{\det(\underbrace{J(\bar{q}_s)\, J(\bar{q}_s)^T}_{m \times m \text{ square}}) = 0}$

(B.2) find $\bar{q}_s$ such that all $m \times m$ minors (square matrix) that you can extract from J have $\det = 0$ (e.g. are singular)

EXAMPLE DEXTER ROBOT 8R

$$J = \begin{array}{cc} & 8 \\ 6 & \left[\begin{array}{cccccccc} \; & \; & \; & \; & \; & \; & \; & \; \end{array}\right] \end{array} \longrightarrow \text{rectangular FAT}$$

how many combinations of 6 columns out of 8 columns ?

$$\binom{8}{6} = \frac{8!}{6!\,(8-6)!} = \frac{8 \cdot 7}{2} = \frac{56}{2} = 28$$

↑ binomial coefficient

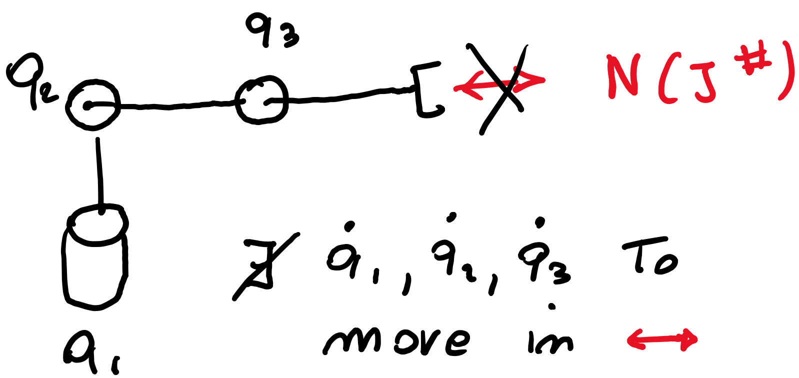
Trick: express J in a frame where is simpler

TYPES OF SINGULARITIES

BOUNDARY: singular config corresponding to cartesian points on the boundary of the reachable workspace

- stretched / retracted

ELBOW LOCK

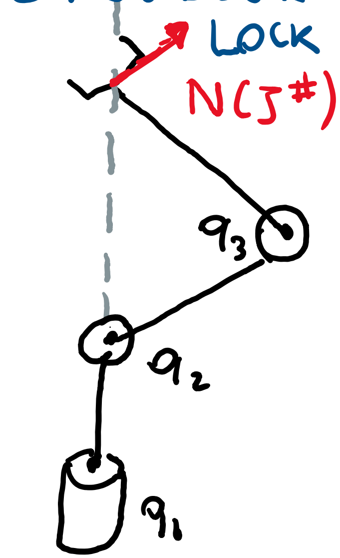


$\nexists$ $\dot{q}_1, \dot{q}_2, \dot{q}_3$ To move in ↔

→ Can be avoided setting joint limits

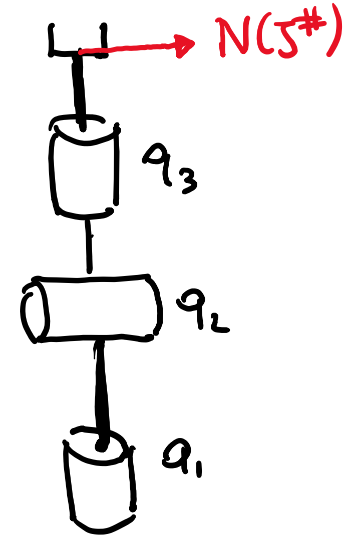
INTERNAL: They are inside the workspace usually related to the alignment of 2 or more joint axes

SHOULDER LOCK



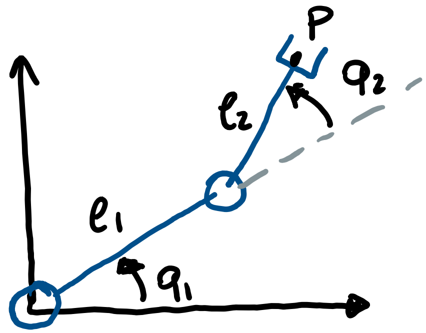
∞ q_1 for the same EE position

WRIST LOCK



∞ q_1, q_3

EXERCISE 1.1: SINGULARITY OF PLANAR RR-ROBOT



direct kinematics:

$$p_x = l_1 c_1 + l_2 c_{12}$$

$$p_y = l_1 s_1 + l_2 s_{12}$$

analitic jacobian:

$$\dot{p} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix} \dot{q} = J(q)\,\dot{q}$$

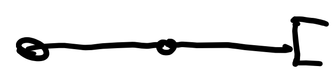
square jacobian

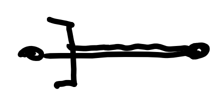
$$\det(J(q)) = -l_1 l_2 s_1 c_{12} - \cancel{l_2^2 s_{12} c_{12}} + l_1 l_2 c_1 s_{12} + \cancel{l_2^2 c_{12} s_{12}}$$

$$= -l_1 l_2 (s_1 c_{12} - c_1 s_{12}) = -l_1 l_2 \sin(\cancel{q_1} - (\cancel{q_1} + q_2)) =$$

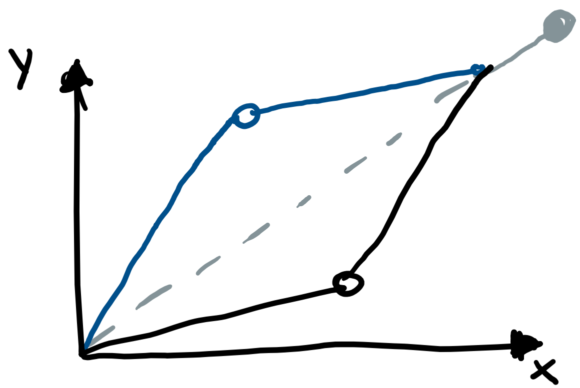
$$= \boxed{l_1 l_2 \sin q_2}$$

singularity: $\sin q_2 = 0$ $\Rightarrow$ no dependency on q_1

(A) $q_2 = 0$ stretched 

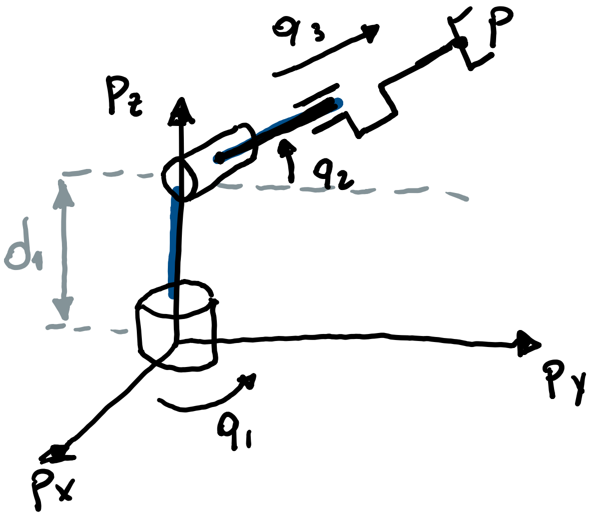
(B) $q_2 = \pi$ folded 

NOTE: boundary sing. separate the configuration space that have distinct inverse kinematic solutions (i.e elbow up/down)



singularity → 2 IK solutions collapse to 1

EXERCISE 1.2: SINGULARITY OF RRP ARM



direct kinematics

$$p_x = q_3 c_2 c_1$$
$$p_y = q_3 c_2 s_1$$
$$p_z = d_1 + q_3 s_2$$

analytical jacobian

$$\dot{p} = \begin{bmatrix} -q_3 s_1 c_2 & -q_3 c_1 s_2 & c_1 c_2 \\ q_3 c_1 c_2 & -q_3 s_1 s_2 & s_1 c_2 \\ 0 & q_3 c_2 & s_2 \end{bmatrix} \dot{q} = J(q)\,\dot{q}$$

$$\det(J(q)) = q_3^2 c_2$$

singularities: $c_2 = 0$ or $q_3 = 0$

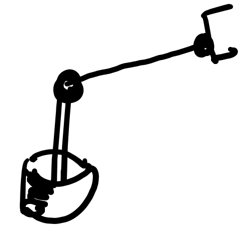
(A) $q_2 = \pm \frac{\pi}{2}$ simple singularity $\Rightarrow$ rank = 2 $\Rightarrow$ you lose mobility in one direction

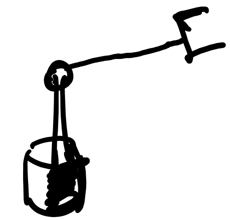
(B) $q_3 = 0$ third joint fully retracted $\Rightarrow$ double singularity $\Rightarrow$ rank drops to 1 $\Rightarrow$ lose mobility in two directions

these singularities are internal to the workspace

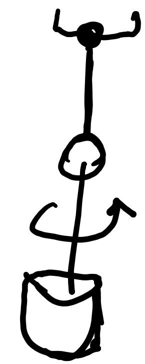
HOW MANY IK SOLUTIONS ?

- $q_3 \neq 0 \quad q_2 \neq \pm\frac{\pi}{2} \Rightarrow$ 2 solutions (LEFT / RIGHT)

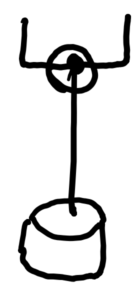




- $q_3 \neq 0 \quad q_2 = \pm\frac{\pi}{2} \Rightarrow \infty^1$ solutions (q_1)

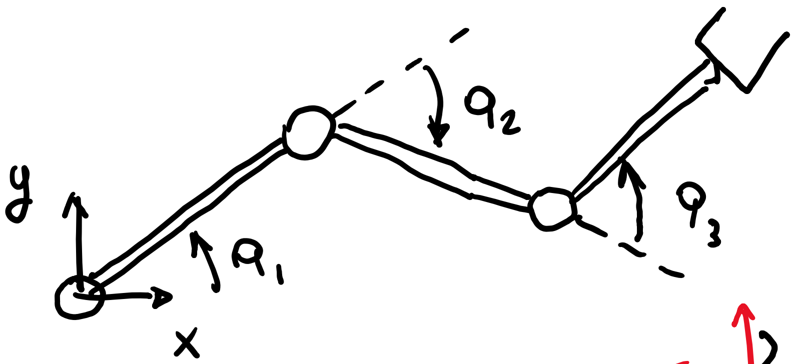


- $q_3 = 0 \Rightarrow \infty^2$ solutions
  any q_1 and q_2 bring to same point



all singularities are on exis of the shoulder axis

EXERCISE 2 : SUBSPACES OF PLANAR 3R ROBOT

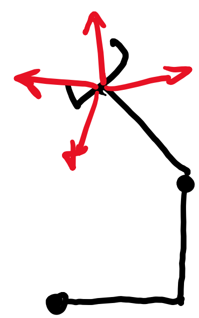


$\dot{p} = J(q)\,\dot{q}$

$p \in \mathbb{R}^2 \qquad q \in \mathbb{R}^3$

$J = {}_{2}\overset{3}{\boxed{}}$

$\boxed{2.1}$ $q = (0, \pi/2, \pi/4)$

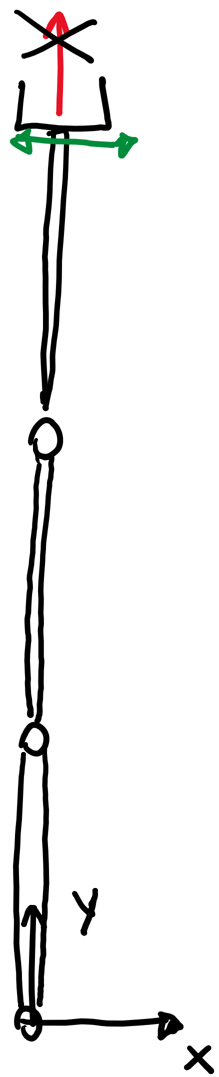


- can I go everywhere ? $\Rightarrow$ check rank (J)
- can I move joints such that end effector does not move ? $\Rightarrow$ check if $\exists$ Null space basis and its dimension

MATLAB: NULL (J)

$\boxed{2.2}$ what is dimension of the range space $\Rightarrow$ MATLAB: ORTH (J)

$\boxed{2.3}$ repeat $\boxed{2.2}$, $\boxed{2.3}$ for the singular configuration $q = \left(\frac{\pi}{2}, 0, 0\right)$



$$J = \begin{bmatrix} -3 & -2 & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

- rank $(J) = ?$ 1 $\Rightarrow$ cannot move in y direction
- Null $(J) = \begin{bmatrix} -0{,}66 & -0{,}33 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \Rightarrow \dim(N(J)) = 2$
- range $(J) = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \Rightarrow$ only direction the E-E can move

INVERSION OF DIFFERENTIAL KINEMATICS

Task $\boxed{\text{velocity}}$ Joint $\boxed{\text{velocity}}$

$$v \longrightarrow \dot{q} = J^{-1}(q)\, v$$

requires: J squared
non-singular

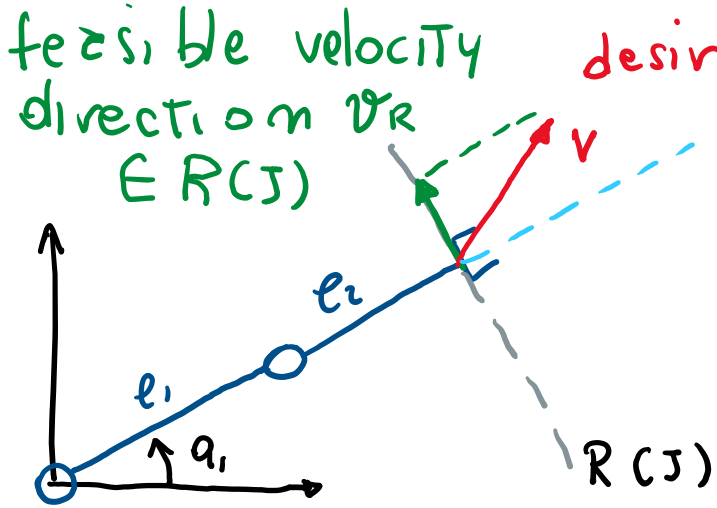
PROBLEM

- singularity : J is not invertible

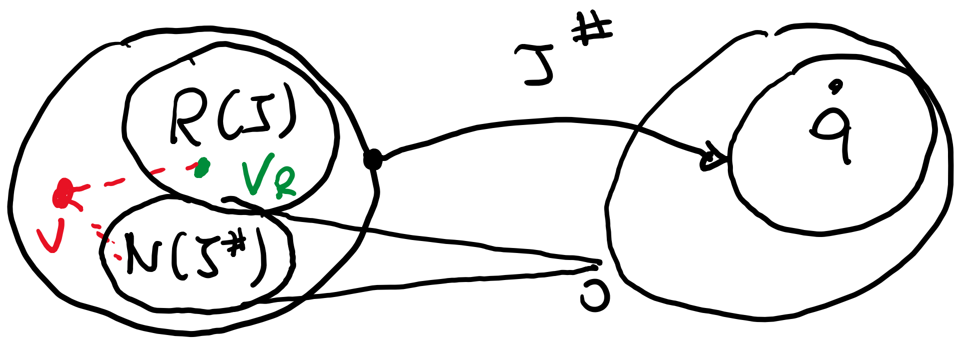
- for redundant robots : J is rectangular $\Rightarrow$ inverse is not defined $\Rightarrow$ use $J^{\#}$

if There is a loss of rank in J Then $J^{\#} v$ finds The solution $\dot{q}$ closest to v (i.e. that minimizes The error $\| J\dot{q} - v \|$)

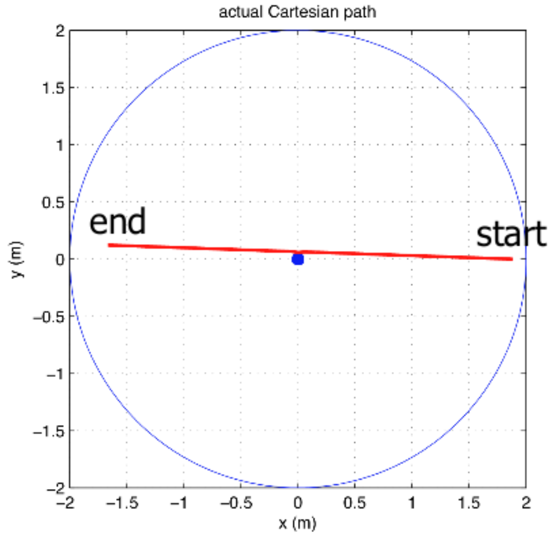
EXAMPLE

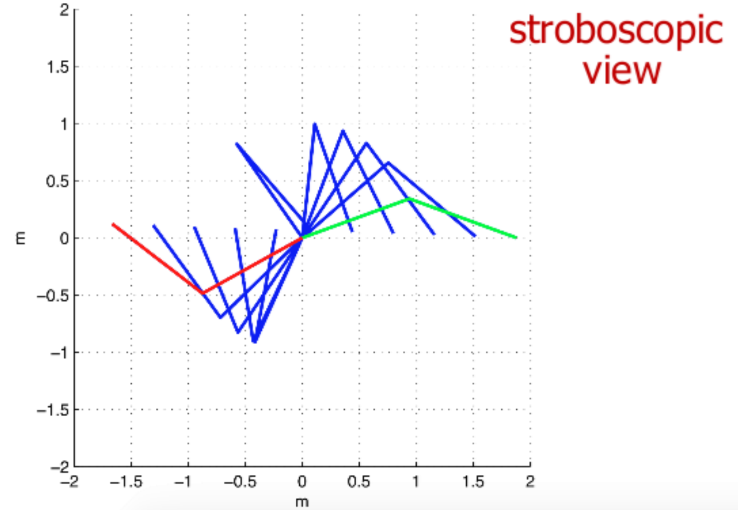


$\dot{q} = J^{\#} v$ realizes only The component V_R of the velocity V That is in The "range" of J

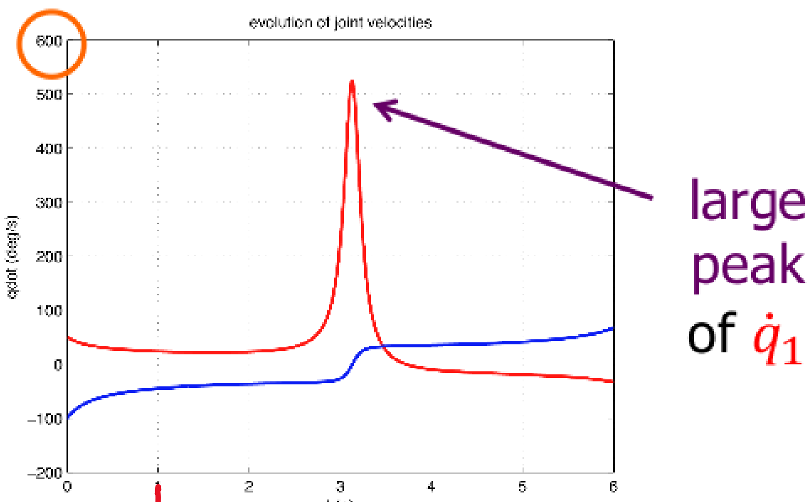


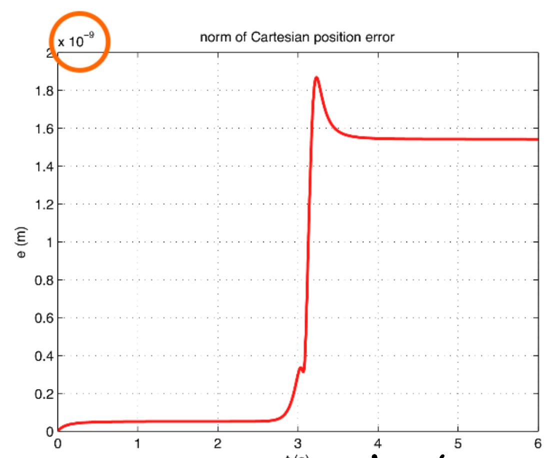
SIMULATION PLANAR RR ROBOT





- close to singularity joint velocities can be very high





↳ not feasible but if I saturate velocity I get a position tracking error

DISTANCE TO SINGULARITY

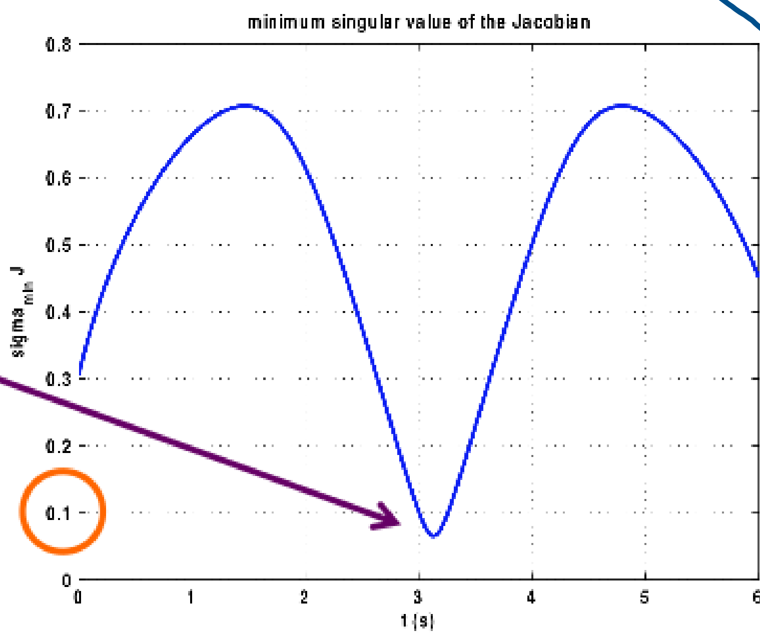
is possible To check how close the robot is to a singularity checking the value of the smallest singular value of the J matrix

$\hookrightarrow \sigma_{min}$

$\sigma_i \triangleq \sqrt{\lambda_i(\underbrace{J^T J}_{\text{square matrix}})}$ SKINNY

$\sigma_i \triangleq \sqrt{\lambda_i(J J^T)}$ FAT

close to singular case



We want a more "robust" inversion method

DAMPED LEAST SQUARE METHOD

Try to find a trade-off between realizying the velocity task at the certesian level and have small joint velocity

$\Rightarrow$ again we can cast this as an optimization problem with 2 terms in the cost:

$$\min_{\dot q} \underbrace{\boxed{\frac{1}{2}\|J\dot q - v\|}}_{A} + \underbrace{\boxed{\frac{\lambda}{2}\|\dot q\|^2}}_{B} \qquad \lambda > 0 \qquad \lambda \downarrow\downarrow$$

$\downarrow \tilde{v}$

$$\dot q = J^{-1} v$$

A $\Rightarrow$ least squar

B $\Rightarrow$ damped

- The solution can be obtained in closed form (put gradient of cost $=0$)

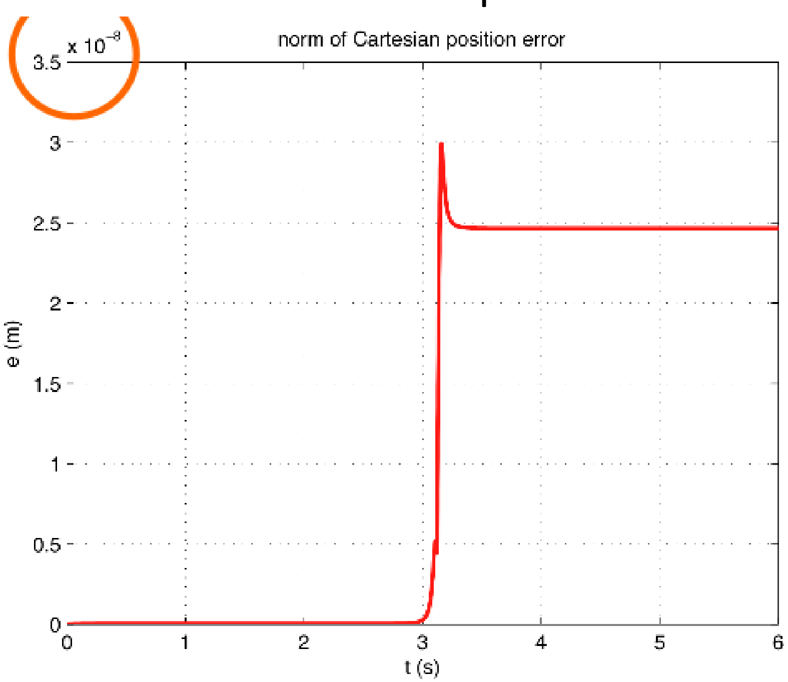
$$\boxed{\dot q^* = (J^T J + \lambda I_m)^{-1} J^T} \checkmark$$

$$\boxed{\dot q^* = J^T (J J^T + \lambda I_m)^{-1}} \checkmark \rightarrow \text{equivalent for redundant robots}$$

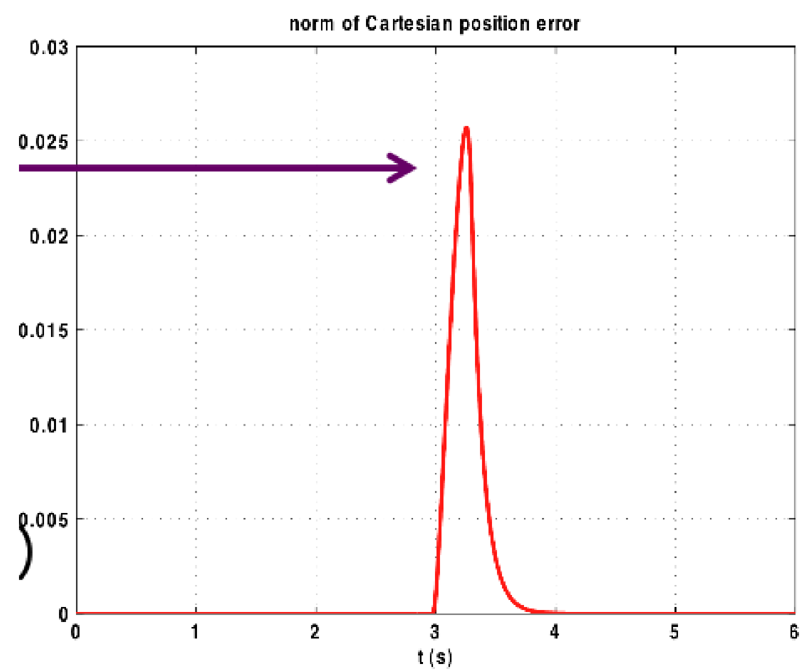
(+) can be used both for recovering from singularity of a square jacobian or if you have a rectangular jacobian to invert (including also the case when it becomes singular).

(−) $\lambda \uparrow\uparrow$ you penalize velocity but you get bigger tracking error.

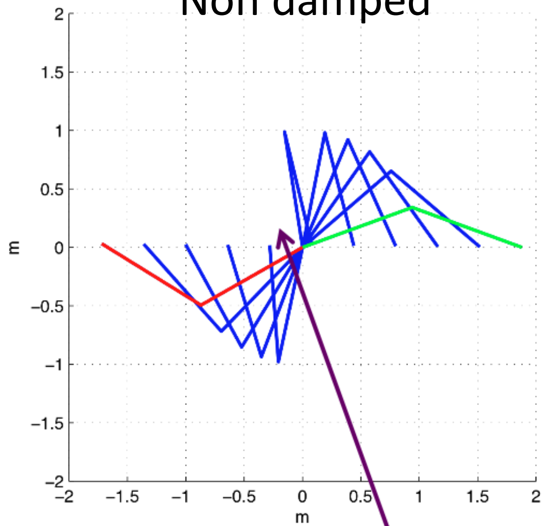
Non damped



Damped

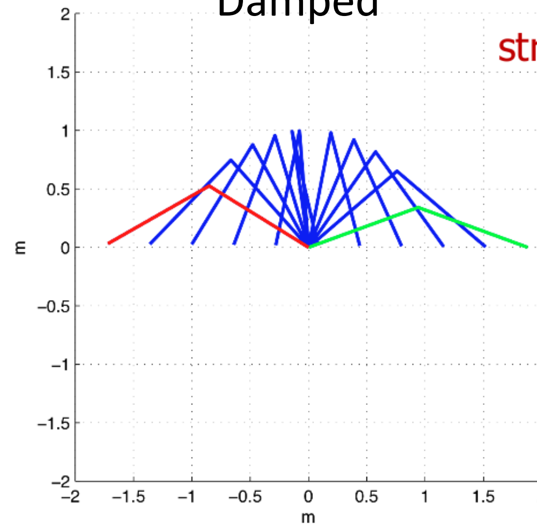


Non damped



Damped

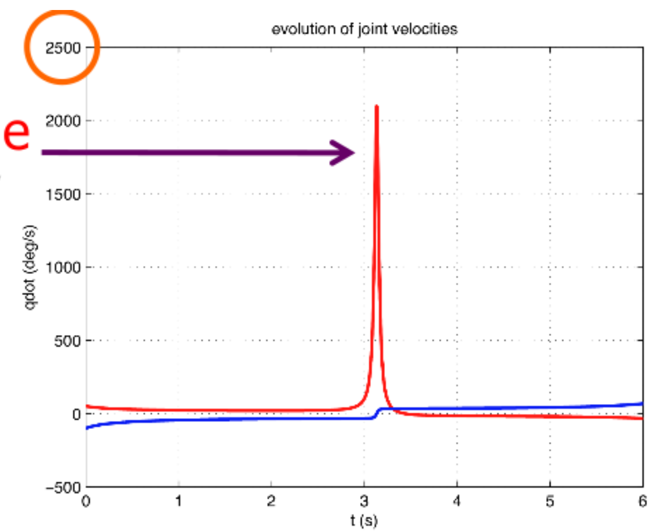
stroboscopic views

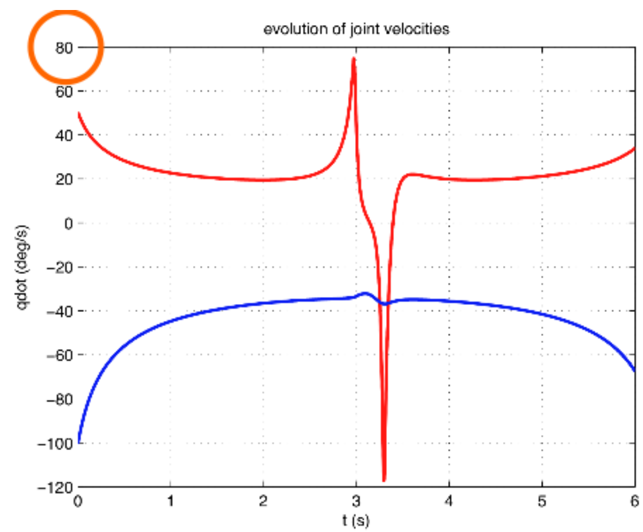


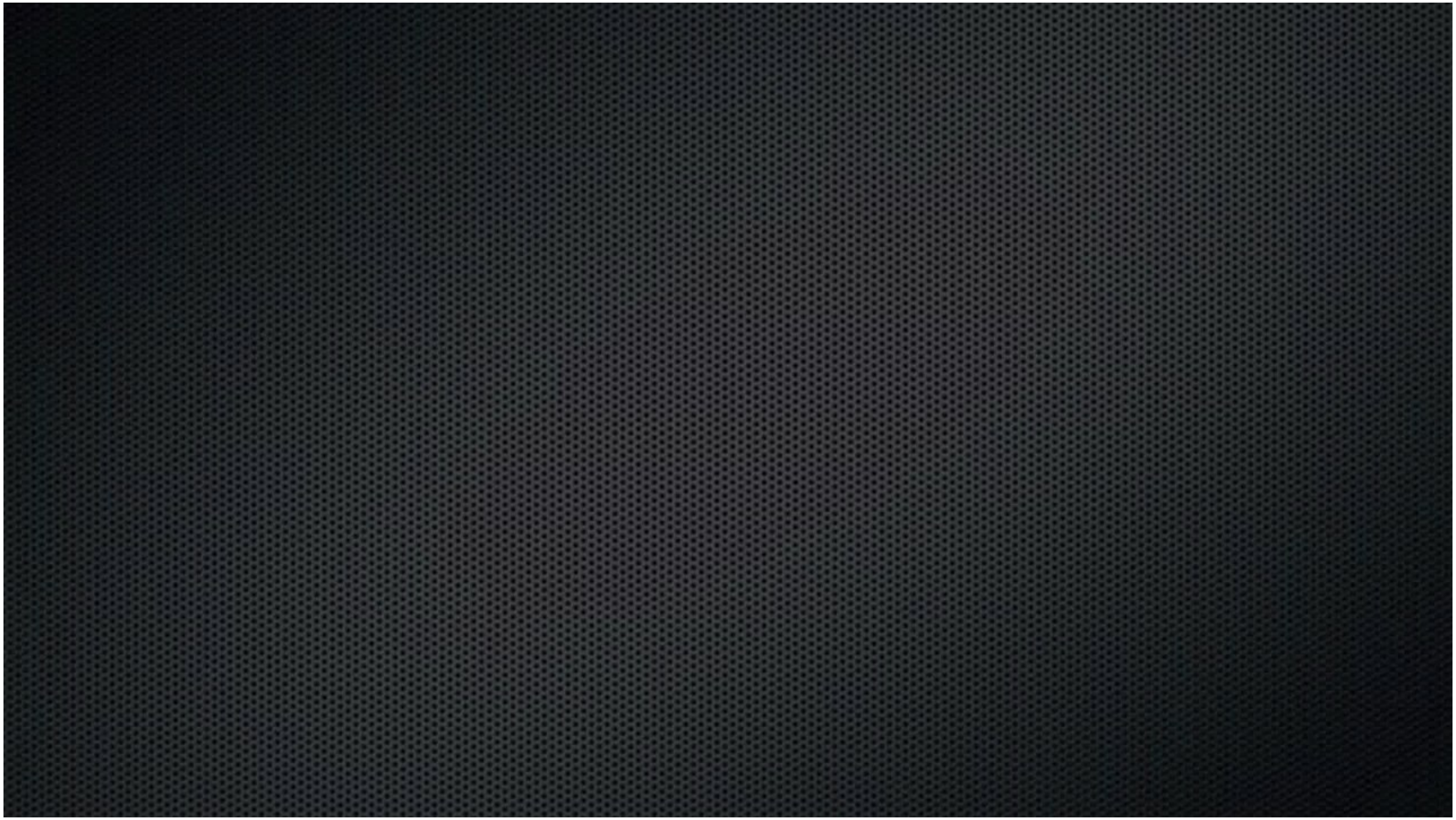
here, a very fast reconfiguration of first joint ...

a completely different inverse solution, around/after crossing the region close to the folded singularity

extremely large peak velocity of first joint!!







https://www.youtube.com/watch?v=zlGCurgsqg8

HIGHER-ORDER DIFFERENTIAL INVERSION

$$\dot{r} = J(q)\,\dot{q}$$

r: generic task function

$$\ddot{r} = J(q)\,\ddot{q} + \dot{J}(q)\,\dot{q}$$

$$\Downarrow$$

$$\ddot{q} = J(q)^{-1}\left(\ddot{r} - \dot{J}(q)\,\dot{q}\right)$$

IK

SUMMARY

$$\boxed{\begin{aligned} q &= IK(r) \\ \dot{q} &= J^{-1}\,\dot{r} \\ \ddot{q} &= J^{-1}\left(\ddot{r} - \dot{J}\,\dot{q}\right) \end{aligned}}$$