Robotics 2

Robots with kinematic redundancy

Part 1: Fundamentals

Prof. Alessandro De Luca

Dipartimento di Ingegneria Informatica Automatica e Gestionale Antonio Ruberti

Sapienza Università di Roma

Redundant robots

- direct kinematics of the task $r = f(q)$

$$f: \mathrm{Q} \rightarrow \mathrm{R}$$

joint space (dim Q = N)

task space (dim R = M)

- a robot is (kinematically) redundant for the task if $N > M$

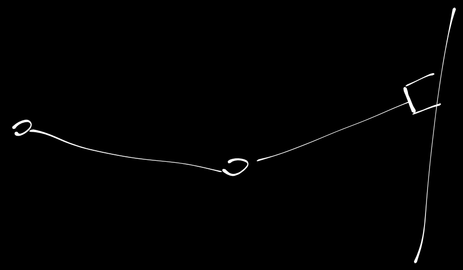
  (more degrees of freedom than strictly needed for executing the task)

- r may contain the position and/or the orientation of the end-effector or, more in general, be any parameterization of the task (even not in the Cartesian workspace)

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

Example 1

When is a 2R redundant?



- when I keep x or y constant or with history

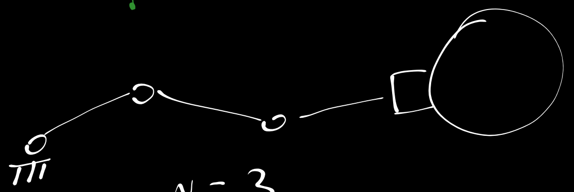
> $p_x = \text{const}$

> $p_x = p_x(t)$

- just orientation

In this case I could maintain always the same inverse kinematics solution, unless I cross a singularity

Example 2



$N = 3$

Case 1

- The robot has to trace the circle
- $M = 2 \Rightarrow$ REDUNDANT

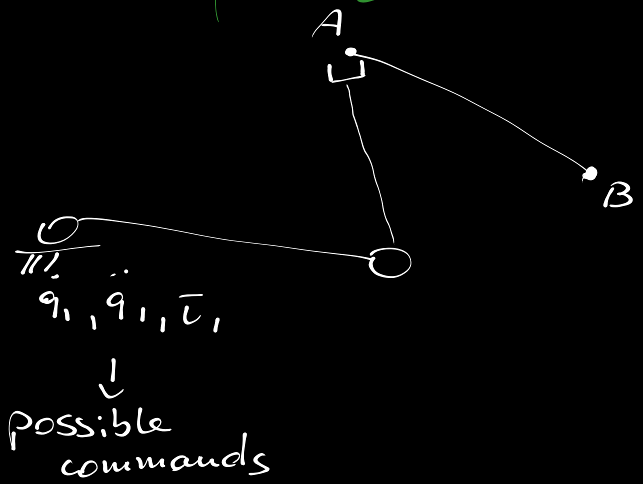
→ In this case I will have infinite inverse kinematics solutions

CASE 2

- The robot has to trace the circle while always pointing towards the center
- $M = 3 \Rightarrow$ NOT REDUNDANT

→ In this case I will still have multiple, but numerable inverse kinematics solutions

Example 3 - Joint velocity Limits



$$|\dot{q}_i| \leq V_i \qquad i = 1, 2$$

$$|\ddot{q}_i| \leq A_i$$

$$|\tau_i| \leq T_i$$

Some E-E tasks and their dimensions

TASKS [for the robot end-effector (E-E)]	dimension 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 ($M = 3$)

Typical cases of redundant robots

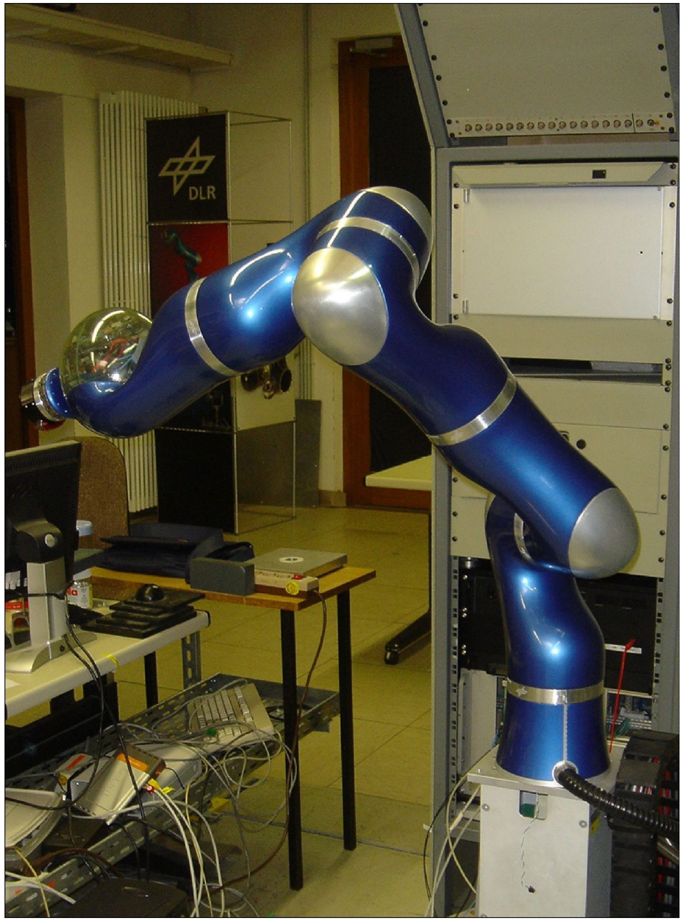
- 6R robot mounted on a linear track/rail
  - 7 dofs for positioning and orienting its end-effector in 3D space
- 6-dof robot used for arc welding tasks
  - task does not prescribe the final roll angle of the welding gun
- dexterous robotic hands
- multiple cooperating manipulators
- manipulator on a mobile base
- humanoid robots, team of mobile robots ...
- "kinematic" redundancy is not the only type…
  - redundancy of components (actuators, sensors)
  - redundancy in the control/supervision architecture

Uses of robot redundancy

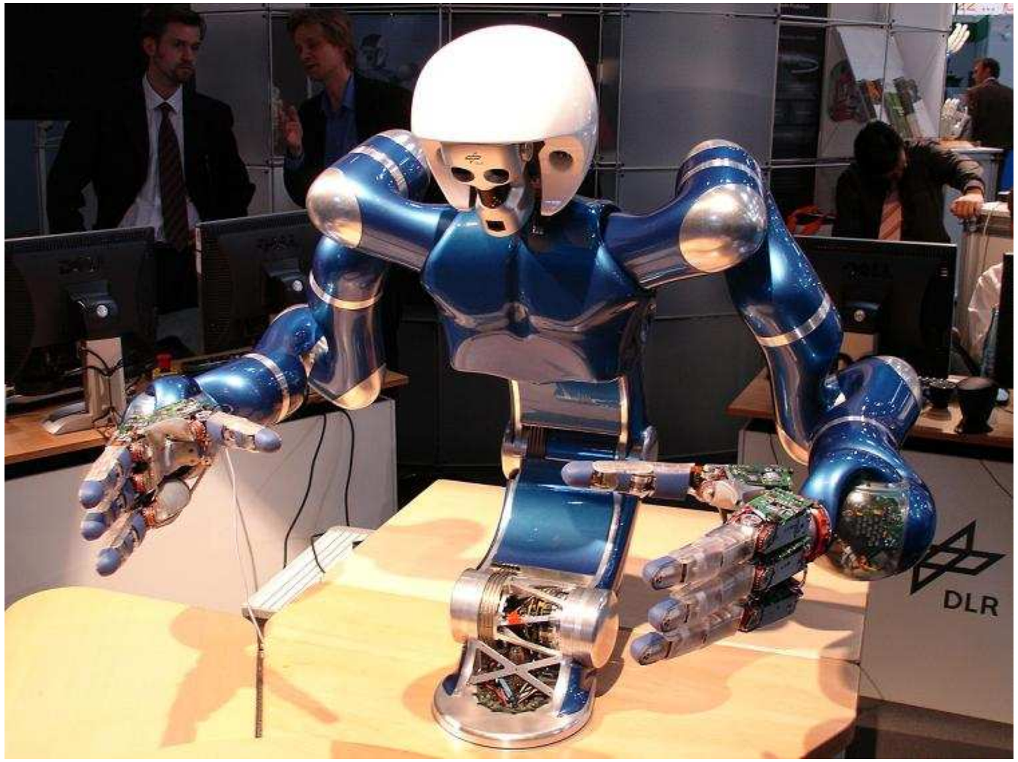
- avoid collision with obstacles (in Cartesian space) …
- … or kinematic singularities (in joint space)
- stay within the admissible joint ranges
- increase manipulability in specified directions
- uniformly distribute/limit joint velocities and/or accelerations
- minimize energy consumption or needed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

all objectives should be quantitatively "measurable"

DLR robots: LWR-III and Justin



7R LWR-III lightweight manipulator:
elastic joints (HD), joint torque sensing,
13.5 kg weight = payload



Justin two-arm upper-body humanoid:
43R actuated =
two arms (2×7) + torso (3*)
+ head (2) + two hands (2×12),
45 kg weight

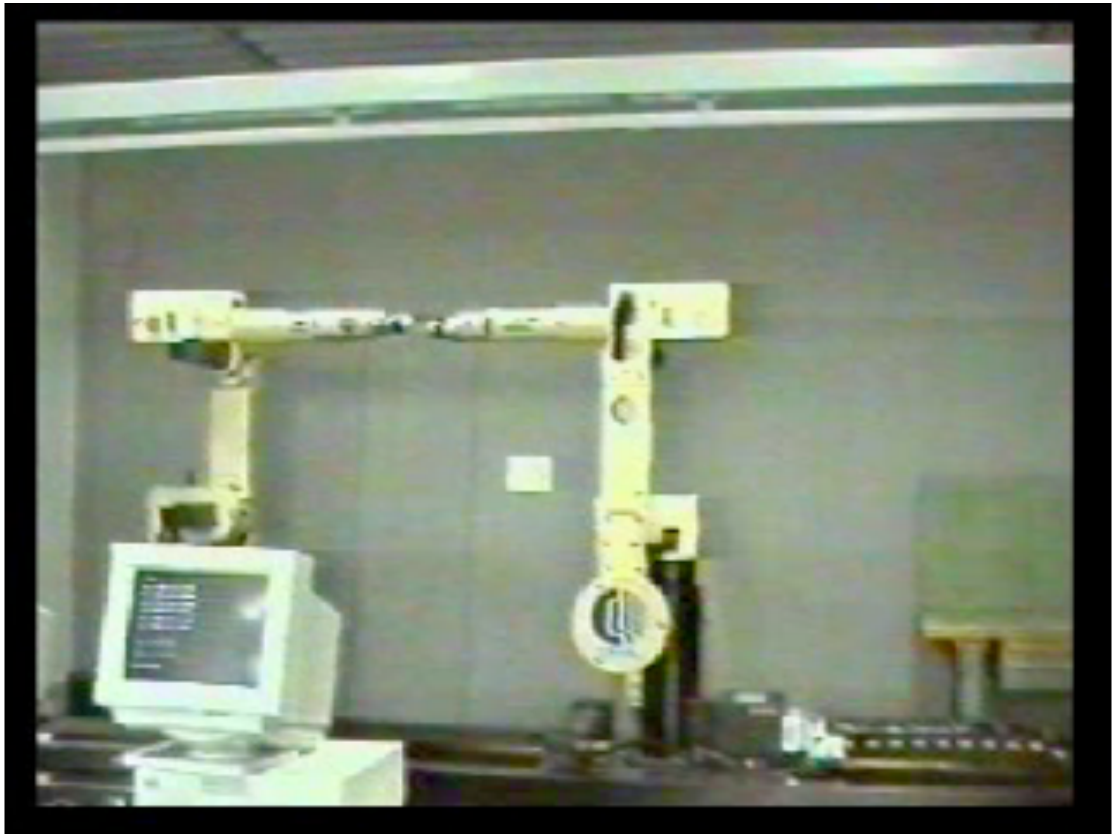
* = one joint is dependent on the motion of the other two

Justin carrying a trailer

video

motion planning for DLR Justin robot in the configuration space, avoiding Cartesian obstacles and using robot redundancy

Dual-arm redundancy



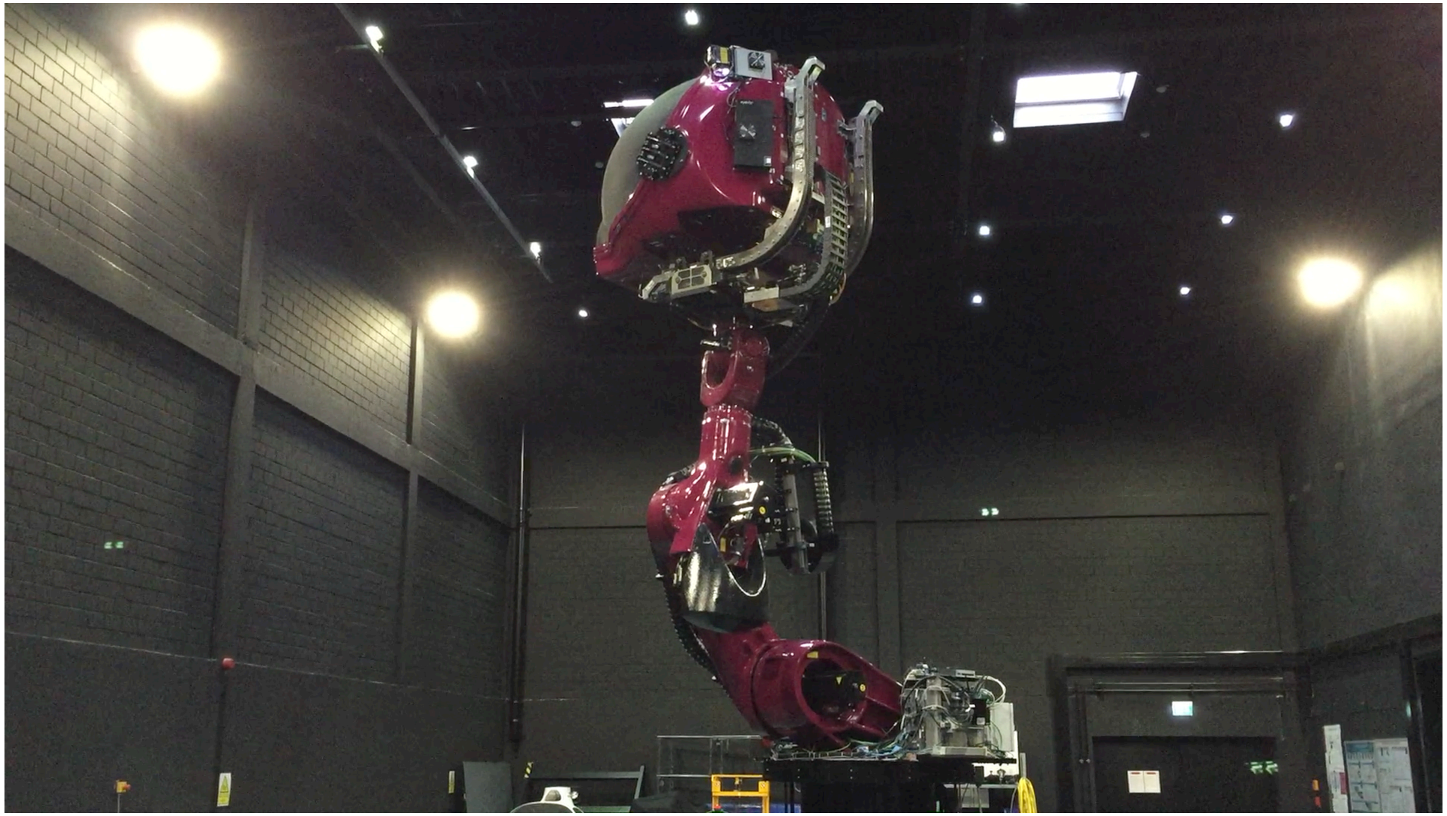
video

DIS, Uni Napoli

two 6R Comau robots, one mounted on a linear track (+1P)
coordinated 6D motion using the null-space of the right-side robot ($N - M = 1$)

Motion cueing from redundancy

video



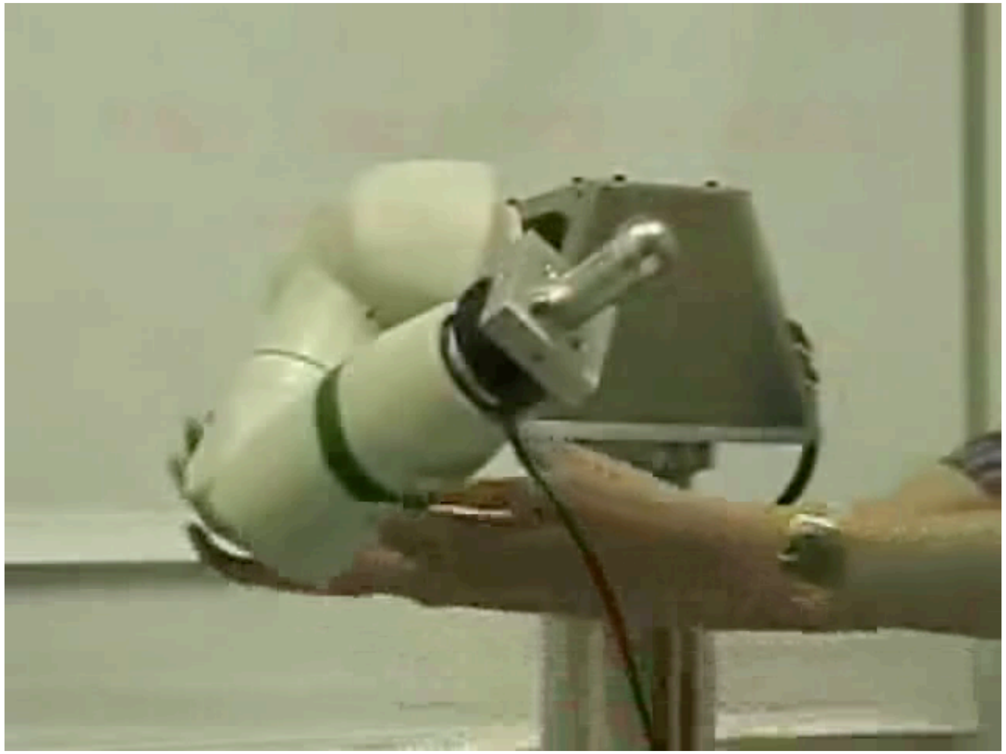
Max Planck Institute for Biological Cybernetics, Tübingen

a 6R KUKA KR500 mounted on a linear track (+1P) with a sliding cabin (+1R), used as a dynamic emulation platform for human perception ($N - M = 2$)

Self-motion

video

video



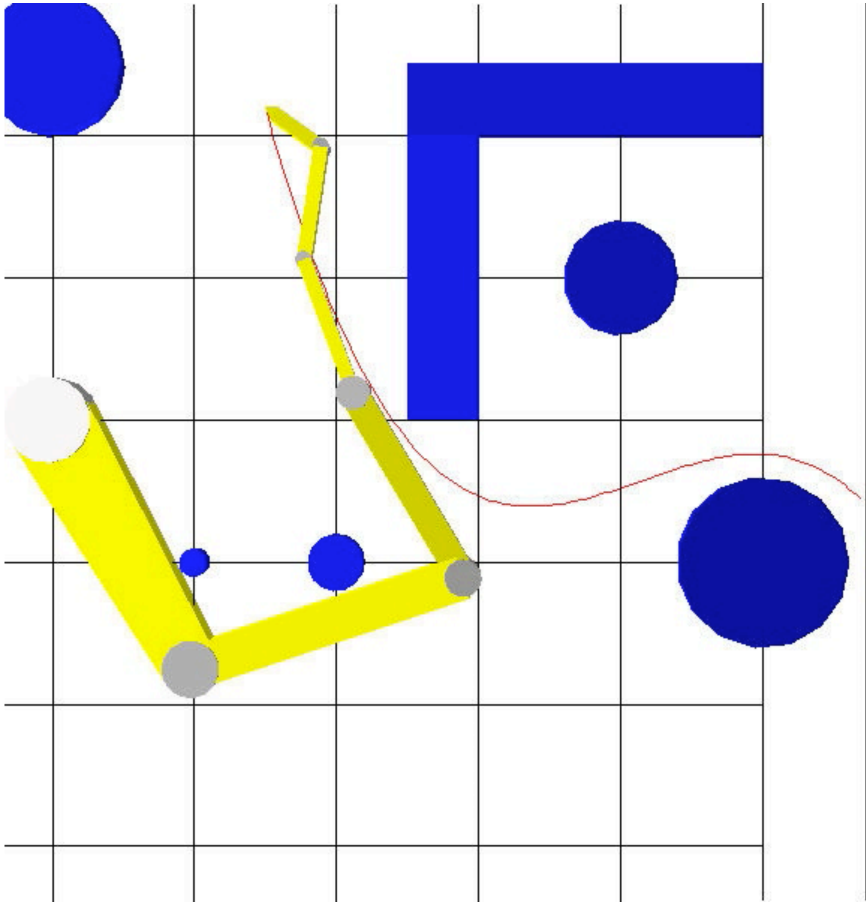
motion parametrized in 2 variables, the other are computed in order not to move the end-effector

Nakamura's Lab, Uni Tokyo

8R Dexter: self-motion with constant 6D pose of E-E ($N - M = 2$)

6R robot with spherical shoulder in compliant tasks for the Cartesian E-E position ($N - M = 3$)

Obstacle avoidance



video

6R planar arm moving on a given geometric path for the E-E ($N - M = 4$)

Disadvantages of redundancy

- potential benefits should be traded off against
  - a greater structural complexity of construction
    - mechanical (more links, transmissions, ...)
    - more actuators, sensors, ...
    - costs
  - more complicated algorithms for inverse kinematics and motion control

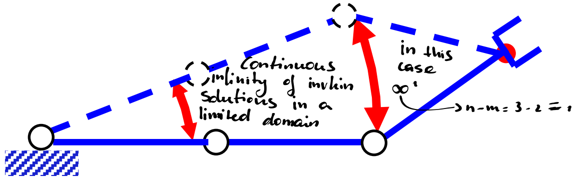
Redundancy changes things only for invkin

Inverse kinematics problem

- find $q(t)$ that realizes the task: $f(q(t)) = r(t)$ (at all times t)
- also in singularities
- infinite solutions exist when the robot is redundant
  (even for $r(t) = r$ = constant)

More columns than rows in the Jacobian => same column vectors are in the null space of J => you can move the corresponding joints, but the EE will stay still

$N = 3 > 2 = M$



r = constant
E-E position

- the robot arm may have "internal displacements" that are unobservable at the task level (e.g., not contributing to E-E motion)
  - these joint displacements can be chosen so as to improve/optimize in some way the behavior of the robotic system
- self-motion: an arm reconfiguration in the joint space that does not change/affect the value of the task variables r
- solutions are mainly sought at differential level (e.g., velocity)

WHY?

Redundancy resolution

via optimization of an objective function

I know the instantaneous velocity (maybe from sensors)

Local methods

given $\dot{r}(t)$ and $q(t)$, $t = kT_s$

this changes

objective function

optimization of $H(q, \dot{q})$

I can't change q because i'm already there

I choose $\dot{q}$ for time instant kT_s that optimizes $H(q, \dot{q})$ ← $\dot{q}(kT_s)$ ← ON-LINE

$$q((k+1)T_s) = q(kT_s) + T_s\,\dot{q}(kT_s)$$

first order Euler integration

discrete-time form

I know the whole trajectory

Global methods

given $r(t)$, $t \in [t_0, t_0 + T]$, $q(t_0)$

optimization of $\int_{t_0}^{t_0+T} H(q, \dot{q})\,dt$

$$q(t),\ t \in [t_0, t_0 + T]$$

OFF-LINE

(+) quick
updatable at any time
(−) no guarantee of success

relatively EASY
(a LQ problem)
[Linear quadratic]

quite DIFFICULT
(nonlinear TPBV problems arise)

(+) globality
(−) more difficult to compute
relies on a priori information

Local resolution methods

three classes of methods for solving $\dot{r} = J(q)\dot{q}$

solution without any additional information => H embedded in the problem

1 Jacobian-based methods (here, analytic Jacobian in general!)

among the infinite solutions, one is chosen, e.g., that minimizes a suitable (possibly weighted) norm

→ of the solution

2 null-space methods

a term is added to the previous solution so as not to affect execution of the task trajectory, i.e., belonging to the null-space $\mathcal{N}(J(q))$

3 task augmentation methods

redundancy is reduced/eliminated by adding $S \leq N - M$ further auxiliary tasks (when $S = N - M$, the problem has been "squared")

$$r = f(q) \quad \Rightarrow \quad \dot{r} = J(q)\dot{q}$$

1 Jacobian-based methods

we look for a solution to $\dot{r} = J(q)\dot{q}$ in the form

$$J = \underbrace{\boxed{}}_{N}\Big\}M \qquad \underbrace{\overset{\mathbb{R}^n}{\dot{q}} = \overset{\mathbb{R}^{n\times m}}{K(q)}\,\overset{\mathbb{R}^m}{\dot{r}}} \qquad \text{dimension of transpose of } J \leftarrow K = \underbrace{\boxed{}}_{M}\Big\}N$$

• linear in $\dot{r}$

• if $\dot{r} = 0 \Rightarrow \dot{q} = 0$

minimum requirement for K: $J(q)K(q)J(q) = J(q) \longrightarrow$ weaker requirement than $JK = I$

(➡ K = generalized inverse of J)

I'm asking the generalized inverse to generate the $\dot{q}$ that realizes $\dot{r}$ $\longrightarrow$ So $\dot{r}$ is a lincomb of J with weights $\dot{q}$

$\Downarrow$

But $\dot{r}$ lies to feasible

$\Downarrow$

$$\forall \dot{r} \in \mathcal{R}(J(q)) \Rightarrow J(q)\underbrace{[K(q)\dot{r}]}_{\dot{q}} = J(q)K(q)J(q)\dot{q} = J(q)\dot{q} = \dot{r}$$

example:

if $\underset{\mathbb{R}^{3\times 6}}{J} = [\underset{\mathbb{R}^{3\times 3}}{J_a} \;\; \underset{\mathbb{R}^{3\times 3}}{J_b}]$, $\det(J_a) \neq 0$, one such generalized inverse of J is $\overset{\mathbb{R}^{6\times 3}}{K_r} = \begin{pmatrix} J_a^{-1} \\ 0 \end{pmatrix}$

$\Downarrow$ full row rank

(actually, this is a stronger right-inverse)

Pseudoinverse $\longrightarrow$ treats all joints equally

$$\dot{q} = J^{\#}(q)\dot{r}$$

... a very common choice: $K = J^{\#}$

- $J^{\#}$ always exists, and is the unique matrix satisfying

$$J\,J^{\#}\,J = J \qquad J^{\#}\,J\,J^{\#} = J^{\#}$$

$$\left(J\,J^{\#}\right)^T = J\,J^{\#} \qquad \left(J^{\#}\,J\right)^T = J^{\#}\,J$$

- if J is full (row) rank, $J^{\#} = J^T(J\,J^T)^{-1}$; else, it is computed numerically using the SVD (Singular Value Decomposition) of J (pinv of Matlab)
- the pseudo-inverse joint velocity is the only that minimizes the norm $\|\dot{q}\|^2 = \dot{q}^T\dot{q}$ among all joint velocities that minimize the task error norm $\|\dot{r} - J(q)\dot{q}\|^2$
- if the task is feasible ($\dot{r} \in \mathcal{R}(J(q))$), there will be no task error

Weighted pseudoinverse

$$\dot{q} = J_W^{\#}(q)\dot{r}$$

another choice: $K = J_W^{\#}$

- the solution $\dot{q}$ minimizes the weighted norm

$$\|\dot{q}\|_W^2 = \dot{q}^T W \, \dot{q}$$

$W > 0$, symmetric (often diagonal)

- if J is full (row) rank, $\boxed{J_W^{\#} = W^{-1}J^T(JW^{-1}J^T)^{-1}}$
- large weight $W_i \Rightarrow$ small $\dot{q}_i$
  - larger weights for proximity joints (carrying/moving more "mass")
  - weights chosen proportionally to the inverse of the joint ranges
- it is NOT a "pseudoinverse" (4th relation does not hold), but it shares similar properties

$q_i \in [q_{i,max} \, ; \, q_{i,min}]$

$\hookrightarrow W_i = \dfrac{1}{(q_{i,max} - q_{i,min})^2}$

with limited joint range ∴ prefer small joint velocities

⇓

bigger corresponding weights

⇓

we could use also the REMAINING joint ranges !!

the nearest joints to its limit will move less faster

$$J_W^{\#} J = W^{-1}J^T(JW^{-1}J^T)^{-1}J \neq (J_W^{\#} J)^T$$

- having a W with equal weights on the diagonal (in a way a scaling factor) it's just like having a normal pseudoinverse without weighting

Singular Value Decomposition (SVD)

- the `SVD` routine of Matlab applied to J provides two orthonormal matrices $U_{M\times M}$ and $V_{N\times N}$, and a matrix $\Sigma_{M\times N}$ of the form

$$\Sigma = \left(\begin{array}{cccc|c} \sigma_1 & & & & \\ & \sigma_2 & & & \\ & & \ddots & & 0_{M\times(N-M)} \\ & & & \sigma_M & \end{array}\right)$$

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_\rho > 0$$
$$\sigma_{\rho+1} = \cdots = \sigma_M = 0$$

singular values of J

where $\rho = \text{rank}(J) \leq M$, so that their product is

min 28
lez. 4.1

$$\boxed{J = U\Sigma V^T}$$

- the columns of U are eigenvectors of $J\,J^T$ (associated to its non-negative eigenvalues σ_i^2), the columns of V are eigenvectors of $J^T J$
- the last $N-\rho$ columns of V are a basis for the null space of J

$$Jv_i = \sigma_i u_i \quad (\text{i} = 1,\cdots,\rho)$$

$$\boxed{Jv_i = 0 \quad (\text{i} = \rho+1,\cdots,N)}$$

TBD

Computation of pseudoinverses

- show that the pseudoinverse of J is equal to

$$J = U\Sigma V^T \quad \Rightarrow \quad J^{\#} = V\Sigma^{\#}U^T \qquad \Sigma^{\#} = \left(\begin{array}{ccc} \frac{1}{\sigma_1} & & \\ & \ddots & \\ & & \frac{1}{\sigma_\rho} \\ & & & 0_{(M-\rho)\times(M-\rho)} \\ \hline & & 0_{(N-M)\times M} & \end{array}\right)$$

  for any rank ρ of J

- show that matrix $J_W^{\#}$ appears when solving the constrained linear-quadratic (LQ) optimization problem (with $W > 0$, symmetric, and assuming J of full rank)

$$\min \frac{1}{2}\|\dot{q}\|_W^2 \quad \text{s.t.} \quad J(q)\dot{q} - \dot{r} = 0$$

  and that the pseudoinverse is a particular case for $W = I$

- show that a weighted pseudoinverse of J can be computed by SVD/pinv as

$$J_{aux} = JW^{-1/2} \qquad\qquad J_W^{\#} = W^{-1/2}\,\mathrm{pinv}(J_{aux})$$

applies equally to square and non-square matrices

Singularity robustness

Damped Least Squares (DLS)

With the normal Pseudoinverse, going towards a singularity means that a $\sigma \to 0$, which means that in the $J^{\#}$ there will be a very big value

I damp velocities (tolerating an error)

unconstrained minimization of a **suitable** objective function

$$\min_{\dot{q}} H(\dot{q}) = \frac{\mu^2}{2}\|\dot{q}\|^2 + \frac{1}{2}\|\dot{r} - J\dot{q}\|^2$$

unconstrained optimization because no hard constraint on task

compromise between large joint velocity and task accuracy

$\mu \propto$ damping

proportional

small $\mu \Rightarrow$ small error, large velocity

big $\mu \Rightarrow$ big error, small velocity

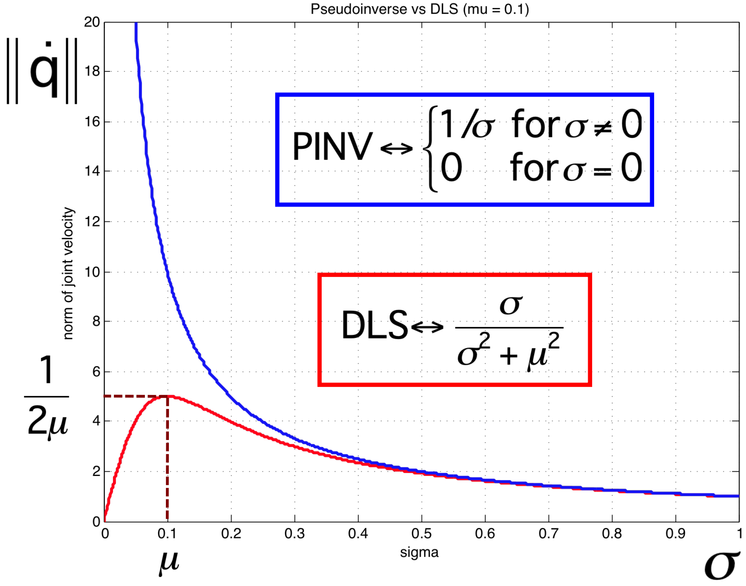
SOLUTION $\dot{q} = J_{DLS}(q)\dot{r} = J^T\underbrace{(\overbrace{J\,J^T}^{\text{semidef}} + \mu^2 I_M)}_{\text{posdef matrix} \Rightarrow \text{invertible} \Rightarrow \text{don't care about } \rho(J)}^{-1}\dot{r}$

- induces a robust behavior when crossing singularities, but in its basic version gives always a task error $\dot{e} = \mu^2(J\,J^T + \mu^2 I_M)^{-1}\dot{r}$ (as for $N = M$)
  - J_{DLS} is not a generalized inverse K → $\dot{r}$ may be feasible but with a too large $\dot{q}$ (which I'll ignore)
  - using SVD: $J = U\,\Sigma\,V^T \Rightarrow J_{DLS} = V\Sigma_{DLS}U^T$, $\Sigma_{DLS} = \left(\begin{array}{c|c} \underset{\rho\times\rho}{\boxed{diag\left\{\dfrac{\sigma_i}{\sigma_i^2 + \mu^2}\right\}}} & 0_{(M-\rho)\times(M-\rho)} \\ \hline 0_{(N-M)\times\rho} & 0_{(N-M)\times(M-\rho)} \end{array}\right)$

    $\dot{e} = \mu^2(J\,J^T + \mu^2 I_M)^{-1}\dot{r}$
  - choice of a variable damping factor $\mu^2(q) \geq 0$, function of the minimum singular value $\sigma_\rho(q) > 0$ of $J \cong$ a measure of distance from a singularity (if $\rho = M$) or of further loss of rank (when $\rho < M$)

    as soon as the min sing val drops below a threshold, I switch on the damped least squares
  - numerical filtering: introduces damping only/mostly in non-feasible directions for the task (see Maciejewski and Klein, *J of Rob Syst*, 1988)

Behavior of DLS solution

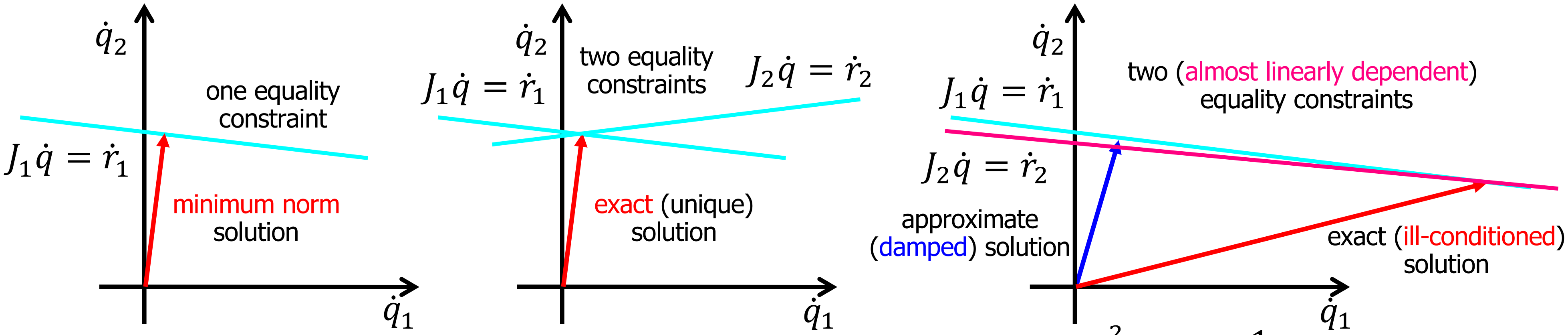


a. comparison of joint velocity norm with PINV (pseudoinverse) or DLS solutions

- in a task direction along a vector u of U, when the associated singular value $\sigma \longrightarrow 0$
- PINV goes to infinity (and then is 0 at $\sigma = 0$)
- DLS peaks a value of $1/2\mu$ at $\sigma = \mu$ (and then smoothly goes to 0...)

$f(\sigma) = \frac{\sigma}{\sigma^2+\mu^2} \Rightarrow f'(\sigma) = \frac{(\sigma^2+\mu^2) - 2\sigma^2}{(\sigma^2+\mu^2)^2} = \frac{\mu^2-\sigma^2}{(\sigma^2+\mu^2)^2} \overset{!!}{=} 0 \overset{!!}{\Rightarrow} \sigma = \mu$

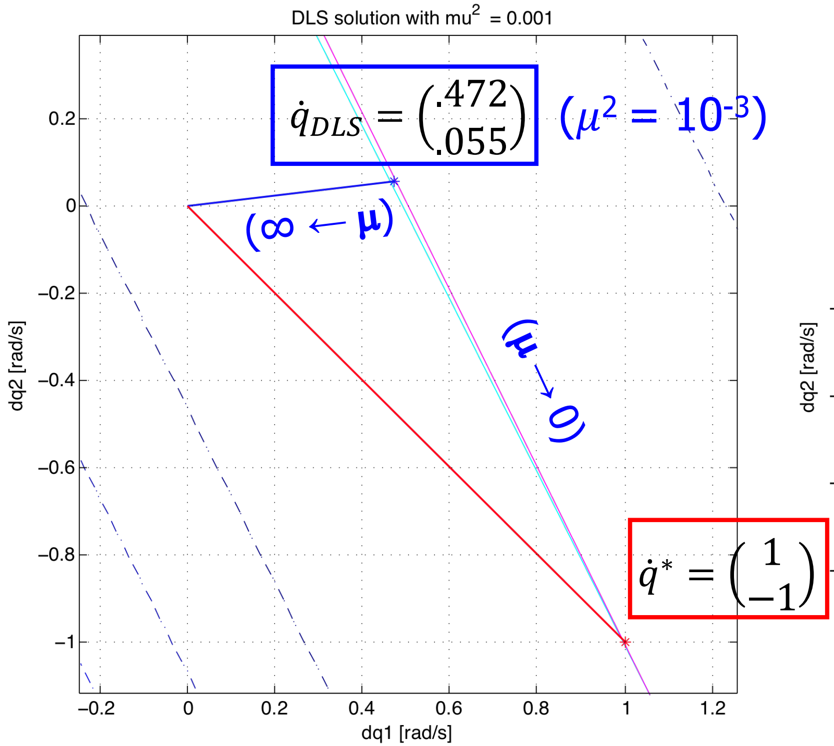
b. graphical interpretation of "damping" effect (here $M = N = 2$, for simplicity)

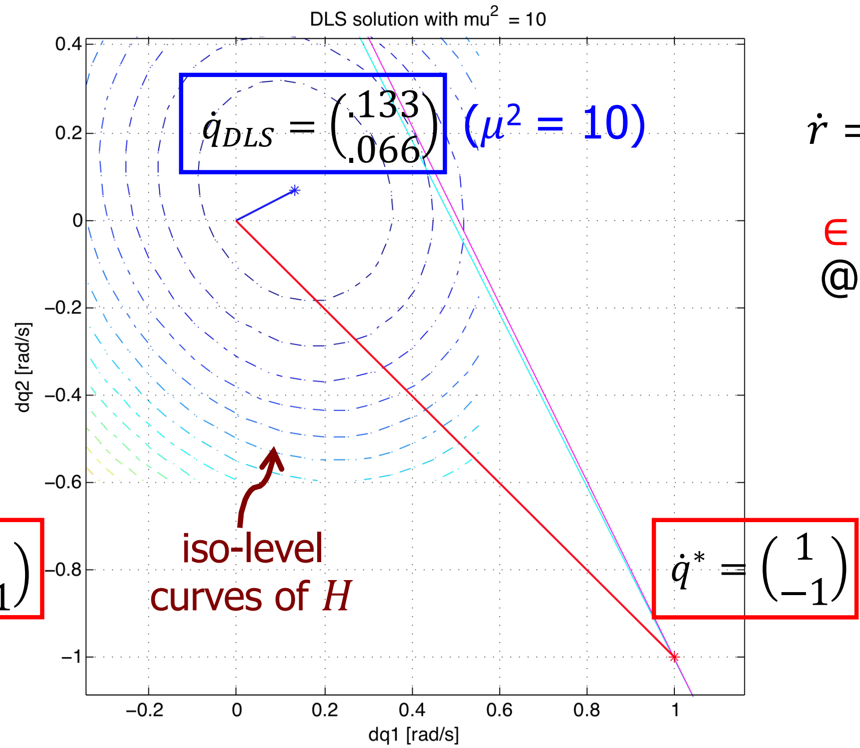


$$H(\dot{q}) = \frac{\mu^2}{2}\|\dot{q}\|^2 + \frac{1}{2}\|\dot{r} - J\dot{q}\|^2$$

Numerical example of DLS solution

planar 2R arm, unit links, close to (stretched) singular configuration $q_1 = 45°$, $q_2 = 1.5°$)





$$\dot{r} = \begin{pmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}$$

$\in \mathcal{R}(J)$ even @singularity!

exact solution (μ=0)

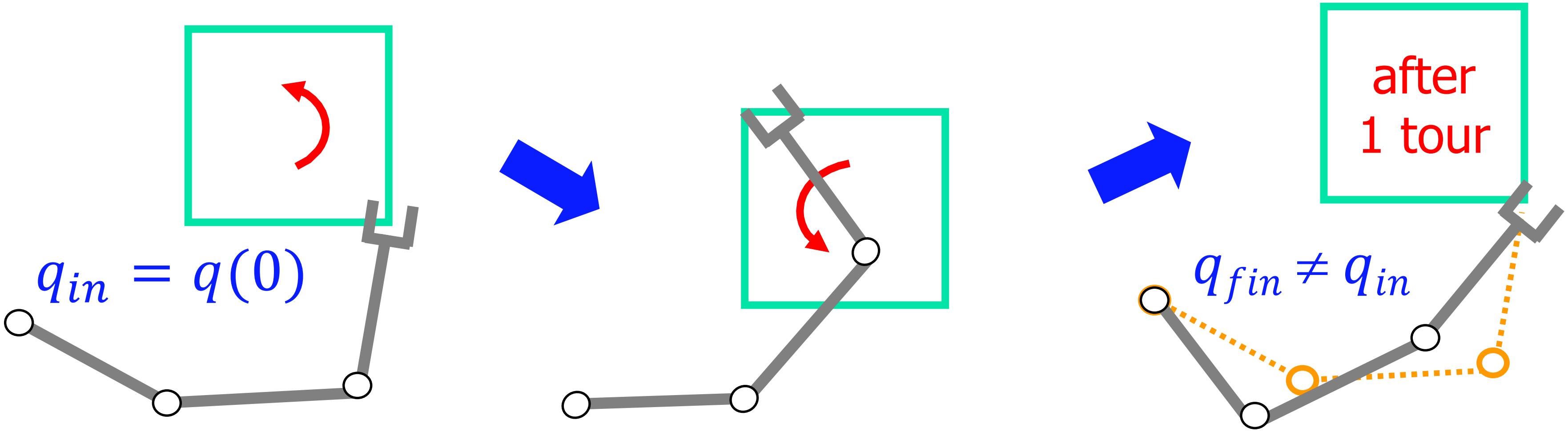
$$\boxed{H = \frac{\mu^2}{2}\|\dot{q}\|^2 + \frac{1}{2}\|\dot{r} - J\dot{q}\|^2}$$

μ^2	0	10^{-4}	10^{-3}	10^{-2}	10
$\|\dot{q}\|$	√2	.8954	.4755	.4467	.1490
$\|\dot{e}\|$	0	$6.6 \cdot 10^{-3}$	$1.4 \cdot 10^{-2}$	$1.6 \cdot 10^{-2}$	.6668
H_{min}	0	$7.7 \cdot 10^{-5}$	$2.2 \cdot 10^{-4}$	$1.2 \cdot 10^{-3}$	$3.4 \cdot 10^{-1}$

I only have objectives, not hard constraints → treated locally

Limits of Jacobian-based methods

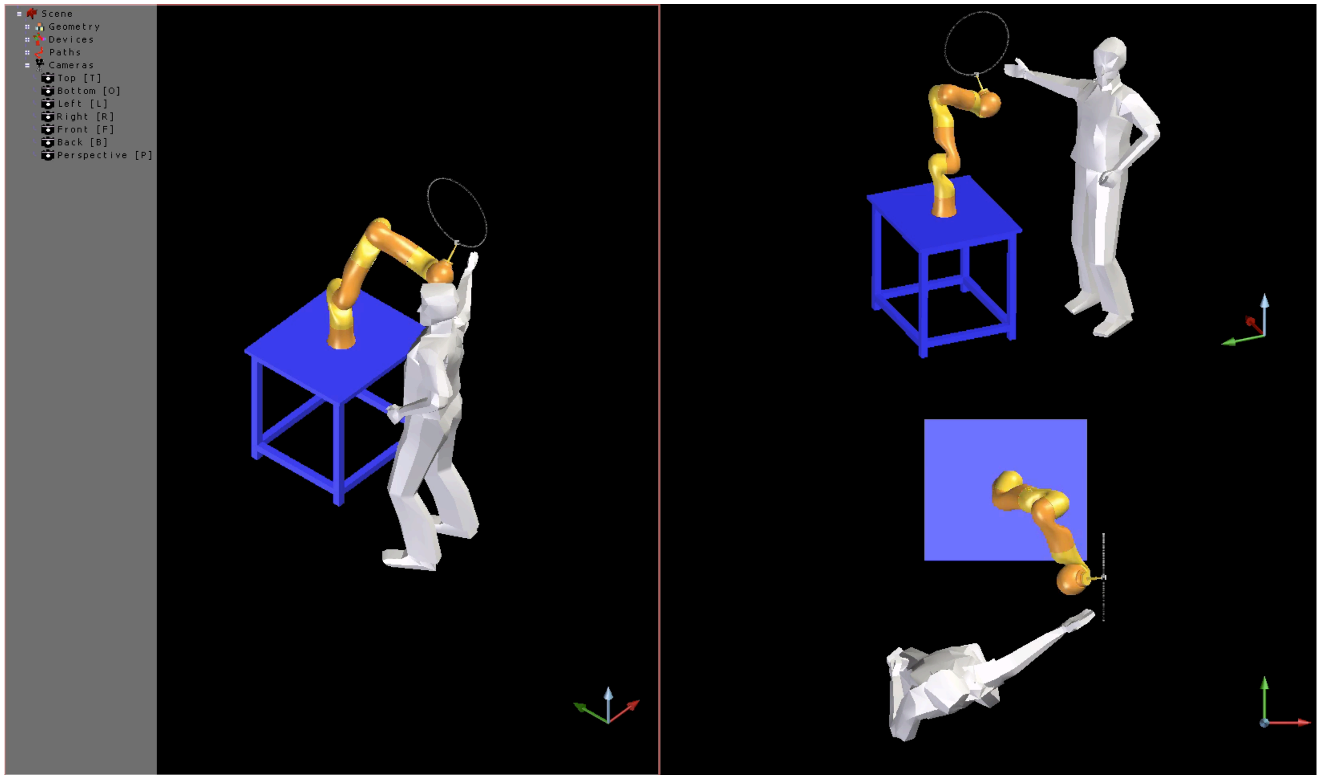
- no guarantee that singularities are globally avoided during task execution
  - despite joint velocities are kept to a minimum, this is only a local property and "avalanche" phenomena may occur
- typically lead to non-repeatable motion in the joint space
  - cyclic motions in task space do not map to cyclic motions in joint space



$$q(t) = q(0) + \int_0^t K\big(q(\tau)\big)\dot{r}(\tau)\, d\tau$$

Drift with Jacobian pseudoinverse

- a 7R KUKA LWR4 robot moves in the vicinity of a human operator
- we command a cyclic Cartesian path (only in position, $M = 3$), to be repeated several times using the pseudoinverse solution
- (unexpected) collision of a link occurs during the third cycle ...



video

2 Null-space methods

general solution of $J\dot{q} = \dot{r}$

$\dot{q}$ that don't generate any EE motion

$$\dot{q} = \underbrace{J^{\#}\dot{r}}_{} + \underbrace{\left(I - J^{\#}J\right)\dot{q}_0}_{}$$

all solutions of the associated homogeneous equation $J\dot{q} = 0$ (self-motions)

a particular solution (here, the pseudoinverse) in $\mathcal{R}(J^T)$

"orthogonal" projection of any $\dot{q}_0$ in $\mathcal{N}(J)$

$\underbrace{J\left(I - J^{\#}J\right)\dot{q}_0 = 0}$

$(I - J^{\#}J)$ projects $\dot{q}_0$ on the $\mathcal{N}(J)$

properties of projector $[I - J^{\#}J]$

- symmetric
- idempotent: $[I - J^{\#}J]^2 = [I - J^{\#}J]$
- $[I - J^{\#}J]^{\#} = [I - J^{\#}J]$
- $J^{\#}\dot{r}$ is orthogonal to $[I - J^{\#}J]\dot{q}_0$

even more in general...

$$\dot{q} = K_1\dot{r} + (I - K_2J)\dot{q}_0$$

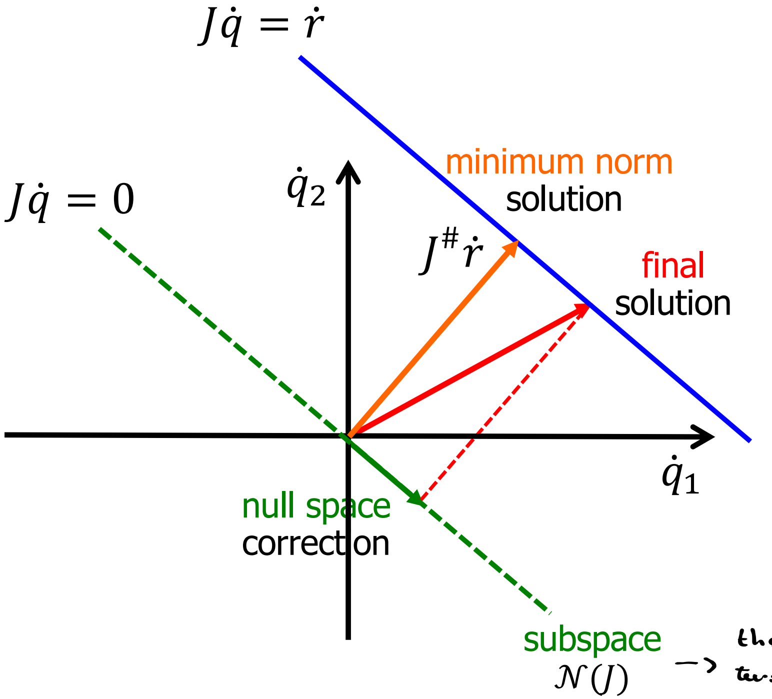
... but with less nice properties!

K_1, K_2 generalized inverses of J $(JK_iJ = J)$

how do we choose $\dot{q}_0$?

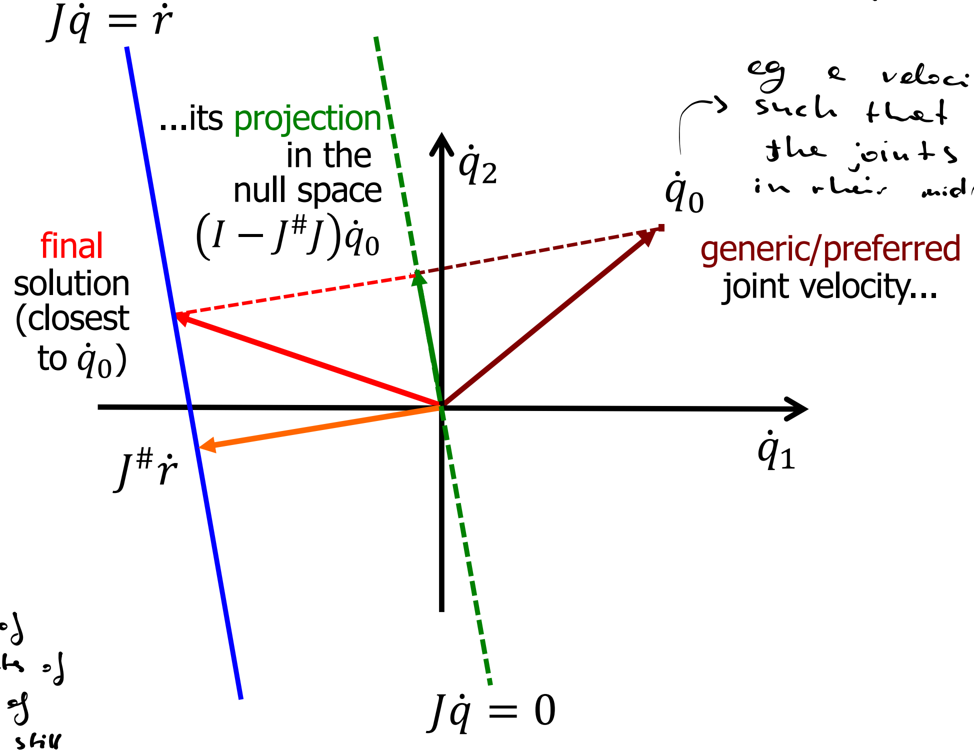
Geometric view on Jacobian null space

in the space of velocity commands



→ the sum of two elements of this set of points, lies still in the set

→ the blue line represents the task that $\dot{q}$ has to realize



→ eg a velocity such that the joints are in their midrange

a correction is added to the original pseudoinverse (minimum norm) solution
i) which is in the null space of the Jacobian
ii) and possibly satisfies additional criteria or objectives

Solution is a saddle point for the Lagrangian
minimum for the x
maximum for the λ

Linear-Quadratic Optimization
generalities

$$\min_x H(x) = \frac{1}{2}(x-x_0)^T W (x-x_0)$$

$M\times N$

$$\text{s.t.}\quad Jx = y \rightarrow \dot{r} \rightarrow \text{task}$$

$x \in \mathbb{R}^N$

$W > 0$ (symmetric)

$y \in \mathbb{R}^M$

$\mathrm{rank}(J) = \rho(J) = M$

$$L(x,\lambda) = H(x) + \lambda^T (Jx - y)$$ ← Lagrangian (with multipliers λ)

necessary conditions

$$\nabla_x L = \left(\frac{\partial L}{\partial x}\right)^T = W(x-x_0) + J^T\lambda = 0 \Rightarrow \boxed{x = x_0 - W^{-1}J^T\lambda}$$

$$\nabla_\lambda L = \left(\frac{\partial L}{\partial \lambda}\right)^T = \overbrace{Jx - y}^{\text{constraint}} = 0 \Rightarrow \boxed{Jx_0 - JW^{-1}J^T\lambda - y = 0}$$

\+ sufficient condition for a minimum

$$\nabla_x^2 L = W > 0$$

$x = x_0 + J^{\#}_W (y - Jx_0)$

$$\boxed{\lambda = \underbrace{(JW^{-1}J^T)^{-1}}_{M\times M \text{ invertible}}(Jx_0 - y)} \Rightarrow \boxed{x = x_0 + W^{-1}J^T(JW^{-1}J^T)^{-1}(y - Jx_0)}$$

only if J full rank

$$x = x_0 + \overbrace{J^{\#}_W y}^{\text{min weighted norm solution}} - J^{\#}_W J x_0 =$$

$$= J^{\#}_W y + (I - J^{\#}_W J)\, x_0 \rightarrow \text{preferred solution}$$

Linear-Quadratic Optimization

application to robot redundancy resolution

PROBLEM

$$\min_{\dot{q}} H(\dot{q}) = \frac{1}{2}(\dot{q} - \dot{q}_0)^T W (\dot{q} - \dot{q}_0)$$

$$\text{s.t.} \quad J\dot{q} = \dot{r}$$

$\dot{q}_0$ is a "privileged" joint velocity

SOLUTION

$$\boxed{\dot{q} = \dot{q}_0 + \underbrace{W^{-1}J^T(JW^{-1}J^T)^{-1}}_{J_W^{\#}}(\dot{r} - J\dot{q}_0)}$$

→ more efficient way

$$\dot{q} = \underbrace{J_W^{\#}\dot{r}}_{\text{minimum weighted norm solution (for } \dot{q}_0 = 0)} + \underbrace{\left(I - J_W^{\#}J\right)\dot{q}_0}_{\text{"projection" matrix in the null-space } \mathcal{N}(J)}$$

min 48:00

Projected Gradient (PG)

$$\dot{q} = J^{\#}\dot{r} + \left(I - J^{\#}J\right)\dot{q}_0$$

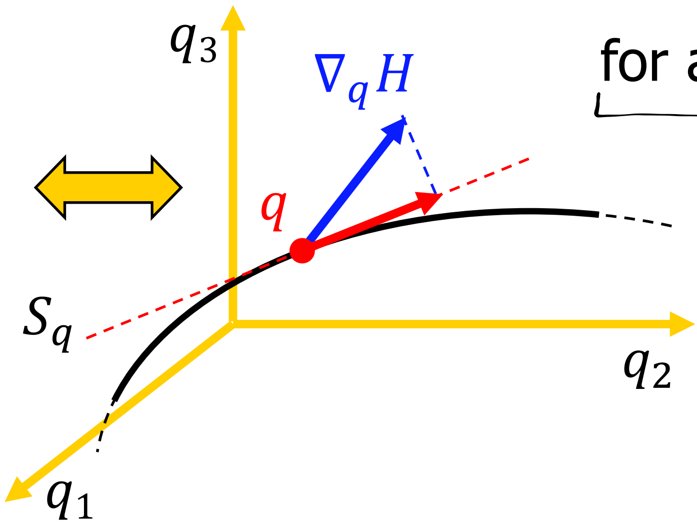
the choice $\dot{q}_0 = \nabla_q H(q)$ → differentiable objective function

realizes one step of a constrained optimization algorithm

We use this

while executing the time-varying task $r(t)$
the robot tries to increase the value of $H(q)$

projected gradient ⇔



for a fixed $\bar{r}$: $\underbrace{S_q}_{\text{eg self motion}} = \{q \in \mathbb{R}^N : f(q) = \bar{r}\}$

$$\Rightarrow \quad \dot{q} = \left(I - J^{\#}J\right)\nabla_q H$$

min 54:00

N-dimensional $\boxed{\left(I - J^{\#}J\right)\nabla_q H = 0}$

constrained optimum when the gradient is orthogonal to the curve

is a necessary condition of constrained optimality

if I hadn't stay on the curve I would use unconstrained optimization

Typical objective functions $H(q)$

- **manipulability** (maximize the "distance" from singularities)

function proportional to the volume of the hyper ellipsoid ←

$$H_{\text{man}}(q) = \sqrt{\det\left[J(q)J^T(q)\right]}$$

min. 59.00

maximising the performance in transform joint vel into EE vel

- **joint range** (minimize the "distance" from the mid points of the joint ranges)

$$q_i \in \left[q_{m,i}, q_{M,i}\right] \qquad \bar{q}_i = \frac{q_{M,i} + q_{m,i}}{2}$$

$$H_{\text{range}}(q) = \frac{1}{2N}\sum_{i=1}^{N}\left(\frac{q_i - \bar{q}_i}{q_{M,i} - q_{m,i}}\right)^2$$

$$\dot{q}_0 = -\nabla_q H(q)$$

$$\nabla_q H = \left[\frac{1}{N}\left(\frac{q_i - \bar{q}_i}{(q_{M,i} - q_{m,i})^2}\right)\right]^T \in \mathbb{R}^{N \times 1}$$

↳ Den is a weighting factor. Smaller ranges mean higher weight of displacement

- **obstacle avoidance** (maximize the minimum distance to Cartesian obstacles)

also known as "clearance"

$$H_{\text{obs}}(q) = \min_{\substack{a \in \text{robot} \\ b \in \text{obstacles}}} \|a(q) - b\|^2$$

potential difficulties due to non-differentiability (this is a max-min problem)

↳ continuous function, but its gradient is not min. 1.10.00

Singularities of planar 3R arm

the robot is redundant for a positioning task in the plane ($M = 2$)

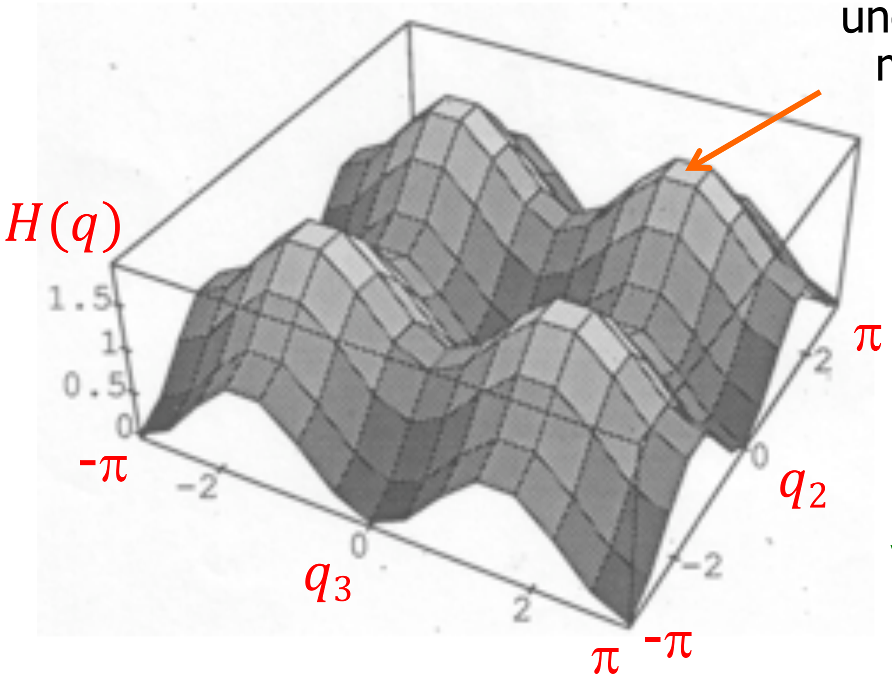
$$H(q) = \sin^2 q_2 + \sin^2 q_3$$

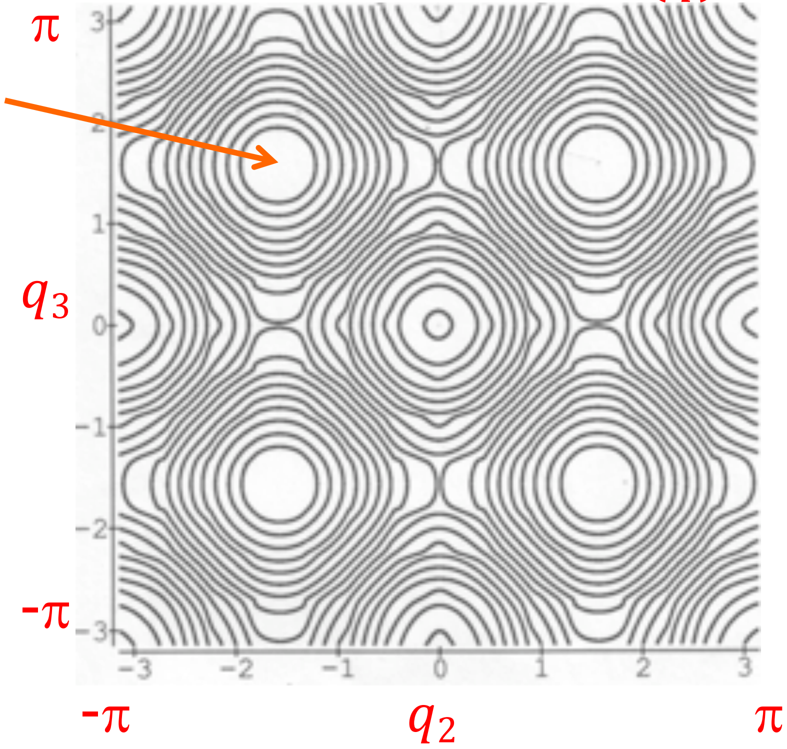
same behavior as manipulability function

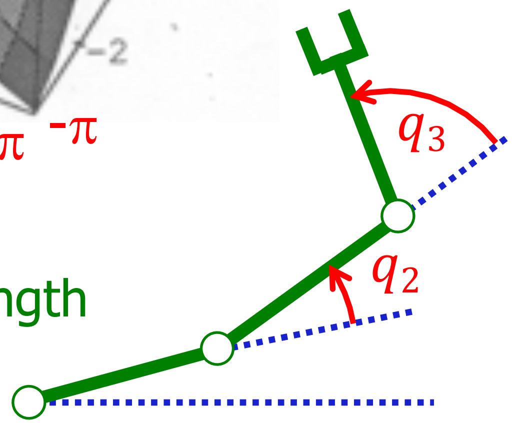
this H is **not** H_{man} but has the same minima

iso-level curves of $H(q)$

unconstrained maxima of $H(q)$







links of **equal** (unit) length

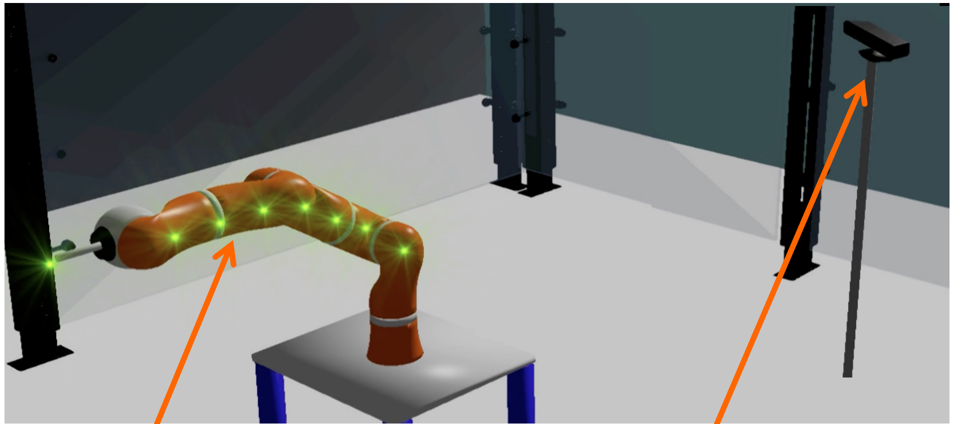
independent from q_1!

↳ in order to optimize H we don't need to move q_1

min 1.13.45

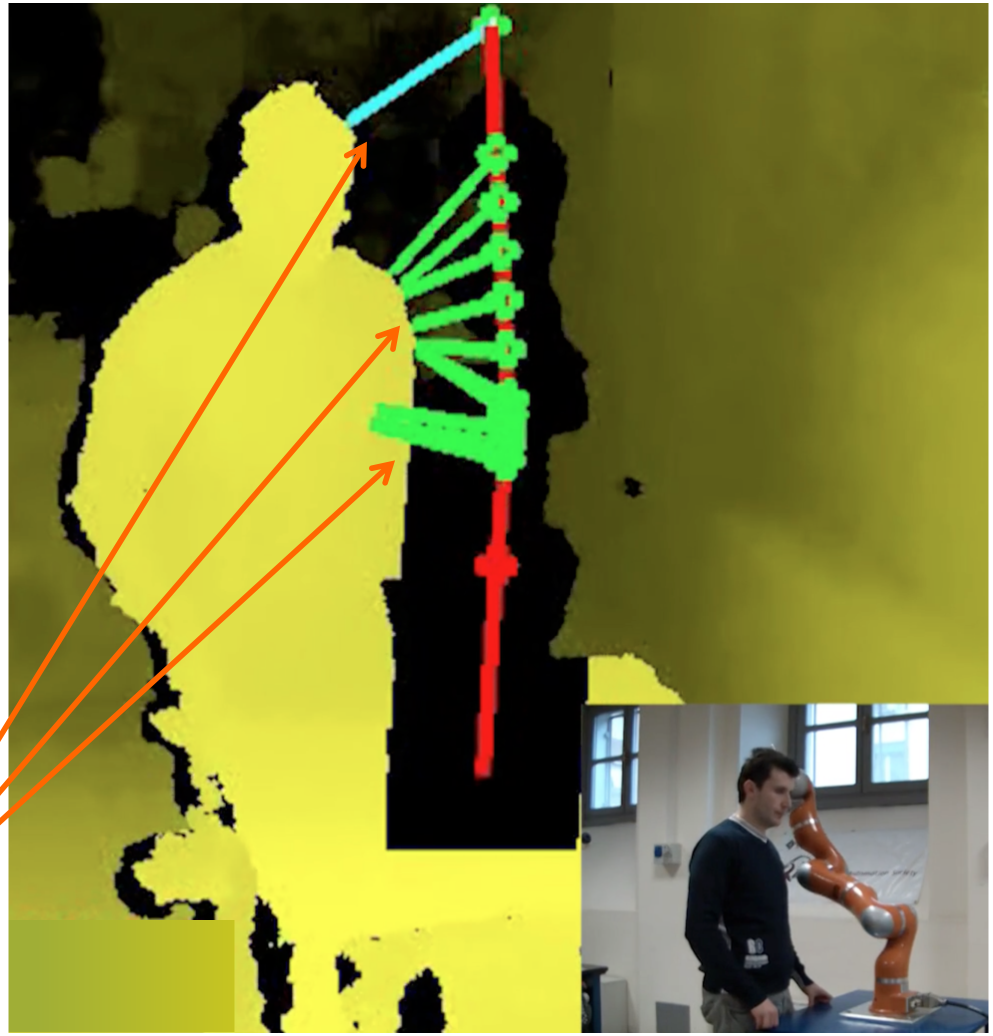
Minimum distance computation

in human-robot interaction



LWR4 robot with a finite number of control points $\boldsymbol{a}(\boldsymbol{q})$ (8, including the E-E)

a Kinect sensor monitors the workspace giving the 3D position of points $\boldsymbol{b}$ on obstacles that are fixed or moving (like humans)



distances in 3D (and then the clearance) are computed in this case as

$$\min_{\substack{a \in \{\text{control points}\} \\ b \in \text{human body}}} \|a(q) - b\|^2$$

Comments on null-space methods

- the projection matrix $(I - J^{\#}J)$ has dimension $N\times N$, but only rank $N - M$ (if J is full rank M), with some waste of information
- actual (efficient) evaluation of the solution

$$\dot{q} = J^{\#}\dot{r} + \left(I - J^{\#}J\right)\dot{q}_0 = \dot{q}_0 + J^{\#}(\dot{r} - J\dot{q}_0)$$

  but the pseudoinverse is needed anyway, and this is computationally intensive (SVD in the general case)
- in principle, the additional complexity of a redundancy resolution method should depend only on the redundancy degree $N - M$
- a constrained optimization method is available, which is known to be more efficient than the projected gradient (PG) —at least when the Jacobian has full rank …

min
1.17.00

Decomposition of joint space

- if $\rho(J(q)) = M$, there exists a decomposition of the set of joints (possibly, after a reordering)

$$q = \begin{pmatrix} q_a \\ q_b \end{pmatrix} \begin{matrix} \} \; M \\ \} \; N-M \end{matrix} \qquad \text{such that } J_a(q) = \overbrace{\frac{\partial f}{\partial q_a}}^{M \times M \;\to\; \text{task dimension}} \text{ is nonsingular}$$

- from the implicit function theorem, there exists an inverse function g

$$f(q_a, q_b) = r \quad \Rightarrow \quad q_a = g(r, q_b)$$

$$J = \frac{\partial f}{\partial q} = \begin{bmatrix} \frac{\partial f}{\partial q_a} & \frac{\partial f}{\partial q_b} \end{bmatrix} = \begin{bmatrix} \underset{\substack{M \times M \\ \text{nonsing}}}{J_a} & \underset{M \times (N-M)}{J_b} \end{bmatrix}$$

$$\text{with } \frac{\partial g}{\partial q_b} = -\left(\frac{\partial f}{\partial q_a}\right)^{-1} \frac{\partial f}{\partial q_b} = -J_a^{-1}(q) J_b(q)$$

- the $N-M$ variables q_b can be selected independently (e.g., they are used for optimizing an objective function $H(q)$, "reduced" via the use of g to a function of q_b only)
- $q_a = g(r, q_b)$ is then chosen so as to correctly execute the task

Reduced Gradient (RG)

→ this new H takes into account the effects of the gradient wrt q_b also on q_a

- $H(q) = H(q_a, q_b) = H(g(r, q_b), q_b) = H'(q_b)$, with r at current value
- the Reduced Gradient (w.r.t. q_b only, but still keeping the effects of this choice into account) is

$$\nabla_{q_b} H' = [\overbrace{-(J_a^{-1} J_b)^T}^{\nabla_{q_b} g} \quad I_{N-M}] \nabla_q H$$

$(\neq \nabla_{q_b} H$ only!!)

$$\boxed{\nabla_{q_b} H' = 0}$$

eg with 6dof robt and 5dim task, to optimize the secondary task, I'll have 1 equation

is a "compact"
(i.e., $N-M$ dimensional)
necessary condition
of constrained optimality

- algorithm

$$\dot{q}_b = \nabla_{q_b} H'$$

step in the gradient direction of the reduced $(N-M)$-dim space

$$J_a \dot{q}_a + J_b \dot{q}_b = \dot{r}$$

satisfaction of the M-dim task constraints

$$\dot{q}_a = J_a^{-1}(\dot{r} - J_b \dot{q}_b)$$

Comparison between PG and RG

- Projected Gradient (PG)

$$\dot{q} = J^{\#}\dot{r} + \left(I - J^{\#}J\right)\nabla_q H$$

- Reduced Gradient (RG)

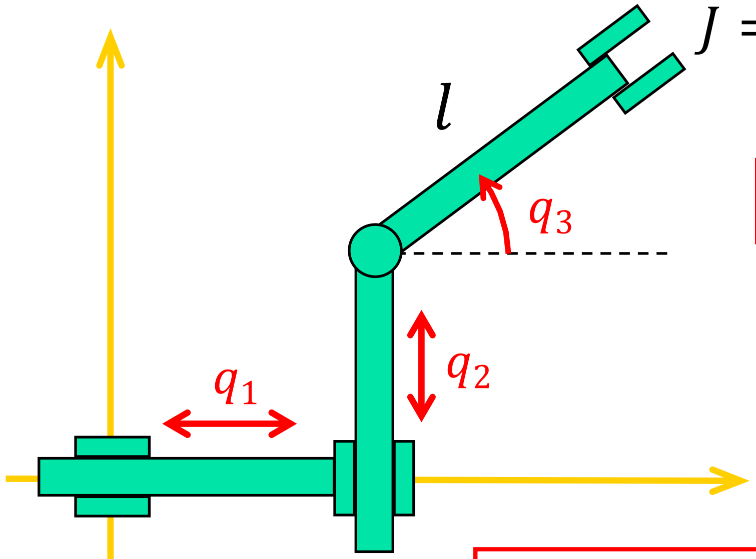
$$\dot{q} = \begin{pmatrix} \dot{q}_a \\ \dot{q}_b \end{pmatrix} = \begin{pmatrix} J_a^{-1} \\ 0 \end{pmatrix}\dot{r} + \begin{pmatrix} -J_a^{-1}J_b \\ I \end{pmatrix}\left(-(J_a^{-1}J_b)^T \quad I\right)\nabla_q H$$

- RG is analytically simpler and numerically faster than PG, but requires the search for a non-singular minor (J_a) of the robot Jacobian
- if $r = \text{cost}$ & $N - M = 1$ $\Rightarrow$ same (unique) direction for $\dot{q}$, but RG has automatically a larger optimization step size
- else $\Rightarrow$ RG and PG methods provide always different evolutions

min 1.27.00

Analytic comparison

PPR robot



$$J = \begin{pmatrix} 1 & 0 & -ls_3 \\ 0 & 1 & lc_3 \end{pmatrix} = (J_a \mid J_b) \quad q_a = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} \quad q_b = q_3$$

RG: $\dot q = \begin{pmatrix} J_a^{-1} \\ 0 \end{pmatrix}\dot r + \begin{pmatrix} -J_a^{-1}J_b \\ I \end{pmatrix}\left(-(J_a^{-1}J_b)^T \quad I\right)\nabla_q H$

$$\dot q = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}\dot r + \underbrace{\begin{pmatrix} ls_3 \\ -lc_3 \\ 1 \end{pmatrix}}_{\text{column vector that spans the nullspace}}(ls_3 \quad -lc_3 \quad 1)\nabla_q H$$

→ you move in that direction by reducing the gradient in that direction

PG: $\dot q = J^{\#}\dot r + (I - J^{\#}J)\nabla_q H$

this outer product gives this matrix

$$J^{\#} = \frac{1}{1+l^2}\begin{pmatrix} 1 + l^2c_3^2 & l^2 s_3 c_3 \\ l^2 s_3 c_3 & 1 + l^2 s_3^2 \\ -ls_3 & lc_3 \end{pmatrix} \qquad I - J^{\#}J = \frac{1}{1+l^2}\begin{pmatrix} l^2 s_3^2 & l^2 s_3 c_3 & ls_3 \\ l^2 s_3 c_3 & l^2 c_3^2 & -lc_3 \\ ls_3 & -lc_3 & 1 \end{pmatrix}$$

RG optimizes faster ← scaling factor is not present in the RG method ← always < 1!!

Joint range limits

$$q = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \theta = T\theta$$

absolute ⇔ relative
coordinates

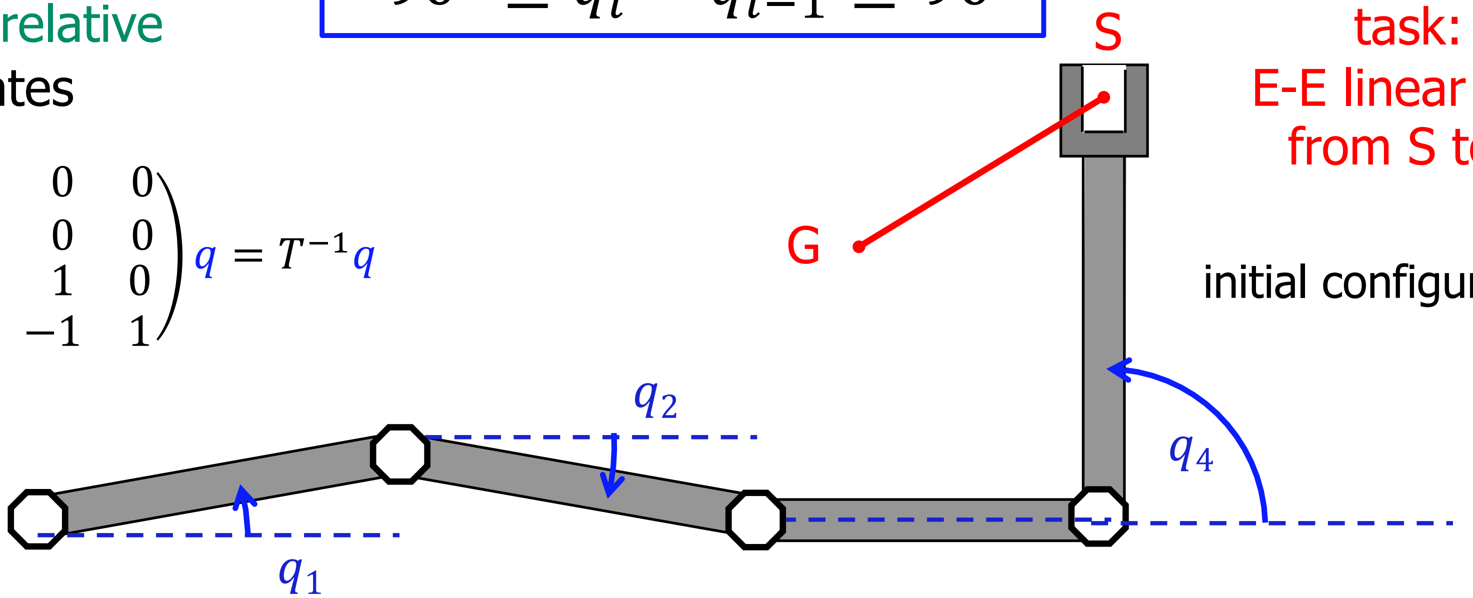
$$\theta = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix} q = T^{-1} q$$

$$\boxed{-90° \le \theta_i \le 90°}$$

$$\Updownarrow$$

$$\boxed{-90° \le q_i - q_{i-1} \le 90°}$$

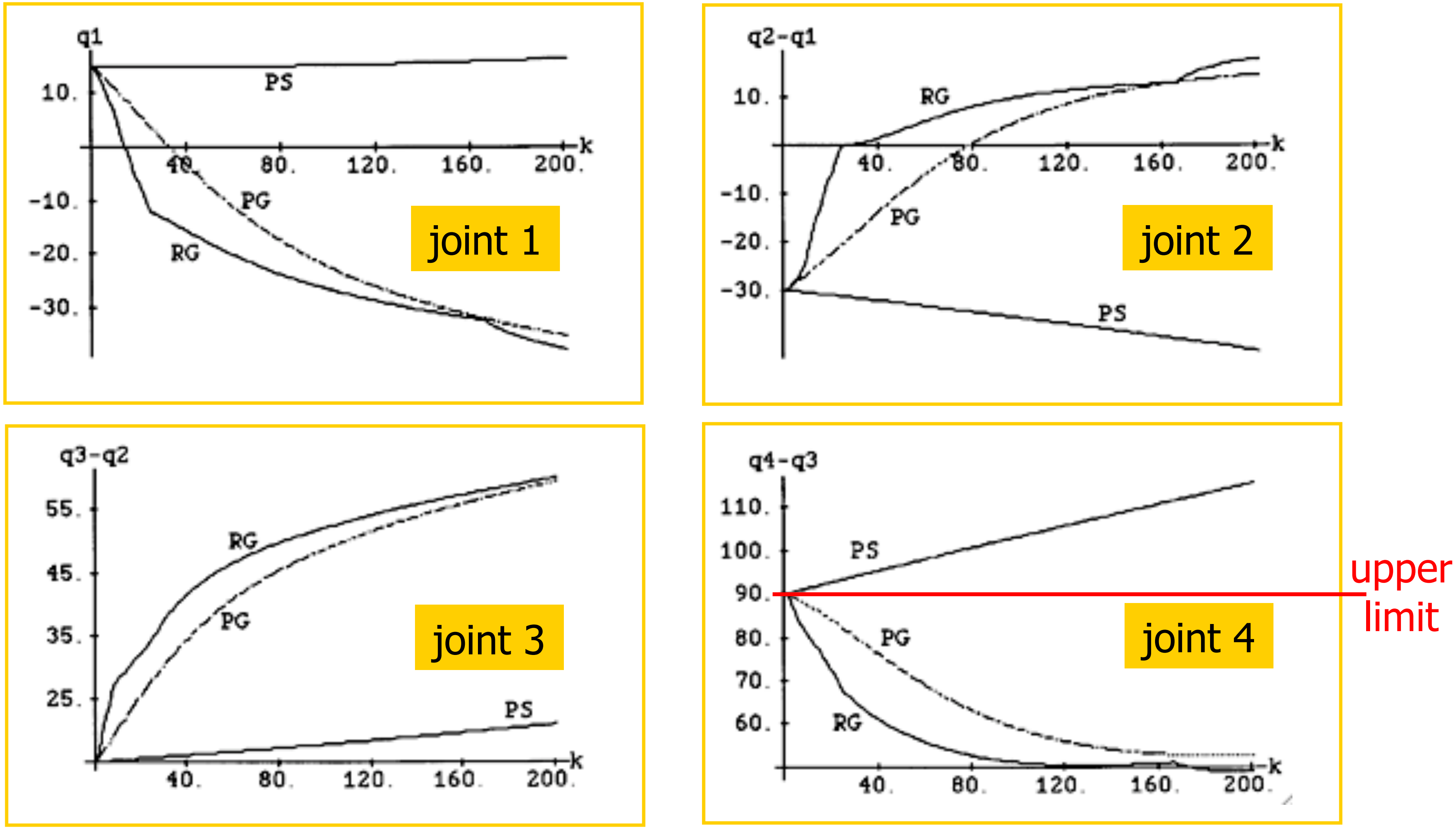
task:
E-E linear path
from S to G



numerical comparison among pseudoinverse (PS), projected gradient (PG), and reduced gradient (RG) methods

Numerical results

minimizing distance from mid joint range



steps of numerical simulation

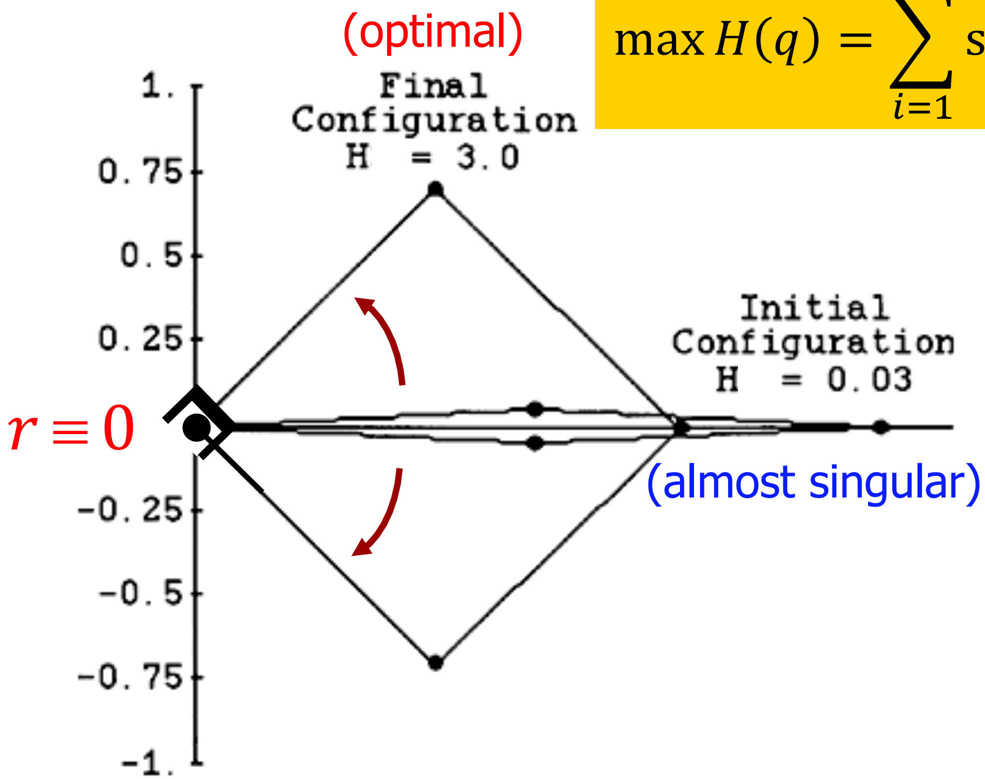
with Jacobian based methods (don't move ←

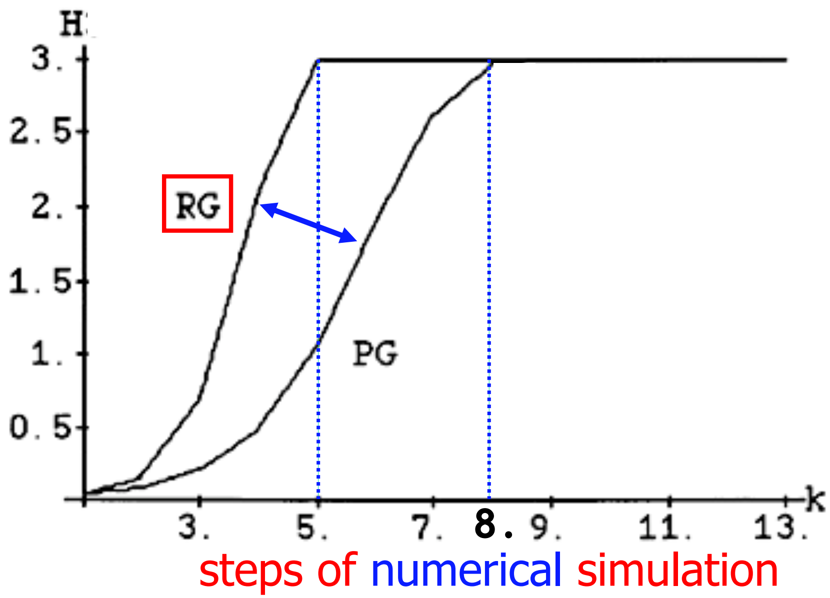
Numerical results

self-motion for escaping singularities

$$\max H(q) = \sum_{i=1}^{3} \sin^2(q_{i+1} - q_i)$$

this function is NOT the manipulability index, but has the same minima (= 0)





RG is faster than PG
(keeping the same accuracy on r)