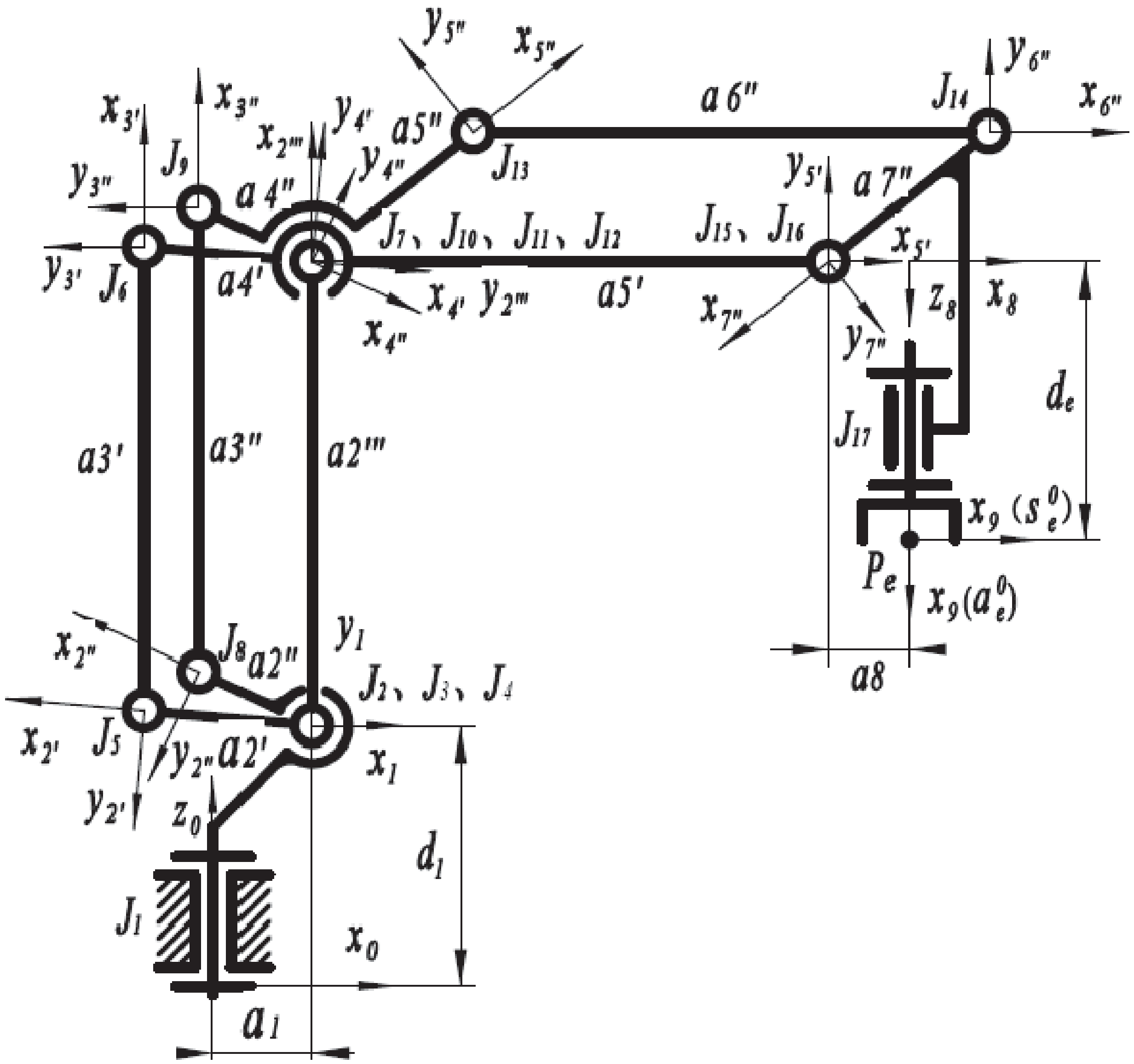
UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

INVERSE DIFFERENTIAL KINEMATICS

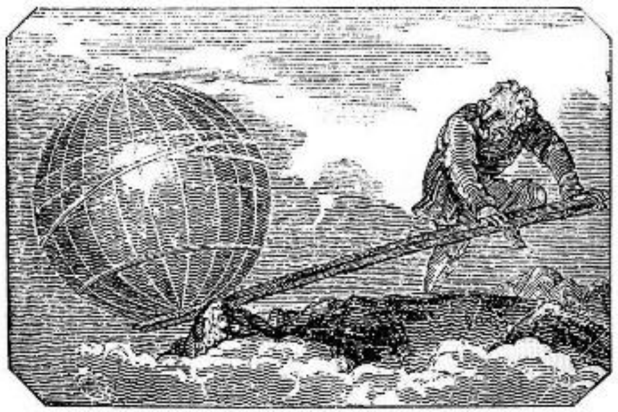


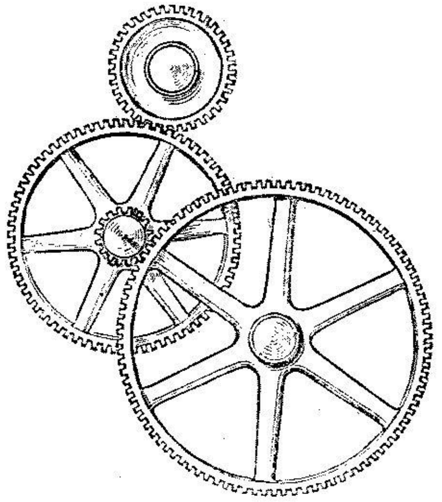
Statics for Robotics

The **principle of virtual work** states that in equilibrium the **virtual work** of the forces applied to a system is zero. Newton's laws state that at equilibrium the applied forces are equal and opposite to the reaction, or constraint forces.
This means the **virtual work** of the constraint forces must be zero as well.

principle of virtual work at equilibrium → ***(dW=dF dx)=0***

The **principle of virtual work** had always been used in some form since antiquity in the study of statics





Statics: Geometric Jacobian (Generalizing of *n-dof*)

We use a new notation

$J_P = J_L$

$J_O = J_A$

$$\begin{pmatrix} v_E \\ \omega_E \end{pmatrix} = \begin{pmatrix} J_L(q) \\ J_A(q) \end{pmatrix} \dot{q}$$

$$\dot{\boldsymbol{p}}_e = \boldsymbol{J}_P(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2}$$

$$\boldsymbol{\omega}_e = \boldsymbol{J}_O(\boldsymbol{q})\dot{\boldsymbol{q}}. \tag{3}$$

In (2) $\boldsymbol{J}_P$ is the $(3 \times n)$ matrix relating the contribution of the joint velocities $\dot{\boldsymbol{q}}$ to the end-effector *linear* velocity $\dot{\boldsymbol{p}}_e$, while in (3) $\boldsymbol{J}_O$ is the $(3 \times n)$ matrix relating the contribution of the joint velocities $\dot{\boldsymbol{q}}$ to the end-effector *angular* velocity $\boldsymbol{\omega}_e$.

Statics: Geometric Jacobian (Generalizing of *n-dof)*

In compact form,

$$\boldsymbol{v}_e = \begin{bmatrix} \dot{\boldsymbol{p}}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

which represents the manipulator *differential kinematics equation*. The $(6 \times n)$ matrix $\boldsymbol{J}$ is the manipulator *geometric Jacobian*

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_P \\ \boldsymbol{J}_O \end{bmatrix},$$

which in general is a function of the joint variables.

Statics

It determines the relationship between the generalized forces applied to the end-effector and the generalized forces applied to the joints

Let $\boldsymbol{\tau}$ **denote the (*n* × 1) vector of infinitesimal joint torques** and $\boldsymbol{\gamma}$ **the (*m* × 1) vector of infinitesimal end effector forces and torques** where ***m*** is the dimension of the operational space of interest.

Let´s apply the principle of virtual work (dW=dF dx)=0

As for the joint torques: $$dW_{\tau} = \boldsymbol{\tau}^T d\boldsymbol{q}.$$

As for the end-effector forces: $$dW_{\gamma} = \boldsymbol{f}_e^T d\boldsymbol{p}_e + \boldsymbol{\mu}_e^T \boldsymbol{\omega}_e dt,$$

$$\boldsymbol{v}_e = \begin{bmatrix} \dot{\boldsymbol{p}}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \qquad \boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_P \\ \boldsymbol{J}_O \end{bmatrix},$$

$$\begin{aligned} dW_{\gamma} &= \boldsymbol{f}_e^T \boldsymbol{J}_P(\boldsymbol{q})d\boldsymbol{q} + \boldsymbol{\mu}_e^T \boldsymbol{J}_O(\boldsymbol{q})d\boldsymbol{q} \\ &= \boldsymbol{\gamma}_e^T \boldsymbol{J}(\boldsymbol{q})d\boldsymbol{q} \end{aligned}$$

$$\boldsymbol{\gamma}_e = [\, \boldsymbol{f}_e^T \quad \boldsymbol{\mu}_e^T \,]^T$$

Statics

$$\delta W_\tau = \boldsymbol{\tau}^T \delta \boldsymbol{q}$$
$$\delta W_\gamma = \boldsymbol{\gamma}_e^T \boldsymbol{J}(\boldsymbol{q})\delta \boldsymbol{q},$$

According to the principle of virtual work, the manipulator is at *static equilibrium* if and only if

$$\delta W_\tau = \delta W_\gamma \qquad \forall \delta \boldsymbol{q},$$

$$\boldsymbol{\tau}^T \delta \boldsymbol{q} \;=\; \boldsymbol{\gamma}_e^T \boldsymbol{J}(\boldsymbol{q})\delta \boldsymbol{q},$$

$$\boxed{\boldsymbol{\tau} = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{\gamma}_e}$$

the relationship between the (m) end effector forces/torques and the (n) joint torques is established by the transpose of the manipulator geometric Jacobian.

Primer on linear algebra

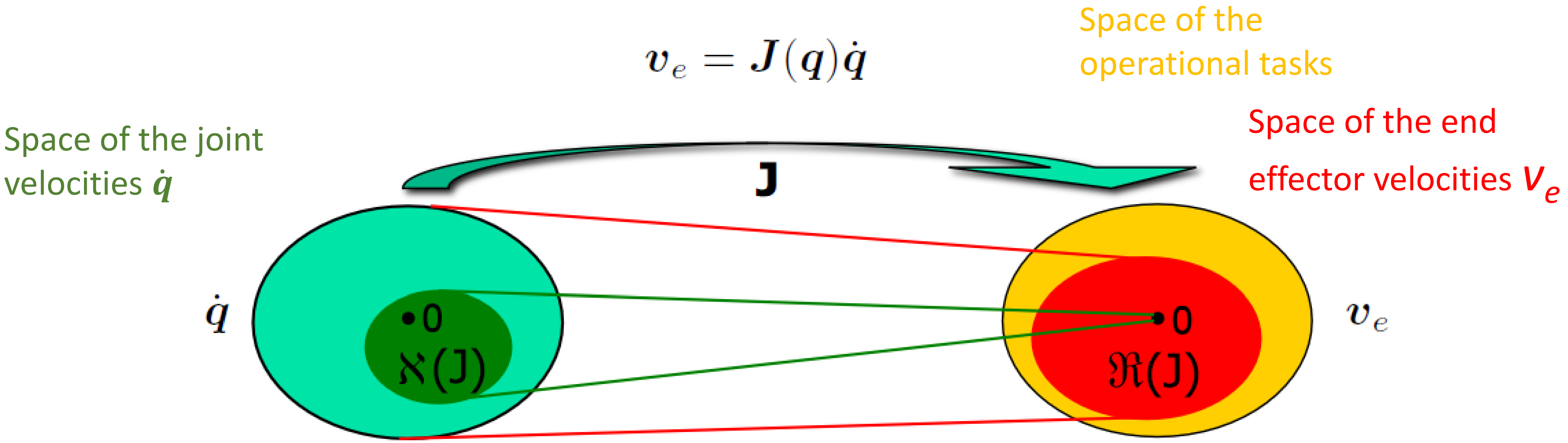
given a matrix J: $m \times n$ (m rows, n columns)

- rank $\rho(J)$ = max # of rows or columns that are linearly independent
  - $\rho(J) \leq \min(m,n)$ (if equality holds, J has "full rank")
  - if m = n and J has full rank, J is "non singular" and the inverse J^{-1} exists
  - $\rho(J)$ = dimension of the largest non singular square submatrix of J
- range $\Re(J)$ = vector subspace generated by all possible linear combinations of the columns of J ← also called "image" of J

$$\Re(J)=\{v \in R^m : \exists\, \xi \in R^n,\ v = J\,\xi\}$$

  - $\dim(\Re(J)) = \rho(J)$
- kernel $\aleph(J)$ = vector subspace of all vectors $\xi \in R^n$ such that $J \cdot \xi = 0$ ← also called "null space" of J
  - $\dim(\aleph(J)) = n - \rho(J)$
- $\Re(J) + \aleph(J^T) = R^m$ e $\Re(J^T) + \aleph(J) = R^n$
  - sum of vector subspaces $V_1 + V_2$ = vector space where any element v can be written as $v = v_1 + v_2$, with $v_1 \in V_1$, $v_2 \in V_2$

Jacobian: decomposition of subspaces
Kinematics



Mobility analysis

- $\rho(J) = \rho(J(q))$, $\Re(J) = \Re(J(q))$, $\aleph(J^T) = \aleph(J^T(q))$ are locally defined, i.e., they depend on the current configuration q
- $\Re(J(q))$ = subspace of all "generalized" velocities (with linear and/or angular components) that can be instantaneously realized by the robot end-effector when varying the joint velocities at the configuration q
- if J(q) has max rank (typically = m) in the configuration q, the robot end-effector can be moved in any direction of the task space R^m
- if $\rho(J(q)) < m$, there exist directions in R^m along which the robot end-effector cannot move (instantaneously!)
  - these directions lie in $\aleph(J^T(q))$, namely the complement of $\Re(J(q))$ to the task space R^m, which is of dimension $m - \rho(J(q))$
- when $\aleph(J(q)) \neq \{0\}$, there exist non-zero joint velocities that produce zero end-effector velocity ("self motions")
  - this always happens for m<n, i.e., when the robot is redundant for the task

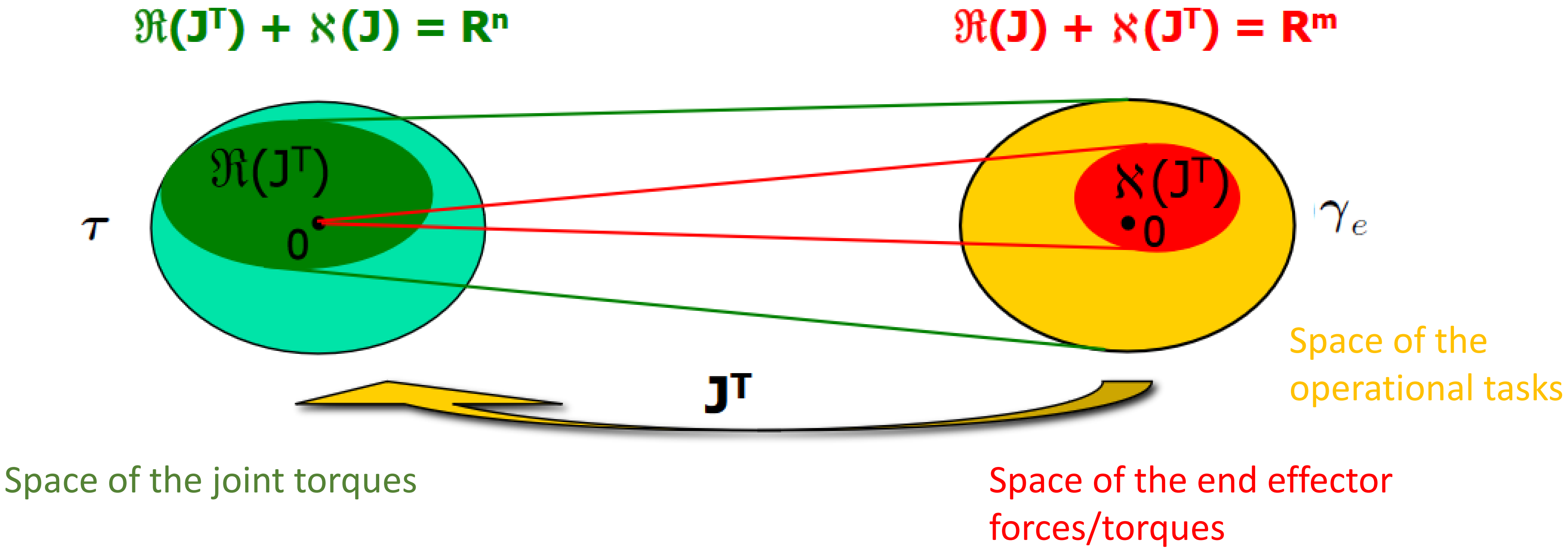
Range Vs Null Space

In fact, the effect of $\dot{\boldsymbol{q}}_0$ is to generate *internal motions* of the structure that do not not change the end-effector position but may allow, for instance, manipulator reconfiguration into more dexterous postures for execution of a given task.

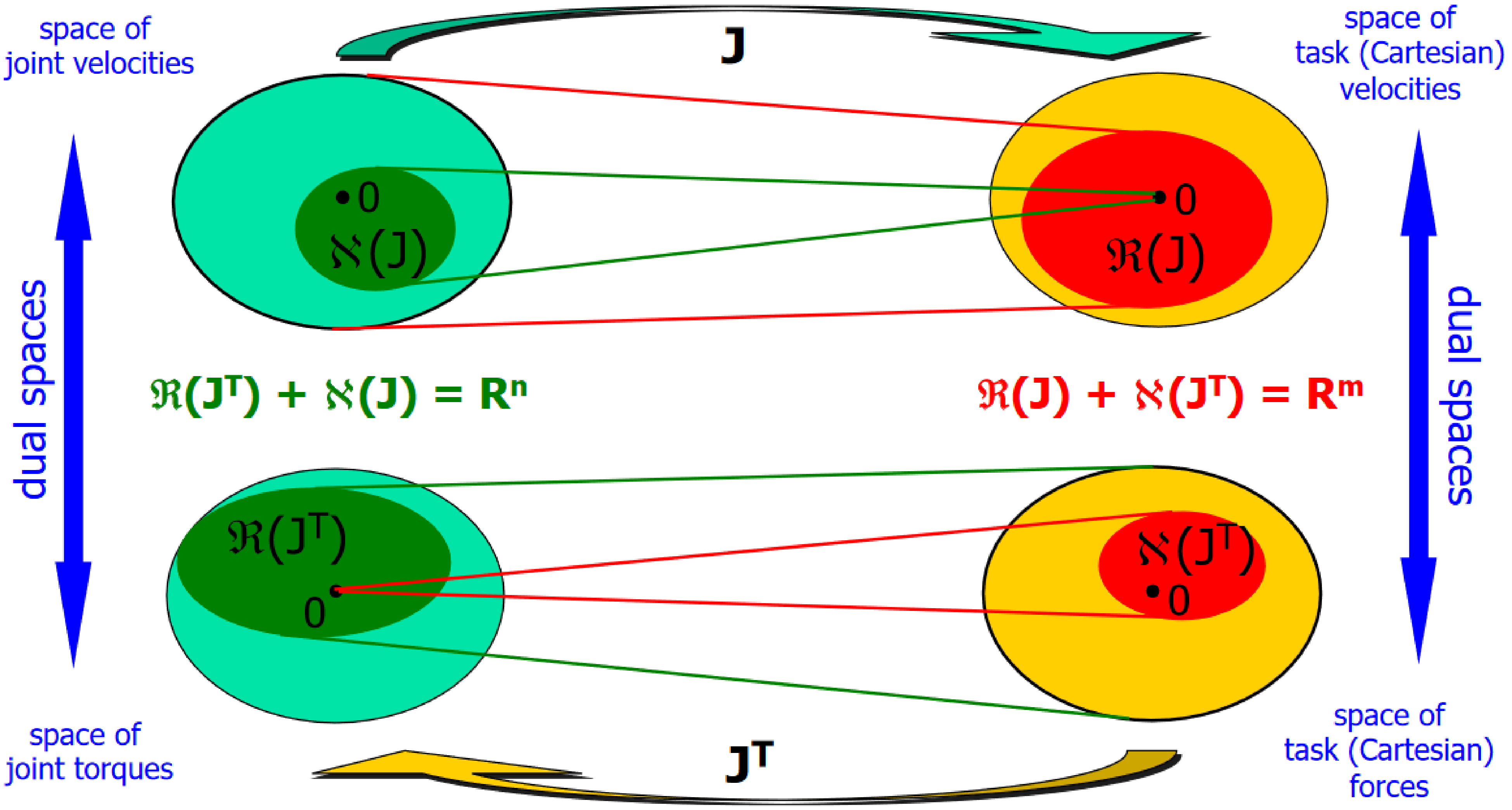
Jacobian: decomposition of subspaces

Statics

$$\tau = J^T(q)\gamma_e$$



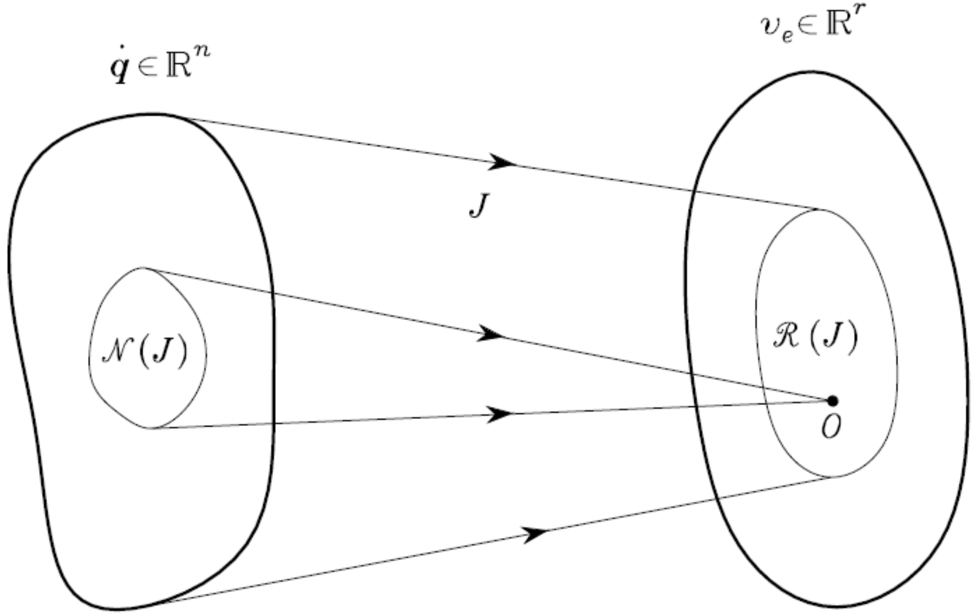
Kinetostatic Duality



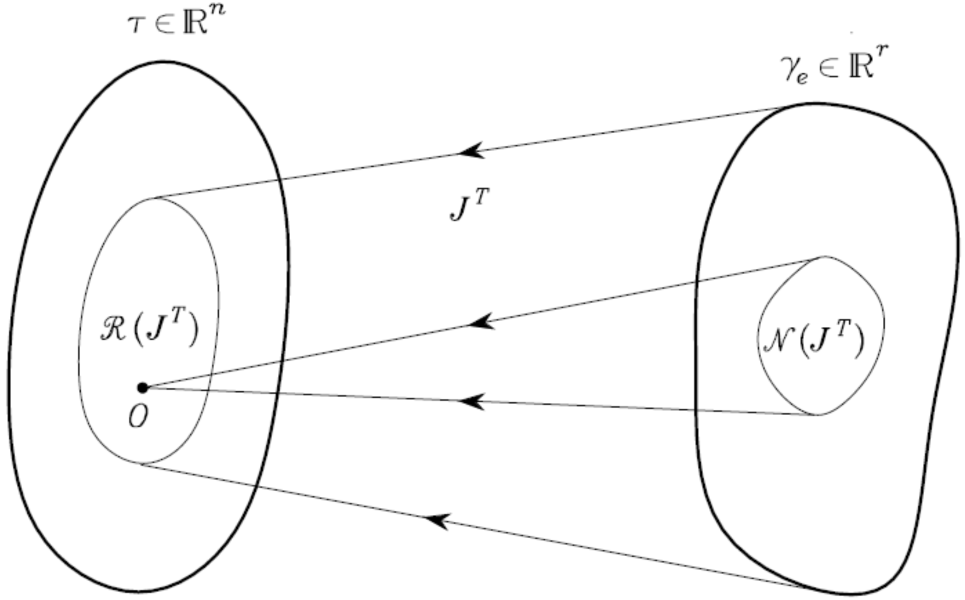
(in a given configuration q)

Kineto-Statics Duality (another notation)

$$\boldsymbol{v}_e = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$$



$$\boldsymbol{\tau} = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{\gamma}_e$$



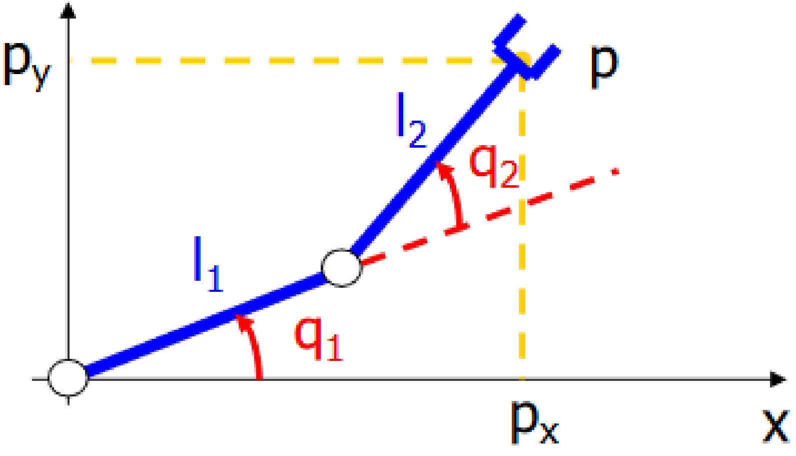
The Null Space N(J) represents in this case those solutions of joint kinematics which do not produce any motion at the end effector.

The Null Space N(J^T) represents in this case those solutions of end effector forces which dot not produce any torques at the joints.

Kinematic singularities

- **configurations where the Jacobian loses rank**
  $\Leftrightarrow$ loss of instantaneous mobility of the robot end-effector
- for $m = n$, they correspond to Cartesian poses at which the number of solutions of the inverse kinematics problem differs from the "generic" case
- "in" a singular configuration, we cannot find a joint velocity that realizes a desired end-effector velocity in an arbitrary direction of the task space
- "close" to a singularity, large joint velocities may be needed to realize some (even small) velocity of the end-effector
- finding and analyzing in advance all singularities of a robot helps in avoiding them during trajectory planning and motion control
  - when $m = n$: find the configurations q such that $\det J(q) = 0$
  - when $m < n$: find the configurations q such that all $m \times m$ minors of J are singular (or, equivalently, such that $\det [J(q)\, J^T(q)] = 0$)
- finding all singular configurations of a robot with a large number of joints, or the actual "distance" from a singularity, is a hard computational task

Singularities of planar 2R arm



direct kinematics

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

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

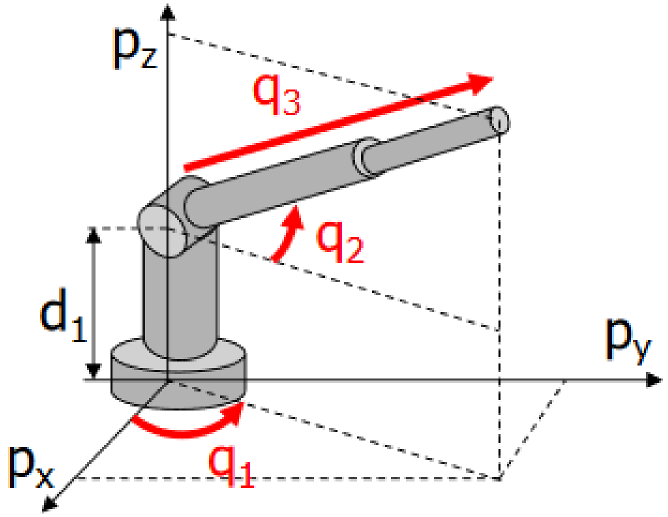
analytical Jacobian

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

$$\det J(q) = l_1 l_2 s_2$$

- singularities: arm is stretched ($q_2 = 0$) or folded ($q_2 = \pi$)
- singular configurations correspond here to Cartesian points on the boundary of the workspace
- in many cases, these singularities separate regions in the joint space with distinct inverse kinematic solutions (e.g., "elbow up" or "down")

Singularities of polar (RRP) arm



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

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

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

- singularities
  - E-E is along the z axis ($q_2 = \pm\pi/2$): simple singularity $\Rightarrow$ rank J = 2
  - third link is fully retracted ($q_3 = 0$): double singularity $\Rightarrow$ rank J drops to 1
- all singular configurations correspond here to Cartesian points internal to the workspace (supposing no limits for the prismatic joint)

Kinematic Singularities

To find the singularities of a manipulator is of great interest for the following reasons:

a) Singularities represent configurations at which mobility of the structure is reduced, i.e., it is not possible to impose an arbitrary motion to the end-effector.

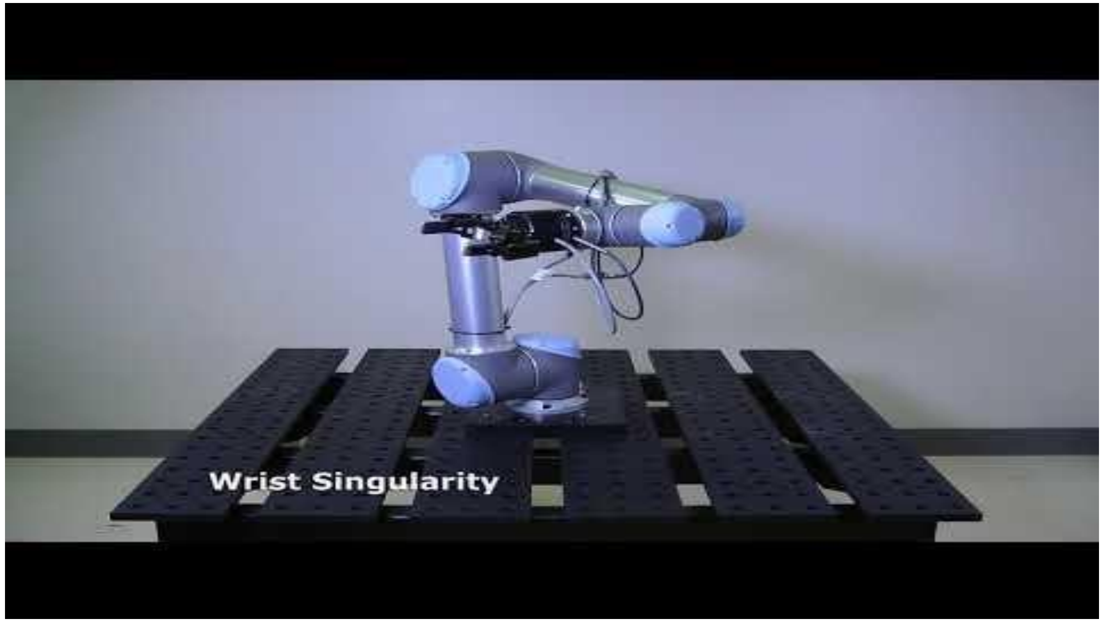
b) When the structure is at a singularity, infinite solutions to the inverse kinematics problem may exist.

c) In the neighbourhood of a singularity, small velocities in the operational space may cause large velocities in the joint space.

Redundant Manipulators

When $\boldsymbol{v}_e$ and Jacobian $\boldsymbol{J}$ are given (for a given configuration $\boldsymbol{q}$), it is desired to find the solutions $\dot{\boldsymbol{q}}$ that satisfy the linear equation $v_e = J(q)\dot{q}$ and *minimize* the quadratic cost functional of joint velocities.

Minimization of the joint velocity is required for the singular position where the robots assume high speed at the end effector for low joint velocity.



Inversion of differential kinematics

- find the joint velocity vector that realizes a desired end-effector "generalized" velocity (linear and angular)

generalized velocity → $v = J(q)\,\dot{q}$ —(J square and non-singular)→ $\dot{q} = J^{-1}(q)\,v$

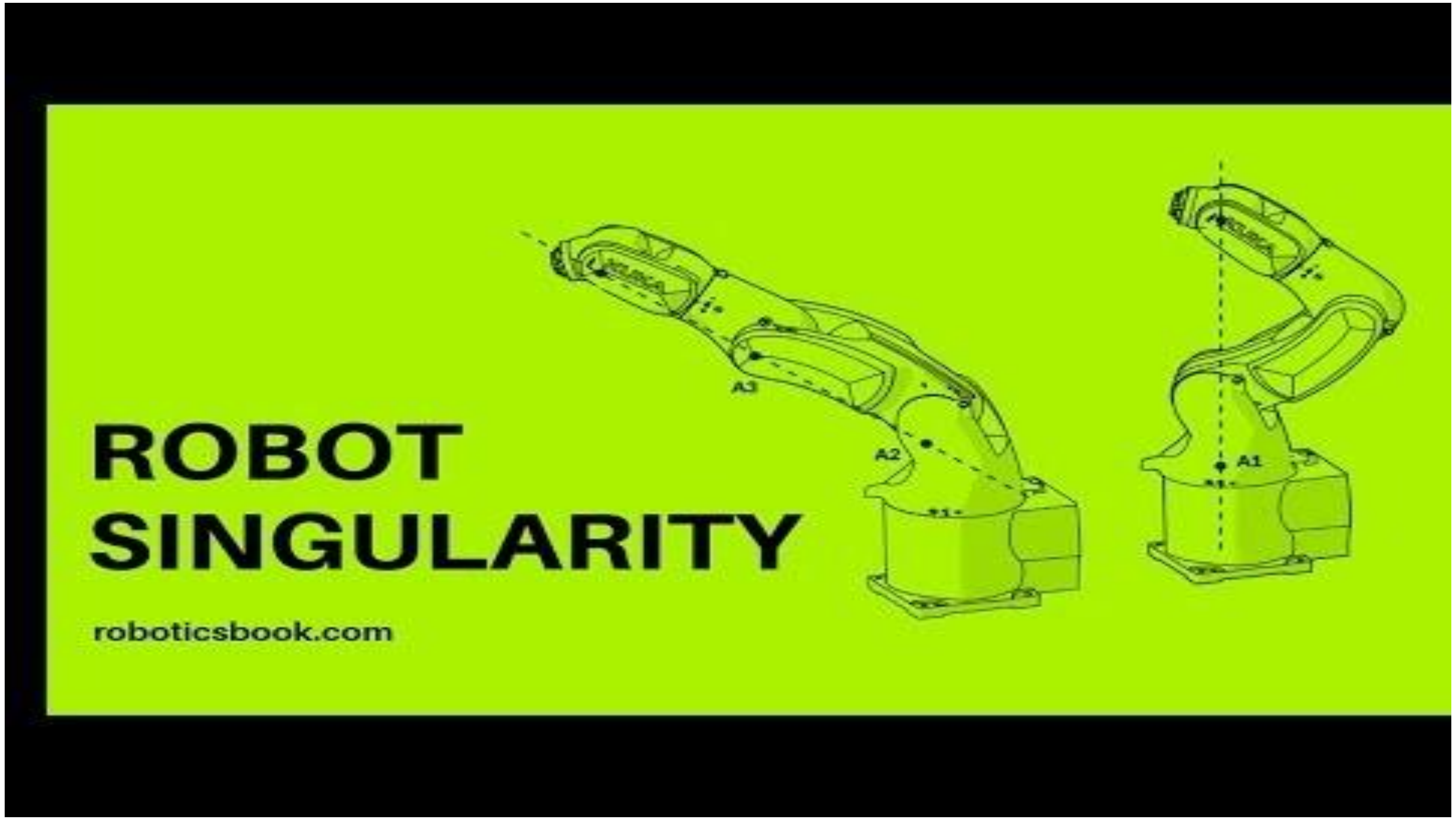
- problems
  - near a singularity of the Jacobian matrix (high $\dot{q}$)
  - for redundant robots (no standard "inverse" of a rectangular matrix)

in these cases, "more robust" inversion methods are needed

Singularity Decoupling
(1) Anthropomorphic Arm

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

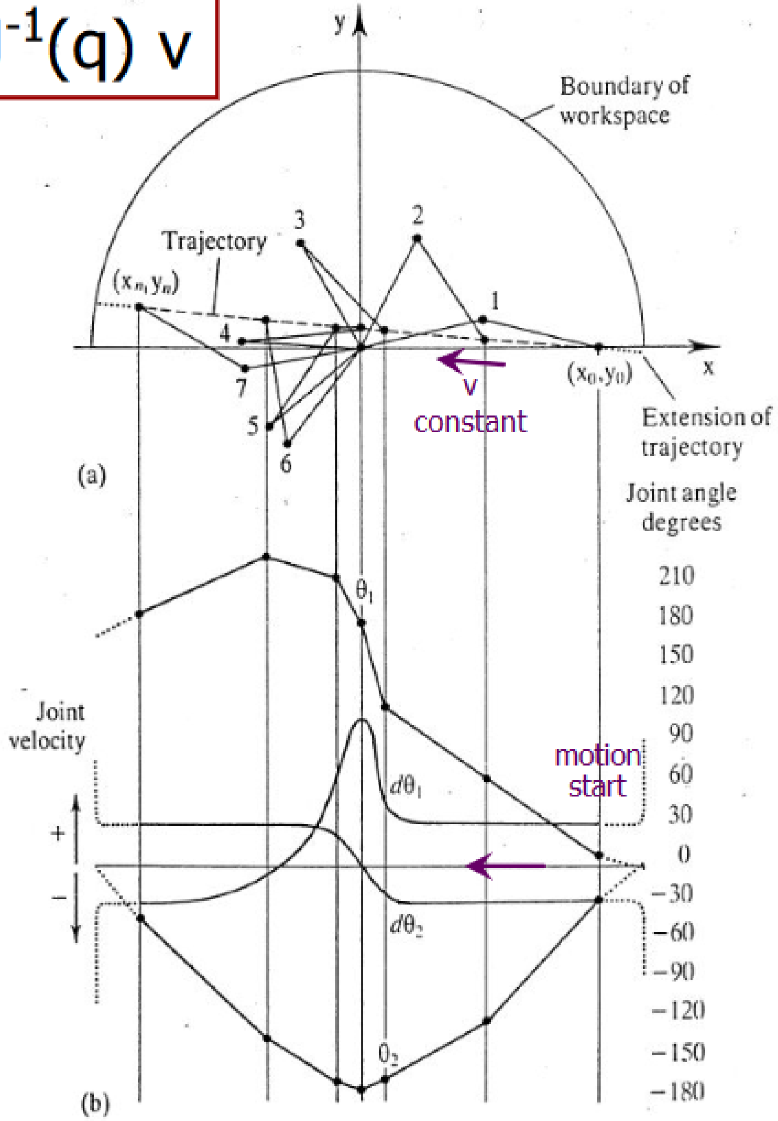
https://www.youtube.com/watch?v=BJnZvwAE0PY



UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Behavior near a singularity

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



- problems arise only when commanding joint motion by inversion of a given Cartesian motion task
- here, a linear Cartesian trajectory for a planar 2R robot
- there is a sudden increase of the displacement/velocity of the first joint near $\theta_2 = -\pi$ (end-effector close to the origin), despite the required Cartesian displacement is small

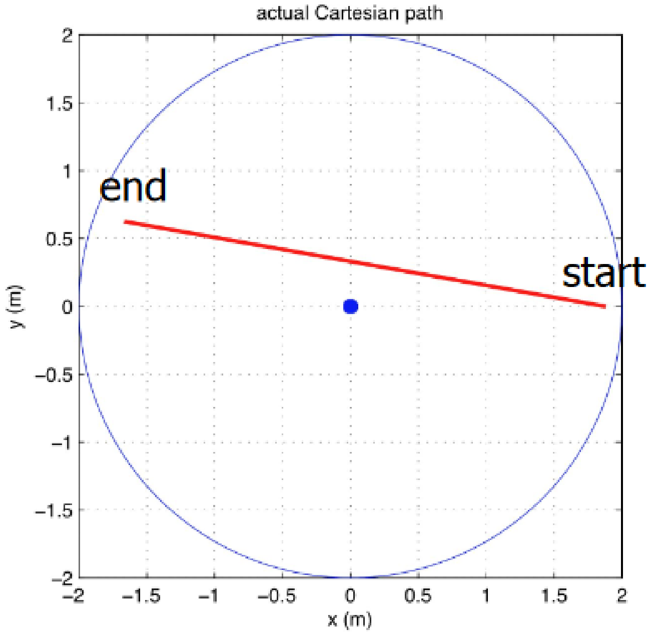
UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

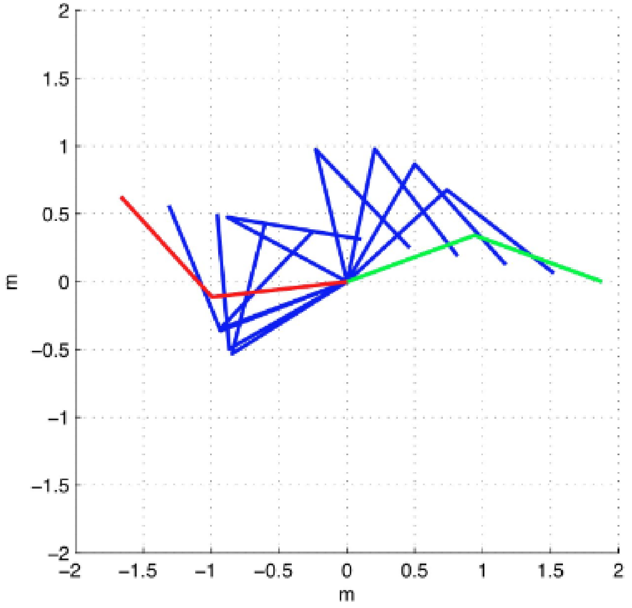
Simulation results

planar 2R robot in straight line Cartesian motion

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

regular case





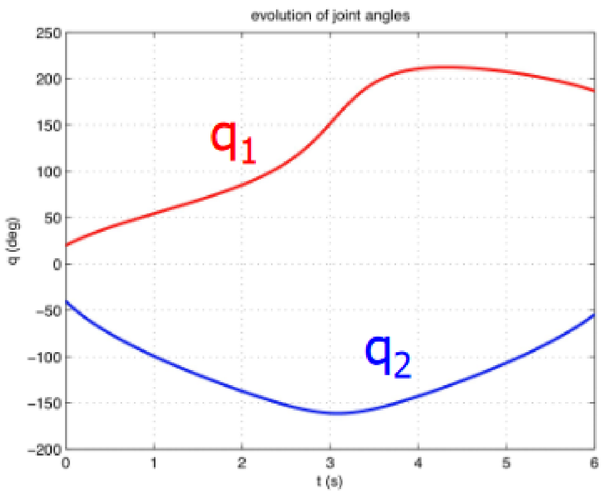
a line from right to left, at α=170° angle with x-axis,
executed at constant speed v=0.6 m/s for T=6 s

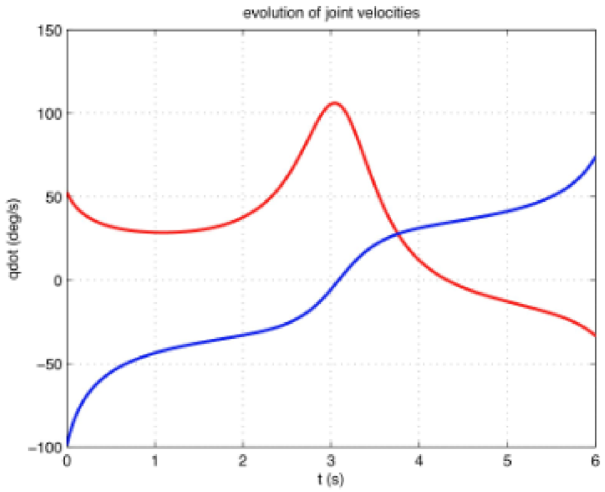
UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Simulation results

planar 2R robot in straight line Cartesian motion

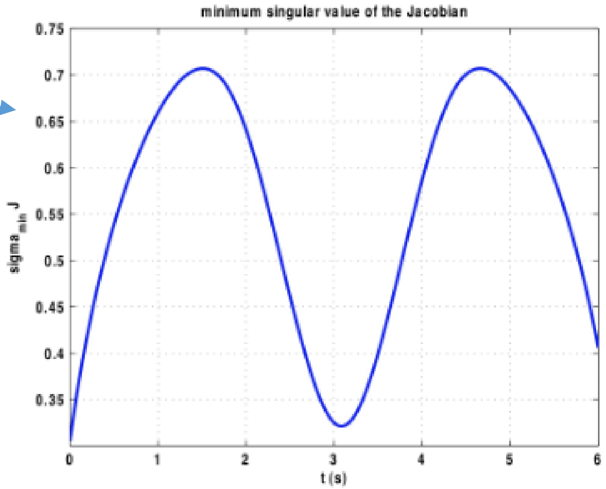
path at $\alpha = 170°$

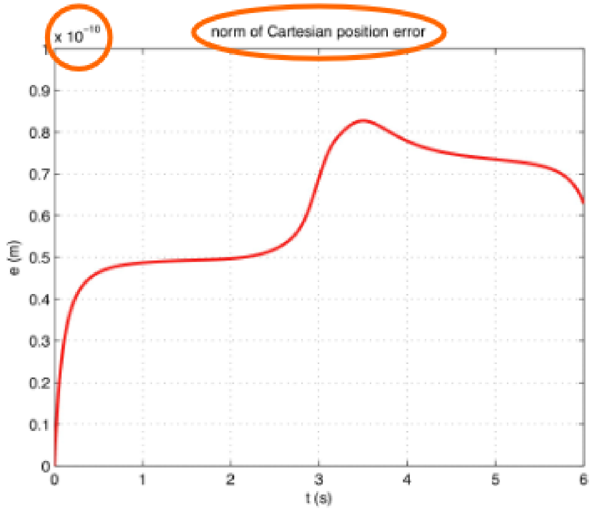




It measures how close we are during motion to the singularity

regular case





error due only to numerical integration (10^{-10})

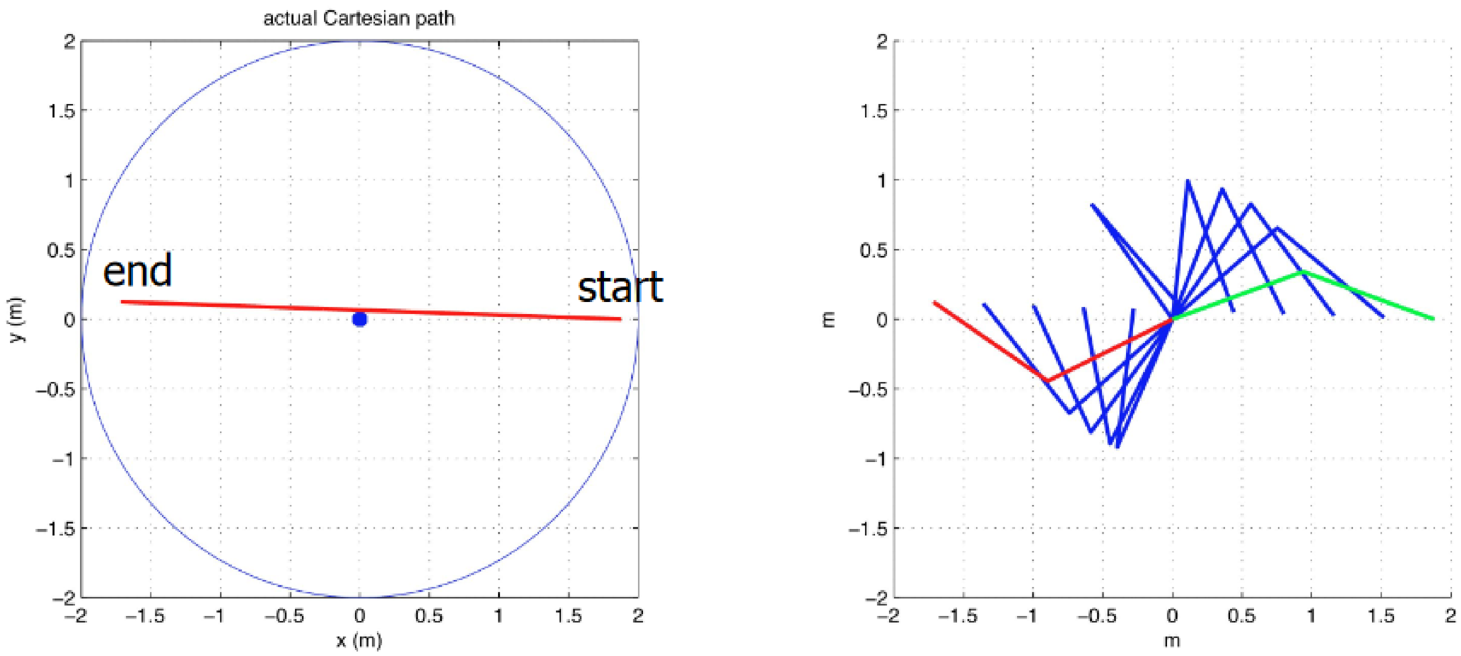
UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Simulation results

planar 2R robot in straight line Cartesian motion

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

close to singular case



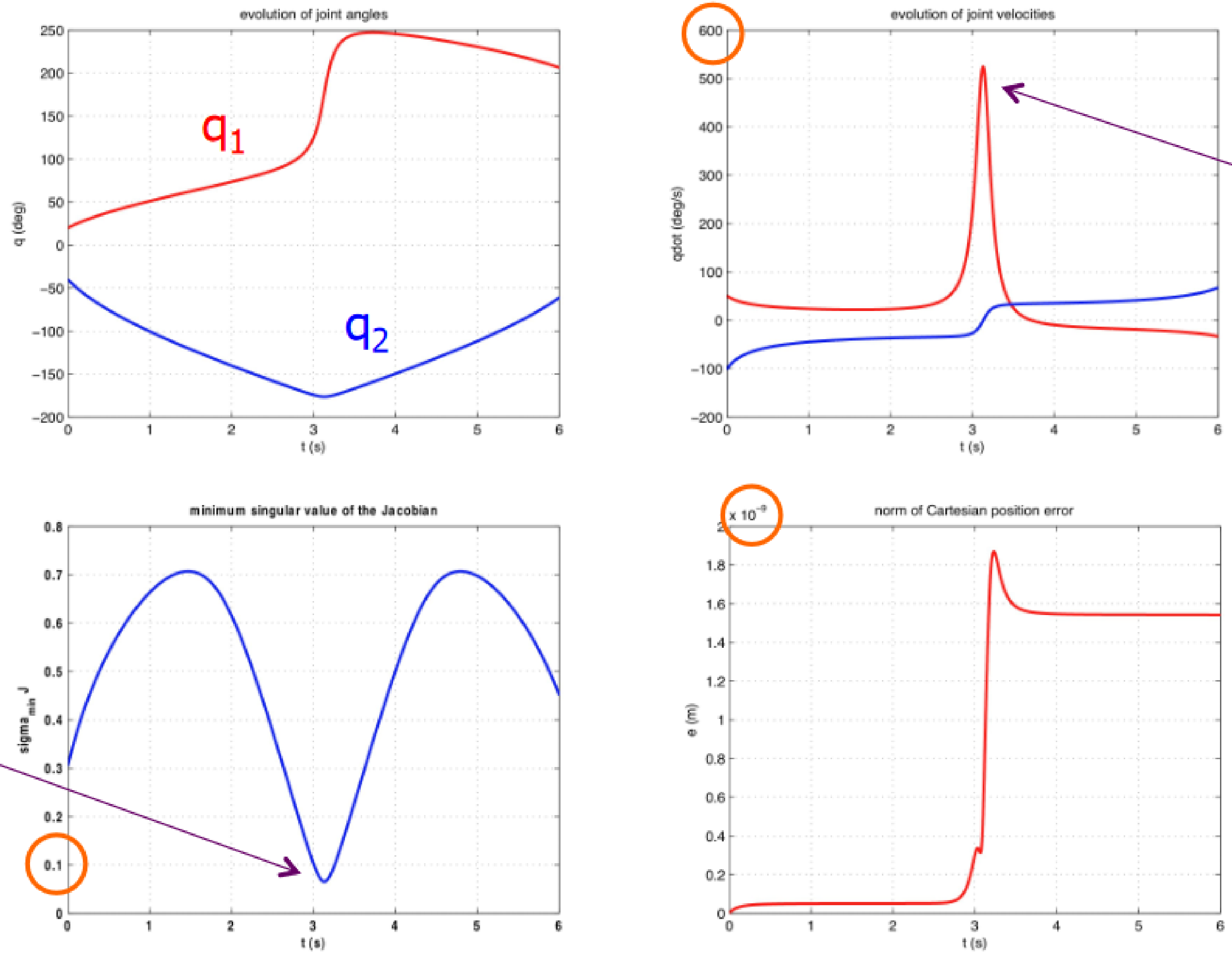
a line from right to left, at $\alpha=178°$ angle with x-axis, executed at constant speed v=0.6 m/s for T=6 s

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Simulation results

planar 2R robot in straight line Cartesian motion

path at α=178°



large peak of $\dot{q}_1$

close to singular case

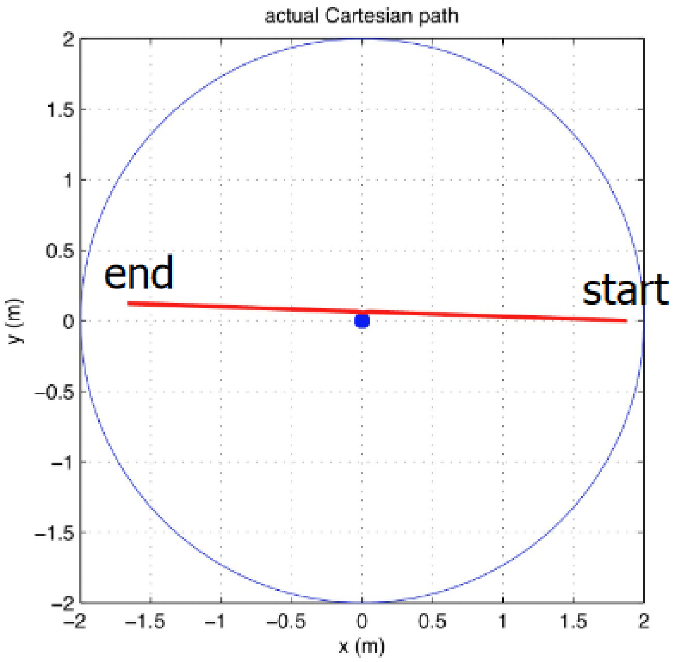
still very small, but increased numerical integration error ($2 \cdot 10^{-9}$)

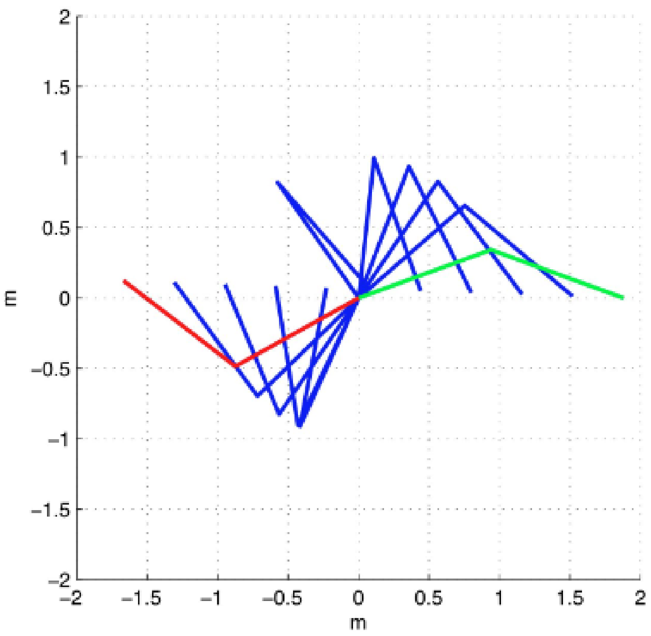
Simulation results

planar 2R robot in straight line Cartesian motion

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

close to singular case
with joint velocity saturation at V_i=300°/s



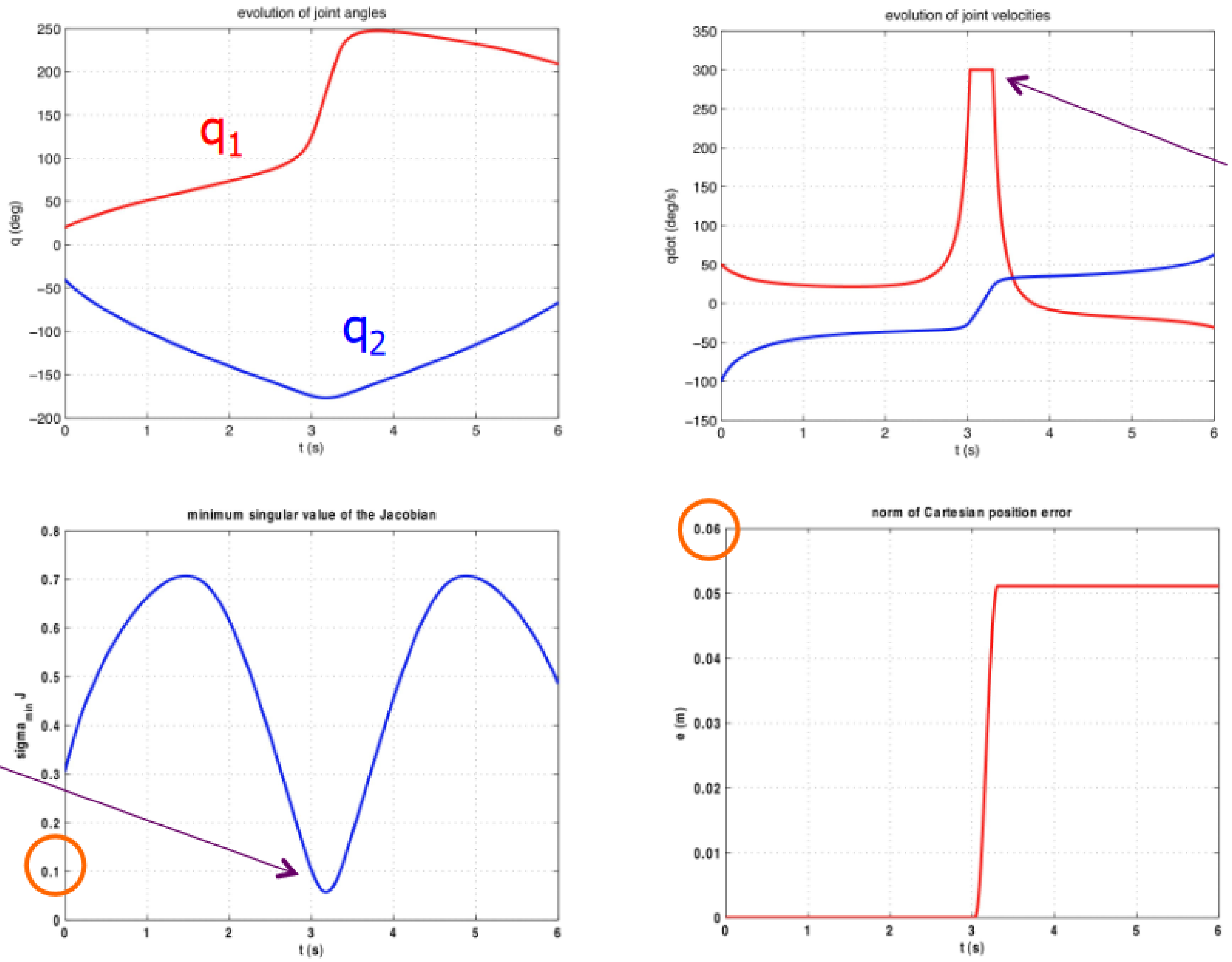


a line from right to left, at α=178° angle with x-axis,
executed at constant speed v=0.6 m/s for T=6 s

Simulation results

planar 2R robot in straight line Cartesian motion

path at
α=178°



saturated
value
of $\dot{q}_1$

close to
singular
case

actual
position
error!!
(6 cm)

Damped Least Squares method

$$\min_{\dot{q}} H = \frac{\lambda}{2}\|\dot{q}\|^2 + \frac{1}{2}\|J\dot{q} - v\|^2, \quad \lambda \geq 0$$

J_{DLS}

$$\dot{q} = (\lambda I_n + J^T J)^{-1} J^T v = J^T(\lambda I_m + J J^T)^{-1} v$$

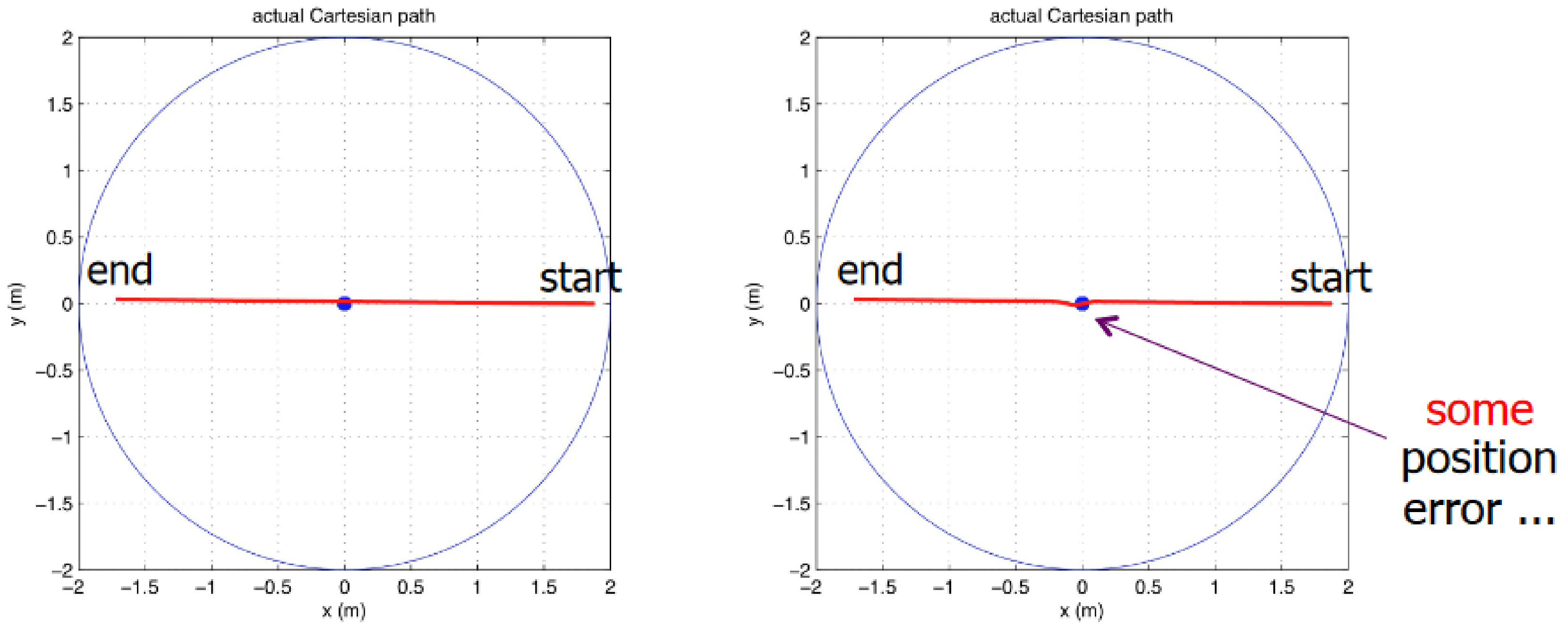
equivalent expressions, but this one is more convenient in redundant robots!

- inversion of differential kinematics as an optimization problem
- function H = weighted sum of two objectives (minimum error norm on achieved end-effector velocity and minimum norm of joint velocity)
- $\lambda = 0$ when "far enough" from a singularity
- with $\lambda > 0$, there is a (vector) error ε $(= v - J\dot{q})$ in executing the desired end-effector velocity v (check that $\varepsilon = \lambda\left(\lambda I_m + J J^T\right)^{-1} v$!), but the joint velocities are always reduced ("damped")
- J_{DLS} can be used for both m = n and m < n cases

planar 2R robot in straight line Cartesian motion

a comparison of inverse and damped inverse Jacobian methods
even closer to singular case

$\dot{q} = J^{-1}(q)\, v$ $\quad\quad$ $\dot{q} = J_{DLS}(q)\, v$



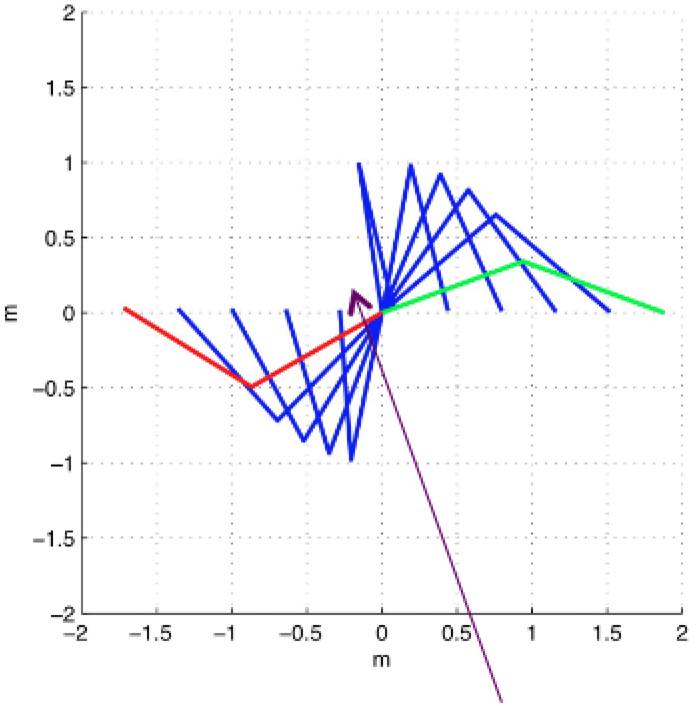
a line from right to left, at α=179.5° angle with x-axis,
executed at constant speed v=0.6 m/s for T=6 s

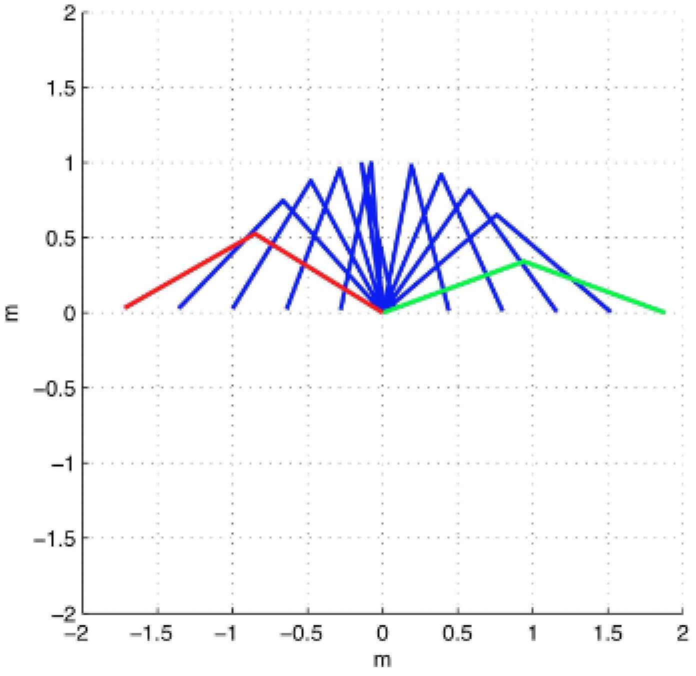
planar 2R robot in straight line Cartesian motion

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

path at $\alpha = 179.5°$

$\dot{q} = J_{DLS}(q)\, v$



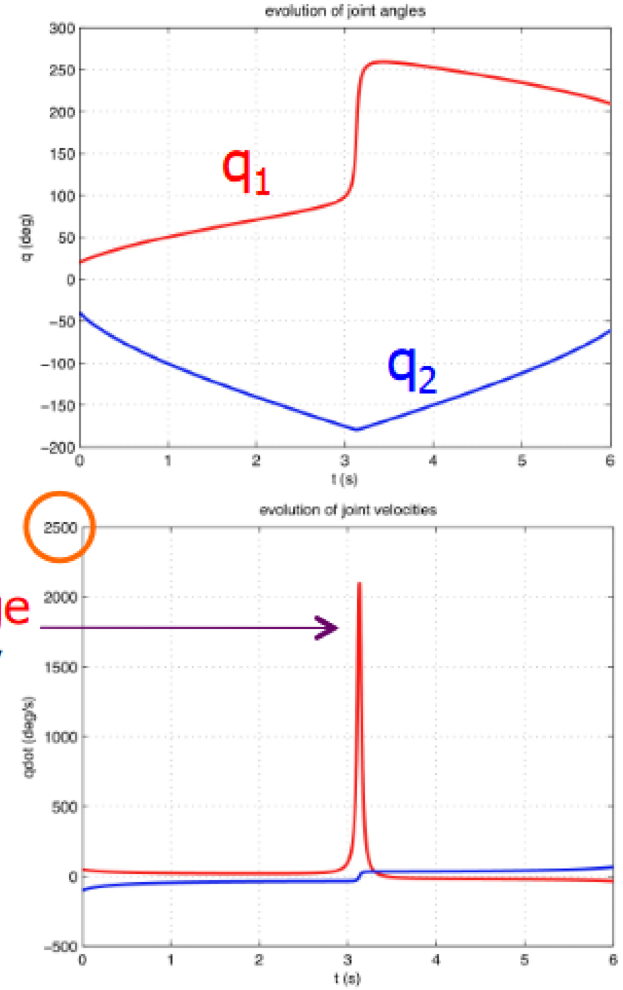


here, a very fast reconfiguration of first joint ...

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

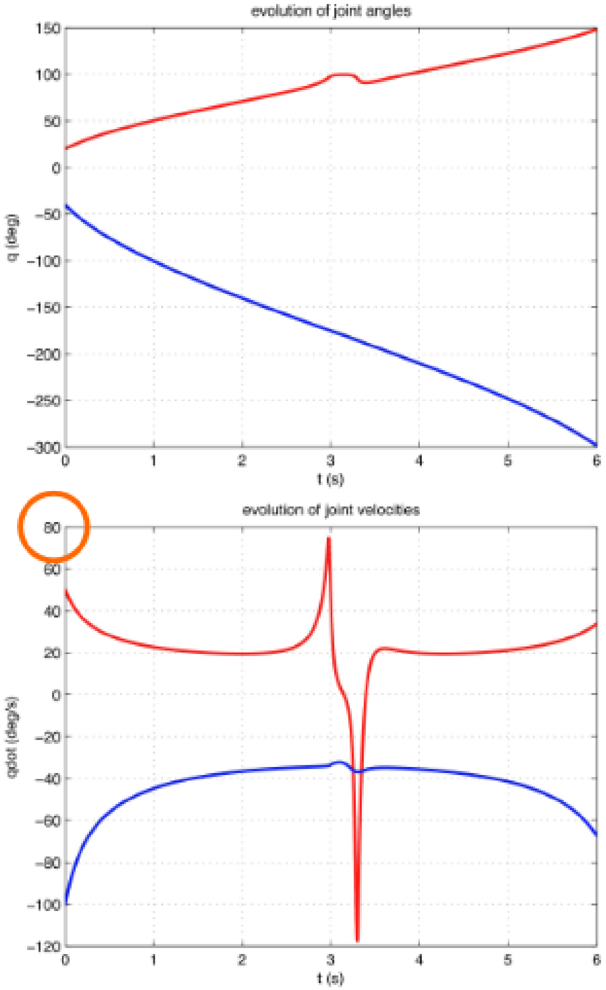
planar 2R robot in straight line Cartesian motion

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



extremely large peak velocity of first joint!!

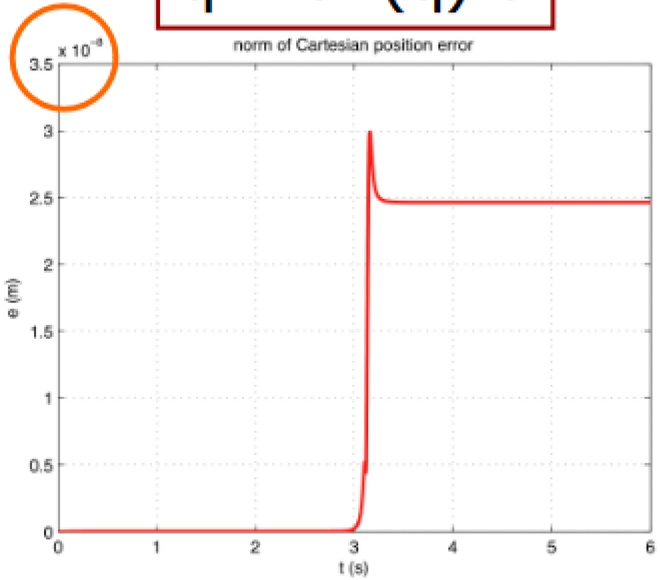
$\dot{q} = J_{DLS}(q)\, v$



smooth joint motion with limited joint velocities!

planar 2R robot in straight line Cartesian motion

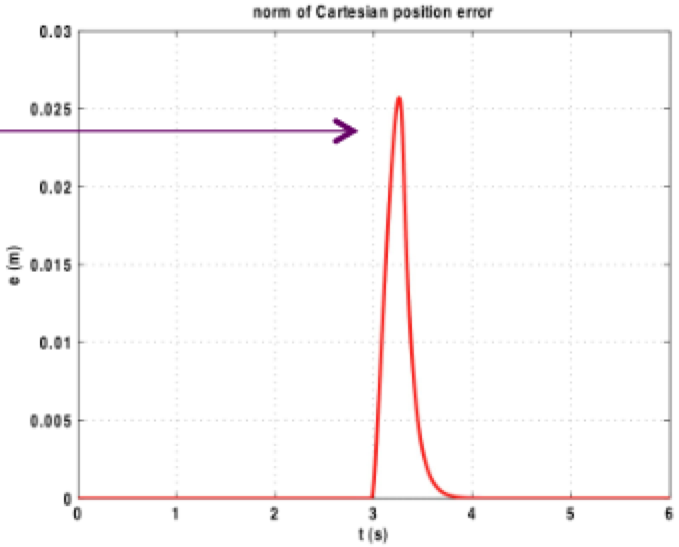
$\dot{q} = J^{-1}(q)\, v$



increased numerical integration error ($3 \cdot 10^{-8}$)

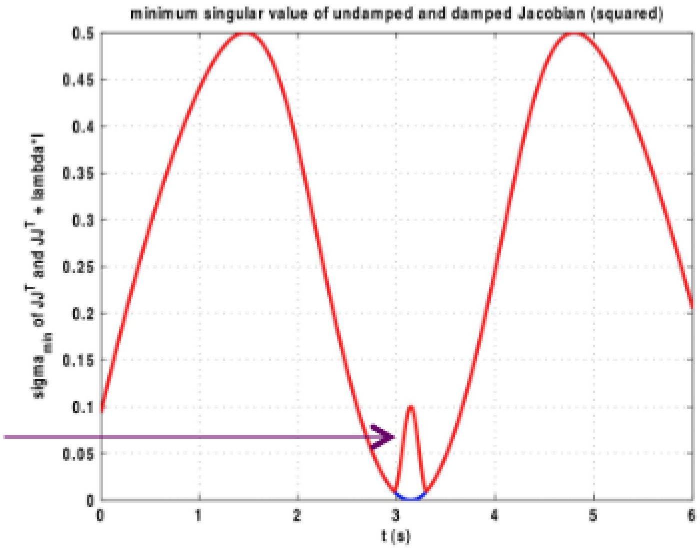
$\dot{q} = J_{DLS}(q)\, v$

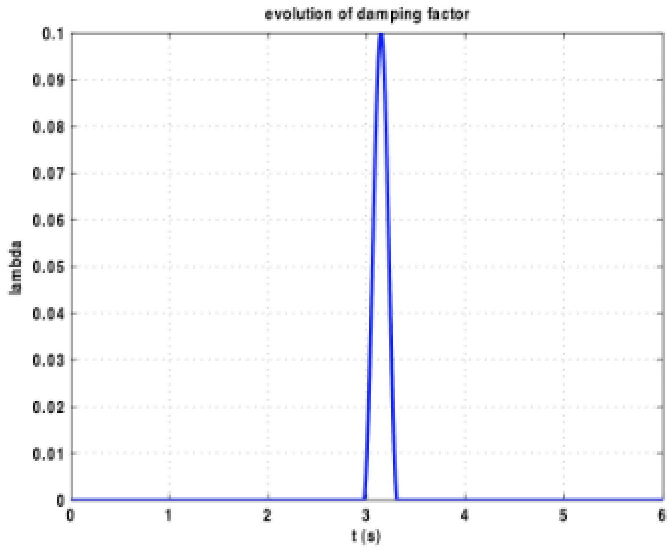
error (25 mm) when crossing the singularity, later recovered by feedback action ($v \Rightarrow v+Ke$)



minimum singular value of JJ^T and $\lambda I+JJ^T$

they differ only when damping factor is non-zero





damping factor λ is chosen non-zero only close to singularity!

Pseudoinverse method

a constrained optimization (minimum norm) problem

$$\min_{\dot{q}} H = \frac{1}{2}\|\dot{q}\|^2 \text{ such that } J\dot{q} - v = 0 \quad \Leftrightarrow \quad \min_{\dot{q}\in S} H = \frac{1}{2}\|\dot{q}\|^2$$

$$S = \left\{\dot{q} \in R^n : \|J\dot{q} - v\| \text{ is minimum}\right\}$$

solution $\dot{q} = J^{\sharp} v$ — pseudoinverse of J

- if $v \in \mathcal{R}(J)$, the constraint is satisfied (v is feasible)
- else $J\dot{q} = v^{\perp}$, where $v^{\perp}$ minimizes the error $\|J\dot{q} - v\|$

$v^{\perp}$: orthogonal projection of v on $\mathcal{R}(J)$

Properties of the pseudoinverse

it is the unique matrix that satisfies the four relationships

- $JJ^\sharp J = J \qquad J^\sharp J J^\sharp = J^\sharp$

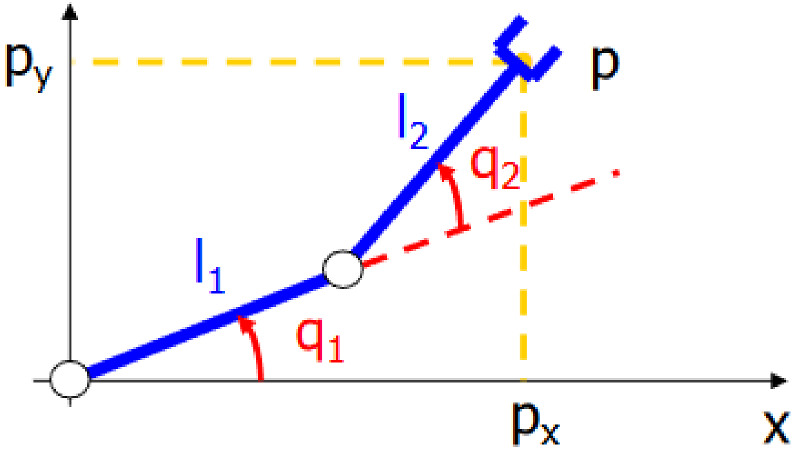
  $(J^\sharp J)^T = J^\sharp J \quad (JJ^\sharp)^T = JJ^\sharp$

- if rank ρ = m = n: $J^\sharp = J^{-1}$
- if ρ = m < n: $J^\sharp = J^T(JJ^T)^{-1}$

it always exists and is computed in general numerically using the SVD = Singular Value Decomposition of J

(e.g., with the MATLAB function **pinv**)

Numerical example



direct kinematics

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

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

analytical Jacobian

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

$$\det J(q) = l_1 l_2 s_2$$

Jacobian of 2R arm with $l_1 = l_2 = 1$ and $q_2 = 0$ (rank $\rho = 1$)

$$J = \begin{bmatrix} -2s_1 & -s_1 \\ 2c_1 & c_1 \end{bmatrix}$$

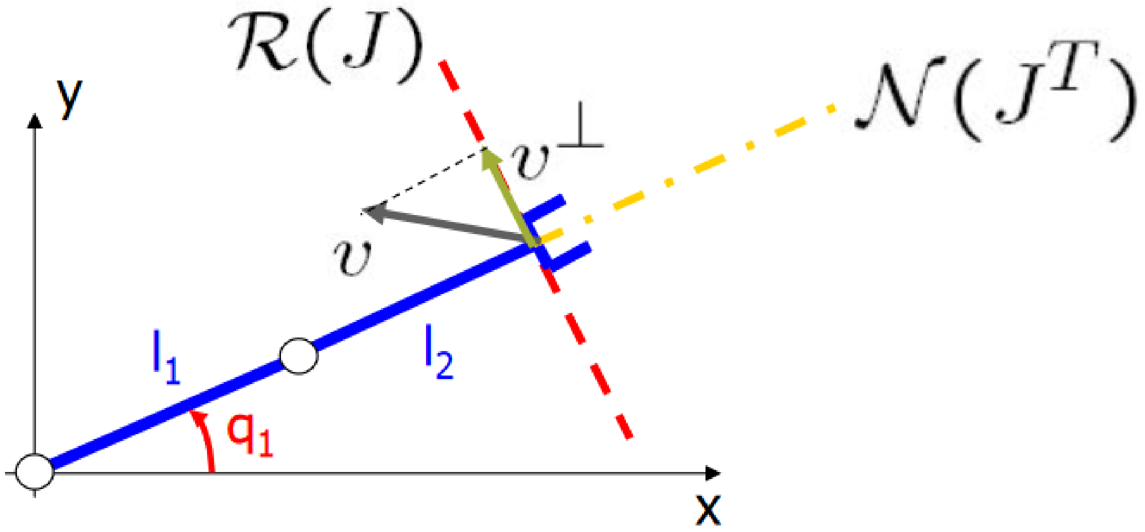
Numerical example

Jacobian of 2R arm with $l_1 = l_2 = 1$ and $q_2 = 0$ (rank $\rho = 1$)

$$J = \begin{bmatrix} -2s_1 & -s_1 \\ 2c_1 & c_1 \end{bmatrix} \quad J^\sharp = \frac{1}{5}\begin{bmatrix} -2s_1 & 2c_1 \\ -s_1 & c_1 \end{bmatrix}$$

$$\dot{q} = J^\sharp v$$

is the minimum norm joint velocity vector that realizes $v^\perp$



The end!