Advanced Robot Control and Learning

Prof. Sami Haddadin

Chapter 2: Task Space Modeling and Control

Advanced Robot Control - Outline

Differential Geometry in Robotics

Task Space Modeling and Control

Modeling

- Review on Kinematics and Differential Kinematics
- Differential Kinematics for Redundant Robots
- Review on Joint Space and Task Space Dynamics
- Dynamics of Redundant Robots

Control

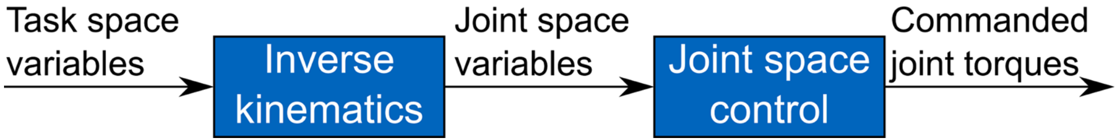
- Joint Space versus Task Space Control
- Example: Computed Torque Control in Task Space
- Motion/Torque Nullspace
- Task Prioritization
- Nullspace and stability

Modern Methods of Robot Control

Differential Kinematics (Revisited)

Tasks mostly related to end-effectors motion and contact forces (dynamic behavior)

- First intuition: kinematic control



Issue: solving inverse kinematics!

- To satisfy requirement of high performance control:
  - ➢ Model dynamic behavior as observed at the end-effector
  - ➢ Control motion and contact forces through control forces directly acting at level of end-effector
  - ➢ Generate these control forces by applying corresponding joint torques (through force transformation)

Differential Kinematics (Revisited)

<u>Manipulator **geometric model**</u>

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{q})$$

with

- Generalized joint coordinates $\boldsymbol{q}$
- End-effector configuration coordinates $\boldsymbol{x}$

<u>Manipulator **kinematic model**</u>

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

- Describes the time-derivatives of the end-effector parameters $\dot{\boldsymbol{x}} \in \mathbb{R}^m$ at a given configuration $\boldsymbol{q} \in \mathbb{R}^n$ as linear functions of the joint velocities $\dot{\boldsymbol{q}} \in \mathbb{R}^n$
- $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{m \times n}$ is the Jacobian matrix with elements

$$J_{ij}(\boldsymbol{q}) = \frac{\partial}{\partial q_j} f_i(\boldsymbol{q})$$

Differential Kinematics (Revisited)

Interpretations of the Jacobian matrix

- Matrix relating the time derivatives

  Manipulator kinematic model (seen previously)

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

- Matrix relating the differential $\delta\boldsymbol{q}$ of joint coordinates to the differential $\delta\boldsymbol{x}$ of end-effector configuration parameters

  Manipulator differential model

$$\delta\boldsymbol{x} = \boldsymbol{J}(\boldsymbol{q})\delta\boldsymbol{q}$$

Dimension of Jacobian matrix $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{m\times n}$

- m is the degree of freedom of the end-effector configuration ($m = 6$ in Cartesian space)
- n is the degree of freedom of the manipulator
- In the following, we assume $m = n$ and that the Jacobian is invertible

Joint Space Dynamics (revisited)

Robot model in joint coordinates:

$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_{\text{in}}$

with:

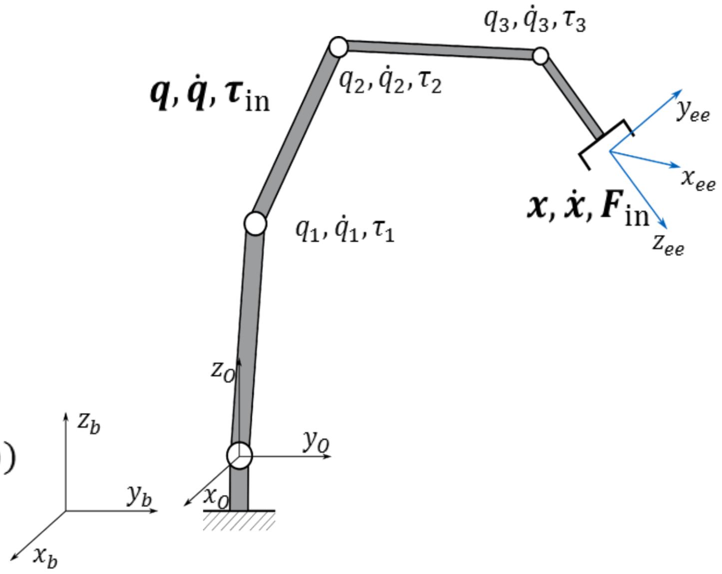
- $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \in \mathbb{R}^n$: joint positions, velocities, accelerations
- $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$: robot mass matrix
- $\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^n$: centrifugal and Coriolis forces
- $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$: gravitational force
- $\boldsymbol{\tau}_{\text{in}} \in \mathbb{R}^n$: command torques in joint space

Forward kinematics: local mapping between manifolds (Q and $SE(3)$)

$$\boldsymbol{f}: Q \rightarrow SE(3), \qquad \boldsymbol{x} = \boldsymbol{f}(\boldsymbol{q})$$

$\boldsymbol{f}$ is a local 1:1 mapping between manifolds for

- Non-redundant manipulators ($\dim(Q) = 6$)
- In non-singular configurations



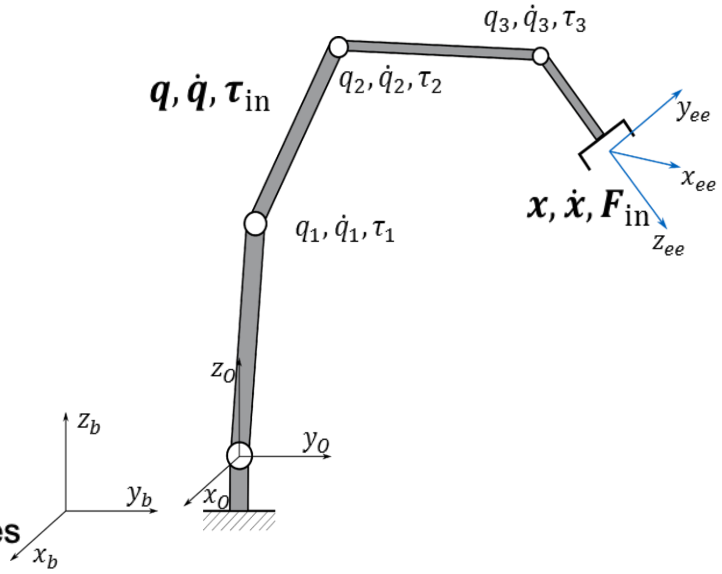
Task Space Dynamics

End-effector equation of motion in $SE(3)$ space

$\boldsymbol{M}_{\mathrm{C}}(\boldsymbol{q})\ddot{\boldsymbol{x}} + \boldsymbol{c}_{\mathrm{C}}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{F}_{\mathrm{g}}\,(\boldsymbol{q}) = \boldsymbol{F}_{\mathrm{in}}$

with:

- $\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}} \in \mathbb{R}^m$: task variables, velocities, accelerations
- $\boldsymbol{M}_{\mathrm{C}}(\boldsymbol{q}) \in \mathbb{R}^{m\times m}$: mass matrix in Cartesian space
- $\boldsymbol{c}_{\mathrm{C}}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^m$: centrifugal and Coriolis forces in Cartesian space
- $\boldsymbol{F}_{\mathrm{g}}(\boldsymbol{q}) \in \mathbb{R}^m$: gravitational force in Cartesian space
- $\boldsymbol{F}_{\mathrm{in}} \in \mathbb{R}^m$: command forces in task space



Note:

- Task space variables commonly reproduced from joint space variables through kinematic mappings!
- Indeed, robots hardly equipped with sensors measuring the end-effectors state

Joint space/Task space relationships

- Identity between two quadratic forms of kinetic energy:

$$\frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} = \frac{1}{2}\dot{\boldsymbol{x}}^T \boldsymbol{M}_{\mathrm{C}}(\boldsymbol{q})\dot{\boldsymbol{x}}$$

with the kinematic model ($\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$) (non-redundant robots)

$$\boldsymbol{M}_{\mathrm{C}}(\boldsymbol{q}) = \boldsymbol{J}^{-T}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})\boldsymbol{J}^{-1}(\boldsymbol{q})$$

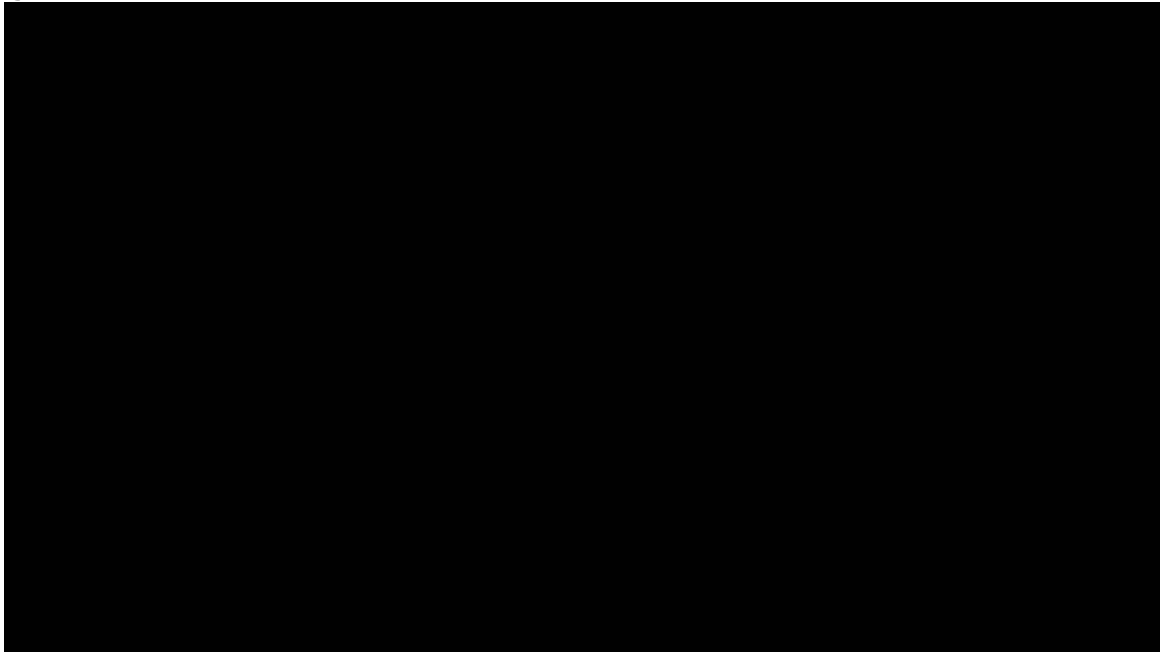
- $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{q})$; $\quad\quad \boldsymbol{q} = \boldsymbol{f}^{-1}(\boldsymbol{x})$
- $\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$; $\quad\quad \dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q})\dot{\boldsymbol{x}}$
- $\boldsymbol{F}_{\mathrm{in}} = \boldsymbol{J}^{-T}(\boldsymbol{q})\boldsymbol{\tau}_{\mathrm{in}}$; $\quad\quad \boldsymbol{\tau}_{\mathrm{in}} = \boldsymbol{J}^{T}(\boldsymbol{q})\boldsymbol{F}_{\mathrm{in}}$

MIRMI TUM

What is Redundancy?

https://www.youtube.com/watch?v=L9jhWTrv9rY

Redundancy in Robotics



https://www.youtube.com/watch?v=sZYBC8Lrmdo

Example – Franka Emika

- Task space: m dimensional $m \leq 6$
- Joint space Q: n dimensional

Redundant manipulator: $n > m$

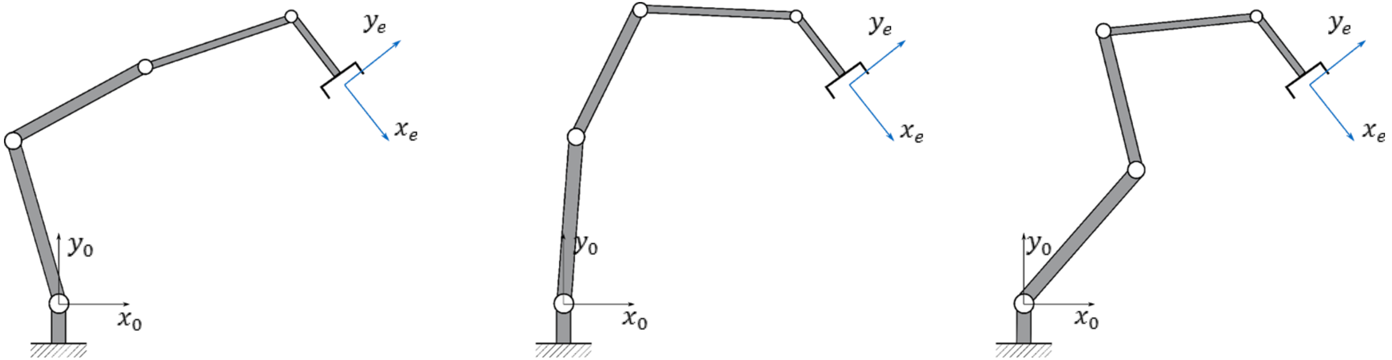
With $(n - m)$ degrees of redundancy

$n = 7, m = 6$

Source: Franka Emika

- Additionally, task redundancy $\longrightarrow$ concerns all type of manipulators!

Example – RRRR Manipulator



- Kinematic redundancy: four DOFs $(n = 4)$, position and orient the end-effector $(m = 3)$
- No influence on solvability of direct kinematics: $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{q})$
- Inverse kinematics: $\boldsymbol{q} = \boldsymbol{f}^{-1}(\boldsymbol{x})$ no unique solution, infinite joint configurations resulting in the desired end-effector configuration
- And differential inverse kinematics $\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$?

Differential Kinematics for Redundant Robots

- Recall: $\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial f(q)}{\partial q} \in \mathbb{R}^{m \times n}$, $\quad m = \dim(\boldsymbol{x}), \;\; n = \dim(\boldsymbol{q})$
- Redundant robots: more independent variables (n robot DOFs) than linear equations (m end-effector DOFs) $\longrightarrow$ underdetermined system of linear equations
- Jacobi Matrix is not quadratic, hence not invertible
- Approach: differential inverse kinematics as an optimization problem
- Cost-function to minimize:

$$\boldsymbol{g}(\dot{\boldsymbol{x}}) = \frac{1}{2} \; \dot{\boldsymbol{q}}^T \dot{\boldsymbol{q}} \; \rightarrow \min$$

- Subject to:

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q}) \dot{\boldsymbol{q}}$$

- Solution: Left-handed Pseudo-inverse

Dynamics of Redundant Robots

- $m < n \longrightarrow$ set of task coordinates $\boldsymbol{x} \in \mathbb{R}^m$ are insufficient to fully specify the configuration $\boldsymbol{q} \in \mathbb{R}^n$ of a redundant robot
- Remember:
  - $\boldsymbol{M}_{\mathrm{C}}(\boldsymbol{q})\ddot{\boldsymbol{x}} + \boldsymbol{c}_{\mathrm{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{F}_{\mathrm{g}}(\boldsymbol{q}) = \boldsymbol{F}_{\mathrm{in}}$
  - $\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q})\dot{\boldsymbol{x}}$
  - $\boldsymbol{F}_{\mathrm{in}} = \boldsymbol{J}^{-T}(\boldsymbol{q})\boldsymbol{\tau}_{\mathrm{in}}$
  - $\boldsymbol{M}_{\mathrm{C}}(\boldsymbol{q}) = \boldsymbol{J}^{-T}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})\boldsymbol{J}^{-1}(\boldsymbol{q})$

Redundant robots: the Jacobian $\boldsymbol{J}_{m\times n}$ is not quadratic, hence not invertible!

- Solution:

RightPseudo-inverse matrix: $\boldsymbol{J}^{\#}_{n\times m}$

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

- Weighted Pseudo-inverse: $\boldsymbol{J}^{\#}_{A} = \boldsymbol{A}^{-1}\boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{A}^{-1}\boldsymbol{J}^T)^{-1}$
- Moore-Penrose Pseudo-inverse: $\boldsymbol{J}^{\#} = \boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{J}^T)^{-1}$ with $\boldsymbol{A} = \boldsymbol{I}$

Dynamics of Redundant Robots

- Dynamic behaviour in taskspace…
  - $\boldsymbol{M}_{\mathrm{C,R}}(\boldsymbol{q})\ddot{\boldsymbol{x}} + \boldsymbol{c}_{\mathrm{C,R}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{F}_{\mathrm{g,R}}(\boldsymbol{q}) = \boldsymbol{F}_{\mathrm{in}}$ … becomes incomplete for redundant manipulators in motion!
  - $\dot{\boldsymbol{q}} = \boldsymbol{J}_A^{\#}(\boldsymbol{q})\dot{\boldsymbol{x}}$
  - $\boldsymbol{F}_{\mathrm{in}} = \boldsymbol{J}_A^{\#T}(\boldsymbol{q})\,\boldsymbol{\tau}_{\mathrm{in}}$
  - $\boldsymbol{M}_{\mathrm{C,R}}(\boldsymbol{q}) = \boldsymbol{J}_A^{\#T}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})\boldsymbol{J}_A^{\#}(\boldsymbol{q})$

- $\boldsymbol{J}_A^{\#}(\boldsymbol{q}) = \boldsymbol{A}^{-1}\boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{A}^{-1}\boldsymbol{J}^T)^{-1}$: different $\boldsymbol{A}^{-1} \longrightarrow$ different $\boldsymbol{M}_{\mathrm{C,R}}$'s!
- Solution…

First compute the inverse, also called mobility matrix: $\boldsymbol{M}_{\mathrm{C,R}}^{-1} = \boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{J}^T$

- We have: $\boldsymbol{p}^T\boldsymbol{M}_{\mathrm{C,R}}^{-1}\boldsymbol{p} = E_{kinetic}$ for $\boldsymbol{p} = \boldsymbol{M}_{\mathrm{C}}\,\dot{\boldsymbol{x}}$

Dynamics of Redundant Robots

- So far…
  - $J_A^{\#}(q) = A^{-1}J^T(JA^{-1}J^T)^{-1}$
  - $M_{\mathrm{C,R}} = (JM^{-1}J^T)^{-1}$
  - $M_{\mathrm{C,R}}(q) = J_A^{\#T}(q)M(q)J_A^{\#}(q)$

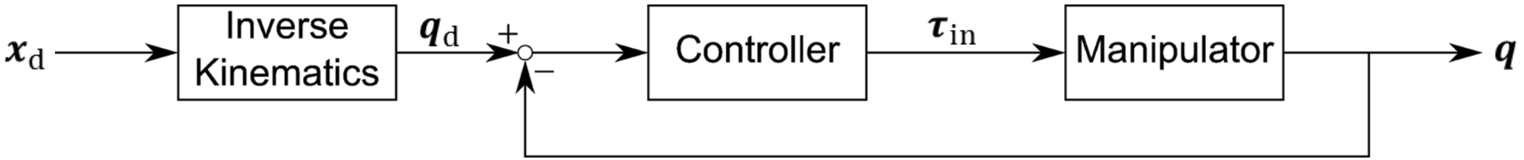
Let $A = M(q)$…

$$J_M^{\#}(q) = M^{-1}J^T(JM^{-1}J^T)^{-1}$$

Exercise: Re-check the equivalence of $M_{\mathrm{C,R}}$ with this expression of Jacobian pseudo-inverse

Joint space versus Task space Control

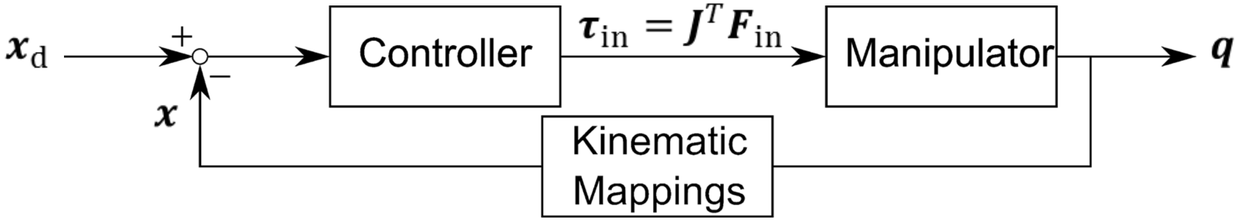
Joint space control



- Controller is typically designed to track desired joint motion
- But user specifies motion in terms of the end-effector coordinates
  - ➢ Motion first needs to be mapped in joint space via kinematic inversion
- During task execution measurements from joint states allow feedback controller to calculate the joint torques needed to track the desired joint motion
- Assumption: a time sequence of joint motions
  - ➢ Online trajectory adjustments are difficult to realize!

Joint space versus Task space Control

Task space control



- Increasing challenges typical robotic environment
  - ➢ Moving from the typical shop floor to human surroundings
- Online modifications become inevitable
- Task is described in the end-effector space and its precise control is of interest
  - ➢ Joint space control schemes are out of place
- While directly minimizing the task error
  - ➢ No need for explicit calculation of inverse kinematics
  - ➢ Inversion on velocity level: (generalized) inverse of the Jacobian

Computed Torque Control in Task space

- Problem: tracking control of time-varying trajectory $\boldsymbol{x}_{\mathrm{d}}(t), \dot{\boldsymbol{x}}_{\mathrm{d}}(t), \ddot{\boldsymbol{x}}_{\mathrm{d}}(t)$
- Including the manipulator dynamic model allows for effective trajectory tracking!
- Recall: End-effector equation of motion in $SE(3)$ space

$$\boldsymbol{M}_{\mathrm{C}}(\boldsymbol{q})\ddot{\boldsymbol{x}} + \boldsymbol{c}_{\mathrm{C}}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{F}_{\mathrm{g}}(\boldsymbol{q}) = \boldsymbol{F}_{\mathrm{in}}$$

- Intuitive approach: cancel out the nonlinear terms and decouple the dynamics

Selected control structure (PD controller):

$$\boldsymbol{F}_{\mathrm{in}} = \widehat{\boldsymbol{M}}_{\mathrm{C}}(\boldsymbol{q})\left(\ddot{\boldsymbol{x}}_{\mathrm{d}} + \boldsymbol{K}_{\mathrm{p}}(\boldsymbol{x}_{\mathrm{d}} - \boldsymbol{x}) + \boldsymbol{K}_{\mathrm{v}}(\dot{\boldsymbol{x}}_{\mathrm{d}} - \dot{\boldsymbol{x}})\right) + \hat{\boldsymbol{c}}_{\mathrm{C}}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \widehat{\boldsymbol{F}}_{\mathrm{g}}(\boldsymbol{q})$$

- $\hat{\cdot}$ denotes estimates of the inertia matrix, centrifugal and Coriolis forces, gravitational force
- $\boldsymbol{K}_{\mathrm{p}}, \boldsymbol{K}_{\mathrm{V}} \in \mathbb{R}^{m\times m}$ are positive definite gain matrices

Requirements:

- Direct drives (no gearbox) or
- Torque control $\boldsymbol{\tau}_{\mathrm{in}} = \boldsymbol{J}(\boldsymbol{q})^T \boldsymbol{F}_{\mathrm{in}}$

Computed Torque Control in Task space

Assumption:

Perfect estimates ($\widehat{\boldsymbol{M}}_{\mathrm{C}}(\boldsymbol{q}) = \boldsymbol{M}_{\mathrm{C}}(\boldsymbol{q}), \hat{\boldsymbol{c}}_{\mathrm{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{c}_{\mathrm{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $\widehat{\boldsymbol{F}}_{\mathrm{g}}(\boldsymbol{q}) = \boldsymbol{F}_{\mathrm{g}}(\boldsymbol{q})$)

- Resulting linear, decoupled error dynamics of the closed loop system:

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_{\mathrm{v}}\, \dot{\boldsymbol{e}} + \boldsymbol{K}_{\mathrm{p}} \boldsymbol{e} = \boldsymbol{0}$$

with

$$\boldsymbol{e} = (\boldsymbol{x}_{\mathrm{d}} - \boldsymbol{x})$$

Choice of diagonal matrices $\boldsymbol{K}_{\mathrm{v}}$ and $\boldsymbol{K}_{\mathrm{p}}$

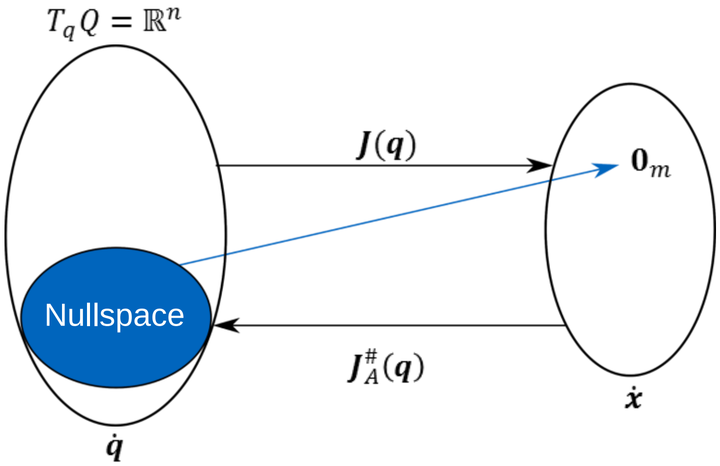
- For critically damped error dynamics:

$$k_{\mathrm{v},i} = 2\sqrt{k_{\mathrm{p},i}}\,, \qquad i = 1, \dots, m$$

The picture can't be displayed.

Controller

Nullspace – Motion



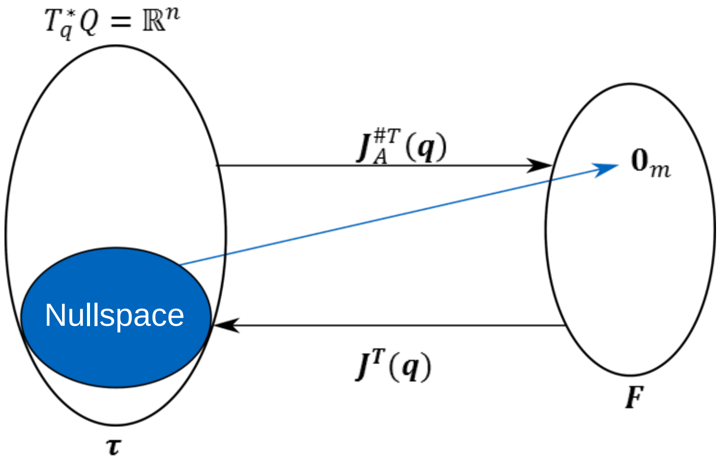
- Self-motions: $\boldsymbol{J}(\boldsymbol{q})\,\dot{\boldsymbol{q}}_N = 0$
- Solutions: $\dot{\boldsymbol{q}}_N = (\boldsymbol{I} - \boldsymbol{J}_A^{\#}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q}))\,\dot{\boldsymbol{q}}_0$
- Inverse kinematics: $\dot{\boldsymbol{q}} = \boldsymbol{J}_A^{\#}(\boldsymbol{q})\,\dot{\boldsymbol{x}} + (\boldsymbol{I} - \boldsymbol{J}_A^{\#}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q}))\,\dot{\boldsymbol{q}}_0$

Interpretations of the Pseudoinverse

- What is the role of the weighting matrix $\boldsymbol{A}$?
- So far:
  - $\dot{\boldsymbol{q}} = \boldsymbol{J}_A^{\#}(\boldsymbol{q})\,\dot{\boldsymbol{x}}$ (no self-motion)
  - $\boldsymbol{J}_A^{\#}(\boldsymbol{q}) = \boldsymbol{A}^{-1}\boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{A}^{-1}\boldsymbol{J}^T)^{-1}$

- For $\boldsymbol{A} = \boldsymbol{I}$ $\longrightarrow$ $\boldsymbol{J}^{\#} = \boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{J}^T)^{-1}$ the inverse kinematics will minimize $||\dot{\boldsymbol{q}}||$ under the constraint
$$\dot{\boldsymbol{x}} - \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} = 0$$
- Generally, for a weighting matrix $\boldsymbol{A}$ we minimize the norm $||\dot{\boldsymbol{q}}||_{\boldsymbol{A}^{-1}}$
- For example $\boldsymbol{A} = \frac{1}{2}\boldsymbol{M}^{-1}(\boldsymbol{q})$ we minimize the kinetic energy
$$||\dot{\boldsymbol{q}}||_{\boldsymbol{A}^{-1}} = \frac{1}{2}\dot{\boldsymbol{q}}^T\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

Nullspace – Torque



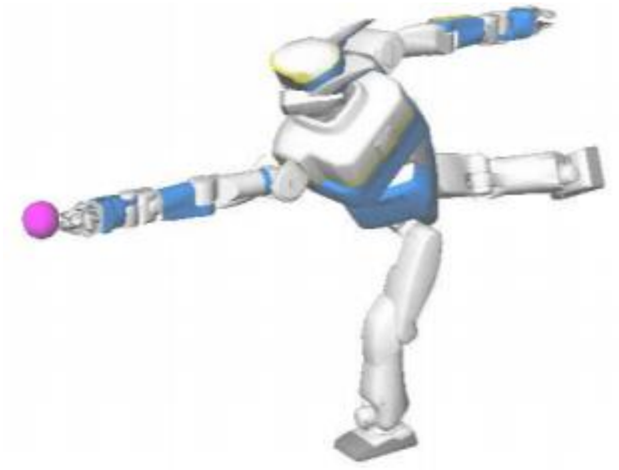
- Self-motions: $\boldsymbol{J}_A^{\#T}(\boldsymbol{q})\boldsymbol{\tau}_N = 0$
- Solutions: $\boldsymbol{\tau}_N = (\boldsymbol{I} - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{J}_A^{\#T}(\boldsymbol{q}))\boldsymbol{\tau}_0$
- Commanded joint torque: $\boldsymbol{\tau}_{\text{in}} = \underbrace{\boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{F}}_{\text{primary task}} + \underbrace{(\boldsymbol{I} - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{J}_A^{\#T}(\boldsymbol{q}))\boldsymbol{\tau}_0}_{\text{secondary task}}$

Nullspace Control and Task Prioritization

- Humanoid robots are highly redundant robots (usually up to 30 or more DoF).
- They can use null space control and task prioritization to handle several tasks at once.

Task: Reach the ball.

But more importantly: **Save your balance!**



Nullspace Control and Task Prioritization

Nullspace control

- No twist

$$0 = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}_N \Rightarrow \dot{\boldsymbol{q}}_N = \left(\boldsymbol{I} - \boldsymbol{J}_A^{\#}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})\right)\dot{\boldsymbol{q}}_0$$

- No wrench

$$\boldsymbol{\tau}_N = \boldsymbol{J}^T(\boldsymbol{q})\,0 \Rightarrow \boldsymbol{J}_A^{\#T}(\boldsymbol{q})\boldsymbol{\tau}_N = 0 \;\Rightarrow \boldsymbol{\tau}_N = \left(\boldsymbol{I} - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{J}_A^{\#T}(\boldsymbol{q})\right)\boldsymbol{\tau}_0$$

Task prioritization

- On velocity level
- On torque level

primary task secondary task

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_A^{\#}(\boldsymbol{q})\,\dot{\boldsymbol{x}} + \left(\boldsymbol{I} - \boldsymbol{J}_A^{\#}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})\right)\dot{\boldsymbol{q}}_0$$

$$\boldsymbol{\tau}_{\text{in}} = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{F} + \left(\boldsymbol{I} - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{J}_A^{\#T}(\boldsymbol{q})\right)\boldsymbol{\tau}_0$$

Stability in the Nullspace

- Robot converges in the Cartesian space, but unstable in the nullspace?!
- Remember:
  - $\boldsymbol{M}_{\mathrm{C,R}}(\boldsymbol{q})\ddot{\boldsymbol{x}} + \boldsymbol{c}_{\mathrm{C,R}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{F}_{\mathrm{g,R}}(\boldsymbol{q}) = \boldsymbol{F}_{\mathrm{in}}$
- The computed torque approach leads to
  - $\ddot{\boldsymbol{e}} + \boldsymbol{K}_{\mathrm{v}}\,\dot{\boldsymbol{e}} + \boldsymbol{K}_{\mathrm{p}}\boldsymbol{e} = \boldsymbol{0}$
- Now the nullspace component‘s possible responsibilities:
  - Providing a damping term

$$\boldsymbol{\tau}_0 = -\boldsymbol{K}_{\mathrm{v}}\,\dot{\boldsymbol{q}}$$

  - Avoidance for end-stops

$$\boldsymbol{\tau}_0 = -\boldsymbol{K}_{\mathrm{p}}\,(\boldsymbol{q} - \boldsymbol{q}_0)$$

  - Singularity avoidance
  - Other optimization criteria

References

- O. Khatib, Lecture Notes: Advanced Robotic Manipulation
- O. Khatib, Inertial Properties in Robotic Manipulation: An Object Level Framework, Int. Journal of Robotics Research, Vol.14, No1, 1995, pp.19-3
- O. Kanoun, Contribution à la planification de mouvement pour robots humanoïdes, PhD diss., 2009.