Institut für Mechatronische Systeme
imes Leibniz Universität Hannover

Leibniz Universität Hannover

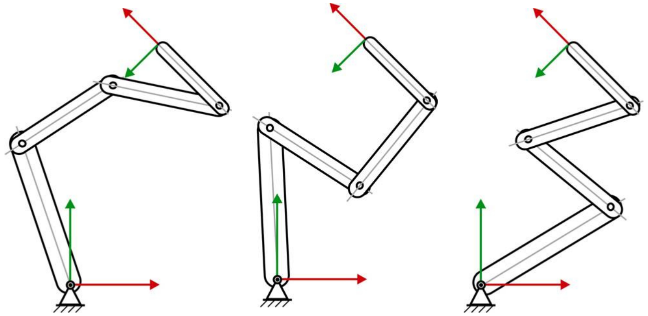
Robotics I

06. Motion Planning (Introduction)

Review

What (design) measures can be taken to avoid collisions and singularities and improve other performance characteristics?

- Introduction
- Redundancy and (differential) kinematics
- Null space as a homogeneous solution of differential kinematics
- Redundancy and inverse kinematics



Example: Planar RRRR robot
$(\dim(\boldsymbol{q}) = 4,\, \dim(\boldsymbol{x}_\mathrm{E}) = 3)$

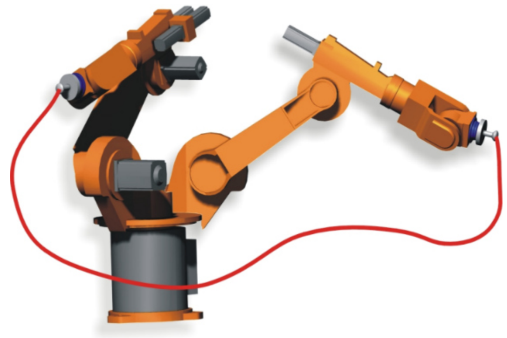
KUKA LBR (kinematically redundant)
$(\dim(\boldsymbol{q}) = 7,\, \dim(\boldsymbol{x}_\mathrm{E}) = 6)$

Literature:
G. Hirzinger et al. - DLR's torque-controlled light weight robot III - are we reaching the technological limits now?
A. Albu-Schäffer et al - The DLR lightweight robot: design and control concepts for robots in human environments
N. N. - Eigenvalues and eigenvectors of matrices

Total Content

Control
- Sensors
- Advanced Control Methods
- Multi-Axis Control
- Single-Axis Control
- Dynamics: Newton-Euler and Lagrange
- Motion Planning
- Kinematically Redundant Robots
- Jacobian Matrix – Velocities and Forces
- Forward and Inverse Kinematics
- Coordinate Transformations
- Introduction

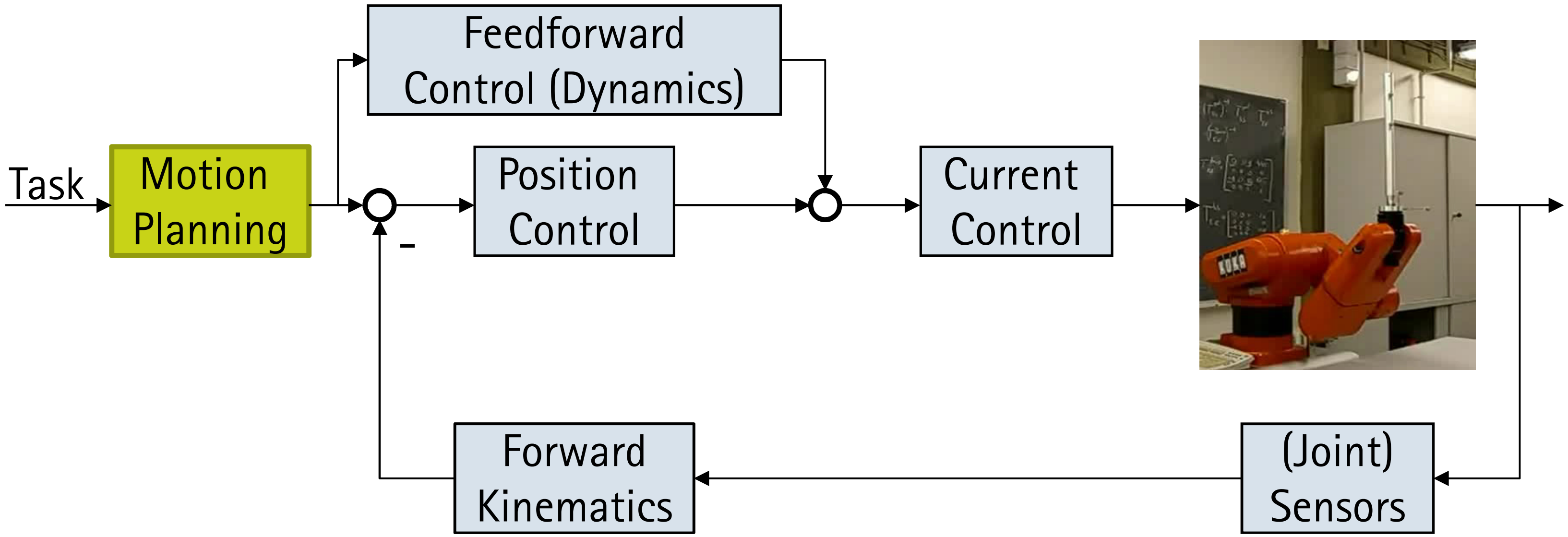
Kinematics and Dynamics



Total Content

Connection of essential lecture contents in the form of a control loop:

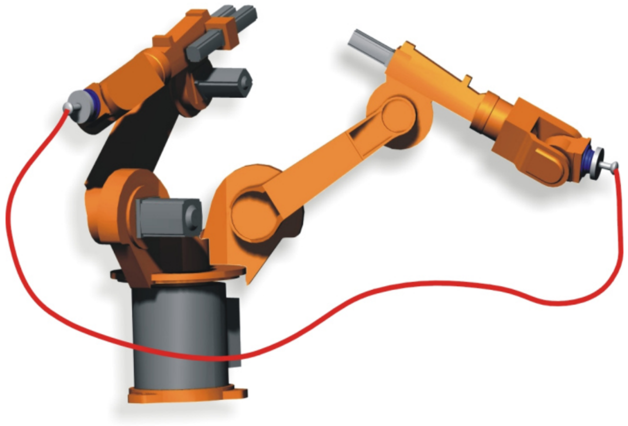
Motion planning: Generation of target trajectories for robots, can be carried out both in the joint and task space



Motion Planning

Design of a continuous motion sequence with consideration of constraints

- Definitions and basics
- Motion planning of a single axis
- Motion planning in the joint space
- Motion planning in the task space
- Interpolation of orientations
- Holistic motion planning in the task space

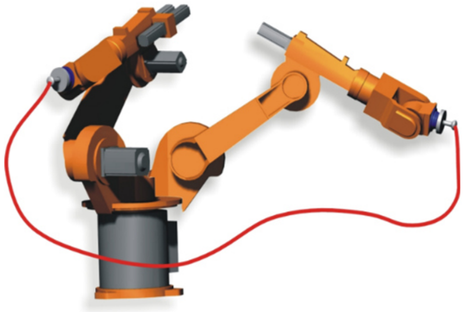


Literature:

B. Heimann et al. - Mechatronics (Components - Methods - Examples), pp. 211-222
L. Sciavicco et al - Modeling and Control of Robot Manipulators, pp. 169-196
J. J. Craig - Introduction to Robotics Mechanics and Control, pp. 227-256
W. Khalil - Modeling, Identification & Control of Robots, pp. 313-346
E. B. Dam - Quaternions, Interpolation and Animation, pp. 40-48
D. Eberly - Quaternion Algebra and Calculus
D. Constantinescu - Smooth and time-optimal trajectory planning for industrial manipulators along specified paths, pp. 233-249
L. Biagiotti - Trajectory planning for automatic machines and robots

Definitions and Basics

Path
Geometric location of all points/locations



Trajectory (Motion)
Path with information about the time schedule

Path planning

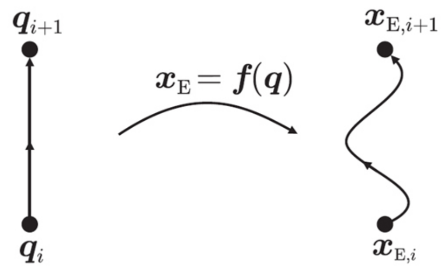
Selection of a path from the set of all possible paths from a point P_{S} to a point P_{G}, for example the geometrically shortest path, regardless of the time.

(Kinematic) Motion/Trajectory planning

Planning a continuous transition from a point P_i to a subsequent point P_{i+1}. Linking of location and time, taking into account a kinematic model (motion specification). With regard to industrial robots, motion planning can take place in the joint space or in the task space (see later).

Definitions and Basics

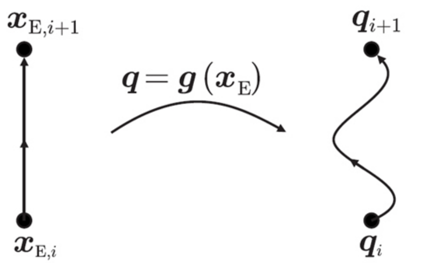
Configuration spaces of motion planning



Motion planning in joint space

Linear movement in the joint space

➡ Complex movements in the task space (due to the forward kinematics $\boldsymbol{x}_{\mathrm{E}} = \boldsymbol{f}(\boldsymbol{q})$)



Motion planning in task space

Analog: linear movement in the task space

➡ Complex movements in the joint space (due to the inverse kinematics $\boldsymbol{q} = \boldsymbol{g}(\boldsymbol{x}_{\mathrm{E}})$)

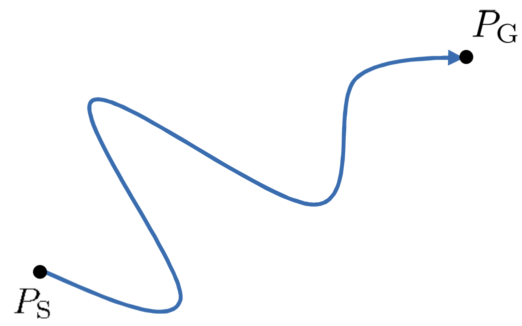
Motion planning in the joint space ➡ Low computing effort and comparatively simple avoidance of singular configurations

Motion planning in the task space ➡ Monitoring of possible collisions

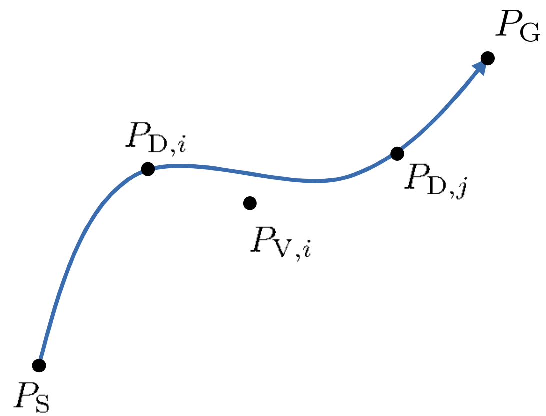
Definitions and Basics

Variants of robot paths

Point-to-Point (PTP)
The robot moves in a predetermined time from one Starting point P_S to a goal point P_G



Continuous-Path (CP)
The robot follows a predetermined path, which is defined by a large number of path points of different types.



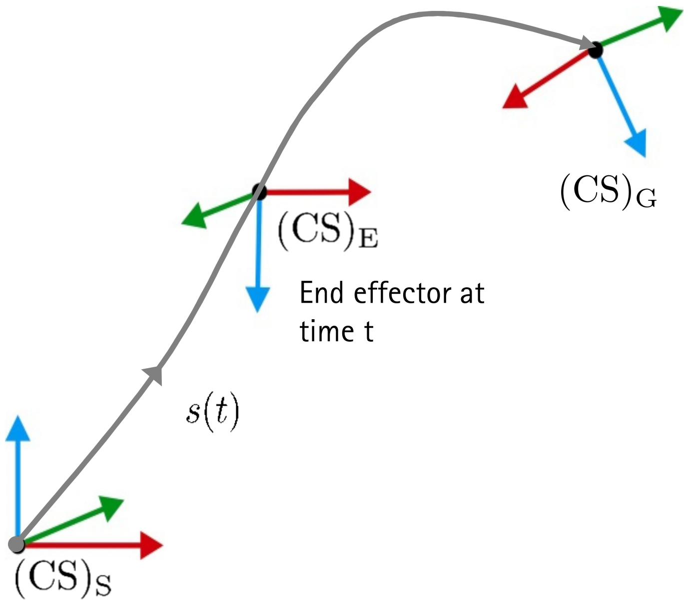
Definitions and Basics

Path coordinate
The path coordinate s represents the course of the space curve $\boldsymbol{x}(s)$ and is normalized to the path length $l\,(0 \leq s \leq 1)$. The partial derivatives according to the path coordinate describe the path gradient $\boldsymbol{x}'(s)$ and curvature $\boldsymbol{x}''(s)$.

$$\boldsymbol{x}'(s) = \frac{\partial \boldsymbol{x}(s)}{\partial s} \qquad \boldsymbol{x}''(s) = \frac{\partial^2 \boldsymbol{x}(s)}{\partial s^2}$$

The path coordinate is given as a time function $s(t)$, and results in a trajectory $\boldsymbol{x}(s(t))$.

➡ Allows changes over time of a trajectory independent of the path.



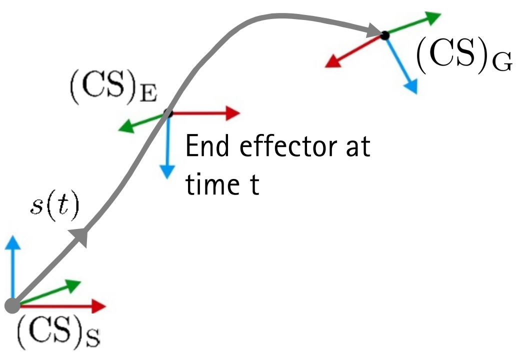
Definitions and Basics

Path coordinate

The time derivative of the path coordinate is required to describe a trajectory:

$$\dot{\boldsymbol{x}}(s(t)) = \frac{\partial \boldsymbol{x}}{\partial s}\frac{\mathrm{d}s}{\mathrm{d}t} = \boldsymbol{x}'\,\dot{s},$$

$$\ddot{\boldsymbol{x}}(s(t)) = \underbrace{\boldsymbol{x}''\,\dot{s}^2}_{\text{Centripetal accel.}} + \underbrace{\boldsymbol{x}'\,\ddot{s}}_{\text{Path accel.}}$$



Attention

- The resulting acceleration of the end effector $\ddot{\boldsymbol{x}}_\mathrm{E}$ can be higher than the specified path acceleration $\ddot{s}$ due to centripetal effects on curved path segments!
- Linear interpolation of rotations, described by compound rotations, do not necessarily have a constant velocity curve due to the ambiguous representation.

Institut für Mechatronische Systeme
imes Leibniz Universität Hannover

Leibniz Universität Hannover

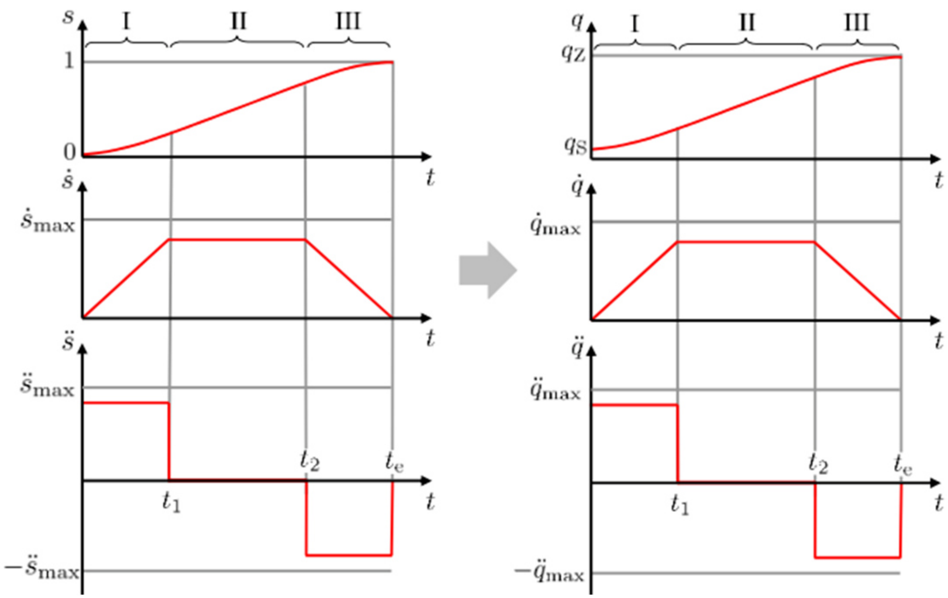
Robotics I

06. Motion Planning (Single Axis)

Motion Planning of a Single Axis

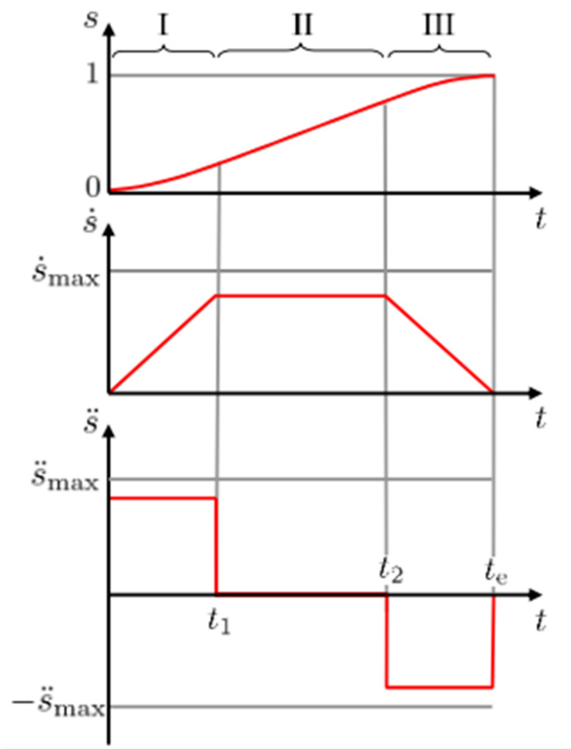
Relationship between path coordinate and joint angle trajectory

Qualitatively, $s(t)$ and $\boldsymbol{q}(t)$ are identical in the joint space. The curves differ in the scaling and the offset.



Motion Planning of a Single Axis

Point-to-point interpolation with composite polynomials
One-dimensional case: Movement of a robot joint from $\boldsymbol{q}_\mathrm{S}$ to $\boldsymbol{q}_\mathrm{G}$



Trapezoidal velocity profile (constant speed):

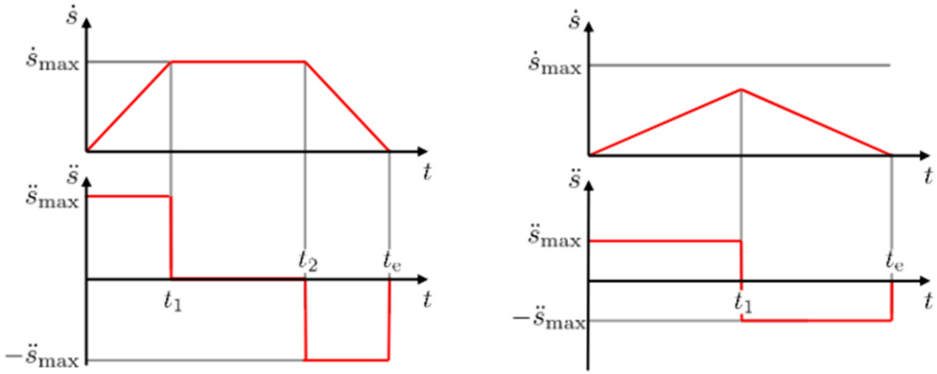
C[1] - continuous trajectory of the joint angles (shock-free, not jerk-free or jerk-limited), approximated by linear interpolation
with quadratic transitions

$$s(t) = \begin{cases} a_{0_1} + a_{1_1}\, t + a_{2_1}\, t^2, & 0 \leq t \leq t_1 \quad \text{(I)} \\ a_{0_2} + a_{1_2}\, t, & t_1 \leq t \leq t_2 \quad \text{(II)} \\ a_{0_3} + a_{1_3}\, t + a_{2_3}\, t^2, & t_2 \leq t \leq t_\mathrm{e} \quad \text{(III)} \end{cases}$$

➡ Continuous position and velocity curve,
discontinuous acceleration curve

Motion Planning of a Single Axis

Point-to-point interpolation with composite polynomials
Bang-bang control: Trapezoidal velocity profile with maximum possible acceleration, with the aim of minimizing the trajectory time t_e.



➡ Movement with maximum possible acceleration until the maximum speed is reached (left)
➡ Maximum speed is not reached if the acceleration phase is too short (right)

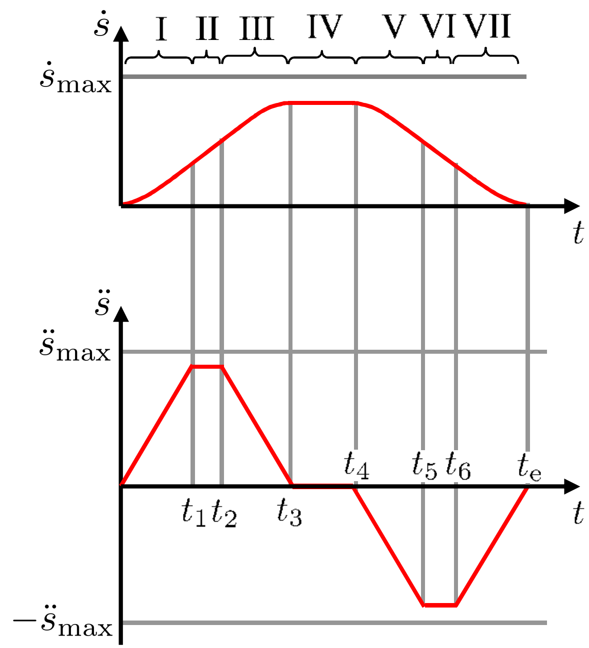
Motion Planning of a Single Axis

Point-to-point interpolation with composite polynomials

Trapezoidal acceleration profile (continuous acceleration)

C² - continuous trajectory of the joint angles (shock- and jerk-free)

$$s(t) = \begin{cases} a_{0_1} + a_{1_1}\,t + a_{2_1}\,t^2 + a_{3_1}\,t^3, & 0 \leq t \leq t_1 \quad \text{(I)} \\ a_{0_2} + a_{1_2}\,t + a_{2_2}\,t^2, & t_1 \leq t \leq t_2 \quad \text{(II)} \\ a_{0_3} + a_{1_3}\,t + a_{2_3}\,t^2 + a_{3_3}\,t^3, & t_2 \leq t \leq t_3 \quad \text{(III)} \\ a_{0_4} + a_{1_4}\,t, & t_3 \leq t \leq t_4 \quad \text{(IV)} \\ \ldots \end{cases}$$



➡ Seven polynomials for calculating a movement of a starting point $\boldsymbol{q}_{\mathrm{S}}$ to a destination $\boldsymbol{q}_{\mathrm{G}}$

➡ Continuous position, velocity and acceleration curve

Robotics I

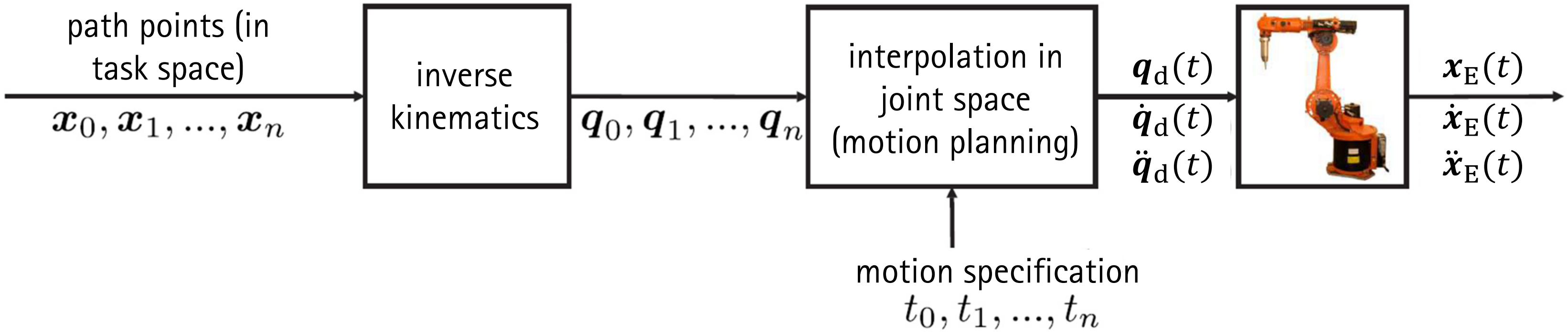
06. Motion Planning (Joint Space)

Motion Planning in the Joint Space

Classic computing process

Path points $\boldsymbol{x}_i$ (given in the task space) are transformed into the joint space using inverse kinematics.
Calculation of the target path, target speed and target acceleration based on the resulting joint angles $\boldsymbol{q}_i$

Advantage: complex calculation of the inverse kinematics not in the robot's control cycle



Problem: Consideration of obstacles (only in the task space)

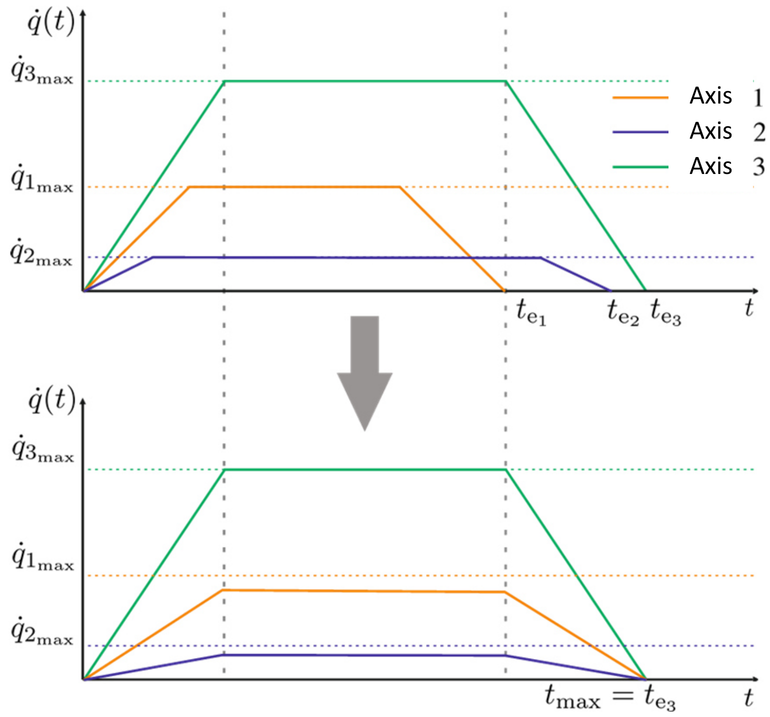
Motion Planning in the Joint Space

Synchronization to "slowest joint movement"
Previously: one-dimensional case of a joint parameter $\boldsymbol{q}_i$
With n degrees of freedom, different drive powers bring the axes to a standstill at different times

➡ Synchronization to the "slowest joint" of the robot for simultaneous movement of all axes

Motion Planning in the Joint Space

Synchronization to "slowest joint movement"



Procedure

1. Determination of t_{e_k} for each joint k
2. Determination of the maximum trajectory time of all joints

$$t_{\mathrm{max}} = \max_k \, t_{\mathrm{e}_k}$$

3. Determination of the trajectory time of all joints based on the maximum time
4. (Re)calculation of the trajectories $(k\text{-}1)$

Robotics I

06. Motion Planning (Task Space)

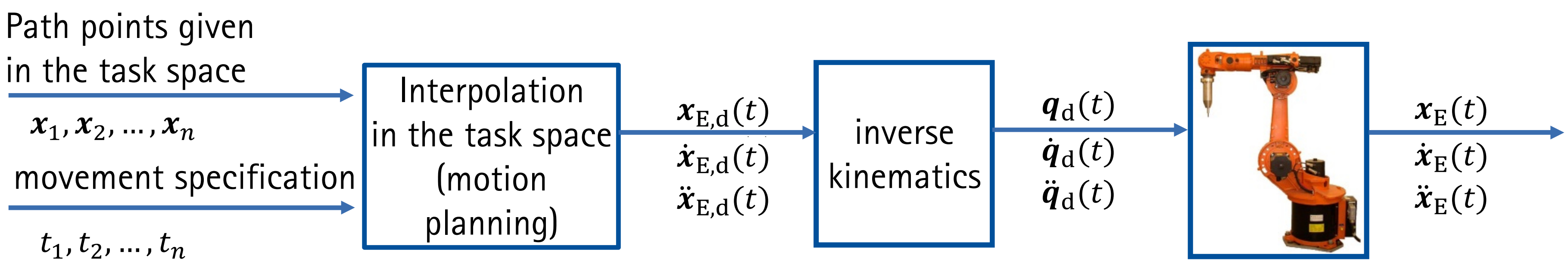
Motion Planning in the Task Space

Classic computing process

Similar to the motion planning in Joint space

The joint variables $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$ are replaced by the end effector variables $\boldsymbol{x}_{\mathrm{E}}$, $\dot{\boldsymbol{x}}_{\mathrm{E}}$, $\ddot{\boldsymbol{x}}_{\mathrm{E}}$

Problem: The inverse kinematics must be calculated in the control loop of the robot (usually $1\ \mathrm{ms}$), which requires considerably higher computing power.



Problem: Dynamic coupling of the drives (more in chapter 7)

Motion Planning in the Task Space

Task space trajectory from geometry primitives

The robot guides its tool along a straight line and/or a circular segment

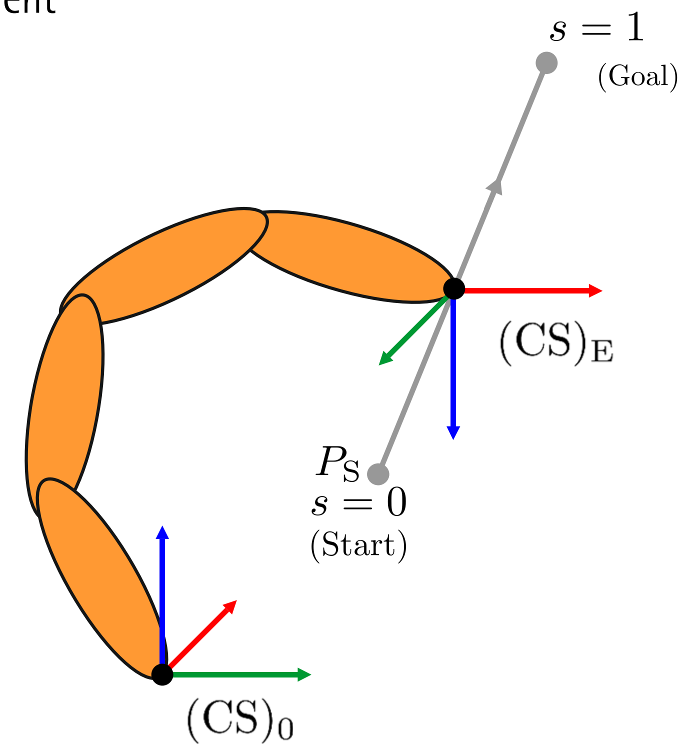
➡ Useful for processes such as gluing or welding

Linear segment

Simplest form of Cartesian path geometry

Movement of the end effector on a straight line

$$\boldsymbol{x}_{\mathrm{E}}(t) = \boldsymbol{x}_{\mathrm{E,S}} + s(t)\,(\boldsymbol{x}_{\mathrm{E,G}} - \boldsymbol{x}_{\mathrm{E,S}})$$



Motion Planning in the Task Space

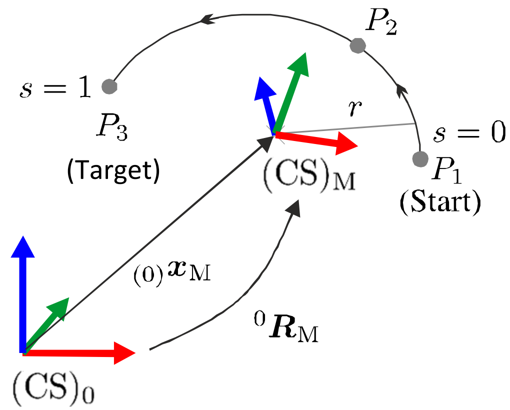
Task space trajectory from geometry primitives
Circular segment

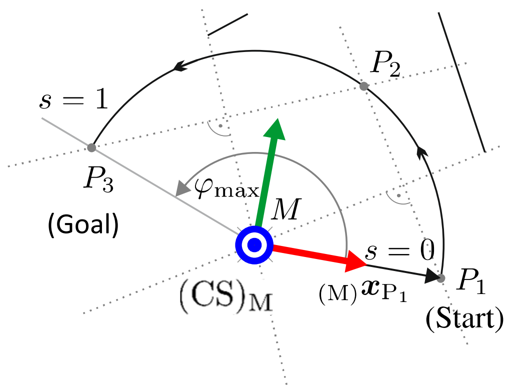
Path coordinate $s(t)$ for interpolation of the circle

Definition of a circular coordinate system $(\mathrm{CS})_\mathrm{M}$

- ${}_{(\mathrm{M})}\boldsymbol{z}\text{-axis}$ points out of the circular plane so that the angle of rotation φ_max is mathematically positive
- ${}_{(\mathrm{M})}\boldsymbol{x}\text{-axis}$ points from the center of the circle M to point P_1
- ${}_{(\mathrm{M})}\boldsymbol{y}\text{-axis}$ is perpendicular to it

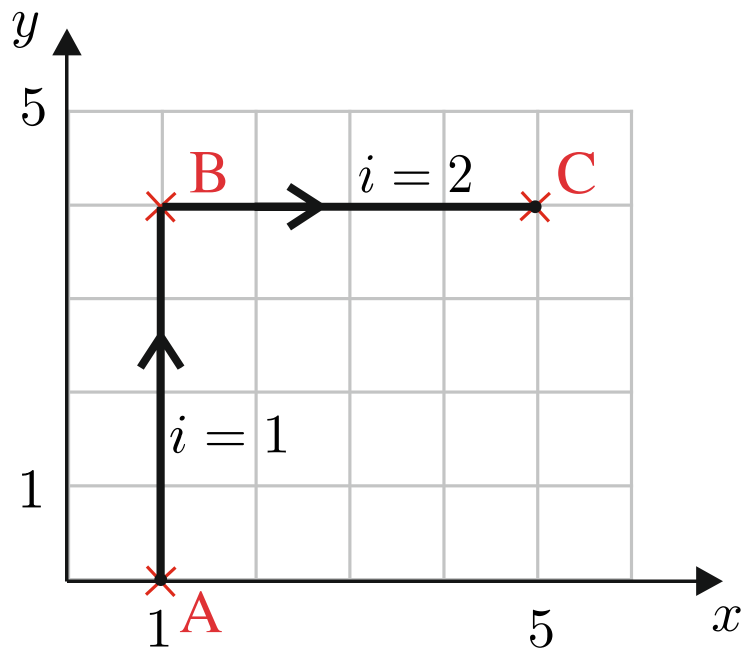
Circular track $\boldsymbol{x}_\mathrm{E}(t) = {}_{(0)}\boldsymbol{x}_\mathrm{M} + {}^0\boldsymbol{R}_\mathrm{M} \begin{pmatrix} \cos(s(t)\,\varphi_\mathrm{max}) \\ \sin(s(t)\,\varphi_\mathrm{max}) \\ 0 \end{pmatrix} \underbrace{\left\| {}_{(\mathrm{M})}\boldsymbol{x}_{\mathrm{P}_1} - {}_{(\mathrm{M})}\boldsymbol{x}_\mathrm{M} \right\|_2}_{=r}$





Motion Planning in the Task Space

Problem with successive movement segments
Exact stop ("Genauhalt") of the robot at transitions that cannot be continuously differentiated.
Example: two consecutive linear segments with different direction vectors

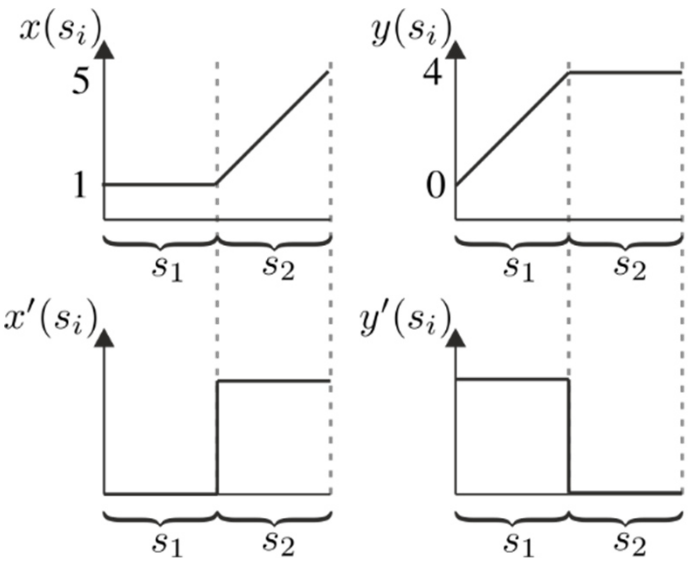


➡ Despite the continuous time profile, infinitely high acceleration $\ddot{\boldsymbol{x}}_{\mathrm{E}}$ of the end effector occurs during the sudden change in direction (at point B)

Solution
Smoothing of the path points, as exact passage is often not absolutely necessary.

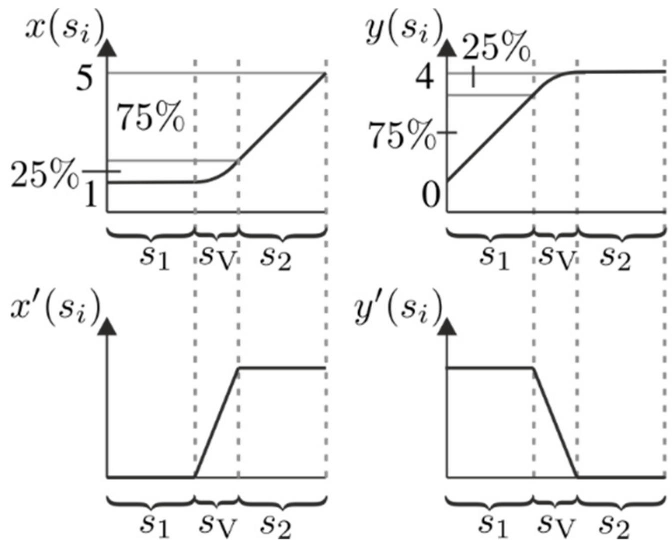
Unsmoothed

Precise stop at transitions, precise path, longer travel time



Smoothed

No exact stop, tolerance range around smoothing segment, shorter travel time

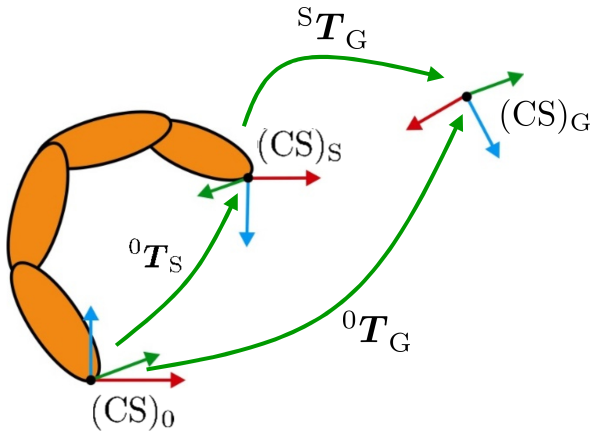


Robotics I

06. Motion Planning (Orientation Interpolation)

Orientation Interpolation

Please note for orientations:
${}^{0}\boldsymbol{R}_{\mathrm{S}}$ and ${}^{0}\boldsymbol{R}_{\mathrm{G}}$ are the rotation matrices that represent the unit vectors of $(\mathrm{CS})_{\mathrm{S}}$ and $(\mathrm{CS})_{\mathrm{G}}$ in $(\mathrm{CS})_{0}$



Simple solution: Interpolation of the unit vectors
Problem: no orthogonality of the current rotation matrix

Reasonable approaches:
Interpolation of the rotation angles of the compound rotations
Rotation around an equivalent axis with a specific angle
Interpolation using quaternions (LERP, SLERP)

Orientation Interpolation

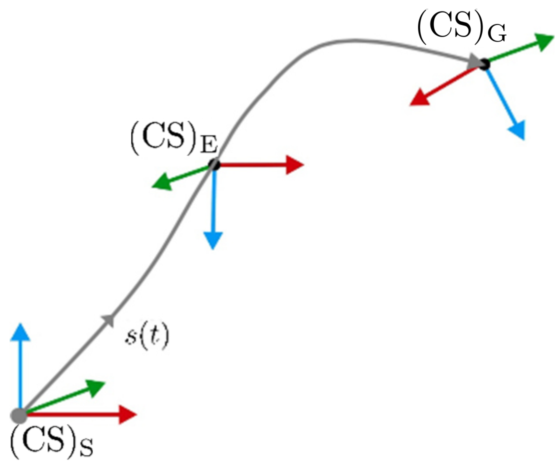
Interpolation of the rotation angles of compound rotations

Example: Cardan angle $\Phi_i = (\alpha_i,\, \beta_i,\, \gamma_i)^{\mathrm{T}}$ (see lecture 02):

$$\phi_{\mathrm{E}_i} \hat{=} \alpha_i = \operatorname{arctan2}\left(-{}^0t_{i,\,(2,3)},\, {}^0t_{i,\,(3,3)}\right),$$
$$\psi_{\mathrm{E}_i} \hat{=} \beta_i = \operatorname{arctan2}\left({}^0t_{i,\,(1,3)},\, {}^0t_{i,\,(1,1)}\, c_\gamma - {}^0t_{i,\,(1,2)}\, s_\gamma\right),$$
$$\theta_{\mathrm{E}_i} \hat{=} \gamma_i = \operatorname{arctan2}\left(-{}^0t_{i,\,(1,2)},\, {}^0t_{i,\,(1,1)}\right)$$

Using a path coordinate $s(t)$, the following applies to the rotation angle, velocity and acceleration at time t:

$$\Phi(t) = \Phi_{\mathrm{S}} + s(t)\ (\Phi_{\mathrm{G}} - \Phi_{\mathrm{S}})$$
$$\dot{\Phi}(t) = \dot{s}(t)\ (\Phi_{\mathrm{G}} - \Phi_{\mathrm{S}})$$
$$\ddot{\Phi}(t) = \ddot{s}(t)\ (\Phi_{\mathrm{G}} - \Phi_{\mathrm{S}})$$



further details in L. Sciavicco et. al. - Modeling and Control of Robot Manipulators, p. 192 ff

Orientation Interpolation

Rotation around an equivalent axis with a specific angle

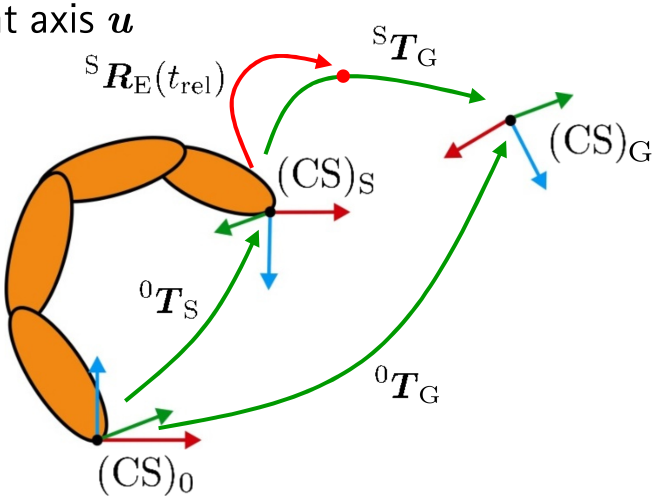
The transformation matrix between the start and the goal:

$$^{\mathrm{S}}\boldsymbol{T}_{\mathrm{G}} = \left({}^{0}\boldsymbol{T}_{\mathrm{S}}\right)^{-1}\, {}^{0}\boldsymbol{T}_{\mathrm{G}} = {}^{\mathrm{S}}\boldsymbol{T}_{0}\, {}^{0}\boldsymbol{T}_{\mathrm{G}} \quad \Rightarrow \quad {}^{\mathrm{S}}\boldsymbol{R}_{\mathrm{G}}$$

The rotation ${}^{\mathrm{S}}\boldsymbol{R}_{\mathrm{G}}$ can be interpreted as a rotation around a constant axis $\boldsymbol{u}$ with the rotation angle θ (see chapter 2):

$$\boldsymbol{u} = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} = \frac{1}{2\,\sin(\theta)} \begin{pmatrix} {}^{\mathrm{S}}r_{\mathrm{G},(3,2)} - {}^{\mathrm{S}}r_{\mathrm{G},(2,3)} \\ {}^{\mathrm{S}}r_{\mathrm{G},(1,3)} - {}^{\mathrm{S}}r_{\mathrm{G},(3,1)} \\ {}^{\mathrm{S}}r_{\mathrm{G},(2,1)} - {}^{\mathrm{S}}r_{\mathrm{G},(1,2)} \end{pmatrix}$$

$$\theta = \arccos\left(\frac{{}^{\mathrm{S}}r_{\mathrm{G},(1,1)} + {}^{\mathrm{S}}r_{\mathrm{G},(2,2)} + {}^{\mathrm{S}}r_{\mathrm{G},(3,3)} - 1}{2} \right)$$



Details in L. Sciavicco et. al. - Modeling and Control of Robot Manipulators, p. 28 ff. and p. 193

Orientation Interpolation

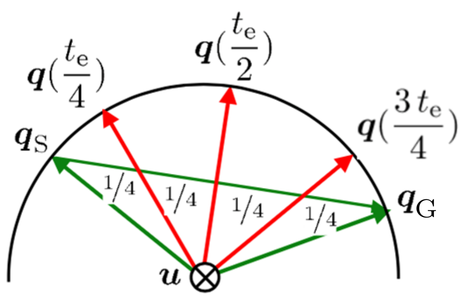
Interpolation of quaternions

Linear quaternion interpolation (LERP):

For $0 \leq t_{\mathrm{rel}} \leq 1$: Shortest connection (straight line) through the four-dimensional unit sphere (from quaternions $\boldsymbol{q}_{\mathrm{S}}$ to $\boldsymbol{q}_{\mathrm{G}}$)

Result: Rotation at variable velocity around a fixed axis of rotation $\boldsymbol{u}$:

$$\mathrm{Lerp}(\boldsymbol{q}_{\mathrm{S}},\, \boldsymbol{q}_{\mathrm{G}},\, t_{\mathrm{rel}}) = \boldsymbol{q}_{\mathrm{S}} + t_{\mathrm{rel}}\ (\boldsymbol{q}_{\mathrm{G}} - \boldsymbol{q}_{\mathrm{S}})$$



Projected interpolation curve of the four-dimensional unit sphere in the case of LERP

Attention

Interpolation by the unit sphere results in a variable rotational velocity (larger velocity in the center).

➡ undesirable, alternative approach necessary

Orientation Interpolation

Interpolation of quaternions

Spherical linear interpolation (SLERP):

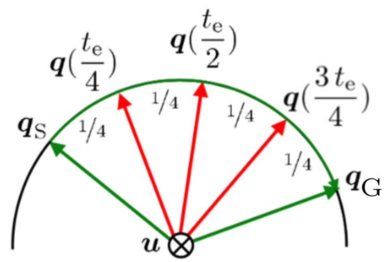
For $0 \leq t_\mathrm{rel} \leq 1$: Shortest connection **on** the four-dimensional unit sphere.

Result: Rotation at **constant** velocity around a fixed axis of rotation $\boldsymbol{u}$. There are four equivalent approaches to interpolation:

$$\begin{aligned}\mathrm{Slerp}(\boldsymbol{q}_\mathrm{S},\, \boldsymbol{q}_\mathrm{G},\, t_\mathrm{rel}) &= \boldsymbol{q}_\mathrm{S} \left(\boldsymbol{q}_\mathrm{S}^{-1}\, \boldsymbol{q}_\mathrm{G}\right)^{t_\mathrm{rel}} = \left(\boldsymbol{q}_\mathrm{S} \boldsymbol{q}_\mathrm{G}^{-1}\right)^{1-t_\mathrm{rel}} \boldsymbol{q}_\mathrm{G} \\ &= \left(\boldsymbol{q}_\mathrm{G} \boldsymbol{q}_\mathrm{S}^{-1}\right)^{t_\mathrm{rel}} \boldsymbol{q}_\mathrm{S} = \boldsymbol{q}_\mathrm{G} \left(\boldsymbol{q}_\mathrm{G}^{-1}\, \boldsymbol{q}_\mathrm{S}\right)^{1-t_\mathrm{rel}},\end{aligned}$$

$$\boldsymbol{q}^k = \mathrm{e}^{k \log(\boldsymbol{q})} = \left(\cos\left(k\,\frac{\theta}{2}\right),\, \boldsymbol{u}^\mathrm{T} \sin\left(k\,\frac{\theta}{2}\right)\right)^\mathrm{T}$$

$$\log(\boldsymbol{q}) = \left(0,\, \frac{\theta}{2}\,\boldsymbol{u}^\mathrm{T}\right)^\mathrm{T} \qquad \left(\boldsymbol{q} = (s,\, \boldsymbol{v}^\mathrm{T})^\mathrm{T}\right)$$



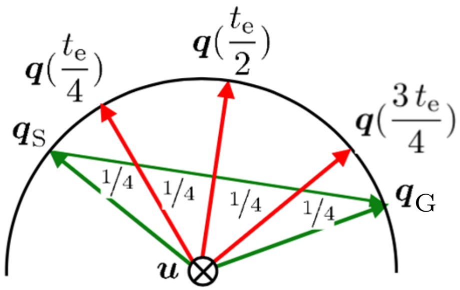
Projected interpolation curve of the four-dimensional unit sphere in the case of SLERP

Details in E. B. Dam - Quaternions, Interpolation and Animation, pp. 40-48, D. Eberly - Quaternion Algebra and Calculus

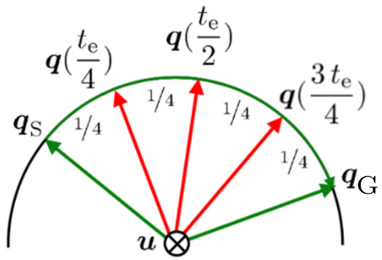
Orientation Interpolation

Interpolation of quaternions

- Low computation effort
- Simple implementation without additional calculations
- Separate interpolation of position and orientation required



Projected interpolation curve of the four-dimensional unit sphere in the case of LERP

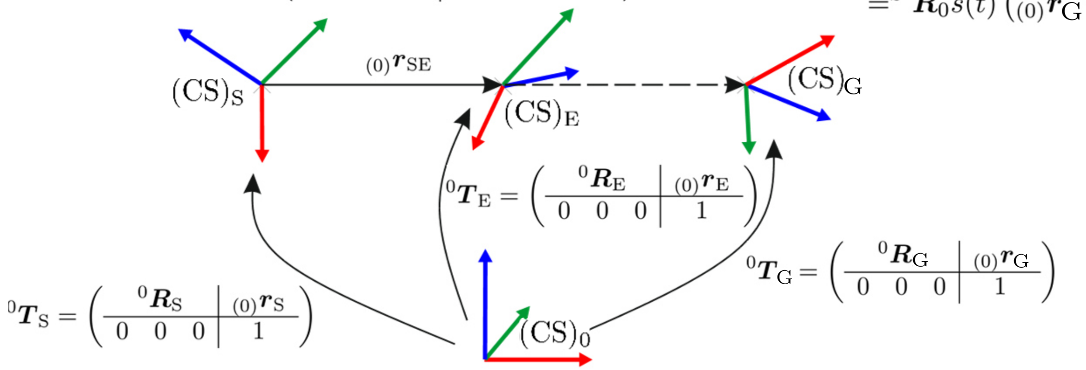


Projected interpolation curve of the four-dimensional unit sphere in the case of SLERP

Holistic Motion Planning in the Task Space

Combination of the methods presented (translation and rotation)
Example: straightforward translation with constant change of orientation

$$ {}^{\mathrm{S}}\boldsymbol{T}_{\mathrm{E}}(t_{\mathrm{rel}}) = \left(\begin{array}{c|c} {}^{\mathrm{S}}\boldsymbol{R}_{\mathrm{E}}(t_{\mathrm{rel}}) & {}_{(\mathrm{S})}\boldsymbol{r}_{\mathrm{SE}}\left(s(t)\right) \\ \hline 0 \quad 0 \quad 0 & 1 \end{array} \right) \qquad {}_{(\mathrm{S})}\boldsymbol{r}_{\mathrm{SE}}(s(t)) = {}^{\mathrm{S}}\boldsymbol{R}_{0\,(0)}\boldsymbol{r}_{\mathrm{SE}} = {}^{\mathrm{S}}\boldsymbol{R}_{0}\, s(t) \left({}_{(0)}\boldsymbol{r}_{\mathrm{G}} - {}_{(0)}\boldsymbol{r}_{\mathrm{S}} \right) $$



- Interpolation of orientation through ${}^{\mathrm{S}}\boldsymbol{R}_{\mathrm{E}}(t_{\mathrm{rel}})$
- Translational interpolation with ${}_{(0)}\boldsymbol{r}_{\mathrm{SE}} = s(t)\left({}_{(0)}\boldsymbol{r}_{\mathrm{G}} - {}_{(0)}\boldsymbol{r}_{\mathrm{S}}\right)$

Attention: Different path coordinates must be taken into account to comply with the specifications of rotational and translational velocity (& acceleration)

Questions for Self-Monitoring

Motion Planning
How do I design a continuous (and efficient) motion sequence for my robot that fulfills certain constraints?

1. Which variants of robot paths are you familiar with? Describe the variants and name the main differences!
2. Qualitatively draw a typical trapezoidal velocity profile and a time-optimal trapezoidal velocity profile in which the maximum velocity is not reached! What is this time-optimal case called? What can be said about the continuity of such a profile?
3. Explain the specific procedure for interpolating the rotation angles of composed rotations and name alternative approaches for the interpolation of orientations!