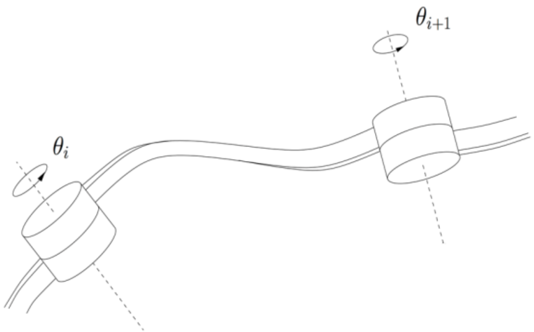
Karlsruher Institut für Technologie

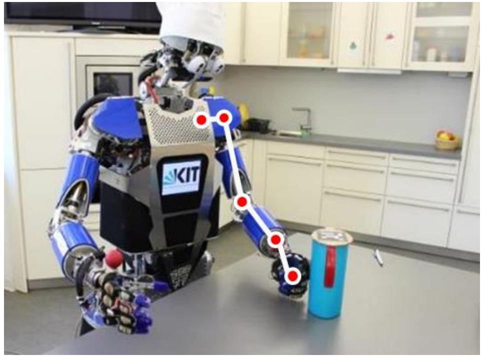
Hochperformante Humanoide Technologien

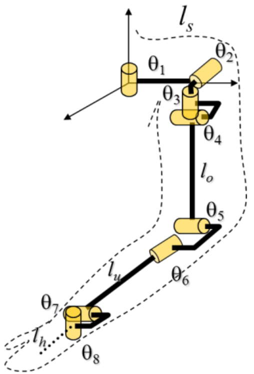
Robotik I: Einführung in die Robotik
Kapitel 1 – Mathematische Grundlagen

Tamim Asfour

http://www.humanoids.kit.edu







Motivation

ARMAR!
Bring me the apple juice from the fridge

Motivation

- Welche mathematischen Methoden werden benötigt?
- Wir müssen Positionen im Raum beschreiben:
  - Wo ist der Apfelsaft? (an welcher Koordinate?)
  - Relativ zu welchem Koordinatensystem?
    - Relativ zum Kamera-Koordinatensystem?
    - Relativ zum Arm/Schulter-Basiskoordinatensystem?
    - Relativ zum Basiskoordinatensystem der mobilen Plattform? (zwei Meter vor dem Roboter?)
    - Relativ zum Weltkoordinatensystem? (befindet sich ganz links in der Ecke der Küche)
- Wir müssen Orientierungen im Raum beschreiben:
  - Befindet sich die Flasche direkt vor dem Roboter?
  - Oder muss sich der Roboter nach links drehen um die Flasche zu sehen?
- **Wir benötigen ein mathematisches Rahmenwerk zum Beschreiben von Positionen (Translationen) und Orientierungen (Rotationen)**

Kinematische Grundlagen

- Dieses Kapitel ist eine Einführung in die mathematischen Grundlagen der Robotik

- Mathematische Methoden zur Beschreibung von Starrkörpertransformationen (basierend auf Linearer Algebra)

- Anwendung dieser Methoden, um Roboter zu modellieren

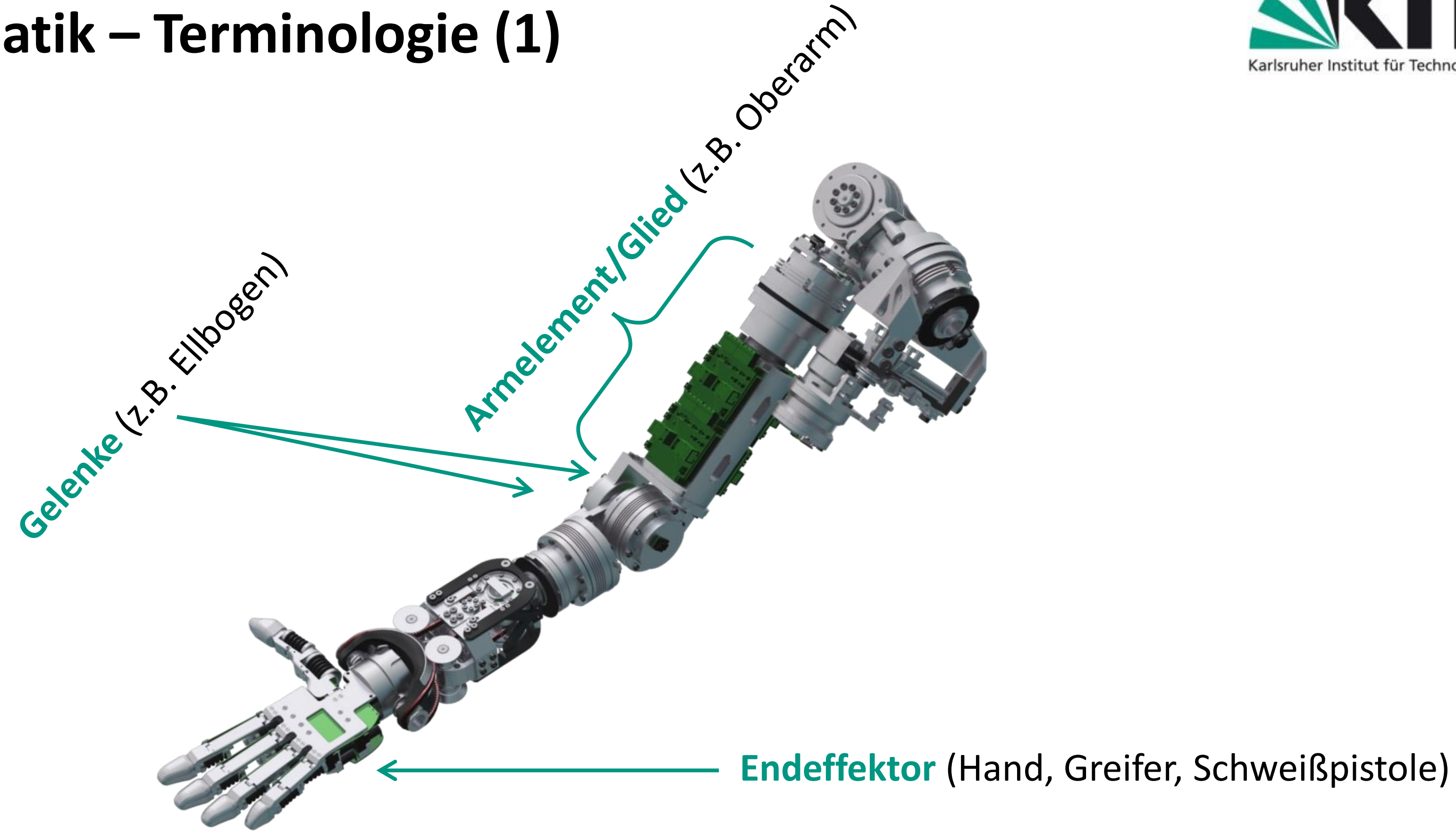
Definitionen

- **Kinematik** ist die reine geometrische Beschreibung von **Bewegung** eines Mechanismus bzw. eines Roboters. Das essentielle Konzept ist die **Position**.

- **Statik** behandelt **Kräfte und Momente**, die sich auf einen **ruhenden** Mechanismus auswirken. Das essentielle Konzept ist die **Steifigkeit**.

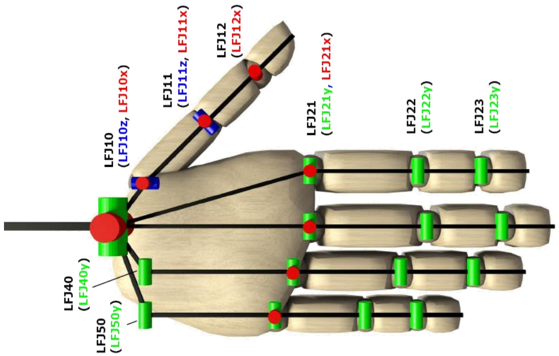
- **Dynamik** analysiert die **Kräfte und Momente**, die durch **Bewegung und Beschleunigung** eines Mechanismus und einer zusätzlichen Last entstehen.

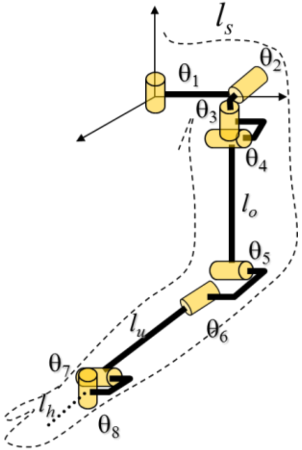
Kinematik – Terminologie (1)

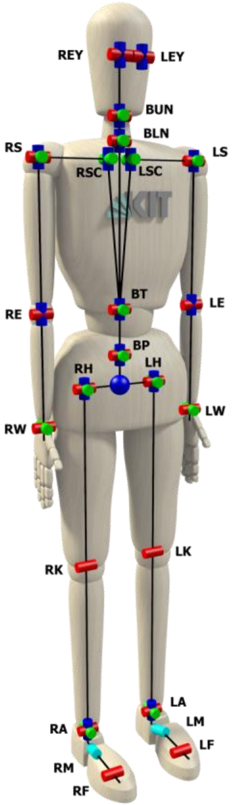


Kinematik – Terminologie (2)

- **Kinematische Kette** ist eine Menge an Gliedern, die durch Gelenke verbunden sind.
- Kinematische Ketten können durch einen Graph repräsentiert werden. Kanten repräsentieren Glieder, Knoten repräsentieren Gelenke.

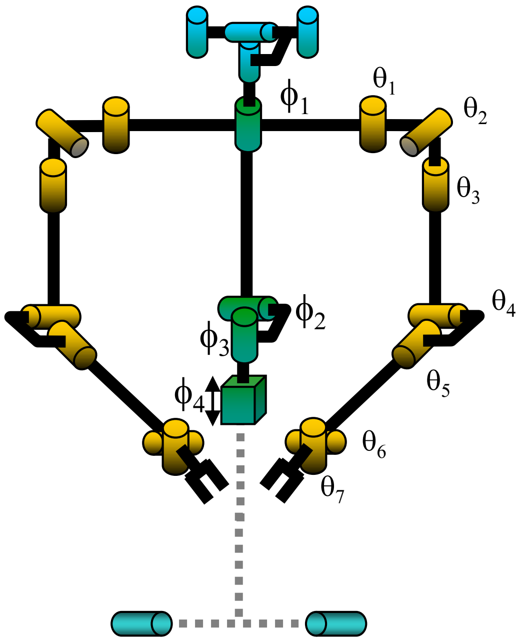


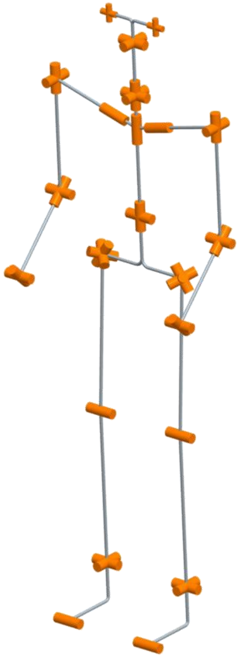




Kinematik – Terminologie (3)

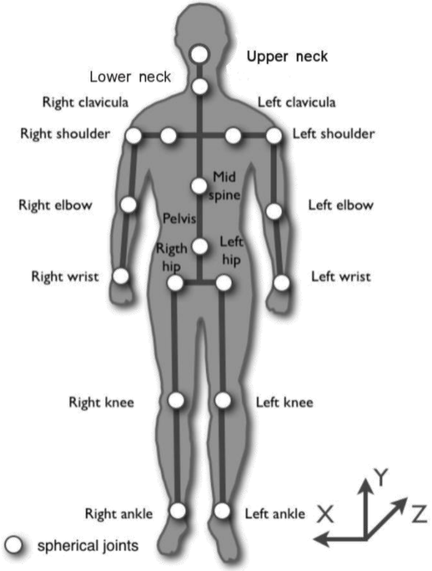
- Kinematische Ketten: weitere Beispiele

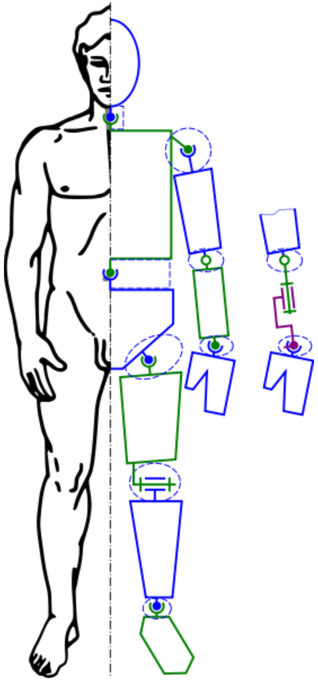


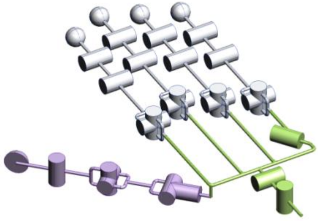


Kinematik – Terminologie (4)

- Kinematische Ketten: weitere Beispiele







Kinematik – Freiheitsgrade

Englisch: **Degrees of Freedom (DoF)**

Freiheitsgrade (weniger formale Definition) ist die Anzahl unabhängiger Parameter, die zur kompletten Spezifikation **der Lage (Position und Orientierung)** eines Mechanismus/Objekts benötigt werden.

Beispiele:

- Ein Punkt auf einer Ebene hat 2 DoF
- Ein Punkt im 3D Raum hat 3 DoF
- Ein Starrkörper in 2D Raum, z.B. auf einer Ebene hat 3 DoF
- Ein Starrkörper im 3D Raum hat 6 DoF

Konventionen

In dieser Vorlesung, werden für Symbole folgende Konventionen verwendet:

- Skalare: kleingeschriebene lateinische Buchstaben
  Beispiel: $s, t \in \mathbb{R}$

- Vektoren: fette kleingeschriebene lateinische Buchstaben
  Beispiel: $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^3$

- Matrizen: großgeschriebene lateinische Buchstaben
  Beispiel: $A \in \mathbb{R}^{3\times 3}$

- Lineare Abbildungen (Transformationen): großgeschriebene griechische Buchstaben
  Beispiel: $\phi(\cdot): \mathbb{R}^3 \rightarrow \mathbb{R}^3$

Starrkörperbewegung

Starrkörperbewegungen werden durch **zwei Eigenschaften** charakterisiert:

1. Die Distanz zweier beliebiger Punkte ist konstant

2. Die Orientierungen (im Körper) bleiben erhalten.
   (Ein rechtsdrehendes Koordinatensystem bleibt rechtsdrehend)

SO(3) und SE(3)

Die folgenden zwei Gruppen sind in der Robotik von besonderem Interesse:

- **SO(3) - die Spezielle Orthogonale Gruppe**, die **Rotationen** repräsentiert
- **SE(3) - die Spezielle Euklidische Gruppe**, die **Transformationen** repräsentiert
- **Elemente aus SO(3)** werden als reale 3×3 orthogonale Matrizen R (Zeilen- und Spaltenvektoren orthonormal) beschrieben und erfüllen

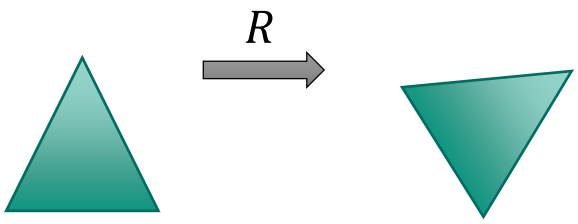
$$R^{\top} R = I \qquad \text{mit} \qquad \det(R) = 1$$

- **Elemente aus $\mathbf{SE(3)}$** sind von der Form $(\mathbf{p}, R)$, mit $\mathbf{p} \in \mathbb{R}^3$ und $R \in \mathrm{SO}(3)$

SO(3) und SE(3)

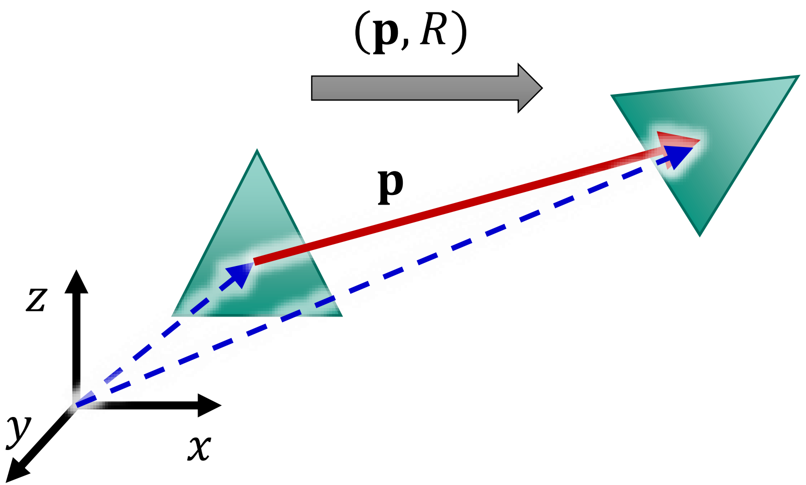
SO(3)

- Rotation (Orientierung)
- $R \in SO(3) \subset \mathbb{R}^{3\times3}$



SE(3)

- Translation und Rotation (Position und Orientierung)
- $(\mathbf{p}, R) \in SE(3)$ mit $\mathbf{p} \in \mathbb{R}^3, R \in SO(3)$



Affine Geometrie

- Beschreibung von räumlichen Transformationen
- Diese räumlichen Transformationen bestehen aus Verknüpfungen von **Rotationen** und **Translationen**

- Unterschiedliche Möglichkeiten zur mathematischen Beschreibung räumlicher Transformationen:
  - Rotationsmatrizen und Translationsvektoren
  - Homogene Matrizen
  - Quaternionen
  - Duale Quaternionen
  - …

- Diese Vorlesung wird die üblichsten Repräsentation einführen

Euklidischer Raum (1)

- Der euklidische Raum ist der **Vektorraum** $\mathbb{R}^3$ mit dem **Skalarprodukt** (auch: inneres Produkt, Punktprodukt)

- Beispiel:

  Ein Punkt $\mathbf{c}$, der auf einer Linie zwischen den zwei Punkten $\mathbf{a}$ und $\mathbf{b}$ liegt, kann folgendermaßen repräsentiert werden

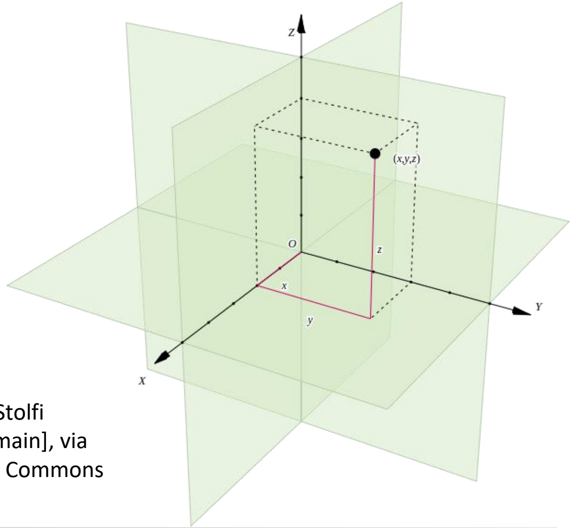
$$\mathbf{c} = t \cdot \mathbf{a} + (1 - t) \cdot \mathbf{b}, \quad t \in [0, 1] \subset \mathbb{R}, \quad \mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^3$$

Euklidischer Raum (2)

- Ein **Punkt a** im euklidischen Raum wird durch Koordinaten repräsentiert, die sich auf ein **Koordinatensystem** $\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$ beziehen:

$$\mathbf{a} = (a_x, a_y, a_z)^\top = a_x \cdot \mathbf{e}_x + a_y \cdot \mathbf{e}_y + a_z \cdot \mathbf{e}_z \qquad \left(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z \in \mathbb{R}^3\right)$$

- Konventionen:
  - Wir benutzen **orthonormale Koordinatensysteme**. Die Basisvektoren $\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$ sind zueinander **orthogonale** (senkrecht) stehende **Einheitsvektoren.**
  - Wir benutzen **rechtsdrehende Koordinatensysteme**.



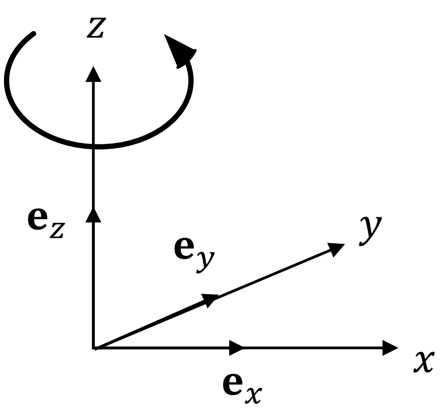
von Jorge Stolfi
[Public domain], via
Wikimedia Commons

Koordinatensysteme (1)

Rechtsdrehendes Koordinatensystem

$$\mathbf{e}_x \times \mathbf{e}_y = \mathbf{e}_z$$

$$\mathbf{x} \times \mathbf{y} = \mathbf{z}$$

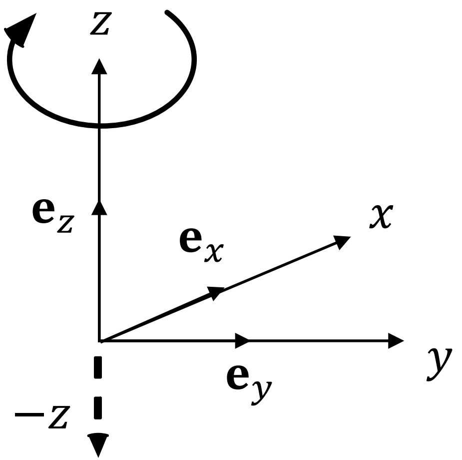


Linksdrehendes Koordinatensystem

$$\mathbf{e}_x \times \mathbf{e}_y = -\mathbf{e}_z$$

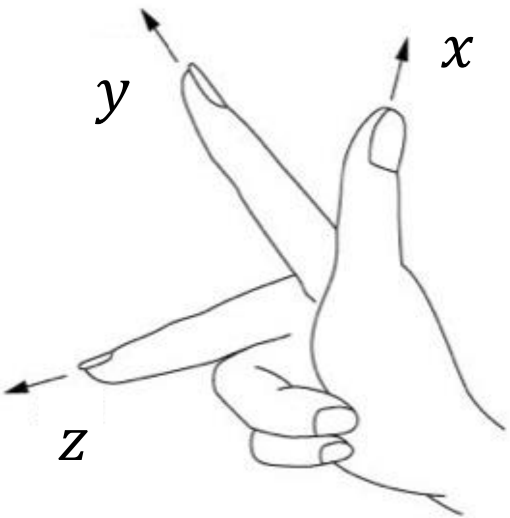
$$\mathbf{x} \times \mathbf{y} = -\mathbf{z}$$

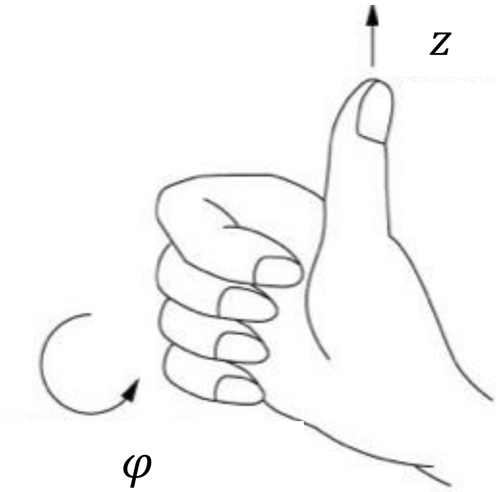
$\times$: Kreuzprodukt



Koordinatensysteme (2)

Rechte-Hand-Regel für rechtsdrehende Koordinatensysteme





Lineare Abbildungen: Endomorphismen

- **Lineare Abbildungen (Transformationen)**, die den euklidischen Raum auf sich selbst abbilden, nennt man **Endomorphismen**:

$$\phi(\cdot) \colon \mathbb{R}^3 \rightarrow \mathbb{R}^3$$

- Endomorphismen können durch **quadratische Matrizen** repräsentiert werden:

$$\phi(\mathbf{a}) = A \cdot \mathbf{a}, \quad A \in \mathbb{R}^{3\times 3}$$

- A beschreibt einen **Basiswechsel** zwischen den originalen Basisvektoren $\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$ und den neuen Basisvektoren $\mathbf{e}'_{\boldsymbol{x}}, \mathbf{e}'_{\boldsymbol{y}}, \mathbf{e}'_{\boldsymbol{z}}$:

$$A = \begin{pmatrix} \mathbf{e}'_{\boldsymbol{x}} & \mathbf{e}'_{\boldsymbol{y}} & \mathbf{e}'_{\boldsymbol{z}} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{e}_x & \mathbf{e}_y & \mathbf{e}_z \end{pmatrix}^{-1}$$

Lineare Abbildungen: Isomorphismen

- **Bijektive** (umkehrbare) Endomorphismen nennt man **Isomorphismen**.

- Isomorphismen können spezielle und interessante Eigenschaften besitzen:
    1. Winkel bleiben erhalten. (Beispiel: Skalierung und Rotation)
    2. Längen bleiben erhalten. (Beispiel: Rotation)
    3. Händigkeit beleibt erhalten. (Beispiel: Rotation. Rechtshändiges Koordinatensystem bleibt erhalten, usw.)

- Eine spezielle Art von Isomorphismen, die alle genannten Kriterien erfüllt, ist die **Rotationsgruppe** (oder spezielle orthogonale Gruppe) $SO(3)$.

Die Rotationsgruppe $\mathbf{SO}(3)$

- $\mathrm{SO}(3)$ beinhaltet **alle möglichen Rotationen** um willkürlich durch den Ursprung verlaufende Rotationsachsen.
- $\mathrm{SO}(3)$ ist **nicht kommutativ** (nicht abelsch), z.B.

$$A \cdot B \cdot \mathbf{x} \neq B \cdot A \cdot \mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^3, \quad A, B \in \mathrm{SO}(3)$$

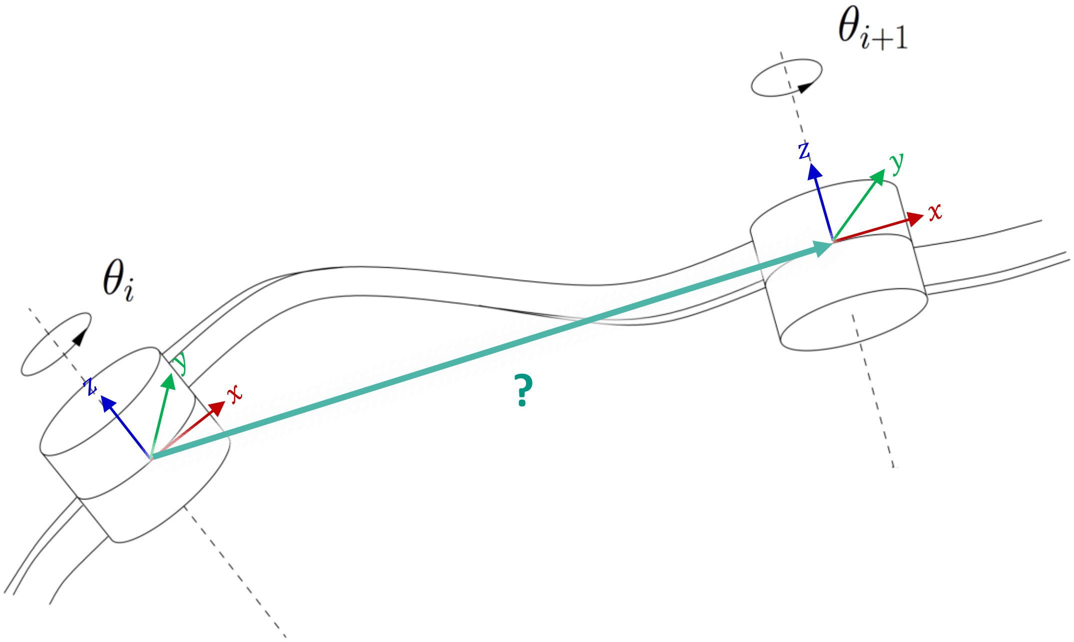
Warum sind $\mathbf{SO}(3)$ und $\mathbf{SE}(3)$ interessant für die Robotik?

- Mit Hilfe von $\mathrm{SO}(3)$ und $\mathrm{SE}(3)$, kann eine **Objektpose** (Position und Orientierung) im Raum sowie **Transformationen zwischen zwei Robotergelenkachsen** als **Verkettung einer Translation und einer Rotation** repräsentiert werden:

$$\varphi(\cdot)\colon \mathbb{R}^3 \to \mathbb{R}^3, \quad \varphi(\mathbf{x}) = R \cdot \mathbf{x} + \mathbf{t}, \quad \mathbf{x}, \mathbf{t} \in \mathbb{R}^3, \quad R \in \mathrm{SO}(3)$$

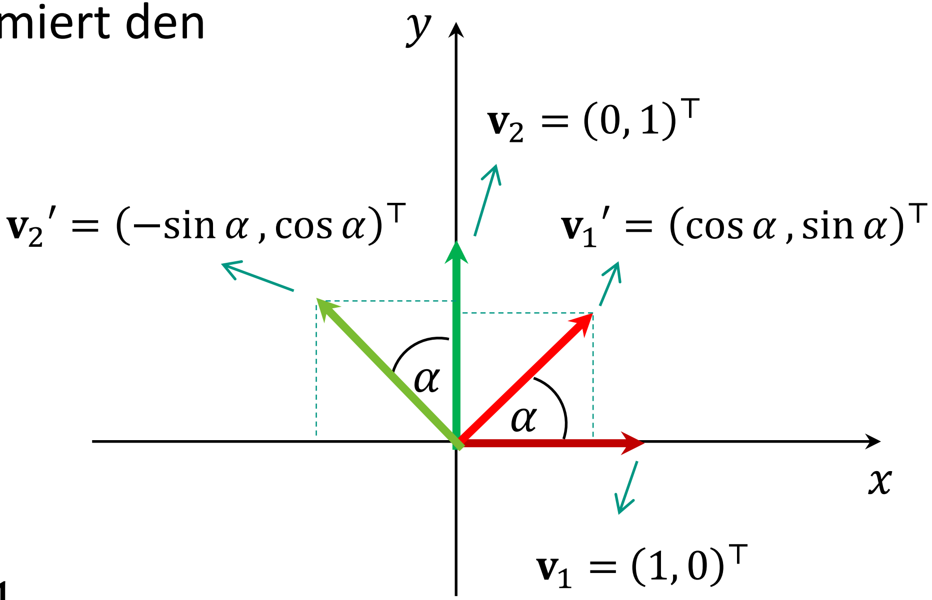
- Die Abbildung $\varphi(\cdot)$ ist nicht linear (wegen der Addition)! Sie wird **affin** genannt.

Transformation zwischen zwei Robotergelenkachsen



Rotationen in 2D

- Eine **Rotation in der xy-Ebene** um $(0,0)$ ist eine **lineare Transformation**
- Eine Rotation von α um $(0,0)$ transformiert den
  - Vektor $(1,0)^\top$ zu $(\cos\alpha, \sin\alpha)^\top$
  - Vektor $(0,1)^\top$ zu $(-\sin\alpha, \cos\alpha)^\top$



- **Rotationsmatrix**

$$R_\alpha(x) = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \cdot \mathbf{x}$$

mit $RR^\top = R^\top R = I, \;\; \det(R) = 1$

Rotationen in 2D

- Eine **Rotation** um einen Punkt $\mathbf{c} \neq (0,0)$ ist **keine lineare Transformation**.
  Sie transformiert den Ursprung $(0,0)$ in einen anderen Punkt als $(0,0)$.

- **Vorgehen** bei der Rotation $R_{\mathbf{c},\alpha}$ um ein beliebiges Rotationszentrum $\mathbf{c}$:
  - Wir **verschieben die Ebene um** $-\mathbf{c}$, so dass das Rotationszentrum in $(0,0)$ liegt.
  - Dann führen wir eine **Rotation** R_α **um** $(0,0)$ durch.
  - Danach verschieben wir die Ebene **um** $+\mathbf{c}$ **zurück**.

$$
\begin{aligned}
R_{\mathbf{c},\alpha}(\mathbf{x}) &= R_\alpha(\mathbf{x} - \mathbf{c}) + \mathbf{c} \\
&= R_\alpha(\mathbf{x}) + (-R_\alpha(\mathbf{c}) + \mathbf{c})
\end{aligned}
$$

Affine Transformation

$$R_{\mathbf{c},\alpha}(\mathbf{x}) = R_\alpha(\mathbf{x} - \mathbf{c}) + \mathbf{c} = R_\alpha(\mathbf{x}) + (-R_\alpha(\mathbf{c}) + \mathbf{c})$$

- $R_{\mathbf{c},\alpha}$ ist eine nichtlineare Transformation. Sie unterscheidet sich von R_α nur durch das Addieren einer Konstante (unabhängig von $\mathbf{x}$).

- Transformationen (wie $R_{\mathbf{c},\alpha}$) der Form

$$\boxed{T(\mathbf{x}) = A \cdot \mathbf{x} + \mathbf{b}}$$

werden **affine Transformation** (oder auch affine Abbildung) genannt.

Rotationen in 3D

- Eine 2D **Rotation in der xy-Ebene** ist eine 3D **Rotation um die z-Achse**.

- Eine **Rotation** von Punkten um die z-Achse **hängt nicht von ihren z-Werten ab.** Punkte auf der z-Achse werden durch solch eine Rotation nicht betroffen.

- Die Rotationsmatrix um die z-Achse besitzt eine einfache Form:
  - Die zu xy zugehörige **Teilmatrix ist identisch zu dem 2D Fall** und
  - die Einträge zum Einfluss von z auf x sowie y und umgekehrt sind jeweils 0.

$$R_\alpha = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \quad \Rightarrow \quad R_{\mathbf{z},\alpha} = \underbrace{\begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{Rotationsmatrix um die } z\text{-Achse}}$$

Rotationen in 3D

Karlsruher Institut für Technologie

- Analog für Rotation um x-Achse und y-Achse

$$R_{\mathbf{z},\alpha} = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R_{\mathbf{x},\alpha} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix}$$

$$R_{\mathbf{y},\alpha} = \begin{pmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{pmatrix}$$

Inverse Rotationsmatrix

- Erinnerung: Für alle Matrizen $R \in SO(3)$ gilt: $R \cdot R^{\top} = I$
- Das heißt: Die inverse Rotationsmatrix R^{-1} entspricht der **transponierten** Rotationsmatrix $R^{\top}$
- Beispiel:

$$R_{\mathbf{x},\theta}^{-1} = R_{\mathbf{x},-\theta} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(-\theta) & -\sin(-\theta) \\ 0 & \sin(-\theta) & \cos(-\theta) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix} = R_{\mathbf{x},\theta}^{\top}$$

- Die Inverse einer Rotationsmatrix kann also leicht berechnet werden.

Verkettung von Rotationen

- Rotationen können verkettet werden:

$$\phi_{\mathbf{z},\gamma}\left(\phi_{\mathbf{y},\beta}\left(\phi_{\mathbf{x},\alpha}(\mathbf{a})\right)\right), \quad \mathbf{a} \in \mathbb{R}^3$$

- **Wichtig:** Es gibt **zwei Möglichkeiten**, die obere Verkettung als **Matrixprodukt** zu *interpretieren*:
  - Von **rechts nach links**: Rotationen werden um **globale Achsen** durchgeführt, die sich nicht ändern.

$$R_{\mathbf{z},\gamma} \cdot \left(R_{\mathbf{y},\beta} \cdot \left(R_{\mathbf{x},\alpha} \cdot \mathbf{a}\right)\right) = R_{\mathbf{z},\gamma} \cdot R_{\mathbf{y},\beta} \cdot R_{\mathbf{x},\alpha} \cdot \mathbf{a}$$

  - Von **links nach rechts**: Mit jeder Rotation ändern sich die Einheitsvektoren. Rotationen werden um **lokale Achsen** durchgeführt.

$$\left(\left(R_{\mathbf{z},\gamma} \cdot R_{\mathbf{y}',\beta}\right) \cdot R_{\mathbf{x}'',\alpha}\right) \cdot \mathbf{a} = R_{\mathbf{z},\gamma} \cdot R_{\mathbf{y}',\beta} \cdot R_{\mathbf{x}'',\alpha} \cdot \mathbf{a}$$

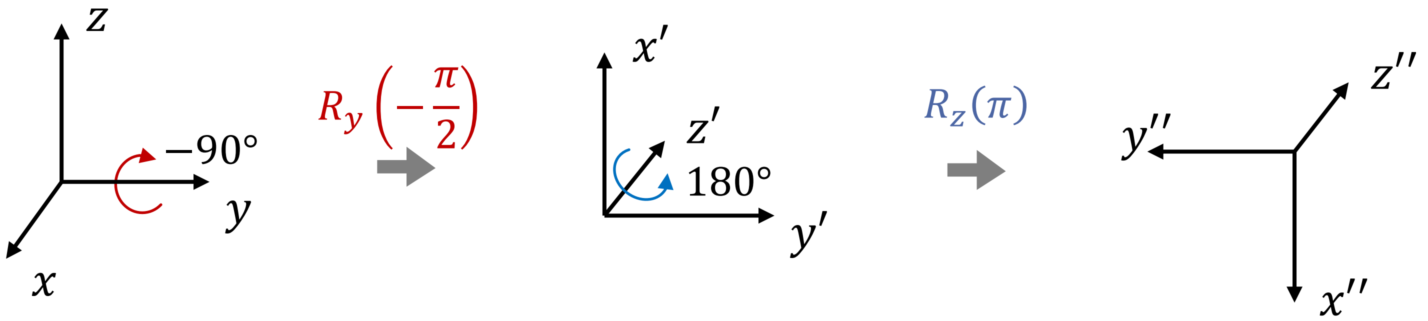
Beispiel: Verkettung von Rotationen (1)

- Verkettung der folgenden Rotationen:
  - Rotation um y-Achse: $-90° \left(-\frac{\pi}{2}\right)$

$$R_y\left(-\frac{\pi}{2}\right) = \begin{pmatrix} \cos\left(-\frac{\pi}{2}\right) & 0 & \sin\left(-\frac{\pi}{2}\right) \\ 0 & 1 & 0 \\ -\sin\left(-\frac{\pi}{2}\right) & 0 & \cos\left(-\frac{\pi}{2}\right) \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

  - Rotation um z-Achse: $180° \ (\pi)$

$$R_z(\pi) = \begin{pmatrix} \cos(\pi) & -\sin(\pi) & 0 \\ \sin(\pi) & \cos(\pi) & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



Beispiel: Verkettung von Rotationen (2)

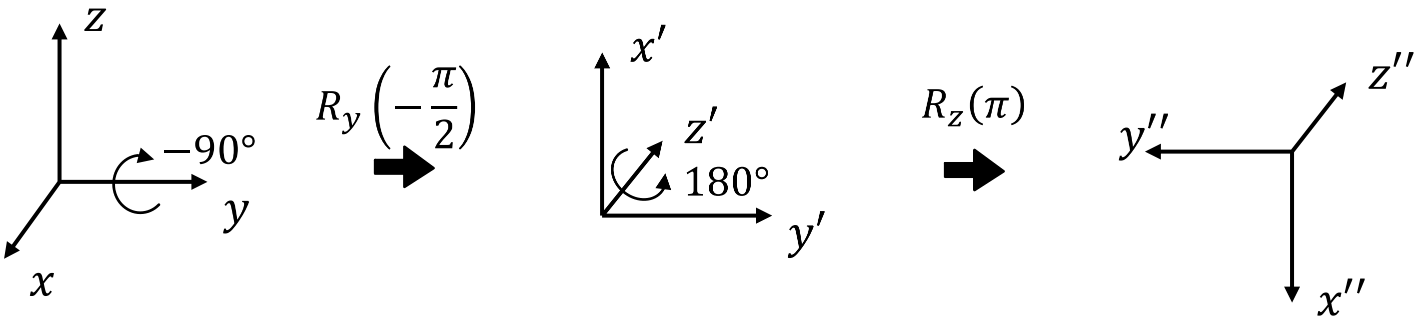
- Berechnung der Rotationsmatrix

$$R = R_y\left(-\frac{\pi}{2}\right) \cdot R_z(\pi) = \begin{pmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

- Transformation eines Vektors

$$\mathbf{p}'' = \begin{pmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{pmatrix} \cdot \mathbf{p} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} -p_3 \\ -p_2 \\ -p_1 \end{pmatrix}$$

Von **links nach rechts**:
Mit jeder Rotation ändern sich die Einheitsvektoren.
Rotationen um **lokale Achsen**



Probleme mit Rotationsmatrizen

- Rotationsmatrizen haben mehrere Nachteile:
  - **Redundanz**: Neun Werte für eine Rotationsmatrix
  - Im Bereich des **maschinellen Lernens**: Wenn die Einträge einer Rotationsmatrix unabhängig voneinander prädiziert werden, ist es wahrscheinlich, dass die resultierende Matrix keine gültige Rotationsmatrix ist.
    → Bestimmung einer Matrix, die „nah“ an der Rotationsmatrix ist

- Wie geht man mit diesen Problemen um?
  - Nutzung anderer Darstellungen für Rotationen (Quaternionen, Eulerwinkel)
  - Orthonormalisieren der Matrix

Eulerwinkel

- Es ist möglich, jede Rotation durch **drei Rotationen** um jeweils eine Rotationsachsen darzustellen.
- Die Achsen können willkürlich gewählt werden. Aus historischen Gründen wird oft die **Euler-Konvention $\mathbf{z\,x'z''}$** verwendet.
- Die Winkel $\boldsymbol{\alpha}$**,** $\boldsymbol{\beta}$ **und** $\boldsymbol{\gamma}$ sind die Eulerwinkel. Sie beschreiben die Rotationsmatrix

$$R_{\mathbf{z},\alpha} \cdot R_{\mathbf{x}',\beta} \cdot R_{\mathbf{z}'',\gamma} =$$

$$\begin{pmatrix} \cos\alpha \cdot \cos\gamma - \sin\gamma \cdot \cos\beta \cdot \sin\alpha & -\sin\gamma \cdot \cos\alpha - \cos\gamma \cdot \cos\beta \cdot \sin\alpha & \sin\beta \cdot \sin\alpha \\ \sin\alpha \cdot \cos\gamma + \sin\gamma \cdot \cos\beta \cdot \cos\alpha & -\sin\gamma \cdot \sin\alpha + \cos\gamma \cdot \cos\beta \cdot \cos\alpha & -\sin\beta \cdot \cos\alpha \\ \sin\gamma \cdot \sin\beta & \cos\gamma \cdot \sin\beta & \cos\beta \end{pmatrix}$$

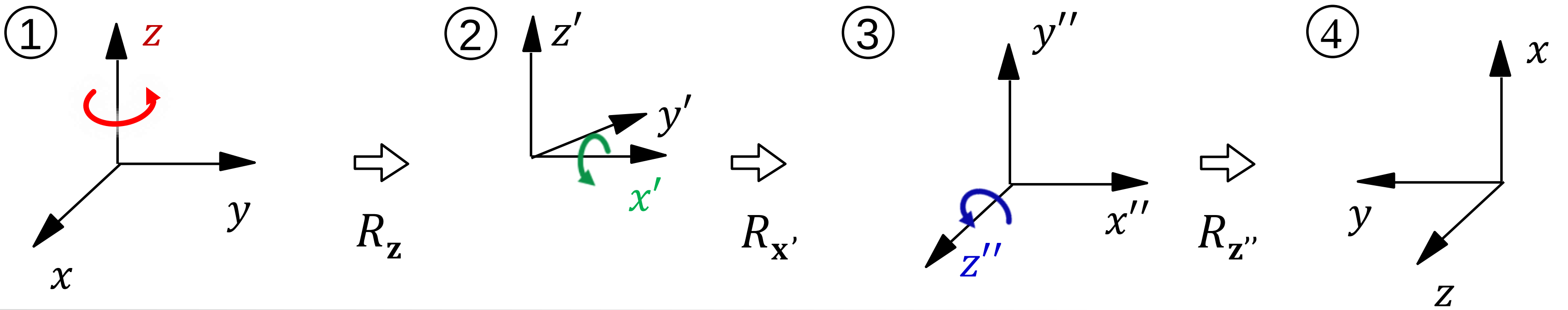
Eulerwinkel-Konvention $\mathbf{z\,x'z''}$

Abfolge der Rotationen

1. Drehung um α um die z-Achse $\mathbf{z}$ — $R_{\mathbf{z}}$
2. Drehung um β um die neue x-Achse $\mathbf{x}'$ — $R_{\mathbf{x}'}$
3. Drehung um γ um die neue z-Achse $\mathbf{z}''$ — $R_{\mathbf{z}''}$

$$R_s = R_{\mathbf{z}}\, R_{\mathbf{x}'}\, R_{\mathbf{z}''}$$

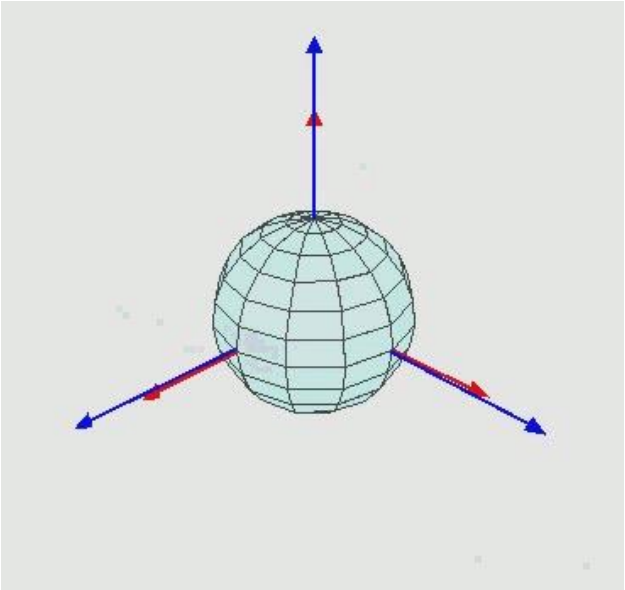
Wichtig: Drehung um jeweils veränderte Achsen!



Eulerwinkel: Konventionen

- **12 verschiedene Abfolgen** sind für die Rotationsmatrizen möglich:
  - *z-x-z*   *x-y-x*   *y-z-y*   *z-y-z*   *x-z-x*   *y-x-y*
  - *x-y-z*   *y-z-x*   *z-x-y*   *x-z-y*   *z-y-x*   *y-x-z*

- Rotationen um **lokale oder feste** Achsen
  ⇒ insgesamt **24 Möglichkeiten**

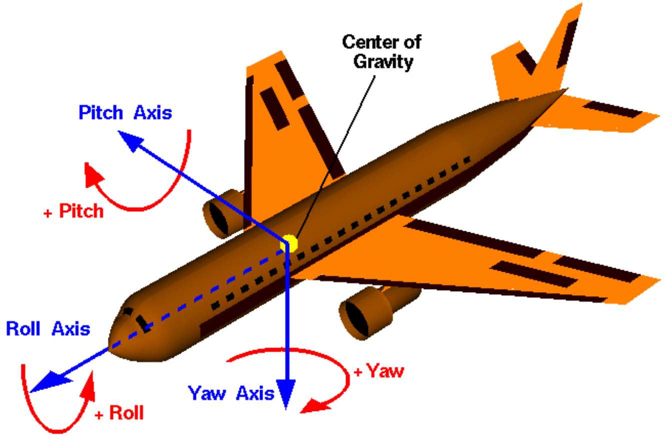


Roll, Pitch und Yaw

- Eine andere übliche Konvention ist die **Euler-Konvention $\mathbf{x}, \mathbf{y}, \mathbf{z}$**

- Diese speziellen Eulerwinkel werden **Roll, Pitch, Yaw** genannt (im Deutschen: Roll-Nick-Gier-Winkel)

- **Abfolge der Rotationen:**
  1. Globale x-Achse um α (Roll)
  2. Globale y-Achse um β (Pitch)
  3. Globale z-Achse um γ (Yaw)



von Nasa [Public domain], via ikimedia Commons

Eulerwinkel: Vor- und Nachteile

Vorteile:

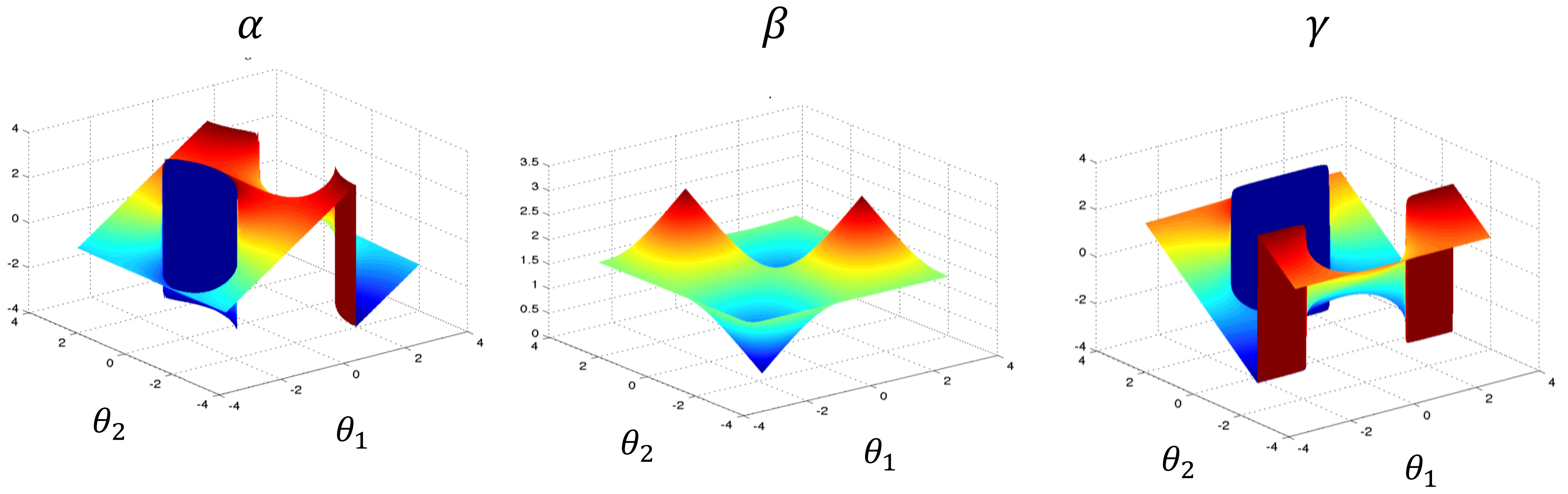
- Kompakter als Rotationsmatrizen
- Aussagekräftiger als Rotationsmatrizen

Nachteile:

- Nicht eindeutig
  - Beispiel: in der Euler-Konvention $\mathbf{x}\,\mathbf{y}'\mathbf{z}''$ beschreiben die Eulerwinkel $(45°, -90°, 45°)$ und $(30°, -90, 60°)$ die gleiche Rotation!
- Nicht kontinuierlich
  - Die Eulerwinkel einer kontinuierlichen Rotation sind nicht kontinuierlich
    - Kleine Änderung in der Orientierung können zu großen Änderungen der Eulerwinkel führen
    - Konsequenz: Eine stetige Interpolation zwischen zwei Eulerwinkeln ist nicht möglich
  - Gimbal Effekt (Gimbal Lock)

Eulerwinkel: Interpolationsproblem

Eulerwinkel α, β, γ einer kontinuierlichen Rotation mit 2 Parametern θ_1, θ_2:



Eulerwinkel – Gimbal Lock (1)

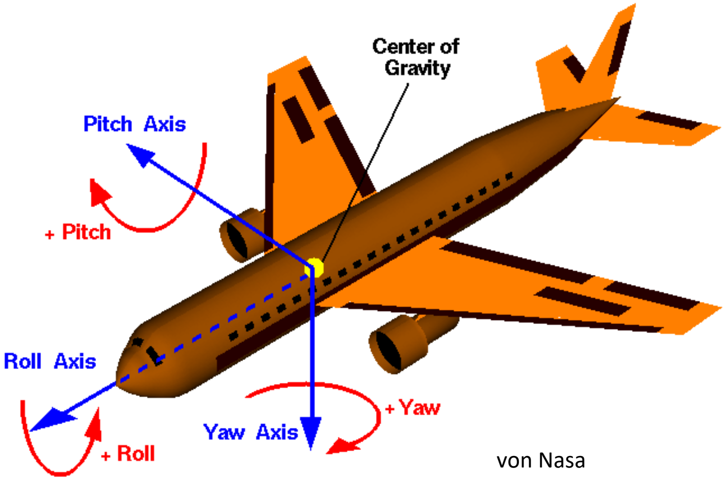
- **12 verschiedene Abfolgen** sind für die Rotationsmatrizen möglich:
  - *z-x-z* *x-y-x* *y-z-y* *z-y-z* *x-z-x* *y-x-y*
  - *x-y-z* *y-z-x* *z-x-y* *x-z-y* *z-y-x* *y-x-z*

- Rotationabfolge x-y-z (Roll-Pitch-Yaw):

$$R_{\mathbf{z},\gamma} = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R_{\mathbf{y},\beta} = \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix}$$

$$R_{\mathbf{x},\alpha} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix}$$



von Nasa [Public domain], via Wikimedia Commons

Eulerwinkel – Gimbal Lock (2)

- Annahme: $\beta = -\frac{\pi}{2}$

  $\sin\left(-\frac{\pi}{2}\right) = -1, \quad \cos\left(-\frac{\pi}{2}\right) = 0$ ⇒ $R_{\mathbf{y},\, \beta=-\frac{\pi}{2}} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$

- Multiplikation der Matrizen:

$$R = R_{z,\gamma} \cdot R_{y,\, \beta=-\frac{\pi}{2}} \cdot R_{x,\alpha} = \begin{pmatrix} 0 & -\sin\gamma & -\cos\gamma \\ 0 & \cos\gamma & -\sin\gamma \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix}$$

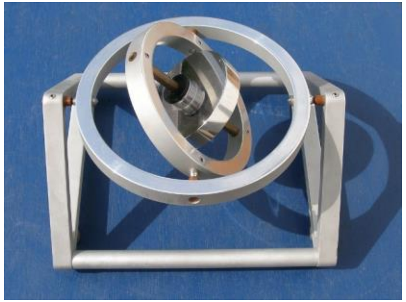
$$= \begin{pmatrix} 0 & -\sin\gamma\cos\alpha - \cos\gamma\sin\alpha & \sin\gamma\sin\alpha - \cos\gamma\cos\alpha \\ 0 & \cos\gamma\ \cos\alpha - \sin\gamma\sin\alpha & -\cos\gamma\sin\alpha - \sin\gamma\cos\alpha \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -\sin(\alpha+\gamma) & -\cos(\alpha+\gamma) \\ 0 & \cos(\alpha+\gamma) & -\sin(\alpha+\gamma) \\ 1 & 0 & 0 \end{pmatrix}$$

⇒ Gemeinsame Drehachse für Drehung um α und γ, 1 DoF geht verloren

Änderungen an α und γ haben momentan den gleichen Effekt

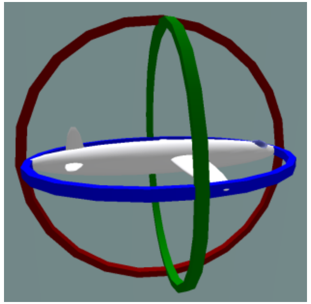
Eulerwinkel – Gimbal Lock (3)

- Gimbal (kardanische Lagerung) erlaubt die Rotation um eine vorgegebene Achse
  - Kombination von 3 Elementen, um freie Bewegung zu ermöglichen
  - Messinstrumente wie Gyroskop, Kompass

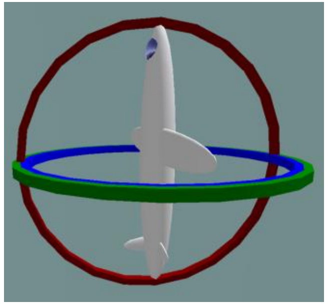


von Bautsch [Public domain], via Wikimedia Commons

- **Gimbal Lock**
  - Bei bestimmten Winkeln werden zwei Achsen voneinander abhängig
  - Ein Freiheitsgrad geht verloren ($\Rightarrow$ keine Momentangeschwindigkeit in diesem Freiheitsgrad möglich)



3 DoF



2 DoF

von MathsPoetry [CC BY-SA 3.0], via Wikimedia Commons

Rotationsmatrix vs. Eulerwinkel

Rotationsmatrizen	Eulerwinkel
■ „Natürliche“ Darstellung aus Sicht der linearen Algebra ■ Eindeutig, kontinuierlich ■ Redundanz durch 9 Werte	■ Kompakter ■ Aussagekräftiger ■ Nicht eindeutig ■ Gimbal Lock ■ Nicht kontinuierlich

Eulerwinkel vs. Roll-Pitch-Yaw

Eulerwinkel $(\mathrm{z}, \mathrm{x}', \mathrm{z}'')$	Roll-Pitch-Yaw $(\mathrm{x}, \mathrm{y}, \mathrm{z})$
■ Multiplikation von links nach rechts $$\mathrm{R_s} = R_{\mathbf{z},\alpha}\, R_{\mathbf{x}',\beta}\, R_{\mathbf{z}'',\gamma}$$ ■ Jede Drehung ist lokal (bezieht sich auf das neue Koordinatensystem) ■ Drehung um jeweils **veränderte** Achsen	■ Multiplikation von rechts nach links $$\mathrm{R_s} = R_{\mathbf{z},\gamma}\, R_{\mathbf{y},\beta}\, R_{\mathbf{x},\alpha}$$ ■ Jede Drehung ist global (bezieht sich auf das globale Koordinatensystem) ■ Drehung jeweils um **feste** Achsen

Darstellung von Orientierung mit 3×3-Matrizen

Bewertung:

- **Vorteil:** Vektor und Rotationsmatrix sind anschaulich und daher eine übliche Form der Eingabe von Posen (z.B. Objekt- und Endeffektorpose)
- **Nachteil:** Vektor- und Matrixoperationen müssen getrennt durchgeführt werden:

$$(\mathbf{p}, R) \text{ mit } \mathbf{p} \in \mathbb{R}^3 \text{ und } R \in \mathrm{SO}(3) \subset \mathbb{R}^{3\times 3}$$

Ziel: **Geschlossene Darstellung** von Rotation und Translation in einer Matrix

→ Benutzung affiner Transformationen
(Einsatz u.a. in der projektiven Geometrie)

Affine Transformationen

- Der **affine Raum** ist eine **Erweiterung des Euklidischen Raums**.
- Er beinhaltet Punkte und Vektoren, die in **erweiterten** („**homogenen**“) **Koordinaten** ausgedrückt werden:

$$\mathbf{a} = (a_x, a_y, a_z, h)^\top, \quad h \in \{0,1\}$$

für Ortsvektoren (Punkte)

für Richtungsvektoren

- Affine Transformationen können so definiert werden, dass **lineare Transformationen** im euklidischen Raum (wie Rotation, Skalierung und Scheren um den Ursprung) **mit Translationen kombiniert** und in homogenen Koordinaten ausgedrückt werden können:

$$\boxed{\mathbf{b} = A \cdot \mathbf{x} + \mathbf{t} \quad \Leftrightarrow \quad \begin{pmatrix} \mathbf{b} \\ 1 \end{pmatrix} = \begin{pmatrix} A & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} + \begin{pmatrix} \mathbf{t} \\ 0 \end{pmatrix} = \begin{pmatrix} A & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}}$$

$\mathbf{0}$ stellt den Nullvektor dar.

Affine Transformationen: Vorteile

- Es wird ermöglicht, **Rotationen um beliebige Achsen** im affinen Raum zu formulieren.

- **Rotationen und Translationen** können in einer **einzigen homogenen 4×4 -Matrix** kombiniert werden.

  Das heißt, Rotationen und Translationen können einheitlich abgehandelt werden.

Homogene 4×4 -Matrizen (1)

- Allgemeine homogene 4×4 Matrix:

$$T = \begin{pmatrix} R & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{pmatrix} \qquad T \in \mathrm{SE}(3) \quad \text{mit } \mathbf{t} \in \mathbb{R}^3 \text{ und } R \in \mathrm{SO}(3)$$

- **Translationsmatrix:** Verschiebung des Objekt-Koordinatensystem (OKS) nach $\mathbf{t} = \left(t_x, t_y, t_z\right)^\top$ im Basis-Koordinatensystem (BKS):

$$T_{\text{trans}} = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Homogene 4×4 -Matrizen (2)

Basisrotationsmatrizen:

$$T_{x,\alpha} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_{y,\beta} = \begin{pmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_{z,\gamma} = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Beispiel: Verschiebung mit homogenen 4×4 –Matrizen

Zwei Punkte $\mathbf{a}$ und $\mathbf{b}$ sollen um $+5$ in x-Richtung und um -3 in z-Richtung verschoben werden.

$$\mathbf{a} = (4, 3, 2, 1)^\top \qquad \mathbf{b} = (6, 2, 4, 1)^\top$$

$$\mathbf{a}' = A \cdot \mathbf{a} = \begin{pmatrix} 1 & 0 & 0 & +5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 4 \\ 3 \\ 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 9 \\ 3 \\ -1 \\ 1 \end{pmatrix}$$

$$\mathbf{b}' = A \cdot \mathbf{b} = \begin{pmatrix} 1 & 0 & 0 & +5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 6 \\ 2 \\ 4 \\ 1 \end{pmatrix} = \begin{pmatrix} 11 \\ 2 \\ 1 \\ 1 \end{pmatrix}$$

Homogene 4×4 -Matrizen: Invertierung

$$\mathbf{b} = R \cdot \mathbf{x} + \mathbf{t} \quad \Leftrightarrow \quad \begin{pmatrix} \mathbf{b} \\ 1 \end{pmatrix} = T \cdot \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} = \begin{pmatrix} R & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$$

1. Rotiere $\mathbf{x}$ um R
2. Verschiebe das Ergebnis um $\mathbf{t}$ (im *rotierten* Koordinatensystem)

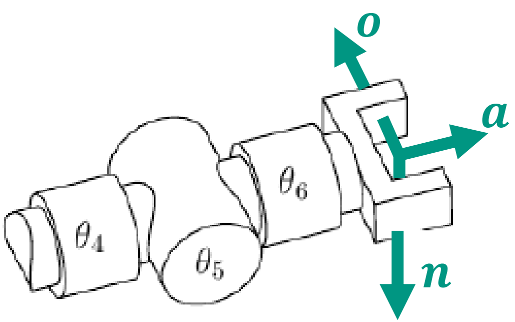
- Wir suchen die homogene Matrix T^{-1}, die $\mathbf{b}$ wieder auf $\mathbf{x}$ abbildet:

$$R \cdot \mathbf{x} + \mathbf{t} = \mathbf{b}$$
$$R \cdot \mathbf{x} = \mathbf{b} - \mathbf{t}$$
$$\mathbf{x} = R^{-1} \cdot (\mathbf{b} - \mathbf{t})$$
$$\mathbf{x} = R^{-1} \cdot \mathbf{b} - R^{-1} \cdot \mathbf{t}$$
$$\mathbf{x} = \left(R^{-1}\right) \cdot \mathbf{b} + \left(-R^{-1} \cdot \mathbf{t}\right)$$
$$\mathbf{x} = \left(R^\top\right) \cdot \mathbf{b} + \left(-R^\top \cdot \mathbf{t}\right)$$

$$T^{-1} = \begin{pmatrix} R^\top & -R^\top \cdot \mathbf{t} \\ \mathbf{0}^\top & 1 \end{pmatrix}$$

Homogene 4×4 -Matrizen als Transformation (1)

Abbildung des Ortsvektors $\mathbf{p}_{OKS}$ (Punkt im OKS) ins BKS:



$$\mathbf{p}_{\text{BKS}} = T \cdot \mathbf{p}_{\text{OKS}}$$

$$\text{mit:} \quad T = \begin{pmatrix} n_x & o_x & a_x & u_x \\ n_y & o_y & a_y & u_y \\ n_z & o_z & a_z & u_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{n} & \mathbf{o} & \mathbf{a} & \mathbf{u} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$\mathbf{u}$: Ursprung des Objekt-KS (OKS)
$\mathbf{n}$, $\mathbf{o}$, $\mathbf{a}$: Einheitsvektoren des OKS bezüglich des Basis-KS (BKS)

$\mathbf{n}$ normal
$\mathbf{a}$ approach
$\mathbf{o}$ orientation

Homogene 4×4 -Matrizen als Transformation (2)

- Invertierung:

$$T = \begin{pmatrix} \mathbf{n} & \mathbf{o} & \mathbf{a} & \mathbf{u} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} n_x & o_x & a_x & u_x \\ n_y & o_y & a_y & u_y \\ n_z & o_z & a_z & u_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T^{-1} = \begin{pmatrix} & R^\top & & -R^\top \mathbf{u} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} n_x & n_y & n_z & -\mathbf{n}^\top \mathbf{u} \\ o_x & o_y & o_z & -\mathbf{o}^\top \mathbf{u} \\ a_x & a_y & a_z & -\mathbf{a}^\top \mathbf{u} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Homogene 4×4 -Matrizen: Eigenschaften

- Eine homogene 4×4 -Matrix enthält **12** $(\mathbf{n}, \boldsymbol{o}, \mathbf{a}, \mathbf{u})$ nichttriviale Größen im Gegensatz zu **6** $(x, y, z, \alpha, \beta, \gamma)$ notwendigen

- Redundanz, aber mit zusätzlichen Randbedingungen, die Orthogonalität ($R \cdot R^\top = I$) garantieren

- Drehachsen und Drehreihenfolge sind implizit enthalten

Vergleich: Kartesische und Homogene Darstellung

- In kartesischen Koordinaten:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$$

- In homogenen Koordinaten:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} n_x & o_x & a_x & t_x \\ n_y & o_y & a_y & t_y \\ n_z & o_z & a_z & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Interpretationen von homogenen 4×4 –Matrizen

- **Lagebeschreibung** eines Koordinatensystems:

$^{A}P_{B}$ beschreibt die Lage (Pose) des Koordinatensystems B relativ zum Koordinatensystem A

- **Transformationsabbildung** (zwischen Koordinatensystemen):

$$^{A}T_{B}: \quad {^{B}P} \rightarrow {^{A}P}, \qquad\qquad {^{A}P} = {^{A}T_{B}} \cdot {^{B}P}$$

- **Transformationsoperator** (innerhalb eines Koordinatensystems):

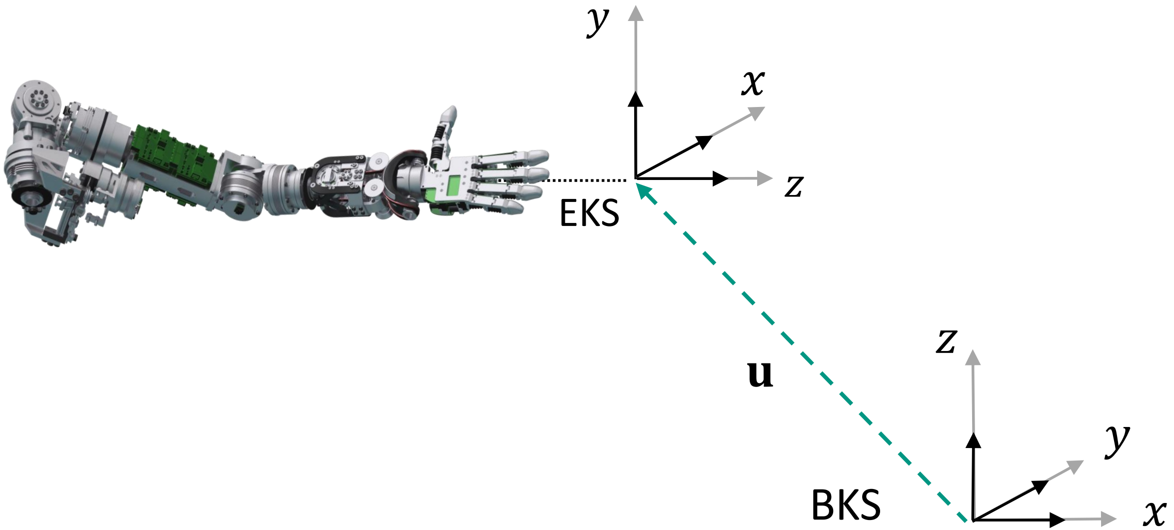
$$T: \quad {^{A}P_1} \rightarrow {^{A}P_2}, \qquad\qquad {^{A}P_2} = T \cdot {^{A}P_1}$$

Beispiel: Koordinatensystem-Transformation (1)

- Gegeben: Punkt im Endeffektor-Koordinatensystem (EKS)

$$^{\mathrm{EKS}}\mathbf{p} = (0, -3, 5)^{\top}$$

- Gesucht: Punkt im Basis-Koordinatensystem (BKS) $^{\mathrm{BKS}}\mathbf{p}$



$$\mathbf{u} = \begin{pmatrix} -7 \\ 0 \\ 8 \end{pmatrix}$$

$$R = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Beispiel: Koordinatensystem-Transformation (2)

- Gegeben: Punkt im Endeffektor-Koordinatensystem (EKS)

$$^{\mathrm{EKS}}\mathbf{p} \ = \ (0, -3, 5)^{\mathsf{T}}$$

- Gesucht: Punkt im Basis-Koordinatensystem (BKS) $^{\mathrm{BKS}}\mathbf{p}$

$$\mathbf{u} = \begin{pmatrix} -7 \\ 0 \\ 8 \end{pmatrix} \qquad R = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

$$^{\mathrm{BKS}}\mathbf{p} = \begin{pmatrix} 0 & 0 & 1 & -7 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 8 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ -3 \\ 5 \\ 1 \end{pmatrix} = \begin{pmatrix} -2 \\ 0 \\ 5 \\ 1 \end{pmatrix}$$

Verkettete Lagebeschreibung (1)

Sei

$^{\mathrm{BKS}}T_A$ die Lagebeschreibung des Objekts A bzgl. BKS

$^{A}T_B$ die Lagebeschreibung eines Objekts B, bezogen auf das OKS von A

$^{\mathrm{BKS}}T_B$ die Lagebeschreibung des Objekts B bzgl. BKS

So gilt $\quad ^{\mathrm{BKS}}T_B = {}^{\mathrm{BKS}}T_A \cdot {}^{A}T_B$

Kompaktere Schreibweise im Vergleich zur kartesischen Darstellung:

$$R_{Bneu} + \mathbf{t}_{Bneu} = R_A \cdot (R_B + \mathbf{t}_B) + \mathbf{t}_A = R_A \cdot R_B + (R_A \cdot \mathbf{t}_B + \mathbf{t}_A)$$

Verkettete Lagebeschreibung (2)

- Lage von Objekt 1 bzgl. BKS: ${}^{\mathrm{BKS}}T_{O1}$
- Lage von Objekt 2 bzgl. Objekt 1: ${}^{O_1}T_{O_2}$
- Lage von Objekt 3 bzgl. Objekt 2: ${}^{O_2}T_{O_3}$
- Lage von Objekt 3 bzgl. BKS: ${}^{\mathrm{BKS}}T_{O_3}$

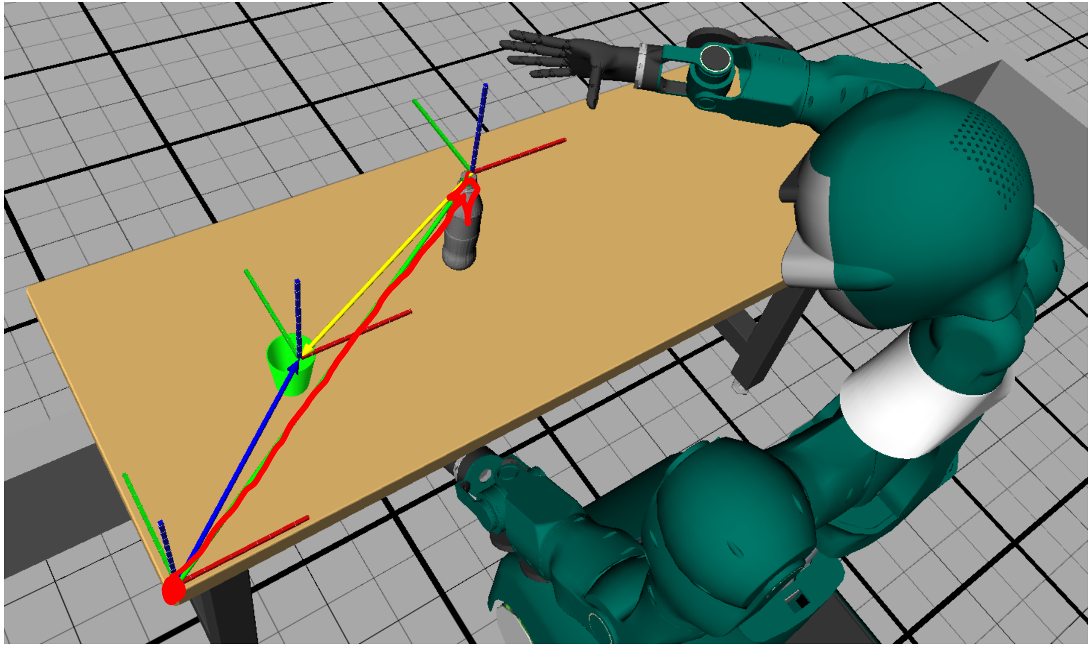
$$ {}^{\mathrm{BKS}}T_{O_3} = {}^{\mathrm{BKS}}T_{O_1} \cdot {}^{O_1}T_{O_2} \cdot {}^{O_2}T_{O_3} $$

Bei einer verketteten Lagebeschreibung durch ein Produkt von Matrizen muss jede Matrix sich auf die durch die jeweils links stehende Matrix definierte Lage beziehen:

$$ {}^{A_0}T_{A_n} = \prod_{i=1}^{n} {}^{A_{i-1}}T_{A_i} \qquad \text{mit } A_0 = \mathrm{BKS} $$

Verkettete Lagebeschreibung: Beispiel

$$ {}^{\mathrm{BKS}}\boldsymbol{H}_{\mathrm{Tasse}} = {}^{\mathrm{BKS}}\boldsymbol{H}_{\mathrm{Flasche}} \cdot {}^{\mathrm{Flasche}}\boldsymbol{H}_{\mathrm{Tasse}} $$



Quaternionen

Karlsruher Institut für Technologie

- Probleme mit Rotationsmatrizen:
  - Redundant
  - Schwierige Interpolation
- Probleme mit Eulerwinkeln:
  - Numerisch instabil
  - Gimbal Lock

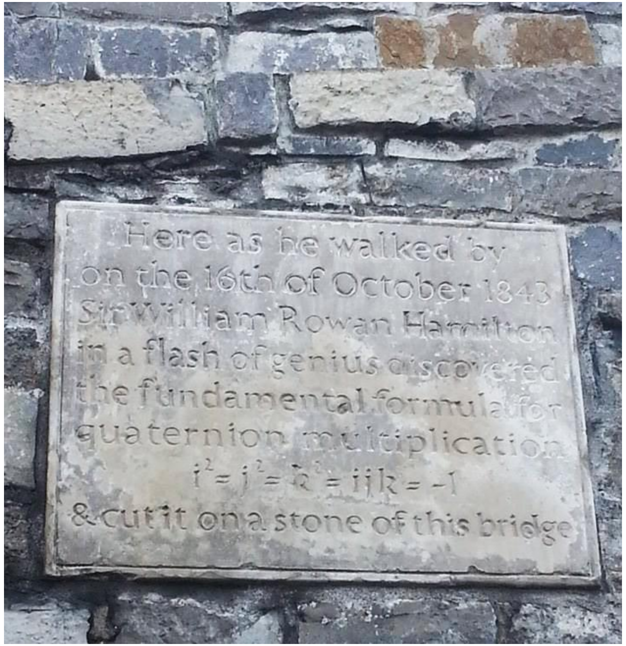
- Existiert eine andere Repräsentation ohne diese Nachteile?
- **Antwort: Ja, Quaternionen!**
  - Quaternionen sind eine **Verallgemeinerung der komplex Zahlen** $\mathbb{C}$ („Hyperkomplexe Zahlen“)
  - Geprägt 1843 (Oktober 16) durch William Rowan Hamilton
  - Siehe (Horn 1987)
  - **Quaternionen können Rotationen in 3D repräsentieren**

Berthold K. P. Horn, **Closed-Form Solution of Absolute Orientation Using Unit Quaternions**,
Journal of the Optical Society of America A 4(4):629-642; April 1987, DOI: 10.1364/JOSAA.4.000629

H2T

Quaternionen

- Broome Bridge in Dublin



Visualisierung zu Quaternionen

- Explorable Video, by Grant Sanderson and Ben Eater
- https://www.youtube.com/watch?app=desktop&v=d4EgbgTm0Bg

Quaternionen: Definition

$h = (a + ib) + (c + id)j$
$= a + ib + cj + ij \cdot d$

- Die Menge der **Quaternionen** $\mathbb{H}$ ist definiert durch

$$\mathbb{H} = \mathbb{C} + \mathbb{C}\,j \quad \text{mit} \quad j^2 = -1 \quad \text{und} \quad i \cdot j = -j \cdot i = k$$

$k^2 = -1$

- Ein Element $\mathbf{q} \in \mathbb{H}$ hat die Form

$$\mathbf{q} = (a, \mathbf{u})^\top = a + u_1 i + u_2 j + u_3 k \quad \text{mit } a \in \mathbb{R}, \mathbf{u} \in \mathbb{R}^3 \text{ und } k = i \cdot j$$

- a wird als **Realteil** bezeichnet
- $\mathbf{u} = (u_1, u_2, u_3)^\top$ wird als **Imaginärteil** bezeichnet

- In Code sind übliche Notationen (w, x, y, z) oder (x, y, z, w) mit $w = a$ und $(x, y, z) = \mathbf{u}$

Quaternionen: Rechenregeln (1)

Karlsruher Institut für Technologie

$$\boldsymbol{q} = (a, \boldsymbol{u})^\top = a + u_1 i + u_2 j + u_3 k$$

$$\boxed{\begin{array}{lllllllll} i^2 & = & j^2 & = & k^2 & = & i \cdot j \cdot k & = & -1 \\ i \cdot j & = & -j \cdot i & = & k & & \text{(nicht kommutativ!)} & & \\ k \cdot i & = & -i \cdot k & = & j & & & & \end{array}}$$

$\cdot$	$\mathbf{1}$	$\boldsymbol{i}$	$\boldsymbol{j}$	$\boldsymbol{k}$
$\mathbf{1}$	1	i	j	k
$\boldsymbol{i}$	i	-1	k	$-j$
$\boldsymbol{j}$	j	$-k$	-1	i
$\boldsymbol{k}$	k	j	$-i$	-1

$i \quad j \quad k$

H2T

Quaternionen: Rechenregeln (2)

- Gegeben zwei Quaternionen $\mathbf{q}, \mathbf{r}$:

$$\mathbf{q} = (a, \mathbf{u})^\top, \qquad \mathbf{r} = (b, \mathbf{v})^\top$$

- **Addition**:

$$\mathbf{q} + \mathbf{r} = (a + b, \mathbf{u} + \mathbf{v})^\top$$

- **Skalarprodukt**:

$$\langle \mathbf{q} | \mathbf{r} \rangle = a \cdot b + \langle \mathbf{v} | \mathbf{u} \rangle = a \cdot b + v_1 \cdot u_1 + v_2 \cdot u_2 + v_3 \cdot u_3$$

- **Multiplikation**:

$$\mathbf{q} \cdot \mathbf{r} = (a + u_1 i + u_2 j + u_3 k) \cdot (b + v_1 i + v_2 j + v_3 k)$$

Quaternionen: Rechenregeln (2)

Karlsruher Institut für Technologie

- Quaternion:

$$\mathbf{q} = (a, \mathbf{u})^\mathsf{T}$$

$q = 1 + i + 2j - 3k$

- **Konjugierte** Quaternion:

$$\mathbf{q}^* = (a, -\mathbf{u})^\mathsf{T}$$

$q^* = 1 - i - 2j + 3k$

- **Norm** einer Quaternion:

$$|\mathbf{q}| = \sqrt{\mathbf{q} \cdot \mathbf{q}^*} = \sqrt{\mathbf{q}^* \cdot \mathbf{q}} = \sqrt{a^2 + u_1^2 + u_2^2 + u_3^2}$$

$\|q\| = 1$

- **Inverse** einer Quaternion:

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{|\mathbf{q}|^2}$$

H2T

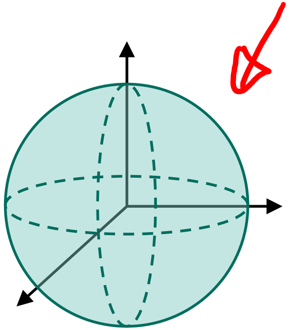
Quaternionen: Rotationen (1)

Einheitsquaternionen $\mathbb{S}^3 = \{\mathbf{q} \in \mathbb{H} \mid \|\mathbf{q}\|^2 = 1\}$

- Existieren auf der Einheitskugel $\mathbb{S}^3$ in 4D
  - Norm = 1

  $\Rightarrow$ 1 von 4 „Freiheitsgraden“ festgelegt

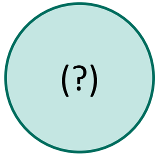
  $\Rightarrow$ 3 „Freiheitsgrade“ übrig



Einheitskugel $\mathbb{S}^2$ in 3D

- Bilden eine Gruppe. Erinnerung Gruppeneigenschaften:
  - Assoziativgesetz
  - Existenz eines inversen Elements zu jedem Gruppenelement
  - Existenz eines Nullelements

Einheitskugel $\mathbb{S}^3$ in 4D



- **Definieren Rotationen**

  Es existiert eine Einbettung von $SO(3) \subset \mathbb{R}^3$ nach $\mathbb{H}$

Quaternionen: Rotationen (2)

Frage: Wie repräsentiert man eine Rotation z.B. 46° um die Achse $(0,1,0)^\top$ als Quaternion?

- Beschreibung eines **Vektors** $\mathbf{p} \in \mathbb{R}^3$ als Quaternion $\mathbf{q}$:

$$\mathbf{p} = (x, y, z)^\top \quad \Longrightarrow \quad \mathbf{q} = (0, \mathbf{p})^\top$$

- Beschreibung eines **Skalars** $s \in \mathbb{R}$ als Quaternion $\mathbf{q}$:

$$\mathbf{q} = (s, \mathbf{0})^\top$$

Quaternionen: Rotationen (3)

- Sei eine Rotation beschrieben durch eine **Drehachse** $\mathbf{a}$ mit $\|\mathbf{a}\| = 1$ und einen **Drehwinkel** θ, dann existiert hierfür eine Repräsentation als Quaternion

$$\mathbf{q} = \left(\cos\frac{\theta}{2}, \mathbf{a}\sin\frac{\theta}{2}\right)$$

- Ein Punkt $\mathbf{v}$ wird mit einer Quaternion $\mathbf{q}$ rotiert durch:

$$\mathbf{v}' = \mathbf{q}\,\mathbf{v}\,\mathbf{q}^{-1}$$

- Da $\mathbf{q}$ eine Einheitsquaternion ist, gilt $\mathbf{q}^{-1} = \mathbf{q}^*$ und somit folglich:

$$\mathbf{v}' = \mathbf{q}\,\mathbf{v}\,\mathbf{q}^*$$

Quaternionen: Rotationen (4)

- Verkettung zweier Rotationen eines Vektors $\mathbf{v}$ mit zwei Quaternionen $\mathbf{q}$ und $\mathbf{r}$:

$$\mathbf{q} = \left(\cos\frac{\theta_q}{2}, \mathbf{u}_q \sin\frac{\theta_q}{2}\right), \qquad \mathbf{r} = \left(\cos\frac{\theta_r}{2}, \boldsymbol{u}_r \sin\frac{\theta_r}{2}\right)$$

- Rotation mit jeweils einer Quaternion:

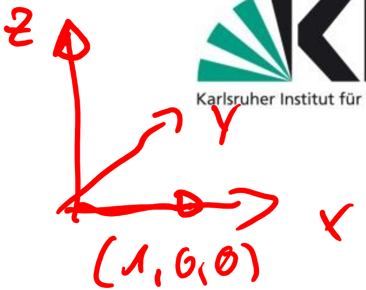
$$f(\mathbf{v}) = \mathbf{q}\,\mathbf{v}\,\mathbf{q}^*, \qquad h(\mathbf{v}) = \mathbf{r}\,\mathbf{v}\,\mathbf{r}^*$$

- Verkettung $f \circ h$:

$$f\big(h(\mathbf{v})\big) = \mathbf{q}\,(\mathbf{r}\,\mathbf{v}\,\mathbf{r}^*)\,\mathbf{q}^*$$

- $f \circ h$ entspricht gerade der Rotation mit der Quaternion $\mathbf{s} = \mathbf{q} \cdot \mathbf{r}$
  $\Rightarrow$ **Verkettung $\widehat{=}$ Multiplikation**

Quaternionen: Beispiel



- Rotation des Punkts um die Drehachse mit Winkel

$$\mathbf{p} = (1,0,9)^\top$$
$$\mathbf{a} = (1,0,0)^\top$$
$$\theta = 90°$$

p als Quaternion: $v = 0 + 1 \cdot i + 0 \cdot j + 9k$

$= i + 9k$ ←

Rotationsquat. $\theta = 90°$ $\quad a = (1,0,0)^T$

$$q = \left(\cos\frac{\theta}{2},\ a \cdot \sin\frac{\theta}{2}\right)$$

$$q = \cos 45^\circ + 1 \cdot i \cdot \sin 45 + 0 \cdot j \sin 45 + 0 \cdot k \cdot \sin 45$$

$$= \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} i = \frac{1}{\sqrt{2}} (1 + i)$$

$$v_r = q \cdot v \cdot q^*$$

$$q^* = \frac{1}{\sqrt{2}} (1 - i)$$

Rotation von v um q: $v_r = q \cdot v \cdot q^*$

$$v_r = \frac{1}{\sqrt{2}}(1+i)(i+gk)\cdot\frac{1}{\sqrt{2}}(1-i)$$

$$= \frac{1}{2}(i+gk+\underbrace{i^2}_{=-1}+g\underbrace{ik}_{=-j})(1-i)$$

$$= \frac{1}{2}(i+gk-1-gj)(1-i)$$

$$= i-gj$$

$$v_r = 0+i-gj+0\cdot k$$

Quaternionen: Beispiel

- Beispiel: Rotation des Punkts $\mathbf{p} = (1,0,9)^\mathsf{T}$
  um die Drehachse $\mathbf{a} = (1,0,0)^\mathsf{T}$
  mit Winkel $\theta = 90°$

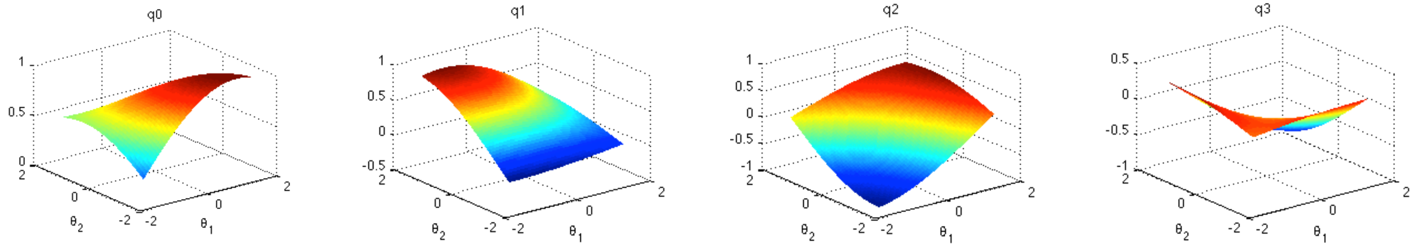
1. Darstellung von $\mathbf{p}$ als Quaternion $\mathbf{v}$ $\quad \mathbf{v} = 0 + 1i + 0j + 9k$
2. Rotationsquaternion $\mathbf{q}$ $\quad \mathbf{q} = \cos\frac{\theta}{2} + 1i \cdot \sin\frac{\theta}{2} + 0j + 0k$
3. Konjugierte Quaternion $\mathbf{q}^*$ $\quad \mathbf{q}^* = \cos\frac{\theta}{2} - 1i \cdot \sin\frac{\theta}{2} - 0j - 0k$
4. Rotation von $\mathbf{v}$ um $\mathbf{q}$ $\quad \mathbf{v}_\mathrm{r} = \mathbf{q}\,\mathbf{v}\,\mathbf{q}^* \quad \rightarrow \mathbf{v}_\mathrm{r} = 0 + 1i - 9j + 0k$
5. Darstellung als Punkt $\mathbf{p}_\mathrm{r}$ $\quad \mathbf{p}_\mathrm{r} = (1,-9,0)^\mathsf{T}$

Achtung: Die Multiplikation von Quaternionen ist <u>nicht</u> kommutativ.

Quaternionen: Bewertung

Vorteile:

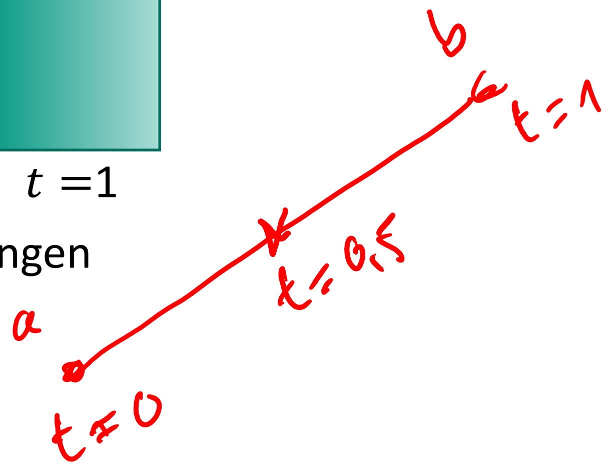
- Kompakte Darstellung: 4 Werte im Vergleich zu 9 bei Rotationsmatrizen
- Anschaulich (angelehnt an Axis-Winkel Repräsentation)
- Verkettung möglich, ähnlich wie bei Rotationsmatrix
- Kann für Berechnung der Inversen Kinematik verwendet werden (späteres Kapitel)
- Kein Gimbal Lock
- Repräsentation ist stetig (keine Sprünge, siehe Bilder)



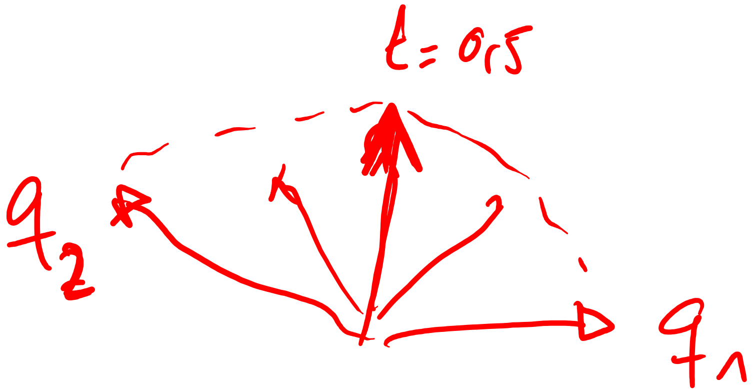
Nachteil:

- Nur Beschreibung von Rotation, keine Translation

Quaternionen: Interpolation



- **Ziel:** Kontinuierliche Rotation zwischen zwei Orientierungen
- **Probleme:**
  - Euler-Winkel sind nicht kontinuierlich
  - Rotationsmatrizen haben viele Freiheitsgrade
- Interpolation von Quaternionen mittels **SLERP** (Spherical Linear Interpolation)
- Ähnlich zu linearer Interpolation: $a \cdot (1 - t) + b \cdot t$



t = 0,5
q2
q1

Quaternionen: SLERP

- SLERP Interpolation von $\mathbf{q}_1$ nach $\mathbf{q}_2$ mit dem Parameter $t \in [0, 1]$:

$$\mathrm{Slerp}(\mathbf{q}_1, \mathbf{q}_2, t) = \mathbf{q_1} \cdot (\mathbf{q}_1^{-1} \cdot \mathbf{q}_2)^t$$

  (Potenzen von Quaternionen wird nicht in der Vorlesung behandelt)

- Direkte Formulierung der SLERP Interpolation:

$$\mathrm{Slerp}(\mathbf{q}_1, \mathbf{q}_2, t) = \frac{\sin\big((1-t)\cdot\theta\big)}{\sin\theta} \cdot \mathbf{q}_1 + \frac{\sin(t\cdot\theta)}{\sin\theta} \cdot \mathbf{q}_2 \qquad \text{mit} \quad \langle \mathbf{q_1} | \mathbf{q_2} \rangle = \cos\theta$$

- Ergebnis: Rotation mit konstanter Winkelgeschwindigkeit

Quaternionen: Interpolationsprobleme

- **Problem:** Orientierungen in $\mathrm{SO}(3)$ werden durch Einheitsquaternionen doppelt abgedeckt, weil die Einheitsquaternionen $\mathbf{q}$ und $-\mathbf{q}$ der gleichen Rotation entsprechen.

  Beweis:
  - Rotation von $\mathbf{v}$ um $\mathbf{q}$ entspricht Rotation von $\mathbf{v}$ um $-\mathbf{q}$.
  - $\mathbf{v}_\mathrm{r} = \mathbf{q}\,\mathbf{v}\,\mathbf{q}^* = (-\mathbf{q})\,\mathbf{v}\,(-\mathbf{q})^*$
  - Die negativen Vorzeichen löschen sich aus.

- SLERP berechnet deshalb nicht immer die kürzeste Rotation
  $\Rightarrow$ Es muss geprüft werden, ob die Rotation von $\mathbf{q}_1$ zu $\mathbf{q}_2$ oder $-\mathbf{q}_1$ zu $\mathbf{q}_2$ kürzer ist

Duale Quaternionen (1)

$q = a + i u_1 + j u_2 + k u_3$

Problem:

- Reelle Quaternionen (wie bisher) eignen sich für die Beschreibung der Orientierung, …
- aber **nicht zur Beschreibung der Position** eines Objektes (Translation fehlt).

Idee:

- Ersetze die 4 reellen Werte einer Quaternion durch **Dualzahlen**
- Erhalte so zusätzliche translatorische Komponente, um die Position eines Objekts ausdrücken zu können
- → **Duale Quaternionen**

Duale Quaternionen (2): Duale Zahlen

- Duale Zahlen sind von der Form

$$\boldsymbol{d} = \boldsymbol{p} + \boldsymbol{\varepsilon} \cdot \boldsymbol{s}, \ \textbf{wobei } \boldsymbol{\varepsilon}^2 = \mathbf{0}$$

- Primärteil p, Sekundärteil s
- Ähnlich wie bei komplexen Zahlen lassen sich die üblichen Operationen ableiten
- Seien $d_1 = p_1 + \varepsilon \cdot s_1$ und $d_2 = p_2 + \varepsilon \cdot s_2$ duale Zahlen, dann gilt:
  - Addition: $d_1 + d_2 = p_1 + p_2 + \varepsilon \cdot (s_1 + s_2)$
  - Multiplikation: $d_1 \cdot d_2 = p_1 \cdot p_2 + \varepsilon \cdot (p_1 \cdot s_2 + p_2 \cdot s_1)$

Duale Quaternionen (3)

Beschreibung

$$DQ = (d_1, d_2, d_3, d_4), \qquad d_i = dp_i + \varepsilon \cdot ds_i$$

- Primärteil dp_i enthält den **Winkelwert** $\theta/2$
- Sekundärteil ds_i enthält die **Translationssgröße** $d/2$

Duale Quaternionen (4)

Multiplikationstabelle für duale Einheitsquaternionen

$\cdot$	1	i	j	k	ε	εi	εj	εk
1	1	i	j	k	ε	εi	εj	εk
i	i	-1	k	$-j$	εi	$-\varepsilon$	εk	$-\varepsilon j$
j	j	$-k$	-1	i	εj	$-\varepsilon k$	$-\varepsilon$	εi
k	k	j	$-i$	-1	εk	εj	$-\varepsilon i$	$-\varepsilon$
ε	ε	εi	εj	εk	0	0	0	0
εi	εi	$-\varepsilon$	εk	$-\varepsilon j$	0	0	0	0
εj	εj	$-\varepsilon k$	$-\varepsilon$	εi	0	0	0	0
εk	εk	εj	$-\varepsilon i$	$-\varepsilon$	0	0	0	0

Duale Quaternionen (5)

- Rotation um eine Achse $\mathbf{a}$ mit dem Winkel θ:

$$\mathbf{q}_r = \left(\cos\left(\frac{\theta}{2}\right), \mathbf{a} \cdot \sin\left(\frac{\theta}{2}\right)\right) + \varepsilon \cdot (0, 0, 0, 0)$$

- Translation mit dem Vektor $\mathbf{t} = (t_x, t_y, t_z)$

$$\mathbf{q}_t = (1, 0, 0, 0) + \varepsilon \cdot \left(0, \frac{t_x}{2}, \frac{t_y}{2}, \frac{t_z}{2}\right)$$

- Kombination zu einer Transformation T:

$$\boldsymbol{q}_T = \boldsymbol{q}_t\, \boldsymbol{q}_r$$

Duale Quaternionen (6)

Karlsruher Institut für Technologie

- Eine Transformation $\boldsymbol{T}$ mit einem rotatorischem Teil $\boldsymbol{r}$ und einem translatorischen Teil $\boldsymbol{t}$, kann als duale Quaternion beschrieben werden durch:

$$\mathbf{q}_T = \mathbf{q}_t\,\mathbf{q}_r$$

- Eine Transformation $\mathbf{q}_T$ wird auf einen Punkt $\mathbf{p}$ (als duale Quaternion) wie folgt angewendet:

$$\mathbf{p}' = \mathbf{q}_T\,\mathbf{p}\,\mathbf{q}_T{}^*, \text{ mit } \mathbf{q}_T{}^* = (\mathbf{q}_t\,\mathbf{q}_r)^* = \mathbf{q}_r{}^*\mathbf{q}_t{}^*$$

- Konjugieren (komplex und dual) von $\mathbf{q} = \mathbf{p} + \varepsilon \cdot \mathbf{s}$:

$$\mathbf{q}^* = \mathbf{p}^* - \varepsilon \cdot \mathbf{s}^*$$

H2T

Duale Quaternionen: Beispiel (1)

- Beispiel: Rotation eines Punkts $\mathbf{p} = (3, 4, 5)^\mathsf{T}$
  um Drehachse $\mathbf{a} = (1, 0, 0)^\mathsf{T}$ mit $\theta = 180°$
  und Translation mit $\mathbf{p}_t = (4, 2, 6)^\mathsf{T}$

- $\mathbf{p}$ als duale Quaternion $\mathbf{v}_d$ $\qquad \mathbf{v}_d = 1 + 3\varepsilon i + 4\varepsilon j + 5\varepsilon k$
- Rotation als duale Quaternion $\mathbf{q}_r$ $\qquad \mathbf{q}_r = \cos\frac{\theta}{2} + 1i \cdot \sin\frac{\theta}{2} + 0j + 0k = i$
- Translation als duale Quaternion $\mathbf{q}_t$ $\qquad \mathbf{q}_t = 1 + 2\varepsilon i + 1\varepsilon j + 3\varepsilon k$

- Kombination als duale Quaternion $\mathbf{q}_T$

$$\mathbf{q}_T = \mathbf{q}_t \cdot \mathbf{q}_r = (1 + 2i\varepsilon + 1j\varepsilon + 3k\varepsilon\,) \cdot i = i - 2\varepsilon - 1\varepsilon k + 3\varepsilon j$$

Duale Quaternionen: Beispiel (2)

- Beispiel: Rotation eines Punkts $\mathbf{p} = (3,4,5)^\mathsf{T}$
  um Drehachse $\mathbf{a} = (1,0,0)^\mathsf{T}$ mit $\theta = 180°$
  und Translation mit $\mathbf{p}_t = (4,2,6)^\mathsf{T}$

$$\mathbf{q}_T = (0 + i) + \varepsilon(-2 - 1k + 3j) = i - 2\varepsilon - 1\varepsilon k + 3\varepsilon j$$
$$\mathbf{q}_T^* = (0 - i) - \varepsilon(-2 + 1k - 3j) = -i + 2\varepsilon + 3\varepsilon j - 1\varepsilon k$$

- Transformation:

$$\mathbf{v}_T = \mathbf{q}_T\, \mathbf{v}_d\, \mathbf{q}_T^* = (i - 2\varepsilon - 1\varepsilon k + 3\varepsilon j)(1 + 3\varepsilon i + 4\varepsilon j + 5\varepsilon k)\, \mathbf{q}_T^*$$

$$= (i - 5\varepsilon - 2\varepsilon j + 3\varepsilon k)(-i + 2\varepsilon + 3\varepsilon j - 1\varepsilon k)$$

$$= 1 + 7\varepsilon i - 2\varepsilon j + 1\varepsilon k$$

- Ergebnis: $\mathbf{p}_T = (7,\ -2,\ 1)^\mathsf{T}$

Duale Quaternionen: Beispiel (3)

- Beispiel: Rotation eines Punkts $\mathbf{p} = (3,4,5)^\top$
  um Drehachse $\mathbf{a} = (1,0,0)^\top$ mit $\theta = 180°$
  und Translation mit $\mathbf{p}_t = (4,2,6)^\top$
- Ergebnis: $\mathbf{p}_T = (7,\ -2,\ 1)^\top$
- Probe:
  - Rotation um die x-Achse mit $\phi = 180°$

$$\mathbf{p}_r = (3, -4, -5)^\top$$

  - Translation mit $\mathbf{p}_t = (4,2,6)^\top$:

$$\mathbf{p}_T = \mathbf{p}_r + \mathbf{p}_t = (3, -4, -5)^\top + (4, 2, 6)^\top = (7, -2, 1)^\top$$

Duale Quaternionen: Bewertung

Vorteile:

- Dualquaternionen sind zur Lagebeschreibung eines Objekts geeignet
- Operationen auf Dualquaternionen erlauben auch alle benötigten Transformationen
- Geringe Redundanz, da nur 8 Werte im Vergleich 12 Werten der homogenen Matrix Darstellung
- In der Regel geringe Anzahl an Einzeloperationen je Rechenoperation

Nachteile:

- Schwierigkeit für den Anwender, eine Lage durch Angabe einer Dualquaternion zu beschreiben
- Komplexe Verarbeitungsvorschriften (z.B. für Multiplikation)

Zusammenfassung

- Verschiedene Darstellungsformen für Rotationen und Translationen im Euklidischen Raum
  - Rotationsmatrix und Verschiebungsvektor
  - Eulerwinkel
  - Homogene $4x4$ Matrix
  - Quaternionen
  - Duale Quaternionen

- Jede Darstellungsart hat spezifische Vor- und Nachteile
- Konkrete Anwendung bestimmt Auswahl der Methode

Englische Begriffe

Deutsch	Englisch
Gelenk	Joint
Glied	Link
Endeffektor	End Effector – Tool Center Point (TCP)
Greifer	Gripper
Kinematische Kette	Kinematic Chain
Freiheitsgrade	Degrees of Freedom (DoF)
Lineare Abbildung	Linear Map
Koordinatensystem	Coordinate System
Euklidischer Raum	Euclidean Space
Basisvektor	Base Vector
Senkrecht	Perpendicular
Basiswechsel	Change of Base