

# Robotik I: Einführung in die Robotik

## Mathematische Grundlagen

**Tamim Asfour**

Institut für Anthropomatik und Robotik

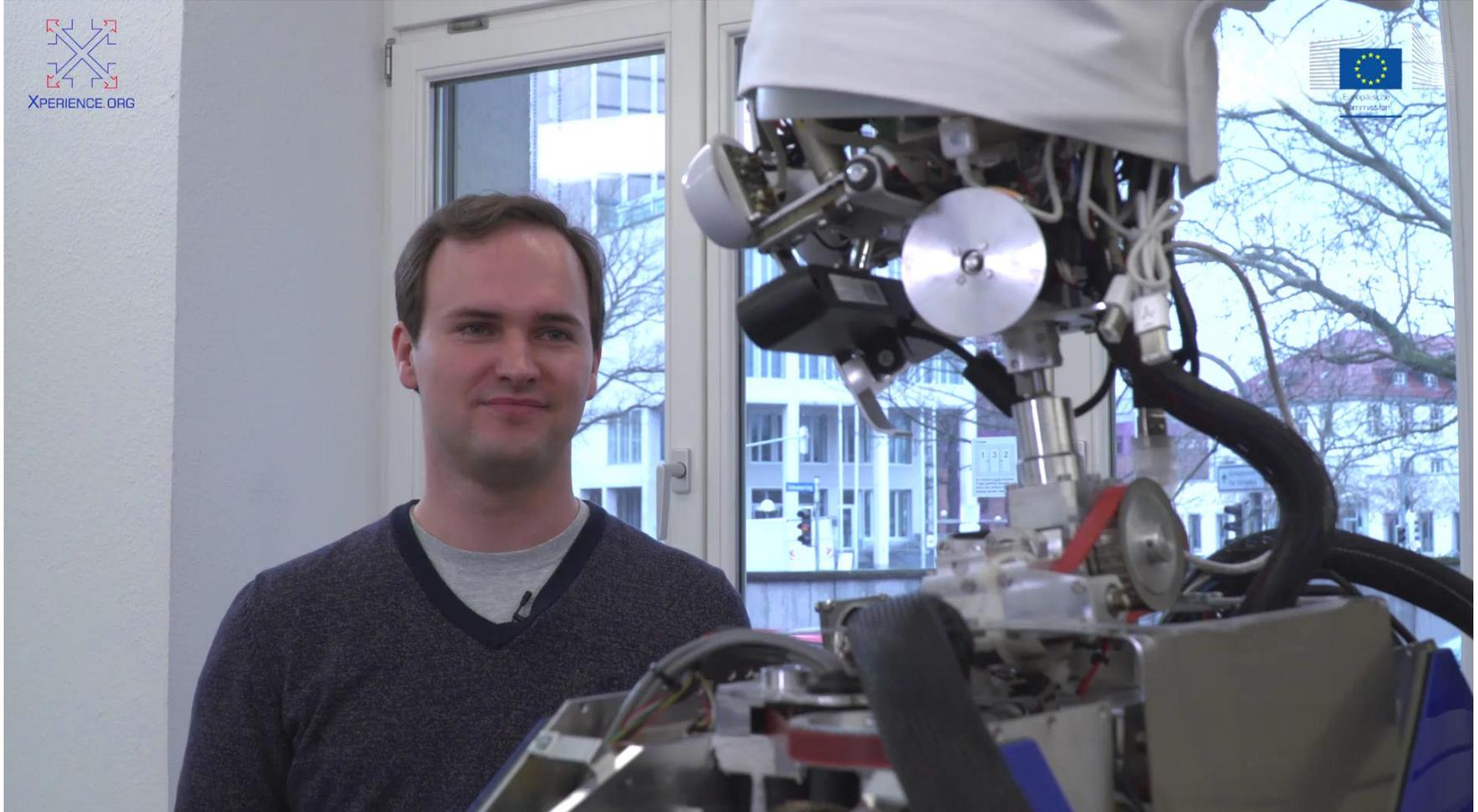
KIT-Fakultät für Informatik, Institut für Anthropomatik und Robotik (IAR)  
Hochperformante Humanoide Technologien (H<sup>2</sup>T)



# Motivation



# Motivation II



# Motivation III



ARMAR!  
Bring me the  
apple juice from  
the fridge

# Motivation

- Welche grundlegenden mathematischen Mittel werden benötigt?
- Wir müssen Punkte und 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 Framework um Positionen (Translationen) und Orientierungen (Rotationen) zu beschreiben**

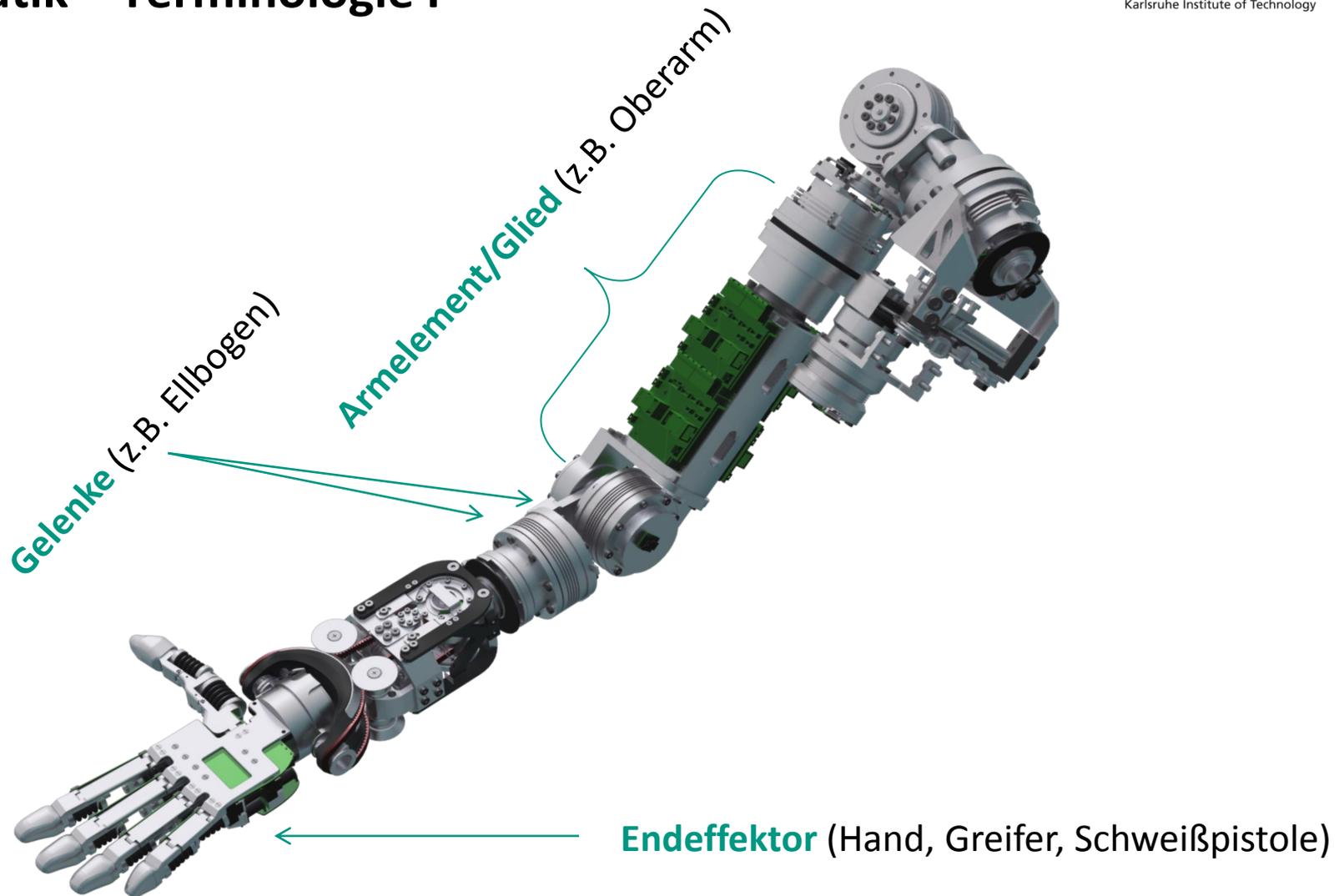
# Kinematische Grundlagen

- Dieses Kapitel ist eine Einführung in die mathematischen Grundlagen der Robotik
- Fokus auf Kinematik (mehr dazu später) :
  - Mathematische Methoden zur Beschreibung von Starrkörper Transformationen  
(Basierend auf Linearer Algebra)
  - Anwendung dieser Methoden um Roboter zu modellieren

# Definitionen

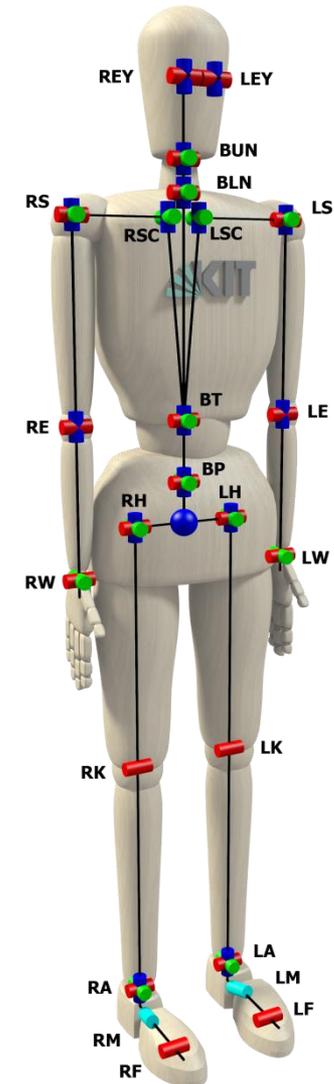
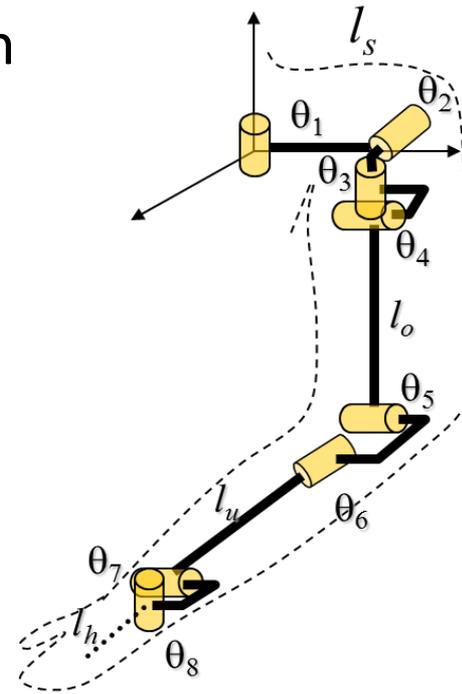
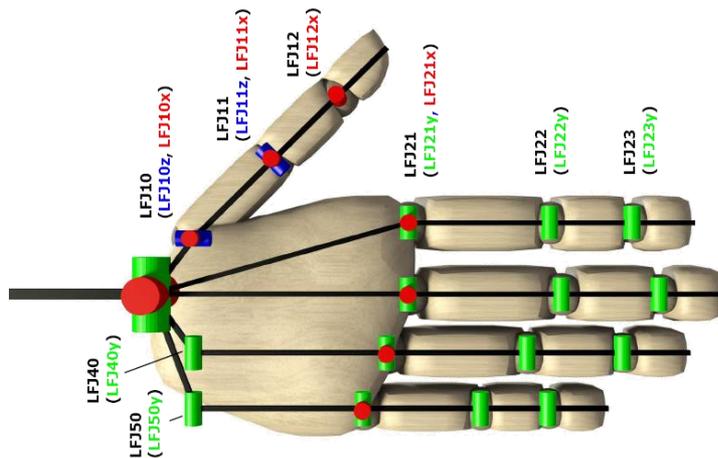
- **Kinematik** analysiert die Geometrie eines Manipulators oder Roboters. Das essentielle Konzept ist die **Position**.
- **Statik** behandelt Kräfte und Momente, die sich auf den ruhenden Mechanismus auswirken. Das essentielle Konzept ist die **Steifigkeit**.
- **Dynamik** analysiert die Kräfte und Momente, die durch Bewegung und Beschleunigung eines Mechanismus und einer zusätzlichen Last entstehen.

# Kinematik – Terminologie I



# Kinematik – Terminologie II

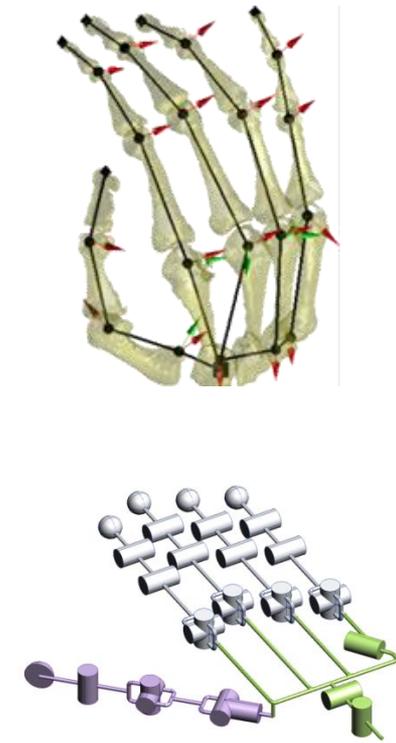
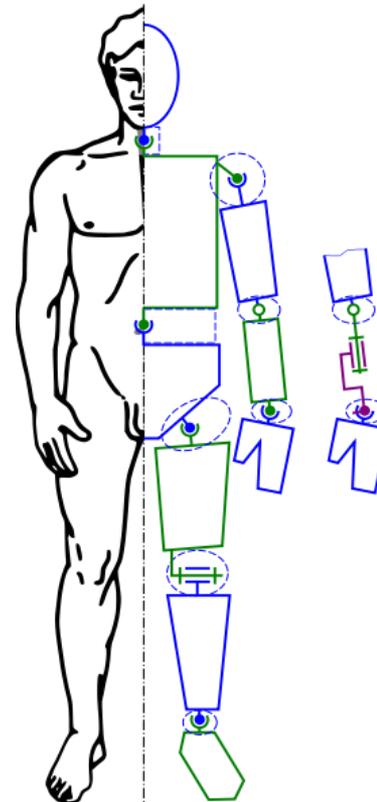
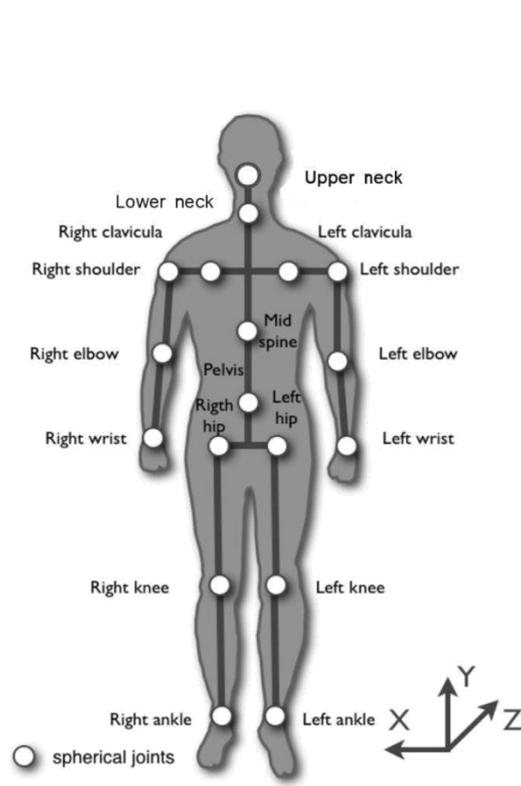
- **Kinematische Kette** ist ein Satz an Gliedern die durch Gelenke verbunden sind.
- Kinematische Ketten können durch einen Graph repräsentiert werden. Kanten repräsentieren Glieder, Knoten repräsentieren Gelenke.





# Kinematik – Terminologie III

## ■ Kinematische Ketten: weitere Beispiele



# Kinematik – Freiheitsgrade

**Freiheitsgrade** (weniger formale Definition) ist eine Anzahl unabhängiger Parameter, die zur kompletten Spezifikation der Position 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

Englisch: **degrees of freedom (DoF)**

# Starrkörperbewegung

Starrkörperbewegungen werden durch zwei Eigenschaften charakterisiert

1. Die Distanz zweier beliebiger Punkte ist konstant
2. Die Orientierung des Körpers bleibt erhalten.  
(Ein rechtsdrehendes Koordinatensystem bleibt rechtsdrehend)

# $SO(3)$ und $SE(3)$

Die folgenden beiden Gruppen sind in der Robotik von besonderem Interesse

- $SO(3)$  - die **Spezielle Orthogonal Gruppe**, die **Rotationen** repräsentiert und
- $SE(3)$  - die **Spezielle Euklidische Gruppe**, die Starrkörper **Bewegungen** repräsentiert
- Elemente aus  $SO(3)$  werden als reale  $3 \times 3$  Matrizen beschrieben und erfüllen

$$R^T R = I \quad \text{with} \quad \det(R) = 1$$

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

# Affine Geometrie

- Wir benutzen affine Geometrie um räumliche Transformationen zu beschreiben.
- Diese Transformationen bestehen aus Verknüpfungen von
  - Rotationen
  - Translationen
- Unterschiedliche Möglichkeiten zur mathematischen Beschreibung räumlicher Transformationen:
  - Rotationsmatrizen und Translationsvektoren
  - Homogene Matrizen
  - Quaternionen
  - Duale Quaternionen
  - ...
- Diese Vorlesung wird die üblichsten Repräsentation einführen.

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

# Euklidischer Raum

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

Ein Punkt **c**, der auf einer Linie zwischen den zwei Punkten **a** und **b** liegt, kann folgendermaßen repräsentiert werden

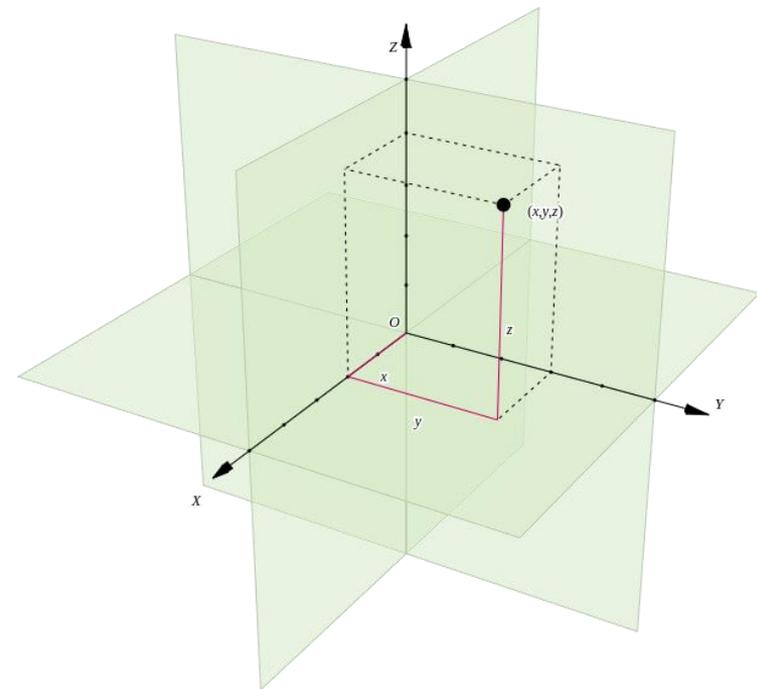
$$\mathbf{c} = t \cdot \mathbf{a} + (1 - t) \cdot \mathbf{b}, \quad t \in (0,1) \subset \mathbb{R}, \quad \mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^3.$$

# Euklidischer Raum II

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

$$\mathbf{a} = a_x \cdot \mathbf{e}_x + a_y \cdot \mathbf{e}_y + a_z \cdot \mathbf{e}_z = (a_x, a_y, a_z)^T. \quad \mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z \in \mathbb{R}^3$$

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

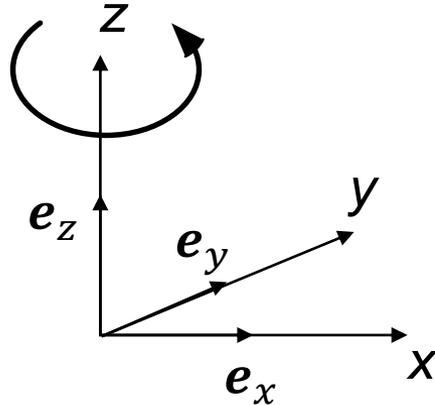


# Koordinatensysteme

## Rechtsdrehendes Koordinatensystem

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

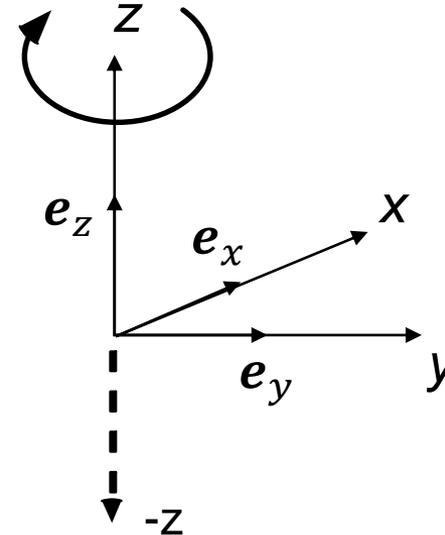
$$\mathbf{x} \times \mathbf{y} = \mathbf{z}$$



## Links-drehendes Koordinatensystem

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

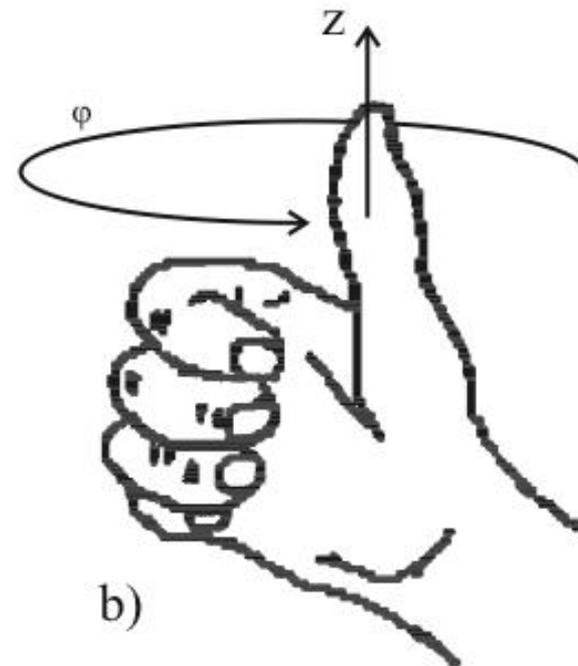
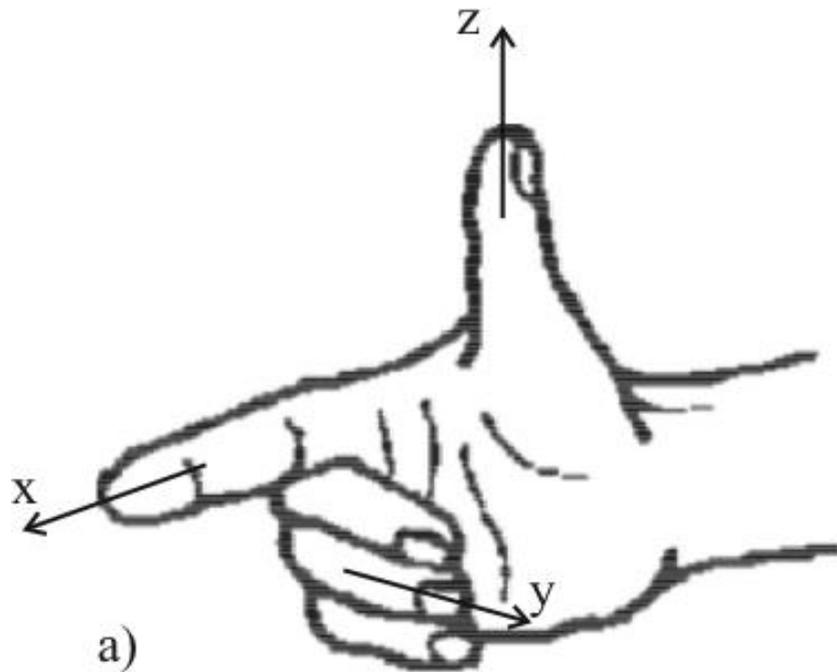
$$\mathbf{x} \times \mathbf{y} = -\mathbf{z}$$



$\times$  = Kreuzprodukt

# Koordinatensysteme II

Rechte-Hand-Regel für rechtsdrehende Koordinatensysteme

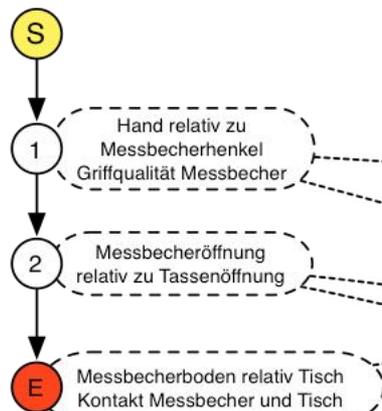


# Koordinatensysteme III

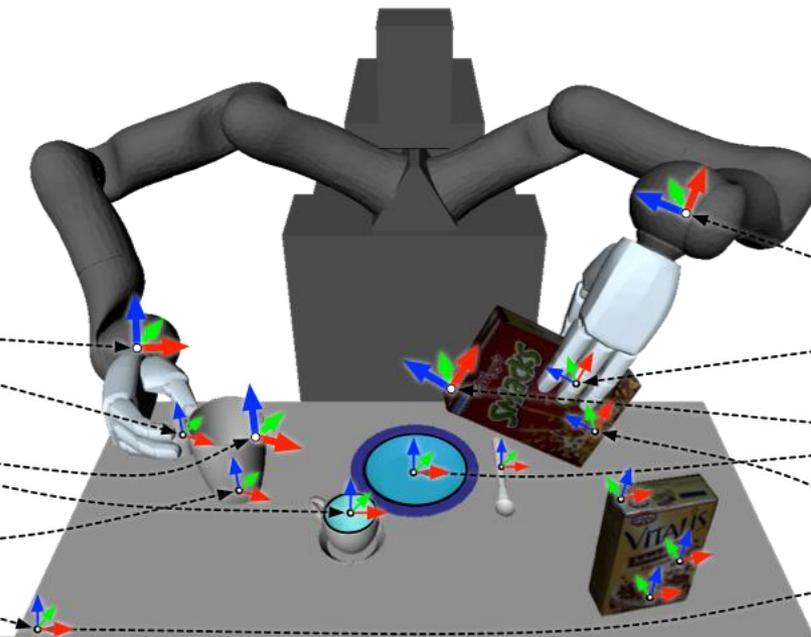
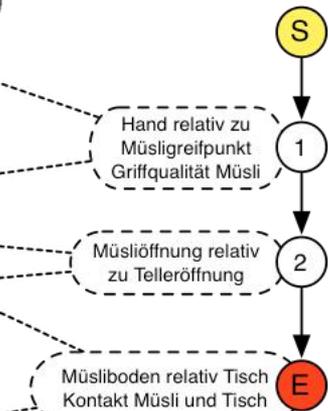
## Basiskoordinatensystem (BKS) als initiales Bezugssystem

- Zusätzlich werden weitere lokale Koordinatensysteme verwendet:
  - OKS: Objektkoordinatensystem
  - EKS: Endeffektorkoordinatensystem
  - SKS: Sensorkoordinatensystem
  - ...

Manipulationsstrategie:  
*Einschenken*  
**Instanz: Messbecher, Tasse**



Manipulationsstrategie:  
*Einschenken*  
**Instanz: Müsli, Teller**



# Lineare Abbildungen, Endomorphismen

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

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

- 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  $\mathbf{e}_x', \mathbf{e}_y', \mathbf{e}_z'$ :

$$A = (\mathbf{e}_x' \quad \mathbf{e}_y' \quad \mathbf{e}_z') \cdot (\mathbf{e}_x \quad \mathbf{e}_y \quad \mathbf{e}_z)^{-1}$$

# Isomorphismen

- **Bijektive** (umkehrbare) Endomorphismen nennt man **Isomorphismem**.
- Isomorphismen können spezielle und interessante Eigenschaften besitzen:
  1. Sie können Winkel erhalten. (Beispiel: Skalierung und Rotation)
  2. Sie können Längen erhalten. (Beispiel: Rotation)
  3. Sie können Händigkeit 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 $SO(3)$

- $SO(3)$  beinhaltet **alle möglichen Rotationen** um willkürlich durch den Ursprung gelegte Achsen.
- $SO(3)$  ist **nicht abelsch (nicht kommutativ)**, z.B.

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

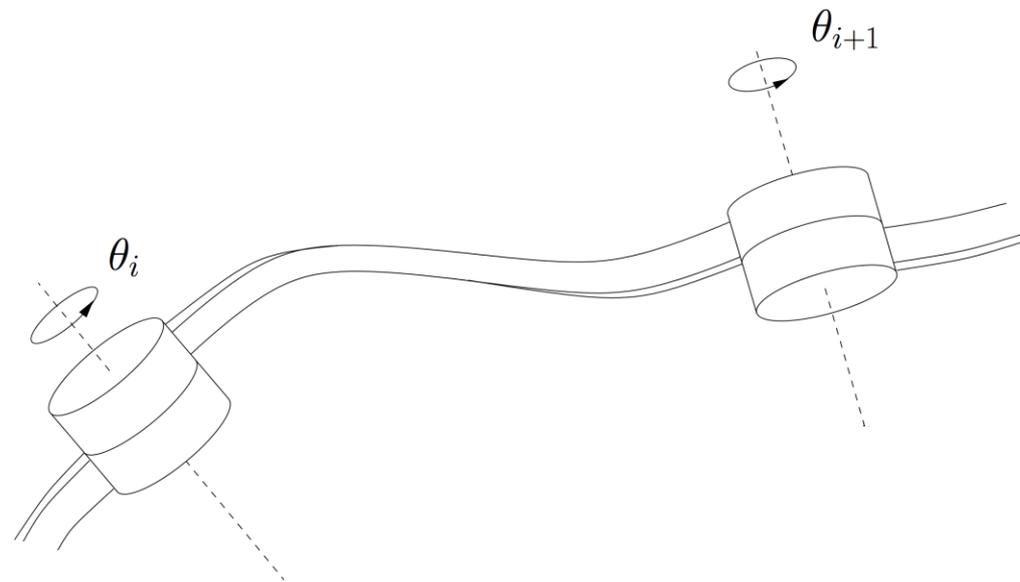
## Warum ist $SO(3)$ interessant für die Robotik?

- Mit Hilfe von  $SO(3)$ , kann eine **Objektpose** (z.B. Position und Orientierung) im Raum, so wie **Transformationen zwischen zwei Robotergelenkachsen** als **Verkettung einer Translation und Rotation** repräsentiert werden:

$$\varphi(\cdot): \mathbb{R}^3 \rightarrow \mathbb{R}^3, \quad \varphi(\mathbf{x}) = \mathbf{t} + R \cdot \mathbf{x}, \quad \mathbf{x}, \mathbf{t} \in \mathbb{R}^3, \quad R \in SO(3).$$

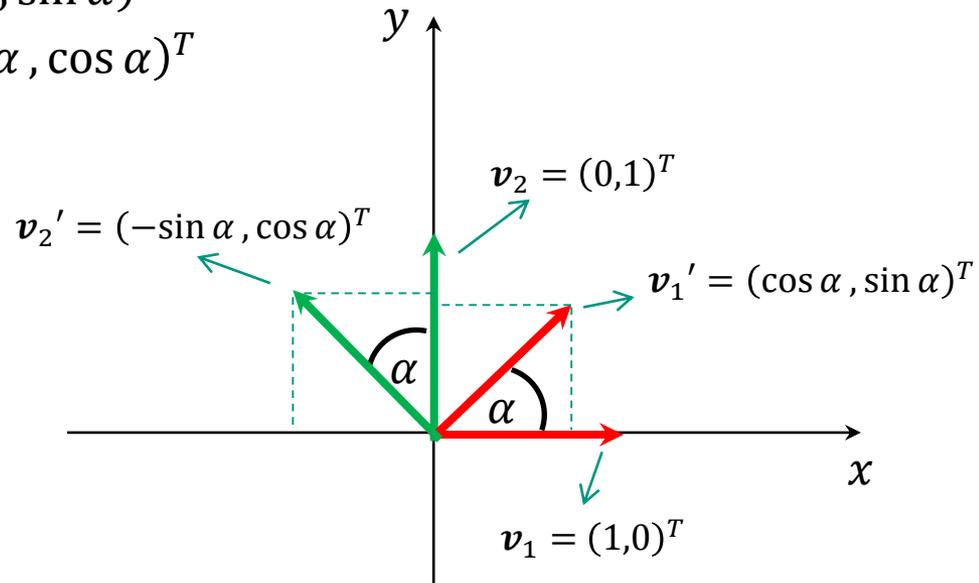
- Die Abbildung  $\varphi(\cdot)$  ist nicht linear! Sie wird **affin** genannt.

# Transformation zwischen zwei Robotergelenkachsen



# Rotationen in 2D

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



- Rotationsmatrix  $R_\alpha(x) = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \cdot x$
- mit  $RR^T = R^T R = I$ ,  $\det(R) = 1$

# Rotationen in 2D

- Eine **Rotation um** einen Punkt  $\mathbf{c} \neq (0,0)$  ist **keine lineare Transformation**. Sie transformiert  $(0,0)$  in einen anderen Punkt als  $(0,0)$ .
- **Vorgehen** bei der Rotation um ein beliebiges Rotationszentrum  $\mathbf{c}$ :
  - Wir **verschieben die Ebene um  $-\mathbf{c}$** , sodass das Rotationszentrum in  $(0,0)$  liegt.
  - Dann führen wir eine **Rotation um  $(0,0)$**  durch.
  - Danach schieben wir die Ebene **um  $+\mathbf{c}$  zurück**.

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

- $R_{\mathbf{c},\alpha}$  ist eine nichtlineare Transformation. Sie unterscheidet sich von  $R_{\alpha}$  nur durch das Addieren einer Konstante.
- Transformationen (wie  $R_{\mathbf{c},\alpha}$ ) der Form  $T(\mathbf{x}) = A(\mathbf{x}) + \mathbf{b}$  werden **affine Transformation** (oder auch affine Abbildung) genannt.

# Rotationen in 3D

- Eine 2D **Rotation in der  $xy$ -Ebene** ist in 3D eine **Rotation um die  $z$ -Achse**.
- Eine **Rotation** von Punkten **um  $z$  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 mit dem 2D Fall** und die Einträge zum Einfluss von  $z$  auf  $x$  sowie  $y$  und umgekehrt sind jeweils 0.

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

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

# Rotationen um Koordinatenachsen

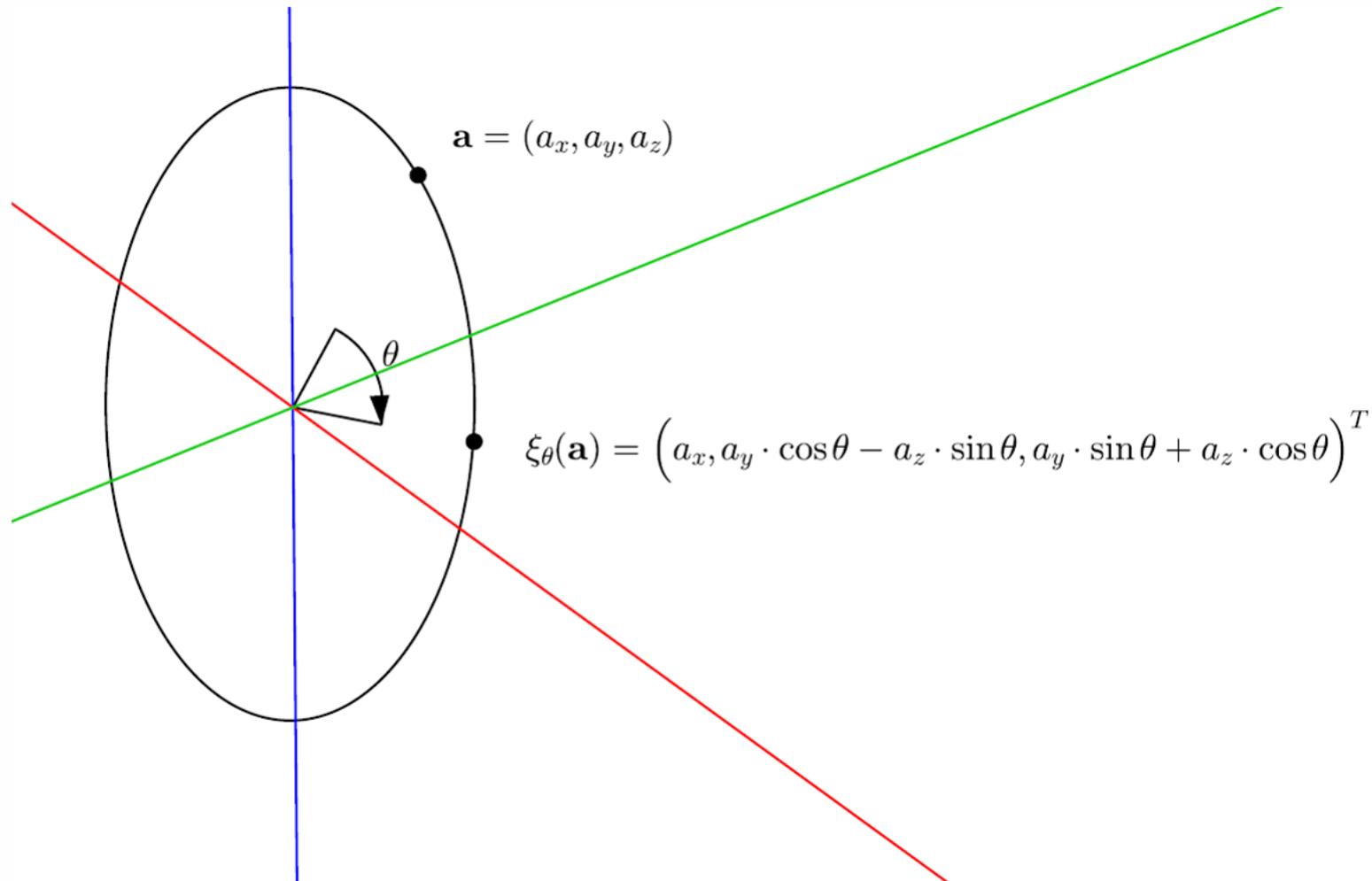
- Die **Rotationen** um die Achsen  $\mathbf{e}_x$ ,  $\mathbf{e}_y$ ,  $\mathbf{e}_z$  des Bezugssystems sind besondere Rotationen. Sie können durch die **Rotationsmatrizen**  $R_x$ ,  $R_y$  und  $R_z$  dargestellt werden.
- Eine Rotation um die Achse  $\mathbf{e}_x$

$$\xi_\theta(\mathbf{a}) = (a_x, a_y \cdot \cos\theta - a_z \cdot \sin\theta, a_y \cdot \sin\theta + a_z \cdot \cos\theta)^T$$

kann durch die folgende Rotationsmatrix dargestellt werden:

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

# Rotationen um Koordinatenachsen II



↗

## Rotationen um Rotationsachsen III

- Analog gilt für Rotationen um die Achsen  $\mathbf{e}_y$ ,  $\mathbf{e}_z$ :

$$R_{y,\theta} = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}, \quad R_{z,\theta} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

# Inverse Rotationsmatrix

- Die inverse Rotationsmatrix kann leicht berechnet werden (da sie der transponierten Rotationsmatrix entspricht):

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

# Verkettung von Rotationen

## ■ Die Verkettung von Rotationen

$$\phi_{z,\theta_3}(\phi_{y,\theta_2}(\phi_{x,\theta_1}(\mathbf{a}))), \quad \mathbf{a} \in \mathbb{R}^3$$

## ■ Wichtig: Es gibt **zwei Möglichkeiten** die obere Verkettung als **Matrixprodukt** zu interpretieren:

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

$$\left( (R_{z,\theta_3} \cdot R_{y',\theta_2}) \cdot R_{x'',\theta_1} \right) \cdot \mathbf{a} = R_{z,\theta_3} \cdot R_{y',\theta_2} \cdot R_{x'',\theta_1} \cdot \mathbf{a}$$

- Von **rechts nach links**: Rotationen werden um **globale Achsen** durchgeführt, die sich nicht ändern.

$$R_{z,\theta_3} \cdot \left( R_{y,\theta_2} \cdot (R_{x,\theta_1} \cdot \mathbf{a}) \right) = R_{z,\theta_3} \cdot R_{y,\theta_2} \cdot R_{x,\theta_1} \cdot \mathbf{a}$$

# Beispiel: Verkettung von Rotationen

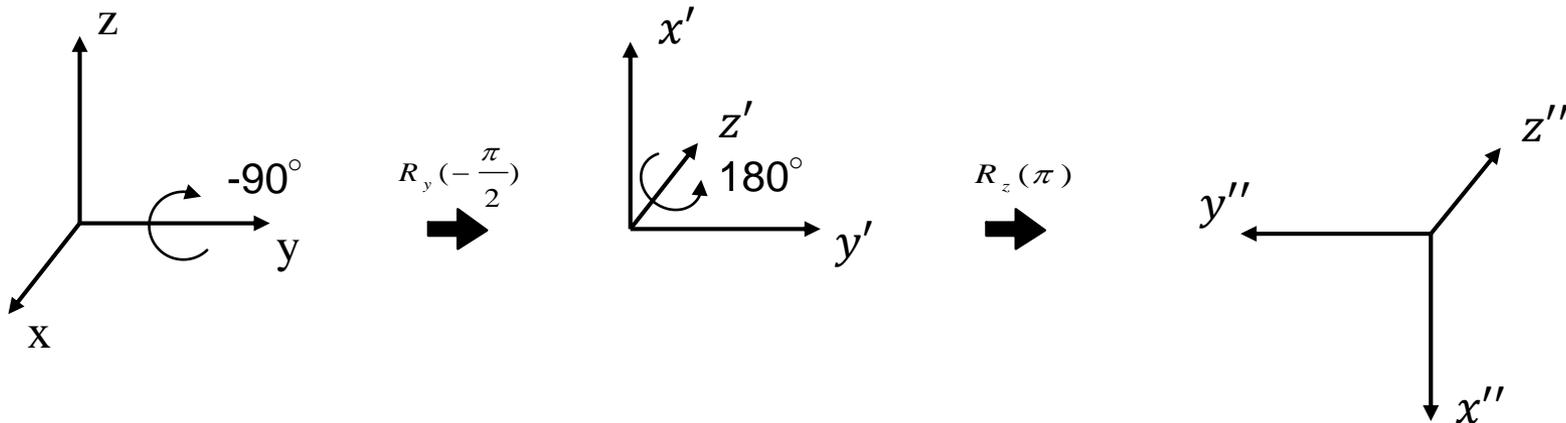
## ■ Verkettung der folgenden Rotationen:

- Rotation um y-Achse:  $-90^\circ$  ( $-\frac{\pi}{2}$ )

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

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

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



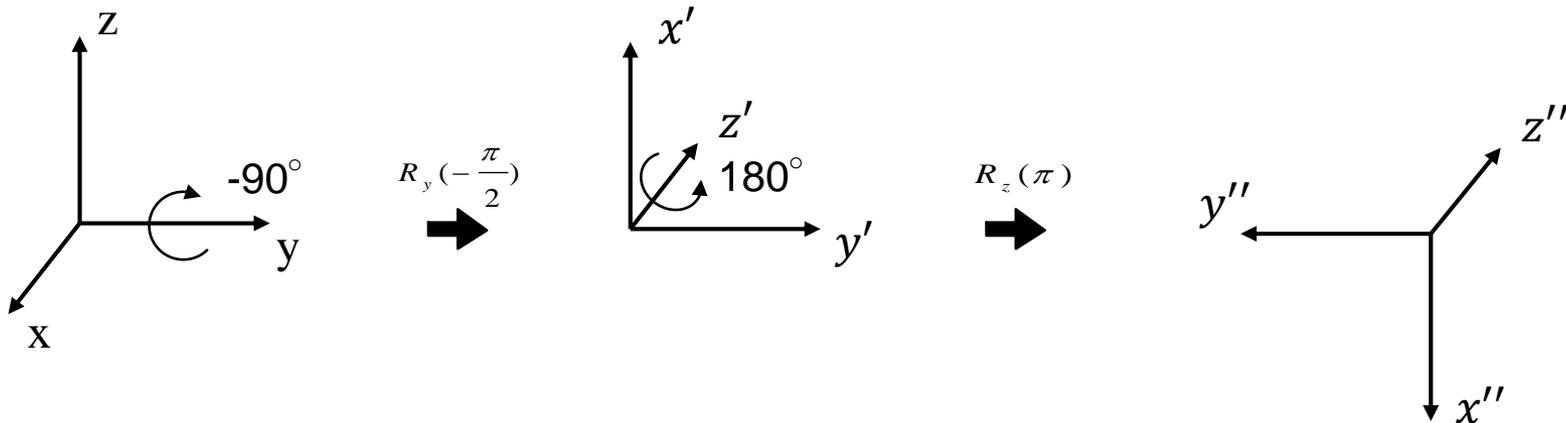
# Beispiel: Verkettung von Rotationen II

## ■ Berechnung der Rotationsmatrix

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

## ■ Transformation eines Vektors

$$u'' = \begin{bmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \cdot u = \begin{bmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} -u_3 \\ -u_2 \\ -u_1 \end{bmatrix}$$



# 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.
- Wie geht man mit diesen Problemen um?
  - Nutzung anderer Darstellungen für Rotationen, z.B. Eulerwinkel.

# Eulerwinkel

- Es ist möglich, jede Rotation durch **drei Rotationen** um drei Koordinatenachsen darzustellen.
- Die Achsen können willkürlich gewählt werden, aber aus historischen Gründen wird oft die sogenannte **Euler-Konvention  $z, x', z''$**  verwendet.
- Die Winkel  $\alpha, \beta$  und  $\gamma$  sind die Eulerwinkel. Sie beschreiben die Rotationsmatrix

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

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

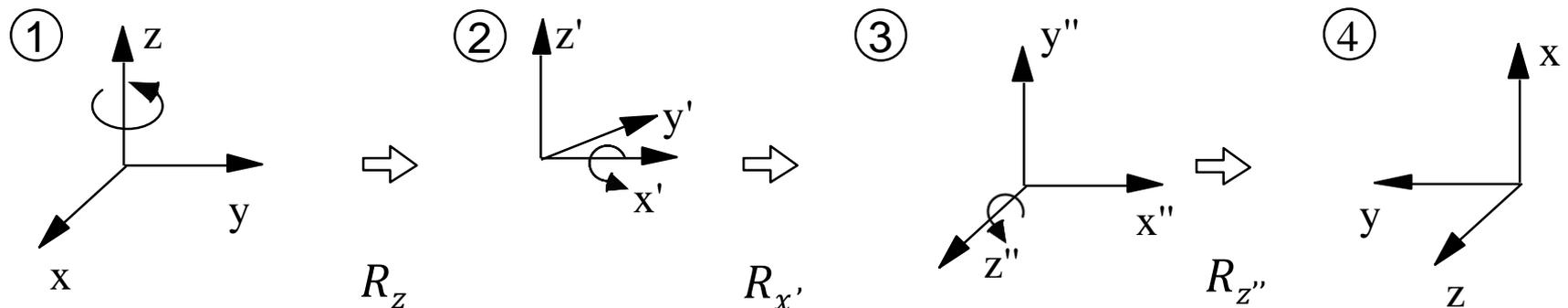
# Euler-Konvention $z, x', z''$

## Euler-Winkel

- Drehung um  $\alpha$  um die z-Achse des BKS  $R_z$
- Drehung um  $\beta$  um die neue x-Achse  $x'$   $R_{x'}$
- Drehung um  $\gamma$  um die neue z-Achse  $z''$   $R_{z''}$

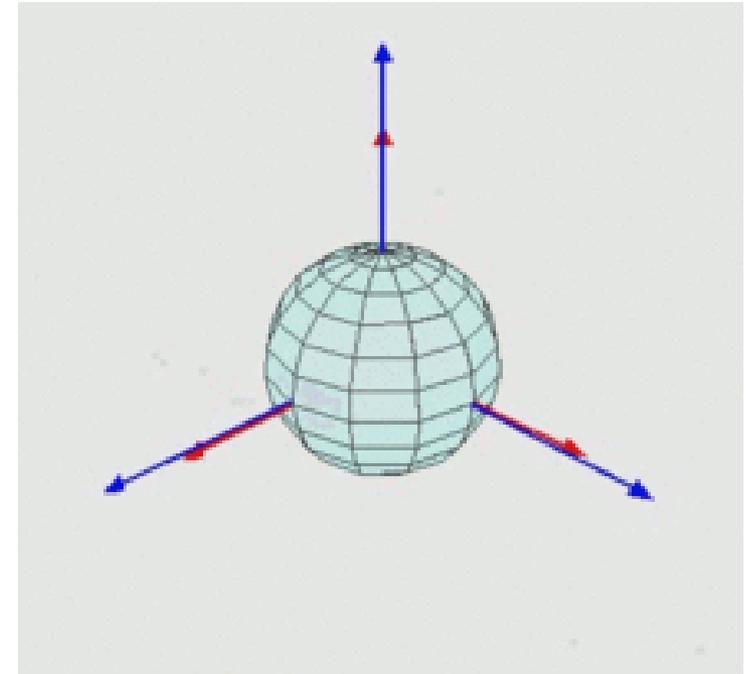
$$R_s = R_z R_{x'} R_{z''}$$

**Wichtig: Drehung um jeweils veränderte Achsen!**



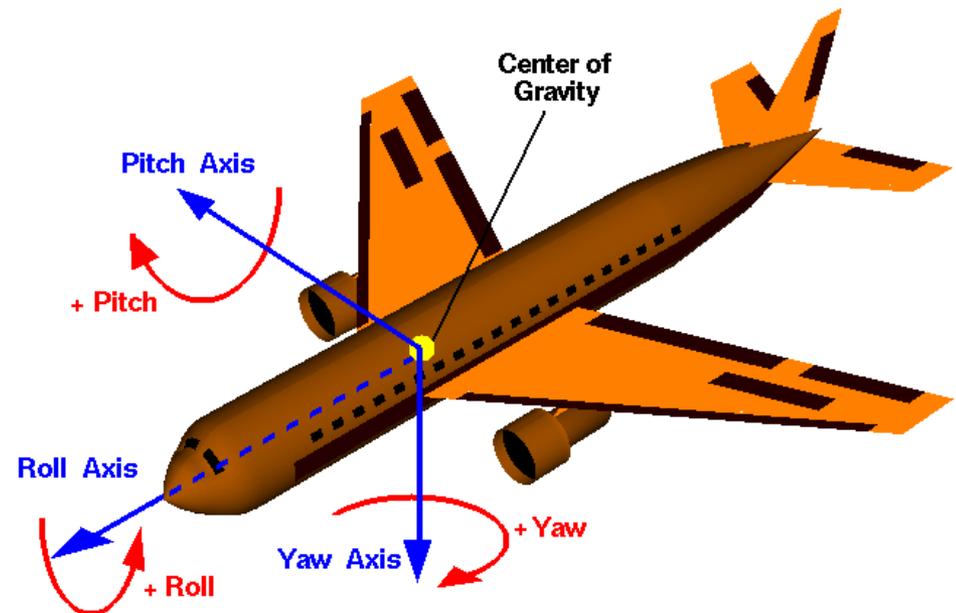
# Eulerwinkel II

- **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

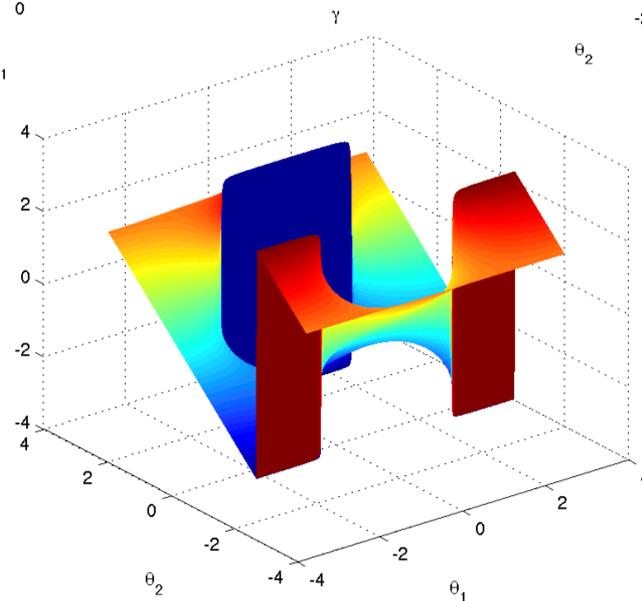
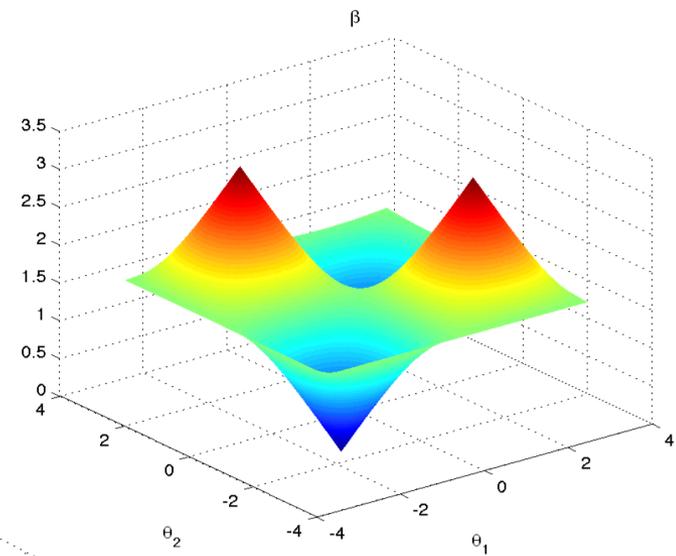
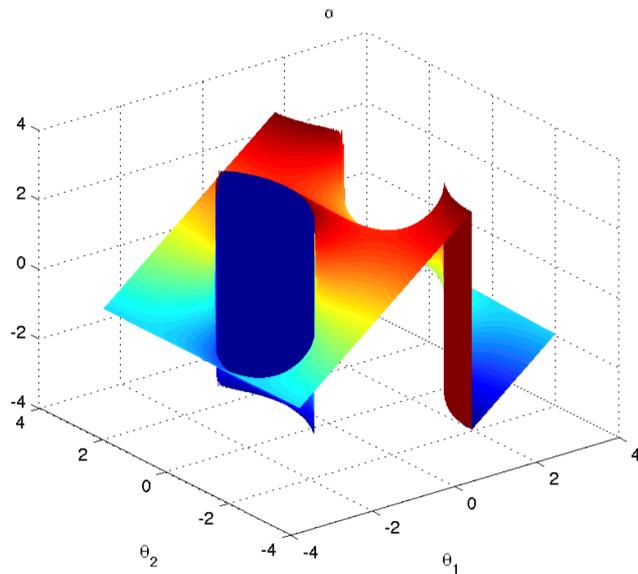
- Eine andere Konvention ist die **Euler-Konvention  $x, y, z$** .
- Diese speziellen Eulerwinkel werden **Roll, Pitch, Yaw** genannt (im Deutschen: Roll-Nick-Gier-Winkel).
- Abfolge der Rotationen
  1.  $x$ -Achse des BKS um  $\alpha$  (Roll)
  2.  $y$ -Achse des BKS um  $\beta$  (Pitch)
  3.  $z$ -Achse des BKS um  $\gamma$  (Yaw)



# Eulerwinkel III

- Vorteile der Eulerwinkel:
  - **Kompakter** als Rotationsmatrizen
  - **Aussagekräftiger** als Rotationsmatrizen
- Nachteil der Eulerwinkel:
  - **Nicht eindeutig:**
    - Beispiel: in der Euler  $z, x', z''$  Konvention, beschreiben die Eulerwinkel  $(45^\circ, 30^\circ, -45^\circ)$  und  $(0^\circ, 30^\circ, -0^\circ)$  die gleiche Rotation!
  - **Nicht kontinuierlich:**
    - 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

# Eulerwinkel: Interpolationsproblem



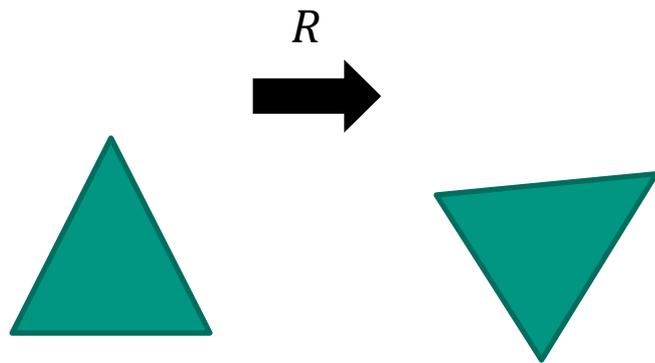
Mathematische Grundlagen I

# FRAGEN

# Unterschied $SO(3)$ und $SE(3)$

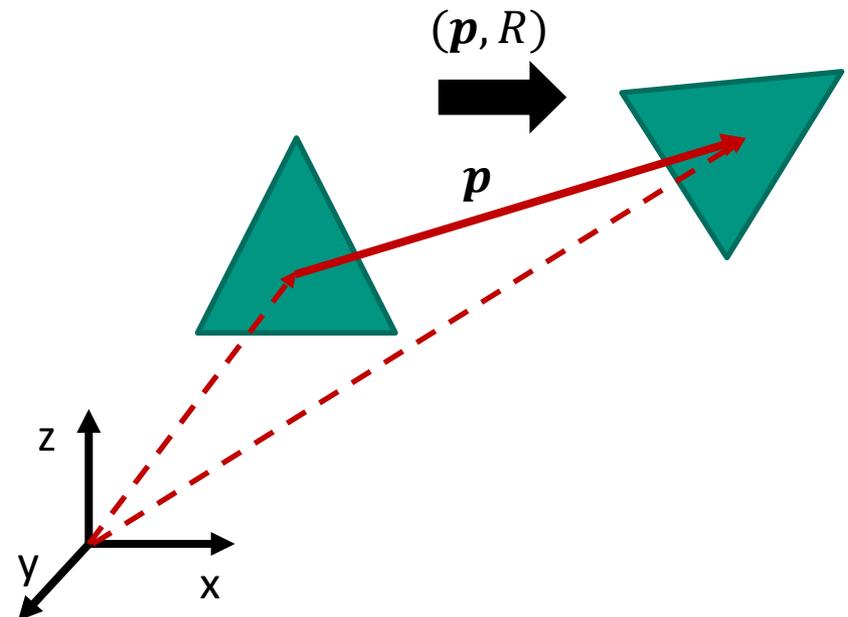
## ■ $SO(3)$

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



## ■ $SE(3)$

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



# Rotationen um lokale Achsen

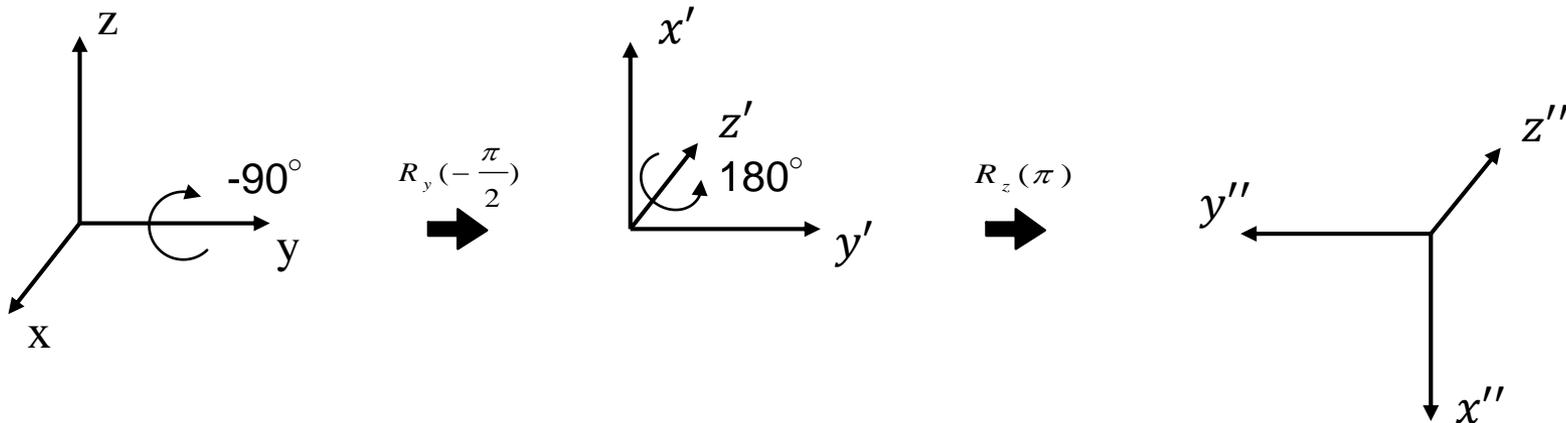
## ■ Berechnung der Rotationsmatrix

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

## ■ Transformation eines Vektors

$$u'' = \begin{bmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \cdot u = \begin{bmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} -u_3 \\ -u_2 \\ -u_1 \end{bmatrix}$$

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



# Robotik I: Einführung in die Robotik

## Mathematische Grundlagen II

**Tamim Asfour**

Institut für Anthropomatik und Robotik

KIT-Fakultät für Informatik, Institut für Anthropomatik und Robotik (IAR)  
Hochperformante Humanoide Technologien (H<sup>2</sup>T)



# Eulerwinkel – Gimbal Lock

■ **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

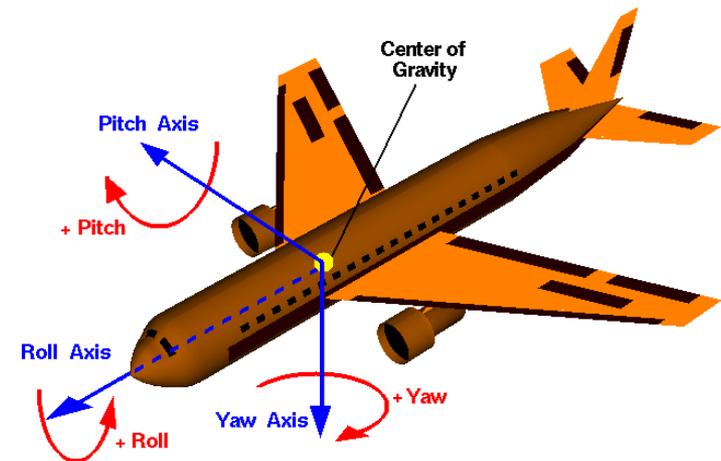
■ x-y-z, y-z-x, z-x-y, x-z-y, z-y-x, y-x-z

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

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

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

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



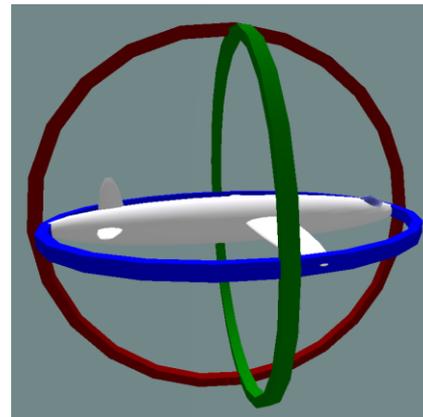
# Eulerwinkel – Gimbal Lock

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

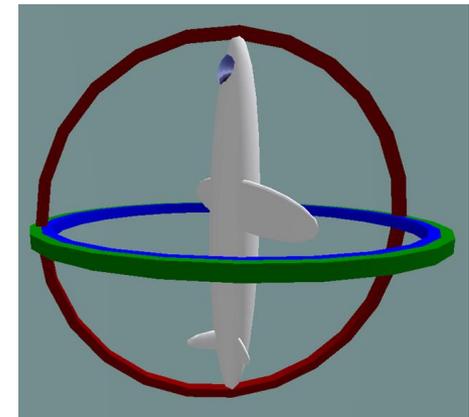


## ■ Gimbal Lock

- Bei bestimmten Winkeln werden zwei Achsen voneinander abhängig
- Ein Freiheitsgrad geht verloren



3 DoF



2 DoF

# Eulerwinkel – Gimbal Lock

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

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

- Multiplikation der Matrizen:

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

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

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

 Gemeinsame Drehachse für Drehung um  $\alpha$  und  $\gamma$ , 1 DOF geht verloren

# Rotationsmatrix vs. Eulerwinkel

## Rotationsmatrizen

- Redundanz durch 9 Werte

## Eulerwinkel

- Kompakter
- Aussagekräftiger
- Nicht eindeutig
- Gimbal Lock
- Nicht kontinuierlich

# Eulerwinkel vs. Roll-Pitch-Yaw

## Eulerwinkel ( $z, x', z''$ )

- Multiplikation von links nach rechts  
 $R_S = R_z(\alpha) R_{x'}(\beta) R_{z''}(\gamma)$
- Jede Drehung bezieht sich auf das neue Koordinatensystem
- Drehung um jeweils veränderte Achsen

## Roll-Pitch-Yaw ( $x, y, z$ )

- Multiplikation von rechts nach links  
 $R_S = R_z(\gamma) R_y(\beta) R_x(\alpha)$
- Jede Drehung bezieht sich auf das BKS
- Drehung jeweils um unveränderte Achsen

# Orientierungsbeschreibung mit 3x3-Matrizen

## Bewertung

- Vorteil: Vektor und Rotationsmatrix sind anschaulich und daher eine übliche Form der Eingabe von Objekt- und Endeffektorpose
- Nachteil: Vektor- und Matrixoperationen werden getrennt durchgeführt  
( $\mathbf{p}, R$ ) mit  $\mathbf{p} \in \mathbb{R}^3$  und  $R \in SO(3) \subset \mathbb{R}^{3 \times 3}$

## 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** (oder **homogenen**) **Koordinaten** ausgedrückt werden:

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

- 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:

$$\mathbf{b} = A\mathbf{x} + \mathbf{t}$$

$$\Leftrightarrow$$

$$\mathbf{b} = \begin{pmatrix} \mathbf{b} \\ 1 \end{pmatrix} = \begin{pmatrix} A & \mathbf{o} \\ \mathbf{o}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} + \begin{pmatrix} \mathbf{t} \\ 0 \end{pmatrix} = \begin{pmatrix} A & \mathbf{t} \\ \mathbf{o}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$$

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

# Affine Transformationen II

## ■ Vorteile:

- Es wird ermöglicht, **Rotationen um beliebige Achsen** im affinen Raum zu formulieren.
- **Rotationen und Translationen** können in einer **einzelnen homogenen Matrix** kombiniert werden. Das heißt, Rotationen und Translationen können einheitlich abgehandelt werden.

# Homogene 4x4-Matrizen

- Allgemeine homogene 4x4 Matrix

$$T = \begin{pmatrix} A & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix}$$

- Translationsmatrix

Verschiebung des Objekt-Koordinatensystem (OKS) nach  $(t_x, t_y, t_z)^T$  im Basis-Koordinatensystem (BKS)

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

# Homogene 4x4-Matrizen

## ■ 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 4x4 Matrizen

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

$$\mathbf{a} = (4,3,2,1)^T$$

$$\mathbf{b} = (6,2,4,1)^T$$

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

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

## Homogene 4x4-Matrizen III

- Abbildung des Ortsvektors  $p_{OKS}$  im OKS ins BKS:

$$p_{BKS} = T p_{OKS}$$

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

- $u$ : Ursprung des OKS
- $n, o, a$ : Einheitsvektoren des OKS bezüglich BKS

# Homogene 4x4-Matrizen IV

- Invertierung  $R^{-1} = R^T$

$$T^{-1} = \begin{pmatrix} n_x & n_y & n_z & -n^T \mathbf{u} \\ o_x & o_y & o_z & -o^T \mathbf{u} \\ a_x & a_y & a_z & -a^T \mathbf{u} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} & & & -n^T \mathbf{u} \\ & R_{3 \times 3}^T & & -o^T \mathbf{u} \\ & & & -a^T \mathbf{u} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

# Homogene 4x4-Matrix: Eigenschaften

- Eine homogene 4x4 Matrix enthält 12  $(\mathbf{n}, \mathbf{o}, \mathbf{a}, \mathbf{u})$  nichttriviale Kenngrößen im Gegensatz zu 6  $(x, y, z, \alpha, \beta, \gamma)$  notwendigen
- Redundanz wegen Orthogonalität  $(RR^T = I)$
- Drehachsen und Drehreihenfolge sind implizit enthalten

# Vergleich: kartesische und homogene Darstellung

- In kartesischen Koordinaten

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$

- In homogenen Koordinaten

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & v_x \\ n_y & o_y & a_y & v_y \\ n_z & o_z & a_z & v_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix}$$

# Koordinatensysteme

Interpretationen von homogenen 4x4 Matrizen

- Lagebeschreibung eines Koordinatensystems:

$${}^A H_B$$

beschreibt die Lage des Koordinatensystems  $B$  relativ zum Koordinatensystem  $A$

- Transformationsabbildung:

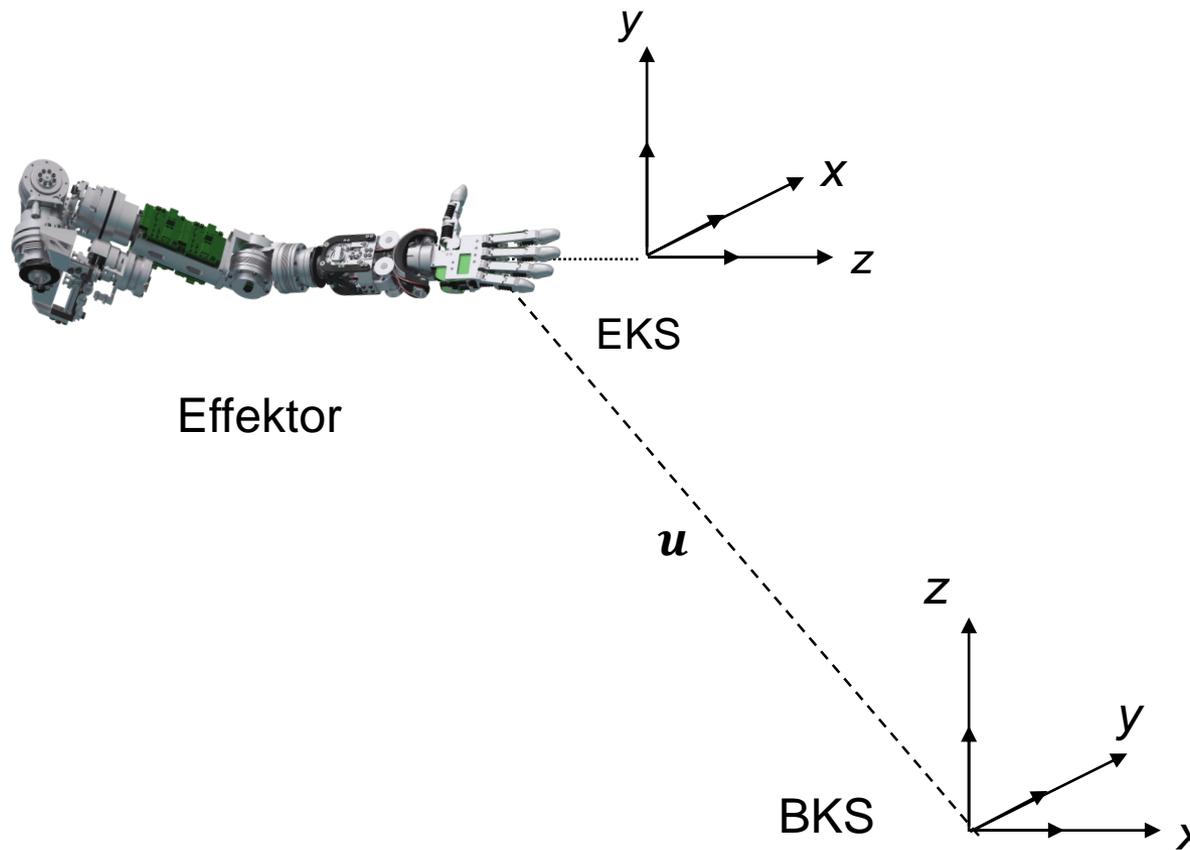
$${}^A H_B : {}^B P \mapsto {}^A P$$

- Transformationsoperator

$$H : {}^A P_1 \mapsto {}^A P_2$$

# Beispiel: Koordinatensystem-Transformation

- Gegeben: Punkt  ${}^{EKS}P = (0, -3, 5)^T$
- Gesucht:  ${}^{BKS}P$



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

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

# Beispiel: Koordinatensystem-Transformation

- Gegeben: Punkt  ${}^{EKS}P = (0, -3, 5)^T$
- Gesucht:  ${}^{BKS}P$

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

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

# Verkettete Lagebeschreibung

- Sei  ${}^{BKS}H_A$  die Lagebeschreibung des Objekts A bzgl. BKS
- Sei  ${}^AH_B$  die Lagebeschreibung eines Objekts B, bezogen auf das OKS von A
- Sei  ${}^{BKS}H_B$  die Lagebeschreibung des Objekts B bzgl. BKS

So gilt

$${}^{BKS}H_B = {}^{BKS}H_A {}^AH_B$$

- Im Vergleich zur kartesischen Darstellung kompaktere Schreibweise:

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

## Verkettete Lagebeschreibung II

- Lage von Objekt 1 bzgl. BKS:  ${}^{BKS}H_{01}$
- Lage von Objekt 2 bzgl. Objekt 1:  ${}^{01}H_{02}$
- Lage von Objekt 3 bzgl. Objekt 2:  ${}^{02}H_{03}$
- Lage von Objekt 3 bzgl. BKS:  ${}^{BKS}H_{03}$

$${}^{BKS}H_{03} = {}^{BKS}H_{01} \cdot {}^{01}H_{02} \cdot {}^{02}H_{03}$$

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

$$\prod_{i=1}^n A_{i-1} H_{A_i} \quad \text{mit } A_0 = BKS$$

# Quaternionen

- Probleme mit Rotationsmatrizen:
  - Redundant
  - Schwierige Interpolation
- Probleme mit Eulerwinkeln:
  - Numerisch instabil
  - Gimbal Lock
- Existiert eine andere Repräsentation ohne diese Nachteile?
- **Antwort:** Quaternionen!
  - Quaternionen sind eine Erweiterung der Komplex Zahlen  $\mathbb{C}$  („Hyperkomplexe Zahlen“)
  - Geprägt 1843 durch William Rowan Hamilton
  - *Closed-form solution of absolute orientation using unit quaternions*, Horn 1987

# Quaternionen: Definition

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

$$\mathbb{H} = \mathbb{C} + \mathbb{C}j$$

$$\text{mit } j^2 = -1 \text{ und } i \cdot j = -j \cdot i.$$

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

$$\mathbf{q} = (a, \mathbf{u})^T = a + u_1i + u_2j + u_3k$$

mit  $a \in \mathbb{R}$ ,  $\mathbf{u} \in \mathbb{R}^3$  und  $k = i \cdot j$ .

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

# Quaternionen: Rechenregeln I

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

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

- Addition:

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

- 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 II

■ Quaternion:

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

■ Konjugiertes Quaternion:

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

■ Norm:

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

■ Inverse:

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

# Quaternionen: Rotationen

- Beschreibung eines Vektor  $\mathbf{p}$  als Quaternion  $\mathbf{q}$

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

- Skalar  $s$  als Quaternion  $\mathbf{q}$

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

- Einheitsquaternionen  $\mathbb{S}^3$  :

- Es existiert eine Einbettung von  $SO(3) \subset \mathbb{R}^3$  nach  $\mathbb{H}$
- Rotationen sind definiert durch Einheitsquaternionen  $\mathbb{S}^3$
- Bilden eine Gruppe  $\mathbb{S}^3 = \{ \mathbf{q} \in \mathbb{H} \mid \|\mathbf{q}\|^2 = 1 \}$ 
  - Gruppeneigenschaften:  
z.B. Existenz eines inversen Elements zu jedem Gruppenelement

## Quaternionen: Rotationen II

■ Sei eine Rotation beschrieben durch

- Eine Drehachse  $\mathbf{a}$  mit  $\|\mathbf{a}\| = 1$
- und ein Drehwinkel  $\phi$ ,

dann existiert hierfür eine Repräsentation als Quaternion

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

■ Ein Punkt  $\mathbf{v}$  wird mit einem Quaternion  $\mathbf{q}$  rotiert durch :

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

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

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

## Quaternionen: Rotationen III

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

$$\boldsymbol{q} = \left( \cos \frac{\phi_q}{2}, \boldsymbol{u}_q \sin \frac{\phi_q}{2} \right), \quad \boldsymbol{r} = \left( \cos \frac{\phi_r}{2}, \boldsymbol{u}_r \sin \frac{\phi_r}{2} \right)$$

- Rotation mit einem Quaternion:

$$f(\boldsymbol{v}) = \boldsymbol{q}\boldsymbol{v}\boldsymbol{q}^*, \quad h(\boldsymbol{v}) = \boldsymbol{r}\boldsymbol{v}\boldsymbol{r}^*$$

- Dann entspricht  $f \circ h$  gerade der Rotation mit dem Quaternion  $\boldsymbol{p} = \boldsymbol{q} \cdot \boldsymbol{r}$

# Quaternionen: Rotationen IV

- Beispiel: Rotation des Punkts um die Drehachse mit Winkel

$$\mathbf{p} = (1, 0, 9)^T$$

$$\mathbf{a} = (1, 0, 0)^T$$

$$\phi = 90^\circ$$

1. Darstellung von  $\mathbf{p}$  als Quaternion  $\mathbf{v}$

$$\mathbf{v} = 0 + 1i + 0j + 9k$$

2. Rotationsquaternion  $\mathbf{q}$

$$\mathbf{q} = \cos \frac{\phi}{2} + 1i \cdot \sin \frac{\phi}{2} + 0j + 0k$$

3. Konjugiertes Quaternion  $\mathbf{q}^*$

$$\mathbf{q}^* = \cos \frac{\phi}{2} - 1i \cdot \sin \frac{\phi}{2} - 0j - 0k$$

4. Rotation von  $\mathbf{v}$  um  $\mathbf{q}$

$$\mathbf{v}_r = \mathbf{q}\mathbf{v}\mathbf{q}^*$$

$$\mathbf{v}_r = 0 + 1i - 9j + 0k$$

5. Darstellung als Punkt  $\mathbf{p}_r$

$$\mathbf{p}_r = (1, -9, 0)^T$$

- **Achtung: Die Multiplikation als Quaternion ist NICHT kommutativ.**

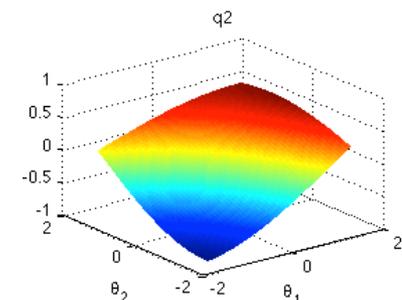
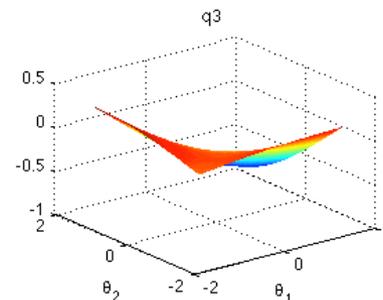
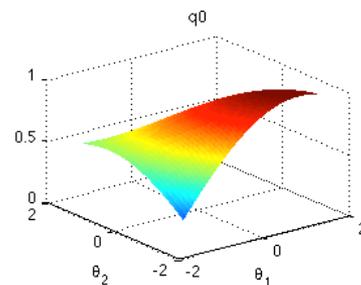
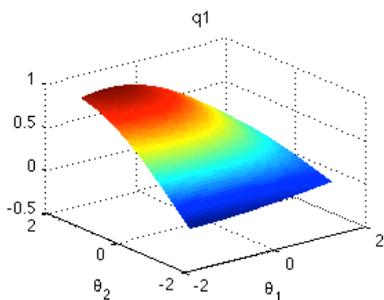
# Quaternionen: Bewertung

## ■ Vorteile:

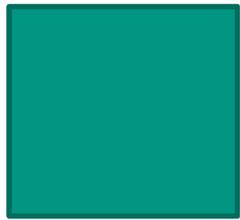
- Kompakte Darstellung: 4 Werte im Vergleich zu 9 bei Rotationsmatrizen
- Anschaulich (angelehnt an Axis-Winkel Repräsentation)
- Konkatenation möglich, ähnlich wie bei Rotationsmatrix
- Kann für Berechnung der Inversen Kinematik verwendet werden (späteres Kapitel)
- Kein Gimbal Lock
- Repräsentation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

## ■ Nachteile:

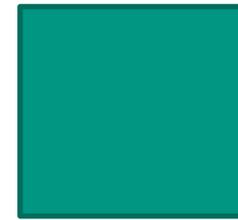
- Nur Beschreibung von Rotation, keine Translation



# Quaternionen: Interpolation



$t = 0$



$t = 1$

- **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:  $a \cdot (1 - t) + b \cdot t$

# Quaternionen: SLERP

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

## ■ SLERP Interpolation:

- $Slerp(\mathbf{q}_1, \mathbf{q}_2, t) = \mathbf{q}_1 \cdot (\mathbf{q}_1^{-1} \cdot \mathbf{q}_2)^t$
- Potenzieren von Quaternionen wird nicht in der Vorlesung behandelt

## ■ Direkte Formulierung der SLERP Interpolation:

- $Slerp(\mathbf{q}_1, \mathbf{q}_2, t) = \frac{\sin(1-t)\theta}{\sin \theta} \cdot \mathbf{q}_1 + \frac{\sin t\theta}{\sin \theta} \cdot \mathbf{q}_2$
- Mit  $\langle \mathbf{q}_1 | \mathbf{q}_2 \rangle = \cos \theta$

## ■ Ergebnis:

- Rotation mit konstanter Winkelgeschwindigkeit

# Quaternionen: Interpolationsprobleme

- **Problem:** Orientierungen in  $SO(3)$  werden durch Einheitsquaternionen doppelt abgedeckt.

- **Zeige:** Die Einheitsquaternionen  $q$  und  $-q$  entsprechen der gleichen Rotation.

## Beweis:

- Rotation von  $v$  um  $q$  entspricht Rotation von  $v$  um  $-q$ .
  - $v_r = qvq^* = (-q)v(-q)^*$
  - Die negativen Vorzeichen löschen sich aus.
- SLERP berechnet deshalb nicht immer die kürzeste Rotation
    - Es muss geprüft werden, ob die Rotation von  $q_1$  zu  $q_2$  oder  $-q_1$  zu  $q_2$  kürzer ist

# Duale Quaternionen

## ■ Warum ?

- Reelle Quaternionen eignen sich für die Beschreibung der Orientierung, nicht aber zur Beschreibung der Lage eines Objektes (Translation fehlt).

## ■ Idee:

- Ersetze die 4 reellen Werte eines Quaternion durch Dualzahlen
- Erhalte so zusätzliche translatorische Komponente um die Lage eines Objekts ausdrücken zu können

# Duale Quaternionen

## Duale Zahlen

- Duale Zahlen sind von der Form

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

- 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

## Beschreibung

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

- Der reale Skalarteil enthält den **Winkelwert**  $\theta/2$  und der imaginäre Skalarteil die **Translationssgröße**  $d$
- Die restliche drei Dualzahlen beschreiben eine **beliebige gerichtete, normierte Gerade** im Raum, bzgl. der die Rotation und Translation erfolgen

# Duale Quaternionen II

Multiplikationstabelle für duale Einheitsquaternionen

$\cdot$	$1$	$i$	$j$	$k$	$\varepsilon$	$\varepsilon i$	$\varepsilon j$	$\varepsilon k$
$1$	$1$	$i$	$j$	$k$	$\varepsilon$	$\varepsilon i$	$\varepsilon j$	$\varepsilon k$
$i$	$i$	$-1$	$k$	$-j$	$\varepsilon i$	$-\varepsilon$	$\varepsilon k$	$-\varepsilon j$
$j$	$j$	$-k$	$-1$	$i$	$\varepsilon j$	$-\varepsilon k$	$-\varepsilon$	$\varepsilon i$
$k$	$k$	$j$	$-i$	$-1$	$\varepsilon k$	$\varepsilon j$	$-\varepsilon i$	$-\varepsilon$
$\varepsilon$	$\varepsilon$	$\varepsilon i$	$\varepsilon j$	$\varepsilon k$	$0$	$0$	$0$	$0$
$\varepsilon i$	$\varepsilon i$	$-\varepsilon$	$\varepsilon k$	$-\varepsilon j$	$0$	$0$	$0$	$0$
$\varepsilon j$	$\varepsilon j$	$-\varepsilon k$	$-\varepsilon$	$\varepsilon i$	$0$	$0$	$0$	$0$
$\varepsilon k$	$\varepsilon k$	$\varepsilon j$	$-\varepsilon i$	$-\varepsilon$	$0$	$0$	$0$	$0$

## Duale Quaternionen III

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

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

- 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  $T$ :

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

## Duale Quaternionen IV

- Eine Transformation  $T$  mit einem rotations Teil  $r$  und einen translatorischen Teil  $t$ , kann als duales Quaternionen beschrieben werden durch:

$$q_T = q_t q_r$$

- Eine Transformation  $q_T$  wird auf einen Punkt  $p$  (als duales Quaternion) wie folgt angewendet:

$$p' = q_T p q_T^*, \text{ mit } q_T^* = (q_t q_r)^* = q_r^* q_t^*$$

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

$$q^* = p^* - \varepsilon \cdot s^*$$

## Duale Quaternionen: Beispiel

- Beispiel: Rotation eines Punkts um Drehachse und Translation mit
 
$$\mathbf{p} = (3,4,5)^T$$

$$\mathbf{a} = (1,0,0)^T \text{ mit } \phi = 180^\circ$$

$$\mathbf{p}_t = (4,2,6)^T$$

- $\mathbf{p}$  als duales Quaternion  $\mathbf{v}_d$

$$\mathbf{v}_d = 1 + 3\varepsilon i + 4\varepsilon j + 5\varepsilon k$$

- Rotation als duales Quaternion  $\mathbf{q}_r$

$$\mathbf{q}_r = \cos \frac{\phi}{2} + 1i \cdot \sin \frac{\phi}{2} + 0j + 0k = i$$

- Translation als duales Quaternion  $\mathbf{q}_t$

$$\mathbf{q}_t = 1 + 2\varepsilon i + 1\varepsilon j + 3\varepsilon k$$

- Kombination als duales 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

- Beispiel: Rotation eines Punkts  
 um Drehachse  
 und Translation mit
 
$$\mathbf{p} = (3,4,5)^T$$

$$\mathbf{a} = (1,0,0)^T \text{ mit } \phi = 180^\circ$$

$$\mathbf{p}_t = (4,2,6)^T$$

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

- Probe:

- Rotation um die x-Achse mit  $\phi = 180^\circ$

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

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

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

# 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 Rotation und Translation im Euklidischen Raum  $\mathbb{E}$ 
  - Rotationsmatrix und Verschiebungsvektor
  - Eulerwinkel
  - Homogene  $4 \times 4$  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