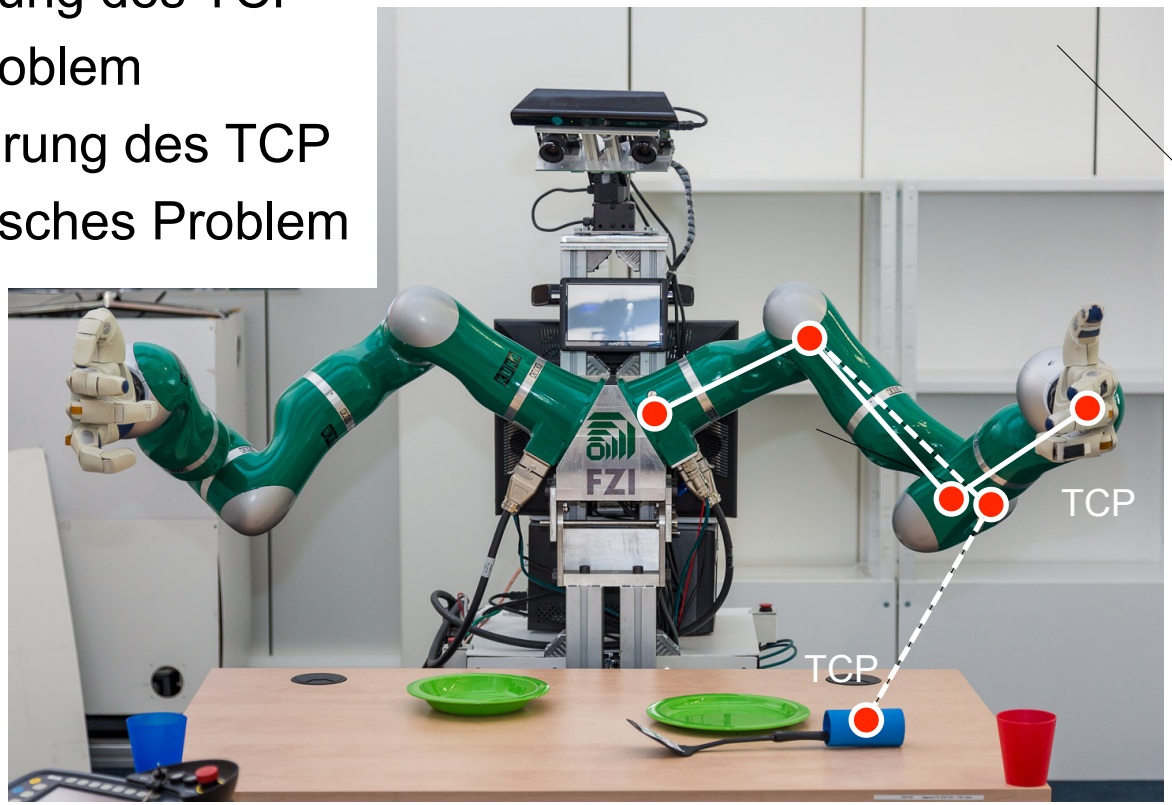
Karlsruhe Institute of Technology

Mathematische Grundlagen

Prof. Dr.-Ing. Rüdiger Dillmann

Motivation der heutigen Vorlesung

- **Ziel:** Bewegen der Arme
- **Kinematik** ist die Lehre der geom. und analyt. Beschreibung der Bewegungszustände mechanischer Systeme
- Problem 1: Lokalisierung des TCP
  - Kinematisches Problem
- Problem 2: Positionierung des TCP
  - Inverses kinematisches Problem
  - Bahnplanung



Inhalt

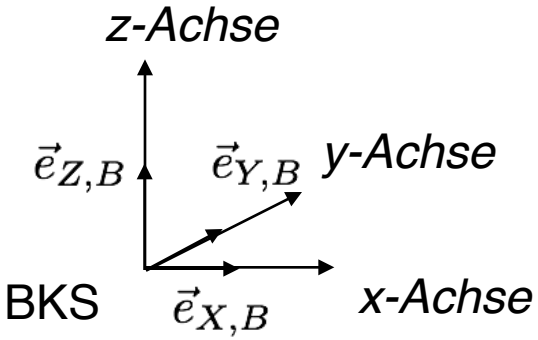
- Beschreibung von Objekten und Objektlagen im 3-dim., euklidischen Raum
- Orientierungsbeschreibung mit 3x3-Matrix
- Homogene 4x4-Matrix
- Rotation und Translation von Punkten

- Verkettete Lagebeschreibung
- Duale 3x3-Matrix
- Quaternionen
- Duale Quaternionen

Objekte und Objektlagen im euklid. 3D-Raum

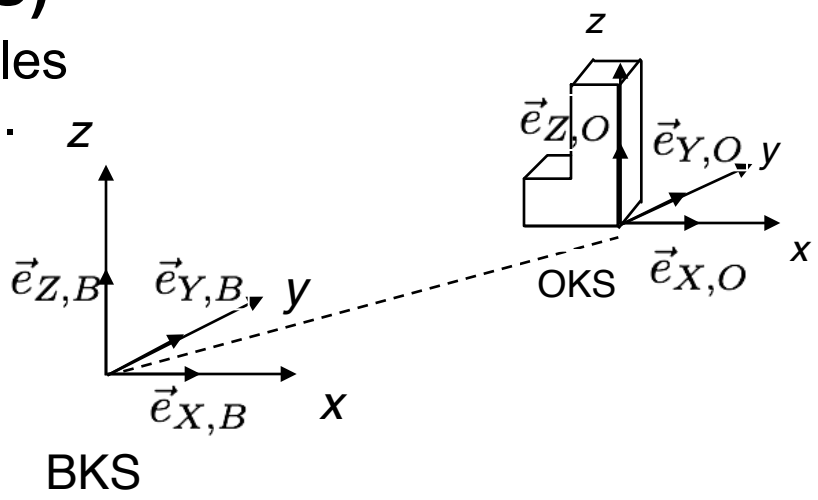
Basiskoordinatensystem (BKS)

- ein 3-dim. Koordinatensystem wird durch orthogonale Einheitsvektoren $\vec{e}_{X,B}, \vec{e}_{Y,B}, \vec{e}_{Z,B}$ definiert



Objektkoordinatensystem (OKS)

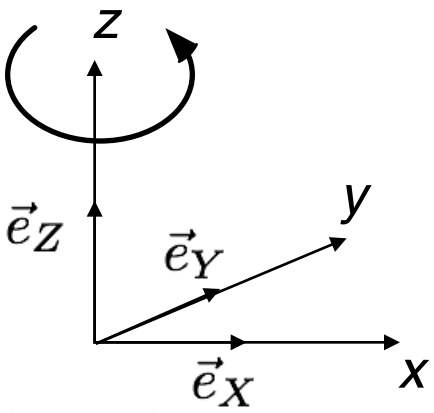
- jeder starre Körper kann auf ein lokales Koordinatensystem bezogen werden. Das lokale Koordinatensystem wird durch orthogonale Einheitsvektoren $\vec{e}_{X,O}, \vec{e}_{Y,O}, \vec{e}_{Z,O}$ definiert



Objekte und Objektlagen im euklid. 3D-Raum

Linksdrehende und rechtsdrehende Koordinatensysteme

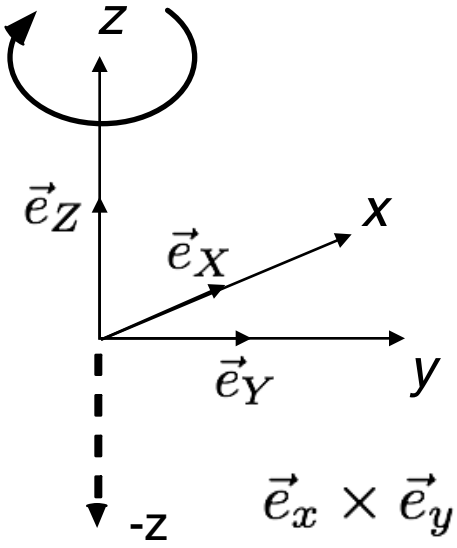
- Man unterscheidet linksdrehende und rechtsdrehende Koordinatensysteme. Der Drehsinn wird per Definition festgelegt



$$\vec{e}_x \times \vec{e}_y = \vec{e}_z$$
$$\vec{x} \times \vec{y} = \vec{z}$$

rechtsdrehendes Koordinatensystem

x = Kreuzprodukt

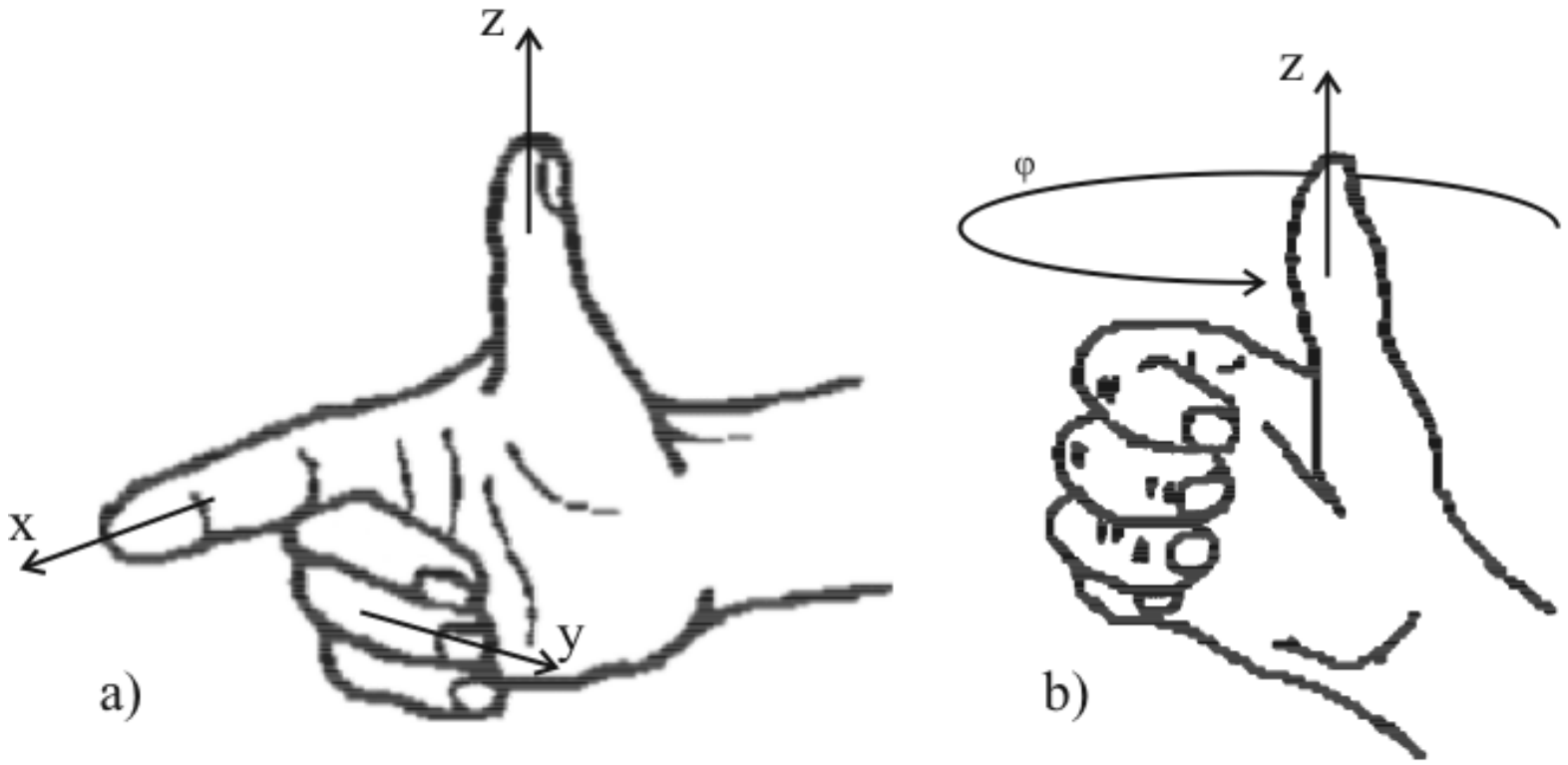


$$\vec{e}_x \times \vec{e}_y = -\vec{e}_z$$
$$\vec{x} \times \vec{y} = -\vec{z}$$

linksdrehendes Koordinatensystem

Objekte und Objektlagen im euklid. 3D-Raum

Rechte-Hand-Regel

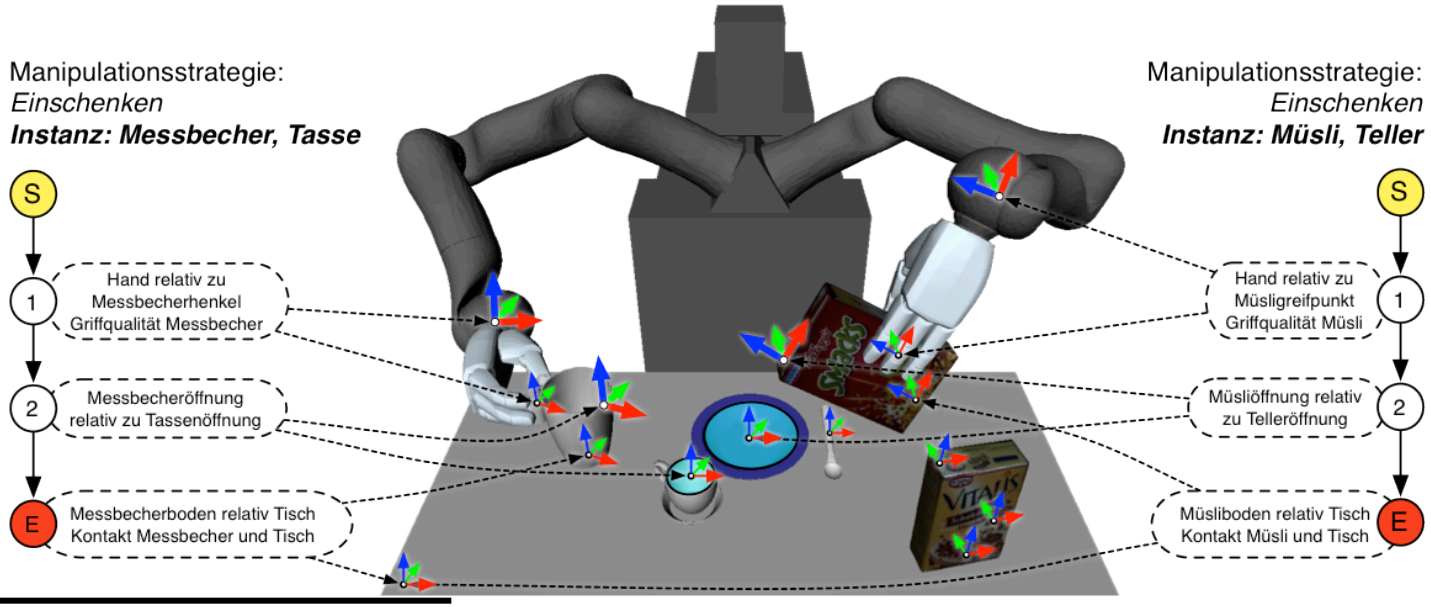


Objekte und Objektlagen im euklid. 3D-Raum

Verwendung mehrerer Koordinatensysteme

- Zur Beschreibung von Roboteranwendungen werden neben dem BKS zahlreiche lokale Koordinatensysteme verwendet:
  - OKS: Objektkoordinatensystem
  - EKS: Effektorkoordinatensystem (TCP)
  - SKS: Sensorkoordinatensystem
  - ...

Bsp.: Handlungs-beschreibung (Robotik II)



Objekte und Objektlagen im euklid. 3D-Raum

Definitionen

- **Ort** eines Objektes bezogen auf ein BKS:
  - Ortsvektor vom Ursprung des BKS zum Ursprung des OKS
- **Orientierung** eines Objektes bezogen auf ein BKS:
  - Rotationsmatrix zur Abbildung der Einheitsvektoren des OKS auf die Einheitsvektoren des BKS
- **Lage** eines Objektes:
  - Ortsvektor und Rotationsmatrix des OKS bezogen auf das BKS

Objekte und Objektlagen im euklid. 3D-Raum

Definitionen

- Die Lage eines Objektes im 3-dim., euklidischen Raum kann durch ein Tupel von sechs reellen Zahlen beschrieben werden

$$\vec{v} = (x, y, z, \alpha, \beta, \gamma)$$

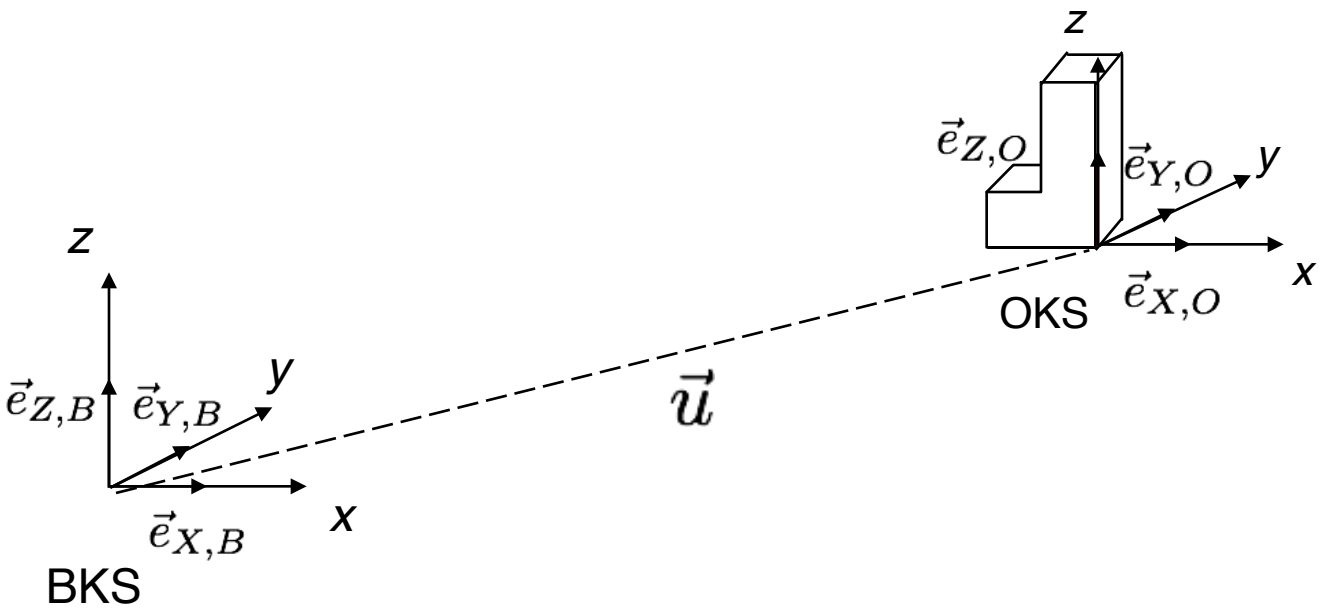
- Die Werte x, y, z sind die Koordinaten des Ursprungs des OKS bezogen auf den Ursprung des BKS (Beschreibung des Ortes)

- Die Werte α, β, γ sind die Drehwinkel, die sich auf die Drehachsen beziehen (Beschreibung der Orientierung)

Objekte und Objektlagen im euklid. 3D-Raum

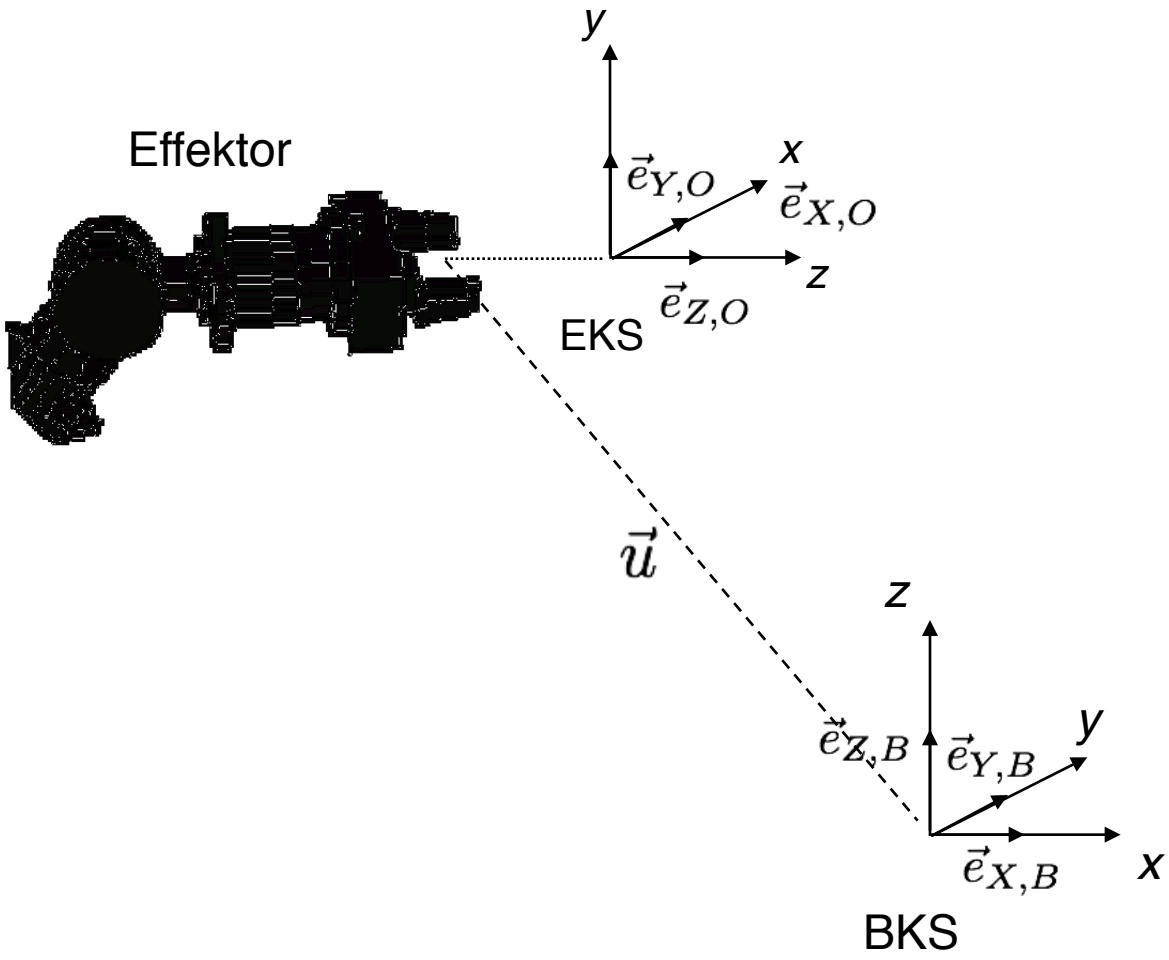
Beispiel (1)

- Translationsvektor: $\vec{u} = a \cdot \vec{e}_{X,B} + b \cdot \vec{e}_{Y,B} + c \cdot \vec{e}_{Z,B}$
- Rotationsmatrix : $R = R_\alpha \bullet R_\beta \bullet R_\gamma$
  - α_x, β_y, γ_z : Rotationswinkel um die Koordinatenachsen x, y, z



Objekte und Objektlagen im euklid. 3D-Raum

Beispiel (2)



Objekte und Objektlagen im euklid. 3D-Raum

Freiheitsgrad f eines Objektes im euklidischen 3D-Raum

- Anzahl möglicher unabhängiger Bewegungen in Bezug auf das BKS; (minimale Anzahl von Translationen und Rotationen zur vollständigen Beschreibung der Lage des Objektes)

- Für im 3-dim. Raum frei bewegliche Objekte gilt: f = 6 (3 Translationen und 3 Rotationen)

Objekte und Objektlagen im euklid. 3D-Raum

Bewegungsfreiheitsgrad F eines Roboters

- Freiheitsgrad eines Rotationsgelenks: $F_R \leq 3$
- Freiheitsgrad eines Translationsgelenks: $F_T = 1$
- Anzahl der Gelenke eines Roboters: n, i.d.R. $n \geq 6$
- Bewegungsfreiheitsgrade: $$F = \sum_{i=1}^{n} (F_{R_i} + F_{T_i})$$

Objekte und Objektlagen im euklid. 3D-Raum

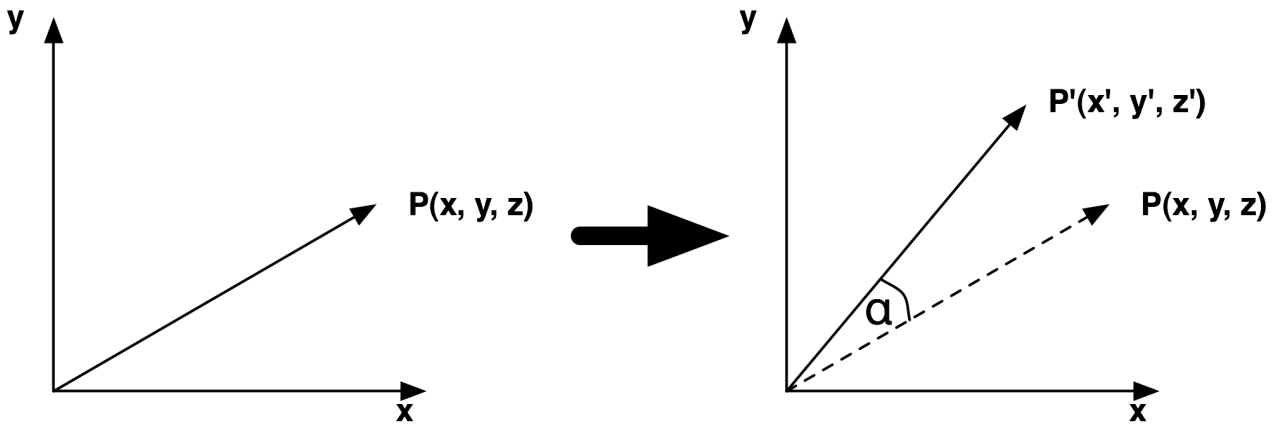
Zusammenhang von f und F

- $F \geq f$
- Beispiele:
  - Freiheitsgrad f eines 7-achsigen Roboters ist 6, Bewegungsfreiheitsgrad F ist 7
  - Freiheitsgrad f einer menschlichen Hand ist 6, Bewegungsfreiheitsgrad F ist 22
- Um einen Freiheitsgrad f = 6 für den Effektor eines Roboters zu erreichen, sind mindestens F = 6 Bewegungsachsen nötig

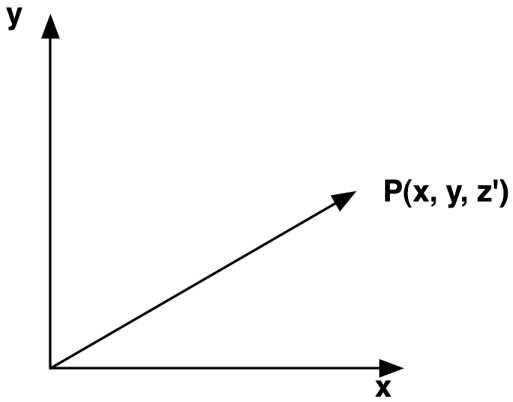
- Englisch: DOF (degrees of freedom)

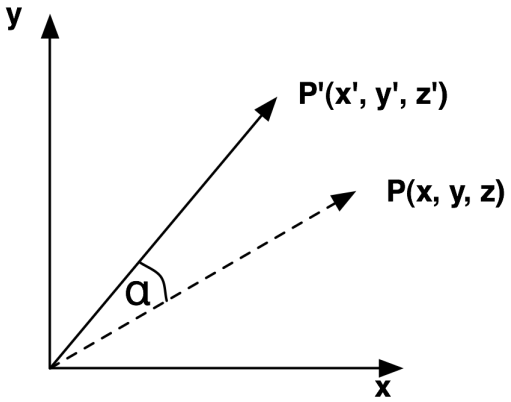
Orientierungsbeschreibung mit 3x3-Matrix

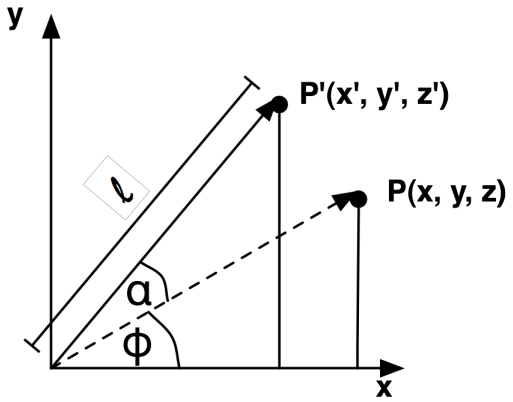
- Jede beliebige Orientierung eines starren Körpers im 3-dim., euklidischen Raum ist erreichbar durch 3 Rotationen um geeignete Achsen. Jede Rotation um eine Achse kann durch eine 3x3 Rotationsmatrix dargestellt werden.
- Beispiel:
  - Rotation eines Punkts P um die z-Achse des zugrundeliegenden Koordinatensystems:



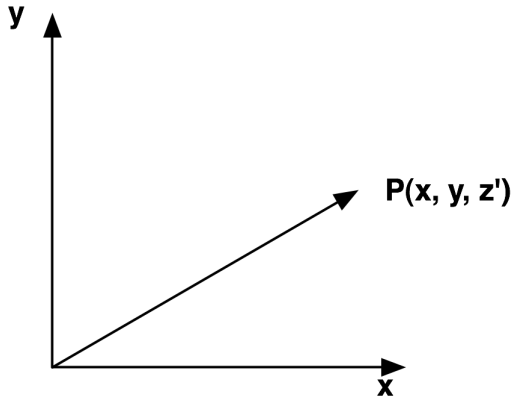
Orientierungsbeschreibung mit 3x3-Matrix

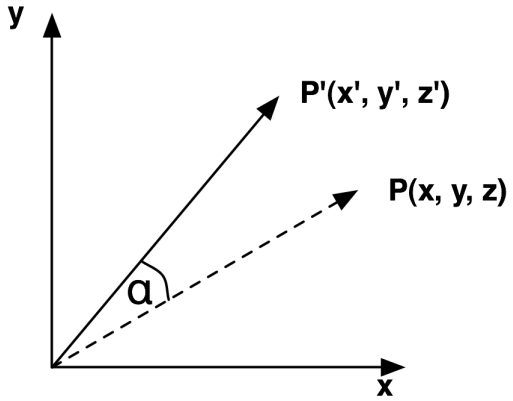


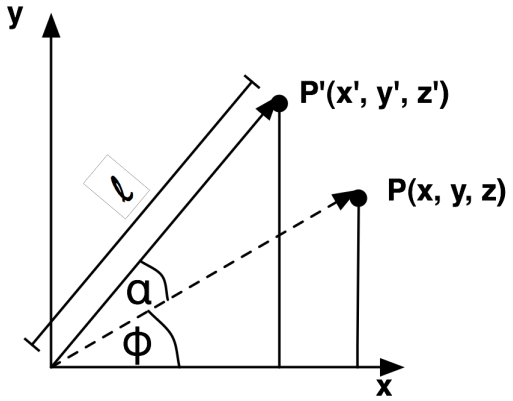




Orientierungsbeschreibung mit 3x3-Matrix







Es gilt (1):

$$x = l \cdot \cos\phi \quad x' = l \cdot \cos(\phi + \alpha)$$
$$y = l \cdot \sin\phi \quad y' = l \cdot \sin(\phi + \alpha)$$

Additionstheorem (2):

$$x' = l \cdot \cos(\phi + \alpha) = l \cdot (\cos\phi \cdot \cos\alpha - \sin\phi \cdot \sin\alpha)$$
$$y' = l \cdot \sin(\phi + \alpha) = l \cdot (\sin\phi \cdot \cos\alpha + \cos\phi \cdot \sin\alpha)$$

Ausmultiplizieren von (2) und Einsetzen von (1) in (2)

$$x' = x \cdot \cos\alpha - y \cdot \sin\alpha$$
$$y' = x \cdot \sin\alpha + y \cdot \cos\alpha$$

Orientierungsbeschreibung mit 3x3-Matrix

- Bei einer Drehung um einen Winkel um die z-Achse gilt:
  - Ein Punkt P mit den Koordinaten $(x,y,z)^T$ geht über in den Punkt P‘ mit den Koordinaten $(x',y',z')^T$ mit:

$$x' = x \cdot \cos\alpha - y \cdot \sin\alpha$$

$$y' = x \cdot \sin\alpha + y \cdot \cos\alpha$$

$$z' = z$$

  - z-Koordinate bleibt fest, da die Drehachse die z-Achse ist

- Schreibweise in Matrixform:

$$\vec{u} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_z(\alpha) \cdot \vec{u}$$

Orientierungsbeschreibung mit 3x3-Matrix

Rotationsmatrizen

- Rotationsmatrix $\underline{\mathbf{R}}_z(\alpha)$ beschreibt diese Drehung

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

- Analog lassen sich die Drehung um die x- bzw. y-Achse beschreiben

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \quad R_y(\alpha) = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix}$$

Orientierungsbeschreibung mit 3x3-Matrix

Rotationsmatrizen

- Beliebige Orientierung
  - Verknüpfung der Rotationen durch Multiplikation der entsprechenden Matrizen

- Anmerkung
  - Matrixmultiplikation ist nicht kommutativ, aber assoziativ

Orientierungsbeschreibung mit 3x3-Matrix

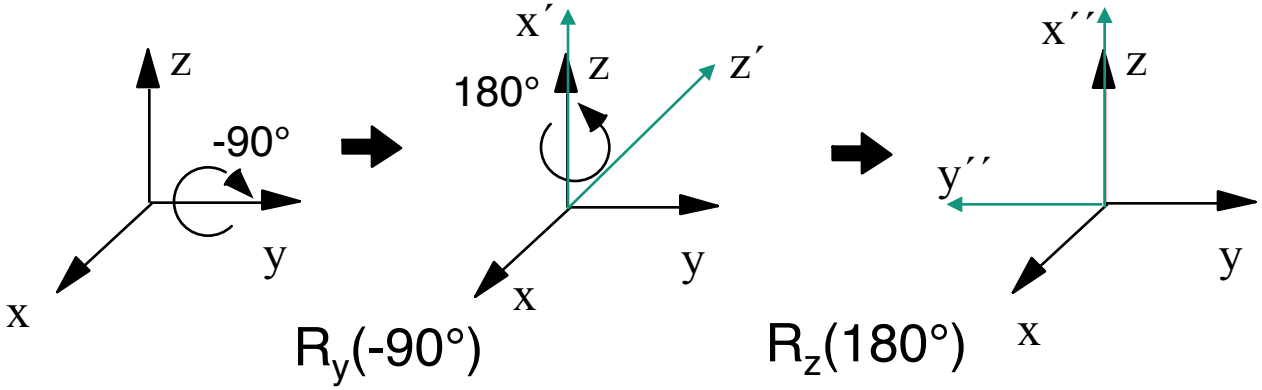
Verkettung von Rotationen

- $R = R_n R_{n-1} \ldots\ldots R_2 R_1$
- Vormultiplikation
  - $R = (R_n (R_{n-1} \ldots\ldots (R_2 R_1) \ldots))$
  - Interpretation: Drehung des momentanen Koordinatensystems um feste Achsen des Ursprungskoordinatensystems
- Nachmultiplikation
  - $R = ((\ldots (R_n R_{n-1}) \ldots\ldots R_2) R_1)$
  - Interpretation: Drehung um momentanes Koordinatensystem

Orientierungsbeschreibung mit 3x3-Matrix

Verkettete, elementare Rotationen

$$\underline{R}_y(-90) = \begin{bmatrix} \cos -\frac{\pi}{2} & 0 & \sin -\frac{\pi}{2} \\ 0 & 1 & 0 \\ -\sin -\frac{\pi}{2} & 0 & \cos -\frac{\pi}{2} \end{bmatrix}$$



$$\underline{R}_z(180) = \begin{bmatrix} \cos\pi & -\sin\pi & 0 \\ \sin\pi & \cos\pi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Sei $\vec{u}$ in OXYZ; Berechnung von $\vec{u}$ nach den beiden Rotationen

$$\vec{u''} = R_z(180)\,\vec{u'} = R_z(180)Ry(-90)\,\vec{u} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} u_3 \\ -u_2 \\ u_1 \end{pmatrix}$$

$$\vec{e_x}'' = \begin{pmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad \vec{e_y}'' = \begin{pmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} \quad \vec{e_z}'' = \begin{pmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{pmatrix}\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

Orientierungsbeschreibung mit 3x3-Matrix

Drehachsen in der Robotik

Es gibt in der Robotik zwei übliche Arten der Festlegung der Rotationsachsen und ihrer Reihenfolge:

a) Euler Winkel
b) Roll, Pitch, Yaw

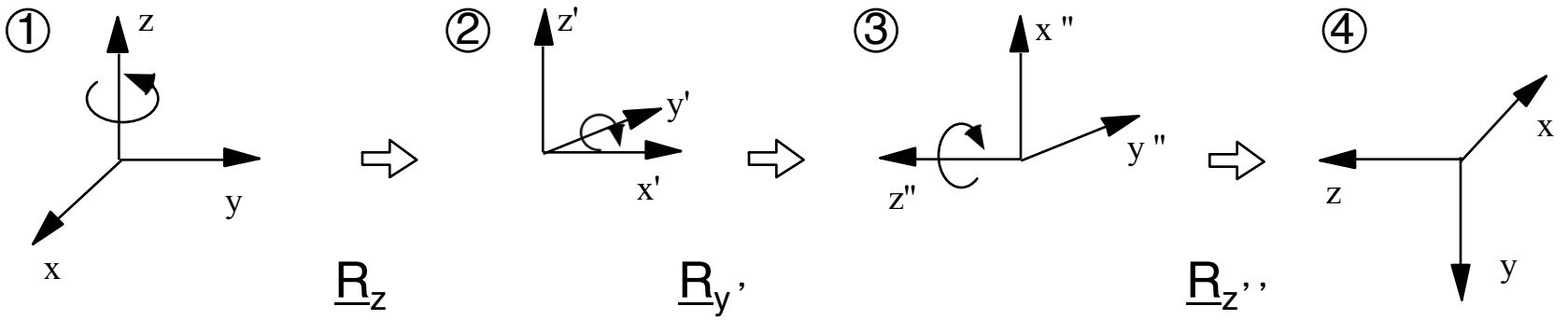
Orientierungsbeschreibung mit 3x3-Matrix

Euler-Winkel

- Drehung α um die z-Achse des BKS $\underline{R}_z$
- Drehung β um die neue y-Achse y′ $\underline{R}_{y'}$
- Drehung γ um die neue z-Achse z″ $\underline{R}_{z''}$

$$\underline{R}_s = \underline{R}_z \bullet \underline{R}_{y'} \bullet \underline{R}_{z''}$$

Wichtig: Drehung um jeweils veränderte Achsen!



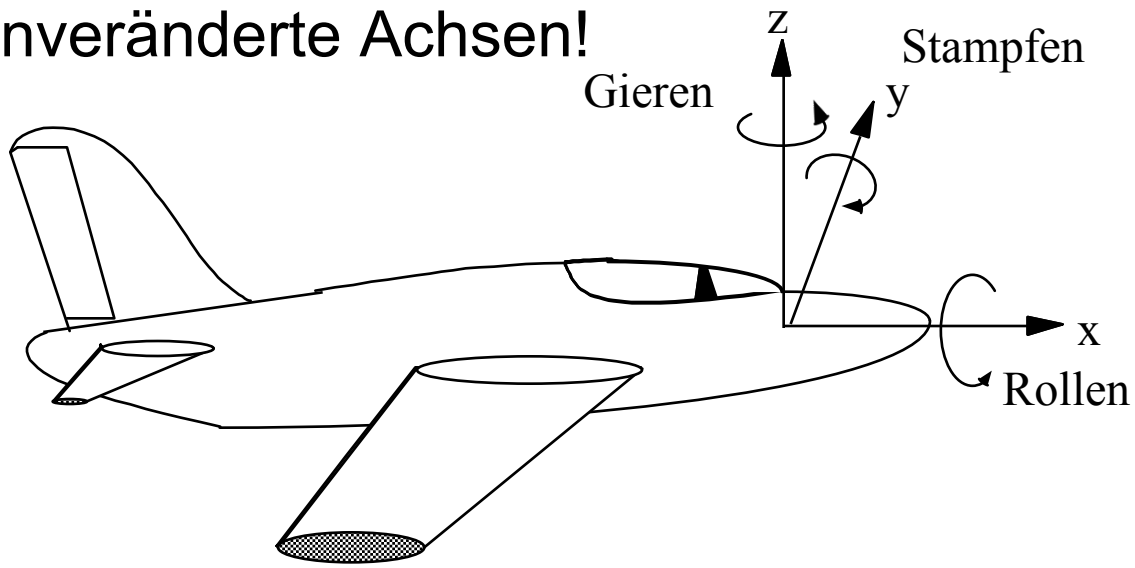
Orientierungsbeschreibung mit 3x3-Matrix

Roll-Pitch-Yaw-Winkel

- Drehung α um die x-Achse des BKS $\underline{R}_x$
- Drehung β um die y-Achse des BKS $\underline{R}_y$
- Drehung γ um die z-Achse des BKS $\underline{R}_z$

$$\underline{R}_s = \underline{R}_z \cdot \underline{R}_y \cdot \underline{R}_x$$

Wichtig: Drehung um unveränderte Achsen!



Orientierungsbeschreibung mit 3x3-Matrix

Eulerwinkel

- Interpretation der Multiplikation $\underline{R}_1 \bullet \underline{R}_2 \bullet \ldots \bullet \underline{R}_n$ von **links nach rechts:** Die Drehung von $\underline{R}_i$ bezieht sich jeweils auf das durch das **linksstehende Matrixprodukt** definierte Koordinatensystem - für $\underline{R}_1$ ist das BKS das Bezugssystem.

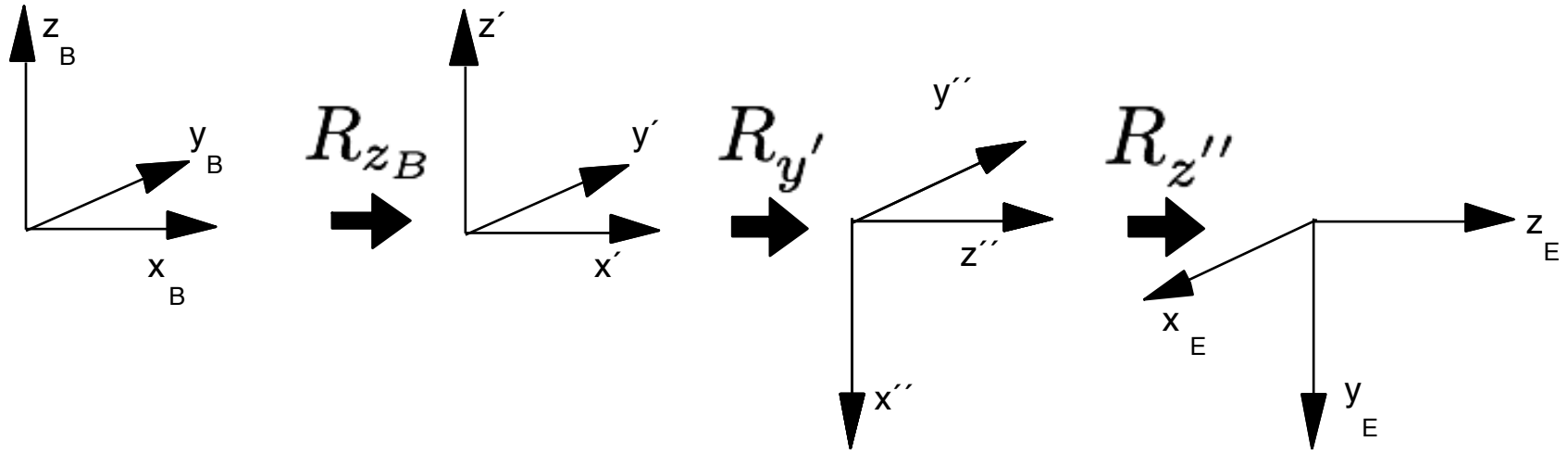
- Anwendung:

$$\underline{R}_s = \underline{R}_z(\alpha) \bullet \underline{R}_{y'}(\beta) \bullet \underline{R}_{z''}(\gamma)$$

  – Interpretation von links nach rechts, da jeweils das neue Koordinatensystem als Bezugssystem definiert wurde

Orientierungsbeschreibung mit 3x3-Matrix

$$RS = R_z(0°) \cdot R_{y'}(90°) \cdot R_{z''}(-90°) =$$



(von links nach rechts aneinanderreihen!)

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

Orientierungsbeschreibung mit 3x3-Matrix

Roll-Pitch-Yaw-Winkel

- Interpretation der Multiplikation $\underline{R}_n \bullet \underline{R}_{n-1} \bullet \ldots \bullet \underline{R}_1$ von **rechts nach links:** Jede Drehung bezieht sich auf das **BKS**

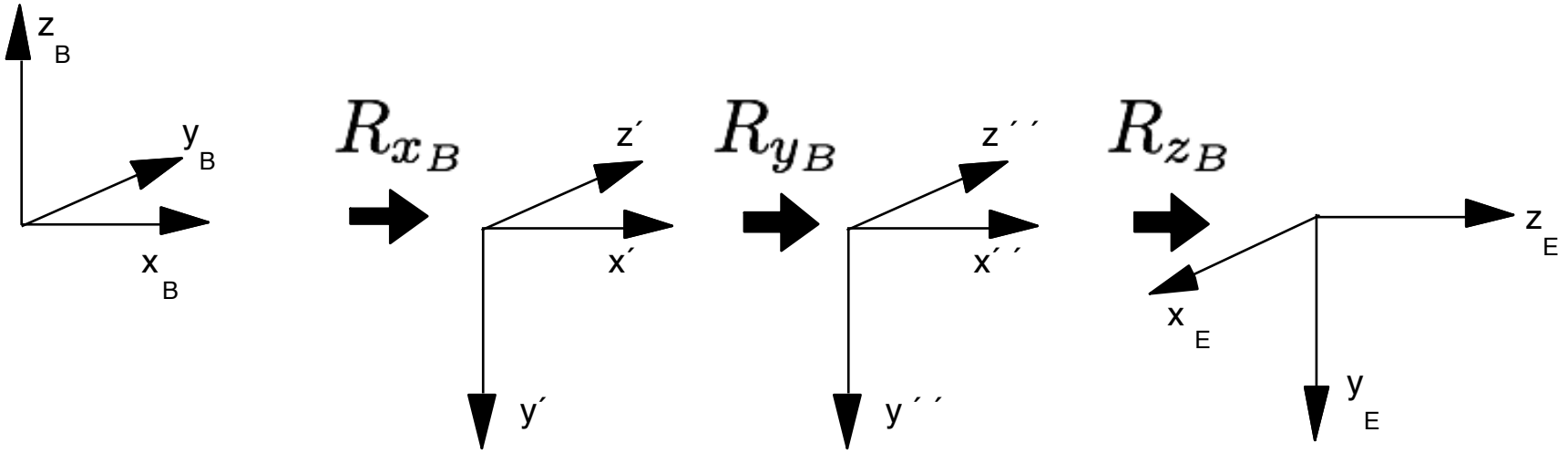
- Anwendung:

$$\underline{R}_s = \underline{R}_z(\gamma) \bullet \underline{R}_y(\beta) \bullet \underline{R}_x(\alpha)$$

  - Multiplikation von rechts nach links, da für alle drei Rotationen das BKS als Bezugssystem definiert wurde

Orientierungsbeschreibung mit 3x3-Matrix

$$RS = R_z(-90°) \cdot R_y(0°) \cdot R_x(-90°) =$$



(von rechts nach links aneinanderreihen!)

$$\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

Orientierungsbeschreibung mit 3x3-Matrix

Vergleich: Euler vs. Roll-Pitch-Yaw

Euler

- Multiplikation von links nach rechts
- Jede Drehung bezieht sich auf das neue Koordinatensystem
- Drehung um jeweils veränderte Achsen

Roll-Pitch-Yaw

- Multiplikation von rechts nach links
- Jede Drehung bezieht sich auf das BKS
- Drehung jeweils um unveränderte Achsen

Orientierungsbeschreibung mit 3x3-Matrix

Bewertung

- Ortsvektor und Rotationsmatrix sind anschaulich und daher die häufigste Form der Eingabe von Objekt- und Endeffektorlagen.
- **Nachteil**: Vektor- und Matrizenoperationen werden getrennt durchgeführt
- **Lösung**: Zusammenfassung von Vektor- und Matrizenoperationen durch Verwendung homogener 4x4 Matrizen

Homogene 4x4-Matrix

- **Ziel:**
  - Geschlossene Darstellung von Rotation und Translation in einer Matrix
  - → Homogene 4x4 Matrix oder Homogene Transformationsmatrix

- Übergang von kartesischen zu homogenen Koordinaten
- Definition der Homogenen Koordinaten kommt aus der Projektiven Geometrie

Homogene 4x4-Matrix

- (x, y, z, w) bezeichnet man als die homogenen Koordinaten des Punktes P(x, y, z).
- Die Verwendung von homogenen Koordinaten erlaubt die Entwicklung von 4x4 Transformationsmatrizen, die die **Rotation**, die **Translation**, die **Skalierung** sowie **perspektivische Transformationen** enthalten.
- Bei homogenen Koordinaten werden Transformationen von n-dim., physikalischen Vektoren in einen n+1-dim. Raum durchgeführt.
- In der Robotik wird w = 1

Homogene 4x4-Matrix

Einführen einer zusätzlichen Komponente

$$P_K = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \Rightarrow P_H = \begin{pmatrix} wx \\ wy \\ wz \\ w \end{pmatrix}$$

Mit $X = \frac{wx}{w}$ $Y = \frac{wy}{w}$ $Z = \frac{wz}{w}$

w ist Skalierungsfaktor, in der Robotik w = 1

Homogene 4x4-Matrix

$$T = \left(\begin{array}{c|c} \mathbf{R}_{3\times 3} & \mathbf{p}_{3\times 1} \\ \hline \mathbf{f}_{1\times 3} & 1\times 1 \end{array} \right)$$

Damit sind alle „Basisoperationen“ darstellbar

Homogene 4x4-Matrix

Homogene „Basisrotationsmatrizen“

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

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

Homogene 4x4-Matrix

Homogene „Basistranslationsmatrix“

$$T_{trans} = \begin{pmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Verschiebung des OKS nach $(dx, dy, dz)^T$ im BKS

Homogene 4x4-Matrix

Perspektivische Transformation

- Hier nicht relevant, deshalb:
- Wird in Computergraphik benutzt

$$\mathbf{f} = (0, 0, 0)$$

Homogene 4x4-Matrix

- Lokale Skalierung:

$$T_{scale} = \begin{pmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} ax \\ by \\ cz \\ 1 \end{pmatrix}$$

- Globale Skalierung:

$$T_{scale} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & s \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ s \end{pmatrix}$$

- Es gilt: $X = \frac{x}{s} \quad Y = \frac{y}{s} \quad Z = \frac{z}{s} \quad w = \frac{s}{s} = 1$

Homogene 4x4-Matrix

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

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

mit:

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

Homogene 4x4-Matrix

Geometrische Interpretation

$$\begin{pmatrix} \mathbf{n} & \mathbf{o} & \mathbf{a} & \mathbf{u} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

u: Verschiebung des OKS

n, o, a: Einheitsvektoren

Homogene 4x4-Matrix

Invertierung

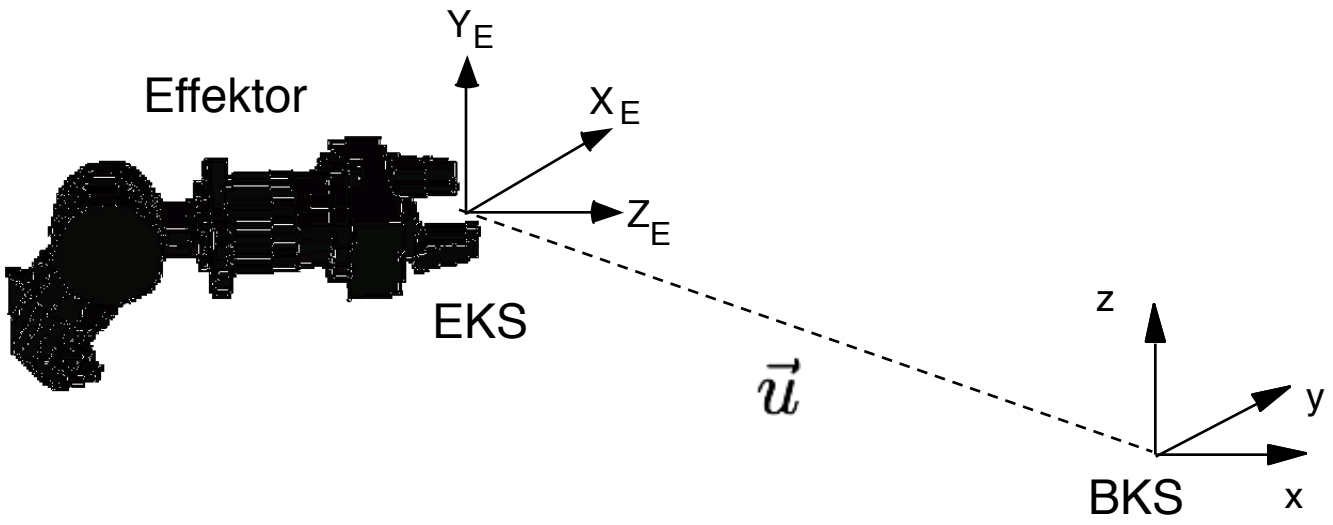
- Es gilt: $R^{-1} = R^T$

$$T^{-1} = \begin{pmatrix} n_x & n_y & n_z & -n^T\mathbf{u} \\ o_x & o_y & o_z & -o^T\mathbf{u} \\ a_x & a_y & a_z & -a^T\mathbf{u} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} & R^T_{3x3} & & -n^T\mathbf{u} \\ & & & -o^T\mathbf{u} \\ & & & -a^T\mathbf{u} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Spaltenvektoren sind die Einheitsvektoren des BKS im OKS

Homogene 4x4-Matrix

Beispiel:



Orientierung

$$(\vec{x_E}, \vec{y_E}, \vec{z_E}) = \begin{bmatrix} 0 & 0 & 1 & u_x \\ 1 & 0 & 0 & u_y \\ 0 & 1 & 0 & u_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ Ort

Homogene 4x4-Matrix

Eigenschaften

- Eine homogene 4x4 Matrix enthält 12 ($\vec{n}, \vec{o}, \vec{a}, \vec{u}$) nichttriviale Kenngrößen im Gegensatz zu 6 (x, y, z, α, β, γ) notwendigen

- Redundanz wegen Orthogonalität

- Drehachsen und Drehreihenfolge sind implizit enthalten

Rotation und Translation von Punkten

- 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

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

Rotation und Translation von Punkten

Interpretationen von homogenen 4x4 Matrizen

- Lagebeschreibung eines Koordinatensystems:

$$^{A}H_{B}$$

  - beschreibt die Lage des Koordinatensystems B relativ zum Koordinatensystem A

- Transformationsabbildung:

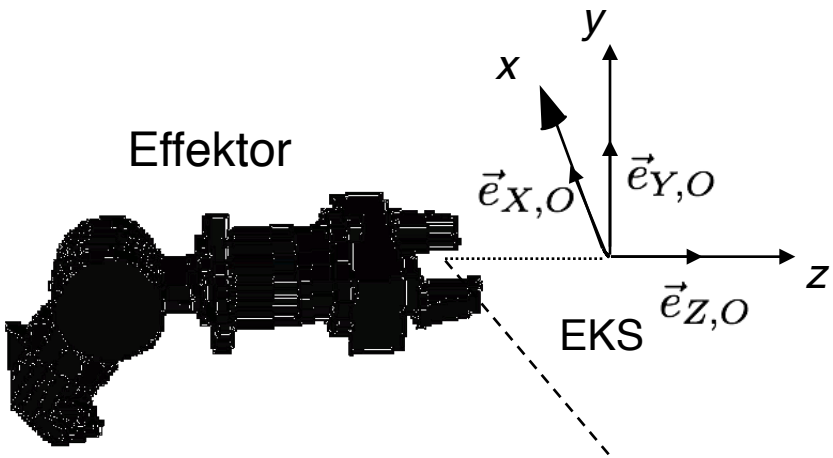
$$^{A}H_{B}: \ ^{B}P \mapsto \ ^{A}P$$

- Transformationsoperator

$$H: \ ^{A}P_{1} \mapsto \ ^{A}P_{2}$$

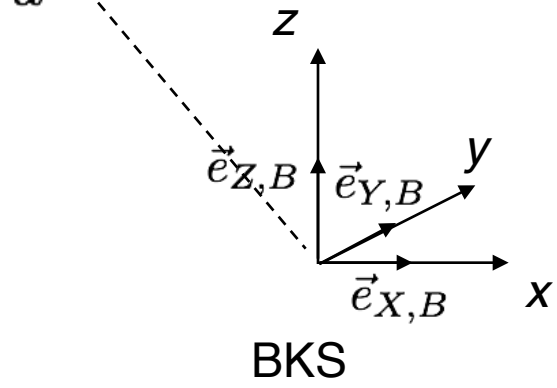
Rotation und Translation von Punkten

Beispiel



$\vec{u}$ = $(-7,0,8)^T$,

$$\underline{R} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$



gegeben sei:

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

gesucht: ^{BKS}P

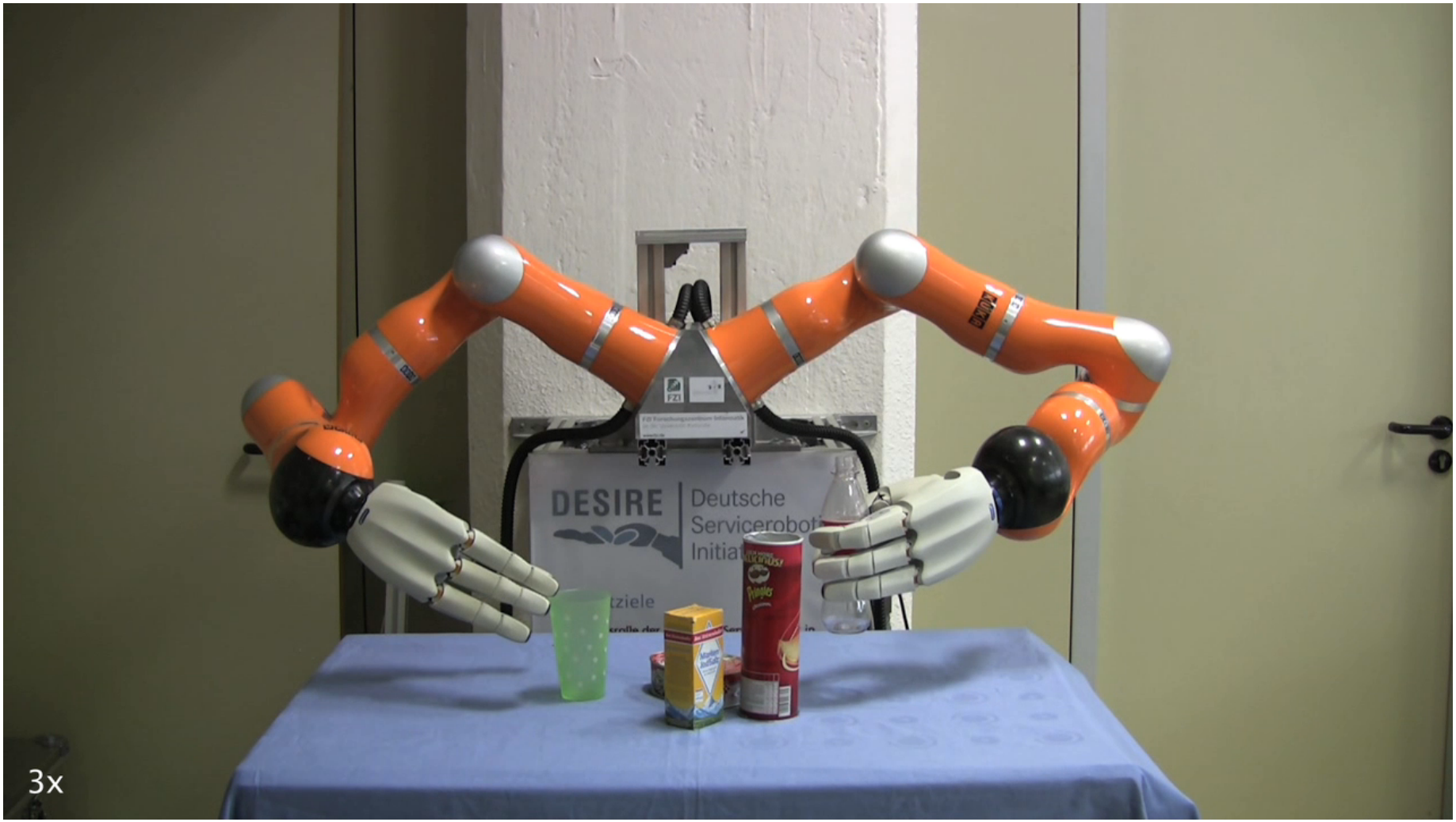
Verkettete Lagebeschreibung

- Eine Lagebeschreibung erfolgt häufig nicht in Bezug auf das BKS, sondern bzgl. eines geeigneter erscheinenden Koordinatensystems (relative Definition)

- Umrechnung von Koordinaten auf verschiedene Bezugssysteme (u.a. BKS) notwendig

- Vorteile der relativen Lagedefinition in der Robotik:
  - Verringerung des Nachführaufwandes bei Objektbewegungen
  - Einzelne Koordinatenangaben beschränken sich auf kürzere Distanzen

Verkettete Lagebeschreibung



Verkettete Lagebeschreibung

- Sei ${}^{BKS}\underline{H}_A$ = (4x4) die Lagebeschreibung des Objekts A bzgl. BKS
- Sei ${}^{A}\underline{H}_B$ = (4x4) die Lagebeschreibung eines Objekts B, bezogen auf das OKS von A
- Sei ${}^{BKS}\underline{H}_B$ = (4x4) die Lagebeschreibung des Objekts B bzgl. BKS
- So gilt

$$ {}^{BKS}\underline{H}_B = {}^{BKS}\underline{H}_A \bullet {}^{A}\underline{H}_B $$

- Im Vergleich zur kartesischen Darstellung kompaktere Schreibweise:

$$ \underline{R}_{B,neu} + \vec{v}_{B,neu} = \underline{R}_A \cdot (\underline{R}_B + \vec{v}_B) + \vec{v}_A = \underline{R}_A \cdot \underline{R}_B + (\underline{R}_A \cdot \vec{v}_B + \vec{v}_A) $$

Verkettete Lagebeschreibung

- Lage von Objekt 1 bzgl. BKS: ${}^{BKS}\underline{H}_1$
- Lage von Objekt 2 bzgl. Objekt 1: ${}^{O_1}\underline{H}_2$
- Lage von Objekt 3 bzgl. Objekt 2: ${}^{O_2}\underline{H}_3$
- Lage von Objekt 3 bzgl. BKS: ${}^{BKS}\underline{H}_3$

$$ {}^{BKS}\underline{H}_3 = {}^{BKS}\underline{H}_1 \bullet {}^{O_1}\underline{H}_2 \bullet {}^{O_2}\underline{H}_3 $$

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

$$ \prod_{i=1}^{n} \underline{H}_{i-1}\,\underline{H}_i \quad \text{mit } \underline{H}_0 = \text{BKS} $$

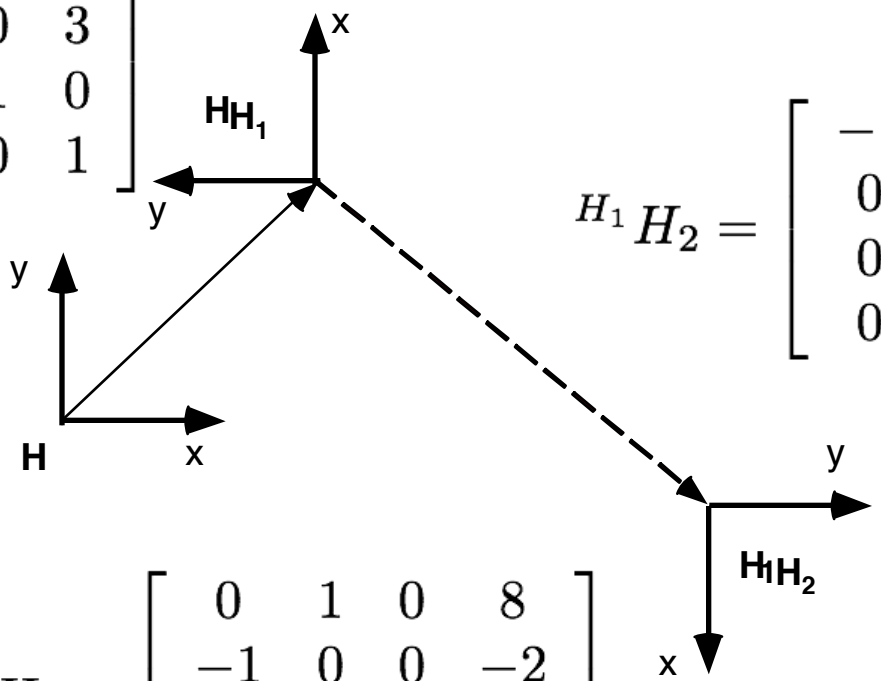
Verkettete Lagebeschreibung

Beispiel

- Objektsystem H_1, entstanden durch eine Transformation $((3,3,0)^T, R_z(90°))$ aus einem beliebigen Bezugsystem B: ${}^{B}\underline{H_1}$
- Objektsystem H_2, entstanden durch eine Transformation $((-5,-5,0)^T, R_z(-180°))$ aus dem System des Objekts H_1: ${}^{H1}\underline{H_2}$
- Gegeben: Punkt $P=(1, 2, 3)^T$ in H_2

Verkettete Lagebeschreibung

$$ {}^{H}H_1 = \begin{bmatrix} 0 & -1 & 0 & 3 \\ 1 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$



$$ {}^{H_1}H_2 = \begin{bmatrix} -1 & 0 & 0 & -5 \\ 0 & -1 & 0 & -5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{H}H_2 = {}^{H}H_1 \cdot {}^{H_1}H_2 = \begin{bmatrix} 0 & 1 & 0 & 8 \\ -1 & 0 & 0 & -2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Verkettete Lagebeschreibung

Bewertung

- Rotationsmatrizen + Translation:
  - 12 Parameter (9 Rotation, 3 Translation)
  - Hohe Redundanz
  - Interpolation schwierig
- Geht‘s besser?
  - Ja, mit Quaternionen
  - Kompakte Darstellung
  - Erstmals beschrieben von Hamilton 1843
  - Einsatz erst seit den 90er Jahren (Computergraphik)

Quaternionen

Definitionen

- Auffassung als hyperkomplexe Zahlen
- Mit $a,b,c,d \in \mathbb{R}$ Quaternion q wie folgt beschrieben:

$$q = a + b \cdot i + c \cdot j + d \cdot k$$

- Mit

$$\begin{array}{lllll} i^2 & = & j^2 & = & k^2 = ijk = -1 \\ ij & = & -ji & = & k \\ jk & = & -kj & = & i \\ ki & = & -ik & = & j \end{array}$$

- a ist der **Realteil**, **u** = (b, c, d)^T der **Imaginärteil**
  q = (a, b, c, d)^T oder auch q = (a, **u**)^T

Quaternionen

Rechenregeln (1)

- Gegeben $q = (a_q, \mathbf{u}_q), r = (a_r, \mathbf{u}_r)$

- Addition

$$q + r = (a_q + a_r, \mathbf{u}_q + \mathbf{u}_r)$$

- Punktprodukt (Skalarprodukt)

$$q \cdot r = a_q a_r + < \mathbf{u}_q, \mathbf{u}_r > = a_q a_r + b_q b_r + c_q c_r + d_q d_r$$

- Quaternionen-Multiplikation

$$q * r = (a_q + i \cdot b_q + j \cdot c_q + k \cdot d_q) \cdot (a_r + i \cdot b_r + j \cdot c_r + k \cdot d_r)$$

Quaternionen

Rechenregeln (2)

- Konjugierte Quaternion

$$q^* = (a_q, -\mathbf{u}_r)$$

- Norm

$$|q| = \sqrt{qq^*} = \sqrt{q^*q} = \sqrt{a^2 + b^2 + c^2 + d^2}$$

- Multiplikatives Inverses Element

$$q^{-1} = q^* / |q|^2$$

Quaternionen

Rotation mit Quaternionen

- Vektor **p** = $(x, y, z)^T$ als Quaternion

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

- Skalar s als Quaternion

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

- Einheitsquaternion

$$|\mathbf{q}| = 1$$

$$q * r = (a_q + i \cdot b_q + j \cdot c_q + k \cdot d_q) \cdot (a_r + i \cdot b_r + j \cdot c_r + k \cdot d_r)$$

Quaternionen

Rotation mit Quaternionen

- geg. 3-dim. Einheitsvektor **u**, Winkel θ
  dann repräsentiert das Quaternion

$$q = (\cos\theta/2, \mathbf{u}\sin\theta/2)$$

  eine Rotation um θ mit Rotationsachse **u**
- Ein Punkt **v** wird rotiert durch:

$$\mathbf{v}' = qvq^* = qvq^{-1}$$

Quaternionen

Beispiel

Punkt $P = (1, 0, 9)^T$

Drehachse $\mathbf{u} = e_x = (1, 0, 0)^T$

Drehwinkel $\theta = 90°$

$q_p = (0, 1, 0, 9)^T = i + 9k$

$q_r = (\cos\ \theta/2, \sin\ \theta/2, 0, 0)^T = \cos\ \theta/2 + i \sin\ \theta/2$

$q_p' = q_r q_p q_r^* = (0, 1, -9, 0)^T$

➡ $p' = (1, -9, 0)^T$

Orientierungsbeschreibung mit 3x3-Matrix

Rotationsmatrizen

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

Formeln

$$\begin{aligned} i^2 &= j^2 = k^2 = ijk = -1 \\ ij &= -ji = k \\ jk &= -kj = i \\ ki &= -ik = j \end{aligned}$$

Rechnung

$q_p = (0, 1, 0, 9)^T = i + 9k, \quad \theta = 90°$

$q_r = (\cos \theta/2, \sin \theta/2, 0, 0)^T = \cos \theta/2 + i \sin \theta/2$

$q_p' = q_r q_p q_r^*$

Quaternionen

Rotation mit Quaternionen

- Hintereinanderschalten von Rotationen:
  geg.:

$$q = (\cos\theta_q/2, \mathbf{u}_q \sin\theta_q/2)$$
$$r = (\cos\theta_r/2, \mathbf{u}_r \sin\theta_r/2)$$

- sowie: $f(v) = qvq^{-1}$ und $h(v) = rvr^{-1}$

- dann entspricht $f \circ h$ gerade der Rotation mit dem Quaternion p = q*r

Quaternionen

Bewertung

- Intuitive Darstellung von Rotationen („direkte“ Angabe von Drehwinkel und -achse)
- Kompakte Darstellung (nur 4 Werte im Vergleich zu 9 Werten bei Rot.Matrix)
- Rotation direkt um gewünschte Achse
- Numerische Stabilität

Aber:

- Nur Rotation, keine Translation

Duale Quaternionen

Warum ?

- Reelle Quaternionen eignen sich für die Beschreibung der Orientierung, nicht aber der Lage eines Objektes

- Um neben der Orientierung auch die Lage in einem Quaternion ausdrücken zu können, werden die 4 reellen Werte durch Dualzahlen ersetzt

Duale Quaternionen

Duale Zahlen

- Duale Zahlen sind von der Form

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

- **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 für
  - 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

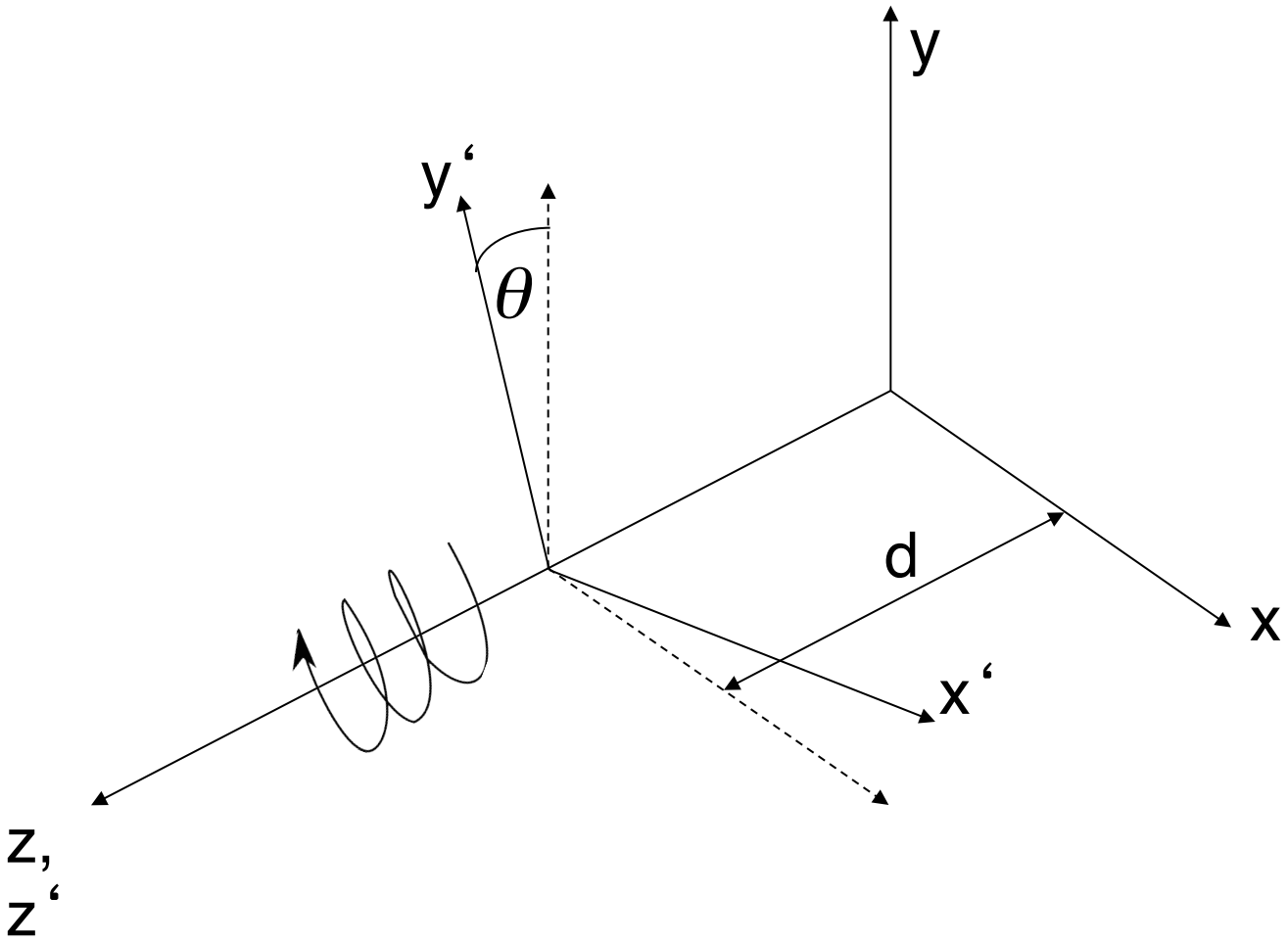
Beschreibung

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

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

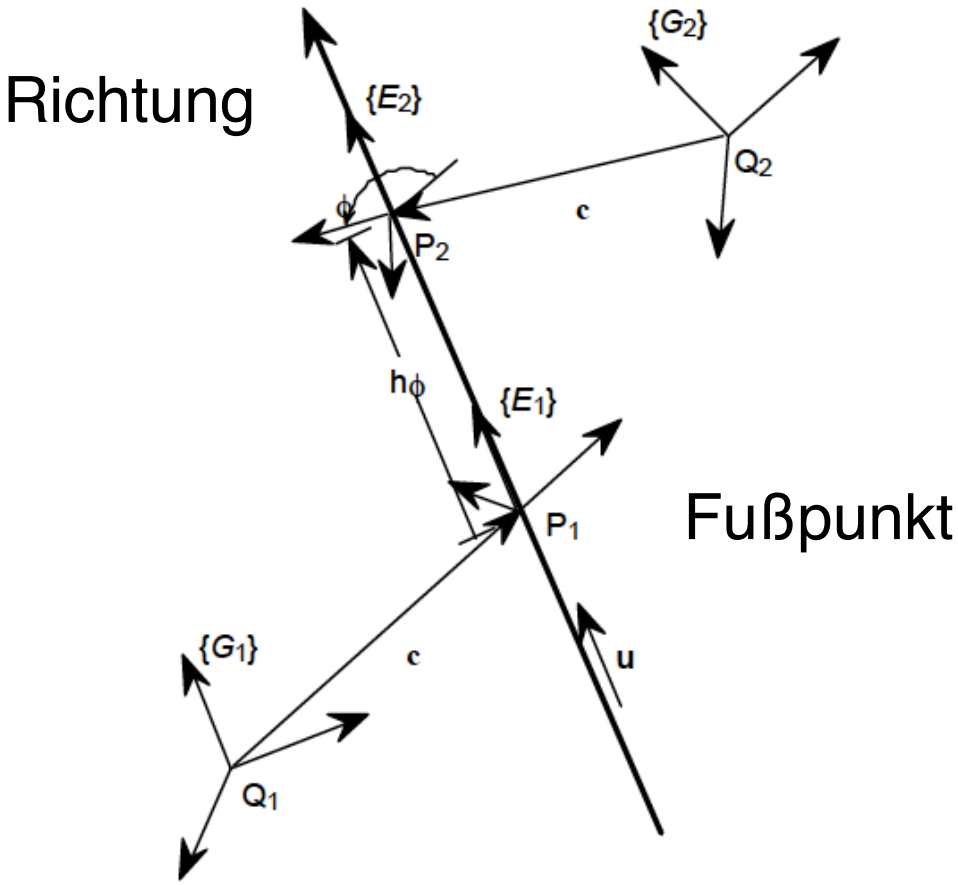
Duale Quaternionen

Rotation und Translation durch Duale Quaternionen



Duale Quaternionen

Rotation und Translation durch Duale Quaternionen



Duale Quaternionen

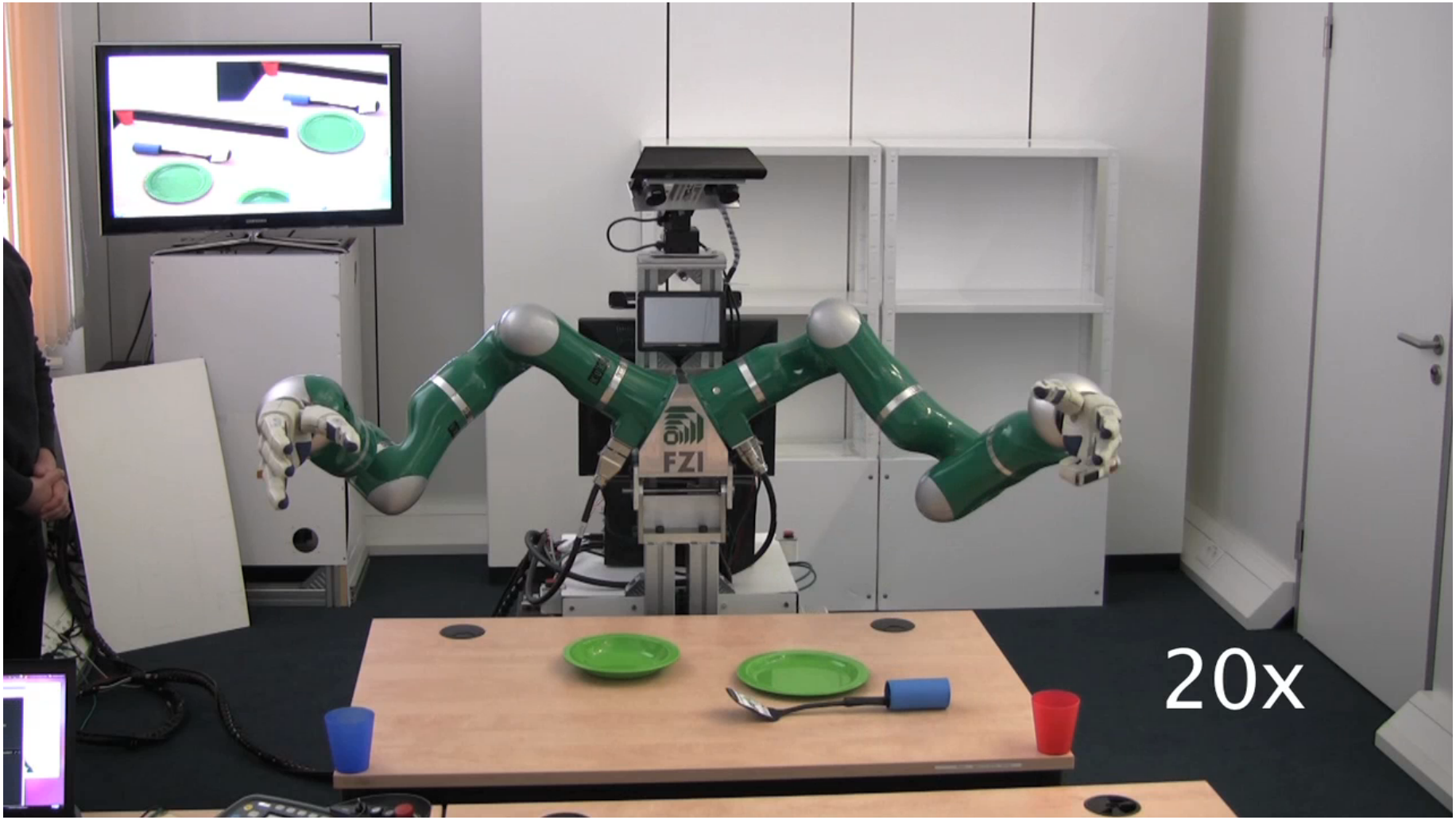
Bewertung

- Dualquaternionen sind zur Stellungsbeschreibung eines Objekts geeignet
- Operationen auf Dualquaternionen erlauben auch alle benötigten Transformationen
- Geringe Redundanz, da nur 8 Kenndaten
- Schwierigkeit für den Anwender, eine Stellung durch Angabe einer Dualquaternion zu beschreiben
- Komplexe Verarbeitungsvorschriften (z.B. für Multiplikation)
- In der Regel geringe Anzahl an Einzeloperationen je Rechenoperation

Zusammenfassung

- Rotationsmatrix und Verschiebungsvektor
- Homogene 4x4-Matrix
- Verkettete Lagebeschreibung
- Quaternionen
- Duale Quaternionen

- Verschiedene Darstellungsarten um Rotationen und Translationen im euklidischen Raum darzustellen
- Jede Darstellungsart hat spezifische Vor- und Nachteile
- Konkrete Anwendung bestimmt Auswahl der Methode


FZI
20x