Angewandte Industrielle Robotik

Zusammenfassung

Joel von Rotz / Quelldateien

Inhaltsverzeichnis

Repetition Linear Algebra 1

- Vektoren 2
  - Skalarprodukt 2
  - Winkel & Orthogonalität 2
  - Kreuzprodukt 2
- Matrizen 2
  - Schiefsymmetrische Matrix: $A = -A^T$ 2
  - Asymmetrische Matrix: $A = A^T$ 2
  - Selektion Untermatrizen oder Vektoren 2
  - Inverse Matrix/Kehrmatrix 2

Rotationen & Translationen 3

- Rotationsmatrix 3
  - Aufbau 3
  - Eigenschaften 3
  - Inverse 3
  - Verkettung 3
- Homogene Matrizen (Frames) 4
  - Inverse 4
  - Verkettung 4
- Orienterung durch Euler-Winkel 4
- Roll Pitch Yaw (Roll Neigung Gier) 5
- Orientierung durch Drehvektor und -winkel 5

Quaternion 5

Umwandlungen $Q \leftrightarrow EA \leftrightarrow {}^k_iA$ 5

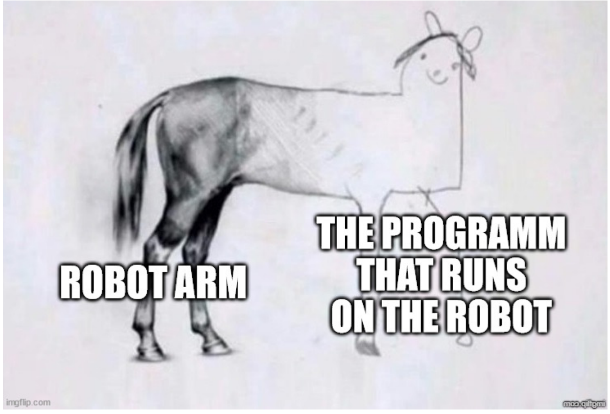
- Euler zu Quartenion 5
- Quartenion zu Euler 5
- Euler zu Rotationsmatrix 5
- Rotationsmatrix zu Euler 5
- Quaternion zu Rotationsmatrix 6
- Rotationsmatrix zu Quaternion 6

Spaghett 6

Bewegungsort & Interpolation 6

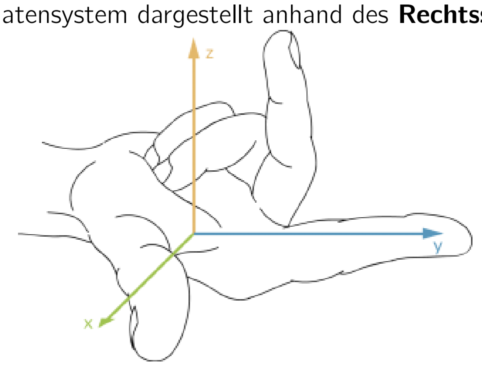
- Bahnparameter $s(t)$ 6

Software mit Schneebeli 6



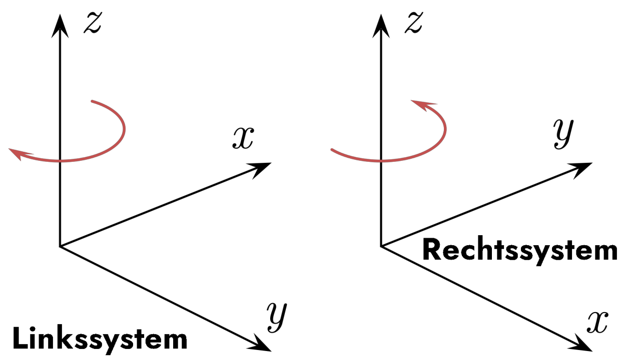
Rechtssystem

Robotereffektoren, Armteile, etc. werden im kartesischen Koordinatensystem dargestellt anhand des **Rechtssystems**.



Drehung mit der Rechtenhandregel

Bei der Rotation von Rechtssystemen kann die rechte Handregel angewendet werden. Der Daumen zeigt in die gleiche Richtung der Drehachse (vom Nullpunkt nach aussen) und die rechstlichen Finger zeigen die Drehrichtung an: **Gegenunzeigersinn**!



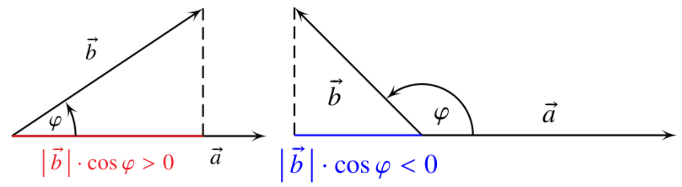
Repetition Linear Algebra

Für das meiste die Linear Algebra Zusammenfassung anschauen.

Aber sonst das Wichtigste

Vektoren

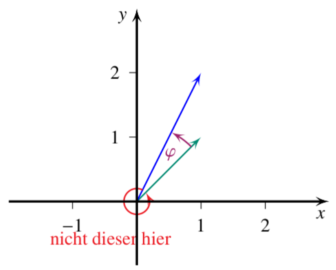
Skalarprodukt



Das Skalarprodukt entspricht der Multiplikation der Projektion $\overrightarrow{b_a}$ auf $\overrightarrow{a}$ mit $\overrightarrow{a}$

$$\overrightarrow{a} \bullet \overrightarrow{b} = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} \bullet \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} = a_1 \cdot b_1 + \dots + a_n \cdot b_n = \sum_{i=1}^{n} a_i \cdot b_i$$

Winkel & Orthogonalität



Beim Berechnen des Winkels zwischen zwei Vektoren

$$\varphi = \arccos \frac{x \bullet y}{\|x\| \cdot \|y\|}$$

Es gilt:

- $\overrightarrow{a} \bullet \overrightarrow{b} > 0 \quad \text{wenn} \quad \varphi < \frac{\pi}{2}$
- $\overrightarrow{a} \bullet \overrightarrow{b} < 0 \quad \text{wenn} \quad \varphi > \frac{\pi}{2}$

! Definition Orthogonalität

Sind zwei Vektoren *orthogonal / senkrecht* zueinander, ergibt das Skalarprodukt

$$\overrightarrow{a} \bullet \overrightarrow{b} = 0 \quad \text{und} \quad \varphi = \frac{\pi}{2}$$

i Richtungswinkel in $\mathbb{R}^3$

$$\cos\alpha = \frac{a_x}{a} \;\&\; \cos\beta = \frac{a_y}{a} \;\&\; \cos\gamma = \frac{a_z}{a}$$

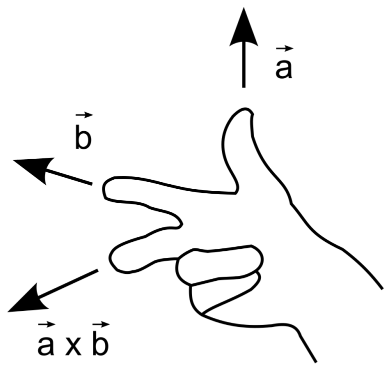
$$\cos^2\alpha + \cos^2\beta + \cos^2\gamma = 1$$

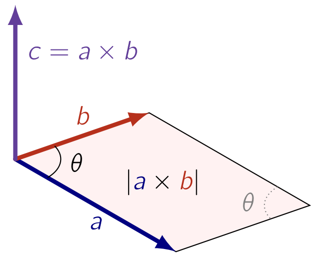
Kreuzprodukt

Mit dem Kreuzprodukt kann ein **orthogonaler** Vektor Teil eines Rechtssystems bestummen werden.

$$\vec{\mathbf{c}} = \vec{\mathbf{a}} \times \vec{\mathbf{b}} = \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} = \begin{bmatrix} \mathbf{a}_y \cdot \mathbf{b}_z - \mathbf{a}_z \cdot \mathbf{b}_y \\ \mathbf{a}_z \cdot \mathbf{b}_x - \mathbf{a}_x \cdot \mathbf{b}_z \\ \mathbf{a}_x \cdot \mathbf{b}_y - \mathbf{a}_y \cdot \mathbf{b}_x \end{bmatrix}$$

Der Mittelfinger der beiden Vektoren ist das Kreuzprodukt. Je nach Betrachtung des Systems zeigt der Normalvektor in die andere Richtung





Matrizen

Schiefsymmetrische Matrix: $A = -A^T$

$$A^T = -A \quad \Rightarrow \quad \begin{bmatrix} \mathbf{0} & 7 & 23 \\ -7 & \mathbf{0} & -4 \\ -23 & 4 & \mathbf{0} \end{bmatrix}^T = \begin{bmatrix} \mathbf{0} & -7 & -23 \\ 7 & \mathbf{0} & 4 \\ 23 & -4 & \mathbf{0} \end{bmatrix}$$

Asymmetrische Matrix: $A = A^T$

$$A^T = -A \quad \Rightarrow \quad \begin{bmatrix} \mathbf{0} & 1 & 2 \\ 1 & \mathbf{0} & 3 \\ 2 & 3 & \mathbf{0} \end{bmatrix}^T = \begin{bmatrix} \mathbf{0} & 1 & 2 \\ 1 & \mathbf{0} & 3 \\ 2 & 3 & \mathbf{0} \end{bmatrix}$$

Selektion Untermatrizen oder Vektoren

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{bmatrix}$$

$$A(1:2, 1:2) \quad A(:, 3)$$

Inverse Matrix/Kehrmatrix

Für 2×2 Matrizen:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$A^{-1} = \frac{1}{\det A} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Anhand der Adjunkte $\mathrm{adj}(B)$ um 3×3 Matrizen zu invertieren (auch grössere möglich).

$$B = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

$$\mathrm{inv}(B) = B^{-1} = \frac{1}{\det(B)} \cdot \mathrm{adj}(B) \qquad \mathrm{adj}(B) = [\mathrm{cof}(B)]^T$$

Bevor diese Berechnung gemacht werden kann, muss die Matrix auf die Invertierbarkeit geprüft werden → Determinante $\det(B) \neq 0$

$$B = \begin{bmatrix} a^+ & b^- & c^+ \\ d^- & e^+ & f^- \\ g^+ & h^- & i^+ \end{bmatrix} \quad \mathrm{adj}(B) = \begin{bmatrix} +\det(\left[\begin{smallmatrix} e & f \\ h & i \end{smallmatrix}\right]) & \cdots & \cdots \\ -\det(\left[\begin{smallmatrix} b & c \\ h & i \end{smallmatrix}\right]) & +\det(\left[\begin{smallmatrix} a & c \\ g & i \end{smallmatrix}\right]) & \\ \vdots & & \ddots \end{bmatrix}^T$$

$$\begin{bmatrix} a & \mathbf{b} & \mathbf{c} \\ \mathbf{d} & e & f \\ g & \mathbf{h} & \mathbf{i} \end{bmatrix} \quad \begin{bmatrix} \mathbf{a} & b & c \\ d & \mathbf{e} & \mathbf{f} \\ g & \mathbf{h} & \mathbf{i} \end{bmatrix} \quad \begin{bmatrix} \mathbf{a} & b & \mathbf{c} \\ d & \mathbf{e} & f \\ \mathbf{g} & h & \mathbf{i} \end{bmatrix}$$

$$E = \begin{bmatrix} -2 & 4 & 1 \\ 4 & -1 & 0 \\ 1 & 0 & 4 \end{bmatrix}$$

$$\mathrm{inv}(E) = E^{-1} = \frac{\mathrm{adj}(E)}{\det(E)}$$

Rotationen & Translationen

Rotationsmatrix

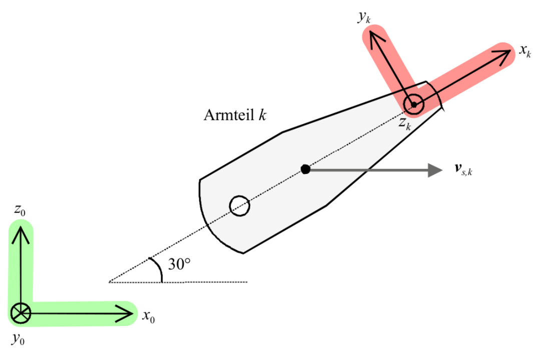
Rotationsmatrix beschreibt eine **Rotation** und wird in Form ${}^a_bR$ dargestellt. Das ${}^a_b$ beschreibt Rotationsmatrix von **a** nach **b**.

Aufbau

Die Matrix wird folgend beschrieben.

$${}^a_bR = \begin{bmatrix} x_a^{(b)} & y_a^{(b)} & z_a^{(b)} \end{bmatrix}$$

Beispiel



$$v_{s,k}^{(k)} = {}^0_kT \cdot v_{s,k}^{(0)}$$

Vektor $v_{s,k}^{(0)}$ wird im 0 Koordinatensystem dargestellt, nun wird es mit ${}^0_kT$ multipliziert. Das Endprodukt ist immer noch der gleiche Vektor, einfach nun im Bezug zum Koordinatensystem k

Eigenschaften

- Zeilen- & Spaltenvektoren sind **orthogonal** zueinander
- Determinante $\det({}^k_0A) = 1$
- Betrag von Spalten & Zeilen = 1

Inverse

$$({}^k_0A)^{-1} = {}^0_kA = {}^k_0A^T$$

Verkettung

$${}^k_iA = {}^{i+1}_{\;\;\;i}A \cdot {}^{i+2}_{i+1}A \cdot \cdots \cdot {}^{k-1}_{k-2}A \cdot {}^{\;\;\;k}_{k-1}A$$

und einer inverse Verkettung:

$$\begin{aligned} {}^i_kA = \left[{}^k_iA\right]^T &= {}^{\;\;\;k}_{k-1}A^T \cdot {}^{k-1}_{k-2}A^T \cdot \cdots \cdot {}^{i+2}_{i+1}A^T \cdot {}^{i+1}_{\;\;\;i}A^T \\ &= {}^{k-1}_{\;\;\;k}A \cdot {}^{k-2}_{k-1}A \cdot \cdots \cdot {}^{i+1}_{i+2}A \cdot {}^{\;\;\;i}_{i+1}A \end{aligned}$$

Beispiel

$$ {}^{3}_{1}A = {}^{2}_{1}A \cdot {}^{3}_{2}A $$

$$ {}^{1}_{3}A = {}^{3}_{2}A^T \cdot {}^{2}_{1}A^T = {}^{2}_{3}A \cdot {}^{1}_{2}A $$

Homogene Matrizen (Frames)

Damit die Lage des Effektors im Raum eindeutig bestimmbar ist, wird die Rotationsmatrix mit einem Verschiebungsvektor erweitert. Dadurch ist **Orientierung** und **Position** bestimmbar.

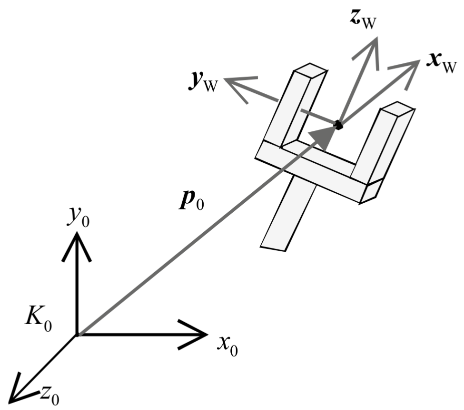
$$ a^{(k)} = {}^{0}_{k}T \cdot a^{(0)} $$

*Vom Bergspitz **0** aus den See **a** anschauen, dann via dem Wanderweg ${}^{0}_{k}T$ zum Bergspitz **k** und von dort auf denselben See **a** schauen.*

Die homogene Matrix/Frame besteht aus einer **Rotation**, einer **Translation** und einem sehr markanten **1**.

$$ {}^{k}_{i}T = \begin{bmatrix} x_k^{(i)} & y_k^{(i)} & z_k^{(i)} & p_{ik}^{(i)} \\ 0 & 0 & 0 & \mathbf{1} \end{bmatrix} $$

Die Verschiebung wird genutzt um ein Frame auf eine Position zu setzen, z.B. am TCP des Werkzeugs.



Beispiel

Vektor a wird vom Koordinatensystem K_k ins System K_i überführt. Bei **Freien** Vektor wird der Wert in der vierten Zeile auf **0** gesetzt, da die Position des Vektors nicht wichtig ist → Rotation würde genügen

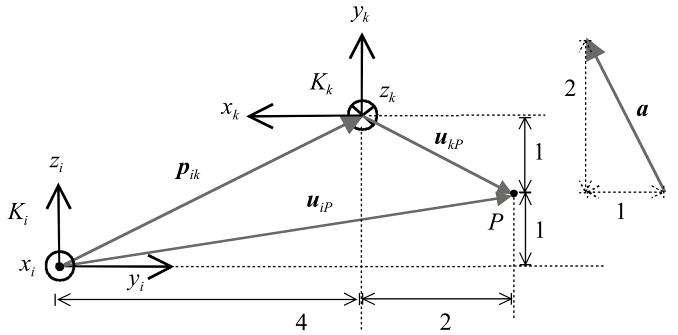
$$ {}^{k}_{i}T = \begin{bmatrix} 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad a^{(k)} = \begin{bmatrix} 1 \\ 2 \\ 0 \\ \mathbf{0} \end{bmatrix} $$

$$ a^{(i)} = {}^{k}_{i}T \cdot a^{(k)} = \begin{bmatrix} 0 & 0 & -1 & \mathbf{0} \\ -1 & 0 & 0 & \mathbf{4} \\ 0 & 1 & 0 & \mathbf{2} \\ 0 & 0 & 0 & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 0 \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 2 \\ 0 \end{bmatrix} $$

Der Nutzen des Frames ist bei Ortsvektoren ersichtlich. Bei diesen Vektoren wird der vierte Wert auf **1** gesetzt. Vektor $u_{kP}^{(k)}$ zeigt auf Punkt P und wird nun im Bezug zu K_i in $u_{iP}^{(i)}$ transformiert.

$$ u_{kP}^{(k)} = \begin{bmatrix} -2 \\ -1 \\ 0 \\ 1 \end{bmatrix} $$

$$ u_{iP}^{(i)} = {}^{k}_{i}T \cdot u_{kP}^{(k)} = \begin{bmatrix} 0 & 0 & -1 & \mathbf{0} \\ -1 & 0 & 0 & \mathbf{4} \\ 0 & 1 & 0 & \mathbf{2} \\ 0 & 0 & 0 & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} -2 \\ -1 \\ 0 \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} 0 \\ 6 \\ 1 \\ 1 \end{bmatrix} $$



Während ein freier Vektor sich nicht verändert, ändern sich Ortsvektoren zu komplett neuen Vektoren ($|u_{kP}^{(k)}| \neq |u_{iP}^{(i)}|$).

Inverse

Zu **jeder** homogenen Matrix ist **immer** die inverse homogene Matrix gegeben und es gilt speziell:

$$ {}^{i}_{k}T = ({}^{k}_{i}T)^{-1} = \begin{bmatrix} {}^{k}_{i}A^T & -\,{}^{k}_{i}A^T \cdot p_{ik}^{(i)} \\ [0 \quad 0 \quad 0] & 1 \end{bmatrix} $$

Verkettung

$$ {}^{k}_{i}T = {}^{i+1}_{i}T \cdot {}^{i+2}_{i+1}T \cdot \dots \cdot {}^{k-1}_{k-2}T \cdot {}^{k}_{k-1}T $$

und einer inverse Verkettung:

$$ \begin{aligned} {}^{i}_{k}T = \left[{}^{k}_{i}T\right]^{-1} &= {}^{k}_{k-1}T^{-1} \cdot {}^{k-1}_{k-2}T^{-1} \cdot \dots \cdot {}^{i+2}_{i+1}T^{-1} \cdot {}^{i+1}_{i}T^{-1} \\ &= {}^{k-1}_{k}T \cdot {}^{k-2}_{k-1}T \cdot \dots \cdot {}^{i+1}_{i+2}T \cdot {}^{i}_{i+1}T \end{aligned} $$

$]^{-1}$ anstatt $]^T$

Da die homogene Matrix nicht einfach so transponiert werden darf, muss die spezielle Umformung verwendet werden!

Orienterung durch Euler-Winkel

Eine Drehung eines Frames im Bezug zu einem anderen ist durch **drei** Winkel angegeben. Diese drei Winkel werden **Euler-Winkel** genannt, wenn diese nacheinander ausgeführt werden.

$$R_{ZYX} = {}^W_R A = \begin{pmatrix} x_W^{(R)} & y_W^{(R)} & z_W^{(R)} \end{pmatrix} = \underbrace{R_Z(A)}_{3.} \cdot \underbrace{R_Y(B)}_{2.} \cdot \underbrace{R_X(C)}_{1.}$$

$$= \begin{bmatrix} C_A & -S_A & 0 \\ S_A & C_A & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} C_B & 0 & S_B \\ 0 & 1 & 0 \\ -S_B & 0 & C_B \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_C & -S_C \\ 0 & S_C & C_C \end{bmatrix}$$

$$= \begin{bmatrix} C_A \cdot C_B & -S_A \cdot C_C + C_A \cdot S_B \cdot S_C & S_A \cdot S_C + C_A \cdot S_B \cdot C_C \\ S_A \cdot C_B & C_A \cdot C_C + S_A \cdot S_B \cdot S_C & -C_A \cdot S_C + S_A \cdot S_B \cdot C_C \\ -S_B & C_B \cdot S_C & C_B \cdot C_C \end{bmatrix}$$

$$B = \arcsin(-\mathbf{R}_{31})$$
$$A = \operatorname{atan2}(\mathbf{R}_{21}, \mathbf{R}_{11})$$
$$C = \operatorname{atan2}(\mathbf{R}_{32}, \mathbf{R}_{33})$$

Tipp

Da cos eine gerade Funktion ist, ergeben $\cos(\pm\alpha)$ den gleichen Wert, z.B. $\cos(\pm 45°) = \frac{1}{\sqrt{2}}$
sin ist ungerade und daher ist der sin-Wert immer verkehrt $\sin(\pm\alpha) = \mp a$

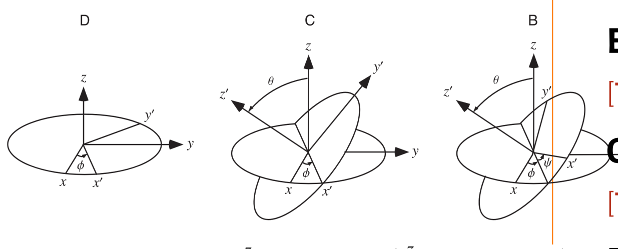
$$\operatorname{atan2}(x, y) = \begin{cases} \arctan\left(\frac{y}{x}\right) & \text{wenn } x > 0, \\ \arctan\left(\frac{y}{x}\right) + \pi & \text{wenn } x < 0 \text{ und } y \geq 0, \\ \arctan\left(\frac{y}{x}\right) - \pi & \text{wenn } x < 0 \text{ und } y < 0, \\ +\frac{\pi}{2} & \text{wenn } x = 0 \text{ und } y > 0, \\ -\frac{\pi}{2} & \text{wenn } x = 0 \text{ und } y < 0, \\ \text{undefiniert} & \text{wenn } x = 0 \text{ und } y = 0. \end{cases}$$

Roll Pitch Yaw (Roll Neigung Gier)

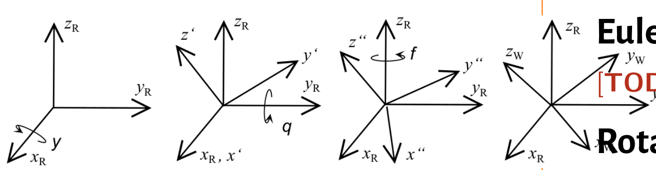
- Drehung um die x-**Basis**achse mit dem Winkel ψ (**yaw**)
- Drehung um die y-**Basis**achse mit dem Winkel θ (**pitch**)
- Drehung um die z-**Basis**achse mit dem Winkel ϕ (**roll**)

Unterschied RPY & Euler

Während bei Euler Drehungen immer vom neuen Koordinatensystem gedreht wird, werden bei RPY Drehungen immer vom Ursprünglichen Koordinatensystem gedreht



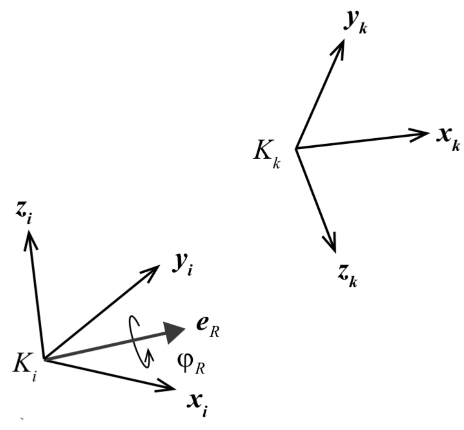
Euler Winkel



Roll Pitch Yaw

Orientierung durch Drehvektor und -winkel

Drehung wird um einen Drehvektor gemacht → den *direkten* Weg



$$\varphi_R = 2 \cdot \arccos(qt_1) \qquad e_R = \begin{bmatrix} qt_2 \\ qt_3 \\ qt_4 \end{bmatrix} / \sin(0.5 \cdot \varphi_R)$$

Da die Ermittelung des Drehvektors e_R und des Drehwinkels φ_R aufwendig ist, wird meistens mit Euler-Winkeln gearbeitet. Ein **Vorteil** von Drehvektor & -winkel bei der Bahnsteuerung/kontinuierliche Veränderung → Dies wird auch mit **Quarternions** gemacht!

Quaternion

Mit Quaternion können Drehungen in kompakter Form dargestellt werden, analog zu komplexen Zahlmultiplikation für Rotationen. Zusätzlich gibt es kein *Gimbal-Lock* (Zwei Rotationsachsen sind gleich und daher hat man ein Freiheitsgrad weniger). In AROB wird mit Einheitsquaternion gearbeitet: $\sqrt{qt_1^2 + qt_2^2 + qt_3^2 + qt_4^2} = 1$.

$$q = \begin{bmatrix} qt_1 \\ qt_2 \\ qt_3 \\ qt_4 \end{bmatrix} = \underbrace{qt_1}_{\text{real/Skalar Komp.}} + \underbrace{qt_2\ i + qt_3\ j + qt_4\ k}_{\text{imaginär/Vektor Komp.}}$$

[TODO]

Umwandlungen $Q \leftrightarrow EA \leftrightarrow {}^k_i A$

[TODO]

Euler zu Quartenion

[TODO]

Quartenion zu Euler

[TODO]

Euler zu Rotationsmatrix

[TODO]

Rotationsmatrix zu Euler

[TODO]

Quaternion zu Rotationsmatrix

[TODO]

Rotationsmatrix zu Quaternion

$$ {}^{k}_{i}A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} $$

$$ \begin{bmatrix} qt_1 \\ qt_2 \\ qt_3 \\ qt_4 \end{bmatrix} = \begin{bmatrix} 0.5 \cdot \sqrt{A_{11} + A_{22} + A_{33} + 1} \\ 0.5 \cdot \operatorname{sgn}(A_{32} - A_{23}) \cdot \sqrt{A_{11} - A_{22} - A_{33} + 1} \\ 0.5 \cdot \operatorname{sgn}(A_{13} - A_{31}) \cdot \sqrt{A_{22} - A_{33} - A_{11} + 1} \\ 0.5 \cdot \operatorname{sgn}(A_{21} - A_{12}) \cdot \sqrt{A_{33} - A_{11} - A_{22} + 1} \end{bmatrix} $$

Spaghett

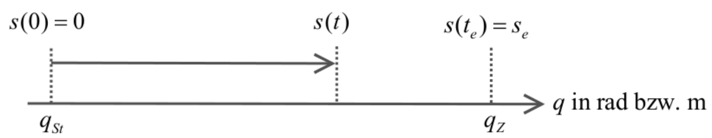
Bewegungsort & Interpolation

Steuerung und Gelenkregelung müssen dafür sorgen, dass die Gelenkkoordinaten die Werte der Zieillage einnehmen.

- Die Zielwerte für die Gelenke werden nicht sprungförmig geändert
- Drehmoment auf einem Maximalwert begrenzen (mechanische Belastung, Schwingungen)

Bahnparameter $s(t)$

Bahnparameter $s(t)$ - Zurückgelegte Wegstrecke bei einem Schubgelenk, bzw. der Winkel bei einem Drehgelenk



$$ S(t_e) = s_e = |q_z - q_{St}| $$

Software mit Schneebeli