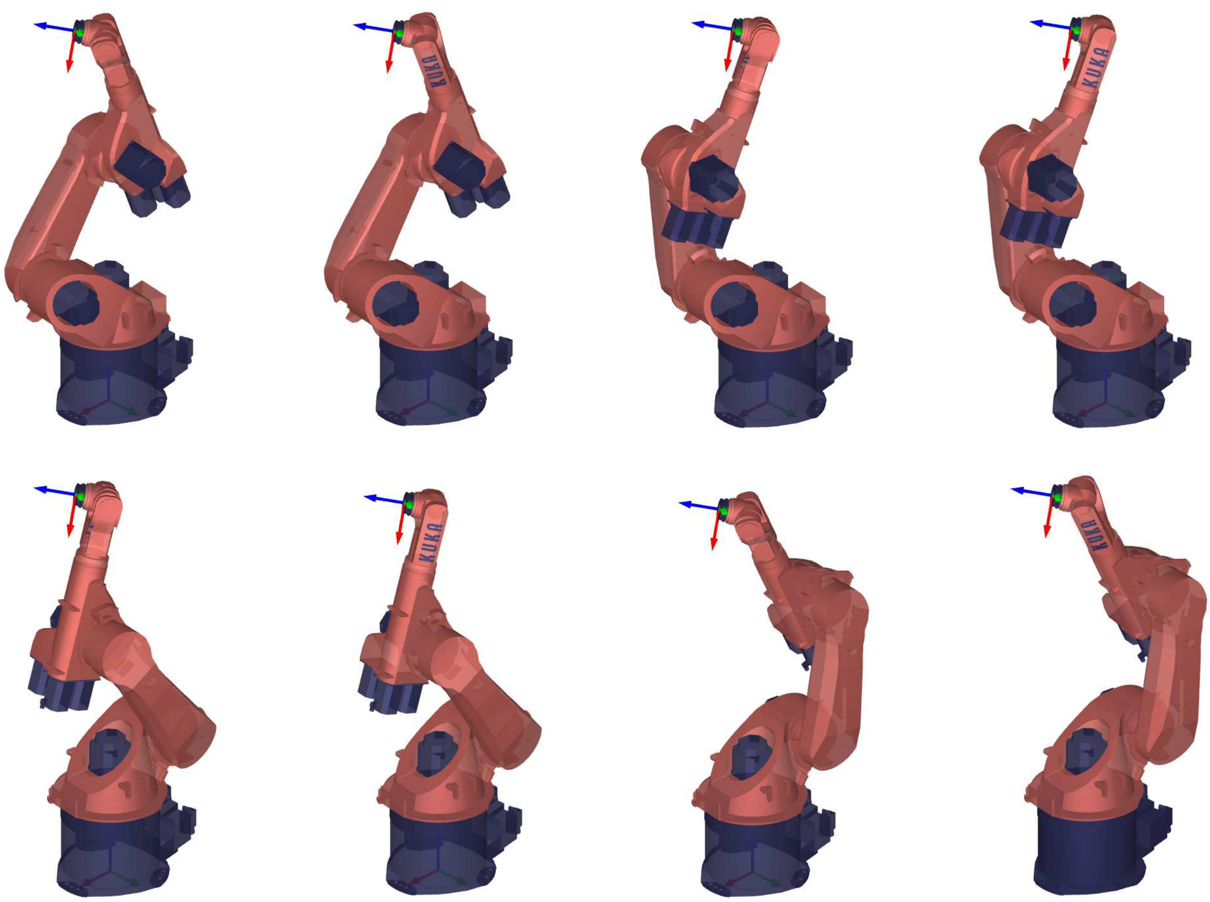
Rückwärtskinematik

Berechnung der Gelenkwinkel bei gegebener Lage und Orientierung des TCP (Inverse Funktion der Vorwärtskinematik)

Anzahl der Lösungen:

- keine
- genaue eine
- mehrere
- unendlich viele



Ansätze für Rückwärtskinematik:

geometrisch
algebraisch
} analytisch

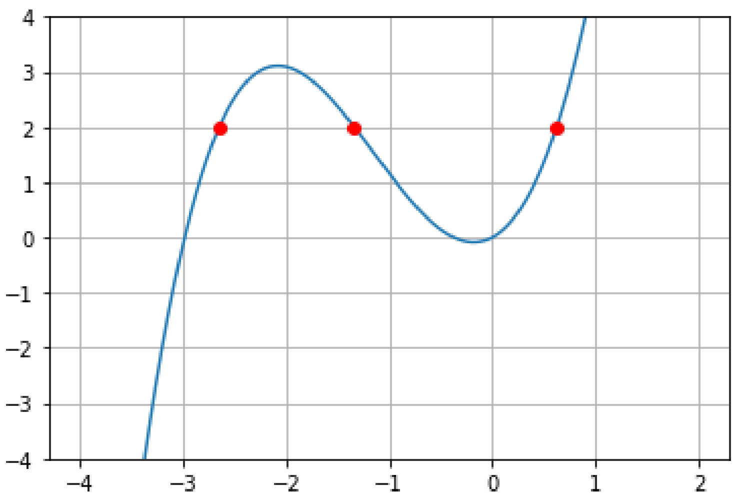
numerisch

Numerische Lösung

Einfaches mathematisches Beispiel

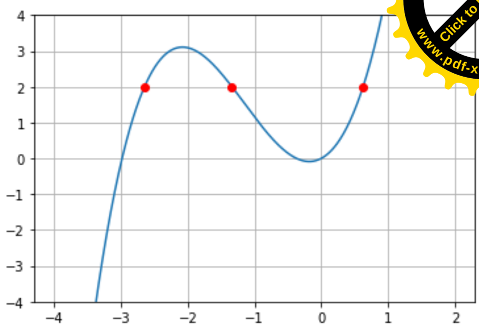
Gegeben ist eine nicht-lineare Funktion f(x)
Gesucht ist der Wert x, für den gilt f(x) = p
also die Inverse Funktion $x = f^{-1}(p)$

$$f(x) = \sin(x) + x^3 + 3x^2 = 2$$



Formulierung als Nullstellensuche

$$\sin(x) + x^3 + 3x^2 - 2 = 0$$



$f(x) = \sin(x) + x^3 + 3x^2 = 2$

Verwendung eines numerischen Verfahrens für die Nullstellensuche

Achtung: Welches Verfahren geeignet ist, kann nicht allgemein beantwortet werden!
Die Theorie hierfür reicht für eine eigene Vorlesung!

Die gefundene Lösung hängt von dem gewählten Startwert ab.

Python

```
# gesucht: f(x) = 2 mit f(x) = sin(x) + x^3 + 3*x^2

from scipy.optimize import fsolve
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(-4, 2, 0.01)


def f(x):
    return np.sin(x) + x**3 + 3*x**2


y = f(x)

plt.plot(x, y)
plt.ylim((-4, 4))
plt.grid()


def func(x):
    return f(x) - 2


x0 = -3
sol1 = fsolve(func, x0)
print(sol1)
plt.plot(sol1, f(sol1), 'ro')

x0 = -1
sol2 = fsolve(func, x0)
print(sol2)
plt.plot(sol2, f(sol2), 'ro')

x0 = 5
sol3 = fsolve(func, x0)
print(sol3)
plt.plot(sol3, f(sol3), 'ro')
```

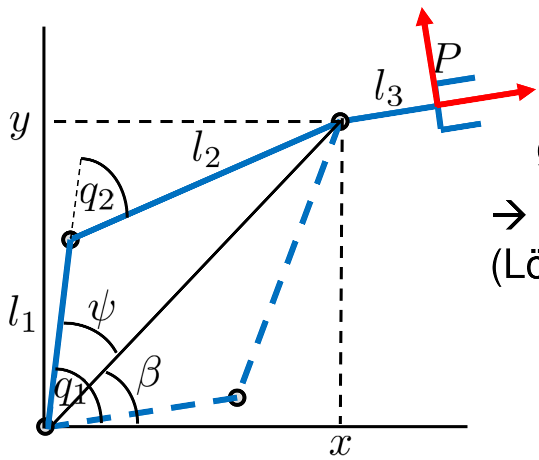
Numerische Lösung für die Rückwärtskinematik

Gesucht sind die Gelenkwinkel q bei gegebener Pose p des TCP

Formulierung als (6-dimensionale) Nullstellensuche

FK(q) – p = 0

Geometrische Lösung bei 3 parallelen Achsen



gegeben: x_P, y_P, ϕ

→ Berechnung von x, y möglich (Lösung für Achse 1, 2 zuerst)

mehrere Positionen möglich

Kosinussatz für Winkel 2

$$x^2 + y^2 = l_1^2 + l_2^2 - 2 l_1 l_2 \cos(180° + q_2)$$

(q2 ist für die erste Lösung negativ, daher 180° + q2)

Winkel 2

$$\cos(q_2) = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2 l_1 l_2}$$

Lösung existiert nur, wenn $\quad x^2 + y^2 \leq l_1 + l_2$

Winkel 1 über Hilfswinkel $\quad q_1 = \beta + \psi \quad$ für $\quad q_2 < 0$

mit $\quad \beta = atan2(y, x)$

$$\cos(\psi) = \frac{x^2 + y^2 + l_1^2 - l_2^2}{2 l_1 \sqrt{x^2 + y^2}} \qquad 0 \leq \psi \leq 180°$$

Zweite Lösung (q2 dreht entgegengesetzt)

→ $-q_2$

$q_1 = \beta - \psi \quad$ für $\quad q_2 > 0$

Winkel 3 über

$$q_1 + q_2 + q_3 = \phi$$

Eine analytische Lösung kann bei Robotern mit Standardkinematik gefunden werden.

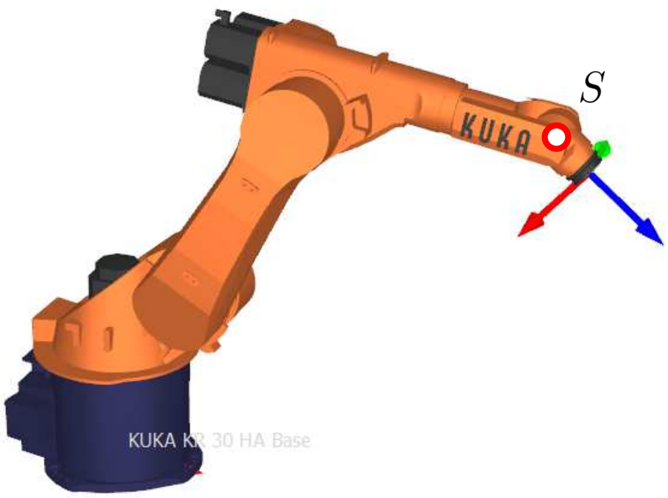
Voraussetzung: 3 Achsen schneiden sich in einem Punkt
(meist Handachsenschnittpunkt)

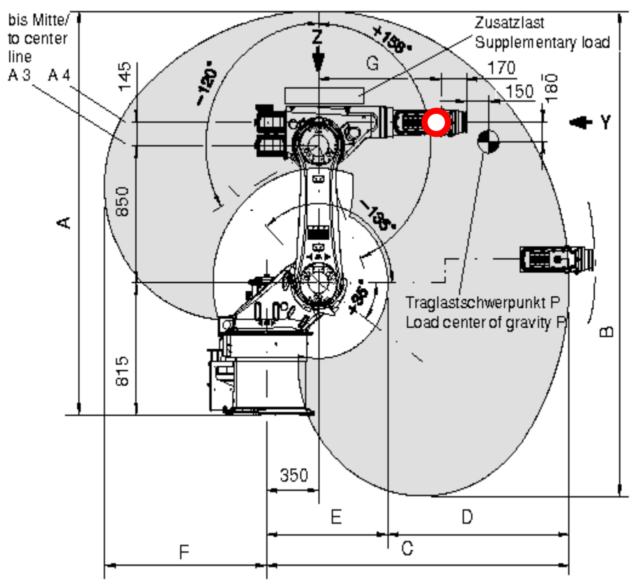
(oder: drei aufeinanderfolgende Achsen sind parallel)

Vorgehen:

1. Berechne Lage (x,y,z) des Handachsenschnittpunkts S
2. Berechne ersten drei Gelenkwinkel aus Lage von S
3. Berechne letzten drei Gelenkwinkel aus Orientierung

Handachsenschnittpunkt S





Detaillierte Rückwärtskinematik am Beispiel KUKA-Roboter

Gegeben ist x, y, z, A, B, C des TCP

→ Berechnung von $^{0}T_{6}$

Schnittpunkt S

Koordinaten von S im System 6 $$^{6}\tilde{s} = \begin{bmatrix} 0 \\ 0 \\ -170 \\ 1 \end{bmatrix}$$

Koordinaten von S im System 0 $$^{0}\tilde{s} = {}^{0}T_{6}\,{}^{6}\tilde{s}$$

Winkel Achse 1

(Sicht von oben)

$$q_1 = -atan2({}^{0}s_y, {}^{0}s_x)$$

Anpassung der Drehrichtung für KUKA Achse 1

Zweite Lösung: (front / rear oder left / right)

$$q_1' = q_1 - \pi$$

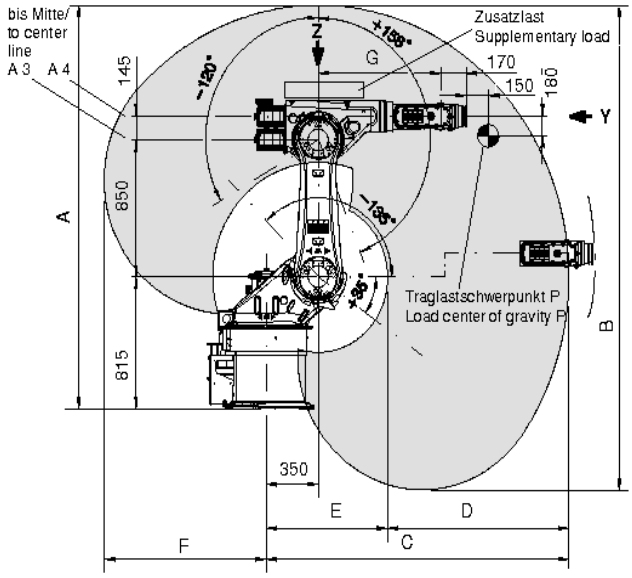
Hilfskoordinatensystem H

(in Achse 2 verschoben und um q1 gedreht → ebenes Problem in x-z-Ebene)

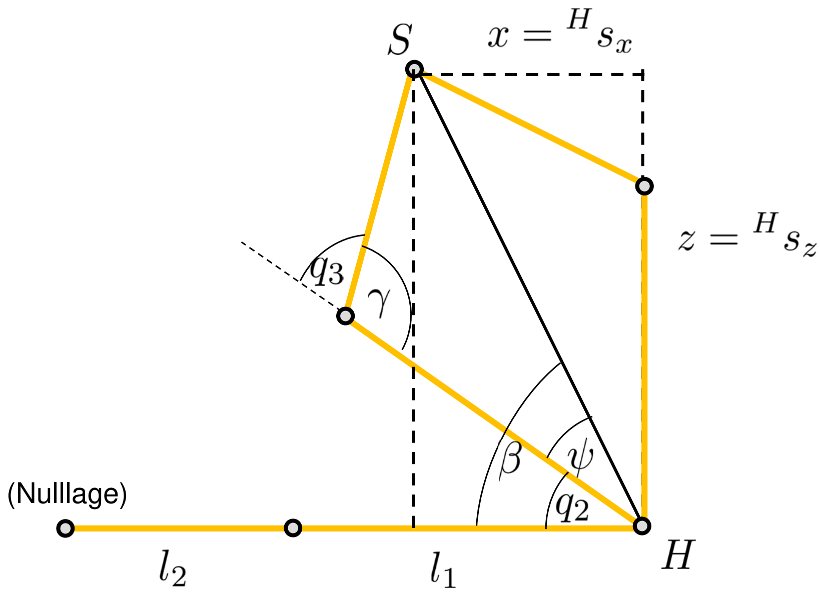
$$^{0}T_{H} = Rotz(-q_1) \cdot Transl(350, 0, 815)$$

Koordinaten von S im System H

$$^{H}\tilde{s} = {}^{H}T_{0}\,{}^{0}\tilde{s} = \left({}^{0}T_{H}\right)^{-1}\,{}^{0}\tilde{s}$$



Geometrische (Teil-) Lösung für zwei parallele Achsen (ohne Offset)



Kosinussatz

$$x^2 + z^2 = l_1^2 + l_2^2 - 2 l_1 l_2 \cos(\gamma)$$

Winkel Achse 3

$$\cos(\gamma) = -\frac{x^2 + z^2 - l_1^2 - l_2^2}{2 l_1 l_2}$$

Zwei Lösungen (elbow up / down):

$$q_3 = \pi - \gamma$$

$$q_3' = -q_3 = -\pi + \gamma$$

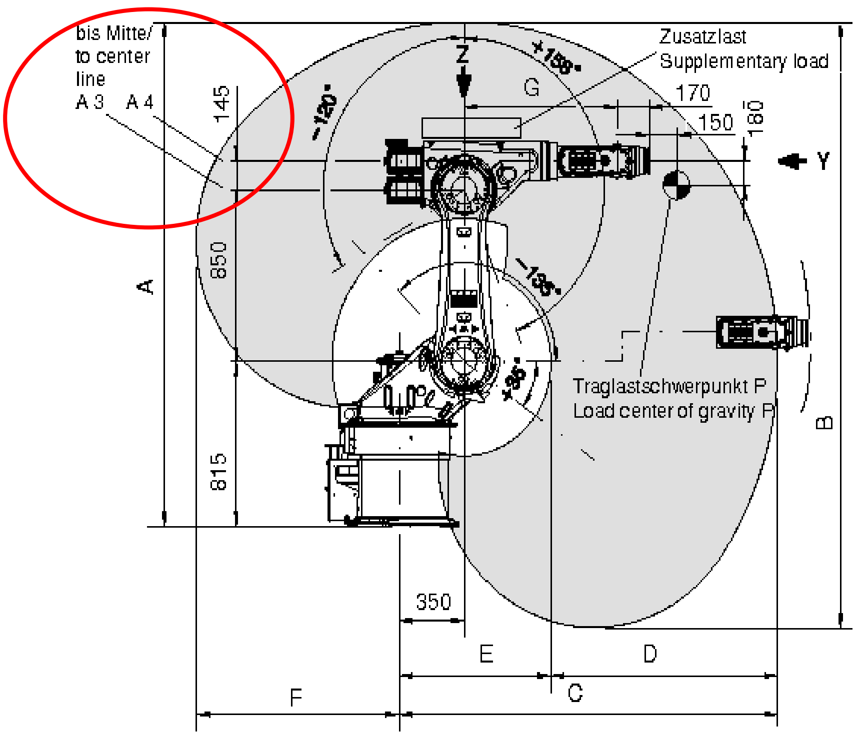
Winkel Achse 2

$$\beta = atan2({}^H s_z, {}^H s_x)$$

$$\psi = acos(\frac{x^2 + z^2 + l_1^2 - l_2^2}{2 l_1 \sqrt{x^2 + z^2}})$$

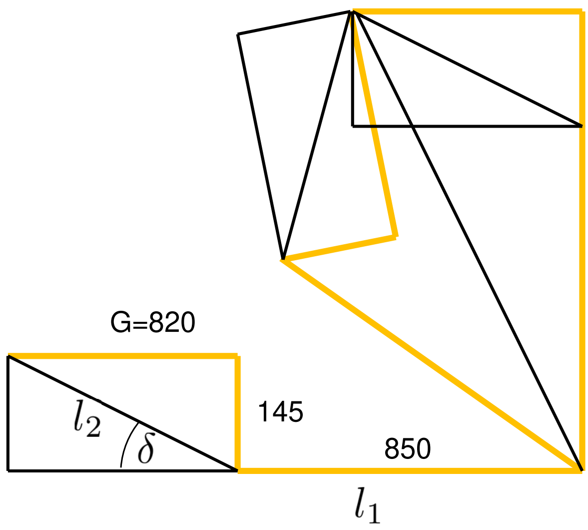
$$q_2 = \psi \pm \beta$$

Bei manchen Roboterherstellern muss ein Offset berücksichtigt werden



	A	B	C	D	E	F	G
KR 30-3	2498	3003	2033	1218	815	1084	820
KR 60-3	2498	3003	2033	1218	815	1084	820
KR 60 L45-3	2695	3398	2233	1362	868	1283	1020
KR 60 L30-3	2894	3795	2429	1445	983	1480	1220

Geometrische (Teil-) Lösung für zwei parallele Achsen (mit Offset)



Länge l2 mit Pythagoras, Offset Winkel mit atan2

Zwei Lösungen (elbow up / down):

$$q_3 = \pi - \gamma + \delta$$

$$q_3 = -\pi + \gamma + \delta$$

Orientierung

Winkel Achse 4, 5, 6

$$^3T_6 = {}^3T_0\, {}^0T_6$$

$$^3T_6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

aus DH-Parametern folgt

$$^3T_6 = {}^3T_4\, {}^4T_5\, {}^5T_6$$

$$R_6 = Rot_x(\alpha_3)Rot_z(\theta_4)Rot_x(\alpha_4)Rot_z(\theta_5)Rot_x(\alpha_5)Rot_z(\theta_6)$$

$$^3R_6 = Rot_x(-90^\circ)Rot_z(-q_4)Rot_x(90^\circ)Rot_z(q_5)Rot_x(-90^\circ)Rot_z(-q_6+180)$$

$$^3R_6 = \begin{bmatrix} c_4 & s_4 & 0 \\ 0 & 0 & 1 \\ s_4 & -c_4 & 0 \end{bmatrix} \begin{bmatrix} c_5 & -s_5 & 0 \\ 0 & 0 & -1 \\ s_5 & c_5 & 0 \end{bmatrix} \begin{bmatrix} -c_6 & -s_6 & 0 \\ 0 & 0 & 1 \\ -s_6 & c_6 & 0 \end{bmatrix}$$

$$^3R_6 = \begin{bmatrix} s_4s_6 - c_4c_5c_6 & -s_4c_6 - c_4c_5s_6 & -c_4s_5 \\ -s_5c_6 & -s_5s_6 & c_5 \\ -c_4s_6 - s_4c_5c_6 & c_4c_6 - s_4c_5s_6 & -s_4s_5 \end{bmatrix}$$

Winkel Achse 5

$$q_5 = \pm \arccos(r_{23})$$

Zwei Lösungen (non-flip / flip)

$$s_5 = \sin(q_5)$$

Winkel Achse 4

$$q_4 = atan2(-r_{33}/s_5, -r_{13}/s_5)$$

Winkel Achse 6

$$q_6 = atan2(-r_{22}/s_5, -r_{21}/s_5)$$

Detaillierte Rückwärtskinematik am Beispiel Universal Robots

PDF-XChange Editor Click to BUY NOW! www.pdf-xchange.com

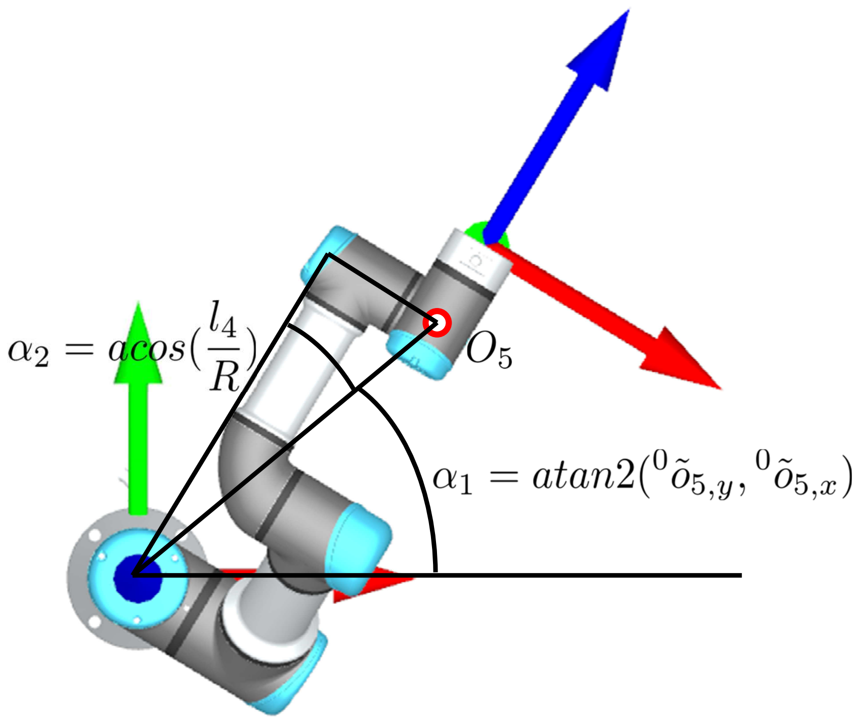
Gegeben ist x, y, z, rx, ry, rz des TCP

→ Berechnung von $\quad {}^{0}T_6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Ursprung O5 des System 5

Koordinaten von O5 im System 6 $\quad {}^{6}\tilde{o}_5 = \begin{bmatrix} 0 \\ 0 \\ -l_6 \\ 1 \end{bmatrix}$

Koordinaten von O5 im System 0 $\quad {}^{0}\tilde{o}_5 = {}^{0}T_6\,{}^{6}\tilde{o}_5$



Winkel q1

$$\alpha_1 = atan2({}^0\tilde{o}_{5,y}, {}^0\tilde{o}_{5,x})$$

$$R = \sqrt{{}^0\tilde{o}_{5,x}{}^2 + {}^0\tilde{o}_{5,y}{}^2}$$

$$\alpha_2 = acos(\frac{l_4}{R})$$

$$q_1 = \alpha_1 \pm \alpha_2 + \pi/2$$

Winkel q5

$$s_1 = \sin(q_1), c_1 = \cos(q_1)$$

$$q_5 = \pm acos(\frac{xs_1 - yc_1 - l_4}{l_6})$$

Winkel q6

$$s_5 = \sin(q_5), c_5 = \cos(q_5)$$

$$q_6 = atan2(\frac{-r_{12}s_1 + r_{22}c_1}{s_5}, \frac{r_{11}s_1 - r_{21}c_1}{s_5})$$

Berechnung von q2, q3, q4 über ebenes Problem mit drei parallelen Achsen

$${}^1T_4 = {}^1T_0\, {}^0T_6\, {}^6T_5\, {}^5T_4$$

Details siehe Paper:
Kelsey P. Hawkins: „Analytic Inverse Kinematics for the Universal Robots UR-5/UR-10 Arms“, 2013
http://hdl.handle.net/1853/50782