

Roboterkinematik

Vorwärtskinematik

Von Gelenkwinkeln θ zu Werkzeugposition / -orientierung X .

$$X = f(\theta)$$

Homogene Transformation

$${}^0\vec{r} = {}^0\vec{p} + {}^0R \cdot {}^1\vec{r}$$

Das $\vec{r}$ entspricht der Ausrichtung, das 0 bzw das 1 davor dem Ort 0 = Basis.

Das $\vec{p}$ ist die Translation und das R die Rotationsmatrix.

Meist schreibt man es aber so:

$$\begin{bmatrix} {}^0r_x \\ {}^0r_y \\ {}^0r_z \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} {}^1p_x \\ {}^1p_y \\ {}^1p_z \\ 1 \end{bmatrix} = \begin{bmatrix} {}^0\vec{r} \\ 1 \end{bmatrix} = {}^0T \cdot \begin{bmatrix} {}^1\vec{r} \\ 1 \end{bmatrix}$$

Das T ist die Transformationsmatrix, sie beinhaltet **Translation** wie auch **Rotation**.

Unterhalb der **Rotationsmatrix** wird eine Linie mit **0** eingefügt, überall **sonst** eine **1**.

Diese Transformationsmatrizen können auch verlinkt werden.

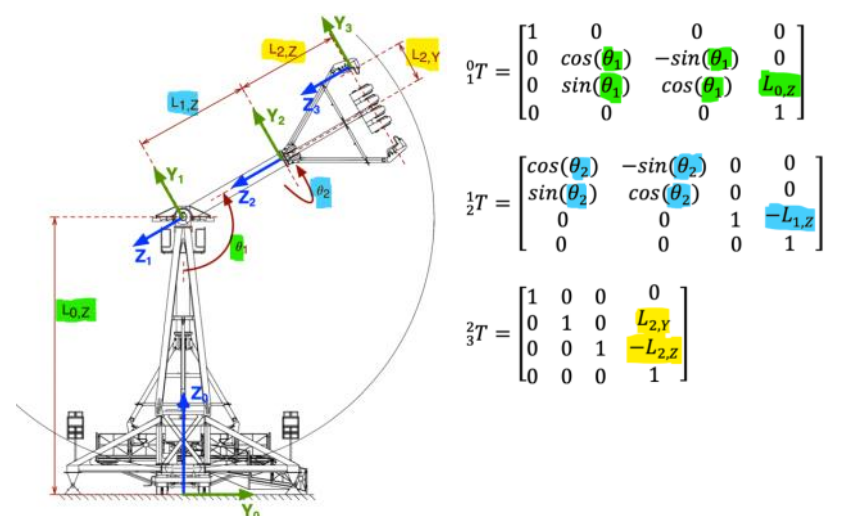
$$\begin{bmatrix} {}^0\vec{r} \\ 1 \end{bmatrix} = {}^0T \cdot {}^1T \cdot {}^2T \cdot {}^3T \cdot \begin{bmatrix} {}^4\vec{r} \\ 1 \end{bmatrix} = {}^0T \cdot \begin{bmatrix} {}^4\vec{r} \\ 1 \end{bmatrix}$$

Vorgehen

1. Zeichne Koordinatensysteme in jedem Gelenk, der Basis und dem Toolcenterpoint (TCP) ein.
2. Erstelle die Transformationsmatrizen zwischen den jeweiligen Koordinatensystemen.
3. Multipliziere die Transformationsmatrizen.
4. Aus der Transformationsmatrix kann nun die Position abgelesen werden (letzte Spalte).

Die Drehung ist in der Rotationsmatrix verpackt und kann schwer sein herauszulesen. Siehe dazu Kapitel Orientierung

Beispiel Freak Out



$${}^0T_3 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & -\sin(\theta_2) \cdot L_{2,Y} \\ \cos(\theta_1) \cdot \sin(\theta_2) & \cos(\theta_1) \cdot \cos(\theta_2) & -\sin(\theta_1) & \cos(\theta_1) \cdot \cos(\theta_2) \cdot L_{2,Y} + \sin(\theta_1) \cdot (L_{1,Z} + L_{2,Z}) \\ \sin(\theta_1) \cdot \sin(\theta_2) & \sin(\theta_1) \cdot \cos(\theta_2) & \cos(\theta_1) & \sin(\theta_1) \cdot \cos(\theta_2) \cdot L_{2,Y} - \cos(\theta_1) \cdot (L_{1,Z} + L_{2,Z}) + L_{0,Z} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$x = -\sin(\theta_2) \cdot L_{2,Y}$$

$$y = \cos(\theta_1) \cdot \cos(\theta_2) \cdot L_{2,Y} + \sin(\theta_1) \cdot (L_{1,Z} + L_{2,Z})$$

$$z = \sin(\theta_1) \cdot \cos(\theta_2) \cdot L_{2,Y} - \cos(\theta_1) \cdot (L_{1,Z} + L_{2,Z}) + L_{0,Z}$$

Denavit-Hartenberg Parameter

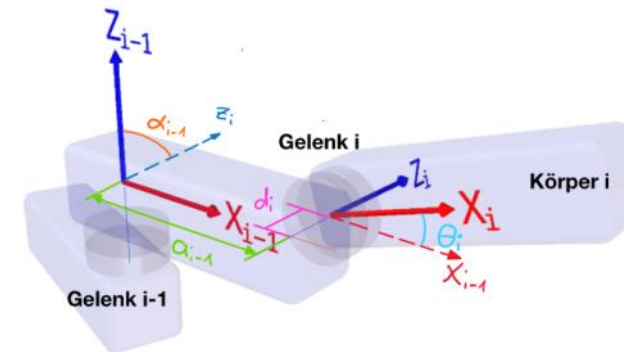
Voraussetzung:

- Es gibt nur **einen Freiheitsgrad** von **einem Gelenk** zum nächsten

Regeln

 (modifiziert durch J. Craig):

- Die N bewegten Körper werden von 1 bis N nummeriert, die Basis hat Nummer 0.
- Die N Gelenke des Körpers werden von 1 bis N nummeriert, wobei das Gelenk i zwischen den Körpern $i-1$ und i liegt.
- Die z_i - **Achse** liegt entlang der Gelenkachse i , sie entspricht der **Drehachse** bei **Drehgelenken** resp. der **Verschiebungsachse** bei **Translationsachsen**
- Die x_{i-1} - **Achse** steht **senkrecht** auf der z_i - **Achse** und schneidet diese
 - Kann dazu führen, dass der Ursprung nicht in dem Gelenk liegt, sondern gar ausserhalb des Roboters



Definition der 4 Denavit-Hartenberg Parameter:

- α_{i-1} ist der Winkel von z_{i-1} nach z_i um die Achse x_{i-1}
- d_i ist die Distanz von z_{i-1} bis z_i entlang der x_{i-1} Achse
- θ_i ist der Winkel von x_{i-1} nach x_i um die Achse z_i
- d_i ist die Distanz von x_{i-1} bis x_i entlang der z_i Achse

Mit diesen Parametern kann wieder eine **Transformationsmatrix** erstellt werden (ist **immer** die **gleiche**):

$${}^{i-1}T_i = T_{rot}(\vec{z}, \alpha_{i-1}) \cdot T_{trans}(\vec{x}, d_i) \cdot T_{rot}(\vec{z}, \theta_i) \cdot T_{trans}(\vec{x}, d_i)$$

$${}^{i-1}T_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \cos(\alpha_{i-1}) \cdot \sin(\theta_i) & \cos(\alpha_{i-1}) \cdot \cos(\theta_i) & -\sin(\alpha_{i-1}) & -d_i \cdot \sin(\alpha_{i-1}) \\ \sin(\alpha_{i-1}) \cdot \sin(\theta_i) & \sin(\alpha_{i-1}) \cdot \cos(\theta_i) & \cos(\alpha_{i-1}) & d_i \cdot \cos(\alpha_{i-1}) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

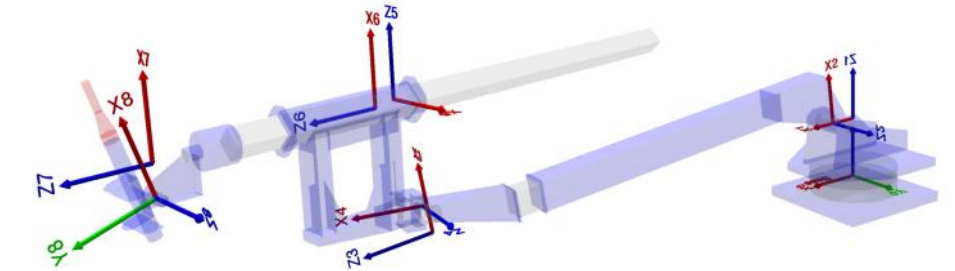
Vorteile

- Transformation kann mit 4 Parametern beschrieben werden anstelle 4x4-Matrix
- DH-Parameter werden von vielen Software-Tools für Modellierung von Robotern verwendet.

Nachteile

- Das Platzieren der Koordinatensystemen ist nicht einfach
- Das Verfahren ist nicht eindeutig: es existieren verschiedene Definitionen und es sind auch mehrere Lösungen möglich
- Basis und Werkzeug-Koordinatensysteme eines Roboters sind nicht vorgesehen und müssen separat berücksichtigt werden.

Beispiel



Gelenk	α_{i-1}	a_{i-1}	θ_i	d_i
1	0	0	θ_1	l_{1Z}
2	$-\pi/2$	l_{1X}	$\theta_2 - \pi/2$	0
3	$-\pi/2$	l_{2Z}	0	$l_{2X} + \theta_3$
4	$\pi/2$	l_{3Z}	$\theta_4 + \pi/2$	$-l_{4Y}$
5	$\pi/2$	l_{4X}	$\theta_5 + \pi/2$	l_{5Z}
6	$\pi/2$	l_{4Y}	$\pi/2$	θ_6
7	0	0	θ_7	l_{5X}
8	$\pi/2$	$-l_{7Z}$	θ_8	0

Hier sieht man das bei Gelenk 1 eine Translation (von Basis bis zum Ursprung des Koordinatensystems 1) erfolgt. Man sieht auch das der Ursprung nicht im Gelenk selber liegt sondern oberhalb auf derselben Höhe wie Gelenk 2.

Dazu gibt es noch einen Drehwinkel abhängig von dem Gelenkwinkel.

Rückwärtskinematik

Von Werkzeugposition / -orientierung X zu Gelenkwinkeln θ
 $\theta = f^{-1}(X)$

Symbolische Methoden

In der Regel für allgemeine 6-achsige Roboter nicht lösbar.
 Ausser Industrieroboter da dafür ausgelegt.

Vorgehen 6-achsiger Roboter (IRB 1100)

Bekannt:

- Orientierung und Position von TCP zu Basis:

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

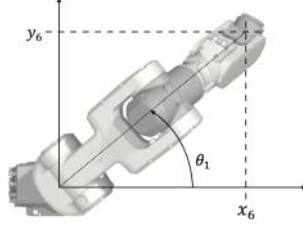
- Roboter Abmasse

- Aufstellen der Vorwärtskinematik
 - Muss ausmultipliziert werden
- Von TCP auf äusserstes Gelenk zurückrechnen

$${}^0_6 T = {}^0_{TCP} T \cdot {}^{TCP}_6 T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -l_6 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0_6 T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x-l_6 r_{11} \\ r_{21} & r_{22} & r_{23} & y-l_6 r_{21} \\ r_{31} & r_{32} & r_{33} & z-l_6 r_{31} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{matrix} x_6 = x - l_6 r_{11} \\ y_6 = y - l_6 r_{21} \\ z_6 = z - l_6 r_{31} \end{matrix}$$

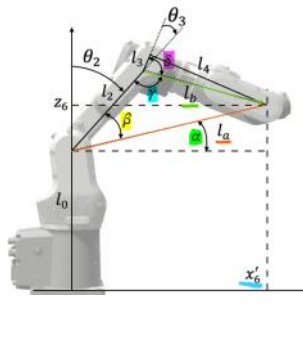
- Berechne θ_1 , θ_2 und θ_3 anhand Position Gelenk 6 und Roboterangaben (und Trigonometrie)



Achtung: es gibt 2 Lösungen!

$$\theta_{1,1} = \text{atan2}(y_6, x_6)$$

$$\theta_{1,2} = \text{atan2}(y_6, x_6) + \pi$$



$$x'_6 = \sqrt{x_6^2 + y_6^2}$$

$$l_a = \sqrt{x_6^2 + y_6^2 + (z_6 - l_0)^2}$$

$$l_b = \sqrt{l_2^2 + l_3^2}$$

$$\theta_2 = \frac{\pi}{2} - \beta - \gamma$$

$$\theta_2 = \frac{\pi}{2} - \arccos\left(\frac{l_2^2 + l_3^2 - l_a^2}{2l_2l_3}\right) - \arctan\left(\frac{z_6 - l_0}{x'_6}\right)$$

$$\theta_3 = \pi - \gamma - \delta$$

$$\theta_3 = \pi - \arccos\left(\frac{l_2^2 + l_3^2 - l_b^2}{2l_2l_3}\right) - \arctan\left(\frac{l_2}{l_3}\right)$$

- Nun ist die Rotationsmatrix von Basis zu Gelenk 3 bekannt, sowie Basis zu Gelenk 6 ist gegeben. Damit lässt sich die von Gelenk 3 bis 6 berechnen

$${}^3_6 R = {}^0_3 R^{-1} \cdot {}^0_6 R = \begin{bmatrix} \cos(\theta_1) \cdot \cos(\theta_2 + \theta_3) & \sin(\theta_1) \cdot \cos(\theta_2 + \theta_3) & -\sin(\theta_2 + \theta_3) \\ -\sin(\theta_1) & \cos(\theta_1) & 0 \\ \cos(\theta_1) \cdot \sin(\theta_2 + \theta_3) & \sin(\theta_1) \cdot \sin(\theta_2 + \theta_3) & \cos(\theta_2 + \theta_3) \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

- Wir kennen auch die Rotationsmatrix Gelenk 3 zu 6 aus der Geometrie.

$${}^3_6 R = \begin{bmatrix} \cos(\theta_5) & \sin(\theta_5) \cdot \sin(\theta_6) & \sin(\theta_5) \cdot \cos(\theta_6) \\ \sin(\theta_4) \cdot \sin(\theta_5) & \cos(\theta_4) \cdot \cos(\theta_6) - \sin(\theta_4) \cdot \sin(\theta_5) \cdot \sin(\theta_6) & -\cos(\theta_4) \cdot \sin(\theta_6) - \sin(\theta_4) \cdot \cos(\theta_5) \cdot \cos(\theta_6) \\ -\cos(\theta_4) \cdot \sin(\theta_5) & \sin(\theta_4) \cdot \cos(\theta_6) + \cos(\theta_4) \cdot \cos(\theta_5) \cdot \sin(\theta_6) & -\sin(\theta_4) \cdot \sin(\theta_6) + \cos(\theta_4) \cdot \cos(\theta_5) \cdot \cos(\theta_6) \end{bmatrix}$$

- Damit lässt sich nun nach den restlichen Winkeln auflösen

$$\theta_5 = \arccos(r_{11} \cos(\theta_1) \cos(\theta_2 + \theta_3) + r_{21} \cos(\theta_2 + \theta_3) \sin(\theta_1) - r_{31} \sin(\theta_2 + \theta_3))$$

$$\theta_4 = \text{atan2}\left(\frac{r_{21} \cos(\theta_1) - r_{11} \sin(\theta_1)}{\sin(\theta_5)}, \frac{r_{31} \cos(\theta_2 + \theta_3) + (r_{11} \cos(\theta_1) + r_{21} \sin(\theta_1)) \sin(\theta_2 + \theta_3)}{-\sin(\theta_5)}\right)$$

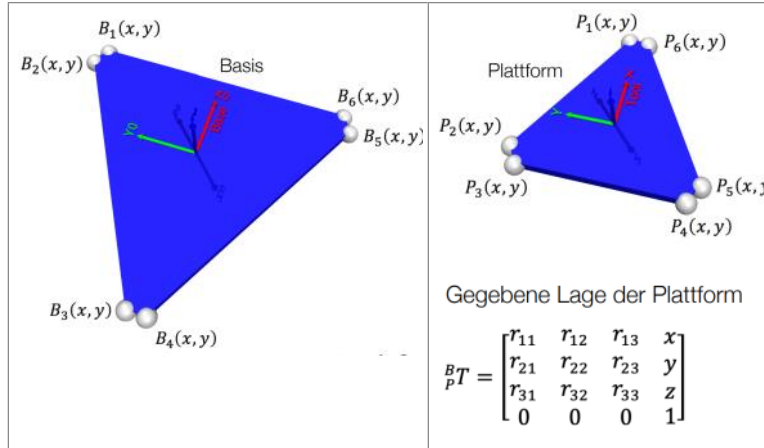
$$\theta_6 = \text{atan2}\left(\frac{r_{12} \cos(\theta_1) \cos(\theta_2 + \theta_3) + r_{22} \cos(\theta_2 + \theta_3) \sin(\theta_1) - r_{32} \sin(\theta_2 + \theta_3)}{\sin(\theta_5)}, \frac{r_{13} \cos(\theta_1) \cos(\theta_2 + \theta_3) + r_{23} \cos(\theta_2 + \theta_3) \sin(\theta_1) - r_{33} \sin(\theta_2 + \theta_3)}{\sin(\theta_5)}\right)$$

Damit erhält man eine von 8 möglichen Lösungen (Da θ_1 zwei mögliche Positionen hat)

Parallelroboter

Bei Parallelrobotern ist die Rückwärtskinematik meist trivial, während die Vorwärtskinematik schwer bis unmöglich ist.

Beispiel Hexapod



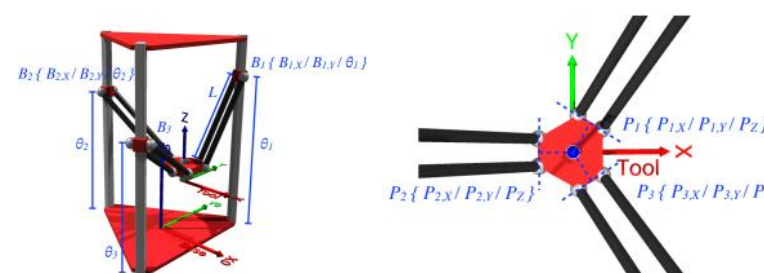
Mit der Lage der Plattform und dem Ankerpunkt kann eine Transformationsmatrix erstellt werden

$${}^B_P T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & P_{1,x} \\ 0 & 1 & 0 & P_{1,y} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{11}P_{1,x} + r_{12}P_{1,y} + x \\ r_{21} & r_{22} & r_{23} & r_{21}P_{1,x} + r_{22}P_{1,y} + y \\ r_{31} & r_{32} & r_{33} & r_{31}P_{1,x} + r_{32}P_{1,y} + z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Aus den Translationsbewegungen und dem Basisankerpunkt kann nun die Länge des ersten Aktuator bestimmt werden. (rinse and repeat)

$$\theta_1 = \sqrt{({}^B P_{1,x} - B_{1,x})^2 + ({}^B P_{1,y} - B_{1,y})^2 + ({}^B P_{1,z} - B_{1,z})^2}$$

Beispiel Linear-Delta Roboter



Gegeben:

$${}^B_P T = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^B P_{1,x} = P_{1,x} + x$$

$${}^B P_{1,y} = P_{1,y} + y$$

$${}^B P_{1,z} = P_{1,z} + z$$

Aus der T-Matrix links können die Positionen der Plattformgelenke bezüglich der Basis bestimmt werden. (siehe rechts)

Nun mit Trigonometrie und umstellen kommt man auf folgende Länge:

$$L^2 = ({}^B P_{1,x} - B_{1,x})^2 + ({}^B P_{1,y} - B_{1,y})^2 + ({}^B P_{1,z} - B_{1,z})^2$$

$$\theta_1 = {}^B P_{1,z} \pm \sqrt{L^2 - ({}^B P_{1,x} - B_{1,x})^2 - ({}^B P_{1,y} - B_{1,y})^2}$$

Numerische Methoden

Roboter bei welchen die Rückwärts-, bzw die Vorwärtskinematik nicht symbolisch hergeleitet werden können, wird ein numerisches Verfahren gewählt. z.B. **Newton-Raphson**

Eine Iterative Methode um den Gelenkwinkel mittels Vorwärts-, bzw Rückwärtskinematik und der inversen Jacobi-Matrix zu ermitteln.

Der Algorithmus zur Berechnung der Gelenkwinkel lautet wie folgt:

- Schätzen der Gelenkwinkel $\theta_{i,est}$
- Berechnen der Vorwärtskinematik $x_{i,est}$ (bei Parallel, die Rückwärtskinematik)
- Berechnen des Schätzfehlers in allen Kartesischen Koordinaten $\Delta x_i = x_i - x_{i,est}$
- Berechnen der inversen Jacobi-Matrix aus den geschätzten Gelenkwinkeln $\theta_{i,est}$
- Berechnen der Schätzfehler der Gelenkwinkel $\Delta \theta_i$ mit der inversen Jacobi-Matrix:

$$\begin{bmatrix} \Delta \theta_1 \\ \Delta \theta_2 \\ \vdots \\ \Delta \theta_n \end{bmatrix} = J^{-1}(\theta) \cdot \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_n \end{bmatrix}$$

- Damit erhält man eine neue Schätzung der Gelenkwinkel: $\theta_{i,est} + \Delta \theta_i$

Sinussatz

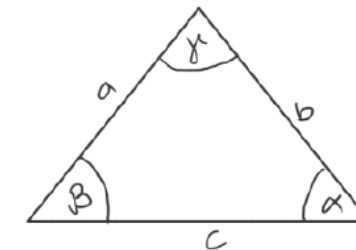
$$\frac{a}{\sin(\alpha)} = \frac{b}{\sin(\beta)} = \frac{c}{\sin(\gamma)}$$

Kosinussatz

$$a^2 = b^2 + c^2 - 2 \cdot b \cdot c \cdot \cos(\alpha)$$

$$b^2 = a^2 + c^2 - 2 \cdot a \cdot c \cdot \cos(\beta)$$

$$c^2 = a^2 + b^2 - 2 \cdot a \cdot b \cdot \cos(\gamma)$$



$$\alpha = \cos^{-1}\left(\frac{-a^2 + b^2 + c^2}{2bc}\right)$$

$$\beta = \cos^{-1}\left(\frac{-b^2 + a^2 + c^2}{2ac}\right)$$

$$\gamma = \cos^{-1}\left(\frac{-c^2 + a^2 + b^2}{2ab}\right)$$

$$\alpha = \arcsin\left(\frac{\sin(\beta) \cdot a}{b}\right) = \arcsin\left(\frac{a \cdot \sin(\gamma)}{c}\right)$$

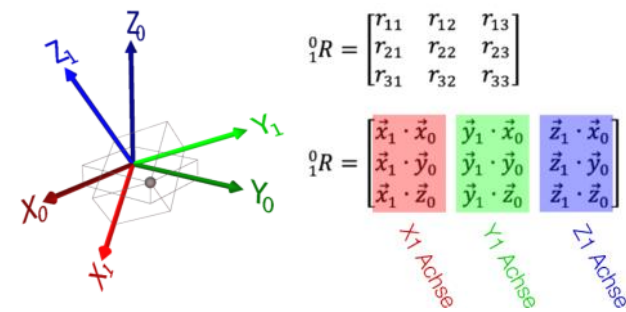
$$\beta = \arcsin\left(\frac{b \cdot \sin(\alpha)}{a}\right) = \arcsin\left(\frac{b \cdot \sin(\gamma)}{c}\right)$$

$$\gamma = \arcsin\left(\frac{\sin(\alpha) \cdot c}{a}\right) = \arcsin\left(\frac{c \cdot \sin(\beta)}{b}\right)$$

Orientierung

Rotationsmatrize

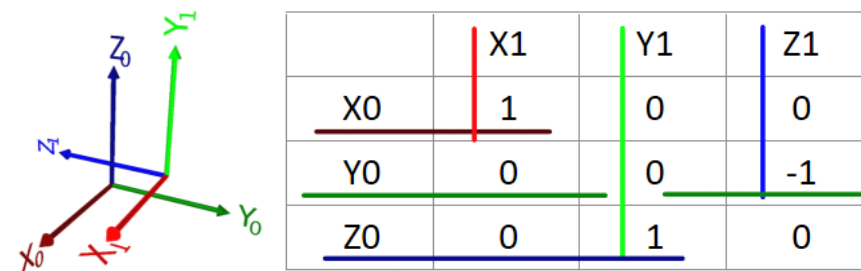
Beschreibt eine Drehung des Koordinatensystems



Rotation um jeweils eine Achse:

X-Achse	Y-Achse	Z-Achse
$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$	$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$	$R_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Bei n*90° Drehungen wie folgt vorgehen:



X1 und X0 sind nach wie vor dieselben, Y1 zeigt in Richtung von Z0 und Z1 in Richtung von -Y0

Eulerwinkel

Jede Rotation im 3D-Raum kann als Sequenz von höchstens 3 Rotationen um Koordinatenachsen betrachtet werden.

Aufeinanderfolgende Rotationen müssen um verschiedenen Achsen erfolgen.

X-Y-Z	Z-Y-X	X-Y-X	X-Z-X
Y-Z-X	Y-X-Z	Y-X-Y	Y-Z-Y
Z-X-Y	X-Z-Y	Z-X-Z	Z-Y-Z

Die Winkel, mit denen um die 3 Koordinatenachsen rotiert wird, nennt man **Eulerwinkel**. Wenn Rotationen um alle 3 Koordinatenachsen erfolgen, spricht man auch von **Kardanwinkel**.

Drehwinkel / -vektor

Ein Koordinatensystem kann auch mit einem Drehwinkel um einen gegebenen Einheitsvektor rotiert werden. Damit lassen sich auch beliebige Orientierungen im Raum beschreiben.

$$\theta \cdot \hat{w}, \quad \theta = \text{Drehwinkel}, \quad \hat{w} = \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \text{Drehvektor}$$

Aus der Rotationsmatrix:

$$\theta = \arccos\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right), \quad \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \frac{1}{2 \cdot \sin(\theta)} \cdot \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}$$

Quaternionen

3D Rotationen können auch als hyperkomplexe Zahl dargestellt werden:

$$\epsilon = \epsilon_0 + \epsilon_1 \cdot i + \epsilon_2 \cdot j + \epsilon_3 \cdot k, \quad i, j, k \text{ sind imaginäre Zahlen}$$

Quaternionen können aus einem gegebenen Drehwinkel und Drehvektor, oder auch aus einer gegebenen Rotationsmatrix berechnet werden:

aus Drehwinkel:

$$\epsilon_0 = \cos\left(\frac{\theta}{2}\right)$$

$$\epsilon_1 = w_x \cdot \sin\left(\frac{\theta}{2}\right)$$

$$\epsilon_2 = w_y \cdot \sin\left(\frac{\theta}{2}\right)$$

$$\epsilon_3 = w_z \cdot \sin\left(\frac{\theta}{2}\right)$$

aus Rotationsmatrix:

$$\epsilon_0 = \frac{1}{2} \sqrt{1 + r_{11} + r_{22} + r_{33}}$$

$$\epsilon_1 = \frac{r_{32} - r_{23}}{4\epsilon_0}$$

$$\epsilon_2 = \frac{r_{13} - r_{31}}{4\epsilon_0}$$

$$\epsilon_3 = \frac{r_{21} - r_{12}}{4\epsilon_0}$$

Retour zu Rotationsmatrix:

$$R = \begin{bmatrix} 1 - 2(\epsilon_2^2 + \epsilon_3^2) & 2(\epsilon_1\epsilon_2 - \epsilon_0\epsilon_3) & 2(\epsilon_1\epsilon_3 + \epsilon_0\epsilon_2) \\ 2(\epsilon_1\epsilon_2 + \epsilon_0\epsilon_3) & 1 - 2(\epsilon_1^2 + \epsilon_3^2) & 2(\epsilon_2\epsilon_3 - \epsilon_0\epsilon_1) \\ 2(\epsilon_1\epsilon_3 - \epsilon_0\epsilon_2) & 2(\epsilon_2\epsilon_3 + \epsilon_0\epsilon_1) & 1 - 2(\epsilon_1^2 + \epsilon_2^2) \end{bmatrix}$$

Geschwindigkeit

Jacobi-Matrizen

Zur Bestimmung der kartesischen Geschwindigkeit anhand der Gelenkwinkelgeschwindigkeit.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = J(\theta) \cdot \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_n \end{bmatrix} = \begin{bmatrix} \frac{dx_1}{d\theta_1} & \frac{dx_1}{d\theta_2} & \dots & \frac{dx_1}{d\theta_n} \\ \frac{dx_2}{d\theta_1} & \frac{dx_2}{d\theta_2} & \dots & \frac{dx_2}{d\theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{dx_n}{d\theta_1} & \frac{dx_n}{d\theta_2} & \dots & \frac{dx_n}{d\theta_n} \end{bmatrix} \cdot \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_n \end{bmatrix}$$

$\dot{\theta}$ = Gelenkgeschwindigkeit
 $\dot{x}_1, \dots, \dot{x}_{n/2}$ = Lineare Geschwindigkeit entlang der Achsen (x,y,z)
 $\dot{x}_{n/2+1}, \dots, \dot{x}_n$ = Rotationsgeschwindigkeit um die Achsen (x,y,z)

J_{11} entspricht also der Ableitung der x-Positionsfunktion nach dem ersten Gelenkwinkel θ_1 .

Beispiel ebener Roboter

$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3)$$

$$y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3)$$

$$\alpha = \theta_1 + \theta_2 + \theta_3$$

$$J(\theta) = \begin{bmatrix} -l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) - l_3 \sin(\theta_1 + \theta_2 + \theta_3) & -l_2 \sin(\theta_1 + \theta_2) - l_3 \sin(\theta_1 + \theta_2 + \theta_3) & -l_3 \sin(\theta_1 + \theta_2 + \theta_3) \\ l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) & l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) & l_3 \cos(\theta_1 + \theta_2 + \theta_3) \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \end{bmatrix} = J(\theta_1, \theta_2, \theta_3) \cdot \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

In der Jacobimatrix entspricht hier $x_1 = x$, $x_2 = y$ und $x_3 = \alpha$
 <-- so kommt man dann auf die kartesischen Geschwindigkeiten

Numerische Approximation

$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} \frac{\Delta x_1}{\Delta \theta_1} & \frac{\Delta x_1}{\Delta \theta_2} & \dots & \frac{\Delta x_1}{\Delta \theta_n} \\ \frac{\Delta x_2}{\Delta \theta_1} & \frac{\Delta x_2}{\Delta \theta_2} & \dots & \frac{\Delta x_2}{\Delta \theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\Delta x_n}{\Delta \theta_1} & \frac{\Delta x_n}{\Delta \theta_2} & \dots & \frac{\Delta x_n}{\Delta \theta_n} \end{bmatrix} \cdot \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_n \end{bmatrix}$	► Berechne x_1 bis x_n mit der Vorwärtskinematik aus gegebenen Gelenkwinkeln θ_1 bis θ_n ► Für alle Gelenkwinkel θ_i von θ_1 bis θ_n: ► Erhöhe den Gelenkwinkel θ_i um einen kleinen Wert, z.B. 0.000001 $\Rightarrow \theta_{i,diff}$ ► Berechne die Vorwärtskinematik aus $\theta_{1,diff}$ bis $\theta_{n,diff} \Rightarrow x_{1,diff}$ bis $x_{n,diff}$ ► Berechne die Elemente der Spalte i der Jacobi-Matrix: $(x_{1,diff} - x_1)/0.000001$ bis $(x_{n,diff} - x_n)/0.000001$
--	--

Singuläre Lagen

Mit der **inversen Jacobi-Matrix** kann aus der Geschwindigkeit des Werkzeuges die Gelenkgeschwindigkeit berechnet werden.

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_n \end{bmatrix} = J^{-1}(\theta) \cdot \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix}$$

Die Inverse kann unter folgenden Bedingungen berechnet werden:

- Die Jacobi-Matrix muss **quadratisch** sein
- Die Jacobi-Matrix muss **vollen Rang** haben
 - Die **Determinante** der Jacobi-Matrix **darf nicht 0** sein

Wenn die Determinante 0 ist dann befindet sich der Roboter in einer **singulären Lage**.

In dieser Lage ist die Beweglichkeit des Roboters in Kartesischen Koordinaten eingeschränkt.

Beispiele für singuläre Lagen:

- Strecklage des Roboterarmes



- Wenn 2 Drehachsen fluchten (gleiche Achse haben)
 - z.B. IRB 1100 Achse 4 und 6

Dynamik

Bewegungsgleichung

Inverse Dynamik:

$$F = m \cdot a + f_v \cdot v + f_c \cdot \text{sign}(v) + g$$

Labels: Beschleunigung, Masse, Geschwindigkeit, Gewicht, Kraft, viskoser Reibungskoeffizient, Coulomb'scher Reibungskoeffizient

Direkte Dynamik:

$$a = \frac{F - f_v \cdot v - f_c \cdot \text{sign}(v) - g}{m}$$

Dynamik von Mehrkörpersystemen

Allgemeine Form:

$$\tau = M(\theta) \cdot \ddot{\theta} + V(\theta, \dot{\theta}) \cdot \dot{\theta} + F(\dot{\theta}) + G(\theta) + J^T(\theta) \cdot f$$

Labels: Massenmatrix, Gelenkwinkel, Gelenkbeschleunigungen, Gewichtsmomente, Jacobi-Matrix, Gelenkmomente, Reibungsmomente, Gelenkgeschwindigkeiten, Kraftvektor am TCP, Geschwindigkeitsabhängige Terme

Direkte Dynamik

$$\ddot{\theta} = M^{-1}(\theta) \cdot (\tau - V(\theta, \dot{\theta}) \cdot \dot{\theta} - F(\dot{\theta}) - G(\theta) - J^T(\theta) \cdot f)$$

Kartesische Koordinaten

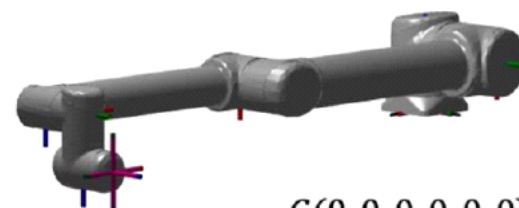
$$f = M(X) \cdot \ddot{X} + V(X, \dot{X}) \cdot \dot{X} + F(\dot{X}) + G(X)$$

$$\tau = J^T(X) \cdot (M(X) \cdot \ddot{X} + V(X, \dot{X}) \cdot \dot{X} + F(\dot{X}) + G(X))$$

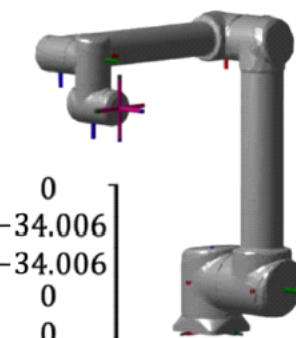
Gemischt

$$\tau = M(\theta) \cdot \ddot{\theta} + F(\dot{\theta}) + J^T(X) \cdot (M(X) \cdot \ddot{X} + V(X, \dot{X}) \cdot \dot{X} + G(X))$$

Gewichtsmomente



$$G(0, 0, 0, 0, 0, 0) = \begin{bmatrix} 0 \\ -120.80 \\ -34.01 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



$$G(0, -\frac{\pi}{2}, \frac{\pi}{2}, 0, 0, 0) = \begin{bmatrix} 0 \\ -34.006 \\ -34.006 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Herleitung Bewegungsgleichung

Newton-Euler

1. Berechne räumliche Geschwindigkeit und Beschleunigung von jedem Körper
 - a. Benutze dafür Gleichungen der Vorwärtskinematik
 - b. Kann Schrittweise ausgeführt werden, starte an der Basis gehe zu Werkzeug
2. Berechne alle aktiven Kräfte an allen Körpern
 - a. Gehe Schrittweise von dem Werkzeug bis zur Basis vor
 - b. Beachte das Kräftegleichgewicht.

Lagrange

Berechnen der kinetischen und potentieller Energie aus der gegebenen Bewegung und der aktuellen Lage des Roboters.

Mit partieller Ableitung dieser Energien erhält man das gesuchte Gelenkmoment.

$$\tau_i = \frac{d}{dt} \left(\frac{\partial K}{\partial \dot{\theta}_i} \right) - \frac{\partial K}{\partial \theta_i} + \frac{\partial P}{\partial \theta_i}$$

Vorgehen:

1. Berechne für alle Körper i die Energie:
 - a. Berechne Vorwärtskinematik bis zum Massenmittelpunkt jedes Körper i
 - b. Berechne Winkelgeschwindigkeiten $\omega_{x,i}$, $\omega_{y,i}$, und $\omega_{z,i}$
 - c. Berechne Geschwindigkeitsvektoren v_i
 - d. Berechne kinetische und potentielle Energien der Körper i
2. Berechne die Summe der kinetischen und potentieller Energien.
3. Leite diese Energien partiell ab um τ_i zu erhalten.

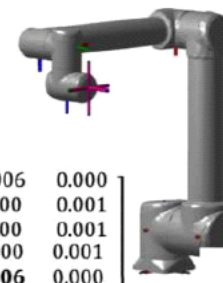
Kann auch verwendet werden bei Parallelrobotern im Gegensatz zum Newton-Euler Verfahren.

Massenmatrix

$$M(\theta) = \begin{bmatrix} 11.384 & 0.004 & 0.004 & 0.004 & -0.006 & 0.000 \\ 0.004 & 10.660 & 3.913 & 0.013 & 0.000 & 0.001 \\ 0.004 & 3.913 & 1.792 & 0.013 & 0.000 & 0.001 \\ 0.004 & 0.013 & 0.013 & 0.013 & 0.000 & 0.001 \\ -0.006 & 0.000 & 0.000 & 0.000 & 0.006 & 0.000 \\ 0.000 & 0.001 & 0.001 & 0.001 & 0.000 & 0.001 \end{bmatrix}$$



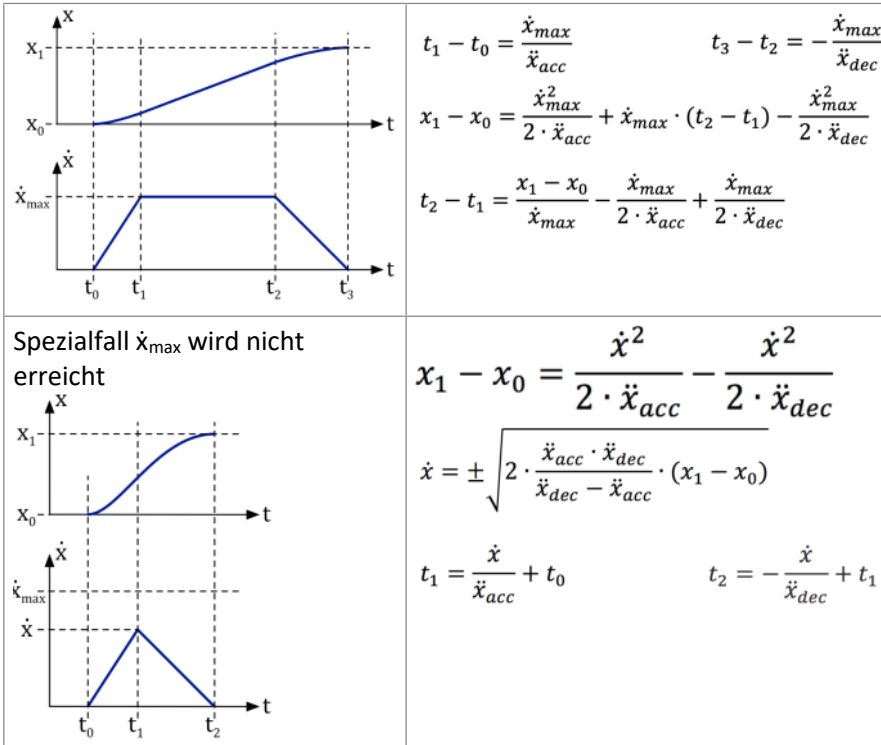
$$M(\theta) = \begin{bmatrix} 2.551 & -1.262 & 0.004 & 0.004 & -0.006 & 0.000 \\ -1.262 & 6.389 & 1.777 & -0.001 & 0.000 & 0.001 \\ 0.004 & 1.777 & 1.792 & 0.013 & 0.000 & 0.001 \\ 0.004 & -0.001 & 0.013 & 0.013 & 0.000 & 0.001 \\ -0.006 & 0.000 & 0.000 & 0.000 & 0.006 & 0.000 \\ 0.000 & 0.001 & 0.001 & 0.001 & 0.000 & 0.001 \end{bmatrix}$$



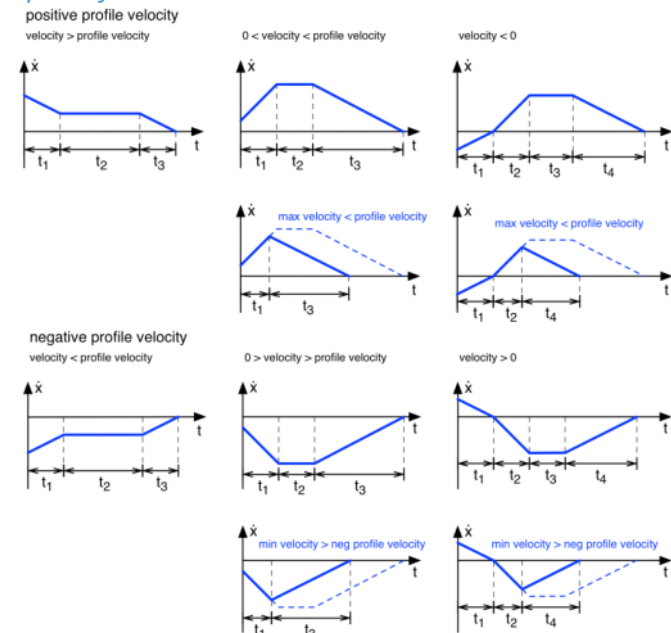
Bahnplanung

Funktionen für Trajektorien

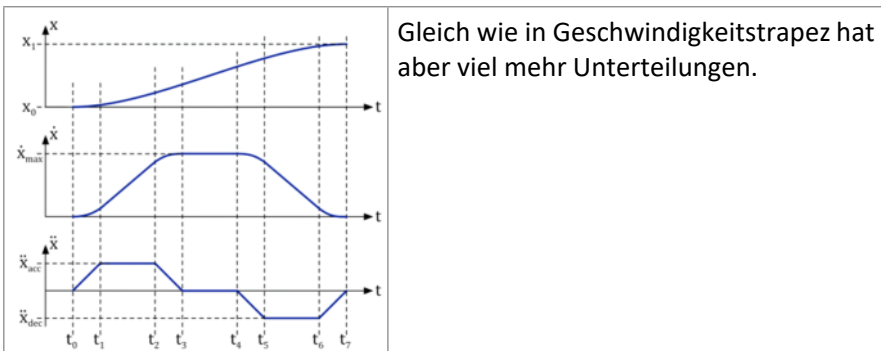
Geschwindigkeitstrapez



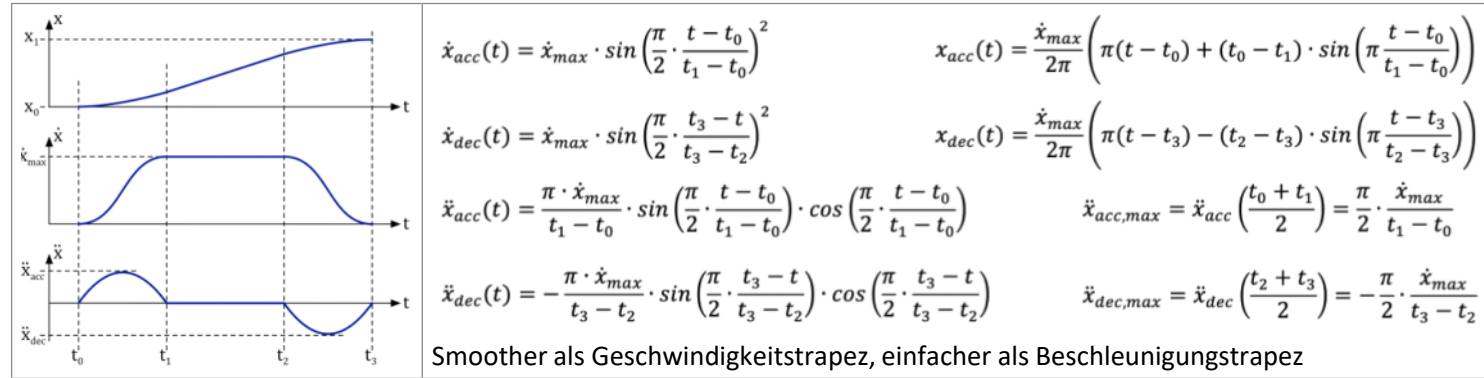
Spezialfälle:



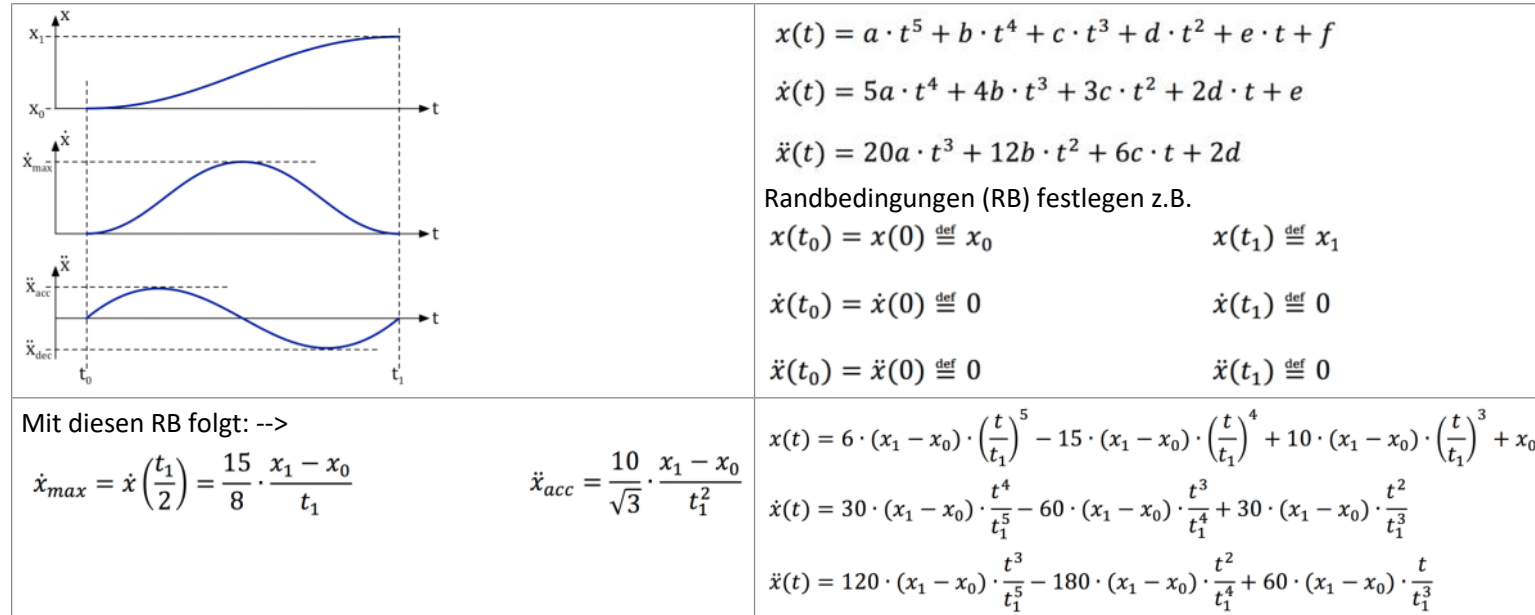
Beschleunigungstrapez



Sin² Beschleunigungsprofil



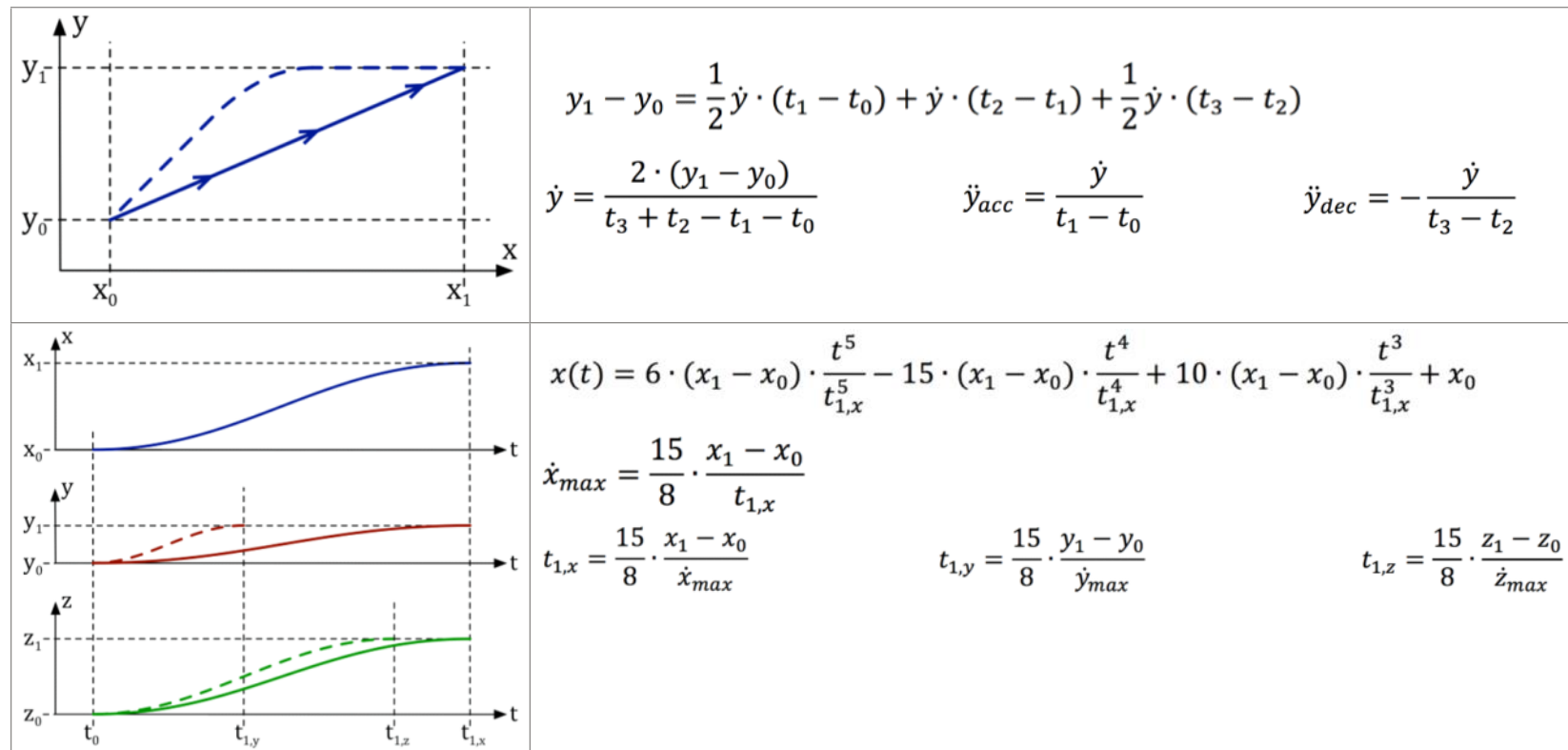
Polynom 5. Ordnung

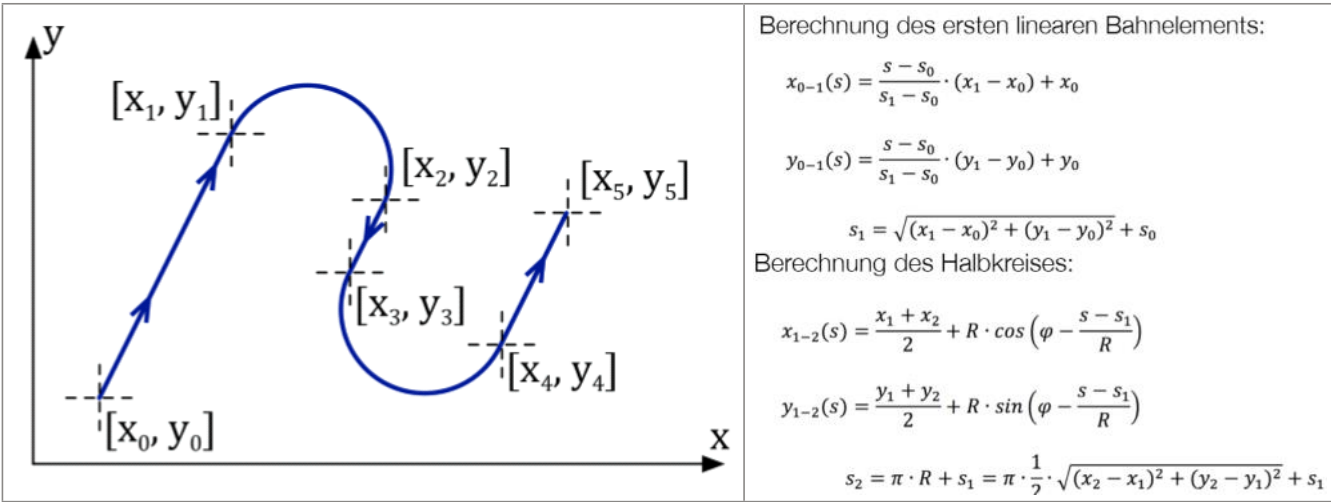


Allgemein

Für $t_0 \neq 0$ ist das ganze um einiges komplizierter.

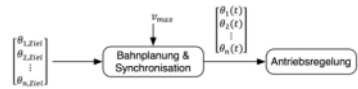
Synchronisation mehrerer Achsen



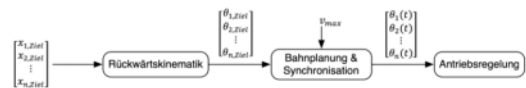


Gelenk- versus Kartesische Bewegung

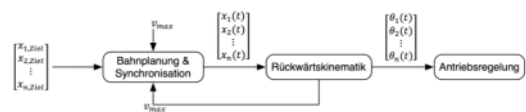
Bewegung zu Gelenkwinkeln mit einer Interpolation der Gelenkwinkel ⇒ **MoveAbsJ**



Bewegung zu Kartesischer Position mit einer Interpolation der Gelenkwinkel ⇒ **MoveJ**



Bewegung zu Kartesischer Position mit Interpolation Kartesischer Koordinaten ⇒ **MoveL**



Gelenkbewegung

- Kann durch **Singularitäten** fahren
- Kann von **beliebiger** Konfiguration in **beliebiger** Konfiguration fahren
- Kann **langsamer** als Kartesische sein

Kartesische Bewegung

- Erlaubt Werkzeug im **Prozesssystem** zu **bewegen**.
- Nur von **gleicher Konfiguration** zur **selben**
- **Nicht** durch **Singularitäten**