

Robotik und Mechatronik 1

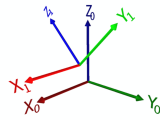
Fabian Starc

Version: January 20, 2026

Rotationsmatrizen

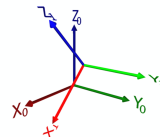
Rotation um die x-Achse:

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



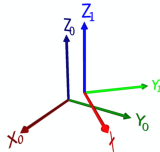
Rotation um die y-Achse:

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



Rotation um die z-Achse:

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

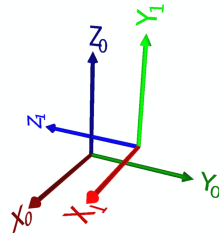


Verkettete Rotationen

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

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

x_1 -Achse bezüglich KOS_0
 y_1 -Achse bezüglich KOS_0
 z_1 -Achse bezüglich KOS_0



Rotationen können verkettet werden:

$${}^0_3R = {}^0_1R \cdot {}^1_2R \cdot {}^2_3R$$

Orientierung

Eulerwinkel

bsp. **Z-Y-Z** Eulerwinkel:

$$R_z(\alpha)R_y(\beta)R_z(\gamma) =$$

$$\begin{bmatrix} c\alpha \cdot c\beta \cdot c\gamma - s\alpha \cdot s\gamma & -c\alpha \cdot c\beta \cdot s\gamma - s\alpha \cdot c\gamma & c\alpha \cdot s\beta \\ s\alpha \cdot c\beta \cdot c\gamma + c\alpha \cdot s\gamma & -s\alpha \cdot c\beta \cdot s\gamma + c\alpha \cdot c\gamma & s\alpha \cdot s\beta \\ -s\beta \cdot c\gamma & s\beta \cdot s\gamma & c\beta \end{bmatrix}$$

Winkel α, β, γ bestimmen:

1. Zuerst β bestimmen
(am wenigsten "Vermischt" und kein Produkt mit anderen Winkeln):

$$\beta = \arccos(r_{33})$$

2. Finde Winkel welche denselben Ausdruck enthalten und teile sodass $\tan()$ entsteht:

$$r_{13} = \cos(\alpha) \sin(\beta) \quad r_{23} = \sin(\alpha) \sin(\beta) \quad \frac{r_{23}}{r_{13}} = \frac{\sin(\alpha) \sin(\beta)}{\cos(\alpha) \sin(\beta)}$$

$\Rightarrow \sin(\beta)$ kürzt sich raus und muss somit wieder in $\arctan2$ eingefügt werden

$$\text{Auflösen nach Alpha} \Rightarrow \alpha = \arctan2\left(\frac{r_{23}}{\sin(\beta)}, \frac{r_{13}}{\sin(\beta)}\right)$$

3. Wiederhole für Gamma γ :

$$\gamma = \arctan2\left(\frac{r_{32}}{\sin(\beta)}, \frac{-r_{31}}{\sin(\beta)}\right)$$

funktioniert nur wenn $\beta \neq 0$

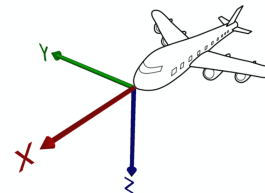
Roll-Pitch-Yaw (Flugwinkel)

$${}^0_1R = \begin{bmatrix} c\theta_p \cdot c\theta_y & s\theta_r \cdot s\theta_p \cdot c\theta_y - c\theta_r \cdot s\theta_y & c\theta_r \cdot s\theta_p \cdot c\theta_y + s\theta_r \cdot s\theta_y \\ c\theta_p \cdot s\theta_y & s\theta_r \cdot s\theta_p \cdot s\theta_y + c\theta_r \cdot c\theta_y & c\theta_r \cdot s\theta_p \cdot s\theta_y - s\theta_r \cdot c\theta_y \\ -s\theta_p & s\theta_r \cdot c\theta_p & c\theta_r \cdot c\theta_p \end{bmatrix}$$

$$\theta_p = \arcsin(-r_{31})$$

$$\theta_r = \arctan2\left(\frac{r_{32}}{\cos(\theta_p)}, \frac{r_{33}}{\cos(\theta_p)}\right)$$

$$\theta_y = \arctan2\left(\frac{r_{21}}{\cos(\theta_p)}, \frac{r_{11}}{\cos(\theta_p)}\right)$$



Drehwinkel und Drehvektor

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

Aus Rotationsmatrix R :

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

Quaternionen

$$\epsilon = \epsilon_0 + \epsilon_1 \cdot i + \epsilon_2 \cdot j + \epsilon_3 \cdot k$$

aus Drehwinkel:

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

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

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

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

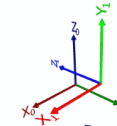
aus Rotationsmatrix:

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

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

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

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



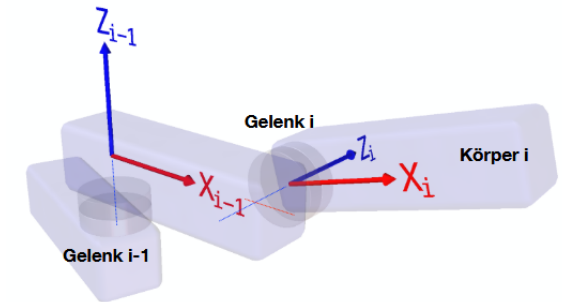
$$\epsilon = 0.70711 + 0.70711 \cdot i + 0 \cdot j + 0 \cdot k$$

$$\sqrt{2}/2 = 0.70711\dots$$

Denavit Hartenberg Parameter

Regeln zur Definition der Koordinatensysteme:

- Die N bewegten Körper des Roboters werden von 1 bis N nummeriert, die Basis hat die Nummer 0.
- Die N Gelenke des Roboters sind 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**.
- Die z_i Achse entspricht der **Verschiebungsachse bei Translationsgelenken**.
- Die x_{i-1} Achse steht senkrecht auf der z_i Achse und schneidet diese.



Definition der DH-Parameter:

- α_{i-1} : Verdrehung zwischen z_{i-1} und z_i um x_{i-1} Achse
- a_{i-1} : Abstand zwischen z_{i-1} und z_i entlang x_{i-1} Achse
- θ_i : Verdrehung von x_{i-1} und x_i um z_i Achse
- d_i : Distanz von x_{i-1} und x_i entlang z_i Achse

Transformationsmatrix zwischen System $(i-1)$ und System i :

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

Homogene Transformation

Die Rotation und die Translation können auch in 3D Koordinatensystemen in einer einzigen Matrix-Multiplikation zusammengefasst werden:

$${}^0\vec{r} = {}^0_1\vec{p} + {}^0_1R \cdot {}^1\vec{r} \quad \begin{bmatrix} {}^0\vec{r} \\ 1 \end{bmatrix} = \begin{bmatrix} {}^0_1R & {}^0_1\vec{p} \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} {}^1\vec{r} \\ 1 \end{bmatrix}$$

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

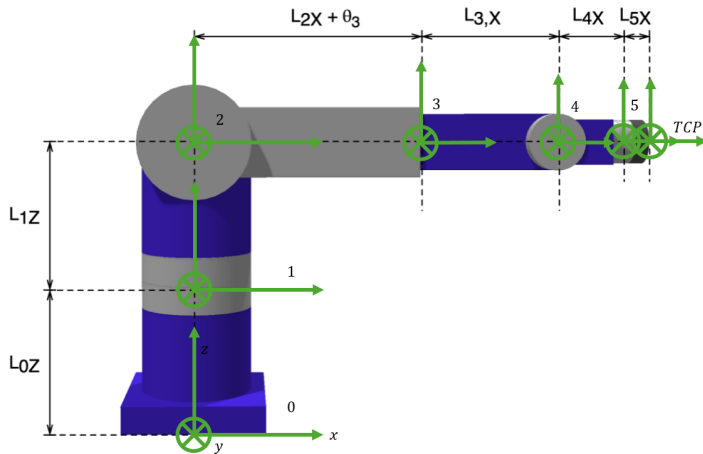
$$\begin{bmatrix} {}^0\vec{r} \\ 1 \end{bmatrix} = {}^0_1T \cdot \begin{bmatrix} {}^1\vec{r} \\ 1 \end{bmatrix}$$

Sequentielle Transformationen können auch verkettet werden:

$$\begin{bmatrix} {}^0\vec{r} \\ 1 \end{bmatrix} = {}^0_1T \cdot {}^1_2T \cdot {}^2_3T \cdot {}^3_4T \cdot \begin{bmatrix} {}^4\vec{r} \\ 1 \end{bmatrix} = {}^0_4T \cdot \begin{bmatrix} {}^4\vec{r} \\ 1 \end{bmatrix}$$

Vorwärtskinematik

Gleichungen für die Vorwärtskinematik mit homogenen Transformationen herleiten:



- geeignete KOS einführen
(bei n KOS werden $(n - 1)$ Transformationsmatrizen benötigt)
- Transformation von System a in System b :

${}^a_bT \Rightarrow$ Rotation in System b für Rotationsmatrix

${}^a_bT \Rightarrow$ Translation von System a nach System b für Translationsvektor

- Verkettung der Transformationsmatrizen:

$${}^0_{TCP}T = {}^0_1T \cdot {}^1_2T \cdot {}^2_3T \cdot {}^3_4T \cdot {}^4_5T \cdot {}^5_{TCP}T$$

Geschwindigkeiten

Jacobi-Matrix

Mit den Gleichungen der Vorwärtskinematik eines Roboters kann mit gegebenen Gelenkwinkeln die Lage eines Werkzeuges im Raum berechnet werden.

$$\dot{X} \Leftrightarrow \dot{\Theta} \quad \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} \Leftrightarrow \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_n \end{bmatrix}$$

Es gibt einen linearen Zusammenhang zwischen den Gelenkgeschwindigkeiten und der Werkzeuggeschwindigkeit mit der Jacobi-Matrix. Wenn die Gelenkgeschwindigkeiten mit der Jacobi-Matrix multipliziert werden, erhält man die Geschwindigkeit des Werkzeuges.

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

Die Elemente der Jacobi-Matrix sind partielle Ableitungen der Vorwärtskinematik nach den Gelenkwinkeln. Die Jacobi-Matrix ist damit abhängig von den aktuellen Gelenkwinkeln, und muss deshalb für jede Lage des Roboters berechnet werden.

Anstelle von partiellen Ableitungen können alternativ auch Differenzen berechnet werden, um die einzelnen Elemente der Jacobi-Matrix zu erhalten. Dies ermöglicht, die Jacobi-Matrix mit den Gleichungen der Vorwärtskinematik rein numerisch zu berechnen:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} \frac{\Delta x_1}{\Delta \theta_1} & \frac{\Delta x_1}{\Delta \theta_2} & \cdots & \frac{\Delta x_1}{\Delta \theta_n} \\ \frac{\Delta x_2}{\Delta \theta_1} & \frac{\Delta x_2}{\Delta \theta_2} & \cdots & \frac{\Delta x_2}{\Delta \theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\Delta x_n}{\Delta \theta_1} & \frac{\Delta x_n}{\Delta \theta_2} & \cdots & \frac{\Delta x_n}{\Delta \theta_n} \end{bmatrix} \cdot \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_n \end{bmatrix}$$

Inverse Jacobi-Matrix

Mit der inversen Jacobi-Matrix können aus einer gegebenen Geschwindigkeit des Werkzeuges die Gelenkgeschwindigkeiten berechnet werden, wie folgt:

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

Bedingungen:

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

Rückwärtskinematik

$$\theta = f^{-1}(X)$$

Mit den Gleichungen der Rückwärtskinematik können die Gelenkwinkel des Roboters aus einer gewünschten Position und Orientierung des Roboter-Werkzeuges berechnet werden.

Ebener Roboter

Die Rückwärtskinematik (Inverse Kinematik) beschreibt die Berechnung der Gelenkvariablen eines Roboters aus einer vorgegebenen Endeffektorposition. Für ebene Roboter erfolgt die Bewegung ausschließlich in der x - y -Ebene.

Geometrische Grundlagen

Gegeben sei die gewünschte Position des Endeffektors (TCP) im Basiskoordinatensystem:

$$P = \begin{bmatrix} X \\ Y \end{bmatrix}$$

Ein ebener Roboter mit zwei rotatorischen Gelenken (2R-Roboter) besitzt zwei Glieder mit den Längen L_1 und L_2 .

Abstand zum Endeffektor

Der Abstand r zwischen Basis und TCP ergibt sich zu:

$$r = \sqrt{X^2 + Y^2}$$

Für eine reale Lösung muss gelten:

$$|L_1 - L_2| \leq r \leq L_1 + L_2$$

Richtungswinkel

Der Winkel zwischen der x -Achse des Basissystems und der Verbindungslinie vom Ursprung zum TCP ist:

$$\beta = \text{atan2}(Y, X)$$

Berechnung der Gelenkwinkel

Mit dem Kosinussatz im Dreieck aus L_1 , L_2 und r ergibt sich:

$$\cos(\gamma) = \frac{L_1^2 + r^2 - L_2^2}{2L_1r}$$

$$\gamma = \arccos\left(\frac{L_1^2 + r^2 - L_2^2}{2L_1r}\right)$$

Der erste Gelenkwinkel besitzt zwei mögliche Lösungen:

$$\theta_1 = \gamma \pm \beta$$

Der zweite Gelenkwinkel ergibt sich aus:

$$\cos(\theta_2) = \frac{r^2 - L_1^2 - L_2^2}{2L_1L_2}$$

$$\theta_2 = \pm \arccos\left(\frac{r^2 - L_1^2 - L_2^2}{2L_1L_2}\right)$$

Mehrdeutigkeit der Lösung

Die Rückwärtskinematik ebener Roboter besitzt im Allgemeinen mehrere Lösungen. Diese entsprechen unterschiedlichen Konfigurationen (z.B. *Ellbogen oben* und *Ellbogen unten*).

Die Auswahl einer physikalisch gültigen Lösung erfolgt unter Berücksichtigung von:

- Gelenkgrenzen
- Kollisionsfreiheit
- Nähe zur aktuellen Konfiguration

Erweiterungen

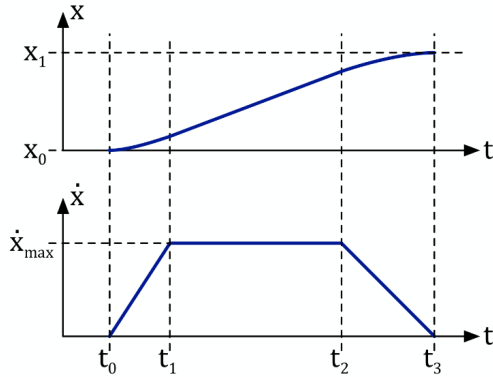
Die dargestellte Methode lässt sich auf:

- ebene SCARA-Roboter
- parallele ebene Mechanismen
- Roboter mit versetzten Basisgelenken

durch geeignete geometrische Verschiebungen und zusätzliche Winkelbeziehungen erweitern.

Bahnplanung

Geschwindigkeitstrapez



Beschleunigungsphase $t_0 \rightarrow t_1$ **Phase konstanter Geschwindigkeit** $t_1 \rightarrow t_2$

Verzögerungsphase $t_2 \rightarrow t_3$

Berechnung der Zeitdauer der Beschleunigungs- und Verzögerungsphase:

$$t_1 - t_0 = \frac{\dot{x}_{max}}{\ddot{x}_{acc}} \quad t_3 - t_2 = -\frac{\dot{x}_{max}}{\ddot{x}_{dec}}$$

Berechnung der Zeitdauer der Phase mit konstanter Geschwindigkeit:

$$x_1 - x_0 = \frac{\dot{x}_{max}^2}{2\ddot{x}_{acc}} + \dot{x}_{max} \cdot (t_2 - t_1) - \frac{\dot{x}_{max}^2}{2\ddot{x}_{dec}}$$

$$t_2 - t_1 = \frac{x_1 - x_0}{\dot{x}_{max}} - \frac{\dot{x}_{max}}{2\ddot{x}_{acc}} + \frac{\dot{x}_{max}}{2\ddot{x}_{dec}}$$

Berechnung der zurückgelegten Strecke:

$$x_1 - x_0 = \frac{\dot{x}^2}{2\ddot{x}_{acc}} - \frac{\dot{x}^2}{2\ddot{x}_{dec}}$$

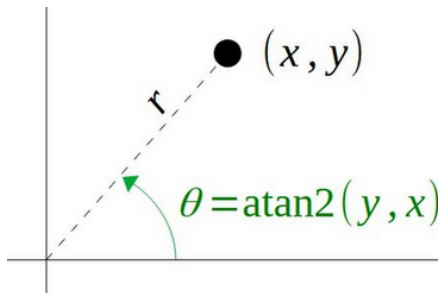
Berechnung der maximalen Geschwindigkeit:

$$\dot{x} = \pm \sqrt{2 \cdot \frac{\ddot{x}_{acc} \cdot \ddot{x}_{dec}}{\ddot{x}_{dec} - \ddot{x}_{acc}} \cdot (x_1 - x_0)}$$

Berechnung der Zeitdauer von der Beschleunigung und der Verzögerung:

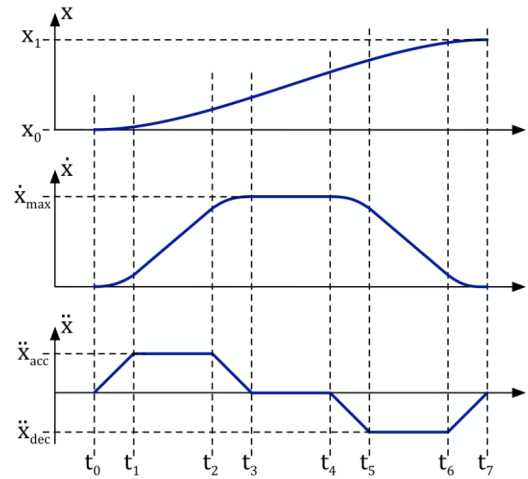
$$t_1 = \frac{\dot{x}}{\ddot{x}_{acc}} + t_0 \quad t_2 = -\frac{\dot{x}}{\ddot{x}_{dec}} + t_1$$

arctan2



$$\vec{r} = \begin{pmatrix} x \\ y \end{pmatrix} \quad \theta = \arctan2(y, x)$$

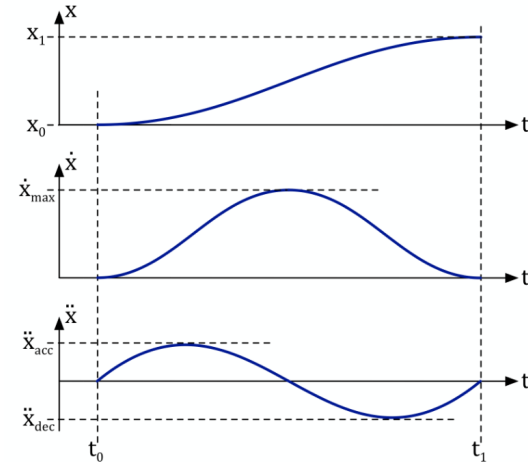
Beschleunigungstrapez



Diese Trajektorie verwendet Trapeze für den Verlauf der Beschleunigung.

Die Änderung der Beschleunigung erfolgt mit konstantem Ruck. Als **Ruck** wird die **Ableitung der Beschleunigung nach der Zeit** bezeichnet.

Polynom 5. Ordnung



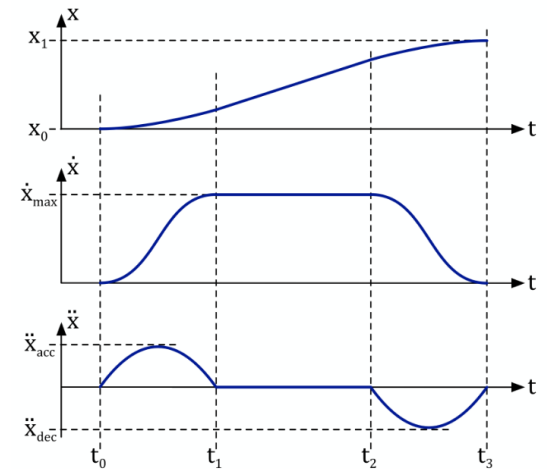
$$x(t) = at^2 + bt^4 + ct^3 + dt^2 + et^2 + f$$

$$\dot{x}(t) = 5at^4 + 4bt^3 + 3ct^2 + 2dt + e$$

$$\ddot{x}(t) = 20at^3 + 12bt^2 + 6ct + 2d$$

Um die 6 Parameter $a \rightarrow f$ zu bestimmen, müssen 6 Randbedingungen definiert werden.

sin-Quadrat Geschwindigkeitsprofil



Diese Trajektorie führt zu sanfteren Bewegungen als das Geschwindigkeitstrapez, ist aber viel einfacher zu berechnen als die Trajektorie mit Beschleunigungstrapezen.

$$\dot{x}_{acc}(t) = \dot{x}_{max} \cdot \sin\left(\frac{\pi}{2} \cdot \frac{t - t_0}{t_1 - t_0}\right)^2$$

$$x_{acc}(t) = \frac{\dot{x}_{max}}{2\pi} \left(\pi(t - t_0) + (t_0 - t_1) \cdot \sin\left(\pi \frac{t - t_0}{t_1 - t_0}\right) \right)$$

$$\dot{x}_{dec}(t) = \dot{x}_{max} \cdot \sin\left(\frac{\pi}{2} \cdot \frac{t_3 - t}{t_3 - t_2}\right)^2$$

$$x_{dec}(t) = \frac{\dot{x}_{max}}{2\pi} \left(\pi(t - t_3) - (t_2 - t_3) \cdot \sin\left(\pi \frac{t - t_3}{t_2 - t_3}\right) \right)$$

$$\ddot{x}_{acc}(t) = \frac{\pi \cdot \dot{x}_{max}}{t_1 - t_0} \cdot \sin\left(\frac{\pi}{2} \cdot \frac{t - t_0}{t_1 - t_0}\right) \cdot \cos\left(\frac{\pi}{2} \cdot \frac{t - t_0}{t_1 - t_0}\right)$$

$$\ddot{x}_{acc,max} = \ddot{x}_{acc}\left(\frac{t_0 + t_1}{2}\right) = \frac{\pi}{2} \cdot \frac{\dot{x}_{max}}{t_1 - t_0}$$

$$\ddot{x}_{dec}(t) = -\frac{\pi \cdot \dot{x}_{max}}{t_3 - t_2} \cdot \sin\left(\frac{\pi}{2} \cdot \frac{t_3 - t}{t_3 - t_2}\right) \cdot \cos\left(\frac{\pi}{2} \cdot \frac{t_3 - t}{t_3 - t_2}\right)$$

$$\ddot{x}_{dec,max} = \ddot{x}_{dec}\left(\frac{t_2 + t_3}{2}\right) = -\frac{\pi}{2} \cdot \frac{\dot{x}_{max}}{t_3 - t_2}$$

Kosinussatz für ebene Dreiecke

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

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

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

Winkel explizit:

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

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

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

