Bahnsteuerung

Prof. Dr.-Ing. Rüdiger Dillmann
Dr.-Ing. Sven R. Schmidt-Rohr
Dr.-Ing. Rainer Jäkel

Inhalt

- **Grundlagen der Bahnsteuerung**
- Programmierung der Schlüsselpunkte
- Interpolationsarten
- Approximierte Bahnsteuerung

Grundlagen der Bahnsteuerung

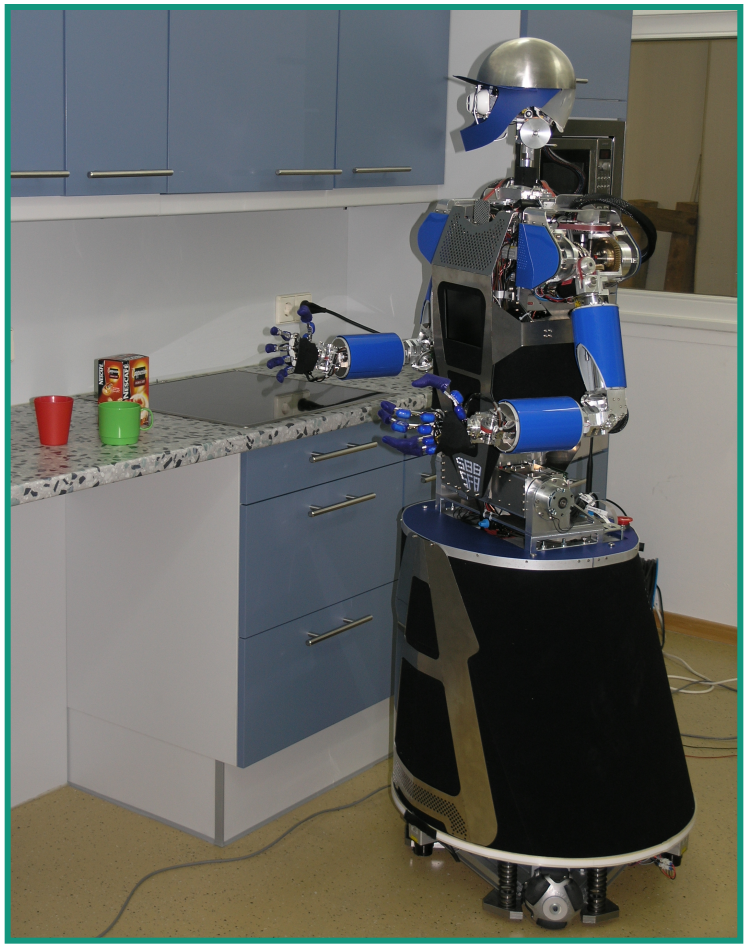
Definition: Trajektorie

- Bewegungen eines Roboters werden aufgefasst als
  - Zustandsänderungen
    - über der Zeit
    - relativ zu einem stationären Koordinatensystem (kartesischer Raum, Gelenkwinkelraum)
  - mit Einschränkungen durch
    - Zwangsbedingungen
    - Gütekriterien
    - Neben- und Randbedingungen

Grundlagen der Bahnsteuerung

Problem

- Gegeben
  - S_{Start} : Zustand zum Startzeitpunkt
  - S_{Ziel} : Zustand zum Zielzeitpunkt
- Gesucht
  - S_i : Zwischenzustände (Stützpunkte), damit die Trajektorie „glatt“ und stetig wird.



Grundlagen der Bahnsteuerung

Beispiel für ein Gelenk

- Anfangsbedingungen

$$q(t_o) = 15\,Grad$$

$$\dot{q}(t_o) = 0\,\frac{Grad}{\sec}$$

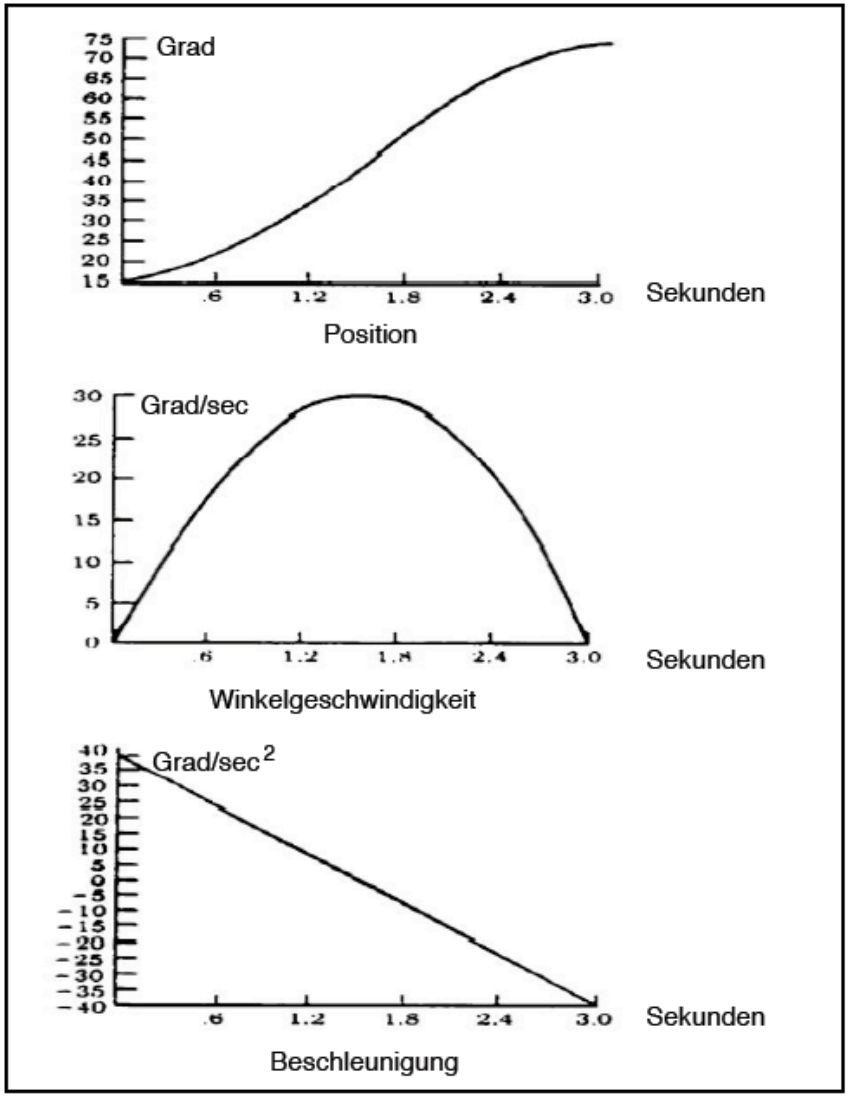
$$\ddot{q}(t_o) = 40\,\frac{Grad}{\sec^2}$$

- Endbedingungen

$$q(t_e) = 75\,Grad$$

$$\dot{q}(t_e) = 0\,\frac{Grad}{\sec}$$

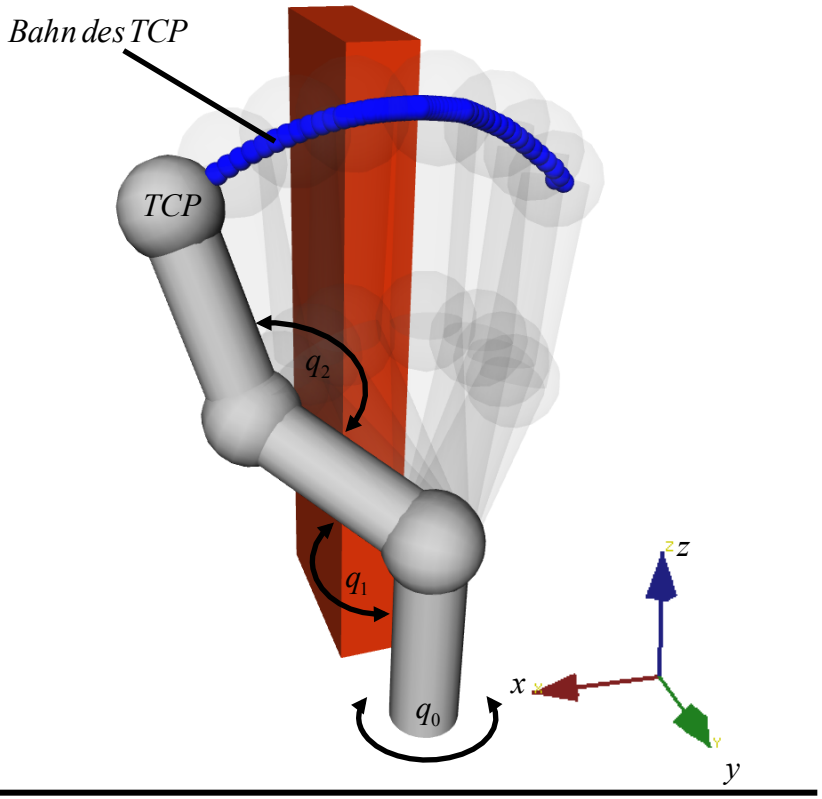
$$\ddot{q}(t_e) = -40\,\frac{Grad}{\sec^2}$$



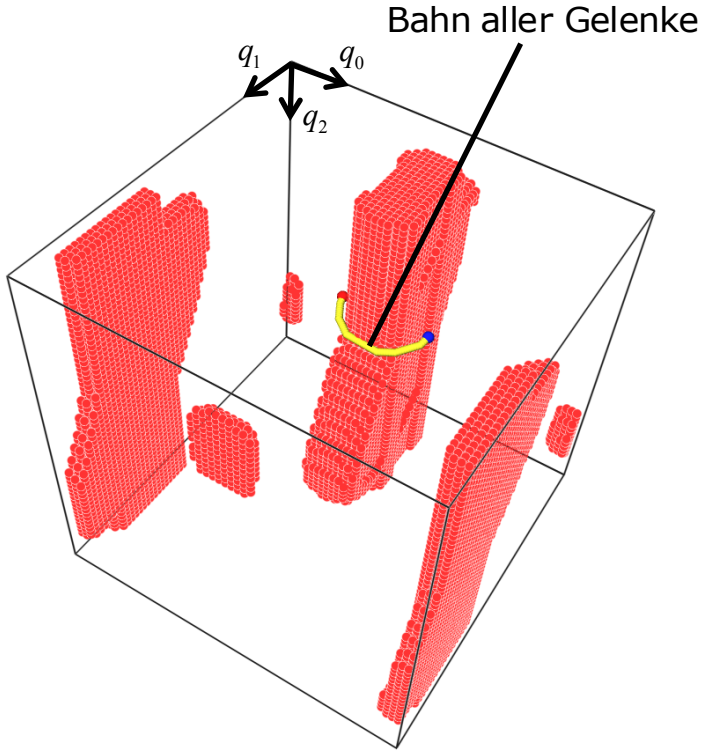
Kartesischer Raum <-> Gelenkwinkelraum

- Zustände können dargestellt werden im

Kartesischen Raum (3D/6D)



Gelenkwinkelraum (n-dim.)



Grundlagen der Bahnsteuerung

Konfigurationsraum

- Zustände können dargestellt werden im
  - Gelenkwinkelraum
  - Kartesischen Raum
- Bahnsteuerung im Gelenkwinkelraum ist näher an der Ansteuerung der Teilsysteme des Roboter (Gelenke, Sensorik)
- Bahnsteuerung im Kartesischen Raum ist näher an der zu lösenden Aufgabe

Bei Steuerung im Kartesischen Raum ist das Lösen der inverse Kinematik nötig

Grundlagen der Bahnsteuerung

Gelenkwinkelraum

- Bahnsteuerung als Funktion der Gelenkwinkelzustände
- Abfahren dieser punktweise spezifizierten Trajektorien durch
  - Steuerung der Achsen unabhängig voneinander (**asynchron**)
    Anwendung: Punktschweißen, Handhabungsaufgaben
  - achsinterpolierte Steuerung (Bewegung aller Achsen beginnt und endet zum gleichen Zeitpunkt, Leitachse) (**synchron**)
    Anwendung: Bahnschweißen, Lackieren, Montieren
- Verlauf der punktweise in Gelenkwinkel spezifizierten Bahn muss im kartesischen Raum nicht notwendigerweise definiert sein

Grundlagen der Bahnsteuerung

Kartesischer Raum (Continuous **P**ath)

- Angabe der Trajektorie erfolgt als Funktion der Zustände des Roboters
  - z.B. mit Beschreibungsvektor des TCP, v_{TCP}, a_{TCP}
- Endeffektor folgt in Lage und Orientierung einer **definierten** Bahn
- Bahntypen
  - lineare Bahnen
  - Polynombahnen
  - Splines

Grundlagen der Bahnsteuerung

Kartesischer Raum

+ Bahn einfacher zu formulieren
+ Interpolation ist einfacher

- Inverse Kinematik ist für jeden Trajektorienpunkt zu lösen
- Geplante Trajektorie nicht immer ausführbar

Gelenkwinkelraum

+ Ansteuerung der Gelenke ist einfacher
+ Trajektorie ist eindeutig und berücksichtigt Grenzen

- Interpolation für mehrere Gelenke
- Formulierung der Trajektorie umständlicher

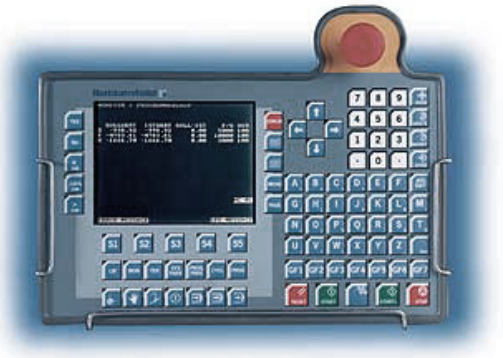
Inhalt

- Grundlagen der Bahnsteuerung
- **Programmierung der Schlüsselpunkte**
- Interpolationsarten
  - Punkt-zu-Punkt (PTP)
  - Linear- und Zirkularinterpolation
  - Splineinterpolation
- Approximierte Bahnsteuerung

Programmierung der Schlüsselpunkte

Direkte Programmierung: Teach-In

- Anfahren markanter Punkte der Bahn mit manueller Steuerung (Teach Box, Teach Panel, weitere: Spacemouse, Teach-Kugel)
- Funktionalität einer Teach Box:
  - Einzelbewegung der Gelenke
  - Bewegung des Effektors in 6 Freiheitsgraden
  - Speichern / Löschen von Anfahrpunkten
  - Eingabe von Geschwindigkeiten
  - Eingabe von Befehlen zur Bedienung des Greifers
  - Starten / Stoppen ganzer Programme



Programmierung der Schlüsselpunkte

Direkte Programmierung: Vorgehen beim Teach-In

- Anfahren markanter Schlüsselpunkte der Bahn
- Speichern der Gelenkwerte
- Ergänzung der gespeicherten Werte um Parameter wie Geschwindigkeit, Beschleunigung usw.
- Anwendung:
  - in der Fertigungsindustrie (Punktschweißen, Nieten)
  - Handhabungsaufgaben (Pakete vom Fließband nehmen)

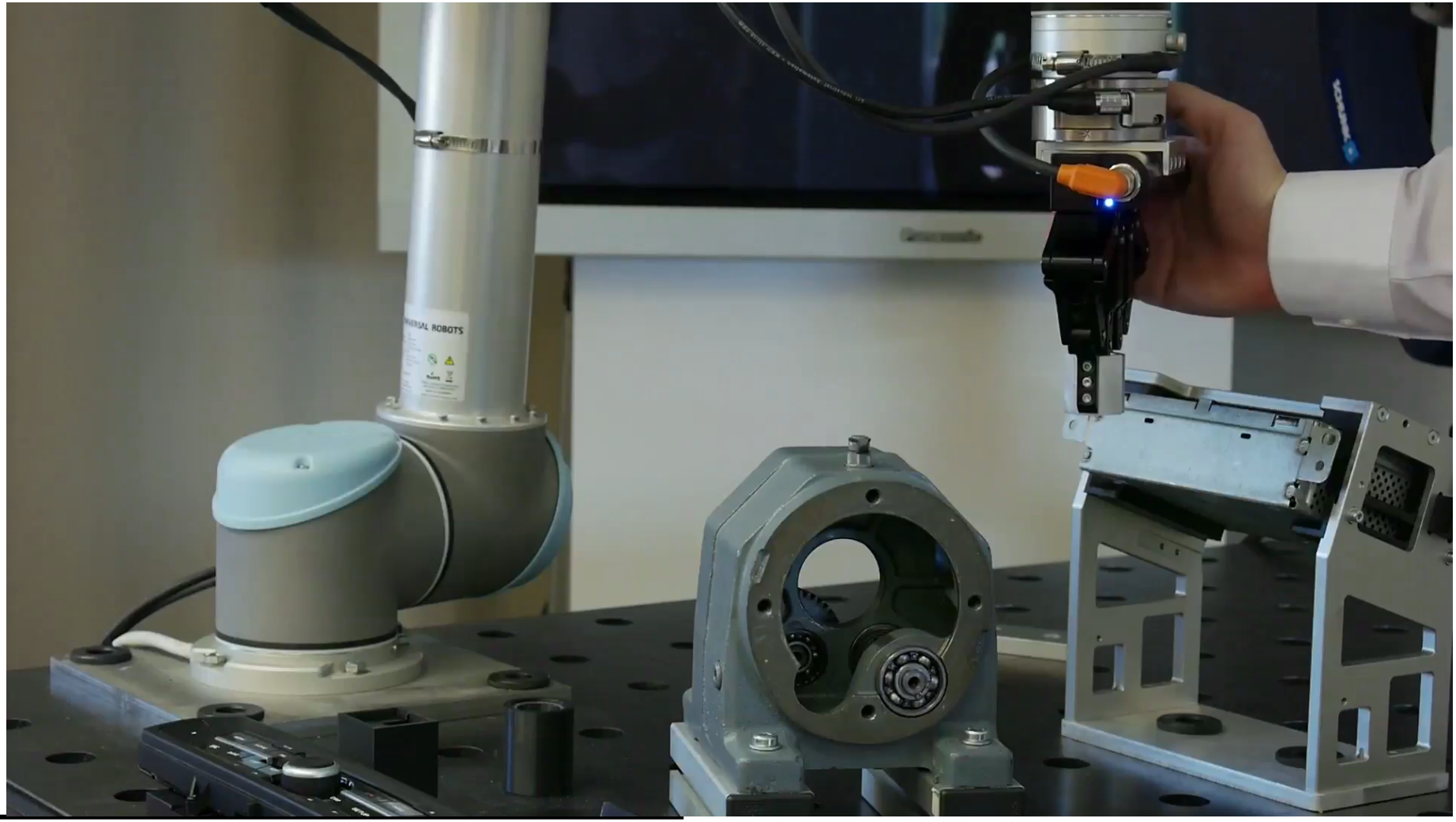
Programmierung der Schlüsselpunkte

Direkte Programmierung: Playback

- Einstellung des Roboters auf Zero-Force-Control (Roboter kann durch den Bediener bewegt werden)
- Abfahren der gewünschten Bahn
- Speichern der Gelenkwerte:
  - automatisch (definierte Abtastfrequenz)
  - oder manuell (durch Tastendruck)
- Anwendung:
  - mathematisch schwer beschreibbare Bewegungsabläufe
  - Integrierung der handwerklichen Erfahrung
  - Typische Einsatzbereiche sind: Lackieren oder Kleben

Programmierung der Schlüsselpunkte

Direkte Programmierung: Playback



Programmierung der Schlüsselpunkte

Direkte Programmierung: Vor/Nachteile Playback

✓ Schnell für komplexe Bahnen

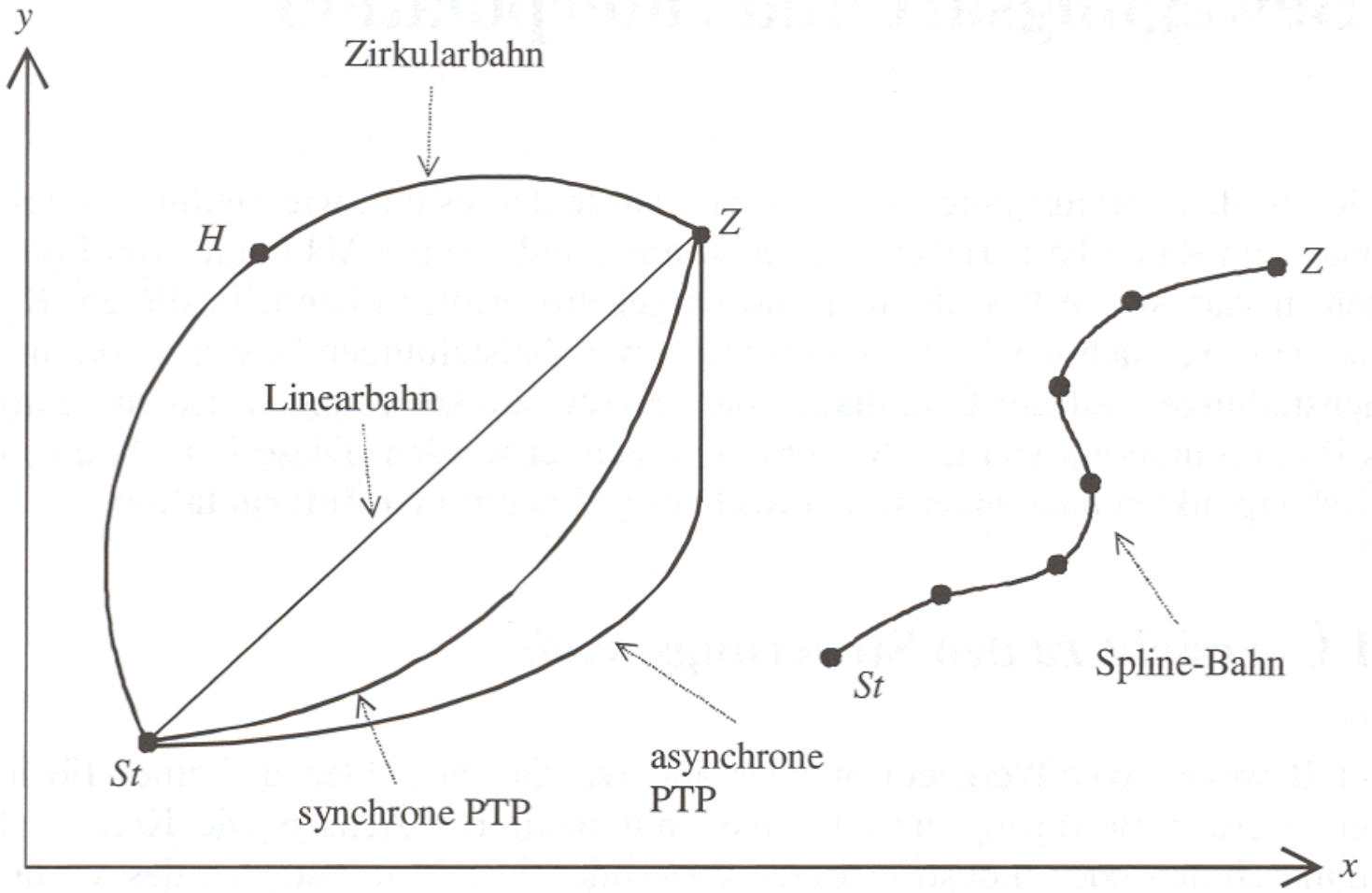
✓ Intuitiv

- schwere Roboter schwierig zu bewegen
- wenig Platz in engen Fertigungszellen für Bediener
  - dadurch Sicherheitsrisiko
- schlechte Korrekturmöglichkeiten
- Optimierung und Kontrolle durch Interpolationsmethoden schwierig:
  - Suboptimale Bahnen

Inhalt

- Grundlagen der Bahnsteuerung
- Programmierung der Schlüsselpunkte
- **Interpolationsarten**
  - Punkt-zu-Punkt (PTP)
  - Linear- und Zirkularinterpolation
  - Splineinterpolation
- Approximierte Bahnsteuerung

Interpolationsarten



Interpolationsarten

Punkt-zu-Punkt-Steuerung (PTP) (1)

- Roboter führt **Punkt-zu-Punkt-Bewegung** aus

- Vorteile:
  - Die Berechnung der Gelenkwinkeltrajektorie ist einfach
  - Keine Probleme mit Singularitäten

- Sequenz von Gelenkwinkelvektoren

$$\vec{q}(t_j) = (q_1(t_j), q_2(t_j), ..., q_n(t_j))^T$$

mit $q_i(t_j)$: Winkel des Gelenks i zum Zeitpunkt t_j mit j = 0,..., k

Interpolationsarten

Punkt-zu-Punkt-Steuerung (PTP) (2)

- Randbedingungen
  - Start- und Zielzustand sind bekannt

$$\vec{q}(t_{Start}) = \vec{q}_{Start}$$
$$\vec{q}(t_{Ziel}) = \vec{q}_{Ziel}$$

  - z.B. Geschwindigkeit zu Beginn und am Ende sind Null

$$\dot{\vec{q}}(t_{Start}) = 0$$
$$\dot{\vec{q}}(t_{Ziel}) = 0$$

  - Der Gelenkwinkelvorrat sowie Geschwindigkeiten und Beschleunigungen sind begrenzt (z.B. schnelles Beschleunigen, langsames Abbremsen)

$$\vec{q}_{min} < \vec{q}(t_j) < \vec{q}_{max}$$
$$\dot{\vec{q}}(t_j) < \dot{\vec{q}}_{max}$$
$$\ddot{\vec{q}}(t_j) < \ddot{\vec{q}}_{max}$$

Interpolationsarten

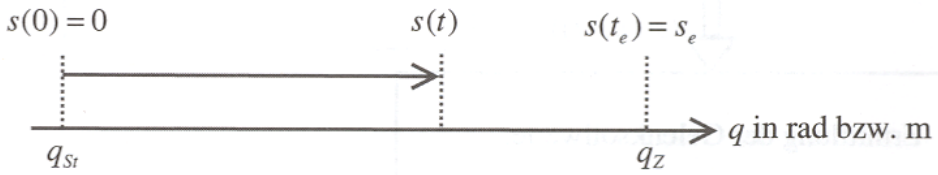
Punkt-zu-Punkt-Steuerung (PTP) (3)

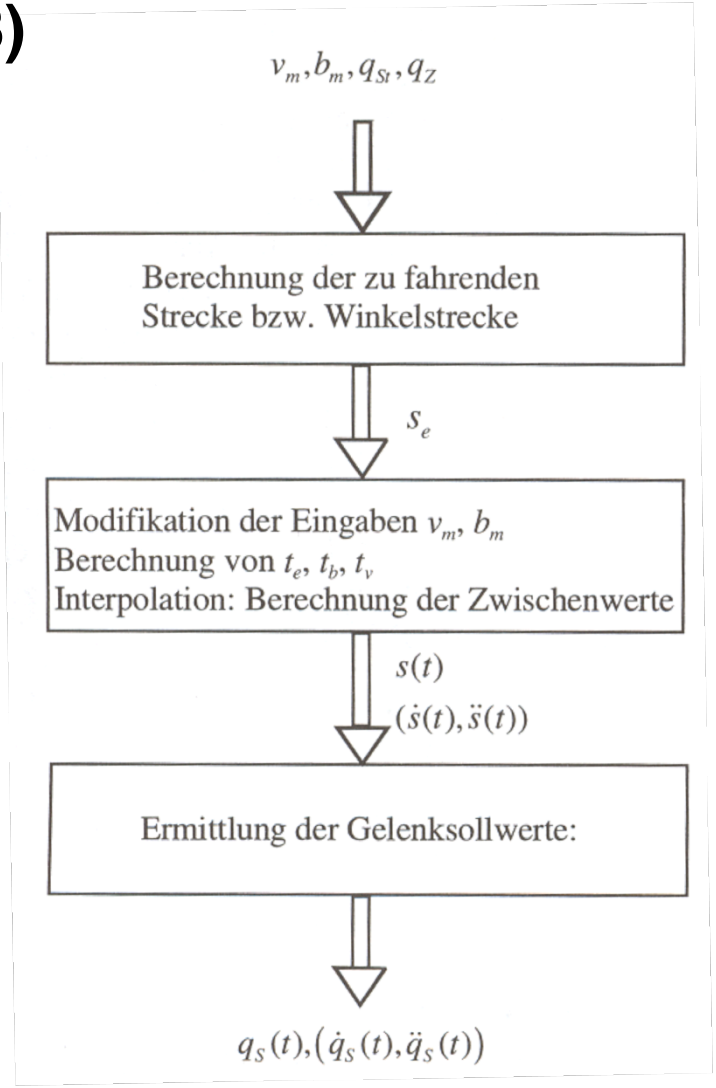
- Ablauf der Steuerung
  - Fahrzeit t_e
  - Beschleunigungszeit t_b
  - Beginn der Bremszeit t_v

$$s(t_e) = s_e = |q_z - q_{st}|$$

$$s(0) = \dot{s}(0) = v(0) = 0$$

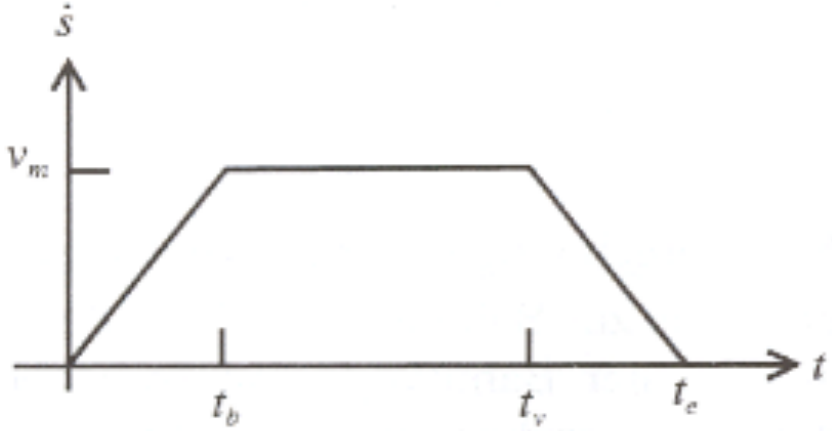
$$\dot{s}(t_e) = v(t_e) = 0$$





Interpolation für PTP mit Rampenprofil (1)

- Einfache Art zur Berechnung der Bahnparameter s(t)
- Sprungförmige Aufschaltung der Beschleunigung (ruckartig)
- Kann zu Eigenschwingungen von mechanischen Teilen führen



Interpolationsarten

Interpolation für PTP mit Rampenprofil (2)

- Phase der Beschleunigung

$$t_b = \frac{v_m}{b_m}$$

$$0 \le t \le t_b: \quad \ddot{s}(t) = b_m$$

$$\dot{s}(t) = b_m \cdot t$$

$$s(t) = \frac{1}{2} \cdot b_m \cdot t^2$$

- Phase der Gleichmäßigen Fahrt

$$t_b \le t \le t_v: \quad \ddot{s}(t) = 0$$

$$\dot{s}(t) = v_m$$

$$s(t) = v_m \cdot t - \frac{1}{2} \cdot b_m \cdot t_b^2 = v_m \cdot t - \frac{1}{2} \cdot \frac{v_m^2}{b_m}$$

Interpolationsarten

Interpolation für PTP mit Rampenprofil (3)

- Phase des Bremsvorganges

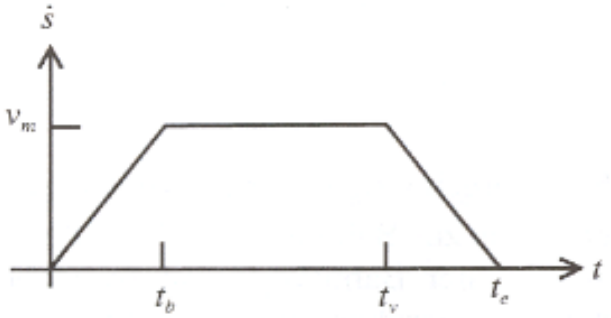
$$t_v \le t \le t_e: \quad \ddot{s}(t) = -b_m$$

$$\dot{s}(t) = v_m - b_m \cdot (t - t_v)$$

$$s(t) = v_m \cdot (t_e - t_b) - \frac{b_m}{2} \cdot (t_e - t)^2$$

- Berechnung der Fahrtzeit

$$t_e = \frac{s_e}{v_m} + t_b = \frac{s_e}{v_m} + \frac{v_m}{b_m}$$

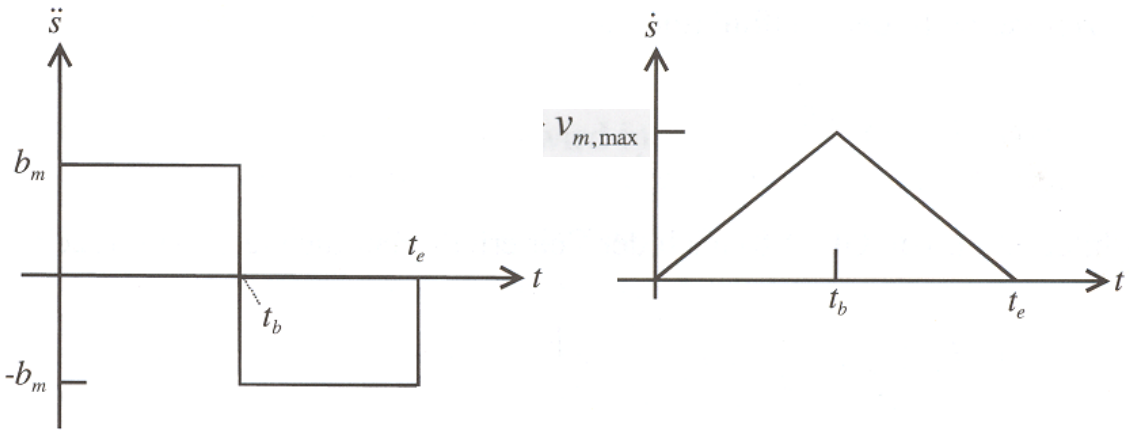


Interpolationsarten

Zeitoptimale Bahn

- Falls v_m zu groß in Bezug auf Beschleunigung und Bahnlänge
  - Bestimmung einer zeitoptimalen Bahn nach

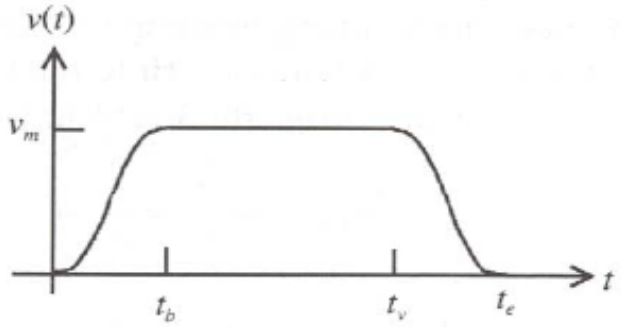
$$s_e = t_b \cdot v_{m,\max} = \frac{v_{m,\max}^2}{b_m} \Rightarrow v_{m,\max} = \sqrt{b_m \cdot s_e}$$



Interpolationsarten

Interpolation für PTP mit Sinoidenprofil (1)

- Weichere Bewegung durch Verwendung einer sinusförmigen Zeitfunktion
- Längere Beschleunigungs- und Bremsphase als beim Rampenprofil
- Roboter wird weniger beansprucht
- Bestimmung der Kurvenparameter für die Phase
  - Beschleunigung
  - Gleichförmige Bewegung
  - Bremsvorgang



Interpolationsarten

Sinoidenprofil zur Interpolation (2)

- Phase der Beschleunigung

$$0 \le t \le t_b: \quad \ddot{s}(t) = b_m \cdot \sin^2\left(\frac{\pi}{t_b} \cdot t\right)$$

$$\dot{s}(t) = b_m \cdot \left(\frac{1}{2} \cdot t - \frac{t_b}{4\pi} \cdot \sin\left(\frac{2\pi}{t_b} \cdot t\right)\right)$$

$$s(t) = b_m \cdot \left(\frac{1}{4} \cdot t^2 + \frac{t_b^{\,2}}{8\pi^2} \cdot \left(\cos\left(\frac{2\pi}{t_b} \cdot t\right) - 1\right)\right)$$

- Aus $\dot{s}(t_b) = b_m \cdot \frac{1}{2} \cdot t_b \stackrel{!}{=} v_m$ folgt $t_b = \frac{2v_m}{b_m}$
- Phase der Gleichmäßigen Fahrt

$$t_b \le t \le t_v: \quad \ddot{s}(t) = 0$$

$$\dot{s}(t) = v_m$$

$$s(t) = v_m \cdot (t - \frac{1}{2} \cdot t_b)$$

Interpolationsarten

Sinoidenprofil zur Interpolation (3)

- Phase des Bremsvorganges

$$t_v \le t \le t_e: \quad \dot{s}(t) = v_m - \int_{t-t_v}^{t} b(\tau - t_v)\cdot d\tau = v_m - b_m \cdot \left(\frac{1}{2}\cdot(t-t_v) - \frac{t_b}{4\pi}\cdot \sin\left(\frac{2\pi}{t_b}\cdot(t-t_v)\right)\right)$$

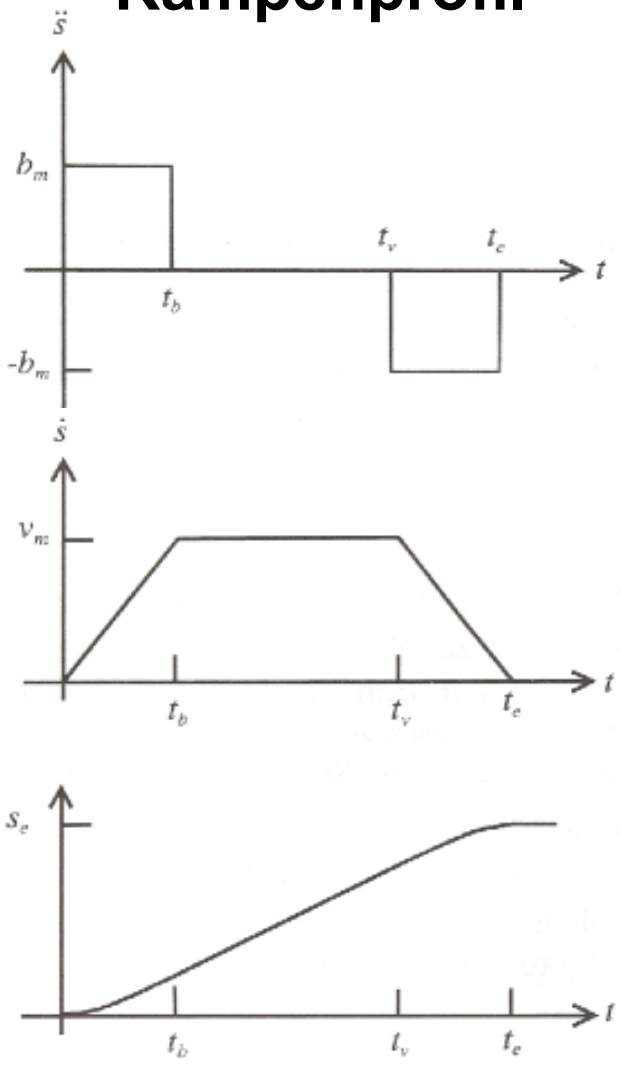
$$s(t) = s(t_v) + \int_{t-t_v}^{t} \dot{s}(\tau - t_v)\cdot d\tau = \frac{b_m}{2}\cdot\left(t_e(t+t_b) - \frac{(t^2 + t_e^{\,2} + 2\cdot t_b^{\,2})}{2} + \frac{t_b^{\,2}}{4\pi^2}\cdot\left(1-\cos\left(\frac{2\pi}{t_b}\cdot(t-t_v)\right)\right)\right)$$

- Berechnung der Fahrtzeit

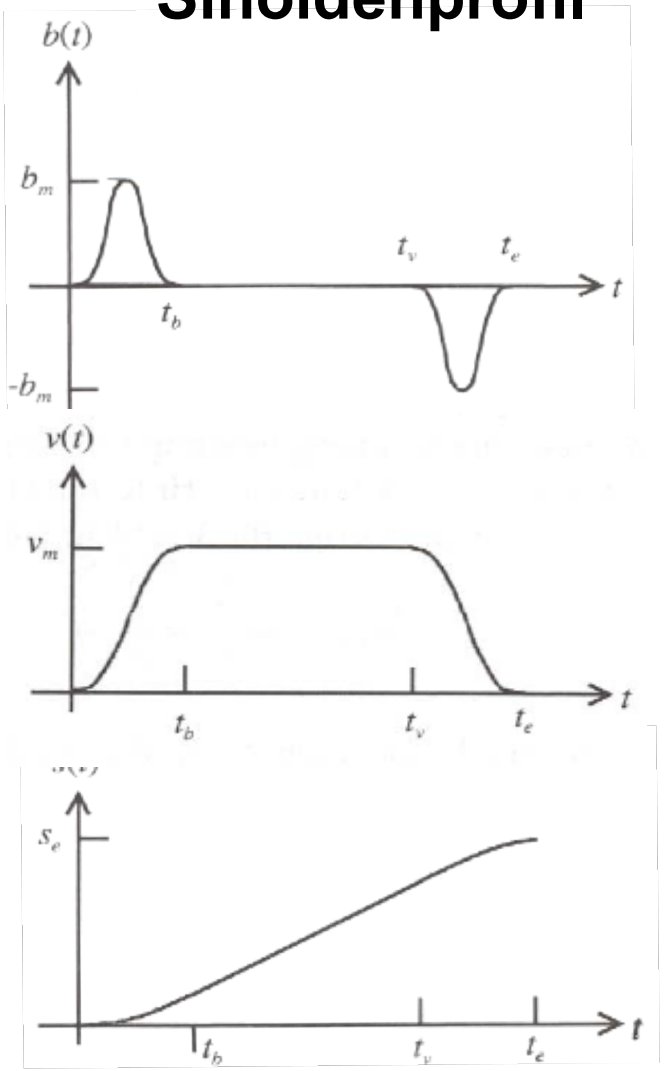
$$t_e = \frac{s_e}{v_m} + t_b = \frac{s_e}{v_m} + \frac{2\cdot v_m}{b_m}$$

Interpolationsarten

Rampenprofil



Sinoidenprofil



Interpolationsarten

Synchrone PTP (1)

- Vorgehen bei **synchronen** PTP-Bahnen
  - Bestimme für jedes Gelenk i PTP-Parameter (analog zur asynchronen PTP)
    - $s_{e,i}$
    - $v_{m,i}$
    - $b_{m,i}$
    - $t_{e,i}$ (Fahrzeit)
  - Bestimme $t_e = t_{e,max} = max(t_{e,i})$
    - Achse mit max. Fahrzeit ist Leitachse
  - Setze $t_{e,i} = t_e$ für **alle** Gelenke

Interpolationsarten

Synchrone PTP (2)

- Bestimme die neuen maximalen Geschwindigkeiten für alle Gelenke
- Umformung Fahrzeit und Berechnung der neuen Geschwindigkeiten
  - Rampenprofil:

$$t_e = \frac{s_{e,i}}{v_{m,i}} + \frac{v_{m,i}}{b_{m,i}} \Rightarrow v_{m,i}^2 = v_{m,i} \cdot b_{m,i} \cdot t_e + s_{e,i} \cdot b_{m,i}$$

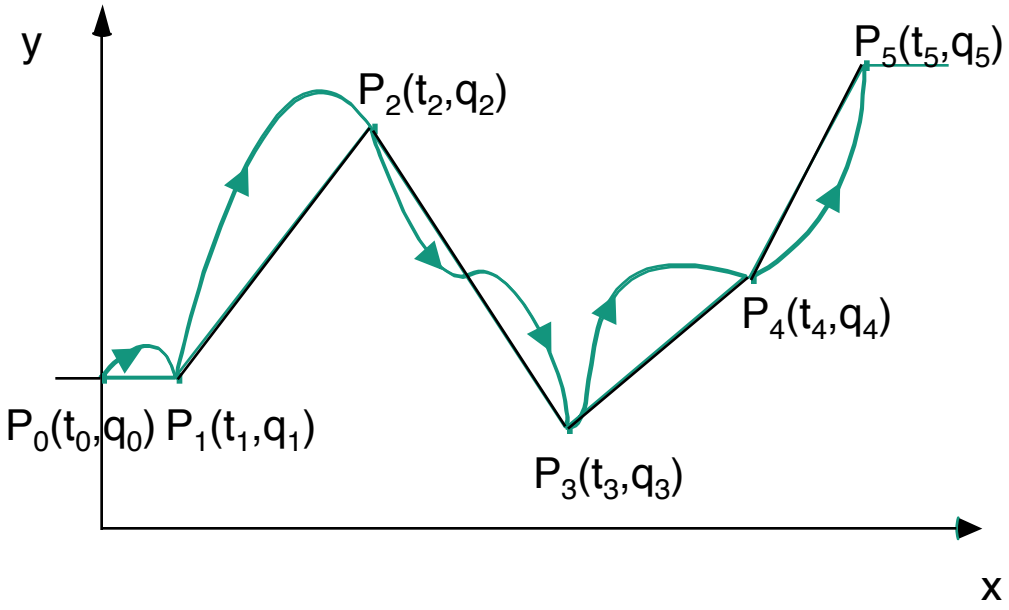
$$v_{m,i} = \frac{b_{m,i} \cdot t_e}{2} - \sqrt{\frac{b_{m,i}^2 \cdot t_e^2}{4} - s_{e,i} \cdot b_{m,i}}$$

  - analoge Berechnung für Sinoidenbahn:

$$v_{m,i} = \frac{b_{m,i} \cdot t_e}{4} - \sqrt{\frac{b_{m,i}^2 \cdot t_e^2 - 8 \cdot s_{e,i} \cdot b_{m,i}}{16}}$$

Interpolationsarten

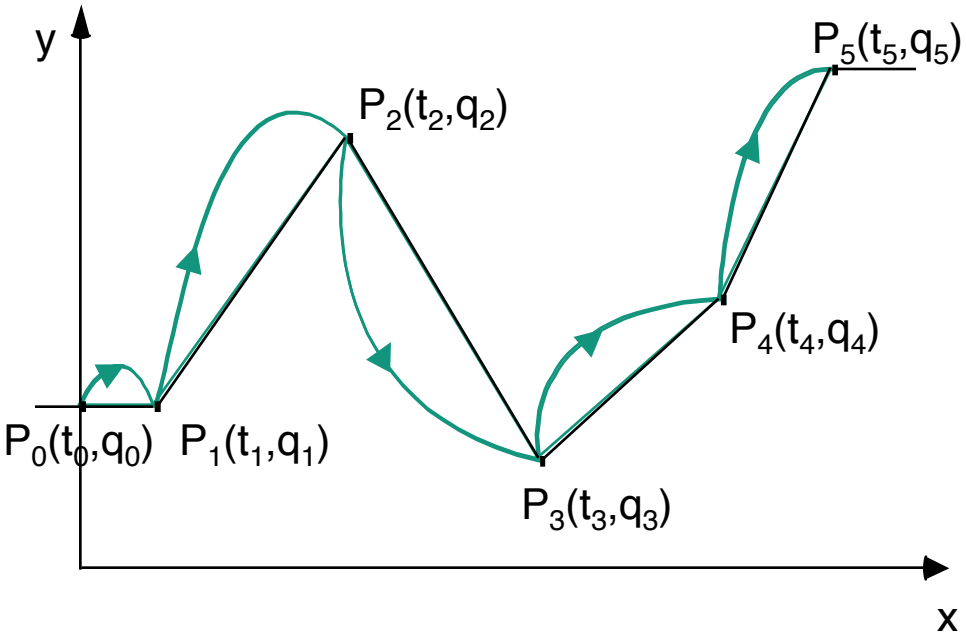
PTP asynchron



Jedes Gelenk wird sofort mit der maximalen Beschleunigung angesteuert.
Jede Gelenkbewegung endet unabhängig von den anderen.

Interpolationsarten

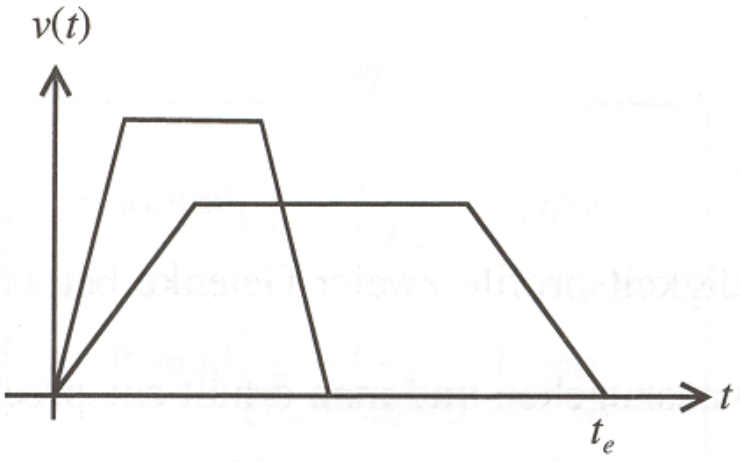
PTP synchron



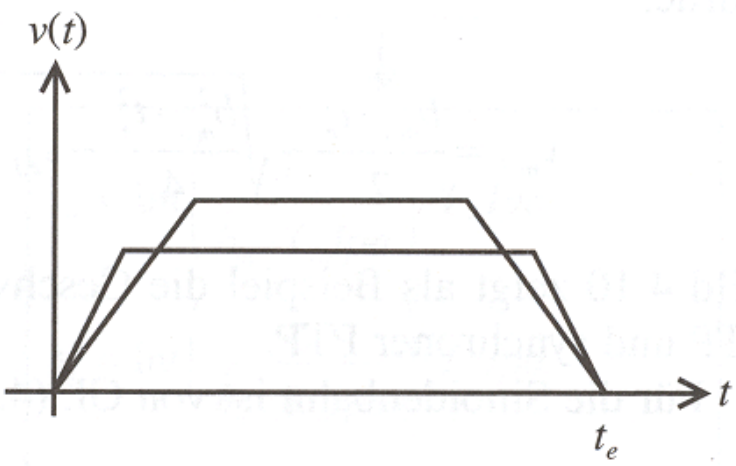
Alle Gelenke beginnen und beenden ihre Bewegungen gemeinsam (synchron).

Interpolationsarten

Vergleich: asynchrone und synchrone PTP



a) asynchrone PTP



b) synchrone PTP

Interpolationsarten

Vollsynchrone PTP

- zusätzliche Berücksichtigung der Beschleunigungs- und Bremszeit
- bessere Annäherung der Start- und Zielpunkte im kartesischen Raum
- Bestimmung Leitachse mit t_e und t_b => $t_v = t_e - t_b$
- Bestimmung der Geschwindigkeit und Beschleunigung der anderen Achse mit

$$v_{m,i} = \frac{s_{e,i}}{t_v}$$

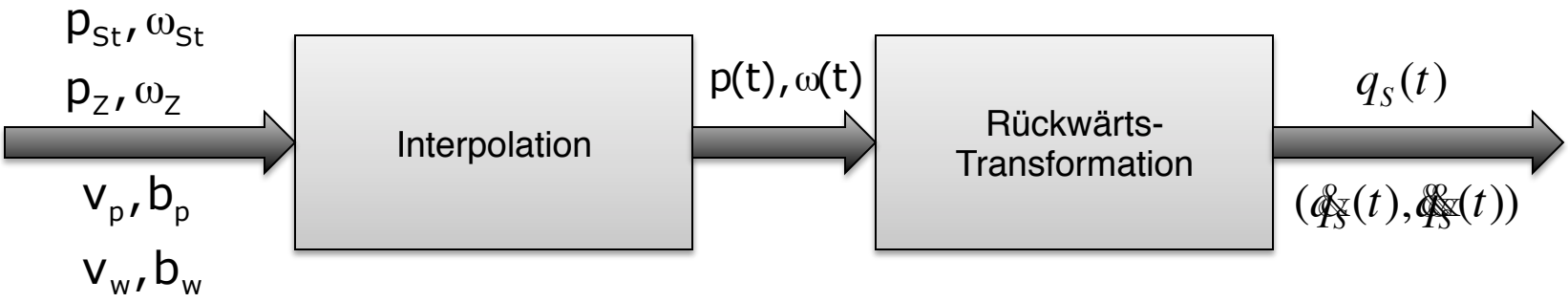
$$b_{m,i} = \frac{v_{m,i}}{t_b}$$

Nachteil: Beschleunigung jeder Achse wird vorgegeben

Interpolationsarten

Steuerung im Kartesischem Raum

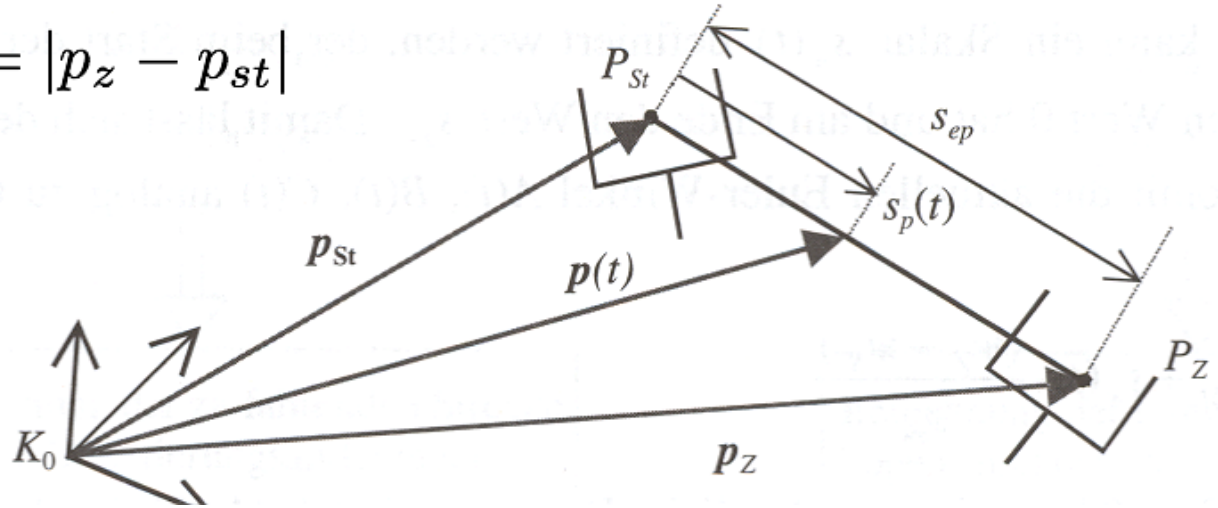
- Continuous Path (CP)
- Endeffektor folgt in Lage und Orientierung einer **definierten** Bahn



Interpolationsarten

Linearinterpolation

$$s_{ep} = |p_z - p_{st}|$$



$$p(t) = p_{St} + s_p(t) \cdot \frac{(p_Z - p_{St})}{s_{ep}}$$

Berechnung von $s_p(t)$ mit Rampen- oder Sinoidenprofil

$$s_p(0) = \dot{s}_p(0) = v_p(0) = 0,\ \dot{s}_p(t_e) = v_p(t_e) = 0$$

$$v_m = v_p,\ b_m = b_p,\ t_e = t_{ep},\ t_b = t_{bp},\ t_v = t_{vp},\ s_e = s_{ep},\ s = s_p$$

Interpolationsarten

Linearinterpolation (2)

- Orientierung in Eulerwinkel $\omega = (\alpha, \beta, \gamma)^T$

$$s_{e\omega} = |\omega_Z - \omega_{St}| = \sqrt{(\alpha_Z - \alpha_{St})^2 + (\beta_Z - \beta_{St})^2 + (\gamma_Z - \gamma_{St})^2}$$

- Berechnung von $s_w(t)$ mit Rampen- oder Sinoidenprofil:

$$v_m = v_\omega, b_m = b_\omega, t_e = t_{e\omega}, t_b = t_{b\omega}, t_v = t_{v\omega}, s_e = s_{e\omega}, v_m = v_\omega, s = s_\omega$$

- Angleich der Fahrzeiten t_{ep} (Position) und t_{ew} (Orientierung)

$$t_e = \max(t_{ep}, t_{e\omega})$$

  - Analog zur Anpassung der Geschwindigkeiten bei synchronen PTP
    - Falls $t_e = t_{ep}$:

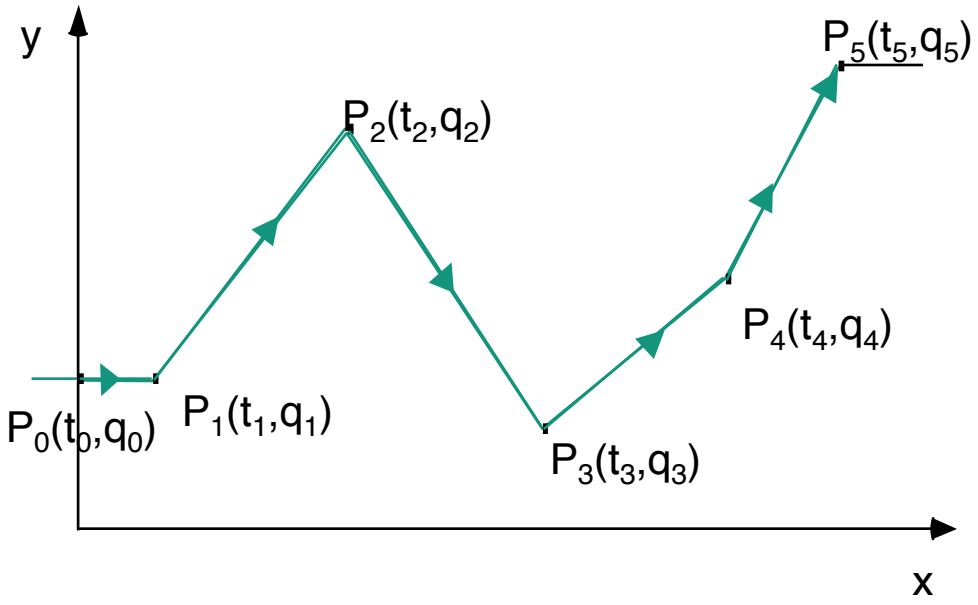
$$v_\omega = \frac{b_\omega \cdot t_e}{2} - \sqrt{\frac{b_\omega^2 \cdot t_e^2}{4} - s_{e\omega} \cdot b_\omega}$$

    - Falls $t_e = t_{ew}$:

$$v_p = \frac{b_p \cdot t_e}{2} - \sqrt{\frac{b_p^2 \cdot t_e^2}{4} - s_{ep} \cdot b_p}$$

Interpolationsarten

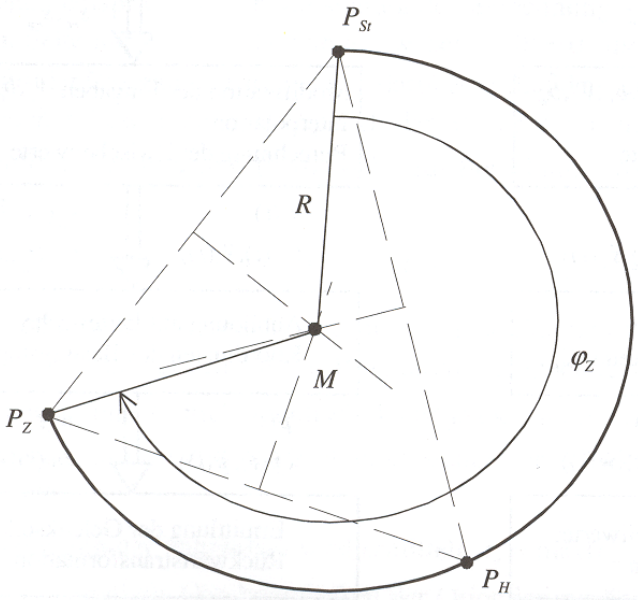
CP linear

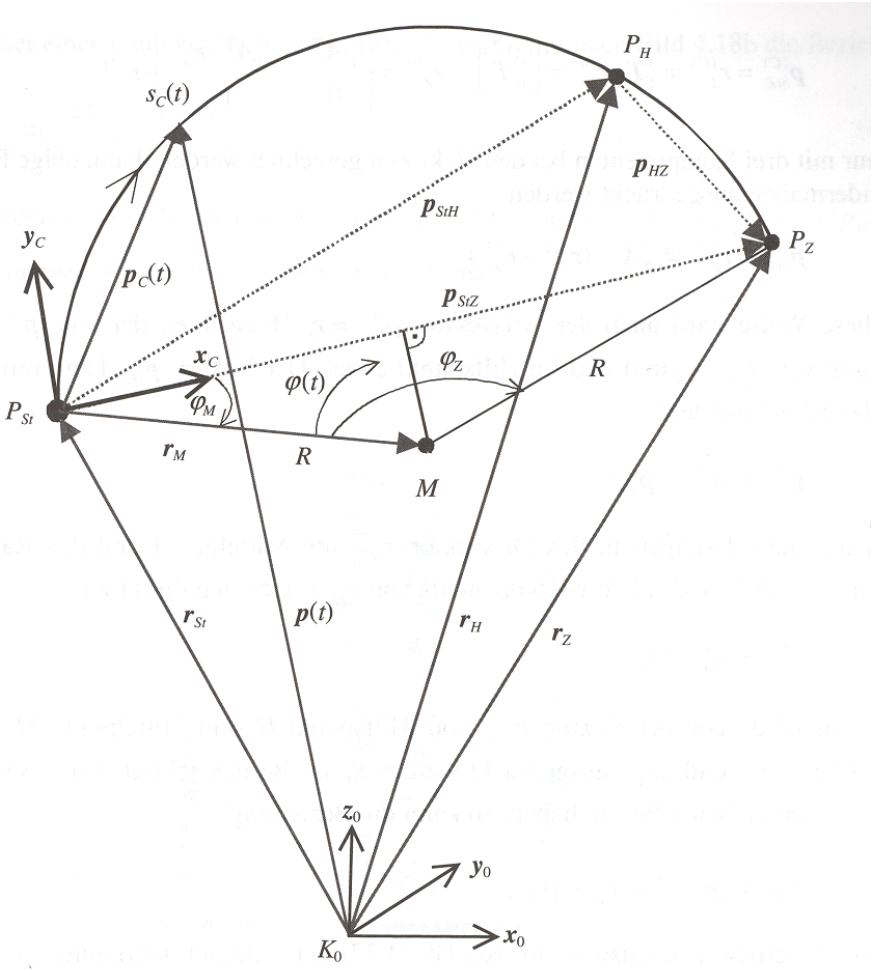


Die Robotersteuerung interpoliert die Bahn zwischen je 2 Teiltrajektorien.

Interpolationsarten

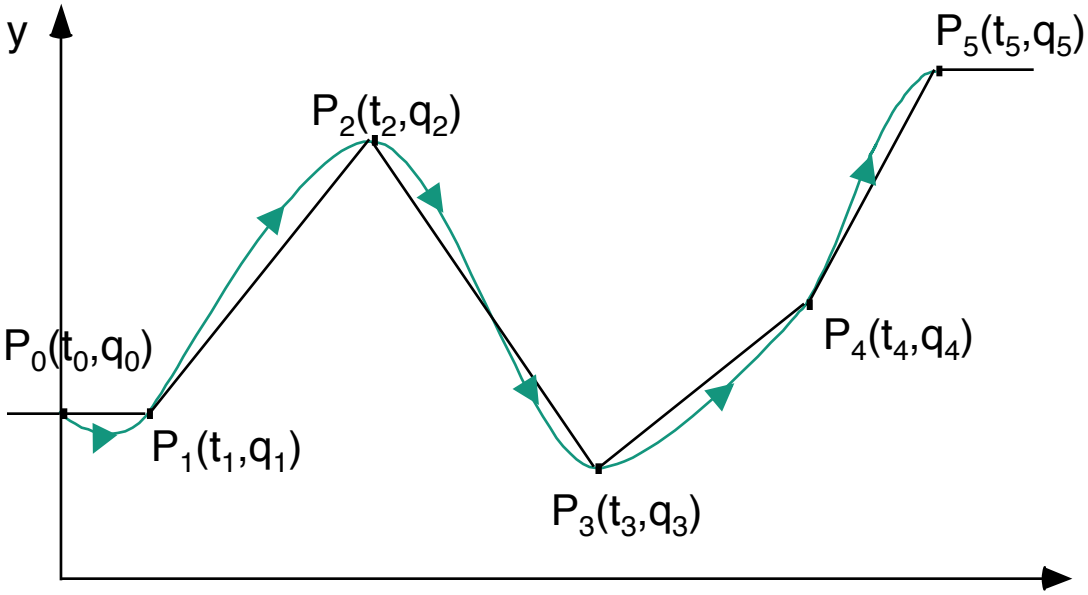
Zirkularinterpolation





Interpolationsarten

Segmentweise Bahninterpolation



- Die Endbedingungen der Teiltrajektorie j-1 (Richtung, Geschwindigkeit, Beschleunigung) und die Anfangsbedingungen der Teiltrajektorie j werden aneinander angeglichen
- Teiltrajektorien werden separat beschrieben (Bsp: Splines)

Interpolationsarten

Beispiel: Kubische Splines (1)

Geg.: * Polynom: $f(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \qquad (a_0, a_1, a_2, a_3 \in R)$ (1)

* Start und Ziel: S_{Start}, S_{Ziel} (2)

Ansatz: $\dot{f}(t) = a_1 + 2a_2 t + 3a_3 t^2$ (3)

$\ddot{f}(t) = 2a_2 + 6a_3 t$ (4)

Aus (1) und (2) folgt

$$f(t_{Start}) = f(0) = a_0 = S_{Start} \quad (5)$$

$$f(t_{Ziel}) = a_0 + a_1 t_{Ziel} + a_2 t_{Ziel}^2 + a_3 t_{Ziel}^3 = S_{Ziel} \quad (6)$$

Interpolationsarten

Beispiel: Kubische Splines (2)

Aus Randbedingungen $(\dot{f}(t_{Start}) = \dot{f}(t_{Ziel}) = 0)$

und (3) folgt:

$$\dot{f}(t_{Start}) = \dot{f}(0) = a_1 = v_{Start} \qquad (7)$$

$$\dot{f}(t_{Ziel}) = a_1 + 2a_2 t_{Ziel} + 3a_3 t_{Ziel}^2 = v_{Ziel} \qquad (8)$$

Aus (7),(8) folgt:

$$a_2 = \frac{v_{Ziel} - v_{Start}}{2t_{Ziel}} - \frac{3}{2} a_3 t_{Ziel} \qquad (9)$$

Aus (5), (6), (7), (9) folgt:

$$a_3 = \frac{2(S_{Start} - S_{Ziel})}{t_{Ziel}^3} + \frac{(v_{Start} - v_{Ziel})}{t_{Ziel}^2} \qquad (10)$$

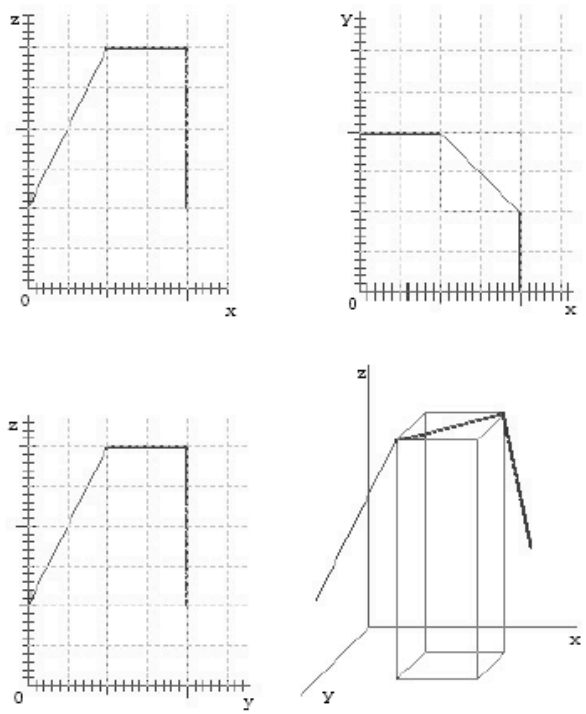
Aus (9), (10) folgt:

$$a_2 = \frac{3(S_{Ziel} - S_{Start})}{t_{Ziel}^2} - \frac{(v_{Ziel} + 2v_{Start})}{t_{Ziel}}$$

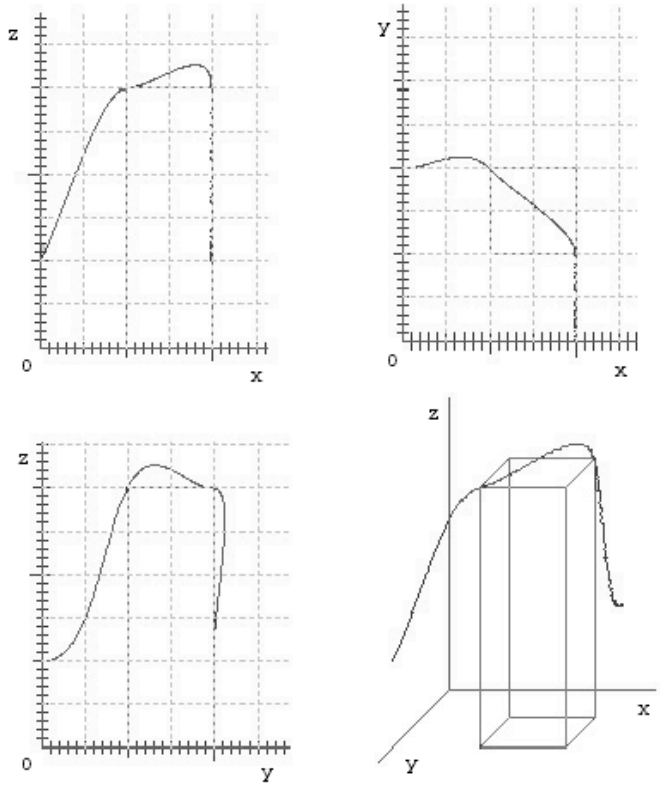
Interpolationsarten

Beispiel: Splines

- Bahn (4 Stützpunkte)



- Splineinterpolation



Inhalt

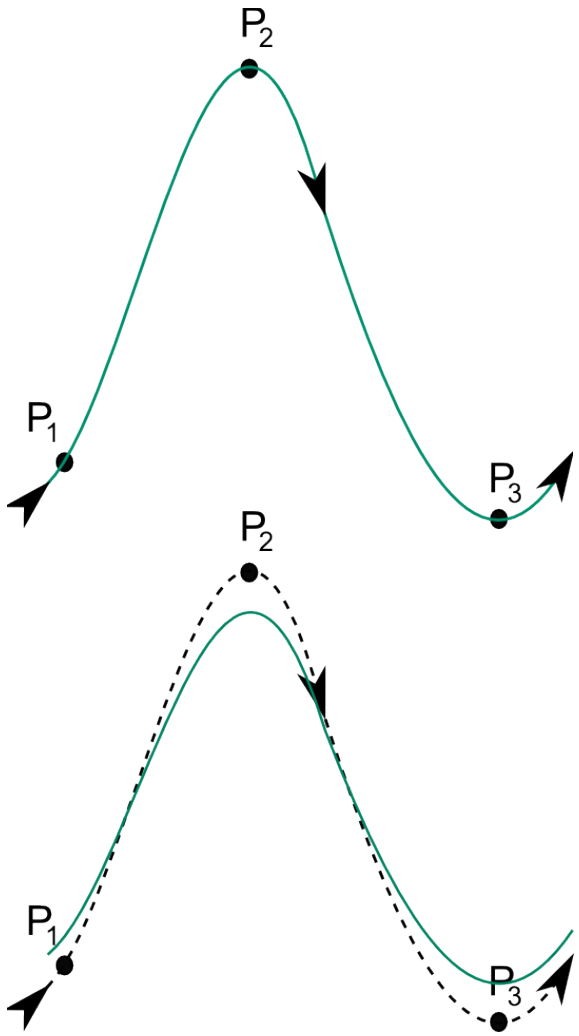
- Grundlagen der Bahnsteuerung
- Programmierung der Schlüsselpunkte
- Interpolationsarten
- **Approximierte Bahnsteuerung**
  - Bernsteinpolynome

Approximierte Bahnsteuerung

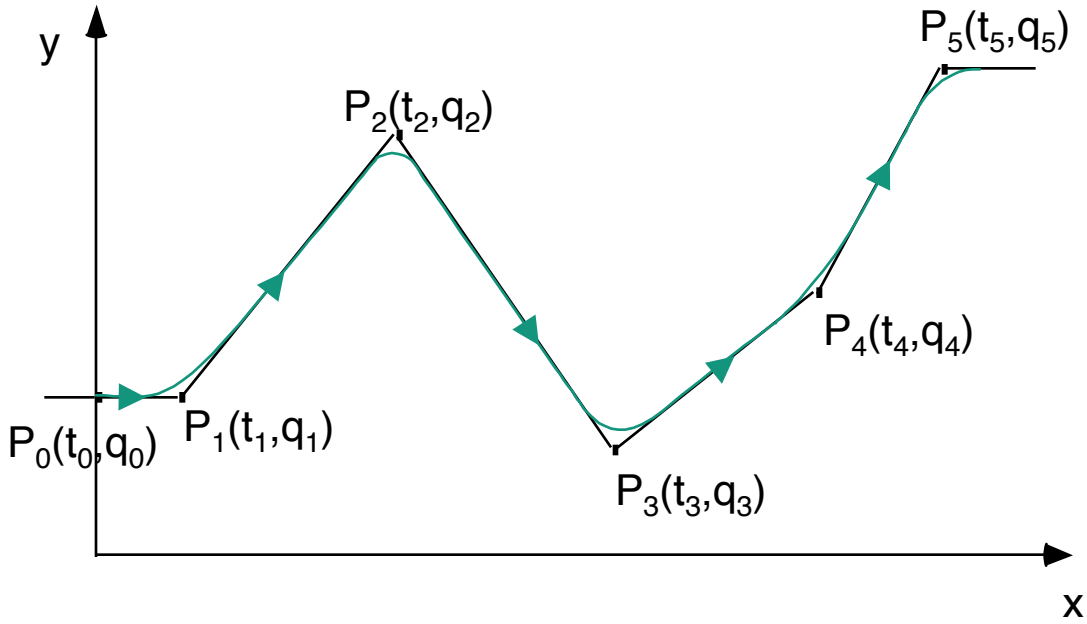
Definition

- **Bahninterpolation**
  - Die ausgeführte Bahn verläuft durch alle Stützpunkte der Trajektorie

- **Bahnapproximation**
  - Die Kontrollpunkte beeinflussen den Bahnverlauf und werden approximiert

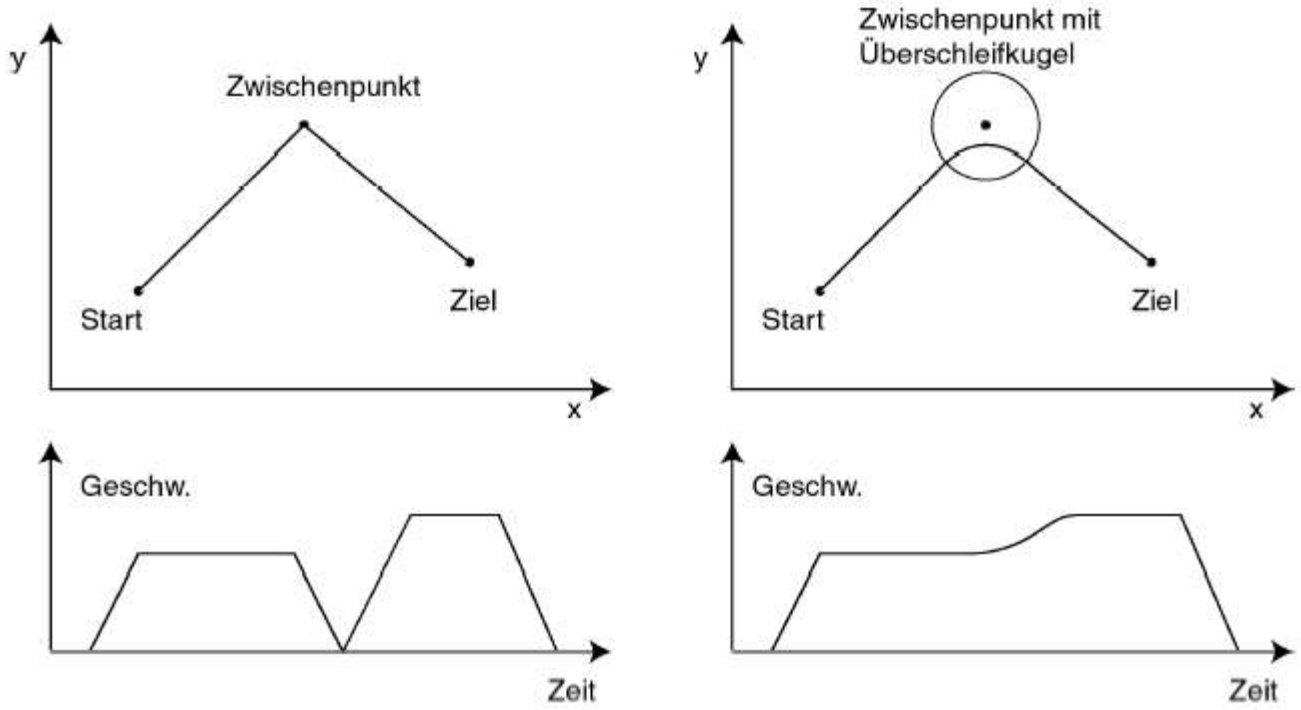


PTP und CP mit Überschleifen



Zum Zeitpunkt t_j -ε wird begonnen die Parameter (Richtung und Geschwindigkeit) der Teiltrajektorie j-1 auf die Parameter der Teiltrajektorie j zu überführen. I.d.R. wird der Stützpunkt i nicht erreicht.

PTP und CP mit Überschleifen (2)



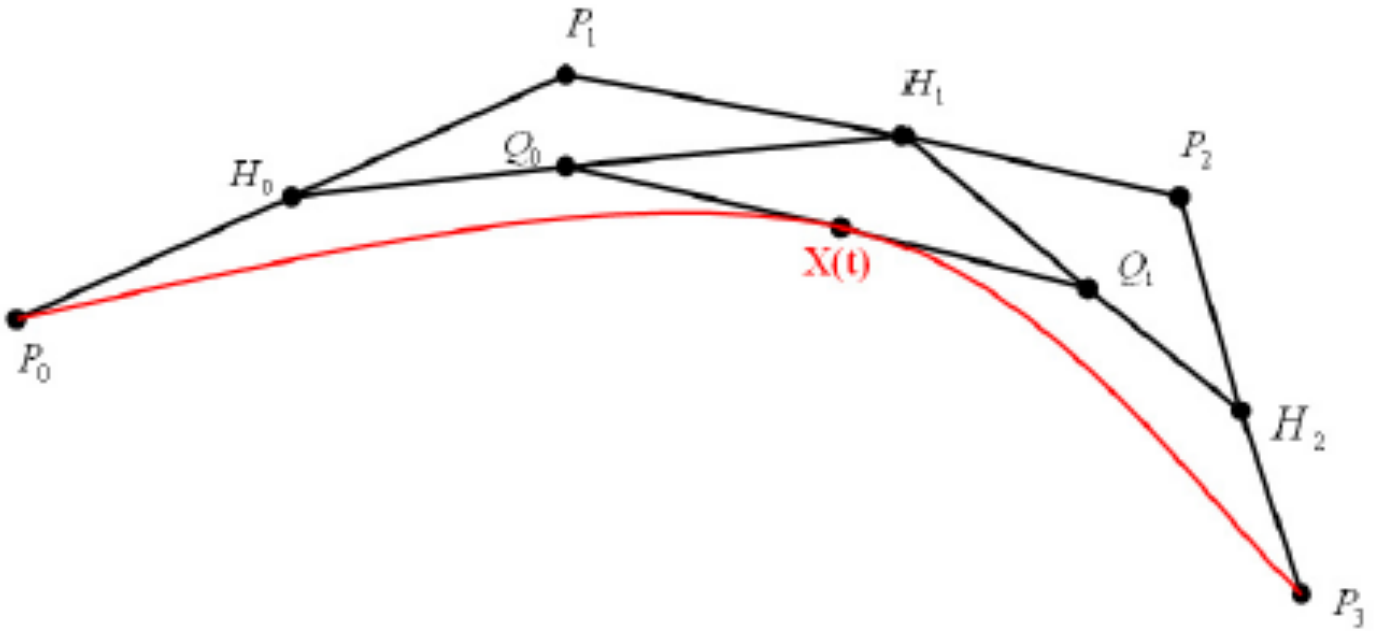
PTP und CP mit Überschleifen (3)

- Geschwindigkeitsüberschleifen
  - Beginn, wenn die Geschwindigkeit einen festgelegten Minimalwert unterschreitet.
  - Nachteil: Abhängig vom Geschwindigkeitsprofil.

- Positionsüberschleifen
  - Beginn, wenn der TCP in die Überschleifkugel eintritt
  - Außerhalb der Überschleifkugel wird die Bahn exakt eingehalten.
  - Vorteil: Gut kontrollierbar

Approximation mit Polynomen

- Beispiel: Bernsteinpolynome



Approximierte Bahnsteuerung

Bernsteinpolynom

- Im Unterschied zu kubischen Splines verlaufen **Bézierkurven** nicht durch alle Stützpunkte, sondern werden nur von ihnen beeinflusst.
- Basisfunktion:

$$P(t) = \sum_{i=0}^{n} B_{i,n}(t) P_i \qquad 0 \leq t \leq 1$$

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

Approximierte Bahnsteuerung

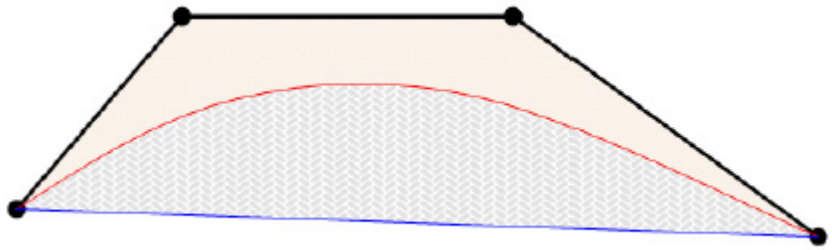
Berechnung beliebiger Zwischenstellungen

- Bernsteinpolynom für kubischen Fall

$$B_{i,3}(t) = \binom{3}{i} t^i (1-t)^{3-i}$$

$$p(t) = p_0(1-t)^3 + 3p_1(1-t)^2 t + 3p_2(1-t)\ t^2 + p_3 t^3$$

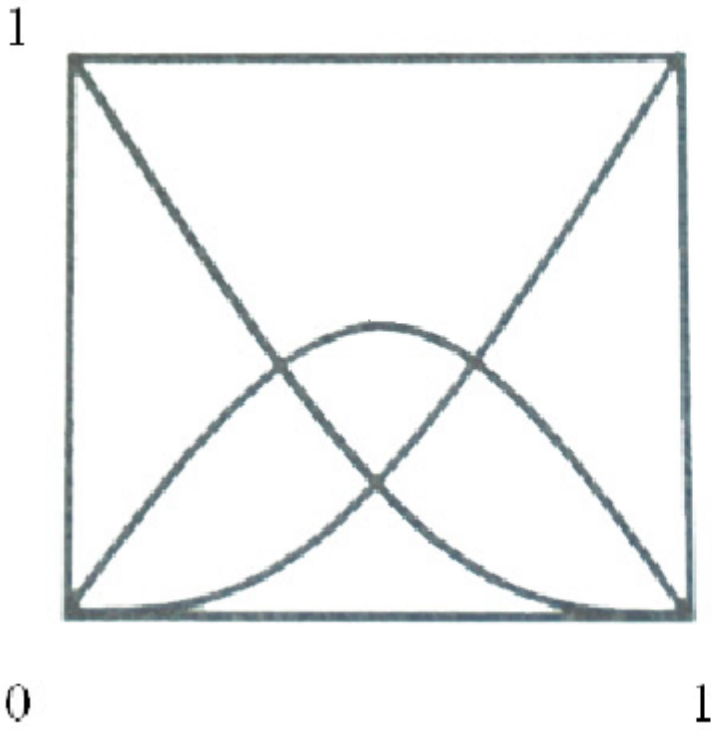
  - Annähern von unten an Stützstellen
  - keine beliebige Form



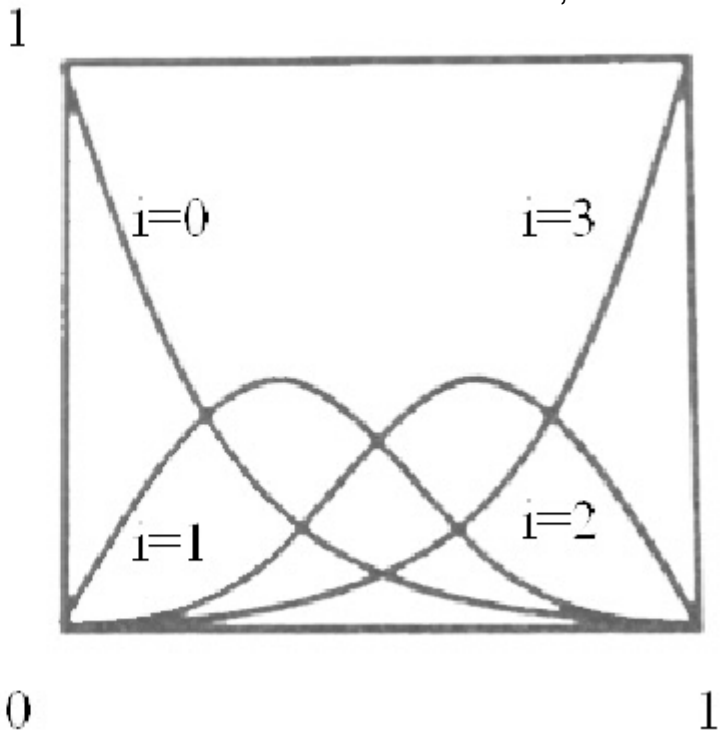
Approximierte Bahnsteuerung

Beispiele für Bernsteinpolynome

Quadratische Polynome $B_{i,2}$



Kubische Polynome $B_{i,3}$



Approximierte Bahnsteuerung

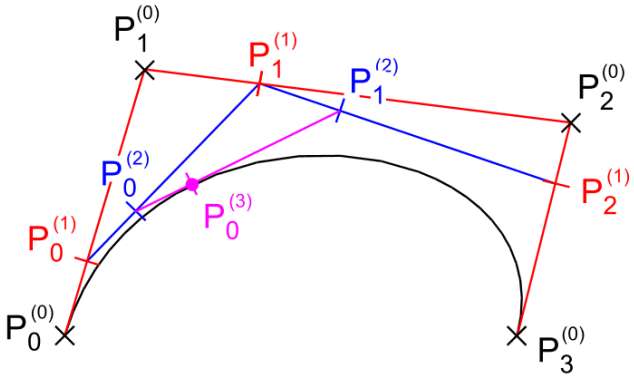
Der De-Casteljau-Algorithmus (1)

- Annäherung an die Bézierkurve
- Iterative Berechnung:
  Kann auch für große *n* effizient berechnet werden
- Gegeben: n Kontrollpunkte P_0, …, P_{n-1}
- Start: $P_i^{(0)} = P_i$
- Iteration k: $P_i^{(k+1)} = (1 - t_0) \cdot P_i^{(k)} + t_0 \cdot P_{i+1}^{(k)}$

Approximierte Bahnsteuerung

Der De-Casteljau-Algorithmus (2)

- Beispiel für P_0 mit k=3 und t_0=0.25:



- Zwei Bézierkurven $C_1(t)$ und $C_2(t)$
- Approximation der Bézierkurve durch Polygonzug