

Robotik I: Einführung in die Robotik

Bahnsteuerung

Tamim Asfour, Rüdiger Dillmann

KIT-Fakultät für Informatik, Institut für Anthropomatik und Robotik (IAR)
Hochperformante Humanoide Technologien (H²T)



Inhalt

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

Grundlagen der Bahnsteuerung: 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

Beispiel für ein Gelenk

■ Anfangsbedingungen

$$q(t_0) = 15^\circ$$

$$\dot{q}(t_0) = 0 \frac{\text{Grad}}{\text{sec}}$$

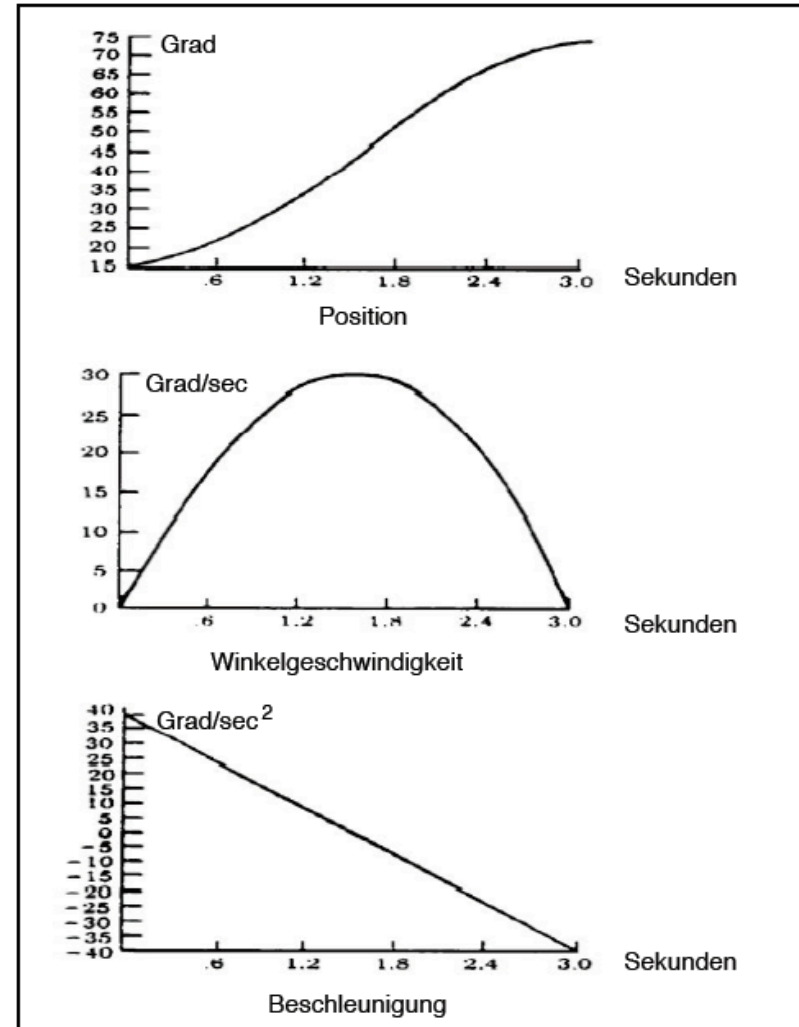
$$\ddot{q}(t_0) = 40 \frac{\text{Grad}}{\text{sec}^2}$$

■ Endbedingungen

$$q(t_e) = 75^\circ$$

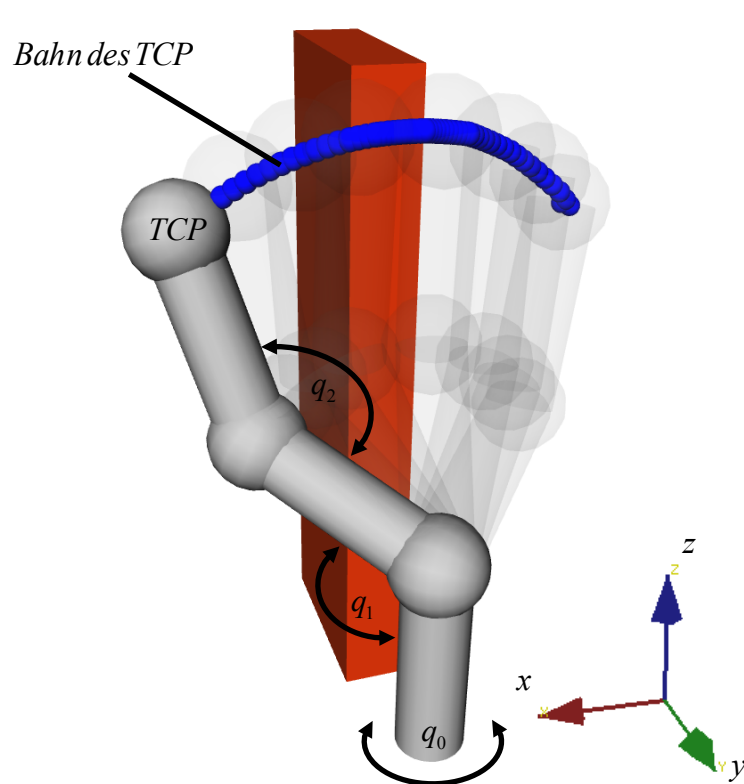
$$\dot{q}(t_e) = 0 \frac{\text{Grad}}{\text{sec}}$$

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

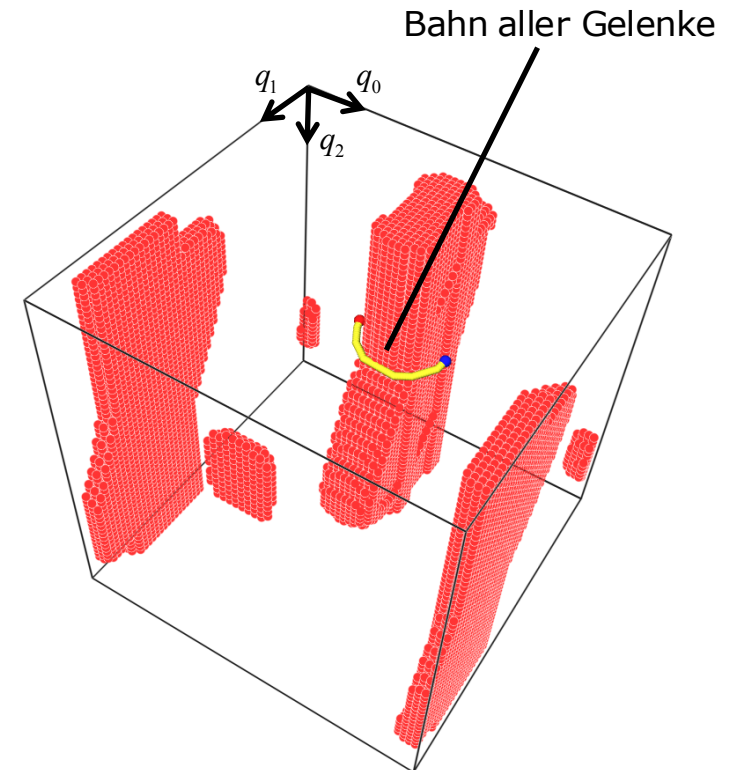


Kartesischer Raum und Gelenkwinkelraum

Darstellung der Zustände



Kartesischen Raum



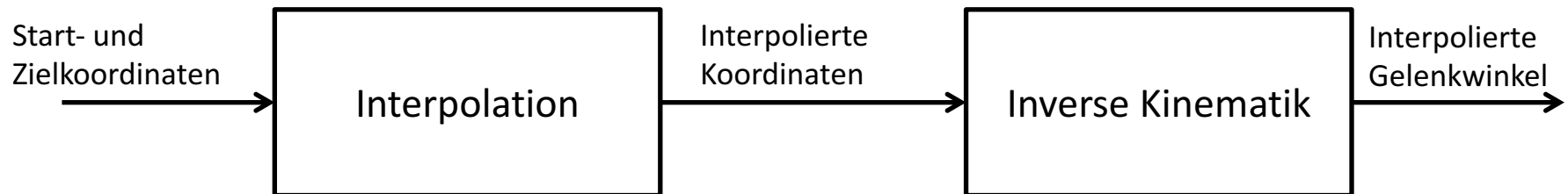
Gelenkwinkelraum

Grundlagen der Bahnsteuerung

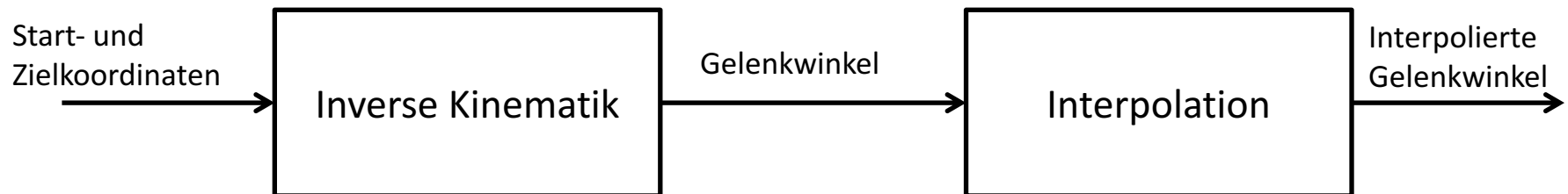
- Zustände können dargestellt werden im
 - Gelenkwinkelraum (Konfigurationsraum): \mathbb{R}^n
 - Kartesischen Raum (Arbeitsraum): $\mathbb{R}^3, SO(3)$
- Bahnsteuerung im Gelenkwinkelraum ist näher an der Ansteuerung der Teilsysteme des Roboters (Gelenke, Sensorik)
- Bahnsteuerung im Kartesischen Raum ist näher an der zu lösenden Aufgabe
 - Bei Steuerung im Kartesischen Raum ist das Lösen der inverse Kinematik nötig

Grundlagen der Bahnsteuerung: Interpolation

■ Interpolation der Weltkoordinaten



■ Interpolation der Gelenkwinkel



Bahnsteuerung im Gelenkwinkelraum

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

Kartesischer Raum (Continuous Path)

- Angabe der Trajektorie erfolgt als Funktion der Zustände des Roboters
 - z.B. mit Beschreibungsvektor des TCP
Position, Geschwindigkeit, Beschleunigung
- 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 die Gelenkwinkelgrenzen
- Interpolation für mehrere Gelenke
- Formulierung der Trajektorie umständlicher

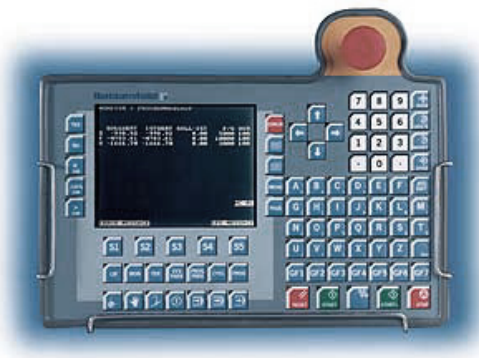
Inhalt

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

Programmierung der Schlüsselpunkte: Teach-In (1)

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: Teach-In (2)

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)



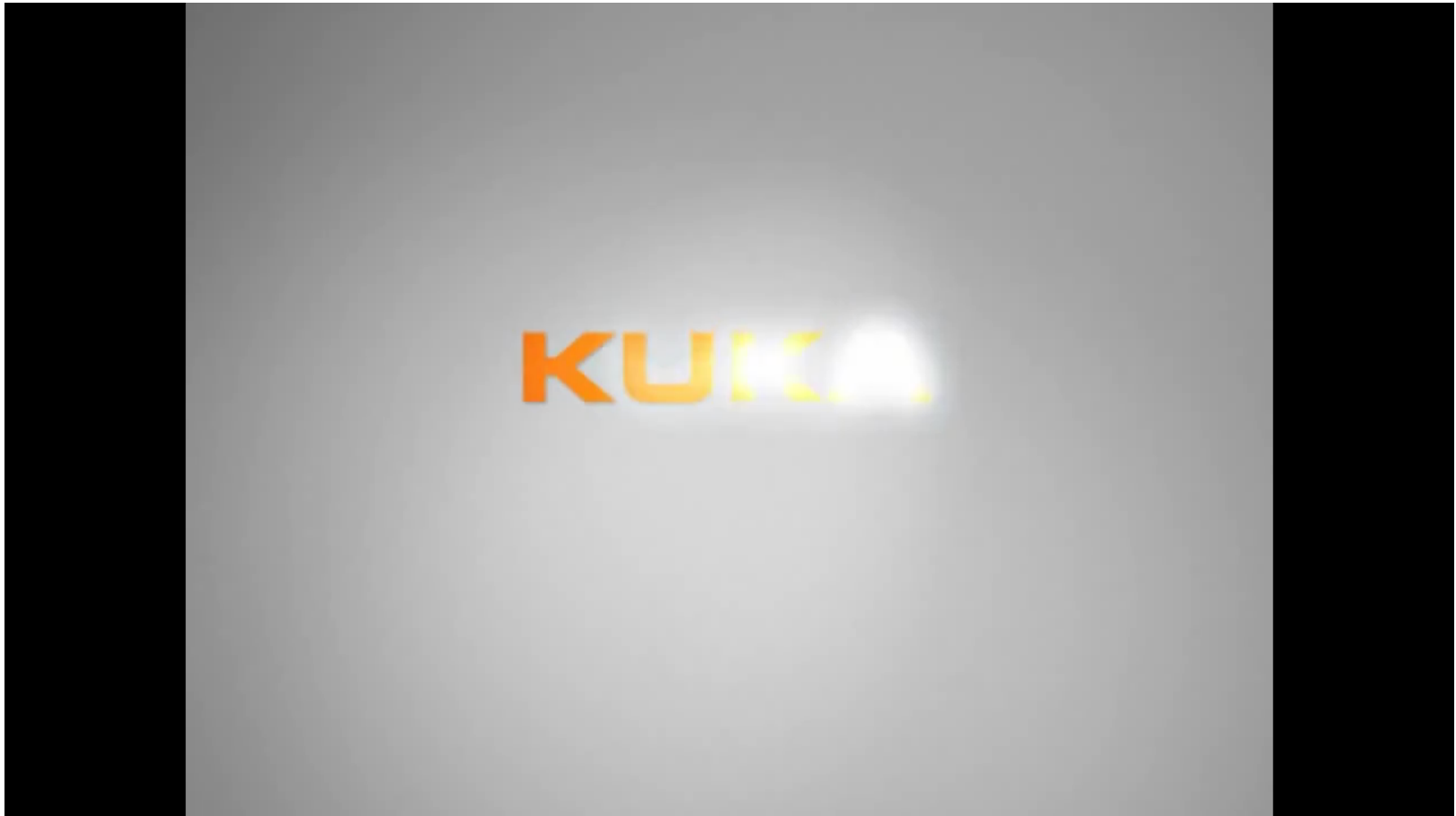
Programmierung der Schlüsselpunkte: Playback (1)

Direkte Programmierung: Playback

- Roboter im Zero-Force-Control Modus
 - 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
 - Integration der handwerklichen Erfahrung
 - Typische Einsatzbereiche sind:
Lackieren oder Kleben

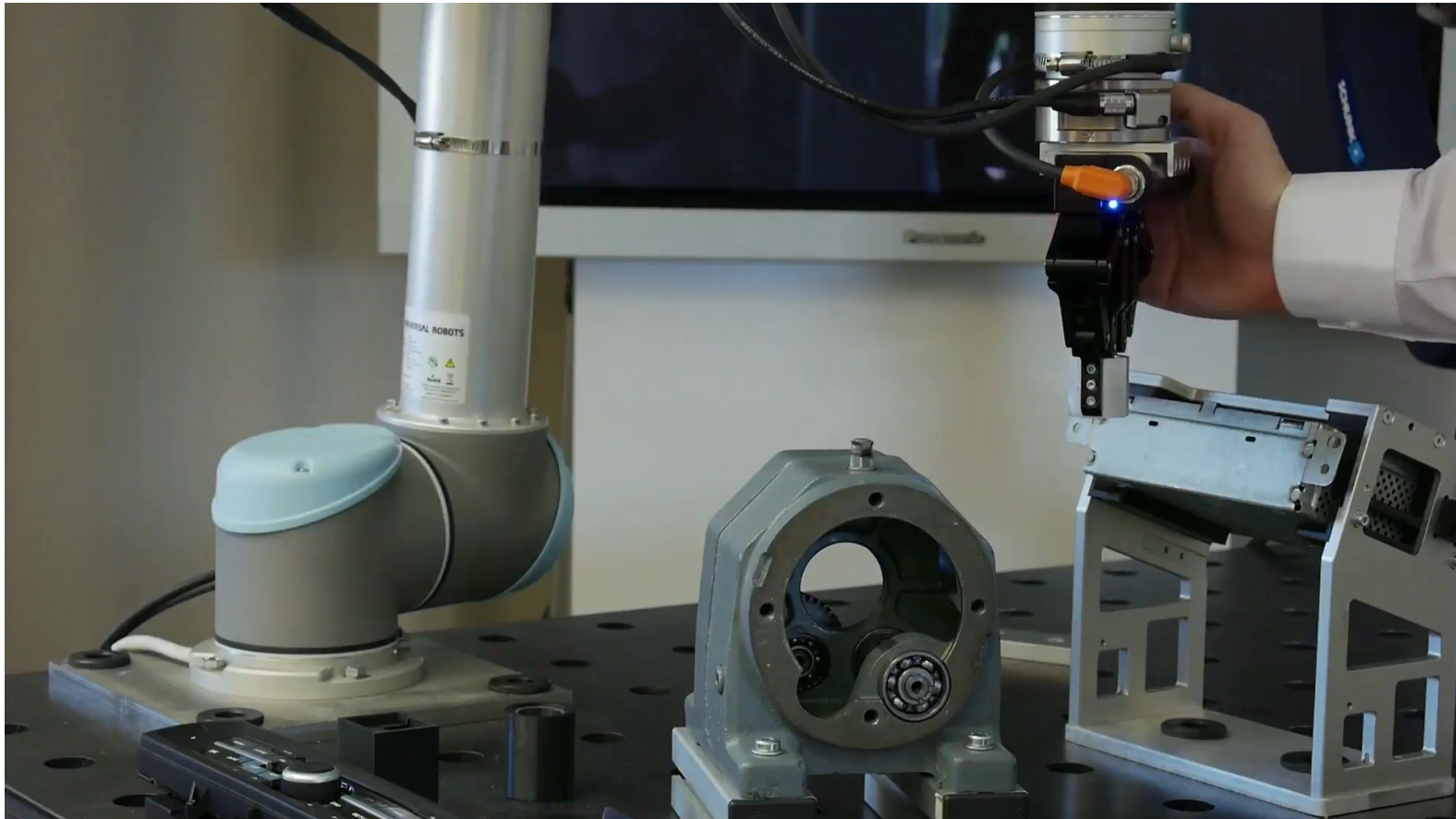
Programmierung der Schlüsselpunkte: Playback (2)

Direkte Programmierung: Playback



Programmierung der Schlüsselpunkte: Playback (3)

Direkte Programmierung: Playback



<http://www.artiminds.com/>

Programmierung der Schlüsselpunkte: Playback (4)

Direkte Programmierung: Vor/Nachteile Playback

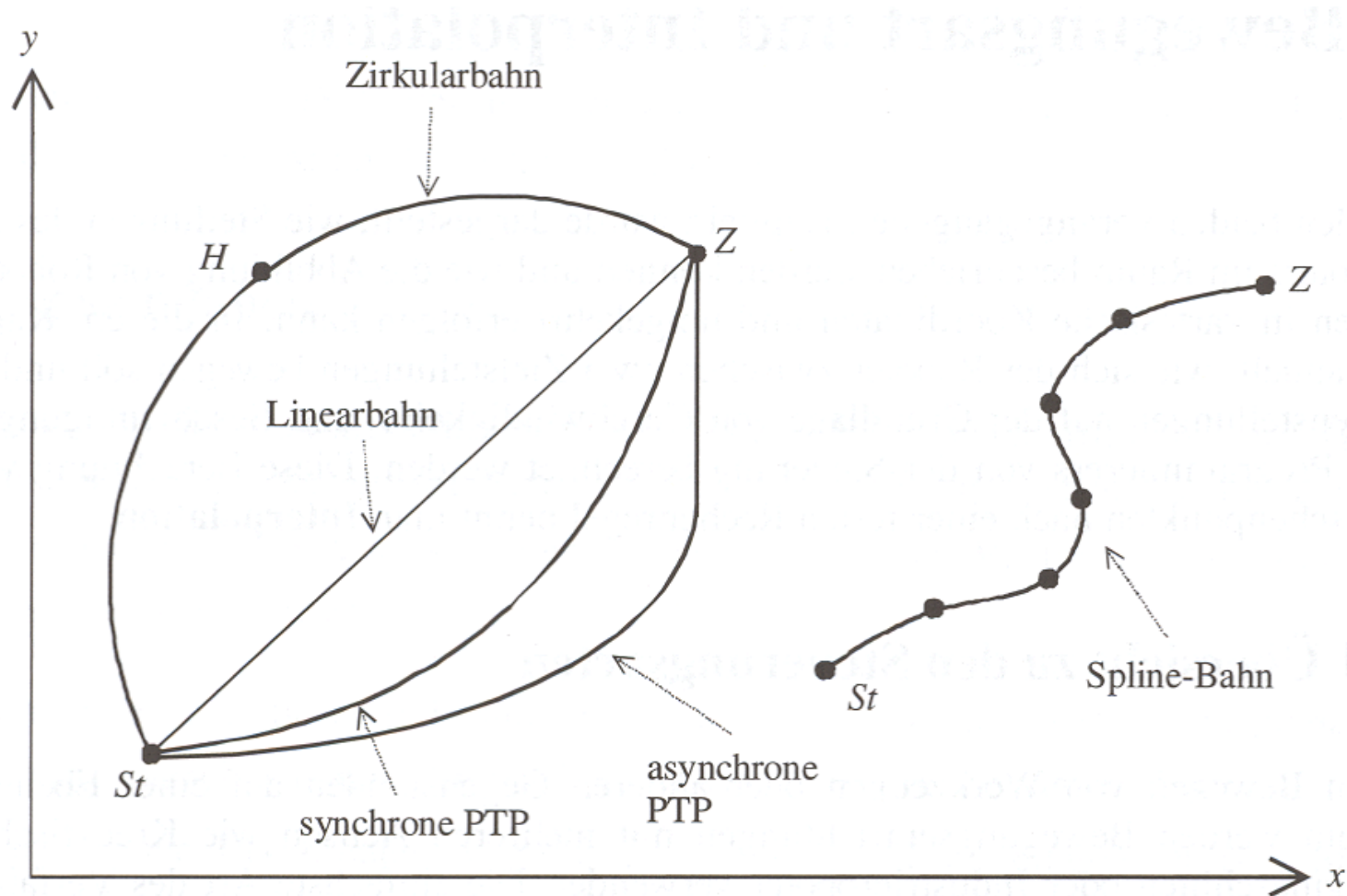
- ✓ Schnell für komplexe Bahnen
- ✓ Intuitiv

- 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: Überblick



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

- Roboter führt **Punkt-zu-Punkt-Bewegung** aus
 - PTP: Point-To-Point
- Vorteile:
 - Die Berechnung der Gelenkwinkeltrajektorie ist einfach
 - Keine Probleme mit Singularitäten
- Sequenz von Gelenkwinkelvektoren

$$\mathbf{q}(t_j) = \left(q_1(t_j), q_2(t_j), \dots, q_n(t_j) \right)^T$$

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

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

■ Randbedingungen

- Start- und Zielzustand sind bekannt
- z.B. Geschwindigkeit zu Beginn und am Ende sind Null
- Der Gelenkwinkelbereich sowie Geschwindigkeiten und Beschleunigungen sind begrenzt (z.B. schnelles Beschleunigen, langsames Abbremsen)

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

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

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

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

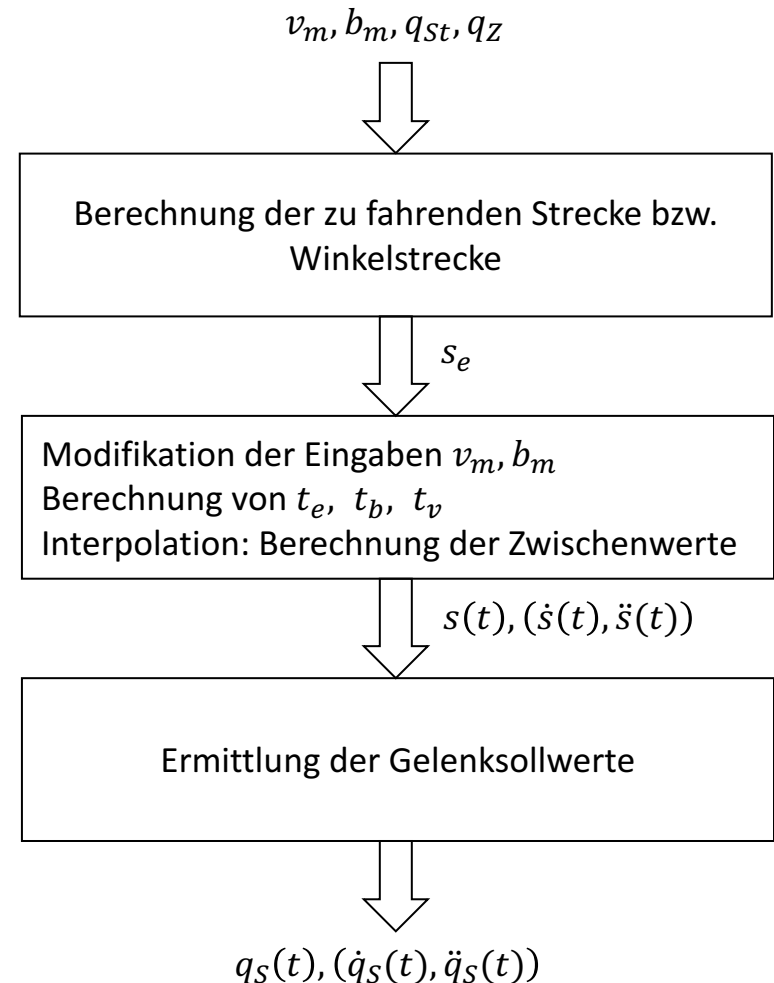
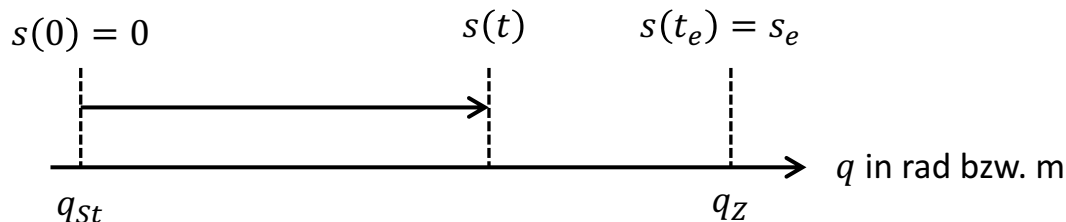
Ablauf der Steuerung

- Fahrzeit t_e
- Beschleunigungszeit t_b
- Beginn der Bremszeit t_v

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

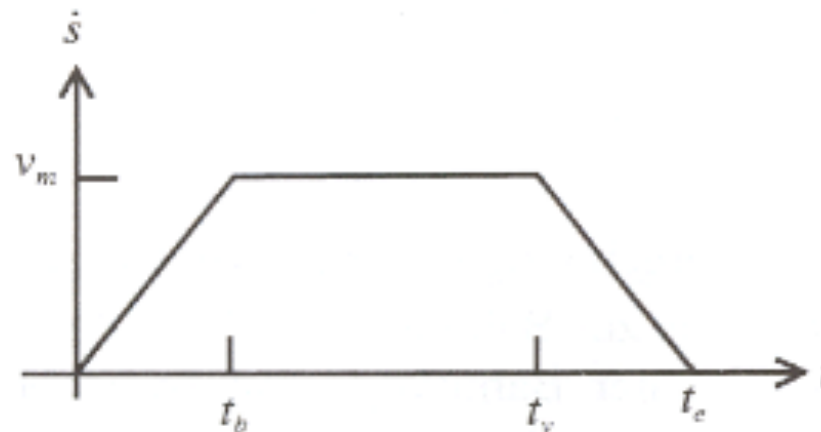
$$s(0) = \dot{s}(0) = v(0) = 0$$

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



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



Interpolation für PTP mit Rampenprofil (2)

■ Phase der Beschleunigung

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

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

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

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

■ Phase der gleichmäßigen Fahrt

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

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

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

Interpolation für PTP mit Rampenprofil (3)

■ Phase des Bremsvorganges

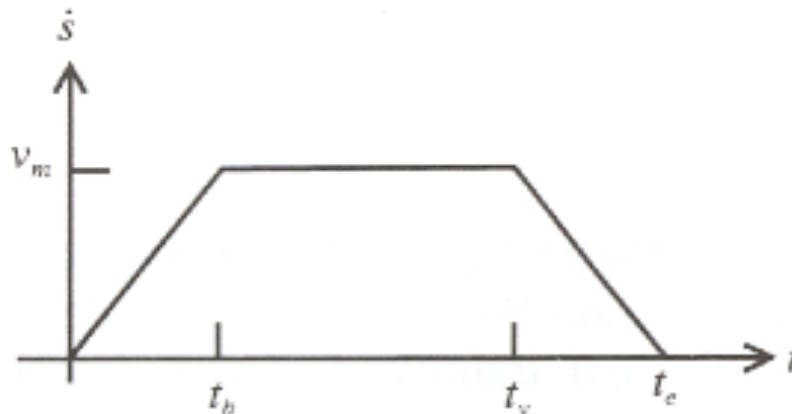
$$\ddot{s}(t) = -b_m \quad t_v \leq t \leq t_e$$

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

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

■ Berechnung der Fahrtzeit

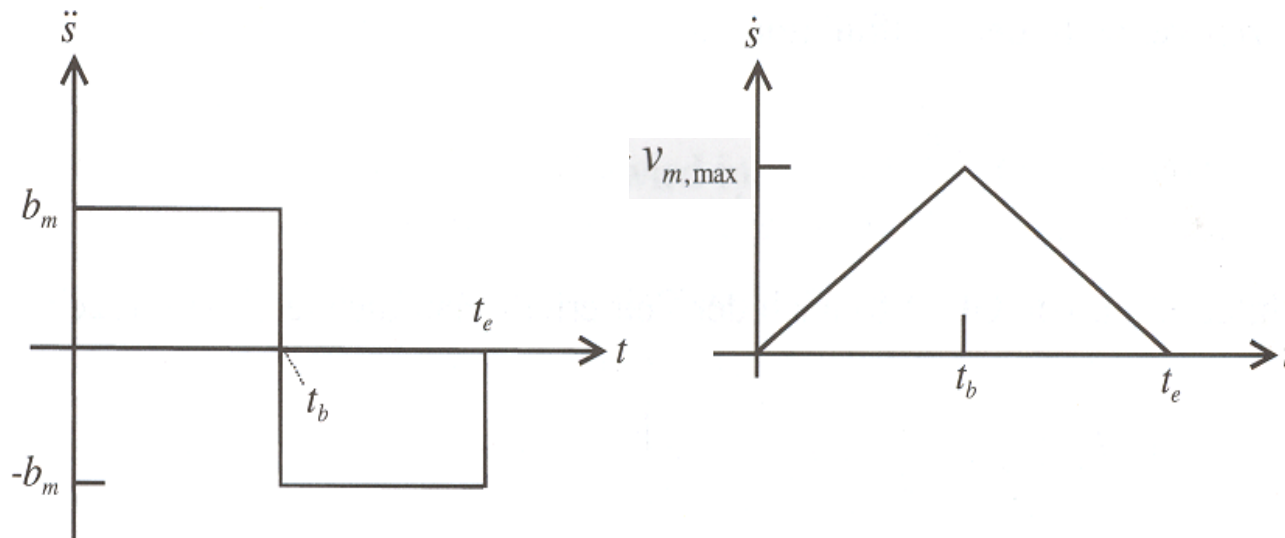
$$t_e = \frac{s_e}{v_m} + t_b = \frac{s_e}{v_m} + \frac{v_m}{b_m}$$



Zeitoptimale Bahn

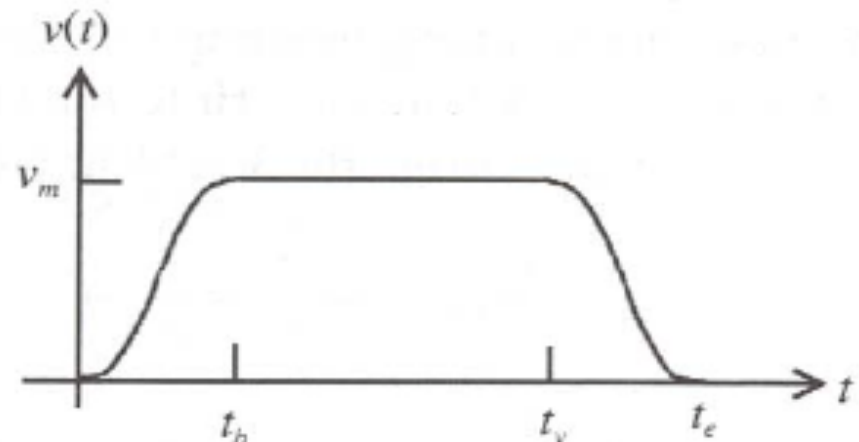
- Falls v_m zu groß in Bezug auf Beschleunigung und Bahnlänge
 - Bestimmung einer zeitoptimalen Bahn nach

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



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



Interpolation für PTP mit Sinoidenprofil (2)

■ Phase der Beschleunigung

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

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

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

■ Aus $\dot{s}(t_b) = b_m \frac{1}{2} t_b = v_m$ folgt $t_b = \frac{2v_m}{b_m}$

■ Phase der gleichmäßigen Fahrt

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

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

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

Interpolation für PTP mit Sinoidenprofil (3)

■ Phase des Bremsvorganges

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

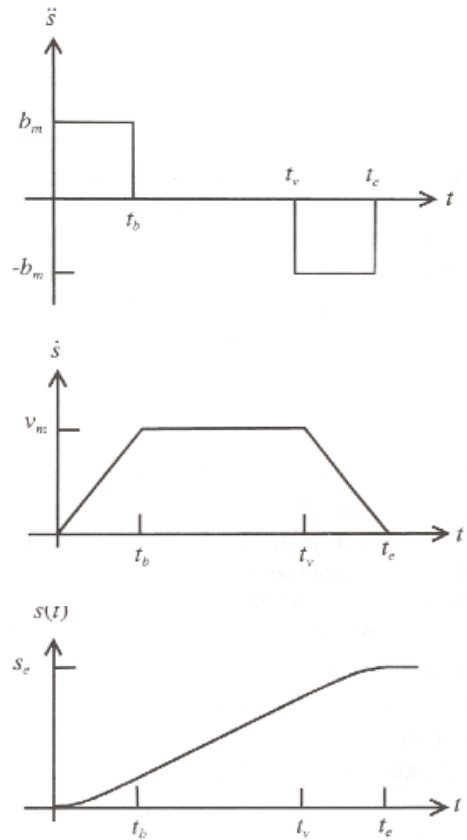
$$s(t) = s(t_v) + \int_{t-t_v}^t \dot{s}(\tau - t_v) d\tau = \frac{b_m}{2} \left(t_e (t + t_b) - \frac{t^2 + t_e^2 + 2 t_b^2}{2} + \frac{t_b^2}{4\pi} \left(1 - \cos \left(\frac{2\pi}{t_b} (t - t_v) \right) \right) \right)$$

■ Berechnung der Fahrtzeit

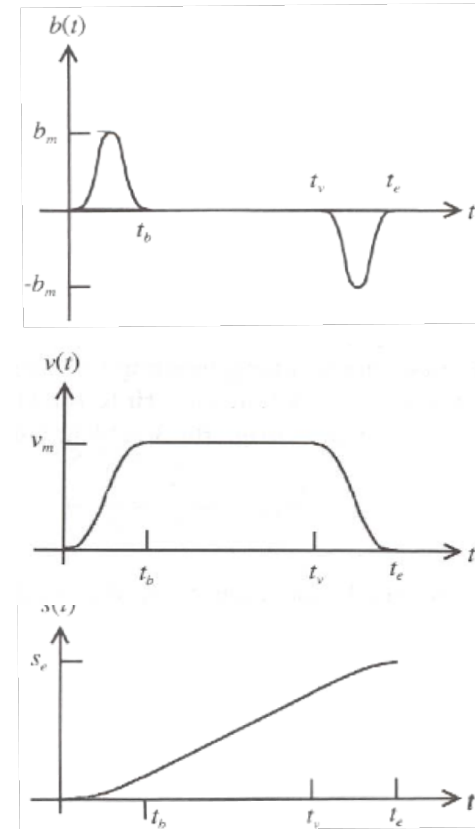
$$t_e = \frac{s_e}{v_m} + t_b = \frac{s_e}{v_m} + \frac{2v_m}{b_m}$$

Interpolationsarten: Rahmen- vs. Sinoidenprofil

Rampenprofil



Sinoidenprofil



Synchrone PTP-Bahnen (1)

- Vorgehen bei **synchronen** PTP-Bahnen
 - Bestimme für jedes Gelenk i die 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

Synchrone PTP-Bahnen (2)

Bestimme die neuen maximalen Geschwindigkeiten für alle Gelenke

- Umformung Fahrzeit und Berechnung der neuen Geschwindigkeiten
 - Rampenprofil:

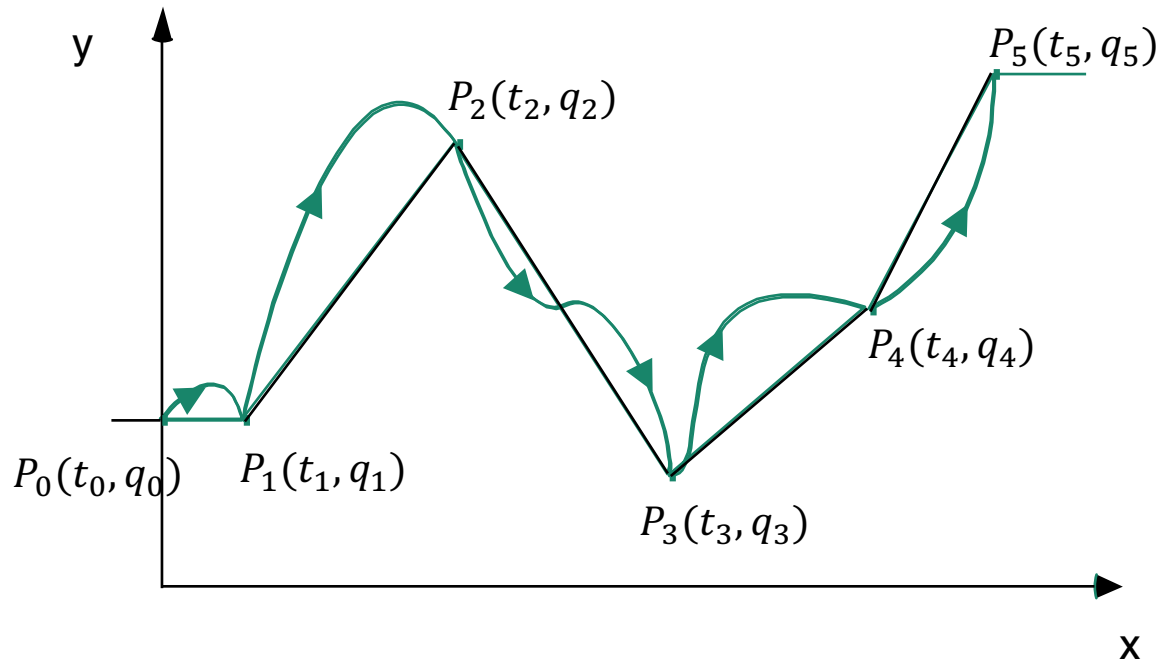
$$t_e = \frac{s_{e,i}}{v_{m,i}} + \frac{v_{m,i}}{b_{m,i}} \rightarrow v_{m,i}^2 = v_{m,i} b_{m,i} t_e + s_{e,i} b_{m,i}$$

- Analoge Berechnung für Sinoidenbahn:

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

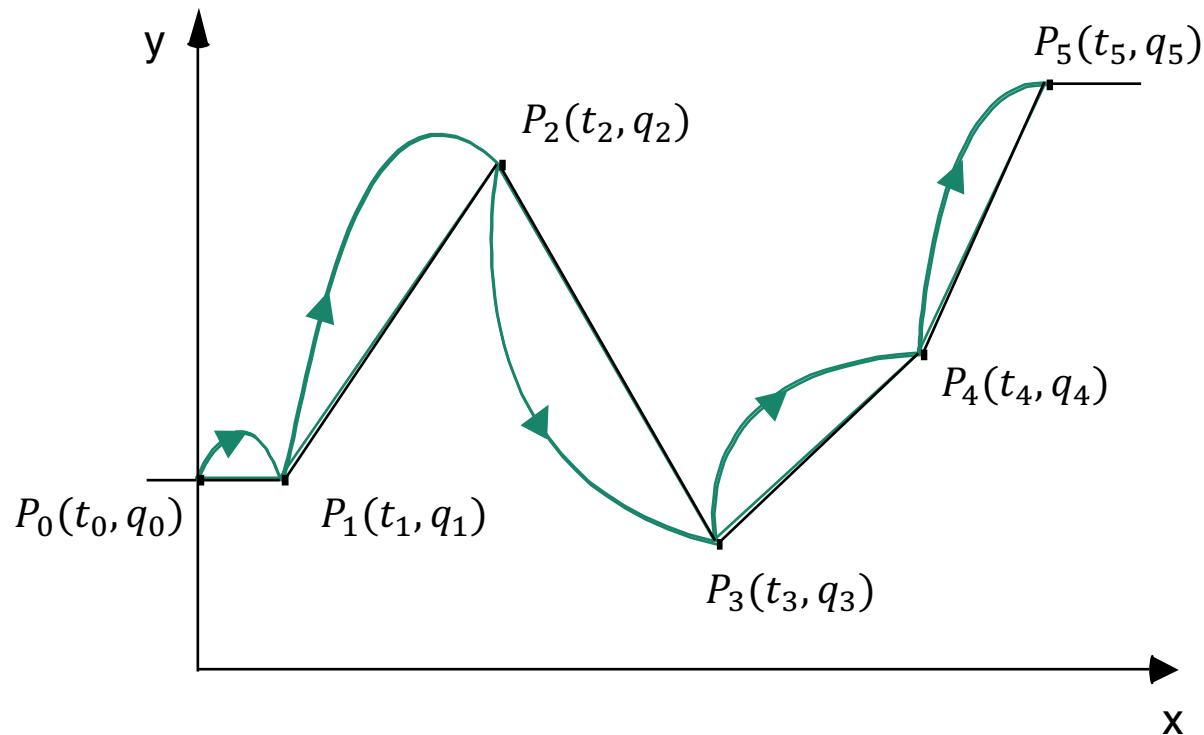
$$v_{m,i} = \frac{b_{m,i} t_e}{4} - \sqrt{\frac{b_{m,i}^2 t_e^2 - 8 s_{e,i} b_{m,i}}{16}}$$

Asynchrone PTP-Bahnen



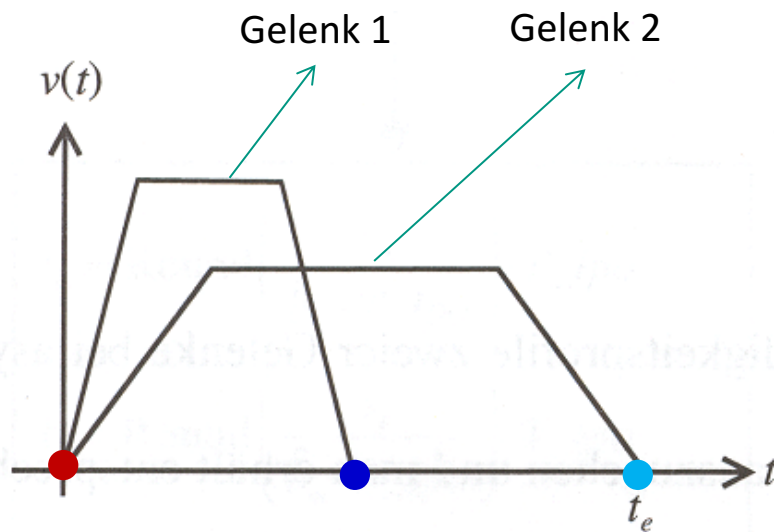
- Jedes Gelenk wird sofort mit der maximalen Beschleunigung angesteuert.
- Jede Gelenkbewegung endet unabhängig von den anderen.

Synchrone PTP-Bahnen

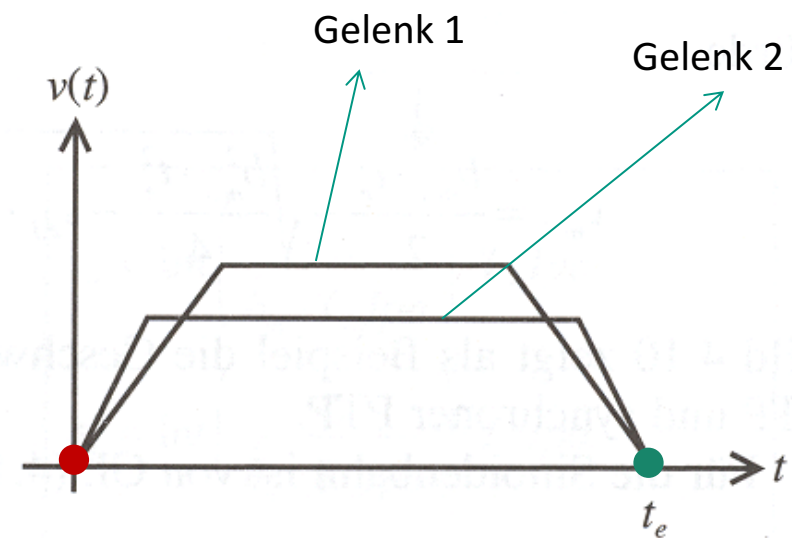


- Alle Gelenke beginnen und beenden ihre Bewegungen gemeinsam (synchron).

Vergleich: Asynchrone und synchrone PTP-Bahnen



a) Asynchron



b) Synchron

Vollsynchrone PTP-Bahnen

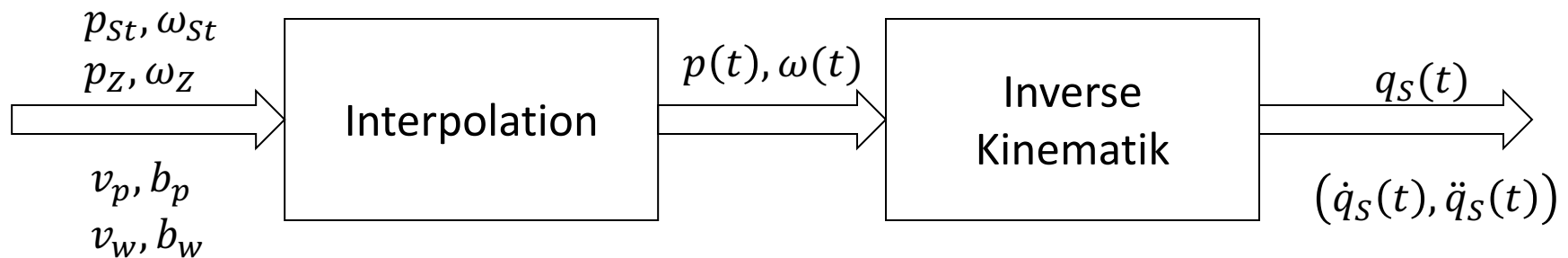
- Zusätzliche Berücksichtigung der Beschleunigungs- und Bremszeit
- Bessere Annäherung der Start- und Zielpunkte im Kartesischen Raum
- Bestimmung Leitachse mit t_e und $t_b \rightarrow t_v = t_e - t_b$
- Bestimmung der Geschwindigkeit und Beschleunigung der anderen Achse mit

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

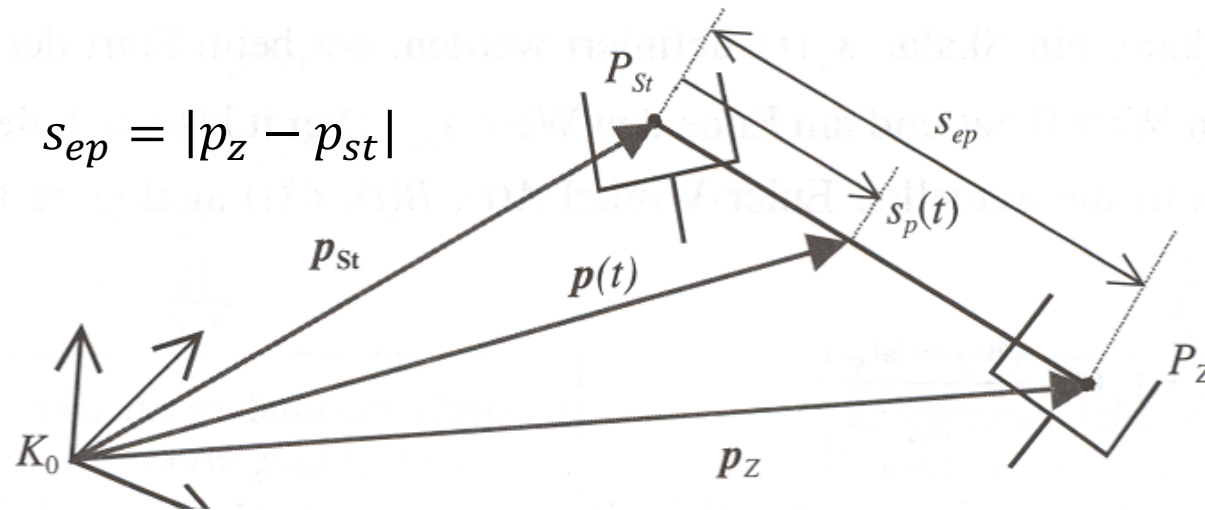
- Nachteil: Beschleunigung jeder Achse wird vorgegeben

Steuerung im Kartesischem Raum

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



Linearinterpolation (1)



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

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

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

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

$$v_m = v_p, b_m = b_p, t_e = t_{ep}, t_b = t_{bp}, t_v = t_{vp}, S_e = S_{ep}, S = S_p$$

Linearinterpolation (2)

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

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

- Berechnung von $s_\omega(t)$ mit Rampen- oder Sinoidenprofil:

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

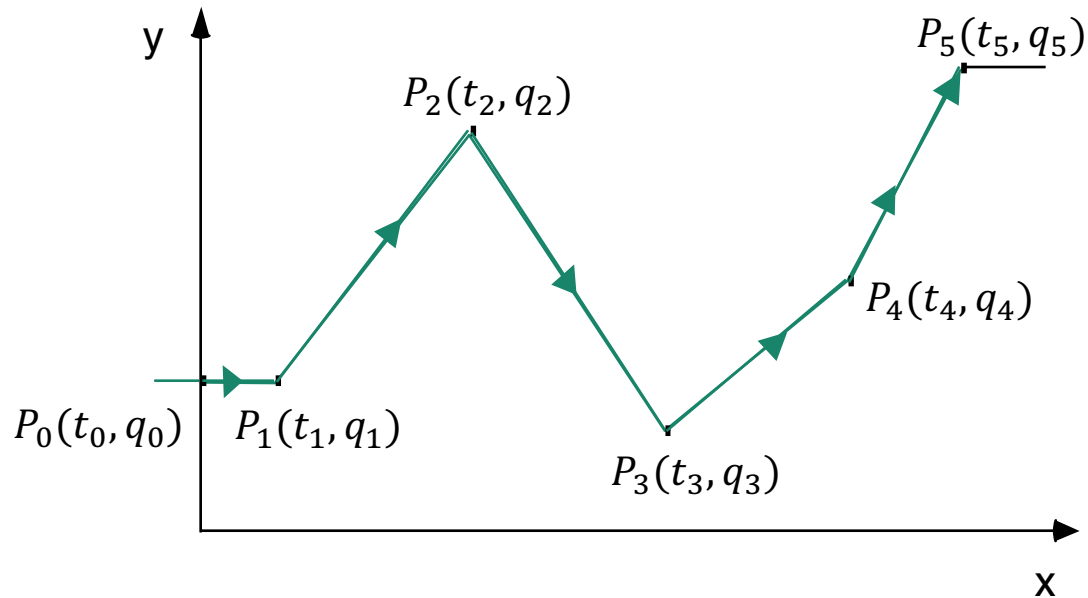
- Angleich der Fahrzeiten t_{ep} (Position) und $t_{e\omega}$ (Orientierung)

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

- Analog zur Anpassung der Geschwindigkeiten bei synchronen PTP

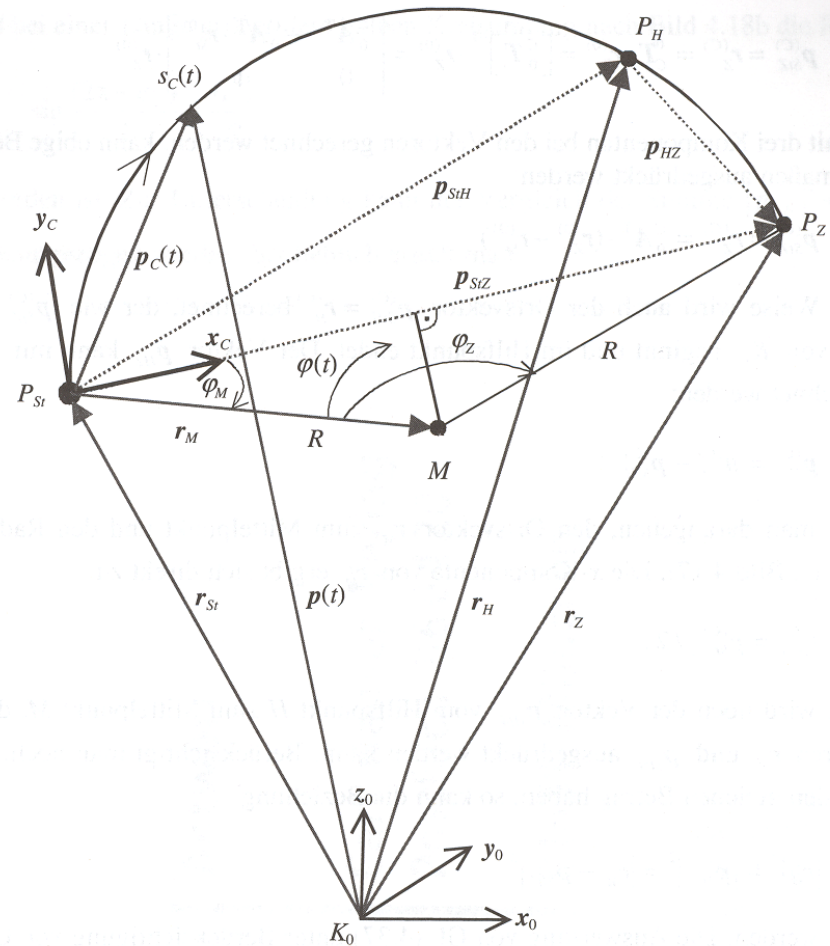
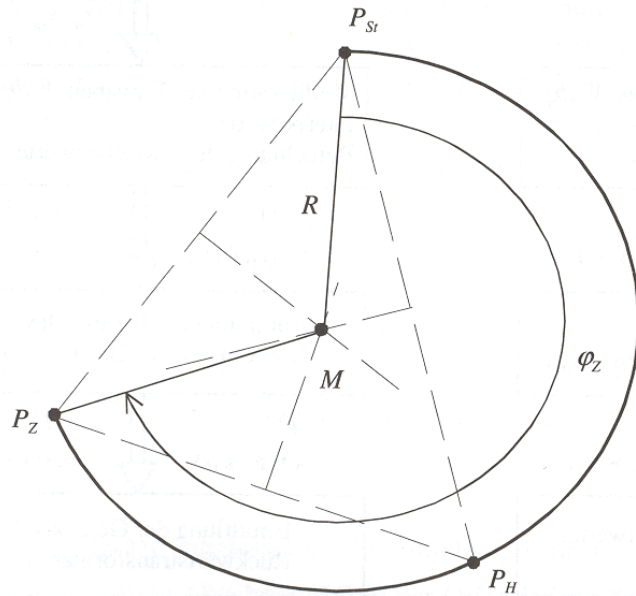
- Falls $t_e = t_{ep}$:
$$v_\omega = \frac{b_\omega t_e}{2} - \sqrt{\frac{b_\omega^2 t_e^2}{4} - s_{e\omega} b_\omega}$$

- Falls $t_e = t_{e\omega}$:
$$v_p = \frac{b_p t_e}{2} - \sqrt{\frac{b_p^2 t_e^2}{4} - s_{ep} b_p}$$



- Die Robotersteuerung interpoliert die Bahn zwischen je 2 Teiltrajektorien.

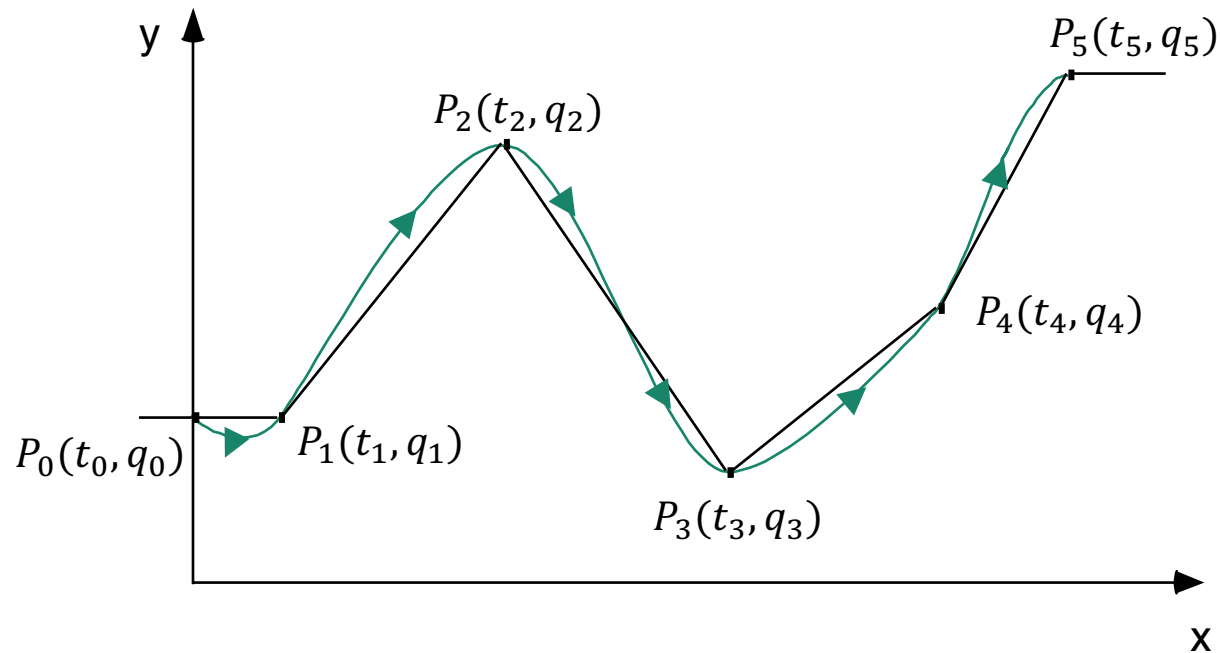
Zirkularinterpolation



Hilfspunkt P_H wird benötigt

Bestimme Mittelpunkt M , Radius R und φ_Z

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)

Beispiel: Kubische Splines (1)

■ Gegeben

■ Polynom:

$$f(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (a_0, a_1, a_2, a_3 \in \mathbb{R}) \quad (1)$$

■ Start und Ziel:

$$S_{\text{Start}}, S_{\text{Ziel}} \quad (2)$$

■ Ansatz

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

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

■ Aus (1) und (2) folgt

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

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

Beispiel: Kubische Splines (2)

- Aus Randbedingungen $\dot{f}(t_{start}) = \dot{f}(t_{Ziel}) = 0$ und (3) folgt:

- $\dot{f}(t_{start}) = \dot{f}(0) = a_1 = v_{start}$ (7)

- $\dot{f}(t_{Ziel}) = a_1 + 2a_2t_{Ziel} + 3a_3t_{Ziel}^2 = v_{Ziel}$ (8)

- Aus (7),(8) folgt:

- $a_2 = \frac{v_{Ziel} - v_{start}}{2t_{Ziel}} - \frac{3}{2}a_3t_{Ziel}$ (9)

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

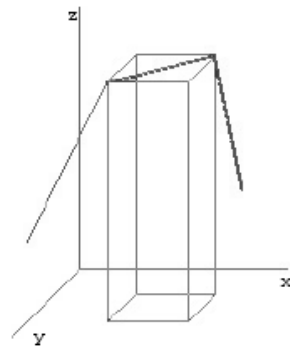
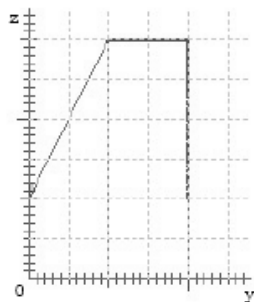
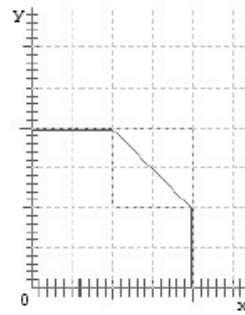
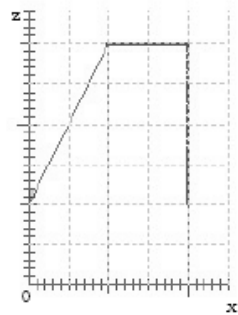
- $a_3 = \frac{2(S_{start} - S_{Ziel})}{t_{Ziel}^3} + \frac{(v_{start} - v_{Ziel})}{t_{Ziel}^2}$ (10)

- Aus (9), (10) folgt:

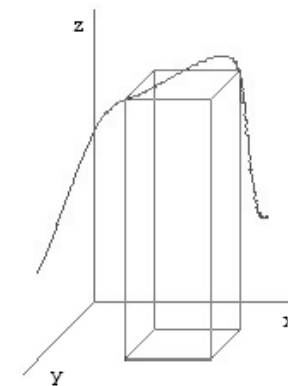
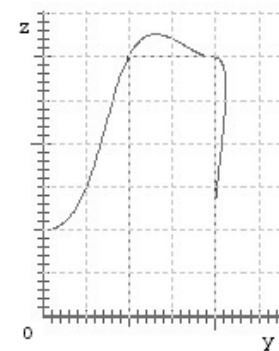
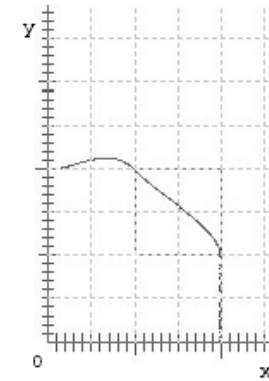
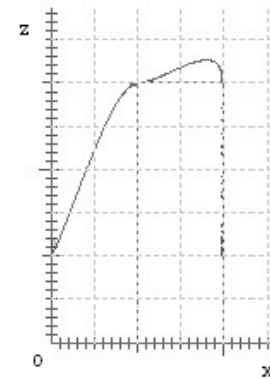
- $a_2 = \frac{3(S_{Ziel} - S_{start})}{t_{Ziel}^2} - \frac{(v_{Ziel} + 2v_{start})}{t_{Ziel}}$

Beispiel: Splines

■ Bahn (4 Stützpunkte)



■ Splineinterpolation



Inhalt

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

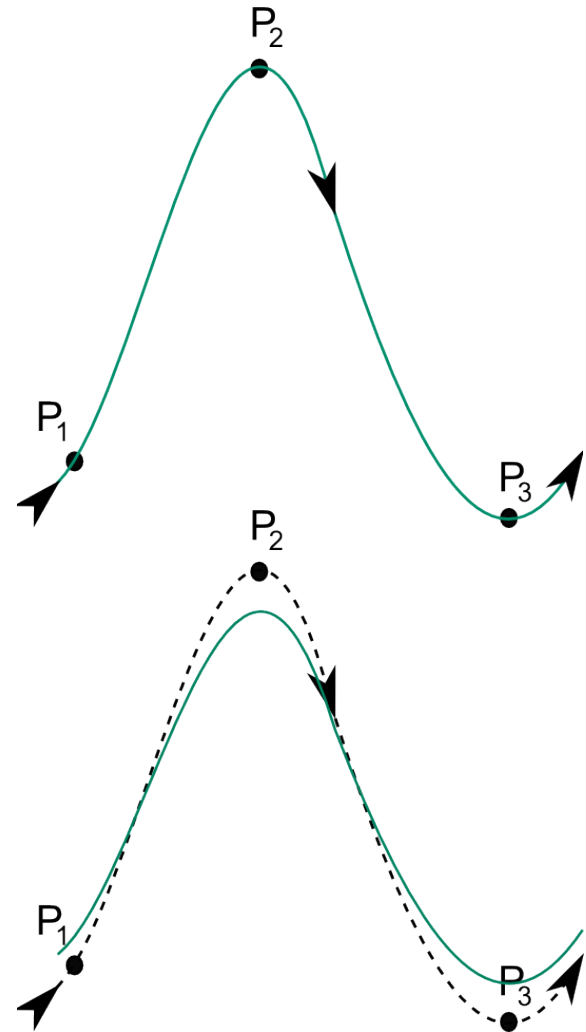
Approximierte Bahnsteuerung: Definition

■ Bahninterpolation

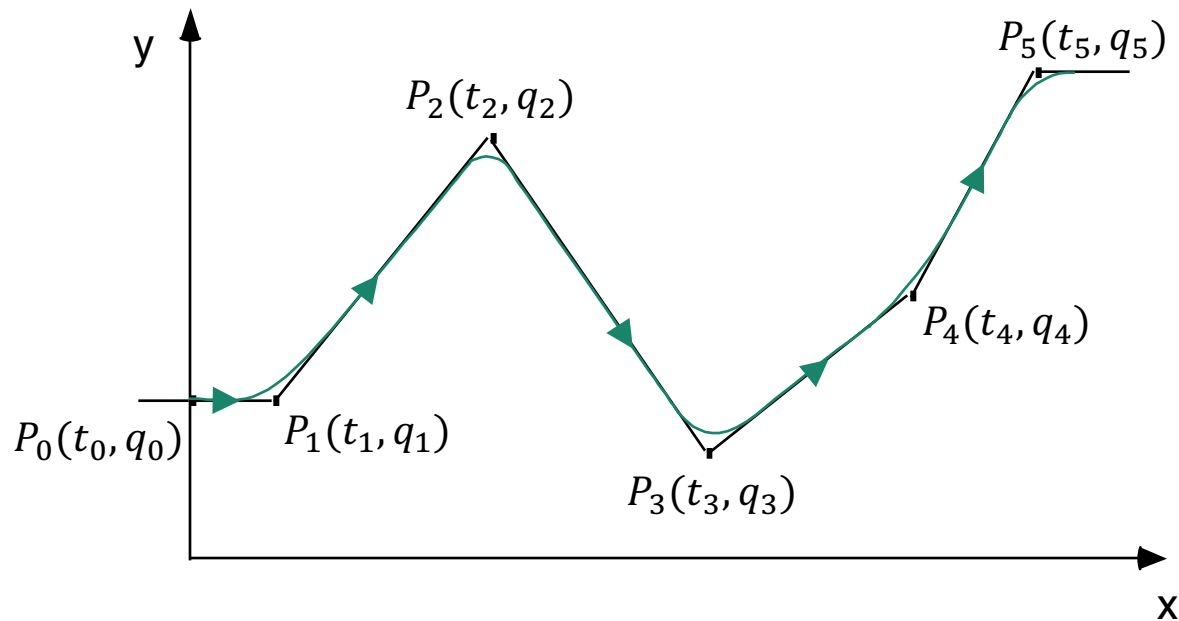
- Die ausgeführte Bahn verläuft durch alle Stützpunkte der Trajektorie

■ Bahnapproximation

- Die Kontrollpunkte beeinflussen den Bahnverlauf und werden approximiert

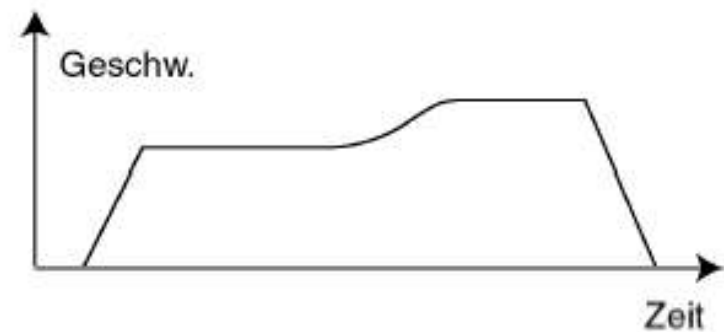
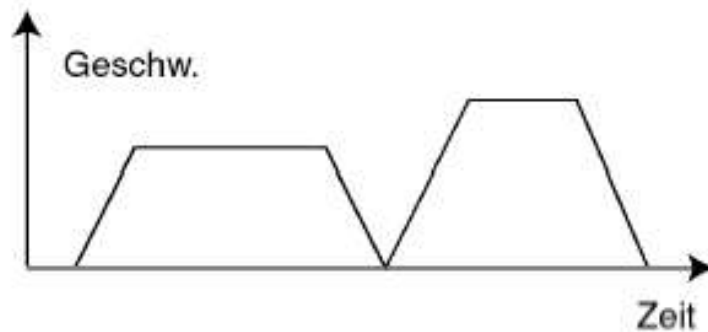
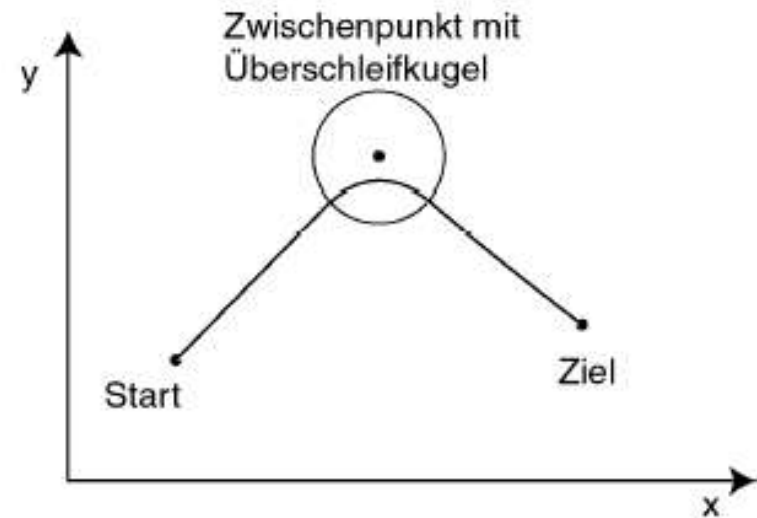
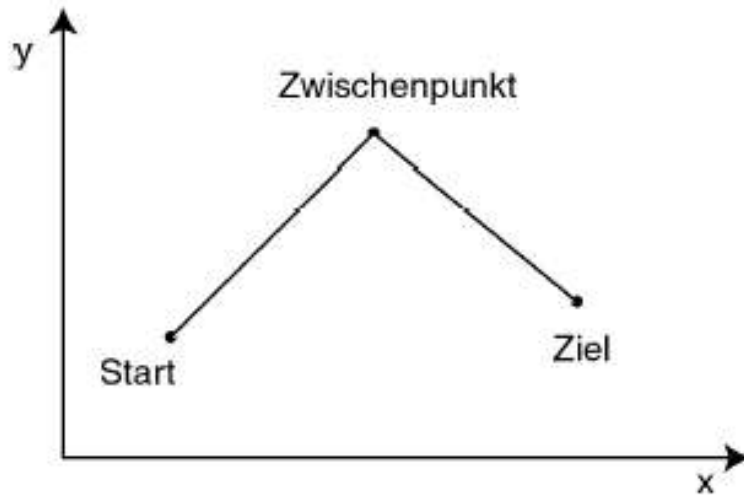


PTP und CP mit Überschleifen (1)



- Zum Zeitpunkt $t_j - \varepsilon$ 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)



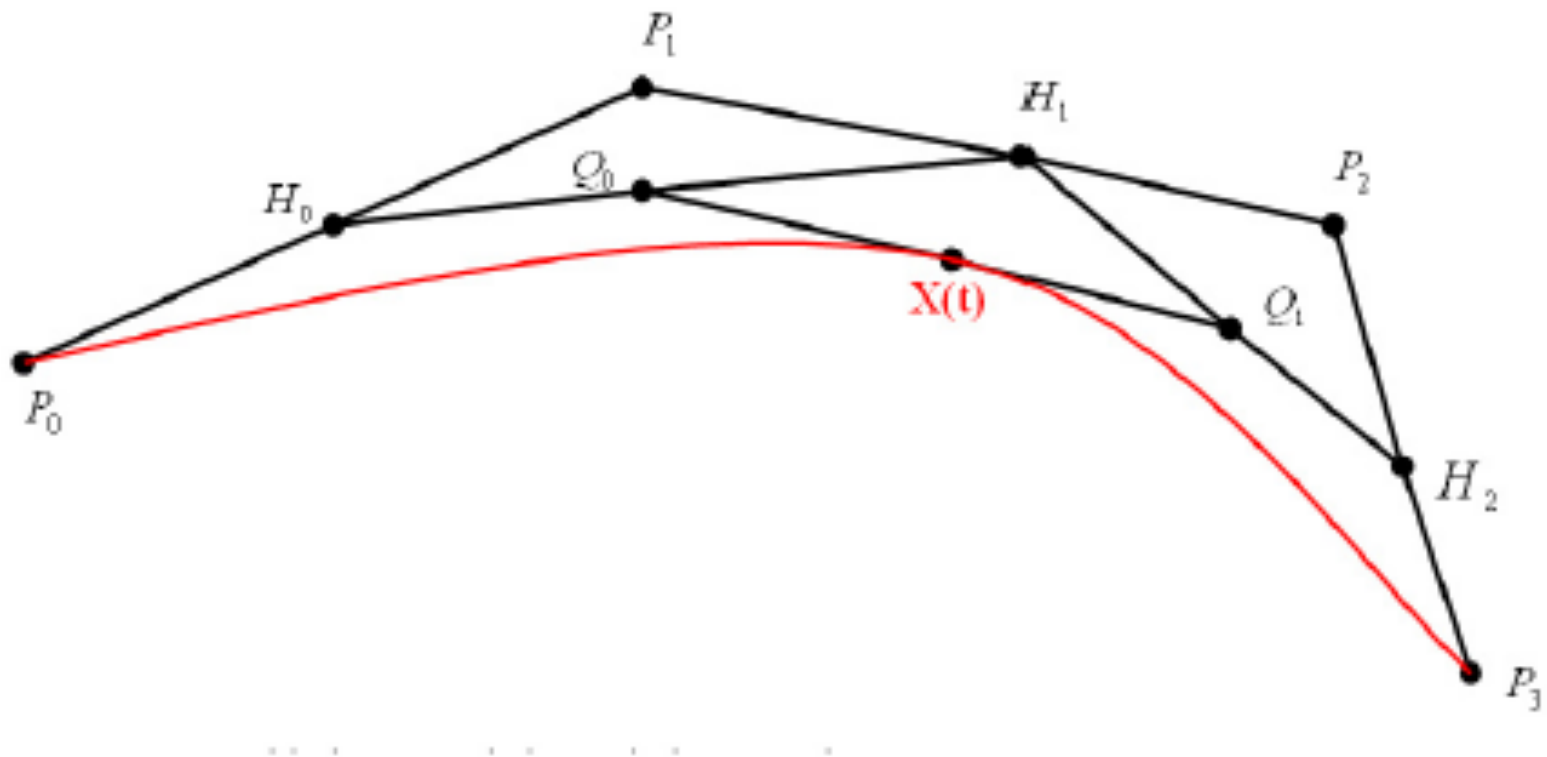
PTP und CP mit Überschleifen (3)

- 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: Bézierkurven (1)

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

- Basisfunktion:

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

- $B_{i,n}(t)$: i -tes Bernsteinpolynom n -ten Grads

Approximierte Bahnsteuerung: Bézierkurven (2)

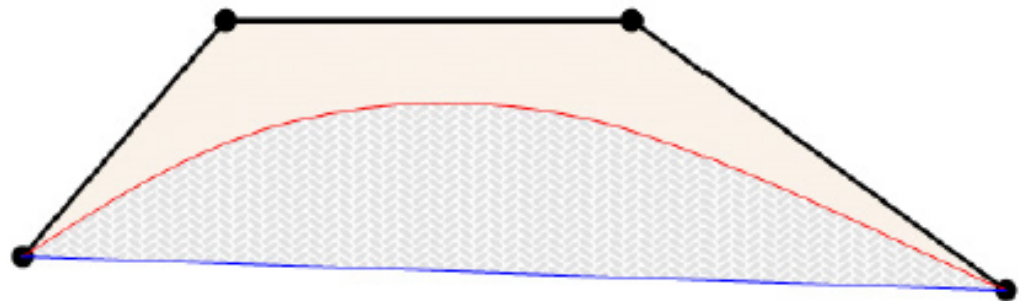
- Berechnung beliebiger Zwischenstellungen

- Bernsteinpolynom für kubischen Fall

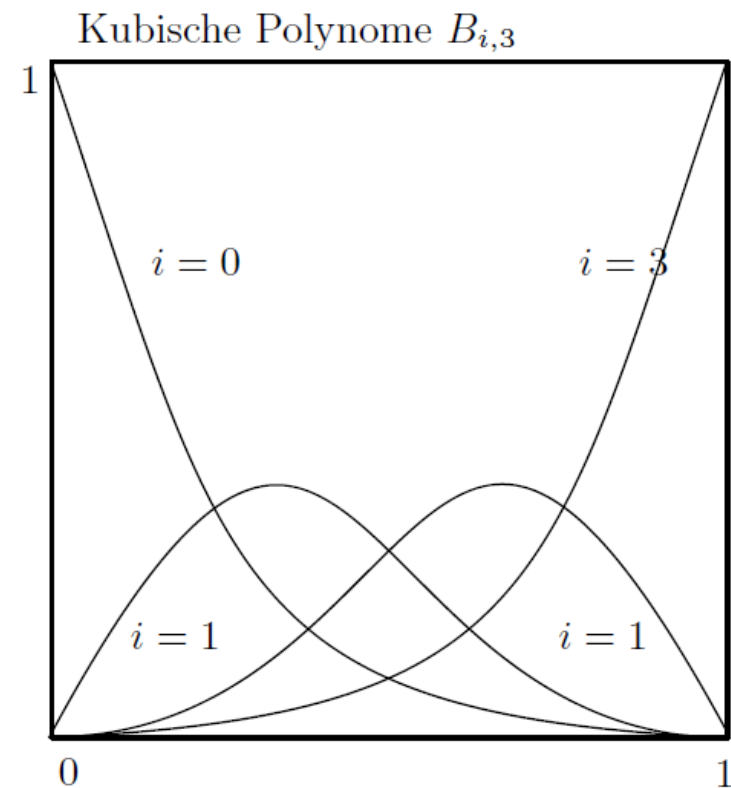
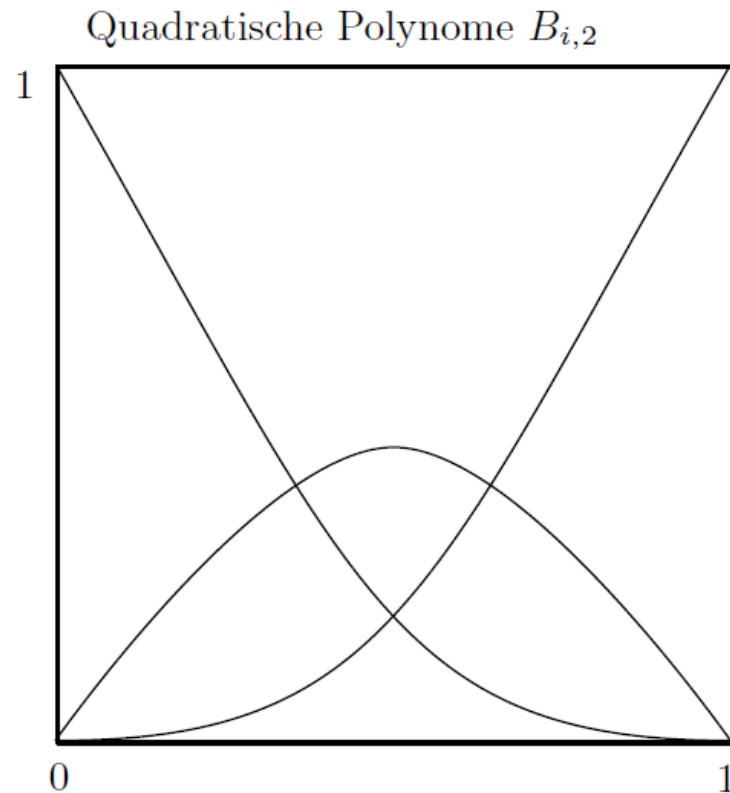
$$B_{i,3}(t) = \binom{3}{i} t^i (1-t)^{3-i}$$

$$p(t) = p_0(1-t)^3 + 3p_1(1-t)^2t + 3p_2(1-t)t^2 + p_3t^3$$

- Annähern von unten an Stützstellen
- keine beliebige Form



Beispiele für Bernsteinpolynome

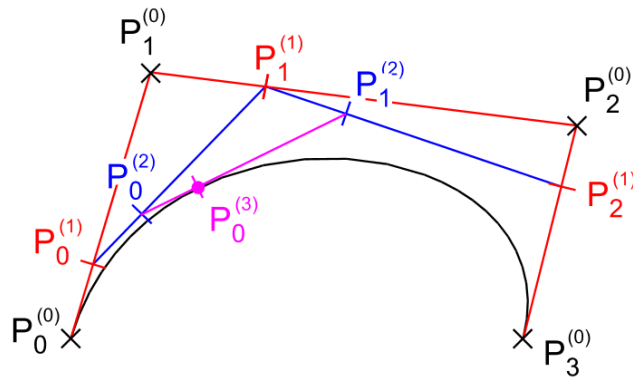


Der De-Casteljau-Algorithmus (1)

- Annäherung an die Bézierkurve: Effiziente Berechnung
Näherungsdarstellung von Bézierkurven durch einen Polygonzug
- Iterative Berechnung:
Kann auch für große n effizient berechnet werden
- Gegeben: n Kontrollpunkte P_0, \dots, P_{n-1}
- Start: $P_i^0 = P_i$
- Iteration k : $P_i^{k+1} = (1 - t_0)P_i^k + t_0P_{i+1}^k$

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

