



Vorlesung Modellierung und Simulation in der Mechatronik

Lehrinhalt:

Weihnachtsprojekt

Prof. Dr.-Ing. Jörg Fehr

Vorlesungsmaterial:

https://ilias3.uni-stuttgart.de/goto_Uni_Stuttgart_crs_34_25429.html



W.1 Trajektorienplanung

W.1.1. Bahnplanung und Navigation

Bahnplanung: Eine konzeptionelle Beschreibung des Problems ist sehr einfach.

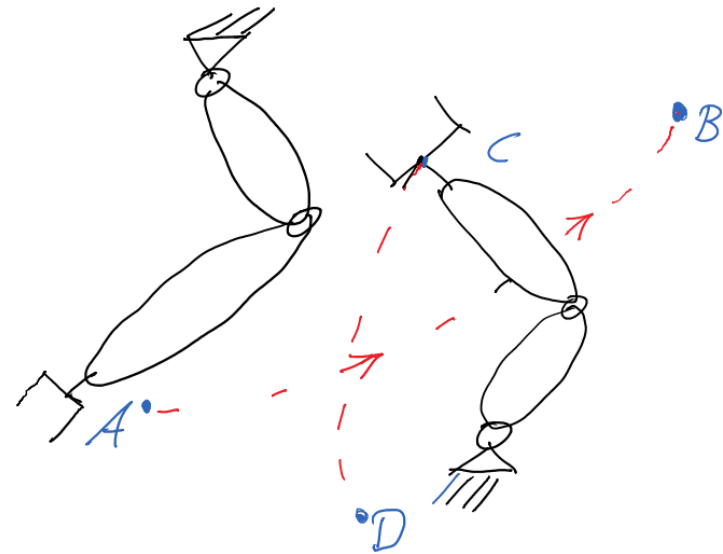
„Fahre von A nach B und weiche allen Hindernissen aus.“

Aus Sicht der Informatik sind

Bahnplanungsprobleme eine der schwierigsten Aufgaben (Komplexität $O(e^x)$).

W.1.2 Bahnplanungsmethoden:

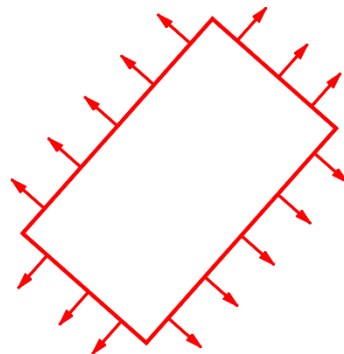
- Potentialfeldmethode



Roboter



Hindernis (abstoßendes Potential)



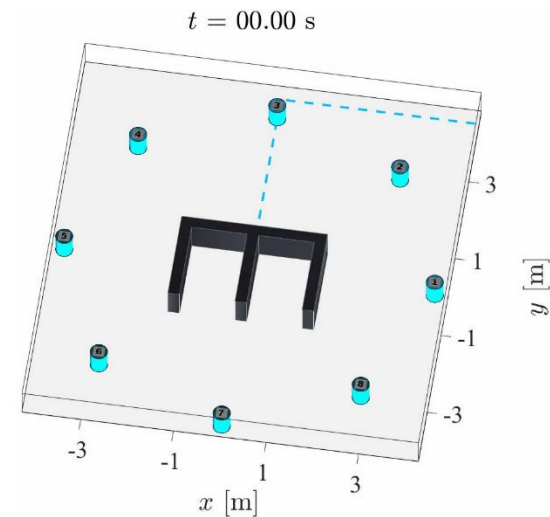
Ziel (minimales Potential)

- Partial Swarm Optimierer
- Genetische Algorithmen
- Modellprädiktive Ansätze
 - Forschungsarbeiten Henrik Ebel / Eshan Sharafian

Ziel:

Transport eines Objektes mit Hilfe eines sich selbst organisierenden Schwarms von mobilen Robotern.

Roboter agieren **unabhängig**, können jedoch jeden k -th Zeitschritt miteinander **kommunizieren**.

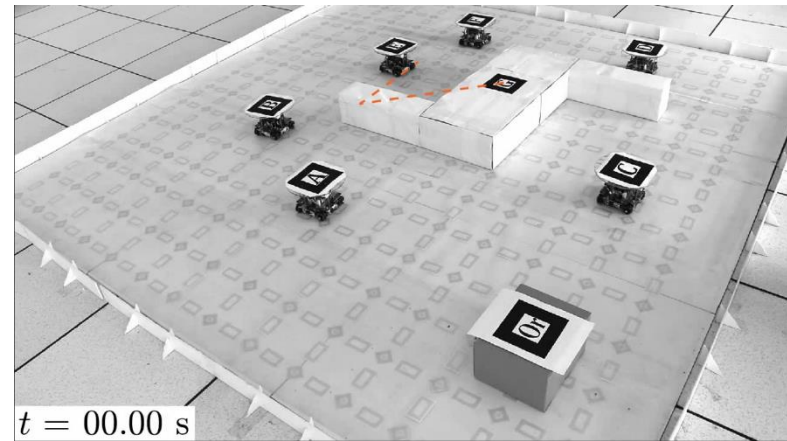


Anwendungen in der Logistik, jedoch ebenfalls ein gutes Beispiel für die **Theoretischen Konzepte der verteilten Regelung**.

Voraussetzungen:

Jede Entscheidung muss **unabhängig** von einer **zentralen Überwachungseinheit** getroffen werden

Szenario Agnostik – d.h. **kein Parametertuning**



Methoden:

Vielversprechende Ergebnisse mit **Distributed model predictive control** (DMPC), auch bei Hardwareexperimenten Verwendung der (D)MPC

⇒ Vorteil: Formulierung der Ziele als Optimierungsproblem mit Nebenbedingung

Model Predictive Control (MPC)

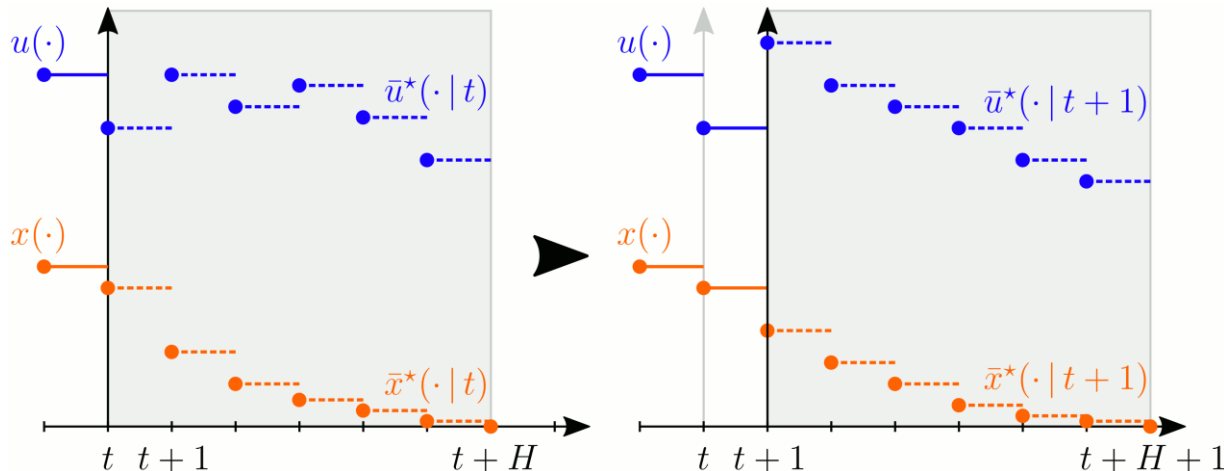
Modellprädiktive Regelung eines (hier zeitdiskreten) Systems

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k), \mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^n$$

mit gleichzeitiger

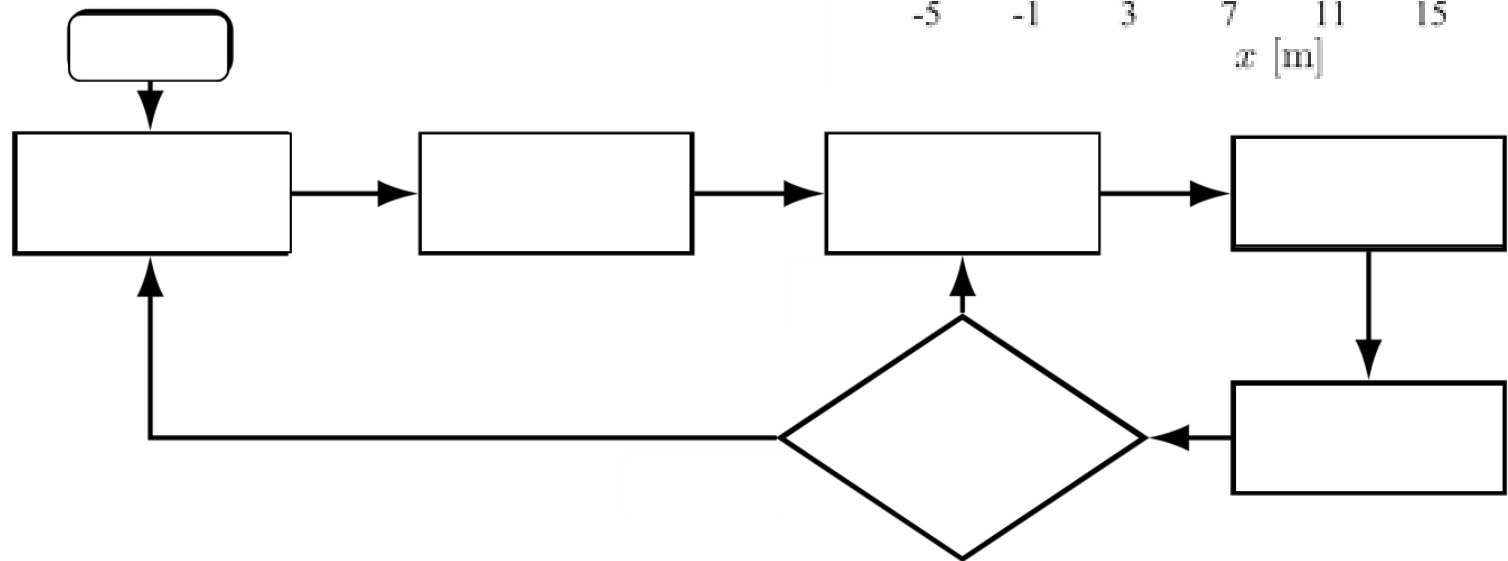
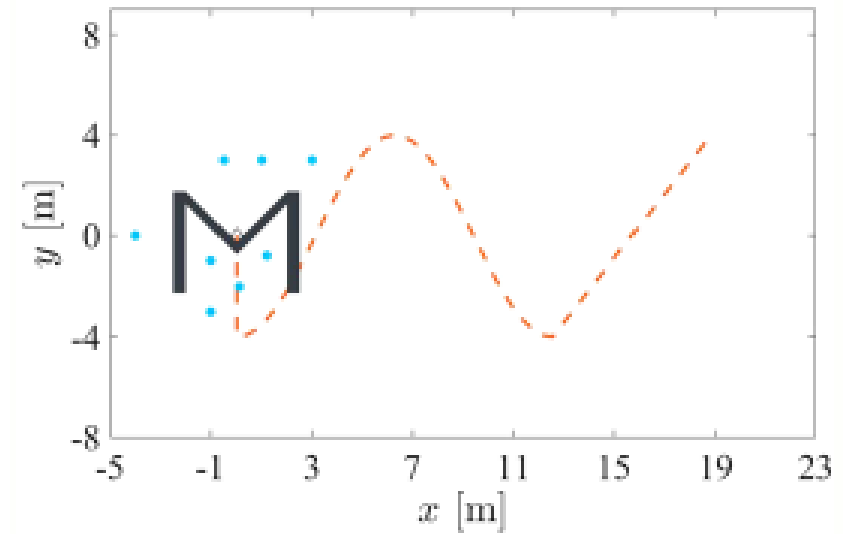
- Minimierung eines Kostenfunktional J in jedem Zeitschritt über einen endlichen Zeithorizont;
- Prädiktion des Systemzustands über diesen Zeithorizont mit Hilfe der Systemdynamik;
- Einhaltung von Nebenbedingungen während der Optimierung.

Nach Berechnung des (aktuell) optimalen Eingangs \mathbf{u} wird dieser für genau einen Zeitschritt auf das System angewandt. Danach wird neu iteriert.



Beispielaufgabe:

Transport eines Σ entlang
einer Trajektorie mit n - Robotern



Zerlegung des Problems in Teilaufgaben

Distributed Model Predictive Control

Bei der verteilten modellprädiktiven Regelung muss zunächst auch wieder ein Optimalsteuerungsproblem (z.B. Kostenfunktional und Nebenbedingungen) definiert werden. Zusätzlich gilt es das Problem verteilt (auf mehreren Rechnern) zu lösen.

In unserem Beispiel werden die Roboter als reine Integratoren angenommen:

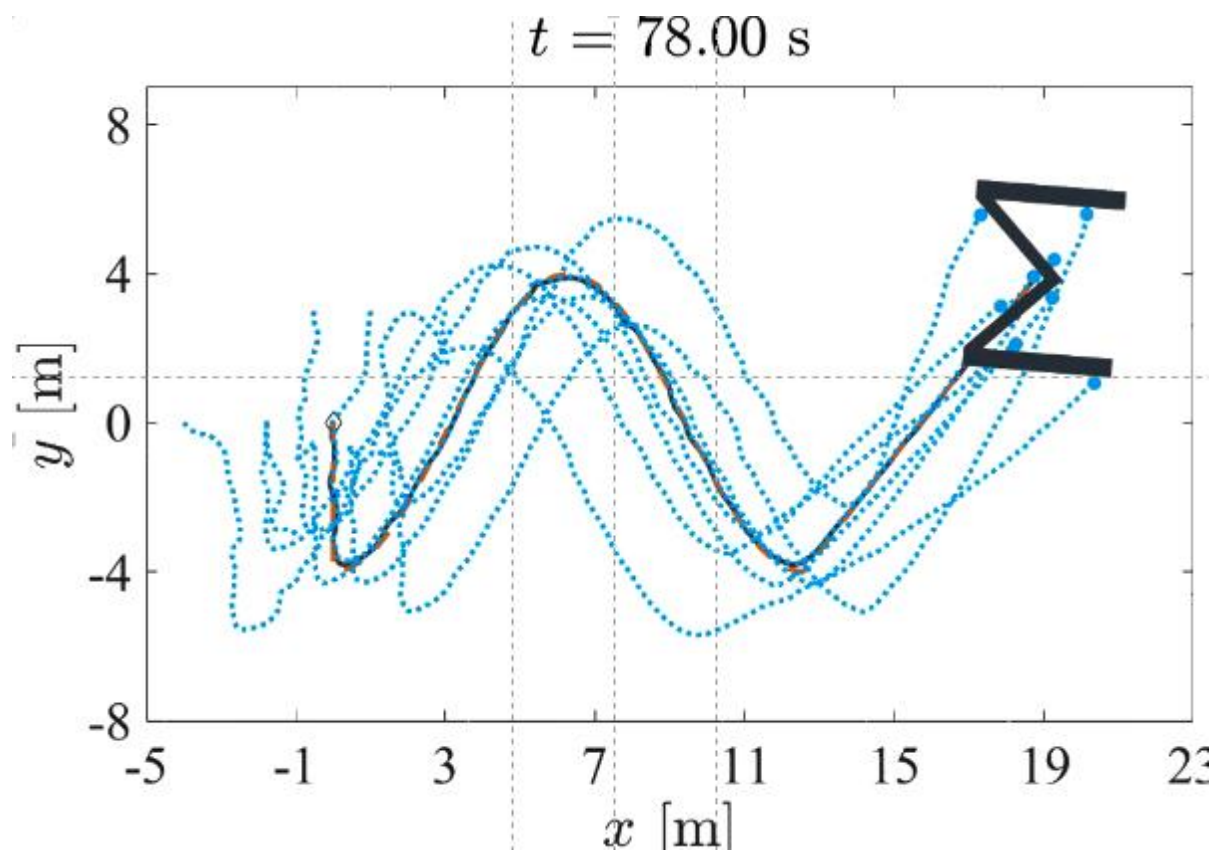
$$\dot{\mathbf{x}}_i := \mathbf{u}_i, \mathbf{x}_i \in \mathbb{R}^2, i \in \{1, \dots, N\}$$

Für eine erfolgreiche, kooperative Lösung des Problems ist Kommunikation zwischen den Robotern nötig

→ Jeder Roboter benötigt Information über den Zustand von relevanten anderen Robotern, sogenannter Nachbarn $\mathcal{N}_i \subset \{1, \dots, N\} \setminus \{i\}$

→ Für Roboter i kann eine kooperative Kostenfunktion aufgestellt werden:

$$J_i(\mathbf{x}_i(\cdot | t), \mathbf{u}_i(\cdot | t)) = \sum_{k=t}^{t+H-1} \left[\underbrace{\|\mathbf{x}_i(k|t) - \mathbf{r}_{ref}^i(k|t)\|_{\mathbf{Q}_t}^2}_{\text{Abweichung von eigener Referenztrajektorie}} + \underbrace{\sum_{j \in \mathcal{N}_i} \|\mathbf{x}_j(k|t) - \mathbf{x}_i(k|t) - \boldsymbol{\rho}_{i \rightarrow j}\|_{\mathbf{Q}_f}^2}_{\text{Abweichung von der erzielten Formation}} + \|\mathbf{u}_i(k|t)\|_{\mathbf{R}}^2 \right]$$





W. 1. 2. Trajektorienplanung

Pfadplanung und Navigation bieten uns eine geometrische Bewertung bzw. Beschreibung der Roboterbewegung. Um die dynamischen Effekte zu berücksichtigen, verwenden wir die Trajektorienplanung. Dies liefert uns als Antwort die notwendigen Geschwindigkeiten und Gelenkbeschleunigungen um dem Pfad zu folgen. Pfadplanungsalgorithmen geben uns nur eine Sequenz von Punkten. Sind nur die Endeffektor-Positionen bekannt, dann muss in einem ersten Schritt mit Hilfe der **inversen Kinematik** eine Sequenz von Gelenkkoordinaten berechnet werden.



Planung der Punkt-zu-Punkt Bewegung unter der Einhaltung von Nebenbedingungen an den Endpositionen

Vereinfachung:

Ab sofort ist $y(t)$ ein Skalar

Anfangsbedingungen: $y(t_0) = y_0$, $\dot{y}(t_0) = v_0$

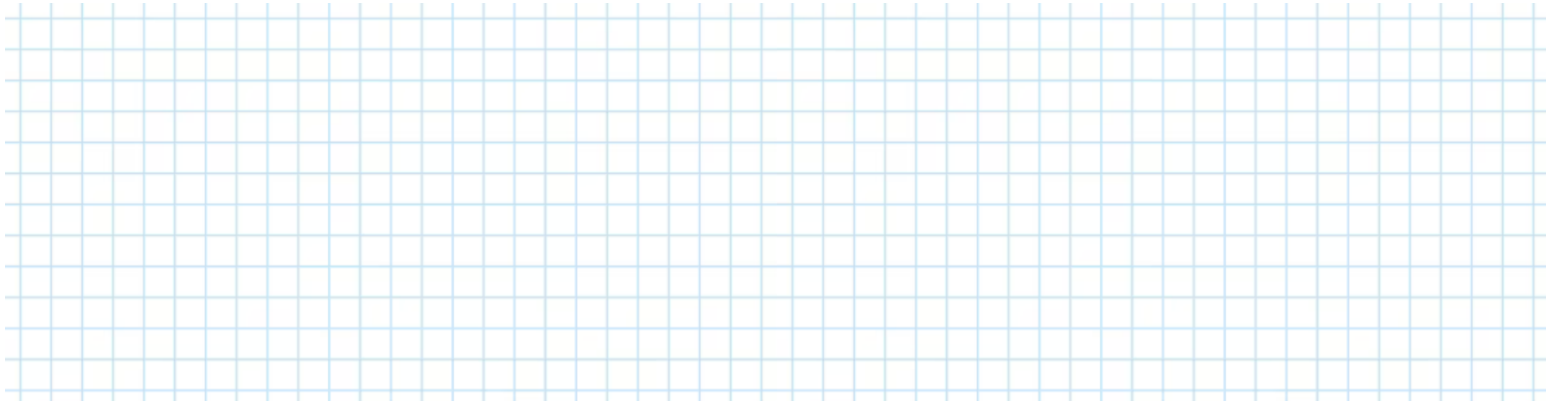
Endbedingungen: $y(t_f) = y_f$, $\dot{y}(t_f) = v_f$

Für die Planung des zeitlichen Verlaufs der Trajektorien werden Polynome bestimmter Ordnung verwendet.

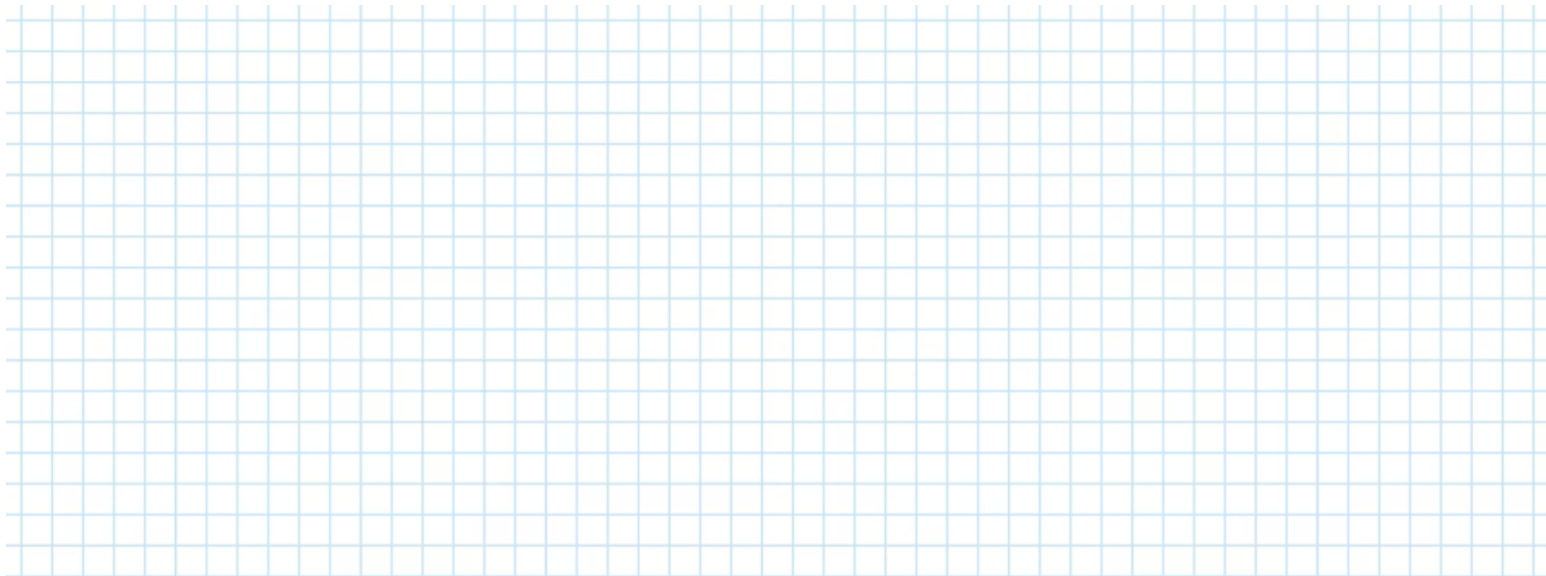
y_0 , v_0 , y_f , v_f sind gegeben \rightarrow Es wird ein Polynom mit 4 unabhängigen Parametern benötigt um alle Nebenbedingungen einzuhalten.



Polynom mit 4 unabhängigen Parametern (Polynom 3. Ordnung, kubischer Spline):



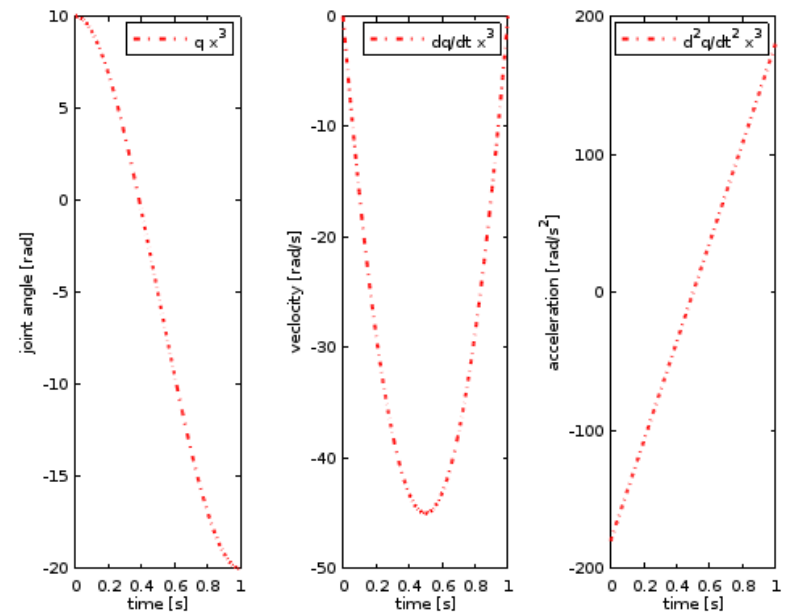
... in Matrizenschreibweise:





```
1 %% Modellierung und Simulation in der Mechatronik WS 2017 / 2018
2 % Homework trajectory planing
3 % Institut fuer Technische und Numerische Mechanik, Uni Stuttgart
4 % Profs Eberhard / Fehr / Hanss
5 % WS 2017 / 2018
6 % Jun. Prof. J. Fehr    Dipl.-Ing. C. Kleinbach
7 % last mod. 2015-12-16 Joerg Fehr
8 %
9 function Trajectoryplanning
10 close all
11 %% First example for trajectory planning
12 % Aim is a smooth transition of one link from
13 % starting position q0=10 [rad], v0=0 [rad/s], to
14 % final position qf=-20 [rad], v0=0 [rad/s],
15 t0=0;    % initial time [s]
16 q0=10;   % initial joint position in [rad]
17 v0=0;    % initial velocity in [rad/s]
18 tf=1;    % final time [s]
19 qf=-20;  % final joint position in [rad]
20 vf=0;    % final velocity in [rad/s]
21
22 a=cubic_func(t0,q0,v0,tf,qf,vf);
23 %% Question 1: Write a Matlab function for quintic sp
24 % 1.a) Write a function |quintic_func(t0,q0,v0,a0,tf,
25 %       returns the polynomial coefficient of the qui
26 %       which fullfils the boundary conditions q0,v0,a
27 %
28 % 1.b) Plot the joint angle, joint velocity and joint
29 % initial and final values given below.
30 %
31 % 1.c) Compare the joint angle, joint velocity and jo
32 %       with the cubic spline.
33 %
34
35 t0=0;    % initial time
36 q0=10;   % initial joint position in rad
37 v0=0;    % initial velocity in [rad/s]
38 a0=0;    % initial acceleration in [rad/s^2]
39 tf=1;    % final time
40 qf=-20;  % final joint position in [rad]
41 vf=0;    % final velocity in [rad/s]
42 af=0;    % final acceleration in [rad/s^2]
43 % cubic_func(t0,q0,v0,tf,qf,vf)
44 % quintic_func(t0,q0,v0,a0,tf,qf,vf,af)
45 % quintic_func(.....)
```

Weiterführende Informationen
siehe ILIAS



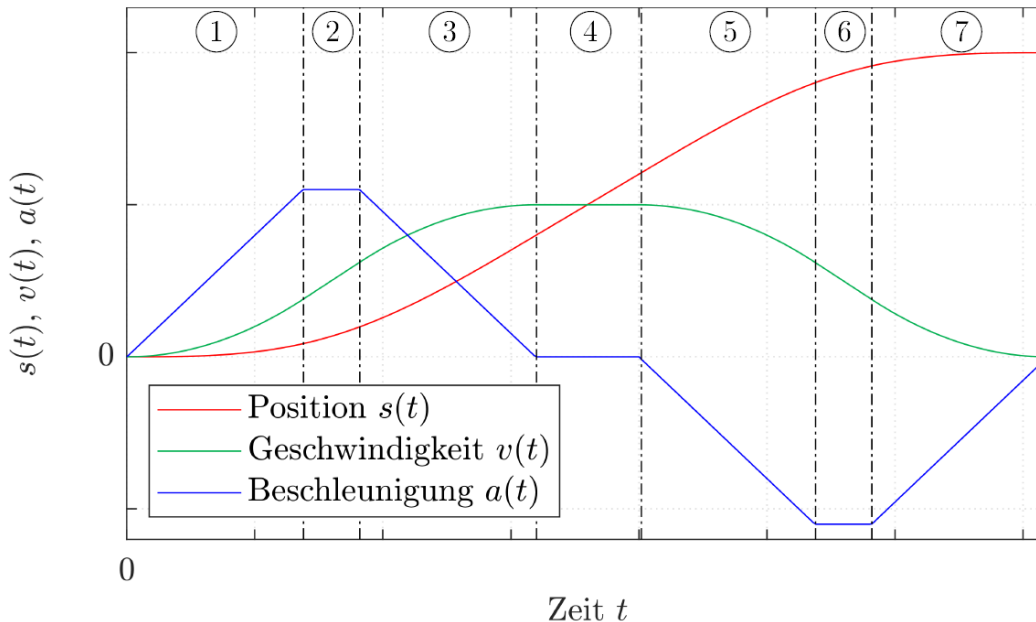


```
47 %%
48 %% Question 2: Write a Function which provides a Trajectory based on "Linear Segments with Parabolic Blends LSPB"
49 %
50 % A LSPB consists of three parts:
51 % * a cubic spline up to the blend time, to accelerate the joint with a constant acceleration up to a
52 % certain velocity,
53 % * a linear function where the joint moves with a constant velocity (v_desired),
54 % * a cubic spline, to decelerate the joint with a constant acceleration to the final state (position/velocity).
55 %
56 % <<LSPB.PNG>>
57 %
58 % 2.a) Write a LSPB function which calculates a LSPB spline
59 % which fullfills the boundary conditions q0,v0 at t0 and qf,vf which
60 % moves with a constant velocity v_c after the blend time.
61 %
62 % 2.b) Calculate the blend time for acceleration and deceleration based on the given parameter and the
63 % constant acceleration.
64 %
65 % 2.c) Make a qualitative sketch of the joint velocity and joint acceleration.
66 % Compare your qualitative sketch with the plot produced with Matlab.
67 %
68 % 2.d) How can you improve LSPB splines?
69 %
70 % 2.e) Make a proper documentation of your code if you want a review of your code hand in your code via ILIAS to get a feedback
71 %
72 % * t0: 'start time'
73 % * q0: 'postion at start time'
74 % * tf: 'final time'
75 % * qf: 'postion at final time'
76 % * V: 'constant velocity after blend time
77 t0=0; % initial time
78 q0=10; % initial joint position in [rad]
79 tf=1; % final time
80 qf=-20; % final joint position in [rad]
81 V=-34; % constant velocity [rad/s]
82 % [q,dq,ddq,tb,a1,a2]=LSPB_func(...)
83
```

7-Phasenregler

Um Vibrationen der Maschinenstruktur zu vermeiden, werden ruckbegrenzte Bewegungsprofile erzeugt. Durch einen stetigen Beschleunigungsverlauf werden so Knicke im Geschwindigkeitsverlauf vermieden und so hohe Belastungen oder eine Anregung der Maschinenstruktur begrenzt. Es gibt zwei etablierte Ansätze zur Erzeugung ruckbegrenzter Bewegungsprofile [Schröder07]

- durch Phasen konstanten Ruckes wird ein rampenförmiges Beschleunigungsprofile erzeugt (siehe unten)
- Änderung der Beschleunigung als Funktion in Form von $\sin^2(t)$



Phase	Merkmal
1	linear zunehmende Beschleunigung
2	linear zunehmende Geschwindigkeit
3	linear abnehmende Beschleunigung
4	konstante Geschwindigkeit
5	linear abnehmende Beschleunigung
6	linear abnehmende Geschwindigkeit
7	linear zunehmende Beschleunigung

[Beckhoff22]



W. 2. Regelung starrer Mehrkörpersysteme

Typische Beispiele geregelter starrer Mehrkörpersysteme sind Roboter und Werkzeugmaschinen. Im Allgemeinen entspricht hier die Anzahl der Stellgrößen den Freiheitsgraden des Systems

Bewegungsgleichungen

End-Effektor-Position

Sollbahn

System ist voll aktuiert

Die Reglerstruktur für starre MKS (im besonderen Roboter) besteht aus drei Teilen:



W. 2.1 Inverse Kinematik

Mit der inversen Kinematik werden für eine gegebene Sollbahn \mathbf{r}_d^{EF} des End-Effektors (Tool Center Point) die dazugehörigen Trajektorien \mathbf{y}_d (inkl. $\dot{\mathbf{y}}_d, \ddot{\mathbf{y}}_d$) der verallgemeinerten Koordinaten berechnet.

Die Bestimmung der inversen Kinematik kann auf unterschiedliche Weise erfolgen, siehe [SicilianoEtAl10], [SpongHutchinsonVidyasagar06]:

- Algebraisches Lösen der nichtlinearen Gleichung
- Numerisches Lösen der nichtlinearen Gleichung
- Differenzielle Kinematik

W. 2.2 Inverse Dynamik (Vorsteuerung, Feedforward Control)

Mit der inversen Dynamik wird für gegebene $\mathbf{y}_d, \dot{\mathbf{y}}_d, \ddot{\mathbf{y}}_d$ die dazugehörige Stellgröße berechnet. Dazu wird die Bewegungsgleichung algebraisch aufgelöst:

$$\mathbf{u}_d = \mathbf{B}^{-1}(\mathbf{y}_d) \cdot [\mathbf{M}(\mathbf{y}_d, t) \cdot \ddot{\mathbf{y}}_d + \mathbf{k}(\mathbf{y}_d, \dot{\mathbf{y}}_d, t) - \mathbf{f}_{allg}(\mathbf{y}_d, \dot{\mathbf{y}}_d, t)]$$

Sind die Anfangsbedingungen $\mathbf{y}_0 = \mathbf{y}(t_0)$, $\dot{\mathbf{y}}_0 = \dot{\mathbf{y}}(t_0)$ des MKS mit denen der Sollbahnen $\mathbf{y}_d(t_0)$, $\dot{\mathbf{y}}_d(t_0)$ identisch, die Modellparameter exakt bekannt und liegen keine externen Störungen vor, so folgt das MKS exakt den gewünschten Trajektorien \mathbf{y}_d bzw. \mathbf{r}_d^{EE} .

W 2.3 Rückführender Regler (Feedback Control)

Da in der Realität immer Unsicherheiten und Störungen auftreten ist ein Regler zwingend notwendig. Eine gute Vorsteuerung trägt den Hauptteil der „Regelungsarbeit“, der Regler muss dann nur kleine Fehler ausgleichen. Daher sind hier oft einfache Regler einsetzbar, z.B. PID-Regler, LQR (Linear Quadratic Regulator), ...

$$\text{PID-Regler: } \mathbf{u}_r = \mathbf{P} \cdot (\mathbf{y}_d - \mathbf{y}) + \mathbf{D} \cdot (\dot{\mathbf{y}}_d - \dot{\mathbf{y}}) + \mathbf{I} \cdot \int (\mathbf{y}_d - \mathbf{y}) dt$$

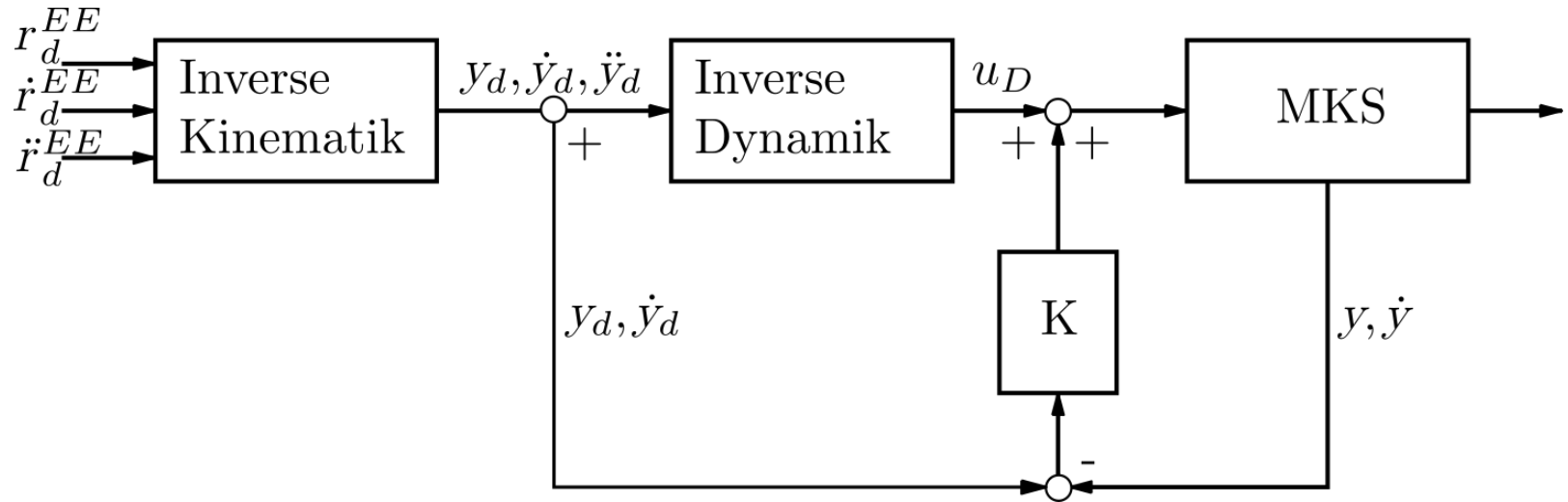
Hierbei sind $\mathbf{P}, \mathbf{D}, \mathbf{I} \in \mathbb{R}^{f \times f}$ Diagonalmatrizen mit dem Reglerverstärkerparameter auf der Diagonalen.

Die Vorsteuerung und der Regler können weitgehend unabhängig voneinander entworfen werden (Reglerstruktur mit Entwurfsfreiheitsgraden).



W 2.4 Reglerstruktur

Die Struktur aus den einzelnen Komponenten setzt sich wie folgt zusammen:





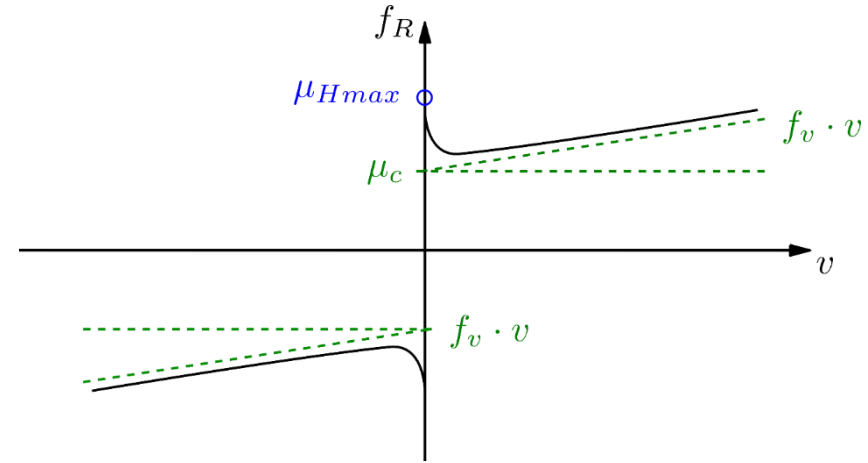
W 3 Reibung

Reibung tritt in mehreren Varianten auf: z.B. Haftreibung F_{RH} , Gleitreibung F_{GH} .
Eine bekannte Reibungskennlinie ist die Reibungskennlinie nach Stribeck (1902)
[StribeckSchröterAustin03]

$$F_R = f_r \cdot F_N \quad (2.1.12.1)$$

mit

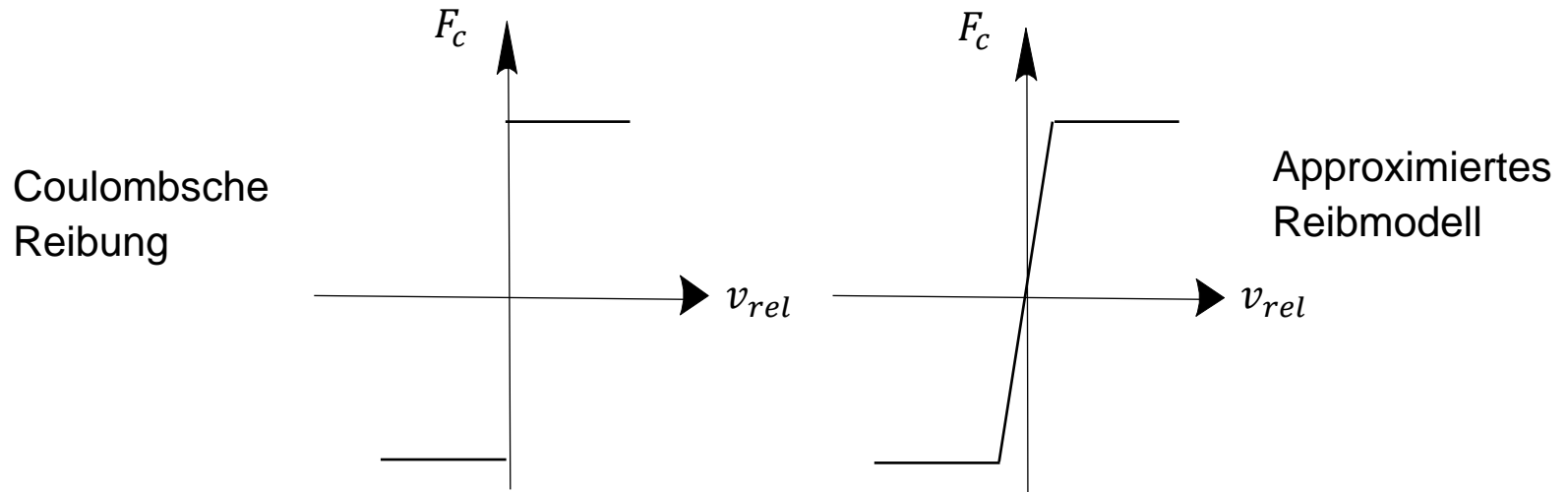
$$f_r = \mu_c \cdot \operatorname{sgn}(v) + f_v \cdot v + f_m \cdot e^{-c|v|} \cdot \operatorname{sgn}(v) \quad (2.1.12.2)$$



Für die Modellierung der Reibung im Simulationsmodell werden die Bereiche der Gleitreibung in denen für die relative Geschwindigkeit zwischen den Reibpartnern $v_{rel} \neq 0$ gilt, durch die sign-Funktion ausgedrückt

$$\operatorname{sgn}(v) = \begin{cases} 1 & \text{für } v_{rel} > 0 \\ -1 & \text{für } v_{rel} < 0 \end{cases}, \quad \operatorname{sgn}(0) \text{ nicht definiert}$$

Dies kann zu Bewegungsgleichungen mit Unstetigkeiten führen, siehe z.B. [Pietruszka16], welche wiederum numerisch Schwierigkeiten bereiten können. Aus diesem Grund entschärft man das Problem oft durch eine stückweise stetige Reibkennlinie (Sättigungsfunktion, sat).



Mit diesen Informationen und dem Inhalt der bisherigen Vorlesungen sind Sie nun in der Lage, die folgende exemplarische Simulationssaufgabe zu bearbeiten.

Quellen

[Pietruszka16] Pietruszka, W. D.: MATLAB® und Simulink® in der Ingenieurpraxis - Modellbildung, Berechnung und Simulation, Springer Vieweg, Wiesbaden, 4. Aufl. 2014.

[SicilianoEtAl10] Siciliano, B.; Sciavicco, L.; Villani, L.; Oriolo, G.: Robotics: Modelling, Planning and Control. London: Springer, 2010.

[SpongHutchinsonVidyasagar06] Spong, M.W.; Hutchinson, S.; Vidyasagar, M.: Robot Modeling and Control. New York: Wiley, 2006.

[Beckhoff22] Beckhoff: Ruckbegrenztes Standard-Bewegungsprofil, 2022.
https://infosys.beckhoff.com/index.php?content=../content/1031/tf5200_dynamic_param/248574987.html&id=

[Schröder07] Schröder, T.: Entwicklung und Evaluation von Algorithmen zur zeitoptimierten Bewegungszerlegung bei kinematisch redundanten Werkzeugmaschinen. Doktorarbeit, Technische Universität Chemnitz, 2007.

[FerramoscaEtAl13] Ferramosca, A.; Limon, D.; Alvarado, I.; Camacho, E.: Cooperative Distributed MPC for Tracking. Automatica, Vol. 49, No. 4, pp. 906-914, 2013.

[StewartEtAl10] Stewart, B.T.; Venkat, A.N.; Rawlings, J.B.; Wright, S.J.; Pannocchia, G.: Cooperative Distributed Model Predictive Control. Systems & Control Letters, Vol. 59, No. 8, pp. 460-469, 2010.