

Differentielle Dynamische Programmierung in der Trajektorienstabilisierung

Arne-Christoph Hildebrandt, Gerald Witt, Mohit Kumar und Sven Kraus*

Zusammenfassung: Autonomes Fahren in immer komplexeren Umgebungen stellt nicht nur an die Bewegungsplanung, sondern auch an die Trajektorienstabilisierung immer höhere Anforderungen. Es geht nicht mehr nur darum eine Querablage zu einer Referenzbahn und eine longitudinale Abweichung von einer Geschwindigkeitsvorgabe zu minimieren, sondern es muss die Gesamtabweichung des Fahrzeugzustands von einer idealen Vorgabe betrachtet werden. In diesem Beitrag stellen wir ein Optimierungsverfahren, basierend auf Differentieller Dynamischer Programmierung, vor. Das Verfahren optimiert eine initiale Stellgrößenfolge, um ein exaktes Folgen einer Referenztrajektorie zu ermöglichen. Die Optimierung basiert auf einer Vorhersage der zukünftigen Regelabweichungen. Dazu können generisch verschiedene Fahrzeugmodelle verwendet werden. Anstatt das Optimierungsproblem als ein Gesamtproblem zu betrachten, teilt das vorgestellte Verfahren das Optimierungsproblem auf und löst es iterativ. Die Optimierung selbst kann jederzeit mit einer suboptimalen Lösung abgebrochen werden, was es besonders für Echtzeit-Anwendungen attraktiv macht.

Schlüsselwörter: Trajektorienoptimierung, Trajektorienstabilisierung, Fahrzeugmodellierung, Differentielle Dynamische Programmierung

1 Einleitung

Die Technik autonom fahrender Fahrzeuge wird immer greifbarer. Neben Funktionen, welche den Fahrer unterstützen, sind bereits Systeme kommerziell erhältlich, die, unter gewissen Limitierungen, das Steuern des Fahrzeugs vollständig übernehmen. Zum jetzigen Zeitpunkt werden diese Systeme vor allem für Landstraßen und Autobahnen angeboten. Vor diesem Hintergrund fängt die Forschung in Hochschulen und Industrie an, sich auf das autonome Fahren in urbanen Umgebungen zu konzentrieren.

Im Vergleich zu Schnellstraßen stellen urbane Umgebungen die Entwicklung autonom fahrender Fahrzeuge vor neue Herausforderungen. Anders als auf Schnellstraßen müssen Fahrzeuge nicht nur in der Lage sein, Fahrspuren zu wechseln oder ihnen zu folgen, sondern sie müssen auf deutlich komplexere Verkehrssituationen reagieren können.

Um auch auf komplexe Situationen noch in Echtzeit reagieren zu können, basiert die Bewegungsplanung autonomer Fahrzeuge meist auf einem hierarchischen Ansatz. Es werden mit abstrahierten Modellen des Fahrzeugs unter Berücksichtigung der Umgebung Trajektorien generiert. Diese Referenztrajektorien werden von Reglern als Vorgabe verwendet. Dabei wird häufig eine entkoppelte Regelung der Längs- und Querdynamik genutzt. [5,

*MAN Truck & Bus AG, Engineering Central Research (e-mail: arne-christoph.hildebrandt@man.eu).

2, 8, 4] Die entkoppelte Längs- und Querdynamik des Reglers vernachlässigt Kopplungseffekte der Gesamtdynamik bzw. des Gesamtfahrverhaltens des Fahrzeugs. Gerade beim urbanen Fahren ist es aufgrund von starkem Verkehrsaufkommen und schwer zu präzisierenden Situationen unabdingbar, dass das Fahrzeug nicht nur entkoppelt relativ zu einer Bahn geregelt und ein Geschwindigkeitsprofil verfolgt wird, sondern dass sich das Fahrzeug zu einem bestimmten Zeitpunkt exakt in allen seinen Zuständen auf der gewünschten Trajektorie befindet.

In diesem Beitrag versuchen wir uns einer Antwort auf folgende Fragestellung zu nähern: Wie kann, basierend auf einer ideal geplanten Referenztrajektorie, eine optimale Folge an Stellgrößen bestimmt werden, die generisch komplexe Modelle von unterschiedlichen Fahrzeugen berücksichtigt und die Abweichungen des realen Fahrzeugs zur Referenztrajektorie minimiert?

Im Folgenden werden wir ein Verfahren präsentieren, welches basierend auf [7], es ermöglicht, optimale Stellgrößen zu einer gegebenen Referenztrajektorie im Sinne einer Steuerung zu bestimmen. Das Verfahren ermöglicht, generisch Fahrzeugmodelle zu verwenden, welche Längs- und Querdynamik berücksichtigen. Des Weiteren wird das Verfahren erweitert um es als Regelung zur Trajektorienstabilisierung nutzen zu können.

2 Methodik

Wir gehen davon aus, dass als Input der hier vorgestellten Methode Trajektorien genutzt werden können, die von einer hierarchisch höheren Recheneinheit, der Bewegungsplanung [5], geplant wurden. Wir gehen davon aus, dass diese die aktuelle Verkehrssituation berücksichtigen und ein kollisionsfreies Fahren ermöglichen (vergleiche [4]). Die hier vorgestellte Methode soll die Stellgröße u , welche sich aus Lenkwinkel δ und Beschleunigung acc zusammensetzt, bestimmen, um ein möglichst exaktes Einhalten der Referenztrajektorien zu ermöglichen.

Modellierung und Kosten Die Basis unseres Algorithmus ist eine Beschreibung des Systems als dynamisches Modell

$$z_{i+1} = f(z_i, u_i) \quad (1)$$

mit dem Zustandsvektor z und der Stellgröße u . Hier ist wichtig anzumerken, dass unsere Systembeschreibung nicht auf LTI-Systeme beschränkt ist, sondern auch generisch nicht-lineare und zeit-variable Effekte abbilden kann. Die Folgen $Z = [z_0, \dots, z_i]$ und $U = [u_0, \dots, u_i]$ bilden die Zustands- und Stellgrößenfolgen über den Zeithorizont t_0, \dots, t_i ab. Die Abweichung des Fahrzeugs von den Referenztrajektorien wird über die Abweichung der Zustände zu den korrespondierenden Werten der Referenztrajektorie z_{ref} als Kosten dargestellt.

Die Kosten für Zustandsabweichungen

$$l(z) = l(z, z_{ref})$$

zusammen mit den Kosten für Stellgrößen $l(u)$ definieren die laufenden Kosten

$$l(z, u) = l(z) + l(u).$$

Über die konkrete Kostendefinition kann das Fahrzeugverhalten beeinflusst werden. Die Gesamtkosten J setzen sich aus der Summe der laufenden Kosten l und den Kosten für Abweichungen des Endzustands l_f zusammen. Der initiale Zustand z_0 ist gegeben und die Zustände $z_i \in X$ für $i > 0$ werden über die Integration von (1) über den Zeithorizont $i = 0, \dots, N$ berechnet. Für die Gesamtkosten ergibt sich

$$J(z_0, U) = \sum_{i=0}^{N-1} l(z_i, u_i) + l_f(x_N). \quad (2)$$

Optimierungsproblem Das Optimierungsproblem zur Bestimmung einer optimalen Folge an Stellgrößen wird durch die Minimierung der Kostenfunktion J gelöst:

$$U^* = \arg \min_u J(z_0, U). \quad (3)$$

Das hier verwendete Verfahren zur Lösung dieses Problems basiert auf der *Dynamischen Programmierung* [1] und ihrer Erweiterung zur *Differentiellen Dynamischen Programmierung* (DDP) [7]. Der DDP Algorithmus, detailliert dargestellt in [3, 6, 7], berechnet ausgehend von einem initialem Zustand z_0 und einer nominalen Stellgrößenfolge U_0 iterativ in einem Vorwärtslauf eine nominale Zustandstrajektorie. Anschließend werden im Rückwärtslauf die Kosten (2) approximiert und lokale Verbesserungen berechnet. In der nächsten Iteration werden diese im Vorwärtslauf angebracht. Demnach wird iterativ die Stellgrößenfolge verbessert bis sie zu einem Optimum konvergiert. Algorithmus 1 fasst das Verfahren zusammen.

Algorithm 1: Zusammenfassung des Ablaufs des DDP Algorithmus.

```

N ← Länge der Zustandstrajektorie
U ← Initiale Stellgrößenfolge U0
j ← 0
Initiale Vorwärtsintegration von (1) mit U
% Optimierungsschleife
while Optimum noch nicht erreicht do
  % Gradientenberechnung
  for  $i=N-1:0$  do
    | Berechnung Kosten  $J_j$  und ihrer Ableitungen
    | Berechnung lokale Verbesserungen der Stellgrößenfolge  $\Delta U$ 
  % Liniensuche
  while  $(J_j - J_{j-1}) < \Delta\epsilon$  do
    |  $U \leftarrow U + \alpha\Delta U$ 
    | Vorwärtsintegration von (1) mit  $U$ ,  $z_0$  und Berechnung  $J_j$ 
    |  $\alpha \leftarrow \alpha + \Delta\alpha$ 
   $j \leftarrow j + 1$ 

```

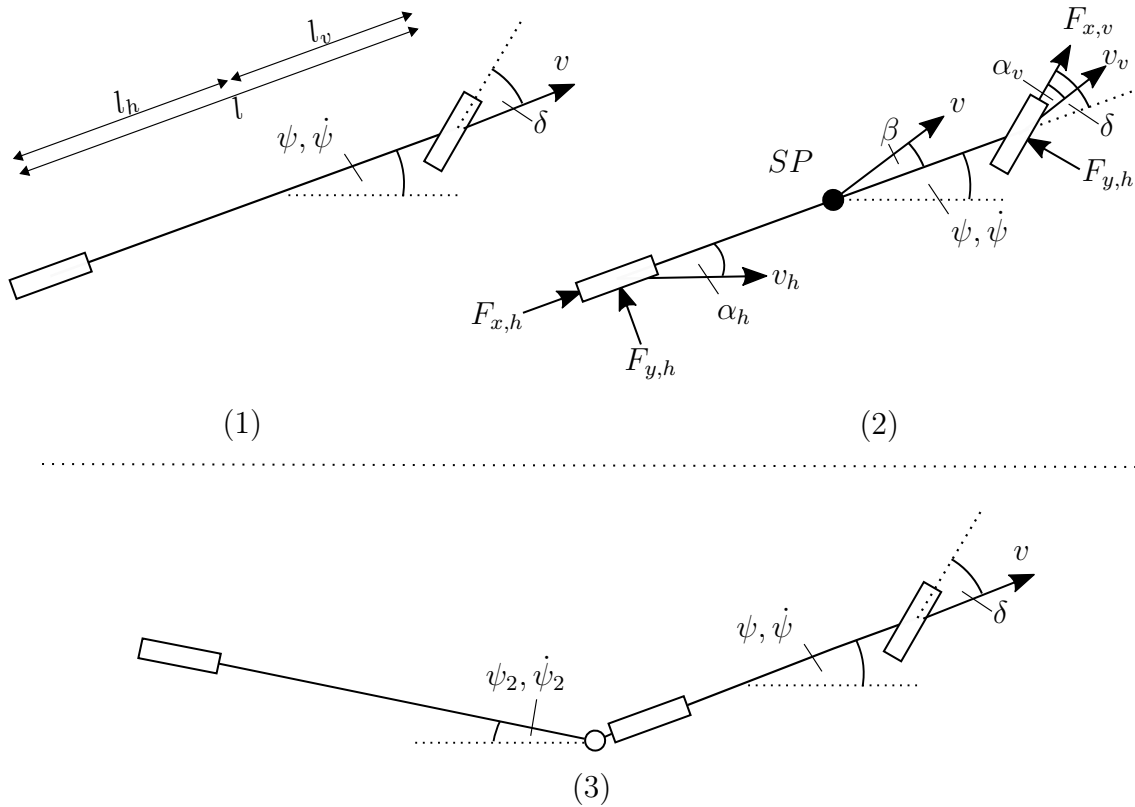
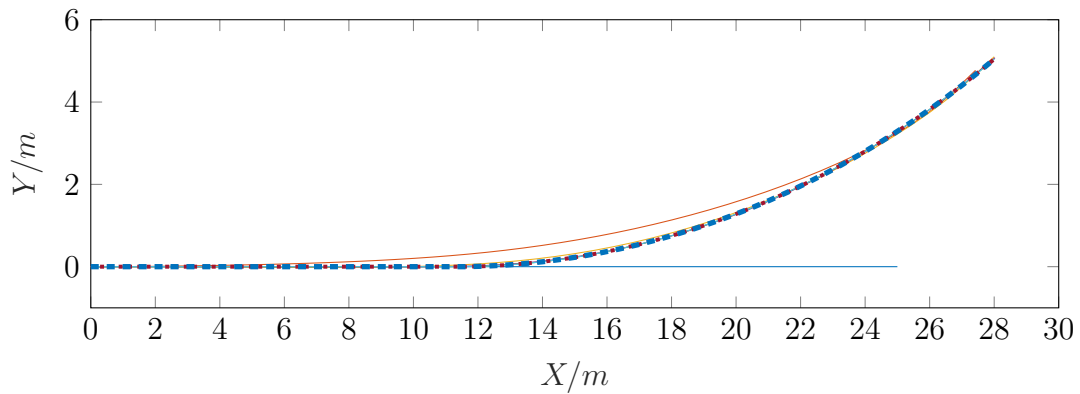


Abbildung 1: Kinematisches und dynamisches Einspurmodell, sowie das kinematische Einspurmodell erweitert um einen Anhänger. Unterschiedliche Modelle, welche hier zur Optimierung verwendet werden.

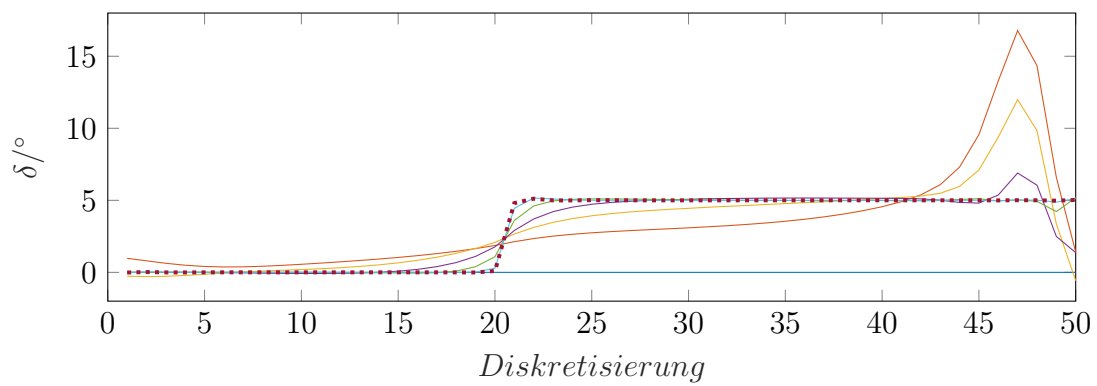
3 Anwendungen

Die großen Vorteile der vorgestellten Methode liegen darin, dass (a) beliebige Modelle in (1) verwendet werden können. (b) Zudem ermöglichen die iterativen Verbesserungen ein Abbrechen der Optimierung nach jeder Iteration. Wir haben den ursprünglich in [1] vorgestellten Algorithmus erweitert um beide Eigenschaften auszunutzen. Im Folgenden geben wir einen Einblick in unsere Erweiterungen und diskutieren mehrere Anwendungsbeispiele.

Optimierung In Abbildung 1 werden schematisch verschiedene Fahrzeugmodelle aufgezeigt, die wir für unterschiedliche Anwendungsfälle untersucht haben. In einem ersten Beispiel versuchen wir die Konvergenzeigenschaften des Optimierungsalgorithmus aufzuzeigen. In Abbildung 2 wird exemplarisch das Ergebnis des DDP Algorithmus für die Position des Fahrzeugmodells und des Lenkwinkels als eine Stellgröße der Fahrzeugbewegung dargestellt. Als Input verwenden wir eine ideale Referenztrajektorie. Der resultierende Pfad wird in Abbildung 2 (b) dargestellt. Die Geschwindigkeitverläufe haben wir aus Platzgründen vernachlässigt. Für den initialen Stellgrößenverlauf U_0 wird hier kein Vorwissen verwendet. Der Verlauf wird in Abbildung 2 dargestellt. Die Abbildung 2 zeigt sehr deutlich, dass selbst bei einer initial sehr schlechten Stellgrößenvorgabe ($U_0 = [0, \dots, 0]$) sich das Fahrzeug schon nach wenigen Iterationen stark der Referenztrajektorie annähert.



(a) Zustände



(b) Lenkwinkel

Iterationen: — 1, — 2, — 3, — 4, — 5, — 6, ···· 7, - - - Referenzzustände

Abbildung 2: Exemplarischer Verlauf der Lenkwinkelstellgröße und der Position des Fahrzeugs über die Iterationen des DDP Algorithmus für ein kinematisches Einspurmodell. Als initiale Stellgröße wurde 0 für den Lenkwinkel angenommen.

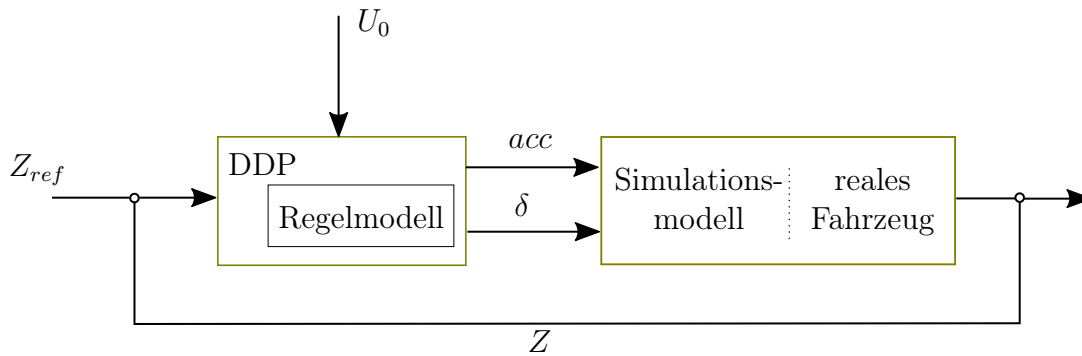


Abbildung 3: Schematische Abbildung des Aufbaus zur Untersuchung des Verfahrens und seiner Anwendung im realen Fahrzeug.

Trajektorienstabilisierung Ein weiterer Vorteil, das Optimierungsproblem nach dem Vorbild des DDP Algorithmus zu lösen, liegt darin, dass Teilprobleme rekursiv gelöst werden können und dass die Lösung des Gesamtproblems iterativ verbessert wird. Dadurch ist es möglich, einerseits viele Rechenschritte zu parallelisieren, was die Performance des Algorithmus deutlich verbessert und andererseits kann prinzipiell nach jedem Iterationsschritt die Optimierung abgebrochen werden. Damit liegt zwar keine vollständig konvergierte Lösung vor, aber eine suboptimale Lösung, welche die Initiallösung im Sinne der Kostenfunktion verbessert und in den gegebenen Zeitlimitierungen berechnet wurde. Bedingt durch diese Eigenschaften verwenden wir das vorgestellte DDP Verfahren wie folgt zur Trajektorienstabilisierung:

Zu jedem Regelzyklus wird der Algorithmus, wie in Algorithmus 1 beschrieben, aufgerufen. Der Zustand z_0 wird mit dem aktuellen Zustand des realen Fahrzeugs initialisiert. Daraufhin wird das Optimierungsproblem mit der aktuell vorliegenden Referenztrajektorie Z_{ref} als Input gelöst (vergleiche Abbildung 3). Je nach Zeithorizont der vorliegenden Referenztrajektorie kann der DDP Algorithmus als *moving horizon* Ansatz implementiert und gelöst werden. Dies bedeutet auch, dass das Optimierungsergebnis des vorherigen Regelzyklus als Initiallösung weiter verwendet werden kann, was die Konvergenz des Algorithmus nach einer Anfangsphase deutlich verbessert. Abbildung 4 und 5 zeigen die Ergebnisse für den identischen Referenzinput, wie er auch in dem vorherigen Beispiel verwendet wurde, aber mit einem dynamischen Einspurmodell als Regelmodell. Im Gegensatz zu dem vorherigen Beispiel wird nicht über den gesamten Zeithorizont optimiert, sondern der Zeithorizont im Sinne eines *moving horizon* auf $N = 25$ beschränkt. Zudem wird ein dynamisches Einspurmodell als Simulationsmodell verwendet. Um die Rechenzeit für die Anwendung im realen Fahrzeug zu reduzieren, haben wir in jedem Regelzyklus die maximale Anzahl der Iterationen auf eine beschränkt. Man kann deutlich erkennen, dass die Kostenreduzierung der Optimierung durch die Weiterverwendung der Optimierungsergebnisse mit voranschreitender Zeit deutlich abnimmt - womit sich die Lösung auch schon nach wenigen Iterationen nah an einem Optimum befindet.

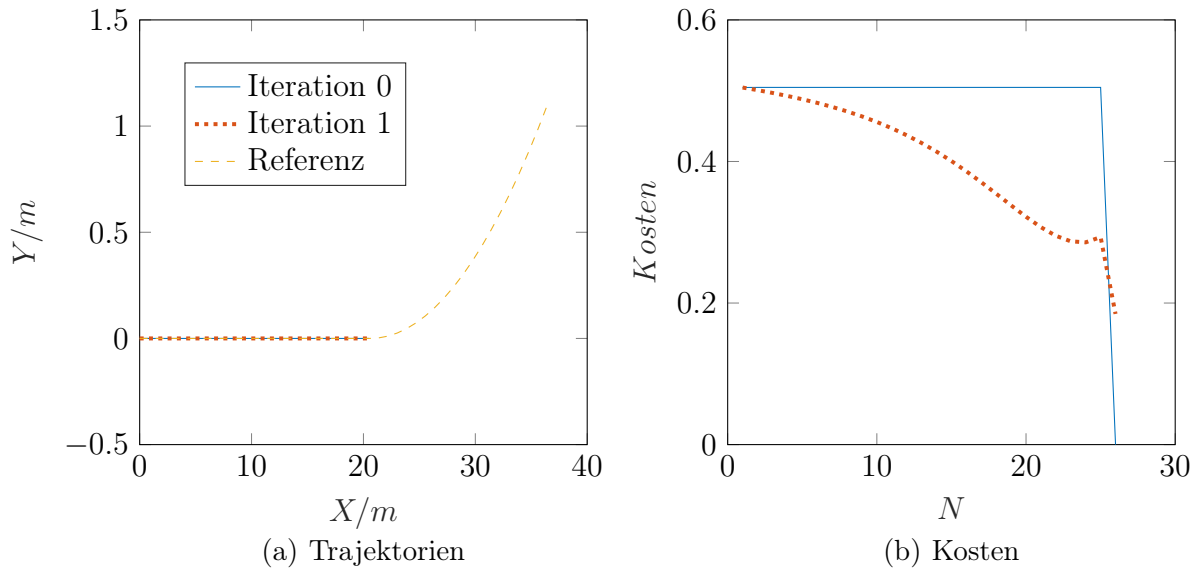


Abbildung 4: *moving horizon* Implementierung mit Zeithorizont $N = 25$. Erster Regelzyklus mit dynamischen Einspurmodell als verwendetes Regelmodell.

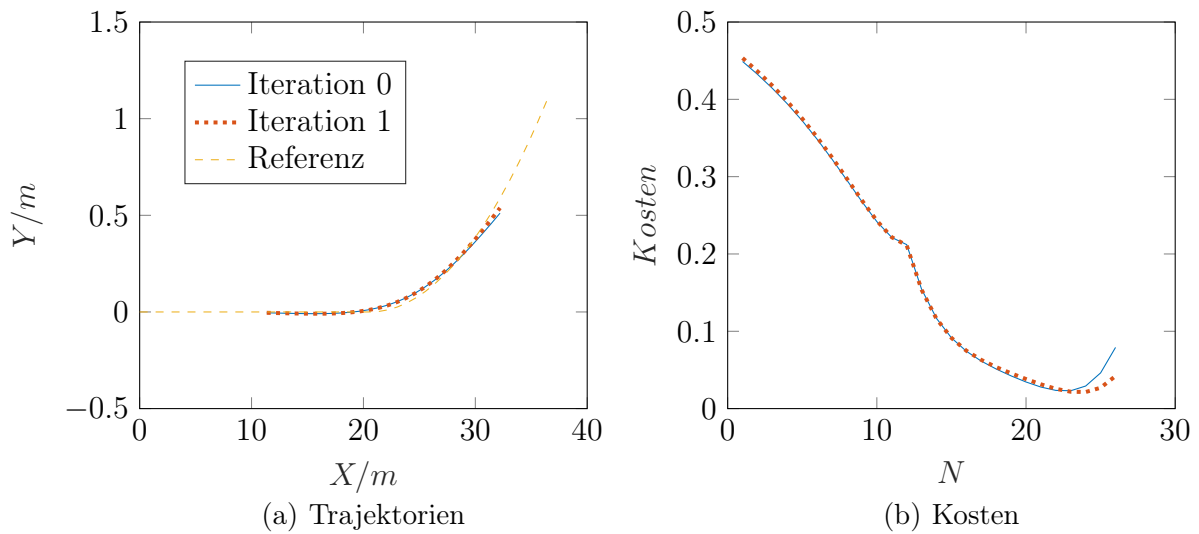


Abbildung 5: *moving horizon* Implementierung mit Zeithorizont $N = 25$. 15. Regelzyklus mit dynamischen Einspurmodell als verwendetes Regelmodell.

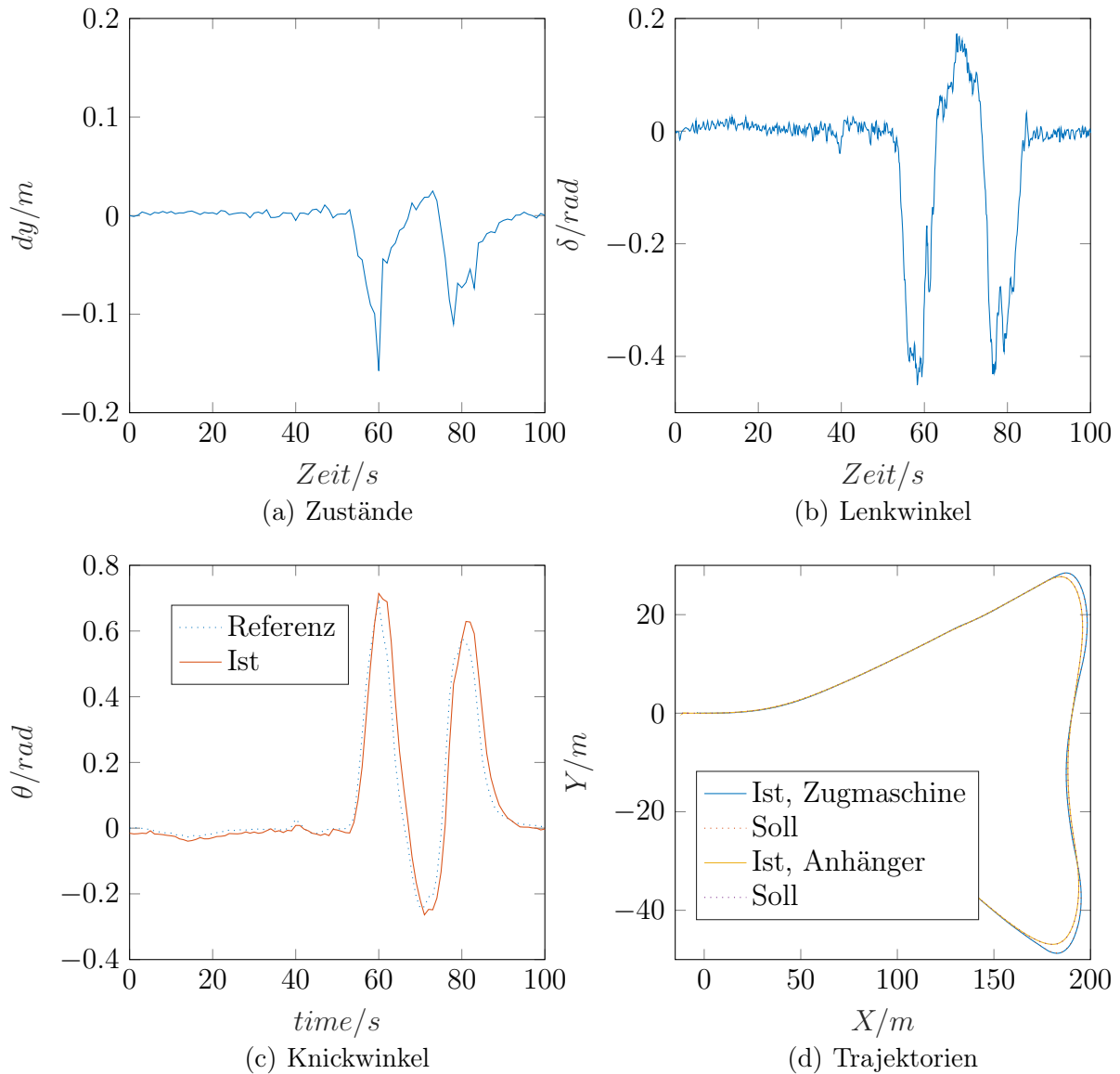


Abbildung 6: Untersuchung der *moving horizon* Implementierung für ein kinematisches Einspurmodell mit Anhänger in einer Mehrkörpersimulationsumgebung. Referenzdaten sind mit echtem Fahrzeug aufgenommen worden.

Zusätzlich haben wir das vorgestellte Verfahren in einem typischen Rangiervorgang untersucht. Der LKW fährt mit Geschwindigkeiten unter $10 \frac{\text{km}}{\text{h}}$, in unserem Fall vorwärts, einen komplexen Pfad entlang. Der in Abbildung 6 (d) dargestellte Pfad wurde mit einem realen Fahrzeug aufgenommen. Für dieses Szenario haben wir das Regelmodell des kinematischen Einspurmodells um einen Anhänger erweitert. Im Gegensatz zu den vorherigen Untersuchungen verwenden wir zudem als Simulationsmodell ein erweitertes Mehrkörpersimulationsmodell aus einer kommerziellen Software und beschränken die Stellgröße auf die Größe δ . Den Zustand Z erweitern wir um den Knickwinkel θ zwischen Zugmaschine und Anhänger. Abbildung 6 stellt neben den Referenzbahnen des Anhängers und der Zugmaschine den Regelfehler des Knickwinkels, den berechneten Lenkwinkelverlauf und die Querablage dar.

4 Zusammenfassung und Ausblick

In diesem Beitrag haben wir einen Algorithmus vorgestellt, mit dem ideale Referenztrajektorien im Sinne einer definierten Kostenfunktion optimal verfolgt werden können. Die Vorteile des vorgestellten Algorithmus liegen darin, dass (1) der Algorithmus nicht auf einer zeit-invarianten und linearisierten Modellierung des Fahrzeugs basiert. Stattdessen können generisch verschieden komplexe Modellierungsstufen genutzt werden. (2) Anders als oft in der Literatur beschrieben [2, 4], behandelt der Algorithmus Längs- und Querdynamik nicht separat, sondern kann auch Kopplungseigenschaften beider Stellgrößen berücksichtigen. (3) Zudem wird das Optimierungsproblem nicht als Gesamtproblem gelöst, sondern in Teilprobleme aufgeteilt und iterativ gelöst. Dadurch ist es gerade für Echtzeitanforderungen höchst geeignet.

Wir haben diese Eigenschaften exemplarisch in drei Anwendungen aufgezeigt. In zukünftigen Arbeiten werden wir die vorgestellte Methode in echten Fahrzeugen und auf seine Anwendbarkeit für unterschiedliche Anwendungen hin untersuchen.

Literatur

- [1] Richard Bellman. *Dynamic Programming*. 1. Aufl. Princeton, NJ, USA: Princeton University Press, 1957.
- [2] Benjamin Gutjahr, Lutz Gröll und Moritz Werling. “Lateral vehicle trajectory optimization using constrained linear time-varying MPC”. In: *IEEE Transactions on Intelligent Transportation Systems* 18.6 (2017), S. 1586–1595.
- [3] D.H. Jacobson und D.Q. Mayne. *Differential dynamic programming*. Modern analytic and computational methods in science and mathematics. American Elsevier Pub. Co., 1970.
- [4] Pedro F. Lima u. a. “Spatial Model Predictive Control for Smooth and Accurate Steering of an Autonomous Truck”. In: *IEEE Transactions on Intelligent Transportation Systems* 2.4 (2017), S. 238–250.
- [5] Brian Paden u. a. *A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles*. arXiv:1604.07446v1, 2016.
- [6] Yuval Tassa, Tom Erez und Emanuel Todorov. “Synthesis and stabilization of complex behaviors through online trajectory optimization”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012, S. 4906–4913.
- [7] Yuval Tassa, Nicolas Mansard und Emo Todorov. “Control-limited differential dynamic programming”. In: *IEEE International Conference on Robotics and Automation*. 2014, S. 1168–1175.
- [8] Moritz Werling. “Ein neues Konzept für die Trajektoriengenerierung und-stabilisierung in zeitkritischen Verkehrsszenarien”. In: *at-Automatisierungstechnik Methoden und Anwendungen der Steuerungs-, Regelungs- und Informationstechnik* 60.1 (2012), S. 53–54.