

Technische Universität Dresden

Fakultät Elektrotechnik und Informationstechnik

Institut für Regelungs- und Steuerungstheorie

Studienarbeit

Verfahrensvergleich zur Trajektorienplanung für dynamische Systeme

vorgelegt von: Patrick Rüdiger
geboren am: 20. Juli 1994 in Schmölln

Betreuer:	Dr.-Ing. C. Knoll
Verantwortlicher Hochschullehrer:	Prof. Dr.-Ing. habil. Dipl.-Math. K. Röbenack
Tag der Einreichung:	13. August 2018

Aufgabenstellung zur Studienarbeit
für Herrn Patrick Rüdiger

Verfahrensvergleich zur Trajektorienplanung für dynamische Systeme

Unter Trajektorienplanung bzw. Steuerungsentwurf versteht man die Berechnung eines Eingangssignals $\mathbf{u} : \mathbb{R} \rightarrow \mathbb{R}^m$ für ein durch $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ gegebenes dynamisches System, welches (ggf. unter Beachtung von Beschränkungen) einen gegebenen Anfangs-Zustand in einen gewünschten End-Zustand überführt. Aus mathematischer Sicht kann man diese Aufgabe als Randwertproblem mit freien Parametern auffassen. Für diese Problemklasse existieren verschieden Lösungsmöglichkeiten, z. B.:

- Kollokationsverfahren
- (Mehrfach)-Schießverfahren
- Dynamische Programmierung

Ziel der Arbeit ist es, in einem systematischen Vergleich die Leistungsfähigkeit der unterschiedlichen Verfahren theoretisch und anhand geeigneter Beispiele zu untersuchen. Ausschlaggebend sind dabei Zuverlässigkeit der Lösungsfindung, Rechenzeit und Genauigkeit.

Für die Untersuchungen kann auf die am Institut entwickelte Software *pytrajectory* zurückgegriffen werden.

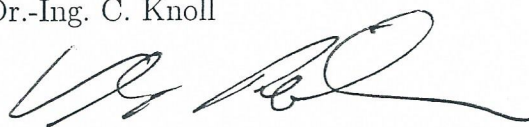
Im Einzelnen ergeben sich folgende Teilaufgaben:

- Literaturrecherche zur Trajektorienplanung
- Theoretische Analyse ausgewählter Verfahren (jeweilige Grundidee, Umsetzungsvarianten, Verfahrenseigenschaften)
- Charakterisierung des „Schwierigkeitsgrades“ von konkreten Überführungsaufgaben
- Umsetzung von unterschiedlichen Beispiel-Problemen mit allen Verfahren
- Dokumentation der Ergebnisse

Bearbeitungszeitraum: 30. 04. 2018 – 31. 07. 2018

Prüfer : Prof. Dr.-Ing. habil. Dipl.-Math. K. Röbenack,
Dr.-Ing. J. Winkler

Betreuer: Dr.-Ing. C. Knoll



Dresden, den 26.04.2018

Prof. Dr.-Ing. habil. Dipl.-Math. K. Röbenack
Verantwortlicher Hochschullehrer

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die von mir am heutigen Tage an der Fakultät Elektrotechnik und Informationstechnik eingereichte Studienarbeit zum Thema

Verfahrensvergleich zur Trajektorienplanung für dynamische Systeme

selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Schriften entnommen sind, wurden als solche kenntlich gemacht.

Bei der Auswahl und Auswertung des Materials sowie bei der Herstellung des Manuskripts habe ich Unterstützungsleistungen in Form von inhaltsbezogenen Fachdiskussionen und Korrekturvorschlägen zum Text von folgenden Personen erhalten:

Dr.-Ing. C. Knoll

Weitere Personen waren an der geistigen Herstellung der vorliegenden Arbeit nicht beteiligt.

Dresden, 13. August 2018

Patrick Rüdiger

Kurzfassung

Die vorliegende Arbeit befasst sich mit dem Vergleich von Verfahren zur Trajektorienplanung für dynamische Systeme. Aus theoretischer Sicht wird insbesondere auf die zugrundeliegenden numerischen Verfahren und die Formulierung von Problemen der Trajektorienplanung als Optimalsteuerungsproblem eingegangen, welche mit direkten und indirekten Verfahren den Zugang zu zwei unterschiedlichen Lösungsansätzen eröffnet. Im Rahmen dieser Arbeit wurden die mit Blick auf die Literatur besonders bedeutsamen direkten und indirekten Schieß- und Kollokationsverfahren implementiert. Die Implementierung erfolgte mithilfe des Software-Frameworks CasADi. Alle implementierten Verfahren nutzen mit IPOPT ein Innere-Punkte-Verfahren zur statischen Optimierung. Die Leistungsfähigkeit der implementierten Verfahren wird empirisch anhand der Kriterien Genauigkeit, Zuverlässigkeit und Rechenaufwand mithilfe von Beispielproblemen untersucht und verglichen. Diese Beispielprobleme umfassen die Trajektorienplanung für drei verschiedene Wagen-Pendel-Systeme. Dabei wird auch auf die Schwierigkeit dieser Probleme eingegangen. Von allen implementierten Verfahren erwies sich ein direktes orthogonales Kollokationsverfahren mit einer stückweise konstanten Parametrierung des Eingangssignals als das leistungsfähigste und als das einzige Verfahren, welches Trajektorien zum Aufschwingen des Dreifach-Wagen-Pendels bestimmen konnte. Dieses Problem erwies sich erwartungsgemäß als das schwierigste aller untersuchten Probleme.

Abstract

This thesis deals with the comparison of trajectory planning methods for dynamical systems. The theory of underlying numerical methods is described in detail. The optimal control problem is introduced as special case of trajectory planning and allows two different approaches to the problem, namely direct and indirect methods. Specifically, direct and indirect variants of shooting and collocation methods were implemented due to their importance stressed in literature. The implementation of said methods is based on the software framework CasADi. All implemented methods use IPOPT, which implements an interior point method to solve problems of nonlinear optimization. The performance of the implemented methods in terms of accuracy, reliability, and computational effort was empirically examined and compared. For this purpose, different trajectory planning problems for three different pendulum on cart systems were considered and briefly discussed in terms of difficulty. A direct orthogonal collocation method using a piecewise constant parametrization of the input signal turned out to be method with the best overall performance. It was also the only method that was able to compute trajectories for the swing up of the triple pendulum on cart, which proved to be the most difficult of all problems considered.

Inhaltsverzeichnis

1	Einleitung	1
2	Theoretische Grundlagen	4
2.1	Polynominterpolation	4
2.2	Nichtlineare Gleichungssysteme	5
2.3	Numerische Integration	6
2.4	Anfangswertprobleme	7
2.5	Randwertprobleme	9
2.6	Statische Optimierung	14
2.7	Nichtlineare dynamische Systeme	15
2.8	Optimale Steuerung	17
3	Verfahren zur Trajektorienplanung	20
3.1	Problemdefinition	20
3.2	Verfahren im Überblick	21
3.3	Direkte Verfahren	24
3.4	Indirekte Verfahren	26
3.5	Implementierung und Software	27
4	Empirischer Verfahrensvergleich	33
4.1	Methodik	33
4.2	Beispielsysteme	35
4.3	Beispielprobleme	37
4.4	Ergebnisse	38
5	Zusammenfassung	53
	Literatur	55
	Abkürzungsverzeichnis	58
	Abbildungsverzeichnis	60
	Tabellenverzeichnis	61

Kapitel 1

Einleitung

Unter Trajektorienplanung versteht man im Kontext der Regelungstechnik die Bestimmung eines Eingangssignals, welches ein dynamisches System in endlicher Zeit von einem gegebenen Anfangszustand in einen gewünschten Endzustand überführt. Es liegt ein Problem der Steuerung (engl. open-loop control, feed forward control) vor, welche von der Regelung (engl. closed-loop control, feedback control) zu unterscheiden ist. Im Unterschied zur Steuerung erfolgt bei der Regelung eine Rückkopplung der zu beeinflussenden Systemgröße.

Eine Zustandsrückführung soll ein dynamisches System von einem beliebigen Anfangszustand in einen gewünschten Endzustand überführen und kann insofern als Verallgemeinerung der Trajektorienplanung aufgefasst werden. Während man bei der Trajektorienplanung den zeitlichen Verlauf des Einganges bestimmt, handelt es sich bei einer Zustandsrückführung um eine Funktion, welche jedem Zustand einen Eingang eindeutig so zuordnet, dass der Endzustand eine asymptotische stabile Ruhelage des dynamischen Systems ist. Im Gegensatz zur Trajektorienplanung impliziert eine Zustandsrückführung somit keine bestimmte Überführungszeit, sondern im Allgemeinen einen unendlichen Zeithorizont. [19, vgl. S. 96f.]

Für nichtlineare Systeme ist die Bestimmung einer Zustandsrückführung ggf. nur schwer, nur mit eingeschränktem Definitionsbereich oder gar nicht möglich. Es ist anzunehmen, dass die Bedingungen für die Existenz einer geeigneten Trajektorie im Allgemeinen weniger restriktiv sind. Weitere Vorteile der Trajektorienplanung sind die prinzipielle Planbarkeit von Überführungsdauer sowie ggf. die Möglichkeit Nebenbedingungen bezüglich von Eingang und Zustand wie etwa Beschränkungen beim Entwurf direkt zu berücksichtigen. Beim Entwurf einer Zustandsrückführung lassen sich derartige Eigenschaften einer konkreten Trajektorie hingegen nicht direkt beeinflussen. [19, vgl. S. 96f.]

Eine Zustandsrückführung ist aufgrund der Rückkopplung prinzipiell dazu in der Lage ein System trotz Störungen, Modellunbestimmtheiten oder Anfangsfehlern des Zustandes in den Endzustand zu überführen. Im Falle einer Steuerung auf Basis einer Trajektorienplanung ist dies aufgrund des offenen Wirkungskreislaufes nicht möglich und es sind

für diesen Fall Abweichungen von der Solltrajektorie und insbesondere vom Endzustand zu erwarten. [19, vgl. S. 96f.]

Daher wird die Steuerung eines Systems mithilfe einer geplanten Eingangstrajektorie in der Praxis üblicherweise mit einer Regelung durch eine Zustandsrückführung kombiniert. Man spricht in diesem Fall von *Vorsteuerung*, *Trajektorien-Folgeregelung* oder *Zwei-Freiheitsgrade-Regelung* [14], [19, S. 96f.]. Dieses Konzept der Regelung wird insbesondere zum Wechsel zwischen verschiedenen Arbeitspunkten bzw. Ruhelagen eines Systems oder zur Realisierung von Bewegungsabläufen angewendet.

Eine weitere prinzipielle Anwendungsmöglichkeit der Trajektorienplanung ist die *nicht-lineare modellprädiktive Regelung* (NMPC, engl. nonlinear model predictive control), bei der eine konsekutive optimierungsbasierte Berechnung von Trajektorien jeweils ausgehend vom aktuellen Systemzustand erfolgt. Dies entspricht einer dynamischen Zustandsrückführung auf Basis einer Echtzeit-Trajektorienplanung und erfordert somit leistungsfähige rechentechnische Systeme in Kombination mit effizienten numerischen Verfahren. Zur Senkung des Rechenaufwandes wird dabei das Verfahren zur Trajektorienplanung mithilfe der im vorherigen Zeitschritt berechneten Trajektorie initialisiert. [16, vgl. S. 1-3, 367ff.], [10]

Die klassischen Anwendungsgebiete der Trajektorienplanung liegen insbesondere in der chemischen Prozessindustrie sowie der Luft- und Raumfahrttechnik. Ein weiteres wichtiges Anwendungsgebiet ist die Robotik. Die praktische Relevanz der Trajektorienplanung ist auch im Kontext steigender Anforderungen an Regelungstechnik im Allgemeinen sowie das Führungsverhalten im Speziellen zu sehen. Dabei könnte insbesondere die NMPC die Vorteile einer expliziten Trajektorienplanung für ein größeres Spektrum technischer Systeme nutzbar machen.

Die vielfältigen Anwendungsmöglichkeiten der Trajektorienplanung sind nicht zuletzt auch eine wichtige Motivation zum Vergleich von Verfahren zur Trajektorienplanung wie er Ziel dieser Studienarbeit ist. In Kapitel 2 werden hierzu zunächst die mathematischen Grundlagen der Verfahren sowie die Beschreibung und Steuerung nichtlinearer dynamischer Systeme behandelt. Anschließend werden in Kapitel 3 wichtige Verfahren zur Trajektorienplanung beschrieben. Ein besonderer Fokus liegt dabei auf den im Rahmen dieser Arbeit implementierten Verfahren.

Mit Blick auf die Implementierung konkreter Verfahren ist auch der hierfür nötige Aufwand und dessen Verhältnis zur Leistungsfähigkeit interessant. In diesem Zusammenhang spielt insbesondere die Nutzung bereits existierender Software eine wichtige Rolle, auf welche daher neben der Implementierung der Verfahren selbst ausführlich eingegangen wird.

Der Vergleich der implementierten Verfahren zur Trajektorienplanung erfolgt schließlich in Kapitel 4 anhand geeigneter Beispielpunkte auf Basis verschiedener dynamischer Systeme. Dabei wird neben Zuverlässigkeit, Rechenaufwand und Genauigkeit der Ver-

fahren auch die Schwierigkeit der Beispielp Probleme untersucht und verglichen.

Hinweis zu Abkürzungen

Aufgrund der großen Wortlänge einiger Fachbegriffe wurde verstärkt mit Abkürzungen gearbeitet. Ein Verzeichnis der verwendeten Abkürzungen findet sich ab Seite 58.

Hinweis zu Literaturverweisen

Die vorkommenden Literaturverweise besitzen je nach Position eine unterschiedliche Bedeutung. Ein Verweis, welcher sich vor dem Punkt eines Satzes befindet, bezieht sich nur auf diesen Satz. Steht der Verweis hingegen nach dem Punkt eines Satzes, so bezieht dieser sich auf alle vorangegangenen Sätze in diesem Absatz bzw. auf alle vorangegangenen Sätze bis zu einem anderen gleichrangigen Verweis im selben Absatz. Davon ausgenommen sind einzelne Sätze eines Absatzes, welche mit einem gesonderten Verweis (vor dem Punkt) versehen sind. Ein Absatz in diesem Sinne wird durch Gleichungen, Tabellen oder Abbildungen nicht unterbrochen.

Insbesondere bei indirekten Zitaten wurden bei umfangreicheren Literaturquellen die betreffenden Seiten oder Abschnitte angegeben. Wurden lediglich einzelne Sachverhalte übernommen oder waren diese Ausgangspunkt für eigene Überlegungen, so wurde dies in Fällen, in denen es sinnvoll erschien, mit dem Zusatz „vgl.“ gekennzeichnet. Auf eine wörtliche Wiedergabe ganzer Sätze wurde vollständig verzichtet.

Hinweis zur Notation

Für einen Vektor $\boldsymbol{\theta} \in \mathbb{R}^n$ wird in dieser Arbeit immer die Komponentendarstellung $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_n)^T$ angenommen.

Kapitel 2

Theoretische Grundlagen

Die Trajektorienplanung für dynamische Systeme ist im Allgemeinen ein anspruchsvolles mathematisches Problem. Insgesamt ist die Lösung verschiedener mathematischer Teilprobleme abnehmender Komplexität unter Zuhilfenahme numerischer Lösungsverfahren erforderlich.

In diesem Kapitel werden zunächst relevante auftretende mathematische (Teil-) Probleme sowie die Grundlagen gängiger Lösungsverfahren mit Verweisen auf weiterführende Literatur vorgestellt. Weiterhin erfolgt die Definition von Eigenschaften und Darstellungsformen nichtlinearer dynamischer Systeme. Schließlich wird eine kurze Einführung in die Grundlagen der optimalen Steuerung dynamischer Systeme gegeben.

2.1 Polynominterpolation

Gesucht ist ein reelles Polynom

$$P(x) := \sum_{k=0}^M a_k x^k \quad (2.1)$$

vom Höchstgrad M und den $M + 1$ Koeffizienten a_0, a_1, \dots, a_M , welches an $M + 1$ voneinander verschiedenen Stützstellen x_0, x_1, \dots, x_M die *Interpolationsbedingung*

$$P(x_k) = y_k, \quad k = 0, 1, \dots, M, \quad x_k \neq x_j \quad \forall k \neq j \quad (2.2)$$

erfüllt, also durch die Punkte $(x_k, y_k) \in \mathbb{R}^2$ verläuft.

Ein mögliches Lösungsverfahren ist die *Lagrange-Interpolation*. Die Konstruktion des Interpolationspolynoms erfolgt dabei durch Linearkombination sogenannter Lagrange-Polynome: Ein mit dem Stützwert y_k gewichtetes Lagrange-Polynom nimmt an der zugehörigen Stützstelle x_k den Wert eins an und besitzt an allen anderen Stützstellen eine Nullstelle.

Formal wird das Interpolationspolynom P vom Höchstgrad M als

$$P(x) := \sum_{k=0}^M y_k L_k(x) \quad (2.3)$$

angesetzt. Für die $M + 1$ Lagrange-Polynome

$$L_k(x) := \prod_{j=0, j \neq k}^M \frac{x - x_j}{x_k - x_j} \quad (2.4)$$

vom Grad M gilt

$$L_k(x_i) = \begin{cases} 1, & \text{für } k = i \\ 0, & \text{für } k \neq i \end{cases}. \quad (2.5)$$

Somit erfüllt P die Interpolationsbedingung (2.2). Existenz und Eindeutigkeit eines solchen Interpolationspolynoms sind immer gegeben. [29, vgl. S. 38f.]

Anzumerken ist, dass die Lagrange-Interpolation im Allgemeinen kein effizientes numerisches Verfahren, sondern in erster Linie von theoretischer Bedeutung ist. Dennoch kann eine Anwendung zur Lösung mehrerer Interpolationsprobleme mit identischen Stützstellen und damit identischen Lagrange-Polynomen sinnvoll sein. [29, S. 38f.] Zudem erlaubt die Lagrange-Interpolation eine Parametrierung von Polynomen in Abhängigkeit der Punkte zwischen denen interpoliert werden soll.

2.2 Nichtlineare Gleichungssysteme

Ein *nichtlineares algebraisches Gleichungssystem* (NLGS) lässt sich mithilfe einer nichtlinearen Funktion $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ allgemein als

$$\mathbf{f}(\mathbf{x}) = \mathbf{0} \quad (2.6)$$

darstellen. Eine Lösung \mathbf{x}^* des NLGS entspricht einer Nullstelle von \mathbf{f} .

Da es im Allgemeinen nicht möglich ist eine analytische Lösung von (2.6) zu bestimmen, wendet man iterative Näherungsverfahren an. Ein klassischer Vertreter derartiger Verfahren ist das *Newton-Raphson-Verfahren* (kurz oft Newton-Verfahren). Voraussetzung für dessen Anwendbarkeit ist die stetige Differenzierbarkeit von \mathbf{f} . Ausgehend von einer Startschätzung \mathbf{x}_0 erfolgt eine Iteration gemäß der Vorschrift

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \mathbf{J}_i^{-1} \mathbf{f}(\mathbf{x}_i), \quad (2.7)$$

mit der Jacobi-Matrix

$$\mathbf{J}_i = \left. \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_i} \quad (2.8)$$

von $\mathbf{f}(\mathbf{x})$ an der Stelle \mathbf{x}_i , wobei vorausgesetzt wird, dass \mathbf{J}_i regulär ist. [29, S. 292f.]

Die *Konvergenz* gibt als Eigenschaft numerischer Iterationsverfahren Aufschluss darüber, ob und ggf. wie schnell sich die Iterationswerte einer Lösung annähern. Eine konvergente Folge $\{\mathbf{x}_i\}$ mit dem Grenzwert \mathbf{x}^* besitzt mindestens die *Konvergenzordnung* $p \geq 1$, wenn es eine Konstante $K > 0$ gibt, sodass

$$\|\mathbf{x}_{i+1} - \mathbf{x}^*\| \leq K \|\mathbf{x}_i - \mathbf{x}^*\|^p, \quad \forall i \geq 0. \quad (2.9)$$

Je kleiner K und je größer p ist, desto höher ist die *Konvergenzgeschwindigkeit* der Folge. Der Einfluss der Konvergenzordnung p ist dabei wesentlich stärker. [27, S. 186]

Das Newton-Verfahren ist unter bestimmten Voraussetzungen lokal quadratisch konvergent, d. h. das Verfahren besitzt die Konvergenzordnung 2, wenn sich die Startschätzung bereits hinreichend nah an der gesuchten Lösung befindet und diese Lösung in dem entsprechenden Gebiet eindeutig ist (s. [29, S. 298-302]). Eine Divergenz des Newton-Verfahrens kann sich dadurch äußern, dass die Iterationsfolge über alle Grenzen hinaus wächst oder seltener durch ein oszillierendes Verhalten [27, S. 193].

Insbesondere durch die hohe Konvergenzgeschwindigkeit ist das Newton-Verfahren praxisrelevant. Da jedoch nur lokal Konvergenz auftritt, wird sich die vorgestellte Vorgehensweise ohne eine ungefähre Kenntnis der Lösung des NLGS als unzureichend erweisen. Daher zielen die in der Praxis anzutreffenden Erweiterungen des Verfahrens insbesondere auf eine Vergrößerung des Konvergenzgebietes z. B. durch Liniensuchverfahren sowie eine Verbesserung der Startschätzung ab. Darüber hinaus sind effiziente numerische Verfahren zur Bestimmung der Jacobi-Matrix und der Lösung des mit (2.7) korrespondierenden linearen Gleichungssystems nötig, um den sehr hohen Rechenaufwand pro Iterationsschritt zu reduzieren. [5, S. 327ff.]

2.3 Numerische Integration

Unter numerischer Integration oder numerischer Quadratur versteht man die Approximation eines bestimmten Integrals

$$I = \int_a^b f(x) dx \quad (2.10)$$

mithilfe einer *Quadraturformel*

$$Q = \sum_{i=1}^n w_i f(x_i) \quad (2.11)$$

mit den Integrationsgewichten w_i und Stützstellen x_i . [27, S. 307]

Bei der *Gauß-Quadratur* werden die Integrationsgewichte und Stützstellen so gewählt, dass ein möglichst hoher Genauigkeitsgrad d erreicht wird. Der Genauigkeitsgrad gibt

an, bis zu welchem Grad eine polynomiale Funktion f exakt integriert werden würde. Es existiert genau eine Quadraturformel mit n Stützstellen und dem maximal erreichbaren Genauigkeitsgrad $d = 2n - 1$. Bei dieser Quadraturformel entsprechen die Stützstellen den Nullstellen des n -ten *Legendre-Polynoms*

$$P_n(x) := \frac{1}{2^n \cdot n!} \frac{d^n}{dx^n} (x^2 - 1)^n, \quad n \in \mathbb{N} \quad (2.12)$$

und die Integrationsgewichte sind durch

$$w_i := \int_{-1}^1 \prod_{j=1, j \neq i}^n \left(\frac{x - x_j}{x_i - x_j} \right)^2 dx, \quad i = 1, 2, \dots, n \quad (2.13)$$

festgelegt. [27, S. 323f.] Im Allgemeinen ist f kein Polynom. Dennoch wird ein Taylor-Polynom vom Grad d von f exakt integriert. Für eine möglichst genaue Integration einer beliebigen Funktion muss diese somit hinreichend oft differenzierbar auf $[a, b]$ und die Taylor-Polynome von f um s für $\forall s \in [a, b]$ annähernd identisch sein. -> kürzer und einfacher formulieren: hinreichend glatt und kleine Intervalle

2.4 Anfangswertprobleme

Bei einem *Anfangswertproblem (AWP)* einer gewöhnlichen *Differentialgleichung (DGL)*

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{x}(t) \in \mathbb{R}^n, \quad t \in [t_0, t_f] \subset \mathbb{R} \quad (2.14)$$

ist deren Lösung $\mathbf{x}(t)$ für einen gegebenen Anfangswert \mathbf{x}_0 gesucht. Existenz und Eindeutigkeit der Lösung seien vorausgesetzt (siehe [29, S. 467]).

AWP lassen sich in den meisten Fällen nicht analytisch lösen. Insbesondere zur rechnergestützten Lösung bedarf es daher numerischer Verfahren, welche eine hinreichend genaue Näherungslösung $\hat{\mathbf{x}}(t)$ liefern. Derartigen Verfahren ist gemeinsam, dass die Lösung nur an bestimmten Stützstellen t_i ausgehend vom gegebenen Anfangswert \mathbf{x}_0 rekursiv mit einer (variablen) Schrittweite $h_i := t_{i+1} - t_i$ gelöst wird. Der Ansatz ist dabei die Approximation von Ableitungen mithilfe von Differenzenquotienten. Im einfachsten Fall lässt sich z. B. aus der Vorwärtsdifferenz

$$\mathbf{f}(t_i, \mathbf{x}(t_i)) \approx \frac{\mathbf{x}(t_{i+1}) - \mathbf{x}(t_i)}{h_i} \quad (2.15)$$

die Rekursionsvorschrift

$$\hat{\mathbf{x}}_{i+1} = \hat{\mathbf{x}}_i + h_i \mathbf{f}(t_i, \hat{\mathbf{x}}_i) \quad (2.16)$$

ableiten. Bei diesem Verfahren nach Euler handelt es sich um ein *Einschrittverfahren*, da zur Bestimmung des Näherungswertes $\hat{\mathbf{x}}_{i+1}$ an der Stützstelle t_{i+1} nur der letzte Näherungswert $\hat{\mathbf{x}}_i$ an der Stützstelle t_i herangezogen wird. Werden mehrere zurückliegende

Näherungswerte zur Bestimmung eines folgenden Näherungswertes herangezogen, so spricht man von *Mehrschrittverfahren*. [27, S. 342-347]

Einschrittverfahren besitzen die allgemeine Form

$$\hat{\mathbf{x}}_0 = \mathbf{x}_0 \quad (2.17a)$$

$$\hat{\mathbf{x}}_{i+1} = \hat{\mathbf{x}}_i + h_i \varphi(t_i, t_{i+1}, \hat{\mathbf{x}}_i, \hat{\mathbf{x}}_{i+1}) \quad (2.17b)$$

mit der Verfahrensfunktion φ . Ist die Verfahrensfunktion vom zu bestimmenden Näherungswert $\hat{\mathbf{x}}_{i+1}$ selbst abhängig, so ist die Rekursionsvorschrift (2.17b) und das zugehörige Verfahren implizit. Andernfalls spricht man von expliziten Verfahren. Bei impliziten Verfahren muss in jedem Integrationsschritt ein im Allgemeinen nichtlineares Gleichungssystem gelöst werden. [27, S. 342-347, 351]

Die Konvergenz von Einschrittverfahren folgt aus *Konsistenz* und *Stabilität*. Alle relevanten Verfahren sind konsistent. Besitzt die Lösung des AWP eine starke Empfindlichkeit gegenüber kleinen Änderungen des Anfangswertes spricht man unabhängig vom verwendeten Verfahren von *inhärenter Instabilität* des AWP. Die Näherungslösung kann sich dann exponentiell von der tatsächlichen Lösung des AWP entfernen. Im Falle von stabilen AWP sind alle impliziten Verfahren numerisch stabil. Bei expliziten Verfahren ist in diesem Fall die Kombination von Verfahren und verwendeter Schrittweite entscheidend. [27, S. 376-380]

Eine bedeutende Klasse von Einschrittverfahren sind die *Runge-Kutta-Verfahren (RKV)*. Ein M -stufiges RKV besitzt die allgemeine Form

$$\hat{\mathbf{x}}_0 = \mathbf{x}_0 \quad (2.18a)$$

$$\hat{\mathbf{x}}_{i+1} = \hat{\mathbf{x}}_i + h_i \sum_{j=1}^M \gamma_j k_j(t_i, \hat{\mathbf{x}}_i) \quad (2.18b)$$

$$k_j(t, \mathbf{x}) = \mathbf{f} \left(t + \alpha_j h_j, \mathbf{x} + h_j \sum_{l=1}^M \beta_{jl} k_l(t, \mathbf{x}) \right) \quad (2.18c)$$

mit bekannten Verfahrensparametern γ_j , α_j und β_{jl} . Für $\beta_{jl} = 0$, $\forall l \geq j$ lassen sich die Funktionsauswertungen k_j rekursiv bestimmen und es liegt ein explizites Verfahren vor. Andernfalls handelt es sich um ein implizites RKV. [27, S. 353, 5, S. 69]

Ein Einschrittverfahren mit der maximalen Schrittweite $h_{\max} = \max_i h_i$ heißt *konvergent*, falls

$$\max_i \|\hat{\mathbf{x}}_i - \mathbf{x}(t_i)\| \rightarrow 0 \text{ mit } h_{\max} \rightarrow 0 \quad (2.19)$$

und besitzt die *Konvergenzordnung* p , falls es eine Konstante K mit

$$\max_i \|\hat{\mathbf{x}}_i - \mathbf{x}(t_i)\| \leq K h_{\max}^p \text{ mit } h_{\max} \rightarrow 0 \quad (2.20)$$

gibt. [27, S. 351]

Bei der Implementierung eines Verfahrens zur Lösung von AWP spielt insbesondere die Steuerung der Schrittweite eine große Rolle. Dabei wird die Schrittweite gerade so gewählt, dass eine vom Nutzer vorgegebene Fehlertoleranz bei möglichst kleinem Rechenaufwand, d. h. möglichst großer Schrittweite, eingehalten wird. Diese Toleranz ϵ sollte gemäß

$$\epsilon \geq K \epsilon_M \quad (2.21)$$

nicht kleiner als das Produkt aus relativer Maschinentoleranz ϵ_M und der zu erwartenden Obergrenze K der Lösung des AWP sein. [29, S. 485f.]

2.5 Randwertprobleme

Bei einem *Randwertproblem (RWP)* einer gewöhnlichen DGL ist deren Lösung $\mathbf{x}(t)$ unter Randbedingungen \mathbf{g} für mindestens zwei gegebene Randwerte, d. h. Werte von \mathbf{x} an verschiedenen Stellen t , gesucht. Im engeren Sinne (und auch in dieser Arbeit) sind damit Probleme mit Bedingungen für die beiden Werte an den Rändern des Lösungsintervalls $[t_0, t_f]$ gemeint. Diese besitzen die Form

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}), \quad \mathbf{g}(\mathbf{x}(t_0), \mathbf{x}(t_f)) = \mathbf{0}, \quad \mathbf{x}(t), \mathbf{g} \in \mathbb{R}^n, \quad t \in [t_0, t_f] \subset \mathbb{R} \quad (2.22)$$

und werden in Abgrenzung zum allgemeineren Fall von mehr als zwei Randwerten manchmal auch *Zwei-Punkt-Randwertprobleme* genannt. Dabei ist weder der Anfangs- noch der Endwert von \mathbf{x} vollständig durch die Randbedingungen bestimmt, da sonst entweder ein AWP oder ein überbestimmtes RWP vorliegen würde. Treten zusätzlich freie Parameter $\mathbf{p} \in \mathbb{R}^l$ in (2.22) auf, so können die Randwerte für $l \geq n$ vollständig durch Randbedingungen $\mathbf{g} \in \mathbb{R}^{2n}$ vorgegeben werden, ohne dass dies zu einem überbestimmten RWP führt.

Einen umfassenden und detaillierten Überblick zu Verfahren zur numerischen Lösung von RWP bietet [5]. Zu den wichtigsten Verfahren zählen die Schieß- und Kollokationsverfahren. Diese werden im Folgenden anhand zwei spezieller RWP, welche in Kapitel 3 auftreten werden, erläutert. Zum einen wird dies ein Problem

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}), \quad x_i(t_0) = x_{0,i}, \quad x_j(t_f) = x_{f,j}, \quad \mathbf{x} \in \mathbb{R}^n, \quad t \in [t_0, t_f] \subset \mathbb{R} \quad (2.23a)$$

$$i \in \mathcal{I}, \quad j \in \mathcal{J} \quad \text{mit} \quad \mathcal{I}, \mathcal{J} \subset \{1, 2, \dots, n\} \quad \text{und} \quad |\mathcal{I}| + |\mathcal{J}| = n \quad (2.23b)$$

mit gemischten Randbedingungen $\mathbf{x}(t_0) = \mathbf{x}_0$, $\mathbf{x}(t_f) = \mathbf{x}_f$ mit unvollständig bekannten Randwerten $\mathbf{x}_0, \mathbf{x}_f$ sein. Zum anderen wird ein Problem

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{p}), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_f, \quad \mathbf{x} \in \mathbb{R}^n, \quad t \in [t_0, t_f] \subset \mathbb{R} \quad (2.24)$$

mit vollständig bekannten Randwerten, jedoch unbekannten freien Parametern $\mathbf{p} \in \mathbb{R}^l$, $l \geq n$ zu lösen sein. Bei beiden Problemen ist also ein Vektor von unbekannten

Variablen \mathbf{v} gesucht: Beim **RWP** (2.23) ist $\mathbf{x}_0 = \mathbf{x}_0(\mathbf{v})$ und beim **RWP** (2.24) ist $\mathbf{p} = \mathbf{v}$. Im Folgenden wird die Abhängigkeit von den unbekannten Problem-Variablen \mathbf{v} zur Wahrung der Allgemeinheit und Lesbarkeit nicht explizit angegeben.

Die zur Lösung verwendbaren Schieß- und Kollokationsverfahren basieren zu einem großen Teil auf unterschiedlichen Verfahren zur Lösung von **AWP**. Die Bedingungen für Existenz und Eindeutigkeit der Lösung eines **RWP** sind jedoch wesentlich restriktiver als bei **AWP** (siehe [5, S. 87-93]).

Schießverfahren

Bei *Einfach-Schießverfahren* (**ESV**) wird zunächst das durch eine willkürliche Vorgabe der Variablen \mathbf{v} entstehende **AWP** gelöst. Der Wert der Lösung im Endpunkt $\mathbf{x}(t_f, \mathbf{x}_0)$ wird zu Beginn höchstwahrscheinlich nicht mit dem Endwert \mathbf{x}_f übereinstimmen. Es ergibt sich somit ein **NLGS**

$$\mathbf{g}_{\text{ESV}}(\mathbf{v}) := \mathbf{x}(t_f, \mathbf{x}_0) - \mathbf{x}_f \stackrel{!}{=} \mathbf{0}, \quad (2.25)$$

dessen Lösung durch eine iterative Anpassung von \mathbf{v} bestimmt werden kann. Dabei muss bei jeder Iteration zunächst ein neues **AWP** gelöst werden. [5, S. 170f.] Handelt es sich bei den unbekannten Variablen um Anfangswerte so ist dieses Vorgehen anschaulich mit dem Einschießen von Artilleriegeschützen, bei dem z. B. der Abschusswinkel solange variiert wird, bis das Ziel getroffen wurde, vergleichbar. Insofern erklärt sich der Name des Verfahrens.

„Ungünstige“ Vorgaben von \mathbf{v} können zu inhärent instabilen **AWP** führen, für welche keine Lösung auf dem gesamten Intervall $[t_0, t_f]$ existiert. Die Lösung wächst dann vor Erreichen der rechten Intervallgrenze über alle Grenzen hinaus. Eine Gegenmaßnahme ist die Zerlegung des Intervalls und die Vorgabe eines Anfangswertes für jedes Teilintervall, um die entstehenden **AWP** separat zu lösen. Zur Unterscheidung vom bisherigen Ansatz spricht man dann von *Mehrfach-Schießverfahren* (**MSV**). [5, S. 174ff.]

Bei **MSV** ergibt sich durch die Forderung nach Stetigkeit der Lösung an den Teilintervallgrenzen bei N Teilintervallen $[t_i, t_{i+1}]$, $i = 0, 1, \dots, N-1$, $t_N := t_f$ mit den zusätzlich vorgegebenen Anfangswerten $\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \dots, \hat{\mathbf{x}}_{N-1}$ ein entsprechend größeres **NLGS**

$$\mathbf{g}_{\text{MSV}}(\mathbf{v}) := \begin{pmatrix} \mathbf{x}(t_1, \mathbf{x}_0) - \hat{\mathbf{x}}_1 \\ \mathbf{x}(t_2, \hat{\mathbf{x}}_1) - \hat{\mathbf{x}}_2 \\ \mathbf{x}(t_3, \hat{\mathbf{x}}_2) - \hat{\mathbf{x}}_3 \\ \vdots \\ \mathbf{x}(t_N, \hat{\mathbf{x}}_{N-1}) - \mathbf{x}_f \end{pmatrix} \stackrel{!}{=} \mathbf{0}, \quad (2.26)$$

bei dessen Lösung in jedem Iterationsschritt N **AWP** gelöst werden müssen. [5, S. 174ff.] Die zusätzlichen Anfangswerte sind unbekannt und somit Teil des Variablen-Vektors \mathbf{v} .

Bei Schießverfahren wird zur iterativen Lösung des entstehenden **NLGS** mit einem Newton-Verfahren die Jacobi-Matrix bezüglich der unbekannten Variablen \mathbf{v} benötigt.

Hierzu muss die *Sensitivität* der Lösung der **AWP** bezüglich der unbekannten Variablen

$$\frac{\partial \mathbf{x}(t, \mathbf{v})}{\partial \mathbf{v}} \quad (2.27)$$

bestimmt werden. Die Sensitivität der Lösung eines **AWP** kann z. B. mithilfe des Differenzenquotienten

$$\frac{\partial \mathbf{x}(t, \mathbf{v})}{\partial v_j} \approx \frac{\mathbf{x}(t, \mathbf{v} + \Delta v_j \mathbf{e}_j) - \mathbf{x}(t, \mathbf{v})}{\Delta v_j}, \quad \mathbf{v} \in \mathbb{R}^m, \quad j = 1, 2, \dots, m \quad (2.28)$$

mit dem Einheitsvektor \mathbf{e}_j für jede Komponente v_j approximiert werden. Dazu muss das **AWP** jeweils für eine kleine Änderung Δv_j der entsprechenden Komponente erneut gelöst werden. Insbesondere für eine große Anzahl von Problemvariablen ist dieser Ansatz somit nicht praktikabel und man muss auf effizientere Verfahren zurückgreifen. Einen Überblick zur Sensitivitätsanalyse von **AWP** bieten [29, S. 538f. 11, S. 192ff.].

Kollokationsverfahren

Kollokationsverfahren (KKV) nähern die Lösung von **RWP** auf Teilintervallen mithilfe von Funktionen mit einer endlichen Anzahl von Parametern an. Dabei werden üblicherweise polynomiale Funktionen verwendet. Als Lösung ergibt sich im Allgemeinen eine stückweise definierte Funktion. Die Parameter der einzelnen Polynome werden iterativ so angepasst, dass die dem Problem zugrundeliegende **DGL** innerhalb jedes Teilintervalls nur in bestimmten Punkten, den sogenannten *Kollokationspunkten*, erfüllt wird. Dabei ist es nicht nur aus Notationsgründen vorteilhaft, Lagrange-Interpolationspolynome anzusetzen (siehe Abschnitt 2.1).

Wie Abbildung 2.1 zeigt, wird das Intervall $[t_0, t_f]$ in N Teilintervalle zerlegt. Innerhalb der Teilintervalle $[t_i, t_{i+1}]$, $i = 0, 1, \dots, N-1$, $t_N := t_f$ liegen jeweils die Punkte $t_{ij} := t_i + hc_j$, $j = 1, \dots, M$ mit den M Kollokationspunkten $0 < c_1 < \dots < c_M < 1$ und der Schrittweite $h := t_{i+1} - t_i$. Auf jedem Teilintervall wird die Lösung $\mathbf{x}(t)$ des **RWP** als Interpolationspolynom $\xi_i(t)$ vom Höchstgrad M

$$\xi_i(t) := \sum_{j=0}^M \hat{\mathbf{x}}_{ij} L_j(c(t)), \quad c(t) := \frac{t - t_i}{h} \in [0, 1] \quad (2.29)$$

mit den unbekannten Stützwerten $\hat{\mathbf{x}}_{ij}$ und den (bekannten) Lagrange-Polynomen

$$L_j(c) := \prod_{k=0, k \neq j}^M \frac{c - c_k}{c_j - c_k}, \quad c_0 := 0 \quad (2.30)$$

angenähert. Die Interpolationspolynome erfüllen per Konstruktion jeweils die Zusammenhänge

$$\xi_i(t_{ij}) = \hat{\mathbf{x}}_{ij}, \quad (2.31a)$$

$$\dot{\xi}_i(t_{ij}) = \frac{1}{h} \sum_{k=0}^M \hat{\mathbf{x}}_{ik} \left. \frac{\partial L_k(c)}{\partial c} \right|_{c=c_j} \quad (2.31b)$$

in den Punkten t_{ij} und

$$\xi_i(t_{i+1}) = \sum_{k=0}^M \hat{\mathbf{x}}_{ik} L_k(1) \quad (2.31c)$$

am rechten Rand der Teilintervalle. Wie Abbildung 2.1 verdeutlicht, ist dabei die Stetigkeit der Näherungslösung an den Teilintervallgrenzen nicht sichergestellt. Außerdem genügt die Näherungslösung im Allgemeinen nicht der dem RWP zugrundeliegenden DGL. Um beides sicherzustellen, fordert man daher, dass die Interpolationspolynome die *Kollokationsbedingungen*

$$\dot{\xi}_i(t_{ij}) - \mathbf{f}(t_{ij}, \hat{\mathbf{x}}_{ij}) = \mathbf{0}, \quad i = 0, 1, \dots, N-1, \quad j = 1, \dots, M \quad (2.32a)$$

mit $\dot{\xi}_i(t_{ij})$ aus (2.31b) sowie die Stetigkeitsbedingungen

$$\xi_i(t_{i+1}) - \hat{\mathbf{x}}_{(i+1)j} = \mathbf{0}, \quad i = 0, 1, \dots, N-1, \quad j = 0 \quad (2.32b)$$

mit $\xi_i(t_{i+1})$ aus (2.31c) erfüllen¹. [8, vgl. S. 289f., 293] Die Stützwerte entsprechen dabei den Parametern der Interpolationspolynome. Der Stützwert $\hat{\mathbf{x}}_{00}$, welcher somit implizit in (2.32) auftritt, ist der Anfangswert \mathbf{x}_0 . Der Wert $\hat{\mathbf{x}}_{N0}$ in (2.32b) ist der Endwert \mathbf{x}_f und daher kein Stützwert eines Interpolationspolynoms.

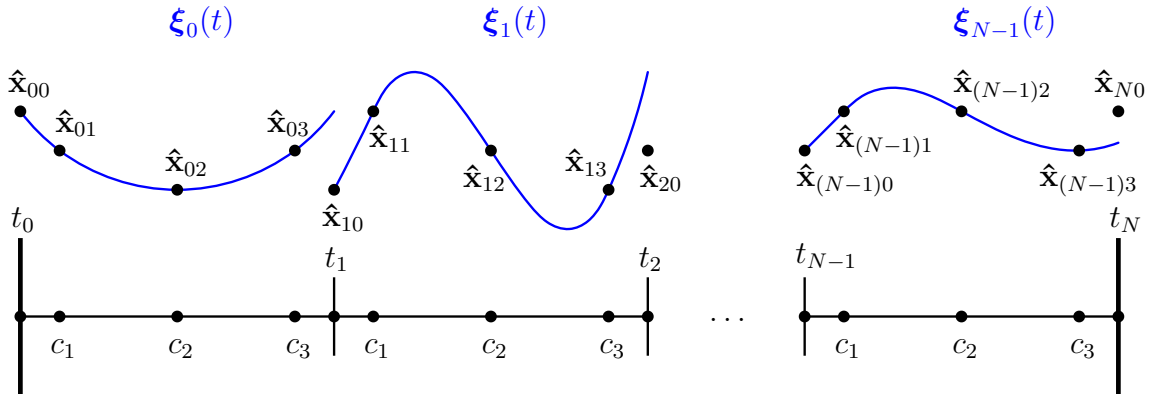


Abbildung 2.1 – Schematische Darstellung des Kollokationsverfahrens für $M = 3$.

Die Stützwerte sind Teil der Problemvariablen \mathbf{v} und müssen bei der Initialisierung des Verfahrens vorgegeben werden. Für das RWP (2.23) ist der unvollständig bekannte Anfangswert Teil der Stützwerte und der unvollständig bekannte Endwert muss zusätzlich vorgegeben werden. Beim RWP (2.24) müssen zusätzlich die unbekannten Parameter \mathbf{p}

¹Man beachte die doppelte Rolle der Punkte c_i : Bei den Punkten c_1, \dots, c_M handelt es sich um Kollokationspunkte, welche gleichzeitig den Stützpunkten der Lagrange-Polynome entsprechen. Für die Lagrange-Polynome (2.30) ist ein zusätzlicher Stützpunkt c_0 nötig, welcher kein Kollokationspunkt ist, d. h. nicht in den Kollokationsbedingungen (2.32a) auftritt.

vorgegeben werden. Insgesamt ergeben die Kollokations- und Stetigkeitsbedingungen ein **NLGS** der Form

$$\mathbf{g}_{\text{KKV}}(\mathbf{v}) := \begin{pmatrix} \dot{\xi}_0(t_{01}, \hat{\mathbf{x}}_{00}, \dots, \hat{\mathbf{x}}_{0M}) - \mathbf{f}(t_{01}, \hat{\mathbf{x}}_{01}) \\ \vdots \\ \dot{\xi}_0(t_{0M}, \hat{\mathbf{x}}_{00}, \dots, \hat{\mathbf{x}}_{0M}) - \mathbf{f}(t_{0M}, \hat{\mathbf{x}}_{0M}) \\ \dot{\xi}_1(t_{11}, \hat{\mathbf{x}}_{10}, \dots, \hat{\mathbf{x}}_{1M}) - \mathbf{f}(t_{11}, \hat{\mathbf{x}}_{11}) \\ \vdots \\ \dot{\xi}_{N-1}(t_{(N-1)M}, \hat{\mathbf{x}}_{(N-1)0}, \dots, \hat{\mathbf{x}}_{(N-1)M}) - \mathbf{f}(t_{(N-1)M}, \hat{\mathbf{x}}_{(N-1)M}) \\ \xi_0(t_1, \hat{\mathbf{x}}_{00}, \dots, \hat{\mathbf{x}}_{0M}) - \hat{\mathbf{x}}_{10} \\ \vdots \\ \xi_{N-1}(t_N, \hat{\mathbf{x}}_{(N-1)0}, \dots, \hat{\mathbf{x}}_{(N-1)M}) - \hat{\mathbf{x}}_{N0} \end{pmatrix} \stackrel{!}{=} \mathbf{0}. \quad (2.33)$$

Im Gegensatz zu den Schießverfahren müssen zum Aufstellen des **NLGS** keine **AWP** gelöst werden. Das **NLGS** lässt sich somit analytisch explizit in den Problemvariablen \mathbf{v} darstellen und kann iterativ gelöst werden, ohne dass in jedem Schritt eine Sensitivitätsanalyse durchgeführt werden muss.

Es existieren verschiedene Formulierungen von **KKV**, welche im Ergebnis jedoch gleichwertig sind. Diese Formulierungen entstehen durch unterschiedlich angesetzte Interpolationspolynome. So kann abweichend von der bisherigen Formulierung in (2.29), bei der $\mathbf{x}(t)$ mit Interpolationspolynomen $\xi_i(t)$ vom Höchstgrad M angenähert wird, alternativ auch $\dot{\mathbf{x}}(t)$ mit Interpolationspolynomen vom Höchstgrad $M - 1$ angenähert werden. Die Kollokations- und Stetigkeitsbedingungen lassen sich dann analog formulieren. Man kann zeigen, dass diese alternative Formulierung der Bedingungen äquivalent zu M -stufigen impliziten **RKV** der Form (2.18) mit den Verfahrensparametern

$$\gamma_j := \int_0^1 L_j(c) \, dc, \quad \alpha_j := c_j, \quad \beta_{jl} := \int_0^{c_j} L_l(c) \, dc, \quad L_j(c) := \prod_{k=1, k \neq j}^M \frac{c - c_k}{c_j - c_k} \quad (2.34)$$

ist. [8, vgl. S. 289f.] Die Lösung von **RWP** mit **KKV** entspricht somit einer simultanen Lösung der bei M -stufigen impliziten **RKV** in jedem Rekursionsschritt, d. h. pro Teilintervall auftretenden **NLGS**.²

Die **RKV**-Formulierung von **KKV** mag zunächst wenig intuitiv erscheinen. Allerdings erlaubt diese Formulierung einen leichteren Zugang zur Wahl der Kollokationspunkte, welche bisher nicht thematisiert wurde, jedoch für die Genauigkeit eines **KKV** von entscheidender Bedeutung ist.

²Im Unterschied dazu erfolgt die Lösung von **AWP** durch implizite **RKV** sequentiell und es treten weder unbekannte Variablen \mathbf{v} auf noch Endbedingungen auf. Bei **KKV** für **RWP** müssen die **NLGS** der Teilintervalle simultan gelöst werden, da die einzelnen **NLGS** über die Stetigkeitsbedingungen mit dem geforderten Endwert verkoppelt sind.

Die Wahl der Kollokationspunkte bei **KKV** geht auf die Theorie der numerischen Integration zurück (siehe Abschnitt 2.3). **KKV** entsprechen der simultanen Anwendung der Rekursionsschritte eines impliziten **RKV** auf alle Teilintervalle. Ein Rekursionsschritt eines **RKV** entspricht der näherungsweise Lösung eines bestimmten Integrals durch numerische Integration. Aus Abschnitt 2.3 ist bekannt, dass bei der Gauß-Quadratur eine bestimmte Wahl der Integrationsgewichte und Stützpunkte zu einem hohen Genauigkeitsgrad der Integration führt. Als Stützpunkte wählt man für einen maximalen Genauigkeitsgrad die Nullstellen eines Legendre-Polynoms (2.12). Da diese Stützpunkte bei **KKV** den Kollokationspunkten entsprechen, motiviert dies die Wahl der Kollokationspunkte als Nullstellen des (verschobenen³) Legendre-Polynoms vom Grad M . Dies führt zu einem Genauigkeitsgrad von $2M - 1$ in Bezug auf $\dot{\mathbf{x}}(t)$ und entsprechend $2M$ für $\mathbf{x}(t)$.⁴ [8, vgl. S. 290-293]

Legendre-Polynome gehören zur Klasse der *Gauß-Jacobi-Polynome*, welche wiederum zu den orthogonalen Polynomen zählen. Die Eigenschaft der Orthogonalität ist an dieser Stelle jedoch nur insofern relevant als dass sie namensgebend für die Klasse der *orthogonalen Kollokationsverfahren* ist. Weitere relevante Gauß-Jacobi-Polynome sind z. B. die *Radau-* und *Lobatto-Polynome*, deren Nullstellen als Integrationsstützpunkte zu Genauigkeitsgraden von $2M - 2$ bzw. $2M - 3$ führen. Im Rahmen von **KKV** führen diese somit ebenfalls zu einer hohen Genauigkeit. **KKV** besitzen je nach Wahl der Kollokationspunkte unterschiedliche numerische Eigenschaften, welche in Bezug auf die dem **RWP** zugrundeliegenden **DGL** mehr oder weniger gut geeignet sein können. Für steife **DGL** ist z. B. die Radau-Kollokation numerisch günstig. [8, vgl. S. 290-293]

Neben der Wahl der M Kollokationspunkte auf den Teilintervallen ist auch die Anzahl N der Teilintervalle ein relevanter Verfahrensparameter von **KKV**. Ein nennenswerter Spezialfall sind dabei die *pseudo-spektralen Kollokationsverfahren*, bei denen $N = 1$ gewählt wird, sodass keine Unterteilung des Gesamtintervalls stattfindet. Die Lösung des **RWP** wird dabei mit einem einzigen Interpolationspolynom ausreichend hohen Grades angenähert. Für **RWP** mit hinreichend glatten Lösungen können pseudo-spektrale **KKV** genaue und effiziente Lösungsverfahren sein. [8, S. 295f.]

2.6 Statische Optimierung

Die allgemeine Problemstellung der statischen Optimierung

$$\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}), \quad \mathcal{X} = \{\mathbf{x} \mid \mathbf{g}(\mathbf{x}) = \mathbf{0}, \mathbf{h}(\mathbf{x}) \leq \mathbf{0}\} \quad (2.35)$$

³Da die Kollokationspunkte c_i in $(0, 1)$ liegen, wählt man die Nullstellen des auf das Intervall $[0, 1]$ verschobenen Legendre-Polynoms

⁴Da die einem Rekursionsschritt eines impliziten **RKV** entsprechende Quadraturformel das Integral über einem Teilintervall approximiert, gilt dieser Genauigkeitsgrad folglich nur für die Werte der Lösung $\mathbf{x}(t_i)$ an den Teilintervallgrenzen, nicht jedoch für Werte innerhalb der Teilintervalle.

entspricht der Minimierung einer *Zielfunktion* $f : \mathbb{R}^n \rightarrow \mathbb{R}$ mehrerer Variablen $\mathbf{x} \in \mathbb{R}^n$ unter *Gleichungsnebenbedingungen (GNB)* $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ und *Ungleichungsnebenbedingungen (UNB)* $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^p$ [22, S. 11]. Für den hier betrachteten allgemeinen Fall nichtlinearer Zielfunktionen f wird das Problem auch häufig als *nichtlineares Programm (engl. nonlinear program) (NLP)* bezeichnet. Die wichtigsten numerischen Verfahren zur statischen Optimierung sind die Verfahren der sequentiellen quadratischen Programmierung (SQP-Verfahren) [22, Kap. 5.4.4] sowie die Innere-Punkte-Verfahren [22, Kap. 5.4.5].

2.7 Nichtlineare dynamische Systeme

In dieser Arbeit werden zeitinvariante nichtlineare dynamische Systeme mit jeweils zeitabhängigem⁵ Zustand \mathbf{x} und Eingang \mathbf{u} in allgemeiner Zustandsraumdarstellung

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad \mathbf{x}(t) \in \mathbb{R}^n, \quad \mathbf{u}(t) \in \mathbb{R}^m, \quad t \in \mathbb{R} \quad (2.36)$$

betrachtet.

Mechanische Systeme

Die Bewegungsgleichungen eines mechanischen Systems können mithilfe des *Lagrange-Formalismus* bestimmt werden. Ein mechanisches System mit den Konfigurationskoordinaten $\boldsymbol{\theta} \in \mathbb{R}^v$ und dem Vektor der von $w \leq v$ unabhängigen Aktoren eingprägten Kräfte $\boldsymbol{\tau} \in \mathbb{R}^w$ wird durch die Bewegungsgleichungen in Matrix-Darstellung

$$\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{K}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \mathbf{B}(\boldsymbol{\theta})\boldsymbol{\tau} \quad (2.37)$$

beschrieben und heißt *vollständig aktuiert*, falls $w = v$ und *unteraktuiert*, falls $w < v$. [19, S. 7-9]

In beiden Fällen kann mit $\mathbf{x} := \begin{pmatrix} \boldsymbol{\theta}^T, \dot{\boldsymbol{\theta}}^T \end{pmatrix}^T \in \mathbb{R}^{2v}$ und $\mathbf{u} := \boldsymbol{\tau}$ eine Zustandsraumdarstellung

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{pmatrix} \dot{\boldsymbol{\theta}} \\ \mathbf{M}^{-1}(\boldsymbol{\theta}) \left(\mathbf{B}(\boldsymbol{\theta})\boldsymbol{\tau} - \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) - \mathbf{K}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \right) \end{pmatrix} \quad (2.38)$$

angegeben werden.

Unteraktuierte mechanische Systeme, welche in dieser Arbeit von besonderem Interesse sind, können in den meisten Fällen in ein vollständig aktuiertes und ein damit verkoppeltes nicht aktuiertes Teilsystem aufgespalten werden. Die Konfigurationskoordinaten

⁵Auf die explizite Angabe der Abhängigkeit vom Zeitpunkt t wird im Folgenden zur Verbesserung der Lesbarkeit meist verzichtet.

werden dabei mit $\boldsymbol{\theta} = (\mathbf{p}^T, \mathbf{q}^T)^T$ in „passive“ Koordinaten $\mathbf{p} \in \mathbb{R}^{v-w}$ und „aktive“ Koordinaten $\mathbf{q} \in \mathbb{R}^w$ entsprechend der Darstellung

$$\begin{pmatrix} \mathbf{M}_{11}(\boldsymbol{\theta}) & \mathbf{M}_{12}(\boldsymbol{\theta}) \\ \mathbf{M}_{21}^T(\boldsymbol{\theta}) & \mathbf{M}_{22}(\boldsymbol{\theta}) \end{pmatrix} \begin{pmatrix} \ddot{\mathbf{p}} \\ \ddot{\mathbf{q}} \end{pmatrix} + \begin{pmatrix} \mathbf{C}_1(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \\ \mathbf{C}_2(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \end{pmatrix} + \begin{pmatrix} \mathbf{K}_1(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \\ \mathbf{K}_2(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \boldsymbol{\tau} \end{pmatrix} \quad (2.39)$$

unterteilt. [19, S. 10]

Definiert man statt den eingprägten Kräften $\boldsymbol{\tau}$ die Beschleunigungen $\ddot{\mathbf{q}} =: \mathbf{u}$ in den aktuierten Koordinaten als neuen Eingang \mathbf{u} , so lässt sich mit $\mathbf{x} := (\mathbf{p}^T, \mathbf{q}^T, \dot{\mathbf{p}}^T, \dot{\mathbf{q}}^T)^T$ eine partiell linearisierte Zustandsraumdarstellung

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{pmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{q}} \\ -\mathbf{M}_{11}^{-1}(\boldsymbol{\theta}) (\mathbf{C}_1(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{K}_1(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})) - \mathbf{M}_{11}^{-1}(\boldsymbol{\theta}) \mathbf{M}_{12}(\boldsymbol{\theta}) \mathbf{u} \\ \mathbf{u} \end{pmatrix} \quad (2.40)$$

angeben. Diese Darstellung besitzt den Vorteil, dass sich für konkrete Systeme meist wesentlich einfachere Ausdrücke ergeben. [19, S. 11] Die Bezeichnung als *partielle Linearisierung* versteht sich insofern, als dass das aktuierte Teilsystem im Unterschied zu (2.38) ein lineares Teilsystem darstellt.

Der Vektor der Stellkräfte ergibt sich aus

$$\boldsymbol{\tau} = (\mathbf{M}_{22} - \mathbf{M}_{12}^T \mathbf{M}_{11}^{-1} \mathbf{M}_{12}) \mathbf{u} - \mathbf{M}_{12}^T \mathbf{M}_{11}^{-1} (\mathbf{C}_1 + \mathbf{K}_1) + \mathbf{C}_2 + \mathbf{K}_2. \quad (2.41)$$

in Abhängigkeit des neuen Eingangs \mathbf{u} . [19, S. 11]

Beispiel 2.1 (Partielle Linearisierung). *Die Unterschiede in Bezug auf die Zustandsdarstellung sollen am Beispiel des in Abbildung 2.2 dargestellten Einfach-Wagen-Pendel-Systems verdeutlicht werden. Dabei handelt es sich um einen horizontal verschiebbaren Wagen mit der Masse m_0 , welcher über ein Drehgelenk mit einem Pendel verbunden ist. Das Pendel besteht aus einem masselosen Stab der Länge s_1 und einer Punktmasse m_1 .*

Es handelt sich offensichtlich um ein unteraktuiertes mechanisches System, da es mit der Kraft F , welche auf den Wagen wirkt, nur eine eingprägte Kraft jedoch mit der Wagenposition s und dem Pendelwinkel θ zwei Konfigurationskoordinaten gibt. Der Wagen stellt somit das vollständig aktuierte Teilsystem und das Pendel das nicht aktuierte Teilsystem dar.

Aus den Bewegungsgleichungen des Systems mit $\boldsymbol{\theta} := (\theta, s)^T$ und $\tau := F$ folgt mit $\mathbf{x} := (\theta, s, \omega, v)^T := (\theta, s, \dot{\theta}, \dot{s})^T$ sowie $u := F$ die Zustandsraumdarstellung

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u) = \begin{pmatrix} \omega \\ v \\ \frac{m_1 \sin(\theta) (g \cos(\theta) - s_1 \omega^2) + F}{m_0 + m_1 \sin^2(\theta)} \\ \frac{((m_0 + m_1)g - m_1 s_1 \omega^2 \cos(\theta)) \sin(\theta) + F \cos(\theta)}{s_1 (m_0 + m_1 \sin^2(\theta))} \end{pmatrix} \quad (2.42)$$

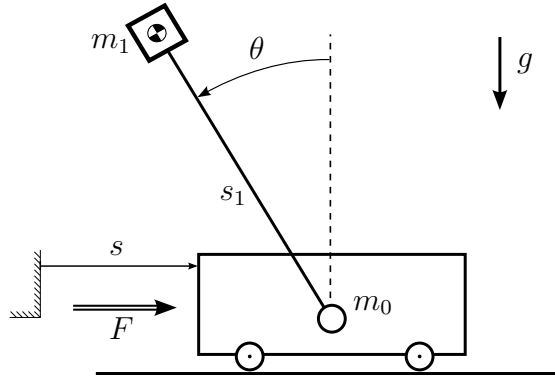


Abbildung 2.2 – Schematische Darstellung des Einfach-Wagen-Pendel-Systems [19, Abb. 4.3, S. 38].

und aus $p := \theta$ und $q := s$ mit in diesem Fall unverändertem \mathbf{x} sowie $u := a := \ddot{s}$ die partiell linearisierte Zustandsraumdarstellung

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u) = \begin{pmatrix} \omega \\ v \\ \frac{g}{s_1} \sin(\theta) + \frac{1}{s_1} \cos(\theta) a \\ a \end{pmatrix}, \quad (2.43)$$

welche gegenüber der Darstellung (2.42) offensichtlich eine wesentliche Vereinfachung bedeutet.

2.8 Optimale Steuerung

Die optimale Steuerung dynamischer Systeme ist ein spezielles Problem der dynamischen Optimierung. Ziel der optimalen Steuerung ist die Bestimmung einer optimalen Eingangstrajektorie zur Steuerung eines dynamischen Systems ausgehend von einem Anfangszustand. Optimalität bedeutet hierbei die Minimierung eines Kostenfunktional, welches das Ziel der Steuerung formal definiert. Zusätzlich kann auch die Einhaltung von Gleichungs- und Ungleichungsnebenbedingungen gefordert werden. Insgesamt spricht man von einem *Optimalsteuerungsproblem (OSP)*. [13, S. 82]

Ein System der Form (2.36) soll unter Minimierung eines skalaren Kostenfunktional J von einem Anfangszustand \mathbf{x}_0 zur Anfangszeit t_0 in einen Endzustand $\mathbf{x}(t_f)$ überführt werden. Dies soll unter Einhaltung von Gleichungsbedingungen für den Endzustand und Ungleichungsbedingungen für Zustand und Eingang erfolgen. Betrachtet wird also ein

OSP der Form

$$\min_{\mathbf{u}(\cdot)} J(\mathbf{u}) = V(\mathbf{x}(t_f)) + \int_{t_0}^{t_f} l(\mathbf{x}, \mathbf{u}) \, dt \quad (2.44a)$$

$$\text{u. B. v. } \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \, t \in [t_0, t_f] \subset \mathbb{R}, \, \mathbf{x}(t_0) = \mathbf{x}_0 \quad (2.44b)$$

$$\mathbf{g}(\mathbf{x}(t_f)) = \mathbf{0} \quad (2.44c)$$

$$\mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{0}, \, \forall t. \quad (2.44d)$$

Das Kostenfunktional J setzt sich aus einer Endzustandsbewertung V und dem Integralanteil l zusammen. Dabei ist der Endzustand $\mathbf{x}(t_f)$ und im Allgemeinen auch die Endzeit t_f frei. [13, S. 82]

Im Rahmen dieser Arbeit ist jedoch in erster Linie der Fall einer festen Endzeit t_f und eines festen Endzustandes \mathbf{x}_f relevant. Da in diesem Fall die Bedingung $\mathbf{x}(t_f) = \mathbf{x}_f$ erfüllt sein muss, entfällt die Endzustandsbewertung V . Außerdem sollen als Ungleichungsbedingungen nur jeweils Beschränkungen von Zustand und Eingang nach unten und oben gelten. Dies entspricht einem OSP der Form

$$\min_{\mathbf{u}(\cdot)} J(\mathbf{u}) = \int_{t_0}^{t_f} l(\mathbf{x}, \mathbf{u}) \, dt \quad (2.45a)$$

$$\text{u. B. v. } \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \, t \in [t_0, t_f] \subset \mathbb{R}, \, \mathbf{x}(t_0) = \mathbf{x}_0, \, \mathbf{x}(t_f) = \mathbf{x}_f \quad (2.45b)$$

$$\mathbf{x}(t) \in \mathcal{X}, \, \forall t, \, \mathcal{X} = \{\mathbf{x} | \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max}\} \quad (2.45c)$$

$$\mathbf{u}(t) \in \mathcal{U}, \, \forall t, \, \mathcal{U} = \{\mathbf{u} | \mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}\}. \quad (2.45d)$$

Aus der Anwendung der Variationsrechnung (siehe [13, S. 84-87, 89f.]) folgen die notwendigen Optimalitätsbedingungen, welche zur Lösung eines OSP herangezogen werden können. Die Systemgleichungen werden dazu als dynamische GNB mithilfe von Lagrange-Multiplikatoren $\boldsymbol{\lambda}(t) \in \mathbb{R}^n$ im Kostenfunktional berücksichtigt:

$$\bar{J}(\mathbf{u}) = \int_{t_0}^{t_f} \left(l(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}^T(\mathbf{f}(\mathbf{x}, \mathbf{u}) - \dot{\mathbf{x}}) \right) \, dt \quad (2.46)$$

Durch Einführen der Hamilton-Funktion $H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}) = l(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}^T \mathbf{f}(\mathbf{x}, \mathbf{u})$ lassen sich die notwendigen Optimalitätsbedingungen für ein unbeschränktes OSP (2.45) mit $\mathcal{X} = \mathbb{R}^n$, $\mathcal{U} = \mathbb{R}^m$ in kompakter Form darstellen: [13, S. 92]

$$\dot{\mathbf{x}} = \left(\frac{\partial H}{\partial \boldsymbol{\lambda}} \right)^T = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (2.47a)$$

$$\dot{\boldsymbol{\lambda}} = - \left(\frac{\partial H}{\partial \mathbf{x}} \right)^T \quad (2.47b)$$

$$\mathbf{0} = \left(\frac{\partial H}{\partial \mathbf{u}} \right)^T \quad (2.47c)$$

Lässt sich \mathbf{u} durch Auflösen von (2.47c) explizit als Funktion

$$\varphi : \mathbf{u} \mapsto \mathbf{z} \text{ mit } \mathbf{z} := (\mathbf{x}^T, \boldsymbol{\lambda}^T)^T \in \mathbb{R}^{2n} \quad (2.48)$$

darstellen, so ergibt sich durch Einsetzen von $\varphi(\mathbf{z})$ in (2.47a) und (2.47b) mit

$$\dot{\mathbf{z}} = \left((\mathbf{f}(\mathbf{z}, \varphi(\mathbf{z})))^T, -\frac{\partial H(\mathbf{z}, \varphi(\mathbf{z}))}{\partial \mathbf{x}} \right)^T, \quad t \in [t_0, t_f] \subset \mathbb{R}, \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_f \quad (2.49)$$

ein RWP der Form (2.23), dessen Lösung die optimale Trajektorie $\mathbf{z}^*(t)$ liefert. Durch Einsetzen in $\varphi(\mathbf{z})$ ergibt sich schließlich die optimale Eingangstrajektorie $\mathbf{u}^*(t)$. [13, S. 98]

Allerdings ist zu beachten, dass eine auf diese Weise bestimmte Trajektorie nicht zwingend global optimal ist, da sie lediglich die notwendigen Optimalitätsbedingungen (2.47) erfüllt. Falls das RWP (2.49) eine eindeutige Lösung besitzt, so sind diese Bedingungen jedoch auch hinreichend und die bestimmte Trajektorie ist global optimal. [13, vgl. S. 98]

Die Erweiterung der beschriebenen Vorgehensweise auf ein OSP mit Eingangsbeschränkungen der Form (2.45d) ist durch Anwendung von *Pontrjagins Maximumprinzip* möglich [13, S. 109ff.]. Eine Berücksichtigung von Zustandsbeschränkungen der Form (2.45c) ist mit dem vorgestellten Ansatz jedoch nicht ohne Weiteres möglich [13, S. 119]. In [14] werden Methoden zur Berücksichtigung von Eingangs- und Ausgangsbeschränkungen beim Vorsteuerungsentwurf beschrieben, welche in abgewandelter Form möglicherweise auch auf OSP der Form (2.45) angewendet werden können.

Kapitel 3

Verfahren zur Trajektorienplanung

Das folgende Kapitel befasst sich mit verschiedenen Verfahren zur Trajektorienplanung auf einer theoretischen Ebene. Hierzu wird das Problem der Trajektorienplanung zunächst formal definiert, um anschließend die verschiedenen Grundansätze zur Lösung aufzuzeigen und anschließend zu vertiefen. Aufgrund ihrer Bedeutung werden die Schieß- und Kollokationsverfahren und deren Implementierung am ausführlichsten behandelt. Auf weitere Verfahren wird nur kurz eingegangen. Schließlich erfolgt ein theoretischer Vergleich der implementierten Schieß- und Kollokationsverfahren, welcher zugleich als Vorbetrachtung zum empirischen Vergleich anhand von Beispielproblemen dient.

3.1 Problemdefinition

Ziel der Trajektorienplanung ist grundsätzlich die Bestimmung einer Eingangstrajektorie $\mathbf{u}(\cdot)$, welche ein System (2.36) von einem gegebenen Anfangszustand \mathbf{x}_0 zur Anfangszeit t_0 in einen gewünschten Endzustand \mathbf{x}_f zur Endzeit t_f überführt. Die Forderung nach Einhaltung allgemeiner GNB und UNB sowie insbesondere von Bedingungen an den Anfangs- und Endwert des Eingangssignals wird in dieser Arbeit nicht betrachtet. Es wird lediglich die Einhaltung von Zustands- und Eingangsbeschränkungen gefordert. Insgesamt entspricht dies einem RWP

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad t \in [t_0, t_f] \subset \mathbb{R}, \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_f \quad (3.1a)$$

$$\mathbf{x}(t) \in \mathcal{X}, \quad \forall t, \quad \mathcal{X} = \{\mathbf{x} \mid \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max}\} \quad (3.1b)$$

$$\mathbf{u}(t) \in \mathcal{U}, \quad \forall t, \quad \mathcal{U} = \{\mathbf{u} \mid \mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}\} \quad (3.1c)$$

mit UNB und wird in dieser Arbeit als *Überführungsproblem (ÜFP)* bezeichnet. Bei einem OSP der Form (2.45) fordert man zusätzlich Optimalität im Sinne der Minimierung eines Kostenfunktional J . Insofern kann man ÜFP als Spezialfall von OSP mit $J = 0$ betrachten. Dennoch besitzt die Unterscheidung zwischen ÜFP und OSP ihre Berechtigung. Zur Beschreibung der Gesamtheit der im Rahmen dieser Arbeit

betrachteten Probleme wird daher der Oberbegriff *Trajektorienplanungsproblem (TPP)* verwendet.

3.2 Verfahren im Überblick

Das zentrale mathematische Problem bei den betrachteten TPP ist das RWP (2.45b) bzw. (3.1a), welches als RWP mit unendlich vielen freien Parametern aufgefasst werden kann. Daher ist es wenig verwunderlich, dass Verfahren zur Trajektorienplanung im engeren Sinne Verfahren zur Lösung von RWP sind. Die bedeutendsten Methoden sind die in Abschnitt 2.5 vorgestellten Schieß- und Kollokationsverfahren. Um diese Verfahren anwenden zu können, muss das RWP mit unendlich vielen freien Parametern zunächst in ein RWP mit einer endlichen Anzahl von unbekannten Variablen wie etwa (2.23) bzw. (2.24) überführt werden.

Abbildung 3.1 veranschaulicht die im Folgenden vorgestellte allgemeine Vorgehensweise zur Trajektorienplanung mit Schieß- und Kollokationsverfahren. Dabei erweist sich insbesondere die Zerlegung der auftretenden mathematischen Probleme in ihre einzelnen Teilprobleme als aufschlussreich. Wie in der Abbildung angedeutet wird, kann ein ÜFP durch Formulierung eines Kostenfunktional J in ein OSP umgewandelt werden.

Mithilfe eines Kostenfunktional können zusätzliche Anforderungen wie z. B. Energieeffizienz bei der Trajektorienplanung berücksichtigt werden. Eine Berücksichtigung derartiger Anforderungen ist jedoch nicht die Hauptmotivation für die zusätzliche Behandlung von OSP im Rahmen dieser Arbeit. Vielmehr ist die Forderung nach Optimalität auch ein Hilfsmittel beim Entwurf einer Trajektorie, da diese Forderung zu einer Reduktion der Lösungsmöglichkeiten oder sogar zur Eindeutigkeit der Lösung führt. Außerdem stehen zur Lösung von OSP mehr Verfahren zu Verfügung. Somit könnte die Formulierung eines ÜFP als OSP die Wahrscheinlichkeit erhöhen, dass jenes gelöst werden kann. Zunächst wird die Vorgehensweise zur Lösung von OSP beschrieben.

Lösung von OSP

In der Literatur werden fast ausschließlich Verfahren zur Lösung von OSP behandelt. Diesbezüglich bieten [7, 23] einen umfassenden Überblick. Eine ausführliche Beschreibung von Lösungsverfahren sowie deren mathematischen Grundlagen erfolgt mit jeweils unterschiedlichem Fokus in [6, 8, 22, 13]. Bei der Lösung von OSP sind zwei verschiedene Grundansätze zu unterscheiden [22, S. 387].

Direkte Verfahren überführen das zugrundeliegende RWP durch eine Diskretisierung des Einganges und ggf. auch des Zustandes in ein RWP mit freien Parametern der Form (2.24). Bei Schieß- und Kollokationsverfahren erfolgt die Lösung dieses RWP durch Aufstellen eines NLGS mithilfe von Verfahren zur Lösung von AWP. Fasst man dieses NLGS als GNB auf, erhält man zusammen mit dem Kostenfunktional und den UNB ein NLP

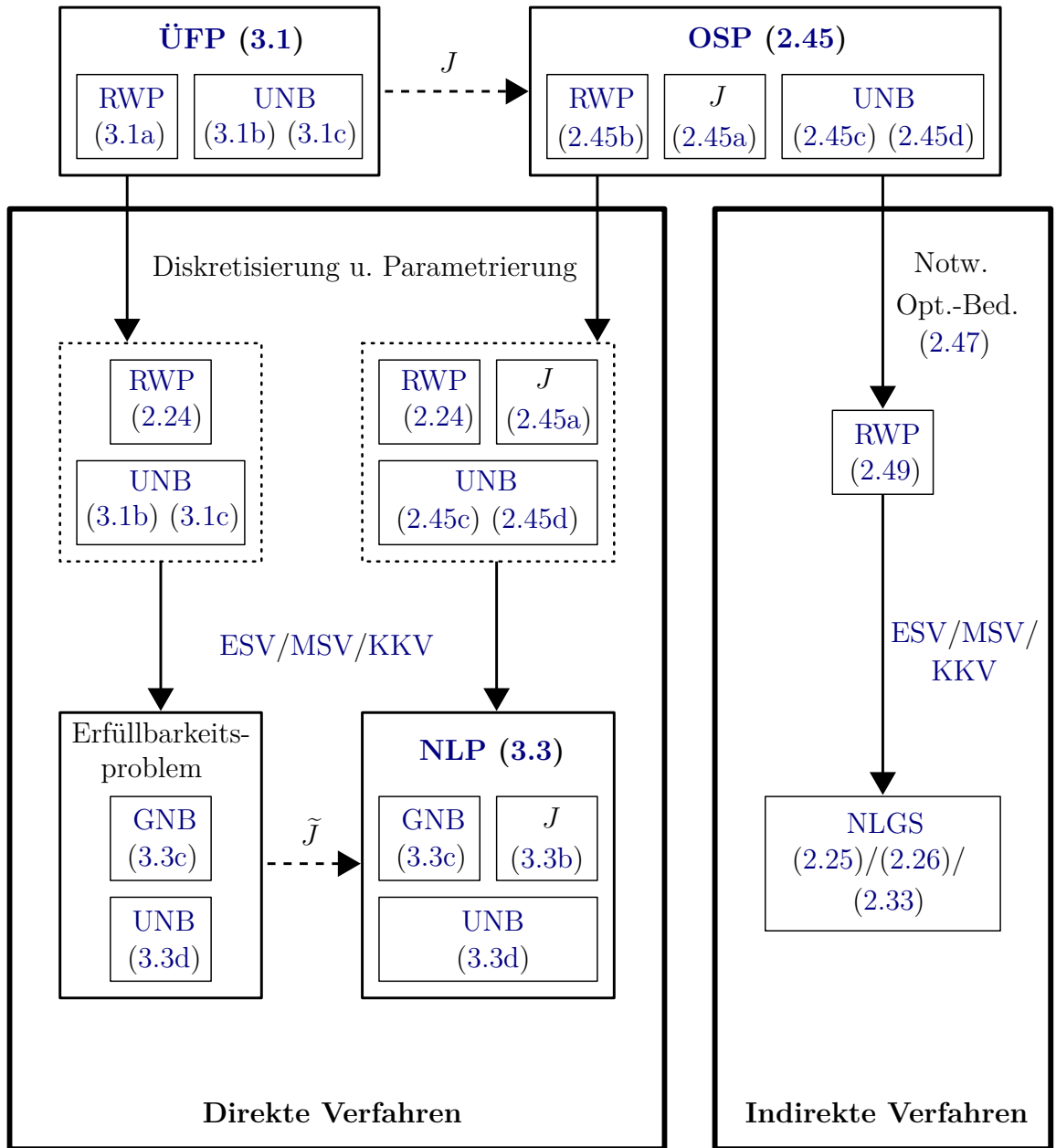


Abbildung 3.1 – Schematische Darstellung der Vorgehensweise zur Trajektorienplanung mit Schieß- und Kollokationsverfahren.

der Form (2.35). Zur Lösung des entstehenden **NLP** existieren verschiedene numerische Verfahren (siehe Abschnitt 2.6). Hierzu muss bei den Schießverfahren im Allgemeinen bei jeder Iteration ein **AWP** gelöst und eine Sensitivitätsanalyse durchgeführt werden [8, S. 251ff.]. Im Unterschied dazu erfolgt bei Kollokationsverfahren keine explizite Lösung von **AWP**, sondern die Formulierung eines **NLGS** wie bei der Anwendung von impliziten

Verfahren zur Lösung von AWP (siehe Abschnitt 2.5). Daher kann man nur im Falle von KKV von einer tatsächlichen Umwandlung in ein NLP sprechen. Diesen Vorgang der Umwandlung nennt man auch *Transkription* und die direkten KKV deswegen auch *Transkriptionsverfahren* [8, S. 287]. Da bei direkten Verfahren zuerst eine Diskretisierung und anschließend eine Optimierung der Parameter vorgenommen wird, spricht man auch von „*first discretize, then optimize*“-Verfahren [22, S. 387].

Indirekte Verfahren lösen das Optimierungsproblem insofern indirekt, als dass ein Hilfsproblem gelöst wird. Mithilfe der mit dem Variationsansatz bestimmbaren notwendigen Optimalitätsbedingungen (siehe Kapitel 2.8) erhält man mit (2.49) ein RWP mit gemischten Randbedingungen der Form (2.23), dessen Lösung die Optimalitätsbedingungen zwangsläufig erfüllt. Man kann daher von einer diesem RWP innewohnenden (lokalen) Optimalität sprechen. Die Anwendung von Schieß- oder Kollokationsverfahren auf dieses RWP führt lediglich auf ein NLGS. Analog zu direkten Verfahren spricht man auch von „*first optimize, then discretize*“-Verfahren [22, S. 387], wobei eine Diskretisierung streng genommen nur bei indirekten MSV und KKV vorgenommen wird.

Lösung von ÜFP

Zur Lösung von ÜFP kann man weitestgehend analog zu den direkten Verfahren für OSP vorgehen.¹ Aufgrund des fehlenden Kostenfunktional ergibt sich allerdings kein NLP (2.35), sondern lediglich dessen GNB und UNB. Wie Abbildung 3.1 andeutet, kann ein solches *Erfüllbarkeitsproblem* mithilfe eines Kostenfunktional \tilde{J} in ein NLP umformuliert werden. Ein solches Kostenfunktional quantifiziert die Abweichung von den GNB, verhindert ggf. die Verletzung von den UNB durch Strafterme und dient somit einem anderen Zweck als das Kostenfunktional eines OSP, von welchem es folglich zu unterscheiden ist.

Bemerkung zu differentiell flachen Systemen

Abschließend sei der Vollständigkeit halber bemerkt, dass für die Klasse der differentiell flachen Systeme kein RWP gelöst, sondern lediglich eine Trajektorie in den Komponenten des flachen Ausgangs entworfen werden muss, da alle anderen Systemgrößen sich explizit durch die Komponenten des flachen Ausgangs und dessen Ableitungen darstellen lassen (siehe [25]). Diese Aufgabe entspricht einem Interpolationsproblem, dessen Lösung im Vergleich zum RWP deutlich einfacher ist. Ein Ansatz zur optimalen Steuerung differentiell flacher Systeme scheint hingegen weniger offensichtlich zu sein. Dennoch werden die Methoden der flachheitsbasierten Trajektorienplanung im Rahmen dieser Arbeit nicht behandelt.

Obwohl die in dieser Arbeit vorgestellten Verfahren zur Trajektorienplanung für differentiell flacher Systeme wahrscheinlich aufwendiger als flachheitsbasierte Methoden sind, spricht selbstverständlich nichts gegen deren Anwendung für derartige Systeme. Im Falle von Systemen, bei denen das Vorliegen der Flachheitseigenschaft ungeklärt ist,

¹Der Begriff der direkten Verfahren ist nur im Kontext von OSP gebräuchlich.

können die behandelten Verfahren zudem eine Alternative zu einem ggf. aufwändigen und evtl. ergebnislosen bzw. negativen Flachheitsnachweis darstellen.

3.3 Direkte Verfahren

Um Verfahren zur Lösung von AWP auf ein OSP (2.45) anwenden zu können, muss das Eingangssignal zu jedem Zeitpunkt bekannt sein. Direkte Verfahren lösen dieses Problem durch eine Parametrierung des Eingangssignals, d. h. die Beschreibung des Eingangssignals mit einer endlichen Anzahl von Parametern. Ein möglicher Ansatz ist die Diskretisierung des betrachteten Zeitintervalls $[t_0, t_f]$ und ein stückweise konstantes Eingangssignal

$$\mathbf{u}_i = \mathbf{p}_i, \quad t_i < t < t_{i+1}, \quad i = 0, \dots, N-1, \quad t_N := t_f, \quad \mathbf{p} = \left(\mathbf{p}_0^T, \dots, \mathbf{p}_{N-1}^T \right)^T \quad (3.2)$$

mit den Parametern $\mathbf{p} \in \mathbb{R}^{mN}$ und N Teilintervallen. Weiterhin besteht auch die Möglichkeit das Eingangssignal auf den Teilintervallen jeweils als Polynom anzusetzen. Dabei kann zusätzlich die Stetigkeit des Eingangssignals, d. h. die Stetigkeit des Eingangssignals an den Teilintervallgrenzen, gefordert werden. Zunächst wird jedoch von einem stückweise konstanten Eingangssignal ausgegangen.

Eine Parametrierung des Eingangssignals überführt ein RWP der Form (2.45b) in ein RWP der Form (2.24). Das allgemeine Vorgehen zur Lösung eines solchen RWP wurde bereits in Abschnitt 2.5 beschrieben. Im Falle des OSP (2.45) ist jedoch speziell eine Lösung gesucht, welche unter Berücksichtigung der UNB (2.45c) und (2.45d) das Kostenfunktional (2.45a) minimiert. Fasst man das bei der Lösung des RWP mit Schieß- und Kollokationsverfahren entstehende NLGS als GNB auf, wird deutlich, dass diese Aufgabe einem NLP der Form (2.35) entspricht. Dabei variiert das sich ergebende NLGS je nach Verfahren (siehe Abschnitt 2.5). Fasst man die Parameter des Eingangssignals \mathbf{p} als Teil eines Vektors unbekannter Variablen \mathbf{v} auf, so ergibt sich für alle Verfahren allgemein ein NLP

$$\min_{\mathbf{v} \in \mathcal{V}} J(\mathbf{v}), \quad \mathcal{V} = \{ \mathbf{v} \mid \mathbf{g}(\mathbf{v}) = \mathbf{0}, \mathbf{h}(\mathbf{v}) \leq \mathbf{0} \} \quad (3.3a)$$

$$J(\mathbf{v}) = \int_{t_0}^{t_f} l(\mathbf{x}(\mathbf{v}), \mathbf{u}(\mathbf{v})) \, dt \quad (3.3b)$$

$$\mathbf{g}(\mathbf{v}) \text{ je nach Verfahren gemäß (2.25), (2.26) bzw. (2.33)} \quad (3.3c)$$

$$\mathbf{h}(\mathbf{v}) = \begin{pmatrix} \mathbf{x}_{\min} - \mathbf{x}(\mathbf{v}) \\ \mathbf{x}(\mathbf{v}) - \mathbf{x}_{\max} \\ \mathbf{u}_{\min} - \mathbf{u}(\mathbf{v}) \\ \mathbf{u}(\mathbf{v}) - \mathbf{u}_{\max} \end{pmatrix}. \quad (3.3d)$$

Zur Vollständigkeit muss daher nur noch die konkrete Gestalt des Variablen-Vektors \mathbf{v} für die einzelnen Verfahren angegeben werden. Schließlich wird noch kurz auf direkte Verfahren zur Lösung zeitdiskreter **OSP** und mit der differentiellen Dynamischen Programmierung auf ein weiteres direktes Verfahren zur Lösung kontinuierlicher **OSP** eingegangen.

Direktes Einfach-Schießverfahren (DESV)

Bei den **DESV** besteht der Vektor der Problemvariablen \mathbf{v} lediglich aus den Parametern des Einganges aus Gleichung (3.2):

$$\mathbf{v} := \mathbf{p} \in \mathbb{R}^{mN} \quad (3.4)$$

Direktes Mehrfach-Schießverfahren (DMSV)

Im Vergleich zu den **DESV** treten im Falle der **DMSV** zusätzlich die an den $N - 1$ Teilintervallgrenzen vorgegebenen Anfangswerte $\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_{N-1}^T$ als Problemvariablen auf:

$$\mathbf{v} := (\mathbf{p}^T, \hat{\mathbf{x}}_1^T, \dots, \hat{\mathbf{x}}_{N-1}^T)^T \in \mathbb{R}^{(m+n)N-n} \quad (3.5)$$

Direktes Kollokationsverfahren (DKKV)

Bei den **DKKV** müssen im Vergleich zu den **DMSV** außerdem noch die in jedem der N Teilintervalle an den M Kollokationspunkten vorgegebenen Werte des Zustandes im Vektor der Problemvariablen berücksichtigt werden (siehe Abschnitt 2.5):

$$\mathbf{v} := (\mathbf{p}^T, \hat{\mathbf{x}}_{01}^T, \dots, \hat{\mathbf{x}}_{0M}^T, \hat{\mathbf{x}}_{10}^T, \dots, \hat{\mathbf{x}}_{1M}^T, \dots, \hat{\mathbf{x}}_{(N-1)0}^T, \dots, \hat{\mathbf{x}}_{(N-1)M}^T)^T \in \mathbb{R}^{(m+n(M+1))N-n} \quad (3.6)$$

Weitere direkte Verfahren

Für das **OSP** (2.45) kann man ein zeitdiskretes Ersatzproblem

$$\min_{\mathbf{u}_0, \dots, \mathbf{u}_{N-1}} \sum_{i=0}^{N-1} l(\mathbf{x}_i, \mathbf{u}_i) \quad (3.7a)$$

$$\text{u. B. v. } h\varphi(t_i, \mathbf{x}_i, \mathbf{u}_i) - \mathbf{x}_{i+1} = \mathbf{0}, \quad \mathbf{x}_0, \mathbf{x}_N = \mathbf{x}_f \quad (3.7b)$$

$$\mathbf{x}_{\min} \leq \mathbf{x}_i \leq \mathbf{x}_{\max}, \quad \forall i \quad (3.7c)$$

$$\mathbf{u}_{\min} \leq \mathbf{u}_i \leq \mathbf{u}_{\max}, \quad \forall i \quad (3.7d)$$

mit den Teilintervallen $[t_i, t_{i+1}]$, $i = 0, 1, \dots, N - 1$, $t_N := t_f$, $h := t_{i+1} - t_i$ und einem geeigneten Einschrittverfahren mit der Verfahrensfunktion φ (siehe Abschnitt 2.4) formulieren und beispielsweise durch Anwendung der *Dynamischen Programmierung* lösen. Dies ist aufgrund des „Fluchs der Dimensionen“ jedoch nur für Probleme kleiner Dimension praktikabel [7]. Dies gilt folglich umso mehr für die naive *vollständige Enumeration*. Daher werden diese beiden Verfahren hier nicht näher erläutert.

Schließlich ist die vorgeschlagene Umwandlung in ein zeitdiskretes Ersatzproblem je nach Einschnittverfahren und Schrittweite entweder mit einem erheblichen Diskretisierungsfehler oder einer sehr großen Anzahl von Optimierungsvariablen verbunden. Angesichts der Existenz von Verfahren zur Lösung des kontinuierlichen Originalproblems besteht somit kein Anlass eine solche Umwandlung überhaupt durchzuführen. Die prinzipielle Möglichkeit sollte an dieser Stelle dennoch nicht unerwähnt bleiben.

Differentielle Dynamische Programmierung

Die *differentielle Dynamische Programmierung* kann als direktes Einfach-Schießverfahren zweiter Ordnung aufgefasst werden und erlaubt im Gegensatz zur klassischen Dynamischen Programmierung auch die Lösung kontinuierlicher *OSP*. Das gesuchte Eingangssignal wird ausgehend von einer Schätzung iterativ angepasst. Dabei wird abwechselnd durch Vorwärtsintegration evaluiert und im Rahmen einer Rückwärtsrechnung unter Ausnutzung des Optimalitätsprinzips eine optimale Trajektorie durch Minimierung einer lokalen quadratischen Näherung bestimmt. [30]

3.4 Indirekte Verfahren

Bei indirekten Verfahren werden die mithilfe des Variationsansatzes bestimmbar notwendigen Optimalitätsbedingungen zur Lösung des *OSP* herangezogen. Aus diesen Bedingungen lässt sich das *RWP* (2.49) als Hilfsproblem formulieren, dessen Lösung jenen Bedingungen genügt. Voraussetzung hierfür ist, dass sich aus (2.47c) eine Funktion $\mathbf{u} = \boldsymbol{\varphi}(\mathbf{z})$ (2.48) in expliziter Darstellung gewinnen lässt. Von den Randwerten $\mathbf{z}_0, \mathbf{z}_f$ des *RWP* sind dabei $\mathbf{x}_0, \mathbf{x}_f$ gegeben und $\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_f$ unbekannt. Es handelt sich somit um ein *RWP* mit gemischten Randbedingungen der allgemeinen Form (2.23).

Bemerkenswert ist, dass die Eingangstrajektorie aufgrund der Zusammenhänge (2.49) und (2.48) bereits allein durch den Anfangswert $\boldsymbol{\lambda}_0$ eindeutig beschrieben wird. Da das *RWP* (2.49) unabhängig vom Eingang ist, muss zudem keine Parametrierung desselben vorgenommen werden. Die Lösung eines solchen *RWP* mit Schieß- und Kollokationsverfahren wurde bereits in Abschnitt 2.5 behandelt. Für diese Verfahren werden daher nun lediglich noch die konkreten Vektoren \mathbf{v} unbekannter Variablen angegeben. Anschließend wird noch kurz auf zwei weitere indirekte Verfahren eingegangen.

Indirektes Einfach-Schießverfahren (IESV)

Bei den *IESV* wird das *RWP* (2.49) lediglich durch eine iterative Anpassung des Anfangswertes $\boldsymbol{\lambda}_0$ gelöst:

$$\mathbf{v} := \boldsymbol{\lambda}_0 \in \mathbb{R}^n \quad (3.8)$$

Indirektes Mehrfach-Schießverfahren (IMSV)

Die IMSV variieren im Vergleich zu den IESV zusätzlich die jeweils an den $N - 1$ Teilintervallgrenzen vorgegebenen Anfangswerte von \mathbf{x} und $\boldsymbol{\lambda}$ zur Lösung des RWP (2.49) in $\mathbf{z} := (\mathbf{x}^T, \boldsymbol{\lambda}^T)^T \in \mathbb{R}^{2n}$:

$$\mathbf{v} := (\boldsymbol{\lambda}_0, \hat{\mathbf{z}}_1^T, \dots, \hat{\mathbf{z}}_{N-1}^T)^T \in \mathbb{R}^{n(2N-1)} \quad (3.9)$$

Indirektes Kollokationsverfahren (IKKV)

Im Falle der IKKV werden im Vergleich zu den IMSV die Werte von \mathbf{z} zusätzlich innerhalb der N Teilintervalle an den M Kollokationspunkten vorgegeben:

$$\mathbf{v} := (\boldsymbol{\lambda}_0, \hat{\mathbf{z}}_{01}^T, \dots, \hat{\mathbf{z}}_{0M}^T, \hat{\mathbf{z}}_{10}^T, \dots, \hat{\mathbf{z}}_{1M}^T, \dots, \hat{\mathbf{z}}_{(N-1)0}^T, \dots, \hat{\mathbf{z}}_{(N-1)M}^T)^T \in \mathbb{R}^{n(2(M+1)N-1)} \quad (3.10)$$

Weitere indirekte Verfahren

Für das *Gradientenverfahren* wird das OSP (2.44) mit gegebenem Anfangszustand \mathbf{x}_0 , freiem Endzustand \mathbf{x}_f und der Endzustandsbewertung $V(\mathbf{x}(t_f))$ betrachtet. Als zweite Randbedingung muss dann

$$\boldsymbol{\lambda}(t_f) = \left. \frac{\partial V}{\partial \mathbf{x}} \right|_{t=t_f} \quad (3.11)$$

erfüllt werden [13, S. 92]. Setzt man eine Parametrierung des Eingangssignals wie in Gleichung (3.2) beschrieben an, kann man (2.47a) vorwärts und (2.47b) unter Nutzung des sich durch Einsetzen des erhaltenen $\mathbf{x}(t_f)$ in (3.11) ergebenden $\boldsymbol{\lambda}(t_f)$ rückwärts integriert werden. Schließlich kann der Gradient von $\bar{J}(\mathbf{u})$ gemäß (2.46)

$$\left(\frac{\partial \bar{J}(\mathbf{u})}{\partial \mathbf{u}} \right)^T = \int_{t_0}^{t_f} \left(\frac{\partial H}{\partial \mathbf{u}} \right)^T dt \quad (3.12)$$

zur iterativen Anpassung der Parameter des Eingangssignals genutzt werden. Verschwindet dieser Gradient, so erfüllt die entsprechende Trajektorie die notwendigen Optimalitätsbedingungen aus (2.47). [13, vgl. Kap. 6.1.4, 22, vgl. Kap. 15.3.2]

Bei der *Quasi-Linearisierung* wird das RWP (2.49) um eine geschätzte Lösungstrajektorie linearisiert und durch eine iterative Anpassung der Trajektorie gelöst. Es kann als Verallgemeinerung des Newton-Verfahrens für den dynamischen Fall betrachtet werden. [22, Kap. 15.3.3]

3.5 Implementierung und Software

Im Rahmen der Studienarbeit wurden Varianten der direkten und indirekten Schieß- und Kollokationsverfahren implementiert. Als Programmiersprache wurde dabei *Python*

in Version 3.6.5 [12] verwendet. Neben den Verfahren zur Trajektorienplanung selbst, wurde eine Klasse für **TPP** zur Untersuchung der Verfahren implementiert. Im Sinne der Nachvollziehbarkeit und Reproduzierbarkeit der Ergebnisse findet sich die vollständige Implementierung unter [26].

CasADi

Alle zur Lösung von **TPP** benötigten numerischen Verfahren wurden durch *CasADi* [4] in Version 3.4.3 [2] bereitgestellt. Was zunächst ein Software-Framework zum algorithmischen Differenzieren (AD) war, dessen Syntax sich an der von Computer-Algebra-Systemen (CAS) anlehnt, hat sich zu einem Framework zur numerischen Lösungen von Optimierungsproblemen entwickelt. Ein besonderer Fokus liegt dabei auf der Lösung von Optimalsteuerungsproblemen (siehe auch [1]). CasADi selbst ist in *C++* geschrieben, bietet jedoch eine Python-Schnittstelle. [4]

Den Kern von CasADi bildet nach wie vor das namensgebende Framework zum algorithmischen Differenzieren von symbolischen Ausdrücken, welche sich aus Matrix-Objekten zusammensetzen. Diese Matrix-Objekte werden im CCS-Format (engl. compressed column storage) gespeichert, welches sich insbesondere für dünnbesetzte Matrizen (engl. sparse matrices) eignet. Symbolische Ausdrücke werden als gerichteter azyklischer Graph (DAG, engl. directed acyclic graph) repräsentiert und können zur Definition von Funktions-Objekten genutzt werden. CasADi erlaubt das algorithmische Differenzieren sowohl von explizit als auch von implizit definierten Funktions-Objekten. [4] Da das algorithmische Differenzieren (auch: automatisches Differenzieren) für die vorliegende Arbeit nur indirekt von Bedeutung ist, wird an dieser Stelle im Zusammenhang mit CasADi auf [4, S. 8ff.], im Kontext der Regelungstechnik auf [24] und für weitere Informationen auf [15] verwiesen.

Als spezielle vordefinierte Funktions-Objekte sind solche zur Lösung von **AWP**, **NLGS** und **NLP** nennenswert. Diese Objekte implementieren Schnittstellen zu entsprechenden numerischen Lösungsverfahren unter Bereitstellung von Funktionalitäten zum Repräsentieren und Differenzieren symbolischer Ausdrücke. CasADi selbst kann einem Verfahren bei Bedarf z. B. Jacobi-Matrizen durch algorithmisches Differenzieren bereitstellen. [4] Bei **NLGS** wäre z. B. die Jacobi-Matrix der Gleichungen bezüglich der unabhängigen Variablen von Interesse (siehe Abschnitt 2.2).

CasADi bietet Schnittstellen zu verschiedenen Implementierungen numerischer Lösungsverfahren von Drittanbietern. Insbesondere wenn diese Verfahren dem aktuellen Stand der Technik entsprechen, ist deren flexibler Einsatz auf Basis einer einheitlichen Problemdefinition eine aus Anwendersicht vielversprechende Möglichkeit. Mit Blick auf die Trajektorienplanung, welche sich als anspruchsvolles mathematisches Problem aus Teilproblemen zusammensetzt, ist dies ein einleuchtendes Konzept. CasADi stellt dabei lediglich die zur Trajektorienplanung benötigten „Werkzeuge“ bereit und überlässt deren Anwendung dem Nutzer.

Lösung von AWP mit CasADi

Das Funktions-Objekt *integrator* eignet sich zur Lösung von AWP gewöhnlicher DGL der Form

$$\dot{\mathbf{x}} = \mathbf{f}_{\text{DGL}}(t, \mathbf{x}, \mathbf{p}), \quad \mathbf{x}(0) = \mathbf{x}_0 \quad (3.13a)$$

$$\dot{q} = f_{\text{quad}}(t, \mathbf{x}, \mathbf{p}), \quad q(0) = 0 \quad (3.13b)$$

mit den Parametern \mathbf{p} und dem sogenannten Quadratur-Zustand q , welcher sich aus der numerischen Integration von (3.13b) ergibt und z. B. für eine simultane Berechnung der Kosten bei OSP verwendet werden kann. [2, vgl. S. 30f.]

Die bei einer Installation von CasADi bereitgestellte Programmbibliothek *SUNDIALS* [18] (engl. suite of nonlinear and differential/algebraic equation solvers) enthält mit *CVODES* [28] (engl. sensitivity-enabled ode solver) Implementierungen numerischer Verfahren zur Lösung von AWP. Die SUNDIALS-Schnittstelle von CasADi erlaubt eine Nutzung des in CVODES implementierten BDF-Verfahrens (engl. backward differentiation formula), einem Mehrschrittverfahren zur Lösung von AWP. Dabei besteht auch die Möglichkeit einer Sensitivitätsanalyse von AWP wie sie bei der Lösung von RWP mit Schießverfahren nötig ist (siehe Abschnitt 2.5).

Die Lösung von AWP ist nicht nur im Rahmen der Schießverfahren sondern auch bei der Simulation dynamischer Systeme notwendig. Dabei wird die aus dem Anfangswert des Zustandes und der Eingangstrajektorie folgende Zustandstrajektorie möglichst genau bestimmt. Eine solche Simulation ist notwendig, um zu prüfen, ob die durch ein Verfahren zur Trajektorienplanung bestimmte Eingangstrajektorie das dynamische System auch tatsächlich in der gewünschten Weise steuert. Eine möglichst hohe Genauigkeit der Zustandstrajektorie setzt einen möglichst kleinen globalen Diskretisierungsfehler bei der Lösung des AWP voraus. Dafür ist bei einem gegebenen Verfahren die Wahl der Toleranz ϵ_{sim} für den lokalen Diskretisierungsfehler entscheidend (siehe Abschnitt 2.4).

Ausgehend von der Bedingung $\epsilon_{\text{sim}} \geq K \epsilon_{\text{M}}$ aus Gleichung (2.21), der als $\epsilon_{\text{M}} \approx 1,1 \cdot 10^{-16}$ bestimmten relativen Maschinengenauigkeit und der großzügigen Abschätzung

$$\max_i |x_i| \leq 10^3 =: K \quad (3.14)$$

wurde $\epsilon_{\text{sim}} = 10^{-12}$ als Toleranz für den lokalen Diskretisierungsfehler gewählt.² CVODES erlaubt sowohl die Vorgabe einer Toleranz für den absoluten als auch für den relativen lokalen Diskretisierungsfehler [17, S. 4f.]. Für beide Toleranzen wurde $\epsilon_{\text{sim}} = 10^{-12}$ vorgegeben.

Mit der Wahl der Simulationsschrittweite $h_{\text{sim}} = 10^{-3}$ soll eine hinreichend feine Auflösung der Trajektorien für eine grafische Darstellung gewährleistet werden. Für die

²Diese Fehlertoleranz mag sehr klein erscheinen. Es hat sich jedoch gezeigt, dass bei größeren Werten eine erhebliche Akkumulation von Rundungsfehlern nicht ausgeschlossen werden kann.

Genauigkeit der Simulation spielt diese Wahl jedoch keine Rolle, da die Schrittweitensteuerung von CVODES die Schrittweite gerade so wählt, dass ϵ_{sim} bei möglichst geringem Rechenaufwand, d. h. möglichst großer Schrittweite, eingehalten wird [17, S. 4f.].

Lösung von NLGS und NLP mit CasADi

Zur Lösung von NLGS und NLP wird mit IPOPT auf die Implementierung eines Innere-Punkte-Verfahrens genutzt. Im Falle von NLP ist dessen Rückmeldung über lokale Optimalität für die späteren Betrachtungen relevant. Für weitere Informationen sei auf [31, 32] verwiesen. CasADi besitzt auch Schnittstellen zu weiteren Implementierungen von Verfahren zur statischen Optimierung und z. B. mit WORHP [9] eine Schnittstelle zu einem Verfahren der sequentiellen quadratischen Programmierung (SQP-Verfahren).

Implementierte Verfahren

Der Ausgangspunkt zur Implementierung der einzelnen Verfahren zur Trajektorienplanung waren zusammen mit CasADi bereitgestellte Implementierungsbeispiele [3] zur Lösung konkreter Optimalsteuerungsprobleme. Diese beispielhaften Implementierungen der Verfahren wurden im Rahmen der Studienarbeit zur Lösung auf allgemeine TPP der Form (3.1) bzw. (2.45) angepasst und erweitert. Diese Verallgemeinerung erfolgte mithilfe einer eigens implementierten Klasse zur Beschreibung allgemeiner TPP und Verwaltung konkreter Probleme. Eine Übersicht der implementierten Verfahren zeigt Tabelle 3.1. Als Verfahrensparameter ergeben sich ggf. die Anzahl der Teilintervalle N und bei KKV die Anzahl der Kollokationspunkte M . Die angesetzte Parametrierung des Eingangs und der Charakter der entstehenden Eingangstrajektorie stellen wesentliche Unterscheidungsmerkmale der Verfahren dar.

Tabelle 3.1 – Übersicht der implementierten Verfahren.

Verfahren	Parameter	Eingangsparametrierung	Eingangstrajektorie
DESV	N	stückweise konstant	stückweise konstant
DMSV	N	stückweise konstant	stückweise konstant
DKKV	N, M	stückweise konstant	stückweise konstant
DKKV*	N, M	stückweise polynomial	stetig
IESV	keine	stetig (2.48)	stetig
IMSV	N	stetig (2.48)	stetig
IKKV	N, M	stetig (2.48)	stetig

Bei den implementierten KKV handelt es sich um orthogonale KKV. Konkret werden die Nullstellen des verschobenen Legendre-Polynoms als Kollokationspunkte verwendet.

Direkte Verfahren

Bei den Verfahren [DESV](#), [DMSV](#) und [DKKV](#) wurde der Eingang wie durch (3.2) beschrieben parametrisiert. Die in Abschnitt 3.3 vorgenommene identische Wahl der N Teilintervalle in Bezug auf das stückweise konstant angesetzte Eingangssignal und den bei [DMSV](#) und [DKKV](#) an den Teilintervallgrenzen vorzugebenden Zustandswerten (siehe 3.5 und 3.6) erfolgte stillschweigend. Grundsätzlich könnten bei der Parametrierung des Einganges auch andere Teilintervalle gewählt werden. Da dies jedoch einen nicht unerheblichen Mehraufwand bei der Implementierung ohne abzusehenden Mehrwert bedeuten würde, wurde eine in Bezug auf Eingang und Zustand identische zeitliche Diskretisierung angesetzt.

Der Ansatz eines stückweise polynomialen Eingangssignals stellt bei zusätzlicher Forderung von Stetigkeit an den Teilintervallgrenzen eine Möglichkeit dar, um eine stetige Eingangstrajektorie mithilfe von direkten Verfahren zu bestimmen. Im Rahmen des [DKKV*](#) wurde dieser Ansatz implementiert. Dieses Verfahren ist somit von dem als [DKKV](#) bezeichneten Verfahren zu unterscheiden.

Indirekte Verfahren

Die Formulierung des RWP (2.49), auf dessen Lösung die indirekten Verfahren basieren, erfolgte mithilfe von *SymPy* [21]. Anschließend wurde der SymPy-Ausdruck in einen CasADi-Ausdruck umgewandelt.

Theoretischer Vergleich

In Tabelle 3.2 zeigt die Größe und Struktur der Jacobi-Matrix \mathbf{J} des je nach Verfahren entstehenden [NLGS](#) \mathbf{g} bezüglich der unbekannten Variablen \mathbf{v} . Die Dimensionen der Jacobi-Matrix ergeben sich aus

$$\dim(\mathbf{J}) = \dim(\mathbf{g}) \times \dim(\mathbf{v}). \quad (3.15)$$

Vermutlich kann man auch für die Anzahl der Nichtnullelemente der Jacobi-Matrix und damit für deren Besetzungsdichte $\rho_{\mathbf{J}}$ eine Formel angeben.

Tabelle 3.2 – Jacobi-Matrix je nach Verfahren.

Verfahren	$\dim(\mathbf{g})$	$\dim(\mathbf{v})$	$\rho_{\mathbf{J}}$
DESV	n	mN	100 %
DMSV	nN	$(m+n)N - n$	< 100 %
DKKV	$n(M+1)N$	$(m+n(M+1))N - n$	< 100 %
DKKV*	$n(M+1)N + m(N-1)$	$(mM + n(M+1))N - n - m$	< 100 %
IESV	n	n	100 %
IMSV	$2n(N+1)$	$2n(N+1)$	< 100 %
IKKV	$2n((M+1)N+1)$	$2n((M+1)N+1)$	< 100 %

Kapitel 4

Empirischer Verfahrenvergleich

4.1 Methodik

Die zum Vergleich der einzelnen Verfahren zur Trajektorienplanung herangezogenen Kriterien sind Genauigkeit, Zuverlässigkeit, Rechenaufwand und im Falle von [OSP](#) zusätzlich die Gesamtkosten. Die zugehörigen positiven Verfahrenseigenschaften werden in dieser Arbeit mit dem Begriff der Leistungsfähigkeit eines Verfahrens zusammengefasst. Im Folgenden werden die einzelnen Vergleichskriterien näher erläutert.

Genauigkeit

Je kleiner die euklidische Norm der normierten Abweichung des Endzustandes vom gewünschten Endwert

$$e_f := \left\| (\mathbf{I}_n \cdot [\mathbf{x}])^{-1} \cdot (\mathbf{x}(t_f) - \mathbf{x}_f) \right\|_2 := \sqrt{\sum_{i=1}^n \left| \frac{x_i(t_f) - x_{i,f}}{[x_i]} \right|^2} \quad (4.1)$$

ist, desto höher ist die Genauigkeit einer Trajektorie. Dabei extrahieren die eckigen Klammern in (4.1) jeweils (ggf. komponentenweise) die physikalische Einheit des darin stehenden Ausdrucks [20, S. 15]. Somit ergibt der Ausdruck $(\mathbf{I}_n \cdot [\mathbf{x}])^{-1}$ mit der Einheitsmatrix $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ eine Diagonalmatrix der reziproken Einheiten von \mathbf{x} und sorgt für eine Normierung der Abweichung $(\mathbf{x}(t_f) - \mathbf{x}_f)$, sodass e_f einheitslos ist.

Die Abweichung e_f einer Trajektorie wird durch Simulation des betrachteten dynamischen Systems mit der berechneten Eingangstrajektorie $\mathbf{u}(\cdot)$ bestimmt. Eine solche Simulation entspricht der Lösung eines [AWP](#) einer autonomen [DGL](#). Die sich auf diese Weise ergebende Zustandstrajektorie $\mathbf{x}(\cdot)$ erfüllt somit immer die Systemgleichungen. Die Betrachtung der Abweichung nach (4.1) genügt somit zur Beschreibung der Genauigkeit einer Trajektorie.

Zuverlässigkeit

Beim Vorliegen von [UNB](#) in Form von Zustands- und Eingangsbeschränkungen muss eine Trajektorie auf Verletzung dieser Beschränkungen untersucht werden. Hierfür genügt die

Auswertung der Bedingungen (3.1b) bzw. (2.45c) durch Bildung von $\max(\mathbf{x}(\cdot))$ sowie $\min(\mathbf{x}(\cdot))$ und (3.1c) bzw. (2.45d) durch Bildung von $\max(\mathbf{u}(\cdot))$ sowie $\min(\mathbf{u}(\cdot))$.

Als Zuverlässigkeit Z eines Verfahrens wird die relative Häufigkeit der Lösungsfindung über alle Beispielprobleme verstanden. Ein TPP gilt im Folgenden genau dann als gelöst, wenn $e_f \leq \epsilon$ mit der festgelegten Toleranz ϵ ist und die UNB nicht verletzt werden.¹ Die zugehörige Trajektorie heißt dann *Lösungstrajektorie*.

Als Toleranz wird $\epsilon = 10^{-3}$ gewählt. Diese Wahl soll auf pragmatische Weise gewährleisten, dass e_f für alle im Rahmen der folgenden Untersuchungen zu erwartenden SI-Einheiten der einzelnen Zustandskomponenten einen vom Autor als sinnvoll erachteten Wert unterschreitet. Bezogen auf die Abweichung in nur einer Zustandskomponente wären dies z. B. bei Längen 1 mm, bei Winkeln 10^{-3} rad (etwa $0,057^\circ$) oder bei Geschwindigkeiten $1 \frac{\text{mm}}{\text{s}}$.

Rechenaufwand

Die Gesamtrechenzeit T wird als Maß für den Rechenaufwand herangezogen. Zusätzlich wird die Anzahl der benötigten Iterationen n_{Iter} sowie die sich daraus ergebende Rechenzeit pro Iteration T_{Iter} betrachtet. Diese Werte beziehen sich ausschließlich auf die Rechenzeit und die Iterationsschritte des durch IPOPT implementierten Innere-Punkte-Verfahrens, welches für alle implementierten Verfahren genutzt wird.

Während die Anzahl der Iterationen aufgrund der Deterministik des Innere-Punkte-Verfahrens keinen Schwankungen bei identischen Problemen unterworfen sein sollte, so sind bei der Rechenzeit erfahrungsgemäß durchaus Schwankungen zu erwarten. Es wird jedoch davon ausgegangen, dass diese Schwankungen in Bezug auf die Gesamtrechenzeit entweder vernachlässigbar sind oder der Rechenaufwand an sich bei sehr kleinen Gesamtrechenzeiten vernachlässigbar ist. Die Untersuchung der genauen Relation von vernachlässigbaren Rechenzeiten zueinander erscheint wenig zielführend. Auf eine wiederholte Ausführung derselben Simulationsexperimente mit anschließender Mittelwertbildung der Rechenzeiten wird aufgrund des höheren Simulationsaufwandes daher verzichtet.

Gesamtkosten

Bei OSP ist der Vergleich der Gesamtkosten J der Trajektorien insofern interessant, als das gravierend unterschiedliche Werte meist qualitativ unterschiedliche Lösungen bedeuten. Unterschiedliche Lösungen können auftreten, wenn ein OSP verschiedene lokale Minima besitzt. Die einzelnen Verfahren können dann zu unterschiedlichen Lösungen konvergieren, was jedoch im Allgemeinen unerwünscht ist. Vielmehr wird das Auffinden einer global optimalen Trajektorie gefordert.

Der Integralanteil $l(\mathbf{x}, \mathbf{u})$ des Kostenfunctionals J des betrachteten OSP (2.45) wird für

¹Bei OSP muss zusätzlich ein (lokales) Minimum der Gesamtkosten vorliegen.

alle Probleme einheitlich als

$$l(\mathbf{x}, \mathbf{u}) := (\mathbf{x} - \mathbf{x}_f)^T \mathbf{Q}(\mathbf{x} - \mathbf{x}_f) + \mathbf{u}^T \mathbf{R} \mathbf{u}, \quad \mathbf{Q} := (\mathbf{I}_n \cdot [\mathbf{x}])^{-2}, \quad \mathbf{R} := (\mathbf{I}_m \cdot [\mathbf{u}])^{-2} \quad (4.2)$$

mit den Wichtungsmatrizen \mathbf{Q}, \mathbf{R} gewählt. Somit sind die Kosten proportional zum quadratischen Abstand des Istzustandes zum Sollzustand sowie zum Quadrat des Einganges, welches als Stellaufwand betrachtet werden kann.

4.2 Beispielsysteme

Der Vergleich der Verfahren zur Trajektorienplanung soll anhand von drei verschiedenen mechanischen Systemen erfolgen. Dabei handelt es sich um verschiedene Varianten von Wagen-Pendel-Systemen, d. h. horizontal verschiebbare Wagen, an denen unterschiedliche Pendelvarianten gelenkig angebracht sind. Im Folgenden werden diese Systeme kurz vorgestellt.

System 1: *Einfach-Wagen-Pendel-System (EWPS)*

Das erste System ist das bereits in Abschnitt 2.7 beschriebene *EWPS*. Dieses soll sowohl in der Darstellungsvariante (2.42) als auch in der partiell linearisierten Zustandsraumdarstellung (2.43) betrachtet werden, um zu untersuchen, wie sich die partielle Linearisierung auf die Schwierigkeit von *TPP* auswirkt. Der als Differenz aus Anzahl der Konfigurationskoordinaten und der Anzahl der unabhängig eingeprägten Kräfte definierte Unteraktuiertheitsgrad des Systems beträgt 1. Das *EWPS* besitzt die Ruhelagen $\mathbf{x}_{\text{RL}} = (k \cdot 180^\circ, s_{\text{RL}}, 0, 0)^T$, $k \in \mathbb{Z}$, $s_{\text{RL}} \in \mathbb{R}$, $u_{\text{RL}} = 0$, wobei für ungerade k eine stabile und für gerade k eine instabile Ruhelage vorliegt.

System 2: *Tandem-Wagen-Pendel-System (TWPS)*

Bei dem in Abbildung 4.1 dargestellten *TWPS* sind zwei Pendel bestehend aus masselosen Stäben mit üblicherweise unterschiedlichen Längen s_1, s_2 sowie den Punktmassen m_1, m_2 gelenkig direkt mit einem horizontal verschiebbaren Wagen verbunden. Aus den mithilfe des Lagrange-Formalismus hergeleiteten Bewegungsgleichungen folgt mit $\mathbf{x} := (\theta_1, \theta_2, s, \omega_1, \omega_2, v)^T := (\theta_1, \theta_2, s, \dot{\theta}_1, \dot{\theta}_2, \dot{s})^T$ und $u := a := \ddot{s}$ die partiell linearisierte Zustandsraumdarstellung

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u) = \begin{pmatrix} \omega_1 \\ \omega_2 \\ v \\ \frac{g}{s_1} \sin(\theta_1) + \frac{1}{s_1} \cos(\theta_1) a \\ \frac{g}{s_2} \sin(\theta_2) + \frac{1}{s_2} \cos(\theta_2) a \\ a \end{pmatrix}. \quad (4.3)$$

Der Unteraktuiertheitsgrad des Systems beträgt 2. Das *TWPS* besitzt die Ruhelagen

$\mathbf{x}_{\text{RL}} = (k_1 \cdot 180^\circ, k_2 \cdot 180^\circ, s_{\text{RL}}, 0, 0, 0)^T$, $k_1, k_2 \in \mathbb{Z}$, $s_{\text{RL}} \in \mathbb{R}$, $u_{\text{RL}} = 0$. Sind k_1 und k_2 beide ungerade, so liegt eine stabile und ansonsten eine instabile Ruhelage vor.

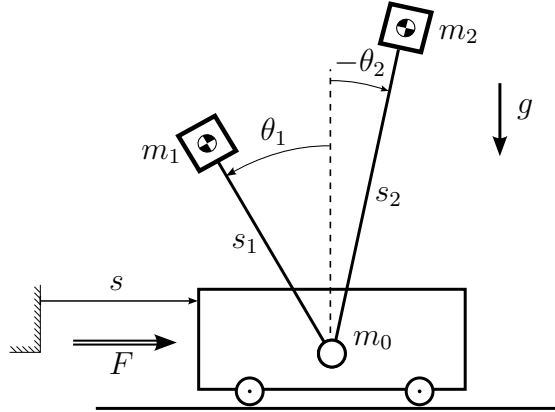


Abbildung 4.1 – Schematische Darstellung des TWPS [19, Abb. 4.3, S. 38].

System 3: *Dreifach-Wagen-Pendel-System (DWPS)*

Wie Abbildung 4.2 zeigt, sind beim DWPS drei Pendel hintereinander über Gelenke mit dem Wagen verbunden. Die Pendelwinkel sind dabei jeweils relativ zum „Vorgänger-Pendel“ definiert.

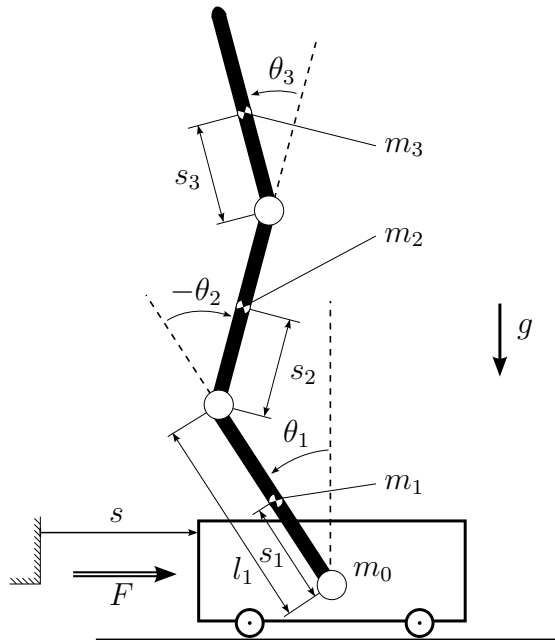


Abbildung 4.2 – Schematische Darstellung des DWPS [19, Abb. 4.3, S. 38 und Abb. 4.6, S. 67].

Im Unterschied zu den anderen beiden Wagen-Pendel-System erfolgte die Modellbildung

mit massebehafteten Pendelkörpern statt der Verbindung aus masselosem Stab und Punktmasse. Die Pendellängen l_1, l_2, l_3 sind damit nicht mehr gleich den Abständen der Pendelschwerpunkte zu den Gelenken s_1, s_2, s_3 und müssen neben den Trägheitsmomenten J_1, J_2, J_3 der Pendel als zusätzliche Modellparameter eingeführt werden.

Auf eine Angabe der verwendeten partiell linearisierten Zustandsraumdarstellung des Systems wird aus Platzgründen verzichtet. Zustand und Eingang des Systems sind durch $\mathbf{x} := (\theta_1, \theta_2, \theta_3, s, \omega_1, \omega_2, \omega_3, v)^T := (\theta_1, \theta_2, \theta_3, s, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{s})^T$ bzw. $u := a := \ddot{s}$ definiert. Der Unteraktuiertheitsgrad des Systems beträgt 3. Das **DWPS** besitzt die Ruhelagen $\mathbf{x}_{\text{RL}} = (k_1 \cdot 180^\circ, k_2 \cdot 180^\circ, k_3 \cdot 180^\circ, s_{\text{RL}}, 0, 0, 0, 0)^T$, $k_1, k_2, k_3 \in \mathbb{Z}$, $s_{\text{RL}} \in \mathbb{R}$, $u_{\text{RL}} = 0$. Ist k_1 ungerade und sind k_2 und k_3 beide gerade, so liegt eine stabile und ansonsten eine instabile Ruhelage vor.

4.3 Beispielprobleme

Die Tabelle 4.1 zeigt die in dieser Arbeit untersuchten **TPP**. Die insbesondere aufgrund des Unteraktuiertheitsgrades zu erwartende Schwierigkeit der Probleme spiegelt sich dabei bereits in deren Reihenfolge wider. Im Rahmen des Verfahrensvergleiches gilt ein Problem als umso schwieriger, je weniger Verfahren in der Lage sind dieses Problem zu lösen und je höher der Rechenaufwand dabei ist. Bei keinem der Probleme tritt eine Eingangs- oder Zustandsbeschränkung auf.

Tabelle 4.1 – Übersicht der Beispielprobleme ($\mathbf{x}_f := \mathbf{0}$, $t_0 := 0$).

#	System	\mathbf{x}_0	t_f/s	ÜFP	OSP	Besonderheit
1	EWPS	$(45^\circ, 0, 0, 0)^T$	1	Nein	Ja	keine
2	EWPS	$(180^\circ, 0, 0, 0)^T$	1	Ja	Ja	keine
3	EWPS	$(180^\circ, 0, 0, 0)^T$	1	Ja	Ja	nicht partiell linearisiert
4	EWPS	$(180^\circ, 0, 0, 0)^T$	$0, 1k$	Nein	Ja	$k \in [2, 50] \subset \mathbb{N}$
5	TWPS	$(180^\circ, 180^\circ, 0, 0, 0, 0)^T$	2	Ja	Ja	s_1 variabel
6	DWPS	$(180^\circ, 0, 0, 0, 0, 0, 0, 0)^T$	4	Nein	Ja	keine

Die Pendellänge s_1 des **TWPS** wird wie folgt variiert: $s_1 = k \cdot 0,05$ m, $k \in [1, 40] \subset \mathbb{N}$. In Tabelle 4.2 sind die konkreten Werte der Systemparameter angegeben. Die nicht angegebenen Parameter treten in der (partiell linearisierten) Zustandsraumdarstellung des betreffenden Systems nicht auf.

Tabelle 4.2 – Werte der Parameter der Beispielsysteme ($m_0 = 1$ kg, $g = 9,81$ ms⁻²).

Beispielsystem	EWPS	TWPS	DWPS ($i = 1, 2, 3$)
Pendellängen	$s_1 = 0,25$ m	$s_2 = 0,5$ m	$l_i = 0,5$ m
Pendelschwerpunkte	s_1	s_1, s_2	$s_i = \frac{1}{2}l_i$
Pendelmassen	$m_1 = 0,1$ kg	-	$m_i = 0,1$ kg
Pendelträgheitsmomente	-	-	$J_i = \frac{4}{3}m_i l_i^2$

4.4 Ergebnisse

Problem 1: Einfangen des EWPS

Für das Einfangen des EWPS konnten alle implementierten Verfahren eine (lokal) optimale Trajektorie bestimmen. Tabelle 4.3 zeigt die Werte der Vergleichsgrößen für die einzelnen Verfahren. Die zur Einordnung relevanten Minima und Maxima der jeweiligen Größen sind hervorgehoben. Zunächst fällt auf, dass das DESV mit großem Abstand den höchsten Rechenaufwand erforderte während die KKV im Vergleich allesamt besonders wenig rechenaufwändig waren.

Tabelle 4.3 – Einfangen des EWPS: Werte der Vergleichsgrößen nach Verfahren.

Verfahren	n_{Iter}	T/s	T_{Iter}/s	e_f	J
DES _V ($N = 100$)	271	610	2,25	$7,4 \cdot 10^{-8}$	343,6
DMS _V ($N = 100$)	5	3,3	0,66	$3,0 \cdot 10^{-6}$	262,3
DKKV ($N = 100, M = 4$)	4	0,12	0,03	$1,3 \cdot 10^{-5}$	262,3
DKKV* ($N = 10, M = 4$)	5	0,03	0,006	0,6	261,9
IES _V	25	9,3	0,37	$1,9 \cdot 10^{-8}$	342,9
IMS _V ($N = 15$)	15	10,2	0,68	$2,7 \cdot 10^{-5}$	263,2
IKKV ($N = 9, M = 9$)	17	0,41	0,02	$1,9 \cdot 10^{-4}$	263,2

Bei der Lösung von NLGS bzw. NLP liefert IPOPT Informationen zur Anzahl der Gleichungen bzw. Gleichungsnebenbedingungen und Problemvariablen sowie der Anzahl der Nichtnullelemente der Jacobi-Matrix der Gleichungen bezüglich der Problemvariablen. Für das Einfach-Wagen-Pendel sind die entsprechenden Werte sowie die sich daraus ergebende Besetzungsdichte der Jacobi-Matrix in Tabelle 4.4 angegeben (vgl. Tabelle 3.2).

Es ist naheliegend, dass höhere Werte für Dimensionen und Besetzungsdichte der Jacobi-Matrix mit einem höheren Rechenaufwand einhergehen. Diese Faktoren können

Tabelle 4.4 – EWPS: Dimension und Besetzungsdichte der Jacobi-Matrix des je nach Verfahren entstehenden NLGS.

Verfahren	$\dim(\mathbf{g})$	$\dim(\mathbf{v})$	$\rho_{\mathbf{J}}$	Nichtnullelemente
DESV ($N = 100$)	4	100	100 %	400
DMSV ($N = 100$)	400	496	0,75 %	1489
DKKV ($N = 100, M = 4$)	2000	2096	0,30 %	12376
DKKV* ($N = 10, M = 4$)	209	236	2,61 %	1288
IESV	4	4	100 %	16
IMSV ($N = 15$)	128	128	6,64 %	1088
IKKV ($N = 9, M = 9$)	728	728	1,62 %	8576

den unterschiedlichen Rechenaufwand der einzelnen Verfahren jedoch alleine nicht erklären. So könnte bei Schießverfahren die Lösung von AWP sowie die Bestimmung der Sensitivität bezüglich der Problemvariablen den Rechenaufwand zusätzlich erhöhen.

Außerdem fällt beim Vergleich der Gesamtkosten auf, dass die (sequentiellen) ESV offenbar zu einem anderen lokalen Minimum als die (simultanen) MSV und KKV geführt haben. Dies könnte ebenfalls zu den Unterschieden beim Rechenaufwand beigetragen haben. Der Vergleich des Rechenaufwandes wäre aussagekräftiger, wenn alle Verfahren eine Konvergenz zur (qualitativ) gleichen Lösung bewirkt hätten. In Bezug auf die Gesamtkosten ist das Ergebnis ein Indiz dafür, dass eine simultane Optimierung der Zustands- und Eingangstrajektorie wie bei MSV und KKV besser geeignet ist, um eine global optimale Trajektorie zu bestimmen als eine sequentielle Optimierung durch die ESV.

Bei Betrachtung der Abbildung 4.3 zeigt sich zunächst beim Vergleich der beiden Diagramme, dass man in der Tat im Wesentlichen von zwei unterschiedlichen Lösungen sprechen kann. Außerdem ist jeweils der qualitativ unterschiedliche Verlauf des Eingangssignals je nach Parametrierung (vgl. Tabelle 3.1) zu erkennen. Der quantitative Unterschied der im unteren Diagramm dargestellten Eingangsverläufe von DMSV bzw. DKKV sowie IMSV bzw. IKKV ist jeweils so gering, dass er optisch nicht sinnvoll darstellbar ist. Der mit dem DKKV* bestimmte Eingangsverlauf weist Abweichungen zum dargestellten Verlauf auf. Da diese jedoch nur schlecht sichtbar wären und an dieser Stelle der qualitative Verlauf im Vordergrund steht, wurde auf eine zusätzliche Abbildung verzichtet.

Der Verlauf der Zustandstrajektorien in Abbildung 4.4 offenbart, dass die mit den ESV bestimmten Eingangstrajektorien (linkes Diagramm) kein Einfangen des Pendels im eigentlichen Sinne bewirken. Vielmehr führt der Beschleunigungsverlauf des Wagens dazu, dass das Pendel seine untere Ruhelage durchläuft, dabei stark abgebremst wird,

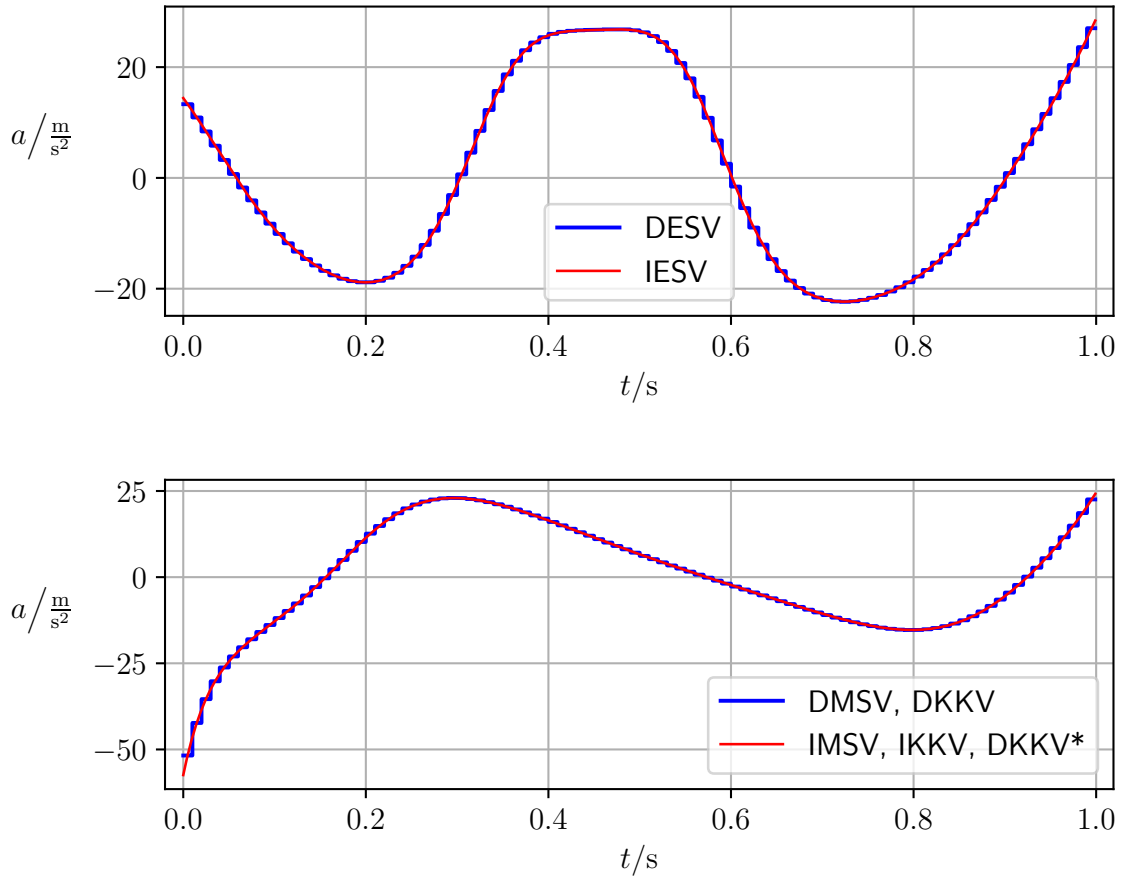


Abbildung 4.3 – Einfangen des EWPS: Eingangstrajektorien.

um dann wieder aufzuschwingen und schließlich seine obere Ruhelage zu erreichen. Die kinetische Energie des Pendels geht dabei unter Aufwendung von Energie zur Beschleunigung des Wagens größtenteils verloren. Mit Blick auf die Berücksichtigung des Stellaufwandes im Kostenfunktional würde man eher eine Nutzung der kinetischen Energie zum Aufschwingen des Pendels erwarten. Für eine Plausibilität der Trajektorie sprechen jedoch zwei Dinge:

Erstens ist der Betrag des Pendelwinkels im Zustandsraummodell nicht auf 180° beschränkt. Als gewünschter Endwert des Pendelwinkels ist mit 0° die obere Ruhelage vorgegeben. Das Aufschwingen des Pendels muss daher immer von der Seite erfolgen, von welcher die Anfangsauslenkung bestand.

Zweitens erfordert die Nutzung der kinetischen Energie zum Aufschwingen von der „richtigen“ Seite ein Durchlaufen der unteren Ruhelage und damit eine Vergrößerung des Pendelwinkels auf deutlich über 180° , was als Abweichung des Pendelwinkels vom Endwert durch das Kostenfunktional bestraft werden würde. Außerdem könnte die begrenzte Überführungszeit dazu führen, dass das Zurückschwingen des Pendels nicht

„abgewartet“ werden kann.²

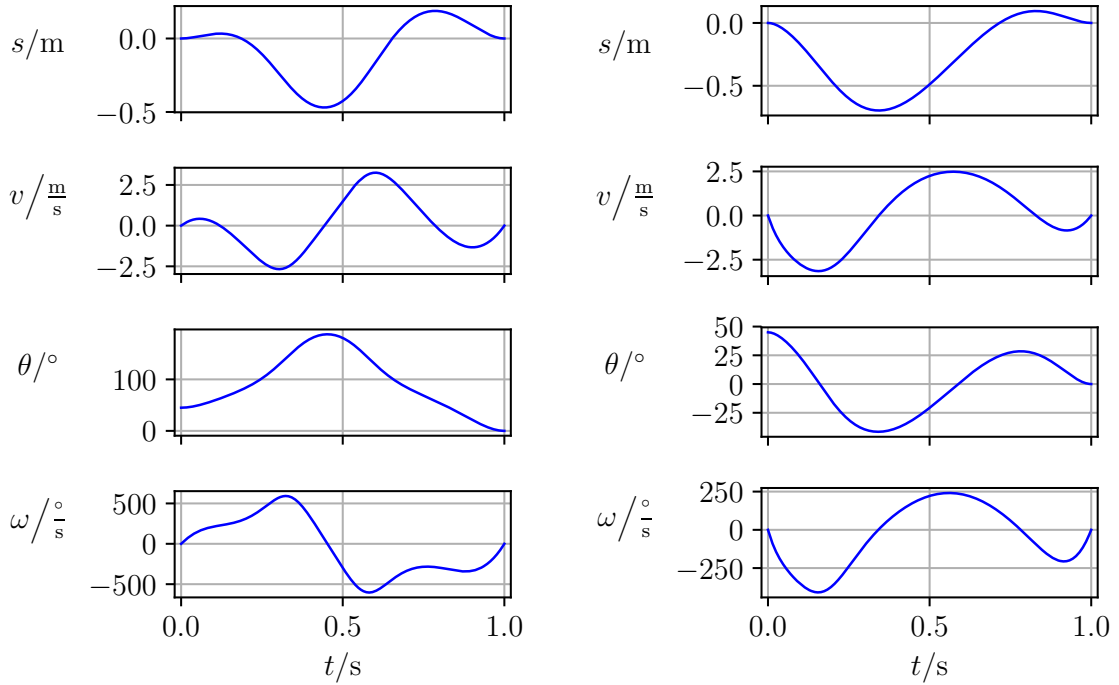


Abbildung 4.4 – Einfangen des EWPS: Zustandstrajektorien (links: DESV, IESV; rechts: DMSV, DKKV, DKKV*, IMSV, IKKV).

Der im rechten Diagramm dargestellte Zustandsverlauf zeigt, dass der mit den anderen Verfahren bestimmte Beschleunigungsverlauf des Wagens zu einem tatsächlichen Einfangen des Pendels führt. Ein Vergleich der Diagramme bezüglich der Verläufe von Winkel und Winkelgeschwindigkeit des Pendels erklärt die unterschiedlichen Gesamtkosten der beiden Lösungen. Die unterschiedlichen Parametrierungen des Eingangs führten aufgrund der relativ feinen Diskretisierung jeweils zu keiner sichtbaren Abweichung bei den Zustandstrajektorien. Lediglich die mithilfe des DKKV* bestimmte Zustandstrajektorie wies geringe Abweichungen zu den im rechten Diagramm dargestellten Verläufen auf. Auf eine separate Darstellung wurde jedoch auch hier verzichtet.

Schließlich ist mit Blick auf die Abweichungen vom Endzustand in Tabelle 4.4 zu bemerken, dass Schießverfahren in diesem Fall zu einer höheren Genauigkeit als Kollationsverfahren führen. Das DKKV* überschreitet dabei als einziges Verfahren die festgelegte Toleranz $\epsilon = 10^{-3}$. Tatsächlich ist dies nicht auf das Verfahren selbst, sondern auf ein Problem bei der Simulation zurückzuführen. Zur Lösung des AWP (3.13) müssen

²In Kombination mit der fehlenden Begrenzung des Pendelwinkelbetrages im Zustandsraummodell erfolgt durch das Kostenfunktional keine „Überbewertung“ von Pendelwinkeln mit Beträgen größer 180° wie man zunächst vermuten könnte. Vielmehr wird dadurch dem Umstand Rechnung getragen, dass das System tatsächlich weiter vom gewünschten Endwert entfernt ist.

in jedem Simulationsschritt die Parameter \mathbf{p} vorgegeben werden. In diesem Fall ist das der Wert des Einganges zum entsprechenden Zeitpunkt. Im Unterschied zu einem stückweise konstanten Eingangssignal, kann bei einem polynomialen Eingangssignal jedoch nicht davon ausgegangen werden, dass dieses über einen Simulationsschritt hinweg konstant ist. In der Folge kann bei der Simulation nicht mehr effektiv von einer Schrittweitensteuerung profitiert werden und die Simulationsschrittweite beeinflusst ungewollt die Genauigkeit: Eine größere Simulationsschrittweite bedeutet einen größeren Fehler bei geringerem Rechenaufwand und umgekehrt.

Daher musste beim **DKKV*** abweichend von den anderen Verfahren eine Simulationsschrittweite von $h_{\text{sim}} = 10^{-5}$ gewählt werden, was zu einem gerade noch akzeptablen Rechenaufwand, jedoch immer noch nicht zu einer Einhaltung der geforderten Genauigkeit führte. Das beschriebene Problem ließe sich durch Einsetzen der polynomialen Parametrierung des Eingangssignals in (3.13) und eine Übergabe der Parameter für das jeweilige Teilintervall während der Simulation beheben.

Tatsächlich sind auch beim Ansatz eines stückweise konstanten Eingangssignals Probleme in Bezug auf die Genauigkeit der Simulation zu erwarten, falls die Sprünge des Eingangssignals innerhalb der Simulationsschritte liegen. Um dies auszuschließen wird bei den betreffenden Verfahren (siehe Tabelle 3.1) eine Anpassung der Simulationsschrittweite an die Länge der N Teilintervalle vorgenommen. Konkret wird dasjenige ganzzahlige Vielfache der Teilintervall-Länge als Simulationsschrittweite verwendet, welches dem Zielwert von 10^{-3} am nächsten ist.³

Das Einfangen des **EWPS** war das einzige Problem, welches von allen Verfahren außer dem **DKKV*** gelöst werden konnte. Bei den im Folgenden betrachteten Problemen waren nur noch das **DMSV** und **DKKV** in der Lage dazu Lösungstrajektorien zu bestimmen. Bei den anderen Verfahren traten im Wesentlichen drei verschiedene Fälle auf: Nach den festgelegten 1000 Iterationsschritten konnte keine Lösung gefunden werden. Das Innere-Punkte-Verfahren konvergierte zu einem Punkt, der die Nebenbedingungen verletzt. Insbesondere bei **ESV** führten auch über alle Grenzen hinaus wachsende **AWP**-Lösungen zu einem Abbruch der Iteration.

Problem 2: Aufschwingen des **EWPS**

Das Aufschwingen des **EWPS** wurde als **OSP** und **ÜFP** betrachtet. Dabei konnten jeweils sowohl durch das **DMSV** als auch das **DKKV** Lösungstrajektorien bestimmt werden, wobei die Genauigkeit des Schießverfahrens wieder etwas höher war als die des Kollokationsverfahrens. Vergleicht man die Werte der einzelnen Kriterien für das **OSP** in Tabelle 4.5 mit denen für das **ÜFP** in Tabelle 4.6 so fällt auf, dass das **DKKV** zum Lösen des **OSP** länger als das **DMSV** und wesentlich länger als zum Lösen des **ÜFP** brauchte. Dabei wurde das **ÜFP** vom **DKKV** sehr viel schneller als vom **DMSV** gelöst.

Weiterhin zeigt sich bei Betrachtung der Gesamtkosten, dass das **DKKV** für $M = 4$

³Für eine Teilintervall-Länge kleiner 10^{-3} wird diese als Simulationsschrittweite gewählt.

Tabelle 4.5 – Aufschwingen des EWPS (OSP): Werte der Vergleichsgrößen nach Verfahren.

Verfahren	n_{Iter}	T/s	T_{Iter}/s	e_f	J
DMSV ($N = 100$)	13	10,2	0,78	$8,9 \cdot 10^{-7}$	205,7
DKKV ($N = 100, M = 4$)	359	14,1	0,04	$1,1 \cdot 10^{-6}$	205,7
DKKV ($N = 100, M = 8$)	332	31,2	0,09	$3,5 \cdot 10^{-7}$	100,0

Tabelle 4.6 – Aufschwingen des EWPS (ÜFP): Werte der Vergleichsgrößen nach Verfahren.

Verfahren	n_{Iter}	T/s	T_{Iter}/s	e_f
DMSV ($N = 100$)	191	135,9	0,71	$9,2 \cdot 10^{-7}$
DKKV ($N = 100, M = 4$)	158	3,7	0,02	$7,6 \cdot 10^{-5}$
DKKV ($N = 100, M = 8$)	150	7,7	0,05	$1,1 \cdot 10^{-6}$

zur gleichen Lösungstrajektorie wie das DMSV, für $M = 8$ jedoch zu einer anderen Trajektorie führt. Dies wird auch durch die entsprechenden Verläufe in Abbildung 4.5 bestätigt.

Tatsächlich wurde beim DKKV durch IPOPT je nach Anzahl der Kollokationspunkte M ein anderes lokales Minimum gefunden. Die Gesamtkosten waren dabei zum Teil sehr viel höher als für $M = 4$, jedoch nie niedriger als für $M = 8$. Insbesondere für $M > 4$ ist das Auffinden lokaler Minima mit höheren Gesamtkosten als für $M = 4$ ein unerwünschtes Verhalten des Innere-Punkte-Verfahrens, welches vermutlich durch die numerischen Eigenschaften des bei DKKV entstehenden NLP begünstigt wird.

Wie Abbildung 4.6 zeigt, ergab sich beim ÜFP für die beiden DKKV jeweils die gleiche Lösungstrajektorie, für das DMSV jedoch eine andere. Dabei sei angemerkt, dass zur Lösung des ÜFP der Betrag des Einganges auf 100 ms^{-2} begrenzt wurde. Ohne diese Beschränkung führten sehr große Beträge und Sprünge des Einganges zu unbrauchbaren Lösungen. Aufgrund des fehlenden Kostenfunctionals beim ÜFP ist der Verlauf von Zustand und Eingang im Rahmen der Trajektorienplanung irrelevant, solange der Endzustand erreicht wird. Ohne das Stellen weiterer Bedingungen können bei ÜFP somit große Eingangsbeträge und -sprünge nicht ausgeschlossen werden.

Problem 3: Aufschwingen des nicht partiell linearisierten EWPS

In Bezug auf die in den Tabellen 4.7 und 4.8 dargestellten Werte der Vergleichskriterien für das OSP bzw. ÜFP ergaben sich beim nicht partiell linearisierten EWPS qualitativ sehr ähnliche Ergebnisse im Vergleich zur zuvor betrachteten partiell linearisierten Darstellung des Systems. Quantitativ betrachtet hat sich der Rechenaufwand im Vergleich

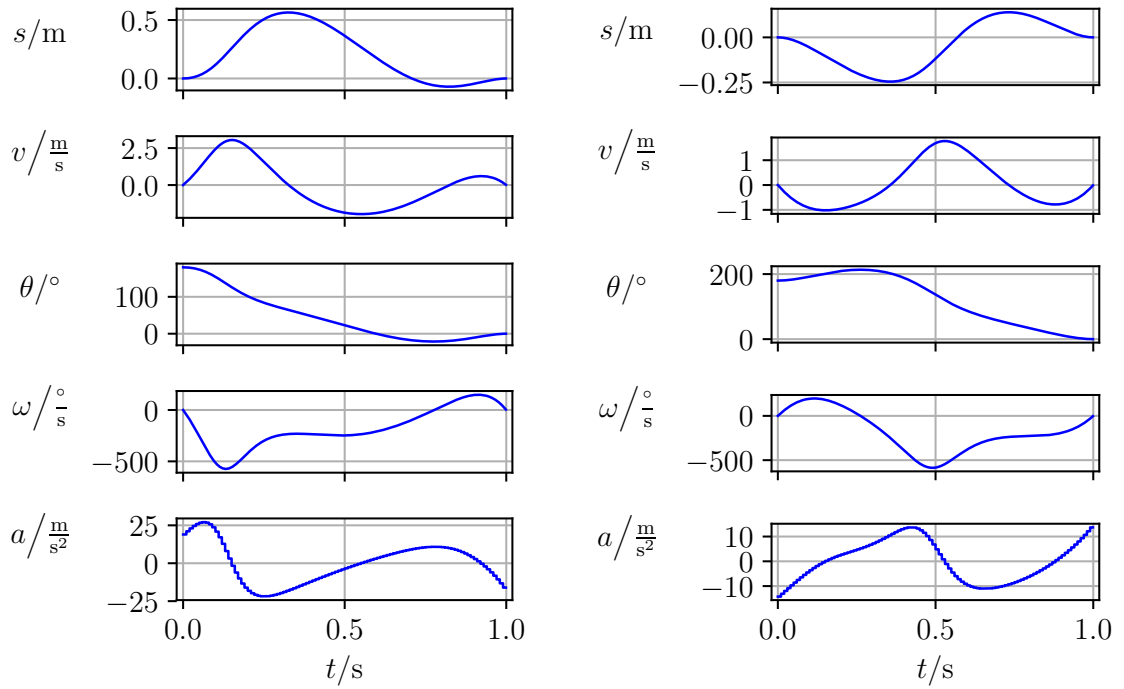


Abbildung 4.5 – Aufschwingen des EWPS (OSP): Lösungstrajektorien (links: DMSV, DKKV ($M = 4$); rechts: DKKV ($M = 8$)).

zum vorherigen Problem wesentlich erhöht. Die Erhöhung fiel beim DMSV mit einem Faktor von etwa 3 beim OSP und etwa 4 beim ÜFP besonders groß aus. Beim DKKV stieg die Gesamtrechnenzeit für $M = 4$ etwa um den Faktor 2 und für $M = 9$ noch um etwa 50 %.

Tabelle 4.7 – Aufschwingen des nicht partiell linearisierten EWPS (OSP): Werte der Vergleichsgrößen nach Verfahren.

Verfahren	n_{Iter}	T/s	T_{Iter}/s	e_f	J
DMSV ($N = 100$)	13	36,4	2,80	$9,0 \cdot 10^{-7}$	222,5
DKKV ($N = 100, M = 4$)	329	24,8	0,08	$8,9 \cdot 10^{-7}$	222,5
DKKV ($N = 100, M = 9$)	232	39,2	0,17	$3,4 \cdot 10^{-7}$	103,5

Beim DKKV lieferte IPOPT wie auch zuvor beim partiell nicht linearisierten EWPS je nach Anzahl M der Kollokationspunkte unterschiedliche lokale Minima. Hier ergab sich die Lösungstrajektorie mit den geringsten Gesamtkosten allerdings für $M = 9$ statt wie vorher für $M = 8$. Qualitativ entsprechen die jeweils für das OSP und ÜFP bestimmten Lösungstrajektorien denen des zuvor betrachteten partiell linearisierten EWPS (siehe Abbildungen 4.5 und 4.6). Quantitativ ergeben sich dabei lediglich in

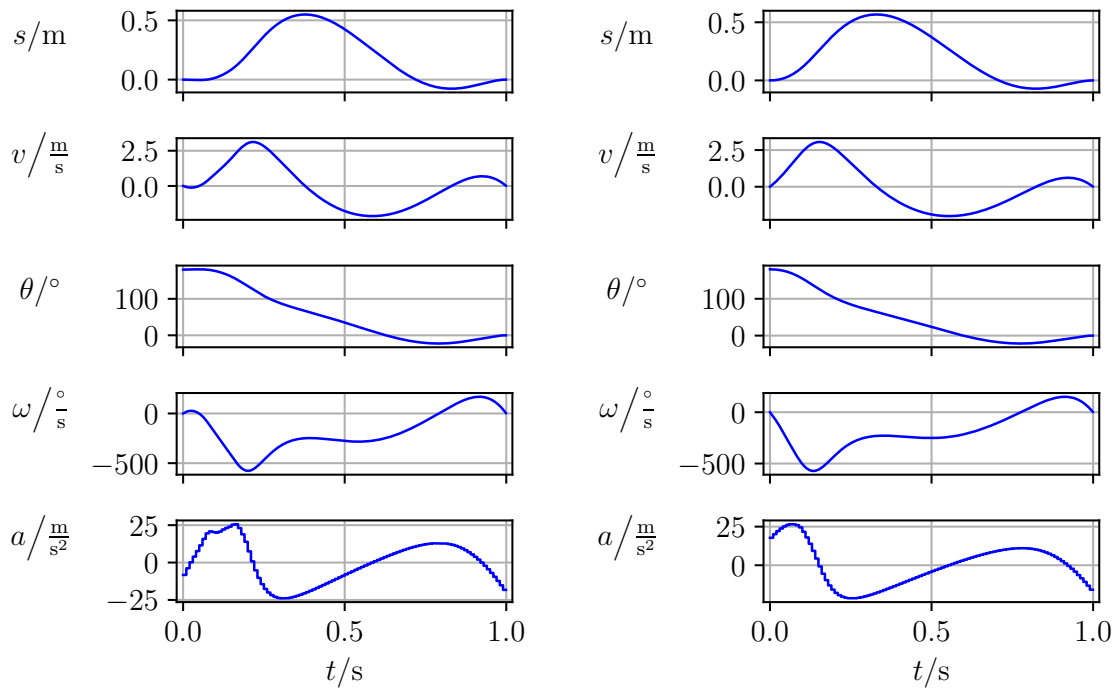


Abbildung 4.6 – Aufschwingen des EWPS (ÜFP): Lösungstrajektorien (links: DMSV; rechts: DKKV ($M = 4$), DKKV ($M = 8$)).

Tabelle 4.8 – Aufschwingen des nicht partiell linearisierten EWPS (ÜFP): Werte der Vergleichsgrößen nach Verfahren.

Verfahren	n_{Iter}	T/s	T_{Iter}/s	e_f
DMSV ($N = 100$)	208	535,9	2,58	$2,8 \cdot 10^{-7}$
DKKV ($N = 100, M = 4$)	150	6,3	0,04	$2,6 \cdot 10^{-7}$
DKKV ($N = 100, M = 9$)	118	12,3	0,10	$9,9 \cdot 10^{-4}$

Bezug auf die Eingangstrajektorie Unterschiede, da die beiden Systeme unterschiedliche Eingangsgrößen besitzen (siehe Gleichungen (2.42) und (2.43)). Hierdurch erklären sich auch die Abweichungen der Gesamtkosten zwischen den beiden Systemen.

Problem 4: Aufschwingen des EWPS mit variabler Überführungszeit

Für das Aufschwingen des EWPS mit variabler Überführungszeit konnte das DMSV und das DKKV für alle Überführungszeiten (lokal) optimale Trajektorien bestimmen. Wie Abbildung 4.7 zeigt, kam es allerdings vor allem bei größeren Überführungszeiten zu erheblichen Abweichungen vom Endzustand. Insgesamt hielten nur 36,7 % der mit dem DMSV und nur 49,0 % der mit dem DKKV bestimmten Trajektorien die festgelegte

Toleranz für Abweichungen vom Endzustand ein. Nur diese Trajektorien stellen somit Lösungstrajektorien des Problems dar. Tabelle 4.9 zeigt die durchschnittlichen Werte der Vergleichsgrößen für diese Lösungstrajektorien. Die im Vergleich zu vorherigen Problemen größer gewählte Anzahl der Teilintervalle N soll eine angemessene zeitliche Diskretisierung des Einganges für alle Überführungszeiten bis 5 s gewährleisten.

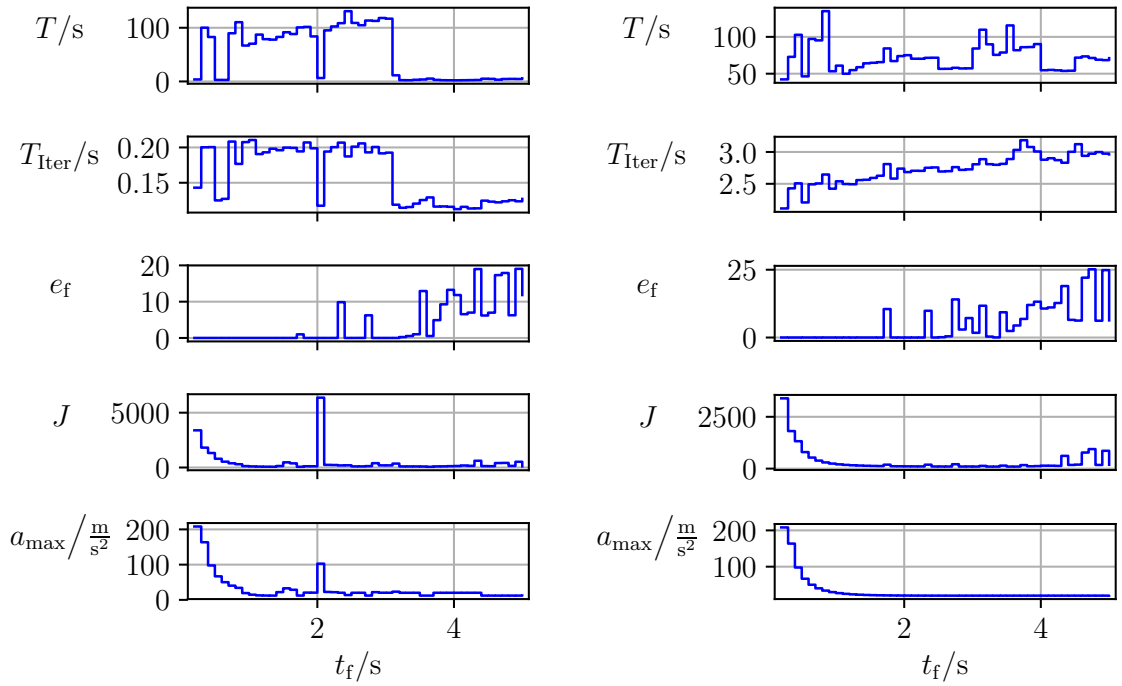


Abbildung 4.7 – Aufschwingen des EWPS: Vergleichskriterien für das DKKV (links) und DMSV (rechts) in Abhängigkeit der Überführungszeit.

Auffällig ist, dass das DKKV durchschnittlich wesentlich mehr Iterationen und somit trotz erheblich geringerer Rechenzeit pro Iteration insgesamt eine etwas höhere Gesamt-rechenzeit als das DMSV benötigte. Betrachtet man den Verlauf der Gesamt-rechenzeit für das DKKV im linken Diagramm von Abbildung 4.7, so fällt ein plötzlicher Rückgang der Gesamt-rechenzeit ab einer Überführungszeit von etwa 3 s auf. Interessanterweise ist dies auch etwa diejenige Überführungszeit, ab welcher die Abweichungen vom End-zustand beim DKKV stark ansteigen. Ab $t_f = 3,2$ s konnten beide Verfahren keine Trajektorie mit der geforderten Genauigkeit bestimmen.

Für die kleinste Überführungszeit von 0,2 s konnten hingegen beide Verfahren eine Lösungstrajektorie bestimmen. Um kleinere Überführungszeiten zu realisieren muss der Wagen allerdings auch zwangsläufig stärker beschleunigt werden. Dies zeigt auch der Verlauf des maximalen Beschleunigungsbetrages a_{\max} in beiden Diagrammen. Zu sehen ist auch, dass der steigende Stellaufwand auch mit höheren Gesamtkosten einhergeht.

Bei der Betrachtung der Gesamtkosten fällt außerdem auf, dass das DKKV für eine

Tabelle 4.9 – Aufschwingen des **EWPS** mit variabler Überführungszeit: Zuverlässigkeit der Verfahren und durchschnittliche Werte der Vergleichsgrößen für Lösungstrajektorien.

Verfahren	Z	\bar{n}_{Iter}	\bar{T}/s	\bar{T}_{Iter}/s	\bar{e}_f
DMSV ($N = 500$)	36,7 %	28	71,1	2,52	$3,8 \cdot 10^{-5}$
DKKV ($N = 500, M = 4$)	49,0 %	392	76,3	0,18	$2,5 \cdot 10^{-5}$

Überführungszeit von 2 s eine Trajektorie mit sehr hohen Gesamtkosten bestimmt hat. Schon bei den Problemen 2 und 3 zeigte sich, dass das Innere-Punkte-Verfahren beim **DKKV** insbesondere je nach Wahl der Verfahrensparameter dazu neigt, zu ungünstigen lokalen Minima zu konvergieren. Der hohe Wert von a_{\max} der betrachteten Trajektorie legt nahe, dass man dieses unerwünschte Verhalten eventuell durch eine Eingangsbeschränkung verhindern könnte.

Dennoch wurde mithilfe des **DKKV** die Lösungstrajektorie mit dem geringsten Gesamtkosten von $J = 70,1$ für $t_f = 1,3$ s bestimmt. Die Gesamtkosten der mithilfe des **DMSV** bestimmten Lösungstrajektorie waren für diese Überführungszeit mit $J = 137,2$ deutlich höher.

Insgesamt kann man sagen, dass die Überführungszeit erhebliche Auswirkungen auf die Schwierigkeit eines konkreten **TPP** hat. Für zu große Überführungszeiten konnten für das konkrete Problem keine Lösungstrajektorien bestimmt werden. Kleine Überführungszeiten scheinen hingegen nur hinsichtlich des Stellaufwandes problematisch zu sein.

Problem 5: Aufschwingen des **TWPS** bei Variation einer Pendellänge

Das Aufschwingen des **TWPS** wurde sowohl in Form eines **OSP** als auch in Form eines **ÜFP** untersucht. Dabei wurde die Länge eines Pendels variiert. Beim **ÜFP** wies das **DMSV** wie auch schon beim Aufschwingen des **EWPS** (siehe Tabelle 4.6) erheblich größere Gesamtrechnenzeiten als das **DKKV** auf, sodass von einer Bestimmung von Trajektorien für alle 40 Pendellängen abgesehen wurde. Für das **DKKV** musste der Betrag des Einganges a_{\max} zur Lösung des **ÜFP** wie bereits beim Aufschwingen des **EWPS** auf 100 ms^{-2} beschränkt werden, um brauchbare Trajektorien zu erhalten. Aufgrund der in den vorangegangenen Problemen beobachteten Neigung des Innere-Punkte-Verfahrens beim **DKKV** zu ungünstigen lokalen Minima zu konvergieren, wurde das **DKKV** beim **OSP** ohne und mit einer Eingangsbeschränkung von ebenfalls $a_{\max} \leq 100 \text{ ms}^{-2}$ angewendet.

Die Tabelle 4.10 zeigt, dass das **OSP** von allen Verfahren mit relativ großer Zuverlässigkeit gelöst werden konnte, wobei das **DKKV** ohne Eingangsbeschränkung die höchste und das **DMSV** die niedrigste Zuverlässigkeit aufwies. Wie bei Problem 4 konnte IPOPT für alle **OSP** lokal optimale Trajektorien bestimmen. Ausschlaggebend für die Zuverlässigkeit war somit auch in diesem Fall die Genauigkeit. Lediglich für die Werte 0,15 m und 0,2 m von s_1 konnte von keinem der Verfahren eine hinreichend genaue Trajektorie bestimmt

werden, wobei es jedoch nur zu relativ kleinen Überschreitungen der Fehlertoleranz kam. Für das betrachtete Aufschwingen des **TWPS** führen demnach insbesondere besonders kleine Werte von s_1 im Vergleich zu s_2 zu einer Erhöhung der Schwierigkeit, jedoch mutmaßlich nicht zu unlösbaren Problemen.

Tabelle 4.10 – Aufschwingen des **TWPS**: Zuverlässigkeit der Verfahren und durchschnittliche Werte der Vergleichsgrößen für die Lösungstrajektorien.

Verfahren	TPP	Z	\bar{n}_{Iter}	\bar{T}/s	\bar{T}_{Iter}/s	\bar{e}_f
DMSV ($N = 200$)	OSP	85,0 %	41	82,2	2,01	$9,3 \cdot 10^{-5}$
DKKV ($N = 200, M = 4$)	OSP	92,5 %	531	68,5	0,12	$4,3 \cdot 10^{-5}$
DKKV ($N = 200, M = 4, a_{\max} \leq 100 \text{ ms}^{-2}$)	OSP	90,0 %	231	13,0	0,06	$8,4 \cdot 10^{-5}$
DKKV ($N = 200, M = 4, a_{\max} \leq 100 \text{ ms}^{-2}$)	ÜFP	67,5 %	271	15,3	0,06	$1,4 \cdot 10^{-4}$

Mit Blick auf Abbildung 4.8 ist bemerkenswert, dass die Eingangsbeschränkung beim **DKKV** tatsächlich zu Trajektorien mit geringeren Gesamtkosten führten. Im Vergleich zum **DMSV** im linken Diagramm von Abbildung 4.9 zeigt sich übereinstimmend ein größtenteils linear anmutender Anstieg der Maximalbeträge des Einganges mit wachsender Länge s_1 , welcher mit in ähnlicher Weise steigenden Gesamtkosten einhergeht. Für beide Verfahren ergaben sich für gleiche Pendellängen geringere Gesamtkosten als für geringfügig unterschiedliche Pendellängen. Beim **DKKV** war dies interessanterweise auch für $s_1 = 1,05 \text{ m}$ der Fall. Das linke Diagramm in Abbildung 4.10 zeigt, dass sich im Vergleich zu $s_1 = 1 \text{ m}$ auch eine völlig andere Lösungstrajektorie ergibt. Die Lösungstrajektorien für $s_1 = 0,25 \text{ m}$ und $s_1 = 0,55 \text{ m}$ im rechten Diagramm ähneln qualitativ bis auf kleinere Unterschiede der Lösungstrajektorie für $s_1 = 1 \text{ m}$ überraschend stark.

Die Beschränkung des Eingangsbetrages beim **DKKV** führte nicht nur zu geringeren Gesamtkosten für bestimmte Werte von s_1 , sondern durchschnittlich auch zu einer erheblichen Verminderung des Rechenaufwandes. Dieser liegt bei diesem Verfahren für das **OSP** auf einem ähnlichen Niveau wie für das **ÜFP**. Allerdings ergab sich durch die Formulierung als **ÜFP** eine starke Reduktion der Zuverlässigkeit des **DKKV**. Für die Eingangstrajektorien der Trajektorien mit geringer Genauigkeit ist ein Springen zwischen dem maximal und minimal zulässigen Eingangswert charakteristisch. Das linke Diagramm in Abbildung 4.9 zeigt, dass der maximal zulässige Eingangsbetrag bei vielen Trajektorien erreicht wurde, während derartig hohe Eingangsbeträge beim **OSP** für das **DKKV** mit Eingangsbeschränkung und das **DMSV** nicht auftraten.

Insgesamt kann man schlussfolgern, dass die Formulierung eines **TPP** als **OSP** für direkte Verfahren mit stückweise konstantem Eingangssignal bei etwas geringerem Rechenaufwand wesentlich zuverlässiger „brauchbare“ Trajektorien liefert als bei einer Formulierung als **ÜFP** ohne Kostenfunktional. Die hierfür ursächlichen großen Ein-

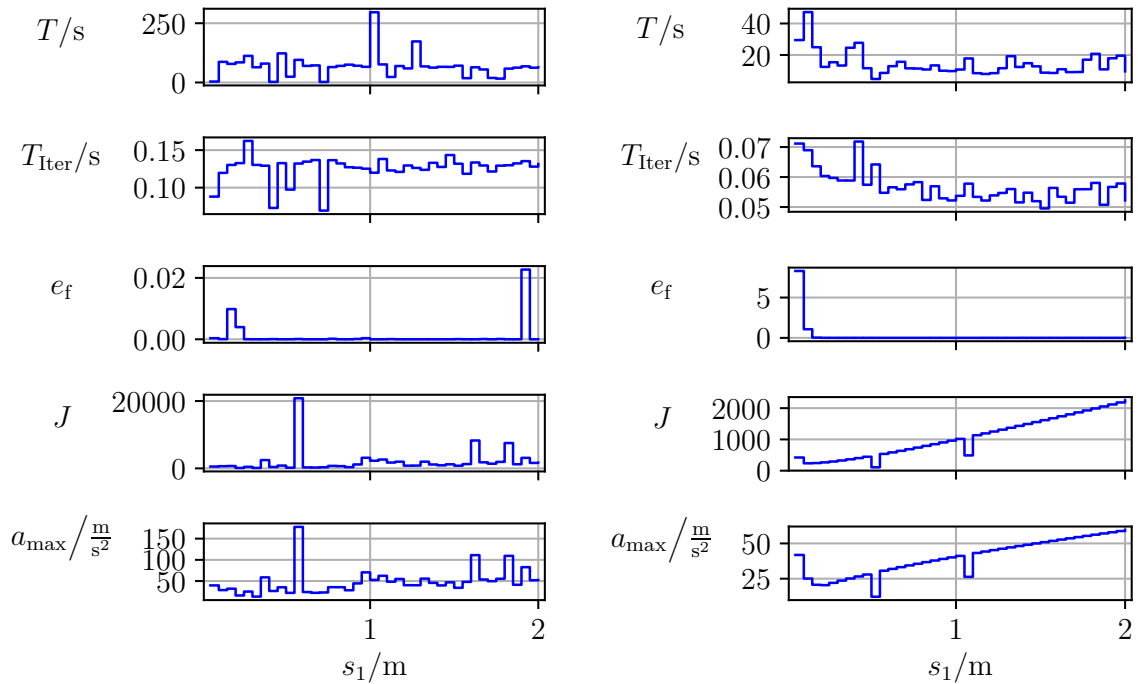


Abbildung 4.8 – Aufschwingen des TWPS (OSP): Vergleichskriterien für das DKKV ohne (links) und mit Eingangsbeschränkung (rechts) in Abhängigkeit der variierten Pendellänge.

gangssprünge bei den Lösungen des ÜFP könnten durch die Forderung von Stetigkeit bei direkten Verfahren mit einem stückweise polynomialen Eingangssignal verhindert werden. Daher kann man an dieser Stelle nicht schlussfolgern, dass die Formulierung als OSP prinzipiell Vorteile gegenüber der Formulierung als ÜFP bietet.

Problem 6: Aufschwingen des DWPS

Für das Aufschwingen des Dreifach-Wagen-Pendel-Systems konnten nur mithilfe des DKKV Lösungstrajektorien bestimmt werden. Dabei mussten die Verfahrensparameter mehrfach variiert werden. Auf eine aufwendigere automatisierte Variation der Parameter wurde aus Zeitgründen verzichtet. Das DMSV stellte sich für das Problem als so rechenaufwändig heraus, dass selbst auf eine manuelle Variation der Verfahrensparameter verzichtet werden musste. Auf Basis der vorherigen Ergebnisse erschien eine Formulierung als OSP am aussichtsreichsten, um Lösungstrajektorien bestimmen zu können. Außerdem wurde eine Beschränkung des Eingangsbetrages in Erwägung gezogen.

Schließlich konnte mit dem DKKV zwar eine Lösungstrajektorie bestimmt werden, diese wiesen jedoch mit etwa 500° sehr große Auslenkungen des zweiten Pendelkörpers gegenüber dem ersten Pendelkörper auf. Ein solches Überschlagen des zweiten (und damit auch dritten) Pendelkörpers scheint ein eher ungünstiges Verhalten darzustellen.

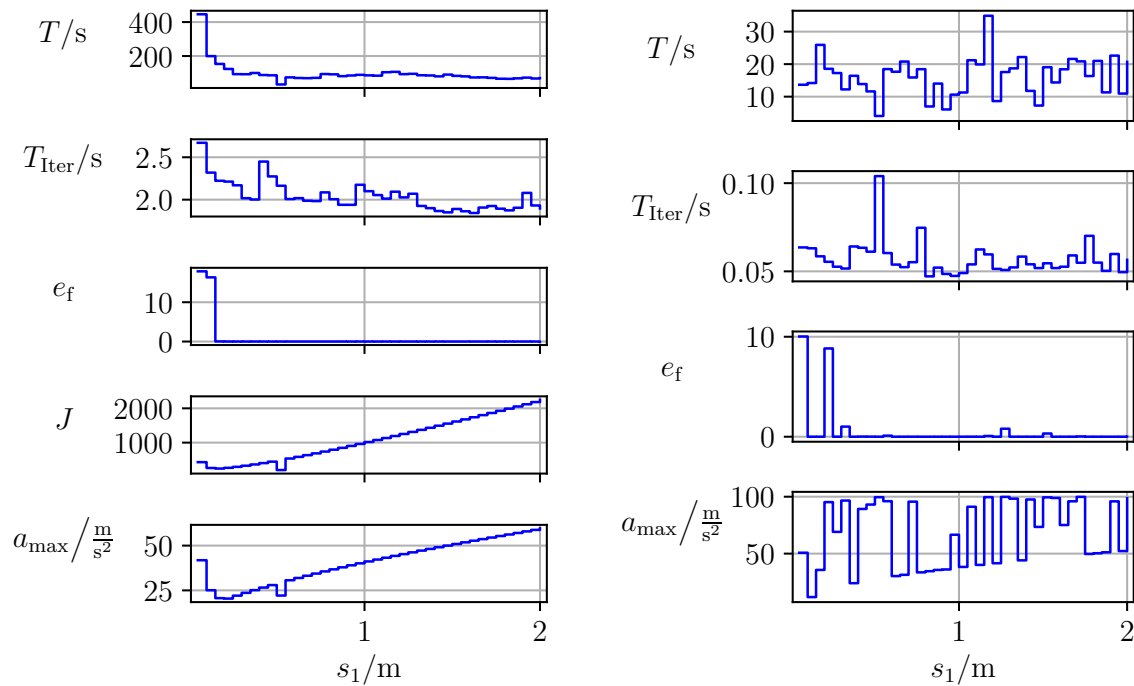


Abbildung 4.9 – Aufschwingen des TWPS: Vergleichskriterien für das DMSV (OSP, links) und das DKKV mit Eingangsbeschränkung (ÜFP, rechts) in Abhängigkeit der variierten Pendellänge.

Diese Trajektorie wird daher nicht näher beschrieben.

Stattdessen konnte mithilfe einer Beschränkung der Pendelwinkel und ohne Eingangsbeschränkung die in Abbildung 4.11 dargestellte Trajektorie bestimmt werden, welche zwar aufgrund der relativ geringen Genauigkeit keine Lösungstrajektorie ist, aber ein Verhalten des DWPS beschreibt, welches eher plausibel erscheint. Tabelle 4.11 zeigt neben der relativ großen Abweichung vom Endzustand, dass die Gesamtrechnzeit von knapp 47 min im Vergleich zu den bisher beobachteten Werten sehr hoch war.

Tabelle 4.11 – Aufschwingen des DWPS: Werte der Vergleichsgrößen für das DKKV.

Verfahren	n_{Iter}	T/s	T_{Iter}/s	e_f	J
DKKV ($N = 400$, $M = 4$)	349	2797,7	8,02	$9,8 \cdot 10^{-2}$	921,1

Wie erwartet stellte sich das Aufschwingen des DWPS als das schwierigste im Rahmen dieser Arbeit untersuchte Problem heraus. Als Ursache für die relativ hohe Schwierigkeit wird insbesondere der Unteraktuiertheitsgrad von 3 in Kombination mit der Nichtlinearität des Systems vermutet. Die Tatsache, dass der Wagen nur indirekt auf Pendelkörper 1 und dieser wiederum nur indirekt über Pendelkörper 2 auf Pendelkörper 3 einwirken

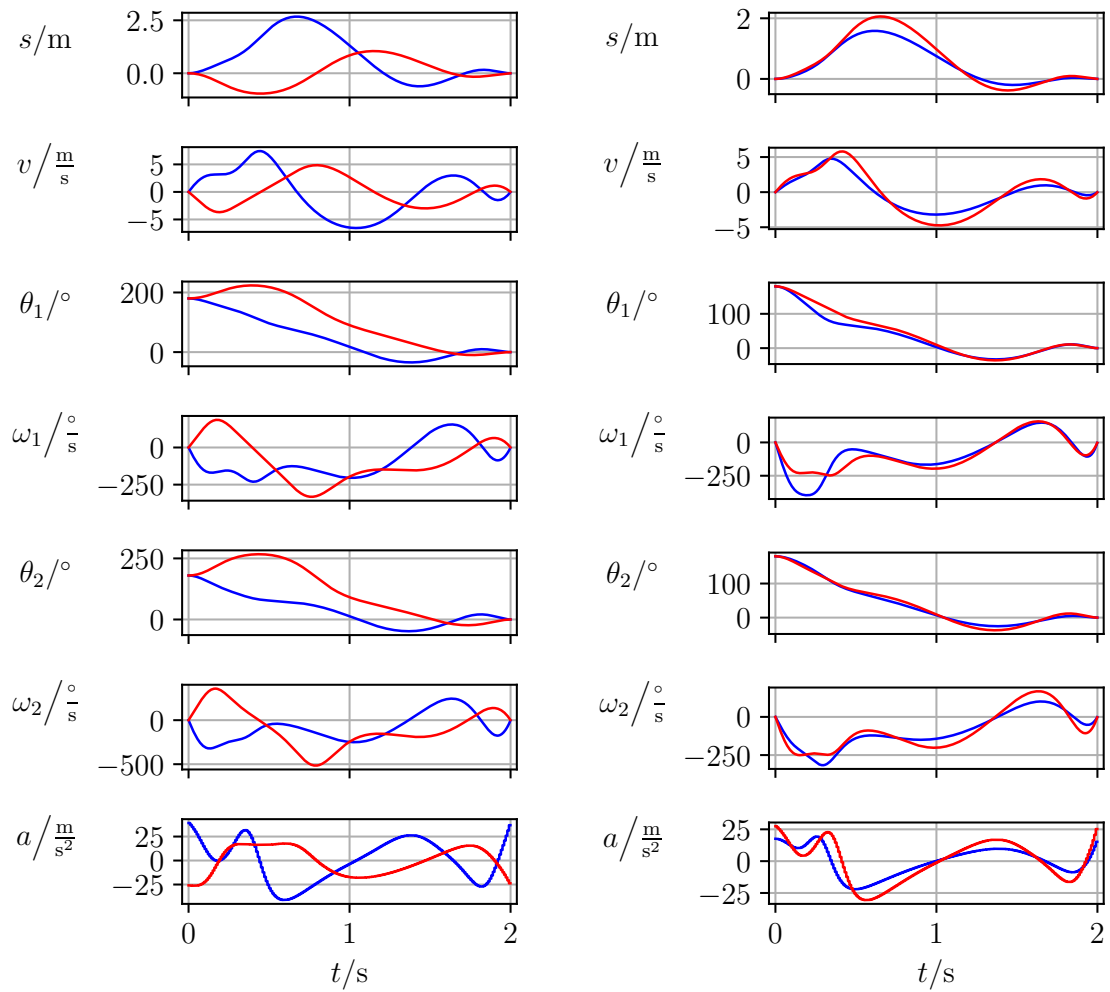


Abbildung 4.10 – Aufschwingen des TWPS: Lösungstrajektorien für das DKKV bei verschiedenen Pendellängen s_1 (links: $s_1 = 1$ m, $s_1 = 1,05$ m; rechts: $s_1 = 0,25$ m, $s_1 = 0,55$ m).

kann, könnte dabei eine wichtige Rolle spielen. Nicht zuletzt mit Blick auf die mechanische Struktur des Systems könnte man somit von einer Verkettung von Nichtlinearitäten sprechen, welche insgesamt zu einer vergleichsweise großen Nichtlinearität des Systems führt.

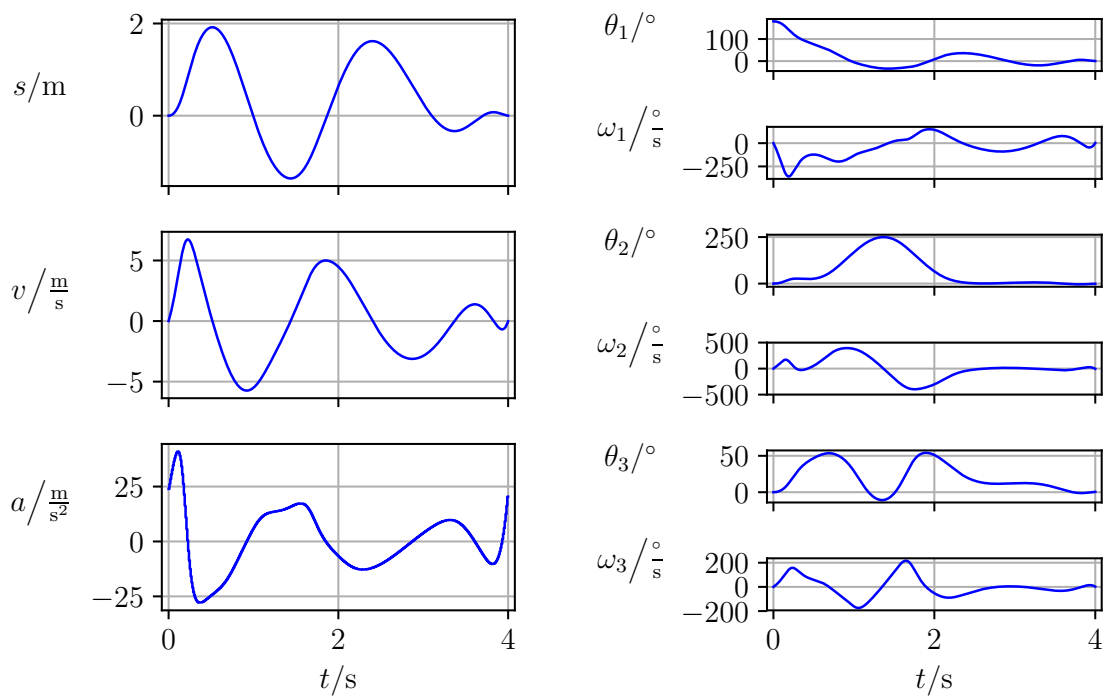


Abbildung 4.11 – Aufschwingen des DWPS: Trajektorie für das DKKV.

Kapitel 5

Zusammenfassung

In der vorliegenden Studienarbeit werden verschiedene Verfahren zur Trajektorienplanung dynamischer Systeme beschrieben und verglichen. In Kapitel 2 werden die theoretischen Grundlagen der im Rahmen dieser Arbeit implementierten Verfahren sowie der betrachteten dynamischen Systeme relativ ausführlich behandelt. Eine Ausnahme davon bilden lediglich die Verfahren zur Lösung von Problemen der statischen Optimierung, welche trotz ihrer großen Bedeutung für bestimmte Verfahren zur Trajektorienplanung nur kurz mit dem Hinweis auf weiterführende Literatur beschrieben werden.

In Kapitel 3 werden mit Überführungsproblemen und Optimalsteuerungsproblemen zwei unterschiedliche Problemstellungen und Herangehensweisen bei der Trajektorienplanung betrachtet. Eine Einordnung und Beschreibung von Verfahren zur Lösung von Optimalsteuerungsproblemen erfolgt auf Basis der in der Literatur vorherrschenden Unterscheidung zwischen direkten und indirekten Verfahren. Diesbezüglich ist bemerkenswert, dass die Lösung von Überführungsproblemen weitestgehend analog zu direkten Verfahren zur Lösung von Optimalsteuerungsproblemen erfolgen kann.

Als Ergebnis der Literaturrecherche haben sich die direkten und indirekten Schieß- und Kollokationsverfahren als besonders wichtige Verfahren zur Trajektorienplanung herausgestellt und wurden daher auch implementiert. Kapitel 3 befasst sich daher ausführlich mit diesen Verfahren und deren Implementierung. Auf weitere Verfahren wird nur kurz mit Verweisen auf weiterführende Literatur eingegangen.

Bei der Implementierung wurde auf das Software-Framework CasADi zurückgegriffen. Dieses erlaubt u. a. eine verfahrensunabhängige Formulierung von Problemen der Trajektorienplanung und deren Lösung mit verschiedenen Implementierungen numerischer Verfahren. Insbesondere nennenswert sind dabei die Schnittstellen zu Verfahren von Drittanbietern. In diesem Zusammenhang ist mit IPOPT ein Innere-Punkte-Verfahren hervorzuheben, welches bei allen implementierten Verfahren zum Einsatz kommt und somit insbesondere deren Vergleichbarkeit gewährleistet.

In Kapitel 4 wird die Leistungsfähigkeit der implementierten Verfahren anhand der Kriterien Genauigkeit, Zuverlässigkeit, Rechenaufwand und ggf. auch Optimalität bezüglich eines Kostenfunktionalis mithilfe von Beispielpunkten untersucht und verglichen.

Diese Beispielp Probleme umfassen die Trajektorienplanung für drei verschiedene Wagen-Pendel-Systeme. Im Rahmen der Untersuchungen wird auch auf die Schwierigkeit dieser Probleme eingegangen. Die Begriffe der Leistungsfähigkeit von Verfahren und Schwierigkeit von Problemen können dabei nicht voneinander getrennt gesehen werden und besitzen durch die zwangsläufig beschränkte Auswahl an Verfahren und Problemen lediglich eine relative Aussagekraft.

Bezüglich der Leistungsfähigkeit der Verfahren wurden schnell erhebliche Unterschiede sichtbar. So haben sich die indirekten Verfahren und das direkte Einfach-Schießverfahren insgesamt als wenig zuverlässig erwiesen. Mit diesen Verfahren konnte nur das Einfangen des Einfach-Wagen-Pendel-Systems gelöst werden. Für dieses konkrete Problem erwiesen sich jedoch zumindest die indirekten Verfahren mit Blick auf die genannten Kriterien als vergleichsweise leistungsfähig. Das direkte Mehrfach-Schießverfahren hat sich bei den meisten Problemen als durchaus leistungsfähig herausgestellt. Insgesamt hat sich das direkte (orthogonale) Kollokationsverfahren mit einem stückweise konstanten Ansatz für die Eingangstrajektorie insbesondere hinsichtlich der Zuverlässigkeit als das leistungsfähigste Verfahren erwiesen. So konnten z. B. nur mithilfe dieses Verfahrens Trajektorien für das Aufschwingen des Dreifach-Wagen-Pendel-Systems bestimmt werden. Insofern hat sich dieses Beispielp Problem erwartungsgemäß auch als das schwierigste erwiesen.

In Bezug auf die Schwierigkeit zeigen die Ergebnisse zum Aufschwingen des Einfach-Wagen-Pendel-Systems, dass konkrete Probleme der Trajektorienplanung durch eine partiell linearisierte Systemdarstellung wesentlich einfacher und insbesondere durch zu große Überführungszeiten erheblich schwieriger werden können. Im Rahmen dieser Arbeit wurden ausschließlich Beispielp Probleme ohne Eingangs- und Zustandsbeschränkung untersucht. Dies könnte ein Grund dafür sein, dass die untersuchte Variation einer Pendellänge beim Tandem-Wagen-Pendel-System kaum einen Einfluss auf die Schwierigkeit des Problems zu haben schien.

Wie die vorliegende Arbeit zeigt, lassen sich mithilfe eines Software-Frameworks wie CasADi relativ leistungsfähige Verfahren zur Trajektorienplanung mit überschaubarem Aufwand implementieren. Potential für Verbesserungen bzw. Erweiterungen besteht dabei konkret beim direkten Kollokationsverfahren mit einem stückweise polynomialen Ansatz für die Eingangstrajektorie sowie allgemein in Bezug auf eine sukzessive Verfeinerung der Diskretisierung (engl. mesh refinement) und die Initialisierung der Verfahren. Als weiteres Verfahren ist insbesondere die differentielle Dynamische Programmierung interessant. Außerdem sei darauf hingewiesen, dass die in dieser Arbeit untersuchten Verfahren streng genommen „nur“ Verfahren zur Transformation von Problemen der Trajektorienplanung auf Standardprobleme der numerischen Mathematik darstellen. Zur Verbesserung der Leistungsfähigkeit müssten letztlich somit auch die numerischen Verfahren zur Lösung dieser Standardprobleme betrachtet werden. Von zentraler Bedeutung sind dabei die Verfahren zur statischen Optimierung.

Literatur

- [1] Joel A. E. Andersson, Johan Åkesson und Moritz Diehl. “Dynamic optimization with CasADi”. In: *51st IEEE Conference on Decision and Control (CDC)*. 2012, S. 681–686. DOI: [10.1109/CDC.2012.6426534](https://doi.org/10.1109/CDC.2012.6426534).
- [2] Joel A. E. Andersson, Joris Gillis und Moritz Diehl. *User Documentation for CasADi v3.4.3*. 2018. URL: http://casadi.sourceforge.net/v3.4.3/users_guide/casadi-users_guide.pdf.
- [3] Joel A. E. Andersson u. a. *CasADi - Python examples*. 2018. URL: <https://github.com/casadi/casadi/tree/master/docs/examples/python>.
- [4] Joel A. E. Andersson u. a. “CasADi – A software framework for nonlinear optimization and optimal control”. In: *Mathematical Programming Computation* (In Press, 2018). URL: http://www.optimization-online.org/DB_FILE/2018/01/6420.pdf.
- [5] Uri M. Ascher, Robert M. M. Mattheij und Robert D. Russell. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. SIAM, 1995. DOI: [10.1137/1.9781611971231](https://doi.org/10.1137/1.9781611971231).
- [6] John T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. 2. Aufl. SIAM, 2010. DOI: [10.1137/1.9780898718577](https://doi.org/10.1137/1.9780898718577).
- [7] John T. Betts. “Survey of Numerical Methods for Trajectory Optimization”. In: *Journal of guidance, control, and dynamics* 21.2 (1998), S. 193–207. DOI: [10.2514/2.4231](https://doi.org/10.2514/2.4231).
- [8] Lorenz T. Biegler. *Nonlinear Programming. Concepts, Algorithms, and Applications to Chemical Processes*. SIAM, 2010. DOI: [10.1137/1.9780898719383](https://doi.org/10.1137/1.9780898719383).
- [9] Christofand Büskens und Dennis Wassel. “The ESA NLP Solver WORHP”. In: *Modeling and Optimization in Space Engineering*. Hrsg. von Giorgio Fasano und János D. Pintér. Springer New York, 2013, S. 85–110. DOI: [10.1007/978-1-4614-4469-5_4](https://doi.org/10.1007/978-1-4614-4469-5_4).
- [10] Mark Cannon. “Efficient nonlinear model predictive control algorithms”. In: *Annual Reviews in Control* 28.2 (2004), S. 229–237. DOI: [10.1016/j.arcontrol.2004.05.001](https://doi.org/10.1016/j.arcontrol.2004.05.001).
- [11] Moritz Diehl und Sébastien Gros. *Numerical Optimal Control (preliminary and incomplete draft)*. 2017. URL: <https://www.syscop.de/files/2017ss/NOC/script/book-NOCSE.pdf>.

- [12] Python Software Foundation. *Python 3.6.5 documentation*. 2018. URL: <https://docs.python.org/release/3.6.5/>.
- [13] Knut Graichen. *Methoden der Optimierung und optimalen Steuerung. Wintersemester 2017/2018*. Institut für Mess-, Regel- und Mikrotechnik, Universität Ulm, 2017. URL: https://www.uni-ulm.de/fileadmin/website_uni_ulm/iui.inst.110/Downloads/Vorlesung/Optimierung/Skript/Skript_MOOS_WS1718.pdf.
- [14] Knut Graichen und Michael Zeitz. “Inversionsbasierter Vorsteuerungsentwurf mit Ein- und Ausgangsbeschränkungen (Inversion-Based Feedforward Control Design under Input and Output Constraints)”. In: *at-Automatisierungstechnik* 54.4/2006 (2006), S. 187–199. DOI: [10.1524/auto.2006.54.4.187](https://doi.org/10.1524/auto.2006.54.4.187).
- [15] Andreas Griewank und Andrea Walther. *Evaluating Derivatives. Principles and Techniques of Algorithmic Differentiation*. 2. Aufl. SIAM, 2008. DOI: [10.1137/1.9780898717761](https://doi.org/10.1137/1.9780898717761).
- [16] Lars Grüne und Jürgen Pannek. *Nonlinear Model Predictive Control. Theory and Algorithms*. 2. Aufl. Springer, Cham, 2017. DOI: [10.1007/978-3-319-46024-6](https://doi.org/10.1007/978-3-319-46024-6).
- [17] Alan C. Hindmarsh und Radu Serban. “User documentation for CVODES, an ODE solver with sensitivity analysis capabilities”. In: *Livermore (California): Lawrence Livermore National Laboratory* 189 (2002). URL: https://computation.llnl.gov/casc/nsde/pubs/cvs_guide.pdf.
- [18] Alan C. Hindmarsh u. a. “SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers”. In: *ACM Transactions on Mathematical Software (TOMS)* 31.3 (2005), S. 363–396. DOI: [10.1145/1089014.1089020](https://doi.org/10.1145/1089014.1089020).
- [19] Carsten Knoll. “Regelungstheoretische Analyse- und Entwurfsansätze für unteraktuierte mechanische Systeme”. Dissertation. Technische Universität Dresden, 2016. URL: <http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-209765>.
- [20] Jens Lienig und Werner Krause. *Empfehlungen für die Ausarbeitung wissenschaftlicher Arbeiten*. Technische Universität Dresden, Fakultät Elektrotechnik und Informationstechnik, 2013. URL: http://www.et.tu-dresden.de/etit/uploads/media/EmpfehlungWissenschArbeiten2013_10.pdf.
- [21] Aaron Meurer u. a. “SymPy: symbolic computing in Python”. In: *PeerJ Computer Science* 3 (2017), e103. DOI: [10.7717/peerj-cs.103](https://doi.org/10.7717/peerj-cs.103).
- [22] Markos Papageorgiou, Marion Leibold und Martin Buss. *Optimierung. Statische, dynamische, stochastische Verfahren für die Anwendung*. 4. Aufl. Springer-Verlag Berlin Heidelberg, 2015. DOI: [10.1007/978-3-662-46936-1](https://doi.org/10.1007/978-3-662-46936-1).
- [23] Anil Rao. “A Survey of Numerical Methods for Optimal Control”. In: *Advances in the Astronautical Sciences* 135 (2010), S. 497–528. URL: <http://vdol.mae.ufl.edu/ConferencePublications/trajectorySurveyAAS.pdf>.
- [24] Klaus Röbenack. *Regler- und Beobachterentwurf für nichtlineare Systeme mit Hilfe des Automatischen Differenzierens*. Shaker Verlag, 2005. ISBN: 978-3-8322-4414-9.

- [25] Ralf Rothfuß, Joachim Rudolph und Michael Zeitz. “Flachheit: Ein neuer Zugang zur Steuerung und Regelung nichtlinearer Systeme”. In: *at-Automatisierungstechnik* 45.11 (1997), S. 517–525. DOI: [10.1524/auto.1997.45.11.517](https://doi.org/10.1524/auto.1997.45.11.517).
- [26] Patrick Rüdiger. *Trajectory-Methods*. *Github-Repository*. 2018. URL: <https://github.com/p-ruediger/Trajectory-Methods>.
- [27] Hans Rudolf Schwarz und Norbert Köckler. *Numerische Mathematik*. 7. Aufl. Vieweg+Teubner Verlag | Springer Fachmedien Wiesbaden GmbH, 2009. DOI: [10.1007/978-3-8348-9282-9](https://doi.org/10.1007/978-3-8348-9282-9).
- [28] Radu Serban und Alan C. Hindmarsh. “CVODES: the sensitivity-enabled ODE solver in SUNDIALS”. In: *ASME 2005 international design engineering technical conferences and computers and information in engineering conference*. American Society of Mechanical Engineers. 2005, S. 257–269. DOI: [10.1115/DETC2005-85597](https://doi.org/10.1115/DETC2005-85597).
- [29] Josef Stoer und Roland Bulirsch. *Introduction to Numerical Analysis*. 3. Aufl. Texts in Applied Mathematics 12. Springer-Verlag New York, 2002. DOI: [10.1007/978-0-387-21738-3](https://doi.org/10.1007/978-0-387-21738-3).
- [30] Yuval Tassa, Nicolas Mansard und Emo Todorov. “Control-limited differential dynamic programming”. In: *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE. 2014, S. 1168–1175. DOI: [10.1109/ICRA.2014.6907001](https://doi.org/10.1109/ICRA.2014.6907001).
- [31] Andreas Wächter. “Short Tutorial: Getting Started With Ipopt in 90 Minutes”. In: *Combinatorial Scientific Computing*. Hrsg. von Uwe Naumann u. a. Dagstuhl Seminar Proceedings 09061. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2009. URL: <http://drops.dagstuhl.de/opus/volltexte/2009/2089>.
- [32] Andreas Wächter und Lorenz T. Biegler. “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming”. In: *Mathematical programming* 106.1 (2006), S. 25–57. DOI: [10.1007/s10107-004-0559-y](https://doi.org/10.1007/s10107-004-0559-y).

Abkürzungsverzeichnis

AWP Anfangswertproblem

DESV Direktes Einfach-Schießverfahren

DGL Differentialgleichung

DKKV Direktes Kollokationsverfahren

DMSV Direktes Mehrfach-Schießverfahren

DWPS Dreifach-Wagen-Pendel-System

ESV Einfach-Schießverfahren

EWPS Einfach-Wagen-Pendel-System

GNB Gleichungsnebenbedingungen

IESV Indirektes Einfach-Schießverfahren

IKKV Indirektes Kollokationsverfahren

IMSV Indirektes Mehrfach-Schießverfahren

KKV Kollokationsverfahren

MSV Mehrfach-Schießverfahren

NLGS nichtlineares algebraisches Gleichungssystem

NLP nichtlineares Programm (engl. nonlinear program)

OSP Optimalsteuerungsproblem

RKV Runge-Kutta-Verfahren

RWP Randwertproblem

TPP Trajektorienplanungsproblem

TWPS Tandem-Wagen-Pendel-System

UNB Ungleichungsnebenbedingungen

ÜFP Überführungsproblem

Abbildungsverzeichnis

2.1	Schematische Darstellung des Kollokationsverfahrens für $M = 3$	12
2.2	Schematische Darstellung des Einfach-Wagen-Pendel-Systems [19, Abb. 4.3, S. 38].	17
3.1	Schematische Darstellung der Vorgehensweise zur Trajektorienplanung mit Schieß- und Kollokationsverfahren.	22
4.1	Schematische Darstellung des TWPS [19, Abb. 4.3, S. 38].	36
4.2	Schematische Darstellung des DWPS [19, Abb. 4.3, S. 38 und Abb. 4.6, S. 67].	36
4.3	Einfangen des EWPS: Eingangstrajektorien.	40
4.4	Einfangen des EWPS: Zustandstrajektorien (links: DESV, IESV; rechts: DMSV, DKKV, DKKV*, IMSV, IKKV).	41
4.5	Aufschwingen des EWPS (OSP): Lösungstrajektorien (links: DMSV, DKKV ($M = 4$); rechts: DKKV ($M = 8$)).	44
4.6	Aufschwingen des EWPS (ÜFP): Lösungstrajektorien (links: DMSV; rechts: DKKV ($M = 4$), DKKV ($M = 8$)).	45
4.7	Aufschwingen des EWPS: Vergleichskriterien für das DKKV (links) und DMSV (rechts) in Abhängigkeit der Überführungszeit.	46
4.8	Aufschwingen des TWPS (OSP): Vergleichskriterien für das DKKV ohne (links) und mit Eingangsbeschränkung (rechts) in Abhängigkeit der variierten Pendellänge.	49
4.9	Aufschwingen des TWPS: Vergleichskriterien für das DMSV (OSP, links) und das DKKV mit Eingangsbeschränkung (ÜFP, rechts) in Abhängigkeit der variierten Pendellänge.	50
4.10	Aufschwingen des TWPS: Lösungstrajektorien für das DKKV bei verschiedenen Pendellängen s_1 (links: $s_1 = 1$ m, $s_1 = 1,05$ m; rechts: $s_1 = 0,25$ m, $s_1 = 0,55$ m).	51
4.11	Aufschwingen des DWPS: Trajektorie für das DKKV.	52

Tabellenverzeichnis

3.1	Übersicht der implementierten Verfahren.	30
3.2	Jacobi-Matrix je nach Verfahren.	32
4.1	Übersicht der Beispielprobleme ($\mathbf{x}_f := \mathbf{0}$, $t_0 := 0$).	37
4.2	Werte der Parameter der Beispielsysteme ($m_0 = 1$ kg, $g = 9,81$ ms ⁻²).	38
4.3	Einfangen des EWPS: Werte der Vergleichsgrößen nach Verfahren.	38
4.4	EWPS: Dimension und Besetzungsdichte der Jacobi-Matrix des je nach Verfahren entstehenden NLGS.	39
4.5	Aufschwingen des EWPS (OSP): Werte der Vergleichsgrößen nach Verfahren.	43
4.6	Aufschwingen des EWPS (ÜFP): Werte der Vergleichsgrößen nach Verfahren.	43
4.7	Aufschwingen des nicht partiell linearisierten EWPS (OSP): Werte der Vergleichsgrößen nach Verfahren.	44
4.8	Aufschwingen des nicht partiell linearisierten EWPS (ÜFP): Werte der Vergleichsgrößen nach Verfahren.	45
4.9	Aufschwingen des EWPS mit variabler Überführungszeit: Zuverlässigkeit der Verfahren und durchschnittliche Werte der Vergleichsgrößen für Lösungstrajektorien.	47
4.10	Aufschwingen des TWPS: Zuverlässigkeit der Verfahren und durchschnittliche Werte der Vergleichsgrößen für die Lösungstrajektorien.	48
4.11	Aufschwingen des DWPS: Werte der Vergleichsgrößen für das DKKV.	50