

Analyse des Einflusses von Messdaten auf die Parameter dynamischer Modelle mittels parametrischer Sensitivitätsanalyse

Masterarbeit

zur Erlangung des Grades
Master of Science

vorgelegt am

4. September 2020



von

Marek Wiesner

Matrikelnummer 4125591

1. Gutachter: Prof. Dr. Christof Büskens, Universität Bremen
2. Gutachter: Dr. Matthias Knauer, Universität Bremen

Inhaltsverzeichnis

Abbildungsverzeichnis	V
Tabellenverzeichnis	VII
Symbolverzeichnis	IX
1 Einleitung	1
1.1 Problemstellung und Zielsetzung	1
1.2 Aufbau der Arbeit	3
2 Grundlagen der nichtlinearen Optimierung	5
2.1 Grundbegriffe der nichtlinearen Optimierung	5
2.2 Optimalitätsbedingungen	7
2.3 Parametrische Sensitivitätsanalyse	9
2.4 Numerische Lösung nichtlinearer Optimierungsprobleme mittels SQP-Verfahren	13
2.4.1 SQP-Verfahren	13
2.4.2 WORHP	17
3 Stückweise Polynome und B-Splines	19
3.1 Stückweise Polynome	19
3.2 Basis der abgeschnittenen Potenzfunktionen	22
3.3 B-Splines	26
3.3.1 Basis der B-Splines	28
3.3.2 Ableitung von B-Splines	32
3.3.3 Effiziente Berechnung von B-Splines	35

4	Grundlagen der Parameteridentifikation	41
4.1	Explizite Parameteridentifikationsprobleme	42
4.1.1	Überblick über numerische Lösungsverfahren von Least-Squares-Problemen	43
4.1.2	Alternative Zielfunktionen	43
4.2	Dynamische Parameteridentifikationsprobleme	44
4.2.1	Überblick über numerische Lösungsverfahren von dynamischen Parameteridentifikationsproblemen	46
4.2.2	Kollokationsmethode	48
4.2.2.1	Approximation der Daten	49
4.2.2.2	Identifikation der Modellparameter	51
4.2.2.3	Vor- und Nachteile der Kollokationsmethode	52
4.2.2.4	Erweiterungen der Kollokationsmethode	53
5	Analyse des Einflusses von Daten auf die Parameter dynamischer Modelle	57
5.1	Motivation	57
5.2	Analyse des Einflusses von Daten in dynamischen Parameteridentifikationsproblemen mittels parametrischer Sensitivitätsanalyse	59
5.2.1	Approximation der Daten durch stückweise Polynome	61
5.2.2	Identifikation der Modellparameter und Berechnung der Sensitivitäten	62
5.2.3	Interpretation der Sensitivitäten	64
5.2.3.1	Beurteilung des Einflusses von Daten durch die kompakten Träger von B-Splines	64
5.2.3.2	Beurteilung des Einflusses von Daten durch die Sensitivitäten der Koeffizienten der B-Splines bezüglich der Datenpunkte	66
5.2.4	Anmerkungen zur Methode	69
5.2.4.1	Vergleich des Einflusses der Daten verschiedener Zustände und Steuerungen	69
5.2.4.2	Alternative Bewertung des Einflusses der Daten	69
5.2.4.3	Grenzen der Methode	70
5.2.4.4	Alternative Lösungsverfahren des Parameteridentifikationsproblems	70

6	Numerische Auswertungen	71
6.1	Visualisierung des Einflusses von Daten auf die Modellparameter parameterabhängiger Modelle	72
6.2	Analyse des Einflusses von Daten auf den Parameter eines einfachen dynamischen Modells	74
6.2.1	Approximation der Daten	74
6.2.2	Identifikation des Modellparameters und Berechnung der Sensitivitäten	75
6.2.3	Interpretation der Sensitivitäten	77
6.2.3.1	Analyse der Änderung des Modellparameters	77
6.2.3.2	Analyse des mittleren absoluten Dateneinflusses	81
6.3	Analyse des Einflusses von Daten auf die Parameter eines Einspurmodells	83
6.3.1	Generieren von Daten	85
6.3.2	Approximation der Daten	87
6.3.3	Identifikation der Modellparameter und Berechnung der Sensitivitäten	88
6.3.4	Interpretation der Sensitivitäten	90
6.3.4.1	Analyse des Einflusses der Daten auf den Modellparameter k	90
6.3.4.2	Analyse des Einflusses der Daten auf den Modellparameter L	93
6.3.4.3	Analyse des Einflusses der Daten auf den Modellparameter r	97
7	Fazit	99
7.1	Zusammenfassung	99
7.2	Ausblick	101
A	Details zur Herleitung der Algorithmen von de Boor	105
A.1	Details zu de Boors erstem Algorithmus	105
A.2	Details zu de Boors zweitem Algorithmus	106
B	Rechnungen zur Analyse des Einflusses von Daten auf den Parameter eines einfachen dynamischen Modells	109
B.1	Identifikation des Modellparameters und Berechnung der Sensitivitäten	109
B.2	Berechnung der Sensitivitäten bezüglich der Datenpunkte	110

C	Ergänzungen zur Analyse des Einflusses von Daten auf die Parameter eines Einspurmodells	113
C.1	Details zur Approximation der Daten	113
C.2	Ergänzende Abbildungen des mittleren absoluten Einflusses der Daten auf den Modellparameter r	117
Literatur		119

Abbildungsverzeichnis

3.1	Stückweises Polynom vom Grad $d \leq 2$ mit den Sprungstellen $\xi = (0, 1, 2, 3, 4)$	20
3.2	ν -reguläre stückweise Polynome vom Grad $d \leq 2$ mit den Sprungstellen $\xi = (0, 1, 2, 3, 4)$	21
3.3	Abgeschnittene Potenzfunktionen $\varphi_{i,j}$ vom Grad $j = 1, 2$ mit den Sprungstellen $\xi = (0, 1, 2, 3, 4)$	23
3.4	ν -reguläres stückweises Polynom zur Illustration der schlechten Kondition der Basis der abgeschnittenen Potenzfunktionen	25
3.5	B-Splines vom Grad $d = 1, 2$ und mit verschiedenen Knotenfolgen	29
3.6	Basisdarstellung durch B-Splines vom Grad $d = 2$	30
3.7	Ableitungen von B-Splines	34
3.8	Zusammenhang zwischen den Polynomen $r_i^{[k]}(x)$	38
3.9	Abhängigkeiten der B-Splines mit Werten ungleich Null auf dem Intervall $[\tau_j, \tau_{j+1})$	39
4.1	Schematisches Vorgehen bei der Kollokationsmethode	52
4.2	Schematisches Vorgehen bei der erweiterten Kollokationsmethode	55
5.1	Schematische Darstellung der Analyse des Einflusses von Daten in dynamischen Parameteridentifikationsproblemen	61
5.2	Änderung der Approximation der Daten infolge einer Störung des Koeffizienten des i -ten B-Splines	65
5.3	Probleme bei der Beurteilung des Einflusses von Daten durch die kompakten Träger von B-Splines	66
6.1	Visualisierung des Einflusses von Daten auf die Modellparameter parameterabhängiger Modelle	73

6.2	Daten des Zustands und der Steuerung sowie deren Approximationen durch ν -reguläre stückweise Polynome im einfachen dynamischen Modell	75
6.3	Änderung des Modellparameters des einfachen dynamischen Modells .	79
6.4	Mittlerer absoluter Einfluss verschiedener Datenbereiche auf den Modellparameter des einfachen dynamischen Modells	82
6.5	Schematische Darstellung des Einspurmodells	84
6.6	Trajektorie des Fahrzeugs im untersuchten Fahrmanöver	86
6.7	Lage der Sprungstellen der ν -regulären stückweisen Polynome zur Approximation der Daten des Einspurmodells	87
6.8	Daten und Approximationen der Daten durch ν -reguläre stückweise Polynome des Einspurmodells	89
6.9	Mittlere absolute Einflüsse der Daten des Einspurmodells auf den Modellparameter k	91
6.10	Summierter mittlerer absoluter Einfluss der Daten des Einspurmodells auf den Modellparameter k , visualisiert auf der Trajektorie des Fahrzeugs	94
6.11	Mittlere absolute Einflüsse der Daten des Einspurmodells auf den Modellparameter L	95
6.12	Summierter mittlerer absoluter Einfluss der Daten des Einspurmodells auf den Modellparameter L , visualisiert auf der Trajektorie des Fahrzeugs	96
6.13	Summierter mittlerer absoluter Einfluss der Daten des Einspurmodells auf den Modellparameter r , visualisiert auf der Trajektorie des Fahrzeugs	97
C.1	Mittlere absolute Einflüsse der Daten des Einspurmodells auf den Modellparameter r	117

Tabellenverzeichnis

6.1	Spezifikationen der verwendeten Hardware und Software	71
6.2	Zustände und Steuerungen des Einspurmodells	85
6.3	Mittelwerte und Standardabweichungen der Fehler der generierten Daten des Einspurmodells	87

Symbolverzeichnis

Grundlagen der nichtlinearen Optimierung

n_z	Anzahl der Optimierungsvariablen	(S. 5)
n_g	Anzahl der Ungleichungsnebenbedingungen	(S. 5)
n_h	Anzahl der Gleichungsnebenbedingungen	(S. 5)
z	Optimierungsvariablen	(S. 5)
F	Zielfunktion	(S. 5)
g	Ungleichungsnebenbedingungen	(S. 5)
h	Gleichungsnebenbedingungen	(S. 5)
\mathcal{I}	Indizes der Ungleichungsnebenbedingungen	(S. 5)
\mathcal{E}	Indizes der Gleichungsnebenbedingungen	(S. 5)
Ω	zulässige Menge	(S. 6)
\mathcal{A}	Indexmenge der aktiven Ungleichungsnebenbedingungen	(S. 6)
λ	Lagrange-Multiplikatoren der Ungleichungsnebenbedingungen	(S. 7)
μ	Lagrange-Multiplikatoren der Gleichungsnebenbedingungen	(S. 7)
L	Langrange-Funktion	(S. 7)
\mathcal{C}	kritischer Kegel	(S. 8)
q	Störparameter	(S. 10)
H	Approximation der Hesse-Matrix der Lagrange-Funktion	(S. 15)
Δz	Suchrichtung der Optimierungsvariable im SQP-Verfahren	(S. 15)

Stückweise Polynome und B-Splines

d	Grad von (ν -regulären stückweisen) Polynomen	(S. 19)
Π_d	Vektorraum der Polynome vom Grad $\leq d$	(S. 19)
$k + 1$	Anzahl der Sprungstellen eines stückweisen Polynoms	(S. 20)
ξ	Sprungstellen eines stückweisen Polynoms	(S. 20)
$\Pi_{d,\xi}$	Vektorraum der stückweisen Polynome mit Sprungstellen ξ vom Grad $\leq d$	(S. 20)

ν	Vektor der Regularitäten an den Sprungstellen von ν -regulären stückweisen Polynomen	(S. 21)
$\Pi_{d,\xi,\nu}$	Vektorraum der ν -regulären stückweisen Polynome vom Grad $\leq d$	(S. 21)
$\varphi_{i,j}$	abgeschnittene Potenzfunktion	(S. 22)
$l + 1$	Anzahl der Knoten des B-Splines $B_{i,d,\tau}$	(S. 26)
τ	Knotenfolge des B-Splines $B_{i,d,\tau}$	(S. 26)
$B_{i,d,\tau}$	B-Spline vom Grad d und der Knotenfolge τ	(S. 26)
$I_{d,\tau}$	Basisintervall der B-Splines $B_{i,d,\tau}$	(S. 26)

Grundlagen der Parameteridentifikation

n_ψ	Anzahl der Größen im expliziten parameterabhängigen Modell	(S. 42)
n_θ	Anzahl der Modellparameter im expliziten parameterabhängigen Modell	(S. 42)
θ	Parametervektor im expliziten parameterabhängigen Modell	(S. 42)
Θ	zulässige Menge des Parametervektors im expliziten parameterabhängigen Modell	(S. 42)
t	Zeit	(S. 42)
ψ	explizites parameterabhängiges Modell	(S. 42)
n_t	Anzahl der Zeitpunkte	(S. 42)
t_i	diskrete Zeitpunkte	(S. 42)
t_0	Startzeit	(S. 42)
t_f	Endzeit	(S. 42)
\bar{y}	Daten im expliziten Parameteridentifikationsproblem	(S. 42)
n_x	Anzahl der Zustände im dynamischen parameterabhängigen Modell	(S. 44)
n_u	Anzahl der Steuerungen im dynamischen parameterabhängigen Modell	(S. 44)
n_p	Anzahl der Modellparameter im dynamischen parameterabhängigen Modell	(S. 44)
p	Parametervektor im dynamischen parameterabhängigen Modell	(S. 44)
\mathcal{P}	zulässige Menge des Parametervektors im dynamischen parameterabhängigen Modell	(S. 44)
f	Vektorfeld des dynamischen parameterabhängigen Modells	(S. 44)
x	Zustandsvektor im dynamischen parameterabhängigen Modell	(S. 44)
u	Steuervektor im dynamischen parameterabhängigen Modell	(S. 44)
$n_{\bar{x}}$	Anzahl der gemessenen Zustände im dynamischen Parameteridentifikationsproblem	(S. 45)

\bar{x}	Daten der Zustände im dynamischen Parameteridentifikationsproblem	(S. 45)
\bar{u}	Daten der Steuerungen im dynamischen Parameteridentifikationsproblem	(S. 45)
u_{app}	Approximation der Daten der Steuerungen	(S. 45)
σ	Zuordnung zwischen den Indizes der gemessenen Zustände und den Indizes der Zustände	(S. 45)
x_σ	Vektor der gemessenen Zustände im dynamischen Parameteridentifikationsproblem	(S. 45)
x_{app}	Approximation der Daten der Zustände	(S. 48)
d_j	Grad der B-Splines zur Approximation der Daten des j -ten Zustands	(S. 49)
τ_j	Knotenfolge der B-Splines zur Approximation der Daten des j -ten Zustands	(S. 49)
$l_j + 1$	Anzahl der Knoten der B-Splines zur Approximation der Daten des j -ten Zustands	(S. 49)
c_j	Koeffizienten der B-Splines zur Approximation der Daten des j -ten Zustands	(S. 49)
N_j	Anzahl der B-Splines zur Approximation der Daten des j -ten Zustands	(S. 49)
E_j	Fehler zwischen dem Vektorfeld des dynamischen Modells und der Ableitung der Approximation der Daten der Zustände	(S. 51)
e_j	diskreter Fehler zwischen dem Vektorfeld des dynamischen Modells und der Ableitung der Approximation der Daten der Zustände	(S. 51)

Analyse des Einflusses von Daten auf die Parameter dynamischer Modelle

Einige Symbole dieses Kapitels gleichen denen des Kapitels *Grundlagen der Parameteridentifikation* und werden nicht erneut aufgeführt.

d_j^x	Grad der B-Splines zur Approximation der Daten des j -ten Zustands	(S. 61)
τ_j^x	Knotenfolge der B-Splines zur Approximation der Daten des j -ten Zustands	(S. 61)
$l_j^x + 1$	Anzahl der Knoten der B-Splines zur Approximation der Daten des j -ten Zustands	(S. 61)

\tilde{x}_{app}	Approximation der Daten der Zustände	(S. 61)
c_j^x	Koeffizienten der B-Splines zur Approximation der Daten des j -ten Zustands	(S. 62)
d_j^u	Grad der B-Splines zur Approximation der Daten der j -ten Steuerung	(S. 62)
\tilde{u}_{app}	Approximation der Daten der Steuerungen	(S. 62)
τ_j^u	Knotenfolge der B-Splines zur Approximation der Daten der j -ten Steuerung	(S. 62)
$l_j^u + 1$	Anzahl der Knoten der B-Splines zur Approximation der Daten der j -ten Steuerung	(S. 62)
c_j^u	Koeffizienten der B-Splines zur Approximation der Daten der j -ten Steuerung	(S. 62)
N_j^x	Anzahl der B-Splines zur Approximation der Daten des j -ten Zustands	(S. 62)
N_j^u	Anzahl der B-Splines zur Approximation der Daten der j -ten Steuerung	(S. 62)
$c_j^{x,0}$	Nominalwerte der Koeffizienten der B-Splines zur Approximation der Daten des j -ten Zustands	(S. 63)
$c_j^{u,0}$	Nominalwerte der Koeffizienten der B-Splines zur Approximation der Daten der j -ten Steuerung	(S. 63)
p^*	optimaler Vektor der Modellparameter	(S. 63)
Δp	Änderung des Modellparameters	(S. 64)
Δc_i	Störung des Koeffizienten des i -ten B-Splines	(S. 64)
\mathcal{D}	Teilmenge der Daten	(S. 66)
\bar{x}^0	Nominalwerte der Daten vom Zustand x	(S. 67)
c^*	optimale Koeffizienten der B-Splines	(S. 67)
$\tilde{\Delta} c_i$	Änderung des Koeffizienten des i -ten B-Splines	(S. 67)
$\Delta \bar{x}_k$	Störung des k -ten Datenpunkts vom Zustand x	(S. 67)
κ	mittlerer absoluter Dateneinfluss	(S. 70)

Numerische Auswertungen

Einige Symbole dieses Kapitels gleichen denen des Kapitels *Analyse des Einflusses von Daten auf die Modellparameter dynamischer Modelle* und werden nicht erneut aufgeführt.

x	x -Position des Fahrzeugs	(S. 84)
y	y -Position des Fahrzeugs	(S. 84)
v	Geschwindigkeit des Fahrzeugs	(S. 84)

ψ	Orientierung des Fahrzeugs	(S. 84)
δ	Lenkwinkel des Fahrzeugs	(S. 84)
a	Beschleunigung des Fahrzeugs	(S. 84)
σ	Lenkwinkelgeschwindigkeit des Fahrzeugs	(S. 84)
k	Verlust bzw. Verstärkung der Beschleunigung	(S. 84)
L	Radstand des Fahrzeugs	(S. 84)
r	Verlust bzw. Verstärkung der Lenkwinkelgeschwindigkeit	(S. 84)
χ	Modellgröße im Einspurmodell	(S. 86)
\tilde{x}_{app}	Approximation der Daten der x-Position	(S. 87)
\tilde{y}_{app}	Approximation der Daten der y-Position	(S. 87)
\tilde{v}_{app}	Approximation der Daten der Geschwindigkeit	(S. 87)
$\tilde{\psi}_{\text{app}}$	Approximation der Daten der Orientierung	(S. 87)
$\tilde{\delta}_{\text{app}}$	Approximation der Daten des Lenkwinkel	(S. 87)
\tilde{a}_{app}	Approximation der Daten der Beschleunigung	(S. 87)
$\tilde{\sigma}_{\text{app}}$	Approximation der Daten der Lenkwinkelgeschwindigkeit	(S. 87)
\mathcal{D}_i^χ	Einteilung der Daten der Modellgröße χ in gleich große, zeitlich zusammenhängende Bereiche	(S. 90)
κ^p	mittlerer absoluter Einfluss der Daten auf den Modellparameter $p \in \{k, L, r\}$	(S. 90)
$\tilde{\mathcal{D}}_i^\chi$	Einteilung der Daten der Modellgröße χ in zeitlich zusammenhängende Bereiche, während denen das Fahrzeug die gleiche Distanz zurücklegt	(S. 93)
κ_Σ^p	summierter mittlerer absoluter Einfluss der Daten auf den Modellparameter $p \in \{k, L, r\}$	(S. 93)

Kapitel 1

Einleitung

Verschiedenste Prozesse können durch mathematische Modelle beschrieben werden. Die Anwendungsbereiche sind vielfältig und reichen von der Modellierung der Bewegung von Kraftfahrzeugen [66] über die Modellierung des Wiedereintritts einer Raumkapsel in die Atmosphäre des Mars [70] bis hin zur Modellierung des ökonomischen Wachstums von Volkswirtschaften [33] und der Ausbreitung von Pandemien [81]. Mithilfe mathematischer Modelle können die Prozesse zum einen simuliert und auf diese Weise neue Erkenntnisse über den Prozess gewonnen werden. Zum anderen können die Modelle genutzt werden, um die Prozesse gezielt zu beeinflussen und ein gewünschtes Verhalten zu erreichen. So können etwa mithilfe von Modellen der Bewegung von Kraftfahrzeugen autonome Fahrmanöver durchgeführt werden [59]. Mathematische Modelle sind somit essentiell für die Entwicklung von Spitzentechnologien wie dem autonomen Fahren. Modelle zur Ausbreitung von Pandemien können beispielsweise dabei helfen, Strategien zur Eindämmung des Infektionsgeschehens [42, 84] sowie optimale Impfstrategien der Bevölkerung zu entwickeln [21]. Erstere sind nicht zuletzt wegen der COVID-19-Pandemie [78] in den Fokus des gesellschaftlichen Lebens gerückt.

1.1 Problemstellung und Zielsetzung

Mathematische Modelle hängen oftmals von verschiedenen Parametern ab, um möglichst viele Situationen beschreiben zu können. Um einen bestimmten Prozess zu simulieren, müssen die Werte der Modellparameter ermittelt werden. In einigen Fällen können sie gemessen werden. Dies ist jedoch nicht immer möglich. Stattdessen werden typischerweise Messdaten von dem Prozess, der modelliert werden soll, aufgenommen. Die unbekannt Parameter werden so bestimmt, dass die Messdaten möglichst gut durch das mathematische Modell beschrieben werden. Dieses Verfahren wird Parameteridentifikation genannt und führt auf ein nichtlineares Optimierungsproblem, welches mithilfe numerischer Algorithmen gelöst werden kann.

Wird der Prozess nicht hinreichend genau durch das mathematische Modell beschrieben, so ist eine Identifikation der Modellparameter oftmals nicht möglich. Auch falls das mathematische Modell den Prozess adäquat beschreibt, ist nicht garantiert, dass die Modellparameter identifiziert werden können. Dies ist zum Beispiel der Fall, wenn die Informationen zum Bestimmen der Modellparameter nicht in den Daten enthalten sind.

Eine statistische Kenngröße, mit der bewertet werden kann, wie gut die Parameter eines mathematischen Modells anhand gegebener Daten identifiziert werden können, ist die sogenannte Fisher-Information. Mithilfe der Fisher-Information können die Varianzen und Kovarianzen der identifizierten Modellparameter abgeschätzt werden. Dies ist eine globale Information, weil die Daten als Ganzes betrachtet werden. Das Ziel der vorliegenden Arbeit ist die Entwicklung einer Methode zum Analysieren des Einflusses bestimmter Bereiche der Messdaten auf die Modellparameter mathematischer Modelle. Im Gegensatz zur Fisher-Information handelt es sich hierbei um eine lokale Information. Die Arbeit konzentriert sich dabei auf sogenannte dynamische Modelle, welche sich aus der Modellierung von zeitabhängigen Prozessen ergeben.

Das Wissen darüber, welche Messdaten tatsächlich einen Einfluss auf bestimmte Modellparameter haben, kann zu einem besserem Verständnis des Modells führen. Oftmals bilden physikalische Überlegungen die Grundlage der mathematischen Modellierung. Um den Einfluss verschiedener physikalischer Effekte zu berücksichtigen, werden Terme zum Modell hinzugefügt, welche typischerweise von Modellparametern abhängen. Durch Vorwissen ist es oftmals möglich, gewisse Bereiche in den Daten einem bestimmten physikalischen Effekt zuzuordnen. Mit der vorgestellten Methode kann analysiert werden, ob die Modellparameter, welche den Effekt im Modell beschreiben sollen, tatsächlich durch diese Daten beeinflusst werden. Ist dies nicht der Fall, so deutet das darauf hin, dass der Effekt nicht korrekt durch das Modell beschrieben wird. Dieses Wissen kann genutzt werden, um das Modell zu verbessern.

Mithilfe der in dieser Arbeit entwickelten Methode kann darüber hinaus analysiert werden, ob es Bereiche in den Messdaten gibt, welche keinen der Modellparameter beeinflussen. Dies würde darauf hindeuten, dass diese Daten nicht wichtig für die Parameteridentifikation sind. Bei der Planung von weiteren Versuchen kann daher vermieden werden, solche Daten zu erzeugen. Dies ist insbesondere interessant, wenn die Durchführung der Versuche mit hohen Kosten verbunden ist. Um Kosten einzusparen, ist es wünschenswert, lediglich notwendige Versuche durchzuführen.

Die entwickelte Methode zum Analysieren des Einflusses der Messdaten auf die Parameter dynamischer Modelle kann in mehrere Schritte aufgeteilt werden. Zunächst werden die Messdaten durch stückweise Polynome approximiert. Als Basisfunktionen werden sogenannte B-Splines verwendet. Auf diese Weise werden die Daten geglättet und mögliches Rauschen reduziert. Als nächstes werden die Modellparameter des

dynamischen Modells identifiziert. Dabei werden statt der Daten die Approximationen der Daten verwendet. In dem resultierenden Optimierungsproblem können die Koeffizienten der B-Splines in der Basisdarstellung der Approximationen als Störparameter interpretiert werden. Mithilfe von Methoden der parametrischen Sensitivitätsanalyse können schließlich die Sensitivitäten der optimalen Modellparameter bezüglich dieser Koeffizienten bestimmt werden. Um den Einfluss bestimmter Bereiche in den Daten auf die Modellparameter zu beurteilen, werden die Sensitivitäten interpretiert. Dabei ist es möglich, Rückschlüsse über den Einfluss beliebiger Datenbereiche zu ziehen. Dazu wird die Störung einzelner Datenpunkten mit Änderungen in den Koeffizienten der B-Splines verknüpft.

1.2 Aufbau der Arbeit

In [Kapitel 2](#) werden zunächst wichtige Grundlagen der nichtlinearen Optimierung dargestellt. Darüber hinaus werden Resultate der parametrischen Sensitivitätsanalyse zusammengefasst und das numerische Lösen von nichtlinearen Optimierungsproblemen mithilfe des SQP-Verfahrens erläutert. Dabei wird insbesondere die Softwarebibliothek WORHP vorgestellt.

[Kapitel 3](#) der Arbeit befasst sich mit stückweisen Polynomen. Der Fokus des Kapitels liegt auf der Darstellung der Eigenschaften von B-Splines. B-Splines sind spezielle Basisfunktionen des Vektorraums der stückweisen Polynome. Sie ermöglichen eine effiziente und stabile Auswertung von stückweisen Polynomen und eignen sich deshalb besonders für numerische Berechnungen.

In [Kapitel 4](#) werden Grundlagen der Parameteridentifikation dargestellt. Der Fokus liegt dabei auf dynamischen Parameteridentifikationsproblemen. Es werden verschiedene Lösungsansätze diskutiert. Insbesondere wird die Kollokationsmethode zum Lösen dynamischer Parameteridentifikationsprobleme vorgestellt.

Auf der Grundlage von [Kapitel 2](#) bis [4](#) wird in [Kapitel 5](#) eine Methode zum Analysieren des Einflusses von Daten auf die Modellparameter dynamischer Modelle entwickelt. Es werden Erweiterungen der Methode sowie ihre Grenzen diskutiert.

Im folgenden [Kapitel 6](#) wird die in [Kapitel 5](#) vorgestellte Methode anhand eines einfachen Beispiels demonstriert. Anschließend wird die Methode auf ein komplexeres Beispiel angewendet sowie die numerischen Ergebnisse vorgestellt und interpretiert.

Abschließend werden die Ergebnisse der Arbeit in [Kapitel 7](#) zusammengefasst. Darüber hinaus wird ein Ausblick über mögliche weiterführende Fragestellungen sowie Erweiterungen der vorgestellten Methode zum Analysieren des Einflusses von Daten auf die Modellparameter dynamischer Modelle gegeben.

Kapitel 2

Grundlagen der nichtlinearen Optimierung

Dieses Kapitel dient als Einleitung in das Thema der nichtlinearen Optimierung. Es werden wichtige Resultate wie die Optimalitätsbedingungen erster und zweiter Ordnung dargestellt. Darüber hinaus werden Resultate der parametrischen Sensitivitätsanalyse und SQP-Verfahren als numerische Methode zum Lösen nichtlinearer Optimierungsprobleme vorgestellt.

2.1 Grundbegriffe der nichtlinearen Optimierung

Die Überlegungen dieses Kapitels werden an folgendem Standardproblem der nichtlinearen Optimierung durchgeführt.

Definition 2.1 (Standardproblem der nichtlinearen Optimierung [24, S. 44]). *Seien $n_z, n_g, n_h \in \mathbb{N}$, $z \in \mathbb{R}^{n_z}$ und $F : \mathbb{R}^{n_z} \rightarrow \mathbb{R}$, $g : \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_g}$, $h : \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_h}$ stetig differenzierbare Funktionen. Das Problem*

$$\begin{aligned} \min_{z \in \mathbb{R}^{n_z}} \quad & F(z) \\ \text{unter} \quad & g_i(z) \leq 0, \quad i \in \mathcal{I} := \{1, \dots, n_g\}, \\ & h_i(z) = 0, \quad i \in \mathcal{E} := \{1, \dots, n_h\} \end{aligned} \tag{NLP}$$

heißt Standardproblem der nichtlinearen Optimierung. Die Funktion F heißt Zielfunktion. Die vektorwertigen Funktionen g und h werden als Ungleichungs- bzw. Gleichungsnebenbedingungen bezeichnet.

Alle weiteren Untersuchungen beziehen sich auf das Problem (NLP). Die Bezeichnungen n_z, n_g, n_h sowie F, g, h und \mathcal{I} und \mathcal{E} werden deshalb im Folgenden als gegeben

vorausgesetzt. Sofern nicht anders angegeben, seien die Funktionen F , g und h stetig differenzierbar.

Die Aufgabe der nichtlinearen Optimierung besteht darin, die Optimierungsvariable $z \in \mathbb{R}^{n_z}$ zu ermitteln, welche die Nebenbedingungen g und h erfüllt und die Zielfunktion F minimiert. Um dies zu konkretisieren, werden die Begriffe der zulässigen Menge und des Minimums definiert.

Definition 2.2 (Zulässige Menge [24, S. 1]). *Die Menge*

$$\Omega := \{z \in \mathbb{R}^{n_z} \mid g_i(z) \leq 0 \text{ für } i \in \mathcal{I} \text{ und } h_i(z) = 0 \text{ für } i \in \mathcal{E}\}$$

heißt zulässige Menge von (NLP). Ein Element $z \in \Omega$ wird als zulässiger Punkt bezeichnet.

Definition 2.3 (Minimum [24, S. 1]). *Sei $z^* \in \mathbb{R}^{n_z}$ ein zulässiger Punkt von (NLP). Dann ist z^**

i. lokales Minimum, wenn eine Umgebung $\mathcal{N} \subset \mathbb{R}^{n_z}$ von z^* existiert, mit

$$F(z^*) \leq F(z) \quad \text{für alle } z \in \mathcal{N} \cap \Omega,$$

ii. striktes lokales Minimum, wenn eine Umgebung $\mathcal{N} \subset \mathbb{R}^{n_z}$ von z^* existiert, mit

$$F(z^*) < F(z) \quad \text{für alle } z \in \mathcal{N} \cap \Omega,$$

iii. globales Minimum, wenn

$$F(z^*) \leq F(z) \quad \text{für alle } z \in \Omega,$$

iv. striktes globales Minimum, wenn

$$F(z^*) < F(z) \quad \text{für alle } z \in \Omega.$$

Für die weiteren Betrachtungen ist außerdem der Begriff der Indexmenge der aktiven Ungleichungsnebenbedingungen von Bedeutung.

Definition 2.4 (Indexmenge der aktiven Ungleichungsnebenbedingungen [24, S. 44]). *Sei $z \in \mathbb{R}^{n_z}$ ein zulässiger Punkt von (NLP) und $i \in \mathcal{I}$. Die Ungleichungsnebenbedingung g_i wird als aktiv bezeichnet, wenn $g_i(z) = 0$ ist und als inaktiv, wenn $g_i(z) < 0$ ist. In diesem Sinne heißt*

$$\mathcal{A}(z) := \{i \in \mathcal{I} \mid g_i(z) = 0\}$$

die Indexmenge der aktiven Ungleichungsnebenbedingungen.

2.2 Optimalitätsbedingungen

In diesem Abschnitt werden Bedingungen zur Charakterisierung von Minima des Optimierungsproblems (NLP) dargestellt. Ein zentraler Bestandteil ist dabei die sogenannte *Lagrange-Funktion*.

Definition 2.5 (Lagrange-Funktion [24, S. 46]). Seien $\lambda \in \mathbb{R}^{n_g}$ und $\mu \in \mathbb{R}^{n_h}$. Die durch

$$L(z, \lambda, \mu) := F(z) + \lambda^\top g(z) + \mu^\top h(z) = F(z) + \sum_{i=1}^{n_g} \lambda_i g_i(z) + \sum_{j=1}^{n_h} \mu_j h_j(z)$$

definierte Funktion $L : \mathbb{R}^{n_z} \times \mathbb{R}^{n_g} \times \mathbb{R}^{n_h} \rightarrow \mathbb{R}$ heißt Lagrange-Funktion von (NLP). Die Komponenten der Vektoren λ und μ heißen Lagrange-Multiplikatoren. Im Folgenden werden auch die Vektoren λ und μ selbst als Lagrange-Multiplikatoren bezeichnet.

Eine Möglichkeit Minima von (NLP) zu charakterisieren ist über die *Karush-Kuhn-Tucker-Bedingungen* (kurz: *KKT-Bedingungen*).

Definition 2.6 (KKT-Bedingungen, KKT-Punkt [24, S. 46]). Seien $z \in \mathbb{R}^{n_z}$, $\lambda \in \mathbb{R}^{n_g}$ und $\mu \in \mathbb{R}^{n_h}$. Die Bedingungen

$$\begin{aligned} \nabla_z L(z, \lambda, \mu) &= 0, \\ g_i(z) &\leq 0 \quad \text{für } i \in \mathcal{I}, \\ h_i(z) &= 0 \quad \text{für } i \in \mathcal{E}, \\ \lambda_i &\geq 0 \quad \text{für } i \in \mathcal{I}, \\ \lambda_i g_i(z) &= 0 \quad \text{für } i \in \mathcal{I} \end{aligned}$$

heißen Karush-Kuhn-Tucker-Bedingungen von (NLP). Dabei ist

$$\nabla_z L(z, \lambda, \mu) = \nabla F(z) + \sum_{i=1}^{n_g} \lambda \nabla g_i(z) + \sum_{j=1}^{n_h} \mu \nabla h_j(z)$$

der Gradient der Lagrange-Funktion bezüglich der Optimierungsvariable z . Ein Tripel $(z^*, \lambda^*, \mu^*) \in \mathbb{R}^{n_z} \times \mathbb{R}^{n_g} \times \mathbb{R}^{n_h}$, welches die KKT-Bedingungen erfüllt, wird KKT-Punkt von (NLP) genannt.

Wie sich herausstellt, sind die KKT-Bedingungen nicht stark genug, um als notwendige Bedingung für ein Minimum zu dienen. Minima müssen zusätzlich gewisse Regularitätseigenschaften aufweisen. In dieser Arbeit wird die Regularitätsbedingung der linearen Unabhängigkeit genutzt.

Definition 2.7 (Regularitätsbedingung der linearen Unabhängigkeit [24, S. 52]). Sei $z \in \mathbb{R}^{n_z}$ ein zulässiger Punkt von (NLP) und $\mathcal{A}(z)$ die zugehörige Indermenge der aktiven Ungleichungsnebenbedingungen. Dann erfüllt z die Regularitätsbedingung der linearen Unabhängigkeit (engl.: linear independence constraint qualification, LICQ), wenn die Gradienten

$$\nabla g_i(z) \quad \text{für } i \in \mathcal{A}(z) \quad \text{und} \quad \nabla h_j(z) \quad \text{für } j \in \mathcal{E}$$

linear unabhängig sind.

Es sei darauf hingewiesen, dass es verschiedene Ansätze gibt die Regularität von zulässigen Punkten zu charakterisieren. Einige dieser Ansätze bedingen einander. So folgt beispielsweise aus der Regularitätsbedingung der linearen Unabhängigkeit die Regularitätsbedingung von Mangasarian-Fromovitz [24, S. 50], aus welcher wiederum die Regularitätsbedingung von Abadie [24, S. 45] abgeleitet werden kann.

All diese Bedingungen haben eines gemeinsam: Sie werden benötigt, um mithilfe der KKT-Bedingungen notwendige Optimalitätsbedingungen für das nichtlineare Optimierungsproblem (NLP) zu formulieren. Wird die LICQ vorausgesetzt, so kann folgender Satz formuliert werden.

Satz 2.8 (Notwendige Optimalitätsbedingung erster Ordnung [24, S. 53]). Sei $z^* \in \mathbb{R}^{n_z}$ ein lokales Minimum von (NLP), welches die LICQ erfüllt. Dann existieren eindeutig bestimmte Lagrange-Multiplikatoren $\lambda^* \in \mathbb{R}^{n_g}$ und $\mu^* \in \mathbb{R}^{n_h}$, sodass das Tripel (z^*, λ^*, μ^*) ein KKT-Punkt von (NLP) ist.

Beweis. Ein Beweis ist in [24, S. 53] angegeben. □

Die Eindeutigkeit der Lagrange-Multiplikatoren ist dabei durch die starke Annahme der LICQ bedingt. Satz 2.8 liefert eine notwendige Bedingung für die Existenz eines Minimums: Erfüllt ein zulässiger Punkt z^* , welcher der LICQ genügt, die KKT-Bedingungen nicht, so kann es sich bei z^* nicht um ein lokales Minimum handeln.

Die notwendigen Optimalitätsbedingungen erster Ordnung basieren auf dem Gradienten der Lagrange-Funktion. Um hinreichende Optimalitätsbedingungen angeben zu können, müssen Informationen über die zweiten Ableitungen der Funktionen F, g und h , bzw. der Hesse-Matrix der Lagrange-Funktion berücksichtigt werden. Dazu wird zunächst der sogenannte kritische Kegel definiert.

Definition 2.9 (Kritischer Kegel). Sei $(z^*, \lambda^*, \mu^*) \in \mathbb{R}^{n_z} \times \mathbb{R}^{n_g} \times \mathbb{R}^{n_h}$ ein KKT-Punkt von (NLP). Die Menge

$$\begin{aligned} \mathcal{C}(z^*, \lambda^*) := \{d \in \mathbb{R}^{n_z} \mid & \nabla g_i(z^*)^\top d = 0 \text{ für } i \in \mathcal{A}(z^*) \text{ falls } \lambda_i^* > 0, \\ & \nabla g_i(z^*)^\top d \leq 0 \text{ für } i \in \mathcal{A}(z^*) \text{ falls } \lambda_i^* = 0, \\ & \nabla h_j(z^*)^\top d = 0 \text{ für } j \in \mathcal{E}\} \end{aligned}$$

heißt kritischer Kegel.

Mithilfe von [Definition 2.9](#) kann folgende notwendige Optimalitätsbedingung zweiter Ordnung formuliert werden.

Satz 2.10 (Notwendige Optimalitätsbedingung zweiter Ordnung [[24](#), S. 65]). *Sei $z^* \in \mathbb{R}^{n_z}$ ein lokales Minimum von (NLP), welches die LICQ erfüllt und seien $\lambda^* \in \mathbb{R}^{n_g}$ und $\mu^* \in \mathbb{R}^{n_h}$ die nach [Satz 2.8](#) eindeutig bestimmten zugehörigen Lagrange-Multiplikatoren. Die Funktionen F , g und h seien zweimal stetig differenzierbar. Dann gilt*

$$d^\top \nabla_{zz}^2 L(z^*, \lambda^*, \mu^*) d \geq 0 \quad \text{für alle } d \in \mathcal{C}(z^*, \lambda^*).$$

Beweis. Ein Beweis ist in [[24](#), S. 65] angegeben. □

Auch [Satz 2.10](#) liefert eine notwendige Bedingung für die Existenz eines Minimums: Ein zulässiger Punkt z^* , welcher der LICQ genügt und in dem die Hesse-Matrix der Lagrange-Funktion nicht positiv semidefinit auf dem kritischen Kegel ist, kann kein lokales Minimum sein.

Ein stärkeres Resultat kann gezeigt werden, wenn die Hesse-Matrix der Lagrange-Funktion positiv definit auf dem kritische Kegel ist. Diese Bedingung ist hinreichend dafür, dass ein KKT-Punkt ein striktes lokales Minimum ist.

Satz 2.11 (Hinreichende Optimalitätsbedingung zweiter Ordnung [[24](#), S. 67]). *Die Funktionen F , g und h seien zweimal stetig differenzierbar. Weiter sei $(z^*, \lambda^*, \mu^*) \in \mathbb{R}^{n_z} \times \mathbb{R}^{n_g} \times \mathbb{R}^{n_h}$ ein KKT-Punkt von (NLP) mit*

$$d^\top \nabla_{zz}^2 L(z^*, \lambda^*, \mu^*) d > 0 \quad \text{für alle } d \in \mathcal{C}(z^*, \lambda^*) \setminus \{0\}. \quad (2.2)$$

Dann ist z^ ein striktes lokales Minimum von (NLP).*

Beweis. Ein Beweis ist in [[24](#), S. 67] angegeben. □

2.3 Parametrische Sensitivitätsanalyse

Oftmals sind die Zielfunktion F und die Nebenbedingungen g und h von Parametern abhängig. Mithilfe der parametrischen Sensitivitätsanalyse kann die Abhängigkeit einer optimalen Lösung von Änderungen in den Parametern quantifiziert werden. Die Überlegungen werden an folgendem Standardproblem der parametrischen nichtlinearen Optimierung durchgeführt.

Definition 2.12 (Standardproblem der parametrischen nichtlinearen Optimierung). Seien $n_z, n_g, n_h, n_q \in \mathbb{N}$, $q \in Q \subset \mathbb{R}^{n_q}$ und $F : \mathbb{R}^{n_z} \times \mathbb{R}^{n_q} \rightarrow \mathbb{R}$, $g : \mathbb{R}^{n_z} \times \mathbb{R}^{n_q} \rightarrow \mathbb{R}^{n_g}$ und $h : \mathbb{R}^{n_z} \times \mathbb{R}^{n_q} \rightarrow \mathbb{R}^{n_h}$ stetig differenzierbare Funktionen. Das Problem

$$\begin{aligned} \min_{z \in \mathbb{R}^{n_z}} \quad & F(z, q) \\ \text{unter} \quad & g_i(z, q) \leq 0, \quad i \in \mathcal{I}, \\ & h_i(z, q) = 0, \quad i \in \mathcal{E} \end{aligned} \tag{NLP}(q)$$

heißt Standardproblem der parametrischen nichtlinearen Optimierung. Der Parameter q wird Störparameter genannt.

Die Menge

$$\Omega(q) := \{z \in \mathbb{R}^{n_z} \mid g_i(z, q) \leq 0 \text{ für } i \in \mathcal{I} \text{ und } h_i(z, q) = 0 \text{ für } i \in \mathcal{E}\}$$

heißt zulässige Menge von (NLP(q)). Ein Element $z \in \Omega(q)$ wird als zulässiger Punkt bezeichnet.

Des Weiteren bezeichne

$$\mathcal{A}(z, q) := \{i \in \mathcal{I} \mid g_i(z, q) = 0\}$$

die Indexmenge der aktiven Ungleichungsnebenbedingungen von (NLP(q)).

Alle weiteren Definitionen, die für das Problem (NLP) eingeführt wurden, werden konsistent auf das Problem (NLP(q)) übertragen.

Für einen fest gewählten sogenannten *Nominalparameter* $q_0 \in Q$ ergibt sich aus (NLP(q)) das sogenannte *ungestörte Problem* (NLP(q_0)). Für dieses können die aus [Abschnitt 2.2](#) bekannten Optimalitätsbedingungen genutzt werden, um ein lokales Minimum zu berechnen. Die Aufgabe der parametrischen Sensitivitätsanalyse besteht darin, zu charakterisieren, wie sich die optimale Lösung verhält, wenn es zu Störungen im Nominalparameter kommt. Durch den folgenden Satz können Rückschlüsse auf die Existenz einer optimalen Lösung und deren Eigenschaften in einer Umgebung des Nominalparameters gezogen werden.

Satz 2.13 (Sensitivitätssatz [[17](#), S. 72]). Sei $q_0 \in Q \subset \mathbb{R}^{n_q}$ der Nominalparameter und $(z^*, \lambda^*, \mu^*) \in \mathbb{R}^{n_z} \times \mathbb{R}^{n_g} \times \mathbb{R}^{n_h}$ ein KKT-Punkt des ungestörten Problems (NLP(q_0)). Die Funktionen F , g und h seien zweimal stetig differenzierbar bzgl. z . Die Gradienten $\nabla_z F$, $\nabla_z g$ und $\nabla_z h$ sowie die Funktionen g und h seien einmal stetig differenzierbar bzgl. q in einer Umgebung von z^* . Der Punkt z^* erfülle die LICQ und es gelte $\lambda_i^* > 0$ für $i \in \mathcal{A}(z^*, q_0)$. Des Weiteren erfülle (z^*, λ^*, μ^*) die hinreichenden Optimalitätsbedingungen zweiter Ordnung ([2.2](#)).

Dann existiert eine Umgebung $Q_0 \subset Q$ von q_0 und eindeutige stetig differenzierbare Funktionen $z : Q_0 \rightarrow \mathbb{R}^{n_z}$, $\lambda : Q_0 \rightarrow \mathbb{R}^{n_g}$ und $\mu : Q_0 \rightarrow \mathbb{R}^{n_h}$ mit folgenden

Eigenschaften:

- i. $z(q_0) = z^*$, $\lambda(q_0) = \lambda^*$ und $\mu(q_0) = \mu^*$,
- ii. die Indexmenge der aktiven Ungleichungsnebenbedingungen bleibt unverändert, d. h.

$$\mathcal{A}(z(q), q) = \mathcal{A}(z^*, q_0) \quad \text{für alle } q \in Q_0,$$

- iii. es gilt weiterhin die LICQ, d. h. die Gradienten

$$\nabla_z g_i(z, q) \quad \text{für } i \in \mathcal{A}(z, q) \quad \text{und} \quad \nabla_z h_j(z, q) \quad \text{für } j \in \mathcal{E}$$

sind linear unabhängig für alle $q \in Q_0$,

- iv. für alle $q \in Q_0$ ist das Tripel $(z(q), \lambda(q), \mu(q))$ ein KKT-Punkt von $(\text{NLP}(q))$, welcher die hinreichenden Optimalitätsbedingungen zweiter Ordnung (2.2) erfüllt. Insbesondere ist $z(q)$ ein striktes lokales Minimum des Problems $(\text{NLP}(q))$ mit Lagrange-Multiplikatoren $\lambda(q)$ und $\mu(q)$.

Beweis. Ein Beweis ist in [17, S. 73] angegeben. □

Einige wichtige Schritte des Beweises sollen an dieser Stelle nachvollzogen werden, da sie zu einer expliziten Darstellung der Sensitivitätsdifferenziale der optimale Lösung und der entsprechenden Lagrange-Multiplikatoren führen. Die KKT-Bedingungen für $(\text{NLP}(q_0))$ können als Gleichungssystem

$$K(z, \lambda, \mu, q) := \begin{pmatrix} \nabla_z L(z, \lambda, \mu, q) \\ \text{diag}(\lambda_1, \dots, \lambda_{n_g}) g(z, q) \\ h(z, q) \end{pmatrix} = \vec{0} \quad (2.3)$$

formuliert werden. Aus den Voraussetzungen von Satz 2.13 folgt die Differenzierbarkeit von K bezüglich (z, λ, μ) in einer Umgebung von (z^*, λ^*, μ^*) und q_0 . Partielles Ableiten von K führt auf die sogenannte *Kuhn-Tucker-Matrix*

$$\nabla_{(z, \lambda, \mu)} K(z^*, \lambda^*, \mu^*, q_0) = \begin{pmatrix} \nabla_{zz}^2 L(z^*, \lambda^*, \mu^*, q_0) & \nabla_z g(z^*, q_0)^\top & \nabla_z h(z^*, q_0)^\top \\ \Lambda \nabla_z g(z^*, q_0) & \Gamma & 0 \\ \nabla_z h(z^*, q_0) & 0 & 0 \end{pmatrix}$$

mit $\Gamma = \text{diag}(g_1(z^*, q_0), \dots, g_{n_g}(z^*, q_0))$ und $\Lambda = \text{diag}(\lambda_1^*, \dots, \lambda_{n_g}^*)$. Es kann gezeigt werden, dass die Kuhn-Tucker-Matrix unter den hinreichenden Optimalitätsbedingungen zweiter Ordnung invertierbar ist. Folglich kann der Satz über implizite Funktionen [20, S. 93] angewendet werden, um die Existenz von differenzierbaren

Funktionen $z(q)$, $\lambda(q)$ und $\mu(q)$ zu zeigen, welche das Gleichungssystem (2.3) in einer Umgebung von q_0 erfüllen. Differenzieren der Gleichung $K(z(q), \lambda(q), \mu(q), q) = \vec{0}$ nach q und Auswerten der Ableitung im Nominalparameter q_0 führt schließlich auf ein lineares Gleichungssystem für die Sensitivitätsdifferenziale der optimalen Lösung sowie der Lagrange-Multiplikatoren. Der besseren Übersicht halber wird die Notation $K[q] := K(z(q), \lambda(q), \mu(q), q)$ verwendet. Es gilt

$$\begin{aligned}\vec{0} &= \frac{dK}{dq}[q_0] = \frac{\partial K}{\partial z}[q_0] \frac{dz}{dq}(q_0) + \frac{\partial K}{\partial \lambda}[q_0] \frac{d\lambda}{dq}(q_0) + \frac{\partial K}{\partial \mu}[q_0] \frac{d\mu}{dq}(q_0) + \frac{\partial K}{\partial q}[q_0] \\ &= \nabla_{(z,\lambda,\mu)} K[q_0] \cdot \begin{pmatrix} \frac{dz}{dq}(q_0) \\ \frac{d\lambda}{dq}(q_0) \\ \frac{d\mu}{dq}(q_0) \end{pmatrix} + \frac{\partial K}{\partial q}[q_0].\end{aligned}$$

Aus dem Sensitivitätssatz folgt also direkt das folgende Korollar.

Korollar 2.14 (Sensitivitäten erster Ordnung der optimalen Lösung). *Es seien die Voraussetzungen des Sensitivitätssatzes (Satz 2.13) erfüllt. Dann gilt*

$$\begin{pmatrix} \frac{dz}{dq}(q_0) \\ \frac{d\lambda}{dq}(q_0) \\ \frac{d\mu}{dq}(q_0) \end{pmatrix} = -M^{-1} \begin{pmatrix} \nabla_{zq}^2 L(z^*, \lambda^*, \mu^*, q_0) \\ \Lambda \nabla_{qg}(z^*, q_0) \\ \nabla_q h(z^*, q_0) \end{pmatrix} \quad (2.4)$$

mit

$$M = \begin{pmatrix} \nabla_{zz}^2 L(z^*, \lambda^*, \mu^*, q_0) & \nabla_{zq} g(z^*, q_0)^\top & \nabla_z h(z^*, q_0)^\top \\ \Lambda \nabla_{zq} g(z^*, q_0) & \Gamma & 0 \\ \nabla_z h(z^*, q_0) & 0 & 0 \end{pmatrix}$$

und

$$\Gamma = \text{diag}(g_1(z^*, q_0), \dots, g_{n_g}(z^*, q_0)), \quad \Lambda = \text{diag}(\lambda_1^*, \dots, \lambda_{n_g}^*).$$

Für diese Arbeit sind lediglich die Sensitivitäten erster Ordnung der optimalen Lösung von Interesse. Es sei jedoch darauf hingewiesen, dass auch Formeln für die Sensitivitäten weiterer Problemgrößen wie der Zielfunktion und den Nebenbedingungen hergeleitet werden können. Für weitere Informationen sei auf Fiacco [17] verwiesen.

Die Sensitivitäten (2.4) können genutzt werden, um mittels Taylor-Entwicklungen erster Ordnung die Lösung des gestörten Problems ($\text{NLP}(q)$) in einer Umgebung des Nominalparameters q_0 zu approximieren. Da zur Auswertung der Taylor-Entwicklung lediglich Matrixmultiplikationen berechnet werden müssen, kann dies extrem effizient und schnell geschehen. In diesem Sinne kann die Lösung des gestörten Problems in Echtzeit approximiert werden [12]. Büskens [11] stellt außerdem ein iteratives Korrekturverfahren vor, welches mithilfe von Sensitivitäten die Approximation der Lösung ohne großen Mehraufwand weiter verbessert.

2.4 Numerische Lösung nichtlinearer Optimierungsprobleme mittels SQP-Verfahren

Theoretisch können nichtlineare Optimierungsprobleme der Form (NLP) durch analytisches Bestimmen von KKT-Punkten und Überprüfen der hinreichenden Optimalitätsbedingungen gelöst werden. Oftmals ist dieser Ansatz jedoch nicht praktikabel. Die KKT-Bedingungen führen auf ein nichtlineares Gleichungssystem, welches im Allgemeinen nicht analytisch lösbar ist. Darüber hinaus ist dieser Ansatz für hochdimensionale Probleme ungeeignet, da der Aufwand des analytischen Lösens der KKT-Bedingungen enorm wäre. Deshalb müssen in der Regel numerische Methoden herangezogen werden, um Lösungen von (NLP) zu berechnen bzw. zu approximieren.

In diesem Abschnitt werden SQP-Verfahren (aus dem Englischen: *Sequential Quadratic Programming*) zum Lösen von (NLP) vorgestellt. SQP-Verfahren sind eine weit verbreitete Methode zum Lösen nichtlinearer Optimierungsprobleme. Beispielsweise seien die Softwarepakete NLPQL [62] und SNOPT [26] als reine Implementierungen von SQP-Verfahren genannt. Darüber hinaus implementieren viele Softwarebibliotheken zum Lösen nichtlinearer Optimierungsprobleme SQP-Verfahren. Beispiele hierfür sind etwa Knitro [14] oder die in Abschnitt 2.4.2 vorgestellte Bibliothek WORHP [13].

2.4.1 SQP-Verfahren

Im Folgenden wird das grundlegende Vorgehen von SQP-Verfahren vorgestellt. SQP-Verfahren verfolgen den Ansatz, das nichtlineare Optimierungsproblem (NLP) lokal durch quadratische Teilprobleme zu approximieren. Mithilfe der Lösungen dieser quadratischen Teilprobleme kann iterativ eine Lösung des übergeordneten Problems bestimmt werden. Für die weiteren Betrachtungen werden zunächst die Ungleichungsnebenbedingungen von (NLP) vernachlässigt und das Problem

$$\begin{aligned} \min_{z \in \mathbb{R}^{n_z}} \quad & F(z) \\ \text{unter} \quad & h_i(z) = 0, \quad i \in \mathcal{E} \end{aligned} \tag{2.5}$$

untersucht. Sei (z^*, μ^*) ein KKT-Punkt von (2.5). Da keine Ungleichungsnebenbedingungen betrachtet werden, besteht der KKT-Punkt lediglich aus der Optimierungsvariable z^* und dem Lagrange-Multiplikator μ^* . Da ein KKT-Punkt per Definition die KKT-Bedingungen erfüllt, gilt

$$\Phi(z^*, \mu^*) := \begin{pmatrix} \nabla_z L(z^*, \mu^*) \\ h(z^*) \end{pmatrix} = 0. \tag{2.6}$$

Durch das Lösen des nichtlinearen Gleichungssystems (2.6) können folglich KKT-Punkte und damit mögliche Lösungen von (2.5) bestimmt werden. Numerisch können Lösungen von (2.6) iterativ mithilfe des mehrdimensionalen Newton-Verfahrens [24, S. 235] approximiert werden. Ausgehend von einem Startwert $(z^{[0]}, \mu^{[0]})^\top \in \mathbb{R}^{n_z} \times \mathbb{R}^{n_h}$ werden dabei neue Iterierte durch die Vorschrift

$$\begin{pmatrix} z^{[k+1]} \\ \mu^{[k+1]} \end{pmatrix} := \begin{pmatrix} z^{[k]} + \Delta z^{[k]} \\ \mu^{[k]} + \Delta \mu^{[k]} \end{pmatrix}$$

berechnet. Die Suchrichtung $(\Delta z^{[k]}, \Delta \mu^{[k]})^\top \in \mathbb{R}^{n_z} \times \mathbb{R}^{n_h}$ ist durch Lösen des Gleichungssystems

$$\begin{aligned} & \nabla_{(z,\mu)} \Phi(z^{[k]}, \mu^{[k]}) \begin{pmatrix} \Delta z^{[k]} \\ \Delta \mu^{[k]} \end{pmatrix} = -\Phi(z^{[k]}, \mu^{[k]}) \\ \Leftrightarrow & \begin{pmatrix} \nabla_{zz}^2 L(z^{[k]}, \mu^{[k]}) & \nabla_z h(z^{[k]})^\top \\ \nabla_z h(z^{[k]}) & 0 \end{pmatrix} \begin{pmatrix} \Delta z^{[k]} \\ \Delta \mu^{[k]} \end{pmatrix} = - \begin{pmatrix} \nabla_z L(z^{[k]}, \mu^{[k]}) \\ h(z^{[k]}) \end{pmatrix} \end{aligned} \quad (2.7)$$

zu bestimmen. Falls (z^*, μ^*) die hinreichenden Optimalitätsbedingungen zweiter Ordnung (2.2) erfüllt und z^* der LICQ genügt, dann ist die Jacobi-Matrix $\nabla_{(z,\mu)} \Phi(z^{[k]}, \mu^{[k]})$ regulär und das Newton-Verfahren konvergiert lokal superlinear gegen den KKT-Punkt (z^*, μ^*) [24, S. 240].

Sei $H^{[k]} := \nabla_{zz}^2 L(z^{[k]}, \mu^{[k]})$ die Hesse-Matrix der Lagrange-Funktion. Für den Gradienten der Lagrange-Funktion gilt

$$\nabla_z L(z^{[k]}, \mu^{[k]}) = \nabla_z F(z^{[k]}) + \nabla_z h(z^{[k]})^\top \mu^{[k]}.$$

Somit ist das Gleichungssystem (2.7) äquivalent zu

$$\begin{pmatrix} H^{[k]} & \nabla_z h(z^{[k]})^\top \\ \nabla_z h(z^{[k]}) & 0 \end{pmatrix} \begin{pmatrix} \Delta z^{[k]} \\ \Delta \mu^{[k]} + \mu^{[k]} \end{pmatrix} = - \begin{pmatrix} \nabla_z F(z^{[k]}) \\ h(z^{[k]}) \end{pmatrix}. \quad (2.8)$$

Wie sich herausstellt, lässt sich das Gleichungssystem (2.8) auch auf eine andere Art interpretieren. Dazu wird das quadratische Optimierungsproblem

$$\begin{aligned} & \min_{\Delta z \in \mathbb{R}^{n_z}} \nabla_z F(z^{[k]})^\top \Delta z + \frac{1}{2} \Delta z^\top H^{[k]} \Delta z \\ & \text{unter } h_i(z^{[k]}) + \nabla_z h_i(z^{[k]})^\top \Delta z = 0, \quad i \in \mathcal{E} \end{aligned} \quad (2.9)$$

betrachtet. Die Lagrange-Funktion von (2.9) ist

$$L(\Delta z, \tilde{\mu}) = \nabla_z F(z^{[k]})^\top \Delta z + \frac{1}{2} \Delta z^\top H^{[k]} \Delta z + \tilde{\mu}^\top (h(z^{[k]}) + \nabla_z h(z^{[k]}) \Delta z)$$

und somit erfüllt ein KKT-Punkt $(\Delta z^*, \tilde{\mu}^*) \in \mathbb{R}^{n_z} \times \mathbb{R}^{n_h}$ von Problem (2.9) die

KKT-Bedingungen

$$\begin{aligned} & \begin{pmatrix} \nabla_z F(z^{[k]}) + H^{[k]} \Delta z^* + \nabla_z h(z^{[k]})^\top \tilde{\mu}^* \\ h(z^{[k]}) + \nabla_z h(z^{[k]}) \Delta z^* \end{pmatrix} = 0 \\ \Leftrightarrow & \begin{pmatrix} H^{[k]} & \nabla_z h(z^{[k]})^\top \\ \nabla_z h(z^{[k]}) & 0 \end{pmatrix} \begin{pmatrix} \Delta z^* \\ \tilde{\mu}^* \end{pmatrix} = - \begin{pmatrix} \nabla_z F(z^{[k]}) \\ h(z^{[k]}) \end{pmatrix}. \end{aligned} \quad (2.10)$$

Wird nun $\tilde{\mu}^* = \Delta \mu^{[k]} + \mu^{[k]}$ gesetzt, so sind (2.8) und (2.10) äquivalent. Anstatt das nichtlineare Gleichungssystem (2.7) zu lösen, um eine Suchrichtung für das Newton-Verfahren zu bestimmen, kann folglich alternativ das quadratische Optimierungsproblem (2.9) gelöst werden.

Die vorangegangenen Überlegungen beziehen sich auf das Optimierungsproblem (2.5) ohne Ungleichungsnebenbedingungen. Sie lassen sich aber auf den allgemeinen Fall übertragen. Analog zu dem Optimierungsproblem (2.9) wird ein quadratisches Teilproblem zu (NLP) definiert.

Definition 2.15 (Quadratisches Teilproblem). Sei $(z^{[k]}, \lambda^{[k]}, \mu^{[k]}) \in \mathbb{R}^{n_z} \times \mathbb{R}^{n_g} \times \mathbb{R}^{n_h}$ und $H^{[k]} \in \mathbb{R}^{n_z \times n_z}$ eine geeignete Approximation der Hesse-Matrix der Lagrange-Funktion $\nabla_{zz}^2 L(z^{[k]}, \lambda^{[k]}, \mu^{[k]})$. Dann heißt das Optimierungsproblem

$$\begin{aligned} \min_{\Delta z \in \mathbb{R}^{n_z}} & \quad \nabla_z F(z^{[k]})^\top \Delta z + \frac{1}{2} \Delta z^\top H^{[k]} \Delta z \\ \text{unter} & \quad g_i(z^{[k]}) + \nabla_z g_i(z^{[k]})^\top \Delta z \leq 0, \quad i \in \mathcal{I}, \\ & \quad h_i(z^{[k]}) + \nabla_z h_i(z^{[k]})^\top \Delta z = 0, \quad i \in \mathcal{E} \end{aligned} \quad (\text{QP})$$

quadratisches Teilproblem von (NLP). Eine Lösung Δz^* von (QP) heißt Suchrichtung der Optimierungsvariable z .

In Algorithmus 1 ist das Vorgehen von SQP-Verfahren zusammengefasst. Durch das Lösen des quadratischen Teilproblems (QP) wird eine Suchrichtung ermittelt und iterativ eine Lösung von (NLP) bestimmt. Da in jeder Iteration das quadratische Teilproblem gelöst werden muss, spricht man von Sequential Quadratic Programming. Zum Lösen von quadratischen Optimierungsproblemen stehen eine Vielzahl von effizienten Algorithmen zur Verfügung. Für eine ausführliche Zusammenfassung verschiedener Algorithmen sei auf Geiger und Kanzow [24, S. 197 ff.] sowie Nocedal und Wright [47, S. 448 ff.] verwiesen.

Die Berechnung der Hesse-Matrix der Lagrange-Funktion ist oftmals aufwändig und muss mithilfe numerischer Methoden wie finiten Differenzen geschehen. Darüber hinaus ist die Hesse-Matrix im Allgemeinen nicht überall positiv definit [24]. Dies erschwert das Lösen der quadratischen Teilprobleme und im Allgemeinen kann keine eindeutige optimale Lösung, sondern lediglich ein KKT-Punkt von (QP) bestimmt werden. In Definition 2.15 ist deshalb $H^{[k]}$ nicht notwendigerweise die Hesse-

Algorithmus 1 : SQP-Verfahren**Eingabe :**

Nichtlineares Optimierungsproblem der Form (NLP)

$(z^{[0]}, \lambda^{[0]}, \mu^{[0]}) \in \mathbb{R}^{n_z} \times \mathbb{R}^{n_g} \times \mathbb{R}^{n_h}$: Startschätzungen für die
Optimierungsvariablen und Lagrange-Multiplikatoren

$H^{[0]} \in \mathbb{R}^{n_z \times n_z}$: Approximation der Hesse-Matrix der Lagrange-Funktion

Ausgabe : KKT-Punkt $(z^*, \lambda^*, \mu^*) \in \mathbb{R}^{n_z} \times \mathbb{R}^{n_g} \times \mathbb{R}^{n_h}$ von (NLP)

Initialisierung : Setze $k = 0$

while $(z^{[k]}, \lambda^{[k]}, \mu^{[k]})$ ist kein KKT-Punkt von (NLP) :

Bestimme Lösung $\Delta z \in \mathbb{R}^{n_z}$ des quadratischen Teilproblems (QP) mit
zugehörigen Lagrange-Multiplikatoren $\lambda^{[k+1]} \in \mathbb{R}^{n_g}$ und $\mu^{[k+1]} \in \mathbb{R}^{n_h}$.

Berechne $z^{[k+1]} := z^{[k]} + \Delta z$.

Setze $k \leftarrow k + 1$.

end

return $(z^{[k]}, \lambda^{[k]}, \mu^{[k]})$

Matrix der Lagrange-Funktion, sondern nur eine geeignete Approximation ebendieser. Ein möglicher Ansatz ist es, die Hesse-Matrix mittels BFGS-Verfahren in jeder Iteration zu approximieren. Unter gewissen Voraussetzungen kann gezeigt werden, dass dieses Vorgehen zu symmetrischen und positiv definiten Approximationen $H^{[k]}$ führt [24, S. 257]. Darüber hinaus zeigt Powell [51], dass das SQP-Verfahren bei diesem Ansatz lokal superlineare Konvergenz aufweist.

Es wird trotzdem noch einmal speziell auf den Fall eingegangen, dass $H^{[k]} = \nabla_{zz}^2 L(z^{[k]}, \mu^{[k]})$ die Hesse-Matrix der Lagrange-Funktion ist. Auch in diesem Fall kann gezeigt werden, dass das SQP-Verfahren lokal superlinear konvergiert [24, S. 245]. Darüber hinaus kann das SQP-Verfahren in diesem Fall zum effizienten Berechnen von Sensitivitäten genutzt werden (vgl. Abschnitt 2.3). Ein KKT-Punkt $(\Delta z^*, \tilde{\lambda}^*, \tilde{\mu}^*) \in \mathbb{R}^{n_z} \times \mathbb{R}^{n_g} \times \mathbb{R}^{n_h}$ von (QP) erfüllt analog zu (2.10) die KKT-Bedingungen

$$\begin{aligned} & \begin{pmatrix} \nabla_z F(z^{[k]}) + H^{[k]} \Delta z^* + \nabla_z g(z^{[k]})^\top \tilde{\lambda}^* + \nabla_z h(z^{[k]})^\top \tilde{\mu}^* \\ \Lambda \cdot (g(z^{[k]}) + \nabla_z g(z^{[k]}) \Delta z^*) \\ h(z^{[k]}) + \nabla_z h(z^{[k]}) \Delta z^* \end{pmatrix} = 0 \\ \Leftrightarrow & \begin{pmatrix} \nabla_{zz}^2 L(z^{[k]}, \mu^{[k]}) & \nabla_z g(z^{[k]})^\top & \nabla_z h(z^{[k]})^\top \\ \Lambda \nabla_z g(z^{[k]}) & \Gamma & 0 \\ \nabla_z h(z^{[k]}) & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta z^* \\ \tilde{\lambda}^* \\ \tilde{\mu}^* \end{pmatrix} = - \begin{pmatrix} \nabla_z F(z^{[k]}) \\ 0 \\ h(z^{[k]}) \end{pmatrix}, \end{aligned}$$

mit $\Gamma = \text{diag}(g_1(\Delta z^*), \dots, g_{n_g}(\Delta z^*))$ und $\Lambda = \text{diag}(\tilde{\lambda}_1^*, \dots, \tilde{\lambda}_{n_g}^*)$. Die Systemmatrix dieses Gleichungssystems ist die bereits aus Abschnitt 2.3 bekannte Kuhn-Tucker-Matrix. Während des Lösens des quadratischen Teilproblems wird die Kuhn-

Tucker-Matrix mitunter bereits faktorisiert. Dieser Umstand kann ausgenutzt werden, um das Gleichungssystem (2.4) zum Berechnen der Sensitivitäten sehr effizient zu lösen.

Das SQP-Verfahren in [Algorithmus 1](#) hat lediglich lokale Konvergenz. Es kann folglich nur in einer Umgebung um die optimale Lösung garantiert werden, dass das Verfahren konvergiert. Diese Umgebung ist jedoch im Allgemeinen nicht bekannt. Um ein global konvergentes Verfahren zu erhalten, kann [Algorithmus 1](#) modifiziert werden. Dazu wird eine Schrittweite $\alpha^{[k]} \in \mathbb{R}$ eingeführt und die Newton-Iteration zum Bestimmen von $z^{[k+1]}$ durch

$$z^{[k+1]} = z^{[k]} + \alpha^{[k]} \Delta z^{[k]}$$

ersetzt. Eine geeignete Schrittweite kann mithilfe sogenannter Bewertungsfunktionen bestimmt werden. Die Schrittweite sollte so gewählt werden, dass der Wert der Bewertungsfunktion abnimmt. Für eine detaillierte Darstellung des Vorgehens sei auf Han [28], Powell [50] und Schittkowski [64] verwiesen.

Praktische Implementierungen von SQP-Verfahren sollten weitere Aspekte berücksichtigen. So kann es beispielsweise vorkommen, dass die quadratischen Teilprobleme keine zulässigen Lösungen besitzen. Folglich kann auch keine Suchrichtung bestimmt werden. In diesem Fall kann etwa eine Relaxation der Nebenbedingungen des quadratischen Teilproblems Abhilfe schaffen [24, S. 263 ff.].

2.4.2 WORHP

In dieser Arbeit wird die Softwarebibliothek WORHP („We Optimize Really Huge Problems“) [13] zum Lösen nichtlinearer Optimierungsprobleme genutzt. WORHP wurde im Rahmen verschiedener Projekte an der Universität Bremen entwickelt. In WORHP ist sowohl ein SQP-Verfahren (vgl. [Abschnitt 2.4.1](#)) als auch ein Innere-Punkte-Verfahren [38] zum Lösen nichtlinearer Optimierungsprobleme implementiert. In diesem Abschnitt werden einige Techniken des in WORHP implementierten SQP-Verfahrens vorgestellt. WORHP nutzt die zu (NLP) äquivalente Problemformulierung

$$\begin{aligned} & \min_{z \in \mathbb{R}^{n_z}} F(z) \\ & \text{unter } \begin{pmatrix} l \\ L \end{pmatrix} \leq \begin{pmatrix} z \\ G(z) \end{pmatrix} \leq \begin{pmatrix} u \\ U \end{pmatrix} \end{aligned} \quad (2.11)$$

mit den Schranken $l, u \in \bar{\mathbb{R}}^{n_z}$ und $L, U \in \bar{\mathbb{R}}^{n_g+n_h}$ sowie den Nebenbedingungen $G : \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_g+n_h}$. Dabei ist $\bar{\mathbb{R}} := \mathbb{R} \cup \{\infty, -\infty\}$. In der Formulierung (2.11) sind sogenannte Box-Beschränkungen $l \leq z \leq u$ explizit als solche gekennzeichnet und können daher effizient in das in das Optimierungsverfahren integriert werden. Sind

Optimierungsvariablen oder Nebenbedingungen nur in einer Richtung beschränkt, so kann dies durch die Wahl der Schranken $\pm\infty$ formuliert werden. Gleichungsnebenbedingungen können durch Gleichsetzten der unteren und oberen Schranke definiert werden.

WORHP zeichnet sich dadurch aus, auch hochdimensionale Optimierungsprobleme mit mehreren Millionen Optimierungsvariablen und Nebenbedingungen lösen zu können. Dies gelingt insbesondere durch das Ausnutzen von dünnbesetzten Strukturen in den Jacobi-Matrizen der Nebenbedingungen und der Hesse-Matrix der Lagrange-Funktion. Für die numerischen Berechnungen ist es ausreichend, die Einträge der Matrizen, die ungleich Null sind, zu speichern. Dies wird in WORHP mittels effizienter Speicherverwaltung umgesetzt [83].

Die Ableitungen der Zielfunktion und der Nebenbedingungen können in WORHP von dem Nutzer angegeben werden. Alternativ können sie mittels finiter Differenzen approximiert werden. WORHP verwendet Methoden der Graphentheorie, um mehrere Ableitungen gleichzeitig und damit effizienter zu berechnen [36]. Die Hesse-Matrix der Lagrange-Funktion kann mittels finiter Differenzen oder durch verschiedene BFGS-Verfahren approximiert werden [36, 58].

WORHP implementiert ein globalisiertes SQP-Verfahren. Zum Bestimmen der Schrittlänge stehen verschiedene Bewertungsfunktionen sowie das Filterverfahren nach Fletcher und Leyffer [18] zur Verfügung. Es kann vorkommen, dass das Bestimmen der Schrittlänge nicht möglich ist. Für diesen Fall sind in WORHP verschiedene Rettungsstrategien implementiert. So wird ein Abbruch des Algorithmus vermieden [46]. Um die Lösbarkeit der quadratischen Teilprobleme sicherzustellen, ist eine Regularisierung der Hesse-Matrix sowie eine Relaxation der Nebenbedingungen implementiert [83].

Geffken [22] sowie Geffken und Büskens [23] zeigen Wege auf, wie die Effizienz von WORHP mittels paralleler Berechnungen noch weiter gesteigert werden kann. Ein solch paralleler Berechnungsansatz würde darüber hinaus die Möglichkeit bieten, das Optimierungsproblem (2.11) für verschiedene Startschätzungen und unterschiedliche Einstellungen von WORHP gleichzeitig zu lösen. Da SQP-Verfahren im Allgemeinen ein lokales Minimum von (2.11) berechnen, können auf diese Weise Erkenntnisse über die Güte der gefundenen Lösungen gewonnen werden.

Eine weitere Funktionalität bietet das in die Bibliothek WORHP integrierte Modul WORHP Zen [39]. Dieses ermöglicht die effiziente Berechnung von parametrischen Sensitivitäten. Dabei wird die während des SQP-Verfahrens berechnete Zerlegung der Kuhn-Tucker-Matrix genutzt, um die Sensitivitäten effizient zu bestimmen (vgl. Abschnitt 2.4.1). Dabei ist zu beachten, dass die Hesse-Matrix der Lagrange-Funktion während des SQP-Verfahrens nicht durch ein BFGS-Verfahren approximiert wird, da zum Bestimmen der Sensitivitäten die tatsächliche Hesse-Matrix benötigt wird.

Kapitel 3

Stückweise Polynome und B-Splines

In diesem Kapitel werden stückweise Polynome eingeführt und klassifiziert. Anschließend werden die Nachteile der kanonischen Basis dieser Funktionen diskutiert. Geeignete Basisfunktionen bilden die sogenannten B-Splines. Ein wesentlicher Teil dieses Abschnitts besteht darin, die Eigenschaften von B-Splines herauszuarbeiten und effiziente Algorithmen zur Auswertung von B-Splines herzuleiten. Sofern nicht explizit anders angegeben, basiert dieser Abschnitt auf dem Lehrbuch von de Boor [7], insbesondere den Kapiteln VII bis X, sowie weiteren Arbeiten von de Boor [8, 9].

3.1 Stückweise Polynome

Um stückweise Polynome definieren zu können, wird zunächst der Vektorraum der Polynome eingeführt.

Definition 3.1 (Polynome). *Sei $d \in \mathbb{N}$. Dann bezeichnet*

$$\Pi_d := \left\{ f : \mathbb{R} \rightarrow \mathbb{R} \mid f(x) = \sum_{i=0}^d a_i x^i, a_i \in \mathbb{R}, i = 0, \dots, d \right\}$$

den Vektorraum der Polynome vom Grad $\leq d$.

Es sei darauf hingewiesen, dass de Boor [7, 8, 9] den Vektorraum der Polynome von der *Ordnung* d als Menge aller Polynome vom Grad $< d$ definiert und mit $\Pi_{<d}$ bezeichnet. So wären zum Beispiel Polynome erster Ordnung konstant und Polynome zweiter Ordnung linear. In dieser Arbeit wurden Polynome bewusst wie in [Definition 3.1](#) definiert, da diese Definition intuitiver erscheint und in anderen Lehrbüchern üblich ist (z. B. [60, 68]). Die Diskrepanz in der Bedeutung des Indexes d

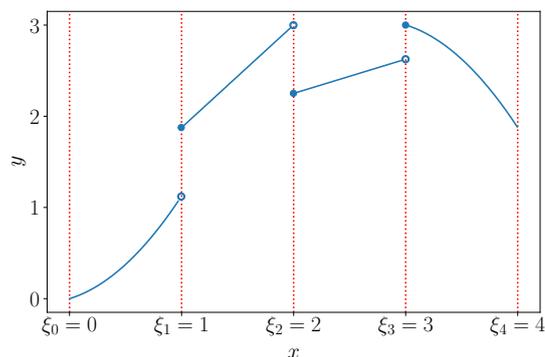


Abbildung 3.1: Stückweises Polynom vom Grad $d \leq 2$ mit den Sprungstellen $\xi = (0, 1, 2, 3, 4)$

ist zu beachten, sollten einige der folgenden Resultate in den Arbeiten von de Boor nachgeschlagen werden.

Mithilfe von Polynomen lassen sich stückweise Polynome definieren.

Definition 3.2 (Stückweise Polynome). Seien $d, k \in \mathbb{N}$ und $\xi := (\xi_i)_{i=0, \dots, k} \in \mathbb{R}^{k+1}$ eine streng monoton wachsende Folge. Dann bezeichnet

$$\Pi_{d, \xi} := \{f : [\xi_0, \xi_k) \rightarrow \mathbb{R} \mid \exists P_i \in \Pi_d : f|_{[\xi_i, \xi_{i+1})} = P_i, i = 0, \dots, k-1\}$$

den Vektorraum der stückweisen Polynome vom Grad $\leq d$ mit Sprungstellen ξ .

In [Abbildung 3.1](#) ist ein stückweises Polynom vom Grad $d \leq 2$ mit den Sprungstellen $\xi = (0, 1, 2, 3, 4)$ abgebildet. Es ist zu erkennen, dass die Funktion zwischen je zwei Sprungstellen linear oder quadratisch verläuft. An den Sprungstellen sind Unstetigkeiten festzustellen.

Stückweise Polynome sind im Allgemeinen an den Sprungstellen nicht stetig und folglich auch nicht differenzierbar. Es wird dennoch eine Ableitung für sie definiert.

Definition 3.3. Seien $d, k, j \in \mathbb{N}$ und $\xi := (\xi_i)_{i=0, \dots, k} \in \mathbb{R}^{k+1}$ eine streng monoton wachsende Folge. Weiter seien $P_i \in \Pi_d$ für $i = 0, \dots, k-1$ und $f \in \Pi_{d, \xi}$ mit

$$f(x) = P_i(x), \quad x \in [\xi_i, \xi_{i+1}) \quad \text{für } i = 0, \dots, k-1.$$

Dann ist die j -te Ableitung von f definiert durch

$$f^{(j)}(x) := P_i^{(j)}(x), \quad x \in [\xi_i, \xi_{i+1}) \quad \text{für } i = 0, \dots, k-1. \quad (3.1)$$

Für $j \leq d$ ist die j -Ableitung von $f \in \Pi_{d, \xi}$ ein stückweises Polynom vom Grad $\leq d-j$, d. h. $f^{(j)} \in \Pi_{d-j, \xi}$. Allgemein ist $f^{(j)}$ nicht stetig in den Sprungstellen. Jedoch gibt es stückweise Polynome, welche an den Sprungstellen stetig und

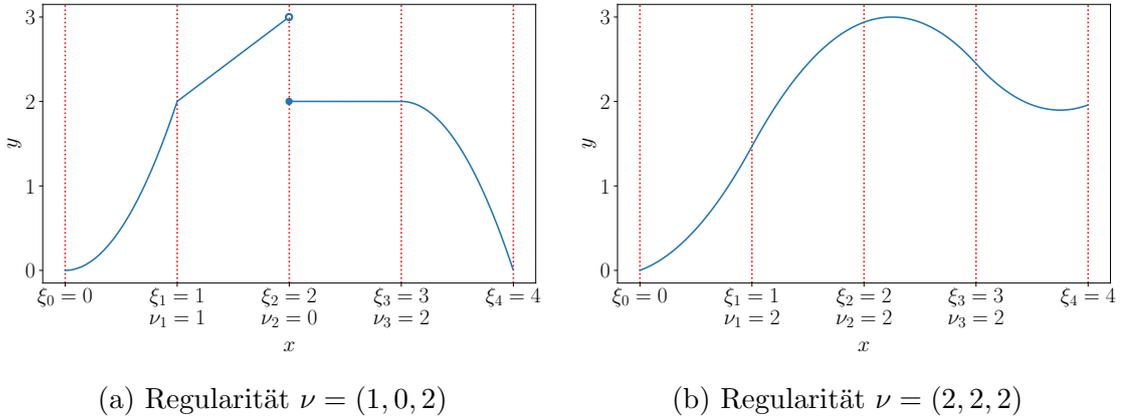


Abbildung 3.2: ν -reguläre stückweise Polynome vom Grad $d \leq 2$ mit den Sprungstellen $\xi = (0, 1, 2, 3, 4)$

sogar zu einem gewissen Grad stetig differenzierbar sind. Um stückweise Polynome besser klassifizieren zu können, werden deshalb Untervektorräume von $\Pi_{d,\xi}$ definiert, welche die Differenzierbarkeit an den Sprungstellen berücksichtigen.

Definition 3.4 (ν -reguläre stückweise Polynome).

Seien $d, k \in \mathbb{N}$, $\xi := (\xi_i)_{i=0,\dots,k} \in \mathbb{R}^{k+1}$ eine streng monoton wachsende Folge und $\nu := (\nu_i)_{i=1,\dots,k-1} \in \mathbb{N}^{k-1}$ mit $\max_{i=1,\dots,k-1} \nu_i \leq d$. Dann bezeichnet

$$\Pi_{d,\xi,\nu} := \{f \in \Pi_{d,\xi} \mid f^{(j-1)} \text{ ist stetig in } \xi_i \text{ für } j = 1, \dots, \nu_i \text{ und } i = 1, \dots, k-1\}$$

den Vektorraum der ν -regulären stückweisen Polynome vom Grad $\leq d$.

Die weiteren Untersuchungen dieses Kapitels beziehen sich auf ν -reguläre stückweise Polynome gemäß Definition 3.4. Die Bezeichnungen d , ξ , ν und k werden deshalb im Folgenden als gegeben vorausgesetzt. Insbesondere sei ξ eine streng monoton wachsende Folge und für ν gelte $\max_{i=1,\dots,k-1} \nu_i \leq d$.

Ein ν -reguläres stückweises Polynom $f \in \Pi_{d,\xi,\nu}$ ist $(\nu_i - 1)$ -Mal stetig differenzierbar an den Sprungstellen ξ_i . Ist $\nu_i = 0$, so wird keine Bedingung an die Stetigkeit von f in ξ_i gestellt. In [Abbildung 3.2](#) sind ν -reguläre stückweise Polynome vom Grad $d \leq 2$ mit den Sprungstellen $\xi = (0, 1, 2, 3, 4)$ abgebildet. Das ν -reguläre stückweise Polynom in [Abbildung 3.2a](#) zeichnet sich an den inneren Sprungstellen durch die Regularität $\nu = (1, 0, 2)$ aus. Demnach ist die Funktion in ξ_1 stetig und in ξ_3 stetig differenzierbar. Der Wert ν_2 ist Null und folglich wird in ξ_2 keine Bedingung an die Stetigkeit der Funktion gestellt. In [Abbildung 3.2b](#) ist ein ν -reguläres stückweises Polynom mit $\nu = (2, 2, 2)$ abgebildet. An allen Sprungstellen ist die Funktion stetig differenzierbar. Bei dieser Funktion handelt es sich um einen Spline im klassischen Sinne.

3.2 Basis der abgeschnittenen Potenzfunktionen

Eine zweckmäßige Basis für den Vektorraum der ν -regulären stückweisen Polynome zu kennen, ist in verschiedener Hinsicht interessant. Sie würde eine besonders einfache Darstellung von ν -regulären stückweisen Polynomen erlauben: Jede beliebige Funktion $f \in \Pi_{d,\xi,\nu}$ könnte als Linearkombination der Basisfunktionen dargestellt werden. In diesem Abschnitt wird die kanonische Basis der abgeschnittenen Potenzfunktionen vorgestellt.

Definition 3.5 (Abgeschnittene Potenzfunktion). Sei $j \in \mathbb{N}$. Die Funktionen $\varphi_{i,j} : [\xi_0, \xi_k) \rightarrow \mathbb{R}$ mit

$$\varphi_{i,j}(x) = (x - \xi_i)_+^j := \begin{cases} (x - \xi_i)^j & x \geq \xi_i \\ 0 & \text{sonst} \end{cases} \quad (3.2)$$

für $i = 0, \dots, k-1$ heißen abgeschnittene Potenzfunktionen vom Grad j mit den Sprungstellen ξ .

In [Abbildung 3.3](#) sind abgeschnittene Potenzfunktionen vom Grad $j = 1, 2$ mit den Sprungstellen $\xi = (0, 1, 2, 3, 4)$ dargestellt. Es handelt sich um stückweise lineare bzw. quadratische Polynome. Wie der folgende Satz zeigt, kann mithilfe abgeschnittener Potenzfunktionen eine Basis des Vektorraums der stückweisen Polynome konstruiert werden.

Satz 3.6. Die abgeschnittenen Potenzfunktionen $\varphi_{i,j}$ mit $i = 0, \dots, k-1$ und $j = 0, \dots, d$ bilden eine Basis von $\Pi_{d,\xi}$. Insbesondere ist $\dim \Pi_{d,\xi} = k(d+1)$.

Beweis. Ein Beweis ist in [7, S. 83] angegeben. □

Eine Basis für $\Pi_{d,\xi,\nu}$ lässt sich aus der Basis von $\Pi_{d,\xi}$ konstruieren. Dazu werden die Basisfunktionen eliminiert, welche einen Sprung der $(\nu_i - 1)$ -ten Ableitung in ξ_i verursachen würden.

Satz 3.7. Die abgeschnittenen Potenzfunktionen $\varphi_{0,j}$ für $j = 0, \dots, d$ und φ_{i,j_i} für $i = 1, \dots, k-1$ und $j_i = \nu_i, \dots, d$ bilden eine Basis von $\Pi_{d,\xi,\nu}$. Insbesondere ist

$$\dim \Pi_{d,\xi,\nu} = d + 1 + \sum_{i=1}^{k-1} (d + 1 - \nu_i) = k(d + 1) - \sum_{i=1}^{k-1} \nu_i = \dim \Pi_{d,\xi} - \sum_{i=1}^{k-1} \nu_i.$$

Beweis. Ein Beweis ist in [7, S. 84] angegeben. □

Die abgeschnittenen Potenzfunktionen erlauben zwar eine intuitive Darstellung von ν -regulären stückweisen Polynomen, sie sind jedoch für numerische Berechnungen ungeeignet. Dies hat im Wesentlichen zwei Gründe, welche im Folgenden näher erläutert werden. Dazu wird zunächst der Begriff der Kondition eingeführt.

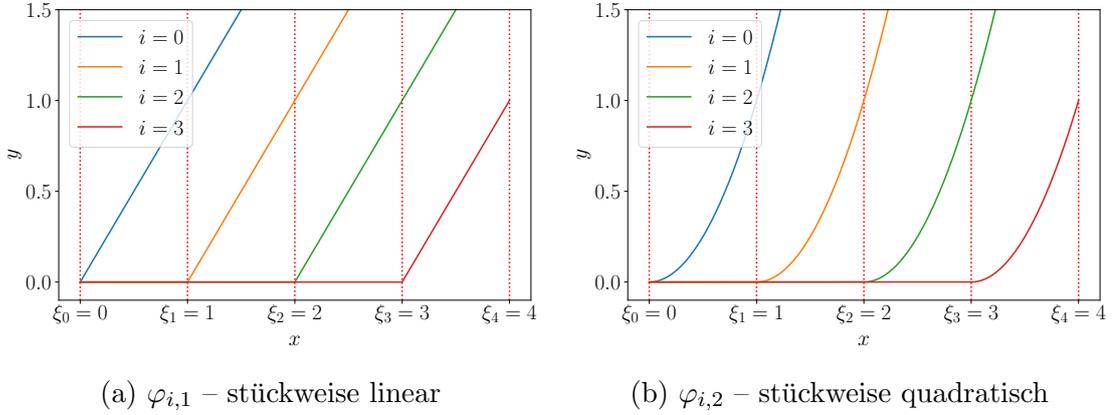


Abbildung 3.3: Abgeschnittene Potenzfunktionen $\varphi_{i,j}$ vom Grad $j = 1, 2$ mit den Sprungstellen $\xi = (0, 1, 2, 3, 4)$

Definition 3.8 (Kondition einer Basis [7, S. 13]). Sei $n_\pi = \dim \Pi_{d,\xi,\nu}$ und $\mathcal{B} = \{b_1, \dots, b_{n_\pi}\}$ eine Basis von $\Pi_{d,\xi,\nu}$. Dann ist die Kondition der Basis \mathcal{B} definiert durch

$$\kappa(\mathcal{B}) := \frac{\max_{\|c\|=1} \left\| \sum_{i=1}^{n_\pi} c_i b_i \right\|}{\min_{\|c\|=1} \left\| \sum_{i=1}^{n_\pi} c_i b_i \right\|}. \quad (3.3)$$

Dabei ist

$$\|f\| := \max_{x \in [\xi_0, \xi_k]} |f(x)| \quad \text{für } f \in \Pi_{d,\xi,\nu}$$

und

$$\|c\| := \max_{i \in \{1, \dots, n_\pi\}} |c_i| \quad \text{für } c \in \mathbb{R}^{n_\pi}.$$

Im weiteren Verlauf des Abschnitts seien die Bezeichnungen n_π , \mathcal{B} und b_i sowie die Definitionen der Normen wie in Definition 3.8. Das folgende Lemma hilft dabei, die Bedeutung der Kondition zu interpretieren.

Lemma 3.9. Sei $c \in \mathbb{R}^{n_\pi}$. Dann gilt

$$\|c\| \min_{\|\tilde{c}\|=1} \left\| \sum_{i=1}^{n_\pi} \tilde{c}_i b_i \right\| \leq \left\| \sum_{i=1}^{n_\pi} c_i b_i \right\| \leq \|c\| \max_{\|\tilde{c}\|=1} \left\| \sum_{i=1}^{n_\pi} \tilde{c}_i b_i \right\|. \quad (3.4)$$

Beweis. Nach Multiplizieren mit $\frac{1}{\|c\|}$ ist die Gleichung (3.4) trivialerweise erfüllt. \square

Die Kondition kann als Maß dafür dienen, wie empfindlich die Basisdarstellung einer Funktion $f \in \Pi_{d,\xi,\nu}$ auf Störungen in den Koeffizienten reagiert. Dies ist insbesondere

dann von Bedeutung, wenn durch numerische Berechnungen Ungenauigkeiten in den Koeffizienten unvermeidbar sind. Seien $c \in \mathbb{R}^{n_\pi}$ die Koeffizienten der Basisdarstellung von f , d. h. $f = \sum_{i=0}^{n_\pi} c_i b_i$. Angenommen, es treten numerische Ungenauigkeiten auf, sodass statt c die gestörten Koeffizienten $c + \Delta c$ gegeben sind. Folglich wird dann durch die Basisdarstellung eine gestörte Funktion $f + \Delta f = \sum_{i=0}^{n_\pi} (c_i + \Delta c_i) b_i$ dargestellt. Nun gilt nach (3.4)

$$\|\Delta f\| \leq \|\Delta c\| \max_{\|\tilde{c}\|=1} \left\| \sum_{i=1}^{n_\pi} \tilde{c}_i b_i \right\| \quad \text{und} \quad \|f\| \geq \|c\| \min_{\|\tilde{c}\|=1} \left\| \sum_{i=1}^{n_\pi} \tilde{c}_i b_i \right\|,$$

also

$$\frac{\|\Delta f\|}{\|f\|} \leq \frac{\|\Delta c\| \max_{\|c\|=1} \left\| \sum_{i=1}^{n_\pi} c_i b_i \right\|}{\|c\| \min_{\|c\|=1} \left\| \sum_{i=1}^{n_\pi} c_i b_i \right\|} \stackrel{(3.3)}{=} \kappa(\mathcal{B}) \frac{\|\Delta c\|}{\|c\|}.$$

Eine relative Störung von $\frac{\|\Delta c\|}{\|c\|}$ in den Koeffizienten verursacht demnach maximal einen relativen Fehler von $\kappa(\mathcal{B}) \frac{\|\Delta c\|}{\|c\|}$ in der Funktion f . Je größer die Kondition der Basis ist, desto stärker können die Auswirkungen kleiner Störungen in den Koeffizienten sein. Deshalb ist es unter numerischen Gesichtspunkten wünschenswert, eine möglichst kleine Kondition zu erreichen.

Wie sich herausstellt, kann die Kondition der Basis der abgeschnittenen Potenzfunktionen beliebig groß werden. Dies wird an einem Beispiel verdeutlicht. Sei $\mathcal{B}_\varphi = \{\varphi_1, \dots, \varphi_{n_\pi}\}$ die Basis der abgeschnittenen Potenzfunktionen. Zur besseren Übersicht werden die Basisfunktionen mit einem fortlaufenden Index versehen. Eine Funktion $f \in \Pi_{d,\xi,\nu}$ besitzt dann die Darstellung

$$f(x) = \sum_{i=0}^{n_\pi} c_i \varphi_i(x) \quad \text{für alle } x \in [\xi_0, \xi_k] \quad (3.5)$$

mit geeigneten Koeffizienten $c_i \in \mathbb{R}$. Wird $c = (1, 0, \dots, 0) \in \mathbb{R}^{n_\pi}$ gewählt, so ist

$$f(x) = \varphi_0(x) = \varphi_{0,0}(x) = (x - \xi_0)_+^0 = 1$$

und damit

$$\max_{\|c\|=1} \left\| \sum_{i=1}^{n_\pi} c_i \varphi_i \right\| \geq 1. \quad (3.6)$$

Im Folgenden sei die Funktion f das in [Abbildung 3.4](#) dargestellte ν -reguläre stückweise Polynom. Die Funktion hat den Grad $d = 1$, die Sprungstellen $\xi = (0.0, 0.5 - \Delta\xi, 0.5, 0.5 + \Delta\xi, 1.0)$ und ist an den inneren Sprungstellen stetig,

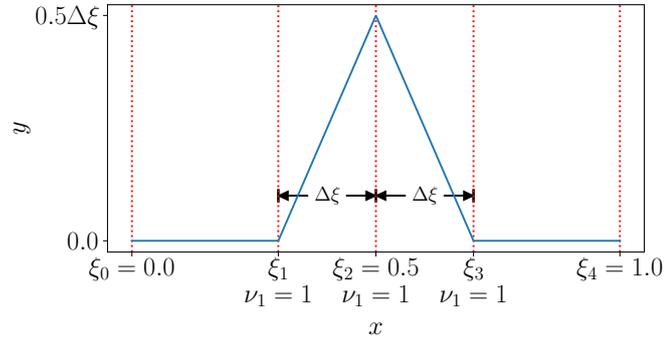


Abbildung 3.4: ν -reguläres stückweises Polynom zur Illustration der schlechten Kondition der Basis der abgeschnittenen Potenzfunktionen

d. h. $\nu = (1, 1, 1)$. Es kann leicht verifiziert werden, dass f in der Basis der abgeschnittenen Potenzfunktionen durch

$$\begin{aligned} f(x) &= 0 \cdot 1 + 0 \cdot x + \frac{1}{2} \cdot (x - \xi_1)_+ - 1 \cdot (x - \xi_2)_+ + \frac{1}{2} \cdot (x - \xi_3)_+ \\ &= 0 \cdot \varphi_{0,0} + 0 \cdot \varphi_{0,1} + \frac{1}{2} \cdot \varphi_{1,1} - 1 \cdot \varphi_{2,1} + \frac{1}{2} \cdot \varphi_{3,1} \end{aligned}$$

dargestellt werden kann. Folglich ist $c = (0, 0, \frac{1}{2}, -1, \frac{1}{2})$ und damit $\|c\| = 1$. Somit ist

$$\min_{\|c\|=1} \left\| \sum_{i=1}^{n_\pi} c_i \varphi_i \right\| \leq \|f(x)\| = \left| f\left(\frac{1}{2}\right) \right| = \frac{1}{2} \Delta \xi. \quad (3.7)$$

Aus (3.6) und (3.7) folgt schließlich für die Kondition der Basis \mathcal{B}_φ

$$\kappa(\mathcal{B}_\varphi) = \frac{\max_{\|c\|=1} \left\| \sum_{i=1}^{n_\pi} c_i \varphi_i \right\|}{\min_{\|c\|=1} \left\| \sum_{i=1}^{n_\pi} c_i \varphi_i \right\|} \geq \frac{2}{\Delta \xi} \xrightarrow{\Delta \xi \downarrow 0} \infty.$$

Für eine ungünstige Folge von Sprungstellen kann die Kondition der Basis der abgeschnittenen Potenzfunktionen daher beliebig groß werden. Störungen in den Koeffizienten können folglich beliebig große Fehler in der Basisdarstellung verursachen.

Ein weiterer Nachteil der Basis der abgeschnittenen Potenzfunktionen ist, dass sich der Träger von $\varphi_{i,j}$ über das Intervall $[\xi_i, \xi_k]$ erstreckt. Auf dem Intervall (ξ_l, ξ_{l+1}) sind deshalb alle Basisfunktionen $\varphi_{i,j}$ mit $i \leq l$ ungleich Null. In der Auswertung der Basisdarstellung (3.5) an einer Stelle x sind also mitunter viele der Basisfunktionen beteiligt. Die Auswertung gestaltet sich dadurch aufwändiger. Dieser Effekt kommt insbesondere bei vielen Sprungstellen und wenn die Basisdarstellung in einem $x \gg \xi_0$ ausgewertet werden soll zum Tragen.

Aus den zuvor diskutierten Gründen ist die Basis der abgeschnittenen Potenzfunktionen für numerische Berechnungen ungeeignet. Abhilfe schafft die Basis der sogenannten B-Splines.

3.3 B-Splines

Eine alternative Basis für den Raum $\Pi_{d,\xi,\nu}$ bilden die sogenannten *B-Splines* (Basis-Splines). Der Name ist dabei etwas irreführend, da es sich bei B-Splines nicht um Splines im klassischen Sinne handelt. Vielmehr sind B-Splines stückweise Polynome. Wie sich herausstellen wird, sind B-Splines den abgeschnittenen Potenzfunktionen als Basis in vielerlei Hinsicht überlegen.

Definition 3.10 (B-Splines). *Seien $d, l \in \mathbb{N}$ mit $l > 2d$ und $\tau = (\tau_i)_{i=0,\dots,l} \in \mathbb{R}^{l+1}$ eine monoton wachsende Folge. Dann sind die B-Splines $B_{i,d,\tau} : [\tau_0, \tau_l) \rightarrow \mathbb{R}$ vom Grad d und der Knotenfolge τ für $d = 0$ durch*

$$B_{i,0,\tau}(x) = \begin{cases} 1 & x \in [\tau_i, \tau_{i+1}) \\ 0 & \text{sonst} \end{cases} \quad (3.8)$$

und für $d > 0$ durch die Rekursion

$$B_{i,d,\tau}(x) = \begin{cases} 0 & \text{falls } \tau_i = \tau_{i+d+1} \\ \frac{x - \tau_i}{\tau_{i+d} - \tau_i} B_{i,d-1,\tau}(x) & \text{falls } \tau_{i+1} = \tau_{i+d+1} \\ \frac{\tau_{i+d+1} - x}{\tau_{i+d+1} - \tau_{i+1}} B_{i+1,d-1,\tau}(x) & \text{falls } \tau_i = \tau_{i+d} \\ \frac{x - \tau_i}{\tau_{i+d} - \tau_i} B_{i,d-1,\tau}(x) \\ + \frac{\tau_{i+d+1} - x}{\tau_{i+d+1} - \tau_{i+1}} B_{i+1,d-1,\tau}(x) & \text{sonst} \end{cases} \quad (3.9)$$

für $i = 0, \dots, l - d - 1$ definiert. Das Intervall $I_{d,\tau} := [\tau_d, \tau_{l-d})$ wird Basisintervall der B-Splines $B_{i,d,\tau}$ genannt.

Im weiteren Verlauf der Arbeit wird die Bezeichnung des Grads d , der Knotenfolge τ und der Anzahl der Knotenpunkte $l + 1$ als gegeben vorausgesetzt. Insbesondere sei die Knotenfolge monoton wachsend und es gelte $l > 2d$. B-Splines zeichnen sich durch folgende Eigenschaften aus.

Satz 3.11 (Eigenschaften von B-Splines). *Es gilt*

$$i. \ B_{i,d,\tau}(x) \geq 0 \quad \text{für } x \in [\tau_0, \tau_{l-1}) \quad \text{und } i = 0, \dots, l - d - 1, \quad (3.10)$$

$$ii. \sum_{i=0}^{l-d-1} B_{i,d,\tau}(x) = 1 \quad \text{für } x \in I_{d,\tau}, \quad (3.11)$$

$$iii. \text{supp } B_{i,d,\tau} = [\tau_i, \tau_{i+d+1}] \quad \text{für } i = 0, \dots, l-d-1, \quad (3.12)$$

$$iv. B_{i,d,\tau}(\tau_{i+d+1}) = 0 \quad \text{für } i = 0, \dots, l-d-1. \quad (3.13)$$

Beweis. Ein Beweis für (3.10), (3.12) und (3.13) ist in [7, S. 91] und für (3.11) in [7, S. 96] angegeben. \square

B-Splines haben folglich einen kompakten Träger und aus (3.10) und (3.11) folgt, dass ihr Wertebereich sich auf das Intervall $[0, 1]$ beschränkt. Mithilfe von Satz 3.11 lässt sich die Fallunterscheidung in (3.9) vereinfachen. Das folgende Korollar dient als Motivation für die Vereinfachung.

Korollar 3.12. *Es gilt*

$$\lim_{\tau_i \rightarrow \tau_{i+d+1}} \frac{1}{\tau_{i+d+1} - \tau_i} B_{i,d,\tau}(x) = 0 \quad \forall x \in [\tau_0, \tau_l]. \quad (3.14)$$

Beweis. Fall 1: $x > \tau_{i+d+1}$. Dann ist $x \notin \text{supp } B_{i,d,\tau}$ nach (3.12) und folglich $\frac{1}{\tau_{i+d+1} - \tau_i} B_{i,d,\tau}(x) = 0$.

Fall 2: $x < \tau_{i+d+1}$. Sei $\tau_{i+d+1} > \tau_i > x$. Dann ist $x \notin \text{supp } B_{i,d,\tau}$ nach (3.12) und folglich $\frac{1}{\tau_{i+d+1} - \tau_i} B_{i,d,\tau}(x) = 0$.

Fall 3: $x = \tau_{i+d+1}$. Dann ist $\frac{1}{\tau_{i+d+1} - \tau_i} B_{i,d,\tau}(\tau_{i+d+1}) = 0$ nach (3.13). \square

Aus Korollar 3.12 folgt, dass

$$\lim_{\tau_i \rightarrow \tau_{i+d}} \frac{x - \tau_i}{\tau_{i+d} - \tau_i} B_{i,d-1,\tau}(x) = 0 \quad \forall x \in [\tau_0, \tau_l]$$

und

$$\lim_{\tau_{i+1} \rightarrow \tau_{i+d+1}} \frac{\tau_{i+d+1} - x}{\tau_{i+d+1} - \tau_{i+1}} B_{i+1,d-1,\tau}(x) = 0 \quad \forall x \in [\tau_0, \tau_l].$$

Darüber hinaus ist per Definition $B_{i,d-1,\tau}(x) = 0$ falls $\tau_i = \tau_{i+d}$ und $B_{i+1,d-1,\tau}(x) = 0$ falls $\tau_{i+1} = \tau_{i+d+1}$. Dieses Resultat dient als Motivation, um im Folgenden in (3.9) auf die Fallunterscheidung zu verzichten und lediglich

$$B_{i,d,\tau}(x) = \frac{x - \tau_i}{\tau_{i+d} - \tau_i} B_{i,d-1,\tau}(x) + \frac{\tau_{i+d+1} - x}{\tau_{i+d+1} - \tau_{i+1}} B_{i+1,d-1,\tau}(x) \quad (3.15)$$

zu schreiben. Sollte in (3.15) einer der Nenner Null sein, ist der zugehörige Term durch Null zu ersetzen.

In [Abbildung 3.5](#) sind B-Splines der Grade $d = 1$ und $d = 2$ für verschiedene Knotenfolgen dargestellt. [Abbildung 3.5a](#) zeigt B-Splines vom Grad $d = 1$ für die äquidistante Knotenfolge $\tau = (0, 1, 2, 3, 4, 5, 6, 7)$. Es handelt sich um stetige Funktionen, welche bis auf eine Verschiebung in x -Richtung identisch sind. Das Basisintervall erstreckt sich von $\tau_1 = 1$ bis $\tau_6 = 6$. Es ist zu erkennen, dass die Summe der B-Splines auf dem Basisintervall konstant Eins ist. Alle B-Splines $B_{i,1,\tau}$ sind positiv und lediglich zwischen τ_i und τ_{i+2} ungleich Null sind.

[Abbildung 3.5b](#) zeigt die B-Splines vom Grad $d = 2$ für dieselbe Knotenfolge. Analog zum Fall $d = 1$ sind die Funktionen bis auf eine Verschiebung in x -Richtung identisch und auf dem Basisintervall summieren sie sich zu Eins auf. Die B-Splines sind stetig differenzierbar und ihr Träger beschränkt sich auf das Intervall $[\tau_i, \tau_{i+3}]$.

In [Abbildungen 3.5c](#) und [3.5d](#) sind B-Splines vom Grad $d = 2$ dargestellt. Im Gegensatz zu [Abbildung 3.5b](#) ist der Knoten $\tau_3 = 3$ doppelt bzw. dreifach in der Knotenfolge enthalten. Im Falle eines doppelten Knotens wird ein weiterer B-Spline erzeugt, welcher in $x = 3$ nicht mehr differenzierbar ist (s. [Abb. 3.5c](#)). Im Falle eines dreifachen Knotens entstehen zwei neue B-Splines, welche in $x = 3$ eine Unstetigkeit aufweisen (s. [Abb. 3.5d](#)).

In [Abbildung 3.5e](#) sind B-Splines vom Grad $d = 2$ abgebildet, in deren Knotenfolge der erste und letzte Knoten dreifach enthalten ist. Dies hat zur Folge, dass sich das Basisintervall vom ersten bis zum letzten Knoten erstreckt. [Abbildung 3.5f](#) zeigt B-Splines vom Grad $d = 2$ für eine unregelmäßige Knotenfolge mit einem doppelten Knoten in $x = 3$ und zwei dreifachen Knoten in $x = 0$ und $x = 4$. Es ist gut zu erkennen, dass die resultierenden B-Splines alle Eigenschaften aus [Satz 3.11](#) aufweisen.

3.3.1 Basis der B-Splines

Ihren Namen verdanken B-Splines der Tatsache, dass sie für eine geeignete Wahl der Knotenfolge τ eine Basis des Vektorraums $\Pi_{d,\xi,\nu}$ bilden. Der folgende Satz fasst dieses Resultat zusammen.

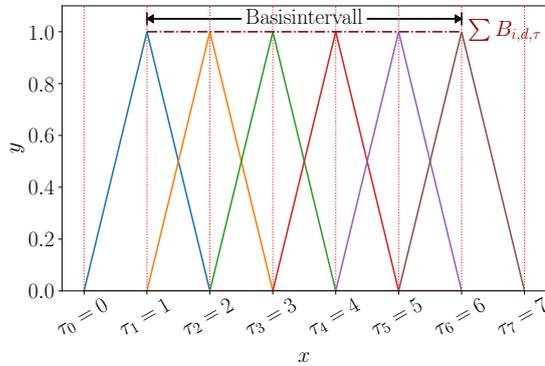
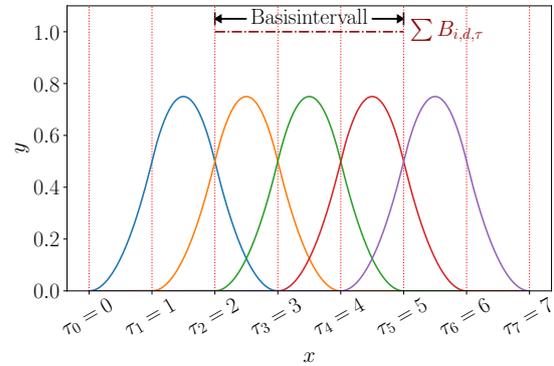
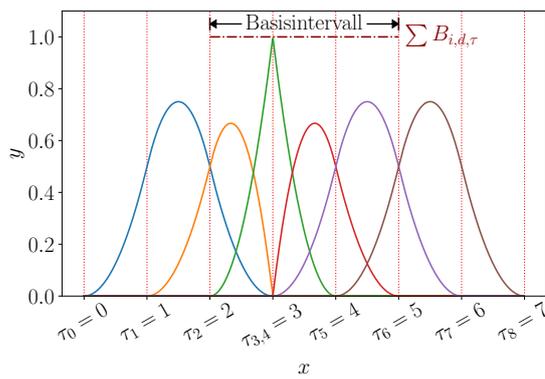
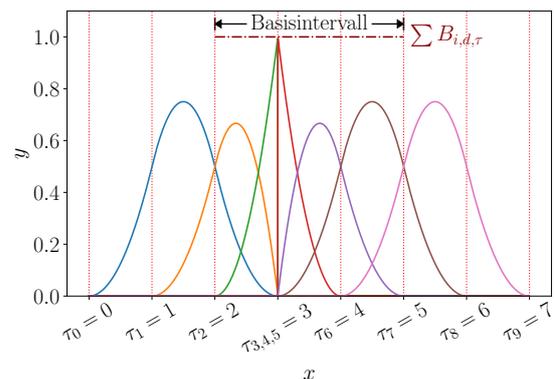
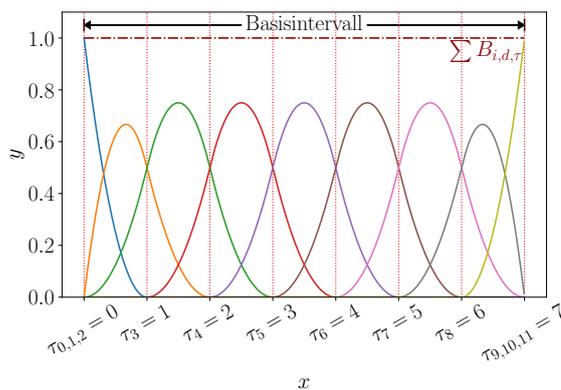
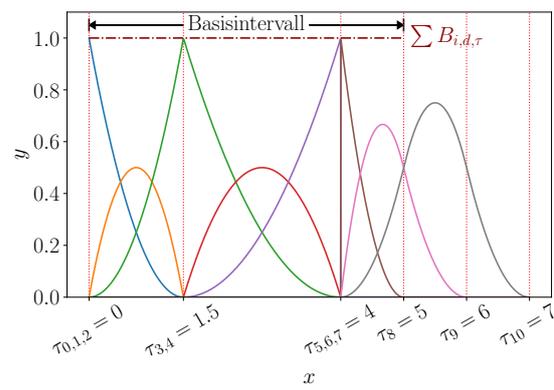
Satz 3.13 (Satz von Curry und Schoenberg). *Sei $n_\pi = \dim \Pi_{d,\xi,\nu}$. Die monoton wachsende Knotenfolge $\tau = (\tau_i)_{i=0,\dots,n_\pi+d} \in \mathbb{R}^{n_\pi+d+1}$ habe die Eigenschaften:*

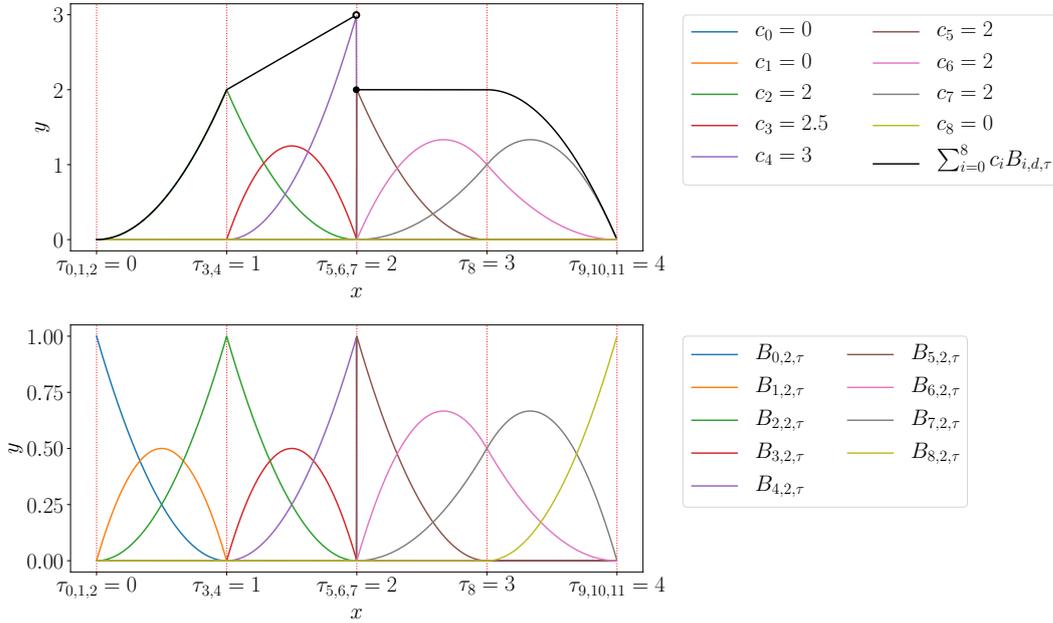
$$i. \text{ für } i = 1, \dots, k-1 \text{ ist } \xi_i \text{ genau } (d+1-\nu_i)\text{-Mal in } \tau \text{ enthalten,} \quad (3.16)$$

$$ii. \tau_0 \leq \tau_1 \leq \dots \leq \tau_d = \xi_0 \text{ und } \xi_k = \tau_{n_\pi} \leq \tau_{n_\pi+1} \leq \dots \leq \tau_{n_\pi+d}. \quad (3.17)$$

Dann bilden die B-Splines $B_{i,d,\tau}$ für $i = 0, \dots, n_\pi - 1$ eine Basis von $\Pi_{d,\xi,\nu}$ auf dem Basisintervall $I_{d,\tau} = [\tau_d, \tau_{n_\pi}] = [\xi_0, \xi_k]$.

Beweis. Ein Beweis ist in [\[7, S. 97f.\]](#) ausgeführt. □

(a) $d = 1$ und äquidistante Knoten(b) $d = 2$ und äquidistante Knoten(c) $d = 2$ und doppelter Knoten(d) $d = 2$ und dreifacher Knoten(e) $d = 2$ und dreifache Randknoten(f) $d = 2$ und unregelmäßige KnotenAbbildung 3.5: B-Splines vom Grad $d = 1, 2$ und mit verschiedenen Knotenfolgen

Abbildung 3.6: Basisdarstellung durch B-Splines vom Grad $d = 2$

Satz 3.13 lässt die Wahl der d ersten und der d letzten Knoten offen. Es muss lediglich $\tau_0 \leq \tau_1 \leq \dots \leq \tau_d$ und $\tau_{n_\pi} \leq \tau_{n_\pi+1} \leq \dots \leq \tau_{n_\pi+d}$ gelten. Eine zweckmäßige Wahl ist

$$\tau_0 = \tau_1 = \dots = \tau_d = \xi_0 \quad \text{und} \quad \xi_k = \tau_{n_\pi} = \tau_{n_\pi+1} = \dots = \tau_{n_\pi+d}.$$

Dies würde der Eigenschaft (3.16) mit der Wahl $\nu_0 = \nu_k = 0$ entsprechen. In gewisser Weise würde keine Stetigkeit der B-Splines an den Endpunkten ξ_0 und ξ_k gefordert werden.

Sei $\tau \in \mathbb{R}^{l+1}$ die gemäß (3.16) und (3.17) konstruierte Knotenfolge. Dann kann $f \in \Pi_{d,\xi,\nu}$ auf dem Basisintervall $I_{d,\tau} = [\tau_d, \tau_{l-d}) = [\xi_0, \xi_k)$ als Linearkombination der B-Spline Basis dargestellt werden:

$$f(x) = \sum_{i=0}^{l-d-1} c_i B_{i,d,\tau}(x) \quad \text{für} \quad x \in I_{d,\tau}. \quad (3.18)$$

In [Abbildung 3.6](#) ist das stückweise Polynom aus [Abbildung 3.2a](#) als Linearkombination von B-Splines des Grads $d = 2$ dargestellt. Der untere Plot zeigt die Basisfunktionen. Da die darzustellende Funktion in $x = 1$ nicht differenzierbar ist, enthält die Knotenfolge den doppelten Knoten $\tau_3 = \tau_4 = 1$. An der Stelle $x = 2$ ist die darzustellende Funktion unstetig. Aus diesem Grund ist der Knoten $\tau_5 = \tau_6 = \tau_7 = 2$ dreifach in der Knotenfolge enthalten. Der obere Plot zeigt das Produkt der B-Splines $B_{i,2,\tau}$ mit den Koeffizienten c_i sowie die Linearkombination dieser Terme.

In [Abschnitt 3.2](#) wurden zwei Schwachpunkte der Basis der abgeschnittenen Potenzfunktionen ausgemacht. Einer der Schwachpunkte war, dass sich der Träger abgeschnittener Potenzfunktionen über einen großen Teil des Definitionsbereichs erstreckt. Deswegen mussten für die Auswertung der Basisdarstellung in einem Punkt viele Basisfunktionen ausgewertet werden. B-Splines zeichnen sich dadurch aus, dass ihr Träger kompakt ist (s. [Satz 3.11](#)). Um die Basisdarstellung [\(3.18\)](#) auszuwerten, müssen deshalb lediglich $d + 1$ Basisfunktionen evaluiert werden. Die B-Splines sind den abgeschnittenen Potenzfunktionen in diesem Punkt überlegen. Der zweite Schwachpunkt der Basis der abgeschnittenen Potenzfunktionen ist, dass ihre Kondition beliebig groß werden kann. Störungen in den Koeffizienten der Basisdarstellung können deshalb zu großen Fehlern in der dargestellten Funktion führen. Auch in diesem Punkt sind B-Splines überlegen. [Satz 3.14](#) liefert die Grundlage dafür.

Satz 3.14. *Es existiert eine positive Konstante $C_d \in \mathbb{R}$, mit der für alle $c \in \mathbb{R}^{l-d}$ gilt*

$$C_d^{-1} \|c\| \leq \left\| \sum_{i=0}^{l-d-1} c_i B_{i,d,\tau} \right\| \leq \|c\|. \quad (3.19)$$

Insbesondere hängt C_d nur vom Grad d und nicht von der Knotenfolge τ ab. Die Normen sind dabei wie in [Definition 3.8](#) die Maximumsnorm für Funktionen bzw. Vektoren.

Beweis. Ein Beweis ist in [\[7, S. 133\]](#) ausgeführt. □

Aus [Satz 3.14](#) folgt eine obere Schranke für die Kondition κ der B-Spline Basis.

Korollar 3.15. *Sei $\tau = (\tau_i)_{i=0,\dots,l} \in \mathbb{R}^{l+1}$ die gemäß [\(3.16\)](#) und [\(3.17\)](#) konstruierte Knotenfolge und $\mathcal{B}_B = \{B_{0,d,\tau}, \dots, B_{l-d-1,d,\tau}\}$ die Basis der B-Splines für $\Pi_{d,\xi,\nu}$. Dann gilt*

$$\kappa(\mathcal{B}_B) \leq C_d$$

mit der Konstanten C_d aus [Satz 3.14](#).

Beweis. Sei $c \in \mathbb{R}^{l-d}$. Aus [\(3.19\)](#) folgt

$$\frac{1}{\|c\|} \left\| \sum_{i=0}^{l-d-1} c_i B_{i,d,\tau} \right\| \leq 1 \quad \Rightarrow \quad \max_{\|c\|=1} \left\| \sum_{i=0}^{l-d-1} c_i B_{i,d,\tau} \right\| \leq 1$$

und

$$\frac{1}{\|c\|} \left\| \sum_{i=0}^{l-d-1} c_i B_{i,d,\tau} \right\| \geq C_d^{-1} \quad \Rightarrow \quad \min_{\|c\|=1} \left\| \sum_{i=0}^{l-d-1} c_i B_{i,d,\tau} \right\| \geq C_d^{-1}.$$

Folglich ist

$$\kappa(\mathcal{B}_B) = \frac{\max_{\|c\|=1} \left\| \sum_{i=0}^{l-d-1} c_i B_{i,d,\tau} \right\|}{\min_{\|c\|=1} \left\| \sum_{i=0}^{l-d-1} c_i B_{i,d,\tau} \right\|} \leq \frac{1}{C_d^{-1}} = C_d. \quad \square$$

Die Konstante C_d ist also eine obere Schranke für die Kondition der B-Spline Basis. Es sei erneut darauf hingewiesen, dass C_d unabhängig von der Wahl der Knotenfolge und damit auch unabhängig von den Sprungstellen der ν -regulären stückweisen Polynome aus $\Pi_{d,\xi,\nu}$ ist. Dies ist bemerkenswert, da die Kondition der Basis der abgeschnittenen Potenzfunktionen beliebig groß werden kann, wenn die Sprungstellen ungünstig gewählt werden (vgl. [Abschnitt 3.2](#)). Die Basis der B-Splines weist keinen der beiden Schwachpunkte der Basis der abgeschnittenen Potenzfunktionen auf und ist deshalb in numerischen Berechnungen zu bevorzugen.

[Satz 3.14](#) liefert lediglich die Existenz der Konstante C_d . Auf der Grundlage von numerischen Experimenten wird vermutet, dass $C_d \propto 2^{d-\frac{1}{2}}$ ist. Ein Beweis dafür konnte jedoch noch nicht gefunden werden. Scherer et. al [61] konnten lediglich zeigen, dass $C_d \leq (d+1)2^d$ ist und damit die Vermutung bis auf einen polynomiellen Faktor bestätigen.

3.3.2 Ableitung von B-Splines

In diesem Abschnitt wird die Ableitung von B-Splines diskutiert. Mithilfe dieser Ableitung kann eine Formel für die Ableitung von ν -regulären stückweisen Polynomen hergeleitet werden. Die Ableitungen sind dabei im Sinne von [Definition 3.3](#) zu verstehen. Die Überlegungen dieses Kapitels beruhen auf dem folgenden Satz.

Satz 3.16. *Es gilt*

$$\frac{d}{dx} B_{i,d,\tau}(x) = \begin{cases} 0 & \text{falls } \tau_i = \tau_{i+d+1} \\ \frac{d}{\tau_{i+d} - \tau_i} B_{i,d-1,\tau}(x) & \text{falls } \tau_{i+1} = \tau_{i+d+1} \\ \frac{d}{\tau_{i+d+1} - \tau_{i+1}} B_{i+1,d-1,\tau}(x) & \text{falls } \tau_i = \tau_{i+d} \\ \frac{d}{\tau_{i+d} - \tau_i} B_{i,d-1,\tau}(x) & \\ -\frac{d}{\tau_{i+d+1} - \tau_{i+1}} B_{i+1,d-1,\tau}(x) & \text{sonst} \end{cases} \quad (3.20)$$

für $i = 0, \dots, l-d-1$.

Beweis. Ein Beweis ist in [7, S. 115 f.] angegeben. □

Wie bei der Definition der B-Splines (3.9) wird im Folgenden auf die Fallunterscheidung in (3.20) verzichtet und lediglich

$$\frac{d}{dx}B_{i,d,\tau}(x) = \frac{d}{\tau_{i+d} - \tau_i}B_{i,d-1,\tau}(x) - \frac{d}{\tau_{i+d+1} - \tau_{i+1}}B_{i+1,d-1,\tau}(x) \quad (3.21)$$

geschrieben. Per Konvention werden die Terme, in denen als Nenner Null auftritt, durch Null ersetzt. Dies ist dadurch motiviert, dass die Funktionen $B_{i,d-1,\tau}$ und $B_{i+1,d-1,\tau}$ per Definition die Nullfunktionen sind, falls $\tau_i = \tau_{i+d}$ bzw. $\tau_{i+1} = \tau_{i+d+1}$. Darüber hinaus konvergieren $\frac{d}{\tau_{i+d}-\tau_i}B_{i,d-1,\tau}$ und $\frac{d}{\tau_{i+d+1}-\tau_{i+1}}B_{i+1,d-1,\tau}$ nach [Korollar 3.12](#) punktweise gegen die Nullfunktion für $\tau_i \rightarrow \tau_{i+d}$ bzw. $\tau_{i+1} \rightarrow \tau_{i+d+1}$.

In [Abbildung 3.7](#) sind die Ableitungen einiger B-Splines dargestellt. [Abbildung 3.7a](#) zeigt die Ableitung eines B-Splines vom Grad $d = 1$ mit äquidistanten Knoten. Der B-Spline ist stückweise linear mit einem „Knick“ in $x = 1$. Die Ableitung ist folglich eine stückweise konstante Funktion mit einem Sprung in $x = 1$. Im Punkt $x = 1$ ist die Ableitung nach [Definition 3.3](#) rechtsseitig zu verstehen. In [Abbildung 3.7b](#) ist die Ableitung eines B-Splines vom Grad $d = 2$ mit äquidistanten Knoten abgebildet. Der B-Spline ist stetig differenzierbar und folglich ist die Ableitung eine stetige stückweise lineare Funktion. Im Falle eines doppelten Knoten in $x = 1$ ist der B-Spline an dieser Stelle nicht mehr differenzierbar. Dies ist in [Abbildung 3.7c](#) zu sehen. Die Ableitung ist folglich an der Stelle $x = 1$ unstetig. Die Ableitung ist auch hier rechtsseitig zu verstehen. [Abbildung 3.7d](#) zeigt die Ableitung eines B-Splines vom Grad $d = 3$. Dabei handelt es sich um eine stückweise quadratische Funktion. Ein doppelter Knoten in $x = 1$ hat einen Sprung in der Krümmung des B-Splines zur Folge. Die Ableitung hat deshalb einen „Knick“ an dieser Stelle.

Um die Ableitung eines ν -regulären stückweisen Polynoms $f \in \Pi_{d,\xi,\nu}$ zu bestimmen, kann die Darstellung von f in der Basis der B-Splines (3.18) herangezogen werden. Für die Ableitung von f folgt damit

$$\begin{aligned} f^{(1)}(x) &\stackrel{(3.18)}{=} \sum_{i=0}^{l-d-1} c_i \frac{d}{dx} B_{i,d,\tau}(x) \\ &\stackrel{(3.21)}{=} \sum_{i=0}^{l-d-1} c_i \left(\frac{d}{\tau_{i+d} - \tau_i} B_{i,d-1,\tau}(x) - \frac{d}{\tau_{i+d+1} - \tau_{i+1}} B_{i+1,d-1,\tau}(x) \right) \\ &= \sum_{i=0}^{l-d-1} \frac{c_i d}{\tau_{i+d} - \tau_i} B_{i,d-1,\tau}(x) - \sum_{i=0}^{l-d-1} \frac{c_i d}{\tau_{i+d+1} - \tau_{i+1}} B_{i+1,d-1,\tau}(x) \\ &= \sum_{i=0}^{l-d-1} \frac{c_i d}{\tau_{i+d} - \tau_i} B_{i,d-1,\tau}(x) - \sum_{i=1}^{l-d} \frac{c_{i-1} d}{\tau_{i+d} - \tau_i} B_{i,d-1,\tau}(x) \end{aligned}$$

(Fortsetzung der Rechnung auf Seite 35)

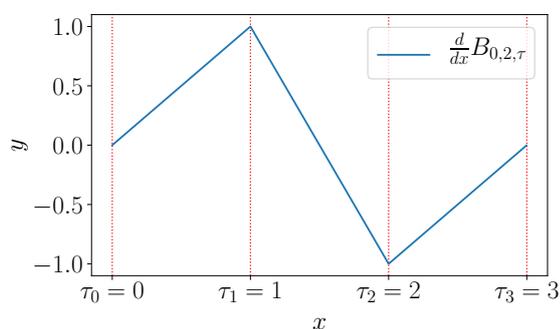
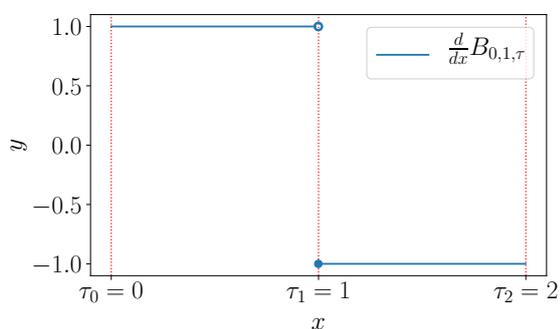
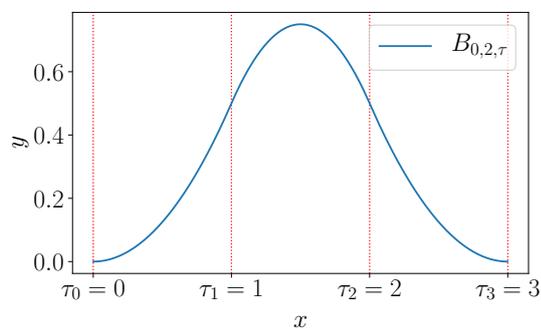
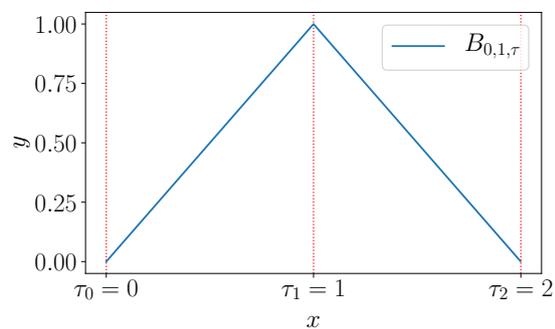
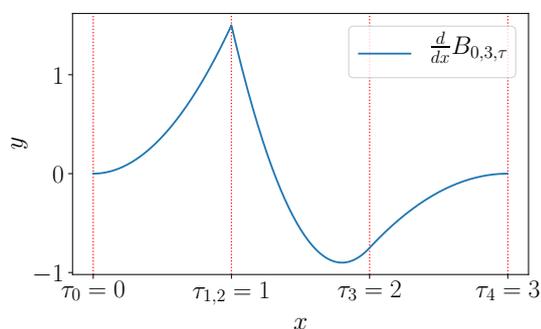
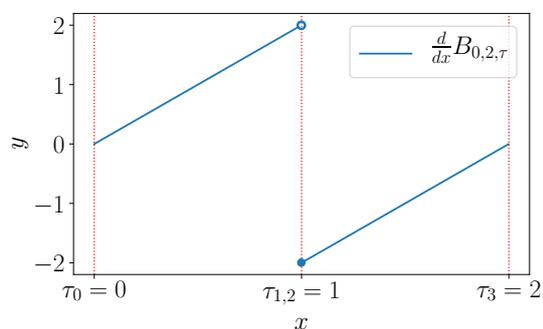
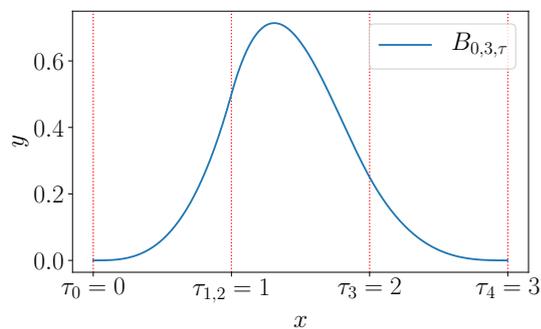
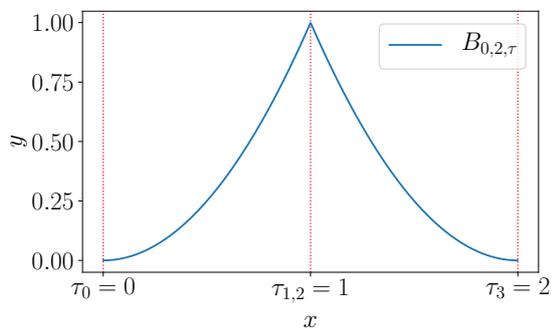
(a) $d = 1$ und äquidistante Knoten(b) $d = 2$ und äquidistante Knoten(c) $d = 2$ und doppelter Knoten(d) $d = 3$ und doppelter Knoten

Abbildung 3.7: Ableitungen von B-Splines

(Fortsetzung der Rechnung von Seite 33)

$$\begin{aligned}
&= \frac{c_0 d}{\tau_d - \tau_0} B_{0,d-1,\tau}(x) + \sum_{i=1}^{l-d-1} \frac{d(c_i - c_{i-1})}{\tau_{i+d} - \tau_i} B_{i,d-1,\tau}(x) \\
&\quad - \frac{c_{l-d-1} d}{\tau_l - \tau_{l-d}} B_{l-d,d-1,\tau}(x) \\
&\stackrel{(3.12)}{=} \sum_{i=1}^{l-d-1} \frac{d(c_i - c_{i-1})}{\tau_{i+d} - \tau_i} B_{i,d-1,\tau}(x). \tag{3.22}
\end{aligned}$$

Gleichung (3.22) ist mit Vorsicht zu genießen, da der Fall $\tau_i = \tau_{i+d}$ undefiniert ist. Wie bereits argumentiert, ist in diesem Fall jedoch $B_{i,d-1,\tau}$ die Nullfunktion und $\frac{d(c_i - c_{i-1})}{\tau_{i+d} - \tau_i} B_{i,d-1,\tau}$ konvergiert punktweise gegen die Nullfunktion für $\tau_i \rightarrow \tau_{i+d}$ nach Korollar 3.12. Die Ableitung von f lässt sich folglich durch eine Linearkombination von B-Splines der Ordnung $d - 1$ mit den Koeffizienten

$$q_i := \begin{cases} d \frac{c_i - c_{i-1}}{\tau_{i+d} - \tau_i} & \text{falls } \tau_i \neq \tau_{i+d} \\ 0 & \text{sonst} \end{cases} \quad \text{für } i = 1, \dots, l - d - 1$$

darstellen. Höhere Ableitungen von f lassen sich damit analog zur obigen Herleitung von $f^{(1)}$ bestimmen. Sukzessives Anwenden von (3.22) führt auf

$$f^{(n)}(x) = \sum_{i=n}^{l-d-1} q_i^{[n]} B_{i,d-n,\tau}(x) \quad \text{für } x \in I_{d,\tau} \tag{3.23}$$

mit

$$q_i^{[n]} := \begin{cases} c_i & \text{falls } n = 0 \\ (d + 1 - n) \frac{q_i^{[n-1]} - q_{i-1}^{[n-1]}}{\tau_{i+d+1-n} - \tau_i} & \text{falls } n \geq 1 \text{ und } \tau_i \neq \tau_{i+d+1-n} \\ 0 & \text{sonst} \end{cases} \tag{3.24}$$

für $0 \leq n \leq d$ und $i = n, \dots, l - d - 1$. Für $n > d$ ist $f^{(n)}$ trivialerweise die Nullfunktion.

3.3.3 Effiziente Berechnung von B-Splines

In numerischen Berechnungen müssen Funktionen häufig sehr oft ausgewertet werden. Deshalb ist es wichtig, dass die Auswertung möglichst effizient geschieht. In diesem Abschnitt werden zwei Algorithmen vorgestellt, welche eine effiziente Berechnung von B-Splines und damit auch von ν -regulären stückweisen Polynomen ermöglichen.

Sei $f \in \Pi_{d,\xi,\nu}$ und τ die gemäß (3.16) und (3.17) konstruierte Knotenfolge. Aus der Eigenschaft (3.12) folgt, dass auf dem Intervall $[\tau_j, \tau_{j+1})$ lediglich die B-Splines $B_{i,d,\tau}$ mit $i = j - d, \dots, j$ ungleich Null sind. Die Basisdarstellung (3.18) kann deshalb zu

$$f(x) = \sum_{i=j-d}^j c_i B_{i,d,\tau}(x) \quad \text{für } x \in [\tau_j, \tau_{j+1}). \quad (3.25)$$

vereinfacht werden. Dieser Umstand lässt sich ausnutzen, um effiziente Algorithmen zum Auswerten von f herzuleiten. Analog zu (3.25) vereinfacht sich die Darstellung (3.23) von $f^{(n)}$ zu

$$f^{(n)}(x) = \sum_{i=j-d+n}^j q_i^{[n]} B_{i,d-n,\tau}(x) \quad \text{für } x \in [\tau_j, \tau_{j+1}). \quad (3.26)$$

Die Koeffizienten $q_i^{[n]}$ lassen sich mithilfe der Rekursion

$$q_i^{[n]} = \begin{cases} c_i & n = 0 \\ (d+1-n) \frac{q_i^{[n-1]} - q_{i-1}^{[n-1]}}{\tau_{i+d+1-n} - \tau_i} & n > 0 \end{cases} \quad (3.27)$$

für $0 \leq n \leq d$ und $i = j - d + n, \dots, j$ aus den Koeffizienten c_i bestimmen. Im Unterschied zu (3.24) kann in (3.27) auf die Fallunterscheidung, ob $\tau_i = \tau_{i+d+1-n}$ ist, verzichtet werden, wie Lemma 3.17 beweist.

Lemma 3.17. *Seien $j, n \in \mathbb{N}$ mit $0 \leq n \leq d$. Des Weiteren sei $\tau_j < \tau_{j+1}$. Dann gilt*

$$\tau_{i+d+1-n} - \tau_i \neq 0 \quad \text{für } i = j - d + n, \dots, j$$

Beweis. Da τ monoton steigend ist, gilt

$$i \leq j \Rightarrow \tau_i \leq \tau_j$$

und

$$i \geq j - d + n \Rightarrow \tau_{i+d+1-n} \geq \tau_{j+1}.$$

Folglich ist $\tau_i \leq \tau_j < \tau_{j+1} \leq \tau_{i+d+1-n}$, also $\tau_{i+d+1-n} - \tau_i \neq 0$. \square

Sowohl (3.25) als auch (3.26) lassen sich darauf reduzieren, die Werte der B-Splines zu bestimmen, welche auf dem Intervall $[\tau_j, \tau_{j+1})$ ungleich Null sind. Im Folgenden werden zwei Algorithmen vorgestellt, welche diese Berechnung durchführen. Für eine übersichtlichere Darstellung wird für die Herleitung $f(x)$ herangezogen. Es sei darauf hingewiesen, dass beide Algorithmen auch verwendet werden können, um die Ableitung $f^{(n)}(x)$ auszuwerten. Dazu müssen lediglich die Koeffizienten der Ableitung

mithilfe von (3.27) berechnet und die Algorithmen mit dem Grad $d - n$ durchgeführt werden. Im weiteren Verlauf sei der Index j so gewählt, dass $x \in [\tau_j, \tau_{j+1})$ ist. Insbesondere sei $\tau_j < \tau_{j+1}$.

De Boors erster Algorithmus De Boors erster Algorithmus greift auf die Rekursionsformel (3.15) zurück. Für $x \in [\tau_j, \tau_{j+1})$ gilt

$$\begin{aligned}
f(x) &= \sum_{i=j-d}^j c_i B_{i,d,\tau}(x) \\
&\stackrel{(3.15)}{=} \sum_{i=j-d}^j \frac{(x - \tau_i) c_i}{\tau_{i+d} - \tau_i} B_{i,d-1,\tau}(x) + \sum_{i=j-d}^j \frac{(\tau_{i+d+1} - x) c_i}{\tau_{i+d+1} - \tau_{i+1}} B_{i+1,d-1,\tau}(x) \\
&= \sum_{i=j-d}^j \frac{(x - \tau_i) c_i}{\tau_{i+d} - \tau_i} B_{i,d-1,\tau}(x) + \sum_{i=j-d+1}^{j+1} \frac{(\tau_{i+d} - x) c_{i-1}}{\tau_{i+d} - \tau_i} B_{i,d-1,\tau}(x) \\
&= \sum_{i=j-d+1}^j \frac{(x - \tau_i) c_i + (\tau_{i+d} - x) c_{i-1}}{\tau_{i+d} - \tau_i} B_{i,d-1,\tau}(x) \\
&\quad + \frac{(x - \tau_{j-d}) c_{j-d}}{\tau_j - \tau_{j-d}} B_{j-d,d-1,\tau}(x) + \frac{(\tau_{j+1+d} - x) c_j}{\tau_{j+1+d} - \tau_{j+1}} B_{j+1,d-1,\tau}(x) \\
&\stackrel{(3.12)}{=} \sum_{i=j-d+1}^j \frac{(x - \tau_i) c_i + (\tau_{i+d} - x) c_{i-1}}{\tau_{i+d} - \tau_i} B_{i,d-1,\tau}(x).
\end{aligned}$$

Folglich ist

$$f(x) = \sum_{i=j-d}^j c_i B_{i,d,\tau}(x) = \sum_{i=j-(d-1)}^j r_i(x) B_{i,d-1,\tau}(x) \quad \text{für } x \in [\tau_j, \tau_{j+1})$$

mit

$$r_i(x) := \frac{(x - \tau_i) c_i + (\tau_{i+d} - x) c_{i-1}}{\tau_{i+d} - \tau_i}, \quad i = j - (d - 1), \dots, j.$$

Sukzessives Anwenden dieser Prozedur führt auf den Zusammenhang

$$f(x) = \sum_{i=j-(d-k)}^j r_i^{[k]}(x) B_{i,d-k,\tau}(x) \quad \text{für } x \in [\tau_j, \tau_{j+1}) \quad (3.28)$$

mit

$$r_i^{[k]}(x) := \begin{cases} c_i & k = 0 \\ \frac{(x - \tau_i) r_i^{[k-1]}(x) + (\tau_{i+d+1-k} - x) r_{i-1}^{[k-1]}(x)}{\tau_{i+d+1-k} - \tau_i} & k > 0 \end{cases} \quad (3.29)$$

für $0 \leq k \leq d$ und $i = j - (d - k), \dots, j$. Der Fall $\tau_i = \tau_{i+d+1-k}$ ist nach Lemma 3.17

$$\begin{array}{ccccccc}
c_{j-d} = r_{j-d}^{[0]}(x) & & & & & & \\
c_{j-d+1} = r_{j-d+1}^{[0]}(x) & \longrightarrow & r_{j-d+1}^{[1]}(x) & & & & \\
\vdots & & & \cdots & & & \\
c_{j-1} = r_{j-1}^{[0]}(x) & & & & r_{j-1}^{[d-1]}(x) & & \\
c_j = r_j^{[0]}(x) & \longrightarrow & r_j^{[1]}(x) & \cdots & r_j^{[d-1]}(x) & \longrightarrow & r_j^{[d]}(x)
\end{array}$$

Abbildung 3.8: Zusammenhang zwischen den Polynomen $r_i^{[k]}(x)$

ausgeschlossen. Das ν -reguläre stückweise Polynom f vom Grad d kann folglich als Linearkombination von B-Splines vom Grad $d - k$ dargestellt werden. Anstelle der konstanten Koeffizienten c_i treten dann Polynome $r_i^{[k]}(x)$ vom Grad k . Setzt man nun $k = d$, so vereinfacht sich (3.28) zu

$$f(x) = r_j^{[d]}(x)B_{j,0,\tau}(x) = r_j^{[d]}(x) \quad \text{für } x \in [\tau_j, \tau_{j+1}).$$

Der Wert $f(x)$ kann also bestimmt werden, indem das rekursiv definierte Polynom $r_j^{[d]}(x)$ ausgewertet wird.

In [Abbildung 3.8](#) ist der Zusammenhang zwischen den Polynomen $r_i^{[k]}(x)$ dargestellt. Ausgehend von den Koeffizienten $c_i = r_i^{[0]}(x)$ kann der Wert $r_j^{[d]}(x) = f(x)$ mithilfe der Rekursionsformel (3.29) berechnet werden. Diese Rekursion bildet den Kern von de Boors erstem Algorithmus. Durch geschicktes Ausführen kann sie sehr effizient geschehen. Insbesondere muss der Fall, dass der Nenner in (3.29) gleich Null ist, nicht behandelt werden. Per Konstruktion ist dieser Fall ausgeschlossen. In [Anhang A.1](#) ist eine ausführliche Herleitung von de Boors erstem Algorithmus dargestellt. In [Algorithmus 2](#) ist der Algorithmus zusammengefasst.

De Boors zweiter Algorithmus Auch de Boors zweiter Algorithmus basiert auf der Rekursionsformel (3.15). Es wird ausgenutzt, dass auf dem Intervall $[\tau_j, \tau_{j+1})$ nach [Satz 3.11](#) alle B-Splines außer $B_{i,d,\tau}$ mit $i = j - d, \dots, j$ konstant Null sind. Dieses Wissen kann genutzt werden, um in der Rekursionsformel (3.15) lediglich die notwendigen Terme zu berechnen: Terme, die ohnehin Null sind, müssen nicht berechnet werden.

In [Abbildung 3.9](#) sind die Abhängigkeiten zwischen den B-Splines, die auf dem Intervall $[\tau_j, \tau_{j+1})$ ungleich Null sind, schematisch dargestellt. Mit Pfeilen ist gekennzeichnet, wie die B-Splines in der Rekursionsformel (3.15) zusammenhängen. Durch Nullen sind die Terme dargestellt, die sicher Null sind und daher nicht in der Rekursion berücksichtigt werden müssen. Ausgehend von $B_{j,0,\tau}(x)$, dessen Wert per Definition trivialerweise Eins ist, können die Werte der B-Splines $B_{j,d,\tau}(x)$ berechnet werden.

Algorithmus 2 : De Boors erster Algorithmus

Eingabe :

- $d \in \mathbb{N}$: Grad der B-Splines,
- $\tau \in \mathbb{R}^{l+1}$: Knotenfolge,
- $c \in \mathbb{R}^{l-d}$: Koeffizienten der B-Spline Basisdarstellung,
- $x \in \mathbb{R}$: Stelle, an der die B-Spline Basisdarstellung ausgewertet wird

Ausgabe : $r_j^{[d]}(x) = \sum_{i=j-d}^j c_i B_{i,d,\tau}(x)$

Initialisierung :

- Wähle Index $j \in \mathbb{N}$, sodass $x \in [\tau_j, \tau_{j+1})$ ist,
- Setze $r = (c_{j-d}, \dots, c_j) \in \mathbb{R}^{d+1}$

for $k = 1, \dots, d$:

- for** $s = 0, \dots, d - k$:
- $\gamma_s = \frac{\tau_{j+s+1} - x}{\tau_{j+s+1} - \tau_{j-d+k+s}}$
- $r_s = (1 - \gamma_s) r_{s+1} + \gamma_s r_s$
- end**

end

return r_0

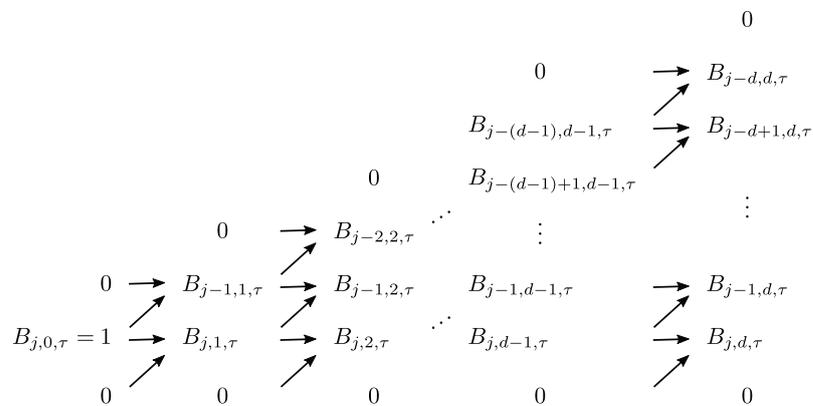


Abbildung 3.9: Abhängigkeiten der B-Splines mit Werten ungleich Null auf dem Intervall $[\tau_j, \tau_{j+1})$

Ausgabe von de Boors zweitem Algorithmus sind die Werte aller B-Splines vom Grad d , welche auf dem Intervall $[\tau_j, \tau_{j+1})$ ungleich Null sind. Für gegebene Koeffizienten c_i kann aus diesen der Wert von $f(x)$ aus der Basisdarstellung (3.25) bestimmt werden.

Bei der Herleitung des Algorithmus wird ausgenutzt, dass einige Terme mehrfach auftreten. Eine doppelte Berechnung dieser Terme wird vermieden und dadurch die Effizienz des Algorithmus gesteigert. Darüber hinaus wird durch geschicktes Aus-

führen der Rekursion lediglich ein Vektor zum Speichern der Werte der B-Splines benötigt. Eine ausführliche Herleitung von de Boors zweitem Algorithmus ist in [Anhang A.2](#) ausgeführt. In [Algorithmus 3](#) ist der Algorithmus zusammengefasst.

Algorithmus 3 : De Boors zweiter Algorithmus

Eingabe :

$d \in \mathbb{N}$: Grad der B-Splines,

$\tau \in \mathbb{R}^{l+1}$: Knotenfolge,

$x \in \mathbb{R}$: Stelle, an der die B-Splines ausgewertet werden

Ausgabe : Werte der B-Splines mit $B_{i,d,\tau}(x) \neq 0$ ($i = j - d, \dots, j$)

Initialisierung :

Wähle Index j , sodass $x \in [\tau_j, \tau_{j+1})$ ist,

Setze $b = (0, \dots, 0) \in \mathbb{R}^{d+1}$

for $s = 1, \dots, d$:

for $r = s, \dots, 1$:

$$\tau_{r-1}^R = \tau_{j+r}$$

$$\tau_{r-1}^L = \tau_{j-s+r}$$

$$w_{r-1} = \frac{b_{r-1}}{\tau_{r-1}^R - \tau_{r-1}^L}$$

$$b_r += (x - \tau_{r-1}^L)w_{r-1}$$

$$b_{r-1} = (\tau_{r-1}^R - x)w_{r-1}$$

end

end

return b

Kapitel 4

Grundlagen der Parameteridentifikation

In den verschiedensten Wissenschaftsgebieten werden mathematische Modelle verwendet, um Prozesse zu beschreiben. Typische Anwendungen finden sich in den Natur- und Ingenieurwissenschaften. Dort wird beispielsweise die Bewegung von Fahrzeugen modelliert, um autonome Fahrmanöver durchzuführen [59]. Die Anwendungsbereiche mathematischer Modelle reichen jedoch deutlich weiter. So werden etwa in den Wirtschaftswissenschaften mathematische Modelle herangezogen, um das ökonomische Wachstum von Volkswirtschaften zu charakterisieren [33]. In der Medizin werden mathematische Modelle entwickelt, um die Ausbreitung von Pandemien [81] zu beschreiben. Solche Modelle können genutzt werden, um Strategien zur Eindämmung des Infektionsgeschehens [42, 84] sowie optimale Impfstrategien der Bevölkerung zu entwickeln [21].

Bei der mathematischen Modellierung werden die Prozesse auf die wesentlichen Effekte reduziert und Gleichungen hergeleitet, welche diese Effekte beschreiben. Mathematische Modelle beschreiben daher meist eine vereinfachte Form der Realität. Um dennoch möglichst viele Situationen beschreiben zu können, sind die Modelle typischerweise von Parametern abhängig. In einigen Fällen können die Parameter direkt gemessen werden, oft ist dies jedoch nicht möglich. Stattdessen werden Messdaten des Prozesses aufgenommen, welcher modelliert werden soll. Die Aufgabe der Parameteridentifikation besteht darin, die unbekannt Parameter so zu bestimmen, dass das mathematische Modell die gemessenen Daten möglichst gut beschreibt.

In diesem Kapitel werden explizite und dynamische Parameteridentifikationsprobleme vorgestellt. Der Schwerpunkt liegt dabei auf dynamischen Parameteridentifikationsproblemen. Es werden die Herausforderungen der Problemklassen herausgearbeitet und verschiedene Lösungsansätze diskutiert.

4.1 Explizite Parameteridentifikationsprobleme

Besonders einfache mathematische Modelle lassen sich explizit durch Funktionen beschreiben. Darüber hinaus können die Parameter der Modelle Beschränkungen unterliegen. Im Folgenden werden solche Modelle als explizite parameterabhängige Modelle bezeichnet.

Definition 4.1 (Explizites parameterabhängiges Modell). *Seien $n_\psi, n_\theta \in \mathbb{N}$, $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$ und $t \in \mathbb{R}$. Die Funktion $\psi : \mathbb{R} \times \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_\psi}$, $(t, \theta) \mapsto \psi(t, \theta)$ heißt explizites parameterabhängiges Modell. Die Variable t wird Zeit genannt. Der Vektor θ wird als Parametervektor und seine Einträge als Modellparameter bezeichnet. Die Menge Θ beschreibt die zulässigen Werte der Modellparameter.*

Die Definitionen von n_ψ, n_θ sowie ψ , θ und Θ werden im Folgenden als gegeben vorausgesetzt.

Die Aufgabe der Parameteridentifikation besteht darin, die Modellparameter des Modells zu identifizieren, sodass gegebene Daten bestmöglich durch das Modell approximiert werden. Dies wird in der folgenden Definition präzisiert.

Definition 4.2 (Explizites Parameteridentifikationsproblem). *Sei $n_t \in \mathbb{N}$. Gegeben sei eine streng monoton wachsende Folge von diskreten Zeitpunkten $(t_i)_{i=1, \dots, n_t} \in \mathbb{R}^{n_t}$ mit $t_1 \geq t_0$ und $t_{n_t} \leq t_f$. Darüber hinaus seien diesen Zeitpunkten zugeordnete Datenpunkte $\bar{y}_i \in \mathbb{R}^{n_\psi}$ für $i = 1, \dots, n_t$ gegeben. Dann heißt das Optimierungsproblem*

$$\min_{\theta \in \Theta} F(\theta) := \sum_{i=1}^{n_t} \|\bar{y}_i - \psi(t_i, \theta)\|_2^2 \quad (\text{EPIP})$$

explizites Parameteridentifikationsproblem.

Es ist möglich, mithilfe sogenannter Gewichte in der Zielfunktion F von (EPIP) unterschiedliche Größenordnungen der Daten zu berücksichtigen und einzelne Datenpunkte stärker zu gewichten [63, S. 120]. Im Folgenden wird darauf jedoch verzichtet. Hat die zulässige Menge der Modellparameter die Form

$$\Theta = \{\theta \in \mathbb{R}^{n_\theta} \mid g(\theta) \leq 0 \text{ und } h(\theta) = 0\}$$

mit stetig differenzierbaren Funktionen $g : \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_g}$ und $h : \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_h}$, so ist (EPIP) ein nichtlineares Optimierungsproblem gemäß Definition 2.1. In (EPIP) wird die quadrierte Abweichung zwischen den Datenpunkten und der Ausgabe des Modells minimiert. Es handelt sich deshalb um ein sogenanntes *Least-Squares-Problem*. Für diese spezielle Klasse nichtlinearer Optimierungsprobleme wurden bereits viele effiziente Lösungsalgorithmen entwickelt, über welche im folgenden Abschnitt ein Überblick gegeben wird.

4.1.1 Überblick über numerische Lösungsverfahren von Least-Squares-Problemen

In diesem Abschnitt werden einige wichtige Resultate zu numerischen Lösungsverfahren von Least-Squares-Problemen zusammengefasst. Für einen umfassenden Überblick über bekannte Methoden sei auf Björck [5] verwiesen.

Sind die Funktionen ψ_j linear und gibt es keine Nebenbedingungen, d. h. $\Theta = \mathbb{R}^{n_\theta}$, so kann die Lösung von (EPIP) direkt berechnet werden. Dies kann zum Beispiel mithilfe einer QR-Zerlegung geschehen, welche effizient über Householdertransformationen berechnet werden kann [10]. Durch eine Modifizierung der QR-Zerlegung kann der Ansatz auch auf lineare Least-Squares-Probleme mit linearen Nebenbedingungen erweitert werden [27]. Darüber hinaus existieren verschiedene Algorithmen, welche dünnbesetzte Strukturen in diesen Problemen ausnutzen [4, 19, 29].

Ein einfaches Verfahren zum Lösen von Least-Squares-Problemen mit nichtlinearen Funktionen ψ_j ohne Nebenbedingungen ist das Gauß-Newton-Verfahren [47, S. 254 ff.]. Dabei handelt es sich um ein iteratives Verfahren. Die grundlegende Idee ist es, die Zielfunktion mithilfe einer Taylor-Entwicklung erster Ordnung lokal um den Wert der aktuellen Iteration zu linearisieren. Durch Lösen des linearisierten Problems wird eine Suchrichtung zum Bestimmen des Wertes der nächsten Iteration berechnet. Ein weiteres weit verbreitetes Verfahren zum Lösen nichtlinearer Least-Squares-Probleme ohne Nebenbedingungen ist der Levenberg-Marquardt-Algorithmus. Er wurde bereits 1944 von Levenberg [40] entwickelt und 1963 von Marquardt [45] wiederentdeckt. Der Levenberg-Marquardt-Algorithmus verbindet das Gauß-Newton-Verfahren mit einem Gradientenverfahren. Führt die Suchrichtung des Gauss-Newton-Verfahrens zu keiner wesentlichen Verbesserung im Zielfunktionswert, so wird als Suchrichtung die Richtung des steilsten Abstiegs stärker gewichtet.

Methoden zum Lösen des allgemeinen nichtlinearen Least-Squares-Problems (EPIP) wurden unter anderem von Mahdavi-Amiri et al. [43, 44] entwickelt. Schittkowski [65] schlägt vor, (EPIP) mittels SQP-Verfahren (vgl. Abschnitt 2.4.1) zu lösen, um von bereits vorhandenen Algorithmen zu profitieren. Er gibt jedoch zu bedenken, dass Least-Squares-Probleme häufig schlecht konditioniert sind. Um sie dennoch effektiv mit SQP-Verfahren zu lösen, empfiehlt Schittkowski [65] deshalb eine Transformation von (EPIP).

4.1.2 Alternative Zielfunktionen

In (EPIP) wird die quadrierte Abweichung zwischen den Datenpunkten und der Ausgabe des Modells minimiert. Es kann gezeigt werden, dass dies die natürliche Wahl ist, sollten die Daten einem normalverteilten Rauschen unterliegen [2]. Es

ist jedoch auch möglich, andere Zielfunktionen zu betrachten. Beispielfhaft seien an dieser Stelle

$$F(\theta) = \sum_{i=1}^{n_t} \sum_{j=1}^{n_\psi} \omega_{i,j} \left| (\bar{y}_i)_j - \psi_j(t_i, \theta) \right| \quad (4.1)$$

und

$$F(\theta) = \max_{i,j} \omega_{i,j} \left| (\bar{y}_i)_j - \psi_j(t_i, \theta) \right| \quad (4.2)$$

genannt. Beide Zielfunktionen sind nicht differenzierbar. Deshalb kann die in [Kapitel 2](#) vorgestellte Theorie zur nichtlinearen Optimierung nicht ohne Weiteres angewendet werden. Es ist jedoch durch das Einführen von weiteren Optimierungsvariablen und Nebenbedingungen möglich, die Zielfunktionen (4.1) und (4.2) in differenzierbare Funktionen umzuformulieren. Für eine Darstellung der Umformulierung sei auf Schittkowski [63, S. 35] verwiesen.

4.2 Dynamische Parameteridentifikationsprobleme

Viele mathematische Modelle beschreiben die zeitliche Änderung eines Zustands mithilfe gewöhnlicher Differentialgleichungen. Ein solches zeitabhängiges Modell wird im weiteren Verlauf der Arbeit als dynamisches Modell bezeichnet.

Definition 4.3 (Parameterabhängiges dynamisches Modell). *Seien $n_x, n_u, n_p \in \mathbb{N}$, $[t_0, t_f] \subset \mathbb{R}$ und $p \in \mathcal{P} \subset \mathbb{R}^{n_p}$. Die Funktion $f : [t_0, t_f] \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_x}$ sei stetig und bezüglich des zweiten und dritten Arguments stetig partiell differenzierbar. Die Differentialgleichung*

$$\dot{x}(t) = f(t, x(t), u(t), p), \quad t_0 \leq t \leq t_f \quad (4.3)$$

wird parameterabhängiges dynamisches Modell *genannt*.

Die Größe $x(t) = (x_1(t), \dots, x_{n_x}(t))^T \in \mathbb{R}^{n_x}$ heißt Zustandsvektor zum Zeitpunkt t . Durch den Steuervektor $u(t) = (u_1(t), \dots, u_{n_u}(t))^T \in \mathbb{R}^{n_u}$ kann das Modell beeinflusst werden. Die Einträge des Zustands- und Steuervektors werden als Zustände bzw. Steuerungen bezeichnet. Der Vektor p wird Parametervektor und seine Einträge Modellparameter genannt. Die Menge \mathcal{P} beschreibt die zulässigen Werte der Modellparameter.

Die Definitionen von n_x, n_u, n_p, t_0, t_f sowie f, p und \mathcal{P} werden im Folgenden als gegeben vorausgesetzt. Die Funktion f sei stetig und bezüglich des zweiten und dritten Arguments stetig partiell differenzierbar. Es wird davon ausgegangen, dass das dynamische Modell keinen Unsicherheiten unterliegt.

Die Aufgabe der Parameteridentifikation ist folgende: Gegeben seien Daten für die Zustände und die Steuerungen. Gesucht sind die Modellparameter, welche das Modell so beeinflussen, dass es die gemessenen Zustände bestmöglich approximiert. Dies wird in [Definition 4.4](#) präzisiert.

Definition 4.4 (Dynamisches Parameteridentifikationsproblem). *Seien $n_t, n_{\bar{x}} \in \mathbb{N}$ mit $n_{\bar{x}} \leq n_x$. Gegeben sei eine streng monoton wachsende Folge von diskreten Zeitpunkten $(t_i)_{i=1, \dots, n_t} \in \mathbb{R}^{n_t}$ mit $t_1 \geq t_0$ und $t_{n_t} \leq t_f$. Darüber hinaus seien diesen Zeitpunkten zugeordnete Daten für die Zustände $\bar{x}_i \in \mathbb{R}^{n_{\bar{x}}}$ und die Steuerungen $\bar{u}_i \in \mathbb{R}^{n_u}$ für $i = 1, \dots, n_t$ gegeben. Die Funktion $u_{\text{app}} : [t_0, t_f] \rightarrow \mathbb{R}^{n_u}$ sei eine geeignete Approximation der Daten der Steuerungen. Die Zuordnung zwischen den Indizes der gemessenen Zustände und den Indizes der Zustände sei durch eine injektive Abbildung $\sigma : \{1, \dots, n_{\bar{x}}\} \rightarrow \{1, \dots, n_x\}$ gegeben. Der Vektor $x_\sigma(t) := (x_{\sigma(1)}(t), \dots, x_{\sigma(n_{\bar{x}})}(t))^\top \in \mathbb{R}^{n_{\bar{x}}}$ bezeichne den Vektor der gemessenen Zustände zum Zeitpunkt t . Dann heißt das Optimierungsproblem*

$$\begin{aligned} \min_{x_0 \in \mathbb{R}^{n_x}, p \in \mathcal{P}} F(p, x_0) &:= \sum_{k=1}^{n_t} \|\bar{x}_k - x_\sigma(t_k)\|_2^2 \\ \text{unter } \dot{x}(t) &= f(t, x(t), u_{\text{app}}(t), p), \quad t_0 \leq t \leq t_f, \\ x(t_0) &= x_0 \end{aligned} \tag{DPIP}$$

dynamisches Parameteridentifikationsproblem zum Anfangswert $x_0 \in \mathbb{R}^{n_x}$.

Wie beim expliziten Parameteridentifikationsproblem ([EPIP](#)) ist es möglich, in der Zielfunktion von ([DPIP](#)) mithilfe von Gewichten unterschiedliche Größenordnungen der Zustände zu berücksichtigen und einzelne Datenpunkte stärker zu gewichten. Darauf wird jedoch im weiteren Verlauf verzichtet.

Zum Auswerten des dynamischen Modells muss der Steuervektor zu beliebigen Zeitpunkten bekannt sein. Typischerweise sind dagegen lediglich Daten der Steuerungen zu diskreten Zeitpunkten gegeben. Die Funktion u_{app} ist eine Approximation dieser Daten und dient zum Erzeugen des Steuervektors zu beliebigen Zeitpunkten. Im einfachsten Fall ist u_{app} eine stückweise konstante oder lineare Approximation der Daten der Steuerungen. Der Ansatz lässt sich verallgemeinern, indem ein allgemeines ν -reguläres stückweises Polynom zur Approximation verwendet wird. Dieser Ansatz wird im weiteren Verlauf der Arbeit näher untersucht und deshalb an dieser Stelle nicht weiter vertieft (vgl. [Abschnitt 4.2.2.1](#)).

Im Folgenden wird davon ausgegangen, dass die Menge der zulässigen Werte der Modellparameter von der Form

$$\mathcal{P} = \{p \in \mathbb{R}^{n_p} \mid g(p) \leq 0 \text{ und } h(p) = 0\} \tag{4.4}$$

mit stetig differenzierbaren Funktionen $g : \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_g}$ und $h : \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_h}$ ist.

Dies ermöglicht das Lösen von (DPIP) mithilfe von Methoden der nichtlinearen Optimierung. Die Darstellung (4.4) schließt insbesondere den Fall ein, dass für die Modellparameter Box-Beschränkungen der Form $p_l \leq p \leq p_u$ mit $p_l, p_u \in \mathbb{R}^{n_p}$ vorliegen.

Analog zum expliziten Parameteridentifikationsproblem ist die Minimierung der quadrierten Abweichung zwischen den Datenpunkten und der Ausgabe des Modells die natürliche Wahl, sollten die Daten einem normalverteilten Rauschen unterliegen [2]. Es ist auch möglich, wie in Abschnitt 4.1.2 diskutiert, andere Zielfunktionen zu betrachten. Im folgenden Abschnitt wird ein Überblick über numerische Lösungsverfahren von dynamischen Parameteridentifikationsproblemen gegeben.

4.2.1 Überblick über numerische Lösungsverfahren von dynamischen Parameteridentifikationsproblemen

Eine wesentliche Herausforderung beim Lösen von (DPIP) besteht darin, dass für die Auswertung der Zielfunktion das Anfangswertproblem

$$\begin{aligned} \dot{x}(t) &= f(t, x(t), u_{\text{app}}(t), p), & t_0 \leq t \leq t_f, \\ x(t_0) &= x_0 \end{aligned} \tag{4.5}$$

gelöst werden muss. Im Allgemeinen kann die Lösung von (4.5) nicht analytisch bestimmt werden und muss deshalb numerisch approximiert werden. Zu gegebenem Anfangswert x_0 und Modellparametern p kann das Anfangswertproblem (4.5) mit numerischen Verfahren zum Lösen gewöhnlicher Differentialgleichungen gelöst werden. Für eine ausführliche Übersicht über die Thematik sei auf Deuffhard und Bornemann [15] verwiesen.

Ein naiver Lösungsansatz für (DPIP) ist das sogenannte Schießverfahren. Sofern nicht anders angegeben, beruht dieser Absatz auf Schittkowski [63, S. 128 ff.]. Beim Schießverfahren wird (DPIP) mithilfe eines iterativen Optimierungsverfahrens gelöst. Für jede Auswertung der Zielfunktion wird das Anfangswertproblem (4.5) numerisch gelöst. Dies kann zum Beispiel mittels eines Einschrittverfahrens [15] geschehen. Der Lösungsansatz mag auf den ersten Blick intuitiv sein, hat jedoch mehrere Nachteile. Zum einen ist die Zielfunktion des formulierten Optimierungsproblems stark nichtlinear. Dadurch besitzt das Optimierungsproblem möglicherweise viele lokale Minima und ist nicht konvex. Infolgedessen ist es schwierig, eine gute Lösung von (DPIP) zu finden. Zum anderen hängt die Lösung des Anfangswertproblems (4.5) oft empfindlich vom Anfangswert und den Modellparametern ab. Außerdem kommt es beim numerischen Lösen des Anfangswertproblems unweigerlich zu Fehlern, welche den Einfluss einer schlechten Schätzung für den Anfangswert und die Modellparameter noch weiter verstärken. Dieser Effekt kommt insbesondere dann zu tragen,

wenn (4.5) über einen langen Zeitraum gelöst werden muss. Deshalb ist es wichtig, gute Startschätzungen für den Anfangswert und die Optimierungsvariablen anzugeben. Eine passable Startschätzung für den Anfangswert kann möglicherweise aus den Daten für die Zustände bestimmt werden. Für die Modellparameter ist es dagegen mitunter ein Problem, eine gute Startschätzung zu bestimmen. Selbst bei einer scheinbar guten Startschätzung kann der Fehler zwischen der initialen und der optimalen Lösung groß sein, wenn es sich um eine instabile Differentialgleichung handelt. Darüber hinaus kann es vorkommen, dass das Anfangswertproblem (4.5) aufgrund von Singularitäten oder numerischen Instabilitäten nicht auf dem gesamten Zeitintervall gelöst werden kann. Zumindest letztere Probleme können verringert werden, wenn zum Lösen des Anfangswertproblems (4.5) ein Mehrfach-Schießverfahren verwendet wird. Für Details zu diesem Ansatz sei auf Bock et al. [6] und Voss et al. [80] verwiesen.

Der Ansatz der vollen Diskretisierung verfolgt eine andere Herangehensweise, um (DPIP) zu lösen. Nach dem Vorbild der direkten Verfahren zum Lösen von Optimalsteuerungsproblemen wird (DPIP) mithilfe einer Diskretisierung in ein nichtlineares Optimierungsproblem überführt [3]. Das Vorgehen ist dabei wie folgt: Anstatt eines kontinuierlichen Zeitintervalls werden diskrete Zeitpunkte betrachtet. Für jeden diskreten Zeitpunkt und Zustand wird eine Optimierungsvariable eingeführt, welche den Wert des Zustands an dem Zeitpunkt beschreibt. Die Lösung des Anfangswertproblems wird an den diskreten Zeitpunkten mithilfe eines geeigneten Diskretisierungsverfahrens approximiert. Eine Möglichkeit, das Anfangswertproblem (4.5) zu diskretisieren, ist mittels impliziter Einschrittverfahren [15, S. 128 ff.]. Dieses Vorgehen führt auf ein im Allgemeinen nichtlineares Gleichungssystem, welches von einer Lösung des diskretisierten Problems erfüllt sein muss. Das Gleichungssystem wird als Nebenbedingung in das Optimierungsproblem aufgenommen. Eine Lösung des Optimierungsproblems ist folglich auch eine Lösung des diskretisierten Anfangswertproblems.

Der Lösungsansatz der vollen Diskretisierung führt auf ein hochdimensionales Optimierungsproblem mit einer großen Anzahl an Nebenbedingungen. Auf den ersten Blick scheint dieser Ansatz komplizierter, als (DPIP) mittels Schießverfahren zu lösen. Der Ansatz der vollen Diskretisierung hat jedoch auch Vorteile. Beim direkten Lösen von (DPIP) muss in jeder Iteration das Anfangswertproblem (4.5) gelöst werden, um die Zielfunktion auswerten zu können. Die damit einhergehenden Probleme wurden zuvor bereits diskutiert. Beim Ansatz der vollen Diskretisierung ist dies nicht nötig, da die Lösung des Anfangswertproblems implizit während des Lösungsprozesses des Optimierungsproblems bestimmt wird. Zur Lösung des formulierten Optimierungsproblems kann ein SQP-Verfahren verwendet werden (vgl. Abschnitt 2.4). Für eine möglichst effiziente Implementierung kann die spezielle Struktur des Diskretisierungsverfahrens für das Anfangswertproblem ausgenutzt werden. Es stellt sich heraus, dass die Jacobi-Matrix der Nebenbedingungen sowie die Hesse-Matrix der

Lagrange-Funktion dünn besetzt sind [16]. Dies ermöglicht ein effizientes Lösen des formulierten Optimierungsproblems sowohl in Hinblick auf die Laufzeit als auch den Speicherbedarf des Algorithmus.

Für den Ansatz der vollen Diskretisierung ist es auch möglich, ein anderes Diskretisierungsverfahren zur Approximation der Lösung des Anfangswertproblems (4.5) zu verwenden. So nutzen Tsang [75] sowie Tjoa und Biegler [74] ein Kollokationsverfahren zum Diskretisieren des Anfangswertproblems [15, S. 268 ff.]. Während Tsang [75] Polynome als Ansatzfunktionen für das Kollokationsverfahren verwendet, nutzen Tjoa und Biegler [74] Lagrange-Polynome.

Ein weiteres Lösungsverfahren von (DPIP), das keine Methoden der nichtlinearen Optimierung verwendet, ist der Kalman-Filter [35]. Dabei handelt es sich um einen sogenannten Zustandsschätzer, welcher iterativ die Modellparameter und Zustände identifiziert¹. Kalman-Filter beruhen auf statistischen Überlegungen und können auch Unsicherheiten im Modell berücksichtigen [72]. In seiner ursprünglichen Form kann der Kalman-Filter lediglich auf lineare und zeitlich diskrete Modelle angewendet werden. Er lässt sich aber auf nichtlineare und kontinuierliche Probleme, wie (DPIP) erweitern. Es wird dann vom erweiterten kontinuierlichen Kalman-Filter (engl.: *continuous-time extended Kalman filter*) gesprochen [71]. Da in dieser Arbeit schwerpunktmäßig Methoden der nichtlinearen Optimierung thematisiert werden, wird auf eine ausführliche Darstellung der Theorie zum Kalman-Filter verzichtet. Für weitere Informationen zu dem Thema sei auf Simon [72] verwiesen.

Im Folgenden wird ein alternativer Ansatz zum Lösen von (DPIP) vorgestellt: die sogenannte Kollokationsmethode.

4.2.2 Kollokationsmethode

Kollokationsmethoden² verfolgen einen anderen Lösungsansatz für (DPIP). Die von Swartz et al. [73] und Varah [76] vorgestellte Methode besteht aus zwei Schritten. Zunächst werden die Daten der Zustände \bar{x} durch eine Funktion x_{app} approximiert. Mithilfe dieser Approximation wird schließlich im zweiten Schritt das Optimierungsproblem

$$\min_{p \in \mathcal{P}} \frac{1}{n} \sum_{i=1}^n \left\| \dot{x}_{\text{app}}(\tilde{t}_i) - f(\tilde{t}_i, x_{\text{app}}(\tilde{t}_i), u_{\text{app}}(\tilde{t}_i), p) \right\|_2^2, \quad (4.6)$$

mit $n \in \mathbb{N}$ und $\tilde{t}_i \in [t_0, t_f]$ gelöst, um die Modellparameter zu identifizieren. Eine ausführliche Herleitung von (4.6) ist in Abschnitt 4.2.2.2 gegeben. Um das Vektorfeld

¹Um die Methode anwenden zu können, wird der Zustandsvektor um die Modellparameter erweitert.

²In der Literatur findet man die Methoden auch unter dem Namen *Principal Differential Analysis (PDA)*.

f in (4.6) auswerten zu können, müssen im Allgemeinen die Werte aller Zustände bekannt sein. Aus diesem Grund müssen für jeden Zustand Daten vorliegen. Im weiteren Verlauf wird deshalb davon ausgegangen, dass $n_x = n_{\bar{x}}$ und $x_\sigma = x$ ist.

Der Ansatz der Kollokationsmethode umgeht das Lösen des Anfangswertproblems (4.5) und die damit einhergehenden zuvor diskutierten Schwierigkeiten. Wie gut die Modellparameter durch die Kollokationsmethode identifiziert werden, ist allerdings stark davon abhängig, wie gut x_{app} die Daten der Zustände und insbesondere \dot{x}_{app} deren Ableitung approximiert [76].

In den folgenden Abschnitten werden die beiden Schritte der Kollokationsmethode, Approximation der Daten und Identifikation der Modellparameter, näher erläutert.

4.2.2.1 Approximation der Daten

Ursprünglich wurden für x_{app} Polynomfunktionen verwendet [73]. Es können jedoch auch andere Ansatzfunktionen, wie zum Beispiel Fourierreihen oder Wavelets gewählt werden [52]. Besonders bewährt haben sich die flexiblen ν -regulären stückweisen Polynome (vgl. Kapitel 3) [52]. Sie haben mehrere Vorteile: Zum einen sind ν -reguläre stückweise Polynome flexibel genug, um auch komplizierte Verläufe in den Daten abbilden zu können. Zum anderen kann über den Vektor ν beeinflusst werden, welche Regularität an den Sprungstellen gegeben sein soll. Auf diese Weise kann Wissen über die Regularität der Zustände in die Approximation einfließen.

Ist das Vektorfeld f des dynamischen Modells (4.3) n -Mal stetig differenzierbar bezüglich t , so folgt trivialerweise, dass jede Lösung von (4.3) $(n + 1)$ -Mal stetig differenzierbar ist. Dieses Wissen kann verwendet werden, um die Regularität der Approximation an den Sprungstellen festzulegen. Unstetige Steuerungen führen im Allgemeinen auf ein unstetiges Vektorfeld f im dynamischen Modell (4.3). Dies ist durchaus üblich und entspricht oftmals einer optimalen Steuerung (z. B. Bang-Bang Steuerung [41, S. 234]). Das dynamische Modell (4.3) kann in diesem Fall nicht mehr im klassischen Sinne gelöst werden. Es können jedoch schwache Lösungen formuliert werden, welche nicht überall differenzierbar sind. Die Lösungen weisen möglicherweise „Knicke“ an Stellen auf, an denen das Vektorfeld f unstetig ist. Durch ν -reguläre stückweise Polynome können auch diese „Knicke“ abgebildet werden.

Um möglichst variabel zu sein, werden sowohl der Grad, als auch die Sprungstellen und Regularitätsbedingungen an den Sprungstellen für jeden Zustand separat gewählt. Werden für den j -ten Zustand B-Splines vom Grad d_j und mit der Knotenfolge $\tau_j \in \mathbb{R}^{l_j+1}$ als Basisfunktionen gewählt, so hat x_{app} die Gestalt

$$x_{\text{app},j}(t; c_j) = \sum_{i=0}^{l_j-d_j-1} c_{j,i} B_{i,d_j,\tau_j}(t) \quad \text{für } j = 1, \dots, n_x \quad (4.7)$$

mit geeigneten Koeffizienten $c_{j,i} \in \mathbb{R}$ (vgl. Abschnitt 3.3). Sei $N_j := l_j - d_j$. Die

Koeffizienten der Approximationen $x_{\text{app},j}$ werden so gewählt, dass der Fehler zwischen den Daten und den Approximationen minimal ist. Übliche Fehlermaße sind die quadrierte Abweichung, der Absolutbetrag oder die negative Log-Likelihood-Funktion [55]. Die folgenden Untersuchungen konzentrieren sich auf den Fall der quadrierten Abweichung, wie ursprünglich von Varah [76] vorgeschlagen wurde. Da die Koeffizienten der Approximationen unterschiedlicher Zustände unabhängig voneinander sind, können die Optimierungsprobleme

$$\min_{c_j \in \mathbb{R}^{N_j}} \sum_{k=1}^{n_t} \left((\bar{x}_k)_j - x_{\text{app},j}(t_k; c_j) \right)^2 \quad \text{für } j = 1, \dots, n_x \quad (4.8)$$

formuliert werden. Es wird im weiteren Verlauf davon ausgegangen, dass keine Nebenbedingungen an die Koeffizienten c_j gestellt werden. In der Praxis kann dies durchaus der Fall sein, sollten Informationen über bestimmte Eigenschaften der Daten bekannt sein. Bei (4.8) handelt es sich um explizite Parameteridentifikationsprobleme gemäß Definition 4.2. Sie können mittels der in Abschnitt 4.1 diskutierten Methoden gelöst werden. Da die Funktionen $x_{\text{app},j}$ gemäß ihrer Definition (4.7) linear in den Parametern $c_{j,i}$ sind, kann das Lösen von (4.8) äußerst effizient geschehen.

An dieser Stelle wird explizit auf die Vorzüge von B-Splines als Basisfunktionen der Approximationen eingegangen. Einsetzen von (4.7) in (4.8) führt auf

$$\min_{c_j \in \mathbb{R}^{N_j}} \sum_{k=1}^{n_t} \left((\bar{x}_k)_j - \sum_{i=0}^{l_j-d_j-1} c_{j,i} B_{i,d_j,\tau_j}(t_k) \right)^2 \quad \text{für } j = 1, \dots, n_x. \quad (4.9)$$

Durch Definition von $\hat{x}^j := \left((\bar{x}_1)_j, \dots, (\bar{x}_{n_t})_j \right) \in \mathbb{R}^{n_t}$ und $A^j \in \mathbb{R}^{n_t \times N}$ mit $A_{k,i}^j := B_{i,d_j,\tau_j}(t_k)$ kann (4.9) zu

$$\min_{c_j \in \mathbb{R}^{N_j}} \left\| \hat{x}^j - A^j c_j \right\|_2^2 \quad \text{für } j = 1, \dots, n_x \quad (4.10)$$

umgeformt werden. Weil B-Splines kompakte Träger haben und lediglich $d+1$ Basisfunktionen in einem t_k ungleich Null sind, ist die Matrix A^j dünnbesetzt. Dies kann ausgenutzt werden, um in numerischen Verfahren Speicherplatz zu sparen. Algorithmen können die dünnbesetzte Struktur ausnutzen, um (4.10) effizient zu lösen [29, 19]. Die Einträge einer Zeile von A^j können effizient mithilfe von de Boors zweitem Algorithmus (Alg. 3) berechnet werden.

Abschließend sei darauf hingewiesen, dass das in diesem Abschnitt diskutierte Vorgehen in analoger Weise genutzt werden kann, um eine geeignete Approximation u_{app} der Daten der Steuerungen \bar{u} zu bestimmen. Insbesondere ist es durch den Ansatz mit ν -regulären stückweisen Polynomen möglich, Sprünge in den Steuerungen, wie sie beispielsweise bei Bang-Bang Steuerungen auftreten, abzubilden.

Im nächsten Schritt der Kollokationsmethode werden die Modellparameter des dynamischen Modells identifiziert. Dazu werden die Approximationen $x_{\text{app},j}$ genutzt.

4.2.2.2 Identifikation der Modellparameter

Um im zweiten Schritt der Kollokationsmethode die Modellparameter p des dynamischen Modells zu identifizieren, wird der Fehler zwischen dem Vektorfeld f und der Ableitung von x_{app} minimiert [76]. Der Fehler hängt von den Koeffizienten c_j der Approximation sowie den Modellparametern p ab. Dabei sind die Koeffizienten c_j aus dem ersten Schritt der Kollokationsmethode bekannt. Für den j -ten Zustand sei der Fehler durch

$$E_j(c_j, p) := \int_{t_0}^{t_f} (\dot{x}_{\text{app},j}(t; c_j) - f_j(t, x_{\text{app}}(t; c_j), u_{\text{app}}(t), p))^2 dt \quad (4.11)$$

definiert. Um das Lösen des Integrals zu vermeiden, schlägt Varah [76] vor, statt E_j einen diskreten Fehler zu betrachten. Sei dazu $n \in \mathbb{N}$ und $(\tilde{t}_i)_{i=1, \dots, n} \in \mathbb{R}^n$ mit $\tilde{t}_1 \geq t_0$ und $\tilde{t}_n \leq t_f$ eine streng monoton wachsende Folge von diskreten Zeitpunkten. Der diskrete Fehler ist definiert durch

$$e_j(c_j, p) := \frac{1}{n} \sum_{i=1}^n (\dot{x}_{\text{app},j}(\tilde{t}_i; c_j) - f_j(\tilde{t}_i, x_{\text{app}}(\tilde{t}_i; c_j), u_{\text{app}}(\tilde{t}_i), p))^2.$$

Um die Modellparameter p zu identifizieren, werden schließlich die diskreten Fehler e_j minimiert. Es ergibt sich das Optimierungsproblem

$$\min_{p \in \mathcal{P}} \sum_{j=1}^{n_x} e_j(c_j, p) = \frac{1}{n} \sum_{i=1}^n \|\dot{x}_{\text{app}}(\tilde{t}_i; c) - f(\tilde{t}_i, x_{\text{app}}(\tilde{t}_i; c), u_{\text{app}}(\tilde{t}_i), p)\|_2^2. \quad (4.12)$$

Dabei ist $c := (c_1, \dots, c_{n_x})$ der Vektor aller Koeffizienten von x_{app} . Wird $n = n_t$ und $\tilde{t}_i = t_i$ gewählt, so können die Daten der Steuerungen direkt in der Auswertung von f verwendet werden und müssen nicht durch u_{app} approximiert werden. Mit ν -regulären stückweisen Polynomen als Ansatzfunktionen für x_{app} lässt sich die Ableitung \dot{x}_{app} effizient und numerisch stabil mithilfe von [de Boors erstem Algorithmus](#) (Alg. 2) berechnen.

Bei (4.12) handelt es sich um ein explizites Parameteridentifikationsproblem gemäß [Definition 4.2](#). Nebenbedingungen sind lediglich durch die Bedingung $p \in \mathcal{P}$ gegeben. Es kann mit den in [Abschnitt 4.1](#) diskutierten Methoden gelöst werden. Im Falle einer linearen Abhängigkeit der Funktion f von p ist (4.12) ein lineares Problem und kann besonders effizient gelöst werden.

In [Abbildung 4.1](#) sind die beiden Schritte der Kollokationsmethode schematisch dargestellt. Im ersten Schritt werden aus den Daten \bar{x} die Approximationen $x_{\text{app},j}$ bestimmt. Im zweiten Schritt werden mithilfe dieser Approximationen unter Berücksichtigung des dynamischen Modells die Modellparameter identifiziert.

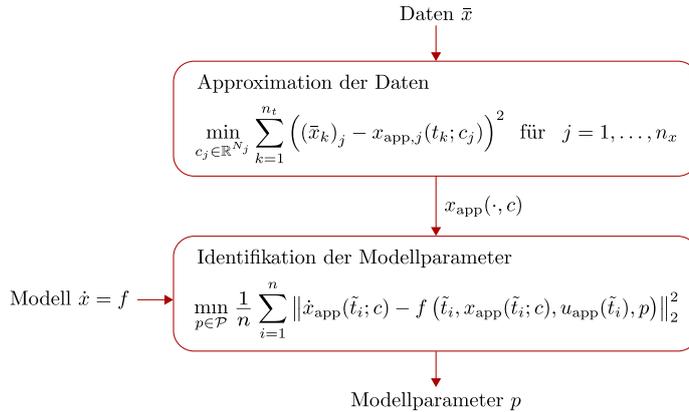


Abbildung 4.1: Schematisches Vorgehen bei der Kollokationsmethode

4.2.2.3 Vor- und Nachteile der Kollokationsmethode

In diesem Abschnitt werden Vor- und Nachteile der Kollokationsmethode diskutiert. Die Kollokationsmethode zeichnet sich dadurch aus, das Parameteridentifikationsproblem (DPIP) in mehrere Teilprobleme aufzuteilen. Jedes Teilproblem für sich ist verhältnismäßig leicht zu lösen. Um im ersten Schritt die Approximation der Daten x_{app} zu bestimmen, muss eine Reihe von linearen Least-Squares-Problemen gelöst werden. Dies kann mithilfe der in [Abschnitt 4.1.1](#) vorgestellten Algorithmen sehr effizient geschehen. Im zweiten Schritt der Kollokationsmethode werden die Modellparameter identifiziert. Dazu wird die quadrierte Abweichung zwischen dem Vektorfeld f des dynamischen Modells und der Ableitung von x_{app} minimiert. Optimierungsvariablen sind in diesem Schritt lediglich die Modellparameter. Da das Lösen des Anfangswertproblems (4.5) umgangen wird, können die Modellparameter verhältnismäßig schnell und mit wenig Aufwand identifiziert werden.

Damit die Modellparameter gut mithilfe der Kollokationsmethode identifiziert werden können, ist es essentiell, dass die Approximation x_{app} einerseits die Daten der Zustände gut repräsentiert und andererseits näherungsweise das dynamische Modell erfüllt. Oftmals sind die Daten der Zustände verrauscht. Eine Überanpassung von x_{app} an die Daten ist zu vermeiden, da dann das dynamische Modell voraussichtlich nur schlecht erfüllt wird. Eine große Herausforderung bei der Kollokationsmethode ist daher, eine gute Approximation x_{app} zu bestimmen. Ein weiterer Nachteil ist, dass für jeden Zustand Daten vorliegen müssen, um die Kollokationsmethode anwenden zu können. Dies ist in praktischen Anwendung häufig nicht der Fall.

Im nächsten Abschnitt werden verschiedene Erweiterungen der Kollokationsmethode diskutiert. Ihr Zweck besteht darin, eine Überanpassung von x_{app} an die Daten zu vermeiden und eine gute Approximation der Daten zu bestimmen. Wie sich herausstellt, kann eine der Erweiterungen auch verwendet werden, um Parameteridentifikationsprobleme zu lösen, bei denen nicht alle Zustände gemessen wurden.

4.2.2.4 Erweiterungen der Kollokationsmethode

Das in den [Abschnitten 4.2.2.1](#) und [4.2.2.2](#) vorgestellte Vorgehen wird im weiteren Verlauf als klassische Kollokationsmethode bezeichnet. In diesem Abschnitt werden Erweiterungen der klassischen Kollokationsmethode diskutiert, deren Ziel es ist, eine gute Approximation x_{app} zu bestimmen.

Werden ν -reguläre stückweise Polynome als Ansatzfunktionen gewählt und diese in der Basis der B-Splines dargestellt, so gibt es mehrere Möglichkeiten die Approximation der Daten zu beeinflussen. Zum einen kann dies über den Grad der B-Splines geschehen. Je höher der Grad gewählt wird, desto glatter ist im Allgemeinen auch die Approximation x_{app} . Jedoch ist zu beachten, dass mit zunehmendem Grad auch die Anzahl der Freiheitsgrade zunimmt. Dies kann mitunter dazu führen, dass es zur Überanpassung an die Daten kommt. Des Weiteren kann über den Grad der B-Splines Wissen über die Daten eingebracht werden. Ist etwa aus physikalischen Überlegungen bekannt, dass die Daten einem linearen Verlauf folgen, so sollten auch B-Splines vom Grad Eins gewählt werden. Zum anderen kann die Approximation durch die Anzahl und Platzierung der Knotenpunkte beeinflusst werden. Werden zu viele Knoten verwendet, so wird mögliches Rauschen in den Daten abgebildet und es kommt zur Überanpassung. Werden zu wenige Knoten gewählt, so bildet die Approximation die Daten möglicherweise zu ungenau ab und das dynamische Modell wird nicht mehr richtig repräsentiert. Darüber hinaus bietet die Platzierung der Knotenpunkte die Möglichkeit, Wissen über die Daten in die Approximation einfließen zu lassen. Sollte etwa aus physikalischen Überlegungen bekannt sein, dass ein Zustand an einer bestimmten Stelle nicht differenzierbar ist, so kann an dieser Stelle ein Knoten mit der entsprechenden Häufigkeit platziert werden.

Im Ansatz von Varah [\[76\]](#) werden der Grad der B-Splines sowie die Anzahl und Platzierung der Knotenpunkte manuell angepasst, bis die Daten nach subjektivem Empfinden bestmöglich approximiert werden. Zur Beurteilung wird eine grafische Darstellung der Approximation verwendet.

Um eine Überanpassung von x_{app} an die Daten zu vermeiden, verfolgen verschiedene Variationen der Kollokationsmethode die Strategie, über einen Strafterm in der Zielfunktion die Glätte der Approximation zu regulieren. In vielen Ansätzen wird die L_2 -Norm der r -ten Ableitung von x_{app} minimiert [\[54, 55, 56, 67, 69\]](#). Das Optimierungsproblem [\(4.8\)](#) wird damit zu

$$\min_{c_j \in \mathbb{R}^{N_j}} \sum_{k=1}^{n_t} \left((\bar{x}_k)_j - x_{\text{app},j}(t_k; c_j) \right)^2 + \alpha_j \int_{t_0}^{t_f} \left[x_{\text{app},j}^{(r_j)}(t; c_j) \right]^2 dt \quad (4.13)$$

mit $r_j \in \{0, \dots, d_j\}$ und $\alpha_j \in \mathbb{R}_{>0}$ für $j = 1, \dots, n_x$. Über die Faktoren α_j kann

reguliert werden, wie stark der Einfluss des Strafterms auf die Approximation sein soll. Der Strafterm kann für verschiedene Zustände unterschiedlich gewichtet werden. Typischerweise wird $r_j = 2$ gewählt [52]. Dies bedeutet anschaulich, dass die Krümmung der Approximation x_{app} möglichst klein sein soll. Schwetlick und Schütze [67, 69] stellen Methoden vor, wie darüber hinaus die Knotenfolgen der zugrundeliegenden B-Splines als Optimierungsvariablen in das Problem (4.13) aufgenommen werden können.

Poyton et al. [52] geben zu bedenken, dass die durch (4.13) bestimmte Approximation das zugrunde liegende dynamische Modell unter Umständen nicht mehr zufriedenstellend erfüllt. Sie schlagen deshalb eine andere Strategie vor, um eine Überanpassung von x_{app} zu vermeiden. Statt des auf der L_2 -Norm beruhenden Strafterms wird ein auf dem dynamischen Modell basierender Strafterm in der Zielfunktion addiert. Zu gegebenen Koeffizienten c_j und Modellparametern p ist der Fehler im dynamischen Modell für den j -ten Zustand durch (4.11) gegeben:

$$E_j(c_j, p) = \int_{t_0}^{t_f} (\dot{x}_{\text{app},j}(t; c_j) - f_j(t, x_{\text{app}}(t; c_j), u_{\text{app}}(t), p))^2 dt.$$

Anders als in der klassischen Methode approximieren Poyton et al. [52] den Fehler E_j jedoch mittels eines numerischen Integrationsverfahrens. Zum Bestimmen der Koeffizienten von x_{app} schlagen Poyton et al. [52] das Optimierungsproblem

$$\min_{c_1 \in \mathbb{R}^{N_1}, \dots, c_{n_x} \in \mathbb{R}^{N_{n_x}}} \sum_{j=1}^{n_x} \sum_{k=1}^{n_t} \left((\bar{x}_k)_j - x_{\text{app},j}(t_k; c_j) \right)^2 + \beta_j E_j(c_j, p) \quad (4.14)$$

mit $\beta_j \in \mathbb{R}_{>0}$ vor. Durch die Koeffizienten β_j kann beeinflusst werden, ob der Schwerpunkt der Optimierung auf einer guten Approximation der Daten oder auf dem Einhalten des dynamischen Modells durch x_{app} liegt. Es sollte ein kleiner Wert für β_j gewählt werden, wenn genaue Daten vorliegen und Unsicherheiten im dynamischen Modell bestehen. Sind die Daten dagegen ungenau und es gibt nur wenige Unsicherheiten im dynamischen Modell, so sollte ein großer Wert für β_j gewählt werden. Eine gute Wahl von β_j ist ausschlaggebend für das Gelingen des Ansatzes von Poyton et al. [52]. Varziri et al. [77] leiten eine Formel her, welche unter gewissen Annahmen eine optimale Wahl von β_j aus Unsicherheiten in den Daten und dem dynamischen Modell berechnet. Die Schwierigkeit besteht dabei darin, diese Unsicherheiten zu quantifizieren.

Bei der klassischen Kollokationsmethode werden im ersten Schritt die Koeffizienten von x_{app} bestimmt. Im zweiten Schritt wird die Approximation der Daten genutzt, um die Modellparameter p zu identifizieren. Das Problem (4.14) soll den ersten Schritt der Kollokationsmethode ersetzen, also genutzt werden, um die Koeffizienten von x_{app} zu bestimmen. Nun hängt (4.14) jedoch explizit von den Modell-

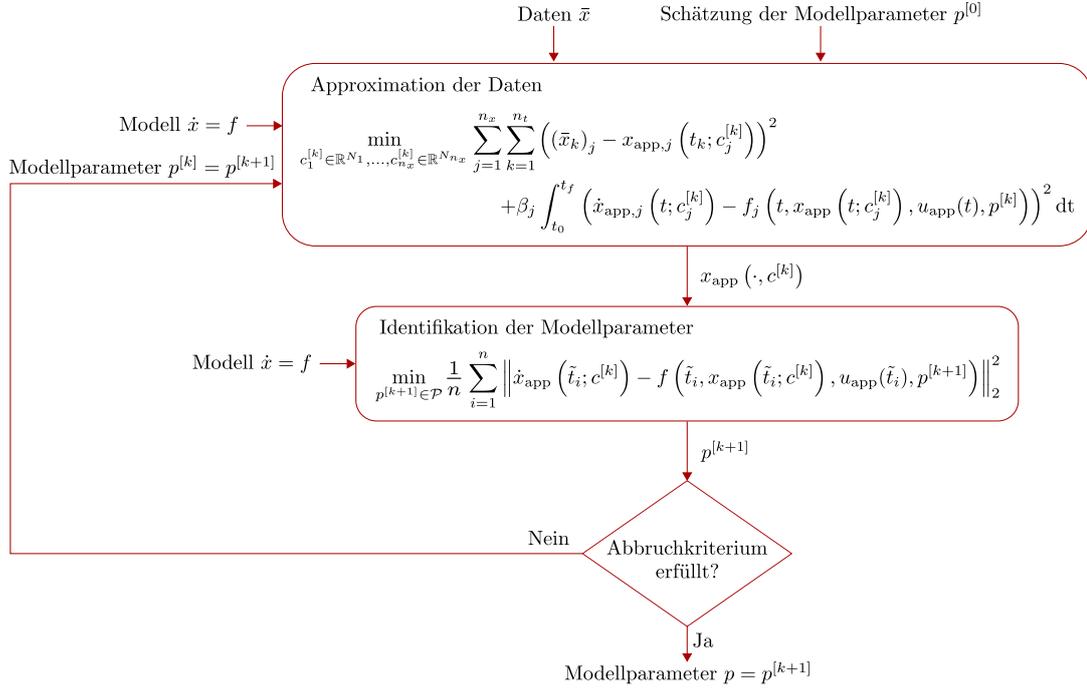


Abbildung 4.2: Schematisches Vorgehen bei der erweiterten Kollokationsmethode nach Poyton et al. [52]

parametern p ab, welche erst im zweiten Schritt der Methode bestimmt werden. Poyton et al. [52] und Varziri et al. [77] stellen zwei Auswege aus diesem Dilemma vor.

Von Poyton et al. [52] wird folgendes iteratives Vorgehen vorgeschlagen. Zunächst wird eine Schätzung $p^{[0]}$ der Modellparameter bestimmt. Dies kann beispielsweise durch die klassische Kollokationsmethode geschehen. Mit dieser Startschätzung wird das Problem (4.14) gelöst, um Koeffizienten $c^{[1]}$ der Approximation zu berechnen. Mithilfe von $x_{\text{app}}(\cdot, c^{[1]})$ können wie in der klassischen Kollokationsmethode Modellparameter $p^{[1]}$ bestimmt werden. Diese Iteration wird so lange wiederholt, bis eine Konvergenz von $c^{[k]}$ und $p^{[k]}$ festgestellt wird. Das Vorgehen ist in [Abbildung 4.2](#) schematisch dargestellt. Die zuvor beschriebene Iteration wird so lange ausgeführt, bis ein geeignetes Abbruchkriterium erfüllt ist. Die Wahl eines geeigneten Abbruchkriteriums ist dabei dem Anwender überlassen. Dem Autor sind keine Untersuchungen zum Konvergenzverhalten des iterativen Ansatzes von Poyton et al. [52] bekannt.

Von Varziri et al. [77] wird als alternativer Lösungsweg angeregt, die Modellparameter p ebenfalls als Optimierungsvariablen in das Problem (4.14) aufzunehmen. Es können dann sowohl c als auch p als Lösung des Optimierungsproblems

$$\min_{c_1 \in \mathbb{R}^{N_1}, \dots, c_{n_x} \in \mathbb{R}^{N_{n_x}}, p \in \mathcal{P}} \sum_{k=1}^{n_t} \sum_{j=1}^{n_x} \left((\bar{x}_k)_j - x_{\text{app},j}(t_k; c_j) \right)^2 + \beta_j E_j(c_j, p) \quad (4.15)$$

berechnet werden. Sowohl der iterative Ansatz von Poyton et al. [52] als auch der Ansatz von Varziri et al. [77] können auf Probleme übertragen werden, bei denen nicht für alle Zustände Messdaten vorliegen. Die Approximation x_{app} wird in diesem Fall lediglich mithilfe der Informationen über das dynamische Modell bestimmt.

Problem (4.15) kann auch auf eine andere Weise interpretiert werden. Tsang [75] sowie Tjoa und Biegler [74] verwenden den Lösungsansatz der vollen Diskretisierung, um das Parameteridentifikationsproblem (DPIP) zu lösen. Als Diskretisierungsverfahren für das Anfangswertproblem (4.5) verwenden sie dabei ein Kollokationsverfahren [15, S. 268 ff.]. Bei diesem Ansatz werden die Nebenbedingungen

$$\dot{x}_{\text{app}}(\tau_i; c) - f(\tau_i, x_{\text{app}}(\tau; c), u_{\text{app}}(\tau_i), p) = 0 \quad (4.16)$$

formuliert. An den sogenannten Kollokationspunkten $\tau_i \in [t_0, t_f]$ soll die Ansatzfunktion x_{app} die Differentialgleichung erfüllen. In Problem (4.15) ist (4.16) nicht als Nebenbedingung enthalten. Stattdessen wird das Residuum der Gleichung (4.16) minimiert. Das Optimierungsproblem (4.15) kann in diesem Sinne als Relaxation des Ansatzes der vollen Diskretisierung mit Kollokationsverfahren als Diskretisierungsverfahren interpretiert werden. Ein weiterer Unterschied zwischen den Ansätzen ist, dass Tsang [75] sowie Tjoa und Biegler [74] Polynome bzw. Lagrange-Polynome als Ansatzfunktionen wählen und keine ν -regulären stückweisen Polynome.

Kapitel 5

Analyse des Einflusses von Daten auf die Parameter dynamischer Modelle

Die Aufgabe der Parameteridentifikation besteht darin, anhand von Daten die Parameter eines mathematischen Modells zu identifizieren. In diesem Kapitel wird eine Methode zum Analysieren des Einflusses der Daten auf die Modellparameter vorgestellt. Dabei werden insbesondere dynamische parameterabhängige Modelle betrachtet.

5.1 Motivation

In [Kapitel 4](#) wurden Methoden zum Identifizieren der Modellparameter in expliziten und dynamischen parameterabhängigen Modellen vorgestellt. Dabei wurde nicht hinterfragt, ob die Daten überhaupt geeignet sind, um die Modellparameter zu identifizieren. Werden die Daten nicht hinreichend genau durch das mathematische Modell beschrieben, ist eine Identifikation der Modellparameter oftmals nicht möglich. Doch auch falls das mathematische Modell den zugrundeliegenden Prozess adäquat beschreibt, ist nicht garantiert, dass die Modellparameter identifiziert werden können. Dies ist zum Beispiel der Fall, wenn die Informationen zum Bestimmen der Modellparameter nicht in den Daten enthalten sind.

Eine Kenngröße aus der Statistik, mit der die Qualität der identifizierten Modellparameter bewertet werden kann, ist die Fisher-Information [[25](#), S. 228 ff.]. Die sogenannte Cramér-Rao-Ungleichung [[57](#)] besagt, dass unter gewissen Voraussetzungen das Inverse der Fisher-Information eine untere Schranke für die Kovarianzmatrix der Modellparameter ist. Damit kann beurteilt werden, wie genau die Modellparameter anhand der gegebenen Daten theoretisch identifiziert werden können. Je kleiner die

Varianzen und Kovarianzen der Modellparameter sind, desto genauer können die Modellparameter bestimmt werden. Vor diesem Hintergrund ist es wünschenswert, dass die Daten bzw. das Experiment, welches zu den Daten führt, eine möglichst große Fisher-Information aufweisen.

Bei der sogenannten optimalen Versuchsplanung wird der Zusammenhang zwischen einer großen Fisher-Information und der Möglichkeit einer genauen Identifikation der Modellparameter ausgenutzt [53]. Es wird ein Optimalsteuerungsproblem formuliert, welches als Ziel hat, die Steuerungen eines dynamischen Modells zu bestimmen, welche die Fisher-Information maximieren. Ein Experiment, das mit diesen Steuerungen durchgeführt wird, erzeugt zumindest theoretisch Daten, welche eine genaue Identifikation der Modellparameter erlauben.

Für eine ausführlichere Darstellung der Hintergründe zur Fisher-Information und der optimalen Versuchsplanung müssen zunächst einige theoretische Grundlagen der Statistik und optimalen Steuerung erläutert werden. Dies würde über den Rahmen dieser Arbeit hinaus gehen. Für mehr Informationen sei auf Georgii [25] und Pukelsheim [53] verwiesen.

Mithilfe der Fisher-Information kann bestimmt werden, wie gut Daten unter statistischen Gesichtspunkten geeignet sind, um die Modellparameter eines mathematischen Modells zu bestimmen. Dies ist eine globale Information, weil die Daten als Ganzes betrachtet werden. In dieser Arbeit wird eine Methode vorgestellt, wie mithilfe parametrischer Sensitivitätsanalyse (vgl. [Abschnitt 2.3](#)) beurteilt werden kann, welchen Einfluss bestimmte Bereiche der Daten auf die Identifikation der Modellparameter haben. Im Gegensatz zur Fisher-Information handelt es sich hierbei um eine lokale Information.

Es sei darauf hingewiesen, dass der hier vorgestellten Methode keine statistischen Überlegungen zugrunde liegen. Die Information, welche Datenbereiche die Modellparameter beeinflussen, kann folglich nicht genutzt werden, um zu quantifizieren wie genau die Modellparameter unter statistischen Gesichtspunkten identifiziert werden können. Dennoch ist die Information in verschiedener Hinsicht interessant.

Das Wissen darüber, welche Daten tatsächlich einen Einfluss auf bestimmte Modellparameter haben, kann zu einem besseren Verständnis des zugrundeliegenden mathematischen Modells führen. Oftmals sind physikalische Überlegungen die Grundlage der mathematischen Modellierung. Um den Einfluss verschiedener physikalischer Effekte zu berücksichtigen, werden Terme zum Modell hinzugefügt, welche typischerweise von Modellparametern abhängen. Durch Vorwissen ist es oftmals möglich, gewisse Bereiche in den Daten einem bestimmten physikalischen Effekt zuzuordnen. Mit der vorgestellten Methode kann analysiert werden, ob die Modellparameter, welche den Effekt im Modell beschreiben sollen, tatsächlich durch diese Daten beeinflusst werden. Ist dies nicht der Fall, so deutet das darauf hin, dass der Effekt nicht korrekt durch das Modell beschrieben wird. Dieses Wissen kann genutzt werden, um

das Modell zu verbessern.

Eine weitere mögliche Anwendung der in diesem Kapitel vorgestellten Methode greift das Thema der optimalen Versuchsplanung auf. Mithilfe der Methode kann analysiert werden, welche Datenbereiche einen Einfluss auf die Identifikation bestimmter Modellparameter haben. Sollte es Bereiche geben, welche keinen der Modellparameter beeinflussen, so deutet das darauf hin, dass diese Daten nicht wichtig für die Parameteridentifikation sind. Bei der Planung von weiteren Versuchen kann daher vermieden werden, solche Daten zu erzeugen. Dies ist insbesondere interessant, wenn die Durchführung der Versuche mit hohen Kosten verbunden ist. Um Kosten einzusparen, ist es wünschenswert, lediglich notwendige Versuche durchzuführen.

Im Zentrum der folgenden Untersuchungen stehen dynamische Parameteridentifikationsprobleme (vgl. [Abschnitt 4.2](#)). Es sei jedoch darauf hingewiesen, dass die Ideen der vorgestellten Methode auch auf explizite Parameteridentifikationsprobleme übertragen werden können.

5.2 Analyse des Einflusses von Daten in dynamischen Parameteridentifikationsproblemen mittels parametrischer Sensitivitätsanalyse

In diesem Abschnitt wird untersucht, welchen Einfluss Daten auf die Modellparameter eines dynamischen parameterabhängigen Modells nach [Definition 4.3](#) haben. Es wird das dynamische Parameteridentifikationsproblem

$$\begin{aligned} \min_{x_0 \in \mathbb{R}^{n_x}, p \in \mathcal{P}} F(p, x_0) &:= \sum_{k=1}^{n_t} \|\bar{x}_k - x_\sigma(t_k)\|_2^2 \\ \text{unter } \dot{x}(t) &= f(t, x(t), u_{\text{app}}(t), p), \quad t_0 \leq t \leq t_f, \\ x(t_0) &= x_0 \end{aligned} \tag{DPIP}$$

nach [Definition 4.4](#) betrachtet. Es wird davon ausgegangen, dass (DPIP) zum Lösen auf ein nichtlineares Optimierungsproblem zurückgeführt wird. Die Menge der zulässigen Werte der Modellparameter sei daher von der Form $\mathcal{P} = \{p \in \mathbb{R}^{n_p} | g(p) \leq 0 \text{ und } h(p) = 0\}$ mit stetig differenzierbaren Funktionen $g: \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_g}$ und $h: \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_h}$. Formal kann das resultierende Problem als parametrisches Optimierungsproblem mit den Datenpunkten \bar{x}_k als Störparametern aufgefasst werden. Mithilfe der parametrischen Sensitivitätsanalyse (vgl. [Abschnitt 2.3](#)) könnten die Sensitivitäten der optimalen Modellparameter bezüglich der Datenpunkte bestimmt werden. Mithilfe der Sensitivitäten ließe sich abschätzen, wie der Einfluss einzelner Datenpunkte auf die identifizierten Modellparameter

ist. Dieser Ansatz ist jedoch nicht immer praktikabel, da es mitunter extrem viele Datenpunkten gibt. Das Bestimmen der Sensitivitäten wäre dadurch trotz effizienter Berechnungsmöglichkeiten sehr zeitaufwändig. Außerdem wäre der Einfluss eines einzelnen Datenpunktes auf die identifizierten Modellparameter sehr gering und die Sensitivitäten könnten durch Rauschen in den Daten beeinflusst werden.

Auch aus praktischer Sicht ist der Einfluss eines einzelnen Datenpunktes auf die Modellparameter oftmals nicht von Interesse. Vielmehr soll der Einfluss eines Datenbereichs $\mathcal{D} \subset \{\bar{x}_1, \dots, \bar{x}_{n_t}\}$ beurteilt werden. Der Einfluss des Datenbereichs \mathcal{D} kann durch Summieren der Sensitivitäten bezüglich der einzelnen Datenpunkte $\bar{x}_k \in \mathcal{D}$ bestimmt werden. Diese Berechnung ließe sich vereinfachen, indem die Daten zunächst in den Bereich \mathcal{D} eingeteilt werden würden. Alle Datenpunkte des Bereichs könnten mit einem künstlichen Störparameter q multipliziert werden. Für den Nominalwert $q_0 = 1$ ergäbe sich das ursprüngliche Parameteridentifikationsproblem. Durch Bestimmen der Sensitivitäten der optimalen Modellparameter bezüglich des Störparameters q könnte der Einfluss der Datenpunkte $\bar{x}_k \in \mathcal{D}$ auf die Modellparameter bestimmt werden. Dieser Ansatz hat zum einen den Vorteil, dass weniger Sensitivitätsdifferentialrechner berechnet werden müssen. Zum anderen wäre der Einfluss von verrauschten Messdaten vermutlich kleiner, da das Rauschen innerhalb eines Bereichs idealerweise ausgeglichen wird. Ein Nachteil des Ansatzes ist, dass bereits vor dem Lösen des Parameteridentifikationsproblems die Daten in Bereiche eingeteilt werden müssen. Es ist nicht ohne Weiteres möglich, im Nachhinein eine andere Einteilung der Daten vorzunehmen. Dieser Nachteil kommt insbesondere dann zum Tragen, wenn das Lösen des Parameteridentifikationsproblems aufwändig ist und viel Zeit in Anspruch nimmt.

Im Folgenden wird ein weiterer Ansatz zum Bestimmen des Einflusses einzelner Datenbereiche auf die Modellparameter vorgestellt. Dieser zeichnet sich dadurch aus, dass der Einfluss beliebiger Datenbereiche untersucht werden kann, ohne das Parameteridentifikationsproblem mehrmals lösen zu müssen. Darüber hinaus wird das Berechnen der Sensitivitäten der Modellparameter bezüglich aller Datenpunkte vermieden. Das Vorgehen ist schematisch in [Abbildung 5.1](#) dargestellt. Zunächst werden die Daten durch ν -reguläre stückweise Polynome approximiert (vgl. [Abschnitt 3.1](#)). Als Basisfunktionen werden dabei B-Splines verwendet (vgl. [Abschnitt 3.3](#)). Auf diese Weise werden die Daten geglättet und mögliches Rauschen reduziert. Als nächstes werden die Modellparameter des dynamischen Modells identifiziert. Dabei werden statt der Daten die Approximationen der Daten verwendet. In dem resultierenden Optimierungsproblem können die Koeffizienten der B-Splines in der Basisdarstellung der Approximationen als Störparameter interpretiert werden. Mithilfe parametrischer Sensitivitätsanalyse können die Sensitivitäten der optimalen Modellparameter bezüglich dieser Koeffizienten bestimmt werden. Im Allgemeinen gibt es deutlich weniger Koeffizienten als Datenpunkte. Andernfalls würde die Approximation der Daten keine glättende Eigenschaft haben. Aus diesem Grund reduziert sich die Anzahl

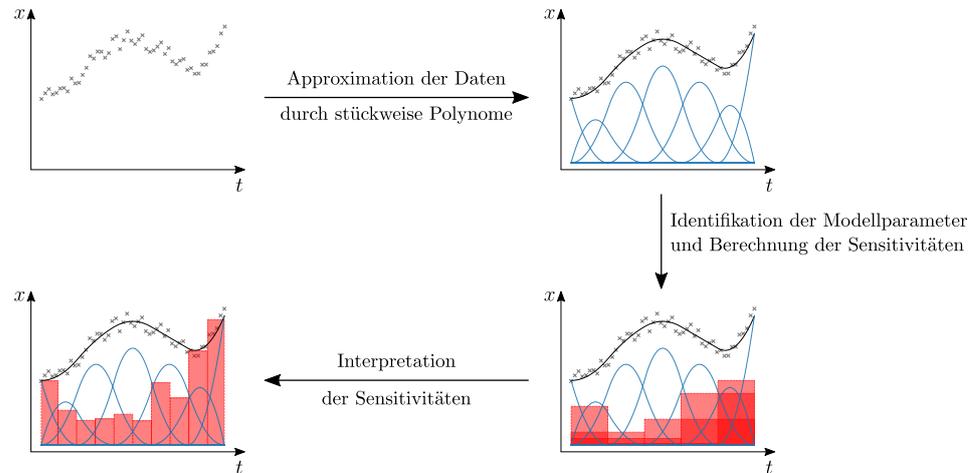


Abbildung 5.1: Schematische Darstellung der Analyse des Einflusses von Daten in dynamischen Parameteridentifikationsproblemen

der zu berechnenden Sensitivitätsdifferenziale. Um den Einfluss bestimmter Bereiche in den Daten auf die Modellparameter zu beurteilen, werden die Sensitivitäten interpretiert. Zunächst beziehen sich die berechneten Sensitivitäten lediglich auf Änderungen in den Koeffizienten der B-Splines. Dennoch ist es möglich, Rückschlüsse über den Einfluss beliebiger Datenbereiche zu ziehen. Dazu wird die Störung einzelner Datenpunkte mit Änderungen in den Koeffizienten der B-Splines verknüpft. In den folgenden Abschnitten werden die drei Schritte im Detail dargestellt.

5.2.1 Approximation der Daten durch stückweise Polynome

Im ersten Schritt der vorgestellten Methode werden die Daten der Zustände und der Steuerungen durch ν -reguläre stückweise Polynome approximiert. Diese Wahl ist aus mehrerer Hinsicht geeignet. Zum einen sind stückweise Polynome flexibel genug, um auch komplizierte Verläufe in den Daten abbilden zu können. Zum anderen kann über den Vektor ν beeinflusst werden, welche Regularität an den Sprungstellen gegeben sein soll. Auf diese Weise kann Wissen über die Regularität der Zustände und der Steuerungen in die Approximation einfließen. Dies wurde bereits ausführlich in [Abschnitt 4.2.2.1](#) diskutiert.

Als Basisfunktionen der ν -regulären stückweisen Polynome werden B-Splines gewählt. Für den j -ten gemessenen Zustand werde ein ν -reguläres stückweises Polynom vom Grad d_j^x zur Approximation verwendet. Die zugehörige B-Spline Basis sei durch die Knotenfolge $\tau_j^x \in \mathbb{R}^{l_j^x+1}$ definiert. Dann hat die Approximation \tilde{x}_{app} die

Gestalt

$$\tilde{x}_{\text{app},j}(t; c_j^x) = \sum_{i=0}^{l_j^x - d_j^x - 1} c_{j,i}^x B_{i,d_j^x, \tau_j^x}(t) \quad \text{für } j = 1, \dots, n_{\bar{x}} \quad (5.1)$$

mit geeigneten Koeffizienten $c_j^x \in \mathbb{R}^{l_j^x - d_j^x}$. Analog werden die Daten der Steuerungen durch ν -reguläre stückweise Polynome vom Grad d_j^u approximiert. Es sei

$$\tilde{u}_{\text{app},j}(t; c_j^u) = \sum_{i=0}^{l_j^u - d_j^u - 1} c_{j,i}^u B_{i,d_j^u, \tau_j^u}(t) \quad \text{für } j = 1, \dots, n_u \quad (5.2)$$

mit geeigneten Knotenfolgen $\tau_j^u \in \mathbb{R}^{l_j^u + 1}$ und Koeffizienten $c_j^u \in \mathbb{R}^{l_j^u - d_j^u}$. Es wird eine Tilde zur Kennzeichnung der Approximationen verwendet, um Verwechslungen mit den Approximation x_{app} und u_{app} aus [Kapitel 4](#) zu vermeiden.

Seien $N_j^x := l_j^x - d_j^x$ und $N_j^u := l_j^u - d_j^u$. Die Koeffizienten c_j^x und c_j^u werden so gewählt, dass die quadrierte Abweichung zwischen den Daten und den Approximationen minimal ist:

$$\min_{c_j^x \in \mathbb{R}^{N_j^x}} \sum_{k=1}^{n_t} \left((\bar{x}_k)_j - \tilde{x}_{\text{app},j}(t_k; c_j^x) \right)^2 \quad \text{für } j = 1, \dots, n_{\bar{x}} \quad (5.3)$$

und

$$\min_{c_j^u \in \mathbb{R}^{N_j^u}} \sum_{k=1}^{n_t} \left((\bar{u}_k)_j - \tilde{u}_{\text{app},j}(t_k; c_j^u) \right)^2 \quad \text{für } j = 1, \dots, n_u. \quad (5.4)$$

Die Verwendung der B-Spline Basis hat zum einen die bereits in [Abschnitt 4.2.2.1](#) diskutierten numerischen Vorzüge. Zum anderen ist es für die vorgestellte Methode von Vorteil, dass B-Splines kompakte Träger haben. Die Koeffizienten $c_{j,i}^x$ und $c_{j,i}^u$ haben aus diesem Grund nur lokal einen Einfluss auf die Approximationen \tilde{x}_{app} bzw. \tilde{u}_{app} . Dadurch kann effizient berechnet werden, wie sich eine Störung der Daten auf die Koeffizienten der B-Splines auswirkt. Dies wird ausführlicher in [Abschnitt 5.2.3](#) beschrieben.

5.2.2 Identifikation der Modellparameter und Berechnung der Sensitivitäten

Im zweiten Schritt der Methode werden die Modellparameter des dynamischen parameterabhängigen Modells identifiziert. Dabei werden statt der Daten die Approximationen \tilde{x}_{app} und \tilde{u}_{app} verwendet. Aus (DPIP) ergibt sich das Parameteridentifi-

kationsproblem

$$\begin{aligned} \min_{x_0 \in \mathbb{R}^{n_x}, p \in \mathcal{P}} F(p, x_0) &:= \sum_{k=1}^{n_t} \|\tilde{x}_{\text{app}}(t_k; c^x) - x_\sigma(t_k)\|_2^2 \\ \text{unter } \dot{x}(t) &= f(t, x(t), \tilde{u}_{\text{app}}(t; c^u), p), \quad t_0 \leq t \leq t_f, \\ x(t_0) &= x_0 \end{aligned} \quad (5.5)$$

Dabei bezeichnen $c^x = (c_1^x, \dots, c_{n_{\bar{x}}}^x)$ und $c^u = (c_1^u, \dots, c_{n_u}^u)$ die Vektoren der Koeffizienten der B-Splines in den Basisdarstellungen der Approximationen aller Zustände bzw. Steuerungen. In Problem (5.5) sind im Vergleich zu (DPIP) zum einen die Daten \bar{x} durch die Approximation \tilde{x}_{app} ersetzt worden. Zum anderen ist die in (DPIP) noch beliebige Approximation der Steuerungen u_{app} durch die konkrete Approximation \tilde{u}_{app} ersetzt worden.

Die im ersten Schritt bestimmten Approximationen \tilde{x}_{app} und \tilde{u}_{app} können verwendet werden, um (5.5) mit der Kollokationsmethode zu lösen (vgl. Abschnitt 4.2.2). Dazu wird vorausgesetzt, dass für alle Zustände Daten vorliegen. Zum Identifizieren der Modellparameter wird in der Kollokationsmethode das Optimierungsproblem

$$\min_{p \in \mathcal{P}} \frac{1}{n_t} \sum_{k=1}^{n_t} \|\dot{\tilde{x}}_{\text{app}}(t_k; c^x) - f(t_k, \tilde{x}_{\text{app}}(t_k; c^x), \tilde{u}_{\text{app}}(t_k; c^u), p)\|_2^2 \quad (5.6)$$

gelöst. Die Koeffizienten der B-Splines c^x und c^u können in (5.6) als Störparameter interpretiert werden. Seien $c^{x,0}$ und $c^{u,0}$ die Nominalwerte der B-Spline Koeffizienten im Optimierungsproblem (5.6) und p^* die zugehörige optimale Lösung. Mithilfe der in Abschnitt 2.3 vorgestellten Konzepte der parametrischen Sensitivitätsanalyse können die Sensitivitätsdifferenziale

$$\frac{dp_r}{dc_j^x}(p^*, c^{x,0}) \quad \text{für } j = 1, \dots, n_{\bar{x}}, r = 1, \dots, n_p \quad (5.7)$$

und

$$\frac{dp_r}{dc_j^u}(p^*, c^{u,0}) \quad \text{für } j = 1, \dots, n_u, r = 1, \dots, n_p \quad (5.8)$$

der identifizierten Modellparameter bezüglich der Koeffizienten der B-Splines berechnet werden. Die Lösung von (5.6) hängt im Allgemeinen nichtlinear von den Modellparametern p sowie den Koeffizienten der B-Splines c^x und c^u ab. Deshalb müssen die Sensitivitäten (5.7) und (5.8) in der Regel numerisch berechnet werden.

Durch eine geeignete Interpretation von (5.7) und (5.8) kann der Einfluss bestimmter Datenbereiche auf die identifizierten Modellparameter bestimmt werden. Dies wird im folgenden Abschnitt näher erläutert.

5.2.3 Interpretation der Sensitivitäten

Der besseren Übersicht halber konzentrieren sich die folgenden Überlegungen auf die Sensitivitäten dp_r/dc_j^x des r -ten Modellparameters bezüglich des j -ten Zustands. Sie lassen sich in analoger Weise auf die Sensitivitäten der anderen Modellparameter und bezüglich der übrigen Zustände sowie der Steuerungen übertragen. Auf die Indizes r und j wird im Folgenden verzichtet. Dazu wird in diesem Abschnitt $p := p_r$, $c := c_j^x$, $\bar{x}_k := (\bar{x}_k)_j$ und $\tilde{x}_{\text{app}} := \tilde{x}_{\text{app},j}$ definiert.

Seien c^0 die Nominalwerte der Koeffizienten der B-Splines im Optimierungsproblem (5.6) und p^* die zugehörige optimale Lösung. Aus einer Taylor-Entwicklung erster Ordnung um p^* folgt

$$p(c_i) = p^* + \frac{dp}{dc_i}(p^*, c^0)(c_i - c_i^0) + \mathcal{O}\left(|(c_i - c_i^0)|^2\right) \quad \text{für } i = 0, \dots, N-1$$

bzw.

$$\Delta p = \frac{dp}{dc_i}(p^*, c^0) \Delta c_i + \mathcal{O}\left(|\Delta c_i|^2\right) \quad \text{für } i = 0, \dots, N-1 \quad (5.9)$$

mit $\Delta p := p(c_i) - p^*$ und $\Delta c_i := c_i - c_i^0$. Eine Störung Δc_i im i -ten Koeffizienten c_i^0 der Approximation \tilde{x}_{app} verursacht folglich in linearer Näherung eine Änderung $\Delta p = \frac{dp}{dc_i}(p^*, c^0) \Delta c_i$ im identifizierten Modellparameter p^* . Je größer der Betrag der Sensitivitäten dp/dc_i ist, desto stärker wirkt sich eine Störung des Koeffizienten c_i^0 auf den Modellparameter p^* aus. In den nächsten beiden Abschnitten wird diskutiert, wie die Störungen Δc_i auf bestimmte Bereiche in den Daten zurückgeführt werden können. Mit dieser Information können mithilfe von (5.9) Rückschlüsse auf den Einfluss der Bereiche in den Daten auf den Modellparameter p gezogen werden.

5.2.3.1 Beurteilung des Einflusses von Daten durch die kompakten Träger von B-Splines

Im Folgenden wird ein auf den ersten Blick intuitiver Ansatz beschrieben, wie mithilfe der Sensitivitäten dp/dc_i der Einfluss bestimmter Bereiche in den Daten auf die Modellparameter beurteilt werden kann. Bei diesem Vorgehen treten jedoch einige Probleme auf. Die Probleme werden dargestellt und dienen als Motivation, im weiteren Verlauf der Arbeit eine alternative Vorgehensweise zur Beurteilung des Einflusses bestimmter Bereiche der Daten auf die Modellparameter zu wählen.

Ein vermeintlich intuitiver Ansatz zum Beurteilen des Einflusses bestimmter Datenbereiche auf die Modellparameter beruht auf der Tatsache, dass die Träger von B-Splines kompakt sind. Nach Satz 3.11 beschränkt sich der Träger von $B_{i,d,\tau}$ auf das Intervall $[\tau_i, \tau_{i+d+1}]$. Aus der Darstellung (5.1) folgt, dass eine Störung von c_i die Approximation \tilde{x}_{app} lediglich lokal auf dem Träger von $B_{i,d,\tau}$ beeinflusst. In Abbildung 5.2 ist die Änderung von \tilde{x}_{app} infolge einer Störung Δc_i dargestellt. Folglich

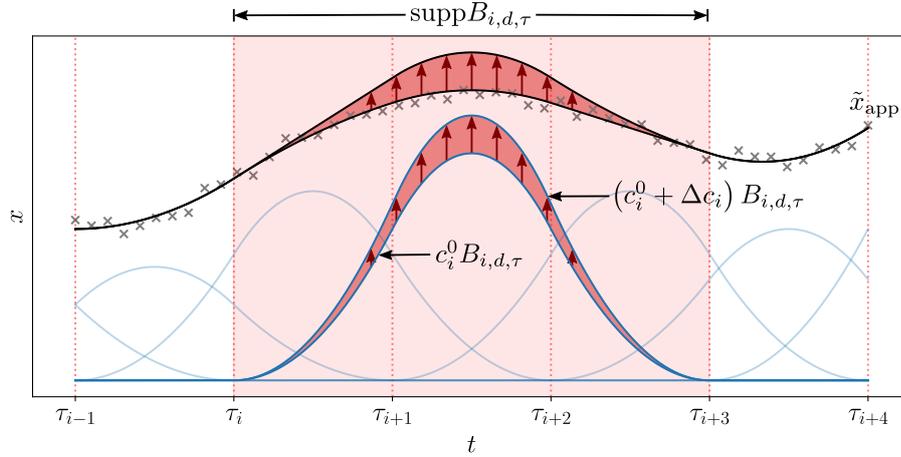
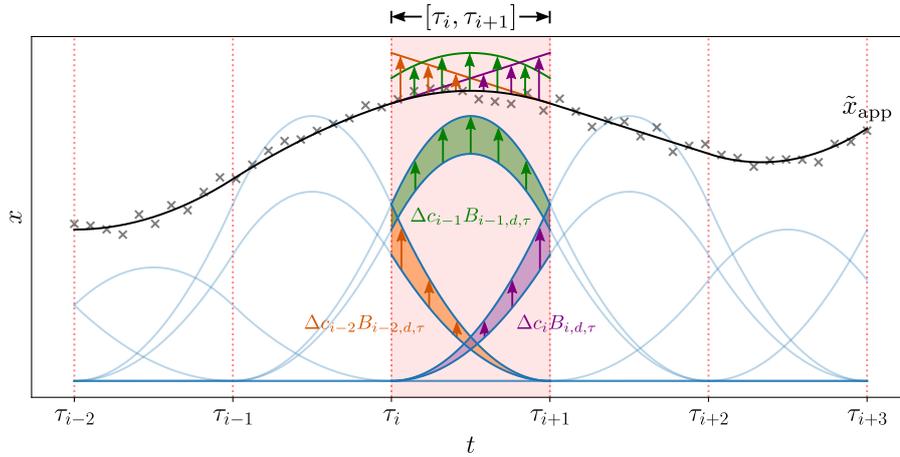


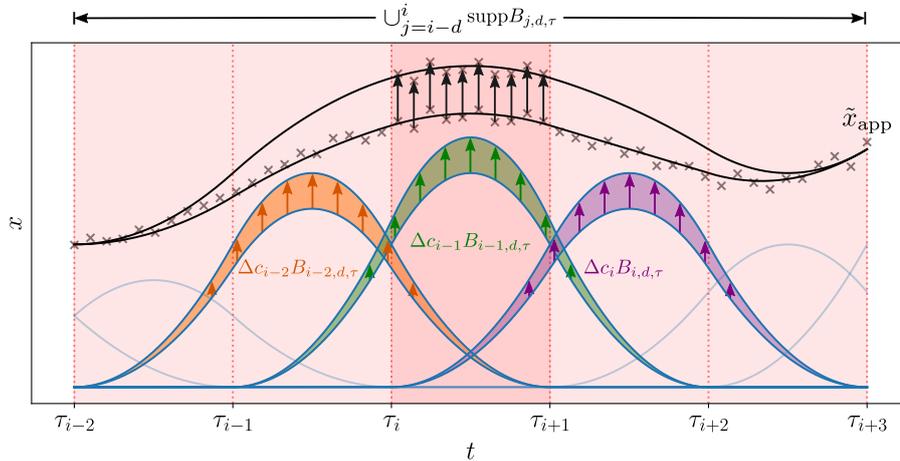
Abbildung 5.2: Änderung der Approximation der Daten \tilde{x}_{app} infolge einer Störung des Koeffizienten des i -ten B-Splines um Δc_i

kann auch das Sensitivitätsdifferential $\frac{dp}{dc_i}$ mit dem Intervall $\text{supp } B_{i,d,\tau}$ in Beziehung gesetzt werden: Je größer die Sensitivität $\frac{dp}{dc_i}$ ist, desto stärker ist der Einfluss von \tilde{x}_{app} im Intervall $\text{supp } B_{i,d,\tau}$ auf den Modellparameter p . Die Funktion \tilde{x}_{app} ist eine Approximation der Daten \bar{x}_k , welche an den diskreten Zeitpunkten t_k für $k = 1, \dots, n_t$ gegeben sind. Aus den vorherigen Überlegungen könnte gefolgert werden, dass die Daten \bar{x}_k mit $k \in \{k' \in \{1, \dots, n_t\} \mid t_{k'} \in \text{supp } B_{i,d,\tau}\}$ einen starken Einfluss auf die Identifikation des Modellparameters p haben, wenn die Sensitivitäten $\frac{dp}{dc_i}$ groß sind.

Weil die Träger verschiedener B-Splines für $d \geq 1$ nicht disjunkt sind, entstehen bei dieser Argumentation Probleme. Zwischen je zwei Knotenpunkten τ_i und τ_{i+1} sind die $d + 1$ B-Splines $B_{i-d,d,\tau}, \dots, B_{i,d,\tau}$ ungleich Null. All diese B-Splines beeinflussen die Approximation \tilde{x}_{app} im Intervall $[\tau_i, \tau_{i+1}]$. Folglich führt auch eine Störung jedes Koeffizienten c_{i-d}, \dots, c_i zu einer Änderung der Approximation \tilde{x}_{app} im Intervall $[\tau_i, \tau_{i+1}]$. Dies ist in [Abbildung 5.3a](#) für den Fall $d = 2$ dargestellt. Um den Einfluss der Daten innerhalb des Intervalls auf den Modellparameter p zu bestimmen, sollten folglich die Sensitivitätsdifferentialen $\frac{dp}{dc_{i-d}}, \dots, \frac{dp}{dc_i}$ berücksichtigt werden. Es ist allerdings unklar, in welcher Form dies geschehen soll. Wie in [Abbildung 5.3b](#) dargestellt, würde eine Störung der Daten im Intervall $[\tau_i, \tau_{i+1}]$ die Koeffizienten aller B-Splines $B_{i-d,d,\tau}, \dots, B_{i,d,\tau}$ beeinflussen. Folglich würde sich eine Störung dieser Daten im gesamten Intervall $\cup_{j=i-d}^i \text{supp } B_{j,d,\tau}$ auf die Approximation \tilde{x}_{app} auswirken. Es ist daher schwierig, den Einfluss der Daten im Intervall $[\tau_i, \tau_{i+1}]$ nur anhand der Sensitivitäten $\frac{dp}{dc_{i-d}}, \dots, \frac{dp}{dc_i}$ zu beurteilen. Im nächsten Abschnitt wird deshalb ein alternativer Ansatz zur Beurteilung des Einflusses bestimmter Bereiche der Daten auf die Modellparameter erläutert.



(a) Auswirkung einer Störung der Koeffizienten der B-Splines im Intervall $[\tau_i, \tau_{i+1}]$ auf die Approximation \tilde{x}_{app}



(b) Auswirkung einer Störung der Daten im Intervall $[\tau_i, \tau_{i+1}]$ auf die Approximation \tilde{x}_{app}

Abbildung 5.3: Probleme bei der Beurteilung des Einflusses von Daten durch die kompakten Träger von B-Splines, beispielhaft für den Fall $d = 2$

5.2.3.2 Beurteilung des Einflusses von Daten durch die Sensitivitäten der Koeffizienten der B-Splines bezüglich der Datenpunkte

In diesem Abschnitt wird ein weiterer Ansatz vorgestellt, wie mithilfe der Sensitivitäten dp/dc_i der Einfluss bestimmter Bereiche der Daten auf die Modellparameter beurteilt werden kann. Dabei ist eine beliebige Aufteilung der Daten zulässig. Es soll der Einfluss der Daten $\mathcal{D} \subset \{\bar{x}_k \mid k = 1, \dots, n_t\}$ auf den Modellparameter p beurteilt werden. Dazu wird bestimmt, wie sich eine Störung der Daten \mathcal{D} auf die Koeffizienten der B-Splines auswirkt. Dies kann mithilfe einer parametrischen Sensitivitätsanalyse des Optimierungsproblems (5.3) geschehen. Durch Lösen von (5.3)

werden die Koeffizienten der B-Splines bestimmt, welche im zweiten Schritt der vorgestellten Methode, dem Lösen von (5.6), als Nominalwerte verwendet werden.

Durch die Konvention dieses Abschnittes, dass der Index des konkreten Zustands nicht mitgeführt wird, vereinfacht sich die Notation von Problem (5.3) zu

$$\min_{c \in \mathbb{R}^N} F(c) := \sum_{k=1}^{n_t} (\bar{x}_k - \tilde{x}_{\text{app}}(t_k; c))^2 \stackrel{(5.1)}{=} \sum_{k=1}^{n_t} \left(\bar{x}_k - \sum_{i=0}^{N-1} c_i B_{i,d,\tau} \right)^2. \quad (5.10)$$

In (5.10) können die Datenpunkte \bar{x}_k als Störparameter aufgefasst werden. Seien \bar{x}^0 die Nominalwerte der Daten und c^* eine optimale Lösung von (5.10). Mithilfe von Methoden der parametrischen Sensitivitätsanalyse können die Sensitivitätsdifferentialie $\frac{dc}{d\bar{x}}(c^*, \bar{x}^0)$ bestimmt werden. Analog zu (5.9) folgt aus einer Taylor-Entwicklung erster Ordnung

$$\tilde{\Delta}c_i = \frac{dc_i}{d\bar{x}_k}(c^*, \bar{x}^0) \Delta\bar{x}_k + \mathcal{O}(|\Delta\bar{x}_k|^2) \quad \text{für } i=0, \dots, N-1, k=1, \dots, n_t \quad (5.11)$$

mit $\tilde{\Delta}c_i := c_i(\bar{x}_k) - c_i^*$ und $\Delta\bar{x}_k := \bar{x}_k - \bar{x}_k^0$. Per Konstruktion der vorgestellten Methode wird die optimale Lösung c^* von Problem (5.10) als Nominalwert c^0 der Koeffizienten der B-Splines in Problem (5.6) zum Identifizieren der Modellparameter verwendet. Folglich ist $c^* = c^0$ und aus (5.9) und (5.11) folgt näherungsweise

$$\Delta p = \frac{dp}{dc_i}(p^*, c^0) \frac{dc_i}{d\bar{x}_k}(c^0, \bar{x}^0) \Delta\bar{x}_k \quad \text{für } i=0, \dots, N-1, k=1, \dots, n_t.$$

Störungen $\Delta\bar{x}_k$ der Datenpunkte $\bar{x}_k \in \mathcal{D}$ verursachen folglich näherungsweise eine Änderung

$$\Delta p(\mathcal{D}) := \sum_{\bar{x}_k \in \mathcal{D}} \sum_{i=0}^{N-1} \frac{dp}{dc_i}(p^*, c^0) \frac{dc_i}{d\bar{x}_k}(c^0, \bar{x}^0) \Delta\bar{x}_k \quad (5.12)$$

im identifizierten Modellparameter p^* . Mithilfe von (5.12) kann der Einfluss der Daten \mathcal{D} auf den Modellparameter p beurteilt werden. Resultiert eine Störung der Daten in einem großen $\Delta p(\mathcal{D})$, so ist der Einfluss von \mathcal{D} groß zu bewerten. Mehr noch bietet (5.12) die Möglichkeit, den Wert des identifizierten Modellparameters zu approximieren, sollten die Datenpunkte $\bar{x}_k \in \mathcal{D}$ um $\Delta\bar{x}_k$ gestört werden.

Die Verwendung der Sensitivitäten $dc/d\bar{x}$ zum Zurückführen des Einflusses der Daten \mathcal{D} auf die Koeffizienten der B-Splines scheint auf den ersten Blick kontraproduktiv zu sein. So war es eine Motivation der Verwendung von B-Splines, dass nicht die Sensitivitäten bezüglich aller Daten bestimmt werden müssen. Eine nähere Betrachtung zeigt jedoch, dass die Sensitivitäten $dc/d\bar{x}$ sehr effizient berechnet werden können.

Da es sich bei (5.10) um ein Optimierungsproblem ohne Nebenbedingungen handelt,

kann die Berechnung der Sensitivitätsdifferenziale vereinfacht werden. Nach [Satz 2.13](#) gilt

$$\nabla_{cc}^2 F(c^*, \bar{x}^0) \frac{dc}{d\bar{x}_k}(c^*, \bar{x}^0) = -\nabla_{c\bar{x}_k}^2 F(c^*, \bar{x}^0) \quad \text{für } k = 1, \dots, n_t. \quad (5.13)$$

Für den Gradienten der Zielfunktion F von [\(5.10\)](#) gilt

$$\left(\nabla_c F\right)_j = 2 \sum_{k=1}^{n_t} \left(\bar{x}_k - \sum_{i=0}^N c_i B_{i,d,\tau}(t_k) \right) (-B_{j,d,\tau}(t_k)) \quad \text{für } j = 0, \dots, N-1.$$

Für die zweiten Ableitungen der Zielfunktion folgt daraus

$$\left(\nabla_{cc}^2 F\right)_{j,l} = 2 \sum_{k=1}^{n_t} B_{l,d,\tau}(t_k) B_{j,d,\tau}(t_k) \quad \text{für } j, l = 0, \dots, N-1 \quad (5.14)$$

und

$$\left(\nabla_{c\bar{x}_k}^2 F\right)_{j,k}^2 = -2B_{j,d,\tau}(t_k) \quad \text{für } j = 0, \dots, N-1, k = 1, \dots, n_t. \quad (5.15)$$

Das Gleichungssystem [\(5.13\)](#) vereinfacht sich mit [\(5.14\)](#) und [\(5.15\)](#) zu

$$A \frac{dc}{d\bar{x}_k}(c^*, \bar{x}^0) = b_k \quad \text{für } k = 1, \dots, n_t \quad (5.16)$$

mit

$$A \in \mathbb{R}^{N \times N}, (A)_{l,j} := \sum_{k=1}^{n_t} B_{l,d,\tau}(t_k) B_{j,d,\tau}(t_k) \quad \text{für } k = 1, \dots, n_t$$

und

$$b^k \in \mathbb{R}^N, \quad b_j^k := B_{j,d,\tau}(t_k) \quad \text{für } k = 1, \dots, n_t.$$

Insbesondere sind sowohl die Matrix A als auch der Vektor b^k unabhängig von der optimalen Lösung c^* und den Nominalparametern \bar{x}^0 . Sie hängen lediglich von den Zeitpunkten der Daten sowie dem Grad und der Knotenfolge der B-Splines ab. Weil B-Splines nach [Satz 3.11](#) kompakte Träger haben, handelt es sich bei der Matrix A um eine symmetrische Bandmatrix mit Bandbreite d . Es sind lediglich die Werte der Hauptdiagonalen und der ersten d Nebendiagonalen ungleich Null.

Sowohl die Symmetrie als auch die Bandstruktur und damit einhergehende Dünnbesetztheit von A kann ausgenutzt werden, um das Gleichungssystem [\(5.16\)](#) effizient zu lösen. Im Allgemeinen ist die Matrix A indefinit. Wegen der Symmetrie von A

bietet sich das sogenannte MINRES-Verfahren [49] zum iterativen Lösen von (5.16) an. Da (5.16) für eine Vielzahl von verschiedenen rechten Seiten b^k gelöst werden muss, kann es von Vorteil sein, zunächst eine Zerlegung der Matrix A zu berechnen. Diese Zerlegung kann genutzt werden, um das Gleichungssystem zu den verschiedenen rechten Seiten durch effizientes Vorwärts- und Rückwärtseinsetzen direkt zu lösen. Irony und Toledo [34] sowie Kaufman [37] stellen Algorithmen zur Zerlegung von indefiniten symmetrischen Bandmatrizen vor.

5.2.4 Anmerkungen zur Methode

Im Folgenden werden einige Anmerkungen zur vorgestellten Methode zur Analyse des Einflusses von Daten auf die Modellparameter dynamischer Modelle mittels parametrischer Sensitivitäten gemacht.

5.2.4.1 Vergleich des Einflusses der Daten verschiedener Zustände und Steuerungen

Die Überlegungen in Abschnitt 5.2.3 wurden der besseren Übersicht halber lediglich für die Sensitivitäten dp_r/dc_j^z des r -ten Modellparameters bezüglich des j -ten Zustands durchgeführt. Es sei an dieser Stelle noch einmal darauf hingewiesen, dass die Überlegungen vollständig analog für die Sensitivitäten der anderen Modellparameter bezüglich der übrigen Zustände sowie der Sensitivitäten der Modellparameter bezüglich der Steuerungen durchgeführt werden können. Folglich kann mithilfe von (5.12) berechnet werden, wie sich die Modellparameter ändern würden, wenn die Daten der verschiedenen Zustände und Steuerungen in bestimmten Bereichen variieren. Um den Einfluss einzelner Datenbereiche zu beurteilen, ist es oftmals hilfreich, die Resultate für verschiedene Zustände und Steuerungen in Relation zueinander zu betrachten. Da die Modellparameter im Allgemeinen mit einer Einheit versehen sind, sollte der Vergleich für jeden Modellparameter einzeln erfolgen.

5.2.4.2 Alternative Bewertung des Einflusses der Daten

Mithilfe von (5.12) kann der Einfluss von Daten $\mathcal{D} \subset \{\bar{x}_k \mid k = 1, \dots, n_t\}$ auf den Modellparameter p beurteilt werden. Resultiert eine Störung der Daten \mathcal{D} in einer großen Änderung des Modellparameters p , so ist der Einfluss von \mathcal{D} auf p groß. Ein Nachteil dieses Ansatzes ist, dass es in (5.12) zur Auslöschung von Termen mit unterschiedlichen Vorzeichen kommen kann. Um den Einfluss der Daten auf den Modellparameter zu beurteilen, ist die explizite Änderung des Modellparameters nicht zwingend erforderlich. Es können auch andere Maße herangezogen werden. Als

alternative Bewertung des Einflusses von \mathcal{D} auf p wird

$$\kappa(\mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{\bar{x}_k \in \mathcal{D}} \sum_{i=0}^{N-1} \left| \frac{dp}{dc_i}(p^*, c^0) \frac{dc_i}{d\bar{x}_k}(c^0, \bar{x}^0) \Delta \bar{x}_k \right| \quad (5.17)$$

vorgeschlagen. Der Wert $\kappa(\mathcal{D})$ kann als mittlerer absoluter Einfluss der Datenpunkte in \mathcal{D} auf p interpretiert werden. Es scheint zweckmäßig, die Störung der Datenpunkte in (5.17) konstant zu wählen, d. h. $\Delta \bar{x}_k = \delta$ für $\delta \in \mathbb{R}$.

5.2.4.3 Grenzen der Methode

Die vorgestellte Methode beruht auf der Berechnung der parametrischen Sensitivitäten dp/dc sowie $dc/d\bar{x}$ und $dc/d\bar{u}$. Diese werden in den optimalen Lösungen der Optimierungsprobleme (5.3), (5.4) und (5.6) ausgewertet. Die theoretische Grundlage für die Berechnung der Sensitivitäten ist die Gültigkeit des [Sensitivitätssatzes](#) ([Satz 2.13](#)). Damit die vorgestellte Methode angewendet werden kann, müssen folglich die Voraussetzungen des Sensitivitätssatzes erfüllt sein. Insbesondere müssen die Lösungen von (5.3), (5.4) und (5.6) die hinreichenden Optimalitätsbedingungen zweiter Ordnung erfüllen. Dies ist vor allem in Problem (5.6) zum Identifizieren der Modellparameter zu beachten. Sollten die Modellparameter nicht anhand der Daten identifiziert werden können, so sind die hinreichenden Optimalitätsbedingungen zweiter Ordnung oftmals nicht erfüllt. Konsequenterweise kann die vorgestellte Methode in einem solchen Fall nicht angewendet werden.

5.2.4.4 Alternative Lösungsverfahren des Parameteridentifikationsproblems

In [Abschnitt 5.2.2](#) wurde vorausgesetzt, dass für alle Zustände Daten gegeben sind. Dies ist notwendig, um Problem (5.5) mithilfe der klassischen Kollokationsmethode zu lösen. Sollten nicht alle Zustände gemessen worden sein, kann die erweiterte Kollokationsmethode nach Poyton et al. [[52](#)] als Lösungsverfahren genutzt werden (vgl. [Abb. 4.2](#)). Die Sensitivitäten können in diesem Fall in den Optimierungsproblemen der letzten Iteration bestimmt werden.

Darüber hinaus wäre es denkbar, das Parameteridentifikationsproblem (5.5) mithilfe eines anderen Lösungsansatzes, wie dem Ansatz der vollen Diskretisierung, auf ein nichtlineares Optimierungsproblem zurückzuführen. In dem resultierenden Problem können die Koeffizienten der B-Splines als Störparameter aufgefasst und die Sensitivitäten mithilfe parametrischer Sensitivitätsanalyse berechnet werden.

Kapitel 6

Numerische Auswertungen

In diesem Kapitel wird die in [Kapitel 5](#) vorgestellte Methode zum Analysieren des Einflusses von Daten auf die Parameter dynamischer Modelle anhand von Beispielen untersucht. Zum Lösen der Optimierungsprobleme (5.3), (5.4) und (5.6) wird die C++ Schnittstelle von WORHP (vgl. [Abschnitt 2.4.2](#)) verwendet. Darüber hinaus werden Funktionalitäten der im Rahmen des Projekts TOPAS³S¹ entwickelten Software zum Lösen von dynamischen Parameteridentifikationsproblemen genutzt. Die Sensitivitätsdifferentialie dp_r/dc_j^x und dp_r/dc_j^y in Problem (5.6) werden mithilfe des in WORHP integrierten Moduls WORHP Zen berechnet.

Die Interpretation der Sensitivitäten dp_r/dc_j^x und dp_r/dc_j^y ist in Python implementiert. Dazu werden die Bibliotheken *NumPy* [48, 82] und *SciPy* [79] verwendet. Das Visualisieren der Ergebnisse geschieht mithilfe der Bibliothek *Matplotlib* [32]. In [Tabelle 6.1](#) sind die Spezifikationen der verwendeten Hardware und Software aufgelistet.

Tabelle 6.1: Spezifikationen der verwendeten Hardware und Software

Hardware	
Betriebssystem	Ubuntu 18.04.4 LTS
Prozessor	Intel [®] Core [™] i5-6200U CPU @ 2.30 GHz
Arbeitsspeicher	16 GB
Software	Version
WORHP	1.13-2
Python	3.6.9
SciPy	1.5.2
NumPy	1.19.1
Matplotlib	3.1.2

¹<http://www.math.uni-bremen.de/zetem/cms/detail.php?id=20043>

In [Abschnitt 6.1](#) wird erläutert, in welcher Form der Einfluss bestimmter Datenbereiche auf die Modellparameter eines parameterabhängigen Modells visualisiert werden kann. Anschließend wird die in [Kapitel 5](#) vorgestellte Methode anhand eines einfachen dynamischen Parameteridentifikationsproblems demonstriert. Schließlich wird die Methode verwendet, um in [Abschnitt 6.3](#) den Einfluss von Daten auf die Modellparameter eines Einspurmodells zu analysieren.

6.1 Visualisierung des Einflusses von Daten auf die Modellparameter parameterabhängiger Modelle

In diesem Abschnitt wird eine Visualisierung des Einflusses von Daten auf die Modellparameter parameterabhängiger Modelle vorgestellt. Im Zuge der numerischen Auswertungen wird diese Darstellung verwendet, um den Einfluss visuell zu beurteilen.

Es werden Daten $\bar{x}_1, \dots, \bar{x}_{n_t}$ zu den diskreten Zeitpunkten $t_1 \leq \dots \leq t_{n_t}$ betrachtet. Des Weiteren wird davon ausgegangen, dass der Einfluss von zeitlich zusammenhängenden Daten $\mathcal{D} = \{\bar{x}_i, \dots, \bar{x}_{i+N}\}$ für $1 \leq i \leq n_t$ und $0 \leq N \leq n_t - i$ beurteilt werden soll. Sei $\Delta p(\mathcal{D})$ die Änderung des Modellparameters p , wenn die Datenpunkte $\bar{x}_k \in \mathcal{D}$ um $\Delta \bar{x}_k$ gestört werden. Der Einfluss der Daten \mathcal{D} auf den Modellparameter kann mithilfe von $\Delta p(\mathcal{D})$ beurteilt werden. Ist die Änderung des Parameters groß, so haben die Daten \mathcal{D} einen großen Einfluss auf p . Ist die Änderung des Parameters klein, so haben sie keinen großen Einfluss. Der Wert $\Delta p(\mathcal{D})$ kann mittels (5.12) berechnet werden.

In [Abbildung 6.1a](#) ist der Einfluss der Daten \mathcal{D} visualisiert. Die horizontale Koordinatenachse des Diagramms kennzeichnet den Verlauf der Zeit. Auf der linken vertikalen Koordinatenachse ist der Wert des Zustands x aufgetragen. Die Datenpunkte $\bar{x}_k \in \mathcal{D}$ werden durch rote Kreuze in dem Diagramm dargestellt. Die übrigen Datenpunkte sind durch schwarze Punkte gekennzeichnet. Auf der rechten vertikalen Koordinatenachse ist der Wert der Änderung des Modellparameters Δp aufgetragen.

Da die Daten \mathcal{D} zeitlich zusammenhängend sind, können sie mit dem Intervall $[t_i, t_{i+N}]$ assoziiert werden. Der Einfluss von \mathcal{D} auf den Modellparameter p ist durch ein Rechteck in [Abbildung 6.1a](#) visualisiert. Die Breite des Rechtecks entspricht der zeitlichen Ausdehnung der Daten. In diesem Beispiel erstreckt sich das Rechteck über das Intervall $[t_i, t_{i+N}]$. Die Höhe des Rechtecks hat den Wert $\Delta p(\mathcal{D})$ und ist auf der rechten Koordinatenachse abzulesen. Um den Einfluss der Daten \mathcal{D} zu beurteilen, sollte die Änderung $\Delta p(\mathcal{D})$ mit den Änderungen der Modellparameter bei Störungen von anderen Datenbereichen in Relation gesetzt werden.

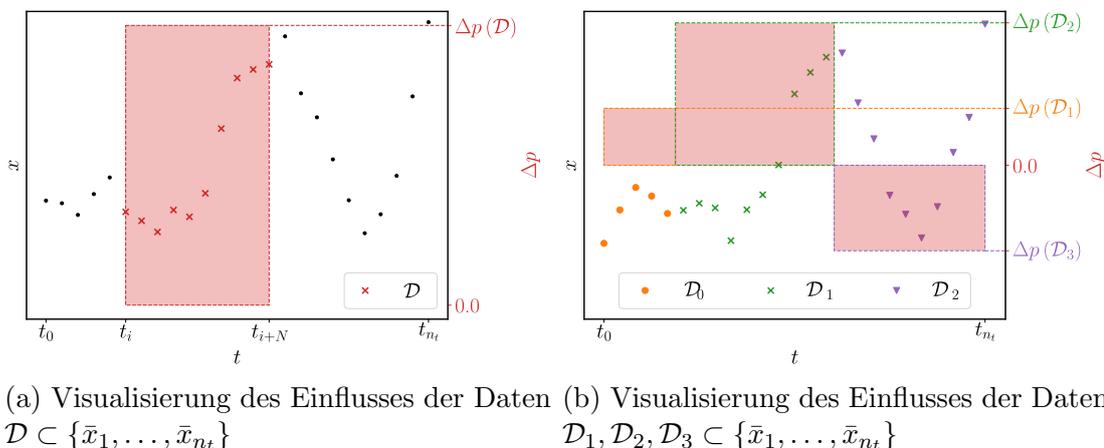


Abbildung 6.1: Visualisierung des Einflusses von Daten auf die Modellparameter parameterabhängiger Modelle

Es seien zeitlich zusammenhängende, disjunkte Daten $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3 \subset \{\bar{x}_1, \dots, \bar{x}_{n_t}\}$ gegeben. Eine Störung der Datenpunkte $\bar{x}_k \in \mathcal{D}_i$ um $\Delta\bar{x}_k$ resultiere in einer Änderung $\Delta p(\mathcal{D}_i)$ des Modellparameters p . Diese Änderungen können wie zuvor beschrieben visualisiert werden. Die Visualisierung ist in [Abbildung 6.1b](#) dargestellt. Die Datenmengen $\mathcal{D}_1, \mathcal{D}_2$ und \mathcal{D}_3 sind durch orangefarbene Punkte, grüne Kreuze bzw. lilafarbene Dreiecke gekennzeichnet. Es sind drei Rechtecke dargestellt, um den Einfluss der drei Datenmengen zu visualisieren. Die Breite der Rechtecke ist etwas größer als die zeitliche Ausdehnung der Daten und so gewählt, dass die Rechtecke aneinander anschließen. Die Höhe der Rechtecke entspricht dem Wert $\Delta p(\mathcal{D}_i)$. Das zu \mathcal{D}_1 gehörige Rechteck ist schmaler als die anderen beiden Rechtecke. Folglich erstrecken sich die Daten \mathcal{D}_1 über ein kleineres Zeitintervall als die Daten \mathcal{D}_2 und \mathcal{D}_3 .

Mithilfe der Rechtecke kann visuell beurteilt werden, wie der Einfluss der Daten $\mathcal{D}_1, \mathcal{D}_2$ und \mathcal{D}_3 auf den Modellparameter p ist. Das zu \mathcal{D}_2 gehörige Rechteck ist höher als die zu \mathcal{D}_1 und \mathcal{D}_3 gehörigen Rechtecke. Folglich würden Störungen $\Delta\bar{x}_k$ der Datenpunkte $\bar{x}_k \in \mathcal{D}_2$ die größte Änderung des Modellparameters p verursachen. Die Daten \mathcal{D}_2 hätten somit den größten Einfluss auf den Modellparameter p . Eine Störung der Daten \mathcal{D}_1 hat dagegen den kleinsten Einfluss auf den Modellparameter. Werden die Datenpunkte $\bar{x}_k \in \mathcal{D}_3$ um $\Delta\bar{x}_k$ gestört, so hätte dies eine negative Änderung des Modellparameters zur Folge. Dies ist dadurch zu erkennen, dass das zu \mathcal{D}_3 gehörige Rechteck in der unteren Halbebene des Koordinatensystems liegt.

Der Einfluss von Daten \mathcal{D} auf den Modellparameter p wird in [Abbildung 6.1](#) durch die Änderung des Modellparameters $\Delta p(\mathcal{D})$ bewertet. Wie in [Abschnitt 5.2.4.2](#) diskutiert, sind auch alternative Bewertungen, wie beispielsweise der mittlere absolute Dateneinfluss $\kappa(\mathcal{D})$ gemäß (5.17), denkbar. Der Einfluss ließe sich komplett analog zu den vorigen Überlegungen visualisieren. Die Höhe der Rechtecke wäre dann nicht durch $\Delta p(\mathcal{D})$ sondern durch $\kappa(\mathcal{D})$ gegeben.

6.2 Analyse des Einflusses von Daten auf den Parameter eines einfachen dynamischen Modells

Im Folgenden wird das einfache dynamische Modell

$$\dot{x}(t) = pu(t), \quad t \in [0, 1] \quad (6.1)$$

betrachtet. Dabei ist sowohl der Zustand x als auch die Steuerung u eindimensional und $p \in \mathbb{R}$ ein skalarer Modellparameter. Es wird der Fall einer konstanten Steuerung $u(t) = \hat{u} \in \mathbb{R}$ untersucht. Die Lösung von (6.1) kann in diesem Fall analytisch berechnet werden und ist durch

$$x(t) = p\hat{u}t + x_0 \quad (6.2)$$

mit $x_0 := x(0)$ gegeben, wobei im Folgenden $x_0 = 0$ angenommen wird. Bei x handelt es sich um eine lineare Funktion mit der Steigung $p\hat{u}$. Der Modellparameter habe den Wert $p^* = 2$ und es gelte $\hat{u} = 0.5$. Die Steigung von x ist folglich 1. Es seien äquidistante Zeitpunkte $0 = t_1 \leq \dots \leq t_{n_t} = 1$ und zugehörige exakte Daten der Zustände $\bar{x} \in \mathbb{R}^{n_t}$ und der Steuerungen $\bar{u} \in \mathbb{R}^{n_t}$ gegeben. Im Folgenden sei $n_t = 50$.

6.2.1 Approximation der Daten

Zum Approximieren der Daten des Zustands x wird ein ν -reguläres stückweises Polynom \tilde{x}_{app} vom Grad $d^x = 1$ verwendet. Da es sich bei x um eine lineare Funktion handelt, können die Daten auf diese Weise exakt approximiert werden. Um die Daten der Steuerungen zu approximieren wird ein ν -reguläres stückweises Polynom \tilde{u}_{app} vom Grad $d^u = 0$ verwendet. Die konstante Steuerung kann damit exakt approximiert werden. Als Basisfunktionen für \tilde{x}_{app} und \tilde{u}_{app} werden B-Splines mit den Knotenfolgen $\tau^x = (0, 0, 1, 1)$ bzw. $\tau^u = (0, 1)$ verwendet. Folglich gilt

$$\tilde{x}_{\text{app}}(t) = c_0^x B_{0,1,\tau^x}(t) + c_1^x B_{1,1,\tau^x}(t) = c_0^x(1-t) + c_1^x t \quad (6.3)$$

und

$$\tilde{u}_{\text{app}}(t) = c_0^u B_{0,0,\tau^u}(t) = c_0^u. \quad (6.4)$$

Durch die Wahl von $c_0^{x,0} = 0$, $c_1^{x,0} = 1$ und $c_0^{u,0} = 0.5$ als Koeffizienten werden die Daten exakt approximiert. Das Lösen der Optimierungsprobleme (5.3) und (5.4) ist folglich nicht notwendig. In [Abbildung 6.2](#) sind Approximation der Daten dargestellt. Der linke Plot zeigt die Approximation der Daten des Zustands und der rechte Plot die Approximation der Daten der Steuerung. Es ist zu erkennen, dass die Datenpunkte exakt durch die \tilde{x}_{app} bzw. \tilde{u}_{app} approximiert werden.

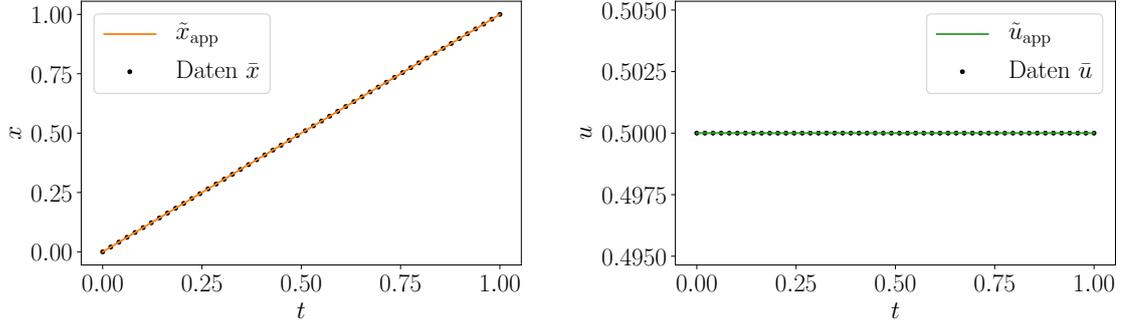


Abbildung 6.2: Daten des Zustands x und der Steuerung u im einfachen dynamischen Modell (6.1) mit $p = 2$ und $u(t) = \hat{u} = 0.5$. Zudem sind die Approximationen \tilde{x}_{app} und \tilde{u}_{app} der Daten durch ν -reguläre stückweise Polynome gemäß (6.3) und (6.4) dargestellt

6.2.2 Identifikation des Modellparameters und Berechnung der Sensitivitäten

Die Approximationen der Daten werden genutzt, um das dynamische Parameteridentifikationsproblem (5.5) mit der Kollokationsmethode zu lösen. Zum Bestimmen des Modellparameters wird das Optimierungsproblem (5.6) gelöst. Anschließend werden die Sensitivitäten des Modellparameters bezüglich der Koeffizienten der B-Splines berechnet. Dies kann für das einfache Beispiel analytisch geschehen. Die Rechnung ist in [Anhang B.1](#) ausgeführt. Dabei wird der Wert $p^* = 2$ bestätigt und für die Werte der Sensitivitäten ergeben sich

$$\frac{dp}{dc_0^x}(p^*, c^{x,0}) = -2, \quad \frac{dp}{dc_1^x}(p^*, c^{x,0}) = 2 \quad \text{und} \quad \frac{dp}{dc_0^u}(p^*, c^{u,0}) = -4. \quad (6.5)$$

Um den Einfluss von Daten $\mathcal{D}^x \subset \{\bar{x}_1, \dots, \bar{x}_{n_t}\}$ und $\mathcal{D}^u \subset \{\bar{u}_1, \dots, \bar{u}_{n_t}\}$ auf den Modellparameter p zu beurteilen, werden im nächsten Abschnitt die Sensitivitäten der Koeffizienten der B-Splines bezüglich der Datenpunkte berechnet (vgl. [Abschnitt 5.2.3](#)). Für ein besseres Verständnis der Methode wird jedoch zunächst die Bedeutung der Sensitivitätsdifferentialen (6.5) erläutert.

Mithilfe der Sensitivitäten (6.5) kann die Änderung des Modellparameters Δp berechnet werden, wenn die Koeffizienten der B-Splines gestört werden. Um die Überlegungen zu vereinfachen, wird zunächst angenommen, dass lediglich die Koeffizienten der Approximation des Zustands $c_0^{x,0}$ und $c_1^{x,0}$ variieren. Der Wert der Steuerung \hat{u} ändert sich folglich nicht und die Änderung der Steigung von x ist durch

$$\Delta m_x := \Delta p \hat{u} \quad (6.6)$$

gegeben. Die Steigung von \tilde{x}_{app} entspricht nach (6.3) der Differenz $c_1^{x,0} - c_0^{x,0}$. Stö-

rungen der Koeffizienten um Δc_0^x und Δc_1^x führen folglich zu einer Änderung

$$\Delta m_{\tilde{x}_{\text{app}}} := \Delta c_1^x - \Delta c_0^x \quad (6.7)$$

der Steigung von \tilde{x}_{app} . Da die Approximation \tilde{x}_{app} exakt ist und unverrauschte Daten betrachtet werden, gilt $\Delta m_x = \Delta m_{\tilde{x}_{\text{app}}}$ und aus (6.6) und (6.7) folgt die Änderung des Modellparameters

$$\Delta p = \frac{\Delta c_1^x - \Delta c_0^x}{\hat{u}} = 2(\Delta c_1^x - \Delta c_0^x).$$

Betrachten der Sensitivitäten (6.5) führt auf dasselbe Resultat. Nach (5.9) kann die Änderung des Modellparameters durch

$$\Delta p = \frac{dp}{dc_0^x}(p^*, c^{x,0}) \Delta c_0^x + \frac{dp}{dc_1^x}(p^*, c^{x,0}) \Delta c_1^x = 2(\Delta c_1^x - \Delta c_0^x). \quad (6.8)$$

berechnet werden.

Als nächstes wird angenommen, dass sich lediglich der Koeffizient der Approximation der Steuerung $c_0^{u,0}$ um Δc_0^u ändert. Die Daten der Zustände bleiben folglich konstant. Demzufolge ist auch die Steigung von x konstant. Da \tilde{u}_{app} die Daten der Steuerung exakt approximiert und unverrauschte Daten betrachtet werden, gilt $\hat{u} = c_0^{u,0}$. Folglich ist

$$p^* c_0^{u,0} = (p^* + \Delta p)(c_0^{u,0} + \Delta c_0^u) = p^* c_0^{u,0} + p^* \Delta c_0^u + \Delta p(c_0^{u,0} + \Delta c_0^u)$$

und damit

$$\Delta p = \frac{-p^* \Delta c_0^u}{c_0^{u,0} + \Delta c_0^u} = -p^* \frac{\Delta c_0^u}{c_0^{u,0}} + \mathcal{O}((\Delta c_0^u)^2).$$

Dabei folgt letztere Gleichheit aus einer Taylor-Entwicklung um $\Delta c_0^u = 0$. In linearer Näherung gilt also

$$\Delta p = -p^* \frac{\Delta c_0^u}{c_0^{u,0}} = -4\Delta c_0^u.$$

Betrachten der Sensitivitäten (6.5) führt auf dasselbe Resultat. Nach (5.9) ergibt sich

$$\Delta p = \frac{dp}{dc_0^u}(p^*, c^{u,0}) \Delta c_0^u = -4\Delta c_0^u. \quad (6.9)$$

Auf diese Weise kann der Einfluss der Koeffizienten der Approximationen \tilde{x}_{app} und \tilde{u}_{app} auf den Modellparameter p ermittelt werden. Es ist jedoch nicht ohne Weiteres

möglich, aus den Sensitivitäten (6.5) den Einfluss von beliebigen Bereichen in den Daten auf den Modellparameter zu bestimmen. Dies gelingt, wie in [Abschnitt 5.2.3](#) dargestellt, durch eine Interpretation der Sensitivitäten (6.5).

6.2.3 Interpretation der Sensitivitäten

Um den Einfluss beliebiger Daten $\mathcal{D}^x \subset \{\bar{x}_1, \dots, \bar{x}_{n_t}\}$ und $\mathcal{D}^u \subset \{\bar{u}_1, \dots, \bar{u}_{n_t}\}$ auf den Modellparameter p zu beurteilen, werden die Sensitivitäten

$$\frac{dc_0^x}{d\bar{x}_k}(c^{x,0}, \bar{x}^0), \frac{dc_1^x}{d\bar{x}_k}(c^{x,0}, \bar{x}^0) \quad \text{und} \quad \frac{dc_0^u}{d\bar{u}_k}(c^{u,0}, \bar{u}^0) \quad \text{für} \quad k = 1, \dots, n_t \quad (6.10)$$

bestimmt. Dies geschieht durch Lösen der linearen Gleichungssysteme (5.16). Für das einfache Beispiel können die Gleichungssysteme analytisch gelöst werden. In [Anhang B.2](#) ist die Rechnung ausgeführt. Mithilfe der Sensitivitäten (6.10) kann der Einfluss von \mathcal{D}^x und \mathcal{D}^u auf den Modellparameter p bestimmt werden.

In [Abschnitt 6.2.3.1](#) wird der Einfluss der Daten mithilfe der Änderung des Modellparameters, die sich bei Störungen der Datenpunkte $\bar{x}_k \in \mathcal{D}^x$ und $\bar{u}_k \in \mathcal{D}^u$ um $\Delta\bar{x}_k$ bzw. $\Delta\bar{u}_k$ ergibt, bewertet. Die Änderung des Modellparameters kann mithilfe von (5.12) berechnet werden. In (5.12) kann es zur Auslöschung von Termen mit unterschiedlichen Vorzeichen kommen. In [Abschnitt 5.2.4.2](#) wurde deshalb als Alternative der mittlere absolute Dateneinfluss eingeführt, um den Einfluss der Daten \mathcal{D}^x und \mathcal{D}^u zu bewerten. In [Abschnitt 6.2.3.2](#) werden Auswertungen zu diesem Ansatz vorgestellt.

6.2.3.1 Analyse der Änderung des Modellparameters

Zunächst wird die Änderung des Modellparameters $\Delta p(\mathcal{D})$ gemäß (5.12) zum Beurteilen des Einflusses der Daten herangezogen. Es wird untersucht, wie sich Störungen $\Delta\bar{x}_k = 0.1$ und $\Delta\bar{u}_k = 0.1$ der Datenpunkte auf den Modellparameter auswirken. Die Größen der Störungen sind willkürlich gewählt. Es ist zu beachten, dass die Daten nicht zu stark gestört werden, wenn durch $\Delta p(\mathcal{D})$ die tatsächliche Änderung des Modellparameters approximiert werden soll. Da $\Delta p(\mathcal{D})$ durch (5.12) lediglich näherungsweise approximiert wird, kann der Fehler bei starken Störungen der Daten groß sein. Meist ist die tatsächliche Änderung des Modellparameters jedoch nicht von Interesse. Um zu beurteilen, welcher der Datenbereiche \mathcal{D}_1 und \mathcal{D}_2 einen größeren Einfluss auf den Modellparameter hat, ist lediglich wichtig, ob eine Störung der Datenpunkte in \mathcal{D}_1 oder in \mathcal{D}_2 eine größere Änderung des Modellparameters verursacht.

Die in [Kapitel 5](#) entwickelte Methode zeichnet sich dadurch aus, dass der Einfluss beliebiger Bereiche der Daten auf den Modellparameter beurteilt werden kann. Im Folgenden wird daher der Einfluss verschiedener Aufteilungen der Daten untersucht.

Fall 1: Alle Datenpunkte Als erstes werden die gesamten Datenpunkte betrachtet, d. h. $\mathcal{D}^x = \{\bar{x}_1, \dots, \bar{x}_{50}\}$ und $\mathcal{D}^u = \{\bar{u}_1, \dots, \bar{u}_{50}\}$. Es wird die Änderung des Modellparameters berechnet, wenn alle Daten um $\Delta\bar{x}_k = 0.1$ und $\Delta\bar{u}_k = 0.1$ gestört werden. Das Ergebnis ist in [Abbildung 6.3a](#) dargestellt. Es sind zwei Plots abgebildet. Der linke Plot zeigt das Ergebnis zu den Daten des Zustands und der rechte Plot das Ergebnis zu den Daten der Steuerung. In den Plots sind zum einen die Datenpunkte abgebildet. Zum anderen ist der Einfluss der Datenbereiche \mathcal{D}^x und \mathcal{D}^u dargestellt. Wie in [Abschnitt 6.1](#) erläutert, werden Rechtecke verwendet um den Einfluss zu visualisieren. Die Breite eines Rechtecks entspricht der zeitlichen Ausdehnung der Daten \mathcal{D}^x bzw. \mathcal{D}^u . Die Höhe eines Rechtecks ist die Größe der Änderung des Modellparameters $\Delta p(\mathcal{D}^x)$ bzw. $\Delta p(\mathcal{D}^u)$. Die Werte von $\Delta p(\mathcal{D}^x)$ und $\Delta p(\mathcal{D}^u)$ sind auf der rechten vertikalen Koordinatenachse abzulesen. In diesem Fall umfassen \mathcal{D}^x und \mathcal{D}^u alle Daten. Folglich ist lediglich ein Rechteck zu sehen, welches sich über das gesamte Zeitintervall $[0, 1]$ erstreckt.

Für den Zustand x ist zu erkennen, dass eine Störung der Daten \mathcal{D}^x keine Änderung des Modellparameters verursacht. Dies ist plausibel, da eine Störung aller Datenpunkte lediglich eine vertikale Verschiebung von x bedeutet. Diese Verschiebung hat jedoch keinen Einfluss auf die Steigung der Funktion und beeinflusst folglich auch nicht den Modellparameter p . Das Resultat deckt sich auch mit den Überlegungen des vorherigen Abschnitts zum Einfluss der Koeffizienten der B-Splines auf den Modellparameter. Eine Störung aller Datenpunkte um $\Delta\bar{x}_k = 0.1$ würde einer Änderung $\Delta c_1^x = \Delta c_2^x = 0.1$ der Koeffizienten der B-Splines entsprechen. Aus [\(6.8\)](#) folgt damit ebenfalls $\Delta p = 0$.

Im rechten Plot in [Abbildung 6.3a](#) ist zu erkennen, dass eine Störung der Daten \mathcal{D}^u um $\Delta\bar{u}_k = 0.1$ eine Änderung $\Delta p(\mathcal{D}^u) = -0.4$ des Modellparameters nach sich zieht. Dies ist plausibel und kann mit den Überlegungen des vorherigen Abschnitts begründet werden. Werden alle Datenpunkte um $\Delta\bar{u}_k = 0.1$ gestört, so entspricht dies einer Änderung des Koeffizienten c_0^u um $\Delta c_0^u = 0.1$. Aus [\(6.9\)](#) folgt damit ebenfalls $\Delta p = -0.4$. Der Einfluss aller Daten auf den Modellparameter hätte folglich auch mithilfe der Sensitivitäten [\(6.5\)](#) und den Formeln [\(6.8\)](#) und [\(6.9\)](#) bestimmt werden können.

Fall 2: Zwei gleich große Datenbereiche Als nächstes werden die Daten in zwei gleich große, zeitlich zusammenhängende Bereiche aufgeteilt:

$$\mathcal{D}_1^x = \{\bar{x}_1, \dots, \bar{x}_{25}\}, \quad \mathcal{D}_2^x = \{\bar{x}_{26}, \dots, \bar{x}_{50}\} \quad (6.11)$$

und

$$\mathcal{D}_1^u = \{\bar{u}_1, \dots, \bar{u}_{25}\}, \quad \mathcal{D}_2^u = \{\bar{u}_{26}, \dots, \bar{u}_{50}\}. \quad (6.12)$$

Die Datenpunkte werden um $\Delta\bar{x}_k = \Delta\bar{u}_k = 0.1$ gestört. Das Ergebnis ist in [Abbildung 6.3b](#) dargestellt. Der Einfluss der Daten \mathcal{D}_1^x und \mathcal{D}_2^x sowie \mathcal{D}_1^u und \mathcal{D}_2^u ist durch

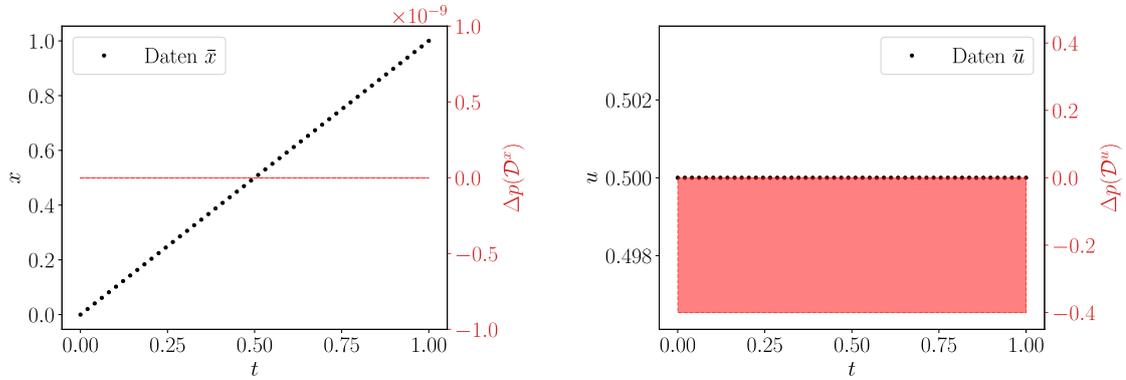
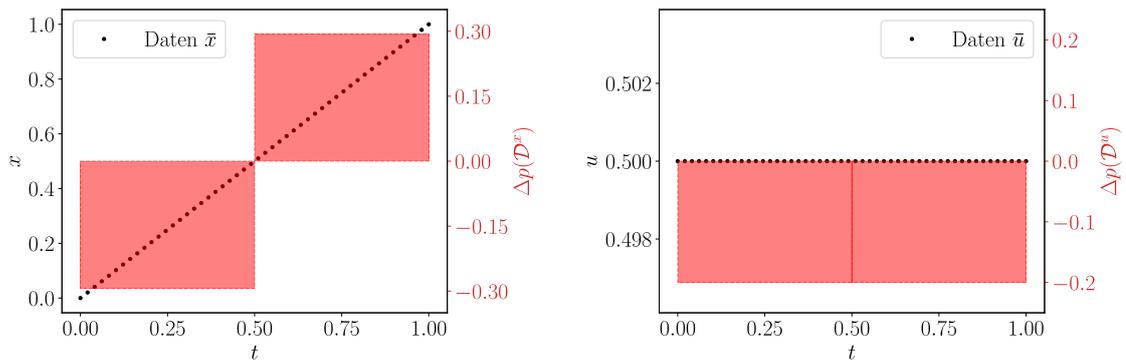
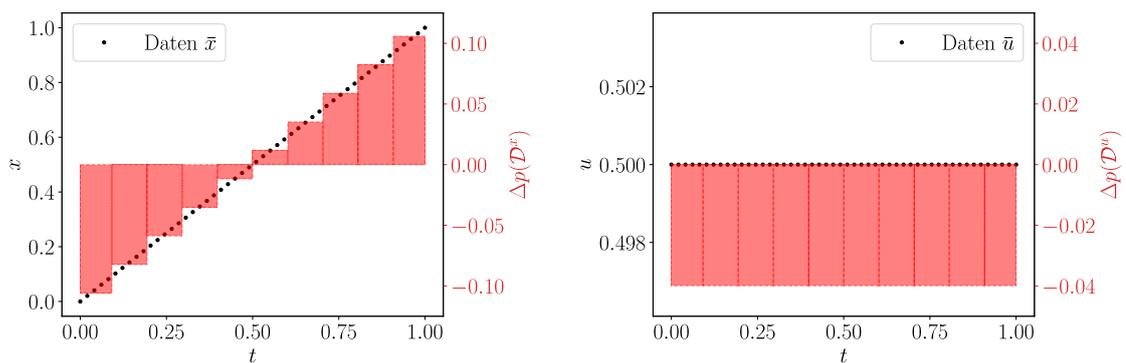
(a) Alle Datenpunkte: $\mathcal{D}^x = \{\bar{x}_1, \dots, \bar{x}_{50}\}$ und $\mathcal{D}^u = \{\bar{u}_1, \dots, \bar{u}_{50}\}$ (b) Zwei gleich große Datenbereiche \mathcal{D}_i^x und \mathcal{D}_i^u für $i = 1, 2$ gemäß (6.11) und (6.12)(c) Zehn gleich große Datenbereiche \mathcal{D}_i^x und \mathcal{D}_i^u für $i = 1, \dots, 10$ gemäß (6.13) und (6.14)

Abbildung 6.3: Änderung des Modellparameters p des einfachen dynamischen Modells bei einer Störung der Daten um $\Delta\bar{x}_k = \Delta\bar{u}_k = 0.1$

je zwei rote Rechtecke visualisiert. Da die Daten zeitlich äquidistant vorliegen und in je zwei gleich große, zusammenhängende Bereiche aufgeteilt werden, erstrecken sich die Rechtecke über die Intervalle $[t_0, t_{25}] \approx [0, 0.5]$ und $[t_{26}, t_{50}] \approx [0.5, 1]$. Es ist zu erkennen, dass eine Störung der Datenpunkte in \mathcal{D}_1^x eine Änderung $\Delta p(\mathcal{D}_1^x) \approx -0.3$ des Modellparameters versucht. Für die Daten \mathcal{D}_2^x kann $\Delta p(\mathcal{D}_2^x) \approx 0.3$ abgelesen werden. Eine Störung der Daten im Intervall $[0, 0.5]$ würde für eine größere mittlere Steigung sorgen und folglich ist Δp negativ. Umgekehrt würde eine Störung der Daten im Intervall $[0.5, 1]$ in einer größeren mittleren Steigung resultieren und deshalb ist Δp positiv.

Für die Daten der Steuerung \mathcal{D}_1^u und \mathcal{D}_2^u ergibt sich bei einer Störung um $\Delta \bar{u}_k = 0.1$ eine Änderung des Modellparameters um $\Delta p(\mathcal{D}_1^u) = \Delta p(\mathcal{D}_2^u) = -0.2$. Der Einfluss der Hälfte der Datenpunkte ist folglich halb so groß wie der Einfluss der gesamten Daten (vgl. Abb. 6.3a). Dies ist plausibel: Da die Steuerung konstant ist, hat jeder Datenpunkt den gleichen Einfluss auf die Approximation der Daten und damit auf den Modellparameter.

Fall 3: Zehn gleich große Datenbereiche Als drittes werden die Daten in zehn gleich große, zeitlich zusammenhängende Bereiche aufgeteilt. Es sei

$$\mathcal{D}_i^x = \{\bar{x}_j \mid 1 + 5(i-1) \leq j \leq 5i\} \quad \text{für } i = 1, \dots, 10 \quad (6.13)$$

und

$$\mathcal{D}_i^u = \{\bar{u}_j \mid 1 + 5(i-1) \leq j \leq 5i\} \quad \text{für } i = 1, \dots, 10. \quad (6.14)$$

Die Datenpunkte werden um $\Delta \bar{x}_k = \Delta \bar{u}_k = 0.1$ gestört. Der Einfluss der Datenbereiche ist in [Abbildung 6.3c](#) dargestellt.

In [Abbildung 6.3c](#) ist Folgendes zu erkennen: Je näher die Datenbereiche am Rand des Zeitintervalls $[0, 1]$ liegen, desto stärker ist ihr Einfluss auf den Modellparameter p . Dies lässt sich damit erklären, dass eine Störung der Daten am Rand des Zeitintervalls zu einer größeren Asymmetrie in den Daten führt als eine Störung der Daten in der Mitte des Zeitintervalls. Deshalb würde sich die Steigung von x bei einer Störung der Daten am Rand stärker ändern. Wie bereits bei der Aufteilung der Daten in zwei Bereiche ist zu erkennen, dass eine Störung der Daten des Zustands im Intervall $[0, 0.5]$ zu einer Abnahme des Modellparameters führt, während sie im Intervall $[0.5, 1]$ in einer Zunahme des Modellparameters resultiert.

Für die Daten der Steuerung wird beobachtet, dass der Einfluss eines einzelnen Bereiches abnimmt. Für jeden Bereich \mathcal{D}_i^u beträgt die Änderung im Modellparameter $\Delta p(\mathcal{D}_i^u) = -0.04$. Dies ist ein Zehntel im Vergleich zum Einfluss der gesamten Daten und erscheint sinnvoll, da die Bereiche \mathcal{D}_i^u je ein Zehntel aller Datenpunkte beinhalten.

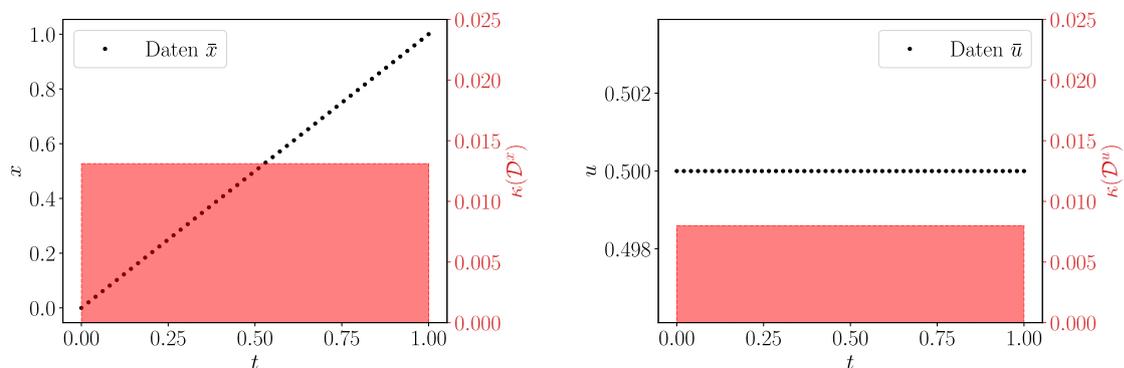
6.2.3.2 Analyse des mittleren absoluten Dateneinflusses

Bei der Bewertung des Einflusses der Daten mithilfe der Änderung im Modellparameter kann es zur Auslöschung von Termen mit unterschiedlichen Vorzeichen kommen. Dies wurde in Fall 1, in welchem alle Datenpunkte gemeinsam betrachtet wurden, beobachtet (vgl. [Abschnitt 6.2.3.1](#)). Werden alle Datenpunkte gleichzeitig gestört, so hebt sich der Einfluss der Daten des Zustands auf den Modellparameter auf und es ist $\Delta p(\mathcal{D}^x) = 0$. Werden die Daten wie in Fall 2 und Fall 3 in mehrere Bereiche aufgeteilt, so wird eine Änderung des Modellparameters festgestellt.

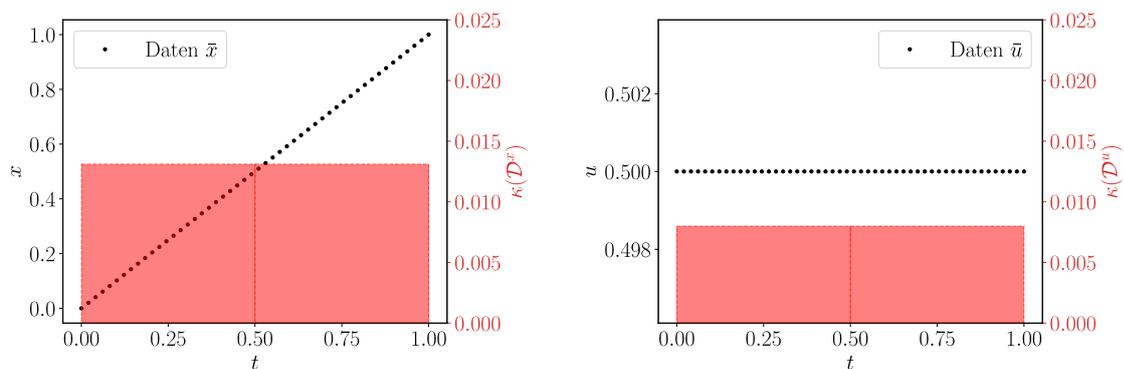
Im Folgenden wird der Einfluss der Datenbereiche mithilfe des in [Abschnitt 5.2.4.2](#) vorgestellten mittleren absoluten Dateneinflusses untersucht. Dieser wird durch (5.17) berechnet und mit $\kappa(\mathcal{D}^x)$ bzw. $\kappa(\mathcal{D}^u)$ bezeichnet. In (5.17) werden die Beträge der Sensitivitäten summiert. Dadurch kommt es zu keiner Auslöschung von Termen mit unterschiedlichen Vorzeichen. Die Werte $\kappa(\mathcal{D}^x)$ bzw. $\kappa(\mathcal{D}^u)$ können jedoch nicht mehr als explizite Änderung des Modellparameters interpretiert werden und sollten in Relation zueinander betrachtet werden. Es werden dieselben Aufteilungen wie in [Abschnitt 6.2.3.1](#) betrachtet.

Fall 1: Alle Datenpunkte Als erstes werden die gesamten Datenpunkte betrachtet, d. h. $\mathcal{D}^x = \{\bar{x}_1, \dots, \bar{x}_{50}\}$ und $\mathcal{D}^u = \{\bar{u}_1, \dots, \bar{u}_{50}\}$. Mithilfe von (5.17) wird der mittlere absolute Dateneinfluss berechnet. Dabei wird wie in [Abschnitt 6.2.3.1](#) eine Störung der Daten um $\Delta \bar{x}_k = \Delta \bar{u}_k = 0.1$ angenommen. Das Ergebnis ist in [Abbildung 6.4a](#) dargestellt. Es werden, wie in [Abschnitt 6.1](#) erläutert, Rechtecke zur Visualisierung des Einflusses verwendet. Die Höhe der Rechtecke entspricht den Werten $\kappa(\mathcal{D}^x)$ und $\kappa(\mathcal{D}^u)$ und kann auf der rechten vertikalen Koordinatenachse abgelesen werden: Es ist $\kappa(\mathcal{D}^x) \approx 0.013$ und $\kappa(\mathcal{D}^u) \approx 0.008$. Der mittlere absolute Einfluss der Daten der Zustände auf den Modellparameter ist folglich etwas größer als der Einfluss der Daten der Steuerungen. Insbesondere fällt auf, dass der mittlere absolute Dateneinfluss von \mathcal{D}^x im Gegensatz zur Beurteilung des Einflusses mithilfe der Änderung des Modellparameters nicht Null ist (vgl. [Abschnitt 6.2.3.1](#), Fall 1).

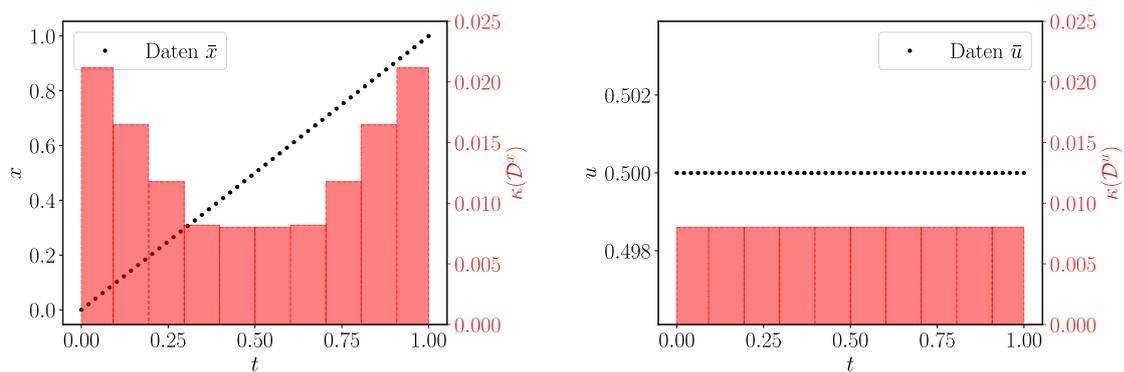
Fall 2: Zwei gleich große Datenbereiche Als zweites werden die Daten wie in [Abschnitt 6.2.3.1](#) gemäß (6.11) und (6.12) in zwei gleich große Bereiche aufgeteilt. In [Abbildung 6.4b](#) ist der mittlere absolute Einfluss der Datenbereiche dargestellt. Es ergeben sich die gleichen Werte wie in Fall 1: $\kappa(\mathcal{D}_1^x) = \kappa(\mathcal{D}_2^x) \approx 0.013$ und $\kappa(\mathcal{D}_1^u) = \kappa(\mathcal{D}_2^u) \approx 0.008$. Im Gegensatz zur Beurteilung des Einflusses mithilfe der Änderung des Modellparameters bleibt der mittlere Einfluss der Bereiche \mathcal{D}_1^u und \mathcal{D}_2^u im Vergleich zu Fall 1 konstant. Dies kommt zustande, weil in (5.17) durch die Anzahl der Datenpunkte geteilt wird.



(a) Alle Datenpunkte: $\mathcal{D}^x = \{\bar{x}_1, \dots, \bar{x}_{50}\}$ und $\mathcal{D}^u = \{\bar{u}_1, \dots, \bar{u}_{50}\}$



(b) Zwei gleich große Datenbereiche \mathcal{D}_i^x und \mathcal{D}_i^u für $i = 1, 2$ gemäß (6.11) und (6.12)



(c) Zehn gleich große Datenbereiche \mathcal{D}_i^x und \mathcal{D}_i^u für $i = 1, \dots, 10$ gemäß (6.13) und (6.14)

Abbildung 6.4: Mittlerer absoluter Einfluss verschiedener Datenbereiche auf den Modellparameter p des einfachen dynamischen Modells bei einer Störung der Daten um $\Delta\bar{x}_k = \Delta\bar{u}_k = 0.1$

Fall 3: Zehn gleich große Datenbereiche Als drittes werden die Daten wie in [Abschnitt 6.2.3.1](#) gemäß (6.13) und (6.14) in zehn gleich große Bereiche aufgeteilt. Der mittlere absolute Einfluss dieser Datenbereiche ist in [Abbildung 6.4c](#) dargestellt. Für die Daten der Zustände kann zumindest qualitativ das gleiche Ergebnis wie bei der Beurteilung des Einflusses mithilfe der Änderung des Modellparameters erzielt werden: Je näher die Bereiche am Rand des Zeitintervalls liegen, desto größer ist ihr Einfluss auf den Modellparameter (vgl. [Abschnitt 6.2.3.1](#), Fall 3). Für die Daten der Steuerungen ergibt sich wie in den Fällen 1 und 2 für jeden Bereich der Wert $\kappa(\mathcal{D}_i^u) \approx 0.008$.

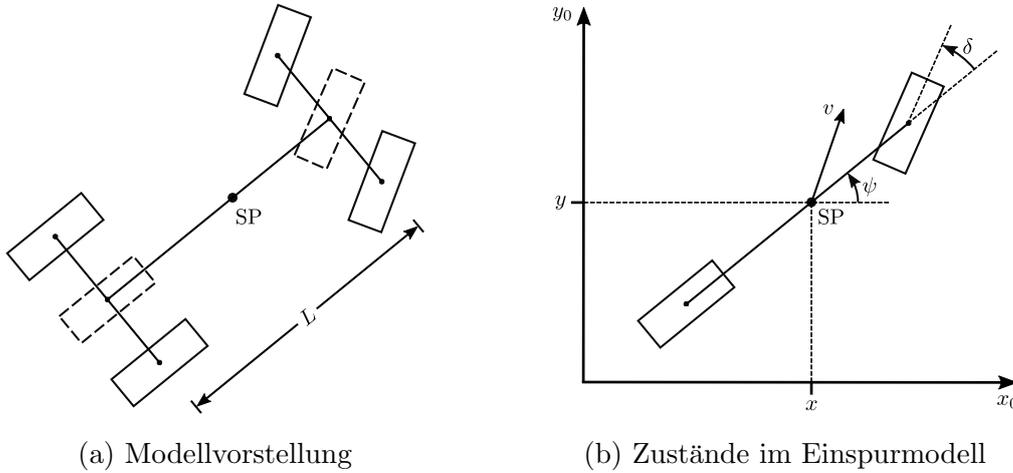
Zusammenfassend lässt sich sagen, dass die in [Kapitel 5](#) entwickelte Methode geeignet ist, um den Einfluss von Daten auf den Modellparameter das einfache dynamische Modell (6.1) zu analysieren. Der Einfluss kann sowohl mithilfe der Änderung des Modellparameters (5.12) als auch durch den mittleren absoluten Dateneinfluss (5.17) beurteilt werden. Welche der beiden Bewertungsansätze genutzt wird, ist im konkreten Anwendungsfall zu entscheiden. Im nächsten Abschnitt wird die Methode auf ein komplexeres dynamisches Modell angewendet.

6.3 Analyse des Einflusses von Daten auf die Parameter eines Einspurmodells

In diesem Abschnitt wird der Einfluss von Daten auf die Modellparameter eines sogenannten Einspurmodells analysiert. Einspurmodelle beschreiben die Bewegung von zweiachsigen Fahrzeugen. In [Abbildung 6.5a](#) ist das ebene Modell eines zweiachsigen Fahrzeugs abgebildet. Es sind die beiden Achsen mit den Vorder- und Hinterrädern dargestellt. Darüber hinaus ist der Schwerpunkt (SP) des Fahrzeugs sowie der Radstand L gekennzeichnet.

Einspurmodelle beruhen auf verschiedenen vereinfachenden Annahmen, um die Bewegung des Fahrzeugs zu modellieren. So werden etwa die Räder der Vorder- und Hinterachse durch je ein Einzelrad ersetzt und die Bewegung wird auf die Bewegung des Schwerpunkts (vgl. [Abb. 6.5a](#)). Da der Fokus dieser Arbeit nicht auf der Modellierung der Bewegung von Fahrzeugen liegt, werden die weiteren Annahmen des Einspurmodells nicht weiter diskutiert. Für eine ausführliche Behandlung der Thematik sei auf Schramm et al. [66] verwiesen.

In [Abbildung 6.5b](#) sind die Zustände des in dieser Arbeit untersuchten Einspurmodells skizziert. Die Position des Schwerpunkts in einem Referenzkoordinatensystem (Koordinaten x_0, y_0) wird durch die Zustände x und y beschrieben. Der Zustand v bezeichnet die Geschwindigkeit im Schwerpunkt. Die Orientierung des Fahrzeugs im Referenzkoordinatensystem wird mit ψ bezeichnet. Der Winkel δ beschreibt den Lenkwinkel des Fahrzeugs. Durch Beschleunigen bzw. Bremsen und Ändern des



(a) Modellvorstellung

(b) Zustände im Einspurmodell

Abbildung 6.5: Schematische Darstellung des Einspurmodells

Lenkwinkels kann die Bewegung des Fahrzeugs beeinflusst werden. Im hier untersuchten dynamischen Modell wird diese Aktuatorik als Steuerung gemäß [Definition 4.3](#) aufgefasst. Die Beschleunigung wird mit a und die Lenkwinkelgeschwindigkeit mit σ bezeichnet. Negative Werte für a entsprechen einem Bremsen des Fahrzeugs. In [Tabelle 6.2](#) sind die Zustände und Steuerungen des Einspurmodells sowie deren Einheiten zusammengefasst.

Das dynamische Modell

$$\dot{x}(t) = v(t) \cos \psi(t) \quad (6.15a)$$

$$\dot{y}(t) = v(t) \sin \psi(t) \quad (6.15b)$$

$$\dot{v}(t) = ka(t) \quad (6.15c)$$

$$\dot{\psi}(t) = \frac{v(t)}{L} \tan \delta(t) \quad (6.15d)$$

$$\dot{\delta}(t) = r\sigma(t) \quad (6.15e)$$

für $t \in [t_0, t_f]$ kann zur Beschreibung der Zustandsänderungen im Einspurmodell hergeleitet werden [\[59\]](#). Im Folgenden wird [\(6.15\)](#) als Einspurmodell bezeichnet. Das Modell hängt von den Parametern $k, L, r > 0$ ab. Dabei ist L der Radstand. Die Parameter k und r modellieren Verluste bzw. Verstärkungen der Steuerungen. Ist $k < 1$, so ist tatsächliche Beschleunigung \dot{v} des Fahrzeugs um den Faktor k kleiner als die durch die Steuerung a vorgegebene Beschleunigung. Analog bedeutet ein Wert $r < 1$, dass die tatsächliche Lenkwinkelgeschwindigkeit $\dot{\delta}$ um den Faktor r kleiner ist als die durch die Steuerung σ vorgegebene Lenkwinkelgeschwindigkeit.

Im Folgenden wird anhand eines beispielhaften Fahrmanövers untersucht, wie der Einfluss von Daten auf die Modellparameter k , L und r ist.

Tabelle 6.2: Zustände und Steuerungen des Einspurmodells

Zustand	Bezeichnung	Einheit
x -Position des Schwerpunkts im Referenzsystem	x	m
y -Position des Schwerpunkts im Referenzsystem	y	m
Geschwindigkeit des Schwerpunkts	v	m/s
Orientierung im Referenzsystem	ψ	rad
Lenkwinkel	δ	rad
Steuerung	Bezeichnung	Einheit
Beschleunigung des Schwerpunkts	a	m/s ²
Lenkwinkelgeschwindigkeit	σ	rad/s

6.3.1 Generieren von Daten

Der Einfluss von Daten auf die Modellparameter des Einspurmodells (6.15) wird anhand des folgenden Fahrmanövers untersucht. Das Fahrzeug startet im Ursprung des Referenzkoordinatensystems aus der Ruhelage. Es ist parallel zur y_0 -Achse ausgerichtet und kein Lenkwinkel eingestellt. Als Startzeit wird $t_0 = 0$ s gewählt. Daraus ergeben sich die Anfangswerte

$$x(0) = y(0) = 0 \text{ m}, \quad v(0) = 0 \text{ m/s}, \quad \psi(0) = \frac{\pi}{2} \text{ rad} \quad \text{und} \quad \delta(0) = 0 \text{ rad}. \quad (6.16)$$

Zunächst erfolgt eine gleichmäßig beschleunigte Geradeausfahrt des Fahrzeugs für fünf Sekunden. Anschließend wird bei konstanter Geschwindigkeit eine 90 Grad Rechtskurve gefahren. Nach der Kurve bremst das Auto gleichmäßig auf die Anfangsgeschwindigkeit 0 m/s herunter. Das Fahrmanöver ist in [Abbildung 6.6](#) dargestellt. Durch die schwarze Linie ist die Trajektorie des Fahrzeugs und durch die rote Linie die Geschwindigkeit während der Bewegung visualisiert. Der Wert der Geschwindigkeit kann auf der vertikalen Achse abgelesen werden. Durch die roten gestrichelten Linien ist der Beginn und das Ende der Kurve gekennzeichnet.

Für die Modellparameter werden die Werte

$$k_0 = 0.9, \quad L_0 = 2.8 \text{ m} \quad \text{und} \quad r_0 = 0.6 \quad (6.17)$$

angenommen. Die Steuerungen für dieses Fahrmanöver sind stückweise konstant und haben die maximalen Werte $a_{\max} = 1 \text{ m/s}^2$ und $\sigma_{\max} = 0.3 \text{ rad/s}$. Während Beschleunigungsphase fahre das Auto mit der maximalen Beschleunigung. Zu Beginn der Kurve hat das Auto deshalb eine Geschwindigkeit von 4.5 m/s . Die Lenkwinkelgeschwindigkeit sei eine Bang-Bang Steuerung. Aus den Werten von L_0 , r_0 und σ_{\max} ergibt sich, dass die Kurvenfahrt bei der konstanten Geschwindigkeit 4.5 m/s etwa

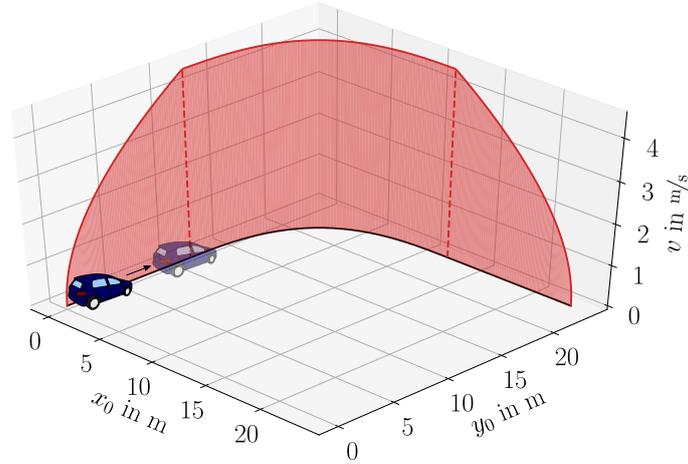


Abbildung 6.6: Trajektorie des Fahrzeugs im untersuchten Fahrmanöver

4.6 Sekunden dauert. Für die Steuerungen folgt daraus

$$a(t) = \begin{cases} 1 \text{ m/s}^2 & t \in [0.0 \text{ s}, 5.0 \text{ s}) \\ 0 \text{ m/s}^2 & t \in [5.0 \text{ s}, 9.6 \text{ s}) \\ 1 \text{ m/s}^2 & t \in [9.6 \text{ s}, 14.6 \text{ s}] \end{cases} \quad \text{und} \quad \sigma(t) = \begin{cases} 0.0 \text{ rad/s} & t \in [0.0 \text{ s}, 5.0 \text{ s}) \\ -0.3 \text{ rad/s} & t \in [5.0 \text{ s}, 7.3 \text{ s}) \\ 0.3 \text{ rad/s} & t \in [7.3 \text{ s}, 9.6 \text{ s}) \\ 0.0 \text{ rad/s} & t \in [9.6 \text{ s}, 14.6 \text{ s}] \end{cases}. \quad (6.18)$$

Durch eine numerische Simulation des Fahrmanövers werden Daten für die Zustände generiert. Dazu wird das Einspurmodell (6.15) mit den Anfangswerten (6.16), den Modellparametern (6.17) und den Steuerungen (6.18) numerisch gelöst. Zum Lösen wird die Funktion `scipy.integrate.odeint`² verwendet, welche das Programm *LSODA* [31] der Fortran Bibliothek *ODEPACK* [30] nutzt. Die Lösung wird an $n_t = 300$ äquidistanten Zeitpunkten $t_k \in [0 \text{ s}, 14.6 \text{ s}]$ mit $t_1 = 0 \text{ s}$ und $t_{300} = 14.6 \text{ s}$ approximiert und mit x_k, y_k, v_k, ψ_k und δ_k bezeichnet. Um Daten für die Steuerungen zu generieren, werden die Funktionen $a(t)$ und $\sigma(t)$ an den diskreten Zeitpunkten ausgewertet. Die Daten der Steuerungen werden mit a_k und σ_k bezeichnet.

Auf die generierten Daten wird ein normalverteilter Fehler addiert. Es sei

$$\bar{\chi}_k = \chi_k + \varepsilon_k^\chi \quad \text{mit} \quad \varepsilon_k^\chi \sim \mathcal{N}(\mu_\chi, \sigma_\chi^2) \quad \text{für} \quad \chi \in \{x, y, v, \psi, \delta, a, \sigma\}.$$

Die Werte der Erwartungswerte μ_χ und Standardabweichungen σ_χ sind in Tabelle 6.3 aufgelistet. Die verrauschten Daten sind in Abbildung 6.8 dargestellt. In der Abbildung sind außerdem die Approximationen der Daten abgebildet. Das Bestimmen der Approximationen wird im nächsten Abschnitt näher erläutert.

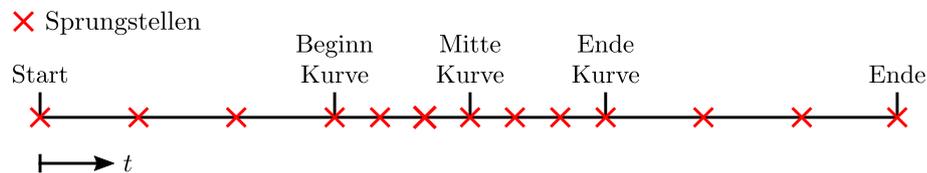
²<https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.odeint.html>

Tabelle 6.3: Mittelwerte und Standardabweichungen der Fehler der generierten Daten des Einspurmodells

Modellgröße χ	Mittelwert μ_χ	Standardabweichung σ_χ
x	0 m	1.00 m
y	0 m	1.00 m
v	0 m/s	0.20 m/s
ψ	0 rad	0.03 rad
δ	0 rad	0.03 rad
a	0 m/s ²	0.05 m/s ²
σ	0 rad/s	0.02 rad/s

6.3.2 Approximation der Daten

Im ersten Schritt der in [Kapitel 5](#) entwickelten Methode werden die Daten der Zustände und der Steuerungen durch ν -reguläre stückweise Polynome \tilde{x}_{app} , \tilde{y}_{app} , \tilde{v}_{app} , $\tilde{\psi}_{\text{app}}$, $\tilde{\delta}_{\text{app}}$, \tilde{a}_{app} und $\tilde{\sigma}_{\text{app}}$ approximiert. Für die Sprungstellen der stückweisen Polynome werden charakteristische Zeitpunkte des untersuchten Fahrmanövers gewählt. Dabei handelt es sich um den Start- und Endzeitpunkt der Simulation sowie den Zeitpunkten des Beginns, der Mitte und des Endes der Kurvenfahrt. Zwischen diesen Zeitpunkten werden zwei weitere Sprungstellen eingeführt. Die Sprungstellen sind in [Abbildung 6.7](#) dargestellt. Ihre exakten Werte sind in [Anhang C.1](#) aufgeführt.

Abbildung 6.7: Lage der Sprungstellen der ν -regulären stückweisen Polynome zur Approximation der Daten des Einspurmodells

Der Grad der stückweisen Polynome sowie die Regularität an den Sprungstellen werden für alle Zustände und Steuerungen separat gewählt. Um die Daten der x - und y -Positionen zu approximieren, werden stückweise kubische Polynome verwendet. Die Daten der Geschwindigkeit und des Lenkwinkels werden durch stückweise lineare Polynome approximiert. Die Approximation $\tilde{\psi}_{\text{app}}$ ist stückweise quadratisch und die Approximationen der beiden Steuerungen \tilde{a}_{app} und $\tilde{\sigma}_{\text{app}}$ sind stückweise konstant. Bei der Wahl der Regularitätsbedingungen der Approximationen an den Sprungstellen werden die Unstetigkeiten der Steuerungen berücksichtigt. Insbesondere wurde beachtet, dass die Zustände x und y zu den Zeitpunkten des Beginns und des Endes der Kurve unstetig in der zweiten Ableitung sind. Die Wahl der Regularitätsbedingungen an den Sprungstellen ist für alle Modellgrößen in [Anhang C.1](#) aufgeführt.

Als Basisfunktionen der ν -regulären stückweisen Polynome werden B-Splines verwendet. Die Knotenfolge der B-Splines kann nach [Satz 3.13](#) aus den Sprungstellen und den Regularitätsbedingungen an den Sprungstellen berechnet werden. Für die exakten Werte sei auf [Anhang C.1](#) verwiesen. Die Koeffizienten der B-Splines in der Basisdarstellung der Approximationen der Daten werden durch Lösen der Optimierungsprobleme (5.3) und (5.4) bestimmt. Das Lösen geschieht mit WORHP und benötigt nur wenige Millisekunden. Die optimalen Koeffizienten sind in [Anhang C.1](#) aufgelistet.

In [Abbildung 6.8](#) sind die Approximationen der Daten dargestellt. Die Datenpunkte sind durch schwarze Punkte visualisiert. [Abbildungen 6.8a bis 6.8e](#) zeigen die Approximationen der Daten der Zustände als blaue Linien. Es ist gut zu erkennen, dass die Daten der Zustände sinnvoll durch die stückweisen Polynome approximiert werden und das Rauschen reduziert wird. Insbesondere werden auch die „Knicke“ in den Daten der Geschwindigkeit und des Lenkwinkels abgebildet (s. [Abb. 6.8c](#) und [6.8e](#)). In den [Abbildungen 6.8f](#) und [6.8g](#) sind die Daten der Steuerungen sowie deren Approximationen durch orangefarbene Linien visualisiert. Die Daten werden gut durch die ν -regulären stückweisen Polynome angenähert und auch die Sprünge der Steuerungen werden durch die Approximationen repräsentiert.

6.3.3 Identifikation der Modellparameter und Berechnung der Sensitivitäten

Im nächsten Schritt der Methode zum Analysieren des Einflusses der Daten auf die Parameter des Einspurmodells (6.15) werden die Modellparameter identifiziert. Dies geschieht mittels der Kollokationsmethode. Dazu wird das Optimierungsproblem (5.6) mithilfe von WORHP gelöst. Die Rechenzeit beträgt etwa 0.05 Sekunden und die identifizierten Modellparameter haben die Werte

$$k^* = 0.89235568, \quad L^* = 2.8344354 \text{ m} \quad \text{und} \quad r^* = 0.60042114. \quad (6.19)$$

Der absolute Fehler zwischen den Referenzmodellparametern (6.17) und den identifizierten Modellparametern (6.19) hat die Größenordnung 10^{-2} . Da der Fokus dieser Arbeit nicht auf einer präzisen Parameteridentifikation sondern auf der Analyse des Einflusses der Daten auf die Modellparameter liegt, wird die Güte der identifizierten Modellparameter an dieser Stelle nicht weiter diskutiert.

Mithilfe von WORHP werden im Anschluss des Lösens von Problem (5.6) die Sensitivitäten der Modellparameter bezüglich der Koeffizienten der B-Splines berechnet. Im nächsten Abschnitt werden diese Sensitivitäten interpretiert, um den Einfluss der Daten auf die Modellparameter des Einspurmodells (6.15) zu analysieren.

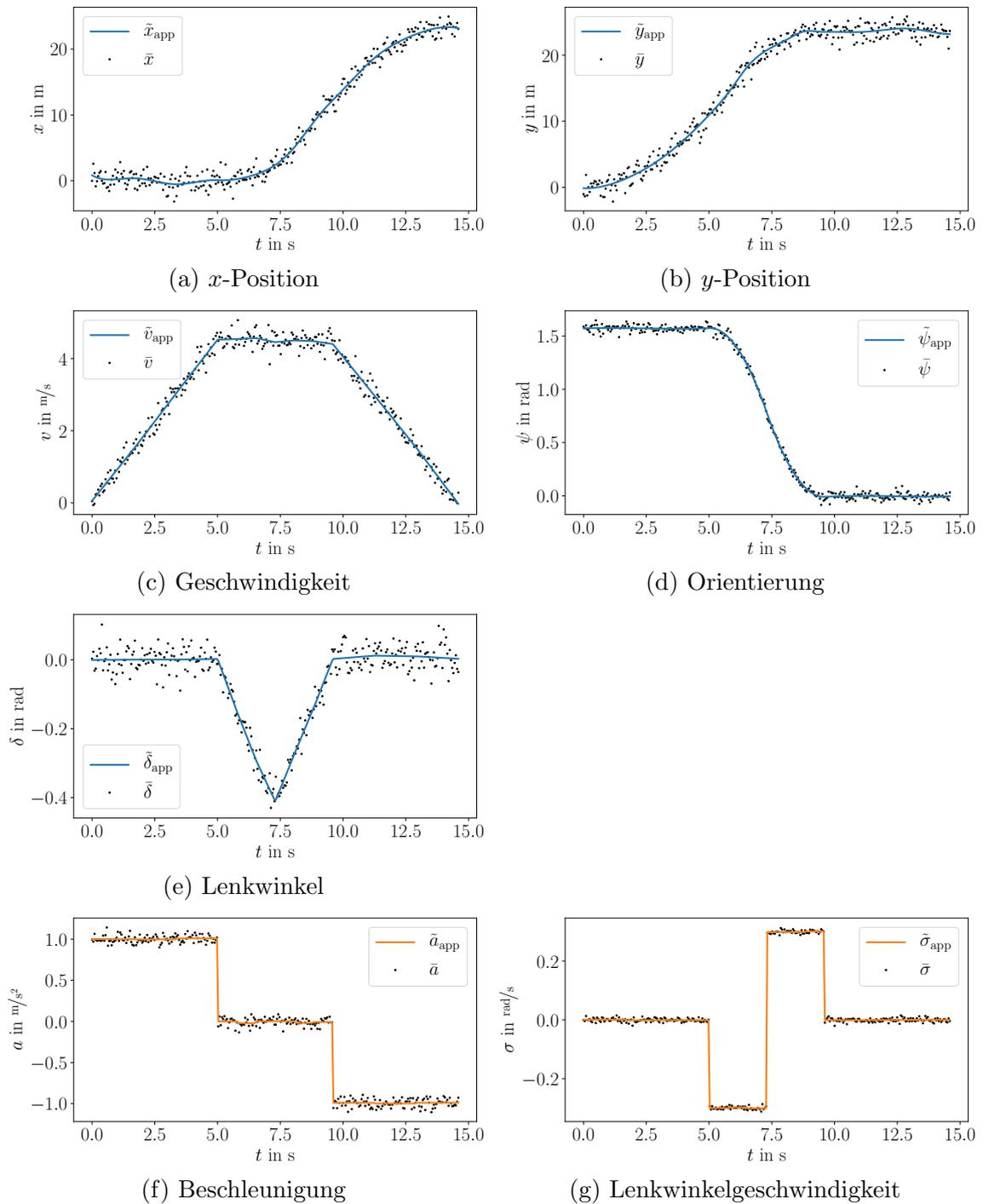


Abbildung 6.8: Daten und Approximationen der Daten durch ν -reguläre stückweise Polynome des Einspurmodells

6.3.4 Interpretation der Sensitivitäten

Im Folgenden wird der Einfluss der Daten auf die Modellparameter k , L und r analysiert. Zur Bewertung wird der mittlere absolute Einfluss (5.17) herangezogen. Um (5.17) auswerten zu können, werden zunächst die Sensitivitäten der B-Spline Koeffizienten bezüglich der Datenpunkte bestimmt. Dies geschieht durch Lösen der linearen Gleichungssysteme (5.16). Zum Lösen wird die Funktion `scipy.linalg.solve_banded`³ verwendet. Sie nutzt die Routinen `gtsv`⁴ und `gbsv`⁵ der Fortran Bibliothek *LAPACK* [1] zum effizienten Lösen von linearen Gleichungssystemen mit Bandmatrizen.

Für die Analyse werden die Daten der Zustände und Steuerungen in 30 gleich große, zusammenhängende Bereiche unterteilt. Für jede Modellgröße $\chi \in \{x, y, v, \psi, \delta, a, \sigma\}$ sei

$$\mathcal{D}_i^X := \left\{ \bar{\chi}_k \mid t_0 + (i-1) \frac{t_f - t_0}{30} \leq t_k < t_0 + i \frac{t_f - t_0}{30} \right\} \text{ für } i = 1, \dots, 30. \quad (6.20)$$

Es werden die mittleren absoluten Einflüsse $\kappa^p(\mathcal{D}_i^X)$ für jeden Modellparameter $p \in \{k, L, r\}$ und eine Störung der Datenpunkte $\bar{\chi}_k \in \mathcal{D}_i^X$ um $\Delta \bar{\chi}_k = 0.1$ berechnet.

6.3.4.1 Analyse des Einflusses der Daten auf den Modellparameter k

Zunächst wird der Einfluss der Daten auf den Modellparameter k untersucht. Dazu werden die mittleren absoluten Einflüsse $\kappa^k(\mathcal{D}_i^X)$, wie in Abschnitt 6.1 erläutert, durch Rechtecke visualisiert. Die Visualisierung ist in Abbildung 6.9 dargestellt. Dabei ist die Skalierung der rechten vertikalen Koordinatenachsen in allen Plots gleich gewählt, um die Werte der mittleren absoluten Einflüsse direkt miteinander vergleichen zu können.

Die Abbildungen 6.9a bis 6.9e zeigen die mittleren absoluten Einflüsse der Daten der Zustände und Abbildungen 6.9f und 6.9g die mittleren absoluten Einflüsse der Daten der Steuerungen auf den Modellparameter k . Es fällt auf, dass lediglich die Daten der Geschwindigkeit und der Beschleunigung (s. Abb. 6.9c und 6.9f) einen nennenswerten Einfluss auf k haben. Für die Daten der x - und y -Positionen (s. Abb. 6.9a und 6.9b) sowie der Orientierung (s. Abb. 6.9d), des Lenkwinkels (s. Abb. 6.9e) und der Lenkwinkelgeschwindigkeit (s. Abb. 6.9g) ist der mittlere absolute Einfluss aufgrund der Skalierung der rechten vertikalen Koordinatenachsen nicht zu erkennen.

³https://docs.scipy.org/doc/scipy/reference/generated/scipy.linalg.solve_banded.html

⁴https://scc.ustc.edu.cn/zlsc/sugon/intel/mkl/mkl_manual/GUID-1811FAD9-A0DB-4845-9F2C-7CE5D07BDF15.htm

⁵https://scc.ustc.edu.cn/zlsc/tc4600/intel/2016.0.109/mkl/common/mklman_f/GUID-36E347A0-A1DF-48F0-A554-A98A3704A3B5.htm

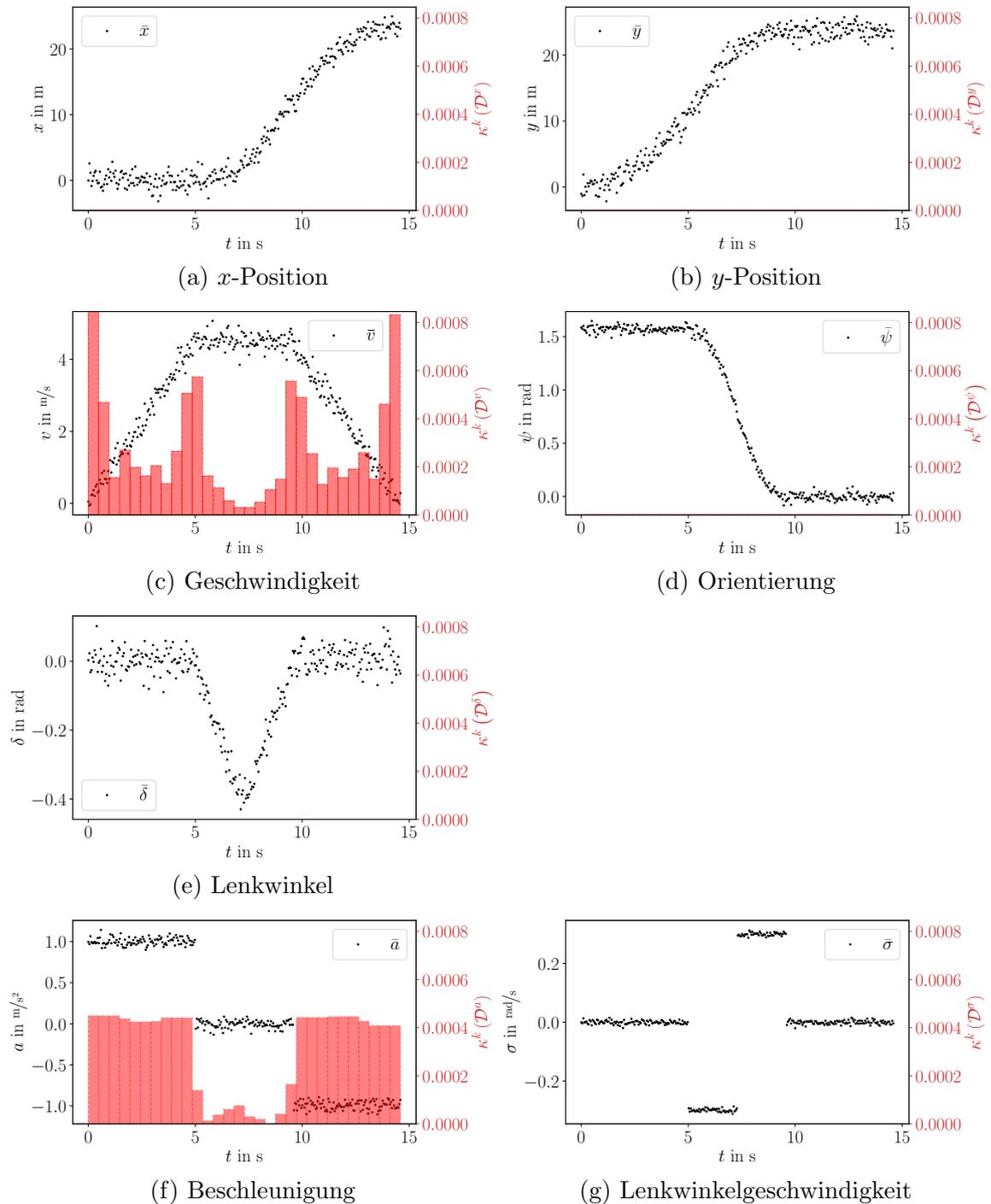


Abbildung 6.9: Mittlere absolute Einflüsse der Daten des Einspurmodells auf den Modellparameter k , die Daten sind gemäß (6.20) in 30 gleich große Bereiche aufgeteilt

Folglich kann ihr Einfluss im Vergleich zum Einfluss der Daten der Geschwindigkeit und der Beschleunigung vernachlässigt werden. Dies erscheint logisch, da der Modellparameter k im Einspurmodell (6.15) die Ableitung der Geschwindigkeit mit der Beschleunigung in Beziehung setzt (vgl. (6.15c)). Der Zusammenhang ist unabhängig von den anderen Zuständen und Steuerungen. Daher haben die Daten der übrigen Zustände und Steuerungen keinen Einfluss auf k .

Des Weiteren fällt in den Abbildungen 6.9c und 6.9f auf, dass die Daten der Geradeausfahrten ($t \leq 5$ s und $t \geq 9.6$ s) einen größeren mittleren absoluten Einfluss auf den Modellparameter k haben als die Daten der Kurvenfahrt (5 s $\leq t \leq 9.6$ s). Dies hängt damit zusammen, dass beim untersuchten Fahrmanöver während der Kurvenfahrt keine Beschleunigung stattfindet. Somit ist auch die Änderung der Geschwindigkeit konstant Null und die Gleichung (6.15c) unabhängig von k erfüllt. Die Daten der Kurvenfahrt können daher nicht genutzt werden, um den Modellparameter k zu identifizieren. Folglich ist auch der Einfluss der Daten der Kurvenfahrt kleiner als der Daten der Geradeausfahrt. Dass die mittleren absoluten Einflüsse dennoch positive Werte während der Kurvenfahrt aufweisen, hängt möglicherweise damit zusammen, dass die Approximationen \tilde{a}_{app} und \tilde{v}_{app} anhand verrauschter Messdaten bestimmt wurden. Aus diesem Grund ist weder \tilde{a}_{app} exakt Null, noch ist \tilde{v}_{app} wirklich konstant während der Kurvenfahrt.

Für die Daten der Geschwindigkeit (s. Abb. 6.9c) erkennt man einen besonders großen mittleren absoluten Einfluss bei $t \approx 0$ s und $t \approx 5$ s sowie bei $t \approx 10$ s und $t \approx 15$ s. Die Bereiche am Anfang und am Ende der Geradeausfahrten haben folglich einen großen Einfluss auf den Modellparameter k . Dieser Effekt wurde bereits bei der Analyse des einfachen dynamischen Modells beobachtet (vgl. Abb. 6.4c).

In den vorherigen Überlegungen wurde festgestellt, dass bei der Bewertung des Einflusses der Daten die Geradeaus- und die Kurvenfahrt eine zentrale Rolle einnehmen. Daher scheint es geeignet, den Einfluss der Daten in der x - y -Ebene gemeinsam mit der Trajektorie des Fahrzeugs, ähnlich wie in Abbildung 6.6, darzustellen. Da das Fahrzeug während der Geradeausfahrt beschleunigt, legt es in gleichen Zeiträumen unterschiedliche Distanzen zurück. Daher ist die zeitlich äquidistante Aufteilung der Daten (6.20) für eine Darstellung in der x - y -Ebene ungeeignet. Stattdessen sollte bei der Aufteilung der Daten die räumliche Komponente berücksichtigt werden. Da die in Kapitel 5 vorgestellte Methode zum Analysieren des Einflusses beliebiger Datenbereiche genutzt werden kann, ist dies leicht möglich.

Im Folgenden werden die Daten so aufgeteilt, dass das Fahrzeug in jedem Datenbereich die gleiche Distanz zurücklegt. Sei dazu

$$l(t_a, t_b) := \int_{t_a}^{t_b} \sqrt{\dot{\tilde{x}}_{\text{app}}^2(t) + \dot{\tilde{y}}_{\text{app}}^2(t)} dt$$

die Länge der Trajektorie zwischen den Zeitpunkten t_a und t_b . Die Zeitpunkte \tilde{t}_i

für $i = 1, \dots, n+1$ mit $\tilde{t}_1 = t_0$ und $\tilde{t}_{n+1} = t_f$ werden so gewählt, dass die Länge der Trajektorie zwischen je zwei Zeitpunkten gleich ist. Es soll gelten

$$l(\tilde{t}_i, \tilde{t}_{i+1}) = l(\tilde{t}_j, \tilde{t}_{j+1}) \quad \text{für } i, j = 1, \dots, n.$$

Für jede Modellgröße $\chi \in \{x, y, v, \psi, \delta, a, \sigma\}$ werden die Daten schließlich gemäß

$$\tilde{\mathcal{D}}_i^\chi := \{\tilde{\chi}_k \mid t_k \in [\tilde{t}_i, \tilde{t}_{i+1}]\} \quad \text{für } i = 1, \dots, n \quad (6.21)$$

in n Bereiche aufgeteilt. Dabei ist zu beachten, dass die Mengen $\tilde{\mathcal{D}}_i^\chi$ nicht leer sind, da der mittlere absolute Einfluss (5.17) sonst nicht definiert ist. Um den Einfluss der Daten zwischen \tilde{t}_i und \tilde{t}_{i+1} zu bewerten, werden die mittleren absoluten Einflüsse summiert. Es sei

$$\kappa_\Sigma^k(\tilde{t}_i, \tilde{t}_{i+1}) := \sum_{\chi \in \{x, y, v, \psi, \delta, a, \sigma\}} \kappa^k(\tilde{\mathcal{D}}_i^\chi) \quad \text{für } i = 1, \dots, n. \quad (6.22)$$

Mithilfe von $\kappa_\Sigma^k(\tilde{t}_i, \tilde{t}_{i+1})$ kann der Einfluss der Daten der Trajektorie zwischen den Zeitpunkten \tilde{t}_i und \tilde{t}_{i+1} bewertet werden.

In [Abbildung 6.10](#) ist der Einfluss der Daten in der x - y -Ebene visualisiert. Die Trajektorie $(\tilde{x}_{\text{app}}(t), \tilde{y}_{\text{app}}(t))$ ist als schwarze Linie dargestellt. Die Datenpunkte $(\tilde{x}_k, \tilde{y}_k)$ sind durch schwarze Punkte gekennzeichnet. Die Daten werden zum Analysieren des Einflusses gemäß (6.21) in $n = 30$ Bereiche aufgeteilt. In jedem Bereich legt das Fahrzeug die gleiche Distanz zurück. In vertikaler Koordinatenrichtung ist der summierte mittlere absolute Einfluss (6.22) aufgetragen. Ähnlich wie in den vorigen Abbildungen wird der Einfluss mithilfe von roten Rechtecken visualisiert. Die Höhe der Rechtecke entspricht dem Wert von $\kappa_\Sigma^k(\tilde{t}_i, \tilde{t}_{i+1})$. Im Gegensatz zu den zweidimensionalen Visualisierungen sind die Rechtecke jedoch auf dem Abschnitt der Trajektorie $(\tilde{x}_{\text{app}}(t), \tilde{y}_{\text{app}}(t))$ für $t \in [\tilde{t}_i, \tilde{t}_{i+1}]$ positioniert. In [Abbildung 6.10](#) ist nochmals deutlich zu erkennen, dass der Modellparameter k im Wesentlichen während den beiden Geradeausfahrten beeinflusst wird. Darüber hinaus ist zu erkennen, dass der Einfluss zu Beginn und zum Ende der Geradeausfahrt größer ist als in der Mitte.

6.3.4.2 Analyse des Einflusses der Daten auf den Modellparameter L

Als nächstes wird der Einfluss der Daten auf den Modellparameter L analysiert. Dazu werden die Daten gemäß (6.20) in 30 gleich große Bereiche \mathcal{D}_i^χ für $\chi \in \{x, y, v, \psi, \delta, a, \sigma\}$ aufgeteilt und die mittleren absoluten Einflüsse $\kappa^L(\mathcal{D}_i^\chi)$ bestimmt. In [Abbildung 6.11](#) sind die mittleren absoluten Einflüsse visualisiert. Die Skalen der rechten vertikalen Koordinatenachsen sind in allen Abbildungen gleich, um die Werte der mittleren absoluten Einflüsse direkt miteinander vergleichen zu können.

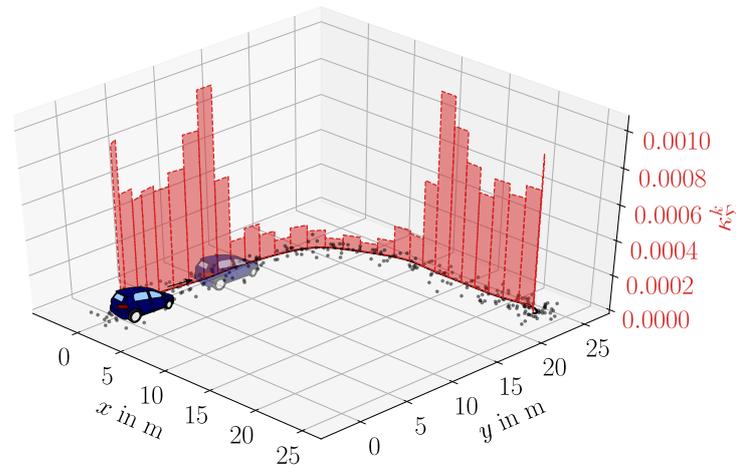


Abbildung 6.10: Summierter mittlerer absoluter Einfluss der Daten des Einspurmodells auf den Modellparameter k , visualisiert auf der Trajektorie des Fahrzeugs. Die Daten sind gemäß (6.21) in 30 Bereiche eingeteilt, während denen das Fahrzeug die gleiche Distanz zurücklegt

Es fällt auf, dass lediglich die Daten der Geschwindigkeit, der Orientierung und des Lenkwinkels (s. Abb. 6.11c bis 6.11e) einen nennenswerten Einfluss auf den Modellparameter L haben. Für die Daten der x - und y -Positionen (s. Abb. 6.11a und 6.11b) sowie der Beschleunigung (s. Abb. 6.9f) und der Lenkwinkelgeschwindigkeit (s. Abb. 6.11g) ist der mittlere absolute Einfluss aufgrund der Skalierung der rechten vertikalen Koordinatenachsen nicht zu erkennen. Folglich kann ihr Einfluss im Vergleich zum Einfluss der Daten der Geschwindigkeit, der Orientierung und des Lenkwinkels vernachlässigt werden. Dies kann damit erklärt werden, dass der Modellparameter L im Einspurmodell (6.15) in der Gleichung der Änderung der Orientierung (6.15d) auftritt. Diese Gleichung hängt von der Geschwindigkeit und dem Lenkwinkel ab. Die Geschwindigkeit und der Lenkwinkel werden zwar durch die Beschleunigung und die Lenkwinkelgeschwindigkeit gesteuert, laut der Analyse ist der direkte Einfluss der Daten der Steuerungen auf den Modellparameter jedoch gering.

In Abbildungen 6.11c bis 6.11e kann abgelesen werden, dass die Daten des Lenkwinkels (s. Abb. 6.11e) den größten mittleren absoluten Einfluss auf L aufweisen. Der mittlere absolute Einfluss der Daten der Geschwindigkeit ist dagegen gering (s. Abb. 6.11e). Darüber hinaus fällt auf, dass die Daten der Kurvenfahrt ($5\text{ s} \leq t \leq 9.6\text{ s}$) einen größeren Einfluss auf den Modellparameter L haben als die Daten der Geradeausfahrten ($t \leq 5\text{ s}$ und $t \geq 9.6\text{ s}$). Dies ist ein sinnvolles Ergebnis, da während der Geradeausfahrt der Lenkwinkel Null ist. Folglich ändert sich die Orientierung des Fahrzeugs nicht und (6.15d) ist identisch erfüllt. Die Daten der Geradeausfahrten enthalten daher keine relevante Informationen zum Identifi-

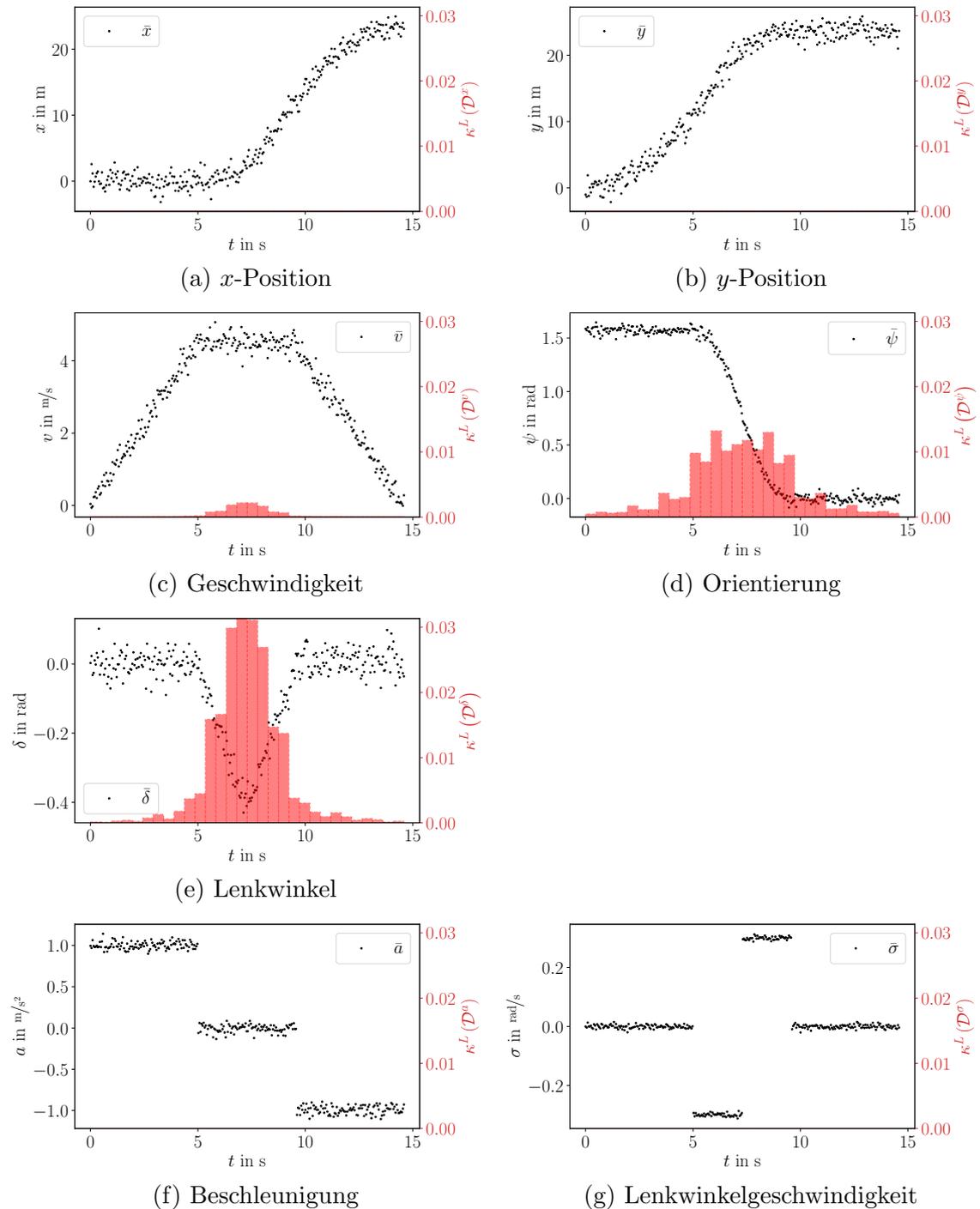


Abbildung 6.11: Mittlere absolute Einflüsse der Daten des Einspurmodells auf den Modellparameter L , die Daten sind gemäß (6.20) in 30 gleich große Bereiche aufgeteilt

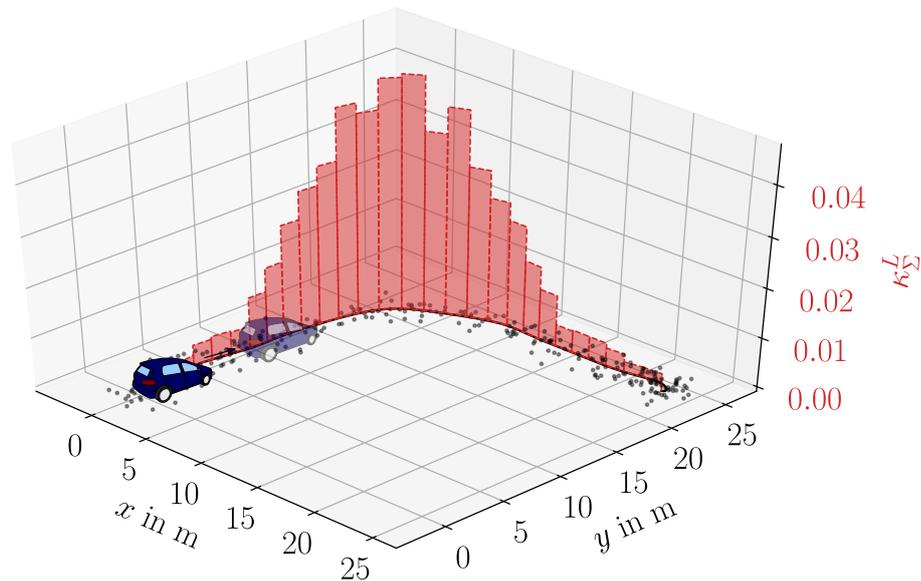


Abbildung 6.12: Summierter mittlerer absoluter Einfluss der Daten des Einspurmodells auf den Modellparameter L , visualisiert auf der Trajektorie des Fahrzeugs. Die Daten sind gemäß (6.21) in 30 Bereiche eingeteilt, während denen das Fahrzeug die gleiche Distanz zurücklegt

zieren des Modellparameters L . Insbesondere bei $t \approx 7.5$ s ist für die Daten der drei Zustände ein besonders großer mittlerer absoluter Einfluss zu erkennen. Die Daten in der Mitte der Kurve sind daher besonders wichtig, um den Modellparameter L zu identifizieren.

Wie im vorigen Abschnitt wird der Einfluss der Daten auf den Modellparameter L zusätzlich in der x - y -Ebene auf der Trajektorie des Fahrzeugs visualisiert. Die Daten werden dazu gemäß (6.21) in 30 Bereiche aufgeteilt, während denen das Fahrzeug die gleiche Distanz zurücklegt. Zum Bewerten des Einflusses eines Datenbereiches werden die mittleren absoluten Einflüsse gemäß (6.22) summiert.

In [Abbildung 6.12](#) ist die Summe der mittleren absoluten Einflüsse κ_{Σ}^L auf den jeweiligen Abschnitten der Trajektorie des Fahrzeugs visualisiert. Die schwarze Linie stellt die Trajektorie des Fahrzeugs dar. Die schwarzen Punkte sind die Datenpunkte (\bar{x}_k, \bar{y}_k) . In vertikaler Koordinatenrichtung ist der summierte mittlere absolute Einfluss κ_{Σ}^L aufgetragen. Der Einfluss auf den Abschnitten der Trajektorie ist durch rote Rechtecke visualisiert. Die Höhe eines Rechtecks ist der Wert κ_{Σ}^L und die Breite entspricht dem zugehörigen Abschnitt der Trajektorie. In [Abbildung 6.12](#) ist deutlich zu erkennen, dass der Einfluss der Daten während der Kurvenfahrt größer ist als während den Geradeausfahrten. Insbesondere in der Mitte der Kurve ist der Einfluss besonders groß.

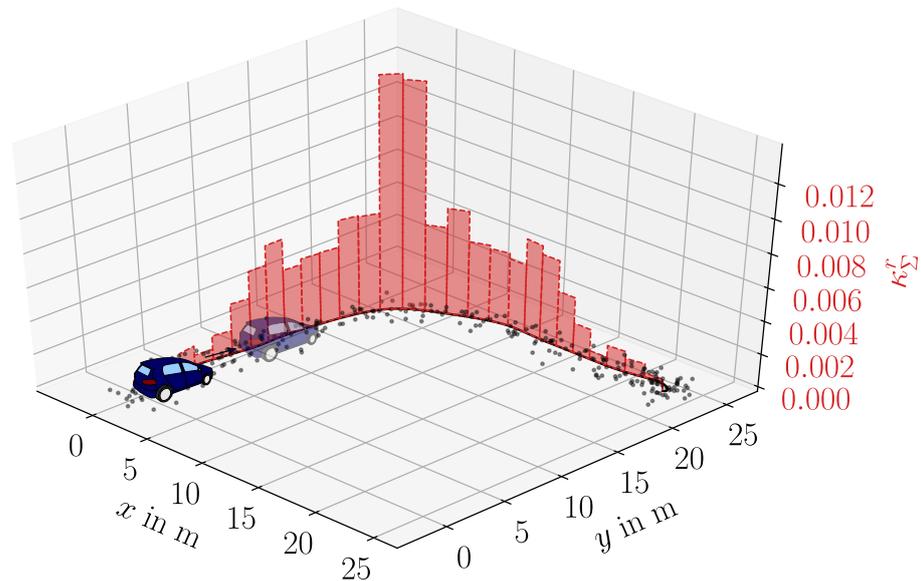


Abbildung 6.13: Summierter mittlerer absoluter Einfluss der Daten des Einspurmodells auf den Modellparameter r , visualisiert auf der Trajektorie des Fahrzeugs. Die Daten sind gemäß (6.21) in 30 Bereiche eingeteilt, während denen das Fahrzeug die gleiche Distanz zurücklegt

6.3.4.3 Analyse des Einflusses der Daten auf den Modellparameter r

Der Vollständigkeit halber wird abschließend der Einfluss der Daten auf den Modellparameter r analysiert. Der Modellparameter r verknüpft die Ableitung des Lenkwinkels $\dot{\delta}$ mit der gesteuerten Lenkwinkelgeschwindigkeit σ (s. (6.15e)). Er verhält sich deshalb ähnlich wie der Parameter k , welcher die Ableitung der Geschwindigkeit \dot{v} mit der gesteuerten Beschleunigung a verknüpft. Die Analyse und Interpretation der Ergebnisse wäre daher ähnlich wie in Abschnitt 6.3.4.1. Auf eine ausführliche Analyse wie in den vorigen beiden Abschnitten wird deshalb an dieser Stelle verzichtet und es werden lediglich die wichtigsten Ergebnisse zusammengefasst.

In Abbildung 6.13 ist die Summe der mittleren absoluten Einflüsse κ_{Σ}^r gemäß (6.22) auf den jeweiligen Abschnitten der Trajektorie des Fahrzeugs visualisiert. Die Daten wurden dafür gemäß (6.21) in 30 Bereiche aufgeteilt, in denen das Fahrzeug die gleiche Distanz zurücklegt. Die schwarze Linie stellt die Trajektorie des Fahrzeugs dar. Die schwarzen Punkte sind die Datenpunkte (\bar{x}_k, \bar{y}_k) . In vertikaler Koordinatenrichtung ist der summierte mittlere absolute Einfluss κ_{Σ}^r aufgetragen. Der Einfluss auf den Abschnitten der Trajektorie ist durch rote Rechtecke visualisiert. Die Höhe eines Rechtecks ist der Wert κ_{Σ}^L und die Breite entspricht dem zugehörigen Abschnitt der Trajektorie.

In Abbildung 6.13 ist deutlich zu erkennen, dass der Einfluss der Daten während der

Kurvenfahrt größer ist als während den Geradeausfahrten. Insbesondere die Daten in der Mitte der Kurvenfahrt haben einen großen Einfluss auf den Parameter r . Dies entspricht den Beobachtungen aus den vorigen beiden Abschnitten: Während der Geradeausfahrten ist (6.15e) für jeden Wert von r erfüllt und folglich können die Daten der Geradeausfahrten nicht genutzt werden, um den Modellparameter r zu identifizieren.

Eine nähere Analyse zeigt, dass lediglich die Daten des Lenkwinkels und der Lenkwinkelgeschwindigkeit den Modellparameter r beeinflussen (s. ergänzende Abbildungen in Anhang C.2). Das kann damit erklärt werden, dass Gleichung (6.15e) lediglich vom Lenkwinkel und der Lenkwinkelgeschwindigkeit abhängt und diese Größen unabhängig von den anderen Zuständen und Steuerungen sind.

Zusammenfassend lässt sich sagen, dass die in Kapitel 5 entwickelte Methode geeignet ist, um den Einfluss von Daten auf die Modellparameter des Einspurmodells zu analysieren. Mithilfe des mittleren absoluten Dateneinflusses (5.17) kann der Einfluss der Daten nachvollziehbar bewertet werden. Zudem ergibt die Analyse, dass die Daten der x - und y -Positionen keinen der drei Modellparameter k , L und r beeinflussen. Da die Gleichungen (6.15c), (6.15d) und (6.15e) unabhängig von den Zuständen x und y sind, können die Modellparameter wohl auch anhand des reduzierten dynamischen Modells

$$\begin{aligned}\dot{v}(t) &= ka(t) \\ \dot{\psi}(t) &= \frac{v(t)}{L} \tan \delta(t) \quad \text{für } t \in [t_0, t_f] \\ \dot{\delta}(t) &= r\sigma(t)\end{aligned}$$

identifiziert werden. Die Größe des Parameteridentifikationsproblems könnte dadurch reduziert werden.

Kapitel 7

Fazit

Zum Abschluss der Arbeit werden die wichtigsten Resultate zusammengefasst. Darüber hinaus wird in [Abschnitt 7.2](#) ein Ausblick über mögliche weiterführende Fragestellungen und Erweiterungen der vorgestellten Methode zum Analysieren des Einflusses von Daten auf die Modellparameter dynamischer Modelle gegeben.

7.1 Zusammenfassung

Das Ziel der vorliegenden Arbeit war die Entwicklung einer Methode zum Analysieren des Einflusses von Daten auf die Parameter dynamischer Modelle. Die Methode beruht auf einer geeigneten Approximation der Daten durch stückweise Polynome sowie der Berechnung und Interpretation von parametrischen Sensitivitäten. In [Kapitel 2](#) bis [4](#) der Arbeit wurden die Grundlagen der Methode erarbeitet. [Kapitel 5](#) beschäftigte sich schließlich mit der Herleitung der Methode. In [Kapitel 6](#) wurden numerische Auswertungen zu der Methode vorgestellt. Im Folgenden werden die wichtigsten Resultate der Arbeit für jedes Kapitel zusammengefasst.

In [Kapitel 2](#) wurden Grundlagen der nichtlinearen Optimierung dargestellt. In diesem Zuge wurden auch Resultate der parametrischen Sensitivitätsanalyse zusammengefasst. Darüber hinaus wurde die Softwarebibliothek WORHP vorgestellt, welche ein SQP-Verfahren zum numerischen Lösen von nichtlinearen Optimierungsproblemen implementiert. WORHP zeichnet sich dadurch aus, auch hochdimensionale Optimierungsprobleme mit mehreren Millionen Optimierungsvariablen und Nebenbedingungen lösen zu können. In [Abschnitt 2.4.2](#) wurden verschiedene von WORHP implementierte Techniken dargestellt, mit denen dies gelingt. Darüber hinaus ermöglicht das in WORHP integrierte Modul WORHP Zen die effiziente Berechnung von parametrischen Sensitivitäten.

In [Kapitel 3](#) wurden stückweise Polynome thematisiert und klassifiziert. Insbesondere wurden ν -reguläre stückweise Polynome eingeführt. Diese zeichnen sich durch ge-

wisse Regularitätseigenschaften an den Übergängen zwischen zwei Polynomstücken aus und eignen sich gut zur Approximation von Daten. Sie sind ein wesentlicher Bestandteil der entwickelten Methode zum Analysieren des Einflusses der Messdaten auf die Modellparameter dynamischer Modelle. Des Weiteren wurden B-Splines als Basisfunktionen des Vektorraums der ν -regulären stückweisen Polynome vorgestellt und deren Eigenschaften diskutiert. B-Splines bilden eine gut konditionierte Basis der ν -regulären stückweisen Polynome und sind daher besonders für numerisch Berechnungen geeignet. In [Abschnitt 3.3.3](#) wurden zwei von de Boor [7, 8] entwickelten Algorithmen vorgestellt, mit denen ν -reguläre stückweise Polynome effizient und numerisch stabil mithilfe von B-Splines ausgewertet werden können.

In [Kapitel 4](#) der Arbeit wurden Grundlagen der Parameteridentifikation vorgestellt. Es wurden zunächst explizite Parameteridentifikationsprobleme behandelt. Explizite Parameteridentifikationsprobleme können häufig auf Least-Squares-Probleme zurückgeführt werden. Dabei handelt es sich um eine spezielle Klasse nichtlinearer Optimierungsprobleme. In [Abschnitt 4.1.1](#) wurde ein Überblick über numerische Lösungsverfahren von Least-Squares-Problemen gegeben. Der Fokus von [Kapitel 4](#) lag auf der Parameteridentifikation in dynamischen Modellen. Als dynamisches Modell wurde dabei ein System gewöhnlicher Differentialgleichungen $\dot{x} = f(t, x, u, p)$ bezeichnet, welches gegebenenfalls von Steuerungen u und Parametern p abhängt. Verschiedene Lösungsverfahren dynamischer Parameteridentifikationsprobleme wurden in [Abschnitt 4.2](#) diskutiert. Insbesondere wurde die Kollokationsmethode vorgestellt. Bei der Kollokationsmethode werden zunächst die Daten durch ν -reguläre stückweise Polynome approximiert. Im zweiten Schritt der Methode werden die Modellparameter identifiziert. Dazu wird die quadrierte Abweichung zwischen der Ableitung der Approximation der Daten und dem Vektorfeld f minimiert. Das resultierende Problem ist ein Least-Squares-Problem und kann effizient gelöst werden. Damit die Modellparameter gut mithilfe der Kollokationsmethode identifiziert werden können, ist es essentiell, dass die Approximationen einerseits die Daten gut repräsentieren und andererseits näherungsweise das dynamische Modell erfüllen. Eine große Herausforderung bei der Kollokationsmethode ist daher, eine gute Approximation der Daten zu bestimmen. In [Abschnitt 4.2.2.4](#) wurden verschiedene Erweiterungen der Kollokationsmethode diskutiert, welche sich mit dieser Thematik beschäftigen.

Aufbauend auf [Kapitel 2](#) bis [4](#) wurde in [Kapitel 5](#) eine Methode zum Analysieren des Einflusses der Daten auf die Modellparameter dynamischer Modelle entwickelt. Im ersten Schritt der Methode wurden die Daten durch ν -reguläre stückweise Polynome approximiert. Als Basisfunktionen wurden dabei B-Splines verwendet. Auf diese Weise wurden die Daten geglättet und mögliches Rauschen reduziert. Als nächstes wurden die Modellparameter des dynamischen Modells identifiziert. Dazu wurden statt der Daten die Approximationen der Daten verwendet. Das Parameteridentifikationsproblem konnte daher auf natürliche Weise mit der Kollokationsmethode gelöst werden. In dem resultierenden Optimierungsproblem wurden die Koeffizien-

ten der B-Splines in der Basisdarstellung der Approximationen als Störparameter interpretiert. Mithilfe der parametrischen Sensitivitätsanalyse wurden schließlich die Sensitivitäten der identifizierten Modellparameter bezüglich dieser Koeffizienten bestimmt. Um den Einfluss bestimmter Bereiche in den Daten auf die Modellparameter zu beurteilen, wurde berechnet, wie sich eine Störung der Daten auf die Koeffizienten der B-Splines auswirkt. So war es möglich, den Einfluss beliebiger Datenbereiche auf die Modellparameter zu analysieren. Dabei wurde insbesondere das mitunter aufwändige Parameteridentifikationsproblem nur einmal gelöst werden.

In [Kapitel 6](#) wurden numerische Auswertungen zu der in [Kapitel 5](#) vorgestellte Methode präsentiert. Zunächst wurde die Methode anhand eines einfachen dynamischen Modells mit einem Modellparameter demonstriert. Die Resultate der entwickelten Methode zur Analyse des Einfluss der Daten auf den Modellparameter waren nachvollziehbar und wurden zudem durch analytische Überlegungen gestützt. Anschließend wurde die Methode auf ein Einspurmodell angewendet und ein beispielhaftes Fahrmanöver analysiert. Dieses setzte sich aus zwei Geradeausfahrten und einer Kurvenfahrt zusammen. Anhand analytischer Überlegungen konnte begründet werden, dass die Modellparameter entweder während der Geradeausfahrten oder lediglich in der Kurve einen Einfluss auf das Einspurmodell haben. Zu dem Fahrmanöver wurden durch eine numerische Simulation Daten erzeugt. Mithilfe der entwickelten Methode wurde der Einfluss dieser Daten auf die Modellparameter des Einspurmodells analysiert. Es wurde gezeigt, dass die entwickelte Methode sinnvolle Ergebnisse erzielt. Für die Modellparameter, die lediglich während der Kurvenfahrt einen Einfluss auf das Modell haben, wurden die Daten der Kurvenfahrt als besonders wichtig erkannt. Für die Daten der Geradeausfahrten wurde dagegen ein geringer Einfluss ermittelt. Für die Modellparameter, die lediglich während der Geradeausfahrten einen Einfluss auf das Modell haben, wurden die Daten der Geradeausfahrten als besonders wichtig erkannt.

7.2 Ausblick

In dieser Arbeit wurde anhand der Analyse eines einfachen Einspurmodells gezeigt, dass die in [Kapitel 5](#) entwickelte Methode geeignet ist, um den Einfluss von Daten auf die Modellparameter dynamischer Modelle zu analysieren. In weiteren Untersuchungen sollte die Tauglichkeit der Methode anhand konkreter Anwendungen untersucht werden. Insbesondere sollten für die Analyse Messdaten von realen Prozessen herangezogen werden.

Die entwickelte Methode nutzt die Kollokationsmethode zum Lösen des dynamischen Parameteridentifikationsproblems. In der klassischen Variante der Kollokationsmethode müssen für jeden Zustand Daten vorliegen. Dies ist in praktischen Anwendung häufig nicht der Fall. Daher wäre es sinnvoll, weitere Lösungsverfahren dynamischer

Parameteridentifikationsprobleme in die entwickelte Methode zu integrieren. Als Erweiterung der klassischen Kollokationsmethode würde sich die iterative Kollokationsmethode nach Poyton et al. [52] anbieten. Die Sensitivitäten könnten in diesem Fall anhand der Optimierungsprobleme der letzten Iteration bestimmt werden. Darüber hinaus wäre es denkbar, das Parameteridentifikationsproblem mithilfe eines anderen Lösungsansatzes, wie dem Ansatz der vollen Diskretisierung, auf ein nichtlineares Optimierungsproblem zurückzuführen und zu lösen.

Für eine erfolgreiche Identifikation der Modellparameter mithilfe der Kollokationsmethode ist essentiell, dass die Approximationen der Daten einerseits die Daten gut repräsentieren und andererseits näherungsweise das dynamische Modell erfüllen. Zudem bildet die Approximation der Daten durch ν -reguläre stückweise Polynome einen zentralen Bestandteil der entwickelten Methoden zum Analysieren des Einflusses von Daten auf die Modellparameter dynamischer Modelle. Deshalb ist es wünschenswert, die Daten möglichst gut zu approximieren. Die Güte der Approximation kann oftmals durch geschicktes Platzieren der Knotenpunkte verbessert werden. Dies kann im konkreten Anwendungsfall jedoch aufwändig sein und ein Hindernis für die Benutzung der entwickelten Methode darstellen. Deshalb ist es erstrebenswert, automatisiert die optimale Anordnung der Knotenpunkte zu bestimmen. Wie in [Abschnitt 4.2.2.4](#) beschrieben, gibt es bereits Untersuchungen zu diesem Thema. Aufbauend auf den Ergebnissen könnte eine automatisierte Bestimmung der Knotenpunkte entwickelt werden.

In den numerischen Auswertungen wurde der Einfluss von zeitlich zusammenhängenden Bereichen der Daten analysiert. Dies ist zunächst eine intuitive Aufteilung und erlaubt eine übersichtliche grafische Darstellung der Ergebnisse. Die entwickelte Methode ist jedoch nicht auf zeitlich zusammenhängende Daten angewiesen. So könnten die Daten auch in einer anderen Form geclustert werden. Durch die Analyse der Einflüsse dieser Cluster ergibt sich möglicherweise ein noch besseres Verständnis des zugrunde liegenden dynamischen Modells. Dabei wäre denkbar, verschiedene Methoden des Clusterings zu untersuchen.

Darüber hinaus könnte untersucht werden, ob die Ergebnisse der entwickelten Methode zur optimalen Versuchsplanung genutzt werden können. Das Ziel der optimalen Versuchsplanung ist die Entwicklung von Versuchen zur Aufnahme von Daten, anhand derer die Modellparameter möglichst gut identifiziert werden können. Mit der entwickelten Methode kann analysiert werden, ob durch einen Versuch Daten produziert werden, welche keinen der Modellparameter beeinflussen. Dies deutet darauf hin, dass diese Daten nicht wichtig für die Parameteridentifikation sind. Bei der Planung von weiteren Versuchen kann daher vermieden werden, solche Daten zu erzeugen.

Die in [Kapitel 5](#) vorgestellte Methode wurde anhand dynamischer parameterabhängiger Modelle entwickelt. Eine mögliche Erweiterung wäre die Übertragung der

Methode auf explizite parameterabhängige Modelle. Eine interessante Anwendung wäre beispielsweise den Einfluss von Daten bei der Identifikation von Kennfeldern mithilfe der Methode zu analysieren.

Anhang A

Details zur Herleitung der Algorithmen von de Boor

Im Folgenden werden die beiden Algorithmen von de Boor [Alg. 2](#) und [Alg. 3](#) detailliert hergeleitet. Herleitungen der Algorithmen sind auch in den den Arbeiten von de Boor [\[7, 8\]](#) zu finden. Wie zu Beginn von [Kapitel 3](#) diskutiert, gibt es jedoch eine Diskrepanz zwischen der Bedeutung des Grads von stückweisen Polynomen in den Arbeiten von de Boor und dieser Arbeit. Darüber hinaus hat sich der Autor im Gegensatz zu de Boor dazu entschieden, die Indizierung der Knotenfolge τ bei Null zu beginnen. Dies hat praktische Gründe und erlaubt eine direkte Implementierung der Algorithmen in Programmiersprachen wie C++, bei denen die Indizierung von Datenstrukturen typischerweise bei Null beginnt. Aus obigen Gründen gibt es einige Unterschiede in der Darstellung der Algorithmen zwischen den Arbeiten von de Boor [\[7, 8\]](#) und dieser Arbeit. Um die Herleitungen von [Alg. 2](#) und [Alg. 3](#) bei Bedarf nachvollziehen zu können, werden sie in den folgenden beiden Abschnitten in der Notation dieser Arbeit detailliert dargestellt.

A.1 Details zu de Boors erstem Algorithmus

De Boors erster Algorithmus ([Alg. 2](#)) beruht auf dem rekursiven Zusammenhang [\(3.29\)](#). In diesem Abschnitt wird ein effizientes Vorgehen zum Auswerten von [\(3.29\)](#) hergeleitet, welches schließlich in [de Boors erstem Algorithmus](#) resultiert.

Es wird angenommen, dass die Werte, $r_i^{[k-1]}(x)$ für ein $k \in \{1, \dots, d\}$ und $i = j - d + k - 1, \dots, j$ bekannt seien. Gesucht sind die Werte $r_i^{[k]}(x)$ für

$i = j - d + k, \dots, j$. Der besseren Übersicht halber sei

$$a := \begin{pmatrix} r_{j-d+k-1}^{[k-1]}(x) \\ r_{j-d+k}^{[k-1]}(x) \\ \vdots \\ r_j^{[k-1]}(x) \end{pmatrix} \in \mathbb{R}^{d-k+2} \quad \text{und} \quad a' := \begin{pmatrix} r_{j-d+k}^{[k]}(x) \\ r_{j-d+k+1}^{[k]}(x) \\ \vdots \\ r_j^{[k]}(x) \end{pmatrix} \in \mathbb{R}^{d-k+1}.$$

Es gilt der Zusammenhang

$$a_s = r_{j-d+k-1+s}^{[k-1]}(x) \quad \text{für} \quad s = 0, \dots, d - k + 1 \quad (\text{A.1})$$

und

$$a'_s = r_{j-d+k+s}^{[k]}(x) \quad \text{für} \quad s = 0, \dots, d - k. \quad (\text{A.2})$$

Folglich ist

$$\begin{aligned} a'_s &\stackrel{(\text{A.2})}{=} r_{j-d+k+s}^{[k]}(x) \\ &\stackrel{(3.29)}{=} \frac{(x - \tau_{j-d+k+s}) r_{j-d+k+s}^{[k-1]}(x) + (\tau_{j+s+1} - x) r_{j-d+k+s-1}^{[k-1]}(x)}{\tau_{j+s+1} - \tau_{j-d+k+s}} \\ &\stackrel{(\text{A.1})}{=} \frac{(x - \tau_{j-d+k+s}) a_{s+1} + (\tau_{j+s+1} - x) a_s}{\tau_{j+s+1} - \tau_{j-d+k+s}} \end{aligned} \quad (\text{A.3})$$

für $s = 0, \dots, d - k$. Durch Definition von

$$\gamma_s := \frac{\tau_{j+s+1} - x}{\tau_{j+s+1} - \tau_{j-d+k+s}}$$

vereinfacht sich (A.3) zu

$$a'_s = (1 - \gamma_s) a_{s+1} + \gamma_s a_s.$$

Diese Rekursionsformel bildet den Kern von [de Boors erstem Algorithmus](#). Ausgehend von $r_i^{[0]}(x) = c_i$ für $i = j - d, \dots, j$ wird der Wert $f(x) = r_j^{[d]}(x)$ berechnet. Um die Effizienz des Algorithmus zu erhöhen, wird lediglich ein Vektor zum Speichern der Elemente verwendet, dessen Einträge überschrieben werden.

A.2 Details zu de Boors zweitem Algorithmus

[De Boors zweiter Algorithmus](#) (Alg. 3) beruht auf der Tatsache, dass auf dem Intervall $[\tau_j, \tau_{j+1})$ alle B-Splines außer $B_{i,d,\tau}$ mit $i = j - d, \dots, j$ konstant Null sind. Dies kann ausgenutzt werden, um in der Rekursionsformel (3.15) lediglich die notwendigen Terme zu berechnen: Terme, die ohnehin Null sind, müssen nicht berechnet

werden.

Ausgabe von de Boors zweitem Algorithmus sind die Werte aller B-Splines vom Grad d , welche auf dem Intervall $[\tau_j, \tau_{j+1})$ ungleich Null sind. In diesem Abschnitt wird ein effizientes Vorgehen zum Berechnen dieser Werte hergeleitet, welches schließlich in de Boors zweitem Algorithmus resultiert.

Es wird angenommen, dass die Werte $B_{j-(s-1),s-1,\tau}(x), \dots, B_{j,s-1,\tau}(x)$ für ein $s \in \{1, \dots, d\}$ bekannt sind. Gesucht sind die Werte $B_{j-s,s,\tau}(x), \dots, B_{j,s,\tau}(x)$. Der besseren Übersicht halber sei

$$b := \begin{pmatrix} B_{j-(s-1),s-1,\tau}(x) \\ B_{j-(s-1)+1,s-1,\tau}(x) \\ \vdots \\ B_{j,s-1,\tau}(x) \end{pmatrix} \in \mathbb{R}^s \quad \text{und} \quad b' := \begin{pmatrix} B_{j-s,s,\tau}(x) \\ B_{j-s+1,s,\tau}(x) \\ \vdots \\ B_{j,s,\tau}(x) \end{pmatrix} \in \mathbb{R}^{s+1}.$$

Es gilt der Zusammenhang

$$b_r = B_{j-(s-1)+r,s-1,\tau}(x) \quad \text{für} \quad r = 0, \dots, s-1 \quad (\text{A.4})$$

und

$$b'_r = B_{j-s+r,s,\tau}(x) \quad \text{für} \quad r = 0, \dots, s. \quad (\text{A.5})$$

Folglich ist

$$\begin{aligned} b'_r &\stackrel{(\text{A.5})}{=} B_{j-s+r,s,\tau}(x) \\ &\stackrel{(3.15)}{=} \frac{x - \tau_{j-s+r}}{\tau_{j+r} - \tau_{j-s+r}} B_{j-s+r,s-1,\tau}(x) + \frac{\tau_{j+r+1} - x}{\tau_{j+r+1} - \tau_{j-s+r+1}} B_{j-s+r+1,s-1,\tau}(x) \\ &\stackrel{(\text{A.4})}{=} (x - \tau_{j-s+r}) \frac{b_{r-1}}{\tau_{j+r} - \tau_{j-s+r}} + (\tau_{j+r+1} - x) \frac{b_r}{\tau_{j+r+1} - \tau_{j-s+r+1}} \end{aligned} \quad (\text{A.6})$$

für $r = 1, \dots, s-1$. Damit (A.6) auch für $r = 0$ und $r = s$ gilt, sei $b_{-1} = b_s := 0$. Dies ist dadurch gerechtfertigt, dass die B-Splines $B_{j-s,s-1,\tau}$ und $B_{j+1,s-1,\tau}$ konstant Null auf $[\tau_j, \tau_{j+1})$ sind (vgl. (3.12) und Abb. 3.9). Es ist hervorzuheben, dass die Nenner in Gleichung (A.6) für $r = 1, \dots, s-1$ ungleich Null sind: Da τ monoton steigend ist, gilt

$$r \leq s-1 \Rightarrow \tau_{j-s+r} \leq \tau_{j-1}$$

und

$$r \geq 1 \Rightarrow \tau_{j+r} \geq \tau_{j+1}.$$

Per Voraussetzung ist $\tau_j < \tau_{j+1}$ und folglich $\tau_{j-s+r} \leq \tau_{j-1} < \tau_{j+1} \leq \tau_{j+r}$, also $\tau_{j+r} - \tau_{j-s+r} \neq 0$. Analog ist $\tau_{j+r+1} - \tau_{j-s+r+1} \neq 0$. Beim Auswerten von (A.6) kann also auf eine Fallunterscheidung, ob einer der Nenner Null ist, verzichtet werden.

Durch Definition von

$$\tau_r^R := \tau_{j+r+1}, \quad \tau_r^L := \tau_{j-s+r+1} \quad \text{und} \quad w_r := \frac{b_r}{\tau_r^R - \tau_r^L}$$

vereinfacht sich (A.6) zu

$$\begin{aligned} b'_r &= (x - \tau_{r-1}^L) \frac{b_{r-1}}{\tau_{r-1}^R - \tau_{r-1}^L} + (\tau_r^R - x) \frac{b'_r}{\tau_r^R - \tau_r^L} \\ &= (x - \tau_{r-1}^L) w_{r-1} + (\tau_r^R - x) w_r. \end{aligned}$$

Diese Rekursionsformel bildet den Kern von [de Boors zweitem Algorithmus](#). Bei der Berechnung kann ausgenutzt werden, dass der Term w_{r-1} sowohl in der Gleichung für b'_r als auch für b'_{r-1} auftritt.

Durch Umkehren der Iteration über r kann der Algorithmus noch effizienter gestaltet werden. Es wird dann lediglich ein Vektor zum Speichern der Werte der B-Splines der letzten Iteration benötigt, dessen Einträge überschrieben werden.

Anhang B

Rechnungen zur Analyse des Einflusses von Daten auf den Parameter eines einfachen dynamischen Modells

Im Folgenden sind die analytischen Berechnungen zur Analyse des Einflusses von Daten auf den Modellparameter eines einfachen dynamischen Modells dargestellt (vgl. [Abschnitt 6.2](#)).

B.1 Identifikation des Modellparameters und Berechnung der Sensitivitäten

In diesem Abschnitt wird das Optimierungsproblem (5.6) zum Identifizieren des Modellparameters p des einfachen dynamischen Modells

$$\dot{x}(t) = p \cdot u(t), \quad t \in [0, 1]$$

gelöst. Anschließend werden die Sensitivitäten dp/dc_0^x , dp/dc_1^x und dp/dc_0^u des identifizierten Modellparameters bezüglich der Koeffizienten der Approximationen

$$\tilde{x}_{\text{app}}(t) = c_0^x(1-t) + c_1^x t \quad \text{und} \quad \tilde{u}_{\text{app}}(t) = c_0^u.$$

berechnet. Es gilt $\dot{\tilde{x}}_{\text{app}}(t) = c_1^x - c_0^x$ und folglich vereinfacht sich (5.6) zu

$$\min_{p \in \mathbb{R}} F(p, c_1^x, c_0^x, c_0^u) := \frac{1}{n_t} \sum_{k=1}^{n_t} (c_1^x - c_0^x - pc_0^u)^2 = (c_1^x - c_0^x - pc_0^u)^2. \quad (\text{B.1})$$

Zum Bestimmen der optimalen Lösung sowie der Sensitivitäten wird die erste Ableitungen der Zielfunktion F bezüglich p sowie die zweiten Ableitungen bezüglich p und den Koeffizienten c_0^x , c_1^x und c_0^u berechnet:

$$\begin{aligned}\frac{\partial F}{\partial p}(p, c_1^x, c_0^x, c_0^u) &= -(1 - pc_0^u) = -\left(1 - \frac{p}{2}\right), \\ \frac{\partial^2 F}{\partial p^2}(p, c_1^x, c_0^x, c_0^u) &= 2(c_0^u)^2 = \frac{1}{2}, \\ \frac{\partial^2 F}{\partial p \partial c_0^x}(p, c_1^x, c_0^x, c_0^u) &= 2c_0^u = 1, \\ \frac{\partial^2 F}{\partial p \partial c_1^x}(p, c_1^x, c_0^x, c_0^u) &= -2c_0^u = -1, \\ \frac{\partial^2 F}{\partial p \partial c_0^u}(p, c_1^x, c_0^x, c_0^u) &= -2(c_1^x - c_0^x - 2pc_0^u) = 2.\end{aligned}$$

Dabei wurden bereits die Nominalwerte der Koeffizienten $c_0^{x,0} = 0$, $c_1^{x,0} = 1$ und $c_0^{u,0} = 0.5$ eingesetzt. Die optimale Lösung von (B.1) ergibt sich aus $\frac{\partial F}{\partial p} = 0$ zu $p^* = 2$ und weil $\frac{\partial^2 F}{\partial p^2}$ positiv ist, handelt es sich bei p^* um eine Minimalstelle. Die Sensitivitäten der optimalen Lösung bezüglich der Koeffizienten sind gemäß [Korollar 2.14](#) durch

$$\begin{pmatrix} \frac{dp}{dc_0^x} \\ \frac{dp}{dc_1^x} \\ \frac{dp}{dc_0^u} \end{pmatrix} = - \left(\frac{\partial^2 F}{\partial p^2} \right)^{-1} \begin{pmatrix} \frac{\partial^2 F}{\partial p \partial c_0^x} \\ \frac{\partial^2 F}{\partial p \partial c_1^x} \\ \frac{\partial^2 F}{\partial p \partial c_0^u} \end{pmatrix} \Leftrightarrow \begin{pmatrix} \frac{dp}{dc_0^x} \\ \frac{dp}{dc_1^x} \\ \frac{dp}{dc_0^u} \end{pmatrix} = -2 \begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix} = \begin{pmatrix} -2 \\ 2 \\ -4 \end{pmatrix}$$

gegeben.

B.2 Berechnung der Sensitivitäten bezüglich der Datenpunkte

Im Folgenden werden die Sensitivitäten

$$\frac{dc_0^x}{d\bar{x}_k}(c^{x,0}, \bar{x}^0), \quad \frac{dc_1^x}{d\bar{x}_k}(c^{x,0}, \bar{x}^0) \quad \text{und} \quad \frac{dc_0^u}{d\bar{u}_k}(c^{u,0}, \bar{u}^0) \quad \text{für} \quad k = 1, \dots, n_t$$

bestimmt. Dies geschieht durch Lösen der linearen Gleichungssysteme (5.16). Für die Sensitivitäten $\frac{dc_0^x}{d\bar{x}_k}$ und $\frac{dc_1^x}{d\bar{x}_k}$ vereinfacht sich (5.16) wegen $B_{0,1,\tau^x}(t) = (1 - t)$ und

$B_{1,1,\tau^x}(t) = t$ zu

$$\begin{aligned}
& \begin{pmatrix} \sum_{i=1}^{n_t} (1-t_i)^2 & \sum_{i=1}^{n_t} (1-t_i)t_i \\ \sum_{i=1}^{n_t} (1-t_i)t_i & \sum_{i=1}^{n_t} t_i^2 \end{pmatrix} \begin{pmatrix} \frac{dc_0^x}{d\bar{x}_k} \\ \frac{dc_1^x}{d\bar{x}_k} \end{pmatrix} = \begin{pmatrix} 1-t_k \\ t_k \end{pmatrix} \\
\Leftrightarrow & \begin{pmatrix} \sum_{i=1}^{n_t} 1 - 2t_i + t_i^2 & \sum_{i=1}^{n_t} t_i - t_i^2 \\ \sum_{i=1}^{n_t} t_i - t_i^2 & \sum_{i=1}^{n_t} t_i^2 \end{pmatrix} \begin{pmatrix} \frac{dc_0^x}{d\bar{x}_k} \\ \frac{dc_1^x}{d\bar{x}_k} \end{pmatrix} = \begin{pmatrix} 1-t_k \\ t_k \end{pmatrix} \\
\stackrel{I+II}{\Leftrightarrow} & \begin{pmatrix} n_t - \sum_{i=1}^{n_t} t_i & \sum_{i=1}^{n_t} t_i \\ \sum_{i=1}^{n_t} t_i - \sum_{i=1}^{n_t} t_i^2 & \sum_{i=1}^{n_t} t_i^2 \end{pmatrix} \begin{pmatrix} \frac{dc_0^x}{d\bar{x}_k} \\ \frac{dc_1^x}{d\bar{x}_k} \end{pmatrix} = \begin{pmatrix} 1 \\ t_k \end{pmatrix}. \tag{B.2}
\end{aligned}$$

Es werden n_t äquidistante diskrete Zeitpunkte $t_k \in [0, 1]$ betrachtet, folglich ist $t_k = \frac{k-1}{n_t-1}$ und es gilt

$$\sum_{i=1}^{n_t} t_i = \sum_{i=1}^{n_t} \frac{i-1}{n_t-1} = \frac{1}{n_t-1} \sum_{i=0}^{n_t-1} i = \frac{1}{n_t-1} \frac{n_t(n_t-1)}{2} = \frac{n_t}{2} \tag{B.3}$$

und

$$\begin{aligned}
\sum_{i=1}^{n_t} t_i^2 &= \sum_{i=1}^{n_t} \left(\frac{i-1}{n_t-1} \right)^2 = \frac{1}{(n_t-1)^2} \sum_{i=0}^{n_t-1} i^2 = \frac{1}{(n_t-1)^2} \frac{(n_t-1)n_t(2n_t-1)}{6} \\
&= \frac{n_t(2n_t-1)}{6(n_t-1)}. \tag{B.4}
\end{aligned}$$

Mit (B.3) und (B.4) vereinfacht sich (B.2) zu

$$\begin{aligned}
& \begin{pmatrix} \frac{n_t}{2} & \frac{n_t}{2} \\ \frac{n_t}{2} - \frac{n_t(2n_t-1)}{6(n_t-1)} & \frac{n_t(2n_t-1)}{6(n_t-1)} \end{pmatrix} \begin{pmatrix} \frac{dc_0^x}{d\bar{x}_k} \\ \frac{dc_1^x}{d\bar{x}_k} \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{k-1}{n_t-1} \end{pmatrix} \\
\Leftrightarrow & \begin{pmatrix} n_t & n_t \\ 3n_t(n_t-1) - n_t(2n_t-1) & n_t(2n_t-1) \end{pmatrix} \begin{pmatrix} \frac{dc_0^x}{d\bar{x}_k} \\ \frac{dc_1^x}{d\bar{x}_k} \end{pmatrix} = \begin{pmatrix} 2 \\ 6(k-1) \end{pmatrix} \\
\stackrel{II-(2n_t-1) \cdot I}{\Leftrightarrow} & \begin{pmatrix} n_t & n_t \\ -n_t(n_t+1) & 0 \end{pmatrix} \begin{pmatrix} \frac{dc_0^x}{d\bar{x}_k} \\ \frac{dc_1^x}{d\bar{x}_k} \end{pmatrix} = \begin{pmatrix} 2 \\ 6(k-1) - 2(2n_t-1) \end{pmatrix}. \tag{B.5}
\end{aligned}$$

Aus (B.5) folgt schließlich

$$\frac{dc_0^x}{d\bar{x}_k} = \frac{6(k-1) - 2(2n_t-1)}{-n_t(n_t+1)}$$

und

$$\frac{dc_1^x}{d\bar{x}_k} = \frac{1}{n_t} \left(2 - n_t \frac{dc_0^x}{d\bar{x}_k} \right) = \frac{2}{n_t} - \frac{6(k-1) - 2(2n_t-1)}{-n_t(n_t+1)}$$

für $k = 1, \dots, n_t$.

Für die Sensitivitäten $\frac{dc_0^u}{d\bar{u}_k}$ vereinfacht sich (5.16) wegen $B_{0,0,\tau^u}(t) = 1$ zu

$$\sum_{i=1}^{n_t} 1 \cdot \frac{dc_0^u}{d\bar{u}_k} = 1 \quad \Leftrightarrow \quad n_t \cdot \frac{dc_0^u}{d\bar{u}_k} = 1$$

und folglich ist

$$\frac{dc_0^u}{d\bar{u}_k} = \frac{1}{n_t} \quad \text{für } k = 1, \dots, n_t.$$

Anhang C

Ergänzungen zur Analyse des Einflusses von Daten auf die Parameter eines Einspurmodells

Im Folgenden sind Details zur Analyse des Einflusses von Daten auf die Modellparameter eines Einspurmodells aufgeführt (vgl. [Abschnitt 6.3](#)).

C.1 Details zur Approximation der Daten

In diesem Abschnitt sind Details zu den Approximationen $\tilde{\chi}_{\text{app}}$ für die Modellgrößen $\chi \in \{x, y, v, \psi, \delta, a, \sigma\}$ des Einspurmodells (6.15) zusammengefasst. Die Grade der Approximationen sind

$$d^x = 3, d^y = 3, d^v = 1, d^\psi = 2, d^\delta = 1, d^a = 0 \text{ und } d^\sigma = 0.$$

In den folgenden Tabellen sind die Sprungstellen ξ^x , Regularitätseigenschaften an den Sprungstellen ν^x , Knotenpunkte τ^x und Koeffizienten c^x der Approximationen aufgelistet.

i	ξ_i^x	ν_i^x	τ_i^x	c_i^x
0	0.0000000000000000	0	0.0000000000000000	0.8233728302784916
1	1.6666666666666667	3	0.0000000000000000	-0.3458803620812444
2	3.3333333333333333	3	0.0000000000000000	1.2081674330196661
3	5.0000000000000000	2	0.0000000000000000	-1.3377738084411328
4	5.765403199752827	3	1.6666666666666667	0.2111075960126559
5	6.530806399505653	3	3.3333333333333333	0.0384074997845061
6	7.296209599258479	3	5.0000000000000000	0.1609247570949964
7	8.061612799011305	3	5.0000000000000000	0.8773015186701724

8	8.827015998764130	3	5.765403199752827	2.1580344867860390
9	9.592419198516957	2	6.530806399505653	4.8444388569447720
10	11.259085865183625	3	7.296209599258479	9.0356228445559830
11	12.925752531850291	3	8.061612799011305	11.2927635642672840
12	14.592419198516957	0	8.827015998764130	14.3382716899776470
13	-	-	9.592419198516957	19.4252501301093100
14	-	-	9.592419198516957	22.5360564003642130
15	-	-	11.259085865183625	23.6396371995396500
16	-	-	12.925752531850291	23.1738442851478760
17	-	-	14.592419198516957	-
18	-	-	14.592419198516957	-
19	-	-	14.592419198516957	-
20	-	-	14.592419198516957	-

i	ξ_i^y	ν_i^y	τ_i^y	c_i^y
0	0.0000000000000000	0	0.0000000000000000	-0.1289100558299209
1	1.6666666666666667	3	0.0000000000000000	-0.1957613184615900
2	3.3333333333333333	3	0.0000000000000000	0.9828516045468526
3	5.0000000000000000	2	0.0000000000000000	4.6389188885206190
4	5.765403199752827	3	1.6666666666666667	8.6948555292594600
5	6.530806399505653	3	3.3333333333333333	11.9960395322939830
6	7.296209599258479	3	5.0000000000000000	14.0518217373830550
7	8.061612799011305	3	5.0000000000000000	18.6612180763473300
8	8.827015998764130	3	5.765403199752827	20.9140809259723460
9	9.592419198516957	2	6.530806399505653	22.1710177921394660
10	11.259085865183625	3	7.296209599258479	24.0899978928042520
11	12.925752531850291	3	8.061612799011305	23.5013233191785400
12	14.592419198516957	0	8.827015998764130	23.5724189741839820
13	-	-	9.592419198516957	23.2418621014381100
14	-	-	9.592419198516957	24.5796234375123180
15	-	-	11.259085865183625	23.2514363196137100
16	-	-	12.925752531850291	23.1760089336846420
17	-	-	14.592419198516957	-
18	-	-	14.592419198516957	-
19	-	-	14.592419198516957	-
20	-	-	14.592419198516957	-

i	ξ_i^v	ν_i^v	τ_i^v	c_i^v
0	0.0000000000000000	0	0.0000000000000000	0.0581780639049942
1	1.6666666666666667	1	0.0000000000000000	1.5016914806638777

2	3.333333333333333	1	1.666666666666667	3.0234435923373777
3	5.000000000000000	1	3.333333333333333	4.5232668099996200
4	5.765403199752827	1	5.000000000000000	4.5388037223044440
5	6.530806399505653	1	5.765403199752827	4.5682614201249580
6	7.296209599258479	1	6.530806399505653	4.4537239272147700
7	8.061612799011305	1	7.296209599258479	4.5039850386794040
8	8.827015998764130	1	8.061612799011305	4.4829744817264950
9	9.592419198516957	1	8.827015998764130	4.4030529615470740
10	11.259085865183625	1	9.592419198516957	2.9690102946904986
11	12.925752531850291	1	11.259085865183625	1.5219104378565087
12	14.592419198516957	0	12.925752531850291	-0.0256615253084002
13	-	-	14.592419198516957	-
14	-	-	14.592419198516957	-

i	ξ_i^ψ	ν_i^ψ	τ_i^ψ	c_i^ψ
0	0.000000000000000	0	0.000000000000000	1.5727823596133248
1	1.666666666666667	2	0.000000000000000	1.5751780951332620
2	3.333333333333333	2	0.000000000000000	1.5720273546127925
3	5.000000000000000	2	1.666666666666667	1.5639087763595910
4	5.765403199752827	2	3.333333333333333	1.5785219964808737
5	6.530806399505653	2	5.000000000000000	1.3991307836488270
6	7.296209599258479	2	5.765403199752827	1.0444699053304405
7	8.061612799011305	2	6.530806399505653	0.5227021838200059
8	8.827015998764130	2	7.296209599258479	0.1701103650958495
9	9.592419198516957	2	8.061612799011305	-0.0128688237384354
10	11.259085865183625	2	8.827015998764130	0.0001383411152317
11	12.925752531850291	2	9.592419198516957	-0.0003141224215418
12	14.592419198516957	0	11.259085865183625	-0.0111882256544886
13	-	-	12.925752531850291	-0.0026171702171879
14	-	-	14.592419198516957	-
15	-	-	14.592419198516957	-
16	-	-	14.592419198516957	-

i	ξ_i^δ	ν_i^δ	τ_i^δ	c_i^δ
0	0.000000000000000	0	0.000000000000000	-0.000587918250056000
1	1.666666666666667	1	0.000000000000000	0.000497165025709300
2	3.333333333333333	1	1.666666666666667	0.000294778186075011
3	5.000000000000000	1	3.333333333333333	0.002378791229287679
4	5.765403199752827	1	5.000000000000000	-0.148962026362392780
5	6.530806399505653	1	5.765403199752827	-0.291256457637693530

6	7.296209599258479	1	6.530806399505653	-0.410766166167301700
7	8.061612799011305	1	7.296209599258479	-0.272944562582395000
8	8.827015998764130	1	8.061612799011305	-0.140117012389432700
9	9.592419198516957	1	8.827015998764130	0.002273415904535285
10	11.259085865183625	1	9.592419198516957	0.011913241400745630
11	12.925752531850291	1	11.259085865183625	0.009850487703722885
12	14.592419198516957	0	12.925752531850291	0.002769670190432274
13	-	-	14.592419198516957	-
14	-	-	14.592419198516957	-

i	ξ_i^δ	ν_i^δ	τ_i^δ	c_i^δ
0	0.0000000000000000	0	0.0000000000000000	1.000051399699038600
1	1.6666666666666667	0	1.6666666666666667	0.997318215823377800
2	3.3333333333333333	0	3.3333333333333333	1.009019121541835200
3	5.0000000000000000	0	5.0000000000000000	0.000519761578060544
4	5.765403199752827	0	5.765403199752827	-0.020473405488347585
5	6.530806399505653	0	6.530806399505653	0.002599049602537977
6	7.296209599258479	0	7.296209599258479	0.002852908541849933
7	8.061612799011305	0	8.061612799011305	-0.016998774280937712
8	8.827015998764130	0	8.827015998764130	-0.006950445486662794
9	9.592419198516957	0	9.592419198516957	-0.988803686814095600
10	11.259085865183625	0	11.259085865183625	-0.997146606429260100
11	12.925752531850291	0	12.925752531850291	-0.987723897530637900
12	14.592419198516957	0	14.592419198516957	-

i	ξ_i^σ	ν_i^σ	τ_i^σ	c_i^σ
0	0.0000000000000000	0	0.0000000000000000	0.000745593022248768
1	1.6666666666666667	0	1.6666666666666667	-0.001384785899423004
2	3.3333333333333333	0	3.3333333333333333	-0.001240298046138092
3	5.0000000000000000	0	5.0000000000000000	-0.299945843169477800
4	5.765403199752827	0	5.765403199752827	-0.297977080239261450
5	6.530806399505653	0	6.530806399505653	-0.301359974934755860
6	7.296209599258479	0	7.296209599258479	0.298465939573267950
7	8.061612799011305	0	8.061612799011305	0.298749775026320940
8	8.827015998764130	0	8.827015998764130	0.300821428564610470
9	9.592419198516957	0	9.592419198516957	-0.001240250347994771
10	11.259085865183625	0	11.259085865183625	-0.000764742499357510
11	12.925752531850291	0	12.925752531850291	-0.000500678599492079
12	14.592419198516957	0	14.592419198516957	-

C.2 Ergänzende Abbildungen des mittleren absoluten Einflusses der Daten auf den Modellparameter r

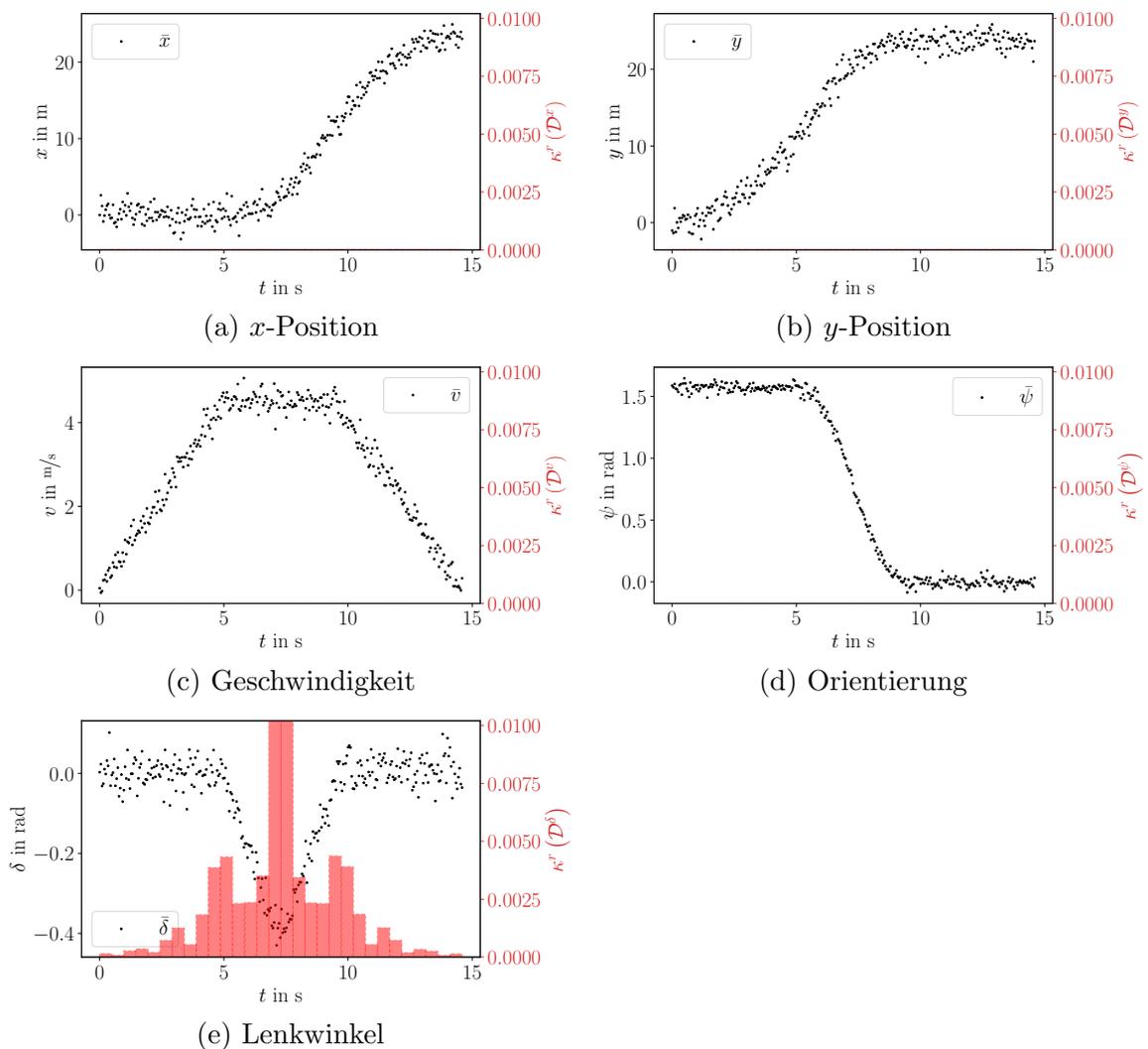


Abbildung C.1: Mittlere absolute Einflüsse der Daten des Einspurmodells auf den Modellparameter r , die Daten sind gemäß (6.20) in 30 gleich große Bereiche aufgeteilt (Teil 1)

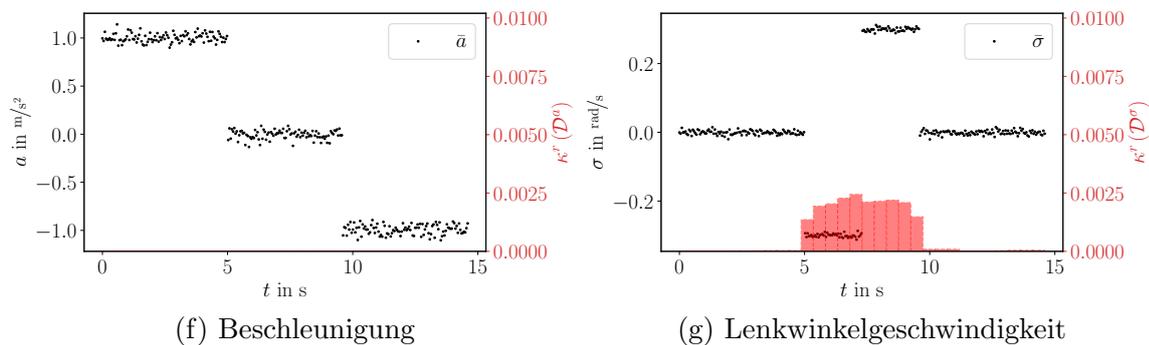


Abbildung C.1: Mittlere absolute Einflüsse der Daten des Einspurmodells auf den Modellparameter r , die Daten sind gemäß (6.20) in 30 gleich große Bereiche aufgeteilt (Teil 2)

Literatur

- [1] E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney und D. Sorensen. *LAPACK Users' Guide*. Society for Industrial und Applied Mathematics, Jan. 1999. DOI: [10.1137/1.9780898719604](https://doi.org/10.1137/1.9780898719604).
- [2] R. C. Aster, B. Borchers und C. H. Thurber. *Parameter Estimation and Inverse Problems*. 3. Aufl. Elsevier, 2019. DOI: [10.1016/c2015-0-02458-3](https://doi.org/10.1016/c2015-0-02458-3).
- [3] J. T. Betts. *Practical Methods for Optimal Control and Estimation Using Non-linear Programming*. 2. Aufl. Philadelphia: Society for Industrial und Applied Mathematics, Jan. 2010. DOI: [10.1137/1.9780898718577](https://doi.org/10.1137/1.9780898718577).
- [4] Å. Björck. „A General Updating Algorithm for Constrained Linear Least Squares Problems“. In: *SIAM Journal on Scientific and Statistical Computing* 5.2 (Juni 1984), S. 394–402. DOI: [10.1137/0905029](https://doi.org/10.1137/0905029).
- [5] Å. Björck. *Numerical Methods for Least Squares Problems*. Society for Industrial und Applied Mathematics, Jan. 1996. DOI: [10.1137/1.9781611971484](https://doi.org/10.1137/1.9781611971484).
- [6] H. G. Bock, E. Kostina und J. P. Schlöder. „Direct Multiple Shooting and Generalized Gauss-Newton Method for Parameter Estimation Problems in ODE Models“. In: *Contributions in Mathematical and Computational Sciences*. Springer International Publishing, 2015, S. 1–34. DOI: [10.1007/978-3-319-23321-5_1](https://doi.org/10.1007/978-3-319-23321-5_1).
- [7] C. de Boor. *A Practical Guide to Splines*. en. 1. Aufl. Bd. 27. Applied Mathematical Sciences. New York: Springer-Verlag, 2001. ISBN: 978-0-3879-5366-3. URL: <https://www.springer.com/de/book/9780387953663>.
- [8] C. de Boor. „On calculating with B-splines“. en. In: *Journal of Approximation Theory* 6.1 (Juli 1972), S. 50–62. DOI: [10.1016/0021-9045\(72\)90080-9](https://doi.org/10.1016/0021-9045(72)90080-9).
- [9] C. de Boor, T. Lyche und L. L. Schumaker. „On Calculating with B-Splines II. Integration“. en. In: *Numerical Methods of Approximation Theory*. Hrsg. von L. Collatz, H. Werner und G. Meinardus. International Series of Numerical Mathematics. Basel: Birkhäuser, 1976, S. 123–146. DOI: [10.1007/978-3-0348-7692-6_6](https://doi.org/10.1007/978-3-0348-7692-6_6).

- [10] P. Businger und G. H. Golub. „Linear least squares solutions by householder transformations“. In: *Numerische Mathematik* 7.3 (Juni 1965), S. 269–276. DOI: [10.1007/bf01436084](https://doi.org/10.1007/bf01436084).
- [11] C. Büskens. „Real-Time Solutions for Perturbed Optimal Control Problems by a Mixed Open- and Closed-Loop Strategy“. In: *Online Optimization of Large Scale Systems*. Springer Berlin Heidelberg, 2001, S. 105–116. DOI: [10.1007/978-3-662-04331-8_7](https://doi.org/10.1007/978-3-662-04331-8_7).
- [12] C. Büskens und H. Maurer. „Sensitivity Analysis and Real-Time Optimization of Parametric Nonlinear Programming Problems“. In: *Online Optimization of Large Scale Systems*. Springer Berlin Heidelberg, 2001, S. 3–16. DOI: [10.1007/978-3-662-04331-8_1](https://doi.org/10.1007/978-3-662-04331-8_1).
- [13] C. Büskens und D. Wassel. „The ESA NLP Solver WORHP“. In: *Springer Optimization and Its Applications*. Springer New York, 2012, S. 85–110. DOI: [10.1007/978-1-4614-4469-5_4](https://doi.org/10.1007/978-1-4614-4469-5_4).
- [14] R. H. Byrd, J. Nocedal und R. A. Waltz. „Knitro: An Integrated Package for Nonlinear Optimization“. In: *Nonconvex Optimization and Its Applications*. Springer US, 2006, S. 35–59. DOI: [10.1007/0-387-30065-1_4](https://doi.org/10.1007/0-387-30065-1_4).
- [15] P. Deuffhard und F. Bornemann. *Gewöhnliche Differentialgleichungen*. 4. Aufl. De Gruyter, Jan. 2013. DOI: [10.1515/9783110316360](https://doi.org/10.1515/9783110316360).
- [16] M. Echim. „Modellbasierte optimale Mehrgrößenregelung eines aufgeladenen Dieselmotors mittels Methoden der nichtlinearen Optimierung“. Diss. Universität Bremen, Okt. 2014. URL: <http://nbn-resolving.de/urn:nbn:de:gbv:46-00104021-10>.
- [17] A. Fiacco. *Introduction to sensitivity and stability analysis in nonlinear programming*. New York: Academic Press, 1983. ISBN: 978-0-1225-4450-7.
- [18] R. Fletcher und S. Leyffer. „Nonlinear programming without a penalty function“. In: *Mathematical Programming* 91.2 (Jan. 2002), S. 239–269. DOI: [10.1007/s101070100244](https://doi.org/10.1007/s101070100244).
- [19] D. C.-L. Fong und M. Saunders. „LSMR: An Iterative Algorithm for Sparse Least-Squares Problems“. In: *SIAM Journal on Scientific Computing* 33.5 (Jan. 2011), S. 2950–2971. DOI: [10.1137/10079687x](https://doi.org/10.1137/10079687x).
- [20] O. Forster. *Analysis 2*. 10. Aufl. Springer Fachmedien Wiesbaden, 2013. DOI: [10.1007/978-3-658-02357-7](https://doi.org/10.1007/978-3-658-02357-7).
- [21] H. Gaff und E. Schaefer. „Optimal control applied to vaccination and treatment strategies for various epidemiological models“. In: *Mathematical Biosciences and Engineering* 6.3 (2009), S. 469–492. DOI: [10.3934/mbe.2009.6.469](https://doi.org/10.3934/mbe.2009.6.469).
- [22] S. Geffken. „Effizienzsteigerung numerischer Verfahren der nichtlinearen Optimierung“. Diss. Universität Bremen, Aug. 2017. URL: <http://nbn-resolving.de/urn:nbn:de:gbv:46-00106042-11>.

-
- [23] S. Geffken und C. Büskens. „Worhp Multi-Core Interface, Parallelisation Approaches for an NLP Solver“. In: *Proceedings of the 6th International Conference on Astrodynamics Tools and Techniques, 14.03. - 17.03.2016, Darmstadt, Germany*. 2016. URL: <https://indico.esa.int/indico/event/111/session/25/contribution/54/material/paper/0.pdf>.
- [24] C. Geiger und C. Kanzow. *Theorie und Numerik restringierter Optimierungsaufgaben*. Berlin, Heidelberg: Springer, 2002. DOI: [10.1007/978-3-642-56004-0](https://doi.org/10.1007/978-3-642-56004-0).
- [25] H.-O. Georgii. *Stochastik*. 5. Aufl. De Gruyter, Jan. 2015. DOI: [10.1515/9783110359701](https://doi.org/10.1515/9783110359701).
- [26] P. E. Gill, W. Murray und M. A. Saunders. „SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization“. In: *SIAM Review* 47.1 (Jan. 2005), S. 99–131. DOI: [10.1137/s0036144504446096](https://doi.org/10.1137/s0036144504446096).
- [27] M. Gulliksson und P.-å. Wedin. „Modifying the QR-Decomposition to Constrained and Weighted Linear Least Squares“. In: *SIAM Journal on Matrix Analysis and Applications* 13.4 (Okt. 1992), S. 1298–1313. DOI: [10.1137/0613079](https://doi.org/10.1137/0613079).
- [28] S.-P. Han. „A globally convergent method for nonlinear programming“. In: *Journal of Optimization Theory and Applications* 22.3 (Juli 1977), S. 297–309. DOI: [10.1007/bf00932858](https://doi.org/10.1007/bf00932858).
- [29] M. T. Heath. „Numerical Methods for Large Sparse Linear Least Squares Problems“. In: *SIAM Journal on Scientific and Statistical Computing* 5.3 (Sep. 1984), S. 497–513. DOI: [10.1137/0905037](https://doi.org/10.1137/0905037).
- [30] A. Hindmarsh. *ODEPACK. A Collection of ODE System Solvers*. Jan. 1992. URL: <https://www.osti.gov/biblio/145725>.
- [31] A. C. Hindmarsh und L. R. Petzold. *LSODA, Ordinary Differential Equation Solver for Stiff or Non-Stiff System*. Sep. 2005. URL: <https://www.osti.gov/etdeweb/biblio/21352532>.
- [32] J. D. Hunter. „Matplotlib: A 2D Graphics Environment“. In: *Computing in Science & Engineering* 9.3 (2007), S. 90–95. DOI: [10.1109/mcse.2007.55](https://doi.org/10.1109/mcse.2007.55).
- [33] M. D. Intriligator. *Mathematical Optimization and Economic Theory*. Society for Industrial und Applied Mathematics, Jan. 2002. DOI: [10.1137/1.9780898719215](https://doi.org/10.1137/1.9780898719215).
- [34] D. Irony und S. Toledo. „The Snap-Back Pivoting Method for Symmetric Banded Indefinite Matrices“. In: *SIAM Journal on Matrix Analysis and Applications* 28.2 (Jan. 2006), S. 398–424. DOI: [10.1137/040610106](https://doi.org/10.1137/040610106).
- [35] R. E. Kalman. „A New Approach to Linear Filtering and Prediction Problems“. In: *Journal of Basic Engineering* 82.1 (März 1960), S. 35–45. DOI: [10.1115/1.3662552](https://doi.org/10.1115/1.3662552).

-
- [36] P. Kalmbach. „Effiziente Ableitungsbestimmung bei hochdimensionaler nichtlinearer Optimierung“. Diss. Universität Bremen, Juni 2011. URL: <http://nbn-resolving.de/urn:nbn:de:gbv:46-00102087-10>.
- [37] L. Kaufman. „The retraction algorithm for factoring banded symmetric matrices“. In: *Numerical Linear Algebra with Applications* 14.3 (2007), S. 237–254. DOI: [10.1002/nla.529](https://doi.org/10.1002/nla.529).
- [38] R. Kuhlmann. „A Primal-Dual Augmented Lagrangian Penalty-Interior-Point Algorithm for Nonlinear Programming“. Diss. Universität Bremen, Dez. 2018. URL: <http://nbn-resolving.de/urn:nbn:de:gbv:46-00106924-13>.
- [39] R. Kuhlmann, S. Geffken und C. Büskens. „WORHP Zen: Parametric Sensitivity Analysis for the Nonlinear Programming Solver WORHP“. In: *Operations Research Proceedings*. Springer International Publishing, 2018, S. 649–654. DOI: [10.1007/978-3-319-89920-6_86](https://doi.org/10.1007/978-3-319-89920-6_86).
- [40] K. Levenberg. „A method for the solution of certain non-linear problems in least squares“. In: *Quarterly of Applied Mathematics* 2.2 (Juli 1944), S. 164–168. DOI: [10.1090/qam/10666](https://doi.org/10.1090/qam/10666).
- [41] F. L. Lewis, D. L. Vrabie und V. L. Syrmos. *Optimal Control*. John Wiley & Sons, Inc., Jan. 2012. DOI: [10.1002/9781118122631](https://doi.org/10.1002/9781118122631).
- [42] F. Lin, K. Muthuraman und M. Lawley. „An optimal control theory approach to non-pharmaceutical interventions“. In: *BMC Infectious Diseases* 10.1 (Feb. 2010). DOI: [10.1186/1471-2334-10-32](https://doi.org/10.1186/1471-2334-10-32).
- [43] N. Mahdavi-Amiri und M. R. Ansari. „Superlinearly convergent exact penalty projected structured Hessian updating schemes for constrained nonlinear least squares: asymptotic analysis“. In: *Bulletin of the Iranian Mathematical Society* 38.3 (2012), S. 767–786. ISSN: 1017-060X. URL: http://bims.iranjournals.ir/article_240.html.
- [44] N. Mahdavi-Amiri und R. H. Bartels. „Constrained nonlinear least squares: an exact penalty approach with projected structured quasi-Newton updates“. In: *ACM Transactions on Mathematical Software* 15.3 (Sep. 1989), S. 220–242. DOI: [10.1145/66888.66891](https://doi.org/10.1145/66888.66891).
- [45] D. W. Marquardt. „An Algorithm for Least-Squares Estimation of Nonlinear Parameters“. In: *Journal of the Society for Industrial and Applied Mathematics* 11.2 (Juni 1963), S. 431–441. DOI: [10.1137/0111030](https://doi.org/10.1137/0111030).
- [46] T. Nikolayzik. „Korrekturverfahren zur numerischen Lösung nichtlinearer Optimierungsprobleme mittels Methoden der parametrischen Sensitivitätsanalyse“. Diss. Universität Bremen, Feb. 2012. URL: <http://nbn-resolving.de/urn:nbn:de:gbv:46-00102497-19>.
- [47] J. Nocedal und S. Wright. *Numerical Optimization*. 2. Aufl. Springer Series in Operations Research and Financial Engineering. New York: Springer, 2006. DOI: [10.1007/978-0-387-40065-5](https://doi.org/10.1007/978-0-387-40065-5).

-
- [48] T. E. Oliphant. *A guide to NumPy*. Bd. 1. Trelgol Publishing USA, 2006. ISBN: 978-1-5173-0007-4.
- [49] C. C. Paige und M. A. Saunders. „Solution of Sparse Indefinite Systems of Linear Equations“. In: *SIAM Journal on Numerical Analysis* 12.4 (Sep. 1975), S. 617–629. DOI: [10.1137/0712047](https://doi.org/10.1137/0712047).
- [50] M. J. D. Powell. „A fast algorithm for nonlinearly constrained optimization calculations“. In: *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, 1978, S. 144–157. DOI: [10.1007/bfb0067703](https://doi.org/10.1007/bfb0067703).
- [51] M. J. D. Powell. „The convergence of variable metric methods for nonlinearly constrained optimization calculations“. In: *Nonlinear Programming 3*. Academic Press, 1978, S. 27–63. DOI: [10.1016/b978-0-12-468660-1.50007-4](https://doi.org/10.1016/b978-0-12-468660-1.50007-4).
- [52] A. A. Poyton, M. S. Varziri, K. B. McAuley, P. J. McLellan und J. O. Ramsay. „Parameter estimation in continuous-time dynamic models using principal differential analysis“. In: *Computers & Chemical Engineering* 30.4 (Feb. 2006), S. 698–708. DOI: [10.1016/j.compchemeng.2005.11.008](https://doi.org/10.1016/j.compchemeng.2005.11.008).
- [53] F. Pukelsheim. *Optimal Design of Experiments*. Society for Industrial and Applied Mathematics, Jan. 2006. DOI: [10.1137/1.9780898719109](https://doi.org/10.1137/1.9780898719109).
- [54] J. O. Ramsay. „Principal Differential Analysis: Data Reduction by Differential Operators“. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.3 (Sep. 1996), S. 495–508. DOI: [10.1111/j.2517-6161.1996.tb02096.x](https://doi.org/10.1111/j.2517-6161.1996.tb02096.x).
- [55] J. O. Ramsay, G. Hooker, D. Campbell und J. Cao. „Parameter estimation for differential equations: a generalized smoothing approach“. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69.5 (Okt. 2007), S. 741–796. DOI: [10.1111/j.1467-9868.2007.00610.x](https://doi.org/10.1111/j.1467-9868.2007.00610.x).
- [56] J. O. Ramsay und B. W. Silverman. *Functional Data Analysis*. Springer New York, 2005. DOI: [10.1007/b98888](https://doi.org/10.1007/b98888).
- [57] C. R. Rao. „Minimum variance and the estimation of several parameters“. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 43.2 (Apr. 1947), S. 280–283. DOI: [10.1017/s0305004100023471](https://doi.org/10.1017/s0305004100023471).
- [58] S. Rauski. „Limited Memory BFGS method for Sparse and Large-Scale Nonlinear Optimization“. Diss. Universität Bremen, Sep. 2014. URL: <http://nbn-resolving.de/urn:nbn:de:gbv:46-00104042-12>.
- [59] M. Rick, J. Clemens, L. Sommer, A. Folkers, K. Schill und C. Büskens. „Autonomous Driving Based on Nonlinear Model Predictive Control and Multi-Sensor Fusion“. In: *IFAC-PapersOnLine* 52.8 (2019), S. 182–187. DOI: [10.1016/j.ifacol.2019.08.068](https://doi.org/10.1016/j.ifacol.2019.08.068).
- [60] R. Schaback und H. Wendland. *Numerische Mathematik*. 5. Aufl. Heidelberg: Springer-Verlag, 2005. DOI: [10.1007/b137970](https://doi.org/10.1007/b137970).

- [61] K. Scherer und A. Y. Shadrin. „New Upper Bound for the B-Spline Basis Condition Number“. In: *Journal of Approximation Theory* 99.2 (Aug. 1999), S. 217–229. DOI: [10.1006/jath.1998.3310](https://doi.org/10.1006/jath.1998.3310).
- [62] K. Schittkowski. „NLPQL: A fortran subroutine solving constrained nonlinear programming problems“. In: *Annals of Operations Research* 5.2 (Juni 1986), S. 485–500. DOI: [10.1007/bf02022087](https://doi.org/10.1007/bf02022087).
- [63] K. Schittkowski. *Numerical Data Fitting in Dynamical Systems*. 1. Aufl. Springer Science & Business Media, 2002. DOI: [10.1007/978-1-4419-5762-7](https://doi.org/10.1007/978-1-4419-5762-7).
- [64] K. Schittkowski. „On the convergence of a sequential quadratic programming method with an augmented lagrangian line search function“. In: *Mathematische Operationsforschung und Statistik. Series Optimization* 14.2 (Jan. 1983), S. 197–216. DOI: [10.1080/02331938308842847](https://doi.org/10.1080/02331938308842847).
- [65] K. Schittkowski. „Solving Constrained Nonlinear Least Squares Problems by a General Purpose SQP-Method“. In: *Trends in Mathematical Optimization*. Birkhäuser Basel, 1988, S. 295–309. DOI: [10.1007/978-3-0348-9297-1_19](https://doi.org/10.1007/978-3-0348-9297-1_19).
- [66] D. Schramm, M. Hiller und R. Bardini. *Modellbildung und Simulation der Dynamik von Kraftfahrzeugen*. Springer Berlin Heidelberg, 2018. DOI: [10.1007/978-3-662-54481-5](https://doi.org/10.1007/978-3-662-54481-5).
- [67] Schütze, Torsten and Schwetlick, Hubert. „Constrained approximation by splines with free knots“. In: *BIT Numerical Mathematics* 37.1 (März 1997), S. 105–137. DOI: [10.1007/bf02510176](https://doi.org/10.1007/bf02510176).
- [68] H. R. Schwarz und N. Köckler. *Numerische Mathematik*. 8. Aufl. Wiesbaden: Vieweg+Teubner Verlag, 2011. DOI: [10.1007/978-3-8348-8166-3](https://doi.org/10.1007/978-3-8348-8166-3).
- [69] Schwetlick, Hubert and Schütze, Torsten. „Least squares approximation by splines with free knots“. In: *BIT Numerical Mathematics* 35.3 (Sep. 1995), S. 361–384. DOI: [10.1007/bf01732610](https://doi.org/10.1007/bf01732610).
- [70] D. Seelbinder. „On-board Trajectory Computation for Mars Atmospheric Entry Based on Parametric Sensitivity Analysis of Optimal Control Problems“. Diss. Universität Bremen, Aug. 2017. URL: <http://nbn-resolving.de/urn:nbn:de:gbv:46-00106076-16>.
- [71] D. Simon. „Nonlinear Kalman filtering“. In: *Optimal State Estimation*. John Wiley & Sons, Inc., Jan. 2006. Kap. 13, S. 393–431. DOI: [10.1002/0470045345.ch13](https://doi.org/10.1002/0470045345.ch13).
- [72] D. Simon. *Optimal State Estimation*. John Wiley & Sons, Inc., Jan. 2006. DOI: [10.1002/0470045345](https://doi.org/10.1002/0470045345).
- [73] J. Swartz und H. Bremermann. „Discussion of parameter estimation in biological modelling: Algorithms for estimation and evaluation of the estimates“. In: *Journal of Mathematical Biology* 1.3 (Sep. 1975), S. 241–257. DOI: [10.1007/bf01273746](https://doi.org/10.1007/bf01273746).

- [74] I. B. Tjoa und L. T. Biegler. „Simultaneous solution and optimization strategies for parameter estimation of differential-algebraic equation systems“. In: *Industrial & Engineering Chemistry Research* 30.2 (Feb. 1991), S. 376–385. DOI: [10.1021/ie00050a015](https://doi.org/10.1021/ie00050a015).
- [75] T. H. Tsang, D. M. Himmelblau und T. F. Edgar. „Optimal control via collocation and non-linear programming“. In: *International Journal of Control* 21.5 (Mai 1975), S. 763–768. DOI: [10.1080/00207177508922030](https://doi.org/10.1080/00207177508922030).
- [76] J. M. Varah. „A Spline Least Squares Method for Numerical Parameter Estimation in Differential Equations“. In: *SIAM Journal on Scientific and Statistical Computing* 3.1 (1982), S. 28–46. DOI: [10.1137/0903003](https://doi.org/10.1137/0903003).
- [77] M. S. Varziri, A. A. Poyton, K. B. McAuley, P. J. McLellan und J. O. Ramsay. „Selecting optimal weighting factors in iPDA for parameter estimation in continuous-time dynamic models“. In: *Computers & Chemical Engineering* 32.12 (Dez. 2008), S. 3011–3022. DOI: [10.1016/j.compchemeng.2008.04.005](https://doi.org/10.1016/j.compchemeng.2008.04.005).
- [78] T. P. Velavan und C. G. Meyer. „The COVID-19 epidemic“. In: *Tropical Medicine & International Health* 25.3 (Feb. 2020), S. 278–280. DOI: [10.1111/tmi.13383](https://doi.org/10.1111/tmi.13383).
- [79] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde u. a. „SciPy 1.0: fundamental algorithms for scientific computing in Python“. In: *Nature Methods* 17.3 (Feb. 2020), S. 261–272. DOI: [10.1038/s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2).
- [80] H. U. Voss, J. Timmer und J. Kurths. „Nonlinear dynamical system identification from uncertain and indirect measurements“. In: *International Journal of Bifurcation and Chaos* 14.06 (Juni 2004), S. 1905–1933. DOI: [10.1142/s0218127404010345](https://doi.org/10.1142/s0218127404010345).
- [81] E. Vynnycky und R. White. *An introduction to infectious disease modelling*. Oxford: Oxford University Press, 2010. ISBN: 978-0-1985-6576-5.
- [82] S. van der Walt, S. C. Colbert und G. Varoquaux. „The NumPy Array: A Structure for Efficient Numerical Computation“. In: *Computing in Science & Engineering* 13.2 (März 2011), S. 22–30. DOI: [10.1109/mcse.2011.37](https://doi.org/10.1109/mcse.2011.37).
- [83] D. Wassel. „Exploring novel designs of NLP solvers: Architecture and Implementation of WORHP“. Diss. Universität Bremen, Apr. 2013. URL: <http://nbn-resolving.de/urn:nbn:de:gbv:46-00103260-11>.
- [84] M. Zamir, Z. Shah, F. Nadeem, A. Memood, H. Alrabaiiah und P. Kumam. „Non Pharmaceutical Interventions for Optimal Control of COVID-19“. In: *Computer Methods and Programs in Biomedicine* 196 (Nov. 2020). DOI: [10.1016/j.cmpb.2020.105642](https://doi.org/10.1016/j.cmpb.2020.105642).