

**UNIVERSITÄT  
BAYREUTH**

FAKULTÄT FÜR MATHEMATIK UND PHYSIK  
MATHEMATISCHES INSTITUT

# **Digitale Implementierung kontinuierlicher Regler: Analyse und Optimierung**

Diplomarbeit

von

Karl Worthmann

Datum: 17. Juli 2006

Aufgabenstellung / Betreuung:  
Prof. Dr. L. Grüne



# Danksagung

An dieser Stelle möchte ich mich bei Herrn Professor Dr. Grüne für die hervorragende Betreuung während der Diplomarbeit und das Heranführen an den Themenbereich der Optimalsteuerung durch seine Vorlesungen und Seminare bedanken.

Mein besonderer Dank gilt auch meiner Familie und meiner Verlobten Nataliya Samiylenko für die Unterstützung während meines gesamten Studiums. Außerdem möchte ich mich bei Ekue-sse Situ Tomety bedanken, der mir immer mit Rat und Tat zur Seite stand.



# Inhaltsverzeichnis

|                                                               |           |
|---------------------------------------------------------------|-----------|
| Abbildungsverzeichnis                                         | III       |
| Tabellenverzeichnis                                           | V         |
| <b>1 Einleitung</b>                                           | <b>1</b>  |
| <b>2 Kontrollsysteme und Fließentwicklung</b>                 | <b>3</b>  |
| 2.1 Grundlegende Definitionen . . . . .                       | 3         |
| 2.2 Kontrollsysteme . . . . .                                 | 6         |
| 2.3 Fließentwicklung . . . . .                                | 9         |
| 2.4 Sampled-data Systeme und Problemstellung . . . . .        | 16        |
| <b>3 Fließ- und Taylorentwicklung des closed-loop Systems</b> | <b>19</b> |
| 3.1 Fließentwicklung eines sampled-data Systems . . . . .     | 19        |
| 3.2 Fließentwicklung des zeitdiskreten Systems . . . . .      | 21        |
| 3.3 Taylorentwicklung des kontinuierlichen Systems . . . . .  | 24        |
| 3.4 Vergleich von Fließ- und Taylorentwicklung . . . . .      | 30        |
| <b>4 Hinreichende Bedingungen</b>                             | <b>39</b> |
| 4.1 für beliebige Ordnung . . . . .                           | 39        |
| 4.2 Abschwächung für endliche Ordnung . . . . .               | 47        |
| <b>5 Notwendige und hinreichende Bedingungen</b>              | <b>55</b> |
| 5.1 für eine beliebige Output-Funktion $h$ . . . . .          | 55        |
| 5.2 für die Lösung $\Phi$ der DGL . . . . .                   | 60        |
| <b>6 Beispiele</b>                                            | <b>65</b> |
| 6.1 Akademisches Beispiel . . . . .                           | 65        |
| 6.2 Moore-Greitzer Flugzeugmotor . . . . .                    | 71        |

|          |                                                                     |            |
|----------|---------------------------------------------------------------------|------------|
| <b>7</b> | <b>Matlab-Simulationen zur Evaluation der Methode</b>               | <b>73</b>  |
| 7.1      | Vergleich der sampled-data Lösung mit der exakten Lösung . . . . .  | 73         |
| 7.2      | Optimierungsansatz zur Berechnung einer Steuerung . . . . .         | 74         |
| 7.3      | Moore-Greitzer Modell: sampled-data und optimierte Lösung . . . . . | 77         |
| 7.4      | Akademisches Beispiel: sampled-data und optimierte Lösung . . . . . | 86         |
| 7.5      | Fazit und Ausblick . . . . .                                        | 90         |
| <b>A</b> | <b>Programmparameter und Matlabcode</b>                             | <b>93</b>  |
| A.1      | Programmparameter . . . . .                                         | 93         |
| A.2      | Matlabcode . . . . .                                                | 95         |
| A.3      | Matlabfehler . . . . .                                              | 112        |
| <b>B</b> | <b>Tabellen</b>                                                     | <b>115</b> |
| B.1      | Akademisches Beispiel . . . . .                                     | 115        |
| B.2      | Moore-Greitzer Modell . . . . .                                     | 115        |
| <b>C</b> | <b>Material auf der beiliegenden CD</b>                             | <b>119</b> |
|          | <b>Literaturverzeichnis</b>                                         | <b>121</b> |

# Abbildungsverzeichnis

|     |                                                                                                                                                                                                     |    |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 6.1 | akademisches Beispiel: Vergleich der zeitkontinuierlichen und der sampled-data Lösung mit Regler (6.2, o), (6.3, x) und (6.4, $\diamond$ ). . . . .                                                 | 66 |
| 6.2 | akademisches Beispiel: Differenz zwischen der zeitkontinuierlichen und der sampled-data Lösung bei $t = 1$ für die sampled-data Feedbacks (6.2, o), (6.3, x) und (6.4, $\diamond$ ). . . . .        | 67 |
| 6.3 | akademisches Beispiel: Differenz zwischen der zeitkontinuierlichen und der sampled-data Lösung für die sampled-data Feedbacks (6.4, x), (6.6, $\diamond$ ) und (6.8, o). . . . .                    | 70 |
| 6.4 | akademisches Beispiel: Differenz zwischen der zeitkontinuierlichen und der sampled-data Lösung bei $t = 1$ für die sampled-data Feedbacks (6.2, $\diamond$ ), (6.4), (6.6, x) und (6.8, o). . . . . | 71 |
| 6.5 | Vergleich der zeitkontinuierlichen und der sampled-data Lösung für das Moore-Greitzer Modell eines Flugzeugmotors. . . . .                                                                          | 72 |
| 7.1 | Moore-Greitzer Modell mit Gauß-Newton optimiert; $T=0.2$ , max. eine Iteration mit „cubicpoly“ als linesearch. . . . .                                                                              | 78 |
| 7.2 | Moore-Greitzer Modell mit Gauß-Newton optimiert; $T=0.2$ , max. zwei Iterationen mit „cubicpoly“ als linesearch. . . . .                                                                            | 79 |
| 7.3 | Moore-Greitzer Modell mit Gauß-Newton optimiert; $T=0.2$ , max. drei Iterationen mit „cubicpoly“ als linesearch. . . . .                                                                            | 79 |
| 7.4 | Moore-Greitzer Modell mit Levenberg-Marquardt optimiert; $T=0.2$ , max. eine Iteration mit „cubicpoly“ als linesearch. . . . .                                                                      | 80 |
| 7.5 | Moore-Greitzer Modell mit Levenberg-Marquardt optimiert; $T=0.2$ , max. zwei Iterationen mit „cubicpoly“ als linesearch. . . . .                                                                    | 80 |
| 7.6 | Moore-Greitzer Modell mit Levenberg-Marquardt optimiert; $T=0.2$ , max. drei Iterationen mit „cubicpoly“ als linesearch. . . . .                                                                    | 81 |
| 7.7 | Moore-Greitzer Modell mit Levenberg-Marquardt optimiert; $T=0.2$ , max. vier Iterationen mit „cubicpoly“ als linesearch. . . . .                                                                    | 81 |
| 7.8 | Moore-Greitzer Modell mit Levenberg-Marquardt optimiert; $T=0.2$ , max. fünf Iterationen mit „cubicpoly“ als linesearch. . . . .                                                                    | 82 |

|      |                                                                                                                                                                                                                                                      |    |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 7.9  | Moore-Greitzer Modell mit Gauß-Newton optimiert; $T=0.2$ , max. eine Iteration mit „quadcubic“ als linesearch. . . . .                                                                                                                               | 83 |
| 7.10 | Moore-Greitzer Modell mit Gauß-Newton optimiert; $T=0.2$ , max. zwei Iterationen mit „quadcubic“ als linesearch. . . . .                                                                                                                             | 83 |
| 7.11 | Moore-Greitzer Modell mit Levenberg-Marquardt optimiert; $T=0.2$ , max. eine Iteration mit „quadcubic“ als linesearch. . . . .                                                                                                                       | 84 |
| 7.12 | Moore-Greitzer Modell mit Levenberg-Marquardt optimiert; $T=0.2$ , max. zwei Iterationen mit „quadcubic“ als linesearch. . . . .                                                                                                                     | 84 |
| 7.13 | Vergleich der sampled-data Lösung ( $\diamond$ ) mit optimierten Trajektorien. Levenberg-Marquardt mit einer ( $x$ ) beziehungsweise drei ( $o$ ) Iterationen. . . . .                                                                               | 86 |
| 7.14 | Abstand der sampled-data ( $x$ ) beziehungsweise der optimierten Lösung ( $o$ ) von der exakten Trajektorie mit Samplingperiode $T = 0.2$ . Optimierung ausgehend von (6.6) mit maximal zwei Iterationen und vier Messpunkten pro Intervall. . . . . | 87 |
| 7.15 | Abstand der sampled-data ( $x$ ) beziehungsweise der optimierten Lösung ( $o$ ) von der exakten Trajektorie mit Samplingperiode $T = 0.2$ . Optimierung ausgehend von (6.6) mit maximal zwei Iterationen und acht Messpunkten pro Intervall. . . . . | 89 |
| 7.16 | Abstand der sampled-data ( $x$ ) beziehungsweise der optimierten Lösung ( $o$ ) von der exakten Trajektorie mit Samplingperiode $T = 0.1$ . Optimierung ausgehend von (6.6) mit maximal zwei Iterationen und vier Messpunkten pro Intervall. . . . . | 90 |



# Tabellenverzeichnis

|     |                                                                                                                                                             |     |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 6.1 | Differenz zwischen der zeitkontinuierlichen und der sampled-data Lösung mit Regler (6.2) - (6.4) für das akademische Beispiel. . . . .                      | 67  |
| 6.2 | Differenz zwischen der zeitkontinuierlichen und der sampled-data Lösung mit Regler (6.4), (6.6) und (6.8) für das akademische Beispiel. . . . .             | 70  |
| 7.1 | Vergleich der Residuen der beiden Algorithmen mit beziehungsweise ohne Anfangswert. Linesearch: cubicpoly. . . . .                                          | 82  |
| 7.2 | Vergleich der Residuen der beiden Algorithmen mit beziehungsweise ohne Anfangswert. Linesearch: quadcubic. . . . .                                          | 85  |
| 7.3 | Residuen der sampled-data Lösung für vier bzw. acht Auswertungspunkte. . . . .                                                                              | 86  |
| 7.4 | Residuen der Nachoptimierung ausgehend von (6.3) mit verschiedener Gewichtung und vier Auswertungspunkten. . . . .                                          | 88  |
| 7.5 | Residuen der Nachoptimierung ausgehend von (6.6) mit verschiedener Gewichtung und vier Auswertungspunkten. . . . .                                          | 88  |
| 7.6 | Residuen der Nachoptimierung ausgehend von (6.6) mit zwei beziehungsweise vier Messpunkten pro Samplingintervall; $T=0.1$ , 2 Optimierungsschritte. . . . . | 89  |
| B.1 | Residuen der Nachoptimierung ausgehend von (6.3) mit verschiedener Gewichtung und acht Auswertungspunkten. . . . .                                          | 115 |
| B.2 | Residuen der Nachoptimierung ausgehend von (6.6) mit verschiedener Gewichtung und acht Auswertungspunkten. . . . .                                          | 115 |
| B.3 | Vergleich der Optimalsteuerung mit den Werten des sampled-data Feedbacks. Linesearch: cubicpoly. . . . .                                                    | 116 |
| B.4 | Vergleich des Aufwandes gemessen in Funktionsauswertungen der beiden Algorithmen mit bzw. ohne Anfangswert. Linesearch: cubicpoly. . . . .                  | 117 |
| B.5 | Vergleich des Aufwandes gemessen in Funktionsauswertungen der beiden Algorithmen mit bzw. ohne Anfangswert. Linesearch: quadcubic. . . . .                  | 117 |



# Kapitel 1

## Einleitung

Sowohl in den Natur- als auch in den Wirtschaftswissenschaften gibt es viele Probleme, die in Form von Kontrollsystemen modelliert werden. Auf der Basis dieser Modelle kann man Kontrollfunktionen finden, welche - könnte man sie kontinuierlich implementieren - gewisse Aufgaben bezüglich der Steuerung dieser Systeme erfüllen. Allerdings ist diese kontinuierliche Umsetzung im Allgemeinen nicht möglich, weil Rechner nur in diskreter Zeit arbeiten können. Deshalb bedient man sich so genannter sampled-data Systeme und berechnet die Steuerung nur an den Anfangspunkten der jeweiligen Samplingintervalle, also nur zu diskreten Zeitpunkten. Die so erhaltenen Regler verwendet man für die gesamte nächste Samplingperiode. Eine erneute Auswertung des Systemzustandes ist erst bei Erreichen des nächsten Intervallendes möglich. Darauf basierend kann die Steuerung neu berechnet werden, ehe sie abermals für den kompletten nächsten Abschnitt vorgegeben werden muss. Entsprechend kann es passieren, dass die so erhaltene sampled-data Feedback-Steuerung nicht mit der kontinuierlichen Regelung übereinstimmt. Infolge dessen kann das Verhalten des gesteuerten von dem des zeitkontinuierlichen Systems abweichen, weil man im Zuge von Hardwarelimitationen nicht beliebig oft steuern kann. Deshalb kann es nötig sein, die Steuerung an die Erfordernisse der digitalen Implementierung anzupassen.

Genau an dieser Stelle setzt diese Arbeit ein. Sie ist in zwei Teile gegliedert. Der erste Teil ist theoretischer Natur und beschäftigt sich damit, die Abweichungen des diskretisierten von dem zeitkontinuierlichen System bis zu einer vorgegebenen Ordnung exakt darzustellen. Darauf basierend werden Aussagen gemacht, unter welchen Voraussetzungen man auftretende Abweichungen durch eine Restrukturierung der Feedback-Steuerung bis zu einer gewissen Ordnung in den Griff bekommen kann. Hierbei wird vorausgesetzt, dass die Samplingperiode hinreichend klein ist. Anschließend wird anhand von konkreten Beispielen untersucht, welchen Wert diese Resultate in der Praxis haben, womit der zweite Teil dieser Arbeit beginnt. Ausgehend von einer gefundenen sampled-data Lösung wird versucht, bessere Resultate zu erzielen, indem man eine Nachoptimierung auf die berechnete Steuerung anwendet. Entschei-

dend dabei ist, dass der Aufwand dieser Nachoptimierung nicht übermäßig groß wird.

Um diese beiden Themenbereiche behandeln zu können, werden in Kapitel 2 einige grundlegende Resultate und Definitionen gegeben. Vor allem die Begriffe der Richtungsableitung, des Kontrollsystems und des sampled-data Systems sind für die weitere Arbeit von großer Bedeutung. Abschnitt 2.3 führt zudem die Fließentwicklung als für Kontrollsysteme besonders geeignete Form einer Reihenentwicklung ein. Aufbauend auf Kapitel 2 und einem Theorem in Abschnitt 3.1 werden im Rest des 3. Kapitels Reihenentwicklungen des kontinuierlichen und des zeitdiskreten Systems vorgenommen, um in Abschnitt 3.4 erste Resultate bezüglich der Differenzen herzuleiten. Kapitel 4 liefert eine hinreichende Bedingung, unter der das kontinuierliche System durch das sampled-data System mit beliebiger Ordnung approximiert werden kann. Abschnitt 4.2 zeigt Probleme auf, die auftreten, wenn man versucht, diese Bedingung für konkrete Ordnungen abzuschwächen. Zum Abschluß des theoretischen Teils dieser Arbeit werden in Kapitel 5 notwendige Bedingungen gesucht. Diese Vorgehensweise ermöglicht eine wesentlich größere Flexibilität bei der Wahl der Feedback-Steuerungen. Im Anschluss daran werden in Kapitel 6 zwei Beispiele betrachtet, an denen Anwendungsmöglichkeiten der zuvor bewiesenen Resultate demonstriert werden. Im Zuge dessen wird auch die zuvor getroffene Annahme hinreichend kleiner Samplingperioden aufgehoben. Entsprechend kann man neben den Möglichkeiten auch die Grenzen der hergeleiteten Resultate in der praktischen Anwendung beobachten. Deshalb wird im 7. Kapitel ein Optimierungsansatz verfolgt, der genau die Probleme des sampled-data Ansatzes kompensieren soll. Mit Hilfe dieses Ansatzes könnte man, basierend auf der sampled-data Lösung, zum Beispiel Startlösungen für die modellprädiktive Steuerung gewinnen.

Abschließend bedarf es noch einer kurzen Anmerkung, die den Zusammenhang dieser Arbeit mit [11] klären soll. Aufbauend auf [21, Remark 4.13]<sup>1</sup> ist eine Zusammenarbeit mit den beiden Autoren Herrn Prof. Dr. Grüne und Herrn Dr. Nešić dieses Artikels entstanden. Aus dieser ist der Artikel [11] entstanden, in den einige der hier aufgeführten Resultate eingegangen sind. Ebenso lieferte dieser Artikel Beiträge zu dieser Arbeit. Dies gilt insbesondere für Teile des 5. Kapitels.

---

<sup>1</sup>siehe [11] für genauere Ausführungen

# Kapitel 2

## Kontrollsysteme und Fließentwicklung

In Kapitel 2 soll die Problemstellung dieser Arbeit formuliert werden. Bevor dies möglich ist, werden einige grundlegende Resultate benötigt, die in den Abschnitten 2.1 bis 2.3 bereitgestellt werden. So dient Abschnitt 2.1 in erster Linie zur Einführung einiger Schreibweisen und Definitionen, insbesondere des Begriffs der Richtungsableitung, und eines kombinatorischen Lemmas. In dem darauf folgenden Abschnitt werden die Begriffe der gewöhnlichen Differentialgleichung und des Kontrollsystems definiert, sowie deren Lösbarkeit ganz allgemein untersucht. Abschnitt 2.3 widmet sich in Anlehnung an [12] der Fließentwicklung. Im abschließenden Teil dieses Kapitels werden sampled-data Systeme eingeführt und darauf basierend die zentrale Fragestellung dieser Arbeit formuliert und erläutert.

### 2.1 Grundlegende Definitionen

Die Menge der natürlichen Zahlen wird mit  $\mathbb{N}$ , die der ganzen Zahlen mit  $\mathbb{Z}$  und die Menge der reellen Zahlen mit  $\mathbb{R}$  bezeichnet. Für  $\mathbb{N} \cup \{0\}$  wird  $\mathbb{N}_0$  geschrieben.

**Definition 2.1** Für einen Vektor  $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$  bezeichne

$$\|x\| = \sqrt{\sum_{i=1}^n |x_i|^2}$$

die euklidische Norm und

$$\|x\|_\infty = \max\{|x_i|\}$$

die Maximumsnorm, wobei  $\cdot^T$  transponiert bedeutet.

Es wird ein kombinatorisches Resultat aus [15] benötigt, in dessen Formulierung die folgende Definition eingeht.

**Definition 2.2** *Mit*

$$\binom{n}{n_0 n_1 \dots n_M} := \frac{n!}{n_0! n_1! \dots n_M!},$$

wird ein *Multinomialindex* definiert. Zudem wird ein *Multiindex*  $\nu := (n_0, n_1, \dots, n_M)$  benötigt, für den die Ausdrücke

$$|\nu| := \sum_{i=0}^M n_i \quad \text{und} \quad (2.1)$$

$$\|\nu\| := \sum_{j=0}^M j \cdot n_j \quad (2.2)$$

definiert werden.

**Lemma 2.3** *Seien  $a_i, i = 0, 1, 2, \dots, M$ , reelle Zahlen und  $n \in \mathbb{N}$ . Dann gilt*

$$(a_0 + a_1 + \dots + a_M)^n = \sum_{\substack{|\nu|=n \\ n_0, n_1, \dots, n_M \geq 0}} \binom{n}{n_0 n_1 \dots n_M} a_0^{n_0} \dots a_M^{n_M}.$$

*Beweis:* Betrachte das Produkt

$$(a_0^1 + a_1^1 + a_2^1 + \dots + a_M^1)(a_0^2 + a_1^2 + a_2^2 + \dots + a_M^2) \dots (a_0^n + a_1^n + a_2^n + \dots + a_M^n), \quad (2.3)$$

wobei das  $j$  von  $a_i^j$  ein weiterer Index und kein Exponent einer Potenz ist. Ein Term dieses Produktes hat die Form

$$(a_0^{i_1} a_0^{i_2} \dots a_0^{i_{n_0}})(a_1^{j_1} a_1^{j_2} \dots a_1^{j_{n_1}}) \dots (a_M^{k_1} a_M^{k_2} \dots a_M^{k_{n_M}}), \quad (2.4)$$

wobei  $n_0 + n_1 + \dots + n_M = n$  und  $n_i \geq 0$  für alle  $i = 0, 1, \dots, M$ . Fixiere nun  $n_0, n_1, \dots, n_M$ . Dann entspricht jedes Monom vom Typ (2.4) eindeutig einer Sortierung der Menge  $X := \{1, 2, \dots, n\}$  in  $M+1$  Boxen, so dass die  $i$ -te Box genau eine  $n_i$ -elementige Teilmenge von  $X$  enthält. Hierbei gibt  $n_i$  an, wie oft dabei ein  $a_i^j$  aus (2.3) ausgewählt worden ist. Umgekehrt legt eine solche Sortierung ein Monom des Typs (2.4) eindeutig fest. So könnten beispielsweise der  $i$ -ten Box die Elemente 2, 5 und  $n-1$  zugeordnet werden, was den Faktoren  $a_i^2, a_i^5$  und  $a_i^{n-1}$  im Monom (2.4) entspräche. Die Anzahl der Monome, die zu der fixierten Partition  $n_0 + n_1 + \dots + n_M = n$  gehören ist folglich

$$\frac{n!}{n_0! n_1! \dots n_M!} = \binom{n}{n_0 n_1 \dots n_M}. \quad (2.5)$$

Die Summe aller solcher Monome mit Vielfachheit entsprechend dem Multinomialkoeffizienten (2.5) ist das Ergebnis von (2.3). Setzt man nun  $a_i^1 = a_i^2 = \dots = a_i^n = a_i$  in (2.3) ein, erhält man die Behauptung des Lemmas.  $\square$

Bevor der für diese Arbeit sehr wichtige Begriff der Richtungsableitung eingeführt und an einem Beispiel erläutert wird, soll noch kurz an einige in Analogie zu [5] definierte Objekte erinnert werden. Es wird dabei vorausgesetzt, dass die Begriffe des Vektorfeldes, sowie der partiellen und totalen Ableitung bekannt sind.

**Definition 2.4** Sei  $U \subset \mathbb{R}^n$  offen und  $h : U \rightarrow \mathbb{R}$  partiell differenzierbar. Dann heißt der Vektor

$$\text{grad } h(x) := \left( \frac{\partial h}{\partial x_1}(x), \dots, \frac{\partial h}{\partial x_n}(x) \right) \quad (2.6)$$

Gradient von  $h$  im Punkt  $x \in U$ . Anstelle von  $\text{grad } h$  kann man auch  $\nabla h$  schreiben.  $\nabla$  ist ein vektorwertiger Differentialoperator.

**Erinnerung 2.5** Das Differential bzw. die Jacobi-Matrix (oder auch Funktional-Matrix) eines Vektorfeldes  $f : U \rightarrow V$  im Punkte  $x$  ist gegeben durch

$$(Df)(x) := J_f(x) := \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \dots & \frac{\partial f_1}{\partial x_n}(x) \\ \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_1}(x) & \dots & \frac{\partial f_m}{\partial x_n}(x) \end{pmatrix}, \quad (2.7)$$

wobei  $U \subset \mathbb{R}^n$  und  $V \subset \mathbb{R}^m$ .

**Definition 2.6** Sei  $h : \mathbb{R}^n \rightarrow \mathbb{R}$  eine differenzierbare Funktion und  $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$  ein Vektorfeld. Dann bezeichnet

$$L_f h := \text{grad } h \cdot f = \left( \frac{\partial h}{\partial x_1}, \dots, \frac{\partial h}{\partial x_n} \right) \begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} = \sum_{i=1}^n \frac{\partial h}{\partial x_i} f_i \quad (2.8)$$

die Richtungsableitung von  $h$  in Richtung von  $f$ . In dieser Schreibweise wurde das Argument  $x$  aus notationstechnischen Gründen weggelassen.

Um den Gebrauch der Richtungsableitung zu demonstrieren, dient das folgende Beispiel. Dabei werden  $h$  und  $f$  analog zu Definition 2.6 verwendet und  $g$  ist, ebenso wie  $f$ , ein Vektorfeld. Es wird vorausgesetzt, dass alle Ableitungen wohldefiniert sind.

**Beispiel 2.7**

$$\begin{aligned}
L_f L_g h &\stackrel{(2.8)}{=} \sum_{j=1}^n \frac{\partial}{\partial x_j} \left( \sum_{i=1}^n \frac{\partial h}{\partial x_i} g_i \right) f_j \\
&= \sum_{j=1}^n \sum_{i=1}^n \left( \frac{\partial^2 h}{\partial x_j \partial x_i} g_i f_j + \frac{\partial h}{\partial x_i} \frac{\partial g_i}{\partial x_j} f_j \right) \\
&= \sum_{j=1}^n \sum_{i=1}^n \left( \frac{\partial^2 h}{\partial x_j \partial x_i} g_i f_j \right) + \sum_{i=1}^n \frac{\partial h}{\partial x_i} \left( \sum_{j=1}^n \frac{\partial g_i}{\partial x_j} f_j \right) \\
&= g^T \frac{d^2}{dx^2} h f + \nabla h J_g f
\end{aligned} \tag{2.9}$$

**Bemerkung 2.8** In Anlehnung an [20] kann man Potenzen des Differentialoperators  $L_f$  definieren. Dafür setzt man  $L_f^0 = 1$  und definiert anschließend

$$L_f^p h := \frac{\partial L_f^{p-1} h}{\partial x} f$$

für  $p \in \mathbb{N}$ . Es ist sogar möglich eine Exponentialreihe für diesen Operator durch

$$e^{L_f} = 1 + \sum_{p \geq 1} \frac{L_f^p}{p!}$$

einzuführen.

## 2.2 Kontrollsysteme

Bevor wir in der Lage sind, Kontrollsysteme zu definieren, benötigen wir einige Resultate aus der Theorie der gewöhnlichen Differentialgleichungen.

**Definition 2.9** Eine gewöhnliche Differentialgleichung (DGL) im  $\mathbb{R}^n$ ,  $n \in \mathbb{N}$ , ist gegeben durch die Gleichung

$$\frac{d}{dt} x(t) = f(t, x(t)), \tag{2.10}$$

wobei  $f : D \rightarrow \mathbb{R}^n$  eine stetige Funktion ist und Vektorfeld genannt wird, deren Definitionsbereich  $D$  eine offene Teilmenge von  $\mathbb{R} \times \mathbb{R}^n$  ist. Abkürzend wird die Kurzschreibweise  $\dot{x}(t)$  für  $\frac{d}{dt} x(t)$  verwendet.

**Bemerkung 2.10** Falls das Vektorfeld  $f$  in (2.10) nicht explizit zeitabhängig ist, also  $\dot{x}(t) = f(x(t))$  ist, spricht man von einer autonomen Differentialgleichung. Die rechte Seite solch einer Differentialgleichung ist gegenüber einer Substitution der Zeitvariablen  $t$  durch  $\tau = t - a$  invariant.



Um die Eindeutigkeit einer Lösung einer gewöhnlichen Differentialgleichung der Form (2.10) garantieren zu können, benötigt man die Anfangsbedingung

$$x(t_0) = x_0. \quad (2.11)$$

Zusammen mit (2.10) nennt man (2.11) ein Anfangswertproblem. Für das Anfangswertproblem (2.10), (2.11) kann nun folgender Satz formuliert werden, der die Frage nach der Eindeutigkeit einer Lösung beantwortet.

**Satz 2.11** *Betrachte die gewöhnliche Differentialgleichung (2.10) für ein  $f : D \rightarrow \mathbb{R}^n$  mit  $D \subset \mathbb{R} \times \mathbb{R}^n$ . Das Vektorfeld  $f$  sei stetig, darüberhinaus sei  $f$  Lipschitz-stetig im zweiten Argument im folgenden Sinne: Für jede kompakte Teilmenge  $K \subset D$  existiere eine Konstante  $L > 0$ , so dass die Ungleichung*

$$\| f(t, x) - f(t, y) \| \leq L \| x - y \| \quad (2.12)$$

für alle  $t \in \mathbb{R}$  und  $x, y$  mit  $(t, x), (t, y) \in K$  gilt. Dann gibt es für jede Anfangsbedingung  $(t_0, x_0) \in D$  genau eine Lösung  $x(t; t_0, x_0)$  des Anfangswertproblems (2.10), (2.11). Diese ist definiert für alle  $t$  aus einem offenen maximalen Existenzintervall  $I_{t_0, x_0} \subseteq \mathbb{R}$  mit  $t_0 \in I_{t_0, x_0}$ .

Dieser Satz ist zusammen mit einem Beweis in [7] zu finden. Eine Version, die lediglich stückweise Stetigkeit in  $t$  voraussetzt findet sich zusammen mit ergänzenden Ausführungen zur Lipschitz-Bedingung in [13]<sup>1</sup>. Nun können wir analog zu [9]<sup>2</sup> und [24]<sup>3</sup> ein Kontrollsystem sowohl in kontinuierlicher als auch diskreter Zeit definieren.

**Definition 2.12** *Seien  $m, n \in \mathbb{N}$  natürliche Zahlen und  $U \subset \mathbb{R}^m$ . Dann ist ein Kontrollsystem in kontinuierlicher Zeit  $\mathbb{T} = \mathbb{R}$  im  $\mathbb{R}^n$  gegeben durch die gewöhnliche Differentialgleichung*

$$\frac{d}{dt}x(t) = f(x(t), u(t)), \quad (2.13)$$

wobei  $f : \mathbb{R}^n \times U \rightarrow \mathbb{R}^n$  ein Vektorfeld ist, das bezüglich  $x(t)$  und  $u(t)$  stetig partiell differenzierbar ist.

Ein Kontrollsystem in diskreter Zeit  $\mathbb{T} = h\mathbb{Z} = \{hk \mid k \in \mathbb{Z}\}$  im  $\mathbb{R}^n$  für ein  $h \in \mathbb{R}$ ,  $h > 0$ , wird über die Differenzengleichung

$$x(t+h) = f_h(x(t), u(t)), \quad (2.14)$$

definiert, wobei  $f_h : \mathbb{R}^n \times U \rightarrow \mathbb{R}^n$  eine stetige Abbildung ist.

Hierbei bezeichnet  $U \subseteq \mathbb{R}^m$  den Kontrollwertebereich. Das ist die Menge aller Werte, die von der Steuerung  $u(t)$  angenommen werden dürfen. Mit  $\mathcal{U}$  bzw.  $\mathcal{U}_h$  werden die Räume der zulässigen Kontrollfunktionen für kontinuierliche bzw. diskrete Zeit bezeichnet, wobei es sich jeweils um Funktionen von  $\mathbb{T} \rightarrow U$  handelt. Was man genau unter Zulässigkeit zu verstehen hat, werden wir in diesem Abschnitt noch klären.

---

<sup>1</sup>Theorem 3.1

<sup>2</sup>Definition 1.1

<sup>3</sup>Definition 2.9

Man kann Kontrollsysteme zusätzlich noch explizit zeitabhängig definieren. Dann lautet die erzeugende Differentialgleichung in Analogie zu (2.14) für das kontinuierliche System

$$\frac{d}{dt}x(t) = f(t, x(t), u), \quad (2.15)$$

wobei

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad u = \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix}, \quad f(t, x(t), u) = \begin{pmatrix} f_1(t, x(t), u) \\ \vdots \\ f_n(t, x(t), u) \end{pmatrix}.$$

Im Allgemeinen bezeichnet  $t$  die Zeit,  $u_1, \dots, u_m$  bezeichnen die Systeminputs und  $x_1(t), \dots, x_n(t)$  charakterisieren den Systemzustand zur jeweiligen Zeit. Dabei ist es möglich  $u = u(t)$  als Funktion der Zeit bzw.  $u = u(x)$  als Zustandsfeedback oder sowohl zeit- als auch zustandsabhängig, also  $u = u(t, x)$ , zu wählen.

**Bemerkung 2.13** *Kontrollsysteme sind eine konkrete Ausprägung der allgemeineren dynamischen Systeme. Auf diese soll aber an dieser Stelle nicht näher eingegangen werden. Für die Definition eines dynamischen Systems sei auf [8] verwiesen.*

Bei (2.13) handelt es sich um ein autonomes Problem. Folglich kann man ohne Beschränkung der Allgemeinheit den Anfangszeitpunkt  $t_0 = 0$  setzen. Konkret wird in dieser Arbeit ein nichtlineares, affines Kontrollsystem der Form

$$\dot{x}(t) = g_0(x(t)) + g_1(x(t))u(t) \quad (2.16)$$

betrachtet, wobei  $g_0, g_1 : \mathbb{R}^n \rightarrow \mathbb{R}^n$  Vektorfelder und  $u(t) : \mathbb{R} \rightarrow \mathbb{R}$  eine Kontrollfunktion sind.

**Bemerkung 2.14** *Wie man bereits an (2.16) erkennen kann, beschränken wir uns auf den Fall  $m = 1$ . Dies dient lediglich der besseren Darstellbarkeit. Die Ergebnisse sind unter gewissem technischen Aufwand auf den höher dimensional Fall übertragbar.*

Zur Existenz und Eindeutigkeit einer Lösung von (2.13) und damit auch von (2.16) sei auf [24] verwiesen. In dieser Arbeit findet sich in Anlehnung an [9] eine Definition der Messbarkeit und der darauf aufbauende Satz von Caratheodory mit einer Beweisskizze und Hinweisen, wo ein ausführlicher Beweis zu finden ist. Benötigt wird dazu lediglich noch die Definition der Zulässigkeit. Für kontinuierliche Systeme seien alle messbaren Funktionen mit Werten in  $U$  zulässig. Für die diskreten Systeme bedarf es keinerlei Einschränkungen, um die Existenz und Eindeutigkeit der Lösungen zu zeigen. Die Lösung  $x_h : h\mathbb{N}_0 \times \mathbb{R}^n \times \mathcal{U}_h \rightarrow \mathbb{R}^n$  von (2.14) erfüllt für jeden Anfangswert  $x \in \mathbb{R}^n$  und jede Funktion  $u_h \in \mathcal{U}_h$  die Gleichungen

$$x_h(0, x, u_h) = x \quad \text{und} \quad x_h(t+h, x, u_h) = f_h(x_h(t, x, u_h), u_h(t)). \quad (2.17)$$

Es gibt verschiedene Möglichkeiten Steuerungen zu berechnen. So ist es zum Beispiel möglich, die Steuerung zeitabhängig zu wählen. Allerdings ergibt sich hier das Problem, dass bereits kleine Störungen sich langfristig gravierend auswirken können, weil keine nachträgliche Korrektur stattfindet, die auf diese Abweichungen reagiert. Deshalb wird in dieser Arbeit ein Zustandsfeedback betrachtet, d.h. die Steuerung wird nicht mehr im Voraus berechnet, sondern abhängig vom jeweils vorliegenden Zustand bestimmt.

**Definition 2.15** *Gegeben sei ein Kontrollsystem (2.13). Eine Abbildung  $u_0 : \mathbb{R}^n \rightarrow U$  heißt Zustandsfeedback oder Rückkopplung. Die zugehörige Differentialgleichung*

$$\dot{x}(t) = f(x(t), u_0(x(t))) \quad (2.18)$$

*bezeichnet man als Feedback-geregeltes, rückgekoppeltes oder geregeltes System. Desweiteren wird auch der aus dem Englischen geläufige Begriff des closed-loop Systems verwendet. Die Lösungen von (2.18) mit Anfangswert  $x_0$  und Anfangszeit  $t_0 = 0$  werden mit  $\Phi(t, x_0)$  bezeichnet.*

Im nächsten Abschnitt wird ein für diese Arbeit zentrales Hilfsmittel eingeführt, bevor die so genannten sampled-data Systeme definiert und damit der Fokus auf zeitdiskrete Kontrollsysteme gelegt wird. Im Anschluss daran sind wir in der Lage, unsere Problemstellung zu formulieren.

## 2.3 Fließentwicklung

Dieser Abschnitt orientiert sich an [12] und berücksichtigt Resultate und Ideen aus [10]. Anzumerken ist außerdem, dass in diesem Abschnitt analytische Funktionen und Vektorfelder verwendet werden. Entscheidend bezüglich dieser Eigenschaft ist für uns, dass eine analytische Funktion beziehungsweise ein analytisches Vektorfeld in jedem Punkt durch eine konvergente Potenzreihe entwickelbar und damit unter anderem beliebig oft differenzierbar ist.

Wir betrachten Systeme, die durch die Differentialgleichung

$$\begin{aligned} \dot{x} &= f(x) + \sum_{i=1}^m g_i(x)u_i \quad \text{und} \\ y_j &= h_j(x) \quad 1 \leq j \leq p \end{aligned} \quad (2.19)$$

gegeben und auf einer offenen Menge  $U \subset \mathbb{R}^n$  definiert sind. Zudem seien  $f, g_1, \dots, g_m$  analytische, auf  $U$  definierte Vektorfelder. Genauso seien die Output-Funktionen  $h_1, \dots, h_p$  analytische Funktionen auf  $U$ .

**Notation 2.16** *Es wird die Bezeichnung*

$$y = h(x) = (h_1(x), \dots, h_p(x))^T$$

*verwendet.*

**Kombinatorische Resultate:** Betrachte die Indexmenge

$$I = \{0, 1, \dots, m\}$$

mit  $(m+1)$ -Elementen und bezeichne mit  $I_k$  die Menge aller Folgen  $(i_1, \dots, i_k)$  mit  $k$  Elementen von  $I$ . Alternativ kann man für  $(i_1, \dots, i_k)$  auch kurz  $(i_k \dots i_1)$  schreiben. Ein Element aus  $I_k$  wird Multiindex der Länge  $k$  genannt. Zusätzlich definiere die Mengen  $I_0 := \emptyset$  und  $I^* := \bigcup_{k \geq 0} I_k$ . Auf der Menge  $I^*$  kann man einen freien Monoid mit der Verknüpfungsvorschrift

$$(i_k \dots i_1)(j_h \dots j_1) \mapsto (i_k \dots i_1 j_h \dots j_1)$$

und neutralem Element  $\emptyset$  definieren.

**Definition 2.17** *Eine formale Potenzreihe in  $(m+1)$ -nichtkommutativen Unbekannten und Koeffizienten in  $\mathbb{R}$  ist eine Abbildung*

$$c : I^* \mapsto \mathbb{R}.$$

*Wir bezeichnen den Wert von  $c$  bei einem Element  $(i_k \dots i_0) \in I^*$  mit dem Symbol  $c(i_k \dots i_0)$ .*

**Definition 2.18 (Iteriertes Integral)** *Wir definieren auf  $[0, T]$  ein iteriertes Integral einer gegebenen Menge von Funktionen. Für jeden Multiindex  $(i_k \dots i_0)$  ist das korrespondierende iterierte Integral eine reellwertige Funktion in  $t$*

$$E_{i_k \dots i_1 i_0}(t) = \int_0^t d\xi_{i_k} \dots d\xi_{i_1} d\xi_{i_0},$$

*definiert für  $0 \leq t \leq T$  durch iterative Reduktion der Länge. Dafür setzt man*

$$\begin{aligned} \xi_0(t) &= t \\ \xi_i(t) &= \int_0^t u_i(\lambda) d\lambda \end{aligned} \tag{2.20}$$

*und*

$$\int_0^t d\xi_{i_k} \dots d\xi_{i_0} = \int_0^t d\xi_{i_k}(\lambda) \int_0^\lambda d\xi_{i_{k-1}} \dots d\xi_{i_0}. \tag{2.21}$$

*Das zu  $I_0$  korrespondierende iterierte Integral ist die reelle Zahl 1.*

**Beispiel 2.19** Als erstes Beispiel betrachte man ein iteriertes Integral für einen Multiindex der Länge eins. Dieses Beispiel folgt nahezu direkt aus der Definition. Es dient in erster Linie zur detaillierten Aufschlüsselung einiger notationstechnischer Besonderheiten in [12].

$$\begin{aligned}
\int_0^t d\xi_{i_0} &\stackrel{(2.21)}{=} \int_0^t 1 \cdot d\xi_{i_0}(\lambda) \\
&\stackrel{(2.20)}{=} \begin{cases} \int_0^t 1 \cdot d\lambda & \text{für } i_0 = 0 \\ \int_0^t 1 \cdot d \int_0^\lambda u_{i_0}(\theta) d\theta & \text{für } m \geq i_0 \geq 1 \end{cases} \\
&= \begin{cases} t & \text{für } i_0 = 0 \\ \int_0^t u_{i_0}(\lambda) d\lambda & \text{für } m \geq i_0 \geq 1 \end{cases} \quad (2.22)
\end{aligned}$$

Analog wird als zweites Beispiel das allgemeine iterierte Integral für einen Multiindex der Länge zwei behandelt:

$$\begin{aligned}
\int_0^t d\xi_{i_1} d\xi_{i_0} &\stackrel{(2.21)}{=} \int_0^t d\xi_{i_1}(\lambda) \int_0^\lambda d\xi_{i_0} \\
&\stackrel{(2.22)}{=} \begin{cases} \int_0^t \lambda d\lambda = \frac{1}{2}t^2 & i_0 = i_1 = 0 \\ \int_0^t \int_0^\lambda u_{i_0}(\theta) d\theta d\lambda & i_1 = 0 \text{ und } m \geq i_0 \geq 1 \\ \int_0^t u_{i_1}(\lambda) \cdot \lambda d\lambda & i_0 = 0 \text{ und } m \geq i_1 \geq 1 \\ \int_0^t u_{i_1}(\lambda) \int_0^\lambda u_{i_0}(\theta) d\theta d\lambda & m \geq i_0, i_1 \geq 1 \end{cases} \quad (2.23)
\end{aligned}$$

Als drittes und letztes Beispiel soll ein iteriertes Integral mit Multiindex der Länge drei dienen, bei dem für  $i_0, i_1$  und  $i_2$  konkrete Werte eingesetzt worden sind, es ergibt sich

$$\begin{aligned}
\int_0^t d\xi_1 d\xi_0 d\xi_2 &= \int_0^t d\xi_1(\lambda) \int_0^\lambda d\xi_0(\theta) \int_0^\theta d\xi_2 \\
&\stackrel{(2.23)}{=} \int_0^t d\xi_1(\lambda) \int_0^\lambda \int_0^\theta u_2(\mu) d\mu d\theta \\
&= \int_0^t u_1(\lambda) \int_0^\lambda \int_0^\theta u_2(\mu) d\mu d\theta d\lambda. \quad (2.24)
\end{aligned}$$

Gegeben sei eine formale Potenzreihe in  $(m+1)$ -nichtkommutativen Unbekannten. Es ist möglich dieser Reihe ein Funktional von  $u_1, \dots, u_m$  zuzuordnen, indem man die Summe aller Produkte der Form

$$c(i_k \dots i_0) \int_0^t d\xi_{i_k} \dots d\xi_{i_0}$$

über  $I^*$  bildet. Die Konvergenz dieser Summe wird durch eine Wachstumbedingung an die „Koeffizienten“  $c(i_k \dots i_0)$  sichergestellt.

**Lemma 2.20** *Angenommen es existieren reelle Zahlen  $K, M > 0$ :*

$$|c(i_k \dots i_0)| < K(k+1)!M^{k+1} \quad (2.25)$$

für alle  $k \geq 0$  und alle Multiindices  $(i_k \dots i_0)$ . Dann existiert eine reelle Zahl  $T > 0$ , so dass für alle  $0 \leq t \leq T$  und für jede Menge stückweiser stetiger, auf  $[0, T]$  definierter Funktionen  $u_1, \dots, u_m$ , die der Bedingung

$$\max_{0 \leq \lambda \leq T} |u_i(\lambda)| \leq 1 \quad (2.26)$$

genügen, die Reihe

$$y(t) = c(\emptyset) + \sum_{k=0}^{\infty} \sum_{i_0=0, \dots, i_k=0}^m c(i_k \dots i_0) \int_0^t d\xi_{i_k} \dots d\xi_{i_0} \quad (2.27)$$

absolut und gleichmäßig konvergiert.

*Beweis:* Wenn  $u_1, \dots, u_m$  die Bedingung (2.26) erfüllen und man zusätzlich  $u_0(t) = 1$  für alle  $t \geq 0$  definiert, folgt aus der Definition des iterierten Integrals

$$\begin{aligned} \int_0^t d\xi_{i_k} \dots d\xi_{i_0} &\stackrel{(2.21)}{=} \int_0^t d\xi_{i_k}(t_k) \int_0^{t_k} d\xi_{i_{k-1}}(t_{k-1}) \int_0^{t_{k-1}} \dots \int_0^{t_3} d\xi_{i_2}(t_2) \int_0^{t_2} d\xi_{i_1}(t_1) \int_0^{t_1} d\xi_{i_0} \\ &\stackrel{(2.20)}{=} \int_0^t u_{i_k}(t_k) \int_0^{t_k} u_{i_{k-1}}(t_{k-1}) \dots \int_0^{t_3} u_{i_2}(t_2) \int_0^{t_2} u_{i_1}(t_1) \underbrace{\int_0^{t_1} u_{i_0}(t_0) dt_0}_{\leq t_1 \text{ nach (2.26)}} dt_1 dt_2 \dots dt_{k-1} dt_k \\ &\quad \underbrace{\leq \frac{1}{2} t_2^2 \text{ nach (2.26)}}_{\leq \frac{1}{3!} t_3^3 \text{ nach (2.26)}} \\ &\quad \underbrace{\leq \frac{1}{k!} t_k^k \text{ nach (2.26)}} \\ &\leq \int_0^t u_{i_k}(t_k) \frac{1}{k!} t_k^k dt_k \\ &= \begin{cases} \int_0^t \frac{1}{k!} t_k^k dt_k = \frac{t^{k+1}}{(k+1)!} & \text{für } i_k = 0 \\ \int_0^t u_{i_k}(t_k) \frac{1}{k!} t_k^k dt_k \stackrel{(2.26)}{\leq} \frac{1}{(k+1)!} t^{k+1} & \text{für } i_k > 0 \end{cases} \\ &\leq \frac{t^{k+1}}{(k+1)!}. \end{aligned}$$

Falls die Wachstumschranke (2.25) erfüllt ist, gilt

$$\begin{aligned} \left| \sum_{i_0=0, \dots, i_k=0}^m c(i_k \dots i_0) \int_0^t d\xi_{i_k} \dots d\xi_{i_0} \right| &\leq \sum_{i_0=0, \dots, i_k=0}^m \underbrace{|c(i_k \dots i_0)|}_{\substack{2.25 \\ \leq K(k+1)!M^{k+1}}} \underbrace{\left| \int_0^t d\xi_{i_k} \dots d\xi_{i_0} \right|}_{\leq \frac{t^{k+1}}{(k+1)!}} \\ &\leq \sum_{i_0=0, \dots, i_k=0}^m K(tM)^{k+1}. \end{aligned}$$

Es gibt insgesamt  $(m+1)^{k+1}$  Summanden, da jedes  $i_j$   $m+1$  Werte annehmen kann. Also gilt

$$|\dots| \leq K[M(m+1)t]^{k+1}.$$

Daraus folgt für hinreichend kleines  $T$ , dass die Reihe (2.27) absolut und gleichmäßig auf  $[0, T]$  konvergiert.  $\square$

Anzumerken ist noch, dass wir abweichend von [12] in (2.26) nicht die strikte Ungleichung gefordert haben. Dies ist allerdings auch für den Beweis nicht notwendig. Der Ausdruck (2.27) definiert ein Funktional von  $u_1, \dots, u_m$ . Eine Repräsentation der Form (2.27) ist, wie das folgende Lemma zeigt, eindeutig.

**Lemma 2.21** *Seien  $c^a$  und  $c^b$  zwei formale Potenzreihen in  $(m+1)$ -nichtkommutativen Unbekannten und seien die korrespondierenden Funktionale von der Form (2.27) auf dem gleichen Intervall  $[0, T]$  definiert. Dann stimmen die Funktionale genau dann überein, falls*

$$c^a = c^b$$

*gilt.*

*Beweis:* Dieses Lemma beweist man, indem man den formalen Potenzreihen  $c^a$  und  $c^b$  Funktionale der Form (2.27) zuordnet und feststellt, dass auch  $y(t) = y^a(t) - y^b(t)$  ein Funktional des Typs (2.27) ist, dessen Koeffizienten als Differenz von  $c^a$  und  $c^b$  definiert sind. Um das Lemma zu beweisen, muss man also zeigen, dass alle diese Koeffizienten gleich Null sind. Für  $c(\emptyset), c(0), c(00), \dots$  folgt das, wenn man  $u_1 = \dots = u_m = 0$  setzt und  $t$  auf dem Intervall  $t \in [0, T]$  variiert. Für die restlichen Multiindices der Länge eins, erhält man die gewünschte Aussage über die erste Ableitung von (2.27). Die Behauptung folgt schließlich unter zusätzlicher Beachtung der höheren Ableitungen. Für einen ausführlichen Beweis sei auf [12] und die dort aufgeführte Literatur verwiesen.  $\square$

Als nächstes zeigen wir, dass die Output-Funktion  $y(t)$  des nichtlinearen Systems (2.19) als Funktional der Inputs  $u_1, \dots, u_m$  in der Form (2.27) geschrieben werden kann. Für dieses Ziel benötigen wir einige vorbereitende Resultate.

**Lemma 2.22** Seien  $g_0, \dots, g_m$  analytische Vektorfelder und  $\lambda$  eine auf  $U$  definierte reellwertige Funktion. Gegeben ein Punkt  $x^0 \in U$ , betrachte die formale Potenzreihe definiert durch

$$\begin{aligned} c(\emptyset) &= \lambda(x^0) \\ c(i_k \dots i_0) &= L_{g_{i_0}} \dots L_{g_{i_k}} \lambda(x^0). \end{aligned} \quad (2.28)$$

Dann existieren reelle Zahlen  $K, M > 0$ , so dass die Wachstumsbeschränkung (2.25) erfüllt ist.

Für einen Beweis sei auf [12] und die dort aufgeführte Literatur verwiesen. In Anbetracht des Resultats von Lemma 2.20 kann man mit  $g_0, \dots, g_m$  und  $\lambda$  das Funktional

$$v(t) = \lambda(x^0) + \sum_{k=0}^{\infty} \sum_{i_0=0, \dots, i_k=0}^m L_{g_{i_0}} \dots L_{g_{i_k}} \lambda(x^0) \int_0^t d\xi_{i_k} \dots d\xi_{i_0} \quad (2.29)$$

definieren.

**Lemma 2.23** Seien  $g_0, \dots, g_m$  wie in Lemma 2.22 definiert und  $\lambda_1, \dots, \lambda_l$  reellwertige analytische Funktionen, definiert auf  $U$ . Desweiteren sei  $\gamma$  eine reellwertige analytische Funktion definiert auf dem  $\mathbb{R}^l$ . Bezeichnen  $v_1(t), \dots, v_l(t)$  Funktionale analog (2.29) mit  $\lambda = \lambda_1, \dots, \lambda = \lambda_l$ . Dann ist die Komposition  $\gamma(v_1(t), \dots, v_l(t))$  wieder ein Funktional der Form (2.29), zugehörig zu  $\lambda = \gamma(\lambda_1, \dots, \lambda_l)$ .

*Beweis:* Der Beweis erfolgt in mehreren Schritten. Zuerst zeigt man, dass auch  $\alpha_1 v_1(t) + \alpha_2 v_2(t)$  ein Funktional des Typs (2.29) ist. Anschließend zeigt man selbiges für das Produkt  $v_1(t)v_2(t)$  zweier Funktionale. Falls dann  $\gamma$  eine beliebige reellwertige analytische Funktion auf dem  $\mathbb{R}^l$  ist, nehme man die Taylorentwicklung am Ursprung und nutze rekursiv die vorherigen Resultate, um zu zeigen, dass die Komposition  $\gamma(v_1(t), \dots, v_l(t))$  durch eine Reihe der Form (2.29) repräsentiert werden kann, wobei  $\lambda$  durch die Taylorentwicklung von  $\gamma(\lambda_1, \dots, \lambda_l)$  ersetzt wird. Für Details sei erneut auf [12] verwiesen.  $\square$

**Theorem 2.24** Erfüllen die Inputfunktionen  $u_1, \dots, u_m$  des Kontrollsystems (2.19) die Bedingung (2.26) und sei  $T$  hinreichend klein, dann gilt für alle  $0 \leq t \leq T$ , dass die  $j$ -te Outputfunktion  $y_j(t)$  des Systems (2.19) als Reihe

$$y_j(t) = h_j(x^0) + \sum_{k=0}^{\infty} \sum_{i_0=0, \dots, i_k=0}^m L_{g_{i_0}} \dots L_{g_{i_k}} h_j(x^0) \int_0^t d\xi_{i_k} \dots d\xi_{i_0}, \quad (2.30)$$

mit  $g_0 = f$  entwickelt werden kann.



*Beweis:* Zuerst zeigt man, dass die  $j$ -te Komponente der Lösung der Differentialgleichung (2.19) durch

$$x_j(t) = x_j(x^0) + \sum_{k=0}^{\infty} \sum_{i_0=0, \dots, i_k=0}^m L_{g_{i_0}} \dots L_{g_{i_k}} x_j(x^0) \int_0^t d\xi_{i_k} \dots d\xi_{i_0} \quad (2.31)$$

ausgedrückt werden kann, wobei  $x_j(x)$  die Projektion des Vektors  $x$  auf die  $j$ -te Komponente bezeichnet. Beachte dabei, dass aus der Definition des iterierten Integrals

$$\frac{d}{dt} \int_0^t d\xi_{i_k} d\xi_{i_{k-1}} \dots d\xi_{i_0} = \begin{cases} 1 \cdot \int_0^t d\xi_{i_{k-1}} \dots d\xi_{i_0} & \text{für } i_k = 0 \\ u_{i_k}(t) \int_0^t d\xi_{i_{k-1}} \dots d\xi_{i_0} & \text{für } i_k > 0 \end{cases}$$

folgt. Nimmt man nun die Ableitung der rechten Seite von (2.31) bzgl. der Zeit und ordnet die Terme um, erhält man

$$\begin{aligned} \dot{x}_j(t) &= L_f x_j(x^0) + \sum_{k=0}^{\infty} \sum_{i_0=0, \dots, i_k=0}^m L_{g_{i_0}} \dots L_{g_{i_k}} L_f x_j(x^0) \int_0^t d\xi_{i_k} \dots d\xi_{i_0} \\ &+ \sum_{i=1}^m \left[ L_{g_i} x_j(x^0) + \sum_{k=0}^{\infty} \sum_{i_0=0, \dots, i_k=0}^m L_{g_{i_0}} \dots L_{g_{i_k}} L_{g_i} x_j(x^0) \int_0^t d\xi_{i_k} \dots d\xi_{i_0} \right] u_i(t). \end{aligned}$$

Bezeichnen  $f_j$  und  $g_{ij}$  die  $j$ -te Komponente von  $f$  und  $g_i$ , dann gilt  $L_f x_j = f_j(x_1, \dots, x_n)$  und man kann mit Hilfe von Lemma 2.23

$$\begin{aligned} &L_f x_j(x^0) + \sum_{k=0}^{\infty} \sum_{i_0=0, \dots, i_k=0}^m L_{g_{i_0}} \dots L_{g_{i_k}} L_f x_j(x^0) \int_0^t d\xi_{i_k} \dots d\xi_{i_0} \\ &= f_j(x^0) + \sum_{k=0}^{\infty} \sum_{i_0=0, \dots, i_k=0}^m L_{g_{i_0}} \dots L_{g_{i_k}} f_j(x^0) \int_0^t d\xi_{i_k} \dots d\xi_{i_0} \\ &= f_j(x_1(t), \dots, x_n(t)) \end{aligned}$$

schreiben. Analog kann man für die anderen Terme verfahren und erhält schließlich

$$\dot{x}_j(t) = f_j(x_1(t), \dots, x_n(t)) + \sum_{i=1}^m g_{ij}(x_1(t), \dots, x_n(t)) u_i(t).$$

Desweiteren genügen die  $x_j(t)$  der Bedingung  $x_j(0) = x_j^0$  und sind deshalb die Komponenten der Lösung  $x(t)$  der Differentialgleichung in (2.19). Eine weitere Anwendung von Lemma 2.23 zeigt, dass der Output von (2.19) durch (2.30) ausgedrückt werden kann.  $\square$

## 2.4 Sampled-data Systeme und Problemstellung

Der sampled-data Ansatz ist ein Konzept für eine unstetige Feedback-Steuerung. Zuerst muss man die Begriffe Sampling-Folge, sampled-data Zeit und sampled-data Lösung definieren.

**Definition 2.25** *Eine Folge*

$$t = (t_i)_{i \in \mathbb{N}_0}, \quad 0 = t_0 < t_1 < t_2 < \dots < t_i < \dots \subset \mathbb{R}$$

heißt *Sampling-Folge beziehungsweise Abtastfolge, falls gilt*

- $t_i \rightarrow \infty$  für  $i \rightarrow \infty$  und
- $\Delta_t := \sup_{i \in \mathbb{N}_0} (t_{i+1} - t_i) < \infty$ .

Die Zahl  $\Delta_t$  wird *sampled-data Zeit* genannt.

Im weiteren Verlauf dieser Arbeit gehen wir von einer konstanten Schrittweite aus. Diese Schrittweite  $T$  wird auch als Samplingperiode bezeichnet, d.h. die Sampling-Folge ist  $(kT)_{k \in \mathbb{N}_0}$ . Zusätzlich kann man noch die sampled-data Rate als  $1/T$  definieren.

Für den Rest der Arbeit wird darüberhinaus folgende Annahme benötigt.

**Annahme 2.26** *Es sei ein statisches Zustandsfeedback  $u_0 : \mathbb{R}^n \rightarrow \mathbb{R}$  gegeben, das eine gewisse Kontrollaufgabe für das zeitkontinuierliche System (2.18) erfüllt. Zusätzlich wird angenommen, dass alle verwendeten Funktionen hinreichend glatt sind, so dass die Ableitungen im Folgenden stets wohldefiniert und stetig sind.*

Hierfür ist es ausreichend zu fordern, dass die Funktionen hinreichend glatt sind, weil wir im Gegensatz zu Abschnitt 2.3 nur an Approximationen endlicher Ordnung interessiert sind. Unser Ziel ist es eine Feedback-Steuerung  $u_T(x)$  zu finden, so dass die Lösungstrajektorien des sampled-data closed-loop Systems

$$\dot{x}(t) = f(x(t), u_T(x(t_k))), \quad t \in [t_k, t_{k+1}), k = 0, 1, \dots \quad (2.32)$$

für die Sampling-Folge  $t_k = kT$  und Samplingperiode  $T > 0$  die Lösungskurve des kontinuierlichen Systems (2.18) möglichst gut approximieren. Dabei bezeichnet  $\Phi_T(t, x_0, u_T)$  die Lösung von (2.32). Sicherlich ist der Begriff „möglichst gut“ noch etwas unpräzise. Genauer ist eine sampled-data Feedback-Steuerung  $u_T$  gesucht, so dass die Differenz nach einem Samplingintervall

$$\Delta\Phi(T, x_0, u_T) := \|\Phi(T, x_0) - \Phi_T(T, x_0, u_T)\|_\infty \quad (2.33)$$

möglichst klein wird. Dazu ist es notwendig zu definieren, welche Kontrollfunktionen für die Lösung dieses Problems zulässig sein sollen. Antwort darauf gibt die folgende Definition.

**Definition 2.27** Eine zulässige sampled-data Feedback-Regelung  $u_T$  ist eine Familie von Abbildungen  $u_T : \mathbb{R}^n \rightarrow \mathbb{R}$ , parametrisiert durch die Samplingperiode  $T \in (0, T^*]$  für eine maximale Samplingperiode  $T^*$ , so dass für jede kompakte Menge  $K \subset \mathbb{R}^n$  die Ungleichung

$$\sup_{x \in K, T \in (0, T^*]} |u_T(x)| < \infty$$

gilt.

Zu beachten ist hierbei, dass für die Existenz und Eindeutigkeit der Lösungen von (2.32) keinerlei Stetigkeitsannahmen an  $u_T$  benötigt werden. Beschränktheit ist dagegen sowohl aus theoretischer als auch praktischer Sicht sinnvoll, weil ansonsten die Lösungen sehr sensitiv gegenüber Modellierungs- oder Approximationsfehlern bzw. unmöglich zu implementieren sind. Eine spezielle Klasse dieser Steuerungen ist aus [21] bekannt:

$$u_T(x) = \sum_{i=0}^M T^i u_i(x), \quad (2.34)$$

wobei  $u_0$  aus (2.18) ist und  $u_1, \dots, u_M : \mathbb{R}^n \rightarrow \mathbb{R}$  lokal beschränkte Funktionen sind. Unser Ziel in dieser Arbeit sind asymptotische Abschätzungen für  $T \rightarrow 0$ . Entsprechend benötigen wir die folgende Definition, um die Güte unserer Approximationen besser abschätzen zu können.

**Definition 2.28** Sei  $k \in \mathbb{N}$ . Wir schreiben

$$\Delta\Phi(T, x_0, u_T) = \mathcal{O}(T^k) \quad \text{auf } K$$

für eine kompakte Menge  $K$ , falls eine Konstante  $C > 0$  existiert, so dass die Ungleichung

$$\Delta\Phi(T, x_0, u_T) \leq CT^k \quad \forall x_0 \in K$$

erfüllt ist und

$$\Delta\Phi(T, x_0, u_T) = \mathcal{O}(T^k),$$

falls dieser Ausdruck für jede kompakte Teilmenge des  $\mathbb{R}^n$  gültig ist, wobei die Konstante  $C > 0$  jeweils in Abhängigkeit von  $K$  gewählt werden kann.

Falls man  $\Delta\Phi(T, x_0, u_T) = \mathcal{O}(T^k)$  zeigen kann, folgt durch ein einfaches Induktionsargument für jedes Intervall der Form  $[0, t^*]$

$$\|\Phi(t, x_0) - \Phi_T(t, x_0, u_T)\|_\infty \leq \mathcal{O}(T^{k-1}) \quad (2.35)$$

für alle Zeitpunkte  $t = iT, i \in \mathbb{N}_0$  mit  $t \in [0, t^*]$ . Mit Hilfe von (2.35) kann man zeigen, dass sich einige Stabilitätskonzepte von  $\Phi$  auf  $\Phi_h$  in einem semiglobalen praktischen Sinne

übertragen lassen. Für nähere Details siehe [22] und [26].

Um Abschätzungen für (2.33) zu gewinnen, betrachte man eine hinreichend glatte, reellwertige Funktion  $h : \mathbb{R}^n \rightarrow \mathbb{R}$  und leite Abschätzungen für

$$\Delta h(T, x_0, u_T) := | h(\Phi(T, x_0)) - h(\Phi_T(T, x_0, u_T)) | \quad (2.36)$$

her.

**Bemerkung 2.29** *Die Funktion  $h$  kann als eine Hilfsfunktion oder als eine Outputfunktion betrachtet werden. Wendet man beispielsweise die für (2.36) erhaltenen Resultate auf  $h_j(x) := x_j$ ,  $j = 1, \dots, n$ , an, so entspricht das Maximum über diese Ausdrücke der Form (2.36) genau (2.33).*

Abgesehen davon wird die Frage behandelt, ob andere Feedback-Steuerung als (2.34) bessere Resultate liefern können.

# Kapitel 3

## Fliess- und Taylorentwicklung des closed-loop Systems

In diesem Kapitel wird zuerst ein Theorem aus [21] hergeleitet, dass die Differenz  $\frac{h(\Phi_T(T, x, u_T)) - h(x)}{T}$  mit Hilfe der in Abschnitt 2.3 eingeführten Fliessentwicklung ausdrückt. Anschließend wird dieses Theorem genutzt, um Abschätzungen verschiedener Ordnungen für  $h(\Phi_T(T, x, u_T))$  explizit darstellen zu können. Analog wird mittels der Taylorentwicklung für das kontinuierliche System  $h(\Phi(T, x))$  verfahren. Der letzte Abschnitt vergleicht die Fliessreihenentwicklung für das sampled-data System mit der Taylorentwicklung des kontinuierlichen Systems.

### 3.1 Fliessentwicklung eines sampled-data Systems

In Abschnitt 2.4 haben wir das Ziel formuliert, über (2.36) Aussagen bezüglich  $\Delta\Phi(T, x, u_T)$  herzuleiten. Allerdings beinhaltet (2.36) den Term  $h(\Phi_T(T, x, u_T))$  mit

$$u_T(x) = \sum_{i=0}^M T^i u_i(x).$$

Da es im Allgemeinen nicht möglich ist,  $h(\Phi_T(T, x, u_T))$  exakt zu berechnen, wird im Folgenden eine Approximationstechnik eingesetzt. Diese führt zu einem aus [21] und [11] bekannten Theorem.

**Theorem 3.1** *Betrachte das System (2.16), eine hinreichend glatte Funktion  $h : \mathbb{R}^n \rightarrow \mathbb{R}$ , das zeitkontinuierliche closed-loop System (2.18) und das sampled-data System (2.32) mit Regler (2.34). Dann gilt für hinreichend kleine Samplingperiode  $T$*

$$\frac{h(\Phi_T(T, x, u_T)) - h(x)}{T} = \sum_{s=0}^M T^s [L_{g_1} h \cdot u_s + p_s(x, u_0, \dots, u_{s-1})] + \mathcal{O}(T^{M+1}), \quad (3.1)$$

wobei  $p_0(x) = L_{g_0}h(x)$ ,

$$p_s(x, u_0, \dots, u_{s-1}) = \sum_{k=1}^s \sum_{i_0=0, \dots, i_k=0}^1 \frac{L_{g_{i_0}} \cdots L_{g_{i_k}} h(x)}{(k+1)!} \left( \sum_{\substack{|\nu|=|I_k| \\ \|\nu\|=s-k}} \binom{|I_k|}{n_0 n_1 \dots n_M} \prod_{j=0}^{s-1} u_j^{n_j} \right) \quad (3.2)$$

für  $s \geq 1$  und  $I_k$  den Multiindex  $I_k = (i_0, i_1, \dots, i_k)$  bezeichnet.

Beweis: Betrachte die Lösung von (2.16) initialisiert mit  $x(0) = x$ , Input  $u(\cdot)$  und Output  $y(t) = h(x(t))$ . Dann gilt für hinreichend kleine  $t$  mit Theorem 2.24

$$h(x(t)) - h(x) = \sum_{k=0}^{\infty} \sum_{i_0=0, \dots, i_k=0}^{m=1} L_{g_{i_0}} \cdots L_{g_{i_k}} h(x) \int_0^t d\xi_{i_k} \cdots d\xi_{i_0}, \quad (3.3)$$

wobei  $\int_0^t d\xi_{i_k} \cdots d\xi_{i_0}$  das in Abschnitt 2.3 eingeführte iterierte Integral ist. Zu beachten ist, dass  $m = 1$  gilt, weil eindimensionale Inputsysteme betrachtet werden. Folglich nehmen die Indices  $i_j$ ,  $j = 0, \dots, k$ , Werte aus der Menge  $\{0, 1\}$  an. Falls  $t = T$  hinreichend klein ist und  $u(\cdot) = u = \text{konstant}$ , kann man (3.3) schreiben als

$$\frac{h(\Phi_T(T, x, u_T)) - h(x)}{T} = \sum_{k=0}^{\infty} \sum_{i_0=0, \dots, i_k=0}^{m=1} L_{g_{i_0}} \cdots L_{g_{i_k}} h(x) \frac{T^k}{(k+1)!} u^{|I_k|}, \quad (3.4)$$

wobei  $|I_k| := i_0 + i_1 + \dots + i_k$ . Benutzt man nun Lemma 2.3 für  $u$  aus (2.34) erhält man

$$u^{|I_k|} = \sum_{|\nu|=|I_k|} \binom{|I_k|}{n_0 n_1 \dots n_M} u_0^{n_0} \cdots u_M^{n_M} T^{\|\nu\|}, \quad (3.5)$$

wobei die Notation aus (2.1) und (2.2) benutzt worden ist. Nun können wir (3.4) mit Hilfe von (3.5) umschreiben zu

$$\begin{aligned} \frac{h(\Phi_T(T, x, u_T)) - h(x)}{T} &= \sum_{k=0}^M \sum_{i_0=0, \dots, i_k=0}^{m=1} L_{g_{i_0}} \cdots L_{g_{i_k}} h(x) \frac{T^k}{(k+1)!} \\ &\cdot \left( \sum_{|\nu|=|I_k|} \binom{|I_k|}{n_0 n_1 \dots n_M} \prod_{j=0}^M u_j^{n_j} \cdot T^{\|\nu\|} \right) + H_1, \quad (3.6) \end{aligned}$$

wobei  $H_1 = \mathcal{O}(T^{M+1})$ . Führt man nun einen neuen Index  $s := k + \sum_{j=0}^M j n_j$  ein, sortiert alle Summanden in (3.6) nach ihrer Ordnung in  $T$  und fasst alle Summanden der Ordnung

$M + 1$  und größer mit  $H_1$  in  $\mathcal{O}(T^{M+1})$  zusammen, erhält man

$$\begin{aligned} \frac{h(\Phi_T(T, x, u_T)) - h(x)}{T} &= \sum_{s=0}^M T^s \sum_{k=0}^s \sum_{i_0=0, \dots, i_k=0}^{m=1} \frac{L_{g_{i_0}} \cdots L_{g_{i_k}} h(x)}{(k+1)!} \\ &\cdot \left( \sum_{\substack{|\nu|=|I_k| \\ \|\nu\|=s-k}} \binom{|I_k|}{n_0 n_1 \dots n_M} \prod_{j=0}^M u_j^{n_j} \right) + \mathcal{O}(T^{M+1}). \end{aligned} \quad (3.7)$$

Aus  $s = 0$  folgt  $k = 0$ , also wird für die Terme 0. Ordnung nur  $i_0$  benötigt. Dieser Index hat den Wertebereich  $\{0, 1\}$  und es ergibt sich der Term

$$L_{g_1} h(x) \cdot u_0 + \underbrace{L_{g_0} h(x)}_{=p_0(x)}. \quad (3.8)$$

Für beliebiges  $s = 1, \dots, M$  und  $k = 0$  ergibt sich jeweils der Term  $L_{g_1} h(x) \cdot u_s$  wegen der kombinatorischen Bedingung  $|\nu| = |I_k|$  und  $\|\nu\| = s - k$ . Setzt man nun die Definition (3.2) von  $p_s$  in (3.7) ein, erhält man die Behauptung. Zu beachten ist dabei noch, dass  $p_s$  in (3.2) für  $M \geq s - 1$  unabhängig von dem Wert von  $M$  ist, weil die Bedingung  $\|\nu\| = s - k \leq s - 1$  impliziert, dass  $n_s = \dots = n_M = 0$ .  $\square$

## 3.2 Fließentwicklung des zeitdiskreten Systems

Es werden in den nächsten Abschnitten die Ausdrücke

$$\Delta h_{ct}(T, x) = h(\Phi(T, x)) - h(x) \quad \text{und} \quad (3.9)$$

$$\Delta h_{dt}(T, x, u_T) = h(\Phi_T(T, x, u_T)) - h(x) \quad (3.10)$$

verwendet, weil sie besser zu Theorem 3.1 passen und für die in (2.36) definierte Differenz

$$\Delta h(T, x, u_T) = | \Delta h_{ct}(T, x) - \Delta h_{dt}(T, x, u_T) | \quad (3.11)$$

gilt. In diesem Abschnitt wird (3.10) genauer untersucht, d.h. in erster Linie wird (3.10) explizit entwickelt, damit man das Resultat der Reihendarstellung in Abschnitt 3.4 besser mit (3.9) vergleichen kann.

Aus Theorem 3.1 erhalt man die Darstellung

$$\begin{aligned}
 \Delta h_{at}(T, x, u_T) &:= h(\Phi_T(T, x, u_T)) - h(x) \\
 &\stackrel{(3.1)}{=} T[L_{g_1} h u_0 + L_{g_0} h] \\
 &\quad + T^2[L_{g_1} h u_1 + p_1(x, u_0)] \\
 &\quad + T^3[L_{g_1} h u_2 + p_2(x, u_0, u_1)] \\
 &\quad + T^4[L_{g_1} h u_3 + p_3(x, u_0, u_1, u_2)] \\
 &\quad + T^5[L_{g_1} h u_4 + p_4(x, u_0, u_1, u_2, u_3)] \\
 &\quad + T^6[L_{g_1} h u_5 + p_5(x, u_0, u_1, u_2, u_3, u_4)] + \mathcal{O}(T^7). \tag{3.12}
 \end{aligned}$$

Dabei tauchen die Ausdrucke  $p_1, \dots, p_5$  auf, die nun ausgerechnet werden:

$$\begin{aligned}
 p_1 &\stackrel{(3.2)}{=} \sum_{i_0=0, i_1=0}^1 \frac{1}{2!} L_{g_{i_0}} L_{g_{i_1}} h(x) \left( \sum_{\substack{|\nu|=|I_1| \\ \|\nu\|=0}} \binom{|I_1|}{n_0 n_1 \dots n_M} \prod_{j=0}^0 u_j^{n_j} \right) \\
 &= \frac{1}{2} \left[ L_{g_0} L_{g_0} h(x) + [L_{g_0} L_{g_1} h(x) + L_{g_1} L_{g_0} h(x)] u_0 + L_{g_1} L_{g_1} h(x) u_0^2 \right], \tag{3.13}
 \end{aligned}$$

$$\begin{aligned}
 p_2 &\stackrel{(3.2)}{=} \sum_{k=1}^2 \sum_{i_0=0, \dots, i_k=0}^1 \frac{1}{(k+1)!} L_{g_{i_0}} \dots L_{g_{i_k}} h(x) \left( \sum_{\substack{|\nu|=|I_k| \\ \|\nu\|=2-k}} \binom{|I_k|}{n_0 n_1 \dots n_M} \prod_{j=0}^1 u_j^{n_j} \right) \\
 &= \frac{1}{2} \left[ [L_{g_0} L_{g_1} h(x) + L_{g_1} L_{g_0} h(x)] u_1 + 2L_{g_1} L_{g_1} h(x) u_0 u_1 \right] \\
 &\quad + \frac{1}{6} \left[ L_{g_0}^3 h(x) + [L_{g_0} L_{g_0} L_{g_1} h(x) + L_{g_0} L_{g_1} L_{g_0} h(x) + L_{g_1} L_{g_0} L_{g_0} h(x)] u_0 \right. \\
 &\quad \left. + [L_{g_0} L_{g_1} L_{g_1} h(x) + L_{g_1} L_{g_0} L_{g_1} h(x) + L_{g_1} L_{g_1} L_{g_0} h(x)] u_0^2 + L_{g_1}^3 h(x) u_0^3 \right], \tag{3.14}
 \end{aligned}$$



$$\begin{aligned}
p_3 &\stackrel{(3.2)}{=} \sum_{k=1}^3 \sum_{i_0=0, \dots, i_k=0}^1 \frac{1}{(k+1)!} L_{g_{i_0}} \cdots L_{g_{i_k}} h(x) \left( \sum_{\substack{|\nu|=|I_k| \\ \|\nu\|=3-k}} \binom{|I_k|}{n_0 n_1 \dots n_M} \prod_{j=0}^2 u_j^{n_j} \right) \\
&= \frac{1}{2} \left[ [L_{g_0} L_{g_1} h(x) + L_{g_1} L_{g_0} h(x)] u_2 + L_{g_1} L_{g_1} h(x) [u_1^2 + 2u_0 u_2] \right] \\
&+ \frac{1}{6} \left[ [L_{g_0} L_{g_0} L_{g_1} h(x) + L_{g_0} L_{g_1} L_{g_0} h(x) + L_{g_1} L_{g_0} L_{g_0} h(x)] u_1 \right. \\
&+ \left. [L_{g_0} L_{g_1} L_{g_1} h(x) + L_{g_1} L_{g_0} L_{g_1} h(x) + L_{g_1} L_{g_1} L_{g_0} h(x)] 2u_0 u_1 + 3L_{g_1} L_{g_1} L_{g_1} h(x) u_0^2 u_1 \right] \\
&+ \frac{1}{24} \left[ L_{g_0} L_{g_0} L_{g_0} L_{g_0} h \right. \\
&+ \left. [L_{g_0} L_{g_0} L_{g_0} L_{g_1} h + L_{g_0} L_{g_0} L_{g_1} L_{g_0} h + L_{g_0} L_{g_1} L_{g_0} L_{g_0} h + L_{g_1} L_{g_0} L_{g_0} L_{g_0} h] u_0 \right. \\
&+ \left. [L_{g_0} L_{g_0} L_{g_1} L_{g_1} h + L_{g_0} L_{g_1} L_{g_0} L_{g_1} h + L_{g_1} L_{g_0} L_{g_0} L_{g_1} h \right. \\
&+ \left. L_{g_0} L_{g_1} L_{g_1} L_{g_0} h + L_{g_1} L_{g_0} L_{g_1} L_{g_0} h + L_{g_1} L_{g_1} L_{g_0} L_{g_0} h] u_0^2 \right. \\
&+ \left. [L_{g_1} L_{g_1} L_{g_1} L_{g_0} h + L_{g_1} L_{g_1} L_{g_0} L_{g_1} h + L_{g_1} L_{g_0} L_{g_1} L_{g_1} h + L_{g_0} L_{g_1} L_{g_1} L_{g_1} h] u_0^3 \right. \\
&+ \left. L_{g_1} L_{g_1} L_{g_1} L_{g_1} h u_0^4 \right], \tag{3.15}
\end{aligned}$$

$$\begin{aligned}
p_4 &\stackrel{(3.2)}{=} \sum_{k=1}^4 \sum_{i_0=0, \dots, i_k=0}^1 \frac{1}{(k+1)!} L_{g_{i_0}} \cdots L_{g_{i_k}} h(x) \left( \sum_{\substack{|\nu|=|I_k| \\ \|\nu\|=4-k}} \binom{|I_k|}{n_0 n_1 \dots n_M} \prod_{j=0}^3 u_j^{n_j} \right) \\
&= \frac{1}{2} \left[ [L_{g_0} L_{g_1} h(x) + L_{g_1} L_{g_0} h(x)] u_3 + L_{g_1} L_{g_1} h(x) 2[u_0 u_3 + u_1 u_2] \right] \\
&+ \frac{1}{6} \left[ [L_{g_0} L_{g_0} L_{g_1} h(x) + L_{g_0} L_{g_1} L_{g_0} h(x) + L_{g_1} L_{g_0} L_{g_0} h(x)] u_2 \right. \\
&+ \left. [L_{g_0} L_{g_1} L_{g_1} h(x) + L_{g_1} L_{g_0} L_{g_1} h(x) + L_{g_1} L_{g_1} L_{g_0} h(x)] (2u_0 u_2 + u_1^2) \right. \\
&+ \left. 3L_{g_1} L_{g_1} L_{g_1} h(x) (u_0^2 u_2 + u_1^2 u_0) \right] \\
&+ \frac{1}{24} \left[ [L_{g_0} L_{g_0} L_{g_0} L_{g_1} h + L_{g_0} L_{g_0} L_{g_1} L_{g_0} h + L_{g_0} L_{g_1} L_{g_0} L_{g_0} h + L_{g_1} L_{g_0} L_{g_0} L_{g_0} h] u_1 \right. \\
&+ \left. [L_{g_0} L_{g_0} L_{g_1} L_{g_1} h + L_{g_0} L_{g_1} L_{g_0} L_{g_1} h + L_{g_1} L_{g_0} L_{g_0} L_{g_1} h \right. \\
&+ \left. L_{g_0} L_{g_1} L_{g_1} L_{g_0} h + L_{g_1} L_{g_0} L_{g_1} L_{g_0} h + L_{g_1} L_{g_1} L_{g_0} L_{g_0} h] 2u_0 u_1 \right. \\
&+ \left. [L_{g_1} L_{g_1} L_{g_1} L_{g_0} h + L_{g_1} L_{g_1} L_{g_0} L_{g_1} h + L_{g_1} L_{g_0} L_{g_1} L_{g_1} h + L_{g_0} L_{g_1} L_{g_1} L_{g_1} h] 3u_0^2 u_1 \right. \\
&+ \left. L_{g_1} L_{g_1} L_{g_1} L_{g_1} h 4u_0^3 u_1 \right] + \frac{1}{5!} R_4 \tag{3.16}
\end{aligned}$$

und

$$\begin{aligned}
 p_5 &\stackrel{(3.2)}{=} \sum_{k=1}^5 \sum_{i_0=0, \dots, i_k=0}^1 \frac{1}{(k+1)!} L_{g_{i_0}} \cdots L_{g_{i_k}} h(x) \left( \sum_{\substack{|\nu|=|I_k| \\ \|\nu\|=5-k}} \binom{|I_k|}{n_0 n_1 \dots n_M} \prod_{j=0}^4 u_j^{n_j} \right) \\
 &= \frac{1}{2} \left[ [L_{g_0} L_{g_1} h + L_{g_1} L_{g_0} h] u_4 + L_{g_1} L_{g_1} h [2u_3 u_1 + u_2^2 + 2u_0 u_4] \right] \\
 &+ \frac{1}{6} \left[ [L_{g_0} L_{g_0} L_{g_1} + L_{g_0} L_{g_1} L_{g_0} + L_{g_1} L_{g_0} L_{g_0}] h u_3 \right. \\
 &+ [L_{g_1}^2 L_{g_0} + L_{g_1} L_{g_0} L_{g_1} + L_{g_0} L_{g_0}^2] h 2[u_2 u_1 + u_0 u_3] + L_{g_1}^3 h [3u_3 u_0^2 + u_1^3 + 6u_0 u_1 u_2] \left. \right] \\
 &+ \frac{1}{24} \left[ [L_{g_0}^3 L_{g_1} + L_{g_0}^2 L_{g_1} L_{g_0} + L_{g_0} L_{g_1} L_{g_0}^2 + L_{g_1} L_{g_0}^3] h u_2 \right. \\
 &+ [L_{g_0}^2 L_{g_1}^2 + L_{g_0} L_{g_1} L_{g_0} L_{g_1} + L_{g_1} L_{g_0}^2 L_{g_1} + L_{g_0} L_{g_1}^2 L_{g_0} + L_{g_1} L_{g_0} L_{g_1} L_{g_0} + L_{g_0}^2 L_{g_1}^2] h [2u_2 u_0 + u_1^2] \\
 &+ [L_{g_0} L_{g_1}^3 + L_{g_1} L_{g_0} L_{g_1}^2 + L_{g_1}^2 L_{g_0} L_{g_1} + L_{g_0} L_{g_1}^3] h [3u_2 u_0^2 + 3u_1^2 u_0] + L_{g_1}^4 h [4u_2 u_0^3 + 6u_1^2 u_0^2] \left. \right] \\
 &+ \frac{1}{120} \left[ [L_{g_0}^4 L_{g_1} + L_{g_0}^3 L_{g_1} L_{g_0} + L_{g_0}^2 L_{g_1} L_{g_0}^2 + L_{g_0} L_{g_1} L_{g_0}^3 + L_{g_1} L_{g_0}^4] h u_1 \right. \\
 &+ [L_{g_0}^3 L_{g_1}^2 + L_{g_0}^2 L_{g_1} L_{g_0} L_{g_1} + L_{g_0} L_{g_1} L_{g_0}^2 L_{g_1} + L_{g_1} L_{g_0}^3 L_{g_1} + L_{g_0}^2 L_{g_1}^2 L_{g_0} + L_{g_0} L_{g_1} L_{g_0} L_{g_1} L_{g_0} \\
 &+ L_{g_1} L_{g_0}^2 L_{g_1} L_{g_0} + L_{g_0} L_{g_1}^2 L_{g_0}^2 + L_{g_1} L_{g_0} L_{g_1} L_{g_0}^2 + L_{g_0}^2 L_{g_1}^3] h [2u_0 u_1] \\
 &+ [L_{g_1}^3 L_{g_0}^2 + L_{g_1}^2 L_{g_0} L_{g_1} L_{g_0} + L_{g_1} L_{g_0} L_{g_1}^2 L_{g_0} + L_{g_0} L_{g_1}^3 L_{g_0} + L_{g_1}^2 L_{g_0}^2 L_{g_1} + L_{g_1} L_{g_0} L_{g_1} L_{g_0} L_{g_1} \\
 &+ L_{g_0} L_{g_1}^2 L_{g_0} L_{g_1} + L_{g_1} L_{g_0}^2 L_{g_1}^2 + L_{g_0} L_{g_1} L_{g_0} L_{g_1}^2 + L_{g_0}^2 L_{g_1}^3] h [3u_0^2 u_1] \\
 &+ [L_{g_0} L_{g_1}^4 + L_{g_1} L_{g_0} L_{g_1}^3 + L_{g_1}^2 L_{g_0} L_{g_1}^2 + L_{g_1}^3 L_{g_0} L_{g_1} + L_{g_1}^4 L_{g_0}] h 4u_0^3 u_1 \\
 &+ L_{g_1}^5 h 5u_0^4 u_1 \left. \right] + \frac{1}{6!} R_5. \tag{3.17}
 \end{aligned}$$

Hierbei steht  $R_k$  für

$$\sum_{i_0=0, \dots, i_{k-1}=0}^1 L_{g_{i_0}} \cdots L_{g_{i_{k-1}}} h(x) \cdot u_0^{|I_k|}.$$

### 3.3 Taylorentwicklung des kontinuierlichen Systems

In diesem Abschnitt wird (3.9) genauer untersucht. Hierzu sei an die Taylorentwicklung einer Funktion  $h$  an der Stelle  $t = T$  erinnert

$$h(\Phi(T, x)) - h(x) = \sum_{i=1}^{\infty} \frac{T^i}{i!} \left. \frac{d^i h(\Phi(t, x))}{dt^i} \right|_{t=0}. \tag{3.18}$$

Besonders zu beachten ist hierbei, dass wegen der Definition unseres nichtlinearen affinen Kontrollsystems (2.16)

$$\left. \frac{d^i h(\Phi(t, x))}{dt^i} \right|_{t=0} = L_{g_0+g_1 u_0}^i h(x) \quad (3.19)$$

gilt. Folglich liefert die Auswertung der ersten Ableitung

$$\begin{aligned} \left. \frac{dh(\Phi(t, x))}{dt} \right|_{t=0} &= L_{g_0+g_1 u_0} h(x) \\ &= L_{g_0} h(x) + L_{g_1} h(x) \cdot u_0(x). \end{aligned} \quad (3.20)$$

Bei der Darstellung der weiteren Ableitungen wird auf das Argument aus notationstechnischen Gründen verzichtet. So ergeben sich für die zweite bis fünfte Ableitung folgende Ausdrücke:

$$\begin{aligned} \left. \frac{d^2 h(\Phi(t, x))}{dt^2} \right|_{t=0} &= L_{g_0+g_1 u_0}^2 h \\ &= L_{g_0+g_1 u_0} [L_{g_0} h + L_{g_1} h \cdot u_0] \\ &= L_{g_0} L_{g_0} h + L_{g_1} L_{g_0} h u_0 + L_{g_0} L_{g_1} h u_0 + L_{g_1} L_{g_1} h u_0^2 \\ &\quad + L_{g_1} h \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \\ &= L_{g_0} L_{g_0} h + [L_{g_1} L_{g_0} h + L_{g_0} L_{g_1} h] \cdot u_0 + L_{g_1} L_{g_1} h u_0^2 \\ &\quad + L_{g_1} h \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0], \end{aligned} \quad (3.21)$$

$$\begin{aligned}
 \left. \frac{d^3 h(\Phi(t, x))}{dt^3} \right|_{t=0} &= L_{g_0+g_1} u_0 \left[ L_{g_0} L_{g_0} h + [L_{g_1} L_{g_0} h + L_{g_0} L_{g_1} h] u_0 + L_{g_1} L_{g_1} h u_0^2 \right. \\
 &+ \left. L_{g_1} h \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \right] \\
 &= L_{g_0} L_{g_0} L_{g_0} h + L_{g_1} L_{g_0} L_{g_0} h u_0 \\
 &+ L_{g_0} [L_{g_1} L_{g_0} h + L_{g_0} L_{g_1} h] u_0 + L_{g_1} [L_{g_1} L_{g_0} h + L_{g_0} L_{g_1} h] u_0^2 \\
 &+ [L_{g_1} L_{g_0} h + L_{g_0} L_{g_1} h] \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \\
 &+ L_{g_0} L_{g_1} L_{g_1} h u_0^2 + L_{g_1} L_{g_1} L_{g_1} h u_0^3 + 2L_{g_1} L_{g_1} h u_0 \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \\
 &+ [L_{g_0} L_{g_1} h + L_{g_1} L_{g_1} h u_0] \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \\
 &+ L_{g_1} h \frac{\partial}{\partial x} \left[ \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \\
 &= L_{g_0} L_{g_0} L_{g_0} h + [L_{g_1} L_{g_0} L_{g_0} h + L_{g_0} L_{g_1} L_{g_0} h + L_{g_0} L_{g_0} L_{g_1} h] u_0 \\
 &+ [L_{g_1} L_{g_1} L_{g_0} h + L_{g_1} L_{g_0} L_{g_1} h + L_{g_0} L_{g_1} L_{g_1} h] u_0^2 + L_{g_1} L_{g_1} L_{g_1} h u_0^3 \\
 &+ [L_{g_1} L_{g_0} h + 2L_{g_0} L_{g_1} h + 3L_{g_1} L_{g_1} h u_0] \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \\
 &+ L_{g_1} h \frac{\partial}{\partial x} \left[ \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0], \tag{3.22}
 \end{aligned}$$

$$\begin{aligned}
\left. \frac{d^4 h(\Phi(t, x))}{dt^4} \right|_{t=0} &= L_{g_0} L_{g_0} L_{g_0} L_{g_0} h + L_{g_1} L_{g_0} L_{g_0} L_{g_0} h u_0 \\
&+ L_{g_0} [L_{g_1} L_{g_0} L_{g_0} h + L_{g_0} L_{g_1} L_{g_0} h + L_{g_0} L_{g_0} L_{g_1} h] u_0 \\
&+ L_{g_1} [L_{g_1} L_{g_0} L_{g_0} h + L_{g_0} L_{g_1} L_{g_0} h + L_{g_0} L_{g_0} L_{g_1} h] u_0^2 \\
&+ [L_{g_1} L_{g_0} L_{g_0} h + L_{g_0} L_{g_1} L_{g_0} h + L_{g_0} L_{g_0} L_{g_1} h] \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \\
&+ L_{g_0} [L_{g_1} L_{g_1} L_{g_0} h + L_{g_1} L_{g_0} L_{g_1} h + L_{g_0} L_{g_1} L_{g_1} h] u_0^2 \\
&+ L_{g_1} [L_{g_1} L_{g_1} L_{g_0} h + L_{g_1} L_{g_0} L_{g_1} h + L_{g_0} L_{g_1} L_{g_1} h] u_0^3 \\
&+ [L_{g_1} L_{g_1} L_{g_0} h + L_{g_1} L_{g_0} L_{g_1} h + L_{g_0} L_{g_1} L_{g_1} h] 2u_0 \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \\
&+ L_{g_0} L_{g_1} L_{g_1} L_{g_1} h u_0^3 + L_{g_1} L_{g_1} L_{g_1} L_{g_1} h u_0^4 + L_{g_1} L_{g_1} L_{g_1} h 3u_0^2 \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \\
&+ L_{g_0} [L_{g_1} L_{g_0} h + 2L_{g_0} L_{g_1} h + 3L_{g_1} L_{g_1} h u_0] \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \\
&+ L_{g_1} [L_{g_1} L_{g_0} h + 2L_{g_0} L_{g_1} h + 3L_{g_1} L_{g_1} h u_0] u_0 \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \\
&+ [L_{g_1} L_{g_0} h + 2L_{g_0} L_{g_1} h + 3L_{g_1} L_{g_1} h u_0] \frac{\partial}{\partial x} \left[ \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \\
&+ 3L_{g_1} L_{g_1} h \left( \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \right)^2 \\
&+ [L_{g_0} L_{g_1} h + L_{g_1} L_{g_1} h u_0] \frac{\partial}{\partial x} \left[ \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \\
&+ L_{g_1} h \frac{\partial}{\partial x} \left[ \frac{\partial}{\partial x} \left[ \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \\
&= L_{g_0} L_{g_0} L_{g_0} L_{g_0} h \\
&+ [L_{g_0} L_{g_0} L_{g_0} L_{g_1} h + L_{g_0} L_{g_0} L_{g_1} L_{g_0} h + L_{g_0} L_{g_1} L_{g_0} L_{g_0} h + L_{g_1} L_{g_0} L_{g_0} L_{g_0} h] u_0 \\
&+ [L_{g_0} L_{g_0} L_{g_1} L_{g_1} h + L_{g_0} L_{g_1} L_{g_0} L_{g_1} h + L_{g_1} L_{g_0} L_{g_0} L_{g_1} h \\
&+ L_{g_0} L_{g_1} L_{g_1} L_{g_0} h + L_{g_1} L_{g_0} L_{g_1} L_{g_0} h + L_{g_1} L_{g_1} L_{g_0} L_{g_0} h] u_0^2 \\
&+ [L_{g_1} L_{g_1} L_{g_1} L_{g_0} h + L_{g_1} L_{g_1} L_{g_0} L_{g_1} h + L_{g_1} L_{g_0} L_{g_1} L_{g_1} h + L_{g_0} L_{g_1} L_{g_1} L_{g_1} h] u_0^3 \\
&+ L_{g_1} L_{g_1} L_{g_1} L_{g_1} h u_0^4 \\
&+ [L_{g_1} L_{g_0} L_{g_0} h + 2L_{g_0} L_{g_1} L_{g_0} h + 3L_{g_0} L_{g_0} L_{g_1} h] \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \\
&+ [3L_{g_1} L_{g_1} L_{g_0} h + 4L_{g_1} L_{g_0} L_{g_1} h + 5L_{g_0} L_{g_1} L_{g_1} h] u_0 \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \\
&+ 6L_{g_1} L_{g_1} L_{g_1} h u_0^2 \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \\
&+ [L_{g_1} L_{g_0} h + 3L_{g_0} L_{g_1} h + 4L_{g_1} L_{g_1} h u_0] \frac{\partial}{\partial x} \left[ \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \\
&+ 3L_{g_1} L_{g_1} h \left( \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \right)^2 \\
&+ L_{g_1} h \frac{\partial}{\partial x} \left[ \frac{\partial}{\partial x} \left[ \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0], \tag{3.23}
\end{aligned}$$

$$\begin{aligned}
 \left. \frac{d^5 h(\Phi(t, x))}{dt^5} \right|_{t=0} &= L_{g_0}^5 h + [L_{g_0}^4 L_{g_1} h + L_{g_0}^3 L_{g_1} L_{g_0} h + L_{g_0}^2 L_{g_1} L_{g_0}^2 h + L_{g_0} L_{g_1} L_{g_0}^3 h + L_{g_1} L_{g_0}^4 h] u_0 \\
 &+ [L_{g_0}^3 L_{g_1}^2 h + L_{g_0}^2 L_{g_1} L_{g_0} L_{g_1} h + L_{g_0} L_{g_1} L_{g_0}^2 L_{g_1} h + L_{g_1} L_{g_0}^3 L_{g_1} h \\
 &+ L_{g_0}^2 L_{g_1}^2 L_{g_0} h + L_{g_0} L_{g_1} L_{g_0} L_{g_1} L_{g_0} h + L_{g_1} L_{g_0}^2 L_{g_1} L_{g_0} h + L_{g_0} L_{g_1}^2 L_{g_0}^2 h \\
 &+ L_{g_1} L_{g_0} L_{g_1} L_{g_0}^2 h + L_{g_1}^2 L_{g_0}^3 h] u_0^2 \\
 &+ [L_{g_0}^2 L_{g_1}^3 h + L_{g_0} L_{g_1} L_{g_0} L_{g_1}^2 h + L_{g_1} L_{g_0}^2 L_{g_1}^2 h + L_{g_0} L_{g_1}^2 L_{g_0} L_{g_1} h \\
 &+ L_{g_1} L_{g_0} L_{g_1} L_{g_0} L_{g_1} h + L_{g_1}^2 L_{g_0}^2 L_{g_1} h + L_{g_0} L_{g_1}^3 L_{g_0} h + L_{g_1} L_{g_0} L_{g_1}^2 L_{g_0} h \\
 &+ L_{g_1}^2 L_{g_0} L_{g_1} L_{g_0} h + L_{g_1}^3 L_{g_0}^2 h] u_0^3 \\
 &+ [L_{g_0} L_{g_1}^4 h + L_{g_1} L_{g_0} L_{g_1}^3 h + L_{g_1}^2 L_{g_0} L_{g_1}^2 h + L_{g_1}^3 L_{g_0} L_{g_1} h + L_{g_1}^4 L_{g_0} h] u_0^4 + L_{g_1}^5 h u_0^5 \\
 &+ [L_{g_1} L_{g_0}^3 h + 2L_{g_0} L_{g_1} L_{g_0}^2 h + 3L_{g_0}^2 L_{g_1} L_{g_0} h + 4L_{g_0}^3 L_{g_1} h] \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \\
 &+ [3L_{g_1}^2 L_{g_0}^2 h + 4L_{g_1} L_{g_0} L_{g_1} L_{g_0} h + 5L_{g_0} L_{g_1}^2 L_{g_0} h \\
 &+ 5L_{g_1} L_{g_0}^2 L_{g_1} h + 6L_{g_0} L_{g_1} L_{g_0} L_{g_1} h + 7L_{g_0}^2 L_{g_1}^2 h] u_0 \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \\
 &+ [6L_{g_1}^3 L_{g_0} h + 7L_{g_1}^2 L_{g_0} L_{g_1} h + 8L_{g_1} L_{g_0} L_{g_1}^2 h + 9L_{g_0} L_{g_1}^3 h] u_0^2 \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \\
 &+ 10L_{g_1}^4 h u_0^3 \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \\
 &+ [3L_{g_1}^2 L_{g_0} h + 4L_{g_1} L_{g_0} L_{g_1} h + 8L_{g_0} L_{g_1}^2 h] \left( \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \right)^2 \\
 &+ 15L_{g_1}^3 u_0 \left( \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \right)^2 \\
 &+ [L_{g_1} L_{g_0}^2 h + 3L_{g_0} L_{g_1} L_{g_0} h + 6L_{g_0}^2 L_{g_1} h] \frac{\partial}{\partial x} \left[ \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \\
 &+ [4L_{g_1}^2 L_{g_0} h + 7L_{g_1} L_{g_0} L_{g_1} h + 9L_{g_0} L_{g_1}^2 h] u_0 \frac{\partial}{\partial x} \left[ \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \\
 &+ 10L_{g_1}^3 h u_0^2 \frac{\partial}{\partial x} \left[ \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \\
 &+ 10L_{g_1}^2 h \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \frac{\partial}{\partial x} \left[ \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \\
 &+ [L_{g_1} L_{g_0} h + 4L_{g_0} L_{g_1} h] \frac{\partial}{\partial x} \left[ \frac{\partial}{\partial x} \left[ \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \\
 &+ 5L_{g_1}^2 h u_0 \frac{\partial}{\partial x} \left[ \frac{\partial}{\partial x} \left[ \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \\
 &+ L_{g_1} h \frac{\partial}{\partial x} \left\{ \frac{\partial}{\partial x} \left[ \frac{\partial}{\partial x} \left[ \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \right\} [g_0 + g_1 u_0]. \quad (3.24)
 \end{aligned}$$

Der Ausdruck für die sechste Ableitung ist bereits derart lang, dass es sinnvoll ist, ihn in seine einzelnen Teile zu zerlegen.

$$\begin{aligned}
\left. \frac{d^6 h(\Phi(t, x))}{dt^6} \right|_{t=0} &= L_{g_0}^6 h + [L_{g_1} L_{g_0}^5 + L_{g_0} L_{g_1} L_{g_0}^4 + L_{g_0}^2 L_{g_1} L_{g_0}^3 + L_{g_0}^3 L_{g_1} L_{g_0}^2 + L_{g_0}^4 L_{g_1} L_{g_0} + L_{g_0}^5 L_{g_1}] h u_0 \\
&+ [L_{g_1} [L_{g_0}^4 L_{g_1} + L_{g_0}^3 L_{g_1} L_{g_0} + L_{g_0}^2 L_{g_1} L_{g_0}^2 + L_{g_0} L_{g_1} L_{g_0}^3 + L_{g_1} L_{g_0}^4] \\
&+ L_{g_0} [L_{g_0}^3 L_{g_1}^2 + L_{g_0}^2 L_{g_1} L_{g_0} L_{g_1} + L_{g_0} L_{g_1} L_{g_0}^2 L_{g_1} + L_{g_1} L_{g_0}^3 L_{g_1} \\
&+ [L_{g_0}^2 L_{g_1}^2 + L_{g_0} L_{g_1} L_{g_0} L_{g_1} + L_{g_1} L_{g_0}^2 L_{g_1} + L_{g_0} L_{g_1}^2 L_{g_0} + L_{g_1} L_{g_0} L_{g_1} L_{g_0} + L_{g_0}^2 L_{g_1}^2] L_{g_0}] h u_0^2 \\
&+ [L_{g_1} [L_{g_0}^3 L_{g_1}^2 + L_{g_0}^2 L_{g_1} L_{g_0} L_{g_1} + L_{g_0} L_{g_1} L_{g_0}^2 L_{g_1} + L_{g_1} L_{g_0}^3 L_{g_1} \\
&+ [L_{g_0}^2 L_{g_1}^2 + L_{g_0} L_{g_1} L_{g_0} L_{g_1} + L_{g_1} L_{g_0}^2 L_{g_1} + L_{g_0} L_{g_1}^2 L_{g_0} + L_{g_1} L_{g_0} L_{g_1} L_{g_0} + L_{g_0}^2 L_{g_1}^2] L_{g_0}] \\
&+ L_{g_0} [L_{g_1}^3 L_{g_0}^2 + L_{g_1}^2 L_{g_0} L_{g_1} L_{g_0} + L_{g_1} L_{g_0} L_{g_1}^2 L_{g_0} + L_{g_0} L_{g_1}^3 L_{g_0} \\
&+ [L_{g_1}^2 L_{g_0}^2 + L_{g_1} L_{g_0} L_{g_1} L_{g_0} + L_{g_0} L_{g_1}^2 L_{g_0} + L_{g_1} L_{g_0}^2 L_{g_1} + L_{g_0} L_{g_1} L_{g_0} L_{g_1} + L_{g_0}^2 L_{g_1}^2] L_{g_1}] h u_0^3 \\
&+ [L_{g_1} [L_{g_0}^3 L_{g_1}^2 + L_{g_0}^2 L_{g_1} L_{g_0} L_{g_1} + L_{g_0} L_{g_1} L_{g_0}^2 L_{g_1} + L_{g_0} L_{g_1}^3 L_{g_0} \\
&+ [L_{g_1}^2 L_{g_0}^2 + L_{g_1} L_{g_0} L_{g_1} L_{g_0} + L_{g_0} L_{g_1}^2 L_{g_0} + L_{g_1} L_{g_0}^2 L_{g_1} + L_{g_0} L_{g_1} L_{g_0} L_{g_1} + L_{g_0}^2 L_{g_1}^2] L_{g_1}] \\
&+ L_{g_0} [L_{g_1}^4 L_{g_0} + L_{g_1}^3 L_{g_0} L_{g_1} + L_{g_1}^2 L_{g_0} L_{g_1}^2 + L_{g_1} L_{g_0} L_{g_1}^3 + L_{g_0} L_{g_1}^4] h u_0^4 \\
&+ [L_{g_1} [L_{g_0}^4 L_{g_1} + L_{g_0}^3 L_{g_1} L_{g_0} + L_{g_0}^2 L_{g_1} L_{g_0}^2 + L_{g_0} L_{g_1} L_{g_0}^3 + L_{g_0} L_{g_1}^4] + L_{g_0} L_{g_1}^5] h u_0^5 \\
&+ L_{g_1}^6 h u_0^6 + A_1, \tag{3.25}
\end{aligned}$$

wobei  $A_1$  für alle Terme mit einer Ableitung von  $u_0$  steht. Die Terme mit  $\frac{\partial u_0}{\partial x} [g_0 + g_1 u_0]$  lauten

$$\begin{aligned}
&[L_{g_1} L_{g_0}^4 + 2L_{g_0} L_{g_1} L_{g_0}^3 + 3L_{g_0}^2 L_{g_1} L_{g_0}^2 + 4L_{g_0}^3 L_{g_1} L_{g_0} + 5L_{g_0}^4 L_{g_1}] h \tag{3.26} \\
&+ [L_{g_1} [3L_{g_1} L_{g_0}^3 + 4L_{g_0} L_{g_1} L_{g_0}^2 + 5L_{g_0}^2 L_{g_1} L_{g_0} + 6L_{g_0}^3 L_{g_1}] \\
&+ L_{g_0} [5L_{g_1}^2 L_{g_0}^2 + 6L_{g_1} L_{g_0} L_{g_1} L_{g_0} + 7L_{g_1} L_{g_0}^2 L_{g_1} + 7L_{g_0} L_{g_1}^2 L_{g_0} + 8L_{g_0} L_{g_1} L_{g_0} L_{g_1} + 9L_{g_0}^2 L_{g_1}^2]] h u_0 \\
&+ [L_{g_1}^2 [6L_{g_1} L_{g_0}^2 + 7L_{g_0} L_{g_1} L_{g_0} + 8L_{g_0}^2 L_{g_1}] + L_{g_1} L_{g_0} [8L_{g_1}^2 L_{g_0} + 9L_{g_1} L_{g_0} L_{g_1} + 10L_{g_0} L_{g_1}^2] \\
&+ L_{g_0} [9L_{g_1}^3 L_{g_0} + 10L_{g_1}^2 L_{g_0} L_{g_1} + 11L_{g_1} L_{g_0} L_{g_1}^2 + 12L_{g_0} L_{g_1}^3]] h u_0^2 \\
&+ [L_{g_1} [10L_{g_1}^3 L_{g_0} + 11L_{g_1}^2 L_{g_0} L_{g_1} + 12L_{g_1} L_{g_0} L_{g_1}^2 + 13L_{g_0} L_{g_1}^3] + 14L_{g_0} L_{g_1}^4] h u_0^3 + 15L_{g_1}^5 h u_0^4,
\end{aligned}$$

für die zweite Ableitung  $\frac{\partial}{\partial x} \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0][g_0 + g_1 u_0]$  ergeben sich

$$\begin{aligned}
&[L_{g_1} L_{g_0}^3 + 3L_{g_0} L_{g_1} L_{g_0}^2 + 6L_{g_0}^2 L_{g_1} L_{g_0} + 10L_{g_0}^3 L_{g_1}] h \\
&+ [L_{g_1} [4L_{g_1} L_{g_0}^2 + 7L_{g_0} L_{g_1} L_{g_0} + 11L_{g_0}^2 L_{g_1}] + L_{g_0} [9L_{g_1}^2 L_{g_0} + 13L_{g_1} L_{g_0} L_{g_1} + 16L_{g_0} L_{g_1}^2]] h u_0 \\
&+ [10L_{g_1}^3 L_{g_0} + 14L_{g_1}^2 L_{g_0} L_{g_1} + 17L_{g_1} L_{g_0} L_{g_1}^2 + 19L_{g_0} L_{g_1}^3] h u_0^2 + 20L_{g_1}^4 h u_0^3. \tag{3.27}
\end{aligned}$$

Die Terme mit dritter Ableitung sind

$$\begin{aligned}
&[L_{g_1} L_{g_0}^2 + 4L_{g_0} L_{g_1} L_{g_0} + 10L_{g_0}^2 L_{g_1}] h \\
&+ [5L_{g_1}^2 L_{g_0} + 11L_{g_1} L_{g_0} L_{g_1} + 14L_{g_0} L_{g_1}^2] h u_0 + 15L_{g_1}^3 h u_0^2, \tag{3.28}
\end{aligned}$$

für die vierte Ableitung ergibt sich  $[L_{g_1}L_{g_0} + 5L_{g_0}L_{g_1}]h + 6L_{g_1}^2 hu_0$  und für die fünfte Ableitung  $L_{g_1}h$ . Es bleiben die Terme, in denen mehr als ein Ableitungsausdruck für  $u_0$  vorkommt. So lauten die entsprechenden Terme für  $(\frac{\partial u_0}{\partial x}[g_0 + g_1 u_0])^2$

$$\begin{aligned} & [L_{g_1}[3L_{g_1}L_{g_0}^2 + 4L_{g_0}L_{g_1}L_{g_0} + 5L_{g_0}^2L_{g_1}]h + L_{g_0}[8L_{g_1}^2L_{g_0} + 10L_{g_1}L_{g_0}L_{g_1} + 15L_{g_0}L_{g_1}^2]]h \\ & + [15L_{g_1}^3L_{g_0} + 18L_{g_1}^2L_{g_0}L_{g_1} + 24L_{g_1}L_{g_0}L_{g_1}^2 + 33L_{g_0}L_{g_1}^3]hu_0 + 45L_{g_1}^4hu_0^2 \\ & + 15L_{g_1}^3hu_0\frac{\partial u_0}{\partial x}[g_0 + g_1u_0]. \end{aligned} \quad (3.29)$$

Es fehlen noch die gemischten Terme höherer Ordnung

$$\begin{aligned} & [[10L_{g_1}^2L_{g_0} + 15L_{g_1}L_{g_0}L_{g_1} + 35L_{g_0}L_{g_1}^2]h + 60L_{g_1}^3hu_0]\frac{\partial}{\partial x}\frac{\partial u_0}{\partial x}[g_0 + g_1u_0][g_0 + g_1u_0]\frac{\partial u_0}{\partial x}[g_0 + g_1u_0] \\ & + 10L_{g_1}^2h\left(\frac{\partial}{\partial x}\frac{\partial u_0}{\partial x}[g_0 + g_1u_0][g_0 + g_1u_0]\right)^2 \\ & + 15L_{g_1}^2h\frac{\partial}{\partial x}\frac{\partial}{\partial x}\frac{\partial u_0}{\partial x}[g_0 + g_1u_0][g_0 + g_1u_0][g_0 + g_1u_0]\frac{\partial u_0}{\partial x}[g_0 + g_1u_0]. \end{aligned} \quad (3.30)$$

### 3.4 Vergleich von Fließ- und Taylorentwicklung

Nachdem wir nun sowohl Aussagen über (3.10) als auch über (3.9) hergeleitet haben, können wir (3.11) genauer untersuchen. Auffällig dabei ist, dass in der Taylorentwicklung nur  $u_0$  und Ableitungsausdrücke von  $u_0$  vorkommen. In der Fließentwicklung treten im Gegensatz dazu Kontrollterme  $u_0, u_1, u_2$  usw. auf. In [21] wurde zur Lösung für ein ähnliches Problem

$$u_T^M(x) = \sum_{i=0}^M \frac{T^i}{(i+1)!} \frac{d^i u(\Phi(t, x))}{dt^i} \Big|_{t=0} \quad (3.31)$$

genutzt, um diese Tatsache in den Griff zu bekommen. Konkret heißt das, dass in (2.34)

$$u_i(x) = \frac{1}{(i+1)!} \frac{d^i u(\Phi(t, x))}{dt^i} \Big|_{t=0} \quad (3.32)$$

eingesetzt wird. Es wird also die Form (3.32) für die gesuchten Kontrollfunktionen  $u_1, u_2, \dots, u_M$  vorausgesetzt. Nach dieser Konvention sind erste Aussagen über (3.11) möglich. Dies geschieht in Form eines Theorems<sup>1</sup> aus [21], das allerdings nach der Vorarbeit in den Abschnitten 3.2 und 3.3 nur noch eine einfache, aber wichtige Folgerung darstellt. Bevor wir dieses Theorem allerdings rezitieren, können wir aufbauend auf (3.32) folgende Bemerkung über die Koeffizienten der beiden Reihenentwicklungen machen.

<sup>1</sup>Theorem 4.11

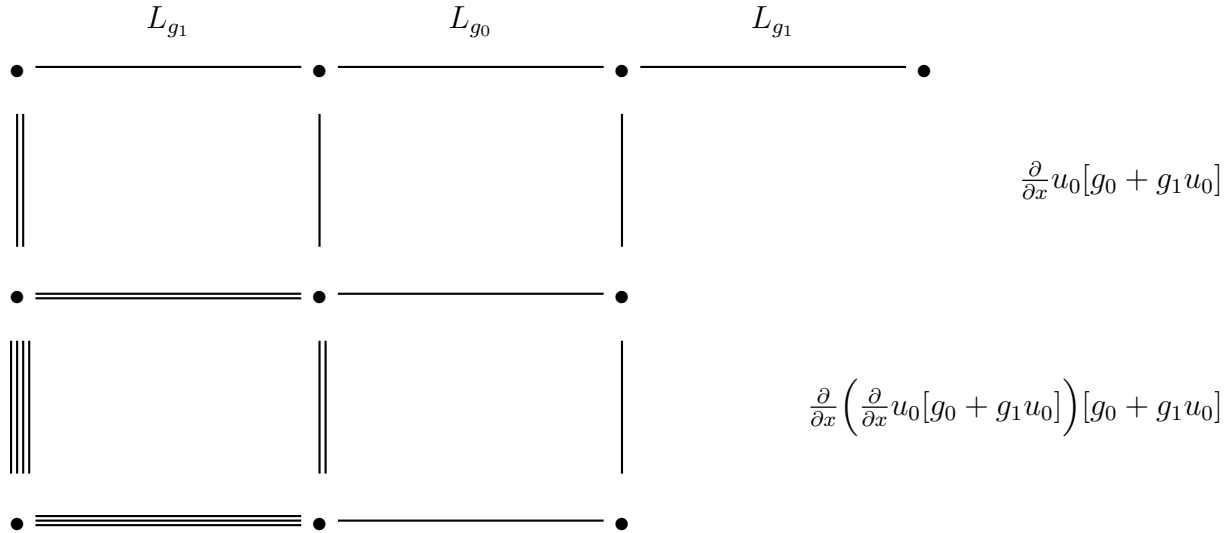


**Bemerkung 3.2** In der Fließentwicklung für die Ordnung  $M$  tauchen diesselben Terme wie in der Taylorentwicklung der gleichen Ordnung auf. Allerdings unterscheiden sich die Koeffizienten vor den einzelnen Termen der jeweiligen Reihenentwicklungen. Die Fließentwicklung besitzt ihre Symmetrieeigenschaften auch nach Verwendung von (3.32), d.h.  $L_{g_1} L_{g_1} L_{g_0} L_{g_0} h u_0 \frac{\partial}{\partial x} u_0 [g_0 + g_1 u_0]$  hat beispielsweise den gleichen Koeffizienten wie  $L_{g_1} L_{g_0} L_{g_1} L_{g_0} h u_0 \frac{\partial}{\partial x} u_0 [g_1 u_0]$ . Es entscheidet nur die Anzahl der  $L_{g_1}$ - beziehungsweise  $L_{g_0}$ -Operatoren und nicht deren Anordnung über die Koeffizienten. In der Taylorentwicklung hingegen haben die oben erwähnten Ausdrücke unterschiedliche Koeffizienten. Es stellt sich die Frage, ob man diese algebraisch oder analytisch bestimmen kann, ohne iterativ die gesamten Ableitungsausdrücke zu berechnen. Problemlos sind dabei die Koeffizienten von Termen ohne Ableitungsausdruck einer Kontrollfunktion, diese haben immer den Koeffizienten eins. Wir wollen uns an dieser Stelle lediglich mit Termen beschäftigen, die nur einen Ableitungsausdruck besitzen, also

$$L_{g_{i_1}} \dots L_{g_{i_j}} u_0^{k-1} \frac{\partial}{\partial x} \left( \dots \left[ \frac{\partial}{\partial x} u_0 [g_0 + g_1 u_0] \right] \dots \right) [g_0 + g_1 u_0] \quad (3.33)$$

mit  $i_1, \dots, i_j \in \{0, 1\}$  und  $j, k \in \mathbb{N}$  mit  $k \leq j$  und  $\sum_{h=1}^j i_h = k$ .

Nun gibt es zwei Methoden, mit denen man die Koeffizienten bestimmen kann. Die erste sei exemplarisch für  $L_{g_1} L_{g_0} L_{g_1} h u_0 \frac{\partial}{\partial x} \left( \frac{\partial}{\partial x} u_0 [g_0 + g_1 u_0] \right) [g_0 + g_1 u_0]$  demonstriert. Begonnen wird in der folgenden Graphik mit dem Punkt oben rechts. Es wird zuerst der Operator  $L_{g_1}$  angewendet. Danach kann entweder erst der nächste Operator angewendet werden oder das nun existierende  $u_0$  erstmalig differenziert werden. Dies wird durch die Linien nach links beziehungsweise unten dargestellt. Betrachtet man den Punkt oben links fällt auf, dass zwei Linien nach unten zeigen. Dies lässt sich dadurch erklären, dass mittlerweile zweimal  $u_0$  vorhanden ist und man jedes der beiden differenzieren könnte. Es ergibt sich die Zahl sieben als Koeffizient, was mit (3.24) für diesen Term übereinstimmt. Mit dieser Methode lassen sich sogar die Koeffizienten für alle vorkommenden Terme berechnen, allerdings benötigt man für die Darstellung für jeden weiteren Ableitungsausdruck eines  $u_0$  eine zusätzliche Dimension. Außerdem ist der Aufwand für jeden Koeffizienten verhältnismäßig hoch, so dass das Verfahren schnell an seine Grenze in Bezug auf die praktische Anwendung stößt. Es ist also nur geeignet, wenn man einen ganz speziellen Koeffizienten bestimmen möchte.



Beschränken wir uns aber auf Terme der Form (3.33) kann man die Koeffizienten algebraisch mit Hilfe des aus [4] bekannten Pascalschen Dreiecks bestimmen. So liefert uns jeder weitere Ableitungsausdruck eine zusätzliche Gegendiagonale in der nachfolgenden Abbildung beziehungsweise eine zusätzliche Zeile des Pascalschen Dreiecks ohne die abschließende Eins. Um dies einzusehen, bedarf die Graphik jedoch einer kurzen Erklärung. Die Spalte gibt die Position des  $L_{g_1}$ -Operators an, so steht beispielsweise die erste Spalte für  $L_{g_1} L_{g_0}^{j-1}$ . Tauschen mehrere  $L_{g_1}$ -Operatoren in einem Term auf, so ergibt sich der gesuchte Koeffizient als Summe der entsprechenden Summanden der einzelnen  $L_{g_1}$ -Operatoren. Bezogen auf unser Beispiel resultiert daraus ebenfalls die Zahl sieben. Dieses Mal als Summe der Zahlen eins und sechs, die in der zweiten Zeile  $\left( \frac{\partial}{\partial x} \left( \frac{\partial}{\partial x} u_0 [g_0 + g_1 u_0] \right) [g_0 + g_1 u_0] \right)$  an erster und dritter Position ( $L_{g_1} L_{g_0} L_{g_1}$ ) stehen. In der Graphik ist die Fortführung für Terme höherer Ordnung angedeutet. Diese erfolgt analog dem Pascalschen Dreieck, was wir anhand des aufgeführten Koeffizienten erläutern wollen. Es geht um den Term

$$L_{g_0} L_{g_0} L_{g_0} L_{g_1} \frac{\partial}{\partial x} \left[ \frac{\partial}{\partial x} \left( \frac{\partial}{\partial x} u_0 [g_0 + g_1 u_0] \right) [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \quad (3.34)$$

aus der siebten Ableitung mit  $L_{g_1}$  an vierter Position. Dieser kann sich sowohl aus

$$L_{g_0} L_{g_0} L_{g_1} \frac{\partial}{\partial x} \left[ \frac{\partial}{\partial x} \left( \frac{\partial}{\partial x} u_0 [g_0 + g_1 u_0] \right) [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0]$$

als auch aus

$$L_{g_0} L_{g_0} L_{g_0} L_{g_1} \frac{\partial}{\partial x} \left( \frac{\partial}{\partial x} u_0 [g_0 + g_1 u_0] \right) [g_0 + g_1 u_0] \quad (3.35)$$

ergeben. Also folgt  $10 + 10 = 20$  und damit die Behauptung.

$$\begin{array}{c}
 \frac{\partial}{\partial x} u_0 [g_0 + g_1 u_0] \\
 \frac{\partial}{\partial x} \left( \frac{\partial}{\partial x} u_0 [g_0 + g_1 u_0] \right) [g_0 + g_1 u_0] \\
 \frac{\partial}{\partial x} \left[ \frac{\partial}{\partial x} \left( \frac{\partial}{\partial x} u_0 [g_0 + g_1 u_0] \right) [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \\
 \vdots
 \end{array}
 \qquad
 \begin{array}{c}
 \begin{array}{cccccc}
 & & (3.21) & & \dots & & (3.25) \\
 & & \swarrow & & \swarrow & & \swarrow \\
 1 & 2 & 3 & 4 & 5 & & \\
 1 & 3 & 6 & 10 & & & \\
 1 & 4 & 10 & 20 & & & \\
 1 & 5 & & & & & \\
 1 & & & & & & 
 \end{array}
 \end{array}$$

**Theorem 3.3** *Seien die in Kapitel 2 getroffenen Annahmen erfüllt. Dann gilt:*

$$\Delta h_{ct}(T, x) - \Delta h_{dt}(T, x, u_0(x)) = \mathcal{O}(T^2). \tag{3.36}$$

Definiere man nun analog (3.31) die Kontrollfunktion  $u_T^1(x)$ , d.h. (2.34) mit

$$u_1(x) = \frac{1}{2} \frac{\partial u_0(x)}{\partial x} [g_0(x) + g_1(x)u_0(x)], \tag{3.37}$$

also (3.32). Dann gilt

$$\Delta h_{ct}(T, x) - \Delta h_{dt}(T, x, u_T^1(x)) = \mathcal{O}(T^3). \tag{3.38}$$

*Beweis:* (3.36) folgt direkt aus dem Vergleich von (3.12) und (3.18) unter Beachtung von (3.20). Für (3.38) sind zusätzlich (3.13), (3.21) und die Tatsache, dass aus (3.37) unmittelbar

$$2u_1 = \frac{\partial u_0}{\partial x} [g_0 + g_1 u_0]$$

folgt, zu berücksichtigen. □

Aus diesem Theorem ergibt sich in natürlicher Weise die Fragestellung, ob allgemein für beliebiges, aber festes  $M \in \mathbb{N}$

$$\Delta h_{ct}(T, x) - \Delta h_{dt}(T, x, u_T^M(x)) = \mathcal{O}(T^{M+2}) \tag{3.39}$$

gilt. Also sich die Aussage des Theorems für beliebige Ordnungen mit Hilfe der entsprechend definierten Kontrollfunktionen verallgemeinern lässt. Nach (3.32) lauten die Kontrollfunktionen für  $M = 1, \dots, 5$

$$u_1(x) = \frac{1}{2!} \frac{\partial u_0}{dx} [g_0 + g_1 u_0], \quad (3.40)$$

$$u_2(x) = \frac{1}{3!} \frac{\partial}{\partial x} \left[ \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0], \quad (3.41)$$

$$u_3(x) = \frac{1}{4!} \frac{\partial}{\partial x} \left\{ \frac{\partial}{\partial x} \left[ \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \right\} [g_0 + g_1 u_0], \quad (3.42)$$

$$u_4(x) = \frac{1}{5!} \frac{\partial}{\partial x} \left[ \frac{\partial}{\partial x} \left\{ \frac{\partial}{\partial x} \left[ \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \right\} [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0], \quad (3.43)$$

$$u_5(x) = \frac{1}{6!} \frac{\partial}{\partial x} \left\{ \dots \left[ \dots \left\{ \dots \left[ \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \right] \dots \right\} \dots \right] \dots \right\} [g_0 + g_1 u_0]. \quad (3.44)$$

Beginnen wollen wir mit  $M = 2$ . Also ist ein Vergleich zwischen (3.22) mit dem Faktor  $\frac{T^3}{3!}$  und der Summe aus  $T^3 L_{g_1} h u_2$  und (3.14) mit dem Faktor  $T^3$  zu ziehen. Für die Differenz ergibt sich

$$\begin{aligned} & \left( \frac{1}{6} \frac{d^3 h(\Phi(t, x))}{dt^3} \Big|_{t=0} \right) - (L_{g_1} h u_2 + p_2(x, u_0, u_1)) \\ &= \left( \frac{1}{6} \left\{ [L_{g_1} L_{g_0} h + 2L_{g_0} L_{g_1} h + 3L_{g_1} L_{g_1} h u_0] \underbrace{\frac{\partial u_0}{dx} [g_0 + g_1 u_0]}_{(3.40)_{2u_1}} + L_{g_1} h \underbrace{\frac{\partial}{\partial x} \left[ \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0]}_{(3.41)_{6u_2}} \right\} \right) \\ &- \left( L_{g_1} h u_2 + \frac{1}{2} \left[ [L_{g_0} L_{g_1} h(x) + L_{g_1} L_{g_0} h(x)] u_1 + 2L_{g_1} L_{g_1} h(x) u_0 u_1 \right] \right) \\ &= \frac{1}{6} \left[ L_{g_0} L_{g_1} h - L_{g_1} L_{g_0} h \right] u_1, \end{aligned}$$

wobei wir den Faktor  $T^3$  der Übersichtlichkeit halber weggelassen haben. (3.39) gilt also bereits für  $M = 2$  im Allgemeinen nicht mehr. Es bedarf der zusätzlichen Bedingung

$$\frac{1}{6} \left[ L_{g_0} L_{g_1} h - L_{g_1} L_{g_0} h \right] u_1 = 0, \quad (3.45)$$

damit (3.39) für  $M = 2$  gilt.

**Bemerkung 3.4** Wertet man den Klammerausdruck in (3.45) mit Hilfe von (2.9) aus, erhält man

$$\begin{aligned} L_{g_0} L_{g_1} h - L_{g_1} L_{g_0} h &= g_1^T \frac{d^2}{dx^2} h g_0 + \nabla h J_{g_1} g_0 - g_0^T \frac{d^2}{dx^2} h g_1 - \nabla h J_{g_0} g_1 \\ &= \nabla h J_{g_1} g_0 - \nabla h J_{g_0} g_1 \\ &= \nabla h (J_{g_1} g_0 - J_{g_0} g_1), \end{aligned} \quad (3.46)$$

also die Ableitung von  $h$  in Richtung von  $J_{g_1} g_0 - J_{g_0} g_1$ . Diese muss multipliziert mit  $u_1$  gleich Null sein, damit (3.45) erfüllt ist.

Möchte man nun für  $M = 3$ , also Terme fünfter Ordnung Bedingungen finden, benötigt man (3.40), (3.41) und (3.42) um die Terme aus der Taylorentwicklung (3.23) und  $L_{g_1} h u_3 + p_3(x, u_0, u_1, u_2)$  aus der Fließentwicklung unter Beachtung von (3.15) vergleichen zu können. Es ergibt sich folgender Ausdruck

$$\begin{aligned}
& \left( \frac{1}{24} \frac{d^4 h(\Phi(t, x))}{dt^4} \Big|_{t=0} \right) - (L_{g_1} h u_3 + p_3(x, u_0, u_1, u_2)) \\
&= \left( \frac{1}{24} \left\{ [L_{g_1} L_{g_0} L_{g_0} h + 2L_{g_0} L_{g_1} L_{g_0} h + 3L_{g_0} L_{g_0} L_{g_1} h] \underbrace{\frac{\partial u_0}{dx} [g_0 + g_1 u_0]}_{=2u_1} \right. \right. \\
&+ [3L_{g_1} L_{g_1} L_{g_0} h + 4L_{g_1} L_{g_0} L_{g_1} h + 5L_{g_0} L_{g_1} L_{g_1} h] u_0 \underbrace{\frac{\partial u_0}{dx} [g_0 + g_1 u_0]}_{=2u_0 \cdot u_1} \\
&+ 6L_{g_1} L_{g_1} L_{g_1} h \underbrace{u_0^2 \frac{\partial u_0}{dx} [g_0 + g_1 u_0]}_{=2u_0^2 \cdot u_1} \\
&+ [L_{g_1} L_{g_0} h + 3L_{g_0} L_{g_1} h + 4L_{g_1} L_{g_1} h u_0] \underbrace{\frac{\partial}{\partial x} \left[ \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0]}_{=6u_2} \\
&+ 3L_{g_1} L_{g_1} h \underbrace{\left( \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \right)^2}_{=4u_1^2} \\
&+ \left. L_{g_1} h \underbrace{\frac{\partial}{\partial x} \left[ \frac{\partial}{\partial x} \left[ \frac{\partial u_0}{dx} [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0] \right] [g_0 + g_1 u_0]}_{=24u_3} \right\} \right) \\
&- \left( \frac{1}{2} \left[ [L_{g_0} L_{g_1} h(x) + L_{g_1} L_{g_0} h(x)] u_2 + L_{g_1} L_{g_1} h(x) [u_1^2 + 2u_0 u_2] \right] \right. \\
&+ \frac{1}{6} \left[ [L_{g_0} L_{g_0} L_{g_1} h(x) + L_{g_0} L_{g_1} L_{g_0} h(x) + L_{g_1} L_{g_0} L_{g_0} h(x)] u_1 \right. \\
&+ [L_{g_0} L_{g_1} L_{g_1} h(x) + L_{g_1} L_{g_0} L_{g_1} h(x) + L_{g_1} L_{g_1} L_{g_0} h(x)] 2u_0 u_1 \\
&+ \left. \left. 3L_{g_1} L_{g_1} L_{g_1} h(x) u_0^2 u_1 \right] \right) \\
&= \frac{1}{12} [L_{g_0} L_{g_0} L_{g_1} h - L_{g_1} L_{g_0} L_{g_0} h] u_1 \\
&+ \frac{1}{4} [L_{g_0} L_{g_1} h - L_{g_1} L_{g_0} h] u_2 \\
&+ \frac{1}{12} [L_{g_0} L_{g_1} L_{g_1} h - L_{g_1} L_{g_1} L_{g_0} h] u_0 u_1.
\end{aligned}$$

Erneut wurde der Faktor  $T^4$  der besseren Darstellung wegen weggelassen. Es ergibt sich also für  $M = 3$  folgende Bedingung

$$\begin{aligned}
 0 &= \frac{1}{12}[L_{g_0}L_{g_0}L_{g_1}h - L_{g_1}L_{g_0}L_{g_0}h]u_1 \\
 &+ \frac{1}{4}[L_{g_0}L_{g_1}h - L_{g_1}L_{g_0}h]u_2 \\
 &+ \frac{1}{12}[L_{g_0}L_{g_1}L_{g_1}h - L_{g_1}L_{g_1}L_{g_0}h]u_0u_1.
 \end{aligned} \tag{3.47}$$

Analog kann man zusätzliche Bedingungen für  $M = 4$  und  $M = 5$  herleiten. Hier verwendet man (3.23) und (3.16) bzw. (3.24) und (3.17), sowie (3.40)-(3.44). Die resultierenden Bedingungen lauten

$$\begin{aligned}
 0 &= \frac{3}{10}[L_{g_0}L_{g_1}h - L_{g_1}L_{g_0}h]u_3 \\
 &+ \frac{1}{60}[8L_{g_0}L_{g_0}L_{g_1}h - L_{g_0}L_{g_1}L_{g_0}h - 7L_{g_1}L_{g_0}L_{g_0}h]u_2 \\
 &+ \frac{1}{60}[7L_{g_0}L_{g_1}L_{g_1}h + L_{g_1}L_{g_0}L_{g_1}h - 8L_{g_1}L_{g_1}L_{g_0}h]u_0u_2 \\
 &+ \frac{1}{30}[3L_{g_0}L_{g_1}L_{g_1}h - L_{g_1}L_{g_0}L_{g_1}h - 2L_{g_1}L_{g_1}L_{g_0}h]u_1^2 \\
 &+ \frac{1}{120}[L_{g_0}L_{g_0}L_{g_1}L_{g_0}h + 3L_{g_0}L_{g_0}L_{g_0}L_{g_1}h - 3L_{g_1}L_{g_0}L_{g_0}L_{g_0}h - L_{g_0}L_{g_1}L_{g_0}L_{g_0}h]u_1 \\
 &+ \frac{1}{60}[2L_{g_0}L_{g_0}L_{g_1}L_{g_1}h + L_{g_0}L_{g_1}L_{g_0}L_{g_1}h - L_{g_1}L_{g_0}L_{g_1}L_{g_0}h - 2L_{g_1}L_{g_1}L_{g_0}L_{g_0}h]u_0u_1 \\
 &+ \frac{1}{120}[3L_{g_0}L_{g_1}L_{g_1}L_{g_1}h + L_{g_1}L_{g_0}L_{g_1}L_{g_1}h - L_{g_1}L_{g_1}L_{g_0}L_{g_1}h - 3L_{g_1}L_{g_1}L_{g_1}L_{g_0}h]u_0^2u_1
 \end{aligned} \tag{3.48}$$

und

$$\begin{aligned}
0 &= \frac{1}{3}[L_{g_0}L_{g_1} - L_{g_1}L_{g_0}]hu_4 \\
&+ \frac{1}{30}[5L_{g_0}^2L_{g_1} - 4L_{g_1}L_{g_0}^2 - L_{g_0}L_{g_1}L_{g_0}]hu_3 \\
&+ \frac{1}{30}[4L_{g_0}L_{g_1}^2 + L_{g_1}L_{g_0}L_{g_1} - 5L_{g_1}^2L_{g_0}]hu_3u_0 \\
&+ \frac{1}{12}[3L_{g_0}L_{g_1}^2 - 2L_{g_1}^2L_{g_0} - L_{g_1}L_{g_0}L_{g_1}]hu_2u_1 \\
&+ \frac{1}{360}[L_{g_0}[15L_{g_0}L_{g_1}^2 + 5L_{g_1}L_{g_0}L_{g_1} + L_{g_1}^2L_{g_0}] - L_{g_1}[5L_{g_0}^2L_{g_1} + 7L_{g_0}L_{g_1}L_{g_0} + 9L_{g_1}L_{g_0}^2]]hu_1^2 \\
&+ \frac{1}{120}[7L_{g_0}L_{g_1}^3 + L_{g_1}L_{g_0}L_{g_1}^2 - 3L_{g_1}^2L_{g_0}L_{g_1} - 5L_{g_1}^3L_{g_0}]hu_0u_1^2 \\
&+ \frac{1}{120}[5L_{g_0}^3L_{g_1} + L_{g_0}^2L_{g_1}L_{g_0} - 2L_{g_0}L_{g_1}L_{g_0}^2 - 4L_{g_1}L_{g_0}^3]hu_2 \\
&+ \frac{1}{120}[6L_{g_0}^2L_{g_1}^2 + 3L_{g_0}L_{g_1}L_{g_0}L_{g_1} + L_{g_1}L_{g_0}^2L_{g_1} - L_{g_0}L_{g_1}^2L_{g_0} - 3L_{g_1}L_{g_0}L_{g_1}L_{g_0} - 6L_{g_1}^2L_{g_0}^2]hu_0u_2 \\
&+ \frac{1}{120}[4L_{g_0}L_{g_1}^3 + 2L_{g_1}L_{g_0}L_{g_1}^2 - L_{g_1}^2L_{g_0}L_{g_1} - 5L_{g_1}^3L_{g_0}]hu_0^2u_2 \\
&+ \frac{1}{360}[2L_{g_0}^4L_{g_1} + L_{g_0}^3L_{g_1}L_{g_0} - L_{g_0}L_{g_1}L_{g_0}^3 - 2L_{g_1}L_{g_0}^4]hu_1 \\
&+ \frac{1}{360}[3L_{g_0}^3L_{g_1}^2 + 2L_{g_0}^2L_{g_1}L_{g_0}L_{g_1} + L_{g_0}^2L_{g_1}^2L_{g_0} + L_{g_0}L_{g_1}L_{g_0}^2L_{g_1} \\
&\quad - L_{g_0}L_{g_1}^2L_{g_0}^2 - L_{g_1}L_{g_0}^2L_{g_1}L_{g_0} - 2L_{g_1}L_{g_0}L_{g_1}L_{g_0}^2 - 3L_{g_1}^2L_{g_0}^3]hu_0u_1 \\
&+ \frac{1}{360}[3L_{g_0}^2L_{g_1}^3 + 2L_{g_0}L_{g_1}L_{g_0}L_{g_1}^2 + L_{g_0}L_{g_1}^2L_{g_0}L_{g_1} + L_{g_1}L_{g_0}^2L_{g_1}^2 \\
&\quad - L_{g_1}L_{g_0}L_{g_1}^2L_{g_0} - L_{g_1}^2L_{g_0}^2L_{g_1} - 2L_{g_1}^2L_{g_0}L_{g_1}L_{g_0} - 3L_{g_1}^3L_{g_0}^2]hu_0^2u_1 \\
&+ \frac{1}{360}[2L_{g_0}L_{g_1}^4 + L_{g_1}L_{g_0}L_{g_1}^3 - L_{g_1}^3L_{g_0}L_{g_1} - 2L_{g_1}^4L_{g_0}]hu_0^3u_1. \tag{3.49}
\end{aligned}$$

Zusammenfassend lässt sich folgendes auf Theorem 3.3 basierende Korollar beweisen

**Korollar 3.5** *Seien die in Kapitel 2 getroffenen Annahmen erfüllt und definiert man  $u_1, u_2, u_3, u_4$  und  $u_5$  analog (3.40)-(3.44) in (3.31). Dann gilt:*

$$\Delta h_{ct}(T, x) - \Delta h_{dt}(T, x, u_T^M(x)) = \mathcal{O}(T^{M+2}) \tag{3.50}$$

für

- $M = 2$ , falls (3.45) erfüllt ist.
- $M = 3$ , falls (3.45) und (3.47) gelten.

- $M = 4$ , falls (3.45), (3.47) und zusätzlich (3.48) erfüllt sind.
- $M = 5$ , falls (3.45), (3.47), (3.48) und darüberhinaus (3.49) gelten.

Beweis: Die Behauptung folgt unmittelbar aus den in diesem Abschnitt durchgeführten Rechnungen. □



# Kapitel 4

## Hinreichende Bedingungen

In Kapitel 3 wurde in Theorem 3.3 gezeigt, dass man mit Hilfe der Kontrollfunktion (3.31) mit  $M = 1$  das kontinuierliche closed-loop System (2.18) durch das sampled-data closed-loop System (2.32) bis zur dritten Ordnung exakt approximieren kann. Möchte man jedoch eine höhere Approximationsordnung erreichen, so benötigt man zusätzliche Bedingungen.

### 4.1 für beliebige Ordnung

Wie aus Korollar 3.5 zu entnehmen ist, werden diese mit steigender Ordnung zunehmend komplexer. In diesem Abschnitt wird eine hinreichende und unter gewissen Voraussetzungen auch notwendige Bedingung eingeführt, um mittels (3.31) Trajektorien für das sampled-data System zu erhalten, die das kontinuierliche System für einen hinreichend kleinen Zeitschritt beliebig gut approximieren. Dafür wird folgende Definition aus [12] benötigt.

**Definition 4.1** Eine Lie-Klammer  $[f, g]$  zweier Vektorfelder  $f, g : \mathbb{R}^n \rightarrow \mathbb{R}^n$  ist definiert durch

$$[f, g] := \frac{d}{dx}g \cdot f - \frac{d}{dx}f \cdot g. \quad (4.1)$$

**Bemerkung 4.2** Falls  $[f, g] = 0$  gilt, folgt die Gleichung

$$\frac{d}{dx}g \cdot f = \frac{d}{dx}f \cdot g.$$

In der bisher verwendeten Notation ausgedrückt also

$$L_f g = L_g f. \quad (4.2)$$

Hieraus folgt unmittelbar, dass die Operatoren kommutieren, d.h. es gilt  $L_f L_g = L_g L_f$ .

Es lässt sich damit folgendes Korollar formulieren:

**Korollar 4.3** *Betrachte das System (2.16), das zeitkontinuierliche closed-loop System (2.18) und das sampled-data closed-loop System (2.32) mit Kontrollfunktion  $u_T^2$  definiert durch (3.31). Angenommen die Bedingung*

$$[g_0, g_1] = 0 \quad (4.3)$$

*ist erfüllt, dann gilt*

$$\Delta h_{ct}(T, x) - \Delta h_{dt}(T, x, u_T^2(x)) = \mathcal{O}(T^4). \quad (4.4)$$

Beweis: Für den Beweis reicht es nach Theorem 3.3 zu zeigen, dass

$$\left( \frac{1}{6} \frac{d^3 h(\Phi(t, x))}{dt^3} \Big|_{t=0} \right) - (L_{g_1} h u_2 + p_2(x, u_0, u_1)) \stackrel{!}{=} 0 \quad (4.5)$$

ist. Das ist aber nach (3.45) genau dann der Fall, falls

$$\frac{1}{6} [L_{g_0} L_{g_1} - L_{g_1} L_{g_0}] h u_1 = 0$$

gilt. Es geht also um die aus Korollar 3.5 bekannte Bedingung für eine Approximation vierter Ordnung. Der Beweis folgt nun unmittelbar aus Bemerkung 4.2.  $\square$

Das Korollar könnte man mit den Resultaten aus Kapitel 3 problemlos bis zur siebten Ordnung ausbauen. Tatsächlich soll hier allerdings ein Theorem für beliebig vorgebbare Ordnung bewiesen werden und Korollar 4.3, insbesondere Gleichung (4.5), dient lediglich als Induktionsanfang. Zu erwähnen hierbei ist, dass dieser Abschnitt mit den folgenden Beweisen die Grundlage für Abschnitt 3 und Anhang A in [11] geliefert hat. Bevor wir das Theorem für beliebige Ordnungen formulieren, wird zuerst ein Lemma, das Gleichung (4.5) auch für höhere Ordnungen unter der Voraussetzung (4.3) zeigt, bewiesen.

**Lemma 4.4** *Betrachte das closed-loop System (2.18) mit Lösung  $\Phi(T, x)$  und nehme an, dass Bedingung (4.3) erfüllt ist. Dann gilt die Gleichung*

$$\frac{1}{(s+1)!} \frac{d^{s+1} h(\Phi(t, x))}{dt^{s+1}} \Big|_{t=0} = L_{g_1} h u_s + p_s(x, u_0, u_1, \dots, u_{s-1}) \quad \forall s \in \mathbb{N}_0 \quad (4.6)$$

*mit  $p_s$  aus (3.2) und  $u_s$  aus (3.32).*

Beweis: Die Aussage (4.6) soll per Induktion bewiesen werden. Die ersten drei Induktionsschritte und damit den Induktionsanfang haben wir in den Beweisen zu Theorem 3.3 bzw. Korollar 4.3 bereits durchgeführt. Hierbei ist insbesondere Gleichung (4.5) zu beachten, weil hier erstmals Bedingung (4.3) eingeht. Als nächstes soll der Induktionsschritt von  $s-1$  auf  $s$

durchgeführt werden. Die linke Seite der zu beweisenden Gleichung lässt sich folgendermaßen umformen

$$\begin{aligned}
& \left. \frac{1}{(s+1)!} \frac{d^{s+1}}{dt^{s+1}} h(\phi(t, x)) \right|_{t=0} \tag{4.7} \\
\stackrel{(3.19)}{=} & \frac{1}{(s+1)!} L_{g_0+g_1 u_0}^{s+1} h(x) \\
= & \frac{1}{s+1} L_{g_0+g_1 u_0} \frac{1}{((s-1)+1)!} L_{g_0+g_1 u_0}^s h(x) \\
\stackrel{I.V.}{=} & \frac{1}{s+1} L_{g_0+g_1 u_0} \left[ L_{g_1} h(x) \cdot u_{s-1}(x) + p_{s-1}(x, u_0, \dots, u_{s-2}) \right] \\
= & \frac{1}{s+1} \left\{ L_{g_0} L_{g_1} h(x) u_{s-1}(x) + L_{g_1} L_{g_1} h(x) u_0(x) u_{s-1}(x) \right. \\
+ & \left. L_{g_1} h(x) \underbrace{L_{g_0+g_1 u_0} u_{s-1}(x)}_{=(s+1)u_s(x)} + L_{g_0+g_1 u_0} p_{s-1}(x, u_0, \dots, u_{s-2}) \right\}.
\end{aligned}$$

Lässt man die Argumente der besseren Übersichtlichkeit halber weg, gilt es zu zeigen, dass

$$\frac{L_{g_0} L_{g_1} h \cdot u_{s-1} + L_{g_1} L_{g_1} h \cdot u_0 u_{s-1} + L_{g_0+g_1 u_0} p_{s-1}}{s+1} = p_s \tag{4.8}$$

für  $p_s$  aus (3.2) gilt.

Um Gleichung (4.8) zu zeigen, werden wir folgendermaßen vorgehen: Betrachte die Summanden der äußeren Summe in der Definition von  $p_s$ . Diese sind gegeben durch

$$\sum_{i_0=0, \dots, i_k=0}^1 \frac{L_{g_{i_0}} \cdots L_{g_{i_k}} h(x)}{(k+1)!} \left( \sum_{\substack{|\nu|=|I_k| \\ \|\nu\|=s-k}} \binom{|I_k|}{n_0 \ n_1 \ \dots \ n_M} \prod_{j=0}^{s-1} u_j^{n_j} \right) \tag{4.9}$$

für  $k = 1, \dots, s$  und zeige, dass (4.9) genau aus den Termen der linken Seite von (4.8) besteht, die exakt  $k+1$  auf  $h$  angewendete  $L_{g_i}$  Operatoren haben. Da jeder Term auf der linken Seite von (4.8) wenigstens zwei und maximal  $s+1$   $L_{g_i}$  Operatoren hat, beweist dies (4.8).

Es werden dabei die Fälle  $k = 1$  und  $k = s$  gesondert behandelt, auch weil sie sehr instruktiv für allgemeines  $2 \leq k \leq s-1$  sind. Begonnen wird mit  $k = 1$ :

In diesem Fall wird (4.9) zu

$$\frac{1}{2} (L_{g_0} L_{g_1} h + L_{g_1} L_{g_0} h) u_{s-1} + \frac{1}{2} L_{g_1} L_{g_1} h \left( \sum_{i+j=s-1} u_i u_j \cdot \chi_{ij} \right) \quad \text{mit} \quad \chi_{ij} = \begin{cases} 2 & i \neq j \\ 1 & i = j \end{cases}.$$

Falls  $s - 1$  gerade ist, wird (4.9) folglich zu

$$\frac{1}{2}(L_{g_0}L_{g_1}h + L_{g_1}L_{g_0}h)u_{s-1} + \frac{1}{2}L_{g_1}L_{g_1}h\left(2u_0u_{s-1} + 2\sum_{i=1}^{\frac{s-1}{2}-1}u_iu_{(s-1)-i} + u_{\frac{s-1}{2}}^2\right), \quad (4.10)$$

falls  $s - 1$  jedoch ungerade ist, wird (4.9) zu

$$\frac{1}{2}(L_{g_0}L_{g_1}h + L_{g_1}L_{g_0}h)u_{s-1} + \frac{1}{2}L_{g_1}L_{g_1}h\left(2u_0u_{s-1} + 2\sum_{i=1}^{\frac{s-2}{2}}u_iu_{(s-1)-i}\right). \quad (4.11)$$

Alle Terme in diesen Ausdrücken beinhalten exakt zwei  $L_{g_i}$  Operatoren. Entsprechend betrachtet man nur die Terme der linken Seite in Gleichung (4.8), die ebenfalls genau zwei  $L_{g_i}$  Operatoren haben. Diesbezüglich schreiben wir zuerst die Terme in  $p_{s-1}$  auf, die diese Bedingung erfüllen. Dies sind für (4.10)

$$\frac{1}{2}(L_{g_0}L_{g_1}h + L_{g_1}L_{g_0}h)u_{s-2} + \frac{1}{2}L_{g_1}L_{g_1}h\left(2u_0u_{s-2} + 2\sum_{i=1}^{\frac{s-3}{2}}u_iu_{(s-2)-i}\right) \quad (4.12)$$

und für (4.11)

$$\frac{1}{2}(L_{g_0}L_{g_1}h + L_{g_1}L_{g_0}h)u_{s-2} + \frac{1}{2}L_{g_1}L_{g_1}h\left(2u_0u_{s-2} + 2\sum_{i=1}^{\frac{s-2}{2}-1}u_iu_{(s-2)-i} + u_{\frac{s-2}{2}}^2\right). \quad (4.13)$$

Nun sind wir in der Lage, ausgehend von (4.12) bzw. (4.13), die entsprechenden Terme des Zählers der linken Seite von (4.8) aufzuschreiben. Zu beachten ist hierbei, dass (4.3) in die Umformungen eingeht. Wir werden diese Rechnungen lediglich für (4.12) detailliert durchführen, für (4.13) sind die einzelnen Rechenschritte analog durchführbar. Dabei wird der Übersichtlichkeit halber  $L_{[\dots]}$  an Stelle von  $L_{g_0+g_1u_0}$  verwendet und es ergibt sich

$$\begin{aligned} & L_{g_0}L_{g_1}hu_{s-1} + L_{g_1}^2hu_0u_{s-1} + \frac{1}{2}(L_{g_0}L_{g_1} + L_{g_1}L_{g_0})hL_{[\dots]}u_{s-2} + L_{g_1}^2h\left(\sum_{i=0}^{\frac{s-3}{2}}(L_{[\dots]}u_iu_{(s-2)-i} + u_iL_{[\dots]}u_{(s-2)-i})\right) \\ &= (s+1)L_{g_0}L_{g_1}hu_{s-1} + L_{g_1}^2hu_0u_{s-1} + L_{g_1}L_{g_1}h\left(\sum_{i=0}^{\frac{s-3}{2}}((i+2)u_{i+1}u_{(s-2)-i} + (s-i)u_iu_{(s-1)-i})\right) \\ &= (s+1)L_{g_0}L_{g_1}hu_{s-1} + L_{g_1}^2h\left((s+1)u_0u_{s-1} + \sum_{i=1}^{\frac{s-3}{2}}((i+1)u_iu_{(s-1)-i} + (s-i)u_iu_{(s-1)-i}) + \frac{s+1}{2}u_{\frac{s-1}{2}}^2\right) \\ &= (s+1)L_{g_0}L_{g_1}hu_{s-1} + (s+1)L_{g_1}L_{g_1}h\left(u_0u_{s-1} + \sum_{i=1}^{\frac{s-3}{2}}u_iu_{(s-1)-i} + \frac{1}{2}u_{\frac{s-1}{2}}^2\right). \end{aligned}$$

Beachtet man nun, dass der Faktor  $s + 1$  im Nenner der linken Seite von (4.8) steht und (4.3) folgt die Behauptung für  $k = 1$ .

Wie oben bereits angekündigt, wird nun auch der Fall  $k = s$  separat behandelt. Aus  $k = s$  folgt unmittelbar  $\|\nu\| = 0$  in (4.9), also ist nur  $n_0$  ungleich Null möglich, falls  $|\nu| > 0$ . Folglich wird (4.9) für  $k = s$  unter Berücksichtigung von (4.3) zu

$$\sum_{i=0}^{s+1} \frac{1}{(s+1)!} \binom{s+1}{i} \underbrace{L_{g_0} \dots L_{g_0}}_{i\text{-mal}} \underbrace{L_{g_1} \dots L_{g_1}}_{(s+1-i)\text{-mal}} h \cdot u_0^{s+1-i}, \quad (4.14)$$

wobei der Binomialkoeffizient

$$\binom{s+1}{i} = \frac{(s+1)!}{i!(s+1-i)!}$$

die Anzahl der möglichen Anordnungen der Differentialoperatoren angibt. Nun gilt es aus dem Zähler der linken Seite von (4.8) die Terme rauszufiltern, die ebenfalls genau  $s + 1$   $L_{g_i}$  Operatoren haben. Es gilt folglich aus  $p_{s-1}(x, u_0, \dots, u_{s-2})$  Terme mit  $n$  Operatoren rauszunehmen und auf diese den Operator  $L_{g_0+g_1 u_0}$  anzuwenden. Hierbei ist zu beachten, dass die Ableitungsterme der Kontrollfunktionen weiterhin nur  $n$   $L_{g_i}$  Operatoren beinhalten, also an dieser Stelle nicht berücksichtigt werden müssen. Unter Beachtung von (4.3) ergibt sich

$$(L_{g_0} + L_{g_1} u_0) \left( \sum_{i=0}^s \frac{1}{s!} \binom{s}{i} \underbrace{L_{g_0} \dots L_{g_0}}_{i\text{-mal}} \underbrace{L_{g_1} \dots L_{g_1}}_{(s-i)\text{-mal}} h \cdot u_0^{s-i} \right). \quad (4.15)$$

Um also (4.8) für  $k = s$  zu zeigen, muss für jeden der  $s + 2$  Summanden aus (4.14) gezeigt werden, dass dieser Term multipliziert mit  $s + 1$  exakt den Termen aus (4.15) mit genauso vielen  $L_{g_0}$  und  $L_{g_1}$  Operatoren entspricht, also

$$\begin{aligned} \binom{s+1}{i} \underbrace{L_{g_0} \dots L_{g_0}}_{i\text{-mal}} \underbrace{L_{g_1} \dots L_{g_1}}_{(s+1-i)\text{-mal}} h \cdot u_0^{s+1-i} &\stackrel{!}{=} \left( \underbrace{L_{g_0} L_{g_0} \dots L_{g_0}}_{(i-1)\text{-mal}} \underbrace{L_{g_1} \dots L_{g_1}}_{(s+1-i)\text{-mal}} h u_0^{s+1-i} \cdot \binom{s}{i-1} \right) + \\ &+ \left( L_{g_1} \underbrace{L_{g_0} \dots L_{g_0}}_{i\text{-mal}} \underbrace{L_{g_1} \dots L_{g_1}}_{(s-i)\text{-mal}} h u_0^{s-i} \cdot \binom{s}{i} \cdot u_0 \right). \end{aligned}$$

Hieraus folgt die Behauptung, weil

$$\binom{s}{i-1} + \binom{s}{i} = \binom{s+1}{i}$$

und nach [4]

$$\binom{s}{-1} = 0 \quad \text{und} \quad \binom{s}{s+1} = 0$$

gilt.

Um den Beweis zu vervollständigen, muss noch Gleichung (4.8) für  $2 \leq k \leq s-1$  gezeigt werden. Mit Hilfe von (4.3) kann man die Summanden in (4.9) wie folgt umschreiben

$$\frac{1}{(k+1)!} \binom{k+1}{i} \underbrace{L_{g_0} \dots L_{g_0}}_{i\text{-mal}} \underbrace{L_{g_1} \dots L_{g_1}}_{(k+1-i)\text{-mal}} h \cdot \left( \sum_{\substack{|\nu|=k+1-i \\ \|\nu\|=s-k}} \binom{k+1-i}{n_0 \dots n_M} \prod_{j=0}^{s-1} u_j^{n_j} \right), \quad (4.16)$$

dabei nimmt  $k+1-i = |I_k|$  Werte im Bereich von  $0, \dots, k+1$  an. Der Ausdruck (4.16) enthält genau  $i$   $L_{g_0}$  Operatoren und  $k+1-i$   $L_{g_1}$  Operatoren. Betrachtet man nun die linke Seite von (4.8), wieder unter Verwendung von (4.3), sind die Terme, die jeweils exakt diese Anzahl an  $L_{g_0}$  und  $L_{g_1}$  Operatoren haben

$$\begin{aligned} & \frac{1}{s+1} \left[ L_{g_0} \left( \frac{1}{k!} \binom{k}{i-1} \underbrace{L_{g_0} \dots L_{g_0}}_{(i-1)\text{-mal}} \underbrace{L_{g_1} \dots L_{g_1}}_{(k+1-i)\text{-mal}} h \left( \sum_{\substack{|\nu|=k+1-i \\ \|\nu\|=s-k}} \binom{k+1-i}{n_0 \dots n_M} \prod_{j=0}^{s-1} u_j^{n_j} \right) \right) \right. \\ & + L_{g_1} \left( \frac{1}{k!} \binom{k}{i} \underbrace{L_{g_0} \dots L_{g_0}}_{i\text{-mal}} \underbrace{L_{g_1} \dots L_{g_1}}_{(k-i)\text{-mal}} h \left( \sum_{\substack{|\nu|=k-i \\ \|\nu\|=s-k}} \binom{k-i}{n_0 \dots n_M} \prod_{j=0}^{s-1} u_j^{n_j} \right) \right) u_0 \\ & \left. + \frac{1}{(k+1)!} \binom{k+1}{i} \underbrace{L_{g_0} \dots L_{g_0}}_{i\text{-mal}} \underbrace{L_{g_1} \dots L_{g_1}}_{(k+1-i)\text{-mal}} h \left( \sum_{\substack{|\nu|=k+1-i \\ \|\nu\|=s-(k+1)}} \binom{k+1-i}{n_0 \dots n_M} L_{g_0+g_1 u_0} \prod_{j=0}^{s-1} u_j^{n_j} \right) \right]. \quad (4.17) \end{aligned}$$

Entsprechend muss gezeigt werden, dass die Ausdrücke (4.16) und (4.17) für  $i = 0, \dots, k+1$  übereinstimmen. Um dieses Ziel zu erreichen fixiere man einen Summanden in (4.16), d.h. insbesondere einen Multiindex  $(n_0, \dots, n_M)$ , und betrachte alle Summanden in (4.17), die das Produkt aus Kontrollfunktionen  $\prod_{j=0}^{s-1} u_j^{n_j}$  enthalten. Nachdem man gezeigt hat, dass diese Summanden übereinstimmen, folgt die Gleichheit der Ausdrücke (4.16) und (4.17), weil man einfach nachprüfen kann, dass (4.17) keine Produkte von Kontrollfunktionen enthält, die nicht auch in (4.16) auftauchen.

Um die passenden Terme in (4.17) zu finden, muss man die Indices  $\nu$  identifizieren, für die das Produkt von Kontrollfunktionen in den drei Termen in (4.17)  $\prod_{j=0}^{s-1} u_j^{n_j}$  entspricht. Für den ersten Term in (4.17) bedeutet dies lediglich  $\nu = (n_0, \dots, n_M)$  zu setzen und im

zweiten Term erhält man die gewünschte Kombination von Kontrollfunktionen, indem man  $\nu = (n_0 - 1, n_1, \dots, n_M)$  wählt. Vorausgesetzt  $n_0 \geq 1$ , weil ansonsten dieser Term das gesuchte Produkt nicht enthält. Der letzte Term in (4.17) ist der am mühsamsten zu behandelnde. In diesem lässt sich die im letzten Ausdruck auftauchende Ableitung von  $\prod_{j=0}^{s-1} u_j^{n_j}$  mit Hilfe der Definition von  $u_j$  in (3.32) zu

$$L_{g_0+g_1u_0} \prod_{j=0}^{s-1} u_j^{n_j} = \sum_{l=0}^{s-2} n_l u_l^{n_l-1} (l+2) u_{l+1} \prod_{\substack{j=0 \\ j \neq l}}^{s-1} u_j^{n_j} \quad (4.18)$$

auswerten, wobei es ausreichend ist, die Summe von  $l$  bis  $s-2$  zu nehmen, weil  $\|\nu\| = s - (k+1)$  und  $k \geq 2$  implizieren, dass  $n_{s-1} = 0$ . Dieses Argument ließe sich sogar für  $k = 1$  aufrechterhalten, wir haben diesen Fall also lediglich des besseren Verständnisses wegen gesondert behandelt. Entsprechend muss man im dritten Term die Multiindices  $\nu = (n_0, \dots, n_{l-1}, n_l + 1, n_{l+1} - 1, n_{l+1}, \dots, n_M)$  für alle  $l = 0, \dots, s-2$  mit  $n_{l+1} \geq 1$  nehmen (falls  $n_{l+1} = 0$  enthält der dritte Term dieses Produkt nicht), um das Produkt  $\prod_{j=0}^{s-1} u_j^{n_j}$  zu erhalten.

Beachtet man dies, ist die Gleichheit von (4.16) und (4.17) äquivalent zu

$$\begin{aligned} & \frac{1}{(k+1)!} \binom{k+1}{i} \binom{k+1-i}{n_0 \dots n_M} \cdot (s+1) \\ &= \frac{1}{k!} \binom{k}{i-1} \binom{k+1-i}{n_0 \dots n_M} \\ &+ \frac{1}{k!} \binom{k}{i} \binom{k-i}{n_0-1 \ n_1 \dots n_M} \\ &+ \frac{1}{(k+1)!} \binom{k+1}{i} \sum_{l=0}^{s-2} \binom{k+1-i}{n_0 \dots n_{l-1} \ n_l+1 \ n_{l+1}-1 \ n_{l+2} \dots n_M} (l+2)(n_l+1) \end{aligned} \quad (4.19)$$

mit der Konvention, dass  $\binom{n}{n_0 \ n_1 \dots n_M} = 0$ , falls eines der  $n_i$  negativ ist. Gleichung (4.19) ist äquivalent zu

$$\begin{aligned} \frac{s+1}{i!} \cdot \frac{1}{n_0! \dots n_M!} &= \frac{i}{i!} \cdot \frac{1}{n_0! \dots n_M!} + \frac{1}{i!} \frac{n_0}{n_0! \dots n_M!} \\ &+ \frac{1}{i!} \frac{1}{n_0! \dots n_M!} \sum_{l=0}^{s-2} \frac{(l+2) \cdot n_{l+1} \cdot (n_l+1)}{n_l+1} \\ &= \frac{1}{i! n_0! \dots n_M!} \left( i + n_0 + \sum_{l=0}^{s-2} (l+2) n_{l+1} \right). \end{aligned} \quad (4.20)$$

Beachte, dass die Gleichungen (4.19) und (4.20) sogar equivalent sind, falls  $n_0 = 0$  oder  $n_{l+1} = 0$  für gewisse  $l = 0, \dots, s-2$ , weil in diesem Fall die korrespondierenden Summanden in (4.19) wegen unserer Konvention, ebenso wie die entsprechenden Summanden in (4.20), verschwinden. Nun folgt (4.20) aus

$$\begin{aligned} i + n_0 + \sum_{l=0}^{s-2} (l+2)n_{l+1} &= i + n_0 + \sum_{l=1}^{s-1} (l+1)n_l \\ &= i + \underbrace{\sum_{l=0}^{s-1} n_l}_{=|\nu|=k+1-i} + \underbrace{\sum_{l=0}^{s-1} l n_l}_{=|\nu|=s-k} = s+1. \end{aligned}$$

Dies zeigt, dass sich (4.16) und (4.17) für  $k = 2, \dots, s-1$  entsprechen. Tatsächlich beinhaltet dieser Beweisteil auch den Fall  $k = s$ . Folglich gilt (4.8) und damit die Aussage des Lemmas.  $\square$

Ausgehend von Lemma 4.4 kann man analog zu [11] folgendes Theorem beweisen.

**Theorem 4.5** *Betrachte das System (2.16), das kontinuierliche closed-loop System (2.18) und das sampled-data closed-loop System (2.32) mit Kontrollfunktion  $u_T^M$  gegeben durch (3.31) für ein  $M \in \mathbb{N}$ . Falls Bedingung (4.3) gilt, d.h. die Vektorfelder  $g_0$  und  $g_1$  kommutieren, gilt*

$$\Delta h(T, x, u_T^M) = \mathcal{O}(T^{M+2}) \quad (4.21)$$

für jede hinreichend glatte Funktion  $h : \mathbb{R}^n \rightarrow \mathbb{R}$  und als Folge analog Bemerkung 2.29

$$\Delta \Phi(T, x, u_T^M) = \mathcal{O}(T^{M+2}). \quad (4.22)$$

*Beweis:* Der Beweis dieses Theorems beruht auf dem technischen Lemma 4.4. Dieses Lemma wird helfen die Taylorentwicklung in  $t = 0$  von  $h(\Phi(T, x))$  umzuformen:

$$\begin{aligned} h(\phi(T, x)) &= h(x) + \sum_{i=1}^{M+1} \frac{T^i}{i!} \frac{d^i}{dt^i} h(\phi(t, x)) \Big|_{t=0} + \mathcal{O}(T^{M+2}) \\ &= h(x) + \sum_{i=0}^M \frac{T^{i+1}}{(i+1)!} \frac{d^{i+1}}{dt^{i+1}} h(\phi(t, x)) \Big|_{t=0} + \mathcal{O}(T^{M+2}) \\ &= h(x) + \sum_{i=0}^M T^{i+1} [L_{g_1} h(x) u_i(x) + p_i(x, u_0, u_1, \dots, u_{i-1})] + \mathcal{O}(T^{M+2}). \quad (4.23) \end{aligned}$$



Andererseits erhalten wir aus (3.1) durch Multiplikation mit  $T$  und Addition von  $h(x)$

$$h(\phi_T(T, x, u_T)) = h(x) + \sum_{s=0}^M T^{s+1} [L_{g_1} h(x) u_s + p_s(x, u_0, \dots, u_{s-1})] + \mathcal{O}(T^{M+2}). \quad (4.24)$$

Vergleicht man nun (4.23) und (4.24) erhält man die Behauptung.  $\square$

**Bemerkung 4.6** *Der obige Beweis zeigt außerdem, dass unter Bedingung (4.3) die Feedback-Regelung  $u_T^M$  aus (3.31) bis auf Terme der Ordnung  $\mathcal{O}(T^{M+1})$  die einzige Feedback-Regelung ist, für die (4.21) gilt, solange nicht  $g_1(x) = 0$ . Um dies einzusehen, betrachte man eine beliebige Feedback-Regelung  $\tilde{u}_T(x)$ . Vergleicht man (4.23) und (4.24) für  $M = 0$  unter der Voraussetzung, dass  $g_1(x) \neq 0$ , sieht man, dass  $\tilde{u}_T(x)$  von der Form  $\tilde{u}_T(x) = u_0(x) + T\tilde{u}_1(x)$  sein muss, um (4.21) mit  $M = 0$  zu erfüllen. Wiederholt man dieses Argument induktiv, kann man für jedes  $M \in \mathbb{N}$  zeigen, dass die Feedback-Regelung  $\tilde{u}_T(x)$  von der Form*

$$\tilde{u}_T(x) = u_T^M(x) + \mathcal{O}(T^{M+1})$$

*sein muss, um (4.21) und damit auch (4.22) zu erfüllen.*

## 4.2 Abschwächung für endliche Ordnung

Im letzten Abschnitt wurde gezeigt, dass man unter Bedingung (4.3) das System (2.16) und das zeitkontinuierliche closed-loop System (2.18) mit dem sampled-data System (2.32) beliebig gut approximieren kann, wenn man die Kontrollfunktion analog (3.31) definiert. Allerdings ist es sehr einschränkend (4.3) grundsätzlich vorauszusetzen. In diesem Abschnitt werden die hinreichenden Bedingungen aus Korollar 3.5 genauer untersucht, bevor im nächsten Kapitel notwendige Bedingungen für eine Approximation gesucht werden.

Für eine Approximation vierter Ordnung muss nach Korollar 3.5 Bedingung (3.45) erfüllt sein. Dies ist genau dann der Fall, wenn

$$L_{g_0} L_{g_1} h \stackrel{!}{=} L_{g_1} L_{g_0} h \quad (4.25)$$

gilt oder  $u_1 = 0$  ist. Dabei ist  $u_1$  analog (3.32) definiert. Die zweite Gleichung ist also erfüllt, wenn  $u_0(\Phi(t, x))$  konstant ist, was im Allgemeinen nicht erfüllt sein dürfte. Die erste Gleichung ist gleichbedeutend mit

$$\frac{\partial}{\partial x} h \left[ \frac{\partial}{\partial x} g_1 \cdot g_0 - \frac{\partial}{\partial x} g_0 \cdot g_1 \right] = \frac{\partial}{\partial x} h[g_0, g_1] \stackrel{!}{=} 0. \quad (4.26)$$

Wie man an dieser Gleichung erkennen kann, wird für  $M = 2$  Bedingung (4.3) nicht global, sondern lediglich entlang  $\frac{\partial}{\partial x}h$  gefordert.

Es bleibt die Frage zu beantworten, inwiefern (3.47) aus Korollar 3.5 eine Verschärfung gegenüber (4.26) darstellt. Hierbei wird der Fall, dass  $u_i$ ,  $i = 0, 1, 2$  gleich Null ist, weglassen. Dies ist zwar ein wichtiger Sonderfall, allerdings sind wir in diesem Abschnitt daran interessiert, Aussagen darüber herzuleiten, unter welchen Voraussetzungen man (4.3) für endliche Ordnungen abschwächen kann. Konkret suchen wir Bedingungen, unter denen eine endliche Anzahl der  $L_{g_i}$  Differentialoperatoren kommutiert. Beginnen wollen wir mit dem Ausdruck

$$L_{g_0}L_{g_1}L_{g_1}h \stackrel{!}{=} L_{g_1}L_{g_1}L_{g_0}h \quad (4.27)$$

aus Bedingung (3.47). Hierbei ist zu beachten, dass bereits nach (4.25)  $L_{g_1}L_{g_1}L_{g_0}h$  gleich  $L_{g_1}L_{g_0}L_{g_1}h$  ist. Entsprechend reicht es aus,  $L_{g_0}L_{g_1}L_{g_1}h$  mit  $L_{g_1}L_{g_0}L_{g_1}h$  zu vergleichen. Zu diesem Zweck berechnen wir zuerst beide Ausdrücke separat. Das ergibt

$$\begin{aligned} L_{g_0}L_{g_1}L_{g_1}h &= L_{g_0}L_{g_1} \underbrace{\left( \frac{\partial}{\partial x_1}h, \dots, \frac{\partial}{\partial x_n}h \right)}_{=\sum_{i=1}^n \frac{\partial h}{\partial x_i} g_{1i}} \cdot \begin{pmatrix} g_{11} \\ \vdots \\ g_{1n} \end{pmatrix} \\ &= L_{g_0} \sum_{j=1}^n \left[ \frac{\partial}{\partial x_j} \left( \sum_{i=1}^n \frac{\partial h}{\partial x_i} \cdot g_{1i} \right) \cdot g_{1j} \right] \\ &\quad \sum_{i=1, j=1}^n \left( \left[ \frac{\partial^2 h}{\partial x_i \partial x_j} \cdot g_{1i} + \frac{\partial h}{\partial x_i} \cdot \frac{\partial}{\partial x_j} g_{1i} \right] g_{1j} \right) \\ &= \sum_{k=1}^n \frac{\partial}{\partial x_k} \sum_{i=1, j=1}^n \left( \left[ \frac{\partial^2 h}{\partial x_i \partial x_j} \cdot g_{1i} + \frac{\partial h}{\partial x_i} \cdot \frac{\partial}{\partial x_j} g_{1i} \right] g_{1j} \right) \cdot g_{0k} \\ &= \sum_{i=1, j=1, k=1}^n \left( \frac{\partial^3 h}{\partial x_i \partial x_j \partial x_k} \cdot g_{1i} g_{1j} g_{0k} + \frac{\partial^2 h}{\partial x_i \partial x_j} \frac{\partial}{\partial x_k} g_{1i} g_{1j} g_{0k} \right. \\ &\quad \left. + \frac{\partial^2 h}{\partial x_i \partial x_j} g_{1i} \frac{\partial}{\partial x_k} g_{1j} g_{0k} + \frac{\partial^2 h}{\partial x_i \partial x_k} \frac{\partial}{\partial x_j} g_{1i} g_{1j} g_{0k} \right. \\ &\quad \left. + \frac{\partial}{\partial x_i} h \frac{\partial^2}{\partial x_j \partial x_k} g_{1i} g_{1j} g_{0k} + \frac{\partial}{\partial x_i} h \frac{\partial}{\partial x_j} g_{1i} \frac{\partial}{\partial x_k} g_{1j} g_{0k} \right) \end{aligned}$$

und

$$\begin{aligned}
L_{g_1} L_{g_0} L_{g_1} h &= L_{g_0} \sum_{i=1, k=1}^n \left( \left[ \frac{\partial^2 h}{\partial x_i \partial x_k} \cdot g_{1i} + \frac{\partial h}{\partial x_i} \cdot \frac{\partial}{\partial x_k} g_{1i} \right] g_{0k} \right) \\
&= \sum_{i=1, j=1, k=1}^n \left( \frac{\partial^3 h}{\partial x_i \partial x_j \partial x_k} \cdot g_{1i} g_{0k} g_{1j} + \frac{\partial^2 h}{\partial x_i \partial x_k} \frac{\partial}{\partial x_j} g_{1i} g_{0k} g_{1j} \right. \\
&\quad + \frac{\partial^2 h}{\partial x_i \partial x_k} g_{1i} \frac{\partial}{\partial x_j} g_{0k} g_{1j} + \frac{\partial^2 h}{\partial x_i \partial x_j} \frac{\partial}{\partial x_k} g_{1i} g_{0k} g_{1j} \\
&\quad \left. + \frac{\partial}{\partial x_i} h \frac{\partial^2}{\partial x_k \partial x_j} g_{1i} g_{0k} g_{1j} + \frac{\partial}{\partial x_i} h \frac{\partial}{\partial x_k} g_{1i} \frac{\partial}{\partial x_j} g_{0k} g_{1j} \right).
\end{aligned}$$

Daraus folgt, dass für die Differenz

$$\begin{aligned}
L_{g_0} L_{g_1} L_{g_1} h - L_{g_1} L_{g_0} L_{g_1} h &= \sum_{i=1, j=1, k=1}^n \frac{\partial^2 h}{\partial x_i \partial x_j} g_{1i} \left( \frac{\partial}{\partial x_k} g_{1j} \cdot g_{0k} - \frac{\partial}{\partial x_k} g_{0j} \cdot g_{1k} \right) \\
&\quad + \sum_{i=1, j=1, k=1}^n \frac{\partial h}{\partial x_i} \frac{\partial}{\partial x_j} g_{1i} \left( \frac{\partial}{\partial x_k} g_{1j} \cdot g_{0k} - \frac{\partial}{\partial x_k} g_{0j} \cdot g_{1k} \right) \quad (4.28)
\end{aligned}$$

und unter Berücksichtigung von (4.25) und Bemerkung 3.4

$$\begin{aligned}
L_{g_0} L_{g_1} L_{g_1} h - L_{g_1} L_{g_0} L_{g_1} h &= \sum_{i=1, j=1, k=1}^n \frac{\partial^2 h}{\partial x_i \partial x_j} g_{1i} \left( \frac{\partial}{\partial x_k} g_{1j} \cdot g_{0k} - \frac{\partial}{\partial x_k} g_{0j} \cdot g_{1k} \right) \quad (4.29) \\
&\quad + \sum_{i=1, j=1, k=1, i \neq j}^n \frac{\partial h}{\partial x_i} \frac{\partial}{\partial x_j} g_{1i} \left( \frac{\partial}{\partial x_k} g_{1j} \cdot g_{0k} - \frac{\partial}{\partial x_k} g_{0j} \cdot g_{1k} \right)
\end{aligned}$$

gilt. Es fallen also wegen (4.25) die Terme für  $i = j$  weg. Deutlich anders verhält es sich im eindimensionalen Fall ( $n = 1$ ). In diesem fällt der zweite Summand komplett weg und es folgt

$$\begin{aligned}
L_{g_0} L_{g_1} L_{g_1} h - L_{g_1} L_{g_0} L_{g_1} h &= \frac{\partial^2}{\partial x^2} h \left[ \frac{\partial}{\partial x} g_{1g_0} - g_1 \frac{\partial}{\partial x} g_0 \right] g_1 + \underbrace{\frac{\partial}{\partial x} h \left[ \frac{\partial}{\partial x} g_{1g_0} - \frac{\partial}{\partial x} g_0 g_1 \right]}_{=0 \text{ nach (4.26)}} \frac{\partial}{\partial x} g_1 \\
&= \frac{\partial^2}{\partial x^2} h \left[ \frac{\partial}{\partial x} g_1 \cdot g_0 - \frac{\partial}{\partial x} g_0 \cdot g_1 \right] g_1.
\end{aligned}$$

Es muss also für (4.27) lediglich

$$\frac{\partial^2}{\partial x^2} h \left[ \frac{\partial}{\partial x} g_1 \cdot g_0 - \frac{\partial}{\partial x} g_0 \cdot g_1 \right] g_1 = \frac{\partial^2}{\partial x^2} h [g_0, g_1] g_1 = 0 \quad (4.30)$$

und aus Symmetriegründen auch

$$\frac{\partial^2}{\partial x^2} h \left[ \frac{\partial}{\partial x} g_1 \cdot g_0 - \frac{\partial}{\partial x} g_0 \cdot g_1 \right] g_0 = \frac{\partial^2}{\partial x^2} h [g_0, g_1] g_0 = 0. \quad (4.31)$$

zusätzlich zu (4.26) gefordert werden. Im restlichen Verlauf dieses Abschnitts beschränken wir uns auf den Fall  $n = 1$ . Der höher dimensionale Fall wird dagegen, wie bereits (4.29) zeigt, für  $M \geq 3$  sehr technisch. Festzuhalten bleibt, dass für (4.27) Gleichung (4.26) allein nicht ausreicht. Stattdessen müssen zusätzlich die Gleichungen (4.30) und (4.31) gefordert werden.

Analog zu Bedingung (3.47) wird (3.48) betrachtet. Besonderes Interesse verdient die Frage, welche Form potentiell neu hinzukommende Terme haben. Beginnen wollen wir mit  $L_{g_1} L_{g_0} L_{g_1} L_{g_1} h$  und  $L_{g_0} L_{g_1} L_{g_1} L_{g_1} h$ , weil dies nach den bisherigen Betrachtungen auch hinreichende Bedingungen für  $L_{g_1} L_{g_1} L_{g_1} L_{g_0} h$  und  $L_{g_0} L_{g_1} L_{g_1} L_{g_1} h$  liefert.

$$\begin{aligned} L_{g_1} L_{g_0} L_{g_1} L_{g_1} h &= \frac{\partial}{\partial x} \left\{ \frac{\partial^3}{\partial x^3} h g_1^2 g_0 + 3 \frac{\partial^2}{\partial x^2} h \frac{\partial}{\partial x} g_1 g_1 g_0 + \frac{\partial}{\partial x} h \left[ \frac{\partial^2}{\partial x^2} g_1 g_1 + \left( \frac{\partial}{\partial x} g_1 \right)^2 \right] g_0 \right\} g_1 \quad (4.32) \\ &= \frac{\partial^4}{\partial x^4} h g_1^3 \cdot g_0 \\ &+ \frac{\partial^3}{\partial x^3} h \left[ 5 \frac{\partial}{\partial x} g_1 \cdot g_1^2 g_0 + \frac{\partial}{\partial x} g_0 \cdot g_1^3 \right] + \\ &+ \frac{\partial^2}{\partial x^2} h \left[ 4 \frac{\partial^2}{\partial x^2} g_1 \cdot g_1^2 g_0 + 4 \left( \frac{\partial}{\partial x} g_1 \right)^2 \cdot g_1 g_0 + 3 \frac{\partial}{\partial x} g_1 \frac{\partial}{\partial x} g_0 \cdot g_1^2 \right] + \\ &+ \frac{\partial}{\partial x} h \left[ \frac{\partial^3}{\partial x^3} g_1 \cdot g_1^2 g_0 + \frac{\partial^2}{\partial x^2} g_1 \frac{\partial}{\partial x} g_0 \cdot g_1^2 + \left( \frac{\partial}{\partial x} g_1 \right)^2 \frac{\partial}{\partial x} g_0 \cdot g_1 + 3 \frac{\partial^2}{\partial x^2} g_1 \frac{\partial}{\partial x} g_1 \cdot g_1 g_0 \right] \end{aligned}$$

$$\begin{aligned} L_{g_0} L_{g_1} L_{g_1} L_{g_1} h &= \frac{\partial}{\partial x} \left\{ \frac{\partial^3}{\partial x^3} h g_1^3 + 3 \frac{\partial^2}{\partial x^2} h \frac{\partial}{\partial x} g_1 g_1^2 + \frac{\partial}{\partial x} h \left[ \frac{\partial^2}{\partial x^2} g_1 g_1 + \left( \frac{\partial}{\partial x} g_1 \right)^2 \right] g_1 \right\} g_0 \\ &= \frac{\partial^4}{\partial x^4} h g_1^3 g_0 \\ &+ \frac{\partial^3}{\partial x^3} h \left[ 6 \frac{\partial}{\partial x} g_1 \cdot g_1^2 g_0 \right] + \\ &+ \frac{\partial^2}{\partial x^2} h \left[ 4 \frac{\partial^2}{\partial x^2} g_1 \cdot g_1^2 g_0 + 7 \left( \frac{\partial}{\partial x} g_1 \right)^2 \cdot g_1 g_0 \right] + \\ &+ \frac{\partial}{\partial x} h \left[ \frac{\partial^3}{\partial x^3} g_1 \cdot g_1^2 g_0 + 4 \frac{\partial^2}{\partial x^2} g_1 \frac{\partial}{\partial x} g_1 \cdot g_1 g_0 + \left( \frac{\partial}{\partial x} g_1 \right)^3 \cdot g_0 \right] \quad (4.33) \end{aligned}$$

Betrachtet man nun die Differenz  $L_{g_0}L_{g_1}L_{g_1}L_{g_1}h - L_{g_1}L_{g_0}L_{g_1}L_{g_1}h$ , erhält man

$$\begin{aligned}
L_{g_0}L_{g_1}L_{g_1}L_{g_1}h - L_{g_1}L_{g_0}L_{g_1}L_{g_1}h &= \frac{\partial^3}{\partial x^3}h \left[ \frac{\partial}{\partial x}g_1 \cdot g_0 - \frac{\partial}{\partial x}g_0 \cdot g_1 \right] \cdot g_1^2 \\
&+ \underbrace{3 \frac{\partial^2}{\partial x^2}h \left[ \frac{\partial}{\partial x}g_1 \cdot g_0 - \frac{\partial}{\partial x}g_0 \cdot g_1 \right] g_1 \cdot \frac{\partial}{\partial x}g_1}_{= 0 \text{ nach (4.30)}} \\
&+ \frac{\partial}{\partial x}h \left\{ \underbrace{\left[ \frac{\partial}{\partial x}g_1 \cdot g_0 - \frac{\partial}{\partial x}g_0 \cdot g_1 \right]}_{= 0 \text{ nach (4.26)}} \left[ \left( \frac{\partial}{\partial x}g_1 \right)^2 + \frac{\partial^2}{\partial x^2}g_1 \cdot g_1 \right] \right\}.
\end{aligned} \tag{4.34}$$

Folglich muss man für die Gleichheit fordern, dass

$$\frac{\partial^3}{\partial x^3}h \left[ \frac{\partial}{\partial x}g_0 \cdot g_1 - \frac{\partial}{\partial x}g_1 \cdot g_0 \right] \cdot g_1^2 = 0 \tag{4.35}$$

und aus Symmetriegründen wegen  $L_{g_1}L_{g_0}L_{g_0}L_{g_0} - L_{g_0}L_{g_1}L_{g_0}L_{g_0}$  auch, dass

$$\frac{\partial^3}{\partial x^3}h \left[ \frac{\partial}{\partial x}g_0 \cdot g_1 - \frac{\partial}{\partial x}g_1 \cdot g_0 \right] \cdot g_0^2 = 0. \tag{4.36}$$

Wie man bereits an der Herleitung der Bedingungen (4.35) und (4.36) erkennen kann, wächst der Rechenaufwand für Bedingungen bzgl. höherer Ordnungen stark an. Deshalb wird für die Berechnungen von  $L_{g_0}L_{g_1}L_{g_1}L_{g_0}h - L_{g_1}L_{g_0}L_{g_1}L_{g_0}h$  und den entsprechenden Termen für  $M = 5$  eine kurze Maple-Routine eingesetzt. Anfangs werden die beiden Operatoren  $L_{g_0}$  und  $L_{g_1}$  definiert und anschließend die bereits bekannten Ausdrücke berechnet, um sie zur Vereinfachung der zu analysierenden Terme einsetzen zu können.

```

restart; alias(h=h(x)):
alias(g[0]=g[0](x)): alias(g[1]=g[1](x)):

Lg1 := proc(h::algebraic)
  g[1]*diff(h,x); end:
Lg0 := proc(h::algebraic)
  g[0]*diff(h,x); end:

b1:=simplify(Lg0(Lg1(h))-Lg1(Lg0(h)))=0:

Lg0(Lg1(Lg1(h)))-Lg1(Lg0(Lg1(h))):
b2:=factor(simplify(%,{b1}))=0:

```

```

Lg1(Lg0(Lg0(h)))-Lg0(Lg1(Lg0(h))):
b3:=factor(simplify(%,{b1}))=0:

expand(Lg1(Lg0(Lg1(Lg0(h))))): expand(Lg0(Lg1(Lg1(Lg0(h))))):
Lg0(Lg1(Lg1(Lg0(h)))-Lg1(Lg0(Lg1(Lg0(h))))):
b4:=factor(simplify(%,{b1,b2,b3}))=0;
Lg0(Lg1(Lg1(Lg1(h)))-Lg1(Lg0(Lg1(Lg1(h))))):
b5:=factor(simplify(%,{b1,b2,b3}))=0;
Lg0(Lg1(Lg0(Lg0(h)))-Lg1(Lg0(Lg0(Lg0(h))))):
b6:=factor(simplify(%,{b1,b2,b3}))=0:

Lg0(Lg1(Lg1(Lg1(Lg1(h)))))-Lg1(Lg0(Lg1(Lg1(Lg1(h))))):
b7:=factor(simplify(%,{b1,b2,b3,b4,b5,b6}))=0;
Lg0(Lg1(Lg0(Lg0(Lg0(h)))))-Lg1(Lg0(Lg0(Lg0(Lg0(h))))):
b8:=factor(simplify(%,{b1,b2,b3,b4,b5,b6}))=0;
Lg0(Lg1(Lg1(Lg0(Lg1(h)))))-Lg1(Lg0(Lg1(Lg0(Lg1(h))))):
b9:=factor(simplify(%,{b1,b2,b3,b4,b5,b6}))=0;
Lg0(Lg1(Lg0(Lg1(Lg0(h)))))-Lg1(Lg0(Lg0(Lg1(Lg0(h))))):
b10:=factor(simplify(%,{b1,b2,b3,b4,b5,b6}))=0;

```

Es ergibt sich als dritte Bedingung für  $M = 4$

$$\frac{\partial^3}{\partial x^3} h \left[ \frac{\partial}{\partial x} g_1 \cdot g_0 - \frac{\partial}{\partial x} g_0 \cdot g_1 \right] \cdot g_0 g_1 = 0. \quad (4.37)$$

Die Bedingungen (4.35) bis (4.37) sind für  $M = 4$  unter der Voraussetzung, dass (4.26), (4.30) und (4.31) erfüllt sind, hinreichend. Für  $M = 5$  ergeben sich vier weitere Bedingungen. Die Berechnungen erfolgen erneut mit Hilfe der Maple-Routine und liefern

$$\begin{aligned} \frac{\partial^4}{\partial x^4} h \left[ \frac{\partial}{\partial x} g_1 \cdot g_0 - \frac{\partial}{\partial x} g_0 \cdot g_1 \right] \cdot g_1^3 &= 0, \\ \frac{\partial^4}{\partial x^4} h \left[ \frac{\partial}{\partial x} g_1 \cdot g_0 - \frac{\partial}{\partial x} g_0 \cdot g_1 \right] \cdot g_0^3 &= 0, \\ \frac{\partial^4}{\partial x^4} h \left[ \frac{\partial}{\partial x} g_1 \cdot g_0 - \frac{\partial}{\partial x} g_0 \cdot g_1 \right] \cdot g_0 g_1^2 &= 0, \\ \frac{\partial^4}{\partial x^4} h \left[ \frac{\partial}{\partial x} g_1 \cdot g_0 - \frac{\partial}{\partial x} g_0 \cdot g_1 \right] \cdot g_0^2 g_1 &= 0. \end{aligned}$$

Dabei ist auffällig, dass jeweils  $M - 1$  zusätzliche Bedingungen für eine höhere Ordnung  $M$  hinzukommen und der erste Faktor jeweils  $\frac{\partial^{M-1}}{\partial x^{M-1}} h[g_0, g_1]$  ist. Der zweite Faktor besteht jeweils aus einem Produkt der Funktionen  $g_0$  und  $g_1$ , wobei immer genau  $M - 2$  dieser

Funktionen verwendet werden. Es gilt nun ausgehend von den Beobachtungen für Terme bis zur siebten Ordnung ein allgemeines Theorem zu beweisen. Dieses klärt die Frage, inwieweit man die Voraussetzung (4.3) von Lemma 4.4 und Theorem 4.5 für beliebiges, aber festes  $M$  im eindimensionalen Fall abschwächen kann. Denn für den Beweis des Lemmas und damit auch für den Beweis des Theorems wurde lediglich benötigt, dass die Operatoren  $L_{g_0}$  und  $L_{g_1}$  kommutieren.

**Theorem 4.7** *Sei  $M \in \mathbb{N}_{\geq 2}$  beliebig, aber fest und sei  $n = 1$ . Seien  $L_{g_{i_1}}, \dots, L_{g_{i_M}}$  Differentialoperatoren analog Definition 2.6 mit  $i_j \in \{0, 1\}$  für  $j = 1, \dots, M$ . Gilt darüberhinaus*

$$\frac{\partial^{k-1}}{\partial x^{k-1}} h[g_0, g_1] \cdot g_0^n g_1^m = 0 \quad (4.38)$$

für alle  $m, n \in \mathbb{N}$  mit  $m + n = k - 2$  und  $m = 0, \dots, k - 2$  für alle  $k = 2, \dots, M$ . Dann gilt

$$L_{g_{i_1}} L_{g_{i_2}} \cdots L_{g_{i_M}} h = L_{g_{j_1}} L_{g_{j_2}} \cdots L_{g_{j_M}} h, \quad (4.39)$$

falls  $\sum_{h=1}^M i_h = \sum_{h=1}^M j_h$ .

*Beweis:* Wir zeigen die Aussage per Induktion nach  $M$ . Für  $M = 2$  vereinfacht sich (4.38) zu  $\frac{\partial}{\partial x} h[g_0, g_1] = 0$  und die Aussage des Theorems ist im Wesentlichen  $L_{g_0} L_{g_1} h - L_{g_1} L_{g_0} h = 0$ , weil alle anderen Kombinationen von den beiden Operatoren  $L_{g_0}$  und  $L_{g_1}$  trivialerweise kommutieren. Die Behauptung folgt nun aus (4.26).

Für den Induktionsschritt nehme man an, dass die Aussage des Theorems für  $M - 1$  gelte. Zu zeigen ist also, dass

$$L_{g_{i_1}} L_{g_{i_2}} \cdots L_{g_{i_M}} h - L_{g_{j_1}} L_{g_{j_2}} \cdots L_{g_{j_M}} h = 0$$

gilt, wobei

$$\sum_{h=1}^M i_h = \sum_{h=1}^M j_h =: b. \quad (4.40)$$

Falls  $L_{g_{i_1}} = L_{g_{j_1}}$  gilt, folgt die Behauptung aus der Induktionsannahme. Sei also für den weiteren Verlauf des Beweises ohne Einschränkung  $L_{g_{i_1}} = L_{g_0}$  und  $L_{g_{j_1}} = L_{g_1}$  angenommen. Dann folgt wegen (4.40), dass ein Index  $h \neq 1$  existiert, so dass  $L_{g_1} = L_{g_{i_h}} \neq L_{g_{j_h}} = L_{g_0}$ . O.E. kann man annehmen, dass  $h = 2$  gilt, weil  $M - 1$   $L_{g_i}$  Operatoren angewendet auf  $h$  nach Induktionsannahme beliebig kommutieren. Es bleibt also  $L_{g_0} L_{g_1} L_{g_{i_3}} \cdots L_{g_{i_M}} h = L_{g_1} L_{g_0} L_{g_{j_3}} \cdots L_{g_{j_M}} h$  zu zeigen. Nach Induktionsannahme gilt aber auch

$$L_{g_{i_3}} \cdots L_{g_{i_M}} h = L_{g_{j_3}} \cdots L_{g_{j_M}} h,$$

weil wegen (4.40)  $\sum_{h=3}^M i_h = \sum_{h=3}^M j_h$  gilt. Folglich bleibt zu zeigen, dass  $L_{g_0} L_{g_1} L_{g_{i_3}} \cdots L_{g_{i_M}} h = L_{g_1} L_{g_0} L_{g_{i_3}} \cdots L_{g_{i_M}} h$  gilt.

$$\begin{aligned}
[L_{g_0} L_{g_1} - L_{g_1} L_{g_0}][L_{g_{i_3}} \cdots L_{g_{i_M}} h] &= [g_0, g_1] \left[ \frac{\partial^{M-2+1}}{\partial x^{M-2+1}} h \prod_{h=3}^M g_{i_h} \right. \\
&\quad \left. + \sum_{k=1}^{M-2} \frac{\partial^k}{\partial x^k} h \sum_{\substack{j+h=k-1 \\ 0 \leq h \leq b-1 \\ 0 \leq j \leq M-1-b}} a_{kh} g_0^j g_1^h T_{kh} \right], \tag{4.41}
\end{aligned}$$

wobei  $a_{kh} \in \mathbb{Z}$  und die Faktoren  $T_{kh}$  Polynome der aus  $g_0, \frac{\partial}{\partial x} g_0, \dots, \frac{\partial^{M-b-k+h}}{\partial x^{M-b-k+h}} g_0$  und  $g_1, \frac{\partial}{\partial x} g_1, \dots, \frac{\partial^{b-1-h}}{\partial x^{b-1-h}} g_1$  zusammengesetzten Monome sind. Betrachtet man nun (4.41), fällt der erste Summand wegen (4.38) für  $k = M$  und die restlichen Summanden wegen (4.38) für  $k = 2, \dots, M-1$  weg.  $\square$

**Bemerkung 4.8** *Anschaulich liefert Theorem 4.7 Bedingungen um garantieren zu können, dass bis zu  $M$   $L_{g_i}$  Operatoren, die auf  $h$  angewendet werden sollen, kommutieren. Folglich reichen die Voraussetzungen dieses Theorems anstatt Bedingung (4.3) aus, um Lemma 4.4 für alle  $s \leq M$  zu beweisen. Das Theorem beantwortet also die Frage, wie man Bedingung (4.3) für spezielles  $M \in \mathbb{N}$  abschwächen kann. Allerdings gilt diese Abschwächung nur für den eindimensionalen Fall. Für den höherdimensionalen Fall haben wir hingegen gesehen, dass bereits für  $M = 3$  die zusätzlich zu fordernden Bedingungen sehr komplex werden.*



# Kapitel 5

## Notwendige und hinreichende Bedingungen

Bisher wurde in dieser Arbeit immer von einer Kontrollfunktion des Typs (3.31) ausgegangen. Es wurde also vorausgesetzt, dass in (2.34) nur Terme der Form (3.32) eingesetzt werden. In diesem Abschnitt wird diese Annahme aufgehoben, um eine größere Flexibilität bei der Wahl von  $u_T$  zu erhalten. Außerdem erhält man auf diesem Weg nicht nur hinreichende, sondern auch notwendige Bedingungen, damit

$$\Delta h(T, x, u_T) = \mathcal{O}(T^{M+2}) \quad (5.1)$$

beziehungsweise

$$\Delta \Phi(T, x, u_T) = \mathcal{O}(T^{M+2}) \quad (5.2)$$

für  $M \geq 2$  gilt. Für  $M < 2$  haben wir diese Frage bereits in Theorem 3.3 beantwortet und gezeigt, dass es keinerlei weiterer Bedingungen bedarf, falls man  $u_T^1$  analog (3.31) definiert. Es geht folglich um Terme in der Kontrollfunktion der Ordnung  $\mathcal{O}(T^2)$  oder höher. Es wird sich herausstellen, dass man für (5.1) und (5.2) unterschiedliche Bedingungen benötigt. Deshalb werden die jeweiligen Resultate in getrennten Abschnitten dargestellt. Im nächsten Abschnitt wird (5.1) behandelt, bevor wir uns auf der Grundlage der dort erhaltenen Resultate mit Ausdruck (5.2) beschäftigen.

### 5.1 für eine beliebige Output-Funktion $h$

Begonnen wird dieser Abschnitt mit einem Theorem aus [11], das eine sowohl hinreichende als auch notwendige Bedingung für (5.1) mit  $M = 2$  liefert.

**Theorem 5.1** *Betrachte das System (2.16), das zeitkontinuierliche closed-loop System (2.18), eine hinreichend glatte Funktion  $h : \mathbb{R}^n \rightarrow \mathbb{R}$  und eine kompakte Menge  $K \subset \mathbb{R}^n$ . Falls*

$$| L_{[g_0, g_1]} h(x) \cdot L_{g_0 + g_1 u_0} u_0(x) | \leq c | L_{g_1} h(x) | \quad (5.3)$$

für eine Konstante  $c \geq 0$  und alle  $x \in K$  gilt, existiert eine zulässige Feedback-Regelung  $u_T : \mathbb{R}^n \rightarrow \mathbb{R}$ , die (5.1) auf  $K$  mit  $M = 2$  erfüllt. Diese ist gegeben durch jedes Feedback  $u_T : \mathbb{R}^n \rightarrow \mathbb{R}$  der Form

$$u_T(x) = \begin{cases} u_T^2(x) + \frac{T^2}{12} \frac{L_{[g_0, g_1]} h(x) \cdot L_{g_0 + g_1 u_0} u_0(x)}{L_{g_1} h(x)} + \mathcal{O}(T^3), & x \in \tilde{K} \\ u_T^1(x) + \mathcal{O}(T^2), & x \notin \tilde{K} \end{cases} \quad (5.4)$$

mit  $u_T^1$  und  $u_T^2$  definiert gemäß (3.31) und

$$\tilde{K} := \{x \in K \mid L_{g_1} h(x) \neq 0\}.$$

Umgekehrt gilt, dass wenn eine zulässige Feedback-Regelung  $u_T : \mathbb{R}^n \rightarrow \mathbb{R}$  existiert, die (5.1) auf  $\tilde{K}$  mit  $M = 2$  erfüllt, Bedingung (5.3) für alle  $x \in \text{cl } \tilde{K}$  gelten muss. In diesem Fall muss das Feedback  $u_T$  von der Form (5.4) für alle  $x \in \tilde{K}$  sein.

*Beweis:* Beachtet man, dass  $L_{g_0 + g_1 u_0} u_0(x) = \frac{\partial}{\partial x} u_0 [g_0 + g_1 u_0]$  gilt, erhält man aus der Taylorentwicklung von  $h(\Phi(t, x))$  in  $t = 0$  unter Berücksichtigung von (3.40), (3.41) und (3.45)

$$\begin{aligned} h(\Phi(T, x)) &\stackrel{(3.18)}{=} h(x) + \sum_{i=1}^3 \frac{T^i}{i!} \left. \frac{d^i h(\Phi(t, x))}{dt^i} \right|_{t=0} + \mathcal{O}(T^4) \\ &= h(x) + \sum_{i=0}^2 T^{i+1} [L_{g_1} h(x) u_i(x) + p_i(x, u_0, \dots, u_{i-1})] \\ &\quad + \frac{T^3}{12} \left( (L_{g_0} L_{g_1} h(x) - L_{g_1} L_{g_0} h(x)) L_{g_0 + g_1 u_0} u_0(x) \right) + \mathcal{O}(T^4) \end{aligned} \quad (5.5)$$

mit  $p_i$  aus (3.2). Unter Ausnutzung der Tatsache, dass

$$L_{g_0} L_{g_1} h(x) - L_{g_1} L_{g_0} h(x) = L_{[g_0, g_1]} h(x)$$

und aus dem induktiven Vergleich der Koeffizienten von (5.5) mit (3.1) für  $i = 0, 1, 2$  erhält man für  $x \notin \tilde{K}$ , dass das in (5.4) angegebene Feedback (5.1) mit  $M = 2$  unter Bedingung (5.3) realisiert. Für  $x \in \tilde{K}$  zeigt dieser Koeffizientenvergleich, dass jedes Feedback  $\tilde{u}_T$  der Form

$$\tilde{u}_T(x) = \begin{cases} u_0(x) + T u_1(x) + T^2 \tilde{u}_2(x) + \mathcal{O}(T^3), & x \in \tilde{K} \\ u_0(x) + T u_1(x) + \mathcal{O}(T^2), & x \notin \tilde{K} \end{cases} \quad (5.6)$$

mit  $u_0$  und  $u_1$  aus (3.32) und Kontrollfunktion  $\tilde{u}_2(x)$ , für die

$$\tilde{u}_2(x) L_{g_1} h(x) = \frac{1}{12} \left( L_{[g_0, g_1]} h(x) \cdot L_{g_0 + g_1 u_0} u_0(x) \right) + u_2(x) L_{g_1} h(x) \quad (5.7)$$

mit  $u_2(x)$  auch aus (3.32) gilt, (5.1) mit  $M = 2$  erfüllt.

Nehme man jetzt an, dass (5.3) für alle  $x \in K$  gilt. Dann folgt, dass  $u_T$  aus (5.4) (5.6) und (5.7) erfüllt und das Feedback zulässig im Sinne von Definition 2.27 ist, weil es wegen (5.3) beschränkt ist. Insbesondere zeigt dies die Existenz eines Feedbacks  $u_T$ , das (5.1) auf  $\text{cl } \tilde{K}$  mit  $M = 2$  erfüllt.

Umgekehrt nehme man an, dass  $u_T$  ein zulässiges Feedback ist, dass (5.1) auf  $K$  mit  $M = 2$  erfüllt. Dann muss diese Steuerung die Bedingungen (5.6) und (5.7) erfüllen. Da  $u_T$  zulässig ist, ist es insbesondere auch beschränkt. Deshalb impliziert (5.7) (5.3) für  $x \in \tilde{K}$ . Weil alle Ausdrücke in (5.3) stetig in  $x$  sind, erhält man (5.3) sogar für alle  $x \in \text{cl } \tilde{K}$ . Zusätzlich zeigt der induktive Vergleich von (5.5) mit (3.1), dass jedes Feedback, das (5.1) mit  $M = 2$  erfüllen soll, die Bedingungen (5.6) und (5.7) für  $x \in \tilde{K}$  erfüllen muss. Das zeigt, dass  $u_T$  von der behaupteten Form sein muss.  $\square$

**Bemerkung 5.2** Für  $x \notin \tilde{K}$  ist  $L_{g_1}h(x) = 0$ . Entsprechend hat der Term mit  $u_2$  in der Fließentwicklung keine Auswirkung auf Terme vierter oder niedrigerer Ordnung. Betrachtet man nun Bedingung (5.3) ist diese gleich der vorher gezeigten hinreichenden Bedingung (3.45), d.h. es ist nicht gelungen diese Bedingung auf dem Komplement von  $\tilde{K}$  abzuschwächen, wohl aber auf der im Allgemeinen wesentlich größeren Menge  $\tilde{K}$  selbst.

Bevor Konsequenzen von Theorem 5.1 genauer betrachtet werden, wird es zunächst auf den Fall  $M = 3$  erweitert.

**Theorem 5.3** Betrachte das System (2.16), das zeitkontinuierliche closed-loop System (2.18), eine hinreichend glatte Funktion  $h : \mathbb{R}^n \rightarrow \mathbb{R}$  und eine kompakte Menge  $K \subset \mathbb{R}^n$ . Falls (5.3) und

$$\begin{aligned} |L_{g_1}h(x) \cdot \tilde{c}| &\geq \left| \left[ L_{[g_0, g_1]}h(x)L_{g_0+g_1u_0}^2 u_0(x) \right. \right. \\ &+ [L_{g_0}^2 L_{g_1}h(x) - L_{g_1}L_{g_0}^2 h(x)]L_{g_0+g_1u_0} u_0(x) \\ &+ [L_{g_0}L_{g_1}^2 h(x) - L_{g_1}^2 L_{g_0}h(x)]u_0(x)L_{g_0+g_1u_0} u_0(x) \left. \right] \\ &- ([L_{g_0}L_{g_1} + L_{g_1}L_{g_0}]h(x) + 2L_{g_1}^2 h(x)u_0(x)) \cdot \frac{1}{12}L_{[g_0, g_1]}h(x)L_{g_0+g_1u_0} u_0(x) \left. \right|. \end{aligned} \quad (5.8)$$

für alle  $x \in K$  und Konstanten  $c, \tilde{c} \geq 0$  gelten, existiert eine zulässige Feedback-Regelung  $u_T : \mathbb{R}^n \rightarrow \mathbb{R}$ , die (5.1) auf  $K$  mit  $M = 3$  erfüllt. Diese ist gegeben durch jedes Feedback

$u_T : \mathbb{R}^n \rightarrow \mathbb{R}$  der Form

$$\begin{aligned}
u_T(x) &:= u_T^3 + \frac{T^2 L_{[g_0, g_1]} h(x) \cdot L_{g_0+g_1} u_0(x)}{12 L_{g_1} h(x)} \\
&+ \frac{T^3 L_{[g_0, g_1]} h(x) L_{g_0+g_1}^2 u_0(x)}{24 L_{g_1} h(x)} \\
&+ \frac{T^3 [L_{g_0}^2 L_{g_1} h(x) - L_{g_1} L_{g_0}^2 h(x) + [L_{g_0} L_{g_1}^2 h(x) - L_{g_1}^2 L_{g_0} h(x)] u_0(x)] L_{g_0+g_1} u_0(x)}{24 L_{g_1} h(x)} \\
&- \frac{T^3 ([L_{g_0} L_{g_1} + L_{g_1} L_{g_0}] h(x) + 2L_{g_1}^2 h(x) u_0(x)) \cdot L_{[g_0, g_1]} h(x) L_{g_0+g_1} u_0(x)}{24 L_{g_1} h(x)} \\
&+ \mathcal{O}(T^4)
\end{aligned} \tag{5.9}$$

für  $x \in \tilde{K}$  und

$$u_T(x) := u_T^2 + \mathcal{O}(T^3) \tag{5.10}$$

für  $x \notin \tilde{K}$  mit  $u_T^2$  und  $u_T^3$  aus (3.31) und

$$\tilde{K} := \{x \in K \mid L_{g_1} h(x) \neq 0\}.$$

Umgekehrt gilt, dass wenn eine zulässige Feedback-Regelung  $u_T : \mathbb{R}^n \rightarrow \mathbb{R}$  existiert, die (5.1) auf  $\tilde{K}$  mit  $M = 3$  erfüllt, dann gelten (5.3) und (5.8) für alle  $x \in \text{cl } \tilde{K}$ . In diesem Fall muss das Feedback  $u_T$  von der oben beschriebenen Form für alle  $x \in \tilde{K}$  sein.

*Beweis:* Analog zu dem Beweis von Theorem 5.1 erhält man aus der Taylorentwicklung von  $h(\Phi(t, x))$  in  $t = 0$  unter Berücksichtigung von (3.40)-(3.42), sowie (3.45) und (3.47)

$$\begin{aligned}
h(\Phi(T, x)) &\stackrel{(3.18)}{=} h(x) + \sum_{i=1}^4 \frac{T^i}{i!} \left. \frac{d^i h(\Phi(t, x))}{dt^i} \right|_{t=0} + \mathcal{O}(T^5) \\
&= h(x) + \sum_{i=0}^3 T^{i+1} [L_{g_1} h(x) u_i(x) + p_i(x, u_0, \dots, u_{i-1})] \\
&+ \frac{T^3}{12} \left( (L_{g_0} L_{g_1} h(x) - L_{g_1} L_{g_0} h(x)) L_{g_0+g_1} u_0(x) \right) \\
&+ \frac{T^4}{24} \left( [L_{g_0}^2 L_{g_1} h(x) - L_{g_1} L_{g_0}^2 h(x)] L_{g_0+g_1} u_0(x) \right. \\
&\quad + [L_{g_0} L_{g_1} h(x) - L_{g_1} L_{g_0} h(x)] L_{g_0+g_1}^2 u_0(x) \\
&\quad \left. + [L_{g_0} L_{g_1}^2 h(x) - L_{g_1}^2 L_{g_0} h(x)] u_0 L_{g_0+g_1} u_0(x) \right) + \mathcal{O}(T^5) \tag{5.11}
\end{aligned}$$

mit  $p_i$  aus (3.2). Analog zu dem Beweis von Theorem 5.1 folgt aus einem induktiven Koeffizientenvergleich von (5.11) mit (3.1) für  $i = 0, 1, 2, 3$  und  $x \notin \tilde{K}$ , dass das in (5.10) angegebene

Feedback (5.1) mit  $M = 3$  unter den Bedingungen (5.3) und (5.8) erfüllt. Für  $x \in \tilde{K}$  liefert diese Vorgehensweise, dass jedes Feedback  $\tilde{u}_T$  der Form

$$\tilde{u}_T(x) = \begin{cases} u_0(x) + Tu_1(x) + T^2\tilde{u}_2(x) + T^3\tilde{u}_3(x)\mathcal{O}(T^4), & x \in \tilde{K} \\ u_0(x) + Tu_1(x) + T^2u_2(x) + \mathcal{O}(T^3), & x \notin \tilde{K} \end{cases}$$

mit  $u_0(x), u_1(x), u_2(x)$  aus (3.32),  $\tilde{u}_2(x)$ , das (5.7) erfüllt, und Kontrollfunktion  $\tilde{u}_3(x)$ , für die

$$\begin{aligned} L_{g_1}h(x)\tilde{u}_3(x) &= \frac{1}{24} \left[ L_{[g_0, g_1]}h(x)L_{g_0+g_1u_0}^2u_0(x) \right. \\ &+ [L_{g_0}^2L_{g_1}h(x) - L_{g_1}L_{g_0}^2h(x)]L_{g_0+g_1u_0}u_0(x) \\ &+ [L_{g_0}L_{g_1}^2h(x) - L_{g_1}^2L_{g_0}h(x)]u_0(x)L_{g_0+g_1u_0}u_0(x) \left. \right] \\ &- \left( \frac{1}{2}[L_{g_0}L_{g_1} + L_{g_1}L_{g_0}]h(x) + L_{g_1}^2h(x)u_0(x) \right) \cdot \frac{1}{12}L_{[g_0, g_1]}h(x)L_{g_0+g_1u_0}u_0(x) \\ &+ L_{g_1}h(x)u_3(x). \end{aligned} \quad (5.12)$$

mit  $u_3(x)$  aus (3.32) gilt, (5.1) erfüllt. Zu beachten ist dabei, dass  $\tilde{u}_2$  bereits wegen der Terme dritter Ordnung festgelegt ist. Man muss also  $L_{g_1}h(x)u_3(x)$  dazu nutzen, sowohl die explizit in (5.11) aufgeführten Terme vierter Ordnung als auch die Auswirkungen der Wahl von  $\tilde{u}_2$  auszugleichen. Entscheidend sind hierfür alle Terme in  $p_3(x, u_0, u_1, u_2)$ , die tatsächlich von  $u_2$  abhängen, also nach (3.15)

$$\left( \frac{1}{2}[L_{g_0}L_{g_1} + L_{g_1}L_{g_0}] + u_0L_{g_1}^2 \right) h(x)u_2. \quad (5.13)$$

Der Rest des Beweises verläuft vollkommen analog zu dem Beweis von Theorem 5.1.  $\square$

**Bemerkung 5.4** *An Theorem 5.3 kann man gut die Veränderung der Sichtweise dieses Kapitels gegenüber den Resultaten in Kapitel 4 beobachten. In Theorem 4.5 wurde eine Bedingung an das Kontrollsystem (2.16) selbst gestellt. Dagegen wird in diesem Kapitel versucht, Bedingungen anzugeben, unter denen garantiert werden kann, dass die Feedback-Regelung zulässig im Sinne von Definition 2.27 bleibt.*

**Bemerkung 5.5** *Bedingung (5.3) aus Theorem 5.1 und die zusätzliche Bedingung (5.8) aus Theorem 5.3 sind nur auf  $\text{cl } \tilde{K}$  notwendig und hinreichend wie in der auf [19] beruhenden Bemerkung in [11, Remark 4.2] für Theorem 5.1 gezeigt worden ist. Interessant ist die potenzielle Abschwächung von (5.3) beziehungsweise (5.8) außerhalb von  $\text{cl } \tilde{K}$ , falls  $L_{g_1}L_{g_0}h(x) \neq 0$ . Dies sei exemplarisch für (5.8) demonstriert. Wie bereits aus Theorem 5.1 bekannt, bekommt*

man die Terme dritter Ordnung unter Bedingung (5.3) durch eine entsprechende Wahl der Feedback-Regelung in den Griff. Hierbei ist zu beachten, dass die Terme zweiter Ordnung in  $u_T(x)$  für  $L_{g_1}h(x) = 0$  beliebig vorgebar waren. Dies können wir in diesem Fall ausnutzen, um die Terme vierter Ordnung auszugleichen. Zu diesem Zweck betrachtet man abermals (5.13), also alle von  $u_2$  abhängigen Terme in  $p_3(x, u_0, u_1, u_2)$ . Zu beachten ist hierbei, dass die Terme  $\frac{1}{2}L_{g_0}L_{g_1}h(x)u_2$  und  $u_0L_{g_1}^2h(x)u_2$  wegen  $L_{g_1}h(x) = 0$  wegfallen. Berücksichtigt man nun Bedingung (5.3) und nimmt ohne Einschränkung an, dass  $L_{g_0+g_1u_0}u_0(x) \neq 0$  ist. Ansonsten ist die Geschwindigkeit des Wechsels von  $h$  in Richtung  $[g_0, g_1]$ , sowie bezüglich der anderen beiden relevanten Richtungen, gleich Null und die Terme aus (5.11), die es auszugleichen gilt, verschwinden. So folgt, dass  $\tilde{u}_2$  außerhalb von  $\tilde{K}$  so zu setzen ist, dass folgende Gleichung erfüllt ist

$$\begin{aligned} \frac{1}{2}L_{g_1}L_{g_0}h(x)\tilde{u}_2 &= \frac{1}{2}L_{g_1}L_{g_0}h(x)u_2 \\ &\quad - \frac{1}{24} \left[ L_{g_1}L_{g_0}^2h(x) + L_{g_1}^2L_{g_0}h(x)u_0 \right] L_{g_0+g_1u_0}u_0(x). \end{aligned} \quad (5.14)$$

In diesem Zusammenhang spielt folgende Definition aus [13] eine zentrale Rolle

**Definition 5.6** Ein auf  $D$  definiertes nichtlineares Kontrollsystem der Form (2.16) mit Outputfunktion  $h(x)$  besitzt den Relativgrad  $\rho$ ,  $1 \leq \rho \leq n$ , in einem Gebiet  $D_0 \subset D$ , falls

$$L_{g_1}L_{g_0}^{i-1}h(x) = 0, \quad i = 1, 2, \dots, \rho - 1; \quad L_{g_1}L_{g_0}^{\rho-1}h(x) \neq 0 \quad (5.15)$$

für alle  $x \in D_0$ .

Wir haben also in der letzten Bemerkung den Fall mit Relativgrad gleich eins behandelt. Für weiterführende Ergebnisse bezüglich des Relativgrades sei auf [19] verwiesen.

**Bemerkung 5.7** Für eine genauere Interpretation von (5.3) sei auf [11, Remark 4.3] verwiesen. Im Wesentlichen sorgt diese Bedingung dafür, dass die Feedback-Regelung auch für Folgen  $x_n : L_{g_1}h(x_n) \rightarrow 0$  beschränkt bleibt.

**Bemerkung 5.8** Es ließen sich mit Hilfe der in Kapitel 3 hergeleiteten Resultate weitere Theoreme für  $M = 4$  und  $M = 5$  formulieren und beweisen. Allerdings muss man, wie bereits in Theorem 5.3 gezeigt, auch die Auswirkungen der zusätzlichen Terme wegen  $\tilde{u}_2$  und  $\tilde{u}_3$  berücksichtigen. Es reicht also nicht, die aus Abschnitt 3.4 bekannten Terme auszugleichen.

## 5.2 für die Lösung $\Phi$ der DGL

Einer der Gründe für die verhältnismäßig einfach zu erhaltenen Formeln für (5.1) ist, dass die zu vergleichenden Werte  $h(\Phi)$  und  $h(\Phi_T)$  eindimensional sind. Als Konsequenz dessen wird die notwendige und hinreichende Bedingung für (5.2) restriktiver ausfallen, wie das folgende Theorem zeigt.

**Theorem 5.9** Betrachte das System (2.16), das zeitkontinuierliche closed-loop System (2.18) und eine kompakte Menge  $K \subset \mathbb{R}^n$ , für die gilt, dass  $K = \text{cl int } K$  ist. Dann existiert eine zulässige Feedback-Regelung  $u_T : \mathbb{R}^n \rightarrow \mathbb{R}$ , die (5.2) auf  $K$  mit  $M = 2$  erfüllt, genau dann, wenn eine beschränkte Funktion  $\alpha : K \rightarrow \mathbb{R}$  existiert, die der Gleichung

$$[g_0, g_1](x)L_{g_0+g_1u_0}u_0(x) = \alpha(x)g_1(x) \quad (5.16)$$

genügt. In diesem Fall erreicht jedes Feedback  $u_T(x) : \mathbb{R}^n \rightarrow \mathbb{R}$  der Form

$$u_T(x) = \begin{cases} u_T^2(x) + \frac{T^2}{12}\alpha(x) + \mathcal{O}(T^3), & x \in \text{cl } \tilde{K} \\ \text{beliebig}, & x \notin \text{cl } \tilde{K} \end{cases} \quad (5.17)$$

mit  $u_T^2$  aus (3.31) und

$$\tilde{K} := \{x \in K \mid g_1(x) \neq 0\}$$

(5.2) mit  $M = 2$ . Darüberhinaus ist jedes Feedback, das (5.2) mit  $M = 2$  erfüllt, von der Form (5.17) für  $x \in \tilde{K}$  und die Funktion  $\alpha$  in (5.16) kann als  $\alpha(x) = 0$  für  $x \notin \tilde{K}$  gewählt werden.

Beweis: Man zeigt zuerst, dass unter Annahme (5.16) jede Steuerung der Form (5.17) die Behauptung erfüllt. Hierzu unterscheidet man die folgenden beiden Fälle:

- $x \notin \text{cl } \tilde{K}$ : Der Wert der Feedback-Steuerung ist beliebig. Dies folgt, weil sich auf  $K \setminus \text{cl } \tilde{K}$  das Kontrollsystem (2.16) zu  $\dot{x} = g_0(x)$  vereinfacht. Deshalb stimmen auf der offenen Menge  $\text{int } K \setminus \text{cl } \tilde{K}$  die Taylorentwicklungen von  $\Phi(t, x)$  und  $\Phi_T(t, x, u_T)$  für beliebige Ordnung überein. Dies ist unabhängig von den Werten von  $u_0$  und  $u_T$ , d.h. man erhält (5.2) für jedes  $M > 0$  und beliebiges  $u_T$ . Wegen der Stetigkeit der Ausdrücke in der Taylorentwicklung lässt sich diese Eigenschaft auf  $\text{cl int } K \setminus \text{cl } \tilde{K}$  übertragen, der  $K \setminus \text{cl } \tilde{K}$  enthält, weil vorausgesetzt worden ist, dass  $K = \text{cl int } K$ .
- $x \in \text{cl } \tilde{K}$ : Angenommen es existiert eine beschränkte Funktion  $\alpha$ , die (5.16) erfüllt. Fixiere  $i \in \{1, \dots, n\}$  und betrachte die Funktion  $h_i(x) = x_i$ . Dann zeigt eine einfache Rechnung, in welche die Identitäten

$$L_{g_1}h_i(x) = g_1(x)_i \quad \text{und} \quad L_{[g_0, g_1]}h_i(x) = [g_0, g_1](x)_i$$

eingehen, dass die Funktion  $\alpha$  aus (5.16) der Gleichung

$$\alpha(x) = \frac{[g_0, g_1](x)_i L_{g_0+g_1u_0}u_0(x)}{g_1(x)_i} = \frac{L_{[g_0, g_1]}h_i(x) \cdot L_{g_0+g_1u_0}u_0(x)}{L_{g_1}h_i(x)}$$

genügt, falls  $g_1(x)_i \neq 0$ . Falls jedoch  $g_1(x)_i = 0$ , ist die Steuerung von der Form  $u_T^1 + \mathcal{O}(T^2)$  mit  $u_T^1$  aus (3.31). Also ist das Feedback von der Form (5.4) für  $h =$

$h_i$  und man kann Theorem 5.1 verwenden, um  $\Delta h_i(T, x, u_T) = \mathcal{O}(T^4)$  für alle  $x \in \text{cl } \tilde{K}$  zu folgern. Weil  $i \in \{1, \dots, n\}$  beliebig gewählt war, impliziert dieses Argument  $\Delta \Phi(T, x, u_T) = \mathcal{O}(T^4)$ . Desweiteren gilt nach Theorem 5.1, dass jedes Feedback, das  $\Delta \Phi(T, x, u_T) = \mathcal{O}(T^4)$  erfüllt, von der Form (5.17) sein muss, wenn  $L_{g_1} h_i(x) = g_1(x)_i \neq 0$ . Und weil für jedes  $x \in \tilde{K}$  ein  $i \in \{1, \dots, n\}$  existiert, s.d.  $g_1(x)_i \neq 0$  muss (5.17) für alle  $x \in \tilde{K}$  gelten.

Umgekehrt nehme man an, dass eine zulässige Feedback-Regelung existiert, die (5.2) mit  $M = 2$  erfüllt. Dann gilt für jedes  $x \in \tilde{K}$ , dass  $g_1(x)_i \neq 0$  für ein geeignetes  $i \in \{1, \dots, n\}$ . Deshalb folgt aus Theorem 5.1 für  $h = h_i$ , dass  $u_T$  von der Form (5.4) für  $i \in \{1, \dots, n\}$  sein muss, d.h. von der Form (5.17). Insbesondere existiert  $\alpha(x)$ , so dass (5.16) auf  $\tilde{K}$  erfüllt ist und weil  $u_T$  zulässig ist, muss diese Funktion  $\alpha$  auf  $\tilde{K}$  beschränkt sein. Auf der offenen Menge  $\text{int}(K \setminus \text{cl } \tilde{K})$  gilt  $g_1 \equiv 0$ . Entsprechend gilt, dass  $[g_0, g_1] = 0$ , also (4.3) erfüllt ist. Aus Stetigkeitsgründen gilt dies auch auf  $\text{cl int}(K \setminus \text{cl } \tilde{K}) = K \setminus \text{cl } \tilde{K}$ . Deshalb können wir  $\alpha(x) = 0$  für  $x \in K \setminus \text{cl } \tilde{K}$  wählen. Dies definiert eine beschränkte Funktion  $\alpha$  für  $x \in \tilde{K} \cup (K \setminus \text{cl } \tilde{K}) = K \setminus (\text{cl } \tilde{K} \setminus \tilde{K})$ . Es muss noch  $\alpha$  für  $\text{cl } \tilde{K} \setminus \tilde{K}$  definiert werden. Wir erhalten, dass  $\text{cl } \tilde{K} = \text{cl int } \tilde{K}$ , weil  $\text{cl int } K = K$  und  $\tilde{K}$  relativ offen zu  $K$  ist. Deshalb kann man zu jedem  $x \in \text{cl } \tilde{K}$  eine Folge  $x_n \rightarrow x$  mit  $x_n \in \text{int } \tilde{K}$  finden, d.h. insbesondere  $x_n \notin \text{cl } \tilde{K} \setminus \tilde{K}$ . Da  $\alpha$  bereits auf dieser Menge definiert und beschränkt ist, sowie (5.16) erfüllt, erhält man aus Stetigkeitsüberlegungen

$$\begin{aligned} \|[g_0, g_1](x) L_{g_0+g_1 u_0} u_0(x)\| &= \lim_{n \rightarrow \infty} \|[g_0, g_1](x_n) L_{g_0+g_1 u_0} u_0(x_n)\| \\ &\leq \lim_{n \rightarrow \infty} \underbrace{|\alpha(x_n)|}_{\text{beschränkt}} \underbrace{\|g_1(x_n)\|}_{\rightarrow 0} = 0. \end{aligned}$$

Daraus folgt, dass  $[g_0, g_1](x) L_{g_0+g_1 u_0} u_0(x) = g_1(x) = 0$  und man entsprechend  $\alpha(x) = 0$  auf  $\text{cl } \tilde{K} \setminus \tilde{K}$  setzen kann, um (5.16) zu erfüllen. Dies vervollständigt den Beweis.  $\square$

**Bemerkung 5.10** Die zu (5.16) analoge Bedingung für  $M = 3$  ist, dass eine beschränkte Funktion  $\beta : K \rightarrow \mathbb{R}$  existiert, für die

$$\begin{aligned} 24\beta(x)g_1(x)_i &= (L_{g_0}^2 g_1(x)_i - L_{g_1} L_{g_0} g_0(x)_i) L_{g_0+g_1 u_0} u_0(x) \\ &\quad + [g_0, g_1](x)_i L_{g_0+g_1 u_0}^2 u_0(x) \\ &\quad + (L_{g_0} L_{g_1} g_1(x)_i - L_{g_1}^2 g_0(x)_i) u_0(x) L_{g_0+g_1 u_0} u_0(x) \\ &\quad - (L_{g_0} g_1(x)_i + L_{g_1} g_0(x)_i + 2L_{g_1} g_1(x)_i u_0(x)) \alpha(x) \end{aligned}$$

für jedes  $i = 1, \dots, n$  und  $\alpha$  aus (5.16). Dabei wurde für jedes  $i$   $h_i(x) = x_i$  gewählt und ausgenutzt, dass  $L_{g_k} h_i(x) = g_k(x)_i$  ist. Die Komplexität der Bedingungen wächst also exponentiell an.



**Bemerkung 5.11** *Obwohl die Bedingungen für sampled-data Feedback-Steuerungen für höhere Ordnungen recht aufwendig erscheinen, ist es für ein gegebenes zeitkontinuierliches, rückgekoppeltes System möglich, eine verhältnismäßig einfache, rekursive Maple-Routine anzugeben, die*

- für beliebige Ordnung überprüft, ob die geforderten Bedingungen erfüllt sind und
- die entsprechende sampled-data Steuerung berechnet, falls dies möglich ist.

*Der Code des zu diesem Zweck entwickelten Maple-Programms findet sich in [11].*

Ausgehend von den in diesem Kapitel hergeleiteten Resultaten betrachten wir nun erneut die Steuerung  $u_T^M$  aus (3.31) und zeigen, dass Bedingung (4.3) sogar notwendig für (4.22) beziehungsweise (5.2) ist. Dies gilt zumindest für eine geeignete Zustandsmenge.

**Korollar 5.12** *Betrachte das System (2.16), das zeitkontinuierliche closed-loop System (2.18) und das sampled-data closed-loop System (2.32) mit Kontrollfunktion  $u_T^M$  aus (3.31) für  $M \geq 2$ . Sei angenommen, dass (4.22) gilt. Dann ist Bedingung (4.3) für jedes  $x \in \mathbb{R}^n$  mit  $L_{g_0+g_1 u_0} u_0(x) \neq 0$  erfüllt.*

*Beweis:* Falls (4.22) für ein spezielles  $M \geq 2$  erfüllt ist, gilt es insbesondere auch für  $M = 2$  in jedem kompakten Ball  $K = \text{cl } B_r(0)$ . Mit Hilfe von Theorem 5.9 folgt die Existenz einer Funktion  $\alpha$ , die Gleichung (5.16) genügt. Desweiteren erhalten wir  $u_T^M = u_T + \mathcal{O}(T^3)$  für  $u_T$  aus (5.17). Das ist aber gemäß unseren Voraussetzungen nur möglich, falls  $\alpha(x) = 0$  für  $x \in \tilde{K}$  gilt. Weil der Wert von  $\alpha(x)$  für  $x \notin \tilde{K}$  als Null gewählt werden kann, erhält man  $\alpha \equiv 0$  auf  $K$ . Dies impliziert aber, dass die rechte Seite von (5.16) für alle  $x \in K$  gleich Null ist. Daraus folgt mit der Tatsache, dass  $K$  ein beliebiger kompakter Ball ist, dass

$$[g_0, g_1]L_{g_0+g_1 u_0} u_0(x) = 0$$

für alle  $x \in \mathbb{R}^n$  gilt und damit die Behauptung. □



# Kapitel 6

## Beispiele

In diesem Kapitel werden die hergeleiteten Resultate anhand von zwei Beispielen illustriert. Dies geschieht unter anderem mit Hilfe der in Bemerkung 5.11 erwähnten Maple-Routine aus [11], die für ein gegebenes System Feedback-Steuerungen berechnet bzw. angibt, dass dies für die gewünschte Ordnung nicht möglich ist. Das erste Beispiel ist akademischer Natur. Es zeigt, wie sich Lösungstrajektorien ändern, wenn man Feedback-Steuerungen höherer Ordnung zugrunde legt. Anschließend wird das Moore-Greitzer Modell eines Flugzeugmotors betrachtet, an dem eine weitere Einsatzmöglichkeit des Verfahrens demonstriert wird.

### 6.1 Akademisches Beispiel

Betrachtet wird ein System, das zwar Bedingung (4.3) nicht erfüllt, dafür allerdings unter anderem Bedingung (5.16). Es ist durch

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} x_2^2 \\ 2x_1^2 \end{pmatrix} u \quad (6.1)$$

gegeben, wobei die für das kontinuierliche System berechnete Regelung  $u_0(x) = -x_1 - 1$  ist. Man kann leicht verifizieren, dass

$$[g_0, g_1] = \begin{pmatrix} x_2^2 \\ 2x_1^2 \end{pmatrix} = g_1(x)$$

gilt. Hieraus folgt sofort, dass (5.16) für jede kompakte Menge mit  $\alpha(x) = L_{g_0+g_1u_0}u_0(x) = x_2^2x_1 + x_2^2 - x_1$  erfüllt ist. Die resultierenden sampled-data Feedback-Steuerungen für  $M = 0, 1$

und 2 sind entsprechend

$$u_T^0(x) = -x_1 - 1, \quad (6.2)$$

$$u_T^1(x) = u_T^0(x) + \frac{T}{2} (x_1 x_2^2 + x_2^2 - x_1) \quad \text{und} \quad (6.3)$$

$$u_T^2(x) = u_T^1(x) + T^2 \left( -\frac{1}{4}x_1 + \frac{3}{4}x_1 x_2^2 - \frac{2}{3}x_1^4 x_2 - \frac{4}{3}x_1^3 x_2 - \frac{2}{3}x_1^2 x_2 - \frac{1}{6}x_1 x_2^4 - \frac{1}{6}x_2^4 + \frac{7}{12}x_2^2 \right). \quad (6.4)$$

Hierbei gehen (3.32) und (5.17) in die Berechnung von (6.4) ein. Abbildung 6.1 zeigt die  $x_1$ -Komponente der Trajektorien für den Anfangswert  $x_0 = (-1, 1)^T$  und die Samplingperiode  $T = 0.1$ . Dabei zeigt die Linie ohne Markierungen die zeitkontinuierliche Trajektorie.

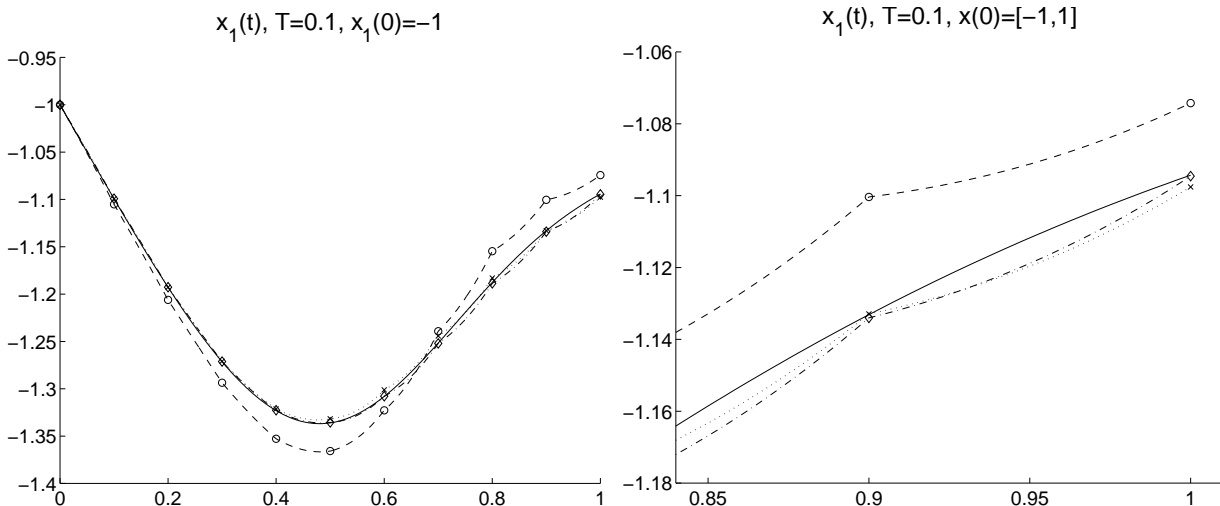


Abbildung 6.1: Vergleich der zeitkontinuierlichen und der sampled-data Lösung mit Regler (6.2, o), (6.3, x) und (6.4, ◇).

Man erwartet, dass zum Zeitpunkt  $t = 1$ , d.h. nach  $1/T$  Samplingperioden, die Differenz zwischen der zeitkontinuierlichen und der sampled-data Lösung von der Ordnung  $T^{M+1}$  ist. Dies zeigt auch die doppelt logarithmische Darstellung der Abweichungen der sampled-data Trajektorien von denen der zeitkontinuierlichen Lösung in Abbildung 6.2. Zu beachten ist dabei, dass die Markierungen in diesem Plot lediglich der Kennzeichnung der einzelnen Trajektorien dienen und keine Auswertungspunkte der sampled-data Lösungen sind.

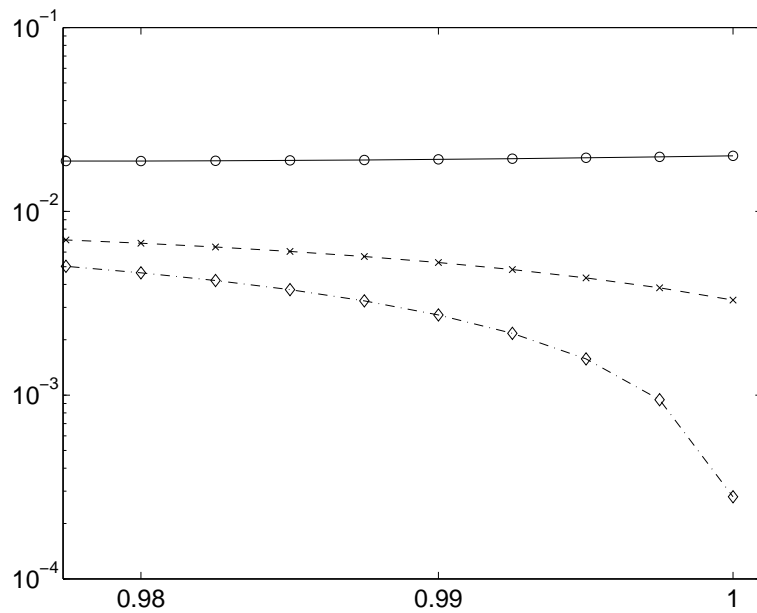


Abbildung 6.2: Differenz zwischen der zeitkontinuierlichen und der sampled-data Lösung bei  $t = 1$  für die Sampled-data Feedbacks (6.2, o), (6.3, x) und (6.4,  $\diamond$ ).

Tabelle 6.1 bestätigt sogar, dass jeweils am Ende eines Samplingintervalls die gewünschte Genauigkeit gewährleistet ist.

| Zeit | Feedback (6.2) | Feedback (6.3) | Feedback (6.4) |
|------|----------------|----------------|----------------|
| 0.0  | 0. 0 00000     | 0. 00 0000     | 0. 000 000     |
| 0.1  | 0. 0 05949     | 0. 00 0076     | 0. 000 023     |
| 0.2  | 0. 0 13389     | 0. 00 0008     | 0. 000 079     |
| 0.3  | 0. 0 22153     | 0. 00 0463     | 0. 000 199     |
| 0.4  | 0. 0 29793     | 0. 00 1797     | 0. 000 392     |
| 0.5  | 0. 0 29795     | 0. 00 4273     | 0. 000 529     |
| 0.6  | 0. 0 14681     | 0. 00 6930     | 0. 000 300     |
| 0.7  | 0. 0 12413     | 0. 00 7436     | 0. 000 380     |
| 0.8  | 0. 0 33013     | 0. 00 4613     | 0. 000 969     |
| 0.9  | 0. 0 32748     | 0. 00 0197     | 0. 000 891     |
| 1.0  | 0. 0 20056     | 0. 00 3288     | 0. 000 280     |

Tabelle 6.1: Differenz zwischen der zeitkontinuierlichen und der sampled-data Lösung mit Regler (6.2) - (6.4) für das akademische Beispiel.

Für dieses System ist es außerdem möglich mit Hilfe der in [11] vorgestellten Maple-Routine Feedback-Steuerungen zu finden, die eine wesentlich genauere Approximation realisieren. So berechnet man beispielsweise für  $M = 3$

$$u_T^3 := T^3 \left[ + \frac{1}{24}x_1x_2^6 + x_1^4x_2^3 + x_1^6 + x_1^5 + \frac{1}{24}x_2^6 + \frac{5}{3}x_1^2x_2^3 - \frac{5}{12}x_1x_2^4 + \frac{7}{3}x_1^3x_2^3 - \frac{17}{6}x_1^3x_2 \right. \\ \left. + \frac{1}{3}x_1x_2^3 - \frac{3}{8}x_2^4 + \frac{1}{3}x_1^4 + \frac{17}{24}x_1x_2^2 - \frac{5}{3}x_1^4x_2 + \frac{1}{3}x_1^7 - \frac{7}{6}x_1^2x_2 + \frac{5}{12}x_2^2 - \frac{1}{12}x_1 \right] + u_T^2 \quad (6.5)$$

und  $M = 4$

$$u_T^4 := T^4 \left[ - \frac{13}{6}x_1^2x_2^5 - \frac{9}{10}x_1^4x_2^5 + \frac{7}{48}x_1x_2^6 + \frac{7}{2}x_1^4x_2^3 - \frac{1}{120}x_1x_2^8 + \frac{22}{3}x_1^3x_2^3 - \frac{11}{15}x_1x_2^5 \right. \\ - \frac{5}{3}x_1^7x_2^2 - \frac{11}{3}x_1^4x_2^2 - \frac{1}{15}x_2^5 - \frac{43}{18}x_1^4x_2 - 7x_1^5x_2^2 + \frac{31}{144}x_2^2 + \frac{27}{10}x_1^5 - \frac{12}{5}x_1^3x_2^5 \\ - \frac{2}{3}x_1^3x_2^2 + \frac{31}{10}x_1^6 - \frac{43}{72}x_1x_2^4 - \frac{617}{180}x_1^3x_2 - \frac{1}{120}x_2^8 + \frac{23}{5}x_1^2x_2^3 - \frac{17}{3}x_1^6x_2^2 + \frac{11}{80}x_2^6 \\ \left. + \frac{23}{30}x_1x_2^3 + \frac{7}{6}x_1^7 + \frac{23}{30}x_1^4 - \frac{41}{36}x_1^2x_2 - \frac{343}{720}x_2^4 - \frac{1}{48}x_1 + \frac{1}{2}x_1x_2^2 \right] + u_T^3. \quad (6.6)$$

Klar erkennbar ist dabei, dass die Kontrollfunktion für jede weitere Ordnung immer mehr zusätzliche Summanden benötigt. Um die auftretenden Effekte noch besser darstellen zu können, seien die Feedback-Steuerungen für  $M = 5$

$$u_T^5 := T^5 \left[ + \frac{1}{720}x_2^{10} + \frac{1}{720}x_1x_2^{10} - \frac{2099}{90}x_1^6x_2^2 + \frac{118}{9}x_1^4x_2^4 + \frac{28}{45}x_1^4x_2^7 + \frac{64}{9}x_1^7x_2 + \frac{44}{9}x_1^9x_2 \right. \\ + \frac{118}{9}x_1^6x_2^4 + \frac{31}{9}x_1^7x_2^4 + \frac{38}{45}x_1x_2^7 + \frac{26}{9}x_1^6x_2 - \frac{781}{30}x_1^5x_2^2 - \frac{299}{30}x_1^3x_2^5 + \frac{76}{9}x_1^8x_2 \\ - \frac{491}{60}x_1^2x_2^5 + \frac{9}{32}x_1x_2^6 + \frac{83}{45}x_1^3x_2^7 + \frac{9}{4}x_1^7 + \frac{10}{9}x_1^{10}x_2 - \frac{361}{120}x_1^3x_2 - \frac{15}{2}x_1^7x_2^2 \\ + \frac{37}{9}x_1^3x_2^4 - \frac{81}{20}x_1^4x_2^5 + \frac{35}{18}x_1^2x_2^7 - \frac{3}{80}x_1x_2^8 + \frac{172}{9}x_1^5x_2^4 + \frac{317}{60}x_1^6 + \frac{4}{9}x_1^5x_2 \\ - \frac{121}{10}x_1^4x_2^2 + \frac{27}{4}x_1^4x_2^3 + \frac{4}{9}x_1^2x_2^4 + \frac{19}{20}x_1^4 + \frac{119}{480}x_2^6 + \frac{137}{480}x_1x_2^2 - \frac{17}{90}x_2^5 \\ - \frac{209}{480}x_2^4 - \frac{17}{9}x_1^3x_2^2 + \frac{623}{90}x_1^2x_2^3 - \frac{13}{360}x_2^8 - \frac{1}{240}x_1 + \frac{11}{90}x_2^7 + \frac{7}{80}x_2^2 + \frac{253}{20}x_1^3x_2^3 \\ \left. - \frac{5}{2}x_1^4x_2 + \frac{239}{60}x_1^5 - \frac{5}{8}x_1x_2^4 + \frac{19}{20}x_1x_2^3 - \frac{221}{90}x_1x_2^5 - \frac{19}{24}x_1^2x_2 \right] + u_T^4 \quad (6.7)$$

und  $M = 6$

$$\begin{aligned}
u_T^6 := T^6 & \left[ -\frac{20}{63}x_1^{13} - \frac{116}{21}x_1^{10}x_2^3 - \frac{211}{45}x_1^7x_2^6 - \frac{151}{420}x_1^4x_2^9 - \frac{1}{5040}x_1x_2^{12} - \frac{12}{7}x_1^{12} - \frac{1688}{63}x_1^9x_2^3 \right. \\
& - \frac{6179}{315}x_1^6x_2^6 - \frac{727}{630}x_1^3x_2^9 - \frac{1}{5040}x_2^{12} - \frac{80}{21}x_1^{11} + \frac{55}{9}x_1^{10}x_2 - \frac{1108}{21}x_1^8x_2^3 + \frac{341}{18}x_1^7x_2^4 \\
& - \frac{487}{15}x_1^5x_2^6 + \frac{154}{45}x_1^4x_2^7 - \frac{27}{20}x_1^2x_2^9 + \frac{11}{1440}x_1x_2^{10} - \frac{40}{9}x_1^{10} + \frac{7858}{315}x_1^9x_2 - \frac{3376}{63}x_1^7x_2^3 \\
& + \frac{7054}{105}x_1^6x_2^4 - \frac{8417}{315}x_1^4x_2^6 + \frac{2999}{315}x_1^3x_2^7 - \frac{71}{105}x_1x_2^9 + \frac{5}{672}x_2^{10} + \frac{607}{1260}x_2^7 + \frac{94}{63}x_1^2x_2^4 \\
& - \frac{1105}{252}x_1x_2^5 + \frac{89}{280}x_2^6 - \frac{20}{7}x_1^9 + \frac{12494}{315}x_1^8x_2 - \frac{655}{36}x_1^7x_2^2 - \frac{1844}{63}x_1^6x_2^3 + \frac{28492}{315}x_1^5x_2^4 \\
& - \frac{393}{40}x_1^4x_2^5 - \frac{3524}{315}x_1^3x_2^6 + \frac{3919}{420}x_1^2x_2^7 - \frac{131}{1440}x_1x_2^8 - \frac{38}{315}x_2^9 + \frac{1403}{90}x_1^3x_2^3 + \frac{251}{60}x_1^5 \\
& - \frac{2267}{1080}x_1^4x_2 - \frac{17}{6}x_1^3x_2^2 - \frac{20}{21}x_1^8 + \frac{9584}{315}x_1^7x_2 - \frac{21697}{420}x_1^6x_2^2 - 8x_1^5x_2^3 + \frac{1982}{35}x_1^4x_2^4 \\
& - \frac{9349}{420}x_1^3x_2^5 - \frac{32}{15}x_1^2x_2^6 + \frac{26}{7}x_1x_2^7 - \frac{47}{560}x_2^8 + \frac{127}{4320}x_2^2 - \frac{2267}{4320}x_1x_2^4 + \frac{188}{63}x_1^7 \\
& + \frac{499}{45}x_1^6x_2 + \frac{2663}{360}x_1^2x_2^3 + \frac{536}{63}x_1^4x_2^3 + \frac{10021}{630}x_1^3x_2^4 - \frac{41747}{2520}x_1^2x_2^5 - \frac{21823}{420}x_1^5x_2^2 + \frac{149}{180}x_1x_2^3 \\
& - \frac{1361}{4320}x_2^4 + \frac{149}{180}x_1^4 + \frac{11}{42}x_1x_2^6 - \frac{3839}{180}x_1^4x_2^2 + \frac{11}{80}x_1x_2^2 + \frac{451}{70}x_1^6 - \frac{455}{216}x_1^3x_2 \\
& \left. - \frac{17}{60}x_2^5 - \frac{1}{1440}x_1 + \frac{94}{63}x_1^5x_2 - \frac{467}{1080}x_1^2x_2 \right] + u_T^5 \tag{6.8}
\end{aligned}$$

angegeben. Nun wollen wir analog zu Tabelle 6.1 und Abbildung 6.2 die Daten des zeitkontinuierlichen Systems mit denen des sampled-data Systems für  $M = 2, 4$  und  $6$  vergleichen. Dies liefert Tabelle 6.2 und graphisch Abbildung 6.3. Interessant dabei ist insbesondere, dass die Ordnung  $M + 1$  der Übereinstimmung von kontinuierlicher und sampled-data Lösungstrajektorie für  $M = 4$  und  $6$  nicht mehr erreicht wird, sondern diese lediglich für die ersten Samplingperioden gilt, wie aus Tabelle 6.2 hervorgeht. Diesen Effekt kann man auch an Abbildung 6.3 erkennen. Zudem erhält man eine Approximation der jeweiligen Güte nur für die Endpunkte der Samplingperioden wie Abbildung 6.4 deutlich zeigt, wobei die Trajektorie ohne Markierungen Steuerung (6.4) verwendet. Analog zu Abbildung 6.2 werden auch in dieser Abbildung die Markierungen nur zur besseren Kennzeichnung verwendet und abermals nicht als Auswertungspunkte des sampled-data Systems. Das liegt an der Herleitung der jeweiligen Feedback-Steuerungen über Reihenentwicklungen.

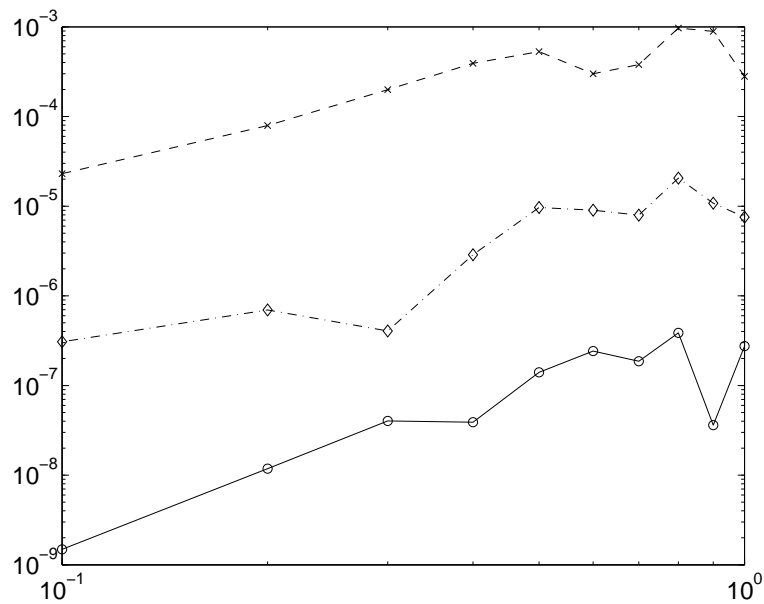


Abbildung 6.3: Differenz zwischen der zeitkontinuierlichen und der sampled-data Lösung für die sampled-data Feedbacks (6.4, x), (6.6,  $\diamond$ ) und (6.8, o).

| Zeit | Feedback (6.4) | Feedback (6.6) | Feedback (6.8) |
|------|----------------|----------------|----------------|
| 0.0  | 0. 000 000000  | 0. 00000 0000  | 0. 00000000 00 |
| 0.1  | 0. 000 023077  | 0. 00000 0307  | 0. 00000000 01 |
| 0.2  | 0. 000 079170  | 0. 00000 0696  | 0. 00000000 12 |
| 0.3  | 0. 000 199237  | 0. 00000 0405  | 0. 00000000 40 |
| 0.4  | 0. 000 392144  | 0. 00000 2873  | 0. 00000000 39 |
| 0.5  | 0. 000 529389  | 0. 00000 9666  | 0. 00000001 40 |
| 0.6  | 0. 000 300001  | 0. 00000 9036  | 0. 00000002 42 |
| 0.7  | 0. 000 379747  | 0. 00000 7952  | 0. 00000001 86 |
| 0.8  | 0. 000 968978  | 0. 00002 0587  | 0. 00000003 86 |
| 0.9  | 0. 000 891246  | 0. 00001 0807  | 0. 00000000 36 |
| 1.0  | 0. 000 279889  | 0. 00000 7533  | 0. 00000002 75 |

Tabelle 6.2: Differenz zwischen der zeitkontinuierlichen und der sampled-data Lösung mit Regler (6.4), (6.6) und (6.8) für das akademische Beispiel.



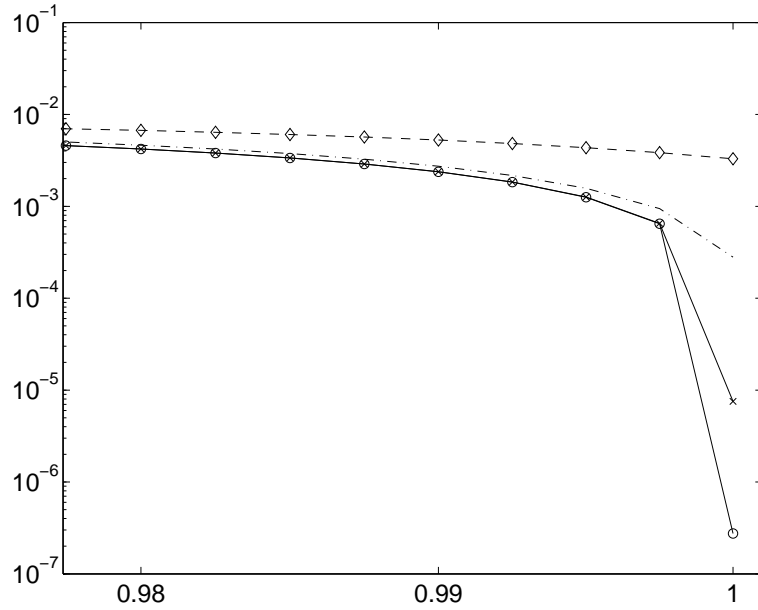


Abbildung 6.4: Differenz zwischen der zeitkontinuierlichen und der sampled-data Lösung bei  $t = 1$  für die sampled-data Feedbacks (6.2,  $\diamond$ ), (6.4), (6.6,  $\times$ ) und (6.8,  $\circ$ ).

## 6.2 Moore-Greitzer Flugzeugmotor

Als nächstes Beispiel wird das Moore-Greitzer Modell eines Flugzeugmotors betrachtet. Dieses ist aus [21] und [11] entnommen. Es ist durch die Differentialgleichung

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} -x_2 - 3x_1^2/2 - x_1^3/2 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$

gegeben. In [16] wurde  $u_0(x) = -7x_1 + 5x_2$  durch eine Backstepping-Methode als zeitkontinuierliche Feedback-Steuerung für dieses Modell hergeleitet. Mit Hilfe unserer Resultate können wir beweisen, dass kein sampled-data Feedback existiert, das  $\Delta\Phi(T, x, u_T) \leq \mathcal{O}(T^3)$  erfüllt. Um dies zu zeigen, werten wir zuerst

$$[g_0, g_1](x)L_{g_0+g_1u_0}u_0(x) = \begin{pmatrix} 32x_2 + \frac{21}{2}x_1^2 + \frac{7}{2}x_1^3 - 35x_1 \\ 0 \end{pmatrix}$$

aus. Dieser Ausdruck ist aber nicht von der Form

$$\alpha(x)g_1(x) = \begin{pmatrix} 0 \\ \alpha(x) \end{pmatrix}$$

für eine beliebige skalare Funktion  $\alpha : \mathbb{R}^2 \rightarrow \mathbb{R}$ . Folglich ist Bedingung (5.16) verletzt und es kann keine Regelung mit der oben angesprochenen Genauigkeit existieren. Dies ist eine

weitere Möglichkeit unsere Resultate anzuwenden.

Betrachtet man nun den Anfangswert  $(21, 22)^T$  und das Intervall  $[0, 1]$  mit einer Samplingperiode der Länge  $T = 0.2$  erhält man Abbildung 6.5, die das Verhalten der beiden Systeme für die einzelnen Komponenten vergleicht. An diesen Graphiken erkennt man auch gut, dass die

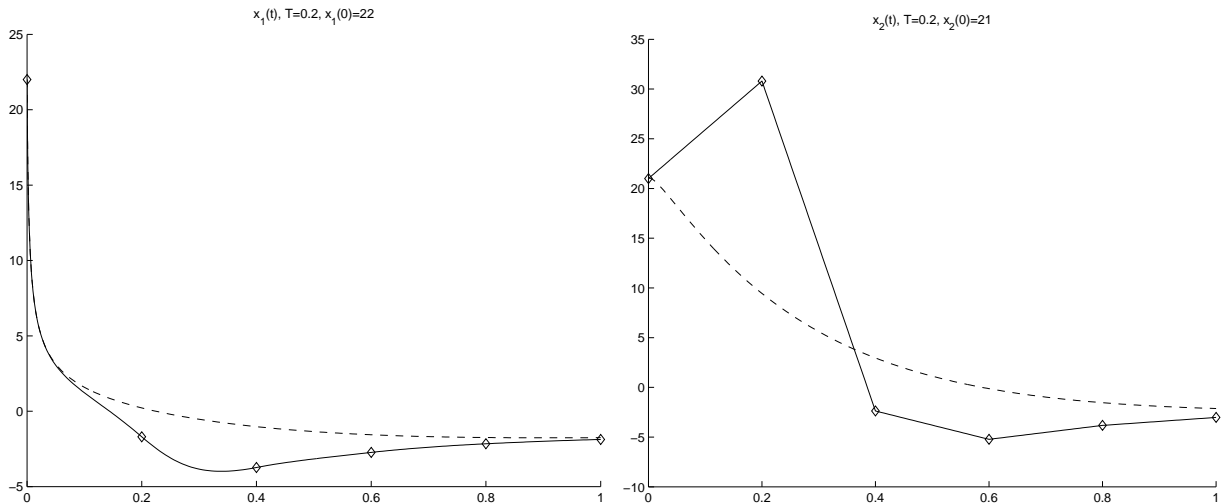


Abbildung 6.5: Vergleich der zeitkontinuierlichen und der sampled-data Lösung für das Moore-Greitzer Modell eines Flugzeugmotors.

hergeleiteten Resultate nur für hinreichend kleine  $T$  gelten. Dies zeigt die deutliche Abweichung nach dem ersten Samplingintervall, die auf den für dieses System sehr ungünstigen Anfangswert zurückzuführen ist. Dass sich die sampled-data Trajektorie der des zeitkontinuierlichen Systems nach  $1/T$  Samplingintervallen dennoch im Rahmen der gewünschten Genauigkeit annähert, liegt an den Stabilitätseigenschaften des gegebenen Systems.

# Kapitel 7

## Matlab-Simulationen zur Evaluation der Methode

In dieser Arbeit haben wir verschiedene Ergebnisse hergeleitet. Jedoch wurde stets das Verhalten des sampled-data Systems für eine gegen Null konvergierende Samplingperiode betrachtet. Dies ist für ein sampled-data System eher unrealistisch, weil man wegen Hardwarelimitationen nicht beliebig oft auswerten und darauf basierend einen neuen Kontrollwert festlegen kann. Außerdem haben wir in dieser Arbeit vorausgesetzt, dass die Steuerung unseres Abtastsystems während einer Samplingperiode konstant ist. Nun soll zuerst einmal festgelegt werden, nach welchen Kriterien die Güte der so erhaltenen Regler für die Steuerung des closed-loop Systems zu beurteilen ist. Anschließend werden zwei Optimierungsalgorithmen kurz vorgestellt und die zuvor betrachteten Beispiele mit Hilfe dieser Routinen gesteuert. Hierbei werden wir meistens unsere sampled-data Steuerung als Startwert für die Optimierung verwenden.

### 7.1 Vergleich der sampled-data Lösung mit der exakten Lösung

Um die Güte einer Lösung beurteilen zu können, benötigt man eine Referenzlösung. Diese erhält man aus dem zeitkontinuierlichen System mit der entsprechenden Steuerung  $u_0$ . Dafür wird ein Runge-Kutta-Verfahren zur numerischen Lösung der Differentialgleichung eingesetzt. Betrachtet wird dazu abermals das Moore-Greitzer Modell eines Flugzeugmotors aus [21] und [11], also das Differentialgleichungsmodell

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} -x_2 - 3x_1^2/2 - x_1^3/2 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$

mit der dazugehörigen Steuerung  $u_0(x) = -7x_1 + 5x_2$ . Es wird die Matlab-Routine `ode45` genutzt, die eine Implementierung des Runge-Kutta-Verfahrens von Dormand und Prince ist, um eine Referenzlösung zu generieren. Für nähere Informationen bzgl. Runge-Kutta-Verfahren bzw. speziell Dormand-Prince-RK5(4) siehe [7]. Die Lösung des sampled-data Systems erhält man ebenfalls mit `ode45`, allerdings wird nun nur noch zu Beginn einer Samplingperiode der Kontrollwert ausgewertet und dieser für den Rest des Samplingintervalls konstant gehalten.

## 7.2 Optimierungsansatz zur Berechnung einer Steuerung

In diesem Abschnitt wird erläutert mit welchen Optimierungsmethoden versucht wird, bessere Ergebnisse für eine Feedback-Steuerung zu gewinnen. Betrachtet man Abbildung 6.5 kann man deutliche Abweichungen zwischen der zeitkontinuierlichen und der sampled-data Lösung erkennen. Ergo muss man sich an dieser Stelle überlegen, wie diese Differenzen gemessen werden sollen. Wir haben uns dafür entschieden, für jedes Samplingintervall  $(kT, (k+1)T]$  speziell den Abstand an den Punkten  $kT + T/n, kT + 2T/n, \dots, kT + (n-1)T/n, (k+1)T$  zu wählen und die Summe der Quadrate dieser Abstände als Abweichung zu definieren, wobei  $n$  die Anzahl der Messpunkte ist. Dadurch lässt sich für jedes Samplingintervall ein nichtlineares Ausgleichsproblem aufstellen. Als Zielfunktion des Optimierungsproblems erhält man

$$\min_u F(u) = \frac{1}{2} \sum_{i=1}^n f_i(u)^2, \quad (7.1)$$

wobei  $f_i(u)$  die euklidische Norm des Abstandes der zeitkontinuierlichen von der optimierten Lösung an der Stelle  $kT + iT/n$  angibt.  $u$  ist dabei aus dem  $\mathbb{R}^m$  und folglich ist eine so genannte „best fit“-Lösung für das überbestimmte Gleichungssystem  $f(u) = 0$  gesucht, wobei  $f = (f_1 \ f_2 \ \dots \ f_n)^T$  ist. Um dieses Ausgleichsproblem zu lösen, werden das Gauß-Newton und das Levenberg-Marquardt-Verfahren zuerst erläutert und anschließend auf unsere Problemformulierung angewendet. Bevor jedoch die Optimierungsverfahren genauer betrachtet werden, wird an das Newton-Verfahren zur Lösung nichtlinearer Gleichungssysteme erinnert. Die folgenden Ausführungen basieren auf [18], wobei für das Newton bzw. Gauß-Newton-Verfahren zusätzlich [2] und [17] benutzt worden sind. Für nähere Informationen bzgl. des Levenberg-Marquardt-Verfahrens siehe auch [25].

Zur Formulierung der Verfahren benötigt man den Gradienten

$$g(u) := \nabla F(u) = \left[ \frac{\partial F}{\partial u_1} \quad \frac{\partial F}{\partial u_2} \quad \dots \quad \frac{\partial F}{\partial u_m} \right]^T$$

von  $F(u)$  und die Hessematrix

$$G(u) := \left[ \frac{\partial^2 F}{\partial u_i \partial u_j} \right]_{\substack{i=1,\dots,m \\ j=1,\dots,m}},$$

wobei  $F = F(u)$  verwendet wurde. Für ein lokales Minimum von  $F(u)$  bei  $u^*$  ist es hinreichend, dass

- $g(u^*) = 0$  und
- $s^T G(u^*) s > 0 \quad \forall s$

gilt. Für ein globales Minimum benötigt man weitere Bedingungen an  $F$  wie zum Beispiel strikte Konvexität. Die zweite Bedingung ist äquivalent zur positiven Definitheit der Matrix  $G(u^*)$ , die zum Beispiel gegeben ist, falls alle Eigenwerte echt größer als Null sind, siehe hierzu auch [3]. Wir setzen deshalb für diesen Abschnitt die Voraussetzung, dass  $F(u)$  eine positiv definite Hessematrix besitzt. Nun funktioniert der allgemeine Algorithmus zur Bestimmung einer lokalen Minimallösung wie folgt:

- i) Bestimme eine Abstiegsrichtung  $s^{(k)}$ , d.h.  $s^{(k)}$  erfüllt die Ungleichung

$$g(u^{(k)})^T s^{(k)} < 0,$$

weil  $g(u^{(k)})$  in die Richtung des steilsten Anstiegs zeigt (siehe [5]).

- ii) Bestimme eine Schrittweite  $\alpha_k$  mit

$$F(u^{(k)} + \alpha_k s^{(k)}) < F(u^{(k)}).$$

- iii) Setze  $u^{(k+1)} = u^{(k)} + \alpha_k s^{(k)}$  und führe einen Konvergenztest durch.

Für weitere Details bzgl. des Konvergenztests siehe [18, Seite 242]. Die Schrittweitenbestimmung soll einen hinreichenden Abstieg bzgl. des Funktionswertes sicherstellen. Hierzu eignet sich unter anderem die Goldstein-Armijo Bedingung [25]. An dieser Stelle soll aber speziell das Newton-Verfahren vorgestellt werden, das eine quadratische Approximation zugrunde legt. Diese basiert auf der  $m$ -dimensionalen Taylorentwicklung, die für  $F$  bzw.  $\nabla F$

$$F(u + s) = F(u) + s^T g(u) + \frac{1}{2} s^T G(u) s + \dots \quad \text{bzw.} \quad (7.2)$$

$$\nabla F(u + s) = g(u) + G(u) s + \dots \quad (7.3)$$

liefert. Entsprechend ist

$$F(u^{(k)} + s) \approx q^{(k)}(s) := F(u^{(k)}) + s^T g(u^{(k)}) + \frac{1}{2} s^T G(u^{(k)}) s$$

eine quadratische Approximation. Das so erhaltene  $q^{(k)}(s)$  wird minimal, falls  $G(u^{(k)})s = -g(u^{(k)})$  gilt. Die so bestimmte Richtung  $s^{(k)}$  wird Newton-Richtung genannt. Nahe der Lösung definiert man einfach  $u^{(k+1)}$  als  $u^{(k)} + s^{(k)}$ . Befindet man sich jedoch weiter entfernt vom Optimum wird erneut eine „line search“ benötigt. Zu beachten ist dabei, dass  $s^{(k)}$  eine Abstiegsrichtung ist.

Aufbauend auf dem Newton-Verfahren kann man das Gauß-Newton-Verfahren definieren. Hierbei wird insbesondere die spezielle Struktur von (7.1) ausgenutzt. Folglich gilt

$$\nabla F(u) = J^T f(u), \quad (7.4)$$

wobei  $J$  die  $n \times m$  Jacobi-Matrix von  $f$  bezüglich  $u$  ist und

$$G(u) = J^T J + \sum_{i=1}^n f_i(u) H_i(u), \quad (7.5)$$

wobei  $H_i(u)$  die Hessematrix von  $f_i(u)$  ist. Ausgehend von einem Punkt  $u^{(k)}$  kann man nun einen Newton-Schritt ausführen:

$$G(u^{(k)}) \underbrace{\delta u^{(k)}}_{\text{Newton-Schritt}} = -\nabla F(u^{(k)}). \quad (7.6)$$

Nahe des Optimums kann man (7.5) approximieren durch  $G(u) = J^T J$ . Setzt man das und (7.4) in (7.6) ein, erhält man

$$(J(u^{(k)}))^T J(u^{(k)}) \delta u^{(k)} = -(J(u^{(k)}))^T f(u^{(k)}). \quad (7.7)$$

Eine alternative Herleitung dieser Gleichung bekommt man, indem man die einzelnen  $f_i(u)$  linearisiert. (7.7) stellt die Normalengleichungen des überbestimmten linearen Systems  $J(u^{(k)}) \delta u^{(k)} = -f(u^{(k)})$  dar. Nun ist man in der Lage den Gauß-Newton Algorithmus zu formulieren.

### Algorithmus 7.1 (Gauß-Newton)

- i) Wähle einen Startwert  $u^{(0)}$
- ii) Für  $k = 0, 1, 2, \dots, k_{max}$  berechne  $f(u^{(k)})$ ,  $F(u^{(k)})$  und  $J(u^{(k)})$  und bestimme  $s^{(k)}$
- iii) Minimiere  $F(u^{(k)} + \alpha s^{(k)})$  bzgl.  $\alpha$ .
- iv) Setze  $u^{(k+1)} = u^{(k)} + \alpha_k s^{(k)}$
- v) Konvergenztest; ist die maximale Iterationszahl erreicht? Falls eines der Abbruchkriterien erfüllt ist, beende den Algorithmus, ansonsten gehe zu ii).

**Bemerkung 7.2** Um die Abstiegsrichtung  $s^{(k)}$  aus (7.7) zu bestimmen, sollte man eine QR-Faktorisierung benutzen, für nähere Informationen bzgl. der QR-Zerlegung siehe [6].

Als nächstes wird ein weiterer Algorithmus zur Lösung eines kleinsten Quadrate-Problems vorgestellt, der zu den Trust-Region-Algorithmen gehört, siehe hierzu auch [17]. Der wesentliche Unterschied zum Gauß-Newton Algorithmus liegt in der Bestimmung der Abstiegsrichtung. Dies geschieht beim Levenberg-Marquardt Algorithmus über

$$[(J(u^{(k)}))^T J(u^{(k)}) + \lambda_k I_k] \delta u^{(k)} = -(J(u^{(k)}))^T f(u^{(k)}),$$

wobei  $\lambda_k$  eine Konstante ist, deren Wert implizit die Größe von  $\delta u^{(k)}$  kontrolliert. Genauer ist  $\lambda_k$  der Lagrange-Multiplikator des Problems:

$$\min_{\delta u^{(k)}} L(\delta u^{(k)}) = \min_{\delta u^{(k)}} \| f(u^{(k)}) + J(u^{(k)})\delta u^{(k)} \| \quad (7.8)$$

unter der Nebenbedingung  $\| \delta u^{(k)} \| \leq \Delta_k$ .  $L(\delta u^{(k)})$  ist dabei die lineare Approximation von  $F(u^{(k)} + \delta u^{(k)})$  bzgl.  $u^{(k)}$ . Für nähere Informationen bzgl. Lagrange-Multiplikatoren siehe [5] und [17]. Es ergibt sich also entweder

$$\| \delta u^{(k)} \| < \Delta_k \quad \text{und} \quad \lambda_k = 0$$

oder

$$\| \delta u^{(k)} \| = \Delta_k \quad \text{und} \quad \lambda_k \geq 0.$$

Also bestimmt  $\Delta_k$   $\lambda_k$ . Ist  $\lambda_k$  sehr groß, betont das Verfahren den steilsten Abstieg. Dies kann zu langsamer Konvergenz führen. Dafür überwindet der Levenberg-Marquardt Algorithmus Probleme bei fehlender Spaltenregularität des Gauß-Newton Algorithmus, die allerdings in unserem Fall,  $m = 1$ , keine Rolle spielen. Für nähere Details sei auf die bereits erwähnte Literatur verwiesen.

## 7.3 Moore-Greitzer Modell: Vergleich der sampled-data Lösung mit der optimierten Lösung

Nun können wir testen, ob eine auf einem der beiden in Abschnitt 7.2 basierende kleinste Quadrate-Optimierung bessere Resultate als die sampled-data Feedback-Regelung für das Moore-Greitzer Modell liefert. Dafür betrachten wir (7.1) mit  $n = 4$  und setzen erneut die Länge des Samplingintervalls auf  $T = 0.2$ . In Matlab sind zwei Methoden für eine Linesearch implementiert:

- cubicpoly und
- quadcubic.

Die Routine `cubicpoly` sollte man wählen, wenn die Auswertung von Ableitungen nicht zu teuer ist und man verhältnismäßig wenige Funktionsauswertungen haben möchte. `Quadcubic` ist hingegen vorzuziehen, falls Funktionsauswertungen relativ günstig sind. Wir werden zuerst mit `cubicpoly` als Optimierungsparameter arbeiten, weil sich so die Veränderungen der Lösung abhängig vom Optimierungsaufwand besser darstellen lässt. Für nähere Erläuterungen siehe Bemerkung 7.3. Bezüglich der Algorithmen werden zuerst Graphiken für den Gauß-Newton Algorithmus dargestellt. Hierbei limitiert die maximale Iterationszahl pro Optimierungsschritt den Aufwand des Verfahrens. Zusätzlich werden zwei verschiedene Lösungstrajektorien generiert. Eine, die keinen Startwert verwendet beziehungsweise Null als solchen vorgegeben bekommt und eine die das oben erwähnte  $u_0(x)$  als Startwert verwendet, also die `sampled-data` Feedback-Steuerung. Implementiert wird das durch den Startwert 1. Es wird folglich die Steuerung in Relation zur `sampled-data` Feedback Regelung dargestellt, siehe hierzu auch Tabelle B.3. Die entsprechenden Abbildungen sind Abbildung 7.1, Abbildung 7.2 und Abbildung 7.3.

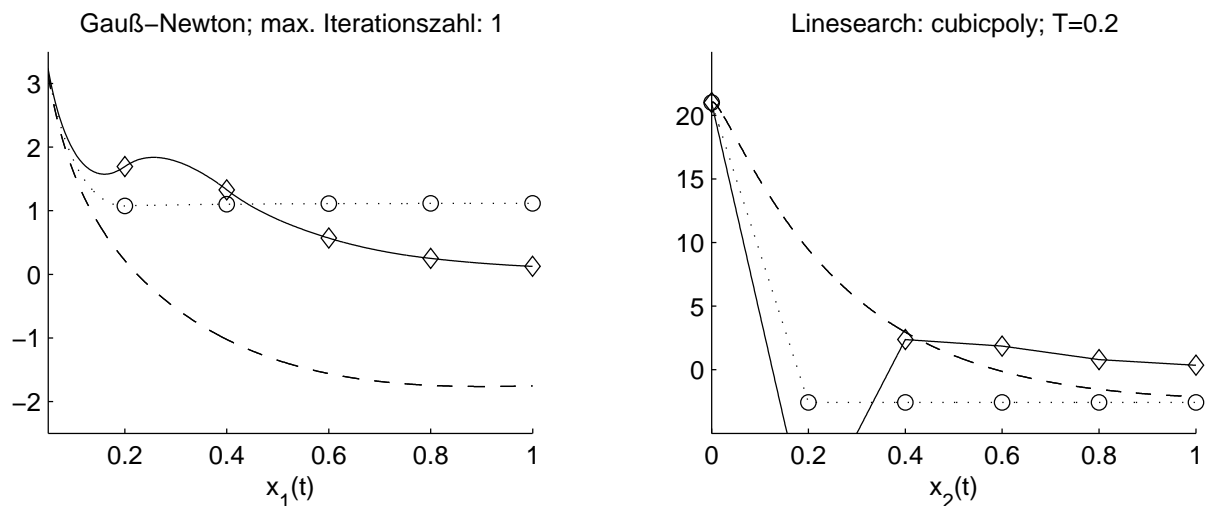


Abbildung 7.1: Vergleich der zeitkontinuierlichen mit den optimierten Lösungen. Ohne Anfangswert (o) bzw. mit `sampled-data` Feedback als Anfangswert ( $\diamond$ ).

Klar erkennbar ist dabei, dass bereits in Abbildung 7.3 die Optimierung für das Auge des Betrachters erfolgreich abgeschlossen ist. Es wurden dafür 3 Iterationen und lediglich 8 Funktionsauswertungen benötigt. Lässt man die Optimierungsroutine weiter laufen, stellt sich keine signifikante Verbesserung mehr ein. Allerdings werden für die einzelnen Intervalle bis zum Abbruch der Routine deutlich mehr Funktionsauswertungen benötigt. Ohne `sampled-data` Feedback als Startwert sind dies 10, 12, 14, 16 und 46 beziehungsweise mit `sampled-data`



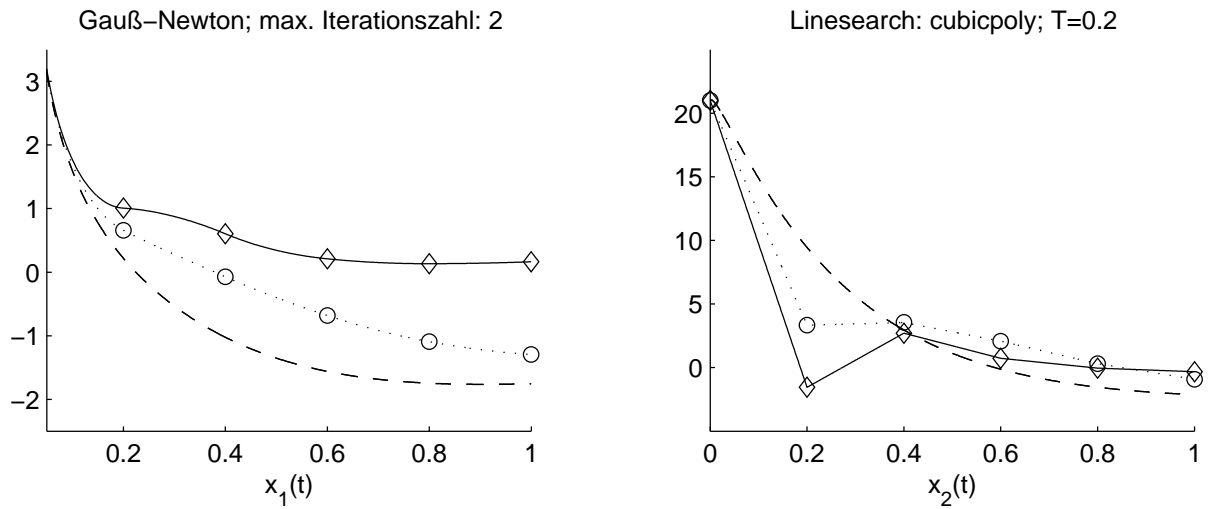


Abbildung 7.2: Vergleich der zeitkontinuierlichen mit den optimierten Lösungen. Ohne Anfangswert (o) bzw. mit sampled-data Feedback als Anfangswert ( $\diamond$ ).

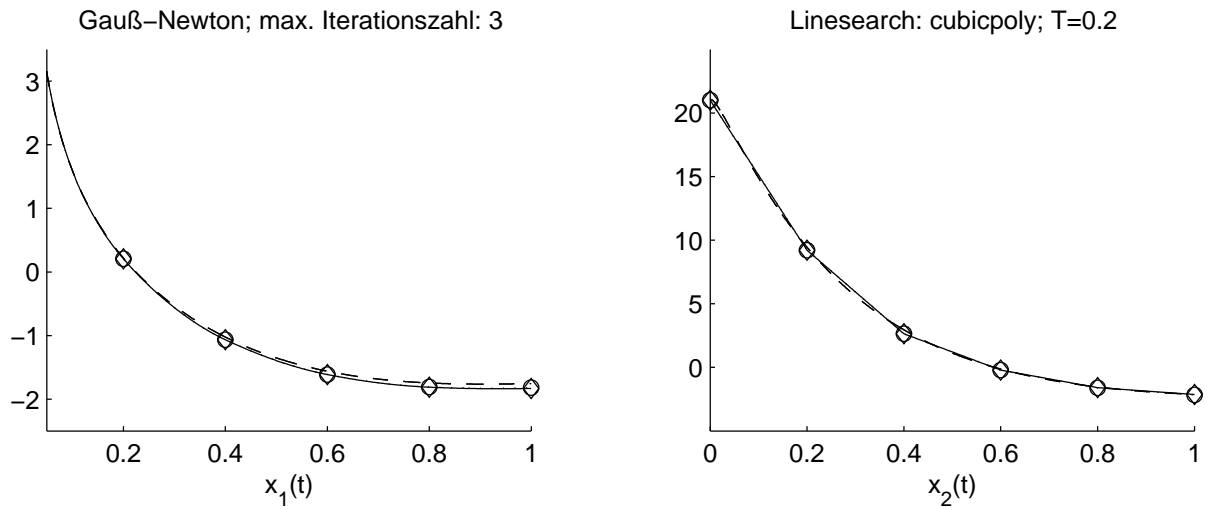


Abbildung 7.3: Vergleich der zeitkontinuierlichen mit den optimierten Lösungen. Ohne Anfangswert (o) bzw. mit sampled-data Feedback als Anfangswert ( $\diamond$ ).

Feedback als Anfangswert 12, 12, 14, 16 und 20. Den Wert der einzelnen Residuen entnehme man Tabelle 7.1.

Im Vergleich dazu sei nun die gleiche Vorgehensweise für den Levenberg-Marquardt Algorithmus durchgeführt. Dieser benötigt 4 bzw. 5 Iterationen bis ein vergleichbares Resultat erzielt ist, siehe hierzu Abbildungen 7.4 - 7.8.

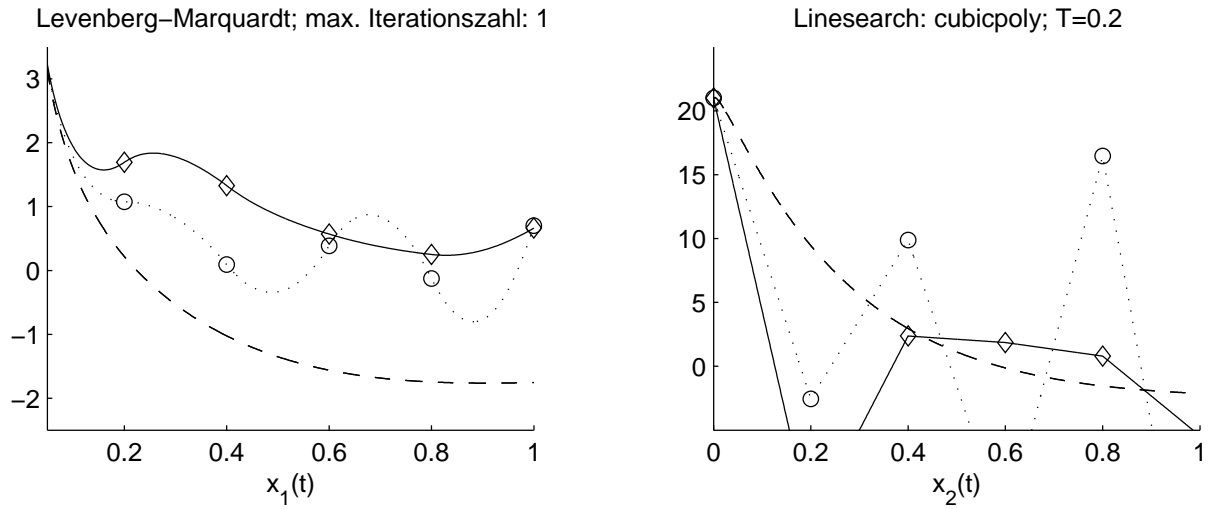


Abbildung 7.4: Vergleich der zeitkontinuierlichen mit den optimierten Lösungen. Ohne Anfangswert (o) bzw. mit sampled-data Feedback als Anfangswert ( $\diamond$ ).

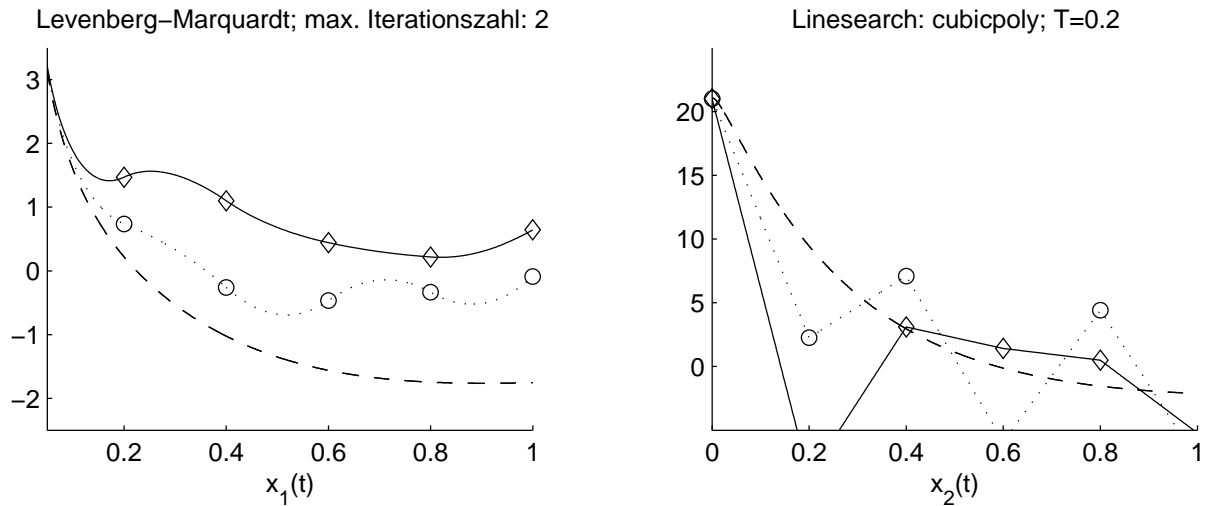


Abbildung 7.5: Vergleich der zeitkontinuierlichen mit den optimierten Lösungen. Ohne Anfangswert (o) bzw. mit sampled-data Feedback als Anfangswert ( $\diamond$ ).

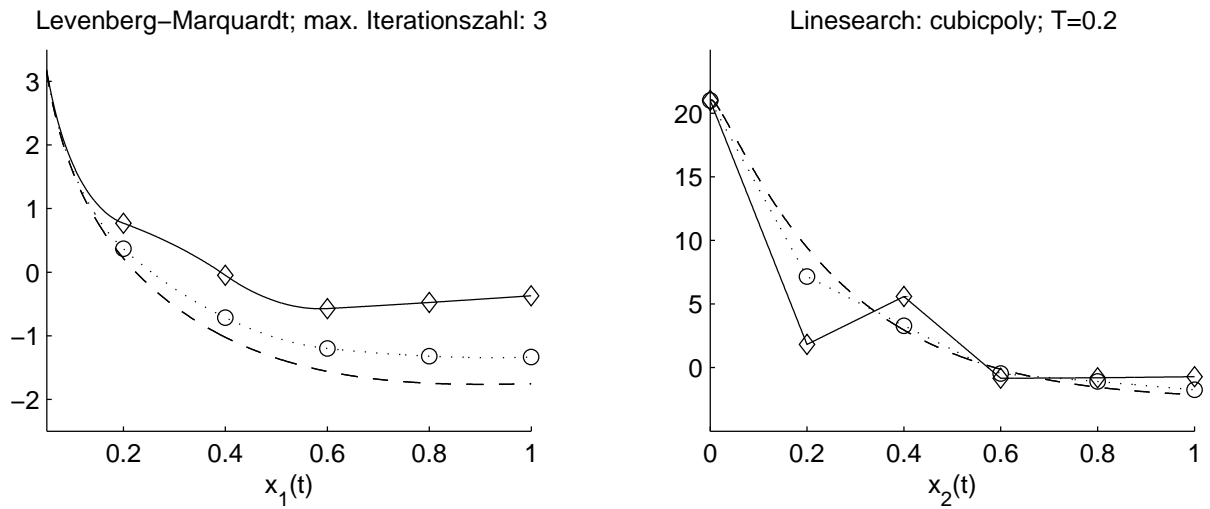


Abbildung 7.6: Vergleich der zeitkontinuierlichen mit den optimierten Lösungen. Ohne Anfangswert (o) bzw. mit sampled-data Feedback als Anfangswert ( $\diamond$ ).

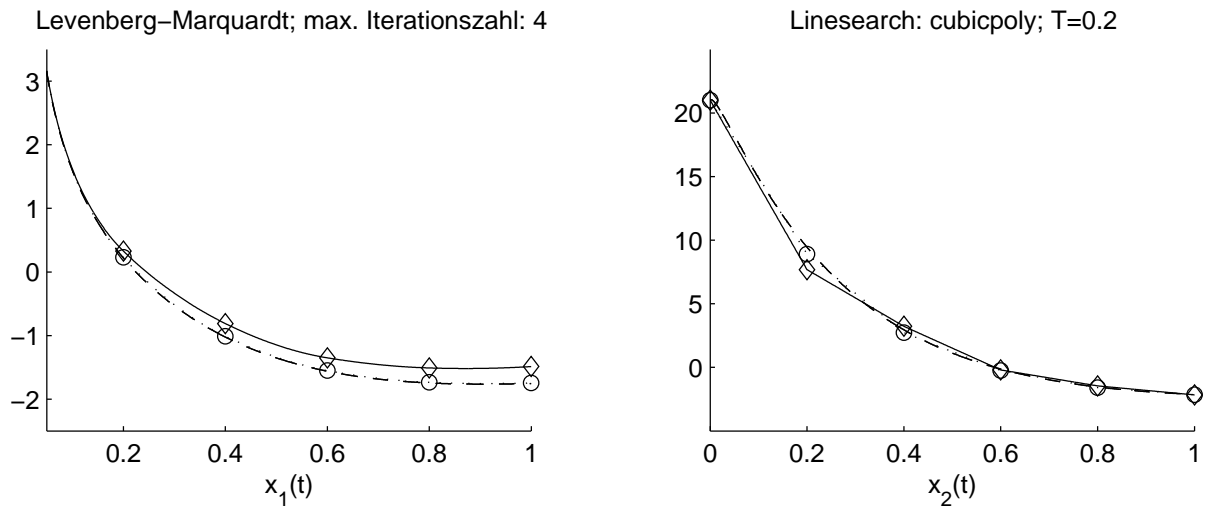


Abbildung 7.7: Vergleich der zeitkontinuierlichen mit den optimierten Lösungen. Ohne Anfangswert (o) bzw. mit sampled-data Feedback als Anfangswert ( $\diamond$ ).

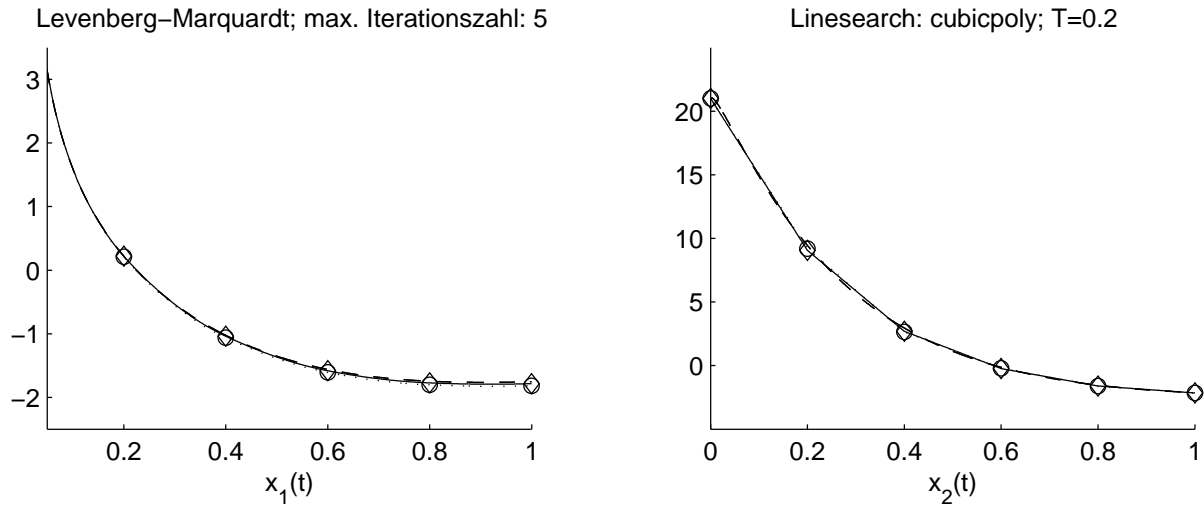


Abbildung 7.8: Vergleich der zeitkontinuierlichen mit den optimierten Lösungen. Ohne Anfangswert (o) bzw. mit sampled-data Feedback als Anfangswert ( $\diamond$ ).

Analytisch erfolgt der Vergleich durch die Residuen der Optimierungsverfahren. Diese sind für beide Algorithmen und für die jeweiligen Anfangswerte in Tabelle 7.1 dargestellt.

| max. zugelassene<br>Iterationen | Gauß-Newton    |                 | Levenberg-Marquardt |                 |
|---------------------------------|----------------|-----------------|---------------------|-----------------|
|                                 | Anfangswert: 0 | Anfangswert: SD | Anfangswert: 0      | Anfangswert: SD |
| 0                               | 7134.3478384   | 1416.5737152    | 7134.3478384        | 1416.5737152    |
| 1                               | 664.1266187    | 1399.7981057    | 1892.0175492        | 1392.7404852    |
| 2                               | 135.8739288    | 380.9615298     | 291.0995045         | 959.3003448     |
| 3                               | 0.5179516      | 0.5252419       | 13.4147175          | 167.9652881     |
| 4                               | 0.5195589      | 0.5231933       | 0.5079698           | 6.5964399       |
| 5                               | 0.5192796      | 0.5195870       | 0.4983828           | 0.4386300       |
| 6                               | 0.5193534      | 0.5193629       | 0.5186704           | 0.5117832       |
| beliebig                        | 0.5193499      | 0.5193448       | 0.5193426           | 0.5193424       |

Tabelle 7.1: Vergleich der Residuen der beiden Algorithmen. Linesearch: cubicpoly.

Die Veränderung der Optimalsteuerung jeweils gemessen gegenüber dem sampled-data Wert ist in Tabelle B.3 ersichtlich. Auffällig dabei ist, dass erst der erste Optimierungswert fixiert sein muss, bevor sich die restlichen Werte langsam einpendeln. Das ist auch verhältnismäßig leicht zu erklären, weil eine Veränderung des ersten Optimierungswertes den Startwert für das zweite Optimierungsintervall verändert. Dieser Prozess setzt sich bis zum letzten Intervall

7.3. MOORE-GREITZER MODELL: SAMPLED-DATA UND OPTIMIERTE LÖSUNG 83

iterativ fort. Als nächstes seien die Ergebnisse für die Linesearch-Methode quadcubic dargestellt. Siehe hierzu Abbildungen 7.9 bis 7.12, wobei die ersten beiden für den Gauß-Newton und die anderen beiden für den Levenberg-Marquardt Algorithmus sind.

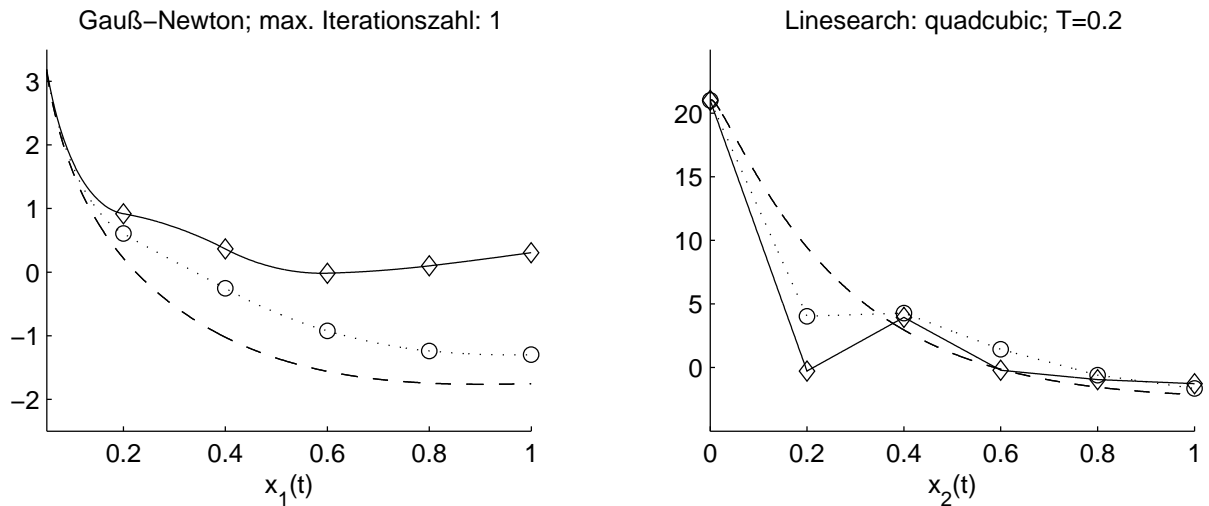


Abbildung 7.9: Vergleich der zeitkontinuierlichen mit den optimierten Lösungen. Ohne Anfangswert (o) bzw. mit sampled-data Feedback als Anfangswert ( $\diamond$ ).

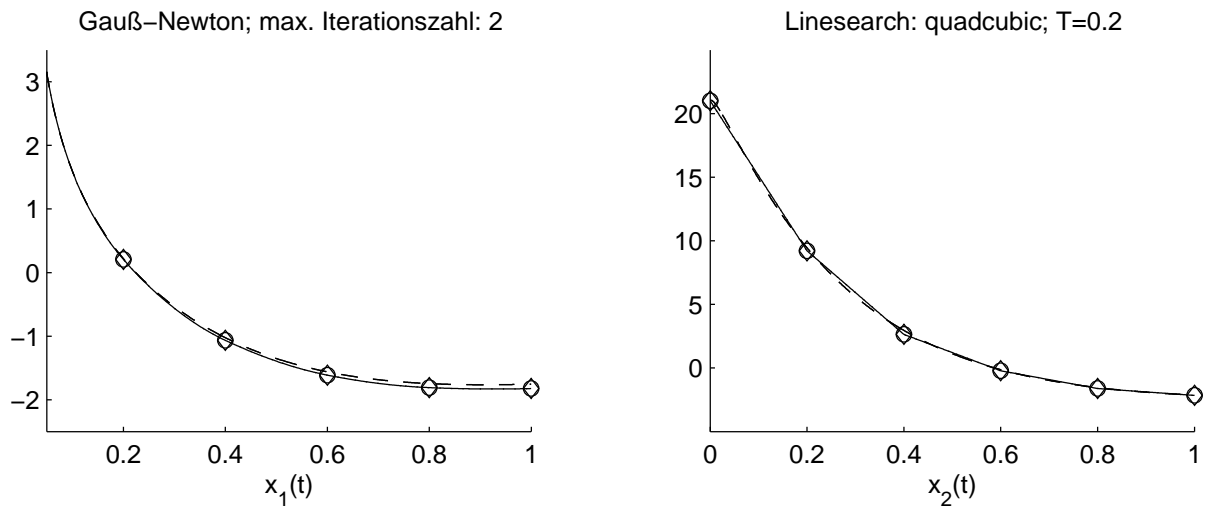


Abbildung 7.10: Vergleich der zeitkontinuierlichen mit den optimierten Lösungen. Ohne Anfangswert (o) bzw. mit sampled-data Feedback als Anfangswert ( $\diamond$ ).

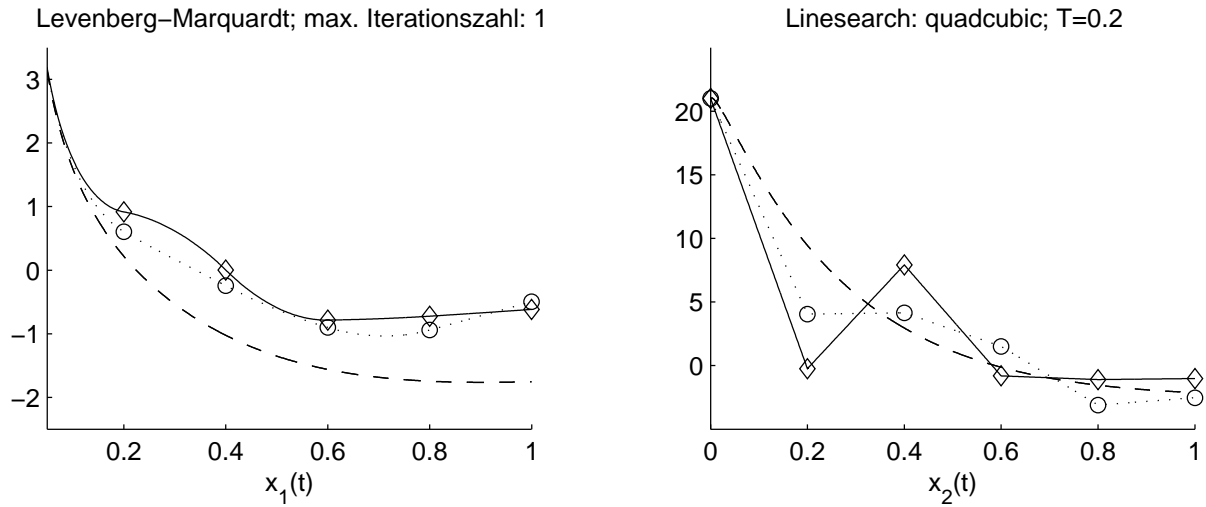


Abbildung 7.11: Vergleich der zeitkontinuierlichen mit den optimierten Lösungen. Ohne Anfangswert (o) bzw. mit sampled-data Feedback als Anfangswert ( $\diamond$ ).

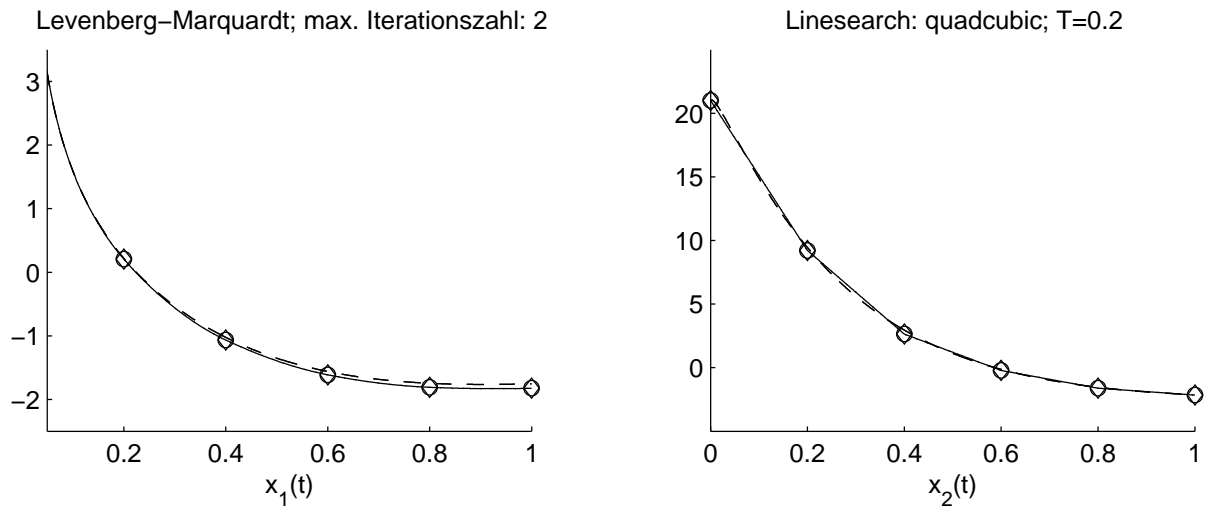


Abbildung 7.12: Vergleich der zeitkontinuierlichen mit den optimierten Lösungen. Ohne Anfangswert (o) bzw. mit sampled-data Feedback als Anfangswert ( $\diamond$ ).

Auch hierzu seien die Residuen dargestellt, siehe Tabelle 7.2.

**Bemerkung 7.3** *Es erscheint anhand der Graphiken so, als ob die „line search“ quadcubic wesentlich effizienter als cubicpoly ist. Allerdings benötigt quadcubic für die erste Iteration*

| max. zugelassene<br>Iterationen | Gauß-Newton    |                 | Levenberg-Marquardt |                 |
|---------------------------------|----------------|-----------------|---------------------|-----------------|
|                                 | Anfangswert: 0 | Anfangswert: SD | Anfangswert: 0      | Anfangswert: SD |
| 0                               | 7134.3478384   | 1416.5737152    | 7134.3478384        | 1416.5737152    |
| 1                               | 90.8651802     | 277.4450411     | 93.386067           | 270.9418438     |
| 2                               | 0.5193397      | 0.5193377       | 0.5193457           | 0.5193374       |
| 3                               | 0.5193436      | 0.5193377       | 0.5193457           | 0.5193374       |
| beliebig                        | 0.5193436      | 0.5193377       | 0.5193457           | 0.5193374       |

Tabelle 7.2: Vergleich der Residuen der beiden Algorithmen. Linearsuch: quadcubic.

sieben bis acht Funktionsauswertungen. Cubicpoly benötigt aber für drei Iterationen lediglich acht Funktionsauswertungen, liefert folglich gemessen am Aufwand vergleichbare Ergebnisse und gibt uns die Chance die einzelnen Veränderungen bzgl. des Aufwandes besser zu messen. Siehe hierzu auch die Tabellen B.4 und B.5.

**Bemerkung 7.4** Klar ersichtlich aus Tabelle 7.1 und Tabelle 7.2 ist, dass die Matlab-Optimierung unabhängig vom eingesetzten Algorithmus, der verwendeten „line search“ und des jeweiligen Anfangswertes den Wert 0.5193... als Optimum liefert. Allerdings kann man bereits in Tabelle 7.1 erkennen, dass es sich dabei um ein lokales Minimum handelt, weil zum Beispiel der Levenberg-Marquardt Algorithmus mit maximal fünf Iterationen bessere Werte liefert. Dies liegt daran, dass die Samplingintervalle jeweils isoliert optimiert werden und das jeweils vorausgegangene Intervall lediglich den Startwert für das aktuelle Intervall liefert. Dies geschieht aber nicht in Form eines Optimierungsparameters, sondern in Form eines fixierten Wertes.

Eine Möglichkeit diese Probleme anzugehen, könnte eine Gewichtung der einzelnen Optimierungspunkte sein. Berücksichtigt man den letzten Messpunkt eines jeden Intervalls mit doppelter Wertigkeit erhält man aber Werte in Höhe von 0.7355..., also eine Verschlechterung von über vierzig Prozent. Gewichtet man jedoch den vierten und letzten Punkt nur halb so stark wie alle anderen, erhält man 0.4569, also eine Verbesserung von ca. zwölf Prozent gegenüber dem lokalen Minimum. Man kann also durch verschiedene Gewichtungen weiteren Einfluss auf die Optimierung nehmen.

In den Abbildungen 7.1 bis 7.12 wurde jeweils die exakte Lösung zusammen mit den optimierten Trajektorien bezüglich einem der beiden Algorithmen und vorgegebener Iterationszahl mit beziehungsweise ohne sampled-data Anfangswert dargestellt. Abschließend sei für den Levenberg-Marquardt Algorithmus mit sampled-data Optimierungsstartwert die Entwicklung der Lösungstrajektorien abhängig von der Iterationszahl in einem Plot dargestellt. Zusätzlich ist auch die sampled-data Lösung eingezeichnet. Beschränkt haben wir uns in dieser Graphik auf eine beziehungsweise drei Iterationen pro Optimierungsschritt.

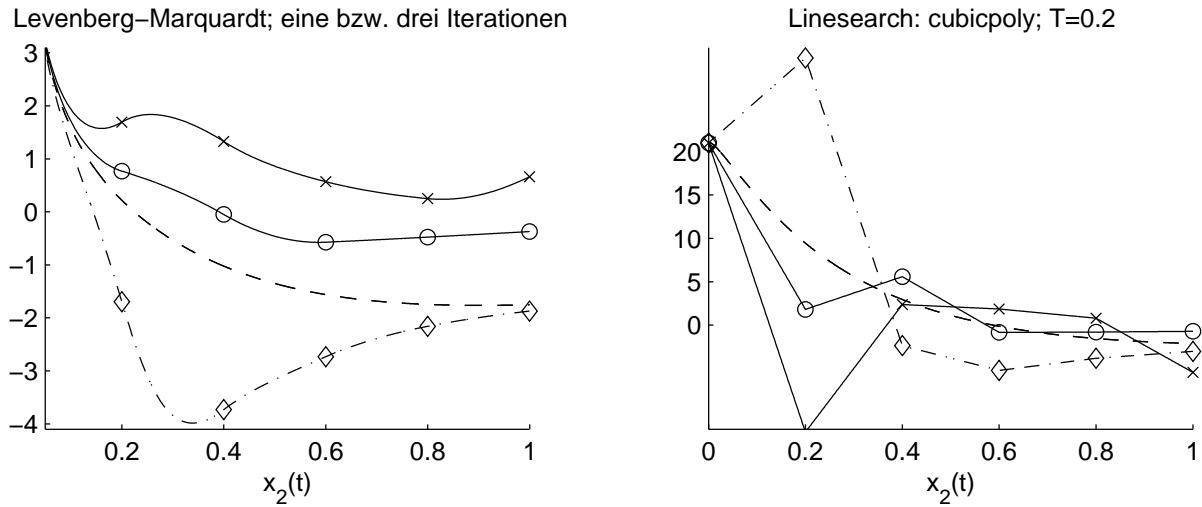


Abbildung 7.13: Vergleich der sampled-data Lösung ( $\diamond$ ) mit optimierten Trajektorien. Levenberg-Marquardt mit einer (x) beziehungsweise drei (o) Iterationen.

## 7.4 Akademisches Beispiel: Vergleich der sampled-data Lösung mit der optimierten Lösung

Wir haben in diesem Kapitel festgelegt, wie die Güte einer Approximation zu beurteilen ist. Entsprechend sind wir mittlerweile in der Lage, die sampled-data Trajektorien für das akademische Beispiel aus Kapitel 6 zu beurteilen. Sicherlich ist der Wert des Residuums abhängig von der Zahl  $n$  der Messpunkte. Tabelle 7.3 zeigt die Residuen für  $n = 4$  beziehungsweise  $n = 8$ .

| verwendete<br>Kontrollfunktion | Auswertungspunkte |                  |
|--------------------------------|-------------------|------------------|
|                                | 4                 | 8                |
| $u_T^0$                        | 0.08645977188801  | 0.16631921129912 |
| $u_T^1$                        | 0.00613403488105  | 0.01135582541741 |
| $u_T^2$                        | 0.00621196891899  | 0.01259157821518 |
| $u_T^3$                        | 0.00411649253091  | 0.00827160007994 |
| $u_T^4$                        | 0.00374212682363  | 0.00750272236082 |
| $u_T^6$                        | 0.00383914059746  | 0.00770830054058 |
| $u_T^8$                        | 0.00383229950894  | 0.00769389526756 |

Tabelle 7.3: Residuen der sampled-data Lösung für vier bzw. acht Auswertungspunkte.



Im Gegensatz zu Abschnitt 6.1 werden also nicht nur die Abstände an den Endpunkten der einzelnen Samplingintervalle berücksichtigt, sondern auch drei beziehungsweise sieben zusätzliche Zwischenpunkte. Damit ist es wesentlich besser möglich, die Güte der gesamten Trajektorie im Vergleich zu der gegebenen Referenzlösung zu beurteilen. Klar ersichtlich ist dabei der signifikante Fortschritt von (6.3) gegenüber (6.2). Die beste Steuerung ist nach Tabelle 7.3 (6.6) und das sogar unabhängig von der gewählten Zahl der Messpunkte. Die Frage, mit der wir uns in diesem Abschnitt hauptsächlich beschäftigen wollen, ist, ob man innerhalb weniger Iterationsschritte durch eine Nachoptimierung mit dem Levenberg-Marquardt Algorithmus die Lösung deutlich verbessern kann. Ausgehen wollen wir bei diesem Vorhaben normalerweise von (6.6) und zwei Iterationsschritten. Allerdings werden in den Tabellen auch häufig Werte für (6.3) als Startwert der Nachoptimierung angegeben, sowie die Werte für genau eine beziehungsweise drei Iterationen. Hierzu betrachten wir zuerst Abbildung 7.14, welche die sampled-data Lösung (6.6) im Vergleich zu der optimierten Lösung darstellt.

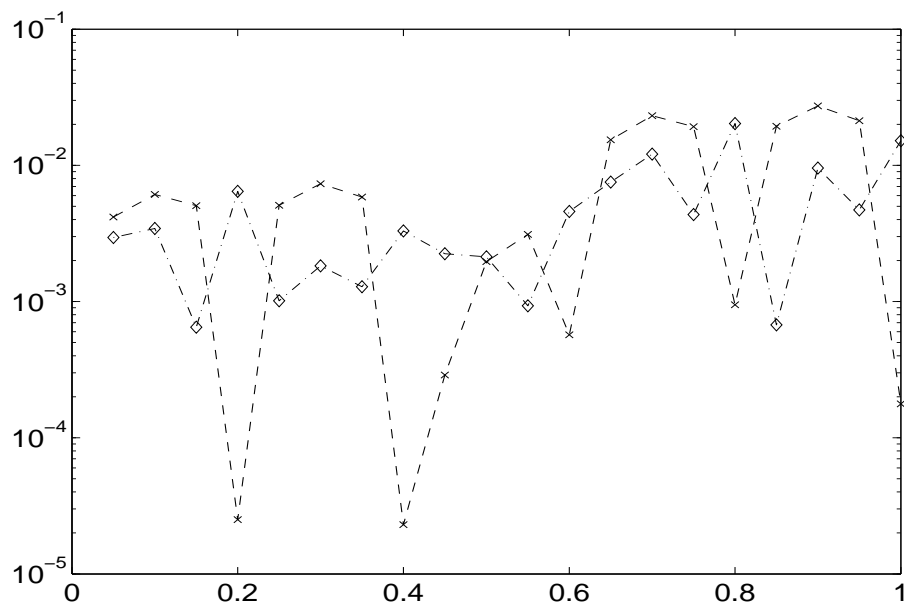


Abbildung 7.14: Abstand der sampled-data (x) beziehungsweise der optimierten Lösung (o) von der exakten Trajektorie mit Samplingperiode  $T = 0.2$ . Optimierung ausgehend von (6.6) mit maximal zwei Iterationen und vier Messpunkten pro Intervall.

Beobachten kann man dabei, dass die optimierte Lösung zwar an den Endpunkten der Samplingintervalle schlechter ist, aber ansonsten an fast allen Messpunkten deutlich besser ist als die sampled-data Lösung. So hat sich das Residuum bereits nach zwei Iterationen mehr als halbiert, wie Tabelle 7.4 zeigt. Diese Aussage gilt nach Tabelle 7.5 sogar, wenn man die Nachoptimierung ausgehend von (6.3) startet. In den beiden angesprochenen Tabellen

befinden sich zudem die Residuenwerte für eine auf anderen Gewichtungen basierende Optimierung. Abermals zeigt sich, dass es mehr Sinn macht, den letzten Punkt weniger stark zu gewichten als diesen doppelt zu berücksichtigen. Diese Beobachtung ist für dieses Beispiel gut erklärbar, weil die Abweichung von der extremen Beachtung der Randpunkte der sampled-data Steuerung bereits zu deutlich besseren Ergebnissen geführt hat.

| Iterationen | Standard         | letzter Punkt doppelt | letzter Punkt halb |
|-------------|------------------|-----------------------|--------------------|
| 1           | 0.01472553873385 | 0.01298385751520      | 0.02342999746111   |
| 2           | 0.00179858736833 | 0.00219517778928      | 0.00170335520495   |
| 3           | 0.00152966119645 | 0.00196482692207      | 0.00141918609674   |

Tabelle 7.4: Residuen der Nachoptimierung ausgehend von (6.3) mit verschiedener Gewichtung und vier Auswertungspunkten.

| Iterationen | Standard         | letzter Punkt doppelt | letzter Punkt halb |
|-------------|------------------|-----------------------|--------------------|
| 1           | 0.00193677521675 | 0.00160886345341      | 0.01176776259481   |
| 2           | 0.00152392430448 | 0.00195786137622      | 0.00142831145951   |
| 3           | 0.00152945427986 | 0.00196462318347      | 0.00141902864750   |

Tabelle 7.5: Residuen der Nachoptimierung ausgehend von (6.6) mit verschiedener Gewichtung und vier Auswertungspunkten.

Analog zu Abbildung 7.14 und den Tabellen 7.4 und 7.5 kann man die gleichen Rechnungen mit acht statt vier Messpunkten je Samplingintervall durchführen. Es ergeben sich Abbildung 7.15, sowie die Tabellen B.1 und B.2. Bemerkenswert ist vor allem, dass der optimierte Wert nach zwei Iterationen, unabhängig ob man von (6.3) oder (6.6) startet, deutlich geringer ist als der Wert aus der linken Spalte von Tabelle 7.3, obwohl dieser lediglich halb so viele Messpunkte berücksichtigen muss. Der nicht optimierte Wert für (6.6) mit acht Messpunkten liegt bei 0.00750272236082, also ca. dreimal so hoch wie die Werte der beiden angesprochenen Tabellen.

**Bemerkung 7.5 (Gewichtung)** *Anzumerken bezüglich der Gewichtung bleibt aber, dass bei einer auf der sampled-data Feedback Steuerung basierenden Nachoptimierung zumindest zwei Optimierungsschritte benötigt werden, damit der Schwerpunktwechsel weg von den Intervallendpunkten hin zur Trajektorienorientierung gelingt. Falls man nur einen Optimierungsschritt durchführen kann, sollte man ausgehend von diesem Startwert für die Steuerung die Betonung auf den Endpunkt<sup>1</sup> des Samplingintervalls legen.*

<sup>1</sup>tatsächlich wird in der durchgeführten Optimierung das letzte Viertel der Messpunkte und nicht nur der Intervallendpunkt doppelt gewichtet.

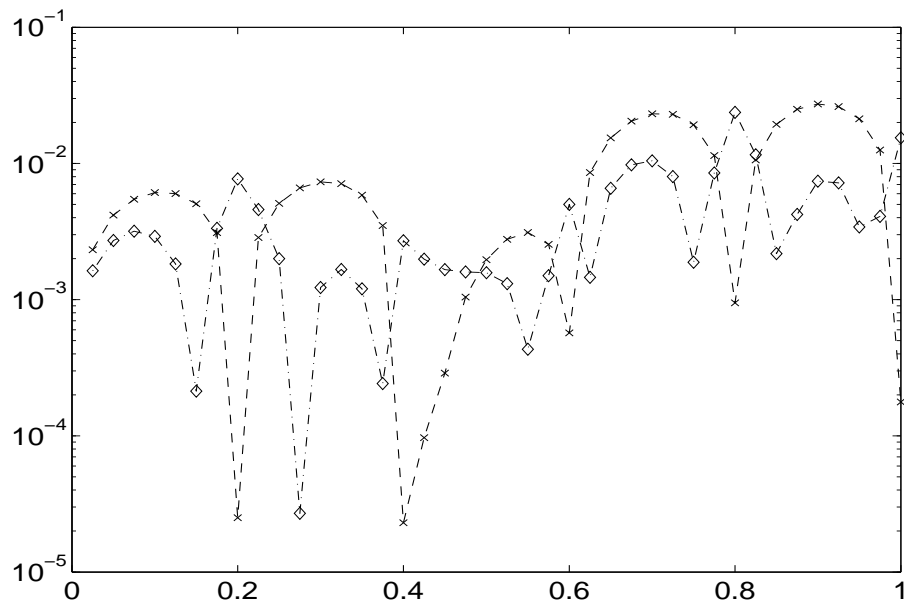


Abbildung 7.15: Abstand der sampled-data (x) beziehungsweise der optimierten Lösung (o) von der exakten Trajektorie mit Samplingperiode  $T = 0.2$ . Optimierung ausgehend von (6.6) mit maximal zwei Iterationen und acht Messpunkten pro Intervall.

Abschließend wird das Samplingintervall halbiert, also  $T = 0.1$  gesetzt. Dies liefert uns Abbildung 7.16. Auch für diese kleinere Samplingperiode sieht man an Tabelle 7.6, dass die Nachoptimierung mit nur zwei Optimierungsschritten deutliche Verbesserungen bringt. Dies wird umso deutlicher je mehr Messpunkte in den einzelnen Intervallen gesetzt werden.

| Auswertungspunkte | sampled-data Lösung   | optimierte Lösung     |
|-------------------|-----------------------|-----------------------|
| 4                 | 0.0004833864276840967 | 0.0001598128356019349 |
| 2                 | 0.0002274328184089890 | 0.0001364155306820320 |

Tabelle 7.6: Residuen der Nachoptimierung ausgehend von (6.6) mit zwei beziehungsweise vier Messpunkten pro Samplingintervall;  $T=0.1$ , 2 Optimierungsschritte.

Auch in diesem Beispiel erreicht man durch eine geringere Gewichtung des letzten Intervallpunktes eine Verbesserung auf 0.00014609... für vier und 0.00012363... für zwei Messpunkte.

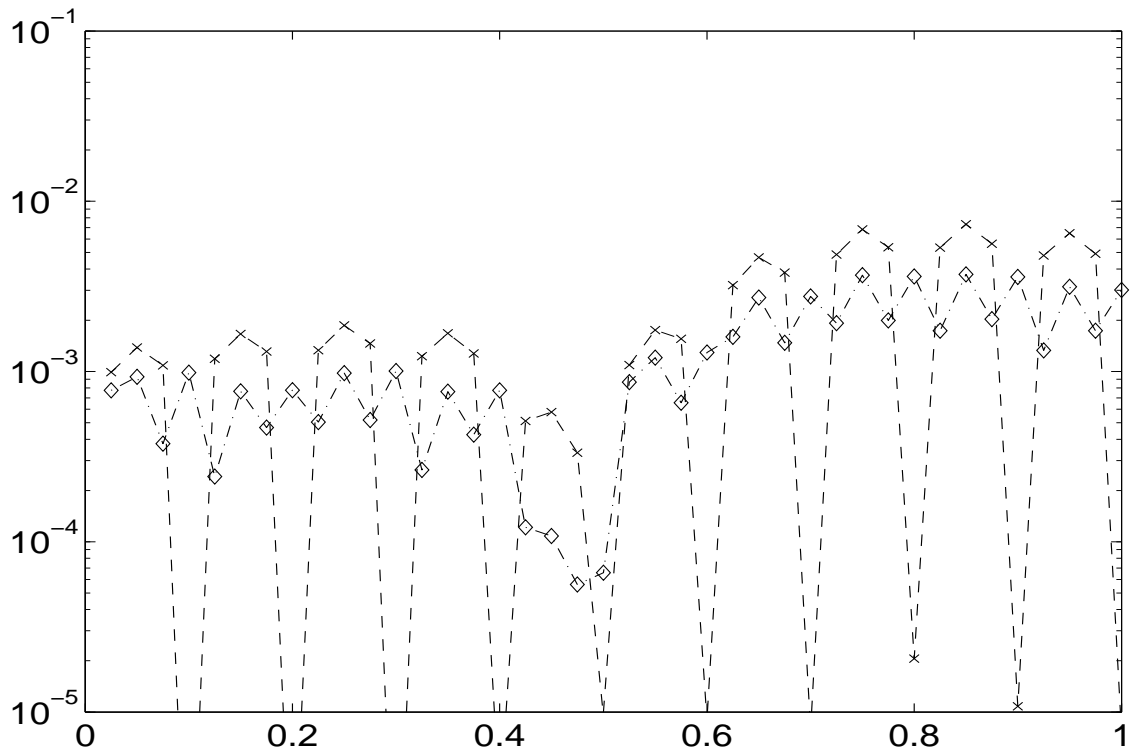


Abbildung 7.16: Abstand der sampled-data (x) beziehungsweise der optimierten Lösung (o) von der exakten Trajektorie mit Samplingperiode  $T = 0.1$ . Optimierung ausgehend von (6.6) mit maximal zwei Iterationen und vier Messpunkten pro Intervall.

## 7.5 Fazit und Ausblick

Es wurden bis einschließlich Kapitel 5 theoretische Resultate unter der Voraussetzung einer hinreichend kleinen Schrittweite hergeleitet. Die Wichtigsten sind

- eine exakte Darstellung der Abweichungen des sampled-data Systems vom kontinuierlichen closed-loop System bis zur siebten Ordnung mittels Fließ- beziehungsweise Taylorreihenentwicklung,
- eine hinreichende Bedingung in Form einer Lie-Klammer für eine Approximation beliebiger Ordnung,

- sowie notwendige Bedingungen, die eine wesentlich größere Flexibilität bei der Wahl der Feedback-Steuerungen ermöglichen.

Im Anschluss daran wurden die theoretischen Resultate auf ihre Praxistauglichkeit untersucht. Bemerkenswert ist dabei, dass sowohl die Existenz einer Funktion  $\alpha$  im Sinne von (5.16) als auch der Nachweis, dass es keine solche geben kann, interessante Aussagen ermöglichen. Im positiven Fall kann man mit Hilfe der Maple-Routine aus [11] sampled-data Feedback-Steuerungen explizit berechnen und im negativen Fall angeben, dass solche nicht existieren können. Danach wurden mit Hilfe eines Optimierungsansatzes die erhaltenen Lösungen weiter verbessert, indem man den Fokus weg von den Endpunkten der Samplingintervalle hin zu einem trajektorienbasierten Ansatz verschiebt. Ausgehend von der sampled-data Lösung erzielt diese Nachoptimierung bereits nach wenigen Iterationen signifikante Verbesserungen. Allerdings gibt es in Bezug auf diese Vorgehensweise und die vorher angesprochenen theoretischen Resultate noch einige Fragestellungen, die einer weiteren Untersuchung bedürfen:

- Nach Abschnitt 7.3 wäre es interessant zu wissen, ob man Kriterien finden kann, die vorhersagen, ob die sampled-data Lösung als Startwert für die Optimierung geeignet ist. So könnte es sinnvoll sein bei fast allen Intervallen die sampled-data Lösung als Initialisierung zu nutzen, sie jedoch bei besonders ungünstig gelegenen Anfangswerten in Kombination mit einer geringen Ordnung der sampled-data Steuerung wegzulassen.
- In Bemerkung 7.4 und Abschnitt 7.4 haben wir uns unter anderem mit unterschiedlichen Gewichtungen der einzelnen Messpunkte beschäftigt. Dieser Bereich war jedoch kein Schwerpunkt dieser Arbeit und bietet Raum für weitergehende Untersuchungen.
- Außerdem könnte man testen, ob man mit Hilfe dieses Verfahrens brauchbare Startlösungen für die modellprädiktive Regelung generieren kann.
- Zudem stellt sich die Frage, inwieweit sich die Ergebnisse dieser Arbeit auf partielle Differentialgleichungen verallgemeinern lassen beziehungsweise einige Ansätze übertragbar sind.



# Anhang A

## Programmparameter und Matlabcode

In diesem Abschnitt soll auf die in Kapitel 7 verwendete Optimierungsroutine eingegangen werden. Dabei werden insbesondere die wichtigsten einstellbaren Parameter für den Programmaufruf erläutert und der komplette Programmcode, abgesehen von dem Programmteil für das Erzeugen der Graphiken, angegeben. Im letzten Abschnitt wird auf einen kleinen Matlabfehler eingegangen, der erklärt, warum die Residuenberechnung nicht einfach von Matlab übernommen werden konnte.

### A.1 Programmparameter

In diesem Abschnitt sollen die verschiedenen Programmparameter vorgestellt und erläutert werden. Zu diesem Zweck betrachten wir das File `opttest.m`, das zum Start der Routine aufgerufen werden muss.

Datei: `opttest.m`

```
1 function opttest
2
3
4 select_prob = 2;           % 1=moore-greitzer or 2=academic example
5 select_initial = 1;       % 1=sd or 2=without initial value or 3=both
6 Opt_max_iter = 2;        % maximum number of iterations
7                           %           for the optimisation routine
8 Opt_max_f_ev = inf;      % max. number of function evaluations
9                           %           for the optimisation routine
10 select_alg = 2;          % 1=GN or 2=LM
11 select_linesearch = 1;   % 1=cubicpoly or 2=quadcubic
12 select_precision = 4;    % number of points for least square-optimisation
13 select_control = 4;      % choose control function !for the academic example!
14                           %           possible are 0,1,2,3,4,6,8
15 select_control_sd = 4;   % like select_control, but only
16                           %           for the sampled-data solution
```

```

17 plotoption=2;           % 1=comparision of exact and optimisation solution;
18                        %           2=exact, sd and optimisation solution
19 select_weight=1;       %1=normal, 2=the last 25 percent*2,
20                        %           3=the last 25 percent/2
21
22
23 time_segment = 1;
24 n             = 5;      % determines the stepsize
25
26
27 go(select_prob, select_initial, Opt_max_iter, Opt_max_f_ev,...
28     select_alg, select_linesearch, select_precision,...
29     select_control, select_control_sd, plotoption, select_weight,...
30     time_segment,n);

```

Die erste Variable `select_prob` wählt aus, welches der beiden Beispiele man betrachten möchte, also das Moore-Greitzer Modell aus Abschnitt 6.2 oder das akademische Beispiel aus Abschnitt 6.1. Desweiteren kann man verschiedene Parameter für die Optimierung wählen. Mit `select_alg` entscheidet man, welchen der beiden Optimierungsalgorithmen man verwenden möchte. Dabei bedeutet eins, dass der Gauß-Newton Algorithmus benutzt werden soll und zwei verwendet stattdessen den Levenberg-Marquardt Algorithmus. Nachdem der Optimierungsalgorithmus festgelegt ist, kann man weitere Parameter für diesen festsetzen:

- `Opt_max_iter`: bestimmt die maximale Anzahl an Iterationen pro Optimierungsschritt.
- `Opt_max_f_ev`: legt die maximale Anzahl an Funktionsauswertungen je Optimierung fest.
- `select_linesearch`: wählt die zu verwendende linesearch-Methode aus. Für nähere Informationen bezüglich der einzelnen Verfahren sei auf Kapitel 7 verwiesen.
- `select_precision`: gibt die Anzahl der Messpunkte pro Samplingintervall an.
- `select_initial`: gibt vor, ob die Optimierung die sampled-data Steuerung oder Null als Startwert verwenden soll. Falls das akademische Beispiel betrachtet wird, kann man mit `select_control` noch zusätzlich festlegen, welches sampled-data Feedback als Startwert verwendet werden soll.
- `select_weight`: legt die Gewichtung für die Optimierung fest. Bei eins werden alle Messpunkte identisch gewichtet. Bei zwei werden die letzten fünfundzwanzig Prozent der Auswertungspunkte mit doppelter Gewichtung berücksichtigt und bei drei nimmt der Einfluß dieser Punkte um fünfzig Prozent ab.

Der Parameter `n` gibt die Anzahl der Samplingperioden an. Die Länge der einzelnen Samplingintervalle ergibt sich dann als Quotient aus `time_segment` und `n`. Als letzter Parameter



soll an dieser Stelle `select_control_sd` erläutert werden. Dieser gibt an, welche sampled-data Steuerung für die sampled-data Lösung verwendet werden soll, hat aber keinen Einfluß auf die Optimierung. Auf die Plotoptionen soll hier nicht näher eingegangen werden.

## A.2 Matlabcode

Der Hauptteil des Programms befindet sich in der Datei `go.m`. Ab Zeile 349 bis zum eigentlichen Dateiende sind die Anweisungen für die graphische Ausgabe zu finden, die allerdings an dieser Stelle weggelassen worden sind, weil sie für das Verständnis des Programmablaufs irrelevant sind. In dem File `go.m` werden die beiden in Matlab vorimplementierten Routinen `nonlinlsp` und `ode45` genutzt. Diese realisieren mit den oben beschriebenen Optionen unseren Optimierer beziehungsweise erzeugen die Referenz- und die sampled-data Lösung. Es werden außerdem folgende Hilfsroutinen verwendet

- `euklidnorm.m`
- `moore_greitzer.m`
- `moore_greitzer_sd.m`
- `control.m`
- `control_2.m`,

die später kurz erläutert werden.

Datei: `go.m`

```

1 %function go
2 function go(select_prob, select_initial, Opt_max_iter, Opt_max_f_ev,...
3             select_alg, select_linesearch, select_precision,...
4             select_control, select_control_sd, plotoption, select_weight,...
5             time_segment,n)
6
7 %-----
8 %   moore-greitzer.m
9 %   An approximation of the Moore-Greitzer jet engine model
10 %-----
11
12
13 h           = time_segment/n;%
14
15
16 %-----
17 %-----   initialisation   -----
18 %-----

```

```
19
20
21 if (select_prob==1)
22     ex_name = 'Moore-Greitzer_jet_engine_model';
23     fprintf('\nBeispiel: Moore-Greitzer_jet_engine_model\n')
24 elseif(select_prob==2)
25     ex_name = 'academic_example';
26     fprintf('\nBeispiel: academic_example\n')
27 else
28     select_prob==2;
29     ex_name = 'Moore-Greitzer_jet_engine_model';
30     fprintf('\nkein Beispiel gewählt.')
31     fprintf('\nEs wird das Moore-Greitzer_jet_engine_model verwendet!\n')
32 end
33 if (select_prob==2)
34     aw_exact = [-1;1];
35 else
36     aw_exact = [22;21];
37 end
38 aw_sd      = aw_exact;
39 aw_opt     = aw_exact;
40 aw_optsd   = aw_exact;
41
42
43 Output     = [];
44 Output3    = [0 aw_exact(1) aw_exact(2)];
45 Output_sd  = [];
46 Output_sd2 = [0 aw_exact(1) aw_exact(2)];
47 Output_sd3 = [0 aw_exact(1) aw_exact(2)];
48 Output_opt = [];
49 Output_opt2 = [0 aw_exact(1) aw_exact(2)];
50 Output_opt3 = [0 aw_exact(1) aw_exact(2)];
51 Output_optsd = [];
52 Output_optsd2 = [0 aw_exact(1) aw_exact(2)];
53 Output_optsd3 = [0 aw_exact(1) aw_exact(2)];
54 value_opt  = [];
55 value_optsd = [];
56 res_opt    = [];
57 res_optsd  = [];
58 resid_opt  = [];
59 resid_optsd = [];
60 exit_opt   = [];
61 exit_optsd = [];
62 res_lsq    = [];
63 res_lsqsd  = [];
64 lsq_f_opt  = [];
65 lsq_f_optsd = [];
66
```

```

67
68 if (select_alg==1)
69     sel_alg = 'off';
70     alg_name = 'Gau -Newton';
71     fprintf('\nAlgorithmus: Gau -Newton\n')
72 elseif(select_alg==2)
73     sel_alg = 'on';
74     alg_name = 'Levenberg-Marquardt';
75     fprintf('\nAlgorithmus: Levenberg-Marquardt\n')
76 else
77     sel_alg = 'on';
78     alg_name = 'Levenberg-Marquardt';
79     fprintf('\nkein Algorithmus gewhlt.')
80     fprintf('\nEs wird Levenberg-Marquardt verwendet!\n')
81 end
82
83 if (select_initial==1)
84     sel_ini = 'sampled-data_feedback_as_initial_value';
85 elseif (select_initial==2)
86     sel_ini = 'without_initial_value';
87 elseif (select_initial==3)
88     sel_ini_1 = 'sampled-data_feedback_as_initial_value';
89     sel_ini_2 = 'without_initial_value';
90 else
91     fprintf('\ninitial_value_for_optimisation_not_chosen,')
92     fprintf('\ni.e. optimisation without initial_value')
93     sel_ini = 'without_initial_value';
94     select_initial=2;
95 end
96
97 if (select_linesearch==1)
98     sel_ls = 'cubicpoly';
99 elseif (select_linesearch==2)
100     sel_ls = 'quadcubic';
101 else
102     fprintf('\nlinesearch_method_for_optimisation_not_chosen,')
103     fprintf('\ni.e. linesearch: cubicpoly')
104     sel_ls = 'cubicpoly';
105     select_linesearch=1;
106 end
107
108 if (select_prob==2)
109     if (select_control==1 | select_control==2 | select_control==3 |...
110         select_control==4 | select_control==6 | select_control==8)
111         fprintf('\nselected_control_function_for_optimisation: u_%1.1f^%d\n',...
112             h,select_control)
113     else
114         select_control=0;

```

```

115         fprintf('\nselected_control_function_for_optimisation: u_%1.1f^d\n',...
116             h,select_control_sd)
117     end
118     if (select_control_sd==1 | select_control_sd==2 |...
119         select_control_sd==3 | select_control_sd==4 |...
120         select_control_sd==6 | select_control_sd==8)
121         fprintf('\nselected_control_function_for_sd-solution: u_%1.1f^d\n',...
122             h,select_control_sd)
123     else
124         select_control_sd=0;
125         fprintf('\nselected_control_function_for_sd-solution: u_%1.1f^d\n',...
126             h,select_control_sd)
127     end
128 end
129
130 if (select_weight==2 | select_weight==3)
131     fprintf('\nweights_for_optimisation_changed!')
132     fprintf('\nselected_weight: %d\n',select_weight)
133 else
134     select_weight==1;
135 end
136
137
138 %-----
139 %----- main program -----
140 %-----
141
142
143 for j=1:n
144
145     tspan=[(j-1)*h, j*h];
146
147     options = optimset('Display','off',...
148         'LargeScale','off','LevenbergMarquardt',sel_alg,...
149         'MaxIter',Opt_max_iter,'MaxFunEvals',Opt_max_f_ev,...
150         'LineSearchType',sel_ls);
151     %'LineSearchType','quadcubic');
152     %'LineSearchType','cubicpoly');
153
154     if (select_initial~=1)
155         [x_opt ,resnorm_opt ,residual_opt ,...
156             exitflag_opt ,lsqoutput_opt ] = ...
157             lsqnonlin(@euklidnorm,0.0,-inf,inf,options,...
158                 tspan,aw_exact,aw_opt ,select_precision,...
159                 select_prob,select_control,select_weight);
160         %lsqoutput_opt
161         lsq_f_opt = [lsq_f_opt ;lsqoutput_opt.funcCount ];
162     end

```

```

163     if (select_initial~=2)
164         [x_optsd, resnorm_optsd, residual_optsd, ...
165             exitflag_optsd, lsqoutput_optsd] = ...
166             lsqnonlin(@euklidnorm, 1.0, -inf, inf, options, ...
167                 tspan, aw_exact, aw_optsd, select_precision, ...
168                 select_prob, select_control, select_weight);
169         %lsqoutput_optsd
170         lsq_f_optsd = [lsq_f_optsd; lsqoutput_optsd.funcCount];
171     end
172
173     %if (j<=3)
174     %     x_optsd=1.0;
175     %end
176
177     %     size(resnorm_opt)
178     %     size(resnorm_optsd)
179     %     size(residual_opt)
180     %     size(residual_optsd)
181     %     residual_opt
182     %     residual_optsd
183
184     aw_sd_save = aw_sd;
185     aw_opt_save = aw_opt;
186     aw_optsd_save = aw_optsd;
187
188     for k=1:select_precision
189
190         tspan2 = [tspan(1)+(k-1)*h/select_precision, ...
191                 tspan(1)+k*h/select_precision];
192
193         options = odeset('AbsTol', 1e-7, 'RelTol', 1e-4);
194         [t, y] = ode45(@moore_greitzer, tspan2, aw_exact, ...
195             options, ...
196             select_prob); % exact solution
197         [t_sd, y_sd] = ode45(@moore_greitzer_sd, tspan2, aw_sd, ...
198             options, aw_sd_save, 1, ...
199             select_prob, select_control_sd, h); % sd solution
200         if (select_initial~=1)
201             [t_opt, y_opt] = ode45(@moore_greitzer_sd, tspan2, aw_opt, ...
202                 options, aw_opt_save, x_opt, ...
203                 select_prob, select_control, h); % opt solution
204         end
205         if (select_initial~=2)
206             [t_optsd, y_optsd] = ode45(@moore_greitzer_sd, tspan2, aw_optsd, ...
207                 options, aw_optsd_save, x_optsd, ...
208                 select_prob, select_control, h); % opt sol.
209         end
210

```

```

211 [size_time      ,size_tmp      ] = size(t);
212 [size_time_sd   ,size_tmp_sd   ] = size(t_sd);
213 if (select_initial~=1)
214     [size_time_opt ,size_tmp_opt ] = size(t_opt);
215 end
216 if (select_initial~=2)
217     [size_time_optsd ,size_tmp_optsd] = size(t_optsd);
218 end
219
220 if (k==select_precision)
221 fprintf('\n%1.0f. Schritt t y1 y2\n',j)
222 fprintf('-----\n')
223     if (j==100)
224         fprintf('exakte Lösung: %6.4f %12.8f %12.8f\n' ,...
225             t(1)      ,y(1,1)      ,y(1,2)      )
226         fprintf('SD-L sung: %6.4f %12.8f %12.8f\n' ,...
227             t_sd(1)   ,y_sd(1,1)   ,y_sd(1,2)   )
228         fprintf('Optimierung1: %6.4f %12.8f %12.8f\n' ,...
229             t_opt(1)  ,y_opt(1,1)  ,y_opt(1,2)  )
230         fprintf('Optimierung2: %6.4f %12.8f %12.8f\n\n' ,...
231             t_optsd(1),y_optsd(1,1),y_optsd(1,2))
232     end
233     fprintf('exakte Lösung: %6.4f %12.8f %12.8f\n' ,...
234         t(size_time)      ,y(size_time,1)      ,...
235         y(size_time,2)      )
236     fprintf('SD-L sung: %6.4f %12.8f %12.8f\n' ,...
237         t_sd(size_time_sd)      ,y_sd(size_time_sd,1)      ,...
238         y_sd(size_time_sd,2)      )
239     if (select_initial~=1)
240         fprintf('Optimierung1: %6.4f %12.8f %12.8f\n' ,...
241             t_opt(size_time_opt)      ,y_opt(size_time_opt,1)      ,...
242             y_opt(size_time_opt,2)      )
243     end
244     if (select_initial~=2)
245         fprintf('Optimierung2: %6.4f %12.8f %12.8f\n\n' ,...
246             t_optsd(size_time_optsd) ,y_optsd(size_time_optsd,1) ,...
247             y_optsd(size_time_optsd,2))
248     end
249 end
250
251 Output      = [Output      ;t y      ];
252 Output_sd   = [Output_sd   ;t_sd y_sd   ];
253 if (k==select_precision)
254     Output_sd2 = [Output_sd2 ;t_sd(size_time_sd)...
255                 y_sd(size_time_sd,1) y_sd(size_time_sd,2)];
256 end
257 Output3     = [Output3     ;t(size_time)      y(size_time,1)...
258                 y(size_time,2)      ];

```

```

259     Output_sd3 = [Output_sd3 ;t_sd(size_time_sd) y_sd(size_time_sd,1)...
260                 y_sd(size_time_sd,2)];
261     if (select_initial~=1)
262         Output_opt = [Output_opt ;t_opt y_opt ];
263         res_lsq = [res_lsq ;...
264                 sqrt( (y(size_time,1)-y_opt(size_time_opt,1))^2 +...
265                 (y(size_time,2)-y_opt(size_time_opt,2))^2 )];
266         if (k==select_precision)
267             Output_opt2 = [Output_opt2 ;t_opt(size_time_opt)...
268                             y_opt(size_time_opt,1) y_opt(size_time_opt,2)];
269             value_opt = [value_opt ;j*h x_opt ];
270             res_opt = [res_opt ;j*h resnorm_opt ];
271             resid_opt = [resid_opt ;residual_opt ];
272             exit_opt = [exit_opt ;exitflag_opt ];
273         end
274         Output_opt3 = [Output_opt3 ;t_opt(size_time_opt)...
275                       y_opt(size_time_opt,1) y_opt(size_time_opt,2)];
276     end
277     if (select_initial~=2)
278         Output_optsd = [Output_optsd ;t_optsd y_optsd ];
279         res_lsqs = [res_lsqs ;...
280                  sqrt( (y(size_time,1)-y_optsd(size_time_optsd,1))^2 +...
281                  (y(size_time,2)-y_optsd(size_time_optsd,2))^2 )];
282         if (k==select_precision)
283             Output_optsd2= [Output_optsd2 ;t_optsd(size_time_optsd)...
284                             y_optsd(size_time_optsd,1) y_optsd(size_time_optsd,2)];
285             value_optsd = [value_optsd ;j*h x_optsd ];
286             res_optsd = [res_optsd ;j*h resnorm_optsd ];
287             resid_optsd = [resid_optsd ;residual_optsd];
288             exit_optsd = [exit_optsd ;exitflag_optsd ];
289         end
290         Output_optsd3= [Output_optsd3 ;t_optsd(size_time_optsd)...
291                       y_optsd(size_time_optsd,1) y_optsd(size_time_optsd,2)];
292     end
293
294     aw_exact = [y(size_time,1) ,y(size_time,2) ];
295     aw_sd = [y_sd(size_time_sd,1) ,y_sd(size_time_sd,2) ];
296     if (select_initial~=1)
297         aw_opt = [y_opt(size_time_opt,1) ,y_opt(size_time_opt,2)];
298     end
299     if (select_initial~=2)
300         aw_optsd = [y_optsd(size_time_optsd,1),y_optsd(size_time_optsd,2)];
301     end
302
303     end
304 end
305
306

```

```

307 %-----
308 %----- results -----
309 %-----
310
311
312 if (select_initial~=1)
313     res_lsq;
314     res_tmp=res_lsq.^2;
315     [res_tmp3, res_tmp2]=size(res_lsq);
316     res_tmp2=0;
317     for l=1:res_tmp3
318         res_tmp2=res_tmp2+res_tmp(l);
319     end
320     format long;
321     res_tmp2
322     lsq_f_opt
323     value_opt
324     %     res_opt
325     %     resid_opt
326     %     exit_opt
327 end
328 if (select_initial~=2)
329     res_lsqsd;
330     res_tmpsd=res_lsqsd.^2;
331     [res_tmpsd3, res_tmpsd2]=size(res_lsqsd);
332     res_tmpsd2=0;
333     for l=1:res_tmpsd3
334         res_tmpsd2=res_tmpsd2+res_tmpsd(l);
335     end
336     format long;
337     res_tmpsd2
338     lsq_f_optsd
339     value_optsd
340     %     res_optsd
341     %     resid_optsd
342     %     exit_optsd
343 end
344
345 %exit_optsd
346 %exit_opt
347
348
349 %-----
350 %----- plots -----
351 %-----

```



Die Routine `euklidnorm.m` wird als Argument der Matlabfunktion `nonlinlsp` verwendet und soll die einzelnen Abstände an den Auswertungspunkten gemessen in der euklidischen Norm in einem Vektor liefern. Die Quadrierung und Aufsummation erfolgt in der Matlabfunktion selbst.

Datei: `euklidnorm.m`

```

1 function f = euklidnorm(u, tspan, aw_exact, aw_opt,...
2                     select_precision,select_prob,...
3                     select_control,select_weight)
4
5 n2          = select_precision;
6 h2          = (tspan(2)-tspan(1))/n2;
7 aw_exact2   = aw_exact;
8 aw_opt2     = aw_opt;
9 g           = [];
10
11 for j=1:n2
12
13     tspan2=[tspan(1)+(j-1)*h2,tspan(1)+j*h2];
14
15     options      = odeset('AbsTol',1e-7,'RelTol',1e-4);
16     [t2,y2]      = ode45(@moore_greitzer      ,tspan2,aw_exact2,options,...
17                         select_prob);
18     [t_opt2,y_opt2] = ode45(@moore_greitzer_sd,tspan2,aw_opt2  ,options,...
19                             aw_opt,u,select_prob,select_control,h2*n2);
20
21     [size_time2,size_tmp2]=size(t2);
22     [size_time_opt2,size_tmp_opt2]=size(t_opt2);
23
24     aw_exact2=[y2(size_time2,1),y2(size_time2,2)];
25     aw_opt2=[y_opt2(size_time_opt2,1),y_opt2(size_time_opt2,2)];
26
27     g = [g; sqrt( (y2(size_time2,1)-y_opt2(size_time_opt2,1))^2 +...
28                (y2(size_time2,2)-y_opt2(size_time_opt2,2))^2 )];
29     if ((select_weight == 2 & j/select_precision>0.75) |...
30         (select_weight == 3 & j/select_precision<=0.75))
31         g = [g; sqrt( (y2(size_time2,1)-y_opt2(size_time_opt2,1))^2 +...
32                    (y2(size_time2,2)-y_opt2(size_time_opt2,2))^2 )];
33     end
34
35 end
36
37 f = g;
```

Die Files `moore_greitzer.m` und `moore_greitzer_sd.m` liefern die Auswertung des Systems an einem vorgegebenen Punkt. Für die exakte Lösung wird dabei immer der aktuelle Zustand für die Berechnung der Feedback-Steuerung verwendet. Für die sampled-data bezie-

ungsweise die optimierten sampled-data Lösungen wird hingegen immer die auf dem zuletzt erreichten Samplingendpunkt basierende Steuerung verwendet. Die Files sind für beide Beispiele, die Nomenklatur beruht auf der Tatsache, dass zuerst das Moore-Greitzer Modell implementiert worden ist.

Datei: moore\_greitzer.m

```

1 function z=moore_greitzer(t,y,select_prob)
2 if (select_prob==2)
3     z = [ y(1)+y(2)^2*(-y(1)-1);...
4           y(2)+2*y(1)^2*(-y(1)-1)];
5 else
6     z = [ -y(2)-(3/2)*y(1)^2-(1/2)*y(1)^3;...
7           7*y(1)-5*y(2)];
8 end

```

Datei: moore\_greitzer\_sd.m

```

1 function z=moore_greitzer_sd(t,y,last_aw,u,select_prob,select_control,stepsize)
2 if (select_prob==2)
3     z = [ y(1)+y(2)^2*u*control_2(t,last_aw,select_control,stepsize);...
4           y(2)+2*y(1)^2*u*control_2(t,last_aw,select_control,stepsize)];
5 else
6     z = [ -y(2)-(3/2)*y(1)^2-(1/2)*y(1)^3;...
7           u*control(t,last_aw)];
8 end

```

Entsprechend den Files zur Funktionsauswertung gibt es auch für die Kontrollfunktionen zwei Files. Das zweite control\_2.m ist verhältnismäßig lang, weil es die sampled-data Steuerung bis  $M = 8$  realisieren kann. Es ist in erster Linie der Vollständigkeit halber in ungekürzter Fassung angegeben.

Datei: control.m

```

1 function u=control(t,last_aw)
2 u = 7*last_aw(1)-5*last_aw(2);

```

Datei: control\_2.m

```

1 function u=control_2(t,last_aw,select_control,stepsize)
2
3 T = stepsize;
4 evaluate_control=0;
5
6 if (select_control>=8)%r8
7     evaluate_control = evaluate_control +...
8     1/120960*T^7*(-12*last_aw(1)+68208*last_aw(1)^4+...
9     419664*last_aw(1)^5-2825088*last_aw(1)^10-...
10    1665792*last_aw(1)^9-502848*last_aw(1)^8+...
11    353928*last_aw(1)^7-249600*last_aw(1)^13-...

```

```
12      1263168*last_aw(1)^12-2616192*last_aw(1)^11+...
13      753744*last_aw(1)^6+3*last_aw(2)^14+38862*last_aw(2)^6+...
14      1020*last_aw(2)^2-73488*last_aw(2)^9-153*last_aw(2)^12-...
15      14701*last_aw(2)^8-35840*last_aw(2)^5-22974*last_aw(2)^4+...
16      2555*last_aw(2)^10+120776*last_aw(2)^7+...
17      10224*last_aw(2)^11+74232*last_aw(2)^11*last_aw(1)^3+...
18      51048*last_aw(2)^11*last_aw(1)+...
19      563520*last_aw(1)^13*last_aw(2)^2+...
20      3305280*last_aw(1)^12*last_aw(2)^2+...
21      2687736*last_aw(2)^8*last_aw(1)^6+...
22      8891136*last_aw(2)^5*last_aw(1)^9+...
23      1685952*last_aw(1)^10*last_aw(2)^5+...
24      594120*last_aw(1)^7*last_aw(2)^8+...
25      8161920*last_aw(1)^11*last_aw(2)^2+...
26      19478016*last_aw(2)^5*last_aw(1)^8+...
27      4915512*last_aw(2)^8*last_aw(1)^5+...
28      3*last_aw(2)^14*last_aw(1)-...
29      358400*last_aw(1)^3*last_aw(2)^2-...
30      3193344*last_aw(1)^4*last_aw(2)^2-...
31      8845464*last_aw(1)^5*last_aw(2)^2+...
32      6942*last_aw(2)^2*last_aw(1)-...
33      23688*last_aw(2)*last_aw(1)^2-...
34      44940*last_aw(2)^4*last_aw(1)-...
35      179760*last_aw(2)*last_aw(1)^4-...
36      149688*last_aw(2)*last_aw(1)^3+...
37      93240*last_aw(2)^11*last_aw(1)^2-...
38      4343040*last_aw(1)^10*last_aw(2)^3-...
39      976368*last_aw(2)^5*last_aw(1)^4-...
40      4175100*last_aw(2)^5*last_aw(1)^3-...
41      2856780*last_aw(2)^5*last_aw(1)^2-...
42      664064*last_aw(2)^5*last_aw(1)-...
43      2850720*last_aw(1)^7*last_aw(2)^2-...
44      9696296*last_aw(1)^6*last_aw(2)^2+...
45      317440*last_aw(2)^4*last_aw(1)^2+...
46      15647248*last_aw(2)^4*last_aw(1)^4+...
47      3922864*last_aw(2)^4*last_aw(1)^3+...
48      317440*last_aw(2)*last_aw(1)^5+...
49      8333728*last_aw(2)*last_aw(1)^7+...
50      2715104*last_aw(2)*last_aw(1)^6+...
51      2161600*last_aw(2)*last_aw(1)^10+...
52      8179040*last_aw(2)*last_aw(1)^9+...
53      11953504*last_aw(2)*last_aw(1)^8+...
54      2702*last_aw(2)^10*last_aw(1)+...
55      1210496*last_aw(2)^7*last_aw(1)^4+...
56      3155392*last_aw(2)^7*last_aw(1)^3+...
57      2868808*last_aw(2)^7*last_aw(1)^2+...
58      1043560*last_aw(2)^7*last_aw(1)+...
59      6700960*last_aw(2)^4*last_aw(1)^7+...
```

```

60      22112560*last_aw(2)^4*last_aw(1)^6+...
61      27453424*last_aw(2)^4*last_aw(1)^5+...
62      4621992*last_aw(2)^8*last_aw(1)^4+...
63      2340384*last_aw(2)^8*last_aw(1)^3+...
64      604800*last_aw(2)^8*last_aw(1)^2+...
65      22792704*last_aw(2)^5*last_aw(1)^7+...
66      15205056*last_aw(2)^5*last_aw(1)^6+...
67      5677056*last_aw(2)^5*last_aw(1)^5+...
68      10953600*last_aw(1)^10*last_aw(2)^2+...
69      8577600*last_aw(1)^9*last_aw(2)^2+...
70      3885120*last_aw(1)^8*last_aw(2)^2+...
71      21816*last_aw(2)^11*last_aw(1)^4+...
72      47532*last_aw(2)^8*last_aw(1)-...
73      34316160*last_aw(2)^3*last_aw(1)^7+...
74      853464*last_aw(2)^3*last_aw(1)^4+...
75      1840272*last_aw(2)^3*last_aw(1)^3+...
76      753200*last_aw(2)^3*last_aw(1)^2+...
77      68208*last_aw(2)^3*last_aw(1)-...
78      7839*last_aw(2)^6*last_aw(1)-...
79      1142784*last_aw(2)^6*last_aw(1)^2-...
80      17204352*last_aw(2)^3*last_aw(1)^6-...
81      282672*last_aw(2)^9*last_aw(1)^4-...
82      951012*last_aw(2)^9*last_aw(1)^2-...
83      861492*last_aw(2)^9*last_aw(1)^3-...
84      445680*last_aw(2)^9*last_aw(1)-...
85      36433728*last_aw(2)^3*last_aw(1)^8-...
86      22568544*last_aw(2)^6*last_aw(1)^5-...
87      4251456*last_aw(2)^3*last_aw(1)^5-...
88      6641568*last_aw(2)^6*last_aw(1)^3-...
89      17266560*last_aw(2)^6*last_aw(1)^4-...
90      156*last_aw(2)^12*last_aw(1)-...
91      14546880*last_aw(1)^6*last_aw(2)^6-...
92      19800384*last_aw(1)^9*last_aw(2)^3-...
93      3686592*last_aw(1)^7*last_aw(2)^6)-...
94      1/3628800*T^8*(45*last_aw(1)-1153656*last_aw(1)^4-8710632*last_aw(1)^5+...
95      233976960*last_aw(1)^10+126312960*last_aw(1)^9+34432320*last_aw(1)^8-...
96      7830720*last_aw(1)^7+25632000*last_aw(1)^13+121512000*last_aw(1)^12+...
97      234376320*last_aw(1)^11-18312696*last_aw(1)^6-665*last_aw(2)^14-...
98      981207*last_aw(2)^6-7665*last_aw(2)^2+5801600*last_aw(2)^9+...
99      15375*last_aw(2)^12+387583*last_aw(2)^8+874944*last_aw(2)^5+...
100     357429*last_aw(2)^4+69265*last_aw(2)^10-5216760*last_aw(2)^7-...
101     1877520*last_aw(2)^11-...
102     15996480*last_aw(2)^11*last_aw(1)^3-...
103     9952680*last_aw(2)^11*last_aw(1)-...
104     126792000*last_aw(1)^13*last_aw(2)^2-...
105     703966400*last_aw(1)^12*last_aw(2)^2-...
106     575515160*last_aw(2)^8*last_aw(1)^6-...
107     1897856640*last_aw(2)^5*last_aw(1)^9-...

```

```
108      379339200*last_aw(1)^10*last_aw(2)^5-...
109      133677000*last_aw(1)^7*last_aw(2)^8-...
110      1638422400*last_aw(1)^11*last_aw(2)^2-...
111      3926705280*last_aw(2)^5*last_aw(1)^8-...
112      997028280*last_aw(2)^8*last_aw(1)^5-...
113      675*last_aw(2)^14*last_aw(1)+...
114      8749440*last_aw(1)^3*last_aw(2)^2+...
115      92806560*last_aw(1)^4*last_aw(2)^2+...
116      293333400*last_aw(1)^5*last_aw(2)^2-...
117      76680*last_aw(2)^2*last_aw(1)+...
118      276060*last_aw(2)*last_aw(1)^2+...
119      829740*last_aw(2)^4*last_aw(1)+...
120      3318960*last_aw(2)*last_aw(1)^4+...
121      2282604*last_aw(2)*last_aw(1)^3-...
122      19163040*last_aw(2)^11*last_aw(1)^2+...
123      445996800*last_aw(1)^10*last_aw(2)^3-...
124      74914296*last_aw(2)^5*last_aw(1)^4+...
125      148571084*last_aw(2)^5*last_aw(1)^3+...
126      95842620*last_aw(2)^5*last_aw(1)^2+...
127      19489824*last_aw(2)^5*last_aw(1)+...
128      16551600*last_aw(1)^7*last_aw(2)^2+...
129      350203720*last_aw(1)^6*last_aw(2)^2-...
130      11606400*last_aw(2)^4*last_aw(1)^2-...
131      748799760*last_aw(2)^4*last_aw(1)^4-...
132      167029200*last_aw(2)^4*last_aw(1)^3-...
133      11606400*last_aw(2)*last_aw(1)^5-...
134      395617440*last_aw(2)*last_aw(1)^7-...
135      114861600*last_aw(2)*last_aw(1)^6-...
136      68864000*last_aw(2)*last_aw(1)^10-...
137      452721760*last_aw(2)*last_aw(1)^9-...
138      624418400*last_aw(2)*last_aw(1)^8+...
139      4087120*last_aw(2)^10*last_aw(1)+...
140      70103040*last_aw(2)^7*last_aw(1)^4-...
141      163137080*last_aw(2)^7*last_aw(1)^3-...
142      152112320*last_aw(2)^7*last_aw(1)^2-...
143      50701480*last_aw(2)^7*last_aw(1)-...
144      62457600*last_aw(2)^4*last_aw(1)^7-...
145      1203840640*last_aw(2)^4*last_aw(1)^6-...
146      1442065600*last_aw(2)^4*last_aw(1)^5-...
147      882747720*last_aw(2)^8*last_aw(1)^4-...
148      417501280*last_aw(2)^8*last_aw(1)^3-...
149      99611520*last_aw(2)^8*last_aw(1)^2-...
150      4315376640*last_aw(2)^5*last_aw(1)^7-...
151      2683983040*last_aw(2)^5*last_aw(1)^6-...
152      925015680*last_aw(2)^5*last_aw(1)^5-...
153      2061494400*last_aw(1)^10*last_aw(2)^2-...
154      1503361600*last_aw(1)^9*last_aw(2)^2-...
155      628478400*last_aw(1)^8*last_aw(2)^2-...
```

```
156 4908600*last_aw(2)^11*last_aw(1)^4-...
157 9152922*last_aw(2)^8*last_aw(1)+...
158 2861830080*last_aw(2)^3*last_aw(1)^7-...
159 10892160*last_aw(2)^3*last_aw(1)^4-...
160 45057600*last_aw(2)^3*last_aw(1)^3-...
161 15830136*last_aw(2)^3*last_aw(1)^2-...
162 1153656*last_aw(2)^3*last_aw(1)+...
163 2171160*last_aw(2)^6*last_aw(1)+...
164 79288320*last_aw(2)^6*last_aw(1)^2+...
165 1314446400*last_aw(2)^3*last_aw(1)^6+...
166 89523680*last_aw(2)^10*last_aw(1)^3+...
167 28185600*last_aw(2)^10*last_aw(1)^2+...
168 254873600*last_aw(2)*last_aw(1)^13+...
169 260416000*last_aw(2)*last_aw(1)^12+...
170 166169600*last_aw(2)*last_aw(1)^11+...
171 154132880*last_aw(2)^10*last_aw(1)^4+...
172 1559910400*last_aw(2)^7*last_aw(1)^7+...
173 1207257600*last_aw(2)^7*last_aw(1)^6+...
174 555795200*last_aw(2)^7*last_aw(1)^5+...
175 2073414400*last_aw(2)^4*last_aw(1)^10+...
176 1889040000*last_aw(2)^4*last_aw(1)^9+...
177 1057001600*last_aw(2)^4*last_aw(1)^8+...
178 170160*last_aw(2)^13+10*last_aw(2)^16+...
179 1191654400*last_aw(2)^7*last_aw(1)^8+...
180 791760*last_aw(2)^13*last_aw(1)+...
181 10*last_aw(2)^16*last_aw(1)+...
182 80617600*last_aw(2)^4*last_aw(1)^13+...
183 1377120000*last_aw(2)^4*last_aw(1)^11+...
184 509200000*last_aw(2)^4*last_aw(1)^12+...
185 291160*last_aw(2)^13*last_aw(1)^4+...
186 1364200*last_aw(2)^13*last_aw(1)^2+...
187 1033760*last_aw(2)^13*last_aw(1)^3+...
188 29028240*last_aw(2)^9*last_aw(1)^4+...
189 87259740*last_aw(2)^9*last_aw(1)^2+...
190 83901060*last_aw(2)^9*last_aw(1)^3+...
191 38180000*last_aw(2)^9*last_aw(1)+...
192 3280744320*last_aw(2)^3*last_aw(1)^8+...
193 2044974240*last_aw(2)^6*last_aw(1)^5+...
194 292856640*last_aw(2)^3*last_aw(1)^5+...
195 512134720*last_aw(2)^6*last_aw(1)^3+...
196 1452196000*last_aw(2)^6*last_aw(1)^4+...
197 74661040*last_aw(2)^10*last_aw(1)^6+...
198 7513600*last_aw(2)*last_aw(1)^16+...
199 147961200*last_aw(2)^10*last_aw(1)^5+...
200 16020*last_aw(2)^12*last_aw(1)+...
201 1407809280*last_aw(1)^6*last_aw(2)^6+...
202 1909900800*last_aw(1)^9*last_aw(2)^3+...
203 51584000*last_aw(2)*last_aw(1)^15+...
```

```

204         15462640*last_aw(2)^10*last_aw(1)^7+...
205         87884800*last_aw(2)^7*last_aw(1)^10+...
206         497964800*last_aw(2)^7*last_aw(1)^9+...
207         152896000*last_aw(2)*last_aw(1)^14+...
208         378584640*last_aw(1)^7*last_aw(2)^6);
209     end
210
211     if (select_control>=6)%r6
212         evaluate_control = evaluate_control +...
213         1/1440*T^5*(-52*last_aw(2)^8-272*last_aw(2)^5-...
214         54*last_aw(2)^8*last_aw(1)-5832*last_aw(2)^5*last_aw(1)^4-...
215         14352*last_aw(2)^5*last_aw(1)^3-...
216         11784*last_aw(2)^5*last_aw(1)^2-3536*last_aw(2)^5*last_aw(1)-...
217         10800*last_aw(1)^7*last_aw(2)^2-...
218         33584*last_aw(1)^6*last_aw(2)^2-...
219         2720*last_aw(1)^3*last_aw(2)^2-...
220         17424*last_aw(1)^4*last_aw(2)^2-...
221         37488*last_aw(1)^5*last_aw(2)^2+640*last_aw(2)^4*last_aw(1)^2+...
222         18880*last_aw(2)^4*last_aw(1)^4+...
223         5920*last_aw(2)^4*last_aw(1)^3+640*last_aw(2)*last_aw(1)^5+...
224         10240*last_aw(2)*last_aw(1)^7+4160*last_aw(2)*last_aw(1)^6+...
225         1600*last_aw(2)*last_aw(1)^10+7040*last_aw(2)*last_aw(1)^9+...
226         12160*last_aw(2)*last_aw(1)^8+2*last_aw(2)^10*last_aw(1)+...
227         896*last_aw(2)^7*last_aw(1)^4+2656*last_aw(2)^7*last_aw(1)^3+...
228         2800*last_aw(2)^7*last_aw(1)^2+1216*last_aw(2)^7*last_aw(1)+...
229         4960*last_aw(2)^4*last_aw(1)^7+...
230         18880*last_aw(2)^4*last_aw(1)^6+...
231         27520*last_aw(2)^4*last_aw(1)^5+...
232         2*last_aw(2)^10+176*last_aw(2)^7+...
233         126*last_aw(2)^2+357*last_aw(2)^6-...
234         627*last_aw(2)^4+1368*last_aw(1)^4+...
235         5736*last_aw(1)^5+3240*last_aw(1)^7+7608*last_aw(1)^6+...
236         411*last_aw(2)^2*last_aw(1)+1368*last_aw(2)^3*last_aw(1)+...
237         9968*last_aw(2)^3*last_aw(1)^2+...
238         18216*last_aw(2)^3*last_aw(1)^3+...
239         9720*last_aw(2)^3*last_aw(1)^4+405*last_aw(2)^6*last_aw(1)-...
240         900*last_aw(2)^4*last_aw(1)-1140*last_aw(2)*last_aw(1)^2-...
241         4332*last_aw(2)*last_aw(1)^3-6*last_aw(1)-...
242         3600*last_aw(2)*last_aw(1)^4)-...
243     1/30240*T^6*(2538*last_aw(2)^8+8568*last_aw(2)^5+...
244     2751*last_aw(2)^8*last_aw(1)+297108*last_aw(2)^5*last_aw(1)^4+...
245     673128*last_aw(2)^5*last_aw(1)^3+...
246     500964*last_aw(2)^5*last_aw(1)^2+...
247     132600*last_aw(2)^5*last_aw(1)+...
248     550200*last_aw(1)^7*last_aw(2)^2+...
249     1562184*last_aw(1)^6*last_aw(2)^2+...
250     85680*last_aw(1)^3*last_aw(2)^2+...
251     644952*last_aw(1)^4*last_aw(2)^2+...

```

```

252     1571256*last_aw(1)^5*last_aw(2)^2-...
253     45120*last_aw(2)^4*last_aw(1)^2-...
254     1712448*last_aw(2)^4*last_aw(1)^4-...
255     481008*last_aw(2)^4*last_aw(1)^3-...
256     45120*last_aw(2)*last_aw(1)^5-920064*last_aw(2)*last_aw(1)^7-...
257     335328*last_aw(2)*last_aw(1)^6-...
258     184800*last_aw(2)*last_aw(1)^10-...
259     754368*last_aw(2)*last_aw(1)^9-...
260     1199424*last_aw(2)*last_aw(1)^8-231*last_aw(2)^10*last_aw(1)-...
261     103488*last_aw(2)^7*last_aw(1)^4-...
262     287904*last_aw(2)^7*last_aw(1)^3-...
263     282168*last_aw(2)^7*last_aw(1)^2-...
264     112320*last_aw(2)^7*last_aw(1)-...
265     572880*last_aw(2)^4*last_aw(1)^7-...
266     2031552*last_aw(2)^4*last_aw(1)^6-...
267     2735232*last_aw(2)^4*last_aw(1)^5+...
268     64512*last_aw(2)^6*last_aw(1)^2+...
269     885120*last_aw(2)^3*last_aw(1)^6+...
270     241920*last_aw(2)^3*last_aw(1)^5+...
271     1620480*last_aw(2)^3*last_aw(1)^7+...
272     808032*last_aw(2)^6*last_aw(1)^4+...
273     338304*last_aw(2)^6*last_aw(1)^3+...
274     10872*last_aw(2)^9*last_aw(1)^4+...
275     34896*last_aw(2)^9*last_aw(1)^3+...
276     40824*last_aw(2)^9*last_aw(1)^2+20448*last_aw(2)^9*last_aw(1)+...
277     981792*last_aw(2)^6*last_aw(1)^5+...
278     1595520*last_aw(2)^3*last_aw(1)^8+6*last_aw(2)^12*last_aw(1)+...
279     810240*last_aw(1)^9*last_aw(2)^3+...
280     593184*last_aw(1)^6*last_aw(2)^6+...
281     141792*last_aw(1)^7*last_aw(2)^6+...
282     167040*last_aw(1)^10*last_aw(2)^3-...
283     225*last_aw(2)^10-14568*last_aw(2)^7+134400*last_aw(1)^10+...
284     86400*last_aw(1)^9+28800*last_aw(1)^8+9600*last_aw(1)^13+...
285     51840*last_aw(1)^12+115200*last_aw(1)^11+3648*last_aw(2)^9+...
286     6*last_aw(2)^12-889*last_aw(2)^2-9612*last_aw(2)^6+...
287     9527*last_aw(2)^4-25032*last_aw(1)^4-126504*last_aw(1)^5-...
288     90240*last_aw(1)^7-194832*last_aw(1)^6-...
289     4158*last_aw(2)^2*last_aw(1)-25032*last_aw(2)^3*last_aw(1)-...
290     223692*last_aw(2)^3*last_aw(1)^2-...
291     471408*last_aw(2)^3*last_aw(1)^3-...
292     257280*last_aw(2)^3*last_aw(1)^4-7920*last_aw(2)^6*last_aw(1)+...
293     15869*last_aw(2)^4*last_aw(1)+13076*last_aw(2)*last_aw(1)^2+...
294     63700*last_aw(2)*last_aw(1)^3+21*last_aw(1)+...
295     63476*last_aw(2)*last_aw(1)^4);
296 end
297
298 if (select_control >= 4) %r4
299     evaluate_control = evaluate_control -...

```



```

300     1/720*T^4*(6*last_aw(2)^8+48*last_aw(2)^5+6*last_aw(2)^8*last_aw(1)+...
301         648*last_aw(2)^5*last_aw(1)^4+1728*last_aw(2)^5*last_aw(1)^3+...
302         1560*last_aw(2)^5*last_aw(1)^2+528*last_aw(2)^5*last_aw(1)+...
303         1200*last_aw(1)^7*last_aw(2)^2+4080*last_aw(1)^6*last_aw(2)^2+...
304         480*last_aw(1)^3*last_aw(2)^2+2640*last_aw(1)^4*last_aw(2)^2+...
305         5040*last_aw(1)^5*last_aw(2)^2-155*last_aw(2)^2-...
306         99*last_aw(2)^6+343*last_aw(2)^4-552*last_aw(1)^4-...
307         1944*last_aw(1)^5-840*last_aw(1)^7-2232*last_aw(1)^6-...
308         360*last_aw(2)^2*last_aw(1)-552*last_aw(2)^3*last_aw(1)-...
309         3312*last_aw(2)^3*last_aw(1)^2-5280*last_aw(2)^3*last_aw(1)^3-...
310         2520*last_aw(2)^3*last_aw(1)^4-105*last_aw(2)^6*last_aw(1)+...
311         430*last_aw(2)^4*last_aw(1)+820*last_aw(2)*last_aw(1)^2+...
312         2468*last_aw(2)*last_aw(1)^3+15*last_aw(1)+...
313         1720*last_aw(2)*last_aw(1)^4);
314 end
315
316 if (select_control>=3)%r3
317     evaluate_control = evaluate_control +...
318     1/24*T^3*(24*last_aw(1)^6+...
319         40*last_aw(2)^3*last_aw(1)^2+8*last_aw(1)^4-...
320         10*last_aw(2)^4*last_aw(1)-9*last_aw(2)^4-...
321         28*last_aw(2)*last_aw(1)^2+last_aw(2)^6+8*last_aw(1)^7-...
322         68*last_aw(2)*last_aw(1)^3-40*last_aw(2)*last_aw(1)^4+...
323         24*last_aw(1)^5+56*last_aw(2)^3*last_aw(1)^3+...
324         17*last_aw(2)^2*last_aw(1)+24*last_aw(2)^3*last_aw(1)^4+...
325         last_aw(2)^6*last_aw(1)+8*last_aw(2)^3*last_aw(1)-...
326         2*last_aw(1)+10*last_aw(2)^2);
327 end
328
329 if (select_control>=2)%r2
330     evaluate_control = evaluate_control -...
331     1/12*T^2*(3*last_aw(1)+8*last_aw(2)*last_aw(1)^2-...
332         9*last_aw(2)^2*last_aw(1)+8*last_aw(2)*last_aw(1)^4+...
333         16*last_aw(2)*last_aw(1)^3+2*last_aw(2)^4*last_aw(1)-...
334         7*last_aw(2)^2+2*last_aw(2)^4);
335 end
336
337 if (select_control>=1)
338     u = evaluate_control -last_aw(1)-1+...
339         T/2*(last_aw(1)*last_aw(2)^2+last_aw(2)^2-last_aw(1));
340 else
341     u = -last_aw(1)-1;
342 end

```

### A.3 Matlabfehler

Auffällig bei der Implementierung des Programms ist, dass sowohl die Abstände, gemessen in der euklidischen Norm, als auch die Residuen nach der Optimierung separat berechnet werden. Betrachtet man aber die Parameter der Matlab-Routine `nonlinsq` genauer, stellt man fest, dass man sich diese beiden Ausdrücke als Ergebnis der Optimierung mit ausgeben lassen kann. Es stellt sich also die Frage, warum dieser einfachere Weg für die Umsetzung des Programms nicht genutzt worden ist. Dafür betrachten wir das folgende Beispiel aus der Matlab-Hilfe, an dem dort die Routine `nonlinsq` demonstriert wird. Es lautet

$$\min_{x \in \mathbb{R}^2} \sum_{k=1}^{10} (2 + 2k - e^{kx_1} - e^{kx_2})^2, \quad (\text{A.1})$$

der Startwert ist  $(0.3, 0.4)^T$ . Möchte man nun die erwähnte Matlab-Routine zur Lösung dieses nichtlinearen Ausgleichsproblem verwenden, sind die einzelnen „Abstände“  $2+2k-e^{kx_1}-e^{kx_2}$ . Das ergibt für den Startwert folgenden Vektor mit zehn Komponenten

$$\begin{pmatrix} 1.1583 \\ 1.9523 \\ 2.2203 \\ 1.7269 \\ 0.1293 \\ -3.0728 \\ -8.6108 \\ -17.5557 \\ -31.4780 \\ -52.6837 \end{pmatrix}$$

mit Residuum 4171.3. Dagegen liefert uns die Optimierungsroutine mit 0 Iterationen zwar den gleichen  $x$ -Wert zurück, allerdings andere Werte für die „Abstände“ und das Residuum:

$$1000 \cdot \begin{pmatrix} 0.0009 \\ 0.0006 \\ -0.0020 \\ -0.0095 \\ -0.0270 \\ -0.0653 \\ -0.1466 \\ -0.3166 \\ -0.6702 \\ -1.4031 \end{pmatrix}$$

und  $1000 \cdot 2544.7$ . Analog verhält es sich nach einer Iteration der Matlab-Routine, hier ergeben sich die Ungleichungen

$$\begin{pmatrix} 1.1577 \\ 1.9528 \\ 2.2258 \\ 1.7462 \\ 0.1794 \\ -2.9603 \\ -8.3793 \\ -17.1060 \\ -30.6398 \\ -51.1695 \end{pmatrix} \neq \begin{pmatrix} 1.3068 \\ 2.3729 \\ 3.1143 \\ 3.4181 \\ 3.1316 \\ 2.0489 \\ -0.1077 \\ -3.7132 \\ -9.2739 \\ -17.4731 \end{pmatrix}$$

und  $3941.9 \neq 447.8405$ . Die Resultate stimmen folglich nicht überein. Entscheidend ist an dieser Stelle, dass die neu berechneten Werte zur Lösung der Optimierungsroutine passen, wohingegen die von der Matlab-Routine selbst gelieferten Werte nicht zu dem Ergebnis selbiger gehören. Untersucht man die Matlab-Routine genauer, bemerkt man, dass in der Optimierungsroutine in jedem Iterationsschritt eine zusätzliche Funktionsauswertung durchgeführt wird, dessen Ergebnis Matlab ausgibt. Allerdings geschieht das ohne entsprechende Veränderung des Optimums. Die Entscheidung, ob das Optimum oder die beiden oben diskutierten Werte falsch sind, fällt anhand der Optimierung ohne eine einzige Iteration. In dieser kann sich das Optimum nicht ändern, was es auch nicht tut. Folglich stimmen die gelieferten Optimalwerte, nicht jedoch die angegebenen „Abstände“ und Residuen. Dies ist der Grund, warum eine separate Implementierung in `go.m` erfolgt ist.



# Anhang B

## Tabellen

An dieser Stelle werden einige Tabellen abgebildet, auf die in Kapitel 7 Bezug genommen wird.

### B.1 Akademisches Beispiel

| Iterationen | Standard         | letzter Punkt doppelt | letzter Punkt halb |
|-------------|------------------|-----------------------|--------------------|
| 1           | 0.03015493068000 | 0.02093123139474      | 0.05585563320683   |
| 2           | 0.00267506214598 | 0.00310738437677      | 0.00265123110582   |
| 3           | 0.00220641665847 | 0.00266745961441      | 0.00220415758290   |

Tabelle B.1: Residuen der Nachoptimierung ausgehend von (6.3) mit verschiedener Gewichtung und acht Auswertungspunkten.

| Iterationen | Standard         | letzter Punkt doppelt | letzter Punkt halb |
|-------------|------------------|-----------------------|--------------------|
| 1           | 0.01182913454845 | 0.00288455441567      | 0.06362164282846   |
| 2           | 0.00220489413848 | 0.00265255416069      | 0.00225990708550   |
| 3           | 0.00220602160632 | 0.00266704130374      | 0.00220417708202   |

Tabelle B.2: Residuen der Nachoptimierung ausgehend von (6.6) mit verschiedener Gewichtung und acht Auswertungspunkten.

### B.2 Moore-Greitzer Modell

| max zugelassene<br>Iterationen | Gauß-Newton    |                 | Levenberg-Marquardt |                 |
|--------------------------------|----------------|-----------------|---------------------|-----------------|
|                                | Anfangswert: 0 | Anfangswert: SD | Anfangswert: 0      | Anfangswert: SD |
| 1                              | -2.4045        | -3.4037         | -2.4033             | -3.4014         |
|                                | 0.0000         | 1.0000          | 3.0643              | 1.0000          |
|                                | 0.0000         | 1.0000          | 2.3444              | 1.0000          |
|                                | 0.0000         | 1.0000          | 2.1760              | 1.0000          |
|                                | 0.0000         | 1.0000          | 2.5229              | 13.9591         |
| 2                              | -1.8028        | -2.3009         | -1.9118             | -3.0248         |
|                                | -0.0891        | 1.4376          | -3.9233             | 1.0973          |
|                                | 0.4021         | 1.0628          | 1.7379              | 1.0808          |
|                                | 0.5831         | 1.7756          | 1.9692              | 1.1581          |
|                                | 0.6668         | -1.2263         | 2.4049              | 30.2531         |
| 3                              | -1.2026        | -1.2027         | -1.4119             | -1.9572         |
|                                | 0.7367         | 0.7364          | 0.5824              | -5.0808         |
|                                | 0.6924         | 0.6911          | 0.8762              | 1.1364          |
|                                | 0.6860         | 0.6749          | 0.5170              | 0.9872          |
|                                | 0.6044         | 0.5719          | 0.8618              | 0.6329          |
| 4                              | -1.2025        | -1.2025         | -1.2334             | -1.3587         |
|                                | 0.7366         | 0.7367          | 0.7194              | 0.6166          |
|                                | 0.6917         | 0.6925          | 0.7183              | 0.7865          |
|                                | 0.6868         | 0.6738          | 0.6991              | 0.7419          |
|                                | 0.6033         | 0.5746          | 0.6597              | 1.0749          |
| 5                              | -1.2025        | -1.2025         | -1.2047             | -1.2187         |
|                                | 0.7366         | 0.7366          | 0.7355              | 0.7270          |
|                                | 0.6919         | 0.6919          | 0.6937              | 0.7062          |
|                                | 0.6838         | 0.6826          | 0.6851              | 0.6911          |
|                                | 0.5907         | 0.5880          | 0.5960              | 0.6256          |
| 6                              | -1.2025        | -1.2025         | -1.2026             | -1.2033         |
|                                | 0.7366         | 0.7366          | 0.7366              | 0.7362          |
|                                | 0.6919         | 0.6919          | 0.6919              | 0.6927          |
|                                | 0.6835         | 0.6834          | 0.6840              | 0.6825          |
|                                | 0.5881         | 0.5900          | 0.5912              | 0.5890          |
| beliebig                       | -1.2025        | -1.2025         | -1.2025             | -1.2025         |
|                                | 0.7366         | 0.7366          | 0.7366              | 0.7366          |
|                                | 0.6919         | 0.6919          | 0.6919              | 0.6919          |
|                                | 0.6835         | 0.6835          | 0.6835              | 0.6835          |
|                                | 0.5893         | 0.5893          | 0.5893              | 0.5893          |

Tabelle B.3: Vergleich der Optimalsteuerung mit den Werten des sampled-data Feedbacks.  
Linesearch: cubicpoly.

| max.<br>Iter. | Gauß-Newton    |    |    |    |    | Levenberg-Marquardt |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---------------|----------------|----|----|----|----|---------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|               | Anfangswert: 0 |    |    |    |    | Anfangswert: SD     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 0             | 2              |    |    |    |    | 2                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 1             | 4              |    |    |    |    | 4                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2             | 6              |    |    |    |    | 6                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 3             | 8              |    |    |    |    | 8                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 4             | 10             |    |    |    |    | 10                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 5             | 10             | 12 | 12 | 12 | 12 | 12                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 6             | 10             | 12 | 14 | 14 | 14 | 12                  | 12 | 14 | 14 | 14 | 14 |    |    |    |    |    |    |    |    |    |
| bel.          | 10             | 12 | 14 | 16 | 46 | 12                  | 12 | 14 | 16 | 20 | 18 | 16 | 28 | 24 | 24 | 20 | 12 | 30 | 24 | 24 |

Tabelle B.4: Vergleich des Aufwandes gemessen in Funktionsauswertungen der beiden Algorithmen mit bzw. ohne Anfangswert. Linesearch: cubicpoly.

| max.<br>Iter. | Gauß-Newton    |    |    |    |    | Levenberg-Marquardt |    |    |    |    |      |    |    |    |    |    |    |    |    |    |
|---------------|----------------|----|----|----|----|---------------------|----|----|----|----|------|----|----|----|----|----|----|----|----|----|
|               | Anfangswert: 0 |    |    |    |    | Anfangswert: SD     |    |    |    |    |      |    |    |    |    |    |    |    |    |    |
| 0             | 2              |    |    |    |    | 2                   |    |    |    |    |      |    |    |    |    |    |    |    |    |    |
| 1             | 8              | 7  | 7  | 7  | 7  | 8                   | 7  | 7  | 7  | 7  | 8    | 7  | 7  | 8  | 7  | 8  | 7  | 7  | 7  | 7  |
| 2             | 13             | 12 | 12 | 12 | 12 | 13                  | 12 | 12 | 12 | 12 | 13   | 13 | 12 | 12 | 12 | 13 | 13 | 12 | 12 | 12 |
| 3             | 13             | 17 | 12 | 12 | 12 | s.o.                |    |    |    |    | s.o. |    |    |    |    |    |    |    |    |    |

Tabelle B.5: Vergleich des Aufwandes gemessen in Funktionsauswertungen der beiden Algorithmen mit bzw. ohne Anfangswert. Linesearch: quadcubic.





# Anhang C

## Material auf der beiliegenden CD

Auf der beiliegenden CD sind folgende drei Verzeichnisse enthalten:

- **Abbildungen**,
- **Diplomarbeit** und
- **Matlab**.

Das Verzeichnis **Abbildungen** beinhaltet die Abbildungen aus den Kapiteln 6 und 7 im eps-Format. Die Bezeichnung der einzelnen Abbildungen orientiert sich an der Numerierung in den angeführten Kapiteln. Der Ordner **Diplomarbeit** enthält eine pdf-Version dieser Arbeit. Die in Anhang A erwähnten Matlab-Files befinden sich in dem Verzeichnis **Matlab**. Für den Programmaufruf und die Einstellung einzelner Parameter dient das File `opttest.m`. Für weitere Informationen bezüglich der einzelnen Files sei auf Anhang A verwiesen.



# Literaturverzeichnis

- [1] Hans Benker, *Mathematik mit MATLAB: Eine Einführung für Ingenieure und Naturwissenschaftler*; Springer, 2000.
- [2] Peter Deuffhard, Andreas Hohmann, *Numerische Mathematik I*; 3. Auflage, de Gruyter Lehrbuch, Berlin, 2002.
- [3] Gerd Fischer, *Lineare Algebra*; zwölfte Auflage, Vieweg Verlag, September 2000.
- [4] Otto Forster, *Analysis 1*; fünfte Auflage, Vieweg Verlag, 1999.
- [5] Otto Forster, *Analysis 2*; fünfte Auflage, Vieweg Verlag, 1999.
- [6] Lars Grüne, *Numerische Mathematik I*; Ausarbeitung einer Vorlesung gehalten im Wintersemester 2002/03 an der Universität Bayreuth.
- [7] Lars Grüne, *Numerische Mathematik II: Differentialgleichungen*; Ausarbeitung einer Vorlesung gehalten im Sommersemester 2003 an der Universität Bayreuth.
- [8] Lars Grüne, *Numerik Dynamischer Systeme*; Ausarbeitung einer Vorlesung gehalten im Wintersemester 2003/04 an der Universität Bayreuth.
- [9] Lars Grüne, *Numerische Dynamik von Kontrollsystemen*; Ausarbeitung einer Vorlesung gehalten im Sommersemester 2004 an der Universität Bayreuth.
- [10] Lars Grüne, P.E. Kloeden, *Higher order numerical schemes for affinely controlled nonlinear systems*; Numer. Math., 89 (2001), pp. 669-690.
- [11] Lars Grüne, Karl Worthmann, D. Nešić, *Continuous-time controller redesign for digital implementation: a trajectory based approach*; Preprint, Universität Bayreuth, Februar 2006, submitted.
- [12] A. Isidori, *Nonlinear Control Systems*; 3rd ed, Springer Verlag, London, 2002.
- [13] Hassan K. Khalil, *Nonlinear Systems*; Third Edition, Prentice Hall, 2002.

- [14] Michael Kofler, Gerhard Bitsch und Michael Komma, *Maple - Einführung, Anwendung, Referenz*; Addison-Wesley Longman Verlag, 2002.
- [15] V. Krishnamurthy, *Combinatorics: theory and applications*; Affiliated East-West Press: Madras 1985.
- [16] M. Krstić, I. Kanellakopoulos, P.V. Kokotović, *Nonlinear and adaptive control design*; John Wiley & Sons, New York, 1995.
- [17] Florian Jarre, Josef Stoer, *Optimierung*; Springer, 2004.
- [18] J.L. Mohamed, J. Walsh, *Numerical Algorithms*; Oxford Science Publications, Oxford, 1986.
- [19] S. Monaco und D. Normand-Cyrot, *Issues on nonlinear digital control*; Europ. J. Control, 7 (2001), pp. 160-178.
- [20] S. Monaco und D. Normand-Cyrot, *Input-state matching under digital control*; Preprint submitted to 45th IEEE Conference on Decision and Control, Received March 3, 2006.
- [21] D. Nešić, Lars Grüne, *Lyapunov based continuous-time nonlinear controller redesign for sampled-data implementation*; Automatica, 41 (2005), pp. 1143-1156.
- [22] D. Nešić und Andrew R. Teel, *A Framework for Stabilization of Nonlinear Sampled-Data Systems Based on Their Approximate Discrete-Time Models*; IEEE Transactions on automatic control, VOL. 49, NO. 7, JULY 2004.
- [23] D. Nešić, A.R. Teel and P.V. Kokotović, *Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete-time approximations*; Syst. Control Lett, 38 (1999), pp. 259-270.
- [24] Jürgen Pannek, *Modellprädiktive Regelung nichtlinearer sampled-data Systeme*; Diplomarbeit abgegeben am 17. Februar 2005.
- [25] P. Spellucci, *Numerische Verfahren der nichtlinearen Optimierung*; Birkhäuser, Basel, 1993.
- [26] Ekue-sse Situ Tomety, Karl Worthmann, *Eine Einführung in die Stabilisierung nichtlinearer Sampled-data Systeme, aufbauend auf dem approximierten zeitdiskreten Modell*; Seminararbeit, Wintersemester 2004/05, Seminar geleitet von Lars Grüne.

# Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

Bayreuth, den 17. Juli 2006

.....  
Karl Worthmann