

**UNIVERSITÄT
BAYREUTH**

Modellprädiktive Regelung nichtlinearer sampled-data Systeme

Diplomarbeit

von

Jürgen Pannek

FAKULTÄT FÜR MATHEMATIK UND PHYSIK
MATHEMATISCHES INSTITUT

Datum: 17. Februar 2005

Aufgabenstellung / Betreuung:
Prof. Dr. L. Grüne

Zweiter Gutachter:
Prof. Dr. F. Lempio

Danksagung

An dieser Stelle möchte ich mich bei Herrn Grüne für die hervorragende Betreuung während der Erstellung dieser Arbeit bedanken.

Ebenso danke ich Herrn Büskens und Herrn Gerdts dafür, dass sie in mir das Interesse für den Themenbereich der optimalen Steuerung geweckt und mich während des Studiums laufend motiviert haben.

Ein besonderer Dank geht auch an meine Eltern, die mich während des gesamten Studiums unterstützt haben.

Nicht zu vergessen sind außerdem Frau Becker und Frau Alka sowie die Herren Schick und Blume, die mir in dieser Zeit mir Rat und Tat, fachlich wie privat, beistanden.

Inhaltsverzeichnis

Tabellenverzeichnis	III
Abbildungsverzeichnis	VI
1 Einleitung	1
2 Kontrollsysteme	5
2.1 Grundlegende Definitionen	5
2.2 Definition eines Kontrollsystems	11
2.3 Existenz und Eindeutigkeit	12
2.4 Asymptotische Kontrollierbarkeit	16
2.5 Feedback-Stabilisierbarkeit	17
2.6 Lyapunov-Kontroll-Funktionen	19
2.7 Sampled-data Lösungen	21
3 Optimale Steuerprozesse	25
3.1 Definition eines unbeschränkten optimalen Steuerprozesses	26
3.2 Bellmansches Optimalitätsprinzip und Hamilton-Jacobi-Bellman Gleichung	27
3.3 Modellprädiktive Regelung nichtlinearer sampled-data Systeme	28
3.3.1 Grundannahmen	29
3.3.2 Problemstellung des modellprädiktiven Regelungsansatzes	30
3.4 Indirekte Verfahren zur Lösung optimaler Steuerprozesse	32
3.5 Direkte Verfahren zur Lösung optimaler Steuerprozesse	35
3.5.1 Lösung mittels voller Diskretisierung	37
3.5.2 Lösung mittels eines rekursiven Ansatzes	38
3.5.3 Konvergenz der Euler-Diskretisierung	39
3.6 Numerische Lösung optimaler Steuerprozesse	45
4 Stabilität	51
4.1 Stabilitätseigenschaften zeitdiskreter modellprädiktiver Regelungsverfahren . .	51
4.1.1 Stabilität des zeitdiskreten Verfahrens	52

4.1.2	Stabilitätszusammenhänge zwischen approximiertem und exaktem zeitdiskreten System	55
4.1.3	Stabilitätszusammenhänge zwischen sampled-data und zeitdiskretem System	62
4.2	Stabilität des modellprädiktiven Regelungsansatzes	65
5	Optimalität	75
5.1	Inverse Optimalität zeitdiskreter modellprädiktiver Regelungsverfahren . . .	76
5.1.1	Direkte Methode	76
5.1.2	Monotonie-Methode	77
5.1.3	Der zeitdiskrete Fall	78
5.2	Optimalität des modellprädiktiven Regelungsansatzes	78
6	Beispiele und Ergebnisse	83
6.1	1D Beispiel	83
	Einschub: Moore-Greitzer Modell eines Strahltriebwerks	86
6.2	Vereinfachtes Moore-Greitzer Modell	87
6.3	Moore-Greitzer Modell eines Strahltriebwerks	88
6.4	Reihenfolge zur Anpassung der Einflußfaktoren	89
6.5	Sensitivität der Lösung bei Änderung der Anfangswerte	90
6.6	Verbesserung der gesampleten Lösung durch modellprädiktive Regelung . . .	94
6.6.1	Verbesserung des Fehlers	94
6.6.2	Vergrößerung der sampled-data Rate	99
6.7	Einfluß der sampled-data Rate	102
6.8	Einfluß der Länge des Horizonts	109
6.9	Verwendung einer Lyapunov-Kontroll-Funktion	111
6.10	Änderungen der optimalen Lösungen der einzelnen optimalen Steuerprozesse	112
6.11	Fazit und Ausblick	113
A	Implementierung des modellprädiktiven Regelungsansatzes	115
A.1	Software Beschreibung von NUDOCCS	115
A.1.1	Verfahren zur Integration und Interpolation	115
A.1.2	Parameterzusammenhänge der Eingaben	118
A.1.3	Eingabe des optimalen Steuerprozesses	118
A.2	Veränderungen am Algorithmus	120
A.2.1	Berechnung der Startschätzung einer Steuerung	120
A.2.2	Der modellprädiktive Ansatz: Variablendefinition und Übergabe der Werte zur Lösung des folgenden Optimalsteuerungsproblems	122
A.2.3	Subroutine der Schrittweitensteuerung	124
A.2.4	Subroutine zur Ausgabe aller berechneter Trajektorien	136

B Fortran Programme	139
B.1 Fortran-Code des Programms und dessen Aufruf	139
B.2 Programm zu Beispiel 6.1	146
B.3 Programm zu Beispiel 6.2	150
B.4 Programm zu Beispiel 6.3	154
C Visualisierung	159
C.1 Eingabe der Differentialgleichungen	159
C.2 Matlab Programm zur Visualisierung	160
C.3 Fehlerberechnung	167
C.4 Plots der Beispiele zur Verlaufsanalyse	171
D Material auf der beiliegenden CD	189
Literaturverzeichnis	193

Tabellenverzeichnis

6.1	Fehler in den Lösungen auf dem Intervalls $[0, 10]$ bei modellprädiktiver Regelung des Beispiels 6.1 mit unterschiedlichen Anfangswerten	91
6.2	Fehler durch die gesamplete zeitkontinuierliche Steuerung des Beispiels 6.2	94
6.3	Fehler in den Lösungen durch die modellprädiktive Regelung des Beispiels 6.2	95
6.4	Fehler in den Lösungen durch die gesamplete zeitkontinuierliche Steuerung des Beispiels 6.3	95
6.5	Fehler in den Lösungen durch die modellprädiktive Regelung des Beispiels 6.3	96
6.6	Verbesserung durch den modellprädiktiven Regelungsansatz bei Vergrößerung der sampled-data Rate	99
6.7	Fehler in den Lösungen durch die modellprädiktive Regelung des Beispiels 6.3 für unterschiedliche sampled-data Raten und Horizontlängen	102
6.8	Fehler in den Lösungen durch die modellprädiktive Regelung des Beispiels 6.3 für unterschiedliche sampled-data Raten	103
6.9	Vergleich der Fehler in den Lösungen durch die modellprädiktive Regelung des Beispiels 6.3 für unterschiedliche sampled-data Raten und Fehlertoleranzen	106
6.10	Vergleich der Fehler in den Lösungen durch die modellprädiktive Regelung des Beispiels 6.3 für unterschiedliche Horizontlängen bei kritischer sampled-data Rate	109
6.11	Vergleich der Fehler in den Lösungen durch die modellprädiktive Regelung des Beispiels 6.3 für unterschiedliche Horizontlängen	110
6.12	Abweichungen der x - und ξ -Trajektorien des Beispiels 6.3 für unterschiedliche Horizontlängen	111
6.13	Fehler in den Lösungen durch die modellprädiktive Regelung des Beispiels 6.2 für unterschiedliche sampled-data Raten	112
6.14	Fehler in den Lösungen durch die modellprädiktive Regelung des Beispiels 6.2 für unterschiedliche sampled-data Raten und Horizontlängen	112
6.15	Rechenzeitvergleich	113
A.1	Interpolations- und Integrationsverfahren in NUDOCCCS	116
A.2	Interpolations- und Integrationsverfahren für modellprädiktive Regelung	117
A.3	Neueingeführte Parameter	122

A.4	Bedeutung der Dateinamen aus dem Verfahren der modellprädiktiven Regelung	136
B.1	Bedeutung der Dateinamen aus NUDOCCCS	145
B.2	Eingabe der Problemstellung	146

Abbildungsverzeichnis

2.1	Vergleichsfunktionen der Klassen \mathcal{G} , \mathcal{K} und \mathcal{L}	6
2.2	Sampled-data Folge und sampled-data Rate	21
3.1	Ansätze zur Lösung optimaler Steuerprozesse	25
3.2	Vorgehensweise des modellprädiktiven Regelungsansatzes	31
6.1	Musterlösung des 1D-Beispiels	85
6.2	Lösung des 1. optimalen Steuerungsproblems des 1D Beispiels für $T = 1.0$ und $H = 5.0$ bei Anfangswert $x_0 = 0.439$	92
6.3	Lösung des 1D Beispiels für $T = 1.0$ und $H = 5.0$ bei Anfangswert $x_0 = 0.25$	93
6.4	Lösung des Beispiels 6.3 für $T = 0.05$ und $H = 0.5$ mit zeitkontinuierlicher, gesampter und modellprädiktiver Steuerung	98
6.5	Lösung des Beispiels 6.3 für $T = 0.045$ und $H = 0.45$ mit zeitkontinuierlicher, gesampter und modellprädiktiver Steuerung mit geänderten Koeffizienten .	101
6.6	Fehlerentwicklungen für verschiedene sampled-data Raten	103
6.7	Falsche Integration bei der Lösung des Beispiels 6.3 für $T = 0.095$ und $H =$ 0.95 für $Tol = 10^{-4}$	105
6.8	Lösung des Beispiels 6.3 für $T = 0.095$ und $H = 0.95$ für $Tol = 10^{-2}$	108
6.9	Rechenzeitverlauf für kritische sampled-data Rate	114
6.10	Rechenzeitverlauf für unkritische sampled-data Rate	114
C.1	Beispiels 6.3 für $T = 0.05$ und $H = 1.00$	171
C.2	Beispiels 6.3: 1. optimales Steuerungsproblem	172
C.3	Beispiels 6.3: 2. optimales Steuerungsproblem	173
C.4	Beispiels 6.3: 3. optimales Steuerungsproblem	174
C.5	Beispiels 6.3: 4. optimales Steuerungsproblem	175
C.6	Beispiels 6.3: 5. optimales Steuerungsproblem	176
C.7	Beispiels 6.3: 10. optimales Steuerungsproblem	177
C.8	Beispiels 6.3: 15. optimales Steuerungsproblem	178
C.9	Beispiels 6.3: 20. optimales Steuerungsproblem	179
C.10	Beispiels 6.3 für $T = 0.085$ und $H = 1.70$	180
C.11	Beispiels 6.3: 1. optimales Steuerungsproblem	181

C.12 Beispiels 6.3: 2. optimales Steuerungsproblem	182
C.13 Beispiels 6.3: 3. optimales Steuerungsproblem	183
C.14 Beispiels 6.3: 4. optimales Steuerungsproblem	184
C.15 Beispiels 6.3: 5. optimales Steuerungsproblem	185
C.16 Beispiels 6.3: 10. optimales Steuerungsproblem	186
C.17 Beispiels 6.3: 15. optimales Steuerungsproblem	187
C.18 Beispiels 6.3: 20. optimales Steuerungsproblem	188

Kapitel 1

Einleitung

In dieser Arbeit wird ein Verfahren zur modellprädiktiven Regelung nichtlinearer sampled-data Systeme entwickelt. Zweck dieses Verfahrens ist es dabei, eine stückweise konstante Steuerfunktion zu berechnen, die zu einem gewünschten Langzeitverhalten eines kontinuierlichen Systems führt. Eine derartige Steuerung ermöglicht es mit Hilfe von digitalen Rechner, die nur zeitdiskret arbeiten können, kontinuierliche Systeme zu steuern, wie sie in der Natur vorkommen. Da die Berechnung *on-line* durchgeführt wird, also zu jedem Zeitpunkt der sampled-data Folge, ist es möglich dieses Verfahren auszubauen, so dass auf aktuelle Störungen durch Updates der Anfangszustände eingegangen werden kann, die etwa durch Ungenauigkeiten des Modells oder andere Fehlerquellen verursacht werden.

In einem ersten Schritt wird dabei die *Steuerung* für einen unbeschränkten optimalen Steuerprozess auf einem Horizont fester Länge mit Hilfe eines direkten Verfahrens bestimmt. Dabei werden zwei Zustandssysteme betrachtet, eines mit zeitkontinuierlich implementierter Steuerung und eines mit zu bestimmender stückweise konstanter Steuerung. Diese Steuerung liefert im zweiten Schritt eine *Regelung*. Dazu wird nur das erste Folgenglied dieser stückweise konstanten Funktion implementiert und anschließend der Horizont um die sampled-data Rate verschoben. Somit können die beiden Schritte iterativ angewandt werden und man erhält die gewünschte Lösung. Der Begriff der *Modellprädiktivität* rührt dabei daher, dass ein wesentlich längerer Horizont betrachtet wird als dies für die Implementierung lediglich eines Folgengliedes der Steuerung notwendig wäre.

Gewählt wurde dieser Ansatz, da damit eine numerischen Berechnung auch für große nicht-lineare Systeme möglich ist. Andere Ansätze benötigen etwa die meist unmöglich zu bestimmende Lösung der Hamilton-Jacobi-Bellman Gleichung oder verwenden die Diskretisierung einer zeitkontinuierlichen Steuerung, bei deren Durchführung die sampled-data Rate bereits im linearen Fall außerhalb technischer Möglichkeiten liegen kann.

Neben der Untersuchung einer derartigen Regelung in Hinsicht auf Optimalität der Regelung und Stabilität des geregelten Systems wird explizit auf die numerische Lösung der einzelnen optimalen Steuerprozesse eingegangen.

Der Aufbau dieser Arbeit gliedert sich in vier Hauptabschnitte. Im ersten dieser Abschnitte werden für Kontrollsysteme die grundlegenden Definitionen und Begriffe eingeführt und Zusammenhänge der Sätze für derartige Systeme gezeigt. Dazu wird in Kapitel 2 ausgehend von dynamischen Systemen zunächst die Definition eines Kontrollsystems gegeben. Im Weiteren wird auf die Existenz und Eindeutigkeit einer Lösung dieses Systems sowie auf die Kontrollierbarkeit eingegangen und der Begriff eines Zustandsfeedbacks eingeführt, wobei auch im Kurzen die Zusammenhänge erläutert werden. Zum Abschluss werden sampled-data Systeme definiert und auf die Stabilität dieser mit unstetigem Zustandfeedback gesteuerten Systeme eingegangen. Dabei spielt der zuvor eingeführte Begriff einer Lyapunov-Funktion eine wichtige Rolle.

Der zweite Hauptabschnitt, bestehend aus Kapitel 3, basiert auf den Arbeiten von Ch.Büskens [2], K.Malanowski, Ch.Büskens und H. Maurer [16] und P.E.Gill, W.Murray und M.H.Wright [6]. Nach der Definition eines optimalen Steuerprozesses werden darin die Theorie und die Numerik derartiger Probleme dargelegt. Zudem wird dabei auf die Problemstellung des modellprädiktiven Regelungsansatzes nichtlinearer sampled-data Systeme, entsprechend der Arbeit von D.Nešić und L.Grüne [23], eingegangen und an relevanten Stellen der Theorie werden Anmerkungen zur praktischen Implementierung dieses Ansatzes gegeben. Neben dem ausführlich erläuterten Ansatz der direkten Verfahren wird zudem auf indirekte Verfahren mit ihren Vor- und Nachteilen eingegangen. Anschließend wird ein Ergebnis zur Konvergenz der Eulerdiskretisierung im Rahmen eines direkten Verfahrens gegeben, wobei zudem die Varianten der vollständigen Diskretisierung und des rekursiven Ansatzes erwähnt werden. Zum Abschluss dieses Abschnitts wird mit Hilfe der notwendigen und hinreichenden Bedingungen der Optimalität des aus der Diskretisierung entstandenen Optimierungsproblems ein numerisches Lösungsverfahren, ein sogenanntes SQP-Verfahren, dargelegt.

Die Theorie zu Stabilität und Optimalität des modellprädiktiven Regelungsansatzes wird im dritten Hauptabschnitt behandelt. Dieser besteht aus den Kapiteln 4 und 5 und basiert auf den Arbeiten von G.Grimm, M.J.Messina, A.R.Teel, S.Tuna [7], D.Nešić, A.R.Teel, P.V.Kokotović [20, 21] und D.Nešić und L.Grüne [23].

Kapitel 4 geht dabei zunächst auf Stabilitätsergebnisse ein, die für eine derartige modellprädiktive Regelung bei Vorliegen exakter zeitdiskreter Systeme gezeigt werden konnten. Da allerdings meist keine exakten zeitdiskreten Systeme vorliegen wird danach der Schluss von der Stabilität des approximierten zeitdiskreten Systems auf die Stabilität des exakten zeitdiskreten Systems bewiesen, wobei die Steuerung eine Konsistenzbedingung erfüllen muss. Anschließend werden die Stabilitätszusammenhänge zwischen exakten zeitdiskreten und sampled-data Systemen gezeigt. Hauptergebnis dieses Kapitels ist dabei, dass unter annehmbaren Voraussetzungen die semiglobal praktisch asymptotische Stabilität des modellprädiktiv geregelten kontinuierlichen closed-loop sampled-data Systems bewiesen werden kann.

Weiter wird in Kapitel 5 die inverse Optimalität der vorliegenden modellprädiktiven Regelung unter gewissen Annahmen gezeigt. Zudem wird darauf eingegangen, warum der Mayer-

Anteil des Zielfunktional nicht notwendigerweise eine Lyapunov-Kontroll-Funktion des exakten zeitdiskreten closed-loop sampled-data Systems sein muss.

Der letzte Abschnitt besteht aus Kapitel 6 und befasst sich mit der Anwendung des in Kapitel 3 beschriebenen Verfahrens auf die Problemstellung. Darin wird mittels der Theorie der Kontrollsysteme und der Theorie über die Stabilität des modellprädiktiven Regelungsansatzes eine allgemeine Vorgehensweise zur Generierung einer Lösung formuliert und das implementierte Verfahren anhand von Beispielen untersucht. Auf Grund der praktischen Zielsetzung die sampled-data Rate möglichst groß zu wählen werden zudem die dadurch auftretenden Probleme erläutert. Um mit Hilfe der beeinflussbaren Parameter mögliche Auswege aufzeigen zu können werden dazu anhand der Beispiele die Auswirkungen von Wertänderungen dieser Parameter untersucht und zum Abschluß ein Ausblick für mögliche Verbesserungen gegeben.

Anhang A enthält Erläuterungen zu dem implementierten Verfahren, insbesondere zur Eingabe der Problemstellung, und zeigt die vorgenommenen Änderungen am Basisprogramm NUDOCCCS von Ch.Büskens auf, das er im Rahmen seines wissenschaftlichen Werdegangs entwickelt hat.

Im Anhang B sind die Programmcodes der in Kapitel 6 aufgeführten Beispiele sowie des implementierten modellprädiktiven Regelungsansatzes abgedruckt. Für die in Kapitel 6 abgebildeten Darstellungen und die Berechnung auftretender Fehler wurden die in Anhang C enthaltenen Matlab-Programme verwendet. Zusätzlich sind hier ausgegliederte Darstellungen zur Verlaufsuntersuchung der Lösung abgedruckt.

Anhang D schließlich stellt eine Auflistung der auf der beiliegenden CD enthaltenen Programme und Ergebnisse dar.

Bayreuth, den 17. Februar 2005

Kapitel 2

Kontrollsysteme

Zunächst sollen hier einige Grundbegriffe dynamischer Systeme, die einen Sonderfall der Kontrollsysteme darstellen, erklärt und auf verwendete Schreibweisen eingegangen werden. Im Weiteren wird die Problembetrachtung von gewöhnlichen Differentialgleichungen auf Kontrollsysteme erweitert. Dabei versteht man unter einem Kontrollsystem ein dynamisches System, das nicht mehr nur vom aktuellen Zustand $x(t) \in \mathbb{R}^n$ und eventuell von der Zeit $t \in \mathbb{R}$ abhängig ist, sondern auch von einem beeinflussbaren Parameter $u \in \mathbb{R}^m$.

Dieser Parameter kann verschieden interpretiert werden, zum einen als Steuerung, zum anderen als Störung, jedoch ergeben sich aus den zwei Konzepten keine strukturellen Unterschiede, so dass hier lediglich die sogenannten Kontrollsysteme untersucht werden.

Insbesondere wird hier auf die Existenz und Eindeutigkeit der Lösung sowie die Kontrollierbarkeit und Feedback-Stabilisierbarkeit eingegangen. Für die spätere Anwendung sind ebenso die Begriffe der Lyapunov-Kontroll-Funktion und der sampled-data-Lösung von Bedeutung, die hier ebenfalls kurz eingeführt werden. Der Aufbau orientiert sich dabei an dem Buch von H.K.Khalil [13] sowie den Arbeiten von L.Grüne [8, 9] und Ch.Büskens [2].

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 wird hier mit \mathbb{R} bezeichnet.

Definition 2.1. Für eine komplexe Zahl $c = a + ib \in \mathbb{C}$, $a, b \in \mathbb{R}$, sei die konjugiert komplexe Zahl $\bar{c} = a - ib$ definiert und der Betrag durch $|c| = \sqrt{c\bar{c}}$ gegeben. Für einen Vektor $x = (x_1, \dots, x_n)^T \in \mathbb{C}^n$ bezeichne

$$\|x\|_1 = \sum_{k=1}^n |x_k| \tag{2.1}$$

die Betragssummennorm,

$$\|x\| = \sqrt{\sum_{k=1}^n |x_k|^2} \quad (2.2)$$

die euklidische Norm und

$$\|x\|_\infty = \max \{|x_k|\} \quad (2.3)$$

die Maximumsnorm, wobei \cdot^T transponiert bedeutet. Für eine Matrix

$$A = \begin{pmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \cdots & a_{n,n} \end{pmatrix} \in \mathbb{C}^{n \times n} \quad (2.4)$$

wird die zugehörige induzierte Matrixnorm

$$\|A\| = \sup_{x \in \mathbb{C}^n \setminus \{0\}} \frac{\|Ax\|}{\|x\|} \quad (2.5)$$

verwendet. Weiter wird die k -fache Matrixmultiplikation von A mit sich selbst mit A^k und die Einheitsmatrix in $\mathbb{R}^{n \times n}$ bzw. $\mathbb{C}^{n \times n}$ mit Id bezeichnet. Dabei entspricht $A^0 = Id$.

Zudem werden Klassen von Funktionen, die sogenannten Vergleichsfunktionen, verwendet und wie in der Literatur [7, 8, 9, 13, 20, 21, 22, 23] üblich betitelt.

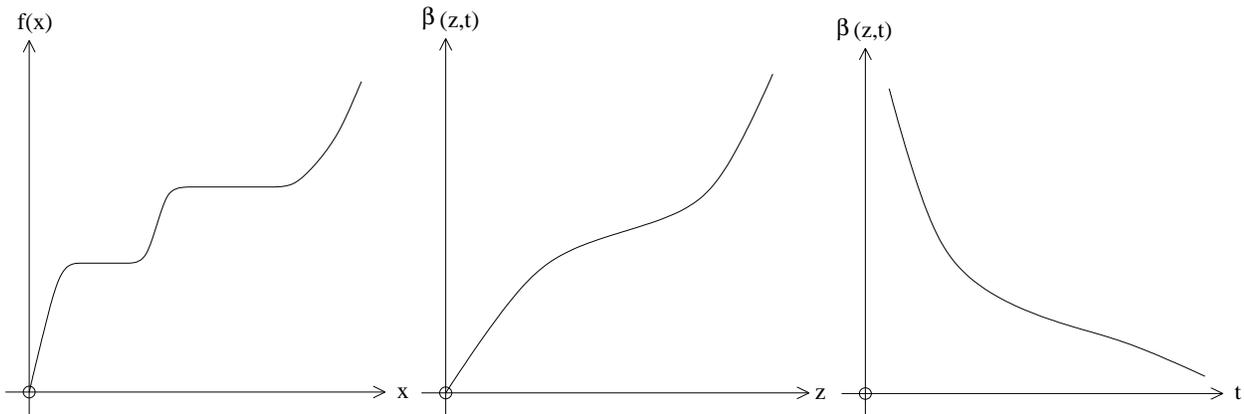


Abbildung 2.1: Vergleichsfunktionen der Klassen \mathcal{G} , \mathcal{K} und \mathcal{L}

Dabei bezeichne \mathcal{G} die Klasse der Funktionen $f : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$, die stetig und nicht-fallend sind und für die $f(0) = 0$ gilt.

Ist $f(\cdot)$ stetig, streng monoton steigend und gilt $f(0) = 0$, so wird diese Klasse mit \mathcal{K} bezeichnet.

Weiter sei $K_\infty := \{\alpha \in \mathcal{K} | \alpha \text{ ist unbeschränkt}\}$.

Mit \mathcal{L} wird die Klasse der nichtnegativen Funktionen $f : \mathbb{R} \rightarrow \mathbb{R}_0^+$ bezeichnet, für die $\lim_{x \rightarrow \infty} f(x) = 0$ gilt.

Eine Funktion $g : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$ sei aus der Klasse \mathcal{KL} , wenn für alle festen $t \geq 0$ die Funktion $g(\cdot, t)$ aus der Klasse \mathcal{K} und für alle festen $s > 0$ die Funktion $g(s, \cdot)$ aus der Klasse \mathcal{L} ist.

Zudem sei \mathcal{N} die Klasse der stetigen und monoton wachsenden Funktionen $\delta : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$.

Dabei sind folgende Eigenschaften von Vergleichsfunktionen erwähnenswert:

- (i) Sei $\alpha \in \mathcal{K}$ und $\alpha^+ := \sup_{r \geq 0} \alpha(r)$. Dann existiert eine Umkehrfunktion $\alpha^{-1} : [0, \alpha^+) \rightarrow \mathbb{R}_0^+$ mit $\alpha^{-1}(\alpha(r)) = r$ für alle $r \in \mathbb{R}, r \geq 0$ und $\alpha(\alpha^{-1}(r)) = r$ für alle $r \in [0, \alpha^+) \subset \mathbb{R}$.
- (ii) Sei $\alpha \in \mathcal{K}_\infty$. Dann ist die Umkehrfunktion α^{-1} aus (i) auf ganz \mathbb{R}_0^+ definiert und es gilt $\alpha^{-1} \in \mathcal{K}_\infty$.
- (iii) Für alle Konstanten $C, \sigma \in \mathbb{R}$ mit $C, \sigma > 0$ ist die Funktion $\beta(r, t) = Ce^{-\sigma t} r$ aus \mathcal{KL} .

Der Abstand eines Punktes $x \in \mathbb{R}^n$ von einer Menge $\mathcal{A} \subset \mathbb{R}^n$ sei durch

$$\|x\|_{\mathcal{A}} := \inf_{s \in \mathcal{A}} \|x - s\| \quad (2.6)$$

festgelegt. Zudem definiert man den Gradienten einer Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}$ bezüglich $x \in \mathbb{R}^n$ als

$$\nabla_x f = \left(\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right)^T, \quad (2.7)$$

die Hesse-Matrix von $f : \mathbb{R}^n \rightarrow \mathbb{R}$ bezüglich $x \in \mathbb{R}^n$ durch

$$\nabla_{xx} f = \nabla_x^2 f = f_{xx} = \frac{\partial^2 f}{\partial x^2} = \left(\frac{\partial^2 f}{\partial x_i \partial x_j} \right) \quad (2.8)$$

und die Jacobi-Matrix einer Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ bezüglich $x \in \mathbb{R}^n$ mittels

$$\nabla_x f = f_x = \frac{\partial f}{\partial x} = \begin{pmatrix} \frac{\partial f_1}{\partial x} \\ \vdots \\ \frac{\partial f_m}{\partial x} \end{pmatrix}. \quad (2.9)$$

Diese Definitionen bilden die Grundlage zur Untersuchung der in dieser Arbeit aufgeführten Probleme und zur Darstellung der Eigenschaften dieser Probleme. Den Einstiegspunkt hierzu bilden die dynamischen Systeme, wobei das Standardmodell eines dynamischen Systems eine endliche Anzahl von gewöhnlichen Differentialgleichungen erster Ordnung

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

ist mit

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{pmatrix}, \quad u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix}, \quad f(t, x, u) = \begin{pmatrix} f_1(t, x, u) \\ f_2(t, x, u) \\ \vdots \\ \vdots \\ f_n(t, x, u) \end{pmatrix}. \quad (2.11)$$

Hierbei bezeichnet man im Allgemeinen t als die Zeit, u_1, \dots, u_m als Eingabevariable und x_1, \dots, x_n als Zustand. Ein Sonderfall dieser Gleichung ist die sogenannte Zustandsgleichung, bei der die Differentialgleichung nicht explizit von u abhängt und die auch ungestörte gewöhnliche Differentialgleichung genannt wird.

Definition 2.2 (Gewöhnliche Differentialgleichung).

Eine gewöhnliche Differentialgleichung im \mathbb{R}^n , $n \in \mathbb{N}$, ist gegeben durch die Gleichung

$$\frac{d}{dt}x(t) = f(t, x(t)), \quad (2.12)$$

wobei $f : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ eine stetige Funktion ist und Vektorfeld genannt wird. Im Weiteren wird in Kurzform oft $\dot{x}(t)$ statt $\frac{d}{dt}x(t)$ verwendet.

Bemerkung 2.3. Verwendet man eine Gleichung dieser Art, so bedeutet dies aber nicht notwendigerweise, dass $u = 0$ gilt. So kann zum Beispiel $u = u(t)$ als Funktion der Zeit, als Zustandsfeedback $u = u(x)$ oder als Mischung aus beidem vorgegeben sein, $u = u(t, x)$, und in (2.10) substituiert werden, so dass eine Gleichung der Form (2.12) vorliegt.

Einen weiteren Sonderfall von (2.12) stellt die sogenannte autonome oder zeitinvariante gewöhnliche Differentialgleichung

$$\dot{x}(t) = f(x(t)) \quad (2.13)$$

dar, da sich die rechte Seite dieser Differentialgleichung durch den Tausch der Zeitvariablen von t zu $\tau = t - a$ nicht verändert.

Um eine eindeutige Lösung zu erhalten ist noch eine weitere Bedingung, die sogenannte Anfangswertbedingung, festzulegen. Dazu wird zu einem $t_0 \in \mathbb{R}$ ein $x \in \mathbb{R}^n$ vorgegeben und die Lösungsfunktion $x(t)$ gesucht, die (2.12) und zudem $x(t_0) = x$ erfüllt. Im Weiteren wird die Lösung dieses Problems mit $x(t; t_0, x)$, oder im Fall $t_0 = 0$ mit $x(t, x)$ bezeichnet. Zum Beweis der Existenz und Eindeutigkeit dieser Lösung (Satz von Picard-Lindelöf) sei auf den folgenden Beweis der Existenz und Eindeutigkeit einer Lösung von Kontrollsystemen verwiesen, da Kontrollsysteme eine Verallgemeinerung dieses Problems darstellen. Alternativ findet sich ein Beweis dazu in Kapitel 10 des Buchs von W.Walter [28].

Eine wichtige Eigenschaft von Differentialgleichungen der Form (2.12) ist die sogenannte Lipschitz-Stetigkeit, die eine Schlüsselbedingung für den Nachweis der Existenz und Eindeutigkeit der Lösung dieses Problems darstellt.

Definition 2.4 (Lokale Lipschitz-Stetigkeit).

Eine Funktion $f : U \rightarrow \mathbb{R}$ heißt Lipschitz-stetig, wenn

$$|f(x) - f(y)| \leq C|x - y| \quad (2.14)$$

für alle $x, y \in U$ und eine Konstante $C \in \mathbb{R}$. Man schreibt

$$\text{Lip}[f] := \sup_{x, y \in U, x \neq y} \frac{|f(x) - f(y)|}{|x - y|}. \quad (2.15)$$

Von besonderem Interessen im Umgang mit Differentialgleichungen ist das Langzeitverhalten dynamischer Systeme, etwa bei technischen Anwendungen oder in der klassischen ökonomischen Theorie. Ziel einer solchen Untersuchung ist zumeist das Auffinden von Ruhelagen.

Definition 2.5 (Ruhelage).

Als Ruhelage wird ein Punkt $x^* \in \mathbb{R}^n$ bezeichnet, falls für die Lösung einer gewöhnlichen Differentialgleichung (2.12) gilt:

$$x(t, x^*) = x^* \quad \forall t \in \mathbb{R}. \quad (2.16)$$

Daraus folgt insbesondere, dass für $f(x^*) = 0$ eine Ruhelage vorliegt.

Bemerkung 2.6. Ohne Beschränkung der Allgemeinheit kann der Ursprung als Ruhelage eines Systems betrachtet werden, da x^* verschoben werden kann mittels $y = x - x^*$. Somit folgt aus (2.12) das neue Differentialgleichungssystem

$$\dot{y} = \dot{x} = f(x) = f(y + x^*) =: g(y), \quad \text{wobei } g(0) = 0. \quad (2.17)$$

Über das Verhalten der Lösung außerhalb von Ruhelagen sagt dies allerdings noch nichts aus, deshalb wurde der Begriff der Stabilität in der Systemtheorie definiert und spielt eine zentrale Rolle, insbesondere die Stabilität von Ruhelagen, auf die im weiteren Verlauf eingegangen werden soll. Zudem kann der Begriff der Stabilität in der Literatur auch im Zusammenhang mit periodischen Orbits und Input-Output-Stabilität gefunden werden. Da dies hier nicht betrachtet werden soll sei dazu lediglich auf die Kapitel 5 und 8 im Buch von H.K.Khalil [13] verwiesen.

Definition 2.7 (Stabilität und asymptotische Stabilität).

Sei $x^* \in \mathbb{R}^n$ eine Ruhelage einer gewöhnlichen Differentialgleichung (2.12).

- (i) Die Ruhelage x^* heißt stabil, falls eine Umgebung $N \subset \mathbb{R}^n$ von x^* und eine Funktion $\alpha \in \mathcal{K}$ existieren mit

$$\|x(t, x) - x^*\| \leq \alpha(\|x - x^*\|) \quad \forall t \geq 0 \quad (2.18)$$

für alle Anfangswerte $x \in N$.

(ii) Als (lokal) asymptotisch stabil bezeichnet man die Ruhelage x^* , falls eine Umgebung $N \subset \mathbb{R}^n$ von x^* und eine Funktion $\beta \in \mathcal{KL}$ existieren, so dass

$$\|x(t, x) - x^*\| \leq \beta(\|x - x^*\|, t) \quad \forall x \in N, t \geq 0. \quad (2.19)$$

(iii) Die Ruhelage heißt global asymptotisch stabil, falls (ii) mit $N = \mathbb{R}^n$ erfüllt ist.

(iv) Lokale (bzw. globale) exponentielle Stabilität liegt bei einer Ruhelage x^* vor, falls Konstanten $C, \sigma \in \mathbb{R}$ mit $C, \sigma > 0$ existieren, so dass (ii) (bzw. (iii)) mit

$$\beta(r, t) \leq Ce^{-\sigma t} r \quad (2.20)$$

gilt.

Weiter heißt eine Ruhelage $x^* \in \mathbb{R}^n$ instabil, wenn sie nicht stabil ist. Der Begriff der Stabilität, wie er hier dargestellt wird, wird auch Lyapunov-Stabilität genannt, wobei Stabilität und Instabilität durch hinreichende Bedingungen an die Linearisierung überprüft werden können. Dazu berechnet man die Eigenwerte λ_i der Matrix

$$A := \frac{d}{dx} f(x^*) \in \mathbb{R}^{n \times n} \quad (2.21)$$

und kann für

$$\sigma := \max(\operatorname{Re}(\lambda_i)), \quad (2.22)$$

wobei Re den Realteil bezeichnet, aus $\sigma < 0$ auf Stabilität und aus $\sigma > 0$ auf Instabilität der Ruhelage schließen.

Bemerkung 2.8. Bei der Definition ist darauf zu achten, dass implizit die Existenz der Lösung $x(t, x)$ der Differentialgleichung (2.12) für alle $t \geq 0$ vorausgesetzt wird. Die hier verwendete Definition von Stabilität und asymptotischer Stabilität ist im eigentlichen Sinn lediglich eine Folgerung aus der standardmäßigen $\varepsilon - \delta$ -Definition dieser Begriffe für Differentialgleichungen der Form (2.12).

Zum Nachweis dessen sei auf den Beweis zu Lemma 4.5 im Buch von H.K.Khalil [13] verwiesen. In der Definition zu Stabilität und asymptotischer Stabilität wurde bereits zwischen lokaler und globaler Stabilität unterschieden. Im lokalen Fall ist man dabei oft daran interessiert, wie weit entfernt von der Ruhelage x^* die Trajektorie verlaufen und insbesondere starten kann und sie trotzdem für $t \rightarrow \infty$ zur Ruhelage konvergiert. Diese Menge wird mit Basin bezeichnet und ist definiert durch:

$$\operatorname{Basin}(x^*) := \left\{ x \mid x(t, x) \in \mathbb{R}^n \text{ ist definiert } \forall t \geq 0, \lim_{t \rightarrow \infty} x(t, x) = x^* \right\} \quad (2.23)$$

Dabei ist es oftmals schwierig bis unmöglich das exakte Basin analytisch zu bestimmen.

2.2 Definition eines Kontrollsystems

Ein Kontrollsystem wird dabei wie folgt festgelegt:

Definition 2.9 (Zustand, Steuerung, Dynamik).

Seien $m, n \in \mathbb{N}$ natürliche Zahlen und $U \subset \mathbb{R}^m$. Weiter sei $f : \mathbb{R}^n \times U \rightarrow \mathbb{R}^n$ eine stetige und bezüglich $x(t) \in \mathbb{R}^n$ und $u(t) \in U$ stetig partiell differenzierbare Funktion. Das System von n Differentialgleichungen 1. Ordnung

$$\dot{x}(t) := \frac{d}{dt}x(t) = f(x(t), u(t)), \quad t \in \mathbb{R} \quad (2.24)$$

heißt Kontrollsystem in kontinuierlicher Zeit $\mathbb{T} = \mathbb{R}$ (oder auch Dynamik des Systems in kontinuierlicher Zeit).

Ein Kontrollsystem in diskreter Zeit $\mathbb{T} = h\mathbb{Z} = \{hk | k \in \mathbb{Z}\}$ im \mathbb{R}^n , $n \in \mathbb{N}$ für ein $h \in \mathbb{R}$, $h > 0$ ist gegeben durch die Differenzengleichung

$$x(t+h) = f_h(x(t), u(t)), \quad (2.25)$$

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

$x(t) = (x_1(t), \dots, x_n(t))^T \in \mathbb{R}^n$ wird Zustandsvektor zum Zeitpunkt $t \in \mathbb{R}$ genannt.

Durch den Steuervektor

$$u(t) = (u_1(t), \dots, u_m(t))^T \in U \subset \mathbb{R}^m \quad (2.26)$$

kann das System zum Zeitpunkt t beeinflusst werden, wobei $U \subset \mathbb{R}^m$ Kontrollwertebereich oder Steuerbereich genannt wird. Im Weiteren soll U als kompakt angenommen werden.

\mathcal{U} und \mathcal{U}_h bezeichnen den Raum der zulässigen Kontrollfunktionen in kontinuierlicher beziehungsweise diskreter Zeit,

$$\mathcal{U} := \{u : \mathbb{R} \rightarrow U | u \text{ zulässig}\} \quad \text{und} \quad \mathcal{U}_h := \{u_h : h\mathbb{Z} \rightarrow U | u_h \text{ zulässig}\} \quad (2.27)$$

der mit der Norm

$$\|u\|_\infty := \inf \{C \geq 0 | \|u(t)\| \leq C \text{ für alle } t \in \mathbb{R}\} \quad (2.28)$$

versehen ist.

Ist t aus einem Intervall $[t_0, t_f]$, $t_f > t_0$ zu wählen, so wird $t_0 \in \mathbb{R}$ mit Anfangs- und $t_f \in \mathbb{R}$ mit Endzeitpunkt bezeichnet. Ebenso werden $x(t_0), x(t_f) \in \mathbb{R}^n$ Anfangs- und Endzustand genannt.

Bemerkung 2.10. Da es sich um ein autonomes Problem handelt kann für den Anfangszeitpunkt ohne Einschränkung der Wert $t_0 = 0$ gewählt werden, wohingegen der Endzeitpunkt t_f entweder fest vorgeschrieben oder frei wählbar sein kann.

Eine allgemeinere Form der Anfangswertbedingung ist die sogenannte Randbedingung, die an dieser Stelle schon im Vorgriff auf die Behandlung des Kontrollsystems im Rahmen eines Optimalsteuerungsproblems eingeführt werden soll:

Definition 2.11 (Randbedingungen).

Sei $r, n \in \mathbb{N}$, $0 \leq r \leq 2n$. Die Bedingung

$$\psi(x(t_0), x(t_f)) = 0 \tag{2.29}$$

für Anfangs- und Endzustand mit einer bezüglich $x(t_0)$ und $x(t_f)$ stetig differenzierbaren Funktion $\psi : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^r$ wird allgemeine Randbedingung genannt.

Als Lösungspaar eines Kontrollsystems (2.24) bezeichnet man dabei das Funktionenpaar $(x(t), u(t))$, wenn es die Gleichung

$$\dot{x}(t) := \frac{d}{dt}x(t) = f(x(t), u(t)) \tag{2.30}$$

für alle Stetigkeitsstellen $t \in [t_0, t_f]$ erfüllt. Gilt für ein solches Lösungspaar $(x(t), u(t))$ von (2.24) die Randbedingungen (2.29), so wird es auch zulässig zur Endzeit t_f genannt.

2.3 Existenz und Eindeutigkeit

Die Existenz und Eindeutigkeit einer Lösung ist im zeitdiskreten Fall leicht durch Induktion zeigbar. Hierzu sei

$$\mathcal{U}_h := \{u_h : h\mathbb{Z} \rightarrow U\} \tag{2.31}$$

die Menge aller möglichen Funktionen u_h . Die Lösung von (2.25) ist somit eine Funktion $x_h : h\mathbb{N}_0 \times \mathbb{R}^n \times \mathcal{U}_h \rightarrow \mathbb{R}^n$ für jeden Anfangswert $x \in \mathbb{R}^n$ und jede Funktion $u_h \in \mathcal{U}_h$, für die gilt:

$$x_h(0, x, u_h) = x \text{ und } x_h(t+h, x, u_h) = f_h(x_h(t, x, u_h), u_h(t)) \tag{2.32}$$

Bei der Wahl des Kontrollfunktionenraumes \mathcal{U} ist vor allem zu beachten, dass zum einen die Existenz und Eindeutigkeit für die Lösungen von (2.24) gegeben ist, zum anderen aber auch die Menge der zulässigen Kontrollfunktionen nicht zu klein ist, wie zum Beispiel die Einschränkung auf die Menge der stetigen Funktionen für viele Anwendungen zu eng ist. Bekanntestes Beispiel für eine unstetige Steuerung ist hierbei die sogenannte bang-bang Steuerung, die häufig bei beschränkten optimalen Steuerprozessen vorkommt. Es ist aber möglich, für die Klasse der messbaren Funktionen die Existenz und Eindeutigkeit von Kontrollsystemen zu zeigen. Hierzu sei an die Definition messbarer Funktionen erinnert:

Definition 2.12 (Messbarkeit).

Sei $I = [a, b] \in \mathbb{R}$ ein abgeschlossenes Intervall.

- (i) Eine Funktion $g : I \rightarrow \mathbb{R}^m$ heißt *stückweise konstant*, falls eine Zerlegung von I in endlich viele Teilintervalle I_j , $j = 1, \dots, n$ existiert, so dass g auf I_j konstant ist für alle $j = 1, \dots, n$.
- (ii) Eine Funktion $g : I \rightarrow \mathbb{R}^m$ wird (Lebesque-)messbar genannt, falls eine Folge von stückweise konstanten Funktionen $g_i : I \rightarrow \mathbb{R}^m$, $i \in \mathbb{N}$, existiert mit $\lim_{i \rightarrow \infty} g_i(x) = g(x)$ für fast alle $x \in I$.
- (iii) Eine Funktion $g : \mathbb{R} \rightarrow \mathbb{R}^m$ heißt (Lebesque-)messbar, falls für jedes abgeschlossene Teilintervall $I = [a, b] \subset \mathbb{R}$ die Einschränkung $g|_I$ messbar im Sinne von (ii) ist.
- (iv) Als *lokal essentiell beschränkt* bezeichnet man eine messbare Funktion $g : \mathbb{R} \rightarrow \mathbb{R}^m$, falls für jedes kompakte Intervall $I \subset \mathbb{R}$ eine Konstante $C \in \mathbb{R}$, $C > 0$ existiert, so dass $\|u(t)\| \leq C$ ist für fast alle $x \in I$.

Satz 2.13 (Satz von Caratheodory).

Gegeben sei ein Kontrollsystem (2.24) mit folgenden Eigenschaften:

- (1) Der Raum der Kontrollfunktionen ist gegeben durch

$$\mathcal{U} := \{u : \mathbb{R} \rightarrow U \mid u \text{ ist messbar und lokal essentiell beschränkt}\}. \quad (2.33)$$

- (2) Das Vektorfeld $f : \mathbb{R}^n \times U \rightarrow \mathbb{R}^n$ ist stetig.

- (3) Für jedes $R \in \mathbb{R}$, $R > 0$ existiert eine Konstante $M_R \in \mathbb{R}$, $M_R > 0$, so dass die Abschätzung

$$\|f(x, u)\| \leq M_R \quad (2.34)$$

für alle $x \in \mathbb{R}^n$ und alle $u \in U$ mit $\|x\| \leq R$ und $\|u\| \leq R$ erfüllt ist.

- (4) Für jedes $R \in \mathbb{R}$, $R > 0$ existiert eine Konstante $L_R \in \mathbb{R}$, $L_R > 0$, so dass die Abschätzung

$$\|f(x_1, u) - f(x_2, u)\| \leq L_R \|x_1 - x_2\| \quad (2.35)$$

für alle $x_1, x_2 \in \mathbb{R}^n$ und alle $u \in U$ mit $\|x_1\| \leq R$, $\|x_2\| \leq R$ und $\|u\| \leq R$ erfüllt ist.

Dann existiert für jeden Punkt $x \in \mathbb{R}^n$ und jede Kontrollfunktion $u \in \mathcal{U}$ ein maximales Intervall $J = (\tau_{\min}, \tau_{\max}) \subset \mathbb{R}$ mit $\tau_{\max} > 0 > \tau_{\min}$, auf dem eine eindeutige und absolut stetige Funktion $x(t)$ existiert, die die Integralgleichung

$$x(t) = x + \int_0^t f(x(\tau), u(\tau)) d\tau \quad (2.36)$$

für alle $t \in J$ erfüllt.

Als Lösung der Gleichung (2.24) wird in diesem Sinne die eindeutige Funktion $x(t)$ zum Anfangswert $x \in \mathbb{R}^n$ und Kontrollfunktion $u \in \mathcal{U}$ gewählt und im Folgenden mit $x(t, x, u)$ bezeichnet.

Bemerkung 2.14. Diese Bezeichnung lässt sich mit der Beobachtung rechtfertigen, dass $x(t, x, u) \in \mathbb{R}^n$ die Differentialgleichung (2.24) für fast alle $t \in I$ erfüllt, da $x(t, x, u)$ absolut stetig und somit für fast alle $t \in I$ nach t differenzierbar ist.

Weiter bezeichnet man für eine Kontrollfunktion $u \in \mathcal{U}$ und $t_1, t_2, t \in \mathbb{R}$ mit $u(t_1 + \cdot)$ die um t_1 verschobene Kontrollfunktion. Diese hat für $t_1, t_2 > 0$ folgende Eigenschaften:

- (i) Ist $x_1 : [0, t_1] \rightarrow \mathbb{R}^n$ eine Lösung von (2.36) zur Kontrollfunktion u und $x_2 : [0, t_2] \rightarrow \mathbb{R}^n$ eine Lösung zur Kontrollfunktion $u(t_1 + \cdot)$ mit $x_2(0) = x_1(t_1)$, so ist die Funktion $x : [0, t_1 + t_2] \rightarrow \mathbb{R}^n$ mit

$$x(t) := \begin{cases} x_1(t) & , t \in [0, t_1], \\ x_2(t - t_1) & , t \in [t_1, t_1 + t_2], \end{cases} \quad (2.37)$$

eine Lösung von (2.36).

- (ii) Ist $x_1 : [0, t_1 + t_2] \rightarrow \mathbb{R}^n$ eine Lösung von (2.36) zur Kontrollfunktion u , so ist die Funktion $x_2 : [0, t_2] \rightarrow \mathbb{R}^n$, gegeben durch

$$x_2(t) := x_1(t + t_1), \quad (2.38)$$

eine Lösung von (2.36) mit $x_2(0) = x_1(t_1)$ zur Kontrollfunktion $u(t_1 + \cdot)$.

Beweis des Satzes von Caratheodory.

Gezeigt wird der Satz zur Existenz und Eindeutigkeit für Intervalle der Form $J = [0, \tau_{max})$. Für $t < 0$ erhält man mit der gleichen Argumentation die Aussagen, wenn $x(t), f(x(t), u(t))$ durch $x(-t), -f(x(-t), u(-t))$ ersetzt und mit den vorab genannten Eigenschaften die Lösungen zusammengesetzt werden.

Existenz auf $[0, \delta]$:

Wähle $x \in \mathbb{R}^n$ und ein $u \in \mathcal{U}$. Zunächst wird gezeigt, dass ein $\delta \in \mathbb{R}, \delta > 0$ existiert, so dass auf dem Intervall $[0, \delta] \subset \mathbb{R}$ eine eindeutige Lösung $x(t) \in \mathbb{R}^n$ existiert. Dazu wird $R \in \mathbb{R}, R > 0$ so groß gewählt, dass $\|u(t)\| \leq R$ für fast alle $t \in [0, 1]$ und $\|x\| \leq \frac{R}{2}$. Nun wählt man $\delta \in (0, 1]$ so, dass $\delta L_R =: \lambda < 1, \lambda \in \mathbb{R}$, und $\delta M_R \leq \frac{R}{2}$ für die Konstanten aus (3) und (4) gilt.

Für eine stetige Funktion $\xi : [0, \delta] \rightarrow \mathbb{R}^n$ definiert man $\|\xi\|_\infty := \sup_{t \in [0, \delta]} \|\xi(t)\|$. Sei $C([0, \delta], R)$ der Raum der stetigen Funktionen $\xi : [0, \delta] \rightarrow \mathbb{R}^n$ mit $\|\xi\|_\infty \leq R$.

Gezeigt wird:

$$S(\xi)(t) := x + \int_0^t f(\xi(s), u(s)) ds \quad (2.39)$$

bildet die Menge $C([0, \delta], R)$ auf sich selbst ab.

- $S(\xi)$ ist stetig, falls ξ stetig ist.
- Aus Annahme (4) und der Wahl von R folgt, dass $\|f(\xi(t), u(t))\| \leq M_R$ ist für fast alle $t \in [0, \delta]$. Daraus folgt für $t \in [0, \delta]$ die Abschätzung

$$\left\| x + \int_0^t f(\xi(s), u(s)) ds \right\| \leq \|x\| + \int_0^t \|M_R\| dt \leq \|x\| + \delta M_R \leq \frac{R}{2} + \frac{R}{2} = R, \quad (2.40)$$

also $\|S(\xi)\|_\infty \leq R$.

- Für zwei Funktionen $\xi_1, \xi_2 \in C([0, \delta], R)$ folgt außerdem aus der Wahl für R die Abschätzung

$$\|f(\xi_1(t), u(t)) - f(\xi_2(t), u(t))\| \leq L_R \|\xi_1(t) - \xi_2(t)\| \quad (2.41)$$

für fast alle $t \in [0, \delta]$ gilt. Es folgt

$$\begin{aligned} \|S(\xi_1)(t) - S(\xi_2)(t)\| &= \left\| \int_0^t f(\xi_1(s), u(s)) ds - \int_0^t f(\xi_2(s), u(s)) ds \right\| \\ &\leq \int_0^t L_R \|\xi_1(s) - \xi_2(s)\| ds \leq \delta L_R \|\xi_1 - \xi_2\|_\infty = \lambda \|\xi_1 - \xi_2\|_\infty. \end{aligned} \quad (2.42)$$

Da $\lambda < 1$ gilt ist S eine Kontraktion auf $C([0, \delta], R)$, und da der Raum $C([0, \delta], R)$ mit der Norm $\|\cdot\|_\infty$ ein Banach-Raum ist, so folgt nach dem Banachschen Fixpunktsatz die Existenz eines eindeutigen Fixpunktes $x \in C([0, \delta], R)$ mit $S(x) = x$. Dies entspricht gerade der gesuchten eindeutigen Lösung von (2.36).

Fortsetzbarkeit auf $J = [0, \tau_{max})$:

Sei

$$\tau_{max} := \sup \{t > 0 \mid \text{es existiert eine Lösung von (2.36) auf } [0, t]\}. \quad (2.43)$$

Offenbar ist $J = [0, \tau_{max})$ das gesuchte Intervall. Wegen des ersten Teils des Beweises und dem vorher genannten Eigenschaften muss dieses Intervall rechts offen sein, da es ansonsten auf $[\tau_{max}, \tau_{max} + \delta]$ fortgesetzt werden könnte und somit der angenommenen Maximalität widerspräche.

Eindeutigkeit auf $J = [0, \tau_{max})$:

Seien $x_1, x_2 \in \mathbb{R}^n$ zwei verschiedene Lösungen mit $x_1(0) = x_2(0) = x$. Somit existiert ein $t \in J$ mit $x_1(t) \neq x_2(t)$. Sei $t^* := \inf \{t \in J \mid x_1(t) \neq x_2(t)\}$, dann gilt $x_1(t^*) = x_2(t^*) =: x^*$. Gemäß den aufgezählten Eigenschaften sind die Funktionen $x_1(t^* + \cdot)$ und $x_2(t^* + \cdot)$ Lösungen von (2.36) mit Anfangswert $x = x^*$ und Kontrollfunktion $u(t^* + \cdot)$. Aus dem ersten Teil des Beweises folgt, dass ein $\delta^* > 0$ existiert, so dass diese Integralgleichung eine eindeutige Lösung auf $[0, \delta^*]$ hat. Also müssen x_1 und x_2 auf $[t^*, t^* + \delta^*]$ übereinstimmen und es ergibt sich ein Widerspruch zur Definition von t^* . \square

Bemerkung 2.15. *Bei der Verwendung unstetiger Steuerungen muss die Robustheit einer Lösung gegen Störungen beachtet werden. Für Ergebnisse zu dieser Frage sei auf Kapitel 6 der Arbeit von C.M.Kellet [12] verwiesen.*

2.4 Asymptotische Kontrollierbarkeit

Es soll nun das Konzept der asymptotischen Stabilität für Differentialgleichungen der Form (2.12) auf nichtlineare Kontrollsysteme der Form (2.24) verallgemeinert werden. Analog zur Idee der gleichmäßigen Konvergenz gegen 0, wie die asymptotische Stabilität sie verlangt, soll dies für das Kontrollsystem ebenso gelten für eine passende Wahl von $u \in \mathcal{U}$. Dabei wird aber nicht verlangt, dass $f(x^*, u) = 0$ für alle $u \in \mathcal{U}$ gelten muss, sondern ein Gleichgewicht x^* des Kontrollsystems (2.24) liegt dann vor, wenn

$$f(x^*, 0) = 0 \quad (2.44)$$

gilt. Hierbei kann ohne Beschränkung der Allgemeinheit $0 \in U$ angenommen werden. Die asymptotische Kontrollierbarkeit definiert sich dann wie folgt:

Definition 2.16 (Asymptotische Kontrollierbarkeit).

Sei $x^* \in \mathbb{R}^n$ ein Gleichgewicht des Kontrollsystems (2.24).

- (i) x^* heißt lokal asymptotisch kontrollierbar, falls eine Umgebung N von x^* , eine Funktion $\delta \in \mathcal{N}$ und eine Funktion $\beta \in \mathcal{KL}$ existieren, so dass für alle $x \in N$ eine Funktion $u_x \in \mathcal{U}$ mit $\|u_x\|_\infty \leq \delta(\|x - x^*\|)$ existiert, für die gilt:

$$\|x(t, x, u_x) - x^*\| \leq \beta(\|x - x^*\|, t) \quad \forall t \geq 0. \quad (2.45)$$

- (ii) x^* heißt global asymptotisch kontrollierbar, falls (i) mit $N = \mathbb{R}^n$ gilt.

Der Begriff der asymptotischen Stabilität wird für Kontrollsysteme noch weiter differenziert. Dabei unterscheidet man zwischen starker und schwacher asymptotischer Stabilität:

- (i) Ein Punkt $x^* \in \mathbb{R}^n$ heißt starkes (oder robustes) Gleichgewicht, falls für alle $u \in \mathcal{U}$ und alle $t \geq 0$ gilt:

$$x(t, x^*, u) = x^* \quad (2.46)$$

- (ii) Ein starkes Gleichgewicht x^* heißt stark asymptotisch stabil (oder robust asymptotisch stabil), falls eine offene Umgebung $N(x^*)$ und eine \mathcal{KL} -Funktion β existieren, so dass für alle $x \in N(x^*)$ die Ungleichung

$$\|x(t, x, u) - x^*\| \leq \beta(\|x - x^*\|, t) \quad (2.47)$$

für alle $u \in \mathcal{U}$ und alle $t \geq 0$ gilt.

- (iii) Ein Punkt $x^* \in \mathbb{R}^n$ heißt schwaches (oder kontrolliertes) Gleichgewicht, falls ein $u \in \mathcal{U}$ existiert, so dass

$$x(t, x^*, u) = x^* \quad (2.48)$$

für alle $t \geq 0$ gilt.

- (iv) Ein schwaches Gleichgewicht x^* heißt schwach asymptotisch stabil (oder asymptotisch kontrollierbar), falls eine offene Umgebung $N(x^*)$ und eine \mathcal{KL} Funktion β existieren, so dass für jedes $x \in N(x^*)$ ein $u_x \in \mathcal{U}$ existiert, so dass die Ungleichung

$$\|x(t, x, u_x) - x^*\| \leq \beta(\|x - x^*\|, t) \quad (2.49)$$

für alle $t \geq 0$ gilt.

Bemerkung 2.17. *Diese Definition geht auf die anfangs erwähnten unterschiedlichen Sichtweisen des Problems (2.24) zurück. Im Fall einer starken Stabilität etwa wirken sich Störungen innerhalb einer Menge \mathcal{U} nicht negativ auf die asymptotisch stabile Eigenschaft eines Gleichgewichts x^* aus. Besonders interessant ist diese Eigenschaft bei möglichem Auftreten von Diskretisierungsfehler, etwa in der numerischen Berechnung, oder auch bei Fehlern im Modell.*

Bemerkung 2.18. *Die schwache Stabilität dagegen geht bei der Auswahl eines $u \in \mathcal{U}$ davon aus, dass ein bestimmtes Verhalten des Systems (2.24) erzwungen werden soll. Dies schließt nicht nur mögliche andere existente Lösungen aus, sondern verfolgt bereits ein bestimmtes Ziel. Weiterführend soll das Erreichen dieses Zieles durch ein Optimalitätskriterium bewertet werden.*

Der Ansatz zur Kontrollierbarkeit ist somit eine natürliche Erweiterung der asymptotischen Stabilität, bringt jedoch verschiedene Probleme mit sich.

- Die Steuerung u_x muss bereits zum Anfangszeitpunkt für alle zukünftigen Zeitpunkte $t > 0$ berechnet worden sein.
- Da nicht immer nur von einem Anfangswert x ausgegangen werden kann, ist die Steuerung u_x für eine Vielzahl dieser Anfangswerte im Voraus zu berechnen.
- Abweichungen oder Störungen sind nicht immer im mathematischen Modell des echten Systems miteinbezogen, so dass dies für das berechnete u_x und wachsende t zu immer ungenaueren Ergebnissen führen kann.

2.5 Feedback-Stabilisierbarkeit

Eine Möglichkeit zur Lösung dieser Probleme stellt die Implementierung der Steuerung als Feedback dar. Das Konzept des Zustandsfeedbacks, das zur Feedback-Stabilisierbarkeit führen soll, setzt dabei daran an, dass die Steuerung $u \in \mathcal{U}$ nicht mehr im Voraus zu berechnen ist, sondern abhängig von vorliegendem Zustand $x \in \mathbb{R}^n$ bestimmt werden soll. Daraus ergibt sich die folgende Definition.

Definition 2.19 (Zustandsfeedback).

Gegeben sei das Kontrollsystem (2.24). Eine Abbildung $F : \mathbb{R}^n \rightarrow U$ heißt Zustandsfeedback oder Rückkopplung. Die zugehörige Differentialgleichung

$$\dot{x}(t) = f_F(x(t)) \quad , \quad f_F(x) = f(x, F(x)) \quad (2.50)$$

bezeichnet man als Feedback-geregeltes oder rückgekoppeltes oder geregeltes System. Für die Lösungen von (2.50) wird im Weiteren die Notation $x(t, x, F)$ benutzt.

Eine derartige Steuerung, die *on-line*, das heißt zur Rechenzeit, berechenbar ist, wird closed-loop Steuerung oder auch Regelung genannt.

Definition 2.20 (Feedback-Stabilisierbarkeit).

Sei $x^* \in \mathbb{R}^n$ ein Gleichgewicht des Kontrollsystems (2.24). Existiert eine stetige Funktion $F : \mathbb{R}^n \rightarrow U$, so dass das geregelte System (2.50) lokal asymptotisch stabil ist, dann heißt x^* lokal stabilisierbar mit stetigem Feedback (oder auch feedback-stabilisierbar).

Hier wurde das Feedback auf die Klasse der stetigen Funktionen eingeschränkt, um die Existenz einer Lösung garantieren zu können. Da f_F nicht notwendigerweise Lipschitz-stetig sein muss ist die Eindeutigkeit einer Lösung von (2.50) nicht beweisbar, so dass zu einem Anfangswert verschiedene Lösungen existieren können.

Zwischen der Kontrollierbarkeit und der Feedback-Stabilisierbarkeit ist aber der im nächsten Satz folgende Zusammenhang nachweisbar.

Satz 2.21 (Feedback-Stabilisierbarkeit \Rightarrow Asymptotische Kontrollierbarkeit).

Sei $x^* = 0$, $x^* \in \mathbb{R}^n$ Ruhelage eines Kontrollsystems der Form (2.24). Wenn die Ruhelage lokal beziehungsweise global feedback-stabilisierbar ist, so ist sie auch lokal beziehungsweise global asymptotisch kontrollierbar.

Beweis. Aus der Feedback-Stabilisierbarkeit folgt für das geregelte System f_F mit geeignetem Feedback F , dass eine Funktion $\beta \in \mathcal{KL}$ existiert mit:

$$\|x(t, x, F)\| \leq \beta(\|x\|, t) \quad \forall x \in N \subseteq \mathbb{R}^n, t \geq 0 \quad (2.51)$$

Setzt man $\delta(r) := \max\{\|F(x)\| \mid \|x\| \leq \beta(r, 0)\}$, so folgt aus der Stetigkeit von F , dass $F \in \mathcal{N}$ ist. Für $x \in N$ gilt

$$u_x(t) = F(x(t, x, F)) \quad (2.52)$$

und u_x stellt eine der Lösungen von (2.50) dar. Wegen der Stetigkeit von F und $x(t, x, F)$ ist u_x zudem stetig und messbar. Aus

$$\begin{aligned} x(t, x, F) &= x + \int_0^t f(x(\tau, x, F)) d\tau \\ &= x + \int_0^t f(x(\tau, x, F), F(x(\tau, x, F))) d\tau \\ &= x + \int_0^t f(x(\tau, x, F), u_x(\tau)) d\tau \end{aligned} \quad (2.53)$$

folgt mit der Lösung

$$x(t, x, u_x) = x + \int_0^t f(x(\tau, x, u_x), u_x(\tau)) d\tau \quad (2.54)$$

von (2.24) und der Existenz und Eindeutigkeit nach Caratheodory die Behauptung

$$x(t, x, u_x) = x(t, x, F) \quad \forall t \geq 0. \quad (2.55)$$

□

Bemerkung 2.22. *Die Bestimmung eines solchen Zustandsfeedbacks wirft allerdings große Probleme auf. Da zudem von stetigen Funktionen ausgegangen wird stellt die Verwendung von Zustandsfeedbacks eine zu starke Einschränkung dar.*

2.6 Lyapunov-Kontroll-Funktionen

Ein weiteres wesentliches Hilfsmittel zur Analyse der Stabilität und Stabilisierbarkeit von Kontrollsystemen ist die Lyapunov-Kontroll-Funktion.

Definition 2.23 (Lyapunov-Kontroll-Funktion).

Gegeben sei ein Kontrollsystem (2.24) mit $f(0,0) = 0$. Als eine lokale Lyapunov-Kontroll-Funktion bezeichnet man eine stetige Funktion $V : \mathbb{R}^n \rightarrow \mathbb{R}$, falls Funktionen $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$, eine stetige Funktion $W : \mathbb{R}^n \rightarrow \mathbb{R}$, eine Funktion $\delta \in \mathcal{N}$, eine Umgebung $N \subset \mathbb{R}^n$ mit $0 \in N$ und eine Konstante $C \in \mathbb{R}$, $C > 0$ existieren, so dass $W(x) > 0$ für alle $x \in \text{cl}N$, $V(x) < C$ für alle $x \in N$, $V(x) = C$ für alle $x \in \partial N$ gilt und die Ungleichungen

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|) \quad (2.56)$$

$$\inf_{u \in \mathcal{U}; \|u\|_\infty \leq \delta(\|x\|)} \left\{ V(x(t, x, u)) + \int_0^t W(x(s, x, u)) ds \right\} \leq V(x) \quad (2.57)$$

für alle $x \in N$ und alle $t > 0$ erfüllt sind. Weiter wird eine Funktion V globale Lyapunov-Kontroll-Funktion genannt, falls V und W die Bedingungen (2.56), (2.57) für $N = \mathbb{R}^n$ und $C = \infty$ erfüllen.

Das Paar (V, W) heißt auch Lyapunov-Kontroll-Paar.

Dabei bezeichnet $\text{cl}N$ den Abschluss der Umgebung N und ∂N den Rand von N . Für gewöhnliche Differentialgleichungen vereinfacht sich dies zu der sogenannten Lyapunov-Funktion:

Definition 2.24 (Lyapunov-Funktion).

Gegeben sei eine gewöhnliche Differentialgleichung (2.12) mit $f(0) = 0$. Eine stetige und auf $\mathbb{R}^n \setminus \{0\}$ stetig differenzierbare Funktion $V : \mathbb{R}^n \rightarrow \mathbb{R}$ heißt lokale Lyapunov-Funktion, falls Funktionen $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$, eine stetige Funktion $W : \mathbb{R}^n \rightarrow \mathbb{R}$, eine Umgebung $N \subset \mathbb{R}^n$ mit

$0 \in N$ und eine Konstante $C \in \mathbb{R}$, $C > 0$ existieren, so dass $W(x) > 0$ für alle $x \in \text{cl}N$, $V(x) < C$ für alle $x \in N$, $V(x) = C$ für alle $x \in \partial N$ gilt und die Ungleichungen

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|) \quad (2.58)$$

$$DV(x)f(x) \leq -W(x) \quad (2.59)$$

für alle $x \in N$ erfüllt sind.

Bemerkung 2.25. Für die Lyapunov-Kontroll-Funktion lässt sich beweisen, dass sich die Existenz dieser Funktion und die asymptotische Kontrollierbarkeit gegenseitig implizieren. Eine entsprechende Beweisskizze ist in Kapitel 5.2 der Arbeit von L.Grüne [8] gegeben.

Unter den Bedingungen, dass

- die Lyapunov-Kontroll-Funktion zweimal stetig differenzierbar ist
- und für diese zudem eine Funktion $\delta \in \mathcal{N}$ mit $\delta(0) = 0$ gewählt werden kann,

ist es zudem möglich explizit ein stetiges Zustandsfeedback zu bestimmen, das für die Differentialgleichung (2.50) global asymptotisch stabil ist. Die entsprechende Formel ist in der Literatur auch als Sontag-Formel bekannt und entsprechend bewiesene Sätze finden sich unter anderem in Kapitel 6.2 der Arbeit von L.Grüne [8].

Wichtigstes Ergebnis ist aber folgendes Theorem, dessen Beweis in H.K.Khalils Buch [13] zu Theorem 4.1 zu finden ist.

Theorem 2.26 (Lyapunov Stabilitätstheorem).

Sei $x^* \in \mathbb{R}^n$, $x^* = 0$ eine Ruhelage für das System (2.12) und sei $N \subset \mathbb{R}^n$ eine Umgebung von x^* . Gilt für eine stetig differenzierbare Funktion $V : N \rightarrow \mathbb{R}$

$$V(x) > 0 \quad \forall x \in N \setminus \{0\} \quad \text{und} \quad V(0) = 0 \quad (2.60)$$

$$\dot{V}(x) \leq 0 \quad \forall x \in N, \quad (2.61)$$

dann ist die Ruhelage x^* stabil. Gilt zudem

$$\dot{V}(x) < 0 \quad \forall x \in N \setminus \{0\}, \quad (2.62)$$

dann ist x^* asymptotisch stabil.

Bemerkung 2.27. Lyapunov-Funktionen stellen ein probates Mittel zum Nachweis der asymptotischen Stabilität dar, da lediglich das Vektorfeld bekannt sein muss, nicht aber die tatsächlichen Lösungen des Systems. Theorem 2.26 stellt allerdings nur hinreichende, nicht aber notwendige Bedingungen zur Verfügung. Dagegen liefern die sogenannten umgekehrten Lyapunov Theoreme notwendige Bedingungen, für die im kontinuierlichen Fall auf Kapitel 4.7 des Buchs von H.K.Khalil [13] und im zeitdiskreten Fall auf Kapitel 7 und 8 der Arbeit von C.M.Kellet [12] verwiesen sei.

2.7 Sampled-data Lösungen

Im Gegensatz zu den bisher betrachteten Implementierungen der Steuerung, die immer von einer stetigen Funktion ausgingen, ist der sampled-data Ansatz ein Konzept für eine unstetige Feedback-Steuerung. Da im Folgenden mit derartigen Steuerungen gearbeitet werden soll werden an dieser Stelle neben einer grundsätzlichen Definition auch die Zusammenhänge zu den zuvor aufgeführten Themen aufgezeigt.

Definition 2.28 (Sampled-data Lösung).

Als sampled-data Folge wird die Folge

$$t = (t_i)_{i \in \mathbb{N}_0}, \quad 0 = t_0 < t_1 < t_2 < \dots, \quad t_i \in \mathbb{R} \quad (2.63)$$

bezeichnet, falls $t_i \rightarrow \infty$ für $i \rightarrow \infty$ und

$$\Delta_t := \sup_{i \in \mathbb{N}_0} (t_{i+1} - t_i) < \infty \quad (2.64)$$

gilt. Die Zahl $\Delta_t \in \mathbb{R}$ wird auch sampled-data Rate genannt.

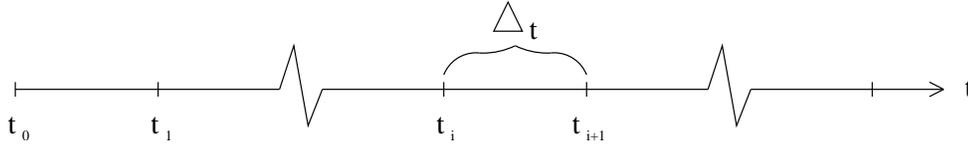


Abbildung 2.2: Sampled-data Folge und sampled-data Rate

Gegeben sei ein Kontrollsystem (2.24) und eine Funktion $F : \mathbb{R}^n \rightarrow U$, die $\|F(x)\| \leq \delta(\|x\|)$ für ein $\delta \in \mathcal{N}$ und alle $x \in \mathbb{R}^n$ erfüllt. Die sampled-data Lösung des Problems

$$\dot{x}(t) = f(x(t), F(x(t))), \quad x(0) = x$$

definiert man zu einer gegebenen sampled-data Folge t für $\bar{t} \geq 0$ induktiv durch

$$x_t(\bar{t}, x, F) = x(\bar{t} - t_i, x_i, F(x_i)) \quad \forall \bar{t} \in [t_i, t_{i+1}], \quad (2.65)$$

wobei $x(\cdot, x_i, F(x_i)) \in \mathbb{R}^n$ die Lösung von (2.24) zum Anfangswert $x_i := x_t(t_i, x, F)$ und konstanter Kontrollfunktion $u(\bar{t}) \equiv F(x_i)$ bezeichnet.

Setzt man hierbei die Bedingungen zum Existenz und Eindeutigkeitsatz von Caratheodory voraus, so existiert wegen der Beschränktheit von F eine eindeutige Lösung x_t für jede sampled-data Folge t . Eine Definition von Stabilisierbarkeit für sampled-data Lösungen lautet dabei wie folgt.

Definition 2.29 (Semiglobale praktisch asymptotische Stabilisierbarkeit).

Gegeben sei ein Kontrollsystem (2.24) mit $f(0,0) = 0$. Eine Familie von Feedbacks $F_\Delta : \mathbb{R}^n \rightarrow U$ für $\Delta \in (0, \Delta^*] \subset \mathbb{R}$ stabilisiert das Gleichgewicht $x^* = 0$ des sampled-data Systems semiglobal praktisch asymptotisch, falls eine Funktion $\beta \in \mathcal{KL}$ existiert, so dass für je zwei Konstanten $R, \varepsilon \in \mathbb{R}$, $R > \varepsilon > 0$, ein $\Delta_0 \in \mathbb{R}$, $\Delta_0 > 0$ existiert, so dass für alle $\Delta \in (0, \Delta_0]$ die sampled-data Lösungen $x_{t_\Delta}(t, x, F_\Delta) \in \mathbb{R}^n$ der Abschätzung

$$\|x_{t_\Delta}(t, x, F_\Delta)\| \leq \max \{\beta(\|x\|, t), \varepsilon\} \quad (2.66)$$

für alle $t \geq 0$ und alle Anfangswerte $x \in \mathbb{R}^n$ mit $\|x\| \leq R$ genügen.

Dabei bezieht sich *praktisch* auf die Konstante ε und der Begriff *semiglobal* auf die Konstante R . Es kann also eine stabilisierende Steuerung gefunden werden, so dass jede beliebige kompakte Menge im Basin eingeschlossen werden kann. Somit ist die sampled-data Rate umso kleiner, die sampled-data Folge damit umso feiner zu wählen, je näher man dem Gleichgewicht $x^* = 0$ kommen will und je weiter man mit dem Anfangswert x von x^* entfernt ist. Kann man $R = \infty$ setzen, so spricht man auch von praktisch asymptotischer Stabilisierbarkeit.

Ebenso wie bei stetigem Feedback kann auch für unstetiges Feedback die Lyapunov-Funktion zur Entwicklung eines stabilisierenden Feedbacks benutzt werden.

Definition 2.30 (Semiglobale praktische Familie von Lyapunov-Funktionen).

Betrachte ein Kontrollsystem (2.24) mit $f(0,0) = 0$. Falls Funktionen $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$, eine stetige Funktion $W : \mathbb{R}^n \rightarrow \mathbb{R}$ und eine Funktion $\delta \in \mathcal{N}$ derart existieren, so dass $W(x) > 0$ gilt für alle $x \in \mathbb{R}^n \setminus \{0\}$, die Ungleichungen

$$\alpha_1(\|x\|) \leq V_\Delta(x) \leq \alpha_2(\|x\|) \quad (2.67)$$

für alle $\Delta \in (0, \Delta^*] \subset \mathbb{R}$ erfüllt sind und für alle Konstanten $C_1, C_2 \in \mathbb{R}$, $C_2 > C_1 > 0$, ein $\Delta_0 \in \mathbb{R}$, $\Delta_0 > 0$ existiert, so dass

$$\inf_{u \in U; \|u\| \leq \delta(\|x\|)} V_\Delta(x(\Delta, x, u)) \leq \max \{V_\Delta(x) - \Delta W(x), C_1\} \quad (2.68)$$

für alle $x \in \mathbb{R}^n$ mit $V_\Delta(x) \leq C_2$, alle $t > 0$ und alle $\Delta \in (0, \Delta_0]$ gilt, dann bezeichnet man eine Familie von stetigen Funktionen $V_\Delta : \mathbb{R}^n \rightarrow \mathbb{R}$ für $\Delta \in (0, \Delta^*]$ als semiglobale praktische Familie von Lyapunov-Funktionen.

Da im Gegensatz zu den Lyapunov-Kontroll-Funktionen hier lediglich ein konstanter Kontrollwert $u \in U$ und keine messbare Kontrollfunktion aus $u \in \mathcal{U}$ zu bestimmen ist, vereinfacht sich das Problem dementsprechend.

Satz 2.31. Für ein Kontrollsystem (2.24) mit $f(0,0) = 0$ sei eine semiglobale praktische Familie von Lyapunov-Funktionen V_Δ für $\Delta \in (0, \Delta^*] \subset \mathbb{R}$ gegeben. Dann ist die Familie von Feedbacks F_Δ , die durch

$$V_\Delta(x(\Delta, x, F_\Delta(x))) = \min_{u \in U; \|u\| \leq \delta(\|x\|)} V_\Delta(x(\Delta, x, u)) \quad (2.69)$$

definiert ist, eine Familie von semiglobal praktisch asymptotisch stabilisierenden Feedbacks im Sinne von Definition 2.29.

Um diesen Satz beweisen zu können wird zunächst das folgende Lemma gezeigt, das eine äquivalente Formulierung der Ungleichung (2.59) darstellt.

Lemma 2.32. *Sei V eine stetige Funktion, die auf $\mathbb{R}^n \setminus \{0\}$ zweimal stetig differenzierbar ist. Weiter erfülle V die Bedingung (2.56) für eine Umgebung $N \subset \mathbb{R}^n$ mit $0 \in N$ und eine Konstante $C \in \mathbb{R}$, $C > 0$ (für $N = \mathbb{R}^n$ ist $C = \infty$). Dann existiert genau dann eine Funktion W , die die Bedingungen von Definition 2.24 erfüllt, wenn es eine lokal Lipschitz-stetige Funktion $g : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$ gibt, so dass $g(r) > 0$ für alle $r \in (0, C]$ (im Fall $N = \mathbb{R}^n$ für alle $r > 0$) und die Ungleichung*

$$DV(x)f(x) \leq -g(V(x)) \quad (2.70)$$

für alle $x \in N$ gilt.

Beweis. Analog zum lokalen Fall kann im Globalen mit $C = \infty$, $r > 0$ und $N = \mathbb{R}^n$ der Beweis geführt werden.

“ \Leftarrow “: Da g gegeben ist erfüllt $W(x) := g(V(x))$ die Bedingungen aus Definition 2.24.

“ \Rightarrow “: W sei gegeben. Für $r \in [0, C]$ sei

$$\tilde{g}(r) := \inf \{W(x) | x \in \mathbb{R}^n, V(x) = r\}. \quad (2.71)$$

Weiter setzt man $\tilde{g}(r) = \tilde{g}(C)$ für alle $r > C$. Somit ist nur noch die Lipschitz-Stetigkeit von \tilde{g} zu zeigen:

Wählt man eine Folge $(t_k)_{k \in \mathbb{Z}}$, $t_k \in \mathbb{R}$ derart, dass gilt:

$$0 < t_{k+1} < t_k \quad \forall k \in \mathbb{Z}, \quad \lim_{k \rightarrow \infty} t_k = 0, \quad \lim_{k \rightarrow -\infty} t_k = \infty \quad (2.72)$$

und setzt

$$g(0) = 0, \quad g_k := \min_{r \in [t_{k+1}, t_k]} g(r) \quad (2.73)$$

für alle $k \in \mathbb{N}_0$, so definiert man

$$g(r) := g_{k+1} + \frac{t - t_{k+1}}{t_k - t_{k+1}}(g_k - g_{k+1}), \quad r \in [t_{k+1}, t_k] \quad (2.74)$$

Somit gilt $0 < g(r) \leq \tilde{g}(r)$ für $r \in (0, C]$ und die Bedingungen an g sind erfüllt, insbesondere folgt die lokale Lipschitz-Stetigkeit, da g eine stückweise lineare Funktion ist. \square

Beweis zu Satz 2.31. Analog zum Beweis des vorangegangenen Lemmas kann man ohne Beschränkungen der Allgemeinheit annehmen, dass $W(x) \geq g(V_\Delta(x))$ für ein geeignetes stetiges $g : \mathbb{R}_0 \rightarrow \mathbb{R}_0$ mit $g(r) > 0$ für $r > 0$, $r \in \mathbb{R}$ ist. Da alle V_Δ durch die gleichen \mathcal{K}_∞ Funktionen α_1, α_2 beschränkt werden, kann g von Δ unabhängig gewählt werden.

Für gegebene $R > \varepsilon > 0$ wird $C_2 = \alpha_2(R)$, $C_1 = \alpha_1^{-1}(\frac{\varepsilon}{2})$ gewählt. Es sei Δ_0 gemäß Definition 2.30 gegeben und $\Delta \in (0, \Delta_0]$ beliebig.

$$\Rightarrow V_\Delta(x) \leq \alpha_2(\|x\|) \leq \alpha_2(R) = C_2 \quad (2.75)$$

Aus (2.68) und der Definition von F_Δ folgt

$$V_\Delta(x(\Delta, x, F_\Delta(x))) \leq \max \{V_\Delta(x) - \Delta g(V_\Delta(x)), C_1\} \quad (2.76)$$

Definiert man eine Funktion $\mu(r, t)$ durch

$$\mu(r, t + \tau) = \mu(r, t) - \tau \inf_{s \in [0, \Delta_0]} g(\mu(r, t + s)) \quad \forall \tau \in (0, \Delta_0], \mu(r, 0) = r, \quad (2.77)$$

so ist diese Funktion streng monoton fallend in t und konvergiert gegen 0.

Annahme: $\gamma := \lim_{t \rightarrow \infty} \mu(r, t) > 0$

Für $\varepsilon_0 > 0$ folgt für jedes $\varepsilon \in (0, \varepsilon_0]$ und alle $t > 0$ mit $\mu(r, t) \leq \gamma + \varepsilon$:

$$\begin{aligned} \mu(r, t + \Delta) &= \mu(r, t) - \Delta \inf_{s \in [0, \Delta_0]} g(\mu(r, t + s)) \\ &\leq \mu(r, t) - \Delta \inf_{r \in [\gamma, \gamma + \varepsilon_0]} g(r) = \mu(r, t) - \Delta \alpha_0 \leq \gamma + \varepsilon - \Delta \alpha_0 \end{aligned} \quad (2.78)$$

Für $\varepsilon < \Delta \alpha_0$ gilt $\mu(r, t + \Delta) < \gamma$. \Rightarrow Widerspruch

Somit gilt $\lim_{t \rightarrow \infty} \mu(r, t) = 0$ und μ ist eine \mathcal{KL} -Funktion. Mittels Induktion folgt für $t = (i\Delta)_{i \in \mathbb{N}_0}$

$$V_\Delta(x_t(t, x, F_\Delta)) \leq \max \{x(V_\Delta(x), t), C_1\} \quad (2.79)$$

für alle $t = i\Delta$ und alle $x \in \mathbb{R}^n$ mit $\|x\| \leq R$.

$$\Rightarrow \|x\| \leq \alpha_1^{-1}(V_\Delta(x)), \quad V_\Delta(x) \leq \alpha_2(\|x\|) \quad (2.80)$$

Zudem gilt für $\beta(r, t) := \alpha_1^{-1}(\mu(\alpha_2(r), t)) \in \mathcal{KL}$ die Ungleichung

$$\|x_t(t, x, F_\Delta)\| \leq \max \left\{ \alpha_1^{-1}(x(\alpha_2(\|x\|), t)), \alpha_1^{-1}(C_1) \right\} = \max \left\{ \beta(\|x\|, t), \frac{\varepsilon}{2} \right\} \quad (2.81)$$

für $t = i\Delta$. Wegen der Stetigkeit von $\|x_t(t, x, F_\Delta)\|$ gilt diese Abschätzung für hinreichend kleines $\Delta > 0$ auch für beliebige t in den Zwischenintervallen $[i\Delta, (i+1)\Delta]$, wenn $\frac{\varepsilon}{2}$ durch ε und β durch $C\beta$ für eine geeignete Konstante $C \in \mathbb{R}$, $C > 0$ ersetzt werden. \square

Bemerkung 2.33. V_Δ muss nicht für alle $\Delta \in (0, \Delta_0]$ berechnet werden, sondern es genügt zu gegebenem $R > \varepsilon > 0$ eine Lyapunov-Funktion V_{Δ_0} mit $\alpha_2(R) \leq C_2$ und $C_1 \leq \alpha_1^{-1}(\frac{\varepsilon}{2})$ zu berechnen. Mittels Satz 2.31 kann dann das stabilisierende Feedback F_{Δ_0} mit Parametern R und ε bestimmt werden ohne V_Δ für $\Delta < \Delta_0$ zu kennen.

Zudem kann auch folgender Satz bewiesen werden, wobei hier auf den Beweis zu Satz 8.4 aus der Arbeit von L.Grüne [8] verwiesen sei.

Satz 2.34. Gegeben sei ein Kontrollsystem (2.24) mit $f(0, 0) = 0$. Wenn das System asymptotisch kontrollierbar ist, so existiert eine Familie von Lyapunov-Funktionen gemäß Definition 2.30 und das sampled-data System ist semiglobal praktisch asymptotisch stabil.

Kapitel 3

Optimale Steuerprozesse

Optimale Steuerprozesse bewerten die Lösungen von Kontrollsystemen anhand eines Optimalitätskriteriums, dem sogenannten Zielfunktional. Die Gestaltung des Zielfunktional ist dabei sinnvoll vom Anwender vorzugeben und kann verschiedenste Ziele verfolgen. Bekannteste Beispiele hierfür sind die zeitoptimale Steuerung, die auf eine kürzest mögliche Endzeit abstellt um ein bestimmtes Ziel zu erreichen, und die energieoptimale Steuerung, bei der mit möglichst geringem Steueraufwand ein Ziel erreicht werden soll. In das Zielfunktional können natürlich auch die Zustandstrajektorien eingehen, etwa um den Verlauf der Lösung des Kontrollsystems zu bewerten.

Das sich hieraus ergebende Ziel der optimalen Steuerung ist demzufolge die Bestimmung einer Kontrollfunktion $u \in \mathcal{U}$ in Abhängigkeit von x , die den Wert des Zielfunktional optimiert.

Zur Lösung optimaler Steuerprozesse stehen zwei verschiedene Ansätze zur Verfügung:

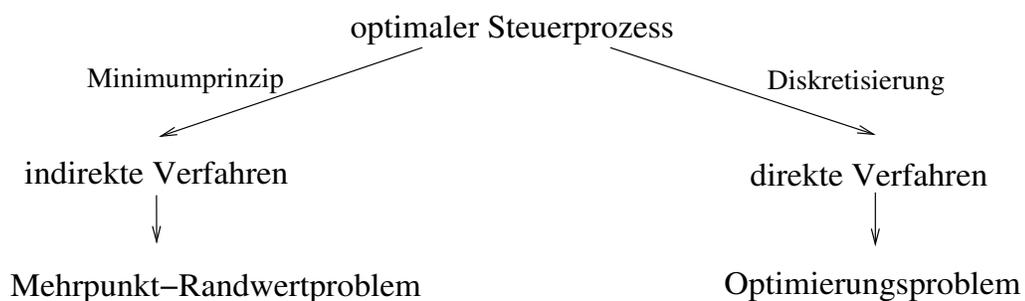


Abbildung 3.1: Ansätze zur Lösung optimaler Steuerprozesse

Aufbau und Inhalt dieses Kapitels orientieren sich weitgehend an den Kapitel 3, 4 und 5 der Arbeit von Ch.Büskens [2]. Dabei wird neben der Definition eines optimalen Steuerprozesses auf verschiedene numerische Lösungsverfahren eingegangen und zusätzlich die Problemstellung des zu untersuchenden modellprädiktiven Regelungsansatzes festgelegt. Letzterem liegt die Arbeit von D.Nešić und L.Grüne [23] zugrunde.

3.1 Definition eines unbeschränkten optimalen Steuerprozesses

Die hier angeführte Definition eines optimalen Steuerprozesses ist allgemeiner für Randbedingungen der Form (2.29) gehalten, so dass für gegebene Anfangswerte $x(t_0) := x$

$$\psi(x(t_0), x(t_f)) = x(t_0) - x \quad (3.1)$$

zu setzen ist. Als letzten noch fehlenden Bestandteil sei hier das Optimalitätskriterium folgendermaßen definiert:

Definition 3.1 (Zielfunktional).

Sei $U \subset \mathbb{R}^m$, $\Phi_0 : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ eine stetig differenzierbare und $L_0 : \mathbb{R}^n \times U \rightarrow \mathbb{R}$ eine stetig und bezüglich $x(t) \in \mathbb{R}^n$ und $u(t) \in U$ stetig partiell differenzierbare Funktion. Dann wird

$$J(x(t), u(t)) := \Phi_0(x(t_0), x(t_f)) + \int_{t_0}^{t_f} L_0(x(t), u(t)) dt \quad (3.2)$$

als Zielfunktional bezeichnet.

In der Literatur werden die Funktionen $\Phi_0(x(t_0), x(t_f))$ Mayer-Anteil und $L_0(x(t), u(t))$ Lagrange-Anteil genannt. Der Lagrange-Anteil stellt dabei die Bewertung des Lösungspaares $(x(t), u(t))$ für alle Zeitpunkte $t \in [t_0, t_f]$ dar, das bedeutet insbesondere, dass sowohl der Zustand als auch die Steuerung bestraft werden können, wohingegen der Mayer-Anteil nur von Anfangs- und Endzustand abhängig ist. Somit lässt sich mit dem Mayer-Anteil etwa eine Bestrafung für die Nichteinhaltung von Randwerten (2.29) realisieren.

Kompakt stellt sich der optimale Steuerprozess durch die Vereinbarungen zu den Definitionen von (2.24), (2.26), (2.29) und (3.2) folgendermaßen dar:

Definition 3.2 (Optimaler Steuerprozess ohne Beschränkungen).

Das Problem

$$\begin{aligned} \text{Minimiere} \quad J(x(t), u(t)) &= \Phi_0(x(t_0), x(t_f)) + \int_{t_0}^{t_f} L_0(x(t), u(t)) dt \\ \text{unter} \quad \dot{x}(t) &= f(x(t), u(t)) \\ \psi(x(t_0), x(t_f)) &= 0 \\ u(t) &\in U, \quad t \in [t_0, t_f] \end{aligned}$$

heißt optimaler Steuerprozess **OCP** oder optimales Steuerungsproblem.

Die hier vorliegende Darstellung wird auch Bolza-Problem genannt. Andere Arten von Steuerungsproblemen unterscheiden sich nur in der Beschreibung des Zielfunktional und können ineinander überführt werden:

$$\begin{aligned} \text{Lagrange-Problem} : \quad & \Phi_0 \equiv 0 \\ \text{Mayer-Problem} : \quad & L_0 \equiv 0 \\ \text{Tschebyscheff-Problem} : \quad & J(x, u) = \max\{L_0(x(t), u(t), t)\} \end{aligned}$$

Ebenso können Probleme mit freier Endzeit auf Probleme mit fester Endzeit und nicht-autonome auf autonome Probleme transformiert werden. Entsprechende explizite Formeln hierzu finden sich in Kapitel 4.2 der Arbeit von Ch.Büskens [2]. Das Optimalsteuerungsproblem heißt dabei autonom, wenn L_0 und f nicht explizit von t abhängen.

Bemerkung 3.3. *Dabei eignen sich bestimmte Darstellungsweisen des Zielfunktional besser für analytische oder numerische Zwecke, obwohl sie äquivalent ineinander überführt werden können. Beispielsweise hat sich das Mayer-Funktional in der numerischen Anwendung als vorteilhaft herausgestellt. In der analytischen Betrachtung wird jedoch häufig die Lagrange-Form zugrunde gelegt.*

Im weiteren Verlauf spielt die sogenannte Wertefunktion

$$V(x) := \inf_{u \in \mathcal{U}} J(x, u) \quad (3.3)$$

vor allem in der Beweisführung eine besondere Rolle, sie bildet aber auch die Grundlage für das folgende fundamentale Prinzip.

3.2 Bellmansches Optimalitätsprinzip und Hamilton-Jacobi-Bellman Gleichung

Als Basis einer numerischen Approximation einer optimalen Lösung stellt sich eine Eigenschaft der Wertefunktion V dar, die unter dem Namen Bellmansches Optimalitätsprinzip bekannt geworden ist. Dieses besagt, dass aus der Optimalität einer Trajektorie die Optimalität jedes Endstückes folgt. Bei Vorliegen einer optimalen Steuerung wird also zu jedem Zeitpunkt optimal gesteuert.

Satz 3.4 (Bellmansches Optimalitätsprinzip).

Gegeben sei ein optimales Steuerungsproblem gemäß Definition 3.2. Dann gilt

$$V(x) = \inf_{u \in \mathcal{U}} \left\{ \int_0^T L_0(x(t, x, u), u(t)) dt + V(x(T, x, u)) \right\} \quad (3.4)$$

für alle $x \in \mathbb{R}^n$ und alle $T > 0$.

Ein entsprechender Beweis für das sogenannte diskontierte Zielfunktional

$$J(x, u) := \int_0^\infty e^{-\delta t} L_0(x(t, x(t_0), u), u(t)) dt, \quad (3.5)$$

das einen Sonderfall von (3.2) darstellt, ist in Kapitel 2.4 der Arbeit von L.Grüne [9] nachzulesen. Dort wird sogar weitergehend bewiesen, dass die durch das Optimalitätsprinzip bestimmte Wertefunktion V eindeutig ist.

Aus dem Optimalitätsprinzip kann man auf eine weitere wichtige Gleichung schließen, die hier zumindest kurz für ein Zielfunktional der Form (3.5) erwähnt werden soll.

Satz 3.5 (Hamilton-Jacobi-Bellman Gleichung).

Gegeben sei die partielle Differentialgleichung

$$\delta V(x) + \sup_{u \in \mathcal{U}} \{-DV(x) \cdot f(x, u) - L_0(x, u)\} = 0, \quad (3.6)$$

die sogenannte Hamilton-Jacobi-Bellman Gleichung, und ein optimales Steuerungsproblem **OCP** mit Zielfunktional (3.5) und optimaler Wertefunktion V . Ist die optimale Wertefunktion V differenzierbar in $x \in \mathbb{R}^n$, dann gilt die Hamilton-Jacobi-Bellman Gleichung für V in diesem Punkt.

Da die Lösung differenzierbar sein muss stellt dies eine starke Einschränkung dar, die in der Beweisführung aus Kapitel 2.5 der Arbeit von L.Grüne [9] explizit verwendet wird. Für nichtdifferenzierbare Lösungen dieser partiellen Differentialgleichung lassen sich dennoch Aussagen zur Existenz und Eindeutigkeit treffen, die auf dem Konzept sogenannter Viskositätslösungen basieren. Hierzu sei auf die Bücher von M.Bardi und I.Capuzzo Dolcetta sowie L.C.Evans [1, 4] verwiesen.

Bemerkung 3.6. Eine direkte Folgerung ist jedoch die Aussage, dass jedes (numerische) Schema zur Bestimmung einer optimalen Steuerung und somit einer optimalen Wertefunktion als Schema zur Lösung der Hamilton-Jacobi-Bellman Gleichung angesehen werden kann und umgekehrt.

3.3 Modellprädiktive Regelung nichtlinearer sampled-data Systeme

Implizit ergibt sich der Ansatz der modellprädiktiven Regelung aus dem Optimalitätsprinzip (Bellman, 1957), im Speziellen seien hier Lee und Markus aus dem Jahr 1967 mit folgender Aussage zitiert:

One technique for obtaining a feedback controller synthesis from knowledge of open-loop controllers is to measure the current control process state and then compute very rapidly for the open-loop control function. The first portion of this function is then used during a short time interval, after which a new measurement of the process state is made and a new open-loop control function is computed for this new measurement. The procedure is then repeated.

Bevor näher auf die Problemstellung eingegangen wird sollen Anforderungen formuliert werden, denen die zu untersuchenden Systeme genügen müssen. Die Problemstellung orientiert sich dabei an der Arbeit von D.Nešić und L.Grüne [23].

3.3.1 Grundannahmen

Für alle folgenden Kapitel sollen die folgenden Annahmen für das zeitkontinuierliche System

$$\dot{x} = f(x, u) \quad (3.7)$$

mit einem Zustandsvektor $x \in \mathbb{R}^n$, einer Steuerung $u \in \mathcal{U}$ und einer Funktion $f : \mathbb{R}^n \times \mathcal{U} \rightarrow \mathbb{R}^n$ gelten:

- Ein zeitkontinuierliches Zustandsfeedback $u = u(x(t))$ mit $u(x(t)) \in U$ für alle $x(t) \in \mathbb{R}^n$ wurde für das System derart konzipiert, dass das zeitkontinuierliche closed-loop System

$$\dot{x}(t) = f(x(t), u(x(t))), \quad x(t_0) = x \quad (3.8)$$

global asymptotisch stabil ist.

- Zudem wird angenommen, dass $f(x(t), u(x(t)))$ lokal Lipschitz-stetig ist in x , so dass für jedes $x(t_0) = x$ eine eindeutige Lösung des zeitkontinuierlichen closed-loop Systems (3.8) existiert.

Im Folgenden wird die Lösung des Systems (3.7) zum Zeitpunkt t und Anfangswert x mit $x(t, x)$ oder auch $x(t)$ bezeichnet.

Für das sampled-data System ist die Steuerung derart zu implementieren, dass sie stückweise konstant ist zwischen zwei Diskretisierungspunkten $t_k, t_{k+1} \in \mathbb{R}$ der festen Periode $T \in \mathbb{R}$, $T > 0$.

$$\Rightarrow t_k := k \cdot T \quad (3.9)$$

T stellt somit gleichzeitig die sampled-data Rate und die Folge $(t_k)_{k \in \mathbb{N}}$ die sampled-data Folge dar. Es gilt also für

$$\dot{\xi}(t) = f(\xi(t), v(t)) \quad (3.10)$$

mit Zustandsvektor $\xi \in \mathbb{R}^n$, einem Steuersignal $v(t) = v_k \in U \subset \mathbb{R}^m$, $t \in [t_k, t_{k+1})$, $k \in \mathbb{N}$ und einer Funktion $f : \mathbb{R}^n \times U \rightarrow \mathbb{R}^n$:

- Die Steuerfolge v_k , $k \in [0, M]$ wird mit $v_{[0, M]}$ bezeichnet.
- Sei $\xi_1 = \xi(t_1) \in \mathbb{R}^n$ die Lösung des Systems

$$\dot{\xi}(t) = f(\xi(t), v_0), \quad \xi(t_0) = \xi,$$

dann ergibt sich die Lösung des sampled-data Systems im Zeitintervall $[t_1, t_2]$ aus der Lösung des zeitkontinuierlichen Systems

$$\dot{\xi}(t) = f(\xi(t), v_1), \quad \xi(t_1) = \xi_1.$$

- Die Lösung des sampled-data Systems zum Zeitpunkt t und Startwert ξ unter der Steuerfolge $v_{[0, M]}$ wird mit $\xi(t, \xi, v_{[0, M]})$ oder auch $\xi(t)$ bezeichnet.

Bemerkung 3.7. *Da es sich um autonome Differentialgleichungen handelt kann ohne Beschränkung der Allgemeinheit der Anfangszeitpunkt $t_0 = 0$ gesetzt werden. Zudem wird aus Vereinfachungsgründen die Notation $(\xi, x) := (\xi^T, x^T)^T$ verwendet.*

3.3.2 Problemstellung des modellprädiktiven Regelungsansatzes

Das Problem der stückweise konstanten Implementierung einer kontinuierlichen Steuerung soll hier im Rahmen der Lösung optimaler Steuerungsprobleme und des sogenannten modellprädiktiven Regelungsansatzes erfolgen. Zweck der so bestimmten Steuerung ist es, dass kontinuierliche Systeme mit Hilfe von Computern gesteuert werden können, die auf Grund ihrer zeitdiskreten Natur nur in der Lage sind zu jeweils festgelegten Zeitpunkten einzugreifen. Dabei sind optimale Steuerungsprobleme der Form

$$\text{Finde } \hat{u}_{[0, M-1]} = \arg \inf_{v_{[0, M-1]}} J_M(\xi(t), x(t), v_{[0, M-1]}) \quad (3.11)$$

unter

$$J_M(\xi(t), x(t), v_{[0, M-1]}) := \sum_{i=0}^{M-1} \int_{t_i}^{t_{i+1}} l(\xi(t) - x(t), v_i) dt + F(\xi(t_M), x(t_M)) \quad (3.12)$$

$$v_i \in U \quad \forall i \in \{0, \dots, M-1\} \quad (3.13)$$

$$\dot{\xi}(t) = f(\xi(t), v_i), \quad \xi(t_0) = \xi, \quad t \in [t_i, t_{i+1}) \quad (3.14)$$

$$\dot{x}(t) = f(x(t), u(x(t))), \quad x(t_0) = x \quad (3.15)$$

$$t_i = t_0 + i \cdot T, \quad i \in \{0, \dots, M\} \quad (3.16)$$

auf Zeitintervallen

$$[t_0, t_M] = [j \cdot T, (j+M) \cdot T], \quad j \in \mathbb{N}, \quad (3.17)$$

im Folgenden auch Horizont genannt, zu betrachten. Das Zielfunktional J_M in Bolza-Form bewertet also mit dem Lagrange-Anteil $l: \mathbb{R}^n \times U \rightarrow \mathbb{R}_0^+$ die Differenz zwischen der sampled-data Lösung $\xi(\cdot)$ und der zeitkontinuierlich gesteuerten Lösung $x(\cdot)$ auf dem zu betrachtenden Horizont ebenso wie die aufgebrauchte Steuerenergie. Da dies zu jedem Zeitpunkt des Horizonts erfolgt wird l auch als "stage-cost" bezeichnet. Der Mayer-Anteil $F: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}_0^+$ dagegen muss nicht unbedingt als Strafe für die Abweichung im Endzeitpunkt gesehen werden, sondern kann auch zur Abschätzung der Trajektorienentwicklung über den Horizont hinaus verwendet werden. Dabei sollen für das zeitkontinuierliche und das sampled-data System die Grundannahmen gelten.

Bemerkung 3.8. *Es ist für diese Art des modellprädiktiven Regelungsansatzes nicht unbedingt notwendig, dass die Anfangswerte $\xi \in \mathbb{R}^n$ für die sampled-data Lösung mit den Anfangswerten $x \in \mathbb{R}^n$ der zeitkontinuierlichen Lösung übereinstimmen. Da man allerdings die Differenz zwischen beiden zu minimieren versucht, wird im Allgemeinen $\xi = x$ gewählt.*

Bemerkung 3.9. *F kann dabei zudem eine Abschätzung für den Fall des unendlichen Horizonts darstellen und wird auch 'cost-to-go' genannt:*

$$V_\infty(\xi, x) := \inf_{v_{[0, \infty)}} \sum_{i=0}^{\infty} \int_{t_i}^{t_{i+1}} l(\xi(t) - x(t), v_i) dt \quad (3.18)$$

Im Allgemeinen kann hierfür eine Lyapunov-Kontroll-Funktion verwendet werden, die grob gesagt eine obere Schranke darstellen soll. Idealerweise soll F dem Kostenfunktional mit unendlichen Horizont entsprechen. Damit wäre die online Optimierung unnötig und die bekannten Vorteile des unendlichen Horizontes, Stabilität und Robustheit, würden automatisch zutreffen. Entsprechende Aussagen finden sich in [11, 19]. Im linearen Fall führt dies zu H_2 und H_∞ Lösungen, im nichtlinearen Fall jedoch ist zwingend die Lösung einer Hamilton-Jacobi Gleichung erforderlich, die nur sehr schwer zu erhalten ist.

Bemerkung 3.10. Es wird im Weiteren aber davon ausgegangen und bewiesen, dass F nicht zwingend eine Lyapunov-Kontroll-Funktion sein muss um die Stabilität zu garantieren. Hierzu sei auf das Kapitel zur Optimalität der modellprädiktiven Regelung verwiesen. Da die Bestimmung einer Lyapunov-Kontroll-Funktion sehr schwierig sein kann stellt dies eine enorme Erleichterung dar.

Die Vorgehensweise des modellprädiktiven Regelungsansatzes wird dadurch verwirklicht, dass für die bestimmte sampled-data Lösung nur der erste Wert der Steuerfunktion

$$u = u_M(\xi, x) := \hat{u}_0(\xi, x) \quad (3.19)$$

in der optimalen Steuerfolge $\hat{u}_{[0, M-1]}$ übernommen wird. Anschließend wird, wie in Abbildung 3.2 veranschaulicht, der zu betrachtende Horizont $[0, M \cdot T]$ um die sampled-data Rate $T > 0$ verschoben zu $[T, (M + 1) \cdot T]$.

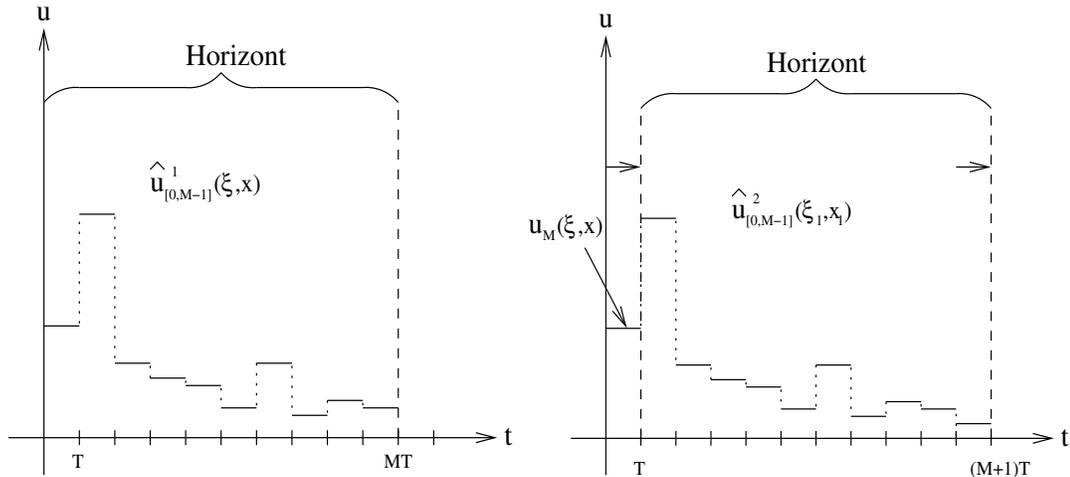


Abbildung 3.2: Vorgehensweise des modellprädiktiven Regelungsansatzes

Die zur Lösung des sich daraus ergebenden neuen optimalen Steuerungsproblems notwendigen Anfangswerte der zeitkontinuierlichen und sampled-data Systeme ergeben sich durch die Anwendung des übernommenen ersten Wertes der optimalen Steuerfolge. Somit lässt sich eine optimale Steuerfolge für das neue optimale Steuerungsproblem bestimmen und die

beschriebene Vorgehensweise kann iterativ angewandt werden.

Die modellprädiktive Regelung (3.19) liefert daher ein statisches Zustandsfeedback, das diskretisiert implementiert ist, so dass das closed-loop System folgendermaßen beschrieben werden kann:

$$\dot{\xi}(t) = f(\xi(t), u_M(\xi(t_k), x(t_k))), \quad \xi(0) = \xi, \quad t \in [t_k, t_{k+1}) \quad (3.20)$$

$$\dot{x}(t) = f(x(t), u(x(t))), \quad x(0) = x \quad (3.21)$$

Ziel ist es nun zu zeigen, dass dieses System asymptotisch stabil und die mit Hilfe des modellprädiktiven Ansatzes bestimmte Regelung in angebrachter Weise suboptimal ist. Dabei würde die asymptotische Stabilität von (3.20), (3.21) die Existenz einer Funktion $\beta \in \mathcal{KL}$ gemäß Definition 2.7 (ii) implizieren, weswegen Tracking, also

$$\lim_{t \rightarrow \infty} \|\xi(t) - x(t)\| = 0, \quad (3.22)$$

möglich wäre und so das gewünschte Langzeitverhalten garantiert werden kann.

Explizit formuliert hat das Problem des modellprädiktiven Regelungsansatzes die Form **MPC**

$$\text{Bestimme } u = u_M(\xi, x) := \hat{u}_0^j(\xi, x), \quad (3.23)$$

wobei

$$\hat{u}_{[0, M-1]}^j = \arg \inf_{v_{[0, M-1]}} J_M^j(\xi(t), x(t), v_{[0, M-1]}) \quad (3.24)$$

unter

$$J_M^j(\xi(t), x(t), v_{[0, M-1]}) := \sum_{i=0}^{M-1} \int_{t_i^j}^{t_{i+1}^j} l(\xi(t) - x(t), v_i) dt + F(\xi(t_M^j), x(t_M^j)) \quad (3.25)$$

$$v_i \in U \quad \forall i \in \{0, \dots, M-1\} \quad (3.26)$$

$$\dot{\xi}(t) = f(\xi(t), v_i), \quad \xi(t_0^j) = \xi_j, \quad t \in [t_i^j, t_{i+1}^j) \quad (3.27)$$

$$\dot{x}(t) = f(x(t), u(x(t))), \quad x(t_0^j) = x_j \quad (3.28)$$

$$t_i^j = t_0^0 + (i + j - 1) \cdot T, \quad i \in \{0, \dots, M\}, \quad j \in \mathbb{N} \setminus \{0\} \quad (3.29)$$

Auf die Eigenschaften dieses Problems soll in den folgenden Kapiteln eingegangen werden. Jedoch werden zunächst verschiedene Ansätze zur Lösung des dem modellprädiktiven Ansatz zugrunde liegenden optimalen Steuerungsproblems erörtert.

3.4 Indirekte Verfahren zur Lösung optimaler Steuerprozesse

Bei den indirekten Verfahren wird zunächst analytisch der optimale Steuerprozess mittels der notwendigen Bedingungen des Pontryaginschen Minimumprinzips in ein Randwertproblem überführt, das anschließend mit numerischen Methoden wie etwa Einfach-, Mehrfach-Schießverfahren oder Kollokationsverfahren gelöst werden kann.

Bemerkung 3.11. *Trotz der hohen Genauigkeit der Lösung und der leichten Überprüfbarkeit notwendiger Bedingungen werden diese Verfahren nicht mehr sehr häufig eingesetzt, da mit direkten Verfahren eine größere Problemklasse leichter abgedeckt werden kann und nicht für jedes einzelne Problem die Vorarbeiten durchzuführen sind, wie sie das Pontryaginsche Minimumprinzip erfordert. Zudem entstehen Mehrpunkt-Randwertprobleme, wenn der optimale Steuerprozess beschränkt ist, die weitere numerische Probleme verursachen.*

Zentrales Hilfsmittel des Pontryaginschen Minimumprinzips ist die sogenannte adjungierte Differentialgleichung:

Definition 3.12 (Hamiltonfunktion und adjungierte Variable).

Sei $\lambda_0 \in \mathbb{R}$ und $\lambda \in \mathbb{R}^n$, dann heißt unter Beachtung der Vereinbarungen zu den Definitionen von (2.24), (2.26), (2.29) und (3.2)

$$H(x(t), \lambda, u(t), t) = \lambda_0 \cdot L_0(x(t), u(t)) + \lambda^T \cdot f(x(t), u(t)) \quad (3.30)$$

*Hamiltonfunktion zu einem unbeschränkten optimalen Steuerprozess **OCP**.*

Dabei wird $\lambda \in \mathbb{R}^n$ adjungierte Variable oder Kozustand zu x genannt.

Ökonomisch lässt sich die adjungierte Variable λ_j , $j \in \{1, \dots, n\}$, als Schattenpreis interpretieren, also nicht als Marktpreis, der für eine marginale zusätzliche Menge von x_j zu zahlen wäre, sondern den Preis, den man bereit wäre zu zahlen ohne dabei den Wert der optimalen Wertefunktion zu ändern.

Die totale Auswirkung einer Steuerung u zu einem Zeitpunkt t wird von der Hamiltonfunktion beschrieben. Die Auswirkung kann dabei zerlegt werden in

- die Wirkung, die diese Steuerung mit dem vorliegenden Zustand x auf den Zustand des Systems über dessen Dynamik $f(x, u)$ haben (ökonomisch entspricht dies der Wertänderung des Kapitalstocks) und
- den Effekt, der direkt mit L_0 in das Zielfunktional eingeht (die Profitrate).

Eine optimale Lösung des Steuerprozesses **OCP** zur festen (freien) Endzeit t_f zu finden bedeutet, ein zulässiges Funktionenpaar (x^*, u^*) zu bestimmen, so dass

$$J(x^*, u^*) \leq J(x, u) \quad (3.31)$$

für alle zulässigen Paare (x, u) zur (beliebigen) Endzeit $t_f (> 0)$ gilt. Dabei wird x^* auch als optimale Trajektorie und u^* als optimale Steuerung bezeichnet.

Zur Berechnung einer optimalen Lösung eignet sich diese Aussage allerdings nicht, da hierzu das Zielfunktional für alle zulässigen Lösungen ausgewertet werden müsste. Daher werden für unbeschränkte optimale Steuerprozesse **OCP** die notwendigen Bedingungen des Pontryaginschen Minimumprinzips verwendet.

Satz 3.13 (Pontryaginsches Minimumprinzip - Notwendige Bedingungen).

Für eine optimale Lösung (x^*, u^*) eines optimalen Steuerprozesses ohne Beschränkungen **OCP** mit Hamiltonfunktion (3.30) existieren eine reelle Zahl $\lambda_0 \geq 0$, eine stetige und stückweise stetig differenzierbare Funktion $\lambda : [t_0, t_f] \rightarrow \mathbb{R}^n$ sowie ein Vektor $\sigma \in \mathbb{R}^r$, so dass gilt:

- $(\lambda_0, \lambda^T(t), \sigma^T) \neq 0 \quad \forall t \in [t_0, t_f]$
- An allen Stetigkeitsstellen t von u^* in $[t_0, t_f]$ gelten
 - Minimumbedingung

$$H(x^*, u^*, \lambda, t) = \min_{u \in U} \{H(x^*, u, \lambda, t)\} \quad (3.32)$$

- Adjungierte Differentialgleichung

$$\dot{\lambda}^T = -\nabla_x H(x^*, u^*, \lambda, t) \quad (3.33)$$

- Transversalitätsbedingung

$$\lambda(t_0) = -\nabla_{x(t_0)} (\lambda_0 \cdot \Phi_0(x^*(t_0), x^*(t_f)) + \sigma^T \cdot \psi(x^*(t_0), x^*(t_f)))^T \quad (3.34)$$

$$\lambda(t_f) = \nabla_{x(t_f)} (\lambda_0 \cdot \Phi_0(x^*(t_0), x^*(t_f)) + \sigma^T \cdot \psi(x^*(t_0), x^*(t_f)))^T \quad (3.35)$$

- Im Fall freier Endzeit gilt für die optimale Endzeit t_f^*

$$H(x^*(t_f^*), u^*(t_f^*), \lambda(t_f^*), t_f^*) = 0. \quad (3.36)$$

- Im Fall autonomer Probleme gilt

$$H(x^*, u^*, \lambda, t) = \text{konstant} \quad \forall t \in [t_0, t_f]. \quad (3.37)$$

Der Beweis des Satzes ist in der Arbeit von L.S.Pontryagin, V.G.Boltjanskij, R.V.Gamkrelidze und E.F.Miscenko [25] nachlesbar.

Die Idee des Minimumprinzips ist dabei die Konstruktion eines Systems von adjungierten Differentialgleichungen, wodurch auch die indirekten Folgen einer Steuerung berücksichtigt werden und damit das Ergebnis minimiert wird. Ergebnis dieser analytischen Lösung ist ein Differentialgleichungssystem, das aus doppelt so vielen Differentialgleichungen besteht wie das zugrunde liegende Kontrollsystem (2.24). Dieses entsteht aus der Minimierung der Hamiltonfunktion zu jedem Zeitpunkt $t \in [t_0, t_f]$ und der dementsprechenden Bestimmung der adjungierten Differentialgleichung. Die Transversalitätsbedingungen schließlich liefern die zusätzlichen Randwerte, die zur Lösung des Randwertproblems notwendig sind.

Bemerkung 3.14. Für den modellprädiktiven Regelungsansatz würde dies bedeuten, dass ein Mehrpunkt-Randwertproblem mit $4n$ Differentialgleichungen zu lösen ist und zudem die Unstetigkeitsstellen der Steuerung berücksichtigt werden müssen.

3.5 Direkte Verfahren zur Lösung optimaler Steuerprozesse

Bei direkten Verfahren diskretisiert man den optimalen Steuerprozess und reduziert diesen dadurch auf ein Optimierungsproblem, welches im Anschluss durch numerische Verfahren, etwa SQP-Verfahren, gelöst werden kann. Bei der Diskretisierung kann man zwischen verschiedenen Implementierungen wählen. Hier soll im Kurzen auf die sogenannte volle Diskretisierung und den rekursiven Ansatz eingegangen werden.

Zentraler Vorteil der direkten Verfahren ist die Tatsache, dass das optimale Steuerungsproblem nicht zuerst analytisch umzuformen ist, sondern dass man dieses direkt diskretisiert. Somit sind keine Vorkenntnisse der optimalen Steuerungstheorie, insbesondere der Struktur der Lösung und Schätzungen kritischer Punkte erforderlich. Ebenso kann auf die Berechnung der adjungierte Differentialgleichung verzichtet werden, die aber ex post nachgeholt werden kann.

Hierbei soll ein allgemeineres Problem betrachtet werden, als dies eigentlich für das modellprädiktive Regelungsproblem nötig wäre. Dazu werden die sogenannten Steuerbeschränkungen $u(t) \in U$, $t \in [t_0, t_f]$ in allgemeinerer Form über den Vektor der Beschränkungen aufgenommen.

Definition 3.15 (Vektor der Beschränkungen).

Sei $s \in \mathbb{N}$, $s \geq 1$ und $c(x(t), u(t)) = (c_1(x(t), u(t)), \dots, c_s(x(t), u(t)))^T$.

(1) Sei $c_k : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ für $k \in \{1, \dots, s\}$. Dann heißen die Funktionen c_k mit

$$c_k(x(t), u(t)) \leq 0 \quad \forall t \in [t_0, t_f] \quad (3.38)$$

gemischte Beschränkungen.

(2) Sei $c_k : \mathbb{R}^n \rightarrow \mathbb{R}$ für $k \in \{1, \dots, s\}$. Dann werden Beschränkungen der Form

$$c_k(x(t)) \leq 0 \quad \forall t \in [t_0, t_f] \quad (3.39)$$

Zustandsbeschränkungen genannt.

(3) Seien $s_m, s_z, s \in \mathbb{N}$, $s := s_m + s_z$ und $c : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^s$ mit

$$c_k(x(t), u(t)) := \begin{cases} c_k(x(t), u(t)), & k = 1, \dots, s_m, \\ c_k(x(t)), & k = s_m + 1, \dots, s. \end{cases} \quad (3.40)$$

Dann wird

$$c(x(t), u(t)) \leq 0 \quad \forall t \in [t_0, t_f] \quad (3.41)$$

Vektor der Beschränkungen genannt.

Für den optimalen Steuerprozess bedeutet dies, dass neben den Steuerbeschränkungen nun auch Beschränkungen an den Zustand und gemischte Beschränkungen an Zustand und Steuerung vorliegen können. Die Ordnung einer Beschränkung wird dabei wie folgt definiert:

Definition 3.16 (Ordnung einer Beschränkung).

Sei $p \in \mathbb{N}^s$ mit $0 \leq p_i \leq p_{max}$, wobei $p_{max} \in \mathbb{N}$ mit $p_{max} := \max(p_i : i = 1, \dots, s)$ gesetzt wird. Weiter sei f eine $C^{p_{max}}$ -Funktion. Dann heißt p_i die Ordnung der Beschränkung c_i bezüglich der Differentialgleichung $\dot{x}(t) = f(x(t), u(t))$ mit

$$p_i := \begin{cases} 0 & i \in \{1, \dots, s_m\}, \\ \tilde{p} : \tilde{p} \text{ ist Ordnung der Funktion } c_i & i \in \{s_m + 1, \dots, s\}. \end{cases} \quad (3.42)$$

p heißt dabei Ordnung der Funktion $F : \mathbb{R}^n \rightarrow \mathbb{R}$ bezüglich der Differentialgleichung $\dot{x}(t) = f(x(t), u(t))$, falls

$$\nabla_u F^i = 0, \quad i = 0, \dots, p - 1 \quad (3.43)$$

$$\nabla_u F^p \neq 0 \quad (3.44)$$

für C^p -Funktionen $F^i : \mathbb{R}^n \rightarrow \mathbb{R}$, die sich rekursiv definieren durch:

$$F^0 = F, \quad F^i = \nabla_x (F^{i-1})f, \quad i = 1, \dots, p. \quad (3.45)$$

Da für das zu betrachtende Problem der modellprädiktiven Regelung eine einheitliche sampled-data Rate einzuhalten ist soll hier die Diskretisierung äquidistant auf einem Gitter der Stufe n erfolgen, das wie folgt definiert ist:

Definition 3.17 (Gitter der Stufe \hat{n}).

Gegeben sei ein Intervall $\mathcal{D} = [t_0, t_f] \subset \mathbb{R}$ mit $-\infty < t_0 < t_f < \infty$, $n_0 \in \mathbb{N}$ fest, $l_{\hat{n}} := n_0 \cdot 2^{\hat{n}}$ und $h_{\hat{n}} := \frac{t_f - t_0}{l_{\hat{n}} - 1}$, $\hat{n} \in \mathbb{N}$. Dann wird die Menge

$$\mathcal{D}_{\hat{n}} := \{y_j \in \mathbb{R} | y_j = t_0 + j \cdot h_{\hat{n}}, j = 1, \dots, l_{\hat{n}}\} \quad (3.46)$$

als Gitter der Stufe \hat{n} auf \mathcal{D} bezeichnet.

Bemerkung 3.18. Für die Anwendung auf die vorliegende Problemstellung (3.24) unter (3.25), (3.26), (3.27), (3.28) und (3.29) setzt man dabei $n_0 = M$, da auf Grund der möglichen Unstetigkeit der Steuerung $\hat{u}_{[0, M-1]}^j \in \mathcal{U}$ an den Stellen $t_k = k \cdot T$, $k \in \{1, \dots, M\}$, $k \in \mathbb{N}$, zur Vermeidung von Fehler bei der numerischen Integration die äquidistante Diskretisierung der sampled-data Rate T angepasst werden sollte.

Bemerkung 3.19. Die Lösung liegt nach einer Diskretisierung nur noch an den isolierten Gitterpunkten vor und Werte zwischen diesen diskreten Punkten können lediglich approximativ bestimmt werden. Zudem führt die Diskretisierung zu Problemen bezüglich der Stabilität, der Konvergenz und insbesondere der Genauigkeit.

3.5.1 Lösung mittels voller Diskretisierung

Da es sich aus numerischer Sicht empfiehlt ein optimales Steuerungsproblem mit Zielfunktional in Mayer-Form zu berechnen, soll bei der Erläuterung der verschiedenen Diskretisierungsarten auf diese spezielle Form zurückgegriffen werden. Da die Probleme äquivalent ineinander überführbar sind stellt dies zudem keine Einschränkung dar. Es wird nun die allgemeine Vorgehensweise der Diskretisierungsansätze erläutert.

Die Diskretisierung muss dabei nicht äquidistant sein, zur Vereinfachung der Darstellung wird hier allerdings ein äquidistantes Gitter verwendet. Es sei $N_t \in \mathbb{N}, N_t > 0$ die Zahl der Diskretisierungsstellen und

$$\tau_i \in \mathbb{R}, i = 1, \dots, N_t \text{ mit } \tau_i = t_0 + h \cdot (i - 1), i = 1, \dots, N_t, h = \frac{t_f - t_0}{N_t - 1} \quad (3.47)$$

seien äquidistante Gitterpunkte. Die approximierten Werte $x(\tau_i) \in \mathbb{R}^n$ und $u(\tau_i) \in \mathbb{R}^m$ werden mit x^i und u^i bezeichnet. Somit kann der optimale Steuerprozess **OCP** mit Beschränkungen der Form (3.41) mittels des Euler-Verfahrens durch

$$\text{Minimiere } J_0(x, u) = \Phi_0(x^1, x^{N_t}) \quad (3.48)$$

$$\text{unter } x^{i+1} = x^i + h \cdot f(x^i, u^i), i = 1, \dots, N_t - 1 \quad (3.49)$$

$$\psi(x^1, x^{N_t}) = 0 \quad (3.50)$$

$$c(x^i, u^i) \leq 0, i = 1, \dots, N_t \quad (3.51)$$

approximativ ersetzt werden.

Dies kann auch als nichtlineares Optimierungsproblem dargestellt werden, wobei

$$z := (x^1, \dots, x^{N_t}, u^1, \dots, u^{N_t}) \in \mathbb{R}^{N_z}, N_z \in \mathbb{N}, N_z = (n + m) \cdot N_t \quad (3.52)$$

die neue Optimierungsvariable darstellt. Bezeichnet man die Anzahl an Gleichungsnebenbedingungen mit $N_E \in \mathbb{N}$ und die der Ungleichungsnebenbedingungen mit $N_I \in \mathbb{N}$, dann hat das nichtlineare Optimierungsproblem folgende Form:

$$\text{Minimiere } F(z) = \Phi_0(x^1, x^{N_t}) \quad (3.53)$$

$$\text{unter } G(z) = \begin{pmatrix} [-x^{i+1} + x^i + h \cdot f(x^i, u^i)]_{i=1, \dots, N_t-1} \\ \psi(x^1, x^{N_t}) \\ c(x^1, u^1) \\ \vdots \\ c(x^{N_t}, u^{N_t}) \end{pmatrix} \in \mathbb{R}^{N_E + N_I} \quad (3.54)$$

mit $N_E = n \cdot (N_t - 1) + r$, $N_E + N_I = (n + k) \cdot N_t - n + r$.

3.5.2 Lösung mittels eines rekursiven Ansatzes

Es gelte wieder (3.47). Die approximierten Werte $x(\tau_i) \in \mathbb{R}^n$ und $u(\tau_i) \in \mathbb{R}^m$ werden weiter mit x^i und u^i bezeichnet. Jedoch wird hierbei der Optimierungsvektor

$$z := (x^1, u^1, \dots, u^{N_t}) \in \mathbb{R}^{N_z}, \quad N_z = n + m \cdot N_t \quad (3.55)$$

verwendet, wobei mit x^1 lediglich der Anfangswert in z dazu aufgenommen wird, um rekursiv mit der Euler-Approximation die Zustandstrajektorie

$$x^1(z) = x^1, \quad x^{i+1}(z) = x^i(z) + h \cdot f(x^i(z), u^i), \quad i = 1, \dots, N_t - 1 \quad (3.56)$$

berechnen zu können. Die Anzahl an Gleichungsnebenbedingungen wird mit $N_E \in \mathbb{N}$ und die der Ungleichungsnebenbedingungen mit $N_I \in \mathbb{N}$ bezeichnet. Das neue nichtlineare Optimierungsproblem, das man damit aus dem optimalen Steuerprozess **OCP** mit Beschränkungen der Form (3.41) erhält, hat folgende Form:

$$\text{Minimiere} \quad F(z) = \Phi_0(x^1(z), x^{N_t}(z)) \quad (3.57)$$

$$\text{unter} \quad G(z) = \begin{pmatrix} \psi(x^1(z), x^{N_t}(z)) \\ c(x^1(z), u^1) \\ \vdots \\ c(x^{N_t}(z), u^{N_t}) \end{pmatrix} \in \mathbb{R}^{N_E + N_I} \quad (3.58)$$

mit $N_E = r$, $N_E + N_I = k \cdot N_t + r$.

Bemerkung 3.20. Sowohl die Anzahl der Optimierungsvariablen als auch die Anzahl der Nebenbedingungen verringern sich bei diesem Verfahren drastisch. Die Zustandsvariablen jedoch sind jetzt Funktionen der diskreten Steuervariablen.

Da es sich beim Ergebnis der vollständigen Diskretisierung wie des rekursiven Ansatzes um ein Optimierungsproblem handelt, können entsprechend bekannte Routinen zum Einsatz kommen. Als besonders geeignet haben sich dabei die sogenannten SQP-Verfahren erwiesen.

Bemerkung 3.21. Um ein Gitter der Stufe 0 zu implementieren und dieses zudem auf die Unstetigkeitsstellen der Steuerung im vorliegenden Problem abzustimmen ist dabei $N_t = M + 1$ zu wählen.

Daraus ergibt sich das Problem, dass $h = T$ gilt, wobei es das Ziel des modellprädiktiven Regelungsansatzes ist, dass T möglichst groß ist. Bei beiden Ansätzen ist es allerdings wichtig, dass die Differentialgleichung möglichst exakt gelöst wird, h also klein sein muss. Um die Optimierungsvariable und die Anzahl der Nebenbedingungen nicht zu vergrößern, wie dies für ein Gitter der Stufe $n > 0$ geschehen würde, kann hierzu eine Schrittweitensteuerung auf den Intervallen $[\tau_i, \tau_{i+1})$ der Länge T für τ_i aus (3.47) verwendet werden. Für die entsprechende Theorie sei dazu auf Kapitel 5 des Buchs von P.Deuffhard und F.Bornemann [3] verwiesen.

3.5.3 Konvergenz der Euler-Diskretisierung

Nach der Darstellung der Diskretisierung sollen an dieser Stelle notwendige und hinreichende Bedingungen für die Optimalität eines optimalen Steuerungsproblems gezeigt werden, da die daraus resultierenden Ergebnisse wesentliche Voraussetzungen für den Konvergenzbeweis diskretisierter optimaler Steuerprozesse darstellen.

Bemerkung 3.22. Die Konvergenz ergibt sich dabei für Schrittweiten $h \rightarrow 0$, die Problemstellung erfordert es aber $h = T$ zu setzen, welches möglichst groß gewählt werden soll. Die Konvergenz der Euler-Diskretisierung rechtfertigt jedoch den verwendeten Ansatzpunkt direkter Verfahren.

Hierfür soll das Problem $\overline{\text{OCP}}$

$$\begin{aligned} \text{Minimiere} \quad J_0(x, u) &= \Phi_0(x(t_f)) + \int_0^{t_f} L_0(x(t), u(t)) dt \\ \text{unter} \quad \dot{x}(t) &= f(x(t), u(t)), \quad \text{für fast alle } t \in [0, t_f] \\ x(0) &= x, \quad \phi(x(t_f)) = 0 \\ c(x(t), u(t)) &\leq 0, \quad \text{für fast alle } t \in [0, t_f] \end{aligned}$$

betrachtet werden mit Funktionen $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^r$, $c : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^s$, $\Phi_0 : \mathbb{R}^n \rightarrow \mathbb{R}$, $L_0 : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ und $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$.

Die optimale Lösung werde mit $(x^*, u^*) \in W^{1,\infty}(0, t_f; \mathbb{R}^n) \times L^\infty(0, t_f; \mathbb{R}^m)$ bezeichnet, wobei

$$W^{1,\infty}(t_0, t_f; \mathbb{R}^n) = \{x \in L^\infty(t_0, t_f; \mathbb{R}^n) \mid \dot{x} \in L^\infty(t_0, t_f; \mathbb{R}^n)\} \quad (3.59)$$

der Sobolev-Raum mit geeignetem Banach-Raum

$$L^\infty(t_0, t_f; \mathbb{R}^n) = \{x : [t_0, t_f] \rightarrow \mathbb{R}^n \mid x \text{ ist Lebesgue messbar, } \|x\|_{L^\infty([t_0, t_f])} < \infty\}, \quad (3.60)$$

$$\|x\|_{L^\infty([t_0, t_f])} = \text{ess sup}_{[t_0, t_f]} \|x\| \quad (3.61)$$

ist. Für dieses Problem lassen sich die notwendigen Bedingungen mittels der erweiterten Hamilton-Funktion formulieren:

Definition 3.23 (Erweiterte Hamiltonfunktion).

Seien $\lambda_0 \in \mathbb{R}$, $\lambda \in \mathbb{R}^n$ und $\mu \in \mathbb{R}^s$. Dann heißt

$$\tilde{H}(x, \lambda, \mu, u, t) := \lambda_0 \cdot L_0(x, u, t) + \lambda^T \cdot f(x, u, t) + \mu^T \cdot c(x, u) \quad (3.62)$$

die erweiterte Hamilton-Funktion zum optimalen Steuerprozess $\overline{\text{OCP}}$.

Zusammengefaßt präsentieren sich die notwendigen Bedingungen beschränkter optimaler Steuerprozesse in ähnlicher Form wie im Pontryaginschen Minimumprinzip. Dazu definiert man den Funktionenraum $C^k(U)$ wie folgt:

$$C^k(U) = \{u : U \rightarrow \mathbb{R} \mid u \text{ ist k-mal stetig differenzierbar}\} \quad (3.63)$$

Als Randstück der k-ten Komponente $c_k(x, u)$ wird ein Teilintervall $[t_1, t_2] \subset [t_0, t_f]$ bezeichnet, für das $c_k(x(t), u(t)) = 0$ für alle $t \in [t_1, t_2]$ gilt. Verbindungspunkte heißen dabei die Punkte $t_1, t_2 \in \mathbb{R}$, wenn ein $\delta \in \mathbb{R}, \delta > 0$ mit $c_k(x(t_1 - \varepsilon), u(t_1 - \varepsilon)) < 0$ und $c_k(x(t_2 + \varepsilon), u(t_2 + \varepsilon)) < 0$ für alle $\varepsilon \in]0, \delta[$ existiert. Fallen dabei die Verbindungspunkte des Randstücks der k-ten Komponente $c_k(x, u)$ zusammen, dann nennt man t_1 Kontaktpunkt der k-ten Komponente.

Theorem 3.24 (Erweitertes Minimumprinzip).

$x^*(t)$ und $u^*(t)$ seien die optimale Lösung eines autonomen optimalen Steuerprozesses $\overline{\text{OCP}}$ mit einer Beschränkung der Ordnung p . Weiterhin seien f und J_0 C^{2p} -Funktionen und die beiden folgenden Bedingungen seien erfüllt:

- Die Gleichung $c^p(x, u) = 0$ ist eindeutig auflösbar nach $u = u(x)$ mit einer C^{p+1} -Funktion $u(x)$.
- Auf jedem Randstück $[t_1, t_2]$ gilt für die Randsteuerung $u(t)$:

$$\nabla_u(c^p)(x(t), u(t)) \neq 0, \quad t \in [t_1, t_2] \tag{3.64}$$

Außerdem gelte: $u^*(t) \in \text{int}(U)$ für $t \in]t_1, t_2[$. Dann existieren

- $\lambda_0 \geq 0, \lambda_0 \in \mathbb{R}$
- $\lambda : [0, t_f] \rightarrow \mathbb{R}^n$ eine stückweise stetige und stückweise stetig differenzierbare (adjungierte) Funktion
- $\sigma \in \mathbb{R}^r$
- $\mu : [0, t_f] \rightarrow \mathbb{R}$ stückweise stetige Multiplikatorfunktion
- $\nu(t_i) \geq 0, \nu(t_i) \in \mathbb{R}$ in jedem Verbindungs- oder Kontaktpunkt t_i

so dass die folgenden Aussagen gelten:

- An allen Stetigkeitsstellen t von u^* in $[0, t_f]$ sind die beiden Bedingungen erfüllt:

– Minimumbedingung

$$\tilde{H}(x^*, \lambda, \mu, u^*) = \min_{u \in U} \tilde{H}(x^*, \lambda, \mu, u) = \text{konstant} \tag{3.65}$$

– Adjungierte Differentialgleichungen und Transversalitätsbedingungen

$$\dot{\lambda}^T = -\nabla_x H(x^*, \lambda, \mu, u^*) \tag{3.66}$$

$$\lambda(t_f)^T = \lambda_0 \nabla_x \Phi(x^*(t_f)) + \sigma^T \nabla_x \psi(x^*(t_f)) \quad \text{für } c(x^*(t_f), u^*(t_f)) < 0 \tag{3.67}$$

- Für alle $t \in [0, t_f]$ gilt

$$\mu(t) \geq 0, \quad \mu(t)c(x^*, u^*) = 0 \quad (3.68)$$

- In jedem Verbindungs- oder Kontaktpunkt t_i gelten (im Falle reiner Zustandsbeschränkungen) die Sprungbedingungen

$$\lambda(t_i^+)^T = \lambda(t_i^-)^T - \nu^T(t_i) \nabla_x c(x^*(t_i)). \quad (3.69)$$

Bei gemischten Beschränkungen sind die adjungierten Variablen stetig in $[0, t_f]$, d.h. es ist $\nu(t_i) = 0$.

- Bei freier Endzeit gilt

$$\tilde{H}(x^*(t_f), \lambda(t_f), \mu(t_f), u^*(t_f)) = 0. \quad (3.70)$$

Ein Beweis dieses Satzes findet sich für eine gemischte Beschränkung in dem Buch von L.W. Neustadt [24] und im Fall von Zustandsbeschränkungen in der Arbeit von H. Maurer [17]. Dabei bezeichne $\text{int}(U)$ das Innere von U .

Die hinreichenden Optimalitätsbedingungen des beschränkten optimalen Steuerungsproblems lassen sich dabei wie folgt zusammenfassen. Für einen Beweis sei in der Literatur auf die Arbeit von H. Maurer und S. Pickenhain [18] verwiesen. Zur leichteren Lesbarkeit werden die Voraussetzungen des Satzes danach in kompakter Form dargestellt.

Satz 3.25 (Hinreichende Optimalitätsbedingungen).

Funktionen $\lambda : [0, t_f] \rightarrow \mathbb{R}^n$, $\mu : [0, t_f] \rightarrow \mathbb{R}$ und ein Multiplikator $\rho \in \mathbb{R}^r$ sollen derart existieren, so dass die notwendigen Optimalitätsbedingungen des Satzes 3.24 erfüllt sind mit $\lambda_0 = 1$ und zudem die Voraussetzungen (V.1) - (V.6) für den Steuerprozess $\overline{\text{OCP}}$ für ein Paar $(x^*(t), u^*(t))$ gelten. Dann ist $(x^*(t), u^*(t))$ eine optimale Lösung und es existieren Konstanten $\varepsilon > 0$ und $c > 0$, $\varepsilon, c \in \mathbb{R}$, so dass für alle zulässigen Funktionen $(x, u) \in B_\varepsilon(x^*, u^*)$ mit

$$B_\varepsilon(x^*, u^*) := \{(x, u) \in L^\infty([0, t_f], \mathbb{R}^{n+m}) \mid \|(x, u) - (x^*, u^*)\|_\infty \leq \varepsilon\} \quad (3.71)$$

die Ungleichung

$$J_0(x, u) \geq J_0(x^*, u^*) + c \left\{ \|(x, u) - (x^*, u^*)\|^2 + \|(x(0), x(t_f)) - (x^*(0), x^*(t_f))\|^2 \right\} \quad (3.72)$$

gilt.

Die Voraussetzung (V.1) - (V.6) sind dabei:

(V.1) Die Funktionen $L_0(\cdot, \cdot)$, $\Phi_0(\cdot)$, $f(\cdot, \cdot)$, $\psi(\cdot)$ und $c(\cdot, \cdot)$ seien zweimal Fréchet differenzierbar bezüglich aller Argumente. Die Fréchet Differenzierbarkeit einer Funktion F impliziert die Existenz eines linearen Operators $\Lambda : X \rightarrow Y$, so dass

$$F(x + h) = F(x) + \Lambda h + r(h) \quad (3.73)$$

mit $\|r(h)\|_Y \cdot \|h\|_X^{-1} \rightarrow 0$ gilt für $\|h\|_X \rightarrow 0$. $\|\cdot\|_Z$ bezeichnet dabei die Norm auf dem Raum Z .

(V.2) Es gibt eine lokal zulässige Lösung

$$(x^*, u^*) \in C^1(0, t_f; \mathbb{R}^n) \times C(0, t_f; \mathbb{R}^m). \quad (3.74)$$

Für $I := \{1, \dots, s\}$, $s \in \mathbb{N}$, und $\pi \in \mathbb{R}$ mit $\pi \geq 0$ sei folgende Nähebedingung zu einer Beschränkung eingeführt:

$$I_\pi(t) = \{i \in I \mid c_i(x^*(t), u^*(t)) \geq -\pi\}, \quad (3.75)$$

$$\iota_\pi(t) = \#I_\pi(t), \quad (3.76)$$

$$c_x^\pi(t) := \nabla_x c_i(x^*(t), u^*(t)), \quad i \in I_\pi(t), \quad (3.77)$$

$$c_u^\pi(t) := \nabla_u c_i(x^*(t), u^*(t)), \quad i \in I_\pi(t). \quad (3.78)$$

(V.3) Es existieren Konstanten $\pi, \beta \in \mathbb{R}$, $\pi > 0$ und $\beta > 0$, so dass

$$\| [c_u^\pi(t)]^T \| \geq \beta \|z\| \quad (3.79)$$

für alle $z \in \mathbb{R}^{\iota_\pi(t)}$ und fast alle $t \in [0, t_f]$ gilt.

(V.4) Für $v(t) \in \mathbb{R}^m$ und jedes $e \in \mathbb{R}^r$ habe das Randwertsystem

$$\dot{y}(t) = \tilde{A}(t)y(t) - \tilde{B}(t)v(t), \quad (3.80)$$

$$y(0) = 0, \quad \nabla_x \psi(x^*(t_f))y(t_f) = e \quad (3.81)$$

mit

$$f^*(t) = f(x^*(t), u^*(t)),$$

$$\tilde{A}(t) = \nabla_x f^*(t) - \nabla_u f^*(t) \nabla_u c^\pi(t)^T (\nabla_u c^\pi(t) \nabla_u c^\pi(t)^T)^{-1} \nabla_x c^\pi(t),$$

$$\tilde{B}(t) = \nabla_u f^*(t) \left[\text{Id} - \nabla_u c^\pi(t)^T (\nabla_u c^\pi(t) \nabla_u c^\pi(t)^T)^{-1} \nabla_u c^\pi(t) \right]$$

eine Lösung $v(\cdot)$.

Für die Bedingungen (V.5) und (V.6) werden folgende Abkürzungen eingeführt:

Für $\alpha \geq 0$, $\alpha \in \mathbb{R}$, sei $I_+^\alpha(t_f) := \{i \in I_0(t) \mid (\mu^*)_i(t) > \alpha\}$, womit für $i \in I_+^\alpha(t)$

$$\nabla_x \hat{c}^\alpha(t) := \nabla_x c_i(x^*(t), u^*(t)), \quad (3.82)$$

$$\nabla_u \hat{c}^\alpha(t) := \nabla_u c_i(x^*(t), u^*(t)) \quad (3.83)$$

definiert wird.

(V.5) Es gebe ein $\gamma \in \mathbb{R}$ mit $\gamma > 0$, so dass für

$$\tilde{H}^*(t) := \tilde{H}(x^*(t), \lambda(t), \mu(t), u^*(t)) \quad (3.84)$$

die Legendre-Clebsch-Bedingung

$$v^T \nabla_u^2 \tilde{H}^*(t)v \geq \gamma \|v\|^2, \quad (3.85)$$

für alle $v \in \mathbb{R}^m$ mit $\nabla_u \hat{c}^\alpha(t)v = 0$ gilt.

(V.6) Für $v(t) \in \mathbb{R}^m$ und $y(t) \in \mathbb{R}^n$, die die Bedingungen

$$\dot{y}(t) = \nabla_x f^*(t)y(t) + \nabla_u f^*(t)v(t), \quad y(0) = 0 \quad (3.86)$$

$$0 = \nabla_x \hat{c}^\alpha(t)y(t) + \nabla_u \hat{c}^\alpha(t)v(t) \quad (3.87)$$

erfüllen, gilt

$$\int_0^{t_f} \begin{pmatrix} y(t) \\ v(t) \end{pmatrix}^T \begin{pmatrix} \nabla_{xx}^2 \tilde{H}^*(t) & \nabla_{xu}^2 \tilde{H}^*(t) \\ \nabla_{ux}^2 \tilde{H}^*(t) & \nabla_{uu}^2 \tilde{H}^*(t) \end{pmatrix} \begin{pmatrix} y(t) \\ v(t) \end{pmatrix} dt + y(t_f)^T \Gamma y(t_f) > 0 \quad (3.88)$$

und zudem

$$\nabla_x \psi(x^*(t_f))y(t_f) = 0 \quad (3.89)$$

wobei $\Gamma := \nabla_x^2 \left(\Phi(x^*(t_f)) + \rho^{*T} \psi(x^*(t)) \right)$ ist.

Nun sei die Euler-Diskretisierung des optimalen Steuerungsproblems $\overline{\mathbf{OCP}}$ angegeben. Dafür beschreibe $h = \frac{t_f}{N_t - 1}$, $N_t \in \mathbb{N}$, für $t_1 = 0$ den Gitterabstand, der als Parameter der Approximation behandelt werden soll. Somit erhält man das Gitter

$$t_{i+1} = ih, \quad i = 0, \dots, N_t - 1. \quad (3.90)$$

Der Raum $L_h^\infty(0, t_f; \mathbb{R}^m)$ sei für festes h auf $[0, t_f]$ definiert als der Raum der stückweise konstanten Funktionen v ,

$$v(t) = v(t_i) \quad \text{für } t \in [t_i, t_{i+1}), \quad i = 1, \dots, N_t - 1. \quad (3.91)$$

Die zugehörige Norm ist

$$\|v\|_\infty = \max\{|v(t_i)| : i = 1, \dots, N_t - 1\}. \quad (3.92)$$

Mit $W_h^{1,\infty}(0, t_f; \mathbb{R}^n)$ wird zudem der Raum der stückweise linearen Funktionen

$$y(t) = y(t_i) + (t - t_i) \frac{y(t_{i+1}) - y(t_i)}{h} \quad \text{für } t \in [t_i, t_{i+1}], \quad i = 1, \dots, N_t - 1 \quad (3.93)$$

eingeführt und mit der Norm

$$\|y\|_{1,\infty} = \max \left\{ \|y(t_0)\|, \left\| \frac{y(t_{i+1}) - y(t_i)}{h} \right\| : i = 1, \dots, N_t - 1 \right\} \quad (3.94)$$

versehen. Die Euler-Diskretisierung lässt sich dann wie folgt darstellen und wird mit $\overline{\mathbf{OCP}}_h$ bezeichnet:

$$\begin{aligned} \text{Minimiere } J_0(x_h, u_h) &= \Phi_0(x_h(t_f)) + h \sum_{i=0}^{N_t-1} L_0(x_h(t_i), u_h(t_i)) \\ \text{unter } -x_h(t_{i+1}) + x_h(t_i) + hf(x_h(t_i), u_h(t_i)) &= 0, \quad i = 1, \dots, N_t - 1 \\ x - x_h(t_1) &= 0 \\ \phi(x_h(t_{N_t})) &= 0 \\ c(x_h(t), u_h(t)) &\leq 0, \quad i = 1, \dots, N_t \end{aligned}$$

Da $\overline{\text{OCP}}_h$ ein Optimierungsproblem darstellt sei zunächst auf die Lösungstheorie dieser Probleme verwiesen, die im anschließenden Abschnitt genauer behandelt wird. Danach lässt sich die Lagrange-Funktion mit Hilfe der Multiplikatoren $\lambda_h \in L_h^\infty(0, t_f; \mathbb{R}^n)$, $\vartheta_h \in \mathbb{R}^n$, $\sigma_h \in \mathbb{R}^r$ und $\mu_h \in L_h^\infty(0, t_f; \mathbb{R}^s)$ angeben als Funktion

$$L : W_h^{1,\infty}(0, t_f; \mathbb{R}^n) \times L_h^\infty(0, t_f; \mathbb{R}^m) \times L_h^\infty(0, t_f; \mathbb{R}^n) \times \mathbb{R}^n \times \mathbb{R}^r \times L_h^\infty(0, t_f; \mathbb{R}^s) \rightarrow \mathbb{R}, \quad (3.95)$$

$$\begin{aligned} L(x_h, u_h, \lambda_h, \vartheta_h, \sigma_h, \mu_h) &= \Phi(x_h(t_{N_t})) + h \sum_{i=0}^{N_t-1} L_0(x_h(t_i), u_h(t_i)) \\ &+ \sum_{i=1}^{N_t-1} \lambda_h(t_{i+1})^T (-x_h(t_{i+1}) + x_h(t_i) + hf(x_h(t_i), u_h(t_i))) \\ &+ \vartheta_h^T (x_h - x_h(t_1)) + \sigma_h^T \psi(x_h(t_{N_t})) \\ &+ h \sum_{i=1}^{N_t} \mu_h(t_i)^T c(x_h(t_i), u_h(t_i)). \end{aligned} \quad (3.96)$$

Die Auswertung der KKT-Bedingungen, auf die bei der numerischen Lösung optimaler Steuerprozesse explizit eingegangen wird, ergibt folgende Resultate:

$$\frac{\lambda_h(t_{i+1}) - \lambda_h(t_i)}{h} + \nabla_x \tilde{H}(x_h(t_i), u_h(t_i), \lambda_h(t_{i+1}), \mu_h(t_i)) = 0, \quad i = 1, \dots, N_t - 1 \quad (3.97)$$

$$\lambda_h(t_1) - \vartheta_h = 0 \quad (3.98)$$

$$-\lambda_h(t_{N_t}) + \nabla_x (\Phi(x_h(t_{N_t})) + \sigma_h^T \psi(x_h(t_{N_t}))) = 0 \quad (3.99)$$

$$\nabla_u \tilde{H}(x_h(t_i), u_h(t_i), \lambda_h(t_{i+1}), \mu_h(t_i)) = 0, \quad i = 1, \dots, N_t - 1 \quad (3.100)$$

Damit kann das Hauptergebnis dieses Abschnittes formuliert werden:

Satz 3.26 (Konvergenz der Euler-Diskretisierung).

Es seien die Voraussetzungen (V.1) - (V.6) erfüllt. Dann existiert ein $\eta \in \mathbb{R}$, $\eta > 0$, so dass es für jede Schrittweite $h < \eta$ einen lokal eindeutigen KKT-Punkt $(x_h, u_h, \lambda_h, \vartheta_h, \sigma_h, \mu_h)$ des diskretisierten optimalen Steuerungsproblems $\overline{\text{OCP}}_h$ gibt und die Aussagen

$$\|x_h - x^*\|_{1,\infty} \leq l'|h| \quad (3.101)$$

$$\|u_h - u^*\|_\infty \leq l'|h| \quad (3.102)$$

$$\|\lambda_h - \lambda^*\|_{1,\infty} \leq l'|h| \quad (3.103)$$

$$\|\vartheta_h - \vartheta^*\| \leq l'|h| \quad (3.104)$$

$$\|\sigma_h - \sigma^*\| \leq l'|h| \quad (3.105)$$

$$\|\mu_h - \mu^*\|_\infty \leq l'|h| \quad (3.106)$$

für eine von h unabhängige Konstante $l' > 0$ gelten.

Für den Beweis sei auf die Arbeit von K.Malanowski, Ch.Büskens und H. Maurer [16] verwiesen.

Bemerkung 3.27. *Auffallend ist dabei, dass das Ergebnis unabhängig ist von Nichtdifferenzierbarkeitsstellen.*

Bemerkung 3.28. *Um eine Verbesserung bei der Zustandsapproximation zu erreichen lassen sich andere Verfahren wie etwa Runge-Kutta-Verfahren anwenden. Die Tatsache, dass diese jedoch interpolierte Werte der Steuerung an ihren Stützstellen benötigen, stellt für den vorliegenden Fall keine Einschränkung dar, da innerhalb eines zu betrachtenden Intervalls auf Grund der Wahl der Diskretisierung nur konstante Funktionen vorliegen.*

Bemerkung 3.29. *Die Handhabung steifer Differentialgleichungen erfordert die Verwendung impliziter Löser oder Verfahren hoher Ordnung bei kleinen Schrittweiten. Um diese Genauigkeitsproblematik zu umgehen kann auch die bereits erwähnte Implementierung einer Schrittweitensteuerung verwendet werden.*

3.6 Numerische Lösung optimaler Steuerprozesse

Das bekannteste Hilfsmittel zur analytischen Lösung differenzierbarer nichtlinearer Optimierungsprobleme sind die sogenannten Karush-Kuhn-Tucker Bedingungen. Diese Probleme haben dabei die Form **NLP**

$$\text{Minimiere } F(z) \quad (3.107)$$

$$\text{unter } G_i(z) = 0, \quad i = 1, \dots, N_E \quad (3.108)$$

$$G_i(z) \leq 0, \quad i = N_E + 1, \dots, N_E + N_I \quad (3.109)$$

mit $z \in \mathbb{R}^{N_z}$, $N_E, N_I \in \mathbb{N}$, $F : \mathbb{R}^{N_z} \rightarrow \mathbb{R}$ und $G : \mathbb{R}^{N_z} \rightarrow \mathbb{R}^{N_G}$, $N_G = N_E + N_I$. Hierbei wird die Funktion F als Zielfunktion und die Vektorfunktion G als Nebenbedingung bezeichnet. Für dieses Problem definiert man die zulässige Menge wie folgt:

Definition 3.30 (Zulässige Menge).

Die Menge

$$S := \{z \in \mathbb{R}^{N_z} \mid G_i(z) = 0, \quad i = 1, \dots, N_E, \\ G_i(z) \leq 0, \quad i = N_E + 1, \dots, N_G\} \quad (3.110)$$

wird Menge der zulässigen Punkte oder zulässige Menge genannt. $z \in S$ heißt dabei zulässiger Punkt.

Bemerkung 3.31. *Allerdings stellt schon allein die Größe des durch die Diskretisierung entstandenen Optimierungsproblems ein derartiges Hindernis dar, dass nur mehr eine numerische Lösung in Frage kommt.*

Jedoch beruhen auch die meisten numerischen Lösungsverfahren für nichtlineare Optimierungsprobleme wieder auf der Auswertung der Karush-Kuhn-Tucker Bedingungen, die sich wie folgt darstellen:

Satz 3.32 (Karush-Kuhn-Tucker Bedingungen).

Sei F differenzierbar und G stetig differenzierbar bezüglich z in einer Umgebung N der Minimalstelle $z^* \in \mathbb{R}^{N_z}$ des nichtlinearen Optimierungsproblems **NLP**, dessen Lagrange-Funktion $L : \mathbb{R}^{N_z} \times \mathbb{R}^{N_G} \rightarrow \mathbb{R}$ definiert ist durch

$$L(z, \eta) := F(z) + \eta^T G(z) \tag{3.111}$$

mit $\eta \in \mathbb{R}^{N_G}$. η wird dabei auch Lagrange-Multiplikator genannt. Weiter sei

$$I(z^*) = \{i \in \{N_E + 1, \dots, N_E + N_I\} \mid G_i(z^*) = 0\}, \tag{3.112}$$

$$J(z^*) = I(z^*) \cup \{1, \dots, N_E\}. \tag{3.113}$$

(1) Ist z^* regulär, d.h.

- $\nabla_z G_1(z^*), \dots, \nabla_z G_{N_E}(z^*)$ sind linear unabhängig,
- $\exists v \in \mathbb{R}^{N_z} \setminus \{0\}$ mit

$$\nabla_z G_i(z^*)v < 0, \quad i \in I(z^*), \tag{3.114}$$

$$\nabla_z G_i(z^*)v = 0, \quad i = 1, \dots, N_E, \tag{3.115}$$

so gibt es $\eta \in \mathbb{R}^{N_G}$ mit

$$\nabla_z L(z^*, \eta) = \eta_0 \nabla_z F(z^*) + \eta^T \nabla_z G(z^*) = 0, \tag{3.116}$$

$$\eta_i \geq 0, \quad i \in I(z^*), \tag{3.117}$$

$$\eta_i = 0, \quad i \notin J(z^*). \tag{3.118}$$

(2) Ist z^* normal (oder streng regulär), d.h. $\nabla_z G_i(z^*)$ sind linear unabhängig $\forall i \in J(z^*)$, so gilt (1) und der Lagrange-Multiplikator η ist eindeutig bestimmt.

Der Beweis dieses Satzes kann in den Büchern von R.Fletcher [5], Kapitel 9.2, oder P.E.Gill, W.Murray und M.H.Wright [6], Kapitel 3.4, nachgelesen werden.

Als kritischen oder auch KKT-Punkt bezeichnet man einen zulässigen Punkt $z^* \in \mathbb{R}^{N_z}$, der zusammen mit einem Multiplikator $\eta \in \mathbb{R}^{N_G}$ die KKT-Bedingungen erfüllt. Für Optimierungsprobleme lassen sich zudem hinreichende Bedingungen angeben, deren Beweis in Kapitel 9.3 des Buchs von R.Fletcher [5] oder in Kapitel 3.4 des Buchs von P.E.Gill, W.Murray und M.H.Wright [6] enthalten ist.

Satz 3.33 (Hinreichende Optimalitätsbedingungen).

Es sei $z^* \in \mathbb{R}^{N_z}$ ein zulässiger Punkt von **NLP** und weiter sei z^* normal. F sei zweimal differenzierbar bezüglich z und G sei zweimal stetig differenzierbar bezüglich z in einer Umgebung $N(z^*) \subset \mathbb{R}^{N_z}$. Der Kegel C sei definiert durch

$$C := \{v \in \mathbb{R}^{N_z} \mid \begin{array}{l} \nabla_z G_i(z^*)v \leq 0, \quad i \in I(z^*), \text{ falls } \eta_i = 0, \\ \nabla_z G_i(z^*)v = 0, \quad i \in I(z^*), \text{ falls } \eta_i > 0, \\ \nabla_z G_i(z^*)v = 0, \quad i = 1, \dots, N_E \end{array}\} \quad (3.119)$$

(1) Notwendige Bedingung zweiter Ordnung

Ist z^* eine lokale Minimalstelle von **NLP**, so gibt es eindeutig bestimmte Zahlen $\eta_i : \eta_i \in \mathbb{R}, i \in \{1, \dots, N_G\}$ mit

$$\eta_i \geq 0, \quad i \in I(z^*) \text{ und } \eta_i = 0, \quad i \notin J(z^*) \quad (3.120)$$

$$\nabla_z L(z^*, \eta) = 0 \quad (3.121)$$

$$v^T \nabla_z^2 L(z^*, \eta)v \geq 0, \quad \forall v \in C \setminus \{0\} \quad (3.122)$$

(2) Hinreichende Bedingungen zweiter Ordnung

Es gebe Zahlen $\eta_i \in \mathbb{R}, i \in \{1, \dots, N_G\}$ mit

$$\eta_i \geq 0, \quad i \in I(z^*) \text{ und } \eta_i = 0, \quad i \notin J(z^*) \quad (3.123)$$

$$\nabla_z L(z^*, \eta) = 0 \quad (3.124)$$

$$v^T \nabla_z^2 L(z^*, \eta)v > 0, \quad \forall v \in C \setminus \{0\} \quad (3.125)$$

Dann gibt es $\epsilon > 0$ und eine Konstante $c > 0, c, \epsilon \in \mathbb{R}$ mit

$$F(z) \geq F(z^*) + c\|z - z^*\|^2 \text{ für } z \in C, \|z - z^*\| \leq \epsilon. \quad (3.126)$$

(3) Strenge hinreichende Bedingung zweiter Ordnung

Es gebe Zahlen $\eta_i \in \mathbb{R}, i \in \{1, \dots, N_G\}$, so dass die hinreichenden Bedingungen zweiter Ordnung mit $\eta_i > 0 \forall i \in I(z^*)$ erfüllt sind. Dann gilt die Aussage ebenso und der Kegel C vereinfacht sich zu

$$C = \text{Kern}(\nabla_z G_i(\bar{z}) : i \in J(z^*)). \quad (3.127)$$

Bemerkung 3.34. Die notwendigen Optimalitätsbedingungen unterscheiden sich lediglich durch die bei den hinreichenden Bedingungen geforderte positive Definitheit in (3.125) von diesen.

Bemerkung 3.35. Die hinreichenden Bedingungen zweiter Ordnung erlauben Einblicke in die Sensitivität der optimalen Lösung bei Störungen.

Da in der Praxis besonders bei hochdimensionalen nichtlinearen Optimierungsproblemen die sogenannten SQP-Verfahren sich als stabil und effektiv erwiesen haben und zur numerischen Lösung des modellprädiktiven Problems die Routine 'E04UCF' der Nag benutzt wird, die im Wesentlichen auf den Arbeiten von P.E.Gill, W.Murray und M.H.Wright [6] basiert, soll hier im Kurzen auf die Funktionsweise dieses Verfahrens eingegangen werden.

Annahme 3.36. *Es wird dabei vorausgesetzt, dass die Funktionen F und G hinreichend oft differenzierbar in einer ausreichend großen Umgebung $N(z^*)$ der lokalen Minimalstelle $z^* \in \mathbb{R}^{N_z}$ sind.*

Eine weitere Vereinfachung wird durch die notwendigen (und hinreichenden) Bedingungen möglich gemacht, denn auf Grund der Aussage $\eta_i = 0, i \notin J(z)$ müssen nichtaktive Nebenbedingungen bei der Auswertung nicht beachtet werden. Daher kann man sich ausschließlich auf Probleme mit Gleichungsnebenbedingungen beschränken, wobei der Vektor der aktiven Nebenbedingungen

$$G^a := (G_i)_{i \in J(z)}, \quad \eta_a \in \mathbb{R}^{N_a}, \quad N_a := \#J(z) \quad (3.128)$$

für zulässige Punkte $z \in S$ dem Rechnung trägt.

Die prinzipielle Idee aber liegt in der Generierung und Lösung quadratischer Unterprobleme, sogenannten **QP**-Problemen, die lokal eine Näherung des Ausgangsproblems **NLP** darstellen.

Definition 3.37 (Quadratisches Unterproblem).

Mit $B^{[k]}$ sei eine geeignete Approximation der Hesse-Matrix der Zielfunktion gegeben. Weiter sei $z^{[k]} \in \mathbb{R}^{N_z}$ eine Näherung der Optimallösung $z^* \in \mathbb{R}^{N_z}$. Dann nennt man

$$\begin{array}{ll} \text{Minimiere} & \nabla_z F(z^{[k]})p + \frac{1}{2}p^T B^{[k]}p \\ p \in \mathbb{R}^{N_z} & \end{array} \quad (3.129)$$

$$\text{unter} \quad G^a(z^{[k]}) + \nabla_z G^a(z^{[k]})p = 0 \quad (3.130)$$

quadratisches Unterproblem des nichtlinearen Optimierungsproblems **NLP** und $p^{[k]} \in \mathbb{R}^{N_z}$ Suchrichtung der SQP-Iteration.

Bemerkung 3.38. *Die Bedingung, dass $z^{[k]} \in \mathbb{R}^{N_z}$ eine Näherung der Optimallösung $z^* \in \mathbb{R}^{N_z}$ sein soll, ist in der Praxis dabei sehr problematisch. Dies rührt daher, dass die Optimierungsvariable zumindest aus dem Anfangszustand und der Steuerung besteht und eine Startlösung der Steuerung für das vorliegende Problem nur sehr schwer zu erhalten ist.*

SQP-Verfahren lassen sich in Haupt- und Nebeniterationen aufteilen, die sequentiell abgearbeitet werden. Dabei wird in der Hauptiteration mittels Newton- oder Quasi-Newton-Verfahren eine Folge $(z^{[k]})$ berechnet durch

$$z^{[k+1]} = z^{[k]} + \alpha^{[k]}p^{[k]}. \quad (3.131)$$

Darin beschreibt $z^{[k]}$ den aktuellen Zustand der Näherung, $\alpha^{[k]}$ die Schrittweite und $p^{[k]}$ die Suchrichtung, die mit der Nebeniteration als Lösung des QP-Unterproblems bestimmt wird. Für genauere Ausführungen zu Newton- und Quasi-Newton-Verfahren sei an dieser Stelle auf Kapitel 3 der Arbeit von Ch.Büskens [2] verwiesen.

Bemerkung 3.39. *Da die Berechnung der Hesse-Matrix sehr aufwendig ist und die Verwendung einer einmal berechneten Hesse-Matrix über mehrere Iterationsschritte zu großen Fehlern führen kann verwendet man eine sogenannte Quasi-Newton-Korrektur mittels der BFGS-Formel, die explizit im abschließenden Algorithmus aufgeführt ist.*

Zur Ermittlung der Schrittweite $\alpha^{[k]}$ wird eine sogenannte Bewertungsfunktion, auch Merit-Funktion genannt, eingeführt. Diese besteht immer aus der Lagrange-Funktion und einem additiven Ausdruck, wobei die Bewertungsfunktion nicht zwingend differenzierbar sein muss. In der Literatur finden sich sowohl differenzierbare wie auch nichtdifferenzierbare Funktionen, die hierfür benutzt werden können. In 'E04UCF' wird dabei folgende Bewertungsfunktion angewandt:

$$\tilde{L}(z, \eta, \rho) = L(z, \eta) + \frac{1}{2} \sum_{j=1}^{N_a} \rho_j G_j^2(z) \quad (3.132)$$

Bemerkung 3.40. *Durch die Wahl der Schrittweite $\alpha^{[k]}$ mittels*

$$\tilde{L}(z^{[k+1]}, \eta^{[k]}, \rho) \approx \min_{0 < \tilde{\alpha}^{[k]} \leq 1} \tilde{L}(z^{[k]} + \tilde{\alpha}^{[k]} p^{[k]}, \eta^{[k]}, \rho) \quad (3.133)$$

reduziert die Bewertungsfunktion allerdings die natürliche Schrittweite des Newton-Verfahrens $\alpha^{[k]} = 1$, jedoch ist zu erwarten, dass $\alpha^{[k]}$ in der Nähe der optimalen Lösung näherungsweise 1 beträgt. Außerdem muss beachtet werden, dass beim Abstieg entlang der Suchrichtung $p^{[k]}$ Beschränkungen aktiv werden können.

Somit ergibt sich folgender Algorithmus, die sogenannte Strategie der aktiven Mengen:

1. Wähle zulässiges $z^{[0]} \in \mathbb{R}^{N_z}$, bestimme die Indexmenge der aktiven Nebenbedingungen $\mathcal{A}^{(0)}$ und setze $k \in \mathbb{R}$ auf $k = 0$.
2. Berechne Suchrichtung $p^{[k]}$ durch Lösen des QP-Unterproblems. Ist $p^{[k]} = 0$ die Lösung, dann gehe zu 3., sonst gehe zu 4.
3. Da $p^{[k]} = 0$ ist, löst $z^{[k]}$ das Problem

$$\text{Minimiere} \quad \nabla_z F(z^{[k]}) z^{[k]} + \frac{1}{2} (z^{[k]})^T B^{[k]} z^{[k]} \quad (3.134)$$

$$\text{unter} \quad G_i(z^{[k]}) = 0, \quad i \in \mathcal{A}^{[k]}. \quad (3.135)$$

Berechne den zugehörigen Lagrange-Multiplikator $\eta_i^{[k]}$, $i \in \mathcal{A}^{[k]}$ durch

$$-\nabla_z G^{[k]} \eta^{[k]} = B^{[k]} z^{[k]} + \nabla_z f(z^{[k]}), \quad (3.136)$$

wobei $G^{[k]} = [G_i(z^{[k]})]$, $i \in \mathcal{A}^{[k]}$ Vektor der aktiven Nebenbedingungen ist. Wähle j , so dass $\eta_j^{[k]} = \min_{i \in \mathcal{A}^{[k]} \cap (J \setminus I)} \eta_i^{[k]}$.

Wenn $\eta_j^{[k]} \geq 0$, dann beende mit $z^* = z^{[k]}$.

Sonst setze $A^{[k]} = A^{[k]} \setminus \{j\}$ und gehe zu 2.

4. Da $p^{[k]} \neq 0$ setze $z^{[k+1]} = z^{[k]} + \alpha^{[k]} p^{[k]}$ mit

$$\alpha^{[k]} = \min(\tilde{\alpha}^{[k]}, \bar{\alpha}^{[k]}), \quad (3.137)$$

$$\text{wobei } \bar{\alpha}^{[k]} = \min_{i: \nabla_z G_i(z^{[k]}) p^{[k]} < 0} \frac{G_i(z^{[k]})}{\nabla_z G_i(z^{[k]}) p^{[k]}} \quad (3.138)$$

und $\tilde{\alpha}^{[k]}$ aus (3.133). Ist $\alpha^{[k]} < \tilde{\alpha}^{[k]}$, dann setze $\mathcal{A}^{[k+1]} = \mathcal{A}^{[k]} \cup p$, wobei $p \in \{i : \nabla_z G_i(z^{[k]}) p^{[k]} < 0\}$ Minimum für $\bar{\alpha}^{[k]}$, sonst setze $\mathcal{A}^{[k+1]} = \mathcal{A}^{[k]}$.

Bestimme Approximation der Hessematrix neu mit BFGS-Formel:

$$B^{[k+1]} = B^{[k]} + \frac{\gamma^{[k]}(\gamma^{[k]})^T}{(s^{[k]})^T \gamma^{[k]}} - \frac{B^{[k]} s^{[k]} (s^{[k]})^T B^{[k]}}{(s^{[k]})^T B^{[k]} s^{[k]}} \quad (3.139)$$

$$\text{mit } s^{[k]} = z^{[k+1]} - z^{[k]}, \quad (3.140)$$

$$\gamma^{[k]} = \nabla_z L(z^{[k+1]}, \eta^{[k]}) - \nabla_z L(z^{[k]}, \eta^{[k]}) \quad (3.141)$$

Setze $k = k + 1$ und gehe zu 2.

Bemerkung 3.41. *Nachteilig an dem Verfahren ist offenkundig die Verwendung einer lediglich approximativen Hessematrix. Würde an dieser Stelle die exakte Hessematrix verwendet werden, so entspräche es dem Newton-Verfahren und hätte die Konvergenzordnung 2. In der dargelegten Form als Quasi-Newton-Verfahren reduziert sich die Konvergenzordnung auf 1, jedoch kann für die Quasi-Newton-Korrektur in Schritt 4 superlineare Konvergenz nachgewiesen werden. Ein entsprechender Konvergenzbeweis findet sich in der Arbeit von K.Schittkowski [26]. Zudem garantiert die BFGS-Formel die positive Definitheit der approximierten Hessematrix, falls die zu Beginn des Algorithmus berechnete Hessematrix positiv definit ist.*

Kapitel 4

Stabilität

Den modellprädiktiven Regelungsverfahren liegen verschiedene Ideen zugrunde, eine davon ist die Optimalität der Regelung, auf die im nächsten Kapitel eingegangen werden soll, die andere ist die Stabilität des geregelten Systems. Eine der frühen Beobachtungen (Kalman, 1960) war dabei, dass Optimalität die Stabilität nicht impliziert, aber dass unter Stabilisierbarkeits- und Detektierbarkeitsbedingungen eine optimale Steuerung auf einem unendlichen Horizont stabilisierend ist, wobei eine entsprechende Lyapunov-Funktion zur Ermöglichung der Stabilität ein Zielfunktional ist, das mit dem Optimalsteuerungsproblem mit unendlichen Horizont assoziiert ist. Frühere Verfahren stellten Stabilität nicht automatisch sicher, sondern erreichten dies durch Tuning, indem etwa nur stabile Systeme untersucht oder der Zeithorizont entsprechend groß gewählt wurde. Eine ausführliche Zusammenfassung dazu findet sich in der Arbeit von D.Q.Mayne, J.B.Rawling, C.V.Rao und P.O.M.Scokaert [19]. Der Aufbau dieses Kapitels richtet sich nach den Arbeiten von G.Grimm, M.J.Messina, A.R.Teel, S.Tuna [7], D.Nešić, A.R.Teel, P.V.Kokotović [20, 21] und D.Nešić und L.Grüne [23].

4.1 Stabilitätseigenschaften zeitdiskreter modellprädiktiver Regelungsverfahren

Im Folgenden werden einige der Ergebnisse aus der Arbeit von G.Grimm, M.J.Messina, A.R.Teel und S.Tuna [7] zusammengefaßt. Um diese auf das vorliegende Problem anwenden zu können wird ein zeitdiskretes Hilfssystem eingeführt. Dabei hat das exakte zeitdiskrete Modell des kontinuierlichen sampled-data Systems

$$\dot{x}(t) = f(x(t), u(x(t))), \quad x(0) = x \quad (4.1)$$

$$\dot{\xi}(t) = f(\xi(t), v(t)), \quad \xi(0) = \xi, \quad v(t) = v_k \quad \text{für } t \in [t_k, t_{k+1}) \quad (4.2)$$

die Form

$$\xi^+ = G(\xi, u) := \xi(T, \xi, u), \quad \xi(0) = \xi \quad (4.3)$$

$$x^+ = H(x) := x(T, x), \quad x(0) = x. \quad (4.4)$$

Exakt bedeutet, dass die Lösungen dieses Systems,

$$\xi_k = \xi(k, \xi, v_{[0,k-1]}) \quad (4.5)$$

$$x_k = x(k, x), \quad (4.6)$$

tatsächlich exakt an den diskreten Punkten berechnet werden, wobei man die Steuerung aus

$$\hat{u}_{[0,M-1]} = \arg \inf_{v_{[0,M-1]}} \sum_{i=0}^{M-1} Q(\xi_i, x_i, v_i) + F(\xi_M, x_M) \quad (4.7)$$

$$v_i \in U \quad \forall i \in \{0, 1, \dots, M-1\} \quad (4.8)$$

$$\text{mit } Q(\xi, x, u) := \int_0^T l(\xi(s, \xi, u) - x(s, x), u) ds \quad (4.9)$$

bestimmt und so das exakte zeitdiskrete Modell des closed-loop sampled-data Systems mit $u_M(\xi, x)$ aus (3.19) erhält:

$$\xi^+ = G(\xi, u_M(\xi, x)) \quad (4.10)$$

$$x^+ = H(x) \quad (4.11)$$

4.1.1 Stabilität des zeitdiskreten Verfahrens

Es werden nun die entsprechenden Definitionen und Ergebnisse aus [7] angepasst, so dass sie zur Stabilitätsanalyse des exakten zeitdiskreten Modells des closed-loop sampled-data Systems (4.10), (4.11) verwendet werden können. Besonders zu bemerken ist dabei, dass weder der Mayer-Anteil F noch der Lagrange-Anteil l des Zielfunktionals positiv definite Zustandsfunktionen sein müssen, wie dies im zu betrachtenden Problem der Fall ist.

Definition 4.1 (Detektierbarkeit).

Gegeben sei das System (4.3), (4.4) und eine Funktion $Q = Q(\xi, x, u)$. Dann nennt man das System (4.3), (4.4) detektierbar durch Q mit $(\bar{\alpha}_W, \alpha_W, \gamma_W)$, wenn $\alpha_W, \gamma_W \in \mathcal{K}_\infty$ und $\bar{\alpha}_W \in \mathcal{G}$ und eine stetige Funktion $W : \mathbb{R}^{2n} \rightarrow \mathbb{R}_0^+$ existieren, so dass die Abschätzungen

$$W(\xi, x) \leq \bar{\alpha}_W(\|(\xi, x)\|) \quad (4.12)$$

$$W(G(\xi, u), H(x)) - W(\xi, x) \leq -\alpha_W(\|(\xi, x)\|) + \gamma_W(Q(\xi, x, u)) \quad (4.13)$$

für alle $(\xi, x) \in \mathbb{R}^{2n}$ und alle $u \in \mathcal{U}$ gelten.

Definition 4.2 (Lyapunov-Kontroll-Funktion).

Die Mayer-Anteil $F(\cdot, \cdot)$ des Zielfunktionals (3.12) heißt Lyapunov-Kontroll-Funktion des Systems (4.3), (4.4), wenn F zerlegt werden kann in

$$F(\xi, x) = F_N(\xi, x) = \Gamma(N) \cdot \bar{F}(\xi, x) \quad (4.14)$$

mit einer nicht-fallenden und unbeschränkten Funktion $\Gamma_{\geq 1} : \mathbb{Z}_{\geq 1} \rightarrow \mathbb{Z}_{\geq 1}$ und wenn zudem Funktionen $\underline{\alpha}_F, \bar{\alpha}_Q \in \mathcal{K}_{\infty}$ existieren, so dass

$$\underline{\alpha}_F(\|(\xi, x)\|) \leq \bar{F}(\xi, x) \quad (4.15)$$

gilt und für alle $(\xi, x) \in \mathbb{R}^{2n}$ ein $u \in \mathcal{U}$ existiert mit

$$\bar{F}(G(\xi, u), H(x)) - \bar{F}(\xi, x) \leq 0 \quad (4.16)$$

$$Q(\xi, x, u) \leq \bar{\alpha}_Q(\|(\xi, x)\|). \quad (4.17)$$

In [7] wurde bewiesen, dass die Stabilität auch aus den angegebenen Bedingungen gefolgert werden kann ohne das F eine Lyapunov-Kontroll-Funktion sein muss. Die dort gemachten Annahmen und die sich ergebende Aussage lassen sich wie folgt zusammenfassen:

Theorem 4.3 (Stabilität des zeitdiskreten modellprädiktiven Regelungsverfahrens).

Es gelten die folgenden Bedingungen:

- (i) F und Q aus (4.7), (4.9) seien stetig;
- (ii) $U \subset \mathbb{R}^m$ aus (4.8) ist beschränkt;
- (iii) Das exakte zeitdiskrete System (4.3), (4.4) ist detektierbar durch $Q(\xi, x, u)$ mit $(\bar{\alpha}_W, \alpha_W, \gamma_W)$;
- (iv) Die optimale Wertefunktion ist von der Gestalt, dass für $\bar{\alpha} \in \mathcal{K}_{\infty}$ gilt:

$$V_i(\xi, x) \leq \bar{\alpha}(\|(\xi, x)\|) \quad \forall i \geq 0 \text{ und } \forall (\xi, x) \in \mathbb{R}^{2n} \quad (4.18)$$

Dann existieren für alle $M \geq 2$ Funktionen $\underline{\alpha}_Y, \bar{\alpha}_Y, \alpha_Y \in \mathcal{K}_{\infty}$, $\beta_Y \in \mathcal{KL}$ und eine stetige Funktion $Y_M : \mathbb{R}^{2n} \rightarrow \mathbb{R}_0^+$, so dass für alle $(\xi, x) \in \mathbb{R}^{2n}$ die Ungleichungen

$$\underline{\alpha}_Y(\|(\xi, x)\|) \leq Y_M(\xi, x) \leq \bar{\alpha}_Y(\|(\xi, x)\|) \quad (4.19)$$

$$Y_M(G(\xi, u_M(\xi, x)), H(x)) - Y_M(\xi, x) \leq -\alpha_Y(\|(\xi, x)\|) + \beta_Y(\|(\xi, x)\|, M) \quad (4.20)$$

gelten. Ist zudem die Voraussetzung

- (v) Der Mayer-Anteil des Zielfunktionals $F(\cdot, \cdot)$ aus (4.7) ist eine Lyapunov-Kontroll-Funktion des Systems (4.3), (4.4) mit $(\underline{\alpha}_F, \bar{\alpha}_Q)$.

erfüllt, dann existiert eine stetige Funktion Y_M und eine Funktion $\hat{\beta}_Y \in \mathcal{KL}$ derart, so dass die Ungleichungen (4.19) und

$$Y_M(G(\xi, u_M(\xi, x)), H(x)) - Y_M(\xi, x) \leq -\alpha_Y(\|(\xi, x)\|) + \hat{\beta}_Y(\|(\xi, x)\|, \Gamma(M)) \quad (4.21)$$

für alle $(\xi, x) \in \mathbb{R}^{2n}$ gültig sind.

Bemerkung 4.4. In dem Theorem wurde die Voraussetzung (i) mit aufgenommen um garantieren zu können, dass die Funktion Y_M stetig ist. Im Fall von unstetig vorliegenden Funktionen F und Q gelten zwar die genannten Beschränkungen, jedoch kann bei Vorliegen einer stetigen Lyapunov-Funktion Y_M für das exakte zeitdiskrete closed-loop sampled-data System (4.10), (4.11) trotz der möglichen Unstetigkeit der Feedbacksteuerung aus (3.19) die Robustheit des Systems gezeigt werden. Für die entsprechende Theorie sei auf Kapitel 6.1 der Arbeit von C.M.Kellet [12] verwiesen.

Einige direkte Folgerungen aus Theorem 4.3 sollen hier ohne Beweis aufgenommen werden, wobei diese in [7] enthalten sind. Im Zusammenhang mit dem konkret vorliegenden Verfahren werden sie noch von Bedeutung sein.

Proposition 4.5. Es gelten die Bedingungen (i), (ii), (iii) und (iv) des Theorems 4.3. Dann existiert für jedes Paar positiver reeller Zahlen (Δ, δ) ein $M_1^* \in \mathbb{Z}_{\geq 1}$ und eine Funktion $\beta \in \mathcal{KL}$, so dass für alle $(\xi, x) \in B_\Delta$ die Lösungen des Systems (4.10), (4.11) die Ungleichung

$$\|(\xi_k, x_k)\| \leq \max\{\beta(\|(\xi, x)\|, k), \delta\} \quad \forall k \geq 0 \quad (4.22)$$

erfüllen. Gilt zudem die Bedingung (v) des Theorems 4.3, dann existiert für jedes Paar positiver reeller Zahlen (Δ, δ) ein $M_2^* \in \mathbb{Z}_{\geq 1}$, so dass für alle $(\xi, x) \in B_\Delta$ die Lösungen des Systems (4.10), (4.11) die Ungleichung (4.22) erfüllen.

Bemerkung 4.6. Dies bedeutet, dass Y_M eine Lyapunov-Funktion für das exakte zeitdiskrete closed-loop sampled-data System (4.10), (4.11) ist, wenn die Länge des Horizonts M entsprechend groß gewählt wird, so dass β_Y beziehungsweise $\hat{\beta}_Y$ hinreichend klein werden. Da $\beta_Y, \hat{\beta}_Y \in \mathcal{KL}$ sind ist dies für beschränktes Γ beziehungsweise für eine Lyapunov-Kontrollfunktion F und unbeschränktes Γ immer möglich.

Proposition 4.7. Es gelten die Bedingungen (i), (ii), (iii) und (iv) des Theorems 4.3. Zudem gelte für gegebenes $\Delta_l > 0$, $\Delta_l \in \mathbb{R}$, so dass für alle $s \in [0, \Delta_l]$ die Funktionen $\alpha_W, \bar{\alpha}_W, \gamma_W, \bar{\alpha}$ aus den Bedingungen (iii) und (iv) die Ungleichungen

$$\alpha_W(s) \geq a_W \cdot s^2 \quad (4.23)$$

$$\bar{\alpha}_W(s) \leq \bar{a}_W \cdot s^2 \quad (4.24)$$

$$\gamma_W(s) \leq s^2 \quad (4.25)$$

$$\bar{\alpha}(s) \leq \bar{a} \cdot s^2 \quad (4.26)$$

mit $a_W, \bar{a} \in \mathbb{R}^+$, $a_W \in \mathbb{R}_0^+$ erfüllen. Dann existiert eine Funktion $\beta \in \mathcal{KL}$ derart, dass für jedes $\Delta \in \mathbb{R}$, $\Delta > 0$ ein $M_1^* \in \mathbb{Z}_{\geq 1}$ existiert, so dass für alle $M \geq M_1^*$ und alle $(\xi, x) \in B_\Delta$ die Lösungen des Systems (4.10), (4.11) die Ungleichung

$$\|(\xi_k, x_k)\| \leq \beta(\|(\xi, x)\|, k) \quad \forall k \geq 0 \quad (4.27)$$

erfüllen. Gilt zudem die Bedingung (v) des Theorems 4.3 mit

$$\underline{\alpha}_F(s) \geq \underline{a}_F \cdot s^2 \quad (4.28)$$

$$\bar{\alpha}_Q(s) \leq \bar{a}_Q \cdot s^2 \quad (4.29)$$

für alle $s \in [0, \Delta_1]$ mit $\underline{a}_F, \bar{a}_Q \in \mathbb{R}^+$, dann existiert eine Funktion $\beta \in \mathcal{KL}$, so dass für alle $\Delta \in \mathbb{R}$, $\Delta > 0$ ein $M_2^* \in \mathbb{Z}_{\geq 1}$ existiert, für welches für alle $(\xi, x) \in B_\Delta$ die Lösungen des Systems (4.10), (4.11) die Ungleichung (4.27) erfüllen.

Proposition 4.8. *Es gelten global die Bedingungen der Proposition 4.7. Weiter seien alle Funktionen global quadratisch. Gilt entweder*

- $M > \frac{\bar{a}(\bar{a} + \bar{a}_W)}{\underline{a}_W} + 1$ oder
- F ist eine Lyapunov-Kontroll-Funktion des Systems (4.3), (4.4) mit

$$\underline{\alpha}_F(s) \geq \underline{a}_F \cdot s^2 \quad (4.30)$$

$$\bar{\alpha}_Q(s) \leq \bar{a}_Q \cdot s^2, \quad (4.31)$$

wobei $\underline{a}_F, \bar{a}_Q \in \mathbb{R}^+$ sind und für M gilt:

$$\Gamma(M) > \frac{\bar{a}_Q \bar{a}}{\underline{a}_F \underline{a}_W} \quad (4.32)$$

dann existieren Konstanten $K, \lambda \in \mathbb{R}$, $K \geq 1, \lambda > 0$, so dass die Ungleichung

$$\|(\xi_k, x_k)\| \leq K e^{-\lambda k} \|(\xi, x)\| \quad \forall k \geq 0 \quad (4.33)$$

für die Lösungen des Systems (4.10), (4.11) erfüllt ist.

4.1.2 Stabilitätszusammenhänge zwischen approximiertem und exaktem zeitdiskreten System

Im Normalfall liegt kein exaktes zeitdiskretes System der Form (4.3), (4.4) vor, woraus sich die Frage ergibt, ob ein berechnetes Feedback der Form (3.19) des approximierten zeitdiskreten Systems auch das exakte zeitdiskrete Problem stabilisiert.

Daher soll unter Bezug auf die Arbeit von D.Nešić, A.R.Teel und P.V.Kokotović [20] gezeigt werden, dass dies unter bestimmten Forderungen an die Steuerung, die das approximierte zeitdiskrete System stabilisiert für $T \in]0, T^*[$, $T, T^* \in \mathbb{R}$, für das exakte zeitdiskrete System mit $T \in]0, \hat{T}[$, $\hat{T} \in \mathbb{R}$, $\hat{T} \leq T^*$ gefolgert werden kann. Das exakte zeitdiskrete System sei dabei durch die Differenzgleichung

$$x(k+1) = F_T^e(x(k), u(k)) \quad (4.34)$$

dargestellt, die parametrisiert ist durch die sampled-data Rate T . In analoger Weise sei das approximierte System mittels

$$x(k+1) = F_T^a(x(k), u(k)), \quad (4.35)$$

gegeben, wobei beide global für alle kleinen T definiert sein sollen. Dazu beschränkt man die Lösungspaare (x_k, u_k) für eventuell auftretende endliche Entweichzeiten und die entsprechende Schranke darf für $T \rightarrow 0$ wachsen. Die Lösung des exakten zeitdiskreten Systems wird dabei mit $x^e(k, x)$ oder x_k^e , die des approximierten zeitdiskreten Systems mit $x^a(k, x)$ oder x_k^a bezeichnet und $u(k) = u_T(x(k))$ ist ein statischen Zustandsfeedback für ein festes $T > 0$.

In [20] wird die Aussage für zwei verschiedene Arten von Konsistenzen gezeigt, die Ein-Schritt- und die Mehr-Schritt-Konsistenz. Da sich die beiden Konsistenzen nicht gegenseitig implizieren seien hier beide erwähnt:

Definition 4.9 (Ein-Schritt-Konsistenz).

Falls für jede kompakte Menge $\mathcal{A} \subset \mathbb{R}^n$ eine Funktion $\rho \in \mathcal{K}_\infty$ existiert und es ein $T^* \in \mathbb{R}$, $T^* > 0$ gibt, so dass die Ungleichung

$$\|F_T^e(x, u_T(x)) - F_T^a(x, u_T(x))\| \leq T\rho(T) \quad (4.36)$$

für alle $x \in \mathcal{A}$ und $T \in]0, T^*[$ erfüllt ist, dann wird eine Familie (u_T, F_T^a) Ein-Schritt-konsistent mit (u_T, F_T^e) genannt.

Definition 4.10 (Mehr-Schritt-Konsistenz).

Eine Familie (u_T, F_T^a) bezeichnet man als Mehr-Schritt-konsistent mit (u_T, F_T^e) , wenn für jedes $\delta, L, \eta \in \mathbb{R}$, $\delta > 0$, $L > 0$, $\eta > 0$ und jede kompakte Menge $\mathcal{A} \subset \mathbb{R}^n$ eine Funktion $\alpha : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+ \cup \{\infty\}$ und ein $T^* \in \mathbb{R}$, $T^* > 0$ existieren, so dass

$$\{x, z \in \mathcal{A}, \|x - z\| \leq \delta\} \Rightarrow \|F_T^e(x, u_T(x)) - F_T^a(z, u_T(z))\| \leq \alpha(\delta, T) \quad (4.37)$$

und

$$k \leq \frac{L}{T} \Rightarrow \alpha^k(0, T) := \underbrace{\alpha(\dots \alpha(\alpha(0, T), T) \dots, T)}_{k\text{-mal}} \leq \eta \quad (4.38)$$

für alle $T \in]0, T^*[$ gilt.

Bemerkung 4.11. Der Unterschied in den Definitionen liegt also darin, dass bei Ein-Schritt-Konsistenz garantiert wird, dass der Fehler zwischen den Lösungen x_k^a , x_k^e ausgehend vom gleichen Anfangswert für einen Schritt relativ zur Schrittweite klein ist. Bei der Mehr-Schritt-Konsistenz dagegen wird garantiert, dass der Fehler zwischen den Lösung, wieder mit gleichem Anfangswert, klein ist über mehrere einzelne Schritte hinweg.

Theorem 4.12 (Stabilisierbarkeit des zeitdiskreten Systems).

Sei $\beta \in \mathcal{KL}$ und $N \subset \mathbb{R}^n$ eine beschränkte Umgebung des Ursprungs.

(1) Dann ist das System (4.34) für die Familie (u_T, F_T^e) semiglobal praktisch asymptotisch stabil, wenn

(1i) die Familie (u_T, F_T^a) Mehr-Schritt-konsistent ist mit der Familie (u_T, F_T^e)

(1ii) und ein $T^* \in \mathbb{R}$, $T^* > 0$ existiert, so dass das System (4.35) für die Familie (u_T, F_T^a) asymptotisch stabil ist für alle $T \in]0, T^*[$.

(2) Es gelten die Bedingungen

(2i) Die Familie (u_T, F_T^a) ist Ein-Schritt-konsistent mit (u_T, F_T^e) .

(2ii) Es existiert eine semiglobale praktische Familie von Lyapunov-Funktionen V_T für die Familie (u_T, F_T^a) und für jede kompakte Menge $\mathcal{A} \subset \mathbb{R}^n \setminus \{0\}$ gibt es $M, T^* \in \mathbb{R}$, $M > 0$, $T^* > 0$, so dass die lokale Lipschitz-Bedingung

$$|V_T(x) - V_T(z)| \leq M \|x - z\| \quad (4.39)$$

für alle $x, z \in \mathcal{A}$ und alle $T \in]0, T^*[$ gilt.

Dann ist das System (4.34) für die Familie (u_T, F_T^e) semiglobal praktisch asymptotisch stabil.

Beweis. Der Beweis wird für beide Teile getrennt gezeigt:

(1) Sei eine Konstante $R \in \mathbb{R}$, $R > 0$, eine Funktion $\beta \in \mathcal{KL}$ und eine beschränkte Umgebung N des Ursprungs gegeben. Zu zeigen ist hier, dass ein $\hat{T} \in \mathbb{R}$, $\hat{T} > 0$, existiert, für das für alle $T \in]0, \hat{T}[$, $x(0) \in N$, $k \in \mathbb{N}$, $k \geq 0$, die Abschätzung

$$\|x^e(k, x(0))\| \leq \beta(\|x(0)\|, kT) + R \quad (4.40)$$

gilt.

Sei $M = \sup_{\xi \in N} \|\xi\|$ und sei B ein abgeschlossener Ball mit Radius $\beta(M, 0)$. Zudem sei $\eta \in \mathbb{R}$, $\eta > 0$ derart gegeben, dass der abgeschlossene Ball mit Radius 2η in N enthalten ist und $\beta(2\eta, 0) < \frac{R}{2}$ gilt. Die Existenz eines solchen η ist gesichert, da N eine Umgebung des Ursprungs und $\beta(s, 0)$ eine \mathcal{K} -Funktion ist. Aus (1ii) folgt für die Ungleichung

$$\|x(k, x(0))\| \leq \beta(\|x(0)\|, kT) + R \quad \forall x(0) \in N, k \geq 0 \quad (4.41)$$

für $k = 0$, dass $\beta(s, 0) \geq s$ und dass $2\eta < \frac{R}{2}$. Sei $L \in \mathbb{R}$, $L > 0$ derart, dass $\beta(M, t) \leq \eta$ für alle $t \geq L - 1$ gilt. Aus der asymptotischen Stabilität der Familie (u_T^a, F_T^a) folgt, dass $x_k^a(\xi) \in B$ in B verweilt für alle $\xi \in N$ und alle $k \geq 0$. Mit Annahme (1i) lässt sich der Fehler ε_k , gegeben durch

$$\varepsilon_k(\xi, z) := x^e(k, \xi) - x^a(k, z), \quad (4.42)$$

folgendermaßen abschätzen für $x(0) \in N$, $T \in]0, \hat{T}[$ und $k : kT \in [0, L]$:

$$\|\varepsilon_k(x(0))\| \leq \eta \quad (4.43)$$

Dies kann gezeigt werden, da (u_T, F_T^a) Mehr-Schritt-konsistent ist und dadurch für (u_T, F_T^e) eine Funktion $\alpha(\cdot, \cdot)$ und ein $T^* \in \mathbb{R}$, $T^* > 0$ existieren, so dass die Bedingungen (4.37) und (4.38) erfüllt sind für $(\bar{\mathcal{X}}, L, \eta)$, wobei man

$$\bar{\mathcal{X}} := N(\mathcal{X}, \eta) := \{x \mid \inf_{z \in N} \|x - z\| \leq \eta\} \quad (4.44)$$

für eine kompakte Menge $\mathcal{X} \in \mathbb{R}^n$ definiert.

Für den Induktionsanfang gilt

$$\|\varepsilon_0(x(0))\| = 0 \leq \alpha^1(0, T) \leq \eta. \quad (4.45)$$

Zudem ist

$$\|\varepsilon_k(x(0))\| \leq \alpha^k(0, T) \leq \eta \quad (4.46)$$

erfüllt und $(k+1)T \in [0, L]$. Da $x_k^a(x(0)) \in \mathcal{X}$ folgt aus der Definition von $\bar{\mathcal{X}}$, dass $x_k^a(x(0))$, $x_k^e(x(0)) \in \bar{\mathcal{X}}$. Daher gilt wegen (4.37)

$$\|\varepsilon_{k+1}(\xi)\| \leq \alpha^{k+1}(0, T). \quad (4.47)$$

Da $(k+1)T \in [0, T]$ ist folgt aus (4.38), dass $\alpha^{k+1}(0, T) \leq \eta$. Somit ergibt sich die Behauptung (4.43). Damit ist für ein den zuvor genannten Bedingungen entsprechendes Tripel $(T, k, x(0))$ die Ungleichung

$$\|x^e(k, x(0))\| \leq \|x^a(k, x(0))\| + \|\varepsilon_k(x(0))\| \leq \beta(\|x(0)\|, kT) + \eta \quad (4.48)$$

erfüllt. Da $\eta < \frac{R}{4}$ ist, gilt (4.40) für alle $x(0) \in N$, $T \in]0, \hat{T}[$ und alle $k \in \mathbb{N}$ mit $kT \in [0, L]$. Für größere k sei ohne Beschränkung der Allgemeinheit angenommen, dass $\hat{T} \leq 1$ ist. Weiter sei

$$k_T^* = \max \{k \in \mathbb{N} \mid kT \leq L\}. \quad (4.49)$$

Dann gilt für alle $T \in]0, \hat{T}[$

$$k_T^* T > L - T > L - 1. \quad (4.50)$$

Mit der Definition von L und Ungleichung (4.48) kann

$$\|x^e(k_T^*, x(0))\| \leq 2\eta \quad (4.51)$$

gefolgert werden. Für $k \in [k_T^*, 2k_T^*]$ gilt wegen der Zeitinvarianz, (4.43) und (4.48)

$$\begin{aligned} \|x^e(k, x(0))\| &= \|x^e(k - k_T^*, x^e(k_T^*, x(0)))\| \\ &\leq \|x^a(k - k_T^*, x^e(k_T^*, x(0)))\| + \|\varepsilon_{k-k_T^*}(x^e(k_T^*, x(0)))\| \\ &\leq \beta(2\eta, (k - k_T^*)T) + \eta, \end{aligned} \quad (4.52)$$

wobei zudem ausgenutzt wurde, dass die Kugel mit Radius 2η in N enthalten ist. Somit gilt für alle $k \in [k_T^*, 2k_T^*]$

$$\|x^e(k, x(0))\| \leq \beta(2\eta, 0) + \eta \leq \frac{R}{2} + \frac{R}{4} \leq R \quad (4.53)$$

woraus (4.40) folgt. Ausnutzung der Definition von k_T^* liefert außerdem

$$\|x^e(2k_T^*, x(0))\| \leq \beta(2\eta, k_T^*T) + \eta \leq 2\eta. \quad (4.54)$$

(2) Es sei angenommen, dass für jedes Paar (d, D) , $d, D \in \mathbb{R}^+$, ein $T^* \in \mathbb{R}$, $T^* > 0$ existiert, so dass für alle $T \in]0, T^*[$ die Folgerung

$$\left\{ \begin{array}{l} \|x\| \leq D \\ \max\{V_T(F_T^e(x, u_T(x))), V_T(x)\} \geq d \end{array} \right\} \\ \Rightarrow V_T(F_T^e(x, u_T(x))) - V_T(x) \leq -\frac{T}{2}\alpha_3(\|x\|) \quad (4.55)$$

gelte. Sei weiter N eine beliebige, aber feste beschränkte Umgebung des Ursprungs und es sei

$$D := \sup_{x \in N} \alpha_1^{-1} \circ \alpha_2(\|x\|). \quad (4.56)$$

Zudem sei d aus dem Intervall $]0, \alpha_1(D)[$. Für ein solches Paar (d, D) folgt aus der Annahme, dass ein $T^* > 0$ existiert, so dass für alle $T \in]0, T^*[$ und alle $x(0) \in N$

$$V_T(x_k^e) \leq \max\{V_T(x_0^e), d\} \quad (4.57)$$

und

$$\begin{aligned} \max\{V_T(x_{k+1}^e), V_T(x_k^e)\} &\geq d \\ \Rightarrow V_T(x_{k+1}^e) - V_T(x_k^e) &\leq -\frac{T}{2}\alpha_3(\alpha_2^{-1}(V_T(x_k^e))) \\ &=: -T\alpha(V_T(x_k^e)) \end{aligned} \quad (4.58)$$

gilt. (4.55) folgt per Induktion, wobei der Fall $k = 0$ klar ist. Die Definitionen von d, D , sowie die Ungleichungen (2.56) und (4.57) implizieren

$$\|x_k^e\| \leq \max\{\alpha_1^{-1}(V_T(x_0^e)), \alpha_1^{-1}(d)\} \leq D. \quad (4.59)$$

Also gilt entweder $V_T(x_{k+1}^e) \geq d$, woraus durch (4.55) und (4.59) $V_T(x_{k+1}^e) \leq V_T(x_k^e)$ folgt, oder es gilt $V_T(x_{k+1}^e) \leq d$. Damit ist der Induktionsschritt gezeigt für (4.57). Mit (4.55), (4.59) und (2.56) folgt zudem Behauptung (4.58).

Es wird nun die stetige, nicht negative und in $t \in \mathbb{R}$ stückweise lineare Funktion $y_T(t)$ eingeführt durch

$$y_T(t) := V_T(x_k^e) + \frac{t}{T-k}(V_T(x_{k+1}^e) - V_T(x_k^e)) \quad \forall t \in [kt, (k+1)T[, \quad k \geq 0. \quad (4.60)$$

Somit ist diese absolut stetig und die Ableitung ist für fast alle t gegeben durch

$$\dot{y}_T(t) = \frac{V_T(x_{k+1}^e) - V_T(x_k^e)}{T}, \quad t \in [kT, (k+1)T[. \quad (4.61)$$

Aus (4.58) folgt für $y_T(t)$:

$$\begin{aligned} \{y_T(t) \geq d, t \in [kT, (k+1)T]\} &\Rightarrow \max\{V_T(x_{k+1}^e), V_T(x_k^e)\} \geq d \\ &\Rightarrow V_T(x_{k+1}^e) - V_T(x_k^e) \leq -T\alpha(V_T(x_k^e)) \\ &\Rightarrow y_T(t) \leq V_T(x_k^e) \end{aligned} \quad (4.62)$$

Somit gilt mit (4.61) für fast alle t :

$$y_T(t) \geq d \Rightarrow \dot{y}_T(t) \leq -\alpha(V_T(x_k^e)), \quad k : t \in [kT, (k+1)T[\quad (4.63)$$

$$\leq -\alpha(y_T(t)) \quad (4.64)$$

Nach E.D.Sontags Arbeit [27] existiert eine Funktion $\beta_1 \in \mathcal{KL}$, die durch $\alpha = \alpha_3 \circ \alpha_2^{-1}$ festgelegt und unabhängig ist von T und d , so dass gilt:

$$y_T(t) \leq \max\{\beta_1(y_T(0), kT), d\}, \quad \forall t \geq 0 \quad (4.65)$$

Daraus folgt

$$V_T(k) \leq \max\{\beta_1(V_T(x_0^e), kT), d\}, \quad \forall k \geq 0 \quad (4.66)$$

unter Verwendung von $V_T(x_k^e) = y_T(t)$ mit $t = kT$, $y_T(0) = V_T(x_0^e)$. Nutzt man (2.56), so erhält man

$$\|x(k)\| \leq \alpha_1^{-1}(\beta_1(\alpha_2(\|x(0)\|), kT)) + \alpha_1^{-1}(d), \quad \forall x(0) \in N, k \geq 0 \quad (4.67)$$

Somit ist $\beta(s, \tau) := \alpha_1^{-1}(\beta_1(\alpha_2(s), \tau)) \in \mathcal{KL}$.

Zu zeigen bleibt die ursprüngliche Annahme (4.55). Dazu definiert man für ein Paar (d, D)

$$\delta := \frac{1}{2}\alpha_2^{-1}\left(\frac{d}{2}\right) \quad (4.68)$$

$$\Delta := \alpha_1^{-1} \circ \alpha_2(D) + \frac{1}{2}\alpha_2^{-1}\left(\frac{d}{2}\right). \quad (4.69)$$

Aus der Annahme (2ii) generiert man für die Menge

$$\bar{\mathcal{X}} := \{z \in \mathbb{R}^n : \|z\| \in [\delta, \Delta]\} \quad (4.70)$$

und die Werte $M, T^* \in \mathbb{R}$, $M, T^* > 0$. Aus Annahme (2i) erhält man für eine abgeschlossene Kugel \mathcal{X}_1 mit Radius D eine Funktion $\rho \in \mathcal{K}$ und eine Konstante $T_2^* \in \mathbb{R}$, $T_2^* > 0$. Man wählt $T_3^*, T_4^* \in \mathbb{R}$, $T_3^*, T_4^* > 0$, so dass die Ungleichungen

$$M\rho(T_3^*) \leq \frac{1}{2}\alpha_3 \circ \alpha_2^{-1} \circ \alpha_1(\delta) \quad (4.71)$$

$$T_4^*\rho(T_4^*) \leq \frac{1}{2}\alpha_2^{-1}\left(\frac{d}{2}\right) \quad (4.72)$$

erfüllt sind. Anschließend wird $T^* = \min\{T_1^*, T_2^*, T_3^*, T_4^*\}$ gesetzt.

Sei $\|x\| \leq D$. Aus (2.56) folgt

$$\begin{aligned} \|F_T^a(x, u_T(x))\| &\leq \alpha_1^{-1}(V_T(F_T^a(x, u_T(x)))) \\ &\leq \alpha_1^{-1}(V_T(x)) \leq \alpha_1^{-1} \circ \alpha_2(\|x\|) \\ \Rightarrow \|F_T^a(x, u_T(x))\| &\leq \alpha_1^{-1} \circ \alpha_2(D) \leq \Delta. \end{aligned} \quad (4.73)$$

Aus T^* und (4.36) ergibt sich für alle $T \in]0, T^*[$

$$\|F_T^e(x, u_T(x))\| \leq \alpha_1^{-1} \circ \alpha_2(D) + \frac{1}{2} \alpha_2^{-1} \left(\frac{d}{2} \right) = \Delta \quad (4.74)$$

Sei $V_T(F_T^e(x, u_T(x))) \geq \frac{d}{2}$. Wegen (2.56) folgt daraus

$$\|F_T^e(x, u_T(x))\| \geq \alpha_2^{-1} \left(\frac{d}{2} \right) \geq \delta. \quad (4.75)$$

Wiederum aus T^* und (4.36) erhält man für alle $T \in]0, T^*[$

$$\|F_T^a(x, u_T(x))\| \geq \frac{1}{2} \alpha_2^{-1} \left(\frac{d}{2} \right) = \delta \quad (4.76)$$

und mit (4.73) gilt

$$\|x\| \geq \alpha_2^{-1} \circ \alpha_1(\delta). \quad (4.77)$$

Die Ungleichungen (4.73) - (4.77) erlauben es zu zeigen, dass $V_T(F_T^e(x, u_T(x))) \geq \frac{d}{2}$ und $\|x\| \leq D$ Folgendes implizieren:

$$\begin{aligned} &V_T(F_T^e(x, u_T(x))) - V_T(x) = \\ &= V_T(F_T^a(x, u_T(x))) - V_T(x) + V_T(F_T^e(x, u_T(x))) - V_T(F_T^a(x, u_T(x))) \\ &\leq -T\alpha_3(\|x\|) + MT\rho(T^*) \leq -\frac{T}{2}\alpha_3(\|x\|) \end{aligned} \quad (4.78)$$

Sei nun angenommen, dass $V_T(F_T^e(x, u_T(x))) \leq \frac{d}{2}$ und $V_T(x) \geq d$ gelten. Aus (2.56) folgt $V_T(x) \geq T\alpha_3(\|x\|)$. Somit gilt

$$V_T(F_T^e(x, u_T(x))) - V_T(x) \leq \frac{1}{2}(d - V_T(x) - V_T(x)) \leq -\frac{T}{2}\alpha_3(\|x\|), \quad (4.79)$$

woraus sich die Annahme (4.55) ergibt. \square

4.1.3 Stabilitätszusammenhänge zwischen sampled-data und zeitdiskretem System

Da hier sampled-data Systeme zu betrachten sind, soll nun die Stabilität solcher Systeme untersucht werden. In der Arbeit von D.Nešić, A.R.Teel und P.V.Kokotović [21] wurde dabei gezeigt, dass unter gewissen Voraussetzungen die Stabilität des zeitdiskreten Systems (4.3), (4.4) und die Stabilität des sampled-data Systems (4.1), (4.2) sich gegenseitig implizieren. Dazu werden die dort aufgeführten Definitionen und Theoreme in geeigneter Weise abgeändert und zusammengefaßt, so dass sie dem hier vorliegenden Problem und der entsprechenden Notation genügen.

Theorem 4.13 (Sampled-data System asymptotisch stabil \iff zeitdiskretes System asymptotisch stabil).

Gegeben seien das kontinuierliche sampled-data System (4.1), (4.2) und das exakte zeitdiskrete sampled-data System (4.3), (4.4). Genau dann existieren Konstanten $r^s, r^b \in \mathbb{R}$, $r^s, r^b > 0$, und Funktionen $\beta \in \mathcal{KL}$ und $\tilde{\gamma} \in \mathcal{K}_\infty$, so dass die Bedingungen

(i) Das zeitdiskrete Modell (4.3), (4.4) ist lokal gleichmäßig asymptotisch stabil.

$$\begin{aligned} & \|x(k_0) - \xi(k_0)\| \leq r^s \Rightarrow \\ & \|x(k, x(k_0)) - \xi(k, \xi(k_0))\| \leq \beta(\|x(k_0) - \xi(k_0)\|, k - k_0), \quad k \geq k_0 \geq 0 \end{aligned} \quad (4.80)$$

(ii) Die Lösungen (4.5), (4.6) sind gleichmäßig beschränkt in $T \in \mathbb{R}$, wobei T die sampled-data Rate ist.

$$\begin{aligned} & \forall t_0 \geq 0, \|x(t_0) - \xi(t_0)\| \leq r^b \Rightarrow \\ & \|x(t, x(t_0)) - \xi(t, \xi(t_0))\| \leq \tilde{\gamma}(\|x(t_0) - \xi(t_0)\|), \quad t \in [t_0, t_0 + T] \end{aligned} \quad (4.81)$$

erfüllt sind, wenn das kontinuierliche sampled-data System (4.1), (4.2) lokal gleichmäßig asymptotisch stabil ist, also wenn

$$\begin{aligned} & \|x(t_0) - \xi(t_0)\| \leq r_x \Rightarrow \\ & \|x(t, x(t_0)) - \xi(t, \xi(t_0))\| \leq \bar{\beta}(\|x(t_0) - \xi(t_0)\|, t - t_0), \quad \forall t \geq t_0 \geq 0 \end{aligned} \quad (4.82)$$

gilt, wobei $r_x \in \mathbb{R}$ gegeben ist durch

$$r_x = \min \{ \tilde{\gamma}^{-1}(r^s), \tilde{\gamma}^{-1} \circ \beta_0^{-1}(r^b) \}, \quad (4.83)$$

mit $\beta_0(s) = \beta(s, 0)$, $\beta_0 \in \mathcal{K}_\infty$. Zudem wird $\bar{\beta} \in \mathcal{KL}$ durch die folgenden Bedingungen bestimmt:

(1) Wenn $\hat{\beta}(s, \tau) := \tilde{\gamma}(\beta(\tilde{\gamma}(s), \tau)) \leq P\hat{\beta}(s, \tau + 1)$ mit $P > 1$, $P, \tau \in \mathbb{R}$ beschränkt ist, dann kann man

$$\bar{\beta}(s, \tau) = \max \left\{ \tilde{\gamma}(s)e^{T-\tau}, P^2 \hat{\beta} \left(s, \frac{\tau}{T} \right) \right\} \quad (4.84)$$

verwenden.

(2) *Generell kann man*

$$\bar{\beta}(s, \tau) = \max \left\{ \tilde{\gamma}(s)e^{T-\tau}, 4 \max_{\eta \in [0, \tau]} 2^{-\eta} \hat{\beta} \left(s, \frac{\tau - \eta}{T} \right) \right\} \quad (4.85)$$

verwenden.

Beweis. Sei $r_x = \min \{ \tilde{\gamma}^{-1}(r^s), \tilde{\gamma}^{-1} \circ \beta_0^{-1}(r^b) \}$ mit $\beta_0(s) = \beta(s, 0) \in \mathcal{K}_\infty$. Dann gilt wegen $\tilde{\gamma}(s) \geq s$ und $\beta_0(s) \geq s$, $s \in \mathbb{R}$, die Abschätzung

$$r_x \leq \min \{ r^s, \beta_0^{-1}(r^b) \} \leq \min \{ \beta_0(r^s), r^b \} > 0. \quad (4.86)$$

Durch diese Wahl von r_x gilt für beliebiges $t_0 \geq 0$, $t_0 \in \mathbb{R}$, und $N \in \mathbb{N}$ mit $t_0 \in [NT, (N+1)T[$ die Abschätzung

$$\begin{aligned} & \|x(t_0) - \xi(t_0)\| \leq r_x \\ \Rightarrow & \|x(t, x(t_0)) - \xi(t, \xi(t_0))\| \leq \tilde{\gamma}(\|x(t_0) - \xi(t_0)\|) \quad \forall t \in [t_0, (N+1)T] \end{aligned} \quad (4.87)$$

Aus (4.87) und der Bedingung (4.80) folgt:

$$\begin{aligned} & \|x(t_0) - \xi(t_0)\| \leq r_x \\ \Rightarrow & \|x((N+1)T, x(t_0)) - \xi((N+1)T, \xi(t_0))\| \leq \min \{ r^s, \beta_0^{-1}(r^b) \} \\ \Rightarrow & \|x(kT, x(t_0)) - \xi(kT, \xi(t_0))\| \leq \min \{ \beta_0(r^s), r^b \} \quad \forall k \geq N+1 \end{aligned} \quad (4.88)$$

Hierbei impliziert die Bedingung (4.81), dass, wenn $\|x(t_0) - \xi(t_0)\| \leq r_x$ gilt, dann existiert $\|x(t, x(t_0)) - \xi(t, \xi(t_0))\|$ und ist beschränkt für alle $t \geq t_0$. Im Weiteren werden daher Anfangswerte mit $\|x(t_0) - \xi(t_0)\| \leq r_x$ betrachtet.

Da $e^{T-\tau} \geq 1$ für alle $\tau \leq T$ ist ergibt sich aus (4.87) folgende Abschätzung:

$$\begin{aligned} & \|x(t, x(t_0)) - \xi(t, \xi(t_0))\| \leq \tilde{\gamma}(\|x(t_0) - \xi(t_0)\|) \leq \tilde{\gamma}(\|x(t_0) - \xi(t_0)\|)e^{T-(t-t_0)} \\ & =: \beta_1(s, t - t_0) \quad \forall t \in [t_0, (N+1)T] \end{aligned} \quad (4.89)$$

Zudem gilt:

$$\begin{aligned} & \|x(t, x(t_0)) - \xi(t, \xi(t_0))\| \leq \\ & \tilde{\gamma}(\|x((k+N+1)T, x(t_0)) - \xi((k+N+1)T, \xi(t_0))\|), \\ & t \in [(k+N+1)T, (k+N+2)T], \quad k \geq 0 \end{aligned} \quad (4.90)$$

Zusammen mit der Bedingung der gleichmäßigen asymptotischen Stabilität des zeitdiskreten Systems (4.80) gilt für beliebige $t_0 \geq 0$ und $N \in \mathbb{N}$ mit $t_0 \in [NT, (N+1)T[$, $k \geq 0$ und $t \in [(k+N+1)T, (k+N+2)T[$:

$$\begin{aligned} & \|x(t, x(t_0)) - \xi(t, \xi(t_0))\| \leq \\ & \tilde{\gamma}(\beta(\|x((N+1)T, x(t_0)) - \xi((N+1)T, \xi(t_0))\|, kT)) \\ & \leq \tilde{\gamma}(\beta(\tilde{\gamma}(\|x(t_0) - \xi(t_0)\|), kT)) \\ & =: \hat{\beta}(\|x(t_0) - \xi(t_0)\|, kT). \end{aligned} \quad (4.91)$$

(1) Wenn ein $P \in \mathbb{R}$, $P > 0$ existiert, so dass für die \mathcal{KL} -Funktion die Ungleichung

$$\hat{\beta}(s, \tau) := \tilde{\gamma}(\beta(\tilde{\gamma}(s), \tau)) \leq P\hat{\beta}(s, \tau + 1) \quad (4.92)$$

für alle $s \geq 0$, $\tau \in \mathbb{N}$ gilt, dann kann man folgende Abschätzung formulieren für $t \in [(k + N + 1)T, (k + N + 2)T[$ und $k \geq 0$.

$$\begin{aligned} \|x(t, x(t_0)) - \xi(t, \xi(t_0))\| &\leq \hat{\beta}(\|x(t_0) - \xi(t_0)\|, kT) \leq \\ &\leq P^2 \hat{\beta}(\|x(t_0) - \xi(t_0)\|, (k + 2)T) \end{aligned} \quad (4.93)$$

Für t_0 , N und t gilt nach deren Definition $\frac{t-t_0}{T} < \frac{(k+N+2)T-NT}{T} = k + 2$ für alle $k \geq 0$. Da $\hat{\beta}$ eine \mathcal{KL} Funktion ist lässt sich (4.93) abschätzen durch:

$$\begin{aligned} \|x(t, x(t_0)) - \xi(t, \xi(t_0))\| &\leq P^2 \hat{\beta}(\|x(t_0) - \xi(t_0)\|, (k + 2)T) \\ &< P^2 \hat{\beta}\left(\|x(t_0) - \xi(t_0)\|, \frac{t - t_0}{T}\right) \\ &=: \beta_2(\|x(t_0) - \xi(t_0)\|, t - t_0), \quad t \geq (N + 1)T. \end{aligned} \quad (4.94)$$

Führt man eine neue \mathcal{KL} Funktion $\bar{\beta}$ mit $\bar{\beta}(s, \tau) := \max\{\beta_1(s, \tau), \beta_2(s, \tau)\}$ ein, so folgt aus (4.89) und (4.94):

$$\begin{aligned} \|x(t_0) - \xi(t_0)\| &\leq r_x \\ \Rightarrow \|x(t, x(t_0)) - \xi(t, \xi(t_0))\| &\leq \bar{\beta}(\|x(t_0) - \xi(t_0)\|, t - t_0), \quad t \geq t_0. \end{aligned} \quad (4.95)$$

(2) Existiert jedoch kein solches $P > 0$, so lässt sich nach Lemma 1 und Korollar 1 aus [21] eine Funktion $\tilde{\beta} \in \mathcal{KL}$ finden, die $\hat{\beta}$ mit $P = 2$ majorisiert:

$$\tilde{\beta} = \max_{\eta \in [0, \tau]} 2^{-\eta} \tilde{\gamma}\left(\beta\left(\tilde{\gamma}(s), \frac{\tau - \eta}{T}\right)\right) \quad (4.96)$$

Für dieses $\tilde{\beta}$ lassen sich die unter (1) durchgeführten Berechnungen wiederholen.

□

In der besagten Arbeit von D.Nešić, A.R.Teel und P.V.Kokotović [21] sind zudem Theoreme zur globalen asymptotischen, lokalen und globalen exponentiellen Stabilität des sampled-data Systems nachlesbar.

Bemerkung 4.14. *Eine Konsequenz der so konstruierten Abschätzungen ist, dass man die Art der Stabilität des sampled-data Systems aus der gleichen Art Stabilität des exakten zeitdiskreten Systems unter einer gleichmäßigen Beschränktheitsannahme folgern kann, wobei im vorherigen Kapitel bereits gezeigt wurde, dass die Stabilität des approximierten zeitdiskreten Systems die Stabilität des exakten zeitdiskreten Systems unter gewissen Bedingungen impliziert.*

Bemerkung 4.15. *Alternativ kann die Aussage auch ohne die gleichmäßige Beschränktheitsannahme gezeigt werden, denn für eine \mathcal{KL} -Funktion β kann eine Funktion $\beta_1 \in \mathcal{KL}$ gefunden werden, für die*

$$\beta(s, \tau) \leq \beta_1(s, \tau + 2), \quad \forall s, \tau \geq 0 \quad (4.97)$$

gilt. Zum Beweis der Aussage sei auf Bemerkung 3 in der Arbeit von von D.Nešić, A.R.Teel und P.V.Kokotović [21] verwiesen. Der hier verwendete Ansatz erlaubt es jedoch expliziter den Zusammenhang zwischen β und β_1 darzustellen.

4.2 Stabilität des modellprädiktiven Regelungsansatzes

Mit Hilfe der Vorarbeiten der vorangegangenen Kapitel lassen sich nun die Ergebnisse der Arbeit von G.Grimm, M.J.Messina, A.R.Teel und S.Tuna [7] auf das dem vorliegende Problem **MPC** anwenden. Unter annehmbaren Bedingungen soll damit die Folgerung bewiesen werden, dass aus der asymptotischen Stabilität des zeitkontinuierlichen Systems die semiglobal praktisch asymptotische Stabilität des zeitkontinuierlichen closed-loop sampled-data Systems folgt. Die nun folgenden Theoreme und Beweise sind der Arbeit von D.Nešić und L.Grüne [23] entnommen.

Theorem 4.16 (Stabilität des modellprädiktiven Regelungsansatzes).

Es gelten die folgenden Bedingungen für das Problem (3.24) unter (3.25), (3.26), (3.27), (3.28), (3.29):

- (i) *Der Lagrange-Anteil l und der Mayer-Anteil F sind stetig;*
- (ii) *Der Steuerbereich $U \subset \mathbb{R}^m$ ist beschränkt;*
- (iiia) *Das zeitkontinuierliche closed-loop System (3.8) mit kontinuierlichem Zustands-Feedback $u = u(x)$, $u(x) \in U \forall x \in \mathbb{R}^n$, ist global asymptotisch stabil;*
- (iiib) *Es existiert eine Konstante $r_0 \in \mathbb{R}$, $r_0 > 0$ und eine Funktion $\gamma \in \mathcal{K}_\infty$, so dass für den Lagrange-Anteil gilt:*

$$l(y, u) \geq \max \left(\max_{\|x\| \leq 2\|y\|} \|f(x, u)\|, \gamma(\|y\|) \right) \quad \forall y \in \mathbb{R}^n \text{ mit } \|y\| \geq r_0; \quad (4.98)$$

- (iiic) *Das Kontrollsystem $f(., .)$ und die Steuerung $u(., .)$ sind lokal Lipschitz-stetig in ihren Argumenten;*
- (iv) *Die Wertefunktion ist von der Gestalt, dass für die Funktion $\bar{\alpha} \in \mathcal{K}_\infty$ und $i \in \mathbb{N}$ gilt:*

$$V_i(\xi, x) \leq \bar{\alpha}(\|(\xi, x)\|) \quad \forall i \geq 0, \forall (\xi, x) \in \mathbb{R}^{2n}. \quad (4.99)$$

Dann existiert eine Funktion $\beta \in \mathcal{KL}$, so dass für jedes Paar (Δ, δ) , $\Delta, \delta \in \mathbb{R}$, ein Horizont für die sampled-data Rate T durch ein derartiges $M_1^* \in \mathbb{Z}_{\geq 1}$ existiert, für welches für alle $(\xi, x) \in B_\Delta$ und $M \geq M_1^*$ die Lösungen des closed-loop Systems (3.20), (3.21) die Abschätzung

$$\|(\xi(t), x(t))\| \leq \max\{\beta(\|(\xi, x)\|, t), \delta\} \quad \forall t \geq 0 \quad (4.100)$$

erfüllen. Gilt zudem die Bedingung

- (v) Der Mayer-Anteil des Zielfunktional $F(\cdot, \cdot)$ aus (4.7) ist eine Lyapunov-Kontroll-Funktion des exakten zeitdiskreten sampled-data Systems (4.3), (4.4) mit Funktionen $\underline{\alpha}_F, \bar{\alpha}_Q \in \mathcal{K}_\infty$.

dann existiert eine Funktion $\beta \in \mathcal{KL}$, so dass für jedes Paar (Δ, δ) , $\Delta, \delta \in \mathbb{R}$, eine Zahl $M_2^* \in \mathbb{Z}_{\geq 1}$ existiert, so dass für alle $(\xi, x) \in B_\Delta$ und $M \geq M_2^*$ die Lösungen des closed-loop Systems (3.20), (3.21) die Ungleichung (4.100) erfüllen.

Bemerkung 4.17. Hierbei wird die Mindestlänge des Horizonts im Allgemeinen kürzer, wenn die Bedingung (v) erfüllt ist.

Zum Beweis dieses Theorems wird nun zuerst folgende Proposition bewiesen:

Proposition 4.18. Seien $U \subset \mathbb{R}^m$ und Q aus (4.9) gegeben und es gelten die Bedingungen (iiia), (iiib) und (iiic) aus Theorem 4.16. Dann existiert eine Vergleichsfunktion $\alpha_Q \in \mathcal{K}_\infty$, so dass gilt:

$$\|x\| + Q(\xi, x, u) \geq \alpha_Q(\|(\xi, x)\|) \quad \forall (\xi, x) \in \mathbb{R}^{2n}, u \in U \quad (4.101)$$

Beweis. Es sei angenommen, dass $Q(\xi, x, u) > 0$ gilt, wenn $\xi \neq x$. Dabei impliziert $\xi \neq x$ die Existenz eines $\tau \in \mathbb{R}^n$ mit $\tau = \tau(\xi, x) > 0$, so dass $\|\xi(s, \xi, u) - x(s, x)\| > 0$ für alle $s \in [0, \tau]$ gilt. Eingesetzt in (4.9) folgt

$$Q(\xi, x, u) = \int_0^T l(\xi(s, \xi, u) - x(s, x), u) ds \geq \int_0^\tau \gamma(\|\xi(s, \xi, u) - x(s, x)\|) ds > 0. \quad (4.102)$$

Da offensichtlich $Q(0, 0, u) \geq 0$ für alle $u \in U$ gilt, kann $\|x\| + Q(\xi, x, u)$ durch eine positiv definite Funktion abhängig von (ξ, x) von unten beschränkt werden.

Sei eine Folge von Anfangswerten (ξ_j, x_j) , $j \in \mathbb{N}$ mit $\|(\xi_j, x_j)\| \rightarrow \infty$, $j \rightarrow \infty$ gegeben. Dann gilt

$$\|x_j\| + Q(\xi_j, x_j, u) \rightarrow \infty \quad (4.103)$$

falls $\|x_j\| \rightarrow \infty$.

Ist x_j dagegen beschränkt, dann gilt $\|\xi_j\| \rightarrow \infty$. Gezeigt werden soll nun, dass auch hierfür (4.103) erfüllt ist.

Da das zeitkontinuierliche closed-loop System nach Voraussetzung asymptotisch stabil ist existiert eine Konstante $R \in \mathbb{R}$, $R > 0$, so dass $\|x(t, x_j)\| \leq R$ für alle $j \in \mathbb{N}$, $t \geq 0$ gilt. Es kann also ohne Beschränkung der Allgemeinheit $R \geq r_0$ für r_0 aus Bedingung (iiib)

angenommen werden. Wählt man nun $j > 0$ derart, dass $\|\xi_j\| \geq 9R$ gilt, so folgt $\|\xi_j - x_j\| \geq 8R$. Zu betrachten sind für beliebige $u \in U$ und die Lösungen $\xi(t, \xi_j, u)$ des sampled-data Systems auf dem Intervall $[0, T]$. Hierfür ergeben sich die folgenden beiden Fälle:

$$1. \quad \|x(t, x_j) - \xi(t, \xi_j, u)\| \geq \frac{\|x_j - \xi_j\|}{2} \quad \forall t \in [0, T]$$

Daraus folgt

$$Q(\xi_j, x_j, u) \geq T\gamma \left(\frac{\|x_j - \xi_j\|}{2} \right) \quad (4.104)$$

direkt aus der Ungleichung $l(r, u) \geq \gamma(r)$.

$$2. \quad \|x(t, x_j) - \xi(t, \xi_j, u)\| < \frac{\|x_j - \xi_j\|}{2} \quad \forall t \in [0, T]$$

Sei $\tau \in [0, T]$ die minimale Zeit für die diese Ungleichung gilt. Da die Ungleichungen $\|x_j - x(t, x_j)\| \leq 2R \leq \frac{\|x_j - \xi_j\|}{4}$ gelten, folgt

$$\begin{aligned} \|\xi(\tau, \xi_j, u) - \xi_j\| &\geq \|\xi_j - x(\tau, x_j)\| - \|x(\tau, x_j) - \xi(\tau, \xi_j, u)\| \\ &\geq \|\xi_j - x(\tau, x_j)\| - \frac{\|x_j - \xi_j\|}{2} \\ &\geq \|\xi_j - x_j\| - 2R - \frac{\|x_j - \xi_j\|}{2} \\ &\geq \|\xi_j - x_j\| - \frac{\|x_j - \xi_j\|}{4} - \frac{\|x_j - \xi_j\|}{2} = \frac{\|x_j - \xi_j\|}{4}. \end{aligned}$$

$$\Rightarrow \int_0^\tau \|f(\xi(t, \xi_j, u), u)\| dt \geq \frac{\|x_j - \xi_j\|}{4} \quad (4.105)$$

Aus der Minimalität von τ folgt, dass für alle $t \in [0, \tau]$ die Ungleichung

$$\|(\xi(t, \xi_j, u) - x(t, x_j))\| \geq \frac{\|x_j - \xi_j\|}{2} \geq R \quad (4.106)$$

gilt. Daraus ergibt sich

$$\|\xi(t, \xi_j, u)\| \leq \|\xi(t, \xi_j, u) - x(t, x_j)\| + R \leq 2\|\xi(t, \xi_j, u) - x(t, x_j)\| \quad (4.107)$$

Zusammen mit der Ungleichung aus Bedingung (iiib) impliziert dies

$$l(\|\xi(t, \xi_j, u) - x(t, x_j)\|, u) \geq \|f(\xi(t, \xi_j, u), u)\|. \quad (4.108)$$

$$\begin{aligned} \Rightarrow Q(\xi_j, x_j, u) &\geq \int_0^\tau l(x(t, x_j) - \xi(t, \xi_j, u), u) dt \\ &\geq \int_0^\tau \|f(\xi(t, \xi_j, u), u)\| dt \\ &\geq \frac{\|x_j - \xi_j\|}{4} \end{aligned} \quad (4.109)$$

In beiden Fällen gilt: Es existiert eine von u unabhängige untere Schrankenfunktion für $Q(\xi_j, x_j, u)$, die für $j \rightarrow \infty$ gegen ∞ strebt. Somit folgt weiter $Q(\xi_j, x_j, u) \rightarrow \infty$ und es gilt auch für diesen Fall (4.103).

Folglich sind alle Voraussetzungen des Lemmas 4.3 aus dem Buch von H.K.Khalil [13] erfüllt, da $\|x\| + Q(\xi, x, u)$ eine stetige positiv definite Funktion ist und radial unbeschränkt ist. Somit ist die Existenz einer Funktion $\alpha_Q \in \mathcal{K}_\infty$ gesichert, für welche die Behauptung gilt. \square

Beweis zu Theorem 4.16.

Gezeigt wird, dass die Bedingungen des Theorems implizieren, dass alle Bedingungen der Proposition 4.5, die sich aus dem Theorem 4.3 ergab, für das zugrunde liegende exakte zeitdiskrete System gelten. Dabei ergeben sich aus dem Theorem folgende Implikationen:

- Da l stetig ist folgt die Stetigkeit von Q aus (4.9) \Rightarrow (i) aus Proposition 4.5 gilt;
- (ii), (iv) und (v) des Theorems entsprechen den jeweiligen Bedingungen von Proposition 4.5;
- Aus Bedingung (iiia) des Theorems folgt, dass das exakte zeitdiskrete closed-loop sampled-data System (4.10), (4.11) asymptotisch stabil ist.

Aus dem zeitdiskreten umgekehrten Lyapunov-Theorem folgt, wie es in Kapitel 7 der Arbeit von C.M.Kellet [12] gezeigt wurde, dass eine Lyapunov-Funktion $U(x)$ und Vergleichsfunktionen $\alpha_1, \alpha_2, \alpha_3 \in \mathcal{K}_\infty$ existieren, so dass für alle $x \in \mathbb{R}^n$ gilt:

$$\alpha_1(\|x\|) \leq U(x) \leq \alpha_2(\|x\|) \quad (4.110)$$

$$U(H(x)) - U(x) \leq -\alpha_3(\|x\|) \quad (4.111)$$

Gemäß der zuvor gezeigten Proposition 4.18 folgt aus (iiia), (iiib) und (iiic) die Existenz von $\alpha_Q \in \mathcal{K}_\infty$, so dass

$$\|x\| + Q(\xi, x, u) \geq \alpha_Q(\|(\xi, x)\|) \quad \forall (\xi, x) \in \mathbb{R}^{2n}, u \in U \quad (4.112)$$

gilt. Sei $\gamma_W \in \mathcal{K}_\infty$ beliebig, aber fest. Sei weiter $W(\xi, x) := U(x)$. Dann kann man zeigen, dass das System (4.3), (4.4) detektierbar ist durch Q mit $(\alpha_W, \gamma_W, \bar{\alpha}_W)$, die festgelegt sind durch

$$\bar{\alpha}_W(s) := \alpha_2(s); \alpha_W(s) := \tilde{\alpha} \circ \frac{1}{2}\alpha_Q(s), \quad (4.113)$$

$$\tilde{\alpha}(s) := \min \{ \alpha_3(s), \gamma_W(s) \}, \quad (4.114)$$

denn es gilt

$$W(\xi, x) = U(x) \leq \alpha_2(\|x\|) \leq \alpha_2(\|(\xi, x)\|) =: \bar{\alpha}_W(\|(\xi, x)\|) \quad \forall (\xi, x) \in \mathbb{R}^{2n}, \quad (4.115)$$

wodurch (4.12) mit $\bar{\alpha}_W(s) := \alpha_2(s)$ erfüllt ist. Aus der Definition von $\tilde{\alpha}$, der Dreiecksungleichung für \mathcal{K} -Funktionen, $\alpha(\frac{1}{2}(s_1 + s_2)) \leq \alpha(s_1) + \alpha(s_2)$ für alle $s_1, s_2 \geq 0$, und (4.112) folgt dann die Behauptung:

$$\begin{aligned}
W(G(\xi, u), H(x)) - W(\xi, x) &= U(H(x)) - U(x) \\
&\leq -\alpha_3(\|x\|) \\
&= -\alpha_3(\|x\|) - \gamma_W(Q(\xi, x, u)) + \gamma_W(Q(\xi, x, u)) \\
&\leq -\tilde{\alpha}(\|x\|) - \tilde{\alpha}(Q(\xi, x, u)) + \gamma_W(Q(\xi, x, u)) \\
&\leq -\tilde{\alpha} \left(\frac{1}{2}(\|x\| + Q(\xi, x, u)) \right) + \gamma_W(Q(\xi, x, u)) \\
&\leq -\tilde{\alpha} \circ \frac{1}{2} \alpha_Q(\|(\xi, x)\|) + \gamma_W(Q(\xi, x, u)). \quad (4.116)
\end{aligned}$$

Daher gilt:

\Rightarrow Nach Theorem 4.3 ist das exakte zeitdiskrete closed-loop sampled-data System (4.10), (4.11) im Zeithorizont M semiglobal praktisch asymptotisch stabil.

\Rightarrow Durch Ausnutzung des Theorems 4.13 kann man auf die semiglobale praktische asymptotische Stabilität des sampled-data Systems (4.1), (4.2) im Zeithorizont M schließen.

□

Bemerkung 4.19. *Obwohl hier von unbeschränkten optimalen Steuerprozessen ausgegangen wird kann es dennoch sein, dass das Problem keine zulässigen Lösungen $(\xi, x) \in \mathbb{R}^{2n}$ und $u \in \mathcal{U}$ besitzt. Zum einen kann die Kontrollierbarkeit des sampled-data Systems und damit des zeitdiskreten Systems wegen Sampling verloren gehen, zum anderen ist es bei Vorliegen endlicher Entweichzeiten möglich, dass die Lösung des sampled-data Systems nicht definiert ist für manche Startwerte und Steuerungen. Beide Probleme können allerdings gelöst werden für sehr kleine sampled-data Raten.*

Bemerkung 4.20. *In der Arbeit von G.Grimm, M.J.Messina, A.R.Teel und S.Tuna [7] werden Schranken für die Horizontlänge konstruiert, die die zweckmäßigen Stabilitätseigenschaften des zu betrachtenden zeitdiskreten Systems garantieren. Da hier kein konstruktiver Beweis vorliegt werden derartige Schranken hier nicht bestimmt.*

Bemerkung 4.21. *Um die Detektierbarkeit des exakten zeitdiskreten Systems zu zeigen wurde im Beweis des Theorems 4.16 eine beliebige Funktion $\gamma_W \in \mathcal{K}_\infty$ verwendet. Um eine Lyapunov-Funktion für das exakte zeitdiskrete closed-loop sampled-data System (4.10), (4.11) zu erhalten kann man sich Ergebnisse aus der Arbeit von G.Grimm, M.J.Messina, A.R.Teel und S.Tuna [7] zunutze machen, indem man*

$$\gamma_W(s) \leq s, \quad \forall s \geq 0 \quad (4.117)$$

wählt. Die Funktion

$$V(\xi, x) := V_M(\xi, x) + U(x) \quad (4.118)$$

stellt dann die Lyapunov-Funktion dar, wobei U die Ungleichungen (4.110), (4.111) erfüllen muss.

In fast analoger Weise können zusätzliche Resultate für asymptotische Stabilisierbarkeit und exponentielle Stabilisierbarkeit mittels der Propositionen 4.7 und 4.8 gewonnen werden:

Theorem 4.22. *Es gelten die folgenden Bedingungen für das Problem (3.24) unter (3.25), (3.26), (3.27), (3.28), (3.29):*

- (i) *Der Lagrange-Anteil l und der Mayer-Anteil F sind stetig;*
- (ii) *Der Steuerbereich $U \subset \mathbb{R}^m$ ist beschränkt;*
- (iiia) *Das zeitkontinuierliche closed-loop System (3.8) mit kontinuierlichem Zustands-Feedback $u = u(x)$, $u(x) \in U \forall x \in \mathbb{R}^n$, ist lokal exponentiell stabil und global asymptotisch stabil;*
- (iiib) *Es existiert eine Konstante $r_0 \in \mathbb{R}$, $r_0 > 0$ und eine Funktion $\gamma \in \mathcal{K}_\infty$, so dass für den Lagrange-Anteil gilt:*

$$l(y, u) \geq \max \left(\max_{\|x\| \leq 2\|y\|} \|f(x, u)\|, \gamma(\|y\|) \right) \quad \forall y \in \mathbb{R}^n \text{ mit } \|y\| \geq r_0; \quad (4.119)$$

Weiter existieren $a_l, \Delta_l \in \mathbb{R}$, $a_l > 0$ und $\Delta_l > 0$, so dass gilt:

$$l(y, u) \geq a_l(\|y\|^2 + \|u\|^2) \quad \forall \|y\| \leq \Delta_l, u \in U \quad (4.120)$$

- (iiic) *Das Kontrollsystem $f(.,.)$ und die Steuerung $u(.)$ sind lokal Lipschitz-stetig in ihren Argumenten;*
- (iv) *Die Wertefunktion ist von der Gestalt, dass für die lokal quadratische Funktion $\bar{\alpha} \in \mathcal{K}_\infty$ und $i \in \mathbb{N}$ gilt:*

$$V_i(\xi, x) \leq \bar{\alpha}(\|(\xi, x)\|) \quad \forall i \geq 0, \forall (\xi, x) \in \mathbb{R}^{2n}. \quad (4.121)$$

Dann existiert eine Funktion $\beta \in \mathcal{KL}$, so dass für jedes $\Delta \in \mathbb{R}$, $\Delta > 0$, ein Horizont für die sampled-data Rate T durch ein derartiges $M_1^* \in \mathbb{Z}_{\geq 1}$ existiert in der Art, dass für alle $(\xi, x) \in B_\Delta$ und $M \geq M_1^*$ die Lösungen des closed-loop Systems (3.20), (3.21) die Abschätzung

$$\|(\xi(t), x(t))\| \leq \beta(\|(\xi, x)\|, t) \quad \forall t \geq 0 \quad (4.122)$$

erfüllt. Gilt weiter

- (v) *Der Mayer-Anteil des Zielfunktional $F(.,.)$ aus (4.7) ist eine Lyapunov-Kontroll-Funktion des Systems (4.3), (4.4) mit lokal quadratischen Funktionen $\bar{\alpha}_Q, \underline{\alpha}_F \in \mathcal{K}_\infty$.*

dann existiert eine Funktion $\beta \in \mathcal{KL}$, so dass für jedes $\Delta \in \mathbb{R}$, $\Delta > 0$, ein $M_2^* \in \mathbb{Z}_{\geq 1}$ existiert, so dass für alle $(\xi, x) \in B_\Delta$ und $M \geq M_2^*$ die Lösungen des closed-loop Systems (3.20), (3.21) die Ungleichung (4.122) erfüllen.

Proposition 4.23. Sei $U \subset \mathbb{R}^m$ und Q aus (4.9). Es gelten die Bedingungen (iiia), (iiib) und (iiic) aus Theorem 4.22. Dann existieren $D, a \in \mathbb{R}$ mit $D, a > 0$, so dass gilt:

$$\|x\|^2 + Q(\xi, x, u) \geq a\|(\xi, x)\|^2 \quad \forall \|x\| \leq D, u \in U \quad (4.123)$$

Gelten alle Bedingungen global, dann gilt (4.123) für alle $(\xi, x) \in \mathbb{R}^{2n}$ und alle $u \in U$.

Beweis. Seien $D, t_1 \in \mathbb{R}$, $D > 0$ und $t_1 > 0$ derart, dass für alle $\|(\xi, x)\| \leq D$ und $u \in U$ gilt:

$$\max\{\|\xi(t)\|, \|x(t)\|\} \leq \frac{\Delta_l}{2} \quad \forall t \in [0, t_1] \quad (4.124)$$

Somit gilt für den Abstand der Trajektorien $\|\xi(t) - x(t)\| \leq \Delta_l$ für alle t auf dem Intervall $[0, t_1]$ und da f und u lokal Lipschitz-stetig sind, ist es immer möglich derartige D und t_1 zu finden. Weiter sei $L \in \mathbb{R}$, $L > 0$, so dass

$$\|f(x, u)\| \leq L(\|x\| + \|u\|) \quad \forall \|x\| \leq \Delta_l, u \in U \quad (4.125)$$

$$\|f(x, u(x))\| \leq L\|x\| \quad \forall \|x\| \leq \Delta_l \quad (4.126)$$

gilt.

$$\Rightarrow \|x(t, x)\| \leq e^{Lt}\|x\| \quad \forall t \in [0, t_1], x \in B_D \quad (4.127)$$

Weiter gilt für alle $(\xi, x) \in B_D$, $u \in U$ und $t \in [0, t_1]$:

$$\begin{aligned} \left\| \frac{d}{dt} \xi^T(t) \xi(t) \right\| &= \|2\xi^T(t) f(\xi(t), u)\| \\ &\leq 2L(\|\xi(t)\|^2 + \|\xi(t)\| \cdot \|u\|) \\ &\leq 2L(3\|\xi(t)\|^2 + 2\|u\|^2) \\ &\leq 6L(\|\xi(t)\|^2 + \|u\|^2) \\ &\Rightarrow \frac{d}{dt} \xi^T(t) \xi(t) \geq -6L(\|\xi(t)\|^2 + \|u\|^2). \end{aligned} \quad (4.128)$$

Da die Bedingungen des Vergleichslemmas erfüllt sind, vergleiche hierzu Lemma 3.4 aus dem Buch von H.K.Khalil [13], und zudem $\|\xi(t)\| \geq 0 \forall t$ gilt, kann man folgern, dass

$$\|\xi(t)\|^2 \geq \max\{e^{-6Lt}\|\xi\|^2 + (e^{-6Lt} - 1)\|u\|^2, 0\} =: b(t, \xi, x). \quad (4.129)$$

Wähle t_2 so, dass $a_1 t_2 e^{2Lt_2} \leq \frac{1}{2}$ gilt. Da für alle $a, b \geq 0$

$$(\max\{a - b, 0\})^2 + b^2 \geq \frac{a^2}{4} \quad (4.130)$$

gilt, folgt aus der Definition von $b(t, \xi, u)$:

$$\begin{aligned}
b(t_2, \xi, u) + \|u\|^2 &= \max \{ e^{-6Lt_2} \|\xi\|^2 + (e^{-6Lt_2} - 1) \|u\|^2, 0 \} + \|u\|^2 \\
&\geq e^{-6Lt_2} \|\xi\|^2 + e^{-6Lt_2} \|u\|^2 \\
&\geq e^{-6Lt_2} \|\xi\|^2
\end{aligned} \tag{4.131}$$

Zum Abschluss verwendet man (4.129), (4.130) in:

$$\begin{aligned}
\|x\|^2 + Q(\xi, x, u) &= \|x\|^2 + \int_0^T l(\xi(t) - x(t), u) dt \\
&\geq \|x\|^2 + a_l \int_0^T (\|\xi(t) - x(t)\|^2 + \|u\|^2) dt \\
&\geq \|x\|^2 + a_l \int_0^{t_2} ((\max \{ \|\xi(t)\| - \|x(t)\|, 0 \})^2 + \|u\|^2) dt \\
&\geq \|x\|^2 + a_l t_2 \left(\left(\max \{ \sqrt{b(t_2, \xi, u)} + e^{Lt_2} \|x\|, 0 \} \right)^2 + \|u\|^2 \right) \\
&\geq \frac{1}{2} \|x\|^2 + a_l t_2 \left(e^{2Lt_2} \|x\|^2 + \left(\max \{ \sqrt{b(t_2, \xi, u)} - e^{Lt_2} \|x\|, 0 \} \right)^2 + \|u\|^2 \right) \\
&\geq \frac{1}{2} \|x\|^2 + a_l t_2 \left(\frac{b(t_2, \xi, u)}{4} + \|u\|^2 \right) \\
&\geq \frac{1}{2} \|x\|^2 + \frac{a_l t_2 e^{-6Lt_2}}{4} \|\xi\|^2 \\
&\geq a \|(\xi, x)\|^2,
\end{aligned} \tag{4.132}$$

wobei

$$a := \min \left\{ \frac{1}{2}, \frac{a_l t_2 e^{-6Lt_2}}{4} \right\}. \tag{4.133}$$

Wenn alle Bedingungen global gelten, so ist auch das Resultat global gültig. \square

Beweis zu Theorem 4.22.

Analog zum Beweis von Theorem 4.16 mit folgenden Änderungen:

- Da (3.7) lokal exponentiell stabil und global asymptotisch stabil ist existiert nach Lemma 10.1.5 aus dem Buch von A.Isidori [10] eine Funktion $U(\cdot)$, so dass die Gleichungen (4.110), (4.111) mit lokal quadratischen Funktionen $\alpha_1, \alpha_2, \alpha_3$ gelten.
- (iiib) und (iiic) implizieren durch Proposition 4.23, dass α_Q in (4.112) lokal quadratisch ist.
- Zudem kann man ein beliebiges lokal quadratisches γ_W wählen, so dass gilt: $\bar{\alpha}_W, \alpha_W$, definiert in (4.113), sind lokal quadratisch.

\Rightarrow Alle Bedingungen der Proposition 4.7 gelten für das System (4.10), (4.11) und der Beweis verläuft in denselben Schritten wie der Beweis zu Theorem 4.16. \square

Theorem 4.24. *Es gelten die folgenden Bedingungen für das Problem (3.24) unter (3.25), (3.26), (3.27), (3.28), (3.29):*

- (i) *Der Lagrange-Anteil l und der Mayer-Anteil F sind stetig;*
- (ii) *Der Steuerbereich $U \subset \mathbb{R}^m$ ist beschränkt;*
- (iiia) *Das zeitkontinuierliche closed-loop System (3.8) mit kontinuierlichem Zustands-Feedback $u = u(x)$, $u(x) \in U \forall x \in \mathbb{R}^n$, ist global exponentiell stabil;*
- (iiib) *Es existiert eine Konstante $a_l \in \mathbb{R}$, $a_l > 0$, so dass gilt:*

$$l(y, u) \geq a_l(\|y\|^2 + \|u\|^2) \quad \forall y \in \mathbb{R}^n, u \in U; \quad (4.134)$$

- (iiic) *Das Kontrollsystem $f(.,.)$ und die Steuerung $u(.)$ sind global Lipschitz-stetig in ihren Argumenten;*
- (iv) *Die Wertefunktion ist von der Gestalt, dass für die global quadratische Funktion $\bar{\alpha} := \bar{a} \cdot s^2$, $\bar{a}, s \in \mathbb{R}$, $\bar{a} > 0$, $i \in \mathbb{N}$ gilt:*

$$V_i(\xi, x) \leq \bar{\alpha}(\|(\xi, x)\|) \quad \forall i \geq 0, \forall (\xi, x) \in \mathbb{R}^{2n}. \quad (4.135)$$

Dann existieren Konstanten $K, \lambda, \Delta \in \mathbb{R}$, $K \leq 1$, $\lambda > 0$, $\Delta > 0$, und ein Horizont für die sampled-data Rate T durch ein derartiges $M_1^* \in \mathbb{Z}_{\geq 1}$, so dass für alle Paare $(\xi, x) \in B_\Delta$ und $M \geq M_1^*$ die Lösungen des Systems (3.20), (3.21) die Ungleichung

$$\|(\xi(t), x(t))\| \leq K e^{-\lambda t} \|(\xi, x)\| \quad \forall t \geq 0 \quad (4.136)$$

erfüllen. Wenn zusätzlich gilt:

- (v) *F ist eine Lyapunov-Kontroll-Funktion des Systems (4.1), (4.2) mit lokal quadratischen Funktionen $\bar{\alpha}_Q, \underline{\alpha}_F \in \mathcal{K}_\infty$.*

Dann existieren $K, \lambda, \Delta \in \mathbb{R}$, $K \leq 1$, $\lambda > 0$, $\Delta > 0$, und ein $M_2^* \in \mathbb{Z}_{\geq 1}$, so dass für alle $(\xi, x) \in B_\Delta$ und $M \geq M_2^*$ die Lösungen des Systems (3.20), (3.21) die Ungleichung (4.136) erfüllen.

Beweis. Analog zum Beweis von Theorem 4.22 mit den folgenden Änderungen:

- Da (3.7) global exponentiell stabil ist existiert eine Funktion $U(.,.)$, so dass mittels Theorem 7.1 aus der Arbeit von C.M.Kellet [12] gezeigt werden kann, dass (4.110), (4.111) für lokal quadratische Funktionen $\alpha_1, \alpha_2, \alpha_3$ erfüllt sind.

- (iiib) und (iiic) implizieren durch Proposition 4.23, dass α_Q in (4.112) global quadratisch ist.
- Zudem kann man ein beliebiges global quadratisches $\gamma_W \in \mathcal{K}_\infty$ wählen, so dass gilt: $\bar{\alpha}_W, \alpha_W$, definiert in (4.113), sind global quadratisch.

⇒ Alle Bedingungen der Proposition 4.8 gelten für das System (4.10), (4.11) und der Beweis verläuft in denselben Schritten wie der Beweis zu Theorem 4.16. \square

Bemerkung 4.25. *Der modellprädiktive Regelungsansatz kann als Tracking-Problem angesehen werden, da versucht wird die Trajektorie des sampled-data Systems in der Nähe der Trajektorien des zeitkontinuierlichen Systems zu halten. Im Gegensatz zu anderen Verfahren wird hier nicht die Poisson Stabilität des beeinflussbaren Systems angenommen, wodurch die Regelung nicht die bekannte Struktur aufweist.*

Bemerkung 4.26. *Es kann zudem gezeigt werden, dass falls das zeitkontinuierliche System asymptotisch kontrollierbar ist, dann existiert für eine kompakte Menge $N \subset \mathbb{R}^{2n}$ ein $T^* > 0$, $T^* \in \mathbb{R}$, so dass für alle sampled-data Raten $T \in (0, T^*)$ die Lösungen für alle $(\xi, x) \in N$, $u \in \mathcal{U}$ existieren und das sampled-data System kontrollierbar ist auf der Menge N , falls das kontinuierliche System asymptotisch kontrollierbar ist. Dies soll später in der Analyse der numerischen Ergebnisse verwendet werden. Zudem liefert es eine erste Reihenfolge für die Wahl von sampled-data Rate und Horizontlänge, da somit für eine beliebige kompakte Menge N zuerst die sampled-data Rate T angepasst werden sollte, um die Existenz der Lösungen und die asymptotische Kontrollierbarkeit auf N zu gewährleisten, und danach die Horizontlänge zu behandeln, die garantiert, dass N eine Teilmenge des Basin von (3.20), (3.21) ist.*

Kapitel 5

Optimalität

Wie bereits im vorangegangenen Kapitel erwähnt liegen den modellprädiktiven Regelungsverfahren unter anderem die Ideen der Stabilität und der Optimalität zugrunde. Das vorliegende Verfahren ist von der Natur, dass für das zu betrachtende Problem die Regelung *on-line* berechnet wird. Dies geschieht durch Lösen des Optimalsteuerungsproblems mit dem vorliegenden Zustand als Anfangszustand und der Anwendung der ersten berechneten Steuerung aus der Steuerfolge. Der Ansatz stellt folglich keine neue Methode zur Steuerungsdarstellung dar. Im Gegensatz zur Bestimmung einer Feedback-Steuerung weist es jedoch den Vorteil auf, dass hierbei nicht die Lösung der Hamilton-Jacobi-Bellman Differential- oder Differenzgleichung zu bestimmen ist. Letzteres ist im Allgemeinen ein sehr hartnäckiges Problem und Lösungen beschränken sich bisher auf optimale lineare H_2 und H_∞ Steuerungen. Jedoch muss das open-loop Problem in entsprechend kurzer Zeit lösbar sein, und dies wiederum macht es notwendig, einen endlichen Zeithorizont zu verwenden, wodurch wie bereits erläutert weitere Probleme entstehen.

Somit stehen sich hier die Theorien der dynamische Programmierung, auch als Hamilton-Jacobi-Bellman Theorie bekannt, und die des Pontryaginschen Minimumprinzips gegenüber. Die dynamische Programmierung liefert dabei hinreichenden Bedingungen für die Optimalität und stellt einen konstruktiven Ansatz zur Bestimmung der optimalen Feedback-Steuerung $u = u(x)$ bereit, während das Minimumprinzip Aussagen über notwendige Bedingungen für die Optimalität trifft und die Entwicklung von Algorithmen zur Berechnung einer optimalen open-loop Steuerfolge $\hat{u}_{[0,M-1]}$ anregt. Das modellprädiktive Regelungsverfahren verbindet beide dabei in der Art, dass die optimale Feedback-Steuerung mit dem ersten Element der optimalen open-loop Steuerfolge gleichgesetzt wird $u_M = \hat{u}_0 = u(x)$. Die folgende Darstellung orientiert sich an der Zusammenfassung von D.Q.Mayne, J.B.Rawlings, C.V.Rao und P.O.M.Scokaert [19] und stellt anschließend den Bezug zum vorliegenden Verfahren her, für das Ergebnisse aus der Arbeit von D.Nešić und L.Grüne [23] präsentiert werden.

5.1 Inverse Optimalität zeitdiskreter modellprädiktiver Regelungsverfahren

Unter gewissen zusätzlichen Annahmen lässt sich zeigen, dass die durch das Verfahren berechnete Steuerung invers-optimal ist. Dabei unterscheidet man zwei Ansätze, die aber beide davon ausgehen, dass das zu minimierende Zielfunktional J_M eine Lyapunov-Funktion ist. Inverse Optimalität bedeutet dabei, dass das Zielfunktional J_M mit endlichem Horizont einem Zielfunktional eines modifizierten Problems mit unendlichem Horizont entspricht, wodurch dann die Eigenschaften der optimalen Steuerung mit unendlichem Horizont, Stabilität und Robustheit, auch hier zutreffen, falls eine Monotonie-Eigenschaft der Wertefunktion vorliegt. In der Arbeit von L.Magni und R.Sepulchre [15] wird dies für nichtlineare unbeschränkte zeitkontinuierliche Systeme gezeigt.

5.1.1 Direkte Methode

Für jede Funktion $F(x)$ bezeichne $\tilde{F}(x, u)$ die Änderung in $F(\cdot)$ wenn der Zustand von x zu $x^+ = f(x, u)$ verschoben wird:

$$\tilde{F}(x, u) := F(f(x, u)) - F(x) \quad (5.1)$$

Weiter sei $X \subset \mathbb{R}^n$ eine abgeschlossene Menge. Sei für jedes $k \in \mathbb{N}$ die Menge von Zuständen x , die durch zulässige Steuerfolgen in $X_f \subset \mathbb{R}^n$ mit k oder weniger Schritten überführt werden können, durch X_k bezeichnet. Eine zulässige Steuerfolge $u = \{u(0), u(1), \dots, u(k-1)\}$ erfüllt dabei die Steuerbeschränkungen $u(i) \in U \subset \mathbb{R}^m$, U kompakt (und eventuell vorliegende Zustands- und Endzustandsbeschränkungen). Die Zustandsmenge X_M ist die Menge der Anfangszustände x , die durch das Verfahren in M Schritten in X_f gesteuert werden kann. Es soll angenommen werden, dass $u^0(x)$ eine derartige Steuerung ist. Die optimale Zustandstrajektorie wird mit

$$x^0(x) = \{x, x^0(1, x), \dots, x^0(M, x)\} \quad (5.2)$$

bezeichnet.

Ziel ist es, eine zulässige Steuerfolge $\hat{u}(x)$ für x^+ zu finden und damit eine obere Schranke für das Zielfunktional $J_M(x^+)$.

Die Steuerfolge $\{u^0(1, x), \dots, u^0(M-1, x)\}$ überführt x^+ in $x^0(M, x) \in X_f$. Nun wird ein weiteres Element u zu dieser Folge hinzugefügt. Diese neue Folge

$$\{u^0(1, x), \dots, u^0(M-1, x), u\} \quad (5.3)$$

ist dabei zulässig, falls $u \in U$ ist und $x^0(M, x) \in X_f$ in einen Zustand $f(x^0(M, x), u) \in X_f$ überführt. Dies ist der Fall, falls für $u = u_f(x^0(M, x))$, X_f und $u_f(\cdot)$ gilt:

$$X_f \subset X, u_f(x) \in U \text{ und } f(x, u_f(x)) \in X_f \quad \forall x \in X_f, \quad (5.4)$$

und somit X_f positiv invariant unter $u_f(\cdot)$ ist.

$$\Rightarrow \hat{u}(x) := \{u^0(1, x), \dots, u^0(M-1, x), u_f(x^0(M, x))\} \text{ ist zulässig.} \quad (5.5)$$

Die zugehörige Zustandstrajektorie für den Anfangszustand $x^+ = x^0(1, x)$ mit Steuerfolge $u(x)$ ist demnach gegeben durch

$$\hat{x}(x) = \{x^0(1, x), \dots, x^0(M, x), f(x^0(M, x), u_f(x^0(M, x)))\}. \quad (5.6)$$

Für die Wertefunktion gilt folglich

$$\begin{aligned} V_M(x^+, \hat{u}(x)) &= V_M(x) - Q(x, u_M(x)) - F(x^0(M, x)) \\ &\quad + Q(x^0(M, x), u_f(x^0(M, x))) + F(f(x^0(M, x), u_f(x^0(M, x)))). \end{aligned} \quad (5.7)$$

Somit ist eine obere Schranke für $V_M(x^+)$ gegeben und die Ungleichung

$$V_M(x^+, \hat{u}(x)) \leq V_M(x) - Q(x, u_M(x)) \quad (5.8)$$

ist erfüllt, falls $[\tilde{F} + Q](x, u_f(x)) \leq 0$ für alle $x \in X_f$ gilt. Dies ist der Fall, wenn $F(\cdot)$ eine Lyapunov-Kontroll-Funktion in einer Umgebung des Ursprungs ist und $u_f(\cdot)$ und X_f entsprechend gewählt werden. Trifft dies zu, so gilt

$$\tilde{V}_M(x, u_M(x)) + Q(x, u_M(x)) \leq 0 \quad (5.9)$$

für alle $x \in X_M$. Dies ist eine hinreichende Bedingung für die Konvergenz des Zustandes des closed-loop Systems $x^+ = f(x, u_M(x)) \rightarrow 0$ für $k \rightarrow \infty$, falls $x \in X_M$.

Es lassen sich also folgende Bedingungen für die asymptotische Stabilität des closed-loop Systems aufstellen:

(A1) $X_f \in X$, X_f ist abgeschlossen, $0 \in X_f$ (Zustandsbeschränkungen in X_f erfüllt)

(A2) $u_f(x) \in U \quad \forall x \in X_f$ (Steuerbeschränkungen in X_f erfüllt)

(A3) $f(x, u_f(x)) \in X_f \quad \forall x \in X_f$ (X_f positiv invariant unter $u_f(x)$)

(A4) $[\tilde{F} + Q](x, u_f(x)) \leq 0 \quad \forall x \in X_f$ (lokale Lyapunov-Funktion)

5.1.2 Monotonie-Methode

Nach dem Optimalitätsprinzip gilt für $x^+ = f(x, u_M(x))$:

$$\begin{aligned} V_M(x) &= Q(x, u_M(x)) + V_{M-1}(x^+) \quad \forall x \in X_M \\ \Rightarrow V_M(x) &= Q(x, u_M(x)) + V_M(x^+) + [V_{M-1}(x^+) - V_M(x^+)] \end{aligned} \quad (5.10)$$

und somit ergibt sich

$$\tilde{V}_M(x, u_M(x)) + Q(x, u_M(x)) = [V_M(x^+) - V_{M-1}(x^+)]. \quad (5.11)$$

Es sollen die Bedingungen (A1) – (A4) gelten. Um (5.9) zu erhalten fügt man zur optimalen Steuerfolge, bestehend aus $M - 1$ Gliedern, für das Problem mit Anfangswert x^+ eine weitere Steuerung $u_f(x^0(N - 1), x^+) = u_f(x^0(N, x))$ hinzu und erhält die Folge $\hat{u}(x)$, die der Folge aus der direkten Methode entspricht. Weiter gilt

$$\begin{aligned} V_M(x^+, \hat{u}(x)) &= V_{M-1}(x^+) + Q(x^0(N, x), u_f(x^0(N, x))) - F(x^0(N, x)) \\ &\quad + F(f(x^0(N, x)), u_f(f(x^0(N, x)))) \geq V_M(x^+). \end{aligned} \quad (5.12)$$

Aus (A4) folgt nun $V_M(x^+) \leq V_{M-1}(x^+)$, woraus sich mit (5.11) die Ungleichung (5.9) ergibt.

5.1.3 Der zeitdiskrete Fall

Für den zeitdiskreten Fall kann man die Gleichung (5.11) umschreiben zu

$$V_M(x) = \bar{Q}(x, u_M(x)) + V_M(f(x, u_M(x))) \quad (5.13)$$

für alle $x \in X_M$, wobei man \bar{Q} festlegt durch

$$\bar{Q}(x, u) := Q(x, u) + [V_{M-1} - V_M](f(x, u_M(x))). \quad (5.14)$$

Falls die Bedingungen (A1) – (A4) gelten, so folgt

$$\bar{Q}(x, u) \geq Q(x, u) \geq c\|(x, u)\|^2. \quad (5.15)$$

Dann ist (5.13) eine algebraische Hamilton-Jacobi-Bellman Gleichung für das Optimalsteuerungsproblem mit unendlichem Horizont, wobei $Q(\cdot)$ durch $\bar{Q}(\cdot)$ zu ersetzen ist. Da u_M die optimale Steuerung für das modifizierte Problem ist werden die Eigenschaften der optimalen Steuerung mit unendlichem Horizont auf die optimale Steuerung mit endlichem Zeithorizont vererbt.

5.2 Optimalität des modellprädiktiven Regelungsansatzes

Um die Optimalität des vorliegenden Verfahrens zu zeigen werden folgende Annahmen getroffen:

Annahme 5.1. Gegeben sei das Problem (3.11) unter (3.12), (3.13), (3.14), (3.15). Sei F derart, dass eine abgeschlossene Teilmenge $X_f \subset \mathbb{R}^{2n}$ und eine Steuerung $u = u_f(\xi, x)$ existieren, so dass die folgenden Bedingungen gelten:

- (i) $u_f(\xi, x) \in U \quad \forall (\xi, x) \in X_f$
- (ii) $(\xi, x) \in X_f \Rightarrow (G(\xi, u_f(\xi, x)), H(x)) \in X_f$

(iii) $\forall(\xi, x) \in X_f :$

$$F(G(\xi, u_f(\xi, x)), H(x)) - F(\xi, x) \leq -Q(\xi, x, u_f(\xi, x)).$$

Dabei werden diese durch die Annahmen (A1) - (A4) impliziert.

Beweis. Wähle dazu

- Für x aus (A1) - (A4) gelte $x = (\xi, x)$
- Für u_f aus (A1) - (A4) sei $u_f(x) = u_f(\xi, x) \Rightarrow$ (i)
- Somit gilt $x^+ = f(x, u) = (G(\xi, u), H(x)) \Rightarrow$ (ii)

Zudem gelte weiterhin $\tilde{F}(x, u) := F(f(x, u)) - F(x) \Rightarrow$ (iii) aus (A4) □

Theorem 5.2. Gegeben sei das zeitdiskrete Modell (4.3), (4.4). Gelten zudem die oben aufgeführten Bedingungen, dann existiert eine Menge $X_M \subset \mathbb{R}^{2n}$ und eine Funktion $\Phi : \mathbb{R}^n \times \mathbb{R}^n \times U \rightarrow \mathbb{R}$ mit

$$\Phi(\xi, x, u_M) \geq Q(\xi, x, u_M) \quad \forall(\xi, x) \in X_M, u \in U, \quad (5.16)$$

so dass für alle $(\xi, x) \in X_M$ die Steuerung (3.19) das Zielfunktional

$$\Upsilon(\xi, x, u_{[0, \infty]}) := \sum_{i=0}^{\infty} \Phi(\xi_i, x_i, u_i) \quad (5.17)$$

minimiert.

Beweis. Ausgehend vom Optimalitätsprinzip gilt für die zum modellprädiktiven Regelungsansatz gehörende Wertefunktion

$$\begin{aligned} V_M(\xi, x) &= Q(\xi, x, u_M(\xi, x)) + V_{M-1}(G(\xi, u_M(\xi, x)), H(x)) = \\ &= V_M(G(\xi, u_M(\xi, x)), H(x)) + Q(\xi, x, u_M(\xi, x)) + \\ &\quad + [V_{M-1}(G(\xi, u_M(\xi, x)), H(x)) - V_M(G(\xi, u_M(\xi, x)), H(x))] = \\ &= V_M(G(\xi, u_M(\xi, x)), H(x)) + \Phi(\xi, x, u_M(\xi, x)). \end{aligned} \quad (5.18)$$

Dabei wurde eine Umformung auf das unendliche Intervall vorgenommen. Wenn gezeigt werden kann, dass die Monotonie-Eigenschaft

$$V_{M-1}(G(\xi, u_M(\xi, x)), H(x)) - V_M(G(\xi, u_M(\xi, x)), H(x)) \geq 0 \quad (5.19)$$

gilt, dann kann gefolgert werden, dass die Steuerung $u_M(\xi, x)$ optimal für das neudefinierte Zielfunktional ist, wobei

$$\Phi(\xi, x, u) := Q(\xi, x, u) + [V_{M-1}(G(\xi, u), H(x)) - V_M(G(\xi, u), H(x))] \quad (5.20)$$

gesetzt wird. Es gelte:

- X_M ist derart, dass gilt: $(\xi, x) \in X_M \Rightarrow (\xi(t_M), x(t_M)) \in X_f$
- Sei $(\xi, x) \in X_M$ beliebig aber fest.
- Es sei $\hat{u}_{[0, M-1]} = \hat{u}_{[0, M-1]}(\xi, x)$ die optimale Steuerung für $J_M(\xi, x, v_{[0, M-1]})$.
- $\xi(t, \xi, u_{[0, M-1]})$ bezeichne die optimale Zustandstrajektorie des sampled-data Systems.
- Die Steuerung $u = \hat{u}_0(\xi, x)$ überführt den Anfangszustand ξ in seinen Nachfolgezustand $\xi^+ = \xi(T, \xi, \hat{u}_0(\xi, x)) =: G(\xi, \hat{u}_0(\xi, x))$.
- Weiter bezeichne:

$$x^+ = x(T, x) =: H(x), \quad (5.21)$$

$$x_f := x(t_M, x), \quad (5.22)$$

$$\xi_f := \xi(t_M, \xi, \hat{u}_{[0, M-1]}). \quad (5.23)$$

Da $\hat{u}_{[0, M-1]} = \{\hat{u}_0, \dots, \hat{u}_{M-1}\}$ die optimale Steuerung darstellt, $(\xi_f, x_f) \in X_f$ ist und zudem $\{\hat{u}_0, \dots, \hat{u}_{M-1}\}$ die Folge ist, die ξ^+ in ξ_f überführt, kann man ein weiteres Element zu dieser Folge hinzufügen, um eine weitere suboptimale Steuerung $\tilde{u} := \{\hat{u}_0, \dots, \hat{u}_{M-1}, u_f(\xi, x)\}$ für den Zustand (ξ^+, x^+) zu erhalten. Hierbei ergibt sich $u_f(\cdot, \cdot)$ aus den Annahmen 5.1 zur Steuerung.

Aus Annahme 5.1 (ii) folgt somit $(G(\xi_f, u_f(\xi_f, x_f)), H(x_f)) \in X_f$. Die Anwendung von Annahme 5.1 (iii) ermöglicht dann die Abschätzung

$$\begin{aligned} J_M(\xi^+, x^+, \tilde{u}) &= V_{M-1}(G(\xi, \hat{u}_0), H(x)) + Q(\xi_f, x_f, u_f(\xi_f)) - \\ &\quad - F(\xi_f, x_f) + F(G(\xi_f, u_f(\xi_f)), H(x_f)) \\ &\geq V_M(G(\xi, \hat{u}_0), H(x)) \quad \forall (\xi, x) \in X_M. \end{aligned} \quad (5.24)$$

Da $u_M(\xi, x) := u_0(\xi, x)$ ist kann dies umformuliert werden zu

$$V_{M-1}(G(\xi, u_M(\xi, x)), H(x)) \geq V_M(G(\xi, u_M(\xi, x)), H(x)) \quad (5.25)$$

und die Behauptung ist gezeigt. \square

Zudem lässt sich zeigen, dass wenn F eine lokale Lyapunov-Kontroll-Funktion für das closed-loop sampled-data System (4.10), (4.11) ist, dann kann eine Funktion l gefunden werden, so dass die Annahmen 5.1 erfüllt sind:

Proposition 5.3. *Seien $u_f(\xi, x)$ und $F(\xi, x)$ gegeben, die die Bedingungen (i) und (ii) aus Annahme 5.1 erfüllen. Zudem existiere eine Funktion $\alpha_u \in \mathcal{K}_\infty$ mit*

$$\|u_f(\xi, x)\| \leq \alpha_u(\|(\xi, x)\|) \quad \forall (\xi, x) \in X_f. \quad (5.26)$$

Weiter gelte für alle $(\xi, x) \in X_f$

$$\alpha_1(\|(\xi, x)\|) \leq F(\xi, x) \leq \alpha_2(\|(\xi, x)\|) \quad (5.27)$$

$$\Delta F := F(G(\xi, u_f(\xi, x)), H(x)) - F(\xi, x) \leq -\alpha_3(\|(\xi, x)\|) \quad (5.28)$$

für Funktionen $\alpha_1, \alpha_2, \alpha_3 \in \mathcal{K}_\infty$. Wenn $L \in \mathbb{R}$, $L > 0$ für $(f(\xi, u), f(x, u(x)))$ auf der Menge $X_f \times U$ eine Lipschitz-Konstante ist, dann gilt die Bedingung (iii) aus Annahme 5.1 für alle Funktionen $l : \mathbb{R}^{2n} \times U \rightarrow \mathbb{R}_0^+$, so dass

$$l(y, u) \leq \frac{1}{T} \alpha_3 \circ \alpha \left(\frac{e^{(-LT)} \|y\|}{2} + \frac{\|u\|}{2} \right) \quad (5.29)$$

mit $\alpha(s) := \frac{1}{2} \min\{\frac{s}{2}, \alpha_u^{-1}(\frac{s}{2})\}$ erfüllt ist.

Beweis. Sei $L > 0$ die Lipschitz-Konstante für das zusammengesetzte Differentialgleichungssystem $(f(\xi, u), f(x, u(x)))$ auf der Menge $X_f \times U$. Aus (5.28) folgt daher mit (5.26):

$$\begin{aligned} \Delta F &\leq -\alpha_3(\|\xi\|_1 + \|x\|_1) = -\alpha_3 \left(\frac{\|\xi\|_1 + \|x\|_1}{2} + \frac{\|(\xi, x)\|_1}{2} \right) \\ &\leq -\alpha_3 \left(\frac{\|\xi\|_1 + \|x\|_1}{2} + \frac{1}{2} \alpha_u^{-1}(\|u_f(\xi, x)\|_1) \right) \end{aligned} \quad (5.30)$$

Verwendet man weiter die Dreiecksungleichung für \mathcal{K} -Funktionen $\alpha(\frac{1}{2}(s_1 + s_2)) \leq \alpha(s_1) + \alpha(s_2)$, so gilt:

$$\Delta F \leq -\alpha_3 \circ \alpha(\|\xi\|_1 + \|x\|_1 + \|u_f(\xi, x)\|_1). \quad (5.31)$$

Aus der Invarianz der Menge X_f folgt für alle $(\xi, x) \in X_f$:

$$\begin{aligned} \|\xi(t)\|_1 + \|x(t)\|_1 &\leq e^{(LT)}(\|\xi\|_1 + \|x\|_1) + (e^{(LT)} - 1)\|u_f(\xi, x)\|_1 \\ &\leq e^{(LT)}(\|\xi\|_1 + \|x\|_1 + \|u_f(\xi, x)\|_1) \quad \forall t \in [0, T]. \end{aligned} \quad (5.32)$$

Verwendet man dies nun zusammen mit (5.31), so folgt:

$$\begin{aligned} \Delta F &\leq -\frac{1}{T} \int_0^T \alpha_3 \circ \alpha(\|\xi\|_1 + \|x\|_1 + \|u_f(\xi, x)\|_1) dt \\ &\leq -\frac{1}{T} \int_0^T \alpha_3 \circ \alpha \left(e^{(-LT)} \frac{\|\xi(t)\|_1 + \|x(t)\|_1}{2} + \frac{\|u_f(\xi, x)\|_1}{2} \right) dt \\ &\leq -\frac{1}{T} \int_0^T \alpha_3 \circ \alpha \left(e^{(-LT)} \frac{\|\xi(t) - x(t)\|_1}{2} + \frac{\|u_f(\xi, x)\|_1}{2} \right) dt \\ &\leq -\int_0^T l(\xi(t) - x(t), u_f(\xi, x)) dt \\ &=: -Q(\xi, x, u_f(\xi, x)). \end{aligned} \quad (5.33)$$

□

Gleichzeitig ist es aber nicht notwendig, dass F eine lokale Lyapunov-Kontroll-Funktion sein muss um die Annahmen 5.1 zu erfüllen. Sei dazu etwa $l(y, u) \geq \gamma(\|y\|)$ für ein $\gamma \in \mathcal{K}$. Dann folgt aus Proposition 4.18 unter der zusätzlichen Annahme $\xi \neq x$, dass $Q(\xi, x, u) > 0$ ist. Daraus ergibt sich

$$\xi, x \in X_f, \xi \neq x \Rightarrow F(\xi, x) > 0. \quad (5.34)$$

Würden nun die Bedingungen aus Annahme 5.1 gelten, dann existieren $\xi^*, x^* \in X_f, \xi^* \neq x^*$, so dass $F(\xi^*, x^*) = 0$ ist.

$$\Rightarrow F(G(\xi^*, x^*), H(x^*)) \leq -Q(\xi^*, x^*, u_f(\xi^*, x^*)) < 0 \quad (5.35)$$

Dies stellt einen Widerspruch zur positiven Semidefinitheit von $F(.,.)$ dar und es lässt sich aus der Bedingung (iii) der Annahme 5.1 folgern:

- $F(0, 0) = 0$
- $\xi, x \in X_f, \xi \neq x \Rightarrow F(\xi, x) > 0$
- $\xi, x \in X_f, \xi \neq x \Rightarrow F(G(\xi, x), H(x)) - F(\xi, x) < 0$

Im Allgemeinen ist F somit keine lokale Lyapunov-Kontroll-Funktion des exakten zeitdiskreten closed-loop sampled-data Systems (4.10), (4.11). $U(x) + F(\xi, x)$ allerdings ist eine solche Lyapunov-Kontroll-Funktion für das System, wenn $U(x)$ die Bedingungen (4.110), (4.111) erfüllt.

Um Funktionen F und l zu finden, die den Annahmen 5.1 genügen, bietet sich folgende Proposition an. Diese nutzt umgekehrte Lyapunov Theoreme für die einzelnen Subsysteme aus, um eine additiv zusammengesetzte Lyapunov-Funktion

$$F(\xi, x) := V_1(\xi) + V_2(x) \quad (5.36)$$

aus den erhaltenen Lyapunov-Funktionen $V_1(\xi)$ und $V_2(x)$ zu bestimmen und greift dann auf Proposition 5.3 zurück.

Proposition 5.4. *Es sei die sampled-data Rate $T \in \mathbb{R}, T > 0$ derart gegeben, so dass das sampled-data Subsystem $\dot{\xi}$ stabilisierbar ist. Wenn zudem eine Steuerung $u_f(\xi)$ vorliegt, so dass*

$$\xi^+ = G(\xi, u_f(\xi)) \quad (5.37)$$

asymptotisch stabil ist, und ein $\alpha_u \in \mathcal{K}_\infty$ existiert, für das die Ungleichung (5.26) gilt, dann existiert eine Funktion F und die Annahmen 5.1 sind für eine beliebige Funktion l erfüllt, die der Ungleichung (5.29) genügt.

Kapitel 6

Beispiele und Ergebnisse

Das entwickelte Programm zur modellprädiktiven Regelung soll an bekannten Beispielen getestet werden. Die Beispiele 6.1, 6.2 und 6.3 sind daher aus Kapitel 2.4 des Buchs von M.Krstić, I.Kanellakopoulos und P.Kokotović [14] sowie der Arbeit von D.Nešić und L.Grüne [22] entnommen.

6.1 1D Beispiel

In dem ersten Beispiel sollen das eindimensionale Differentialgleichungssystem und die stabilisierende zeitkontinuierliche Steuerung durch

$$\dot{x} = x^3 + u \quad (6.1)$$

$$u(x) = -x^3 - x\sqrt{x^4 + 1} \quad (6.2)$$

gegeben sein. Mit

$$V(x) = \frac{x^2}{2} \quad (6.3)$$

ist zudem eine Lyapunov-Funktion für (6.1) vorgegeben, die für die Lösung aber nicht notwendigerweise im Zielfunktional verwendet werden muss.

Es soll nun einmal explizit die Darstellung des modellprädiktiven Regelungsproblems erfolgen:

Will man die modellprädiktive Steuerung auf dem Intervall $[0, 10]$ berechnen für $T = t_{k+1} - t_k = 1.0$, so sind 10 Optimalsteuerungsprobleme zu lösen. Weiter werde $M = 5$ gewählt, so dass sich eine Horizontlänge von $H = t_M^j - t_0^j = M \cdot T = 5.0$ ergibt. Die äquidistante Diskretisierung der Optimalsteuerungsprobleme auf den Intervallen $[t_0^j, t_M^j]$ soll zudem wieder der verlangten Unterteilung der modellprädiktiven Steuerung entsprechen, daher gilt:

$$t_0^1 = 0, \quad t_M^1 = t_0^1 + M \cdot T, \quad t_i^j = t_i^{j-1} + T, \quad t_{i+1}^j = t_i^j + T \quad \text{für } i = 1, \dots, 5, \quad j = 1, \dots, 10$$

Für die Folge von Optimalsteuerungsproblemen $(OCP_j)_{j=1,\dots,10}$

$$\text{Finde } \hat{u}_{[0,4]} = \arg \inf_{v_{[0,4]}} J_5(\xi(t), x(t), v_{[0,4]})$$

unter

$$\begin{aligned} J_5(\xi(t), x(t), v_{[0,4]}) &:= \int_{t_0^j}^{t_M^j} \frac{(\xi(t) - x(t))^2}{2} dt + \frac{(\xi(t_M) - x(t_M))^2}{2} \\ v_i &\in \mathbb{U} \quad \forall i \in \{0, \dots, 4\} \\ t_i^j &:= (i + j - 1) \cdot 1.0 \quad , i \in \{0, \dots, 5\}, j \in \{1, \dots, 10\} \\ \dot{\xi}(t) &= \xi(t)^3 + v_i, \quad \xi(t_0) = \xi_j, \quad t \in [t_i^j, t_{i+1}^j) \\ \dot{x}(t) &= -x(t)\sqrt{x(t)^4 + 1}, \quad x(t_0^j) = x_j, \end{aligned}$$

mit Anfangswerten $(x, \xi)^T = (0.01, 0.01)^T$ werden die Werte

$$\begin{pmatrix} x(j * 1.0) \\ \xi(j * 1.0) \\ \hat{u}_{[0]} \end{pmatrix} =: \begin{pmatrix} x_j \\ \xi_j \\ u_{j-1} \end{pmatrix}$$

gespeichert, so dass die Folgen

$$(x_j)_{j=0,\dots,10}, \quad (\xi_j)_{j=0,\dots,10}, \quad (u_j)_{j=0,\dots,9}$$

für dieses Beispiel die Abbildung 6.1 ergeben. Zum Vergleich wurde die gesamplete Lösung dieses Beispiels für dieselbe sampled-data Rate T hinzugefügt. Diese berechnet sich aus der Lösung des Systems

$$\dot{x}(t) = x(t)^3 + u(t) \tag{6.4}$$

$$u(t) = -x(t_k)^3 - x(t_k)\sqrt{x(t_k)^4 + 1}, \quad t \in [t_k, t_{k+1}) \tag{6.5}$$

$$t_k = k \cdot T, \quad k \in \{0, \dots, 10\}. \tag{6.6}$$

Bemerkung 6.1. *Generell empfiehlt es sich dabei den optimalen Steuerprozess in Mayer-Form zu transformieren, da hierbei die Integration des Lagrange-Anteils in den Nebenbedingungen vorgenommen wird, für die die implementierte Schrittweitensteuerung wesentlich bessere Ergebnisse liefert als die Integration auf dem verwendeten Diskretisierungsgitter der Stufe 0 mit Schrittweite $h = T$.*

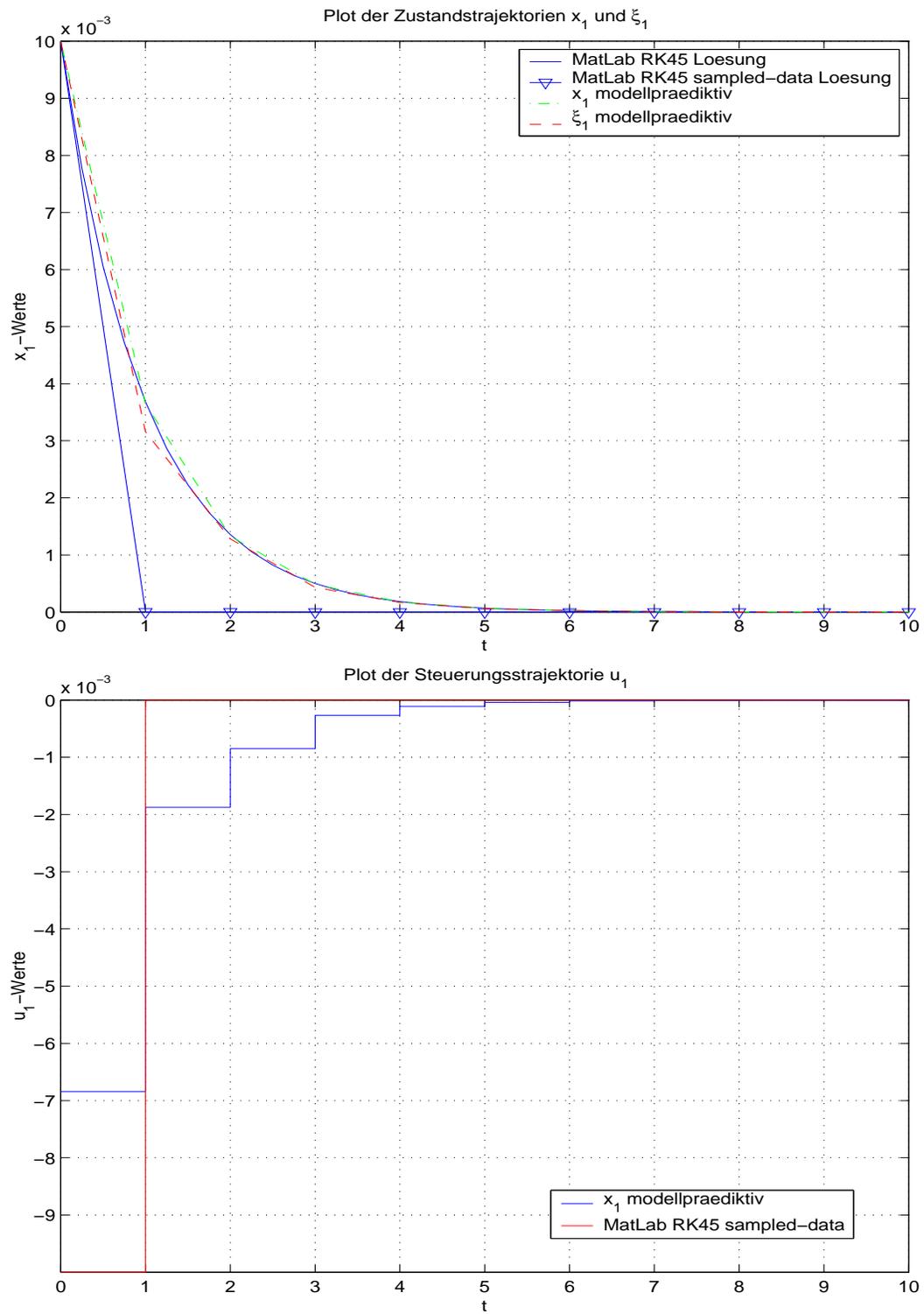


Abbildung 6.1: Musterlösung des 1D-Beispiels

Einschub: Moore-Greitzer Modell eines Strahltriebwerks

Hierbei soll eine Galerkin-Approximation des Modells von Moore und Greitzer untersucht werden, das in seiner ursprünglichen Form aus einer partiellen Differentialgleichung und zwei gewöhnlichen Differentialgleichungen besteht

$$\frac{\partial \varphi}{\partial t} = \nu \frac{\partial^2 \varphi}{\partial \theta^2} - \frac{1}{2} \frac{\partial \varphi}{\partial \theta} + \psi_c(\Phi + \varphi) - \psi_c, \quad \theta \in [0, 2\pi] \quad (6.7)$$

$$\frac{d\Phi}{dt} = \frac{1}{l_c}(\psi_c - \Psi) \quad (6.8)$$

$$\frac{d\Psi}{dt} = \frac{1}{4l_c B_2}(\Phi - \gamma F_T^{-1}(\Psi)), \quad (6.9)$$

wobei gilt

$$\psi_C := \frac{1}{2\pi} \int_0^{2\pi} \psi_c(\Phi(t) + \varphi(t, \theta)) d\theta. \quad (6.10)$$

Die Galerkin-Approximation ergibt das System

$$\dot{\Phi} = -\Psi + \Psi_C(\Phi) - 3\Phi R \quad (6.11)$$

$$\dot{\Psi} = \frac{1}{\beta^2}(\Phi - \Phi_T) \quad (6.12)$$

$$\dot{R} = \sigma R(1 - \Phi^2 - R), \quad R(0) \geq 0 \quad (6.13)$$

mit den Eigenschaften von Kompressor und Schub, $\Psi_C(\Phi)$ und $\Phi_T(\Psi)$, die durch

$$\Psi_C(\Phi) = \Psi_{C0} + 1 + \frac{3}{2}\Phi - \frac{1}{2}\Phi^3 \quad (6.14)$$

$$\Psi = \frac{1}{\gamma}(1 + \Phi_T(\Psi))^2 \quad (6.15)$$

charakterisiert sind. Dabei entspricht Φ der Durchflußrate, Ψ der Leistungszunahme des Triebwerks und $R \geq 0$ stellt die normierte quadrierte Größe der Abrissströmungszelle dar. Ψ_{C0} ist in diesem Gleichungssystem eine Konstante und γ kann durch Änderung der Schuböffnung festgelegt werden. Zudem beschreibt Φ_T die Schubdurchflußrate und σ, β sind positive Konstanten.

Das Hauptziel bei der Untersuchung des Moore-Greitzer Modells ist die Vermeidung zweier Instabilitätstypen:

- rotierende Abrissströmungen
- Druckstöße

Im Buch von M.Krstic, I.Kanellakopoulos und P.Kokotovic [14] wird hierzu eine Verschiebung des Ursprungs zu einem gewünschten Gleichgewicht mittels $\phi = \Phi - 1$ und $\psi = \Psi - \Psi_{C0} - 2$ durchgeführt und die Steuerung

$$u = \frac{1}{\beta^2}(\Phi_T - 1 - \phi) \quad (6.16)$$

gewählt. Damit stellt sich das Kontrollsystem wie folgt dar:

$$\dot{\phi} = -\psi - \frac{3}{2}\phi^2 - \frac{1}{2}\phi^3 - 3R\phi - 3R \quad (6.17)$$

$$\dot{\psi} = -u \quad (6.18)$$

$$\dot{R} = -\sigma R(R + 2\phi + \phi^2) \quad (6.19)$$

6.2 Vereinfachtes Moore-Greitzer Modell

Unter der Annahme, dass das Triebwerk nicht absterben kann, lässt sich das Modell durch die Differentialgleichungen

$$\dot{\phi} = -\psi - \frac{3}{2}\phi^2 - \frac{1}{2}\phi^3 \quad (6.20)$$

$$\dot{\psi} = -u \quad (6.21)$$

darstellen. Für die Bestimmung eines Zustandsfeedbacks sei auf Kapitel 2.4 des erwähnten Buchs [14] verwiesen.

Die zeitkontinuierliche Steuerung

$$u(x) = -k_1\phi + k_2\psi \quad (6.22)$$

und die Lyapunov-Funktion

$$V(x) = \frac{1}{2}\phi^2 + \frac{c_0}{8}\phi^4 + \frac{1}{2}(\psi - c_0\phi)^2 \quad (6.23)$$

wurden dort hergeleitet, wobei für die Parameter

$$k_1 = 1 + c_0c_2 + \frac{9c_0^2}{8c_1} \quad (6.24)$$

$$k_2 = c_2 + c_0 + \frac{9c_0}{8c_1} \quad (6.25)$$

$$c_0 = c_1 + \frac{9}{8} \quad (6.26)$$

mit $c_1, c_2 > 0$ die Werte $c_1 = \frac{7}{8}$ und $c_2 = \frac{3}{7}$ gewählt wurden. Damit ergeben sich die zeitkontinuierliche Steuerung und die Lyapunov-Funktion zu

$$u(x) = -7\phi + 5\psi \quad (6.27)$$

$$V(x) = \frac{1}{2}\phi^2 + \frac{1}{4}\phi^4 + \frac{1}{2}(\psi - 2\phi)^2. \quad (6.28)$$

und es gelten die Differentialgleichungen

$$\dot{\phi} = -\psi - \frac{3}{2}\phi^2 - \frac{1}{2}\phi^3 \quad (6.29)$$

$$\dot{\psi} = 7\phi - 5\psi, \quad (6.30)$$

die im Folgenden zusammen mit (6.20), (6.21) als Beispiel 6.2 bezeichnet werden, wobei die Werte

$$\begin{pmatrix} \phi \\ \psi \end{pmatrix} = \begin{pmatrix} 22 \\ 21 \end{pmatrix}$$

als Anfangswerte verwendet werden sollen.

6.3 Moore-Greitzer Modell eines Strahltriebwerks

Zur Generierung eines Zustandsfeedbacks wurde für ψ als virtuelle Steuerung der $\dot{\phi}$ -Gleichung die Wahl

$$\alpha(\phi, R) := c_1\phi - \frac{3}{2}\phi^2 - 3R \quad (6.31)$$

getroffen, um die nützlichen nichtlinearen Anteile $\frac{1}{2}\phi^3$ und $-3R\phi$ nicht aus der Gleichung zu entfernen. Für eine Rechtfertigung dieser Wahl sei auf das 3-Schritt-Design verwiesen, das in Kapitel 2 des Buchs von M.Krstic, I.Kanellakopoulos und P.Kokotovic [14] beschrieben ist. Durch die Substitution

$$\tilde{\psi} = \psi - \alpha(\phi, R) \quad (6.32)$$

ergibt sich das dynamische System

$$\dot{\phi} = -\tilde{\psi} - c_1\phi - \frac{1}{2}\phi^3 - 3R\phi \quad (6.33)$$

$$\dot{\tilde{\psi}} = \phi - c_2\tilde{\psi} \quad (6.34)$$

$$\dot{R} = -\sigma R (R + 2\phi + \phi^2) \quad (6.35)$$

für die Wahl der Steuerung gemäß

$$\begin{aligned} u &= c_2\tilde{\psi} - \phi - \frac{\partial\alpha}{\partial\phi}\dot{\phi} - \frac{\partial\alpha}{\partial R}\dot{R} \\ &= c_2\tilde{\psi} - \phi - (c_1 - 3\phi) \left(-\tilde{\psi} - c_1\phi - \frac{1}{2}\phi^3 - 3\phi R \right) \\ &\quad - 3\sigma R (R + 2\phi + \phi^2). \end{aligned} \quad (6.36)$$

Zur einfacheren Implementierung soll hier das Problem resubstituiert werden. Mittels $\psi = \tilde{\psi} + \alpha(\phi, R)$ erhält man wieder das Kontrollsystem (6.17) - (6.19), das hier als Beispiel 6.3

bezeichnet werden soll.

Die zeitkontinuierliche Steuerung hat nach der Resubstitution folgende Gestalt:

$$\begin{aligned}
 u &= c_2 \tilde{\psi} - \phi - \frac{\partial \alpha}{\partial \phi} \dot{\phi} - \frac{\partial \alpha}{\partial R} \dot{R} \\
 &= c_2 \left(\psi - c_1 \phi + \frac{3}{2} \phi^2 + 3R \right) - \phi - 3\sigma R (R + 2\phi + \phi^2) \\
 &\quad - (c_1 - 3\phi) \left(-\psi - \frac{3}{2} \phi^2 - \frac{1}{2} \phi^3 - 3\phi R - 3R \right)
 \end{aligned} \tag{6.37}$$

In der numerischen Berechnung werden für die Anfangswerte und die Parameter die Werte

$$\begin{pmatrix} \phi \\ \psi \\ R \end{pmatrix} = \begin{pmatrix} 6 \\ 25 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} c_1 \\ c_2 \\ \sigma \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}$$

gewählt.

6.4 Reihenfolge zur Anpassung der Einflußfaktoren

Um mittels des hier dargelegten modellprädiktiven Regelungsansatzes für beliebige Beispiele eine Lösung zu erhalten, empfiehlt es sich die Einflußfaktoren in der folgenden Reihenfolge festzulegen:

- (1) Anfangswerte $(x, \xi) \in \mathbb{R}^{2n}$
- (2) Parameteranpassung (falls möglich)
- (3) sampled-data Rate $T \in \mathbb{R}$
- (4) Horizontlänge $H \in \mathbb{R}$
- (5) Wahl des Integrationsverfahrens
- (6) Fehlertoleranz der Integration

Hierzu sei auf Bemerkung 4.26 verwiesen, welche die Grundidee verdeutlicht.

Ratsam ist dabei für eine gute Ausgangsbasis die Lösung des ungesteuerten Problems zu betrachten. Dies ist durch Ersetzen der Steuerung durch die zeitkontinuierliche Feedback-Steuerung leicht möglich. Unter einer guten Ausgangsbasis versteht man dabei Trajektorien, die

- (a) keine starken Steigungen aufweisen und
- (b) monoton innerhalb eines Intervalls der Länge T des äquidistanten Gitters sind.

Da es sich bei (a) um eine vom Verfahren unabhängige Eigenschaft handelt sollte dies zuerst bedacht und so (1) und (2) festgelegt werden. Im Anschluß kann man die Faktoren (3) und (4) unter Berücksichtigung von (b) bestimmen. Für die Fehlertoleranz (6) haben sich der Wert 10^{-4} und für das Integrationsverfahren (5) die Runge-Kutta Verfahren der Ordnung 4 und 5 als brauchbar erwiesen.

Der Reihenfolge liegt dabei die Logik zugrunde, zunächst eine Lösung des optimalen Steuerungsproblems berechnen zu können und anschließend die sampled-data Rate zu vergrößern.

- So ist es zunächst nötig im Schritt (1) Anfangswerte zu finden, für die das Kontrollsystem mit Parametern (2) stabilisierbar ist.
- Da für die Steuerung nur stückweise konstante Steuerfunktionen verwendet werden, die zudem zu einer bestimmten sampled-data Rate (3) gehören, wird T zunächst klein gewählt.
- Während der Auswertung der numerischen Ergebnisse hat sich gezeigt, dass längere Horizonte bessere Ergebnisse liefern. Somit sollte H in (4) groß gewählt werden.
- Nun wird die sampled-data Rate T vergrößert. Dadurch werden numerische Probleme verstärkt, etwa bei der Diskretisierung, der Integration und der Optimierung. Insbesondere die Länge des Horizonts spielt dabei eine Rolle, so dass dieser eventuell verkleinert werden muss.
- Mittels (5) und (6) kann nur noch versucht werden, die Grenzen des Verfahrens für ein vorliegendes Problem zu erweitern.

Bemerkung 6.2. (2) kann nur bei einer möglichen Parameterwahl angewandt werden und im Fall von (5) und (6) ist eine Vergrößerung der sampled-data Rate lediglich auf Kosten der Fehlerordnung beziehungsweise der Genauigkeit möglich.

Einzelne Auswirkungen sollen nun anhand der drei zuvor genannten Beispiele erörtert werden.

6.5 Sensitivität der Lösung bei Änderung der Anfangswerte

Da bei Beispiel 6.1 nur ein eindimensionaler Anfangswert $x_0 \in \mathbb{R}$ vorliegt soll diese Untersuchung ausschließlich an diesem Beispiel vollzogen werden. Als Fehler wird dabei generell die Abweichung der gesteuerten sampled-data Lösung von der zeitkontinuierlich gesteuerten Lösung betrachtet.

$$\text{Fehler} := \sum_i |\xi_i - x_i| \tag{6.38}$$

Bemerkung 6.3. Die zeitkontinuierliche Lösung wird dabei hier und im Folgenden mit Matlab berechnet.

Der maximale Fehler kann daher nur für die diskreten Punkten der sampled-data Folge aus den vorliegenden Werten berechnet werden. Verwendet wurden dabei eine sampled-data Rate

x_0	Fehler bei $t = 1$	Fehler bei $t = 2$	Fehler bei $t = 10$	max. Fehler
0.01	5.5893e-04	8.4033e-05	1.1097e-06	5.5893e-04
0.02	0.0011	1.7192e-04	9.6614e-07	0.0011
0.03	0.0018	2.7580e-04	8.7688e-07	0.0018
0.05	0.0026	3.5274e-04	8.0813e-07	0.0026
0.10	0.0054	3.8713e-04	3.4148e-07	0.0054
0.15	0.0081	5.7479e-04	6.0691e-07	0.0081
0.20	0.0110	8.5601e-04	1.1473e-06	0.0110
0.25	0.1753	2.8289e-04	1.0772e-07	0.1753
0.30	0.0174	0.0014	1.6421e-06	0.0174
0.35	0.0207	0.0013	1.8825e-06	0.0207
0.40	0.0245	0.0013	2.2081e-06	0.0245
0.42	0.0259	0.0012	2.3605e-06	0.0259
0.43	0.0266	0.0011	2.6127e-06	0.0266

Tabelle 6.1: Fehler in den Lösungen auf dem Intervalls $[0, 10]$ bei modellprädiktiver Regelung des Beispiels 6.1 mit unterschiedlichen Anfangswerten

von $T = 1.0$ und eine Horizontlänge von $H = 5.0$. Für alle Anfangswerte wurde die gleiche Integrationsroutine ("NART" = 5) verwendet. Feststellbar ist hier, dass der maximale Fehler fast ausschließlich bei $t = 1$ vorliegt. Zudem lässt sich ab $x_0 \approx 0.439$ das Problem mit dem Verfahren nicht mehr lösen, da, wie aus Abbildung 6.2 ersichtlich ist, bereits für die Lösung des ersten optimalen Steuerungsproblems mit der Optimierungsroutine keine stabilisierende Steuerung gefunden werden kann.

Um die Anfangswerte weiter erhöhen zu können empfiehlt es sich, die sampled-data Rate T zu verkleinern. Auffallend ist hier außerdem, dass für $x_0 = 0.25$ der maximale Fehler sich stark erhöht hat, dass aber die optimal gesteuerte Trajektorie im weiteren Verlauf dennoch gegen 0 konvergiert, wie auch Abbildung 6.3 dies veranschaulicht. Dieses Phänomen wird in Kapitel 6.10 noch genauer untersucht.

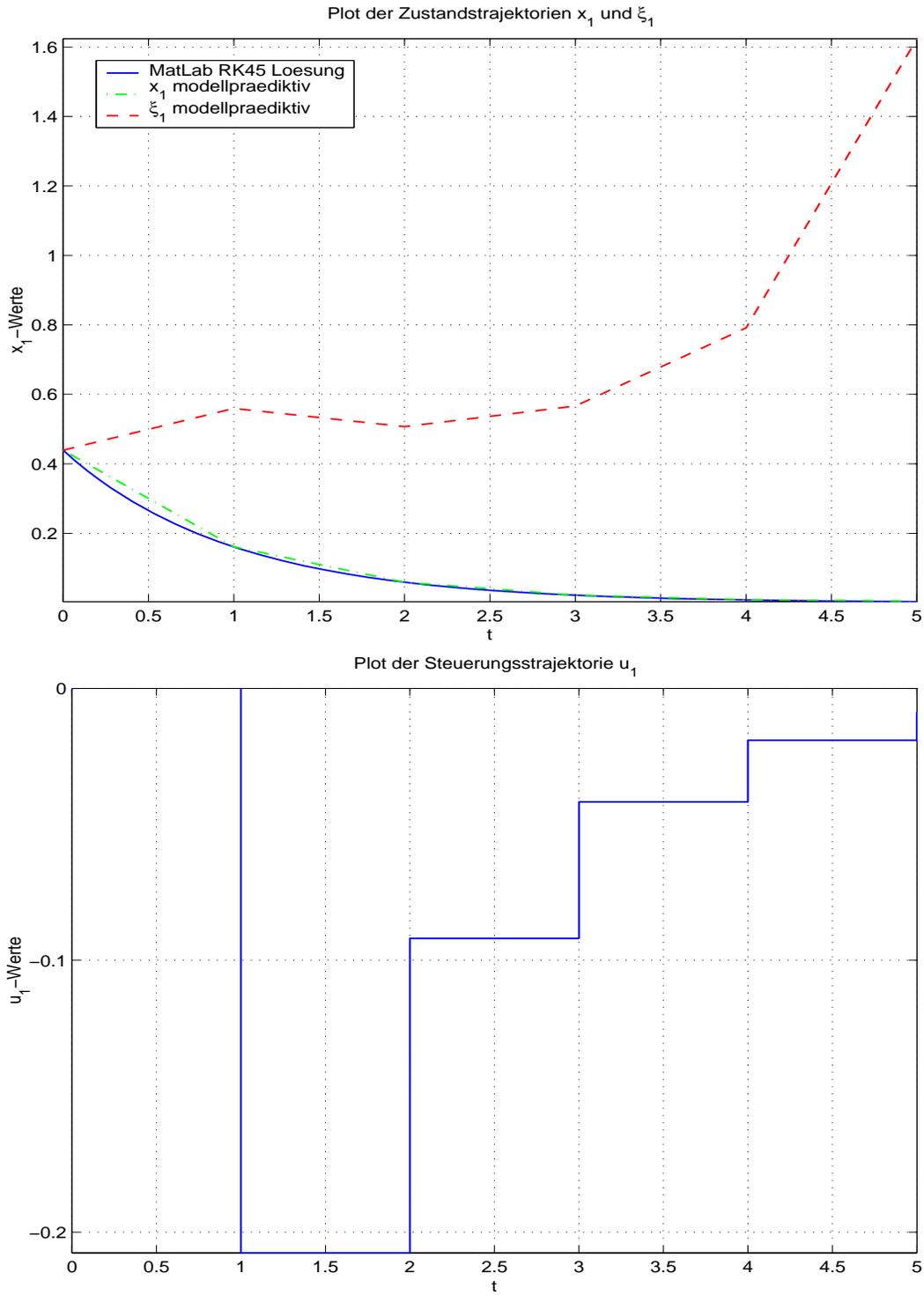
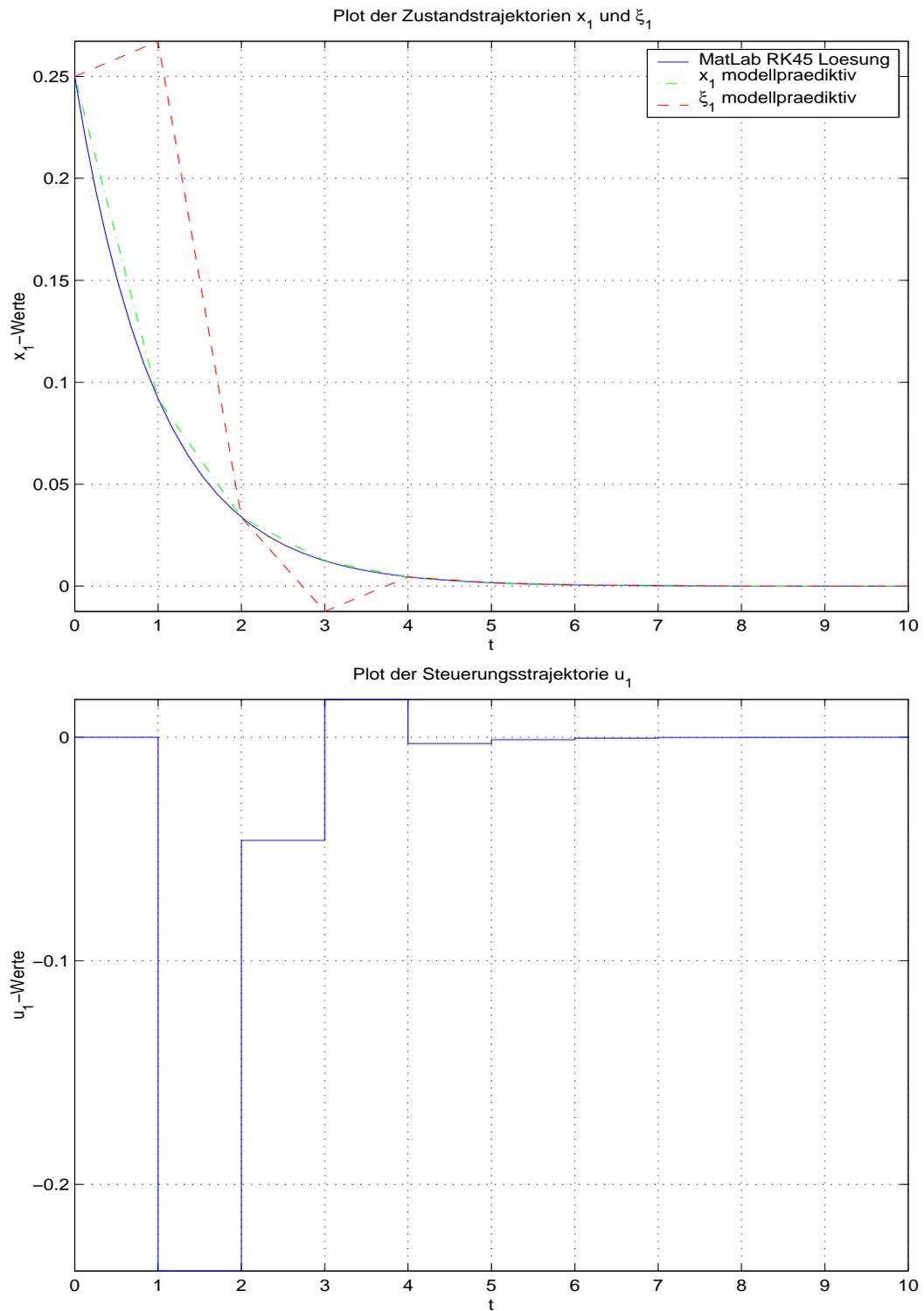


Abbildung 6.2: Lösung des 1. optimalen Steuerungsproblems des 1D Beispiels für $T = 1.0$ und $H = 5.0$ bei Anfangswert $x_0 = 0.439$

Abbildung 6.3: Lösung des 1D Beispiels für $T = 1.0$ und $H = 5.0$ bei Anfangswert $x_0 = 0.25$

6.6 Verbesserung der gesampten Lösung durch modellprädiktive Regelung

Hier soll die Verbesserung untersucht werden, die man sich durch die modellprädiktive Regelung erhofft. Verglichen werden diese Ergebnisse dabei mit der Lösung, die durch Implementierung einer sampled-data Steuerung der Form

$$u_M(t) := u(x(t_k)) \quad , \quad u(x(t_k)) \in U, \quad t \in [t_k, t_{k+1}), \quad t_k := k \cdot T, \quad k = 0, \dots, M - 1 \quad (6.39)$$

gewonnen wird. Dabei berechnet sich $u(x(t_k)) \in U$ aus der Formel des zeitkontinuierlichen Feedbacks und wird stückweise konstant auf den Intervallen $[t_k, t_{k+1})$ implementiert.

Bemerkung 6.4. Die Lösung eines Systems mit gesampteter zeitkontinuierlicher Steuerung (6.39) wird dabei jeweils mit Matlab berechnet.

6.6.1 Verbesserung des Fehlers

Der Vergleich soll anhand von Beispiel 6.2 und Beispiel 6.3 erfolgen, für die sich Fehler ergeben, die in den folgenden Tabellen enthalten sind. Für das Beispiel 6.2 gibt Tabelle 6.2 hierfür die Fehler an, die bei Verwendung der gesampten Steuerung (6.39) entstehen. Tabelle 6.3 zeigt die zugehörigen auftretenden Fehler bei den verwendeten Verfahren "NART"

T	Fehler bei $t = T$	Fehler bei $t = 4$	max. Fehler
0.05	5.2997	0.1478	5.2997
0.10	11.3817	0.1125	11.3817
0.20	23.2665	0.1031	23.2665

Tabelle 6.2: Fehler durch die gesamptete zeitkontinuierliche Steuerung des Beispiels 6.2

und sampled-data Raten T , die bei der Nutzung der modellprädiktiven Regelung entstehen. Das Beispiel 6.2 wurde dazu immer auf einem Horizont der Länge $H = 10 \cdot T$ gelöst und es wurde das Intervall $[0, 4]$ betrachtet. Deutlich zu erkennen ist, dass die Lösung des optimalen Steuerungsproblem für "NART" = 0 nicht funktioniert. Grund hierfür sind die Anfangswerte, werden diese verkleinert, so liefert auch das Euler-Verfahren zur Integration gute Ergebnisse.

NART	T	Fehler bei $t = T$	Fehler bei $t = 4$	max. Fehler
0	0.05	21.6395	43.6491	46.9858
1	0.05	21.6395	0.0745	21.6395
2	0.05	21.6395	0.0682	21.6395
3	0.05	21.6395	0.0682	21.6395
4	0.05	5.2994	0.1301	5.2994
5	0.05	3.3799	0.0953	3.3799
0	0.10	26.5200	43.6491	46.9849
1	0.10	26.5200	0.1300	26.5200
2	0.10	26.5200	0.1296	26.5200
3	0.10	26.5200	0.1296	26.5200
4	0.10	10.2005	0.0915	10.2005
5	0.10	10.2172	0.0994	10.2172
0	0.20	33.3292	43.6491	46.9849
1	0.20	33.3292	0.2984	33.3292
2	0.20	33.3292	0.2848	33.3292
3	0.20	33.3292	0.2848	33.3292
4	0.20	18.8498	0.0976	18.8498
5	0.20	19.5561	0.0927	19.5561

Tabelle 6.3: Fehler in den Lösungen durch die modellprädiktive Regelung des Beispiels 6.2

Für Beispiel 6.3 ergeben sich die in der Tabelle 6.4 für Lösungen mit gesampteter zeitkontinuierlicher Steuerung (6.39) auftretenden Fehler. Tabelle 6.5 zeigt entstandene Fehler in den Lösungen bei Verwendung der modellprädiktiven Regelung auf. Gelöst wurde dabei das

T	Fehler bei $t = T$	Fehler bei $t = 0.8$	max. Fehler
0.01	13.3791	1.5686	22.2618
0.02	39.7538	3.8088	49.6045
0.05	143.9110	14.1300	143.9110

Tabelle 6.4: Fehler in den Lösungen durch die gesamptete zeitkontinuierliche Steuerung des Beispiels 6.3

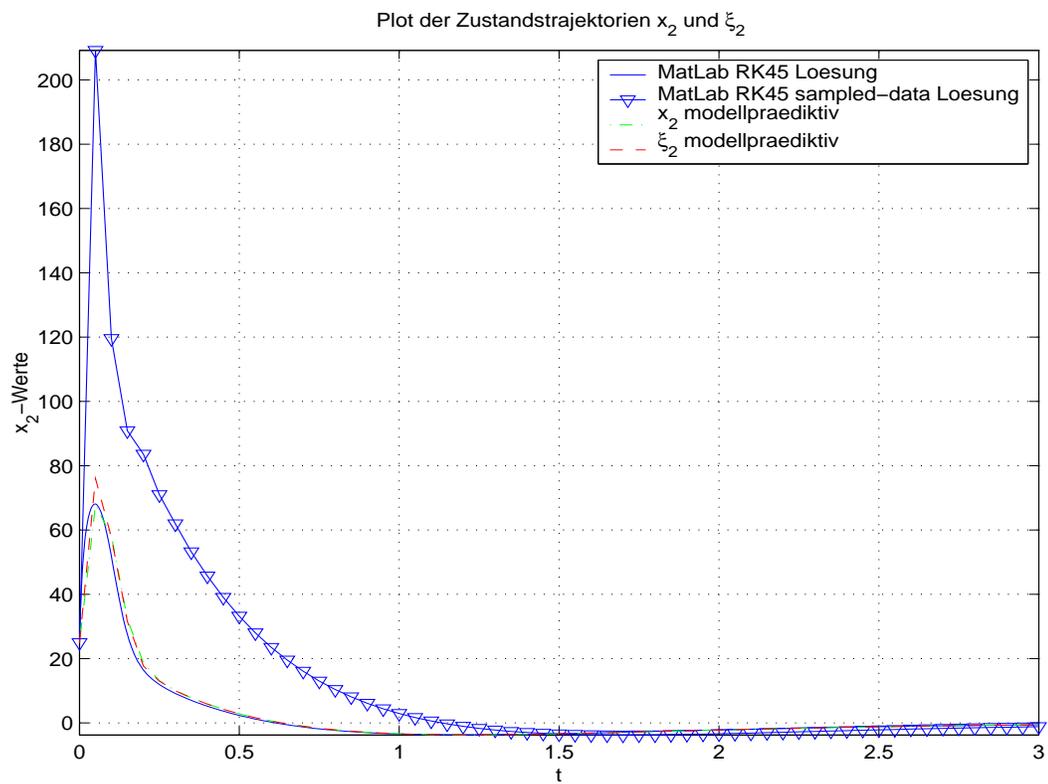
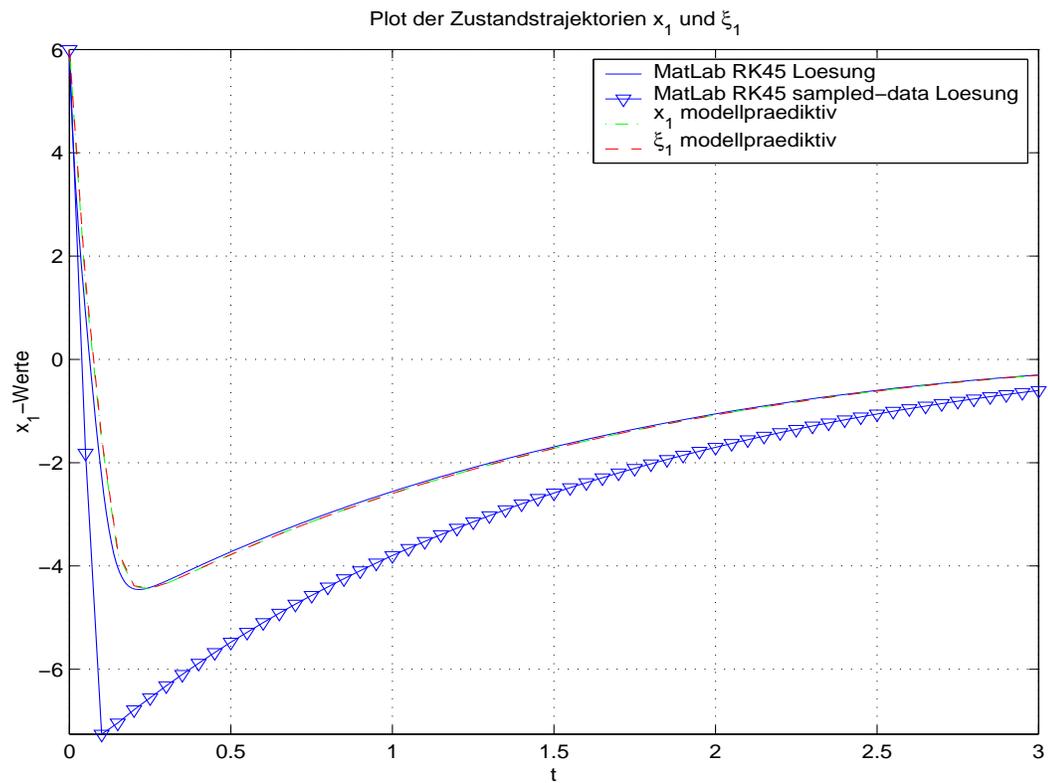
Problem auf dem Intervall $[0, 0.8]$ und einer jeweiligen Horizontlänge von $H = 10 \cdot T$. Besser zeigt die Abbildung 6.4 für die bereits angegebenen Werte $T = 0.05$ und einem Horizont $H = 0.5$ die Verbesserung durch den modellprädiktiven Regelungsansatz.

NART	T	Fehler bei $t = T$	Fehler bei $t = 0.8$	max. Fehler
0	0.01	1.6845	0.0139	1.6845
1	0.01	0.8556	0.0655	2.9571
2	0.01	1.4090	0.0444	1.5253
3	0.01	1.4090	0.0444	1.5253
4	0.01	0.4750	0.0090	0.6435
5	0.01	0.2356	0.0081	0.4952
0	0.02	3.8480	0.0164	3.8480
1	0.02	3.3026	0.1002	3.3026
2	0.02	3.5063	0.0774	3.5063
3	0.02	3.5063	0.0774	3.5063
4	0.02	1.2761	0.0760	3.2911
5	0.02	1.1546	0.0464	2.4709
0	0.05	9.8416	0.0118	9.8416
1	0.05	9.7243	0.1168	9.7243
2	0.05	9.6443	0.1076	9.6443
3	0.05	9.6443	0.1076	9.6443
4	0.05	8.5058	0.2570	8.8058
5	0.05	8.3649	0.2084	8.6184

Tabelle 6.5: Fehler in den Lösungen durch die modellprädiktive Regelung des Beispiels 6.3

Für die Beispiele 6.2 und 6.3 lässt sich erkennen, dass mit der modellprädiktiven Regelung im Vergleich zur Lösung mit gesamelter zeitkontinuierlicher Steuerung im Allgemeinen deutlich geringere Abweichungen von der zu approximierenden zeitkontinuierlich gesteuerten Lösung erreichbar sind. Beim Vergleich der verschiedenen Integrationsverfahren zeigt sich zudem, dass man generell bessere Ergebnisse für die Verfahren "NART" = 4 und "NART" = 5 erhält, aber auch, dass dies nicht in allen Fällen gelten muss.

Zudem sind für alle betrachteten Beispiele kleine sampled-data Raten für den Erhalt einer exakteren Lösung besser geeignet. Die Abweichungen durch eine Verkleinerung der sampled-data Rate T zu reduzieren steht allerdings im Widerspruch zur praktischen Zielsetzung des Verfahrens, denn es soll eine möglichst große sampled-data Rate verwendet werden.



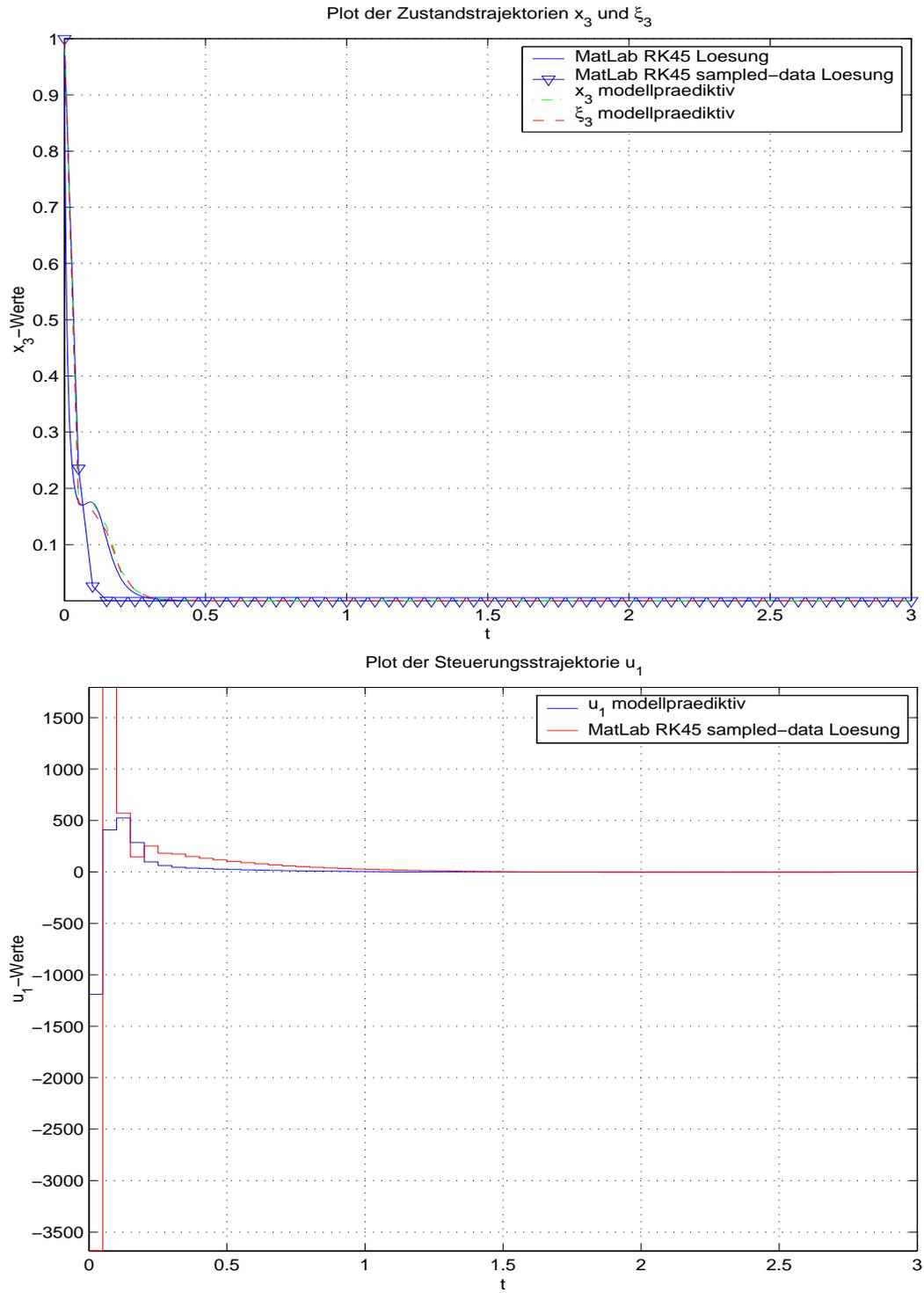


Abbildung 6.4: Lösung des Beispiels 6.3 für $T = 0.05$ und $H = 0.5$ mit zeitkontinuierlicher, gesamelter und modellprädiktiver Steuerung

6.6.2 Vergrößerung der sampled-data Rate

Wie Tabelle 6.4 nahelegt muss die gesamptete zeitkontinuierliche Steuerung (6.39) nicht notwendigerweise das System stabilisieren. Ein solcher Fall ist in Abbildung 6.5 für das Beispiel 6.3 gezeigt. Dabei wurden in den Differentialgleichungen für diese Abbildung die Parameter

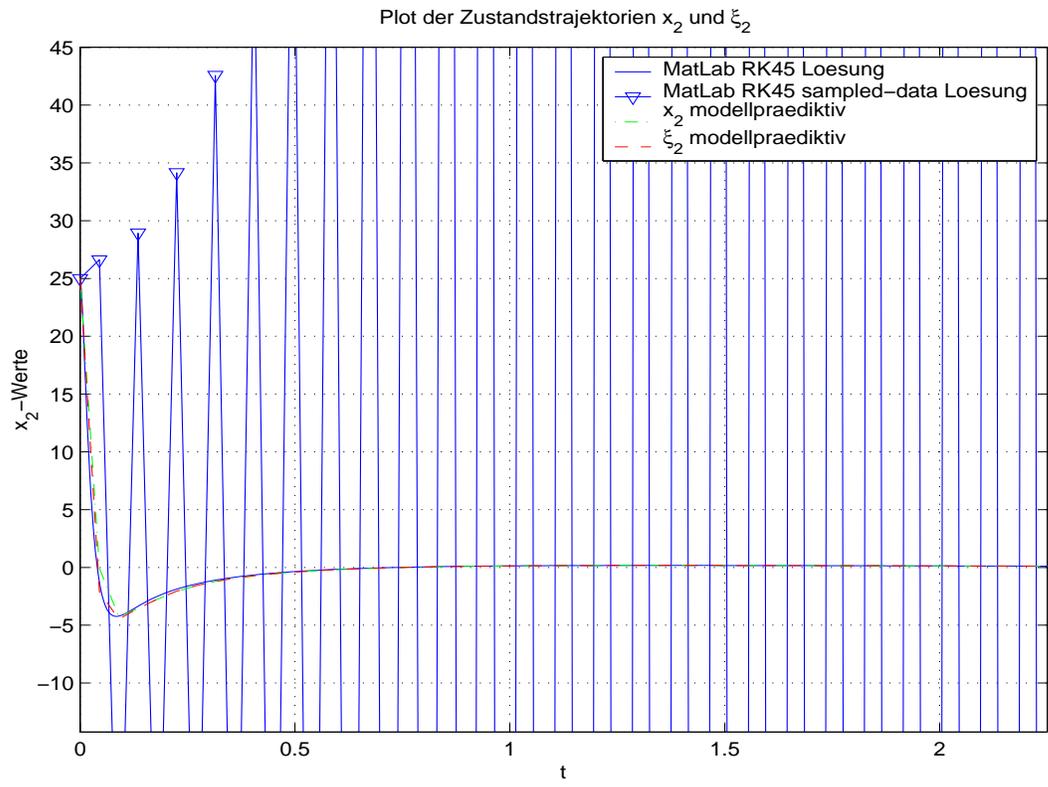
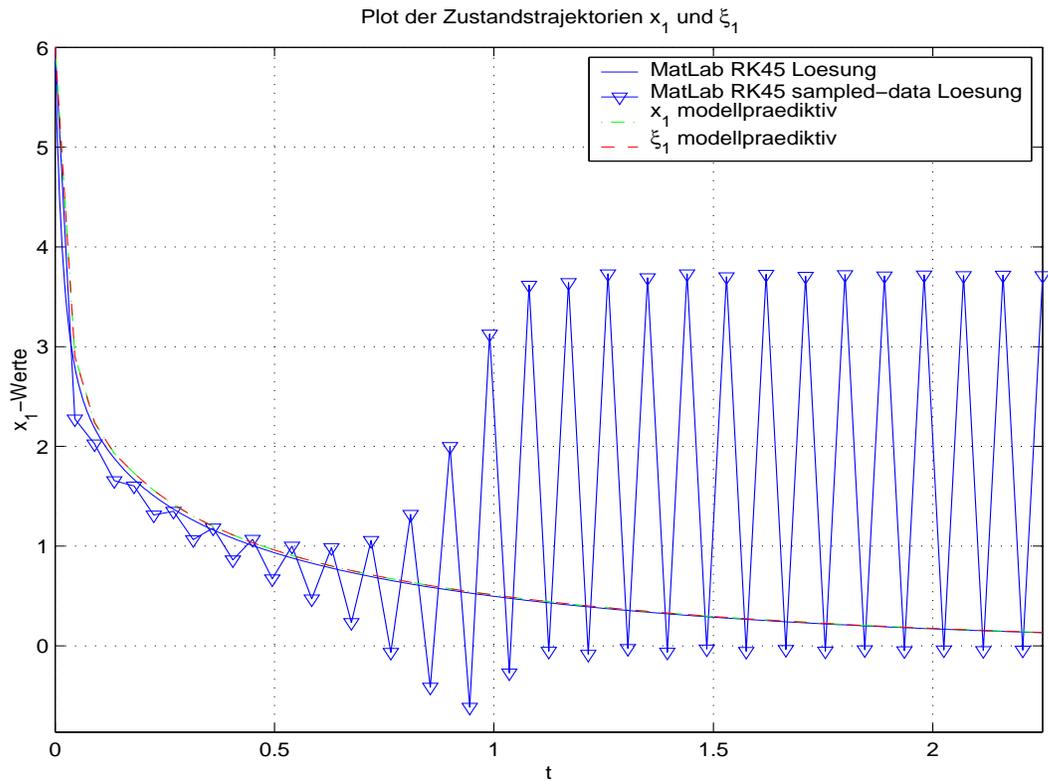
$$\begin{pmatrix} c_1 \\ c_2 \\ \sigma \end{pmatrix} = \begin{pmatrix} 1 \\ 50 \\ 2 \end{pmatrix} \quad (6.40)$$

gewählt. Gerade bei derartigen Problemen kommen die Vorteile des modellprädiktiven Regelungsansatzes voll zur Geltung, da die künftige Entwicklung antizipiert wird und durch Sampling entstehende Fehler berücksichtigt werden. Tabelle 6.6 zeigt dies für die gesamptete Lösung und den modellprädiktiven Ansatz mit den oben angegebenen Parametern. Betrachtet wird dabei das Intervall $[0, 0.8]$ und es werden Horizonte der Länge $H = 10 \cdot T$ verwendet.

T	Modellprädiktive Regelung		Direkt gesamptete Lösung	
	Fehler bei $t = T$	Fehler bei $t = 0.8$	Fehler bei $t = T$	Fehler bei $t = 0.8$
0.01	1.4079	0.0015	6.9402	9.0011e-04
0.02	2.3152	0.0057	15.8659	0.0092
0.03	2.0151	0.0130	22.4503	0.0228
0.04	0.2187	0.0299	26.8193	4.1819
0.05	1.6804	0.0290	29.6262	739.2543
0.08	10.9888	0.0336	33.3520	4.1876e+05
0.10	5.3392	0.0473	34.3776	4.7030e+05
0.20	20.7075	0.4500	38.7403	3.5122e+04

Tabelle 6.6: Verbesserung durch den modellprädiktiven Regelungsansatz bei Vergrößerung der sampled-data Rate

Der modellprädiktive Regelungsansatz liefert also im Allgemeinen nicht nur bessere Lösungen, sondern ermöglicht es zudem die Vergrößerung der sampled-data Rate T . Im Folgenden soll nun auf Problembereiche des Verfahrens und insbesondere auf dessen Grenzen eingegangen werden.



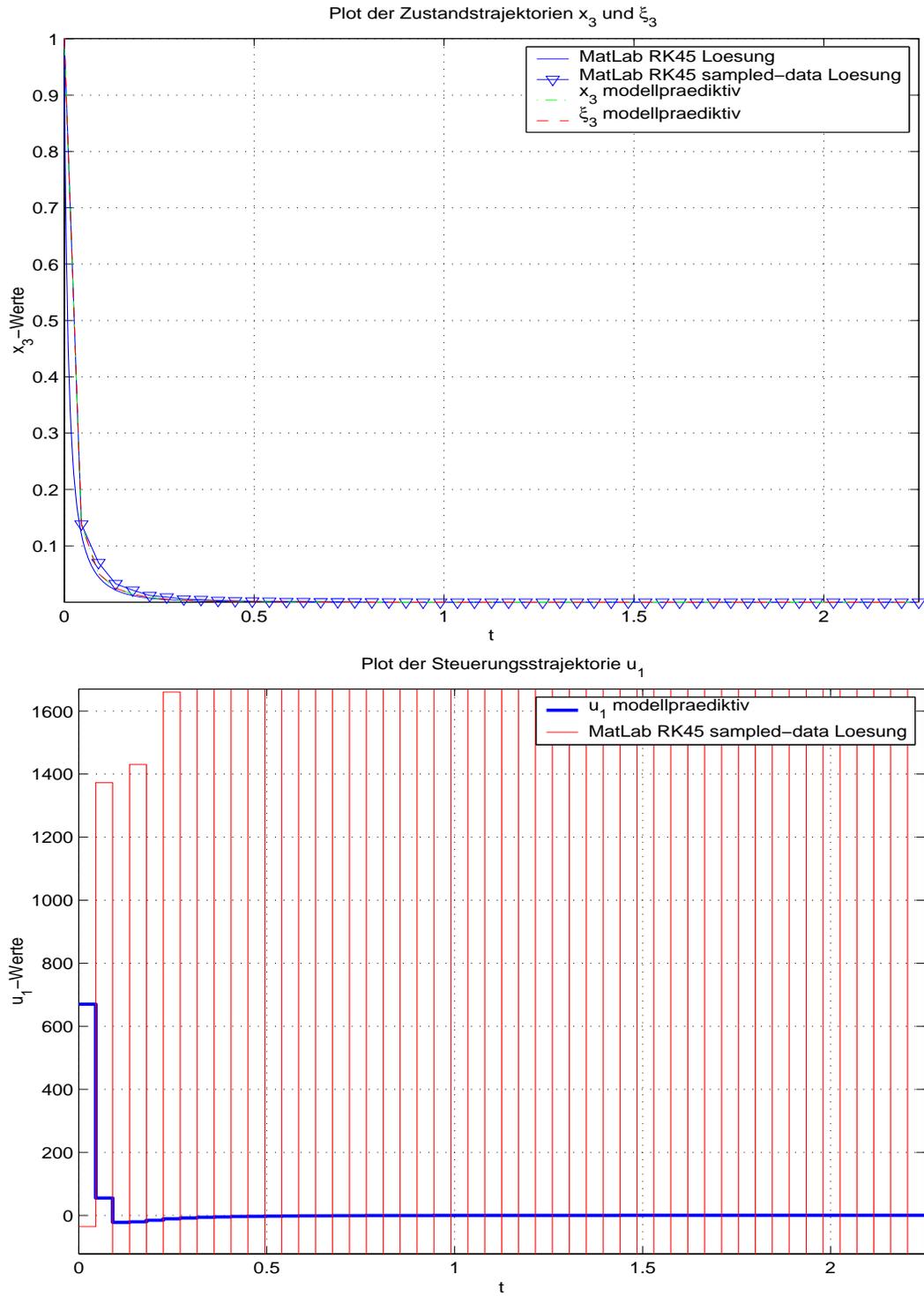


Abbildung 6.5: Lösung des Beispiels 6.3 für $T = 0.045$ und $H = 0.45$ mit zeitkontinuierlicher, gesampteter und modellprädiktiver Steuerung mit geänderten Koeffizienten

6.7 Einfluß der sampled-data Rate

Wie bereits erwähnt ist es wahrscheinlicher, dass man für kleinere sampled-data Raten gute Lösungen oder auch überhaupt Lösungen erhält. Ziel ist es aber, eine möglichst große sampled-data Rate $T \in \mathbb{R}$ zu finden.

Bemerkung 6.5. *Wie die vorgeschlagene Reihenfolge bereits suggeriert bieten die Länge des Horizonts, die Wahl der Parameter und die Toleranz Möglichkeiten, um T vergrößern zu können. Darauf soll im Anschluss eingegangen werden. Als Integrationsverfahren wird dabei immer “NART = 5“ gewählt und auf einen Vergleich verzichtet, da hiermit, wie bereits gezeigt, im Allgemeinen gute Ergebnisse erzielt werden.*

In Tabelle 6.8 wird nun das Beispiel 6.3 für verschiedene sampled-data Raten T betrachtet. Die Horizontlänge $H \in \mathbb{R}$ wird dabei auf zweierlei Arten berücksichtigt:

- $H := k_1 \cdot T$, wobei $k_1 \in \mathbb{R}$ für alle betrachteten T gleich sein soll
- $H := k_2$, wobei $k_2 \in \mathbb{R}$ fest sein soll mit $k_2 = n \cdot T$, $n \in \mathbb{N}$

Für den Vergleich in 6.7 wurde dazu $k_1 = 40$ und $k_2 = 0.4$ gewählt und das Problem jeweils auf dem Intervall $[0, 0.8]$ gelöst. Da für verschiedene Horizontlängen H bei gleicher sampled-

T	H	max. Fehler	Fehler bei $t = 0.8$
0.01	0.40	0.4953	0.0081
0.02	0.40	2.4716	0.0465
0.025	0.40	2.8150	0.0622
0.04	0.40	6.2825	0.1441
0.05	0.40	8.5844	0.2086
0.08	0.40	220.4735	0.7174
0.10	0.40	271.9608	261.4819
0.02	0.80	2.4709	0.0464
0.025	1.00	2.8208	0.0622
0.04	1.60	6.2518	0.1432
0.05	2.00	8.6413	0.2083
0.08	3.20	226.7734	154.3511
0.10	4.00	271.9635	260.7752

Tabelle 6.7: Fehler in den Lösungen durch die modellprädiktive Regelung des Beispiels 6.3 für unterschiedliche sampled-data Raten und Horizontlängen

data Rate T erkennbar ist, dass man für kleinere T bessere Ergebnisse erzielt, soll dies für den Fall $H := k_1 \cdot T$ genauer untersucht werden. Tabelle 6.8 zeigt dies für $k_1 = 10$ wobei wieder das Intervall $[0, 0.8]$ betrachtet wird.

T	max. Fehler	min. Fehler	T	max. Fehler	min. Fehler
0.005	0.6938	7.1083e-05	0.055	12.5565	0.3142
0.010	0.4952	0.0076	0.060	11.5928	0.3171
0.015	1.0623	0.0170	0.065	179.1451	0.4663
0.020	2.4709	0.0464	0.070	550.5021	9.0297
0.025	3.0125	0.0645	0.075	179.4059	0.4200
0.030	4.4905	0.0827	0.080	211.8231	0.5146
0.035	5.8094	0.1747	0.085	236.9181	19.4715
0.040	6.2825	0.1441	0.090	242.5041	2.2443
0.045	7.9847	0.2477	0.095	253.5749	216.9978
0.050	8.6184	0.2084	0.100	273.2365	260.4605

Tabelle 6.8: Fehler in den Lösungen durch die modellprädiktive Regelung des Beispiels 6.3 für unterschiedliche sampled-data Raten

Bemerkung 6.6. Dabei ist zu Tabelle 6.8 anzumerken, dass für die Werte bei $T \geq 0.095$ die verwendete Optimierungsroutine Integrationsfehler bei der Lösung des zeitkontinuierlich gesteuerten Systems verursacht. Abbildung 6.7 veranschaulicht dies graphisch. Die Fehlerquelle lässt sich dadurch erkennen, dass man gute Lösungen für das ungesteuerte Problem erhält, das heißt indem man in der Subroutine "DGLSYS" die sampled-data Steuerung durch die zeitkontinuierliche Steuerung $u(x(t))$ ersetzt.

Bemerkung 6.7. Der minimale Fehler tritt bei den angeführten Beispielrechnungen zumeist am Ende des untersuchten Intervalls auf und wird daher hier und im Folgenden zur Qualitätsuntersuchung einer Lösung angegeben. Sehr große minimale Fehler deuten dabei auf Instabilität des Verfahrens hin und müssen somit nicht am Ende des Intervalls vorliegen. Kleine minimale Fehler dagegen stellen ein Indiz für Stabilität dar.

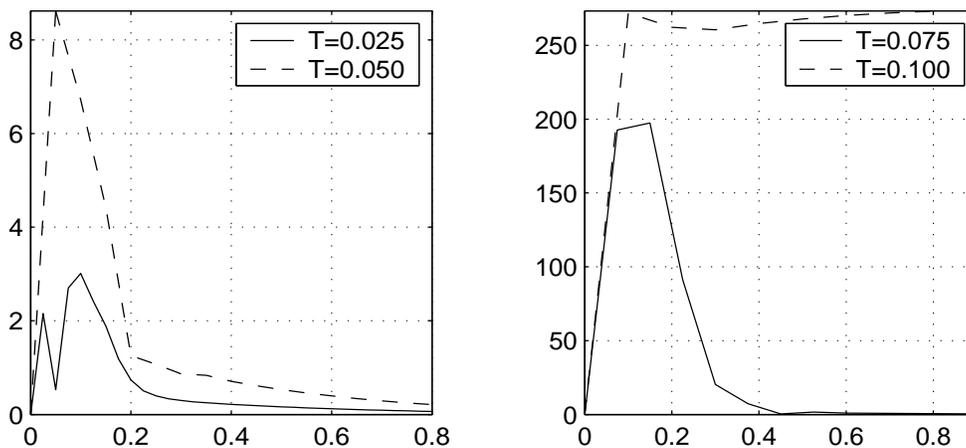
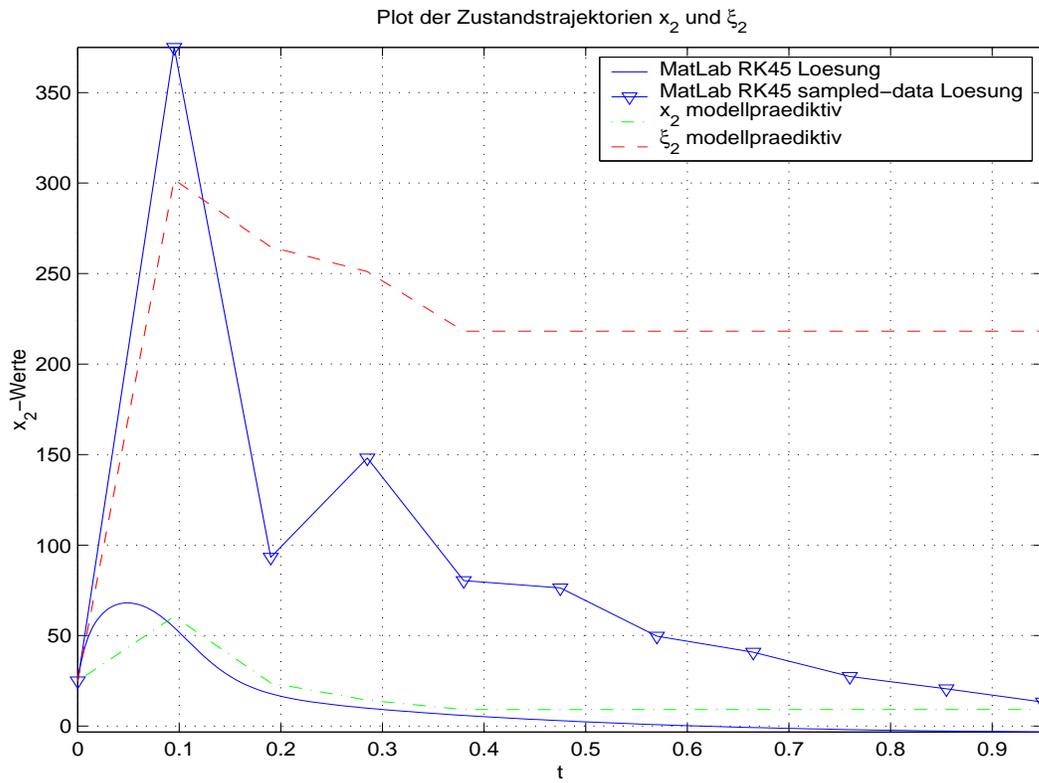
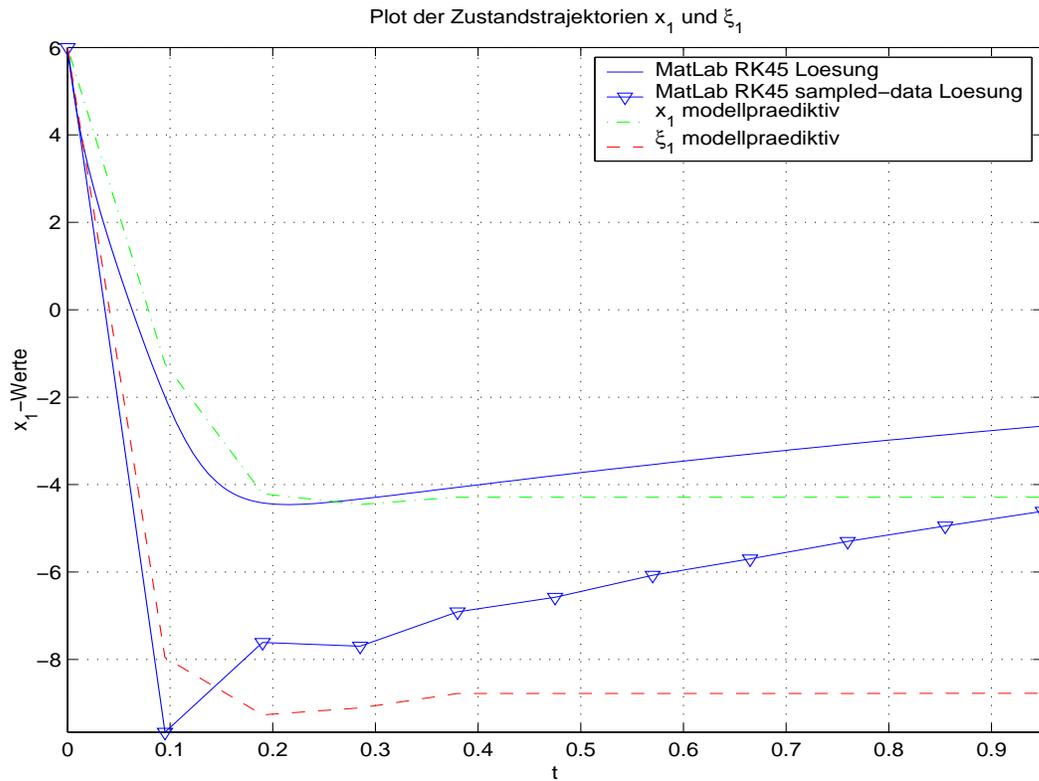


Abbildung 6.6: Fehlerentwicklungen für verschiedene sampled-data Raten



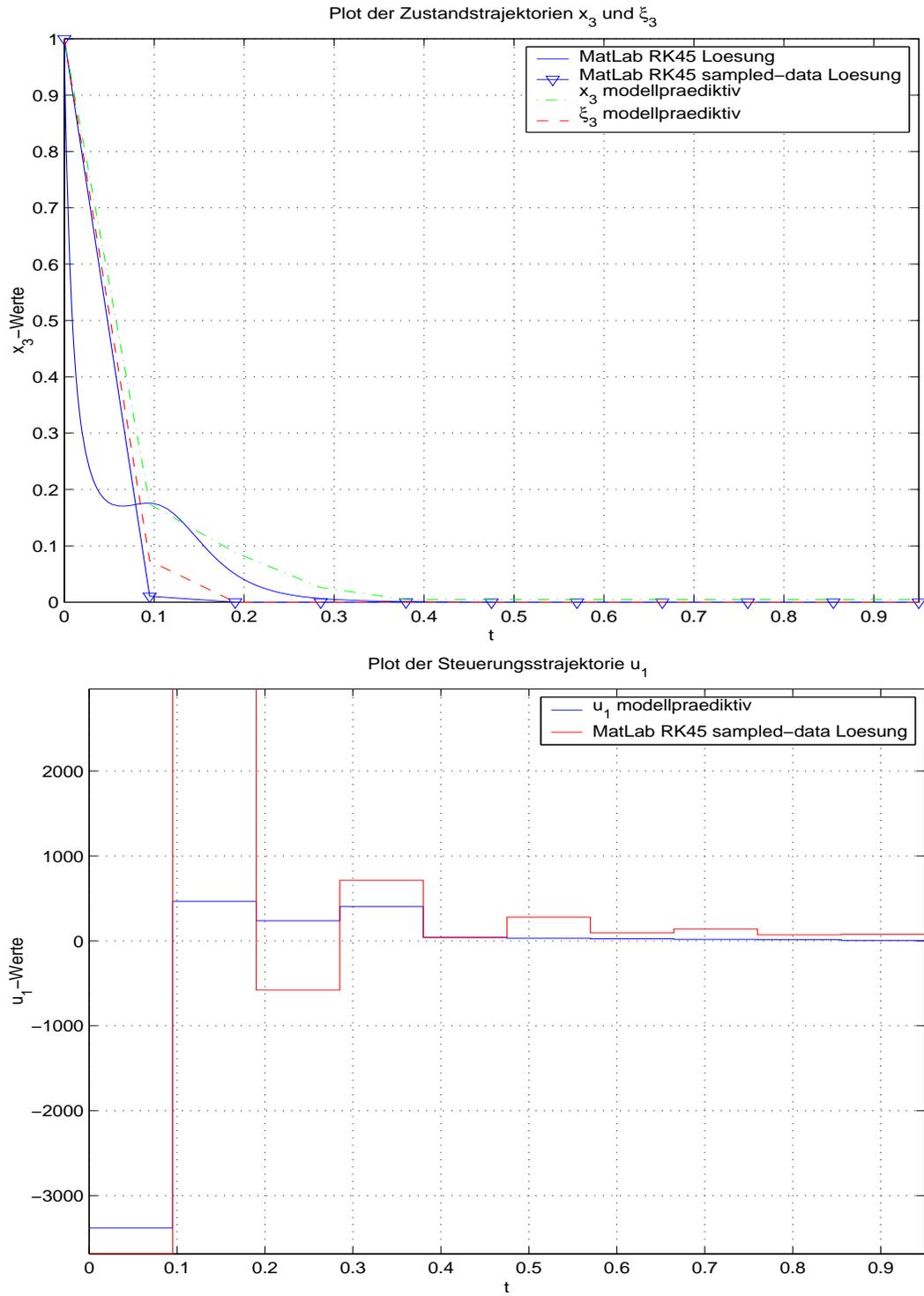


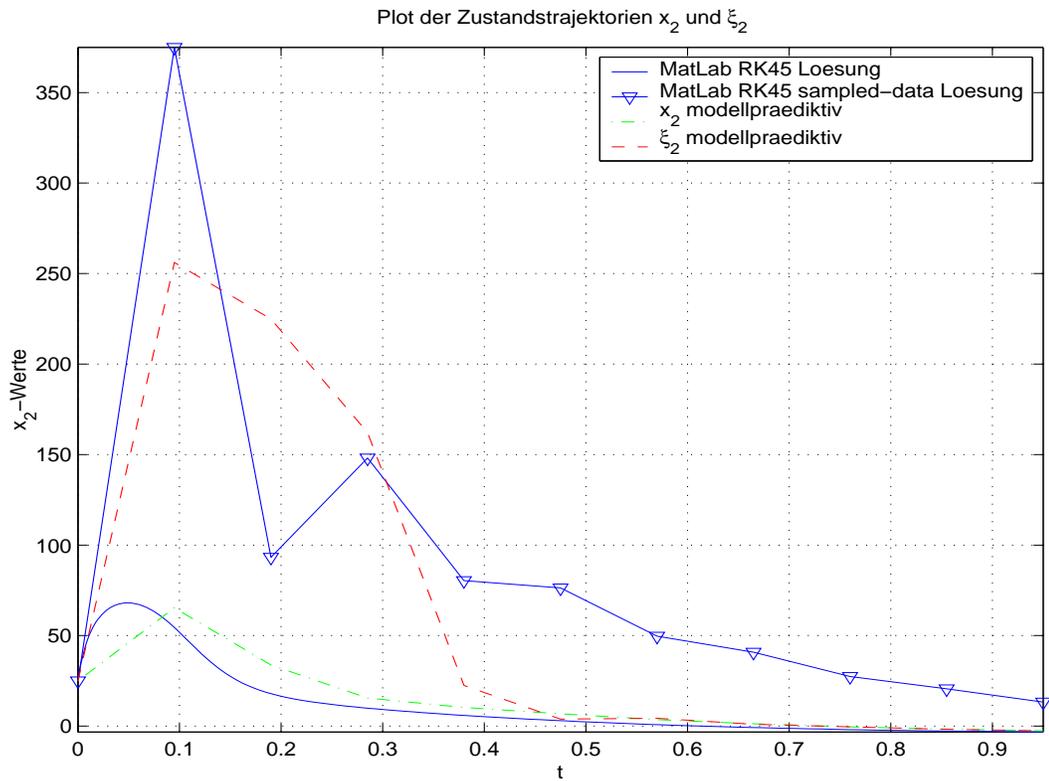
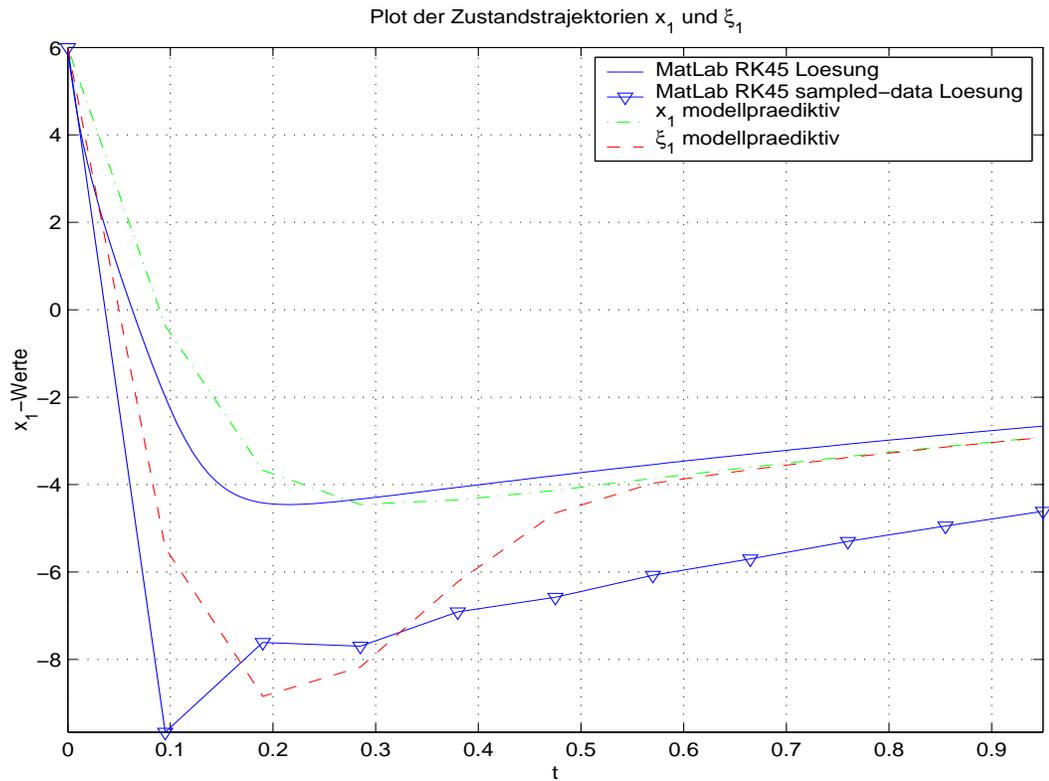
Abbildung 6.7: Falsche Integration bei der Lösung des Beispiels 6.3 für $T = 0.095$ und $H = 0.95$ für $Tol = 10^{-4}$

Um die Integrationsfehler der Optimierungsroutine zu verbessern kann man die Fehlertoleranz "Tol" im Hauptprogramm vergrößern. Dabei muss aber darauf geachtet werden, dass dies teilweise bessere Ergebnisse liefert, dass aber auch mittels der modellprädiktiven Regelung versucht wird eine dann wesentlich stärker fehlerbehaftete Trajektorie nachzusteuern, weswegen hier die Abweichungen der berechneten Trajektorie von der mit Matlab bestimmten Referenzbahn aufgenommen wurden. Abbildung 6.8 verdeutlicht dies für einen Toleranzwert $Tol = 10^{-2}$. In Tabelle 6.9 werden dazu die maximalen und minimalen Fehler für verschiedene Toleranzen und sampled-data Raten dargestellt. Dabei wird wieder $H = k_1 \cdot T$ mit $k_1 = 10$ verwendet und das Problem auf dem Intervall $[0, 0.8]$ gelöst.

T	Tol	max. Fehler	Abweichung	min. Fehler	Abweichung
0.005	1.0e-04	0.6938	0.0543	7.1083e-05	2.1763e-05
0.010	1.0e-04	0.4952	0.4644	7.6544e-03	0.0079
0.020	1.0e-04	2.4709	2.2976	0.0464	0.0476
0.060	1.0e-04	11.5928	1.4406	0.3171	0.3292
0.070	1.0e-04	550.5021	10.6113	9.0297	1.5139
0.075	1.0e-04	179.4059	9.9453	0.4200	1.4512
0.080	1.0e-04	211.8231	7.9537	0.5146	0.5435
0.085	1.0e-04	236.9181	7.1878	19.4715	5.9040
0.090	1.0e-04	242.5041	5.2128	2.2443	0.9901
0.095	1.0e-04	253.5749	6.9978	216.9978	3.4601
0.100	1.0e-04	273.2365	18.1100	260.4605	5.3595
0.005	1.0e-02	0.6497	0.0171	4.6661e-05	1.0495e-04
0.010	1.0e-02	1.8072	0.1242	8.1759e-04	5.2187e-04
0.020	1.0e-02	1.7410	2.1060	0.0247	0.0259
0.060	1.0e-02	11.2210	1.7976	0.1759	0.1873
0.070	1.0e-02	168.7857	23.1161	0.5653	2.5921
0.075	1.0e-02	186.8339	21.6568	0.7450	2.7446
0.080	1.0e-02	443.1021	14.5353	132.7333	8.9159
0.085	1.0e-02	187.5645	8.3479	0.9480	0.9631
0.090	1.0e-02	194.2320	10.5402	0.8992	0.9032
0.095	1.0e-02	211.4320	16.6961	1.6994	4.1523
0.100	1.0e-02	227.6964	18.4562	185.2055	13.9658

Tabelle 6.9: Vergleich der Fehler in den Lösungen durch die modellprädiktive Regelung des Beispiels 6.3 für unterschiedliche sampled-data Raten und Fehlertoleranzen

Tabelle 6.9 macht zudem deutlich, dass eine kleinere Fehlertoleranz selbst bei kleinen sampled-data Raten nicht immer bessere Ergebnisse liefert. Grund hierfür sind die vorliegenden verschiedenen Fehlerquellen.



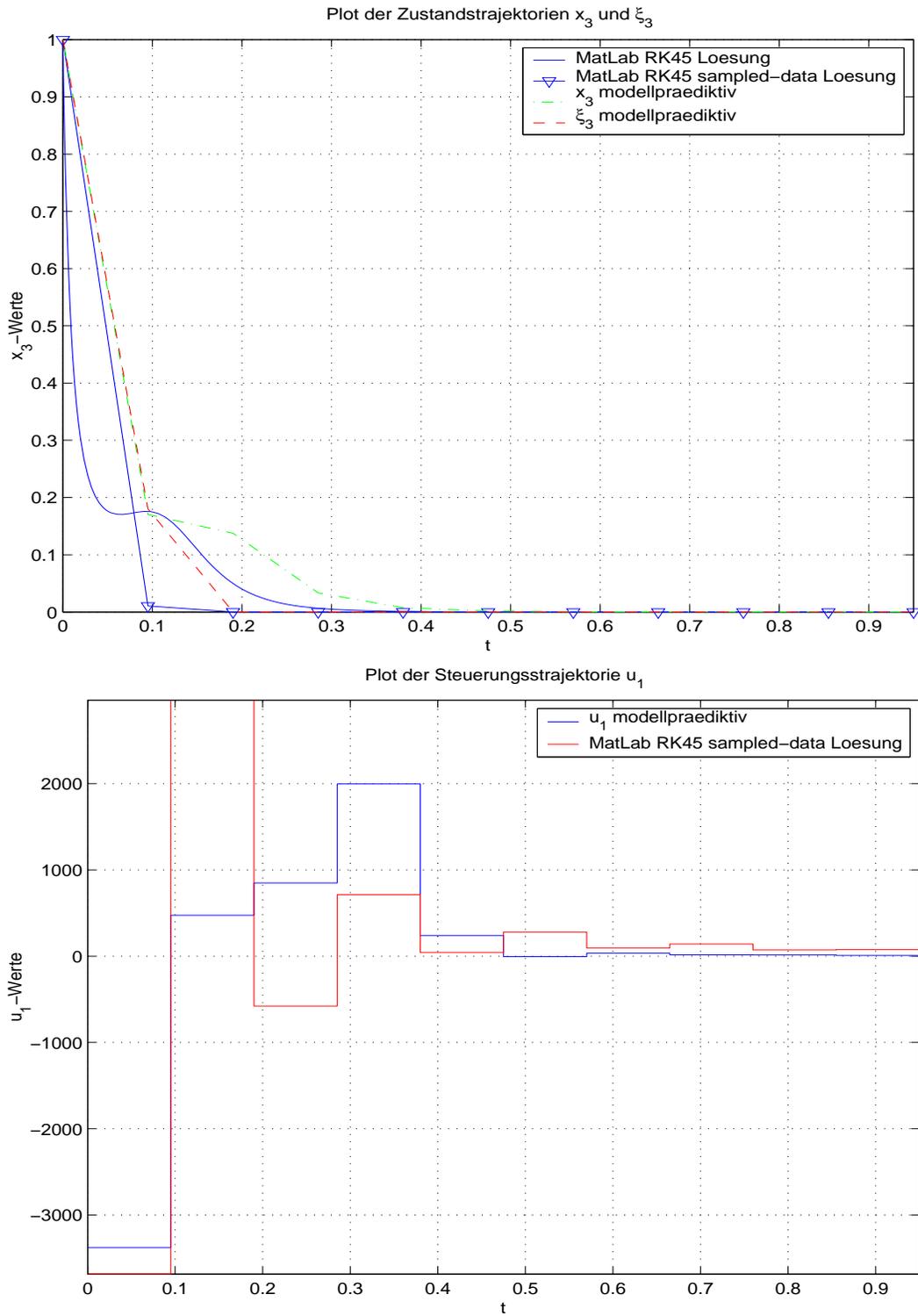


Abbildung 6.8: Lösung des Beispiels 6.3 für $T = 0.095$ und $H = 0.95$ für $Tol = 10^{-2}$

6.8 Einfluß der Länge des Horizonts

In der Theorie, vergleiche Theoreme 4.16, 4.22, 4.24, gelten die Aussagen für alle Horizontlängen $H = M \cdot T$, wobei mit M^* eine untere Schranke für M gegeben ist. In Tabelle 6.10 ist am Beispiel 6.3 zu erkennen, dass zudem eine obere Schranke existiert, die insbesondere bei kritischen sampled-data Raten auftritt.

Bemerkung 6.8. $T \in \mathbb{R}$ wird dabei als kritische sampled-data Rate bezeichnet, wenn die durch T verursachten Abweichungen der sampled-data Steuerung von der zeitkontinuierlichen Steuerung von der Optimierungsroutine gerade noch ausgeglichen werden können. Die zumeist vorliegende Nichtlinearität der zeitkontinuierlichen Steuerung verursacht insbesondere bei der Bestimmung einer Startlösung große Probleme, wie auch Kapitel 6.10 zeigt.

T	H	max. Fehler	min. Fehler	Rechenzeit
0.085	0.850	586.5731	19.4715	70s
0.085	1.190	235.3311	13.5757	98s
0.085	1.275	235.0651	0.0802	225s
0.085	1.360	226.8418	0.0586	290s
0.085	1.445	1317.0502	7.9271	176s
0.085	1.530	534.4092	10.8980	113s
0.085	1.615	233.7771	0.0735	311s
0.085	1.700	232.7984	0.0078	295s
0.085	1.785	265.5365	147.8932	182s
0.085	1.870	233.8774	89.0648	165s
0.085	1.955	325.6937	27.0719	244s

Tabelle 6.10: Vergleich der Fehler in den Lösungen durch die modellprädiktive Regelung des Beispiels 6.3 für unterschiedliche Horizontlängen bei kritischer sampled-data Rate

Betrachtet wurde dabei das Intervall $[0, 3]$. Im Vergleich dazu lässt sich bei nicht kritischen sampled-data Raten dies nicht beobachten, wie in Tabelle 6.11 veranschaulicht ist.

Bemerkung 6.9. Als Konsequenzen ergeben sich:

- Generell sollte die Untersuchung mit kleiner Horizontlänge gestartet werden, wobei sich $H = 10 \cdot T$ bewährt hat.
- Problematisch ist besonders die Startlösung, da diese die Steuerung an den Gitterpunkten mittels Approximation des zeitkontinuierlichen Systems bestimmt. Die stückweise konstante Implementierung mit dieser Startlösung führt wegen bei nichtlinearen Steuerungen zu großen Fehlern, wodurch die Lösung des Optimierungsproblems erschwert wird. Eine Verlängerung des Horizonts vergrößert die Fehler und damit die Probleme bei der Lösung weiter.

T	H	max. Fehler	min. Fehler	Rechenzeit
0.050	0.15	8.6147	0.0251	33s
0.050	0.20	8.6147	0.0251	22s
0.050	0.25	8.6147	0.0251	33s
0.050	0.50	8.6184	0.0252	135s
0.050	0.75	8.6325	0.0252	343s
0.050	1.00	8.6140	0.0252	621s
0.050	1.25	8.6326	0.0252	877s
0.050	1.50	8.6668	0.0252	1336s
0.050	2.00	8.6413	0.0252	2300s
0.050	3.00	8.6614	0.0252	4028s
0.075	0.225	222.4712	0.0765	13s
0.075	0.300	192.6341	0.0514	20s
0.075	0.375	216.8851	0.0681	38s
0.075	0.750	197.4059	0.0609	119s
0.075	1.125	206.1420	0.0599	217s
0.075	1.500	210.5395	0.0625	446s
0.075	1.875	314.0418	8.0358	227s
0.075	2.250	230.6622	1.1778	699s
0.075	3.000	209.2934	2.0524	648s
0.075	4.500	208.0678	0.8336	703s

Tabelle 6.11: Vergleich der Fehler in den Lösungen durch die modellprädiktive Regelung des Beispiels 6.3 für unterschiedliche Horizontlängen

- *Bei diesen Untersuchungen traten die minimalen Fehler immer am Ende des untersuchten Intervalls und die Maximalen nach dem ersten Zeitschritt auf. Dies legt nahe, dass die sukzessive Lösung des Problem begünstigt wird durch die Übergabe der als optimal gefundenen Lösung der Steuerung an den folgenden optimalen Steuerprozess, der dies als Startlösung verwendet. Dies soll im Kapitel 6.10 genauer untersucht werden.*
- *Generell erhöht sich die Rechenzeit bei Betrachtung längerer Horizonte. Wichtiger allerdings ist, dass eine Verkleinerung des Horizonts es eventuell überhaupt erst möglich macht, Lösungen für eine sampled-data Rate bestimmen zu können. Kürzere Gesamtzeiten bei kritischen sampled-data Raten kommen dabei durch die anfänglich schlechte Optimierung zustande.*

Ein weiterer Aspekt der Untersuchung ist das Ergebnis, das sich aus Tabelle 6.12 ablesen lässt. Dabei ergibt sich, dass in den meisten Fällen für längere Horizonte bessere Ergebnisse in Hinsicht auf den Fehler innerhalb des implementierten Verfahrens erzielt werden. Deshalb werden für Tabelle 6.12 die mit Hilfe des Verfahrens bestimmten Werte für das zeitkontinuierlich gesteuerte System zur Fehlerberechnung verwendet und das Intervall $[0, 10]$ betrachtet.

Bemerkung 6.10. *Dementsprechend sollte immer versucht werden für eine feste sampled-data Rate T die Horizontlänge möglichst groß zu wählen. Hierbei steht aber noch immer das Ziel im Vordergrund, eine möglichst große sampled-data Rate T zu finden.*

T	H	max. Fehler	Fehler bei T=10	min. Fehler
0.050	0.25	8.3882	8.2419e-05	3.7140e-05
0.050	0.50	8.3919	2.9570e-05	2.0920e-05
0.050	0.75	8.4061	2.0529e-05	1.3932e-05
0.050	1.00	8.3875	1.7543e-05	8.7496e-06
0.050	2.00	8.4148	1.3076e-05	7.4336e-06
0.050	3.00	8.4350	1.1517e-05	7.7164e-06
0.075	0.375	190.5467	3.2031e-03	8.7346e-04
0.075	0.75	190.7359	6.5275e-04	1.2590e-04
0.075	1.125	197.4236	5.6298e-04	2.6734e-04
0.075	1.50	201.4378	4.8859e-04	2.6635e-04
0.075	3.00	200.1877	1.5674e-03	1.5674e-03
0.075	4.50	200.0452	4.4957e-04	4.4957e-04

Tabelle 6.12: Abweichungen der x - und ξ -Trajektorien des Beispiels 6.3 für unterschiedliche Horizontlängen

6.9 Verwendung einer Lyapunov-Kontroll-Funktion

Wenn der Mayer-Anteil des Zielfunktional eine Lyapunov-Kontroll-Funktion für das zu betrachtende System ist, so ist aus der Theorie zu erwarten, dass der minimale Horizont kleiner gewählt werden kann. Zudem sollte dadurch eine genauere obere Schranke für den unendlichen Horizont gegeben sein und dementsprechend wären zumindest verbesserte Stabilität und Robustheit zu erwarten.

Da für Beispiel 6.2 die Lyapunov-Kontroll-Funktion bekannt ist soll die Untersuchung anhand dieses Beispiels erfolgen. Tatsächlich kann beobachtet werden, dass die Horizontlänge verkleinert werden kann, falls F als Lyapunov-Kontroll-Funktion gewählt wird, wie Tabellen 6.13 und 6.14 zeigen.

Durchgeführt wurden die Berechnungen jeweils auf dem Intervall $[0, 12]$. Tabelle 6.14 zeigt zudem, dass auch für lange Horizonte die Verwendung einer Lyapunov-Kontroll-Funktion bessere Ergebnisse liefert. Die Untersuchung der kritischen sampled-data Rate $T = 0.4$ zeigt zudem, dass sich damit außerdem die sampled-data Rate vergrößern lässt. Bei Vorliegen einer Lyapunov-Kontroll-Funktion sollte diese daher immer verwendet werden.

T	H	max. Fehler	min. Fehler	Lyapunov
0.1	0.5	10.2328	0.3028	nein
0.2	1.0	19.2792	0.0021	nein
0.3	1.5	28.9446	1.1475e-04	nein
0.1	0.5	6.6661	4.8589e-08	ja
0.2	1.0	13.97741	1.4579e-07	ja
0.3	1.5	43.9957	9.1893e-07	ja

Tabelle 6.13: Fehler in den Lösungen durch die modellprädiktive Regelung des Beispiels 6.2 für unterschiedliche sampled-data Raten

T	H	max. Fehler	min. Fehler	Lyapunov
0.1	1.0	10.2172	1.9024e-04	nein
0.1	1.0	10.1792	1.0502e-07	ja
0.1	2.0	10.1787	9.9400e-07	nein
0.1	2.0	10.1443	2.2067e-07	ja
0.2	2.0	19.5561	1.7630e-05	nein
0.2	2.0	18.7068	1.4150e-07	ja
0.2	4.0	18.1828	1.0053e-05	nein
0.2	4.0	17.9253	2.2310e-07	ja
0.3	3.0	972.9952	7.8027	nein
0.3	3.0	32.2292	1.9110e-06	ja
0.3	6.0	248.0926	4.2219	nein
0.3	6.0	30.0373	6.4298e-07	ja
0.4	4.0	630.9820	16.0373	nein
0.4	4.0	38.5546	3.6958e-06	ja
0.4	8.0	39.6654	25.2648	nein
0.4	8.0	38.5394	1.2721e-04	ja

Tabelle 6.14: Fehler in den Lösungen durch die modellprädiktive Regelung des Beispiels 6.2 für unterschiedliche sampled-data Raten und Horizontlängen

6.10 Änderungen der optimalen Lösungen der einzelnen optimalen Steuerprozesse

Betrachtet wird nun das Beispiel 6.3 und die Lösungen, die mittels NUDOCCCS auf den jeweils verschobenen Horizonten berechnet worden sind. Zunächst wird dies für eine nicht kritische sampled-data Rate $T = 0.05$ dargestellt. Generell sollte dabei zu erkennen sein, dass die optimalen Steuerungen der einzelnen optimalen Steuerprozesse nicht stark voneinander abweichen, wie dies auch für diese sampled-data Rate aus den Abbildungen C.1-C.9 gut zu erkennen ist. Zudem erhofft man sich eine Verkürzung der Rechenzeit zur Lösung des

Folgeproblems dadurch, dass man eine bereits gute Startlösung dafür vorgeben kann. Tabelle 6.15 stellt dies im Vergleich dar, wobei die Angaben in Sekunden gemessen werden.

Bemerkung 6.11. *Dies trifft dabei nicht auf alle Schritte zu, jedoch lässt sich eine deutliche Rechenzeitverkürzung für später im Verlauf zu lösende Probleme beobachten.*

T	H	max. Rechenzeit	min. Rechenzeit	Gesamtzeit
0.025	0.50	13s (im 1.OCP)	6s (im 20.OCP)	135s
0.050	1.00	17s (im 1.OCP)	5s (im 18.OCP)	195s
0.070	1.40	23s (im 4.OCP)	6s (im 20.OCP)	224s
0.085	1.70	46s (im 3.OCP)	1s (im 21.OCP)	130s

Tabelle 6.15: Rechenzeitvergleich

Bemerkung 6.12. *Dies kann auch für kritische sampled-data Raten T beobachtet werden, allerdings ist dabei darauf zu achten, dass die Optimierung hierfür auf Grund der schlechten Startlösung nicht immer gute Ergebnisse liefert. Wird allerdings eine gute Lösung gefunden, so ist dies auch an der Rechenzeit erkennbar, die dann deutlich länger ist. Im Anschluss ist die oben gemachte Beobachtung auch hier festzustellen.*

Das zweite hier dargestellte Beispiel für eine kritischere sampled-data Rate $T = 0.085$ zeigt dagegen ein völlig anderes Verhalten, wie die Abbildungen C.10-C.18 veranschaulichen. Hier ist deutlich zu erkennen, dass die Teillösungen der optimalen Steuerung in aufeinander folgenden optimalen Steuerprozessen stark voneinander abweichen. Da die Startlösungen bei großen sampled-data Raten entsprechend schlecht sind kann hier mit der Optimierung kaum eine Verbesserung erreicht werden, wodurch sich auch die kurze Rechenzeit erklärt. Weiter kann allerdings beobachtet werden, dass wenn eine optimale Steuerung gefunden wurde, die die Zustandstrajektorien in relativer Nähe der zeitkontinuierlich Feedback-gesteuerten Trajektorien hält, dann stimmen die optimalen Steuerungen aufeinander folgender optimaler Steuerprozesse gut überein und es ergeben sich die gleichen Aussagen zur Rechenzeit wie bei nicht kritischen sampled-data Raten.

Generell stellen sich die Rechenzeitverläufe ein, wie sie schematisch in den Abbildungen 6.9 und 6.10 dargestellt sind.

6.11 Fazit und Ausblick

Der modellprädiktive Regelungsansatz basiert auf einer einfachen Idee und liefert im Allgemeinen wesentlich bessere Ergebnisse als eine gesamplete zeitkontinuierliche Steuerung. Selbst wenn mit Letzterer keine Stabilität mehr erreicht wird ist dies mit dem Regelungsansatz noch möglich und die sampled-data Rate kann noch deutlich vergrößert werden.

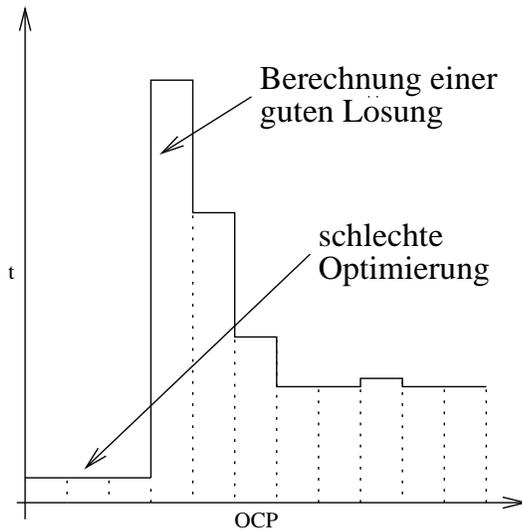


Abbildung 6.9: Rechenzeitverlauf für kritische sampled-data Rate

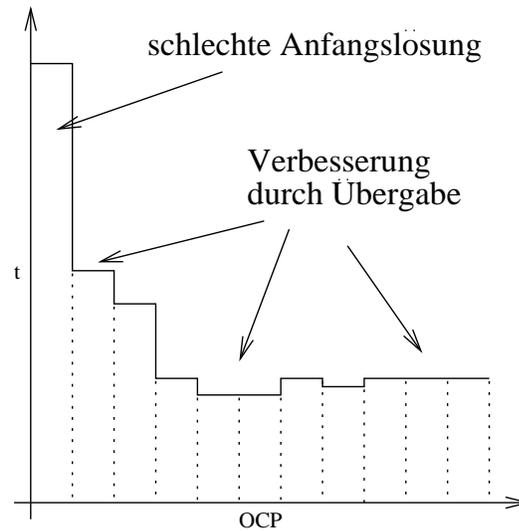


Abbildung 6.10: Rechenzeitverlauf für un-kritische sampled-data Rate

Die Vorgabe einer Anfangsschätzung allerdings gestaltet sich sehr problematisch. Für die verwendete Berechnung aus der zeitkontinuierlichen Steuerung kann nicht einmal die Stabilität des Systems garantiert werden. Ein möglicher anderer Ansatzpunkt wäre die Implementierung des mit gesamplter Steuerung gelösten Systems, das zudem zum Vergleich verwendet wird. Auch das Zusammensetzen einer Startlösung ist denkbar, etwa um bei Nichtdefiniertheit einer Lösung die Horizontlänge trotzdem vergrößern zu können und somit bessere Ergebnisse zu erhalten. Für ein solches Vorgehen ist es allerdings notwendig für kürzere Horizonte, für die die Startlösung definiert ist, die Lösung mit dem modellprädiktiven Regelungsansatz für den längeren Horizont zu berechnen. Da dies allerdings einen sehr hohen Zeitaufwand verursacht wird es in der praktischen Anwendung in einer on-line Berechnung nicht durchführbar sein.

Generell fällt auf, dass der maximale Fehler in den Berechnung fast immer zum Zeitpunkt $t = T$ vorliegt. Eine Gewichtung im Zielfunktional könnte dazu eingeführt werden um diese Fehler zu verkleinern, zum Beispiel mit Hilfe des diskontierten Zielfunktional.

Ein weiterer Punkt, den es zu untersuchen gilt, ist die durch die Optimierungsroutine verursachte fehlerhafte Integration. Möglicher Ansatzpunkt wäre hier eine Trennung der Differentialgleichungssysteme und die separate Integration dieses Systems um das Problem zu umgehen.

Anhang A

Implementierung des modellprädiktiven Regelungsansatzes

A.1 Software Beschreibung von NUDOCCCS

NUDOCCCS stellt ein Programm zur Diskretisierung optimaler Steuerprozesse und der anschließenden Lösung der sich daraus ergebenden Optimierungsprobleme mittels des in der Nag-Routine enthaltenen Verfahrens "E04UCF" dar. Der in diesem Verfahren implementierte Algorithmus ist dabei ein SQP-Verfahren, das im Wesentlichen auf den Ausführungen von P.E.Gill, W.Murray und M.H.Wright [6] beruht.

Obwohl es für das vorliegenden Verfahren nicht relevant ist, lassen sich mit NUDOCCCS auch beschränkte optimale Steuerprozesse lösen. Zusätzlich lassen sich die hierfür ebenfalls nicht benötigten Lösungen der adjungierten Differentialgleichung mitberechnen. Dementsprechend wurden diese Programmteile deaktiviert um eine möglichst schnelle Berechnung der Steuerung- und Lösungstrajektorien zu ermöglichen.

A.1.1 Verfahren zur Integration und Interpolation

Tabelle A.1 gibt eine Übersicht der Kombinationen von Interpolations- und Integrationsverfahren, die in NUDOCCCS standardmäßig über den Eingabeparamter "NART" zur Auswahl bereitgestellt sind.

Hierbei wird folgende Bezeichnung verwendet:

ξ_1	Fehlerordnung des Integrationsverfahrens
ξ_2	Fehlerordnung des Interpolationsverfahrens
RSA	Anzahl der Auswertungen der rechten Seite des Differentialgleichungssystems

NART	Integration	ξ_1	Interpolation	ξ_2	RSA
0	Euler	$O(h)$	/	/	1
1	Heun	$O(h^2)$	/	/	2
2	verb.Poly.	$O(h^2)$	konstant	$O(h)$	2
3	verb.Poly.	$O(h^2)$	linear	$O(h^2)$	2
4	RK 4 England	$O(h^4)$	linear	$O(h^2)$	4
5	RK 5 England	$O(h^5)$	linear	$O(h^2)$	6
6	RK 4 England	$O(h^4)$	kubisch	$O(h^4)$	4
7	RK 5 England	$O(h^5)$	kubisch	$O(h^4)$	6
8, RKEPS > 0	RKF 7/8	$O(h^7)$	konstant	$O(h)$	
8, RKEPS = 0	RADAU5	$O(h^5)$	konstant	$O(h)$	
9, RKEPS > 0	RKF 7/8	$O(h^7)$	linear	$O(h^2)$	
9, RKEPS = 0	RADAU5	$O(h^5)$	linear	$O(h^2)$	

Tabelle A.1: Interpolations- und Integrationsverfahren in NUDOCSS

Mittels der Eingabe einer Zahl zwischen 10 und 19 werden dieselben Integrations- und Interpolationsroutinen angesprochen, jedoch werden zur Berechnung der Gradienten der Zielfunktion und der Nebenbedingungen bei Bedarf zentrale Differenzen verwendet. Durch Eingabe einer Zahl zwischen 20 und 29 werden die Gradienten der Zielfunktion und der Nebenbedingungen mit Hilfe eines einfachen Differenzenquotienten bestimmt. Sollte es notwendig sein wird auch hier der zentrale Differenzenquotient zur Berechnung herangezogen. Die Schrittweite des Differenzenquotienten wird dabei vor Beginn der Optimierung für jede Optimierungsvariable, das heißt für alle Mehrzielknoten, fehlenden Anfangswerte und die zeitdiskrete Steuerung, bestimmt. Zusätzlich werden verschiedene Tests auf Richtigkeit und Lösbarkeit des Problems durchgeführt und rechnerabhängige Standardwerte werden für die Genauigkeit der Lösungsberechnung der optimalen Zielfunktionswerte und Nebenbedingungen, der linearen und nichtlinearen Erreichbarkeit des Problems sowie der Genauigkeit der Abweichung vom Optimalwert verwendet.

Um auf die vorliegende Problemstellung anwendbar zu sein wurden die Verfahren "NART" = 0 bis "NART" = 7 derart verändert, dass die Steuerung innerhalb eines Intervalls $[t_j, t_{j+1}]$ mit Länge T als konstant angenommen und dementsprechend interpoliert wird. Die Verfahren "NART" = 4 und 5 entsprechen somit den Verfahren "NART" = 6 und 7.

Um eine Schrittweitensteuerung innerhalb der Diskretisierungsintervalle $[t_k, t_{k+1})$ implementieren zu können wurde ein Runge-Kutta Verfahren der Ordnung 8 zur Fehlerkontrolle verwendet. Für die Theorie zur Schrittweitensteuerung sei auf Kapitel 5 des Buch von P.Deuffhard und F.Bornemann [3] verwiesen und an dieser Stelle lediglich der verwendete Algorithmus erwähnt:

1. Folgende Werte sind zu übergeben:

Anfangswert x , Anfangszeitpunkt t_0 , Endzeitpunkt t_f , Toleranz $tol > 0$, Mindest-

schrittweise $\tau_{min} \geq 0$, obere Fehlerschranke $\varepsilon_{max} > 0$ und Sicherheitsfaktor $0 < fac < 1$
 Setze $x_0 = x$, $\tau_0 = t_f - t_0$ und $k = 0$.

2. Gilt $t_k = t_f \rightarrow$ Beenden

Gilt $t_k + \tau_k > t_f$, so setze $\tau_k = t_f - t_k$

3. Setze $t_{k+1} = t_k + \tau_k$ und berechne

$$x_{k+1} = \Phi(t_k, x_k, \tau_k) \quad (\text{A.1})$$

$$\hat{x}_{k+1} = \hat{\Phi}(t_k, x_k, \tau_k) \quad (\text{A.2})$$

$$\varepsilon = \|x_{k+1} - \hat{x}_{k+1}\| \quad (\text{A.3})$$

Gilt $\varepsilon \geq \varepsilon_{max}$, so setze $\varepsilon = \varepsilon_{max}$.

Gilt $\varepsilon = 0$, so setze $\varepsilon = \text{tol} \cdot 0.1$.

Gilt $\varepsilon > 0$ und $\varepsilon \leq \varepsilon_{max}$, so berechne

$$\tau_k^* = \sqrt[p+1]{fac \cdot \frac{tol}{\varepsilon} \cdot \tau_k} \quad (\text{A.4})$$

Sonst setze $\tau_k^* = \min\{\tau_{min}, t_f - t_k\}$ und $\varepsilon = \text{tol} \cdot 10$.

4. Gilt $\varepsilon > \text{tol}$, so setze $\tau_k = \tau_k^*$ und gehe zu 3.

5. Gilt $\varepsilon \leq \text{tol}$, so setze $\tau_{k+1} = \tau_k^*$, $k = k + 1$ und gehe zu 2.

Dabei sind Verfahren $\Phi, \hat{\Phi}$ unterschiedlicher Konsistenzordnungen p, \hat{p} mit $p \geq \hat{p} + 1$ gegeben und zu jedem Zeitpunkt t_{k+1} wird der Wert $x_{k+1} = \Phi(t_k, x_k, \tau_k)$ weiterverwendet, da das Verfahren mit höherer Konsistenzordnung im Allgemeinen als genauer zu erachten ist.

Da das Verfahren ‘‘NART’’ = 9 von linearer Steuerung ausgeht ist es für die Problemstellung des modellprädiktiven Regelungsansatzes nicht geeignet. Somit ergibt sich folgende Tabelle:

NART	Integr. Verf.	ξ_1	Interpol. Verf.	ξ_2	$\min(\xi_1, \xi_2)$
0	Euler	$O(h)$	konstant	$O(h)$	$O(h)$
1	Heun	$O(h^2)$	konstant	$O(h)$	$O(h)$
2	verb.Poly.	$O(h^2)$	konstant	$O(h)$	$O(h)$
3	verb.Poly.	$O(h^2)$	konstant	$O(h)$	$O(h)$
4	RK 4 England	$O(h^4)$	konstant	$O(h)$	$O(h)$
5	RK 5 England	$O(h^5)$	konstant	$O(h)$	$O(h)$
8, RKEPS > 0	RKF 7/8	$O(h^7)$	konstant	$O(h)$	$O(h)$
8, RKEPS = 0	RADAU5	$O(h^5)$	konstant	$O(h)$	$O(h)$

Tabelle A.2: Interpolations- und Integrationsverfahren für modellprädiktive Regelung

A.1.2 Parameterzusammenhänge der Eingaben

Durch die Subroutine

```
SUBROUTINE HORIZON(TNDIS1)
  IMPLICIT DOUBLE PRECISION (A-H, O-Z)
```

```
TNDIS1 = ...
```

```
RETURN
```

```
END
```

wird die sampled-data Rate an das Programm übergeben. Dabei entspricht diese dem Zeitschritt der Diskretisierung des modellprädiktiven Ansatzes $T = \text{“TNDIS1“}$, woraus sich für den Horizont eine Länge von $T \cdot (\text{NDIS1} - 1)$ ergibt. Die sampled-data Folge $(t_i)_{i=0, \text{NDIS2}-1}$ des modellprädiktiven Regelungsansatzes ist somit durch

$$t_0 = 0, \quad t_i = t_{i-1} + T \quad \forall i \in \{1, \text{NDIS2} - 1\}$$

gegeben. “NDIS1“ ist dabei vom Benutzer bei Programmstart einzugeben. Der Parameter “NDIS2“ gibt an, wieviele optimale Steuerprozesse hintereinander zu lösen sind und wird ebenso bei Programmstart abgefragt.

A.1.3 Eingabe des optimalen Steuerprozesses

Zur Eingabe des vorliegenden Optimalsteuerungsproblems

$$\text{Minimiere } \hat{u}_{[0, M-1]} = \arg \inf_{v_{[0, M-1]}} J_M(\xi(t), x(t), v_{[0, M-1]}) \quad (\text{A.5})$$

unter

$$J_M(\xi(t), x(t), v_{[0, M-1]}) := \sum_{i=0}^{M-1} \int_{t_i}^{t_{i+1}} l(\xi(t) - x(t), v_i) dt + F(\xi(t_M), x(t_M)) \quad (\text{A.6})$$

$$v_i \in \mathbb{U} \quad \forall i \in \{0, \dots, M-1\} \quad (\text{A.7})$$

$$\dot{\xi}(t) = f(\xi(t), v_i), \quad \xi(0) = \xi, \quad t \in [t_i, t_{i+1}] \quad (\text{A.8})$$

$$\dot{x}(t) = f(x(t), u(x(t))), \quad x(0) = x. \quad (\text{A.9})$$

sind verschiedene Subroutinen zu ändern:

Zur Eingabe des Zielfunktionals müssen für den Mayer-Anteil $F(\xi(t_M), x(t_M))$ die entsprechenden Variablen in der Subroutine

```
SUBROUTINE MINFKT(X, U, T, MIN, NDGL, NSTEUER, NDISKRET)
  IMPLICIT DOUBLE PRECISION (A-H, O-Z)
  DOUBLE PRECISION MIN, LAGINT
```

```
DIMENSION U(NSTEUER,NDISKRET),X(NDGL,NDISKRET),T(NDISKRET)
```

```
MIN = ...
```

```
RETURN
```

```
END
```

und für den Lagrange-Anteil $l(\xi(t) - x(t), v_i)$ in

```
SUBROUTINE INTEGRAL(INT,X,U,T,NDGL,NSTEUER)
```

```
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

```
DOUBLE PRECISION INT
```

```
DIMENSION U(NSTEUER),X(NDGL)
```

```
INT = ...
```

```
RETURN
```

```
END
```

ergänzt werden. Zu beachten ist dabei, dass NUDOCCCS keine Unterscheidung zwischen dem x - und dem ξ -System macht. Für die spätere Visualisierung und Fehlerberechnung ist es dabei nötig, dass folgende Reihenfolge in der NUDOCCCS-Variable "X" eingehalten wird:

$$X(:, I) = (x_1 \quad \xi_1 \quad \dots \quad x_n \quad \xi_n) \quad (\text{A.10})$$

Hierbei ist es jedoch aus Rechenzeitgründen zu empfehlen, ein in Bolza- oder in Lagrange-Form vorliegendes Zielfunktional in ein Problem in Mayer-Form zu transformieren und entsprechend das Differentialgleichungssystem und die Anfangswertbedingungen zu ergänzen. Für diese Operationen sei auf Kapitel 4.2 der Arbeit von Ch.Büskens [2] verwiesen. Das Differentialgleichungssystem

$$\begin{pmatrix} \dot{x}(t) \\ \dot{\xi}(t) \end{pmatrix} = \begin{pmatrix} f(x(t), u(x(t))) \\ f(\xi(t), v_i) \end{pmatrix}$$

und die eventuelle Ergänzung werden in der zuvor genannten Reihenfolge in die Subroutine

```
SUBROUTINE DGLSYS(X,U,T,DX,NDGL,NSTEUER)
```

```
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

```
DIMENSION X(NDGL),U(NSTEUER),DX(NDGL)
```

```
DX(.) = ...
```

```
...
```

```
RETURN
```

```
END
```

eingetragen. Zuletzt sind noch die Anfangswerte x, ξ , ebenfalls in der angegebenen Reihenfolge, in der Subroutine

```

SUBROUTINE ANFANGSWERTE(XMPC,NDGL,NDISKRET)
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  DIMENSION XMPC(NDGL,NDISKRET)

  XMPC(.,1) = ...
  ...

  RETURN
END

```

zu vermerken. Wurde das Zielfunktional dabei in Mayer-Form transformiert, so sind auch die Anfangswerte entsprechend zu ergänzen.

A.2 Veränderungen am Algorithmus

A.2.1 Berechnung der Startschätzung einer Steuerung

Bei NUDOCCCS wird standardmäßig die Nullsteuerung als Startlösung der Steuerung verwendet. Da für das modellprädiktive Problem allerdings eine Funktion der zeitkontinuierlichen Steuerung vorgegeben ist und zudem anzunehmen ist, dass die zu berechnende, auf den Intervallen T konstante sampled-data Lösung in der Nähe der zeitkontinuierlichen Steuerung liegt, wurde zur Bestimmung der Startschätzung ein einfaches Euler-Verfahren verwendet. Die Startschätzung $v_{[0,M-1]}^{st}$ ergibt sich somit aus

$$\dot{x}(t) = f(x(t), u(x(t))) \quad (\text{A.11})$$

$$v_k^{st} = u(x(t_k)), \quad t_k = k \cdot T, \quad k = 0, \dots, M - 1. \quad (\text{A.12})$$

Berechnet wird dies in der Subroutine “UANFANGSW“.

```

SUBROUTINE UANFANGSW(X,U,NDGL,NDIS1,NSTEUER,TNDIS1)
  IMPLICIT DOUBLE PRECISION (A-H, O-Z)
  PARAMETER(EULERSCHRITT = 100.0D0)
  DIMENSION U(NSTEUER,NDIS1),X(NDGL,NDIS1),
1          UHELP(NSTEUER,EULERSCHRITT),XHELP(NDGL,EULERSCHRITT)

C INITIAL VALUES -----
  XHELP(1,1) = X(1,1)
  ...
  DO 520 I=2,NDIS1
    DO 521 J=2,EULERSCHRITT
      XHELP(1,J) = XHELP(1,J-1)+(TNDIS1/(NDIS1*EULERSCHRITT))*( ... )

```

```
C INSERT STANDARD EULER -----
    ...
521  CONTINUE
C HANDOVER OF VALUES AT MPC-GRIDPOINTS -----
    X(1,I) = XHELP(1,EULERSCHRITT)
    XHELP(1,1) = X(1,I)
    ...
C CALCULATING CONTROL VALUES AT MPC-GRIDPOINTS -----
    U(1,I) = ...
    ...
520 CONTINUE
    RETURN
    END
```

Über die Konstante “EULERSCHRITT“ kann der Benutzer dabei die verwendete Schrittweite sehr einfach verändern ohne damit Einfluß auf die sampled-data Rate T zu nehmen.

Bemerkung A.1. Für diese Steuerung kann nicht davon ausgegangen werden, dass sie das zu betrachtende System stabilisiert. Dies stellt einen großen Nachteil dar, da somit gerade für längere Horizonte die Lösung nicht definiert sein muss. Zudem ist es dadurch möglich, dass die Forderung für ein SQP-Verfahren, dass die Startlösung in einer kleinen Umgebung der optimalen Lösung liegen muss, nicht erfüllt ist.

A.2.2 Der modellprädiktive Ansatz: Variablendefinition und Übergabe der Werte zur Lösung des folgenden Optimalsteuerungsproblems

Ausgehend von der NUDOCCCS-Basisprogrammierung wurden folgende Variablen hinzugefügt:

UMPC(N)	Array zur Speicherung der Steuerung nach dem modellprädiktiven Regelungsverfahren. Die Größe des Arrays ist dabei genauso definiert wie die Größe des Arrays zur Speicherung der Steuerung eines einzelnen optimalen Steuerprozesses "U(N)". Soll der zu betrachtende Zeitraum verlängert werden, so muss der Parameter "NDISKRET" eventuell erhöht werden.
XMPC(NDGL,NDISKRET)	Array zur Speicherung der Zustände des modellprädiktiv geregelten Systems. Dabei ist es analog zu dem Array "X(NDGL,NDISKRET)" festgelegt, das die Zustände eines einzelnen optimalen Steuerprozesses speichert.
TMPC(NDISKRET)	Array zur Speicherung der Diskretisierung der Zeit, deren Länge ebenfalls durch den Parameter "NDISKRET" vergrößert werden kann.
ZEIT(NDISKRET)	Array zur Speicherung der Rechenzeit zur Lösung eines einzelnen optimalen Steuerprozesses
TM	Parameter zur Angabe der sampled-data Rate, wird mit der Subroutine "HORIZON" an das Programm übergeben.
NDIS1	Parameter zur Längenangabe des zu untersuchenden Horizonts. "NDIS1" stellt zudem die Anzahl der diskreten Gitterpunkte an, die äquidistant mit Abstand "TM" angeordnet sind.
NDIS2	Parameter zur Angabe der Anzahl der durchzuführenden Verschiebungen des Horizonts.
TNDIS1	Länge des Horizonts

Tabelle A.3: Neueingeführte Parameter

Bei der Übergabe der Werte nach dem Lösen des i -ten optimalen Steuerprozesses und der Verschiebung des Horizonts wird folgendermaßen vorgegangen:

- Speicherung der Steuerungen zum Zeitpunkt t_i
- Speicherung der Zustände zum Zeitpunkt t_{i+1}
- Speicherung der Steuerungs- und Zustandstrajektorien in speziellen Dateien
- Neuberechnung des um die sampled-data Rate T verschobenen äquidistanten Gitters
- Festlegen der neuen Startlösung für die Steuerungen
- Sichern der neuen Anfangswerte in der Datei "AWX.DAT"

Die Verschiebung des äquidistanten Gitters ist dabei nicht einfach als Addition implementiert, sondern wird für jeden optimalen Steuerprozess aus der sampled-data Rate T und der Anzahl der Gitterpunkte "NDIS1" neu berechnet, da die bereits in NUDOCCCS implementierte Gitteranpassungsroutine für zukünftige Anwendungen erhalten bleiben soll.

Schwieriger gestaltet sich die Bestimmung der neuen Startlösung. Da NUDOCCCS den Wert der Steuerung am Ende des Horizonts nicht mitoptimiert, aber einen Startwert benötigt, und zudem durch die Verschiebung der Wert am Anfang des zuvor betrachteten Intervalls wegfällt, fehlen zwei Steuerungswerte für die Startlösung. Auf Grund der Tatsache, dass das System mit der zeitkontinuierlichen Steuerung in eine Ruhelage gebracht werden soll und die zeitkontinuierliche Steuerung zudem ein Zustandsfeedback ist, kann man jedoch davon ausgehen, dass auch die Steuerwerte konvergieren. Somit werden hier die Werte der neuen Startlösung für die letzten beiden Gitterpunkte im um T verschobenen Intervall mit dem Wert der Steuerung für den drittletzten Gitterpunkt gleichgesetzt.

Um nicht den normalen Programmablauf von NUDOCCCS abzuändern werden die Anfangswerte des gerade zu behandelnden optimalen Steuerprozesses in einer separaten Datei "AWX.DAT" abgelegt und bei Bedarf durch die Subroutine "ANFANGSW" ausgelesen, in der diese sonst vom Benutzer hätten gesetzt werden müssen.

A.2.3 Subroutine der Schrittweitensteuerung

Um die Programmstruktur von NUDOCCCS nicht zu verändern wurde in der Subroutine "RUKU046" die verwendeten linearen und kubischen Interpolationen der Steuerung deaktiviert und stattdessen eine konstante Interpolation hinzugefügt. Nach der Berechnung der Zustandswerte am Ende des Intervalls in nur einem Schritt wird die Subroutine "SCHRITTW" aufgerufen, in der die bereits erwähnte Schrittweitensteuerung für die Verfahren "NART" = 0 bis 7 unter Verwendung eines Runge-Kutta-Verfahrens der Ordnung 8 erfolgt. Die Zustandswerte am Ende des Intervalls werden nach dem vollständigen Durchlauf an die Subroutine "RUKU046" zurückgegeben und der normale Ablauf von NUDOCCCS wird wieder aufgenommen.

```

SUBROUTINE SCHRITTW(X,U,T,H,UNBE,NDGL,NSTEUER,NDISKRET,
1   NART,NUNBE,NSTUETZ,MSDGL,MSSTUETZ,K,XH,UH,F,AM,AH1,AH2,
2   AH3,AJ)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DOUBLE PRECISION K
DIMENSION X(NDGL,NDISKRET),U(NSTEUER,NDISKRET+2),T(NDISKRET),
1   K(NDGL,5),XH(NDGL),UH(NSTEUER),F(NDGL),UNBE(NUNBE,NSTUETZ),
2   H(NDISKRET),MSDGL(NUNBE),MSSTUETZ(NSTUETZ),
3   AM(NSTEUER,NDISKRET),AH1(NDISKRET),AH2(NDISKRET),
4   AH3(NDISKRET),UHELP(NSTEUER,2),XH1(NDGL),
5   X1(NDGL),T1(2),UH1(NSTEUER),W(NDGL,12),XSAVE(NDGL)
COMMON/FAST/NVON,NBIS,NADJ,MNART
COMMON/RK/rkeps,tol
EXTERNAL CALC1,CALC2
NUSTUE=2
L=F(1)
C APPROXIMATION FACTOR -----
    FACTOR=0.9D0
2000 DO 2010 I = 1,NSTEUER
        UHELP(I,1) = U(I,AJ)
        UHELP(I,2) = U(I,AJ+1)
2010 CONTINUE
C SPLITTING VALUES TO START TWO DIFFERENT METHODS + SAVE -----
    DO 2020 J = 1,NDGL
        X1(J) = X(J,AJ)
        XH1(J) = XH(J)
        XSAVE(J) = XH(J)
2020 CONTINUE
        T1(1) = T(AJ)
        T1(2) = T(AJ+1)

```

```

      TSAVE = T(AJ)-1
      H1 = T1(2) - T1(1)
C SETTING MINIMUM STEP SIZE -----
C   HMIN=0.0D0
      HMIN = 1.0D-14
C-----
C STEP 1 OF ALGORITHM: SECOND INTEGRATION TO MEASURE ERROR IN FIRST PLACE
C-----
C RUNGE-KUTTA 8 -----
      NHELP = 12*NDGL+1
      HI    = H1
      TH    = T1(1)
      DO 2050 I=1,NSTEUER
          UH1(I)=UH(I)
2050 CONTINUE
      CALL CALCU1(T1(1),T1(1),T1(2),UHELP(1,1),UHELP(1,2),UH1,NSTEUER)
      CALL DGLSYS(XH1,UH1,T1(1),W(1,1),NDGL,NSTEUER)
      DO 5200 I=1,NDGL
5200 W(I,12)=XH1(I)+H1*W(I,1)*(2.D0/27.D0)
      CALL CALCU1(T1(1)+H1*(2.E0/27.E0),T1(1),T1(2),UHELP(1,1),
*           UHELP(1,2),UH1,NSTEUER)
      CALL DGLSYS(W(1,12),UH1,T1(1)+H1*(2.E0/27.E0),W(1,2),NDGL,
*           NSTEUER)
      DO 5201 I=1,NDGL
5201 W(I,12)=XH1(I)+H1*(W(I,1)*(1.D0/36.D0) + W(I,2)*(1.D0/12.D0))
      CALL CALCU1(T1(1)+H1*(1.D0/9.D0),T1(1),T1(2),UHELP(1,1),
*           UHELP(1,2),UH1,NSTEUER)
      CALL DGLSYS(W(1,12),UH1,T1(1)+H1*(1.D0/9.D0),W(1,3),NDGL,
*           NSTEUER)
      DO 5202 I=1,NDGL
5202 W(I,12)=XH1(I)+H1*(W(I,1)*(1.D0/24.D0) + W(I,3)*(1.D0/8.D0))
      CALL CALCU1(T1(1)+H1*(1.D0/6.D0),T1(1),T1(2),UHELP(1,1),
*           UHELP(1,2),UH1,NSTEUER)
      CALL DGLSYS(W(1,12),UH1,T1(1)+H1*(1.D0/6.D0),W(1,4),NDGL,
*           NSTEUER)
      DO 5203 I=1,NDGL
5203 W(I,12)=XH1(I)+H1*(W(I,1)*(5.D0/12.D0) - W(I,3)*(25.D0/16.D0)
*           + W(I,4)*(25.D0/16.D0) )
      CALL CALCU1(T1(1)+H1*(5.D0/12.D0),T1(1),T1(2),UHELP(1,1),
*           UHELP(1,2),UH1,NSTEUER)

```

```

CALL DGLSYS(W(1,12),UH1,T1(1)+H1*(5.DO/12.DO),W(1,5),NDGL,
*   NSTEUER)
DO 5204 I=1,NDGL
5204 W(I,12)=XH1(I)+H1*(W(I,1)*(1.DO/20.DO) + W(I,4)*(1.DO/4.DO)
*           + W(I,5)*(1.DO/5.DO) )
CALL CALCUI(T1(1)+H1*0.5DO,T1(1),T1(2),UHELP(1,1),
*   UHELP(1,2),UH1,NSTEUER)
CALL DGLSYS(W(1,12),UH1,T1(1)+H1*0.5DO,W(1,6),NDGL,NSTEUER)
DO 5205 I=1,NDGL
5205 W(I,12)=XH1(I)+H1*(-W(I,1)*(25.DO/108.DO) + W(I,4)*(125.DO/
*   108.DO) - W(I,5)*(65.DO/27.DO) + W(I,6)*(125.DO/54.DO) )
CALL CALCUI(T1(1)+H1*(5.DO/6.DO),T1(1),T1(2),UHELP(1,1),
*   UHELP(1,2),UH1,NSTEUER)
CALL DGLSYS(W(1,12),UH1,T1(1)+H1*(5.DO/6.DO),W(1,7),NDGL,
*   NSTEUER)
DO 5206 I=1,NDGL
5206 W(I,12)=XH1(I)+H1*(W(I,1)*(31.DO/300.DO)+W(I,5)*(61.DO/225.DO)
*           - W(I,6)*(2.DO/9.DO) + W(I,7)*(13.DO/900.DO) )
CALL CALCUI(T1(1)+H1*(1.DO/6.DO),T1(1),T1(2),UHELP(1,1),
*   UHELP(1,2),UH1,NSTEUER)
CALL DGLSYS(W(1,12),UH1,T1(1)+H1*(1.DO/6.DO),W(1,8),NDGL,
*   NSTEUER)
DO 5207 I=1,NDGL
5207 W(I,12)=XH1(I)+H1*(W(I,1)*2.DO - W(I,4)*(53.DO/6.DO)
*   + W(I,5)*(704.DO/45.DO) - W(I,6)*(107.DO/9.DO)
*   + W(I,7)*(67.DO/90.DO) + W(I,8)*3.DO )
CALL CALCUI(T1(1)+H1*(2.DO/3.DO),T1(1),T1(2),UHELP(1,1),
*   UHELP(1,2),UH1,NSTEUER)
CALL DGLSYS(W(1,12),UH1,T1(1)+H1*(2.DO/3.DO),W(1,9),NDGL,
*   NSTEUER)
DO 5208 I=1,NDGL
5208 W(I,12)=XH1(I)+H1*(-W(I,1)*(91.DO/108.DO)+W(I,4)*(23.DO/108.DO)
*           - W(I,5)*(976.DO/135.DO) + W(I,6)*(311.DO/54.DO)
*           - W(I,7)*(19.DO/60.DO) + W(I,8)*(17.DO/6.DO)
*           - W(I,9)*(1.DO/12.DO) )
CALL CALCUI(T1(1)+H1*(1.DO/3.DO),T1(1),T1(2),UHELP(1,1),
*   UHELP(1,2),UH1,NSTEUER)
CALL DGLSYS(W(1,12),UH1,T1(1)+H1*(1.DO/3.DO),W(1,10),NDGL,
*   NSTEUER)
DO 5209 I=1,NDGL

```

```

5209 W(I,12)=XH1(I)+H1*(W(I,1)*(2383.DO/4100.DO)-W(I,4)*(341.DO/
*   164.DO) + W(I,5)*(4496.DO/1025.DO) - W(I,6)*(301.DO/82.DO)
*   + W(I,7)*(2133.DO/4100.DO) + W(I,8)*(45.DO/82.DO)
*   + W(I,9)*(45.DO/164.DO) + W(I,10)*(18.DO/41.DO) )
CALL CALCU1(T1(2),T1(1),T1(2),UHELP(1,1),
*           UHELP(1,2),UH1,NSTEUER)
CALL DGLSYS(W(1,12),UH1,T1(2),W(1,11),NDGL,NSTEUER)
DO 5210 I=1,NDGL
5210 W(I,12)=XH1(I)+H1*(W(I,1)*(3.DO/205.DO) -W(I,6)*(6.DO/41.DO)
*       - W(I,7)*(3.DO/205.DO) - W(I,8)*(3.DO/41.DO)
*       + W(I,9)*(3.DO/41.DO) + W(I,10)*(6.DO/41.DO) )
CALL CALCU1(T1(1),T1(1),T1(2),UHELP(1,1),
*           UHELP(1,2),UH1,NSTEUER)
CALL DGLSYS(W(1,12),UH1,T1(1),W(1,2),NDGL,NSTEUER)
DO 5211 I=1,NDGL
5211 W(I,12)=XH1(I)+H1*(-W(I,1)*(1777.DO/4100.DO)-W(I,4)*(341.DO/
*   164.DO) + W(I,5)*(4496.DO/1025.DO) - W(I,6)*(289.DO/82.DO)
*   + W(I,7)*(2193.DO/4100.DO) + W(I,8)*(51.DO/82.DO)
*   + W(I,9)*(33.DO/164.DO) + W(I,10)*(12.DO/41.DO)+ W(I,2) )
CALL CALCU1(T1(2),T1(1),T1(2),UHELP(1,1),
*           UHELP(1,2),UH1,NSTEUER)
CALL DGLSYS(W(1,12),UHA,T1(2),W(1,3),NDGL,NSTEUER)
DO 5401 I=1,NDGL
5401 XH1(I)=XH1(I)+H1*(W(I,1)*(41.DO/840.DO)+W(I,6)*(34.DO/105.DO)
*       + W(I,7)*(9.DO/35.DO) + W(I,8)*(9.DO/35.DO)
*       + W(I,9)*(9.DO/280.DO) + W(I,10)*(9.DO/280.DO)
*       + W(I,11)*(41.DO/840.DO) )
GOTO 2120
C-----
C CALCULATING STEP SIZE
C-----
C-----
C STEP 2 OF ALGORITHM: CASE END OF INTEGRATION INTERVAL
C-----
2100 IF (T1(1).EQ.T(AJ+1)) THEN
C GOING TO END OF STEP SIZING, INTERVAL DONE -----
GOTO 5000
ELSEIF ((T1(1) + TAU.GE.T(AJ+1)).AND.(T1(1).LT.T(AJ+1))) THEN
TAU = T(AJ+1) - T1(1)
C SETTING CONTROL FOR CASE END OF INTERVAL -----

```

```

DO 2102 I=1,NSTEUER
    UHELP(I,1) = U(I,AJ)
    UHELP(I,2) = U(I,AJ+1)
2102 CONTINUE
    H1 = TAU
    T1(2) = T(AJ+1)
    ELSE
        DO 2103 I=1,NSTEUER
            UHELP(I,1) = U(I,AJ)
            UHELP(I,2) = U(I,AJ)
2103 CONTINUE
        ENDIF

C-----
C STEP 3 OF ALGORITHM: CALCULATING ERROR
C-----
C CALCULATING VALUES OF XI AND XHI -----
2110 GOTO 2130
C ERROR ESTIMATIONG -----
2120 FEHLER = 0.0D0
    TAU = 0.0D0
    DO 2126 I=1,NDGL
        FEHLER = FEHLER + (XH1(I)-XH(I))**2.0D0
2126 CONTINUE
    FEHLER = DSQRT(FEHLER)
C NOT ALLOWING FEHLER TO BE INF OR ZERO -----
    IF (FEHLER.GE.1.0D20) THEN
        FEHLER = 1.0D20
    ELSEIF (FEHLER.EQ.0.0D0) THEN
        FEHLER = TOL*0.1D0
    ELSEIF ((FEHLER.GT.0.0D0).AND.(FEHLER.LE.1.0D20)) THEN
C GOING TO CALCULATING TAU -----
        GOTO 2127
    ELSE
C IN CASE OF FEHLER = NAN -----
        TAU = HMIN
        FEHLER = TOL * 10.0D0
C GOING TO STEP SIZE ADJUSTING IN CASE OF END OF INTERVAL -----
        GOTO 2128
2127 ENDIF
C CALCULATING TAU FOR DIFFERENT NART -----

```

```

      IF (NART.EQ.0.0DO) THEN
        TAU = (FACTOR*TOL/FEHLER)**(1.0DO/2.0DO) * H1
      ELSEIF ((NART.EQ.1.0DO).OR.(NART.EQ.2.0DO).OR.(NART.EQ.3.0DO))
        * THEN
        TAU = (FACTOR*TOL/FEHLER)**(1.0DO/3.0DO) * H1
      ELSEIF ((NART.EQ.4.0DO).OR.(NART.EQ.6.0DO)) THEN
        TAU = (FACTOR*TOL/FEHLER)**(1.0DO/5.0DO) * H1
      ELSEIF ((NART.EQ.5.0DO).OR.(NART.EQ.7.0DO)) THEN
        TAU = (FACTOR*TOL/FEHLER)**(1.0DO/6.0DO) * H1
      ENDIF
C MINIMUM SETTING FOR TAU -----
      IF (TAU.LT.HMIN) THEN
        TAU = HMIN
2128  ENDIF
      IF (TAU+T1(1).GT.T(AJ+1)) THEN
        TAU = T(AJ+1)-T1(1)
      ENDIF
      T1(2) = T1(1) + TAU
      H1 = TAU
C-----
C STEP 4 OF ALGORITHM: SETTINGS FOR RECALCULATION
C-----
      IF (FEHLER.GT.TOL) THEN
C      IF (TAU.EQ.HMIN) THEN
C COUNTER FOR ABORT -----
C      L = L + 1.0DO
C      IF (L.EQ.1.0D2) THEN
C      DO 4999 I=1,NDGL
C      XH(I) = XSAVE(I)
C      WRITE(*,*) 'JUMP'
C4999  CONTINUE
C      GOTO 5000
C      ENDIF
C      ENDIF
C GOING TO STEP 2 -----
      GOTO 2100
C-----
C STEP 5 OF ALGORITHM: ERROR LESS THAN TOLERANCE
C-----
      ELSE

```

130 Kapitel A: Implementierung des modellprädiktiven Regelungsansatzes

```

        L = 0.0D0
        DO 2129 I=1,NDGL
            X1(I) = XH1(I)
2129    CONTINUE
C MOVING TO NEXT STEP SIZING INTERVAL, TAKING TAU AS NEW GUESS -----
        T1(1) = T1(1) + TAU
        T1(2) = T1(1) + TAU
C GOING TO STEP 2 -----
        GOTO 2100
        ENDIF
C-----
C CALCULATION WITH NEW STEP SIZE
C-----
2130 DO 2132 I=1,NDGL
        XH1(I) = X1(I)
        XH(I) = X1(I)
2132 CONTINUE
        NHELP = 12*NDGL+1
        HI    = H1
        TH    = T1(1)
        DO 2140 I=1,NSTEUER
            UH1(I)=UH(I)
2140 CONTINUE
        CALL CALCU1(T1(1),T1(1),T1(2),UHELP(1,1),UHELP(1,2),UH1,NSTEUER)
        CALL DGLSYS(XH1,UHELP(1,1),T1(1),W(1,1),NDGL,NSTEUER)
        DO 6200 I=1,NDGL
6200 W(I,12)=XH1(I)+H1*W(I,1)*(2.DO/27.DO)
        CALL CALCU1(T1(1)+H1*(2.E0/27.E0),T1(1),T1(2),UHELP(1,1),
*           UHELP(1,2),UH1,NSTEUER)
        CALL DGLSYS(W(1,12),UH1,T1(1)+H1*(2.E0/27.E0),W(1,2),NDGL,
*           NSTEUER)
        DO 6201 I=1,NDGL
6201 W(I,12)=XH1(I)+H1*(W(I,1)*(1.DO/36.DO) + W(I,2)*(1.DO/12.DO))
        CALL CALCU1(T1(1)+H1*(1.DO/9.DO),T1(1),T1(2),UHELP(1,1),
*           UHELP(1,2),UH1,NSTEUER)
        CALL DGLSYS(W(1,12),UH1,T1(1)+H1*(1.DO/9.DO),W(1,3),NDGL,
*           NSTEUER)
        DO 6202 I=1,NDGL
6202 W(I,12)=XH1(I)+H1*(W(I,1)*(1.DO/24.DO) + W(I,3)*(1.DO/8.DO))
        CALL CALCU1(T1(1)+H1*(1.DO/6.DO),T1(1),T1(2),UHELP(1,1),

```

```

*          UHELP(1,2),UH1,NSTEUER)
CALL DGLSYS(W(1,12),UH1,T1(1)+H1*(1.DO/6.DO),W(1,4),NDGL,
*      NSTEUER)
DO 6203 I=1,NDGL
6203 W(I,12)=XH1(I)+H1*(W(I,1)*(5.DO/12.DO) - W(I,3)*(25.DO/16.DO)
*          + W(I,4)*(25.DO/16.DO) )
CALL CALCU1(T1(1)+H1*(5.DO/12.DO),T1(1),T1(2),UHELP(1,1),
*          UHELP(1,2),UH1,NSTEUER)
CALL DGLSYS(W(1,12),UH1,T1(1)+H1*(5.DO/12.DO),W(1,5),NDGL,
*      NSTEUER)
DO 6204 I=1,NDGL
6204 W(I,12)=XH1(I)+H1*(W(I,1)*(1.DO/20.DO) + W(I,4)*(1.DO/4.DO)
*          + W(I,5)*(1.DO/5.DO) )
CALL CALCU1(T1(1)+H1*0.5DO,T1(1),T1(2),UHELP(1,1),
*          UHELP(1,2),UH1,NSTEUER)
CALL DGLSYS(W(1,12),UH1,T1(1)+H1*0.5DO,W(1,6),NDGL,NSTEUER)
DO 6205 I=1,NDGL
6205 W(I,12)=XH1(I)+H1*(-W(I,1)*(25.DO/108.DO) + W(I,4)*(125.DO/
+          108.DO) - W(I,5)*(65.DO/27.DO) + W(I,6)*(125.DO/54.DO) )
CALL CALCU1(T1(1)+H1*(5.DO/6.DO),T1(1),T1(2),UHELP(1,1),
*          UHELP(1,2),UH1,NSTEUER)
CALL DGLSYS(W(1,12),UH1,T1(1)+H1*(5.DO/6.DO),W(1,7),NDGL,
*      NSTEUER)
DO 6206 I=1,NDGL
6206 W(I,12)=XH1(I)+H1*(W(I,1)*(31.DO/300.DO)+W(I,5)*(61.DO/225.DO)
*          - W(I,6)*(2.DO/9.DO) + W(I,7)*(13.DO/900.DO) )
CALL CALCU1(T1(1)+H1*(1.DO/6.DO),T1(1),T1(2),UHELP(1,1),
*          UHELP(1,2),UH1,NSTEUER)
CALL DGLSYS(W(1,12),UH1,T1(1)+H1*(1.DO/6.DO),W(1,8),NDGL,
*      NSTEUER)
DO 6207 I=1,NDGL
6207 W(I,12)=XH1(I)+H1*(W(I,1)*2.DO - W(I,4)*(53.DO/6.DO)
*          + W(I,5)*(704.DO/45.DO) - W(I,6)*(107.DO/9.DO)
*          + W(I,7)*(67.DO/90.DO) + W(I,8)*3.DO )
CALL CALCU1(T1(1)+H1*(2.DO/3.DO),T1(1),T1(2),UHELP(1,1),
*          UHELP(1,2),UH1,NSTEUER)
CALL DGLSYS(W(1,12),UH1,T1(1)+H1*(2.DO/3.DO),W(1,9),NDGL,
*      NSTEUER)
DO 6208 I=1,NDGL
6208 W(I,12)=XH1(I)+H1*(-W(I,1)*(91.DO/108.DO)+W(I,4)*(23.DO/108.DO)

```

```

*           - W(I,5)*(976.DO/135.DO) + W(I,6)*(311.DO/54.DO)
*           - W(I,7)*(19.DO/60.DO) + W(I,8)*(17.DO/6.DO)
*           - W(I,9)*(1.DO/12.DO) )
CALL CALCU1(T1(1)+H1*(1.DO/3.DO),T1(1),T1(2),UHELP(1,1),
*           UHELP(1,2),UH1,NSTEUER)
CALL DGLSYS(W(1,12),UH1,T1(1)+H1*(1.DO/3.DO),W(1,10),NDGL,
*           NSTEUER)
DO 6209 I=1,NDGL
6209 W(I,12)=XH1(I)+H1*(W(I,1)*(2383.DO/4100.DO)-W(I,4)*(341.DO/
*           164.DO) + W(I,5)*(4496.DO/1025.DO) - W(I,6)*(301.DO/82.DO)
*           + W(I,7)*(2133.DO/4100.DO) + W(I,8)*(45.DO/82.DO)
*           + W(I,9)*(45.DO/164.DO) + W(I,10)*(18.DO/41.DO) )
CALL CALCU1(T1(2),T1(1),T1(2),UHELP(1,1),
*           UHELP(1,2),UH1,NSTEUER)
CALL DGLSYS(W(1,12),UH1,T1(2),W(1,11),NDGL,NSTEUER)
DO 6210 I=1,NDGL
6210 W(I,12)=XH1(I)+H1*(W(I,1)*(3.DO/205.DO) -W(I,6)*(6.DO/41.DO)
*           - W(I,7)*(3.DO/205.DO) - W(I,8)*(3.DO/41.DO)
*           + W(I,9)*(3.DO/41.DO) + W(I,10)*(6.DO/41.DO) )
CALL CALCU1(T1(1),T1(1),T1(2),UHELP(1,1),
*           UHELP(1,2),UH1,NSTEUER)
CALL DGLSYS(W(1,12),UH1,T1(1),W(1,2),NDGL,NSTEUER)
DO 6211 I=1,NDGL
6211 W(I,12)=XH1(I)+H1*(-W(I,1)*(1777.DO/4100.DO)-W(I,4)*(341.DO/
*           164.DO) + W(I,5)*(4496.DO/1025.DO) - W(I,6)*(289.DO/82.DO)
*           + W(I,7)*(2193.DO/4100.DO) + W(I,8)*(51.DO/82.DO)
*           + W(I,9)*(33.DO/164.DO) + W(I,10)*(12.DO/41.DO)+ W(I,2) )
CALL CALCU1(T1(2),T1(1),T1(2),UHELP(1,1),
*           UHELP(1,2),UH1,NSTEUER)
CALL DGLSYS(W(1,12),UHA,T1(2),W(1,3),NDGL,NSTEUER)
DO 6401 I=1,NDGL
6401 XH1(I)=XH1(I)+H1*(W(I,1)*(41.DO/840.DO)+W(I,6)*(34.DO/105.DO)
*           + W(I,7)*(9.DO/35.DO) + W(I,8)*(9.DO/35.DO)
*           + W(I,9)*(9.DO/280.DO) + W(I,10)*(9.DO/280.DO)
*           + W(I,11)*(41.DO/840.DO) )
C-----
C NART <= 7
C-----
DO 2160 I=1,NSTEUER
UH1(I)=UHELP(I,1)

```

```
2160 CONTINUE
      CALL DGLSYS(XH,UH1,T1(1),F,NDGL,NSTEUER)
C-----
C NART = 1
C-----
      IF (NART.EQ.1) THEN
        DO 2170 I=1,NDGL
          K(I,1)=F(I)
          XH(I)=X1(I)+H1*K(I,1)
2170 CONTINUE
        DO 2180 I=1,NSTEUER
          UH1(I)=UHELP(I,1)
2180 CONTINUE
        CALL DGLSYS(XH,UH1,T1(2),F,NDGL,NSTEUER)
      ENDIF
C-----
C NART >= 2
C-----
      IF (NART.GE.2) THEN
        DO 2200 I=1,NDGL
          K(I,1)=F(I)
          XH(I)=X1(I)+0.5D0*H1*K(I,1)
2200 CONTINUE
C-----
C NART >= 3 UND NART <= 5
C-----
      IF ((NART.GE.3).OR.(NART.LE.5)) THEN
        DO 2300 I=1,NSTEUER
          UH1(I) = UHELP(I,1)
2300 CONTINUE
C-----
C NART >= 6
C-----
      ELSEIF (NART.GE.6) THEN
        DO 2350 I=1,NSTEUER
          UH1(I) = UHELP(I,1)
2350 CONTINUE
      ENDIF
      CALL DGLSYS(XH,UH1,(T1(2)+T1(1))*0.5D0,F,NDGL,NSTEUER)
    ENDIF
```

```

C-----
C NART >= 4
C-----
      IF (NART.GE.4) THEN
        DO 2400 I=1,NDGL
          K(I,2)=F(I)
          XH(I)=X1(I)+0.25D0*H1*(K(I,1)+K(I,2))
2400    CONTINUE
        CALL DGLSYS(XH,UH1,(T1(2)+T1(1))*0.5D0,F,NDGL,NSTEUER)
        DO 2500 I=1,NDGL
          K(I,3)=F(I)
          XH(I)=X1(I)-H1*K(I,2)+2.0D0*H1*K(I,3)
2500    CONTINUE
        DO 2600 I=1,NSTEUER
          UH1(I)=UHELP(I,1)
2600    CONTINUE
        CALL DGLSYS(XH,UH1,T1(2),F,NDGL,NSTEUER)
      ENDIF

C-----
C NART = 5 ODER NART = 7
C-----
      IF ((NART.EQ.7).OR.(NART.EQ.5)) THEN
        DO 2650 I=1,NDGL
          K(I,4)=F(I)
          XH(I)=X1(I)+H1/27.0D0*(7.0D0*K(I,1)+10.0D0*K(I,2)
F          +K(I,4))
2650    CONTINUE
        DO 2655 I=1,NSTEUER
          IF (NART.EQ.5) THEN
            UH1(I) = UHELP(I,1)
          ELSEIF (NART.EQ.7) THEN
            UH1(I) = UHELP(I,1)
          ENDIF
2655    CONTINUE
        CALL DGLSYS(XH,UH1,T1(1)+2.0D0/3.0D0*H1,F,NDGL,NSTEUER)
        DO 2670 I=1,NDGL
          K(I,5)=F(I)
          XH(I)=X1(I)+H1/625.0D0*(28.0D0*K(I,1)-125.0D0*K(I,2)
F          +546.0D0*K(I,3)+54.0D0*K(I,4)-378.0D0*K(I,5))
2670    CONTINUE

```

```

DO 2680 I=1,NSTEUER
C-----
C NART = 5
C-----
      IF (NART.EQ.5) THEN
          UH1(I) = UHELP(I,1)
C-----
C NART = 7
C-----
      ELSEIF (NART.EQ.7) THEN
          UH1(I) = UHELP(I,1)
      ENDIF
2680 CONTINUE
      CALL DGLSYS(XH,UH1,T1(1)+0.2D0*H1,F,NDGL,NSTEUER)
      ENDIF
      DO 2700 I=1,NDGL
      IF ((NART.EQ.7).OR.(NART.EQ.5)) THEN
          XH(I) = X1(I)+(14.0D0*K(I,1)+35.0D0*K(I,4)
F          +162.0D0*K(I,5)+125.0D0*F(I))*H1/336.0D0
      ELSEIF ((NART.EQ.6).OR.(NART.EQ.4)) THEN
          XH(I) = X1(I)+(K(I,1)+4.0D0*K(I,3)+F(I))
F          *H1/6.0D0
      ELSEIF ((NART.EQ.3).OR.(NART.EQ.2)) THEN
          XH(I) = X1(I)+H1*F(I)
      ELSEIF (NART.EQ.1) THEN
          XH(I) = X1(I)+0.5D0*H1*(K(I,1)+F(I))
      ELSEIF (NART.EQ.0) THEN
          XH(I) = X1(I)+H1*F(I)
      ENDIF
2700 CONTINUE
C CASE END OF STEP -----
      GOTO 2120
C-----
C END STEP SIZE
C-----
5000 CONTINUE
      RETURN
      END

```

A.2.4 Subroutine zur Ausgabe aller berechneter Trajektorien

Die folgende Fortran Subroutine speichert die Trajektorien sowohl der Steuerungen als auch die der Zustände, die NUDOCCCS als Lösung eines Optimalsteuerungsproblems berechnet. Hierzu werden zu jedem Schritt des modellprädiktiven Regelungsverfahrens Dateien angelegt, deren Name hier am Beispiel der Datei "tmp_001x001.dat" exemplarisch erläutert werden soll:

Namensanteile:	Bedeutung:
tmp_	Dateibezeichnung
001	(dreistellige) Zahl des OCPs
x	x für Zustand, u für Steuerung
001	(dreistellige) Zahl der DGL bzw. Steuerung
.dat	Dateiendung

Tabelle A.4: Bedeutung der Dateinamen aus dem Verfahren der modellprädiktiven Regelung

Mittels der angegebenen Matlab-Routine können diese dann visualisiert werden und bieten etwa Einblicke in die Entwicklung der Steuerung im Zeitverlauf. Ebenso kann es sich lohnen, die Ergebnisse der einzelnen Optimalsteuerungsprobleme näher zu betrachten, etwa um Instabilitäten, wie in Abbildung 6.2 veranschaulicht, zu untersuchen.

```

SUBROUTINE AUSGABE1(T,UHELP,X,NSTEUER,NDGL,NDIS1,K,NDISKRET,
*
*           NZUSATZ)
IMPLICIT DOUBLE PRECISION (A-H, O-Z)
DIMENSION UHELP(NDIS1*NSTEUER),
*
*           X(NDGL,NDIS1),T(NDIS1)
CHARACTER*50 C
DO 1100 I=1,NDGL
  IF (I.GE.100) THEN
    IF (K.GE.100) THEN
      WRITE(C,9977) 'tmp_',K,'x',I,'.dat'
    ELSEIF (K.GE.10) THEN
      WRITE(C,9978) 'tmp_0',K,'x',I,'.dat'
    ELSE
      WRITE(C,9979) 'tmp_00',K,'x',I,'.dat'
    ENDIF
  ELSEIF (I.GE.10) THEN
    IF (K.GE.100) THEN
      WRITE(C,9987) 'tmp_',K,'x0',I,'.dat'
    ELSEIF (K.GE.10) THEN
      WRITE(C,9988) 'tmp_0',K,'x0',I,'.dat'

```

```

ELSE
    WRITE(C,9989) 'tmp_00',K,'x0',I,'.dat'
ENDIF
ELSE
    IF (K.GE.100) THEN
        WRITE(C,9997) 'tmp_',K,'x00',I,'.dat'
    ELSEIF (K.GE.10) THEN
        WRITE(C,9998) 'tmp_0',K,'x00',I,'.dat'
    ELSE
        WRITE(C,9999) 'tmp_00',K,'x00',I,'.dat'
    ENDIF
ENDIF
OPEN(13,FILE=C)
DO 1200 J=1,NDIS1
    WRITE(13,9930) T(J),' ',X(I,J)
1200 CONTINUE
CLOSE(13)
1100 CONTINUE
DO 1300 I=1,NSTEUER
    IF (I.GE.100) THEN
        IF (K.GE.100) THEN
            WRITE(C,9977) 'tmp_',K,'u',I,'.dat'
        ELSEIF (K.GE.10) THEN
            WRITE(C,9978) 'tmp_0',K,'u',I,'.dat'
        ELSE
            WRITE(C,9979) 'tmp_00',K,'u',I,'.dat'
        ENDIF
    ELSEIF (I.GE.10) THEN
        IF (K.GE.100) THEN
            WRITE(C,9987) 'tmp_',K,'u0',I,'.dat'
        ELSEIF (K.GE.10) THEN
            WRITE(C,9988) 'tmp_0',K,'u0',I,'.dat'
        ELSE
            WRITE(C,9989) 'tmp_00',K,'u0',I,'.dat'
        ENDIF
    ELSE
        IF (K.GE.100) THEN
            WRITE(C,9997) 'tmp_',K,'u00',I,'.dat'
        ELSEIF (K.GE.10) THEN
            WRITE(C,9998) 'tmp_0',K,'u00',I,'.dat'

```

```
        ELSE
            WRITE(C,9999) 'tmp_00',K,'u00',I,'.dat'
        ENDIF
    ENDIF
    OPEN(13,FILE=C)
    DO 1400 J=1,NDIS1-1
        WRITE(13,9930) T(J),' ',UHELP((J-1)*NSTEUER+I)
1400    CONTINUE
        CLOSE(13)
1300    CONTINUE
9930    FORMAT(F25.13,A2,F25.13)
9977    FORMAT(A4,I3,A1,I3,A4)
9978    FORMAT(A5,I2,A1,I3,A4)
9979    FORMAT(A6,I1,A1,I3,A4)
9987    FORMAT(A4,I3,A2,I2,A4)
9988    FORMAT(A5,I2,A2,I2,A4)
9989    FORMAT(A6,I1,A2,I2,A4)
9997    FORMAT(A4,I3,A3,I1,A4)
9998    FORMAT(A5,I2,A3,I1,A4)
9999    FORMAT(A6,I1,A3,I1,A4)
    RETURN
END
```

Anhang B

Fortran Programme

B.1 Fortran-Code des Programms und dessen Aufruf

Der Code in der vorliegenden Form ist zur Berechnung des Beispiels 6.1 geeignet, das heißt für andere Beispiele sind die Parameter “NDGL“ und “NSTEUER“ anzupassen, die die Anzahl vom Nutzer einzugebender Differentialgleichungen und Steuerungen darstellen. Desweiteren muss beachtet werden, dass die hier angegebene Anzahl an Gitterpunkten “NDISKRET“ bei 201 liegt und somit sowohl die Länge des betrachtbaren Horizonts H jedes einzelnen optimalen Steuerungsproblem beschränkt ist auf “(NDISKRET-1) · TM“ mit “TM“ als sampled-data Rate, als auch die Länge des modellprädiktiven Regelungsproblems auf diesen Wert beschränkt ist. Sollen entsprechend längere Zeiträume betrachtet werden ist dieser Parameter anzupassen. Zu beachten ist dabei, dass die Ausgabe von Zwischenergebnissen maximal 999 Schritte bewältigen kann und bei längeren Untersuchungen dann zu deaktivieren ist. Zum leichteren Verständnis des Programmes sind entsprechende Kommentare im Quellcode enthalten.

```
C-----  
C----- START OF PROGRAM -----  
C-----  
  
PROGRAM MAIN  
IMPLICIT DOUBLE PRECISION (A-H,O-Z)  
PARAMETER(  
A NDGL = 3, ! #ODE (>0)  
B NSTEUER = 1, ! #CONTROLS (>0)  
C NDISKRET = 201, ! MAX # OF GRIDPOINTS (>1)  
D NUNBE = 1, ! #UNKNOWN INITIAL VALUES (>0)  
E NSTUETZ = 1, ! #MULTIPLE SHOOTING KNOTES(>0)  
F NNEBEN = 1, ! #STATE OR MIXED CONSTRAINTS  
G NUGLNB = 1, ! #THEREOF INEQUALITY CONSTRAINTS  
H NRAND = 0, ! #POINTEQUALITIES (E.G. TERMINAL CONDITIONS)
```

```

I  NARTADJ = 2,          ! TYPE OF APPROXIMATING ADJOINTS
J  IPRINT  = 5,          ! PRINT LEVEL
K  DEL1    = 1.0D-6,    ! FINITE DIFFERENCE FOR OPTIMIZER
L  DEL2    = 1.0D-4,    ! FINITE DIFFERENCE FOR GRIDFIT
M  EPS     = 1.0D-12,   ! HIGHEST REACHABLE PRECISION OF SOLUTION
N  EPS2    = 1.0D-120,  ! SMOOTHING OPERATOR FOR TRUNCATION ERRORS
O  EPS3    = 1.0d-6,    ! PRECISION OF GRIDREFINEMENT
P  NZUSATZ = NUNBE*NSTUETZ,
Q  N       = (NDISKRET+2)*NSTEUER+NZUSATZ,
R  M       = NDISKRET*NNEBEN+NRAND+NZUSATZ-NUNBE,
S  ME      = M-NDISKRET*NUGLNB,
T  MAX1M   = M)

  DIMENSION
A  X(NDGL,NDISKRET),U(NSTEUER,NDISKRET+2),DFDU(N),
B  G(MAX1M),T(NDISKRET),UNBE(NUNBE,NSTUETZ),
C  UHELP(N),DCDU(MAX1M,N),BL(N+M),BU(N+M),
D  WORK(3*N*N+2*N*M+21*N+22*M),IWORK(4*N+3*M),
E  MSDGL(NUNBE),MSSTUETZ(NSTUETZ),IUSER(22+NUNBE+NSTUETZ),
F  USER(10+7*NDGL+NNEBEN+NSTEUER+NDISKRET*(NDGL+NSTEUER+5)),
G  ADJ(NDGL,NDISKRET),ADJH(NDGL),DISERR(NDISKRET),
H  U2(NSTEUER,NDISKRET+2),X2(NDGL,NDISKRET+2),T2(2*NDISKRET),
I  DSDXH(NNEBEN*NDGL+2*NDGL),DFDXH(NDGL*NDGL+2*NDGL),
J  PDSDX(NDGL),PD2SD2X(NDGL,NDGL),PDFDX(NDGL,NDGL),
K  CONORDER(NNEBEN+1),CONH(4*NNEBEN+1),
L  UMPC(N),
M  XMPC(NDGL,NDISKRET),
N  TMPC(NDISKRET),
O  ZEIT(NDISKRET)

  COMMON/RK/RKEPS,TOL
  CHARACTER*8 ZEIT1, ZEIT2

C --- CHOICE OF INTEGRATION AND INTERPOLATION METHOD BY USER -----
  WRITE(*,*) 'CHOOSE INTEGRATION AND INTERPOLATION METHOD
  WRITE(*,*) 'NART (e.g. 4):'
  READ(*,*) NART

C --- ORDER OF CONSTRAINTS --( OPTIONAL, MUST NOT BE SET)-----
  CONORDER(1) = 1

C --- NUMBER OF DISCRETE POINTS-----
  WRITE(*,*) 'NDISKRET AT THE BEGINNING (e.g. 11):'
  READ(*,*) NDIS1 !#OF GRIDPOINTS AT THE BEGINNING (>1)
  NDISSAVE = NDIS1

```

```

C --- FIT DYNAMICAL DIMENSIONS-----
      N1      = (NDIS1+2)*NSTEUER+NZUSATZ
      M1      = NDIS1*NNEBEN+NRAND+NZUSATZ-NUNBE
      ME1     = M1-NDIS1*NUGLNB
      MAX1M1  = MAX(1,M1)
C --- DISCRETIZATION OF TIME-----
      CALL HORIZON(TM)
      TNDIS1 = TM * (NDIS1-1)
      DO 104 I=1,NDIS1
          T(I) = TNDIS1/(NDIS1-1)*(I-1)
104   CONTINUE
C --- TOLERANCE OF RKF- OR DAE-SOLVER AND STEP SIZING -----
      TOL     = 1.0D-4
C --- INITIAL PRECISION OF SOLUTION -----
C --- TENDS TO EPS IF GRIDFIT IS USED SEVERAL TIMES -----
      epsgit = 1.0D-14
      RKEPS  = 1.0D-4
C --- INITIAL GUESS FOR TERMINAL TIME -----
      DO 300 I=1,NUNBE
          DO 300 J=1,NSTUETZ
              UNBE(I,J) = 1.0d0
300   CONTINUE
C-----
C----- MPC - ROUTINE -----
C-----
      TMPC(1) = 0
      WRITE(*,*) 'NUMBER OF MPC-OPERATIONS (e.g. 41):'
      READ(*,*) NDIS2 !#OF ITERATIONS OF CONTROL PROBLEM (>1)
C --- Timer -----
      TIME(ZEIT1)
C --- SET BEFORE EACH CALL TO NUDOCCS-----
      ITER = 0          ! NO. OF ITERATIONS
      IFAIL = -1       ! ERROR MESSAGE
C --- SETTING INITIAL VALUES-----
      CALL ANFANGSWERTE(XMPC,NDGL,NDISKRET)
      DO 402 I=1,NDGL
          X(I,1) = XMPC(I,1)
402   CONTINUE
C --- INITIAL GUESS FOR CONTROL VARIABLES-----
      CALL UANFANGSW(X,U,NDGL,NDIS1,NSTEUER,TNDIS1)

```

```
C --- ONLY USED FOR CUBIC INTERPOLATION OF CONTROL-----
      U(1,ndiskret+1) = 1.0d0
      U(1,ndiskret+2) = 1.0d0
C ----- MPC-ITERATION -----
      DO 500 K=1,NDIS2
C ----- START OPTIMIZATION-----
      OPEN(11,FILE='AWX.DAT')
      DO 501 I=1,NDGL
          WRITE(11,*) XMPC(I,K)
501  CONTINUE
      CLOSE(11)
C ----- MARK FOR GRIDREFINEMENT-----
C      NDIS1 = NDISSAVE  !SETTING BACK GRID AFTER REFINEMENT
      CALL NUOCCCS(NDGL,NSTEUER,NDIS1,NUNBE,NNEBEN,NUGLNB,NRAND,
1     NZUSATZ,NART,N1,M1,ME1,MAX1M1,ITER,IFAIL,IPRINT,DEL1,
2     EPSGIT,X,U,DFDU,FF,G,DCDU,BL,BU,T,UNBE,UHELP,
3     NSTUETZ,MSDGL,MSSTUETZ,IWORK,WORK,IUSER,USER)
C ----- POSTOPTIMAL CALCULATION OF THE ADJOINT-----
C      CALL ADJUNG(NDGL,NSTEUER,NDIS1,NUNBE,NNEBEN,NUGLNB,NRAND,
C      1  NZUSATZ,NART,NARTADJ,N1,M1,ME1,MAX1M1,ITER,IFAIL,IPRINT,DEL1,
C      2  EPSGIT,X,U,DFDU,FF,G,DCDU,BL,BU,T,UNBE,UHELP,NSTUETZ,
C      3  MSDGL,MSSTUETZ,IWORK,WORK,IUSER,USER,ADJ,DSDXH,DFDXH,ADJH)
C ----- SAVING CALCULATED MPC-CONTROL-----
      DO 502 J=1,NSTEUER
          UMPC((K-1)*NSTEUER+J) = UHELP(J)
502  CONTINUE
C ----- SAVING CALCULATED STATE VALUES-----
      DO 503 I=1,NDGL
          XMPC(I,K+1) = X(I,2)
503  CONTINUE
C ----- SAVING STATE AND CONTROL VALUES IN FILE-----
      CALL AUSGABE1(T,UHELP,X,NSTEUER,NDGL,NDIS1,K,NDISKRET,
*           NZUSATZ)
C ----- Timer -----
      TIME(ZEIT2)
      ZEIT(K) = (((ICHAR(ZEIT2(:1))-ICHEAR(ZEIT1(:1))))*10.0D0 +
*           (ICHAR(ZEIT(2:2))-ICHAR(ZEIT1(2:2))))*60.0D0 +
*           (ICHAR(ZEIT2(4:4))-ICHAR(ZEIT1(4:4))))*10.0D0 +
*           (ICHAR(ZEIT2(5:5))-ICHAR(ZEIT1(5:5))))*60.0D0 +
*           (ICHAR(ZEIT2(7:7))-ICHAR(ZEIT1(7:7))))*10.0D0 +
```

```

*          (ICHAR(ZEIT2(8:8))-ICHAR(ZEIT1(8:8)))
ZEIT1 = ZEIT2
C ----- CALCULATING NEW DISCRETIZATION-----
      TMPC(K+1) = T(2)
      DO 504 I=1,NDIS1
          T(I) = TNDIS1/(NDIS1-1)*(I-1)+TMPC(K+1)
504   CONTINUE
C ----- CONTROL STARTSOLUTION-----
      DO 505 J=1,NSTEUER
          DO 505 I=1,NDIS1-2
              U(J,I) = UHELP(I*NSTEUER+J)
505   CONTINUE
      DO 506 J=1,NSTEUER
          U(J,NDIS1) = UHELP((NDIS1-2)*NSTEUER+J)
          U(J,NDIS1) = UHELP((NDIS1-2)*NSTEUER+J)
506   CONTINUE
      U(1,NDISKRET+1) = 1.0d0
      U(1,NDISKRET+2) = 1.0d0
C ----- STATE STARTSOLUTION-----
      IF (NDIS2.NE.1) THEN
          DO 507 J=1,NDGL
              DO 507 I=1,NDIS1-1
                  X(J,I) = X(J,I+1)
507   CONTINUE
          ENDIF
C ----- STARTSOLUTIONS OF UNKNOWN INITIAL VALUES AND MULTIPLE SHOOTING
C ----- POINTS -----
      DO 508 I=1,NUNBE
          DO 508 J=1,NSTUETZ
              UNBE(I,J) = -1.0d0
508   CONTINUE
C   IF (IFAIL.EQ.0) THEN
          ITER = 0           ! NO. OF ITERATIONS
          IFAIL = -1        ! ERROR MESSAGE
C ----- GRIDREFINEMENT -----
C   ELSE
C   ITER = 0 ! NO. OF ITERATIONS
C   IFAIL = -1           ! ERROR MESSAGE
C1243 CALL GITTERFIT(NDGL,NSTEUER,NDISKRET,NUNBE,NNEBEN,NUGLNB,
C   1  NRAND,NZUSATZ,NART,N1,M1,ME1,MAX1M1,ITER,IFAIL,IPRINT,DEL1,

```

```
C 2  DEL2,EPGKIT,EPS,EPS3,X,U,DFDU,FF,G,DCDU,BL,BU,T,UNBE,UHELP,
C 3  NSTUETZ,MSDGL,MSSTUETZ,IWORK,WORK,IUSER,USER,CONORDER,
C 4  NDIS1,N1,M1,ME1,MAX1M1,DISERR,X2,U2,T2,pdsdx,pd2sd2x,pdfdx,
C 5  conh,FINISH)
C  WRITE(*,*) 'Take new grid and optimize new :<ENTER>'
C  WRITE(*,*) '<CTRL/C> for termination ...'
C  READ(*,*)
C  GOTO 401
C  ENDIF
500  CONTINUE
C --- SAVE RESULTS-----
      IF (NDIS2.EQ.1) THEN
          DO 509 I=1,NDIS1
              T(I) = TNDIS1/(NDIS1-1)*(I-1)
509  CONTINUE
          CALL AUSGABE(FF,X,ADJ,UHELP,T,G,NDGL,NSTEUER,NDIS1,NNEBEN,
1      NRAND,N1,M1)
      ELSE
          DO 510 I=1,NDIS2
              T(I) = TNDIS1/(NDIS1-1)*(I-1)
510  CONTINUE
          CALL AUSGABE(FF,XMPC,ADJ,UMPC,TMPC,G,NDGL,NSTEUER,NDIS2,NNEBEN,
1      NRAND,N1,M1)
      ENDIF
C --- SAVE GENERAL DATA FOR VISUALIZATION -----
      OPEN(11,FILE='Problemdaten.dat')
          WRITE(11,*) NDIS1
          WRITE(11,*) NDIS2
          WRITE(11,*) TM
          WRITE(11,*) TOL
          WRITE(11,*) NART
          WRITE(11,*) RKEPS
          WRITE(11,*) NDGL
          WRITE(11,*) NSTEUER
      CLOSE(11)
      WRITE(*,*)'STATE AND CONTROL TRAJECTORIES SAVED.'
      WRITE(*,*)'TYPE makepic TO SHOW DIAGRAMMS OF THE TRAJECTORIES.'
C ----- END OF MPC-ITERATION -----
C --- SAVING TIMER DATA -----
      OPEN(11,FILE='Zeiten.dat')
```

```

      DO 511 I=1:NDIS2
        WRITE(11,*) ZEIT(I)
511  CONTINUE
      CLOSE(11)
      STOP
      END
C-----
C----- END OF PROGRAM -----
C-----

```

Zur Compilierung und zum Aufruf des Programms ist dabei wie folgt zu verfahren:

```

make Dateiname des Programms → Compilierung
Dateiname des Programms    → Aufruf

```

Durch den Aufruf des *make*-Befehls werden aus NUDOCCCS sämtliche, teils modifizierte Subroutinen sowie alle nötigen Bibliotheken zu dem Programm gelinkt.

Zur Speicherung und Veranschaulichung der Werte wird die bereits vorhandene Ausgaberroutine von NUDOCCCS verwendet. Dabei werden sämtliche Werte des modellprädiktiven Regelungsproblems in der Datei "KURVEN.DAT" gespeichert. Mittels des Befehls

makepic

wird diese dann in Dateien wie zum Beispiel "tmpx001.dat" überführt, wobei für die dementsprechenden Werte die Dateinamen sich wie folgt ergeben:

Namensanteil	Bedeutung
tmp	Dateibezeichnung
x	x für Zustand, u für Steuerung
001	(dreistellige) Zahl der DGL bzw. Steuerung
.dat	Dateiendung

Tabelle B.1: Bedeutung der Dateinamen aus NUDOCCCS

Im Programm enthalten sind zudem hier nicht aktivierte Aufrufe der Subroutinen "ADJUNG" und "GITTEFIT", wobei Letztere eine Gitteranpassung ermöglicht. Erstere dient dazu, die Lösung der adjungierten Differentialgleichung postoptimal zu berechnen. Um diese Trajektorien speichern zu können ist in der Subroutine "AUSGABE", enthalten in der NUDOCCCS-Datei "basic.f", der entsprechende Teil zu aktivieren.

B.2 Programm zu Beispiel 6.1

Generell sind für die Beispiele die folgenden Daten einzugeben:

Problemkomponente		Subroutine
Lagrange-Anteil	l	→ INTEGRAL
Mayer-Anteil	F	→ MINFKT
zeitkontinuierlich gesteuertes System	\dot{x}	→ DGLSYS
sampled-data System	$\dot{\xi}$	→ DGLSYS
Anfangswerte	x, ξ	→ ANFANGSWERTE
sampled-data Rate	T	→ HORIZON
Startschätzung der Steuerung	u	→ UANFANGSW

Tabelle B.2: Eingabe der Problemstellung

Die Subroutinen “RANDBED“, “NEBENBED“, “CONBOXES“ und “MAS“ sind auf Grund der Unbeschränktheit des Problems Dummy-Routinen.

Zusätzlich sind im Hauptprogramm Angaben zur Anzahl der Differentialgleichungen und der Steuerungen nötig:

Da es sich bei dem Problem 6.1 um ein eindimensionales Problem handelt ergeben sich zwei Differentialgleichungen, eine mit kontinuierlicher implementierter Steuerung und eine zu steuernde. Desweiteren wurde zur Verwendung eines Zielfunktional in Mayer-Form eine weitere Differentialgleichung hinzugefügt. Somit ist im Hauptprogramm “NDGL“ auf 3 zu setzen. Zudem liegt nur eine eindimensionale Steuerung vor, so dass sich “NSTEUER“ zu 1 ergibt.

```
SUBROUTINE MINFKT(X,U,T,MIN,NDGL,NSTEUER,NDISKRET)
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  DOUBLE PRECISION MIN,LGINT
  DIMENSION U(NSTEUER,NDISKRET),X(NDGL,NDISKRET),T(NDISKRET)
```

```
  MIN = X(3,NDISKRET)
```

```
  RETURN
```

```
  END
```

C -----

```
SUBROUTINE INTEGRAL(INT,X,U,T,NDGL,NSTEUER)
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  DOUBLE PRECISION INT
  DIMENSION U(NSTEUER),X(NDGL)
```

```

RETURN
END
C -----
SUBROUTINE DGLSYS(X,U,T,DX,NDGL,NSTEUER)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION X(NDGL),U(NSTEUER),DX(NDGL)

DX(1) = X(1)**3.0D0-X(1)**3.0D0-X(1)*DSQRT(X(1)**4.0D0+1.0D0)
DX(2) = X(2)**3.0D0 + U(1)
DX(3) = ((X(2)-X(1))**2.0D0)/2.0D0

RETURN
END
C -----
SUBROUTINE ANFANGSWERTE(XMPC,NDGL,NDISKRET)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION XMPC(NDGL,NDISKRET)

XMPC(1,1) = 1.0D0
XMPC(2,1) = 1.0D0
XMPC(3,1) = 0.0D0

RETURN
END
C -----
SUBROUTINE HORIZON(TM)
IMPLICIT DOUBLE PRECISION (A-H, O-Z)

TM = 1.2D0

RETURN
END
C -----
SUBROUTINE UANFANGSW(X,U,NDGL,NDIS1,NSTEUER,TNDIS1)
IMPLICIT DOUBLE PRECISION (A-H, O-Z)
PARAMETER (EULERSCHRITT = 10.0D0)
DIMENSION U(NSTEUER,NDIS1),X(NDGL,NDIS1),
1      XHELP(NDGL,EULERSCHRITT)

XHELP(1,1) = X(1,1)

```

```
DO 520 I=2,NDIS1
DO 521 J=2,EULERSCHRITT
    XHELP(1,J) = XHELP(1,J-1)+(TNDIS1/(NDIS1*EULERSCHRITT))*
1        (XHELP(1,J-1)**3.0DO - XHELP(1,J-1)**3.0DO -
2        XHELP(1,J-1)*DSQRT(XHELP(1,J-1)**4.0DO + 1.0DO))
521 CONTINUE
X(1,I) = XHELP(1,EULERSCHRITT)
XHELP(1,1) = X(1,I)
U(1,I) = -X(1,I)**3.0DO - X(1,I)*DSQRT(X(1,I)**4.0DO+1.0DO)
520 CONTINUE
DO 522 I=2,NDIS1
    DO 522 J=1,NDGL
        X(J,I) = 0.0DO
522 CONTINUE

RETURN
END
```

```
C -----
SUBROUTINE ANFANGSW(AWX,UNKNOWN,NDGL,NUNBE)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION AWX(NDGL),UNKNOWN(NUNBE)

OPEN(11,FILE='AWX.DAT')
DO 511 I=1,NDGL
    READ(11,*) AWX(I)
511 CONTINUE
CLOSE(11)

RETURN
END
```

```
C -----
SUBROUTINE RANDBED(X,U,T,R,NDGL,NSTEUER,NDISKRET,NRAND)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION X(NDGL,NDISKRET),U(NSTEUER,NDISKRET),R(NRAND)
DIMENSION T(NDISKRET)

RETURN
END
```

```
C -----
SUBROUTINE NEBENBED(X,U,T,CON,NDGL,NSTEUER,NNEBEN)
```

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X (NDGL) ,CON (NNEBEN) ,U (NSTEUER)

      RETURN
      END
C -----
      SUBROUTINE CONBOXES (NSTEUER, NDISKRET, NNEBEN, BL, BU, BLCON, BUCON, T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION BL (NSTEUER) ,BU (NSTEUER) ,BLCON (NNEBEN) ,BUCON (NNEBEN)

      RETURN
      END
C -----
      SUBROUTINE MAS (NDGL, AM, LMAS, RPAR, IPAR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION AM (LMAS, NDGL)
      COMMON /MASS /IMAS, MLMAS, MUMAS, IDX1DIM, IDX2DIM, IDX3DIM, MLJAC, MUJAC

      IMAS   = 0
      MLMAS  = NDGL
      MUMAS  = NDGL
      MLJAC  = NDGL
      MUJAC  = NDGL

      IDX1DIM = NDGL
      IDX2DIM = 0
      ICX3DIM = 0

      DO 200 I=1, NDGL
         AM (I-I+MUMAS+1, I) = 1.0D0
200  CONTINUE

      RETURN
      END
```

B.3 Programm zu Beispiel 6.2

Der Parameter "NSTEUER" ist auf 1 zu setzen, da es sich um ein Problem mit eindimensionaler Steuerung handelt. Ebenso ist "NDGL" zu ändern, da es sich nicht mehr um ein eindimensionales sondern ein zweidimensionales Problem handelt. Um das Zielfunktional kompakt schreiben zu können wurde eine weitere Differentialgleichung hinzugefügt, so dass nun für "NDGL" der Wert 5 einzutragen ist.

```
SUBROUTINE MINFKT(X,U,T,MIN,NDGL,NSTEUER,NDISKRET)
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  DOUBLE PRECISION MIN,LAGINT
  DIMENSION U(NSTEUER,NDISKRET),X(NDGL,NDISKRET),T(NDISKRET)

  MIN = X(5,NDISKRET)

  RETURN
END
```

```
C -----
SUBROUTINE INTEGRAL(INT,X,U,T,NDGL,NSTEUER)
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  DOUBLE PRECISION INT
  DIMENSION U(NSTEUER),X(NDGL)

  RETURN
END
```

```
C -----
SUBROUTINE DGLSYS(X,U,T,DX,NDGL,NSTEUER)
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  DIMENSION X(NDGL),U(NSTEUER),DX(NDGL)

  DX(1) = -X(3) - 3.0D0/2.0D0*(X(1))**2.0D0 - 1.0D0/2.0D0*
1      (X(1))**3.0D0
  DX(2) = -X(4) - 3.0D0/2.0D0*(X(2))**2.0D0 - 1.0D0/2.0D0*
1      (X(2))**3.0D0
  DX(3) = 7.0D0*X(1) - 5.0D0*X(3)
  DX(4) = -U(1)
  DX(5) = 1.0D0/2.0D0*(X(2)-X(1))**2.0D0 + 1.0D0/4.0D0*
1      (X(2)-X(1))**4.0D0 + 1.0D0/2.0D0*(X(4)-X(3) -
2      2.0D0*(X(2)-X(1)))**2.0D0
C  DX(5) = 1.0D0/2.0D0*((X(4)-X(3))**2.0D0+(X(2)-X(1))**2.0D0)

  RETURN
```

```

END
C -----
SUBROUTINE ANFANGSWERTE(XMPC,NDGL,NDISKRET)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION XMPC(NDGL,NDISKRET)

XMPC(1,1) = 22.0D0
XMPC(2,1) = 22.0D0
XMPC(3,1) = 21.0D0
XMPC(4,1) = 21.0D0
XMPC(5,1) = 0.0D0

RETURN
END
C -----
SUBROUTINE HORIZON(TM)
IMPLICIT DOUBLE PRECISION (A-H, O-Z)

TM = 1.0D-1

RETURN
END
C -----
SUBROUTINE UANFANGSW(X,U,NDGL,NDIS1,NSTEUER,TNDIS1)
IMPLICIT DOUBLE PRECISION (A-H, O-Z)
PARAMETER (EULERSCHRITT = 10.0D0)
DIMENSION U(NSTEUER,NDIS1),X(NDGL,NDIS1),
1      XHELP(NDGL,EULERSCHRITT)

U(1,1) = -(7.0D0*X(1,1) - 5.0D0*X(3,1))
XHELP(1,1) = X(1,1)
UHELP(1,1) = X(3,1)
DO 520 I=2,NDIS1
DO 521 J=2,EULERSCHRITT
XHELP(1,J) = XHELP(1,J-1)+(TNDIS1/(NDIS1*EULERSCHRITT))*
1      (-UHELP(1,J-1)-3.0D0/2.0D0*(XHELP(1,J-1))**2.0D0-
2      1.0D0/2.0D0*(XHELP(1,J-1))**3.0D0)
UHELP(1,J) = UHELP(1,J-1)+(TNDIS1/(NDIS1*EULERSCHRITT))*
1      (7.0D0*XHELP(1,J-1)-5.0D0*UHELP(1,J-1))
521 CONTINUE

```

```
      U(1,I) = -(7.0DO * XHELP(1,EULERSCHRITT) - 5.0DO *
1          UHELP(1,EULERSCHRITT))
      UHELP(1,1) = UHELP(1,EULERSCHRITT)
      X(1,I) = XHELP(1,EULERSCHRITT)
      XHELP(1,1) = X(1,I)
520  CONTINUE
      DO 522 I=2,NDIS1
          DO 522 J=1,NDGL
              X(J,I) = 0.0DO
522  CONTINUE

      RETURN
      END

C -----
      SUBROUTINE ANFANGSW(AWX,UNKNOWN,NDGL,NUNBE)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION AWX(NDGL),UNKNOWN(NUNBE)

      OPEN(11,FILE='AWX.DAT')
      DO 511 I=1,NDGL
          read(11,*) AWX(I)
511  CONTINUE
      CLOSE(11)

      RETURN
      END

C -----
      SUBROUTINE RANDBED(X,U,T,R,NDGL,NSTEUER,NDISKRET,NRAND)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(NDGL,NDISKRET),U(NSTEUER,NDISKRET),R(NRAND)
      DIMENSION T(NDISKRET)

      RETURN
      END

C -----
      SUBROUTINE NEBENBED(X,U,T,CON,NDGL,NSTEUER,NNEBEN)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(NDGL),CON(NNEBEN),U(NSTEUER)

      RETURN
```

```
END
C -----
SUBROUTINE CONBOXES(NSTEUER,NDISKRET,NNEBEN,BL,BU,BLCON,BUCON,T)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION BL(NSTEUER),BU(NSTEUER),BLCON(NNEBEN),BUCON(NNEBEN)

RETURN
END
C -----
SUBROUTINE MAS(NDGL,AM,LMAS,RPAR,IPAR)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION AM(LMAS,NDGL)
COMMON/MASS/IMAS,MLMAS,MUMAS,IDX1DIM,IDX2DIM,IDX3DIM,MLJAC,MUJAC

IMAS = 0
MLMAS = NDGL
MUMAS = NDGL
MLJAC = NDGL
MUJAC = NDGL

IDX1DIM = NDGL
IDX2DIM = 0
IDX3DIM = 0

DO 200 I=1,NDGL
  AM(I-I+MUMAS+1,I) = 1.0D0
200 CONTINUE

RETURN
END
```

B.4 Programm zu Beispiel 6.3

Für das Beispiel 6.3 ist wegen der eindimensionalen Steuerung "NSTEUER" auf 1 zu setzen. Dabei besteht das Problem aus je drei Differentialgleichungen für das kontinuierlich gesteuerte und das zu steuernde System. Für "NDGL" ergibt sich damit wegen der auch hier vorgenommenen Ergänzung einer Differentialgleichung zur Verwendung des Mayer-Zielfunktional der Wert 7.

```

SUBROUTINE MINFKT(X,U,T,MIN,NDGL,NSTEUER,NDISKRET)
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  DOUBLE PRECISION MIN,LAGINT
  DIMENSION U(NSTEUER,NDISKRET),X(NDGL,NDISKRET),T(NDISKRET)

  MIN = X(7,NDISKRET)

  RETURN
END

```

```

C -----
SUBROUTINE INTEGRAL(INT,X,U,T,NDGL,NSTEUER)
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  DOUBLE PRECISION INT
  DIMENSION U(NSTEUER),X(NDGL)

  RETURN
END

```

```

C -----
SUBROUTINE DGLSYS(X,U,T,DX,NDGL,NSTEUER)
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  DIMENSION X(NDGL),U(NSTEUER),DX(NDGL)

  PARAMETER(C1 = 1.0D0,
1          C2 = 2.0D0,
2          SIGMA = 2.0D0)

  DX(1) = - X(3) - 3.0D0/2.0D0 * X(1)**2.0D0 - 1.0D0/2.0D0 *
1          X(1)**3.0D0 - 3.0D0 * X(5) * X(1) - 3.0D0 * X(5)
  DX(2) = - X(4) - 3.0D0/2.0D0 * X(2)**2.0D0 - 1.0D0/2.0D0 *
1          X(2)**3.0D0 - 3.0D0 * X(6) * X(2) - 3.0D0 * X(6)
  DX(3) = X(1) - C2 * X(3) + C1 * C2 * X(1) - 3.0D0/2.0D0 * C2 *
1          X(1)**2.0D0 - 3.0D0 * C2 * X(5) + (C1 - 3.0D0 * X(1))
2          * (-X(3) - 3.0D0/2.0D0 * X(1)**2.0D0 - 1.0D0/2.0D0 *
3          X(1)**3.0D0 - 3.0D0 * X(5) * X(1) - 3.0D0 * X(5)) -

```

```

4      3.0D0 * (-SIGMA * X(5) * (X(5) + 2.0D0 * X(1) + X(1)**
5      2.0D0))
DX(4) = -U(1)
DX(5) = -SIGMA * X(5) * (X(5) + 2.0D0 * X(1) + X(1)**2.0D0)
DX(6) = -SIGMA * X(6) * (X(6) + 2.0D0 * X(2) + X(2)**2.0D0)
DX(7) = 1.0D0/2.0D0*((X(4)-X(3))**2.0D0+(X(2)-X(1))**2.0D0)

RETURN
END
C -----
SUBROUTINE ANFANGSWERTE(XMPC,NDGL,NDISKRET)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION XMPC(NDGL,NDISKRET)

XMPC(1,1) = 6.0D0
XMPC(2,1) = 6.0D0
XMPC(3,1) = 25.0D0
XMPC(4,1) = 25.0D0
XMPC(5,1) = 1.0D0
XMPC(6,1) = 1.0D0
XMPC(7,1) = 0.0D0

RETURN
END
C -----
SUBROUTINE HORIZON(TM)
IMPLICIT DOUBLE PRECISION (A-H, O-Z)

TM = 2.0D-2

RETURN
END
C -----
SUBROUTINE UANFANGSW(X,U,NDGL,NDIS1,NSTEUER,TNDIS1)
IMPLICIT DOUBLE PRECISION (A-H, O-Z)

PARAMETER(EULERSCHRITT = 1000.0D0,
1      C1 = 1.0D0,
2      C2 = 2.0D0,
3      SIGMA = 2.0D0)

```

```

DIMENSION U(NSTEUER,NDIS1),X(NDGL,NDIS1),
1      UHELP(NSTEUER,EULERSCHRITT),XHELP(NDGL,EULERSCHRITT)

XHELP(1,1) = X(1,1)
XHELP(3,1) = X(3,1)
XHELP(5,1) = X(5,1)
U(1,1) = -(X(1,1)-C2*X(3,1)+C1*C2*X(1,1)-3.0D0/2.0D0*C2*X(1,1)**
1      2.0D0-3.0D0*C2*X(5,1)+(C1-3.0D0*X(1,1))*(-X(3,1)-
2      3.0D0/2.0D0*X(1,1)**2.0D0-1.0D0/2.0D0*X(1,1)**3.0D0-
3      3.0D0*X(1,5)*X(1,1)-3.0D0*X(5,1))-3.0D0*(-SIGMA*X(5,1)
4      *(X(5,1)+2.0D0*X(1,1)+X(1,1)**2.0D0))

DO 520 I=2,NDIS1
DO 521 J=2,EULERSCHRITT
XHELP(1,J) = XHELP(1,J-1)+(TNDIS1/(NDIS1*EULERSCHRITT))*
1      (-XHELP(3,J-1)-3.0D0/2.0D0*XHELP(1,J-1)**2.0D0-
2      1.0D0/2.0D0*XHELP(1,J-1)**3.0D0-3.0D0*XHELP(5,J-1)
3      *XHELP(1,J-1)-3.0D0*XHELP(5,J-1))
XHELP(3,J) = XHELP(3,J-1)+(TNDIS1/(NDIS1*EULERSCHRITT))*
1      (XHELP(1,J-1)-C2*XHELP(3,J-1)+C1*C2*XHELP(1,J-1)-
2      3.0D0/2.0D0*C2*XHELP(1,J-1)**2.0D0-3.0D0*C2*XHELP
3      (5,J-1)+(C1-3.0D0*XHELP(1,J-1))*(-XHELP(3,J-1)-
4      3.0D0/2.0D0*XHELP(1,J-1)**2.0D0-1.0D0/2.0D0*XHELP
5      (1,J-1)**3.0D0-3.0D0*XHELP(5,J-1)*XHELP(1,J-1)-
6      3.0D0*XHELP(5,J-1))-3.0D0*(-SIGMA*XHELP(5,J-1)*
7      (XHELP(5,J-1)+2.0D0*XHELP(1,J-1)+XHELP(1,J-1)
8      **2.0D0))
XHELP(5,J) = XHELP(5,J-1)+(TNDIS1/(NDIS1*EULERSCHRITT))*
1      (-SIGMA*XHELP(5,J-1)*(XHELP(5,J-1)+2.0D0*
2      XHELP(1,J-1)+XHELP(1,J-1)**2.0D0))
521 CONTINUE
U(1,I) = -(XHELP(1,EULERSCHRITT)-C2*XHELP(3,EULERSCHRITT)+C1*C2
1      *XHELP(1,EULERSCHRITT)-3.0D0/2.0D0*C2*XHELP(1,
2      EULERSCHRITT)**2.0D0-3.0D0*C2*XHELP(5,EULERSCHRITT)
3      +(C1-3.0D0*XHELP(1,EULERSCHRITT))*
2      (-XHELP(3,EULERSCHRITT)-3.0D0/2.0D0*XHELP(1,
3      EULERSCHRITT)**2.0D0-1.0D0/2.0D0*XHELP(1,
4      EULERSCHRITT)**3.0D0-3.0D0*XHELP(5,EULERSCHRITT)*
5      XHELP(1,EULERSCHRITT)-3.0D0*XHELP(5,EULERSCHRITT))-
6      3.0D0*(-SIGMA*XHELP(5,EULERSCHRITT)*(XHELP(5,

```

```

7          EULERSCHRITT)+2.0D0*XHELP(1,EULERSCHRITT)+XHELP(1,
8          EULERSCHRITT)**2.0D0)))
X(1,I) = XHELP(1,EULERSCHRITT)
X(3,I) = XHELP(3,EULERSCHRITT)
X(5,I) = XHELP(5,EULERSCHRITT)
XHELP(1,1) = X(1,I)
XHELP(3,1) = X(3,I)
XHELP(5,1) = X(5,I)
520 CONTINUE
DO 522 I=2,NDIS1
  DO 522 J=1,NDGL
    X(J,I) = 0.0D0
522 CONTINUE

RETURN
END
C -----
SUBROUTINE ANFANGSW(AWX,UNKNOWN,NDGL,NUNBE)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION AWX(NDGL),UNKNOWN(NUNBE)

OPEN(11,FILE='AWX.DAT')
DO 511 I=1,NDGL
  READ(11,*) AWX(I)
511 CONTINUE
CLOSE(11)

RETURN
END
C -----
SUBROUTINE RANDBED(X,U,T,R,NDGL,NSTEUER,NDISKRET,NRAND)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION X(NDGL,NDISKRET),U(NSTEUER,NDISKRET),R(NRAND)
DIMENSION T(NDISKRET)

RETURN
END
C -----
SUBROUTINE NEBENBED(X,U,T,CON,NDGL,NSTEUER,NNEBEN)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)

```

```
DIMENSION X(NDGL),CON(NNEBEN),U(NSTEUER)
```

```
RETURN
```

```
END
```

```
C -----
```

```
SUBROUTINE CONBOXES(NSTEUER,NDISKRET,NNEBEN,BL,BU,BLCON,BUCON,T)
```

```
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

```
DIMENSION BL(NSTEUER),BU(NSTEUER),BLCON(NNEBEN),BUCON(NNEBEN)
```

```
RETURN
```

```
END
```

```
C -----
```

```
SUBROUTINE MAS(NDGL,AM,LMAS,RPAR,IPAR)
```

```
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

```
DIMENSION AM(LMAS,NDGL)
```

```
COMMON/MASS/IMAS,MLMAS,MUMAS,IDX1DIM,IDX2DIM,IDX3DIM,MLJAC,MUJAC
```

```
IMAS = 0
```

```
MLMAS = NDGL
```

```
MUMAS = NDGL
```

```
MLJAC = NDGL
```

```
MUJAC = NDGL
```

```
IDX1DIM = NDGL
```

```
IDX2DIM = 0
```

```
IDX3DIM = 0
```

```
DO 200 I=1,NDGL
```

```
AM(I-I+MUMAS+1,I) = 1.0D0
```

```
200 CONTINUE
```

```
RETURN
```

```
END
```

Anhang C

Visualisierung

C.1 Eingabe der Differentialgleichungen

Für das Beispiel 6.1 ergibt sich für das Kontrollsystem mit eingesetzter kontinuierlicher Steuerung folgende Matlab-Schreibweise:

```
function dy = mpc1(t,y)
dy = zeros(1,1);
dy(1) = - y(1)*(y(1)^4+1)^(1/2);
```

Im Fall von Beispiel 6.2 verwendet man:

```
function dy = mpc2(t,y)
dy = zeros(2,1);
dy(1) = -y(2)-3/2*y(1)^2-1/2*y(1)^3;
dy(2) = 7*y(1)-5*y(2);
```

Und für das Beispiel 6.3 schreibt sich das Differentialgleichungssystem wie folgt:

```
function dy = mpc3(t,y)
dy = zeros(3,1);
dy(1) = -y(2)-3/2*y(1)^2-1/2*y(1)^3-3*y(3)*y(1)-3*y(3);
dy(2) = y(1)-2*(y(2)-y(1)+3/2*y(1)^2+3*y(3))
      +(1-2*y(1))*(-y(2)-3/2*y(1)^2-1/2*y(1)^3-3*y(3)*y(1)-3*y(3))
      -3*(-2*y(3)*(y(3)+2*y(1)+y(1)^2));
dy(3) = -2*y(3)*(y(3)+2*y(1)+y(1)^2);
```

Diese Funktionen sind in den Dateien “mpc1.m“, “mpc2.m“ und “mpc3.m“ enthalten.

Zur Visualisierung der gesampten zeitkontinuierlichen Steuerung für Beispiel 6.1 werden die Dateien “mpc1dis.m“,

```
function dy = mpc1dis(t,y)
dy = zeros(2,1);
dy(1) = y(1)^3.0 + y(2);
dy(2) = 0;
```

“mpc2dis.m“ für Beispiel 6.2

```
function dy = mpc2dis(t,y)
dy = zeros(3,1);
dy(1) = -y(2) - 3.0/2.0 * y(1)^2.0 - 1.0/2.0 * y(1)^3.0;
dy(2) = -y(3);
dy(3) = 0;
```

und “mpc3dis.m“ für Beispiel 6.3 verwendet.

```
function dy = mpc3dis(t,y)
dy = zeros(4,1);
c1 = 1.0;
c2 = 2.0;
sigma = 2.0;
dy(1) = - y(2) - 3.0/2.0 * y(1)^2.0 - 1.0/2.0 * y(1)^3.0 - 3.0 *
        y(3) * y(1) - 3.0 * y(3);
dy(2) = - y(4);
dy(3) = - sigma * y(3) * (y(3) + 2.0 * y(1) + y(1)^2.0);
dy(4) = 0;
```

C.2 Matlab Programm zur Visualisierung

Die Programme zur Visualisierung der Ergebnisse der drei Beispiele unterscheiden sich lediglich darin, dass

- der Vektor $Z(1, :)$ die jeweiligen Anfangswerte enthalten muss,
- beim Aufruf von “ode45“ die Funktionen für das entsprechende Beispiel einzutragen sind und
- der Vektor $Z(i, *)$ entsprechend der vorliegenden zeitkontinuierlichen Steuerung zu berechnen ist. Dabei ist $*$ = $\frac{NDGL-1}{2} + 1$ zu setzen.

Zudem sind hier und in den Dateien der Differentialgleichungen die jeweils verwendeten Parameter des Problems anzugeben.

```
clear all;
%Einlesen der Problemdaten
A = load('Problemdaten.dat');
NDIS1 = A(1);
NDIS2 = A(2);
TM = A(3);
TOL = A(4);
NART = A(5);
NDGL = A(7);
```

```

NSTEUER = A(8);

%Berechnung der zeitkontinuierlichen Loesung
Z(1,:) = [6, 25, 1];

options = odeset('RelTol',1e-4,'AbsTol',1e-4);
[T,Y] = ode45('mpc3',[0, TM*(NDIS2-1)],Z(1,:),options);

%Berechnung der gesampteten Loesung
c1 = 1.0;
c2 = 2.0;
sigma = 2.0;
for i=1:NDIS2-1;
    Z(i,4) = -(Z(i,1) - c2 * (Z(i,2) - c1 * Z(i,1) + 3.0/2.0 *
    Z(i,1)^2.0 + 3.0 * Z(i,3)) + (c1 - 3.0 * Z(i,1)) * (- Z(i,2)
    - 3.0/2.0 * Z(i,1)^2.0 - 1.0/2.0 * Z(i,1)^3.0 - 3.0 * Z(i,3)
    * Z(i,1) - 3.0* Z(i,3)) - 3.0 *
    (- sigma * Z(i,3) * (Z(i,3) + 2.0 * Z(i,1) + Z(i,1)^2.0));
    options = odeset('RelTol',1e-4,'AbsTol',1e-4);
    [Tdis,X] = ode45('mpc3dis',[TM*(i-1),TM*i],Z(i,:),options);
    Xind = size(X);
    Z(i+1,:) = X(Xind(1),:);
end

%Einlesen der Dateien
for i=1:NDGL-1;
    if i>=100
        file=['tmpx',int2str(i),'.dat'];
    elseif i >= 10
        file=['tmpx0',int2str(i),'.dat'];
    else
        file=['tmpx00',int2str(i),'.dat'];
    end
    S=load(file);
    B(:, :, i)=S;
end
for i=1:NSTEUER;
    if i>=100
        file=['tmpu',int2str(i),'.dat'];
    elseif i >= 10

```

```
    file=['tmpu0',int2str(i),'.dat'];
else
    file=['tmpu00',int2str(i),'.dat'];
end
S=load(file);
C(:, :, i)=S;
end

%Fenstereinteilung
if (NDGL-1)/2+NSTEUER >= 5
    vertikal = 2;
    horizontal = 3;
    fensterzahl = mod(((NDGL-1)/2+NSTEUER),6);
elseif (NDGL-1)/2+NSTEUER >= 3
    vertikal = 2;
    horizontal = 2;
    fensterzahl = 1;
else
    vertikal = 1;
    horizontal = 2;
    fensterzahl = 1;
end

%Plot der Problemdaten
figure(1);
newplot;
axis off
text(0.2,0.8,'BEISPIEL 3:')
text(0.2,0.8,'_____')
text(0.2,0.5,['NART = ',num2str(NART)])
text(0.2,0.4,['T = ',num2str(TM)])
text(0.2,0.3,['TOL = ',num2str(TOL)])
text(0.2,0.2,['NDIS1 = ',num2str(NDIS1)])
text(0.2,0.1,['NDIS2 = ',num2str(NDIS2)])
text(0.2,0,'RK8 zur Schrittweitensteuerung')
end
k = 1;
l = 1;
%Plots der Trajektorien
for i=1:fensterzahl;
```

```
figure(i+1);
for j=1:vertikal*horizontal;
if j <= NSTEUER+(NDGL-1)/2
    subplot(vertikal,horizontal,j);
    newplot;
    if k <= NSTEUER
%Plot der sampled-data und gesampteten Steuerung
        stairs(C(:,1,k),C(:,2,k),'b-')
        hold on
        stairs(C(:,1,k),Z(:,4),'r-')
        hold off
        legend(['u_{',int2str(1),'}modellpraediktiv'],
                'MatLab RK45 sampled-data Loesung')
        xlabel('t')
        ylabel(['u_{',int2str(k),'}-Werte'])
        title(['Plot der Steuerungsstrajektorie u_{',int2str(k),'}'])
        grid on
        axis tight
        k = k + 1;
    elseif k > NSTEUER
%Plot der Zustandstrajektorien
        plot(T,Y(:,1),'b-',B(:,1,k-NSTEUER),Z(:,1),'v-',
            B(:,1,k-NSTEUER),B(:,2,k-NSTEUER),'g-',
            B(:,1,k-NSTEUER+1),B(:,2,k-NSTEUER+1),'r--')
        legend('MatLab RK45 Loesung','MatLab RK45 sampled-data Loesung',
            ['x_{',int2str(1),'}modellpraediktiv'],['\xi_{',int2str(1),
            '}modellpraediktiv'])
        xlabel('t')
        ylabel(['x_{',int2str(1),'}-Werte'])
        title(['Plot der Zustandstrajektorien x_{',int2str(1),'} und
            \xi_{',int2str(1),'}'])
        grid on
        axis tight
        l = l + 1;
        k = k + 2;
    end
end
end
end
```

```
%Einlesen der Dateien der einzelnen optimalen Steuerprozesse
pause;
for i=1:NDIS2;
    for j=1:NDGL;
        if i>=100
            if j>=100
                file=['tmp_',int2str(i),'x',int2str(j),'.dat'];
            elseif j >= 10
                file=['tmp_',int2str(i),'x0',int2str(j),'.dat'];
            else
                file=['tmp_',int2str(i),'x00',int2str(j),'.dat'];
            end
        elseif i >= 10
            if j>=100
                file=['tmp_0',int2str(i),'x',int2str(j),'.dat'];
            elseif j >= 10
                file=['tmp_0',int2str(i),'x0',int2str(j),'.dat'];
            else
                file=['tmp_0',int2str(i),'x00',int2str(j),'.dat'];
            end
        else
            if j>=100
                file=['tmp_00',int2str(i),'x',int2str(j),'.dat'];
            elseif j >= 10
                file=['tmp_00',int2str(i),'x0',int2str(j),'.dat'];
            else
                file=['tmp_00',int2str(i),'x00',int2str(j),'.dat'];
            end
        end
    end
    S=load(file);
    X(:, :, i, j)=S;
end
for j=1:NSTEUER;
    if i>=100
        if j>=100
            file=['tmp_',int2str(i),'u',int2str(j),'.dat'];
        elseif j >= 10
            file=['tmp_',int2str(i),'u0',int2str(j),'.dat'];
        else
            file=['tmp_',int2str(i),'u00',int2str(j),'.dat'];
        end
    end
end
```

```

    end
elseif i >= 10
    if j>=100
        file=['tmp_0',int2str(i),'u',int2str(j),'.dat'];
    elseif j >= 10
        file=['tmp_0',int2str(i),'u0',int2str(j),'.dat'];
    else
        file=['tmp_0',int2str(i),'u00',int2str(j),'.dat'];
    end
end
else
    if j>=100
        file=['tmp_00',int2str(i),'u',int2str(j),'.dat'];
    elseif j >= 10
        file=['tmp_00',int2str(i),'u0',int2str(j),'.dat'];
    else
        file=['tmp_00',int2str(i),'u00',int2str(j),'.dat'];
    end
end
end
S=load(file);
U(:, :, i, j)=S;
end
end

%Plot der einzelnen Trajektorien
for m=1:NDIS2
    k = 1;
    l = 1;
    for i=1:fensterzahl;
        figure(i+1);
        for j=1:vertikal*horizontal;
            if j <= NSTEUER+(NDGL-1)/2
                subplot(vertikal,horizontal,j);
                newplot;
                if k <= NSTEUER
                    stairs(U(:, 1, m, k), U(:, 2, m, k))
                    xlabel('t')
                    ylabel(['u_{', int2str(k), '} -Werte'])
                    title(['Plot der Steuerungsstrajektorie u_{', int2str(k), '}
                        des ', int2str(m), '. OCPs'])
                end
            end
        end
        grid on
    end
end

```

```
axis tight
k = k + 1;
%mit getframe Generierung von Animationen der Entwicklung der
%Steuerungen moeglich
elseif k > NSTEUER
    plot(X(:,1,m,k-NSTEUER),X(:,2,m,k-NSTEUER),'-.',
         X(:,1,m,k-NSTEUER+1),X(:,2,m,k-NSTEUER+1),'--')
    legend(['x_{',int2str(1),'} modellpraediktiv'],
          ['\xi_{',int2str(1),'} modellpraediktiv'])
    xlabel('t')
    ylabel(['x_{',1,'}-Werte'])
    title(['Plot der Zustandstrajektorien x_{',int2str(1),'}
           und \xi_{',int2str(1),'} des ',int2str(m),'. OCPs'])
    grid on
    axis tight
    l = l + 1;
    k = k + 2;
%mit getframe Generierung von Animationen der Entwicklung der
%Zustandstrajektorien moeglich
end
end
end
end
end
```

In ersten Linie gibt dieses Programm eine Übersicht über die verwendeten Einstellungen bei der Lösung mit dem modellprädiktiven Verfahren aus.

In weiteren Fenstern werden die Trajektorien derart geplottet, dass die zueinander gehörenden zeitkontinuierlichen und optimal gesteuerten Zustände x_i und ξ_i zusammen abgebildet werden. Hinzu kommen noch die mittels Matlab für die zeitkontinuierliche Steuerung berechnete Trajektorie und die Trajektorie der gesampleten zeitkontinuierlichen Steuerung. Ebenfalls werden die stückweise konstanten Funktionen der optimalen Steuerung und der gesampleten zeitkontinuierlichen Steuerung zusammen angezeigt, so dass hiermit ein Vergleich gut möglich ist.

C.3 Fehlerberechnung

Für die einzelnen Beispiele lassen sich verschiedene Fehler betrachten, so etwa die absoluten Abstände zwischen den Lösungen des zeitkontinuierlich gesteuerten Problems, zum einen mittels der Matlab-Routine und zum anderen mit der Integrationsroutine von NUDOC- CS berechnet, und den Lösungen des mit der modellpräditiv Regelungsansatz erhaltenen sampled-data closed-loop Systems.

Da die mit Matlab berechneten Lösungen als verlässlich angesehen werden können erfolgen die Vergleiche immer mit dieser Referenz.

```
clear all;
%Einlesen der Problemdatei
A = load('Problemdatei.dat');
NDIS1 = A(1);
NDIS2 = A(2);
TM = A(3);
TOL = A(4);
NART = A(5);
NDGL = A(7);
NSTEUER = A(8);

%Berechnung der zeitkontinuierlichen Lösung
Z(1,:) = [6, 25, 1];

options = odeset('RelTol',1e-4,'AbsTol',1e-4);
[T,Y] = ode45('mpc3',[0, TM*(NDIS2-1)+1],Z(1,:),options);

%Berechnung der gesampleten Lösung
c1 = 1.0;
c2 = 2.0;
sigma = 2.0;
for i=1:NDIS2-1;
    Z(i,4) = -(Z(i,1) - c2 * (Z(i,2) - c1 * Z(i,1) + 3.0/2.0 *
    Z(i,1)^2.0 + 3.0 * Z(i,3)) + (c1 - 3.0 * Z(i,1)) * (- Z(i,2)
    - 3.0/2.0 * Z(i,1)^2.0 - 1.0/2.0 * Z(i,1)^3.0 - 3.0 * Z(i,3) *
    Z(i,1) - 3.0 * Z(i,3)) - 3.0 * (- sigma * Z(i,3) * (Z(i,3) +
    2.0 * Z(i,1) + Z(i,1)^2.0));
    options = odeset('RelTol',1e-4,'AbsTol',1e-4);
    [Tdis,Xdis] = ode45('mpc3dis',[TM*(i-1), TM*i],Z(i,:),options);
    Xind = size(Xdis);
    Z(i+1,:) = Xdis(Xind(1),:);
end
```

```
%Einlesen der Dateien
for i=1:NDGL-1;
    if i>=100
        file=['tmpx',int2str(i),'.dat'];
    elseif i >= 10
        file=['tmpx0',int2str(i),'.dat'];
    else
        file=['tmpx00',int2str(i),'.dat'];
    end
    S=load(file);
    B(:, :,i)=S;
end for i=1:NSTEUER;
    if i>=100
        file=['tmpu',int2str(i),'.dat'];
    elseif i >= 10
        file=['tmpu0',int2str(i),'.dat'];
    else
        file=['tmpu00',int2str(i),'.dat'];
    end
    S=load(file);
    C(:, :,i)=S;
end

%Fehler zwischen Trajektorien aus NUODOCCCS
Fehler(1) = abs(B(2,2,1) - B(2,2,2)) + abs(B(2,2,3) - B(2,2,4))
           + abs(B(2,2,5) - B(2,2,6));
Fehler(2) = abs(B(NDIS2,2,1) - B(NDIS2,2,2)) + abs(B(NDIS2,2,3)
           - B(NDIS2,2,4)) + abs(B(NDIS2,2,5) - B(NDIS2,2,6));
for j=1:NDIS2;
    MaxFehlerNud(j) = abs(B(j,2,1) - B(j,2,2)) + abs(B(j,2,3)
           - B(j,2,4)) + abs(B(j,2,5) - B(j,2,6));
end
Fehler(3) = max(MaxFehlerNud(2:NDIS2));
Fehler(4) = min(MaxFehlerNud(2:NDIS2));
%Fehler zwischen MatLab und NUODOCCCS
for i=1:size(T);
    if T(i) <= TM
        if T(i+1)>TM
            w = Y(i,:) + (TM-T(i))*(Y(i+1,:) - Y(i,:))/(T(i+1) - T(i));
```

```

        Fehler(5) = abs(B(2,2,2) - w(1)) + abs(B(2,2,4) - w(2))
        + abs(B(2,2,6) - w(3));
    end
elseif T(i)<=(NDIS2-1)*TM
    if T(i+1)>(NDIS2-1)*TM
        w = Y(i,:) + ((NDIS2-1)*TM - T(i))*(Y(i+1,:) - Y(i,:))
            / (T(i+1) - T(i));
        Fehler(6) = abs(B(NDIS2,2,2) - w(1)) + abs(B(NDIS2,2,4)
            - w(2)) + abs(B(NDIS2,2,6) - w(3));
    end
end
end
%Fehler zwischen MatLab und gesampleter Lsg
for i=1:size(T);
    if T(i) <= TM
        if T(i+1)>TM
            w = Y(i,:) + (TM-T(i))*(Y(i+1,:) - Y(i,:))/(T(i+1) - T(i));
            Fehler(7) = abs(Z(2,1) - w(1)) + abs(Z(2,2) - w(2))
                + abs(Z(2,3) - w(3));
        end
    elseif T(i)<=(NDIS2-1)*TM
        if T(i+1)>(NDIS2-1)*TM
            w = Y(i,:) + ((NDIS2-1)*TM - T(i))*(Y(i+1,:)
                - Y(i,:))/(T(i+1)-T(i));
            Fehler(8) = abs(Z(NDIS2,1) - w(1)) + abs(Z(NDIS2,2)
                - w(2)) + abs(Z(NDIS2,3) - w(3));
        end
    end
end
end
%Maximaler Fehler
for i=1:size(T);
    for j=1:NDIS2;
        if T(i) <= (j-1)*TM
            if T(i+1) > (j-1)*TM
%zu untersuchendes Intervall (hier [0.0, 1.0])
                if T(i) <= 1.0
                    w = Y(i,:) + ((j-1)*TM - T(i))*(Y(i+1,:)
                        - Y(i,:))/(T(i+1) - T(i));
                    MaxFehler(j) = abs(B(j,2,2) - w(1)) + abs(B(j,2,4)
                        - w(2)) + abs(B(j,2,6) - w(3));
                end
            end
        end
    end
end

```

```
        MaxFehlersd(j) = abs(Z(j,1) - w(1)) + abs(Z(j,2) - w(2))
                    + abs(Z(j,3) - w(3));
        Abweich(j) = abs(B(j,2,1) - w(1)) + abs(B(j,2,3)
                    - w(2)) + abs(B(j,2,5) - w(3));
    end
end
end
end
end
Fehler(9) = max(MaxFehler);
Fehler(10) = max(MaxFehlersd);
MaxFehler(1) = Fehler(9);
MaxFehlersd(1) = Fehler(10);
[sortAbweich, INDEX] = sort(MaxFehler);
Zahl = size(INDEX);
Fehler(11) = Abweich(INDEX(Zahl(2)));
Fehler(12) = min(MaxFehler);
Fehler(13) = min(MaxFehlersd);
Fehler(14) = Abweich(INDEX(1));
```

C.4 Plots der Beispiele zur Verlaufsanalyse

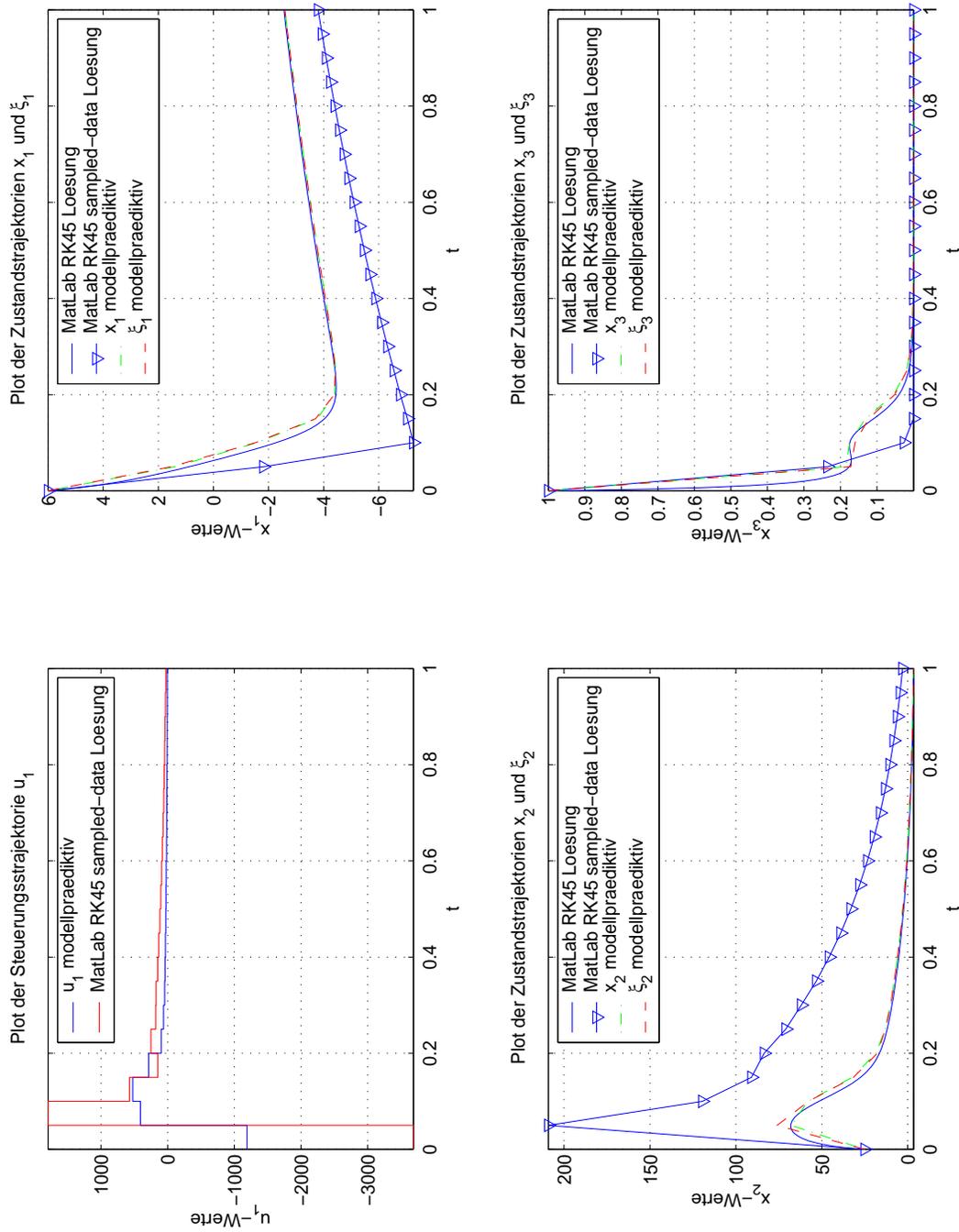


Abbildung C.1: Beispiels 6.3 für $T = 0.05$ und $H = 1.00$

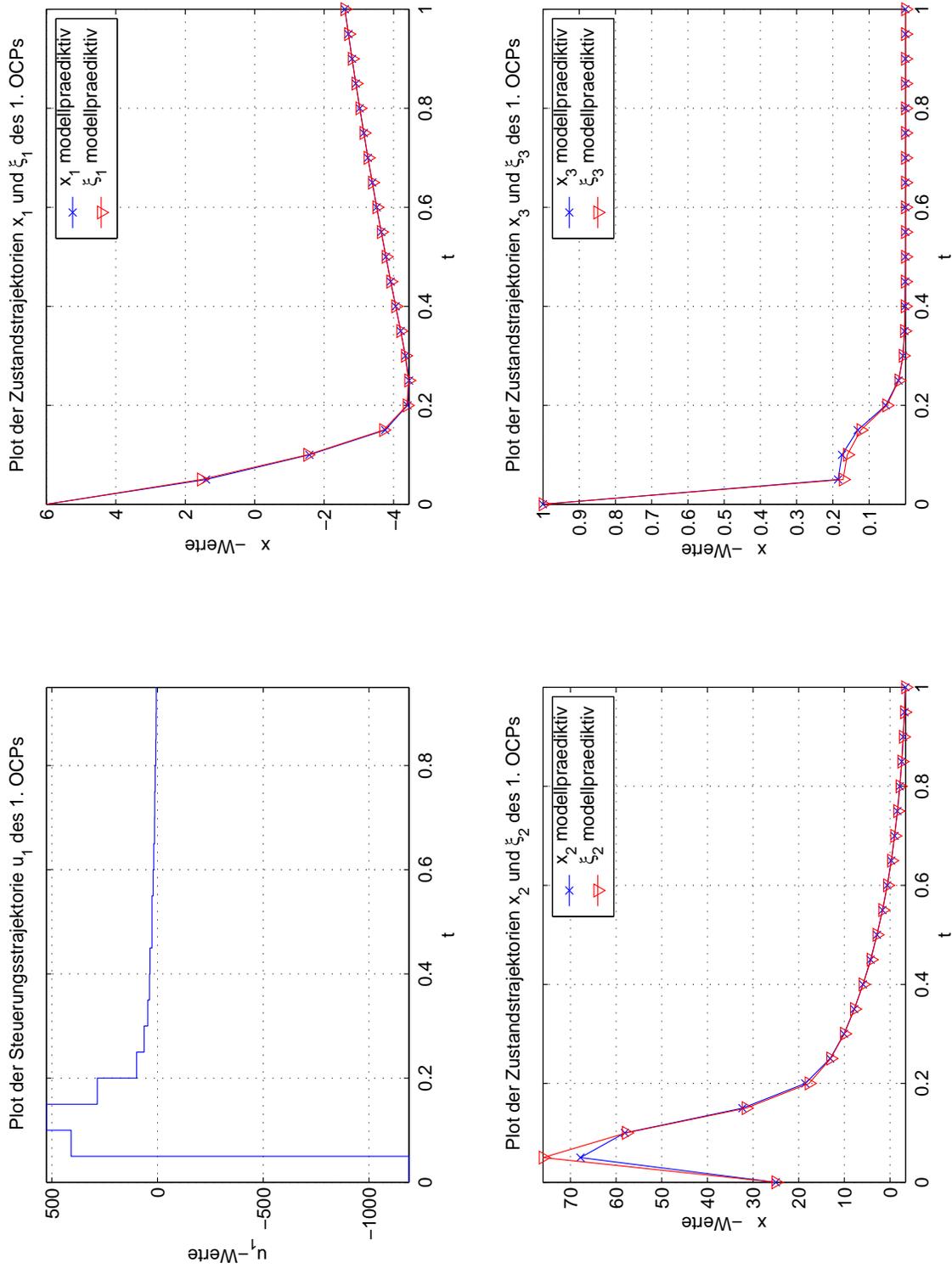


Abbildung C.2: Beispiels 6.3: 1. optimales Steuerungsproblem

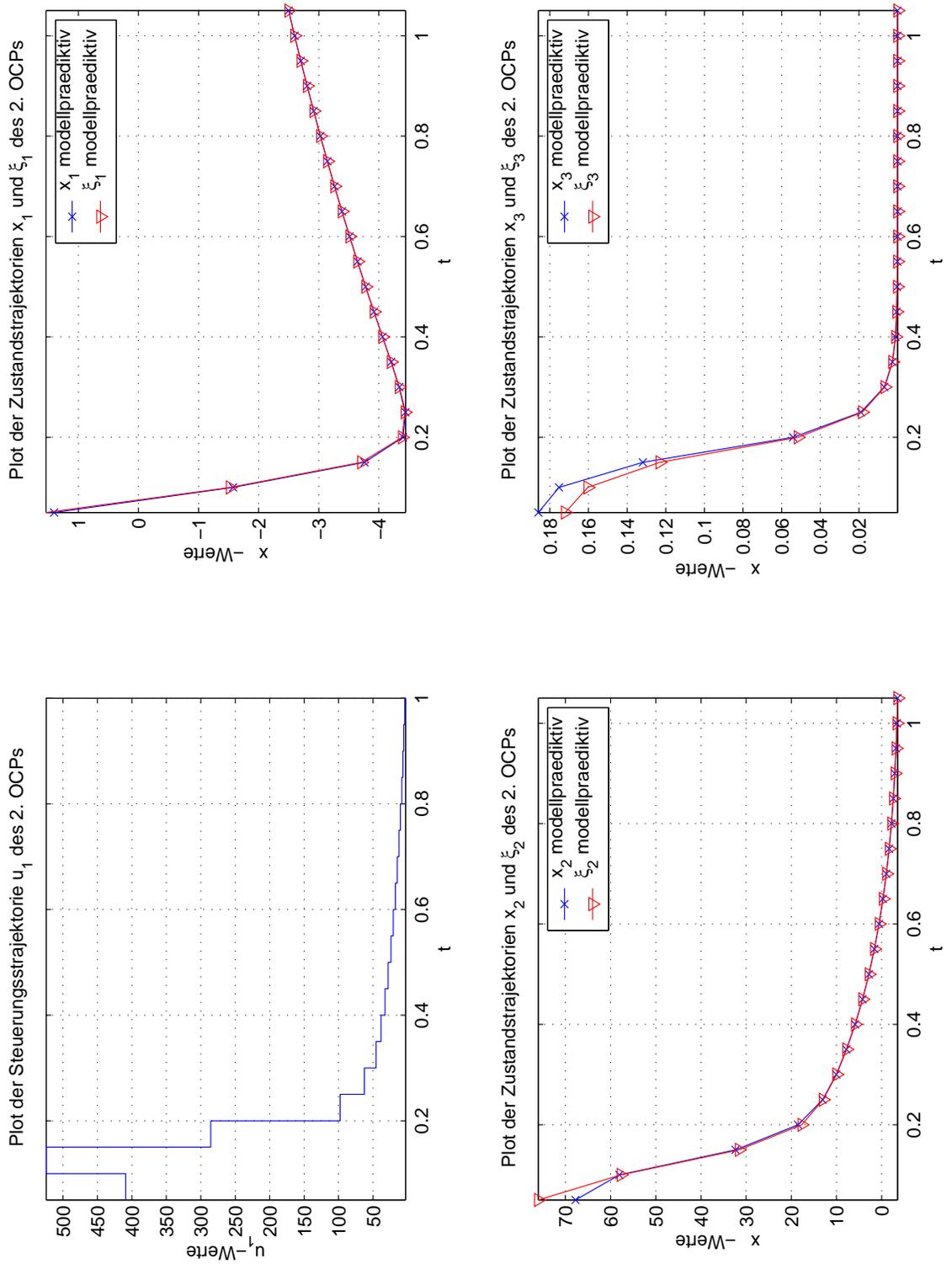


Abbildung C.3: Beispiels 6.3: 2. optimales Steuerungsproblem

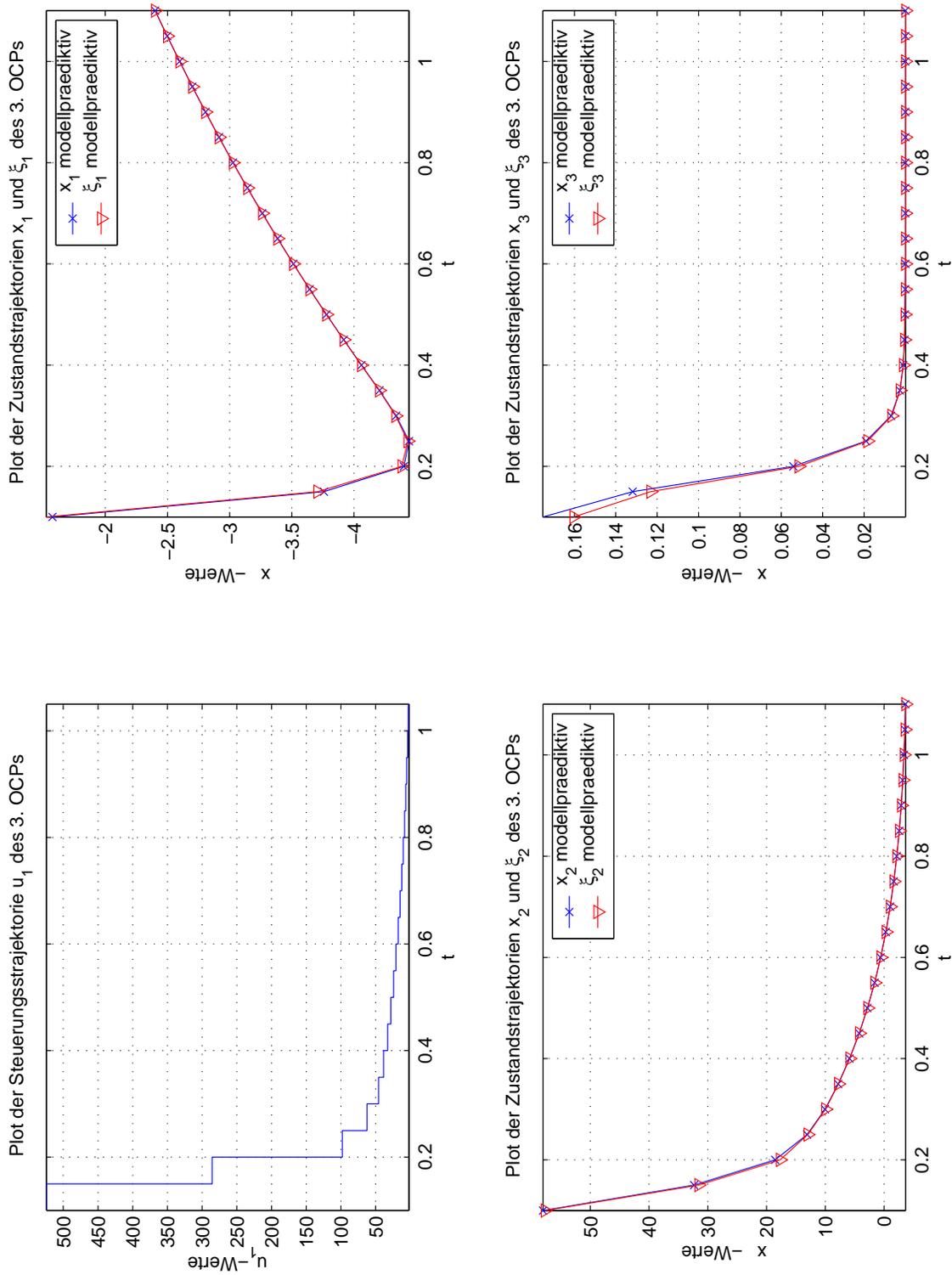


Abbildung C.4: Beispiels 6.3: 3. optimales Steuerungsproblem

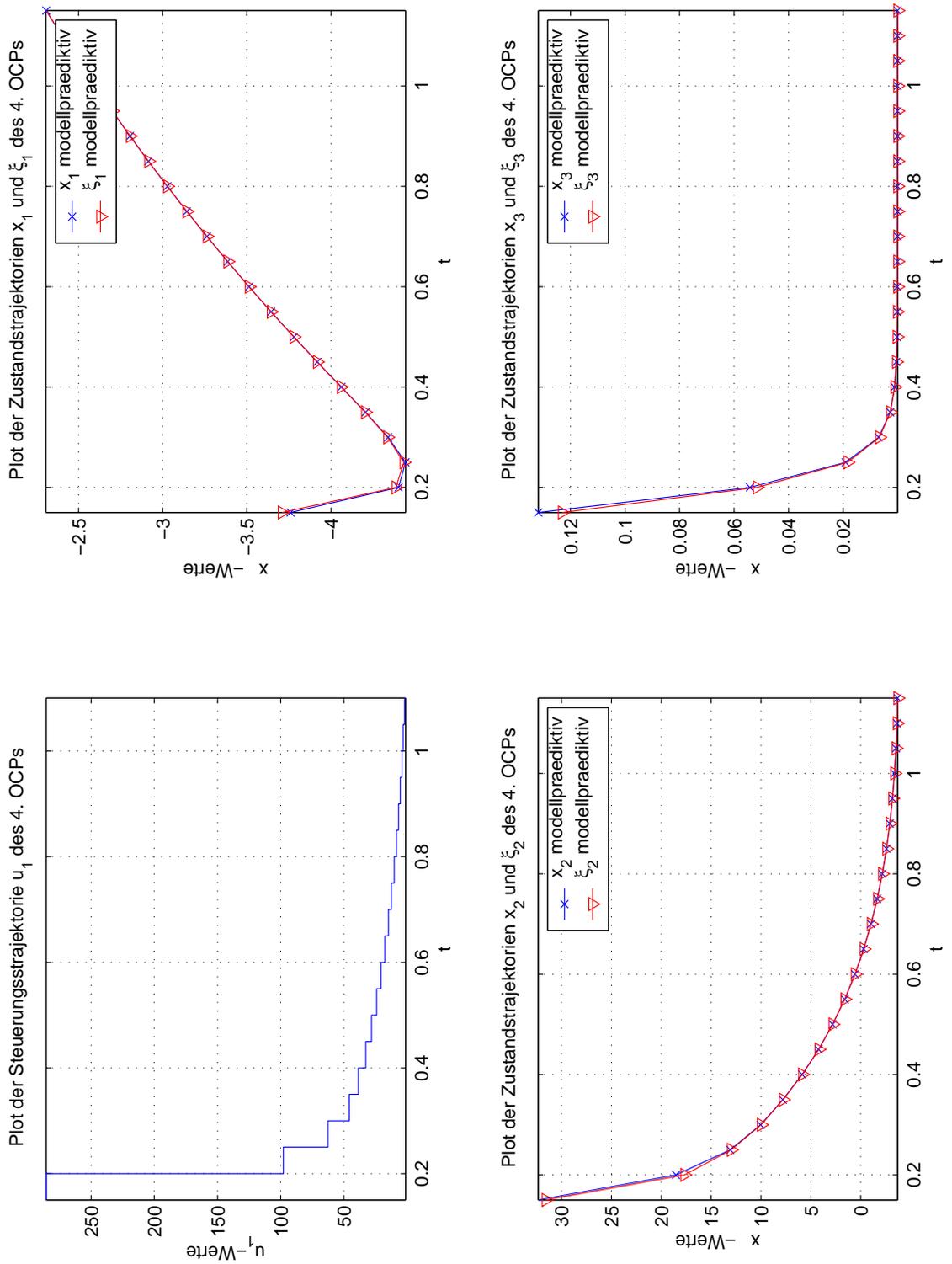


Abbildung C.5: Beispiels 6.3: 4. optimales Steuerungsproblem

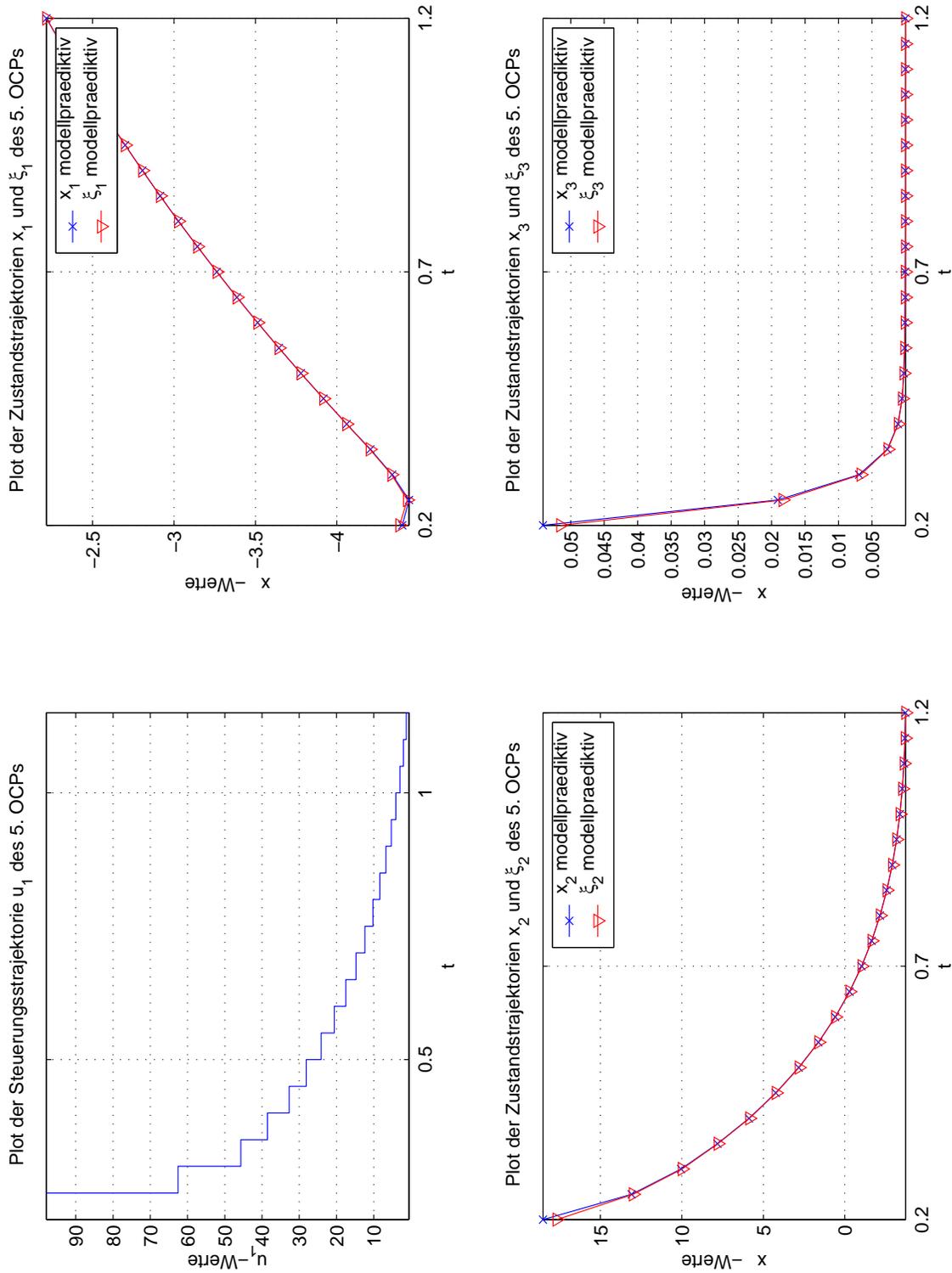


Abbildung C.6: Beispiels 6.3: 5. optimales Steuerungsproblem

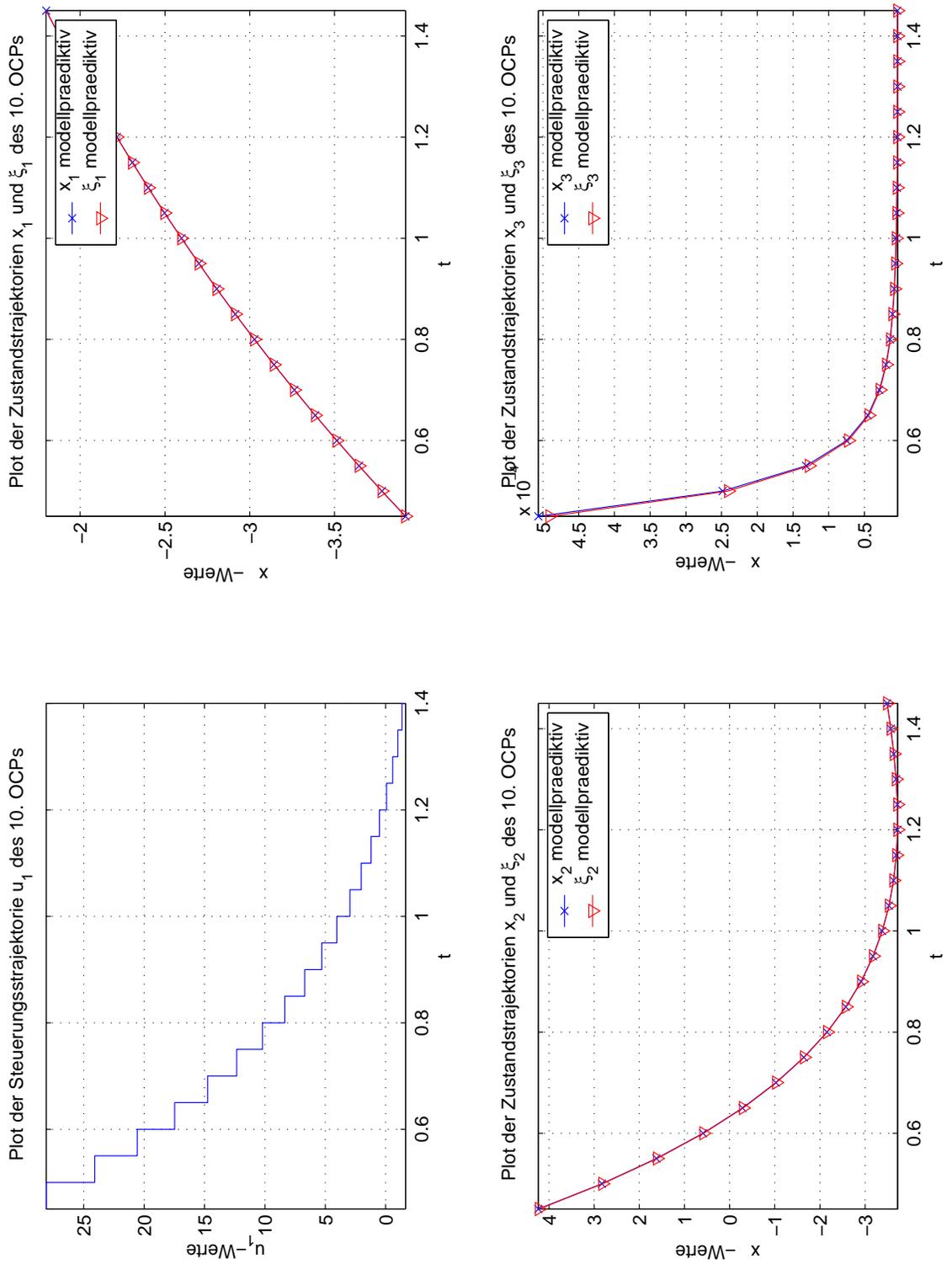


Abbildung C.7: Beispiels 6.3: 10. optimales Steuerungsproblem

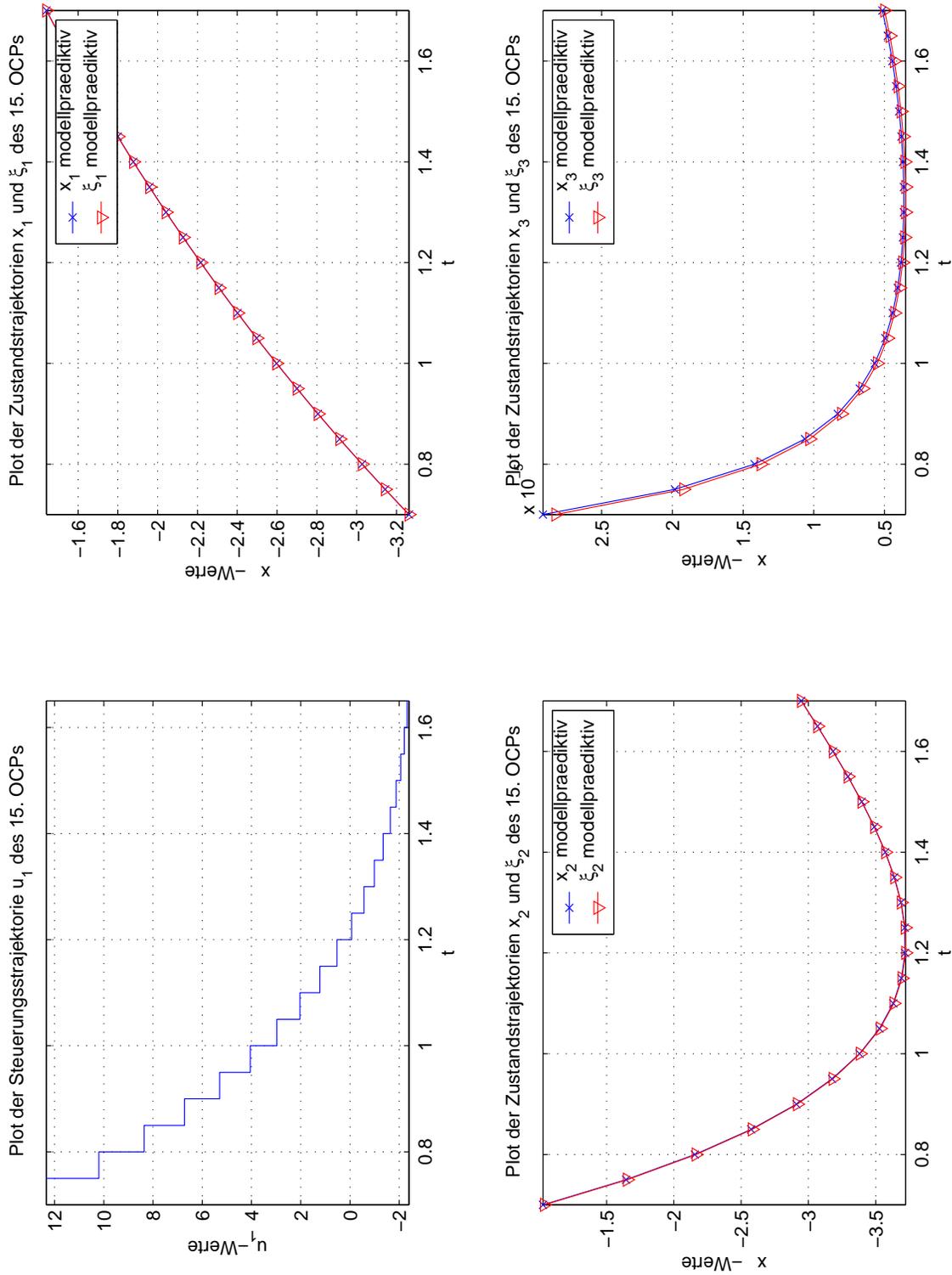


Abbildung C.8: Beispiels 6.3: 15. optimales Steuerungsproblem

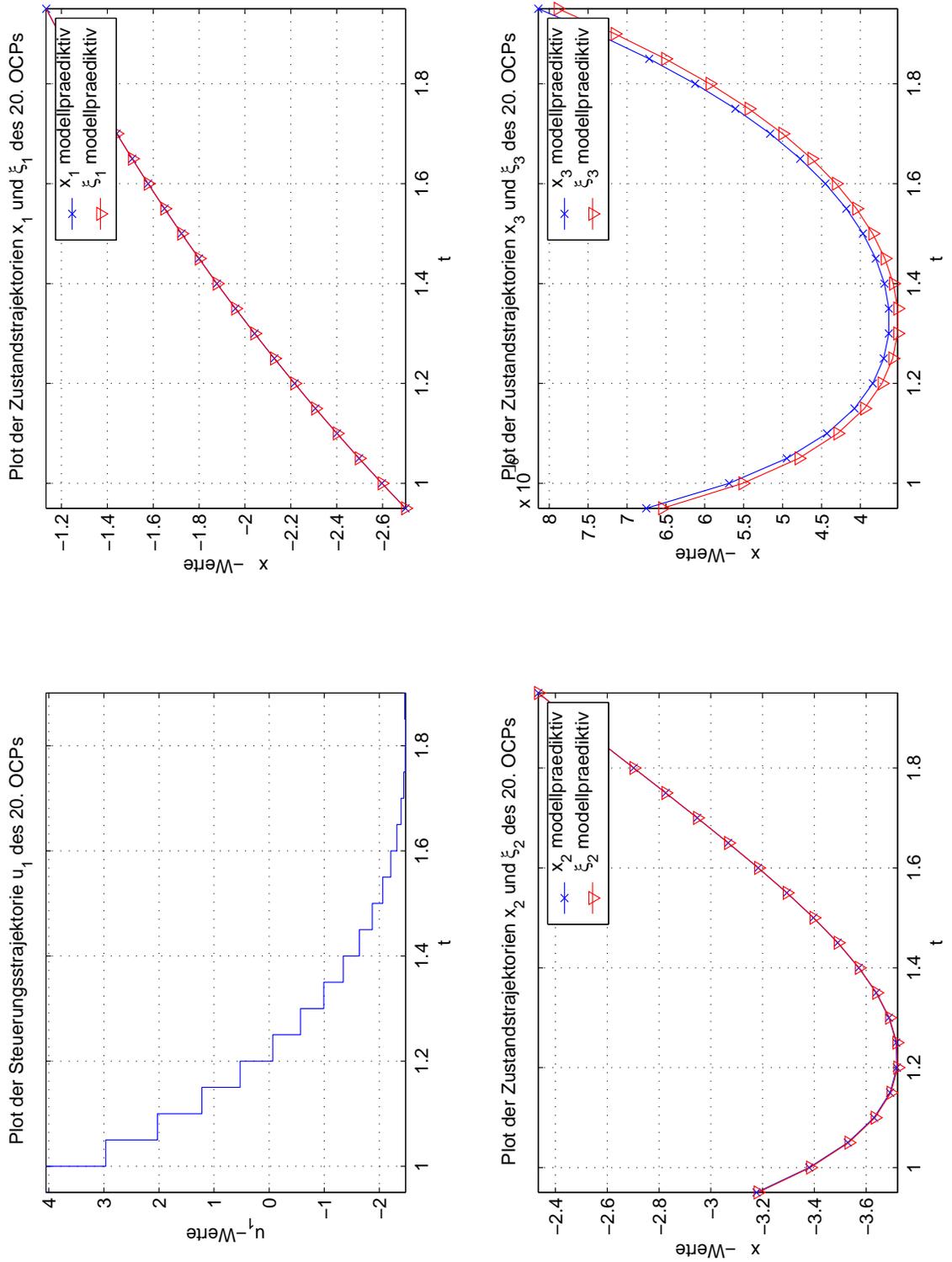


Abbildung C.9: Beispiels 6.3: 20. optimales Steuerungsproblem

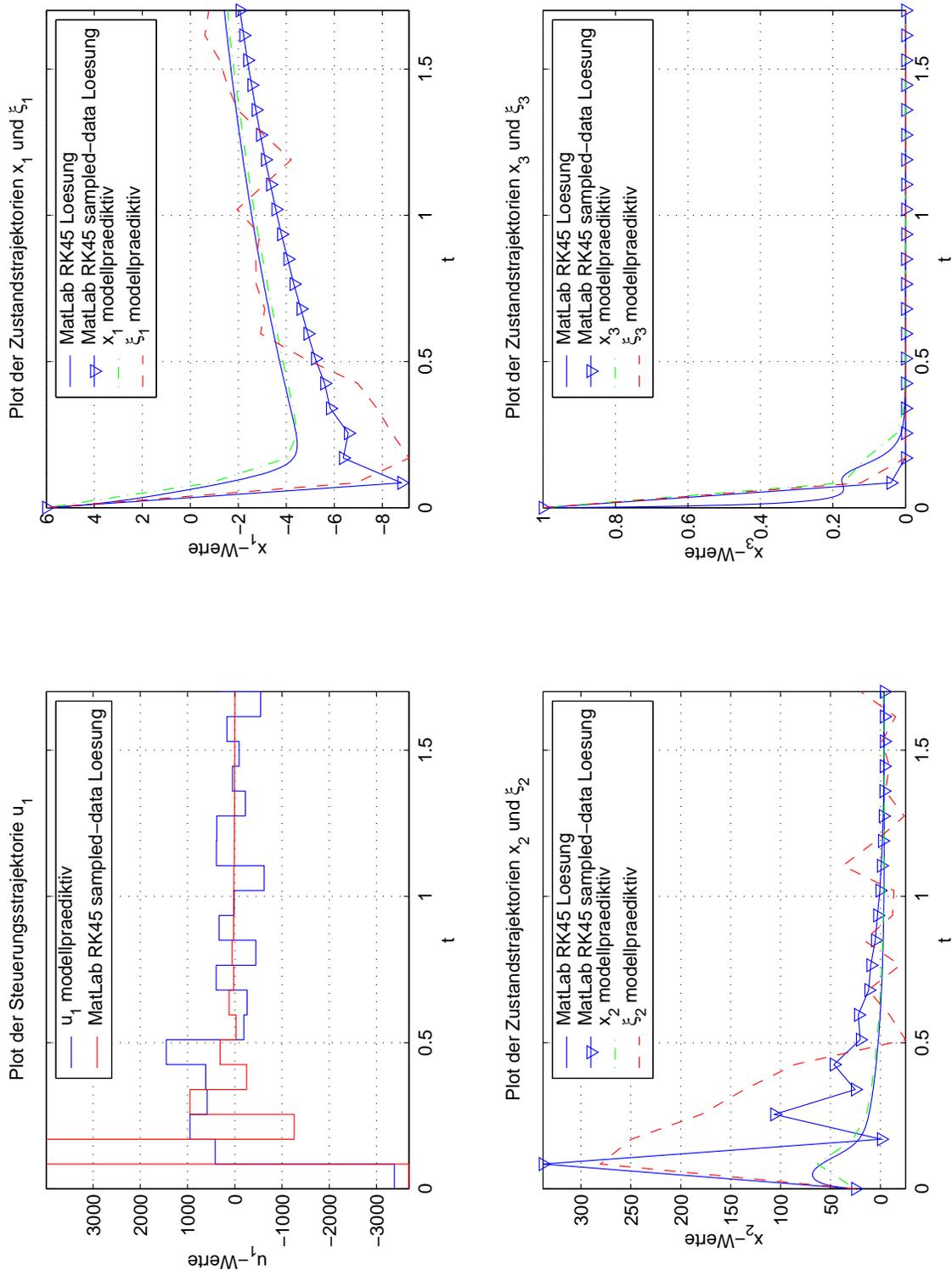


Abbildung C.10: Beispiels 6.3 für $T = 0.085$ und $H = 1.70$

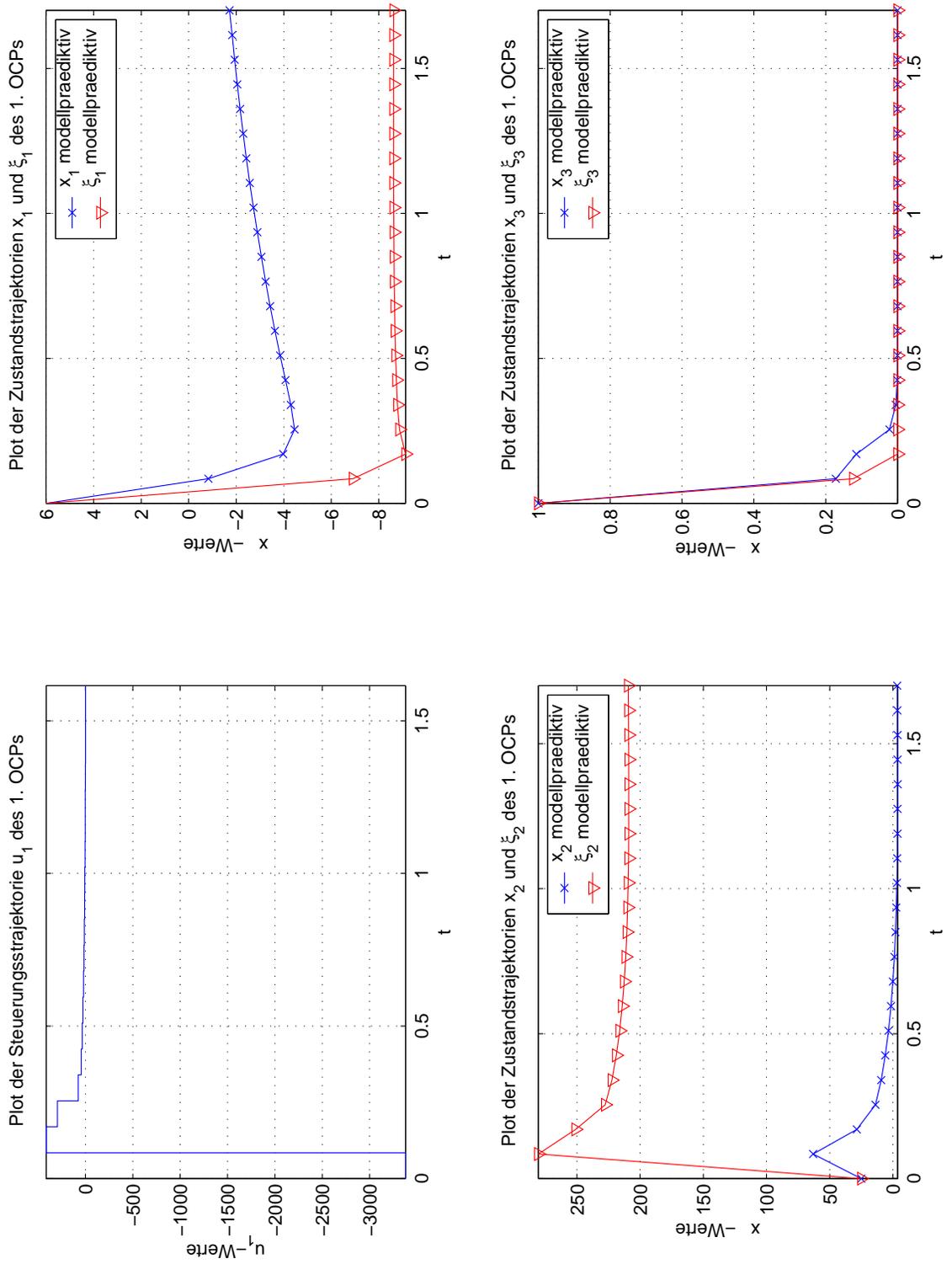


Abbildung C.11: Beispiels 6.3: 1. optimales Steuerungsproblem

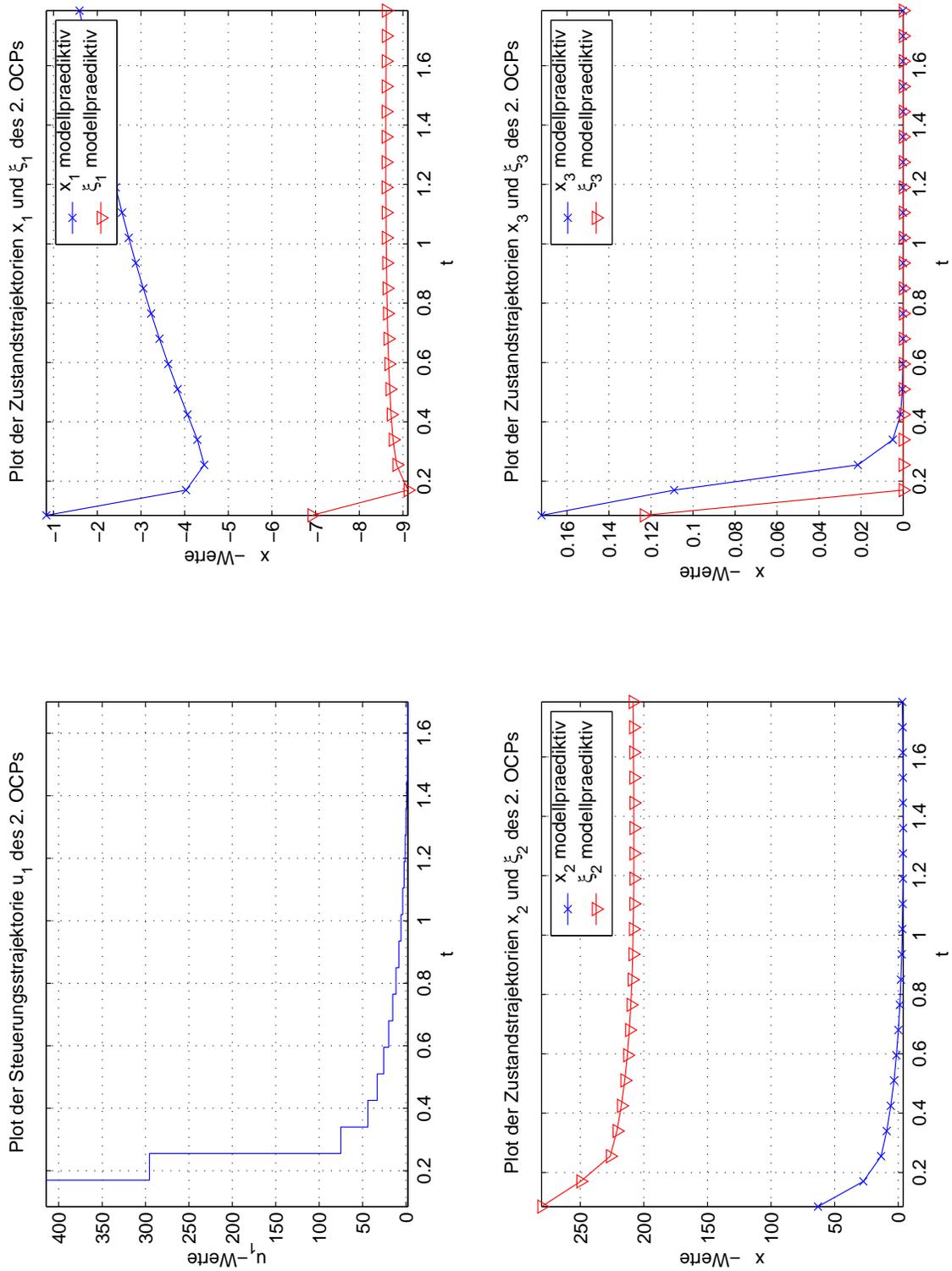


Abbildung C.12: Beispiels 6.3: 2. optimales Steuerungsproblem

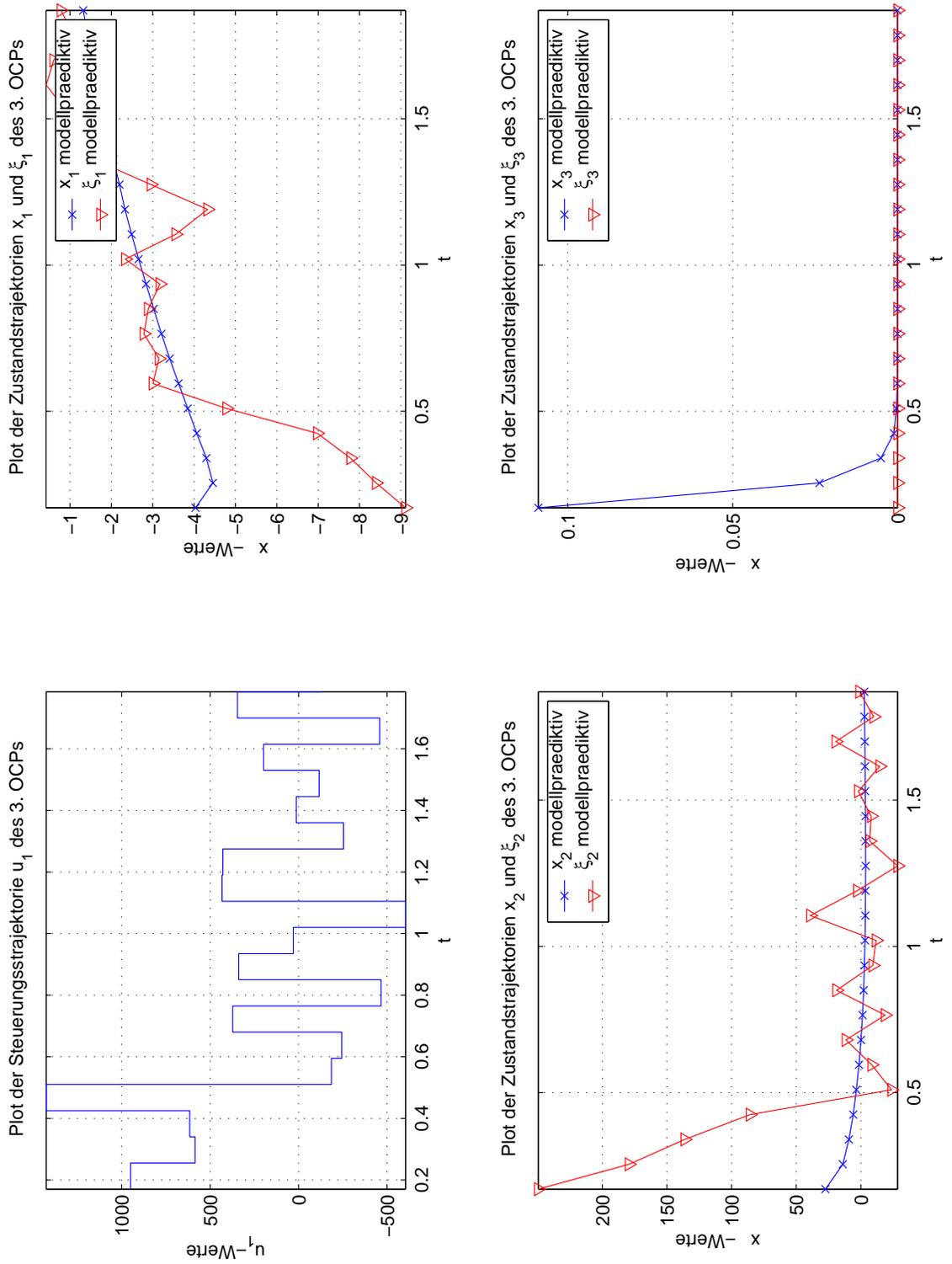


Abbildung C.13: Beispiels 6.3: 3. optimales Steuerungsproblem

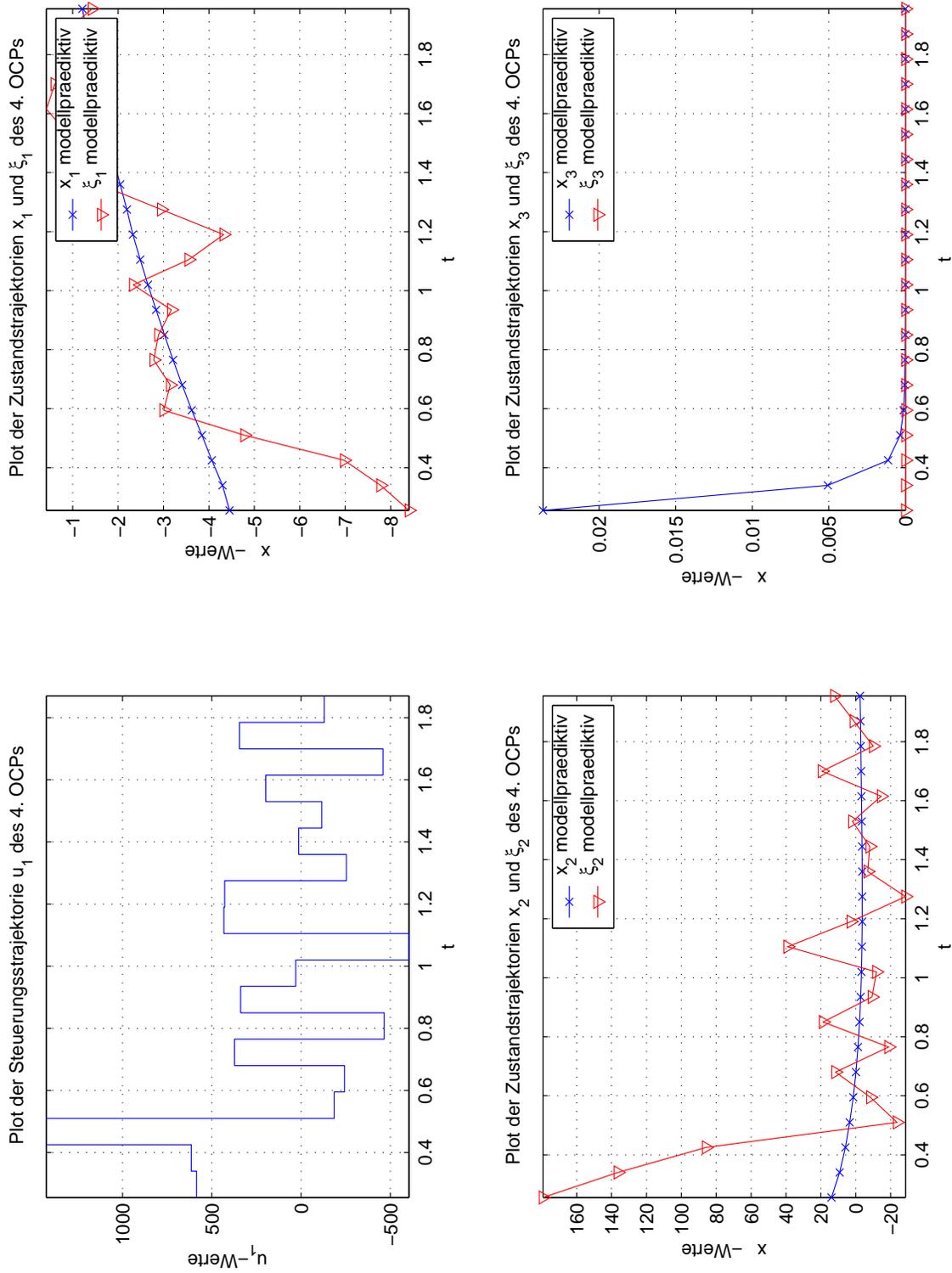


Abbildung C.14: Beispiels 6.3: 4. optimales Steuerungsproblem

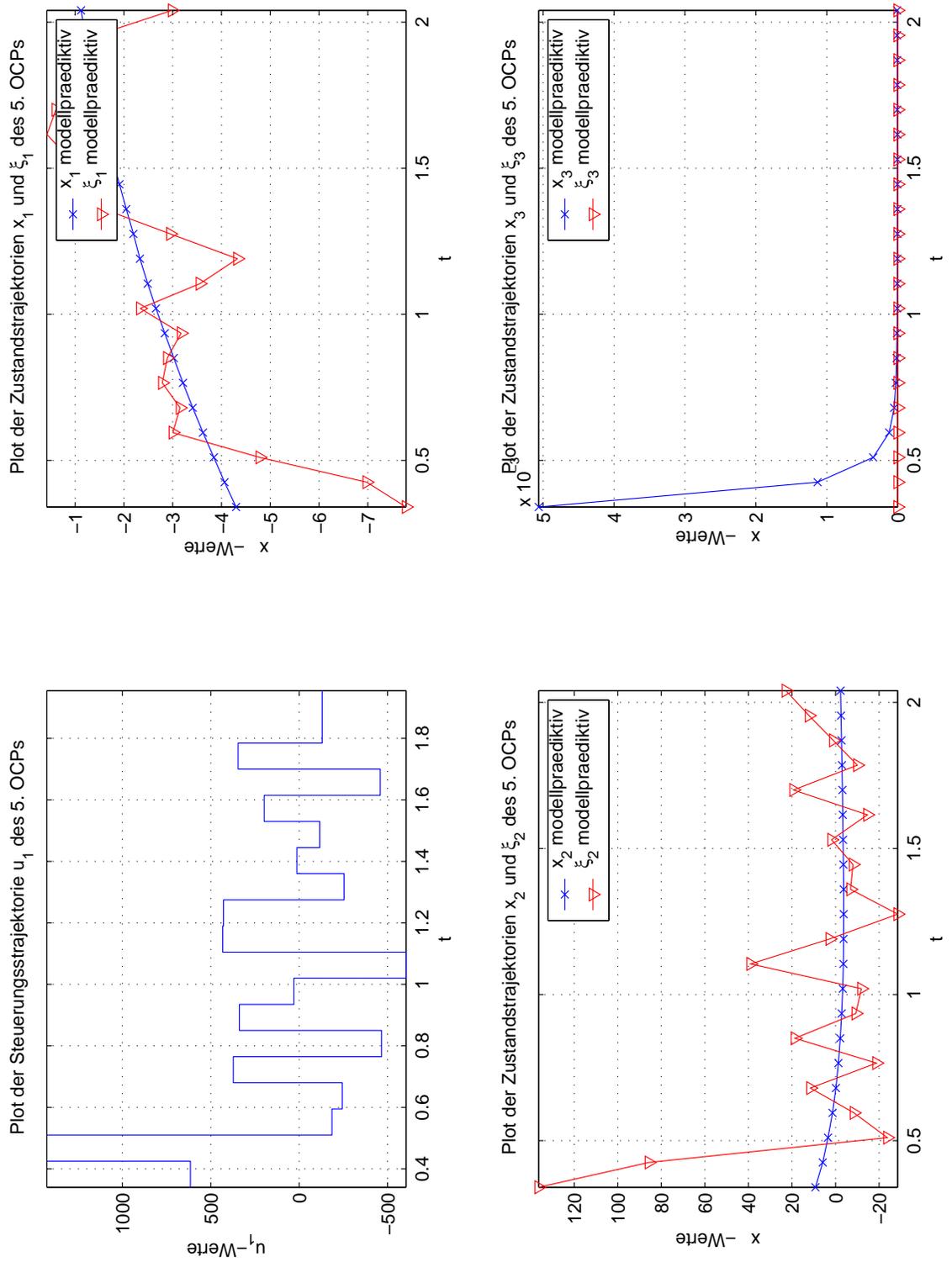


Abbildung C.15: Beispiels 6.3: 5. optimales Steuerungsproblem

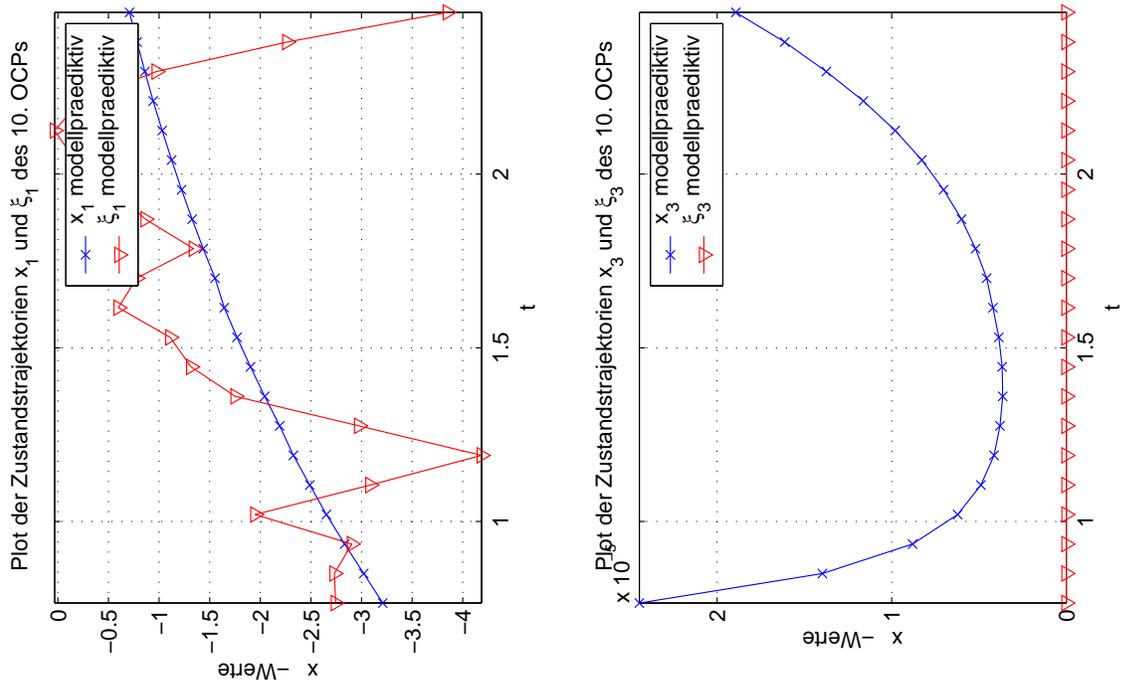


Abbildung C.16: Beispiels 6.3: 10. optimales Steuerungsproblem

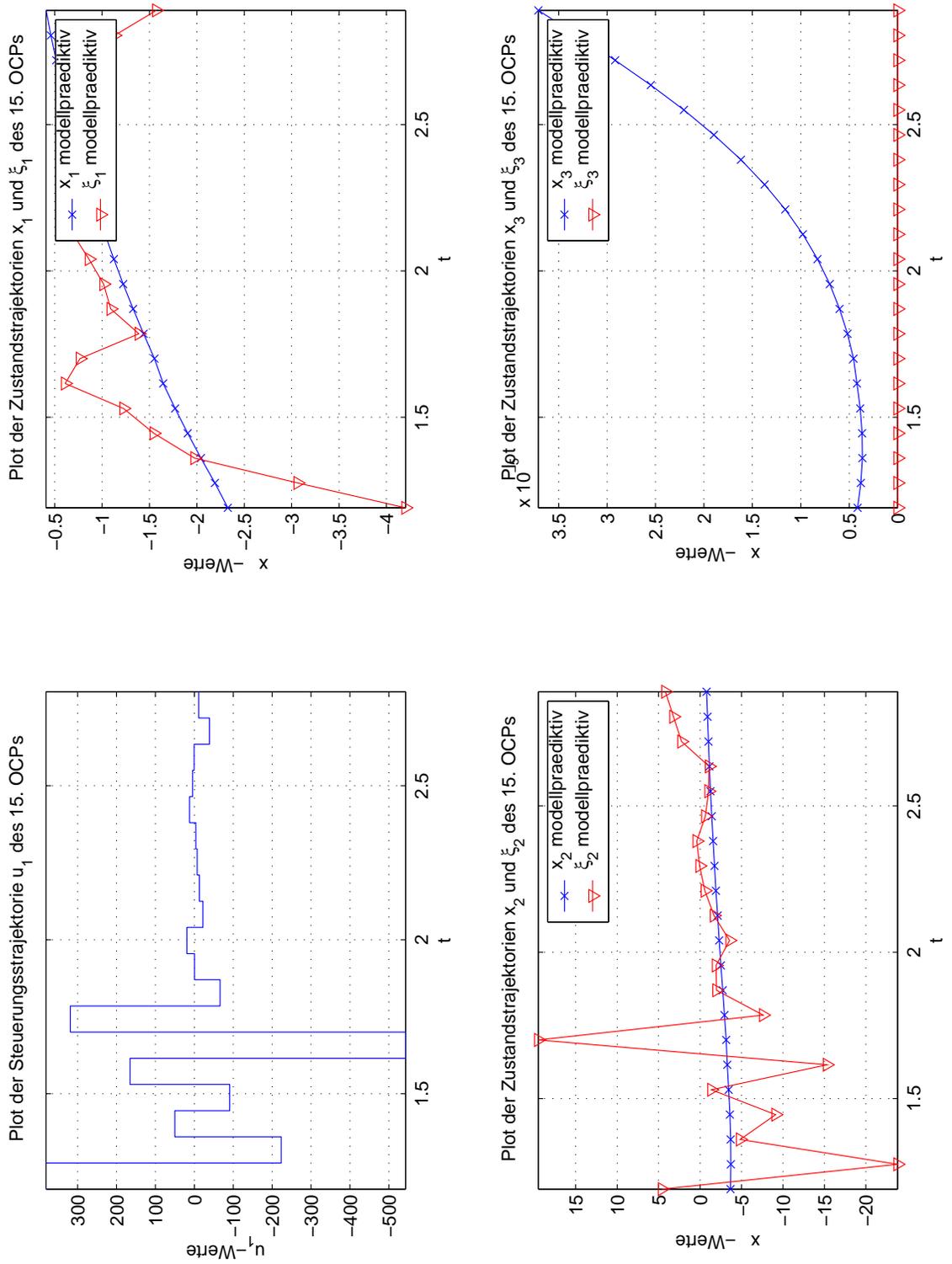


Abbildung C.17: Beispiels 6.3: 15. optimales Steuerungsproblem

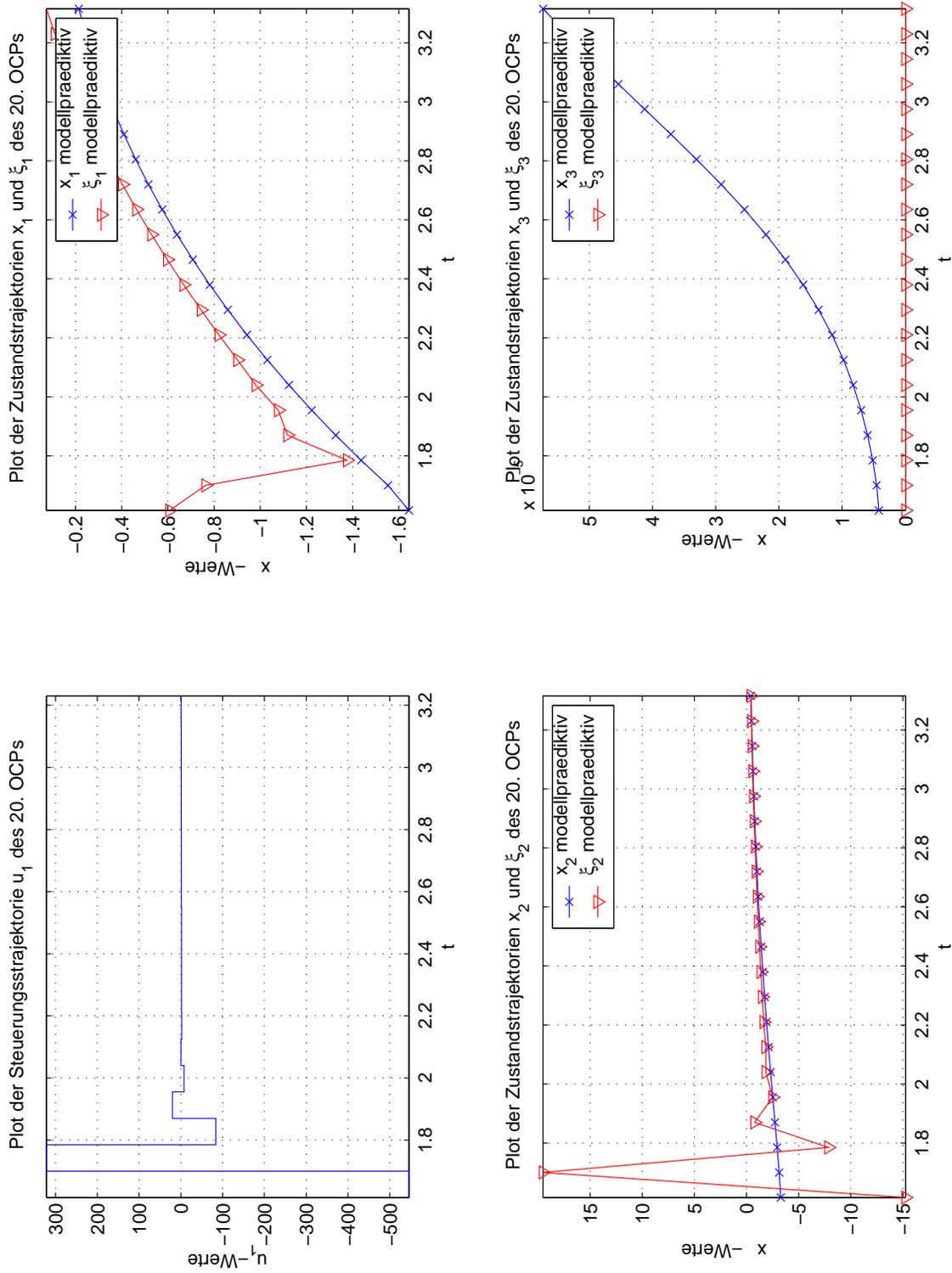


Abbildung C.18: Beispiels 6.3: 20. optimales Steuerungsproblem

Anhang D

Material auf der beiliegenden CD

Die angegebenen Programme zu den Beispielen 6.1-6.3 sowie Ergebnisse zu den in Kapitel 6 angestellten Untersuchungen sind auf der beigefügten CD enthalten. Außerdem enthält diese das Verzeichnis ARBEIT mit der vorliegenden Ausarbeitung in pdf-Format sowie aller verwendeter Abbildungen, jeweils in eps-Format.

Verzeichnis PROGRAMM:

Im Unterverzeichnis FORTRAN sind darin die teils modifizierten NUOCCCS-Dateien enthalten, die in kompilierter Form zudem im Unterverzeichnis OBJECTIVES vorliegen. Die in Fortran 77 geschriebenen Programme der einzelnen Beispiele sowie die Dateien zum Compilierungsaufwurf und zur graphischen Veranschaulichung, *make* und *makepic*, sind im Unterverzeichnis WORK enthalten.

Verzeichnis ERGEBNISSE:

Hierin sind die zu den Beispielen 6.1-6.3 erhaltenen Ergebnisse in Unterverzeichnissen enthalten. Die Namen der Unterverzeichnisse ergeben sich dabei aus den Überschriften in Kapitel 6.

Verzeichnis MATLAB:

Die zur Visualisierung und Fehlerberechnung angesprochenen Dateien sind in diesem Verzeichnis getrennt nach Beispielen in Unterverzeichnissen enthalten.

Dateiname	Funktion
makefile	Compilierung des Programms und Verlinkung mit allen notwendigen Bibliotheken und NUDOCCCS-Dateien
makepic	Generierung der NUDOCCCS-Ausgabe und Veranschaulichung der Ergebnisse
mpc1.m	Differentialgleichungssystem des Beispiels 6.1
mpc1dis.m	Differentialgleichungssystem des Beispiels 6.1 für direktes Sampling
mpc2.m	Differentialgleichungssystem des Beispiels 6.2
mpc2dis.m	Differentialgleichungssystem des Beispiels 6.2 für direktes Sampling
mpc3.m	Differentialgleichungssystem des Beispiels 6.3
mpc3dis.m	Differentialgleichungssystem des Beispiels 6.3 für direktes Sampling
plottingBeispiel1.m	Visualisierung des Beispiels 6.1
plottingBeispiel2.m	Visualisierung des Beispiels 6.2
plottingBeispiel3.m	Visualisierung des Beispiels 6.3
fehler.m	Berechnung der NUDOCCCS-internen Abweichung der Trajektorien und Vergleich mit Matlab-Lösung
Beispiel1.f	Programm zur Berechnung der modellprädiktiven Regelung für das Beispiel 6.1
Beispiel2.f	Programm zur Berechnung der modellprädiktiven Regelung für das Beispiel 6.2
Beispiel3.f	Programm zur Berechnung der modellprädiktiven Regelung für das Beispiel 6.3
adjungierte.f basic.f dc_decsol.f decsol.f gnudelivery.f gridcontr.f nudocccs_9.0.f	NUDOCCCS-Dateien (teils modifiziert)

Notation

In der folgenden Übersicht sind einige wichtige Bezeichnungen und Notationen vermerkt, die im Laufe der vorliegenden Arbeit regelmäßig erscheinen. Für die jeweilige Bedeutung der Ausdrücke sei auf die entsprechenden Kapiteln verwiesen.

$f(x(t), u(t))$	Dynamik des Kontrollsystems
$\psi(x(t_0), x(t_f))$	Randbedingungen
$x(t)$	Zustand (mit kontinuierlicher Feedbacksteuerung)
$u(t)$	Steuerung
$\xi(t)$	sampled-data Zustand
v	sampled-data Steuerung
$(\xi(t), x(t))$	zeitkontinuierliches Modell
u_M	modellprädiktive Regelung
$V(x)$	optimale Wertefunktion
T	sampled-data Rate
H	Horizontlänge
$J_0(x(t), u(t))$	Zielfunktional
$J_M(\xi(t), x(t), v)$	Zielfunktional des modellprädiktiven Regelungsansatzes
$L_0(x(t), u(t))$	Lagrange-Anteil des Zielfunktional J_0
$\Phi_0(x(t_0), x(t_f))$	Mayer-Anteil des Zielfunktional J_0
$l(\xi(t) - x(t), v)$	Lagrange-Anteil des Zielfunktional J_M
$F(\xi(t_M), x(t_M))$	Mayer-Anteil des Zielfunktional J_M
$(t_k)_{k \in \mathbb{N}}$	sampled-data Folge mit $t_k = k \cdot T$
$H(x(t), \lambda(t), u(t), t)$	Hamiltonfunktion
$\tilde{H}(x(t), \lambda(t), \mu(t), u(t), t)$	erweiterte Hamiltonfunktion
$\dot{\lambda}(t)$	adjungierte Differentialgleichung

$\mu(t)$	Multiplikatorfunktion der Beschränkungen
$c(x(t), u(t))$	Vektor der Beschränkungen
$\mathcal{D}_{\hat{n}}$	Gitter der Stufe \hat{n}
$L(z, \eta)$	Lagrange-Funktion
η	Lagrange-Multiplikator
(ξ^+, x^+)	exaktes zeitdiskretes Modell
$x_k^e(t)$	exakte zeitdiskrete Lösung
$x_k^a(t)$	approximierte zeitdiskrete Lösung
$\varepsilon_k(\xi, z)$	Fehler im k-ten Intervall
$\tilde{F}(x(t), u(t))$	Änderung in $F(x(t))$
$\Upsilon(\xi(t), x(t), u_{[0, \infty]})$	neudefiniertes Zielfunktional
$\Phi(\xi_i, x_i, u_i)$	Lagrange-Anteil des Zielfunktional Υ
$B_r(x^*)$	Kugel $\{x \in \mathbb{R}^n \mid \ x - x^*\ \leq r\}$
$\text{int}(B)$	Innere der Menge B
∂B	Rand der Menge B
$\text{cl } B$	Abschluss der Menge B
\mathbb{N}	Raum der natürlichen Zahlen
\mathbb{R}	Raum der reellen Zahlen
\mathbb{R}^n	Raum der n-dimensionalen reellen Zahlen
U	Kontrollwertebereich
\mathcal{U}	Raum der Kontrollfunktionen
\mathbb{Z}	Raum der ganzen Zahlen
$C([a, b])$	Raum der auf dem Intervall $[a, b]$ stetigen Funktionen
$C^k([a, b])$	Raum der auf dem Intervall $[a, b]$ k-mal stetig differenzierbaren Funktionen
$L_\infty(t_0, t_f; \mathbb{R}^n)$	Banach-Raum
$W^{1, \infty}(t_0, t_f; \mathbb{R}^n)$	Sobolev-Raum
$ a $	Absolutwert der Zahl a
$\ x\ _1$	Betragssummennorm des Vektors x
$\ x\ $	euklidische Norm des Vektors x
$\ x\ _\infty$	Maximumsnorm des Vektors x
$\ A\ $	induzierte Norm der Matrix A

Literaturverzeichnis

- [1] M.Bardi, I.Capuzzo Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*; Birkhäuser, Boston, 1997.
- [2] Ch.Büskens, *Diskretisierungsverfahren zur numerischen Lösung von Optimalsteuerungsproblemen*; Ausarbeitung einer Vorlesung gehalten im Sommersemester 2002 an der Universität Bayreuth.
- [3] P.Deuffhard, F.Bornemann: *Numerische Mathematik II*; 2. Auflage, de Gruyter, 2002.
- [4] L.C.Evans, *Partial Differential Equations*; American Mathematical Society, 2002.
- [5] R.Fletcher, *Practical Methods of Optimization*; 2nd. Edition, Wiley, 1987.
- [6] P.E.Gill, W.Murray, M.H.Wright, *Practical Optimization*; Academic Press, 1981.
- [7] G.Grimm, M.J.Messina, A.R.Teel, S.Tuna: Model predictive control: for want of a local control Lyapunov function, all is not lost, soll erscheinen in *IEEE Trans. Automat. Contr.*, 2004.
- [8] L.Grüne, *Stabilität und Stabilisierung nichtlinearer Systeme*; Ausarbeitung einer Vorlesung gehalten im Sommersemester 2002 an der J.W. Goethe Universität Frankfurt.
- [9] L.Grüne, *Numerische Dynamik von Kontrollsystemen*; Ausarbeitung einer Vorlesung gehalten im Sommersemester 2004 an der Universität Bayreuth.
- [10] A.Isidori, *Nonlinear control systems*; 3. Auflage, Springer Verlag: London, 2002.
- [11] A.Jadbabaie, J.Yu, J.Hauser: Unconstrained receding horizon control of nonlinear systems, *IEEE Trans. Automat. Contr.*; vol. 46, pp. 776-783, 2001.
- [12] C.M.Kellet, *Advances in Converse and Control Lyapuno Functions*; PhD thesis, University of California, Santa Barbara; 2002.
- [13] H.K.Khalil, *Nonlinear systems*; 5. Auflage, Prentice Hall, 2001.
- [14] M.Krstić, I.Kanellakopoulos, P.Kokotović, *Nonlinear and Adaptive Control Design*; 1. Auflage, John Wiley & Sons Inc, 1995.

- [15] L.Magni, R.Sepulchre: Stability margins of nonlinear receding-horizon control via inverse optimality, *Syst. Contr. Lett.*; vol. 32, pp. 241-245
- [16] K.Malanowski, Ch.Büskens, H. Maurer: Convergence of approximations to nonlinear optimal control problems, *Mathematical Programming with Data Perturbations*; vol. 195, pp. 253-284, 1998.
- [17] H.Maurer, *On the Minimum Principle for Optimal Control Problems with State Constraints*; Schriftenreihe des Rechenzentrums der Universität Münster, Nr. 41; 1979.
- [18] H.Maurer, S.Pickenhain, Second order sufficient conditions for optimal control problems with mixed control-state constraints, *Journal of Optimization Theory and Applications*; vol. 86, pp. 649-667, 1996.
- [19] D.Q.Mayne, J.B.Rawlings, C.V.Rao, P.O.M.Scokaert: Constrained model predictive control: stability and optimality, *Automatica*; vol. 36, pp. 789-814, 2000.
- [20] D.Nešić, A.R.Teel, P.V.Kokotović: Sufficient conditions for stabilizations of sampled-data systems via discrete-time approximations, *IEEE Trans. Automat. Contr.*; vol. 38, pp. 259-270, 1999.
- [21] D.Nešić, A.R.Teel, P.V.Kokotović: Formulas relating KL stability estimates of discrete-time and sampled-data nonlinear systems, *Syst. Contr. Lett.*; vol. 38, pp. 49-60, 1999.
- [22] D.Nešić, L.Grüne: Lyapunov based continuous-time nonlinear controller redesign for sampled-data implementation, geplante Erscheinung 2004.
- [23] D.Nešić, L.Grüne: A receding horizon control approach to sampled-data implementation of continuous-time controllers, geplante Erscheinung 2004.
- [24] L.W.Neustadt, *Optimization: A Theory of Necessary Conditions*; Princeton University Press, 1976.
- [25] L.S.Pontryagin, V.G.Boltjanskij, R.V.Gamkrelidze, E.F.Miscdenko, *Mathematische Theorie optimaler Prozesse*; Oldenbourg, 1967.
- [26] K.Schittkowski: On the convergence of a sequential quadratic programming method with an augmented Lagrangian search direction; *Mathematische Operationsforschung und Statistik, Series: Optimization*, vol. 14, pp. 197-216; 1983.
- [27] E.D.Sontag, Smooth stabilization implies coprime factorization, *IEEE Trans. Automat. Contr.*; vol. 34, pp. 435-443, 1989.
- [28] W.Walter, *Gewöhnliche Differentialgleichungen*; 6. (überarbeitete und erweiterte) Auflage, Springer; 1996.

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. Februar 2005

.....
Jürgen Pannek