

**UNIVERSITÄT
BAYREUTH**

Modellprädiktive Regelung (MPC) mittels linearer Programmierung

Diplomarbeit

von

Susanne Bauer

FAKULTÄT FÜR MATHEMATIK UND PHYSIK
MATHEMATISCHES INSTITUT

Datum: 8. Mai 2008

Aufgabenstellung und Betreuung:
Prof. Dr. Lars Grüne

Danksagung

An dieser Stelle möchte ich mich bei Herrn Professor Dr. Grüne für die ausgezeichnete Betreuung während der Erstellung dieser Arbeit und besonders für das Heranführen an dieses interessante Themengebiet durch Vorlesungen und Seminare bedanken.

Mein besonderer Dank gilt auch meiner Familie und meinen Freunden, die mich während des gesamten Studiums unterstützt haben.

Inhaltsverzeichnis

Abbildungsverzeichnis	III
Tabellenverzeichnis	V
1 Einleitung	1
2 Einführung in die Kontrolltheorie	3
2.1 Lineare Kontrollsysteme	4
2.2 Existenz und Eindeutigkeit	6
2.3 Kontrollierbarkeit	8
2.4 Stabilität und Stabilisierung	10
2.4.1 Lyapunov-Funktionen	12
2.4.2 Das Stabilisierungsproblem für lineare Kontrollsysteme	14
3 Modellprädiktive Regelung (MPC)	21
3.1 Problemstellung der modellprädiktiven Regelung	21
3.2 Lineare MPC-Modelle	23
3.2.1 Zustandsmodelle (state space models)	23
3.2.2 Übertragungsfunktionen (transfer function models)	24
3.2.3 FIR-Modelle (impulse response-, step response models)	25
3.3 Die Zielfunktion	27
3.3.1 Das quadratische Kriterium	28
3.3.2 1-Norm und ∞ - ∞ Norm	30
3.4 Das „receding-horizon“ Konzept	31
3.5 Der modellprädiktive Regelungsansatz	33
4 Lineare Programmierung	35
4.1 Lineare Programme	35
4.1.1 Definition	35
4.1.2 Die Standardform	36
4.1.3 Geometrische Grundlagen	39
4.2 Das Simplexverfahren	43

4.2.1	Das spezielle Simplexformat	43
4.2.2	Durchführung der Simplex-Methode	46
4.2.3	Abbruchkriterien	48
4.2.4	Zusammenfassung: Ein Simplexschritt	50
4.2.5	Der Befehl LINPROG in Matlab	51
5	Modellprädiktive Regelung mittels linearer Programmierung	55
5.1	Lineare Kontrollsysteme ohne Ausgang	55
5.1.1	Das Optimierungsproblem	56
5.1.2	Die 1-Norm in der Zielfunktion	58
5.1.3	Umformung in ein äquivalentes Optimierungsproblem	59
5.1.4	Das lineare Programm in Matrizenform	62
5.2	Anwendung des MPC-Konzeptes	77
5.3	Kontrollsysteme mit Ausgang	81
5.3.1	Herleitung des äquivalenten Optimierungsproblems	81
5.3.2	Das lineare Programm in Matrizenform	83
5.4	Berücksichtigung von Beschränkungen	90
5.4.1	Erweiterung des linearen Programms	92
6	Beispiele und numerische Ergebnisse	101
6.1	Beispiel 1: Das inverse Pendel	102
6.1.1	Systembeschreibung und Modellbildung	102
6.1.2	Ergebnisse der modellprädiktiven Regelung mittels linearer Programmierung	106
6.2	Beispiel 2: Der 2-DOF Hubschrauber von Quanser	121
6.2.1	Das Hubschrauber-Modell	121
6.2.2	Ergebnisse der modellprädiktiven Regelung mittels linearer Programmierung	123
6.3	Wahl der Gewichtungsmatrizen	134
A	Inhalt der CD	137
	Literaturverzeichnis	139

Abbildungsverzeichnis

2.1	Vergleich der Lösungen – kontinuierlich (blau) und zeitdiskret (grün)	6
3.1	Zielsetzung der MPC-Strategie	22
3.2	Impulse Response	26
3.3	Step Response	27
3.4	Auswirkung der Wahl der Gewichtungsvariablen	29
3.5	y_{ref} im Vergleich zur Approximation \tilde{y}_{ref}	30
4.1	Darstellung eines Polyeders, eine lineare Gleichung	40
4.2	Darstellung eines Polyeders, zwei lineare Gleichungen	41
5.1	Verhalten des Zustandes $x(t)$ bei Anwendung der Kontrollfolge $u(t)$	57
5.2	Kontrollfolge $u(t)$	58
5.3	Schritt 1. und 2. des modellprädiktiven Regelungsansatzes	78
5.4	Schritt 3. bis 5. und Wiederholung von Schritt 2.	79
5.5	Weitere Wiederholung des „receding-horizon“ Konzeptes	79
5.6	Optimale Kontrollfolge $u(t)$	80
5.7	Optimales Verhalten des Zustandes $x(t)$	80
5.8	Optimale Kontrollfolge mit Beschränkungen	91
5.9	Optimales Verhalten des Ausgangs bei Beschränkungen	91
6.1	Das inverse Pendel	102
6.2	Der Segway Personal Transporter	103
6.3	Verhalten des Zustandes für $N_u = 5$ und $N_x = 15$	107
6.4	Optimale Kontrollfolge für $N_u = 5$ und $N_x = 15$	107
6.5	Verhalten des Zustandes für $N_u = 3$ und $N_x = 15$	108
6.6	Optimale Kontrollfolge für $N_u = 3$ und $N_x = 15$	108
6.7	Verhalten des Zustandes für $N_u = 5$ und $N_x = 10$	109
6.8	Optimale Kontrollfolge für $N_u = 5$ und $N_x = 10$	109
6.9	Verhalten des Zustandes für $N_u = 10$ und $N_x = 10$	110
6.10	Optimale Kontrollfolge für $N_u = 10$ und $N_x = 10$	110
6.11	Verhalten des Zustandes für $N_u = 3$ und $N_x = 10$	111

6.12	Optimale Kontrollfolge für $N_u = 3$ und $N_x = 10$	111
6.13	Verhalten des Zustandes für $N_u = 3$ und $N_x = 5$	112
6.14	Optimale Kontrollfolge für $N_u = 3$ und $N_x = 5$	112
6.15	Verhalten des Zustandes bei $\bar{Q} = 0.5$ und $\bar{R} = 2$	115
6.16	Optimale Kontrollfolge bei $\bar{Q} = 0.5$ und $\bar{R} = 2$	115
6.17	Verhalten des Zustandes bei $\bar{Q} = 2$ und $\bar{R} = 2$	116
6.18	Optimale Kontrollfolge bei $\bar{Q} = 2$ und $\bar{R} = 2$	116
6.19	Verhalten des Zustandes bei $\bar{Q} = 5$ und $\bar{R} = 1$	117
6.20	Optimale Kontrollfolge bei $\bar{Q} = 5$ und $\bar{R} = 1$	117
6.21	Verhalten des Zustandes bei $\bar{Q} = 1$ und $\bar{R} = 5$	118
6.22	Optimale Kontrollfolge bei $\bar{Q} = 1$ und $\bar{R} = 5$	118
6.23	Verhalten für Anfangszustand $x_1 = 5, x_2 = 1, x_3 = 2, x_4 = 1.5$	119
6.24	Kontrolle für Anfangszustand $x_1 = 5, x_2 = 1, x_3 = 2, x_4 = 1.5$	119
6.25	Das Hubschrauber-Modell	121
6.26	Verhalten des Ausganges bei $N_u = 5$ und $N_y = 10$	124
6.27	Verhalten des Zustandes bei $N_u = 5$ und $N_y = 10$	125
6.28	Optimale Kontrolle u_1 bei $N_u = 5$ und $N_y = 10$	125
6.29	Optimale Kontrolle u_2 bei $N_u = 5$ und $N_y = 10$	126
6.30	Verhalten des Ausganges bei $N_u = 3$ und $N_y = 5$	126
6.31	Verhalten des Zustandes bei $N_u = 3$ und $N_y = 5$	127
6.32	Optimale Kontrolle u_1 bei $N_u = 3$ und $N_y = 5$	127
6.33	Optimale Kontrolle u_2 bei $N_u = 3$ und $N_y = 5$	128
6.34	Verhalten des Ausganges bei $\bar{Q} = 2$ und $\bar{R} = 0.5$	129
6.35	Verhalten des Zustandes bei $\bar{Q} = 2$ und $\bar{R} = 0.5$	130
6.36	Optimale Kontrolle u_1 bei $\bar{Q} = 2$ und $\bar{R} = 0.5$	130
6.37	Optimale Kontrolle u_2 bei $\bar{Q} = 2$ und $\bar{R} = 0.5$	131
6.38	Verhalten des Ausganges bei $\bar{Q} = 0.5$ und $\bar{R} = 2$	131
6.39	Verhalten des Zustandes bei $\bar{Q} = 0.5$ und $\bar{R} = 2$	132
6.40	Optimale Kontrolle u_1 bei $\bar{Q} = 0.5$ und $\bar{R} = 2$	132
6.41	Optimale Kontrolle u_2 bei $\bar{Q} = 0.5$ und $\bar{R} = 2$	133
6.42	Graph der Zielfunktion Θ	135

Tabellenverzeichnis

5.1	Einträge, welche mit $u_j(t)$ multipliziert werden	67
5.2	Einträge, welche mit $\gamma_{i,t}$ bzw. $\delta_{j,t}$ multipliziert werden	68
5.3	Einträge, über welche nicht minimiert wird	69
6.1	Eingabeparameter inverses Pendel 1	113
6.2	Daten inverses Pendel 1	113
6.3	Eingabeparameter inverses Pendel 2	120
6.4	Daten inverses Pendel 2	120
6.5	Eingabeparameter 2-DOF Hubschrauber 1	128
6.6	Daten 2-DOF Hubschrauber 1	129
6.7	Eingabeparameter 2-DOF Hubschrauber 2	133
6.8	Daten 2-DOF Hubschrauber 2	134
A.1	Übersicht der Matlab-Programme 1	138
A.2	Übersicht der Matlab-Programme 2	138

Kapitel 1

Einleitung

Die Theorie im Bereich der modellprädiktiven Regelung (englisch: model predictive control) ist heutzutage soweit ausgereift, dass diese Methode in der Industrie weit verbreitet eingesetzt wird. Dieser Regelungsansatz basiert auf einem Optimierungsproblem und verwendet zudem ein Modell des Prozesses, um das Verhalten des Systems auf einem endlichen Horizont vorherzusagen. Dazu wird zum aktuellen Zeitpunkt ein optimales Steuerungsproblem gelöst und das erste Glied der berechneten optimalen Kontrollfolge auf das System angewendet. Anschließend wird ein neues Optimierungsproblem formuliert, welches basierend auf aktualisierten Messungen des Zustandes auf dem nun um einen Zeitschritt verschobenen Horizont gelöst wird. Die der Optimierung zugrunde liegende Zielfunktion wird normalerweise durch ein quadratisches oder lineares Kriterium ausgedrückt, so dass das resultierende Optimierungsproblem als quadratisches (QP) bzw. lineares Programm (LP) behandelt werden kann.

Die Durchführbarkeit des Konzeptes der modellprädiktiven Regelung (MPC) ist jedoch zum Teil dadurch eingeschränkt, dass Optimierungsprobleme nicht immer in Echtzeit gelöst werden können. Deshalb eignet sich MPC weniger als Kontrollstrategie für große Systeme. Auf der anderen Seite ist diese Methode sehr beliebt, da Eingangs- und Zustandsbeschränkungen explizit im optimalen Steuerungsproblem berücksichtigt werden können. Für solche Probleme ist das in jedem Schritt zu lösende Optimierungsproblem meist ein konvexes Programm. Obwohl effiziente Algorithmen zur Lösung konvexer Programme existieren, können deutliche Verbesserungen erzielt werden, wenn man die Struktur des in MPC auftretenden Optimierungsproblems ausnutzt. Häufig wird dieses Problem durch eine quadratische Zielfunktion ausgedrückt, dessen Beliebtheit vor allem auf die einfache mathematische Behandlung zurückzuführen ist. Im unbeschränkten Fall kann das resultierende linear-quadratische Kontrollproblem dann effizient mit Methoden der dynamischen Programmierung gelöst werden. Seit einiger Zeit befürworten Wissenschaftler eine 1-Norm im Zielfunktional zu verwenden. Ein Grund dafür ist, dass das daraus resultierende Optimierungsproblem in ein lineares Programm umgeformt werden kann. Die Lösung eines linearen Programms erfordert rechnerisch weniger Aufwand als die entsprechende Lösung eines quadratischen Pro-

gramms gleicher Größe und Komplexität, weshalb das Optimierungsproblem bevorzugt als lineares Programm formuliert werden sollte. Das Konzept der modellprädiktiven Regelung mittels linearer Programmierung ist nicht neu, wurde aber bis heute nur von einigen Autoren untersucht. Während die Verwendung eines linearen Programms vom numerischen Standpunkt aus vorteilhaft erscheint, erhält man durch Anwendung der berechneten Kontrollfolge nicht notwendigerweise ein gutes closed-loop Verhalten. Der größte theoretische Einwand gegen eine Formulierung als lineares Programm ist, dass analytische Lösungen aufgrund der Struktur der Zielfunktion nicht immer verfügbar sind. Jedoch kann durch die Wahl der „tuning-Parameter“ auch ein deutlich besseres Verhalten erzielt werden als bei der Lösung des MPC-Problems mittels quadratischer Programmierung.

In dieser Arbeit wird nun ein Verfahren zur modellprädiktiven Regelung mittels linearer Programmierung für zeitdiskrete Kontrollsysteme entwickelt. Zweck dieses Verfahrens ist, eine optimale Kontrollfolge zu berechnen, die zu einem gewünschten Verhalten des Systems führt. In Kapitel 2 werden zunächst grundlegenden Definitionen der linearen Kontrolltheorie sowie wichtige Begriffe, die zur Analyse von Kontrollsystemen benötigt werden, eingeführt. Außerdem wird auf das Stabilisierungsproblem solcher Systeme eingegangen und ein möglicher Lösungsansatz beschrieben. Diesem Kapitel liegen die Arbeiten von L. Grüne [12] und F. Colonius [7] sowie die Bücher von H.-W. Knobloch, H. Kwakernaak [17] und E. D. Sontag [26] zu Grunde. Anschließend wird in Kapitel 3 das Verfahren der modellprädiktiven Regelung behandelt. Dabei soll besonders auf die unterschiedlichen Gestaltungsmöglichkeiten eingegangen werden. Die Darstellung orientiert sich dabei an [5] und [21]. Kapitel 4 widmet sich der linearen Programmierung. Es wird zunächst das lineare Programm in Standardform definiert und anschließend ein Lösungsverfahren – das sogenannte Simplexverfahren – vorgestellt. Dieses Kapitel orientiert sich dabei an dem Buch von F. Jarre und J. Stoer [16]. In Kapitel 5 wird schließlich das der modellprädiktiven Regelung zugrunde liegende Optimierungsproblem derart formuliert, dass eine Umformung in ein lineares Programm möglich ist. Die Herleitung des linearen Programms wird in den folgenden Abschnitten für verschiedene Situationen dargestellt. Eine Behandlung von modellprädiktiver Regelung und linearer Programmierung findet sich außerdem z.B. in den Arbeiten von C. V. Rao, J. B. Rawlings [23] und A. Bemporad, F. Borrelli, M. Morari [4]. Kapitel 6 dient abschließend der Veranschaulichung modellprädiktiver Regelung mittels linearer Programmierung. Es werden anhand zweier Beispiele die numerischen Ergebnisse dieses Verfahrens präsentiert. Die Beispiele stammen dabei aus [12] und [22] bzw. [27].

Kapitel 2

Einführung in die Kontrolltheorie

In diesem einführenden Kapitel sollen einige Grundlagen der mathematischen Kontrolltheorie vermittelt werden. Zunächst soll die Frage beantwortet werden, was man unter einem linearen Kontrollsystem versteht. Die hier beschriebenen Definitionen bilden die Basis der folgenden Kapitel. Außerdem sollen im weiteren Verlauf wichtige Begriffe wie zum Beispiel Kontrollierbarkeit und Stabilität geklärt und auf das der Arbeit zugrunde liegende Stabilisierungsproblem eingegangen werden.

Im Allgemeinen versteht man unter einem Kontrollsystem ein dynamisches System, das entweder in kontinuierlicher oder diskreter Zeit definiert ist und das außer vom aktuellen Zustand $x \in \mathbb{R}^n$ auch von einem beeinflussbaren Parameter $u \in \mathbb{R}^m$ abhängt. Dieser Parameter kann sich abhängig von der Zeit und/oder dem Zustand des Systems verändern. Wenn u lediglich von der Zeit abhängt, spricht man von Steuerung, bei einer zusätzlichen Abhängigkeit vom aktuellen Zustand $x(t) \in \mathbb{R}^n$ von Regelung. Der deutsche Begriff „Kontrolltheorie“ ist dabei eine etwas missverständliche Übersetzung des englischen Ausdrucks „Control Theory“. Es ist zu beachten, dass es sich hierbei nicht um Kontrolle im Sinne von Überwachung, sondern um Einflussnahme von außen handelt. Der Parameter u kann also als Steuergröße verstanden werden, die von außen aktiv beeinflusst werden kann (z.B. Beschleunigung bei einem Fahrzeug), aber auch als Störung, die auf das System wirkt (z.B. Straßenunebenheiten bei einem Auto, Kursschwankungen bei Wechselkursen). In dieser Arbeit wird sich im Folgenden nur auf Systeme mit linearer Dynamik beschränkt, da diese die Voraussetzung der vorgestellten Methode der modellprädiktiven Regelung mittels linearer Programmierung bilden. Für eine ausführliche Analyse nichtlinearer Kontrollsysteme wird auf [13] verwiesen. Der Aufbau dieses Kapitels orientiert sich an den Arbeiten von L. Grüne [12] und F. Colonijs [7] sowie an den Büchern von H.–W. Knobloch, H. Kwakernaak [17] und E. D. Sontag [26].

2.1 Lineare Kontrollsysteme

In diesem Abschnitt sollen lineare Kontrollsysteme definiert werden. Dazu müssen zunächst die Begriffe „dynamisches System“ und „gewöhnliche Differenzialgleichung“ geklärt werden.

Das Standardmodell eines dynamischen Systems ist eine endliche Anzahl von gewöhnlichen Differenzialgleichungen erster Ordnung

$$\frac{d}{dt}x(t) = f(t, x, u), \quad (2.1)$$

wobei

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix} \quad \text{und} \quad f(t, x, u) = \begin{pmatrix} f_1(t, x, u) \\ f_2(t, x, u) \\ \vdots \\ f_n(t, x, u) \end{pmatrix} \quad (2.2)$$

den Zustand $x \in \mathbb{R}^n$, die Eingabevariable $u \in \mathbb{R}^m$ und das Vektorfeld $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ bezeichnen. Außerdem beschreibt $t \in \mathbb{R}$ im Allgemeinen die Zeit. Ein Sonderfall solcher Systeme ist die sogenannte Zustandsgleichung, bei der das Vektorfeld f nicht explizit von u abhängt. Diese Gleichung wird auch (ungestörte) gewöhnliche Differenzialgleichung genannt.

Definition 2.1. (Gewöhnliche Differenzialgleichung)

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

$$\dot{x}(t) = \frac{d}{dt}x(t) = f(t, x(t)), \quad (2.3)$$

wobei $f : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ eine stetige Funktion ist. Die Notation $\dot{x}(t)$ steht im Folgenden für die zeitliche Ableitung $\frac{d}{dt}x(t)$.

Einen Spezialfall der Gleichung (2.3) stellt die sogenannte autonome oder zeitinvariante gewöhnliche Differenzialgleichung

$$\dot{x}(t) = f(x(t)) \quad (2.4)$$

dar. Dabei resultiert der Name „zeitinvariant“ daraus, dass sich die rechte Seite der Differenzialgleichung bei einem Übergang der Zeitvariablen t zu $\tau = t - a$ nicht ändert. Um eine eindeutige Lösung von (2.3) zu erhalten ist noch eine weitere Bedingung, die sogenannte Anfangswertbedingung, festzulegen. Dazu wird zu der Anfangszeit $t_0 \in \mathbb{R}$ ein Anfangszustand $x_0 \in \mathbb{R}^n$ vorgegeben und schließlich die Lösungsfunktion $x(t)$ gesucht, die (2.3) und zusätzlich $x(t_0) = x_0$ erfüllt. Im Weiteren wird die Lösung dieses Anfangswertproblems mit $x(t; t_0, x_0)$ bzw. für $t_0 = 0$ auch kurz mit $x(t; x_0)$ bezeichnet.

Ausgehend von diesen Grundlagen können die der folgenden Betrachtung zugrunde liegenden linearen zeitinvarianten Kontrollsysteme definiert werden. Im Allgemeinen werden Kontrollsysteme durch gewöhnliche Differenzialgleichungen der Form $\dot{x} = f(t, x(t), u(t))$ beschrieben.

Dabei wird $u(t) \in \mathbb{R}^m$ im Folgenden stets als Kontrolle oder Kontrollwert zur Zeit t bezeichnet. Außerdem sollen lineare Systeme betrachtet werden, die in kontinuierlicher Zeit $t \in \mathbb{R}$ definiert sind.

Definition 2.2. (Lineares zeitinvariantes Kontrollsystem)

Ein lineares zeitinvariantes Kontrollsystem ist gegeben durch die Differenzialgleichung

$$\dot{x}(t) = Ax(t) + Bu(t) =: f(x(t), u(t)) \quad (2.5)$$

mit $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$.

Diese Klasse von Kontrollsystemen ist besonders einfach, da das Vektorfeld linear in x und u ist und zudem nicht explizit von der Zeit t abhängt. Trotzdem kann mit solchen Modellen bereits eine große Anzahl realer Prozesse (z.B. für technische Anwendungen) beschrieben werden. Die in Kapitel 6 vorgestellten Beispiele verdeutlichen, dass tatsächlich auch heute noch viele lineare Modelle eingesetzt werden, wenn auch nicht immer in der einfachen Form (2.5). In der Praxis kann man nicht immer davon ausgehen, dass der gesamte Zustandsvektor $x \in \mathbb{R}^n$ vollständig meßbar ist. Dann kennt man nur gewisse von $x(t)$ abhängige Werte $y(t) = Cx(t) \in \mathbb{R}^l$, aus denen $u(t)$ dann berechnet werden muss. Da in dieser Arbeit nur lineare Systeme betrachtet werden, wird die Funktion $C : \mathbb{R}^n \rightarrow \mathbb{R}^l$ linear gewählt, also als eine Matrix $C \in \mathbb{R}^{l \times n}$.

Definition 2.3. (Lineares Kontrollsystem mit Ausgang)

Ein lineares Kontrollsystem mit Ausgang ist gegeben durch die Gleichungen

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \quad (2.6)$$

mit $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ und $C \in \mathbb{R}^{l \times n}$.

Außerdem werden in der Regelungstechnik oft zeitdiskrete Kontrollsysteme eingesetzt. Das zu (2.5) analoge zeitdiskrete Modell ist gegeben durch

$$x(t+1) = Ax(t) + Bu(t). \quad (2.7)$$

Hier ist sowohl die Kontrollfunktion $u(t)$ als auch die Lösung $x(t)$ nur für $t \in \mathbb{Z}$ definiert und die Dynamik ist durch eine Differenzgleichung gegeben. Das lineare Kontrollsystem (2.5) lässt sich jedoch bei Bedarf in ein entsprechendes System der Form (2.7) umwandeln. Setzt man in (2.5) eine Kontrollfunktion ein, die konstant auf Intervallen der Form $[k, k + t_{abt})$, $k \in \mathbb{Z}$, ist, so liefern beide Modelle für $t \in \mathbb{Z}$ die gleichen Lösungen, wenn man A und B in (2.7) als

$$A = e^{At_{abt}} \quad (2.8)$$

$$\text{und } B = \int_0^{t_{abt}} e^{A(t_{abt}-\tau)} B d\tau \quad (2.9)$$

wählt. Dabei bezeichnet t_{abt} die Abtastzeit. Die Abbildung 2.1 zeigt die Lösung eines in kontinuierlicher Zeit definierten Systems im Vergleich zu der des entsprechenden zeitdiskreten Systems.

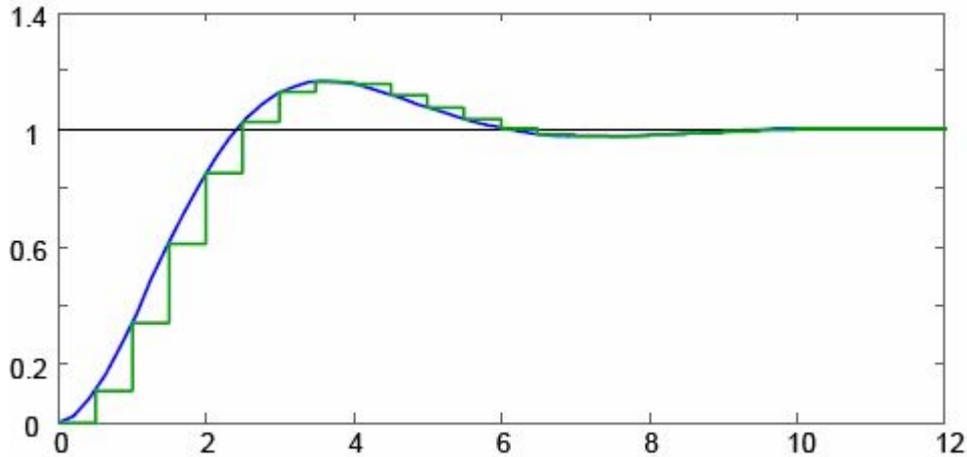


Abbildung 2.1: Vergleich der Lösungen – kontinuierlich (blau) und zeitdiskret (grün)

2.2 Existenz und Eindeutigkeit

Für die in den nächsten Abschnitten folgende Analyse werden nun wieder ausschließlich lineare Kontrollsystemen der Form (2.5) betrachtet. Wenn man sich mit Differentialgleichungen beschäftigt, muss zunächst immer die Existenz und Eindeutigkeit der Lösungen sicherstellt werden. Bevor ein Existenz- und Eindeutigkeitsatz für diese Klasse von Kontrollsystemen formuliert werden kann, muss dafür zuerst ein geeigneter Funktionenraum \mathcal{U} für die Kontrollfunktionen $u(\cdot)$ definiert werden. Sicherlich wären stetige Funktionen geeignet, diese Wahl ist aber zu einschränkend, da des öfteren Kokatationen von Kontrollfunktionen benötigt werden.

Definition 2.4. (Konkatenation von Funktionen)

Für zwei Funktionen $u_1, u_2 : \mathbb{R} \rightarrow \mathbb{R}^m$ und $\tau \in \mathbb{R}$ definiert man die Konkatenation zur Zeit τ als

$$u_1 \&_{\tau} u_2 := \begin{cases} u_1(t), & t < \tau \\ u_2(t), & t \geq \tau. \end{cases}$$

Zur Formulierung eines Existenz- und Eindeutigkeitsatzes wird nun ein Funktionenraum benötigt, der abgeschlossen unter Konkatenation ist. Eine gute Wahl ist sicherlich der Raum der stückweise stetigen Kontrollfunktionen.

Definition 2.5. (Stückweise stetige Funktionen)

Eine Funktion $u : \mathbb{R} \rightarrow \mathbb{R}^m$ heißt *stückweise stetig*, falls für jedes kompakte Intervall $[t_1, t_2]$ eine endliche Folge von Zeiten $t_1 = \tau_1 < \tau_2 < \dots < \tau_k = t_2$ existiert, so dass $u|_{(\tau_i, \tau_{i+1})}$ beschränkt und stetig ist für alle $i = 1, \dots, k-1$. Der Funktionenraum \mathcal{U} wird definiert als der Raum der stückweise stetigen Funktionen von \mathbb{R} nach \mathbb{R}^n .

Zudem existiert das Riemann-Integral

$$\int_{t_1}^{t_2} u(t) dt$$

über die Funktionen $u \in \mathcal{U}$, da es in jedem kompakten Integrationsintervall nur endliche viele Unstetigkeitsstellen gibt. Mit diesem Funktionenraum kann nun folgendes Resultat formuliert werden.

Satz 2.6. (Existenz und Eindeutigkeit)

Betrachte das lineare zeitinvariante Kontrollsystem (2.5)

$$\dot{x}(t) = Ax(t) + Bu(t)$$

mit $x : \mathbb{R} \rightarrow \mathbb{R}^n$ und gegebenen Matrizen $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$.

Dann gilt: Für jede Anfangsbedingung der Form

$$x(t_0) = x_0 \tag{2.10}$$

mit $t_0 \in \mathbb{R}$, $x_0 \in \mathbb{R}^n$ und jede stückweise stetige Kontrollfunktion $u \in \mathcal{U}$ existiert genau eine stetige Funktion $x : \mathbb{R} \rightarrow \mathbb{R}^n$, die (2.10) erfüllt und deren Ableitung für jedes t , in dem u stetig ist, existiert und (2.5) erfüllt. Diese eindeutige Funktion heißt *Lösung* von (2.5), (2.10) und wird mit $x(t; t_0, x_0, u)$ bezeichnet. Für diese Lösung gilt

$$x(t; t_0, x_0, u) = e^{A(t-t_0)}x_0 + \int_{t_0}^t e^{A(t-s)}Bu(s)ds. \tag{2.11}$$

Dabei ist die Matrixexponentialfunktion e^{At} für eine Matrix $A \in \mathbb{R}^{n \times n}$ und eine reelle Zahl $t \in \mathbb{R}$ gegeben durch

$$e^{At} := \sum_{k=0}^{\infty} A^k \frac{t^k}{k!}.$$

Beweis: Es soll zunächst nachgerechnet werden, dass (2.11) tatsächlich die Lösung im angegebenen Sinne ist. Es gilt:

$$\begin{aligned} \frac{d}{dt}x(t) &= \frac{d}{dt}e^{A(t-t_0)}x_0 + \frac{d}{dt} \int_{t_0}^t e^{A(t-s)}Bu(s)ds \\ &= Ae^{A(t-t_0)}x_0 + e^{A(t-t)}Bu(t) + \int_{t_0}^t \frac{d}{dt}e^{A(t-s)}Bu(s)ds \\ &= Ae^{A(t-t_0)}x_0 + Bu(t) + \int_{t_0}^t Ae^{A(t-s)}Bu(s)ds \end{aligned}$$

$$\begin{aligned}
&= A \underbrace{\left(e^{A(t-t_0)}x_0 + \int_{t_0}^t e^{A(t-s)}Bu(s)ds \right)}_{=x(t)} + Bu(t) \\
&= Ax(t) + Bu(t),
\end{aligned}$$

also (2.5). Zudem gilt

$$x(t_0) = e^{A(t_0-t_0)}x_0 + \int_{t_0}^{t_0} e^{A(t_0-s)}Bu(s)ds = Idx_0 + 0 = x_0,$$

also (2.10). Es bleibt noch die Eindeutigkeit zu zeigen. Dazu seien $y_1(t)$ und $y_2(t)$ zwei beliebige Lösungen von (2.5), (2.10). Dann gilt

$$\frac{d}{dt}(y_1(t) - y_2(t)) = Ay_1 + Bu(t) - Ay_2 - Bu(t) = A(y_1(t) - y_2(t)).$$

Man erhält damit eine Funktion $z(t) = y_1(t) - y_2(t)$, welche die Differenzialgleichung $\dot{z}(t) = Az(t)$ für alle $t \in \mathbb{R}$ löst. Da zudem

$$z(t_0) = y_1(t_0) - y_2(t_0) = x_0 - x_0 = 0$$

gilt, erfüllt z das Anfangswertproblem $\dot{z}(t) = Az(t)$, $z(t_0) = z_0$, dessen eindeutige Lösung nach dem Existenz- und Eindeigkeitssatz für lineare Differenzialgleichungen durch $z(t) = e^{A(t-t_0)}0 = 0$ gegeben ist. Also ist $y_1(t) = y_2(t)$ für alle $t \in \mathbb{R}$, womit die Eindeutigkeit folgt. □

2.3 Kontrollierbarkeit

Eine wichtige Eigenschaft bei der Analyse von Kontrollsystemen stellt die Kontrollierbarkeit dar. Dabei geht es um die Frage, für welche Punkte $x_0, x_1 \in \mathbb{R}^n$ und Zeiten t_1 eine Kontrollfunktion u gefunden werden kann, so dass die zwei Punkte durch eine Lösungstrajektorie verbunden werden. Formal wird dies wie folgt definiert.

Definition 2.7. (Kontrollierbarkeit und Erreichbarkeit)

Betrachte das lineare Kontrollsystem (2.5). Ein Zustand $x_0 \in \mathbb{R}^n$ heißt kontrollierbar zu einem Zustand $x_1 \in \mathbb{R}^n$ zur Zeit $t_1 > 0$, falls ein $u \in \mathcal{U}$ existiert, so dass

$$x_1 = x(t_1; x_0, u).$$

Der Zustand x_1 heißt dann erreichbar von x_0 zur Zeit t_1 .

Außerdem nennt man das Kontrollsystem (2.5) vollständig kontrollierbar, wenn jeder Zustand x_0 zu jedem anderen Zustand x_1 kontrollierbar ist. Das folgende Lemma zeigt, dass dies bei der Erreichbarkeit von $x_0 = 0$ der Fall ist.

Lemma 2.8. *Das Kontrollsystem (2.5) ist genau dann kontrollierbar, wenn jeder Zustand $x_1 \in \mathbb{R}^n$ von $x_0 = 0$ erreichbar ist.*

Beweis:

„ \implies “: (2.5) sei kontrollierbar.

Dann ist jedes x_1 von jedem x_0 erreichbar, also auch von $x_0 = 0$.

„ \impliedby “: Es sei der Zustand $\tilde{x}_1 = x_1 - x(t_1, x_0, 0)$ von $\tilde{x}_0 = 0$ erreichbar.

Dann gibt es ein $u(\cdot)$, so dass $x(t_1; \tilde{x}_0, u) = \tilde{x}_1$. Damit gilt für $x_0 \in \mathbb{R}^n$

$$\begin{aligned} x(t_1; x_0, u) &= e^{A(t_1-t_0)}x_0 + \int_{t_0}^{t_1} e^{A(t_1-s)}Bu(s)ds \quad (t_0 = 0) \\ &= e^{At_1}x_0 + \int_0^{t_1} e^{A(t_1-s)}Bu(s)ds \\ &= \underbrace{e^{At_1}x_0}_{=x(t_1; x_0, 0)} + \underbrace{\int_0^{t_1} e^{A(t_1-s)}Bu(s)ds + e^{At_1} \cdot 0}_{=x(t_1, \tilde{x}_0, u) = \tilde{x}_1} \\ &= \tilde{x}_1 + x(t_1; x_0, 0) = x_1 - x(t_1, x_0, 0) + x(t_1; x_0, 0) = x_1. \end{aligned}$$

Also ist auch jeder Zustand $x_1 \in \mathbb{R}^n$ von $x_0 \in \mathbb{R}^n$ erreichbar, womit gezeigt wurde, dass das System kontrollierbar ist. □

Die vollständige Kontrollierbarkeit des Systems (2.5) ist eine wichtige Voraussetzung zur Lösung des in Abschnitt 2.4 folgenden Stabilisierungsproblems. Deshalb soll die Kontrollierbarkeit bzw. Erreichbarkeit der Null genauer betrachtet werden.

Definition 2.9. (Kontrollierbarkeits- und Erreichbarkeitsmenge)

Betrachte das lineare Kontrollsystem (2.5).

(i) *Die Erreichbarkeitsmenge (reachable set) von $x_0 = 0$ zur Zeit $t \geq 0$ ist gegeben durch*

$$\mathcal{R}(t) = \{x(t; 0, u) | u \in \mathcal{U}\}.$$

(ii) *Die Kontrollierbarkeitsmenge (controllable set) nach $x_1 = 0$ zur Zeit $t \geq 0$ ist gegeben durch*

$$\mathcal{C}(t) = \{x_0 \in \mathbb{R}^n | \text{es existiert ein } u \in \mathcal{U} \text{ mit } x(t; x_0, u) = 0\}.$$

Die Verbindung von Lemma 2.8. und Definition 2.9. (i) ergibt also gerade, dass die vollständige Kontrollierbarkeit des Systems (2.5) für $\mathcal{R}(t) = \mathbb{R}^n$ gegeben ist. Diese Eigenschaft kann durch Bedingungen an das Matrizenpaar (A, B) sichergestellt werden.

Definition 2.10. (Erreichbarkeitsmatrix)

Die Matrix $(B \ AB \ \dots \ A^{n-1}B) \in \mathbb{R}^{n \times m \cdot n}$ heißt Erreichbarkeitsmatrix des Systems (2.5).

Die folgenden Kriterien charakterisieren die Kontrollierbarkeit.

Korollar 2.11. (Kalman–Kriterium)

Das System (2.5) ist genau dann vollständig kontrollierbar, wenn eine der folgenden äquivalenten Bedingungen erfüllt ist:

- (i) $\text{rg}(B \ AB \ \dots \ A^{n-1}B) = n$
- (ii) $\det(B \ AB \ \dots \ A^{n-1}B) \neq 0$.

In diesem Fall heißt das Matrizenpaar (A, B) kontrollierbar.

Der folgende Satz liefert alternative Definitionen der Kontrollierbarkeit, die ohne die Berechnung der Kontrollierbarkeitsmatrix auskommen. Hierbei bezeichnet $(\lambda Id - A|B) \in \mathbb{R}^{n \times (n+m)}$ die Matrix, die durch Nebeneinanderschreiben der Matrizen $(\lambda Id - A)$ und B entsteht.

Satz 2.12. (Hautus–Kriterium)

Die folgenden Bedingungen sind äquivalent:

- (i) (A, B) ist kontrollierbar
- (ii) $\text{rg}(\lambda Id - A|B) = n$ für alle $\lambda \in \mathbb{C}$
- (iii) $\text{rg}(\lambda Id - A|B) = n$ für alle Eigenwerte $\lambda \in \mathbb{C}$ von A .

Dieser Satz soll an dieser Stelle nicht bewiesen werden. Es sei dafür auf [12] verwiesen.

2.4 Stabilität und Stabilisierung

In diesem Abschnitt soll das Problem der Stabilisierung linearer Kontrollsysteme untersucht werden. Dazu muss zunächst geklärt werden, was man unter Stabilität versteht.

Im Folgenden werden wichtige Resultate der Stabilitätstheorie linearer zeitinvarianter Differenzialgleichungen der Form

$$\dot{x}(t) = Ax(t) \tag{2.12}$$

eingeführt. Bei der Analyse soll sich dabei auf die Stabilität von Gleichgewichten beschränkt werden, für eine ausführlichere Behandlung dieses Themas siehe z.B. [15]. Anschließend kann das Stabilisierungsproblem für lineare Kontrollsysteme definiert werden.

Definition 2.13. (Gleichgewicht)

Ein Punkt $x^* = 0 \in \mathbb{R}^n$ heißt Gleichgewicht (oder Ruhelage) einer gewöhnlichen Differenzialgleichung, falls für die zugehörige Lösung

$$x(t; x^*) = x^* \quad \text{für alle } t \in \mathbb{R}$$

gilt.

Offensichtlich ist ein Punkt x^* genau dann Gleichgewicht der Differentialgleichung $\dot{x}(t) = f(x(t))$, wenn $f(x^*) = 0$. Insbesondere ist für die lineare Differentialgleichung (2.12) der Punkt $x^* = 0$ immer ein Gleichgewicht. Dieses Gleichgewicht $x^* = 0$ soll deshalb in der folgenden Analyse näher betrachtet werden.

Definition 2.14. (Stabilität)

Sei $x^* = 0$ das Gleichgewicht der linearen Differenzialgleichung (2.12).

- (i) Das Gleichgewicht $x^* = 0$ heißt stabil, falls für jedes $\varepsilon > 0$ ein $\delta > 0$ existiert, so dass die Ungleichung

$$\|x(t; x_0)\| \leq \varepsilon \quad \text{für alle } t \geq 0$$

für alle Anfangswerte $x_0 \in \mathbb{R}^n$ mit $\|x_0\| \leq \delta$ erfüllt ist.

- (ii) Das Gleichgewicht $x^* = 0$ heißt lokal asymptotisch stabil, falls es stabil ist und darüberhinaus

$$\lim_{t \rightarrow \infty} x(t; x_0) = 0$$

gilt für alle Anfangswerte x_0 aus einer offenen Umgebung U von $x^* = 0$.

- (iii) Das Gleichgewicht $x^* = 0$ heißt global asymptotisch stabil, falls (ii) mit $U = \mathbb{R}^n$ erfüllt ist.

- (iv) Das Gleichgewicht $x^* = 0$ heißt lokal bzw. global exponentiell stabil, falls Konstanten $c, \sigma > 0$ existieren, so dass die Ungleichung

$$\|x(t; x_0)\| \leq ce^{-\sigma t} \quad \text{für alle } t \geq 0$$

für alle x_0 aus einer Umgebung U von $x^* = 0$ (mit $U = \mathbb{R}^n$ im globalen Fall) erfüllt ist.

Beachte, dass aus den Definitionen die Implikationen

$$(\text{lokal/global}) \text{ exponentiell stabil} \implies (\text{lokal/global}) \text{ asymptotisch stabil} \implies \text{stabil}$$

folgen. Die zweite Implikation ergibt sich direkt aus der Definition. Dass aus exponentieller Stabilität die asymptotische Stabilität folgt, sieht man folgendermaßen:

Für ein gegebenes ε folgt (i) mit $\delta = \varepsilon/c$, denn damit gilt für $\|x_0\| \leq \delta$ die Ungleichung $\|x(t; x_0)\| \leq ce^{-\sigma t}\|x_0\| \leq c\|x_0\| \leq \varepsilon$. Die in (ii) geforderte Konvergenz ist offensichtlich.

Der folgende Satz gibt ein Kriterium an die Matrix A , mit dem man Stabilität leicht überprüfen kann.

Satz 2.15. (Eigenwertkriterium)

Betrachte die lineare zeitinvariante Differentialgleichung (2.12) für eine Matrix $A \in \mathbb{R}^{n \times n}$. Seien $\lambda_1, \dots, \lambda_d \in \mathbb{C}$, $\lambda_l = a_l + ib_l$, die Eigenwerte der Matrix A . Dann gilt:

- (i) Das Gleichgewicht $x^* = 0$ ist stabil genau dann, wenn alle Eigenwerte λ_l nichtpositiven Realteil $a_l > 0$ besitzen und für alle Eigenwerte mit Realteil $a_l = 0$ der entsprechende Jordanblock eindimensional ist.
- (ii) Das Gleichgewicht $x^* = 0$ ist lokal asymptotisch stabil genau dann, wenn alle Eigenwerte λ_l negativen Realteil $a_l < 0$ besitzen.

Beweis: siehe [13]

2.4.1 Lyapunov–Funktionen

Nun soll ein wichtiges Hilfsmittel zur Untersuchung asymptotische stabiler Differentialgleichungen behandelt werden, nämlich die sogenannten Lyapunov–Funktionen. Asymptotische (und auch exponentielle) Stabilität verlangen nur, dass die Norm $\|x(t)\|$ einer Lösung für $t \rightarrow \infty$ abnimmt. Für viele Anwendungen (insbesondere zur Überprüfung von asymptotischer Stabilität) wäre jedoch strenge Monotonie eine sehr praktische Eigenschaft. Dies ist im Allgemeinen nicht zu erwarten. Man kann diese strenge Monotonie aber erhalten, indem man die euklidische Norm $\|x(t)\|$ durch eine allgemeine Funktion, und zwar gerade die Lyapunov–Funktion, ersetzt. Für lineare Systeme kann man sich auf sogenannte quadratische Lyapunov–Funktionen beschränken.

Definition 2.16. (Lyapunov–Funktion)

Sei $A \in \mathbb{R}^{n \times n}$. Eine stetig differenzierbare Funktion $V : \mathbb{R}^n \rightarrow \mathbb{R}_0^+$ heißt (quadratische) Lyapunov–Funktion für A , falls positive reelle Konstanten $c_1, c_2, c_3 > 0$ existieren, so dass die Ungleichungen

$$c_1\|x\|^2 \leq V(x) \leq c_2\|x\|^2 \quad (2.13)$$

und

$$DV(x) \cdot Ax \leq c_3\|x\|^2 \quad (2.14)$$

für alle $x \in \mathbb{R}^n$ gelten.

Bevor nun ein Satz zur Charakterisierung der Stabilität mittels Lyapunov–Funktionen formuliert werden kann, müssen zunächst einige Hilfsresultate angegeben werden, welche an dieser Stelle nicht bewiesen werden sollen. Dabei wird eine speziellen Klasse von Lyapunov–Funktionen betrachtet, bei denen V durch eine Bilinearform der Form $x^T P x$ dargestellt wird.

Lemma 2.17. Seien $A, P \in \mathbb{R}^{n \times n}$ und $c_3 > 0$, so dass die Funktion $V(x) = x^T P x$ die Ungleichung

$$DV(x) \cdot Ax \leq -c_3 \|x\|^2$$

für alle $x \in \mathbb{R}^n$ erfüllt. Dann gilt: P ist genau dann positiv definit, wenn A exponentiell stabil ist. In diesem Fall ist V eine quadratische Lyapunov-Funktion.

Außerdem kann das Ableitungskriterium vereinfacht werden, indem man die bilineare Form der Lyapunov-Funktion ausnutzt.

Lemma 2.18. Für eine bilineare Lyapunov-Funktion $V(x) = x^T P x$ sind äquivalent:

- (i) $DV(x) \cdot Ax \leq -c_3 \|x\|^2$ für alle $x \in \mathbb{R}^n$ und eine Konstante $c_3 > 0$
- (ii) Die Matrix $C = -A^T P - P A$ ist positiv definit.

Die Gleichung in Lemma 2.18.(ii) wird auch Lyapunov-Gleichung genannt. Es liegt nun nahe, diese Gleichung zur Konstruktion von Lyapunov-Funktionen zu verwenden. Die Frage ist, wann zu einer gegebenen Matrix A und einer gegebenen positiv definiten Matrix C eine Matrix P gefunden werden kann, so dass $A^T P + P A = -C$ gilt? Das folgende Lemma beantwortet diese Frage.

Lemma 2.19. Für eine Matrix $A \in \mathbb{R}^{n \times n}$ und eine positiv definite Matrix $C \in \mathbb{R}^{n \times n}$ hat die Lyapunov-Gleichung

$$A^T P + P A = -C \tag{2.15}$$

genau dann eine (sogar eindeutige) positiv definite Lösung $P \in \mathbb{R}^{n \times n}$, wenn A exponentiell stabil ist, d.h., falls die Realteile aller Eigenwerte λ_i von A negativ sind.

Der folgende Satz beschreibt nun das Hauptresultat dieses Abschnittes:

Satz 2.20. (Charakterisierung der Stabilität (Lyapunov))

Für $A \in \mathbb{R}^{n \times n}$ gilt: Eine quadratische Lyapunov-Funktion existiert genau dann, wenn die Matrix A exponentiell stabil ist.

Beweis:

„ \implies “: Sei V eine quadratische Lyapunov-Funktion.

Aus der Ableitungsbedingung für $x = x(\tau, x_0)$ folgt

$$\left. \frac{d}{dt} \right|_{t=\tau} V(x(t; x_0)) = DV(x(\tau; x_0)) \cdot \dot{x}(\tau; x_0) = DV(x(\tau; x_0)) \cdot Ax(\tau; x_0) \leq -c_3 \|x(\tau; x_0)\|^2.$$

Wegen $-\|x\|^2 \leq -V(x)/c_2$ folgt damit für $\lambda = c_3/c_2$ die Ungleichung

$$\frac{d}{dt} V(x(t; x_0)) \leq -\lambda V(x(t; x_0)).$$

Aus dieser Differentialungleichung folgt die Ungleichung

$$V(x(t; x_0)) \leq e^{-\lambda t} V(x_0).$$

Mit den Abschätzungen für $V(x)$ erhält man daraus

$$\|x(t; x_0)\|^2 \leq \frac{1}{c_1} e^{-\lambda t} V(x_0) \leq \frac{c_2}{c_1} e^{-\lambda t} \|x_0\|^2$$

und damit durch Ziehen der Quadratwurzel auf beiden Seiten die Ungleichung

$$\|x(t; x_0)\| \leq c e^{-\sigma t} \|x_0\|$$

für $c = \sqrt{c_1/c_2}$ und $\sigma = \lambda/2$, d.h. die Matrix A ist exponentiell stabil.

„ \Leftarrow “: Sei A exponentiell stabil.

Dann existiert nach Lemma 2.19. eine positiv definite Matrix P , die die Lyapunov Gleichung (2.15) für eine positiv definite Matrix C löst. Wegen Lemma 2.18. und Lemma 2.17. ist $V(x) = x^T P x$ dann eine quadratische Lyapunov-Funktion.

□

Die Existenz einer quadratischen Lyapunov Funktion ist also eine notwendige und hinreichende Bedingung für die exponentielle Stabilität der Matrix A und liefert damit eine Charakterisierung, die äquivalent zu der Eigenwertbedingung ist.

2.4.2 Das Stabilisierungsproblem für lineare Kontrollsysteme

In diesem Abschnitt soll sich nun wieder dem linearen Kontrollsystem (2.5) gewidmet werden. Da die Vorausberechnung der Kontrollfunktion $u(t)$ im Allgemeinen auf großen Zeithorizonten (z.B. wegen Modellfehlern oder Störungen) nicht gut funktioniert, um ein System in einem gegebenen Arbeitspunkt zu regeln und dort zu halten, soll hier ein anderer Ansatz vorgestellt werden. Statt die Kontrolle als Steuerung (open-loop-Kontrolle) abhängig von der Zeit t anzusetzen, soll jetzt eine Regelung (closed-loop-Kontrolle oder Feedback-Kontrolle) gewählt werden, in der die Kontrollfunktion in jedem Zeitpunkt zustandsabhängig als $u(t) = Fx(t)$ für eine zu bestimmende Funktion $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$ gegeben ist. In der Praxis wird man im Allgemeinen Feedback-Lösungen bevorzugen, da diese auf Modellfehler und Störungen reagieren und sie idealerweise korrigieren können. Eine solche Funktion F , die jedem Zustand einen Kontrollwert zuordnet, nennt man Feedback. Da hier lineare Systeme betrachtet werden, soll auch die Feedback-Funktion F linear gewählt werden, also $u = Fx$ für eine Matrix $F \in \mathbb{R}^{m \times n}$. Diese Methode hat den Vorteil, dass man anstatt des ursprünglichen Kontrollsystems (2.5) eine zeitinvariante Differentialgleichung der Form

$$\dot{x}(t) = Ax(t) + BFx(t) = (A + BF)x(t) \tag{2.16}$$

erhält, auf die die Theorie des vorherigen Abschnittes angewendet werden kann. Das Stabilisierungsproblem kann nun wie folgt formuliert werden.

Definition 2.21. (Stabilisierungsproblem für lineare Systeme)

Gegeben sei ein lineares Kontrollsystem (2.5)

$$\dot{x}(t) = Ax(t) + Bu(t)$$

mit Matrizen $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{m \times n}$. Das (Feedback-) Stabilisierungsproblem für (2.5) besteht darin, eine lineare Abbildung $F : \mathbb{R}^m \rightarrow \mathbb{R}^n$ (bzw. die dazugehörige Matrix $F \in \mathbb{R}^{m \times n}$) zu finden, so dass die gewöhnliche Differenzialgleichung

$$\dot{x}(t) = (A + BF)x(t)$$

asymptotisch stabil ist.

Aus dem Eigenwertkriterium für asymptotische Stabilität kann das folgende Lemma angegeben werden.

Lemma 2.22. Gegeben seien zwei Matrizen $A \in \mathbb{R}^{n \times n}$ und $B \in \mathbb{R}^{n \times m}$. Dann löst die Matrix $F \in \mathbb{R}^{m \times n}$ das Stabilisierungsproblem, falls alle Eigenwerte der Matrix $(A + BF) \in \mathbb{R}^{n \times n}$ negativen Realteil besitzen.

In der Praxis steht oft nicht der gesamte Zustand $x(t)$ eines Systems zur Verfügung, stattdessen hat man nur Zugriff auf einen Ausgangsvektor $y = Cx$ für eine Matrix $C \in \mathbb{R}^{l \times n}$. In diesem Fall kann ein Feedback F nur vom Ausgangsvektor y abhängen, man spricht dann von einem Ausgangsfeedback.

Die wichtigste Frage bei der Analyse von Kontrollsystemen mit Ausgang (2.6) ist dabei, wie viel „Information“ im Ausgang $y(t) = Cx(t)$ enthalten ist. Dies wird durch die folgenden Definitionen formalisiert.

Definition 2.23. (Beobachtbarkeit)

(i) Zwei Zustände $x_1, x_2 \in \mathbb{R}^n$ heißen unterscheidbar, falls ein $u \in \mathcal{U}$ und ein $t > 0$ existieren, so dass

$$Cx(t; x_1, u) \neq Cx(t; x_2, u).$$

(ii) Das System (2.6) heißt beobachtbar, falls alle Zustände $x_1, x_2 \in \mathbb{R}^n$ mit $x_1 \neq x_2$ unterscheidbar sind.

Eine genaue Analyse des Stabilisierungsproblems für Kontrollsysteme mit Ausgang kann in [12] nachgelesen werden.

Im Folgenden sollen jetzt Bedingungen untersucht werden, unter denen eine Lösung des Stabilisierungsproblems für lineare Kontrollsysteme aus Definition 2.16. (zunächst mit eindimensionaler Kontrolle) gefunden werden kann. Insbesondere wird eine hinreichende und

notwendige Bedingung an die Matrizen A und B in (2.5) angegeben, unter der das Problem lösbar ist.

Für eine Transformationsmatrix $T \in \mathbb{R}^{n \times n}$ ist das zu (2.5) transformierte System gegeben durch

$$\dot{x}(t) = \tilde{A}x(t) + \tilde{B}u(t) \quad (2.17)$$

mit $\tilde{A} = T^{-1}AT$ und $\tilde{B} = T^{-1}B$. Ein Feedback F für (2.5) wird mittels $\tilde{F} = FT$ in eines für (2.17) transformiert. Dies folgt aus der Bedingung $T^{-1}(A + BF)T = \tilde{A} + \tilde{B}\tilde{F}$. Das folgende Resultat ist auf den Fall $m = 1$ beschränkt, also auf Kontrollsysteme mit eindimensionaler Kontrolle. Die Matrix B ist dann ein n -dimensionaler Spaltenvektor $B \in \mathbb{R}^{n \times 1}$.

Lemma 2.24. (Regelungsnormalform)

Sei $A \in \mathbb{R}^{n \times n}$ und $B \in \mathbb{R}^{n \times 1}$. Dann gilt: das Paar (A, B) ist kontrollierbar genau dann, wenn es eine invertierbare Matrix $S \in \mathbb{R}^{n \times n}$ gibt, so dass $\tilde{A} = S^{-1}AS$ und $\tilde{B} = S^{-1}B$ die Form

$$\tilde{A} = \begin{pmatrix} 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \end{pmatrix} \quad \text{und} \quad \tilde{B} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

besitzen, wobei die Werte $\alpha_i \in \mathbb{R}$ gerade die Koeffizienten des charakteristischen Polynoms von A sind, d.h. $\chi_A(z) = z^n - \alpha_n z^{n-1} - \dots - \alpha_2 z - \alpha_1$. Diese Form der Matrizen wird *Regelungsnormalform* genannt.

Für einen Beweis sei auf [19] verwiesen.

Die folgenden Ergebnisse (Lemma 2.25. und Definition 2.21.) können für beliebige Kontrolldimensionen $u \in \mathbb{R}^m$ formuliert werden. Ist das Matrizenpaar (A, B) nicht kontrollierbar, kann das System durch eine geeignete Koordinatentransformation in einen kontrollierbaren und einen unkontrollierbaren Anteil zerlegt werden. Dies zeigt das folgende Lemma.

Lemma 2.25. (Kalman Kontrollierbarkeits-Zerlegung)

Sei (A, B) nicht kontrollierbar, d.h. $d = \text{rg}(B \ AB \ \dots \ A^{n-1}B) < n$. Dann existiert ein invertierbares $T \in \mathbb{R}^{n \times n}$, so dass

$$\tilde{A} = T^{-1}AT = \begin{pmatrix} A_1 & A_2 \\ 0 & A_3 \end{pmatrix} \quad \text{und} \quad \tilde{B} = T^{-1}B = \begin{pmatrix} B_1 \\ 0 \end{pmatrix}$$

mit $A_1 \in \mathbb{R}^{d \times d}$, $A_2 \in \mathbb{R}^{d \times (n-d)}$, $A_3 \in \mathbb{R}^{(n-d) \times (n-d)}$ und $B_1 \in \mathbb{R}^{d \times m}$, wobei das Paar (A_1, B_1) kontrollierbar ist.

Beweis: hier sei auf [12] verwiesen.

Mithilfe der Regelungsnormalform kann nun die Lösung des Stabilisierungsproblems angegeben werden. Dafür wird das Stabilisierungsproblem zunächst mittels des charakteristischen Polynoms ausgedrückt.

Definition 2.26. (Vorgebbare Polynome)

Betrachte ein Kontrollsystem (2.5) mit Matrizen $A \in \mathbb{R}^{n \times n}$ und $B \in \mathbb{R}^{n \times m}$. Ein Polynom χ heißt vorgebar für das Kontrollsystem, falls ein lineares Feedback $F \in \mathbb{R}^{m \times n}$ existiert, so dass $\chi = \chi_{A+BF}$ ist für das charakteristische Polynom χ_{A+BF} der Matrix $(A + BF)$.

Bemerkung 2.27. Da die Nullstellen des charakteristischen Polynoms gerade die Eigenwerte der zugehörigen Matrix sind, existiert nach Lemma 2.22. genau dann ein stabilisierendes Feedback F , wenn ein vorgebares Polynom für $(A + BF)$ existiert, dessen Nullstellen über \mathbb{C} alle negative Realteile haben.

Der folgende Satz zeigt abschließend die Beziehung zwischen der Kontrollierbarkeit des Paares (A, B) und der Vorgebarkeit von Polynomen.

Satz 2.28. (Polvorgabe)

Betrachte ein Kontrollsystem (2.5) mit Matrizen $A \in \mathbb{R}^{n \times n}$ und $B \in \mathbb{R}^{n \times 1}$, d.h. mit eindimensionaler Kontrolle. Dann sind die folgenden zwei Eigenschaften äquivalent.

- (i) Das Paar (A, B) ist kontrollierbar.
- (ii) Jedes Polynom der Form $\chi(z) = z^n - \beta_n z^{n-1} - \dots - \beta_2 z - \beta_1$ mit $\beta_1, \dots, \beta_n \in \mathbb{R}$ ist vorgebar.

Beweis:

„(i) \implies (ii)“: Sei (A, B) kontrollierbar und sei S die Koordinatentransformation aus Lemma 2.24. Setzt man

$$\tilde{F} = (\beta_1 - \alpha_1 \quad \beta_2 - \alpha_2 \quad \dots \quad \beta_n - \alpha_n) \in \mathbb{R}^{1 \times n}$$

gilt

$$\begin{aligned} \tilde{A} + \tilde{B}\tilde{F} &= \begin{pmatrix} 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \cdot (\beta_1 - \alpha_1 \quad \beta_2 - \alpha_2 \quad \dots \quad \beta_n - \alpha_n) \\ &= \begin{pmatrix} 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ \beta_1 & \beta_2 & \dots & \beta_n \end{pmatrix}. \end{aligned}$$

Aus der zweiten Aussage von Lemma 2.24. folgt, dass $\chi_{\tilde{A} + \tilde{B}\tilde{F}} = \chi$. Also ist, nach Rücktransformation, $F = \tilde{F}S^{-1}$ die gesuchte Feedback-Matrix, da das charakteristische Polynom einer

Matrix invariant unter Koordinatentransformationen ist.

„(ii) \implies (i)“: es soll hier „nicht (i) \implies nicht (i)“ gezeigt werden.

Sei (A, B) nicht kontrollierbar und sei T die Koordinatentransformation aus Lemma 2.25. Dann ergibt sich für jedes beliebige Feedback $\tilde{F} = (F_1 \ F_2)$

$$\tilde{A} + \tilde{B}\tilde{F} = \begin{pmatrix} A_1 + B_1F_1 & A_2 + B_1F_2 \\ 0 & A_3 \end{pmatrix} = \tilde{D}.$$

Für das charakteristische Polynom dieser Matrix gilt

$$\chi_{\tilde{D}} = \chi_{A_1+B_1F_1} \chi_{A_3},$$

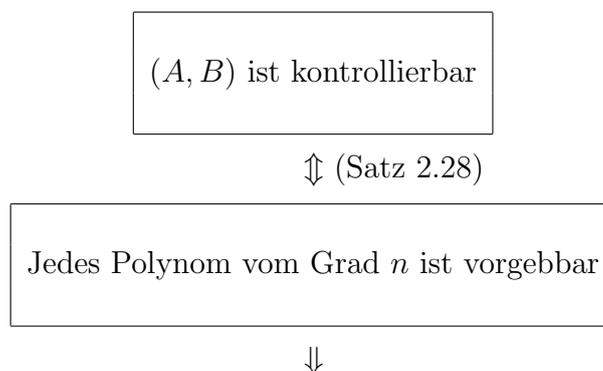
und das Paar (A_1, B_1) ist kontrollierbar. Daher sind die vorgebbaren Polynome gerade der Form $\chi = \chi_c \chi_u$, wobei χ_c ein beliebiges normiertes Polynom vom Grad d und $\chi_u = \chi_{A_3}$. Dies ist gerade das charakteristische Polynom des unkontrollierbaren Anteil der Matrix A . Dies sind sicherlich weniger als die in (ii) angegebenen Polynome, weshalb (ii) nicht gelten kann.

□

Bemerkung 2.29. Zur Stabilisierung ist es natürlich nicht notwendig, dass jedes Polynom vorgebbar ist. Man braucht lediglich eines zu finden, dessen Nullstellen nur negative Realteile besitzen. Der Satz von der Polvorgabe bringt eine wesentliche Eigenschaft steuerbarer Systeme zum Ausdruck, nämlich die Möglichkeit, die Eigenwerte von $(A + BF)$ durch Wahl von F in beliebige Positionen zu bringen. Insbesondere kann man durch die Zustandsrückführung ein instabiles steuerbares System stabilisieren.

Die Resultate für mehrdimensionale Kontrolle $m > 1$ sind völlig analog zu denen für eindimensionale Kontrolle. Bei einer direkten Herangehensweise sind allerdings die Beweise etwas aufwändiger, da nicht direkt auf Lemma 2.24. zurückgegriffen werden kann. Allerdings kann man den mehrdimensionalen Fall auf den Fall $m = 1$ zurückführen.

Es sollen nun abschließend die Ergebnisse über das Stabilisierungsproblem schematisch dargestellt werden:



\Downarrow

Es gibt ein vorgebbares
Polynom, dessen Nullstellen
alle negativen Realteil besitzen

\Leftrightarrow
(Bem. 2.27)

(A, B) ist
stabilisierbar

\Uparrow (Beweis Satz 2.28)

(A, B) ist nicht kontrollierbar und A_3 aus Lemma 2.25
hat nur Eigenwerte mit negativem Realteil

Kapitel 3

Modellprädiktive Regelung (MPC)

In diesem Kapitel wird nun das Verfahren der modellprädiktiven Regelung (MPC) behandelt. Dabei soll zunächst sowohl auf die Problemstellung als auch auf die Zielsetzung dieses Konzeptes eingegangen werden. Danach soll die Idee des modellprädiktiven Regelungsansatzes veranschaulicht werden und verschiedene Ausgestaltungsmöglichkeiten der wichtigsten Elemente der Strategie, wie zum Beispiel die Wahl des zugrunde liegenden Modells oder der Zielfunktion vorgestellt werden. Abschließend wird die Idee der modellprädiktiven Regelung zusammengefasst. Diese Kapitel orientiert sich dabei an der Darstellung von E. F. Camacho und C. Bordons [5]. Tiefere Einblicke in die Theorie der modellprädiktiven Regelung geben außerdem die Arbeiten [18], [10] und [21] sowie das Buch [8].

3.1 Problemstellung der modellprädiktiven Regelung

In den vergangenen Jahren hat sich das Konzept der modellprädiktiven Regelung in der Industrie als eine der beliebtesten Regelungsstrategien etabliert. Das praktische Interesse an dieser Strategie resultiert vor allem daraus, dass Prozesse heutzutage unter schärferen Leistungsanforderungen untersucht werden. Ständig steigende ökologische, ökonomische und sicherheitstechnische Anforderungen führen dazu, dass die betrachteten Prozesse über einen großen Arbeitsbereich betrieben werden müssen. MPC eignet sich außerdem für beschränkte, multivariable Systeme und für Kontrollprobleme, bei denen die Offline-Berechnung der Kontrollfunktion sehr schwierig oder sogar unmöglich ist. Vor allem Eigenschaften wie Stabilität und Robustheit führten zu großer Akzeptanz.

Die Idee der modellprädiktiven Regelung (englisch: model predictive control) stammt aus den späten siebziger Jahren und wurde seitdem kontinuierlich weiterentwickelt. Der Ausdruck „Modellprädiktive Regelung“ beschreibt dabei nicht einen spezifischen Regelalgorithmus, sondern vielmehr eine Strategie, welche mit verschiedenen Algorithmen umgesetzt werden kann. MPC ist eine Methode, mit der Regler sowohl für lineare als auch für nichtlineare Systeme durch Verfahren der Online-Optimierung berechnet werden können. Die Problemstellung

der modellprädiktiven Regelung ist eine verallgemeinerte Form des Stabilisierungsproblems – und zwar das „tracking–Problem“. Das Ziel ist, den Ausgang $y(t)$ in einen gegebenen Arbeitspunkt zu regeln. Das heißt, es soll innerhalb eines bestimmten Zeitfensters – dem sogenannten Prädiktionshorizont – die Abweichung des Ausganges $y(t)$ zu einer bekannten Referenztrajektorie $y_{ref}(t)$ minimiert werden, um das System so nahe wie möglich an dieser Trajektorie zu halten. Das Problem besteht also darin auf einem endlichen Prädiktionshorizont eine optimale Kontrollfolge vorherzusagen, durch dessen Anwendung auf das System das gewünschte Verhalten erzielt wird.

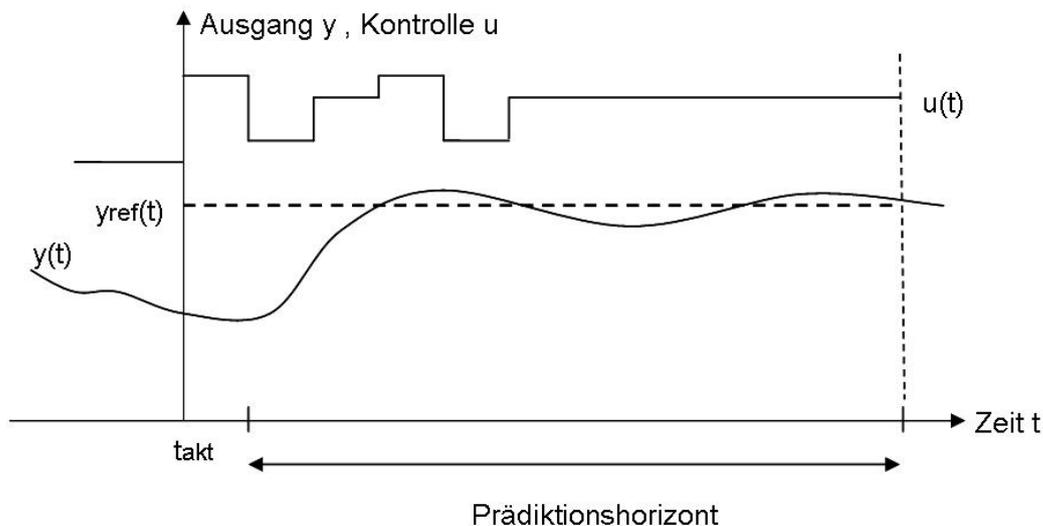


Abbildung 3.1: Zielsetzung der MPC–Strategie

Natürlich ist es auch möglich, direkt den Zustand $x(t)$ in einen Arbeitspunkt zu regeln. Voraussetzung dafür ist jedoch, dass der gesamte Zustandsvektor bekannt ist. Da dies in der Praxis nicht häufig der Fall ist und auf einen von $x(t)$ abhängigen Ausgang zurückgegriffen werden muss, soll in diesem Kapitel einheitlich nur die Regelung eines Ausganges $y(t)$ betrachtet werden.

Im Folgenden wird die Idee zur Lösung dieses Problems beschrieben. Diese kann vorab wie folgt angegeben werden: Zunächst wird online ein optimales Kontrollproblem gelöst, woraus man die optimale Kontrollfolge auf einem Kontrollhorizont erhält. Anschließend wird nur das erste Glied dieser Kontrollfolge auf das System angewendet, die restlichen Kontrollfunktionen werden verworfen. Im nächsten Schritt wird der daraus resultierende Zustand $x(t + 1)$ als Anfangszustand betrachtet und die Online–Berechnung mit verschobenem Zeithorizont fortgesetzt. Dieses Verfahren berechnet eine Kontrollfolge, welche mit Hilfe von digitalen Rechnern, die nur zeitdiskret arbeiten können, in der Lage ist ein zeitkontinuierliches System zu regeln.

Den meisten MPC-Algorithmen liegen folgende gemeinsame Elemente zugrunde.

- Das Modell, um den Ausgang $y(t)$ zur Zeit $t > t_{akt}$ (:=aktueller Zeitpunkt) zu berechnen
- Die Zielfunktion des Optimierungsproblems zur Berechnung der optimalen Kontrollfolge u
- Das „receding-horizon“ Konzept

Es existieren jedoch verschiedene Gestaltungsmöglichkeiten. Die Lösungsverfahren können sich daher durch das verwendete, den Prozess beschreibende Modell und die zu minimierende Zielfunktion unterscheiden.

3.2 Lineare MPC-Modelle

Das Modell spielt in der modellprädiktiven Regelung eine zentrale Rolle. Es muss das dynamische Verhalten des zu regelnden Prozesses so genau wie nötig wiedergeben. Dazu können verschiedene Modellstrukturen verwendet werden. In Industrieanwendungen wird oft ein Impuls- oder Sprungantwortmodell (impulse response- oder step response model) verwendet. Dieses Modell beschreibt das Ausgangsverhalten aufgrund eines Impulses oder Sprunges an den Prozesseingängen. Dieses Verhalten kann durch Versuche an der nicht geregelten Anlage gemessen werden. Dadurch entfällt eine physikalische Modellbildung des Systems. Allerdings ist dies nur bei asymptotisch stabilen Prozessen möglich und ergibt eine große Anzahl von Modellparametern. Wenn ein lineares physikalisches Modell des Prozesses vorhanden ist, können Modelle mittels Übertragungsfunktionen (transfer functions) oder der Zustandsraumdarstellung (state space models) beschrieben werden. Diese Beschreibungen benötigen im Vergleich zu den Impuls- und Sprungantwortmodellen oft weniger Parameter und können auch für instabile Prozesse aufgestellt werden. Zur Modellbildung sind jedoch genaue Kenntnisse über die Vorgänge der zu beschreibenden Prozesse nötig. Ist eine linearisierte Darstellung des Prozesses nicht genügend genau, muss eine nichtlineare Darstellung verwendet werden. Außerdem können auch Störungen berücksichtigt werden, um das nicht durch das Prozessmodell widerspiegelte Verhalten wie zum Beispiel nicht messbare Eingänge, Messgeräusche oder Modellfehler zu beschreiben. In einer MPC-Formulierung können folglich zahlreiche Modelle verwendet werden, die folgenden sind jedoch am Gebräuchlichsten.

3.2.1 Zustandsmodelle (state space models)

Modelle in der Zustandsraumdarstellung wurden bereits in Kapitel 2 behandelt. Ein lineares Kontrollsystem mit Ausgang (2.6) wurde dort wie folgt definiert

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t).\end{aligned}$$

Dabei beschreibt $t \in \mathbb{R}$ die Zeit, $x \in \mathbb{R}^n$ den Zustand, $y \in \mathbb{R}^l$ den Ausgang und $u \in \mathbb{R}^m$ die Kontrolle. Außerdem sind $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ und $C \in \mathbb{R}^{l \times n}$ die Matrizen des Systems. Viele modellprädiktive Algorithmen verwenden im Gegensatz zu (2.6) auch zeitdiskrete Modelle. Das dem Kontrollsystem (2.6) entsprechende zeitdiskrete Modell ist gegeben durch

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t), \end{aligned} \quad (3.1)$$

wobei nun $t \in \mathbb{Z}$. Der Ausgang $y(t)$ kann für dieses Modell rekursiv für $t > t_{akt}$ berechnet werden

$$y(t) = Cx(t) = C \left(A^{(t-t_{akt})} x(t_{akt}) + \sum_{r=0}^{t-t_s-1} A^r Bu(t-1-r) \right). \quad (3.2)$$

Bemerkung 3.1. Ein Vorteil solcher Zustandsmodelle ist, dass diese für multivariable Prozesse verwendet werden können. Die optimale Kontrollfolge ist dann einfach die Rückführung einer Linearkombination des Zustandsvektors. Die Berechnung des Ausganges wird jedoch deutlich komplizierter, wenn aufgrund nicht messbarer Zustände zusätzlich Beobachter eingeführt werden müssen. Wenn aber Modelle dieser Form für ein gegebenes Problem bekannt sind und in einem gegebenen Algorithmus effizient verwendet werden können, werden normalerweise diese Modelle benutzt.

3.2.2 Übertragungsfunktionen (transfer function models)

Es kann jedoch sein, dass geeignete Zustandsmodelle nicht bekannt sind oder die Modelle sich aus strukturellen Gründen für gewisse Algorithmen nicht eignen. Aus diesen Gründen gibt es weitere Modelle, die alternativ verwendet werden können. Es gibt eine Reihe von Modellen, in denen der Zustand $x(t)$ gar nicht auftaucht, da sie direkt durch eine Abbildung von u nach y gegeben sind. Da $x(t)$ dabei völlig vernachlässigt wird, fixiert man üblicherweise $x_0 = 0$, was für lineare Systeme keinen wesentlichen Informationsverlust bedeutet, da man die Ausgangsfunktion \tilde{y} für $\tilde{x}_0 \neq 0$ durch Addition von $Ce^{At}\tilde{x}_0$ bzw. $CA^t\tilde{x}_0$ zu y erhalten kann. Die wahrscheinlich wichtigste Klasse solcher Modelle sind die Übertragungsfunktionen (transfer functions):

Hierbei werden im kontinuierlichen Fall $t \in \mathbb{R}$ die Funktionen u und y zunächst durch ihre Laplace-Transformationen

$$\hat{u}(s) = \int_0^\infty u(t)e^{-st} dt, \quad \hat{y}(s) = \int_0^\infty y(t)e^{-st} dt \quad (3.3)$$

für $s \in \mathbb{C}$ dargestellt (Darstellung im Frequenzbereich). Die Transferfunktion ist dann die Abbildung $G : \mathbb{C} \rightarrow \mathbb{R}^{l \times m}$, für welche die Relation

$$\hat{y}(s) = G(s)\hat{u}(s) \quad (3.4)$$

für $x_0 = 0$ gilt. Für die von (2.6) induzierte y - u -Relation gilt gerade $G(s) = C(sId - A)^{-1}B$. Umgekehrt lassen sich zu jeder Übergangsfunktion G Matrizen A , B und C finden, so dass (2.6) das gleiche Verhalten wie G beschreibt. Im zeitdiskreten Fall $t \in \mathbb{Z}$ kann man analog vorgehen, wenn man statt der Laplace-Transformation eine z -Transformation verwendet. Ein anderes Modell dieser Art ist das sogenannte ARIMAX-Modell, das aus der Zeitreihenanalyse stammt und bei dem die Beziehung zwischen y und u mittels einer Gleichung der Form

$$a_0y(t) + a_1y(t-1) + \dots + a_{r_a}y(t-r_a) = b_0u(t-1) + b_1u(t-2) + \dots + b_{r_b}u(t-r_b) \quad (3.5)$$

beschrieben wird. Hierbei werden die Parameter a_i und b_i durch statistische Methoden aus dem gemessenen Systemverhalten geschätzt. Definiert man den Shift- (Verzögerungs-) Operator q^{-1} mittels $q^{-1}y(t) := y(t-1)$, die Potenz $(q^{-1})^k$ als k -malige Ausführung dieses Operators und die Polynome $A(\lambda) = \sum_{k=0}^{r_a} a_k \lambda^k$ sowie $B(\lambda) = \sum_{k=0}^{r_b} b_k \lambda^k$, kann man (3.5) kurz als

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) \quad (3.6)$$

schreiben. Für Modelle der Form (3.6) ist die Berechnung des Ausgangs für $t > t_{akt}$ gegeben durch

$$y(t) = \frac{B(q^{-1})}{A(q^{-1})}u(t-1). \quad (3.7)$$

Bemerkung 3.2. Wie bereits in der Einleitung erwähnt, benötigen solche Übertragungsfunktionen meist wenige Parameter und können auch für instabile Prozesse aufgestellt werden. Allerdings ist ein *a priori* Wissen über den Prozess notwendig, insbesondere der Grad der Polynome A und B .

3.2.3 FIR-Modelle (impulse response-, step response models)

Impulse-response Modelle

Auch Impulse-response Modelle sind direkt durch eine Abbildung von u nach y gegeben. Die Beziehung zwischen Ausgang und Kontrolle kann hier durch die Gleichung

$$y(t) = \sum_{i=1}^{\infty} h_i u(t-i) \quad (3.8)$$

beschrieben werden. Dabei bezeichnet h_i den Ausgang zur Abtastzeit $t+i$, der sich durch Anwendung eines einheitlichen Impuls ergibt (siehe Abbildung 3.2). Die unendliche Reihe in (3.8) kann auf N Glieder beschränkt, womit man

$$y(t) = \sum_{i=1}^N h_i u(t-i) = H(q^{-1})u(t) \quad (3.9)$$

mit $H(q^{-1}) = h_1q^{-1} + h_2q^{-2} + \dots + h_Nq^{-N}$ erhält (q^{-1} ist wieder der Shift-Operator). Auf diese Weise können jedoch nur stabile Prozesse beschrieben werden. Ein weiterer Nachteil dieser Methode ist, dass eine große Anzahl von Modellparametern benötigt wird, da N normalerweise sehr groß ist. Auf der anderen Seite sind impulse-response Modelle in der Industrie sehr beliebt, da sie sehr intuitiv sind und den Einfluss jeder Kontrolle auf den Ausgang deutlich machen.

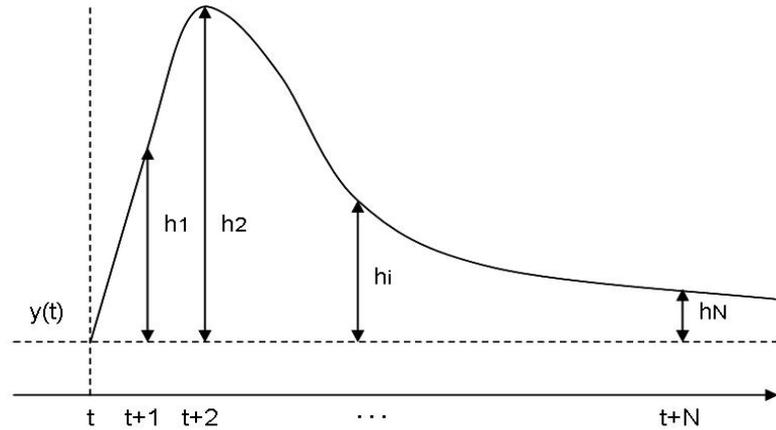


Abbildung 3.2: Impulse Response

Step-response Modelle

Step-response Modelle sind den gerade beschriebenen Modellen sehr ähnlich, jedoch wird hier das Ausgangsverhalten durch einen Sprung an den Prozesseingängen beschrieben. Für stabile Systeme ist ein solches Modell gegeben als

$$y(t) = y_0 + \sum_{i=1}^N g_i \Delta u(t-i) = y_0 + G(q^{-1})(1-q^{-1})u(t), \quad (3.10)$$

wobei g_i den Ausgang des i -ten Kontrollschrittes bezeichnet, außerdem gilt $\Delta u(t) = u(t) - u(t-1)$ (siehe Abbildung 3.3). Diese Methode hat dieselben Vor- und Nachteile wie die des vorherigen Abschnittes. y_0 kann im Folgenden Null gesetzt werden, da dadurch für lineare Systeme kein erheblicher Informationsverlust entsteht. Die Berechnung des Ausganges für $t > t_{akt}$ kann also mittels der Gleichung

$$y(t) = \sum_{i=1}^N g_i \Delta u(t-i) \quad (3.11)$$

erfolgen. Da ein Impuls als die Abweichung zweier Schritte mit der Verzögerung einer Abtastperiode betrachtet werden kann, können beide Modelle für ein lineares Systems mittels

$$h_i = g_i - g_{i-1}, \quad \text{bzw.} \quad g_i = \sum_{j=1}^i h_j \quad (3.12)$$

transformiert werden.

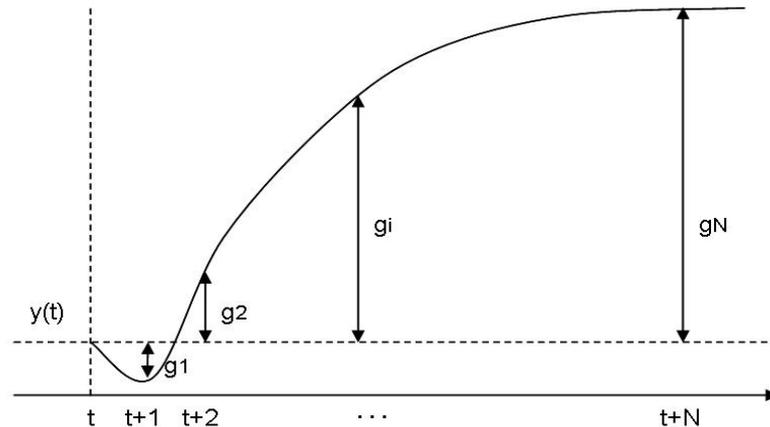


Abbildung 3.3: Step Response

3.3 Die Zielfunktion

Der zweite zentrale Teil der MPC-Strategie ist die Optimierung. Ist das Modell linear, das Gütekriterium quadratisch und sind keine Beschränkungen der Kontrolle, der Ausgangs- oder Zustandsgrößen gegeben, kann die Lösung explizit angegeben werden. Kommen Beschränkungen dazu, so muss die Lösung numerisch bestimmt werden. Auch bei nichtlinearen Modellen ist eine numerische Optimierung nötig. Die Komplexität des Optimierungsproblems und damit der numerische Aufwand sind abhängig von der Modellstruktur, den Beschränkungen und der Anzahl der zu optimierenden Parameter. Bei schnell zu regelnden Prozessen mit kleinen Zeitschritten ist zu beachten, dass der Zeitbedarf für die Berechnung der Optimierung nicht größer wird als die Schrittweite. Bei verfahrenstechnischen und thermischen Prozessen ist die Schrittweite meistens groß, so dass von dieser Seite keine Probleme zu erwarten sind.

Die zahlreichen MPC-Algorithmen verwenden zur Berechnung der optimalen Kontrollfolge verschiedene Zielfunktionen. Das Ziel der modellprädiktiven Regelung war, den Ausgang $y(t)$

auf einem endlichen Prädiktionshorizont in einen Arbeitspunkt zu regeln und im Idealfall gleichzeitig den dafür notwendigen Aufwand zu minimieren. Dazu wird nun ein Optimierungsproblem mit zugehöriger Zielfunktion $J(u, y(t))$ aufgestellt. Die Art und Weise wie das System die gewünschte Referenztrajektorie erreicht, hängt dabei von der Wahl der Zielfunktion ab. Die Sollwerttrajektorie y_{ref} wird dabei als bekannt vorausgesetzt.

3.3.1 Das quadratische Kriterium

Die Wahl eines quadratischen Kriteriums ist wohl die am Häufigsten verwendete Form der Zielfunktion $J(u, y(t))$. Die allgemeinste Formulierung solch einer Zielfunktion lautet

$$J_1(u, y(t)) = \sum_{t=N_1}^{N_2} \delta(t)[y(t) - y_{ref}(t)]^2 + \sum_{t=N_3}^{N_4} \lambda(t)[\Delta u(t-1)]^2. \quad (3.13)$$

Die Parameter N_1 und N_2 bilden dabei die Grenzen des Prädiktionshorizontes, dessen Länge im Folgenden mit N_y bezeichnet wird. Außerdem sind N_3 und N_4 die untere bzw. obere Schranke des Kontrollhorizontes der Länge N_u , welcher nicht notwendigerweise mit dem Prädiktionshorizont zusammenfallen muss. Falls der Prädiktionshorizont länger ist als der Kontrollhorizont, d.h. $N_y > N_u$, gilt

$$\Delta u(t) = 0 \quad \text{für } t > N_4 + 1. \quad (3.14)$$

Das ist äquivalent zu einer stärkeren Gewichtung des Kontrollaufwandes Δu ab diesem Zeitpunkt. Außerdem kann N_1 so groß gewählt werden, dass für zu vernachlässigende Abweichungen in den ersten Instanzen keine unnötige Berechnung erforderlich ist. Die Koeffizienten $\delta(j)$ und $\lambda(j)$ sind Gewichtungsvariablen, welche das Verhalten des Ausgangs und den Kontrollaufwand berücksichtigen. Wählt man zum Beispiel

$$\delta(t) = \alpha^{N_2-t} \quad \text{und} \quad \lambda(t) = 1$$

wird für $\alpha \in [0, 1]$ der zweite Term in (3.13) stärker gewichtet. Das bedeutet, dass die Minimierung des Kontrollaufwands im Vordergrund steht. Umgekehrt wird bei einer Wahl von $\alpha > 1$ die Abweichung von $y(t)$ und $y_{ref}(t)$ mehr bestraft als der zur Minimierung notwendige Kontrollaufwand. All die beschriebenen Variablen können also als „tuning“-Parameter verwendet werden. In einigen Anwendungen wird der zweite Term in (3.13) sogar völlig vernachlässigt. In anderen Fällen treten nicht die Kontrolländerungen, sondern die Kontrollen selbst direkt in der Zielfunktion auf.

Ein weiterer Vorteil prädiktiver Regelung ist, dass die Referenztrajektorie *a priori* bekannt ist, d.h. das System kann reagieren, bevor eine Veränderung eintritt. Somit kann eine verzögerte Reaktion vermieden werden. Die Referenztrajektorie $y_{ref}(t)$ ist in vielen Anwendungen vorher bekannt. In anderen Fällen kann man sogar für konstante Werte y_{ref} eine deutliche Verbesserung des Verhaltens erzielen, wenn man den Zeitpunkt kennt, zu dem sich dieser

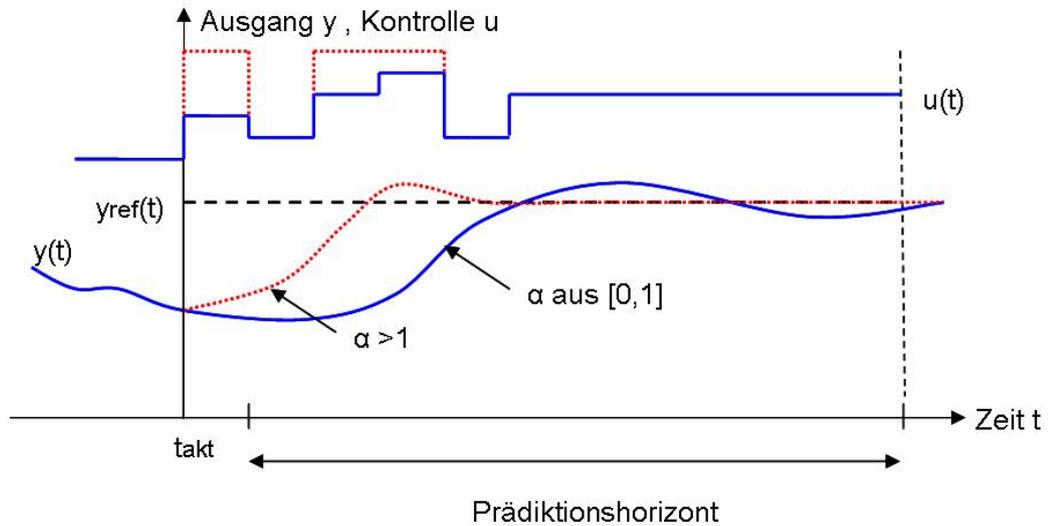


Abbildung 3.4: Auswirkung der Wahl der Gewichtungswinkel

Wert ändert. Bei der Optimierung von (3.13) verwenden die meisten MPC-Algorithmus eine Näherung \tilde{y}_{ref} , welche nicht notwendigerweise mit der tatsächlichen Referenztrajektorie y_{ref} übereinstimmen muss. Eine mögliche Approximation von y_{ref} ist gegeben durch

$$\tilde{y}_{ref}(t_{akt}) = y(t_{akt}), \quad \tilde{y}_{ref}(t) = \beta \tilde{y}_{ref}(t-1) + (1+\beta)y_{ref}(t), \quad k = 1, \dots, N.$$

Dabei wird der Parameter $\beta \in [0, 1]$ gewählt (je näher der 1, desto gleichmäßiger die Approximation). Die Abbildung 3.5 zeigt die konstante Referenztrajektorie $y_{ref}(t)$ im Vergleich zu den Approximationen für zwei verschiedene β .

Vielfältige Gründe, wie z.B. Sicherheitsbarrieren oder physikalische Grenzen, können Beschränkungen der Prozessvariablen verursachen (z.B. der Wasserpegel in Tanks, Maximaltemperaturen und -druck). Weiterhin sind die Operationsbedingungen aus grundlegenden ökonomischen Gründen durch den Schnittpunkt verschiedener Beschränkungen definiert, so dass das Kontrollsystem nahe an den Grenzen operieren wird. Zum Beispiel ist es für ein Wirtschaftsunternehmen, welches nur eine bestimmte Schadstoffmenge ausstoßen darf, wünschenswert, so nahe wie möglich an dieser vorgegebenen Grenze zu operieren. All das macht das Einführen von Beschränkungen notwendig. Solche Restriktionen sind meist gegeben durch

$$u_{min} \leq u(t) \leq u_{max} \quad (3.15)$$

$$y_{min} \leq y(t) \leq y_{max} \quad (3.16)$$

für alle t .

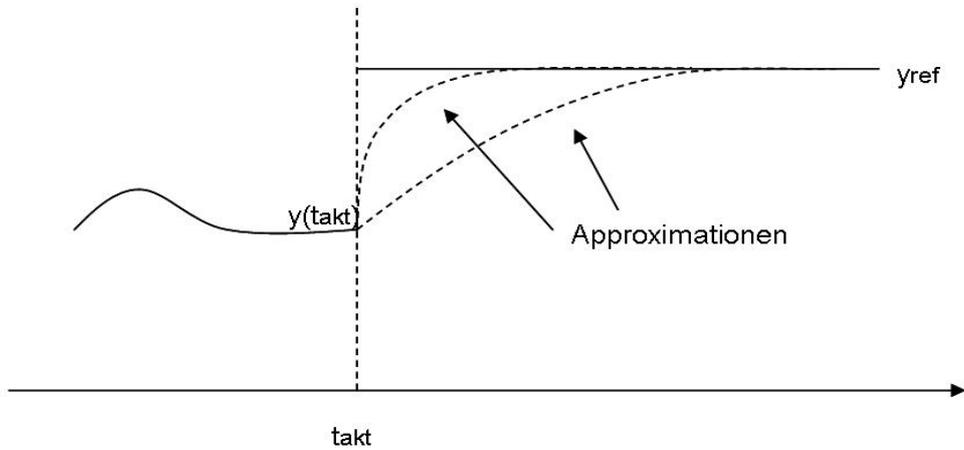


Abbildung 3.5: y_{ref} im Vergleich zur Approximation \tilde{y}_{ref}

Eine Erweiterung des Optimierungsproblems um solche Nebenbedingungen macht das Problem natürlich komplexer, so dass die Lösung im Gegensatz zum unbeschränkten Fall nicht explizit berechnet werden kann.

3.3.2 1-Norm und ∞ -Norm

Die Zielfunktion $J(u, y(t))$ kann jedoch auch durch andere Kriterien beschrieben werden (zum Beispiel mittels einer 1-Norm oder ∞ -Norm). Diese Methode bietet ebenfalls einige Vorteile. Die Zielfunktion kann bei Verwendung einer ∞ -Norm wie folgt angegeben werden.

$$J_2(u, y(t)) = \max_{t=N_1, \dots, N_2} \|y(t) - y_{ref}(t)\|_\infty = \max_{t=N_1, \dots, N_2} \max_{i=1, \dots, n} |y_i(t) - y_{ref,i}(t)| \quad (3.17)$$

Dabei bezeichnen y_i und $y_{ref,i}$ jeweils den i -ten Eintrag der Vektoren $y, y_{ref} \in \mathbb{R}^l$. Durch Lösung des der modellprädiktiven Regelung zugrunde liegenden Optimierungsproblems wird hier die maximalen Abweichungen von $y(t)$ zu $y_{ref}(t)$ für den schlimmstmöglichen Fall minimiert, der Kontrollaufwand wird hier vernachlässigt. Es wurde bereits gezeigt, dass das mit (3.17) folgende Min-Max Problem als ein lineares Programm formuliert werden kann, welches dann mit geringerem Rechenaufwand und mit bekannten Standardalgorithmen gelöst werden kann.

Es ist ebenso möglich eine 1-Norm zur Formulierung der Zielfunktion zu verwenden. Ein Grund für diese Wahl ist, dass bei Verwendung einer ∞ -Norm nur die maximalen Ab-

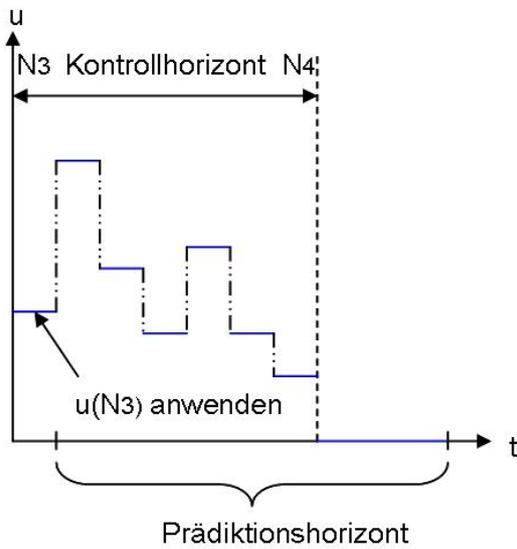
weichungen und nicht das gesamte Verhalten explizit berücksichtigt wird. Eine Zielfunktion der Form

$$\begin{aligned}
 J_3(u, y(t)) &= \sum_{t=N_1}^{N_2} \|y(t) - y_{ref}(t)\|_1 + \lambda \sum_{t=N_3}^{N_4} \|\Delta u(t)\|_1 \\
 &= \sum_{t=N_1}^{N_2} \sum_{i=1}^n |y_i(t) - y_{ref,i}(t)| + \lambda \sum_{t=N_3}^{N_4} \sum_{j=1}^m |\Delta u_j(t)| \quad (3.18)
 \end{aligned}$$

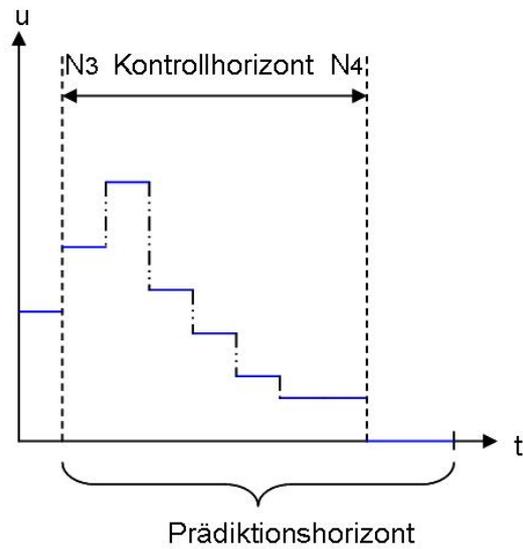
beschreibt gerade die absoluten Werte der Abweichungen von $x(t)$ und $x_{ref}(t)$ und (zusätzlich) der Kontrolländerungen. Das daraus resultierende Minimierungsproblem kann dann ebenfalls in ein lineares Programm umgeschrieben werden, was das Ziel des Kapitels 5 sein wird. Dort wird eine Zielfunktion der Form (3.18) verwendet, in der jedoch nicht $\Delta u(t)$, sondern die Kontrollen direkt auftreten. Außerdem werden später beide Terme in (3.18) gewichtet.

3.4 Das „receding–horizon“ Konzept

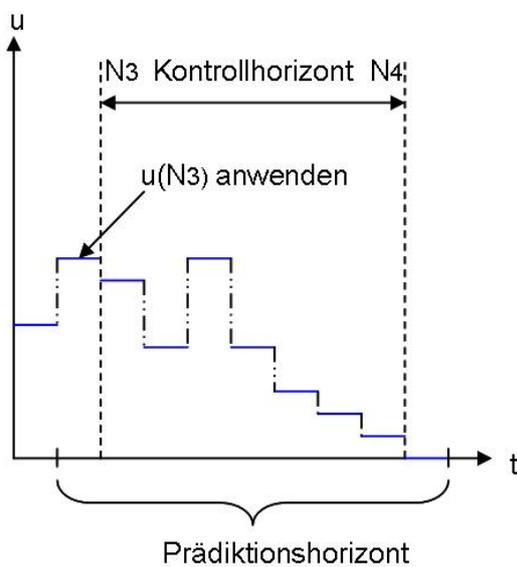
Das „receding–horizon“ Konzept ist wichtiger Bestandteil aller MPC–Algorithmen. Diese Strategie beruht dabei auf folgender Idee (der Formulierung liegt eine zeitdiskreten Darstellung zugrunde, da diese im weiteren Verlauf verwendet wird): die nach Lösung des Optimierungsproblems – mit einer Zielfunktion der Form (3.13), (3.17) oder (3.18) – auf dem Kontrollhorizont berechnete optimale Kontrollfolge wird nicht komplett auf das System angewendet, sondern es wird nur das erste Glied $u(N_3)$ zur Berechnung des Zustandes $x(t+1)$ verwendet. Dieser Zustand wird dann als Anfangszustand betrachtet und die Optimierung **mit verschobenem Horizont**, d.h. $t = t + 1$, wiederholt. Die folgende Graphik dient zur Verdeutlichung dieses Konzeptes. Eine genaue Beschreibung mit den in Kapitel 5 verwendeten Bezeichnungen folgt außerdem noch in 5.2.



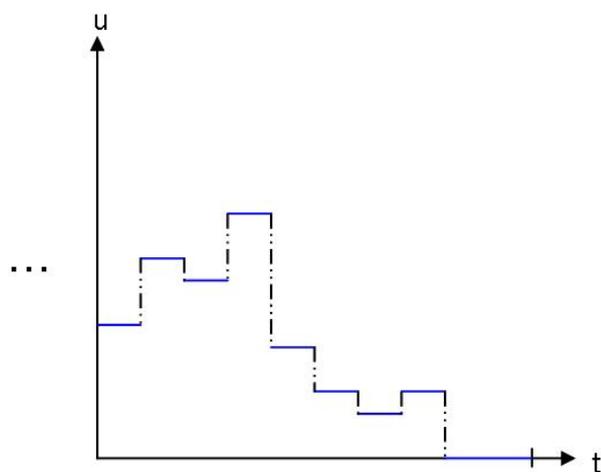
Schritt 1: Berechnung der optimalen Kontrollfolge auf dem Kontrollhorizont und Anwendung des ersten Gliedes



Schritt 2: Verschieben des Kontrollhorizontes und Berechnung einer neuen optimalen Kontrollfolge auf dem Kontrollhorizont



Wiederholung der Schritte 1. und 2.



Optimale Kontrollfolge

3.5 Der modellprädiktive Regelungsansatz

Der modellprädiktive Regelungsansatz kann nun abschließend wie folgt zusammengefasst werden.

Zeitdiskrete Darstellung:

1. Zur Zeit $t = t_{akt}$ wird ein Minimierungsproblem mit Zielfunktional $J(u, y(t))$ gelöst. Diese Optimierung liefert die optimale Kontrollfolge $u = [u(N_3) \dots u(N_4)]$ auf dem Kontrollhorizont.
2. Von der berechneten Kontrollfolge wird dann nur das erste Glied $u(N_3)$ (N_3 untere Grenze des Kontrollhorizontes) auf das Kontrollsystem angewendet, d.h. mit diesem Kontrollwert lässt sich der Zustand $x(t + 1)$ bestimmen.
3. Anschließend wird der Kontrollhorizont verschoben, d.h. setze $t = t + 1$.
4. Die Optimierung wird nun mit verschobenem Horizont wiederholt, dabei dient $x(t + 1)$ als neuer Anfangszustand.
5. Diese Vorgehensweise wird iterativ fortgesetzt bis $t = N_2 - 1$. (N_2 obere Schranke des Prädiktionshorizontes)
6. Man erhält schließlich die optimale Kontrollfolge, durch dessen Anwendung das System auf dem Prädiktionshorizont in den gegebenen Arbeitspunkt geregelt wird.

Kapitel 4

Lineare Programmierung

Wie bereits in Kapitel 3 erwähnt, kann das der modellprädiktiven Regelung zugrunde liegende Optimierungsproblem durch geeignete Wahl der Zielfunktion in ein lineares Programm umgeschrieben werden, welches dann mit bekannten Algorithmen der linearen Programmierung gelöst werden kann. Bevor diese Vorgehensweise in Kapitel 5 beschrieben wird, sollen zunächst einige Grundlagen der linearen Programmierung behandelt werden. Dieses Kapitel folgt der Darstellung von [16]. Eine Einführung in die lineare Optimierung findet sich außerdem z.B. in [11], [20] oder [29].

4.1 Lineare Programme

4.1.1 Definition

Ziel der linearen Programmierung ist, eine lineare Funktion in Abhängigkeit von einer Menge linearer Gleichungen und Ungleichungen zu optimieren. Diese lineare Funktion soll zunächst definiert werden. Es seien $c = c_1, \dots, c_n$ und $x = x_1, \dots, x_n \in \mathbb{R}^n$. Dann nennt man f mit

$$f(x) = c^T x = \sum_{i=1}^n c_i x_i \quad (4.1)$$

die Zielfunktion des linearen Programms. Die allgemeinste Form eines linearen Programms ist dann

$$\begin{aligned} \min_x \quad & c^T x \\ \text{s.t.} \quad & \underline{b} \leq Ax \leq \bar{b} \\ & l \leq x \leq u, \end{aligned} \quad (4.2)$$

wobei die Eingabedaten aus einer Matrix $A \in \mathbb{R}^{m \times n}$, den Vektoren $\underline{b}, \bar{b} \in \mathbb{R}^m$ und den Vektoren $c, l, u \in \mathbb{R}^n$ bestehen. Es soll im Folgenden die Notation $s \geq t$ für Vektoren $s, t \in \mathbb{R}^n$ verwendet werden, wenn $s_i \geq t_i$ gilt für alle $i \in \{1, \dots, n\}$. Der Vektor Ax soll

also im linearen Programm (kurz LP) also komponentenweise zwischen \underline{b} und \bar{b} liegen. In „Summennotation“ hat das Problem (4.2) die Form

$$\begin{aligned} \min_x \quad & \sum_{i=1}^n c_i x_i & (4.3) \\ \text{s.t.} \quad & \underline{b}_j \leq \sum_{i=1}^n A_{ji} x_i \leq \bar{b}_j & \text{für } 1 \leq j \leq m \\ & l_i \leq x_i \leq u_i & \text{für } 1 \leq i \leq n. \end{aligned}$$

Die Komponenten \underline{b}_j und l_i dürfen dabei in $\mathbb{R} \cup \{-\infty\}$ liegen bzw. \bar{b}_j und u_i in $\mathbb{R} \cup \{\infty\}$. Falls $\underline{b}_j > \bar{b}_j$ für ein j oder $l_i > u_i$ für ein i hat das lineare Programm offenbar keine Lösung. Lineare Gleichungen können durch die Wahl $\underline{b}_j = \bar{b}_j$ dargestellt werden. Außerdem werden meistens die entsprechenden Ungleichungen weggelassen, falls $\underline{b}_j = -\infty$ oder $\bar{b}_j = \infty$. Eine Ungleichung heißt aktiv in \bar{x} , wenn der Punkt \bar{x} , der alle Nebenbedingungen erfüllt, die Ungleichung mit Gleichheit erfüllt. Falls also z.B. $\sum_{i=1}^n A_{ji} \bar{x}_i = \bar{b}_j$ gilt, ist die Ungleichung $\sum_{i=1}^n A_{ji} x_i \leq \bar{b}_j$ in \bar{x} aktiv. Im Folgenden wird fast ausschließlich die einfachere und kürzere Schreibweise (4.2) benutzt. Die Summennotation soll nur verwendet werden, wenn sie zur Erklärung eines Ansatzes nötig ist.

4.1.2 Die Standardform

Zur leichteren Darstellung eines Lösungsverfahrens soll nun gezeigt werden, wie das allgemeine lineare Programm (4.2) in eine gewisse Standardform umgeformt werden kann. Offenbar sind Minimierungs- und Maximierungsprobleme äquivalent. Es gilt

$$\min c^T x = - \max -c^T x.$$

Das Problem (4.2) als Minimierungsproblem zu schreiben ist also keine wirkliche Einschränkung. Außerdem lässt sich jede Ungleichung der Form

$$a_j^T x \leq \bar{b}_j$$

mit einem Zeilenvektor a_j^T von A und einem endlichem \bar{b}_j mittels einer zusätzlich eingeführten „Schlupfvariable“ s_j in eine lineare Gleichung und eine Nichtnegativitätsbedingungen umformen,

$$a_j^T x + s_j = \bar{b}_j \quad \text{und} \quad s_j \geq 0.$$

Ähnliches gilt für Ungleichungen der Form $\underline{b}_j \leq a_j^T x$ und die Schranken l_i sowie u_i von x_i . Auf diese Weise erhält man ein zu (4.2) äquivalentes lineares Programm der Form

$$\begin{aligned} \inf_x \quad & c^T x & (4.4) \\ \text{s.t.} \quad & Ax = b \\ & x_i \geq 0 & \text{für } i \in I \end{aligned}$$

mit einem neuen Variablenvektor x , der auch die zusätzlich eingeführte Schlupfvariable enthält, und mit geänderten Vektoren b , c sowie geänderter Matrix A . Die Variable x_i mit $i \in I$ heißen vorzeichenbeschränkte Variablen, die übrigen freie Variablen.

Lineare Programme dieses Typs lassen sich mit geeigneten Varianten des Simplexverfahrens lösen, das nach wie vor eines der wichtigsten Verfahren zur Lösung linearer Programme darstellt und dessen Grundform in nächsten Abschnitt vorgestellt wird. Diese Methode und dessen Beschreibung lässt sich erheblich vereinfachen, wenn man sie auf lineare Programme (4.4) ohne freie Variablen anwendet, d.h. auf lineare Programme in **Standardform**

$$\inf \{c^T x \mid Ax = b, x \geq 0\}, \quad (4.5)$$

in dem alle Variablen vorzeichenbeschränkt sind. Im Folgenden wird mit

$$\mathcal{P} := \{x \mid Ax = b, x \geq 0\} \quad (4.6)$$

die Menge der zulässigen Lösungen oder zulässigen Punkte von (4.5) bezeichnet.

Man kann nun jedes lineare Programm der Form (4.4), das noch freie Variablen enthält, in ein äquivalentes lineares Programm in Standardform transformieren. Dabei kann zum Beispiel ausgenutzt werden, dass sich jede reelle Zahl als Differenz zweier nichtnegativer Zahlen schreiben lässt. Ersetzt man daher jede freie Variable x_i , $i \notin I$ durch die Differenz zweier neuer vorzeichenbeschränkter Variablen ($x_i = x'_i - x''_i$), erhält man ein zu (4.4) äquivalentes LP in Standardform (4.5). Diese Methode hat jedoch den Nachteil, dass sich die Anzahl der Variablen und der Spalten von A vergrößert. Dies kann jedoch durch Eliminationstechniken vermieden werden, in denen durch eine sukzessive Elimination aller freien Variablen schließlich sogar ein äquivalentes lineares Programm in Standardform mit geringerer Anzahl von Variablen entsteht:

Sei x_i eine freie Variable und A_i die i -te Spalte von A . Man unterscheidet folgende Fälle.

1. $A_i = 0$ und $c_i = 0$:

Dann erhält man durch Streichen der Variablen x_i , der Spalte A_i und der Komponente c_i des Vektors c ein reduziertes äquivalentes LP, das die freie Variable x_i nicht mehr enthält. Eine Lösung des reduzierten Programms liefert dann eine Optimallösung des ursprünglichen Problems, wenn man sie durch eine zusätzliche Komponente x_i mit beliebigem Wert erweitert.

2. $A_i = 0$ und $c_i \neq 0$:

Falls das lineare Programm überhaupt eine zulässige Lösung x besitzt, ist auch jedes \bar{x} mit

$$\bar{x}_j := \begin{cases} x_j, & \text{für } j \neq i \\ x_j + \alpha, & \text{für } j = i \end{cases}$$

für beliebiges α eine zulässige Lösung. Wegen $c_i \neq 0$ kann die Zielfunktion

$$f(x) = c^T \bar{x} = c^T x + c_i \alpha$$

durch passende Wahl von α beliebig klein gemacht werden, so dass

$$-\infty = \inf c^T x \quad (4.7)$$

$$\begin{aligned} \text{s.t. } Ax &= b \\ x_i &\geq 0 \quad \text{für } i \in I. \end{aligned}$$

Andernfalls besitzt das lineare Programm keine zulässige Lösung. Es kann deshalb die Untersuchung dieses Falles abgebrochen werden, da das LP keine endliche Optimallösung besitzt.

3. $A_i \neq 0$:

Dann existiert in der i -ten Spalte von A ein Element $A_{ji} \neq 0$. Die Variable x_i kann deshalb durch die übrigen Variablen ausgedrückt werden, indem man die j -te Gleichung in $Ax = b$ nach x_i auflöst

$$x_i = (b_j - \sum_{k=1, k \neq i}^n A_{jk} x_k) / A_{ji}. \quad (4.8)$$

Anschließend ersetzt man in den übrigen Gleichungen von $Ax = b$ und $c^T x$ die Variable x_i durch die rechte Seite von (4.8). Wie bei der Gauß-Elimination kann man dies durch Umformen der Matrix A , der rechten Seite von b und von c , d.h. durch den Übergang von A_{rk} , b_r und c_k zu

$$\begin{aligned} b_r &\longrightarrow b_r - \frac{A_{ri}}{A_{ji}} b_j \\ A_{rk} &\longrightarrow A_{rk} - \frac{A_{ri} A_{jk}}{A_{ji}} \\ c_k &\longrightarrow c_k - \frac{A_{jk}}{A_{ji}} c_i \end{aligned}$$

für $r \neq j$ und $k \neq i$ erreichen. Man erhält also ein reduziertes äquivalentes lineares Programm, das die Variable x_i nicht mehr erhält. Jede Optimallösung des reduzierten linearen Programms kann mittels (4.8) zu einer Optimallösung des unreduzierten Programms erweitert werden.

Auf diese Weise lassen sich im Prinzip alle freien Variablen der Reihe nach eliminieren.

4.1.3 Geometrische Grundlagen

Es wird zunächst die Form der zulässigen Menge eines linearen Programms genauer betrachtet.

Definition 4.1. (Polyeder)

Der Durchschnitt von endlich vielen Halbräumen $\{x \in \mathbb{R}^n \mid a_j^T x \leq b_j\}$ mit festen Vektoren a_j und reellen Zahlen b_j wird Polyeder genannt.

Offenbar ist ein Polyeder nicht immer beschränkt. Jede Hyperebene $\{x \mid a_j^T x = b_j\}$ lässt sich als Schnitt zweier Halbräume $\{x \mid a_j^T x \leq b_j\}$ und $\{x \mid -a_j^T x \leq -b_j\}$ darstellen. Die Menge \mathcal{P} aus (4.6) ist also ein Polyeder, das durch Gleichungen und besonders einfache Ungleichungen beschrieben wird. Solche speziellen Polyeder werden im Folgenden eine wichtige Rolle spielen.

Definition 4.2. (Konvex)

Eine Menge $\mathcal{K} \subset \mathbb{R}^n$ heißt konvex, falls für alle $x, y \in \mathcal{K}$ und $\lambda \in [0, 1]$ stets $\lambda x + (1 - \lambda)y \in \mathcal{K}$ folgt. Eine Funktion $f : \mathcal{K} \rightarrow \mathbb{R}$ heißt konvex, falls \mathcal{K} nicht leer und konvex ist und falls für alle $x, y \in \mathcal{K}$ und $\lambda \in [0, 1]$ stets $f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$ folgt. Eine Funktion $f : \mathcal{K} \rightarrow \mathbb{R}$ heißt streng konvex, falls \mathcal{K} konvex ist und falls für alle $x, y \in \mathcal{K}$ und $\lambda \in (0, 1)$ stets $f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y)$ folgt.

Anschaulich gesprochen muss bei einer konvexen Menge \mathcal{K} mit x und y auch stets die Verbindungsstrecke zwischen x und y in \mathcal{K} liegen. Für jedes $x, y \in \mathcal{K}$ verläuft der Graph einer konvexen Funktion $f : \mathcal{K} \rightarrow \mathbb{R}$ entlang der Verbindungsstrecke zwischen x und y unterhalb oder auf der Sekante die durch $(x, f(x))$ und $(y, f(y))$ geht.

Bemerkung 4.3. Halbräume sind konvexe Mengen und lineare Funktionen sind konvex, aber nicht streng konvex. Falls \mathcal{K} konvex ist, folgt aus $x_i \in \mathcal{K}$, $\lambda_i \geq 0$ für $1 \leq i \leq m$ und $\sum_{i=1}^m \lambda_i = 1$ stets

$$\sum_{i=1}^m \lambda_i x_i \in \mathcal{K}.$$

Diese nützliche Beziehung lässt sich mit Induktion nach m leicht zeigen.

Im Folgenden wird mit $\text{aff}(\mathcal{K})$ die affine Hülle einer konvexen Menge \mathcal{K} bezeichnet, d.h. die kleinste affine Mannigfaltigkeit, die \mathcal{K} enthält. Ihre Dimension definiert die Dimension von \mathcal{K} , $\dim \mathcal{K} = \dim \text{aff}(\mathcal{K})$.

Der Durchschnitt konvexer Mengen ist wieder konvex:

Lemma 4.4. Seien \mathcal{K}_i konvexe Mengen für alle i aus einer beliebigen Indexmenge I . Dann ist auch $\mathcal{K} := \bigcap_{i \in I} \mathcal{K}_i$ konvex.

Der Beweis ist trivial. Als Korollar erhält man daraus, dass jedes Polyeder konvex ist, weil jeder Halbraum konvex ist. Die zulässige Menge \mathcal{P} aus (4.6) ist also ein konvexes Polyeder. Im \mathbb{R}^1 ist ein Polyeder ein abgeschlossenes Intervall und auch im \mathbb{R}^2 oder \mathbb{R}^3 kann man sich ein Polyeder leicht vorstellen. In den Abbildungen 4.1 und 4.2 sind Polyeder der Form (4.6) dargestellt.

In Abbildung 4.1 sieht man links die Hyperebene $a^T x = 1$ mit $a = (1, 1, 1)$ geschnitten mit dem positiven Orthanten des \mathbb{R}^3 als schraffierte Fläche. Der zulässige Bereich ist Polyeder \mathcal{P} der Dimension 2. Legt man die Zeichenebene in die affine Mannigfaltigkeit $a^T x = 1$, so hat \mathcal{P} die Form wie auf der rechten Seite der Abbildung. Die Ecken entsprechen dabei den Punkten in der Hyperebene, die durch $x_1 = x_2 = 0$ sowie $x_1 = x_3 = 0$ und $x_2 = x_3 = 0$ gegeben sind, die Kanten erfüllen $x_1 = 0$ oder $x_2 = 0$ oder $x_3 = 0$.

Für $b = (4, 1)^T$ und $A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ ist der zulässige Bereich \mathcal{P} eindimensional wie in Abbildung 4.2 skizziert. Obwohl die Form von \mathcal{P} sehr speziell aussieht (es treten nur Halbräume der Form $x \geq 0$ und Hyperebenen auf), lässt sich jedes beliebige beschränkte Polyeder in der Form $\{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$ darstellen. Dies soll hier jedoch nicht bewiesen werden.

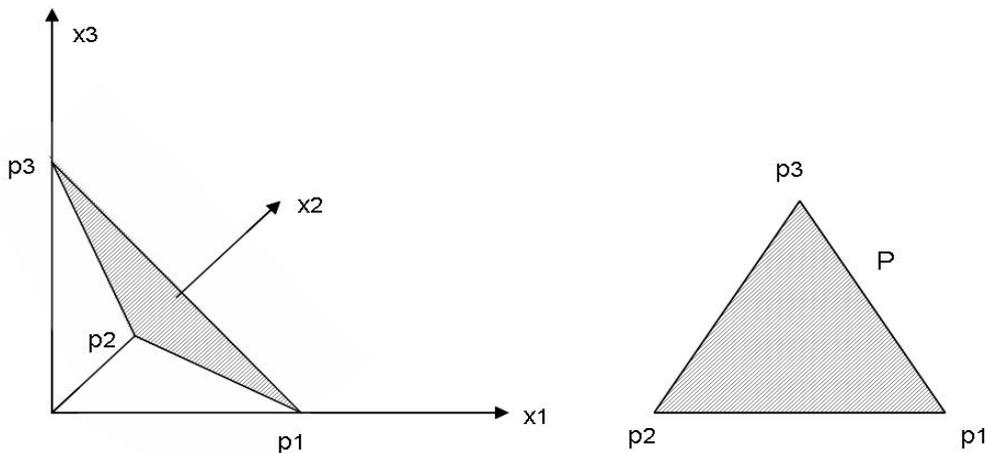


Abbildung 4.1: Darstellung eines Polyeders, eine lineare Gleichung

Es sollen nun noch die Ecken eines Polyeders charakterisiert werden. Dazu werden zunächst Extrempunkte allgemeiner konvexer Mengen definiert.

Definition 4.5. (Extrempunkt)

Ein Punkt $a \in \mathcal{K}$ heißt Extrempunkt einer konvexen Menge \mathcal{K} , falls $x, y \in \mathcal{K}$ und $\lambda \in (0, 1)$ aus $a = \lambda x + (1 - \lambda)y$ stets $a = x = y$ folgt.

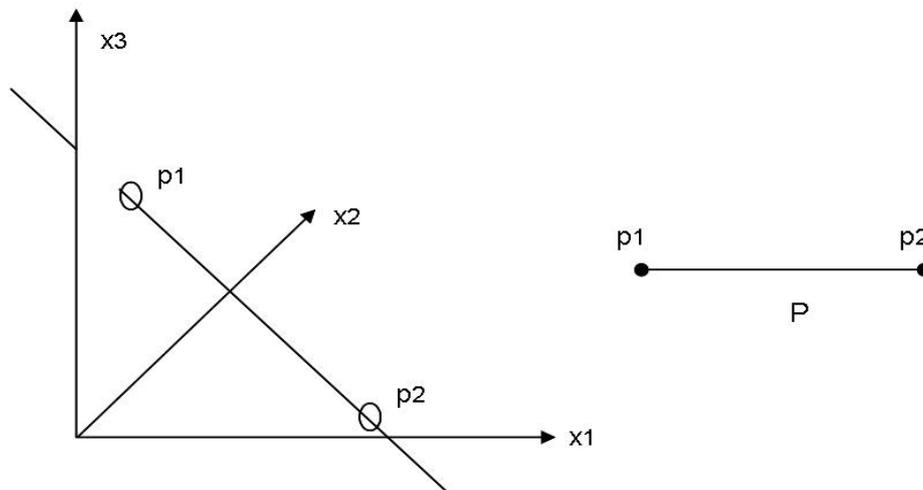


Abbildung 4.2: Darstellung eines Polyeders, zwei lineare Gleichungen

Anschaulich besagt die Definition, dass sich a nicht als echte Konvexkombination (mit $\lambda \neq 1$ und $\lambda \neq 0$) verschiedener Punkte x und y aus \mathcal{K} darstellen lässt. Man kann anhand von Beispielen von Polyedern im \mathbb{R}^2 oder \mathbb{R}^3 sehen, dass ihre Extrempunkte gerade den „Ecken“ des Polyeders entsprechen. Es sollen deshalb die Ecken eines Polyeders \mathcal{K} als seine Extrempunkte definiert werden. Eine andere intuitive Definition der Ecken eines Polyeders \mathcal{K} der Dimension $\dim \mathcal{K} = m$ ist, dass sie 0-dimensionale Durchschnitte von $m - 1$ -dimensionalen Seitenflächen von \mathcal{K} sind. Dies trifft bei Polyedern gerade auf Extrempunkte zu. Um das zu zeigen, muss jedoch zuerst der Begriff „Seitenfläche eines Polyeders“ geklärt werden. Zunächst sollen Extrepmengen konvexer Mengen definiert werden.

Definition 4.6. (Extrepmenge)

Eine konvexe Teilmenge $\varepsilon \subset \mathcal{K}$ einer konvexen Menge \mathcal{K} heißt Extrepmenge von \mathcal{K} , falls aus $a \in \varepsilon$, $x, y \in \mathcal{K}$, $\lambda \in (0, 1)$ und $a = \lambda x + (1 - \lambda)y$ stets $x, y \in \varepsilon$ folgt.

Diese Definition besagt anschaulich, dass sich kein $a \in \varepsilon$ als echte Konvexkombination verschiedener Punkte x, y aus $\mathcal{K} \setminus \varepsilon$ darstellen lässt. Extrempunkte des Polyeders \mathcal{K} sind gerade seine 0-dimensionalen Extrepmengen. Außerdem ist \mathcal{K} selbst Extrepmenge von \mathcal{K} maximaler Dimension. Extrepmengen von Polyedern haben auch wieder eine anschauliche Bedeutung: außer \mathcal{K} sind sie gerade seine Ecken, Kanten und allgemein seine Seitenflächen der verschiedenen Dimensionen. Man definiert deshalb die Seitenflächen von Polyedern als ihre Extrepmengen.

Es gilt nun folgender Sachverhalt: Jede Extremalmenge ε einer beliebigen konvexen Menge \mathcal{K} ist der Durchschnitt von \mathcal{K} mit der affinen Hülle von ε

$$\varepsilon = \mathcal{K} \cap \text{aff}(\varepsilon). \quad (4.9)$$

Eine Extremalmenge ist also bereits durch ihre affine Hülle $\text{aff}(\varepsilon)$ bestimmt. Eine echte Teilmenge ε_1 einer Extremalmenge ε mit $\text{aff}(\varepsilon_1) = \text{aff}(\varepsilon)$ kann deshalb keine Extremalmenge sein. Ein Beweis von (4.9) kann in [16] gefunden werden. Jetzt soll sich wieder den allgemeinen linearen Programmen gewidmet werden. Die Resultate können auf das Problem (4.2) und die Menge ε seiner Optimallösungen angewendet werden:

Satz 4.7. *Wenn das lineare Programm (4.2) Optimallösungen besitzt, ist die Menge ε der Optimallösungen eine Extremalmenge (also Seitenfläche) des Polyeders der zulässigen Lösungen von (4.2).*

Beweis: Das allgemeine lineare Programm (4.2) besitzt die Form

$$\inf \{c^T x \mid x \in S\},$$

wobei S das Polyeder der zulässigen Lösungen von (4.2) ist. Falls (4.2) Optimallösungen besitzt, existiert

$$\alpha := \min \{c^T x \mid x \in S\} \in \mathbb{R}$$

und es ist

$$\varepsilon = \{x \in S \mid c^T x = \alpha\} \subset S$$

ein Polyeder in S , also eine konvexe Teilmenge von S . Es wird nun angenommen, dass ε keine Extremalmenge ist. Sei dazu $a \in \varepsilon$ und $a = \lambda x + (1 - \lambda)y$ mit $0 < \lambda < 1$ und $x, y \in S$ sowie o.B.d.A. $x \notin \varepsilon$. Es folgt wegen $c^T x > \alpha$, $c^T y \geq \alpha$ sofort der Widerspruch

$$\alpha = c^T a = \lambda c^T x + (1 - \lambda)c^T y > \lambda \alpha + (1 - \lambda)\alpha = \alpha.$$

Also ist ε Extremalmenge von S .

□

Jede konvexe Menge \mathcal{K} besitzt zumindest \mathcal{K} und die leere Menge als „triviale“ Extremalmengen. Die Existenz weiterer Extremalmengen, insbesondere von Extrempunkten, ist selbst für Polyeder nicht gesichert. Anders ist die Situation bei Polyedern, die keine Gerade enthalten. Dies trifft wegen $\mathcal{P} \subset \{x \in \mathbb{R}^n \mid x \geq 0\}$ auf das Polyeder aus \mathcal{P} (4.6) zu. Solche Polyeder besitzen stets Extrempunkte. Aus dem Satz folgt daher:

Korollar 4.8. *Wenn das Standardprogramm (4.5) Optimallösungen besitzt, dann gibt es unter ihnen auch Extrempunkte von \mathcal{P} .*

4.2 Das Simplexverfahren

Die Idee des Simplexverfahrens zur Lösung linearer Programme in Standardform ist, die Eckpunkte des zulässigen Polyeders \mathcal{P} in einer geeigneten Weise nach der optimalen Ecke abzusuchen. Dies ist wegen des Korrolars 4.8. gerechtfertigt. Das Verfahren lässt sich folgendermaßen beschreiben:

1. Finde eine Ecke von \mathcal{P}
2. Gehe entlang einer absteigenden Kante (d.h. entlang welcher $c^T x$ kleiner wird) zu einer benachbarten Ecke
3. Wiederhole Schritt 2. solange bis es keine absteigende Kante mehr gibt.

Die Simplexmethode soll im Folgenden anhand eines speziellen Formates, dem sogenannten Simplexformat, beschrieben werden. Die Herleitung soll dabei relativ knapp gehalten werden, für eine genaue Behandlung der Simplex-Methode siehe z.B. [16] oder [11].

4.2.1 Das spezielle Simplexformat

Das Simplexformat erlaubt eine besonders einfache Darstellung der Simplexmethode. Grundsätzlich ist dieses Verfahren jedoch nicht an ein spezielles Format des lineares Programms gebunden. Im Folgenden bezeichnet $A = (a_1, \dots, a_n)$ eine $m \times n$ -Matrix mit Spalten a_i , $N = 1, \dots, n$, und $J = (j_1, \dots, j_k)$ einen Indexvektor der Länge $|J| = k$ bestehend aus paarweise verschiedenen Indizes j_i mit $j_i \in N$ für $1 \leq i \leq k$. Dann bezeichnet

$$A_J := [a_{j_1}, \dots, a_{j_k}]$$

die $m \times k$ -Matrix, die aus den Spalten von A besteht und deren Indizes zu J gehören. Außerdem heißen die Indexvektoren J und K komplementär, wenn $|J| + |K| = |N| = n$ und jeder Index entweder in J oder in K vorkommt. Man schreibt dann auch kurz $J \oplus K = N$. Ebenso bezeichnet

$$x_J := (x_{j_1}, \dots, x_{j_k})^T$$

den Teilvektor eines Vektors $x \in \mathbb{R}^n$. Für $J \oplus K = N$ gilt dann

$$Ax = A_J x_J + A_K x_K.$$

Ein Indexvektor J heißt Basis von A , falls $|J| = m$ und A_J regulär ist. In diesem Fall heißen die Variablen x_{j_k} , $k = 1, \dots, m$ Basisvariable.

Es soll zunächst ein Programm in Standardform

$$\min \{c^T x \mid Ax = b, x \geq 0\} \tag{4.10}$$

mit der Menge der zulässigen Lösungen

$$\mathcal{P} := \{x \mid Ab = b, x \geq 0\}$$

betrachtet werden. Hierbei sei $A \in \mathbb{R}^{m \times n}$ und die Vektoren x , b und c entsprechend dimensioniert. Außerdem seien die Zeilen der Matrix A linear unabhängig. Dieses Programm (4.10) soll nun in die Simplexform überführt werden. Dies geschieht durch Einführen einer weiteren Variablen $z \in \mathbb{R}$ und einer weiteren linearen Gleichung $c^T x + z = 0$. Mit dieser Gleichung ist die Minimierung von $c^T x$ auf \mathcal{P} äquivalent zur Maximierung von z , d.h. das lineare Programm (4.10) ist äquivalent zu

$$\begin{aligned} & \min_{x,z} z \\ \text{s.t.} \quad & \begin{pmatrix} A & 0 \\ c^T & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ z \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}, \quad x \geq 0. \end{aligned}$$

Definiert man die Matrix \hat{A} und die Vektoren \hat{b} und \hat{x} durch

$$\hat{A} = \begin{pmatrix} A & 0 \\ c^T & 1 \end{pmatrix}, \quad \hat{b} = \begin{pmatrix} b \\ 0 \end{pmatrix}, \quad \hat{x} = \begin{pmatrix} x \\ z \end{pmatrix},$$

erhält man das zu (4.10) äquivalente lineare Programm

$$\max \left\{ z \mid \hat{A}\hat{x} = \hat{A} \begin{pmatrix} x \\ z \end{pmatrix} = \hat{b}, x \geq 0 \right\} \quad (4.11)$$

mit der zulässigen Menge

$$\hat{\mathcal{P}} = \left\{ \hat{x} = \begin{pmatrix} x \\ z \end{pmatrix} \mid \hat{A}\hat{x} = \hat{b}, x \geq 0 \right\}.$$

In diesem linearen Programm ist z eine freie Variable, sie unterliegt keiner Vorzeichenbeschränkung.

Mit A besitzt auch die erweiterte Matrix \hat{A} linear unabhängige Zeilen. Wenn $J = (x_{j_1}, \dots, x_{j_m})$ eine Basis von A ist, ist insbesondere der erweiterte Indexvektor $\hat{J} = (x_{j_1}, \dots, x_{j_m}, z) =: J \oplus \{z\}$ eine Basis von \hat{A} . Umgekehrt gilt: Ist $\hat{J} = J \oplus \{z\}$ eine Basis von \hat{A} , dann ist $J = \hat{J} \setminus \{z\}$ eine Basis von A . Es gibt eine einfache Beziehung zwischen den Basislösungen $\hat{x}(\hat{J})$ von $\hat{A}\hat{x} = \hat{b}$ und $x(J)$ von $Ax = b$:

$$\hat{x}(\hat{J}) = \begin{pmatrix} x(J) \\ -c^T x(J) \end{pmatrix},$$

so dass J zulässige Basis von $Ax = b$ genau dann ist, wenn $\hat{J} = J \oplus \{z\}$ eine zulässige Basis von (4.11) ist. Da \hat{J} eindeutig durch J bestimmt ist, sagt man auch, dass J eine Basis von \hat{A} ist, wenn J eine Basis von A ist. Dementsprechend soll statt $\hat{x}(\hat{J})$ auch einfacher

$$\hat{x}(J) = \begin{pmatrix} x(J) \\ z(J) \end{pmatrix}$$

geschrieben werden. Die Ecken (d.h. die Extrempunkte) der zulässigen Polyeder von (4.10) bzw. von (4.11) hängen nun eng mit Basislösungen zusammen. Das verdeutlicht folgender Satz.

Satz 4.9.

- (i) Der Vektor \bar{x} ist Ecke von $\mathcal{P} := \{x \mid Ax = b, x \geq 0\}$ genau dann, wenn der erweiterte Vektor

$$\hat{x} := \begin{pmatrix} \bar{x} \\ -c^T \bar{x} \end{pmatrix}$$

Ecke von

$$\hat{\mathcal{P}} = \left\{ \hat{x} = \begin{pmatrix} x \\ z \end{pmatrix} \mid \hat{A}\hat{x} = \hat{b}, x \geq 0 \right\}$$

ist.

- (ii) Zu jeder Ecke \bar{x} von \mathcal{P} gibt es eine zulässige Basis J , so dass $\bar{x} = x(J)$ Basislösung von $Ax = b$ zur Basis J ist und umgekehrt: Jede zulässige Basislösung $x(J)$ zu einer Basis J von $Ax = b$ ist eine Ecke von \mathcal{P} . Wegen (i) ist dann der erweiterte Vektor $\hat{\bar{x}}$ Basislösung von $\hat{A}\hat{x} = \hat{b}$ zur zulässigen erweiterten Basis $\hat{J} = J \oplus \{z\}$.

Ein Beweis dieses Satzes findet sich in [16].

Bemerkung 4.10. Die Zuordnung „Ecke–zulässige Basis“ ist nicht eindeutig, eine Ecke kann Basislösung verschiedener Basen sein. Diese Beobachtung veranlasst zu folgender Definition: Eine zulässige Basis J von (4.10) heißt nichtentartet, falls für die Basislösung $\bar{x} = x(J)$ gilt: $\bar{x}_J > 0$. Das Problem (4.10) heißt nichtentartet, falls alle zulässigen Basen nichtentartet sind, andernfalls heißt (4.10) entartet. Es kann also sein, dass eine Ecke, die zu einer nichtentarteten Basis gehört, Basislösung nur einer zulässigen Basis ist. Aus der Entartung einer zulässigen Basis folgt aber nicht unbedingt, dass zu der entsprechenden Ecke mehr als eine Basis gehört.

Satz 4.9. besagt nun, dass man alle Ecken eines Polyeders durch zugehörige zulässige Basen beschreiben kann. Dabei hängt die Ecke stets eindeutig von der Basis ab, aber zu einer Ecke gehören eventuell mehrere zulässige Basen, falls eine dieser Basen entartet ist. Die Möglichkeit der Entartung veranlasst zu einer kleinen Änderung der eingangs beschriebenen Formulierung der Simplexmethode. Genau genommen läuft das Simplexverfahren nicht von einer Ecke zu einer benachbarten Ecke, sondern von einer zulässigen Basis zu einer (davon verschiedenen) „benachbarten“ zulässigen Basis. Falls das lineare Programm (4.11) nicht entartet ist, entspricht dies gerade tatsächlich dem Wechsel zu einer anderen (benachbarten) Ecke.

4.2.2 Durchführung der Simplex-Methode

Zur besseren Verständlichkeit soll zunächst noch einige Notation eingeführt werden. Für die Darstellung der Simplex-Methode wird ein lineares Programm in Simplexform (4.11) mit einer Matrix $A \in \mathbb{R}^{m \times n}$ und einem Vektor $x \in \mathbb{R}^n$ betrachtet. Da man z als $(n+1)$ -te Komponente von \hat{x} auffassen kann, ist es möglich, das Programm (4.11) auch in der folgenden Form zu schreiben:

$$\max \{ \hat{x}_{n+1} \mid \hat{A}\hat{x} = \hat{b}, x_i \geq 0 \text{ für } i \in N \}$$

mit $N = \{1, \dots, n\}$. Dementsprechend können eine Basis J von A und die erweiterte Basis $\hat{J} := J \oplus \{z\}$ mittels der Basisvariablen und der Basisindizes dargestellt werden.

$$\begin{aligned} J &= (x_{i_1}, \dots, x_{i_m}) = (i_1, \dots, i_m), & i_1, \dots, i_m \in N \\ \hat{J} &= (x_{i_1}, \dots, x_{i_m}, z) = (i_1, \dots, i_m, n+1). \end{aligned}$$

Zu einer Basis J von A und dem Gleichungssystem $Ax = b$ gehört nun das Tableau

$$(J, (\bar{A} \ \bar{b})) \quad \text{mit} \quad (\bar{A} \ \bar{b}) = A_J^{-1}(A \ b).$$

Das Tableau, das zu der erweiterten Basis $\hat{J} = J \oplus \{z\}$ von \hat{A} gehört, hängt damit eng zusammen. Zunächst gehört zu \hat{J} die folgende Basismatrix von \hat{A} :

$$\hat{A}_{\hat{J}} = \begin{pmatrix} A_J & 0 \\ c_J^T & 1 \end{pmatrix}.$$

Ihre Inverse besitzt die Form

$$\hat{A}_{\hat{J}}^{-1} = \begin{pmatrix} A_J^{-1} & 0 \\ -\pi & 1 \end{pmatrix}$$

mit einem Zeilenvektor π , der wegen

$$\hat{A}_{\hat{J}}^{-1} \hat{A}_{\hat{J}} = \begin{pmatrix} A_J^{-1} & 0 \\ -\pi & 1 \end{pmatrix} \begin{pmatrix} A_J & 0 \\ c_J^T & 1 \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & 1 \end{pmatrix}$$

die Lösung der linearen Gleichung

$$\pi A_J = c_J^T, \quad \text{d.h. } \pi = c_J^T A_J^{-1}$$

ist. Das Tableau zur erweiterten Basis $\hat{J} = J \oplus \{z\}$ und dem Gleichungssystem $\hat{A}\hat{x} = \hat{b}$ kann nun wie folgt angegeben werden

$$\left(J \oplus \{z\}, \begin{pmatrix} \bar{A} & 0 & \bar{b} \\ \bar{c}^T & 1 & \beta \end{pmatrix} \right),$$

wobei

$$\begin{pmatrix} \bar{A} & 0 & \bar{b} \\ \bar{c}^T & 1 & \beta \end{pmatrix} = \begin{pmatrix} A_J^{-1} & 0 \\ -\pi & 1 \end{pmatrix} \cdot \begin{pmatrix} A & 0 & b \\ c^T & 1 & 0 \end{pmatrix},$$

also

$$\begin{aligned} \bar{A} &= A_J^{-1} A, & \bar{b} &= A_J^{-1} b \\ \bar{c}^T &= -\pi A + c^T, & \beta &= -\pi b. \end{aligned}$$

Definition 4.11. (Benachbarte Basen)

Zwei Basen J und J' von A heißen benachbart, falls sie sich durch genau einen Index aus N unterscheiden, d.h. es gibt Indizes s und q aus N , so dass $q \in J$, $s \notin J$ und $J' = (J \cup \{s\}) \setminus \{q\}$. Die erweiterten Basen $\hat{J} = J \oplus \{z\}$ und $\hat{J}' = J' \oplus \{z\}$ heißen benachbart, wenn J und J' benachbart sind.

Zum Beispiel sind $J = (i_1, \dots, i_{r-1}, i_r, i_{r+1}, \dots, i_m)$ und $J' = (i_1, \dots, i_{r-1}, s, i_{r+1}, \dots, i_m)$ mit $q = i_r$ und $s \notin J$ benachbart. Es gilt jetzt folgender Satz.

Satz 4.12. *Es sei $J = (i_1, \dots, i_m)$ eine Basis von $Ax = b$ und $s \notin J$. Sei ferner*

$$\left(\hat{J}, \begin{pmatrix} \bar{A} & 0 & \bar{b} \\ \bar{c}^T & 1 & \beta \end{pmatrix} \right)$$

das Tableau zur erweiterten Basis $\hat{J} = J \oplus \{z\} = (i_1, \dots, i_m, n+1)$ des zu \hat{A} gehörigen linearen Gleichungssystems

$$\begin{pmatrix} A & 0 \\ c^T & 1 \end{pmatrix} \begin{pmatrix} x \\ z \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}.$$

Ferner sei

$$\begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_m \\ \alpha_{m+1} \end{pmatrix} := \begin{pmatrix} \bar{A}_s \\ \bar{c}_s \end{pmatrix}$$

die s -te Spalte der Matrix $\begin{pmatrix} \bar{A} \\ \bar{c} \end{pmatrix}$. Dann ist

$$\hat{J}' = (i_1, \dots, i_{r-1}, s, i_{r+1}, \dots, i_m, n+1) = J' \oplus \{z\},$$

wobei $J' = (i_1, \dots, i_{r-1}, s, i_{r+1}, \dots, i_m)$, genau dann eine Nachbarbasis von $\hat{J} = J \oplus \{z\}$, wenn $\alpha_r \neq 0$. In diesem Fall ist das zu $\hat{J}' = J' \oplus \{z\}$ gehörige Tableau

$$\left(\hat{J}', \begin{pmatrix} \bar{A}' & 0 & \bar{b}' \\ (\bar{c}')^T & 1 & \beta' \end{pmatrix} \right)$$

gegeben durch

$$\begin{pmatrix} \bar{A}' & 0 & \bar{b}' \\ (\bar{c}')^T & 1 & \beta' \end{pmatrix} = F \begin{pmatrix} \bar{A} & 0 & \bar{b} \\ \bar{c}^T & 1 & \beta \end{pmatrix},$$

wobei F die $m + 1$ -reihige Matrix

$$F = \left(\begin{array}{cccc|c} 1 & & -\alpha_1/\alpha_r & & 0 \\ & \ddots & \vdots & & \\ & & 1 & -\alpha_{r-1}/\alpha_r & \\ & & & 1/\alpha_r & \\ & & -\alpha_{r+1}/\alpha_r & 1 & \\ & & \vdots & & \ddots \\ -\alpha_m/\alpha_r & & & & 1 \\ \hline -\alpha_{m+1}/\alpha_r & & & & 1 \end{array} \right)$$

mit der folgenden Inversen

$$G := F^{-1} = \left(\begin{array}{cccc|c} 1 & & \alpha_1 & & 0 \\ & \ddots & \vdots & & \\ & & 1 & \alpha_{r-1} & \\ & & & \alpha_r & \\ & & \alpha_{r+1} & 1 & \\ & & \vdots & & \ddots \\ \alpha_m & & & & 1 \\ \hline \alpha_{m+1} & & & & 1 \end{array} \right) = \left(\begin{array}{c|c} \bar{A}_{J'} & 0 \\ \hline \bar{c}_{J'}^T & 1 \end{array} \right).$$

Beweis: siehe [16]

4.2.3 Abbruchkriterien

Im Folgenden wird nun das zum Standardprogramm

$$\min \{c^T x \mid Ax = b, x \geq 0\} \quad (4.12)$$

gehörige erweiterte Programm in Simplexform

$$\min \left\{ z \mid \begin{pmatrix} A & 0 \\ c^T & 1 \end{pmatrix} \begin{pmatrix} x \\ z \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}, x \geq 0 \right\} \quad (4.13)$$

betrachtet. Sei J eine zulässige Basis von (4.12) und K der komplementäre Indexvektor der Nichtbasisvariablen. Zur zugehörigen zulässigen erweiterten Basis \hat{J} von (4.13) gehöre das Tableau

$$\left(\hat{J}, \begin{pmatrix} \bar{A} & 0 & \bar{b} \\ \bar{c}^T & 1 & \beta \end{pmatrix} \right) \quad \text{mit} \quad \begin{pmatrix} \bar{A}_J & 0 \\ \bar{c}_J^T & 1 \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & 1 \end{pmatrix}.$$

Die linearen Gleichungen des Tableaus reduzieren sich also auf

$$x_J + \bar{A}_K x_K = \bar{b}, \quad (4.14)$$

$$z + \bar{c}_K^T x_K = \beta. \quad (4.15)$$

Für die Basislösung $\bar{x} = x(J)$, $\bar{z} = z(J)$ gilt also wegen $\bar{x}_K = 0$

$$\bar{x}_J = \bar{b}, \quad \bar{x}_K = 0, \quad \bar{z} = \beta.$$

Entsprechend der Vorzeichen von \bar{c}_K^T unterscheidet man nun zwei Fälle:

1. $\bar{c}_k \geq 0$ für alle $k \in K$:

In diesem Fall ist die Basislösung $\bar{x} = x(J)$, $\bar{z} = z(J)$ eine Optimallösung von (4.13).

Beweis: Auch wenn die Form der Gleichungen (4.14) und (4.15) von der Wahl von K abhängt, sind sie trotzdem für jedes K äquivalent zu „ $Ax = b$ “ und „ $c^T x + z = 0$ “. Insbesondere müssen die Gleichungen also für alle (x, z) erfüllt sein, die für (4.13) zulässig sind, so dass für eine beliebige zulässige Lösung (x, z) von (4.13) wegen $x \geq 0$ und $\bar{c}_K \geq 0$ die Ungleichung

$$z = \beta - \bar{c}_K^T x_K \leq \beta$$

folgt, während für die zulässigen Basislösung (\bar{x}, \bar{z}) wegen $\bar{x}_K = 0$ die Gleichung $\bar{z} = \beta$ gilt.

□

2. Es gibt ein $s \in K$ mit $\bar{c}_s < 0$:

Sei s eine Komponente mit $\bar{c}_s < 0$. Weiterhin bezeichnet $\bar{a} := (\alpha_1, \dots, \alpha_m)^T = A_J^{-1} a_s$ die s -te Spalte von \bar{A} . Aus (4.14), (4.15) kann man nun ablesen, dass sich wegen $\bar{c}_s < 0$ der Wert $z = \beta - \sum_{k \in K} \bar{c}_k x_k$ der Zielfunktion von (4.13) vergrößert, wenn man x_s vergrößert, die Komponenten x_k , $k \in K \setminus \{s\}$, bei Null belässt und x_J so bestimmt, dass die Gleichungen (4.14), (4.15) erfüllt bleiben. Das heißt, es wird ein Strahl $(x(\theta), z(\theta))$, $\theta \geq 0$, bestimmt, so dass

$$x_s(\theta) = \theta, \quad x_{K \setminus \{s\}}(\theta) = 0, \quad x_J(\theta) = \bar{b} - \theta \bar{a}, \quad z(\theta) = \beta - \bar{c}_s \theta.$$

$(x(\theta), z(\theta))$ ist eine zulässige Lösung von (4.13), falls $x(\theta) \geq 0$, d.h. falls $\theta \geq 0$ und $x_J(\theta) = \bar{b} - \theta \bar{a} \geq 0$. Da $z(\theta)$ mit θ streng monoton wächst, versucht man ein maximales $\bar{\theta} \geq 0$ zu bestimmen, so dass $(x(\bar{\theta}), z(\bar{\theta}))$ gerade noch eine zulässige Lösung von (4.13) ist, d.h. ein maximales $\bar{\theta}$ mit $\bar{b} - \theta \bar{a} \geq 0$. Dabei werden wieder zwei Fälle unterschieden:

- (a) $\alpha_j \leq 0$ für alle j mit $1 \leq j \leq m$.
- (b) Es gibt ein r mit $\alpha_r > 0$ und $1 \leq r \leq m$.

Im Fall (a) ist $x(\theta)$ für alle $\theta \geq 0$ zulässig, da die Nichtnegativitätsbedingung für wachsende θ nie verletzt wird. Es gibt dann auch keine endliche Optimallösung, da die Zielfunktion $z(\theta)$ für wachsendes θ beliebig groß wird.

Im Fall (b) gibt es ein maximales $\bar{\theta}$, nämlich

$$\begin{aligned}\bar{\theta} &= \max \{ \theta \mid \bar{b}_j - \theta \alpha_j \geq 0 \text{ für alle } j = 1, \dots, m \} \\ &= \min_{1 \leq j \leq m} \left\{ \frac{\bar{b}_j}{\alpha_j} \mid \alpha_j > 0 \right\} < \infty.\end{aligned}$$

Man wählt dann ein r aus $\{1, \dots, m\}$ mit $\alpha_r > 0$, für welches das Minimum $\bar{b}_r/\alpha_r = \bar{\theta}$ angenommen wird. Die zulässige Lösung $x(\bar{\theta})$ erfüllt dann $x_s(\bar{\theta}) = 0$ und $x_s(\bar{\theta}) > 0$. Wegen $x_{K \setminus \{s\}}(\bar{\theta}) = 0$ ist $x(\bar{\theta})$ die Basislösung von

$$J' = (i_1, \dots, i_{r-1}, s, i_{r+1}, \dots, i_m), \quad K' = (K \cup \{i_r\}) \setminus \{s\},$$

d.h. man führt mit dem Wechsel $x(J) \rightarrow x(\bar{\theta})$ einen sog. Simplexschritt $J \rightarrow J' = (J \setminus \{i_r\}) \cup \{s\}$ durch. Wegen Satz 4.3. und $\alpha_r \neq 0$ ist J' eine Nachbarbasis von J .

Nach Konstruktion ist dann

- $J' \oplus \{z\}$ zulässige Nachbarbasis von $J \oplus \{z\}$ von (4.13) und
- $z(J') = z(J) - \bar{c}_s \bar{\theta} \geq z(J)$, mit Gleichheit nur dann, wenn die zu J gehörige Basislösung entartet ist, sonst ist $\bar{\theta} > 0$.

4.2.4 Zusammenfassung: Ein Simplexschritt

Ein Simplexschritt kann nun wie folgt zusammengefasst werden:

Algorithmus (Simplexschritt)

Start: Sei $J = (i_1, \dots, i_m)$ eine zulässige Basis von (4.10) und K der Indexvektor der Nichtbasisvariablen, $J \oplus K = N$, $N = \{1, \dots, n\}$. Sei ferner das zugehörige Tableau zur Basis $\hat{J} = J \oplus \{z\}$ von (4.10) gegeben:

$$\left(\hat{J}, \begin{pmatrix} \bar{A} & 0 & \bar{b} \\ \bar{c}^T & 1 & \beta \end{pmatrix} \right).$$

1. Setze $\bar{x} = x(J)$, $\bar{z} = z(J)$, d.h. $\bar{x}_J = \bar{b}$, $\bar{x}_K = 0$, $\bar{z} = \beta$.
2. Prüfe, ob $\bar{c}_k \geq 0$ für alle $k \in K$.

- (a) Falls ja, STOPP: (\bar{x}, \bar{z}) ist Optimallösung von (4.11).
- (b) Sonst wähle $s \in K$ mit $\bar{c}_s < 0$. (Pivotschritt)

3. Setze $\bar{a} = (\alpha_1, \dots, \alpha_m)^T = \bar{A}_s$, die s -te Spalte von \bar{A} .
4. Falls $\alpha_1 \leq 0, \dots, \alpha_m \leq 0$, STOPP: der Optimalwert von (4.11) ist unendlich groß.
5. Sonst wähle $r \in \{1, \dots, m\}$ mit $\alpha_r > 0$ und

$$\frac{\bar{b}_r}{\alpha_r} = \min_{1 \leq j \leq m} \left\{ \frac{\bar{b}_j}{\alpha_j} \mid \alpha_j > 0 \right\}.$$

6. Setze $J' = (i_1, \dots, i_{r-1}, s, i_{r+1}, \dots, i_m)$ und bestimme das neue Tableau zu $J' \oplus \{z\}$ von (4.10)

$$\begin{pmatrix} \bar{A}' & 0 & \bar{b}' \\ (\bar{c}')^T & 1 & \beta' \end{pmatrix} = F \begin{pmatrix} \bar{A} & 0 & \bar{b} \\ \bar{c}^T & 1 & \beta \end{pmatrix},$$

wobei F die Matrix aus Satz 4.12. ist.

Dann ist $J' \oplus \{z\}$ zulässige Nachbarbasis von $J \oplus \{z\}$ und

$$\left(J' \oplus \{z\}, \begin{pmatrix} \bar{A}' & 0 & \bar{b}' \\ (\bar{c}')^T & 1 & \beta' \end{pmatrix} \right)$$

das zugehörige Tableau von (4.11).

4.2.5 Der Befehl LINPROG in Matlab

In Matlab steht der Befehl `linprog` der Optimization-Toolbox zur Lösung linearer Programme der Form

$$\begin{aligned} \min_x \quad & f \cdot x \\ \text{s.t.} \quad & A \cdot x \leq b \end{aligned} \tag{4.16}$$

zur Verfügung. Dieser Befehl implementiert gerade den Simplex-Algorithmus. An einem einfachen Beispiel soll kurz gezeigt werden, wie man mithilfe von Matlab solche Probleme löst. Als Beispiel soll ein als Landwirtschaftsproblem bekanntes (frei erfundenes) Diätproblem

$$\begin{aligned} \max_{x_1, x_2} \quad & 143x_1 + 60x_2 \\ \text{s.t.} \quad & x_1 + x_2 \leq 75 \\ & 110x_1 + 30x_2 \leq 4.000 \\ & 120x_1 + 210x_2 \leq 15.000 \\ & x_1 \geq 0, \quad x_2 \geq 0 \end{aligned} \tag{4.17}$$

dienen. Diese Programm muss zunächst in ein LP der Form (4.16) gebracht werden.

$$\min_{x_1, x_2} -143x_1 - 60x_2 \quad (4.18)$$

s.t.

$$\underbrace{\begin{pmatrix} 1 & 1 \\ 110 & 30 \\ 120 & 210 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}}_{=A} \underbrace{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}_{=x} \leq \underbrace{\begin{pmatrix} 75 \\ 4.000 \\ 15.000 \\ 0 \\ 0 \end{pmatrix}}_{=b}$$

Die Anweisungen

```
f = [-143 -60];
A = [1 1; 110 30; 120 210; -1 0; 0 -1];
b = [75; 4000; 15000; 0; 0];
```

```
x = linprog(f,A,b);
x_1 = x(1,:);
x_2 = x(2,:);
max_Ertrag = -f*x
```

ergeben

```
x_1 =
    21.8750
```

```
x_2 =
    53.1250
```

```
max_Ertrag =
    6.3156e+003
```

Das bedeutet, dass (4.18) sein Minimum für $x_1 = 21.875$ und $x_2 = 53.125$ annimmt. Das Maximum in (4.17), welches in diesem Beispiel gerade dem maximalen Ertrag entspricht, ist damit ca. 6316.

Allgemein gilt: Mit der Funktion `linprog` können lineare Optimierungsaufgaben der Form

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f \cdot x \\ \text{s.t.} \quad & A \cdot x \leq b \\ & B \cdot x = d \\ & l \leq x \leq u \end{aligned}$$

gelöst werden. Falls der Aufruf

```
x = linprog(f,A,b,B,d,l,u)
fWert = f*x
```

erfolgreich war, findet man in der Variable x den berechneten Lösungsvektor und in der Variable $fWert$ den zugehörigen Wert der Zielfunktion $f(x) = f \cdot x$. Wie bei allen anderen Matlab-Funktionen entsprechenden Typs können Argumente weggelassen werden, wenn nicht alle Formen der Nebenbedingungen auftreten. Damit aber die Reihenfolge erhalten bleibt, müssen dafür gegebenenfalls eckige Klammern `[]` geschrieben werden. Ferner können Optionen gesetzt werden und weitere Ausgabeargumente angegeben werden (für weitere Hinweise siehe `doc linprog(help linprog)`).

Kapitel 5

Modellprädiktive Regelung mittels linearer Programmierung

In diesem und dem folgenden Kapitel soll nun der Einsatz von linearer Programmierung in der modellprädiktiven Regelung untersucht werden. Die Grundlage bilden dabei grundsätzlich zwei verschiedene Zustandsmodelle: je nach dem, ob der gesamte Zustandsvektor $x(t)$ zur Verfügung steht, werden im Folgenden lineare Kontrollsysteme ohne bzw. mit Ausgang betrachtet. Weiterhin wird sich ausschließlich auf zeitinvariante und zeitdiskrete Systeme beschränkt. Die Wahl einer zeitdiskreten Darstellung liegt vor allem darin begründet, dass sich diese für die Implementierung des Programms besser eignet.

Im ersten Abschnitt wird das der modellprädiktiven Regelung zugrunde liegende Optimierungsproblem für lineare Kontrollsysteme ohne Ausgang formuliert. Anschließend wird ein Ansatz vorgestellt, wie dieses in ein äquivalentes Optimierungsproblem umgeformt werden kann. Danach wird das lineare Programm für die in Kapitel 6 folgende Implementierung in Matrizenform umgeschrieben. Im folgenden Punkt soll die Durchführung des modellprädiktiven Regelungsansatzes veranschaulicht werden. Die Idee beruht dabei auf dem receding-horizon Konzept, welches hier verdeutlicht werden soll. In Abschnitt 5.3 werden die wesentlichen Unterschiede für lineare Kontrollsysteme mit Ausgang beschrieben. Ein in der Regelungstechnik häufig auftretendes Problem ist, dass sowohl die Eingänge als auch die Ausgänge Beschränkungen unterliegen, im Fall der Eingänge meist physikalischer Natur (z.B. beschränkte Energie), bei den Ausgängen in Form von Sicherheitsbarrieren oder Toleranzen. Deshalb soll im dritten Abschnitt die Berücksichtigung solcher Beschränkungen behandelt werden und die dafür notwendige Erweiterung des linearen Programms hergeleitet werden.

5.1 Lineare Kontrollsysteme ohne Ausgang

Im Folgenden werden lineare zeitinvariante und zeitdiskrete Kontrollsysteme der Form

$$x(t+1) = Ax(t) + Bu(t) \tag{5.1}$$

betrachtet. Die Variable $t \in \mathbb{Z}$ beschreibt dabei die Zeit, $x \in \mathbb{R}^n$ den Zustand und $u \in \mathbb{R}^m$ die Kontrolle (oder auch Eingang). Weiterhin seien $A \in \mathbb{R}^{n \times n}$ und $B \in \mathbb{R}^{n \times m}$ Matrizen.

Die Zielsetzung ist nun den Zustand $x(t)$ in einen gegebenen Arbeitspunkt zu regeln. Für die theoretische Behandlung wird dieser Referenzzustand $x_{ref}(t)$ in den meisten Fällen gleich Null gesetzt (d.h. der Zustand $x(t)$ soll in den Ursprung geregelt werden, $x_{ref}(t) = 0$ für alle $t \in \mathbb{Z}$). Diese Annahme kann o.B.d.A. gemacht werden, da dies für jedes Kontrollsystem mittels einer Koordinatentransformation erzielt werden kann. Zur Lösung dieses Stabilisierungsproblems soll hier das Konzept der modellprädiktiven Regelung genutzt werden und das in MPC auftretende Optimierungsproblem mittels linearer Programmierung gelöst werden. Diese Vorgehensweise hat vor allem den Vorteil, dass eine Vielzahl von effizienten Algorithmen zur Lösung linearer Programme sowie umfangreiche theoretische Ergebnisse existieren. Außerdem ist der Rechenaufwand im Allgemeinen geringer als bei der Lösung quadratischer Programme.

5.1.1 Das Optimierungsproblem

Das Ziel der modellprädiktiven Regelung ist eine optimale Kontrollfolge zu berechnen, durch dessen Anwendung der Zustand $x(t)$ auf dem Prädiktionshorizont dem Referenzzustand $x_{ref}(t)$ folgen soll. Im ersten Schritt des modellprädiktiven Regelungsansatzes wird dafür ein Optimierungsproblem formuliert, durch dessen Lösung man zunächst die optimale Kontrollfolge u auf dem Kontrollhorizont erhält. Die Art und Weise wie das System die gewünschte Referenztrajektorie erreicht, hängt dabei von der Wahl der Zielfunktion ab. Obwohl quadratische Programme sehr effizient sind, kann das Problem mit der deutlich besseren Methode der linearen Programmierung gelöst werden. Dafür ist es sinnvoll, die Zielfunktion mittels einer 1-Norm zu formulieren. Es werden also im Gegensatz zu einer quadratischen Zielfunktion die absoluten Werte der Kontrollen und der Abweichung zwischen $x(t)$ und $x_{ref}(t)$ berücksichtigt. Es wäre ebenfalls möglich, eine ∞ - ∞ Norm zu verwenden. Ein Grund für die Wahl einer 1-Norm ist unter anderem, dass die ∞ - ∞ Norm nur die maximale Abweichung und nicht das gesamte Verhalten explizit in Betracht zieht.

Ausgehend von diesen Annahmen kann nun folgendes Optimierungsproblem formuliert werden.

$$\min_{u := [u^T(N_3) \dots u^T(N_4)]^T} \left\{ J(u, x(t)) = \sum_{t=N_1}^{N_2} Q \|x(t) - x_{ref}(t)\|_1 + \sum_{t=N_3}^{N_4} R \|u(t)\|_1 \right\} \quad (5.2)$$

s.t.

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), & t &= N_1 - 1, \dots, N_2 - 1 \\ u(t) &= 0, & N_4 + 1 &\leq t \leq N_2 - 1 \end{aligned}$$

Dabei bezeichnet

t_{akt} den aktuellen Zeitpunkt,

t_s den Zeitpunkt, an dem die Regelung gestartet wird – in den meisten Fällen $t_s = t_{akt} = 0$,

N_x die Länge des Prädiktionshorizontes, auf welchem der Zustand x dem Referenzzustand $x_{ref} \in \mathbb{R}^n$ folgen soll (N_1 untere, N_2 obere Grenze) und

N_u die Länge des Kontrollhorizontes (N_3 untere, N_4 obere Schranke).

Außerdem gilt

$$N_1 = (t_s + 1) \text{ und } N_2 = (N_1 + N_x - 1) = (t_s + N_x),$$

$$N_3 = t_s \text{ und } N_4 = (N_3 + N_u - 1) = (t_s + N_u - 1),$$

$Q \in \mathbb{R}^{n \times n}$ und $R \in \mathbb{R}^{m \times m}$ Diagonalmatrizen (Gewichtungs-) mit positiven Einträgen.

(Dies ist eine Vereinfachung, die Matrizen Q und R können auch wie im linear-quadratischen Problem als beliebige positiv definite Matrizen gewählt werden.)

Diese Optimierung liefert die optimale Kontrollfolge $u = [u^T(N_3) \ u^T(N_3 + 1) \ \dots \ u^T(N_4)]^T$ auf dem Kontrollhorizont. Die Abbildungen 5.1 und 5.2 veranschaulichen das Verhalten des Zustandes $x(t)$ bei einer Anwendung der Kontrollfolge $u(t)$ auf dem Prädiktionshorizont.

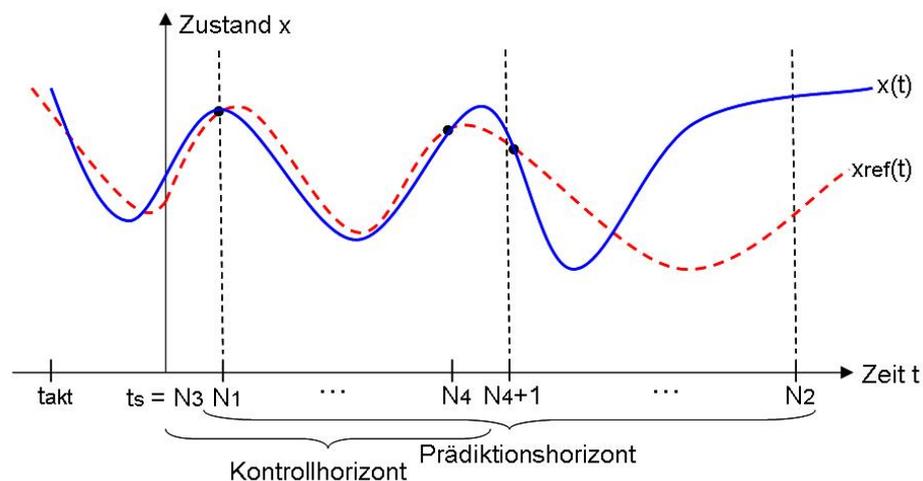
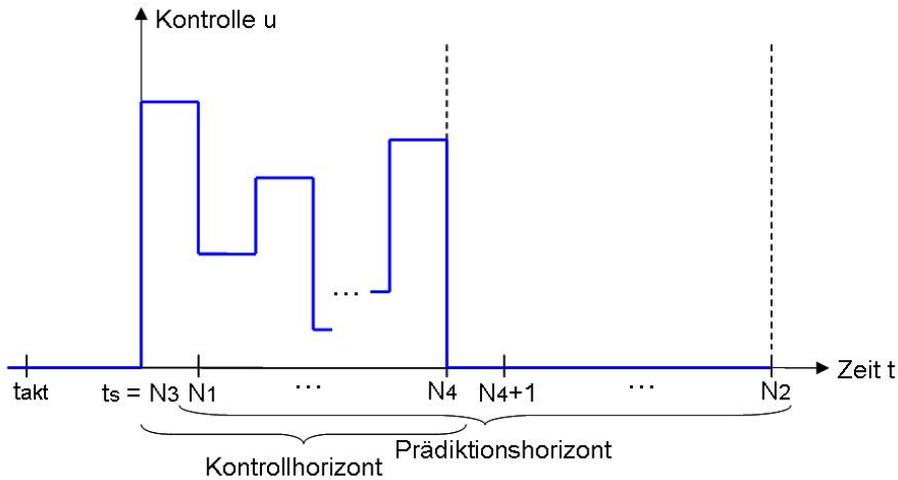


Abbildung 5.1: Verhalten des Zustandes $x(t)$ bei Anwendung der Kontrollfolge $u(t)$

Abbildung 5.2: Kontrollfolge $u(t)$

5.1.2 Die 1-Norm in der Zielfunktion

In diesem Abschnitt soll zunächst die 1-Norm (auch Summennorm genannt) definiert und anschließend die Zielfunktion $J(u, x(t))$ für die weitere Behandlung ausformuliert werden. Die dabei verwendete Schreibweise wird abschließend anhand eines Beispiels verdeutlicht.

Definition 5.1. (p-Norm)

Sei $x \in \mathbb{R}^n$. Für $p \in \mathbb{N}$ ist die p -Norm definiert als

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}.$$

Aus dieser Definition folgt für die Ausdrücke der Zielfunktion $J(u, x(t))$ mit $p = 1$:

$$\begin{aligned} \|x(t) - x_{ref}(t)\|_1 &= \sum_{i=1}^n |x_i(t) - x_{ref,i}(t)| \\ &= |x_1(t) - x_{ref,1}(t)| + |x_2(t) - x_{ref,2}(t)| + \dots + |x_n(t) - x_{ref,n}(t)| \\ \|u(t)\|_1 &= \sum_{j=1}^m |u_j(t)| = |u_1(t)| + |u_2(t)| + \dots + |u_m(t)| \end{aligned}$$

Dabei bezeichnet x_i den i -ten Eintrag des Vektors $x \in \mathbb{R}^n$, $x_{ref,i}$ den i -ten Eintrag des Vektors $x_{ref} \in \mathbb{R}^n$ und u_j den j -ten Eintrag des Vektors $u \in \mathbb{R}^m$.

Mittels dieser Schreibweise erhält man die Zielfunktion $J(u, x(t))$ der Form

$$J(u, x(t)) = \sum_{t=N_1}^{N_2} \sum_{i=1}^n Q_{ii} |x_i(t) - x_{ref,i}(t)| + \sum_{t=N_3}^{N_4} \sum_{j=1}^m R_{jj} |u_j(t)|, \quad (5.3)$$

wobei Q_{ii} der Eintrag der i -ten Zeile und i -ten Spalte der Diagonalmatrix $Q \in \mathbb{R}^{n \times n}$ und R_{jj} der Eintrag der j -ten Zeile und j -ten Spalte der Diagonalmatrix $R \in \mathbb{R}^{m \times m}$ ist.

Beispiel 5.2. Es sei $n = 3$, $m = 2$, $t_{akt} = 0$, $t_s = 2$, $N_x = 5$ und $N_u = 3$. Daraus folgt für die weiteren Parameter: $x \in \mathbb{R}^3$, $u \in \mathbb{R}^2$, $A \in \mathbb{R}^{3 \times 3}$, $B \in \mathbb{R}^{3 \times 2}$, $Q \in \mathbb{R}^{3 \times 3}$ und $R \in \mathbb{R}^{2 \times 2}$ sowie $N_1 = 3$, $N_2 = 7$, $N_3 = 2$ und $N_4 = 4$. Mit diesen Annahmen erhält man folgende dem MPC-Problem zugrunde liegende Zielfunktion:

$$\begin{aligned} J(u, x(t)) &= \sum_{t=3}^7 Q \|x(t) - x_{ref}(t)\|_1 + \sum_{t=2}^4 R \|u(t)\|_1 \\ &= \sum_{t=3}^7 \sum_{i=1}^3 Q_{ii} |x_i(t) - x_{ref,i}(t)| + \sum_{t=2}^4 \sum_{j=1}^2 R_{jj} |u_j(t)| \\ &= Q_{11} |x_1(3) - x_{ref,1}(3)| + Q_{22} |x_2(3) - x_{ref,2}(3)| + Q_{33} |x_3(3) - x_{ref,3}(3)| + \\ &+ \dots + \\ &+ Q_{11} |x_1(7) - x_{ref,1}(7)| + Q_{22} |x_2(7) - x_{ref,2}(7)| + Q_{33} |x_3(7) - x_{ref,3}(7)| + \\ &+ R_{11} |u_1(2)| + R_{22} |u_2(2)| + \\ &+ \dots + \\ &+ R_{11} |u_1(4)| + R_{22} |u_2(4)| \end{aligned}$$

5.1.3 Umformung in ein äquivalentes Optimierungsproblem

Bevor man das Optimierungsproblem (5.2) mit (5.3) in ein lineares Programm umformen kann, muss zunächst die Frage geklärt werden, wie die in (5.3) auftretenden Beträge in einer äquivalenten Formulierung ausgedrückt werden können. Das Ziel ist also ein Optimierungsproblem aufzustellen, dessen Minimierung dasselbe Ergebnis liefert wie die Lösung des ursprünglichen Problems (5.2), in dem dann aber keine Beträge mehr auftreten. Die Vorgehensweise beruht dabei auf folgender Idee:

$$\min_x \{|Ax + b|\} \quad (5.4)$$

\iff

$$\min_{z=[x^T \ y^T]^T} \{y\} \quad \text{s.t.} \quad -y \leq Ax + b \leq y \quad (5.5)$$

Man sieht leicht, dass beide Optimierungsprobleme das gleiche Minimum erreichen. Des Weiteren liefert die Lösung von (5.5) denselben Lösungsvektor x wie das ursprüngliche Problem (5.4). Es ist also möglich, durch Einführen von Schlupfvariablen ein äquivalentes Optimierungsproblem aufzustellen, in dem die in der Zielfunktion ursprünglich auftretenden Beträge nicht mehr explizit vorkommen.

Diese Idee kann folglich auch auf das Optimierungsproblem (5.2), (5.3) angewendet werden. Man erhält nach Einführen der Schlupfvariablen $\gamma_{i,t}$ und $\delta_{j,t}$ folgendes äquivalente Problem:

$$\min_{z=[u \ \gamma \ \delta]^T} \left\{ \sum_{t=N_1}^{N_2} \sum_{i=1}^n \gamma_{i,t} + \sum_{t=N_3}^{N_4} \sum_{j=1}^m \delta_{j,t} \right\} \quad (5.6)$$

s.t.

$$\begin{aligned} -\gamma_{i,t} &\leq Q_{ii}(x_i(t) - x_{ref,i}(t)) \leq \gamma_{i,t}, & i &= 1, \dots, n \\ & & t &= N_1, \dots, N_2, \\ -\delta_{j,t} &\leq R_{jj}u_j(t) \leq \delta_{j,t}, & j &= 1, \dots, m \\ & & t &= N_3, \dots, N_4 \\ x(t+1) &= Ax(t) + Bu(t), & t &= N_1 - 1, \dots, N_2 - 1 \\ u(t) &= 0, & N_4 + 1 &\leq t \leq N_2 - 1 \end{aligned}$$

Dabei gilt

$$u = [u^T(N_3) \ u^T(N_3 + 1) \ \dots \ u^T(N_4)] = \quad (5.7)$$

$$= [u_1(N_3) \ \dots \ u_m(N_3) \ u_1(N_3 + 1) \ \dots \ u_m(N_3 + 1) \ \dots \ u_1(N_4) \ \dots \ u_m(N_4)],$$

$$\gamma = [\gamma_{1,N_1} \ \dots \ \gamma_{n,N_1} \ \gamma_{1,(N_1+1)} \ \dots \ \gamma_{n,(N_1+1)} \ \dots \ \gamma_{1,N_2} \ \dots \ \gamma_{n,N_2}] \quad \text{und} \quad (5.8)$$

$$\delta = [\delta_{1,N_3} \ \dots \ \delta_{m,N_3} \ \delta_{1,(N_3+1)} \ \dots \ \delta_{m,(N_3+1)} \ \dots \ \delta_{1,N_4} \ \dots \ \delta_{m,N_4}]. \quad (5.9)$$

Auf diese Weise lassen sich die in der Zielfunktion (5.3) auftretenden Beträge formulieren. Zudem stellt die Summe der Komponenten des Vektors $[\gamma \ \delta] = [\gamma_{1,N_1} \ \dots \ \gamma_{n,N_1} \ \dots \ \dots \ \gamma_{1,N_2} \ \dots \ \dots \ \gamma_{n,N_2} \ \delta_{1,N_3} \ \dots \ \delta_{m,N_3} \ \dots \ \dots \ \delta_{1,N_4} \ \dots \ \delta_{m,N_4}]$ eine obere Schranke des Zielfunktional $J(u, x(t))$ dar. Der Vektor $z = [u \ \gamma \ \delta]^T$, der diese Summe minimiert und zusätzlich den Nebenbedingungen genügt, löst somit auch das ursprüngliche Problem (5.2).

Das folgende Beispiel soll die kompakte Schreibweise des Problems verdeutlichen.

Beispiel 5.3. Es gelten die Annahmen aus Beispiel 5.2. ($n = 3$, $m = 2$, $N_1 = 3$, $N_2 = 7$, $N_3 = 2$ und $N_4 = 4$). Das Optimierungsproblem (5.6) lautet dann

$$\min_{[u \ \gamma \ \delta]^T} \{(\gamma_{1,3} + \dots + \gamma_{3,3} + \gamma_{1,4} + \dots + \gamma_{3,4} + \dots \dots + \gamma_{1,7} + \dots + \gamma_{3,7}) + (\delta_{1,2} + \delta_{2,2} + \dots + \delta_{1,4} + \delta_{2,4})\}$$

s.t.

$$\begin{aligned} -\gamma_{1,3} &\leq Q_{11} [x_1(3) - x_{ref,1}(3)] \leq \gamma_{1,3} \\ &\vdots \\ -\gamma_{3,3} &\leq Q_{33} [x_3(3) - x_{ref,3}(3)] \leq \gamma_{3,3} \\ -\gamma_{1,4} &\leq Q_{11} [x_1(4) - x_{ref,1}(4)] \leq \gamma_{1,4} \\ &\vdots \\ -\gamma_{3,4} &\leq Q_{33} [x_3(4) - x_{ref,3}(4)] \leq \gamma_{3,4} \\ &\vdots \\ &\vdots \\ -\gamma_{1,7} &\leq Q_{11} [x_1(7) - x_{ref,1}(7)] \leq \gamma_{1,7} \\ &\vdots \\ -\gamma_{3,7} &\leq Q_{33} [x_3(7) - x_{ref,3}(7)] \leq \gamma_{3,7} \\ -\delta_{1,2} &\leq R_{11}u_1(2) \leq \delta_{1,2} \\ -\delta_{2,2} &\leq R_{22}u_2(2) \leq \delta_{2,2} \\ &\vdots \\ -\delta_{1,4} &\leq R_{11}u_1(4) \leq \delta_{1,4} \\ -\delta_{2,4} &\leq R_{22}u_2(4) \leq \delta_{2,4} \\ x(3) &= Ax(2) + Bu(2) \\ &\vdots \\ x(7) &= Ax(6) + Bu(6) \\ u(5) &= 0 \\ u(6) &= 0 \end{aligned}$$

mit

$$\begin{aligned} u &= [u^T(2) \ u^T(3) \ u^T(4)] = [u_1(2) \ u_2(2) \ u_1(3) \ u_2(3) \ u_1(4) \ u_2(4)], \\ \gamma &= [\gamma_{1,3} \ \dots \ \gamma_{3,3} \ \gamma_{1,4} \ \dots \ \gamma_{3,4} \ \dots \ \dots \ \gamma_{1,7} \ \dots \ \gamma_{3,7}] \text{ und} \\ \delta &= [\delta_{1,2} \ \delta_{2,2} \ \dots \ \dots \ \delta_{1,4} \ \delta_{2,4}]. \end{aligned}$$

Berechnung des Zustandes $x(t)$ für alle $t > t_s$:

Die Gleichung

$$x(t) = A^{(t-t_s)}x(t_s) + \sum_{r=0}^{t-t_s-1} A^r B u(t-1-r) \quad (5.10)$$

mit

$$x(t_s) = A^{(t_s-t_{akt})}x(t_{akt}), \quad \text{falls } t_s > t_{akt} \quad (5.11)$$

erleichtert es den Zustand $x(t)$ in jedem Schritt rekursiv auszurechnen. Diese Methode hat den Vorteil, dass man die Zustände direkt in Abhängigkeit des aktuellen Zustandes erhält. Unter Berücksichtigung der Gleichungen (5.10), (5.11) kann das zu Problem (5.2) äquivalente lineare Programm aufgestellt werden:

$$\min_{[u \ \gamma \ \delta]^T} \left\{ \sum_{t=N_1}^{N_2} \sum_{i=1}^n \gamma_{i,t} + \sum_{t=N_3}^{N_4} \sum_{j=1}^m \delta_{j,t} \right\} \quad (5.12)$$

s.t.

$$\begin{aligned} -\gamma_{i,t} \leq Q_{ii} \left(\left[A^{(t-t_s)} A^{(t_s-t_{akt})} x(t_{akt}) + \sum_{r=0}^{t-t_s-1} A^r B u(t-1-r) \right]_i - x_{ref,i}(t) \right) \leq \gamma_{i,t}, \\ i = 1, \dots, n \\ t = N_1, \dots, N_2 \\ -\delta_{j,t} \leq R_{jj} u_j(t) \leq \delta_{j,t}, \quad j = 1, \dots, m \\ t = N_3, \dots, N_4 \\ u(t) = 0, \quad N_4 + 1 \leq t \leq N_2 - 1 \end{aligned}$$

mit (5.7), (5.8) und (5.9). Weiterhin bezeichnet $[\cdot]_i$ den i -ten Eintrag des rekursiv berechneten Zustandes $x(t)$.

5.1.4 Das lineare Programm in Matrizenform

Zur Implementierung des linearen Programms, muss dieses in der Matrizenform

$$\begin{aligned} \min_z f \cdot z \\ \text{s.t. } E \cdot z - d \leq 0 \end{aligned} \quad (5.13)$$

vorliegen. Um zu verdeutlichen, wie das Optimierungsproblem (5.12) in ein lineares Programm der Form (5.13) umgeschrieben werden kann, soll dies hier zunächst anhand eines überschaubaren Beispiels hergeleitet werden. Anschließend wird das der modellprädiktiven Regelung zugrunde liegende lineare Programm für Kontrollsysteme ohne Ausgang in allgemeiner Form aufgestellt.

Herleitung des linearen Programms in Matrizenform

Es sei $n = 3$, $m = 2$, $t_{akt} = 0$, $t_s = 2$, $N_x = 3$, $N_u = 2$. Daraus folgt: $x \in \mathbb{R}^3$, $u \in \mathbb{R}^2$, $A \in \mathbb{R}^{3 \times 3}$, $B \in \mathbb{R}^{3 \times 2}$, $Q \in \mathbb{R}^{3 \times 3}$ und $R \in \mathbb{R}^{2 \times 2}$ sowie $N_1 = 3$, $N_2 = 5$, $N_3 = 2$ und $N_4 = 3$.

Das Optimierungsproblem (5.6) lautet für dieses Beispiel

$$\min_{[u \ \gamma \ \delta]^T} \left\{ (\gamma_{1,3} + \gamma_{2,3} + \gamma_{3,3} + \gamma_{1,4} + \gamma_{2,4} + \gamma_{3,4} + \gamma_{1,5} + \gamma_{2,5} + \gamma_{3,5}) + (\delta_{1,2} + \delta_{2,2} + \delta_{1,3} + \delta_{2,3}) \right\}$$

s.t.

$$\begin{aligned} -\gamma_{1,3} &\leq Q_{11} [x_1(3) - x_{ref,1}(3)] \leq \gamma_{1,3} \\ -\gamma_{2,3} &\leq Q_{22} [x_2(3) - x_{ref,2}(3)] \leq \gamma_{2,3} \\ -\gamma_{3,3} &\leq Q_{33} [x_3(3) - x_{ref,3}(3)] \leq \gamma_{3,3} \\ -\gamma_{1,4} &\leq Q_{11} [x_1(4) - x_{ref,1}(4)] \leq \gamma_{1,4} \\ -\gamma_{2,4} &\leq Q_{22} [x_2(4) - x_{ref,2}(4)] \leq \gamma_{2,4} \\ -\gamma_{3,4} &\leq Q_{33} [x_3(4) - x_{ref,3}(4)] \leq \gamma_{3,4} \\ -\gamma_{1,5} &\leq Q_{11} [x_1(5) - x_{ref,1}(5)] \leq \gamma_{1,5} \\ -\gamma_{2,5} &\leq Q_{22} [x_2(5) - x_{ref,2}(5)] \leq \gamma_{2,5} \\ -\gamma_{3,5} &\leq Q_{33} [x_3(5) - x_{ref,3}(5)] \leq \gamma_{3,5} \\ -\delta_{1,2} &\leq R_{11} u_1(2) \leq \delta_{1,2} \\ -\delta_{2,2} &\leq R_{22} u_2(2) \leq \delta_{2,2} \\ -\delta_{1,3} &\leq R_{11} u_1(3) \leq \delta_{1,3} \\ -\delta_{2,3} &\leq R_{22} u_2(3) \leq \delta_{2,3} \\ x(3) &= Ax(2) + Bu(2) \\ x(4) &= Ax(3) + Bu(3) \\ x(5) &= Ax(4) + Bu(4) \\ u(4) &= 0. \end{aligned}$$

Man rechnet leicht nach, dass die rekursive Berechnung (5.10), (5.11) des Zustandes $x(t)$ für $t = 3, \dots, 5$ folgendes Ergebnis liefert:

$$\begin{aligned} x(3) &= AA^2x(0) + Bu(2) \\ x(4) &= A^2A^2x(0) + Bu(3) + ABu(2) \\ x(5) &= A^3A^2x(0) + Bu(4) + ABu(3) + A^2Bu(2) \end{aligned}$$

Die Zustände hängen nun somit nur noch vom aktuellen und damit bekannten Zustand $x(0) = x(t_{akt})$ ab. Daraus ergibt sich nach Umstellen und Ausmultiplizieren für die Nebenbedingungen

$$\begin{aligned}
& -\gamma_{1,3} \pm Q_{11}[AA^2x(0)]_1 \pm Q_{11}[Bu(2)]_1 \mp Q_{11}x_{ref,1}(3) \leq 0 \\
& -\gamma_{2,3} \pm Q_{22}[AA^2x(0)]_2 \pm Q_{22}[Bu(2)]_2 \mp Q_{22}x_{ref,2}(3) \leq 0 \\
& -\gamma_{3,3} \pm Q_{33}[AA^2x(0)]_3 \pm Q_{33}[Bu(2)]_3 \mp Q_{33}x_{ref,3}(3) \leq 0 \\
& -\gamma_{1,4} \pm Q_{11}[A^2A^2x(0)]_1 \pm Q_{11}[Bu(3)]_1 \pm Q_{11}[ABu(2)]_1 \mp Q_{11}x_{ref,1}(4) \leq 0 \\
& -\gamma_{2,4} \pm Q_{22}[A^2A^2x(0)]_2 \pm Q_{22}[Bu(3)]_2 \pm Q_{22}[ABu(2)]_2 \mp Q_{22}x_{ref,2}(4) \leq 0 \\
& -\gamma_{3,4} \pm Q_{33}[A^2A^2x(0)]_3 \pm Q_{33}[Bu(3)]_3 \pm Q_{33}[ABu(2)]_3 \mp Q_{33}x_{ref,3}(4) \leq 0 \\
& -\gamma_{1,5} \pm Q_{11}[A^3A^2x(0)]_1 \pm \underbrace{Q_{11}[Bu(4)]_1}_{=0} \pm Q_{11}[ABu(3)]_1 \pm Q_{11}[A^2Bu(2)]_1 \mp Q_{11}x_{ref,1}(5) \leq 0 \\
& -\gamma_{2,5} \pm Q_{22}[A^3A^2x(0)]_2 \pm \underbrace{Q_{22}[Bu(4)]_2}_{=0} \pm Q_{22}[ABu(3)]_2 \pm Q_{22}[A^2Bu(2)]_2 \mp Q_{22}x_{ref,2}(5) \leq 0 \\
& -\gamma_{3,5} \pm Q_{33}[A^3A^2x(0)]_3 \pm \underbrace{Q_{33}[Bu(4)]_3}_{=0} \pm Q_{33}[ABu(3)]_3 \pm Q_{33}[A^2Bu(2)]_3 \mp Q_{33}x_{ref,3}(5) \leq 0 \\
& -\delta_{1,2} \pm R_{11}u_1(2) \leq 0 \\
& -\delta_{2,2} \pm R_{22}u_2(2) \leq 0 \\
& -\delta_{1,3} \pm R_{11}u_1(3) \leq 0 \\
& -\delta_{2,3} \pm R_{22}u_2(3) \leq 0 \\
& u(4) = 0.
\end{aligned}$$

Um zu verdeutlichen, wie das lineare Programm in Matrizenform aufgestellt werden kann, werden zunächst noch die Ausdrücke $[\cdot u(t)]_i$ berechnet. Dabei kann wie folgt vorgegangen werden.

Es gilt

$$Bu(t) = \underbrace{\begin{pmatrix} B_{11} & B_{12} & \dots & B_{1m} \\ B_{21} & B_{22} & \dots & B_{2m} \\ \vdots & & & \vdots \\ B_{n1} & B_{n2} & \dots & B_{nm} \end{pmatrix}}_{=B} \underbrace{\begin{pmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_m(t) \end{pmatrix}}_{=u(t)} = \begin{pmatrix} B_{11}u_1(t) + B_{12}u_2(t) + \dots + B_{1m}u_m(t) \\ B_{21}u_1(t) + B_{22}u_2(t) + \dots + B_{2m}u_m(t) \\ \vdots \\ B_{n1}u_1(t) + B_{n2}u_2(t) + \dots + B_{nm}u_m(t) \end{pmatrix}$$

sowie

$$\begin{aligned}
A^k Bu(t) &= \underbrace{\begin{pmatrix} A_{11}^k & A_{12}^k & \dots & A_{1n}^k \\ A_{21}^k & A_{22}^k & \dots & A_{2n}^k \\ \vdots & & & \vdots \\ A_{n1}^k & A_{n2}^k & \dots & A_{nn}^k \end{pmatrix}}_{=A^k} \underbrace{\begin{pmatrix} B_{11}u_1(t) + B_{12}u_2(t) + \dots + B_{1m}u_m(t) \\ B_{21}u_1(t) + B_{22}u_2(t) + \dots + B_{2m}u_m(t) \\ \vdots \\ B_{n1}u_1(t) + B_{n2}u_2(t) + \dots + B_{nm}u_m(t) \end{pmatrix}}_{=Bu(t)} = \\
&= \begin{pmatrix} A_{11}^k(B_{11}u_1(t) + \dots + B_{1m}u_m(t)) + \dots + A_{1n}^k(B_{n1}u_1(t) + \dots + B_{nm}u_m(t)) \\ \vdots \\ A_{n1}^k(B_{11}u_1(t) + \dots + B_{1m}u_m(t)) + \dots + A_{nn}^k(B_{n1}u_1(t) + \dots + B_{nm}u_m(t)) \end{pmatrix}.
\end{aligned}$$

Daraus folgt für die i -ten Einträge $[u(t)]_i$

$$[Bu(t)]_i = \sum_{j_1=1}^m B_{ij_1} u_{j_1}(t) = B_{i1}u_1(t) + B_{i2}u_2(t) + \dots + B_{im}u_m(t) \quad (5.14)$$

$$\begin{aligned} [A^k Bu(t)]_i &= \sum_{j_2=1}^n A_{ij_2}^k [Bu(t)]_{j_2} = A_{i1}^k [Bu(t)]_1 + \dots + A_{in}^k [Bu(t)]_n \quad \implies \\ [A^k Bu(t)]_i &= \sum_{j_2=1}^n \left(A_{ij_2}^k \sum_{j_1=1}^m B_{j_2 j_1} u_{j_1}(t) \right) \quad (5.15) \\ &= A_{i1}^k (B_{11}u_1(t) + \dots + B_{1m}u_m(t)) + \dots + A_{in}^k (B_{n1}u_1(t) + \dots + B_{nm}u_m(t)) \end{aligned}$$

Nach Anwendung der Gleichungen (5.14) bzw. (5.15) erhält man schließlich die Nebenbedingungen

$$\begin{aligned} -\gamma_{1,3} \pm Q_{11}[AA^2x(0)]_1 \pm Q_{11}B_{11}u_1(2) \pm Q_{11}B_{12}u_2(2) \mp Q_{11}x_{ref,1}(3) &\leq 0 \\ -\gamma_{2,3} \pm Q_{22}[AA^2x(0)]_2 \pm Q_{22}B_{21}u_1(2) \pm Q_{22}B_{22}u_2(2) \mp Q_{22}x_{ref,2}(3) &\leq 0 \\ -\gamma_{3,3} \pm Q_{33}[AA^2x(0)]_3 \pm Q_{33}B_{31}u_1(2) \pm Q_{33}B_{32}u_2(2) \mp Q_{33}x_{ref,3}(3) &\leq 0 \\ -\gamma_{1,4} \pm Q_{11}[A^2A^2x(0)]_1 \pm Q_{11}B_{11}u_1(3) \pm Q_{11}B_{12}u_2(3) \pm Q_{11}(A_{11}B_{11}u_1(2) + A_{11}B_{12}u_2(2) + \\ + A_{12}B_{21}u_1(2) + A_{12}B_{22}u_2(2) + A_{13}B_{31}u_1(2) + A_{13}B_{32}u_2(2)) \mp Q_{11}x_{ref,1}(4) &\leq 0 \\ -\gamma_{2,4} \pm Q_{22}[A^2A^2x(0)]_2 \pm Q_{22}B_{21}u_1(3) \pm Q_{22}B_{22}u_2(3) \pm Q_{22}(A_{21}B_{11}u_1(2) + A_{21}B_{12}u_2(2) + \\ + A_{22}B_{21}u_1(2) + A_{22}B_{22}u_2(2) + A_{23}B_{31}u_1(2) + A_{23}B_{32}u_2(2)) \mp Q_{22}x_{ref,2}(4) &\leq 0 \\ -\gamma_{3,4} \pm Q_{33}[A^2A^2x(0)]_3 \pm Q_{33}B_{31}u_1(3) \pm Q_{33}B_{32}u_2(3) \pm Q_{33}(A_{31}B_{11}u_1(2) + A_{31}B_{12}u_2(2) + \\ + A_{32}B_{21}u_1(2) + A_{32}B_{22}u_2(2) + A_{33}B_{31}u_1(2) + A_{33}B_{32}u_2(2)) \mp Q_{33}x_{ref,3}(4) &\leq 0 \\ -\gamma_{1,5} \pm Q_{11}[A^3A^2x(0)]_1 \pm Q_{11}(A_{11}B_{11}u_1(3) + A_{11}B_{12}u_2(3) + A_{12}B_{21}u_1(3) + A_{12}B_{22}u_2(3) + \\ + A_{13}B_{31}u_1(3) + A_{13}B_{32}u_2(3)) \pm Q_{11}(A_{11}^2B_{11}u_1(2) + A_{11}^2B_{12}u_2(2) + A_{12}^2B_{21}u_1(2) + \\ + A_{12}^2B_{22}u_2(2) + A_{13}^2B_{31}u_1(2) + A_{13}^2B_{32}u_2(2)) \mp Q_{11}x_{ref,1}(5) &\leq 0 \\ -\gamma_{2,5} \pm Q_{22}[A^3A^2x(0)]_2 \pm Q_{22}(A_{21}B_{11}u_1(3) + A_{21}B_{12}u_2(3) + A_{22}B_{21}u_1(3) + A_{22}B_{22}u_2(3) + \\ + A_{23}B_{31}u_1(3) + A_{23}B_{32}u_2(3)) \pm Q_{22}(A_{21}^2B_{11}u_1(2) + A_{21}^2B_{12}u_2(2) + A_{22}^2B_{21}u_1(2) + \\ + A_{22}^2B_{22}u_2(2) + A_{23}^2B_{31}u_1(2) + A_{23}^2B_{32}u_2(2)) \mp Q_{22}x_{ref,2}(5) &\leq 0 \\ -\gamma_{3,5} \pm Q_{33}[A^3A^2x(0)]_3 \pm Q_{33}(A_{31}B_{11}u_1(3) + A_{31}B_{12}u_2(3) + A_{32}B_{21}u_1(3) + A_{32}B_{22}u_2(3) + \\ + A_{33}B_{31}u_1(3) + A_{33}B_{32}u_2(3)) \pm Q_{33}(A_{31}^2B_{11}u_1(2) + A_{31}^2B_{12}u_2(2) + A_{32}^2B_{21}u_1(2) + \\ + A_{32}^2B_{22}u_2(2) + A_{33}^2B_{31}u_1(2) + A_{33}^2B_{32}u_2(2)) \mp Q_{33}x_{ref,3}(5) &\leq 0 \\ -\delta_{1,2} \pm R_{11}u_1(2) &\leq 0 \\ -\delta_{2,2} \pm R_{22}u_2(2) &\leq 0 \\ -\delta_{1,3} \pm R_{11}u_1(3) &\leq 0 \\ -\delta_{2,3} \pm R_{22}u_2(3) &\leq 0 \end{aligned}$$

Nun stellt die Umformung der Nebenbedingungen in Matrizenform kein Problem dar. Die Ausdrücke werden zunächst zur besseren Verständlichkeit in Tabellenform sortiert, woraus

schließlich die Matrix E als auch der Vektor d konstruiert werden kann.

Konstruktion der Matrix E :

$\cdot \mathbf{u}_1(\mathbf{2})$	$\cdot \mathbf{u}_2(\mathbf{2})$
$Q_{11}B_{11}$	$Q_{11}B_{12}$
$Q_{22}B_{21}$	$Q_{22}B_{22}$
$Q_{33}B_{31}$	$Q_{33}B_{32}$
$Q_{11}(A_{11}B_{11} + A_{12}B_{21} + A_{13}B_{31})$	$Q_{11}(A_{11}B_{12} + A_{12}B_{22} + A_{13}B_{32})$
$Q_{22}(A_{21}B_{11} + A_{22}B_{21} + A_{23}B_{31})$	$Q_{22}(A_{21}B_{12} + A_{22}B_{22} + A_{23}B_{32})$
$Q_{33}(A_{31}B_{11} + A_{32}B_{21} + A_{33}B_{31})$	$Q_{33}(A_{31}B_{12} + A_{32}B_{22} + A_{33}B_{32})$
$Q_{11}(A_{11}^2B_{11} + A_{12}^2B_{21} + A_{13}^2B_{31})$	$Q_{11}(A_{11}^2B_{12} + A_{12}^2B_{22} + A_{13}^2B_{32})$
$Q_{22}(A_{21}^2B_{11} + A_{22}^2B_{21} + A_{23}^2B_{31})$	$Q_{22}(A_{21}^2B_{12} + A_{22}^2B_{22} + A_{23}^2B_{32})$
$Q_{33}(A_{31}^2B_{11} + A_{32}^2B_{21} + A_{33}^2B_{31})$	$Q_{33}(A_{31}^2B_{12} + A_{32}^2B_{22} + A_{33}^2B_{32})$
$-Q_{11}B_{11}$	$-Q_{11}B_{12}$
$-Q_{22}B_{21}$	$-Q_{22}B_{22}$
$-Q_{33}B_{31}$	$-Q_{33}B_{32}$
$-Q_{11}(A_{11}B_{11} + A_{12}B_{21} + A_{13}B_{31})$	$-Q_{11}(A_{11}B_{12} + A_{12}B_{22} + A_{13}B_{32})$
$-Q_{22}(A_{21}B_{11} + A_{22}B_{21} + A_{23}B_{31})$	$-Q_{22}(A_{21}B_{12} + A_{22}B_{22} + A_{23}B_{32})$
$-Q_{33}(A_{31}B_{11} + A_{32}B_{21} + A_{33}B_{31})$	$-Q_{33}(A_{31}B_{12} + A_{32}B_{22} + A_{33}B_{32})$
$-Q_{11}(A_{11}^2B_{11} + A_{12}^2B_{21} + A_{13}^2B_{31})$	$-Q_{11}(A_{11}^2B_{12} + A_{12}^2B_{22} + A_{13}^2B_{32})$
$-Q_{22}(A_{21}^2B_{11} + A_{22}^2B_{21} + A_{23}^2B_{31})$	$-Q_{22}(A_{21}^2B_{12} + A_{22}^2B_{22} + A_{23}^2B_{32})$
$-Q_{33}(A_{31}^2B_{11} + A_{32}^2B_{21} + A_{33}^2B_{31})$	$-Q_{33}(A_{31}^2B_{12} + A_{32}^2B_{22} + A_{33}^2B_{32})$
R_{11}	0
0	R_{22}
0	0
0	0
$-R_{11}$	0
0	$-R_{22}$
0	0
0	0

$\cdot \mathbf{u}_1(\mathbf{z})$	$\cdot \mathbf{u}_2(\mathbf{z})$
0	0
0	0
0	0
$Q_{11}B_{11}$	$Q_{11}B_{12}$
$Q_{22}B_{21}$	$Q_{22}B_{22}$
$Q_{33}B_{31}$	$Q_{33}B_{32}$
$Q_{11}(A_{11}B_{11} + A_{12}B_{21} + A_{13}B_{31})$	$Q_{11}(A_{11}B_{12} + A_{12}B_{22} + A_{13}B_{32})$
$Q_{22}(A_{21}B_{11} + A_{22}B_{21} + A_{23}B_{31})$	$Q_{22}(A_{21}B_{12} + A_{22}B_{22} + A_{23}B_{32})$
$Q_{33}(A_{31}B_{11} + A_{32}B_{21} + A_{33}B_{31})$	$Q_{33}(A_{31}B_{12} + A_{32}B_{22} + A_{33}B_{32})$
0	0
0	0
0	0
$-Q_{11}B_{11}$	$-Q_{11}B_{12}$
$-Q_{22}B_{21}$	$-Q_{22}B_{22}$
$-Q_{33}B_{31}$	$-Q_{33}B_{32}$
$-Q_{11}(A_{11}B_{11} + A_{12}B_{21} + A_{13}B_{31})$	$-Q_{11}(A_{11}B_{12} + A_{12}B_{22} + A_{13}B_{32})$
$-Q_{22}(A_{21}B_{11} + A_{22}B_{21} + A_{23}B_{31})$	$-Q_{22}(A_{21}B_{12} + A_{22}B_{22} + A_{23}B_{32})$
$-Q_{33}(A_{31}B_{11} + A_{32}B_{21} + A_{33}B_{31})$	$-Q_{33}(A_{31}B_{12} + A_{32}B_{22} + A_{33}B_{32})$
0	0
0	0
R_{11}	0
0	R_{22}
0	0
0	0
$-R_{11}$	0
0	$-R_{22}$

Tabelle 5.1: Einträge, welche mit $u_j(t)$ multipliziert werden

$\cdot\gamma_{1,3}$	$\cdot\gamma_{2,3}$	$\cdot\gamma_{3,3}$	$\cdot\gamma_{1,4}$	$\cdot\gamma_{2,4}$	$\cdot\gamma_{3,4}$	$\cdot\gamma_{1,5}$	$\cdot\gamma_{2,5}$	$\cdot\gamma_{3,5}$	$\cdot\delta_{1,2}$	$\cdot\delta_{2,2}$	$\cdot\delta_{1,3}$	$\cdot\delta_{2,3}$
-1	0	0	0	0	0	0	0	0	0	0	0	0
0	-1	0	0	0	0	0	0	0	0	0	0	0
0	0	-1	0	0	0	0	0	0	0	0	0	0
0	0	0	-1	0	0	0	0	0	0	0	0	0
0	0	0	0	-1	0	0	0	0	0	0	0	0
0	0	0	0	0	-1	0	0	0	0	0	0	0
0	0	0	0	0	0	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	-1	0	0	0	0	0
0	0	0	0	0	0	0	0	-1	0	0	0	0
-1	0	0	0	0	0	0	0	0	0	0	0	0
0	-1	0	0	0	0	0	0	0	0	0	0	0
0	0	-1	0	0	0	0	0	0	0	0	0	0
0	0	0	-1	0	0	0	0	0	0	0	0	0
0	0	0	0	-1	0	0	0	0	0	0	0	0
0	0	0	0	0	-1	0	0	0	0	0	0	0
0	0	0	0	0	0	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	-1	0	0	0	0	0
0	0	0	0	0	0	0	0	-1	0	0	0	0
0	0	0	0	0	0	0	0	0	-1	0	0	0
0	0	0	0	0	0	0	0	0	0	-1	0	0
0	0	0	0	0	0	0	0	0	0	0	-1	0
0	0	0	0	0	0	0	0	0	0	0	0	-1
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	-1	0
0	0	0	0	0	0	0	0	0	0	0	0	-1

Tabelle 5.2: Einträge, welche mit $\gamma_{i,t}$ bzw. $\delta_{j,t}$ multipliziert werden

Konstruktion des Vektors d :

$$\begin{array}{c}
 Q_{11} [AA^2x(0)]_1 - Q_{11}x_{ref,1}(3) \\
 Q_{22} [AA^2x(0)]_2 - Q_{22}x_{ref,2}(3) \\
 Q_{33} [AA^2x(0)]_3 - Q_{33}x_{ref,3}(3) \\
 \hline
 Q_{11} [A^2A^2x(0)]_1 - Q_{11}x_{ref,1}(4) \\
 Q_{22} [A^2A^2x(0)]_2 - Q_{22}x_{ref,2}(4) \\
 Q_{33} [A^2A^2x(0)]_3 - Q_{33}x_{ref,3}(4) \\
 \hline
 Q_{11} [A^3A^2x(0)]_1 - Q_{11}x_{ref,1}(5) \\
 Q_{22} [A^3A^2x(0)]_2 - Q_{22}x_{ref,2}(5) \\
 Q_{33} [A^3A^2x(0)]_3 - Q_{33}x_{ref,3}(5) \\
 \hline
 -Q_{11} [AA^2x(0)]_1 + Q_{11}x_{ref,1}(3) \\
 -Q_{22} [AA^2x(0)]_2 + Q_{22}x_{ref,2}(3) \\
 -Q_{33} [AA^2x(0)]_3 + Q_{33}x_{ref,3}(3) \\
 \hline
 -Q_{11} [A^2A^2x(0)]_1 + Q_{11}x_{ref,1}(4) \\
 -Q_{22} [A^2A^2x(0)]_2 + Q_{22}x_{ref,2}(4) \\
 -Q_{33} [A^2A^2x(0)]_3 + Q_{33}x_{ref,3}(4) \\
 \hline
 -Q_{11} [A^3A^2x(0)]_1 + Q_{11}x_{ref,1}(5) \\
 -Q_{22} [A^3A^2x(0)]_2 + Q_{22}x_{ref,2}(5) \\
 -Q_{33} [A^3A^2x(0)]_3 + Q_{33}x_{ref,3}(5) \\
 \hline
 0 \\
 0 \\
 \hline
 0 \\
 0 \\
 \hline
 0 \\
 0 \\
 \hline
 0 \\
 0
 \end{array}$$

Tabelle 5.3: Einträge, über welche nicht minimiert wird

Auf Basis dieser Tabellen kann für dieses Beispiel problemlos das lineare Programm der Form (5.13) aufgestellt werden. Dieses lautet dann wie folgt:

$$\begin{array}{ll}
 \min_z & f \cdot z \\
 \text{s.t.} & E \cdot z - d \leq 0
 \end{array}$$

Dabei gilt

$$\mathbf{f} = [0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$$

$$\mathbf{z} = [u \ \gamma \ \delta]^T$$

mit

$$u = [u_1(2) \ u_2(2) \ u_1(3) \ u_2(3)],$$

$$\gamma = [\gamma_{1,3} \ \gamma_{2,3} \ \gamma_{3,3} \ \gamma_{1,4} \ \gamma_{2,4} \ \gamma_{3,4} \ \gamma_{1,5} \ \gamma_{2,5} \ \gamma_{3,5}] \quad \text{und}$$

$$\delta = [\delta_{1,2} \ \delta_{2,2} \ \delta_{1,3} \ \delta_{2,3}].$$

Weiterhin ist die Matrix E gegeben durch

$$\mathbf{E} = \begin{pmatrix} E_1 & E_3 & Z_2 \\ -E_1 & E_3 & Z_2 \\ E_2 & Z_1 & E_4 \\ -E_2 & Z_1 & E_4 \end{pmatrix},$$

wobei

$$E_1 = \begin{pmatrix} E_{11} & Z_{11} \\ E_{12} & E_{11} \\ E_{13} & E_{12} \end{pmatrix}$$

mit

$$E_{11} = \begin{pmatrix} Q_{11}B_{11} & Q_{11}B_{12} \\ Q_{22}B_{21} & Q_{22}B_{22} \\ Q_{33}B_{31} & Q_{33}B_{32} \end{pmatrix}$$

$$E_{12} = \begin{pmatrix} Q_{11}(A_{11}B_{11} + A_{12}B_{21} + A_{13}B_{31}) & Q_{11}(A_{11}B_{12} + A_{12}B_{22} + A_{13}B_{32}) \\ Q_{22}(A_{21}B_{11} + A_{22}B_{21} + A_{23}B_{31}) & Q_{22}(A_{21}B_{12} + A_{22}B_{22} + A_{23}B_{32}) \\ Q_{33}(A_{31}B_{11} + A_{32}B_{21} + A_{33}B_{31}) & Q_{33}(A_{31}B_{12} + A_{32}B_{22} + A_{33}B_{32}) \end{pmatrix}$$

$$E_{13} = \begin{pmatrix} Q_{11}(A_{11}^2B_{11} + A_{12}^2B_{21} + A_{13}^2B_{31}) & Q_{11}(A_{11}^2B_{12} + A_{12}^2B_{22} + A_{13}^2B_{32}) \\ Q_{22}(A_{21}^2B_{11} + A_{22}^2B_{21} + A_{23}^2B_{31}) & Q_{22}(A_{21}^2B_{12} + A_{22}^2B_{22} + A_{23}^2B_{32}) \\ Q_{33}(A_{31}^2B_{11} + A_{32}^2B_{21} + A_{33}^2B_{31}) & Q_{33}(A_{31}^2B_{12} + A_{32}^2B_{22} + A_{33}^2B_{32}) \end{pmatrix}$$

$$Z_{11} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix},$$

Der Vektor d kann für dieses Beispiel wie folgt angegeben werden

$$\mathbf{d} = \begin{pmatrix} d_1 \\ -d_1 \\ d_2 \\ d_2 \end{pmatrix},$$

wobei

$$d_1 = \begin{pmatrix} d_{11} \\ d_{12} \\ d_{13} \end{pmatrix}$$

mit

$$d_{11} = \begin{pmatrix} -Q_{11} [AA^2x(0)]_1 + Q_{11}x_{ref,1}(3) \\ -Q_{22} [AA^2x(0)]_2 + Q_{22}x_{ref,2}(3) \\ -Q_{33} [AA^2x(0)]_3 + Q_{33}x_{ref,3}(3) \end{pmatrix}$$

$$d_{12} = \begin{pmatrix} -Q_{11} [A^2A^2x(0)]_1 + Q_{11}x_{ref,1}(4) \\ -Q_{22} [A^2A^2x(0)]_2 + Q_{22}x_{ref,2}(4) \\ -Q_{33} [A^2A^2x(0)]_3 + Q_{33}x_{ref,3}(4) \end{pmatrix}$$

$$d_{13} = \begin{pmatrix} -Q_{11} [A^3A^2x(0)]_1 + Q_{11}x_{ref,1}(5) \\ -Q_{22} [A^3A^2x(0)]_2 + Q_{22}x_{ref,2}(5) \\ -Q_{33} [A^3A^2x(0)]_3 + Q_{33}x_{ref,3}(5) \end{pmatrix},$$

$$\text{und } d_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Das lineare Programm in allgemeiner Form (für Kontrollsysteme ohne Ausgang)

Es kann jetzt das Hauptresultat dieses Abschnittes formuliert werden. Das Optimierungsproblem (5.2)

$$\min_{u:=[u^T(N_3) \dots u^T(N_4)]^T} \left\{ J(u, x(t)) = \sum_{t=N_1}^{N_2} Q \|x(t) - x_{ref}(t)\|_1 + \sum_{t=N_3}^{N_4} R \|u(t)\|_1 \right\}$$

s.t.

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), & t &= N_1 - 1, \dots, N_2 - 1 \\ u(t) &= 0, & N_4 + 1 &\leq t \leq N_2 - 1 \end{aligned}$$

kann in ein äquivalentes lineares Programm in Matrizenform umgeschrieben werden. Die Lösung dieses Programms liefert die optimale Kontrollfolge $u = [u^T(N_3) u^T(N_3+1) \dots u^T(N_4)]^T$ auf dem Kontrollhorizont, dessen erstes Glied zur Durchführung des modellprädiktiven Regelungsansatzes benötigt wird.

Das lineare Programm kann folgendermaßen formuliert werden:

$$\begin{aligned} \min_z \quad & f \cdot z \\ \text{s.t.} \quad & E \cdot z - d \leq 0 \end{aligned}$$

Dabei sind die Vektoren f und z gegeben als

$$\mathbf{f} = \underbrace{[0 \dots 0]}_{m \cdot N_u} \underbrace{[1 \dots 1]}_{n \cdot N_x} \underbrace{[1 \dots 1]}_{m \cdot N_u} \in \mathbb{R}^{1 \times (2 \cdot m \cdot N_u + n \cdot N_x)} \quad (5.16)$$

$$\mathbf{z} = [u \quad \gamma \quad \delta]^T \in \mathbb{R}^{2 \cdot m \cdot N_u + n \cdot N_x} \quad (5.17)$$

mit

$$u = [u_1(N_3) \dots u_m(N_3) \quad u_1(N_3+1) \dots u_m(N_3+1) \dots \dots u_1(N_4) \dots u_m(N_4)] \in \mathbb{R}^{1 \times (m \cdot N_u)}$$

$$\gamma = [\gamma_{1,N_1} \dots \gamma_{n,N_1} \quad \gamma_{1,(N_1+1)} \dots \gamma_{n,(N_1+1)} \dots \dots \gamma_{1,N_2} \dots \gamma_{n,N_2}] \in \mathbb{R}^{1 \times (n \cdot N_x)}$$

$$\delta = [\delta_{1,N_3} \dots \delta_{m,N_3} \quad \delta_{1,(N_3+1)} \dots \delta_{m,(N_3+1)} \dots \dots \delta_{1,N_4} \dots \delta_{m,N_4}] \in \mathbb{R}^{1 \times (m \cdot N_u)}.$$

Die Nebenbedingungsmatrix E wird wie folgt definiert

$$\mathbf{E} = \begin{pmatrix} E_1 & E_3 & Z_2 \\ -E_1 & E_3 & Z_2 \\ E_2 & Z_1 & E_4 \\ -E_2 & Z_1 & E_4 \end{pmatrix} \in \mathbb{R}^{2 \cdot (n \cdot N_x + m \cdot N_u) \times (2 \cdot m \cdot N_u + n \cdot N_x)}, \quad (5.18)$$

wobei

$$E_1 = \begin{pmatrix} E_{11} & Z_{11} & \dots & Z_{11} \\ E_{12} & E_{11} & & \vdots \\ \vdots & \ddots & \ddots & Z_{11} \\ E_{1N_u} & \dots & E_{12} & E_{11} \\ \hline E_{1(N_u+1)} & E_{1N_u} & \dots & E_{12} \\ \vdots & \ddots & \ddots & \vdots \\ E_{1(N_x)} & \dots & E_{1(N_u+1)} & E_{1N_u} \end{pmatrix} \in \mathbb{R}^{(n \cdot N_x) \times (m \cdot N_u)} \quad (5.19)$$

mit

$$E_{11} = \begin{pmatrix} Q_{11}B_{11} & Q_{11}B_{12} & \dots & Q_{11}B_{1m} \\ Q_{22}B_{21} & Q_{22}B_{22} & \dots & Q_{22}B_{2m} \\ \vdots & \vdots & \dots & \vdots \\ Q_{nn}B_{n1} & Q_{nn}B_{n2} & \dots & Q_{nn}B_{nm} \end{pmatrix} \in \mathbb{R}^{n \times m}$$

$$E_{12} = \begin{pmatrix} Q_{11}(A_{11}B_{11} + A_{12}B_{21} + \dots + A_{1n}B_{n1}) & \dots & Q_{11}(A_{11}B_{1m} + A_{12}B_{2m} + \dots + A_{1n}B_{nm}) \\ Q_{22}(A_{21}B_{11} + A_{22}B_{21} + \dots + A_{2n}B_{n1}) & \dots & Q_{22}(A_{21}B_{1m} + A_{22}B_{2m} + \dots + A_{2n}B_{nm}) \\ \vdots & \dots & \vdots \\ Q_{nn}(A_{n1}B_{11} + A_{n2}B_{21} + \dots + A_{nn}B_{n1}) & \dots & Q_{nn}(A_{n1}B_{1m} + A_{n2}B_{2m} + \dots + A_{nn}B_{nm}) \end{pmatrix} \in \mathbb{R}^{n \times m}$$

$$E_{1N_u} = \begin{pmatrix} Q_{11}(A_{11}^{N_u-1}B_{11} + \dots + A_{1n}^{N_u-1}B_{n1}) & \dots & Q_{11}(A_{11}^{N_u-1}B_{1m} + \dots + A_{1n}^{N_u-1}B_{nm}) \\ Q_{22}(A_{21}^{N_u-1}B_{11} + \dots + A_{2n}^{N_u-1}B_{n1}) & \dots & Q_{22}(A_{21}^{N_u-1}B_{1m} + \dots + A_{2n}^{N_u-1}B_{nm}) \\ \vdots & \dots & \vdots \\ Q_{nn}(A_{n1}^{N_u-1}B_{11} + \dots + A_{nn}^{N_u-1}B_{n1}) & \dots & Q_{nn}(A_{n1}^{N_u-1}B_{1m} + \dots + A_{nn}^{N_u-1}B_{nm}) \end{pmatrix} \in \mathbb{R}^{n \times m}$$

$$E_{1(N_u+1)} = \begin{pmatrix} Q_{11}(A_{11}^{N_u}B_{11} + \dots + A_{1n}^{N_u}B_{n1}) & \dots & Q_{11}(A_{11}^{N_u}B_{1m} + \dots + A_{1n}^{N_u}B_{nm}) \\ Q_{22}(A_{21}^{N_u}B_{11} + \dots + A_{2n}^{N_u}B_{n1}) & \dots & Q_{22}(A_{21}^{N_u}B_{1m} + \dots + A_{2n}^{N_u}B_{nm}) \\ \vdots & \dots & \vdots \\ Q_{nn}(A_{n1}^{N_u}B_{11} + \dots + A_{nn}^{N_u}B_{n1}) & \dots & Q_{nn}(A_{n1}^{N_u}B_{1m} + \dots + A_{nn}^{N_u}B_{nm}) \end{pmatrix} \in \mathbb{R}^{n \times m}$$

$$E_{1N_x} = \begin{pmatrix} Q_{11}(A_{11}^{N_x-1}B_{11} + \dots + A_{1n}^{N_x-1}B_{n1}) & \dots & Q_{11}(A_{11}^{N_x-1}B_{1m} + \dots + A_{1n}^{N_x-1}B_{nm}) \\ Q_{22}(A_{21}^{N_x-1}B_{11} + \dots + A_{2n}^{N_x-1}B_{n1}) & \dots & Q_{22}(A_{21}^{N_x-1}B_{1m} + \dots + A_{2n}^{N_x-1}B_{nm}) \\ \vdots & \dots & \vdots \\ Q_{nn}(A_{n1}^{N_x-1}B_{11} + \dots + A_{nn}^{N_x-1}B_{n1}) & \dots & Q_{nn}(A_{n1}^{N_x-1}B_{1m} + \dots + A_{nn}^{N_x-1}B_{nm}) \end{pmatrix} \in \mathbb{R}^{n \times m}$$

$$Z_{11} = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix} \in \mathbb{R}^{n \times m},$$

$$E_2 = \begin{pmatrix} E_{21} & Z_{21} & \dots & Z_{21} \\ Z_{21} & E_{21} & & \vdots \\ \vdots & & \ddots & Z_{21} \\ Z_{21} & \dots & Z_{21} & E_{21} \end{pmatrix} \in \mathbb{R}^{(m \cdot N_u) \times (m \cdot N_u)} \quad (5.20)$$

mit

$$E_{21} = R = \begin{pmatrix} R_{11} & 0 & \dots & 0 \\ 0 & R_{22} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & R_{mm} \end{pmatrix} \in \mathbb{R}^{m \times m}$$

$$Z_{21} = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix} \in \mathbb{R}^{m \times m},$$

$$E_3 = \begin{pmatrix} -1 & 0 & \dots & 0 \\ 0 & -1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & -1 \end{pmatrix} \in \mathbb{R}^{(n \cdot N_x) \times (n \cdot N_x)}, \quad (5.21)$$

$$E_4 = \begin{pmatrix} -1 & 0 & \dots & 0 \\ 0 & -1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & -1 \end{pmatrix} \in \mathbb{R}^{(m \cdot N_u) \times (m \cdot N_u)}, \quad (5.22)$$

$$Z_1 = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix} \in \mathbb{R}^{(m \cdot N_u) \times (n \cdot N_x)}, \quad (5.23)$$

$$\text{und } Z_2 = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix} \in \mathbb{R}^{(n \cdot N_x) \times (m \cdot N_u)}. \quad (5.24)$$

Weiterhin ist der Vektor d gegeben als

$$\mathbf{d} = \begin{pmatrix} d_1 \\ -d_1 \\ d_2 \\ d_2 \end{pmatrix} \in \mathbb{R}^{2 \cdot n \cdot N_x + 2 \cdot m \cdot N_u}, \quad (5.25)$$

wobei

$$d_1 = \begin{pmatrix} d_{11} \\ d_{12} \\ \vdots \\ \vdots \\ d_{1N_x} \end{pmatrix} \in \mathbb{R}^{n \cdot N_x} \quad (5.26)$$

mit

$$d_{11} = \begin{pmatrix} -Q_{11} [AA^{t_s-t_{akt}}x(t_{akt})]_1 + Q_{11}x_{ref,1}(N_1) \\ -Q_{22} [AA^{t_s-t_{akt}}x(t_{akt})]_2 + Q_{22}x_{ref,2}(N_1) \\ \vdots \\ -Q_{nn} [AA^{t_s-t_{akt}}x(t_{akt})]_n + Q_{nn}x_{ref,n}(N_1) \end{pmatrix} \in \mathbb{R}^n$$

$$d_{12} = \begin{pmatrix} -Q_{11} [A^2A^{t_s-t_{akt}}x(t_{akt})]_1 + Q_{11}x_{ref,1}(N_1+1) \\ -Q_{22} [A^2A^{t_s-t_{akt}}x(t_{akt})]_2 + Q_{22}x_{ref,2}(N_1+1) \\ \vdots \\ -Q_{nn} [A^2A^{t_s-t_{akt}}x(t_{akt})]_n + Q_{nn}x_{ref,n}(N_1+1) \end{pmatrix} \in \mathbb{R}^n$$

$$d_{1N_x} = \begin{pmatrix} -Q_{11} [A^{N_x} A^{t_s - t_{akt}} x(t_{akt})]_1 + Q_{11} x_{ref,1}(N_2) \\ -Q_{22} [A^{N_x} A^{t_s - t_{akt}} x(t_{akt})]_2 + Q_{22} x_{ref,2}(N_2) \\ \vdots \\ -Q_{nn} [A^{N_x} A^{t_s - t_{akt}} x(t_{akt})]_n + Q_{nn} x_{ref,n}(N_2) \end{pmatrix} \in \mathbb{R}^n,$$

$$\text{und } d_2 = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{m \cdot N_u}. \quad (5.27)$$

Bemerkung 5.4. Die formal etwas kompliziert wirkenden Matrizen E_{12} bis E_{1N_x} lassen sich im später verwendeten Programm zur modellprädiktiven Regelung mittels linearer Programmierung relativ einfach berechnen. Die Einträge dieser Matrizen ergeben sich gerade aus

$$Q_{ii} \left(\sum_{k=1}^n A_{ik}^l B_{kj} \right)$$

für $i = 1, \dots, n$, $j = 1, \dots, m$ und $l = 1, \dots, N_x - 1$. Es ist dabei zu beachten, dass die Parameter i und j zur Konstruktion von $E_{1,l+1}$ richtig durchlaufen werden. Diese beschreiben nämlich zugleich den Zeilen- und Spaltenindex der Matrizen.

5.2 Anwendung des MPC-Konzeptes

Wie bereits im vorherigen Abschnitt erwähnt, erhält man durch die Lösung des linearen Programms die optimale Kontrollfolge $u = [u^T(N_3) \ u^T(N_3 + 1) \ \dots \ u^T(N_4)]^T$ auf dem Kontrollhorizont. Dies entspricht gerade dem ersten Schritt der modellprädiktiven Regelung. Die weitere Vorgehensweise der MPC-Strategie beruht nun auf dem „receding-horizon“ Prinzip. Im nächsten Schritt wird das erste Glied $u(t_s)$ (entspricht $u(N_3)$) der optimalen Kontrollfolge auf das System angewendet. Anschließend wird der neu berechnete Zustand als Anfangszustand verwendet und die Lösung des linearen Programms mit verschobenem Horizont wiederholt. Formal ausgedrückt entsprechen die Schritte der modellprädiktiven Regelung:

1. Setze $t = t_s$ (bzw. $t = N_3$)
2. Berechne die optimale Kontrollfolge auf dem Kontrollhorizont durch Lösen des linearen Programms
3. Wende das erste Glied $u(t)$ auf das System an,
d.h. berechne $x(t+1) = Ax(t) + Bu(t)$ mit $x(t) = A^{(t_s - t_{akt})} x(t_{akt})$

4. Setze $x(t+1) = x(t_{akt})$
5. Setze $t = t + 1$
6. Wiederhole Schritt 2. bis 5. solange $t \leq t_s + N_x - 1$ (bzw. $t \leq N_2 - 1$)
7. Wenn $t > t_s + N_x - 1$ (bzw. $t > N_2 - 1$) \implies Ende

Mittels dieses Algorithmus erhält man schließlich die optimale Kontrollfolge $u = [u(N_3) \ u(N_3+1) \ \dots \ u(N_2-1)]$. Durch dessen Anwendung auf das System wird erreicht, dass der Zustand $x(t)$ für $t = N_1, \dots, N_2$ (d.h. auf dem Prädiktionshorizont) dem Referenzzustand $x_{ref}(t)$ folgt. In Kapitel 6 wird der Referenzzustand $x_{ref}(t)$ für alle $t \in \mathbb{Z}$ gleich Null gesetzt, was das Ziel der Regelung darauf beschränkt, den Zustand in die Null zu steuern und dort zu halten. Diese Annahme vereinfacht sowohl das gesamte Problem als auch das lineare Programm. Außerdem ist die Kenntnis über den Referenzzustand zu den Zeiten $t = N_1, \dots, N_2$ nicht mehr notwendig.

Die folgenden Abbildungen veranschaulichen die oben angegebene Vorgehensweise: Abbildung 5.3 entspricht gerade dem Ergebnis der Lösung des linearen Programms. Die weiteren Schritte und damit die Durchführung des modellprädiktiven Regelungsansatzes veranschaulichen die Abbildungen 5.4 und 5.5. Die letztendlich berechnete optimale Kontrollfolge sowie das optimale Verhalten des Zustandes zeigen Abbildung 5.6 und 5.7.

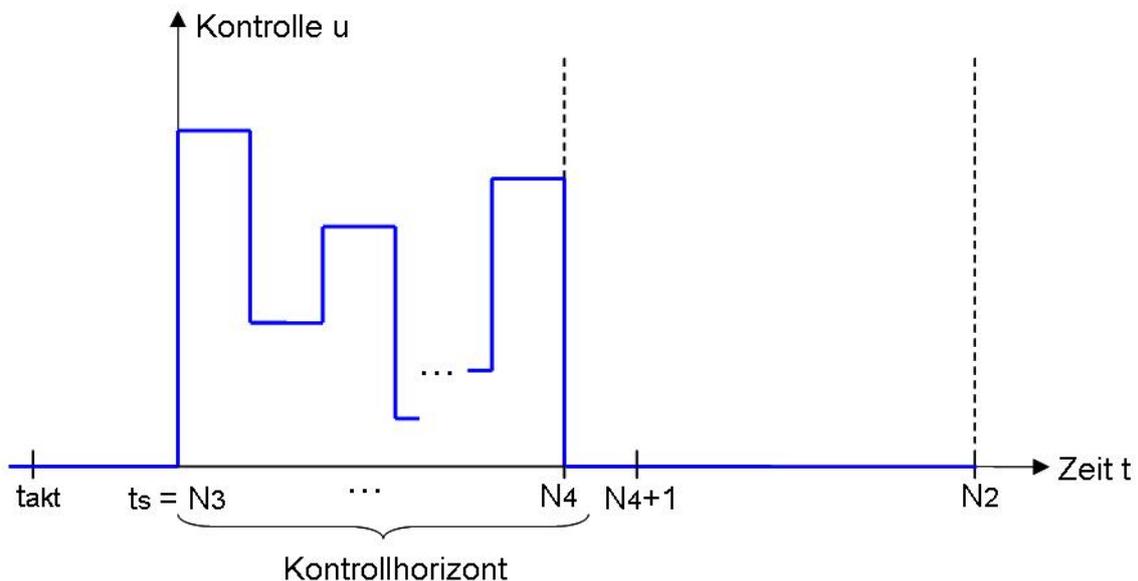


Abbildung 5.3: Schritt 1. und 2. des modellprädiktiven Regelungsansatzes

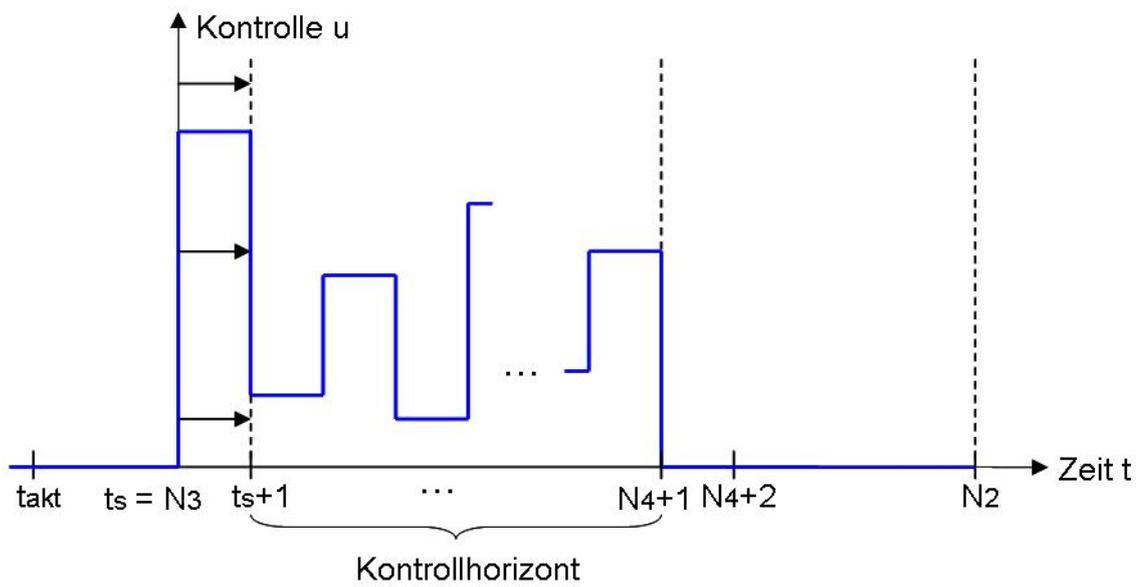


Abbildung 5.4: Schritt 3. bis 5. und Wiederholung von Schritt 2.

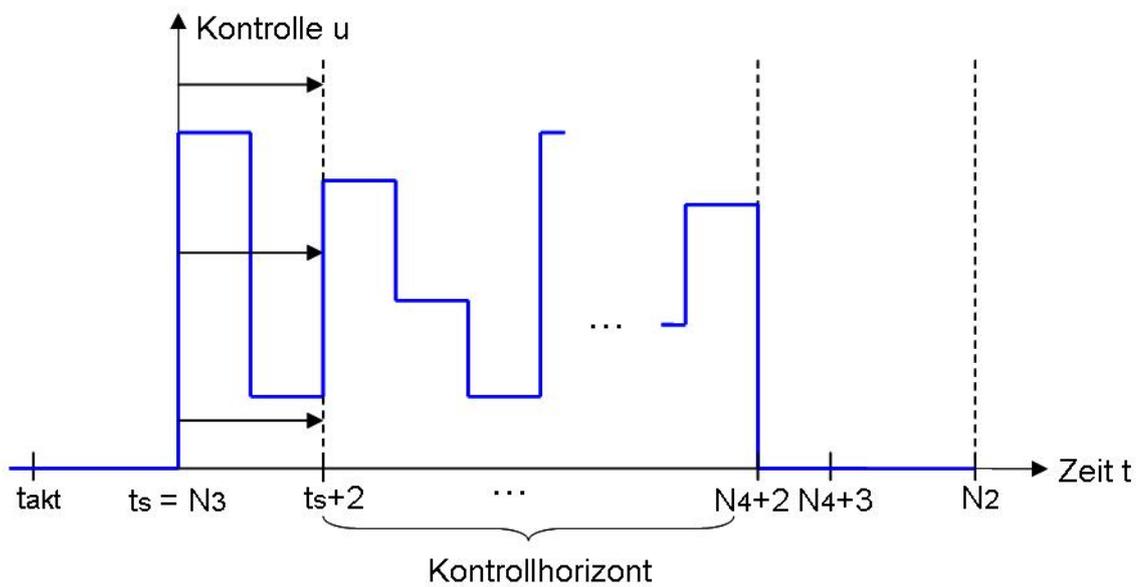
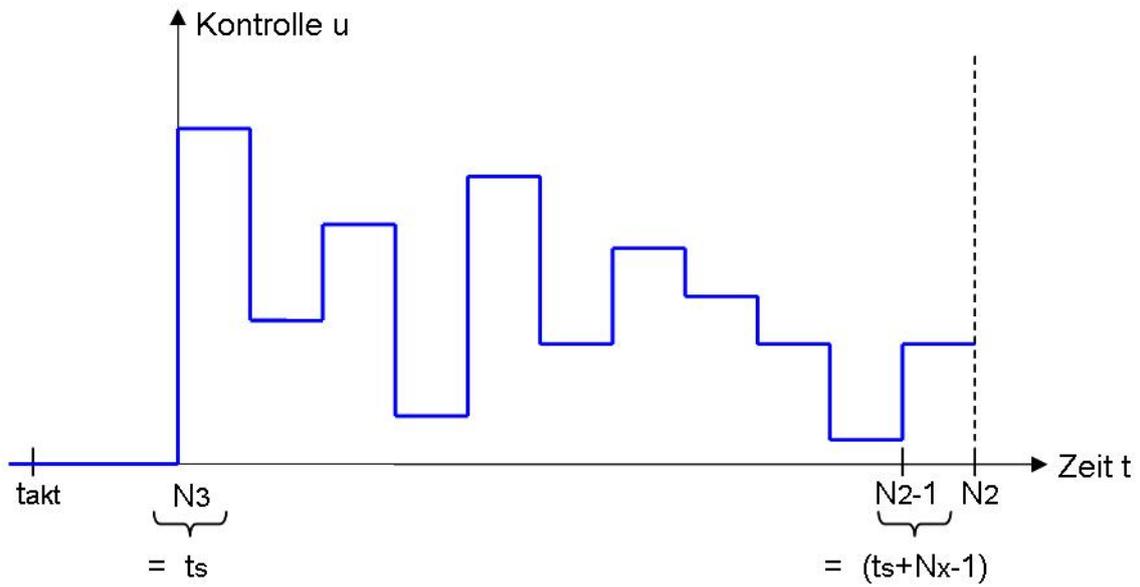
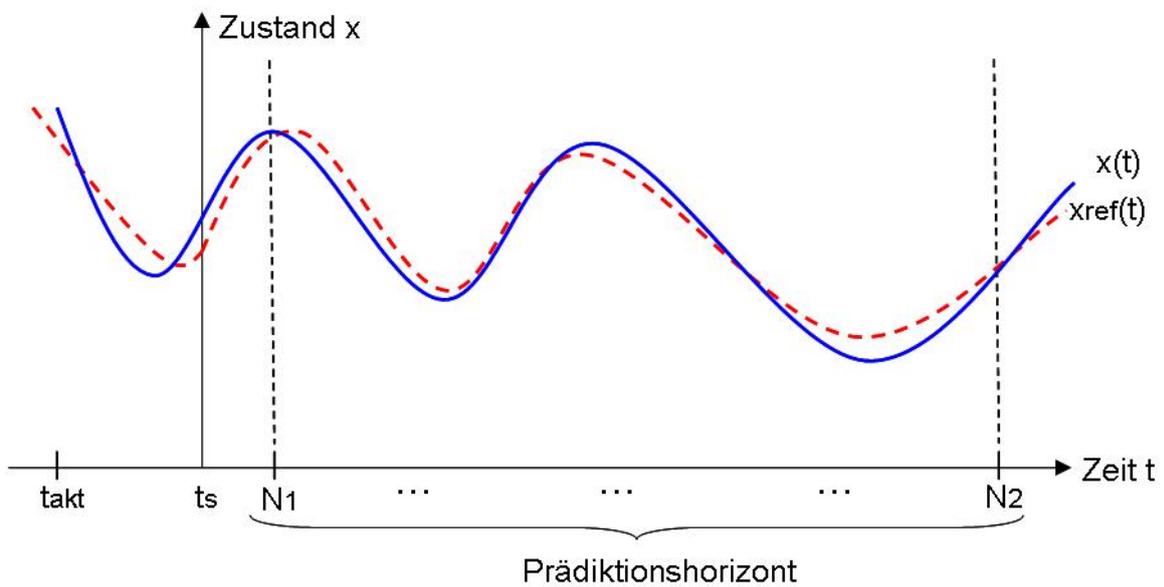


Abbildung 5.5: Weitere Wiederholung des „receding-horizon“ Konzeptes

Abbildung 5.6: Optimale Kontrollfolge $u(t)$ Abbildung 5.7: Optimales Verhalten des Zustandes $x(t)$

5.3 Kontrollsysteme mit Ausgang

Das der modellprädiktiven Regelung zugrunde liegende Optimierungsproblem sowie die Herleitung des linearen Programms für Kontrollsysteme mit Ausgang sind zum großen Teil analog zu Abschnitt 5.1. In diesem Teil sollen daher nur die wesentlichen Unterschiede angegeben werden. Abschließend wird das lineare Programm für Kontrollsysteme mit Ausgang formuliert.

Dem folgenden Abschnitt liegt ein lineares zeitdiskretes Kontrollsystem mit Ausgang

$$\begin{aligned}x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t)\end{aligned}\tag{5.28}$$

zugrunde. Dabei bezeichnet $t \in \mathbb{Z}$ wieder die Zeit, $x \in \mathbb{R}^n$ den Zustand und $u \in \mathbb{R}^m$ die Kontrolle. Da man in der Praxis nicht davon ausgehen kann, dass der gesamte Zustandsvektor $x(t)$ zur Verfügung steht, werden hier außerdem gewisse von $x(t)$ abhängige Werte $y(t) = Cx(t)$ betrachtet. Dieser Vektor $y \in \mathbb{R}^l$ wird Ausgang genannt. Für die Matrizen gilt $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ und $C \in \mathbb{R}^{l \times n}$.

Das Ziel von MPC ist hier den Ausgang $y(t)$ in den Referenzausgang $y_{ref}(t)$ zu regeln und dort zu halten. Es soll dazu wiederum das der Regelung zugrunde liegende Optimierungsproblem mittels linearer Programmierung gelöst werden.

5.3.1 Herleitung des äquivalenten Optimierungsproblems

Das zu (5.2) analoge Optimierungsproblem lautet bei Betrachtung eines linearen Kontrollsystems mit Ausgang

$$\min_{u:=[u^T(N_3) \dots u^T(N_4)]^T} \left\{ J(u, y(t)) = \sum_{t=N_1}^{N_2} Q \|y(t) - y_{ref}(t)\|_1 + \sum_{t=N_3}^{N_4} R \|u(t)\|_1 \right\}\tag{5.29}$$

s.t.

$$\begin{aligned}x(t+1) &= Ax(t) + Bu(t), & t &= N_1 - 1, \dots, N_2 - 1 \\ y(t) &= Cx(t), & t &= N_1, \dots, N_2 \\ u(t) &= 0, & N_4 + 1 &\leq t \leq N_2 - 1.\end{aligned}$$

Dabei gilt

N_y ist die Länge des Prädiktionshorizontes, auf welchem der Ausgang y dem Referenzausgang $y_{ref} \in \mathbb{R}^l$ – in den meisten Fällen $y_{ref} = 0$ – folgen soll,

$Q \in \mathbb{R}^{l \times l}$ und $R \in \mathbb{R}^{m \times m}$.

Die Ausdrücke der Zielfunktion $J(u, y(t))$ können für dieses Problem mit Hilfe von Definition 5.1. wie folgt formuliert werden:

$$\begin{aligned} \|y(t) - y_{ref}(t)\|_1 &= \sum_{k=1}^l |y_k(t) - x_{ref,k}(t)| \\ &= |y_1(t) - y_{ref,1}(t)| + |y_2(t) - y_{ref,2}(t)| + \dots + |y_l(t) - y_{ref,l}(t)| \\ \|u(t)\|_1 &= \sum_{j=1}^m |u_j(t)| = |u_1(t)| + |u_2(t)| + \dots + |u_m(t)| \end{aligned}$$

y_k ist dabei der k -te Eintrag des Vektors $y \in \mathbb{R}^l$, $y_{ref,k}$ der k -te Eintrag des Vektors $y_{ref} \in \mathbb{R}^l$ sowie u_j der j -te Eintrag des Vektors $u \in \mathbb{R}^m$. Daraus erhält man die Zielfunktion

$$J(u, y(t)) = \sum_{t=N_1}^{N_2} \sum_{k=1}^l Q_{kk} |y_k(t) - y_{ref,k}(t)| + \sum_{t=N_3}^{N_4} \sum_{j=1}^m R_{jj} |u_j(t)|, \quad (5.30)$$

wobei Q_{kk} den Eintrag der k -ten Zeile und k -ten Spalte der Gewichtungsmatrix $Q \in \mathbb{R}^{l \times l}$ sowie R_{jj} der Eintrag der j -ten Zeile und j -ten Spalte der Matrix $R \in \mathbb{R}^{m \times m}$ bezeichnet.

Durch Einführen der Schlupfvariablen $\gamma_{k,t}$ und $\delta_{j,t}$ kann das zu (5.6) analoge „äquivalente Optimierungsproblem“ wie folgt definiert werden.

$$\min_{[u \ \gamma \ \delta]^T} \left\{ \sum_{t=N_1}^{N_2} \sum_{k=1}^l \gamma_{k,t} + \sum_{t=N_3}^{N_4} \sum_{j=1}^m \delta_{j,t} \right\} \quad (5.31)$$

s.t.

$$\begin{aligned} -\gamma_{k,t} &\leq Q_{kk} [y_k(t) - y_{ref,k}(t)] \leq \gamma_{k,t}, & k &= 1, \dots, l \\ & & t &= N_1, \dots, N_2 \\ -\delta_{j,t} &\leq R_{jj} u_j(t) \leq \delta_{j,t}, & j &= 1, \dots, m \\ & & t &= N_3, \dots, N_4 \\ x(t+1) &= Ax(t) + Bu(t), & t &= N_1 - 1, \dots, N_2 - 1 \\ y(t) &= Cx(t), & t &= N_1, \dots, N_2 \\ u(t) &= 0, & N_4 + 1 &\leq t \leq N_2 - 1, \end{aligned}$$

wobei

$$\begin{aligned} u &= [u^T(N_3) \ u^T(N_3 + 1) \ \dots \ u^T(N_4)] = \\ &= [u_1(N_3) \ \dots \ u_m(N_3) \ u_1(N_3 + 1) \ \dots \ u_m(N_3 + 1) \ \dots \ u_1(N_4) \ \dots \ u_m(N_4)], \end{aligned} \quad (5.32)$$

$$\gamma = [\gamma_{1,N_1} \ \dots \ \gamma_{l,N_1} \ \gamma_{1,(N_1+1)} \ \dots \ \gamma_{l,(N_1+1)} \ \dots \ \gamma_{1,N_2} \ \dots \ \gamma_{l,N_2}] \text{ und} \quad (5.33)$$

$$\delta = [\delta_{1,N_3} \ \dots \ \delta_{m,N_3} \ \delta_{1,(N_3+1)} \ \dots \ \delta_{m,(N_3+1)} \ \dots \ \delta_{1,N_4} \ \dots \ \delta_{m,N_4}]. \quad (5.34)$$

Berechnung des Ausgangs $y(t)$ für alle $t > t_s$:

Zur Lösung des Optimierungsproblems (5.31) ist es von Vorteil den Ausgang $y(t)$ für alle $t > t_s$ im Voraus zu berechnen. Da es sich bei $y(t)$ um einen vom Zustand $x(t)$ abhängigen Vektor handelt, können die Gleichungen (5.10) und (5.11) verwendet werden. Daraus folgt mit $y(t) = Cx(t)$

$$y(t) = Cx(t) = C \left(A^{(t-t_s)}x(t_s) + \sum_{r=0}^{t-t_s-1} A^r B u(t-1-r) \right) \quad (5.35)$$

mit

$$x(t_s) = A^{(t_s-t_{akt})}x(t_{akt}), \quad \text{falls } t_s > t_{akt}. \quad (5.36)$$

Das zu (5.29) äquivalente lineare Programm kann somit unter Verwendung der Gleichungen (5.35), (5.36) wie folgt formuliert werden:

$$\min_{[u \ \gamma \ \delta]^T} \left\{ \sum_{t=N_1}^{N_2} \sum_{k=1}^l \gamma_{k,t} + \sum_{t=N_3}^{N_4} \sum_{j=1}^m \delta_{j,t} \right\} \quad (5.37)$$

s.t.

$$-\gamma_{k,t} \leq Q_{kk} \left(\left[C \left(A^{(t-t_s)} A^{(t_s-t_{akt})} x(t_{akt}) + \sum_{r=0}^{t-t_s-1} A^r B u(t-1-r) \right) \right]_k - y_{ref,k}(t) \right) \leq \gamma_{k,t},$$

$$k = 1, \dots, l$$

$$t = N_1, \dots, N_2$$

$$-\delta_{j,t} \leq R_{jj} u_j(t) \leq \delta_{j,t}, \quad j = 1, \dots, m$$

$$t = N_3, \dots, N_4$$

$$u(t) = 0, \quad N_4 + 1 \leq t \leq N_2 - 1$$

mit (5.32), (5.33) und (5.34).

Davon ausgehend kann im nächsten Abschnitt das lineare Programm in Matrizenform aufgestellt werden.

5.3.2 Das lineare Programm in Matrizenform

Die Herleitung des linearen Programms für Kontrollsysteme mit Ausgang verläuft größtenteils analog zu 5.1.4. Um die Unterschiede aufzuzeigen, soll im Folgenden dasselbe Beispiel

verwendet werden. Dabei gilt zusätzlich $l = 2$, woraus $C \in \mathbb{R}^{3 \times 2}$ und $Q \in \mathbb{R}^{2 \times 2}$ folgt. Das Optimierungsproblem (5.31) lautet

$$\min_{[u \ \gamma \ \delta]^T} \quad \{(\gamma_{1,3} + \gamma_{2,3} + \gamma_{1,4} + \gamma_{2,4} + \gamma_{1,5} + \gamma_{2,5}) + \\ + (\delta_{1,2} + \delta_{2,2} + \delta_{1,3} + \delta_{2,3})\}$$

s.t.

$$\begin{aligned} -\gamma_{1,3} &\leq Q_{11} [y_1(3) - y_{ref,1}(3)] \leq \gamma_{1,3} \\ -\gamma_{2,3} &\leq Q_{22} [y_2(3) - y_{ref,2}(3)] \leq \gamma_{2,3} \\ -\gamma_{1,4} &\leq Q_{11} [y_1(4) - y_{ref,1}(4)] \leq \gamma_{1,4} \\ -\gamma_{2,4} &\leq Q_{22} [y_2(4) - y_{ref,2}(4)] \leq \gamma_{2,4} \\ -\gamma_{1,5} &\leq Q_{11} [y_1(5) - y_{ref,1}(5)] \leq \gamma_{1,5} \\ -\gamma_{2,5} &\leq Q_{22} [y_2(5) - y_{ref,2}(5)] \leq \gamma_{2,5} \\ -\delta_{1,2} &\leq R_{11} u_1(2) \leq \delta_{1,2} \\ -\delta_{2,2} &\leq R_{22} u_2(2) \leq \delta_{2,2} \\ -\delta_{1,3} &\leq R_{11} u_1(3) \leq \delta_{1,3} \\ -\delta_{2,3} &\leq R_{22} u_2(3) \leq \delta_{2,3} \\ x(3) &= Ax(2) + Bu(2) \\ x(4) &= Ax(3) + Bu(3) \\ x(5) &= Ax(4) + Bu(4) \\ y(3) &= Cx(3) \\ y(4) &= Cx(4) \\ y(5) &= Cx(5) \\ u(4) &= 0. \end{aligned}$$

Nach rekursiver Berechnung der Ausgänge $y(t)$ erhält man für die Nebenbedingungen

$$\begin{aligned} -\gamma_{1,3} \pm Q_{11}[CAA^2x(0)]_1 \pm Q_{11}[CBu(2)]_1 \mp Q_{11}y_{ref,1}(3) &\leq 0 \\ -\gamma_{2,3} \pm Q_{22}[CAA^2x(0)]_2 \pm Q_{22}[CBu(2)]_2 \mp Q_{22}y_{ref,2}(3) &\leq 0 \\ -\gamma_{1,4} \pm Q_{11}[CA^2A^2x(0)]_1 \pm Q_{11}[CBu(3)]_1 \pm Q_{11}[CABu(2)]_1 \mp Q_{11}y_{ref,1}(4) &\leq 0 \\ -\gamma_{2,4} \pm Q_{22}[CA^2A^2x(0)]_2 \pm Q_{22}[CBu(3)]_2 \pm Q_{22}[CABu(2)]_2 \mp Q_{22}y_{ref,2}(4) &\leq 0 \\ -\gamma_{1,5} \pm Q_{11}[CA^3A^2x(0)]_1 \pm \underbrace{Q_{11}[CBu(4)]_1}_{=0} \pm Q_{11}[CABu(3)]_1 \pm Q_{11}[CA^2Bu(2)]_1 \mp Q_{11}y_{ref,1}(5) &\leq 0 \\ -\gamma_{2,5} \pm Q_{22}[CA^3A^2x(0)]_2 \pm \underbrace{Q_{22}[CBu(4)]_2}_{=0} \pm Q_{22}[CABu(3)]_2 \pm Q_{22}[CA^2Bu(2)]_2 \mp Q_{22}y_{ref,2}(5) &\leq 0 \\ -\delta_{1,2} \pm R_{11}u_1(2) &\leq 0 \\ -\delta_{2,2} \pm R_{22}u_2(2) &\leq 0 \\ -\delta_{1,3} \pm R_{11}u_1(3) &\leq 0 \\ -\delta_{2,3} \pm R_{22}u_2(3) &\leq 0 \\ u(4) &= 0. \end{aligned}$$

Der wesentliche Unterschied besteht nun darin, dass hier zur Berechnung der Ausdrücke $[\cdot u(t)]_k$ die folgenden Gleichungen verwendet werden. Aus

$$\begin{aligned}
 CBu(t) &= \underbrace{\begin{pmatrix} C_{11} & C_{12} & \dots & C_{1n} \\ C_{21} & C_{22} & \dots & C_{2n} \\ \vdots & & & \vdots \\ C_{l1} & C_{l2} & \dots & C_{ln} \end{pmatrix}}_{=C} \underbrace{\begin{pmatrix} B_{11}u_1(t) + B_{12}u_2(t) + \dots + B_{1m}u_m(t) \\ B_{21}u_1(t) + B_{22}u_2(t) + \dots + B_{2m}u_m(t) \\ \vdots \\ B_{n1}u_1(t) + B_{n2}u_2(t) + \dots + B_{nm}u_m(t) \end{pmatrix}}_{=Bu(t)} \\
 &= \begin{pmatrix} C_{11}(B_{11}u_1(t) + \dots + B_{1m}u_m(t)) + \dots + C_{1n}(B_{n1}u_1(t) + \dots + B_{nm}u_m(t)) \\ \vdots \\ C_{l1}(B_{11}u_1(t) + \dots + B_{1m}u_m(t)) + \dots + C_{ln}(B_{n1}u_1(t) + \dots + B_{nm}u_m(t)) \end{pmatrix}
 \end{aligned}$$

und

$$\begin{aligned}
 CA^k Bu(t) &= \underbrace{\begin{pmatrix} C_{11} & C_{12} & \dots & C_{1n} \\ C_{21} & C_{22} & \dots & C_{2n} \\ \vdots & & & \vdots \\ C_{l1} & C_{l2} & \dots & C_{ln} \end{pmatrix}}_{=C} \cdot \underbrace{\begin{pmatrix} A_{11}^k(B_{11}u_1(t) + \dots + B_{1m}u_m(t)) + \dots + A_{1n}^k(B_{n1}u_1(t) + \dots + B_{nm}u_m(t)) \\ \vdots \\ A_{n1}^k(B_{11}u_1(t) + \dots + B_{1m}u_m(t)) + \dots + A_{nn}^k(B_{n1}u_1(t) + \dots + B_{nm}u_m(t)) \end{pmatrix}}_{=A^k Bu(t)} \\
 &= \begin{pmatrix} C_{11}(A_{11}^k(B_{11}u_1(t) + \dots + B_{1m}u_m(t)) + \dots + A_{1n}^k(B_{n1}u_1(t) + \dots + B_{nm}u_m(t))) + \dots \\ \dots + C_{1n}(A_{n1}^k(B_{11}u_1(t) + \dots + B_{1m}u_m(t)) + \dots + A_{nn}^k(B_{n1}u_1(t) + \dots + B_{nm}u_m(t))) \\ \vdots \\ C_{l1}(A_{11}^k(B_{11}u_1(t) + \dots + B_{1m}u_m(t)) + \dots + A_{1n}^k(B_{n1}u_1(t) + \dots + B_{nm}u_m(t))) + \dots \\ \dots + C_{ln}(A_{n1}^k(B_{11}u_1(t) + \dots + B_{1m}u_m(t)) + \dots + A_{nn}^k(B_{n1}u_1(t) + \dots + B_{nm}u_m(t))) \end{pmatrix}
 \end{aligned}$$

folgen die zu (5.14), (5.15) analogen Gleichungen

$$[CBu(t)]_i = \sum_{j_2=1}^n C_{ij_2} [Bu(t)]_{j_2} = C_{i1} [Bu(t)]_1 + \dots + C_{in} [Bu(t)]_n \quad (5.38)$$

sowie

$$\begin{aligned}
[CA^k Bu(t)]_i &= \sum_{j_3=1}^n C_{ij_3} [A^k Bu(t)]_{j_3} = C_{i1} [A^k Bu(t)]_1 + \dots + C_{in} [A^k Bu(t)]_n \quad \implies \\
[CA^k Bu(t)]_i &= \sum_{j_3=1}^n \left(C_{ij_3} \sum_{j_2=1}^n \left(A_{j_3 j_2}^k \sum_{j_1=1}^m B_{j_2 j_1} u_{j_1}(t) \right) \right) \quad (5.39) \\
&= C_{i1} (A_{11}^k (B_{11} u_1(t) + \dots + B_{1m} u_m(t)) + \dots + A_{1n}^k (B_{n1} u_1(t) + \dots + B_{nm} u_m(t))) + \\
&+ \dots + \\
&+ C_{in} (A_{n1}^k (B_{11} u_1(t) + \dots + B_{1m} u_m(t)) + \dots + A_{nn}^k (B_{n1} u_1(t) + \dots + B_{nm} u_m(t)))
\end{aligned}$$

Davon ausgehend kann das lineare Programm der Form (5.13) für Kontrollsysteme mit Ausgang definiert werden:

$$\begin{aligned}
&\min_z f \cdot z \\
&\text{s.t. } E \cdot z - d \leq 0
\end{aligned}$$

Dabei sind f und z gegeben durch

$$\mathbf{f} = \left[\underbrace{0 \dots 0}_{m \cdot N_u} \quad \underbrace{1 \dots 1}_{l \cdot N_y} \quad \underbrace{1 \dots 1}_{m \cdot N_u} \right] \in \mathbb{R}^{1 \times (2 \cdot m \cdot N_u + l \cdot N_y)} \quad (5.40)$$

$$\mathbf{z} = [u \quad \gamma \quad \delta]^T \in \mathbb{R}^{2 \cdot m \cdot N_u + l \cdot N_y} \quad (5.41)$$

mit

$$\begin{aligned}
u &= [u_1(N_3) \dots u_m(N_3) \quad u_1(N_3+1) \dots u_m(N_3+1) \dots \dots u_1(N_4) \dots u_m(N_4)] \in \mathbb{R}^{1 \times (m \cdot N_u)} \\
\gamma &= [\gamma_{1,N_1} \dots \gamma_{l,N_1} \quad \gamma_{1,(N_1+1)} \dots \gamma_{l,(N_1+1)} \dots \dots \gamma_{1,N_2} \dots \gamma_{l,N_2}] \in \mathbb{R}^{1 \times (l \cdot N_y)} \\
\delta &= [\delta_{1,N_3} \dots \delta_{m,N_3} \quad \delta_{1,(N_3+1)} \dots \delta_{m,(N_3+1)} \dots \dots \delta_{1,N_4} \dots \delta_{m,N_4}] \in \mathbb{R}^{1 \times (m \cdot N_u)}.
\end{aligned}$$

Die Matrix E ist definiert als

$$\mathbf{E} = \begin{pmatrix} E_1 & E_3 & Z_2 \\ -E_1 & E_3 & Z_2 \\ E_2 & Z_1 & E_4 \\ -E_2 & Z_1 & E_4 \end{pmatrix} \in \mathbb{R}^{2(l \cdot N_y + m \cdot N_u) \times (2 \cdot m \cdot N_u + l \cdot N_y)}, \quad (5.42)$$

wobei

$$E_1 = \begin{pmatrix} E_{11} & Z_{11} & \dots & Z_{11} \\ E_{12} & E_{11} & & \vdots \\ \vdots & \ddots & \ddots & Z_{11} \\ E_{1N_u} & \dots & E_{12} & E_{11} \\ \hline E_{1(N_u+1)} & E_{1N_u} & \dots & E_{12} \\ \vdots & \ddots & \ddots & \vdots \\ E_{1(N_y)} & \dots & E_{1(N_u+1)} & E_{1N_u} \end{pmatrix} \in \mathbb{R}^{(l \cdot N_u + l \cdot (N_y - N_u)) \times (m \cdot N_u)} = (l \cdot N_y) \times (m \cdot N_u) \quad (5.43)$$

mit den nun veränderten Teilmatrizen

$$E_{11} = \begin{pmatrix} Q_{11}(C_{11}B_{11} + C_{12}B_{21} + \dots + C_{1n}B_{n1}) & \dots & Q_{11}(C_{11}B_{1m} + C_{12}B_{2m} + \dots + C_{1n}B_{nm}) \\ Q_{22}(C_{21}B_{11} + C_{22}B_{21} + \dots + C_{2n}B_{n1}) & \dots & Q_{22}(C_{21}B_{1m} + C_{22}B_{2m} + \dots + C_{2n}B_{nm}) \\ \vdots & & \vdots \\ Q_{ll}(C_{l1}B_{11} + C_{l2}B_{21} + \dots + C_{ln}B_{n1}) & \dots & Q_{ll}(C_{l1}B_{1m} + C_{l2}B_{2m} + \dots + C_{ln}B_{nm}) \end{pmatrix} \in \mathbb{R}^{l \times m}$$

$$E_{12} = \begin{pmatrix} Q_{11}(C_{11}A_{11}B_{11} + \dots + C_{11}A_{1n}B_{n1} + C_{12}A_{21}B_{11} + \dots + C_{12}A_{2n}B_{n1} + \dots + C_{1n}A_{n1}B_{11} + \dots + C_{1n}A_{nn}B_{n1}) & \dots & Q_{11}(C_{11}A_{11}B_{1m} + \dots + C_{11}A_{1n}B_{nm} + C_{12}A_{21}B_{1m} + \dots + C_{12}A_{2n}B_{nm} + \dots + C_{1n}A_{n1}B_{1m} + \dots + C_{1n}A_{nn}B_{nm}) \\ \vdots & & \vdots \\ Q_{ll}(C_{l1}A_{l1}B_{11} + \dots + C_{l1}A_{ln}B_{n1} + C_{l2}A_{21}B_{11} + \dots + C_{l2}A_{2n}B_{n1} + \dots + C_{ln}A_{n1}B_{11} + \dots + C_{ln}A_{nn}B_{n1}) & \dots & Q_{ll}(C_{l1}A_{l1}B_{1m} + \dots + C_{l1}A_{ln}B_{lm} + C_{l2}A_{21}B_{1m} + \dots + C_{l2}A_{2n}B_{nm} + \dots + C_{ln}A_{n1}B_{1m} + \dots + C_{ln}A_{nn}B_{nm}) \end{pmatrix} \in \mathbb{R}^{l \times m}$$

⋮

$$Z_{11} = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix} \in \mathbb{R}^{l \times m},$$

$$E_2 = \begin{pmatrix} E_{21} & Z_{21} & \dots & Z_{21} \\ Z_{21} & E_{21} & & \vdots \\ \vdots & & \ddots & Z_{21} \\ Z_{21} & \dots & Z_{21} & E_{21} \end{pmatrix} \in \mathbb{R}^{(m \cdot N_u) \times (m \cdot N_u)} \quad (5.44)$$

mit

$$E_{21} = R = \begin{pmatrix} R_{11} & 0 & \dots & 0 \\ 0 & R_{22} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & R_{mm} \end{pmatrix} \in \mathbb{R}^{m \times m}$$

$$Z_{21} = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix} \in \mathbb{R}^{m \times m},$$

$$E_3 = \begin{pmatrix} -1 & 0 & \dots & 0 \\ 0 & -1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & -1 \end{pmatrix} \in \mathbb{R}^{(l \cdot N_y) \times (l \cdot N_y)}, \quad (5.45)$$

$$E_4 = \begin{pmatrix} -1 & 0 & \dots & 0 \\ 0 & -1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & -1 \end{pmatrix} \in \mathbb{R}^{(m \cdot N_u) \times (m \cdot N_u)}, \quad (5.46)$$

$$Z_1 = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix} \in \mathbb{R}^{(m \cdot N_u) \times (l \cdot N_y)}, \quad (5.47)$$

$$\text{und } Z_2 = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix} \in \mathbb{R}^{(l \cdot N_y) \times (m \cdot N_u)}. \quad (5.48)$$

Außerdem ist der Vektor d definiert als

$$\mathbf{d} = \begin{pmatrix} d_1 \\ -d_1 \\ d_2 \\ d_2 \end{pmatrix}, \quad \in \mathbb{R}^{2 \cdot l \cdot N_y + 2 \cdot m \cdot N_u}, \quad (5.49)$$

wobei

$$d_1 = \begin{pmatrix} d_{11} \\ d_{12} \\ \vdots \\ \vdots \\ d_{1N_y} \end{pmatrix} \in \mathbb{R}^{l \cdot N_y} \quad (5.50)$$

mit

$$d_{11} = \begin{pmatrix} -Q_{11} [CAA^{t_s - t_{akt}} x(t_{akt})]_1 + Q_{11} y_{ref,1}(N_1) \\ -Q_{22} [CAA^{t_s - t_{akt}} x(t_{akt})]_2 + Q_{22} y_{ref,2}(N_1) \\ \vdots \\ -Q_{ll} [CAA^{t_s - t_{akt}} x(t_{akt})]_l + Q_{ll} y_{ref,l}(N_1) \end{pmatrix} \in \mathbb{R}^l$$

$$d_{12} = \begin{pmatrix} -Q_{11} [CA^2 A^{t_s - t_{akt}} x(t_{akt})]_1 + Q_{11} y_{ref,1}(N_1 + 1) \\ -Q_{22} [CA^2 A^{t_s - t_{akt}} x(t_{akt})]_2 + Q_{22} y_{ref,2}(N_1 + 1) \\ \vdots \\ -Q_{ll} [CA^2 A^{t_s - t_{akt}} x(t_{akt})]_l + Q_{ll} y_{ref,l}(N_1 + 1) \end{pmatrix} \in \mathbb{R}^l$$

$$d_{1N_y} = \begin{pmatrix} -Q_{11} [CA^{N_y} A^{t_s - t_{akt}} x(t_{akt})]_1 + Q_{11} y_{ref,1}(N_2) \\ -Q_{22} [CA^{N_y} A^{t_s - t_{akt}} x(t_{akt})]_2 + Q_{22} y_{ref,2}(N_2) \\ \vdots \\ -Q_{ll} [CA^{N_y} A^{t_s - t_{akt}} x(t_{akt})]_l + Q_{ll} y_{ref,l}(N_2) \end{pmatrix} \in \mathbb{R}^l,$$

$$d_2 = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{m \cdot N_u}. \quad (5.51)$$

Die in Abschnitt 5.3 vorgestellte Methode zur Durchführung des modellprädiktiven Regelungsansatzes kann nun unter Verwendung des gerade aufgeführten linearen Programms völlig analog angewendet werden. Der Unterschied besteht dabei darin, dass man hier in Schritt 3. durch die Anwendung der Kontrolle $u(t)$ nicht nur den Zustand $x(t+1)$, sondern daraus zusätzlich den Ausgang $y(t+1) = Cx(t+1)$ erhält. Dieser wird zwar in den weiteren Schritten nicht benötigt, kann aber bereits an dieser Stelle für die Ausgabe im Programm gespeichert werden. Sonst kann der optimale Ausgangsvektor auch am Ende mittels Anwendung des bereits berechneten Zustandsvektors berechnet werden.

5.4 Berücksichtigung von Beschränkungen

In der Praxis unterliegen sowohl die Eingänge als auch die Ausgänge oft Beschränkungen. Das System (5.28)

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned}$$

soll also zusätzlich den Bedingungen

$$y_{min} \leq y(t) \leq y_{max} \quad (5.52)$$

$$u_{min} \leq u(t) \leq u_{max} \quad (5.53)$$

genügen. Die Berücksichtigung solcher Beschränkungen verhindert, dass die Kontrolle Werte außerhalb von z.B. physikalischen Grenzen annimmt bzw. der Ausgang gewisse Sicherheitsbarrieren überschreitet. Folgende Abbildungen zeigen die Wirkung der Restriktionen:

Die Berechnung der optimale Kontrollfolge liefert ohne Beschränkungen zur Zeit $t = t_s + 1$ die optimale Kontrolle $u(t_s + 1)$, um den Referenzausgang $y_{ref}(t_s + 2)$ zu erreichen. Jedoch sind in diesem Fall sowohl die Beschränkung $u(t) \leq u_{max}$ für $t = t_s + 1$ als auch $y(t) \leq y_{max}$ für $t \in [t_s + 1, t_s + 2]$ verletzt. Wenn die Grenzen eingehalten werden sollen, kann also höchstens $u(t_s + 1) = u_{max}$ gewählt werden, wenn dadurch zusätzlich $y(t) \leq y_{max}$ für $t \in [t_s + 1, t_s + 2]$ erzielt wird. Falls dies nicht der Fall ist, ist sogar nur eine optimale Kontrolle der Form $u(t_s + 1) < u_{max}$ möglich.

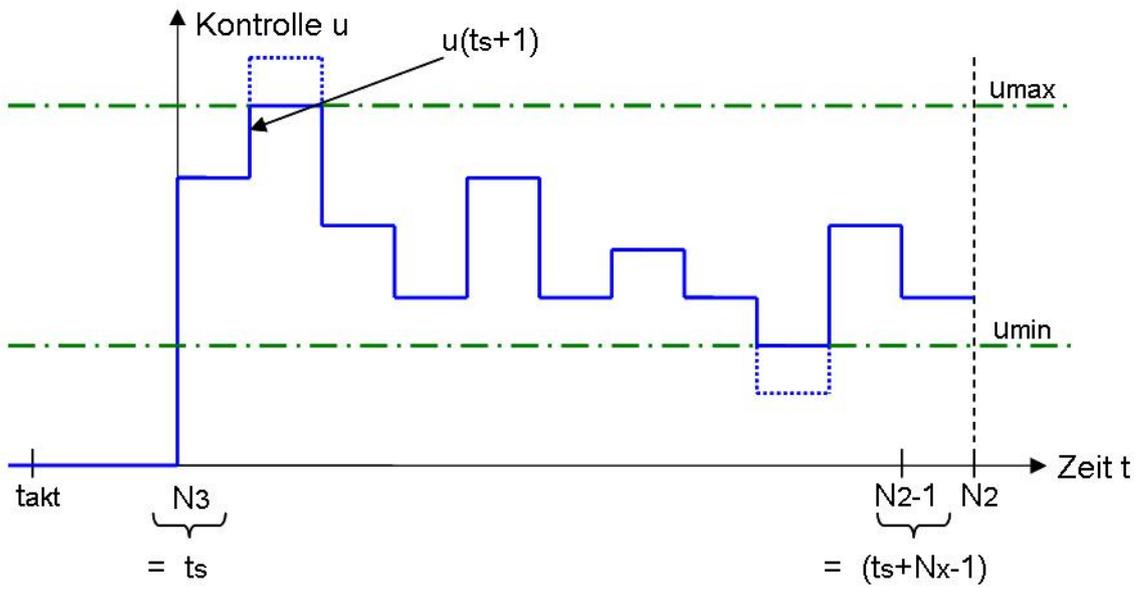


Abbildung 5.8: Optimale Kontrollfolge mit Beschränkungen

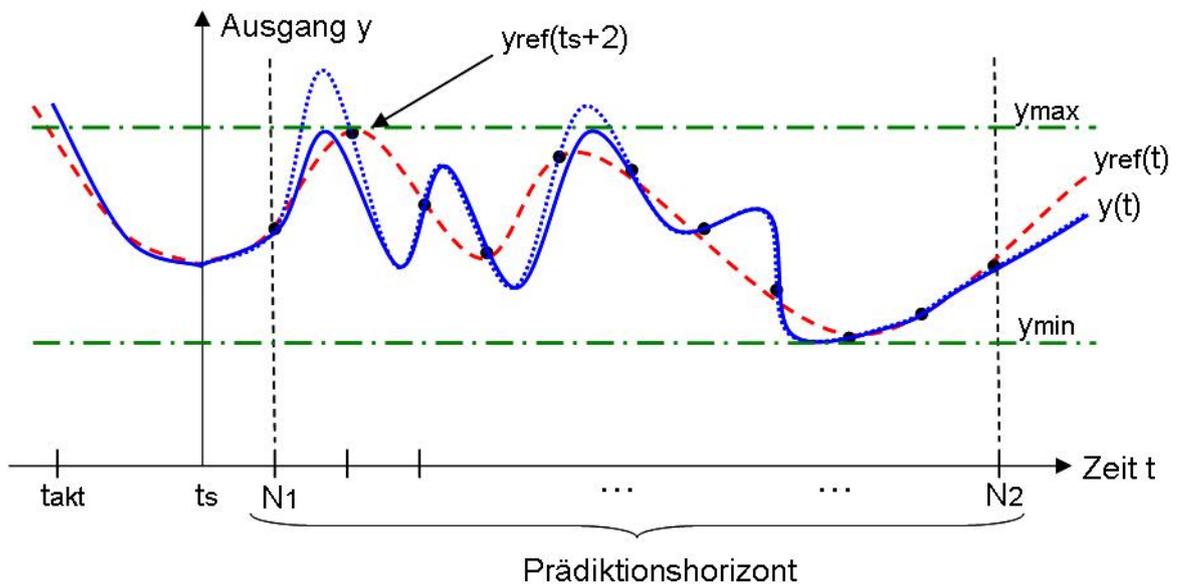


Abbildung 5.9: Optimales Verhalten des Ausgangs bei Beschränkungen

Das Einführen von Beschränkungen stellt jedoch für die vorgestellte Methode der modellprädiktiven Regelung mittels linearer Programmierung in der Theorie kein großes Problem dar. Es ist lediglich nötig, das Optimierungsproblem und das daraus resultierende lineare Programm um die Nebenbedingungen (5.52), (5.53) zu erweitern. Im nächsten Abschnitt werden die Hauptresultate dieser Erweiterung angegeben.

5.4.1 Erweiterung des linearen Programms

Unter Berücksichtigung der Beschränkungen (5.52), (5.53) lautet das Optimierungsproblem (5.29)

$$\min_{u:=[u^T(N_3) \dots u^T(N_4)]^T} \left\{ J(u, y(t)) = \sum_{t=N_1}^{N_2} Q \|y(t) - y_{ref}(t)\|_1 + \sum_{t=N_3}^{N_4} R \|u(t)\|_1 \right\} \quad (5.54)$$

s.t.

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), & t &= N_1 - 1, \dots, N_2 - 1 \\ y(t) &= Cx(t), & t &= N_1, \dots, N_2 \\ u(t) &= 0, & N_4 + 1 &\leq t \leq N_2 - 1 \\ y_{min} &\leq y(t) \leq y_{max}, & t &= N_1, \dots, N_2 \\ u_{min} &\leq u(t) \leq u_{max}, & t &= N_3, \dots, N_4 \end{aligned}$$

wobei u_{min} und u_{max} die Beschränkungen der Kontrolle $u(t)$ bezeichnen, welche über dem Kontrollhorizont eingehalten werden müssen. Weiterhin sind y_{min} und y_{max} die Grenzen des Ausgangs $y(t)$, welcher auf dem Prädiktionshorizont keine Werte außerhalb dieser Schranken annehmen darf. Mittels der in Abschnitt 5.3 angegebenen Gleichungen (5.35), (5.36) sowie (5.38), (5.39) ergibt sich unter Berücksichtigung der zusätzlichen Nebenbedingungen (5.52), (5.53) folgendes lineare Programm:

$$\min_z f \cdot z$$

$$\text{s.t.} \quad E \cdot z - d \leq 0$$

mit (5.40), (5.41),

$$\mathbf{E} = \begin{pmatrix} E_1 & E_3 & Z_2 \\ -E_1 & E_3 & Z_2 \\ E_2 & Z_1 & E_4 \\ -E_2 & Z_1 & E_4 \\ -E_5 & Z_3 & Z_2 \\ E_5 & Z_3 & Z_2 \\ E_6 & Z_1 & Z_4 \\ -E_6 & Z_1 & Z_4 \end{pmatrix} \in \mathbb{R}^{4(l \cdot N_y + m \cdot N_u) \times 2 \cdot m \cdot N_u + l \cdot N_y}, \quad (5.55)$$

(5.43) und (5.44) mit den dazugehörigen Teilmatrizen, (5.45), (5.46), (5.47) sowie (5.48).
Des Weiteren ist

$$E_5 = \begin{pmatrix} E_{51} & Z_{11} & \cdots & Z_{11} \\ E_{52} & E_{51} & & \vdots \\ \vdots & \ddots & \ddots & Z_{11} \\ E_{5N_u} & \cdots & E_{52} & E_{51} \\ \hline E_{5(N_u+1)} & E_{5N_u} & \cdots & E_{52} \\ \vdots & \ddots & \ddots & \vdots \\ E_{5(N_y)} & \cdots & E_{5(N_u+1)} & E_{5N_u} \end{pmatrix} \in \mathbb{R}^{(l \cdot N_y) \times (m \cdot N_u)}, \quad (5.56)$$

wobei

$$E_{51} = \begin{pmatrix} C_{11}B_{11} + C_{12}B_{21} + \cdots + C_{1n}B_{n1} & \cdots & C_{11}B_{1m} + C_{12}B_{2m} + \cdots + C_{1n}B_{nm} \\ C_{21}B_{11} + C_{22}B_{21} + \cdots + C_{2n}B_{n1} & \cdots & C_{21}B_{1m} + C_{22}B_{2m} + \cdots + C_{2n}B_{nm} \\ \vdots & \cdots & \vdots \\ C_{l1}B_{11} + C_{l2}B_{21} + \cdots + C_{ln}B_{n1} & \cdots & C_{l1}B_{1m} + C_{l2}B_{2m} + \cdots + C_{ln}B_{nm} \end{pmatrix} \in \mathbb{R}^{l \times m}$$

$$E_{52} = \begin{pmatrix} C_{11}A_{11}B_{11} + \cdots + C_{11}A_{1n}B_{n1} + & \cdots & C_{11}A_{11}B_{1m} + \cdots + C_{11}A_{1n}B_{nm} + \\ +C_{12}A_{21}B_{11} + \cdots + C_{12}A_{2n}B_{n1} + & \cdots & +C_{12}A_{21}B_{1m} + \cdots + C_{12}A_{2n}B_{nm} + \\ + \cdots \cdots + & \cdots & + \cdots \cdots + \\ +C_{1n}A_{n1}B_{11} + \cdots + C_{1n}A_{nn}B_{n1} & \cdots & +C_{1n}A_{n1}B_{1m} + \cdots + C_{1n}A_{nn}B_{nm} \\ \vdots & \cdots & \vdots \\ C_{l1}A_{11}B_{11} + \cdots + C_{l1}A_{1n}B_{n1} + & \cdots & C_{l1}A_{11}B_{1m} + \cdots + C_{l1}A_{1n}B_{nm} + \\ +C_{l2}A_{21}B_{11} + \cdots + C_{l2}A_{2n}B_{n1} + & \cdots & +C_{l2}A_{21}B_{1m} + \cdots + C_{l2}A_{2n}B_{nm} + \\ + \cdots \cdots + & \cdots & + \cdots \cdots + \\ +C_{ln}A_{n1}B_{11} + \cdots + C_{ln}A_{nn}B_{n1} & \cdots & +C_{ln}A_{n1}B_{1m} + \cdots + C_{ln}A_{nn}B_{nm} \end{pmatrix} \in \mathbb{R}^{l \times m}$$

$$E_{5N_u} = \begin{pmatrix} C_{11}A_{11}^{N_u-1}B_{11} + \cdots + C_{11}A_{1n}^{N_u-1}B_{n1} + & \cdots & C_{11}A_{11}^{N_u-1}B_{1m} + \cdots + C_{11}A_{1n}^{N_u-1}B_{nm} + \\ +C_{12}A_{21}^{N_u-1}B_{11} + \cdots + C_{12}A_{2n}^{N_u-1}B_{n1} + & \cdots & +C_{12}A_{21}^{N_u-1}B_{1m} + \cdots + C_{12}A_{2n}^{N_u-1}B_{nm} + \\ + \cdots \cdots + & \cdots & + \cdots \cdots + \\ +C_{1n}A_{n1}^{N_u-1}B_{11} + \cdots + C_{1n}A_{nn}^{N_u-1}B_{n1} & \cdots & +C_{1n}A_{n1}^{N_u-1}B_{1m} + \cdots + C_{1n}A_{nn}^{N_u-1}B_{nm} \\ \vdots & \cdots & \vdots \\ C_{l1}A_{11}^{N_u-1}B_{11} + \cdots + C_{l1}A_{1n}^{N_u-1}B_{n1} + & \cdots & C_{l1}A_{11}^{N_u-1}B_{1m} + \cdots + C_{l1}A_{1n}^{N_u-1}B_{nm} + \\ +C_{l2}A_{21}^{N_u-1}B_{11} + \cdots + C_{l2}A_{2n}^{N_u-1}B_{n1} + & \cdots & +C_{l2}A_{21}^{N_u-1}B_{1m} + \cdots + C_{l2}A_{2n}^{N_u-1}B_{nm} + \\ + \cdots \cdots + & \cdots & + \cdots \cdots + \\ +C_{ln}A_{n1}^{N_u-1}B_{11} + \cdots + C_{ln}A_{nn}^{N_u-1}B_{n1} & \cdots & +C_{ln}A_{n1}^{N_u-1}B_{1m} + \cdots + C_{ln}A_{nn}^{N_u-1}B_{nm} \end{pmatrix} \in \mathbb{R}^{l \times m}$$

$$E_{5(N_u+1)} = \begin{pmatrix} C_{11}A_{11}^{N_u}B_{11} + \dots + C_{11}A_{1n}^{N_u}B_{n1} + & \dots & C_{11}A_{11}^{N_u}B_{1m} + \dots + C_{11}A_{1n}^{N_u}B_{nm} + \\ +C_{12}A_{21}^{N_u}B_{11} + \dots + C_{12}A_{2n}^{N_u}B_{n1} + & \dots & +C_{12}A_{21}^{N_u}B_{1m} + \dots + C_{12}A_{2n}^{N_u}B_{nm} + \\ & + \dots \dots + & + \dots \dots + \\ +C_{1n}A_{n1}^{N_u}B_{11} + \dots + C_{1n}A_{nn}^{N_u}B_{n1} & \dots & +C_{1n}A_{n1}^{N_u}B_{1m} + \dots + C_{1n}A_{nn}^{N_u}B_{nm} \\ & \vdots & \vdots \\ C_{l1}A_{11}^{N_u}B_{11} + \dots + C_{l1}A_{1n}^{N_u}B_{n1} + & \dots & C_{l1}A_{11}^{N_u}B_{1m} + \dots + C_{l1}A_{1n}^{N_u}B_{nm} + \\ +C_{l2}A_{21}^{N_u}B_{11} + \dots + C_{l2}A_{2n}^{N_u}B_{n1} + & \dots & +C_{l2}A_{21}^{N_u}B_{1m} + \dots + C_{l2}A_{2n}^{N_u}B_{nm} + \\ & + \dots \dots + & + \dots \dots + \\ +C_{ln}A_{n1}^{N_u}B_{11} + \dots + C_{ln}A_{nn}^{N_u}B_{n1} & \dots & +C_{ln}A_{n1}^{N_u}B_{1m} + \dots + C_{ln}A_{nn}^{N_u}B_{nm} \end{pmatrix} \in \mathbb{R}^{l \times m}$$

$$E_{5N_y} = \begin{pmatrix} C_{11}A_{11}^{N_y-1}B_{11} + \dots + C_{11}A_{1n}^{N_y-1}B_{n1} + & \dots & C_{11}A_{11}^{N_y-1}B_{1m} + \dots + C_{11}A_{1n}^{N_y-1}B_{nm} + \\ +C_{12}A_{21}^{N_y-1}B_{11} + \dots + C_{12}A_{2n}^{N_y-1}B_{n1} + & \dots & +C_{12}A_{21}^{N_y-1}B_{1m} + \dots + C_{12}A_{2n}^{N_y-1}B_{nm} + \\ & + \dots \dots + & + \dots \dots + \\ +C_{1n}A_{n1}^{N_y-1}B_{11} + \dots + C_{1n}A_{nn}^{N_y-1}B_{n1} & \dots & +C_{1n}A_{n1}^{N_y-1}B_{1m} + \dots + C_{1n}A_{nn}^{N_y-1}B_{nm} \\ & \vdots & \vdots \\ C_{l1}A_{11}^{N_y-1}B_{11} + \dots + C_{l1}A_{1n}^{N_y-1}B_{n1} + & \dots & C_{l1}A_{11}^{N_y-1}B_{1m} + \dots + C_{l1}A_{1n}^{N_y-1}B_{nm} + \\ +C_{l2}A_{21}^{N_y-1}B_{11} + \dots + C_{l2}A_{2n}^{N_y-1}B_{n1} + & \dots & +C_{l2}A_{21}^{N_y-1}B_{1m} + \dots + C_{l2}A_{2n}^{N_y-1}B_{nm} + \\ & + \dots \dots + & + \dots \dots + \\ +C_{ln}A_{n1}^{N_y-1}B_{11} + \dots + C_{ln}A_{nn}^{N_y-1}B_{n1} & \dots & +C_{ln}A_{n1}^{N_y-1}B_{1m} + \dots + C_{ln}A_{nn}^{N_y-1}B_{nm} \end{pmatrix} \in \mathbb{R}^{l \times m}.$$

Außerdem gilt

$$E_6 = \begin{pmatrix} E_{61} & Z_{21} & \dots & Z_{21} \\ Z_{21} & E_{61} & & \vdots \\ \vdots & & \ddots & Z_{21} \\ Z_{21} & \dots & Z_{21} & E_{61} \end{pmatrix} \in \mathbb{R}^{m \cdot N_u \times m \cdot N_u} \quad (5.57)$$

mit

$$E_{61} = \begin{pmatrix} -1 & 0 & \dots & 0 \\ 0 & -1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & -1 \end{pmatrix} \in \mathbb{R}^{m \times m}$$

sowie

$$Z_3 = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix} \in \mathbb{R}^{l \cdot N_y \times l \cdot N_y} \quad (5.58)$$

$$Z_4 = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix} \in \mathbb{R}^{m \cdot N_u \times m \cdot N_u}. \quad (5.59)$$

Der Vektor d wird unter Berücksichtigung von Beschränkungen wie folgt definiert

$$\mathbf{d} = \begin{pmatrix} d_1 \\ -d_1 \\ d_2 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{pmatrix} \in \mathbb{R}^{2(2 \cdot l \cdot N_y + 2 \cdot m \cdot N_u)} \quad (5.60)$$

mit (5.50) und (5.51) sowie den zugehörigen Teilvektoren.
Des Weiteren gilt

$$d_3 = \begin{pmatrix} d_{31} \\ d_{32} \\ \vdots \\ \vdots \\ d_{3N_y} \end{pmatrix} \in \mathbb{R}^{l \cdot N_y} \quad (5.61)$$

mit

$$d_{31} = \begin{pmatrix} -y_{min,1} + [CAA^2 x(0)]_1 \\ -y_{min,2} + [CAA^2 x(0)]_2 \\ \vdots \\ -y_{min,l} + [CAA^2 x(0)]_l \end{pmatrix} \in \mathbb{R}^l$$

$$d_{32} = \begin{pmatrix} -y_{min,1} + [CA^2A^2x(0)]_1 \\ -y_{min,2} + [CA^2A^2x(0)]_2 \\ \vdots \\ -y_{min,l} + [CA^2A^2x(0)]_l \end{pmatrix} \in \mathbb{R}^l$$

$$d_{3N_y} = \begin{pmatrix} -y_{min,1} + [CA^{N_y}A^2x(0)]_1 \\ -y_{min,2} + [CA^{N_y}A^2x(0)]_2 \\ \vdots \\ -y_{min,l} + [CA^{N_y}A^2x(0)]_l \end{pmatrix} \in \mathbb{R}^l$$

$$\text{und } d_4 = \begin{pmatrix} d_{41} \\ d_{42} \\ \vdots \\ \vdots \\ d_{4N_y} \end{pmatrix} \in \mathbb{R}^{l \cdot N_y} \quad (5.62)$$

mit

$$d_{41} = \begin{pmatrix} y_{max,1} - [CAA^2x(0)]_1 \\ y_{max,2} - [CAA^2x(0)]_2 \\ \vdots \\ y_{max,l} - [CAA^2x(0)]_l \end{pmatrix} \in \mathbb{R}^l$$

$$d_{42} = \begin{pmatrix} y_{max,1} - [CA^2A^2x(0)]_1 \\ y_{max,2} - [CA^2A^2x(0)]_2 \\ \vdots \\ y_{max,l} - [CA^2A^2x(0)]_l \end{pmatrix} \in \mathbb{R}^l$$

$$d_{4N_y} = \begin{pmatrix} y_{max,1} - [CA^{N_y}A^2x(0)]_1 \\ y_{max,2} - [CA^{N_y}A^2x(0)]_2 \\ \vdots \\ y_{max,l} - [CA^{N_y}A^2x(0)]_l \end{pmatrix} \in \mathbb{R}^l.$$

Außerdem sind d_5 und d_6 definiert als

$$d_5 = \begin{pmatrix} d_{51} \\ \vdots \\ d_{51} \end{pmatrix} \in \mathbb{R}^{m \cdot N_u} \quad (5.63)$$

und

$$d_6 = \begin{pmatrix} d_{61} \\ \vdots \\ d_{61} \end{pmatrix} \in \mathbb{R}^{m \cdot N_u}, \quad (5.64)$$

wobei

$$d_{51} = \begin{pmatrix} -u_{min,1} \\ -u_{min,2} \\ \vdots \\ -u_{min,m} \end{pmatrix} \in \mathbb{R}^m$$

$$d_{61} = \begin{pmatrix} u_{max,1} \\ u_{max,2} \\ \vdots \\ u_{max,m} \end{pmatrix} \in \mathbb{R}^m.$$

Lineare Kontrollsysteme ohne Ausgang

Es ist ebenfalls möglich bei dem in Abschnitt 5.1 angegebenen Linearen Programm Beschränkungen zu berücksichtigen. Dabei werden die Nebenbedingungen um die Gleichungen

$$x_{min} \leq x(t) \leq x_{max} \quad (5.65)$$

$$u_{min} \leq u(t) \leq u_{max} \quad (5.66)$$

erweitert. Das dazugehörige lineare Programm kann dann für Kontrollsysteme (5.1)

$$x(t+1) = Ax(t) + Bu(t)$$

wie folgt formuliert werden.

$$\begin{aligned} & \min_z f \cdot z \\ \text{s.t.} \quad & E \cdot z - d \leq 0, \end{aligned}$$

wobei gilt:

- (5.16) und (5.17)
- (5.55) mit $l = n$ und $N_y = N_x$
- (5.19) und (5.20) mit zugehörigen Teilmatrizen
- (5.21), (5.22), (5.23) und (5.24)
- (5.56) mit $l = n$, $N_y = N_x$, wobei

$$E_{51} = \begin{pmatrix} B_{11} & B_{12} & \dots & B_{1m} \\ B_{21} & B_{22} & \dots & B_{2m} \\ \vdots & \vdots & \dots & \vdots \\ B_{n1} & B_{n2} & \dots & B_{nm} \end{pmatrix} \in \mathbb{R}^{n \times m}$$

$$E_{52} = \begin{pmatrix} A_{11}B_{11} + A_{12}B_{21} + \dots + A_{1n}B_{n1} & \dots & A_{11}B_{1m} + A_{12}B_{2m} + \dots + A_{1n}B_{nm} \\ A_{21}B_{11} + A_{22}B_{21} + \dots + A_{2n}B_{n1} & \dots & A_{21}B_{1m} + A_{22}B_{2m} + \dots + A_{2n}B_{nm} \\ \vdots & \dots & \vdots \\ A_{n1}B_{11} + A_{n2}B_{21} + \dots + A_{nn}B_{n1} & \dots & A_{n1}B_{1m} + A_{n2}B_{2m} + \dots + A_{nn}B_{nm} \end{pmatrix} \in \mathbb{R}^{n \times m}$$

$$E_{5N_u} = \begin{pmatrix} A_{11}^{N_u-1}B_{11} + \dots + A_{1n}^{N_u-1}B_{n1} & \dots & A_{11}^{N_u-1}B_{1m} + \dots + A_{1n}^{N_u-1}B_{nm} \\ A_{21}^{N_u-1}B_{11} + \dots + A_{2n}^{N_u-1}B_{n1} & \dots & A_{21}^{N_u-1}B_{1m} + \dots + A_{2n}^{N_u-1}B_{nm} \\ \vdots & \dots & \vdots \\ A_{n1}^{N_u-1}B_{11} + \dots + A_{nn}^{N_u-1}B_{n1} & \dots & A_{n1}^{N_u-1}B_{1m} + \dots + A_{nn}^{N_u-1}B_{nm} \end{pmatrix} \in \mathbb{R}^{n \times m}$$

$$E_{5(N_u+1)} = \begin{pmatrix} A_{11}^{N_u}B_{11} + \dots + A_{1n}^{N_u}B_{n1} & \dots & A_{11}^{N_u}B_{1m} + \dots + A_{1n}^{N_u}B_{nm} \\ A_{21}^{N_u}B_{11} + \dots + A_{2n}^{N_u}B_{n1} & \dots & A_{21}^{N_u}B_{1m} + \dots + A_{2n}^{N_u}B_{nm} \\ \vdots & \dots & \vdots \\ A_{n1}^{N_u}B_{11} + \dots + A_{nn}^{N_u}B_{n1} & \dots & A_{n1}^{N_u}B_{1m} + \dots + A_{nn}^{N_u}B_{nm} \end{pmatrix} \in \mathbb{R}^{n \times m}$$

$$E_{5N_x} = \begin{pmatrix} A_{11}^{N_x-1}B_{11} + \dots + A_{1n}^{N_x-1}B_{n1} & \dots & A_{11}^{N_x-1}B_{1m} + \dots + A_{1n}^{N_x-1}B_{nm} \\ A_{21}^{N_x-1}B_{11} + \dots + A_{2n}^{N_x-1}B_{n1} & \dots & A_{21}^{N_x-1}B_{1m} + \dots + A_{2n}^{N_x-1}B_{nm} \\ \vdots & \dots & \vdots \\ A_{n1}^{N_x-1}B_{11} + \dots + A_{nn}^{N_x-1}B_{n1} & \dots & A_{n1}^{N_x-1}B_{1m} + \dots + A_{nn}^{N_x-1}B_{nm} \end{pmatrix} \in \mathbb{R}^{n \times m}$$

- (5.57) mit zugehöriger Teilmatrix
- (5.58) mit $l = n$
- (5.59)
- (5.60) mit $l = n, N_y = N_x$
- (5.26) mit zugehörigen Teilvektoren
- (5.27)
- (5.61) mit $l = n, N_y = N_x$ und zugehörigen Teilvektoren mit $y_{min} = x_{min}$ und ohne Matrix C
- (5.62) mit $l = n, N_y = N_x$ und zugehörigen Teilvektoren mit $y_{max} = x_{max}$ und ohne Matrix C
- (5.63), (5.64) mit zugehörigen Teilvektoren.

Kapitel 6

Beispiele und numerische Ergebnisse

Das entwickelte Programm zur modellprädiktiven Regelung mittels linearer Programmierung soll in diesem Kapitel an bekannten Beispielen getestet werden. Dabei wurden die Beispiele wie folgt gewählt.

Bei Beispiel 1 (inverses Pendel) handelt es sich um ein lineares Kontrollsystem

- ohne Ausgang
- mit vierdimensionalem Zustandsvektor $x \in \mathbb{R}^4$, $n = 4$
- mit eindimensionaler Kontrolle $u \in \mathbb{R}$, $m = 1$.

Das Beispiel 2 (2-DOF Hubschrauber) liefert ein lineares Modell

- mit zweidimensionalem Ausgang $y \in \mathbb{R}^2$, $l = 2$
- mit vierdimensionalem Zustandsvektor $x \in \mathbb{R}^4$, $n = 4$
- mit zweidimensionaler Kontrolle $u \in \mathbb{R}^2$, $m = 2$.

Es findet also sowohl das in Abschnitt 5.1 hergeleitete lineare Programm für Kontrollsysteme ohne Ausgang als auch das in Punkt 5.3 angegebene LP für lineare Kontrollsystem mit Ausgang bei der Lösung des Stabilisierungsproblems mittels modellprädiktiver Regelung Anwendung.

Außerdem werden sowohl für die Regelung des inversen Pendels als auch für die Stabilisierung des Hubschraubers folgende Annahmen gemacht:

1. $t_s = t_{akt} = 0$
2. $x_{ref}(t) = 0$ bzw. $y_{ref}(t) = 0$ für $t > 0$
3. es gibt keine Beschränkungen

6.1 Beispiel 1: Das inverse Pendel

6.1.1 Systembeschreibung und Modellbildung

Das inverse Pendel dient zur Darstellung von „Balancierungsproblemen“. Ein bekanntes Beispiel eines solchen Problems stellt das Balancieren eines Stabes mit der Handfläche oder einem Finger dar. Um den Stab aufrecht zu halten, müssen dessen Bewegungen dabei ständig ausgeglichen werden. Während sich die Hand nach oben, unten und zur Seite bewegen kann, ist das inverse Pendel auf die Bewegung in einer Ebene beschränkt. Wie ein balancierter Stab ist auch das inverse Pendel ein instabiles System, welches nur durch gezieltes Anwenden von Kräften stabilisiert werden kann. Dieses Stabilisierungsproblem des inversen Pendels ist eines der anschaulichsten klassischen Probleme der Kontrolltheorie.

Der mechanische Aufbau besteht dabei aus einem horizontal frei beweglichen Wagen, an dem drehbar ein Stab angebracht ist. Das Pendel der Länge l mit der Pendelmasse m (kugelförmig, Radius r) ist an seinem unteren Ende auf dem Wagen der Masse M gelagert, so dass es in einer Ebene um den Winkel θ kippen kann. Um die Kippbewegung aufzufangen kann der Wagen durch die Regelkraft u in x -Richtung nach links oder rechts bewegt werden (siehe Abbildung 6.1). Die Kippbewegung wird dabei durch die Schwerkraft mit Gravitationskonstante g verursacht. Aufgabe der Regelung ist es, das Pendel durch eine geeignete Steuerung des Wagens in der aufrechten Position zu halten.

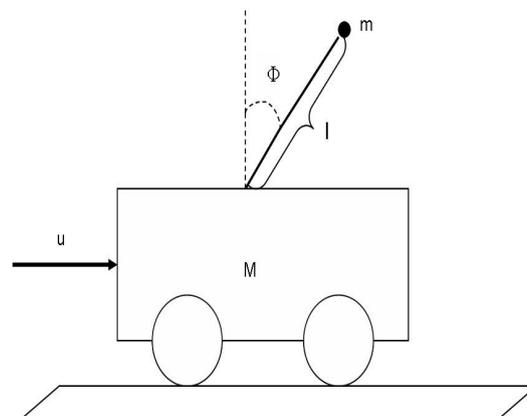


Abbildung 6.1: Das inverse Pendel

Das Problem des inversen Pendels ist allerdings nicht nur von rein akademischem Interesse. In der Praxis existieren unter anderem folgende Anwendungen.

- Das Balancieren einer Rakete bei der Fahrt von der Montagehalle zur Startrampe

- Die Stabilisierung der vertikalen Position eines Space Shuttles in den ersten Flugabschnitten
- Das Halten eines zweibeinigen Roboters in einer aufrechten Position - auch ein stillstehender Mensch kann so als ein inverses Pendel betrachtet werden
- Einachsige, selbststabilisierende Roller, z.B. der Segway Personal Transporter



Abbildung 6.2: Der Segway Personal Transporter

Die beiden erstgenannten Punkte, zusammengefasst als die Bewältigung komplexer Steuerungs- und Regelungsaufgaben in der Luft- und Raumfahrt, sind zudem der Motor für die historische Entwicklung der Kontrolltheorie gewesen.

Mittels physikalischer Gesetze kann für den in Abbildung 6.1 dargestellten mechanischen Aufbau des klassischen inversen Pendels ein Differentialgleichungsmodell der Form $\dot{x}(t) = f(x(t), u(t))$ hergeleitet werden.

$$\begin{aligned}
 \dot{x}_1(t) &= x_2(t) \\
 \dot{x}_2(t) &= -kx_2(t) + g \sin(x_1(t)) + u(t) \cos(x_1(t)) \\
 \dot{x}_3(t) &= x_4(t) \\
 \dot{x}_4(t) &= u(t)
 \end{aligned} \tag{6.1}$$

Hierbei beschreibt x_1 dem Winkel θ des Pendels, der entgegen dem Uhrzeigersinn zunimmt. $x_1 = 0$ entspricht dabei gerade dem aufgerichteten Pendel. x_2 ist die Winkelgeschwindigkeit, x_3 die Position des Wagens und x_4 dessen Geschwindigkeit. Die Kontrolle u ist hier

die Beschleunigung des Wagens, die zur Balancierung des Pendels notwendig ist. Außerdem beschreibt die Konstante k die Reibung des Pendels (je größer k , desto mehr Reibung) und $g \approx 9.81m/s^2$ die für das Kippen verantwortliche Erdbeschleunigung.

Das Differenzialgleichungsmodell (6.1) ist bereits eine Vereinfachung: es wurde angenommen, dass das Pendel so leicht ist, dass es keinen Einfluss auf die Bewegung des Wagens hat. Zudem wurde eine Reihe von Konstanten so gewählt, dass sie sich gegeneinander aufheben (z.B. l , m und M). Außerdem handelt es sich nicht um ein lineares Kontrollsystem, da sich die nichtlinearen Funktionen \sin und \cos nicht mittels der Matrizen A und B darstellen lassen. Trotzdem kann ein lineares Modell der Form $\dot{x}(t) = Ax(t) + Bu(t)$ verwendet werden, um (6.1) in der Nähe gewisser Punkte zu approximieren. Diese Methode der Linearisierung ist möglich in der Nähe von Punkten (x^*, u^*) , in denen $f(x^*, u^*) = 0$ gilt.

Für das inverse Pendel gilt $f(0,0) = 0$, dieser Punkt entspricht im Modell gerade dem aufrecht stehenden Pendel mit still stehendem und unbeschleunigtem Wagen. Man erhält für x und u nahe Null ($x \approx 0$, $u \approx 0$) folgende Näherungen

$$\begin{aligned}\sin(x_1) &\approx x_1 \\ \cos(x_1) &\approx 1.\end{aligned}$$

Daraus folgt für das Differenzialgleichungsmodell (6.1) das lineare Kontrollsystem

$$\dot{x}(t) = \underbrace{\begin{pmatrix} 0 & 1 & 0 & 0 \\ g & -k & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}}_{=A} x(t) + \underbrace{\begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}}_{=B} u(t) \quad (6.2)$$

Bevor nun die in Kapitel 5 vorgestellte Methode der modellprädiktiven Regelung mittels linearer Programmierung zur Stabilisierung des Pendels angewendet werden kann, muss das in kontinuierlicher Zeit $t \in \mathbb{R}$ definierte lineare Modell (6.2) zunächst in ein entsprechendes zeitdiskretes Kontrollsystem der Form

$$x(t+1) = Ax(t) + Bu(t) \quad (6.3)$$

umgewandelt werden. Hier ist sowohl die Kontrollfunktion $u(t)$ als auch die Lösung $x(t)$ nur für $t \in \mathbb{Z}$ definiert. Setzt man in $\dot{x}(t) = Ax(t) + Bu(t)$ eine Kontrollfunktion ein, die konstant auf Intervallen der Form $[k, k + t_{abt})$, $k \in \mathbb{Z}$, ist, liefern beide Modelle für $t \in \mathbb{Z}$ die gleichen Lösungen, wenn man A und B in (6.3) als

$$A = e^{At_{abt}} \quad \text{und} \quad (6.4)$$

$$B = \int_0^{t_{abt}} e^{A(t_{abt}-\tau)} B d\tau \quad (6.5)$$

mit Abtastzeit t_{abt} wählt.

In Matlab steht zur Umwandlung kontinuierlicher Systeme in analoge zeitdiskrete Systeme die Funktion `c2d.m` zur Verfügung. Mit $k = 0.1$, $g = 9.81$ und $t_{abt} = 0.5$ erhält man mittels

```
% -----
%      Bsp.: inverses Pendel
% -----

k = 0.1; % Reibung
g = 9.81; % Erdbeschleunigung

A_kontinuierlich = [0 1 0 0; g -k 0 0; 0 0 0 1; 0 0 0 0];
B_kontinuierlich = [0; 1; 0; 1];

% -----
% Umwandlung in ein zeitdiskretes Modell
% -----

t_abt = 0.5; % Abtastzeit

A = c2d_A(A_kontinuierlich,B_kontinuierlich,t_abt)
B = c2d_B(A_kontinuierlich,B_kontinuierlich,t_abt)
```

die folgenden Matrizen A und B des linearen zeitdiskreten Kontrollsystems (6.3) für das inverse Pendel.

$$A = \begin{pmatrix} 2.4727 & 0.7130 & 0 & 0 \\ 6.9942 & 2.4014 & 0 & 0 \\ 0 & 0 & 1.0000 & 0.5000 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix} \quad \text{und} \quad B = \begin{pmatrix} 0.1501 \\ 0.7130 \\ 0.1250 \\ 0.5000 \end{pmatrix} \quad (6.6)$$

Die Matlab-eigene Funktion `c2d.m` wurde dabei durch eine leichte Modifikation (\implies selbstgeschriebene Routinen: `c2d_A`, `c2d_B`) den verwendeten Bezeichnungen angepasst.

6.1.2 Ergebnisse der modellprädiktiven Regelung mittels linearer Programmierung

Im Folgenden werden die numerischen Ergebnisse der Anwendung modellprädiktiver Regelung mittels linearer Programmierung auf das inverse Pendel genauer betrachtet. Grundlage dafür ist das lineare Kontrollsystem

$$x(t+1) = Ax(t) + Bu(t)$$

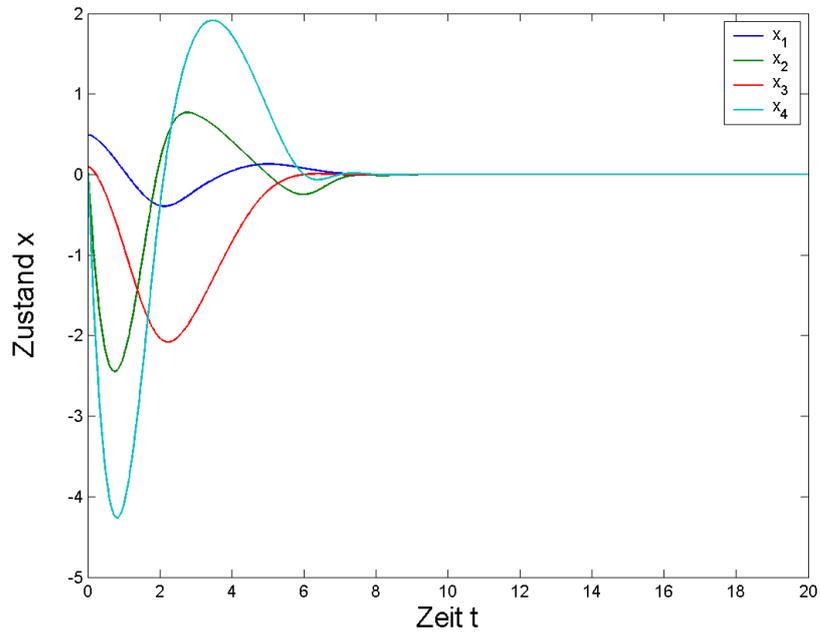
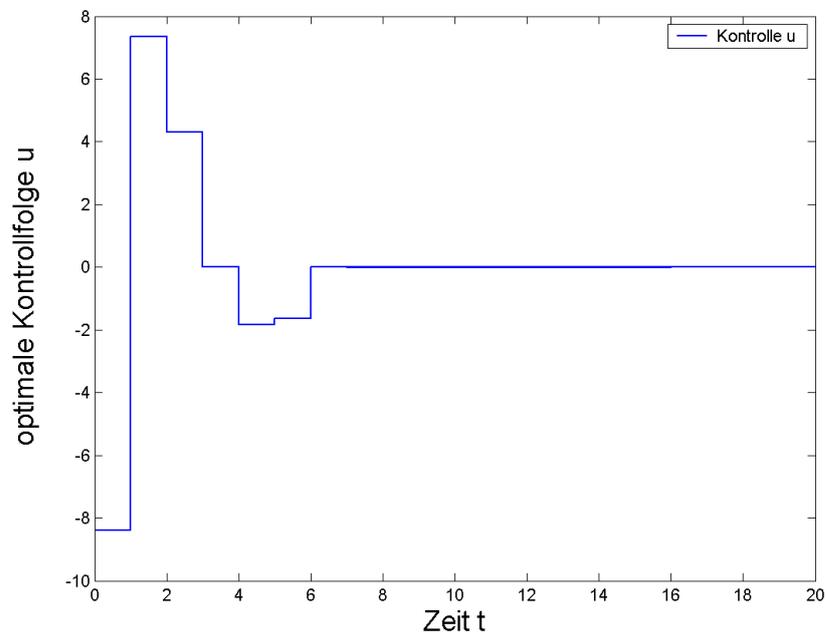
mit (6.6). Im ersten Teil soll zunächst veranschaulicht werden, wie sich die Wahl der Länge des Kontrollhorizontes sowie des Prädiktionshorizontes auf das Verhalten der Lösungen auswirkt. Wie bereits in der Einleitung erwähnt, gilt im Folgenden stets $x_{ref} = 0$. Außerdem werden die weiteren Parameter wie folgt gewählt.

```
m = 1; n = 4;
Q = 2.0*eye(n,n); R = 0.5*eye(m,m);
t_akt = 0; t_s = 0;
x1_t_akt = 0.5; x2_t_akt = 0.1; x3_t_akt = 0.1; x4_t_akt = 0.1;
x_anfang = [x1_t_akt; x2_t_akt; x3_t_akt; x4_t_akt];
```

Die folgenden Abbildungen zeigen das Verhalten des Zustandes $x(t)$ bei modellprädiktiver Regelung mittels linearer Programmierung. Man sieht, dass die Wahl der Horizontlängen dabei tatsächlich eine Rolle spielt. Die Abbildungen 6.3 und 6.4 veranschaulichen die Ergebnisse für $N_u = 5$ und $N_x = 15$. In diesem Fall gilt $x(t) \approx 0$ ab $t = 12$. Die in den Abbildungen dargestellten kontinuierlichen Lösungen wurden mit dem Befehl `spline` in Matlab mittels einer Splineinterpolation berechnet.

```
Zustand_x =
  Columns 1 through 8
    0.5000    0.0464   -0.3870   -0.1968    0.0447    0.1304    0.0781    0.0159
    0.1000   -2.2529    0.1603    0.7454    0.4133   -0.0000   -0.2485   -0.0506
    0.1000   -0.9002   -2.0309   -1.7040   -0.8393   -0.2035    0.0000    0.0000
    0.1000   -4.1009   -0.4217    1.7293    1.7293    0.8140    0.0000   -0.0000
  Columns 9 through 16
    0.0032    0.0007    0.0001    0.0000    0.0000    0.0000    0.0000    0.0000 ...
   -0.0103   -0.0021   -0.0004   -0.0001   -0.0000   -0.0000   -0.0000   -0.0000 ...
   -0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000 ...
   -0.0000    0.0000   -0.0000   -0.0000    0.0000    0.0000   -0.0000   -0.0000 ...

Optimale_Kontrollfolge =
  Columns 1 through 8
   -8.4018    7.3583    4.3021   -0.0000   -1.8307   -1.6279   -0.0000   -0.0000
  Columns 9 through 15
    0.0000   -0.0000    0.0000    0.0000    0.0000   -0.0000    0.0000 ...
```

Abbildung 6.3: Verhalten des Zustandes für $N_u = 5$ und $N_x = 15$ Abbildung 6.4: Optimale Kontrollfolge für $N_u = 5$ und $N_x = 15$

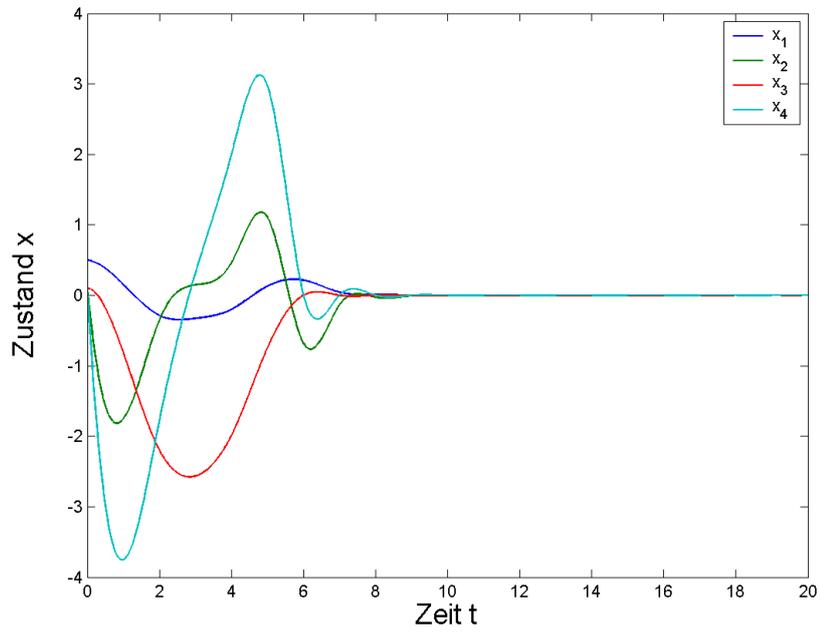


Abbildung 6.5: Verhalten des Zustandes für $N_u = 3$ und $N_x = 15$

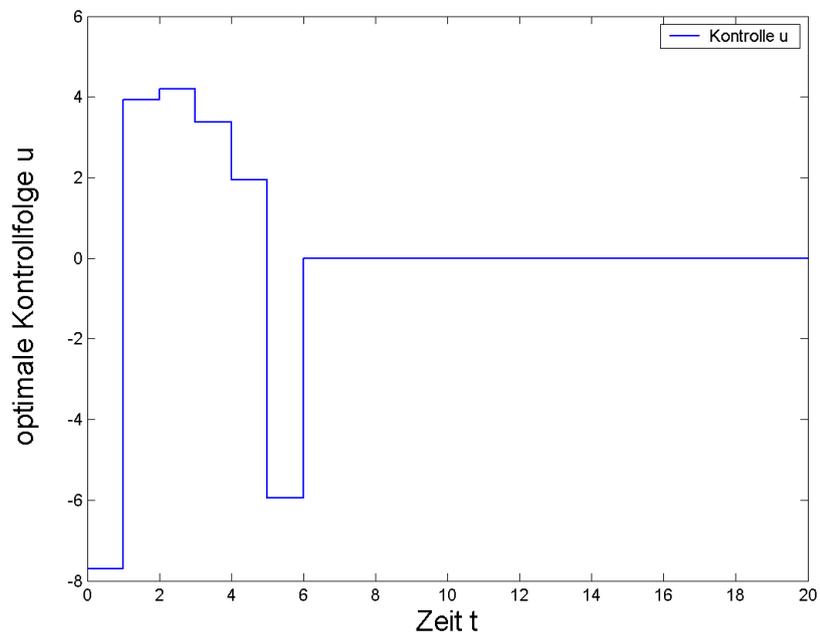
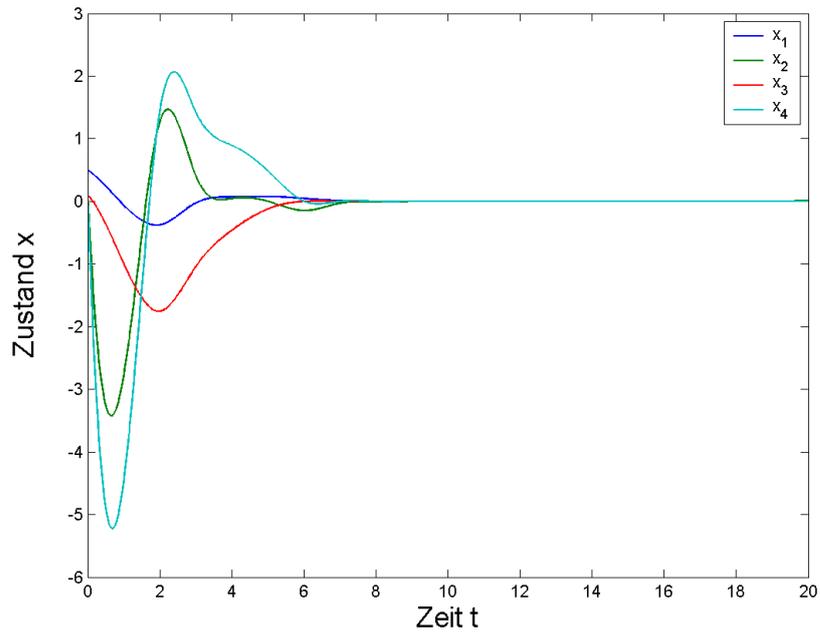
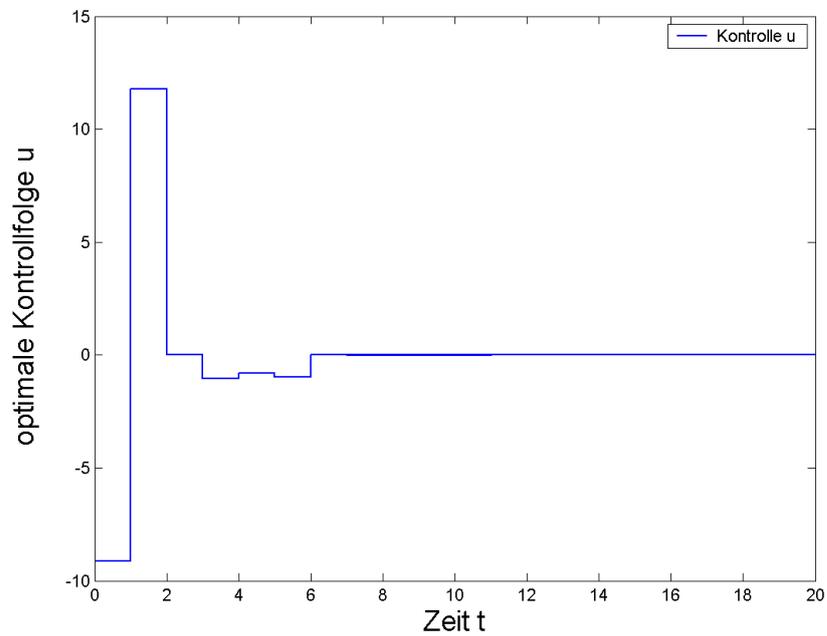
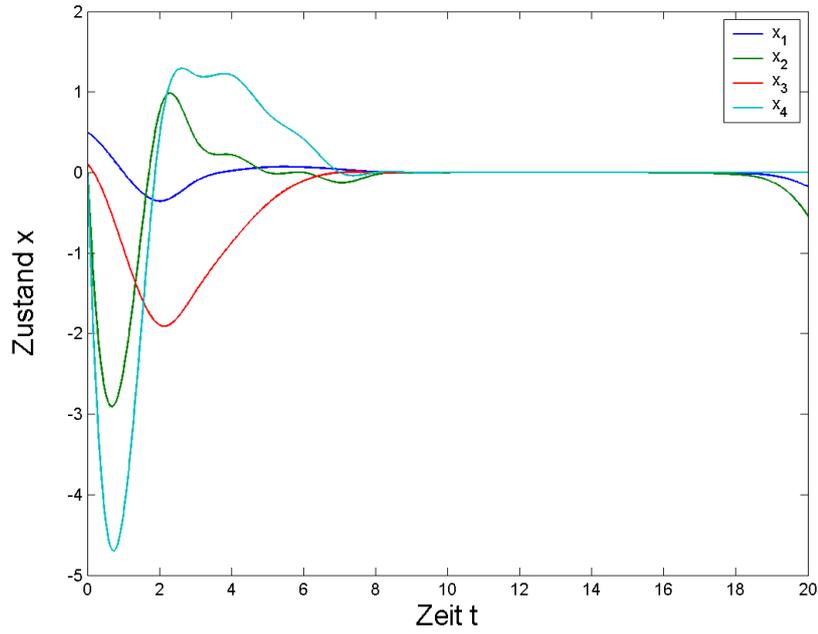
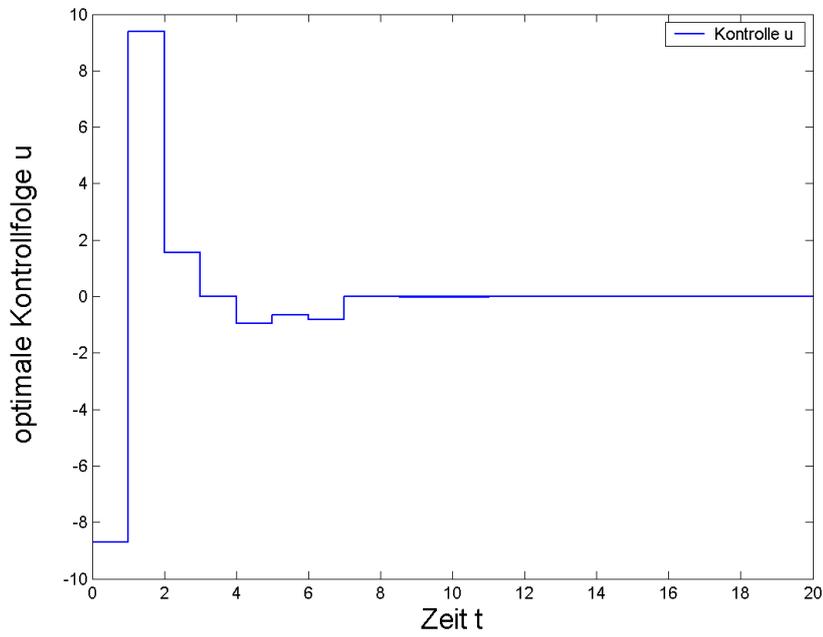
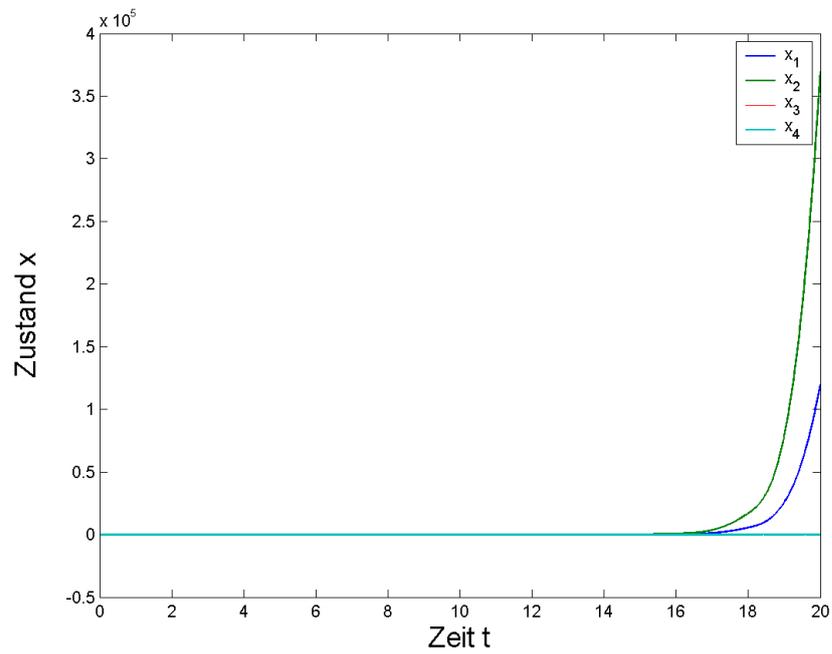
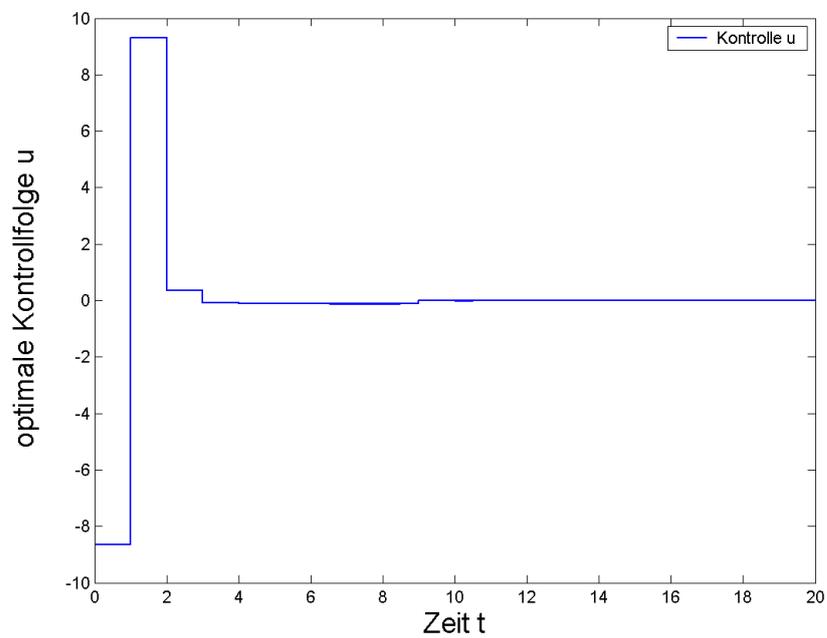
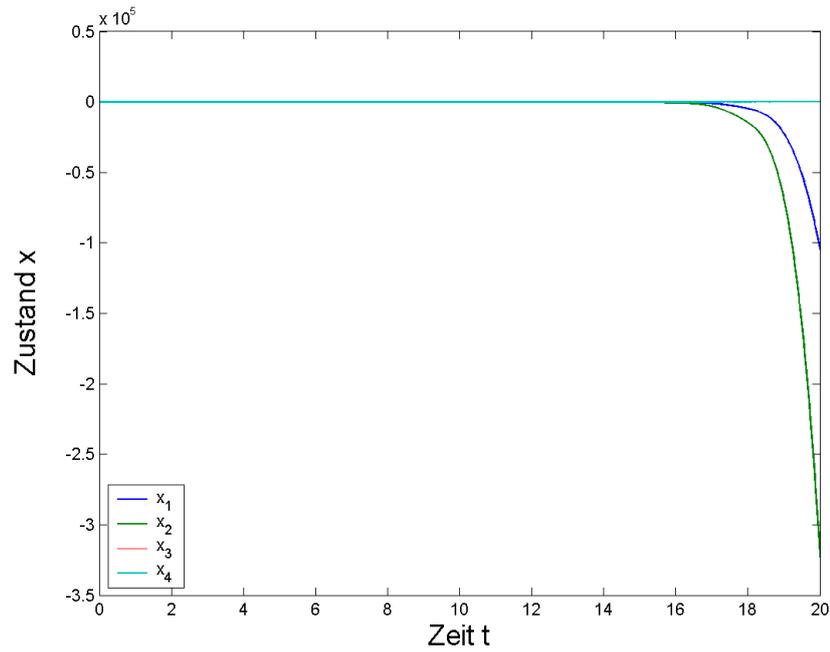
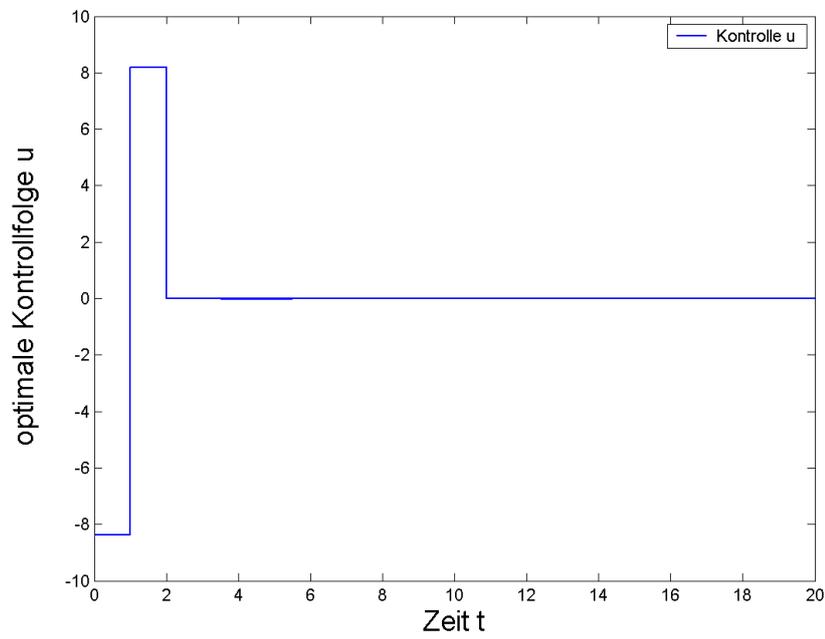


Abbildung 6.6: Optimale Kontrollfolge für $N_u = 3$ und $N_x = 15$

Abbildung 6.7: Verhalten des Zustandes für $N_u = 5$ und $N_x = 10$ Abbildung 6.8: Optimale Kontrollfolge für $N_u = 5$ und $N_x = 10$

Abbildung 6.9: Verhalten des Zustandes für $N_u = 10$ und $N_x = 10$ Abbildung 6.10: Optimale Kontrollfolge für $N_u = 10$ und $N_x = 10$

Abbildung 6.11: Verhalten des Zustandes für $N_u = 3$ und $N_x = 10$ Abbildung 6.12: Optimale Kontrollfolge für $N_u = 3$ und $N_x = 10$

Abbildung 6.13: Verhalten des Zustandes für $N_u = 3$ und $N_x = 5$ Abbildung 6.14: Optimale Kontrollfolge für $N_u = 3$ und $N_x = 5$

Ergebnisse bei unterschiedlicher Wahl der Horizontlängen N_u und N_x

Eingabeparameter	Wert
Dimension der Kontrolle m	1
Dimension des Zustandes n	4
Startzeitpunkt t_s	0
aktueller Zeitpunkt t_{akt}	0
Eintrag der Diagonalmatrix Q	2.0
Eintrag der Diagonalmatrix R	0.5
Anfangszustand x_1	0.5
Anfangszustand x_2	0.1
Anfangszustand x_3	0.1
Anfangszustand x_4	0.1

Tabelle 6.1: Eingabeparameter inverses Pendel 1

Horizontlängen		max. Δu ($=u(t+1) - u(t)$)	Zustand zum Zeitpunkt $t = N_x$			
N_u	N_x		x_1	x_2	x_3	x_4
5	15	15.7601	0.0000	-0.0000	0.0000	-0.0000
3	15	11.6327	0.0000	-0.0000	0.0000	-0.0000
5	10	20.9382	0.0001	-0.0003	-0.0000	0.0000
10	10	18.0900	0.0003	-0.0011	0.0000	-0.0000
3	10	17.9446	0.0258	0.0700	-0.0745	0.2982
3	5	16.5731	-0.0021	0.0067	-1.9215	0.0000

Tabelle 6.2: Daten inverses Pendel 1

Für dieses Beispiel ist $N_u = 5$ und $N_x = 15$ sicherlich eine gute Wahl. Es ist im Allgemeinen darauf zu achten, die Horizontlänge N_x lang genug sowie gleichzeitig die Länge des Kontrollhorizontes nicht zu kurz zu wählen.

Nun soll untersucht werden, wie die Wahl der "tuning"-Parameter Q und R die Regelung beeinflussen kann. Im diesem Teil gilt dabei für die anderen Konstanten

```
m = 1; n = 4;
N_u = 5;
N_x = 15;
t_akt = 0; t_s = 0;
x1_t_akt = 0.5; x2_t_akt = 0.1; x3_t_akt = 0.1; x4_t_akt = 0.1;
x_anfang = [x1_t_akt; x2_t_akt; x3_t_akt; x4_t_akt];
```

Das Programm liefert bei einer Wahl von $\bar{Q} = 0.5$ und $\bar{R} = 2$ folgende Ergebnisse:

Zustand_x =

Columns 1 through 8

0.5000	0.0529	-0.2745	-0.0858	-0.0174	-0.0035	-0.0006	0.0000
0.1000	-2.2216	0.6327	0.2730	0.0557	0.0115	0.0024	0.0006
0.1000	-0.8947	-1.9528	-1.9116	-1.7523	-1.5930	-1.4339	-1.2751
0.1000	-4.0789	-0.1535	0.3186	0.3187	0.3185	0.3180	0.3172

Columns 9 through 16

0.0002	0.0003	0.0004	0.0004	0.0007	0.0029	0.0140	0.0229 ...
0.0002	0.0002	0.0001	0.0001	0.0012	0.0094	0.0431	-0.0000 ...
-1.1168	-0.9590	-0.8020	-0.6459	-0.4906	-0.3354	-0.1799	-0.0597 ...
0.3161	0.3148	0.3132	0.3113	0.3099	0.3110	0.3110	0.1699 ...

Optimale_Kontrollfolge =

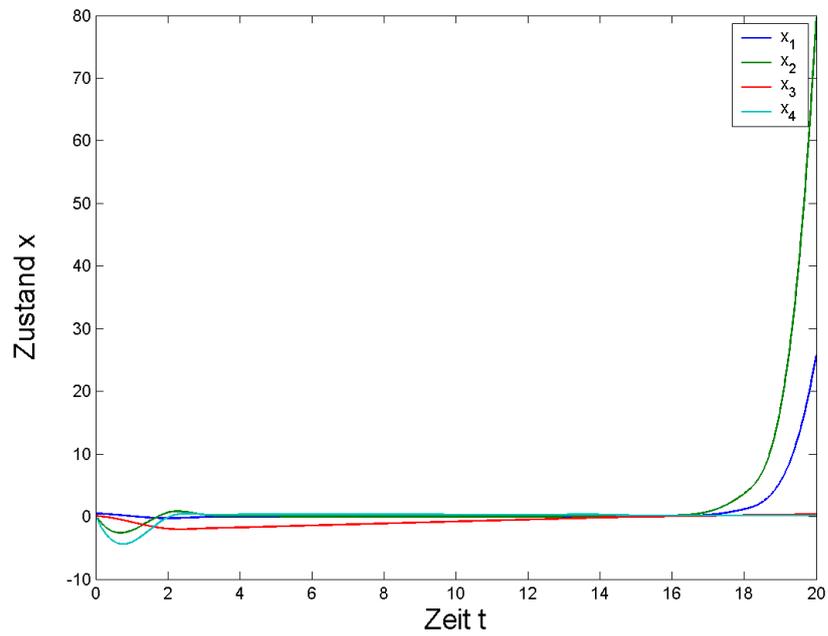
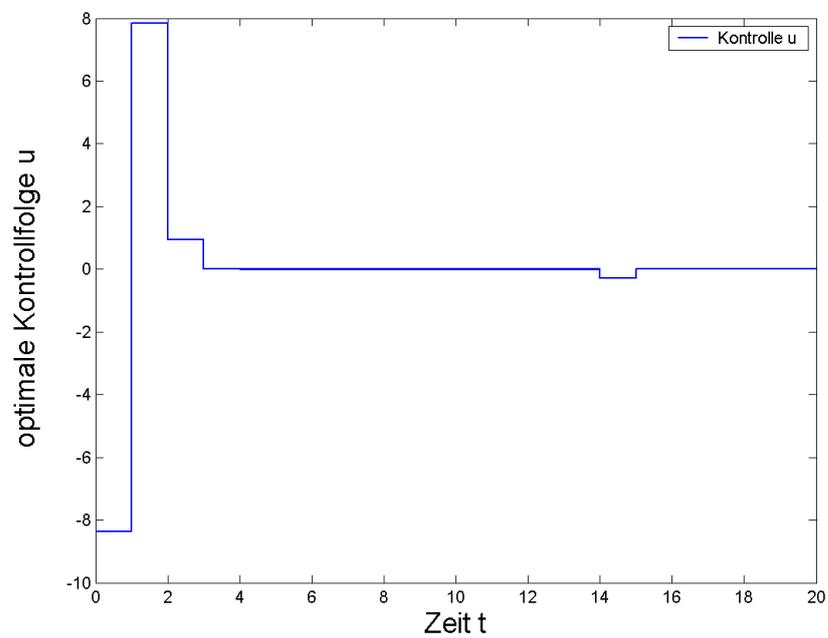
Columns 1 through 8

-8.3579	7.8509	0.9442	0.0002	-0.0004	-0.0010	-0.0016	-0.0021
---------	--------	--------	--------	---------	---------	---------	---------

Columns 9 through 15

-0.0027	-0.0032	-0.0038	-0.0027	0.0022	0.0000	-0.2821 ...
---------	---------	---------	---------	--------	--------	-------------

Das Verhalten ist hier im Vergleich zu der Wahl $\bar{Q} = 2$, $\bar{R} = 0.5$ (Abbildung 6.3) deutlich schlechter. Der Zustand kann auf dem Prädiktionshorizont der Länge 15 nicht ganz in die Null geregelt werden. Zwar nähern sich die Komponenten des Zustandsvektors der Null, erreichen diese jedoch bis auf x_3 nicht in der gegebenen Zeit. Es ist im Allgemeinen zu beachten, die 1-Norm des Zustandes $\|x(t)\|_1$ im Optimierungsproblem stärker zu gewichten als die der Kontrolle $\|u(t)\|_1$. Das heißt, die Einträge der Diagonalmatrix Q sollten im Verhältnis größer gewählt werden als \bar{R} . Dieses Problem wird im letzten Abschnitt des Kapitels näher betrachtet. Die Abbildungen 6.23 und 6.24 zeigen abschließend das Verhalten bei der Wahl eines anderen Anfangszustandes. Wie man sieht, ist auch für einen deutlich größeren Anfangszustand die Regelung in den Referenzzustand $x_{ref} = 0$ gewährleistet.

Abbildung 6.15: Verhalten des Zustandes bei $\bar{Q} = 0.5$ und $\bar{R} = 2$ Abbildung 6.16: Optimale Kontrollfolge bei $\bar{Q} = 0.5$ und $\bar{R} = 2$

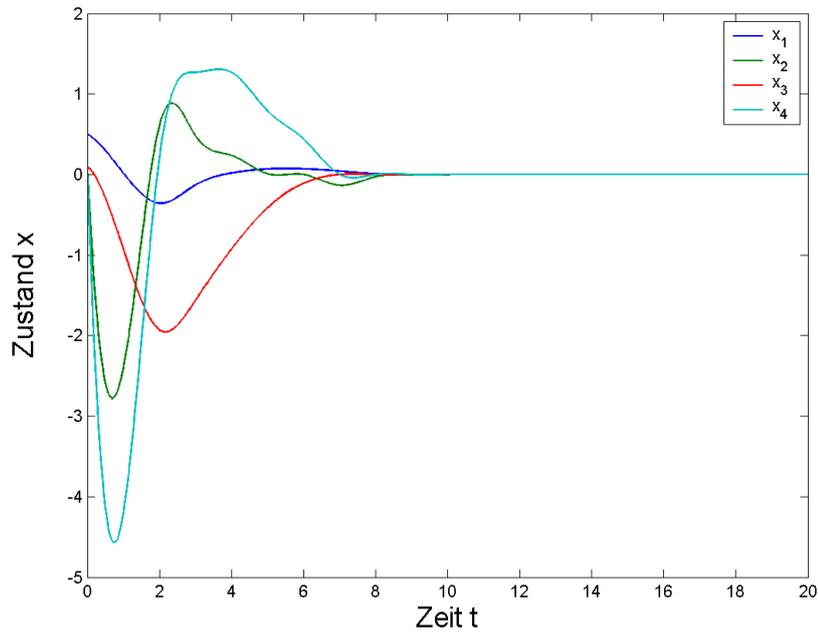


Abbildung 6.17: Verhalten des Zustandes bei $\bar{Q} = 2$ und $\bar{R} = 2$

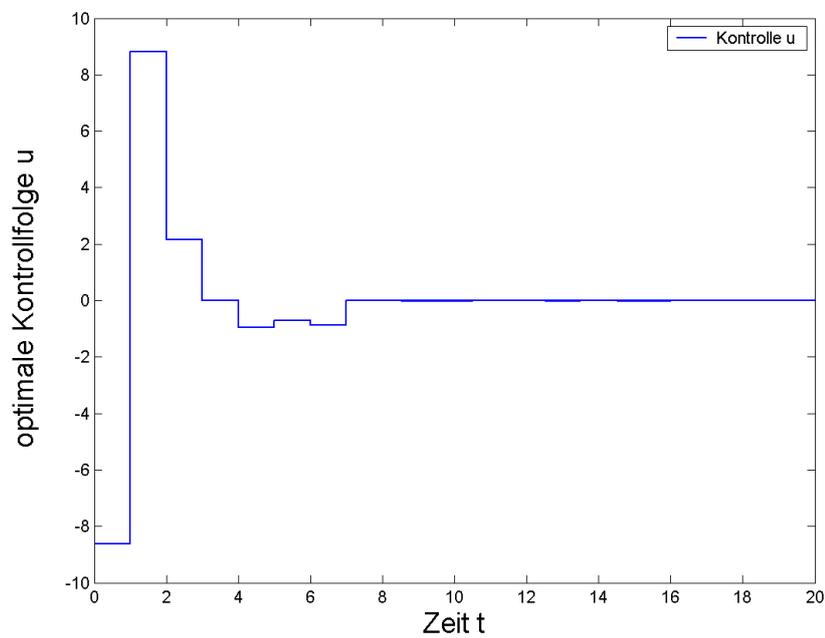
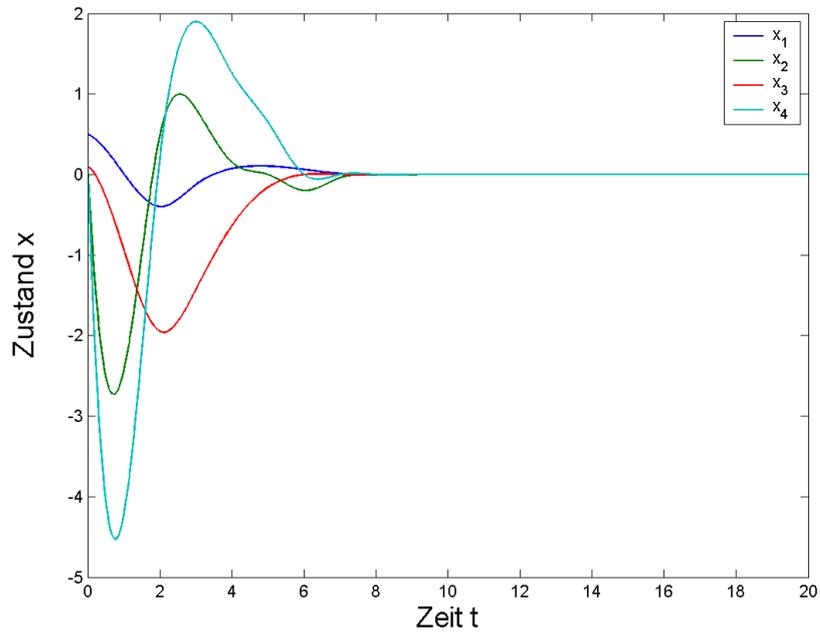
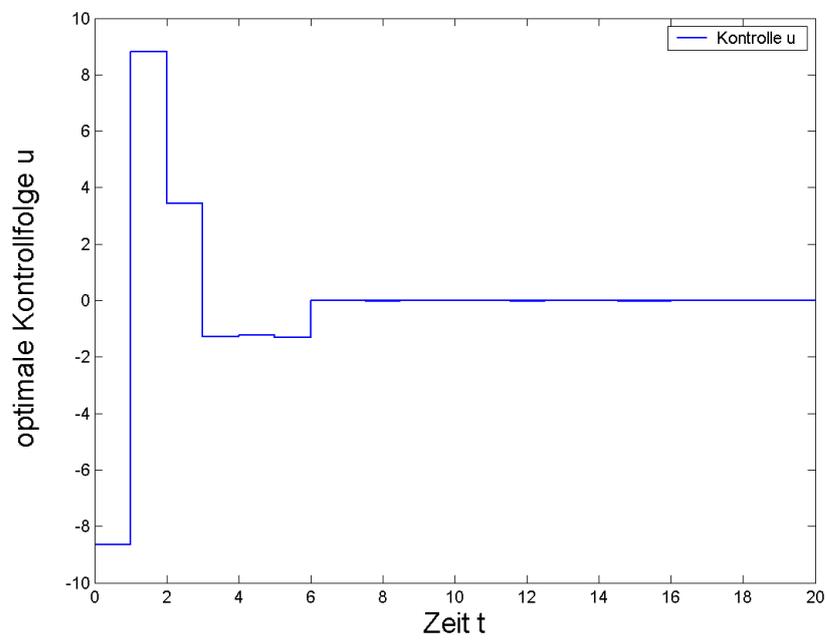
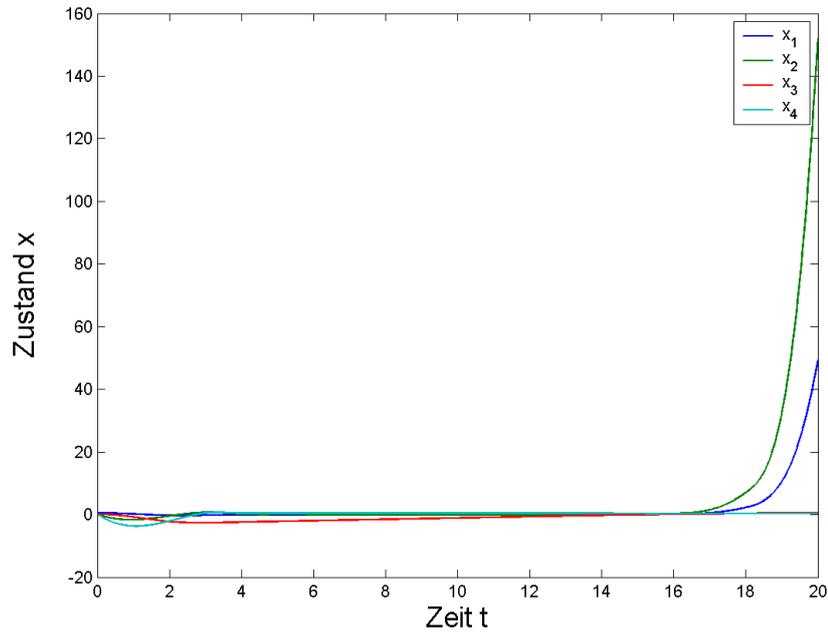
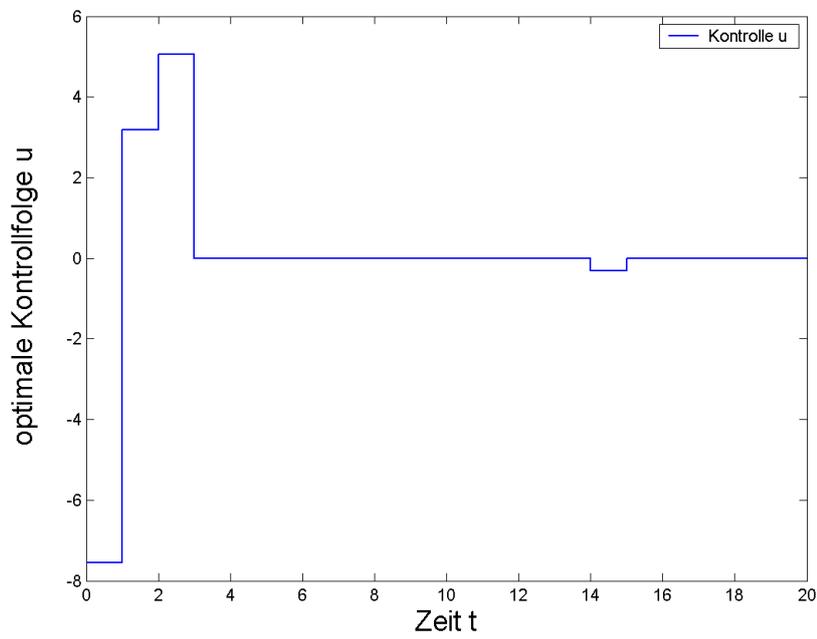


Abbildung 6.18: Optimale Kontrollfolge bei $\bar{Q} = 2$ und $\bar{R} = 2$

Abbildung 6.19: Verhalten des Zustandes bei $\bar{Q} = 5$ und $\bar{R} = 1$ Abbildung 6.20: Optimale Kontrollfolge bei $\bar{Q} = 5$ und $\bar{R} = 1$

Abbildung 6.21: Verhalten des Zustandes bei $\bar{Q} = 1$ und $\bar{R} = 5$ Abbildung 6.22: Optimale Kontrollfolge bei $\bar{Q} = 1$ und $\bar{R} = 5$

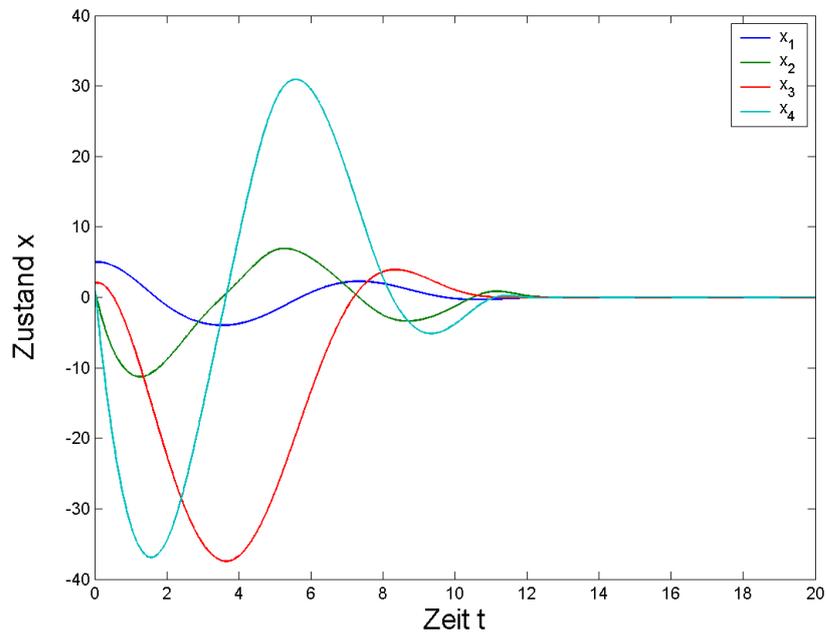


Abbildung 6.23: Verhalten für Anfangszustand $x_1 = 5$, $x_2 = 1$, $x_3 = 2$, $x_4 = 1.5$

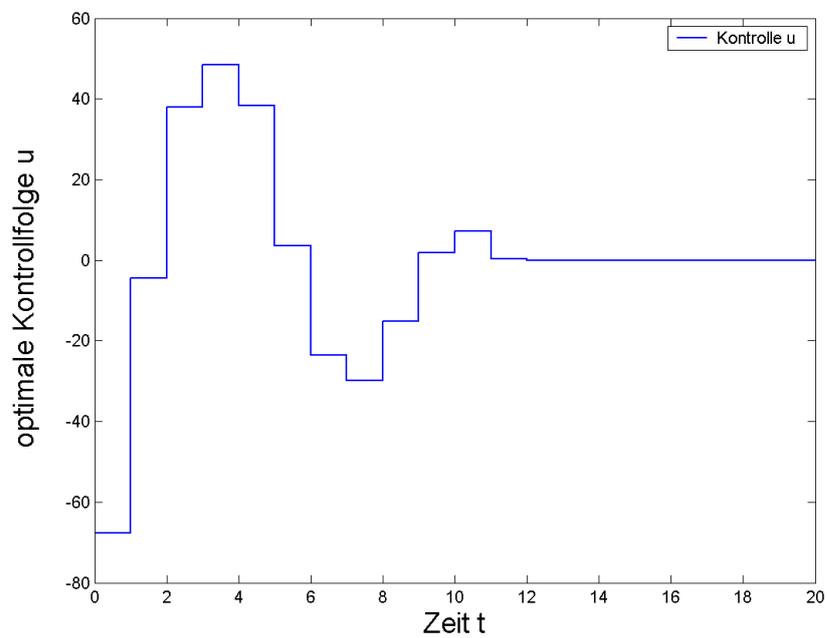


Abbildung 6.24: Kontrolle für Anfangszustand $x_1 = 5$, $x_2 = 1$, $x_3 = 2$, $x_4 = 1.5$

Ergebnisse bei unterschiedlicher Wahl der Gewichtungsmatrizen Q und R

Eingabeparameter	Wert
Dimension der Kontrolle m	1
Dimension des Zustandes n	4
Startzeitpunkt t_s	0
aktueller Zeitpunkt t_{akt}	0
Länge des Kontrollhorizontes N_u	5
Länge des Prädiktionshorizontes N_x	15
Anfangszustand x_1	0.5
Anfangszustand x_2	0.1
Anfangszustand x_3	0.1
Anfangszustand x_4	0.1

Tabelle 6.3: Eingabeparameter inverses Pendel 2

Gewichtungsmatrizen		max. Δu ($=u(t+1) - u(t)$)	Zustand zum Zeitpunkt $t = N_x$			
\bar{Q}	\bar{R}		x_1	x_2	x_3	x_4
0.5	2.0	16.2088	0.0229	-0.0000	-0.0597	0.1699
2.0	2.0	17.4152	0.0000	-0.0000	0.0000	-0.0000
5.0	1.0	17.4724	0.0000	-0.0000	-0.0000	0.0000
1.0	5.0	10.7260	0.0348	0.0284	-0.0682	0.2728

Tabelle 6.4: Daten inverses Pendel 2

6.2 Beispiel 2: Der 2-DOF Hubschrauber von Quanser

6.2.1 Das Hubschrauber-Modell

Der 2-DOF Hubschrauber der kanadischen Firma Quanser (www.quanser.com) wird Hochschulen und Universitäten seit ca. 6 Jahren als intuitiv einsetzbare und dokumentierte Versuchseinrichtung für die Grundlagenvermittlung im Bereich Regelungstechnik und Automatisierung angeboten. "2 DOF Heli[©]" ist dabei die Kurzbezeichnung für Helikopterexperiment mit 2 Freiheitsgraden in der Bewegung.

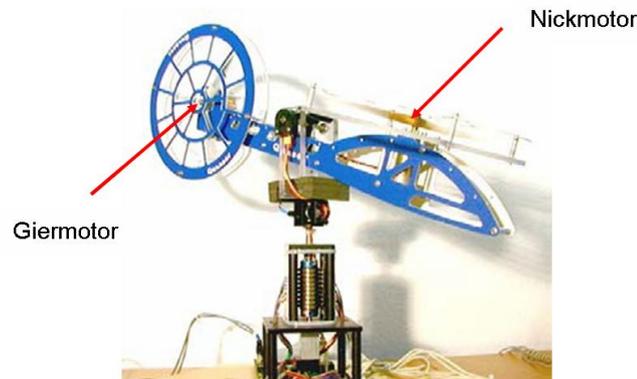


Abbildung 6.25: Das Hubschrauber-Modell

Für dieses Hubschrauber-Modell kann mit Hilfe physikalischer Überlegungen folgendes lineare Zustandsmodell aufgestellt werden.

$$\begin{pmatrix} \dot{\theta} \\ \dot{\psi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \theta \\ \psi \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ L_1 \cdot \frac{K_{fn}}{I_N} & -\frac{K_{tg}}{I_N} \\ -\frac{K_{tn}}{I_G} & L_1 \cdot \frac{K_{fg}}{I_G} \end{pmatrix} \cdot \begin{pmatrix} V_N \\ V_G \end{pmatrix}$$

$$\begin{pmatrix} \theta \\ \psi \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \theta \\ \psi \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \quad (6.7)$$

Hierbei sind der Nickwinkel θ , der Gierwinkel ψ , die Nickwinkelgeschwindigkeit $\dot{\theta}$ und die Gierwinkelgeschwindigkeit $\dot{\psi}$ die Komponenten des Zustandsvektors, auf den mithilfe der Kontrolle $u = [V_N \ V_G]^T$ Einfluss genommen werden kann. Die Größen V_N und V_G entsprechen dabei der Spannung am Gier- bzw. Nickmotor (siehe Abb. 6.25). Außerdem beschreiben L_1 die Abstände zwischen der Aufhängung und den Motoren, I_N und I_G die Trägheitsmomente des Rumpfes um Hoch- und Querachse sowie K_{fn} , K_{tg} , K_{tn} und K_{fg} Motorkonstanten im Arbeitspunkt (je nach Drehrichtung).

Mit den Konstanten $L_1 = 0.203$, $I_N = I_G = 0.03071$, $K_{fn} = 0.8722$, $K_{tg} = 0.01$, $K_{tn} = 0.02$ und $K_{fg} = 0.4214$ folgt für das System (6.7)

$$\begin{aligned} \dot{x}(t) &= \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} x(t) + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 5.765 & -0.326 \\ -0.651 & 2.786 \end{pmatrix} u(t) \\ y(t) &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} x(t) \end{aligned} \quad (6.8)$$

Die Umwandlung in ein zeitdiskretes Kontrollsystem verläuft völlig analog zu der in Abschnitt 6.1 vorgestellten Methode zur Berechnung der Matrizen A und B der zeitdiskreten Darstellung. Die Matrix C bleibt bei dieser Umformung unverändert. Diese Vorgehensweise liefert mit $t_{abt} = 0.5$ das zu (6.8) entsprechende lineare zeitdiskrete Kontrollsystem (mit Ausgang)

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \quad (6.9)$$

mit

$$A = \begin{pmatrix} 1.0000 & 0 & 0.5000 & 0 \\ 0 & 1.0000 & 0 & 0.5000 \\ 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix}, \quad (6.10)$$

$$B = \begin{pmatrix} 0.7206 & -0.0408 \\ -0.0814 & 0.3483 \\ 2.8825 & -0.1630 \\ -0.3255 & 1.3930 \end{pmatrix} \quad \text{und} \quad (6.11)$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}. \quad (6.12)$$

6.2.2 Ergebnisse der modellprädiktiven Regelung mittels linearer Programmierung

Das Ziel dieses Abschnittes ist die Stabilisierung des Hubschraubers (z.B. bei der Landung). Durch gezielte Spannungsänderungen am Nick- und Giermotor, soll der Helikopter in horizontaler Position stabilisiert werden. Die daraus resultierende Minimierung des Nick- und Gierwinkels sowie die Veränderungen der Winkelgeschwindigkeiten sollen dabei ein ungewolltes Kippen des Hubschraubers verhindern.

Der folgenden Untersuchung liegt das lineare Kontrollsystem mit Ausgang

$$\begin{aligned}x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t)\end{aligned}$$

mit (6.10), (6.11) und (6.12) zugrunde. Bei der Regelung des Ausganges $y(t)$ in den Referenzausgang $y_{ref} = 0$ werden zunächst folgende Parameter verwendet.

```
m = 2; n = 4; l = 2;
Q = 100.0*eye(1,1); R = 1.0*eye(m,m);
t_akt = 0; t_s = 0;
x1_t_akt = 0.5; x2_t_akt = 0.7; x3_t_akt = 0.2; x4_t_akt = 0.1;
x_anfang = [x1_t_akt; x2_t_akt; x3_t_akt; x4_t_akt];
y_anfang = C*x_anfang;
```

Das entwickelte Programm zur modellprädiktiven Regelung mittels linearer Programmierung liefert für $N_u = 5$ und $N_y = 10$ folgende Daten:

```
Ausgang_y =
Columns 1 through 7
    0.5000    0.2570    0.0000    0.0000    0.0000    0.0000    0.0000
    0.7000   -0.0000    0.0000    0.0000   -0.0000   -0.0000    0.0000
Columns 8 through 11
    0.0000    0.0000    0.0000    0.0000 ...
   -0.0000   -0.0000   -0.0000   -0.0000 ...
```

```
Zustand_x =
Columns 1 through 7
    0.5000    0.2570    0.0000    0.0000    0.0000    0.0000    0.0000
    0.7000   -0.0000    0.0000    0.0000   -0.0000   -0.0000    0.0000
    0.2000   -1.1719    0.1438   -0.1438    0.1438   -0.1438    0.1438
    0.1000   -2.9000    2.9000   -2.9000    2.9000   -2.9000    2.9000
```

Columns 8 through 11

0.0000	0.0000	0.0000	0.0000	...
-0.0000	-0.0000	-0.0000	-0.0000	...
-0.1438	0.1438	-0.1438	0.1438	...
-2.9000	2.9000	-2.9000	2.9000	...

Optimale_Kontrollfolge =

Columns 1 through 7

-0.6057	0.7012	-0.3397	0.3397	-0.3397	0.3397	-0.3397
-2.2952	4.3275	-4.2431	4.2431	-4.2431	4.2431	-4.2431

Columns 8 through 10

0.3397	-0.3397	0.3397	...
4.2431	-4.2431	4.2431	...

Man sieht, dass der Ausgang zunächst sehr schnell in den Referenzausgang $y_{ref} = 0$ geregelt wird. Dieses Verhalten wird bei einer Veränderung der Horizontlängen beibehalten. Hier wird (anders als in Abschnitt 6.1) der vom Zustand abhängige Ausgang und nicht der Zustand selbst in die Null geregelt. Ohne weiteres Eingreifen der Kontrolle für $t \geq 10$ kommt es jedoch zu einem deutlichen Ausschlag.

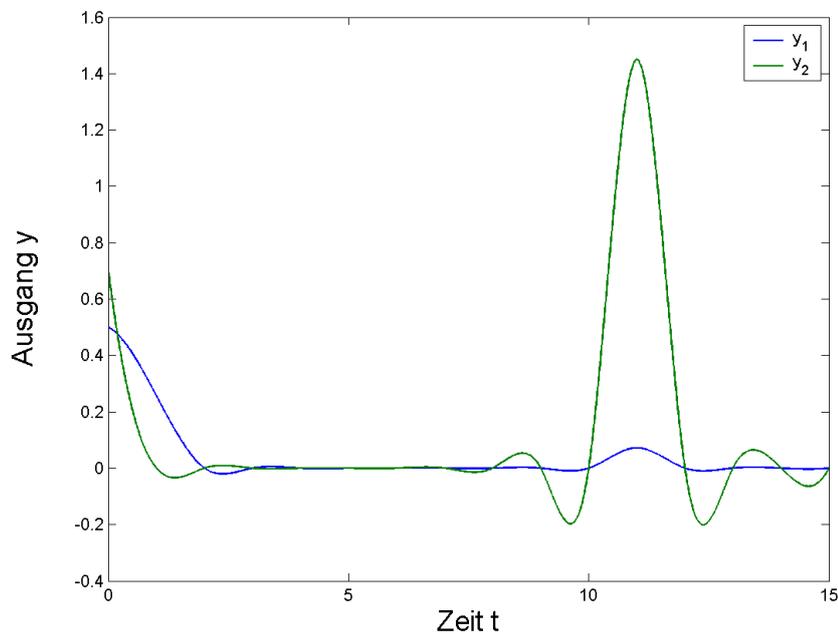
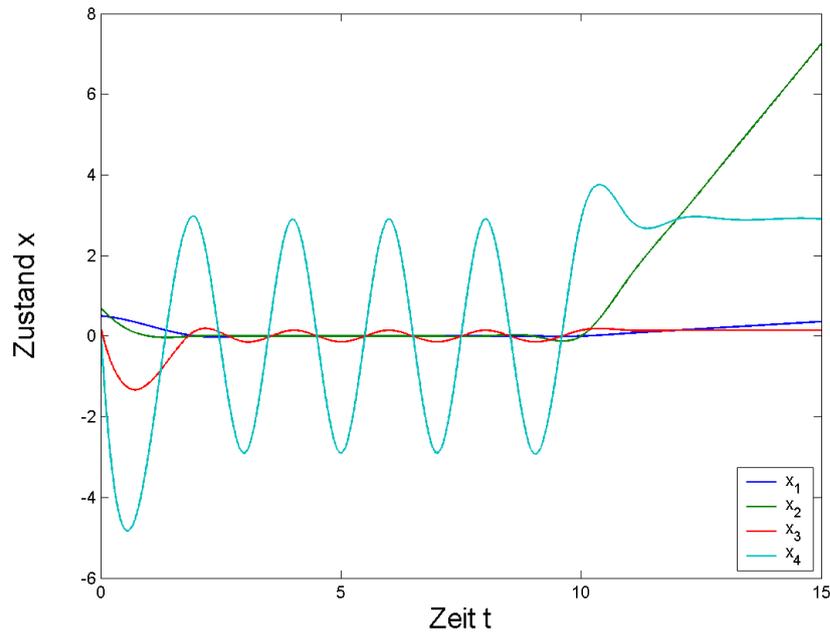
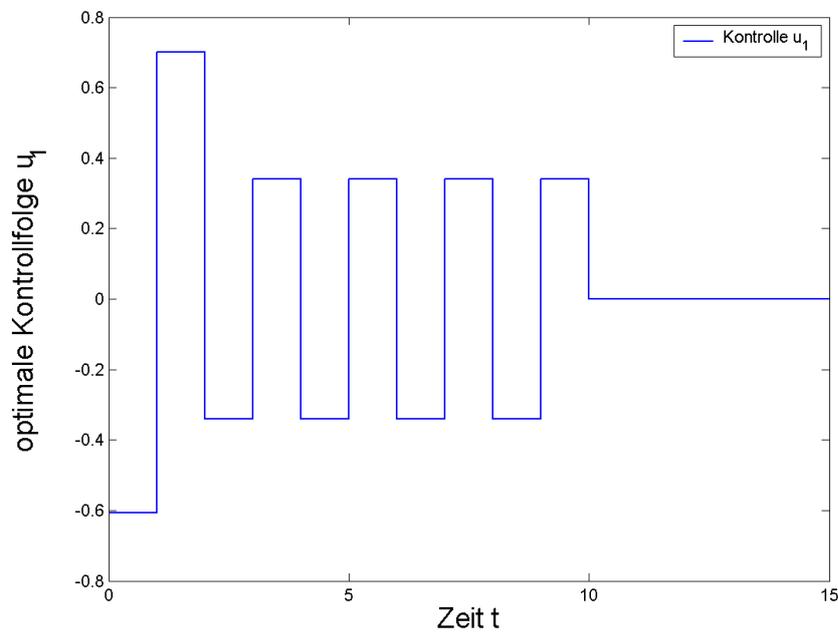
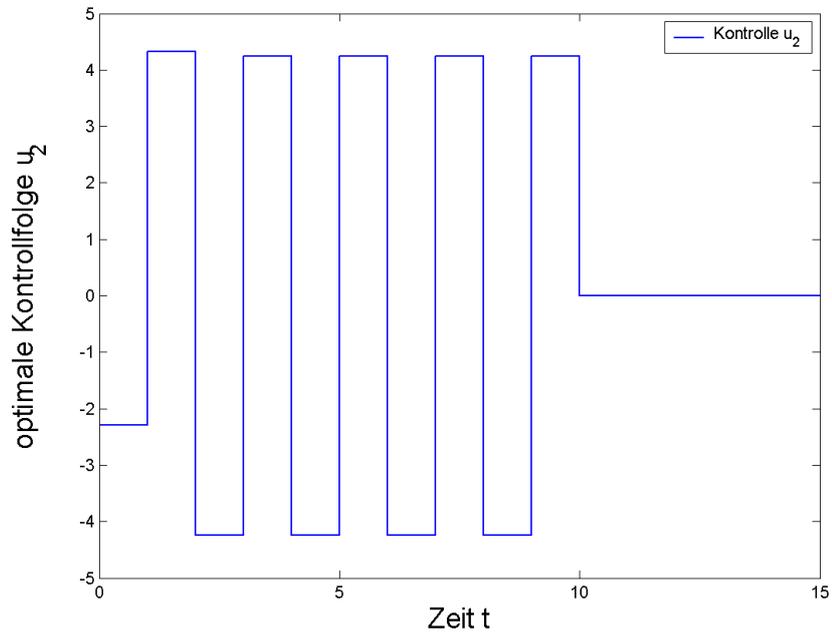
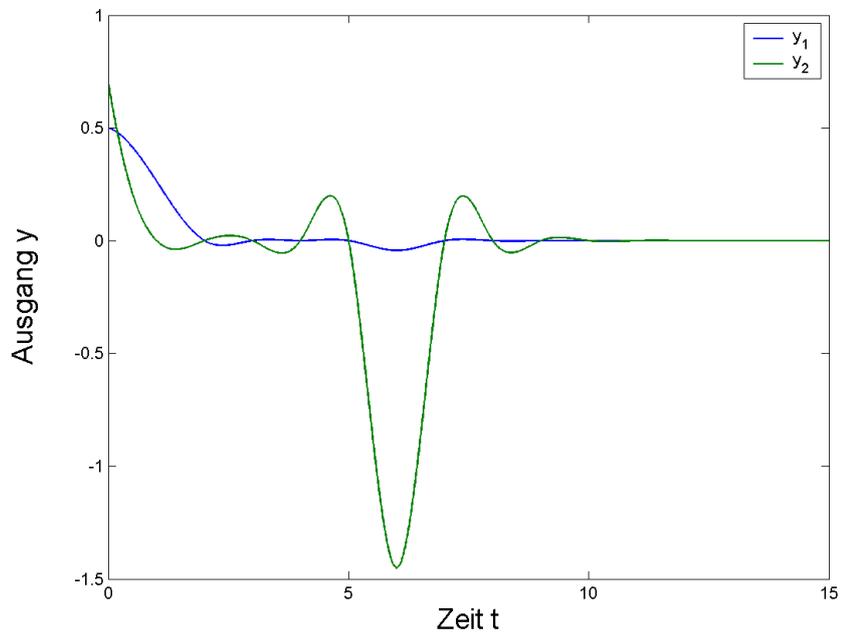
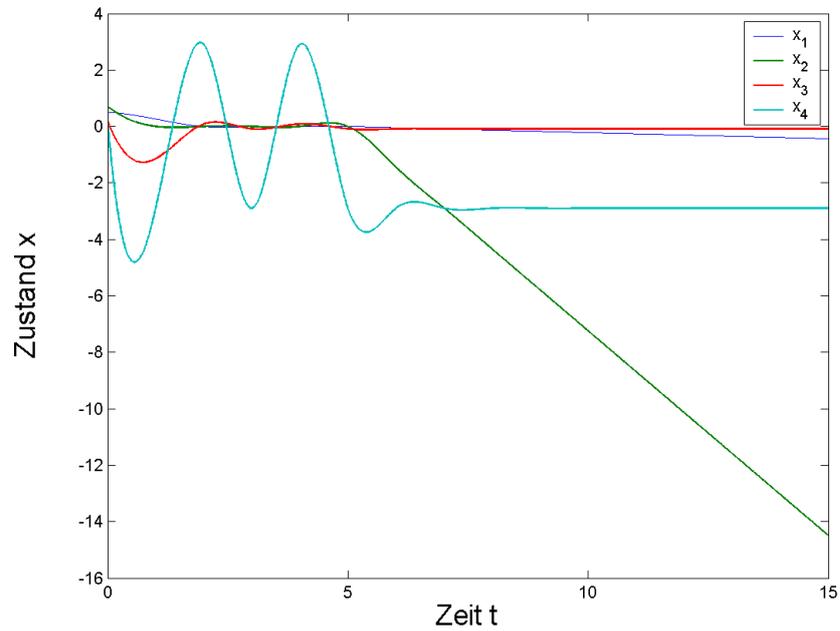
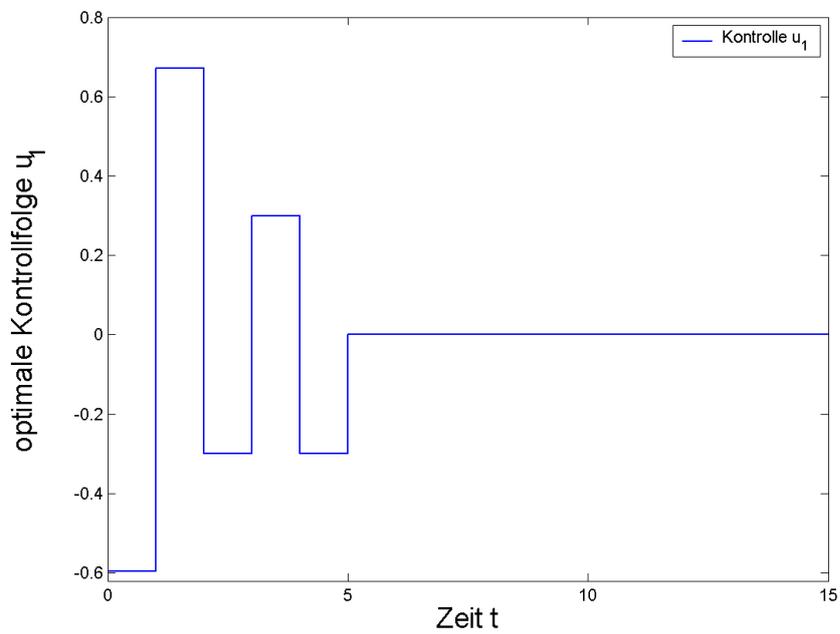
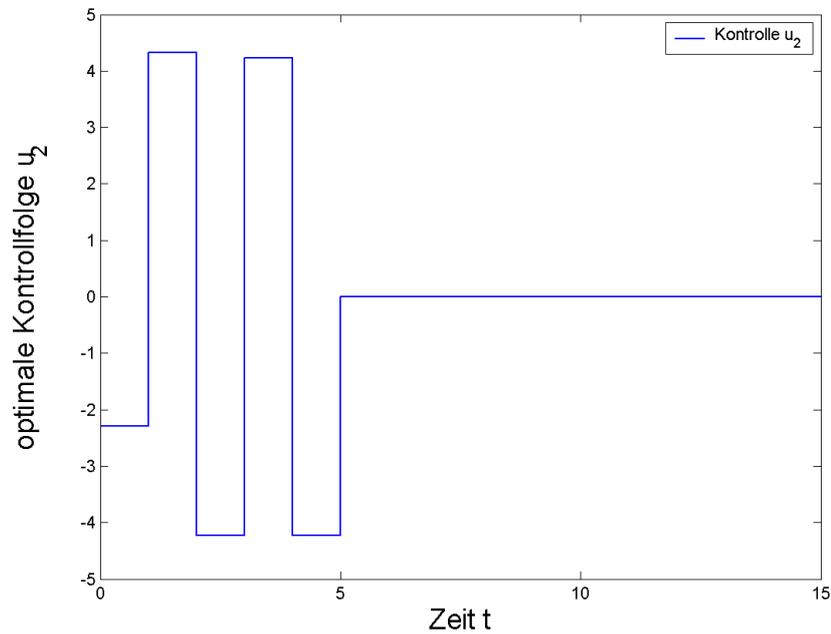


Abbildung 6.26: Verhalten des Ausganges bei $N_u = 5$ und $N_y = 10$

Abbildung 6.27: Verhalten des Zustandes bei $N_u = 5$ und $N_y = 10$ Abbildung 6.28: Optimale Kontrolle u_1 bei $N_u = 5$ und $N_y = 10$

Abbildung 6.29: Optimale Kontrolle u_2 bei $N_u = 5$ und $N_y = 10$ Abbildung 6.30: Verhalten des Ausganges bei $N_u = 3$ und $N_y = 5$

Abbildung 6.31: Verhalten des Zustandes bei $N_u = 3$ und $N_y = 5$ Abbildung 6.32: Optimale Kontrolle u_1 bei $N_u = 3$ und $N_y = 5$

Abbildung 6.33: Optimale Kontrolle u_2 bei $N_u = 3$ und $N_y = 5$

Ergebnisse bei unterschiedlicher Wahl der Horizontlängen N_u und N_x

Eingabeparameter	Wert
Dimension der Kontrolle m	2
Dimension des Zustandes n	4
Dimension des Ausganges l	2
Startzeitpunkt t_s	0
aktueller Zeitpunkt t_{akt}	0
Eintrag der Diagonalmatrix Q	100.0
Eintrag der Diagonalmatrix R	1.0
Anfangszustand x_1	0.5
Anfangszustand x_2	0.7
Anfangszustand x_3	0.2
Anfangszustand x_4	0.1

Tabelle 6.5: Eingabeparameter 2-DOF Hubschrauber 1

Horizontlängen		max. Δu		Ausgang zum Zeitpunkt $t = N_y$	
N_u	N_y	u_1	u_2	y_1	y_2
5	10	1.3069	-8.5706	0.0000	-0.0000
3	5	1.2665	-8.5540	0.0000	-0.0000

Tabelle 6.6: Daten 2-DOF Hubschrauber 1

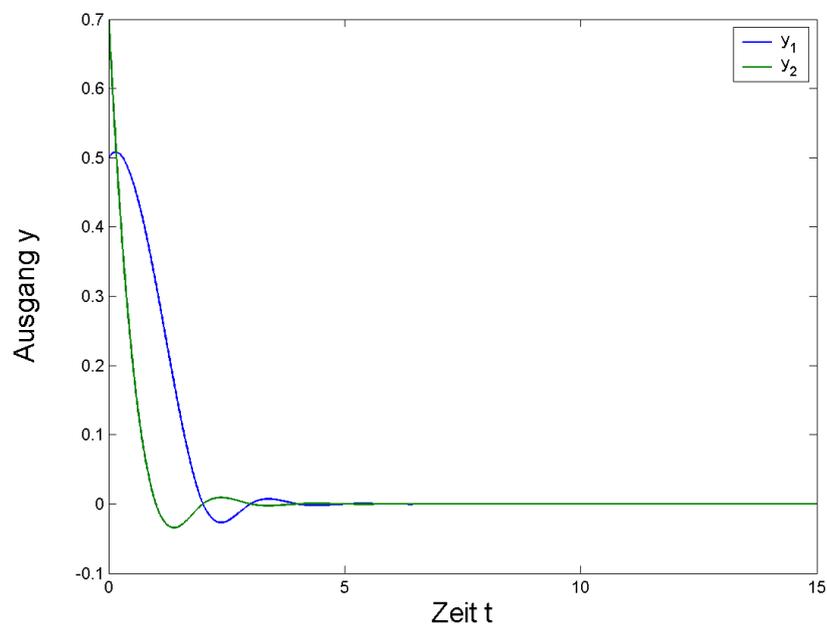
Für die Untersuchung verschiedener "tuning"-Parameter Q und R wurden für die weiteren Variablen folgende Annahmen getroffen.

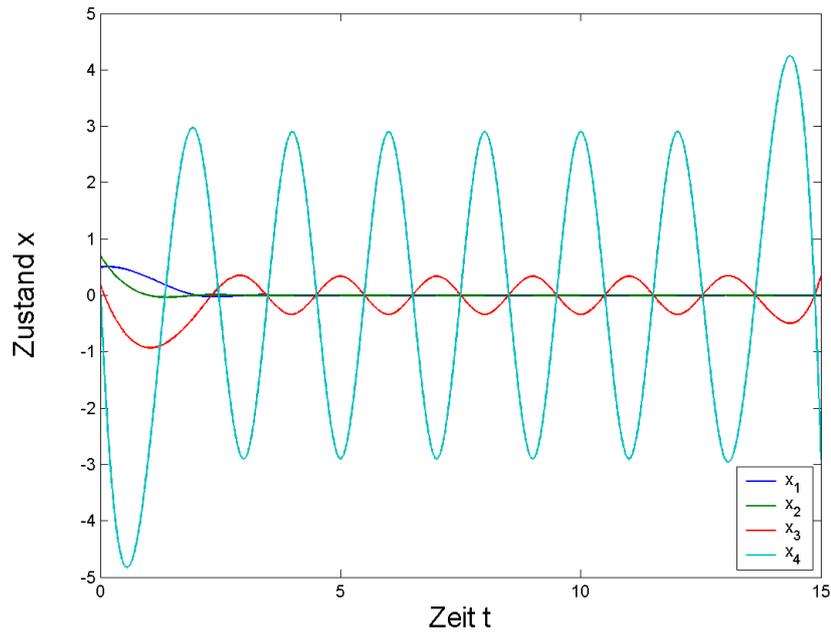
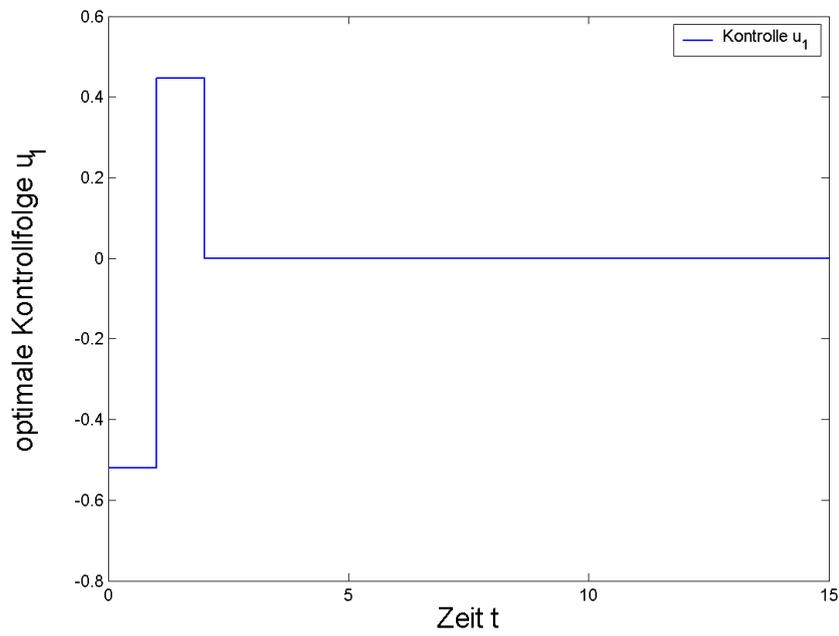
```

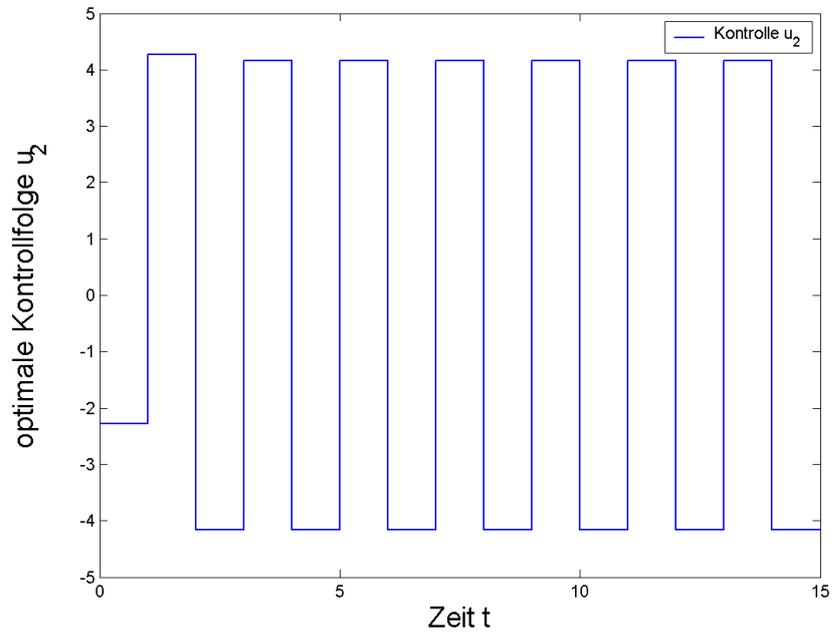
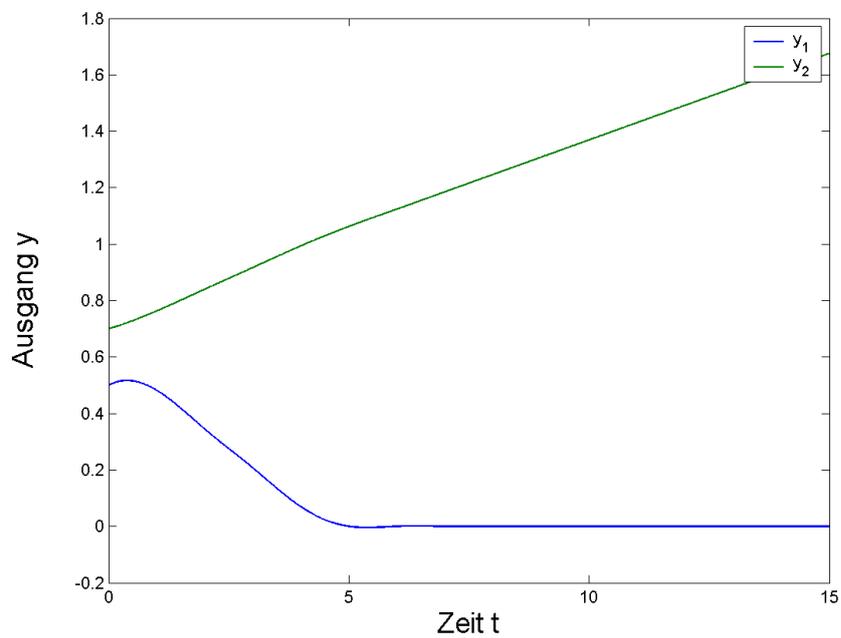
m = 2; n = 4; l = 2;
N_u = 5; N_y = 10;
t_akt = 0; t_s = 0;
x1_t_akt = 0.5; x2_t_akt = 0.7; x3_t_akt = 0.2; x4_t_akt = 0.1;
x_anfang = [x1_t_akt; x2_t_akt; x3_t_akt; x4_t_akt];

```

Die Auswertungen bei unterschiedlich gewählten Gewichtungsmatrizen Q und R ergeben Ähnliches wie in Abschnitt 6.1.2. Die Auswirkungen einer falschen Wahl sind hier jedoch deutlich stärker. Wie in Abbildung 6.38 zu sehen, wird der Ausgang nicht in den Referenzausgang $y_{ref} = 0$ geregelt. Hier nähert sich y_2 nicht einmal der Null, sondern wächst mit fortschreitender Zeit.

Abbildung 6.34: Verhalten des Ausganges bei $\bar{Q} = 2$ und $\bar{R} = 0.5$

Abbildung 6.35: Verhalten des Zustandes bei $\bar{Q} = 2$ und $\bar{R} = 0.5$ Abbildung 6.36: Optimale Kontrolle u_1 bei $\bar{Q} = 2$ und $\bar{R} = 0.5$

Abbildung 6.37: Optimale Kontrolle u_2 bei $\bar{Q} = 2$ und $\bar{R} = 0.5$ Abbildung 6.38: Verhalten des Ausgangs bei $\bar{Q} = 0.5$ und $\bar{R} = 2$

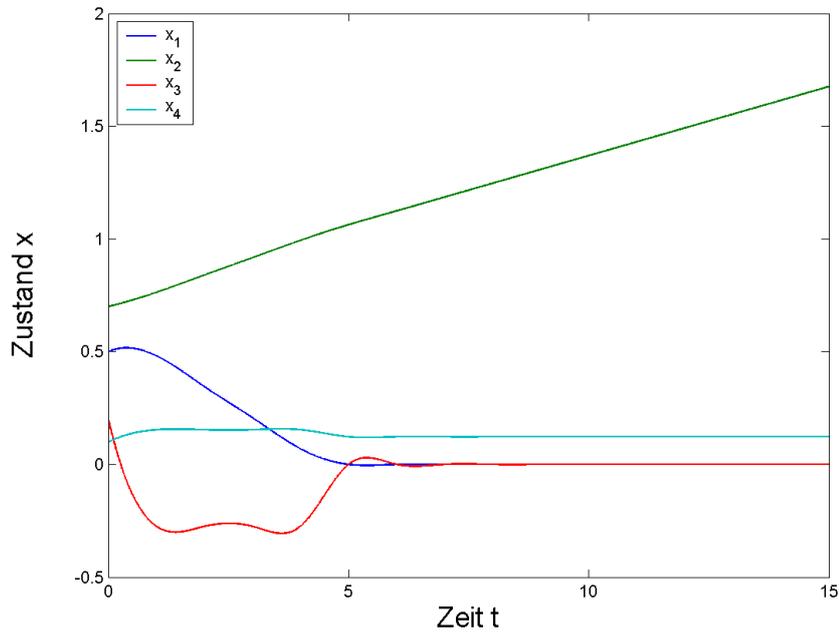


Abbildung 6.39: Verhalten des Zustandes bei $\bar{Q} = 0.5$ und $\bar{R} = 2$

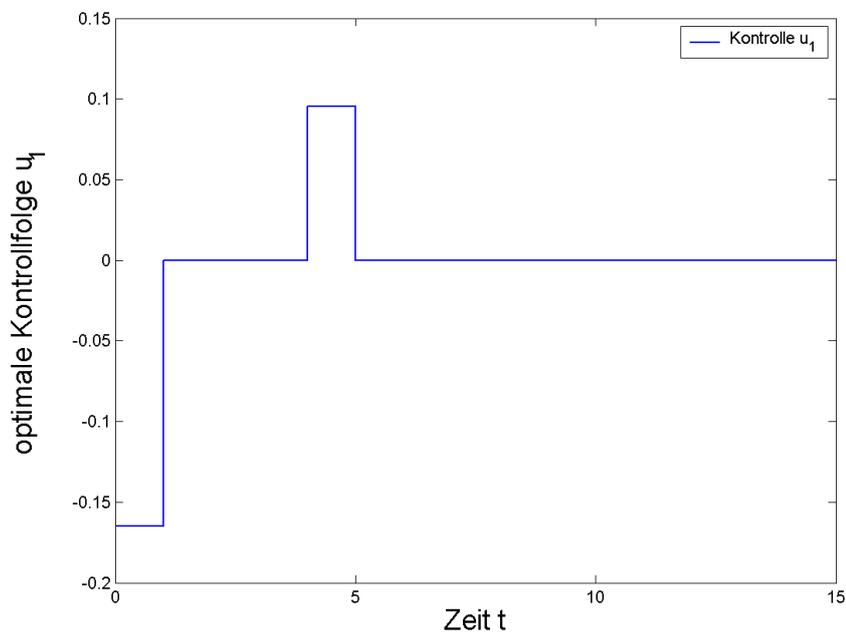
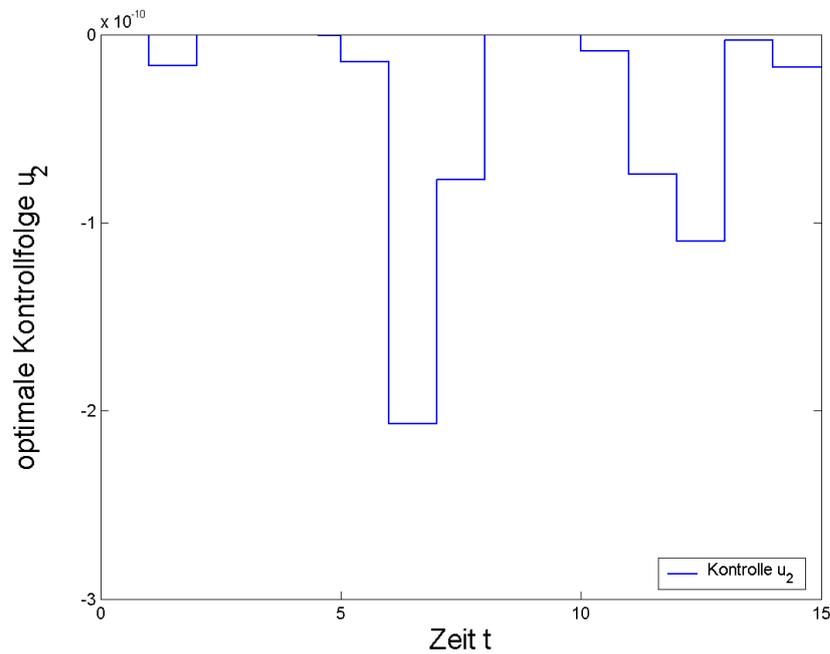


Abbildung 6.40: Optimale Kontrolle u_1 bei $\bar{Q} = 0.5$ und $\bar{R} = 2$

Abbildung 6.41: Optimale Kontrolle u_2 bei $\bar{Q} = 0.5$ und $\bar{R} = 2$

Ergebnisse bei unterschiedlicher Wahl der Gewichtungsmatrizen Q und R

Eingabeparameter	Wert
Dimension der Kontrolle m	2
Dimension des Zustandes n	4
Dimension des Ausganges l	2
Startzeitpunkt t_s	0
aktueller Zeitpunkt t_{akt}	0
Länge des Kontrollhorizontes N_u	5
Länge des Prädiktionshorizontes N_y	10
Anfangszustand x_1	0.5
Anfangszustand x_2	0.7
Anfangszustand x_3	0.2
Anfangszustand x_4	0.1

Tabelle 6.7: Eingabeparameter 2-DOF Hubschrauber 2

Gewichtungsmatrizen		max. Δu		Ausgang zum Zeitpunkt $t = N_y$	
\bar{Q}	\bar{R}	u_1	u_2	y_1	y_2
2.0	0.5	0.9672	-8.4317	0.0000	-0.0000
0.5	2.0	0.1648	0.0000	-0.0000	1.3694

Tabelle 6.8: Daten 2-DOF Hubschrauber 2

Im folgenden Abschnitt soll die Wahl der Gewichtungsmatrizen Q und R näher untersucht werden. Es kann an diesem Punkt zunächst festgehalten werden, dass das Verhältnis der Einträge das Verhalten der Lösungen stark beeinflusst und deshalb bei der Wahl Vorsicht geboten ist.

6.3 Wahl der Gewichtungsmatrizen

Um das gewünschte Verhalten des Zustandes (bzw. Ausgangs) zu erzielen, spielt die Wahl der "tuning"-Parameter Q und R eine wichtige Rolle. Man kann im Allgemeinen festhalten, dass man gute Ergebnisse erzielt, wenn die Matrix Q im Verhältnis zu R relativ groß gewählt wird. Folgendes Beispiel soll diese Idee veranschaulichen.

Betrachte folgendes eindimensionale Optimierungsproblem

$$\begin{aligned} \min_{u_0} \Theta &= q|x_1| + r|u_0| & (6.13) \\ \text{s.t.} \quad x_1 &= ax_0 + bu_0 \end{aligned}$$

Abbildung 6.42 zeigt den Graphen von Θ als eine Funktion von u_0 . Es sei nun $q = 1$. Man sieht leicht, dass die optimale Kontrolle $u_0 = 0$ erzielt wird, wenn $r > b$. Genauso erhält man bei der Wahl $r < b$ die optimale Lösung $u_0 = -\frac{ax_0}{b}$. Gilt jedoch $r = b$, ist die Lösung nicht eindeutig. In diesem Fall sind sowohl $0 \leq u_0 \leq -\frac{ax_0}{b}$ als auch $-\frac{ax_0}{b} \leq u_0 \leq 0$ optimale Lösungen.

Für dieses Beispiel erhält man also bei unterschiedlicher Gewichtung der in (6.13) auftretenden Beträge auch verschiedene Lösungen

In dem der modellprädiktiven Regelung zugrunde liegende Optimierungsproblem der Form

$$\min_u \sum_{t=N_1}^{N_2} Q \underbrace{\|x(t) - x_{ref}(t)\|_1}_{\text{Ziel: } x(t)=x_{ref}(t)} + \sum_{t=N_3}^{N_4} R \|u(t)\|_1$$

steht aufgrund der Zielsetzung die Minimierung des Ausdrucks $\|x(t) - x_{ref}(t)\|_1$ im Vordergrund. Bei der Wahl $\bar{R} > \bar{Q}$ wird der zweite Ausdruck $\|u(t)\|_1$ stärker gewichtet, was

zu einer Lösung $u(t) \approx 0$ führt. Dies minimiert zwar den Kontrollaufwand, wird jedoch im Allgemeinen nicht das gewünschte Verhalten ($x(t) = x_{ref}(t)$) hervorbringen.

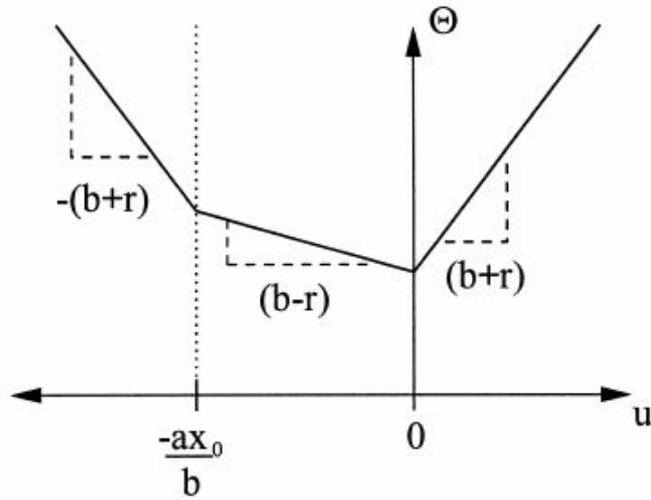


Abbildung 6.42: Graph der Zielfunktion Θ

Anhang A

Inhalt der CD

Die verwendeten Programme zur modellprädiktiven Regelung mittels linearer Programmierung sowie die in Kapitel 6 angegebenen Systeme sind auf der beigefügten CD enthalten. Außerdem beinhaltet diese das Verzeichnis `diplomarbeit` mit der vorliegenden Ausarbeitung in pdf-Format (`diplomarbeit.pdf`):

Verzeichnis `ohne_Ausgang`

Dateiname	Aufgabe
<code>c2d_A.m</code>	Funktion zur Berechnung der Matrix A des zeitdiskreten Systems (Modifikation der Funktion <code>c2d.m</code>)
<code>c2d_B.m</code>	Funktion zur Berechnung der Matrix B des zeitdiskreten Systems (Modifikation der Funktion <code>c2d.m</code>)
<code>matrix_E1.m</code>	Funktion zur Konstruktion der Matrix E_1 (5.19)
<code>matrix_E1_1.m</code>	Unterfunktion zur Konstruktion der Matrix E_1
<code>matrix_E1_2.m</code>	Unterfunktion zur Konstruktion der Matrix E_1
<code>matrix_E2.m</code>	Funktion zur Konstruktion der Matrix E_2 (5.20) bzw. (5.44)
<code>matrix_E3.m</code>	Berechnung der Matrix E_3 (5.21)
<code>matrix_E4.m</code>	Berechnung der Matrix E_4 (5.22) bzw. (5.46)
<code>matrix_E11.m</code>	Unterfunktion zur Berechnung der Teilmatrix E_{11} (für <code>matrix_E1.m</code>)
<code>matrix_E12_bis_E1Nx.m</code>	Unterfunktion zur Berechnung der Teilmatrizen E_{12} bis E_{1N_x} (für <code>matrix_E1.m</code>)
<code>matrix_E21.m</code>	Unterfunktion zur Berechnung der Teilmatrix E_{21} (für <code>matrix_E2.m</code>)
<code>matrix_Z1.m</code>	Berechnung der Matrix Z_1 (5.23)
<code>matrix_Z2.m</code>	Berechnung der Matrix Z_2 (5.24)

programm_AB.m	Programm zur modellprädiktiven Regelung mittels linearer Programmierung, Voraussetzung: lineares zeitinvariantes Kontrollsystem ohne Ausgang
system_1.m	Beispiel inverses Pendel
system_2.m	Beispiel doppeltes inverses Pendel
vektor_d1.m	Berechnung des Vektors d_1 (5.26)
vektor_d2.m	Berechnung des Vektors d_2 (5.27) bzw. (5.51)
vektor_f.m	Berechnung des Vektors f (5.16)
vektor_T1.m	Funktion für graphische Ausgabe der optimalen Kontrollfolge

Tabelle A.1: Übersicht der Matlab-Programme 1

Verzeichnis mit_Ausgang

Dateiname	Funktion
c2d_A.m	siehe Tabelle A.1
c2d_B.m	siehe Tabelle A.1
matrix_E1.m	Funktion zur Konstruktion der Matrix E_1 (5.43)
matrix_E1_1.m	Unterfunktion zur Konstruktion der Matrix E_1
matrix_E1_2.m	Unterfunktion zur Konstruktion der Matrix E_1
matrix_E2.m	siehe Tabelle A.1
matrix_E3.m	Berechnung der Matrix E_3 (5.45)
matrix_E4.m	siehe Tabelle A.1
matrix_E11.m	Unterfunktion zur Berechnung der Teilmatrix E_{11} (für matrix_E1.m)
matrix_E12_bis_E1Ny.m	Unterfunktion zur Berechnung der Teilmatrizen E_{12} bis E_{1N_y} (für matrix_E1.m)
matrix_E21.m	siehe Tabelle A.1
matrix_Z1.m	Berechnung der Matrix Z_1 (5.47)
matrix_Z2.m	Berechnung der Matrix Z_2 (5.48)
programm_ABC.m	Programm zur modellprädiktiven Regelung mittels linearer Programmierung, Voraussetzung: lineares zeitinvariantes Kontrollsystem mit Ausgang
system_3.m	Beispiel 2-DOF Hubschrauber
vektor_d1.m	Berechnung des Vektors d_1 (5.50)
vektor_d2.m	siehe Tabelle A.1
vektor_f.m	Berechnung des Vektors f (5.40)
vektor_T1.m	Funktion für graphische Ausgabe der optimalen Kontrollfolge

Tabelle A.2: Übersicht der Matlab-Programme 2

Literaturverzeichnis

- [1] R. K. Ahuja, J. B. Orlin, *Inverse Optimization*, Operations Research, Vol. 49, No. 5, pp. 771–183, 2001.
- [2] J. Akesson, *Operator Interaction and Optimization in Control Systems*, Lund University, 2003. (<http://www.controll.lth.se/~jakesson/publications.html>)
- [3] B. Aulbach, *Gewöhnliche Differenzialgleichungen*, Elsevier Spektrum Akademischer Verlag, Heidelberg, 2. Auflage, 2004.
- [4] A. Bemporad, F. Borrelli, M. Morari, *Model Predictive Control based on Linear Programming – The Explicit Solution*, IEEE Transactions on Automatic Control, Vol. 47, No. 12, pp. 1975–1984, 2002.
- [5] E. F. Camacho, C. Bordons, *Model Predictive Control*, Springer Verlag, London, 1999.
- [6] M. Chidambaram, *Computer Control of Processes*, Alpha Science International, Pangbourne, 2002.
- [7] F. Colonus, *Kontrolltheorie 1*, Vorlesungsskript, Universität Augsburg, 2003. (<http://www.scmath.uni-augsburg.de/~colonus/Lehre.html>)
- [8] R. Dittmar, B. M. Pfeiffer, *Modellbasierte prädiktive Regelung*, Oldenbourg Verlag, München/Wien, 2004
- [9] H. Fu, M. K. Ng, M. Nikolova, J. L. Barlow, *Efficient Minimization Methods of Mixed l_2 - l_1 and l_1 - l_1 Norms for Image Restoration*, SIAM Journal on Scientific Computing, Vol. 27, No. 6, pp. 1881–1902, 2006.
- [10] C. E. Garcia, D. M. Preth, M. Morari, J. H. Lee, *Model Predictive Control: Theory and Practice – A Survey*, Automatica, Vol. 25, pp. 335–348, 1989.
- [11] M. Grötschel, *Lineare Optimierung*, Vorlesungsskript, Technische Universität Berlin, 2004. (<http://www.zib.de/groetschel/teaching/materials.html>)
- [12] L. Grüne, *Mathematische Kontrolltheorie I : Lineare Systeme*, Vorlesungsskript, Universität Bayreuth, 2006. (<http://www.math.uni-bayreuth.de/~lgruene/lehre>)

- [13] L. Grüne, *Mathematische Kontrolltheorie II : Nichtlineare Systeme*, Vorlesungsskript, Universität Bayreuth, 2006. (<http://www.math.uni-bayreuth.de/~lgruene/lehre>)
- [14] L. Grüne, *Numerik Dynamischer Systeme*, Vorlesungsskript, Universität Bayreuth, 2006. (<http://www.math.uni-bayreuth.de/~lgruene/lehre>)
- [15] L. Grüne, *Stabilität und Stabilisierung linearer Systeme*, Vorlesungsskript, Universität Bayreuth, 2003. (<http://www.math.uni-bayreuth.de/~lgruene/lehre>)
- [16] F. Jarre, J. Stoer, *Optimierung*, Springer Verlag, Berlin, 2004.
- [17] H.-W. Knobloch, H. Kwakernaak, *Lineare Kontrolltheorie*, Springer Verlag, Berlin, 1985.
- [18] P. Krauss *Prädiktive Regelung mit linearen Prozessmodellen im Zustandsraum*, Fortschritt-Berichte VDI, Reihe 8, Nr. 560, VDI-Verlag, Düsseldorf, 1995.
- [19] H. Kwakernaak, R. Sivan, *Linear Optimal Control Systems*, Wiley-Interscience, New York, 1972.
- [20] K. Marti, D. Gröger, *Einführung in die lineare und nichtlineare Optimierung*, Physica-Verlag, Heidelberg, 2000.
- [21] K. R. Muske, J. B. Rawlings, *Model Predictive Control with Linear Models*, AIChE Journal, Vol. 39, NO. 2, pp. 262–278, 1993.
- [22] J. Polzer, *Berechnung der Nulldynamik eines inversen Pendels mit unterschiedlichen mathematischen Theorien*, Forschungsbericht, Universität Duisburg, 2000. (http://www.uni-duisburg-essen.de/imperia/md/content/srs/forschung/msrt_paper/2000/fb01-00.pdf)
- [23] C. V. Rao, J. B. Rawlings, *Linear Programming and Model Predictive Control*, Journal of Process Control, Vol. 10, pp. 283–289, 2000.
- [24] R. Reemtsen, *Lineare Optimierung – Eine Vorlesung*, Shaker Verlag, Aachen, 2001.
- [25] J. A. Rossiter, *Model-Based Predictive Control – A Practical Approach*, CRC Press, Boca Raton, 2003.
- [26] E. D. Sontag, *Mathematical Control Theory*, Springer Verlag, New York, 2nd Ed., 1998.
- [27] F. Svaricek, *Moderne Methoden der Regelungstechnik I*, Vorlesungsskript, Universität der Bundeswehr München, 2008. (http://www.unibw.de/lrt13/srt/lehre/unterlagen/MMR_I)

- [28] J. Vada, O. Slupphaug, T. A. Johansen, B. A. Foss, *Linear MPC with Optimal Prioritized Infeasibility Handling: Application, Computational Issues and Stability*, Automatica, 2001.
(http://www.itk.ntnu.no/ansatte/Johansen_Tor.Arne/optimization.html)
- [29] R. J. Vanderbei, *Linear Programming: Foundations and Extensions*, Online–Buch, 2nd Ed., 2001. (<http://www.princeton.edu/~rvdb/LPbook/onlinebook.pdf>)
- [30] J. Verschelde, *Linear Programming in Matlab*, Vorlesung MATLAB, University of Illinois at Chicago, 2007. (<http://www.math.uic.edu/~jan/mcs320s07/matlec9.pdf>)
- [31] L. A. Zadeh, L. H. Whalen, *On optimal Control and Linear Programming*, IEEE Transactions on Automatic Control, Vol. 7, No. 4, pp. 45–46, 1962.

Eidesstattliche 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, 8. Mai 2008

.....

Susanne Bauer