UNIVERSITÄT
BAYREUTH

MPC Schemata mit variablem Kontrollhorizont

Diplomarbeit

von

Harald Voit

FAKULTÄT FÜR MATHEMATIK UND PHYSIK
MATHEMATISCHES INSTITUT

Datum: 21. Januar 2008

Aufgabenstellung:
Prof. Dr. L. Grüne

Inhaltsverzeichnis

1 Einleitung

Modellprädiktive Regelungen begannen ihren Siegeszug im Jahre 1987 mit der Entwicklung der Generalized Predictive Control Methode, kurz GPC. Doch GPC war nicht der erste Algorithmus mit der Idee eines receding horizon. Diese Idee kann bis in die sechziger Jahre des letzten Jahrhunderts zurückverfolgt werden. Die früheste Form von MPC wurde 1963 von Propoi (Ref. [20]) vorgestellt. Praktische Bedeutung bekam diese Vorgehensweise aber erst mit der Einführung von Dynamic Matrix Control (DMC) und Identification and Command (IDCOM) in der Petrochemie. Die beiden letztgenannten sind eher heuristische Verfahren und besitzen nur eine sehr kleine theoretische Basis, was ihre Stabilität und Robustheit anbelangt. GPC hingegen wurde von vielen Forschern hinsichtlich Störungsanfälligkeit, Robustheit und Stabilität sowie vielen weiterern Aspekten eingehend untersucht. Da dies alles Methoden waren, die auf linearen Modellen basierten, begann man Ende der achtziger Jahre auch Systeme mit nichtlinearen Modellen zu untersuchen. In [16] ist ein Stabilitätsresultat für ein nichtlineares zeitdiskretes System mit Beschränkungen und einer Endbeschränkung angegeben. Heute werden NMPC (Nonlinear MPC) Anwendungen in vielen Bereichen der Industrie eingesetzt. Die Chemiebranche ist vermutlich der größte Anwender dieser Technologie, aber auch die Lebensmittel- und Abfallindustrie setzen diese Anwendungen in verstärktem Maße ein (siehe [21]). Ebenso werden sie für Motorsteuerungen, Staudammregelungen und Autopiloten benutzt.

Die Grundidee von MPC — Benutzung eines Modells, Optimieren über einen endlichen Horizont, Weiterschieben des Horizonts — entspricht dabei doch der natürlichen Vorgehensweise der meisten Menschen. Man hat ein Abbild der Wirklichkeit und versucht ein bestimmtes Ziel zu erreichen. Ausgehend von allen bekannten Einflussgrößen bestimmt man seine optimale Handlungsweise. Nach einer gewissen Zeitspanne treten in der Regel unvorhergesehene Störungen auf, so dass erneut unter Berücksichtigung aller nun bekannten Parameter optimiert werden muss. Am Beispiel eines Autofahrers kann man sich nicht nur das MPC-Vorgehen klar machen, sondern auch den Unterschied zu klassischen Reglern, wie einem PID Regler. Betrachtet man den Autofahrer als MPC Regler, so entspräche die vor ihm liegende Straße einer Referenztrajektorie, die charakteristischen Eigenschaften des Wagens dem Modell und die Pedale und die Lenkung den Steuermöglichkeiten. Da der Autofahrer immer nur einen bestimmten Teil der Straße sieht, kann er maximal bis zu diesem Punkt seine Steuerungen im Voraus berechnen. Das Fahrzeug bewegt sich aber weiter, so dass der Fahrer ein anderes Stück der Straße sieht und damit seine

alten Entscheidungen verwirft und neue optimale Parameter bestimmt. Wäre der Autofahrer ein PID Regler, dann würde er seine Entscheidungen nur auf Grund seiner Beobachtungen im Rückspiegel fällen. Dieses Beispiel ist deutlich überspitzt und trifft auch nicht ganz auf die Realität zu, da z.B. der MPC Fahrer mehr Informationen benutzt als der PID Fahrer. Jedoch verdeutlicht es die Möglichkeiten, die man mit einem MPC geregelten System hat.

Die vorliegende Arbeit beschäftigt sich nun mit MPC Schemata mit variablem Kontrollhorizont. Variabler Kontrollhorizont bedeutet, dass man keine für alle Zeiten feste Anzahl an Schritten macht bevor man den Optimierungshorizont verschiebt und eine neue optimale Steuerfolge berechnet. Speziell wollen wir untersuchen wie die Stabilität des geregelten Systems vom Kontrollhorizont und den dabei auftretenden Parametern abhängt. Zu diesem Zweck gliedert sich diese Arbeit in drei Teile. Im ersten Teil werden die Kontrolltheorie und die Modellprädiktive Regelung eingeführt. Die hierzu benötigten Begriffe wie Stabilität und Ljapunov-Funktion werden definiert und miteinander in Beziehung gesetzt. Nach der Betrachtung von Kontrollsystemen geht es weiter mit der Optimalen Steuerung. Hier werden wir aus dem Bellman'schen Optimalitätsprinzip die relaxierte dynamische Programmierung ableiten. Bevor wir das Hauptresultat zeigen können, gehen wir noch auf modellprädiktive Regelungen ein. Das Hauptresultat zeigt dann schließlich, dass die Stabilität einer modellprädiktiven Regelung durch das Lösen eines linearen Programmes bestimmt werden kann.

Im zweiten Teil geht es um lineare Programmierung. Nach zwei kleinen Beispielen werden die Grundlagen für den Simplex-Algorithmus gelegt. Dieser wird später benutzt, um das im ersten Teil hergeleitete lineare Programm zu lösen. Neben dem Grundalgorithmus des Simplex gehen wir auch kurz auf Erweiterungen und weitere Lösungsmethoden für lineare Programme ein.

Der dritte Teil befasst sich schließlich mit der Untersuchung der Stabilität von modellprädiktiven Regelungen mit variablem Kontrollhorizont. Wir werden untersuchen, wie sich der Kontrollhorizont und die Parameter der beteiligten Funktionen auf die Stabilität des Systems auswirken. Im vierten Teil wird noch eine Anregung gegeben, wie man die Resultate aus dem vorherigen Abschnitt beweisen könnte.

2 Kontrolltheorie und Modellprädiktive Regelungen

2.1 Kontrolltheorie

Die *Mathematische Kontrolltheorie* beschäftigt sich mit dem Steuern und Regeln von technischen Vorgängen. Jedoch liegt nicht nur für den Ingenieur, sondern auch für den Mathematiker ein großer Unterschied zwischen einer Steuerung und einer Regelung. Allgemein bedeutet kontrollieren, durch verändern von bestimmten Parametern einen technischen Prozess oder ein Gerät in einem Arbeitspunkt zu halten oder entlang einer vorgegebenen Trajektorie zu führen. Dies kann in einem gewissen Sinne optimal ablaufen oder nicht.

2.1.1 Beispiele

Beispiel 2.1 (Inverses Pendel)
Ein sehr einfaches und sehr gut untersuchtes Modellproblem ist das sogenannte Inverse Pendel. Hierbei handelt es sich um das Modell eines Stabes, der auf einem Finger balanciert werden soll. Jedoch betrachtet man üblicherweise nicht das dreidimensionale Modell, sondern beschränkt sich auf zwei Dimensionen und setzt das Pendel nicht auf einen Finger, sondern auf einen Wagen. Dies kann man sich dann

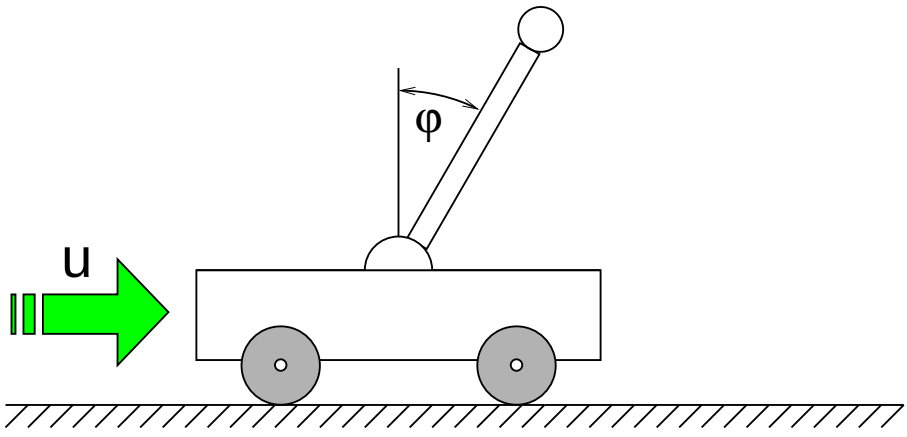

Abbildung 2.1: Ein inverses Pendel

wie in Abbildung 2.1 vorstellen. Will man einen Stab auf seinem Finger balancieren, so wirkt man dem Herunterkippen des Stabes durch „gegensteuern“ entgegen,

d.h. man beschleunigt das untere Ende des Stabes in die gleiche Richtung. Genauso versucht man dies nun beim inversen Pendel. Droht der Stab nach vorne zu kippen, so wird man den Wagen nach vorne beschleunigen, kippt er nach hinten wird man ihn nach hinten beschleunigen. Ziel des Ganzen ist nun den Stab bzw. das Pendel in eine senkrecht nach oben zeigende Position zu bringen. Mathematisch nennt man diese Position ein Gleichgewicht. Neben diesem einen Gleichgewicht gibt es noch ein viel natürlicheres: Das Pendel hängt nach unten. Dieser Gleichgewichtszustand wird immer dann eingenommen, wenn keine äußeren Kräfte, also Beschleunigung des Wagens, mehr auftreten.

Betrachtet man nun die Gleichungen, die so ein Pendel beschreiben, kommt man mit etwas Erfahrung und der Hilfe eines Physikers zu folgendem Differentialgleichungssystem:

$$\begin{aligned}
\dot{x}_1(t) &= x_2(t) && (2.1)\\
\dot{x}_2(t) &= -k\,x_2(t) + g\sin x_1(t) + u(t)\cos x_1(t) && (2.2)\\
\dot{x}_3(t) &= x_4(t) && (2.3)\\
\dot{x}_4(t) &= u(t) && (2.4)
\end{aligned}$$

In diesen Gleichungen treten nur zwei Konstanten auf. Zum einen die Erdbeschleunigung $g \approx 9.81 {}^{m}/_{s^2}$, zum anderen die Reibungszahl k. Alle anderen auftretenden Konstanten wurden so gewählt, dass sie sich herauskürzen. x_1 entspricht dem Winkel des Pendels, x_2 der Winkelgeschwindigkeit, x_3 der Position des Wagens und x_4 der Beschleunigung desselbigen. u gibt die Beschleunigung des Wagens an, also die äußere Kraft, die man aufbringen muss um das Pendel in den oberen Gleichgewichtszustand zu bringen. Die Frage ist nun, wie muss man u wählen, um dies zu erreichen. Und genau diese Frage beantwortet die Kontrolltheorie.

Beispiel 2.2 (Fliehkraftregler)
Ein Beispiel für einen mechanischen Regelkreis ist der Fliehkraftregler von James Watt und Matthew Boulton. 1788 wurde dieser erstmals zur Regelung der Dampfmenge in einer Dampfmaschine eingebaut. Allerdings wurden schon vor dieser Zeit Fliehkraftregler in Mühlen verwendet. Die Funktionsweise ist ebenso einfach wie genial. An einer Drehachse sind zwei Gewichte befestigt, die durch die bei der Drehung entstehenden Zentrifugalkräfte nach außen gedrückt werden. Über einen Scherenzug wird dadurch ein Ring nach unten geschoben. Am Ring ist über einen Hebel die Drosselklappe für die Dampfzufuhr angeschlossen. Diese reduziert dadurch die Dampfzufuhr, woraufhin sich die Drehzahl verringert und damit die Gewichte wieder nach unten sinken und die Drosselklappe wieder etwas weiter öffnen (siehe Abbildung 2.2). Auf diese Weise stellt sich ein Gleichgewichtszustand ein, in dem die Dampfmaschine sicher betrieben werden kann. Später wurden Fliehkraftregler unter anderem auch in Wählscheibentelefonen und Automatikgetrieben eingesetzt.

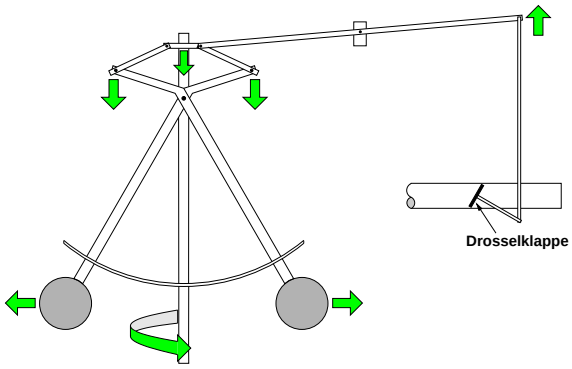

Abbildung 2.2: Schematische Darstellung des Fliehkraftreglers. Durch die Drehbewegung werden die beiden Kugeln nach außen gedrückt und der Hebel bewegt sich auf der rechten Seite nach oben. Dadurch verändert sich der Winkel der Drosselklappe und regelt so die Dampfzufuhr.

2.1.2 Grundlegende Begriffe

Definition 2.1 (Gewöhnliche Differentialgleichung)
Eine gewöhnliche Differentialgleichung *im $\mathbb{R}^n$ ist gegeben durch die Gleichung*

$$\dot{x}(t) := \frac{d}{dt}x(t) = f(t, x(t)) \tag{2.5}$$

wobei $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ eine stetige Funktion ist. f heißt Vektorfeld. *Ist zudem eine Bedingung der Form $x(t_0) = x_0$ für ein $t_0 \in \mathbb{R}$ und ein $x_0 \in \mathbb{R}^n$ gegeben, so nennt man*

$$\dot{x}(t) = f(t, x(t)) \qquad x(t_0) = x_0 \tag{2.6}$$

eine gewöhnliche Differentialgleichung mit Anfangswertbedingung. □

Die Existenz und Eindeutigkeit einer Lösung einer gewöhnlichen Differentialgleichung mit Anfangswertbedingung folgt aus dem Satz von Picard-Lindelöf (siehe hierzu [9, §10, Satz 2 und 3]).

Definition 2.2 (Gleichgewicht)
Ein Punkt $x^* \in \mathbb{R}^n$ heißt Gleichgewicht *(oder* Ruhelage, Fixpunkt, Equilibrium*) der Gleichung (2.5), falls*

$$f(x^*) = 0$$

gilt. Äquivalent dazu ist die Bedingung, dass $x(t, x^*) = x^*$ für alle $t > 0$ erfüllt ist.□

Bemerkung 2.1 Man kann o.B.d.A. annehmen, dass $x^* = 0$ ist. Für den Fall, dass $x^* \neq 0$ gilt, geht man einfach zum transformierten System $\tilde{f}(x) = f(x + x^*)$ mit $\tilde{f}(0) = f(0 + x^*) = x^*$ über.

2.1.3 Stabilität

In der Praxis und auch für theoretische Untersuchungen spielt die Stabilität eine ganz entscheidende Rolle. Unter Stabilität versteht man allgemein, dass sich der

Zustand eines Systems auf einen Arbeitspunkt zubewegt und dann auch dort oder innerhalb einer festen Umgebung um den Arbeitspunkt verbleibt.

Zur Beschreibung und Definition der verschiedenen Stabilitätsbegriffe definieren wir folgende Klassen von Funktionen.

Definition 2.3 (Vergleichsfunktionen)
Folgende Klassen von Funktionen werden definiert:

$$\begin{aligned}
\mathcal{K} &:= \{\alpha : \mathbb{R}_0^+ \to \mathbb{R}_0^+ |\ \alpha \text{ stetig, streng monoton wachsend mit } \alpha(0) = 0\} \\
\mathcal{K}_\infty &:= \{\alpha \in \mathcal{K} |\ \alpha \text{ ist unbeschränkt}\} \\
\mathcal{L} &:= \{\gamma : \mathbb{R}_0^+ \to \mathbb{R}_0^+ |\ \gamma \text{ ist stetig und monoton fallend mit } \lim_{t\to\infty} \gamma(t) = 0\} \\
\mathcal{KL} &:= \{\beta : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+ |\ \beta \text{ ist stetig, } \beta(\cdot, t^*) \in \mathcal{K}, \beta(r^*, \cdot) \in \mathcal{L} \text{ für alle } r^* > 0, t^* > 0\} \\
\mathcal{KL}_0 &:= \{\beta : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+ | \beta \text{ ist stetig, } \beta(\cdot, t^*) \in \mathcal{K}_\infty \text{ oder } \beta(\cdot, t^*) \equiv 0,\ \beta(r^*, \cdot) \in \mathcal{L} \text{ für alle } r^* > 0, t^* > 0\}
\end{aligned}$$

□

Beispiel 2.3
Für alle Konstanten $C, \sigma > 0$ ist die Funktion $\beta(r, t) = Ce^{-\sigma t}r$ aus $\mathcal{KL}$.

Bemerkung 2.2 Die Klasse $\mathcal{KL}_0$ beinhaltet auch Funktionen, die im ersten Argument zunächst aus $\mathcal{K}_\infty$ sind, und für größere t^* identisch Null sind, d.h. sie können ab einem bestimmten $\hat{t}$ auf Null springen und für alle $t > \hat{t}$ auf der Null bleiben.

Damit kann man jetzt die benötigten Stabilitätsbegriffe definieren.

Definition 2.4 (stabil, asymptotisch stabil, exponentiell stabil)
Sei $x^* \in \mathbb{R}^n$ ein Gleichgewichtspunkt der Differentialgleichung (2.5).

1. *x^* heißt* stabil, *falls eine Umgebung N von x^* und eine Funktion $\alpha \in \mathcal{K}$ existieren mit*

$$\|x(t, x)\| \leq \alpha(\|x\|) \quad \text{für alle } x \in N, t \geq 0 \tag{2.7}$$

2. *x^* heißt* lokal asymptotisch stabil, *falls eine Umgebung N von x^* und eine Funktion $\beta \in \mathcal{KL}$ existieren, so dass*

$$\|x(t, x)\| \leq \beta(\|x\|, t) \quad \text{für alle } x \in N, t \geq 0 \tag{2.8}$$

3. *x^* heißt* global (asymptotisch) stabil, *falls 1. bzw. 2. mit $N = \mathbb{R}^n$ gilt.*

4. *x^* heißt* lokal (global) exponentiell stabil, *falls Konstanten $C, \sigma > 0$ existieren, so dass 2. (bzw. 3.) mit $\beta(r, t) \leq Ce^{-\sigma t}r$ gilt.* □

Der erste dieser Stabilitätsbegriffe geht auf Alexander Michailowitsch Ljapunov (1857-1918) zurück, der ihn erstmals 1883 verwendet hat. Aus diesem Grund wird auch der Begriff *Ljapunov-stabil* anstelle von stabil verwendet.

2.1.4 Ljapunov-Funktionen

Ljapunov-Funktionen sind verallgemeinerte Abstandsfunktionen. Sie sind ein wichtiges Hilfsmittel um die Stabilität von Differentialgleichungen zu untersuchen.

Definition 2.5 (Ljapunov-Funktion)
Betrachte eine Differentialgleichung (2.5) mit $f(0) = 0$. Sei O eine offene Umgebung von 0. Dann heißt eine auf $O \setminus \{0\}$ stetig differenzierbare Funktion $V : O \to \mathbb{R}$ lokale Ljapunov-Funktion, *falls Funktionen $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ und eine stetige Funktion $W : O \to \mathbb{R}$ existieren, so dass die Ungleichungen*

$$W(x) > 0 \tag{2.9}$$
$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|) \tag{2.10}$$
$$DV(x)f(x) \leq -W(x) \tag{2.11}$$

für alle $x \in O \setminus \{0\}$ gelten.
Das Paar (V, W) nennt man Ljapunov-Paar. □

Mit Hilfe der Ljapunov-Funktionen kann man nun eine hinreichende und notwendige Bedingung für asymptotische Stabilität angeben. Genauer wird dies im folgenden Satz formuliert. Hierfür braucht man aber noch eine andere Formulierung für die Ableitungsbedingung (2.11). Außerdem zeigen wir noch eine weitere Möglichkeit auf, um ein Ljapunov-Paar zu erhalten.

Lemma 2.1
Sei $V : O \to \mathbb{R}$ eine Ljapunov-Funktion mit W aus Definition (2.5).

1. Sei D eine abgeschlossene Teilmenge von O. Dann existiert eine global Lipschitzstetige Funktion $g : \mathbb{R}_0^+ \to \mathbb{R}_0^+$, so dass $(V, g(V))$ ein Ljapunov-Paar ist.

2. Bedingung (2.11) ist genau dann erfüllt, wenn für alle Lösungen $x(t, x)$ von (2.5) und alle $t \geq 0$, für die $x(s, x) \in O$ für alle $s \in [0, t]$ gilt, die Integralungleichung
$$V(x(t, x)) \leq V(x) - \int_0^t W(x(s, x))ds \tag{2.12}$$
erfüllt ist.

Beweis: Der erste Teil des Lemmas wird in [12, Lemma 2.8] bewiesen. Der zweite Teil ist leicht durch Differenzieren der Integralungleichung nachzuweisen. ■

Bemerkung 2.3 Der Vorteil dieser beiden Formulierungen liegt in der leichteren Handhabung in Beweisen. Die Integraldarstellung der Ableitungsbedingung hat zudem den Vorteil, dass man damit auch Ljapunov-Funktionen behandeln kann, die nicht differenzierbar sind.

Wir formulieren nun eines der wichtigsten Resultate für die Stabilitätsanalyse von Differentialgleichungen.

Satz 2.1 (Ljapunov-Funktion $\Longleftrightarrow$ Asymptotische Stabilität)
Betrachte eine Differentialgleichung (2.5) mit Gleichgewicht $x^* = 0$. Es existiert genau dann eine lokale Ljapunov-Funktion V, wenn das Gleichgewicht x^* lokal asymptotisch stabil ist. Für die Vergleichsfunktion $\beta \in \mathcal{KL}$ gilt dabei

$$\beta(r,t) = \alpha_1^{-1}\big(\mu(t,\alpha_2(r))\big). \tag{2.13}$$

μ ist hierbei die Lösung des Anfangswertproblems

$$\frac{\partial \mu(t,r)}{\partial t} = -g(\mu(t,r)) \qquad \mu(0,r) = r \tag{2.14}$$

mit g aus Lemma (2.1). □

Beweis: Wir geben hier eine kurze Skizze der Beweisschritte für die Richtung Ljapunov-Funktion ⇒ asypmtotische Stabilität. Für einen ausführlichen Beweis beider Richtungen sei auf [12, Satz 2.10 und Satz 2.13] verwiesen.
⇒: Zeige zunächst, dass

$$V(x(t,x)) \leq \mu(t,V(x)) \quad \text{für alle } t \geq 0$$

gilt. Wir definieren $h_\varepsilon(t) = V(x) - \int_0^t g(h_\varepsilon(s)) + \varepsilon\, ds$ für ein $x \in V^{-1}(]0,C])$, $C > 0$ und ein $\varepsilon > 0$ mit g aus Lemma 2.1. Aus Gronwalls Lemma folgt die Konvergenz $h_\varepsilon(t) \to \mu(t,V(x))$ für $\varepsilon \to 0$ und jedes $t \geq 0$. Dann zeigen wir, dass $V(x(t,x)) \leq h_\varepsilon(t)$ für alle $t > 0$ und alle $\varepsilon > 0$ gilt. Zusammen mit der Konvergenz folgt damit die Behauptung. Nun zeigt man, dass $\mu(t,r)$ eine $\mathcal{L}$-Funktion in t und eine $\mathcal{K}$-Funktion in r ist. Damit ist β aus (2.13) eine $\mathcal{KL}$-Funktion und damit folgt auch leicht die asymptotische Stabilität. ■

2.2 Kontrollsysteme

Um nun Systeme, wie sie in den Eingangsbeispielen dargestellt wurden oder reale Prozesse mathematisch zu beschreiben, definieren wir den Begriff des Kontrollsystems.

Definition 2.6 (Nichtlineares Kontrollsystem)
Ein nichtlineares Kontrollsystem *ist gegeben durch die gewöhnliche Differentialgleichung*

$$\dot{x}(t) = f(x(t),u(t)) \tag{2.15}$$

wobei $f : \mathbb{R} \times U \to \mathbb{R}^n$ ein parameterabhängiges stetiges Vektorfeld *ist. $U \subseteq \mathbb{R}^n$ bezeichnet dabei den* Kontrollwertebereich *und $\mathcal{U} = \{u : \mathbb{R} \to U \,|\, u \text{ zulässig}\}$ den* Raum der zulässigen Kontrollfunktionen. □

Bemerkung 2.4 „Zulässig“ kann hier von stückweise stetigen bis hin zu messbaren Funktionen reichen.

Da in realen Anwendungen zumeist digitale Computer verwendet werden und analoge Schaltkreise nur noch äußerst selten zum Einsatz kommen, betrachten wir hier auch zeitdiskrete Kontrollsysteme, da sich damit solche digitalen Prozesse modellieren lassen. Fast alle Prozesse in der Natur sind aber kontinuierlich, d.h. werden durch ein zeitkontinuierliches System beschrieben. Der Übergang vom kontinuierlichen Modell zum digitalen, also zeitdiskreten, geschieht meist über Abtastung. Dies bedeutet, dass man eine feste Abtastrate $T > 0$ wählt und das zeitkontinuierliche System nur zu den Zeitpunkten $0, T, 2T, 3T, \ldots$ auswertet. Die Steuerungen werden ebenso nur zu diesen Zeitpunkten berechnet und dann über das Intervall $[kT, (k+1)T]$ konstant gehalten. Allgemein können wir ein zeitdiskretes Kontrollsystem wie folgt definieren.

Definition 2.7 (Zeitdiskretes nichtlineares Kontrollsystem)
Ein zeitdiskretes nichtlineares Kontrollsystem *ist gegeben durch*

$$x(n+1) = f(x(n), u(n)), \quad x(0) = x_0 \tag{2.16}$$

mit $x(n) \in \mathbb{X}$. $\mathbb{X}$ *ist ein beliebiger metrischer Raum und heißt Zustandsraum.* $\mathcal{U} = \{u : \mathbb{R} \to U \,|\, u \text{ zulässig}\}$ *ist der Raum der zulässigen Kontrollfunktionen. Die Lösungstrajektorie für ein* $u \in \mathcal{U}$ *bezeichnen wir mit* $x_u(n)$. □

Bemerkung 2.5 Für das zeitdiskrete Kontrollsystem kann ein beliebiger metrischer Zustandsraum $\mathbb{X}$ gewählt werden, da wir hierfür keine Differenzierbarkeitseigenschaften wie im zeitkontinuierlichen Fall brauchen.

Für viele Anwendungen reichen zum Beispiel die stetigen Funktionen nicht aus, man denke nur an die sogenannte bang-bang Steuerung, so dass man auch für unstetige Funktionen Aussagen über Existenz und Eindeutigkeit von Lösungen machen möchte. In der Tat kann man dies für die Klasse der messbaren Funktionen zeigen. Dazu zunächst die folgende Definition:

Definition 2.8 (Messbarkeit)
Sei $J = [a, b] \subset \mathbb{R}$ *ein abgeschlossenes Intervall.*

1. *Eine Funktion* $g : J \to \mathbb{R}^m$ *heißt* stückweise konstant, *falls eine Zerlegung von* J *in endlich viele Teilintervalle* $J_i, i = 1, \ldots, n$ *existiert, so dass* g *auf* J_i *konstant ist für alle* $i = 1, \ldots, n$.

2. *Eine Funktion* $g : J \to \mathbb{R}^m$ *heißt* (Lebesgue-) messbar, *falls eine Folge von stückweise konstanten Funktionen* $g_i : J \to \mathbb{R}^m, i \in \mathbb{N}$, *existiert mit* $\lim_{i\to\infty} g_i(x) = g(x)$ *für fast alle* $x \in J$, *d.h. für alle* x *aus einer Menge* $I \subseteq J$ *mit der Eigenschaft, dass* $J \setminus I$ *eine Lebesgue-Nullmenge ist.*

3. *Eine Funktion $g : \mathbb{R} \to \mathbb{R}^m$ heißt* (Lebesgue-) messbar,*falls für jedes abgeschlossene Teilintervall $J = [a, b] \subset \mathbb{R}$ die Einschränkung $g|_J$ messbar im Sinne von 2. ist.*

4. *Eine messbare Funktion $g : \mathbb{R} \to \mathbb{R}^m$ heißt* lokal essentiell beschränkt, *falls für jedes kompakte Intervall $J \subset \mathbb{R}$ eine Konstante $C \in \mathbb{R}, C > 0$ existiert, so dass $\|g(t)\| \leq C$ ist für fast alle $x \in J$.* □

Die Existenz und Eindeutigkeit der Lösungen eines Kontrollsystem im Falle von messbaren Kontrollfunktionen zeigt der folgende Satz:

Satz 2.2 (Satz von Carathéodory)
Betrachte ein Kontrollsystem mit folgenden Eigenschaften:

1. *Der Raum der Kontrollfunktionen ist gegeben durch*

$$\mathcal{U} := \{u : \mathbb{R} \to U \mid u \text{ ist messbar und lokal essentiell beschränkt}\}. \tag{2.17}$$

2. *Das Vektorfeld $f : \mathbb{R}^n \times U \to \mathbb{R}^n$ ist stetig.*

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

$$\|f(x, u)\| \leq M_R \tag{2.18}$$

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

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

$$\|f(x_1, u) - f(x_2, u)\| \leq L_R \|x_1 - x_2\| \tag{2.19}$$

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

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

$$x(t) = x_0 + \int_0^t f(x(\tau), u(\tau))\, d\tau \tag{2.20}$$

für alle $t \in J$ erfüllt. □

Definition 2.9 (Lösung)
Die eindeutige Funktion $x(t)$ in Satz 2.2 wird als Lösung, i.Z. $x(t, x, u)$, des Kontrollsystems zum Anfangswert $x_0 \in \mathbb{R}^n$ und zur Kontrollfunktion $u \in \mathcal{U}$ bezeichnet. □

Bemerkung 2.6 Diese Definition wird durch die folgende Beobachtung gerechtfertigt: $x(t, x, u)$ ist absolut stetig und somit für fast alle $t \in J$ nach t differenzierbar. Dies heißt wiederum, dass $x(t, x, u)$ die Differentialgleichung (2.5) für fast alle $t \in J$ erfüllt.

2.3 Asymptotische Kontrollierbarkeit

Es gibt nun zwei prinzipielle Möglichkeiten das $u(t)$ aus Definition 2.6 zu wählen. Die erste ist die *Steuerung* oder *open-loop Kontrolle*. Hierbei wird u als zeitvariante Funktion definiert, also $u \in L_\infty(\mathbb{R}, U)$. Die zweite Möglichkeit ist die einer *Regelung* oder *closed-loop Kontrolle*. Bei dieser Variante wird u vom aktuellen Zustand abhängig gewählt, also $u(t) = F(x(t))$. $F : \mathbb{R}^n \to U$ bezeichnet dabei das *Feedback-Gesetz*.
Eine Feedback-Lösung hat gegenüber einer Steuerung den Vorteil, dass sie auf Störungen, Meßfehler, Modellfehler usw. reagieren und diese - je nach Qualität des Feedbacks - auch korrigieren kann.

Man kann nun die Stabilitätsbegriffe für Differentialgleichungen auf Kontrollsysteme unter Beachtung der unterschiedlichen Wahl von $u(t)$ verallgemeinern. Dies führt zu den beiden folgenden Definitionen.

Definition 2.10 (asymptotisch kontrollierbar)
Das Gleichgewicht $x^* = 0$ heißt asymptotisch kontrollierbar, *wenn eine offene Umgebung N von x^* und Funktionen $\beta \in \mathcal{KL}$ und $\gamma \in C(\mathbb{R}_0^+, \mathbb{R}_0^+)$ existieren, so dass zu jedem $x \in N$ eine Kontrollfunktion $u_x \in \mathcal{U}$ existiert mit $\|u_x\|_\infty \leq \gamma(\|x\|)$ und*

$$\|x(t, x, u_x)\| \leq \beta(\|x\|, t) \tag{2.21}$$

für alle $t \geq 0$. □

Definition 2.11 (Lipschitz-stetig Feedback-stabilisierbar)
Das Gleichgewicht $x^* = 0$ heißt Lipschitz-stetig Feedback-stabilisierbar, *wenn eine offene Umgebung N von x^*, eine Funktion $\beta \in \mathcal{KL}$ und eine stetige Feedback-Abbildung $F : \mathbb{R}^n \to U$ mit $f(x, F(x))$ lokal Lipschitz in x existieren mit*

$$\|x(t, x, F)\| \leq \beta(\|x\|, t) \tag{2.22}$$

für alle $x \in N$ und $t \geq 0$. □

Es ist vergleichsweise leicht zu zeigen, dass aus der Feedback-Stabilisierbarkeit die asymptotische Kontrollierbarkeit folgt. Grob gesagt, definiert man $u_x(t)$ als das Feedback F und erhält damit die gewünschte open-loop Kontrolle. Die Umkehrung dieser Implikation gilt allerdings nicht, wie man an Brocketts nichtholonomen Integrator (Ref. [6]) sehen kann. So ein Feedback kann man nun zum Beispiel unter bestimmten Voraussetzungen über Sontag's universelle Formel (siehe [25, Seite 246]) oder über Backstepping (Ref. [12, Satz 6.5]) berechnen.

2.4 Optimale Steuerung

Bisher wurde nicht in Betracht gezogen mit welchen Kosten oder mit welcher Geschwindigkeit ein System einen stabilen Zustand erreicht. Genau das ist aber in der Praxis eine entscheidende Frage, denn braucht ein Prozess zu lange oder verschlingt zu viele Resourcen um zu einem gewünschten Ergebnis zu kommen, kann die Rentabilität in Gefahr sein. Um diese Anforderungen mathematisch erfassen zu können, benötigen wir die Begriffe Kostenfunktion und Kostenfunktional. Die Aufgabe eine unter diesen Anforderungen optimale Kontrolle zu finden ist dann das Optimale Steuerungsproblem.

Definition 2.12 (Kostenfunktion, Kostenfunktional)
Für eine stetige nichtnegative Kostenfunktion $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_0^+$ *wird das* Kostenfunktional *als*

$$J(x_0, u) = \int_0^\infty g(x(t, x_0, u), u(t))dt \tag{2.23}$$

definiert. □

Definition 2.13 (Optimales Steuerungsproblem, Optimale Wertefunktion)
Das optimale Steuerungsproblem *ist gegeben durch das Minimierungsproblem*

$$\begin{aligned} \textit{Minimiere} \quad & J(x_0, u) && (2.24)\\ s.t. \quad & \dot{x}(t) = f(x(t), u(t)) && (2.25)\\ & u(t) \in \mathcal{U} && (2.26)\\ & \forall x_0 \in \mathbb{R}^n && (2.27) \end{aligned}$$

Die Funktion

$$V_\infty(x_0) := \inf_{u \in \mathcal{U}} J(x_0, u) \tag{2.28}$$

heißt optimale Wertefunktion *des optimalen Steuerungsproblems.*
Ein Paar (x^*, u^*) *mit* $J(x^*, u^*) = V_\infty(x^*)$ *heißt* optimales Paar. □

Im Jahre 1957 stellte Richard Bellman in seinem Buch „Dynamic Programming“ (Ref. [4]) einen Ansatz zur Lösung von Optimierungsproblemen vor: die Dynamische Programmierung. Er selbst formuliert sein „Principle of Optimality“ wie folgt:

> *An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.* [4, S. 83]

Die Dynamische Programmierung wird seither als wichtiges Instrument zur Lösung vielfältigster Probleme eingesetzt. Wir formulieren eine für unsere Zwecke geeignete Version.

Satz 2.3 (Bellman'sches Optimalitätsprinzip)

1. *Für die optimale Wertefunktion $V_\infty(x_0)$ gilt für jedes $\tau > 0$*

$$V_\infty(x_0) = \inf_{u \in \mathcal{U}} \left\{ \int_0^\tau g(x(t, x_0, u), u(t))dt + V_\infty(x(\tau, x_0, u)) \right\}. \tag{2.29}$$

2. *Für ein optimales Paar (x^*, u^*) gilt für jedes $\tau > 0$*

$$V_\infty(x^*) = \int_0^\tau g(x(t, x^*, u^*), u^*(t))dt + V_\infty(x(\tau, x^*, u^*)). \tag{2.30}$$

□

Bemerkung 2.7 Eine einfache Folgerung aus dem Bellman'schen Optimalitätsprinzip ist, dass Endstücke optimaler Trajektorien wieder optimale Trajektorien sind.

Wir haben ja schon gesehen, dass man ein zeitdiskretes System durch Abtastung eines zeitkontinuierlichen Systems erhält. Ähnlich funktioniert dies nun auch beim Bellman'schen Optimaitätsprinzip. Betrachtet man das Integral über die Kostenfunktion immer für eine feste obere Grenze, so erhält man die für ein diskretes System

$$x(n+1) = f(x(n), u(n)), \qquad x(0) = x_0 \tag{2.31}$$

das Kostenfunktional

$$J_\infty(x_0, u) = \sum_{n=0}^{\infty} l(x(n), u(n)) \tag{2.32}$$

mit den laufenden Kosten $l : \mathbb{X} \times U \to \mathbb{R}_0^+$. Die optimale Wertefunktion ist dann durch die „Bellman-Gleichung“ wie folgt charakterisiert

$$V_\infty(x) = \inf_{u \in U} \left\{ V_\infty(f(x, u)) + l(x, u) \right\} \tag{2.33}$$

Kennt man die optimale Wertefunktion V_∞, dann kann man leicht ein optimales Feedback daraus berechnen. Dieses lautet dann:

$$F(x) := \operatorname*{argmin}_{u \in U} \left\{ V_\infty(f(x, u)) + l(x, u) \right\} \tag{2.34}$$

Man könnte nun versuchen, V_∞ analytisch zu berechnen oder zumindest numerisch zu approximieren, d.h. $\tilde{V} \approx V_\infty$, und daraus eine numerische Approximation

$$\tilde{F}(x) := \operatorname*{argmin}_{u \in U} \left\{ l(x, u) + \tilde{V}(f(x, u)) \right\} \tag{2.35}$$

des optimalen Feedbacks zu erzeugen. Leider ist sowohl eine analytische Darstellung der optimalen Wertefunktion als auch eine numerische Approximation nur in

äußerst einfachen Fällen, z.B. niederdimensionale oder linear-quadratische Probleme, möglich. Für höherdimensionale Zustandsräume $\mathbb{X}$ liefern selbst die besten numerischen Methoden nur sehr schlechte Approximationen. Deshalb sucht man nach Möglichkeiten selbst aus groben Approximationen $\tilde{V}$ von V_∞ noch brauchbare Feedbacks berechnen zu können. Dies liefert die „Relaxierte Dynamische Programmierung“. Hierzu formulieren wir den folgenden Satz.

Satz 2.4
Sei $\tilde{F} : \mathbb{X} \to U$ ein Feedback und $\tilde{V} : \mathbb{X} \to \mathbb{R}_0^+$ eine Funktion, die die Ungleichung

$$\tilde{V}(x) \geq \alpha l(x, \tilde{F}(x)) + \tilde{V}(f(x, \tilde{F}(x))) \tag{2.36}$$

für ein $\alpha \in]0, 1]$ und alle $x \in \mathbb{X}$ erfüllt. Dann gilt für alle $x \in \mathbb{X}$ die Abschätzung

$$\alpha V_\infty(x) \leq \alpha \sum_{n=0}^{\infty} l(x_{\tilde{F}}(n), \tilde{F}(x_{\tilde{F}}(n))) \leq \tilde{V}(x) \tag{2.37}$$

□

Beweis: Das erste Ungleichheitszeichen ist sofort aufgrund der Optimalität von V_∞ klar.
Betrachtet man (2.36) für alle $x = x_{\tilde{F}}(n), n \in \mathbb{N}_0$, so erhält man

$$\alpha l(x_{\tilde{F}}(n), \tilde{F}(x(n))) \leq \tilde{V}(x_{\tilde{F}}(n)) - \tilde{V}(x_{\tilde{F}}(n+1)) \tag{2.38}$$

Summation über n liefert

$$\alpha \sum_{n=0}^{k} l(x_{\tilde{F}}(n), \tilde{F}(x(n))) \leq \tilde{V}(x_{\tilde{F}}(0)) - \tilde{V}(x_{\tilde{F}}(k)) \leq \tilde{V}(x_{\tilde{F}}(0)). \tag{2.39}$$

Der Grenzprozess $k \to \infty$ liefert die gewünschte Ungleichung. ■

Setzt man weiterhin voraus, dass es $\gamma, \delta_1, \delta_2 \in \mathcal{K}_\infty$ gibt, so dass

$$\gamma(\|x\|_{x^*}) \leq \inf_{u \in U} l(x, u) \tag{2.40}$$

$$\delta_1(\|x\|_{x^*}) \leq \tilde{V} \leq \delta_2(\|x\|_{x^*}) \tag{2.41}$$

erfüllt sind, wobei x^* ein Gleichgewicht von f ist und $\|x\|_{x^*} = d(x, x^*)$ mit $d(\cdot, \cdot)$ eine Metrik auf $\mathbb{X}$ ist, dann kann man sogar zeigen, dass $\tilde{F}$ das System asymptotisch für alle $x_0 \in \mathbb{X}$ stabilisiert (Ref. [11, Section 2.2]).

2.5 Modellprädiktive Regelung

2.5.1 MPC Strategie

Modellprädiktive Regelung, kurz MPC, umfasst eigentlich eine große Anzahl an Kontrollmethoden, die alle drei Merkmale gemeinsam haben:

- Benutzung eines Modells zur Vorhersage der Ausgänge des Systems,
- Berechnung einer Kontrollfolge durch Minimierung einer Zielfunktion (Optimales Steuerungsproblem),
- „Receding strategy“, d.h. verschieben des Horizonts nach einer bestimmten Anzahl an Zeitschritten.

Die Grundstruktur aller MPC-Methoden ist dabei die folgende:

1. Mittels des Systemmodells werden die nächsten N Ausgänge berechnet, die sowohl von den bisherigen realen Ausgängen des Systems als auch von den N zukünftigen Steuerwerten abhängen.
2. Die nächsten N Kontrollwerte werden durch Lösen eines Optimalen Steuerungsproblems mit einer bestimmten Zielfunktion berechnet.
3. Anwenden der ersten berechneten Steuerung und verschieben des Horizonts. Alle anderen berechneten Steuerungen werden verworfen, diese werden beim nächsten Durchlauf neu berechnet.

Dieses Vorgehen wird in Abbildung 2.3 verdeutlicht. Im oberen Teil wird das Verschieben des Optimierungshorizontes dargestellt. Im unteren Teil ist ein Blockschaltbild gezeichnet, das den gesamten MPC-Ablauf nochmals grafisch darstellt.

Es ist klar, dass bei einem MPC-Schema keine kontinuierlichen Zeiten verwendet werden können. Diese Tatsache kommt der Praxis sehr nahe, wo heutzutage fast ausschließlich digitale Computersysteme zum Einsatz kommen. Das zeitdiskrete nichtlineare Systemmodell im Zustandsraum ist also gegeben durch

$$\begin{aligned} x(n+1) &= f(x(n), u(n)) \qquad x(0) = x_0 \\ y(n) &= g(x(n)) \end{aligned} \tag{2.42}$$

Hierbei bezeichnet $x(n) \in \mathbb{X}$ die Zustandsvariablen, $u(n) \in U$ die Steuervariablen und $y(n) \in \mathbb{R}^p$ die Ausgangsvariablen. n ist dabei die diskrete Abtastzeit.
Wir betrachten das System hier im Zustandsraum und nicht wie in der klassischen Regelungstechnik üblich im Bildbereich. Dies hat die Vorteile, dass die Ergebnisse

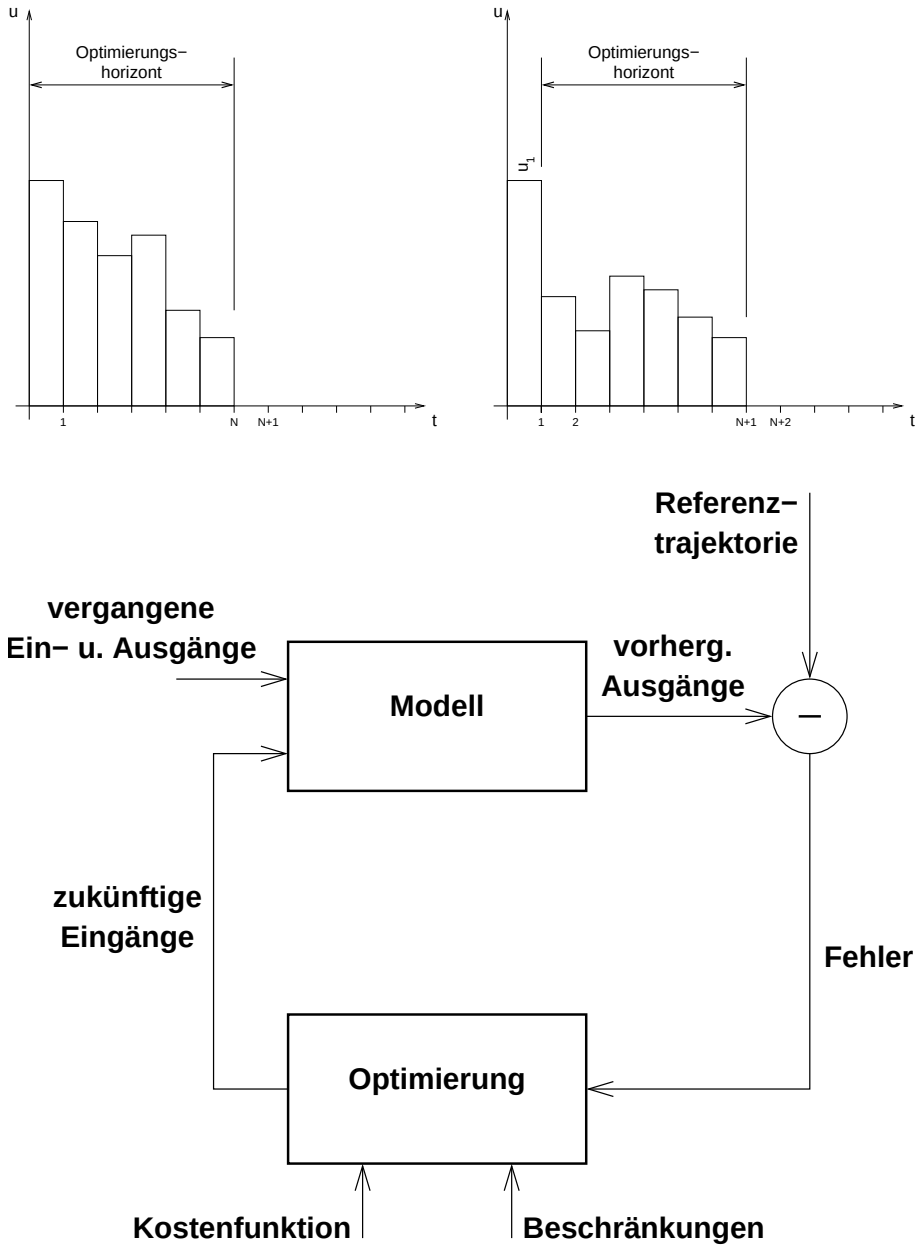

Abbildung 2.3: Das MPC-Schema. Es wird über den ganzen Optimierungshorizont eine opitmale Steuerfolge berechnet, aber nur die erste daraus angewendet. Anschließend wird der Optimierungshorizont verschoben und von Neuem das Optimale Steuerungsproblem gelöst. Im Blockschaltbild sieht man wie die verschiedenen Daten in die jeweiligen Teile des Regelungsprozesses eingehen.

leicht auf Mehr-Variablen-Systeme erweitert werden können und dass man manchmal auf die reichhaltige Theorie der linearen Kontrolltheorie und Optimalen Steuerungen zurückgreifen kann. Nun fehlt noch das Zielfunktional. Dieses ist gegeben durch:

$$J_N(x_0, u) = \sum_{n=0}^{N-1} l(x_u(n), u(n)) \tag{2.43}$$

mit der Kostenfunktion $l : \mathbb{X} \times U \to \mathbb{R}_0^+$.

Das nichtlineare MPC Problem (NMPC) ergibt sich damit insgesamt zu:

Definition 2.14 (NMPC Kontroll Problem)
Ein optimales Steuerungsproblem

$$\begin{aligned}
\min_u \quad & J_N(x(n), u(n)) && (2.44)\\
s.t. \quad & x(i+1|n) = f(x(i|n), u(i|n)) && (2.45)\\
& x(0|n) = x(n) && (2.46)\\
& u(i|n) \in U \quad i \in [0, N-1] && (2.47)\\
& x(i|n) \in \mathbb{X} \quad i \in [0, N] && (2.48)\\
& J_N(x(n), u(n)) := \sum_{n=0}^{N-1} l(x_u(n), u(n)) && (2.49)
\end{aligned}$$

heißt nichtlineares MPC (NMPC) Kontroll Problem.
Dabei bezeichnet $x(i|n)$ den Zustandsvektor zum Zeitpunkt $i \in [0, N-1]$, der zur Zeit n berechnet wurde. N heißt Optimierungshorizont. *Mit $x_u(n)$ wird die Lösungstrajektorie für ein $u \in \mathcal{U}$ bezeichnet.* □

Bemerkung 2.8 Das Kostenfunktional kann noch um einen oder mehrere Terme erweitert werden. Oft werden Endkosten oder Strafterme für Verletzungen von Beschränkungen hinzugefügt. Das brauchen wir hier alles nicht.

Bemerkung 2.9 Die Optimierungsprobleme, die dem NMPC Problem aus Definition 2.14 zu Grunde liegen, sind open loop Probleme, d.h. die Kontrolle wird nicht in Abhängigkeit vom aktuellen Zustand berechnet. Durch das Anwenden der ersten berechneten optimalen Kontrolle für das open loop System und dem Weiterschieben des Horizonts wird allerdings ein closed loop System „simuliert“. Das auf diese Weise gesteuerte System nennen wir *closed loop System mit NMPC Kontrolle*.

Betrachtet man nun einen unendlichen Optimierungshorizont, also $N = \infty$, dann folgt aus dem Bellman'schen Optimalitätsprinzip, dass die Lösungen für das open loop System gleich den Lösungen des closed loop System unter NMPC Kontrolle sind. Dies wird im folgenden Satz konkretisiert (siehe [2, Theorem 3.2]).

Satz 2.5 (NMPC mit unendlichem Horizont)
Angenommen f aus Definition 2.14 ist zweimal stetig differenzierbar und die 0 ist ein Gleichgewicht von f. Wenn das open loop Problem für $n = 0$ zulässig ist, d.h. es gibt eine optimale Kontrollfolge, dann ist das closed loop NMPC Kontroll Problem asymptotisch stabil. □

Das heißt, um asymptotische Stabilität zu erhalten, könnte man einfach den Optimierungshorizont gegen Unendlich gehen lassen. Dies kann bei linearen Regelkreisen durchaus funktionieren. Der unendliche Optimierungshorizont wird hierbei durch einen sehr großen Horizont angenähert. In der Praxis ist diese Vorgehensweise einsetzbar, da sehr effiziente Algorithmen zum Lösen von dabei auftretenden sehr großen quadratischen Optimierungsproblemen existieren. Im nichtlinearen Fall hingegen sind derart große Probleme nur sehr schwer, d.h. mit sehr langen Rechenzeiten, oder überhaupt nicht lösbar.

Aus diesem Sachverhalt folgt, dass für NMPC nur endliche (also kleine) Horizonte gewählt werden können. Dann ist es aber keineswegs mehr klar, dass das System mit diesem Optimierungshorizont stabil ist. Außerdem wird die Einschränkung aufgegeben, dass nur die erste der berechneten Steuerungen angewandt wird. Vielmehr können beliebige Kontrollhorizonte zwischen 1 und $N - 1$, wobei N der Optimierungshorizont ist, verwendet werden. Im folgenden Abschnitt wird ein Verfahren zur Berechnung eines stabilisierenden Optimierungshorizontes mit einem Kontrollhorizont, der nicht notwendigerweise gleich 1 sein muss, vorgestellt.

2.6 Berechnung eines stabilisierenden Optimierungshorizonts

Im letzten Kapitel haben wir gesehen, dass die Wahl des Optimierungshorizontes eine entscheidende Rolle dabei spielt, ob das System stabil ist oder nicht. Um im Voraus entscheiden zu können wie groß N gewählt werden muss, betrachten wir ein NMPC Kontroll System aus Definition 2.14. Jedoch erweitern wir das Konzept von MPC dahingehend, dass wir nicht mehr nur die erste der optimalen Steuerungen anwenden, sondern eine feste Anzahl an Steuerungen zwischen 1 und $N - 1$.

2.6.1 Begriffe und Definitionen

Für die folgenden Abschnitte benötigen wir einige Begriffe und Definitionen, die hier gesammelt dargestellt werden.

Um die Anzahl der angewendeten Steuerungen formal zu erfassen, definieren wir das *m-Schritt Feedback*:

Definition 2.15 (m-Schritt Feedback, m-Schritt MPC-Feedback)
Sei $m \geq 1$. Dann heißt eine Abbildung $\mu : \mathbb{X} \times \{0, \ldots, m-1\} \to U$ ein m-Schritt Feedback. *Dieses wird gemäß der Vorschrift*

$$x_\mu(n+1) = f(x_\mu(n), \mu(x_\mu([n]_m), n - [n]_m)) \quad x_\mu(0) = x_0 \tag{2.50}$$

angewendet. Hierbei bezeichnet $[n]_m$ das größte Produkt $km, k \in \mathbb{Z}$, mit $km \leq n$. Für $m \leq N$ und Anfangswert x_0 ist das m-Schritt MPC-Feedback *definiert als*

$$\mu_{N,m}(x_0, i) = u^*(i), \quad i = 0, \ldots, m-1 \tag{2.51}$$

wobei u^* eine minimierende Steuerfolge für (2.49) ist. □

Dies bedeutet, die Steuerung wird m Schritte lang angewendet, und erst dann wird neu optimiert.

Bemerkung 2.10 Für $m = 1$ erhält man den klassischen Fall.

Bemerkung 2.11 Ein m-Schritt Feedback minimiert kein Kostenfunktional, weder ein unendliches noch ein endliches. Ein m-Schritt MPC-Feedback ist hingegen sowohl vom Kontrollhorizont m als auch vom Optimierungshorizont N abhängig und wird über eine optimale Steuerfolge definiert, die ein Kostenfunktional minimiert.

Den Begriff der asymptotischen Kontrollierbarkeit wenden wir hier nicht auf die Lösungstrajektorie x_u, sondern auf die laufenden Kosten $l(x_u, n)$ gemäß folgender Definition an.

Definition 2.16 (Asymptotische Kontrollierbarkeit)
Ein Gleichgewicht x^* heißt asymptotisch kontrollierbar, *wenn es ein $\beta \in \mathcal{KL}_0$ gibt, so dass für alle $x \in \mathbb{X}$ ein $u_x \in U$ existiert, so dass für alle $n \in \mathbb{N}_0$ gilt*

$$l(x_{u_x}(n), u_x(n)) \leq \beta(l^*(x), n) \tag{2.52}$$

mit $l^*(x) := \min_{u \in U} l(x, u)$. □

Wir werden später meist zwei Arten von $\mathcal{KL}_0$-Funktionen verwenden. Zum einen

$$\beta(r, n) = C\sigma^n r \tag{2.53}$$

mit Konstanten $C \geq 1$ und $\sigma \in]0, 1[$. Mit diesem β erhalten wir exponentielle Kontrollierbarkeit. Zum anderen

$$\beta(r, n) = c_n r \tag{2.54}$$

für eine Folge $(c_n)_{n \in \mathbb{N}_0}$ mit $c_n \geq 0$ und $c_n = 0$ für alle $n \geq n_0$. Hier erhalten wir endliche Kontrollierbarkeit.

Um den Grad der Suboptimalität für ein System, das mit $\mu_{N,m}$ geregelt wird, angeben zu können, benötigen wir noch eine Art Wertefunktion für mittels m-Schritt Feedback gesteuerte Systeme. Hierzu definieren wir

$$V_\infty^\mu(x) := \sum_{n=0}^{\infty} l\big(x_\mu(n), \mu(x_\mu([n]_m), n - [n]_m)\big). \tag{2.55}$$

Ebenso definieren wir eine optimale Wertefunktion für das endliche Kostenfunktional (2.49)

$$V_N(x_0) = \inf_{u \in \mathcal{U}} J_N(x_0, u). \tag{2.56}$$

Zur Vereinfachung der Schreibweise definieren wir eine Funktion $B_N(r)$ wie folgt:

$$B_N(r) := \sum_{n=0}^{N-1} \beta(r, n) \tag{2.57}$$

2.6.2 Asymptotische Kontrollierbarkeit

Mit den Begriffen aus dem vorigen Abschnitt können wir nun erste Resultate formulieren. Als erstes wollen wir eine Abschätzung für die Güte von $V_\infty^{\tilde{\mu}}$ angeben. Hierzu erweitern wir Satz 2.4.

Lemma 2.2
Sei $\tilde{\mu} : \mathbb{X} \times \{0, \dots, m-1\} \to U$ ein m-Schritt Feedback und $x_{\tilde{\mu}}(k)$ mit $x_{\tilde{\mu}}(0) = 0$ die zugehörige Lösung. Sei $\tilde{V} : \mathbb{X} \to \mathbb{R}_0^+$ eine Abbildung, die die Ungleichung

$$\tilde{V}(x_0) \geq \tilde{V}(x_{\tilde{\mu}}(m)) + \alpha \sum_{k=0}^{m-1} l(x_{\tilde{\mu}}(k), \tilde{\mu}(x_0, k)) \tag{2.58}$$

für ein $\alpha \in]0, 1]$ und alle $x_0 \in \mathbb{X}$ erfüllt. Dann gilt für alle $x \in \mathbb{X}$ die Abschätzung

$$\alpha V_\infty(x) \leq \alpha V_\infty^{\tilde{\mu}}(x) \leq \tilde{V}(x). \tag{2.59}$$

Beweis: Der Beweis verläuft analog zum Beweis von Satz 2.4 mit $x_{\tilde{\mu}}(n)$ anstelle von $x_{\tilde{F}}(n)$. ∎

Als nächstes wollen wir eine Folgerung aus der asymptotischen Kontrollierbarkeit betrachten. Dazu formulieren wir die folgende Annahme.

Annahme 1
Für ein $\beta \in \mathcal{KL}_0$ und für alle $x \in \mathbb{X}$ ist das System asymptotisch kontrollierbar gemäß Definition (2.16).

Eine einfache Folgerung aus der asymptotischen Kontrollierbarkeit ist folgendes Lemma.

Lemma 2.3
Es gelte Annahme 1. Dann gilt für alle $N \geq 1$:

$$V_N(x_0) \leq B_N(l^*(x_0)) \tag{2.60}$$

Beweis: Aus Annahme 1 folgt sofort

$$V_N(x) \leq \sum_{n=0}^{N-1} l(x_{u_x}(n), u_x(n)) \leq B_N(l^*(x)) \tag{2.61}$$

■

Die folgenden Lemmata geben Schranken für das Kostenfunktional und die optimale Wertefunktion entlang einer optimalen Trajektorie an.

Lemma 2.4
Unter der Voraussetzung, dass Annahme 1 erfüllt ist, gilt für ein $x_0 \in \mathbb{X}$ und eine optimale Steuerung u^* für (2.49) mit Optimierungshorizont $N \geq 1$ die Ungleichung

$$J_{N-k}(x_{u^*}(k), u^*(k+\cdot)) \leq B_{N-k}(l^*(x_{u^*}(k))) \tag{2.62}$$

für alle $k = 0, \ldots, N-1$.

Lemma 2.5
Unter der Voraussetzung, dass Annahme 1 erfüllt ist, gilt für ein $x_0 \in \mathbb{X}$ und eine optimale Steuerung u^* für (2.49) mit Optimierungshorizont $N \geq 1$ die Ungleichung

$$V_N(x_{u^*}(m)) \leq J_j(x_{u^*}(m), u^*(m+\cdot)) + B_{N-j}(l^*(x_{u^*}(m+j))) \tag{2.63}$$

für alle $m = 1, \ldots, N-1$ und für alle $j = 0, \ldots, N-m-1$.

Beide Resultate sind direkte Folgerungen aus dem Bellman'schen Optimalitätsprinzip und aus der Ungleichung (2.60). Ausführliche Beweise für Lemma 2.4 und 2.5 finden sich in [14, Lemma 3.3 und Lemma 3.4].

Im Folgenden werden wir Folgen $\lambda_n > 0$ und einen Wert $\nu > 0$ betrachten und damit Bedingungen ableiten, mit denen wir für $\mu_{N,m}$ Suboptimalität gemäß Lemma 2.2 angeben können.

Lemma 2.6
Es gelte Annahme 1. Sei $N \geq 1, m \in \{1, \ldots, N-1\}$, $\lambda_n > 0, n = 0, \ldots, N-1$ eine Folge und $\nu > 0$. Für ein $x_0 \in \mathbb{X}$ existiere eine optimale Steuerfolge $u^* \in \mathcal{U}$ für (2.49)

mit Optimierungshorizont N, so dass $\lambda_n = l(x_{u^*}(n), u^*(n))$ für alle $n = 0, \ldots, N-1$ und $\nu = V_N(x_{u^*}(m))$ gilt. Dann gelten die Ungleichungen

$$\sum_{n=k}^{N-1} \lambda_n - B_{N-k}(\lambda_k) \leq 0 \qquad k = 0, \ldots, N-2 \tag{2.64}$$

$$\nu - \sum_{n=0}^{j-1} \lambda_{n+m} - B_{N-j}(\lambda_{j+m}) \leq 0 \qquad j = 0, \ldots, N-m-1 \tag{2.65}$$

Beweis: Unter den gemachten Voraussetzungen gelten die Ungleichnugen aus Lemma 2.4 und Lemma 2.5, die genau den Ungleichungen (2.64) und (2.65) entsprechen.■

Damit können wir nun Lemma 2.2 für $\mu_{N,m}$ formulieren.

Satz 2.6
Sei $\beta \in \mathcal{KL}_0, N \geq 1, m \in \{1, \ldots, N-1\}$. Angenommen alle Folgen $\lambda_n > 0, n = 0, \ldots, N-1$ und $\nu > 0$, die (2.64) und (2.65) erfüllen, erfüllen auch die Ungleichung

$$\sum_{n=0}^{N-1} \lambda_n - \nu \geq \alpha \sum_{n=0}^{m-1} \lambda_n \tag{2.66}$$

für ein $\alpha \in]0, 1]$. Unter der Voraussetzung, dass Annahme 1 gilt, sind für alle optimalen Steuerungsprobleme (2.43),(2.42) alle Vorraussetzungen für Lemma 2.2 erfüllt. Insbesondere gilt für alle $x \in \mathbb{X}$ die Ungleichung

$$\alpha V_\infty(x) \leq \alpha V_\infty^{\mu_{N,m}}(x) \leq V_N(x). \tag{2.67}$$

□

Beweis: Siehe [14, Theorem 4.2]. ■

Satz 2.6 sagt uns also, unter welchen Voraussetzungen ein optimales Steuerungsproblem die Suboptimalitätsabschätzung erfüllt. Wie kann man allerdings sicher stellen, dass all diese Voraussetzungen an λ_n und ν erfüllt werden? Nimmt man die Ungleichungen (2.64) und (2.65) und die Nichtnegativitätsbedingungen $\lambda_n > 0$ und $\nu > 0$ als Nebenbedingungen, so liefert das zusammen mit der Zielfunktion (2.66) ein nichtlineares Optimierungsproblem. Insgesamt also:

Problem 1
Sei $\beta \in \mathcal{KL}_0, N \geq 1, m \in \{1, \ldots, N-1\}$.

$$\alpha := \inf_{\lambda_0,\ldots,\lambda_{N-1},\nu} \frac{\sum_{n=0}^{N-1} \lambda_n - \nu}{\sum_{n=0}^{m-1} \lambda_n} \tag{2.68}$$

$$\sum_{n=k}^{N-1} \lambda_n - B_{N-k}(\lambda_k) \leq 0 \qquad k = 0, \ldots, N-2 \tag{2.69}$$

$$\nu - \sum_{n=0}^{j-1} \lambda_{n+m} - B_{N-j}(\lambda_{j+m}) \leq 0 \qquad j = 0, \ldots, N-m-1 \tag{2.70}$$

$$\lambda_0, \ldots, \lambda_{N-1}, \nu > 0 \tag{2.71}$$

Liegt der Zielfunktionswert α im halboffenen Intervall $]0, 1]$, so sind alle Voraussetzungen für Satz 2.6 erfüllt, und es gelten somit auch die Abschätzungen zur Suboptimalität.

Dieses Optimierungsproblem ist noch relativ schwer zu lösen. Setzt man allerdings voraus, dass $\beta(r, t)$ linear in r ist und somit auch $B_k(r)$ linear in r ist und ersetzt man die Nichtnegativitätsbedingung und die Zielfunktion, dann erhält man ein Lineares Programm (LP) mit dem gleichen Zielfunktionswert. Dieses LP lautet:

Problem 2
Sei $\beta \in \mathcal{KL}_0$ linear in $r, N \geq 1, m \in \{1, \ldots, N-1\}$.

$$\alpha := \min_{\lambda_0,\ldots,\lambda_{N-1},\nu} \sum_{n=1}^{N-1} \lambda_n - \nu \tag{2.72}$$

$$\sum_{n=k}^{N-1} \lambda_n - B_{N-k}(\lambda_k) \leq 0 \qquad k = 0, \ldots, N-2 \tag{2.73}$$

$$\nu - \sum_{n=0}^{j-1} \lambda_{n+m} - B_{N-j}(\lambda_{j+m}) \leq 0 \qquad j = 0, \ldots, N-m-1 \tag{2.74}$$

$$\sum_{n=0}^{m-1} \lambda_n = 1 \tag{2.75}$$

$$\lambda_0, \ldots, \lambda_{N-1}, \nu \geq 0 \tag{2.76}$$

Bemerkung 2.12 Für einen Beweis, dass Problem 1 und Problem 2 den gleichen Zielfunktionswert haben, siehe [14, Lemma 5.4].

Wenn wir also Problem 2 lösen können und einen Zielfunktionswert α aus dem Intervall $]0, 1]$ erhalten, dann wissen wir, dass die Abschätzungen aus Satz 2.6 gelten, jedoch noch nicht, ob das System dann auch asymptotisch stabil ist oder nicht. Dies wollen wir im Folgenden untersuchen.

2.6.3 Asymptotische Stabilität

Wir wollen nun untersuchen unter welchen Voraussetzungen das System asymptotisch stabil ist und betrachten hierzu den Kern von l^*. Zu diesem Zweck machen wir die folgende Annahme.

Annahme 2
Es existiert eine kompakte Menge $A \in \mathbb{X}$ mit folgenden Eigenschaften:

1. Für alle $x \in A$ existiert ein $u \in U$ mit $f(x,u) \in A$ und $l(x,u) = 0$. Das heißt wir bleiben für immer in A ohne Kosten zu verursachen.

2. Es existieren $\mathcal{K}_\infty$-Funktionen α_1, α_2 so, dass die Ungleichung

$$\alpha_1(\|x\|_A) \leq l^*(x) \leq \alpha_2(\|x\|_A) \tag{2.77}$$

   mit $\|x\|_A := \min_{y \in A} \|x - y\|$ für alle $x \in \mathbb{X}$ erfüllt ist.

Wir formulieren das Hauptstabilitätsresultat nun für einen zeitabhängigen Kontrollhorizont $m(n)$, d.h. der Kontrollhorizont kann zu jedem Zeitschritt unterschiedlich sein.

Satz 2.7
Sei $\beta \in \mathcal{KL}_0, N \geq 1, M \subseteq \{1, \ldots, N-1\}$. Angenommen Problem 2 hat für alle $m(n) \in M$ einen Zielfunktionswert $\alpha \in]0,1]$. Dann gilt für jedes Optimale Steuerungsproblem (2.43),(2.42), das die Annahmen 1 und 2 erfüllt, dass das $m(n)$-Schritt MPC Feedback die Menge A aus Annahme 2 asymptotisch stabilisiert, d.h. es existiert ein $\tilde{\beta} \in \mathcal{KL}_0$, so dass $x_{\mu_{N,m(n)}}$ die Abschätzung

$$\|x_{\mu_{N,m(n)}}(n)\|_A \leq \tilde{\beta}(\|x_0\|_A, n) \tag{2.78}$$

erfüllt. Außerdem ist V_N eine $m(n)$-Schritt Ljapunov-Funktion mit der Abnahmebedingung

$$V_N(x_{\mu_{N,m(n)}}(m(n))) \leq V_N(x) - \alpha V_{m(n)}(x). \tag{2.79}$$

□

Beweis: Wir zeigen zuerst, dass $V_N(x)$ eine Ljapunov-Funktion ist. Hierzu benötigen wir die Beschränkung durch zwei $\mathcal{K}_\infty$-Funktionen α_1 und $\tilde{\alpha}_2$. Aus Annahme 1 folgt:

$$\alpha_1(\|x\|_A) \leq l^*(x) = \min_{u \in U} l(x,u) \leq \inf_{u \in U} \sum_{k=0}^{N-1} l(x_u(k), u(k)) = V_N(x) \tag{2.80}$$

Aus Lemma 2.3 erhält man:

$$\begin{aligned} B_N(\alpha_2(\|x\|_A)) &= \sum_{k=0}^{N-1} \beta(\alpha_2(\|x\|_A), k) \\ &\geq \sum_{k=0}^{N-1} \beta(l^*(x), k) \\ &= B_N(l^*(x)) \\ &\geq V_N(x) \end{aligned} \tag{2.81}$$

Definiere nun $\tilde{\alpha}_2(x) := B_N(\alpha_2(\|x\|_A))$ und beachte, dass $\tilde{\alpha}_2(x) \in \mathcal{K}_\infty$ ist, so gilt

$$\alpha_1(\|x\|_A) \leq V_N(x) \leq \tilde{\alpha}_2(\|x\|_A) \tag{2.82}$$

Die Voraussetzungen für Satz 2.6 sind mit $\mu_{N,m(n)}$ erfüllt, also gilt insbesondere Ungleichung (2.59) mit $\tilde{V} \equiv V_N$:

$$V_N(x) \geq V_N(x_{\mu_{N,m(n)}}(m(n))) + \alpha \sum_{k=0}^{m(n)-1} l(x_{\mu_{N,m(n)}}(k), \mu_{N,m(n)}(k)) \tag{2.83}$$

Da $\sum_{k=0}^{m(n)-1} l(x_{\mu_{N,m(n)}}(k), \mu_{N,m(n)}(k)) \geq V_{m(n)}(x)$, folgt die Abnahmebedingung (2.79):

$$V_N(x_{\mu_{N,m(n)}}(m(n))) \leq V_N(x) - \alpha V_{m(n)}(x) \tag{2.84}$$

Um zu zeigen, dass $V_N(x)$ eine Ljapunov-Funktion ist, muss $V_{m(n)}(x)$ von Null wegbeschränkt sein. Dies folgt aus:

$$V_{m(n)}(x) = \sum_{k=0}^{m(n)-1} l(x_{\mu_{m(n),\tilde{m}}}(k), \mu_{m(n),\tilde{m}}(k)) \geq \min_{u \in U} l(x,u) = l^*(x) \geq \alpha_1(\|x\|_A) \tag{2.85}$$

Da bisher die Abnahmebedingung nur zu den Zeitpunkten $m, 2m, 3m, \ldots$ gilt, müssen auch noch die Zwischenschritte beschränkt sein.

$$\begin{aligned} V_N(x_{\mu_{N,m(n)}}(n)) = &\sum_{k=n}^{m(n)-1} l(x_{\mu_{N,m(n)}}(k), \mu_{N,m(n)}(x_{\mu_{N,m(n)}}(0), k)) \\ &+ V_{N-m(n)+n}(x_{\mu_{N,m(n)}}(m(n))) \\ \leq &\sum_{k=0}^{m(n)-1} l(x_{\mu_{N,m(n)}}(k), \mu_{N,m(n)}(x_{\mu_{N,m(n)}}(0), k)) \\ &+ V_{N-m(n)+n}(x_{\mu_{N,m(n)}}(m(n))) \\ \leq &V_N(x) + V_N(x_{\mu_{N,m(n)}}(m(n))) \\ \leq &2\, V_N(x) \end{aligned} \tag{2.86}$$

Wobei wir für die letzte Ungleichung in 2.86 die Ungleichung 2.84 benutzt haben. Damit ist nun gezeigt, dass V_N eine Ljapunov-Funktion für das System ist. Die asymptotische Stabilität des Systems folgt nun aus der Existenz einer Ljapunov-Funktion für das System (siehe Satz 2.1 oder für diskrete Systeme [13, Satz 2.15]).■

Bemerkung 2.13 Die Abschätzung für die Zwischenschritte (2.86) kann anschaulicher dargestellt werden, wenn man anstelle der Wertefunktion $V_{N-m(n)+n}(x_{\mu_{N,m(n)}}(m(n)))$ direkt die laufenden Kosten betrachtet. Allerdings erhält man dann eine schlechtere Konstante als obere Schranke. Um über die laufenden Kosten abzuschätzen, definieren wir

$$p := \left\lceil \frac{n+N}{m(n)} \right\rceil \tag{2.87}$$

Dann gilt für alle $n = 1, \ldots, m(n) - 1$:

$$\begin{aligned} V_N(x_{\mu_{N,m(n)}}(n)) &= \sum_{k=n}^{n+N-1} l(x_{\mu_{N,m(n)}}(k), \mu_{N,m(n)}(x_{\mu_{N,m(n)}}(0), k)) \\ &\leq \sum_{i=0}^{p-1} \sum_{k=i\cdot m(n)}^{(i+1)m(n)-1} l(x_{\mu_{N,m(n)}}(k), \mu_{N,m(n)}(x_{\mu_{N,m(n)}}(i \cdot m(n)), k)) \\ &\leq p \sum_{k=0}^{N-1} l(x_{\mu_{N,m(n)}}(k), \mu_{N,m(n)}(x_{\mu_{N,m(n)}}(0), k)) \\ &= p \cdot V_N(x) \end{aligned} \tag{2.88}$$

In der vorletzten Ungleichung wurde p-mal (2.84) benutzt.

Satz 2.7 sagt uns also, dass das System asymptotisch stabil mit variierendem Kontrollhorizont geregelt werden kann, sofern das lineare Programm aus Problem 2 bzw. das nichtlineare Optimierungsproblem aus Problem 1 für jeden verwendeten Kontrollhorizont einen Zielfunktionswert im Intervall $]0, 1]$ hat. Allerdings schließt Satz 2.7 nicht aus, dass das System auch mit einem kleineren Optimierungshorizont schon stabil ist. In der Tat kann man zeigen, dass es Systeme gibt, die zwar die Annahmen 1 und 2 erfüllen und einen Zielfunktionswert $\alpha < 0$ haben, jedoch nicht stabil sind. Folgender Satz zeigt diesen Sachverhalt:

Satz 2.8
Sei $\beta \in \mathcal{KL}_0$ mit der Eigenschaft $\beta(r, n+m) \leq \beta(\beta(r, n), m)$, für alle $r \geq 0, n, m \in \mathbb{N}_0$ versehen. Sei $N \geq 1, m = 1$. Angenommen Problem 1 hat einen Zielfunktionswert $\alpha < 0$. Dann gibt es ein Optimales Steuerungsproblem (2.43),(2.42), das die Annahmen 1 und 2 erfüllt, aber von $\mu_{N,1}$ nicht asymptotisch stabilisiert wird. □

Beweis: Der Beweis konstruiert explizit ein System, das nicht asymptotisch stabil ist und wird in [14, Theorem 6.3] ausführlich dargestellt. ■

2.7 Zusammenfassung

In diesem Kapitel wurden die Grundlagen für die Betrachtung und Analyse von Kontrollsystemen gelegt. Es wurden Begriffe wie Kontrollsystem, Vergleichsfunktionen und asymptotische Stabilität eingeführt. Desweiteren wurde eine notwendige und hinreichende Bedingung für asymptotische Stabilität formuliert. Im Anschluss haben wir Optimale Steuerungsprobleme und die dazugehörigen Definitionen wie Optimale Wertefunktion betrachtet. Es wurden notwendige und hinreichende Bedingungen angegeben, unter denen ein Optimales Steuerungsproblem eine Lösung besitzt.
Im Anschluss daran haben wir das Konzept von MPC, das in der Praxis großen Anklang findet, eingeführt. Wir haben gesehen, dass die Wahl eines unendlichen Optimierungshorizontes immer zu einem stabilen System führen würde, dies jedoch aufgrund der großen Nichtlinearität im auftretenden Optimierungsproblem praktisch nicht durchführbar ist. Deshalb wurden Bedingungen und Voraussetzungen hergeleitet, unter denen ein Kontrollsystem, das mit einem MPC Ansatz gesteuert wird, stabil ist. Dabei tritt ein lineares Optimierungsproblem auf, dessen Zielfunktionswert eine Aussage über die asymptotische Stabilität des Kontrollsystems liefert. Zum Schluss haben wir gesehen, dass die Länge des Optimierungshorizontes entscheidenden Einfluss auf die Stabilität hat. Diesen Sachverhalt werden wir später noch ausnutzen.

3 Lineare Programmierung

Im letzten Kapitel haben wir gesehen, dass zur Vorhersage ob ein System asymptotisch stabil ist oder nicht, ein lineares Programm gelöst werden muss. Wir wollen nun in diesem Abschnitt die Grundlagen der Linearen Programmierung darlegen und Methoden vorstellen mit denen lineare Programme gelöst werden können. Ganz speziell werden wir dabei auf den sogenannten Simplex-Algorithmus eingehen, der 1947 von George B. Dantzig (siehe [7]) erfunden wurde und heute eine der wichtigsten Methoden der Linearen Programmierung darstellt.

3.1 Beispiele

Beispiel 3.1 (The Diet Problem)
Ein bekanntes Beispiel für ein lineares Programm ist das sogenannte Diet Problem. Dabei geht es unter Einhaltung gewisser Vorgaben um die Zusammenstellung eines möglichst billigen Ernährungsplanes. Diese Vorgaben sind zum Beispiel, dass der Vitamin C-Gehalt pro Mahlzeit innerhalb gewisser Schranken liegt. Oder der Fleischverzehr sollte nicht zu groß sein, da Fleisch ein teures Produkt ist. Ein derartiges Problem könnte bei der Speisenplanung einer Armee auftreten, wo man auf die Gesamtkosten ebenso achten muss, wie auf die gesunde und ausgewogene Ernährung der Soldaten. Betrachtet man beispielsweise die Nahrungsmittel Fleisch, Äpfel, Brot und Eier, dann könnte sich Tabelle 3.1 ergeben.

Ware	Preis pro Einheit	Brennwert kcal	Eiweiß g	Kohlenhydrate g	Fett g	Vitamin C mg
Fleisch	1.50	276	18	0	21	0
Apfel	0.40	60	0.2	13.5	0.6	7.9
Brot	0.05	201	6.8	30	1.2	0
Eier	0.03	158	12.8	0.7	11.5	0

Tabelle 3.1: Tabelle mit (fiktiven) Nährwertangaben für die einzelnen Produkte

So eine Tabelle erstellt man für das komplette zur Verfügung stehende Angebot an Lebensmitteln. Nun hat man noch eine Reihe von Nebenbedingungen, wie sie in Tabelle 3.2 dargestellt sind. Als Zielfunktion hat man schließlich noch die Summe aus der Anzahl der Produkte multipliziert mit deren Preis.

Untergrenze		Produkt		Obergrenze
0	$<$	Anzahl Portionen Fleisch	$<$	2
0	$<$	Anzahl Brote	$<$	3
5	$<$	Vitamin C-Bedarf	$<$	20
		$\vdots$		

Tabelle 3.2: Nebenbedingugen für das Diet Problem

Das Diet Problem war 1947 eines der ersten realen Modelle, dass mit dem Simplex-Algorithmus gelöst wurde. Schon 1945 hat Stigler (siehe [26]) dieses Problem auf heuristische Art gelöst und eine „optimale“ Lösung angegeben, die nur 24 Cent über der wirklich optimalen Lösung lag, die mit dem Simplex gefunden wurde.

Beispiel 3.2 (Grafische Lösung eines LP)
Betrachte das folgende lineare Programm mit zwei Variablen:

$$\begin{array}{lrcrcrl}
\text{Maximiere} & 2x_1 & + & x_2 & & & (c) \\
\text{s.t.} & x_1 & + & x_2 & \leq & 5 & (1) \\
& 2x_1 & + & 3x_2 & \leq & 12 & (2) \\
& -4x_1 & + & 3x_2 & \leq & 3 & (3) \\
& x_1 & & & \leq & 4 & (4) \\
& x_1 & & & \geq & 0 & (5) \\
& & & x_2 & \geq & 0 & (6)
\end{array}$$

In Abbildung 3.1 sind die Nebenbedingungen und der zulässige Bereich aufgezeichnet. Um den optimalen Punkt zu finden, wird die Zielfunktion so lange parallel von Ecke zu Ecke verschoben, bis man den maximalen Zielfunktionswert erreicht hat. In diesem Beispiel ist das der Punkt $x^* = (4, 1)$. Dieses Vorgehen funktioniert nur für Programme mit zwei Variablen (oder 2 Nebenbedingungen), ist also in der Praxis nicht einsetzbar und dient nur der Veranschaulichung.

3.2 Grundlagen

Lineare Programme spielen in der Praxis schon immer eine große Rolle. So werden zum Beispiel die Pannenfahrzeuge beim ADAC durch Lösen eines linearen Programmes den eingehenden Hilferufen zugeordnet. Weitere Beispiele sind die Busumlaufplanung in Städten, die Zuweisung von Handyfrequenzen mit minimaler Interferenz, und die Konfiguration von optischen Netzwerken (siehe [23]).

Wir beginnen mit der Definition eines linearen Programmes.

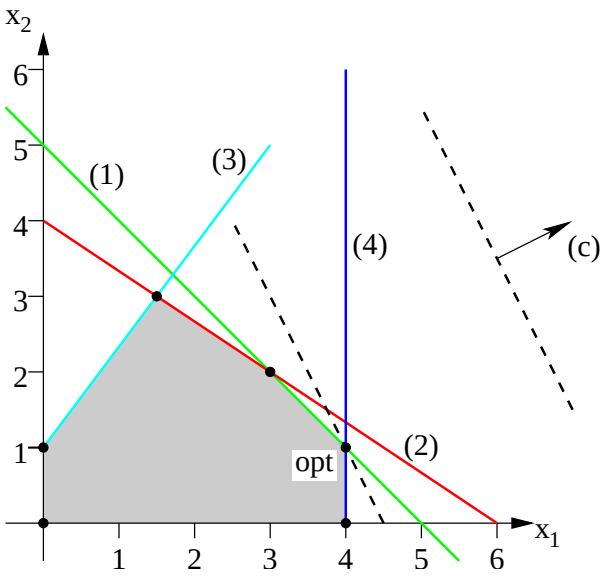

Abbildung 3.1: Grafische Lösung eines linearen Programmes. Die Zielfunktion wird solange parallel verschoben bis $2x_1 + x_2$ seinen maximalen Wert erreicht hat. Der optimale Punkt liegt also bei $x_1^* = 4, x_2^* = 1$ mit dem optimalen Zielfunktionswert 9. Der zulässige Bereich ist grau schattiert.

Definition 3.1 (Lineares Programm)
Gegeben seien $c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$. *Dann heißt*

$$\max c^T x \tag{3.1}$$
$$A\,x \leq b \tag{3.2}$$
$$x \in \mathbb{R}^n \tag{3.3}$$

ein lineares Programm *oder* lineares Optimierungsproblem, *kurz* LP. *Der Term* $c^T x$ *heißt* Zielfunktion *und* $Ax \leq b$ *sind die* Nebenbedingungen. □

Bemerkung 3.1 Sind die Variablen x nicht aus $\mathbb{R}^n$, sondern gilt $x \in \mathbb{Z}$, dann nennt man dieses Optimierungsproblem ein *lineares ganzzahliges Programm* (kurz IP oder ILP). Die Methoden und Lösungsansätze unterscheiden sich dabei in einigen Punkten deutlich von denen für lineare Programme. Wir werden hier aber nur LPs behandeln; für weiterführende Literatur zu ganzzahligen Programmen sei auf [24] verwiesen.

Bemerkung 3.2 Durch Einführen von sogenannten Schlupfvariablen kann jede Ungleichung $a_i x \leq b_i, i = 1, \ldots, m$ in eine Gleichung plus Nichtnegativitätsbedingung überführt werden:

$$a_i x \leq b_i \rightsquigarrow a_i x + s_i = b, \quad s_i \geq 0 \tag{3.4}$$

Es gilt dann für eine Lösung (x, s_i) der Gleichung, dass x auch die Ungleichung erfüllt.

Das Lösen linearer Programme basiert zu einem großen Teil auf Ergebnissen aus der Polyedertheorie. Deshalb definieren wir zuerst den Begriff des Polyeders.

Definition 3.2 (Polyeder)
Eine Teilmenge $P \subseteq \mathbb{R}^n$ heißt Polyeder*, falls es eine Matrix $A \in \mathbb{R}^{m \times n}$ und einen Vektor $b \in \mathbb{R}^m$ gibt mit*

$$P = P(A, b) := \{x \in \mathbb{R}^n \mid Ax \leq b\} \tag{3.5}$$

□

Für ein spezielles Polyeder wollen wir noch eine eigene Bezeichnung einführen. Für $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$ setzen wir

$$P^{=}(A, b) := \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\} \tag{3.6}$$

Die Nebenbedingungen eines LPs sind also Polyeder. Deshalb kann man ein LP auch in folgender Form schreiben:

$$\max c^T x \tag{3.7}$$
$$x \in P(A, b) \tag{3.8}$$

Eines der wichtigsten Resultate der linearen Programmierung ist der Dualitätssatz.

Satz 3.1 (Dualitätssatz für LP)
Es seien $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n$. Dann haben die beiden linearen Programme

$$\begin{array}{lrcll} & \max c^T x & & & \min y^T b \\ (P) & Ax \leq b & \textit{und} & (D) & y^T A \geq c^T \\ & x \geq 0 & & & y \geq 0 \end{array} \tag{3.9–3.11}$$

optimale Lösungen mit gleichem Zielfunktionswert, wenn beide zulässige Lösungen besitzen. Das Programm (P) heißt dabei **primales Programm** *und (D) das zugehörige* **duale Programm**. □

Ein Beweis für den Dualitätssatz findet sich z.B. in [22] oder jedem anderen Buch über Lineare Programmierung (z.B. in [1, 7, 8, 27]).

In allgemeiner Form sagt uns der Dualitätssatz, dass ein Paar zueinander dualer Programme (P) und (D) genau dann optimale Lösungen mit dem gleichen Zielfunktionswert besitzt, wenn beide Programme zulässige Lösungen haben. Desweiteren gilt die Aussage: Besitzt eines der beiden Programme eine optimale Lösung, so auch das andere. Diesen Zusammenhang zwischen primalem und dualem Programm kann man nun ausnutzen um geeignete Schranken zum Lösen des primalen Programmes zu erhalten.

Ein weiterer wichtiger Aspekt bei der Behandlung von linearen Programmen ist die Beschreibung durch eine innere Darstellung.

Satz 3.2 (Darstellungssatz)
Eine Teilmenge $P \subseteq \mathbb{R}^n$ ist genau dann ein Polyeder, wenn es endliche Mengen $V, E \subseteq \mathbb{R}^n$ gibt mit

$$P = conv(V) + cone(E) \tag{3.12}$$

□

Wir wissen bereits, dass man ein Polyeder auch als Schnittmenge von endlich vielen Halbräumen darstellen kann. Diese Art der Beschreibung nennt man **äußere Beschreibung**, da die zu beschreibende Menge von außen eingegrenzt wird. In Satz 3.2 wird eine **innere Beschreibung** für ein Polyeder angegeben. Das Polyeder wird von innen durch eine Kombination aus Vektoren aus der konvexen Hülle von V und Vektoren aus der konischen Hülle von E aufgespannt, also von innen her beschrieben.

Im Folgenden brauchen wir des Öfteren den Begriff der Ecke und der optimalen Ecklösung. Um diese definieren zu können, müssen wir erst definieren, was eine Seitenfläche eines Polyeders ist.

Definition 3.3 (gültige Ungleichung, Seitenfläche)
Sei $P \subseteq \mathbb{R}^n$, $F \subseteq P, c \in \mathbb{R}^n$ und $\gamma \in \mathbb{R}$.

1. *Eine Ungleichung $c^T x \leq \gamma$ heißt* gültig *bezüglich P, falls*

$$P \subseteq \{x \in \mathbb{R}^n \mid c^T x \leq \gamma\}. \tag{3.13}$$

2. *Sei P ein Polyeder. Wenn es eine gültige Ungleichung $c^T x \leq \gamma$ bezüglich P gibt mit*

$$F = P \cap \{x \mid c^T x = \gamma\}, \tag{3.14}$$

dann heißt F Seitenfläche *von P.* □

Nun können wir eine Ecke definieren.

Definition 3.4
Sei $P \subseteq \mathbb{R}^n$ ein Polyeder und $F \subseteq P$ eine Seitenfläche von P. Gibt es ein $x \in \mathbb{R}^n$ mit $F = \{x\}$, so heißt F eine Ecke *von P.* □

Mit diesen Begriffen können wir nun folgenden Satz formulieren; für einen Beweis sei auf [10, Kapitel 8] verwiesen.

Satz 3.3 (Optimale Ecklösungen)
Ein lineares Programm

$$\max c^T x \tag{3.15}$$
$$Ax = b \tag{3.16}$$
$$x \geq 0 \tag{3.17}$$

besitzt genau dann eine endliche Optimallösung, wenn es eine optimale Ecklösung besitzt. □

Um also eine optimale Lösung angeben zu können, müsste man nur alle Ecken kennen. Wir werden in Kürze sehen, dass dies ein nicht sehr praktikables Verfahren zur Lösung von LPs darstellt.

3.3 Die Simplex-Methode

3.3.1 Grundlegende Begriffe

Bevor wir das Verfahren angeben können, benötigen wir noch ein paar Begriffe.

Definition 3.5 (Standardform)
Ein lineares Programm der Form

$$\max c^T x \tag{3.18}$$
$$Ax = b \tag{3.19}$$
$$x \geq 0 \tag{3.20}$$

mit $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$ und $c \in \mathbb{R}^n$ heißt ein lineares Programm in Standardform.□

Bemerkung 3.3 Beachte, dass man ein LP aus Definition 3.1 durch Einführen von Schlupfvariablen in ein LP in Standardform überführen kann. Die Lösungen können dann durch die angegebene Beziehung zurücktransformiert werden. Die durch die beiden Programme definierten Polyeder sind aber zwei völlig verschiedene Objekte in unterschiedlichen Vektorräumen.

Definition 3.6 (Basis, Basislösung, Basisvariable, degeneriert)
Gegeben sei ein Gleichungssystem $Ax = b$ mit $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$ und rang$(A) = m$.

1. *B und N bezeichnen Spaltenindexvektoren von A mit den Eigenschaften:*
   - $B = (p_1, \ldots, _m) \in \{1, \ldots, n\}^m$
   - $N = (q_1, \ldots, q_{n-m}) \in \{1, \ldots, n\}^{n-m}$
   - $B \dot{\cup} N = \{1, \ldots, n\}$
2. *Ist A_B regulär, so heißt A_B* Basis. *A_N heißt dann* Nichtbasis.
3. *Ein Vektor $x \in \mathbb{R}^n$ mit $x_N = 0$ und $x_B = A_B^{-1} b$ heißt* Basislösung zur Basis A_B.
4. *Ist A_B eine Basis, dann heißen die Variablen $x_j, j \in B$* Basisvariablen *und die Variablen $x_j, j \in N$* Nichtbasisvariablen.
5. *Ist A_B eine Basis, dann heißen A_B und die zugehörige Basislösung x* zulässig, *wenn $A_B^{-1} b \geq 0$ gilt.*

6. *Eine Basislösung x zur Basis A_B heißt* nichtdegeneriert, *falls $x_B = A_B^{-1}b > 0$ gilt. Andernfalls heißt sie* degeneriert. □

Zur Herstellung einer Verbindung zwischen den Ecken eines Polyeders und dem Begriff einer Basis geben wir folgenden Satz an.

Satz 3.4
Sei $P = P^=(A, b) \subseteq \mathbb{R}^n$ ein Polyeder mit rang$(A) = m$ und $x \in P$.
x ist genau dann eine zulässige Basislösung, wenn x eine Ecke von $P^=(A, b)$ ist. □

Beweis: Siehe [10, Satz 9.5]. ■

Nun können wir auch nachvollziehen, warum das Enumerieren der Ecken kein praktikables Verfahren zur Lösung von LPs darstellt. Betrachtet man eine $m \times n$-Matrix, so kann diese Matrix bis zu $\binom{n}{m}$ zulässige Basen, und damit Ecken haben. Für reale Probleme liegt diese Zahl schnell über der Anzahl der Atome im Universum.

Um ein einfaches Optimalitätskriterium zu erhalten, schreiben wir den Zielfunktionswert in Abhängigkeit der Basis und Nichtbasis.

Satz 3.5 (Simplexkriterium)
Gegeben sei ein LP in Standardform mit Basis A_B.

1. *Für den Zielfunktionswert $c^T x$ für $x \in P^=(A, b)$ gilt:*

$$c^T x = c_B^T A_B^{-1} b + (c_N^T - c_B^T A_B^{-1} A_N) x_N \tag{3.21}$$

2. *Sei A_B eine zulässige Basis. Gilt*

$$c_N^T - c_B^T A_B^{-1} A_N \leq 0, \tag{3.22}$$

dann ist die zugehörige Basislösung x mit $x_B = A_B^{-1} b, x_N = 0$ optimal für das LP. □

Beweis: 1. Nachrechnen.

2. Sei $y \in P^=(A, b)$ beliebig. Dann gilt:

$$c^T y = c_B^T A_B^{-1} b + \underbrace{(c_N^T - c_B^T A_B^{-1} A_N)}_{\leq 0} \underbrace{y_N}_{\geq 0} \leq c_B^T A_B^{-1} b = c_B^T x_B = c^T x \tag{3.23}$$

■

Definition 3.7
Der Term $\bar{c}^T := c_N^T - c_B^T A_B^{-1} A_N$ heißt reduzierte Kosten. □

Sind also die reduzierten Kosten negativ, so ist die aktuelle Basis optimal. Dies liefert ein ideales Kriterium zum Testen auf Optimalität und damit zum Beenden des Algorithmus.

3.3.2 Der Simplex-Algorithmus

Angenommen wir haben ein LP in Standardform mit den Matrizen $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$ und $c \in \mathbb{R}^n$. Außerdem haben wir eine zulässige Basis B für dieses LP gegeben. Falls eine optimale Lösung des LP existiert, können wir diese mit dem folgenden Algorithmus berechnen.

ALGORITHMUS 1: SIMPLEX-ALGORITHMUS

Schritt 1: **Berechne**

$$\begin{array}{lcl} A_B^{-1} & & \\ \bar{A} & = & A_B^{-1} A_N \\ \bar{b} & = & A_B^{-1} b \\ \bar{c}^T & = & c_N^T - c_B^T A_B^{-1} A_N \end{array} \tag{3.24}$$

Schritt 2: **Optimalitätsprüfung**
Falls $\bar{c} \leq 0$, Basis optimal, RETURN(optimal)
sonst gehe zum nächsten Schritt

Schritt 3: **Bestimmung der Pivotspalte**
Wähle einen Index $s \in \{1, \ldots, n-m\}$ für den gilt $\bar{c}_s > 0$.

Schritt 4: **Prüfung auf Beschränkheit**
Falls für alle $i \in \{1, \ldots, m\}$ gilt: $\bar{a}_{is} \leq 0$, dann ist das LP unbeschränkt, RETURN(unbeschränkt)
sonst gehe zum nächsten Schritt

Schritt 5: **Bestimmung der Pivotzeile**
Wähle einen Index $r \in \{1, \ldots, m\}$ für den gilt

$$\frac{\bar{b}_r}{\bar{a}_{rs}} = \min \left\{ \frac{\bar{b}_i}{\bar{a}_{is}} \mid \bar{a}_{is} > 0, i = 1, \ldots, m \right\} \tag{3.25}$$

Schritt 6: **Basisaustausch**
Setze

$$\begin{array}{lcl} B & := & (p_1, \ldots, p_{r-1}, \mathbf{q_s}, p_{r+1}, \ldots, p_m) \\ N & := & (q_1, \ldots, q_{s-1}, \mathbf{p_r}, q_{s+1}, \ldots, q_{n-m}) \end{array} \tag{3.26}$$

Schritt 7: **gehe zu Schritt 1**

Bemerkung 3.4 Dies ist die Grundversion des Simplex-Algorithmus. Für reale Implementierungen wird der sogenannte *revidierte Simplex* (siehe [8, 27]) benutzt.

3.3.3 Die einzelnen Schritte im Simplex

Schritt 1 und 2

In Schritt 1 werden nur die Matrizen und Vektoren mit der aktuellen Basis berechnet. Die Prüfung auf Optimalität wurde schon in Satz 3.5 gezeigt.

Schritt 3

In diesem Schritt wird der erste der beiden Pivot-Indizes ausgewählt, die bestimmen welche Variable in die neue Basis kommt und welche aus der Basis in die Nichtbasis wechselt. Diejenige mit der Pivotspalte s indizierte Variable wechselt von der Nichtbasis in die Basis. Hier stellt sich die Frage, ob die neue Basis überhaupt noch eine Basis ist. Dies beantwortet der folgende Satz.

Satz 3.6 (Basisaustausch)
Sei $Ax = b$ ein Gleichungssystem mit $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m, x \in \mathbb{R}^n$ und sei A_B eine Basis mit $B = (p_1, \ldots, p_m)$ und $N = (q_1, \ldots, q_{n-m})$. Sei $\bar{A} := A_B^{-1}A_N$. Gilt für $1 \leq r \leq m, 1 \leq s \leq n-m:\ \bar{a}_{rs} \neq 0$, dann ist $A_{B'}$ mit $B' = (p_1, \ldots, p_{r-1}, q_s, p_{r+1}, \ldots, p_m)$ eine Basis von A. □

Beweis: Siehe [10, Satz 9.10]. ■

Schritt 4

In diesem Schritt wird das Polyeder auf Unbeschränktheit in Richtung der Zielfunktion getestet. Dazu muss nur die Pivotspalte der Matrix $\bar{A}$ überprüft werden. Sind alle Einträge dieser Spalte kleiner oder gleich Null, so ist das Problem unbeschränkt und es gibt kein endliches Optimum. Dieses einfache Kriterium reicht schon aus, da man sich folgendes überlegen kann: Für einen Vektor $y \in P^{=}(A, b)$ ist $y_B = A_B^{-1}b - \bar{A}y_N \geq 0$ und $y_N \geq 0$ eine äquivalente Darstellung. Für beliebiges $\lambda \geq 0$ gilt: $y^\lambda \in P^{=}(A, b)$ mit $y_N^\lambda := \lambda e_s$, e_s s-ter Einheitsvektor, und $y_B^\lambda := A_B^{-1}b - \bar{A}y_N^\lambda = A_B^{-1}b - \lambda\bar{A}_{\cdot s}$, da $y_B^\lambda = A_B^{-1}b - \lambda\bar{A}_{\cdot s} \geq A_B^{-1}b \geq 0$ und $y_N^\lambda \geq 0$. Allerdings kann $c^T y^\lambda = c_B^T A_B^{-1}b + \bar{c}^T y_N^\lambda = c_B^T A_B^{-1}b + \lambda\bar{c}_s$ mit λ unbeschränkt wachsen, da $\bar{c}_s > 0$ ist. Also ist $c^T x$ auf $P^{=}(A, b)$ unbeschränkt. Ist aber ein $\bar{a}_{is} > 0$, so ist dies nicht möglich, da λ nicht beliebig groß werden kann ohne eine Nichtnegativitätsbedingung zu verletzen.

Schritt 5

Die letzte Überlegung nutzen wir in diesem Schritt aus um den zweiten Pivotindex zu bestimmen. Angenommen ein $\bar{a}_{is} > 0$, so muss $y_B^\lambda = \bar{b}_i - \lambda\bar{a}_{is} \geq 0$ gelten,

damit kann λ höchstens so groß sein wie $\frac{\bar{b}_i}{\bar{a}_{is}}$. Wählen wir nun wie im Algorithmus beschrieben einen Index r aus, so erhöhen wir im nichtdegenerierten Fall garantiert unseren Zielfunktionswert (siehe [10]).

Schritt 6 und 7

Nun nehmen wir den Basisaustausch vor, erhalten nach Satz 3.6 wieder eine Basis und gehen zurück zu Schritt 1.

3.3.4 Auswahlregeln

In den Schritten 3 und 5 wird jeweils ein Index ausgewählt. Bisher haben wir noch nicht näher angegeben, wie diese Auswahl zu treffen ist. Hierfür gibt es mehrere Möglichkeiten. Für die Auswahl der Pivotspalte gibt es zum Beispiel folgende Regeln:

1. Kleinster-Index-Regel
2. Kleinster-Variablenindex-Regel
3. Steilster-Anstieg-Regel
4. Größter-Fortschritt-Regel
5. und noch viele mehr

Ähnliches gilt für die Zeilenauswahlregeln. Hier seien nur einige genannt.

1. 1. Lexikographische-Zeilenauswahl-Regel
2. 2. Lexikographische-Zeilenauswahl-Regel
3. Kleinster-Index-Regel
4. Kleinster-Variablenindex-Regel

Für nähere Erläuterungen zu den einzelnen Regeln sei auf [8, 24] verwiesen.

Verwendet man die 1. Lexikographische-Zeilenauswahl-Regel und eine beliebige Spaltenauswahlregel, so kann man zeigen, dass der Simplex-Algorithmus nach endlich vielen Schritten endet. Bland hat 1977 gezeigt, dass dies auch gilt, wenn man für beide Auswahlen die Kleinster-Variablenindex-Regel verwendet (Ref. [5])

3.3.5 Varianten und Erweiterungen des Simplex

Wie schon erwähnt wird in praktischen Implementierungen nicht die hier vorgestellte Version des Simplex-Algorithmus verwendet. Es wird vielmehr auf den *revidierten Simplex* zurückgegriffen. Hierbei werden nicht immer alle Spalten der Matrix $\bar{A}$ berechnet, sondern nur solche die im aktuellen Schritt benötigt werden. Dies hat den offensichtlichen Vorteil, dass sowohl der Rechenaufwand als auch der Speicherbedarf wesentlich geringer sind. Der revidierte Simplex wurde 1953 von G. B. Dantzig und W. Orchard-Hays eingeführt (siehe auch [19]).

Neben der hier vorgestellten Grundversion, die mit dem primalen Problem arbeitet, gibt es noch den *dualen Simplex*. Dieser arbeitet mit dem dualen Programm und wird oft in der Postoptimierung eingesetzt, d.h. das schon gelöste Problem wird verändert und soll nun mit möglichst geringem Aufwand erneut gelöst werden. Fügt man zum Beispiel eine neue Nebenbedinugung hinzu, so ist in der Regel die alte optimale Lösung nicht mehr zulässig, allerdings kann man sofort eine zulässige duale Basis angeben und damit den dualen Simplex starten. Der Dualitätssatz liefert dann auch den Zielfunktionswert für das primale Problem. Ähnlich verhält es sich, wenn man die rechte Seite b des Ausgangsproblems ändert.

Ein Problem, das wir als solches bisher nicht gekennzeichnet haben, ist die Voraussetzung, dass wir eine gültige Basis für das LP kennen. Bei kleinen Problemen ($n, m < 10$) kann man eine solche meist direkt ablesen. Für reale in der Praxis auftretende Probleme mit mehreren tausend Nebenbedingungen ist dies nicht mehr möglich. Um dieses Problem zu lösen, wendet man die *Zwei-Phasen-Methode* an. Wie der Name schon sagt, besteht diese aus zwei Phasen: Phase 2 ist der in 3.3.2 beschriebene Algorithmus. Phase 1 testet das Problem auf Zulässigkeit und liefert durch Lösen eines weiteren LPs, bei dem eine Startbasis bekannt ist, eine Startbasis für das eigentliche Problem.

3.4 Laufzeit und weitere Lösungsalgorithmen

Eine Frage, die sich beim Simplex-Algorithmus noch stellt, ist: Wie schnell löst der Simplex in Abhängigkeit von der Problemgröße ein LP? Man möchte also wissen, ob der Simplex polynomiale Laufzeit hat. Eine Klasse von Problemen ist polynomial lösbar bedeutet hier grob gesagt, es gibt ein Polynom p, so dass ein Algorithmus ein Problem der Größe n aus dieser Klasse in höchstens $p(n)$ Operationen löst. Lange Zeit wusste man nicht, ob lineare Programme polynomial lösbar sind. 1972 gaben V. Klee und G. Minty (Ref. [18]) Beispiele an, für die der Simplex $2^n - 1$ Iterationen benötigt. Damit war gezeigt, dass für den Fall, dass lineare Programme polynomial lösbar sind, der Simplex nicht der geeignete Algorithmus war, um dies zu zeigen. Allerdings hat Khachiyan 1979 (Ref. [17]) die *Ellipsoid-Methode* angegeben und damit

gezeigt, dass ein LP in polynomialer Zeit lösbar ist. Die Ellipsoid-Methode basiert auf vollkommen anderen geometrischen Überlegungen wie der Simplex, ist jedoch in der Praxis meist deutlich langsamer als der Simplex. Eine Klasse von Algorithmen, die nicht nur beweisbar polynomial sind, sondern auch in der Praxis polynomial laufen, sind die *Innere-Punkte-Methoden*. Der erste Vertreter dieser Klasse wurde 1984 von Karmarkar (Ref. [15]) vorgestellt, aber erst später den Innere-Punkte-Methoden zugeordnet. Heute gehören die Innere-Punkte-Methoden ebenso zum Standardwerkzeug der linearen Optimierung wie der Simplex-Algorithmus.

3.5 Zusammenfassung

Wir haben in diesem Kapitel einen Überblick über Lineare Programmierung gegeben. Angefangen bei grundlegenden Begriffen und einfachen Lösungsmethoden, über den Begriff des Polyeders bis hin zum Dualitätssatz, haben wir uns hauptsächlich mit den Simplex-Algorithmus beschäftigt. Wir haben hierzu gesehen, dass eine optimale Lösung eines linearen Programmes immer in einer Ecke liegt. Doch eine Enumeration aller Ecken ist kein probates Mittel zur Lösung eines LPs. Der Simplex läuft trotzdem von Ecke zu Ecke, jedoch stets einhergehend mit einer Verbesserung des Zielfunktionswertes (Schritt 5 im Algorithmus). Wir haben dann noch gesehen, dass es für eine praktische Implementierung der Simplex-Methode noch sehr viele Feinheiten zu beachten gibt, z.B. bei der Wahl der Auswahlregeln für die Pivot-Indizes. Darüberhinaus wird neben der hier vorgestellten Grundversion der revidierte Simplex und der duale Simplex verwendet. Zum Schluss haben wir kurz die Laufzeit des Simplex erläutert und weitere Algorithmen (Innere-Punkte-Methode, Ellipsoid-Methode) kennen gelernt.

4 MPC mit variablem Kontrollhorizont

In Kapitel 2 haben wir gesehen, dass wir durch das Lösen eines linearen Programmes bestimmen können, ob ein System stabil geregelt wird oder nicht. Wie man ein lineares Programm löst haben wir in Kapitel 3 gesehen. Nun wollen wir untersuchen, unter welchen Voraussetzungen Systeme mit variablem Kontrollhorizont stabil sind.

4.1 Berechnung von α

Wir wollen nun also das folgende lineare Programm lösen:

Problem 3
Sei $\beta \in \mathcal{KL}_0$ linear in $r, N \geq 1, m \in \{1, \ldots, N-1\}$.

$$\alpha := \min_{\lambda_0,\ldots,\lambda_{N-1},\nu} \sum_{n=1}^{N-1} \lambda_n - \nu \tag{4.1}$$

$$\sum_{n=k}^{N-1} \lambda_n - B_{N-k}(\lambda_k) \leq 0 \qquad k = 0, \ldots, N-2 \tag{4.2}$$

$$\nu - \sum_{n=0}^{j-1} \lambda_{n+m} - B_{N-j}(\lambda_{j+m}) \leq 0 \qquad j = 0, \ldots, N-m-1 \tag{4.3}$$

$$\sum_{n=0}^{m-1} \lambda_n = 1 \tag{4.4}$$

$$\lambda_0, \ldots, \lambda_{N-1}, \nu \geq 0 \tag{4.5}$$

L. Grüne stellt hierfür in [14] einen MATLAB-Code zur Verfügung. Als β wird hierbei (2.53), also $\beta(r, n) = C\sigma^n r$ verwendet. Ein typischer Aufruf sieht zum Beispiel so aus:

```
mpcalpha_Csigma (1.5,0.7,5,3,1,2);
```

Hierbei haben die Parameter die folgende Bedeutung:

1.5	C	C aus der $\mathcal{KL}_0$-Funktion β
0.7	σ	σ aus der $\mathcal{KL}_0$-Funktion β
5	N	Optimierungshorizont
3	m	Kontrollhorizont
1		Endgewicht (optional, 1 $\widehat{=}$ kein Endgewicht, hier nicht verwendet)
2		Ausgabeart (optional, 2 $\widehat{=}$ volle Ausgabe mit grafischer Aufbereitung)

Als Ausgabe erhält man damit:

```
-3.15965 1.0     1.0     1.0     1.0     0.0
 0.0    -2.79950 1.0     1.0     1.0     0.0
 0.0     0.0    -2.28500 1.0     1.0     0.0
 0.0     0.0     0.0    -1.55000 1.0     0.0
 0.0     0.0     0.0    -4.15965 0.0     1.0
 0.0     0.0     0.0    -1.0    -3.79950 1.0

Optimization terminated.
V(5,x(0)) = 1.637472
V(5,x(3)) = 1.447731
alpha     = 0.189741
OptimBound= 5.270352
STABLE

ans =

    0.1897
```

Man erhält also als Zielfunktionswert $\alpha = 0.1897$ und damit sagt uns Satz 2.7, dass das System asymptotisch stabil geregelt wird, wenn man $\beta(r, n) = 1.5 \cdot 0.7^n r$ zu Grunde legt. Verändert man die Parameter C und σ ein wenig, also zum Beispiel

```
mpcalpha_Csigma(1.5,0.85,5,3,1,2);
```

Dann erhält man allerdings

```
Optimization terminated.
V(5,x(0)) = 1.925088
V(5,x(3)) = 2.632868
alpha     = -0.707781
OptimBound= -1.412867
```

```
UNSTABLE

ans =

   -0.7078
```

Hier ist $\alpha = -0.7078 < 0$, also ist das System nicht stabil. Das heißt, die Stabilität hängt stark von den Parametern C und σ, also von der beschränkenden $\mathcal{KL}_0$-Funktion β ab. Das eigentlich interessante ist aber, wie hängt die Stabilität vom Kontrollhorizont m ab. Dazu lösen wir die fünf folgenden LPs:

```
mpcalpha_Csigma(2.5,0.5,6,1,1,2);
mpcalpha_Csigma(2.5,0.5,6,2,1,2);
mpcalpha_Csigma(2.5,0.5,6,3,1,2);
mpcalpha_Csigma(2.5,0.5,6,4,1,2);
mpcalpha_Csigma(2.5,0.5,6,5,1,2);
```

Und erhalten die Zielfunktionswerte wie sie in Tabelle 4.1 aufgelistet sind. Stellt man α in Abhängigkeit von m grafisch dar ergibt sich das in Abbildung 4.1 dargestellte Bild.

m	1	2	3	4	5
α	-0.5358	-0.0708	0.0204	-0.0708	-0.5358

Tabelle 4.1: α für verschiedene Kontrollhorizonte bei gleichem β.

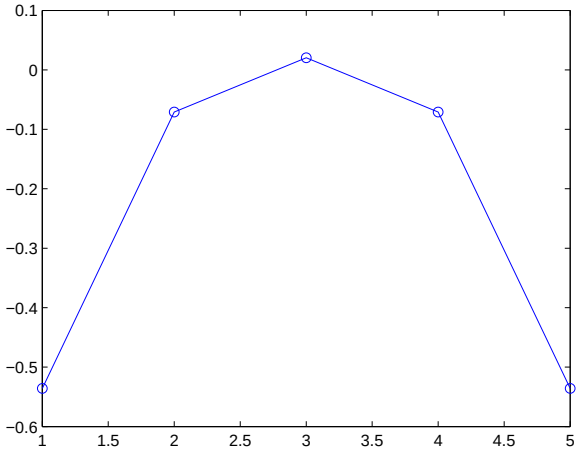

Abbildung 4.1: α in Abhängigkeit vom Kontrollhorizont m.

Es stellen sich nun mehrere Fragen:

1. Welchen Einfluss haben die Parameter C und σ auf den Zielfunktionswert?
2. Ist das in Abbildung 4.1 gezeigte Verhalten typisch? Also bei kleinen und großen Kontrollhorizonten ist das System instabil, bei den mittleren stabil.
3. Sind die Zielfunktionswerte für „symmetrisch“ liegende Kontrollhorizonte immer identisch?
4. Wie groß muss der Optimierungshorizont sein, damit die Zielfunktionswerte unabhängig vom Kontrollhorizont größer als Null sind?

Zur Beantwortung dieser Fragen sollen die nachfolgenden Kapitel einen Teil beitragen.

4.2 Stabilitätsbereiche

In diesem Abschnitt wollen wir nun untersuchen, für welche Werte von C und σ der Zielfunktionswert größer als Null ist. Zu diesem Zweck definieren wir erst einmal den Begriff des Stabilitätsbereiches.

Definition 4.1
Sei $\beta(r,t) \in \mathcal{KL}_0$ linear in r und habe n wählbare Parameter $\xi_i \in \mathbb{R}, i = 1,\dots,n$, $N \geq 1, m \in \{1,\dots,N-1\}$. Die Menge

$$S(N,m) := \left\{ (\xi_1,\dots,\xi_n) \in \mathbb{R}^n \mid \alpha(N,m,\xi_1,\dots,\xi_n) > 0 \right\} \tag{4.6}$$

wobei α der optimale Wert von Problem 3 ist, heißt Stabilitätsbereich. □

Als einfachsten Fall betrachten wir ein System mit $N = 2$ $\beta(r,t) = C\sigma^t r$. Das dazugehörige lineare Programm hat dann folgende Gestalt:

$$\begin{aligned} \min \quad & \lambda_0 + \lambda_1 + \nu & & \\ s.t. \quad & \nu & \leq & \frac{C(\sigma^2-1)}{\sigma-1}\lambda_1 \\ & \lambda_0 + \lambda_1 & \leq & \frac{C(\sigma^2-1)}{\sigma-1}\lambda_0 \\ & \lambda_0 & = & 1 \\ & \lambda_0, \lambda_1, \nu & \geq & 0 \end{aligned}$$

Vereinfacht man die Nebenbedingungen, so ergibt sich:

$$\begin{aligned} \min \quad & 1 + \lambda_1 + \nu & & \\ s.t. \quad & -C(\sigma+1)\lambda_1 + \nu & \leq & 0 \\ & \lambda_1 & \leq & C(\sigma+1) - 1 \\ & \lambda_1, \nu & \geq & 0 \end{aligned}$$

Wir haben hier also ein LP mit zwei Variablen und können somit die Lösung grafisch bestimmen. Aus Abbildung 4.2 sieht man, dass das Optimum auf dem Schnitt der beiden Geraden liegt und folglich die Koordinaten

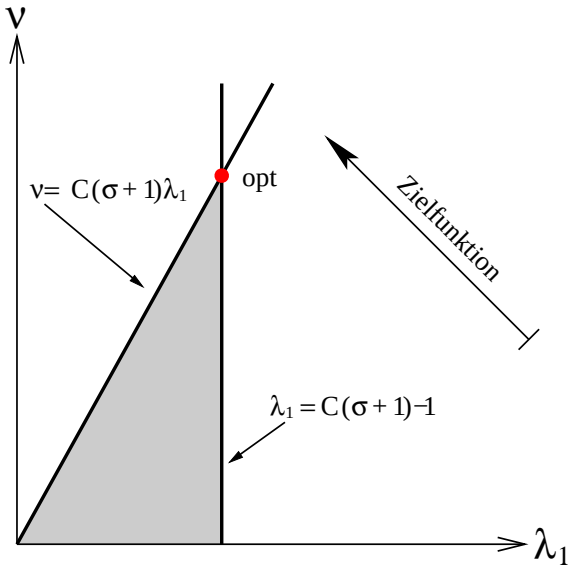

Abbildung 4.2: Grafische Lösung des LPs. Beachte, dass $C(\sigma+1)$ immer größer als 1 ist und damit das Optimum stets am eingezeichneten Punkt angenommen wird

$$(\lambda_0^*, \lambda_1^*, \nu^*) = \big(1,\ C(\sigma+1)-1,\ C(\sigma+1)(C(\sigma+1)-1)\big) \tag{4.7}$$

hat. Damit lässt sich der Zielfunktionswert α berechnen und somit können auch C und σ derart bestimmt werden, dass $\alpha > 0$ gilt.

$$\begin{aligned} & \alpha = \lambda_0^* + \lambda_1^* - \nu^* &> 0 \\ \iff\ & 1 + C(\sigma+1) - 1 - C(\sigma+1)(C(\sigma+1)-1) &> 0 \\ \iff\ & 2 &> C(\sigma+1) \end{aligned} \tag{4.8}$$

D.h. falls $C(\sigma+1) < 2$ gilt, ist das System stabil. In Abbildung 4.3 ist dieser Bereich gezeichnet. Wählt man nun eine $\mathcal{KL}_0$-Funktion mit C und σ aus diesem Bereich, so ist das System asymptotisch stabil. Außerdem ist in Abbildung 4.3 noch der Stabilitätsbereich für $N = 3, m = 1$ eingezeichnet, der mit ähnlichen Überlegungen gewonnen wurde. Eine ähnliche Rechnung kann man nun auch mit einer einfacheren $\mathcal{KL}_0$-Funktion β machen. Wir setzen

$$\beta(r,n) = \begin{cases} \gamma r & n = 0 \\ 0 & n \geq 1 \end{cases}. \tag{4.9}$$

Betrachtet man für dieses β wieder den Fall $N = 2, m = 1$, so erhält man als Stabilitätsbereich das Intervall $\gamma \in [1, 2[$. Für größere N ist dieses einfache geometrische Verfahren nicht mehr anwendbar.

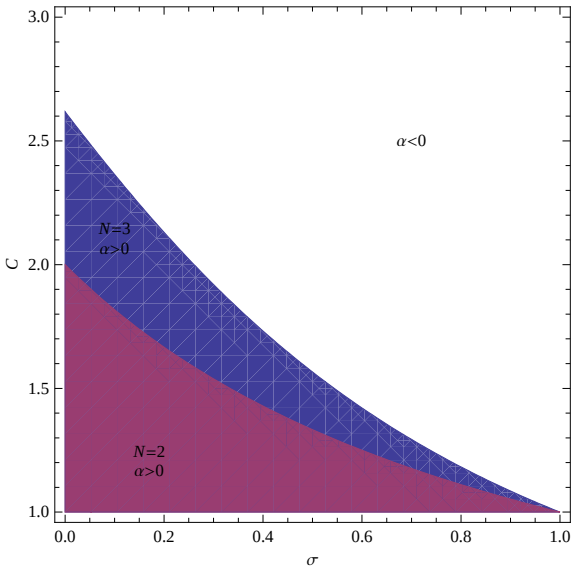

Abbildung 4.3: Stabilitätsbereiche für $N = 2$ und $N = 3$ jeweils für $m = 1$. Sind C und σ aus diesen Bereichen, so sind die Zielfunktionswerte der entsprechenden LPs größer als Null

4.3 Ein symbolischer Simplex

Die allgemeine Berechnung der Lösung und des Zielfunktionswertes stieß im vorherigen Abschnitt an die Grenzen der geometrischen Anschaulichkeit. Nun haben wir aber mit dem Simplex-Algorithmus (siehe Seite 40) ein Verfahren kennen gelernt mit dem man beliebig dimensionale lineare Programme lösen kann.[1] Allerdings stehen in kommerziellen Implementierungen nur numerische Versionen zur Verfügung, so dass eine Aussage über die Stabilitätsbereiche, wie wir sie im vorigen Abschnitt gemacht haben, nicht möglich ist. Aus diesem Grund wurde der in Algorithmus 1 vorgestellte Simplex mit Hilfe eines Computeralgebrasystems (CAS) implementiert, um alle Operationen symbolisch durchführen zu können. Als CAS wurde das System Mathematica von Wolfram Research gewählt, da es neben einer einfachen Benutzeroberfläche, einer hochentwickelten Programmiersprache und ausgeprägten symbolischen Fähigkeiten auch über sehr effiziente numerische Algorithmen verfügt. Dies hat den Vorteil, dass ein direkter Vergleich von symbolischen mit numerischen Ergebnissen innerhalb eines einzigen Programmes möglich ist.

[1] Theoretisch können mit dem Simplex-Algorithmus beliebig große lineare Programme, sowohl an Variablenzahl als auch an Spaltenzahl, gelöst werden. In der Praxis beschränken allerdings die Rechenkapazität und die Rechendauer die Größe der lösbaren linearen Programme. Sollen dennoch sehr große Probleme gelöst werden, muss man auf spezielle Techniken, z.B. Spaltengenerierung, zurückgreifen. Für weitergehende Informationen sei auf [3] verwiesen.

4.3.1 Mathematica

Das CAS Mathematica wird von Wolfram Research entwickelt und vertrieben und ist aktuell in der Version 6 verfügbar. Firmengründer Stephen Wolfram begann 1986 mit der Entwicklung von Mathematica und 1988 wurde Version 1 veröffentlicht. Die Mathematica-eigene Programmiersprache bietet jedem Benutzer die Möglichkeit nach seinen eigenen Vorlieben zu programmieren. So kann man in Mathematica nicht nur in klassischer Weise prozedural oder objektorientiert programmieren, sondern auch funktional und regelbasiert. Eine Mischung aus allen Bereichen ist ebenso möglich. Außerdem erlaubt sie die Eingabe von Symbolen, z.B. $\sum$ oder $\int$, innerhalb eines Ausdrucks. Neben den rein mathematischen Funktionen bietet Mathematica auch vielfältigste Möglichkeiten Ergebnisse ansprechend zu visualisieren. So können nicht nur statische Funktionsplots erstellt werden, sondern auch interaktive Animationen von äußerst komplexen Vorgängen. Die Standardfunktionspalette von Mathematica beschränkt sich nicht nur auf Grundfähigkeiten, wie Addition oder Integralberechnung, sondern umfasst auch Methoden zum Lösen von gewöhnlichen und partiellen Differentialgleichungen, aus der Optimierung, der Statistik und Kombinatorik, sowie aus der diskreten Mathematik und Zahlentheorie. All diese Verfahren können sowohl mit symbolischen als auch mit numerischen Werten verwendet werden. Darüberhinaus gibt es noch eine große Zahl an Paketen, z.B. für Finanzmathematik oder Mechanik, die den Funktionsumfang nochmals erweitern.

4.3.2 Der Quellcode

Im nachfolgenden ist der Quellcode für die Funktion `SymSimplex` angegeben. Wir gehen hier von einem System in Standardform nach Definition 3.5 aus, d.h.

$$\max c^T x \tag{4.10}$$
$$Ax = b \tag{4.11}$$
$$x \geq 0 \tag{4.12}$$

Dies bedeutet für die Nebenbedingungen (4.2) und (4.3), dass die Ungleichungen durch Einführen von Schlupfvariablen zu Gleichungen gemacht werden müssen. Um die Erzeugung der Koeffizientenmatrix `A`, der rechten Seite `b` und des Zielfunktionsvektors `c` zu vereinfachen wurde eine Funktion `Nebenbedingungen` geschrieben, die diese Eingabedaten in passender Weise erzeugt. Die Funktion `Nebenbedingungen` ist in Anhang A.1 beschrieben.

Damit können wir nun zusammenfassend die von der Funktion `SymSimplex` als Input benötigten Werte angeben:

- `A`: Koeffizientenmatrix inklusive der Schlupfvariablen

- `b`: rechte Seite des LP
- `c`: Richtung der Zielfunktion
- `Basis`: Eine gültige Anfangsbasis für A
- `Ausgabe`: Boolscher Wert, der bestimmt ob das Tableux in jedem Schritt ausgegeben wird
- `Annahme`: Einschränkungen und Annahmen an die verwendeten Parameter

In der Matrix `A` können beliebige symbolische Werte verwendet werden, die den in `Annahme` gemachten Einschränkungen unterliegen.

Nun der Quelltext von `SymSimplex`:

```
SymSimplex[A_, b_, c_, Basis_, Ausgabe_, Annahme_] :=
  Module[{AB, AN, B, Ni, ABAN, ABb, cNcBABAN, cBABb, cc, te,
          k, tes, s, r, opt, mm, i, entscheidbar},
    B = Basis;
    Ni = Complement[Table[i, {i, Dimensions[A][[2]]}], B];
    opt = False;

    While[opt != True,
      If[Ausgabe == True,
        Print["aktuelle Basis B= ", B];
        Print["aktuelle Nichtbasis N= ", Ni]; ];

        AB = Transpose[Transpose[A][[B]]];
        AN = Transpose[Transpose[A][[Ni]]];
        ABinv = Inverse[AB];
        ABAN = Simplify[ABinv . AN];
        ABb = Simplify[ABinv . b];
        cNcBABAN = Simplify[c[[Ni]]-c[[B]].ABinv.AN];
        cBABb = Simplify[-c[[B]].ABinv.b];
        cBABb = Flatten[cBABb];

        entscheidbar = Cases[(Refine[#1 > 0, Annahme] & )
                             /@ cNcBABAN, _?(Head[#1] == Greater & )];
        If[entscheidbar != {},
          Print["\n\nKeine optimale Basis gefunden!\n
                 Grund: Folgender Ausdruck kann unter den gemachten
                 Annahmen (",Annahme,") nicht entschieden werden:"];
          Print[entscheidbar];
          Return["ABBRUCH"]];

        If[Ausgabe == True,
          mm = Transpose[Append[Transpose[Prepend[ABAN, cNcBABAN]],
                                Flatten[Transpose[Prepend[ABb, cBABb]]]]];
          Print[Grid[mm, Dividers -> {{-2 -> True}, {2 -> True}},
```

```
                FrameStyle -> Directive[Thickness[2]]]];];

        If[Refine[And@@(Refine[#1<=0,Annahme]&)/@cNcBABAN,Annahme],
            opt = True;
            Print["optimale Basis B=", B];
            Return[cBABb[[1]]];,
            te = Select[cNcBABAN, Refine[#1 > 0, Annahme] & , 1];
            s = Position[cNcBABAN, _?(#1 == te[[1]] & )][[1,1]];
            k = 1;
            te = Cases[tes = Transpose[{Flatten[Transpose[ABb]],
                        ABAN[[All,s]]}], {a_, f_} :> {a, f, k++}];
            r = Select[cc=Cases[te,{a_,(f_)?(Refine[#1>0,Annahme]&),g_}
                        ->{a/f,g}],#1[[1]] == Refine[Min[cc[[All,1]]],
                        Annahme] &][[1,2]];
            te = B[[r]];
            B[[r]] = Ni[[s]];
            Ni[[s]] = te;];
    ];
  ];
```

Programm 4.1: Symbolischer Simplex. Liefert eine optimale Basis, falls alle Ausdrücke eindeutig entschieden werden können.

Falls das LP eine optimale Lösung besitzt, liefert die Funktion `SymSimplex` neben der optimalen Basis auch gleich den Zielfunktionswert als Output. Ein typischer Aufruf sieht dann etwa wie folgt aus:

```
In[3]:= {A43, b43, c43, B43} = Nebenbedingungen[4, 3, 2, True]

Out[3]= {{{0, 0, 0, -Ce - Ce σ - Ce σ^2 - Ce σ^3, 1, 1, 0, 0, 0},
    {1 - Ce - Ce σ - Ce σ^2 - Ce σ^3, 1, 1, 1, 0, 0, 1, 0, 0},
    {0, 1 - Ce - Ce σ - Ce σ^2, 1, 1, 0, 0, 0, 1, 0},
    {0, 0, 1 - Ce - Ce σ, 1, 0, 0, 0, 0, 1}, {1, 1, 1, 0, 0, 0, 0, 0, 0}},
   {{0}, {0}, {0}, {0}, {1}}, {-1, -1, -1, -1, 1, 0, 0, 0, 0}, {1, 6, 7, 8, 9}}

In[4]:= SymSimplex[A43, b43, c43, B43, False, {Ce ≥ 1, 0 < σ < 1}]

optimale Basis B={1, 5, 3, 4, 2}

Out[4]= -(Ce (1 + σ + σ^2 + σ^3)
      (-1 - Ce^2 (1 + σ) (2 + σ + σ^2)^2 + Ce (3 + 3 σ + 2 σ^2 + σ^3) + Ce^3 (1 + σ)^2 (1 + σ + 2 σ^2 + σ^3 + σ^4))) /
    (1 - Ce (3 + 3 σ + 2 σ^2 + σ^3) + Ce^2 (3 + 6 σ + 7 σ^2 + 6 σ^3 + 3 σ^4 + σ^5))
```

Abbildung 4.4: Typischer Ablauf von `SymSimplex`. In einem ersten Schritt werden die Eingabedaten erstellt. In diesem Beispiel für $N = 4, m = 3$ und $\beta(r, n) = C\sigma^n r$. Diese werden an `SymSimplex` übergeben. Zusätzlich werden noch die Ausgabeart und die Annahmen ($C \geq 1, 0 < \sigma < 1$) angegeben. Als Ausgabe erhält man die optimale Basis und den Zielfunktionswert als Funktion in Abhängigkeit von C und σ.

4.4 Weitere Stabilitätsbereiche

Die im vorigen Abschnitt beschriebene Funktion `SymSimplex` wurde nun benutzt um noch weitere Stabilitätsbereiche zu untersuchen und zu vergleichen. In Abbildung

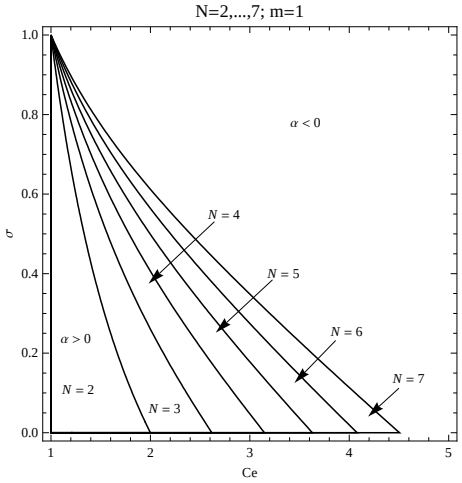

Abbildung 4.5: Stabilitätsbereiche für $N = 2, \ldots, 7$ und $m = 1$

4.6 sind die Stabilitätsbereiche für $\beta(r, n) = C\sigma^n r$ und $N = 2, \ldots, 6$ gezeichnet. Auffällig ist, dass der Stabilitätsbereich $S(N, 1)$ immer eine Teilmenge des Stabilitätsbereiches $S(N, m), m > 1$ ist. Wenn also das System mit einem Optimierungshorizont N schon für den Kontrollhorizont $m = 1$ stabil ist, dann ist es auch für alle anderen Kontrollhorizonte stabil, da $S(N, m) \supseteq S(N, 1)$ gilt. Betrachtet man die Stabilitätsbereiche $S(2, 1), \ldots, S(7, 1)$ (siehe Abbildung 4.5), so erkennt man, dass

$$S(2, 1) \subset S(3, 1) \subset S(4, 1) \subset S(5, 1) \subset S(6, 1) \subset S(7, 1) \tag{4.13}$$

gilt. Man kann also nicht nur den Kontrollhorizont vergrößern, sondern auch den Optimierungshorizont, wenn man mit einem kleineren N schon stabil ist. Dies ist aber nicht verwunderlich, da sich hier Satz 2.5 widerspiegelt. Mit größer werdendem Optimierungshorizont werden die berechneten Steuerungen immer besser, und somit auch α. Also kann ein System, das mit einem großem Optimierungshorizont stabil geregelt wird, bei einem noch größeren nicht instabil werden.

Weiterhin sieht man, dass die Stabilitätsbereiche $S(3, 1)$ und $S(3, 2)$ identisch sind. Dass sie das nicht nur im Plot sind, erkennt man auch an den Zielfunktionswerten $\alpha(3, 1)$ und $\alpha(3, 2)$ (siehe Abbildung 4.7). Bei den anderen Stabilitätsbereichen zeigt sich das gleiche Verhalten, was sich auch jedesmal aus den gleichen Zielfunktionswerten ergibt.

Damit kann man sagen, dass die Wahl der Parameter C und σ einen entscheidenden Einfluss auf die Stabilität des Systems hat. Hat man eine $\mathcal{KL}_0$-Funktion β mit

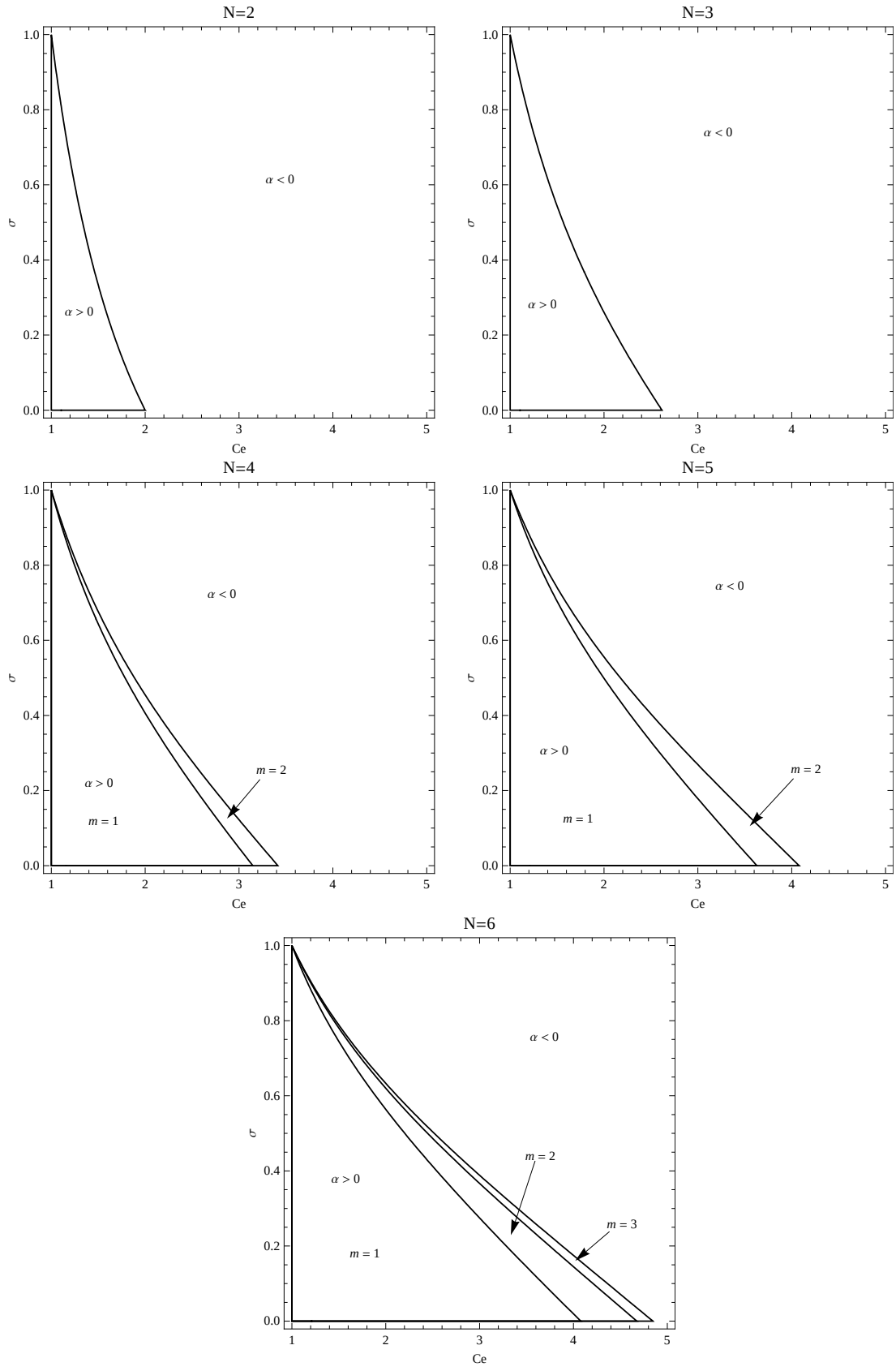

Abbildung 4.6: Stabilitätsbereiche für $N = 2, \ldots, 6$ und $\beta(r, n) = C\sigma^n r$.

```
In[5]:= {A31, b31, c31, B31} = Nebenbedingungen[3, 1, 2, True];
        SymSimplex[A31, b31, c31, B31, False, {Ce ≥ 1, 0 < σ < 1}]

optimale Basis B={1, 4, 2, 3, 8}

Out[6]= -(Ce (1 + σ + σ^2) (1 - Ce (3 + 3 σ + σ^2) + Ce^2 (1 + 2 σ + 2 σ^2 + σ^3))) / (-1 + Ce (2 + 2 σ + σ^2))

In[7]:= {A32, b32, c32, B32} = Nebenbedingungen[3, 2, 2, True];
        SymSimplex[A32, b32, c32, B32, False, {Ce ≥ 1, 0 < σ < 1}]

optimale Basis B={1, 4, 2, 3}

Out[8]= -(Ce (1 + σ + σ^2) (1 - Ce (3 + 3 σ + σ^2) + Ce^2 (1 + 2 σ + 2 σ^2 + σ^3))) / (-1 + Ce (2 + 2 σ + σ^2))
```

Abbildung 4.7: Zielfunktionswerte für $N = 3$ und $m = 1, 2$.

Parametern C und σ gegeben, so kann man den Optimierungshorizont N gerade so wählen, dass C und σ im Stabilitätsbereich liegen.
Desweiteren haben wir gesehen, dass die Zielfunktionswerte von „symmetrisch“ liegenden Kontrollhorizonten gleich sind, d.h.

$$\begin{aligned} \alpha(N,1) &= \alpha(N,N-1) && (4.14)\\ \alpha(N,2) &= \alpha(N,N-2) && (4.15)\\ &\vdots \\ \alpha(N,\lfloor N/2 \rfloor) &= \alpha(N,\lceil N/2 \rceil) && (4.16) \end{aligned}$$

4.5 Stabilitätsbereiche für $\beta(r,t) = \gamma r$

Im vorigen Abschnitt haben wir die Stabilitätsbereiche für die $\mathcal{KL}_0$-Funktion $\beta(r,t) = C\sigma^t r$ betrachtet. Nun wollen wir die Stabilitätsbereiche für

$$\beta(r,n) = \begin{cases} \gamma r & n = 0 \\ 0 & n \geq 1 \end{cases}. \tag{4.17}$$

untersuchen. Hierzu betrachten wir exemplarisch die Zielfunktionswerte für $N = 5, m = 1, 2, 3, 4$, die wir ebenfalls mit der Funktion `SymSimplex` bestimmen können. Diese lauten:

$$\left\{ \frac{\gamma\,(1 - 4\gamma + 6\gamma^2 - 5\gamma^3 + \gamma^4)}{1 - 4\gamma + 6\gamma^2 - 4\gamma^3}, -\frac{\gamma^2\,(-1 + 4\gamma - 5\gamma^2 + \gamma^3)}{-1 + 5\gamma - 9\gamma^2 + 6\gamma^3}, \right.$$
$$\left. -\frac{\gamma^2\,(-1 + 4\gamma - 5\gamma^2 + \gamma^3)}{-1 + 5\gamma - 9\gamma^2 + 6\gamma^3}, \frac{\gamma\,(1 - 4\gamma + 6\gamma^2 - 5\gamma^3 + \gamma^4)}{1 - 4\gamma + 6\gamma^2 - 4\gamma^3} \right\}$$

Hier sieht man ganz deutlich, dass $\alpha(5,1)$ und $\alpha(5,4)$ identisch sind, ebenso wie $\alpha(5,2)$ und $\alpha(5,3)$ übereinstimmen. Zeichnet man diese Kurven, so sieht man auch, dass $\alpha(5,2) \geq \alpha(5,1)$ ist (siehe Abbildung 4.8). Dies entspricht dem schon vorher gesehenen Verhalten, dass $S(5,1) \subseteq S(5,2)$. $S(5,1)$ enspricht hier dem Intervall $[1, 3.62966[$ und $S(5,2) \approx [1, 4.07960[$, wobei die rechten Grenzen der Intervalle die numerisch bestimmten Nullstellen der Zielfunktionen sind.

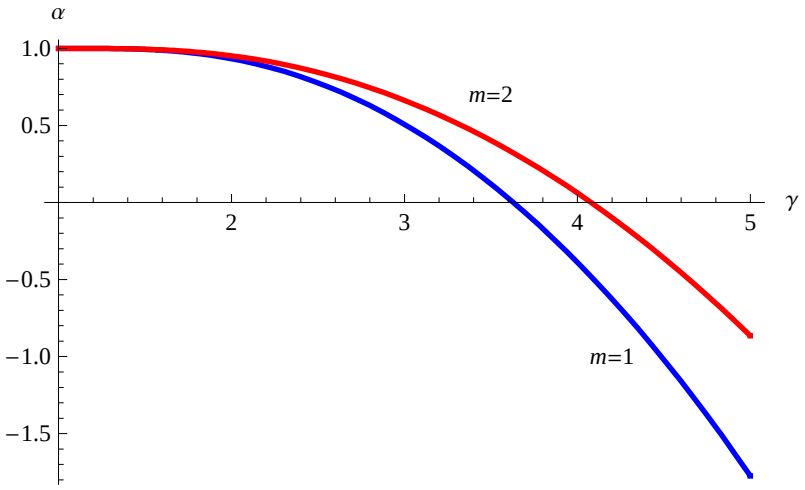

Abbildung 4.8: Zielfunktionswerte α für $N = 5$ und $m = 1, 2$

N \ m	1	2	3	6	5	6	7	8
2	2.00000							
3	2.61803	2.61803						
4	3.14790	3.41421	3.1479					
5	3.62966	4.07960	4.07960	3.62966				
6	4.07960	4.67621	4.84732	4.67621	4.0796			
7	4.50632	5.22823	5.53013	5.53013	5.22823	4.50632		
8	4.91508	5.74816	6.15839	6.28521	6.15839	5.74816	4.91508	
9	5.30933	6.24349	6.74762	6.97714	6.97714	6.74762	6.24349	5.30933

Tabelle 4.2: Nullstellen der Zielfunktionswerte. Das System ist stabil, wenn der Parameter γ in der $\mathcal{KL}_0$-Funktion β zwischen 1 und der Nullstelle gewählt werden kann.

In Abbildung 4.9 sind die Zielfunktionswerte in Abhängigkeit von γ für $N = 2, \ldots, 9$ dargestellt. Hier sieht man auch wieder ganz deutlich, dass $\alpha(N, m = 1) = \alpha(N, m = N-1)$, usw. ist. Betrachtet man nun die Nullstellen dieser Funktionen (siehe Tabelle 4.2), so sieht man, dass die Stabilitätsbereiche $S(N,m) = [1, \chi[$, wobei χ die positive Nullstelle der Zielfunktionen ist, das gleiche Verhalten zeigen, wie für $\beta(r,t) = C\sigma^t r$,

N \ m	1	2	3	4
2	$-(-2+\gamma)\gamma$			
3	$\frac{\gamma\left(1-3\gamma+\gamma^2\right)}{1-2\gamma}$	$\frac{\gamma\left(1-3\gamma+\gamma^2\right)}{1-2\gamma}$		
4	$-\frac{\gamma\left(-1+3\gamma-4\gamma^2+\gamma^3\right)}{1-3\gamma+3\gamma^2}$	$-\frac{\gamma^2\left(2-4\gamma+\gamma^2\right)}{(1-2\gamma)^2}$	$-\frac{\gamma\left(-1+3\gamma-4\gamma^2+\gamma^3\right)}{1-3\gamma+3\gamma^2}$	
5	$\frac{\gamma\left(1-4\gamma+6\gamma^2-5\gamma^3+\gamma^4\right)}{1-4\gamma+6\gamma^2-4\gamma^3}$	$-\frac{\gamma^2\left(-1+4\gamma-5\gamma^2+\gamma^3\right)}{-1+5\gamma-9\gamma^2+6\gamma^3}$	$-\frac{\gamma^2\left(-1+4\gamma-5\gamma^2+\gamma^3\right)}{-1+5\gamma-9\gamma^2+6\gamma^3}$	$\frac{\gamma\left(1-4\gamma+6\gamma^2-5\gamma^3+\gamma^4\right)}{1-4\gamma+6\gamma^2-4\gamma^3}$
6	$-\frac{\gamma\left(-1+5\gamma-10\gamma^2+10\gamma^3-6\gamma^4+\gamma^5\right)}{1-5\gamma+10\gamma^2-10\gamma^3+5\gamma^4}$	$-\frac{\gamma^2\left(1-4\gamma+7\gamma^2-6\gamma^3+\gamma^4\right)}{(1-2\gamma)^2(1-2\gamma+2\gamma^2)}$	$-\frac{\gamma^3\left(-2+6\gamma-6\gamma^2+\gamma^3\right)}{(1-3\gamma+3\gamma^2)^2}$	$-\frac{\gamma^2\left(1-4\gamma+7\gamma^2-6\gamma^3+\gamma^4\right)}{(1-2\gamma)^2(1-2\gamma+2\gamma^2)}$
7	$\frac{\gamma\left(1-6\gamma+15\gamma^2-20\gamma^3+15\gamma^4-7\gamma^5+\gamma^6\right)}{1-6\gamma+15\gamma^2-20\gamma^3+15\gamma^4-6\gamma^5}$	$-\frac{\gamma^2\left(-1+5\gamma-10\gamma^2+11\gamma^3-7\gamma^4+\gamma^5\right)}{-1+7\gamma-20\gamma^2+30\gamma^3-25\gamma^4+10\gamma^5}$	$\frac{\gamma^3\left(1-5\gamma+9\gamma^2-7\gamma^3+\gamma^4\right)}{1-7\gamma+21\gamma^2-34\gamma^3+30\gamma^4-12\gamma^5}$	$\frac{\gamma^3\left(1-5\gamma+9\gamma^2-7\gamma^3+\gamma^4\right)}{1-7\gamma+21\gamma^2-34\gamma^3+30\gamma^4-12\gamma^5}$
8	$-\frac{\gamma\left(-1+7\gamma-21\gamma^2+35\gamma^3-35\gamma^4+21\gamma^5-8\gamma^6+\gamma^7\right)}{1-7\gamma+21\gamma^2-35\gamma^3+35\gamma^4-21\gamma^5+7\gamma^6}$	$-\frac{\gamma^2\left(1-6\gamma+15\gamma^2-20\gamma^3+16\gamma^4-8\gamma^5+\gamma^6\right)}{(1-2\gamma)^2(1-4\gamma+7\gamma^2-6\gamma^3+3\gamma^4)}$	$-\frac{\gamma^3\left(-1+5\gamma-11\gamma^2+13\gamma^3-8\gamma^4+\gamma^5\right)}{1-8\gamma+28\gamma^2-55\gamma^3+65\gamma^4-45\gamma^5+15\gamma^6}$	$-\frac{\gamma^4\left(2-8\gamma+12\gamma^2-8\gamma^3+\gamma^4\right)}{(1-4\gamma+6\gamma^2-4\gamma^3)^2}$
9	$\frac{\gamma\left(1-8\gamma+28\gamma^2-56\gamma^3+70\gamma^4-56\gamma^5+28\gamma^6-9\gamma^7+\gamma^8\right)}{1-8\gamma+28\gamma^2-56\gamma^3+70\gamma^4-56\gamma^5+28\gamma^6-8\gamma^7}$	$-\frac{\gamma^2\left(-1+7\gamma-21\gamma^2+35\gamma^3-35\gamma^4+22\gamma^5-9\gamma^6+\gamma^7\right)}{-1+9\gamma-35\gamma^2+77\gamma^3-105\gamma^4+91\gamma^5-49\gamma^6+14\gamma^7}$	$-\frac{\gamma^3\left(1-6\gamma+15\gamma^2-21\gamma^3+18\gamma^4-9\gamma^5+\gamma^6\right)}{(1-3\gamma+3\gamma^2)^2(-1+3\gamma-3\gamma^2+2\gamma^3)}$	$-\frac{\gamma^4\left(-1+6\gamma-14\gamma^2+16\gamma^3-9\gamma^4+\gamma^5\right)}{-1+9\gamma-36\gamma^2+84\gamma^3-125\gamma^4+120\gamma^5-70\gamma^6+20\gamma^7}$

N \ m	5	6	7	8
6	$-\frac{\gamma\left(-1+5\gamma-10\gamma^2+10\gamma^3-6\gamma^4+\gamma^5\right)}{1-5\gamma+10\gamma^2-10\gamma^3+5\gamma^4}$			
7	$-\frac{\gamma^2\left(-1+5\gamma-10\gamma^2+11\gamma^3-7\gamma^4+\gamma^5\right)}{-1+7\gamma-20\gamma^2+30\gamma^3-25\gamma^4+10\gamma^5}$	$\frac{\gamma\left(1-6\gamma+15\gamma^2-20\gamma^3+15\gamma^4-7\gamma^5+\gamma^6\right)}{1-6\gamma+15\gamma^2-20\gamma^3+15\gamma^4-6\gamma^5}$		
8	$-\frac{\gamma^3\left(-1+5\gamma-11\gamma^2+13\gamma^3-8\gamma^4+\gamma^5\right)}{1-8\gamma+28\gamma^2-55\gamma^3+65\gamma^4-45\gamma^5+15\gamma^6}$	$-\frac{\gamma^2\left(1-6\gamma+15\gamma^2-20\gamma^3+16\gamma^4-8\gamma^5+\gamma^6\right)}{(1-2\gamma)^2(1-4\gamma+7\gamma^2-6\gamma^3+3\gamma^4)}$	$-\frac{\gamma\left(-1+7\gamma-21\gamma^2+35\gamma^3-35\gamma^4+21\gamma^5-8\gamma^6+\gamma^7\right)}{1-7\gamma+21\gamma^2-35\gamma^3+35\gamma^4-21\gamma^5+7\gamma^6}$	
9	$-\frac{\gamma^4\left(-1+6\gamma-14\gamma^2+16\gamma^3-9\gamma^4+\gamma^5\right)}{-1+9\gamma-36\gamma^2+84\gamma^3-125\gamma^4+120\gamma^5-70\gamma^6+20\gamma^7}$	$-\frac{\gamma^3\left(1-6\gamma+15\gamma^2-21\gamma^3+18\gamma^4-9\gamma^5+\gamma^6\right)}{(1-3\gamma+3\gamma^2)^2(-1+3\gamma-3\gamma^2+2\gamma^3)}$	$-\frac{\gamma^2\left(-1+7\gamma-21\gamma^2+35\gamma^3-35\gamma^4+22\gamma^5-9\gamma^6+\gamma^7\right)}{-1+9\gamma-35\gamma^2+77\gamma^3-105\gamma^4+91\gamma^5-49\gamma^6+14\gamma^7}$	$\frac{\gamma\left(1-8\gamma+28\gamma^2-56\gamma^3+70\gamma^4-56\gamma^5+28\gamma^6-9\gamma^7+\gamma^8\right)}{1-8\gamma+28\gamma^2-56\gamma^3+70\gamma^4-56\gamma^5+28\gamma^6-8\gamma^7}$

Abbildung 4.9: Zielfunktionswerte in Abhängigkeit von γ für $N = 2, \ldots, 9$.

d.h. hier gilt auch

$$S(N,m) \subset S(N,m+1) \qquad m = 1,\ldots,\lceil N/2 \rceil \tag{4.18}$$

$$S(2,1) \subset S(3,1) \subset S(4,1) \subset S(5,1) \subset S(6,1) \subset \ldots \tag{4.19}$$

Dies bedeutet, dass die Stabilität für alle Kontrollhorizonte gewährleistet ist, wenn das System für den Kontrollhorizont $m = 1$ stabil ist. Kann man dies nicht erreichen, so kann man unter Umständen noch für $m = 2$ Stabiliät erreichen.

4.6 Bestimmung der Stabilitätsgrenze

Zur Untersuchung der Frage, ob das in Abbildung 4.1 dargestellte Verhalten typisch ist, wurde der MATLAB-Code aus [14] erweitert (siehe Anhang A.2).

Der Aufruf von

```
[alpha,Nall,Nab,mab]=alpham(12,1.5,0.85)
```

berechnet alle Werte von α für $N = 2,\ldots,12$ und alle $m = 1,\ldots,N-1$. Außerdem wird noch das kleinste N zurückgeliefert, bei dem alle $\alpha > 0$ sind. Zuletzt wird noch dasjenige N bestimmt, bei dem als erstes ein $\alpha > 0$ ist. Zu diesem α wird auch der Kontrollhorizont m bestimmt.

In Abbildung 4.11 ist die Ausgabe für den obigen Aufruf wiedergegeben.Für die Parameter $C = 1.5$ und $\sigma = 0.85$ ist also der erste stabile Wert bei $N = 9$ und $m = 3$. α ist für alle $m = 1,\ldots,N-1$ größer als Null ab $N = 12$.

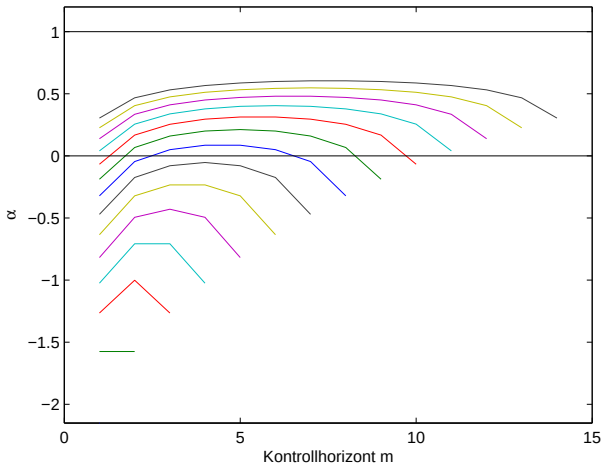

Abbildung 4.10: α aus Abbildung 4.11 grafisch dargestellt. Die konkave und symmetrische Form ist deutlich erkennbar.

```
ans =

Ergebnis: 1.500 0.850   9,3     12


alpha =

      0         0         0         0         0         0         0         0         0         0         0
 -2.1506         0         0         0         0         0         0         0         0         0         0
 -1.5749   -1.5749         0         0         0         0         0         0         0         0         0
 -1.2654   -1.0012   -1.2654         0         0         0         0         0         0         0         0
 -1.0250   -0.7078   -0.7078   -1.0250         0         0         0         0         0         0         0
 -0.8182   -0.4948   -0.4289   -0.4948   -0.8182         0         0         0         0         0         0
 -0.6347   -0.3215   -0.2332   -0.2332   -0.3215   -0.6347         0         0         0         0         0
 -0.4700   -0.1738   -0.0786   -0.0533   -0.0786   -0.1738   -0.4700         0         0         0         0
 -0.3217   -0.0454    0.0500    0.0861    0.0861    0.0500   -0.0454   -0.3217         0         0         0
 -0.1880    0.0676    0.1598    0.2002    0.2120    0.2002    0.1598    0.0676   -0.1880         0         0
 -0.0676    0.1675    0.2549    0.2964    0.3139    0.3139    0.2964    0.2549    0.1675   -0.0676         0
  0.0408    0.2562    0.3380    0.3788    0.3989    0.4051    0.3989    0.3788    0.3380    0.2562    0.0408
```

Abbildung 4.11: α für $N = 2, \ldots, 12$ und jeweils $m = 1, \ldots, N-1$

In Abbildung 4.10 kann man deutlich erkennen, dass die konkave und symmetrische Form der Zielfunktion in Abhänigkeit von m keine Einzelerscheinung ist. Vielmehr spiegelt sie das Verhalten wider, dass wir auch schon bei der Betrachtung der Stabilitätsbereiche gesehen haben. Zum einen sieht man auch hier, dass symmetrisch liegende Kontrollhorizonte den gleichen Zielfunktionswert liefern, zum anderen dass die Stabilität des Systems für alle Kontrollhorizonte gewährleistet ist, wenn sie für den ersten bzw. den letzten gewährleistet ist.

4.7 Minimale stabilisierende Horizonte

Im vorigen Abschnitt haben wir gesehen, dass der Zielfunktionswert für $m = \lceil N/2 \rceil$ am größten ist, also die beste Abschätzung für V_N liefert. Wir wollen nun für β aus Abschnitt 4.5 untersuchen, wie groß der minimale stabilisierende Optimierungshorizont ist, wenn wir $m = \lceil N/2 \rceil$ als Kontrollhorizont wählen. Hierzu betrachten wir die Funktion

$$N(\gamma, m) := \min\{N \in \mathbb{N} | \alpha(N, m, \gamma) > 0\}. \tag{4.20}$$

Wir setzen also $m = \lceil N/2 \rceil$ und vergrößern N so lange bis $\alpha(N, m, \gamma) > 0$ ist. Zum Vergleich betrachten wir auch noch $N(\gamma, 1)$. In Abbildung 4.12 sind die minimalen stabilisierenden Horizonte $N(\gamma, m)$ für $m = 1$ und $m = \lceil N/2 \rceil$ jeweils für $\gamma = 1, \ldots, 50$ dargestellt. Man stellt fest, dass $N(\gamma, 1)$ sehr gut von $\gamma \log \gamma$ und $N(\gamma, \lceil N/2 \rceil)$ von 1.4γ angenähert wird. Ebenso sieht man, dass es „einfacher“ ist, d.h. dass man einen kleineren Optimierungshorizont braucht, das System zu regeln, wenn man $m = \lceil N/2 \rceil$ wählt. In Abbildung 4.13 sind auch die minimalen stabilisierenden Horizonte für $m = 3, 4, 9$ eingezeichnet. Hier erkennt man auch, dass der Optimierungshorizont N umso kleiner gewählt werden kann, je größer man m wählt. Mit $m = \lceil N/2 \rceil$ ist allerdings die Grenze erreicht, da wir aus den vorherigen Abschnitten wissen, dass die Zielfunktionswerte symmetrisch zu den Kontrollhorizonten sind, und dass bei $m = \lceil N/2 \rceil$ das größte α angenommen wird.

4.8 Zusammenfassung

In diesem Kapitel haben wir untersucht, wie sich der Kontrollhorizont auf die Stabilität eines mit MPC geregeltem Systems auswirkt. Wir haben dazu die Stabilitätsbereiche für verschiedene $\mathcal{KL}_0$-Funktionen untersucht und dabei gesehen, dass für Stabilitätsbereiche $S(N, m) \subset S(N, m+1) \qquad m = 1, \ldots, \lceil N/2 \rceil$ gilt. Außerdem konnten wir mit Hilfe einer symbolischen Version des Simplex-Algorithmus zeigen, dass die Zielfunktionswerte für symmetrisch liegende Kontrollhorizonte identisch sind. Hieraus folgt auch, dass die Stabilitätsbereiche $S(N, k)$ und $S(N, N-k)$ für $k = 1, \ldots, \lceil N/2 \rceil$ identisch sind. Dies bedeutet für das System, dass die Stabilität für

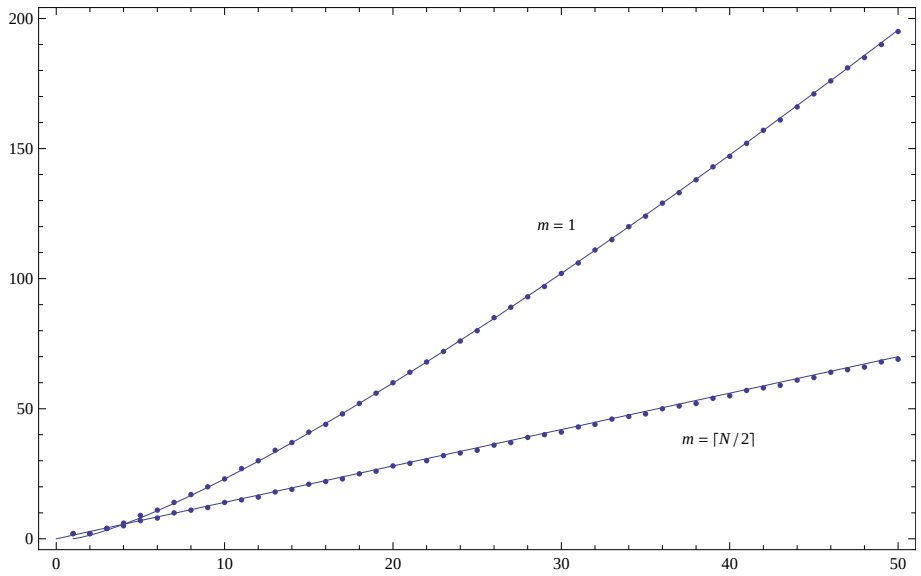

Abbildung 4.12: Minimale stabilisierende Optimierungshorizonte für $m = 1$ und $m = \lceil N/2 \rceil$. Dargestellt sind die Optimierungshorizonte N in Abhängigkeit von γ, so dass $\alpha(N, m, \gamma) > 0$ gilt, d.h. $N(\gamma) := \min\{N \in \mathbb{N} | \alpha(N, m, \gamma) > 0\}$. Man sieht die sehr gute Übereinstimmung mit den Funktionen $\gamma \log \gamma$ für $m = 1$ und 1.4γ für $m = N(\lceil N/2 \rceil)$.

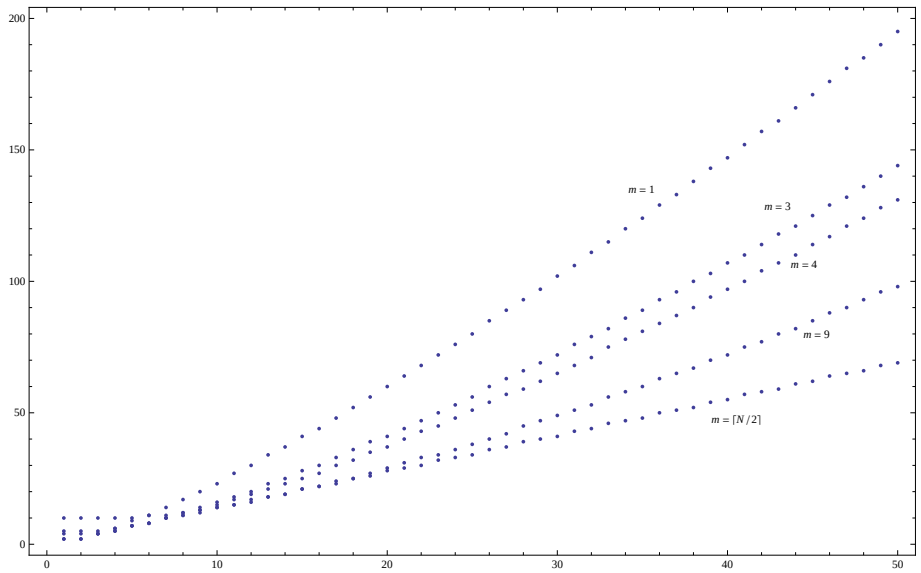

Abbildung 4.13: Minimale stabilisierende Optimierungshorizonte für $m = 1, 3, 4, 9$ und $m = \lceil N/2 \rceil$. Dargestellt sind die Optimierungshorizonte N in Abhängigkeit von γ, so dass $\alpha(N, m) > 0$ gilt, d.h. $N(\gamma) := \min\{N \in \mathbb{N} | \alpha(N, m, \gamma) > 0\}$.

alle Kontrollhorizonte gewährleistet ist, wenn sie schon für den kleinst möglichen, also $m = 1$, gewährleistet ist. Zum Schluss haben wir noch untersucht, wie groß der Optimierungshorizont mindestens sein muss, um bei vorgegebenem m Stabilität zu erreichen. Hier hat sich gezeigt, dass für $m = \lceil N/2 \rceil$ der Optimierunghorizont deutlich kleiner gewählt werden muss, als bei allen anderen Kontrollhorizonten. Dies spiegelt sich auch in der Form der Kurven wider, wenn man den Kontrollhorizont gegen den Zielfunktionswert α aufträgt.

5 Anregungen

Im Rahmen dieser Arbeit ist es nicht gelungen die numerischen Ergebnisse aus Kapitel 4 durch Beweise zu untermauern. Hier soll nun ein möglicher Ansatz für einen solchen Beweis vorgestellt werden.

Während der Berechnungen der Zielfunktionswerte mit Hilfe der Funktion `SymSimplex` fiel auf, dass für den Kontrollhorizont $m = N - 1$ die optimale Nichtbasis stets vollständig aus den Schlupfvariablen besteht. D.h. das Optimum ergibt sich als Lösung des Gleichungssystemes, das aus den $N + 1$ Nebenbedingugen besteht. Was wiederum bedeutet, dass man nur ein lineares Gleichungssystem lösen muss und kein lineares Programm. Um dies zu zeigen könnte man folgendermaßen vorgehen:

1. Zeige, dass die Determinante der Koeffizientenmatrix ungleich Null ist. Damit wäre die Koeffizientenmatrix invertierbar und somit eine Basis.
2. Zeige, dass diese Basis zulässig ist.
3. Zeige, dass die reduzierten Kosten mit dieser Basis negativ sind, d.h. diese ist optimal.

Für die Berechnung der Determinante aus Punkt 1 konnte bereits eine Rekursionsformel erarbeitet werden. Als $\mathcal{KL}_0$-Funktion wurde (4.17) gewählt, da sich damit die Nebenbedingungen stark vereinfachen und nur ein Parameter betrachtet werden muss. Dies können wir nun in einem Lemma formulieren.

Lemma 5.1
Sei β aus (4.17) gegeben. Sei K_N die Koeffizientenmatrix der Nebenbedingungen (4.2)-(4.4). Dann ist die Determinante $D_N(\gamma)$ von K_N gegeben durch

$$D_N(\gamma) = (\gamma - 1)D_{N-1}(\gamma) - \gamma^{N-2} \tag{5.1}$$

mit $D_2(\gamma) = -1$.

Beweis: Wir wollen dies mittels Induktion zeigen.
Induktionsanfang $N = 2$:

$$\det \begin{pmatrix} 0 & -\gamma & 1 \\ 1-\gamma & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} = \det \begin{pmatrix} -\gamma & 1 \\ 1 & 0 \end{pmatrix} = -1 \tag{5.2}$$

Damit lautet die Induktionsvoraussetzung: $D_N(\gamma) = (\gamma-1)D_{N-1}(\gamma) - \gamma^{N-2}$.
Der Induktionsschritt $N \to N+1$ ist gegeben durch

$$D_{N+1}(\gamma) = \det \overbrace{\begin{pmatrix} 0 & 0 & \cdots & 0 & -\gamma & 1 \\ 1-\gamma & 1 & \cdots & \cdots & 1 & 0 \\ 0 & 1-\gamma & \ddots & & \vdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 1-\gamma & 1 & 0 \\ 1 & \cdots & \cdots & 1 & 0 & 0 \end{pmatrix}}^{N+2} \tag{5.3}$$

$$\begin{aligned} &\text{entwickeln nach der N+1 Zeile} \\ &= -(1-\gamma)\det \begin{pmatrix} 0 & 0 & \cdots & 0 & -\gamma & 1 \\ 1-\gamma & 1 & \cdots & \cdots & 1 & 0 \\ 0 & 1-\gamma & \ddots & & \vdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 1-\gamma & 1 & 0 \\ 1 & \cdots & \cdots & 1 & 0 & 0 \end{pmatrix} \\ &\text{nach IV} = D_N(\gamma) \end{aligned} \tag{5.4}$$

$$+ \det \begin{pmatrix} 0 & 0 & \cdots & \cdots & 0 & 1 \\ 1-\gamma & 1 & \cdots & \cdots & 1 & 0 \\ 0 & 1-\gamma & \ddots & & \vdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 1-\gamma & 1 & \vdots \\ 1 & \cdots & \cdots & 1 & 1 & 0 \end{pmatrix} \tag{5.5}$$

$$\begin{aligned} &\text{entwickeln nach der letzten Zeile} \\ &= (\gamma-1)D_N(\gamma) + (-1)^N \det \begin{pmatrix} 1-\gamma & 1 & \cdots & \cdots & 1 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1-\gamma & 1 \\ 1 & \cdots & \cdots & \cdots & 1 \end{pmatrix} \end{aligned} \tag{5.6}$$

$$= (\gamma-1)D_N(\gamma) + (-1)^N \det \begin{pmatrix} -\gamma & 0 & \cdots & \cdots & 0 \\ -1 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -1 & \cdots & -1 & -\gamma & 0 \\ 1 & \cdots & \cdots & \cdots & 1 \end{pmatrix} \tag{5.7}$$

Die letzte Matrix ist eine Diagonalmatrix. Damit ist diese Determinante gleich dem

Produkt ihrer Diagonalelemente:

$$
\begin{aligned}
(5.7) =&(\gamma-1)D_N(\gamma)+(-1)^N(-\gamma)^{N-1} && (5.8)\\
=&(\gamma-1)D_N(\gamma)+(-1)^N(-1)^{N-1}\gamma^{N-1} && (5.9)\\
=&(\gamma-1)D_N(\gamma)+(-1)^{2N-1}\gamma^{N-1} && (5.10)\\
=&(\gamma-1)D_N(\gamma)-\gamma^{N-1} && (5.11)
\end{aligned}
$$

Dies ist die Behauptung. ■

Im nächsten Schritt muss man jetzt zeigen, dass $D_N(\gamma)$ für $\gamma > 1$ stets ungleich Null ist. Hierzu löst man die Rekursions in (5.1) auf und erhält dann

$$D_N(\gamma) = -\sum_{k=0}^{N-2}(\gamma-1)^k\gamma^{N-k-2} \tag{5.12}$$

Mit Hilfe von MATHEMATICA (siehe Abbildung 5.1) kann man (5.12) noch weiter

```
In[1]:= Deter[N_, γ_] := -Sum[(γ - 1)^k γ^(N-k-2), {k, 0, N-2}]

In[2]:= FullSimplify[PowerExpand[Deter[n, γ]]]

Out[2]= (-1 + γ)^(-1+n) - γ^(-1+n)
```

Abbildung 5.1: Vereinfachen von D_N mit MATHEMATICA.

vereinfachen und erhält schließlich

$$D_N(\gamma) = (\gamma-1)^{N-1} - \gamma^{N-1} \tag{5.13}$$

Sowohl aus (5.13) als auch aus (5.12) kann man nun leicht sehen, dass für alle $\gamma > 1$ gilt: $D_N(\gamma) \neq 0$.

Damit ist Punkt 1 abgehandelt und es bleibt nun noch zu zeigen, dass diese Basis zulässig ist. Dies konnte im Rahmen dieser Arbeit leider nicht mehr gezeigt werden. Ebenso ist der dritte Punkt noch offen. Als Anregung für weitere Untersuchungen können diese Überlegungen jedoch dienen.

6 Zusammenfassung

In dieser Arbeit wurden Kontrollsysteme untersucht, die modellprädiktiv mit variablem Kontrollhorizont geregelt werden. Hierzu wurden zu Beginn die Grundlagen für die Betrachtung von Kontrollsystemen geschaffen. Dabei wurde das Konzept von Ljapunov-Funktionen und deren Zusammenhang mit der Stabilität von Differentialgleichungen dargelegt. Im Rahmen des Abschnittes über Optimale Steuerung wurde das Bellman'sche Optimalitätsprinzip vorgestellt und daraus das für das Hauptresultat wichtige Prinzip der relaxierten dynamischen Programmierung abgeleitet. Im Anschluss daran wurde das Konzept der Modellprädiktiven Regelung erläutert und auf die Schwierigkeit bei der Wahl des Optimierungshorizontes hingewiesen. Deshalb wurde ein Ansatz, der auf Ergebnissen der relaxierten dynamischen Programmierung basiert, dargelegt, mit dessen Hilfe es möglich ist zu entscheiden, ob ein System mit den gewählten Parametern stabil geregelt wird oder nicht. Es wurde der Beweis geführt, dass ein System stabil geregelt wird, wenn ein zugehöriges lineares Programm einen Zielfunktionswert aus einem bestimmten Intervall besitzt.
Da nun die Notwendigkeit bestand lineare Programme zu lösen, wurde im nächsten Kapitel die Simplex-Methode beschrieben. Der Simplex war das erste algorithmisch einsetzbare Verfahren mit dem in großem Stil lineare Programme gelöst werden konnten. Die Idee des Simplex basiert auf Ergebnissen aus der Polyedertheorie. Der Algorithmus springt unter steter Verbesserung des Zielfunktionswertes von Ecke zu Ecke im durch die Nebenbedingungen beschriebenen Polyeder bis das Optimum erreicht wurde oder festgestellt wurde, dass das Problem unbeschränkt ist.
Im dritten Teil wurde nun untersucht, wie sich der Kontrollhorizont und die Parameter der $\mathcal{KL}_0$-Funktion auf die Stabilität des Systems auswirken. Zu diesem Zweck wurde der Simplex in einem Computeralgebrasystem implementiert, um das lineare Programm, mit dem bestimmt wird, ob das System stabil ist oder nicht, nicht nur numerisch sondern auch symbolisch lösen zu können. Hierbei hat es sich gezeigt, dass die Zielfunktionswerte dieses linearen Programmes für symmetrische Kontrollhorizonte gleich sind. Darüber hinaus wurde untersucht für welche Parameter der $\mathcal{KL}_0$-Funktion die Zielfunktionswerte größer als Null sind. Diese so genannten Stabilitätsbereiche weisen zwei Beziehungen zueinander auf. Zum einen ist der Stabilitätsbereich für einen Kontrollhorizont Teilmenge des Stabilitätsbereiches mit einem größeren Kontrollhorizont, zumindest solange beide Kontrollhorizonte kleiner als die Hälfte des Optimierungshorizontes sind. Zum anderen ist der Stabilitätsbereich für einen festen Optimierungshorizont Teilmenge der Stabilitätsbereiche mit größeren Optimierungshorizonten.

Im letzten Teil wurden schließlich einige Anregungen gegeben, wie man die Beobachtungen aus dem vorherigen Kapitel beweisen könnte. Es wurde gezeigt, dass die Koeffizientenmatrix der Nebenbedingungen für den Kontrollhorizont $m = N - 1$ eine Basis ist. Dies könnte als Grundlage für einen Beweis dienen, dass die Zielfunktionswerte für symmetrisch liegende Kontrollhorizonte gleich sind.

A Die Hilfsfunktionen

A.1 Die Funktion Nebenbedingungen

```
In[1]:= Nebenbedingungen[Nn_, m_, beta_, symbolisch_] := Module[{A, b, c, B, Be, β, s},
          If[beta == 1,
           β[γ_, r_, n_] := { γ r   n == 1
                            { 0    n > 1 ;
           Be_Nnn_[r_] := Σ_{n=0}^{Nnn-1} β[γ, r, n];,
           β[Ce_, σ_, r_, n_] := Ce σ^n r;
           Be_Nnn_[r_] := Σ_{n=0}^{Nnn-1} β[Ce, σ, r, n];];

          nb42[Nnn_, mm_, j_, sch_] := -Σ_{n=0}^{j-1} λ_{n+m} - Be_{Nnn-j}[λ_{j+mm}] + ν + s_sch == 0;

          nb41[Nnn_, k_, sch_] := Σ_{n=k}^{Nnn-1} λ_n - Be_{Nnn-k}[λ_k] + s_sch == 0;

          nb53[mm_] := Σ_{n=0}^{mm-1} λ_n == 1;

          {b, A} = Normal[CoefficientArrays[Flatten[
              {Table[nb42[Nn, m, j, j + 1], {j, 0, Nn - m - 1}],
               Table[nb41[Nn, i, Nn - m + i + 1], {i, 0, Nn - 2}], nb53[m]}]
             , Flatten[{Table[λ_i, {i, 0, Nn - 1}], ν, Table[s_i, {i, 2 Nn - m - 1}]}]]];
          b = {-b}^T;
          c = Flatten[{Table[-1, {i, 1, Nn}], 1, Table[0, {i, 2 Nn - m - 1}]}];
          B = Flatten[{1, Table[i, {i, Nn + 2, 3 Nn - m}]}];
          If[symbolisch ≠ True,
           b = Append[#, 0] & /@ b;
          ];
          Return[{A, b, c, B}];
         ];
```

Programm A.1: Die Funktion `Nebenbedingungen` erstellt die Inputdaten für die Funktion `SymSimplex`. Es kann zwischen zwei $\mathcal{KL}_0$-Funktionen gewählt werden und ob die Ausgabe für die symbolische oder die numerische Variante des Simplex verwendet werden soll.

Zwei typische Aufrufe der Funktion `Nebenbedingungen` sind in Abbildung A.1 abgebildet. Im ersten Aufruf wird als Optimierungshorizont $N = 4$ gewählt, als Kontrollhorizont $m = 1$ und die $\mathcal{KL}_0$-Funktion $\beta(r,t) = C\sigma^t r$. Im zweiten Fall ist der Optimierungshorizont ebenfalls 4, der Kontrollhorizont aber 3. Als $\mathcal{KL}_0$-Funktion wird hier (4.17) verwendet. In beiden Fällen werden die Inputdaten für den symbolischen Simplex erzeugt.

```
In[5]:= {A41, b41, c41, B41} = Nebenbedingungen[4, 1, 2, True]

Out[5]= {{{0, -Ce - Ce σ - Ce σ^2 - Ce σ^3, 0, 0, 1, 1, 0, 0, 0, 0, 0},
    {0, -1, -Ce - Ce σ - Ce σ^2, 0, 1, 0, 1, 0, 0, 0, 0},
    {0, -1, -1, -Ce - Ce σ, 1, 0, 0, 1, 0, 0, 0},
    {1 - Ce - Ce σ - Ce σ^2 - Ce σ^3, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0},
    {0, 1 - Ce - Ce σ - Ce σ^2, 1, 1, 0, 0, 0, 0, 0, 1, 0},
    {0, 0, 1 - Ce - Ce σ, 1, 0, 0, 0, 0, 0, 0, 1},
    {1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}}, {{0}, {0}, {0}, {0}, {0}, {0}, {1}},
   {-1, -1, -1, -1, 1, 0, 0, 0, 0, 0, 0}, {1, 6, 7, 8, 9, 10, 11}}

In[6]:= {A43, b43, c43, B43} = Nebenbedingungen[4, 3, 1, True]

Out[6]= {{{0, 0, 0, -γ, 1, 1, 0, 0, 0}, {1 - γ, 1, 1, 1, 0, 0, 1, 0, 0},
    {0, 1 - γ, 1, 1, 0, 0, 0, 1, 0}, {0, 0, 1 - γ, 1, 0, 0, 0, 0, 1}, {1, 1, 1, 0, 0, 0, 0, 0, 0}},
   {{0}, {0}, {0}, {0}, {1}}, {-1, -1, -1, -1, 1, 0, 0, 0, 0}, {1, 6, 7, 8, 9}}
```

Abbildung A.1: Zwei Aufrufe der Funktion `Nebenbedingungen`. Im ersten Fall mit der $\mathcal{KL}_0$-Funktion $\beta(r,t) = C\sigma^t r$. Im zweiten Fall mit $\beta(r,t) = \gamma r$.

A.2 Die Funktion alpham

```
function [alpha,Nall,Nab,mab] = alpham(maxN,varargin)
%   Argumente:
%   maxN:       maximaler Optimierungshorizont N
%   varargin: C und sigma
%         oder: gamma
%
%   (C) Harald Voit 2007

  if (nargin == 2)
      gamma = varargin{1};
  else
      if (nargin == 3)
          C = varargin{1};
          sigma = varargin{2};
      else
          fprintf('Anzahl Parameter falsch!\n');
          return;
      end;
  end;
```

```
alpha=zeros(maxN,maxN);
for N=2:maxN
    for m=1:(N-1)
        if (nargin == 3)
            alpha(N,m)=mpcalpha_Csigma(C,sigma,N,m,1,0);
        end;
        if (nargin == 2)
            alpha(N,m)=mpcalpha_gamma(gamma,N,m,1,0);
        end;
    end;
end;

alphaold=alpha;

figure(maxN)
for i=2:maxN
    for j=i:maxN
        alpha(i,j)=alpha(i,i-1);
    end;
end;
mm=ones(maxN,maxN-1);
for i=2:maxN
    for j=i:maxN-1
        mm(i,j)=i;
        mm(j,i)=i;
    end;
end;
mm(maxN,:)=1:maxN-1;

plot(mm,alpha(2:maxN,1:maxN)',[0 maxN],[0 0],[0 maxN],[1 1]);
xlabel('Kontrollhorizont m');
ylabel('\alpha');
title(strcat('Zielfunktionswerte \alpha für N=2,...,',int2str(maxN),...
    ' mit C=',num2str(C),', \sigma=',num2str(sigma)));
axis([0 maxN min(min(alpha)) 1.2]);
print('-depsc',strcat('alpham_alle_',int2str(maxN),'_',num2str(C),...
    '_',num2str(sigma),'.eps'));

alpha=alphaold;

Nall=NaN;
cum=sum(alpha>0,2);
for i=2:maxN
    if (i==cum(i)+1)
        Nall=min(Nall,i);
    end;
end;
[m,Nab]=max(cum>0);
[m,mab]=max(alpha(Nab,:)>0);
```

```
if (nargin == 3)
    sprintf('%4.3f\t%4.3f\t%d,%d\t%d', C, sigma, Nab, mab, Nall)
end;
if (nargin == 2)
    sprintf('%3.1f\t%d,%d\t%d', gamma, Nab, mab, Nall)
end;
```

Programm A.2: alpham(N,C,sigma). Berechnet alle α bis zum gegebenen N und bestimmt, ab wann der erste stabile Horizont auftritt und ab welchem N alle α's größer als Null sind.

Ein typischer Aufruf der Funktion `alpham` ist in 4.6 angegeben.

B Zielfunktionswerte für $\beta(r,t) = C\sigma^t r$

N=2

$$-C(\sigma+1)(\sigma C + C - 2) \tag{B.1}$$

N=3, m=1 und 2

$$-\frac{C\left(\sigma^2+\sigma+1\right)\left(\left(\sigma^3+2\sigma^2+2\sigma+1\right)C^2-\left(\sigma^2+3\sigma+3\right)C+1\right)}{C\left(\sigma^2+2\sigma+2\right)-1} \tag{B.2}$$

N=4, m=1 und 3

$$\frac{-C(\sigma^3+\sigma^2+1)\Big((\sigma+1)^2(\sigma^4+\sigma^3+2\sigma^2+\sigma+1)C^3-(\sigma+1)(\sigma^2+\sigma+2)^2C^2\Big)}{} + \frac{-C(\sigma^3+\sigma^2+\sigma+1)\Big((\sigma^3+2\sigma^2+3\sigma+3)C-1\Big)}{\left(\sigma^5+3\sigma^4+6\sigma^3+7\sigma^2+6\sigma+3\right)C^2-\left(\sigma^3+2\sigma^2+3\sigma+3\right)C+1} \tag{B.3}$$

N=4, m=2

$$\frac{-C^2\left(\sigma^5+2\sigma^4+3\sigma^3+3\sigma^2+2\sigma+1\right)\Big(\left(\sigma^5+2\sigma^4+3\sigma^3+3\sigma^2+2\sigma+1\right)C^2\Big)}{} + \frac{-C^2\left(\sigma^5+2\sigma^4+3\sigma^3+3\sigma^2+2\sigma+1\right)\Big(2\left(\sigma^3+2\sigma^2+2\sigma+2\right)C+2\Big)}{\left(C\left(\sigma^3+2\sigma^2+2\sigma+2\right)-1\right)^2} \tag{B.4}$$

Weitere Zielfunktionswerte für $N = 5, \ldots, 8$ sind auf der beiliegenden CD enthalten (siehe Anhang C).

C Inhalt der CD

Auf der beiliegenden CD sind in drei Verzeichnissen alle verwendeten Programme und die Programmaufrufe für alle Ausgaben und Bilder enthalten, sofern diese nicht ohnehin im Text vorkommen. Außerdem sind im Verzeichniss `Zielfunktionswerte` noch weitere Zielfunktionswerte, zusätzlich zu den in Anhang B angegebenen, enthalten. Im Verzeichnis `Diplomarbeit` ist eine PDF-Version der vorliegenden Arbeit enthalten.

Verzeichnis	enthaltene Dateien	Beschreibung
`Diplomarbeit`	Diplomarbeit.pdf	PDF-Version der vorliegenden Arbeit
`Programme`	Programme.nb	Quelltexte der verwendeten Mathematica-Programme `SymSimplex` und `Nebenbedingungen`.
	Programme.pdf	Die Datei Programme.nb als PDF-Datei.
	Berechnungen.nb	Enthält alle Programmaufrufe in Mathematica, die im Text vorkommen oder mit denen Bilder erzeugt wurden (braucht Programme.nb).
	Berechnungen.pdf	Die Datei Berechnungen.nb als PDF-Datei.
	alpham.m	Matlab-Programm zur Erzeugung von Serien von Zielfunktionswerten (braucht mpcalpha_Csigma.m)
	mpcalpha_Csigma.m	Matlab-Programm von L. Grüne (siehe [14])
	mpcalpha_cn.m	Matlab-Programm von L. Grüne (siehe [14])
`Zielfunkt-ionswerte`	ZF_N=2-9.nb	Enthält die Zielfunktionswerte für $\beta(r,t) = C\sigma^t r$ für $N = 2, \ldots, 9$
	ZF_N=2-9.pdf	Die Datei ZF_N=2-9.nb als PDF-Datei.

Literaturverzeichnis

[1] Alevras, D. und M. W. Padberg: *Linear Optimization and Extensions: Problems and Solutions*. Universitext. Springer, Berlin, 1. Auflage, 2001.

[2] Allgöwer, F., T.A. Badgwell, J.S. Qin, J.B. Rawlings und S.J. Wright: *Nonlinear Predictive Control and Moving Horizon Estimation-An Introductory Overview*. In: Frank, P. M. (Herausgeber): *Advances in Control. Highlights of ECC '99*. Springer-Verlag London Ltd, 1999.

[3] Bastian, M.: *Lineare Optimierung großer Systeme*. Verl.-Gruppe Athenäum, Hain, Scriptor, Hanstein, Königstein/Ts., 1980.

[4] Bellman, R. E.: *Dynamic programming*. A Rand Corporation research study. Princeton Univ.Pr., Princeton, N.J., 6. Auflage, 1972.

[5] Bland, R. G.: *New finite pivoting rules for the simplex method*. Mathematics of Operations Research, 2(2):103–107, 1976.

[6] Brockett, R. W.: *Asymptotic Stability and Feedback Stabilization*. In: R. W. Brockett, R. S. Millman und H. J. Sussmann (Herausgeber): *Differential Geometric Control Theory*, Seiten 181–191. Birkhäuser, Boston, 1983.

[7] Dantzig, G. B. und M. N. Thapa: *Linear Programming 1: Introduction*. Springer Series in Operations Research. Springer, Berlin, 1. Auflage, 1997.

[8] Dantzig, G. B. und M. N. Thapa: *Linear Programming 2: Theory and Extensions*. Springer Series in Operations Research. Springer, Berlin, 1. Auflage, 2003.

[9] Forster, O.: *Analysis 2, Differentialrechnung im $\mathbb{R}^n$, gewöhnliche Differentialgleichungen*. Vieweg Studium : Grundkurs Mathematik. 7. Auflage, 2006.

[10] Grötschel, M.: *Skript zur Vorlesung Lineare Optimierung*. Technische Universität Berlin, 2004.

[11] Grüne, L.: *Optimization based stabilization of nonlinear control systems*. Large-Scale Scientific Computations (LSSC07).

[12] Grüne, L.: *Skript zur Vorlesung Mathematische Kontrolltheorie II.* Universität Bayreuth, 2006.

[13] Grüne, L.: *Skript zur Vorlesung Numerik dynamischer Systeme.* Universität Bayreuth, 2005.

[14] Grüne, L.: *Computing stability and performance bounds for unconstrained NMPC schemes.* In: *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007.

[15] Karmarkar, N.: *A new polynomial-time algorithm for linear programming.* Combinatorica, 4(4):373–395, 1984.

[16] Keerthi, S.S. und E.G. Gilbert: *Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations.* Journal of Optimization Theory and Applications, 57(2):265–293, 1988.

[17] Khachiyan, L.G.: *A polynomial algorithm in linear programming.* Doklady Akademiia Nauk SSSR, 244:1093–1096, 1979.

[18] Klee, V. und G. Minty: *How good is the simplex algorithm?* In: Shisha, O. (Herausgeber): *Inequalities-SOVIET MATHEMATICS : Doklady. - Providence, RIIII*, Seiten 159–175. Academic Press, New York, 1972.

[19] Orchard-Hays, W.: *Background, development, and extensions of the revised simplex method.* RAND Research Memorandum, RM-1433, 1954.

[20] Propoi, A.I.: *Use of linear programming methods for synthesizing sampled-data automatic systems.* Automation and Remote Control, 24(7):837–844, 1963.

[21] Qin, S.J. und T.A. Badgwell: *An Overview of Nonlinear Model Predictive Control Applications.* In: Allgöwer, F. und A. Zheng (Herausgeber): *Nonlinear Model Predictive Control (Progress in Systems and Control Theory).* Birkhäuser Verlag AG, 2007.

[22] Rambau, J.: *Skript zur Vorlesung Lineare Optimierung.* Universität Bayreuth, 2005.

[23] Rambau, J.: *Skritpt zur Blockvorlesung Diskrete Optimierung in Transport, Logistik und Verkehr*, 2006. Universität Bayreuth.

[24] Schrijver, A.: *Theory of linear and integer programming.* Wiley, Chichester, 1999.

[25] Sontag, E.D.: *Mathematical Control Theory: Deterministic Finite Dimensional Systems*. Springer-Verlag Berlin Heidelberg New York, 1998.

[26] Stigler, G. J.: *The Cost of Subsistence*. Journal of Farm Economics, 27(2):303–314, 1945.

[27] Vanderbei, R. J.: *Linear Programming. Foundations and Extensions*. International Series in Operations Research & Management Science. Springer Netherlands, 2. Auflage, 2001.

Erklärung

Hiermit erkläre ich, daß ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.
Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

Bayreuth, den 21. Januar 2008

Harald Voit