



Universität Bremen

Zentrum für Technomathematik

# Optimale Steuerung einer Laufkatze im Hochregallager

Bachelorarbeit

vorgelegt von

Tobias Riemenschneider

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

09. August 2011

# Inhaltsverzeichnis

Abbildungsverzeichnis	iii
Tabellenverzeichnis	iv
<b>1 Einleitung</b>	<b>1</b>
<b>2 Modellierung der Bahntrajektorien</b>	<b>3</b>
<b>3 Grundlagen der Optimierung und optimalen Steuerung</b>	<b>6</b>
3.1 Nichtlineare Optimierung . . . . .	6
3.1.1 Die KKT-Bedingungen . . . . .	7
3.1.2 Bedingungen zweiter Ordnung . . . . .	9
3.1.3 Algorithmische Vorgehensweise . . . . .	9
3.1.3.1 Ein Spezialfall . . . . .	9
3.1.3.2 Innere-Punkte-Verfahren . . . . .	10
3.1.3.3 SQP-Verfahren . . . . .	11
3.2 Optimale Steuerung . . . . .	11
3.2.1 Grundlegende Definitionen . . . . .	11
3.2.2 Lösungsansätze . . . . .	12
<b>4 Optimale Steuerung der Laufkatze</b>	<b>15</b>
4.1 Das dynamische Verhalten . . . . .	15
4.2 Anfangs- und Endbedingungen . . . . .	16
4.3 Zustands- und Steuerbeschränkungen . . . . .	17
4.4 Zielfunktionale . . . . .	18
4.4.1 Minimierung der Verfahrzeit . . . . .	18

---

4.4.2	Minimierung des Rucks	18
4.4.3	Minimierung des Schwingungsverhaltens	19
4.4.4	Kombinationen	19
<b>5</b>	<b>Numerische Ergebnisse</b>	<b>21</b>
5.1	Numerische Umsetzung	21
5.2	Auswertung	22
5.2.1	Wahl und Einfluss der Gewichte	22
5.2.2	Plausibilitätstest	24
5.2.2.1	Durchführung	24
5.2.2.2	Ergebnisse	25
5.2.3	Komplexere Trajektorie und Notstopp	29
5.2.4	Maximaler Ausschlagwinkel	31
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>32</b>
<b>A</b>	<b>Nachweise zum Plausibilitätstest</b>	<b>33</b>
<b>B</b>	<b>Programm</b>	<b>35</b>
	<b>Literaturverzeichnis</b>	<b>41</b>

# Abbildungsverzeichnis

2.1	Skizze zur Modellherleitung. . . . .	4
5.1	Steuerungen und Zustände für unterschiedliche Gewichtungen der Zeit und des Rucks. . . . .	23
5.2	Test-Nr. 2.1 und 2.3. Seilsteuerung und Zustände verändern sich nicht bei einer Verschiebung. . . . .	27
5.3	Test-Nr. 2.2. Im Gegensatz zu Test-Nr. 2.1 wird hier die Last 4 Meter hochgezogen. . . . .	27
5.4	Test-Nr. 2.4. Hier wird im Vergleich zu Test 2.1 (Abbildung 5.2) die doppelte Distanz zurückgelegt. . . . .	27
5.5	Seitenansicht für verschiedene Prioritäten und zwei unterschiedliche Strecken auf gleicher Höhe. . . . .	28
5.6	Seitenansicht für unterschiedliche Distanzen am Beispiel der schnellen Bahn. . . . .	28
5.7	Seitenansicht der unterschiedlich gewichteten Bahntrajektorien für eine weite, schräge Strecke. . . . .	29
5.8	Steuerungen und Zustände für eine weite, schräge Strecke von $s_0 = 0$ , $l_0 = 10$ nach $s_f = 20$ , $l_f = 2$ . . . . .	30
5.9	Seitenansicht für einen Notstopp der schnellen Bahn für eine weite, schräge Strecke. . . . .	30
5.10	Histogramm zum maximalen Winkel für 100 zufällige Fahrten. . . . .	31
A.1	Plausibilitätstest: Symmetrie von Test-Nr. 3.1 und 3.2. . . . .	34
A.2	Plausibilitätstest: Die Bahnen sind unabhängig von Verschiebungen. . . . .	34

# Tabellenverzeichnis

5.1	Verwendete Steuer- und Zustandsbeschränkungen . . . . .	21
5.2	Gemittelte Werte für die Funktionale für 100 zufällige Testbahnen. . .	23
5.3	Gewichtungsfaktoren für unterschiedliche Bahnverläufe. . . . .	24
5.4	Plausibilitätstest: Stillstand und vertikale Bewegungen. . . . .	25
5.5	Plausibilitätstest: Horizontale Bewegungen. . . . .	26
A.1	Ergebnisse des Plausibilitätstests: Fahrzeiten für die vier unterschiedlichen Bahnkurven. . . . .	33

# Kapitel 1

## Einleitung

Bei Kränen bezeichnet man den Wagen, der entlang des Kranauslegers das Hubseil positioniert, als *Laufkatze*. Anwendung finden Laufkatzen aber nicht nur auf Baustellen oder in Häfen, sondern auch in Hochregallagern. Sie sind in modernen Lagerstätten oberhalb der Regalfächer entlang der Regalgasse an Schienen angebracht und ersetzen die klassischen bodengeführten Regalbediengeräte.

Der Einsatz von Laufkatzen im Hochregallager hat zahlreiche Vorteile. Während zum Beispiel bei den klassischen bodengeführten Regalbediengeräten aufwendige Stahlkonstruktionen nötig sind, um die höheren Regalfächer zu erreichen, genügen bei der Laufkatze Stahlseile zum Absenken. Dadurch werden sowohl Materialkosten als auch Energiekosten, aufgrund des geringeren Gewichts, gesenkt.

Da sie nicht flurgebunden sind, können mehrere Laufkatzen in der gleichen Regalgasse agieren und so die Effizienz steigern. Ein weiterer Vorteil besteht darin, dass die Last über Durchgänge oder andere Hindernisse problemlos gehoben werden kann.

Der Nachteil des Systems aus Laufkatze, Seil und Lastaufnahmemittel ist, dass beim Bewegen der Laufkatze die Last in Schwingung gerät. Solange das System schwingt, kann keine Be- oder Entladung stattfinden. Regler können die Last zur Ruhe bringen, benötigen dafür jedoch viel Zeit. Darüber hinaus kann ein zu starkes Ausschwenken zu Sicherheitsproblemen führen oder beispielsweise beim Transportieren von Flüssigkeiten unerwünscht sein. Die genannten Nachteile lassen sich aber durch eine geschickte Steuerung der Laufkatze beheben. Diese Steuerungen zu bestimmen, ist Thema dieser Bachelorarbeit.

In Kapitel 2 wird dazu ein Modell zur Beschreibung der Dynamik des Systems hergeleitet. Das dabei aufgestellte Differentialgleichungssystem besitzt zwei Freiheitsgrade, die es zu optimieren gilt. Kapitel 3 liefert die nötigen Grundlagen der

Optimierung sowie der optimalen Steuerung. In Kapitel 4 wird das optimale Steuerungsproblem der Laufkatze formuliert und schließlich in Kapitel 5 numerisch ausgewertet.

## Kapitel 2

# Modellierung der Bahntrajektorien

Für die Herleitung des Modells wird zunächst eine stehende Laufkatze mit konstanter Seillänge  $l$  betrachtet, dessen Last um den Winkel  $\varphi$  ausgelenkt sei. Die hieraus resultierende Bewegungsgleichung entspricht der eines Pendels und wird anschließend um die Bewegung der Laufkatze und um eine zeitlich veränderliche Seillänge erweitert. Die Herleitung ist an Knauer [1] angelehnt.

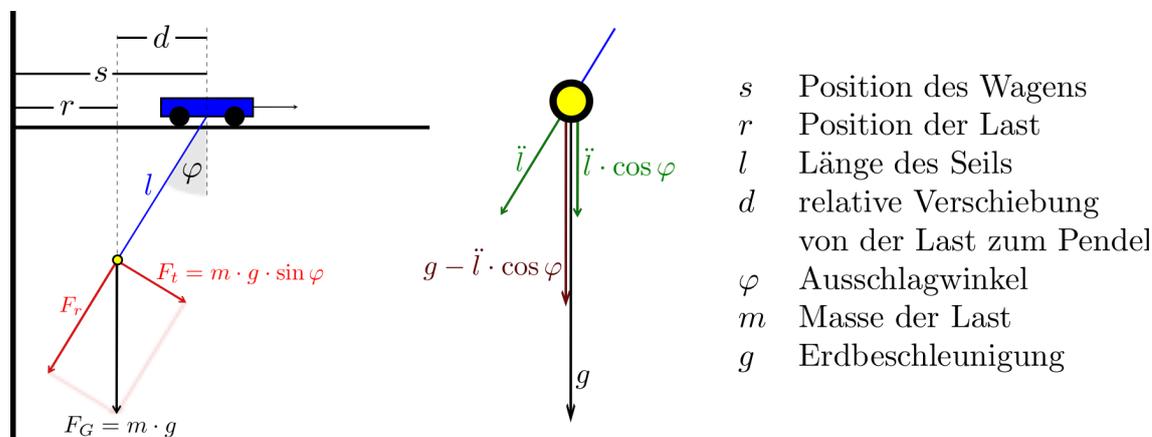
Im Folgenden wird das Lastaufnahmemittel, einschließlich der zu befördernden Nutzlast, als *Last* bezeichnet. Der Einfachheit halber wird angenommen, dass die Masse  $m$  der Last in einem Punkt konzentriert ist, also keine räumliche Ausdehnung besitzt. Darüber hinaus sei die Laufkatze nicht über vier gleich lange Seile mit dem Lastaufnahmemittel verbunden, sondern nur über ein Seil, welches von der Seiltrommel der Laufkatze zu der punktförmigen Last führt.

Da die Masse eines Stahlseils im Vergleich zu der Masse der Last sehr klein ist, kann das Seil als masselos angesehen werden. Vernachlässigt man darüber hinaus sämtliche Reibungskräfte, lässt sich das System aus stehender Laufkatze, Seil und Last als mathematisches Pendel auffassen, wobei der stehende Wagen als Aufhängungspunkt fungiert.

### **Definition 2.1** (*Mathematisches Pendel*)

Bei einem mathematischen Pendel besitzt ein Massepunkt immer die feste Entfernung  $l$  vom Aufhängungspunkt und bewegt sich frei unter dem Einfluss der Schwerkraft.

Die Bewegungsgleichung basiert auf den Newton'schen Axiomen. Man kann die Bewegung des Pendels erklären, indem man die an der Masse angreifende Schwerkraft  $F = m \cdot g$  in eine radiale und eine tangentielle Komponente zerlegt, wie in Abbildung 2.1 gezeigt. Dabei ist  $g \approx 9,81 \frac{m}{s^2}$  die Erdbeschleunigung. Die radiale Komponente



**Abbildung 2.1:** Skizze zur Modellherleitung, links für feste und in der Mitte für variable Seillänge. Die Abbildungen sind nicht maßstabsgetreu, insbesondere ist der reale Ausschlagwinkel kleiner.

ruft im gespannten Seil eine gleich große, entgegengerichtete Kraft hervor und trägt deshalb nicht zur Beschleunigung bei, wohingegen die tangentielle Komponente

$$F_t = m \cdot g \cdot \sin \varphi$$

eine Tangentialbeschleunigung  $a_t = g \cdot \sin \varphi$  bewirkt. Diese kann für kleine Winkel und ein langes Seil näherungsweise als waagerechte Beschleunigung angesehen werden. Mit der Bezeichnung  $r$  für die waagerechte Position der Last gilt

$$\ddot{r} = a_t = g \cdot \sin \varphi. \quad (2.1)$$

Drückt man  $\sin \varphi$  durch die relative waagerechte Verschiebung  $d$  der Last zur Laufkatze und der Seillänge  $l$  aus, folgt aus (2.1)

$$\ddot{r} = g \cdot \frac{d}{l}. \quad (2.2)$$

Um die Bewegung der Laufkatze mit in das Modell einzubeziehen, wird die absolute Position  $s = r + d$  betrachtet. Durch zweimalige Differentiation erhält man die Beschleunigung  $\ddot{s} = \ddot{r} + \ddot{d}$  der Laufkatze. Umstellen nach  $\ddot{r}$  und einsetzen in (2.2) liefert die Differentialgleichung

$$\ddot{s} - \ddot{d} = g \cdot \frac{d}{l} \quad (2.3)$$

zur Beschreibung der Bahntrajektorien bei konstanter Seillänge.

Für zeitlich veränderliche Seillängen muss berücksichtigt werden, dass der vertikale Teil der Seilbeschleunigung  $\ddot{l} \cos \varphi$  der Erdbeschleunigung entgegenwirkt, siehe Abbildung 2.1. Für kleine Winkel gilt in guter Näherung  $\ddot{l} \cos \varphi \approx \ddot{l}$ .<sup>1</sup> Somit ist in (2.3)  $g$  durch  $(g - \ddot{l})$  zu ersetzen. Das Modell zur Beschreibung der Bahntrajektorien lautet also

$$\ddot{d} = \ddot{s} - (g - \ddot{l}) \cdot \frac{d}{l}. \quad (2.4)$$

Eine alternative Herleitung von (2.4) findet man bei Pesch [2].

Die Laufkatze kann über die Beschleunigung des Wagens  $u_1 := \ddot{s}$  und die Seilbeschleunigung  $u_2 := \ddot{l}$  gesteuert werden. Abhängig von diesen Steuergrößen ergibt sich das dynamische Verhalten

$$\begin{aligned} \ddot{s} &= u_1 \\ \ddot{l} &= u_2 \\ \ddot{d} &= u_1 - (g - u_2) \cdot \frac{d}{l}. \end{aligned} \quad (2.5)$$

---

<sup>1</sup>In der numerischen Auswertung 5.2.4 wird  $13^\circ$  als maximaler Winkel angegeben. Für diesen Winkel weicht die Näherung  $\ddot{l}$  nur um 2,6% von  $\ddot{l} \cos(\varphi)$  ab.

# Kapitel 3

## Grundlagen der Optimierung und optimalen Steuerung

### 3.1 Nichtlineare Optimierung

Die Optimierung ist ein wichtiger Zweig der angewandten Mathematik und kommt überall dort zum Einsatz, wo Parameter  $x$  so bestimmt werden sollen, dass eine Zielfunktion  $f$  möglichst groß oder möglichst klein wird. Typische Ziele sind etwa die Minimierung der Kosten, der Zeit oder des Weges. Konkrete Problemstellungen aus der Wirtschaft, Logistik, Biologie und weiteren Bereichen finden sich zum Beispiel bei Allaire [4].

Die in Abschnitt 3.2 vorgestellte Theorie der optimalen Steuerung lässt sich durch Diskretisierung auf ein nichtlineares Optimierungsproblem reduzieren. Aus diesem Grund sollen in diesem Abschnitt die grundlegenden Definitionen sowie die wichtigsten Sätze der nichtlinearen Optimierung angegeben werden. Eine kurze Skizzierung numerischer Methoden zum Lösen von nichtlinearen Optimierungsproblemen schließt diesen Abschnitt ab. Das in Kapitel 5 verwendete numerische Verfahren zur optimalen Steuerung der Laufkatze baut auf diesen Grundlagen auf.

Zu der Theorie der Optimierung gibt es zahlreiche Fachbücher, darunter Jungnickel [3], an dem dieser Abschnitt orientiert ist. Die Bedingungen zweiter Ordnung sind an Knauer [1] angelehnt.

Da die Maximierung einer Funktion  $f$  äquivalent zu der Minimierung von  $-f$  ist, genügt es Minimierungsprobleme zu betrachten.

**Definition 3.1 (Nichtlineares Optimierungsproblem)**

Das allgemeine nichtlineare Optimierungsproblem hat die Form

$$\begin{aligned} \min \quad & f(x) \\ \text{unter} \quad & g_i(x) \leq 0 \quad \text{für } i = 1, \dots, p \\ & h_j(x) = 0 \quad \text{für } j = 1, \dots, m \\ & x \in \mathbb{R}^n. \end{aligned} \tag{3.1}$$

Dabei bilden alle vorkommenden Funktionen von  $\mathbb{R}^n$  nach  $\mathbb{R}$  ab. Die Funktionen  $g_1, \dots, g_p$  heißen Ungleichungsrestriktionen und  $h_1, \dots, h_m$  Gleichungsrestriktionen. Erfüllt  $x$  die Nebenbedingungen, also

$$x \in X := \{x \in \mathbb{R}^n \mid g_i(x) \leq 0 \text{ für } i = 1, \dots, p \text{ und } h_j(x) = 0 \text{ für } j = 1, \dots, m\}$$

so heißt  $x$  zulässige Lösung. Die Menge aller zulässigen Lösungen  $X$  wird Zulässigkeitsbereich genannt.

Ist der Zulässigkeitsbereich der ganze Raum  $\mathbb{R}^n$  heißt das Problem unbeschränkt.

Für die numerische Umsetzung sind quadratische Optimierungsprobleme wichtig.

**Definition 3.2 (Quadratische Optimierungsprobleme)**

Ein quadratisches Optimierungsproblem, oder auch quadratisches Programm (QP) genannt, ist ein Problem der folgenden Form:

$$\begin{aligned} \min \quad & f(x) = x^T Q x + c^T x + \gamma \\ \text{unter} \quad & g(x) := Ax - b \leq 0; \\ & h(x) := Bx - d = 0; \\ & x \in \mathbb{R}^n, \end{aligned} \tag{3.2}$$

wobei  $Q$  eine symmetrische  $(n \times n)$ -Matrix ist und  $A \in \mathbb{R}^{(p,n)}$ ,  $B \in \mathbb{R}^{(m,n)}$ ,  $b \in \mathbb{R}^p$ ,  $d \in \mathbb{R}^m$ ,  $c \in \mathbb{R}^n$  und  $\gamma \in \mathbb{R}$  sind.

**3.1.1 Die KKT-Bedingungen**

Sind die Zielfunktion  $f$  und die Restriktionsfunktionen  $g_1, \dots, g_p$  und  $h_1, \dots, h_m$  differenzierbar, kann man notwendige Optimalitätsbedingungen formulieren. Diese sind nach Karush, Kuhn und Tucker (KKT) benannt. Der Satz setzt die im Folgenden definierte Regularität von  $x$  voraus.

**Definition 3.3 (Linear Independence Constraint Qualification (LICQ))**

Ein zulässiger Punkt  $x \in X$  heißt regulär, wenn die Gradienten

$$\nabla h_j(x), j = 1, \dots, m, \text{ sowie } \nabla g_i, i \in I(x),$$

linear unabhängig sind. Dabei bezeichnet  $I(x) := \{i \in \{1, \dots, p\} | g_i(x) = 0\}$  die Indexmenge der in  $x$  straffen Ungleichungsrestriktionen.

**Satz 3.4 (Notwendige KKT-Bedingungen)**

In Problem (3.1) sei die Zielfunktion  $f$  differenzierbar auf  $\mathbb{R}^n$  und die Restriktionsfunktionen  $g_i$  und  $h_j$  seien stetig differenzierbar auf  $\mathbb{R}^n$ . Ferner sei  $x \in X$  ein regulärer Punkt und  $I$  die Indexmenge der in  $x$  straffen Ungleichungsrestriktionen.

Falls  $x$  ein lokales Minimum für  $f$  ist, gibt es Skalare  $\mu_i$  ( $i = 1, \dots, p$ ) und  $\lambda_j$  ( $j = 1, \dots, m$ ), die die folgenden **KKT-Bedingungen** erfüllen:

$$\nabla f(x) + \sum_{i \in I} \mu_i \nabla g_i(x) + \sum_{j=1}^m \lambda_j \nabla h_j(x) = 0, \quad \mu_i \geq 0 \text{ für } i = 1, \dots, p. \quad (3.3)$$

**Beweis:** Den Beweis findet man zum Beispiel bei Jungnickel [3]. ■

**Definition 3.5**

Die Skalare  $\mu_i$  ( $i = 1, \dots, p$ ) und  $\lambda_j$  ( $j = 1, \dots, m$ ) von Satz 3.4 werden *Lagrange-Multiplikatoren* genannt.

Die für die Optimierung wichtige *Lagrange-Funktion*  $L : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^m \rightarrow \mathbb{R}$  ist gegeben durch

$$L(x, \mu, \lambda) := f(x) + \mu^T g(x) + \lambda^T h(x).$$

Dabei sind  $g = (g_1, \dots, g_p)^T$  und  $h = (h_1, \dots, h_m)^T$  die Restriktionsfunktionen und die Vektoren  $\mu = (\mu_1, \dots, \mu_p)^T$ ,  $\lambda = (\lambda_1, \dots, \lambda_m)^T$  fassen die Lagrange-Multiplikatoren zusammen.

**Bemerkung 3.6**

Mit Definition 3.5 lassen sich die KKT-Bedingungen (3.3) auch kürzer schreiben:

$$\begin{aligned} \nabla_x L(x, \mu, \lambda) &= \nabla f(x) + \mu^T \nabla g(x) + \lambda^T \nabla h(x) = 0, \\ \mu &\geq 0, \\ \mu^T g(x) &= 0. \end{aligned}$$

Die Bedingungen  $\mu \geq 0, \mu^T g(x) = 0$  bedeuten, dass der Lagrange-Multiplikator  $\mu_i$  nur dann ungleich 0 sein kann, wenn  $g_i(x) = 0$  also  $i \in I$  ist.

### 3.1.2 Bedingungen zweiter Ordnung

Wenn zulässige Punkte  $x \in X$  die KKT-Bedingungen für (3.1) erfüllen, heißen sie *kritisch*. Mit Bedingungen zweiter Ordnung können diese kritischen Punkte auf Optimalität untersucht werden. Hierfür wird untersucht, ob die Hesse-Matrix der Lagrange-Funktion

$$\nabla_x^2 L(x, \mu, \lambda) = \nabla^2 f(x) + \sum_{i=1}^p \mu_i \nabla^2 g_i(x) + \sum_{j=1}^m \lambda_j \nabla^2 h_j(x)$$

über den Kegel

$$C := \{v \in \mathbb{R}^n \mid \begin{array}{l} \nabla_x g_i(x)v \leq 0, \quad i \in I, \mu_i = 0, \\ \nabla_x g_i(x)v = 0, \quad i \in I, \mu_i > 0, \\ \nabla_x h_j(x)v = 0, \quad j \in \{1, \dots, m\} \end{array} \}.$$

positiv definit ist.

#### Bemerkung 3.7

Für die Minimierung einer Funktion  $f : \mathbb{R} \rightarrow \mathbb{R}$  ohne Nebenbedingungen ( $m = p = 0$ ) vereinfachen sich die notwendigen und hinreichenden Bedingungen zu den bereits in der Oberstufe gelehrt Bedingungen

$$\nabla f = f'(x) = 0, \quad \nabla^2 f(x) = f''(x) > 0.$$

### 3.1.3 Algorithmische Vorgehensweise

In diesem Abschnitt soll kurz skizziert werden, wie die für diese Arbeit verwendeten numerischen Verfahren funktionieren. Ausführlicher finden sich das SQP- und Innere-Punkte-Verfahren bei Büskens et al. [8]. Weitere Algorithmen werden zum Beispiel von Güler [11] und Bonnans et al. [12] vorgestellt.

#### 3.1.3.1 Ein Spezialfall

Falls alle Restriktionen als Gleichungsrestriktionen vorliegen, vereinfachen sich die KKT-Bedingungen zu

$$\nabla_x L(x, \mu, \lambda) = \nabla f(x) + \lambda^T \nabla h(x) = 0.$$

Zusammen mit den Restriktionsgleichungen  $h(x) = 0$  gilt es also die Nullstelle von

$$F(x, \lambda) := (\nabla f(x) + \lambda^T \nabla h(x), h(x))$$

zu bestimmen. Dies ist numerisch beispielsweise mit dem Newton-Verfahren iterativ möglich. Die Iterationsvorschrift lautet in diesem Fall

$$y^{k+1} = y^k - F'(y^k)^{-1}F(y^k), \quad k = 0, 1, 2, \dots$$

wobei  $F'(y) = [\frac{\partial F_i}{\partial y_j}(y)]_{ij}$  die Jacobi Matrix von  $F$  bezeichnet und  $y := (x, \lambda)$  die Variablen  $x$  und  $\lambda$  zusammenfasst. Um nicht die Inverse der Matrix bestimmen zu müssen, wird stattdessen die nächste Iterierte  $y^{k+1}$  durch das Lösen des linearen Gleichungssystems

$$F(y^k) + F'(y^k)(y^{k+1} - y^k) = 0$$

bestimmt.

### 3.1.3.2 Innere-Punkte-Verfahren

Sind Ungleichungsrestriktionen vorhanden, kann man das Optimierungsproblem (3.1) so umformulieren, dass aus den Ungleichungsrestriktionen Gleichungsrestriktionen werden. Dazu werden in einem ersten Schritt Variablen  $s_i \geq 0$  eingeführt, mit denen die Ungleichungsrestriktionen in Gleichungsrestriktionen transformiert werden können:

$$g_i(x) + s_i = 0, \quad s_i \geq 0, \quad i = 1, \dots, p.$$

Die neuen Ungleichungsbedingungen  $s_i \geq 0$  können eliminiert werden, indem zu der Zielfunktion eine sogenannte Barriere-Funktion, häufig der negative Logarithmus, addiert wird. Diese Funktion sichert bereits die Positivität des Arguments. Darüber hinaus wird der negative Logarithmus für kleine Werte sehr groß, wodurch versucht wird die Lösung des Optimierungsproblems im Inneren der Menge der positiven Koordinaten zu halten [10]. Das neue Optimierungsproblem lautet

$$\begin{aligned} \min \quad & f(x) - \eta \sum_{i=1}^p \log s_i \\ \text{unter} \quad & g_i(x) + s_i = 0 \quad \text{für } i = 1, \dots, p \\ & h_j(x) = 0 \quad \text{für } j = 1, \dots, m \\ & x \in \mathbb{R}^n. \end{aligned} \tag{3.4}$$

und lässt sich, wie in 3.1.3.1 beschrieben, mit dem Newton-Verfahren lösen. Für eine Folge  $\eta_k \rightarrow 0$  ist zu erwarten, dass die Lösungen der Barriereprobleme (3.4) gegen die optimale Lösung des Ursprungsproblems (3.1) konvergiert [10]. Das Hauptanwendungsgebiet der Innere-Punkte-Verfahren (IP-Verfahren) sind lineare und quadratische Optimierungsprobleme.

### 3.1.3.3 SQP-Verfahren

Das SQP-Verfahren<sup>1</sup> gehört zu den besten Algorithmen für nichtlineare Probleme [8]. Das Grundprinzip besteht darin, die nichtlinearen Funktionen in (3.1) mit einer Taylor-Entwicklung bis zum zweiten Grade anzunähern [9]. Dadurch entsteht ein quadratisches Optimierungsproblem, welches mit speziellen QP-Solvern gelöst wird. Für das Lösen des quadratischen Teilprogramms kann man etwa das oben beschriebene IP-Verfahren verwenden [8]. Die Lösung  $d^k$  wird als Suchrichtung in der Iterationsvorschrift

$$x^{k+1} = x^k + d^k \quad k = 0, 1, 2, \dots$$

verwendet. Häufig wird dabei eine variable Schrittweite  $\alpha^k \in (0, 1]$  gewählt, so dass die Iterationsvorschrift durch

$$x^{k+1} = x^k + \alpha^k d^k$$

gegeben ist. Damit die Folge  $(x^k)_{k \in \mathbb{N}}$  für  $k \rightarrow \infty$  und beliebigen Startwert  $x^0$  konvergiert, müssen Globalisierungsstrategien verwendet werden. Methoden zur Globalisierung sowie zur Bestimmung der Schrittweite werden von Büskens et al. [8] vorgestellt.

## 3.2 Optimale Steuerung

### 3.2.1 Grundlegende Definitionen

Während sich die Optimierung mit der optimalen Bestimmung von Parametern, also von endlich vielen Zahlen beschäftigt, ist das Ziel der optimalen Steuerung die optimale Bestimmung von Funktionen. Dabei gilt es eine stückweise stetige *Steuerung*  $u \in \mathcal{C}_p^0([t_0, t_f], \mathbb{R}^m)$  so zu bestimmen, dass unter Einhaltung von Nebenbedingungen, die auch das dynamische Verhalten

$$\dot{x}(t) = f(x(t), u(t), t), \quad t \in [t_0, t_f] \quad (3.5)$$

des *Systemzustands*  $x \in \mathcal{C}_p^1([t_0, t_f], \mathbb{R}^n)$  berücksichtigen, ein Zielfunktional

$$I[x, u] = \Phi(x(t_f)) + \int_{t_0}^{t_f} f_0(x(t), u(t), t) dt$$

minimiert wird. Dabei kann ohne Einschränkungen  $t_0 = 0$  als Anfangszeitpunkt gewählt werden. Mit  $t_f$  wird der Endzeitpunkt bezeichnet. Die Funktionen

<sup>1</sup>SQP steht für *Sequential Quadratic Programming*

$\Phi : \mathbb{R}^n \rightarrow \mathbb{R}$  und  $f : \mathbb{R}^n \times \mathbb{R}^m \times [0, t_f] \rightarrow \mathbb{R}^n$ ,  $f_0 : \mathbb{R}^n \times \mathbb{R}^m \times [0, t_f] \rightarrow \mathbb{R}$  seien bezüglich aller Argumente stetig differenzierbar.  
 Für  $\Phi \equiv 0$  wird die Darstellung des Zielfunktional als *Lagrange-Form* bezeichnet, wohingegen für  $f_0 \equiv 0$  das Zielfunktional in *Mayer-Form* vorliegt.

Da das dynamische Verhalten der Laufkatze (2.5) nicht explizit von der Zeit  $t$  abhängt, also  $f = f(x(t), u(t))$  gilt, werden hier nur solche Systeme betrachtet.  
 Das vollständige *optimale Steuerungsproblem* lautet

$$\begin{aligned} \min_{x,u} \quad & \Phi(x(t_f)) + \int_0^{t_f} f_0(x(t), u(t)) dt \\ \text{unter} \quad & \dot{x}(t) = f(x(t), u(t)), \\ & x(0) = x_0, \\ & \omega(x(t_f)) = 0, \\ & g(x(t), u(t)) \leq 0, \quad t \in [0, t_f]. \end{aligned} \tag{3.6}$$

Die Randbedingungen werden durch  $x(0) = x_0$  sowie durch  $\omega(x(t_f)) = 0$  mit  $\omega : \mathbb{R}^n \rightarrow \mathbb{R}^r$ ,  $r < n$  beschrieben. Über eine hinreichend oft stetig differenzierbare Funktion  $g : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^p$  fließen *Steuer-* sowie *Zustandsbeschränkungen* in das Problem ein. Die Steuerbeschränkungen sind häufig durch *Boxbeschränkungen*

$$u_{i,min} \leq u_i \leq u_{i,max} \quad \text{für } i = 1, \dots, m$$

gegeben.

Ist der Endzeitpunkt  $t_f$  nicht fest vorgegeben, lässt sich durch eine Transformation des Problems (3.6) auf das Zeitintervall  $[0, 1]$  der Endzeitpunkt  $t_f$  als zusätzlicher Freiheitsgrad zur Optimierung nutzen. Das optimale Steuerungsproblem mit variabler Endzeit lautet dann

$$\begin{aligned} \min_{x,u,t_f} \quad & \Phi(x(1)) + \int_0^1 f_0(x(t), u(t)) \cdot t_f dt \\ \text{unter} \quad & \dot{x}(t) = f(x(t), u(t)) \cdot t_f, \\ & x(0) = x_0, \\ & \omega(x(1)) = 0, \\ & g(x(t), u(t), t_f) \leq 0, \quad t \in [0, 1], \end{aligned} \tag{3.7}$$

wobei hier  $u \in \mathcal{C}_p^0([0, 1], \mathbb{R}^m)$  und  $x \in \mathcal{C}_p^1([0, 1], \mathbb{R}^n)$  sind. Für die variable Zeit können, wie für die anderen Variablen, Beschränkungen gesetzt werden, weshalb  $g$  hier von der Endzeit  $t_f$  abhängt. Zum Beispiel sollte  $t_f > 0$  gefordert werden.

### 3.2.2 Lösungsansätze

Es gibt zwei unterschiedliche Ansätze zur Lösung von Optimalsteuerungsproblemen. Die indirekten Verfahren überführen (3.6) mit dem Minimumprinzip von Pontryagin

in ein Randwertproblem, welches anschließend mit speziellen Randwertproblemlösern gelöst werden kann. Hierauf soll in dieser Arbeit nicht weiter eingegangen werden. Die grundlegenden Definitionen und Sätze zu diesem Ansatz, sowie die Anwendung auf das optimale Steuerungsproblem der Laufkatze, sind bei Knauer [1] zu finden.

In dieser Arbeit wird ein direktes Verfahren benutzt. Dazu wird das optimale Steuerungsproblem mit variabler Endzeit (3.7) durch Diskretisierung in ein nichtlineares Optimierungsproblem (3.1) überführt.

Um numerische Integrationsverfahren zu umgehen, können die Integrale der Lagrange-Form mit einer weiteren Zustandsvariable  $F_0$  und einer zusätzlichen Differentialgleichung

$$\dot{F}_0(t) = f_0(x(t), u(t)) \cdot t_f, \quad F_0(0) = 0 \quad (3.8)$$

nach dem Hauptsatz der Differential- und Integralrechnung in die Mayer-Form

$$\Phi(x(1), F_0(1)) = F_0(1)$$

überführt werden. Umgekehrt kann auch die Mayer-Form in eine Lagrange-Form transformiert werden. Die Formulierungen sind also äquivalent. Das vollständige Problem mit freier Verfahrzeit in Mayer-Form lautet

$$\begin{array}{ll} \min_{F_0, x, u, t_f} & F_0(1) \\ \text{unter} & \dot{F}_0(t) = f_0(x(t), u(t)) \cdot t_f, \\ & \dot{x}(t) = f(x(t), u(t)) \cdot t_f, \\ & F_0(0) = 0, \\ & x(0) = x_0, \\ & \omega(x(1)) = 0, \\ & g(x(t), u(t), t_f) \leq 0, \quad t \in [0, 1]. \end{array} \quad (3.9)$$

Für die numerische Auswertung der Differentialgleichungen werden an endlich vielen Stützstellen

$$0 = t_0 < t_1 < \dots < t_N = 1, \quad N \in \mathbb{N}$$

Näherungswerte bestimmt. Die Approximationen von  $u(t_i)$ ,  $x(t_i)$  und  $F_0(t_i)$  werden mit  $u^i$ ,  $x^i$  beziehungsweise  $F_0^i$  bezeichnet.

Verwendet man das explizite Euler-Verfahren

$$\begin{array}{ll} x^0 & := x_0; & x^{i+1} & := x^i + hf(x^i, u^i) \cdot t_f; \\ F_0^0 & := 0; & F_0^{i+1} & := F_0^i + hf_0(x^i, u^i) \cdot t_f; \end{array}$$

für  $i = 0, \dots, N$  zur Lösung der Differentialgleichungen

$$\dot{x}(t) = f(x(t), u(t)) \cdot t_f \quad \text{und} \quad \dot{F}_0(t) = f_0(x(t), u(t)) \cdot t_f,$$

mit den Anfangswerten  $x_0$  beziehungsweise 0, lässt sich das Problem (3.9) als nicht-lineares Optimierungsproblem schreiben:

$$\begin{aligned} \min \quad & F_0(1) \\ \text{unter} \quad & F_0^{i+1} = F_0^i + hf_0(x^i, u^i) \cdot t_f, \quad i = 0, \dots, N, \\ & x^{i+1} = x^i + hf(x^i, u^i) \cdot t_f, \quad i = 0, \dots, N, \\ & F_0^0 = 0, \\ & x^0 = x_0, \\ & \omega(x^N) = 0, \\ & g(x^i, u^i) \leq 0, \quad i = 0, \dots, N. \end{aligned} \tag{3.10}$$

Verfahren höherer Ordnung, wie das Runge-Kutta-Verfahren lassen sich ebenfalls verwenden. Um jeden Zeitpunkt  $t \in [0, 1]$  eine Steuerung zuzuordnen, wird nach der Berechnung der diskreten Steuerungen  $u^i$  linear interpoliert.

# Kapitel 4

## Optimale Steuerung der Laufkatze

In diesem Kapitel wird das optimale Steuerungsproblem (3.6) für die Laufkatze formuliert. Dazu müssen das dynamische Verhalten (3.5), die Randbedingungen  $x(0) = x_0$  und  $\omega(x(t_f)) = 0$ , die Steuer- und Zustandsbeschränkungen  $g(x(t), u(t))$  sowie Zielfunktionale  $I$  konkret angegeben werden. Die Transformation auf freie Endzeit beziehungsweise in die Mayer-Form lassen sich wie in Abschnitt 3.2.2 durchführen.

### 4.1 Das dynamische Verhalten

In Kapitel 2 wurde das dynamische Verhalten (2.5) der Laufkatze formuliert. Damit die Last schwingungsfrei am Ziel ankommt, muss gefordert werden, dass sämtliche, an der Last angreifenden Kräfte, und somit alle Beschleunigungen, für den Endzeitpunkt  $t_f$  auf Null gebracht werden. Endbedingungen können jedoch nur für Zustandsgrößen, nicht aber für Steuerungen gefordert werden. In (2.5) werden allerdings die Beschleunigungen der Laufkatze und des Seils durch Steuerungen beschrieben. Die relative Beschleunigung des Pendels  $\ddot{d}$  hängt ebenfalls von diesen ab. Dieses Problem lässt sich umgehen, indem man als Steuerung nicht die Beschleunigungen, sondern den Ruck verwendet, welcher durch die dritten Ableitungen nach der Zeit definiert ist. Somit ergibt sich die Systemdynamik

$$\begin{aligned}\ddot{s} &= u_1 \\ \ddot{l} &= u_2 \\ \ddot{d} &= \ddot{s} - (g - \ddot{l}) \cdot \frac{d}{l}.\end{aligned}\tag{4.1}$$

Durch die Substitution

$$\begin{aligned}
 x_1 &:= s && \text{Position des Wagens} \\
 x_2 &:= \dot{s} && \text{Geschwindigkeit des Wagens} \\
 x_3 &:= d && \text{relative Verschiebung des Pendels} \\
 x_4 &:= \dot{d} && \text{relative Geschwindigkeit des Pendels} \\
 x_5 &:= \ddot{s} && \text{Beschleunigung des Wagens} \\
 x_6 &:= l && \text{Seillänge} \\
 x_7 &:= \dot{l} && \text{Geschwindigkeit des Seils} \\
 x_8 &:= \ddot{l} && \text{Beschleunigung des Seils}
 \end{aligned} \tag{4.2}$$

lässt sich (4.1) in ein Differentialgleichungssystem erster Ordnung umschreiben:

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= x_5 \\
 \dot{x}_3 &= x_4 \\
 \dot{x}_4 &= x_5 - (g - x_8) \cdot \frac{x_3}{x_6} \\
 \dot{x}_5 &= u_1 \\
 \dot{x}_6 &= x_7 \\
 \dot{x}_7 &= x_8 \\
 \dot{x}_8 &= u_2
 \end{aligned} \tag{4.3}$$

Um die Bezeichnungen aus Abschnitt 3.2 beizubehalten, wird (4.3) mit  $x = (x_1, \dots, x_8)^T$  und  $u = (u_1, u_2)^T$  zusammengefasst zu  $\dot{x} = f(x(t), u(t))$ .

## 4.2 Anfangs- und Endbedingungen

Für die Aufgabe, Ware von  $(s_0, l_0)$  nach  $(s_f, l_f)$  zu transportieren, müssen die Geschwindigkeiten und Beschleunigungen des Wagens und des Seils am Start- und Zielpunkt Null sein, da sonst kein Be- und Entladen möglich ist. Die Last darf dabei auch nicht schwingen, weshalb weiter gefordert wird, dass die relative Verschiebung für  $t = 0$  und  $t = t_f$  ebenfalls Null ist. Hieraus folgt auch für die relative Geschwindigkeit des Pendels zu den Zeitpunkten  $t = 0$  und  $t = t_f$

$$\dot{x}_4(t) = x_5(t) - (g - x_8(t)) \cdot \frac{x_3(t)}{x_6(t)} = 0.$$

Insgesamt erhält man die Randbedingungen

$$\begin{aligned}
x_1(0) &= s_0, & x_1(t_f) &= s_f \\
x_2(0) &= 0, & x_2(t_f) &= 0 \\
x_3(0) &= 0, & x_3(t_f) &= 0 \\
x_4(0) &= 0, & x_4(t_f) &= 0 \\
x_5(0) &= 0, & x_5(t_f) &= 0 \\
x_6(0) &= l_0, & x_6(t_f) &= l_f \\
x_7(0) &= 0, & x_7(t_f) &= 0 \\
x_8(0) &= 0, & x_8(t_f) &= 0.
\end{aligned} \tag{4.4}$$

Mit den Notationen aus Kapitel 3.2 lautet (4.4)

$$x(0) = x_0 = s_0 e_1 + l_0 e_6, \quad \omega(x(t_f)) = x(t_f) - s_f e_1 - l_f e_6 = 0. \tag{4.5}$$

wobei  $e_i \in \mathbb{R}^8$  den  $i$ -ten Einheitsvektor bezeichnet. Alternativ könnte man hier statt  $\omega(x(t_f)) = 0$ , analog zum Anfangswert,  $x(t_f) = x_{t_f}$  schreiben. Die Schreibweise  $\omega(x(t_f)) = 0$  ist aber allgemeiner. So wäre zum Beispiel bei einem Notstopp die Endposition  $(s_f, l_f)$  der Last unerheblich. Dies lässt sich durch

$$\omega(x(t_f)) = \sum_{j \in J} x_j(t_f) e_j = 0, \quad J = \{2, 3, 4, 5, 7, 8\}$$

beschreiben. Für die Simulation eines Notstopps sollte das System bereits in Bewegung sein:

$$x(0) = x_0 = s_0 e_1 + \dot{s}_0 e_2 + d_0 e_3 + \dot{d}_0 e_4 + \ddot{s}_0 e_5 + l_0 e_6 + \dot{l}_0 e_7 + \ddot{l}_0 e_8.$$

## 4.3 Zustands- und Steuerbeschränkungen

Die technischen Einschränkungen müssen bei der Bahnoptimierung berücksichtigt werden. Die Länge und Höhe der Regale, beziehungsweise die Höhe des Aufhängungspunktes beschränken die erlaubte Position des Wagens und die Seillänge:

$$\begin{aligned}
s_{\min} &\leq x_1 \leq s_{\max} \\
l_{\min} &\leq x_6 \leq l_{\max}.
\end{aligned} \tag{4.6}$$

Dabei muss  $l_{\min} > 0$  gelten, da im Modell (2.5) durch die Länge geteilt wird.

Die Motorleistung, beziehungsweise die Wünsche der Betreiber, führen zu Beschränkungen der Geschwindigkeiten und Beschleunigungen:

$$\begin{aligned}
\dot{s}_{\min} &\leq x_2 \leq \dot{s}_{\max} \\
\ddot{s}_{\min} &\leq x_5 \leq \ddot{s}_{\max} \\
\dot{l}_{\min} &\leq x_7 \leq \dot{l}_{\max} \\
\ddot{l}_{\min} &\leq x_8 \leq \ddot{l}_{\max}
\end{aligned} \tag{4.7}$$

Starke Änderungen im Beschleunigungsverhalten, also betragsmäßig große Rucke, lassen sich in der Praxis nur schwer realisieren. Da in dieser Arbeit der Ruck des Wagens sowie der Ruck des Seils als Steuerung gegeben sind, bietet sich die Beschränkung der Steuergrößen

$$\begin{aligned} u_{1,\min} &\leq u_1 \leq u_{1,\max} \\ u_{2,\min} &\leq u_2 \leq u_{2,\max} \end{aligned} \quad (4.8)$$

an, um Trajektorien zu erhalten, die von der Laufkatze leichter nachzufahren sind.

Weitere Beschränkungen, etwa an den Ausschlagwinkel  $\arcsin \frac{x_3}{x_6}$  oder durch Hindernisse, werden von Knauer [1] berücksichtigt.

## 4.4 Zielfunktionale

In diesem Abschnitt werden einige Zielfunktionale vorgestellt, die bei der Bahnplanung eine Rolle spielen. Damit die Minimierung nicht zu einem großen negativen Wert führt, wird hier die Positivität des Zielfunktional gefordert.

### 4.4.1 Minimierung der Verfahzeit

Um zeitoptimale Bahnen zu erhalten, wird als Zielfunktional

$$I_1[x, u] = \int_0^{t_f} dt = t_f$$

gewählt. Bei der Minimierung der Verfahzeit wird die maximale Motorleistung angestrebt, wodurch es zu sogenannten *bang-bang-Steuerungen* kommt, siehe Pesch [2]. Anschaulich bedeutet dies, dass die Steuerungen von einem Extrem zum anderen umschalten. Dadurch werden die Bahnen sehr ruckhaft und lassen sich nur schwer nachfahren. Darüber hinaus ist ein hoher Verschleiß und Energieaufwand zu erwarten. Aus diesen Gründen wird in der Praxis die Verfahzeit nur in Kombination mit weiteren Zielfunktionalen optimiert.

### 4.4.2 Minimierung des Rucks

Das Zielfunktional zur Minimierung des Rucks ist durch

$$I_2[x, u] = \int_0^{t_f} c_1 u_1^2 + c_2 u_2^2 dt \quad (4.9)$$

definiert. Hierbei sind  $c_1$  und  $c_2$  nichtnegative Gewichte, die in dieser Arbeit gleich 1 gesetzt werden. Das Quadrieren der Steuerungen garantiert die Positivität des Integranden und hat gegenüber dem Absolutbetrag den Vorteil der Differenzierbarkeitseigenschaft.

Durch die Ruckminimierung werden die Bahnen glatter und es muss weniger Energie aufgewendet werden. Darüber hinaus können mit  $I_2$  *bang-bang-Steuerungen* verhindert werden [2].

### 4.4.3 Minimierung des Schwingungsverhaltens

Um das Schwingungsverhalten zu minimieren, reicht es nicht zu fordern, dass die relative Auslenkung möglichst klein ist, da dies zur Folge hätte, dass das Seil zu sehr eingezogen wird. Zielführender ist die Betrachtung des Quotienten aus der relativen Verschiebung und der Seillänge:

$$I_3[x, u] = \int_0^{t_f} \left( \frac{x_3}{x_6} \right)^2 dt. \quad (4.10)$$

Da der Ausschlagwinkel durch  $\varphi = \arcsin \frac{x_3}{x_6}$  gegeben ist, wird durch dieses Funktional der mittlere Winkel minimiert.

Für die Minimierung der Schwingung wird in (4.10) die relative Verschiebung  $x_3$  durch die Geschwindigkeit der relativen Verschiebung  $x_4$  ersetzt:

$$I_4[x, u] = \int_0^{t_f} \left( \frac{x_4}{x_6} \right)^2 dt.$$

Weitere Zielfunktionale sind denkbar, zum Beispiel die Minimierung der Winkelgeschwindigkeit <sup>1</sup>

$$I_5[x, u] = \int_0^{t_f} \left( \frac{x_4 \cdot x_6 - x_3 \cdot x_7}{x_6^2} \right)^2 dt.$$

### 4.4.4 Kombinationen

Wird nur eins der oben erwähnten Zielfunktionale minimiert, ist das Ergebnis meistens nicht zufriedenstellend. So wird beispielsweise die Minimierung des Schwingungsverhalten dadurch erzielt, dass die Last sehr langsam bewegt wird.

Besser ist es, eine gewichtete Linearkombination der Zielfunktionale zu minimieren:

$$I[x, u] = \sum_{i=1}^{n_I} w_i I_i[x, u]. \quad (4.11)$$

<sup>1</sup>Hierbei ist für die Winkelgeschwindigkeit die Näherung  $\dot{\varphi}(t) \approx \frac{d}{dt} \left( \frac{d(t)}{l(t)} \right) = \frac{\dot{d}(t)l(t) - d(t)\dot{l}(t)}{l^2(t)}$  verwendet worden [2].

Bei der Gewichtung ist zu beachten, dass die Zielfunktionale unterschiedliche Größenordnungen besitzen. Durch eine geeignete Skalierung der Zielfunktionale kann man diese auf die gleiche Größenordnung bringen, so dass die Gewichte  $w_i$  ungefähr der Relevanz entsprechen. Als Skalare kann man zum Beispiel  $1/\tilde{I}_i$  wählen, wobei  $\tilde{I}_i$  ein für das Zielfunktional  $I_i$  gemittelter Wert ist. Das Ziel ist also

$$I[x, u] = \sum_{i=1}^{n_I} w_i \frac{I_i[x, u]}{\tilde{I}_i} \quad (4.12)$$

zu minimieren.

Alternativ ist auch eine hierarchische Optimierung denkbar, bei der zuerst das Zielfunktional mit der höchsten Priorität optimiert wird. Hiervon abhängig wird in einem zweiten Schritt die zweitwichtigste Zielgröße optimiert und so weiter. Dies ist jedoch rechentechnisch aufwendiger, siehe Knauer [6].

Mehr zu der Minimierung verschiedener Zielfunktionen findet man in der Literatur unter dem Begriff der *Pareto-Optimierung*.

# Kapitel 5

## Numerische Ergebnisse

### 5.1 Numerische Umsetzung

In dieser Arbeit wird zum Lösen des diskretisierten Problems der Solver *WORHP*<sup>1</sup> mit Hilfe einer Studentenversion der Modellierungssprache *AMPL* verwendet.

*WORHP* ist ein iterativer Solver für hochdimensionale dünnbesetzte nichtlineare Optimierungsprobleme und verwendet das in Kapitel 3 vorgestellte SQP und IP-Verfahren. Ausführlichere Informationen finden sich bei Nikolayzik et al [7].

Beispiele und Syntaxerläuterungen zu *AMPL* werden in [13] gegeben. Die Studentenversion von *AMPL* ist auf 300 Variablen und Bedingungen beschränkt. Aus diesem Grund konnten maximal 29 Stützstellen gewählt werden.<sup>2</sup>

Die wichtigsten Ausschnitte des Programms befinden sich im Anhang B. Der gesamte Quellcode, sowie sämtliche Diagramme, sind auf der beigelegten CD zu finden.

Die Berechnungen wurden auf einem MacBook Pro 7.1 mit einem 2.4 GHz schnellen Intel Core 2 Duo Prozessor mit 4 GB Arbeitsspeicher durchgeführt.

Die in dieser Arbeit verwendeten Steuer- und Zustandsbeschränkungen sind in Tabelle 5.1 aufgelistet.

	$u_1$	$u_2$	$s$	$\dot{s}$	$\ddot{s}$	$l$	$\dot{l}$	$\ddot{l}$
untere Grenze	-3	-1	-0.2	-3	-0.8	0.01	-0.75	-1
obere Grenze	3	1	30	3	0.8	16	1	1

**Tabelle 5.1:** Verwendete Steuer- und Zustandsbeschränkungen, sofern nicht explizit anders gegeben.

<sup>1</sup>WORHP steht für 'We optimize really huge problems'.

<sup>2</sup>Acht Zustands- und zwei Steuerfunktionen an 29 Stützstellen ergeben 290 Variablen. Hinzu kommt die Verfahrzeit  $t_f$ .

### Startschätzungen

Der Erfolg des Iterationsverfahrens hängt stark von den geschätzten Startwerten ab, vor allem von der Schätzung der Fahrzeit, da diese als Faktor in jeder Zustandsgleichung von (3.10) vorkommt.

Mit einer gemittelten Geschwindigkeit  $\bar{v}_{Wagen}$  in horizontaler Richtung lässt sich die Fahrzeit für waagerechte Fahrten von  $s_0$  nach  $s_f$  durch

$$t_{Wagen} = \frac{|s_0 - s_f|}{\bar{v}_{Wagen}}$$

berechnen. Analog wird zur Beförderung der Last von  $l_0$  nach  $l_f$  eine Zeit von

$$t_{Seil} = \frac{|l_0 - l_f|}{\bar{v}_{Seil}}$$

benötigt. Die gemittelten Geschwindigkeiten werden auf die Hälfte der maximalen Geschwindigkeiten geschätzt:

$$\bar{v}_{Wagen} := \frac{\dot{s}_{\max}}{2}, \quad \bar{v}_{Seil} := \frac{\dot{l}_{\max}}{2}.$$

Da beide Bewegungen gleichzeitig möglich sind und die Komponenten senkrecht zueinander stehen, lässt sich mit dem Satz von Pythagoras die Fahrzeit schätzen auf

$$t_f \approx 2 \sqrt{\left(\frac{s_f - s_0}{\dot{s}_{\max}}\right)^2 + \left(\frac{l_f - l_0}{\dot{l}_{\max}}\right)^2}.$$

Die Position des Wagens sowie die Seillänge werden zwischen Start- und Endpunkt linear interpoliert. Alle übrigen Werte können auf Null gesetzt werden.

## 5.2 Auswertung

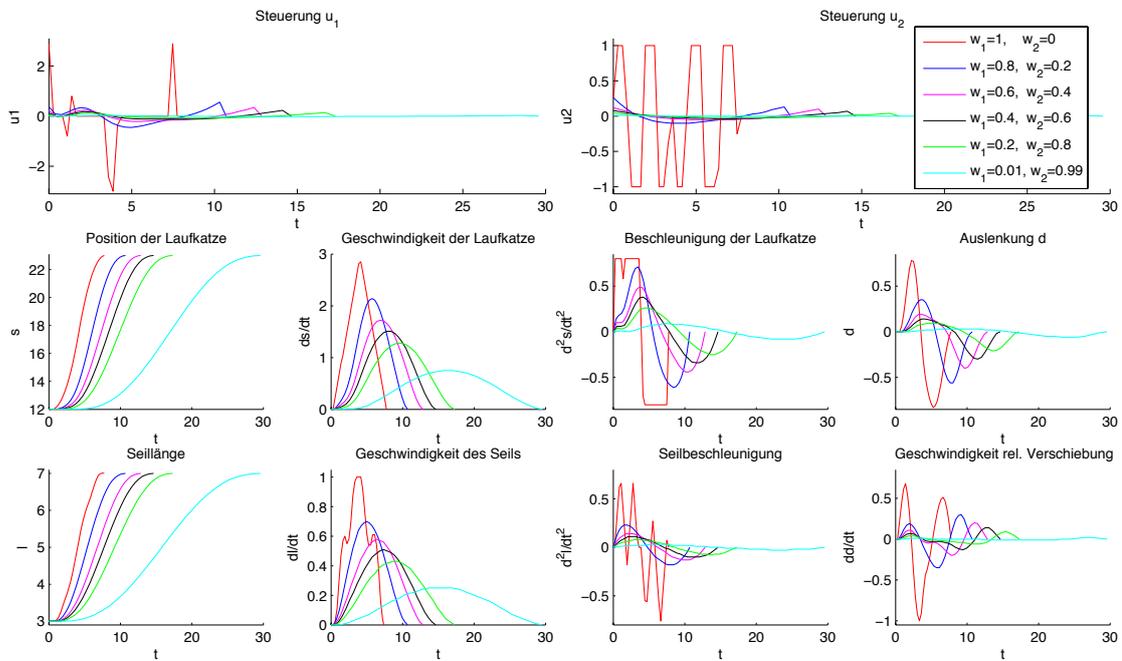
### 5.2.1 Wahl und Einfluss der Gewichte

Wie in Abschnitt 4.4.4 erwähnt wird, ist die Minimierung einer gewichteten Kombination sinnvoller als die Minimierung eines einzelnen Zielfunktionals. Für die Skalierung der Funktionale lassen sich die Reziproken der Mittelwerte  $\tilde{I}_i$  verwenden, siehe (4.12). Unter Berücksichtigung der Beschränkungen aus Tabelle 5.1 wurden anhand von 100 zufälligen Fahrten die in Tabelle 5.2 aufgelisteten Mittelwerte gebildet. Dabei wurden alle Gewichte auf 1 gesetzt. Nach der Skalierung der Funktionale entsprechen die Gewichte  $w_i$  ungefähr der verhältnismäßigen Berücksichtigung des  $i$ -ten Funktionals  $I_i/\tilde{I}_i$ .

$\tilde{I}_1$	$\tilde{I}_2$	$\tilde{I}_3$	$\tilde{I}_4$
11,501	1,178	0,026	0,014

**Tabelle 5.2:** Gemittelte Werte für die Funktionale für 100 zufällige Testbahnen.

Welchen Einfluss das Ändern des Gewichtungsverhältnisses auf die Bahntrajektorien hat, soll anhand eines Beispiels illustriert werden. Dazu werden die Gewichte  $w_3$  und  $w_4$  auf Null gesetzt und das relative Verhältnis zwischen  $w_1$  und  $w_2$  verändert. Abbildung 5.1 zeigt die Steuerungen und Zustände für unterschiedliche Verhältnisse der Gewichte  $w_1$  und  $w_2$  an einer Testfahrt.



**Abbildung 5.1:** Steuerungen und Zustände für eine Testfahrt von  $(s, l) = (12, 3)$  nach  $(23, 7)$  für unterschiedliche Gewichtungen der Zeit ( $w_1$ ) und des Rucks ( $w_2$ ).

Ist auch  $w_2 = 0$ , so erhält man die schnellstmögliche Bahntrajektorie. Bahnen mit dieser Gewichtung werden im Folgenden als *schnellste Bahn* bezeichnet und dienen als Vergleichsbahn. In Abbildung 5.1 fällt die Seilsteuerung für die schnellste Bahn durch das starke Oszillieren zwischen den maximal erlaubten Werten besonders auf. Diese Steuerungen bewirken zwar, dass die Last ihr Ziel in diesem Szenario nach 7,8 Sekunden schwingungsfrei erreicht, aber zugleich auch, dass die Beschleunigungen der Laufkatze und des Seils in kürzester Zeit von einem Extrem auf das andere Extrem umschlagen. Die maximale Auslenkung ist für die schnellste Bahn mit 0,8 Metern am stärksten ausgeprägt. Wie bereits in 4.4.1 beschrieben, eignen sich diese Gewichtungen nicht für die Praxis.

Bezieht man die Minimierung des Rucks mit ein, so sind alle Steuer- und Zustandsfunktionen deutlich glatter. Die maximal angenommenen Steuerungen, Beschleunigungen, Geschwindigkeiten und Auslenkungen werden mit zunehmender Gewichtung des Rucks kleiner. Dafür braucht die Last länger, um ihr Ziel zu erreichen. Würde man nur den Ruck minimieren, also  $w_1$  auf Null setzen, so bräuhete die Laufkatze über 300 Sekunden, also ungefähr 10 mal länger als bei einer Gewichtung von  $w_1 = 0,01$  und  $w_2 = 0,99$ . Aus diesem Grund wird in der Praxis die Zeit stets mitberücksichtigt.

Für  $w_1 > 0$  und  $w_3 = w_4 = 0$  verschieben sich zwar mit zunehmender Gewichtung des Rucks die Extremstellen zeitlich nach hinten und die Funktionen werden flacher. Die Grundgestalt bleibt aber erhalten.

Bei allen Bahnkurven ist kurz vor dem Erreichen des Zielpunktes ein Knick in der Steuerung zu erkennen. Dieser Knick taucht auch bei allen später betrachteten Diagrammen auf. Ursache hierfür ist wahrscheinlich die geringe Anzahl von Stützpunkten bei dem verwendeten Euler-Verfahren, da die Steuerungen im letzten Iterationsschritt keine Auswirkungen mehr haben.

Im Folgenden werden Bahnkurven betrachtet, die bevorzugt die Verfahrzeit  $I_1$ , das Ruckverhalten  $I_2$  oder das Pendel- und Schwingungsverhalten  $I_3$  und  $I_4$  minimieren. Die Bezeichnungen für diese Bahnen und die gewählten Gewichte sind in Tabelle 5.3 verzeichnet. Die bereits vorgestellte schnellste Bahn dient dem Vergleich.

	1. Fall: ruckarm	2. Fall: wenig Pendeln	3. Fall: schnell	4. Fall: schnellste Bahn
Verfahrzeit $w_1$	0,5	0,2	0,95	1
Ruck $w_2$	0,5	0,05	0,05	0
Winkel $w_3$	0	0,5	0	0
Schwingung $w_4$	0	0,25	0	0

**Tabelle 5.3:** Gewichtungsfaktoren für unterschiedliche Bahnverläufe.

## 5.2.2 Plausibilitätstest

### 5.2.2.1 Durchführung

Um herauszufinden, ob die numerisch berechneten Lösungen des optimalen Steuerungsproblems plausibel sind, werden für die vier Fälle aus Tabelle 5.3 die folgenden Tests durchgeführt.

1. **Stillstand:** Stimmen Start- und Endposition der Last überein, ist zu erwarten, dass die Verfahrzeit und die Steuerungen identisch Null sind.

Test-Nr	Beschreibung	$l_0$	$l_f$	Erwartung
1	Stillstand	5	5	$u \equiv 0, t_f = 0$
2.1	Seil ablassen, Vergleichsbahn	5	9	$u_1 \equiv 0$
2.2	Seil einziehen	9	5	$u_1 \equiv 0$ $t_f[2.2] \geq t_f[2.1]$
2.3	Wie 2.1, aber seitlich verschoben	5	9	$u_2[2.3] \equiv u_2[2.1]$ $t_f[2.3] = t_f[2.1]$
2.4	Doppelter Weg	5	13	$t_f[2.4] > t_f[2.1]$

**Tabelle 5.4:** Plausibilitätstest: Stillstand und vertikale Bewegungen. Die Position der Laufkatze bleibt konstant bei  $s_0 = s_f = 2$ . Bei Test-Nr 2.3 wird  $s_0 = s_f = 7$  gewählt.

- Vertikale Bewegung:** Soll die Last nur angezogen oder herunter gelassen werden, ist keine Bewegung der Laufkatze nötig. Die optimalen Lösungen sollten ergeben, dass  $u_1 \equiv 0$  ist. Da  $|\dot{l}_{\min}| < |\dot{l}_{\max}|$  ist, kann die Last schneller abgesenkt als eingezogen werden<sup>3</sup>. Bei gleicher Distanz darf das Einziehen also nicht schneller gehen als das Absenken. Für größere Distanzen muss die Verfahrzeit größer werden.
- Seitliche Bewegung:** Befinden sich Start- und Endpunkt auf gleicher Höhe, ist eine symmetrische Bahnkurve zu erwarten, da Start- und Endpunkt wegen  $|\dot{s}_{\min}| = |\dot{s}_{\max}|$ ,  $|\ddot{s}_{\min}| = |\ddot{s}_{\max}|$  ausgetauscht werden können. Setzt man diese Grenzen nach unten, darf die Verfahrzeit nicht größer werden.

Eine Übersicht der durchgeführten Tests zum Stillstand und zur vertikalen Bewegung ist in Tabelle 5.4 dargestellt. Die Tests zur seitlichen Bewegung sind in Tabelle 5.5 aufgelistet. Dabei bezeichnen zum Beispiel  $t_f[3.1]$  und  $u[3.1]$  die Verfahrzeit beziehungsweise die Steuerung von dem Test mit der Nummer 3.1.

### 5.2.2.2 Ergebnisse

Der Tabelle A.1 ist zu entnehmen, dass die Erwartungen bezüglich der Verfahrzeit erfüllt sind. Insbesondere ist zu sehen, dass die stärkere Gewichtung der Zeit zu kürzeren Verfahrzeiten führt.

Die aus dem Plausibilitätstests resultierenden Trajektorien bestätigen nicht nur die Erwartungen aus Tabelle 5.4 und 5.5, sondern dienen auch dem besseren Verständnis wie die Laufkatze optimal gesteuert werden kann. Zur besseren Anschauung werden die Ergebnisse teilweise in der Seitenansicht präsentiert. Dabei ist die Laufkatze stark vereinfacht als blauer und die Last als gelber Kasten in äquidistanten Zeitpunkten

<sup>3</sup>Negative Geschwindigkeiten bedeuten, dass das Seil eingezogen wird. Demnach gibt  $\dot{l}_{\min}$  die maximale Hubgeschwindigkeit an.  $\dot{l}_{\max}$  legt die maximale Senkgeschwindigkeit fest.

Test-Nr	Beschreibung	$s_0$	$s_f$	Erwartung
3.1	Vergleichsbahn	2	8	-
3.2	Start und Ziel getauscht	8	2	$u_1[3.2] \equiv -u_1[3.1]$ $u_2[3.2] \equiv u_2[3.1]$ $t_f[3.2] = t_f[3.1]$
3.3	Seitlich verschoben	5	11	$u[3.3] \equiv u[3.1]$ $t_f[3.3] = t_f[3.1]$
3.4	Doppelte Seillänge	2	8	$t_f[3.4] > t_f[3.1]$
3.5	Dreifache Seillänge	2	8	$t_f[3.5] > t_f[3.4]$
3.6	Vierfache Strecke	2	26	$t_f[3.6] > t_f[3.1]$
3.7	Kleinere Geschwindigkeitsgrenzen $\dot{s}_{\min} = -1, \dot{s}_{\max} = 1$	2	8	$t_f[3.7] \geq t_f[3.1]$
3.8	Wie 3.7, zusätzlich kleinere Beschleunigungsgrenzen $\ddot{s}_{\min} = -0.4, \ddot{s}_{\max} = 0.4$	2	8	$t_f[3.8] \geq t_f[3.7]$

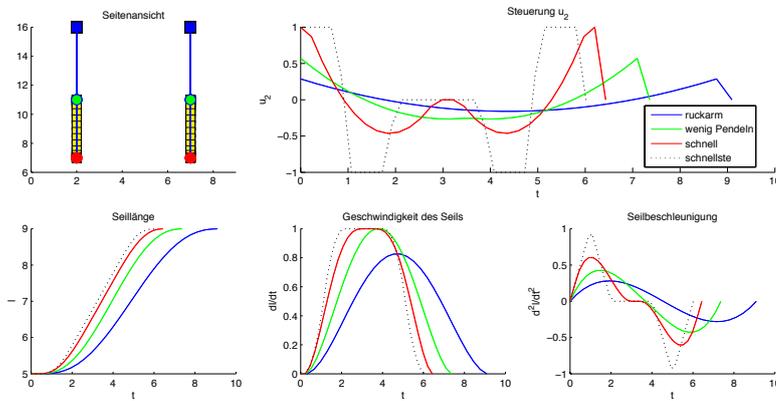
**Tabelle 5.5:** Plausibilitätstest: Horizontale Bewegungen. Hier wurde  $l_0 = l_f = 5$  bzw. bei 3.4  $l_0 = l_f = 10$  und bei 3.5  $l_0 = l_f = 15$  verwendet.

dargestellt. Ein großer Abstand zwischen zwei benachbarten Kästen bedeutet also eine schnelle Bewegung. Die Startkoordinaten sind als grüner Punkt eingezeichnet. Der rote Punkt markiert die Zielkoordinaten.

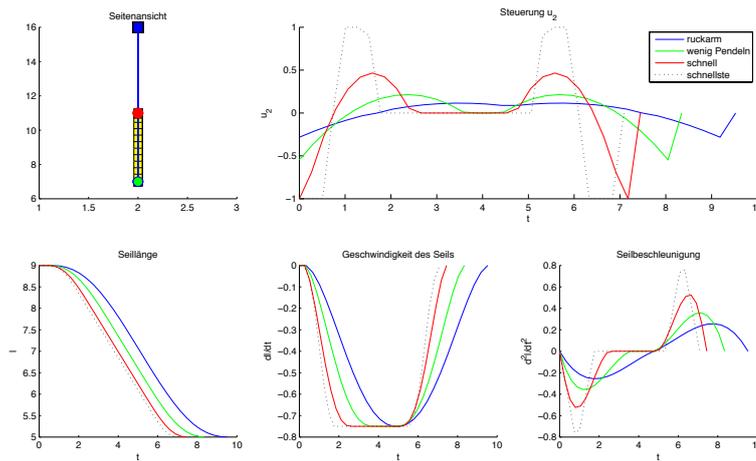
### Vertikale Bewegung

Die Abbildungen 5.2, 5.3 und 5.4 zeigen die Seilsteuerungen der vertikalen Testfahrten des Plausibilitätstest sowie die hieraus resultierenden Zustände. Die Seitenansichten dienen der besseren Übersicht. Es ist zu sehen, dass eine seitliche Verschiebung wie erwartet keine Auswirkungen auf die Steuerung hat. Da  $|\dot{l}_{\min}| < |\dot{l}_{\max}|$  ist, entspricht die Steuerung  $u_2$  beim Anheben der Last nicht der Spiegelung der Steuerung zum Absenken, wie ein Vergleich der Abbildungen 5.2 und 5.3 zeigt. Der Ruck wird in Abbildung 5.3 nach der Beschleunigungsphase auf Null gesetzt, wodurch die Last solange mit der maximalen negativen Geschwindigkeit eingezogen wird bis der, bei  $t = 5$  einsetzende, Ruck die Last abbrems. Muss die Last eine größere Distanz zurücklegen, wie in Abbildung 5.4 gezeigt, wird für die schnelle und pendelarme Bahn ebenfalls die maximale Geschwindigkeit solange angenommen bis die Steuerung ein Bremsmanöver einleitet.

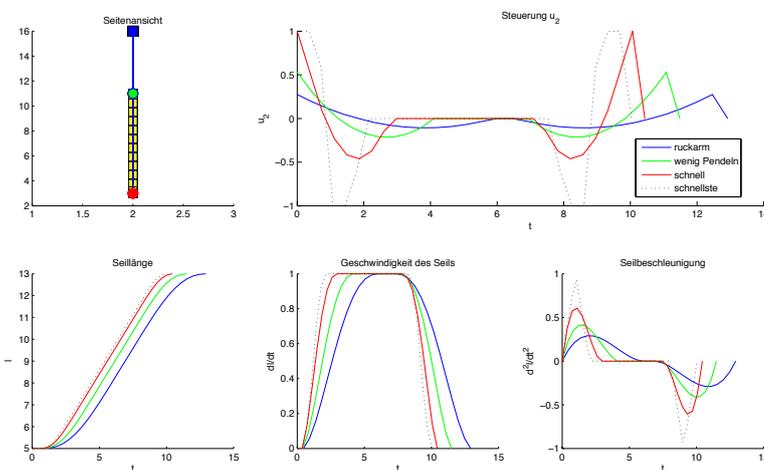
Die ruckarme Bahn erreicht beim Absenken (Abbildung 5.2) nur eine Maximalgeschwindigkeit von  $\dot{l} = 0,82 \frac{m}{s}$ . Da sich dieser Wert vom Betrag kaum von der minimal erlaubten Seilgeschwindigkeit, also der maximalen Einzugs geschwindigkeit,  $\dot{l}_{\min} = -0,75 \frac{m}{s}$  unterscheidet, entspricht die Seilsteuerung der ruckarmen Bahn aus Abbildung 5.3 ungefähr der Spiegelung der Steuerung aus Abbildung 5.2.



**Abbildung 5.2:** Test-Nr. 2.1 und 2.3. Die Steuerung ist unabhängig von der Position der Laufkatze, da die Last nur vertikal bewegt wird. Die Last wird 4 Meter abgesenkt.



**Abbildung 5.3:** Test-Nr. 2.2. Im Gegensatz zu Test-Nr. 2.1 wird hier die Last 4 Meter hochgezogen.

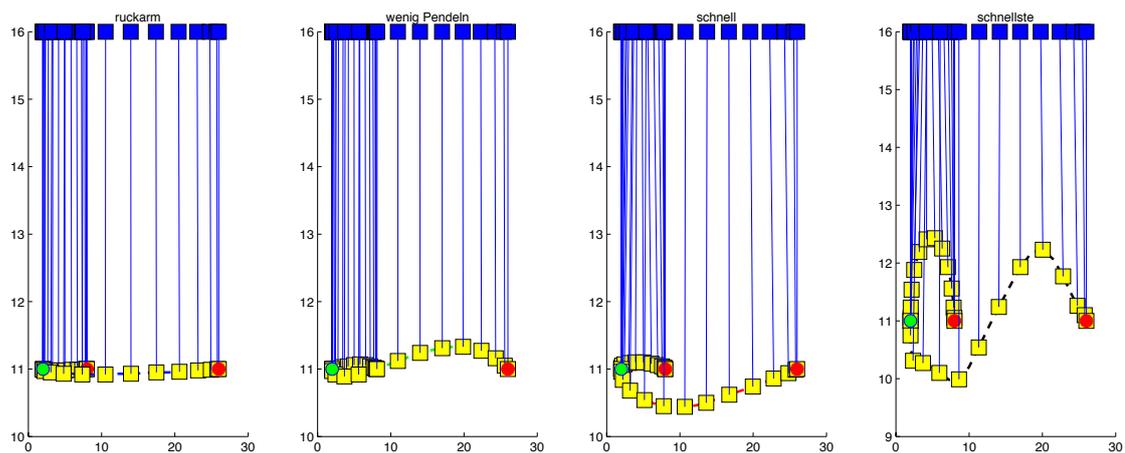


**Abbildung 5.4:** Test-Nr. 2.4. Hier wird im Vergleich zu Test 2.1 (Abbildung 5.2) die doppelte Distanz zurückgelegt.

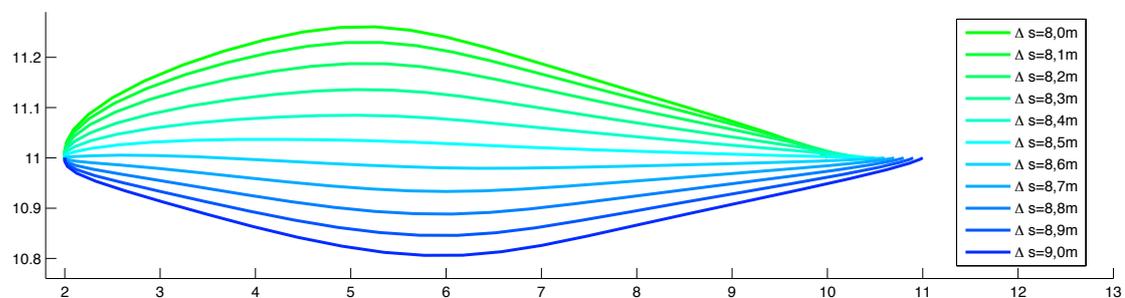
### Seitliche Bewegung

Obwohl Start- und Zielkoordinaten der Last in diesen Tests auf gleicher Höhe liegen, ist die Seilsteuerung ungleich Null. Wie aus Abbildung 5.5 hervorgeht, kann es in manchen Fällen günstiger sein, das Seil einzuziehen und in anderen Fällen das Seil abzusenken. So wird bei der schnellen Bahn das Seil auf der langen Distanz abgesenkt, bei der Bahn mit möglichst geringem Pendelverhalten hingegen wird das Seil unter den gleichen Bedingungen eingezogen. Eine Kombination ist ebenfalls möglich, wie bei der schnellsten Bahn über die große Distanz zu sehen ist.

Die optimale Seilsteuerung hängt nicht nur von den gewählten Gewichten ab, sondern wird auch durch die Distanz beeinflusst. Abbildung 5.6 zeigt am Beispiel der schnellen Bahn diese Abhängigkeit. Der Übergang zwischen Seil-Einziehen und Seil-Absenken ist dabei gleichmäßig.



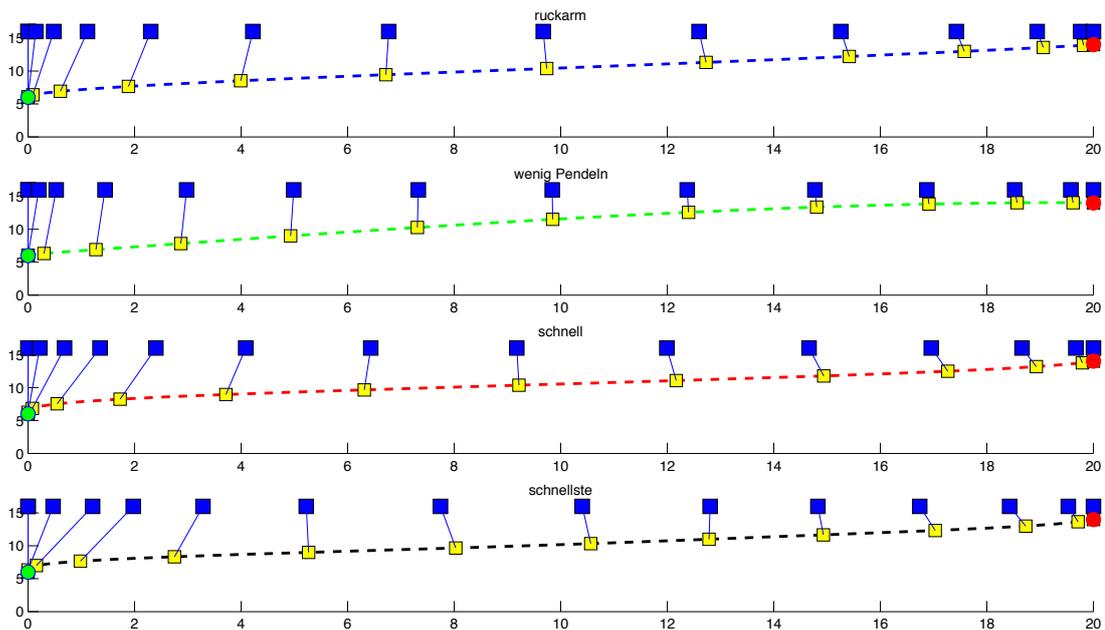
**Abbildung 5.5:** Seitenansicht für verschiedene Prioritäten und zwei unterschiedliche Strecken auf gleicher Höhe.



**Abbildung 5.6:** Seitenansicht für unterschiedliche Distanzen am Beispiel der schnellen Bahn. Zur besseren Übersicht ist nur der Verlauf der Last dargestellt.

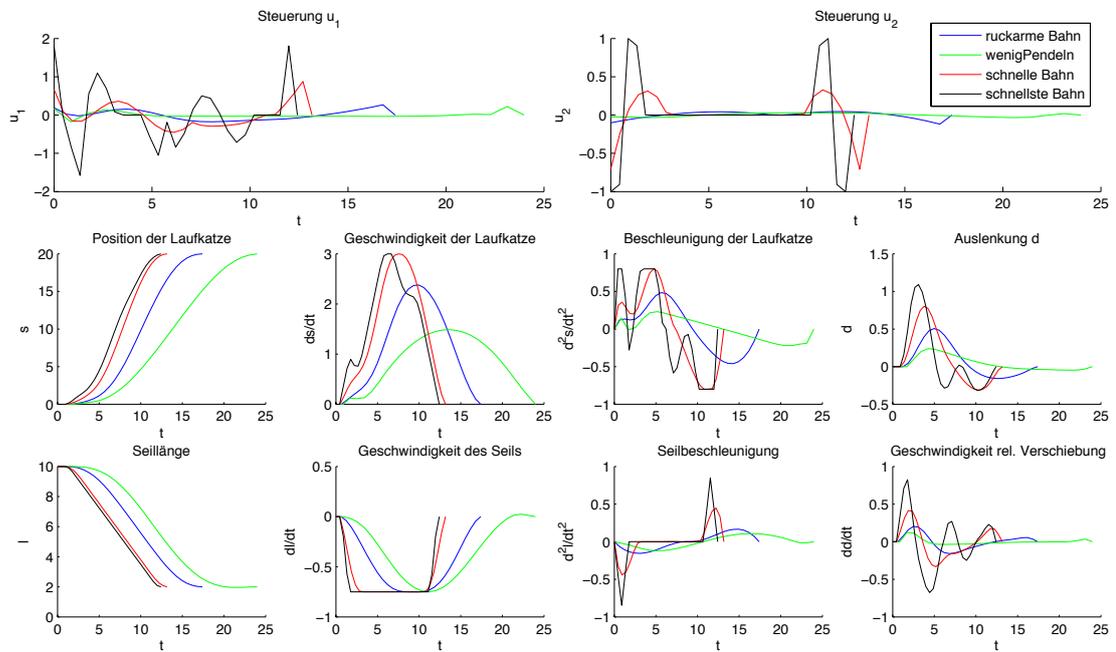
### 5.2.3 Komplexere Trajektorie und Notstopp

Wie sich die Steuerungen und Zustände verhalten, um eine Last 20 Meter weit zu fahren und 8 Meter einzuziehen, wird in Abbildung 5.8 für die vier Fälle gezeigt. In der Seitenansicht 5.7 ist deutlich zu sehen, wie die schnellen Bahnen das Schwingungsverhalten nutzen. Beim Abbremsen des Wagens schwingt die Last in Richtung des Zielpunktes. Die Laufkatze wird so gesteuert, dass in dem Moment, wenn die Last durch das Ausschwenken den Zielpunkt erreicht, der Wagen genau über der Last zum Stillstand kommt. Bei der schnellsten Bahn kann die Last auf diese Weise ihr Ziel in 10 Sekunden und bei der schnellen Bahn in 14 Sekunden schwingungsfrei erreichen. Dabei wird die maximal erlaubte Geschwindigkeit der Laufkatze für kurze Zeit angenommen. Für die verstärkte Minimierung des Rucks, beziehungsweise des Schwingungsverhaltens, wird ungefähr doppelt so viel Zeit gebraucht. Dafür sind die Steuerungen und somit auch die Zustandsgrößen deutlich kleiner und bilden so eine schonende und energiesparende Alternative zu den schnellen Varianten.

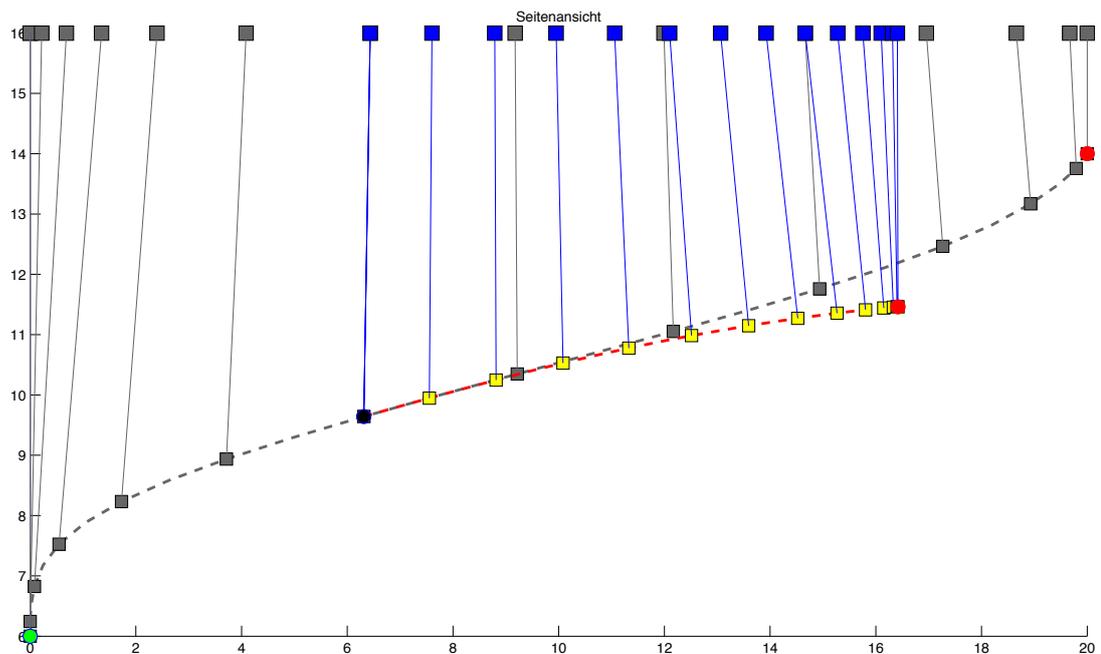


**Abbildung 5.7:** Seitenansicht der unterschiedlich gewichteten Bahntrajektorien für eine weite, schräge Strecke.

In Abbildung 5.9 ist ein Notstopp für die schnelle Bahn simuliert. Da der Notstopp bei hoher Geschwindigkeit eingeleitet wird, kommt das System erst nach 5,5 Sekunden zum Stillstand und legt dabei weitere 10 Meter zurück. Eine Bahnplanung, die zu jedem Zeitpunkt einen möglichst schnellen Notstopp ermöglicht, wird von Knauer [1] betrachtet.



**Abbildung 5.8:** Steuerungen und Zustände für eine weite, schräge Strecke von  $s_0 = 0$ ,  $l_0 = 10$  nach  $s_f = 20$ ,  $l_f = 2$ .



**Abbildung 5.9:** Seitenansicht für einen Notstopp der schnellen Bahn für eine weite, schräge Strecke. Der schwarze Punkt markiert den Startpunkt des Bremsmanövers.

### 5.2.4 Maximaler Ausschlagwinkel

Da das Modell (2.5) nur für kleine Winkel gültig ist, soll hier untersucht werden, von welcher Größenordnung der maximale Ausschlagwinkel für die vier Fälle ist und ob für diese Winkel die in der Modellherleitung gemachten Vereinfachungen zulässig sind.

Das Histogramm 5.10 zeigt für 100 zufällige Fahrten die Häufigkeitsverteilung des maximal erreichten Winkels für die ruckarme, pendelarme, schnelle und schnellste Bahn. Für die primäre Minimierung des Schwingungsverhaltens beträgt in 99 Fällen

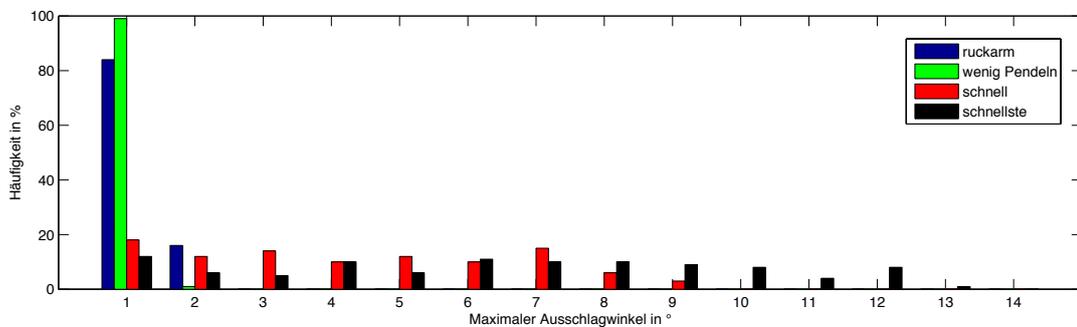


Abbildung 5.10: Histogramm zum maximalen Winkel für 100 zufällige Fahrten.

der maximale Ausschlagwinkel nur ungefähr  $1^\circ$ . Für die ruckarme Bahn liegen 84% in diesem Bereich. Nur die schnelle und die schnellste Bahn erreichen Winkel über  $2^\circ$ . Bei der schnellen Bahn schwingt die Last um maximal  $9^\circ$  aus. Als obere Grenze kann der durch die schnellste Bahn erzielte Winkel von  $13^\circ$  verwendet werden.

Für  $\varphi = 13^\circ$  weicht die, in der Modellierung verwendete, Taylor-Approximation  $\ddot{l}$  2,6% von  $\ddot{l} \cos \varphi$  ab. Für  $\varphi = 9^\circ$  beträgt die Abweichung nur 1,2%.

Für diese kleinen Winkel ist auch die Vereinfachung einer geradlinigen Bewegung statt der Bewegung entlang der Kreisbahn gerechtfertigt.

# Kapitel 6

## Zusammenfassung und Ausblick

In dieser Arbeit wurde die optimale Bahnplanung von Laufkatzen in einem Hochregallager untersucht. Dazu wurde in Kapitel 2 das dynamische Verhalten der Laufkatzen modelliert und in Kapitel 4 das optimale Steuerungsproblem der Laufkatze formuliert. Durch Diskretisierung ließ sich dieses Problem auf ein nichtlineares Optimierungsproblem reduzieren, welches in Kapitel 5 numerisch ausgewertet und anhand von ausgewählten Fahrten erfolgreich auf Plausibilität getestet wurde. Das verwendete numerische Verfahren nutzt die in Kapitel 3 kurz erläuterten Grundlagen der Optimierung. Allerdings zeigt die Auswertung auch, dass für einen Notstopp viel Zeit gebraucht wird. Wie dieses Problem behoben werden kann, wird von Knauer [1] gezeigt.

Sollen mehrere Laufkatzen zum Einsatz kommen, muss auf Kollisionsfreiheit geachtet werden. Diese Problematik wird ebenfalls von Knauer [1] behandelt. Darüber hinaus muss bestimmt werden, welche Laufkatze welchen Auftrag bekommt. Dies führt auf ein ganzzahliges Optimierungsproblem.

Das hier vorgestellte Modell der Laufkatze berücksichtigt nur eine horizontale und eine vertikale Bewegung. Möglich ist aber auch die Erweiterung um eine zusätzliche horizontale Komponente, indem die Fahrschienen senkrecht zu der Bewegungsrichtung der Laufkatze bewegt werden. So lassen sich die Laufkatzen noch flexibler durch die gesamte Halle steuern. Ergänzt man das Modell um eine Rotationsbewegung der Schienen, kann das Steuerungsproblem auf die klassische Verwendung von Laufkatzen, den Kran, übertragen werden. Diese könnten so automatisiert und beispielsweise als Großraumroboter eingesetzt werden.

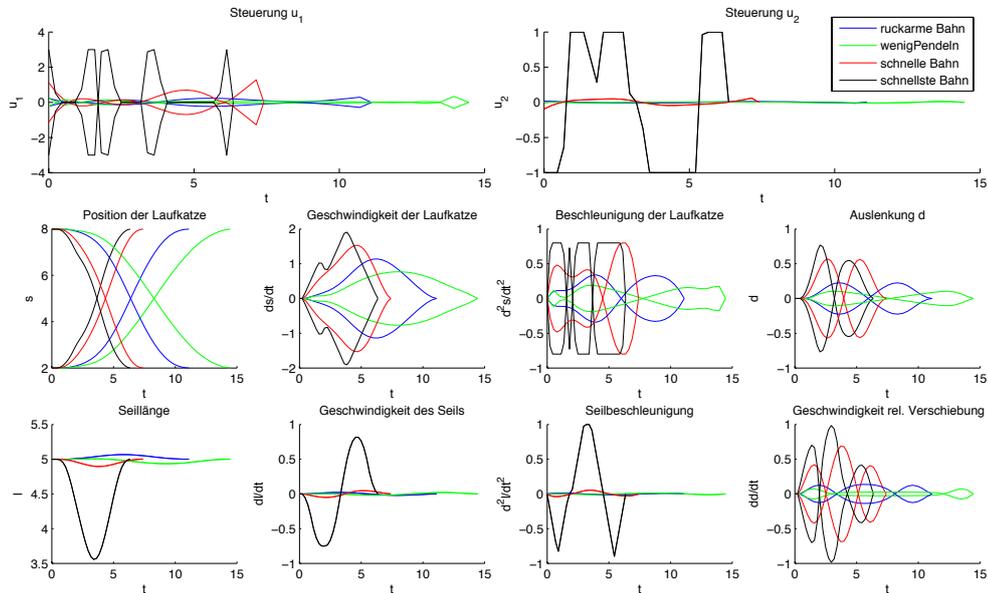
# Anhang A

## Nachweise zum Plausibilitätstest

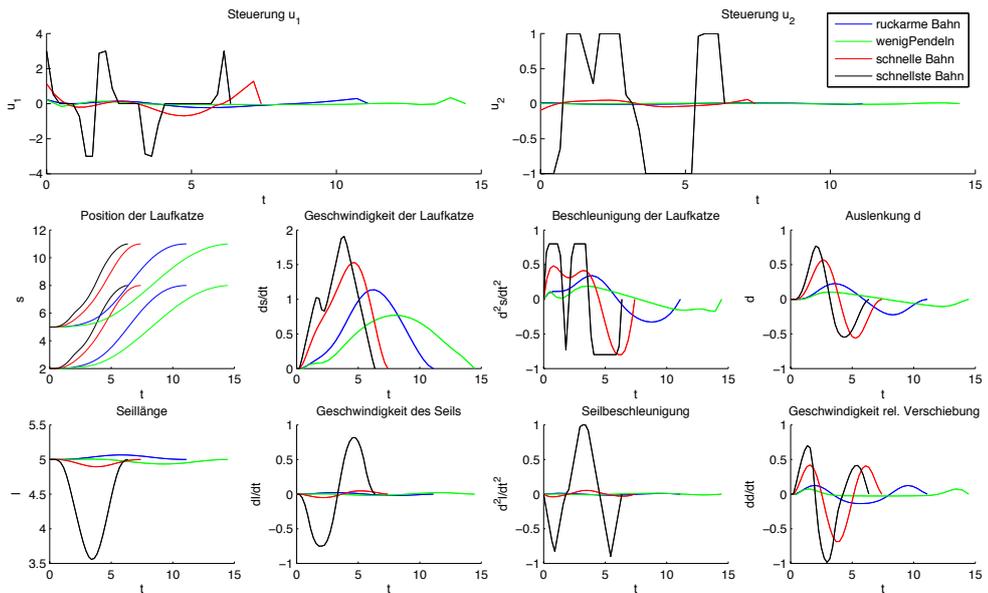
Zur Bestätigung der positiven Ergebnisse des Plausibilitätstest folgen hier die Werte für die Verfahrszeiten sowie die Diagramme zu den Tests 3.1, 3.2 und 3.3. Die Diagramme [A.1](#) und [A.2](#) belegen, dass die Steuerungen ebenfalls die Erwartungen aus [Tabelle 5.4](#) und [5.5](#) erfüllen.

Test-Nr.	Verfahrszeiten in Sekunden			
	ruckarm	wenig Pendeln	schnell	schnellste
1	0	0	0	0
2.1	9,1	7,4	6,4	6,0
2.2	9,5	8,3	7,4	7,1
2.3	9,1	7,4	6,4	6,0
2.4	12,9	11,5	10,4	10,0
3.1	11,1	14,5	7,4	6,4
3.2	11,1	14,5	7,4	6,4
3.3	11,1	14,5	7,4	6,4
3.4	11,4	14,9	8,6	7,3
3.5	12,4	15,2	9,5	8,0
3.6	17,6	26,0	13,8	13,0
3.7	11,9	14,5	9,9	8,6
3.8	11,9	14,5	10,2	9,7

**Tabelle A.1:** Ergebnisse des Plausibilitätstests: Verfahrszeiten für die vier unterschiedlichen Bahnkurven.



**Abbildung A.1:** Test-Nr. 3.1 und 3.2 für die vier Fälle. Da bei Test-Nr. 3.2 die Start- und Endpositionen vertauscht sind, ist  $u_1$  an der Geraden  $u_1 = 0$  gespiegelt. Dies wirkt sich auch auf die, von der Position der Laufkatze, abhängigen Zustände aus. Die Seilsteuerung bleibt wie bei Test 3.1, da Start- und Endpunkt auf gleicher Höhe liegen und so die Vertauschung keine Auswirkung hat.



**Abbildung A.2:** Test-Nr. 3.1 und die verschobenen Bahnen aus Test 3.3 für die vier Fälle. Die Verschiebung hat keine Auswirkungen auf die Steuerungen.

# Anhang B

## Programm

Mit diesem Skript lassen sich die optimalen Lösungen in AMPL mit dem Solver WORHP berechnen. Dazu muss

```
ampl run.txt
```

in der Kommandozeile eingegeben werden.

Die vollständigen Skripte sowie eine ausführlichere Dokumentation sind auf der beigelegten CD zu finden.

```
run.txt
```

```
1 # Reset , um eventuell frühere Daten zu löschen
2 reset ;
3
4 # Verwende WORHP als Solver
5 option solver worhp ;
6
7 # Stützstellen gehen von 0 ,... ,N
8 param N:=28;
9
10 # Gewichte
11 param w1:=0.005;
12 param w2:=0.001;
13 param w3:=1;
14 param w4:=0;
15
16 # Lade die Beschränkungen aus der externen Datei
17 include ./beschraenkungen.txt ;
18
19 # Das Modell ist ebenfalls ausgegliedert. Dieses beschreibt das
    Optimierungsproblem
20 model ./modell.mod;
```

```
21 |
22 | # Daten
23 | param s0:=2;
24 | param l0:=5;
25 | param sf:=8;
26 | param lf:=5;
27 |
28 | # Lade die Startschätzungen
29 | include ./Startwerte/letu2x8tIversion2.txt;
30 |
31 | # Löse das Problem:
32 | solve;
```

beschraenkungen.txt

```
1 | # Beschränkungen
2 | # Steuerbeschränkungen
3 | param u1min:=-3;
4 | param u1max:=3;
5 |
6 | param u2min:=-1;
7 | param u2max:=1;
8 |
9 | # Zustandsbeschränkungen
10 | # Position der Katze
11 | param smin:=-0.2;
12 | param smax:=30;
13 |
14 | # Länge des Seils
15 | param lmin:=0.01;
16 | param lmax:=16;
17 |
18 | # erlaubte Geschwindigkeit der Katze
19 | param dotsmin:=-3;
20 | param dotsmax:=3;
21 |
22 | # erlaubte Hub/Senk-Geschwindigkeit des Seils
23 | param dotlmin:=-0.75;
24 | param dotlmax:=1;
25 |
26 | # erlaubte Beschleunigung der Katze
27 | param ddotsmin:=-0.8;
28 | param ddotsmax:=0.8;
29 |
30 | # erlaubte Beschleunigung des Seils
31 | param ddotlmin:=-1;
32 | param ddotlmax:=1;
```

model.mod

```
1 # Konstanten
2 param g:=9.81; #Erdbeschleunigung
3 param h:=1/N; #Schrittweite
4 param dim:=8; #Anzahl an Zustandsvariablen
5
6 # Variablen
7 # u=(u1,u2) ist die Steuerung.
8 # Dabei entspricht u[2,0] u_2(0) und
9 # u[1,N]=u_1(tf) wobei tf die Endzeit ist.
10 var u{i in 1..2, j in 0..N};
11
12 # Zustandsvariablen
13 var x{i in 1..dim, j in 0..N};
14
15 # Verfahrzeit
16 var tf;
17
18 # Stammfunktion
19 var I{j in 0..N};
20
21 # Zielfunktion
22 # Minimiere die gewichtete Kombination der Zielfunktionale
23 minimize zielfunktion:
24   tf * w1 + I[N];
25
26 # Anfangsbedingungen
27 subject to initial1:
28   x[1,0] = s0;
29 subject to initial2:
30   x[2,0] = 0;
31 subject to initial3:
32   x[3,0] = 0;
33 subject to initial4:
34   x[4,0] = 0;
35 subject to initial5:
36   x[5,0] = 0;
37 subject to initial6:
38   x[6,0] = 10;
39 subject to initial7:
40   x[7,0] = 0;
41 subject to initial8:
42   x[8,0] = 0;
43
44 subject to initialI:
45   I[0] = 0;
46
47 # Zustandsgleichungen
48 # Umgesetzt mit dem expliziten Euler-Verfahren
```

```

49 # y'=f(t,y)
50 # t_k=t_0+k*h
51 # y_(k+1) = y_k + h * f( t_k , y_k )
52
53 subject to ODE1{j in 1..N}:
54 x[1,j] - x[1,j-1] = tf*h*x[2,j-1];
55
56 subject to ODE2{j in 1..N}:
57 x[2,j] - x[2,j-1] = tf*h*x[5,j-1];
58
59 subject to ODE3{j in 1..N}:
60 x[3,j] - x[3,j-1] = tf*h*x[4,j-1];
61
62 subject to ODE4{j in 1..N}:
63 x[4,j] - x[4,j-1] = tf*h*(x[5,j-1]-(g-x[8,j-1])*x[3,j-1]/x[6,j-1]);
64
65 subject to ODE5{j in 1..N}:
66 x[5,j] - x[5,j-1] = tf*h*u[1,j-1];
67
68 subject to ODE6{j in 1..N}:
69 x[6,j] - x[6,j-1] = tf*h*x[7,j-1];
70
71 subject to ODE7{j in 1..N}:
72 x[7,j] - x[7,j-1] = tf*h*x[8,j-1];
73
74 subject to ODE8{j in 1..N}:
75 x[8,j] - x[8,j-1] = tf*h*u[2,j-1];
76
77 # zusätzliche Differentialgleichung
78 subject to Stammfunktion{j in 1..N}:
79 I[j] - I[j-1] = tf*h*(
80     w2*(u[1,j-1]^2+u[2,j-1]^2)
81     + w3*(x[3,j-1]/x[6,j-1])^2
82     + w4*(x[4,j-1]/x[6,j-1])^2
83     ) ;
84
85 # Endbedingungen
86 subject to final1:
87 x[1,N] = sf;
88 subject to final2:
89 x[2,N] = 0;
90 subject to final3:
91 x[3,N] = 0;
92 subject to final4:
93 x[4,N] = 0;
94 subject to final5:
95 x[5,N] = 0;
96 subject to final6:
97 x[6,N] = lf;
98 subject to final7:

```

```

99 | x[7,N] = 0;
100 | subject to final8:
101 | x[8,N] = 0;
102 |
103 | # Steuerbeschränkungen
104 | subject to steuerung1{j in 0..N}:
105 | u1min <= u[1,j] <= u1max;
106 | subject to steuerung2{j in 0..N}:
107 | u2min <= u[2,j] <= u2max;
108 |
109 | # Zustandsbeschränkungen
110 | subject to pos{j in 0..N}:
111 | smin <= x[1,j] <= smax;
112 | subject to laenge{j in 0..N}:
113 | lmin <= x[6,j] <= lmax;
114 | subject to velocity{j in 0..N}:
115 | dotsmin<=x[2,j] <=dotsmax;
116 | subject to acceleration{j in 0..N}:
117 | ddotsmin <= x[5,j] <= ddotsmax;
118 | subject to seilgeschwindigkeit{j in 0..N}:
119 | dotlmin <= x[7,j] <= dotlmax;
120 | subject to seilbeschleunigung{j in 0..N}:
121 | ddotlmin<=x[8,j] <=ddotlmax;
122 | subject to time:
123 | tf >=0;

```

letu2x8tIversion2.txt

```

1 | # Startschätzungen
2 | # setzt:
3 | # u1,u2, x1,...,x8, t, I
4 | #
5 | # x1 und x6 werden linear interpoliert.
6 | # Die Zeit wird auf  $2*\sqrt{((sf-s0)/dotsmax)^2+((lf-10)/dotlmax)^2}$ 
   | geschätzt
7 | # Alle anderen Größen werden auf 0 gesetzt
8 |
9 | #setze Zeit
10 | let tf:= $2*\sqrt{((sf-s0)/dotsmax)^2+((lf-10)/dotlmax)^2}$ ;
11 |
12 | # setze x1 und x6
13 | for {j in 0..N}{
14 |   let x[1,j]:=(sf-s0)/N*j+s0; # lineare Interpolation des Ortes
15 |   let x[6,j]:=(lf-10)/N*j+10; # lineare Interpolation der Seillänge
16 | }
17 |
18 | #setze x_i=0 für i=2,3,4,5,7,8
19 | set S:={2,3,4,5,7,8};
20 | let {i in S, j in 0..N} x[i,j]:=0;

```

```
21 |
22 | #setze u1=0, u2=0
23 | let {i in 1..2, j in 0..N} u[i,j]:=0;
24 |
25 | #setze I=0
26 | let {j in 0..N} I[j]:=0;
```

# Literaturverzeichnis

- [1] Matthias Knauer, *Bilevel-Optimalsteuerung mittels hybrider Lösungsmethoden am Beispiel eines deckengeführten Regalbediengerätes in einem Hochregallager*, Dissertation, Bremen, 2009.
- [2] Hans Josef Pesch, *Schlüsseltechnologie Mathematik, Einblicke in aktuelle Anwendungen der Mathematik*. Teubner Verlag.
- [3] Dieter Jungnickel, *Optimierungsmethoden, Eine Einführung*. Springer Verlag, 2. Auflage, Augsburg, 2008.
- [4] Grégoire Allaire *Numerical Analysis and Optimization - An Introduction to Mathematical Modelling and Numerical Simulation*. Oxford University Press, 1. Auflage, New York, 2007.
- [5] R. Fletcher. *Practical Methods of Optimization*. Wiley, Braunschweig, 1987.
- [6] Matthias Knauer, *Sensitivitätsanalyse verschiedener Gütekriterien bei der optimalen Bahnplanung von Industrierobotern*. Diplomarbeit, Universität Bayreuth, 2001.
- [7] Tim Nikolayzik, Christof Büskens, Matthias Gerdts: Nonlinear large-scale Optimization with WORHP- Report 10-08, Zentrum für Technomathematik Universität Bremen, Dezember 2010. Elektronisch erhältlich unter <http://www.math.uni-bremen.de/zetem/o2c/publikationen>.
- [8] Christof Büskens, Patrik Kalmbach, Tim Nikolayzik, Dennis Wassel: Solver Development Strategy- Report 08-02, Zentrum für Technomathematik Universität Bremen, März 2008. Elektronisch erhältlich unter <http://www.math.uni-bremen.de/zetem/o2c/publikationen>.
- [9] Wikipedia. Stand 12.07.2011 <http://de.wikipedia.org/wiki/Sqp>.
- [10] Wikipedia. Stand 28.07.2011 <http://de.wikipedia.org/wiki/Innere-Punkte-Verfahren>.
- [11] O. Güler *Foundations of Optimization*. Springer Verlag, 1. Auflage, New York, 2010.

- [12] J. F. Bonnans, J. C. Gilbert, C. Lemarechal, C. A. Sagastizabal *Numerical Optimization*. Springer Verlag, 2. Auflage, Berlin Heidelberg, 2006.
- [13] R. Fourer, D. M. Gay, B. W. Kernighan *AMPL: A Modeling Language for Mathematical Programming*. Brooks/Cole Publishing Company / Cengage Learning, 2. Auflage, 2002.



# Eidesstattliche Erklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst und ausschließlich die angegebenen Quellen und Hilfsmittel verwendet habe.

Bremen, den 09. August 2011