

**Friedrich-Alexander-Universität Erlangen-Nürnberg**



**Lehrstuhl für Informationstechnik  
(Schwerpunkt Kommunikationselektronik)**

**LIKE**

Masterarbeit mit dem Thema:

**Modellfehler in optimierungsbasierter kombinierter  
Planung und Regelung für Rennwagen**

Bearbeiter	Weller Sebastian
Matrikelnr.	21777345
Studiengang	Informations und Kommunikationstechnik
Betreuer	Prof. Dr.-Ing. Jörn Thielecke Henrik Bey, M. Sc.
Beginn	08. Januar 2018
Ende	08. Juli 2018

---

# Bestätigung

---

## Erklärung:

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und, dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Erlangen, den (hier Datum eintragen)\_\_\_\_\_

---

# **Danksagung**

---

Ich möchte mich bei meinen Betreuern und meiner Familie bedanken.....

---

# Thema und Aufgabenstellung

---

## **Thema:**

Modellfehler in optimierungsbasierter kombinierter Planung und Regelung für Rennwagen

## **Aufgabenstellung:**

Am Lehrstuhl für Informationstechnik mit dem Schwerpunkt Kommunikationselektronik (LIKE)

Die Automatisierung des Fahrens schließt sowohl die Planung als auch die Regelung des Fahrzeugs mit ein. Häufig werden beide Bestandteile hierarchisch voneinander getrennt. Dies ist sinnvoll, solange das kontrollierte Fahrzeug sicher innerhalb der Aktuatorlimitierungen betrieben werden soll, oder wenn die Trennung bereits durch die Problemstellung gegeben ist (Zieltrajektorie bereits vorgegeben) [williams2016aggressive].

In anderen Fällen, z.B. wenn die gewünschte Dynamik wie in einer Rennsituation im Grenzbereich liegt, bietet sich eine kombinierte Planung und Regelung an. In diesem Beispiel würde die Kostenfunktion eine Minimierung der Rundenzeit beinhalten, während gleichzeitig die Beschränkungen des Fahrzeugs berücksichtigt werden.

Für derartige Probleme ist die modellprädiktive Regelung (MPC) bzw. eines ihrer Derivate besonders geeignet. Dabei kommt es immer zu einem sogenannten Modellfehler, der von der Komplexität und Genauigkeit des verwendeten Modells abhängt.

Das Ziel dieser Arbeit ist es, den Abfall bei der Leistung des Regelungsansatzes durch den Modellfehler zu untersuchen. Dafür soll eine Simulation verwendet werden.

- Auswahl einer passenden Simulationsumgebung und deren Inbetriebnahme
- Implementierung verschiedener (gegebener) Modelle für die Simulation
- Implementierung des MPC-Ansatzes

- Entwicklung einer einfachen Evaluationsmethode um die Leistungsfähigkeit des Reglers zu untersuchen
- Vergleich verschiedener Kombinationen aus Regler- und Simulationsmodellen

---

# Kurzzusammenfassung

---

Mit zunehmender Rechenleistung und Erfahrung der Automobilbranche mit autonomen Fahrzeugen rückt auch das Thema der selbstfahrenden Rennautos immer mehr in den Fokus. Das ROBORACE Projekt ist hier Vorreiter mit seiner ausgefeilten Hardwareplattform und dem bereits in öffentlichen Events gezeigten Fahrleistungen. Auch die Formula Student (FS) verschließt sich nicht vor dem Trend und hat 2017 die Rubrik Driverless ins Leben gerufen.

Diese Masterarbeit beschreibt einen Ansatz zur Echtzeitregelung und Trajektionsplanung für ein eben solches Driverless-Racecar. Die Basis hierfür ist ein Model Predictive Control (MPC) Algorithmus. Er vereint die Regelung und Trajektionsplanung und ist sehr adaptiv bezüglich verschiedener Fahrsituationen und Ziele. Als Ausgangssituation wird angenommen, dass das Fahrzeug bereits eine Runde auf einem unbekannten Kurs absolviert und nun eine genaue Karte des Kurses errechnet hat. Um das MPC nutzen zu können muss ein Fahrzeugmodell hinterlegt werden. Je genauer dieses ist, desto näher kann die Regelung an die Grenzen des realen Fahrzeuges gehen. Neben der Auslegung für das aktuellste FS-Fahrzeug des High Octane Motorsports für die Driverless Umrüstung, wird untersucht ab welchem Punkt ein kinematisches Modell nicht mehr ausreicht um das Fahrzeug sicher auf dem Rennkurs zu führen.

---

# Abstract

---

.....the Abstract is here..... **Bitte nicht löschen oder auskommentieren - ist obligatorisch!**

---

# Inhaltsverzeichnis

---

<b>1</b>	<b>Autonomes Fahren</b>	<b>1</b>
1.1	Formula Student Driverless . . . . .	1
<b>2</b>	<b>Grundlagen</b>	<b>3</b>
2.1	Fahrzeugmodelle . . . . .	3
2.2	Model Predictive Control . . . . .	3
2.3	Simulation . . . . .	3
<b>3</b>	<b>Implementierung von xy</b>	<b>4</b>
3.1	Gliederung . . . . .	4
3.2	Text formatieren . . . . .	4
3.3	Umbrüche . . . . .	5
3.4	Schriftart formatieren . . . . .	6
3.4.1	Default Fonts . . . . .	7
<b>4</b>	<b>Konzept</b>	<b>8</b>
<b>5</b>	<b>Ergebnisse</b>	<b>9</b>
<b>6</b>	<b>Diskussion</b>	<b>10</b>
<b>7</b>	<b>Zusammenfassung</b>	<b>11</b>
<b>A</b>	<b>Anhang</b>	<b>12</b>
	<b>Abkürzungsverzeichnis</b>	<b>13</b>
	<b>Literaturverzeichnis</b>	<b>14</b>



---

# 1 Autonomes Fahren

---

Gegen Ende des Jahres 2018, also nur wenigen Monaten nachdem diese Zeilen verfasst werden, wird Alphabets Tochterunternehmen Waymo die ersten voll autonomen (Level 5) Fahrzeuge als Taxis in Phoenix in Betrieb nehmen [?]. Dieser historische Moment wird eine Flut an neuen Assistenzsystemen nach sich ziehen die in wenigen Jahren das Straßenbild verändern werden.

Während sich die traditionellen Automobilhersteller aktuell auf Assistenzsysteme im Bereich von Level zwei und drei konzentrieren, wollen Unternehmen wie Google, GM und Uber alle Zwischenschritte überspringen und direkt voll autonom fahren um fahrerlose Taxiflotten aufzubauen. Dies ist nach aktuellem Stand der Technik jedoch nur kostengünstig und mit Rechnerplattformen welche in bestehende Fahrzeugkonzepte integrierbar sind realisierbar wenn die Fahrzeuge auf eine genaue Umgebungskarte zurückgreifen können. Erst mit Echtzeitupdates und bei einer Genauigkeit von  $\pm 10$  cm des Kartenmaterials können Fahrzeuge auch im innerstädtischen Verkehr autonom fahren [SH16]. Nicht nur im normalen Straßenverkehr ist ein möglichst genaues Wissen der Umgebung von Vorteil, auch bei autonomen Rennserien spielt die Kenntnis über den Rennkurs eine große Rolle. Kann das Fahrzeug nur auf Sicht fahren muss die Geschwindigkeit so gewählt werden dass das innerhalb der Sichtweite angehalten werden kann. Außerdem ist das Berechnen einer idealen Trajektorie nur möglich wenn die gesamte Strecke bekannt ist.

## 1.1 Formula Student Driverless

Die Formula Student ist ein Ingenieurswettbewerb für Studenten. Er hat seine Wurzeln in den USA im Jahre 1981 und wurde ab 1998 auch in Europa ausgetragen. Dass der Wettbewerb sehr erfolgreich ist, machen nicht nur die inzwischen fast 700 Teams weltweit [FsW] deutlich, sondern auch die Anzahl der verschiedenen Events die überall auf der Welt im Sommer stattfinden. Seit dem Jahr 2017 gibt es neben der ursprünglichen Combustion-Klasse und der vor 10 Jahren eingeführten Electric-Klasse auch noch die Driverless-Klasse. In dieser wird von den Teams ein Altfahrzeug um ein Sensorsystem sowie Aktoren so erweitert, dass das Rennauto die Kurse autonom bestreiten kann. Der Wettbewerb ist unterteilt in Dynamische und Statische Events. In letzteren werden verschiedene Präsentationen von den Teams verlangt. Diese beziehen sich auf die

technische Realisierung, Softwaredesign, Kostenaufstellung und ein Businessplan in dem die Teams ein Konzept erstellen müssen wie man das gebaute Fahrzeug über eine kleine Massenfertigung gewinnbringend verkaufen kann.

Die dynamischen Disziplinen, in denen das Fahrzeug selbstständig fährt, sind hierbei:

- Acceleration  
75 Meter Beschleunigungsstreifen. Punkte werden nach der Zeit nicht nach der Endgeschwindigkeit vergeben.
- Skidpad  
eine liegende 8 bei an der Engstelle eingefahren wird und jeweils 2 rechte und 2 linke Runden gefahren werden. Jeweils die zweite Runde wird gezählt. Die Abmaße sind exakt vorgegeben.
- Trackdrive  
ein bis zu 800 Meter langer Kurs mit maximal 80 Meter langen geraden und Kurven mit minimalem Innenradius von 9 Metern. Es werden 11 runden gefahren und die Teams erhalten im Vorherein keine Möglichkeit Messungen am Kurs vorzunehmen.

Für diese Arbeit ist vorallem der Trackdrive von Interesse. Es wird davon ausgegangen dass das Fahrzeug bereits die erste Runde absolviert, und sich damit eine genaue Karte des Rennkurses erstellt hat. Die Messungen und Vergleiche beziehen sich damit auch immer auf einen Kurs der so (wenn auch nicht so lang) in einem FS-Event für die Driverless Fahrzeuge vorkommen könnte.

## 1.2

---

## **2 Grundlagen**

---

### **2.1 Fahrzeugmodelle**

### **2.2 Model Predictive Control**

### **2.3 Simulation**

---

## 3 Implementierung von xy

---

### 3.1 Gliederung

Befehle zum Gliedern von Texten:

Nummeriert und im InhV.:

```
\chapter{Kapitel – Ebene 1 (1 Einleitung)}  
\section{Abschnitt – Ebene 2 (1.1 Gliederung)}  
\subsection{Unterabschnitt – Ebene3 1.1.1....}  
\subsubsection{Unterunterabschnitt – Ebene 4 (nicht sinnvoll)}
```

#### nicht nummerierte Überschrift

Alles, was nicht ins Inhaltsverzeichnis soll ist mit einem \* versehen: -> steht für Kapitel, die nicht nummeriert sind und somit auch nicht im InhVz. stehen

Befehle:

```
\chapter*{Thema und Aufgabenstellung}  
\section* {Sektion/Abschnitt} unnummiert  
\subsection*{Unterabschnitt – Ebene3 }  
\subsubsection*{Unterunterabschnitt – Ebene 4}
```

### 3.2 Text formatieren

Folgende Befehlen erzeugen einen Zeilenumbruch:

(1ex entspricht der Höhe des Buchstaben „x“ in der aktuellen Schriftart)

```
\\  
\newline  
\\[1ex]
```

Folgende Befehle erzeugen einen neuen Absatz:

```
\par
\par\smallskip
\par\medskip
\par\bigskip
```

Folgende Befehle erzeugen einen zusätzlichen horizontalen Abstand:

<code>\,</code>	% kleiner Abstand
<code>\;</code>	% mittlerer Abstand
<code>\quad</code>	% großer Abstand
<code>\qquad</code>	% sehr großer Abstand
<code>\!</code>	% kleiner negativer Abstand

Folgende Befehle zum Seitenumbruch:

## 3.3 Umbrüche

Der Text wird normalerweise automatisch umbrochen. Zum expliziten Beenden einer Zeile benutzen Sie

```
\\[abstand] % bzw.
\\[2pt]
```

Beispiel

Wenn man zum Beispiel eine Überschrift

\\in zwei Zeilen umbrechen möchten,

bitte ein

```
\protect
```

benutzen. Zusätzlich kann an einer so beendeten Zeile noch gleichzeitig ein Seitenumbruch verhindert werden, indem man

```
\\*[abstand]
```

eingibt. Wenn man Latex einen Spielraum lassen will: Dafür dient der Befehl

```
\linebreak[zwang]
```

Für die Dringlichkeit des-Befehls können Sie die Zahlen von 0 bis 4 eintragen,wobei 0 eine milde Wunschaußerung darstellt und 4 das Zeilenende erzwingt. Gleichzeitig werden beim Beenden einer Zeile die Wörter gestreckt und dabei gleichmäßig überdie Zeile verteilt.

```
Der Zwerg nahm seine Axt und hieb mehrere Äste ab.\\
Der Zwerg nahm seine Axt und hieb mehrere Äste ab.\linebreak[4]
Der Zwerg nahm seine Axt und hieb mehrere Äste ab.\linebreak[2]
Der Zwerg nahm seine Axt und hieb mehrere Äste ab.\linebreak[3]
Sie banden sie mit Bogensehnen zusammen
```

Der Zwerg nahm seine Axt und hieb mehrere Äste ab.

Der Zwerg nahm seine Axt und hieb mehrere Äste ab.

Der Zwerg nahm seine Axt und hieb mehrere Äste ab. Der Zwerg nahm seine Axt und hieb mehrere Äste ab. Sie banden sie mit Bogensehnen zusammen

oder:

```
\nolinebreak[zwang]
```

## 3.4 Schriftart formatieren

Folgende Befehle für die Schriftgröße:

<code>\tiny</code>	<small>tiny</small>
<code>\scriptsize</code>	<small>scriptsize</small>
<code>\footnotesize</code>	<small>footnotesize</small>
<code>\small</code>	<small>small</small>
<code>\normalsize</code>	<small>normalsize</small>
<code>\large</code>	<b>large</b>
<code>\Large</code>	<b>Large</b>
<code>\LARGE</code>	<b>LARGE</b>
<code>\huge</code>	<b>huge</b>
<code>\Huge</code>	<b>Huge</b>

Folgende Befehle für die Schriftart :

<code>\textrm{}</code>	Textbeispiel
<code>\texttt{}</code>	Textbeispiel
<code>\textbf{}</code>	<b>Textbeispiel</b>
<code>\textit{}</code>	<i>Textbeispiel</i>
<code>\textsc{}</code>	TEXTBEISPIEL
<code>\textsl{}</code>	<i>Textbeispiel</i>

Folgende Befehle zum Hervorheben (Argument):

<code>\emph{}</code>	<i>Textbeispiel</i>
<code>\underline{}</code>	<u>Textbeispiel</u>

#### 3.4.1 Default Fonts

Für das gesamte Dokument wurde eine Schriftart im cls.-File generiert

mittes des Befehls->	<code>\addtokomafont{disposition}{\rmfamily}</code>
und dem Paket Times->	<code>\RequirePackage{times}</code>

---

## **4 Konzept**

---



---

## **5 Ergebnisse**

---

---

## **6 Diskussion**

---

---

## **7 Zusammenfassung und Ausblick**

---

---

# A Anhang

---

## Elektronischer Anhang

Inhalt:

1. Beispiel : Entwickeltes MATLAB-Programm
2. ....

Hier können weiterführende Grafiken, Codefragmente oder Ähnliches eingefügt werden.....

## Einbindung Grafik im Anhang



**Abbildung A1:** Unterschrift Bild x Die auf die Rotationsfrequenz des Innenzylinders normierten Eigenfrequenzen der gefundenen Grundmoden der Taylor-Strömung für  $h$  (Die azimutale Wellenzahl ist mit  $m$  bezeichnet.)

---

## **Abkürzungsverzeichnis**

---

---

# Literaturverzeichnis

---

- [FsW] Formula student - world ranking lists. <https://mazur-events.de/fs-world/>. Accessed on 2018-05-14.
- [SH16] Heiko G. Seif and Xiaolong Hu. Autonomous driving in the icity—hd maps as a key challenge of the automotive industry. *Engineering*, 2(2):159 – 162, 2016.

---

# Sebastian Weller

---

## Persönliche Daten

Adresse	An der Kühruh 13 96123 Litzendorf
Mobil	0170 - 9732890
Email	sebastian.weller01@gmail.com
Geburtsdatum	01.04.1992
Staatsangehörigkeit	deutsch

## Studium und Schulbildung

01/2013 - 07/2018	Friedrich-Alexander-Universität Erlangen-Nürnberg Studium: Informations und Kommunikationstechnik
01/2011 - 01/2013	Ohm-Fachhochschule Nürnberg Studium: Elektrotechnik

## Berufliche Erfahrungen / Praktika

01/2016 - 07/2016	Wissenschaftlicher Hilfsmitarbeiter am Fraunhofer IIS
01/2016 - 07/2016	Praktikum bei Siemens Erlangen

## Zusatzqualifikationen

Sprachen	Deutsch (Muttersprache) Englisch (fließend in Wort und Schrift)
Programmiersprachen	Java   C++   C Julia   Python

Erlangen, den (Datum eintragen)

---

Sebastian Weller