



# In der Regel haben wir einen zweizeiligen Bachelorthesistitel

## Bachelorarbeit

für die Prüfung zum  
**Bachelor of Engineering**

des Studiengangs Vorderasiatische Archäologie  
an der Dualen Hochschule Baden-Württemberg Karlsruhe

von

**Vorname Nachname**

August 2011

**Bearbeitungszeitraum**  
**Matrikelnummer, Kurs**  
**Ausbildungsfirma**  
**Betreuer**  
**Gutachter**

12 Wochen  
1234510, ABC2008DE  
Firma GmbH, Firmenort  
Dipl.-Ing. (FH) Peter Pan  
Dr. Silvana Koch-Mehrin

# Sperrvermerk

Der Inhalt dieser Arbeit darf weder als Ganzes noch in Auszügen Personen außerhalb des Prüfungsprozesses und des Evaluationsverfahrens zugänglich gemacht werden, sofern keine anderslautende Genehmigung der Ausbildungsstätte vorliegt.

Abgabeort, August 2011

---

Vorname Nachname

# Erklärung

Ich versichere hiermit, dass ich meine Bachelorarbeit mit dem Thema: *In der Regel haben wir einen zweizeiligen Bachelorthesistitel* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Abgabeort, August 2011

---

Vorname Nachname

## Abstract

Abstract normalerweise auf Englisch. Siehe: [http://www.dhbw.de/fileadmin/user/public/Dokumente/Portal/Richtlinien\\_Praxismodule\\_Studien\\_und\\_Bachelorarbeiten\\_JG2011ff.pdf](http://www.dhbw.de/fileadmin/user/public/Dokumente/Portal/Richtlinien_Praxismodule_Studien_und_Bachelorarbeiten_JG2011ff.pdf) (8.3.1 Inhaltsverzeichnis)

Ein „Abstract“ ist eine prägnante Inhaltsangabe, ein Abriss ohne Interpretation und Wertung einer wissenschaftlichen Arbeit. In DIN 1426 wird das (oder auch der) Abstract als Kurzreferat zur Inhaltsangabe beschrieben.

**Objektivität** soll sich jeder persönlichen Wertung enthalten

**Kürze** soll so kurz wie möglich sein

**Genauigkeit** soll genau die Inhalte und die Meinung der Originalarbeit wiedergeben

Üblicherweise müssen wissenschaftliche Artikel einen Abstract enthalten, typischerweise von 100-150 Wörtern, ohne Bilder und Literaturzitate und in einem Absatz.

Quelle: <http://de.wikipedia.org/wiki/Abstract> Abgerufen 07.07.2011

Diese etwa einseitige Zusammenfassung soll es dem Leser ermöglichen, Inhalt der Arbeit und Vorgehensweise des Autors rasch zu überblicken. Gegenstand des Abstract sind insbesondere

- Problemstellung der Arbeit,
- im Rahmen der Arbeit geprüfte Hypothesen bzw. beantwortete Fragen,
- der Analyse zugrunde liegende Methode,
- wesentliche, im Rahmen der Arbeit gewonnene Erkenntnisse,
- Einschränkungen des Gültigkeitsbereichs (der Erkenntnisse) sowie nicht beantwortete Fragen.

Quelle: [http://www.ib.dhbw-mannheim.de/fileadmin/ms/bwl-ib/Downloads\\_alt/Leitfaden\\_31.05.pdf](http://www.ib.dhbw-mannheim.de/fileadmin/ms/bwl-ib/Downloads_alt/Leitfaden_31.05.pdf), S. 49

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>V</b>
<b>Abbildungsverzeichnis</b>	<b>VI</b>
<b>Tabellenverzeichnis</b>	<b>VII</b>
<b>Listings</b>	<b>VIII</b>
<b>1 Aufgabe</b>	<b>1</b>
1.1 Carolocup, Regeln . . . . .	1
1.2 Simulation . . . . .	1
1.3 Anforderungen . . . . .	1
<b>2 Umfeld</b>	<b>2</b>
2.1 ROS . . . . .	2
2.2 Architektur . . . . .	2
2.3 Hardware . . . . .	2
<b>3 Stand der Technik</b>	<b>3</b>
3.1 Wie kann man Physik simulieren . . . . .	3
3.2 Was gibt Ros für Möglichkeiten . . . . .	3
<b>4 Implementierung</b>	<b>4</b>
4.1 Ausgangslage . . . . .	4
4.2 Simulation als Datenproduktion . . . . .	4
4.3 Kamera . . . . .	4
4.4 statische Fortbewegung . . . . .	4
4.5 dynamische Fortbewegung . . . . .	5
<b>5 Ergebnis</b>	<b>6</b>
5.1 Anforderungen erreicht? . . . . .	6
<b>Literatur</b>	<b>7</b>
<b>Glossar</b>	<b>8</b>
<b>Anhang</b>	<b>9</b>

# Abkürzungsverzeichnis

**ROS**      Robot Operating System  
**AGPL**      Affero GNU General Public License

# Abbildungsverzeichnis

1.1	Das Logo der Musterfirma . . . . .	1
-----	------------------------------------	---

# Tabellenverzeichnis



# Listings

# 1 Aufgabe

Erste Erwähnung eines Akronyms wird als Fußnote angezeigt. Jede weitere wird nur verlinkt: Affero GNU General Public License ([AGPL](#)). [3]

Verweise auf das Glossar: [Glossareintrag](#), [Glossareinträge](#)

Nur erwähnte Literaturverweise werden auch im Literaturverzeichnis gedruckt: [1], [2]

Meine erste Fußnote<sup>1</sup>

## 1.1 Carolocup, Regeln

## 1.2 Simulation

## 1.3 Anforderungen



Abbildung 1.1: Das Logo der Musterfirma<sup>1</sup>

---

<sup>1</sup> Ich bin eine Fußnote  
<sup>1</sup> aus [4]

## 2 Umfeld

### 2.1 ROS

Das Robot Operating System ([ROS](#)) ist ein Betriebssystem das speziell auf Roboter angepasst ist. Es läuft auf einem anderen Betriebssystem, in diesem Fall Ubuntu. Es stellt Funktionalitäten wie Hardware Abstraktion, Paket Management, Kommunikation zwischen Prozessen... zu Verfügung. Das Ziel von Robot Operating System ([ROS](#)) ist die Wiederverwendbarkeit von Code. Dies gelingt dadurch das Robot Operating System ([ROS](#)) ein verteiltes System an Prozessen (Nodes) ist. Die Nodes kommunizieren untereinander über Topics. Topics sind die Busse auf denen Nodes Nachrichten senden. Nachrichten zu senden oder zu empfangen ist anonym, dadurch sind Sender und Empfänger komplett entkoppelt voneinander. [6] Nodes lassen sich so austauschen und wiederverwenden. [5] Die Regelung des Fahrzeugs läuft auf einer ROS Architektur.

### 2.2 Architektur

Als Ausgangssituation für die Simulation wird die open source Simulationsumgebung des KITs genutzt. Diese ist eine Gazebo Simulation mit verschiedenen ROS Nodes. HIER ARCHITEKTUR BILD EINFÜGEN Die eigentliche Fahrsimulation des Fahrzeugs ist nicht Teil der Simulationsumgebung. Um das Fahrzeug zu bewegen erwartet die Simulation ein Topic auf dem neue Position gepublished werden. Da die Simulation aber den Lenkwinkel als Input nutzen soll, muss die Simulation dahingehend erweitert werden.

### 2.3 Hardware

# **3 Stand der Technik**

## **3.1 Wie kann man Physik simulieren**

## **3.2 Was gibt Ros für Möglichkeiten**

# 4 Implementierung

## 4.1 Ausgangslage

## 4.2 Simulation als Datenproduktion

Strecke modellieren Man kann in der Simulation Strecken aufbauen. Dazu wird auch zu jeder Strecke eine Ideallinie für das Fahrzeug generiert. Diese Strecke besteht aus einer Liste von Punkten. Das Modul AutomaticDrive kann genutzt werden damit das Fahrzeug entlang dieser Strecke fährt. Das Fahrzeug fährt dann aber in gerader Linie von Punkt zu Punkt. Dadurch kommt es in Kurven zu einem ruckeligen Fahrverhalten. WIE LÖST MAN DAS PROBLEM. INTERPOLATION IN KURVEN BEI AUTODRIVE

Kamera entlang der Strecke fahren lassen

Um Algorithmen und Model in der Spurerkennung und Situationsklassifizierung zu verifizieren und verbessern, benötigt der Auto drive noch eine Möglichkeit den Lenkwinkel auszugeben. Des Auto drive Modul platziert das Fahrzeug auf der Ideallinie so, dass HIER WAR ICH NICHT GANZ genau

## 4.3 Kamera

Auflösung, Position, FoV eingestellt.

Problem: Für die IPM umrechnung muss der Kamerafeed sehr genau den echten wieder spiegeln. Neue matrix wäre eine Scheiß Lösung.

## 4.4 statische Fortbewegung

Problem: Lenkwinkel muss irgendwie in eine Transformation des Fahrzeugs umgewandelt werden. Der genaue Lösung wäre eine dynamische Simulation der Räder und Lenkung. Dies war nicht so einfach abzustimmen, da sich die Simulation anders Verhalten hat als

das echte Fahrzeug. Vorläufige Lösung: Eine Rechnung die die Transformation abschätzt, aber sehr genau ist bei kleinen Lenkwinkel und hoher Fps.

Gegeben ist der Lenkwinkel  $\alpha$  Die Geschwindigkeit des Fahrzeugs  $v$  Der daraus resultierende Vektor Lenkrichtung  $\text{steering}$  Position des Fahrzeugs  $p$  Orientierung des Fahrzeugs und der daraus resultierende Vektor  $\text{direction}$

Als erstes wird aus dem Lenkwinkel  $\alpha$  der Vektor  $\text{steering}$  bestimmt. Danach wird die neue Position  $p_{\text{Neu}}$  bestimmt mit  $p_{\text{Neu}} = p + \text{steering} * (\text{deltaTime} * v)$  Die neue Orientierung des Fahrzeugs  $\text{direction}_{\text{Neu}}$  wird errechnet mit  $\text{direction}_{\text{Neu}} = \text{direction}$  rotieren um  $k * \alpha$  dabei ist  $k$  Abhängig von der gefahrenen Strecke, sowie der Fahrzeuglänge.

## 4.5 dynamische Fortbewegung

Vier drehende Räder, Antrieb, Lenkung vorne. Problem mit, Kräften, massen, Kollision...

Die Räder des Fahrzeugs werden definiert innerhalb einem Link. Jeder der vier Links besitzt eine visuelle Komponente und ein Kollisions-Komponente. Das visuelle Objekt und das Kollisions-Objekt stimmen überein, beide definieren den gleichen Zylinder. Jeder physikalisch simulierte Link benötigt einen inertial-Tag. Dieser definiert die Masse "mass" und die Massenverteilung als inertia Tensor  $\text{inertia}$ .

Damit sich ein Rad drehen kann, wird es mit einem Joint an dem Fahrzeug befestigt. Der Joint definiert die beiden Links die verbunden werden sollen: child und parent. Der Typ des Joints ist für die Räder "continuous". Das bedeutet, dass sich die Räder ohne Einschränkung um eine Achse drehen können. Der Joint definiert außerdem die Position und Orientierung der Achse.

DiskJoints.

BRAUCHT MAN TRANSMISSIONS ÜBERHAUPT, ODER REICHEN CONTROLLER AUF JOINTS? Um die Räder und die Lenkung zu steuern, benötigt es einen Motor. Für urdf gibt es hierfür Actuator und Transmissions. Eine Transmission verbindet einen Actuator mit einem Joint.

# **5 Ergebnis**

## **5.1 Anforderungen erreicht?**

# Literatur

- [1] Peter Baumgartner, Hartmut Häfele und Kornelia Maier-Häfele. *E-Learning Praxis-handbuch : Auswahl von Lernplattformen; Marktübersicht, Funktionen, Fachbegriffe*. Innsbruck: StudienVerl., 2002. ISBN: 3-7065-1771-X.
- [2] Stuard E. Dreyfus und Hubert L. Dreyfus. *A Five-Stage Model Of The Mental Activities Involved In Directed Skill Acquisition*. Techn. Ber. University of California, Berkley, 1980. URL: <http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA084551&Location=U2&doc=GetTRDoc.pdf>.
- [3] Free Software Foundation. *GNU AFFERO GENERAL PUBLIC LICENSE*. 2007. URL: <http://www.gnu.org/licenses/agpl-3.0.de.html>.
- [4] Max Mustermann. „tolles Musterthema“. Studienarbeit. Musterstadt, 2012.
- [5] Open Robotics. *ROS Introduction*. 2018. URL: <http://wiki.ros.org/ROS/Introduction>.
- [6] Open Robotics. *ROS Topics*. 2019. URL: <http://wiki.ros.org/Topics>.



# Glossar

## Glossareintrag

Ein Glossar beschreibt verschiedenste Dinge in kurzen Worten.

# Anhang

(Beispielhafter Anhang)

A. Assignment

B. List of CD Contents

C. CD

## B. List of CD Contents

└ <b>Literature/</b>	
└ <b>Citavi-Project(incl pdfs)/</b>	⇒ <i>Citavi (bibliography software) project with almost all found sources relating to this report.</i>
	<i>The PDFs linked to bibliography items therein are in the sub-directory ‘CitaviFiles’</i>
– bibliography.bib	⇒ <i>Exported Bibliography file with all sources</i>
– Studienarbeit.ctv4	⇒ <i>Citavi Project file</i>
└ <b>CitaviCovers/</b>	⇒ <i>Images of bibliography cover pages</i>
└ <b>CitaviFiles/</b>	⇒ <i>Cited and most other found PDF resources</i>
└ <b>eBooks/</b>	
└ <b>JournalArticles/</b>	
└ <b>Standards/</b>	
└ <b>Websites/</b>	
└ <b>Presentation/</b>	
– presentation.pptx	
– presentation.pdf	
└ <b>Report/</b>	
– Aufgabenstellung.pdf	
– Studienarbeit2.pdf	
└ <b>Latex-Files/</b>	⇒ <i>editable L<sup>A</sup>T<sub>E</sub>X files and other included files for this report</i>
└ <b>ads/</b>	⇒ <i>Front- and Backmatter</i>
└ <b>content/</b>	⇒ <i>Main part</i>
└ <b>images/</b>	⇒ <i>All used images</i>
└ <b>lang/</b>	⇒ <i>Language files for L<sup>A</sup>T<sub>E</sub>X template</i>