

UNIVERSITÄT DUISBURG-ESSEN  
Institut für Informatik und Wirtschaftsinformatik  
Lehrstuhl Pervasive Computing und User Interface Engineering  
Prof. Dr. Albrecht Schmidt

**Bachelorarbeit**

**Automatisierte Analyse und Vergleich von  
Fahreigenschaften zur Messung der  
Fahrerablenkung**

Marco Müller  
marco.mueller@stud.uni-due.de

Bearbeitungszeitraum: 16. 3. 2008 bis 16. 6. 2008  
Betreuer: Dagmar Kern  
Verantw. Hochschullehrer: Prof. Dr. Albrecht Schmidt

## **Zusammenfassung**

Ziel dieser Arbeit ist die automatisierte Analyse von Fahrdaten zur Messung der Fahrerablenkung. Dazu wird verwandte Literatur auf Einsatzfelder von Fahrsimulatoren und übliche Messwerte in Studien hin untersucht. Die benötigten Analysedaten werden identifiziert und die umzusetzenden Analysemethoden beschrieben. Ein Open-Source-Fahrsimulator wird für die Analyse angepasst und ein Analyseprogramm entwickelt, das die Analysemethoden umsetzt. In zwei Studien wird dann das Analyseprogramm evaluiert. Studienteilnehmer fahren in diesen Studien mit und ohne Nebenaufgaben in dem Fahrsimulator. Die Ablenkung der Teilnehmer bei den Nebenaufgaben während des Fahrens kann erfolgreich mit Hilfe des Analyseprogramms identifiziert werden. In der zweiten Studie kann durch das Analyseprogramm auch die unterschiedliche Ablenkung zweier verschieden komplexer Nebenaufgaben beim Fahren ermittelt werden.

## **Abstract**

The objective of this thesis is the automated analysis of driving data for the measuring of driver distraction. Related literature is examined for this purpose, with focus on the application of driving simulators and data commonly measured using them. After identifying the required analysis data the implemented analysis methods are described. An open source driving simulator is customised to meet the requirements of the analysis and a software implementing the identified analysis methods is developed. Later on, the analysis programme is evaluated by conducting two studies. During these, participants drive with and without secondary tasks in the simulator. The analysis programme successfully identifies the driver distraction caused by secondary tasks in the first study. In the second study the analysis programme successfully detects and assesses the differences in driver distraction caused by two secondary tasks of different complexity.

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig angefertigt, alle Zitate als solche kenntlich gemacht sowie alle benutzten Quellen und Hilfsmittel angegeben habe.

Essen, 14. Juni 2008

.....



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Vorgehen . . . . .	1
<b>2</b>	<b>Technologisches Umfeld</b>	<b>2</b>
2.1	Klassifizierung des Fahrtrainators . . . . .	2
2.2	Technologische Grundlagen . . . . .	2
2.3	Systemarchitektur . . . . .	3
2.3.1	Entitäten . . . . .	3
2.3.2	InputHandler . . . . .	4
2.3.3	ActionHandler . . . . .	7
2.3.4	Kartenmanagement . . . . .	7
2.3.5	Konfigurationsdateien . . . . .	7
2.3.6	Steuerungskomponente . . . . .	7
<b>3</b>	<b>Verwandte Arbeiten</b>	<b>8</b>
3.1	Fahrerbelastung als Funktion der StraÙengeometrie . . . . .	8
3.2	Vergleich zwischen Alkoholeinfluss und Telefonat . . . . .	9
3.3	Fahrtrainatoren in der Fahrertraining . . . . .	11
3.4	Vergleich einer realen Fahrt und Fahrtrainatoren . . . . .	13
3.5	Evaluation von Fahrerablenkung . . . . .	14
3.6	Vergleich der verwandten Arbeiten . . . . .	16
<b>4</b>	<b>Analyse der Simulationsdaten</b>	<b>17</b>
4.1	Ziele der Analyse . . . . .	17
4.2	Gespeicherte Fahrdaten . . . . .	19
4.3	Umsetzung der Analyse . . . . .	21
4.3.1	Mittlere Distanz zur Ideallinie . . . . .	21
4.3.2	Standardabweichung der mittleren Distanz zur Ideallinie . . . . .	22
4.3.3	Mittlere seitliche Geschwindigkeit . . . . .	22
4.3.4	Mittlere Log-Zeit bis zur Spurbegrenzung . . . . .	23
4.3.5	Mittlere Geschwindigkeit . . . . .	24
4.3.6	Standardabweichung der Geschwindigkeit . . . . .	24
4.3.7	Mittlere absolute Beschleunigung . . . . .	24
4.3.8	Standardabweichung der Lenkerstellung . . . . .	25
4.3.9	Frequenz der Steuerkorrektur . . . . .	25
4.3.10	Standardabweichung der Gaspedalstellung . . . . .	25
4.3.11	Frequenz der Geschwindigkeitskorrektur . . . . .	25
4.3.12	Zeit abseits der Fahrbahn . . . . .	25
4.3.13	Häufigkeit des Verlassens der Fahrbahn . . . . .	25
4.3.14	Quotient aus Spurhaltung und Geschwindigkeit . . . . .	26
4.4	Architektur . . . . .	26
4.4.1	View-Ebene . . . . .	26
4.4.2	Control-Ebene . . . . .	27
4.4.3	Model-Ebene . . . . .	28
4.5	Implementierung . . . . .	28

4.5.1	Beispielmodul „Standardabweichung der Distanz zur Ideallinie,, .	29
4.5.2	Lesemodule . . . . .	30
4.6	Erweiterungsmöglichkeiten . . . . .	31
4.6.1	Erweiterung der Statistiken . . . . .	31
4.6.2	Erweiterung der Datenquellen . . . . .	33
4.7	Technische Daten . . . . .	34
<b>5</b>	<b>Vorgehen einer Analyse</b>	<b>37</b>
<b>6</b>	<b>Evaluierung</b>	<b>41</b>
6.1	Studie zur Identifikation von Ablenkung . . . . .	42
6.2	Studie zur Bewertung von Ablenkung . . . . .	47
6.3	Zusammenfassung . . . . .	51
<b>7</b>	<b>Fazit</b>	<b>51</b>
	<b>Abbildungsverzeichnis</b>	<b>53</b>
	<b>Tabellenverzeichnis</b>	<b>54</b>
	<b>Literatur</b>	<b>55</b>
<b>A</b>	<b>Anhang</b>	<b>57</b>

# 1 Einleitung

## 1.1 Motivation

Fahrsimulatoren werden seit einiger Zeit mit zahlreichen und verschiedenen Zielen eingesetzt [23]. Beispiele für Studien, in denen Fahrsimulatoren eingesetzt werden, tragen zur Forschung der Akzeptanz von Verkehrslücken [14], Reduktion der Ablenkungen durch Telefonate [8] und zur Reduktion der optischen Ablenkung beim Fahren [18] bei. Für diese Zwecke werden viele verschiedene Fahrsimulatoren verwendet, die sich hauptsächlich hinsichtlich des Preises und des Anwendungszwecks unterscheiden [7].

In einem Projektseminar am Lehrstuhl Pervasive Computing und User Interface Engineering der Universität Duisburg-Essen wurde eine Software entwickelt mit dem Ziel, einen günstigen und variablen Fahrsimulator anbieten zu können. Die Software soll kostenlos unter einer Open Source-Lizenz vertrieben werden. Damit ist der Fahrsimulator schnell und kostengünstig einzusetzen und bei Bedarf für die jeweils spezifischen Zwecke anzupassen.

Ziel dieser Arbeit ist die automatisierte Analyse und der Vergleich von Fahrdaten, die durch die Fahrt in einem Fahrsimulator gewonnen werden konnten. Die Analyse soll in der Lage sein, das Ausmaß der Ablenkung eines Fahrers zu ermitteln und zu bewerten. Die Lösung soll so flexibel sein, dass sie mit wenigen Anpassungen auch weitere Messergebnisse, wie zum Beispiel zur Interaktion des Fahrers mit weiteren Fahrzeugen, erzeugen kann, die in dem breiten Einsatzfeld von Fahrsimulatoren gewünscht sein können. Diese Arbeit beschäftigt sich erstens mit der Identifikation der Daten, welche für die Analyse einer Simulationsfahrt hinsichtlich Ablenkung besonders interessant sind, zweitens mit der Anpassung des vorhandenen Fahrsimulators, um die gewünschten Daten zu speichern sowie drittens mit deren grafischer und automatisierter rechnerischen Auswertung. Die entwickelte Lösung wird in zwei Studien evaluiert, die zeigen das das Ergebnis unterschiedliche Ablenkungen eines Fahrers messen kann.

## 1.2 Vorgehen

Zunächst wird in Kapitel 2 der Fahrsimulator vorgestellt, welcher im Rahmen dieser Arbeit verwendet wurde. In Kapitel 3 werden verwandte Arbeiten betrachtet, in denen mit Hilfe von Fahrsimulatoren die Fahrleistung der Fahrzeugführer gemessen wird oder die Fahrsimulatoren selbst im Mittelpunkt stehen. Dabei wird untersucht, welche Daten in Studien mit Fahrsimulatoren und bei der Fahrt mit Fahrzeugen mit Sensoren ausgewertet werden. In Kapitel 4 werden Ziele für die Analyse durch das Analyseprogramm erarbeitet und deren Umsetzung durch spezifische Analysetechniken erläutert. Die Daten, die für die Analysetechniken notwendig sind, werden im Anschluss in Kapitel 4.2 identifiziert. In Kapitel 4.4 wird die Architektur des Analyseprogramms sowie die Veränderungen an der Architektur des Fahrsimulators erläutert, bevor ausgewählte Details der Implementierung betrachtet werden. In den Kapiteln 4.5.2 und 4.6 werden die Erweiterungsmöglichkeiten der erstellten Software aufgezeigt und eine Anleitung gegeben, wie das Analyseprogramm um weiteren Analysemethoden erweitert werden kann. Im Anschluss wird in Kapitel 4.7 das Analyseprogramm hinsichtlich der Skalierbarkeit analysiert. Anschließend wird an einem Beispiel die Bedienung des Analyseprogramms erläutert, einschließlich des Aufbaus einer Studie. Das Analyseprogramm wird zum Schluss anhand zweier Studien evaluiert,

die zeigen, dass die Software die Ablenkung eines Fahrers während einer Simulationsfahrt identifizieren und bewerten kann.

## 2 Technologisches Umfeld

Der verwendete Fahrsimulator wurde im Wintersemester 07/08 in einem Projektseminar von Stefan Schneegeß, Lukasz Wolejko-Wolejszo und Marco Müller unter der Betreuung von Dipl.-Medieninf. Dagmar Kern beim Lehrstuhl Pervasive Computing von Prof. Dr. Albrecht Schmidt an der Universität Duisburg-Essen entwickelt. Er ist frei verfügbar und wird auf Anfrage bei dagmar.kern@icb.uni-duisburg-essen.de versendet. Zum Zeitpunkt dieser Arbeit stand der Fahrsimulator noch nicht öffentlich zum Herunterladen bereit.

### 2.1 Klassifizierung des Fahrsimulators

In der Literatur wird zwischen günstigen, mittelteuren und teuren, detailreichen und weniger detailreichen, sowie Fahrsimulatoren mit fester und mit beweglicher Basis unterschieden [6][12][11]. Der Fahrsimulator lässt sich in diesen drei Kriterien wie folgt einordnen: Er ist günstig, da er frei von Lizenzkosten vertrieben wird. Er ist weitgehend detailreich. Es wird eine dreidimensionale Landschaft dargestellt, mit deutlich zu unterscheidenden Straßen und Grünflächen. Weniger detailreiche Fahrsimulatoren zeigen beispielsweise ausschließlich die Linien auf der Fahrbahn [12]. Der Fahrsimulator hat eine feste Basis. Das heißt der Fahrer, das Display und die Eingabegeräte befinden sich auf einem nicht beweglichem Ort, ohne dass die Kräfte, die auf den Fahrer in der Realität einwirken würden, durch eine bewegende Plattform simuliert werden. Eine bewegliche Basis hat keine signifikanten Vorteile gegenüber der Lösung mit einem festen Untergrund. Ebenso hat die Komplexität der Darstellung keinen signifikanten Einfluss auf Analyseergebnisse [12].

### 2.2 Technologische Grundlagen

Der Fahrsimulator wurde mit der Programmiersprache Java implementiert und basiert auf der Engine JMonkeyEngine<sup>1</sup>, die ebenfalls in Java geschrieben wurde. JMonkeyEngine nutzt die Lightweight Java Game Library (LWJGL)<sup>2</sup> als Schnittstelle zu OpenGL [1]. LWJGL ist eine Programmbibliothek für Java, mit dem Fokus auf Spieleentwicklung [2]. OpenGL ist ein Application Programming Interface (API) für 2D- und 3D-Grafik-Applikationen, das in der Industrie weit verbreitet ist [3]. Der Fahrsimulator ist demnach auf allen Plattformen verfügbar, für die eine Java Virtual Machine existiert und die OpenGL darstellen können. Die Analyse der Fahrdaten wird ebenfalls in Java entwickelt.

Als Eingabegerät werden ein Lenkrad und Pedale für den PC verwendet, welche günstig im Handel erworben werden können. Die Szenerie kann über einen Bildschirm oder einen Beamer dargestellt werden.

Die Simulation findet auf einem Computer statt. Der Fahrsimulator sendet Fahrdaten, wie zum Beispiel die aktuelle Geschwindigkeit oder die Betätigung der Bremse mit User Datagram Protocol-Paketen (UDP) in ein lokales Netzwerk. Der Empfänger, ein anderer Computer oder der Computer, der selbst die Simulation durchführt, kann anhand dieser

---

<sup>1</sup><http://www.jmonkeyengine.com>

<sup>2</sup><http://lwjgl.org>



Daten mit einem zusätzlichen Programm beispielsweise Armaturen simulieren, die die aktuelle Geschwindigkeit anzeigen.

Getestet wurde der Fahrsimulator auf einem IBM ThinkPad T42, mit einem Intel Pentium M Prozessor mit 1,7GHz und 1GB RAM sowie einer ATI Mobility Radeon 7500 Grafikkarte. Für die Studien zur Evaluation wurden etwas leistungsstärkere Computer verwendet.

## 2.3 Systemarchitektur

Der Fahrsimulator nutzt die in Abbildung 2.1 dargestellte Architektur.

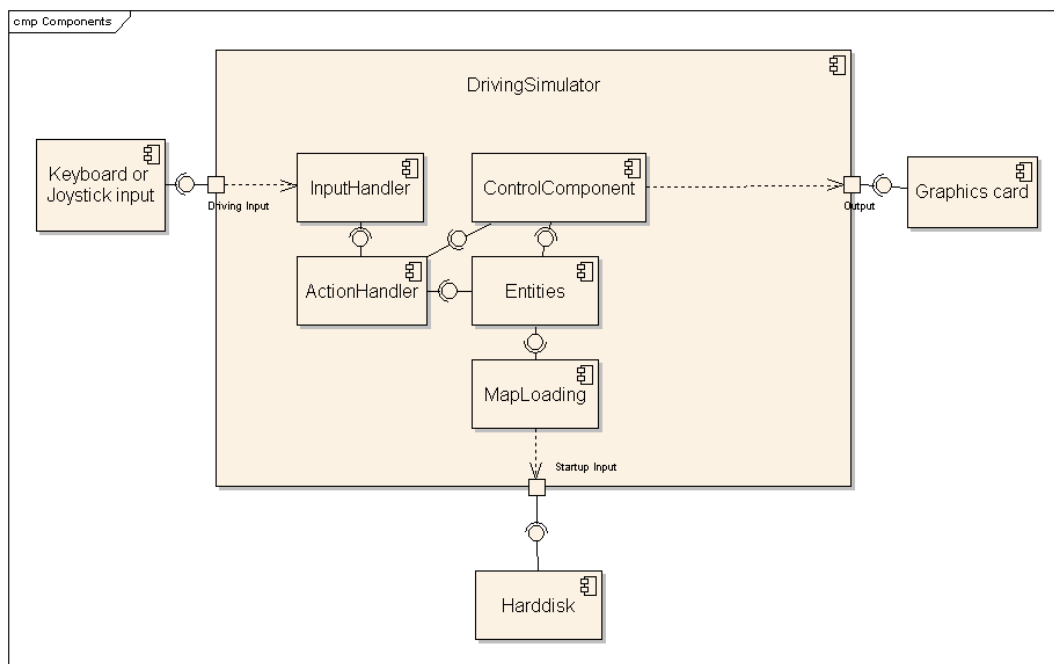


Abbildung 2.1: Systemarchitektur des Fahrsimulators

Die Eingaben werden über *InputHandler* auf *ActionHandler* verteilt, die das Geschehen in der virtuellen Welt beeinflussen. Systemparameter werden zu Beginn der Simulation, d.h. beim Start der Applikation aus Konfigurationsdateien eingelesen und beeinflussen das Fahrverhalten oder definieren die zu ladende Weltkarte. Die Steuerung, die durch die Komponente *ControlComponent* dargestellt wird, erstellt die zu rendernde Szene anhand der Daten der Entitäten.

### 2.3.1 Entitäten

Die Entitäten in dem Fahrsimulator sind das Fahrzeug (*Car*), Kartenfelder (*MapField*) sowie statische und dynamische Kartenobjekte (*StaticObject* bzw. *DynamicObject*). Das Fahrzeug ist durch die in Abbildung 2.2 dargestellten Klassen realisiert. Die Klassendiagramme enthalten nur ausgewählte Operationen und Attribute. Das Fahrzeug hat zwei Achsen (*Suspension*) und jede Achse hat zwei Räder (*Wheel*). Ferner hat das Fahrzeug u.a. Methoden für die üblichen Fahrmanöver: Lenken, Beschleunigen und Bremsen, sowie einige Get-Methoden zum Auslesen der Attribute und beispielsweise zur Berechnung der Geschwindigkeit.

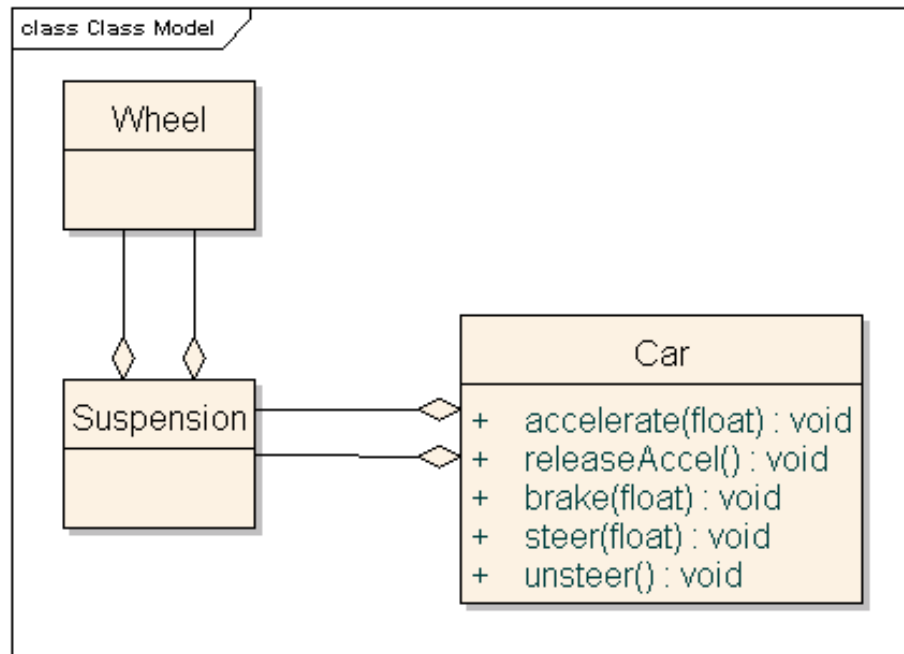


Abbildung 2.2: Klassendiagramm mit Fokus auf das Fahrzeug

Die Karte, die die simulierte Welt bildet, besteht aus den in Abbildung 2.3 dargestellten Klassen. Eine Karte (*Map*) besteht aus einem oder mehreren Kartenfeldern (*MapField*). Kartenfelder enthalten Informationen über die Art des Feldes, z.B. Nord-Süd-Straße oder Gras, ihre Position auf der Karte und die Textur, die über sie gelegt wird.

Ein Kartenobjekt (*MapObject*) ist ein Objekt auf der Karte, wie z.B. Schilder am Straßenrand oder Begrenzungspfähle, als statische Objekte oder bewegliche Objekte, wie z.B. ein Ball als dynamisches Objekt. Zu diesem Zeitpunkt ist in den Fahr Simulator noch kein dynamisches Objekt integriert.

### 2.3.2 InputHandler

*InputHandler* verarbeiten die Eingaben, die während der Simulation getätigt werden. Dies sind z.B. Eingaben für Lenkmanöver oder das Setzen eines Markers in den Analysedaten, der z.B. den Beginn oder das Ende einer Sekundärtätigkeit des Fahrers markieren kann und zum flexiblen Einsatz gedacht ist. Zum Zeitpunkt dieser Arbeit sind drei *InputHandler* implementiert: Der *CommonInputHandler* verarbeitet Eingaben, die für alle *InputHandler* gelten. Dies sind z.B. die o.g. Marker oder das Wechseln zwischen Tastatur- und Lenkradsteuerung, in Form des aktiven erweiternden *InputHandlers*.

Diese erweiternden *InputHandler* sind der *KeyboardInputHandler*, der Tastatureingaben für die Steuerung des Fahrzeugs verarbeitet, sowie der *JoystickInputHandler*, der die Eingaben eines Joysticks oder Lenkrads verarbeitet. Die Struktur wird durch das Klassendiagramm in Abbildung 2.4 verdeutlicht. Jede Eingabe wird durch den aktiven *InputHandler* verarbeitet. Eingaben, die vom *InputHandler* nicht vorgesehen sind, werden verworfen. Alle anderen Eingaben werden als *InputActionEvent* einem *ActionHandler* zugewiesen. Mit dieser Zuweisung endet die Aufgabe des *InputHandlers*.

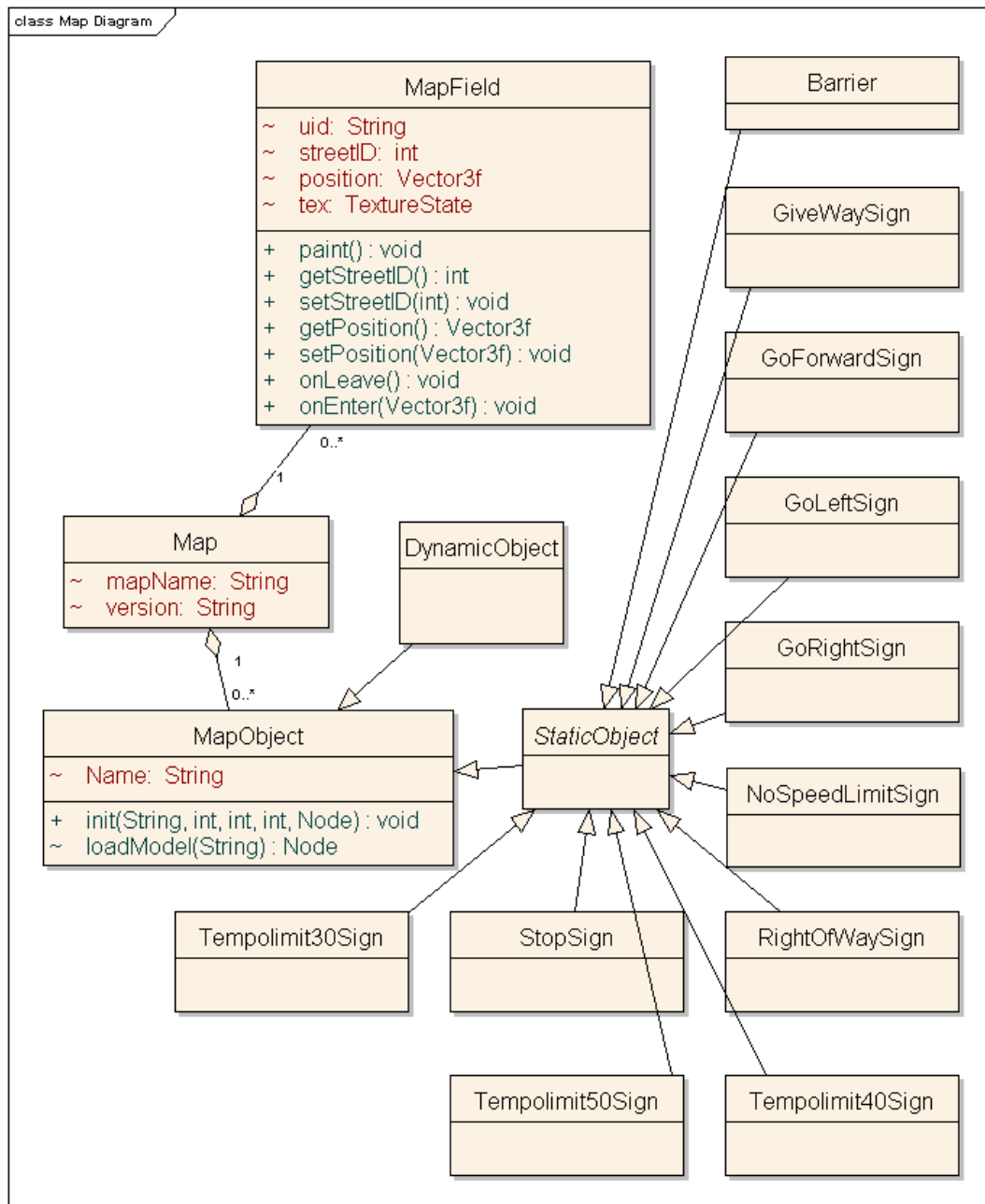


Abbildung 2.3: Klassendiagramm mit Fokus auf die Karte

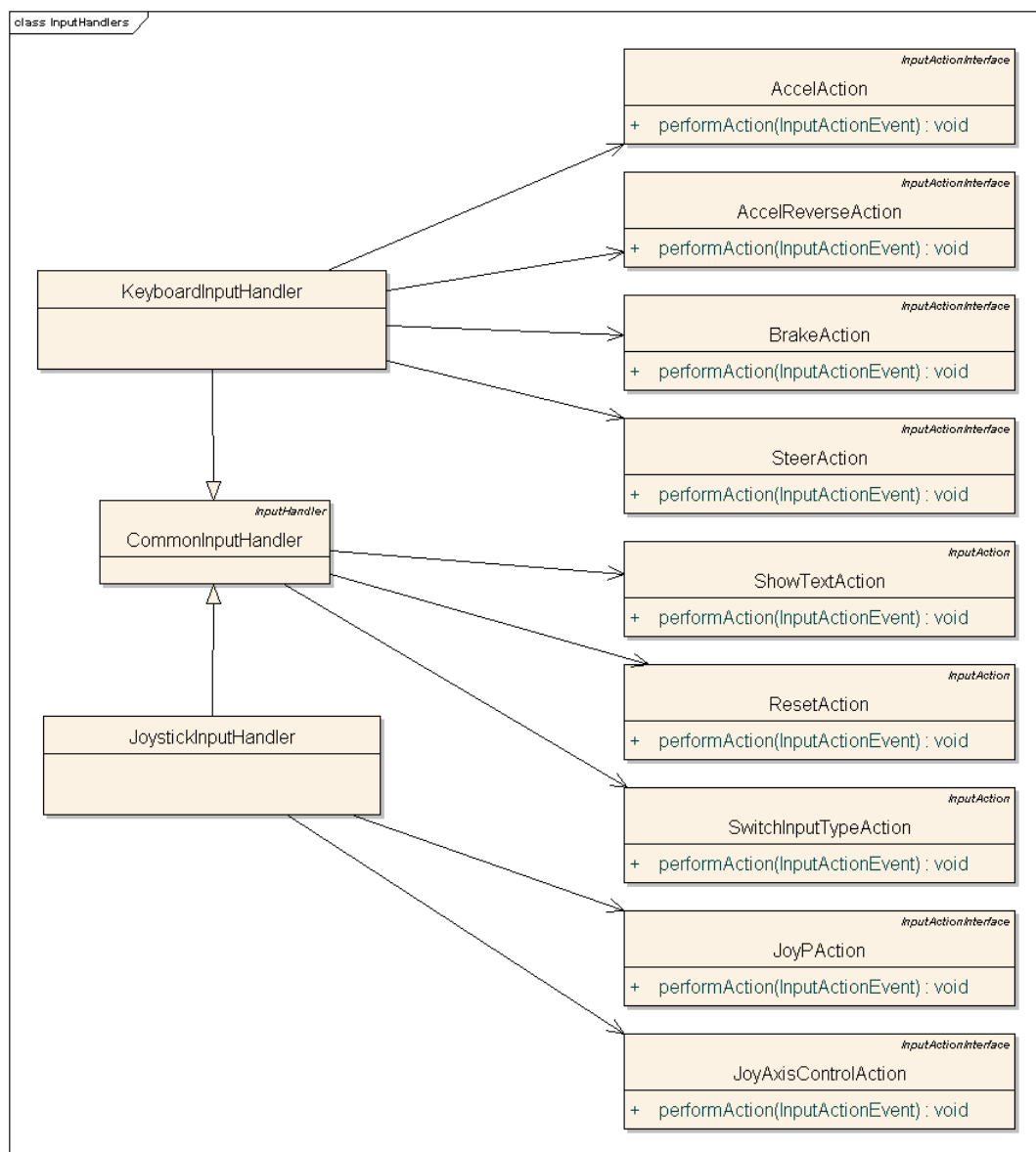


Abbildung 2.4: Klassendiagramm der ActionHandler und InputHandler

### 2.3.3 ActionHandler

Ein *ActionHandler* nimmt ein *InputActionEvent* entgegen, welches die Daten der Eingabe enthält, und setzt die gewünschte Änderung um. Beispiele für *ActionHandler* sind das Setzen eines Markers in den Fahrdaten mit der *SetMarkerAction* und das Lenken des Fahrzeugs mit dem Joystick oder dem Lenkrad mit der *JoyAxisControlAction*. Die Struktur und der Zusammenhang zwischen den *ActionHandlern* und den *InputHandlern* werden in Abbildung 2.4 dargestellt.

### 2.3.4 Kartenmanagement

Eine Karte enthält mehrere Kartenfelder. Jedes Kartenfeld entspricht einem Kartenfeldtyp. Kartenfeldtypen sind z.B. ein Straßenstück von Norden nach Süden, eine Kurve von Osten nach Süden oder Gras. Kartenfeldtypen sind durch eindeutige Identifikationsnummern definiert. Jede dieser IDs sind Zahlen mit drei Ziffern. So bedeutet die Zahl 014 ein Feld mit Gras, 003 ein Feld mit einem Straßenstück von Westen nach Osten und 001 ein Straßenstück von Norden nach Süden und repräsentiert den Startpunkt des Fahrzeugs. Die Karten-Dateien sind wie folgt aufgebaut:

Name der Karte

Versionsnummer der Karte

Anzahl der Felder der Karte auf der X-Achse

Anzahl der Felder der Karte auf der Z-Achse

Kartenfeldtyp-IDs in Form einer Karte:

$i(0,0)i(1,0)\dots i(n,0)$

$i(0,1)i(1,1)\dots i(n,1)$

...

$i(0,n)i(1,n)\dots i(n,n)$

$i(x,z)$  ist dabei die Kartenfeldtyp-ID des Feldes der Koordinaten  $(x,z)$ .

### 2.3.5 Konfigurationsdateien

Zwei Konfigurationsdateien enthalten Systemparameter, die beim Start der Applikation ausgelesen werden. Diese Dateien sind die *messages.conf* und die *simulator.conf*. Die Datei *messages.conf* ist für Entwickler gedacht, um zentral Nachrichten für das System zu speichern, wie z.B. Fehlermeldungen oder Texte auf dem Heads Up Display (HUD). Die Datei *simulator.conf* enthält Daten, die beispielsweise die zu ladende Kartendatei bestimmen oder den Empfänger der UDP-Pakete angeben, der dadurch aktuelle Fahrdaten gesendet bekommt.

### 2.3.6 Steuerungskomponente

Die Simulation wird von der Steuerungskomponente *ControlComponent* vorangetrieben. Sie stellt den Kern der Simulation dar. Sie überwacht die Logikrate und lässt die Szenerie für die Ausgabe auf dem Bildschirm rendern. Die Logikrate ist die Frequenz, mit der die virtuelle Realität in ihrer Datenrepräsentation aktualisiert wird. Im Gegensatz dazu steht die Framerate, welche die Frequenz angibt, mit der die virtuelle Welt für die Darstellung gezeichnet wird. Außerdem laufen bei ihr die Referenzen zu den *InputHandlern*, zu den Karten und den Konfigurationsdateien zusammen.



verwendet. Diese wurden mit dem arithmetischen Mittel und der Standardabweichung der folgenden Messwerte verglichen:

1. Spurposition  
Die Position des Fahrzeugs auf der Fahrbahn
2. Geschwindigkeit
3. Gierwinkel  
Der Winkel zwischen Fahrtrichtung und Mittellinie, bei Kurven zwischen Fahrtrichtung und der Tangente an der Mittellinie.

Ferner wurden die Ergebnisse mit dem genannten Auslastungsmodell von Wierwille verglichen. Die Ergebnisse dieser Studie sind, dass die Standardabweichung der Spurposition bei unterschiedlichen Spurbreiten nicht als Maßstab für Fahrerauslastung gewertet werden kann, da Fahrer in der Regel die komplette, ihnen zur Verfügung stehende Spur für Fahrmanöver verwenden. Größere Spurbreiten führten zu höheren Werten bei der Standardabweichung der Spurposition, auch bei wenig auslastend bewerteten Teilstrecken. Die mittlere Geschwindigkeit und die Standardabweichung der mittleren Geschwindigkeit korrelierten nicht mit anderen Messwerten und Bewertungen der Fahrerauslastung. Dies wird entkräftet, indem auf Schwächen im Versuchsaufbau hingewiesen wird.

## 3.2 Vergleich zwischen Alkoholeinfluss und Telefonat

Ein Vergleich zwischen alkoholisierten Fahrern und Fahrern, die ein Mobiltelefon nutzen wird in [5] durchgeführt. Hier wurde der „PatrolSim“ Fahrsimulator der Firma GE-ISIM verwendet. Heute wird der Fahrsimulator von der Firma MPRI vertrieben<sup>4</sup>. Das Ziel dieser Studie war, eine mögliche Beeinträchtigung der Fahrleistung durch das Telefonieren zu ermitteln. Dazu haben die Teilnehmer kontrolliert Alkohol zu sich genommen, um



Abbildung 3.2: Ein Teilnehmer am PatrolSim Fahrsimulator [5]

<sup>4</sup><http://www.mpri.com/driver/patrolsimiv.html>

eine Blutalkoholkonzentration von 0,08% (wt/vol) zu erreichen. Die Aufgabe des Fahrers war es, einem anderen Fahrzeug zu folgen, welches in unregelmäßigen Abständen bremst, bevor es wieder auf normale Fahrgeschwindigkeit beschleunigt. Das Ziel war es, eine Kollision mit dem vorfahrenden Fahrzeug zu vermeiden, ohne den Anschluss zu verlieren. Dieses Szenario sollte einen Stop-And-Go-Verkehr simulieren. In einer zweiten Sitzung haben die gleichen Teilnehmer die gleiche Aufgabe wahrgenommen, während sie mit einem Headset telefonieren, statt alkoholisiert zu sein. Ein Beispiel für die zweite Sitzung ist in Abbildung 3.2 zu sehen. In dieser Studie wurden die folgenden Messwerte berechnet:

1. Anzahl der Unfälle  
Als Unfälle wurden Kollisionen mit dem vorfahrenden Fahrzeug gewertet.
2. Brems-Reaktionszeit  
Die Zeit von dem Aufleuchten der Bremsleuchten der vorfahrenden Fahrzeugs bis zur Bremsreaktion des Teilnehmers
3. Maximale Bremskraft  
Die maximale Kraft, mit der der Teilnehmer gebremst hat
4. Mittlere Geschwindigkeit
5. Mittlerer Abstand  
Mittlerer Abstand zum vorfahrenden Fahrzeug, bevor das vordere Fahrzeug bremst
6. Standardabweichung des Abstands  
Die Standardabweichung des mittleren Abstands zum vorfahrenden Fahrzeug, bevor das vordere Fahrzeug bremst
7. Zeit bis zur Kollision  
Die Zeit bis zu einer Kollision, für den Fall, dass der Teilnehmer nicht auf das Bremsen des vorfahrenden Fahrzeugs reagiert hätte
8. Zeit bis zur Kollision unter 4 Sekunden  
Die Anzahl der Messungen einer Simulationsfahrt, bei der die Zeit bis zur Kollision kleiner als 4 Sekunden ist
9. Zeit zum Wiederaufbau des halben Geschwindigkeitsverlusts  
Die Zeit, die der Teilnehmer nach dem Bremsmanöver benötigt, um die Hälfte der beim Bremsmanöver verlorenen Geschwindigkeit wieder zu erreichen. Z.B. wenn der Teilnehmer von 60 km/h auf 40 km/h herabgebremst hat, wird die Zeit gemessen, in der er von 40 km/h auf 50 km/h beschleunigt.

In der Studie sind telefonierende Teilnehmer häufiger dem vorfahrenden Fahrzeug aufgefahen als alkoholisierte Fahrer. Auch die Reaktionszeit litt deutlicher unter dem Telefonat als unter dem Alkoholeinfluss. Alkoholisierte Fahrer zeigten ein aggressiveres Fahrverhalten, was aus signifikant erhöhten Werten unter der Rubrik „Zeit bis zur Kollision unter 4 Sekunden“ und signifikant höherer maximaler Bremskraft interpretiert wurde.



### 3.3 Fahrsimulatoren in der Fahrerausbildung

Die Arbeit von Leitao, Moreira, Santos Sousa und Ferreira [17] betrachtet den Einsatz von Fahrsimulatoren in der Fahrerausbildung. Die Autoren erläutern in dieser Arbeit den Aufbau der Studie, mit welcher sie den Lernerfolg durch Fahrsimulatoren prüfen möchten. Das Ziel ist es, den Einfluss von Realitätsnähe des Fahrsimulators auf den Lernprozess zu untersuchen und herauszufinden, wann und wie Fahrsimulatoren in der Lehre eingesetzt werden sollten, um einen maximalen Erfolg daraus ziehen zu können.

In der Studie soll der von ihnen entwickelte Fahrsimulator „DriS“ eingesetzt werden, welcher in [16] beschrieben wird. Der Aufbau des Fahrsimulators ist in Abbildung 3.3 illustriert.

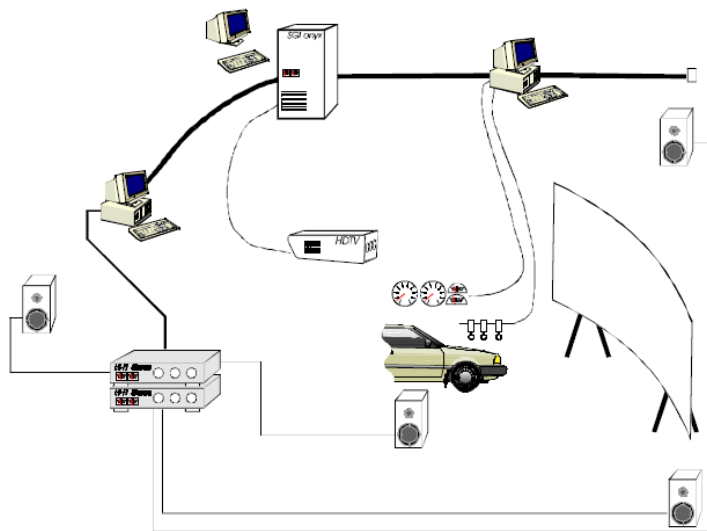


Abbildung 3.3: Der „DriS“ Fahrsimulator [17]

In diesem Fahrsimulator werden drei Computer eingesetzt, die durch ein Netzwerk miteinander verbunden sind. Ein Computer verwaltet die Fahrsimulation und stellt die Szene durch einen Beamer dar, ein weiterer verarbeitet die Audioausgabe und ein dritter verarbeitet die Daten für das Cockpit.

Für die Studie wurde in dem Fahrsimulator eine Umgebung nachgebildet, welche in Porto in Brasilien häufig von Fahrlehrern in der Fahrausbildung genutzt wird. Ein Beispiel der Ausgabe des Fahrsimulators „DriS“ ist in Abbildung 3.4 zu sehen.

Die Teilnehmer der Studie sollen zwischen 16 und 22 Jahren sein, keinen Führerschein besitzen, die Absicht haben einen Führerschein zu bekommen und keine Seh- oder sonstige körperliche Behinderung aufweisen.

Jeder Teilnehmer soll eine Fahrausbildung durchlaufen, welche ebenso Fahrten in einem echten Fahrzeug sowie Fahrten in einem Fahrsimulator enthalten soll. Der Realitätsgrad des Fahrsimulators soll zwischen einem nicht interaktiven Film und interaktivem Fahren im Simulator variieren. Es soll vier unterschiedliche Abläufe der Fahrausbildung geben. Gruppe 1 soll eine Fahrausbildung ohne Fahrsimulator erhalten. Gruppe 2 soll in der Ausbildung den Fahrsimulator nutzen, wobei der Simulator interaktiv genutzt werden soll. Gruppe 3 nutzt den Simulator in der Ausbildung nicht interaktiv. Es sollen jeweils die Fahrleistungen der Teilnehmer vor und nach der Fahrausbildung verglichen werden. Die Autoren versäumen, die Konfiguration in der vierten Gruppe zu erklären.



Abbildung 3.4: Sicht auf eine modellierte Straße im Fahrsimulator „DriS“ [17]

Die Fahrleistungen werden anhand dreier Fahrten ermittelt, die sich hinsichtlich des Verkehrsaufkommens unterscheiden. Dabei werden drei Datengruppen gemessen:

1. Der geistige Aufwand, gemessen durch elektro-magnetische Sensoren am Kopf des Teilnehmers
2. Die Augenbewegung, gefilmt durch eine Kamera am Kopf des Teilnehmers
3. Die Fahrdaten im Simulator

Zudem soll die Fahrt mit einer Kamera aus Sicht des Teilnehmers aufgenommen werden. Die Messung des geistigen Aufwands und der Augenbewegung können der Situation im Fahrsimulator zugeordnet werden um zu messen, wann der Teilnehmer besonders gefordert war (geistiger Aufwand) und auf welche Objekte er seine Aufmerksamkeit gerichtet hat (Augenbewegung). Die im Simulator gemessenen Fahrdaten sind:

1. Reaktionszeit
2. Zeit bis zur Fahrbahnbegrenzung
3. Spurhaltung
4. Geschwindigkeit
5. Anzahl der Kollisionen

Zusätzlich sollen Fahrlehrer und nicht näher genannte Experten die Videoaufzeichnung aus Sicht der Teilnehmer subjektiv anhand eines Evaluationsbogens unter Anderem hinsichtlich Fahrzeugkontrolle, Fähigkeiten in kritischen Situationen und Lernfortschritt während der Fahrt bewerten.

Anhand eines Vergleichs der subjektiven und der objektiven Daten soll dann untersucht werden, welche der Gruppen den größten Lernerfolg hatte, um zu erkennen, wie und wann der Einsatz von Fahrsimulatoren in der Fahrausbildung sinnvoll ist. Die Studie wurde zum Zeitpunkt der Veröffentlichung nicht durchgeführt und es konnte kein Bericht über das Durchführen der beschriebenen Studie gefunden werden.

### 3.4 Vergleich einer realen Fahrt und Fahrsimulatoren

Green und Reed haben in [12] die Fahrt in einem mit Messinstrumenten ausgestatteten Fahrzeug mit der Fahrt in einem detailreichen und in einem detailarmen Fahrsimulator verglichen.

Die Teilnehmer mussten eine sekundäre Aufgabe neben dem Führen des Fahrzeugs durchführen. Die sekundäre Aufgabe war das Wählen von Telefonnummern an einem Mobiltelefon. Bei jeder Fahrt wurden Spurposition, Geschwindigkeit, Lenkerstellung und Gaspedalposition gemessen und gespeichert. Die in dieser Arbeit berechneten Variablen zum Vergleich der Messwerte waren:

1. Mittlere Spurposition  
Das arithmetische Mittel des Abstands zur Mittellinie, positiv auf der rechten Seite der Mittellinie und negativ auf der linken Seite der Mittellinie. Die mittlere Spurposition ist in Abbildung 3.5 veranschaulicht.
2. Standardabweichung der Spurposition  
Die Standardabweichung zur mittleren Spurposition. Die Standardabweichung der mittleren Spurposition ist in Abbildung 3.5 dargestellt.
3. Mittlere seitliche Geschwindigkeit  
Das arithmetische Mittel der seitlichen Geschwindigkeit des Fahrzeugs. Die seitliche Geschwindigkeit ist jene, mit der sich das Fahrzeug quer zur Fahrtrichtung bewegt.
4. Durchschnitt des Logarithmus der Zeit bis zur Linie  
Die Zeit bis zur Linie wird aus der Spurposition des Fahrzeugs und der Mittellinie, bzw. der Seitenlinie berechnet. Dazu werden zwei Datensätze mit Zeitinformationen genutzt. Der Logarithmus des Betrags wird benutzt, um den Einfluss der hohen Werte zu mindern, wenn die seitliche Geschwindigkeit die Richtung wechselt.
5. Standardabweichung der Geschwindigkeit  
Die Standardabweichung der mittleren Geschwindigkeit des Fahrzeugs
6. Mittlere absolute Beschleunigung  
Die mittleren absoluten Werte der Geschwindigkeitsunterschiede zwischen den Datensätzen
7. Standardabweichung der Lenkerstellung  
Die Standardabweichung der mittleren Lenkerstellung in Grad
8. Frequenz der Steuerkorrektur  
Die mittlere Anzahl der Richtungswechsel des Lenkrads pro Sekunde
9. Standardabweichung der Gaspedalposition  
Die Standardabweichung von der mittleren Gaspedalposition
10. Frequenz der Geschwindigkeitskorrektur  
Die mittlere Anzahl der Geschwindigkeitskorrekturen pro Sekunde



Abbildung 3.5: Mittlere Spurposition und Standardabweichung der mittleren Spurposition in [12]

Die Ergebnisse der Studie machen Unterschiede zwischen dem Fahrverhalten im Fahrsimulator und im echten Fahrzeug deutlich. Beispielsweise war die Standardabweichung der Spurposition beim Ausführen der Sekundäraufgabe bei der Fahrt im Fahrsimulator ausgeprägter. Zwischen den Messungen bei den Fahrten im detailreichen und dem detailarmen Fahrsimulator konnten keine signifikanten Unterschiede festgestellt werden.

### 3.5 Evaluation von Fahrerablenkung

S. Mattes evaluiert in [19] das Programm zur Messung der Fahrerablenkung namens Lane-Change-Task (LCT). In dem Programm LCT wird eine Fahrt in einem Fahrzeug simuliert, welches auf einer 3 km langen Straße mit drei Fahrspuren fährt. Auf der Strecke befinden sich keine weiteren Fahrzeuge. Schilder auf der linken und rechten Seite der Fahrstrecke geben an, auf welche Fahrspur der Fahrer wechseln soll. Abbildung 3.6 zeigt eine Szene des LCT zur Veranschaulichung.



Abbildung 3.6: Szene aus dem LCT [19]

LCT wird auf einem PC ausgeführt und wird mit einem Lenkrad gesteuert, welches auch für Computerspiele eingesetzt wird. Die Geschwindigkeit ist bei den Fahrten auf 60 km/h festgelegt, was bei den 3 km langen Strecken zu einer Fahrzeit von ca. 180 Sekunden führt. Für jede Strecke existiert eine normative Fahrlinie. Bei jeder Fahrt wird die Distanz

zwischen der gemessenen Fahrlinie und der normativen Fahrlinie ausgewertet. Abbildung 3.7 skizziert die normative und die gemessene Fahrlinie.

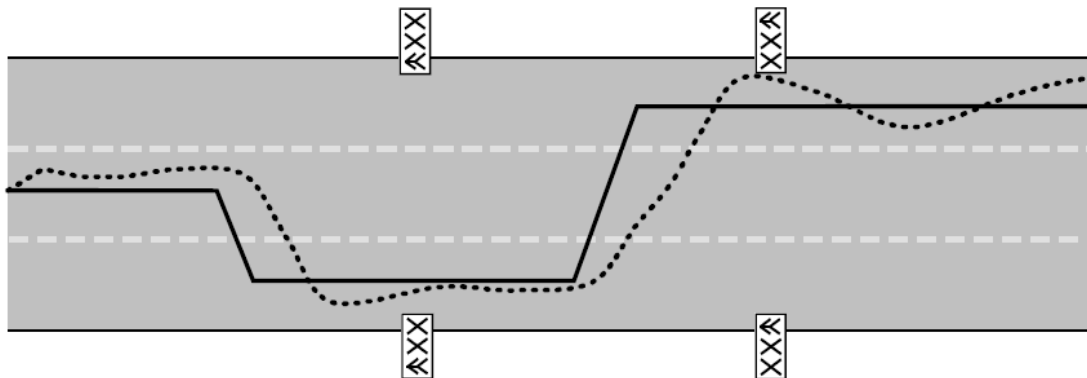


Abbildung 3.7: Normative und gemessene Fahrlinie im LCT

Die Differenz zwischen der normativen und der gemessenen Linie wird bei der Analyse bewertet und bildet die Grundlage zur Erkennung der Fahrerablenkung. Die Analyse der Differenz zwischen normativer und gemessener Fahrlinie deckt laut Mattes die Bewertung der folgenden Fahreigenschaften ab:

1. Wahrnehmung  
Eine späte Wahrnehmung der Schilder oder das Übersehen von Schildern führt zu einer hohen Differenz.
2. Qualität des Lenkmanövers  
Ein langsamer Wechsel der Fahrspur erhöht die Differenz der Fahrlinien.
3. Spurhaltung  
Eine schlechte Spurhaltung erhöht die Differenz zwischen den Spurwechseln.

In der Studie zur Evaluation fuhren 45 Teilnehmer zunächst eine Fahrt zur Gewöhnung an den LCT, bevor sie eine Fahrt ohne Nebenaufgabe als Referenzfahrt durchführten. Im Anschluss fuhr jeder Teilnehmer 8 von 12 Fahrten mit je einer Nebenaufgabe. Die Nebenaufgaben waren:

1. Beantwortung simpler Fragen an einem Telefon (TT)
2. Bonbon auspacken und in den Aschenbecher legen (SW)
3. Taschentuch entfalten und auf den Beifahrersitz legen (KX)
4. Straßennamen mit einem drehbarem Druckknopf in Navigationsgerät eingeben (NS)
5. Vierstellige PIN in ein Mobiltelefon eingeben (CP)
6. Audiokassette wechseln und in die Kassettenhülle legen (CC)
7. Telefonnummer in einem kleinem Adressbuch heraussuchen (AB)
8. Autoradio einstellen, z.B. Höhen verstärken (SA)

9. Frequenz des Autoradios ändern (RT)
10. Karte auf Seite X aufschlagen und die Position zweier Städte vergleichen (MB)
11. Zielpunkt des Navigationssystems setzen (NM)
12. Zwei Münzen aus einem Portemonnaie heraussuchen (CO)

Von der Differenz zwischen normativer und gemessener Fahrlinie aus drei Fahrten ohne Nebenaufgabe eines jeden Teilnehmers wurde der Durchschnitt gebildet, welcher in der Analyse einen Referenzwert darstellt. In der Arbeit ist nicht angegeben, wann die zweite und die dritte Fahrt ohne Nebenaufgabe durchgeführt wurde. Dieser Referenzwert wurde mit den Werten der Fahrten mit Nebenaufgaben verglichen. Der Quotient aus dem Referenzwert  $\sigma_r$  und dem Wert der Fahrt mit der jeweiligen Nebenaufgabe  $\sigma_{an}$  soll die Ablenkung des Fahrers zeigen. Abbildung 3.8 stellt die Ergebnisse der Studie grafisch miteinander in Beziehung.

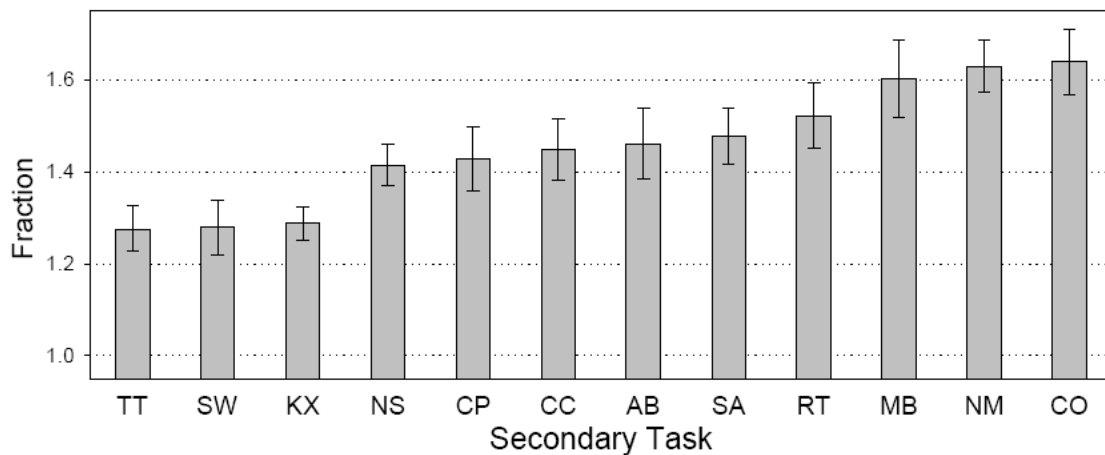


Abbildung 3.8: Ergebnisse der Evaluationsstudie des LCT [19]

Mit Hilfe der Varianzanalyse [20] wurde die Signifikanz der Unterschiede nachgewiesen. Die Ergebnisse korrelieren mit den Ergebnissen einer Studie mit dem DymlerChrysler Fahrsimulator mit beweglicher Basis, die die gleichen zwölf Nebenaufgaben enthielt.

### 3.6 Vergleich der verwandten Arbeiten

In den erläuterten verwandten Arbeiten wurden Fahrsimulatoren in verschiedenen Studien und zu verschiedenen Zwecken eingesetzt oder evaluiert. In 3.1 wurde ein Fahrsimulator eingesetzt, um die Belastung von Fahrern bei unterschiedlicher Straßenarchitektur zu bewerten.

Die Studie in 3.2 hat die Fahreigenschaften von Fahrern unter dem Einfluss von Alkohol bzw. beim Telefonat verglichen. Hier wird ebenfalls die Fahrleistung gemessen, jedoch gehen die Unterschiede der Fahrleistung in 3.1 von der Straßenarchitektur aus, während die Unterschiede in 3.2 vom Fahrer ausgehen.

In 3.3 wird eine Studie beschrieben, die den Einsatz von Fahrsimulatoren in der Fahrausbildung bewerten soll. Hier ist nicht die Fahrleistung im Fokus, sondern der Einsatz des Fahrsimulators und wie er beim Lernerfolg in der Ausbildung beitragen kann.

Unterschiedliche Fahrsimulatoren werden in 3.4 mit dem Fahren in der Realität verglichen. Hier liegen die Fahrsimulatoren selbst im Fokus. Es wurde herausgefunden, dass zwischen dem Fahren in einem echten Fahrzeug und dem Fahren in einem Fahrsimulator signifikante Unterschiede bestehen, zwischen unterschiedlicher Realitätsnähe der Fahrsimulatoren jedoch nicht.

Auch bei 3.5 wird der Fahrsimulator selbst betrachtet. Im Gegensatz zu den anderen Arbeiten wird bei diesem Fahrsimulator jedoch ausschließlich ein Wert gemessen, der Abstand des Fahrzeugs zu einer Soll-Linie. Zudem wird bei dieser Studie die Geschwindigkeit festgelegt, damit die Werte des Abstands zur Soll-Linie vergleichbar sind. Hieran ist zu kritisieren, dass Fahrer bei der Bearbeitung einer Nebenaufgabe in der Regel langsamer fahren, um sich besser auf die Nebenaufgabe konzentrieren zu können, wie im Kapitel 6 gezeigt werden wird. Es bleibt zu klären, ob das Festlegen der Geschwindigkeit ein zu künstlicher Eingriff in die Fahrt ist, um zuverlässige Werte aus der Anwendung des Fahrsimulators zu bekommen.

Die Analyse, die in dieser Arbeit vorgenommen wird, ist angelehnt an der Analyse durch den LCT in Kapitel 3.5. Ein entscheidender Unterschied ist jedoch, dass die Menge der Messwerte, die analysiert werden können umfangreicher ist und insbesondere die Geschwindigkeit in der Simulation nicht festgelegt ist. Zudem wird die Analyse in dieser Arbeit nicht auf den Spurwechsel beschränkt sondern ermöglicht das Messen der Fahrt auf komplexerer Straßenarchitektur.

## 4 Analyse der Simulationsdaten

### 4.1 Ziele der Analyse

Die Ziele der Analyse der Fahrdaten können vielfältig sein. Einige Ziele des Einsatzes von Fahrsimulatoren sind in [23] aufgelistet. Von den dort aufgelisteten Zielen kommen für diesen Fahrsimulator die Folgenden in Betracht:

#### 1. Regeltätigkeit des Fahrers

- Wahrnehmungsuntersuchungen  
Durch Analyse der Fahrspur kann beispielsweise untersucht werden, wie groß ein Signalschild sein muss, um rechtzeitig von einem Fahrer bei einer regulären Fahrt wahrgenommen zu werden. Außerdem kann festgestellt werden, welche Anzahl an Schildern an einem Ort dazu führt, dass einzelne Schilder nicht mehr wahrgenommen werden.
- Reaktionsverhalten  
Mit Hilfe einer Analyse des Bremszeitpunkts kann untersucht werden, wann der Fahrer auf einen Stimulus reagiert hat. Dadurch kann der Einfluss von Bedienelementen im Fahrzeug auf das Reaktionsverhalten des Fahrers ermittelt werden.
- Spurverhalten  
Durch den Abstand zur Mittellinie oder Ideallinie und die Position des Fahrzeugs kann das Spurverhalten analysiert werden. Dabei ist die Ideallinie die Fahrlinie, mit welcher der Fahrer das Fahrzeug am sichersten und am ruhigsten durch eine Kurve lenken kann, ohne die eigene Fahrspur zu verlassen, bzw. auf einer geraden Strecke genau in der Mitte der Fahrbahn.

- Vigilanz- und Ermüdungseinflüsse  
Bremsreaktionen und Spurhaltung können die Basis für Studien über Vigilanz- oder Ermüdungseinflüsse von langem Fahren bilden. Damit ist es möglich, die Ermüdungseinflüsse von unterschiedlichen Fahrersitzen zu analysieren.
- Alterseinflüsse  
Alterseinflüsse können bei der Bedienung von Benutzerschnittstellen im Fahrzeug eine Rolle spielen. Mit der Analyse des Reaktionsverhaltens können die Einflüsse des Alters bei der Bedienung von Eingabemethoden oder dem Ablesen von Informationen von einem Bildschirm getestet werden. Dies ermöglicht es beispielsweise, eine geeignete Schriftgröße für Monitore oder eine geeignete Größe für Knöpfe oder andere Eingabegeräte zu ermitteln.

## 2. Einfluss psychoaktiver Präparate

- Alkohol
- Medikamente
- Rauschgifte
- Nikotin

Durch die Analyse von Spurhaltungs- und Reaktionsdaten kann der Einfluss psychoaktiver Präparate auf das Fahrverhalten analysiert werden.

## 3. Wirksamkeit von Verkehrslenk- und -leiteinrichtungen

- Schilder und Markierungen  
Geeignete Schilder oder Schildergrößen können untersucht werden, indem z.B. die Reaktionszeit auf die entsprechenden Schilder durch Brems- oder Lenkmanöver getestet wird. Es kann auch untersucht werden, ob die Instruktionen von Schildern verständlich sind.
- Leit- und Informationsgeräte  
Leit- und Informationsgeräte können getestet werden, indem untersucht wird, ob der von den Geräten angegebene Kurs eingehalten wird. Damit ist es möglich, beispielsweise unverständliche Anweisungen in Navigationsgeräten zu vermeiden.

## 4. Entwurfsfindung und -überprüfung von Straßen

- Prognostizierung von Gefahrenstellen
- Lokalisierung von bestehenden Unfallschwerpunkten

Durch Analyse der Geschwindigkeit an bestimmten Stellen können Gefahrenpunkte auf existierenden oder geplanten Straßenabschnitten analysiert werden.

## 5. Fahrdiagnostik

- Führerscheinausbildung
- Nachschulung verkehrsauffälliger Fahrer



Die Analyse und Diskussion von Bremspunkten, Geschwindigkeit und anderen Daten kann Teil der Führerscheinausbildung oder der Nachschulung verkehrsauffälliger Fahrer sein. Hierfür ist gerade eine grafische Auswertung von Vorteil, um die Fahrer auf Gefahren aufmerksam zu machen.

Einige weitere, in [23] aufgelistete Ziele sind mit der momentan vorliegenden Version des Fahrsimulators noch nicht zu erreichen. Da der Fahrsimulator mit einer freien Lizenz verbreitet wird, ist eine Anpassung für diese Ziele jedoch möglich. Durch eine Integration weiterer Verkehrsteilnehmer wäre auch eine Analyse der Interaktion des Testteilnehmers mit weiteren Fahrzeugen möglich. Denkbar wäre hier beispielsweise die Untersuchung des Nachfahr- oder Überholverhaltens.

## 4.2 Gespeicherte Fahrdaten

Bei der Fahrt mit dem Fahrsimulator wird in einer Frequenz von 20Hz je ein Datensatz gespeichert. Damit liegt die Speicherfrequenz im Bereich 15 - 30Hz, die in anderen Studien verwendet wird [12][5]. Die Frequenz ist bei ausreichender Leistung des Computers statisch 20Hz, da in dem verwendeten Fahrsimulator der Berechnung der Szene im Datenmodell gegenüber dem Zeichnen der Szene Vorrang gewährt wird. Jeder Datensatz enthält die in der Tabelle 4.1 erläuterten Werte. Die X-, Y- und Z-Achse ist in der Abbildung 4.1 dargestellt. Die Werte der Achsen steigen jeweils in die Richtung, in die die Pfeilspitze zeigt.

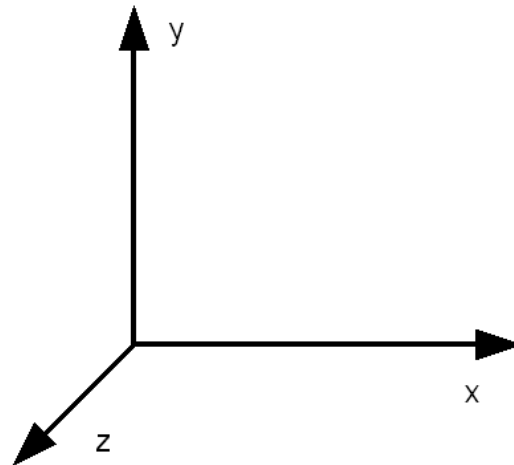


Abbildung 4.1: Zweidimensionale Darstellung der drei Dimensionen im Fahrsimulator

Die Datensätze können in einer für Menschen lesbaren Form oder in einer speichereffizienteren Form gespeichert werden. In der menschenlesbaren Form liegen die Daten in durch Doppelpunkte getrennte Werte vor. In der speichereffizienteren Form liegen die Daten als serialisiertes Objekt der Klasse *ArrayList* vor, das jeden Datensatz enthält. Beide Formen der Analysedaten können mit dem Analyseprogramm, welches ab Kapitel 4.4 beschrieben wird, ausgelesen werden, da eine eindeutige Schnittstelle definiert wurde. Die Schnittstelle zum Lesen dieser Daten heißt *IAnalyzationDataReader* und wird in Kapitel 4.4.3 näher definiert.

Wert	Form	Erklärung
Position auf der X-Achse	Fließkommazahl	Speichert die Position der Fahrzeugmitte auf der X-Achse in der virtuellen Welt
Position auf der Y-Achse	Fließkommazahl	Speichert die Position der Fahrzeugmitte auf der Y-Achse in der virtuellen Welt
Position auf der Z-Achse	Fließkommazahl	Speichert die Position der Fahrzeugmitte auf der Z-Achse in der virtuellen Welt
Geschwindigkeit	Fließkommazahl	Speichert die Fahrzeuggeschwindigkeit in km/h
Aktuelle Zeit	Zeit in Millisekunden nach dem 1.1.1970 als natürliche Zahl	Speichert die aktuelle Zeit, um in Analysen die Zeit zwischen zwei Datensätzen korrekt beachten zu können
Lenkerstellung	Fließkommazahl -1 bedeutet ganz links eingeschlagen 1 bedeutet ganz rechts eingeschlagen 0 bedeutet geradeaus	Speichert die Stellung des Lenkrades
Pedalstellung	Fließkommazahl 0 bedeutet kein Gas 1 bedeutet Vollgas.	Speichert die Position des Gaspedals
Bremse	Boolscher Wert. 1 bedeutet Bremse gedrückt, 0 bedeutet Bremse nicht gedrückt.	Speichert ob die Bremse betätigt wird
Entfernung zur Ideallinie	Fließkommazahl als Entfernung in Zentimetern	Speichert die Entfernung zur Ideallinie der Fahrbahn, oder „INFINITY“, wenn kein Wert ermittelt werden konnte
Marke 1 gesetzt	Boolscher Wert True bedeutet Marke gesetzt, False bedeutet Marke nicht gesetzt.	Speichert ob die Marke 1 in diesem Moment gesetzt wird
Marke 2 gesetzt	Boolscher Wert True bedeutet Marke gesetzt, False bedeutet Marke nicht gesetzt.	Speichert ob die Marke 2 in diesem Moment gesetzt wird
Marke 3 gesetzt	Boolscher Wert True bedeutet Marke gesetzt, False bedeutet Marke nicht gesetzt.	Speichert ob die Marke 3 in diesem Moment gesetzt wird
Runde	Ganze Zahl	Speichert die Nummer der Runde, in der sich das Fahrzeug aktuell befindet

### 4.3 Umsetzung der Analyse

In der Literatur werden viele unterschiedliche Fahrdaten zur Analyse herangezogen, wie in dem Kapitel 3 zu lesen ist. Die häufiger analysierten Daten sind aus [11], [10], [8] und [11] entnommen und werden für die Analyse der Fahrdaten in dieser Arbeit genutzt. Zusätzlich wird die Zeit gemessen, die sich ein Fahrer abseits der Fahrbahn befindet, sowie die Häufigkeit des Verlassens der Fahrbahn. Im Folgenden werden die einzelnen Analysemethoden erläutert, welche in dem ab Kapitel 4.4 Analyseprogramm implementiert wurden.

#### 4.3.1 Mittlere Distanz zur Ideallinie

Die Distanz zur Ideallinie lässt sich aus der Fahrzeugposition und der Position der Ideallinie berechnen. Die Spurbreite muss während der Messung konstant bleiben, damit die mittlere Distanz zur Ideallinie aussagekräftig ist [10]. Jedes Feld auf der Karte, das ein Straßenstück enthält, bekommt eine Ideallinie zugewiesen. Bei einem geraden Straßenstück wird die Ideallinie durch die Zwei-Punkt-Form einer Geraden beschrieben. Die Kurven in dem Fahrsimulator folgen nicht explizit einer Funktion. Deshalb kann die Ideallinie ebenfalls nicht als Funktion angegeben werden. Stattdessen wird den Kurven eine Menge an Punkten zugeordnet, die eine Ideallinie beschreiben sollen. Die Punkte werden mit Geraden verbunden, so dass eine Kurve approximiert wird. Der Abstand von der Ideallinie wird daran identifiziert, zu welcher Geraden zwischen diesen Punkten das Fahrzeug am nächsten ist und welchen Abstand das Fahrzeug zu dieser Geraden hat. Da das Fahrzeug in dem Simulator anhand seines Mittelpunkts bestimmt wird, wird hier eine Formel zur Berechnung des Abstands zwischen einem Punkt und einer Geraden in der Ebene aus [21] verwendet.

Die Zwei-Punkt-Form der Geraden wird für die Berechnung des Abstands zur Ideallinie zunächst in die folgende Form gebracht.

$$a\bar{x} + b\bar{y} + c = 0 \quad (1)$$

Dabei sind  $a$ ,  $b$  und  $c$  konstante Werte. Ferner sind  $\bar{x}$  und  $\bar{y}$  die Koordinaten von Punkten auf der Geraden. Die Punkte sind  $\bar{x} = (x_1, x_2)$ ,  $\bar{y} = (y_1, y_2)$ . Die Konstanten werden wie folgt berechnet:

$$a = x_1 - x_2 \quad (2)$$

$$b = y_1 - y_2 \quad (3)$$

$$c = -(ax_1 + by_1) \quad (4)$$

Die Position des Fahrzeuges wird in dem Fahrsimulator mit der Position der Fahrzeugmitte identifiziert. Der Abstand der Fahrzeugmitte zur Ideallinie wird im Falle einer Geraden wie folgt ermittelt:

$$d = \frac{ax_f + by_f + c}{\sqrt{a^2 + b^2}} \quad (5)$$

Dabei sind  $x_f$  und  $y_f$  die X- und Y-Koodrinaten der Fahrzeugmitte. Das Ergebnis ist eine positive oder negative Zahl. Der Betrag dieser Zahl gibt den Abstand zur Ideallinie an, während das Vorzeichen angibt, ob sich das Fahrzeug links ( $< 0$ ) oder rechts ( $> 0$ ) von der Ideallinie befindet.

In der Analyse wird der Median und der Durchschnitt des Abstandes zur Ideallinie ermittelt. Für diese Aufgabe ist ausschließlich der Betrag des Ergebnisses zu betrachten, da sonst durch die negativen Zahlen das Ergebnis verfälscht würde. Zur Analyse liegt eine Menge von Messwerten  $M = m_1, m_2, m_3, \dots, m_n$  vor. Der Durchschnitt des Abstandes von der Ideallinie berechnet sich wie folgt:

$$r_1 = \frac{\sum_{i=1}^n m_i}{n} \quad (6)$$

Der Median ist der sogenannte Zentralwert. Um den Median berechnen zu können, muss die Menge  $M$  sortiert sein. Der Median wird wie folgt berechnet:

$$r_2 = \begin{cases} m_{\frac{n+1}{2}} & \text{wenn } n \text{ ungerade} \\ \frac{1}{2} * (m_{\frac{n}{2}} + m_{\frac{n}{2}+1}) & \text{wenn } n \text{ gerade} \end{cases} \quad (7)$$

Zur Veranschaulichung ist die mittlere Distanz zur Ideallinie in Abbildung 4.2 grafisch dargestellt.

Wenn ein Fahrer die Fahrbahn verlässt, liegen keine Informationen zu der Distanz zur Ideallinie vor, da durch die Zuordnung jeweils einer Ideallinie zu jedem Straßenstück nicht mehr performant und zuverlässig identifiziert werden kann, welche Ideallinie für den Fahrer relevant ist. Daher wird der letzte Wert der Distanz von der Ideallinie beibehalten, wenn der Fahrer die Fahrbahn verlässt. Dadurch ist gewährleistet, dass der höchstmögliche Wert für das letzte befahrene Straßenstück als Distanz zur Ideallinie gewertet wird.

Eine hohe Distanz zur Ideallinie kann auf schlechte Fahrleistung schließen. Es ist jedoch zu beachten, dass Fahrer bei breiten Straßen durchaus gewollt auf einer Seite der Fahrspur fahren und nicht möglichst mittig.

#### 4.3.2 Standardabweichung der mittleren Distanz zur Ideallinie

Die Standardabweichung der mittleren Distanz zur Ideallinie wird mit Hilfe der Formel der Stichprobenstandardabweichung berechnet (Formel 8). Ebenso wie bei der mittleren Distanz zur Ideallinie muss die Fahrbahnbreite konstant sein, damit die Standardabweichung aussagekräftig ist [10].

$$s = \sqrt{\frac{1}{n-1} * \sum_{j=1}^n (x_j - \bar{x})^2} \quad (8)$$

$\bar{x}$  ist die durchschnittliche Distanz zur Ideallinie bzw. dessen Median, welche in Abschnitt 4.3.1 beschrieben werden. Zur Veranschaulichung ist die Standardabweichung der mittleren Distanz zur Ideallinie in Abbildung 4.2 grafisch dargestellt. Eine hohe Standardabweichung zur mittleren Distanz von der Ideallinie bedeutet, dass der Fahrer unsicher fährt, da das Fahrzeug nicht gleichmäßig in der Spur fährt.

#### 4.3.3 Mittlere seitliche Geschwindigkeit

Die mittlere seitliche Geschwindigkeit ist eine Methode, die Spursicherheit des Fahrers zu bewerten [12]. Führt das Fahrzeug in der Spur parallel zur Ideallinie, ist die seitliche Geschwindigkeit gleich null. Je schneller sich das Fahrzeug während der Fahrt quer zum Straßenverlauf bewegt, desto höher ist die seitliche Geschwindigkeit.

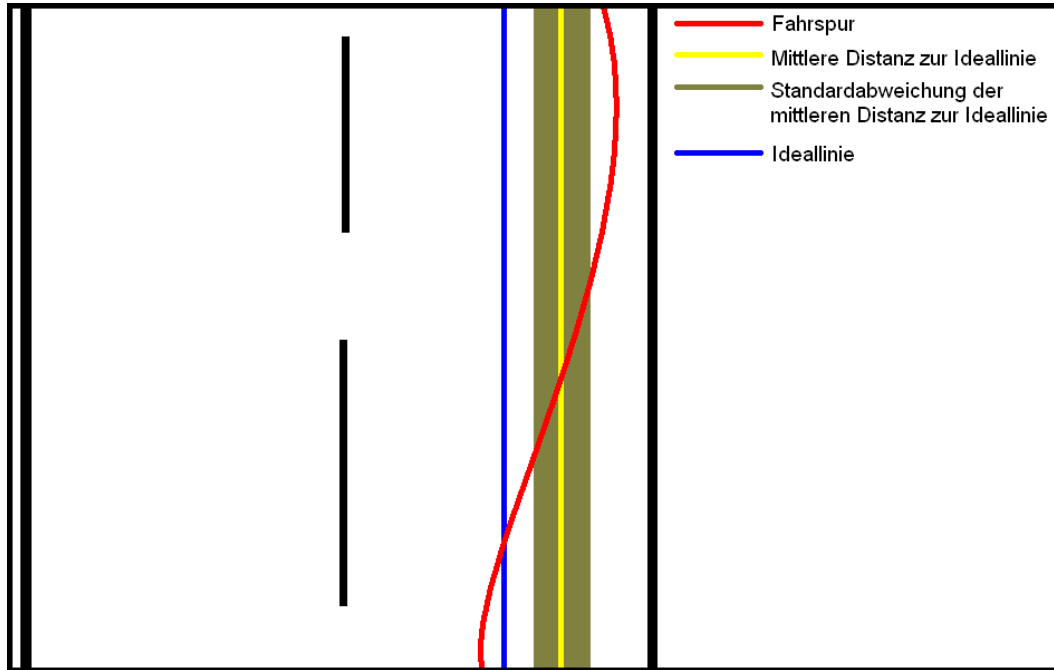


Abbildung 4.2: Distanz zur Ideallinie und deren Abweichung

Die seitliche Geschwindigkeit ist durch die Differenz der Distanz zur Ideallinie über zwei Datensätze zu berechnen. Dazu werden die zeitliche Differenz sowie die Entfernung zur Ideallinie verwendet.

$$\frac{d}{t} \quad (9)$$

$d$  ist der seitlich zurückgelegte Weg, also die Differenz der Distanz zur Ideallinie.  $t$  ist der zeitliche Unterschied der Datensätze. Durch Umwandlung der Einheiten kann daraus die seitliche Geschwindigkeit in m/s berechnet werden. Auch für diesen Messwert werden der Durchschnitt und der Median mit den Formeln 6 und 7 berechnet. Die Menge  $M$  entspricht dabei der Menge der seitlichen Geschwindigkeiten. Eine hohe seitliche Geschwindigkeit kann ein Indikator für schlechte Fahrleistungen sein, da das Fahrzeug sich wenig parallel zur Fahrbahn bewegt.

#### 4.3.4 Mittlere Log-Zeit bis zur Spurbegrenzung

Die Zeit bis zur Spurbegrenzung  $t_{linie}$  benötigt die seitliche Geschwindigkeit  $v_{seite}$  und die Distanz zur Ideallinie aus zwei Datensätzen  $d_{t1}$  und  $d_{t2}$  ebenso wie die Spurbreite  $d_{spur}$ . Zur Berechnung der Zeit bis zur Spurbegrenzung müssen die drei Fälle  $d_{t2} - d_{t1} < 0$ ,  $d_{t2} - d_{t1} > 0$  und  $d_{t2} - d_{t1} = 0$  betrachtet werden. Diese Fälle ergeben sich daraus, dass das Fahrzeug sich in der Spur entweder der inneren oder der äußeren Spurbegrenzung nähern kann oder sich keiner Begrenzung nähert.

$$d_{t2} - d_{t1} > 0:$$

$$d_{t2} + t_{linie} * v_{seite} = d_{spur} \Leftrightarrow t_{linie} = \frac{d_{spur} - d_{t2}}{v_{seite}} \quad (10)$$

$$d_{t2} - d_{t1} < 0:$$

$$d_{t2} + t_{linie} * v_{seite} = 0 \Leftrightarrow t_{linie} = \frac{d_{t2}}{v_{seite}} \quad (11)$$

$$d_{t2} - d_{t1} = 0:$$

$$t_{linie} = \infty \quad (12)$$

Daraus ergibt sich die folgende Fallunterscheidung:

$$t_{linie} = \begin{cases} \frac{d_{spur} - d_{t2}}{v_{seite}} & \text{wenn } d_{t2} - d_{t1} > 0 \\ \frac{d_{t2}}{v_{seite}} & \text{wenn } d_{t2} - d_{t1} < 0 \\ \infty & \text{wenn } d_{t2} - d_{t1} = 0 \end{cases} \quad (13)$$

Wie in Kapitel 3 erwähnt, wird der Logarithmus des Betrags genutzt, um den Einfluss der hohen Werte zu mindern, wenn die seitliche Geschwindigkeit ihre Richtung wechselt.

$$\log |t_{linie}| \quad (14)$$

Das Einsetzen der Menge aller Log-Zeiten als Menge  $M$  in die Formeln 6 und 7 ermöglichen die Berechnung des Durchschnitts und des Medians. Eine geringe Zeit bis zur Spurbegrenzung entspricht in ihrer Auswirkung einer hohen seitlichen Geschwindigkeit und kann ein Indikator für schlechte Fahrleistungen sein [12].

#### 4.3.5 Mittlere Geschwindigkeit

Von der Geschwindigkeit wird der Median und der Durchschnitt berechnet. Dazu werden die Formeln 6 und 7 verwendet. Die Menge  $M$  entspricht dann der Menge der Geschwindigkeiten.

#### 4.3.6 Standardabweichung der Geschwindigkeit

Die Standardabweichung der Geschwindigkeit wird ähnlich wie die Standardabweichung der mittleren Distanz zur Ideallinie berechnet. Dazu wird die Formel 8 verwendet. Für  $\bar{x}$  wird in diesem Falle der wie in 4.3.5 berechnete Durchschnitt und Median verwendet. Eine hohe Standardabweichung der Geschwindigkeit kann aggressives Fahren zeigen [5].

#### 4.3.7 Mittlere absolute Beschleunigung

Der Betrag der Beschleunigung umfasst die Verzögerung ebenso wie die Erhöhung der Geschwindigkeit des Fahrzeugs. Für die Messung werden die Geschwindigkeitswerte  $v_1$  und  $v_2$  von zwei Datensätzen mit den Zeitinformationen  $t_1$  und  $t_2$  benötigt. Die Beschleunigung  $a$  wird wie folgt berechnet:

$$a = \frac{v_2 - v_1}{t_2 - t_1} \quad (15)$$

Die übliche Einheit der Beschleunigung  $a$  ist  $\frac{m}{s^2}$ . Auch für die Beschleunigung wird der Durchschnitt und der Median mit Hilfe der Formeln 6 und 7 berechnet. Die Menge  $M$  ist dabei die Menge der Beschleunigungen. Eine starke mittlere Beschleunigung kann aggressives Fahren zeigen [5].

#### 4.3.8 Standardabweichung der Lenkerstellung

Die Standardabweichung der Lenkerstellung wird analog zu der Standardabweichung der Distanz zur Ideallinie in Abschnitt 4.3.2 berechnet. Dazu muss die mittlere Lenkerstellung berechnet werden und als  $\bar{x}$  in die Formel 8 eingefügt werden. Die mittlere Lenkerstellung wird wie die mittlere Geschwindigkeit durch den Median bzw. den Durchschnitt ermittelt. Auf geraden Strecken kann die Standardabweichung der Lenkerstellung zeigen, wie intensiv der Kurs des Fahrzeugs korrigiert wurde.

#### 4.3.9 Frequenz der Steuerkorrektur

Eine Steuerkorrektur ist, angelehnt an [12], eine monotone Änderung des Lenkwinkels um mehr als 1 Grad, welche mehr als 0,3 Sekunden anhält. Die Anzahl der Steuerkorrekturen  $k$  wird durch die Dauer der Simulationsfahrt geteilt.  $t_1$  ist der Zeitpunkt zu Beginn der Messungen und  $t_2$  ist der Zeitpunkt zum Ende der Messungen.

$$F_{\text{Steuerkorrektur}} = \frac{k}{t_2 - t_1} \quad (16)$$

#### 4.3.10 Standardabweichung der Gaspedalstellung

Analog zur Standardabweichung der Lenkerstellung in Abschnitt 4.3.8 wird für die Berechnung der Standardabweichung der Pedalstellung zunächst die mittlere Gaspedalstellung in Form des Medians und des Durchschnitts berechnet und dann jeweils als  $\bar{x}$  in die Formel 8 eingesetzt. Die Menge  $X$  ist in diesem Falle die Menge der quantifizierten Gaspedalstellungen.

#### 4.3.11 Frequenz der Geschwindigkeitskorrektur

Eine Geschwindigkeitskorrektur ist, angelehnt an [12], eine monotone Änderung von 0,5% mit einer Dauer von mindestens 0,3 Sekunden. Die Berechnung findet analog zu der Berechnung der Frequenz der Steuerkorrektur in Abschnitt 4.3.9 statt.

#### 4.3.12 Zeit abseits der Fahrbahn

Die Zeit, während der sich das Fahrzeug abseits der Straße befindet kann ein wichtiger Wert sein, um Datensätze zu finden, die ungeeignet sind. Es gibt Personen, die mit einem Fahrsimulator auch nach der Eingewöhnungsphase nicht zurecht kommen und zum Teil mehr Zeit abseits der Straße verbringen als darauf. Ein Fahrer wird als abseits der Straße gewertet, wenn er mindestens 100 ms nicht auf der Fahrbahn verbracht hat. Dieser Puffer wurde eingefügt, damit der Fahrer erst als abseits der Straße gilt, wenn er zum Großteil nicht mehr auf der Straße fährt.

#### 4.3.13 Häufigkeit des Verlassens der Fahrbahn

Die Häufigkeit, mit der ein Fahrer mit seinem Fahrzeug die Fahrbahn verlässt wird ähnlich wie die Zeit abseits der Fahrbahn eingesetzt.

#### 4.3.14 Quotient aus Spurhaltung und Geschwindigkeit

Bei Fahrten, in denen die Geschwindigkeit vom Fahrer selbst bestimmt werden kann, konnte beobachtet werden, dass dieser bei einer Ablenkung die Geschwindigkeit häufig senkt. Dies führt dazu, dass die Spur wesentlich leichter zu halten ist, als bei gleichbleibender Geschwindigkeit. Um diese Fahrleistung trotz variabler Geschwindigkeit messen zu können, wird die Standardabweichung der durchschnittlichen Distanz zur Ideallinie mit der Geschwindigkeit wie in Formel 17 zu sehen in Relation gesetzt.

$$q = \frac{\bar{\sigma}}{\bar{v}} \quad (17)$$

$\bar{\sigma}$  ist die Standardabweichung der durchschnittlichen Distanz zur Ideallinie.  $\bar{v}$  ist die durchschnittliche Geschwindigkeit. Ist der Wert  $q$  hoch, bedeutet das, dass der Fahrer die Spur bei gleichbleibender Geschwindigkeit schlechter hält bzw. die Spurhaltung nur durch eine geringere Geschwindigkeit aufrechterhalten konnte. Ein niedrigerer Wert  $q$  ist ein Indikator für eine bessere Fahrleistung. Zu diesem Zeitpunkt kann kein konkreter Wert angegeben werden, der anzeigt, wie stark ein Fahrer bei gegebenen Wert  $q$  abgelenkt ist. Hierzu müssten in weiteren Studien Daten gesammelt werden. In einer Studie zur Bewertung der Ablenkung in Kapitel 6.2 steigt  $q$  bei der Bearbeitung von Nebenaufgaben jedoch signifikant an. Dadurch wird deutlich, dass mit Hilfe von  $q$  der Grad der Ablenkung zumindest relativ zu einem Vergleichswert  $q$  aussagekräftig ist.

### 4.4 Architektur

Zur Analyse der Fahrdaten müssen zunächst die relevanten Daten der Fahrt aufgezeichnet werden. Dazu wurde direkt in den Fahrsimulator eingegriffen. Die in Abbildung 2.1 dargestellte Architektur des Fahrsimulators wurde dabei wie in der Abbildung 4.3 dargestellt ergänzt.

Die Komponente *ControlComponent* nutzt die Schnittstelle *IAnalyzationDataWriter*, um die Daten der Simulationsfahrt zu sichern. Durch die Realisierung als Schnittstelle sind weitere Implementierungen der Datenspeicherung denkbar. Hierauf wird in Kapitel 4.6.2 noch einmal Bezug genommen. Mit Hilfe der Schnittstelle werden die Daten der Simulationsfahrt gespeichert. Die zu sichernden Informationen sind in Tabelle 4.1 dargestellt. Der Zustand des Fahrzeugs, von dem u.a. die X-, Y- und Z-Position sowie die Pedalposition gespeichert werden, wird aus der Komponente *Entities* ausgelesen. Weitere Informationen, wie das Setzen eines Markers, wird der Komponente *ControlComponent* durch die *ActionHandler* mitgeteilt.

Das Analyseprogramm hat eine eigene, von dem Simulator losgelöste Architektur, welche in Form eines Klassendiagramms in Abbildung 4.4 dargestellt ist. Das Klassendiagramm enthält ausschließlich ausgewählte Operationen und Attribute. Die Architektur wurde in die drei Ebenen Model, View und Control (MVC) eingeteilt. Bei dem MVC-Konzept wird das Softwaresystem in die drei o.g. Ebenen aufgeteilt.

#### 4.4.1 View-Ebene

Die View-Ebene wird durch die Klassen *AnalyzerGUI*, *MapPanel* und *ResultsPanel* realisiert. Die Klasse *AnalyzerGUI* stellt die Verbindung zur Control-Ebene her und gibt diese Referenz zur Klasse *AnalyzerBackend* an die zwei Panels weiter. Sie stellt zudem ein



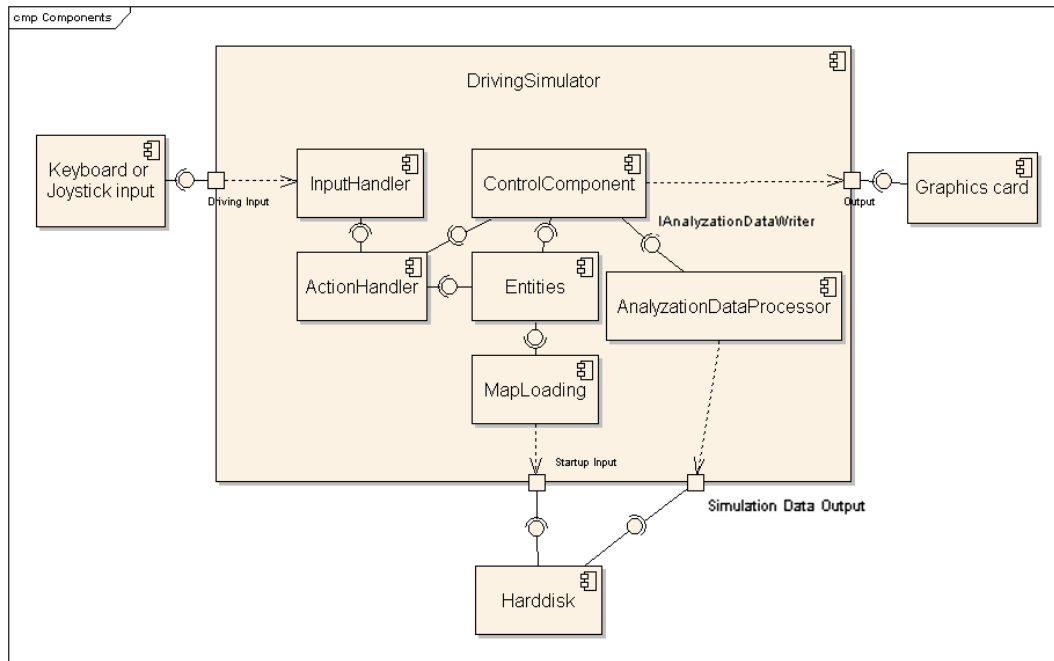


Abbildung 4.3: Modifizierte Systemarchitektur des Fahrsimulators

Fenster bereit, in das die beiden Klassen *MapPanel* und *ResultsPanel* eingebettet werden. Die Klasse *MapPanel* stellt anhand der Daten, die die Control-Ebene liefert, die befahrene Karte sowie die gefahrene Strecke dar. Außerdem werden die Bremspunkte und die Punkte angezeigt, an denen Marker gesetzt wurden. Die Klasse *ResultsPanel* stellt die Daten, die die Control-Ebene liefert, numerisch dar. Abbildung 4.5 zeigt die Oberfläche des Programms, mit der Darstellung der Klasse *MapPanel* auf der linken Seite und der Darstellung der Klasse *ResultsPanel* auf der rechten Seite.

#### 4.4.2 Control-Ebene

Die Control-Ebene wird durch die Klasse *AnalyzerBackend* repräsentiert. Sie berechnet anhand der Daten, die die Model-Ebene bereitstellt, die numerischen Werte und bereitet die Daten für die View-Ebene auf. Die Klasse *AnalyzerBackend* stellt Operationen bereit, um statistische Daten für die gesamte Fahrt oder einzelne Runden zu berechnen. Ferner besteht die Möglichkeit, Statistiken über die Daten, die in einer bestimmten Runde gemessen wurden, zu berechnen. Um wiederholte Berechnungen zu vermeiden werden die Ergebnisse zwischengespeichert. Die Berechnungen der einzelnen Analysedaten wurden in Module ausgelagert, deren Aufgabe jeweils die Berechnung eines einzelnen Ergebnisses ist, um das Programm erweiterbar und wartbar zu gestalten. Jedes Modul verarbeitet die ihm zur Verfügung gestellten Daten völlig selbstständig. In Kapitel 4.5 wird die Entwicklung eines solchen Moduls näher erläutert. Die Berechnung durch eigenständige Module, mit denen ein möglichst breites Spektrum an Daten verarbeitet werden sollen, kann in der Geschwindigkeit nicht mit einem System konkurrieren, das für die Lösung eines bestimmten Problems zugeschnitten wurde. Um den Performanzverlust durch den modularen Ansatz zu bewerten, wurde ein Test durchgeführt, der in Kapitel 4.7 dokumentiert ist.

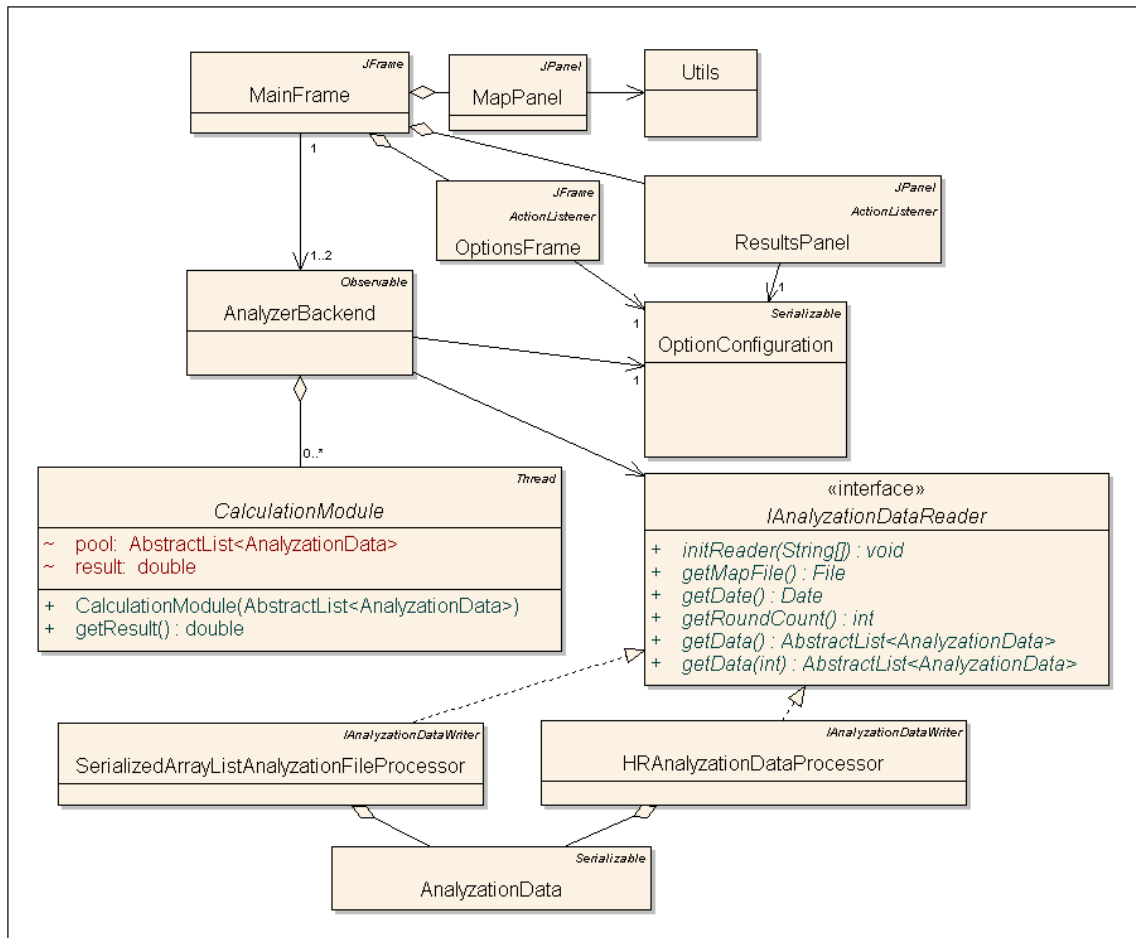


Abbildung 4.4: Klassendiagramm des Analyseprogramms

#### 4.4.3 Model-Ebene

In der Model-Ebene werden die Datensätze, die die Simulationsfahrt repräsentieren, ausgelesen und zur Bearbeitung und Weiterleitung durch die Control-Ebene bereitgestellt. Die Model-Ebene nutzt die Schnittstelle *IAnalyzerDataReader*, um verschiedene Implementierungen der Speicherung der Daten zu ermöglichen. Die Ablage der Daten in einem menschenlesbaren Format auf Festplatte ist durch die Klasse *HumanReadableAnalysisDataFileProcessor* realisiert, während eine speichereffizientere Sicherung durch die Klasse *SerializedArrayListAnalysisDataFileProcessor* umgesetzt ist. Beide Realisierungen der Schnittstelle *IAnalyzerDataReader* nutzen eine geordnete Menge an Objekten der Klasse *AnalysisData*, die nicht als *LinkedHashSet* realisiert ist, obwohl dies naheliegender wäre, da ein *LinkedHashSet* eine geordnete Menge ohne Duplikate darstellt, sondern als *ArrayList*, um den wahlfreien Zugriff zu ermöglichen. Dieser wird bei der Berechnung der statistischen Daten verwendet. Die Klasse *ArrayList* stellt eine geordnete Menge mit Duplikaten dar.

## 4.5 Implementierung

Ein zentraler Bestandteil der Implementierung ist die Berechnung der Statistiken aus den Daten einer Simulationsfahrt. Um das Analyseprogramm möglichst offen für Erweiterun-

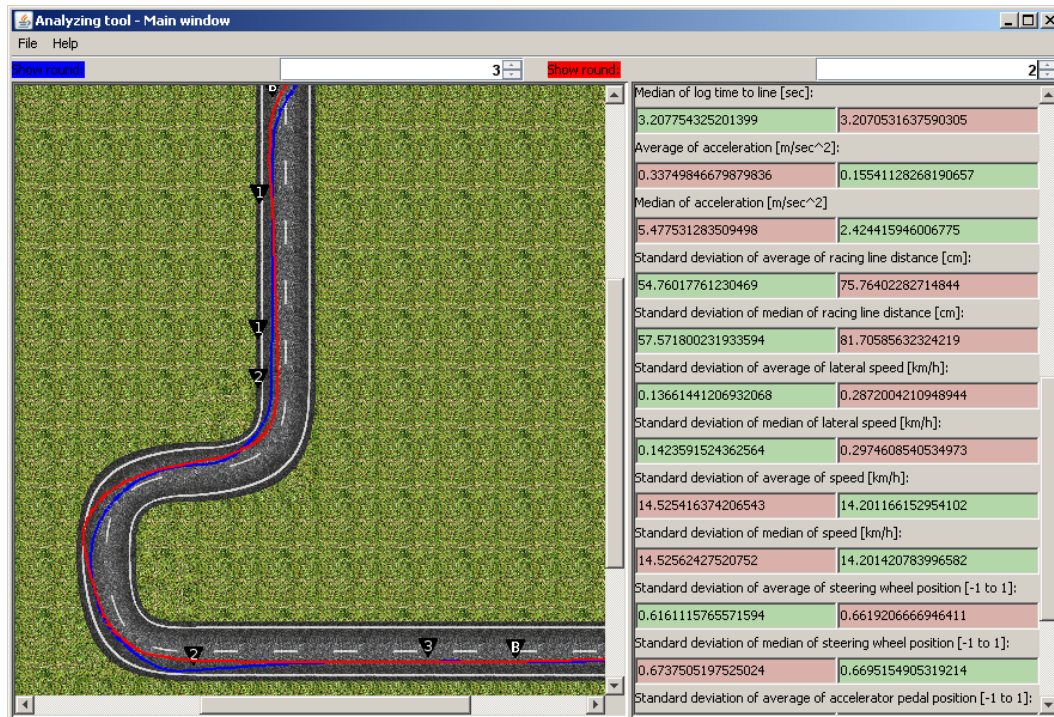


Abbildung 4.5: Bildschirmfoto der Programmoberfläche

gen zu gestalten, wurde ein modularer Ansatz für die Berechnungen gewählt. Jede Berechnung wird von einem eigenen Modul ausgeführt, das alle benötigten Schritte selbst durchführt. Durch die flexible Gestaltung der Module können auch Ergebnisse anderer Module als Wert in die Berechnung mit einfließen. Somit ist eine parallele Berechnung der meisten benötigten Daten möglich, von der gerade moderne Mehrkernprozessoren profitieren. Durch die kleinen Module, die jeweils eine eigene Berechnung durchführen, ist das Programm leichter zu warten, da jede Berechnung als abgeschlossene Einheit auftritt. Nachteil der Modularisierung ist, dass der Datenbestand häufiger als notwendig durchlaufen wird, weil jedes Modul mindestens einmal alle Datensätze durchläuft. Der modulbasierende Ansatz wurde daher mit einem direkteren Ansatz in Kapitel 4.7 verglichen.

Als Beispiel für die Entwicklung eines Berechnungs-Moduls wird im Folgenden das Modul zur Berechnung der durchschnittlichen Distanz zur Ideallinie erläutert.

#### 4.5.1 Beispielmodul „Standardabweichung der Distanz zur Ideallinie,,

Der gesamte Quelltext des Moduls ist in dem Quelltextauszug „Listing 1“ zu finden.

Listing 1: Das Modul „Standardabweichung der Distanz zur Ideallinie“

```

1 public class Rld_sd extends CalculationModule {
2     double centre;
3     double lastRld = 0d;
4
5     public Rld_sd( AbstractList<AnalyzationData> pool ,
6         double centre ) {
7         super( pool );

```

```

8      this.centre = centre;
9  }
10
11  public void run() {
12      ArrayList<Double> rld = new ArrayList<Double>();
13      for (int i = 0; i < pool.size(); i++) {
14          double currentRld = pool.get(i).
15              getRacingLineDistance();
16          if (Double.isInfinite(currentRld)) {
17              currentRld = lastRld;
18          } else {
19              lastRld = currentRld;
20          }
21          rld.add(new Double(currentRld));
22      }
23      float sum = 0;
24      for (int i = 0; i < rld.size(); i++)
25          sum += Math.pow(
26              (float) (rld.get(i) - centre), 2);
27      result = Math.sqrt(sum / (float) (rld.size() - 1));
28  }
29 }

```

Der Konstruktor nimmt in Zeile 4 als Parameter den Datenpool sowie ein Zentrum entgegen, um welches die Standardabweichung berechnet werden soll. Es ist zu beachten, dass die Variable *pool* nicht verändert werden darf, da hier eine Referenz übergeben wird und keine Kopie. Die Übergabe einer Kopie an jedes Modul würde die Verarbeitungszeit gerade bei großen Datenmengen deutlich verlängern. Da das Zentrum, um welches die Standardabweichung berechnet wird, als Parameter übergeben wird, kann die Klasse die Standardabweichung von dem Median ebenso wie die Standardabweichung vom Durchschnitt der Distanz zur Ideallinie berechnen. Dies schließt jedoch ein, dass der Median bzw. der Durchschnitt schon vorher berechnet wurden.

Jedes Modul erbt von der Klasse *CalculationModule*, welche selbst von der Klasse *Thread* erbt. Somit ist sichergestellt, dass alle Module nebenläufig ausgeführt werden können. Als Folge muss die Methode *run()* überschrieben werden, und die eigentliche Berechnung durchführen. In diesem Beispiel werden zunächst die relevanten Daten in eine eigene Liste aufgenommen, bevor diese zur Berechnung der Standardabweichung um das gegebene Zentrum verwendet werden, wie in Kapitel 4.3.2 beschrieben.

#### 4.5.2 Lesemodule

Die Lesemodule werden im Analyseprogramm eingesetzt, um die Daten, die in den Simulationsfahrten erzeugt und gespeichert wurden, zur Verarbeitung in den Speicher zu laden. Es wurden zwei Lesemodule entwickelt. Eines liest die Daten aus einer Datei mit für Menschen lesbaren Daten aus. Das Modul *HumanReadable* arbeitet dabei mit durch Doppelpunkte getrennten Werten. Das zweite Modul namens *SerializedArrayList* liest die Daten aus einem in eine Datei serialisiertes Objekt aus, wodurch die Speicherung der Daten direkt der virtuellen Maschine von Java überlassen wird. Dazu wird ein Objekt der

Klasse *ArrayListDataPoolObject* genutzt, welche die konkreten Fahrdaten, die als Objekt der Klasse *ArrayList<AnalyzationData>* vorliegen, kapselt und mit der eingesetzten Karte und dem Datum der Fahrt als Zusatzinformationen anreichert. Das serialisierte Objekt der Klasse *ArrayListDataPoolObject* wird dabei beim Aufruf des Lesemoduls mit der folgenden Methode deserialisiert:

Listing 2: Auslesen eines serialisierten Objekts

```

1  @Override
2  public void initReader(String[] connection)
3      throws InvalidParameterException {
4      if (connection != null && connection.length >= 1 &&
5          new File(connection[0]).getAbsolutePath().canWrite())
6          {
7              inFile = new File(connection[0]).getAbsolutePath();
8          } else
9              return;
10
11     try {
12         ObjectInputStream in = new ObjectInputStream(
13             new FileInputStream(inFile));
14         Object tmp = in.readObject();
15         if (!(tmp instanceof ArrayListDataPoolObject))
16             throw new InvalidParameterException(
17                 ResourceBundle.get("exception_not_sal_data"));
18         ob = (ArrayListDataPoolObject) tmp;
19     } catch (Exception e) {
20         throw new InvalidParameterException(
21             ResourceBundle.get("exception_not_sal_data"));
22     }

```

Das Objekt wird aus einer Datei ausgelesen und kann direkt umgewandelt werden. Wenn bei dem Vorgang Fehler auftreten, wird eine entsprechende Fehlermeldung ausgegeben, die die grafische Oberfläche verarbeiten und an den Benutzer weiterleiten kann. Auf das Objekt *ob* kann nun innerhalb des Lesemoduls zugegriffen werden, insbesondere auf die Methode *getPool()*, die die gekapselten Fahrdaten zurück gibt. Diese konkrete Umsetzung des Auslesens der Daten ist für die Control-Ebene des Analyseprogramms nicht relevant. Für die Klasse *AnalyzerBackend* ist relevant, dass die Klasse *SerializedArrayListAnalyzationFileProcessor* die Schnittstelle *IAnalyzationDataReader* implementiert und somit die gewünschte Funktionalität bietet. Dieser Aufbau ermöglicht die Modulerstellung und den Austausch von Komponenten im System.

## 4.6 Erweiterungsmöglichkeiten

### 4.6.1 Erweiterung der Statistiken

Durch den modularen Aufbau der Berechnung der Statistiken, wie in Kapitel 4.5 beschrieben, können weitere, auch komplexere Statistiken, erstellt werden. Die aktuelle Version des Analyseprogramms berechnet Statistiken für die gesamte Fahrt sowie für einzelne



Runden. Es können beispielsweise Module erstellt werden, die, über einen anzugebenden Teil der Strecke die maximale Bremskraft messen. Es ist denkbar, die auszuwertende Strecke grafisch in der angezeigten Karte anzugeben. Durch die Selektion von Markern, die innerhalb der Simulationsfahrt gesetzt wurden, können Analysen für Teilstrecken erstellt werden, in der Studienteilnehmer beispielsweise eine Nebenaufgabe erfüllt hat, sowie für die Teilstrecken, in denen keine Nebenaufgabe erfüllt wurde. Dadurch können automatisiert die Unterschiede im Fahrverhalten zwischen den Momenten mit und ohne Nebenaufgaben analysiert werden. Dazu müsste die Klasse *AnalyzerBackend* um eine Methode erweitert werden, die zwei Zeitpunkte in Form von Objekten der Klasse *Date* entgegennimmt, und die Analysedaten zwischen diesen Zeitpunkten zurückgibt. Ein Beispiel dafür findet sich in dem Quelltextauszug „Listing 3“.

Listing 3: Beispielerweiterung für die Klasse *AnalyzerBackend*

```
1 public void calculateRange(Date from, Date to)
2     throws InvalidParameterException {
3     considerWholeDataSet = false;
4     pool = reader.getData();
5     ArrayList<AnalyzationData> rangePool =
6         new ArrayList<AnalyzationData>();
7     for (AnalyzationData ad : pool) {
8         if (ad.getDate().before(from))
9             continue;
10        if (ad.getDate().after(to))
11            break;
12        rangePool.add(ad);
13    }
14    calculateMaximumBrakingForce();
15 }
```

Über die Daten der angegebenen Zeitspanne könnte dann in der hinzuzufügenden Methode *calculateMaximumBrakingForce* durch ein Analysemodul, welches im Aufbau dem Modul im Quelltextauszug „Listing 1“ ähnelt, die maximale Bremskraft ermittelt werden. Die benötigten Daten stehen als Pedalposition schon in den Analysedaten zur Verfügung. Die daraus berechneten Daten sollten dann in einem Attribut der Klasse *AnalyzerBackend* gespeichert werden, um sie bei Bedarf an die Benutzeroberfläche weiterleiten zu können. Sollte die Menge der gespeicherten Daten in der Klasse *AnalyzerBackend* zu groß werden und die Übersichtlichkeit gefährden, können die Werte auch als Schlüssel-Wert-Paare gesichert werden. Durch diese und ähnliche Erweiterungen ist es möglich, noch weitere der in Kapitel 4.1 genannten Ziele der Simulation zu erfüllen.

Eine weitere Möglichkeit, das Analyseprogramm um sinnvolle Funktionen zu erweitern, ist einzelne Fahrdatensätze grafisch anzuzeigen. Es ist vorstellbar, bei der Fahrt mit der Maus über die grafische Darstellung der Karte, beim Berühren der Fahrlinie, die aktuellen Fahrdaten zu dem markierten Moment in der Fahrdatendatei anzuzeigen. So kann der Benutzer des Analyseprogramms die Fahrt anhand der einzelnen gespeicherten Daten nachvollziehen.

### 4.6.2 Erweiterung der Datenquellen

Die flexible Gestaltung des Einlesens der Simulationsdaten ermöglicht nicht nur eine speichereffiziente sowie eine für Menschen lesbare Form der Datensicherung. Dadurch, dass die Daten von jeder Klasse eingelesen werden können, die das Interface *IAnalyzerDataReader* implementiert, ist es möglich auch Daten von anderen Fahrsimulatoren zu analysieren, insofern diese alle Daten bereitstellen, welche die gewählten Module für ihre Berechnungen benötigen. Die Klasse, die die Daten einliest, muss das Attribut *result* mit einem Objekt einer Unterklasse der Klasse *AbstractList<AnalyzerData>* belegen, dessen enthaltene *AnalyzerData*-Objekte zeitlich aufsteigend sortiert sind. In dem Quelltextauszug „Listing 4“ ist beispielhaft aufgeführt, wie eine solche Klasse aufgebaut werden könnte. In diesem Beispiel wird davon ausgegangen, dass die Daten von dem anderen Fahrsimulator in einem Datenbankmanagementsystem gespeichert wurden. Die Daten könnten auch als Datei vorliegen oder beispielsweise auf einem entfernten System zur Verfügung gestellt werden. Im letzten Falle sollte über ein Zwischenspeichern der Daten nachgedacht werden.

Listing 4: Beispiel eines erweiterten Analysedaten-Lesemoduls

```

1 public class OtherSimulatorsDataReader
2     implements IAnalyzerDataReader {
3
4     @Override
5     public AbstractList<AnalyzerData> getData()
6         throws InvalidParameterException {
7         // Use the connection to the database to get the
8         // required data.
9         // Create and return a list containing the data
10        // of the simulation.
11    }
12
13    @Override
14    public AbstractList<AnalyzerData> getData(int round)
15        throws InvalidParameterException {
16        // Use the connection to the database to get the
17        // required data.
18        // Create and return a list containing the data
19        // of the round.
20    }
21
22    // The date when the data was created
23    @Override
24    public Date getDate() {
25        // Use the connection to the database to get the
26        // required data.
27    }
28
29    // The file of the map. This will need more attention
30    // if we are using other driving simulators.

```

```

31     @Override
32     public File getMapFile() {
33         // Use the connection to the database to get the
34         // required data.
35     }
36
37     // The amount of rounds driven in the simulation
38     @Override
39     public int getRoundCount() {
40         // Use the connection to the database to get the
41         // required data.
42     }
43
44     @Override
45     public void initReader(String[] connection)
46         throws InvalidParameterException {
47         // Establish a connection to the database
48         // using the given connection information
49     }
50 }

```

Die Auswertung von Daten ist nicht auf Simulationsfahrten beschränkt. Es ist beispielsweise für die Fahrerausbildung denkbar, ein Fahrschulfahrzeug mit Messinstrumenten auszustatten, welche die benötigten Daten während der realen Fahrt messen und speichern, was in einem anderen Kontext in [12] bereits durchgeführt worden ist. Hier wurden der Abstand zur Mittellinie, die Geschwindigkeit, die Lenkerposition, die Gaspedalposition sowie Videoaufzeichnungen des Fahrers und der Frontsicht gespeichert. Die weiteren möglichen Daten, das Setzen von Markern, die Betätigung der Bremse und die aktuelle Runde können leicht ermittelt werden. Die Position des Fahrzeugs in der X-, Y- und Z-Achse ließe sich berechnen, indem vorher eine Fahrstrecke und eine geografische Koordinate als Nullpunkt in einem räumlichen Koordinatensystem definiert wird. Anhand von Daten aus einem Global Positioning System (GPS) kann die Position des Fahrzeugs in dem Koordinatensystem bestimmt werden. Die so gewonnenen Daten können von dem Analyseprogramm durch ein geeignetes Lesemodul eingelesen und dann verarbeitet werden.

## 4.7 Technische Daten

Der größte Rechenaufwand des Analyseprogramms ist zeigt sich besonders an zwei Stellen: das Laden der relevanten Daten und deren Verarbeitung. Es wurden zwei Lesemodule erstellt, die in Kapitel 4.4.3 näher erläutert sind. Diese beiden Lesemodule wurden hinsichtlich ihrer Performanz getestet. Zudem wurde der modulare Ansatz der Berechnung der Analysedaten mit einem direkten Ansatz verglichen. Im direkten Ansatz wurden die zu berechnenden Analysewerte parallel berechnet. Dabei wurde versucht, die Daten effizient zu bearbeiten, so dass der Datenbestand nicht mehr als notwendig durchgearbeitet werden muss. Die Berechnung wurde, wie in Kapitel 4.5 erwähnt, durch einzelne Module realisiert. Dieser Ansatz soll die Wartbarkeit und Erweiterbarkeit des Programms sicherstellen. Ein modularer Ansatz, der möglichst viele Einsatzfelder abdecken soll, kann



Daten-sätze	Entspricht einer Fahrdauer von [min]	Ladedauer LM 1 [ms]	Ladedauer LM 2 [ms]	modulare Berechnung [ms]	direkte Berechnung [ms]
1200	1	182.2	22	60.2	42.2
2400	2	304.4	42	48.2	36.2
3600	3	478.8	56.2	56.2	52
4800	4	621	70	58	60
6000	5	769.86	86	84.2	58
12000	10	1506	178.2	128.2	98
36000	30	4526.6	596.8	492.6	364.4
72000	60	9027	1127.6	1027.6	767
144000	120	19231.6	2707.8	2497.4	1622.6
Daten-sätze	Entspricht einer Fahrdauer von [min]	Ladedauer pro Fahrmi-nute LM 1 [ms]	Ladedauer pro Fahrmi-nute LM 2 [ms]	Zeit / Fahrmi-nute Modular [ms]	Zeit / Fahrmi-nute Direkt [ms]
1200	1	182.2	22	60.2	42.2
2400	2	152.2	21	24.1	18.1
3600	3	159.6	18.73	18.73	17.33
4800	4	155.25	17.5	14.5	15
6000	5	153.8	17.2	16.84	12
12000	10	150.6	17.82	12.82	5.8
36000	30	150.89	19.89	16.42	3.27
72000	60	150.45	18.79	17.13	6.07
144000	120	160.26	22.57	20.81	6.39

LM 1: Lesemodul *HumanReadable*, LM 2: Lesemodul *SerializedArrayList*

Tabelle 4.2: Ergebnisse des Performanztests

jedoch einem direkten Ansatz, der speziell auf ein Problem zugeschnitten ist, nicht überlegen sein. Mit einem Performanztest wurde überprüft, inwiefern der modulare Ansatz langsamer ist als die direkte Berechnung.

Die Tabelle 4.2 zeigt die Ergebnisse des Performanztests für die Lesemodule und die Berechnungen. In den Abbildungen 4.6 und 4.7 ist die Dauer des Ladens und des Berechnens verschieden großer Datensätze abzulesen.

Das Einlesen der Datenmengen zeigt deutliche Unterschiede zwischen den beiden Lesemodulen *HumanReadable* und *SerializedArrayList*. Beide Lesemodule bieten eine Lesedauer, die linear mit der Datenmenge skaliert. Das Lesemodul *HumanReadable* war auf dem Testsystem in der Lage, je eine Minute Fahrzeit in ca. 150 ms einzulesen. Ebenso linear skalierte das Lesemodul *SerializedArrayList*, jedoch bedeutend schneller, mit ca. 20ms für eine Minute Fahrzeit.

Erst bei größeren Datenmengen zeigte sich, dass die direkte Berechnung erwartungsgemäß schneller fertig war als die Verarbeitung mit den Analysemodulen. Der Unterschied ist jedoch relativ gering. Mit Hilfe des direkten Ansatzes konnten die Daten einer fiktiven zweistündigen Fahrt mit dem Simulator innerhalb von 1622,6 ms berechnet werden, während der modulare Ansatz für die gleichen Daten 2497,4 ms benötigte. Fahrten

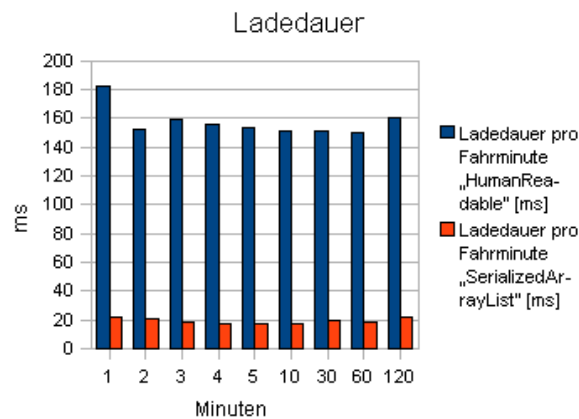


Abbildung 4.6: Ergebnisse des Performanztests zur Ladedauer

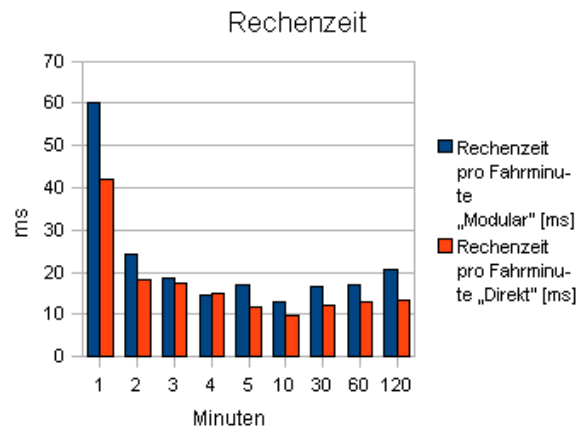


Abbildung 4.7: Ergebnisse des Performanztests zur Rechendauer

bis 10 Minuten werden ohne merklichen zeitlichen Unterschied berechnet.

Das Ergebnis ist insofern nicht überraschend, da sich beide Algorithmen in der gleichen Komplexitätsklasse befinden, die in beiden Fällen der des komplexesten Einzelproblems gleicht. Das komplexeste Einzelproblem bei den Berechnungen ist das Sortieren von Arrays für den Median. Für diese Operation wurde die Methode `sort()` der Klasse `java.util.Arrays` verwendet, die im Sun Java SDK enthalten ist. Diese Methode nutzt einen modifizierten Quicksort-Algorithmus [22], welcher ein Algorithmus der Komplexitätsklasse  $O(n \log(n))$  ist [4].

Die Lesemodule unterscheiden sich ferner in dem Speicherbedarf. Die Speicherung eines Datensatzes bedarf beim Modul *HumanReadable* ca. 180 Byte. Diese Zahl ist nicht konstant, da beispielsweise die Nullen am Ende einer Zahl nicht gespeichert werden. Annähernd konstant ist der durchschnittliche Speicherbedarf für einen Datensatz für das Lesemodul *SerializedArrayList*. Ein Datensatz benötigt ca. 95 Byte. Dieser Wert ist nicht vollkommen konstant, weil zu den Fahrdaten auch Informationen über die verwendete Karte sowie die Startzeit der Simulationsfahrt gespeichert werden, und diese in die Größe der resultierenden Gesamtdatei mit eingehen. Eine Fahrt von einer Minute führte in einem Test zu Dateigrößen von ca. 208 kB für das Modul *HumanReadable* und ca. 112 kB für das Modul *SerializedArrayList*.

## 5 Vorgehen einer Analyse

Die Flexibilität des Analyseprogramms ermöglicht die Analyse vieler unterschiedlicher Fahrdaten. Die Module für das Programm, die im Rahmen dieser Arbeit entstanden sind, wurden hauptsächlich für einen spezifischen Analysevorgang geplant und umgesetzt. Es sollte jedoch bedacht werden, dass mit der Änderung der Analysemodule und je nach eingesetztem Fahrsimulator auch andere Analysevorgänge möglich sind. Zur Orientierung wird hier das Vorgehen der in Kapitel 6.1 durchgeführten Studie erläutert.

Zunächst wird das Schreibmodul des Fahrsimulators gewählt. Die Auswahl besteht bisher aus den Modulen *human* und *arraylist*. Das Modul *human* entspricht dem Modul *HumanReadable* in dem Analyseprogramm, während das Modul *arraylist* dem Modul *SerializedArrayList* im Analyseprogramm entspricht. Die Auswahl geschieht durch die Konfigurationsdatei *simulator.conf* im Ordner *conf* des Fahrsimulators. Der zu ändernde Schlüssel heißt *analyzation\_data\_style\_writing*. Der Abschnitt in der Konfigurationsdatei ist in Abbildung 5.1 dargestellt. In diesem Beispiel wird das Schreibmodul *human*

```

19
20# analyzaion_data_style_writing:
21#   human: human readable but much data consumption (default)
22#   arraylist: not human readable, less data consumption
23analyzation_data_style_writing=human
24
25# map - map to load - defaults to resources/maps/example.txt
26map=resources/maps/example.txt
27
...

```

Abbildung 5.1: Konfiguration für das Schreibmodul im Fahrsimulator in der Datei *simulator.conf*

genutzt. Ferner ist es wichtig, unter dem Schlüssel *map* die zu nutzende Karte anzugeben.

In der Studie führte jede teilnehmende Person in Anlehnung an [19] vier Fahrten durch. Die erste, um sich an den Fahrsimulator zu gewöhnen, eine als Referenzfahrt ohne Nebenaufgaben, sowie zwei Fahrten mit jeweils einer Nebenaufgabe.

Die Studie nutzt den folgenden Aufbau, der in Abbildung 5.2 zu sehen ist. Die Steuerung erfolgt mit Hilfe eines Lenkrads für den PC mit entsprechenden Pedalen zum Bremsen und Beschleunigen. Ein Bildschirm für den PC mit 47" Bilddiagonale stellt die Fahrt dar, während ein kleinerer Bildschirm die Geschwindigkeit durch Simulation der üblichen Armaturen abbildet. Die Armaturen sind durch ein Flash-Programm realisiert, das die UDP-Daten empfängt und verarbeitet, welche der Fahrsimulator während der Fahrt in das lokale Netzwerk schickt. Der Fahrsimulator schickt Fahrdaten, wie z.B. die Geschwindigkeit und den Status der Bremse per UDP an einen Empfänger, der in der Konfigurationsdatei *simulator.conf* eingestellt werden kann.

Die Teilnehmer unterschreiben zunächst ein Dokument, in dem sie unter Anderem darüber aufgeklärt werden, dass sie jederzeit aufhören können, und in dem sie dem Durchführenden das Recht einräumen, die Ergebnisse der Studie anonymisiert zu verwenden und zu veröffentlichen (Dokument siehe Anhang). Die Teilnehmer werden über den Verlauf der Studie informiert, bevor sie, in Anlehnung an [19] vier Minuten lang mit dem Simulator fahren, um ein Gefühl für die Steuerung zu entwickeln. Dadurch wird ein Teil des Lerneffekts kompensiert, da gerade in den ersten Sekunden der Lerneffekt bei der Nutzung eines solchen Systems naturgemäß sehr stark ist. Dadurch, dass die Teilnehmer den Fahrsimulator zum ersten Mal verwenden, lernen sie im Laufe der Studie immer bes-



Abbildung 5.2: Aufbau der Studie

ser mit diesem umzugehen. Durch diesen Lerneffekt verbessert sich die Fahrleistung mit der Zeit. Diesem Effekt ist durch geeignete Maßnahmen entgegen zu wirken.

Nachdem der Teilnehmer die Testfahrt absolviert hat, werden die Fahrten durchgeführt, deren Daten gemessen werden. Jede Fahrt dauert in Anlehnung an [19] exakt vier Minuten. Um vergleichbare Daten zu erhalten ist es notwendig, dass die jeweils miteinander zu vergleichenden Fahrten entweder die gleiche Zeitdauer oder die gleiche Strecke aufweisen. So ist es ebenfalls möglich Fahrten zu simulieren, in deren Verlauf eine bestimmte Strecke zurückgelegt werden muss. Wird nicht eine Dauer, sondern eine Strecke vorgegeben, so kann die Ablenkung eines Fahrers unter Anderem an der Dauer der Fahrt gemessen werden, die durch eine geringere Geschwindigkeit entsteht, während bei einer Fahrt über einen bestimmten Zeitraum die Ablenkung unter anderem durch eine verminderte Geschwindigkeit identifiziert werden kann.

Um den Lerneffekt während der Fahrten zu kompensieren, wird die Reihenfolge, in der die Teilnehmer die Referenzfahrt sowie die Fahrten mit den Aufgaben durchführen, vermischt. So fährt Teilnehmer 1 z.B. zunächst die Referenzfahrt, danach die Fahrt mit Aufgabe 1 und als letztes die Fahrt mit Aufgabe 2, während der zweite Teilnehmer zunächst die Fahrt mit Aufgabe 2 fährt, dann die Referenzfahrt und zuletzt die Fahrt mit Aufgabe 1. Die Reihenfolge, in der die Teilnehmer in den Studien zur Evaluation des Analyseprogramms gefahren sind, ist in der jeweiligen Studienbeschreibung enthalten (siehe Anhang). Bei der Fahrt in dem Simulator wurde eine Geschwindigkeitsbegrenzung von 50 km/h durch Schilder am Straßenrand angegeben. Vor Kurven wurde eine Geschwindigkeitsbegrenzung von 30 km/h angegeben.

Die Aufgaben wurden so gewählt, dass die Teilnehmer über die gesamte Dauer der

## 5 VORGEHEN EINER ANALYSE

Fahrt mit der Aufgabe beschäftigt sind oder diese wiederholt durchführen. Bei Wiederholung ist darauf zu achten, dass die Aufgabe sich leicht ändert, um einen Lerneffekt zu vermeiden. Besteht die Aufgabe beispielsweise darin, den Weg zwischen zwei Städten auf einer Straßenkarte zu finden und der Teilnehmer hat einen Weg gefunden, so sollte die Aufgabe erneut mit zwei anderen Städten geschehen, bei denen sich der Weg signifikant unterscheidet. Soll ein Weg gefunden werden, der dem vorherigen Weg zu sehr ähnelt, kennt der Teilnehmer schon einen Großteil des zu findenden Wegs durch die vorherige Aufgabe. Zur genaueren Untersuchung kann bei Beginn der Bearbeitung jeder Aufgabe im Fahrsimulator ein Marker gesetzt werden. Dadurch können die Teilstrecken, in denen der Teilnehmer aktiv eine Aufgabe bearbeitet hat, identifiziert werden, und von den Teilstrecken, in denen während der Fahrt eine neue Aufgabe gestellt wurde, abgegrenzt werden. Im o.g. Beispiel bedeutet das, dass zwischen Teilstrecken, auf denen der Fahrer damit beschäftigt war einen Weg zu suchen und Teilstrecken, auf denen er neu zu suchende Städte angesagt bekam, abgegrenzt wird.

Vor jeder Fahrt wird der Teilnehmer über den Verlauf und ggf. die Aufgabe während der nächsten Fahrt informiert. Es wird ihm ferner jederzeit Gelegenheit geboten, Fragen zum Ablauf zu stellen, um Missverständnisse zu vermeiden. Vor und nach jeder Fahrt wird der Fahrsimulator mit der Reset-Taste *R* in den Startzustand versetzt, wodurch das Fahrzeug an die Startposition gestellt wird, nachdem die aktuelle Fahrdaten-Datei geschlossen wurde. Mit dem Reset des Fahrsimulators wird automatisch eine neue Fahrdaten-Datei erzeugt.

Nach den Simulationsfahrten befinden sich die Fahrdaten-Dateien in dem Verzeichnis *analyzationData* unterhalb des Verzeichnisses des Fahrsimulators. Die Dateien sind nach dem Zeitpunkt des Beginns der Simulationsfahrt benannt. Anhand der notierten Startzeiten können daher die Fahrdaten-Dateien identifiziert werden, die die zu bewertenden Daten enthalten. Abbildung 5.3 zeigt einen Ausschnitt einer Fahrdaten-Datei, die mit dem Schreibmodul *human* erzeugt wurde. Die erste Zeile speichert eine Referenz zur

```
1C:\Dokumente und Einstellungen\User\workspace\Fahrsimulator\resources\maps\example.txt
22008.05.29 18:52:17
3-1200.0999755859375:5.0:1630.0:1.827988624572754:1212079938262:0.0:0.0:false:0.0:2.0:false:false
4-1200.1082763671875:5.063386917114258:1630.0003662109375:3.640005588531494:1212079938312:0.0:0.0
5-1200.0987548828125:5.395421981811523:1630.00439453125:0.13989455997943878:1212079938362:0.0:0.0
6-1200.083984375:5.338343620300293:1630.0087890625:0.0650368258357048:1212079938412:0.0:0.0:false
7-1200.0806884765625:5.341089725494385:1630.0198974609375:0.11257505416870117:1212079938462:0.0:0
8-1200.080078125:5.338070392608643:1630.015869140625:0.06373222172260284:1212079938823:0.0:0.0:fa
9-1200.0797119140625:5.336135387420654:1630.0128173828125:0.09172749519348145:1212079938873:0.0:0
10-1200.0830078125:5.336671829223633:1630.0177001953125:0.15378399193286896:1212079938943:0.0:0
11-1200.083251953125:5.3346147537231445:1630.0147705078125:0.04047546908259392:1212079938993:0.0:0
12-1200.0838623046875:5.335788249969482:1630.0147705078125:0.05400888994336128:1212079939043:0.0:0
13-1200.083984375:5.336819648742676:1630.0146484375:0.024815060198307037:1212079939093:0.0:0:fa
```

Abbildung 5.3: Daten in einer Fahrdatendatei des Schreibmoduls *human*

Kartendatei, die bei der Simulationsfahrt verwendet wurde. In der zweiten Zeile wird der Zeitpunkt des Starts der Fahrt gespeichert. Die konkreten Fahrdaten werden, wie in der Tabelle 4.1 angegeben ab der dritten Zeile gespeichert.

Die Analyse der Daten erfolgt mit dem Analyseprogramm. Zum Vergleich zweier Fahrten werden die entsprechenden beiden Fahrdaten-Dateien in dem Analyseprogramm geladen. Auf der linken Seite befindet sich die grafische Darstellung der Karte und der Fahrten. Die Fahrt der ersten Fahrdaten-Datei ist blau, die Fahrt der zweiten Fahrdaten-Datei rot eingezeichnet. Bremspunkte sowie das Setzen von Markern werden mit entsprechenden Symbolen vermerkt. Werte, die in der Regel als bessere Fahrleistung interpre-



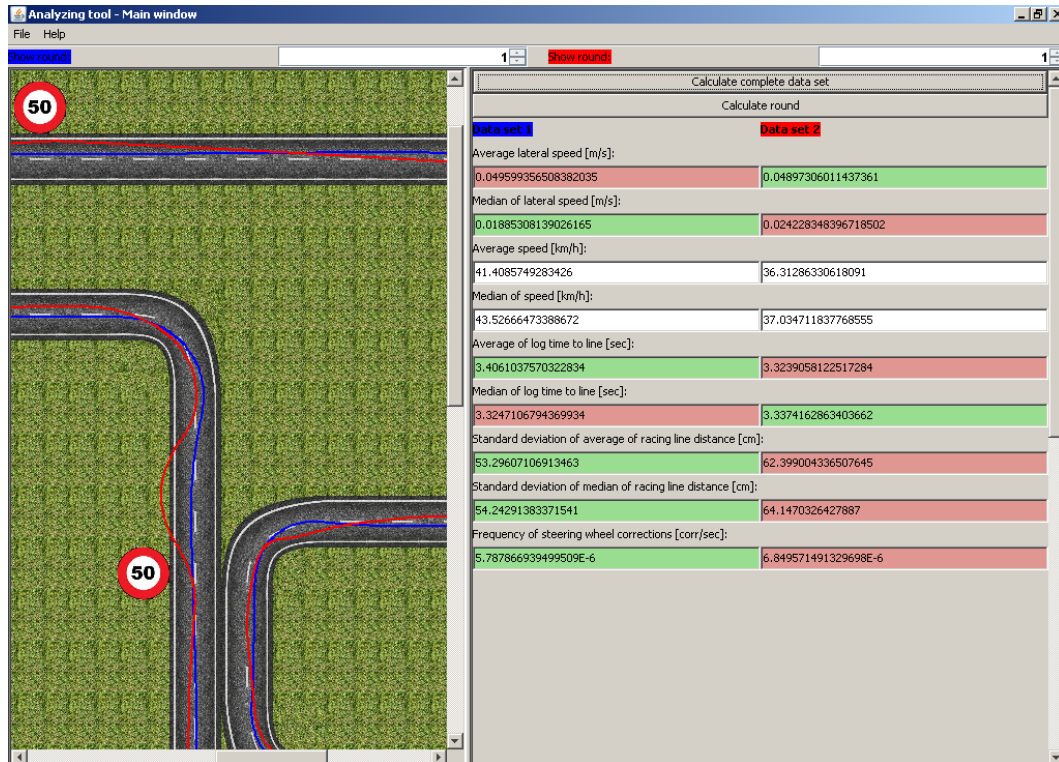


Abbildung 5.4: Bildschirmfoto der Analyse

tiert werden können, sind grün hinterlegt, während Fahrleistungen, die in der Regel als schlechter interpretiert werden rot hinterlegt sind. Die Bewertung, welche Ausprägung einer Fahreigenschaft besser ist, wurde in Kapitel 4.3 beschrieben. Eine bessere Fahrleistung im Vergleich der beiden geladenen Fahrdaten-Dateien ist beispielsweise eine geringere Standardabweichung der mittleren Distanz von der Ideallinie, da dies bedeutet, dass der Fahrer insgesamt ruhiger und damit berechenbarer in seiner Fahrspur gefahren ist, als der Fahrer der jeweils anderen Fahrdaten-Datei. Werte, die nicht als generell „besser“ oder „schlechter“ interpretiert werden können, bleiben weiß hinterlegt. Dies ist zum Beispiel die mittlere Geschwindigkeit. Wenn die Simulation das Fahren durch eine Fußgängerzone simuliert, ist eine geringere Geschwindigkeit besser, während bei einer Simulation einer Autobahnfahrt eine höhere Geschwindigkeit unter Umständen als besser interpretiert werden kann. Das Analyseprogramm bietet die Möglichkeit, alle in Kapitel 4.3 beschriebenen Werte der Fahrdaten zu ermitteln. In einer Simulation zur Messung der Fahrerablenkung sind insbesondere die mittlere Geschwindigkeit sowie die Standardabweichung der mittleren Distanz zur Ideallinie interessant. Dies wird in Kapitel 6 näher erläutert.

## 6 Evaluierung

Zur Evaluierung des Analyseprogramms wurden zwei Studien durchgeführt. Das Ziel der ersten Studie war, zu evaluieren, ob der Fahrsimulator die Ablenkung des Fahrers durch eine Nebenaufgabe erkennt. Dazu haben die Teilnehmer zwei ähnlich ablenkende Nebenaufgaben beim Fahren durchgeführt. Die Aufgaben der ersten Studie sind in Kapitel 6.1 beschrieben. Nachdem die Ablenkung in der ersten Studie erfolgreich erkannt werden konnte, wurde in der zweiten Studie untersucht, ob das Analyseprogramm auch die unterschiedlich starke Ablenkung zweier verschiedener Nebenaufgaben erkennen kann. Dazu wurden Simulationsfahrten durchgeführt, in denen die Teilnehmer als zweite Nebenaufgabe eine offensichtlich aufwändigere Tätigkeit durchführen mussten. Die Nebenaufgaben der zweiten Studie sind in Kapitel 6.2 beschrieben. Die abhängigen Variablen der Studien waren:

1. die durchschnittliche Geschwindigkeit  $\bar{v}$
2. die Standardabweichung der durchschnittlichen Distanz zur Ideallinie  $\bar{\sigma}$
3. der Quotient  $q = \frac{\bar{\sigma}}{\bar{v}}$

Diese Werte sollen die Ablenkung des Fahrers angeben. Die übergeordneten Hypothesen der Studien waren die Folgenden:

1. Die Teilnehmer fahren durch die Ablenkung langsamer.
2. Die Teilnehmer fahren durch die Ablenkung weniger spursicher

Die Spursicherheit wird dadurch messbar, dass die Standardabweichung der durchschnittlichen Distanz zur Ideallinie steigt. Das Analyseprogramm ist in der Lage weitere Faktoren der Fahrdaten zu berechnen, wie in Kapitel 4.3 beschrieben. Diese scheinen sich aus den folgenden Gründen für diese Studie nicht zu eignen. Der Einsatz dieser Werte in zukünftigen Studien bedarf weiterer Untersuchungen.

1. Mittlere seitliche Geschwindigkeit und mittlere Zeit bis zur Spurbegrenzung  
Die mittlere seitliche Geschwindigkeit wurde nicht betrachtet, da bei den Studien zu beobachten war, dass die Fahrer durch die Ablenkung keine größeren Lenkbewegungen ausführen. Erst größere Lenkbewegungen würden zu einer erhöhten seitlichen Geschwindigkeit führen. Die mittlere Zeit bis zur Spurbegrenzung wurde mit der gleichen Begründung nicht betrachtet, weil sie ebenfalls die seitliche Geschwindigkeit misst, wobei sie das Ergebnis mit einem zeitlichen Wert belegt, anstatt mit einer Geschwindigkeit.
2. Standardabweichung der Geschwindigkeit, mittlere absolute Beschleunigung, Standardabweichung der Gaspedalstellung und die Frequenz der Geschwindigkeitskorrektur  
Diese Werte waren für diese Studien nicht relevant. Die Teilnehmer fuhren auf einer Strecke mit festen Geschwindigkeitsbegrenzungen und es wurde keine Geschwindigkeit vorgegeben. Ebenso wurde den Fahrern kein Ziel gesetzt, welches ein möglichst rasches Durchfahren des Kurses motivierte. Die Geschwindigkeit und der Grad der Beschleunigung wurde den Fahrern frei gestellt.

3. Die Standardabweichung der Lenkerstellung und die Frequenz der Steuerkorrektur Diese Werte wurden vernachlässigt, da die zu befahrene Straße Kurven enthielt. Aufgrund der Kurven sind die o.g. Werte nicht aussagekräftig, weil die in den Kurven notwendigen Änderungen der Lenkradstellung in die Messungen mit eingehen.

Die Studien verwenden als unabhängige Variablen die gesammelten Fahrdaten von unterschiedlichen Fahrten. Diese sind in dem jeweiligen Unterkapitel erklärt.

## 6.1 Studie zur Identifikation von Ablenkung

### Versuchsaufbau

Das Ziel der ersten Studie war, zu zeigen, ob das Analyseprogramm die Ablenkung eines Fahrers erkennen kann. Dazu wurden 10 Testpersonen eingeladen. 9 von 10 Teilnehmern waren männlich. Eine Frau nahm an der Studie teil. Das Durchschnittsalter war 25,1 Jahre. Die Teilnehmer hatten seit durchschnittlich 7,3 Jahren ihren Führerschein. 4 Teilnehmer fuhren Durchschnittlich 0 - 10000 km im Jahr, 3 fuhren 10000 - 20000 km im Jahr, 2 fuhren 20000 - 30000 km im Jahr und ein Teilnehmer fuhr mehr als 30000 km im Jahr.

Der Aufbau und die abhängigen Variablen der Studie sind in Kapitel 5 beschrieben. Die unabhängigen Variablen der Studie waren die gesammelten Daten unterschiedlicher Fahrten:

1. eine Referenzfahrt ohne Nebenaufgabe (R)
2. eine Fahrt mit Nebenaufgabe 1 (A1)
3. eine Fahrt mit Nebenaufgabe 2 (A2)

Die Nebenaufgabe 1 bestand daraus, aus einer Tüte mit bunten Bonbons gelbe Bonbons zu holen und in eine Schale zu legen. Schale und Tüte lagen auf dem Beifahrersitz. Diese Nebenaufgabe ist angelehnt an eine Nebenaufgabe in [19], in der Bonbons ausgewickelt und in einen Aschenbecher gelegt werden sollen, und sollte eine leichte Ablenkung des Fahrers während der Fahrt simulieren. Abbildung 6.1 zeigt einen Teilnehmer, der bei der Bearbeitung der Nebenaufgabe 1 beschäftigt ist. Die zweite Nebenaufgabe war, wie in [19], in einem Portemonnaie, das auf dem Beifahrersitz liegt, bestimmte Münzen zu finden. Die Teilnehmer wurden beispielsweise angewiesen, ein Fünfcentstück sowie ein Eurostück aus dem Portemonnaie zu holen. Wenn die Teilnehmer die Münzen gefunden hatten, wurden diese zurück in das Portemonnaie gelegt, bevor die Teilnehmer dazu aufgefordert wurden, die nächsten Münzen zu suchen. Der Schwierigkeitsgrad beider Aufgaben war ähnlich. Beide bedeuteten für den Fahrer eine geringe Ablenkung, weil er auf äußere Reize reagieren musste, welche eine kurzzeitige Abwendung des Blicks von der Straße erforderlich machten. Diese Reize waren die gelbe Farbe der Bonbons in der ersten Aufgabe und das Gefühl der Münzen zwischen den Fingern in der zweiten Aufgabe, welches die Teilnehmer täglich fühlen und so die Münzen ohne viel Aufwand unterscheiden können. In [19] werden ähnliche Nebenaufgaben verwendet, die dort als sehr unterschiedlich Ablenkend erkannt werden. Die erste Nebenaufgabe der hier erläuterten Studie ist jedoch abgewandelt, so dass sie offensichtlich ablenkender ist als die Aufgabe in [19], da die Teilnehmer ihren Blick von der Fahrbahn abwenden müssen.





Abbildung 6.1: Ein Teilnehmer bei der Bearbeitung der Nebenaufgabe 1 in der ersten Studie

### Durchführung

Zunächst wurden die Teilnehmer ebenso wie die Teilnehmer der ersten Studie über den Ablauf der Studie informiert, sowie darüber, dass sie jederzeit die Studie abbrechen können, vor allem wenn sie Übelkeit verspüren sollten. Symptome der sogenannten „Simulator Sickness“ sind ein bekanntes Problem bei Simulationen (siehe z.B. [15]).

Die Reihenfolge der Durchführung der Fahrten wurde vermischt, um den Lerneffekt bei den Teilnehmern in der Bedienung des Fahrsimulators auszugleichen. Ferner haben alle Teilnehmer zunächst eine Fahrt ohne Nebenaufgabe durchgeführt, in der sie sich an den Fahrsimulator gewöhnen konnten.

### Hypothese

Die Hypothesen dieser Studie waren:

1. Die Teilnehmer fahren durch die Ablenkung langsamer.
2. Die Teilnehmer fahren durch die Ablenkung weniger spursicher.
3.  $q = \frac{\bar{\sigma}}{\bar{v}}$  (siehe Kapitel 4.3.14) steigt signifikant zwischen den Referenzfahrten und den Fahrten mit Nebenaufgabe.

### Ergebnisse

In den Abbildungen 6.2 und 6.3 sind die abhängigen Variablen  $\bar{\sigma}$  und  $\bar{v}$  grafisch dargestellt. Es ist zu erkennen, dass die mittlere Geschwindigkeit in den Referenzfahrten tendenziell höher war als bei der Durchführung der Nebenaufgaben ( $p=0,0056$  bzw.

Durchschnittliche Geschwindigkeit

Teilnehmer	R	A1	A2
1	35.3386	37.4286	37.1210
2	34.4190	30.8151	30.4092
3	50.4668	46.8729	43.9229
4	41.4086	36.3129	34.4996
5	37.5806	27.9121	31.4195
6	39.2913	32.2653	31.1368
7	35.7126	33.3396	32.0721
8	48.3109	45.4751	39.7079
9	45.3357	44.7121	36.2342
10	52.3461	43.4085	42.9811
<b>Mittelwert</b>	<b>42.0210</b>	<b>37.8542</b>	<b>35.9504</b>

Tabelle 6.1: Durchschnittliche Geschwindigkeit in der ersten Studie

$p=0,0003$ ). Bei der Studie konnte beobachtet werden, dass die Teilnehmer die Geschwindigkeit drosselten, während sie durch die Beschäftigung mit den Aufgaben abgelenkt waren. Die Standardabweichung der mittleren Distanz zur Ideallinie stieg durch die Bearbeitung der Aufgaben nicht signifikant an ( $p=0,21$  bzw.  $p=0,07$ ). Durch die Verringerung der Geschwindigkeit war es den Fahrern möglich, weiterhin relativ sicher in der Spur zu fahren. Diese Kombination aus geringerer Geschwindigkeit und nur wenig verschlechterter Standardabweichung der mittleren Distanz zur Ideallinie führt dazu, die beiden Werte in Relation zueinander zu betrachten, durch den Wert  $q$ . Die Betrachtung des Quotienten  $q$  zeigt, dass sich die Fahrleistung bei der Bearbeitung der Nebenaufgaben signifikant änderte ( $p=0,0046$  bzw.  $p=0,0029$ ). Abbildung 6.4 zeigt den Quotienten  $q$  für jeden Teilnehmer. Die gesamten Werte von  $q$  sind in Tabelle 6.3 zu finden.

Die Signifikanz der Unterschiede konnte durch den sogenannten T-Test nach Student nachgewiesen werden. Der T-Test nach Student ist eine verbreite Möglichkeit, die Wahrscheinlichkeit zu beziffern, dass zwei Stichproben aus derselben Gesamtmenge genommen wurden. In der Regel werden die Stichproben als signifikant unterschiedlich angenommen, wenn der T-Test nach Student eine Wahrscheinlichkeit von bis zu fünf Prozent ergibt [9]. Die Ergebnisse der T-Tests sind in der Tabelle 6.4 zu finden. Der T-Test ergibt, dass die Fahrleistungen, repräsentiert durch  $q$ , signifikante Unterschiede aufweisen. Dies bedeutet, dass ein signifikanter Unterschied zwischen den Leistungen bei den Referenzfahrten R und den Fahrten A1 (1,8 und 2,09, T-Test:  $p=0,03$ ) sowie den Referenzfahrten R und den Fahrten A2 bestehen (1,8 und 2,44, T-Test:  $p=0,01$ ). Kein signifikanter Unterschied konnte zwischen den Fahrleistungen der Fahrten A1 und A2 identifiziert werden (2,09 und 2,44, T-Test:  $p=0,07$ ). Dies bestätigt die These, dass die Nebenaufgaben einen ähnlichen Schwierigkeitsgrad aufweisen. Die Teilnehmer konnten bei der Bearbeitung der Aufgaben nur noch deshalb ähnlich sicher in der Spur fahren wie in der Referenzfahrt, weil sie ihre Geschwindigkeit signifikant verringert haben. Somit wird angenommen, dass  $q$ , der die Spurhaltung und die Geschwindigkeit in Relation setzt, den aussagekräftigsten Wert liefert. Die Tabellen 6.1 und 6.1 zeigen die gemessenen Werte der durchschnittlichen Geschwindigkeit und der Standardabweichung der durchschnittlichen Distanz zur Ideallinie.

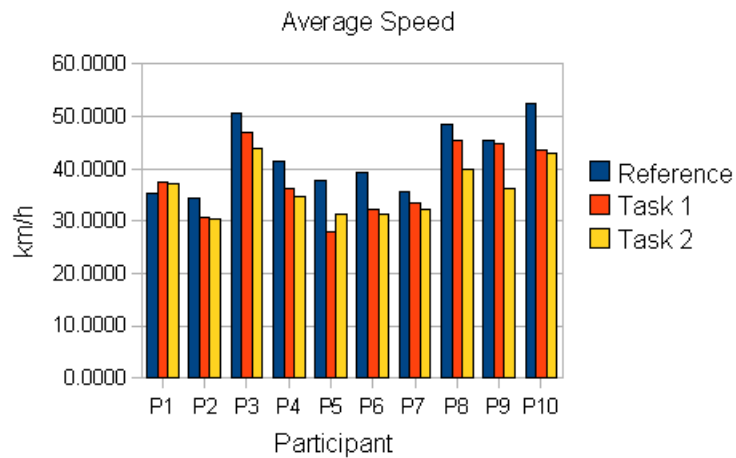


Abbildung 6.2: Durchschnittliche Geschwindigkeit in der ersten Studie

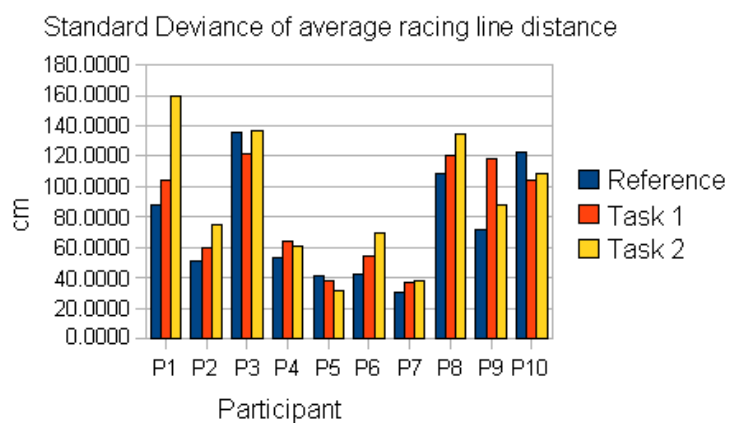


Abbildung 6.3: Standardabweichung von der Distanz zur Ideallinie in der ersten Studie

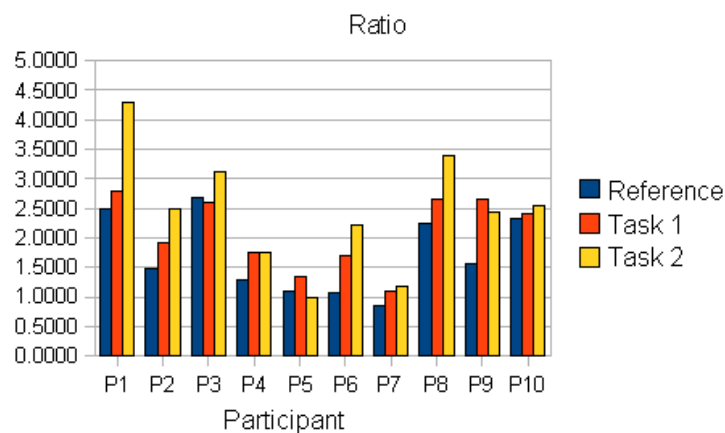


Abbildung 6.4: Quotient in der ersten Studie

Teilnehmer	R	A1	A2
1	87.6259	104.2246	159.6709
2	50.8861	59.3927	75.3738
3	135.6339	121.3885	136.5501
4	53.2961	63.7430	60.6242
5	41.1266	37.6174	31.1862
6	42.0545	54.6124	69.3161
7	30.3910	36.3807	38.0511
8	108.7686	120.5219	134.4695
9	71.3364	118.5404	88.1995
10	122.1968	104.5940	108.9961
<b>Mittelwert</b>	<b>74.3316</b>	<b>82.1016</b>	<b>90.2437</b>

Tabelle 6.2: Standardabweichung der mittleren Distanz von der Ideallinie in der ersten Studie

Teilnehmer	R	A1	A2
1	2.4796	2.7846	4.3014
2	1.4784	1.9274	2.4787
3	2.6876	2.5897	3.1089
4	1.2871	1.7554	1.7572
5	1.0944	1.3477	0.9926
6	1.0703	1.6926	2.2262
7	0.8510	1.0912	1.1864
8	2.2514	2.6503	3.3865
9	1.5735	2.6512	2.4342
10	2.3344	2.4095	2.5359
<b>Mittelwert</b>	<b>1.7108</b>	<b>2.0900</b>	<b>2.4408</b>

Tabelle 6.3: Quotient in der ersten Studie

	R - A1	R - A2	A1 - A2
T-Test ( $\bar{v}$ )	0.0056	0.0003	0.0991
T-Test ( $\bar{\sigma}$ )	0.2070	0.0687	0.2691
T-Test ( $q$ )	0.0046	0.0029	0.0721

Tabelle 6.4: Ergebnisse der ersten Studie

## 6.2 Studie zur Bewertung von Ablenkung

### Versuchsaufbau

In der zweiten Studie sollte gezeigt werden, ob das Analyseprogramm die unterschiedliche Intensität der Ablenkung zweier Nebenaufgaben beim Fahren messen kann. In dieser Studie wurden ebenfalls 10 Teilnehmer eingeladen. Einer der Teilnehmer musste den Versuch wegen Übelkeit abbrechen. Daher stehen nur die Werte von 9 Teilnehmern zur Verfügung. An dieser Studie nahmen 3 Frauen und 7 Männer teil. Das Durchschnittsalter war 26,7. Die Teilnehmer hatten seit durchschnittlich 8,8 Jahren einen Führerschein und sind im Durchschnitt weniger Auto gefahren, als in der ersten Studie. So gaben 7 von 9 Teilnehmern an, 0 - 10000 km im Jahr zu fahren. 2 Teilnehmer haben angegeben, 10000 - 20000 km im Jahr zu fahren.

Der Aufbau der Studie gleicht dem der ersten. Es wurden die gleichen abhängigen Variablen gemessen, die auch in der ersten Studie gemessen wurden. Die unabhängigen Variablen wurden jedoch verändert. Es wurden die folgenden Fahrten als unabhängige Variablen durchgeführt:

1. eine Referenzfahrt ohne Nebenaufgabe (R1)
2. eine Fahrt mit Nebenaufgabe 1 (A1)
3. eine Fahrt mit Nebenaufgabe 2 (A2)
4. eine weitere Referenzfahrt ohne Nebenaufgabe (R2)

Der Durchschnitt der Werte der Fahrten R1 und R2 wurde als Referenzfahrt R definiert. Dieses Vorgehen sollte den Lerneffekt während der Studie ausgleichen. In der ersten Studie wurde hier ein anderes Verfahren angewendet. Die Ergebnisse unterscheiden sich jedoch nicht signifikant ( $p(\bar{v}) = 0,6954$ ,  $p(\bar{\sigma}) = 0,2599$ ,  $p(q) = 0.2506$ ). Hierbei ist zu beachten, dass die Daten der zweiten Studie um ihren jeweiligen Durchschnitt ergänzt wurden, da der T-Test zwei gleichgroße Mengen zum Vergleich erwartet.

Die Nebenaufgabe 1 entsprach der Nebenaufgabe 1 der ersten Studie in Kapitel 6.1 und sollte eine leichte Ablenkung darstellen. Für die Nebenaufgabe 2 bekamen die Teilnehmer eine Straßenkarte. Zunächst wurden die Teilnehmer mit der Bedienung dieser spezifischen Straßenkarte, einem „ADAC TaschenAtlas Nordrhein-Westfalen“ des Jahres 2004, vertraut gemacht. Während der Fahrt wurden den Teilnehmern Fragen gestellt, die sie mit Hilfe der Straßenkarte beantworten sollten. Es wurden Fragen aus vier verschiedenen Kategorien gestellt:

1. Welche Stadt befindet sich näher an der Stadt X: Stadt A oder Stadt B?
2. Welche Autobahn-Ausfahrtnummer führt direkt zur Stadt X?
3. An welcher Autobahn liegt die Stadt X?
4. Zwischen welchen zwei Autobahnen liegt die Stadt X?

Die konkreten Fragen sind in der Studienbeschreibung zu finden (siehe Anhang). Um einen Lerneffekt zu vermeiden, wurden den Teilnehmern die Fragen in unterschiedlicher Reihenfolge gestellt. Während die Teilnehmer für die Bearbeitung der ersten Nebenaufgabe den Blick nur kurzfristig von der Fahrbahn nehmen mussten, war es bei der zweiten

Nebenaufgabe nötig sich während der Fahrt intensiv mit der Straßenkarte zu beschäftigen. Es wurde festgestellt, dass die Teilnehmer durch die komplizierte Handhabung der Straßenkarte und das Suchen der Städte ihre Aufmerksamkeit deutlich von der Fahrt auf die Nebenaufgabe verlagerten. Die Teilnehmer haben durchschnittlich 3,6 Fragen der Nebenaufgabe 2 beantwortet.

### Durchführung

Die Teilnehmer fuhren zunächst eine Fahrt von 4 Minuten zur Gewöhnung an den Fahr Simulator. Danach fuhren sie abwechselnd in der Reihenfolge R1, A1, A2, R2 und R1, A2, A1, R2 um den Lerneffekt zu kompensieren.

### Hypothese

Die Hypothesen der Studie waren:

1. Die Teilnehmer fahren durch die Ablenkung langsamer, in A2 noch langsamer als in A1.
2. Die Teilnehmer fahren durch die Ablenkung weniger spursicher, in A2 noch weniger spursicher als in A1.
3.  $q = \frac{\bar{\sigma}}{\bar{v}}$  (siehe Kapitel 4.3.14) steigt signifikant zwischen den Referenzfahrten und den Fahrten mit Nebenaufgabe sowie zwischen A1 und A2.

### Ergebnisse

Die Abbildungen 6.5 und 6.6 stellen die gemessenen Werte zur durchschnittlichen Geschwindigkeit bzw. der Standardabweichung der durchschnittlichen Distanz zur Ideallinie grafisch dar. Der Quotient zwischen diesen Werten ist in Abbildung 6.7 dargestellt. Es ist zu erkennen, dass die Teilnehmer bei der Bearbeitung der Aufgaben langsamer gefahren sind als in den Referenzfahrten ( $p=0,01$  bzw.  $p=0,0001$ ). Trotzdem fuhren die Teilnehmer bei der Bearbeitung der Aufgaben tendenziell weniger sicher in der Spur als bei den Referenzfahrten ( $p=0,04$  bzw.  $p=0,016$ ). Der T-Test in Tabelle 6.8 zeigt, dass diese Unterschiede signifikant sind. Während die Teilnehmer bei der Bearbeitung der Nebenaufgabe 2 nicht signifikant langsamer gefahren sind als bei der Bearbeitung der Nebenaufgabe 1 ( $p=0,41$ ), sind sie bedeutend schlechter in ihrer Spur gefahren ( $p=0,038$ ). Diese Ergebnisse resultieren darin, dass  $q$  einen signifikanten Unterschied zwischen den Fahrten R und A1 (1,43 und 2,21, T-Test:  $p=0,0032$ ) sowie den Fahrten R und A2 (1,43 und 2,81, T-Test:  $p=0,0052$ ) aufweist. Ferner ist  $q$  zwischen A1 und A2 signifikant höher (2,21 und 2,81, T-Test:  $p=0,0436$ ). Die Tabellen 6.5 und 6.6 zeigen die gemessenen Werte der durchschnittlichen Geschwindigkeit und der Standardabweichung der durchschnittlichen Distanz zur Ideallinie.

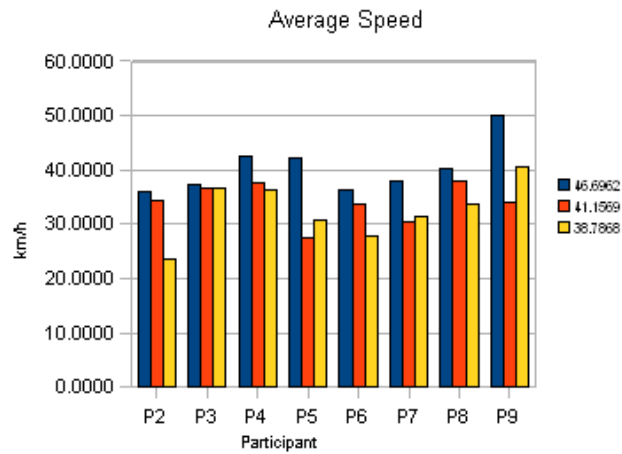


Abbildung 6.5: Durchschnittliche Geschwindigkeit in der zweiten Studie

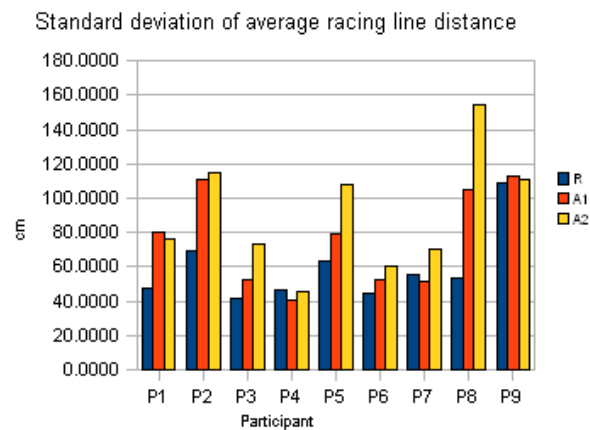


Abbildung 6.6: Standardabweichung von der Distanz zur Ideallinie in der zweiten Studie

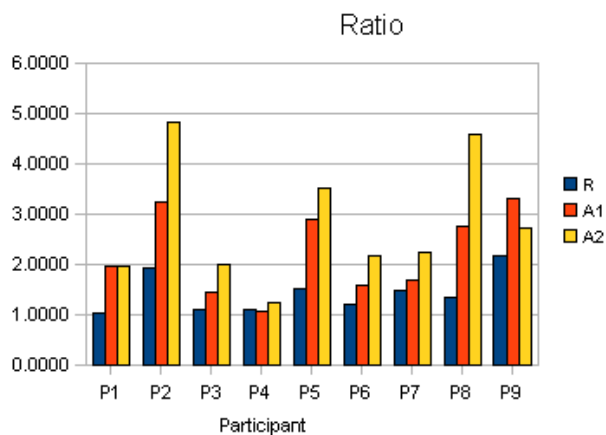


Abbildung 6.7: Quotient in der zweiten Studie

Teilnehmer	R	A1	A2
1	46.6962	41.1569	38.7868
2	35.9119	34.3496	23.7174
3	37.2383	36.5630	36.6029
4	42.3854	37.4867	36.2210
5	42.1801	27.4370	30.7638
6	36.3650	33.7529	27.9444
7	37.8660	30.3543	31.3394
8	40.1897	37.8274	33.6842
9	50.1842	33.9151	40.5355
<b>Mittelwert</b>	<b>41.0018</b>	<b>34.7603</b>	<b>33.2884</b>

Tabelle 6.5: Durchschnittliche Geschwindigkeit in der zweiten Studie

Teilnehmer	R	A1	A2
1	47.4330	80.5609	75.7969
2	69.1642	110.8805	115.0758
3	41.1025	52.5534	73.2671
4	46.7810	40.1320	45.1333
5	63.6073	79.2129	108.3417
6	44.2046	52.7845	60.2226
7	55.6265	51.3378	70.1966
8	53.1200	104.4763	154.3137
9	108.6007	112.4742	110.8729
<b>Mittelwert</b>	<b>58.8489</b>	<b>76.0458</b>	<b>90.3578</b>

Tabelle 6.6: Standardabweichung der mittleren Distanz von der Ideallinie in der zweiten Studie

Teilnehmer	R	A1	A2
1	1.0158	1.9574	1.9542
2	1.9259	3.2280	4.8520
3	1.1038	1.4373	2.0017
4	1.1037	1.0706	1.2461
5	1.5080	2.8871	3.5217
6	1.2156	1.5639	2.1551
7	1.4690	1.6913	2.2399
8	1.3217	2.7619	4.5812
9	2.1640	3.3163	2.7352
<b>Mittelwert</b>	<b>1.4253</b>	<b>2.2126</b>	<b>2.8097</b>

Tabelle 6.7: Quotient in der zweiten Studie

	R - A1	R - A2	A1 - A2
T-Test ( $\bar{v}$ )	0.0109	0.0001	0.4112
T-Test ( $\bar{\sigma}$ )	0.0355	0.0160	0.0381
T-Test ( $q$ )	0.0032	0.0052	0.0436

Tabelle 6.8: Ergebnisse der zweiten Studie



### 6.3 Zusammenfassung

In den Kapiteln 6.1 und 6.2 wurden zwei Studien besprochen, die messen sollten, ob das entwickelte Analyseprogramm die Ablenkung der Teilnehmer durch Nebenaufgaben beim Fahren in dem Fahrsimulator identifizieren kann.

In der ersten Studie wurden zwei Fahrten mit Nebenaufgaben mit einer Referenzfahrt verglichen. Das Analyseprogramm war in der Lage, mit Hilfe des Quotienten aus der Standardabweichung der mittleren Distanz von der Ideallinie und der mittleren Geschwindigkeit, die Ablenkung der Teilnehmer zu identifizieren. Beide Nebenaufgaben waren offensichtlich ähnlich ablenkend. Die Ablenkung durch eine Nebenaufgabe konnte ebenso gezeigt werden, wie die Tatsache, dass beide Nebenaufgaben ähnlich ablenkend waren.

In der zweiten Studie wurde der Durchschnitt zweier Referenzfahrten, welche am Anfang und am Ende der jeweiligen Sitzung gefahren wurden, mit der Fahrt mit zwei offensichtlich unterschiedlich ablenkenden Nebenaufgaben verglichen. Die unterschiedlichen Grade der Ablenkung konnten ebenfalls durch den Quotienten gezeigt werden.

## 7 Fazit

Das Ziel dieser Arbeit war es, Fahrdaten automatisiert zu analysieren, um die Ablenkung eines Fahrers messen zu können. Grundlage der Messungen ist ein Fahrsimulator, der eine Fahrt in einer dreidimensionalen Umgebung ermöglicht und kostengünstig einzusetzen ist, da er an Ausrüstung nur einen PC benötigt. Der Fahrsimulator wurde angepasst, so dass er die für eine Analyse der Fahrt notwendigen Fahrdaten für ein Analyseprogramm zur Verfügung stellt. Es wurde ein Analyseprogramm entwickelt, um die Leistung von Fahrern in einem Fahrsimulator bewerten und vergleichen zu können. Dabei wurde darauf geachtet, dass die Software ihrem breiten Einsatzfeld gerecht werden kann, indem sie sehr flexibel entworfen wurde. So können neue Analysemodule erstellt und in das Programm eingefügt, sowie neue Lesemodule erstellt werden, um Daten aus verschiedensten Formaten und Quellen verarbeiten zu können. Der Preis für funktionale Flexibilität ist häufig eine langsamere Verarbeitungsgeschwindigkeit, da eine Lösung nicht vollkommen auf ihre Aufgaben zugeschnitten werden kann, insbesondere wenn die zukünftigen Aufgaben nicht vollständig bekannt sind. Es konnte jedoch gezeigt werden, dass das Analyseprogramm auch bei der Verarbeitung größerer Datenmengen noch eine angemessene Geschwindigkeit aufweist.

In zwei Studien konnte gezeigt werden, dass das Analyseprogramm erfolgreich in einem Szenario eingesetzt werden kann, in dem die Fahrleistung in einem Fahrsimulator gemessen werden soll. Die Ablenkung eines Fahrers, welche durch eine Nebenaufgabe verursacht wird, kann vom Analyseprogramm erkannt und angezeigt werden. Auch Unterschiede in der Ausprägung von Ablenkung kann von dem Analyseprogramm identifiziert werden, wie an dem Ergebnis der zweiten Studie zu erkennen ist. In den Studien konnten ausschließlich die Standardabweichung der mittleren Distanz zur Ideallinie, die mittlere Geschwindigkeit und der Quotient aus diesen beiden Werten als aussagekräftige Werte zur Messung der Ablenkung eines Fahrers identifiziert werden. Die weiteren, von dem Analyseprogramm zu berechnenden Werte, bedürfen noch weiteren Untersuchungen um konkrete Einsatzgebiete zu finden. Hier bietet es sich an, in weiteren Studien die Möglichkeiten des Analyseprogramms zu nutzen.



# Abbildungsverzeichnis

2.1	Systemarchitektur des Fahrsimulators . . . . .	3
2.2	Klassendiagramm mit Fokus auf das Fahrzeug . . . . .	4
2.3	Klassendiagramm mit Fokus auf die Karte . . . . .	5
2.4	Klassendiagramm der ActionHandler und InputHandler . . . . .	6
3.1	Der Aufbau des UMTRI-Fahrsimulators [10] . . . . .	8
3.2	Ein Teilnehmer am PatrolSim Fahrsimulator [5] . . . . .	9
3.3	Der „DriS“ Fahrsimulator [17] . . . . .	11
3.4	Sicht auf eine modellierte Straße im Fahrsimulator „DriS“ [17] . . . . .	12
3.5	Mittlere Spurposition und Standardabweichung der mittleren Spurposition in [12] . . . . .	14
3.6	Szene aus dem LCT [19] . . . . .	14
3.7	Normative und gemessene Fahrlinie im LCT . . . . .	15
3.8	Ergebnisse der Evaluationsstudie des LCT [19] . . . . .	16
4.1	Zweidimensionale Darstellung der drei Dimensionen im Fahrsimulator . . . . .	19
4.2	Distanz zur Ideallinie und deren Abweichung . . . . .	23
4.3	Modifizierte Systemarchitektur des Fahrsimulators . . . . .	27
4.4	Klassendiagramm des Analyseprogramms . . . . .	28
4.5	Bildschirmfoto der Programmoberfläche . . . . .	29
4.6	Ergebnisse des Performanztests zur Ladedauer . . . . .	36
4.7	Ergebnisse des Performanztests zur Rechendauer . . . . .	36
5.1	Konfiguration für das Schreibmodul im Fahrsimulator in der Datei <i>simulator.conf</i> . . . . .	37
5.2	Aufbau der Studie . . . . .	38
5.3	Daten in einer Fahrdatendatei des Schreibmoduls <i>human</i> . . . . .	39
5.4	Bildschirmfoto der Analyse . . . . .	40
6.1	Ein Teilnehmer bei der Bearbeitung der Nebenaufgabe 1 in der ersten Studie . . . . .	43
6.2	Durchschnittliche Geschwindigkeit in der ersten Studie . . . . .	45
6.3	Standardabweichung von der Distanz zur Ideallinie in der ersten Studie . . . . .	45
6.4	Quotient in der ersten Studie . . . . .	45
6.5	Durchschnittliche Geschwindigkeit in der zweiten Studie . . . . .	49
6.6	Standardabweichung von der Distanz zur Ideallinie in der zweiten Studie . . . . .	49
6.7	Quotient in der zweiten Studie . . . . .	49

## Tabellenverzeichnis

4.1	Tabelle der Datensätze des Fahrsimulators . . . . .	20
4.2	Ergebnisse des Performanztests . . . . .	35
6.1	Durchschnittliche Geschwindigkeit in der ersten Studie . . . . .	44
6.2	Standardabweichung der mittleren Distanz von der Ideallinie in der ersten Studie . . . . .	46
6.3	Quotient in der ersten Studie . . . . .	46
6.4	Ergebnisse der ersten Studie . . . . .	46
6.5	Durchschnittliche Geschwindigkeit in der zweiten Studie . . . . .	50
6.6	Standardabweichung der mittleren Distanz von der Ideallinie in der zweiten Studie . . . . .	50
6.7	Quotient in der zweiten Studie . . . . .	50
6.8	Ergebnisse der zweiten Studie . . . . .	50

## Literatur

- [1] *About - jmonkeyengine.com*. Website. <[http://www.jmonkeyengine.com/wiki/doku.php?id=about\\_jme](http://www.jmonkeyengine.com/wiki/doku.php?id=about_jme)>, besucht am 9. Juni 2008.
- [2] *lwjgl.org - Home of the Leightwheigt Java Game Library*. <<http://www.lwjgl.org/index.php>>, besucht am 9. Juni 2008.
- [3] *OpenGL Overview*. <<http://www.opengl.org/about/overview/>>, besucht am 9. Juni 2008.
- [4] C, ULLENBOOM: *Java ist auch eine Insel*, 7. Auflage. Galileo Computing, November 2007.
- [5] CROUCH D, STRAYER D, DREWS F: *A Comparison of the Cell Phone Driver and the Drunk Driver*. Human Factors, Vol. 48, No. 2:381 – 391, 2006.
- [6] E, BLANA: *Driving simulator validation studies: A literature review*, Dezember 1996.
- [7] E, BLANA: *A survey of driving research simulators around the world*, Dezember 1996.
- [8] GONZÁLEZ, IVÁN E., JACOB O. WOBBOCK, DUEN HORNG CHAU, ANDREW FAULRING BRAD A. MYERS: *Eyes on the road, hands on the wheel: thumb-based interaction techniques for input on steering wheels. GI '07: Proceedings of Graphics Interface 2007*, 95–102, New York, NY, USA, 2007. ACM.
- [9] GOSSET, W. S. (STUDENT): *On the probable error of a mean*. Biometrika 6, 1908.
- [10] GREEN P, LIN B, BAGIAN T: *Driver Workload as a Function of Road Geometry: A Pilot Experiment*. Technical Report UMTRI-93-39, 55ff, Dezember 1993.
- [11] GREEN P, REED M: *Validation of a Low-Cost Driving Simulator Using a Telephone Dialing Task*. Technical Report UMTRI-95-19, Juni 1995.
- [12] GREEN P, REED M: *Comparison of driving performance on-road and in a low-cost simulator using a concurrent telephone dialling task*. Ergonomics, Vol. 42:8:1015 – 1037, August 1999. URL: <<http://dx.doi.org/10.1080/001401399185117>>.
- [13] HEIMSTRA N, LARRABEE L, MAST T: *Effects of fatigue on basic processes involved in human operator performance: simple vigilance and target detection*. Highway Research Board, No 55, Pp17-20, 2FIG, 6 REF, 1964.
- [14] KLEE H, GUO D, RADWAN E YAN X: *Analyses of traffic parameters related to left turn gap acceptance using ucf driving simulator*. DSC North America 2003 Proceedings, Dearborn, Michigan, ISSN1546-5071, Oktober 2003.

- [15] KOLASINSKI, EM: *Simulator sickness in virtual environments*. US Army Research Institute Technical Report 1027, May 1995.
- [16] LEITAO, J. MIGUEL, A. COELHO F. N. FERREIRA: *DriS - A Virtual Driving Simulator*. Proceedings of the Second International Seminar on Human Factors in Road Traffic, 1997. ISBN 972-8098-25-1.
- [17] LEITAO, MIGUEL, ALEXANDRA MOREIRA, J. A. SANTOS, A. A. SOUSA F. N. FERREIRA: *Evaluation of Driving Education Methods in a Driving Simulator*. <<http://citeseer.ist.psu.edu/leitao99evaluation.html>>.
- [18] NABATILAN, LARRY BARGOLA: *Factors that influence visual attention and their effects on safety in driving: an eye movement tracking approach*. Dissertation, Graduate Faculty of The Louisiana State University and Agricultural and Mechanical College, August 2007.
- [19] S, MATTES: *The lane-change-task as a tool for driver distraction evaluation*. Quality of work and products in enterprises of the future, 57–60, 2003.
- [20] SHAPIRO WILK: *An analysis of variance test for normality (complete samples)*. Biometrika 1965 52(3-4):591-611.
- [21] SIGG, BENJAMIN: *java-forum.org - Einiges zur Geometrie (Punkte, Vektoren, Geraden)*. <<http://www.java-forum.org/de/viewtopic.php?p=35533#35533>>. Erstellt am 10. August 2004.
- [22] SUN MICROSYSTEMS, INC., <<http://java.sun.com/javase/6/docs/api/java/util/Arrays.html>>: *Arrays (Java Plattform SE 6)*. besucht am 9. Juni 2008.
- [23] W. ARAND, DR. ING. P. KUPKE PROF. DR.-ING. HABIL.: *Anforderungen an Fahr-simulatoren zur Untersuchung des Fahrer-Fahrzeug-Verhaltens sowie der verkehrs-technisch relevanten Eigenschaften von Straßenentwürfen*, 1982.
- [24] WIERWILLE: *Physiological measures of aircrew mental workload*. Hum Factors, (21):549574, 1979.

## **A Anhang**