

# **Absicherung hochautomatisierten Fahrens durch passiven virtuellen Dauerlaufstest**

Zur Erlangung des akademischen Grades eines  
DOKTOR-INGENIEURS  
von der KIT-Fakultät für  
Elektrotechnik und Informationstechnik  
des Karlsruher Instituts für Technologie (KIT)  
genehmigte

DISSERTATION

von  
Dipl.-Ing. kyb. Alexander Georg König  
geb. in Bielefeld

Tag der mündlichen Prüfung:	30.04.2021
Hauptreferent:	Prof. Dr.-Ing. Sören Hohmann
Korreferent:	Prof. Dr.-Ing. Eric Sax



# Vorwort

Die vorliegende Arbeit ist in einer Kooperation des Instituts für Regelungs- und Steuerungssysteme (IRS) des Karlsruher Instituts für Technologie (KIT) mit der BMW Group in München entstanden. Bei der Erstellung der Arbeit wurde ich von einer Vielzahl an Personen unterstützt. Bei einigen von ihnen möchte ich mich besonders bedanken.

Bedanken möchte ich mich bei Herrn Prof. Dr.-Ing. Sören Hohmann für die engagierte und geduldige Betreuung der Arbeit. Die Diskussionen und Ideen in regelmäßigen Betreuungsterminen haben wesentlich zum Konzept der Arbeit beigetragen.

Weiterer Dank gilt Herrn Prof. Dr.-Ing. Eric Sax für die Übernahme des Korreferats.

Weiterhin richtet sich mein Dank an meinen unternehmensseitigen Betreuer Dr. Dirk Ahrens sowie meine Mitdoktoranden Dr. Tobias Rehder, Dr. Wolfgang Müntst und Leopold Walking für die intensive Zusammenarbeit.

Mein Dank gilt auch meinen (damaligen) Studenten Kathrin Witzlsperger, Florin Leutwiler und Michael Gutbrod für die Unterstützung meiner Arbeit als Bachelorandin, Praktikant und Masterand.

Besonders bedanken möchte ich mich bei Dr. Stefan Schwab für seine detaillierten und hilfreichen Korrekturvorschläge am Ende der Promotion, durch die diese Arbeit erst die erforderliche Reife erreichen konnte.

Zuletzt gilt mein besonderer Dank meiner Frau Helene für ihre Unterstützung und Motivation über die lange Bearbeitungszeit, ohne die diese Arbeit nie fertig gestellt worden wäre.

München, im Februar 2021



# Kurzfassung

Die Entwicklung des hochautomatisierten Fahrens (HAF) schreitet in hoher Geschwindigkeit voran. Verschiedene Fahrzeughersteller haben erste Derivate mit automatisierten Fahrfunktionen, die keine Überwachung durch den Fahrer mehr erfordern, für die kommenden Jahre angekündigt. Es fehlt allerdings noch der Sicherheitsnachweis, dass HAF eine insgesamt positive Risikobilanz gemäß den Anforderungen der vom deutschen Verkehrsministerium eingesetzten Ethik-Kommission hat und somit durch die Einführung dieser Funktion die allgemeine Gefahr von Personenschäden im Straßenverkehr nicht zunimmt. Die aktuell verfügbaren Ansätze zur Erbringung eines Sicherheitsnachweises für Assistenzsysteme sind für HAF ungeeignet. Die wenigen theoretisch möglichen Methoden wie ein Nachweis durch Dauerlaufstest sind mit unrealistisch hohem Kosten- und Zeitaufwand verbunden und somit nicht industriell einsetzbar. In dieser Arbeit wird daher eine neue Absicherungsmethode erarbeitet, die einen Beitrag zum Schließen dieser Lücke leistet.

Bei heutigen Entwicklungen von Fahrerassistenzfunktionen kommen nicht-statistische Absicherungsmethoden zum Einsatz. Die Komplexität der Absicherungsaufgabe wird bei diesen dadurch deutlich reduziert, dass ein wesentlicher Teil des Sicherheitskonzepts auf der permanenten Überwachung der Funktion durch den menschlichen Fahrer beruht, der in kritischen Situationen eingreifen muss. Eine Absicherung ist daher nur noch hinsichtlich der Beherrschbarkeit durch den Fahrer notwendig.

Die Fahrerüberwachung kann beim HAF nicht vorausgesetzt werden, sodass die HAF-Funktion selbst alle auftretenden Situationen bis zu einer Fahrerübernahme beherrschen muss. Zusammen mit der großen Situationsvielfalt und dem Einsatz maschinell trainierter Algorithmen erhöht dies die Komplexität der Absicherungsaufgabe in einem Maße, dass die bisher genutzten Absicherungsverfahren an ihre Grenzen stoßen. Dies motiviert den Ansatz „passives HAF“ zur statistischen Absicherung des hochautomatisierten Fahrens.

Grundidee dieses Ansatzes ist es, einen Teil der jedes Jahr von menschlichen Fahrern gefahrenen Kilometer zur Absicherung zu nutzen, indem ein Teil der Fahrzeuge zusätzlich mit den für HAF erforderlichen Sensoren und Steuergeräten ausgestattet wird. Diese entsprechen dem für den späteren Kundeneinsatz vorgesehen Umfang der HAF-Funktion bis zu der Stelle, an dem Steuer- und Regeleingriffe in die Aktorik des Fahrzeugs erfolgen.

Statt der Umsetzung der vorgegebenen Stellgrößen durch die Fahrzeugaktorik wird der Ausgang der HAF-Funktion mit den vom menschlichen Fahrer tatsächlich durchgeführten Fahrhandlungen verglichen. Die Funktion läuft somit nur „passiv“ im Hintergrund mit, sodass aus fehlerhaften Aktorvorgaben kein Sicherheitsrisiko resultieren kann. Hierdurch entsteht ein offener Regelkreis, sodass der weitere Verlauf des Szenarios bei Eingriff der HAF-Funktion zunächst unbekannt ist. Der Regelkreis wird geschlossen, indem immer dann anhand der durch die Sensorik aufgenommenen Daten ein Simulationsszenario generiert wird, wenn die Fahrhandlungen vom HAF und menschlichem Fahrer voneinander abweichen. Anhand der

Simulation kann überprüft werden, ob die geplante Handlung des HAF zu einer kritischen Situation geführt hätte. Zur Erhöhung der Validität der Simulation werden individualisierte Fahrermodelle verwendet, deren Entwicklung und Zuordnung Teil der entwickelten Methode ist.

Ziel des Ansatzes ist die Nutzung der im realen Straßenverkehr vorkommenden Szenarien als Ausgangspunkt für die Absicherung des HAF mittels Simulation und anschließender Kritikalitätsanalyse. So kann fehlerhaftes Fahrverhalten erkannt und für eine Verbesserung der Funktion verwendet werden.

Die Ergebnisse der Arbeit zeigen, dass mit Hilfe dieser Methode fehlerhafte Fahrentscheidungen des HAF gefunden werden können, welche bei aktivem Funktionsausgang zu kritischen Situationen geführt hätten. Vorteile der Methode sind, dass eine gefahrlose und vergleichsweise kostengünstige und zeiteffiziente Absicherung möglich ist, die aufgrund der Nutzung realer Verkehrsszenarien eine hohe Validität aufweist.

# Abstract

Development of highly automated driving (HAD) is progressing quickly. Various car manufacturers have announced that they will offer a self-driving function without the requirement of driver surveillance within the next years. However, the evidence that HAD is at least as safe as human driving and will not lead to more traffic injuries has still not been produced. Current approaches of verification and validation used for driver assistance systems are not suitable for HAD. A statistical proof through test drives is neither economically nor timely feasible. Therefore, as a contribution for validation of HAD a statistical validation approach is proposed.

Non-statistical methods for system verification are currently employed in driver assistance systems (DAS). The complexity of verification for DAS is reduced by relying on the human driver for constantly monitoring the driving actions. For this reason it is sufficient to verify the controllability by the driver for verification of DAS.

Since monitoring of driving actions cannot be assumed for HAD, the HAD-function has to be capable of controlling all situations itself until take-over of the human driver. Together with the almost infinitely high number of scenarios and the use of machine learning algorithms, this increases the complexity of verification and validation such that current verification methods are not applicable. Instead, a statistical approach is proposed, namely „passive HAD“.

The main idea of this approach is to use a fraction of the billions of kilometers driven every year by everyday drivers for testing of HAD. For this purpose, vehicles are equipped with sensors and control units appropriate for HAD. The HAD functionality is working as intended up to the point where actuators of the vehicle like steering or brakes have to be accessed. Instead of realizing the output of the HAD function, the driving actions of the human driver are executed. The HAD function is running in the background passively, in order to not endanger the customers or other traffic participants.

If the output of the driving function differs from the human drivers' driving decisions, an open-loop problem results which prevents verification of the HAD function. To enable verification, the loop is closed by generating a simulation scenario based on the data acquired by the HAD sensors. In order to increase validity of the simulation, machine learning driver models developed in this thesis are matched to the traffic participants surrounding the passive HAD vehicle to be used in the simulation. Instead of the real life scenario influenced by the driving decisions of the human driver, the simulated scenario is assessed for criticality such that potentially erroneous behavior of the HAD function can be detected.

The goal of the passive HAD approach is to use real life sensor data for the generation of simulation scenarios to assess the performance of the HAD function with high validity. Results show that with the presented approach, erroneous driving decisions of the HAD function which would have led to critical situations can be detected in a safe way. Furthermore passive HAD

enables an economical and efficient verification with high validity due to the use of real life traffic scenarios.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Hochautomatisiertes Fahren	1
1.1.1	Aktuelle Entwicklungen und Key Player	2
1.1.2	Begriffsdefinitionen des automatisierten Fahrens	2
1.2	Absicherung des hochautomatisierten Fahrens	4
1.2.1	Rechtliche Anforderungen an ein Absicherungsergebnis	5
1.2.2	Ethische Anforderungen an ein Absicherungsergebnis	6
<b>2</b>	<b>Stand der Forschung und Technik: Aktuelle Absicherungsmethoden</b>	<b>9</b>
2.1	Anforderungsbasierte Absicherung	11
2.2	Verifikation	14
2.3	Statistische Absicherung	15
2.3.1	Testfahrten	15
2.3.2	Simulation	17
2.3.3	Graduelle Einführung des automatisierten Fahrens	19
2.4	Bewertung der Absicherungsansätze	20
<b>3</b>	<b>Konzept und Aufbau des passiven HAF</b>	<b>23</b>
3.1	Konzept des passiven HAF	23
3.1.1	Zentrale Bausteine	23
3.1.2	Erzielbare Absicherungsleistung des Ansatzes	27
3.2	Simulation der Fahrfunktion auf Grundlage realer Verkehrsszenarien	28
3.3	Simulationsumgebung für das passive HAF	29
3.3.1	Anforderungen an die Simulationsumgebung	29
3.3.2	Kriterien zur Bestimmung des optimalen Fahrermodells	32
3.3.3	Auswahl der Simulationsumgebung	33
3.3.4	Aufbau der gewählten Simulationsumgebung	35
3.3.5	Simulationsablauf	36
3.3.6	Fahrermodell	38
3.4	Szenarioerstellung aus Sensordaten	42
3.4.1	Datenvorverarbeitung von Spur- und Objektdaten	42
3.4.2	Generierung des Simulationsszenarios	43
3.5	Zuordnung von maschinell gelernten Fahrermodellen	44

<b>4</b>	<b>Maschinell gelernte Modelle zur Simulation des Umgebungsverkehrs</b>	<b>47</b>
4.1	Modellierung des Spurwechselverhaltens durch maschinelles Lernen	47
4.2	Modellierung mit Bayes'schen Netzen	49
4.2.1	Grundlagen	50
4.2.2	Modellierung von Spurwechselentscheidungen	61
4.2.3	Hybride Bayes'sche Netze	63
4.2.4	Inferenz in hybriden Netzen	65
4.2.5	Training von hybriden Bayes'schen Netzen	70
4.3	Modellierung mit Input-Output Hidden Markov Modellen	72
4.3.1	Grundlagen	73
4.3.2	Training von IOHMM	76
4.3.3	Modellierung von Spurwechselentscheidungen	77
4.4	Modellierung mit tiefen feedforward neuronalen Netzen	77
4.4.1	Grundlagen	78
4.4.2	Modellierung von Spurwechselentscheidungen mit tiefen neuronalen Netzen	79
4.5	Vergleich konzeptbedingter Vor- und Nachteile der Ansätze	80
<b>5</b>	<b>Kritikalitätserkennung</b>	<b>83</b>
5.1	Kritikalitätskriterium 1: Modifizierte Zeit bis zur Kollision	84
5.2	Kritikalitätskriterium 2: Komfort- und Fahrdynamikgrenzen	85
5.3	Kritikalitätskriterium 3: Letztmöglicher Bremszeitpunkt	86
5.4	Gesamtkritikalität	87
<b>6</b>	<b>Ergebnisse</b>	<b>89</b>
6.1	Validierung der Fahrermodellierung	89
6.1.1	Modellierung mit Simulationsdaten	89
6.1.2	Modellierung mit Realdaten	95
6.2	Verifikation der Szenarioerstellung	98
6.2.1	Szenarioerstellung aus Simulationsdaten	98
6.2.2	Rekonstruktion eines Simulationsszenarios aus Realdaten	108
6.3	Validierung der Kritikalitätserkennung	109
6.3.1	Fehlerfreie Fahrt	110
6.3.2	Fehlerbehaftete Fahrt	110
<b>7</b>	<b>Zusammenfassung und Fazit</b>	<b>113</b>
	<b>Literaturverzeichnis</b>	<b>XXI</b>

# Abbildungsverzeichnis

2.1	Regelndes und steuerndes Assistenzsystem	10
2.2	V-Modell der sicherheitsgerichteten Entwicklung	11
2.3	Onlineverifikation	14
3.1	Strukturdiagramm des Konzeptes passives HAF	25
3.2	Training der Fahrermodelle	26
3.3	Fahrzeug mit Koordinatensystem	30
3.4	Aufbau des Verkehrssimulationsprogramms PELOPS	36
3.5	Wiedemann-Folgemodell	39
3.6	Prozessablauf des passiven HAF	46
4.1	Nomenklatur der Fahrzeuge	48
4.2	Label der Trainingsdaten	49
4.3	Beispielhaftes Bayes'sches Netz	54
4.4	Markov-Netz nach Moralisierung	57
4.5	Cliquenbaum	57
4.6	Bayes'sches Netz zur Spurwechselmodellierung	63
4.7	Kollabieren von zwei Gaussverteilungen	66
4.8	Hidden Markov Model	73
4.9	Input-Output Hidden Markov Model	75
4.10	Modellierung des Spurwechsels als IOHMM	77
4.11	Modellierung des Spurwechsels als DNN	80
5.1	Fahrverhaltens-Beschleunigungs-Diagramm	86
6.1	Beispiel für ROC	92
6.2	ROCs bei simulierten Daten im Vergleich	93
6.3	ROC des mit realen Daten trainierten und evaluierten Netzes	95
6.4	ROCs der DNN nach fahrerindividuellem Training	97
6.5	Originalszenario 1 für Rekonstruktion	99
6.6	In Simulation berücksichtigter Bereich	99
6.7	Vergleich $x$ - und $y$ -Position Szenario 1	100
6.8	Geschwindigkeit Szenario 1	101
6.9	Originalszenario 2 für Rekonstruktion	101
6.10	Vergleich $x$ - und $y$ -Position Szenario 2	102
6.11	Geschwindigkeit Szenario 2	103
6.12	Originalszenario 3 für Rekonstruktion	103

---

6.13	Vergleich $x$ - und $y$ -Position Szenario 3 . . . . .	104
6.14	Geschwindigkeit Szenario 3 . . . . .	105
6.15	Originalszenario 4 für Rekonstruktion . . . . .	106
6.16	Vergleich $x$ - und $y$ -Position Szenario 4 . . . . .	106
6.17	Geschwindigkeit Szenario 4 . . . . .	107
6.18	Kamerabilder des Testfahrzeugs . . . . .	108
6.19	Visualisierung der vom Umfeldmodell zur Verfügung gestellten Daten . . . . .	109
6.20	Aus Realdaten rekonstruiertes Simulationsszenario . . . . .	109
6.21	Ursprungsszenario der kritischen Situation . . . . .	110
6.22	Fehlerhafte HAF-Funktion . . . . .	111
6.23	Kritisches Szenario . . . . .	112
6.24	Auflösung kritisches Szenario . . . . .	112

# Tabellenverzeichnis

2.1	Merkmale zur Situationsbeschreibung . . . . .	12
2.2	Sicherheitsleistung auf Autobahnen . . . . .	16
3.1	Absicherungsleistung nach Anzahl der Fahrzeuge und Fahrertyp . . . . .	27
3.2	Drei-Ebenen-Hierarchie und Drei-Ebenen-Modell . . . . .	31
3.3	Erfüllung der Anforderungen durch IPG Carmaker und PELOPS . . . . .	35
4.1	Eingangsgrößen des zu modellierenden Fahrers . . . . .	48
4.2	Zufallsvariablen mit Einfluss auf die Spurwechselentscheidung . . . . .	62
4.3	Wahrscheinlichkeitsverteilungen . . . . .	64
6.1	Performancevergleich der Modellierungsansätze . . . . .	92
6.2	Klassifikationsdauer . . . . .	94
6.3	Performance individueller Fahrermodelle . . . . .	96
6.4	Bewertungsvariablen Szenario 1 . . . . .	100
6.5	Bewertungsvariablen Szenario 2 . . . . .	102
6.6	Bewertungsvariablen Szenario 3 . . . . .	104
6.7	Bewertungsvariablen Szenario 4 . . . . .	105
6.8	Kritikalitätsmesswerte Szenarioanfang . . . . .	111
6.9	Kritikalitätsmesswerte Szenariomitte . . . . .	111
6.10	Kritikalitätsmesswerte Szenarioende . . . . .	112



# Abkürzungen und Symbole

## Abkürzungen

---

Abkürzung	Beschreibung
ABS	Anti-Blockiersystem
ACC	Adaptive Cruise Control
AUC	Area under Curve
BMW	Bayrische Motorenwerke AG
BN	Bayes'sches Netz
CLG	Conditional Linear Gaussian
DAS	Driver Assistant System
DNN	Deep Neural Network
ESP	Elektronisches Stabilitätsprogramm
FAS	Fahrerassistenz-Systeme
FN	False Negative
FP	False Positive
FPR	False Positive Rate
HAD	Highly Automated Driving
HAF	Hochautomatisiertes Fahren
HMM	Hidden Markov Model
ika	Institut für Kraftfahrzeuge
IOHMM	Input-Output Hidden Markov Model
LG	Linear Gaussian
LIDAR	Light Detection And Ranging
LKW	Lastkraftwagen
MTTC	Modified Time to Collision
PELOPS	Programm zur Entwicklung längsdynamischer mikroskopischer Verkehrsprozesse in systemrelevanter Umgebung
PKW	Personenkraftwagen
RADAR	Radio Detection And Ranging
ROC	Receiver Operator Characteristic
RWTH	Rheinisch-Westfälische Technische Hochschule
SAE	Society of Automotive Engineers
SHA	Spurhalteassistent
TLSB	Time to Last Second Braking
TN	True Negative

---

<b>Abkürzung</b>	<b>Beschreibung</b>
TOR	Take-over Request
TP	True Positive
TPR	True Positive Rate
TRM	Traffic Modelling
TTC	Time to Collision
VAF	Vollautomatisiertes Fahren



# Symbole

Symbol	Beschreibung
$A$	Matrix mit Transitionswahrscheinlichkeiten im IOHMM
$T_{\text{MTTC}}$	Modifizierte Zeit bis zur Kollision (MTTC)
$a_0$	Beschleunigung am Anfang der Simulation
$a_{\text{diff}}$	Differenzbeschleunigung zwischen zwei Fahrzeugen
$a_{F,\text{max}}$	Maximal mögliche Beschleunigung des folgenden Fahrzeugs
$a_{ij}$	Transitionswahrscheinlichkeit vom Zustand $i$ zu Zustand $j$
$a_j$	Ausgangsneuron $j$
$\alpha$	Wahrscheinlichkeitsverteilung von $X$ unter Berücksichtigung aller bis zu diesem Zeitpunkt erfolgten Beobachtungen
$a_{\text{max,komf}}$	Maximum der Beschleunigung, um innerhalb des Komfortbereichs zu bleiben
$a_{\text{max,phys}}$	physikalisches Maximum der Beschleunigung ohne Haftreibung zu verlieren
$a_{\kappa}^{\text{obj}}$	Beschleunigung von Fahrzeug $\kappa$
<b>B</b>	Bayes'sches Netz
$B$	Matrix mit Emissionswahrscheinlichkeiten im IOHMM
$\beta$	Wahrscheinlichkeitsverteilung von $X$ unter Berücksichtigung aller ab diesem Zeitpunkt erfolgten Beobachtungen
$b_{ij}$	Emissionswahrscheinlichkeit, mit der Zustand $j$ Beobachtung $i$ emittiert
$b_m$	Bewertung der Spur $m$
$b_m^p$	Bewertung der Spur $m$ hinsichtlich ihrer Übereinstimmung mit der präferierten Spur $m_p$ des Fahrers
$b_m^r$	Bewertung der Spur $m$ hinsichtlich ihrer Entfernung zur rechtsten Spur
$b_m^d$	Bewertung der Spur $m$ hinsichtlich drängelnder Folgefahrzeuge
$b_m^k$	Bewertung der Spur $m$ wenn ein anderes Fahrzeug auf diese wechseln möchte
$b_m^v$	Bewertung der Spur $m$ hinsichtlich ihrer Geschwindigkeit $v$
$C$	Clique
$\chi$	Zufallszahl aus Gleichverteilung $[0, 0,2]$
$c$	Zufriedenheit mit eigener Spur
$c_{\text{kr}}$	Zufriedenheit mit Abweichung vom Rechtsfahrgebot
$c_{\text{pre}}$	Zufriedenheit mit vorausfahrendem Fahrzeug
$c_{\text{vel}}$	Zufriedenheit mit durchschnittlicher Geschwindigkeit der Spur
$c_L$	Zufriedenheit mit linker Nachbarspur
$c_R$	Zufriedenheit mit rechter Nachbarspur
$c_{\text{suc}}$	Zufriedenheit mit folgendem Fahrzeug
$cur$	Krümmung der Spur
$cur_{\text{max}}$	maximale Krümmung der Spur
$d_{\text{diff}}$	Abstand zwischen zwei Fahrzeugen
$\Delta t$	Abtastzeit
$\Delta T_c$	Zykluszeit der Verkehrssimulation PELOPS
$d_{\kappa}^{\text{obj}}$	Abstand von S zu Fahrzeug $\kappa$

Symbol	Beschreibung
$D$	Fahrer
$d_t$	Daten zum Zeitpunkt $t$
$d_{\text{test}}$	Anzahl an Testkilometern
$d_{\text{test},\text{fat}}$	notwendige Anzahl an Testkilometern für den Nachweis einer geringeren Anzahl an Toten bei Einsatz von HAF
$d_{\text{test}}^{\text{initial}}$	notwendige Anzahl an Testkilometern zur Bestimmung der Grenzen der Sicherheitsleistung
$d_{\text{test},\text{inj}}$	notwendige Anzahl an Testkilometern für den Nachweis einer geringeren Anzahl an Verletzten bei Einsatz von HAF
$e$	Vektor mit Wissen über Zufallsvariablen
$e_{\text{Erfolg}}$	Wahrscheinlichkeit für eine erfolgreiche Freigabe der HAF-Funktion
$e_{\text{Irrtum}}$	Wahrscheinlichkeit für eine irrtümliche Freigabe der HAF-Funktion
$e_j$	Eingangsdaten des Ausgangsneurons $a_j$
$El_G$	Elternknoten im Graphen $G$
$\eta$	Lernrate eines neuronalen Netzes
$\mathbf{E}$	Vector von Zufallsvariablen, über die Wissen vorliegt
$\mathcal{F}$	Fahrer von HAF-Fahrzeugen
$F$	Fehler eines DNN
$F$	folgendes Fahrzeug
$f_S$	Fehler des Spurwechselmodells $S$
$\bar{f}_S$	durchschnittlicher Fehler des Spurwechselmodells $S$
$\mathcal{G}$	Gesellschaft
$g$	<i>a posteriori</i> Emissionswahrscheinlichkeit
$\gamma$	Label
$\gamma_{\text{aktiv}}$	Gesamtwahrscheinlichkeitswert für ein aktives Label
$\gamma_{\text{aktiv}_-}$	Mindestwahrscheinlichkeitswert für ein aktives Label
$\gamma_{\text{inaktiv}}$	Gesamtwahrscheinlichkeitswert für ein inaktives Label
$\gamma_{a,t}$	<i>a posteriori</i> Wahrscheinlichkeit der Ausprägung $a$ eines Softmax-Knotens zum Zeitpunkt $t$
$\hat{\gamma}_{a,t}$	<i>a priori</i> Wahrscheinlichkeit der Ausprägung $a$ eines Softmax-Knotens zum Zeitpunkt $t$
$\hat{\gamma}$	Klassifikationsergebnis
$G$	azyklischer Graph
$h$	<i>a posteriori</i> Transitionswahrscheinlichkeit
$\mathcal{H}$	Markov-Netz
$I$	Menge an Unabhängigkeitsannahmen
$i$	Iterationsindex
$I$	induzierter Graph
$\iota$	Spur relativ zur Spur von $S$ mit $\iota \in \{R, 0, L\}$
$T_{\text{itgap},\kappa}$	inverser Zeitabstand von $S$ zu Fahrzeug $\kappa$
$j$	Iterationsindex
$k$	Anzahl an Ereignissen

Symbol	Beschreibung
$\kappa$	Fahrzeugposition relativ zu S mit $\kappa \in \{\{F, V\}\{R, L\}\}$
$k_D$	Kooperationsbereitschaft eines Fahrers $D$
$K_{Ges}$	Gesamtkritikalität einer Situation
$K_{TLB}$	Rückgabewert des Kritikalitätskennfelds für $T_{LB}$
$K_{T_{MTTC}}$	Rückgabewert des Kritikalitätskennfelds für $T_{MTTC}$
$K_{V_{komf}}$	Rückgabewert des Kritikalitätskennfelds für $V_{komf}$
$K_{V_{phys}}$	Rückgabewert des Kritikalitätskennfelds für $V_{phys}$
L	links von S fahrende Fahrzeuge
$\Lambda$	Plausibilität, mit der ein Modell eine Beobachtung erklären kann. Optimierungsgröße beim maschinellen Lernen von BN und IOHMM
$\epsilon$	Fehlerschwelle
$\lambda$	durchschnittliche Anzahl an Ereignissen in einem definierten Intervall
$\lambda_{HAF}$	durchschnittliche Anzahl an Unfallereignissen in einem definierten Intervall bei Nutzung der HAF-Funktion
$\lambda_{Mensch}$	durchschnittliche Anzahl an Unfallereignissen in einem definierten Intervall bei menschlichem Fahrer
$e_{\iota}^{lane}$	Existenz der Spur $\iota$
LCI	Spurwechselentscheidung
M	Funktion zur Moralisierung eines BN
$m$	Spurindex einer Spur
$m_p$	Spurindex der präferierten Spur eines Fahrers
$m_r$	Spurindex der gewählten Spur eines Fahrers
$M_S$	Spurwechselmodell S
$\mu$	Mittelwert einer linearen Gaußverteilung
$n$	Endwert der Iteration
$\nu_{reg}$	Regularisierungsfaktor
$o$	Anzahl an Ausprägungen der Zufallsvariable
$e_{\kappa}^{obj}$	Existenz von Fahrzeug $\kappa$
$\mathcal{O}$	Ordnung des Laufzeitbedarfs
$o_k$	Verkehrsobjekt k
$\Omega$	Potenzmenge
$P$	Wahrscheinlichkeitsverteilung
$\Phi$	Vektor von Faktoren
$\phi$	Faktor
$\Pi$	Anfangsverteilung der Zustandswahrscheinlichkeiten im IOHMM
$SM$	Polynom, das Spurmarkierung definiert
$p_i$	Positionskordinaten $x$ und $y$ eines Sensordatenpunktes $i$
$P_{m_i}$	Wahrscheinlichkeit der Spur $m_i$
$\mathbf{P}_{M_S}$	Ausgangsvektor von $M_S$ mit Wahrscheinlichkeiten für einzelne Spuren
$p_{o_k}$	Positionskordinaten $x$ und $y$ eines Verkehrsobjekts $o_k$
$\psi$	Parametervektor
$\tilde{P}$	unnormalisierte Wahrscheinlichkeitsverteilung

Symbol	Beschreibung
$Q$	diskrete Zufallsvariable
$q$	Ausprägung der diskreten Zufallsvariable $Q$
$R$	rechts von S fahrende Fahrzeuge
$\mathcal{R}$	Eliminationsreihenfolge
$\mathbb{R}$	Menge der reellen Zahlen
$S$	mit HAF-Sensorik ausgestattetes Fahrzeug
$s$	zurückgelegte Strecke entlang einer Referenz-Spurlinie
$\Sigma$	Vektor von Varianzen von linearen Gaußverteilungen
$\sigma$	Varianz einer linearen Gaußverteilung
$Sit_{\text{ges}}$	Gesamtzahl an zu verifizierenden Situationen
$s_{\text{max}}$	obere Grenze eines Intervalls der Referenz-Spurlinie
$s_{\text{min}}$	untere Grenze eines Intervalls der Referenz-Spurlinie
$SP$	Safety Performance auf Autobahnen
$SP_{\text{fat}}$	Safety Performance für Personenschäden mit Todesfolge auf Autobahnen
$SP_{\text{inj}}$	Safety Performance für Personenschäden ohne Todesfolge auf Autobahnen
$SP_{\text{max}}$	obere Grenze der Safety Performance
$SP_{\text{min}}$	untere Grenze der Safety Performance
$s_{\text{suf}}$	hinreichender Geschwindigkeitsunterschied der Spur zur rechten Spur
$t$	Koordinaten entlang der Zeitachse
$t_{\text{alt}}$	letztmaliger Detektionszeitpunkt von $o_k$
$\mathcal{T}$	Cliquenbaum
$t_{\text{extrapol}}$	Zeitpunkt der extrapolierten Position von $o_k$
$\theta$	Parametervektor
$T_{\text{ittc},\kappa}$	inverse $TTC$ von S zu Fahrzeug $\kappa$
$T_{\text{LB}}$	Zeit bis zum letztmöglichen Bremszeitpunkt, um eine Kollision durch Bremsen zu verhindern
$t_{\text{mess}}$	Zeitraum über den Sensordaten gesammelt werden, nachdem eine Abweichung im Fahrverhalten zwischen HAF und menschlichem Fahrer detektiert wurde
$t_{\text{prep}}$	Zeit vor Spurwechsel-Ereignis, in der Trainingsdaten das Label Spurwechsel erhalten
$Sep$	Trennglied
$TTC$	Time to Collision
$T_{\text{Test}}$	Dauer eines Verifikationstests
$T_{\text{THW}}$	Time Headway
$U$	Eingangsdaten im IOHMM
$u$	Eingang eines Neurons
$V$	vorausfahrendes Fahrzeug
$a$	Beschleunigung
$v$	Geschwindigkeit
$v_0$	Geschwindigkeit am Anfang der Simulation
$v_{\text{diff}}$	Differenzgeschwindigkeit zwischen zwei Fahrzeugen

Symbol	Beschreibung
VE	Funktion zur Variablenelimination
$v_{\text{exp}}$	Erwartete eigene Geschwindigkeit in der Spur
$v_{\text{exp}}^{\text{lane}}$	Erwartete durchschnittliche Geschwindigkeit der Spur
$v_j$	Geschwindigkeit des Fahrzeugs $j$
$V_{\text{komf}}$	Verhältnis von tatsächlicher Beschleunigung zu Komfort-Beschleunigungsgrenze $a_{\text{max, komf}}$
$v_{\text{max}}$	maximale Geschwindigkeit
$v_{\text{min}}$	minimale Geschwindigkeit
$v_{\kappa}^{\text{obj}}$	Geschwindigkeit von Fahrzeug $\kappa$
$V_{\text{phys}}$	Verhältnis von tatsächlicher Beschleunigung zu physikalischer Grenzbeschleunigung $a_{\text{max, phys}}$
$v_{\text{red}}$	Wunschgeschwindigkeit des Fahrers, die aufgrund von Informationen aus der Straßentopologie reduziert wurde
$v_S$	Fahrzeuggeschwindigkeit von S
$v_{\text{wunsch}}$	Wunschgeschwindigkeit eines Fahrers
$\hat{v}_{\text{wunsch}}$	geschätzte Wunschgeschwindigkeit eines Fahrers
$W$	Gewichtsmatrix
$w$	Summengewicht
$X$	Zufallsvariable $X$
$x$	Koordinaten entlang der Längsachse des Fahrzeugs
$x_0$	Position in x-Richtung am Anfang der Simulation
$x_1$	Ausprägung 1 der Zufallsvariable $X$
$\xi$	Ausprägung der Zufallsvariable $\Xi$
$\Xi$	kontinuierliche Zufallsvariable
$x_i$	Ausprägung $i$ der Zufallsvariable $X$
$\mathcal{X}$	Menge an Zufallsvariablen
$x_n$	Ausprägung $n$ der Zufallsvariable $X$
$\mathbf{X}$	Vektor von Zufallsvariablen $X$
$Y$	Zufallsvariable der Emissionswahrscheinlichkeiten
$\Upsilon$	kontinuierliche Zufallsvariable
$y$	Koordinaten entlang der Querachse des Fahrzeugs
$y_0$	Position in y-Richtung am Anfang der Simulation
$y_1$	Ausprägung 1 der Zufallsvariable $Y$
$v$	Ausprägung der kontinuierlichen Zufallsvariable $\Upsilon$
$y_i$	Ausprägung $i$ der Zufallsvariable $Y$
$Z$	zufälliges Ereignis
$z$	Ausgang eines Neurons
$\mathcal{Z}$	Normalisierungskonstante bzw. Zustandssumme



# 1 Einleitung

Aktuelle Fahrzeuge werden durch den Begriff Automobil (abgeleitet von griechisch autos „selbst“ und lateinisch mobilis „beweglich“ [KS89, S. 51]) nur teilweise richtig beschrieben. Spätestens seit dem ausgehenden 19. Jahrhundert konnten sich zwar erste Fahrzeuge selbstständig, d.h. ohne weitere externe Antriebskraft, bewegen [DK15, S. 2]. Erst heute rückt jedoch auch die Erfüllung des zweiten Bedeutungsteils der „Selbst-Beweglichkeit“, ein selbstständig ohne Fahrer fahrendes Automobil, in greifbare Nähe.

In den letzten hundert Jahren wuchs das Automobil heran vom exotischen Fortbewegungsmittel einiger Pioniere zum allgegenwärtigen Gefährt in praktisch allen Ländern der Welt. Die Entwicklung vollzog sich hierbei in Form von immer neuen Design-Varianten, wie z.B. Sports Utility Vehicle, oder leistungstärkerer [KB18a] und verbrauchsärmerer [Bun17a, S. 308f.] Motoren. Weitere Entwicklungsfelder waren die Einführung aktiver Sicherheitsfunktionen wie ABS (1979) und ESP (1995) [Ise06, S. 1f.], durch die die Kontrollierbarkeit von Fahrzeugen im fahrdynamischen Grenzbereich erhöht und somit die Anzahl an Unfällen reduziert werden konnte [Koc04]. Zusammen mit der Fortentwicklung passiver Sicherheitssysteme, die die Folgen von Unfällen abmildern (Knautschzonen 1959 [Dai09], Airbags 1978 [BK08, S. 220]) konnte die Anzahl an Verletzten und Toten in den letzten Jahrzehnten deutlich gesenkt werden. Dies erfolgt bei gleichzeitig steigender Anzahl an gefahrenen Kilometern [Des18], was die Effektivität der genannten Systeme verdeutlicht.

## 1.1 Hochautomatisiertes Fahren

Trotz der sinkenden Zahl von Verkehrstoten in den Industrieländern sterben jedes Jahr über eine Million Menschen weltweit im Straßenverkehr [Wor19], davon 3400 in Deutschland [ADA15, S. 1]. Nach [Nat08, S. 28f] sind mehr als 80 % aller Unfälle allein auf menschliche Fehler zurückzuführen. Das Ziel der Vision Zero des deutschen Verkehrssicherheitsrates ist es, eine weitere Reduktion von Toten und Verletzten im Straßenverkehr zu erreichen [Deu12]. Hierfür gilt es als erforderlich, die Schwächen menschlicher Fahrer soweit zu kompensieren, dass diese nicht mehr zu Unfällen führen. Daher zielen aktuelle Entwicklungen darauf den Fahrer soweit wie möglich von der Fahraufgabe zu entlasten und durch Systeme zu ersetzen, die ein automatisiertes Fahren ermöglichen.

Neben der Vermeidung menschlicher Fehler eröffnet die Automatisierung der Fahraufgabe außerdem ein komfortableres Reisen und neue Nutzungsmöglichkeiten der Zeit im Auto insbesondere für den Fahrer.

### 1.1.1 Aktuelle Entwicklungen und Key Player

Insbesondere der Aspekt gewonnener freier Zeit durch automatisiertes Fahren führt dazu, dass die Entwicklung in diesem Bereich nicht mehr allein von Unternehmen vorangetrieben wird, deren traditioneller Geschäftsbereich die Herstellung von Autos ist.

So hatten zum 1. April 2018 52 Unternehmen im US-amerikanischen Bundesstaat Kalifornien beim zuständigen Department of Motor Vehicles ein „Autonomous Vehicle Testing Permit“ beantragt [Cal19]. Dazu gehören bekannte Automobilhersteller wie Volkswagen, Daimler, General Motors, Nissan, Tesla oder BMW. Neben diesen haben aber auch Technologieunternehmen und Startups (z.B. Apple, Uber, Drive.ai oder Aurora) eine Testerlaubnis beantragt. Die Firma Alphabet (vormals Google) mit ihrer Tochter Waymo forscht sogar bereits seit dem Jahr 2009 in Bereich des automatisierten und autonomen Fahrens [Sil18].

Hierdurch entsteht ein Wettbewerb zwischen zwei Branchen, die bisher in verschiedenen Märkten aktiv waren und große Unterschiede hinsichtlich ihrer Entwicklungszyklen und -prozesse besaßen [SMF19][Fra16][Coh15]. Zu der einen Gruppe gehören die etablierten Automobilhersteller, die zum Teil schon seit über 100 Jahren Fahrzeuge herstellen. Zu der anderen Gruppe gehören etablierte Unternehmen aus der IT-Branche wie Alphabet oder Apple und eine Vielzahl an Startups [Dem19]. Diese Gruppe eint eine große Erfahrung im Bereich der Software-Entwicklung. Hinzu kommt eine hohe Risikotoleranz, da ein Scheitern im Bereich des automatisierten Fahrens nicht das bisherige Geschäftsmodell in entscheidender Weise berührt oder in der Gründung des Startups einkalkuliert ist.

Die Folge dieses Wettbewerbs ist, dass im Vergleich zu den bisherigen Innovationszyklen innerhalb der Automobilindustrie die Entwicklung des automatisierten Fahrens mit hohen Entwicklungsinvestitionen vorangetrieben wird.

### 1.1.2 Begriffsdefinitionen des automatisierten Fahrens

Unter dem Oberbegriff des automatisierten Fahrens sind verschiedene Ausprägungen gefasst, die nach [SAE14] in sechs Level eingeteilt werden. Ihre Entwicklung geht mit unterschiedlichen Herausforderungen einher, die im Folgenden näher beschrieben werden.

*Manuelles Fahren* bezeichnet das Fahren ohne Unterstützung des Fahrers durch Assistenzsysteme bei der Fahrzeugführung.

#### **Level 0**

*Fahrerassistenz-Systeme (FAS)* unterstützen den Fahrer bei seiner Fahrtätigkeit, indem sie in die Längs- oder/und Querverführung eingreifen. Der Fahrer hat die Aufgabe, die Systeme permanent zu überwachen, fehlerhafte Aktionen zu erkennen und zu unterbinden.

#### **Level 1 bzw. 2**

*Hochautomatisiertes Fahren (HAF)* bezeichnet eine automatisierte Fahrfunktion, bei welcher der Fahrer die Fahraufgabe in bestimmten Fahrszenarien vollständig an die Funktion übergeben kann. Eine Überwachung durch den Fahrer ist nicht mehr erforderlich. Stößt die Funktion an ihre Systemgrenzen oder detektiert einen Fehler, wird ein Take-over Request (TOR) gestartet, durch welchen die Fahraufgabe an den Fahrer zurückgegeben



wird. Gelingt dies nicht, wird durch die Fahrfunktion sichergestellt, dass das Fahrzeug einen sicheren Zustand einnimmt. Aus diesem Grund ist neben der technischen Entwicklung auch ein Gegenstand der Forschung, wie und nach welcher Zeit die Fahraufgabe sicher an den Fahrer übergeben werden kann.

### **Level 3**

*Vollautomatisiertes Fahren (VAF)* beschreibt eine Fahrfunktion, die das automatisierte Fahren in allen für den Straßenverkehr zugänglichen Bereichen erlaubt. Wie bei der HAF-Funktion wird die Fahraufgabe vollständig vom Fahrer an die VAF-Funktion übergeben, jedoch ohne die bei der HAF-Funktion vorliegenden Einschränkungen hinsichtlich bestimmter Szenarien wie der Fahrt auf Autobahnen. Typischerweise eignet sich die VAF-Funktion auch für den Stadtverkehr. Es muss jedoch weiterhin ein Fahrer anwesend sein, der bei Erreichen von Systemgrenzen oder Systemfehlern die Kontrolle übernehmen kann. Daher ist auch hier die Entwicklung einer Übernahmestrategie notwendig.

### **Level 4**

*Autonomes Fahren* ist die höchste Stufe des automatisierten Fahrens. Hierbei wird die Anwesenheit eines Fahrers nicht mehr vorausgesetzt, das Fahrzeug kann also ohne einen Insassen am Straßenverkehr teilnehmen. Im Gegensatz zu HAF- und VAF-Funktionen ist keine Übernahmestrategie notwendig.

### **Level 5**

Die vorliegende Arbeit beschäftigt sich mit Möglichkeiten der Absicherung des Automatisierungslevels 3: Hochautomatisiertes Fahren. Dabei ist die HAF-Funktion als eine Funktion definiert, die das automatisierte Fahren auf Straßen erlaubt, welche mehrere Fahrspuren pro Richtung aufweisen und bei der Fahrspuren unterschiedlicher Richtung baulich voneinander getrennt sind. Hierzu zählen beispielsweise Bundesautobahnen in Deutschland oder Freeways in den USA. Diese Szenarien zeichnen sich dadurch aus, dass in der Regel nur eine stark begrenzte Anzahl an Fahrmanövern (Freie Fahrt, Folgefahrt, Spurwechsel) und Situationen (nur PKW und LKW, keine Ampeln und Kreuzungen) erlaubt oder möglich sind, andererseits die Fahrt mit hoher Geschwindigkeit erfolgen kann.

Die Fahraufgabe kann von der HAF-Funktion übernommen werden, sobald der Fahrer auf die Autobahn aufgefahren ist. Die Funktion hält dabei alle gesetzlichen Anforderungen wie Geschwindigkeitsbeschränkungen, Abstand, Überholverbote, die Bildung von Rettungsgassen oder die Nutzung erlaubter und Vermeidung verbotener Spuren (bspw. Carpool-Lane, Taxi- und Busspuren) ein. Durch die Funktion wird das Fahrzeug in der Spur gehalten und führt selbstständig Spurwechsel zur Erreichung der optimalen Geschwindigkeit und des Navigationsziels aus.

Die Fahraufgabe wird übergeben, sobald eine Systemgrenze erreicht ist. Systemgrenzen sind zum Beispiel das Ende der Autobahn, Wetterbedingungen, die zu einer starken Einschränkung der Sensor-Funktionalität führen, oder spezielle Szenarien wie Mautstationen. Ein weiterer Grund für einen TOR ist der Ausfall oder die Detektion eines Fehlers in einem System in der Wirkkette der HAF-Funktion. Für den Fall, dass ein TOR nicht erfolgreich durchgeführt werden kann, begibt sich das Fahrzeug durch ein Minimum Risk Maneuver in einen sicheren Zustand, indem es beispielsweise auf den Seitenstreifen wechselt und dort zum Stehen kommt.

Damit Fahrentscheidungen autonom getroffen werden können, muss die HAF-Funktion Kenntnis über die Umwelt des Fahrzeugs besitzen. Hierzu wird die Umwelt mit Hilfe verschiedener Sensoren erfasst. Mögliche Sensoren sind nach [Die16, S. 409]:

*Optische Bilderfassung* Ein optisches Bilderfassungssystem ist gut für die Erkennung von anderen Verkehrsobjekten geeignet, allerdings schlecht für die Entfernungsschätzung. Selbst bei Stereokameras gelingt dies aktuell nur bis zu einem Abstand von 50 m, der Messfehler steigt quadratisch mit zunehmendem Abstand des Objekts vom Bilderfassungssystem.

*RADAR* Der RADAR (**R**adio **D**etection **A**nd **R**anging) verfügt über eine große Detektionsreichweite bis 200 m und ist gut zur Messung des longitudinalen Abstands geeignet, da der Messfehler in longitudinaler Richtung konstant bleibt. Texturen können allerdings schlecht erkannt werden, die laterale Auflösung ist ebenfalls gering.

*LIDAR* Der LIDAR (**L**ight **D**etection **A**nd **R**anging) verfügt über eine mittlere Detektionsreichweite bis 80 m und eine hohe Auflösung in longitudinaler und lateraler Richtung.

Typischerweise werden mehrere verschiedene Sensortechnologien eingesetzt und ihre Daten in einem Umfeldmodell fusioniert. Beispielsweise eignet sich ein optisches Bilderfassungssystem gut zur Erkennung von Spurmarkierung während die laterale Position von Objekten mit dem LIDAR effektiv bestimmt werden kann. Die Fusion beider Informationen erlaubt eine präzise Spurzuordnung umgebender Verkehrsteilnehmer.

## 1.2 Absicherung des hochautomatisierten Fahrens

Die Übergänge zwischen den Automatisierungsstufen erscheinen zunächst homogen, da in jeder Stufe weitere Aspekte der Fahraufgabe vom Fahrer an das Fahrzeug übergeben werden. Tatsächlich existiert aber am Übergang von Automatisierungsstufe 2 zu 3 eine qualitative Änderung, da hier erstmalig die Verantwortung für die Folgen der Fahrentscheidungen teilweise vom Fahrer auf das Fahrzeug übergeht.

Die Hauptaufgabe von Fahrerassistenz-Systemen ist es den Fahrer zu entlasten. Dazu hält z. B. die Funktion Adaptive Cruise Control (ACC) durch limitiertes Beschleunigen und Bremsen einen definierten Zeitabstand zum Vorderfahrzeug ein. Ein weiteres Beispiel ist die Funktion Spurhalteassistenten (SHA), die durch Stellen eines limitierten Lenkmoments bei relativ geradem Straßenverlauf die Spur hält. Hierbei hat der Fahrer jedoch stets die Aufgabe, die Fahrzeugführung zu überwachen und zu unterstützen. Beispielsweise kann die Funktion ACC nur bis maximal  $a_{\max} = -4,5 \frac{m}{s^2}$  verzögern, sodass bei stärkerem Bremsen des Vorderfahrzeugs der Fahrer dies bemerken und eingreifen muss. Der Fahrer wird bei der Auslegung und Erkennung von Systemgrenzen der Fahrerassistenz-Systeme stets mit berücksichtigt.

Beim hochautomatisierten Fahren ist es dem Fahrer hingegen erstmals erlaubt, das Fahrzeug während der Fahrt innerhalb der Systemgrenzen der HAF-Funktion nicht mehr überwachen zu müssen. Er darf definierten Nebentätigkeiten nachgehen und muss nicht mehr selbst Fehlfunktionen oder Systemgrenzen des Systems erkennen. Stattdessen wird er vom System rechtzeitig und hinreichend deutlich informiert, wenn eine Übernahme erforderlich ist (TOR).

Hierdurch entstehen neue Fragen: Wer haftet für potentiell fehlerhafte Fahrentscheidungen des

HAF-Fahrzeugs? Welche ethischen Anforderungen an die Fahrweise des selbstfahrenden Systems gibt es? Und welche technischen Möglichkeiten gibt es, zu Unfällen führendes Verhalten im Vorhinein ausschließen oder rechtzeitig erkennen und unterbinden zu können?

### 1.2.1 Rechtliche Anforderungen an ein Absicherungsergebnis

Allgemein wird erwartet, dass die Gesamtzahl der Unfälle im Zuge der Einführung des automatisierten Fahrens verringert werden kann, da menschliche Fehler verhindert werden. Hierbei handelt es sich jedoch um eine statistische Aussage, die nicht bedeutet, dass automatisierte Fahrzeuge ab Level 3 in keinerlei Unfälle mehr in Szenarien verwickelt sein werden, in denen der menschliche Fahrer die Fahraufgabe abgeben durfte. Kommt es zu einem Unfall, bei dem ein entsprechend ausgestattetes Auto involviert ist, so stellt sich die Frage nach der Unfallschuld und der Haftung für entstehende Unfallschäden.

Im Jahr 2017 wurde hierzu vom Bundestag ein Gesetz verabschiedet [Deu17]. Durch dieses wird zum einen hochautomatisiertes und vollautomatisiertes Fahren explizit erlaubt. Zum anderen wird klargestellt, dass der Fahrer weiterhin in der Verantwortung des Fahrzeugführers bleibt und bei Unfallschuld gegenüber Unfallgegnern schadensersatzpflichtig ist. Hiervon unbenommen bleiben jedoch mögliche Regressforderungen des Fahrers oder seiner Versicherung gegenüber dem Hersteller des Fahrzeugs, sofern sich das Fahrzeug zum Zeitpunkt des Unfalls im hoch- oder vollautomatisierten Fahrbetrieb befand. Um dies im Nachhinein feststellen zu können sieht der Gesetzgeber explizit den Einsatz einer „Blackbox“ im automatisierten Fahrzeug vor, welche die wesentlichen Fahrdaten speichert, insbesondere den Zustand des automatisierten Fahrsystems.

Auch wenn hierdurch die Haftungsfrage zunächst klar geregelt erscheint, stellt sich die Frage nach der Höhe der auf die Hersteller potentiell zukommenden Forderungen. In [Smi17] wird dargestellt, wie dies den Verkauf und Einsatz von hoch- und vollautomatisierten Fahrzeugen beeinflussen könnte.

Hersteller müssen für automatisierte Fahrzeuge der SAE-Level 3 bis 5 allgemein mit höheren Haftungskosten als für manuell geführte oder teilautomatisierte rechnen. Der Grund hierfür ist, dass bei manuell geführten Fahrzeugen in der Regel der Kunde selbst haftet, während bei hoch- oder vollautomatisierter Fahrt letztendlich der Hersteller haften muss, sofern ein Systemfehler vorlag. Dies könnte auf den Verkaufspreis der Fahrzeuge umgelegt werden.

Allgemein ist die Höhe möglicher Kosten mangels statistischer Daten für neue Systeme unbekannt und muss daher abgeschätzt werden. Mit steigender Komplexität des zu beurteilenden Systems nimmt auch die Schwierigkeit einer realistischen Abschätzung zu. Grund hierfür ist, dass Unsicherheiten insbesondere hinsichtlich zu Personenschäden führenden Fehlern üblicherweise konservativ abgeschätzt werden, also vom ungünstigsten Fall ausgegangen wird. Je mehr Faktoren eines Systems unbekannt sind, desto stärkeren Einfluss haben sie auf die Abschätzung der potentiellen Haftungsrisiken.

Bei automatisierten Fahrfunktionen handelt es sich einerseits um im Vergleich zu bisherigen Fahrzeugfunktionen komplexe Systeme [WW13], da sie über einen längeren Zeitraum in nicht nur auf Spezialsituationen beschränkten realen Szenarien die Fahraufgabe übernehmen. Hierfür ist eine gegenüber aktuell im Markt befindlichen Fahrerassistenzfunktionen deutlich verbesserte Umfeldwahrnehmung notwendig, die als Grundlage für eine generische Funktionslogik

dient, welche in allen im vorgesehenen Einsatzbereich möglichen Situationen eine sichere Fahrt gewährleistet.

Andererseits liegt für diese neuen Funktionen keine belastbare statistische Datengrundlage zur Risikobewertung vor. Daher könnte bei konservativer Schätzung der Aufschlag auf den Verkaufspreis so hoch sein, dass Kunden sich für Neuwagen ohne automatisierte Fahrfunktion der SAE-Level 3 bis 5 entscheiden, obwohl diese möglicherweise zu insgesamt weniger Unfällen und somit auch gesamtwirtschaftlich gesehen geringere Unfallschäden führen.

Für die Einführung des automatisierten Fahrens ist es somit für die Hersteller und ihre Versicherer von Bedeutung, die Wahrscheinlichkeit von Unfällen, die durch HAF-Funktionen verursacht werden könnten, und somit das Risiko und die Höhe möglicher Schadensersatzforderungen einschätzen zu können. Hierfür ist ein quantitatives Absicherungsergebnis unerlässlich.

### **1.2.2 Ethische Anforderungen an ein Absicherungsergebnis**

Neben den rechtlichen Herausforderungen des automatisierten Fahrens müssen für den Automobilbereich neue ethische Fragen beantwortet werden. Aktuelle Diskussionen beschäftigen sich hierbei häufig mit dem ethisch korrekten Verhalten in konstruierten, moralisch besonders schwer zu bewertenden Situationen, in denen ein automatisiertes Fahrzeug beispielsweise zwischen dem Tod der Fahrzeuginsassen und dem von Passanten wählen muss [BSR16].

Diese Dilemmata-Situationen sind aktuell noch von geringer praktischer Relevanz, da Szenarien mit geringer Auftretenswahrscheinlichkeit angenommen werden und großes Wissen über alle an dem Szenario und seinem weiteren Verlauf involvierten Parteien unterstellt wird. Heutige Sensorik ist in der Regel nicht in der Lage, die für die beschriebenen moralischen Entscheidungen notwendigen Informationen aus einem realen Szenario zu extrahieren und in Echtzeit zu verarbeiten [EK17, S. 14].

Mit allgemeinen ethischen Fragen im Bereich des automatisierten Fahrens beschäftigte sich die vom Verkehrsministerium der Bundesrepublik Deutschland eingesetzte Ethik-Kommission [EK17]. Im Abschlussbericht wird als erstes und höchstes Ziel des automatisierten Fahrens die Verbesserung der Verkehrssicherheit definiert. Gleichzeitig wird aber davon ausgegangen, dass mindestens zu Beginn der Einführung von AF-Funktionen (SAE-Level 3 bis 5) keine vollständige Unfallvermeidung möglich ist. Als Voraussetzung für die Zulassung automatisierter Fahrfunktionen wird daher keine absolute Sicherheit gefordert, sondern eine positive Risikobilanz. Präzisiert wird dies in [Hom05]: das durch automatisiertes Fahren entstehende Risiko muss nicht nur gesamtgesellschaftlich geringer sein als das durch manuelles Fahren verursachte (utilitaristische Betrachtung), sondern es muss auch für jeden einzelnen potentiell Betroffenen in einer geringeren Wahrscheinlichkeit einer Unfallbeteiligung resultieren. Insgesamt führt dies zu der Forderung der Ethikkommission: „Automatisierte Fahrfunktionen dürfen nur dann eingesetzt werden, wenn sie statistisch sicherer sind als menschliche Fahrer.“ [EK17, S. 26].

Sowohl für die Erfüllung der rechtlichen als auch der ethischen Anforderungen ist es somit unerlässlich, vor Einführung des automatisierten Fahrens zu einem Absicherungsergebnis mit belastbarer quantitativer Aussage zu gelangen. Ansätze zur Lösung und daraus resultierende Schwierigkeiten werden im Kapitel 2 dargestellt. In Kapitel 3 wird daraufhin das Konzept des in dieser Arbeit verfolgten Absicherungsansatzes „passives HAF“ vorgestellt. Bei diesem läuft

die HAF-Funktion ohne Zugriff auf Aktorik im Fahrzeug mit. Stattdessen wird bei Abweichung der Fahrentscheidung der HAF-Funktion von der des menschlichen Fahrers mit Hilfe der präsentierten Methode simuliert, welchen weiteren Verlauf das Szenario genommen hätte. Zur realistischen Simulation des das HAF-Fahrzeug umgebenden Verkehrs werden in Kapitel 4 mögliche Fahrermodelle beschrieben und zur Detektion von kritischen Situationen der in Kapitel 5 vorgestellte Kritikalitätsalgorithmus genutzt. In Kapitel 6 werden die Ergebnisse der präsentierten Methoden evaluiert. Die Arbeit schließt mit Kapitel 7, in dem die Vorteile und Schwächen der Methode zusammengefasst werden.



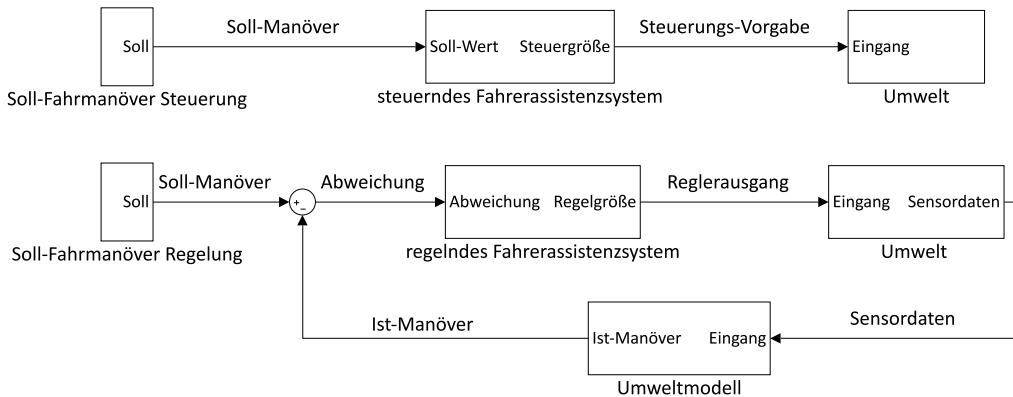
## 2 Stand der Forschung und Technik: Aktuelle Absicherungsmethoden

Außerhalb von Fachmedien werden vor allem die rechtlichen und ethischen Anforderungen bei der Einführung des automatisierten Fahrens diskutiert (z.B. [Köp18], [Hom18]). Weniger bekannt ist, dass auch für den technischen Nachweis der Sicherheit des automatisierten Fahrens weiter nach Lösungen gesucht wird [WW13].

Auf den ersten Blick wirkt das automatisierte Fahren der SAE-Level 3 bis 5 wie eine kontinuierliche Weiterentwicklung aktueller Fahrerassistenzsysteme (FAS), wie sie in Oberklasse- und auch zunehmend Mittelklasse-Fahrzeugen zum Einsatz kommen.

Hierzu gehören einerseits den Fahrer warnende Systeme wie Abbiegewarner, Auffahrtswarner, Spurwechselwarner, Spurverlassenswarner oder Vorfahrtswarner. Bei diesen Systemen wird mit Hilfe einer Sensorik detektiert, dass der Fahrer die Möglichkeit hat oder bereits in Begriff ist eine Fahraktion durchzuführen, die potentiell zu einem Unfall führen könnte. Hierauf wird der Fahrer durch optische, akustische oder sensomotorische Meldungen aufmerksam gemacht. Zum anderen befinden sich bereits Systeme im Markt, die in die Fahraktionen des menschlichen Fahrers korrigierend eingreifen. So können sensorunterstützte Bremssysteme bei Detektion eines schnell abnehmenden Abstands zum Vorderfahrzeug zusätzlich zur Fahrerwarnung bremsen, um einen Auffahrunfall zu verhindern. Weitere Systeme lenken bei Erkennung eines wahrscheinlich unbeabsichtigten Spurwechsels gegen, um das Fahrzeug in der aktuellen Spur zu halten. Diesen Systemen ist gemein, dass sie jeweils in speziellen Situationen auf eine steuernde Art und Weise reagieren, d.h. sie greifen nicht kontinuierlich in die Fahrzeugführung ein (vgl. Abbildung 2.1, oben). Dies ermöglicht eine Open-Loop Absicherung der Systeme, da für die Bewertung ihrer Sicherheit der weitere Verlauf des Szenarios nicht relevant ist, sondern nur die bis zum Eingriffszeitpunkt vorliegenden Informationen. Beispielsweise wird zur Absicherung der Auffahrwarnung mit Bremsen in [Win01] vorgeschlagen, die Bremsfunktion passiv, d.h. ohne tatsächlichen Zugriff auf die Bremsaktorik, mitlaufen zu lassen und im Nachhinein die Angemessenheit der Bremseingriffe zu bewerten. Für die Bewertung liegen zum Zeitpunkt des potentiellen Eingriffs alle notwendigen Informationen vor. Eine Bremsung ist entweder notwendig, um eine Kollision zu verhindern oder ihre Folgen abzumindern, oder nicht notwendig und somit unangemessen. Die Bewertungsfrage ist somit unabhängig vom weiteren Verlauf des Szenarios, sodass einer Open-Loop Absicherung nichts entgegensteht.

Andere derzeit in Serienfahrzeugen vorkommende FAS unterstützen den Fahrer bei der Wahrnehmung der Fahraufgabe in Spezialsituationen. Besonders verbreitet sind hierbei Park- und Stauassistenten. Während Parkassistenten die Längs- und Querverführung im stillstandsnahen Bereich unter  $v = 10 \text{ km h}^{-1}$  für die Dauer des Einparkens vollständig übernehmen (vgl. Art. 5.1.6.1 in [Wir16]) geschieht dies beim Stauassistenten auf vorher festgelegten Strecken in einem limitierten Geschwindigkeitsbereich zwischen üblicherweise  $v_{\min} = 0 \text{ km h}^{-1}$  bis



**Abbildung 2.1:** Steuerndes (oben) und regelndes (unten) Assistenzsystem. Während das regelnde System auf Veränderungen in der Umwelt reagiert, beschränkt sich die Aufgabe des steuernden Systems auf die Vorgabe eines vorab definierten Soll-Verhaltens

$v_{\max} = 60 \text{ km h}^{-1}$ . In beiden Fällen ist der Fahrer verpflichtet, die Aktionen des Fahrzeugs zu überwachen und bei Bedarf einzugreifen. Um die Aufmerksamkeit des Fahrers sicherzustellen muss der Fahrer dabei ein Bedienelement kontinuierlich oder regelmäßig betätigen (z.B. Schalter) oder berühren (z.B. Lenkrad).

Da bei diesen kontinuierlich regelnden Funktionen (vgl. Abbildung 2.1, unten) eine Absicherung wie bei den oben beschriebenen steuernden Funktionen nicht möglich ist, wird der Einsatzbereich der Funktion auf Szenarien von geringer Komplexität oder Schadenswahrscheinlichkeit eingeschränkt und die kontinuierliche Überwachung durch den Fahrer im Sicherheitskonzept vorausgesetzt [LFSR15, S. 1003].

Daraus folgt ein qualitativer Unterschied in der Absicherung von hochautomatisierten Fahrfunktionen gegenüber den bisher beschriebenen. Gründe hierfür sind die aufgrund des weniger stark limitierten Einsatzbereiches erhöhte Komplexität der automatisierten Fahrfunktionen der SAE-Level 3 bis 5 gegenüber bisherigen Funktionen, die Auswertung von Sensordaten durch maschinell trainierte Modelle sowie die fehlende Überwachung durch den menschlichen Fahrer [WW13].

Maschinell trainierte Modelle sind in der Regel für Menschen nicht nachvollziehbar und analysierbar. Der menschliche Fahrer konnte bisher bei Systemversagen als Rückfallebene genutzt werden, wodurch die Zuverlässigkeitsanforderungen an das System selbst geringer waren. Bei automatisierten Fahrfunktionen der SAE-Level 3 bis 5 ist weder eine Absicherung wie bei steuernden Funktionen möglich, noch eine Fahrerüberwachung als Rückfallebene wie bei Park- oder Stauassistenten gegeben, die im Sicherheitskonzept genutzt werden kann. Die einhergehenden Funktionseinschränkungen widersprechen dem Funktionsziel des automatisierten Fahrens ohne Fahreraufmerksamkeit. Daher wird im Folgenden dargelegt, welche Absicherungsmethoden für hochautomatisiertes Fahren stattdessen geeignet sein könnten. Ziel der Absicherung ist dabei einerseits die Verifikation und andererseits die Validierung der automatisierten Fahrfunktion, welche nach [Int11, Band 1] wie folgt definiert sind:



*Verifikation* bezeichnet die Prüfung, ob ein untersuchtes System seine Spezifikation erfüllt.

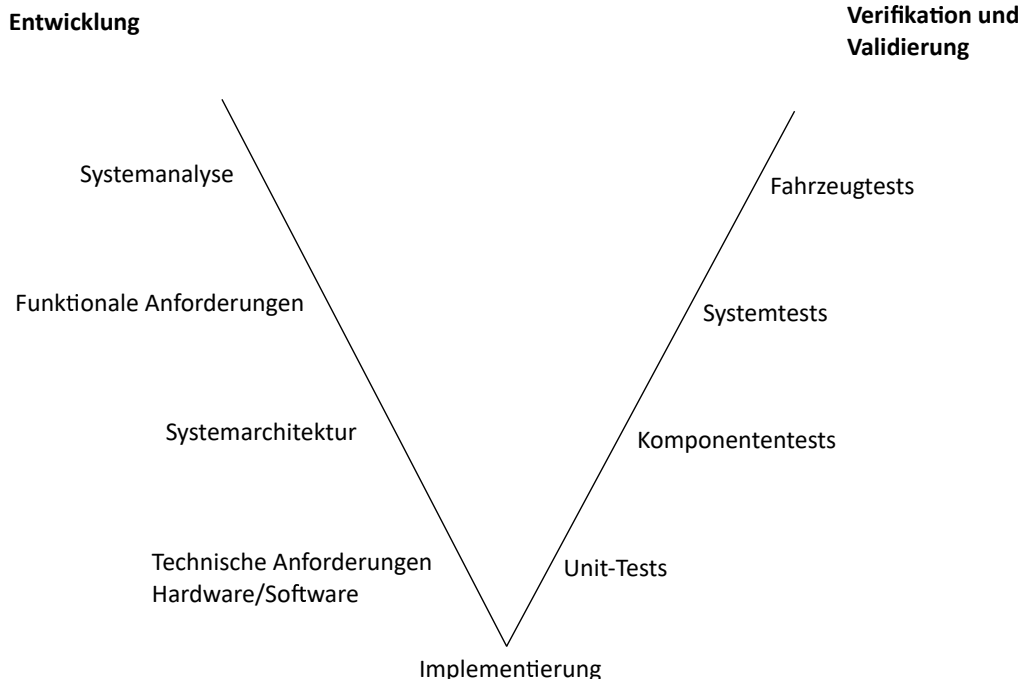
Im Zuge der Entwicklung einer automatisierten Fahrfunktion wird im Rahmen der Verifikation überprüft, ob alle Anforderungen an die Funktion und ihre Komponenten wie Sensoren, Aktoren und Steuergeräte mit der nötigen Integrität umgesetzt wurden.

*Validierung* bezeichnet die Prüfung, ob die Spezifikation eines untersuchten Systems für den gewünschten Einsatzzweck geeignet ist.

Im Kontext der Erstellung eines Sicherheitsnachweises für automatisiertes Fahren wird im Rahmen der Validierung die Frage beantwortet, ob die spezifizierten Anforderungen hinreichend sind um zu verhindern, dass durch die Inverkehrbringung der automatisierten Fahrfunktion intolerable Risiken für Passagiere und Gesellschaft entstehen.

## 2.1 Anforderungsbasierte Absicherung

Eine weitverbreitete Entwicklungs- und Absicherungstechnik für bisherige FAS-Funktionen orientiert sich an dem V-Modell [Int11], welches aus einer linken Entwicklungs- und einer rechten Absicherungsseite besteht. Die linke Seite des V-Modells entsteht durch eine von oben nach



**Abbildung 2.2:** V-Modell der sicherheitsgerichteten Entwicklung (nach [Int11])

unten detailliertere Spezifikation: Links oben erfolgt die Spezifikation einer Funktion anhand vorher definierter Anwendungsfälle und einer Systemanalyse am Anfang der Entwicklung. Diese wird während der Entwicklung soweit erforderlich angepasst und verfeinert. Hiervon

werden wiederum Anforderungen an die Wirkkette und einzelne Komponenten innerhalb der Wirkkette abgeleitet. Ganz unten links erfolgt die Spezifikation von Anforderungen an Hardware- und Software- Komponenten.

Auf der rechten Seite des V-Models werden Verifikations- und Validierungstests definiert. Durch die Verifikationstests wird überprüft, ob die Anforderungen der linken Seite korrekt umgesetzt wurden. Die Validierungstests prüfen, ob die richtigen Anforderungen gestellt wurden und somit die erwarteten Ziele wie beispielsweise Kundennutzen oder eine hinreichende Funktionssicherheit erreicht werden.

Das Resultat ist eine hierarchische Spezifikation, die sowohl analysierbar als auch testbar ist. Zusammen mit den Sicherheitsanalysen, die abhängig von der zu erzielenden Integrität der Funktion durchgeführt werden müssen, erlaubt dies, am Ende des Entwicklungsprozesses eine Sicherheitsargumentation nach Vorgaben der [Int11] zu erstellen. In dieser wird begründet, warum die entwickelte Funktion frei von intolerablen Risiken und eine Zulassung daher möglich ist. Das vollständige V-Modell ist in Abbildung 2.2 dargestellt.

Für die Verifikation des hochautomatisierten Fahrens müsste nach der Methodik des V-Modells allerdings für eine sehr hohe Anzahl an Situationen die spezifikationsgerechte Funktionsweise der HAF-Funktion getestet werden [SSL<sup>+</sup> 13].

Eine Abschätzung der Größenordnung der zu verifizierenden Situationen kann mit Hilfe des Ansatzes zur Situationsbeschreibung in [Adi04, S. 71f] erfolgen. Ein Verkehrsszenario besteht dabei aus einer zeitlichen Abfolge von Situationen, wobei eine Situation als Momentaufnahme eines Verkehrsszenarios definiert ist, die durch die Gesamtheit der Ausprägungen aller Merkmale beschrieben wird. Die Merkmale sind den Kategorien HAF-Manöver, Fahrersituation, Umwelt statisch, Umwelt dynamisch und Fremdobjekte zugeordnet. Sie sind dabei so gewählt, dass ihre Ausprägungen unabhängig voneinander bestimmt werden können. Beispielhaft sind in Tabelle 2.1 für jede Kategorie exemplarische Merkmale angegeben.

**Tabelle 2.1:** Beispielhafte Merkmale der Kategorien zur Situationsbeschreibung

Kategorie	Merkmale
HAF-Manöver	Längsmanöver, Quermanöver
Fahrersituation	Übernahmefähigkeit, Fahreraktion
Umwelt statisch	Straßentyp, Anzahl Spuren, zulässige Fahrstreifennutzung
Umwelt dynamisch	Sichtverhältnis, Geschwindigkeitsbegrenzung, Spuränderung
Fremdobjekte	Position, Geschwindigkeit, Spur

Wenn jedes der in Tabelle 2.1 aufgezählten 10 von Fremdobjekten unabhängige Merkmal nur 3 Ausprägungen hätte und nie mehr als 5 Fremdobjekte in einer Situation auftreten, würde dies zu einer Gesamtzahl an Situationen  $Sit_{ges}$  von

$$Sit_{ges} = 3^{10} * (3^3)^5 = 8,47 \cdot 10^{11} \quad (2.1)$$

führen. Tatsächlich sind deutlich mehr unabhängige Situationsmerkmale vorstellbar, die berücksichtigt werden müssten. Zudem ist insbesondere bei kontinuierlichen Merkmalen eine Diskretisierung mit einer von der Größe der Diskretisierungsintervalle abhängigen Anzahl an Ausprägungen erforderlich.

Zwar wurden für die Absicherung von (teil-)automatisierten Fahrfunktionen Methoden entwickelt, mit denen die Zahl an Testszenarien reduziert werden kann. Mit diesen Heuristiken werden gezielt Szenarien in Verkehrsdaten gesucht [GBH12] [ZE13] oder simulativ generiert [HPH19], die besonders geeignet sind, Fehler in Fahrfunktionen zu entdecken. Sie bieten jedoch keine Garantie, alle für eine HAF-Funktion potentiell relevanten Verkehrssituationen zu bestimmen. Daher kann dieser Ansatz zwar hinreichend sein für Funktionen, deren Aufgabe die Unterstützung des Fahrers ist, sodass dieser bei fehlerhaften Funktionsverhalten eingreifen kann. Die Absicherungsaufgabe beschränkt sich bei diesen darauf die Funktion so auszulegen und zu implementieren, dass der Fahrer jederzeit in der Lage ist die Funktion zu übersteuern und gefährliche Fahrentscheidungen rechtzeitig zu korrigieren.

Im Fall des hochautomatisierten Fahrens existiert eine solche Rückfallebene regelmäßig jedoch nicht, da die Aufmerksamkeit des Fahrers nicht vorausgesetzt werden kann. Weiterhin ist der quantitative Nachweis einer positiven Risikobilanz der HAF-Funktion zur Erfüllung der rechtlichen und ethischen Anforderungen auf diesem Weg nicht möglich. Daher sind diese Verfahren für die Erbringung eines gültigen Sicherheitsnachweises nicht hinreichend.

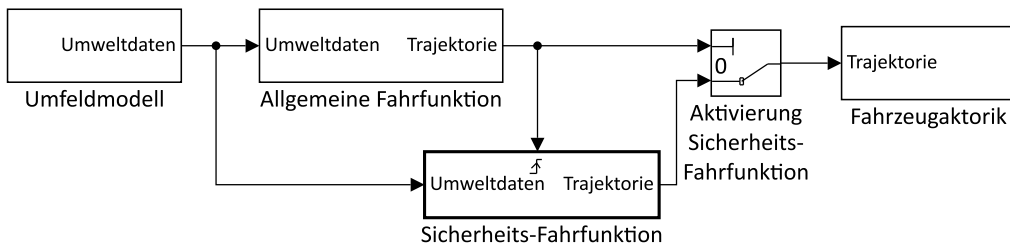
Ein weiterer Hinderungsgrund für die Anwendbarkeit der anforderungsbasierten Absicherung ist die Nutzung von maschinellem Lernen für die Umfelderkennung und daraus folgende Erzeugung eines Umfeldmodells, in dem sich das automatisierte Fahrzeug orientieren kann. Bei den durch maschinelles Lernen trainierten Klassifikatoren handelt es sich in aller Regel um Black-Box Modelle. Der Anforderung-Test-basierte Ansatz des V-Modells innerhalb der durch maschinelles Lernen erzeugten Parameter kann aus diesem Grund nicht angewendet werden. Dabei liegt ein Zielkonflikt zwischen der Verständlichkeit von maschinell gelernten Modellen einerseits und ihrer Güte andererseits vor [HWL17].

Für automatisierte Fahrfunktionen der Level 3-5 werden zudem wegen der Dauer ihres Einsatzes, der schwere von Personenschäden im Fehlerfall sowie der geringen Kontrollierbarkeit aufgrund der fehlenden Fahrerüberwachung in der Regel nach [Int11] Schutzziele ermittelt, die mit einer höheren Integrität abgesichert werden müssen, als durch normale Qualitätsmanagement-Maßnahmen erreichbar ist. Zur Erreichung der notwendigen Integrität werden in der Norm [Int11] Analyse- und Reviewmethoden sehr empfohlen, die nach [SQC18] und [HBE18] für maschinelles Lernen zum Teil nicht durchführbar sind.

In [TFP<sup>+</sup>19] wird ein Ansatz präsentiert, der die beiden vorher genannten Herausforderungen durch eine inverse Absicherungslogik zu lösen versucht. Dabei sucht der entwickelte Algorithmus je nach Absicherungsebene nach einer Situation oder einem Szenario, durch welches vorab spezifizierte Anforderungen verletzt werden. Es handelt sich daher um eine Falsifikation statt einer Verifikation, d.h. das System wird als sicher betrachtet, wenn kein unsicheres Szenario gefunden werden kann. Die Autoren sehen eine frühe Entwicklungsphase als Anwendungsgebiet dieses Ansatzes. Für eine darüber hinausgehende Absicherung wären sowohl für den Szenarien-Suchalgorithmus als auch für die Sensormodelle, die für den Test von maschinell gelernten Klassifikatoren genutzt werden, eigene Absicherungsmethoden erforderlich, um einen durchgehenden Sicherheitsnachweis erbringen zu können.

## 2.2 Verifikation

Eine Methode zum nicht-empirischen Nachweis der Sicherheit des automatisierten Fahrens besteht darin, alle vorstellbaren Szenarien zu abstrahieren und somit ihre Anzahl zu senken. Für die abstrahierten Szenarien wird daraufhin eine Fahrfunktion entwickelt deren Komplexität niedrig genug ist, um sie mit etablierten Methoden zu verifizieren und validieren. Zur Umsetzung dieses Ansatzes erfolgt eine zweistufige Trajektorienplanung mit einer allgemeinen Fahrfunktion und einer Sicherheits-Fahrfunktion wie in Abbildung 2.3 dargestellt. Dabei muss



**Abbildung 2.3:** Onlineverifikation: Die Sicherheits-Fahrfunktion überprüft die von der allgemeinen Fahrfunktion vorgegebene Trajektorie und ersetzt diese, falls sie außerhalb des als sicher erkannten Gebiets liegt

die allgemeine Fahrfunktion nicht verifizierbar und validierbar sein und kann somit beliebig komplex spezifiziert werden. Dadurch können neben einer sicheren Fahrweise verschiedene andere Ziele wie Komfort und Fahrzeit mit einbezogen werden. Dies ist möglich, da in einer zweiten Stufe die Sicherheitsfunktion, welche den oben genannten Anforderungen nach Verifizierbarkeit entspricht, die von der allgemeinen Fahrfunktion vorgegebenen Trajektorien auf ihre Sicherheit hin überprüft.

Ein Konzept hierzu wird in [AD14] vorgeschlagen. Hierbei wird zunächst Kenntnis der maximalen Mess- und Stellfehler von Sensoren und Aktoren vorausgesetzt. Ziel des Konzepts ist es, das maximal erreichbare Gebiet jedes Verkehrsteilnehmers im Szenario zu berechnen. Hierzu wird die Bewegungsrichtung des HAF-Fahrzeugs sowie die Richtung der anderen Verkehrsteilnehmer genutzt und mögliche Mess- und Stellfehler mit einberechnet. Diese erreichbaren Gebiete werden auf Schnittmengen überprüft. Existieren keine, bedeutet dies, dass keine Kollision einzelner Verkehrsteilnehmer in dem Szenario stattfinden kann und die vom HAF-Fahrzeug gewählte Trajektorie somit sicher ist. Ansonsten wird die letzte als sicher erkannte Trajektorie ausgeführt und neue verworfen.

Um die möglichen sicheren Trajektorien nicht zu stark einzuschränken wird die Größe des maximal erreichbaren Gebietes begrenzt, indem im Allgemeinen straßenverkehrsordnungskonformes Verhalten aller Verkehrsteilnehmer angenommen wird. Erst wenn ein Verstoß gegen eine Verkehrsregel durch einen Verkehrsteilnehmer beobachtet wird, wird diese Regel auch nicht mehr zu Einschränkung seines erreichbaren Gebietes genutzt.

Durch die Sicherheits-Fahrfunktion wird gewährleistet, dass das damit ausgestattete automatisierte Fahrzeug keine Trajektorien wählt, durch die es zu einem Unfall kommen kann. Da eine zur Sicherheits-Fahrfunktion äquivalente Maßnahme für das Umfeldmodell nicht möglich ist, werden in diesem Ansatz Kenntnisse über Sensorfehler zur Abschätzung der möglichen Fahr Szenarien genutzt, um dennoch Sicherheit garantieren zu können.

In [SSSS17] wird ein ähnlicher Ansatz verfolgt, wobei statt dem maximal erreichbaren Gebiet der Verkehrsteilnehmer eine semantische Repräsentation möglicher Manöver genutzt wird, welche wiederum auf mögliche Kollisionen überprüft werden. Zusätzlich wird hier der Fokus stark darauf gelegt sicherzustellen, dass das automatisierte Fahrzeug niemals die Schuld an einem Unfall trägt, was bei einem vollständig automatisierten Verkehr zu Unfallfreiheit führen würde.

Beiden Ansätzen ist gemein, dass sie von Annahmen über das Verhalten anderer Verkehrsteilnehmer ausgehen, die vorher spezifiziert werden müssen. Somit ist neben der an sich schon anspruchsvollen Absicherung der Sicherheitsfunktionen zusätzlich der Nachweis der Vollständigkeit dieser Spezifikation notwendig. Weiterhin ist das Ziel beider Ansätze, innerhalb der definierten Spezifikation eine Fahrweise zu wählen, durch die keine Verantwortung für Unfälle auf das automatisierte Fahrzeug fällt. Hierdurch kann allerdings nicht garantiert werden, dass nicht dennoch durch die Fahrweise mehr Unfälle hervorgerufen werden. So könnte nach [WRM<sup>+</sup>19, S. 55] eine besonders defensive Fahrweise menschliche Verkehrsteilnehmer verwirren und zu sicherheitskritischen Ereignissen führen. Da aus ethischer Sicht die Einführung des automatisierten Fahrens aber nur bei insgesamt positiver Risikobilanz vertretbar ist (vgl. Abschnitt 1.2.2), bleibt auch hierfür ein Nachweis erforderlich.

## 2.3 Statistische Absicherung

Ein weiterer Ansatz zur Absicherung des automatisierten Fahrens besteht darin zu zeigen, dass die Zahl der Unfälle statistisch niedrig ist. Insbesondere muss automatisiertes Fahren die in den Abschnitten 1.2.1 und 1.2.2 beschriebenen Anforderungen hinsichtlich hinreichender Sicherheit und positiver Risikobilanz gegenüber nicht-automatisiertem Fahren erfüllen. Möglichkeiten zur Erbringung eines statistischen Sicherheitsnachweises werden im Folgenden vorgestellt.

### 2.3.1 Testfahrten

Testfahrten mit automatisierten Fahrzeugen, die von Testfahrern überwacht werden, sind die direkteste Möglichkeit zur Erbringung eines statistischen Sicherheitsnachweises.

Eine mögliche Herleitung zur Bestimmung der erforderlichen Absicherungstiefe, insbesondere der zu fahrenden Testkilometer, wird in [Wac17, S. 66f] vorgeschlagen. Dabei werden Unfallereignisse als Produkt eines unabhängigen, unerschöpflichen Zufallsprozesses definiert, wodurch eine Modellierung als Poisson-Verteilung möglich wird. Die Poisson-Verteilung ist gegeben durch

$$P_{\lambda}(k) = \frac{\lambda^k}{k!} e^{-\lambda} \quad (2.2)$$

mit  $k$  als Anzahl der Ereignisse und  $\lambda$  als der durchschnittlichen Anzahl an Ereignissen in einem definierten Intervall. Im Fall von Unfallereignissen wird dabei die durchschnittliche

Distanz zwischen zwei Unfällen zur Definition einer Sicherheitsleistung („Safety Performance“, [Wac17, S. 73f]) herangezogen:

$$SP = \frac{\text{Fahrstrecke pro Jahr}}{\text{Anzahl Unfälle pro Jahr}}. \quad (2.3)$$

Für Deutschland werden diese Zahlen jährlich vom Bundesverkehrsministerium veröffentlicht [Bun17a] und sind für Autobahnen als dem für die in dieser Arbeit betrachtete Funktion relevanten Straßentyp für das Jahr 2016 in Tabelle 2.2 dargestellt.

**Tabelle 2.2:** Daten für Sicherheitsleistung auf Autobahnen [Bun17a, S. 106, S. 163]

Unfälle mit Personenschaden	20113
Unfälle mit Todesfolge	414
Auf Autobahnen gefahrene Kilometer	$236,4 \cdot 10^9$ km

Dies führt zu einer Sicherheitsleistung menschlichen Fahrens auf Autobahnen von  $SP_{inj} \approx 12 \cdot 10^6$  km für Personenschäden und  $SP_{fat} \approx 571 \cdot 10^6$  km für tödliche Unfälle. Die zu erwartende Anzahl an Unfällen pro gefahrener Strecke  $d_{test}$  ergibt sich damit zu

$$\lambda = \frac{d_{test}}{SP}. \quad (2.4)$$

Basierend auf dieser theoretischen Grundlage wird in [WW15a] dargestellt, dass eine Zahl an Testkilometern in der Größenordnung von hunderten Millionen bis mehreren Milliarden erforderlich ist, um mit hoher Wahrscheinlichkeit die höhere Sicherheit des automatisierten Fahrens gegenüber menschlichen Fahrern nachweisen zu können. In einer Modellrechnung zeigen die Autoren, dass für die Absicherung einer AF-Funktion die zehnfache Strecke notwendig ist, die durchschnittlich zwischen zwei Unfällen liegt. Der Rechnung liegen die Annahmen zugrunde, dass die AF-Funktion doppelt so sicher ist wie ein durchschnittlicher menschlicher Fahrer ( $\lambda_{HAF} = \frac{1}{2} \lambda_{Mensch}$ ), die Irrtumswahrscheinlichkeit für eine unberechtigte Freigabe unter  $e_{Irrtum} < 5\%$  und die Wahrscheinlichkeit für einen erfolgreichen Nachweis  $e_{Erfolg} > 50\%$  liegen soll [WW15a, S. 455ff].

Hieraus folgt, dass für den Nachweis einer geringeren Anzahl an Verletzten durch automatisierte Fahrzeuge  $d_{test, inj} = 0,12 \cdot 10^9$  km Testkilometer notwendig sind. Für den Nachweis einer geringeren Anzahl an Unfalltoten sind  $d_{test, fat} = 5,71 \cdot 10^9$  km Testkilometer nötig. Da es keine Garantie gibt, dass eine Reduzierung von Unfällen mit Personenschäden auch zu einer Reduzierung von tödlichen Unfällen führt, muss die Überlegenheit für beide Unfallfolgen demonstriert werden.

Aus diesem Grund ist ein statistischer Nachweis der Sicherheit des automatisierten Fahrens durch Testfahrten weder ökonomisch noch vom Zeitaufwand her leistbar. Bei angenommenen Kosten von 2,65 Euro pro Testkilometer wie in [Wac17, S. 178] hergeleitet würde eine Absicherung einer HAF-Funktion über 15,1 Milliarden € kosten, welches knapp dem doppelten des Jahreshaushalts der Stadt München im Jahr 2019 entspricht [SM19]. Um die Absicherung innerhalb eines Monats durchführen zu können, wären über 228000 Testfahrer notwendig, die im Monat jeweils 25 000 km fahren. Sollten während der Absicherung Fehler erkannt werden

oder die Funktion aus anderem Grund später geändert werden müssen, wäre eine erneute Absicherung zu den selben Kosten erforderlich. Die Kosten sind damit so hoch, dass sie weder für einen einzelnen Fahrzeughersteller noch im Rahmen einer herstellerübergreifenden Initiative oder eines staatlich unterstützten Forschungsprojekt tragbar sind.

### 2.3.2 Simulation

Statt der aus oben genannten Gründen undurchführbaren Absicherung durch Testfahrten ist die naheliegende Lösung für die statistische Absicherung des automatisierten Fahrens, die Fahrfunktion in einer Verkehrssimulation auf fehlerhaftes und sicherheitskritisches Verhalten zu testen wie in [HKW<sup>+</sup>17] vorgeschlagen. Voraussetzung für eine Simulation, die reale Tests belastbar ersetzt, sind valide Sensormodelle, eine realistische Modellierung des Verkehrs und repräsentative Simulationsszenarien.

An der Modellierung von Sensoren zur Anwendung im Rahmen der simulativen Verifikation wird weiter geforscht. Für eine Modellierung, die über das Bereitstellen idealer Sensordaten hinausgeht, müssen die Sensoren anhand ihrer physikalischen Wirkungsprinzipien modelliert und evaluiert werden [SZS<sup>+</sup>15]. Allgemein anwendbare Sensormodelle zum Einsatz in der Simulation mit dem Ziel der Validierung hochautomatisierten Fahrens existieren daher aktuell noch nicht.

Für die realistische Modellierung von Verkehrsteilnehmern wurden verschiedene Ansätze entwickelt, die sich hinsichtlich der Art ihrer Modellierung (deterministisch oder probabilistisch) und ihres Parametrierungsansatzes (Parametrierung durch Experten oder datengetriebene Parametrierung durch maschinelles Lernen) unterscheiden lassen. Viele Ansätze beschränken sich dabei auf einen Teilaspekt der Modellierung wie der Modellierung der longitudinalen Fahrdynamik [Wie74][Gip81][THH00], des Abbiegeverhaltens [EWZS13][LRKS13] oder des Spurwechselverhaltens [Gip86][KTH07][CH08].

Insbesondere für die Modellierung des Spurwechselverhaltens können neben den explizit für Simulationsumgebungen entwickelten, in der Regel deterministischen Modellen auch generative Modellierungsansätze der Spurwechselprädiktion genutzt werden. Dabei handelt es sich um Modelle, die einen Spurwechsel anhand der Umgebungsbedingungen eines Fahrzeugs probabilistisch präzisieren. Die Modellierung erfolgt dabei durch Hidden Markov Modelle [TK09], Bayes'sche Netze [KWD<sup>+</sup>12][GBD15][WMT<sup>+</sup>16], Support Vector Machines [KPLL13] oder Neuronale Netze [DEI11].

Während für die auf Grundlage von Expertenwissen parametrierten Modelle in der Regel verschiedene Fahrertypen wie „sportlich“ und „konservativ“ berücksichtigt werden, wird mit der Modellierung durch maschinelles Learning das Ziel verfolgt, die Gesamtheit der Fahrer mit einem Modell für die Spurwechselprädiktion möglichst gut abzubilden. Eine Modellierung individueller Fahrertypen findet in den genannten Ansätzen nicht statt.

Manuelle Parametrierung hat den Nachteil, dass alle Fahrertypen, die zum Einsatz in der Verkehrssimulationen zu Verfügung stehen sollen, vorher parametriert sein müssen. Somit ist für den Nachweis der Validität der Simulation auch der Nachweis der Vollständigkeit der Menge an Fahrertypen notwendig.

Bisherige durch maschinelles Lernen erstellte Modelle berücksichtigen individuelle Fahrertypen

nicht, sodass sie für eine realistische fahrerindividuelle Modellierung nicht direkt geeignet sind.

Die Notwendigkeit der Auswahl repräsentativer Simulationsszenarien ergibt sich aus der weiter oben abgeschätzten Anzahl an überhaupt möglichen Situationen. Sollte jede der einzigartigen Situationen simulativ getestet werden und die Testzeit pro Situation  $T_{\text{Test}} = 1 \text{ s}$  betragen, ergäbe sich eine Gesamttestzeit von über 26 000 Jahren. Daraus folgt, dass auch simulativ nur eine begrenzte Menge an Situationen überprüft werden kann, sodass eine Auswahl der zu simulierenden Szenarien getroffen werden muss. Gleichzeitig liegen für die meisten Bestandteile von Verkehrsszenarien keine Wahrscheinlichkeitsverteilungen vor, weder als unabhängige Wahrscheinlichkeitsverteilung noch als bedingte Verteilung, die von den weiteren Bestandteilen des Szenarios abhängen. Die Auswahl repräsentativer Situationen hat jedoch entscheidenden Einfluss auf die Validität des Verifikationsergebnisses. Die Herausforderungen der Absicherung durch gezielte Simulation oder Testfahrt bestehen daher aus drei Fällen:

1. Situationen, die bekannt sind und die von der HAF-Funktion sicher beherrscht werden.
2. Situationen, deren unbekannt sind und die von der HAF-Funktion nicht sicher beherrscht werden.
3. Situationen, die nicht bekannt sind und deren Beherrschbarkeit durch die HAF-Funktion daher ebenfalls unbekannt ist.

Situationen des Typs 1 und 2 könnten über reale oder simulative Tests abgesichert werden. Dabei ist für eine simulative Absicherung neben der Auswahl einer geeigneten Diskretisierung der kontinuierlichen realen Szenarien die Bestimmung valider Sensormodelle notwendig.

Situationen des Typs 3 lassen sich nicht direkt testen, da sie nicht bekannt, nicht vorstellbar, nicht beachtet oder auf andere Art unerwartet sind. Eine zufallsgesteuerte statistische Absicherung, die einen Absicherungsnachweis für Situationen des Typs 3 erbringen könnte, setzt Wissen darüber voraus, wie wahrscheinlich bestimmte Situationen vorkommen. Ohne die vorher beschriebenen gültigen Wahrscheinlichkeitsverteilungen ist aber kein repräsentatives Generieren von Simulationsszenarien möglich. Aus diesem Grund ist es allgemein nicht möglich, aus den Simulationsergebnissen valide Folgerungen zu schließen.

Um das Fehlen valider Modelle zumindest teilweise zu kompensieren, werden in [DLZ08] und [PL16] X-in-the-Loop-Tests vorgeschlagen. Bei diesen Tests werden Hardware- oder Software-Komponenten oder auch ein ganzes Fahrzeug an ein Simulationsmodell angeschlossen. Im Fall von Vehicle-in-the-Loop-Tests kann die reale Umgebung auch durch simulierte Bestandteile angereichert werden, wodurch eine hohe Vielfalt an Testfällen im Vergleich zu einfachen Straßentests darstellbar ist.

Doch auch bei diesem Ansatz bleibt die Schwierigkeit, für die simulierten Anteile gültige Verteilungen zu finden. Ähnlich wie in [KGHL17] und [KWLH18] wird daher in [PSS19] ein Ansatz vorgestellt, mit dem reale Szenarien in ein Simulationsszenario für die Fahrdynamiksimulation „Carmaker“ übertragen werden können. Hierbei werden Längs- und Quermanöver der Verkehrsteilnehmer eines realen Szenarios extrahiert und in ein Simulationsformat überführt. Dies ermöglicht X-in-the-Loop Tests, die auf realen Szenarien basieren. Darüber hinaus findet eine Clusterung der Szenarien in Abhängigkeit der enthaltenen Manöver statt um sie für die Nutzung in Szenarienkatalogen zu erschließen. Dies ermöglicht die spätere gezielte Absicherung ausgewählter Manöver in einer Vielzahl von realistischen Szenarien. Voraussetzungen für



dieses Verfahren ist eine große Anzahl von im realen Verkehr aufgenommenen Szenarien, um die Repräsentativität der Testszenarien zu gewährleisten.

### 2.3.3 Graduelle Einführung des automatisierten Fahrens

Da der statistische Nachweis der Über- bzw. auch Unterlegenheit des automatisierten Fahrens im Allgemeinen weder durch Testfahrten noch durch Simulation durchführbar ist, wird in [Wac17, S. 96f] ein anderer Ansatz zum Nachweis der Betriebsbewährtheit vorgeschlagen. Zunächst wird zwischen den Sicherheitsanforderungen von zwei Gruppen unterschieden: der Gesellschaft im Allgemeinen ( $\mathcal{G}$ ) und Fahrern automatisierter Fahrzeuge im Besonderen ( $\mathcal{F}$ ). Für  $\mathcal{G}$  ist das Risiko einer höheren Anzahl an Unfällen durch automatisiertes Fahren generell nicht akzeptabel. Als Einschränkung dieser generellen Regel nennt [Wac17], dass ein geringes zusätzliches Risiko in der Größenordnung der statistischen Schwankungen um den allgemeinen Trend der Unfallstatistik konsensfähig sein könnte. Für  $\mathcal{F}$  andererseits könnte ein höheres Risiko akzeptabel sein soweit dieses bekannt und transparent ist, da sich  $\mathcal{F}$  im Gegensatz zu  $\mathcal{G}$  bewusst zur Nutzung von HAF-Fahrzeugen entscheidet und auch davon profitiert.

Zwei Grundideen zeichnen den Ansatz aus: Zum einen wird in einem ersten Schritt kein endgültiger statistischer Nachweis der Überlegenheit des HAF angestrebt, sondern stattdessen ein Bereich definiert, dessen Grenzen die best- und schlechtestmöglich erwartbaren Sicherheitsleistungen der Funktion markieren. Zur Bestimmung dieser Grenzen ist eine gegenüber einer endgültigen Absicherung deutlich geringere Anzahl an Testkilometern  $d_{\text{test}}^{\text{initial}}$  notwendig, da  $d_{\text{test}}^{\text{initial}}$  nur so hoch sein muss, dass eine minimale Sicherheitsleistung statistisch mit hoher Konfidenz bestimmt werden kann. Hierfür wird eine Irrtumswahrscheinlichkeit von ( $\epsilon_{\text{min}} < 5\%$ ) gefordert, d.h. nur mit einer Wahrscheinlichkeit von 5% darf die Sicherheitsleistung tatsächlich geringer sein.

Dies nutzend wird zweitens die Idee verfolgt, automatisierte Fahrzeuge schrittweise in zunächst limitierter Anzahl einzuführen. Vorteil dieser Herangehensweise ist es, dass das Risiko für Personen der Gruppe  $\mathcal{G}$  gering ist, sich in der Nähe eines automatisierten Fahrzeugs und dessen potentiellen Gefahrenbereichs zu befinden. Andererseits können die bereits im Markt befindlichen von den Mitgliedern der Gruppe  $\mathcal{F}$  gefahrenen HAF-Fahrzeuge als Testfahrzeuge genutzt werden, über deren Testkilometer Erkenntnisse über die Sicherheit der Fahrfunktion gesammelt werden können.

Aufgrund der vorangegangenen professionellen Testfahrten der Länge  $d_{\text{test}}^{\text{initial}}$  kann das durch die Einführung des automatisierten Fahrens hervorgerufene Risiko abgeschätzt werden und liegt maximal bei der im vorherigen Absatz beschriebenen minimalen Sicherheitsleistung. Ist diese hinreichend hoch und somit für beiden Gruppen  $\mathcal{F}$  und  $\mathcal{G}$  akzeptabel, steigt im Rahmen der schrittweisen Einführung die Anzahl der Testkilometer  $d_{\text{test}}$ , sodass auch der Bereich zwischen oberer und unterer Grenze der Sicherheitsleistung kleiner wird und die Grenzen letztendlich zur wahren Sicherheitsleistung hin konvergieren.

Zu Beginn dieses Prozesses liegt die minimale Sicherheitsleistung nach den Testfahrten mit hoher Wahrscheinlichkeit<sup>1</sup> bei einem Wert  $SP_{\text{min}}$ , der unterhalb der Leistung der durchschnitt-

<sup>1</sup> Liegt die minimale Sicherheitsleistung von vornherein oberhalb der menschlicher Fahrer, ist der Nachweis der Überlegenheit des automatisierten Fahrens direkt erbracht. Dass dies bei geringer Testkilometer-Anzahl unwahrscheinlich ist, wurde in Unterabschnitt 2.3.1 beschrieben.

lichen menschlichen Fahrer  $SP_{inj}$  und  $SP_{fat}$  liegt. Gleichzeitig liegt die maximale Sicherheitsleistung des automatisierten Fahrens  $SP_{max}$  oberhalb dieser Grenzen<sup>2</sup>.

Übersteigt die untere Grenze  $SP_{min}$  mit zunehmender Anzahl an Testkilometern die durchschnittliche Sicherheitsleistung der menschlichen Fahrer, ist der Nachweis einer Überlegenheit der HAF-Funktion erbracht. Sinkt andererseits die obere Grenze  $SP_{max}$  unterhalb  $SP_{inj}$  bzw.  $SP_{fat}$ , ist der Nachweis der Unterlegenheit erfolgt.

Bewährt sich nun im Zuge der zunehmenden Zahl an Testkilometern die HAF-Funktion und steigt somit die minimale Sicherheitsleistungs-Grenze, so können mehr und mehr automatisierte Fahrzeuge in den Verkehr gebracht werden, ohne das vorher von der Gesellschaft gesetzte maximal akzeptierte Risiko zu erhöhen, sodass wiederum auch die Anzahl an Testkilometern schneller steigt.

Dabei wird in [Wac17, S. 105f] eine übliche Geschwindigkeit, mit der neue Technologien den Markt durchdringen, von vergangenen Automobil-Technologien wie ESP abgeleitet, um zu bestimmen, ob eine graduelle Einführung des automatisierten Fahrens eine deutliche Verzögerung gegenüber einer unreglementierten Einführung zur Folge hätte. Der Autor gelangt zu dem Schluss, dass zum Vermeiden einer Verzögerung eine nach wie vor hohe Anzahl an Testkilometern in der Größenordnung von  $10^8$  km durch professionelle Testfahrer notwendig ist, um eine hinreichend hohe minimale Sicherheitsleistungsgrenze mit hoher Wahrscheinlichkeit bestimmen zu können. Weiterhin muss eine mindestens kleine Toleranz gegenüber möglichen zusätzlichen Sicherheitsrisiken auf Seiten von  $\mathcal{G}$  gegeben und vor allem die wahre, mit diesem Verfahren zu bestimmende, Sicherheitsleistung der zu validierenden HAF-Funktion hoch sein.

Einschränkungen des in [Wac17] vorgestellten Ansatzes sind, dass eine während der Absicherung unveränderliche HAF-Funktion vorausgesetzt wird. Simulationsergebnisse in [Wac17, S. 122f] zeigen außerdem, dass  $d_{test}$  in der Größenordnung von  $10^8$  km liegen muss, damit die minimale Sicherheitsleistungsgrenze hinreichend hoch und damit sicher ist. Dies entspräche Absicherungskosten in Höhe von 265 Millionen Euro und läge damit weiterhin außerhalb eines für einen einzelnen Hersteller tragbaren Rahmens. Durch eine Entwicklung im Rahmen eines Herstellerkonsortiums oder eines öffentlich geförderten Projekts wären die Kosten hingegen im Gegensatz zum Fall einer ausschließlichen Dauerlaufabsicherung möglicherweise tragbar. Bedingung hierfür wäre die Nutzung der gleichen abzusichernden Funktion durch alle Hersteller, sodass keine Möglichkeit der Wettbewerbsdifferenzierung mehr bestünde.

## 2.4 Bewertung der Absicherungsansätze

Die beschriebenen Ansätze zur Absicherung des automatisierten Fahrens zeichnen sich durch unterschiedliche Stärken und Schwächen aus.

Die Entwicklung und Absicherung nach dem V-Modell ist die aktuell in der Automobilindustrie gängigste Methode, während der Ansatz zur Onlineverifikation gezielt für das automatisierte Fahren entwickelt wurde. Beiden gemein ist die Notwendigkeit einer detaillierten Spezifikation.

<sup>2</sup> Liegt die maximale Sicherheitsleistung unterhalb der menschlicher Fahrer, ist der Nachweis der Unterlegenheit direkt erbracht.

Kennzeichnend für den Sicherheitsnachweis über Simulation ist eine hohe Effizienz durch Automatisierung und Nutzung von Rechenzeit anstatt menschlichen Testfahrern und realen Fahrzeugen. Hindernisse sind die unbekannte Modellqualität von Sensoren und Verkehr sowie das Fehlen von Wahrscheinlichkeitsverteilungen zur validen Szenariogenerierung.

Die theoretischen Überlegungen des Ansatzes der graduellen Einführung sind plausibel und zeigen einen Weg aus dem „Absicherungsdilemma“ [WW15a] auf. Nachteilig ist die Notwendigkeit einer immer noch hohen Zahl an Testkilometern, die durch professionelle Testfahrer erbracht werden muss.

Zusammenfassend lassen sich folgende Punkte festhalten:

1. Etablierte Absicherungsmethoden wie in Abschnitt 2.1 beschrieben sind für eine Absicherung des HAF nicht hinreichend.
2. Für eine Onlineverifikation nach Abschnitt 2.2 ist noch weiterer Forschungsaufwand notwendig.
3. Nach aktuellem Wissensstand ist daher nur eine statistische Absicherung möglich.
4. Für eine statistische Absicherung durch Simulation (vgl. Abschnitt 2.3.2) müssen die verwendeten Modelle validiert und für realen Verkehr repräsentative Wahrscheinlichkeitsverteilungen ermittelt werden.
5. Für eine statistische Absicherung durch den graduellen Ansatz (vgl. Abschnitt 2.3.3) müssen hinreichend signifikante Fehlergrenzen zur Bestimmung der Sicherheitsleistung ermittelt werden.
6. Für Punkt 4 ist eine hohe Anzahl an Testkilometern notwendig, um die verwendeten Modelle zu validieren und eine repräsentative Datenbasis aufzubauen.
7. Für Punkt 5 ist für jede Version einer HAF-Funktion eine vorhergehende Absicherung mit Hilfe von Testfahrern zur Bestimmung der Fehlergrenzen notwendig.

In den nächsten Kapiteln wird daher eine Methode vorgestellt und evaluiert, die eine effiziente Lösung für die in den Punkten 6 und 7 aufgezeigten Herausforderungen bietet und darüber hinaus eine eigenständige Möglichkeit zur Absicherung des HAF darstellt.



## 3 Konzept und Aufbau des passiven HAF

In diesem Kapitel wird das Konzept des passiven HAF, sein Aufbau und seine Implementierung erläutert.

### 3.1 Konzept des passiven HAF

Die Fahrleistung durch PKW betrug im Jahr 2015 in Deutschland  $236 \cdot 10^9$  km [ADA15]. Angesichts des in Kapitel 2 beschriebenen Bedarfs an Testkilometern zur Absicherung des automatisierten Fahrens ist das Ziel dieser Arbeit, eine Methode zu entwickeln mit der ein Teil der ohnehin erbrachten Fahrleistung zur Absicherung des hochautomatisierten Fahrens genutzt werden könnte.

In [WW13] wird hierzu vorgeschlagen, ähnlich wie zur Absicherung des Bremsassistenten (vgl. Kapitel 2) die zu testende HAF-Funktion open-loop mitlaufen zu lassen, da hierdurch eine hohe Kilometerzahl ohne zusätzliche Gefährdung der Fahrzeuginsassen oder anderer Verkehrsteilnehmer erbracht werden kann. Wie im vorherigen Kapitel beschrieben, sind Open-Loop Tests für die Absicherung kontinuierlich regelnder Funktion wenig geeignet, da der weitere Verlauf des realen Szenarios von dem durch den Eingriff des Regelsystems entstehenden hypothetischen Szenarioverlauf abweichen und somit nicht zur Bestimmung der Güte des Regelsystems ausgewertet werden kann.

Ein Ansatz, Open-Loop Tests dennoch zur Absicherung des HAF zu nutzen, basiert darauf, den weiteren Verlauf des hypothetischen Szenarios zu simulieren und auf dieser Grundlage die HAF-Funktion abzusichern, wie in [KGHL17] vorgestellt. Entscheidend für die Aussagefähigkeit dieser Art Test ist der Grad der Übereinstimmung des Simulationsverlaufs mit den tatsächlichen Reaktionen der das HAF-Fahrzeug umgebenden Verkehrsteilnehmer. Daher steht die realitätsnahe Simulation im Fokus des konkreten Konzepts des passiven HAF, welches in dieser Arbeit vorgestellt wird. In eine ähnliche Richtung mit geringerem Fokus auf die Modellierung des umgebenden Verkehrs zielt das in [WW15b] vorgestellte Konzept.

#### 3.1.1 Zentrale Bausteine

Zur Umsetzung des Absicherungsansatzes „Passives HAF“ werden Fahrzeuge mit der für hochautomatisiertes Fahren notwendigen Sensorik wie RADAR, LIDAR und Kamera ausgestattet und Kunden beispielsweise durch Verkauf oder Leasing zu Verfügung gestellt. Die zu diesem Zeitpunkt noch nicht final abgesicherte HAF-Funktion ist zunächst nicht aktivierbar, sondern läuft lediglich im Hintergrund ohne Zugriff auf die Aktorik der Fahrzeuge. Stattdessen werden die aufgenommenen Sensordaten zwar als Eingangsgrößen der HAF-Funktion zur Verfügung

gestellt, der Ausgang der HAF-Funktion jedoch in eine Simulation geführt. In dieser wird der weitere Fortgang des aktuellen Szenarios auf Grundlage der Sensordaten simuliert.

Hierdurch kann einerseits sichergestellt werden, dass fehlerhafte Ausgaben der Funktion nicht zu einer Gefährdung führen. Andererseits wird durch die Simulation der Nachteil einfacher Open-Loop Tests vermieden, welche keine Reaktionen anderer Verkehrsteilnehmer auf die Aktionen der zu testenden Funktion bieten. Da das reale Verkehrsszenario, welches durch die Sensorik erfasst wird, als Grundlage für die Simulation genutzt wird, ist zudem anders als bei herkömmlicher Simulation die Repräsentativität des simulierten Szenarios gewährleistet. Als Informationen aus dem realen Szenario wird dabei die gesamte von der Sensorik erfasste statische Umgebung<sup>3</sup> sowie die Positionen der das HAF-Fahrzeug umgebenden Verkehrsteilnehmer bei Simulationsstart verwendet. Darüber hinaus werden mit Hilfe einer vorab erstellten Fahrermodell-Datenbank den einzelnen Verkehrsteilnehmern des realen Szenarios passende Fahrermodelle zugeordnet und in der Simulation genutzt, wodurch die Validität des simulierten Verhaltens des umgebenden Verkehrs erhöht wird.

Zur Bewertung der Sicherheit des von der HAF-Funktion initiierten Manövers wird das simulierte Szenario herangezogen, in welchem die kurz- und mittelfristigen Auswirkungen des Manövers analysierbar sind. Der gesamte Ablauf ist in Abbildung 3.1 dargestellt. Dabei kann jeder Schritt der Simulation und Auswertung des fiktiven Szenarios mit Ausnahme der Sensordatenaufzeichnung je nach Leistungsfähigkeit der im Fahrzeug genutzten Steuergeräte entweder im Fahrzeug selbst oder zu einem späteren Zeitpunkt fahrzeugunabhängig durchgeführt werden.

Der Vorteil einer Simulation und Auswertung im Fahrzeug besteht darin, dass der Umfang der an ein Backend-System zu übertragenden Daten auf die als kritisch erkannten Szenarien beschränkt werden kann. Vorteilhaft bei einer Übertragung von Sensor- oder Umfeldmodelldaten an ein Backend-System und einer nachgelagerten Simulation und Auswertung ist, dass die übertragene Daten immer wieder neu mit unterschiedlichen und veränderten HAF-Funktionen ausgewertet werden können und zudem eine Nutzung der Daten zum Training weiterer Fahrermodelle möglich ist.

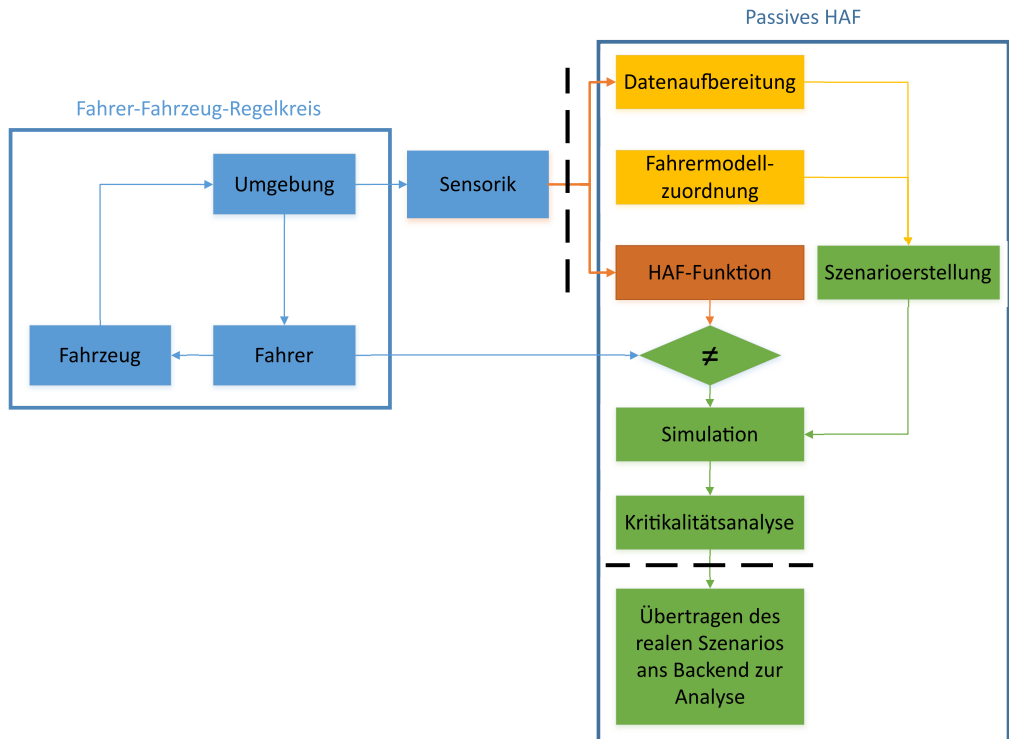
Zum Training neuer Modelle eignen sich die Daten des HAF-Fahrzeugs bei manueller Fahrt am besten, da während der ganzen Fahrt Daten über die Fahrentscheidungen des Fahrers aufgezeichnet wurden. Die so trainierten Fahrermodelle können dann den das HAF-Fahrzeug umgebenden Verkehrsteilnehmer zugeordnet und in der Simulation genutzt werden. In der vorliegenden Arbeit erfolgt daher die Simulation und Auswertung der Daten außerhalb des Fahrzeugs. Bei Einsatz des passiven HAFs durch Kunden wäre aber auch eine stärkere Berücksichtigung des Schutzes ihrer persönlichen Daten möglich, indem Simulation und Auswertung bereits im Fahrzeug durchgeführt werden.

Da die erstellten Simulationsszenarien auf den von der Sensorik zur Verfügung gestellten Daten beruhen, kann eine Absicherung durch das passive HAF auch nur auf dieser Grundlage erfolgen und Fehler in der Umweltwahrnehmung nur teilweise aufgedeckt werden.

Zum Vorteil gereicht dem Absicherungskonzept, dass ein Simulationsszenario nicht sofort

---

<sup>3</sup> Zur statischen Umgebung gehören alle Elemente, die nicht durch das Fahrverhalten von Fahrzeugen beeinflusst werden. Beispiele sind der Straßenverlauf, die Spurzahl oder Straßenschilder.



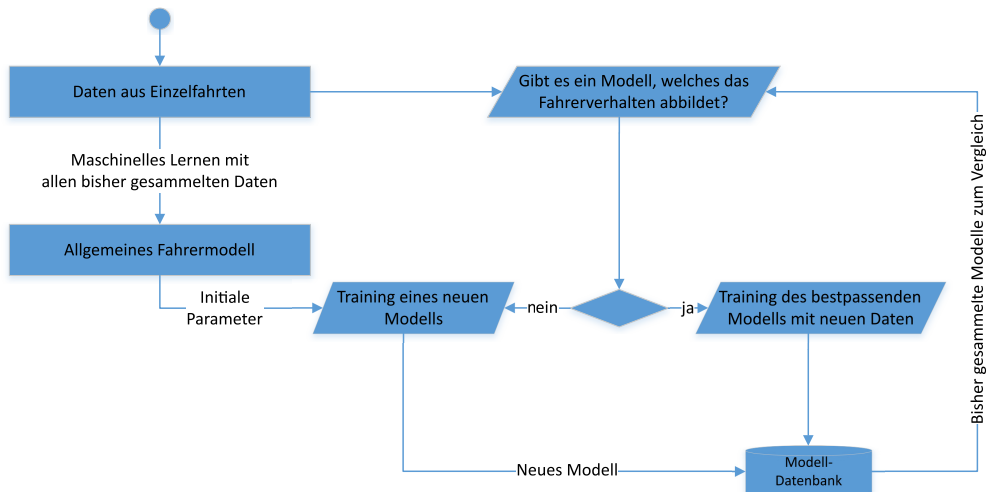
**Abbildung 3.1:** Strukturdiagramm des passiven HAF Konzeptes. Bei Abweichung zwischen HAF und dem Fahrer des Ego-Fahrzeugs wird auf Grundlage der Sensordaten ein Simulationsszenario erstellt. In diesem wird das vom HAF geplante Manöver durchgeführt und auf Kritikalität bewertet. Die gestrichelten Linien geben den frühesten und spätmöglichen Zeitpunkt an, zu dem eine Datenübertragung an ein Backend-System erfolgen kann. Im Backend werden die verbleibenden Konzeptschritte ausgeführt.

erstellt werden muss sondern die Simulation auch zu einem späteren Zeitpunkt gestartet werden kann. Hierdurch ist es möglich, erst zu einem späteren Zeitpunkt detektierte Anteile der statischen Umgebung wie beginnende oder endende Fahrspuren bereits zu einem früheren Zeitpunkt in der Simulation zu berücksichtigen. Im Fall von dynamischen Umgebungsanteilen wie Fahrzeugen ist eine Extra- oder Interpolation ihrer Positionen möglich, falls sie zu Beginn der Realdatenaufzeichnung außerhalb der Sensorreichweite liegen oder zwischenzeitlich von anderen Umgebungsbestandteilen verdeckt sind. Umgebungsbestandteile, die zu keinem Zeitpunkt der Realdatenaufzeichnung von einem Sensor erfasst wurden, können allerdings auf diese Weise nicht in das Simulationsszenario eingebunden werden und bleiben unberücksichtigt. Diese Art von Fehlern der Umfeldwahrnehmung könnte nur durch die Anwendung einer Referenzsensorik gefunden werden.

In wie weit Sensorikbeschränkungen den Realitätsgrad der Simulation im Rahmen des passiven HAF beeinflussen wird im Ergebnisteil dargestellt.

Besondere Bedeutung kommt bei der Simulation der Auswahl der passenden Fahrermodelle zur Simulation der das HAF-Fahrzeug umgebenden Verkehrsteilnehmer zu. Während die

Repräsentativität des simulierten Szenarios dadurch gewährleistet ist, dass es auf dem realen Szenario beruht und dieses als Sample der realen Wahrscheinlichkeitsverteilungen betrachtet werden kann, ist die Validität des Fahrerverhaltens in der Simulation schwieriger nachzuweisen. Daher werden in der vorliegenden Arbeit mit Hilfe von maschinellem Lernen Fahrermodelle trainiert, die das Verhalten aller in realen Verkehrsszenarien auftretenden Fahrertypen mit hoher Güte modellieren. Drei Modellierungsmethoden und Ansätze zur Modellierung von menschlichen Fahrern werden in Kapitel 4 vorgestellt. Zum Training wird ein mehrstufiger Ansatz genutzt (dargestellt in Abbildung 3.2), bei dem in einem ersten Schritt ein allgemeines Fahrermodell unter Nutzung von in Testfahrten aufgezeichneten Daten erstellt wird. Auf diesem allgemeinen Modell aufbauend werden dann individualisierte Fahrermodelle weiter trainiert, um verschiedene Fahrertypen abbilden zu können. Ähnlich wurde in [ACP<sup>+</sup> 18] zur personalisierten Sprachgenerierung vorgegangen, bei dem ein allgemeines neuronales Netz zur Spracherzeugung durch Feinabstimmung individualisiert wurde.



**Abbildung 3.2:** An zentraler Stelle werden die aufgenommenen Daten aus allen Einzelfahrten gesammelt und zum Training der Fahrermodelle genutzt.

In der zweiten Stufe, in der die HAF-Funktion passiv in Kundenfahrzeugen mitläuft, wird jedes sich im Sensorbereich des HAF-Fahrzeugs befindliche Fahrzeug analysiert und das am besten passende Fahrermodell zugeordnet. Stellt sich heraus, dass die Güte des genutzten Fahrermodells trotz der Individualisierung zu gering ist, wird das Szenario abgespeichert und bei nächster Gelegenheit über eine Datenschnittstelle an den Fahrzeughersteller übertragen. Die aufgezeichneten Daten werden genutzt, um auf Grundlage des allgemeinen Fahrermodells ein neues individuelles Fahrermodell zu trainieren. Sofern das neu trainierte Modell das Fahrverhalten in den aufgezeichneten Daten reproduzieren kann, wird die Fahrermodelldatenbank um dieses erweitert. Dies stellt sicher, dass die Fahrermodelle eine hinreichende Güte aufweisen, wenn sie zur Simulation des fiktiven HAF-Szenarios genutzt werden. Eine hohe Validität der Simulation ist wiederum entscheidend für die Bewertung der Sicherheit der HAF-Funktion.



### 3.1.2 Erzielbare Absicherungsleistung des Ansatzes

Ob das Potential des Ansatzes passives HAF genutzt werden kann, hängt entscheidend davon ab, welche Wegstrecke entsprechend ausgestattete Fahrzeuge fahren, wieviele Kilometer davon zur Absicherung der HAF-Funktion genutzt werden können und in welchem Zeitraum somit eine vollständige Absicherung möglich ist.

Nach [ADA15] beträgt die durchschnittliche Fahrleistung eines in Deutschland zugelassenen Autos ca. 14 200 km pro Jahr, wovon ca. 5300 km auf Autobahnen gefahren werden.

Während der Entwicklung des HAF kann angenommen werden, dass regelmäßige Funktionsänderungen und -verbesserungen einfließen, sodass eine periodische Überprüfung der Funktion wünschenswert erscheint. Unter Annahme einer übers Jahr gleichmäßig verteilten Fahrleistung wären zum Erreichen einer Absicherungsleistung von  $10^8$  km innerhalb eines Monats 226 415 mit dem passiven HAF ausgestattete Fahrzeuge notwendig.

Im Jahr 2017 wurden nach [KB18b] 3,44 Mio. PKW neu zugelassen. Dies verdeutlicht, dass die für die Absicherung notwendige Anzahl an Fahrzeugen bereits bei einem geringen Anteil an mit passiven HAF ausgestatteten Fahrzeugen erreicht werden könnte. Unter der Annahme von Teile- und Herstellkosten für HAF-Sensorik und Steuergeräten von 1000 bis 2000 Euro lägen die Kosten für einen solchen Absicherungsansatz bei 227 Millionen bis 454 Millionen Euro.

Wie beim Ansatz der graduellen Einführung des HAF (vgl. Abschnitt 2.3.3) überstiege dies wahrscheinlich die für einen einzelnen Hersteller akzeptablen Kosten, wäre jedoch für einen Herstellerverbund tragbar. Vorteil dieser Lösung wäre zudem, dass eine einheitliche Sensorik-Schnittstelle hinreichend ist und die eigentliche HAF-Funktion herstellerindividuell erstellt werden kann. Nach erfolgreicher Absicherung wäre es zudem möglich, dem Kunden die HAF-Funktion zur Freischaltung anzubieten und somit zumindest einen Teil der vorab investierten Herstellungskosten zu kompensieren.

Zur Veranschaulichung des Potentials diene der Vergleich mit der Absicherung durch professionelle Testfahrer. Unter der Annahme, dass ein Testfahrer durchschnittlich  $100 \text{ km h}^{-1}$  für 10 h pro Tag an 25 Tagen pro Monat fahren kann, könnte er 25 000 km pro Monat an Fahrleistung erbringen. Um mit Testfahrern die gleiche Absicherungsleistung wie durch den Ansatz passives HAF zu erreichen, wären somit 4000 Testfahrer erforderlich (vgl. Tabelle 3.1). Diese Zahl kann durch den beschriebenen Ansatz deutlich reduziert werden, während gleichzeitig die Repräsentativität der für die Absicherung gesammelten Daten durch die höhere Varianz der Fahrer und Strecken erhöht wird.

**Tabelle 3.1:** Absicherungsleistung nach Anzahl der Fahrzeuge und Fahrertyp

Fahrzeuge	Kilometer auf Autobahn pro Jahr	Monate für $10^8$ km Testkilometer	Fahrertyp
1	5300	226415	divers
226415	5300	1	divers
4000	300000	1	Testfahrer

## 3.2 Simulation der Fahrfunktion auf Grundlage realer Verkehrsszenarien

Elementarer Teil des in dieser Arbeit vorgestellten Konzepts „passives HAF“ ist es zu simulieren, wie Szenarien sich weiterentwickeln, wenn statt des menschlichen Fahrers eine HAF-Funktion die Führung des Fahrzeugs übernimmt.

Hierzu muss aus dem realen Szenario ein Simulationsszenario erzeugt werden und den Fahrzeugen des Szenarios Fahrermodelle zugeordnet werden, die das Verhalten ihrer Fahrer widerspiegeln. Dieser Prozess wird im folgenden *Resimulation* genannt. Eine Simulation wird immer dann gestartet, wenn der Ausgang der HAF-Funktion sich vom Verhalten des menschlichen Fahrers signifikant unterscheidet.

Ein Unterschied zwischen dem tatsächlichen Fahrverhalten des menschlichen Fahrers und den nicht real umgesetzten Entscheidungen der HAF-Funktion kann sowohl in longitudinaler Richtung durch unterschiedliches Beschleunigungsverhalten als auch in lateraler Richtung durch unterschiedlichen Lenkwinkel vorliegen. Im Allgemeinen ist davon auszugehen, dass diese kontinuierlichen Stellgrößen nie völlig identisch von Mensch und HAF angefordert werden und zumindest geringfügige Abweichungen vorliegen.

Eine signifikante Abweichung liegt genau dann vor, wenn der Unterschied im Fahrverhalten von Mensch und HAF zu einem Unterschied in der Reaktion des umgebenden Verkehrs führt oder führen könnte, da in Folge dessen eine kritische Situation entstehen könnte. Dies ist immer dann der Fall, wenn sich der für eigene Fahrentscheidungen zur Verfügung stehende Raum anderer Verkehrsteilnehmer durch abweichende Fahrentscheidung des HAF geändert hätte. Offensichtlich ist dies der Fall, wenn eine Abweichung hinsichtlich der Spurwahl vorliegt. In diesem Fall ist ein Wechsel anderer Verkehrsteilnehmer auf die frei gewordene Spur an Stelle des HAF-Fahrzeugs möglich und auf die neu belegte Spur nicht mehr möglich. Gleiches gilt für komplexere Manöver wie Überholvorgänge, Auffahren oder Abfahren, bei denen ein Spurwechsel stets Teil des Manövers ist.

Weniger eindeutig ist die Bestimmung im Fall von Abweichungen im longitudinalen Verhalten. Weicht das Beschleunigungsverhalten von menschlichem Fahrer und HAF voneinander ab, entstehen oder schließen sich Lücken, die von anderen Verkehrsteilnehmern zum Einschleichen genutzt werden könnten. Die Grenze, ab der ein quantitativer Unterschied in der Geschwindigkeit zu einem qualitativ anderen Fahrverhalten eines Verkehrsteilnehmer als im realen Szenario führt, hängt vom jeweiligen Fahrer ab und kann nicht allgemein bestimmt werden. Innerhalb der Systemgrenzen der HAF-Funktion mit einer Geschwindigkeitsgrenze von maximal  $130 \text{ km h}^{-1}$  bzw. dem jeweils gültigen Geschwindigkeitslimit ist allerdings nicht mit Geschwindigkeitsdifferenzen zwischen Fahrer und HAF zu rechnen, die innerhalb kurzer Zeit zu einem veränderten Manöverraum relevanten Umfangs für andere Verkehrsteilnehmer führen.

Daher wird in dieser Arbeit genau dann von einem signifikanten Verhaltensunterschied ausgegangen, wenn sich ein Unterschied im Spurwechselverhalten ergibt. Diese Grenze wurde gewählt, da sich durch einen Spurwechsel ein Szenario qualitativ ändert und gegenüber der Fahrt auf der gleichen Spur neue, potentiell kritische Situationen auftreten können. Unterschiede in komplexen Manövern sind durch diese Grenze ebenfalls mit erfasst. Auf die Bestimmung einer Grenze, ab der Abweichungen im longitudinalen Verhalten zur Resimulation eines Szenari-

os führen, wurde aus oben genannten Gründen in dieser Arbeit verzichtet, sodass Unterschiede hier nicht als Auslösebedingung für die Simulation genutzt werden. Da die weiteren Schritte der Szenariogenerierung, Simulation und Auswertung unabhängig von den Auslösebedingungen für die Simulation sind, wäre eine Erweiterung um diese Bedingung aber möglich.

Nach Start und Ablauf der Simulation wird der Verlauf des simulierten Szenarios bei Umsetzung der Fahrentscheidungen der HAF-Funktion überprüft. Ergebnis des Prozesses ist die Bewertung der Fahrentscheidungen der HAF-Funktion hinsichtlich ihrer Kritikalität für die Sicherheit des eigenen und umgebender Fahrzeuge.

Im Folgenden werden die einzelnen Schritte detaillierter erläutert. Zur Simulation der fiktiven Weiterentwicklung des Verkehrsszenarios unter HAF-Eingriff werden Anforderungen an Simulationsumgebungen formuliert, die zur Nutzung im Rahmen des passiven HAF erfüllt sein müssen. Danach werden mehrere Simulationsumgebungen auf ihre Eignung hin überprüft. In Abschnitt 3.4 wird die Methode vorgestellt, mit der aus mit Sensorik aufgenommenen Daten Simulationsszenarien erzeugt werden. Zuletzt wird in Abschnitt 3.5 dargelegt, wie die passenden Fahrermodelle für die umgebenden Verkehrsteilnehmer bestimmt und in das Simulationsszenario integriert werden.

## **3.3 Simulationsumgebung für das passive HAF**

Der in Abschnitt 3.1 aufgezeigte offene Regelkreis, der durch die Passivität der von der HAF-Funktion angeforderten Stellgrößen entsteht, muss durch eine Verkehrssimulation geschlossen werden. In der Verkehrssimulation werden die im realen Szenario detektierten Fahrzeuge zur Erhöhung der Validität mit individualisierten Fahrermodellen simuliert. Anforderungen an eine Simulationsumgebung zum Einsatz im Rahmen des passiven HAF sowie Kriterien zur Bestimmung des optimalen Fahrermodells werden im Folgenden spezifiziert. Danach erfolgt die Auswahl einer geeigneten Simulationsumgebung anhand der definierten Anforderungen und eine Beschreibung, wie sie im passiven HAF Verwendung findet.

### **3.3.1 Anforderungen an die Simulationsumgebung**

Verkehrssimulationen können nach dem Detailgrad ihrer Modellierung unterschieden werden. Bei makroskopischen Simulationen werden Verkehrsflüsse simuliert. Mikroskopische Simulationen modellieren individuelle Verkehrsteilnehmer. In submikroskopischen Simulationen werden auch die physikalischen Eigenschaften der Fahrzeuge modelliert.

Beim passiven HAF ist das Ziel der Simulation die realitätsgetreue Modellierung des Verkehrsgeschehens zur Detektion fehlerhafter Fahrentscheidungen der HAF-Funktion. Hierzu ist eine detaillierte Modellierung aller Verkehrsteilnehmer erforderlich, die sich in der Umgebung des HAF-Fahrzeugs befinden und deren Verhalten zusammen mit einer fehlerhaften Fahrentscheidung der HAF-Funktion zu einer kritischen Situation führen könnte. Dieses Kriterium wird nur von mikroskopischen und submikroskopischen Verkehrssimulationsumgebungen erfüllt.

Da die simulierten Verkehrsszenarien auf Grundlage von Sensordaten erzeugt werden sollen, ist eine Anforderung an die Simulationsumgebung die Möglichkeit einer automatisierten Erzeugung, Simulation und Auswertung von Simulationsszenarien. Hierzu müssen zum einen die formalen und inhaltlichen Anforderungen an die Szenariospezifikation verfügbar sein. Zum anderen ist es für die Kritikalitätsanalyse erforderlich, auf die Simulationdaten zugreifen und diese nach definierbaren Kriterien auswerten zu können.

Zur Einbindung der HAF-Funktion sollte es darüber hinaus Schnittstellen innerhalb der Simulationsumgebung geben, über die Eingangsdaten im Format der HAF-Sensordaten zur Verfügung gestellt werden können und über die Ausgangsdaten der HAF-Funktion auf die Aktorik des simulierten HAF-Fahrzeugs wirken.

Neben der auf Realdaten basierenden Erstellung des Simulationsszenarios ist es Teil des passiven HAF-Konzepts, allen Verkehrsteilnehmern außer dem von der HAF-Funktion gesteuerten Fahrzeug Fahrermodelle zuzuordnen, die ihr tatsächliches Fahrverhalten bestmöglich modellieren.

Die Kontrollierbarkeit von Autos ist auf zwei Dimensionen beschränkt (vgl. Abbildung 3.3):

1. Entlang der Längsachse ( $x$ -Achse) des Fahrzeugs (Längsführung), bei der Beschleunigung bzw. Verzögerung des Fahrzeugs durch Motormoment und Bremskraft beeinflusst werden können.
2. Quer zur Längsachse ( $y$ -Achse) des Fahrzeugs (Querführung), bei der dem Fahrzeug durch Stellen eines Winkels der Räder der Vorderachse ein Bewegungsanteil senkrecht zu seiner Längsachse aufgeprägt wird.



**Abbildung 3.3:** Fahrzeug mit Koordinatensystem in vorderer Stoßstange

Gleiches gilt für die Bedieneinheiten, die dem Fahrer im Fahrzeug zur Verfügung stehen. Gas- und Bremspedal bestimmen die Längsführung, das Lenkrad die Querführung.

Die Fahraufgabe eines Fahrers zur Steuerung eines Fahrzeugs besteht darin, auf Grundlage des erlernten Fahrverhaltens und des angestrebten Fahrziels unter Berücksichtigung des wahrgenommenen Verkehrsgeschehens Fahrentscheidungen zu treffen. Diese führen zur Bedienung der Steuerelemente des Fahrzeugs, welche letztendlich die Längs- und Querdynamik des Fahrzeugs beeinflussen. Sie lässt sich nach [Don09] in Anlehnung an das 3-Ebenen-Modell von [Ras83] in drei Bereiche gliedern: Navigation, Bahnführung und Stabilisierung (vgl. Tabelle 3.2).

**Tabelle 3.2:** Oben: Drei-Ebenen-Hierarchie der Fahraufgabe nach [Don09]  
Unten: Drei-Ebenen-Modell für zielgerichtete Tätigkeiten nach [Ras83]

Navigation	Bahnführung	Stabilisierung
Handeln in unbekanntem Umgebungen durch Analyse der Umwelt	Bewusstes Handeln in bekannten Situationen	Unbewusstes Handeln nach Handlungsmustern

In der Navigationsebene trifft der Fahrer bewusste Entscheidungen auf Basis von Wissen in unbekanntem und nicht durch feste Regeln definierten Situationen. Die beiden anderen Ebenen werden durch regelbasiertes Verhalten des Fahrers bestimmt. Die Fahrermodellierung ist dabei sowohl auf Bahnführungsebene als auch auf Stabilisierungsebene möglich. Auf der Bahnführungsebene besteht nach [Don09, S. 16] die Aufgabe des Fahrers darin, „aus der vorausliegenden Verkehrssituation sowie aufgrund des geplanten Fahrtablaufs die als sinnvoll erachteten Führungsgrößen wie Sollspur und Sollgeschwindigkeit abzuleiten [...]“.

Die Individualisierung der für die Simulation genutzten Fahrermodelle im Rahmen des passiven HAF muss auf einer oder mehreren der drei Ebenen erfolgen. Dabei führt nur die Ebene Bahnführung zu Manövern als Reaktion auf Verhalten des Umgebungsverkehrs. Zur Reduktion der Modellkomplexität können auf der Bahnführungsebene Fahrentscheidungen in zwei Unterebenen für Entscheidungen in longitudinaler (Geschwindigkeit) und lateraler Richtung (Fahrspur) getrennt modelliert werden. Die Modellierung von komplexeren Manövern wie Überholvorgänge ist dennoch möglich, indem diese entweder von einem übergeordneten allgemeinen Bahnführungsmodell oder von einem der beiden Modelle initiiert werden und die anderen Modellteile von diesem angesteuert werden.

Für die Simulation im Rahmen des passiven HAF ist es das Ziel, die Reaktion des Umgebungsverkehrs auf Fahrhandlungen des HAF zu bestimmen. Hierfür ist das Modell der Bahnführungsebene maßgeblich. Fahrentscheidungen umgebender Verkehrsteilnehmer haben keinen Einfluss auf die gewählte Fahrstrecke oder das Navigationsziel des Fahrers, sodass die Navigationsebene von diesen nicht beeinflusst wird. Auch die Stabilisierungsebene bleibt im Fall von Autobahnscenarien, in denen nicht mehrere Fahrzeuge nebeneinander in derselben Spur fahren, von den Fahrentscheidungen anderer Verkehrsteilnehmer unbeeinflusst.

Für das passive HAF ist eine Individualisierung der Fahrermodellierung auf der Bahnführungsebene demnach hinreichend. Für eine zielgerichtete und effiziente Anpassung ist es daher wünschenswert, dass die Simulationsumgebung über ein bestehendes modular aufgebautes Fahrermodell mit bekannten Schnittstellen verfügt.

Da jedem Fahrer eines Szenarios das Fahrermodell zugeordnet werden soll, welches sein Fahrverhalten am besten abbildet, ist es außerdem notwendig, innerhalb eines Simulationslaufs auf verschiedene Modellierungen zugreifen zu können. Um eine automatische Szenarioerstellung, Simulation und Auswertung zu ermöglichen ist es darüber hinaus erforderlich, die individuellen Modelle im Rahmen der Szenarioerstellung den Fahrern zuordnen zu können.

Daraus resultieren die folgenden Anforderungen an die Simulationsumgebung für die Nutzung im passiven HAF.

Die Simulationsumgebung muss

1. Fahrzeuge mindestens auf mikroskopischer Ebene modellieren.
2. eine Schnittstelle zur externen Steuerung bereitstellen.
3. die Simulationsdaten bzw. das Simulationsergebnis für eine externe Auswertung bereitstellen.
4. eine Ein- und Ausgangsschnittstelle zur Einbindung der HAF-Funktion besitzen.
5. ein Fahrermodell nach den Strukturen von Tabelle 3.2 umsetzen.
6. die Fahrermodellierung modular umsetzen um ein Ersetzen oder Adaptieren einzelner Elemente zu ermöglichen.
7. die Nutzung verschiedener Fahrermodelle innerhalb eines Simulationsszenarios ermöglichen.
8. die Zuordnung der Fahrermodelle im Rahmen der Szenariospezifikation ermöglichen.
9. eine Schnittstellendefinitionen zur Einbindung individueller Modellierungen zur Verfügung stellen.
10. die Anforderungen an eine Szenariospezifikation zur Verfügung stellen.

### 3.3.2 Kriterien zur Bestimmung des optimalen Fahrermodells

Unabhängig von den Anforderungen an die Simulationsumgebung ist die Methode der Zuordnung von Fahrermodellen zu Fahrern eines Szenarios zu bestimmen. Wie weiter oben beschrieben bietet sich zur Reduktion der Komplexität die Zuordnung zu longitudinalen Modellen einerseits und lateralen Modellen andererseits an.

Als Parameter zur Beschreibung des longitudinalen Verhaltens eines Fahrers eignet sich nach [Wie74] und [THH00] in erster Linie die Wunschgeschwindigkeit, zusätzlich der Wunschabstand zu Vorderfahrzeugen und das Beschleunigungsverhalten.

Zur Beschreibung des lateralen Verhaltens existieren wie in Abschnitt 2.3.2 ausgeführt sowohl deterministische als auch probabilistische Ansätze. Neben einfachen Manövern wie Spurhalten, Spurwechsel Links und Spurwechsel Rechts bilden einige Modelle auch komplexere Manöver wie Überholen, Autobahnauffahrt oder Autobahnabfahrt ab. Grundsätzlich lässt sich bei lateralen Manövern zwischen obligatorischen Manövern und taktischen Manövern unterscheiden. Zu den obligatorischen Manövern zählen solche, die zur Erreichung des Navigationsziels erforderlich sind. Ziel taktischer Manöver ist der Wechsel in eine aus Sicht des Fahrers günstigere Fahrspur durch z.B. Wechsel auf eine schnellere Spur, die Einhaltung des Rechtsfahrgebots oder die Kooperation mit einem einscherenden oder überholenden Fahrzeug.

Im Rahmen der Modellierung des Fahrerverhaltens wird das Verkehrsgeschehen, wie es aus Perspektive des Fahrers wahrgenommen wird, als Eingangsdaten. Hierzu gehören die Geschwindigkeit und Position in longitudinaler und lateraler Richtung der umgebenden Fahrzeuge. Neben diesen direkt wahrnehmbaren Größen können aus dem Verhalten der umgebenden Fahrzeuge

und deren eigener Umgebung Schlussfolgerungen durch Prädiktion ihres zukünftigen Verhaltens gezogen werden.

Beispielsweise ist zu erwarten, dass ein Fahrer von Fahrzeug *A*, welches sich einem LKW *B* auf der eigenen Spur nähert und somit eine höhere Geschwindigkeit als *B* hat, diesen überholen will und dafür einen Spurwechsel durchführen wird. Der Fahrer eines Fahrzeugs *C* auf der Spur links von *A* und *B* könnte ein solches Verhalten antizipieren und daher selbst entweder Beschleunigen, um *A* und *B* zu überholen bevor *A* einen Spurwechsel durchführt, oder auf eine Spur weiter links zu wechseln, um *A* einen Spurwechsel auf die bisherige Spur zu ermöglichen. Dies wiederum kann das Verhalten des Fahrers von Fahrzeug *A* beeinflussen, sodass weitere Rückkopplungen entstehen. Je mehr Fahrzeuge in einem Szenario enthalten sind, desto weiter erhöht sich die Komplexität und die Zahl der potentiell möglichen Szenarioverläufe. Zusätzlich zur Modellierung des eigenen Verhaltens muss jedes Fahrzeug Modelle der umgebenden Fahrzeuge enthalten, mit Hilfe derer künftige Aktionen präzisiert und im eigenen Verhalten berücksichtigt werden können.

Die Berücksichtigung des zu erwartenden Verhaltens anderer Verkehrsteilnehmer im Fahrermodell der simulierten Verkehrsteilnehmer erfolgt im passiven HAF nicht explizit. Grund hierfür ist, dass ihr Einfluss auf die Fahrentscheidungen des jeweiligen Fahrers nicht bekannt ist. Die Frage, in wie weit ein Fahrer mögliche Fahrentscheidungen anderer Fahrer in seine eigenen mit einbezieht hängt entscheidend davon ab, wie aufmerksam und vorausschauend dieser fährt. Welche Schlüsse er daraus zieht und wie sich diese auf sein eigenes Verhalten auswirken ist ebenso stark fahrerabhängig.

Die Spurwechselmodelle werden allerdings wie weiter oben beschrieben durch maschinelles Lernen erstellt. Da beim Training die notwendigen Informationen, die ein menschlicher Fahrer zur Prädiktion des Verhaltens anderer Verkehrsteilnehmer nutzen könnte, als von der Fahrzeugsensorik erfasste Eingangsdaten zur Verfügung stehen, wird im Rahmen des Trainings eine implizite Berücksichtigung erfolgen, sofern es zu besseren Trainingsergebnissen der Modelle führt.

Zur Modellierung des longitudinalen Verhaltens eines Fahrers ist es folglich erforderlich, die Parameter Wunschgeschwindigkeit, Wunschabstand und Beschleunigungsverhalten zu bestimmen. Zur Modellierung des lateralen Verhaltens muss das Modell gefunden werden, welches das Spurwechselverhalten des jeweiligen Fahrers bestmöglich reproduziert.

### 3.3.3 Auswahl der Simulationsumgebung

Zur Auswahl der für das passive HAF verwendeten Simulationsumgebung wird ein iterativer Ansatz angewandt. Dabei werden nacheinander Simulationsumgebungen auf ihre Eignung hin überprüft, bis eine Umgebung ermittelt werden kann, die alle Anforderungen erfüllt.

Zunächst wird die Simulationsumgebung IPG Carmaker evaluiert. Die Bewertung beruht auf der öffentlich verfügbaren Produktbeschreibung [IPG20a] und der in der Anwendung zur Verfügung gestellten Dokumentation [IPG20b][IPG20c].

Die Simulationsumgebung IPG Carmaker ist in Industrie und Wissenschaft weit verbreitet und wird für die virtuelle Auslegung und Absicherung unter anderem im Bereich der Fahrdynamik

und Fahrerassistenz verwendet. Die Modellierung der Fahrzeuge erfolgt auf submikroskopischer Ebene. Das automatisierte Erstellen von Simulationsszenarios, eine Auswertung der Simulationsergebnisse durch externe Anwendungen und ein automatisierter Simulationsablauf sind möglich.

Die Fahrermodellierung erfolgt in Carmaker durch den IPGDriver. Einzelne Verkehrsteilnehmer lassen sich durch unterschiedliche Parametrierung des IPGDriver modellieren. Folgende für die Bahnführungsebene relevante Parameter lassen sich nach [IPG20c, S. 209ff] für einzelne Fahrer spezifizieren:

- Wunschgeschwindigkeit
- Maximale longitudinale Beschleunigung
- Minimale Zeitlücke zu vorausfahrendem Fahrzeug
- Maximale Zeitlücke zu vorausfahrendem Fahrzeug
- Minimale Distanz zu einem stehendem Fahrzeug
- Maximale Distanz, bis zu der ein vorausfahrendes Fahrzeug in die Fahrentscheidung mit einbezogen wird
- Überholen anderer Fahrzeuge erlauben
- Überholrate, mit der spezifiziert wird, wie aggressiv ein Fahrer eine Überholabsicht verfolgt
- Minimale Geschwindigkeitsdifferenz zum vorausfahrenden Fahrzeug, ab welcher der Fahrer einen Überholwunsch hat

Das Fahrermodell IPGDriver ist nicht nach dem in Tabelle 3.2 beschriebenen Drei-Ebenen-Modell aufgebaut. Eine Möglichkeit, zur Adaption des Fahrermodells durch fahrerindividuelles Ersetzen einzelner Module wurde nicht gefunden. Es besteht jedoch die Möglichkeit, individuelle Fahrermodelle als zusätzlichen Programmcode einzubinden, welche das IPGDriver-Modell vollständig ersetzen [IPG20b, S. 41, S. 87]. Die Zuordnung unterschiedlicher maschinell trainierter Fahrermodelle zu einzelnen Verkehrsteilnehmer durch Verbindung mehrerer paralleler Simulationen ist ebenfalls möglich. Die Simulationsumgebung IPG Carmaker erfüllt somit alle Anforderungen außer den Anforderungen 5 und 6 aus Abschnitt 3.3.1.

Da IPG Carmaker nicht alle Anforderungen erfüllt, wird als zweites die Simulationsumgebung PELOPS (Programm zur Entwicklung längsdynamischer mikroskopischer Verkehrsprozesse in systemrelevanter Umgebung) untersucht. Die Bewertung erfolgt anhand öffentlich verfügbarer Informationen [Ins14], der in der Anwendung zur Verfügung gestellten Dokumentation und dem während der Bearbeitungszeit vorliegenden Quellcode.

PELOPS wurde am Institut für Kraftfahrzeuge der RWTH Aachen (ika) zusammen mit der BMW AG entwickelt und wird auch vor allem dort für ein breites Anwendungsspektrum aus submikroskopischen, mikroskopischen und makroskopischen Untersuchungen genutzt [HNW97], [HN98].

Die Modellierung der Fahrzeuge erfolgt auf submikroskopischer Ebene. Eine externe Steuerung der Simulation ist ebenso möglich wie die automatische Erzeugung und Auswertung von Simulationsszenarios. Die HAF-Funktion kann entweder als Variante eines Fahrermodells oder



über eine externe Schnittstelle eingebunden werden. Für die Fahrermodellierung wird TRM (**TR**affic **M**odelling) genutzt, welches dem Drei-Ebenen-Modell nach Tabelle 3.2 entspricht. Die Implementierung ist modular, sodass einzelne Ebenen adaptiert oder ersetzt werden können. Alternative Fahrermodelle können als Modul mit definierten Schnittstellen hinzugefügt werden. Eine automatisierte Zuordnung der Fahrermodelle bei Szenariospezifikation durch Vorgaben der zu nutzenden Module für jeden Fahrer ist möglich. Schnittstellen und Anforderungen an die Szenariospezifikation sind dokumentiert und im Quellcode ersichtlich. Damit erfüllt PELOPS alle im vorangegangenen Abschnitt an die Simulationsumgebung gestellten Anforderungen.

**Tabelle 3.3:** Erfüllung der Anforderungen durch IPG Carmaker und PELOPS

#	Anforderung	IPG Carmaker	PELOPS
1	submikroskopisch	ja	ja
2	externe Steuerung	ja	ja
3	externe Auswertung	ja	ja
4	Einbindung HAF	ja	ja
5	Drei-Ebenen-Fahrermodell	nein	ja
6	Modulares Fahrermodell	nein	ja
7	Verschiedene Fahrermodelle in Szenario	ja	ja
8	Fahrermodell Zuordnung durch Szenario	ja	ja
9	Schnittstellen für eigene Modelle bekannt	ja	ja
10	Szenariospezifikation bekannt	ja	ja

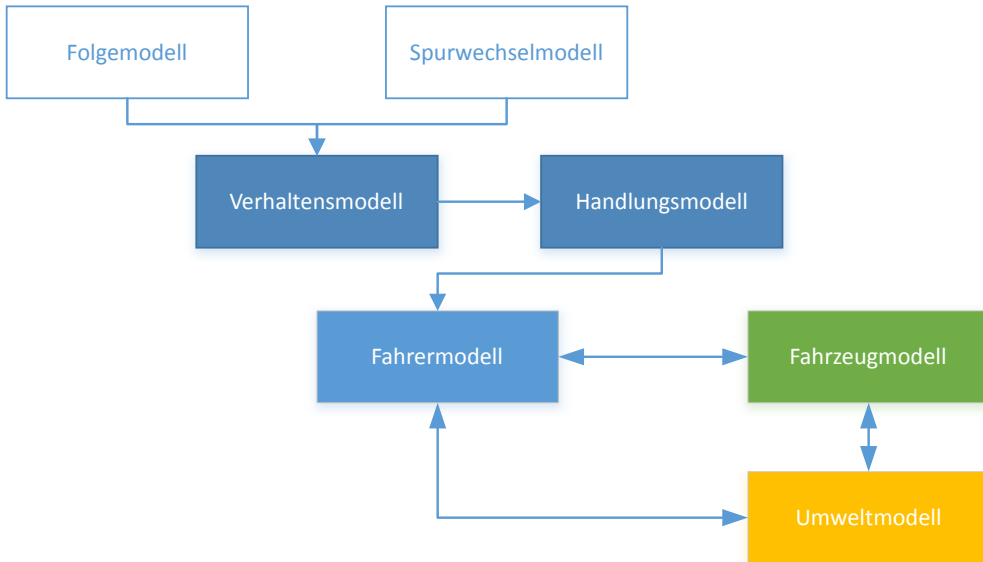
### 3.3.4 Aufbau der gewählten Simulationsumgebung

Die Simulationsumgebung PELOPS besteht aus den drei Komponenten Fahrermodell, Fahrzeugmodell und Umweltmodell, dargestellt in Abbildung 3.4.

Das Fahrermodell, dessen Subkomponente „Spurwechselmodell“ in dieser Arbeit durch maschinell trainierte Modelle ersetzt wird, bildet den Initialpunkt für alle Aktionen der simulierten Verkehrsteilnehmer. Es besteht aus einem Handlungs- und einem Verhaltensmodell, wobei sich letzteres wiederum in ein Folgemodell für die longitudinalen Fahrentscheidungen und in ein Spurwechselmodell für die lateralen Fahrentscheidungen gliedert (nähere Beschreibung siehe Abschnitt 3.3.6). Durch das Verhaltensmodell werden für jede Situation die Attribute Spurwahl und Wunschbeschleunigung bestimmt; das Handlungsmodell setzt diese Vorgaben in fahrzeugseitige Stellgrößen wie Brems- und Gaspedalstellung, Betätigung der Fahrtrichtungsanzeige und Stellen eines Lenkwinkels um.

Diese Größen bilden den Eingang für das Fahrzeugmodell [Dav95], welches sie in Fahrzeugstellgrößen umwandelt, die zu einer Änderung des Motor- und Bremsmoments führen. Aus diesen resultiert wiederum nach Berücksichtigung weiterer Faktoren wie z.B. Fahrwiderständen eine Geschwindigkeitsänderung am Ausgang des Fahrzeugmodells [Ins14], [CH08].

Im Umweltmodell werden die Straßentopologie mit Anzahl an Spuren, Höhenprofil, Kurvenradien und Straßenbeschilderung modelliert. Bei Bedarf können auch Umwelteinflüsse (z.B. Wetter), die eine Auswirkung auf das Fahrer- oder Fahrzeugmodell haben, berücksichtigt werden.



**Abbildung 3.4:** Aufbau des Verkehrssimulationsprogramms PELOPS. Zur Resimulation realer Szenarien wird in dieser Arbeit das Spurwechselmodell durch maschinell trainierte Modelle ersetzt, die den Fahrern individuell zugeordnet werden. Das Folgemodell wird mit fahrerindividuellen Parametern parametrisiert

Zwischen den einzelnen Modellen, die als unabhängige Software-Komponenten umgesetzt sind, werden Daten über eine Schnittstelle ausgetauscht, durch die Umgebungs- und Fahrzeuginformationen zur Verfügung gestellt werden. Dies erfolgt hinsichtlich der Attribute umgebender Verkehrsteilnehmer jeweils relativ zum zu simulierenden Fahrer bzw. Fahrzeug und wird daher als „DrivingView“ bezeichnet, welcher im wesentlichen den Informationen entspricht, die einem HAF-Fahrzeug zur Verfügung stehen. Hierdurch ist auch ohne zusätzlichen Aufwand die Anforderung an die Simulationsumgebung erfüllt, eine Schnittstelle zur Einbindung der HAF-Funktion zur Verfügung zu stellen.

### 3.3.5 Simulationsablauf

Zum Simulieren des fiktiven Verlaufs des HAF-Szenarios muss zunächst das Simulationsszenario erstellt werden. Hierzu werden automatisiert Vorgabedateien auf Grundlage der von der Sensorik erfassten Realdaten generiert, in denen das zu simulierende Szenario beschrieben ist. Diese enthalten folgende Informationen:

- Alle im Szenario zu simulierenden Fahrzeuge
- Initialbedingungen der Fahrzeuge wie Position und Geschwindigkeit
- Spuranzahl
- Spurkrümmung

- Höhenprofil
- Geschwindigkeitsbegrenzungen
- Wetterbedingungen.

Da die bisher in PELOPS verwendeten Simulationsmodelle deterministisch sind, kann auf Grundlage dieser Daten das Szenario beliebig häufig mit identischem Ergebnis simuliert werden. Für die Nutzung des in der Simulation implementierten Fahrermodells müssen die folgenden Parameter vorgegeben werden:

- Wunschgeschwindigkeit
- Sicherheitsbedürfnis
- Überholverbotbefolgung
- Tempolimitbefolgung
- Präferenzspur
- Kooperationsbereitschaft.

Die gewählten Parameter bestimmen das Fahrermodell und beeinflussen die resultierenden Fahrentscheidungen [Lud98]. So könnte bspw. ein sportlicher Fahrer durch hohe Wunschgeschwindigkeit und geringes Sicherheitsbedürfnis beschrieben werden, ein konservativer Fahrer durch geringe Wunschgeschwindigkeit und hohes Sicherheitsbedürfnis.

Für die Straßentopologie sind folgende Vorgaben erforderlich:

- Anzahl der Spuren
- Breite der Spuren
- Abschnitte mit Geraden, Kreisbögen oder Klotoiden
- Höhenverlauf
- Straßenbeschilderung.

Außerdem können zeitlich oder örtlich gesteuerte Ereignisse gesetzt werden, durch welche die Änderung von Parametern zur Laufzeit der Simulation möglich ist. So ist bspw. ein temporäres Geschwindigkeitslimit aktivierbar.

Mit Hilfe dieser Vorgaben kann das Simulationsmodell parametrisiert werden. Zur Laufzeit der Simulation wird in jedem Zeitschritt für jeden Fahrer ein DrivingView generiert, das die Eingangsdaten für das Fahrermodell enthält. Die daraus resultierenden Fahraktionen bestimmen wiederum die Änderungen der DrivingViews im nächsten Zeitschritt.

### 3.3.6 Fahrermodell

Als Fahrermodell wird in PELOPS TRM (**TR**affic **M**odelling) genutzt [CH08], welches nach der Drei-Ebenen-Hierarchie von Donges [Don09] strukturiert ist (vgl. Tabelle 3.2). Dabei werden vornehmlich die beiden unteren Ebenen Bahnführung und Stabilisierung modelliert. Der Navigationsteil wird durch Streckenführung oder festgelegte Ereignisse vorgegeben.

TRM wählt in der Bahnführungsebene zuerst ein Manöver und dann eine an die Umgebungsbedingungen angepasste Geschwindigkeit aus. Zu den Manövern zählen „freie Fahrt“, „Folgefahrt“, „taktischer Spurwechsel“ oder Auf- und Abfahren von Autobahnen, bei denen in lateraler oder longitudinaler Richtung eine dynamische Beschleunigung angefordert wird. Nach Auswahl des aktiven Manövers passt ein Längsbewerter die Geschwindigkeit an Umgebungsbedingungen wie Kurven, Steigungen oder vorausfahrende Fahrzeuge an.

Auf der Bahnführungsebene wird im Rahmen dieser Arbeit das Fahrermodell angepasst, um individuelle Fahrermodelle zu erhalten. Auf der Stabilisierungsebene werden auf Basis des Ergebnisses der Bahnführungsebene Lenkwinkel und Brems- bzw. Gaspedalstellung berechnet und dienen dem Fahrdynamikmodell als Eingangsgrößen. Die Stabilisierungsebene wird in dieser Arbeit nicht angepasst, sondern von PELOPS übernommen.

#### 3.3.6.1 Längsführung

Für die Bestimmung der Sollgeschwindigkeit des Fahrers wird in dieser Arbeit das bereits in TRM implementierte „Wiedemann-Folgemodells“ [Wie74] genutzt. Dieses bestimmt ähnlich wie beispielsweise das in der Simulationsumgebung IPG Carmaker genutzte Fahrermodell IPGDriver Wahrnehmungsschwellen des Fahrers in Abhängigkeit von der gefahrenen Geschwindigkeit und fahrerindividueller Parameter wie Sicherheitsbedürfnis, Schätzvermögen oder Tagesform. Je nach Wahrnehmungsschwelle wird daraufhin in Abhängigkeit von Differenzgeschwindigkeit und Abstand zum jeweils vorausfahrenden Fahrzeug das Fahrverhalten in vier Bereiche unterteilt [Wie74, S. 41ff]:

*freie Fahrt (unbeeinflusstes Fahren):* Kein anderer Verkehrsteilnehmer beeinflusst den Fahrer, sodass dieser seine Wunschgeschwindigkeit erreichen kann.

*Annähern (bewusst beeinflusstes Fahren):* Der Fahrer nähert sich dem vorausfahrendem Fahrzeug bewusst so an, dass zum Zeitpunkt, zu dem der Wunschabstand erreicht wird, die Differenzgeschwindigkeit gering ist.

*Folgen (unbewusst beeinflusstes Fahren):* Der Fahrer passt unbewusst seine eigene Geschwindigkeit an, sodass der Wunschabstand zum Vorderfahrzeug eingehalten wird.

*Bremsen:* Der Fahrer bremst zur Vermeidung einer Kollision bei Unterschreitung des Mindestabstands.

Welches Längsmanöver gewählt wird hängt neben der Wunschgeschwindigkeit  $v_{\text{Wunsch}}$  von Parametern ab, mit denen das Folgemodell parametrisiert wird. Diese sind in Abbildung 3.5 dargestellt und haben die folgenden Bedeutungen:

XS	gewünschter Abstand im Stillstand
XFU	gewünschter minimaler Abstand beim Folgen
XFO	Abstand, ab dem Manöver „Folgen“ aktiv wird
WSDV	Wahrnehmungsschwelle, ab der Geschwindigkeitsdifferenzen bei großen Abständen zum Vorderfahrzeug erkannt werden
WSFS	Wahrnehmungsschwelle, ab der Geschwindigkeitsdifferenzen bei kleinen abnehmenden Abständen zum Vorderfahrzeug erkannt werden
WSFOE	Wahrnehmungsschwelle, ab der Geschwindigkeitsdifferenzen bei kleinen zunehmenden Abständen zum Vorderfahrzeug erkannt werden
XKG	Grenzwert, ab dem Objekte wahrgenommen werden

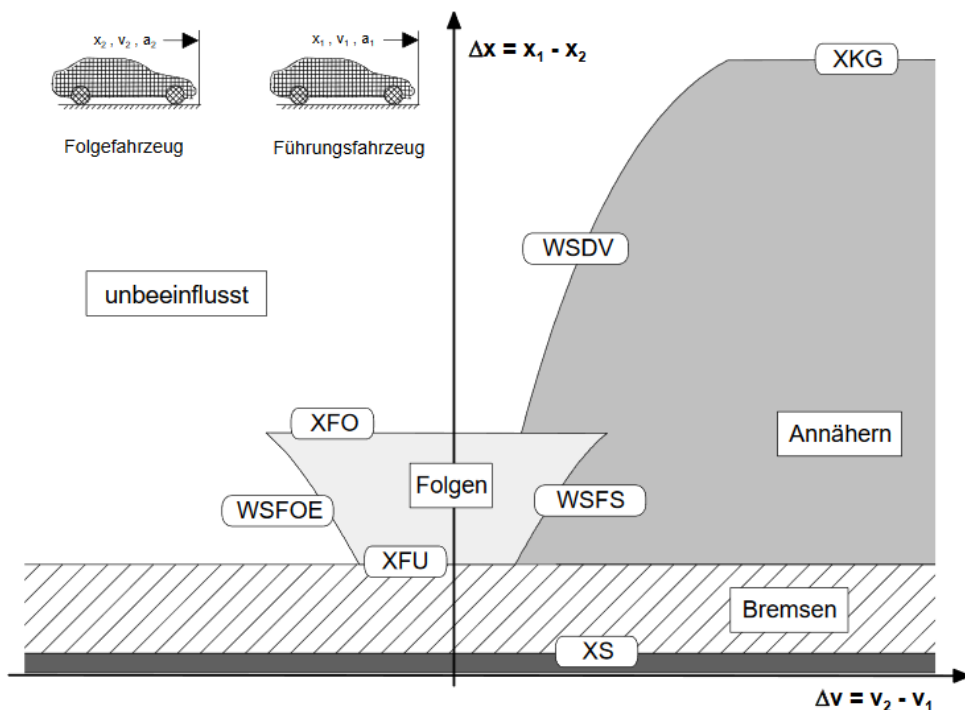


Abbildung 3.5: Wiedemann-Folgemodell nach [Wie74, S. 36]

Das konkrete Verhalten individueller Fahrer hängt je nach Fahrbereich von den gewählten Parametern und der Wunschgeschwindigkeit  $v_{wunsch}$  ab. In freier Fahrt bestimmt  $v_{wunsch}$  das longitudinale Fahrverhalten maßgeblich. Sie wird für die Simulation des, das HAF-Fahrzeug umgebenden, Verkehrs in Abhängigkeit ihres jeweiligen Fahrverhaltens anhand der im Szenarioverlauf gefahrenen Höchstgeschwindigkeit geschätzt. Die übrigen Parameter entscheiden über die Längsführung in den Fahrbereichen Annähern, Folgen und Bremsen. Sie werden im Rahmen einer Parameteroptimierung so gewählt, dass sie das längsdynamische Verhalten des jeweiligen Fahrers im beobachteten Zeitraum bestmöglich abbilden.

Das beschriebene Modell der Längsführung wird für den Einsatz im Rahmen der Methode passives HAF für hinreichend erachtet, da die graduellen Unterschiede zwischen Fahrern über

die geschätzten Parameter abgebildet werden können. Im Gegensatz zur Querführung, bei der durch Spurwechsel immer neue Räume für Fahrentscheidungen anderer Verkehrsteilnehmer entstehen können, ist dies bei der Längsführung nur bei Abweichungen der Fall die so groß sind, dass sich neue Lücken innerhalb der bisherigen Spur öffnen oder schließen. Aus diesem Grund sind auch die Auswirkungen auf die Bestimmung kritischer Situationen geringer.

### 3.3.6.2 Querführung

Neben der Bestimmung der Längsbeschleunigung ist die Bestimmung der in Zukunft zu befahrenden Spur ein Teil komplexer Manöver. Im Folgenden wird das Spurwechselmodell von TRM vorgestellt, da es in Kapitel 4 als Grundlage für die Modellierung von Spurwechseln mit Bayes'schen Netzen genutzt wird.

Neben obligatorischen Spurwechseln, die aufgrund der Routenführung notwendig sind, existieren in TRM taktische Spurwechsel. Mit diesen wird in Abhängigkeit der zur Verfügung stehenden Spuren und des mit einem Wechsel verbundenen Aufwands durch den Fahrer bewertet, ob ein Spurwechsel durchgeführt wird [Ehm02]. Dabei wird ausgehend von der aktuellen Fahrsituation, wie sie sich für den Fahrer durch den DrivingView darstellt, in einem ersten Schritt bewertet, welche Spur die vorteilhafteste ist. Im Fall, dass diese Spur nicht der aktuell eigenen entspricht, wird geprüft, ob der Vorteil gegenüber der eigenen Spur einen Schwellwert überschreitet. Wird dieser Schwellwert überschritten, liegt ein Spurwechselwunsch vor. Ist die Ordnungszahl der Wunschspur geringer als die der aktuellen Spur, handelt es sich um einen Spurwechselwunsch auf die rechts angrenzende Spur, andernfalls um einen Spurwechselwunsch auf die links angrenzende Spur.

Die Bewertung  $b_m$  einer Spur  $m$  beruht auf der Summe der folgenden Einzelwertungen:

*Geschwindigkeit:* Die zu erwartende Geschwindigkeit einer Spur  $m$  wird anhand des Mittelwerts der Geschwindigkeiten  $v_j$  der auf ihr fahrenden Fahrzeuge  $j = 1 \dots n$  abgeschätzt und mit der Wunschgeschwindigkeit  $v_{\text{wunsch}}$  verglichen. Je besser eine Spur geeignet ist, sie mit der Wunschgeschwindigkeit befahren zu können, desto besser ist ihre Bewertung  $b_m^v$  gemäß der Formel

$$b_m^v = \min \left( 1, \frac{\frac{1}{n} \sum_{j=1}^n v_j}{v_{\text{wunsch}}} \right) \quad \forall j \in m. \quad (3.1)$$

*Rechtsfahrgebot:* Je nach Parametrierung des Fahrermodells erhalten rechte Spuren grundsätzlich einen Bonus in ihrer Bewertung gegenüber weiter links liegenden Spuren, da dort das Rechtsfahrgebot eingehalten wird.

$$b_m^r = 1 - \frac{m}{\text{Anzahl Spuren}} \quad (3.2)$$

mit  $m$  als Ordnungszahl der Spur, gezählt von rechts nach links.

*Präferenzspur:* Die bevorzugte Spur  $m_p$  kann in den Fahrerparametern festgelegt werden und bekommt in diesem Fall einen Bonus.

$$b_m^p = \begin{cases} 0,5 & \text{falls } m = m_p \\ 0 & \text{falls } m \neq m_p \end{cases} \quad (3.3)$$

*Umgebende Fahrzeuge:* Drängelnde Fahrzeuge beeinflussen die Bewertung einer Spur negativ. Hierzu wird berechnet, welchen Wert die Time To Collision<sup>4</sup> ( $TTC$ ) zwischen eigenem und folgendem Fahrzeug annimmt.

$$b_m^d = \begin{cases} -1 + \frac{TTC}{3s} & \text{falls } TTC \leq 3s \\ 0 & \text{falls } TTC > 3s \end{cases} \quad (3.4)$$

*Kooperation* Zeigt ein anderes Fahrzeug  $j$  einen Spurwechselwunsch  $SW_{m_j \rightarrow m_D}$  auf die Spur  $m_D$  des Fahrers  $D$  an, erhalten Spuren links von dieser Spur einen Bonus in Abhängigkeit des Fahrerparameters Kooperationsbereitschaft  $k_D$ .

$$b_m^k = \begin{cases} k_D & \text{falls } m > m_D \text{ und } \exists j \text{ mit } SW_{m_j \rightarrow m_D} \\ 0 & \text{sonst} \end{cases} \quad (3.5)$$

Die Gesamtwertung  $b_m$  einer Spur als Summe der Einzelwertungen der Spur in jedem Zeitschritt wird durch einen Tiefpass gefiltert. Dann wird die Differenz zwischen der Gesamtwertung der Nachbarspuren und der Bewertung der aktuellen Spur gebildet. Übersteigt die Differenz zwischen aktueller Spur und der bestbewerteten Nachbarspur einen Schwellwert, erfolgt ein Spurwechsel auf diese. Die Schwellwerte unterliegen einer Hysterese. Hierdurch werden aus einer nur geringen Bewertungsdifferenz resultierende häufige Spurwechsel innerhalb einer kurzen Zeit verhindert [Ins14].

Entscheidet sich ein Fahrer zu einem Spurwechsel, erfolgt die Prüfung der Realisierbarkeit anhand der durch das eigene Fahrzeug notwendigen aufzubringenden Beschleunigung und der von anderen Fahrzeugen aufzubringenden Verzögerung, um einen kollisionsfreien Spurwechsel zu ermöglichen. Liegen diese in einem akzeptablen Bereich, wird der Spurwechsel durchgeführt [Ehm02].

Dieser Teil des in PELOPS genutzten Fahrermodells wird in der vorliegenden Arbeit durch maschinell trainierte Fahrermodelle ersetzt. Diese nutzen die gleichen Eingangsdaten wie das TRM-Spurwechselmodell und geben ebenfalls eine Spurwechselentscheidung aus. Im Gegensatz zum bisher genutzten Modell basieren die maschinell trainierten Modelle jedoch vollständig auf im realen Verkehr beobachteten Spurwechselentscheidungen. Diese wurden als Trainingsdaten genutzt, um alle beobachteten Fahrertypen gemäß dem in Abbildung 3.2 dargestellten Konzept zu trainieren und in der Simulation einsetzen zu können.

<sup>4</sup> Die  $TTC$  beschreibt die Zeit, nach der zwei Fahrzeuge kollidieren, wenn keine Veränderung ihrer Fahrdynamiken erfolgt. Sie ist definiert als Quotient aus Differenzgeschwindigkeit und longitudinalen Abstand zweier Fahrzeuge. Je niedriger die  $TTC$ , desto weniger Zeit bleibt zu reagieren, um eine Kollision zu vermeiden.

### 3.4 Szenarioerstellung aus Sensordaten

Zur Simulation des fiktiven Szenarioverlaufs auf Grundlage eines realen Verkehrsszenarios muss das reale Szenario in die Simulationsumgebung übertragen werden. Die Verarbeitung der Sensordaten kann dabei akausal durchgeführt werden, da eine zeitversetzte Verarbeitung des Szenarios für den Anwendungsfall des passiven HAF hinreichend ist. Dies ist möglich, da die Fahrentscheidungen des passiven HAF nicht im realen Verkehr direkt ausgeführt werden, sondern zu jedem späteren Zeitpunkt in der Simulation überprüft werden können.

Zur Erstellung des Simulationsszenarios werden zur Streckendefinition die im vorigen Abschnitt beschriebenen Vorgabedateien erzeugt, auf deren Grundlage ein Simulationsszenario generiert und dem zu simulierende Fahrzeug ein DrivingView zur Verfügung gestellt wird. Im Folgenden wird zunächst dargestellt, mit Hilfe welcher Vorverarbeitungsschritte die Sensordaten verarbeitet werden, um eine hinreichend hohe Datenqualität zu gewährleisten. Zur Realdatenerfassung wird dabei eine Kamera für die Spurdaten und um das Fahrzeug angeordnete LIDAR für die Objekterkennung genutzt. Anschließend wird gezeigt, wie die verarbeiteten Daten zur Konstruktion eines PELOPS-Szenarios genutzt werden.

#### 3.4.1 Datenvorverarbeitung von Spur- und Objektdaten

Für die Vorverarbeitung der Spurdaten werden die von der Kamera zur Verfügung gestellten Daten einer akausalen Filterung unterzogen.

Die Kamera liefert zum einen Spurmarkierungen in Form von Polynomen dritten Grades. Zum anderen werden die  $x$ - und  $y$ -Koordinaten  $x_i$  und  $y_i$  jedes von der Kamera detektierten Datenpunktes  $p_i$  ausgegeben, anhand derer die Polynome bestimmt werden. Zur Vorverarbeitung der Spurdaten werden alle in einem räumlichen Intervall entlang der Fahrbahn liegenden Datenpunkte genutzt, um die am besten zu den Daten passenden Spurmarkierungen  $SM$  zu finden. Die Akausalität entsteht dadurch, dass hierfür alle gesammelten Punkte unabhängig von ihrem Aufnahmezeitpunkt genutzt werden. Dies kompensiert die prinzipbedingte Schwäche der Kamera, dass von weiter entfernt liegenden Spurmarkierungen nur eine geringe Anzahl an Datenpunkten zur Verfügung steht. Dadurch führen bereits wenige fehlerhaft detektierte Datenpunkte zu fehlerhaften Spurmarkierungs-Polynomen.

Die Filterung der Spurmarkierungen erfolgt durch eine Regression der kleinsten Quadrate der Parameter einer zweidimensionalen Kurve  $SM(s) \in \mathbb{R}^{2 \times 1}$  durch die Messpunkte  $p_i$ . Die Variable  $s$  repräsentiert die im definierten Intervall  $[s_{\min}, s_{\max}]$  bereits zurückgelegte Strecke entlang einer Referenzlinie. Durch Projektion eines Punktes  $p_i$  auf die Referenzlinie kann daher bestimmt werden, auf welcher Streckenhöhe  $s(p_i)$  er liegt. Dies erlaubt die Bestimmung der am besten zu diesem Messpunkt passenden Spurmarkierung  $SM$  durch Lösung des folgenden Optimierungsproblems:

$$SM(s) = \arg \min_{SM(s)} \sum_{i=1}^N \left\| \begin{pmatrix} x_i \\ y_i \end{pmatrix} - SM(s(p_i)) \right\|^2. \quad (3.6)$$

Die beschriebene Filterung wird auf jede in einem Intervall befindliche Spurmarkierung angewandt und daraus für jeden Zeitschritt ein neues Messdatum erzeugt. Auf diese Weise werden



von der Kamera aufgezeichnete Falscherkennungen durch die weit größere Zahl an korrekten Spurmarkierungs-Detektionen marginalisiert. Durch die bessere Qualität der Spurdaten ist auch eine korrekte Spurzuordnung des Sensor-Fahrzeugs selbst sowie der umgebenden Fahrzeuge möglich. Dies erlaubt zum einen eine Nutzung der Daten für anschließendes maschinelles Lernen von Spurwechselmodellen, welches bei fehlerhafter Spurzuordnung der Fahrzeuge aussichtslos wäre. Zum anderen ist es hierdurch möglich, auf realen Szenarien basierende Simulationsszenarien mit hoher Güte zu erzeugen.

Die Filterung der Objekt-Sensordaten erfolgt durch einen Verfolgungs-Algorithmus. Wird ein zum Zeitpunkt  $t = t_{\text{alt}}$  erkanntes Objekt  $o_k$  in folgenden Zeitschritten nicht mehr detektiert, wird seine Position in longitudinaler Richtung zunächst mit einem Konstante-Geschwindigkeit-Modell extrapoliert. Wenn an einer der extrapolierten Positionen  $p_{o_k}(t_{\text{extrapol}})$  von  $o_k$  zu einem Zeitpunkt  $t_{\text{extrapol}}$  mit  $t_{\text{extrapol}} < 5$  s ein Objekt detektiert wird, wird dieses dem ursprünglichen, zum Zeitpunkt  $t_{\text{alt}}$  letztmalig detektierten Objekt, zugeordnet. Für die Zeitschritte, in denen keine Detektion erfolgte, wird ein zwischen den Positionen  $p_{o_k}(t_{\text{alt}})$  und  $p_{o_k}(t_{\text{extrapol}})$  mit dem Konstante-Geschwindigkeit-Modell interpolierter Wert eingefügt. Andernfalls werden die extrapolierten Daten verworfen.

$$p_{o_k}(\tau | \tau \in (t_{\text{alt}}, t_{\text{extrapol}})) = p_{o_k}(t_{\text{alt}}) \cdot \left(1 - \frac{\tau - t_{\text{alt}}}{t_{\text{extrapol}} - t_{\text{alt}}}\right) + p_{o_k}(t_{\text{extrapol}}) \cdot \frac{\tau - t_{\text{alt}}}{t_{\text{extrapol}} - t_{\text{alt}}} \quad (3.7)$$

Diese Vorverarbeitung ist besonders wichtig in Fällen, in denen ein Fahrzeug durch ein anderes verdeckt ist, für die Fahrentscheidungen des HAF-Fahrzeugs oder anderer Verkehrsteilnehmer aber weiterhin relevant bleibt. Dies ist beispielsweise der Fall, wenn zwischen HAF-Fahrzeug und einem PKW ein LKW fährt. Der maximal extrapolierte Zeitraum ist auf  $t_{\text{extrapol}} = 5$  s beschränkt, da ein längerer Zeitraum zu einer zu hohen Zahl an Fehlzuordnungen führen würde.

### 3.4.2 Erstellung eines Umgebungsmodells und Erzeugung eines Simulationsszenarios aus dem Umgebungsmodell

Aus den gefilterten Daten werden Datenobjekte erzeugt, die den in PELOPS genutzten DrivingViews entsprechen. Diese werden der HAF-Funktion übergeben, welche daraufhin einen Fahrbefehl ausgibt. Führen diese zu einem Zeitpunkt zu einer anderen Spurwahl als der des menschlichen Fahrers, wird auf Grundlage des DrivingViews ein PELOPS-Simulationsszenario erstellt. In diesem wird simuliert, welche Folgen das von der HAF-Funktion gewünschte Manöver gehabt hätte. Da das Verfahren nur die realen Sensordaten nutzt, ist es unabhängig von der Verfügbarkeit digitaler Karten des jeweiligen Streckenabschnitts. Dies ist von großem Vorteil gegenüber der Verwendung digitaler Karten, da diese aktuell nicht flächendeckend in der benötigten Genauigkeit verfügbar sind.

Die aus den Sensordaten erzeugten DrivingViews werden nach Feststellung einer Abweichung zwischen menschlichen und HAF-Fahrmanövern für weitere  $t_{\text{mess}} = 10$  s berücksichtigt und in ein globales, fahrzeugunabhängiges Koordinatensystem transformiert, sodass ein PELOPS-Szenario generiert werden kann (vgl. Abschnitt 3.3). Aufgrund der nach unten begrenzten

Abtastung der Sensoren kann dieser Prozess nicht fehlerfrei durchgeführt werden. Beispielsweise wird die Spurkrümmung nur in einem Zyklus von  $\Delta T_c = 20$  ms ermittelt. Für diesen Zeitraum wird daher die Spurkrümmung stets als konstant im Simulationsszenario angenommen, sodass dieses aus einer Verkettung von Streckenabschnitten mit konstanter Krümmung besteht. Gleiches gilt für die Abtastung der aktuellen Geschwindigkeit des HAF-Fahrzeugs, mit deren Hilfe der zurückgelegte Weg und somit auch die Streckenlänge ermittelt wird. Daher kann (neben möglicher Einflüsse von Messfehlern) die Szenarioerstellung bestenfalls so präzise sein wie dies unter Berücksichtigung der Abtastfehler möglich ist, welche sich zudem über die Zeit summieren.

Zum Zeitpunkt der Szenarioerstellung wird außerdem für jeden Fahrer, welcher sich innerhalb des Messbereiches befindet, die Wunschgeschwindigkeit bestimmt. Gleichzeitig erfolgt die Zuordnung des am besten passenden Spurwechselmodells, welche im folgenden Abschnitt 3.5 beschrieben ist.

### 3.5 Zuordnung von maschinell gelernten Fahrermodellen

Ein wichtiger Teil des Konzepts passives HAF ist die Simulation der dynamischen Anteile eines Fahrplanszenarios. Um die Validität der Ergebnisse sicherzustellen werden vorab trainierte Fahrermodelle den Fahrern des im Realverkehr beobachteten Szenarios zugeordnet. Dabei werden alle von einem Fahrzeug erfassten Sensordaten sowie die im realen Szenario getroffenen Spurwechselentscheidungen gespeichert. Zur Erfüllung der in Abschnitt 3.3.2 genannten Kriterien für die Auswahl des Spurwechselmodells werden mit diesen Daten für jedes vorab trainierte und in die Simulation integrierte Modell die Abweichung zum realen Fahrverhalten berechnet und das Modell mit dem geringsten Fehler dem Fahrer zugeordnet.

Hierzu werden die Sensordaten auf die gleiche Weise wie für das Training der Modelle erforderlich aufbereitet. Daraufhin wird für jedes Spurwechselmodell  $M_S$  der Fehler  $f_S$  in jedem Beobachtungszeitpunkt  $t$  berechnet. Dabei werden zunächst für jedes Modell  $M_S$  die Wahrscheinlichkeiten  $\mathbf{P}_{M_S}(t)$  für alle Spuren  $m$  zum Zeitpunkt  $t$  berechnet. Daraufhin wird die Spur  $m_i$  mit der höchsten vom Spurwechselmodell  $M_S$  ausgegebenen Wahrscheinlichkeit  $P_{m_i}(t) = \max(\mathbf{P}_{M_S}(t))$  bestimmt. Je nachdem, ob diese der vom Fahrer tatsächlich gefällten Spurentscheidung  $m_r$  entspricht, berechnet sich der Fehler  $f_S$ :

$$f_S(t) = \begin{cases} 1 - P_{m_i}(t) & \text{für } m_i = m_r \\ 1 + P_{m_i}(t) & \text{für } m_i \neq m_r. \end{cases} \quad (3.8)$$

Der Fehler ist also umso kleiner, je näher die vom Spurwechselmodell ausgegebene Wahrscheinlichkeit der beobachteten Aktion des Fahrers entspricht. Gleichzeitig wird in jedem Zeitschritt der Fehlermittelwert  $\bar{f}_S$  berechnet, um anhand dessen das Modell  $M_S$  mit dem geringsten mittleren Fehler bestimmen zu können:

$$\bar{f}_S = \frac{1}{n} \sum_{t=1}^n f_S(t). \quad (3.9)$$

Da sich umgebende Fahrzeuge jeweils nur für einen begrenzten Zeitraum in der Nähe des HAF-Fahrzeugs befinden, wird dabei nicht berücksichtigt, dass sich das Fahrverhalten eines

Fahrers während einer Fahrt ändern kann. Dies wäre über eine geringere Gewichtung von weiter in der Vergangenheit liegenden Daten für die Berechnung des mittleren Fehlers umsetzbar, wenn Fahrzeuge über einen längeren Zeitraum zu den umgebenden Fahrzeugen des HAF-Fahrzeugs gehören. Dies beträfe insbesondere Länder mit Höchstgeschwindigkeit auf Autobahnen, in welchen sich viele Fahrzeuge sehr gleichmäßig fortbewegen. Da die in dieser Arbeit genutzten Daten im Gebiet der Bundesrepublik Deutschland gesammelt wurden und die maximale Beobachtungsdauer von Fahrzeugen im einstelligen Minutenbereich lag, wird dieser mögliche Einfluss nicht berücksichtigt.

Der Gesamtprozess der Szenarioerstellung ist in Abbildung 3.6 dargestellt. Das reale Szenario wird mit Hilfe der Sensorik erfasst, zur Verbesserung der Datenqualität vorverarbeitet (in der Abbildung blau dargestellt) und an die Software-Komponente Simulation übertragen. Diese besteht aus drei Unterkomponenten:

1. Der Umweltverwaltung, die aus den erhaltenen Daten ein Simulationsszenario erstellt und den Fahrzeugen im Szenario passende Fahrermodelle zuordnet (in der Abbildung gelb dargestellt).
2. Einer Implementierung der HAF-Fahrfunktion, die mit diesem Ansatz überprüft werden soll (in der Abbildung rot dargestellt).
3. Dem Simulationsspeicher mit der Simulationsumgebung, die bei Abweichungen zwischen Fahrer und HAF-Funktion gestartet wird und für  $t_{\text{mess}} = 10 \text{ s}$  den weiteren Verlauf des Szenarios bei Umsetzung der Manöver der HAF-Funktion simuliert (in der Abbildung grün dargestellt).

Die Ergebnisse der Simulation werden an einen Kritikalitätsalgorithmus übergeben, der jeden Simulationsschritt auf Verletzung der Sicherheitskriterien überprüft (siehe Kapitel 5).

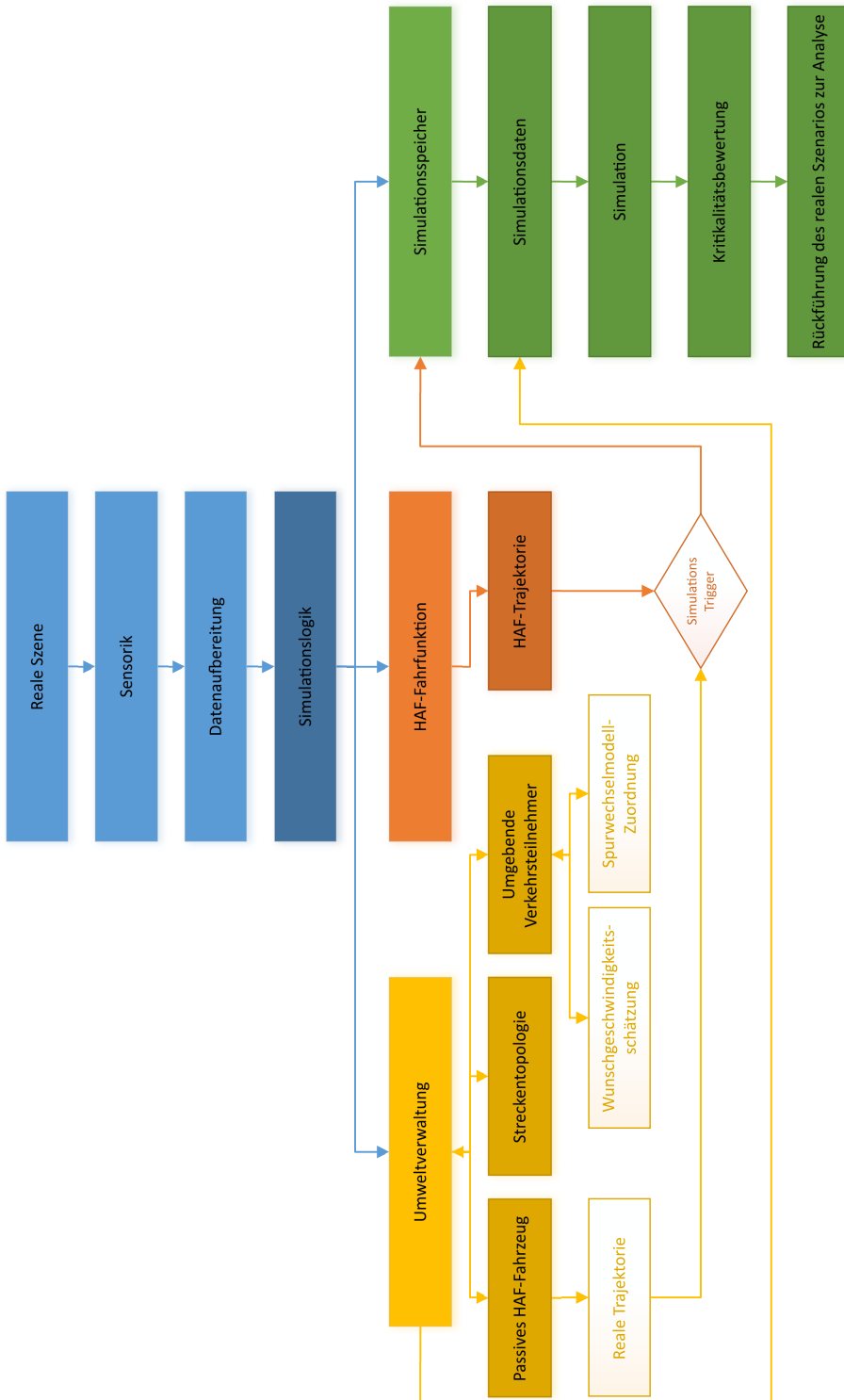


Abbildung 3.6: Prozessablauf des passiven HAF

## **4 Maschinelles Lernen von Fahrermodellen zur Simulation umgebender Verkehrsteilnehmer**

Im vorangegangenen Kapitel 3 wurde das Konzept des passiven HAF erklärt. Wichtiger Bestandteil dieses Konzepts sind die individualisierten Fahrermodelle, die den Fahrern des realen Szenarios zugeordnet und zur Simulation des fiktiven HAF-Szenarios genutzt werden.

Die Modellierung der Fahrer im Rahmen dieses Konzepts erfolgt auf der Bahnführungsebene. Zur Modellierung des Längsverhaltens wird das Wiedemann-Folgemodell verwendet. Eine detailliertere Beschreibung des Modells und seiner konkreten Verwendung erfolgte in Abschnitt 3.3.6 im Rahmen der Beschreibung der Simulationsumgebung. Zur Modellierung des Spurwechselverhaltens werden in diesem Kapitel drei Methoden vorgestellt und Ansätze zur Modellierung von Fahrern mit Hilfe dieser Methoden präsentiert, untersucht und verglichen.

### **4.1 Modellierung des Spurwechselverhaltens durch maschinelles Lernen**

Die Sollspur bzw. Wunschspur der Fahrer soll in dieser Arbeit durch maschinell gelernte Spurwechselmodelle modelliert werden, um danach den, das HAF-Fahrzeug umgebenden, Fahrzeugen das jeweils am besten passende zuzuordnen (vgl. Abschnitt 3.5). Hierbei wird das Ziel verfolgt, die Manöverentscheidungen der Fahrer mit hoher Güte nachzubilden, um eine hohe Validität der Simulation zu erreichen. Im Gegensatz zur Längsführung wird nicht auf ein etabliertes, im Fall des Wiedemann-Modells deterministisches, Modell zurückgegriffen. Stattdessen werden in den folgenden drei Abschnitten unterschiedliche, probabilistische Ansätze zur Modellierung der Spurwechselentscheidungen von Fahrern entwickelt und auf ihre Eignung hin untersucht.

Als Eingangsgrößen für die maschinell zu trainierenden Modelle dienen dabei die von den Sensoren des HAF-Fahrzeugs erfassten Daten. Ausgangs- bzw. Zielgrößen des Lernprozesses sind die tatsächlich durchgeführten Spurwechsel. Die konkreten Eingangsgrößen sind dabei aus Sicht des zu modellierenden Fahrers in Tabelle 4.1 aufgelistet. Die Nomenklatur zur Bezeichnung des HAF-Fahrzeugs und der umgebenden Fahrzeuge ist in Abbildung 4.1 dargestellt. Dabei wird das HAF-Fahrzeug mit S bezeichnet, S folgende Fahrzeuge mit F, vorausfahrende mit V. Fahrzeuge auf einer Nachbarspur von S werden zusätzlich mit R für rechts bzw. L für links gekennzeichnet.

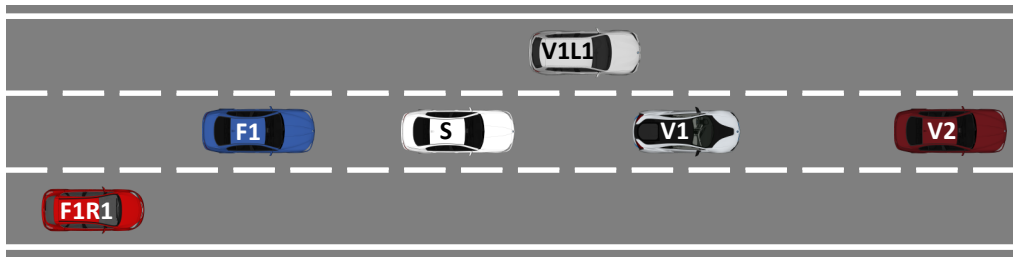


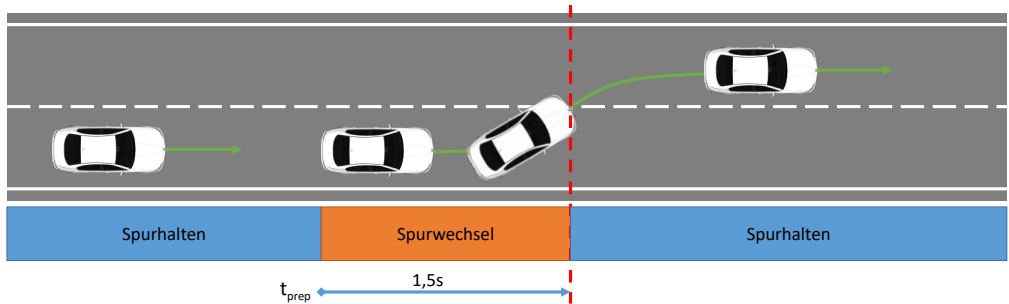
Abbildung 4.1: Nomenklatur der Fahrzeuge

Bei den Eingangsdaten, die den umgebenden Verkehr betreffen, werden die Fahrzeuge auf der eigenen Spur des zu modellierenden Fahrers und auf den beiden benachbarten Spuren berücksichtigt, die sich direkt vor oder hinter diesem befinden. Da sich die Qualität der Sensordaten mit zunehmendem Abstand verschlechtert, werden die fahrdynamischen Größen der umgebenden Fahrzeuge Abstand, Geschwindigkeit und Beschleunigung auf die in den Klammern stehenden Bereiche beschränkt, welche sich in der praktischen Anwendung als Grenzen für noch nutzbare Daten der gewählten Sensorkonfiguration erwiesen haben.

Tabelle 4.1: Eingangsgrößen des zu modellierenden Fahrers

Beschreibung	Größe	Einheit
Fahrzeuggeschwindigkeit	$v_S$	$m/s$
Geschätzte Wunschgeschwindigkeit	$\hat{v}_{\text{wunsch}}$	$m/s$
Existenz von Nachbarspuren rechts ( $\iota = R$ ) und links ( $\iota = L$ )	$e_{\text{lane}}^{\iota}$	-
Anzahl der Spuren zwischen aktueller und rechterster Spur	$k_r$	-
Existenz von vorausfahrendem ( $\kappa = V$ ) und folgendem ( $\kappa = F$ )		
Umgebungsverkehr auf der eigenen und auf Nachbarspuren	$e_{\kappa}^{\text{obj}}$	-
Abstand zu umgebenden Fahrzeugen	$d_{\kappa}^{\text{obj}}$	$[0, 100.0] m$
Geschwindigkeitsdifferenz zu umgebenden Fahrzeugen	$v_{\kappa}^{\text{obj}}$	$[0, 50.0] m/s$
Beschleunigung umgebender Fahrzeuge	$a_{\kappa}^{\text{obj}}$	$[-3.5, 2] m/s^2$
Krümmung der Spur	$cur$	$1/m$
Maximale Krümmung der Spur	$cur_{\text{max}}$	$1/m$
Kehrwert der „time to collision“ zu umgebenden Fahrzeugen	$T_{\text{ittc},\kappa}$	$[0, 0.25] 1/s$
Kehrwert der Zeitlücke zu anderen Fahrzeugen	$T_{\text{itgap},\kappa}$	$[0, 4.0] 1/s$

Zum Training der Modelle werden die Trainingsdaten je nach Fahrverhalten mit den Labels „Spurhalten“, „Spurwechsel Rechts“ oder „Spurwechsel Links“ versehen. Das Label Spurwechsel links bzw. Spurwechsel rechts erhalten dabei die Daten, die bis zu  $t_{\text{prep}} = 1,5 s$  vor einem Spurwechsel aufgezeichnet wurden. Als Spurwechsel ist dabei der Zeitpunkt definiert, bei dem der Mittelpunkt der vorderen Fahrzeugstoßstange die Spurlinie übertritt (vgl. Abbildung 4.2). Alle anderen Daten erhalten das Label Spurhalten. Die Zeit  $t_{\text{prep}}$  wurde nach der in [TGG<sup>+</sup>97] und [LOW04, S. 9] ermittelten Mindestspurwechseldauer auf Autobahnen gewählt, da in diesem Zeitraum sicher von einer Spurwechselentscheidung ausgegangen werden kann.



**Abbildung 4.2:** Trainingsdaten erhalten das Label Spurwechselwunsch ab  $t_{\text{prep}}$  bis zur Überquerung der Spurmarkierung für 1,5 s

Nachdem ein Spurwechsel beobachtet wurde ist seine Eintretenswahrscheinlichkeit, also die *a posteriori* Wahrscheinlichkeit,  $P(\text{Spurwechsel}) = 1$ . Da die Fahrermodelle beim Einsatz im Konzept passives HAF aufgrund der Eingangsdaten und nicht aufgrund der tatsächlich beobachteten Spurwechsel entscheiden sollen, wie ein Fahrer sich in einem Zeitschritt verhält, müssen sie *a priori* Wahrscheinlichkeiten für Spurwechsel ausgeben. Bevor ein Spurwechsel in eine der beiden Richtungen beobachtet wurde, war dieser nur eine von drei Möglichkeiten mit zunächst unbekannter Wahrscheinlichkeit. Um dies zu berücksichtigen und somit keine Labels mit Wahrscheinlichkeiten zu erzeugen, bei denen ein Zustand die Wahrscheinlichkeit 1 und die beiden anderen 0 annehmen, wird dem jeweils aktiven Label eine Mindestwahrscheinlichkeit von  $\gamma_{\text{aktiv}_-} = 0,8$  zugewiesen. Dann wird eine Zufallszahl  $\chi$  aus einer Gleichverteilung im Intervall  $[0, 0, 2]$  gezogen und zur Mindestwahrscheinlichkeit des aktiven Labels addiert. Die verbleibende Differenz zur Gesamtwahrscheinlichkeit von 1 wird auf die beiden inaktiven Labels  $\gamma_{\text{inaktiv}}$  zu gleichen Teilen verteilt:

$$\gamma_{\text{aktiv}} = \gamma_{\text{aktiv}_-} + \chi \quad (4.1)$$

$$\gamma_{\text{inaktiv}} = \frac{1 - \gamma_{\text{aktiv}}}{2}. \quad (4.2)$$

## 4.2 Modellierung mit Bayes'schen Netzen

Eine in dieser Arbeit untersuchte Möglichkeit der probabilistischen Modellierung von Spurwechselentscheidungen erfolgt mit Hilfe von Bayes'schen Netzen, welche bereits in [GBD15] und [WMT<sup>+</sup>16] für die Prädiktion von Spurwechseln verwendet wurden.

Im Folgenden wird dargelegt, wie eine Modellierung von Spurwechselentscheidungen durch Bayes'sche Netze möglich ist und effizient durchgeführt werden kann. Hierzu werden in Abschnitt 4.2.1 zunächst die Eigenschaften der Netze vorgestellt und dargestellt, wie Bayes'sche Netze in Markov-Netze und Cliquesbäume transformiert werden können. Anschließend wird gezeigt, wie durch diese Transformation eine effiziente Inferenz innerhalb der Netze möglich ist. Inferenz bezeichnet dabei das logische Schlussfolgern aufgrund von gegebenem Wissen und der Struktur einer Wahrscheinlichkeitsverteilung. Danach wird in Abschnitt 4.2.2 eine

Netzstruktur für die Spurwechselmodellierung präsentiert. In Abschnitt 4.2.3 wird erläutert, welche Anpassungen an den klassischen Bayes'schen Netzen notwendig sind, um sie für den gewünschten Einsatzzweck nutzen zu können und wie in diesen erweiterten Bayes'schen Netzen Inferenz (Abschnitt 4.2.4) und Training (Abschnitt 4.2.5) erfolgen kann.

## 4.2.1 Grundlagen

Mit Hilfe von Bayes'schen Netzen lassen sich Zusammenhänge zwischen Wahrscheinlichkeitsverteilungen darstellen. Im folgenden Abschnitt werden die Eigenschaften von Bayes'schen Netzen vorgestellt. Die Definitionen sind [KF09] entnommen. Die Beschreibung der Grundlagen erfolgt aufgrund der höheren Anschaulichkeit anhand von diskreten Bayes'schen Netzen, also Netzen, in denen nur diskrete Wahrscheinlichkeitsverteilungen enthalten sind. Dabei wird folgende Notation verwendet: Zufallsvariablen werden durch Großbuchstaben dargestellt; Vektoren von Zufallsvariablen durch fettgedruckte Großbuchstaben. Wissen über eine Zufallsvariablen wird durch Kleinbuchstaben dargestellt, Wissen über mehrere Zufallsvariablen durch einen fettgedruckten Kleinbuchstaben.

Für die Modellierung von Fahrern werden hybride Bayes'sche Netze verwendet, die sowohl diskrete als auch kontinuierliche Wahrscheinlichkeitsverteilungen enthalten.

### 4.2.1.1 Definition und Eigenschaften

Die Definition des Bayes'schen Netzes lautet nach [KF09, S. 57]:

**Definition 4.1 Bayes'sches Netz**

Ein Bayes'sches Netz  $\mathbf{B}$  über die Zufallsvariablen  $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$  ist ein Tupel  $\langle G, \theta \rangle$  mit

- $G$ : azyklischer Graph, dessen Knoten die Zufallsvariablen  $\mathbf{X}$  repräsentieren und dessen Kanten gerichtet sind.
- $\theta$ : bedingte Wahrscheinlichkeitsverteilungen  $\theta$  mit  $\theta_i = P(X_i | El_G(X_i)) \forall i$  mit  $El_G$  als der Menge, die alle Elternknoten von  $X_i$  enthält.
- Elternknoten  $El_G(X_i)$  von  $X_i$  sind alle Knoten, die durch eine gerichtete Kante in Richtung  $X_i$  mit  $X_i$  verbunden sind.

Anhand von Definition 4.1 lassen sich mit der Kettenregel nach [KF09, S. 62] alle Wahrscheinlichkeitsverteilungen innerhalb des Netzes berechnen.



**Definition 4.2 Kettenregel**

$$P(X_1, \dots, X_n) = \prod_i P(X_i | El_G(X_i))$$

Eine Wahrscheinlichkeitsverteilung ist nach [KF09, S. 16] folgendermaßen definiert:

**Definition 4.3 Wahrscheinlichkeitsverteilung**

Eine Wahrscheinlichkeitsverteilung  $P$  über das Tupel  $\langle \Omega, Z \rangle$  ist eine Abbildung von Ereignissen  $Z$  zu reellen Werten, die folgende Bedingungen erfüllt:

- $P(a) \geq 0 \forall a \in Z$ .
- $P(\Omega) = 1$ , mit  $\Omega$  als Potenzmenge über alle Ereignisse.
- Wenn  $a, b \in Z$  und  $a \cap b = \emptyset$  dann ist  $P(a \cup b) = P(a) + P(b)$ .

Um hiermit die Wahrscheinlichkeit für eine konkrete Zustandskombination zu berechnen, müssen die entsprechenden Wahrscheinlichkeitswerte der Zustände eingesetzt werden. Die gemeinsame Wahrscheinlichkeitsverteilung ist dabei der Faktor, in dessen Umfang alle Zufallsvariablen enthalten sind (vgl. Definition 4.4). Ein Faktor kann aber auch eine Untermenge aller Variablen eines Bayes'schen Netzes umfassen. Die folgenden Definitionen sind [KF09, S. 104ff] entnommen.

**Definition 4.4 Faktor und Umfang des Faktors**

Sei  $\mathbf{X}$  eine Menge an Zufallsvariablen. Dann ist der Faktor  $\phi(\mathbf{X})$  eine Funktion, die jede Ausprägungen von  $\mathbf{X}$  auf einen reellen Wert abbildet,  $\text{Ausprägung}(\mathbf{X}) \rightarrow \mathbb{R}$ . Ein Faktor ist nicht-negativ, wenn alle seine Eingänge nicht-negativ sind. Die Menge an Variablen  $\mathbf{X}$  ist der Umfang des Faktors und wird als  $\text{Umfang}[\phi]$  bezeichnet.

**Definition 4.5 Basisoperationen für Faktoren**

Für Faktoren sind die folgenden Basisoperationen definiert:

- Das **Produkt** zweier Faktoren enthält die Umfänge beider Faktoren:  
 $\phi(A) * \phi(B) = \phi(A, B)$ .
- Durch einbringen von **Wissen** über eine Zufallsvariable des Faktors werden alle Zustandskombinationen aus dem Faktor entfernt, die mit diesem Wissen nicht vereinbar sind:  $\phi(A, B, C|A = a) = \phi(a, B, C)$ .
- Durch **Summation** über alle Zustände einer Zufallsvariablen kann diese aus einem Faktor **marginalisiert** werden:  $\sum_A \phi(A, B, C) = \phi(B, C)$ .

Mit Hilfe dieser Operationen lässt sich ein Bayes'sches Netz faktorisieren, d.h. in Faktoren aufteilen (nach [KF09, S. 62]).

**Definition 4.6 Faktorisieren**

Sei  $G$  ein Bayes'sches Netz über die Zufallsvariablen  $X_1, \dots, X_n$ . Eine Verteilung  $P$  über den gleichen Umfang lässt sich nach  $G$  faktorisieren, wenn  $P$  als das Produkt

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | El_G(X_i)) \quad (4.3)$$

dargestellt werden kann.

Eine wichtige Eigenschaft von Bayes'schen Netzen ist es, voneinander abhängige und unabhängige Zufallsvariablen darzustellen. Dabei sind zwei Variablen voneinander abhängig, wenn Wissen über eine Variable die Wahrscheinlichkeitsverteilung der anderen verändert. Es besteht in diesem Fall ein „aktiver Pfad“ zwischen ihnen. Diese Abhängigkeit ist in Bayes'schen Netzen immer symmetrisch und in den folgenden Fällen zwischen den Variablen  $X$  und  $Y$  gegeben:

- Direkte Verbindung:  $X \rightarrow Y$
- Indirekte Verbindung:  $X \rightarrow W \rightarrow Y$
- gemeinsamer Grund  $W$ :  $X \leftarrow W \rightarrow Y$

Liegt Wissen für eine Zufallsvariable vor, so blockiert dies den aktiven Pfad zwischen den Zufallsvariablen vor (Elternknoten) und nach (Kinds-knoten) dieser Variable, sodass diese unabhängig werden.

Keine Abhängigkeit liegt zunächst bei einem gemeinsamen Resultat  $W$  vor.

- V-Struktur:  $X \rightarrow W \leftarrow Y$

Diese Struktur wird als V-Struktur bezeichnet. Dies ändert sich, wenn Wissen über die Zufallsvariable  $W$  bekannt ist. Hierdurch wird der Pfad zwischen  $X$  und  $Y$  aktiv, sie werden abhängig voneinander.

Die Wahrscheinlichkeiten  $P$  lassen sich nach dem Graphen  $G$  in Faktoren aufteilen. Aufgrund der oben beschriebenen Abhängigkeiten ist jeder Knoten unabhängig von allen Knoten, die nicht seine Nachkommen sind, wenn seine Elternknoten gegeben sind. Daher stellt ein Bayes'sches Netz eine Unabhängigkeitsabbildung einer Wahrscheinlichkeitsverteilung dar [KF09, S. 60f]:

#### **Definition 4.7 Unabhängigkeitsabbildung**

Sei  $P$  eine Verteilung über  $\mathcal{X}$ . Sei  $I(P)$  die Menge der Unabhängigkeitsannahmen der Art  $(X \perp Y | Z)$ , die in  $P$  gelten. Wenn  $P$  alle Unabhängigkeiten des Graphen  $G$  einhält, dann ist  $G$  eine Unabhängigkeitsabbildung von  $P$ .  $P$  kann aber auch weitere Unabhängigkeiten enthalten, die nicht in  $G$  dargestellt sind. Dann ist  $I(G) \subset I(P)$ .

#### **Definition 4.8 Minimale Unabhängigkeitsabbildung**

Ein Graph  $G$  ist eine minimale Unabhängigkeitsabbildung für eine Menge an Unabhängigkeiten  $I$ , wenn

- er eine Unabhängigkeitsabbildung von  $I$  gemäß Definition 4.7 ist.
- keine Kante von  $G$  entfernt werden kann, ohne dass  $G$  keine Unabhängigkeitsabbildung von  $I$  mehr ist.

Dann gilt (Beweis siehe [KF09, S. 62]):

#### **Satz 4.1 Faktorisierung von $P$**

Sei  $G$  ein Bayes'sches Netz von einer Menge von Zufallsvariablen  $\mathbf{X}$  und sei  $P$  eine gemeinsame Wahrscheinlichkeitsverteilung über die gleiche Menge. Wenn  $G$  eine Unabhängigkeitsabbildung von  $P$  ist, dann lässt sich  $P$  wie  $G$  faktorisieren.

Die Funktionsweise von Bayes'schen Netzen wird im folgenden Beispiel illustriert.

#### **Beispiel 4.1 Bayes'sches Netz für Ankunftszeit eines Fahrers**

Ein Fahrer führt in Abhängigkeit von der Verkehrslage ( $V$ ) und ob er es eilig hat ( $E$ ) einen Spurwechsel ( $SW$ ) auf eine vermeintlich schnellere Spur ( $SS$ ) durch. Abhängig von der Spurwechselentscheidung und ob die andere Spur tatsächlich schneller ist, hat er eine frühere oder spätere Ankunftszeit ( $A$ ) als geplant. Diese Abhängigkeiten lassen sich als Bayes'sches Netz modellieren, welches in Abbildung 4.3 dargestellt ist.

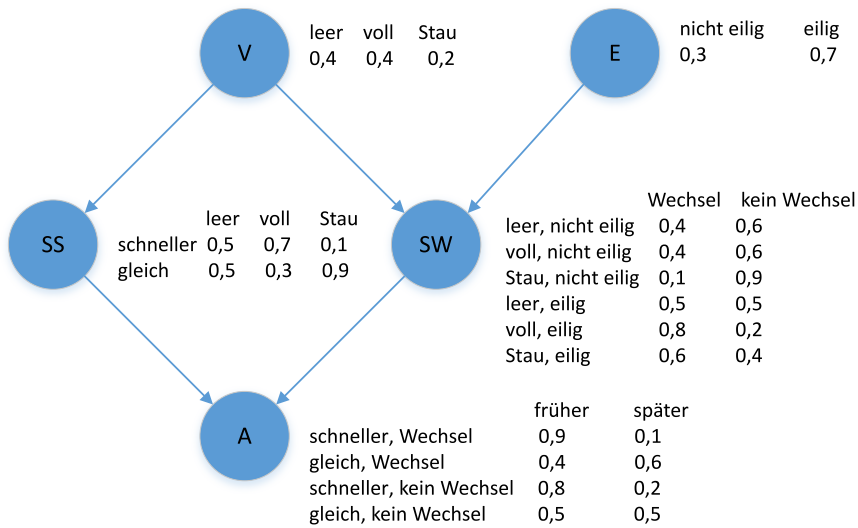


Abbildung 4.3: Beispielhaftes Bayes'sches Netz

Am Beispiel des Fahrers wird deutlich, dass anhand der Definitionen 4.1, 4.2 und 4.3 die Wahrscheinlichkeiten aller Zustandskombinationen berechnet werden können. Die aus der Kettenregel resultierende gemeinsame Wahrscheinlichkeitsverteilung für das Beispiel lautet:

$$P(V, E, SS, SW, A) = P(V) * P(E) * P(SS|V) * P(SW|V, E) * P(A|SS, SW) \quad (4.4)$$

Aus Gleichung (4.4) folgt, dass sich die Variablen gemäß dem in Abbildung 4.3 dargestellten Graph faktorisieren lassen,  $G$  also eine Unabhängigkeitsabbildung nach Definition 4.7 ist.

Die Eigenschaft des aktiven Pfades ist ebenfalls im Beispielnetz dargestellt. Ohne weiteres Wissen sind die Variablen  $V$  und  $E$  unabhängig voneinander, das Verkehrsaufkommen hat keinen Einfluss darauf, ob der Fahrer es eilig hat. Ist aber bekannt, dass der Fahrer einen Spurwechsel durchgeführt hat und es voll war, ist die Wahrscheinlichkeit höher, dass der Fahrer es eilig hatte und die Spur wechselte, um möglichst früh am Ziel anzukommen.

#### 4.2.1.2 Analogie zu Markov-Netzen und Cliquesbäumen

Neben Bayes'schen Netzen lassen sich probabilistische Verteilungen und ein Teil ihrer Unabhängigkeiten auch mit Markov-Netzen darstellen. Im Folgenden wird erklärt, wie eine effiziente Inferenz in Bayes'schen Netzen möglich ist. Hierzu werden sie in Markov-Netze transformiert. Diese werden zur Konstruktion von Cliquesbäumen genutzt, die den Einsatz effizienter Inferenz-Algorithmen erlauben.

Zunächst erfolgt die Definition der Markov-Netze als Gibbs-Verteilung nach [KF09, S. 108].

**Definition 4.9 Gibbs-Verteilung**

Eine Verteilung  $P_{\Phi}$  ist eine Gibbs-Verteilung, die durch eine Menge von Faktoren  $\Phi = \phi_1(\mathbf{D}_1), \dots, \phi_k(\mathbf{D}_k)$  parametrisiert wird, wenn sie folgende Form aufweist:

- $P_{\Phi}(X_1, \dots, X_n) = \frac{1}{Z} \tilde{P}_{\Phi}(X_1, \dots, X_n)$ .
- mit  $\tilde{P}_{\Phi}(X_1, \dots, X_n) = \phi_1(\mathbf{D}_1) \times \phi_2(\mathbf{D}_2) \times \dots \times \phi_m(\mathbf{D}_m)$  als nicht-normalisierter Größe.
- $Z = \sum_{X_1, \dots, X_n} \tilde{P}_{\Phi}(X_1, \dots, X_n)$  als Normalisierungskonstante bzw. Zustandssumme.

Markov-Netze gemäß obenstehender Definition sind Graphen mit ungerichteten Kanten, in denen auch Zyklen möglich sind. Faktoren sind nicht normalisiert sondern müssen durch die Zustandssumme normalisiert werden. Auch Bayes'sche Netze lassen sich als Gibbs-Verteilung abbilden ([KF09, S.134], Beweis ebd.):

**Lemma 4.1 Bayes'sches Netz als Gibbs-Verteilung**

Sei  $\mathbf{B}$  ein Bayes'sches Netz über  $\mathbf{X}$  und  $\mathbf{E} = \mathbf{e}$  bekanntes Wissen über Zustände von  $\mathbf{X}$ . Sei  $\mathbf{Q} = \mathbf{X} - \mathbf{E}$ . Dann ist  $P_{\mathbf{B}}(\mathbf{Q}|\mathbf{e})$  eine Gibbs-Verteilung definiert durch die Faktoren  $\Phi = \{\phi_{X_i}\}_{X_i \in \mathbf{X}}$  mit  $\phi_{X_i} = P_{\mathbf{B}}(X_i | El_{X_i})[\mathbf{E} = \mathbf{e}]$ . Die Zustandssumme dieser Gibbs-Verteilung ist die Wahrscheinlichkeit für das Eintreten des beobachteten Wissens  $P(\mathbf{e})$ .

Die Transformation eines Bayes'schen Netzes in ein Markov-Netz erfolgt durch „Moralisieren“. Dabei werden zum einen alle gerichteten Kanten durch ungerichtete Kanten ersetzt, zum anderen alle Elternknoten eines Kindknotens durch ungerichtete Kanten verbunden. Nach dem Moralisieren gelten folgende Aussagen (aus [KF09, S. 135], Beweis ebd.):

**Korollar 4.1 Moralisierte Bayes'schen Netz als minimale Unabhängigkeitsabbildung**

Ein moralisiertes Bayes'sches Netz  $M(G)$  ist eine minimale Unabhängigkeitsabbildung des Bayes'schen Netzes  $G$ .

**Satz 4.2 Moralisches Bayes'schen Netz als perfekte Unabhängigkeitsabbildung**

Wenn das Bayes'sche Netz  $G$  moralisch ist (d.h. alle Knoten mit gemeinsamen Nachkommen sind durch Kanten untereinander verbunden), dann ist der moralisierte Graph  $M(G)$  eine perfekte Unabhängigkeitsabbildung von  $G$ .

Anzumerken ist, dass durch das Moralisieren von Bayes'schen Netzen aus V-Strukturen re-

sultierende Unabhängigkeiten verloren gehen. Daher entsteht nur bei bereits moralischen Bayes'schen Netzen auch eine perfekte Unabhängigkeitsabbildung, bei Netzen mit V-Strukturen jedoch nur eine minimale.

Eine Sonderform von Markov-Netzen sind triangulierte oder chordale Graphen, die Voraussetzung für eine effiziente Inferenz sind. Nach [KF09, S.139] gilt die folgende Definition:

**Definition 4.10 Triangulierter Graph**

*Ein Graph ist trianguliert, wenn keine zyklische Verbindung im Graphen mit mehr als drei Kanten existiert, ohne dass eine kürzere Verbindung existierte.*

Hierdurch existieren innerhalb des Markov-Netzes vollverbundene Subgraphen, die sich zu Cliques zusammenfassen lassen, welche es wiederum erlauben, das gesamte Netz in einem Cliquenbaum darzustellen. Cliques und Cliquenbäume sind nach [KF09, S. 35 und S. 140] wie folgt definiert:

**Definition 4.11 Clique**

*Jeder vollverbundene Subgraph maximaler Größe ist eine Clique  $C$ .*

**Definition 4.12 Cliquenbaum**

*Ein Baum  $\mathcal{T}$  ist ein Cliquenbaum für ein Markov-Netz  $\mathcal{H}$ , wenn*

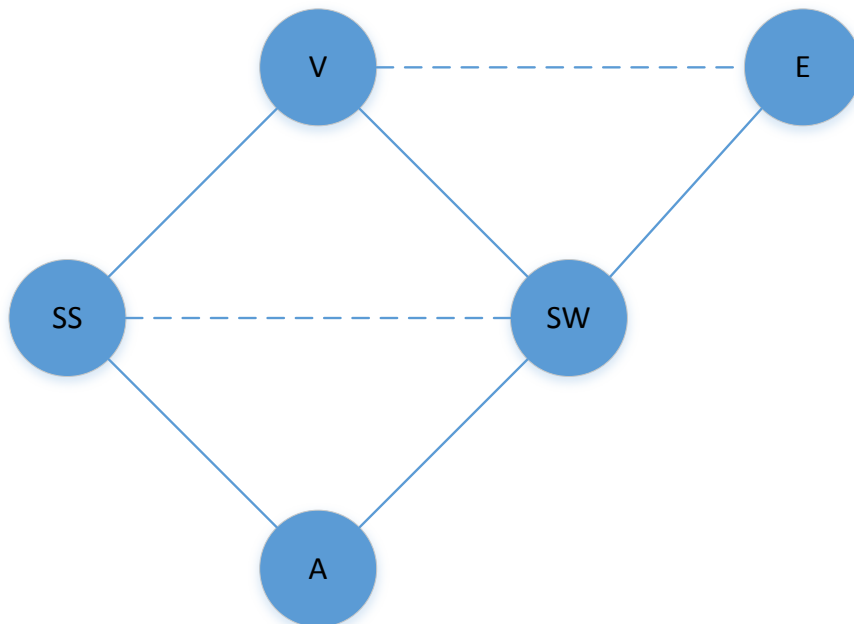
- *jeder Knoten zu einer Clique in  $\mathcal{H}$  gehört, und jede Clique in  $\mathcal{H}$  ein Knoten in  $\mathcal{T}$  ist.*
- *jedes Trennglied  $Sep_{i,j}$  den Cliquenbaum in drei Untermengen teilt:  
Die Seite vor dem Trennglied  $W_{\{<(i,j)\}} - Sep_{i,j}$ , die Seite nach dem Trennglied  $W_{\{>(i,j)\}} - Sep_{i,j}$  und das Trennglied  $Sep_{i,j}$  selbst.*

Existiert nach dem Moralisieren des Bayes'schen Netzes noch kein triangulierter Graph, kann dieser durch Einfügen von zusätzlichen Kanten erzeugt werden. Zur Bestimmung der notwendigen zusätzlichen Kanten wird Knoten für Knoten des Graphen entfernt, und alle vormalig mit dem entfernten Knoten verbundenen Knoten miteinander verbunden. Mit jeder hierdurch entstehenden zusätzlichen Kante gehen allerdings Unabhängigkeitsannahmen über die Zufallsvariablen im Graphen verloren. Ziel der Triangulation ist es daher, durch die Reihenfolge der Elimination der Knoten im Graphen Cliques minimaler Größe zu erzeugen.

Durch vollständige Induktion kann gezeigt werden, dass es zu jedem ungerichteten triangulierten Graphen  $\mathcal{H}$  einen Cliquenbaum  $\mathcal{T}$  gibt (siehe [KF09, S. 140]). Die Vorteile eines Cliquenbaums für die Inferenz in einem Bayes'schen Netz werden im folgenden Abschnitt 4.2.1.3 dargelegt. Moralisieren und Cliquenbildung werden im folgenden Beispiel verdeutlicht.

### Beispiel 4.2 Moralisiertes Beispielnetz und Cliquenbaum

Das moralisierte Beispielnetz aus Abbildung 4.3 ist in Abbildung 4.4 dargestellt. Zu erkennen sind die hinzugefügten ungerichteten Kanten zwischen den Elternknoten von  $SW$  und  $A$ .



**Abbildung 4.4:** Markov-Netz nach Moralisierung des Beispielnetzes aus Abbildung 4.3. Die gestrichelten Kanten wurden aufgrund der Moralisierung hinzugefügt

Der Cliquenbaum für das moralisierte Beispielnetz ist in Abbildung 4.5 dargestellt.



**Abbildung 4.5:** Cliquenbaum mit Cliquen (Oval) und Trenngliedern (Quadratisch)

#### 4.2.1.3 Training und Inferenz

Das Hauptziel von Wahrscheinlichkeitsnetzen ist es, die Wahrscheinlichkeit einer Variablen  $X$  bei gegebenen anderen Variablen  $\mathbf{Y}$  und Wissen  $\mathbf{E} = \mathbf{e}$  zu berechnen,  $p_x = P(X|\mathbf{Y}, \mathbf{e})$ . Im Allgemeinen ist es NP-Schwer, die gesuchte Wahrscheinlichkeitsverteilung zu bestimmen. Ohne Berücksichtigung der im Netz vorhandenen Unabhängigkeiten müssen zunächst die

Wahrscheinlichkeiten für alle Zustandskombinationen der Zufallsvariablen berechnet und alle mit dem Wissen  $e$  inkompatiblen entfernt werden. Danach müssen alle Variablen außer der gesuchten  $X$  marginalisiert werden, indem die verbliebenen Zustandskombinationen addiert werden. Am Ende erfolgt die Renormalisierung auf eine Gesamtwahrscheinlichkeit von 1. Durch Wahl einer geschickten Reihenfolge der Summation von Zustandskombinationen unter Nutzung der Unabhängigkeiten kann die Anzahl der erforderlichen Berechnungen deutlich reduziert werden. Hierzu kann das Verfahren der Variablenelimination verwendet werden [KF09, S. 299]. Bei diesem werden einzelne Zufallsvariablen eliminiert, indem über die sie enthaltenen Faktoren summiert wird. Allgemein gilt nach [KF09, S. 299]:

**Lemma 4.2 Variablenelimination**

Sei  $\mathbf{X}$  eine Menge von Variablen und sei  $\Phi$  eine Menge von Faktoren, sodass für jedes  $\phi \in \Phi$ ,  $\text{Umfang}[\phi] \subseteq \mathbf{X}$ . Sei  $\mathbf{Y} \subseteq \mathbf{X}$  eine Menge von gesuchten Variablen, und  $\mathbf{Q} = \mathbf{X} - \mathbf{Y}$ . Für jede Reihenfolge  $\mathcal{R}$  von  $\mathbf{X}$  erhält man durch die Variablenelimination  $\text{VE}(\Phi, \mathbf{Q}, \mathcal{R})$  einen Faktor  $\phi^*(\mathbf{Y})$  mit  $\phi^*(\mathbf{Y}) = \sum_{\mathbf{Q}} \prod_{\phi \in \Phi} \phi$ .

Der Effizienzgewinn soll an einem Beispiel aus [KF09, S. 293] verdeutlicht werden.

**Beispiel 4.3 Variablenelimination bei einer Kette von Zufallsvariablen**

Gegeben sei eine Kette von Zufallsvariablen  $X_1 \rightarrow X_2 \rightarrow \dots \rightarrow X_n$  mit jeweils  $o$  Zuständen pro Variable. Die Wahrscheinlichkeitsverteilung  $P(X_{i+1})$  des nächsten Kettenglieds  $X_{i+1}$  hängt dabei von den Wahrscheinlichkeiten der Zustände des direkten Vorgängers ab.

$$P(X_{i+1}) = \sum_{x_i} P(X_{i+1}|x_i)P(x_i) \tag{4.5}$$

Die Komplexität jedes einzelnen Schrittes ist  $\mathcal{O}(o^2)$ , da jede Variable  $o$  Zustände hat und für jede Kombination zweier Zustände die resultierende Wahrscheinlichkeit berechnet werden muss. Versuchte man nun, die Wahrscheinlichkeitsverteilung über  $X_n$  in einem Schritt zu berechnen, ergäben sich bei  $n$  Variablen  $o^n$  Kombinationen, die Lösung wäre der Ordnung  $\mathcal{O}(o^n)$ . Wird die Variablenelimination andererseits nacheinander für jeweils die Kombination zweier Variablen durchgeführt, reduziert sich das allgemein NP-schwere Problem zu einem Problem der Ordnung  $\mathcal{O}(n(o^2))$ .

Die Schwierigkeit bei Anwendung der Variablenelimination besteht darin, eine effiziente Eliminationsreihenfolge zu finden. Hierzu können Cliquenbäume nach Definition 4.12 genutzt werden. Faktoren, über die summiert werden kann, können über Cliquen im induzierten Graphen des zugehörigen Bayes'schen Netzes ermittelt werden. Die Definition von induzierten Graphen lautet nach [KF09, S. 308]:



**Definition 4.13 Induzierter Graph**

Sei  $\Phi$  eine Menge von Faktoren über  $\mathbf{X} = X_1, X_2, \dots, X_n$  und sei  $\mathcal{R}$  eine Eliminations-Reihenfolge für eine Untermenge  $\mathbf{Y} \subseteq \mathbf{X}$ . Der induzierte Graph  $I_{\Phi, \mathcal{R}}$  ist ein ungerichteter Graph über  $\mathbf{X}$ , bei dem  $X_i$  und  $X_j$  über eine Kante verbunden sind, falls sie gemeinsam in einem Zwischenfaktor  $\psi$  auftreten, der vom Variableneliminations-Algorithmus bei Verwendung der Eliminationsreihenfolge  $\mathcal{R}$  generiert wurde.

Der Zusammenhang von Variablenelimination und Cliques innerhalb eines induzierten Graphen wird über das folgende Theorem hergestellt (aus [KF09, S. 309], Beweis ebd.):

**Satz 4.3 Zusammenhang zwischen Faktoren und Cliques**

Sei  $I_{\Phi, \mathcal{R}}$  der induzierte Graph einer Menge von Faktoren  $\Phi$  und sei  $\mathcal{R}$  eine Eliminationsreihenfolge. Dann gilt:

1. Der Umfang jedes Faktors, der während des Variableneliminations-Prozesses entsteht, ist eine Clique in  $I_{\Phi, \mathcal{R}}$ .
2. Jede Clique in  $I_{\Phi, \mathcal{R}}$  ist der Umfang eines Zwischenfaktors während der Variablenelimination.

Aufgrund dieses Zusammenhangs kann die Suche nach der optimalen Eliminations-Reihenfolge  $R$  für die Zufallsvariablen umformuliert werden in die Suche nach der Triangulation eines Markov-Netzes  $H$ , bei der die größte Clique so klein wie möglich ist. Dies wiederum hängt von der Reihenfolge der Eliminierung der Knoten des Markov-Netzes im Zuge der Triangulation ab. Die optimale Reihenfolge zu finden, ist ebenso wie bei der Variablenelimination NP-schwer ([KF09, S. 374]). Als heuristisches Verfahren kann jedoch immer der Knoten des Netzes entfernt werden, bei dem die wenigsten zusätzlichen Kanten eingefügt werden müssen.

Die Cliques bilden dabei genau die Faktoren ab, über die summiert wird, und im Umfang der Trennglieder fehlen genau die Variablen, die bei der Variablenelimination entfernt wurden. Auf diese Weise können die zu eliminierenden Variablen und dazugehörigen Faktoren effizient bestimmt werden, um die Wahrscheinlichkeitsverteilung einer gesuchten Zufallsvariablen zu ermitteln. Im folgenden Beispiel wird dargestellt, wie der Cliquenbaum aus Beispiel 4.2 für die Bestimmung einer effizienten Eliminationsreihenfolge genutzt werden kann:

**Beispiel 4.4 Variablenelimination im Beispielnetz**

Ohne Wissen um die Unabhängigkeiten im Netz des Fahrers aus Beispiel 4.1 lautete die

*Berechnungsvorschrift zur Berechnung von  $P(A)$ :*

$$P(A) = \sum_{v,e,ss,sw} P(v, e, ss, sw, P) \quad (4.6)$$

$$= \sum_{v,e,ss,sw} P(v) * P(e) * P(ss|v) * P(sw|v, e) * P(P|ss, sw) \quad (4.7)$$

*Wird stattdessen die Reihenfolge der Variablenelimination anhand des Cliquesbaums aus Abbildung 4.5 bestimmt, kann der Berechnungsaufwand reduziert werden:*

$$P(A) = \sum_{ss,sw} P(A|ss, sw) \sum_v P(v) * P(ss|v) \sum_e P(e) * P(sw|v, e) \quad (4.8)$$

*Dabei handelt es sich bei dem Produkt der Zufallsvariablen, über die summiert wird, jeweils um Faktoren. Die Summation ist also gleichbedeutend mit der Marginalisierung von Variablen aus einem Faktor.*

Der Cliquesbaum bietet einen weiteren Vorteil gegenüber der normalen Variablenelimination. Während bei letzterer die komplette Elimination für jede gesuchte Variable neu durchgeführt werden muss, erlaubt die Inferenz mit Hilfe des Cliquesbaums, während der Inferenz für eine Variable erzeugte Informationen zu speichern und für weitere Abfragen zu nutzen. Dies gelingt mit dem sogenannten Cliquesbaum-Algorithmus. Für die Anwendung dieses Algorithmus muss der Cliquesbaum die Eigenschaften „Familienerhaltung“ und „Fortlaufende Überschneidung“ besitzen. Ihre Definition nach [KF09, S. 346f] lautet:

#### **Definition 4.14 Familienerhaltung**

*Familienerhaltung: Jeder Faktor  $\phi \in \Phi$  muss mit einer Clique  $C_i$  verbunden sein, sodass der Umfang von  $\phi \subseteq C_i$  ist. Jede Kante zwischen zwei Cliques  $C_i$  und  $C_j$  ist die Schnittmenge bzw. das Trennglied  $Sep_{i,j} \subseteq C_i \cap C_j$ .*

#### **Definition 4.15 Fortlaufende Überschneidung**

*Für jede Variable  $X$  muss gelten: wenn  $X \in C_i$  und  $X \in C_j$ , dann ist  $X$  auch in jeder Clique auf dem eindeutigen Pfad zwischen  $C_i$  und  $C_j$ .*

Im Cliquesbaum-Algorithmus werden die Faktoren der Trennglieder berechnet, indem an einem Ende des Cliquesbaums startend alle nicht im jeweiligen Trennglied enthaltenen Variablen eliminiert und das Trennglied dann mit der jeweils nächsten Clique multipliziert wird. Da hierdurch die Informationen einer Clique in die folgenden Cliques eingebracht werden, wird dieser Algorithmus auch Nachrichtenweiterleitung genannt. Eine Nachricht zwischen zwei Cliques  $C_i$  und  $C_j$  wird durch ihr Trennglied  $Sep_{i,j}$  repräsentiert, das alle Informationen aus den vorhergegangenen Cliques enthält:  $C_i \rightleftharpoons Sep_{i,j} \rightleftharpoons C_j$ .

Eine Clique sendet eine Nachricht an eine mit ihr verbundene Clique, sobald sie von allen

anderen mit ihr verbundenen Cliques Nachrichten erhalten hat. Auf diese Weise konvergieren alle Trennglieder nach einem Durchlauf, also nachdem alle Nachrichten einmal von den Start-Cliques bis zu der End-Clique und zurück geschickt wurden. Der Cliquenbaum ist dann „kalibriert“.

Um aus einem kalibrierten Cliquenbaum die Wahrscheinlichkeitsverteilung einer gesuchten Zufallsvariable zu ermitteln genügt nun eine Clique, die diese Variable enthält. Aus dieser müssen alle nicht gesuchten Variablen marginalisiert und die Clique danach normalisiert werden. Die Effizienz dieses Algorithmus beruht darauf, dass zur vollständigen Kalibrierung bei insgesamt  $n$  Cliques nur  $2(n - 1)$  Nachrichten geschickt werden müssen. Es wird also nur die doppelte Anzahl an Rechenschritten benötigt wie für einen einfachen Durchlauf des Variableneliminations-Algorithmus.

Auch Wissen kann in eine Clique eingebracht und mit Hilfe des Cliquenbaum-Algorithmus an die Cliques mit den gesuchten Variablen weiterverteilt werden. Nachteilig auf den Berechnungsaufwand gegenüber einfacher Variablenelimination kann sich die Forderung nach der Eigenschaft der fortlaufenden Überschneidung auswirken. Hierdurch sind Cliques und damit zu marginalisierende Faktoren und die Trennglieder zwischen ihnen je nach Struktur des ursprünglichen Netzes unter Umständen größer, als für die Bestimmung einer Variable erforderlich.

## 4.2.2 Modellierung von Spurwechselentscheidungen

Das Training und die Anwendung von Fahrermodellen zur Simulation von Spurwechselmanövern beruhen auf Umgebungsdaten, die von der Fahrzeugsensorik erfasst wurden. Diese sind durch ihre Verfügbarkeit, Reichweite und fehlerhafte Detektionen eingeschränkt. Das für die Fahrermodellierung vorgeschlagene Bayes'sche Netz nutzt daher die in Tabelle 4.1 aufgelisteten Eingangsdaten.

Durch diese Daten steht in jedem Zeitschritt Wissen über die zugehörigen Sensor-Zufallsvariablen des Netzes zur Verfügung. Diese bilden damit gewissermaßen die Eingangsschicht des Bayes'schen Netzes. Sie sind über Kanten mit einer Zwischenschicht verbunden, in der die in Tabelle 4.2 aufgelisteten Bewertungskriterien für jede Spur abgebildet werden. Diese wiederum sind pro Spur mit einem Knoten verbunden, welcher die Gesamtzufriedenheit mit der Spur abbildet. Diese Spur-Gesamtzufriedenheitsknoten sind mit dem finalen Knoten des Netzes verbunden, der die drei Zustände Spurwechsel Rechts, Spurwechsel Links und Spurhalten enthält. Die Zufallsvariablen, die durch die Knoten der Zwischen- und Endschichten dargestellt werden, sind in Tabelle 4.2 aufgelistet.

Die Modellierung des Spurwechsels ist an die Modellierung in der Verkehrssimulation PELOPS (vgl. Abschnitt 3.3.6.2) angelehnt. Anstatt deterministischer Faktoren werden jedoch Zufallsvariablen genutzt.

Die für die Modellierung der Spurwechselentscheidung vorgeschlagene Netzstruktur ist in Abbildung 4.6 dargestellt.

**Tabelle 4.2:** Zufallsvariablen mit Einfluss auf die Spurwechselentscheidung

Variable	Beschreibung
$v_{\text{red}}$	Wunschgeschwindigkeit des zu simulierenden Fahrers reduziert um Informationen aus der Straßentopologie, die zu Geschwindigkeitsreduktion führen
$c_{\text{pre}}$	Zufriedenheit mit vorausfahrendem Fahrzeug
$c_{\text{suc}}$	Zufriedenheit mit folgendem Fahrzeug
$c_{\text{kr}}$	Zufriedenheit mit Abweichung vom Rechtsfahrgebot
$v_{\text{exp}}^{\text{lane}}$	Erwartete allgemeine Geschwindigkeit in der Spur
$v_{\text{exp}}$	Erwartete eigene Geschwindigkeit in der Spur
$s_{\text{suf}}$	Hinreichender Geschwindigkeitsunterschied der Spur zur rechten Spur
$c_{\text{vel}}$	Zufriedenheit mit durchschnittlicher Geschwindigkeit der Spur
$c$	Gesamtzufriedenheit mit eigener Spur
$c_{\text{R}}, c_{\text{L}}$	Gesamtzufriedenheit mit rechter und linker Nachbarspur
LCI	Spurwechselentscheidung

Die Eingänge der Knoten  $c_{\text{R}}$  und  $c_{\text{L}}$  sind äquivalent zu denen von  $c$  und daher nicht abgebildet. Knoten am Anfang des Netzes, welche die in Tabelle 4.1 aufgelisteten Sensordaten abbilden, sind in Gruppen wie folgt zusammengefasst:

$$\mathbf{f}_{c_{\text{pre}}} = \{T_{\text{ittc},V1} \ T_{\text{ittc},V2} \ T_{\text{itgap},V1} \ e_0^{\text{lane}}\}, \quad (4.9)$$

$$\mathbf{f}_{c_{\text{suc}}} = \{T_{\text{itgap},V1} \ T_{\text{itgap},V2} \ T_{\text{itgap},F1} \ T_{\text{ittc},F1} \ e_0^{\text{lane}} \ e_{\text{R}}^{\text{lane}} \ w, \ e_{V1}^{\text{obj}} \ e_{V2}^{\text{obj}} \ e_{F1}^{\text{obj}}\}, \quad (4.10)$$

$$\mathbf{f}_{c_{\text{kr}}} = \{T_{\text{itgap},V1} \ T_{\text{itgap},V2} \ T_{\text{itgap},F1} \ T_{\text{ittc},F1} \ e_0^{\text{lane}} \ e_{\text{R}}^{\text{lane}} \ w, \ e_{V1}^{\text{obj}} \ e_{V2}^{\text{obj}} \ e_{F1}^{\text{obj}}\}, \quad (4.11)$$

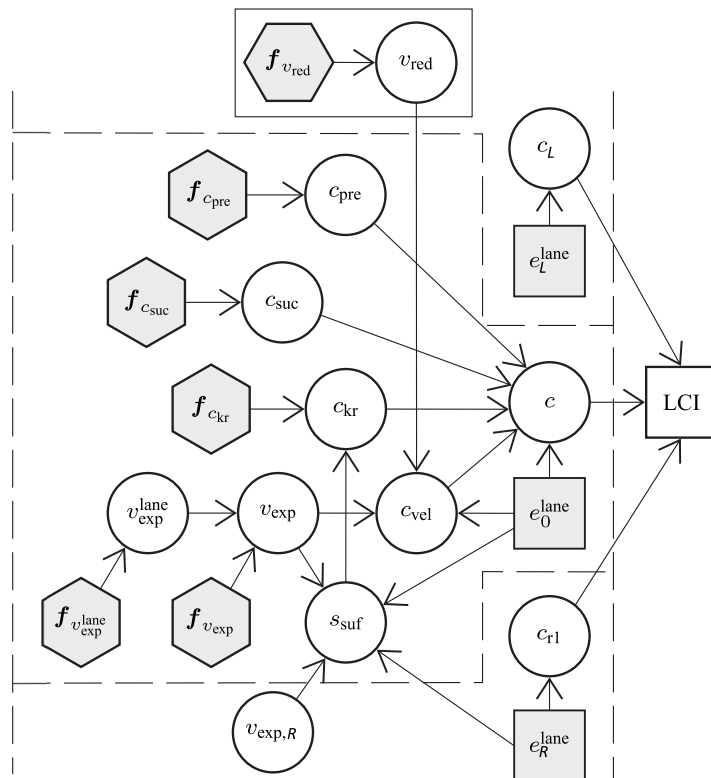
$$\mathbf{f}_{v_{\text{exp}}} = \{v_s \ e_0^{\text{lane}}\}, \quad (4.12)$$

$$\mathbf{f}_{v_{\text{exp}}^{\text{lane}}} = \{v_{V1}^{\text{obj}} \ v_{V2}^{\text{obj}} \ a_{V1}^{\text{obj}} \ a_{V2}^{\text{obj}} \ d_{V1}^{\text{obj}} \ d_{V2}^{\text{obj}} \ e_{V1}^{\text{obj}} \ e_{V2}^{\text{obj}} \ e_0^{\text{lane}}\}, \quad (4.13)$$

$$\mathbf{f}_{v_{\text{red}}} = \{cur \ cur_{\text{max}} \ v_{\text{wunsch}}\}. \quad (4.14)$$

Im in Abbildung 4.6 dargestellten Netz zur Spurwechselmodellierung werden unterschiedliche Darstellungsarten für die Knoten verwendet. Im Fall von rechteckigen Knoten handelt es sich um Knoten mit diskreten Zustandsvariablen, bei runden Knoten um solche mit kontinuierlichen Zufallsvariablen. Die in Gruppen zusammengefassten Eingangsknoten des Netzes beinhalten sowohl kontinuierliche als auch diskrete Zufallsvariablen und sind als Hexagon dargestellt.

In der bisherigen Beschreibung der Bayes'schen Netze wurden stets diskrete Knoten mit einer begrenzten Anzahl an Zuständen vorausgesetzt. Dies hat im Fall von kontinuierlichen Daten wie z. B. Geschwindigkeit oder Abstand den Nachteil, dass eine feine Diskretisierung gewählt werden müsste um Unterschiede hinreichend genau abbilden zu können. Dies würde zu sehr vielen Zuständen führen und Inferenz im Netz aufgrund des erhöhten Berechnungsaufwands erschweren.



**Abbildung 4.6:** Bayes'sches Netz zur Spurwechselmodellierung. Die Knoten im gestrichelten Kasten gelten für die Spurzufriedenheit der Egospur  $c$ . Sie existieren in äquivalenter Form für die Spurzufriedenheiten der linken und rechten Spur,  $c_{l1}$  und  $c_{r1}$ , wurden aber der Übersichtlichkeit halber nicht mit dargestellt

Statt diskreten Knoten können auch kontinuierliche Knoten in Bayes'schen Netzen mit wenigen Einschränkungen auf gleiche Weise verwendet werden, um den oben beschriebenen Nachteil zu vermeiden. Die für die Nutzung des für die Spurwechselmodellierung vorgeschlagenen Bayes'schen Netzes notwendigen Anpassungen werden im folgenden Abschnitt erläutert. Die grundlegenden Beschreibungen basieren dabei auf [KF09], die notwendigen Anpassungen für die Nutzung von diskreten Knoten mit kontinuierlichen Elternknoten auf [LSK01] und [Ler02].

### 4.2.3 Hybride Bayes'sche Netze

Knoten in hybriden Bayes'schen Netzen können sowohl diskrete als auch kontinuierliche Wahrscheinlichkeitsverteilungen enthalten. Abhängig davon, welche Eltern- und Kindknoten ein Knoten hat, sind verschiedene Repräsentationsformen für die Wahrscheinlichkeitsverteilungen eines Knotens im hybriden Bayes'schen Netz geeignet.

### 4.2.3.1 Wahrscheinlichkeitsverteilungen

Wahrscheinlichkeitsverteilungen sind die fundamentalen Komponenten von probabilistischen Modellen wie z. B. Bayes'schen Netzen oder Markov-Netzen. Sie geben die Wahrscheinlichkeit für einen durch die Verteilung abgebildeten Zustand oder Wert an. Alle in dieser Arbeit genutzten Wahrscheinlichkeitsverteilungen sind in Abhängigkeit ihrer Elternknoten in Tabelle 4.3 aufgelistet.

**Tabelle 4.3:** Wahrscheinlichkeitsverteilungen abhängig von den Verteilungen ihrer Elternknoten. LG: lineare Gaußverteilung, CLG: Conditional Linear Gaussian

Knoten/Elternknoten	Diskret	Kontinuierlich	Diskret und Kontinuierlich
Diskret	Tabelle	Softmax	Softmax
Kontinuierlich	CLG	LG	CLG

Im Fall von **diskreten Verteilungen** wird üblicherweise die Wahrscheinlichkeitstabelle verwendet, welche eine Wahrscheinlichkeit  $p_q$  für jeden Zustand  $q$  einer Zufallsvariable  $Q$  angibt:

$$P(Q = q) = p_q \quad (4.15)$$

Diskrete Verteilungen können beispielsweise genutzt werden, wenn diskrete Variablen wie die Existenz eines Fahrzeugs probabilistisch angegeben werden soll.

**Kontinuierliche Verteilungen** werden i.d.R. als lineare Gaußverteilung modelliert, durch die eine Wahrscheinlichkeitsdichte für jeden Wert  $v$  der Zufallsvariable  $\Upsilon$  in Abhängigkeit des Mittelwerts  $\mu$  und der Standardabweichung  $\sigma$  angegeben werden kann:

$$P(\Upsilon = v) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(v - \mu)^2}{2\sigma^2}\right) \quad (4.16)$$

Im Gegensatz zur Summation bei diskreten Verteilungen wird eine kontinuierliche Variable durch Integration marginalisiert. Kontinuierliche Verteilungen können für kontinuierliche Variablen wie beispielsweise dem Abstand zu einem anderen Fahrzeug genutzt werden.

**Kontinuierliche Verteilungen mit diskreten und kontinuierlichen Elternknoten** können als Wahrscheinlichkeitstabelle von Normalverteilungen repräsentiert werden, im englischen „Conditional Linear Gaussian“ (CLG) genannt. Sie können je nach vorzunehmenden Rechenoperationen entweder in Momentenform wie in Gleichung 4.17 oder in kanonischer Form (Gleichung 4.18) dargestellt werden. Die Wahrscheinlichkeiten der Normalverteilungen werden durch die Gewichtsmatrix  $W_q$  repräsentiert.

$$p(\mathbf{v}|\boldsymbol{\xi}, Q = q) = (2\pi)^{-\frac{d}{2}} \Sigma_q^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{v} - W_q\boldsymbol{\xi} - \boldsymbol{\mu}_q)^T \Sigma_q^{-1} (\mathbf{v} - W_q\boldsymbol{\xi} - \boldsymbol{\mu}_q)\right) \quad (4.17)$$

mit  $d = \dim(v)$ ,  $\xi$  als Eingangsvektor der kontinuierlichen Elternknoten  $\Xi$ , der durch die Gewichtsmatrix  $W_q$  gewichtet wird und  $q$  als Zustandskombination der diskreten Elternknoten  $Q$ . Die Darstellung in kanonischer Form lautet:

$$p(v|\xi, Q = q) = \exp\left(-\frac{1}{2}(v - W_q\xi)^T K_q (v - W_q\xi) + \mathbf{h}_q^T (v - W_q\xi) + g_q\right) \quad (4.18)$$

mit  $K_q = \Sigma_q^{-1}$ ,  $\mathbf{h}_q = \Sigma_q^{-1} \boldsymbol{\mu}_q$  und  $g_q = -\frac{1}{2} \boldsymbol{\mu}_q^T \Sigma_q^{-1} \boldsymbol{\mu}_q - \log\left((2\pi)^{\frac{d}{2}} |\Sigma_q|^{\frac{1}{2}}\right)$ .

Marginalisierung von diskreten Zuständen aus CLGs erfolgt, indem die zu den zu marginalisierenden Zuständen gehörenden Gauß-Verteilungen zu einer Verteilung kollabiert werden. Mittelwert und Varianz lassen sich dabei exakt berechnen.

$$\mu_{Kol} = \sum_i w_i \mu_i \quad (4.19)$$

$$\Sigma_{Kol} = \sum_i w_i \Sigma_i + \sum_i w_i (\mu_i - \mu)(\mu_i - \mu)^T \quad (4.20)$$

Dabei sind die Summengewichte  $w_i$  die Wahrscheinlichkeiten  $P(Q = q)$  der einzelnen Zustände der diskreten Zufallsvariablen  $Q$ , die marginalisiert werden sollen. Eine beispielhafte Darstellung dieses Vorgangs findet sich in Abbildung 4.7.

**Diskrete Wahrscheinlichkeitsverteilungen mit diskreten und kontinuierlichen Elternknoten** können mit Hilfe der Softmax-Funktion beschrieben werden. Diese Funktion gibt einen Wahrscheinlichkeitswert für einen ihrer diskreten Zustände  $A = a$  zurück. Dieser hängt vom kontinuierlichen Eingangsvektor  $\xi$  ab, der mit einem vom Zustand der diskreten Elternknoten  $Q = q$  abhängigen Gewichtsvektor  $w_{i,q}$  multipliziert wird ( $i \in A$ ):

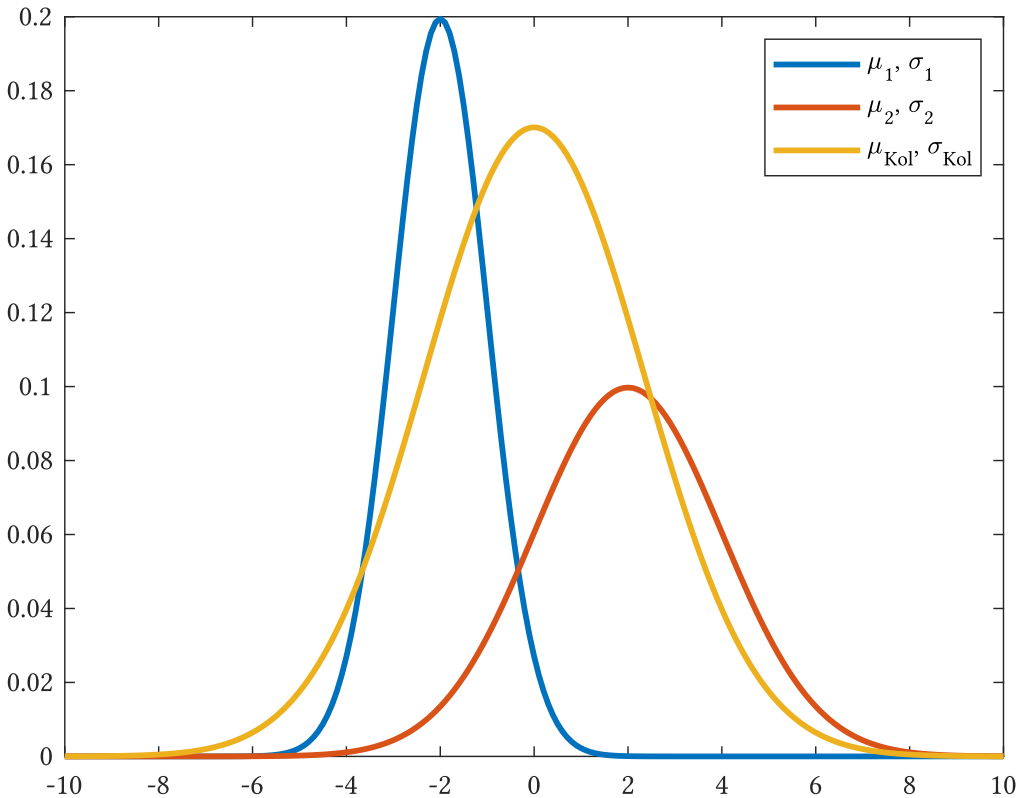
$$P(A = a|\xi, Q = q) = \frac{\exp(\mathbf{w}_{a,q}^T \xi)}{\sum_{i \in A} \exp(\mathbf{w}_{i,q}^T \xi)}. \quad (4.21)$$

#### 4.2.4 Inferenz in hybriden Netzen

Gegenüber der im ersten Teil des Abschnitts beschriebenen Inferenz für diskrete Bayes'sche Netze besteht der Hauptunterschied der Inferenz in hybriden Netzen darin, dass Wahrscheinlichkeitswerte von Knoten, deren Wahrscheinlichkeitsverteilung durch eine Softmax-Funktion gegeben ist, nur für explizite Werte ihrer kontinuierlichen Elternknoten berechnet werden können. Eine Berechnung ist nicht möglich, wenn kein Wissen oder nur Wissen über Mittelwert und Standardabweichung dieser Elternknoten vorliegt.

Grund hierfür ist, dass keine gemeinsame Wahrscheinlichkeitsverteilung existiert, durch die ein Faktor dargestellt werden kann, der sowohl die kontinuierliche Verteilung als auch die Softmax-Funktion enthält - im Gegensatz z.B. zum Fall eines kontinuierlichen Knotens mit diskreten Elternknoten, der durch die CLG-Verteilung repräsentiert werden kann.

Daher kann Inferenz für einen Softmax-Knoten, der eine Spurwechselentscheidung modelliert,



**Abbildung 4.7:** Beispielabbildung für das Kollabieren von zwei linearen Gaussverteilungen auf eine. Grafik mit  $\mu_1 = -2, \sigma_1 = 1, \mu_2 = 2, \sigma_2 = 2, w_{1/2} = 0.5$

mit dem bisherigen Cliquesbaum-Algorithmus nicht durchgeführt werden. Die notwendigen Änderungen und Erweiterungen des Cliquesbaum-Algorithmus wurden erstmals in [LSK01] und [Ler02] veröffentlicht und bestehen aus den Schritten „Aufbau des Cliquesbaums“, „Initialisierung der Cliques“, „Erste Kalibrierung“, „Einfügen der Softmax-Knoten“ und „Zweite Kalibrierung“. Diese werden im Folgenden dargelegt.

#### 4.2.4.1 Aufbau des Cliquesbaums

Aufgrund der hybriden Struktur muss neben der Anzahl der hinzuzufügenden Kanten (vgl. Abschnitt 4.2.1.3) bei der Eliminationsreihenfolge ein weiteres Kriterium beachtet werden: Bei der Marginalisierung von diskreten Zufallsvariablen aus Cliques mit diskreten und kontinuierlichen Variablen, repräsentiert durch ein CLG, entsteht ein Informationsverlust. Dieser resultiert aus dem Kollabieren der Normalverteilungen verschiedener diskreter Zustände zu einer gemeinsamen Normalverteilung. Bei diesem Vorgang können wie weiter oben beschrieben nur Mittelwert und Varianz exakt berechnet werden. Um die in dieser Approximation begründete Ungenauigkeit nicht im Zuge der Kalibrierung auf die folgenden Cliques zu übertragen,



wird die Marginalisierung von diskreten Knoten erst am Ende des Cliquenbaums durchgeführt. Dieser letzte Knoten des Cliquenbaums heißt dann „starke Wurzel“ [LSK01], da er alle diskreten Zustände des Cliquenbaums bzw. des transformierten Bayes'schen Netzes enthält. Hierfür ist es erforderlich die Eliminationsreihenfolge so zu wählen, dass alle kontinuierlichen Knoten entfernt werden, bevor der erste diskrete Knoten eliminiert wird.

#### 4.2.4.2 Initialisierung der Cliquen

Nachdem der Cliquenbaum mit seinen Knoten und Trenngliedern aufgebaut wurde, wird eventuell vorhandenes Wissen in die Zufallsvariablen eingebracht, indem mit dem Wissen inkompatible Zustände entfernt werden. Im Fall der Spurwechselmodellierung liegt Wissen in erster Linie in Form der Sensordaten vor. Im Trainingsfall mit Realdaten ist auch die Ausprägung der Spurwechsel-Variable LCI bekannt. Beim Training mit Simulationsdaten liegt zusätzlich Wissen für die ansonsten immer versteckten Wahrscheinlichkeitsverteilungen im Innern des Netzes vor.

Danach werden die Wahrscheinlichkeitsverteilungen in eine der sie enthaltenen Cliquen eingebracht, indem ihr Faktor mit dem der Clique multipliziert wird. Hierdurch wird für jede Clique ein Faktor erzeugt, der alle beinhalteten Wahrscheinlichkeitsverteilungen darstellen kann. Im konkreten Fall ist dies in der Regel ein CLG, da in den Cliquen sowohl kontinuierliche als auch diskrete Wahrscheinlichkeitsverteilungen enthalten sind.

Während der Initialisierung werden die Zufallsvariablen in der Reihenfolge, in der sie beim Aufbau des Cliquenbaums eliminiert wurden, in die Cliquen eingebracht, damit die Cliquen zu diesem Zeitpunkt so wenige diskrete Zustände wie möglich enthalten. Dies reduziert den Berechnungsaufwand, da die Anzahl der Zustände in einer Clique mit jeder diskreten Zufallsvariable exponentiell steigt.

#### 4.2.4.3 Erste Kalibrierung

Bei hybriden Bayes'schen Netzen ist es erforderlich, die Kalibrierung mit Hilfe des Cliquenbaum-Algorithmus zweifach durchzuführen, da Softmax-Knoten erst in die Cliquen integriert werden können, wenn diese die Informationen aller Elternknoten des Softmax-Knotens enthalten. Dies wird dadurch garantiert, dass durch einen ersten Lauf des Algorithmus alle Wahrscheinlichkeitsverteilungen außer der der Softmax-Knoten in eine Clique integriert und durch den Algorithmus auf alle anderen verteilt werden. Sodann werden die Softmax-Knoten in eine passende Clique integriert und der Cliquenbaum-Algorithmus erneut ausgeführt.

Der Cliquenbaum-Algorithmus beginnt dabei an den Cliquen, die am weitesten von der starken Wurzel entfernt sind, den „Blättern“ des Baums, und wird sukzessive in Richtung der Wurzel durchgeführt (Hinlauf). Sobald alle Nachrichten an der starken Wurzel angekommen sind, sendet diese unter Berücksichtigung der Informationen aus anderen Teilen des Baums Nachrichten in Richtung der Blätter zurück (Rücklauf). Beim Weiterleiten einer Nachricht  $\phi_{C_1 \rightarrow C_2}$  von einer Clique  $C_1$  zu einer Clique  $C_2$  werden alle Zufallsvariablen, die nicht in beiden Cliquen vorkommen, durch Summation (bei diskreten Variablen wie z.B. Existenz eines Fahrzeugs) oder

Integration (bei kontinuierlichen Variablen wie z.B. Abstand zu einem anderen Fahrzeug) aus  $\phi(C_1)$  marginalisiert, sodass das Trennglied  $Sep_{C_1, C_2}$  entsteht:

$$Sep_{C_1, C_2} = C_1 \cap C_2 \quad (4.22)$$

$$\phi_{C_1 \rightarrow C_2}(Sep_{C_1, C_2}) = \sum_{C_1 - Sep_{C_1, C_2}} \phi(C_1). \quad (4.23)$$

Das Trennglied wird mit dem Faktor der Clique  $C_2$  multipliziert, sodass alle Informationen aus  $C_1$  mit  $C_2$  geteilt werden. Um Rückkopplungen zu vermeiden wird von einer Clique erhaltene Information nie an diese zurückgesendet, sondern vor dem Senden wieder entfernt:

$$\phi(C_{2, \text{neu}}) = \phi_{C_1 \rightarrow C_2}(Sep_{C_1, C_2}) \times \phi(C_2) \quad (4.24)$$

$$\phi(C_{1, \text{neu}}) = \phi(C_{2, \text{neu}}) \times \frac{\phi_{C_2 \rightarrow C_1}(Sep_{C_2, \text{neu}, C_1})}{\phi_{C_1 \rightarrow C_2}(Sep_{C_1, C_2})}. \quad (4.25)$$

*Anmerkung:* Im ersten Durchgang ist dies nur für den Rücklauf erforderlich, da beim Hinlauf alle Cliques noch ihre ursprüngliche Verteilung beinhalten.

#### 4.2.4.4 Integration der Softmax-Verteilungen

Nach der ersten Kalibrierung enthalten alle Cliques die korrekten bedingten Wahrscheinlichkeitsverteilungen hinsichtlich der im Netz verfügbaren Informationen. Jetzt können die Softmax-Zufallsvariablen in die Cliques eingefügt werden, aus denen sie vorher entfernt wurden. Da nach der ersten Kalibrierung bereits eine Gauß-Verteilung für jede diskrete Zustandsverteilung in den Cliques existiert, kann nun die Kombination aus Gauß-Verteilung und Softmax berechnet und die beiden Verteilungen kombiniert werden.

Da keine konkreten Werte sondern nur die kontinuierlichen Verteilungen vorliegen, ist dies nicht direkt möglich. Stattdessen wird wie in [LSK01] vorgeschlagen das Produkt aus der Gauß- und der Softmaxverteilung als neue Gauß-Verteilung approximiert. Wie die Marginalisierung diskreter Variablen aus CLGs ist diese Approximation exakt für die ersten beiden Momente.

Für die Approximation wird für jeden Zustand der Softmax-Verteilung  $A = a$  bei gegebener Zustandskombination der diskreten Elternknoten  $Q = q$  die Wahrscheinlichkeit berechnet, indem über die zugehörigen kontinuierlichen Variablen  $\xi$  integriert wird:

$$P(A = a | Q = q) = \int_{-\infty}^{\infty} P(A = a | \xi, Q = q) P(\xi | Q = q) d\xi. \quad (4.26)$$

Damit können die Mittelwerte und Varianzen ebenfalls durch Integration berechnet werden:

$$\boldsymbol{\mu}_q = \frac{1}{P(A = a|Q = q)} \int_{-\infty}^{\infty} \boldsymbol{\xi} P(A = a|\boldsymbol{\xi}, Q = q) P(\boldsymbol{\xi}) d\boldsymbol{\xi} \quad (4.27)$$

$$\Sigma_q = \frac{1}{P(A = a|Q = q)} \int_{-\infty}^{\infty} (\boldsymbol{\xi} - \boldsymbol{\mu}_q)(\boldsymbol{\xi} - \boldsymbol{\mu}_q)^T P(A = a|\boldsymbol{\xi}, Q = q) P(\boldsymbol{\xi}) d\boldsymbol{\xi}. \quad (4.28)$$

mit  $\boldsymbol{\mu}_q$  von der Dimension  $\dim(\boldsymbol{\xi})$  und  $\Sigma_q$  von Dimension  $\dim(\boldsymbol{\xi}) \times \dim(\boldsymbol{\xi})$ . Da es für diese drei Integrale keine geschlossene Lösung gibt, müssen sie durch numerische Integration gelöst werden. Die Gauß-Hermite-Integration approximiert Integrale der Form

$$\int_{-\infty}^{\infty} e^{-\zeta^2} f(\zeta) d\zeta. \quad (4.29)$$

Da sich die Gaußverteilung leicht in die Form  $e^{-\zeta^2}$  umwandeln lässt, eignet sich diese Approximation gut für die gegebene Problemstellung.

Bei Integrationsverfahren der Art Gauß-Quadratur wird die Funktion an definierten Integrationspunkten  $h_i$  ausgewertet, mit vorher berechneten Gewichten  $w_i$  gewichtet und aufsummiert. Gewichte und Integrationspunkte können berechnet oder aus der Literatur ermittelt werden (z.B. [Haz02]). Um die Gaußverteilung in die für die Gauß-Hermite-Integration notwendige Form  $e^{-\zeta^2}$  zu transformieren, müssen die Gewichte einmalig für beliebige Gaußverteilungen mit  $\frac{1}{\sqrt{h}}$  multipliziert werden. Die Integrationspunkte müssen für jede Verteilung individuell berechnet werden mit

$$\mathbf{h}_{\text{neu}} = \sqrt{2}\Sigma^{1/2} \mathbf{h}_{\text{alt}} + \boldsymbol{\mu}. \quad (4.30)$$

Für diese Punkte wird nun die Softmax-Funktion  $f(\mathbf{h}_{\text{neu}}) = P(A = a|\mathbf{h}_{\text{neu}}, Q = q)$  einmalig ausgewertet und kann dann zur Berechnung aller Integrale (4.26), (4.27) und (4.28) einer Zustandskombinationen  $Q = q$  genutzt werden. Hierdurch lassen sich die ersten beiden Momente (Mittelwerte und Varianzen) exakt berechnen. Durch die Kombination aus Softmax und Normalverteilung entstehende weitere Momente werden nicht mehr korrekt wiedergegeben, da das Produkt wieder als lineare Gaußverteilung repräsentiert wird.

Die Anzahl der notwendigen Berechnungen steigt exponentiell mit der Dimension des Integrals und daraus folgend direkt mit der Anzahl der kontinuierlichen Eltern des Softmax-Knotens an. Die Anzahl der auszuwertenden diskreten Zustandskombinationen ist das Produkt der Anzahl an möglichen Zuständen des Softmax-Knotens  $A$  und der Anzahl an diskreten Elternknoten des Softmax-Knotens sowie aller seiner kontinuierlichen Eltern.

Nachdem alle Approximationen für alle Softmax-Knoten durchgeführt wurden, sind diese vollständig in den Cliquesbaum integriert.

#### 4.2.4.5 Zweite Kalibrierung

Um die im vorangegangenen Schritt approximierten Wahrscheinlichkeitsverteilungen über den gesamten Cliquesbaum zu verteilen, wird der Cliquesbaum-Algorithmus erneut ausgeführt. Hierbei muss von einem Knoten empfangene Information sowohl beim Hin- als auch Rücklauf

entfernt werden um Rückkopplung zu vermeiden (vgl. (4.25)). Nach Durchlauf des Algorithmus kann jede Clique, die die entsprechenden Variablen enthält, zur Inferenz genutzt werden.

Im konkreten Fall der Spurwechselmodellierung ist die Zufallsvariable für den Spurwechsel in der starken Wurzel des Cliquenbaums enthalten, da es sich um eine diskrete Variable handelt. Da auch nur an dieser Stelle die Approximation der Kombination von Softmaxverteilung und Gaußverteilungen erfolgt, ist in diesem Netz eine zweite Kalibrierung für die Inferenz nicht erforderlich. Da der Faktor der starken Wurzel bereits alle notwendigen Informationen enthält, kann direkt durch Marginalisierung aller anderen Variablen aus dem Faktor die Wahrscheinlichkeitsverteilung der Zufallsvariable für den Spurwechsel berechnet werden. Für das Training der Wahrscheinlichkeitsverteilungen des Bayes'schen Netzes müssen hingegen alle beschriebenen Schritte durchgeführt werden.

#### 4.2.5 Training von hybriden Bayes'schen Netzen

Da weder für das allgemeine Fahrermodell, noch für die einzelnen individualisierten Fahrermodelle, die Parameter der Wahrscheinlichkeitsverteilungen bekannt sind, werden sie in der vorliegenden Arbeit über ein Training des Bayes'schen Netzes ermittelt. Hierzu werden auf Testfahrten gesammelte und in der Verkehrssimulation generierte Daten verwendet.

Das in Abbildung 4.6 gegebene Netz hat eine vorab definierte Netzstruktur und mindestens teilweise unbekannte Parameter. Zur Bestimmung der Wahrscheinlichkeitsverteilungen kann der Expectation-Maximization (EM) Algorithmus genutzt werden, der erstmals in [DLR77] vorgestellt wurde.

Zunächst werden alle Wahrscheinlichkeitsverteilungen, deren Parameter unbekannt sind, mit zufälligen Werten initialisiert. Danach wird im Expectation-Schritt die Wahrscheinlichkeitsverteilung für alle versteckten Knoten (also Knoten über deren Zustand kein Wissen  $e$  bekannt ist) für jeden Zeitpunkt  $t$  auf die im vorangegangenen Abschnitt beschriebene Weise inferiert. Im Maximization-Schritt werden dann die Parameter so verändert, dass die Plausibilitätsfunktion

$$\Lambda = \prod_t P(e_t) \quad (4.31)$$

maximal wird.

Dabei gibt die Funktion (4.31) die Plausibilität an, mit der das gegebene BN die inferierten Wahrscheinlichkeitsverteilungen erzeugt hat. Für diesen Schritt existieren geschlossene Lösungen für diskrete Verteilungen und Gaußverteilungen, während für Softmax-Knoten auch hier wieder auf numerische Integration zurückgegriffen werden kann. Die Lösungen werden im folgenden Abschnitt dargestellt. Nachdem die Parameter optimiert wurden, werden die beiden Schritte Expectation und Maximization solange wiederholt, bis eine Konvergenz der Plausibilität feststellbar ist. Hierzu wird die Differenz der Plausibilität von zwei aufeinanderfolgenden Iterationen  $i$  und  $i + 1$  überprüft. Ist sie kleiner als ein vorher definierter Wert  $\epsilon > \Lambda_{i+1} - \Lambda_i$ , bricht die Parameteroptimierung ab und die Parameter der letzten Iteration werden weiter genutzt.

#### 4.2.5.1 Lernen der Parameter von diskreten Verteilungen und Gaußverteilungen

Parameter diskreter Verteilungen, die zur maximalen Plausibilität des BN führen, können direkt ermittelt werden, indem die *a posteriori* Wahrscheinlichkeiten der einzelnen Zustände *nach* der Inferenz über alle Daten  $\mathbf{e}$  zu den Zeitpunkten  $t$  gemittelt wird ([TVCG16], [KF09, S. 724]):

$$P(\xi) = \frac{\sum_{t=1}^n P(\xi|\mathbf{e}_t)}{n}. \quad (4.32)$$

Für Gaußverteilungen wird in [Mur03] eine Methode zur Maximierung vorgeschlagen. Dabei werden die neuen Parameter der Gaußverteilungen über den Umweg der suffizienten Statistik (vgl. [Mur12, S. 74]) anhand ihrer *a posteriori* Wahrscheinlichkeitsverteilungen berechnet:

$$w_{q,t} = P(Q = q|\mathbf{e}_t) \quad (4.33)$$

$$w_q = \sum_t w_{q,t}$$

$$S_{\xi,q} = \sum_t w_{q,t} \boldsymbol{\mu}_{\xi,q,t} \quad (4.34)$$

$$S_{\mathbf{v},q} = \sum_t w_{q,t} \boldsymbol{\mu}_{\mathbf{v},q,t}$$

$$S_{\xi\xi,q} = \sum_t w_{q,t} \Sigma_{\xi,q,t} \boldsymbol{\xi}_{q,t} (\boldsymbol{\mu}_{\xi,q,t} \boldsymbol{\mu}_{\xi,q,t}^T) \quad (4.35)$$

$$S_{\mathbf{v}\mathbf{v},q} = \sum_t w_{q,t} \Sigma_{\mathbf{v},q,t} \mathbf{v}_{q,t} (\boldsymbol{\mu}_{\mathbf{v},q,t} \boldsymbol{\mu}_{\mathbf{v},q,t}^T)$$

$$S_{\xi\mathbf{v},q} = \sum_t w_{q,t} \Sigma_{\xi,q,t} \mathbf{v}_{q,t} (\boldsymbol{\mu}_{\xi,q,t} \boldsymbol{\mu}_{\mathbf{v},q,t}^T)$$

mit  $S_{\xi,q}$  und  $S_{\mathbf{v},q}$  als den Suffizienz-Mittelwerten und  $S_{\xi\xi,q}$ ,  $S_{\mathbf{v}\mathbf{v},q}$  und  $S_{\xi\mathbf{v},q}$  als Suffizienz-Kovarianzen.

Um hieraus die für Gaußverteilungen (vgl. (4.17)) notwendigen Parameter  $\boldsymbol{\mu}_q$ ,  $\Sigma_q$  und  $B_q$  zu ermitteln, werden  $\boldsymbol{\mu}_q$  und  $B_q$  gemeinsam berechnet und dann zur Bestimmung von  $\Sigma_q$  genutzt.

$$S_{zz,q} = \begin{bmatrix} S_{\xi\xi,q} & S_{\xi,q} \\ S_{\xi,q}^T & u_q \end{bmatrix}, \quad S_{z\mathbf{v},q} = \begin{bmatrix} S_{\xi\mathbf{v},q} \\ S_{\mathbf{v},q}^T \end{bmatrix} \quad (4.36)$$

$$A_q = \begin{bmatrix} \hat{B}_q & \hat{\boldsymbol{\mu}}_q \end{bmatrix} = S_{z\mathbf{v},q}^T S_{zz,q}^{-1} \quad (4.37)$$

$$\hat{\Sigma}_q = \frac{S_{\mathbf{v}\mathbf{v},q} - S_{z\mathbf{v},q}^T A_q^T - A_q S_{z\mathbf{v},q} + A_q S_{zz,q} A_q^T}{w_q}. \quad (4.38)$$

Für die gemeinsame Berechnung von  $\mu_q$  und  $B_q$  werden die Mittelwerte  $\mu_q$  an das Ende des Gewichtsvektors  $B_q$  angehängt, sodass der Vektor  $A_q$  entsteht. Der Wertevektor  $\xi$  wird mit einer Anzahl an Einsen, die der Anzahl der Mittelwerte entspricht, zum Wertevektor  $z$  erweitert (Gleichungen (4.36) - (4.38)).

#### 4.2.5.2 Lernen der Parameter der Softmax-Verteilung

Für das Training der Softmax-Verteilung wird das Maximum der Plausibilitätsfunktion numerisch bestimmt. Im E-Schritt des EM-Algorithmus wird eine *a posteriori* Wahrscheinlichkeit  $\gamma_{a,t}$  für jeden Softmax-Zustand  $a$  in jedem Datenpunkt  $d_t$  bestimmt. Hierzu wird im Fall des Fahrermodells das durch die Sensordaten bekannte Wissen eingebracht und der Nachrichtenweiterleitungs-Algorithmus ausgeführt. Für den Maximierungsschritt werden die Parameter der *a priori* Wahrscheinlichkeit für jeden Zustand so optimiert, dass die Plausibilität maximal wird. Die *a priori* Wahrscheinlichkeit ist gegeben durch

$$\hat{\gamma}_{a,t} = P(A = a | \xi_t, Q = q, W) = \frac{\exp(\mathbf{w}_{a,q}^T \xi_t)}{\sum_{i \in A} \exp(\mathbf{w}_{i,q}^T \xi_t)} \quad (4.39)$$

mit  $W$  als Gewichtsmatrix, deren Parameter trainiert werden müssen. Die Plausibilität wird maximiert, indem diese Parameter so gewählt werden, dass die Differenz der *a priori* zur *a posteriori* Wahrscheinlichkeit minimal wird. Die zu maximierende Funktion, geläufig unter der Bezeichnung Kreuzentropie, ist daher definiert als

$$\Lambda(W) = \log\left(\prod_t \prod_a \hat{\gamma}_{a,t}^{\gamma_{a,t}}\right) \quad (4.40)$$

$$= \sum_t \sum_a \gamma_{a,t} \times \log(\hat{\gamma}_{a,t}). \quad (4.41)$$

Da die Kreuzentropie eine konvexe Funktion ist, ist jedes gefundene lokale Optimum auch ihr globales Optimum. Daher eignet sich ein Gradientenabstiegsverfahren gut zur Optimierung. Es sei angemerkt, dass für jeden Optimierungsschritt die *a priori* Wahrscheinlichkeit berechnet werden muss, welches zu einem hohen Berechnungsaufwand führen kann.

Mit Hilfe dieser Methoden kann das Training und die Inferenz des in Abschnitt 4.2.2 präsentierten Modells durchgeführt werden. Ergebnisse hierzu werden in Abschnitt 6.1.1 präsentiert und mit den beiden anderen evaluierten Verfahren verglichen.

## 4.3 Modellierung mit Input-Output Hidden Markov Modellen

Die zweite in dieser Arbeit untersuchte Möglichkeit zur Modellierung von Spurwechselentscheidungen menschlicher Fahrer erfolgt mit Hilfe von Input-Output Hidden Markov Modellen

(IOHMM), die in [BF96] vorgestellt wurden. Hierbei handelt es sich um eine Abwandlung von Bayes'schen Netzen, bei der zum einen neben den Eingangsdaten des aktuellen Zeitschritts einer Sequenz an Daten auch die Zustandsverteilung des vergangenen Zeitschritts mit einbezogen wird um die aktuelle Verteilung zu bestimmen. Zum anderen werden die von der Sensorik bereitgestellten Eingangsdaten nicht über Wahrscheinlichkeitsverteilungen in das Netz integriert, sondern dienen als Eingangsdaten für neuronale Netze, deren Ausgänge die Transitionswahrscheinlichkeiten zwischen Zuständen repräsentieren. In diesem Abschnitt werden zunächst die Grundlagen der Input-Output Hidden Markov Modelle dargestellt und dann ein auf dieser Methode basierendes Spurwechsel-Modell präsentiert.

### 4.3.1 Grundlagen

Das Hidden Markov Model (HMM) ist die einfachste Form eines dynamischen, also Zustände mehrerer Zeitschritte miteinbeziehenden, Bayes'schen Netzes [KF09, S. 208]. Dabei hängt die Wahrscheinlichkeitsverteilung der Zufallsvariablen  $X(t)$  zum Zeitpunkt  $t$  gemäß der Markov-Eigenschaft [KF09, S. 207] nur von der vorangegangenen Zustandsvariablen  $X(t - 1)$  zum Zeitpunkt  $t - 1$  und der Beobachtung  $Y(t)$  zum Zeitpunkt  $t$  ab:

$$P(X(t)|X(0:t-1)) = P(X(t)|X(t-1)) \quad (4.42)$$

$$P(Y(t)|X(0:t)) = P(Y(t)|X(t)). \quad (4.43)$$

Ausschnittsweise lässt sich dies als Bayes'sches Netz wie in Abbildung 4.8 darstellen.

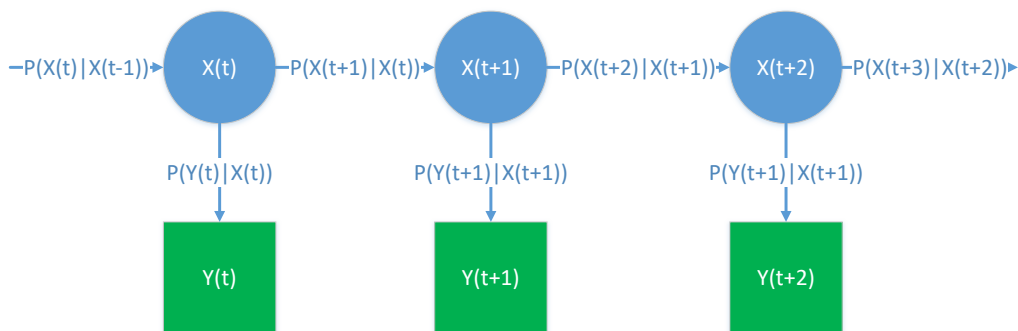


Abbildung 4.8: Hidden Markov Model

Vollständig definiert ist ein HMM nach [RJ86] durch:

**Definition 4.16 Hidden Markov Model**

Ein Hidden Markov Model über die Zufallsvariable  $X$  ist ein Tupel  $\langle X, Y, A, B, \Pi \rangle$  mit

- der Menge der versteckten Zustände  $X = \{x_1, \dots, x_n\}$ .
- der Menge der Beobachtungen  $Y = \{y_1, \dots, y_m\}$ .
- den Transitionswahrscheinlichkeiten  $A$ , von Zustand  $i$  zu Zustand  $j$  überzugehen:  $A = (a_{ij})_{i \in X, j \in X} = P(X(t) = x_j | X(t-1) = x_i)$  mit  $a_{ij} \in [0, 1]$  und  $\sum_{i \in X} a_{ij} = 1 \forall j$ .
- den Emissionswahrscheinlichkeiten  $B$ , mit dem Zustand  $j$  Beobachtung  $i$  emittiert:  $B = (b_{ij})_{i \in Y, j \in X} = P(Y(t) = y_i | X(t) = x_j)$ .
- der Anfangsverteilung der Wahrscheinlichkeiten  $X(t=0) = \Pi = [\pi_1, \dots, \pi_n]$  mit  $\pi_i \in [0, 1]$  und  $\sum_i \pi_i = 1$ .

Im Unterschied zum klassischen HMM besitzt das in dieser Arbeit genutzte IOHMM neben den vom aktuellen Zustand  $X(t)$  abhängigen Beobachtungen  $Y(t)$  zusätzlich Eingänge  $U(t)$ . Neben der vorangegangenen Zufallsvariable  $X(t-1)$  wirken sich diese auf die Wahrscheinlichkeitsverteilung der Variable  $X(t)$  und neben der aktuellen Zufallsvariable  $X(t)$  auf die Emission  $Y(t)$  aus (vgl. Abbildung 4.9). Hierdurch erweitern sich die Transitions- und Emissionswahrscheinlichkeiten für  $X(t)$  und  $Y(t)$ :

$$P(X(t)) = P(X(t)|X(t-1), U(t)) P(X(t-1)) P(U(t)) \quad (4.44)$$

$$P(Y(t)) = P(Y(t)|X(t), U(t)) P(X(t)) P(U(t)). \quad (4.45)$$

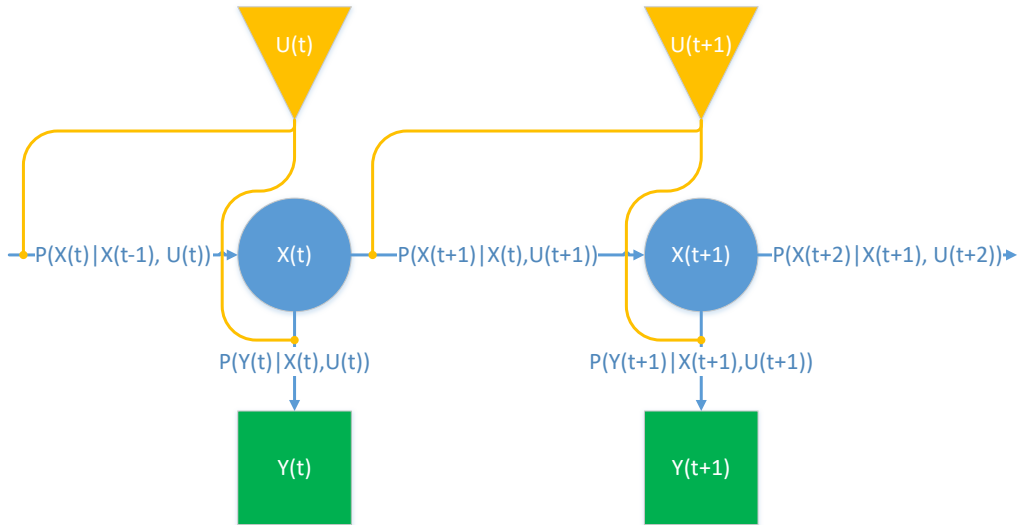
Dabei ergeben sich die Transitions- und Emissionswahrscheinlichkeiten als Ausgangswerte eines neuronalen Netzes, welches  $U(t)$  als Eingangsdaten nutzt. Eine nähere Beschreibung der Funktionsweise von neuronalen Netzen erfolgt in Abschnitt 4.4. Die Anzahl der Ausgangswerte für die Transitionswahrscheinlichkeiten wird durch die Zustände in  $X$  bestimmt und hat die Dimension  $\dim(X) \times \dim(X)$ .

Die Anzahl der Ausgangswerte für die Emissionswahrscheinlichkeiten wird durch die Zustände in  $X$  und die Emissionen in  $Y$  bestimmt und hat die Dimension  $\dim(X) \times \dim(Y)$ .

Zur Bestimmung der Zustands-Wahrscheinlichkeiten  $X(0)$  bis  $X(T)$  bei gegebenen Emissionen  $Y(0)$  bis  $Y(T)$  wird der Vorwärts-Rückwärts Algorithmus genutzt. Ähnlich wie beim Cliquesbaum-Algorithmus werden hierbei ausgehend von einer (bekannten oder trainierten) Anfangs-Zustandswahrscheinlichkeitsverteilung  $\Pi$  und Anfangsemission  $Y(t=0)$  die Zustandswahrscheinlichkeiten für die folgenden Schritte berechnet. Dies ist durch Nutzung der Markov-Eigenschaft besonders effizient möglich.

Die Wahrscheinlichkeit  $\alpha_{i,t}$ , in Zustand  $i$  zum Zeitpunkt  $t$  unter Berücksichtigung der bis dahin erfolgten Beobachtungen  $Y(t=0)$  bis  $Y(t)$  zu sein lässt sich damit nach [BF96, S. 1237, Gl. 20] durch Gleichung 4.46 berechnen:





**Abbildung 4.9:** Input-Output Hidden Markov Model. Die Kreise repräsentieren die Zufallsvariable  $X$ , die Dreiecke die Eingänge  $U$  und die Quadrate die Ausgänge  $Y$ . Die jeweils nächste Zustands-Wahrscheinlichkeitsverteilung wird gemäß (4.44) berechnet. Analog erfolgt die Berechnung der Wahrscheinlichkeitsverteilung des Ausgangs gemäß (4.45)

$$\begin{aligned} \alpha_{i,t} &= P(Y(0:t), X(t) = i | U(0:t)) \\ &= P(Y(t) | X(t) = i, U(t)) \sum_{j=1 \dots \dim(X)} P(X(t) = i | X(t-1) = j, U(t)) \alpha_{j,t-1}. \end{aligned} \quad (4.46)$$

Eine vergleichbare Rekursion ist auch in die entgegengesetzte Richtung möglich. Hierbei gibt  $\beta_{i,t}$  die Wahrscheinlichkeit an, zum Zeitpunkt  $t$  im Zustand  $i$  zu sein unter Berücksichtigung der von da an bis zum Ende der Sequenz  $t = T$  erfolgten Beobachtungen, und lässt sich nach [BF96, S. 1237, Gl. 22] durch Gleichung 4.47 berechnen:

$$\begin{aligned} \beta_{i,t} &= P(Y(t+1:T) | X(t) = i, U(t:T)) \\ &= \sum_{j=1 \dots \dim(X)} P(Y(t+1) | X(t+1) = j, U(t+1)) P(X(t+1) = j | X(t) = i, U(t+1)) \beta_{j,t+1}. \end{aligned} \quad (4.47)$$

Sind die Emissionen der ganzen Sequenz bekannt, so kann die Zustands-Wahrscheinlichkeitsverteilung in jedem Schritt durch Multiplikation der entsprechenden  $\alpha$  und  $\beta$  Werte berechnet werden:

$$P(X(t) = i) = \alpha_{i,t} \beta_{i,t}. \quad (4.48)$$

### 4.3.2 Training von IOHMM

Sind die Transitions- und Emissionswahrscheinlichkeiten bzw. die Parameter der neuronalen Netze, mit denen sie berechnet werden, nicht bekannt, können sie durch maschinelles Lernen ermittelt werden. Dazu wird bei gegebenen Eingängen  $U(t)$  und vorheriger Zustands-Wahrscheinlichkeitsverteilung  $X(t-1)$  ähnlich wie bei den Bayes'schen Netzen der Expectation-Maximization Algorithmus angewandt. Ziel des Algorithmus ist es, die Plausibilität  $\Lambda$  zu maximieren. Diese lässt sich aus der Summe der Wahrscheinlichkeiten von  $\alpha_{i,T}$  berechnen:

$$\Lambda = P(Y(0:T)|U(t=0:T)) = \sum_i \alpha_{i,T}. \quad (4.49)$$

Ausgehend von zufällig parametrisierten neuronalen Netzen werden hierfür die *a posteriori* Transitions- und Emissionswahrscheinlichkeiten  $h$  und  $g$  ermittelt. Diese lassen sich nach [BF96, S. 1238, Gl. 23 und 24] berechnen:

$$\begin{aligned} h_{ij,t} &= P(X(t) = i, X(t-1) = j | Y(t=0:T), U(0:T)) \\ &= \frac{P(Y(t)|X(t) = i, U(t))\alpha_{j,t-1}\beta_{i,t}P(X(t) = i, X(t-1) = j|U(t))}{\Lambda} \end{aligned} \quad (4.50)$$

$$\begin{aligned} g_{i,t} &= P(X(t) = i | Y(0:T), U(0:T)) \\ &= \frac{\alpha_{i,t}\beta_{i,t}}{\Lambda}. \end{aligned} \quad (4.51)$$

Im zweiten Schritt, beschrieben in [BF96, ebd.], werden für jeden Zustand bzw. jede Emission die Parameter  $\theta_j$  bzw.  $\psi_j$  ihres neuronalen Netzes  $N_j$  bzw.  $O_j$  so optimiert, dass das Produkt zwischen den *a posteriori* geschätzten Wahrscheinlichkeiten und dem Logarithmus der *a priori* Wahrscheinlichkeiten der neuronalen Netze minimal wird, also wiederum die Kreuzentropie maximiert wird:

$$\Lambda(\theta_j) = \sum_{t=1}^T \sum_{i=1}^{\dim(x)} \hat{h}_{ij,t} \log P(X(t) = i | X(t-1) = j, U(t)) \quad (4.52)$$

$$\Lambda(\psi_j) = \sum_{t=1}^T \hat{g}_{j,t} \log P(Y(t) | X(t) = j, U(t)). \quad (4.53)$$

Diese beiden Schritte werden wiederholt bis die Plausibilität konvergiert, d.h. die Differenz zwischen den Plausibilitäten zweier Iterationen kleiner als ein Wert  $\epsilon$  ist.

### 4.3.3 Modellierung von Spurwechselentscheidungen

Spurwechselentscheidungen lassen sich mit Hilfe von IOHMM sehr direkt modellieren wie in Abbildung 4.10 dargestellt. Als Beobachtungen bzw. Emissionen  $Y$  kann das tatsächliche laterale Verhalten eines Fahrers genutzt werden. Als Eingänge  $U$  werden die von der Sensorik erfassten Daten wie Geschwindigkeiten, Abstände, die Verfügbarkeit von Nachbarspuren etc. verwendet. Die versteckten Zustände  $X$  entsprechen den Wahrscheinlichkeiten für „Spurwechselentscheidung Links“, „Spur halten“ und „Spurwechselentscheidung Rechts“ des zu modellierenden Fahrers und können mit Hilfe der Eingänge und Emissionen über eine Sequenz bestimmt werden.

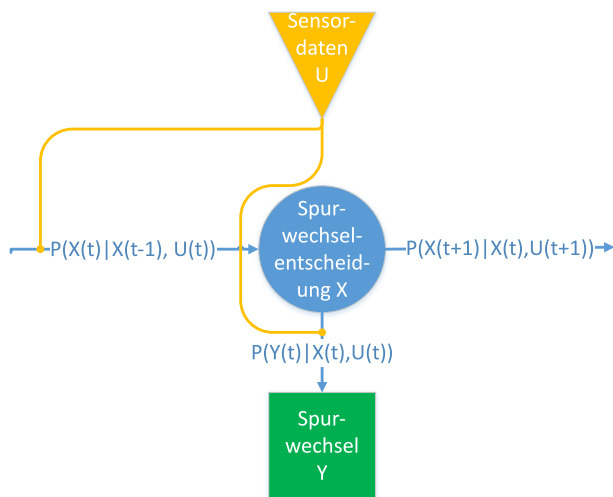


Abbildung 4.10: Modellierung des Spurwechsels als IOHMM

## 4.4 Modellierung mit tiefen feedforward neuronalen Netzen

Tiefe neuronale Netze (DNN) haben sich als mächtige Werkzeuge in verschiedenen Forschungsbereichen wie Bildklassifikation oder Go-Spielen erwiesen ([KSH12] [SHM<sup>+</sup>16]). In der vorliegenden Arbeit werden tiefe feedforward neuronale Netze zur Modellierung von Spurwechseln genutzt. Dabei werden die Parameter  $\theta$  einer Funktion  $\mathcal{S}$  zur Modellierung der Spurwechselentscheidung eines Fahrers so gewählt, dass ihr Ausgang  $z = \mathcal{S}(\mathbf{u}|\theta)$  ist, also ein Eingangsvektor  $\mathbf{u}$  auf eine Fahrentscheidung  $z$  abgebildet wird. Der Eingangsvektor  $\mathbf{u}$  enthält dabei die in Tabelle 4.1 aufgelisteten durch Sensorik erfassten Daten wie Abstände und Geschwindigkeiten.

### 4.4.1 Grundlagen

Unter „tiefen“ neuronalen Netzen wird eine Art von neuronalen Netzen verstanden, bei der zwischen Eingangs- und Ausgangsschicht mehrere versteckte Schichten liegen. Im Gegensatz zu einfachen neuronalen Netzen ist hierdurch ein Lernerfolg des Netzes wahrscheinlicher, da komplexere Funktionen abgebildet werden können [GBC16, S. 7].

„Feedforward“ bedeutet, dass die Eingangsinformationen sukzessive von Zwischenschicht zu Zwischenschicht weitergereicht werden, bis zur Endverarbeitung in der Ausgangsschicht, also keine Schleifen (Rekursionen) auftreten. Hierbei handelt es sich um die einfachste und verbreitetste Form von neuronalen Netzen.

Die Bezeichnung „neuronales Netz“ entstammt der Biologie. In dieser werden mit Hilfe von Neuronen Informationen verarbeitet und weitergeleitet [MP43]. Das Grundprinzip von tiefen feedforward neuronalen Netzen beruht darauf, dass in einem Neuron aus den Ausgaben anderer Neuronen der vorhergehenden Schicht eine gewichtete Summe gebildet wird und diese als Eingang für eine nicht-lineare Funktion, der sogenannten Aktivierungsfunktion, genutzt wird. Der Ausgang des Neurons wird wiederum als Eingang von Neuronen der nächsten Schicht genutzt.

In der vorliegenden Arbeit werden zwei verschiedene Aktivierungsfunktionen verwendet. In den Eingangs- und Zwischenschichten wird der Tangens hyperbolicus eingesetzt, in der Ausgangsschicht die bereits aus Abschnitt 4.2.3.1 bekannte Softmax-Funktion:

$$\tanh(\mathbf{u}) = 1 - \frac{2}{1 + e^{2\mathbf{u}}} \quad (4.54)$$

$$\text{softmax}(u_i|\mathbf{w}) = \frac{e^{w_i \mathbf{u}_i}}{\sum_k e^{w_k \mathbf{u}_k}} \quad (4.55)$$

mit  $\mathbf{u}$  als Eingangsvektor des Neurons. Im Fall des Tangens hyperbolicus wird der Eingangsvektor gewichtet und summiert, bevor er als Argument genutzt wird. Im Fall des Softmax erfolgen Gewichtung und Nutzung der einzelnen Eingangsdaten zur Generierung eines probabilistisch interpretierbaren Ausgangs in der Funktion selbst.

#### 4.4.1.1 Training tiefer neuronaler Netze

Das überwachte Lernen von neuronalen Netzen, also von Netzen bei denen für die zum Training genutzten Daten sowohl die Eingangs- als auch die Ausgangsdaten bekannt sind, erfolgt über den Rückpropagierungs-Algorithmus. Hierbei wird die am Ausgang des Netzes bestehende Differenz („Fehler des Netzes“) zwischen Soll-Ausgangswert, wie er aus den Trainingsdaten bekannt ist, und Ist-Ausgangswert, wie er von dem zu trainierenden neuronalen Netz erzeugt wurde, sukzessive an die vorherigen Schichten des Netzes zurückpropagiert. Dabei wird der am Ausgang berechnete Fehler  $F$  mit Hilfe der Kettenregel (4.56) nach den einzelnen Gewichten  $w_{ij}$  im Netz abgeleitet (4.57) und dann die Gewichte in entgegengesetzte Richtung ihres Einflusses auf den Fehler gemäß der Lernrate  $\eta$  verändert (4.58).

$$\frac{\partial F}{\partial w_{ij,k}} = \frac{\partial F}{\partial a_j} \frac{\partial a_j}{\partial e_j} \frac{\partial e_j}{\partial w_{ij,k}} \quad (4.56)$$

$$\Delta w_{ij} = -\eta \frac{\partial F}{\partial w_{ij}} \quad (4.57)$$

$$w_{ij}^{\text{neu}} = w_{ij,k}^{\text{alt}} + \Delta w_{ij,k} \quad (4.58)$$

mit  $a_j$  als Ausgangsneuron  $j$ ,  $e_j$  als den Eingangsdaten des Ausgangsneurons  $a_j$  und  $w_{ij,k}$  als dem Gewicht zwischen Neuron  $n_i$  in Schicht  $k - 1$  und Neuron  $n_j$  in Schicht  $k$ . So wird der Fehler von Schicht zu Schicht weiter Richtung Eingangsschicht propagiert, bis alle Gewichte angepasst wurden. Danach wird der Vorgang wiederholt, indem erneut die Ausgänge des Netzes berechnet, die Differenz zu den Trainingswerten ermittelt und die Gewichte neu angepasst werden. Das Training entspricht somit einem Gradientenabstiegsverfahren, für welches es inzwischen optimierte Algorithmen gibt die auf mehreren Prozessoren oder speziell zu diesem Zweck entwickelter Hardware lauffähig sind und damit auch für große Netze mit vielen Schichten in endlicher Zeit hinreichend gute Lösungen finden (vgl. [AAB<sup>+</sup>16]).

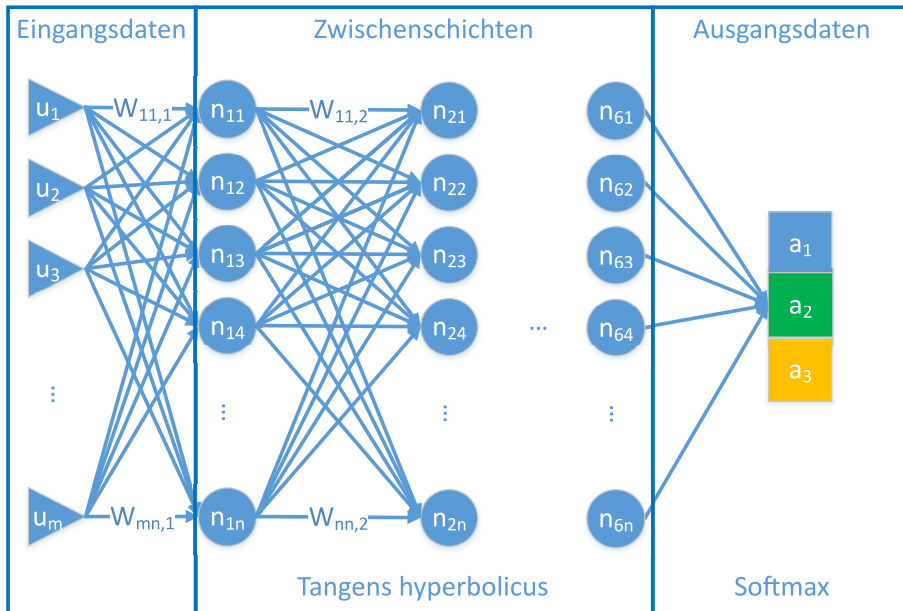
Der Fehlerterm kann zusätzlich zu dem durch die Abweichung von Netzausgabe und Sollwerten entstehenden Fehler um einen Regularisierungsterm, mit dem Gewichte mit hohen Werten bestraft werden, erweitert werden. Dies führt regelmäßig zu einer höheren Generalisierbarkeit des Netzes und wirkt einer Überanpassung des Netzes an die Trainingsdaten entgegen [MKS17] [HDBJ14].

#### 4.4.2 Modellierung von Spurwechselentscheidungen mit tiefen neuronalen Netzen

Als weiteren Modellierungsansatz für Spurwechselentscheidungen von Fahrern zur Nutzung in einer Simulation im Rahmen des passiven HAF werden in dieser Arbeit tiefe feedforward neuronale Netze verwendet. Diese wurden vorher in einer Masterarbeit auf ihre grundsätzliche Eignung zur Modellierung von Spurwechsel hin überprüft [Gut16].

Sie bestehen aus einer Eingangsschicht mit den in Abschnitt 4.1 genannten Eingangsdaten, sechs versteckten Schichten mit jeweils 20 Neuronen und einer Ausgangsschicht mit drei Neuronen, dargestellt in Abbildung 4.11. Als Regularisierungsfaktor wird für zu trainierende Netze  $\nu_{\text{reg}} = 0,5$  verwendet, als Fehlerfunktion die Kreuzentropie von Netzausgabe und Trainings-Sollwerten.

Die Neuronen der Ausgangsschicht repräsentieren dabei die drei möglichen Spurentscheidungen des modellierten Fahrers. Die Spurwahl mit der höchsten Wahrscheinlichkeit wird in der Simulation zur Bestimmung der Querführung des Fahrers genutzt.



**Abbildung 4.11:** Zur Spurwechselmodellierung genutztes tiefes neuronales Netz mit sechs Zwischenschichten von jeweils 20 Neuronen

## 4.5 Vergleich konzeptbedingter Vor- und Nachteile der Ansätze

In diesem Kapitel wurden drei verschiedene Ansätze zur Modellierung von Spurwechselentscheidungen von Fahrern vorgestellt: Hybride Bayes'sche Netze, Input-Output Hidden Markov Modelle und tiefe feedforward neuronale Netze. In dieser Reihenfolge nimmt auch die Mächtigkeit der Ansätze, also die Menge an durch die Modelle repräsentierbaren Funktionen, zu, während ihre Interpretierbarkeit abnimmt.

Bayes'sche Netze erlauben eine Modellierung mit Hilfe der beschriebenen Wahrscheinlichkeitsverteilungen, die in dieser Arbeit noch um die Variante der hybriden Bayes'schen Netze mit der Softmax-Wahrscheinlichkeitsverteilung erweitert wurde. Die Modellierung erfolgt somit im wesentlichen mit Hilfe linearer Modelle und Gaußverteilungen. Jeder Knoten des Netzes steht für eine bestimmte Eigenschaft eines Fahrermodells: die Eingangsknoten für die Sensordaten, die zwischen Eingangs- und Ausgangsknoten liegenden Knoten für Einflussfaktoren wie z.B. Zufriedenheit des Fahrers mit der Geschwindigkeit oder mit den umgebenden Fahrzeugen auf einer Spur. Sie sind somit direkt interpretierbar. Voraussetzung hierfür ist, dass während des Trainings auch für die im Normalfall versteckten Knoten Wissen vorhanden ist, um sie mit den passenden Daten in die für sie vorgesehene Richtung zu führen. Dies ist in dieser Arbeit der Fall, da zunächst Simulationsdaten als Trainingsdaten genutzt werden, bevor ein Training mit Realdaten erfolgt. Bei den Simulationsdaten liegt auch für eigentlich versteckte Zufallsvariablen Wissen vor.

Beim vorgeschlagenen Bayes'schen Netz handelt es sich um ein statisches Bayes'sches Netz, d.h. pro Inferenzschritt werden nur Informationen des jeweiligen Zeitschritts verarbeitet, vorangegangene oder nachfolgende Informationen werden jedoch nicht berücksichtigt. Ein dynamisches Bayes'sches Netz mit Nutzung vergangener Daten ähnlich dem Hidden Markov Model ist möglich, würde das Netz aber um die entsprechende Anzahl an Zeitschritten größer werden lassen und somit den Aufwand der Inferenz stark erhöhen. Die Inferenz und noch mehr das Training von hybriden Bayes'schen Netzen ist jedoch aufgrund der erforderlichen numerischen Integration bereits mit der vorgestellten Topologie mit hohen Anforderungen an die Rechenleistung verbunden, sodass in der vorliegenden Arbeit von einer dynamischen Variante abgesehen wird.

Input-Output Hidden Markov Modelle sind ein Hybrid aus einem dynamischen Bayes'schen Netz, dem Hidden Markov Modell, und neuronalen Netzen. Aufgrund der Nutzung von neuronalen Netzen zur Berechnung der Transitions- und Emissionswahrscheinlichkeiten aus Sensordaten und vorangegangenen Zustandsdaten kann hier eine Vielzahl an Funktionstypen trainiert werden, während die möglichen Zustände und Emissionen auf jeweils drei diskrete Möglichkeiten beschränkt sind. Dadurch ist zum einen eine hohe Flexibilität bei der Verarbeitung der Eingangsdaten gegeben, zum anderen eine gute Interpretierbarkeit der Zustände und Emissionen. Für Menschen unverständlich bleibt jedoch, nach welchen Regeln innerhalb der verwendeten neuronalen Netze die Transitions- und Emissionswahrscheinlichkeiten bestimmt werden.

Vorteil der IOHMM ist, dass vergangene und, soweit bekannt, zukünftige Informationen mit in die Wahrscheinlichkeitsberechnung aktueller Zeitschritte einbezogen werden können. Hierdurch ist es möglich, eine maximale Wahrscheinlichkeit über eine ganze Sequenz und nicht nur für einen Zeitschritt zu berechnen. Der Rechenaufwand ist aufgrund der Nutzung der Markov-Eigenschaft und verbreiteter Optimierer zum Training neuronaler Netze geringer als bei Bayes'schen Netzen. Da der Estimation-Schritt aufwändiger als bei Tiefen neuronalen Netzen ist, sind IOHMM diesen hinsichtlich ihrer Rechenzeit unterlegen.

Tiefe neuronale Netze erlauben die Modellierung einer Vielzahl an Funktionen, sind aufgrund der zur Verfügung stehenden Hardware und Algorithmen gut zu trainieren und führten auf verschiedenen Gebieten zu gegenüber anderen Verfahren deutlich besseren Trainingsergebnissen. Im Gegensatz zu Bayes'schen Netzen und IOHMM sind die trainierten tiefen neuronalen Netze im eigentlichen Sinne Blackbox-Modelle, d.h. eine Interpretierbarkeit von versteckten Schichten ist in der Regel nicht möglich.

Die Nicht-Interpretierbarkeit der trainierten Modelle ist hinsichtlich der Absicherungsfähigkeit der Modelle von Nachteil, da sie nicht von Experten validiert oder zumindest auf Plausibilität überprüft werden können. Lediglich eine Verifikation und Validierung durch Testfälle ist möglich. Diese können aber aufgrund der Komplexität der Modelle nicht jede beliebige Fehlermöglichkeit abdecken, sondern nur ihre Auftretenswahrscheinlichkeit senken.

Für das Konzept passives HAF ist dies allerdings weniger kritisch zu bewerten, da ein in bestimmten Situationen fehlerhaftes Modell lediglich zu einem falschen Simulationsschritt führen würde. Dies würde jedoch nicht zu einer kritischen Situation in einem realen Fahrmanöver führen, sondern lediglich zu einer Abweichung im weiteren simulierten Szenarioverlauf.

Kommt es selten zu einer fehlerhaften Modellierung des Fahrverhaltens umgebender Verkehrsteilnehmer, ist es für eine statistische Absicherungsaussage nicht relevant.

Grund hierfür ist, dass ein Fehler in der HAF-Funktion genau gleichzeitig mit einem Fehler

in der Fahrermodellierung auftreten müsste, durch den eine eigentlich kritische Situation unkritisch erscheint. Nur im Fall dieses kombinierten Auftretens von mehreren Fehlerfällen könnte die Absicherungsaussage beeinflusst werden.

Die HAF-Funktion durchläuft vor Einsatz im passiven HAF einen gewöhnlichen Absicherungsprozess. Trainierte Fahrermodelle kommen erst zum Einsatz, wenn sie eine hohe Güte aufweisen. Träten Modellfehler häufiger auf, fielen diese bei der Zuordnung der Fahrermodelle zu umgebenden Verkehrsteilnehmern auf. Im Fall häufiger Fehler wären die trainierten neuronalen Netze nicht in der Lage, das Verhalten realer Fahrer abzubilden, sodass für diese kein geeignetes Modell nach dem in Abschnitt 3.5 beschriebenen Prozess gefunden werden könnte. Eine Verbesserung der Modelle durch Training mit zusätzlichen Trainingsdaten wäre die Folge, wodurch erkannte Fehler in der Fahrermodellierung behoben würden.

Daher können Einzelfehler in der HAF-Funktion und bei der Fahrermodellierung als selten angenommen werden. Das gleichzeitige auftreten ist damit sehr unwahrscheinlich, sodass es für die statistische Absicherung vernachlässigt werden kann.

Nach Vorstellung der Kritikalitätserkennung in Kapitel 5 werden die drei Ansätze in Kapitel 6 zunächst mit Simulationsdaten darauf überprüft, wie gut sie simulierte Verkehrsszenarien nachbilden können. Je nach erzielter Güte der Modelle werden diese dann weiter mit realen Daten trainiert, um zu einer Aussage der Nutzbarkeit der Modelle für das passive HAF zu kommen.



## 5 Kritikalitätserkennung

Der letzte Schritt im Konzept passives HAF ist es, auf Grundlage des simulierten Szenarios zu bewerten, ob die fiktive Fahrentscheidung der HAF-Funktion zu einer kritischen Situation geführt hätte. Die Kritikalität einer Situation ist dabei der Definition von [RHW<sup>+</sup>12] folgend die Wahrscheinlichkeit, dass ein schädliches Ereignis eintritt. Ein schädliches Ereignis ist im Kontext des HAF ein Unfall. Die Wahrscheinlichkeit für einen Unfall nimmt zu mit der Abnahme an Möglichkeiten für die beteiligten Fahrer, diesen abzuwenden.

Bei jeder Fahrentscheidung des HAF, die von der des menschlichen Fahrers abweicht, wird bei Erfüllung der Auslösebedingungen eine Simulation des weiteren Szenarioverlaufs durchgeführt. In der Regel ist anzunehmen, dass eine Abweichung unkritisch ist und vom HAF lediglich eine alternative, unkritische Fahrentscheidung getroffen wurde. Aufgrund der Anzahl an insgesamt simulierten Szenarien muss daher eine automatisierte Vorauswahl an potentiell kritischen Szenarien getroffen werden. Nur dann ist es möglich, diese detaillierter zu überprüfen und gegebenenfalls bei einem Fehler der HAF-Funktion eine Funktionsänderung vorzunehmen. Die automatisierte Kritikalitätserkennung muss daher alle kritischen Situationen erkennen, gleichzeitig aber möglichst wenige unkritische Situationen als kritisch detektieren.

Bei den verbreiteten Kritikalitätskriterien Zeit bis zur Kollision (TTC) oder Zeit bis zum letztmöglichen Bremszeitpunkt (TLSB) wird jeweils die Zeit berechnet, bis es ohne Fahrereingriff zu einer Kollision kommt oder diese durch eine Fahreraktion nicht mehr abwendbar ist. Aus ihnen kann abgeleitet werden, welche Reaktionszeit ein Fahrer oder System noch hat. Zur Berechnung der Zeiten wird von konstanten Beschleunigungen oder Geschwindigkeiten der involvierten Fahrzeuge ausgegangen. Situationen werden immer dann als kritisch bewertet, wenn die Zeit einen definierten Schwellwert unterschreitet.

Komplexere Kritikalitätsmetriken berücksichtigen zusätzlich die Art und die Schwere einer möglichen Kollision [TDB11]. Dies ermöglicht es automatisierten Fahrsystemen, bei unvermeidbaren Unfällen Fahrmanöver zu wählen, die zumindest die Schadensschwere des Unfalls abmildern.

Weiterhin können Unsicherheiten der Sensorik und Prädiktion bei der Berechnung der Kollisionszeitpunkte mit einbezogen werden [SSBZ15][SW14][RHW<sup>+</sup>12]. Ebenfalls kann der Aufwand zur Abwendung der kritischen Situation in die Bewertung mit einbezogen werden. Ist es beispielsweise trotz negativer TLSB durch einen Spurwechsel möglich, eine Kollision auch noch deutlich nach dem letztmöglichen Bremszeitpunkt zu verhindern, könnte die entsprechende Situation als weniger kritisch betrachtet werden als Situationen, in der diese Möglichkeit nicht existiert.

Ziel dieser komplexeren Kritikalitätsmetriken ist es in der Regel, auf Grundlage der Kritikalitätsbewertung im Rahmen einer Sicherheits-Assistenzfunktion eine Fahrerwarnung oder ein Fahrmanöver zu initiieren, die zu einer Minderung der Unfallschwere oder Vermeidung eines Unfalls führt. Dafür ist es erforderlich, eine Warnung nur auszugeben bzw. nur dann

in die Fahrzeugführung des Fahrers einzugreifen, wenn eine kritische Situation mit hoher Wahrscheinlichkeit festgestellt werden konnte. Bei zu häufigen Fehlwarnungen eines Systems würde dies durch den Fahrer ignoriert oder deaktiviert werden. Unberechtigte Fahreingriffe aufgrund fehlerhaft als kritisch erkannter Situationen durch ein Sicherheitssystem würden, neben Kundenunzufriedenheit mit dem System, selbst ein Sicherheitsrisiko darstellen. Jeder Eingriff wie beispielsweise eine Notbremsung kann bei zu langsamer Reaktion des rückwärtigen Verkehrs zu einer kritischen Situation führen. Die Sicherheits-Assistenzfunktionen sind daher konservativ ausgelegt, greifen also eher zu spät als zu früh ein.

Das Ziel der Kritikalitätserkennung im Rahmen des passiven HAF ist wie oben beschrieben die Erkennung aller kritischen Situationen. Die fehlerhafte Erkennung von nicht-kritischen Situationen als kritisch hat im Gegensatz zum Einsatz in Sicherheits-Assistenzfunktionen nur dann negative Folgen für die Nutzbarkeit des passiven HAF, wenn ihre Anzahl so hoch ist, dass eine nachgelagerte Analyse durch Experten nicht mehr möglich ist.

Zur Detektion kritischer Situationen wurde daher für diese Arbeit ein Kritikalitätsalgorithmus entwickelt, mit dessen Hilfe die simulierten Szenarien nach in der Literatur beschriebenen objektiven Kenngrößen für Kritikalität bewertet werden. Übersteigt die Kritikalität eines Szenarios für eine der definierten Kenngrößen den festgelegten Grenzwert, so wird dieses Szenario dauerhaft gespeichert. Das Szenario wird für die weitere Funktionsentwicklung genutzt, um in künftigen Versionen der HAF-Funktion Änderungen zu implementieren, die zu einer Reduktion der Kritikalität führen. Im Folgenden werden potentielle Metriken zur Kritikalitätsbewertung vorgestellt, die in dieser Arbeit auf ihre Anwendbarkeit hin untersucht werden.

## 5.1 Kritikalitätskriterium 1: Modifizierte Zeit bis zur Kollision

Die Modifizierte Zeit bis zur Kollision (Modified Time to Collision,  $T_{\text{MTTC}}$ ) wurde in [Yan12] vorgestellt. Sie beschreibt die Zeitdauer, bis das folgende Fahrzeug F mit dem vorausfahrenden Fahrzeug V kollidiert, wenn keins der beiden Fahrzeuge Änderungen in seiner longitudinalen Dynamik vornimmt oder auf eine andere Spur wechselt. Im Unterschied zum klassischen Kritikalitätsmaß Time To Collision berücksichtigt die  $T_{\text{MTTC}}$  neben dem Abstand  $d_{\text{diff}} = x_V - x_F$  und der Differenzgeschwindigkeit  $v_{\text{diff}} = v_V - v_F$  auch die Differenzbeschleunigung  $a_{\text{diff}} = a_V - a_F$  zwischen den Fahrzeugen. Für die Berechnung der  $T_{\text{MTTC}}$  wird von einer konstantbleibenden Beschleunigung ausgegangen. In [Yan12] wurde gezeigt, dass die MTTC besser als die Time To Collision (TTC) in der Lage ist, kritische Situationen zu erkennen. Die Berechnung erfolgt anhand der folgenden Vorschriften:

1. Ist die Beschleunigungsdifferenz zwischen den Fahrzeugen  $a_{\text{diff}} \neq 0$  wird zunächst überprüft, ob es zu einer Kollision kommen kann, indem die quadratische Bewegungsgleichung (5.1) gelöst wird.

$$\frac{1}{2}a_{\text{diff}}T_{\text{MTTC}}^2 + v_{\text{diff}}T_{\text{MTTC}} + d_{\text{diff}} = 0 \quad (5.1)$$

Eine Kollision ist möglich, wenn die quadratische Gleichung einen positiven Extremwert mit einer negativen Steigung oder einen negativen Extremwert mit einer positiven Steigung hat. In diesem Fall wird die  $T_{\text{MTTC}}$  nach (5.2) berechnet:

$$T_{\text{MTTC}} = \frac{-v_{\text{diff}} \pm \sqrt{v_{\text{diff}}^2 + 2a_{\text{diff}}d_{\text{diff}}}}{a_{\text{diff}}}. \quad (5.2)$$

Von den beiden möglichen Lösungen von (5.2) werden nur die positiven genutzt, da negative Werte für  $T_{\text{MTTC}}$  eine Kollision in der Vergangenheit beschreiben. Sind beide Werte positiv wird der kleinere Wert für die Kritikalitätsbewertung genutzt.

- Da im Gegensatz zur TTC bei der MTTC auch die Beschleunigung mit berücksichtigt wird, sind theoretisch auch Fahrzeuge mit negativer Geschwindigkeit zum Zeitpunkt der Kollision möglich. Daher werden zunächst die absoluten Geschwindigkeiten des vorausfahrenden und des nachfolgenden Fahrzeugs überprüft. Sind die rechnerischen Werte der Geschwindigkeit des nachfolgenden Fahrzeugs zum Zeitpunkt der Kollision negativ und der des vorausfahrenden Fahrzeugs positiv, kann es nicht zu einer realen Kollision gekommen sein. Ist die errechnete Geschwindigkeit des vorausfahrenden Fahrzeugs zum Zeitpunkt der Kollision negativ, wird angenommen, dass es zum Zeitpunkt der Kollision stillstand. In diesem Fall berechnet sich die  $T_{\text{MTTC}}$  zu

$$T_{\text{MTTC}} = \frac{-V_F \pm \sqrt{V_F^2 + 2a_F(d_{\text{diff}} - \frac{v_V^2}{2a_V})}}{-a_F}. \quad (5.3)$$

Auch hier müssen die Voraussetzungen für eine Kollision überprüft werden, um komplexe Lösungen zu vermeiden.

- Ist die Beschleunigung von vorausfahrendem und folgendem Fahrzeug identisch, entspricht die MTTC der bekannten TTC bis zu dem Zeitpunkt, zu dem eins der Fahrzeuge die Geschwindigkeit  $v = 0$  erreicht. Kommt das folgende Fahrzeug F zum Stillstand, kann keine Kollision erfolgen. Bleibt das Fahrzeug V stehen, berechnet sich die  $T_{\text{MTTC}}$  nach (5.2).
- Für den Sonderfall, dass das vorausfahrende Fahrzeug anhält und das folgende Fahrzeug nicht beschleunigt ( $a_F = 0$ ), wird (5.4) verwendet, um eine Division durch Null zu vermeiden:

$$T_{\text{MTTC}} = \frac{d_{\text{diff}} - \frac{v_V^2}{2a_V}}{v_F}. \quad (5.4)$$

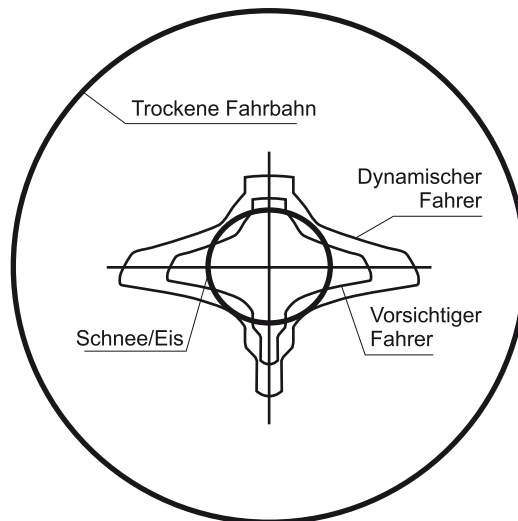
## 5.2 Kritikalitätskriterium 2: Komfort- und Fahrdynamikgrenzen

Als weiteres Kritikalitätsmerkmal wird betrachtet, ob die Beschleunigung des Fahrzeugs in Längs- und Querrichtung im Zuge der HAF-Fahrmanöver Maximalwerte überschreitet. Die

Maximalwerte ergeben sich aus der physikalischen Fahrdynamikgrenze („Kamm’scher Kreis“) und den in [VIA<sup>+</sup>14] und [WHW12] definierten Komfortgrenzen. Hierzu werden die Verhältnisse  $V_{\text{phys}}$  und  $V_{\text{komf}}$  der Beschleunigung des Fahrzeugs  $a_{\text{ist}}$  zu der physikalischen Grenze  $a_{\text{max,phys}}$  und Komfortgrenze  $a_{\text{max,komf}}$  berechnet (vgl. Abbildung 5.1). Als einzige der Bewertungskriterien sind sie unabhängig von umgebenden Verkehrsteilnehmern und bewerten direkt die Qualität der Fahrmanöver.

$$V_{\text{phys}} = \frac{a_{\text{ist}}}{a_{\text{max,phys}}} \quad (5.5)$$

$$V_{\text{komf}} = \frac{a_{\text{ist}}}{a_{\text{max,komf}}} \quad (5.6)$$



**Abbildung 5.1:** Fahrerhaltens-Beschleunigungs-Diagramm aus [WHW12, S. 21] zur Bestimmung des Verhältnisses von tatsächlicher Beschleunigung zur durch Fahrdynamik- und Komfortgrenzen gegebene Beschleunigung

### 5.3 Kritikalitätskriterium 3: Letztmöglicher Bremszeitpunkt

Die Zeit bis zum letztmöglichen Bremszeitpunkt  $T_{\text{LB}}$  nach [ZAG06] beschreibt die Zeit, die dem Fahrer verbleibt, um eine Kollision zu verhindern, indem er eine Notbremsung mit maximaler Verzögerung einleitet. Ähnlich wie bei der *Modifizierten Zeit bis zur Kollision* wird für die Berechnung von konstanten Beschleunigungen ausgegangen. Je nach Fall erfolgt die Berechnung der  $T_{\text{LB}}$  wie folgt:

1. Wenn  $a_{\text{diff}} \neq 0$  und  $T_{\text{MTTC}}$  berechnet werden konnte, also eine Kollision möglich ist:

$$\mathcal{A} = a_{\text{diff}}(a_{F,\text{max}} - a_V) + a_{\text{diff}}^2 \quad (5.7)$$

$$\mathcal{B} = 2v_{\text{diff}}(a_{F,\text{max}} - a_V) + 2v_{\text{diff}}a_{\text{diff}} \quad (5.8)$$

$$\mathcal{C} = 2d_{\text{diff}}(a_{F,\text{max}} - a_V) + v_{\text{diff}}^2 \quad (5.9)$$

$$T_{\text{LB}} = -\mathcal{B} \pm \frac{\sqrt{\mathcal{B}^2 - 4\mathcal{A}\mathcal{C}}}{2\mathcal{A}}. \quad (5.11)$$

Dabei ist  $a_{F,\text{max}}$  die maximal mögliche Verzögerung des folgenden Fahrzeugs.

2. Falls das vorausfahrende Fahrzeug vor dem folgenden zum stehen kommt, also  $t_{V,\text{stop}} < t_{F,\text{stop}}$  ist, muss eine alternative Berechnungsvorschrift für  $\mathcal{A}$ ,  $\mathcal{B}$  und  $\mathcal{C}$  verwendet werden:

$$\mathcal{A} = \frac{a_F}{2} - \frac{a_F^2}{2a_{F,\text{max}}} \quad (5.12)$$

$$\mathcal{B} = v_F - \frac{v_F a_F}{a_{F,\text{max}}} \quad (5.13)$$

$$\mathcal{C} = \frac{v_V^2}{2a_V} - \frac{v_F^2}{2a_{F,\text{max}}} - d_{\text{diff}} \quad (5.14)$$

und

$$t_{V,\text{stop}} = -\frac{v_V}{a_V} \text{ für } a_V < 0 \quad (5.15)$$

$$t_{F,\text{stop}} = T_{\text{LB}} - \frac{(v_F + a_F T_{\text{LB}})^2}{2a_{F,\text{max}}}. \quad (5.16)$$

Der Vergleich des Verhaltens von  $T_{\text{LB}}$  und  $T_{\text{MTTC}}$  erlaubt zwei Erkenntnisse. Wenn beide Zeiten nah beieinander liegen, ist eine Bremsung knapp vor der Kollision notwendig und daher in der Regel auch für den Fahrer leicht erkennbar. Ist hingegen  $T_{\text{MTTC}}$  wesentlich größer als  $T_{\text{LB}}$ , ist eine Bremsung deutlich vor Eintritt der Kollision notwendig, sodass die Kritikalität der Situation durch den Fahrer schwerer bewertbar ist.

## 5.4 Gesamtkritikalität

Zur Bestimmung der Gesamtkritikalität einer Situation werden den berechneten Werten von  $T_{\text{MTTC}}$ ,  $V_{\text{phys}}$ ,  $V_{\text{kofm}}$  und  $T_{\text{LB}}$  mit Hilfe von Kennfeldern die Kritikalitätswerte  $K_{T_{\text{MTTC}}}$ ,  $K_{V_{\text{phys}}}$ ,

$K_{V_{\text{komf}}}$  und  $K_{T_{\text{LB}}}$  zugeordnet. Auswahl und Parametrierung der Kennfelder erfolgte auf Grundlage in der Literatur genutzter Werte (vgl. [WGS13], [ZAG06]) mit Applikation am Fahr Simulator. Jedes Kennfeld gibt, abhängig vom Ausgangswert und dem Zeitabstand  $T_{\text{THW}}$  zwischen vorausfahrenden und folgendem Fahrzeug, einen Wert zwischen 0 und 10 zurück, wobei 0 die niedrigste und 10 die höchste Kritikalität angibt.

$T_{\text{THW}}$  wird oft selbst als Kritikalitätskennzahl verwendet. Da sie jedoch sehr geschwindigkeitsabhängig und daher wenig verallgemeinerbar ist, wird sie in dieser Arbeit nur genutzt, um den Abstand zwischen den Fahrzeugen bei der Kritikalitätsbewertung gezielt zu berücksichtigen. Die durch das Kennfeld definierte Kritikalität steigt z.B. für  $T_{\text{MTTC}}$  für kleinere Werte von  $T_{\text{THW}}$ . Auf diese Weise wird beispielsweise auch eine Situation, bei der zwei Fahrzeuge mit gleicher Beschleunigung und Geschwindigkeit, aber sehr geringem Abstand fahren, als sehr kritisch bewertet. Ohne diese Berücksichtigung wiese sie keine Kritikalität auf.

Die Gesamtkritikalität  $K_{\text{Ges}}$  einer Situation ist der höchste Rückgabewert aller Kritikalitätskennfelder.

$$K_{\text{Ges}} = \max (K_{T_{\text{MTTC}}}, K_{V_{\text{phys}}}, K_{V_{\text{komf}}} \text{ und } K_{T_{\text{LB}}}) \quad (5.17)$$

## 6 Ergebnisse

Zur Bewertung des Ansatzes passives HAF werden die vorgestellten Methoden auf ihre Leistungsfähigkeit hin überprüft. Diese Überprüfung gliedert sich in die drei Teile Fahrermodellierung, Szenarioerstellung und Kritikalitätserkennung. Für jeden einzelnen soll geprüft werden, ob eine hinreichende Qualität für den Einsatz im passiven HAF erreicht werden kann. Nur im Zusammenspiel der beschriebenen drei Teile kann eine Absicherung durch Einsatz des passiven HAF erfolgen.

### 6.1 Validierung der Fahrermodellierung

Die Fahrermodellierung in dieser Arbeit gliedert sich in die Modellierung des Längs- und des Querverhaltens der Fahrer. Während die Modellierung in Längsrichtung auf dem etablierten Modell nach [Wie74] beruht wurden für die Modellierung der von Fahrern getroffenen Spurwechselentscheidungen eigene Ansätze entwickelt. Deren Leistungsvermögen wird im Folgenden untersucht.

Zur Modellierung von Spurwechselentscheidungen wurden drei verschiedene Ansätze vorgestellt: Bayes'sche Netze, Input-Output Hidden Markov Modelle und tiefe feedforward neuronale Netze. Zur Bestimmung des geeignetsten Ansatzes wird zunächst die Reproduzierbarkeit der Spurwechselentscheidungen von simulierten Fahrern überprüft. Dazu werden die Modelle mit simulativ generierten Trainingsdaten trainiert und ihre Leistungsfähigkeit anhand separat generierter Testdaten ermittelt. Der leistungsfähigste der drei Ansätze wird für die Nutzung im passiven HAF weiterverwendet. Hierzu wird das mit Simulationsdaten vortrainierte Modell mit in der Realität aufgenommenen Daten weitertrainiert. Zuletzt erfolgt ein fahrertyp-individuelles Training. Dazu werden die Sensordaten aus Einzelfahrten genutzt und gemäß dem in Abbildung 3.2 dargestellten Prozess einzelnen Fahrertypen zugeordnet. Das Ergebnis sind Modelle, die im folgenden Schritt, der Resimulation, zum Einsatz kommen.

#### 6.1.1 Modellierung mit Simulationsdaten

Das Hauptkriterium für die Leistungsfähigkeit der Ansätze ist ihre Fähigkeit, das Verhalten der Fahrer möglichst exakt abbilden zu können. Weitere Kriterien sind die Komplexität der Modellierung und die für die Generierung der Spurwechselentscheidungen am Modellausgang notwendige Rechenzeit.

### 6.1.1.1 Datengrundlage

Für die erste Bewertung der Leistungsfähigkeit der Modellierungsansätze soll anhand von Simulationsdaten überprüft werden, in wie weit die Modelle das Spurwechselverhalten der simulierten Fahrer reproduzieren können. Darauf folgend wird das beste Modell als Ausgangspunkt für das Training mit Realdaten verwendet.

Mit Hilfe der Simulationsumgebung PELOPS (vgl. Abschnitt 3.3) wurden Daten über eine Fahrtdauer von 60 000 s in einem Szenario mit insgesamt 50 Fahrzeugen auf einer Autobahnstrecke generiert. Von den generierten Daten werden insgesamt 107502 Datenpunkte mit einer Abtastzeit von  $\Delta t = 0,3$  s zwischen den Punkten genutzt. Der ursprüngliche Datensatz wurde dabei so reduziert, dass die Label Spurhalten, Spurwechsel Links und Spurwechsel Rechts in gleicher Anzahl in den Trainingsdaten enthalten sind. Dabei besteht ein Spurhalte-Szenario aus einer Sequenz von Daten, die nur das Label Spurhalten tragen, während das Label bei den Spurwechsel-Szenarien von 'Spurwechsel zu Spurhalten wechselt. Dazu wird die Fahrzeugmitte, gemessen an der vorderen Stoßstange, betrachtet und ausgewertet, wann diese von der bisherigen Spur auf die neue Spur übertritt (vgl. Abbildung 4.2).

Von den generierten Daten werden 75 % zum Training der Modelle verwendet und 25 % als Testdaten zur Überprüfung der Klassifikationsgüte. Hierdurch wird eine Überanpassung der Modelle an die Trainingsdaten verhindert.

### 6.1.1.2 Bewertungskriterien

Die Bewertung der Leistungsfähigkeit erfolgt, indem die Modelle nach dem Training die Testdaten als Eingangsdaten erhalten und ihr Modellausgang mit den vorher für die Testdaten erzeugten Labels verglichen wird. Kriterium für die Güte der Klassifikation ist die Anzahl der richtig erkannten Labels. Hierbei wird zwischen vier Arten von Klassifikation unterschieden:

TP Bei richtig positiver (**T** rue **P** ositive) Klassifizierung ist die Spurwechselentscheidung (Spurwechsel oder Spurhalten) vorhanden und wird auch als vorhanden klassifiziert.

FN Bei falsch negativer (**F** alse **N** egative) Klassifizierung ist die Spurwechselentscheidung vorhanden, wird aber als nicht vorhanden klassifiziert.

FP Bei falsch positiver (**F** alse **P** ositive) Klassifizierung ist die Spurwechselentscheidung nicht vorhanden, wird aber als vorhanden klassifiziert.

TN Bei richtig negativer (**T** rue **N** egative) Klassifizierung ist die Spurwechselentscheidung nicht vorhanden und wird auch als nicht vorhanden klassifiziert.

Eine Bewertung der Modellqualität ist nur unter Berücksichtigung aller vier Kriterien möglich. So ist beispielsweise eine hohe Anzahl an richtig positiven Klassifikationen von Spurwechseln erreichbar, indem der Klassifikator stets und unabhängig von am Eingang anliegenden Daten einen Spurwechsel klassifiziert. Dies hätte jedoch zur Folge, dass die Anzahl an falsch positiven Klassifikationen ebenfalls sehr hoch wäre. Gleiches gilt für den umgekehrten Fall eines Klassifikators, dessen Ausgang nie einen Spurwechsel anzeigt, sodass die Anzahl falsch positiver Klassifizierungen niedrig wäre, aber die falsch negativer sehr hoch. Daher werden die genannten Kriterien in Verhältnis zueinander gesetzt.



TPR Die Richtig-Positiv-Rate (**True Positive Rate**) entspricht einer Trefferquote. Sie gibt den Anteil an richtig positiven Klassifizierungen an allen positiven Labels an:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (6.1)$$

FPR Die Falsch-Positiv-Rate (**False Positive Rate**) entspricht einer Fehlalarmquote. Sie gibt den Anteil an falsch positiven Klassifizierungen an allen tatsächlich negativen Labels an:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (6.2)$$

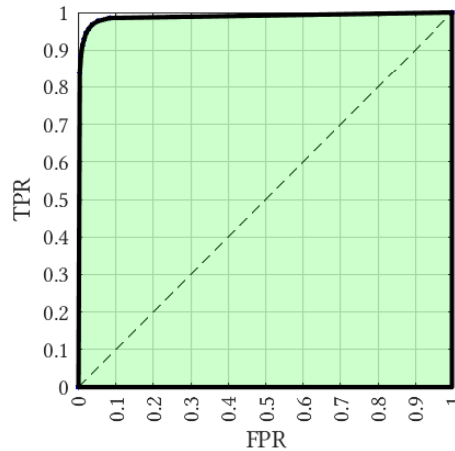
Die probabilistische Natur der Ausgänge der gewählten Modelle mit dem Wertebereich  $[0, 1]$  erlaubt es, Schwellwerte zu bestimmen, ab denen eine Klassifikation als positiv oder negativ gilt. So kann der für den jeweiligen Anwendungsfall optimale Kompromiss aus hoher Sensitivität (TPR) und geringer Falsch-Positiv-Rate gewählt werden. Im Fall der Spurwechselmodellierung sollen sowohl alle vorhandenen Spurwechsel erkannt werden, als auch keine nicht vorhandenen Spurwechsel fehlerhaft als Spurwechsel ausgegeben werden. Da das Ziel eine möglichst realitätsgetreue Modellierung in beiderlei Hinsicht ist, wird kein spezieller Fokus auf eins der beiden Kriterien TPR oder FPR gelegt, sondern nach dem Optimum aus beiden Kriterien gesucht.

Als Unterstützung zur Bestimmung der optimalen Schwellwerte dient die Grenzwertoptimierungskurve (**Receiver Operator Characteristic, ROC**). In dieser werden für bestimmte Schwellwerte die FPR auf der X- und die TPR auf der Y-Achse aufgetragen. Eine perfekte Klassifikation liegt dabei in der oberen linken Ecke bei den Koordinaten  $(0, 1)$ . Die ROC wird auf Grundlage der generierten Daten für die drei vorgestellten Ansätze erzeugt, um einen Vergleich ihrer Qualität zu ermöglichen. Um neben dem grafischen Vergleich auch einen numerischen Vergleich zu ermöglichen, wird zusätzlich die Fläche unter der Grenzwertoptimierungskurve berechnet (**Area under Curve, AUC**). Je höher die TPR bei niedriger FPR ist, desto steiler steigt die ROC an und desto größer ist somit die AUC. Beispielhaft sind ROC und AUC in Abbildung 6.1 dargestellt.

### 6.1.1.3 Bewertung

Die Bewertung der Modellierungsansätze Bayes'sches Netz, Input-Output Hidden Markov Model und tiefes neuronales Netzwerk erfolgt nach den oben genannten Kriterien. In der Abbildung 6.2 sind die Grenzwertoptimierungskurven jeweils für die drei Klassen Spurhalten, Spurwechsel Links und Spurwechsel Rechts sowie eine vergleichende Darstellung mit Kurven einer Klasse und aller Ansätze dargestellt. Die AUCs der verschiedenen Modellierungsansätze werden in Tabelle 6.1 für die einzelnen Klassen und als Durchschnittswert aller Klassen dargestellt. Für die ROC-Erzeugung und AUC-Berechnung wurden zehn Werte im Intervall  $[0, 1]$  in Schritten von 0,1 gewählt, um eine Vergleichbarkeit der Kurven zu ermöglichen.

Innerhalb der präsentierten Ergebnisse gelingt es der Modellierung mit DNN am besten das Spurwechselverhalten der simulierten Fahrer zu reproduzieren. Dies wird anhand der ROCs deutlich, bei denen die anderen Verfahren deutlich häufiger Spurhalten falsch positiv klassifizieren und somit die Klassifikationsgüte für die Klasse Spurhalten im Vergleich besonders beim



**Abbildung 6.1:** Die ROC entspricht der schwarz gezeichneten Linie. Durch diese wird die grüne Fläche nach oben begrenzt. Diese Fläche entspricht der AUC. Die gestrichelte Linie entspricht einer ROC bei einem rein zufällig entscheidenden Klassifikator

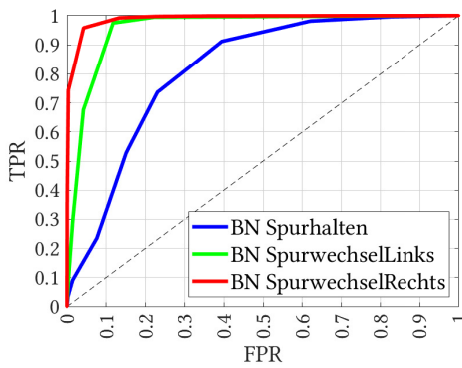
**Tabelle 6.1:** Flächen unter den Grenzwertoptimierungskurven (AUC) der drei Modellierungsansätze

Modell	Spurhalten	Spurwechsel Links	Spurwechsel Rechts	Durchschnitt
BN	0,816	0,957	0,990	0,921
IOHMM	0,932	0,904	0,969	0,935
DNN	0,988	0,992	0,992	0,991

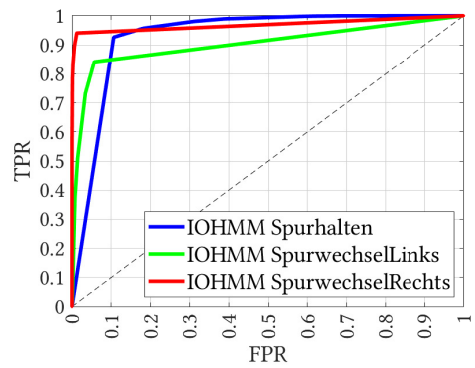
BN-Ansatz niedriger ist. Ähnliches gilt für die Klassifikation des Spurwechsels Links, welcher insbesondere beim IOHMM-Verfahren zu vielen falsch positiven Klassifikationen führt. Einzig mit der DNN-Methode wird in allen Klassen eine konstant hohe Klassifikationsgüte erreicht. Dies wird auch beim numerischen Vergleich der Flächen unter der Kurve deutlich, bei dem das DNN-Verfahren in allen Klassen die größte Fläche aufweist.

Neben dem Hauptkriterium Klassifikationsgüte wurde die Zeit gemessen, welche die verschiedenen Modellierungsmethoden zur Klassifikationen der simulierten Daten benötigen. Für den Geschwindigkeitsvergleich wurde die Klassifikation für 100 Sequenzen mit einer Länge von jeweils 10 s durchgeführt. Bei der genannten Abtastzeit von  $\Delta t = 0,3$  s führt dies zu insgesamt 3400 Datenpunkten, für die eine Klassifikation erfolgte. Die Ergebnisse sind in Tabelle 6.2 dargestellt.

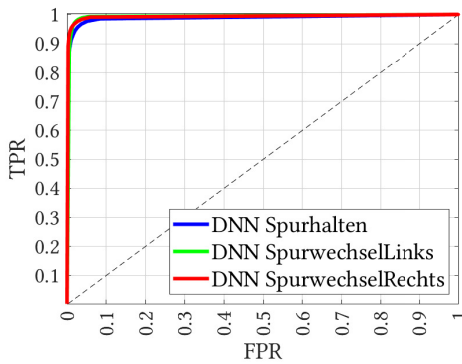
Die Zeitmessung wurde dabei direkt vor Übergabe der Eingangsdaten an die Klassifikatoren gestartet und direkt nach Rückgabe der Ausgangsdaten gestoppt. Zeit für die Erzeugung der Datenstrukturen und Modelle ist nicht inbegriffen. Die Klassifikationen wurden auf einem Intel Xeon E5-1630v3@3.70GHz Prozessor durchgeführt, bei dem alle vier Kerne parallel genutzt wurden. Die Implementierung der Ansätze erfolgte im Programm Matlab 2017b des Unternehmens Mathworks Inc. Dabei zeigt sich eine deutliche höhere Klassifikationsgeschwindigkeit



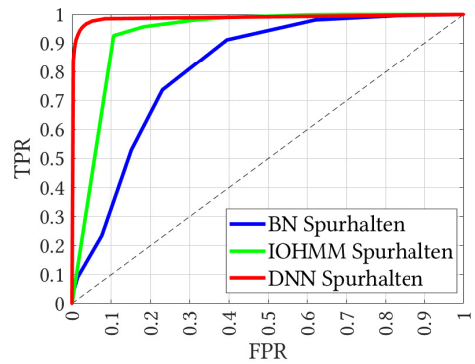
(a) BN



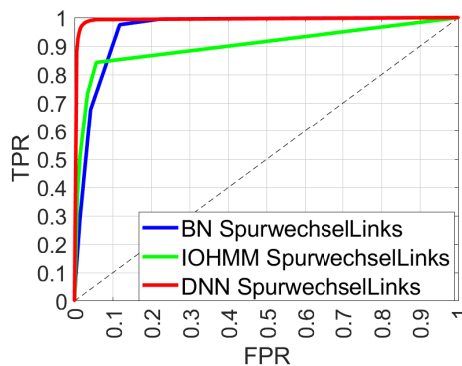
(b) IOHMM



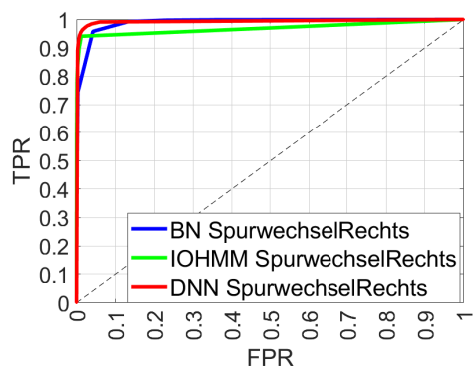
(c) DNN



(d) Spurhalten



(e) Spurwechsel Links



(f) Spurwechsel Rechts

**Abbildung 6.2:** Grenzwertoptimierungskurven der genutzten Klassifikationsmethoden auf simulierten Testdaten nach Training mit simulierten Daten. Dargestellt ist zum einen die Klassifikationsgüte für die einzelnen Klassen bei Modellierung mit den Methoden BN, IOHMM und DNN, zum anderen die Klassifikationsgüte der einzelnen Methoden für die Klassen Spurhalten, Spurwechsel Links und Spurwechsel Rechts

**Tabelle 6.2:** Klassifikationsdauer der einzelnen Verfahren zur Spurwechselklassifikation von 3400 Datenpunkten

Methoden	Rechenzeit
BNN	279 s
IOHMM	58 s
DNN	2 s

des DNN gegenüber der Geschwindigkeit der anderen Verfahren trotz des für Bayes'sche Netze angewendeten besonders effizienten Inferenzverfahrens.

Die Verfahren wurden zu einem unterschiedlichen Grad durch Nutzung vorhandener Funktionen und eigener Programmierung implementiert. Während der Eigenanteil bei den Bayes'schen Netzen besonders hoch ist, wurden die Ausgangsdaten der neuronalen Netze bei den Ansätzen IOHMM und DNN mit Hilfe von Matlab-internen Funktionen erzeugt, welche die zur Verfügung stehende Rechenleistung effizienter nutzen. Dies erklärt einen Teil der stark abweichenden Klassifikationsdauer. Aufgrund der weitaus aufwändigeren Inferenz in BN und IOHMM gegenüber der Klassifikation mit neuronalen Netzen ist die in Tabelle 6.2 dargestellte Reihenfolge der Methoden hinsichtlich ihrer Geschwindigkeiten dennoch plausibel.

Der Methodenvergleich fällt somit eindeutig zu Gunsten der DNN-Modellierungsmethode aus. Neben der hohen Klassifikationsleistung gewinnt diese Methode vor allem gegenüber dem BN-Verfahren aufgrund der deutlich geringeren Rechenzeit des Modells.

Nachteilig ist die für Menschen nicht verständliche Modellierung, wodurch das DNN zum Black-Box Modell wird. Hier bietet der BN-Ansatz deutliche Vorteile, da eine eindeutige Modellstruktur aus Zufallsvariablen gewählt werden kann und im Zuge des Trainings für jede Zufallsvariable eine Verteilungsfunktion trainiert wird.

Das Modellverständnis ist vor allem dann entscheidend, wenn auf Grundlage der Modellausgänge sicherheitskritische Entscheidungen gefällt werden, bspw. wenn das Modell zur Spurwechsel-Prädiktion anderer Fahrzeuge genutzt wird und daraufhin das HAF-Fahrzeug mit Ausweich- oder Bremsmanövern reagiert wie in [RKG<sup>+</sup>19]. Ist das dort verwendete Modell unverständlich und lässt sich daher nicht auf Fehler überprüfen widerspricht dies den Zielen einer sicherheitsgerichteten Entwicklung nach der hierfür einschlägigen Industrienorm ISO 26262 [Int11], da die Mehrzahl der empfohlenen Analysemethoden nicht uneingeschränkt angewendet werden können.

Im Fall der Nutzung im Rahmen des passiven HAF bewirken einzelne auftretende Fehler eine für den Zeitpunkt des Fehlers nicht realitätsgetreue Simulation des fiktiven Verlaufs des Szenarios. Dies hat aber keine direkte Sicherheitsauswirkung, da der Fehler nur in der Simulation auftritt und somit nicht zu einem Personenschaden führen kann. Zum anderen haben einzelne Fehler auch keinen signifikanten Einfluss auf die Aussagekraft des Ansatzes „passives HAF“, da sie innerhalb der geplanten Menge an zu sammelnden Daten (vgl. Unterabschnitt 3.1.2) nur einen sehr geringen Teil ausmachen. Häufige Fehler hingegen würden im Rahmen des Simulations-Realitäts-Abgleichs (vgl. Abbildung 3.2) entdeckt. Bei diesem wird kontinuierlich überprüft, ob die Simulation in der Lage ist, das reale Szenario auch bei Fehlen einer Abweichung zwischen den Fahrentscheidungen des menschlichen Fahrers und des HAF-Algorithmus nachzusimulieren. Gelingt dies für ein Szenario nicht, werden die aufgezeichneten Daten zum

Training eines weiteren Fahrermodells genutzt. Hierdurch lernt das passive HAF, auch bisher unbekannte Fahrertypen zu simulieren. Aus diesen Gründen werden im Folgenden trotz des für Menschen nicht analysierbaren Aufbaus von neuronalen Netzen DNN zum Einsatz in der Simulation für das passive HAF verwendet.

## 6.1.2 Modellierung mit Realdaten

Nach Auswahl des leistungsfähigsten Ansatzes auf Grundlage von Simulationsdaten wird in diesem Abschnitt bewertet, welche Klassifikationsgüte DNN auf im realen Straßenverkehr aufgenommenen Daten erreichen. Dazu wurde das mit simulierten Daten trainierte neuronale Netz zunächst mit allen Trainings-Realdaten weitertrainiert und seine Leistungsfähigkeit mit Test-Realdaten überprüft.

Als Datengrundlage dienen von einem 5er BMW gesammelte Sensordaten, die auf Autobahnen und autobahnähnlichen Straßen mit baulicher Trennung der Spuren unterschiedlicher Fahrrichtung gesammelt wurden. Für die Spurerkennung wurde eine Monokamera der Firma Mobileye verwendet, für die Objekterkennung ein 360° LIDAR der Firma IBEO. Nach der Datenvorverarbeitung können umgebende Verkehrsteilnehmer mit guter Qualität bis zu 100 m vor und hinter dem Sensorfahrzeug auf dessen Spur und direkt angrenzenden Nachbarspuren erfasst werden. Daher wird dieser Bereich für die Szenarioerstellung genutzt. Als Realdaten stehen Sensordaten über eine Aufzeichnungszeit von 32 400 s zur Verfügung, die über eine Strecke von 1100 km auf neun Autobahnfahrten gesammelt wurden. Die resultierende Grenzwertoptimierungskurve ist in Abbildung 6.3 dargestellt, die Flächeninhalte unter den Kurven in Tabelle 6.3.

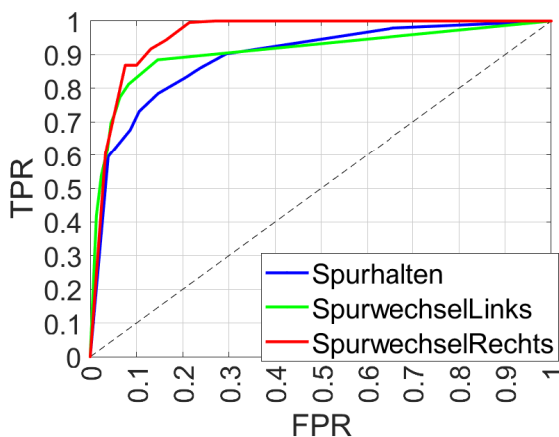


Abbildung 6.3: ROC des mit realen Daten trainierten und evaluierten Netzes

Das so trainierte allgemeine Netz wird weiterverwendet, um damit wie in Abschnitt 3.1.1 beschrieben individuelle Fahrertypen zu trainieren. Im Zuge dessen wurden mit den verfügbaren Realdaten fünf Fahrermodelle gemäß dem Vorgehen in Abbildung 3.2 erzeugt. Ein neues Fahrermodell wurde dabei immer dann trainiert, wenn keins der bisher trainierten Modelle auf dem neuen Realdatensatz eine AUC > 0,95 aufwies. Die Ergebnisse der einzelnen Fahrermodelle

sind in den Grenzwertoptimierungskurven in der Abbildung 6.4, die Flächeninhalte unter den jeweiligen Kurven in Tabelle 6.3 dargestellt.

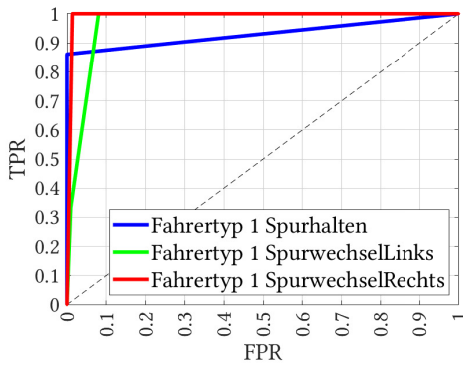
**Tabelle 6.3:** Flächen unter den Grenzwertoptimierungskurven der mit Realdaten trainierten und evaluierten DNN

Modell	Spurhalten	Spurwechsel Links	Spurwechsel Rechts	Durchschnitt
allg. DNN	0,892	0,908	0,955	0,918
Fahrertyp 1	0,931	0,968	0,993	0,964
Fahrertyp 2	0,954	0,981	0,971	0,969
Fahrertyp 3	0,882	0,914	0,982	0,926
Fahrertyp 4	0,917	0,950	0,978	0,948
Fahrertyp 5	0,977	0,988	0,990	0,985

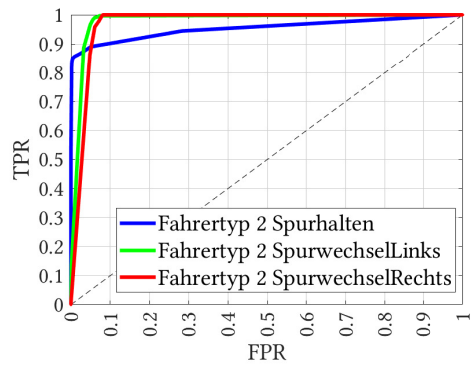
Insbesondere der Vergleich der durchschnittlichen AUC-Werte verdeutlicht, dass durch das fahrertypindividuelle Training der DNN eine deutlich erhöhte Klassifikationsgüte erreicht werden kann. Es ist anzunehmen, dass bei einem größeren Datensatz an Realdaten noch weitere Fahrertypen unterscheidbar und somit trainierbar sein werden.

Weiterhin fällt auf, dass die Klassifikationsgüte der DNN bei Realdaten gegenüber der in der Simulation abfällt. Mögliche Gründe hierfür sind zum einen die trotz der in Abschnitt 3.4 beschriebenen Vorverarbeitung schlechtere Datenqualität. Bedingt durch die Einschränkung der Sensorreichweite auf  $\pm 100$  m in longitudinaler Richtung können nicht alle möglicherweise für die Fahrentscheidungen von menschlichen Fahrern relevanten Faktoren mit verarbeitet werden. Zum anderen stehen für das Training der Modelle geringere Mengen an Realdaten zur Verfügung, als dies beim Training mit Simulationsdaten der Fall war.

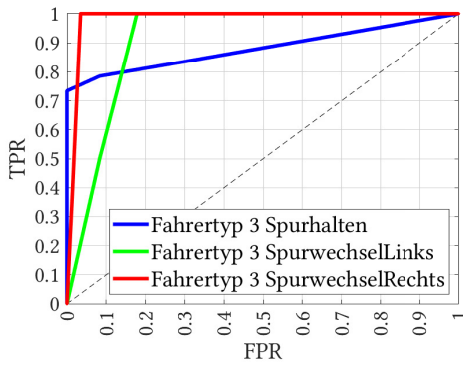
In diesem Abschnitt wurde plausibilisiert, dass durch Nutzung der trainierten Fahrermodelle das Verhalten der modellierten Fahrer präzise wiedergegeben werden kann. Im nächsten Schritt werden die Modelle in die Simulationsumgebung übertragen. Dort werden sie den das HAF-Fahrzeug umgebenden Verkehrsteilnehmern zugeordnet und für die Modellierung ihres Fahrverhaltens genutzt.



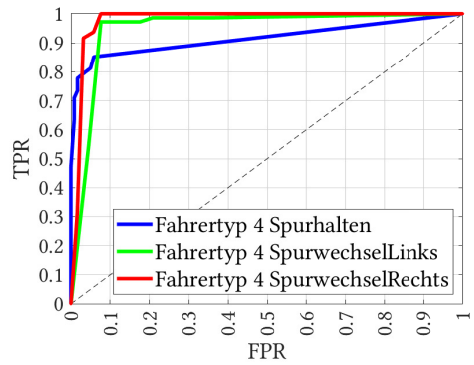
(a) Fahrertyp 1



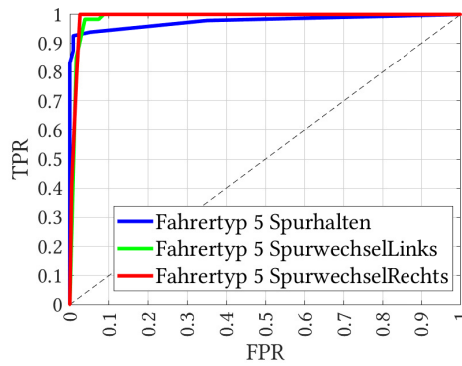
(b) Fahrertyp 2



(c) Fahrertyp 3



(d) Fahrertyp 4



(e) Fahrertyp 5

Abbildung 6.4: ROCs der Evaluation des DNN nach fahrerindividuellem Training mit Realdaten

## 6.2 Verifikation der Szenarioerstellung

Für die Verifikation des Algorithmus zur Szenarioerstellung werden Simulationsdaten genutzt. Dies ermöglicht die Generierung einer Groundtruth („Feldvergleich“) aus dem Simulationssetup. Da das Datenformat der Eingangsgröße DrivingView zur Szenarioerstellung bei Simulations- und Realdaten identisch ist, ist das Ergebnis der Verifikation auch auf Realdaten übertragbar. Die Evaluierung der Szenarioerstellung auf Grundlage von Realdaten kann hingegen nur anhand einer Beispielsituation veranschaulicht und qualitativ bewertet werden, eine Quantifizierung der Rekonstruktionsgüte ist aus oben genanntem Grund weder möglich noch erforderlich.

### 6.2.1 Szenarioerstellung aus Simulationsdaten

Zur Verifizierung der Szenarioerstellung werden Simulationsszenarien mit zunehmender Komplexität verwendet, die in [Wit17] erarbeitet wurden. Die simulierten Daten werden dem Szenarioerstellungs-Algorithmus als DrivingView aus der Perspektive des HAF-Fahrzeugs zur Verfügung gestellt, analog der Realdatenbereitstellung im Fahrzeug durch Sensorik und Umfeldmodell. Für die Bewertung wird die Simulationsausgabe des originalen und des rekonstruierten Szenarios hinsichtlich der folgenden Werte verglichen:

- Startposition in longitudinaler Richtung
- Startposition in lateraler Richtung
- Startspur
- Startgeschwindigkeit
- Startbeschleunigung
- Wunschgeschwindigkeit des Fahrers
- im Originalszenario als Fahrermodell verwendetes bzw. im rekonstruierten Szenario bestimmtes neuronales Netz

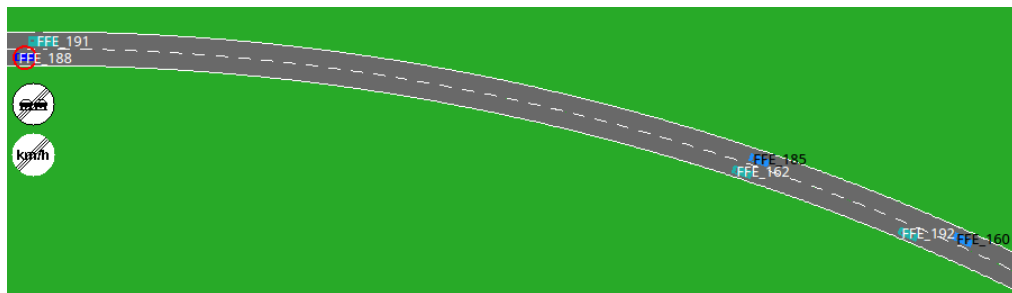
Die Szenariorekonstruktion startet, sobald eine signifikante Abweichung hinsichtlich des Spurwechselverhaltens zwischen den Fahrerentscheidungen des im Originalszenario verwendeten Fahrermodells und der des HAF-Algorithmus vorliegt, analog zum Verhalten in der realen Anwendung des passiven HAF.

#### 6.2.1.1 Rekonstruktion eines Szenarios mit konstanter Spuranzahl und konstanter Krümmung

Zunächst soll der Fall eines Szenarios mit konstanter Spuranzahl und konstanter Krümmung untersucht werden. Das für die Verifikation verwendete Szenario ist in Abbildung 6.5 dargestellt. Das im Bild rot gekennzeichnete Fahrzeug entspricht dabei dem Ego-Fahrzeug, dessen DrivingView als Eingang für die HAF-Funktion dient und welches somit bei Abweichungen zwischen ursprünglichem Fahrermodell und HAF-Funktion die Szenario-Resimulation auslöst.

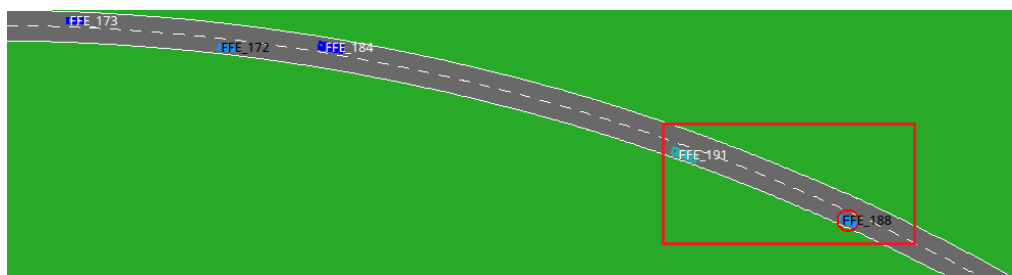


Gleichzeitig dient es als Testfahrzeug, anhand dessen die Qualität der Szenariorekonstruktion gemessen wird.



**Abbildung 6.5:** Originalszenario 1 für Rekonstruktion. Aus [Wit17, S.59]

Zum Zeitpunkt  $t = 6,96\text{s}$  wird aufgrund von Abweichungen zwischen dem Fahrverhalten des Fahrers und der HAF-Funktion eine Resimulation des Szenarios gestartet. Zu diesem Zeitpunkt befinden sich die in Abbildung 6.6 dargestellten Fahrzeuge innerhalb des für die Berücksichtigung von anderen Verkehrsteilnehmern relevanten Bereiches von  $\pm 100\text{m}$  um das Testfahrzeug. Alle weiteren in Abbildung 6.5 dargestellten Fahrzeuge werden für den weiteren Verlauf der Resimulation nicht berücksichtigt. Diese Beschränkung wäre in den gewählten simulationsbasierten Testszenarien nicht notwendig, da in diesen eine perfekte Datenqualität vorliegt. Da der Resimulation-Algorithmus aber nicht gegenüber dem für reale Szenarien verwendeten abgeändert werden soll, um die bei realer Nutzung im Kontext des passiven HAF auftretenden Schwächen des Algorithmus erkennen zu können, wird die Beschränkung beibehalten.



**Abbildung 6.6:** Nur Fahrzeuge innerhalb des roten Bereichs werden aufgrund der in der Realität beschränkten Sensorreichweite in der weiteren Simulation von Szenario 1 berücksichtigt. Aus [Wit17, S.60]

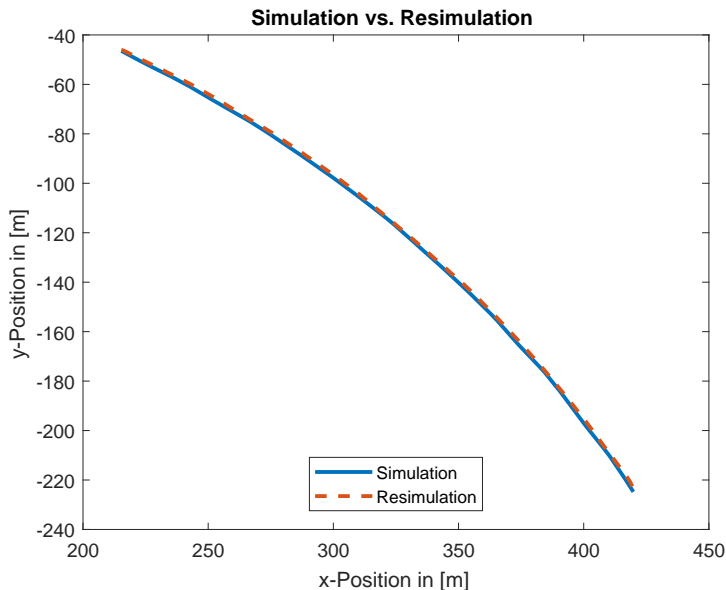
Die für die Bewertung der Rekonstruktion herangezogenen Werte sind für den Startzeitpunkt der Resimulation in Tabelle 6.4 dargestellt. Dabei zeigt der Vergleich, dass die Resimulation der Simulation entspricht. Die Wunschgeschwindigkeit weicht in der Resimulation vom in der Simulation vorgegebenen Parameter ab. Dies liegt an dem Verfahren der Schätzung der Wunschgeschwindigkeit, welches für diese den höchsten Geschwindigkeitswert innerhalb eines Beobachtungszeitraums verwendet. Da der maximale Geschwindigkeitswert jedoch in

der gegebenen Simulation nie die tatsächliche Wunschgeschwindigkeit erreicht hat, konnte letztere für die Rekonstruktion nicht korrekt ermittelt werden. Da sich aber kein langsames Vorderfahrzeug in unmittelbarer Nähe befindet, gibt es in diesem Szenario keinen Anlass für den Fahrer des Ego-Fahrzeugs, einen Spurwechsel durchzuführen, um die Wunschgeschwindigkeit zu erreichen.

**Tabelle 6.4:** Bewertungsvariablen zum Startzeitpunkt der Resimulation von Szenario 1. Nach [Wit17, S.60]

Parameter	Simulation	Resimulation
$x_0$ [m]	222,50	222,50
$y_0$ [m]	-0,06	-0,06
Spur	1	1
$v_0$ [m/s]	33,28	33,28
$a_0$ [m/s <sup>2</sup> ]	0,00	0,00
$v_{\text{wunsch}}$ [m/s]	33,77	33,28
DNN von Fahrertyp	3	3

Weiterhin sind in den Abbildungen 6.7 und 6.8 die  $x$ - und  $y$ -Positionen des Testfahrzeugs übereinander sowie seine Geschwindigkeit über die Simulationszeit dargestellt. Auch hier wird deutlich, dass die Resimulation der Simulation bis auf die genannten Abweichungen entspricht. Lediglich die Geschwindigkeitsabsenkung am Ende der Simulationszeit ist in der Resimulation aufgrund des fehlenden Vorderfahrzeugs nicht zu beobachten. Dies erklärt die geringfügigen Abweichungen hinsichtlich der  $x$ - und  $y$ -Positionen am Ende der Simulationszeit.



**Abbildung 6.7:**  $x$ - und  $y$ -Position des Testfahrzeugs in Simulation und Resimulation von Szenario 1

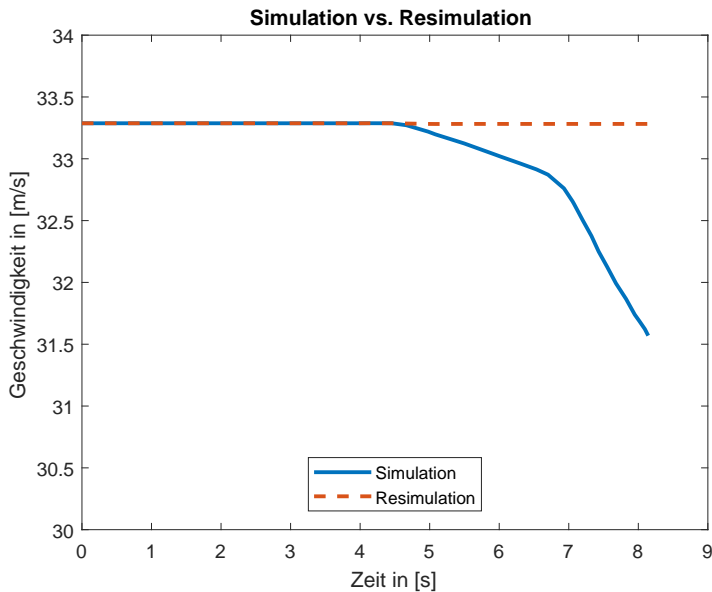


Abbildung 6.8: Geschwindigkeit des Testfahrzeugs in Simulation und Resimulation von Szenario 1. Aus [Wit17, S.62]

### 6.2.1.2 Rekonstruktion eines Szenarios mit steigender Spuranzahl und konstanter Krümmung

Im zweiten getesteten Szenario erhöht sich die Spuranzahl von zwei auf drei Spuren ohne Spürkrümmung. Das Ausgangsszenario in der Originalsimulation ist in Abbildung 6.9 abgebildet, das Testfahrzeug ist wiederum rot markiert und der rote Bereich markiert die Fahrzeuge, die für die Resimulation berücksichtigt werden.



Abbildung 6.9: Originalszenario 2 für Rekonstruktion. Aus [Wit17, S.63]

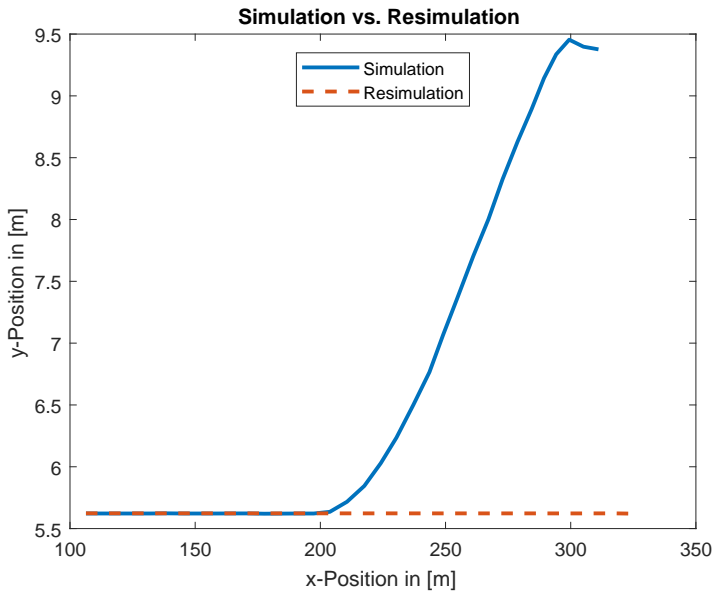
Die Resimulation des Szenarios wird  $t = 3,38$  s nach Start der Simulation gestartet. Die für die Bewertung herangezogenen Werte sind für diesen Zeitpunkt in Tabelle 6.5 dargestellt.

In Abbildung 6.10 sind die  $x$ - und  $y$ -Positionen übereinander für die Originalsimulation und die Resimulation aufgetragen, in Abbildung 6.11 die Geschwindigkeit des Testfahrzeugs. Hierbei ist auffällig, dass gegen Ende der Simulation sowohl  $x$ - als auch  $y$ -Position in der Resimulation vom Original abweichen. Wie im ersten Szenario liegt der Grund hierfür in einem vorausfahrenden Fahrzeug, dargestellt im rechten Teil von Abbildung 6.9, welches nicht in der Resimulation berücksichtigt wird, da es zum Zeitpunkt des Resimulationsstarts außerhalb des Sensorbereichs liegt. Im Verlauf der Originalsimulation wird das Testfahrzeug zunächst kurz ausgebremst und

**Tabelle 6.5:** Bewertungsvariablen zum Startzeitpunkt der Resimulation von Szenario 2. Nach [Wit17, S.63]

Parameter	Simulation	Resimulation
$x_0$ [m]	106,48	106,48
$y_0$ [m]	0,00	0,00
Spur	2	2
$v_0$ [m/s]	36,10	36,10
$a_0$ [m/s <sup>2</sup> ]	0,60	0,60
$v_{\text{Wunsch}}$ [m/s]	36,10	36,10
DNN von Fahrertyp	3	3

überholt dann das vorausfahrende Fahrzeug, insbesondere erkennbar an dem Sprung in der  $y$ -Position.

**Abbildung 6.10:**  $x$ - und  $y$ -Position des Testfahrzeugs in Simulation und Resimulation von Szenario 2

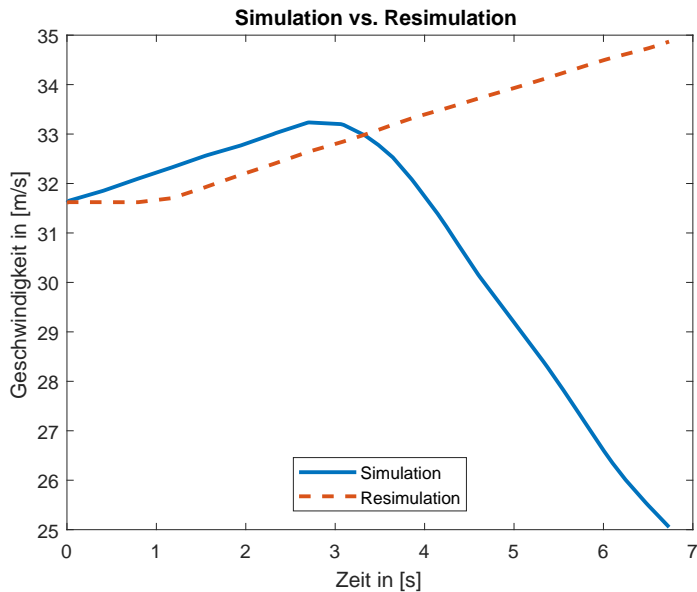


Abbildung 6.11: Geschwindigkeit des Testfahrzeugs in Simulation und Resimulation von Szenario 2

### 6.2.1.3 Rekonstruktion eines Szenarios mit konstanter Spuranzahl und wachsender Krümmung

Im dritten Testszenario wächst die Krümmung der Spuren von einer Gerade zu einer Linkskurve während die Spuranzahl konstant bleibt. Das Originalszenario ist in Abbildung 6.12 dargestellt. Ziel dieses Szenarios ist es vor allem zu prüfen, welchen Einfluss die Abtastzeit von  $\Delta t = 10$  ms auf die Qualität der Rekonstruktion der Szenarios hat.



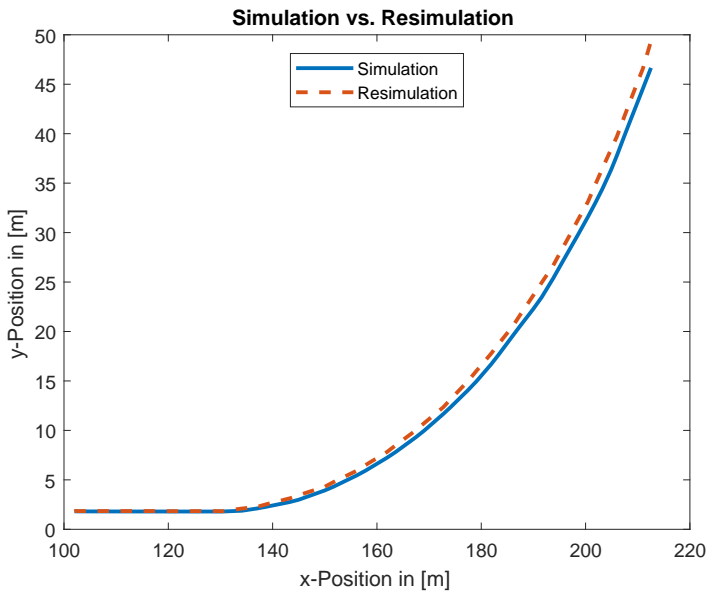
Abbildung 6.12: Originalszenario 3 für Rekonstruktion. Aus [Wit17, S.66]

Die Startwerte sind in Tabelle 6.6 dargestellt.

**Tabelle 6.6:** Bewertungsvariablen zum Startzeitpunkt der Resimulation von Szenario 3. Nach [Wit17, S.66]

Parameter	Simulation	Resimulation
$x_0$ [m]	102,24	102,24
$y_0$ [m]	0,00	0,00
Spur	1	1
$v_0$ [m/s]	23,47	23,47
$a_0$ [m/s <sup>2</sup> ]	-2,90	-2,90
$v_{\text{Wunsch}}$ [m/s]	33,77	33,48
DNN von Fahrertyp	3	3

Aus Abbildung 6.13 wird ersichtlich, dass die im Bereich der Geraden zunächst fehlerfreie Rekonstruktion des Szenarios aufgrund der abschnittsweise als konstant angenommenen Krümmung (das Testfahrzeug fährt innerhalb eines Zeitschritts ca. 3 bis 4 m) zunehmend vom Originalszenario abweicht, obwohl in diesem Fall kein in der Resimulation unberücksichtigtes Fahrzeug einen Einfluss ausübt. Der Fehler würde sich durch eine Reduzierung der Abtastrate verringern.

**Abbildung 6.13:**  $x$ - und  $y$ - Position des Testfahrzeug in Simulation und Resimulation von Szenario 3. Nach [Wit17, S.68]

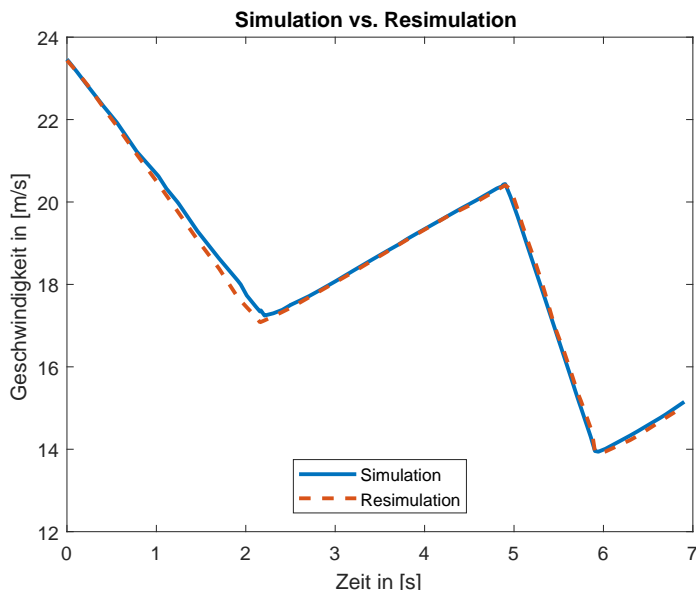


Abbildung 6.14: Geschwindigkeit des Testfahrzeugs in Simulation und Resimulation von Szenario 3

#### 6.2.1.4 Rekonstruktion eines Szenarios mit steigender Spuranzahl und wachsender Krümmung

Zuletzt wird in einem Szenario mit steigender Spuranzahl und wachsender Krümmung die Qualität der Szenariorekonstruktion am komplexesten Beispiel überprüft. Das Originalszenario ist in Abbildung 6.15 dargestellt.

Da die Resimulation erst zu einem Zeitpunkt gestartet wird, zu dem sich das Testfahrzeug bereits nicht mehr im geraden Abschnitt des Szenarios befindet, weicht die Startposition bereits hier geringfügig im  $x$ - und  $y$ -Wert ab. Alle Bewertungsvariablen sind in Tabelle 6.7 dargestellt.

Tabelle 6.7: Bewertungsvariablen zum Startzeitpunkt der Resimulation von Szenario 4. Nach [Wit17, S.70]

Parameter	Simulation	Resimulation
$x_0$ [m]	24,21	24,27
$y_0$ [m]	0,01	-0,04
Spur	2	2
$v_0$ [m/s]	27,53	27,53
$a_0$ [m/s <sup>2</sup> ]	-8,50	-8,50
$v_{\text{Wunsch}}$ [m/s]	33,77	33,48
DNN von Fahrertyp	3	3



Abbildung 6.15: Originalszenario 4 für Rekonstruktion. Aus [Wit17, S.69]

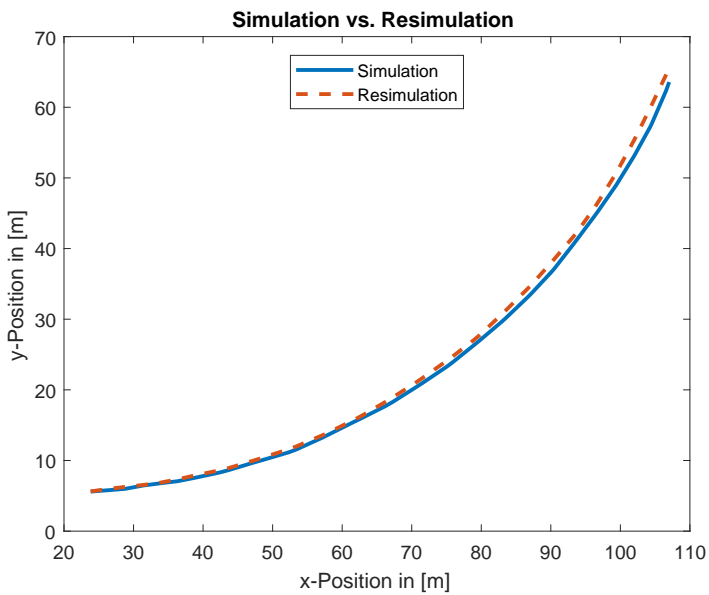


Abbildung 6.16: Hinsichtlich  $x$ - und  $y$ -Position des Testfahrzeugs weichen Simulation und Resimulation in Szenario 4 voneinander ab. Nach [Wit17, S.70]



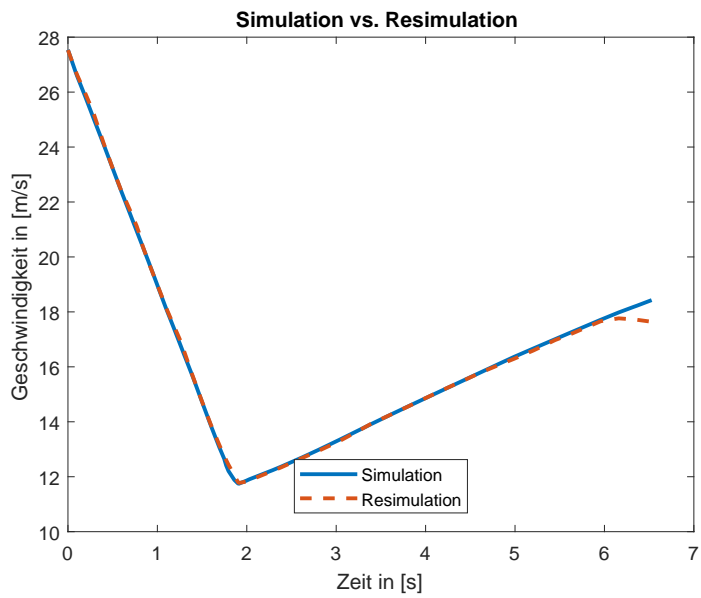


Abbildung 6.17: Geschwindigkeit des Testfahrzeugs in Simulation und Resimulation von Szenario 4

Auch hinsichtlich des zurückgelegten Weges in  $x$ - und  $y$ -Richtung ist eine mit Szenario 3 vergleichbare Abweichung festzustellen (siehe Abbildung 6.16), die wiederum durch die Abtastrate zu erklären ist.

Die Resimulation der gewählten Testszenerien gelingt insgesamt gut. Zwei durch die Sensorik bedingte Fehlerquellen werden jedoch deutlich: Zum einen durch die begrenzte Sichtweite der Sensorik, durch die in einem Szenario möglicherweise relevante Fahrzeuge nicht enthalten sind. Zum anderen durch die Abtastrate, durch die sich kontinuierlich ändernde Krümmungen nicht fehlerfrei reproduzieren lassen. Eine Absicherung durch das passive HAF ist somit nur für die Szenarien möglich, in denen alle relevanten Verkehrsteilnehmern und Fahrspuren durch die Sensorik detektiert werden. Dies entspricht der in Abschnitt 3.1.1 genannten Einschränkung einer Absicherung ohne Referenzsensorik. Mit fortschreitender Entwicklung von Sensorik und Umfeldmodell und hierdurch verbesserter Umgebungswahrnehmung ist eine Erweiterung der Entfernungsgrenze für die genutzten Sensordaten möglich. Dies führt zu einer weiteren Erhöhung der Realitätstreue der simulierten Szenarien, ohne dass Änderungen am passiven HAF selbst notwendig wären.

## 6.2.2 Rekonstruktion eines Simulationsszenarios aus Realdaten

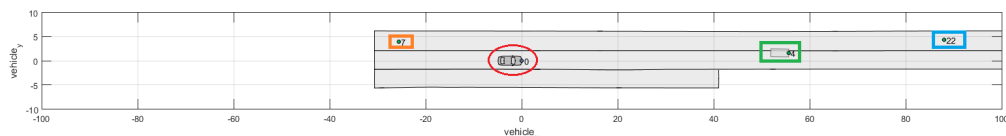
In diesem Abschnitt soll beispielhaft die Rekonstruktion einer Situation aus dem realen Verkehr präsentiert werden, da ohne Groundtruth eine allgemeine Überprüfung der Funktionsfähigkeit der Szenarioresimulation mit Realdaten nicht möglich ist. Die Ergebnisse wurden erstmals in [KWLH18] veröffentlicht.

In Abbildung 6.18 ist die zu rekonstruierende Situation aus dem Blickwinkel der Front- und Heckkamera dargestellt. Das mit Monokamera und LIDAR ausgestattete Testfahrzeug befindet sich auf einer zunächst fünfspurigen Autobahn, die sich auf Höhe des Fahrzeugs auf vier Spuren durch Wegfall der rechtesten Spur verengt. Das Testfahrzeug fährt bis zur Verengung auf der zweitrechtesten Spur, danach durch Spurhalten auf der rechtesten. Im Sichtbereich der Sensorik, die auf  $\pm 100$  m und  $\pm 1$  Spur um das Testfahrzeug beschränkt ist, befinden sich drei Fahrzeuge, die durch farbige Rechtecke in den Bildern markiert sind.



Abbildung 6.18: Kamerabilder des Testfahrzeugs

Die vom Umfeldmodell nach Vorverarbeitung der Sensordaten durch die in Abschnitt 3.4 beschriebenen akausalen Filter zur Verfügung gestellten Daten sind in Abbildung 6.19 dargestellt. Diese Daten werden dem Resimulations-Algorithmus als Driving View zur Verfügung gestellt. Dies entspricht dem Vorgehen, welches im vorangegangenen Abschnitt für Simulationsdaten angewendet wurde.



**Abbildung 6.19:** Visualisierung der vom Umfeldmodell zur Verfügung gestellten Daten. Aus [Wit17, S.72]

Das rekonstruierte Szenario ist in Abbildung 6.20 dargestellt und entspricht dem des Umfeldmodells. Auffällig sind die scheinbar fehlenden Spurlinien, die jedoch lediglich aufgrund von unmittelbaren aufeinanderfolgenden, sehr kurzen Streckenelementen nicht sichtbar sind. Diese werden generiert, da im Gegensatz zu den idealen Simulationsdaten die zur Verfügung gestellten Realdaten nie eine exakt konstante Krümmung der Strecke aufweisen und somit zu jedem Messzeitpunkt eine geringfügig andere Orientierung der Streckenelemente besteht. Dies hat jedoch keine Auswirkung auf die Qualität der Simulation. Es wird deutlich, dass die Rekonstruktion aus Realdaten möglich ist und zu guten Ergebnissen führt. Das vorgestellte Verfahren wird im folgenden Abschnitt genutzt, um auf realen Daten basierende Simulationsszenarien zu generieren und auf potentiell kritische Situationen zu analysieren.



**Abbildung 6.20:** Aus Realdaten rekonstruiertes Simulationsszenario

## 6.3 Validierung der Kritikalitätserkennung

Der Funktionsteil Kritikalitätserkennung soll daraufhin getestet werden, ob er in der Lage ist, kritische Situationen zu erkennen. Aufgrund der großen Datenmengen, die potentiell im Rahmen des Konzepts passives HAF bewertet werden müssen (vgl. Abschnitt 3.1.2), ist es für die effiziente Analyse und Fehlerbehebung unerlässlich, eine automatisierte Vorauswahl an potentiell kritischen Szenarien bereitzustellen. Hierfür werden alle resimulierten Szenarien nach den in Kapitel 5 vorgestellten Kriterien bewertet.

### 6.3.1 Kritikalitätserkennung bei Fahrt mit HAF-Funktion ohne implementierte Fehler

Zur Bewertung von potentiell kritischen Fahrsituationen werden sämtliche aufgezeichneten Realdaten, die im Rahmen von Testfahrten auf Autobahnen gesammelt wurden, mit Hilfe des in Abschnitt 3.2 vorgestellten Verfahrens resimuliert und die Resimulationen auf Kritikalität hin überprüft.

Insgesamt wurden in Testfahrten 1100 km an Realdaten als Datenbasis gesammelt. Dabei kam es in 230 Situationen zu einer signifikanten Abweichung des Fahrverhaltens der passiven HAF-Funktion von dem des menschlichen Fahrers, sodass eine Resimulation gestartet wurde. Bei keiner der simulierten Situationen überstieg einer der für die Kritikalitätsbewertung herangezogenen Messwerte den kritischen Schwellwert. Eine Analyse der visualisierten Simulationsdaten bestätigte dieses Ergebnis. Daraus kann geschlossen werden, dass zumindest innerhalb der genutzten Testdistanz durch die HAF-Funktion keine Fehler verursacht wurden, die zu einer kritischen Situation gemäß den in dieser Arbeit definierten Kriterien geführt hätten.

### 6.3.2 Kritikalitätserkennung bei Fahrt mit HAF-Funktion mit implementiertem Fehler

Das oben beschriebene Ergebnis ist aus Sicht der Funktionsentwicklung erfreulich, jedoch nicht geeignet, die Effektivität der Kritikalitätserkennung zu überprüfen. Daher wurde die innerhalb der resimulierten Szenarien verwendete HAF-Funktion künstlich verschlechtert. Hierzu wurde ein Fehler implementiert, der dazu führt, dass durchschnittlich alle 5 Sekunden ein Spurwechselwunsch in eine zufällige Richtung erzeugt wird. Dieser Wunsch führt zu einem Spurwechsel außer in dem Fall, dass dieser direkt zu einer Kollision mit einem Fahrzeug in der Nachbarspur führen würde. Hierdurch kam es in insgesamt 43 der 230 resimulierten Szenarien zu jeweils mindestens einer Situation, in der die maximale Kritikalität über den Schwellwert von 1 stieg und die somit als kritisch erkannt wurden. Die Kritikalität dieser Situationen resultiert regelmäßig daraus, dass das Testfahrzeug einen Spurwechsel von der linken auf die rechte Spur durchführt und danach stark bremsen muss, um eine Kollision mit dem langsamer fahrenden Verkehr auf der neuen, weiter rechts liegenden Spur zu verhindern. Dies gelingt in den meisten Fällen, in zwei Fällen kommt es jedoch zu einer Kollision. In den Szenarien, in denen hieraus keine kritische Situation entsteht, ist zum Zeitpunkt des Spurwechsels kein potentiell Vorderfahrzeug auf der Nachbarspur oder das Testfahrzeug befindet sich bereits in der langsameren Spur, sodass aus dem Spurwechselwunsch kein Spurwechsel resultiert.

Stellvertretend für die 43 als kritisch bewerteten Szenarien soll ein solches Szenario beschrieben werden. Die Ausgangssituation ist in Abbildung 6.21 dargestellt. Zu diesem Zeitpunkt wird



Abbildung 6.21: Ursprungsszenario der kritischen Situation

**Tabelle 6.8:** Kritikalitätsmesswerte zu Beginn des Szenarios

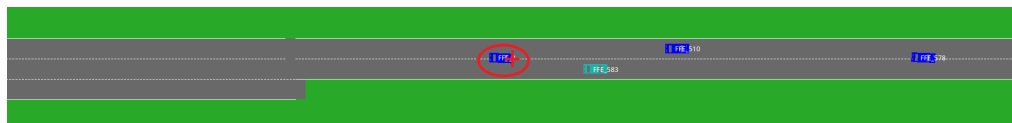
Kritikalitätsgröße	Messwert	Kritikalität
Minimale $T_{MTTC}$	31,70	0
Minimale $T_{LB}$	29,64	0
Maximales $V_{phys}$	0	0
Maximales $V_{komf}$	0	0

**Tabelle 6.9:** Kritikalitätsmesswerte bei einer Simulationszeit von  $t = 4,1$  s

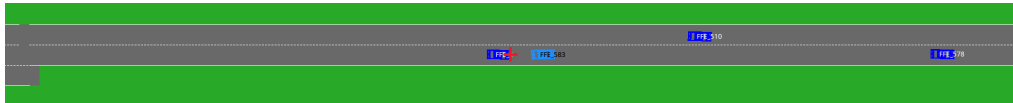
Kritikalitätsgröße	Messwert	Kritikalität
Minimale $T_{MTTC}$	1,89	7,22
Minimale $T_{LB}$	-0,32	10
Maximales $V_{phys}$	0,16	0
Maximales $V_{komf}$	0,45	0

von der Kritikalitätserkennung keine kritische Situation erkannt, alle Werte (vgl. Tabelle 6.8) befinden sich unterhalb der Schwellwerte. Im Bild wird das Ego-Fahrzeug durch ein Kreuz in der Fahrzeugspitze markiert. Ebenfalls fällt auf, dass die Fahrzeuge auf der linken Spur blau dargestellt sind, jene auf der mittleren türkis. Die Geschwindigkeit der Fahrzeuge wird in PELOPS durch ihre Farbe angegeben, von rot für stillstandsnahe Fahrzeuge bis blau für Fahrzeuge mit hoher Geschwindigkeit.

Nach einer Simulationszeit von  $t = 3,6$  s ist in Abbildung 6.22 zu sehen, wie das Testfahrzeug einen Spurwechsel von der linken auf die rechte Spur durchführt. Dies ist der Ausgangspunkt für die kritische Situation. Bei einer Simulationszeit von  $t = 4,1$  s erkennt die Kritikalitätser-

**Abbildung 6.22:** Fehlerhafte HAF-Funktion führt Spurwechsel auf rechte Spur durch

kennung erstmals eine kritische Situation, die Kritikalitätswerte von  $T_{MTTC}$  und  $T_{LB}$  steigen über die Schwellwerte. Da weder vom Testfahrzeug noch vom Fahrer des Vorderfahrzeugs zu diesem Zeitpunkt eine deutliche Reaktion eingeleitet wird, sind die Fahrdynamik- und Komfortwerte noch niedrig. Die entsprechenden Werte sind in Tabelle 6.9 dargestellt. Die vorliegende Situation ist in Abbildung 6.23 dargestellt. Das Testfahrzeug weist zu diesem Zeitpunkt noch eine deutlich höhere Geschwindigkeit als das Vorderfahrzeug auf, sodass ohne Reaktion eine Kollision unvermeidlich ist. Der negative Wert von  $T_{LB}$  bedeutet, dass das Testfahrzeug eine Kollision ohne Mithilfe des Vorderfahrzeugs nicht mehr vermeiden kann, da der letztmögliche Zeitpunkt zum Einleiten einer Bremsung bereits in der Vergangenheit liegt.



**Abbildung 6.23:** Das Testfahrzeug befindet sich sehr nah am Vorderfahrzeug und hat eine deutlich höhere Geschwindigkeit

In Abbildung 6.24 ist dargestellt, wie die kritische Situation aufgelöst wird: das Testfahrzeug bremst stark, das Vorderfahrzeug beschleunigt und wechselt auf die Nachbarspur. Dabei werden aufgrund der starken Bremsung die Schwellwerte von  $V_{\text{phys}}$  und insbesondere  $V_{\text{komf}}$  deutlich überschritten (siehe Tabelle 6.10).



**Abbildung 6.24:** Die kritische Situation wird durch eine Vollbremsung des Testfahrzeugs und eine Beschleunigung und Spurwechsel des Vorderfahrzeugs entschärft

**Tabelle 6.10:** Kritikalitätsmesswerte bei Vollbremsung des Testfahrzeugs bei einer Simulationszeit von  $t = 6,1$  s.

Kritikalitätsgröße	Messwert	Kritikalität
Minimale $T_{\text{MTTC}}$	26,37	0
Minimale $T_{\text{LB}}$	23,57	0
Maximales $V_{\text{phys}}$	0,86	10
Maximales $V_{\text{komf}}$	2,39	10

Das vorgestellte Beispiel verdeutlicht, dass die gewählten Kriterien zur Kritikalitätsbestimmung in diesem Beispiel geeignet sind, um kritische Situationen automatisiert zu detektieren. Dabei tragen die verschiedenen Kriterien zu Erkennung der Kritikalität über den gesamten kritischen Zeitraum bei. Zunächst übersteigt die Kritikalität bei den relativ zu anderen Verkehrsteilnehmern berechneten Kriterien die Schwellwerte. Nachdem die HAF-Funktion auf die kritische Situation durch starkes Bremsen reagiert, übersteigen die verkehrsunabhängigen Kriterien ihre Grenzwerte.

Die Kritikalitätserkennung ermöglicht es, fehlerhafte Fahrentscheidungen der HAF-Funktion in der Simulation zu erkennen und in der statistischen Auswertung der Sicherheit der Funktion zu berücksichtigen. Außerdem kann das gespeicherte Realszenario für Fehlerbehebungen und Funktionsverbesserungen genutzt werden.

Die Funktionsfähigkeit aller Bestandteile des passiven HAF konnte somit gezeigt werden. Aus realen Szenarien werden Simulationsszenarien generiert. In diesen Szenarien werden den Verkehrsteilnehmern, die das HAF-Fahrzeug umgeben, vorab trainierte Fahrermodelle zugeordnet. Auf Grundlage von generiertem Szenario und modellierten Verkehrsteilnehmern wird die Simulation gestartet und der Ablauf auf Kritikalität überprüft. Damit ist eine statistische Absicherung des hochautomatisierten Fahrens gemäß des Konzepts passives HAF möglich.

## 7 Zusammenfassung und Fazit

Für den Serieneinsatz von hochautomatisierten Fahrfunktionen (HAF) ist der Nachweis unerlässlich, dass sie nicht mehr Unfälle verursachen als menschliche Fahrer. Etablierte Methoden oder analytische Ansätze der Verifikation und Validierung stoßen jedoch bei der Absicherung des hochautomatisierten Fahrens an ihre Grenzen. Sie sind nicht oder noch nicht in der Lage, einen Sicherheitsnachweis für hochautomatisiertes Fahren zu erbringen.

Eine sicherheitsgerichtete Entwicklung nach etablierten Methoden wie dem V-Modell scheitert dabei an der zu hohen Anzahl an möglichen Szenarien, die beim hochautomatisierten Fahren auftreten können und abgesichert werden müssen. Für eine simulative Absicherung fehlen Wahrscheinlichkeitsverteilungen für das reale Verkehrsgeschehen, die zur Generierung repräsentativer Simulationsszenarien genutzt werden können. Analytische Ansätze zum Nachweis der Sicherheit des automatisierten Fahrens erfordern die Spezifikation von Annahmen über das Verhalten menschlicher Verkehrsteilnehmer. Der Nachweis der Vollständigkeit der Spezifikation und ihre Validierung stehen jedoch noch aus.

Stattdessen kann zum Nachweis der Sicherheit im Sinne einer positiven Risikobilanz ein Dauerlauf-Test durchgeführt werden, mit dem die Sicherheit der HAF-Funktion statistisch nachgewiesen werden kann. Für diesen muss ein Absicherungsaufwand in der Größenordnung von mehreren hundert Millionen Testkilometer erbracht werden, um die Grundlage für eine schrittweise Einführung des HAF oder eine Datenbasis für die simulative Absicherung zu schaffen. Ein Sicherheitsnachweis allein auf Grundlage von Testfahrten ist aufgrund der hierfür erforderlichen Anzahl an Testkilometern in der Größenordnung von mehreren Milliarden Kilometern noch weit aufwändiger und innerhalb eines realistischen ökonomischen und zeitlichen Rahmens nicht umsetzbar.

Die effiziente Leistbarkeit des notwendigen Absicherungsaufwands wird vor dem Hintergrund regelmäßiger Funktionsverbesserungen und -erweiterungen noch dringlicher, da jede Änderung potentiell neue Fehlerquellen schaffen kann. Daher kann ein statistischer Sicherheitsnachweis für eine bereits abgesicherte Funktion je nach Umfang der Änderung nicht oder nicht vollständig übernommen werden, sofern die Querschnittsfreiheit der Änderung auf andere, sicherheitsrelevante Funktionselemente nicht nachgewiesen werden kann. Mit zunehmender Komplexität der Komponenten einer Funktion und bei einer hohen Anzahl an Einflussgrößen, wie dies bei der HAF-Funktion der Fall ist, kann es mit hohem Aufwand verbunden oder unmöglich sein, einen solchen Nachweis zu erbringen. Gelingt es nicht, muss die Absicherung teilweise oder vollständig wiederholt werden.

Dies motiviert den vorgestellten Ansatz passives HAF, welcher für das Einfahren der notwendigen Testkilometer eine praktikable Lösung aufzeigt. Der Ansatz weist einen Weg aus dem Dilemma, dass eine Funktion, die wahrscheinlich zu einer Reduktion an Verkehrsopferten führt, nicht eingeführt wird, da ihre Sicherheit nicht ohne ihren Einsatz im realen Verkehr nachgewiesen werden kann.

Dabei wird die Grundidee verfolgt, die abzusichernde HAF-Funktion in Serienfahrzeugen zu

verbauen, aber nicht aktiv in die Fahrregelung eingreifen zu lassen. Stattdessen werden die von der HAF-Sensorik erfassten realen Verkehrsszenarien als Szenariengenerator für eine simulative Absicherung der HAF-Funktion verwendet. In jedem simulierten Szenario werden den das HAF-Fahrzeug umgebenden Verkehrsteilnehmern individuelle, maschinell trainierte Fahrermodelle zugeordnet. Existiert kein geeignetes Fahrermodell für einen in realen Szenarien beobachteten Fahrertypen, wird ein neues Fahrermodell anhand der aufgenommenen Sensordaten trainiert und die Fahrermodell-Datenbank erweitert. Auf diese Weise wird die Güte der Fahrermodellierung und damit der Simulation im Laufe des Einsatzes des passiven HAF weiter erhöht.

Die präsentierten Ergebnisse zeigen, dass alle erforderlichen Bausteine des passiven HAF mit hoher Validität für die getesteten Szenarien umsetzbar sind. Für die Modellierung von Fahrertypen mit Hilfe maschinell gelernter Modelle wurden Bayes'sche Netze, Input-Output Hidden Markov Modelle und tiefe neuronale Netze untersucht. Tiefe neuronale Netze erzielen die besten Ergebnisse bei der Modellierung von Fahrern und kommen daher in der Simulation zum Einsatz. Die trainierten Spurwechselmodelle werden in die Verkehrssimulationsumgebung implementiert und den umgebenden Fahrzeugen zugeordnet. Die Rekonstruktion von Simulationsszenarien auf Grundlage von Daten aus der Perspektive eines Fahrzeugs gelingt mit Einschränkungen, die auf Mängel der Eingangsdaten zurückzuführen sind. Durch die automatisierte Kritikalitätserkennung werden kritische Situationen erfolgreich detektiert und das zugehörige Simulationsszenario für die spätere Analyse gesichert.

Die genannten Einschränkungen resultierten aus der Qualität der in dieser Arbeit für das Training der Fahrermodelle und die Szenarioerstellung zur Verfügung stehenden Realdaten. Trotz akausaler Filterung bleibt der Umfang der in den trainierten Modellen und den Szenarien berücksichtigten Informationen auf die eigene Spur des HAF-Fahrzeugs sowie die direkten Nachbarspuren beschränkt. Hierdurch sind nicht alle für die korrekte Simulation notwendigen Daten verfügbar, sodass es zu Abweichungen in den simulierten Szenarien gegenüber den Originalszenarien kommt. Mit zunehmender Qualität der Sensordaten und des daraus generierten Umfeldmodells können die Beschränkungen der genutzten Umfelddaten reduziert oder aufgehoben werden. Anhand einer Simulation auf Grundlage von Simulationsdaten konnte gezeigt werden, dass mit fehlerfreien Sensordaten korrekte Simulationsszenarien erzeugt werden, ohne dass eine Anpassung der vorgestellten Algorithmen erforderlich wäre.

Die vorgestellte Methode beschränkt die Modellierung lateraler Fahrhandlungen auf Spurwechselmanöver. Abweichungen im Verhalten von Fahrer und HAF-Funktion innerhalb der Spur oder hinsichtlich ihrer genauen Spurwechseltrajektorie bleiben daher unberücksichtigt. Für die Absicherung des hochautomatisierten Fahrens wird diese Einschränkung als akzeptabel erachtet. Grund hierfür ist, dass HAF-Funktionen nur auf mehrspurigen Straßen mit baulicher Trennung eingesetzt werden. In diesen Szenarien fahren einzelne Fahrzeuge abgesehen von Spurwechsel-Situationen auf ihrer jeweiligen Höhe alleine auf ihrer Spur, es ist also kein weiteres Fahrzeug auf gleicher Höhe in der gleichen Spur. Daher kann aufgrund eines lateralen Manövers innerhalb der Spur keine kritische Situation resultieren.

Mit Hilfe des passiven HAF können die erforderlichen Datenmengen für den statistischen Sicherheitsnachweis des hochautomatisierten Fahrens gewonnen werden. Ziel des Ansatzes ist es, die Inverkehrbringung des HAF zu ermöglichen und so zu einer Reduktion der Unfälle im Straßenverkehr beizutragen. Dies gelingt durch die vorgestellte Lösung, ohne das durch



Betrieb eines nicht abgesicherten und damit möglicherweise unsicheren Systems eine neue Gefahrenquelle eröffnet würde und so das Gegenteil des eigentlichen Ziels erreicht werden könnte.

Die vorgestellte Arbeit stellt somit einen zielführenden und technisch umsetzbaren Weg zur Absicherung des automatisierten Fahrens vor.



# Literaturverzeichnis

## Öffentlich zugängliche Quellen

- [AAB<sup>+</sup>16] ABADI, M.; AGARWAL, A.; BARHAM, P.; BREVDO, E.; CHEN, Z.; CITRO, C.; CORRADO, G.; DAVIS, A.; DEAN, J. ; DEVIN, M.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. In: CoRR abs/1603.04467, 2016. <http://arxiv.org/abs/1603.04467>
- [ACP<sup>+</sup>18] ARIK, S.; CHEN, J.; PENG, K.; PING, W. ; ZHOU, Y.: Neural voice cloning with a few samples. In: *Advances in Neural Information Processing Systems*, 2018
- [AD14] ALTHOFF, M.; DOLAN, J. M.: Online Verification of Automated Road Vehicles Using Reachability Analysis. In: *IEEE Transactions on Robotics* 30, 2014, Nr. 4
- [ADA15] ADAC E.V. RESSORT VERKEHR ; ADAC E.V. (Hrsg.): *Zahlen, Fakten, Wissen. Aktuelles aus dem Verkehr*. München, 2015
- [Adi04] ADIPRASITO, B.: *Schriftenreihe des Instituts für Fahrzeugtechnik TU Braunschweig*. Bd. 6: *Fahrzeuglängsführung im Niedergeschwindigkeitsbereich: Zugl.: Braunschweig, Techn. Univ., Diss., 2003*. Aachen : Shaker, 2004
- [BF96] BENGIO, Y.; FRASCONI, P.: Input-output HMMs for sequence processing. In: *IEEE Transactions on Neural Networks* 7, 1996, Nr. 5
- [BK08] BALLHAUSEN, H.; KLEINELÜMERN, U.: *Die wichtigsten Erfindungen der Menschheit: Geniale Ideen, die die Welt veränderten*. Gütersloh : Chronik-Verl., 2008
- [BSR16] BONNEFON, J.; SHARIFF, A. ; RAHWAN, I.: The social dilemma of autonomous vehicles. In: *Science* 352, 2016, Nr. 6293
- [Bun17a] BUNDESMINISTERIUM FÜR VERKEHR UND DIGITALE INFRASTRUKTUR: *Verkehr in Zahlen 2016-2017*. Version: 2017. [http://www.bmvi.de/SharedDocs/DE/Anlage/VerkehrUndMobilitaet/verkehr-in-zahlen-pdf-2016-2017.pdf?\\_\\_blob=publicationFile](http://www.bmvi.de/SharedDocs/DE/Anlage/VerkehrUndMobilitaet/verkehr-in-zahlen-pdf-2016-2017.pdf?__blob=publicationFile), Abruf: 13.08.2017
- [Cal19] CALIFORNIA DEPARTMENT OF MOTOR VEHICLES: *Testing of Autonomous Vehicles*. Version: 28.01.2019. <https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/testing>, Abruf: 22.04.2019
- [CH08] CHRISTEN, F.; HUANG, Q.: The Driver Model of the traffic flow simulation PELOPS-modelling and application possibilities. In: *2nd Berlin Expert Conference on Driver Modelling "Driver Modelling in Science and Economy"*, 2008

- [Coh15] COHEN, A.: *5 reasons for short mobile release cycles*. Version: 17.06.2015. <https://techbeacon.com/app-dev-testing/5-reasons-short-mobile-release-cycles>, Abruf: 20.08.2019
- [Dai09] DAIMLER AG: *Patent Nr. DBP 854 157, Lebensretter für Tausende*. <https://media.daimler.com/marsMediaSite/de/instance/ko/Patent-Nr-DBP-854-157-Lebensretter-fuer-Tausende.xhtml?oid=9913733>. Version: 23.01.2009, Abruf: 20.04.2019
- [Dav95] DAVID, W.: *Modulares Simulationsprogramm zur Auslegung und längsdynamischen Untersuchung von Kfz-Antriebssystemen*. Forschungsgesellschaft Kraftfahrwesen mbH Aachen, 1995
- [DEI11] DOGAN, U.; EDELBRUNNER, J.; IOSSIFIDIS, I.: Autonomous driving: A comparison of machine learning techniques by means of the prediction of lane change behavior. In: *2011 IEEE International Conference on Robotics and Biomimetics*, IEEE, 2011, S. 1837–1843
- [Dem19] DEMAITRE, E.: *Driverless investment tops \$1.6B so far this month*. Version: 15.02.2019. <https://www.therobotreport.com/driverless-investment-high-valentines/>, Abruf: 20.08.2019
- [Des18] DESTATIS: *Verkehrsunfälle - Zeitreihen*. Version: 12.06.2018. [https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Verkehrsunfaelle/Publikationen/Downloads-Verkehrsunfaelle/verkehrsunfaelle-zeitreihen-pdf-5462403.pdf?\\_\\_blob=publicationFile&v=4](https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Verkehrsunfaelle/Publikationen/Downloads-Verkehrsunfaelle/verkehrsunfaelle-zeitreihen-pdf-5462403.pdf?__blob=publicationFile&v=4), Abruf: 12.06.2018
- [Deu12] DEUTSCHER VERKEHRSSICHERHEITSRAT (Hrsg.): *Schriftenreihe Verkehrssicherheit*. Bd. 16: *Vision Zero: Grundlagen & Strategien*. 2012 [http://www.dvr.de/download2/p3042/3042\\_0.pdf](http://www.dvr.de/download2/p3042/3042_0.pdf)
- [Deu17] DEUTSCHER BUNDESTAG: *Achtes Gesetz zur Änderung des Straßenverkehrsgesetzes*. [https://dejure.org/BGB1/2017/BGB1.\\_I\\_S.\\_1648](https://dejure.org/BGB1/2017/BGB1._I_S._1648). Version: 20.06.2017, Abruf: 02.02.2020
- [Die16] DIETMAYER, K.: Predicting of Machine Perception for Automated Driving. In: MAURER, Markus (Hrsg.); GERDES, J. C. (Hrsg.); LENZ, Barbara (Hrsg.); WINNER, Hermann (Hrsg.): *Autonomous Driving*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2016
- [DK15] DIETSCHKE, K.; KUHLGATZ, D.: History of the automobile. In: REIF, Konrad (Hrsg.): *Gasoline Engine Management*. Wiesbaden : Springer Fachmedien Wiesbaden, 2015
- [DLR77] DEMPSTER, A. P.; LAIRD, N. M. ; RUBIN, D. B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. In: *Journal of the Royal Statistical Society* 39, 1977, Nr. 1
- [DLZ08] DENG, W.; LEE, Y. H. ; ZHAO, A.: Hardware-in-the-loop simulation for autonomous driving. In: *34th Annual Conference of IEEE Industrial Electronics, 2008*, IEEE, 2008

- [Don09] DONGES, E.: Fahrerhaltensmodelle. In: WINNER, Hermann (Hrsg.); HAKULLI, Stephan (Hrsg.) ; WOLF, Gabriele (Hrsg.): *Handbuch Fahrerassistenzsysteme*. Wiesbaden : Vieweg+Teubner Verlag / GWV Fachverlage GmbH Wiesbaden, 2009 (PRAXIS | ATZ/MTZ-Fachbuch)
- [Ehm02] EHMANN, D.: *Schriftenreihe Automobiltechnik*. Bd. 63: *Modellierung des taktischen Fahrerhaltens bei Spurwechselvorgängen: Zugl.: Aachen, Techn. Hochsch., Diss., 2002*. Aachen : Forschungsges. Kraftfahrwesen (fka), 2002
- [EK17] ETHIK-KOMMISSION ; DI FABIO, Udo (Hrsg.): *Automatisiertes und vernetztes Fahren*. Version: 2017. [http://www.bmvi.de/SharedDocs/DE/Anlage/Presse/084-dobrindt-bericht-der-ethik-kommission.pdf?\\_\\_blob=publicationFile](http://www.bmvi.de/SharedDocs/DE/Anlage/Presse/084-dobrindt-bericht-der-ethik-kommission.pdf?__blob=publicationFile), Abruf: 22.06.2017
- [EWZS13] EICHHORN, A. von; WERLING, M.; ZAHN, P. ; SCHRAMM, D.: Maneuver prediction at intersections using cost-to-go gradients. In: *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, IEEE, 2013, S. 112–117
- [Fra16] FRAHM, C.: *Entwicklungszyklen schrumpfen: Wie die Autobauer Google und Apple nachjagen*. Version: 07.11.2016. <https://www.automobilwoche.de/article/20161107/HEFTARCHIV/161109952/entwicklungszyklen-schrumpfen-wie-die-autobauer-google-und-apple-nachjagen>, Abruf: 20.08.2019
- [GBC16] GOODFELLOW, I.; BENGIO, Y. ; COURVILLE, A.: *Deep Learning*. MIT Press, 2016
- [GBD15] GINDELE, T.; BRECHTEL, S. ; DILLMANN, R.: Learning Driver Behavior Models from Traffic Observations for Decision Making and Planning. In: *IEEE Intelligent Transportation Systems Magazine* 7, 2015, Nr. 1
- [GBH12] GLAUNER, P.; BLUMENSTOCK, A. ; HAUEIS, M.: Effiziente Felderprobung von Fahrerassistenzsystemen. In: *UNI DAS eV (ed.), 8. Workshop Fahrerassistenzsysteme* 8, 2012
- [Gip81] GIPPS, P.: A behavioural car-following model for computer simulation. In: *Transportation Research Part B: Methodological* 15, 1981, Nr. 2, S. 105–111
- [Gip86] GIPPS, P.: A model for the structure of lane-changing decisions. In: *Transportation Research Part B: Methodological* 20, 1986, Nr. 5, S. 403–414
- [Haz02] HAZEWINKEL, M.: *Encyclopaedia of mathematics*. Berlin and New York : Springer-Verlag, 2002
- [HBE18] HENRIKSSON, Jens; BORG, Markus ; ENGLUND, Cristofer: Automotive safety and machine learning. In: STOLLE, Reinhard (Hrsg.): *Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems*. New York, NY : ACM, 2018 (ACM Conferences), S. 47–49
- [HDBJ14] HAGAN, M.; DEMUTH, H.; BEALE, M. ; JESÚS, O. de: *Neural network design*. 2nd ed. 2014

- [HKW<sup>+</sup>17] HELMER, T.; KOMPASS, K.; WANG, L.; KÜHBECK, T. ; KATES, R.: Safety Performance Assessment of Assisted and Automated Driving in Traffic: Simulation as Knowledge Synthesis. In: WATZENIG, Daniel (Hrsg.); HORN, Martin (Hrsg.): *Automated Driving: Safer and More Efficient Future Driving*. Cham : Springer International Publishing, 2017
- [HN98] HOCHSTÄDTER, A.; NEUNZIG, D.: Simulating macroscopic traffic phenomena with the microscopic simulation program Pelops. In: *Proceedings of the 1998 IEEE International Conference on Intelligent Vehicles*. Piscataway, NJ : IEEE Service Center, 1998
- [HNW97] HOCHSTAEDTER, A.; NEUNZIG, D. ; WEILKES, M.: Analyse und Entwicklung von Fahrerwarn- und informationssystemen mit dem Programmsystem PELOPS. In: *Vortrag beim Haus der Technik, Seminar "Fahrerinformationssysteme", Essen, 1997*
- [Hom05] HOMANN, K.: Wirtschaft und gesellschaftliche Akzeptanz: Fahrerassistenzsysteme auf dem Prüfstand. In: *Fahrerassistenzsysteme mit maschineller Wahrnehmung*. Springer, 2005
- [Hom18] HOMMEN, M.: *Autonomes Fahren: Die ethischen Fragen erweisen sich als möglicherweise unlösbar*. Version: 22.08.2018. <https://www.zeit.de/mobilitaet/2018-08/autonomes-fahren-robotik-kuenstliche-intelligenz-auto-mobilitaet/komplettansicht>, Abruf: 20.08.2019
- [HPH19] HAUER, F.; PRETSCHNER, A. ; HOLZMÜLLER, B.: Fitness Functions for Testing Automated and Autonomous Driving Systems. In: ROMANOVSKY, Alexander (Hrsg.); TROUBITSYNA, Elena (Hrsg.) ; BITSCH, Friedemann (Hrsg.): *Computer Safety, Reliability, and Security* Bd. 11698. Cham : Springer International Publishing, 2019
- [HWL17] HENZEL, M.; WINNER, H. ; LATTKE, B.: Herausforderungen in der Absicherung von Fahrerassistenzsystemen bei der Benutzung maschinell gelernter und lernender Algorithmen. In: BENGLER, K. (Hrsg.): *11. Workshop Fahrerassistenzsysteme und automatisiertes Fahren, 2017*
- [Ins14] INSTITUT FÜR KRAFTFAHRWESEN: *Pelops Whitepaper*. Version: 2014. [http://www.fka.de/pdf/pelops\\_whitepaper.pdf](http://www.fka.de/pdf/pelops_whitepaper.pdf), Abruf: 09.01.2015
- [Int11] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *ISO 26262: International Standard Road vehicles - Functional safety*. 2011
- [IPG20a] IPG AUTOMOTIVE GMBH: *CarMaker*. <https://ipg-automotive.com/de/produkte-services/simulation-software/carmaker/#test-manager>. Version: 2020, Abruf: 10.06.2020
- [IPG20b] IPG AUTOMOTIVE GMBH: *CarMaker Programmers Guide Version 9.0*. 2020
- [IPG20c] IPG AUTOMOTIVE GMBH: *CarMaker User Guide Version 9.0*. 2020
- [Ise06] ISERMANN, R.: *Fahrdynamik-Regelung: Modellbildung, Fahrerassistenzsysteme, Mechatronik ; mit 28 Tabellen*. 1. Aufl. Wiesbaden : Friedr. Vieweg & Sohn Verlag | GWV Fachverlage GmbH Wiesbaden, 2006

- [KB18a] KRAFTFAHRT-BUNDESAMT ; KRAFTFAHRT-BUNDESAMT (Hrsg.): *Kraftfahrt-Bundesamt - Motorisierung*. Version: 2018. [https://www.kba.de/DE/Statistik/Fahrzeuge/Neuzulassungen/Motorisierung/motorisierung\\_node.html](https://www.kba.de/DE/Statistik/Fahrzeuge/Neuzulassungen/Motorisierung/motorisierung_node.html), Abruf: 20.04.2019
- [KB18b] KRAFTFAHRT-BUNDESAMT ; KRAFTFAHRT-BUNDESAMT (Hrsg.): *Neuzulassungen: Jahresbilanz der Neuzulassungen 2017*. Version: 2018. [https://www.kba.de/DE/Statistik/Fahrzeuge/Neuzulassungen/n\\_jahresbilanz.html?nn=644522](https://www.kba.de/DE/Statistik/Fahrzeuge/Neuzulassungen/n_jahresbilanz.html?nn=644522), Abruf: 21.07.2018
- [KF09] KOLLER, D.; FRIEDMAN, N.: *Probabilistic graphical models: Principles and techniques*. Cambridge, Mass. : MIT Press, 2009
- [Koc04] KOCHERSCHIEDT, H.: Wege zur Effizienzbetrachtung von Fahrerassistenzsystemen. In: *1. Tagung Aktive Sicherheit durch Fahrerassistenzsysteme*, 2004
- [Köp18] KÖPPE, J.: *Ethik und autonomes Fahren: Einer muss sterben - nur wer?* Version: 24.10.2018. <https://www.spiegel.de/wissenschaft/technik/unfaelle-mit-selbstfahrenden-autos-wer-soll-leben-wer-soll-sterben-a-1234901.html>, Abruf: 20.08.2019
- [KPLL13] KUMAR, P.; PERROLLAZ, M.; LEFÈVRE, S. ; LAUGIER, C.: Learning-based approach for online lane change intention prediction. In: *IEEE Intelligent Vehicles Symposium, Proceedings* , 2013, S. 797–802
- [KS89] KLUGE, F.; SEEBOLD, E.: *Etymologisches Wörterbuch der deutschen Sprache*. Berlin : de Gruyter, 1989
- [KSH12] KRIZHEVSKY, A.; SUTSKEVER, I. ; HINTON, G.: ImageNet Classification with Deep Convolutional Neural Networks. In: *Advances in Neural Information Processing Systems* , 2012
- [KTH07] KESTING, A.; TREIBER, M. ; HELBING, D.: General Lane-Changing Model MOBIL for Car-Following Models. In: *Transportation Research Record: Journal of the Transportation Research Board* 1999, 2007, Nr. 1, S. 86–94
- [KWD<sup>+</sup>12] KASPER, D.; WEIDL, G.; DANG, T.; BREUEL, G.; TAMKE, A.; WEDEL, A. ; ROSENSTIEL, W.: Object-Oriented Bayesian Networks for Detection of Lane Change Maneuvers. In: *Intelligent Vehicles Symposium 4*, 2012, Nr. 3, S. 19–31
- [Ler02] LERNER, U.: *Hybrid Bayesian networks for reasoning about complex systems*, Stanford University, Stanford, CA, Diss., 2002
- [LFSR15] LÜKE, S.; FOCHLER, O.; SCHALLER, T. ; REGENSBURGER, U.: Stauassistentz und -automation. In: WINNER, H. (Hrsg.); HAKULI, S. (Hrsg.); LOTZ, F. (Hrsg.) ; SINGER, C. (Hrsg.): *Handbuch Fahrerassistenzsysteme: Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*. Wiesbaden : Springer Fachmedien Wiesbaden, 2015

- [LOW04] LEE, Suzanne E.; OLSEN, ERIK C. B., 1968- ; WIERWILLE, Walter W.: *A comprehensive examination of naturalistic lane-changes*. 2004 <https://rosap.ntl.bts.gov/view/dot/4191>
- [LRKS13] LIEBNER, M.; RUHAMMER, C.; KLANNER, F. ; STILLER, C.: Generic driver intent inference based on parametric models. In: *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 2013, S. 268–275
- [LSK01] LERNER, U.; SEGAL, E. ; KOLLER, D.: Exact inference in networks with discrete children of continuous parents. In: *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*. Seattle, Washington : Morgan Kaufmann Publishers Inc, 2001
- [Lud98] LUDMANN, J.: *Beeinflussung des Verkehrsablaufes auf Straßen - Analyse mit dem fahrzeugorientierten Verkehrsflusssimulationsprogramm PELOPS*, RWTH Aachen, Diss., 1998
- [MKS17] MERITY, S.; KESKAR, N. S. ; SOCHER, R.: *Regularizing and Optimizing LSTM Language Models*. Version: 2017. <http://arxiv.org/pdf/1708.02182v1>
- [MP43] McCULLOCH, W.; PITTS, W.: A logical calculus of the ideas immanent in nervous activity. In: *The bulletin of mathematical biophysics* 5, 1943, Nr. 4
- [Mur03] MURPHY, K. P.: *Fitting a conditional linear gaussian distribution*. Version: 29.01.2003. <https://www.cs.ubc.ca/~murphyk/Papers/learnCG.pdf>, Abruf: 25.02.2020
- [Mur12] MURPHY, Kevin P.: *Machine learning: A probabilistic perspective*. Cambridge, Mass. : MIT Press, 2012
- [Nat08] NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION: *National Motor Vehicle Crash Causation Survey: DOT HS 811 059*. Version: 07.2008. <https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/811059>, Abruf: 19.04.2015
- [PL16] PFEFFER, R.; LEICHSENRING, T.: Continuous Development of Highly Automated Driving Functions with Vehicle-in-the-Loop Using the Example of Euro NCAP Scenarios. In: GÜHMANN, Clemens (Hrsg.); RIESE, Jens (Hrsg.) ; RÜDEN, Klaus von (Hrsg.): *Simulation and Testing for Vehicle Technology*. Cham : Springer International Publishing, 2016
- [PSS19] PFEFFER, R.; SAX, E. ; SCHMIDT, S.: Realdatenbasierte simulationsgestützte Absicherung hochautomatisierter Fahrfunktionen. In: *ATZelektronik* , 2019, Nr. 11
- [Ras83] RASMUSSEN, J.: Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-13, 1983, Nr. 3
- [RHW<sup>+</sup>12] RODEMERK, Claas; HABENICHT, Stefan; WEITZEL, Alexander; WINNER, Hermann ; SCHMITT, Thomas: Development of a general criticality criterion for the risk estimation of driving situations and its application to a maneuver-based lane



- change assistance system. In: *Intelligent Vehicles Symposium (IV), 2012 IEEE*, 2012, S. 264–269
- [RJ86] RABINER, L.; JUANG, B.: An introduction to hidden Markov models. In: *IEEE ASSP Magazine* 3, 1986, Nr. 1
- [SAE14] SAE ON-ROAD AUTOMATED VEHICLE STANDARD COMMITTEE: *J3016: Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems*. [http://standards.sae.org/j3016\\_201401/](http://standards.sae.org/j3016_201401/). Version: 16.01.2014, Abruf: 25.03.2015
- [SHM<sup>+</sup>16] SILVER, D.; HUANG, A.; MADDISON, C.; GUEZ, A.; SIFRE, L.; VAN DEN DRIESSCHE, G.; SCHRITTWIESER, J.; ANTONOGLU, I.; PANNEERSHELVAM, V.; LANCTOT, M.; DIELEMAN, S.; GREWE, D.; NHAM, J.; KALCHBRENNER, N.; SUTSKEVER, I.; LILICRAP, T.; LEACH, M.; KAVUKCUOGLU, K.; GRAEPEL, T.; HASSABIS, D.: Mastering the game of Go with deep neural networks and tree search. In: *Nature* 529, 2016, Nr. 7587
- [Sil18] SILVER, D.; FORBES (Hrsg.): *Waymo Has The Most Autonomous Miles, By A Lot*. 26.07.2018
- [SM19] STADT MÜNCHEN, Stadtkämmerei: *Der Stadthaushalt*. Version: 2019. [https://www.muenchen.de/rathaus/dam/jcr:dab0a522-4811-447b-87ea-de8d0081419c/lhm\\_haushalt\\_2019\\_barrierefrei.pdf](https://www.muenchen.de/rathaus/dam/jcr:dab0a522-4811-447b-87ea-de8d0081419c/lhm_haushalt_2019_barrierefrei.pdf), Abruf: 18.04.2020
- [SMF19] SABADKA, Dušan; MOLNÁR, Vieroslav; FEDORKO, Gabriel: Shortening of Life Cycle and Complexity Impact on the Automotive Industry. In: *TEM Journal* 8, 2019, Nr. 4, S. 1295
- [Smi17] SMITH, B.: Automated driving and product liability. In: *Mich. St. L. Rev.*, 2017
- [SQC18] SALAY, Rick; QUEIROZ, Rodrigo; CZARNECKI, Krzysztof: An Analysis of ISO 26262: Machine Learning and Safety in Automotive Software. In: *SAE Technical Paper Series*. Warrendale, PA, United States: SAE International, 2018 (SAE Technical Paper Series)
- [SSBZ15] STELETT, Jan E.; SCHUMACHER, Jan; BRANZ, Wolfgang; ZOLLNER, J. M.: Uncertainty propagation in criticality measures for driver assistance. In: *2015 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2015, S. 1187–1194
- [SSL<sup>+</sup>13] SCHULDT, F.; SAUST, F.; LICHT, B.; MAURER, M.; SCHOLZ, S.: Effiziente systematische Testgenerierung für Fahrerassistenzsysteme in virtuellen Umgebungen. In: *Automatisierungssysteme, Assistenzsysteme und Eingebettete Systeme Für Transportmittel*, 2013
- [SSSS17] SHALEV-SHWARTZ, S.; SHAMMAH, S.; SHASHUA, A.: On a Formal Model of Safe and Scalable Self-driving Cars. In: *CoRR* abs/1708.06374, 2017. <http://arxiv.org/abs/1708.06374>

- [SW14] SCHREIER, Matthias; WILLERT, Volker: Bayesian , Maneuver-Based , Long-Term Trajectory Prediction and Criticality Assessment for Driver Assistance Systems. In: *IEEE Intelligent Transportation Systems, International Conference on*, 2014, S. 334–341
- [SZS<sup>+</sup>15] STELLET, Jan E.; ZOFKA, Marc R.; SCHUMACHER, Jan; SCHAMM, Thomas; NIEWELS, Frank ; ZOLLNER, J. M.: Testing of Advanced Driver Assistance Towards Automated Driving: A Survey and Taxonomy on Existing Approaches and Open Questions. In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, IEEE, 2015, S. 1455–1462
- [TDB11] TAMKE, Andreas; DANG, Thao ; BREUEL, Gabi: A flexible method for criticality assessment in driver assistance systems. In: *2011 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2011, S. 697–702
- [TFP<sup>+</sup>19] TUNCALI, C. E.; FAINEKOS, G.; PROKHOROV, D.; ITO, H. ; KAPINSKI, J.: Requirements-driven Test Generation for Autonomous Vehicles with Machine Learning Components. In: *IEEE Transactions on Intelligent Vehicles* , 2019
- [TGG<sup>+</sup>97] TIJERINA, L.; GARROTT, W. R.; GLECKER, M.; STOLTZFUS, D. ; PARMER, E.: Van and passenger car driver eye glance behavior during lane change decision phase, Interim Report. In: *Transportation Research Center Report, National Highway Transportation Safety Administration, Washington DC* , 1997
- [THH00] TREIBER, Martin; HENNECKE, Ansgar ; HELBING, Dirk: Congested traffic states in empirical observations and microscopic simulations. In: *Phys. Rev. E* 62, 2000, Nr. 2, S. 1805–1824
- [TK09] TOLEDO, T.; KATZ, R.: State Dependence in Lane-Changing Models. In: *Transportation Research Record: Journal of the Transportation Research Board* 2124, 2009, S. 81–88
- [TVCG16] TEMBO, S.; VATON, S.; COURANT, J. ; GOSSELIN, S.: A tutorial on the EM algorithm for Bayesian networks: Application to self-diagnosis of GPON-FTTH networks. In: *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, IEEE, 05.09.2016 - 09.09.2016
- [VIA<sup>+</sup>14] VAIANA, R.; IUELE, T.; ASTARITA, V.; CARUSO, M. V.; TASSITANI, A.; ZAFFINO, C. ; GIOFRÈ, V.: Driving behavior and traffic safety: an acceleration-based safety evaluation procedure for smartphones. In: *Modern Applied Science* 8, 2014, Nr. 1
- [Wac17] WACHENFELD, W.: *How Stochastic can Help to Introduce Automated Driving*. Darmstadt, Technische Universität Darmstadt, Diss., 01.01.2017. <http://tuprints.ulb.tu-darmstadt.de/5949/>
- [WGS13] WINNER, H.; GEYER, S. ; SEFATI, M.: Maße für den Sicherheitsgewinn von Fahrerassistenzsystemen. In: *6. Darmstädter Kolloquium Mensch + Fahrzeug 2013*. Stuttgart : Ergonomia Verlag, 2013

- [WHW12] WINNER, H.; HAKULI, S. ; WOLF, G.: *Handbuch Fahrerassistenzsysteme: Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort: mit 550 Abbildungen und 45 Tabellen*. 2. Auflage. Springer, 2012
- [Wie74] WIEDEMANN, R.: *Schriftenreihe des Instituts für Verkehrswesen der Universität Karlsruhe*. Bd. 8: *Simulation des Straßenverkehrsflusses*. Karlsruhe : Inst. für Verkehrswesen der Universität Karlsruhe, 1974
- [Win01] WINNER, H.: Patent DE 101 02 771 A1: Einrichtung zum Bereitstellen von Signalen in einem Kraftfahrzeug. In: *Deutsches Patent-und Markenamt* , 2001
- [Wir16] WIRTSCHAFTSKOMMISSION DER VEREINTEN NATIONEN FÜR EUROPA: *Einheitliche Bedingungen für die Genehmigung der Fahrzeuge hinsichtlich der Lenkanlage: Regelung Nr. 79*. [http://data.europa.eu/eli/reg/2008/79\(2\)/oj](http://data.europa.eu/eli/reg/2008/79(2)/oj). Version: 20.01.2016
- [WMT<sup>+</sup>16] WEIDL, G.; MADSEN, A.; TERESHCHENKO, V.; ZHANG, W.; WANG, S. ; KASPER, D.: Situation Awareness and Early Recognition of Traffic Maneuvers. In: *Proceedings of The 9th EUROSIM Congress on Modelling and Simulation, EUROSIM 2016, The 57th SIMS Conference on Simulation and Modelling SIMS 2016*, 2016
- [Wor19] WORLD HEALTH ORGANIZATION: *Global Status Report on Road Safety 2018*. World Health Organization, 2019
- [WRM<sup>+</sup>19] WOOD, M.; ROBBEL, P.; MAASS, M.; TEBBENS, R. D.; MEIJS, M.; HARB, M.; REACH, J.; ROBINSON, K.; WITTMANN, D.; SRIVASTAVA, T.; ESSAYED BOUZOURAA, M.; LIU, S.; WANG, Y.; KNOBEL, C.; BOYMANNS, D.; LÖHNING, M.; DEHLINK, B.; KAULE, D.; KRÜGER, R.; FRTUNIKJ, J.; RAISCH, F.; GRUBER, M.; STECK, J.; MEIJA-HERNANDEZ, J.; SYGUDA, S.; BLÜHER, P.; KLONECKI, K.; SCHNARZ, P.; WILTSCHKO, T.; PUKALLUS, S.; SEDLACZEK, K.; GARBACIK, N.; SMERZA, D.; LI, D.; TIMMONS, A.; BELLOTTI, M.; O'BRIENS, M.; SCHÖLLHORN, M.; DANNEBAUM, U.; WEAST, J.; TATOURIAN, A.; DORNIEDEN, B.; SCHNETTER, P.; THEMANN, P.; WEIDNER, T. ; SCHLICHT, P.: *Safety first for automated driving*. Version: 2019. <https://www.daimler.com/dokumente/innovation/sonstiges/safety-first-for-automated-driving.pdf>, Abruf: 13.01.2020
- [WW13] WINNER, H.; WACHENFELD, W.: Absicherung automatischen Fahrens. In: *6. FAS-Tagung München*. München, 2013
- [WW15a] WACHENFELD, W.; WINNER, H.: Die freigabe des autonomen fahrens. In: *Autonomes Fahren*. Springer Vieweg, Berlin, Heidelberg, 2015
- [WW15b] WACHENFELD, W.; WINNER, H.: Virtual Assessment of Automation in Field Operation A New Runtime Validation Method. In: *10. Workshop Fahrerassistenzsysteme*. 2015
- [Yan12] YANG, Hong: *Simulation-based evaluation of traffic safety performance using surrogate safety measures*, Rutgers University-Graduate School-New Brunswick, Diss., 2012

- [ZAG06] ZHANG, Y.; ANTONSSON, E. ; GROTE, K.: A new threat assessment measure for collision avoidance systems. In: *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*, 2006
- [ZE13] ZLOCKI, A.; ECKSTEIN, L.: Combined methods for an effective evaluation of automated driving functions. In: *TRBs second annual workshop on road vehicle automation. Stanford*, 2013

## Eigene Veröffentlichungen und Tagungsbeiträge

- [KGHL17] KOENIG, A.; GUTBROD, M.; HOHMANN, S. ; LUDWIG, J.: Bridging the gap between open loop tests and statistical validation for highly automated driving. In: *SAE International journal of transportation safety* , 2017, Nr. 1
- [KRH17] KOENIG, A.; REHDER, T. ; HOHMANN, S.: Exact inference and learning in hybrid Bayesian Networks for lane change intention classification. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2017
- [KWLH18] KOENIG, A.; WITZLSPERGER, K.; LEUTWILER, F. ; HOHMANN, S.: Overview of HAD validation and passive HAD as a concept for validating highly automated cars. In: *at-Automatisierungstechnik Methoden und Anwendungen der Steuerungs-, Regelungs- und Informationstechnik* 66, 2018, Nr. 2
- [RKG<sup>+</sup>19] REHDER, T.; KOENIG, A.; GOEHL, M.; LOUIS, L. ; SCHRAMM, D.: Lane Change Intention Awareness for Assisted and Automated Driving on Highways. In: *IEEE Transactions on Intelligent Vehicles* 4, 2019, Nr. 2

## Betreute studentische Arbeiten

- [Gut16] GUTBROD, M.: *Learning Driver Intentions with Deep Neural Networks*. München, Technische Universität München, Masterarbeit, 2016
- [Wit17] WITZLSPERGER, K.: *Erweiterung einer Verkehrssimulationsumgebung zur passiven Absicherung von Fahrfunktionen durch Simulation und Evaluierung von Szenarien aus Realdaten*. München, Hochschule für Angewandte Wissenschaften, Bachelorarbeit, 2017