

Bewertung von Fahrerassistenzsystemen im Umfeld des szenariobasierten Testens

Zur Erlangung des akademischen Grades eines

**DOKTORS DER INGENIEURWISSENSCHAFTEN
(Dr.-Ing.)**

von der KIT-Fakultät für
Elektrotechnik und Informationstechnik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

M.Sc. Christian King

geb. in Stuttgart

Tag der mündlichen Prüfung:

Hauptreferent:

Korreferentin:

07.03.2023

Prof. Dr.-Ing. Eric Sax

Prof. Dr.-Ing. Ina Schaefer

Kurzfassung

Durch die steigende Automatisierung von Fahrerassistenzsystemen wird die Fahraufgabe zunehmend vom Fahrer auf das Fahrzeug übertragen. Das szenariobasierte Testen ist dabei ein vielversprechender Ansatz, mit dem die Sicherheit von Fahrerassistenzsystemen nachgewiesen werden kann. Bei der Verwendung des Ansatzes zur entwicklungsbegleitenden Bestimmung des Reifegrads eines Fahrerassistenzsystems ergibt sich allerdings eine Anzahl an Herausforderungen, die in der vorliegenden Dissertation herausgearbeitet werden.

Im Anschluss wird der Stand der Wissenschaft und Technik dargestellt, der für die Adressierung der Herausforderungen notwendig ist. Der Fokus liegt dabei auf dem Testen von Fahrerassistenzsystemen während des Entwicklungsprozesses. In der Diskussion des Stands der Technik und Wissenschaft wird anschließend gezeigt, dass die aktuell vorhandenen Testmethoden in der Automobilindustrie nicht ausreichend sind um Fahrerassistenzsysteme in einem szenariobasierten Testkonzept entwicklungsbegleitend zu testen. Als eine zentrale Herausforderung wird herausgearbeitet, dass Szenarien durch die Variation verschiedener Parameter erzeugt werden, wodurch sowohl der zeitliche Ablauf des erzeugten Szenarios im Vorfeld unbekannt ist als auch neue Szenarien entstehen können.

Darauf aufbauend wird in dieser Dissertation ein Testkonzept entwickelt, mit dem Fahrerassistenzsysteme in Fahrszenarien bewertet werden können. Das Konzept ermöglicht es zu überprüfen, ob Systemanforderungen durch ein entwickeltes Fahrerassistenzsystem in unterschiedlichen Fahrszenarien erfüllt werden. Zusätzlich beinhaltet das Testkonzept eine Analyse der durchgeführten Fahrszenarien und deren Inhalt. Aus diesen Ergebnissen wird dann ein szenariospezifischer

Reifegrad des Fahrerassistenzsystems bestimmt. Anschließend erfolgt eine prototypische Umsetzung in einer Software-in-the-Loop und Hardware-in-the-Loop Testumgebung. In der Software-in-the-Loop Testumgebung wird beispielsweise ein automatisiertes Notbremssystem in unterschiedlichen Fahrszenarien getestet.

Die Evaluation der prototypischen Umsetzung zeigt, dass das entwickelte Testkonzept grundsätzlich funktioniert und eine systemspezifische Bewertung von Fahrerassistenzsystemen in Fahrszenarien ermöglicht. Im Vergleich mit dem aktuellem Stand der Technik und Wissenschaft, können mit dem erarbeiteten Konzept darüber hinaus auch die Kosten für die Testerstellung und Testdurchführung reduziert werden.

Abstract

Due to the ongoing automation of driver assistance systems (DAS), the driving task is increasingly being transferred from the driver to the vehicle. In this context, scenario-based testing is a promising approach to prove the safety of DAS. However, when utilizing this approach to determine the level of maturity of a DAS during the product development, a number of challenges arises that are elaborated in this dissertation.

The state of the art necessary to address these challenges is then presented. The focus of the state of the art is on testing DAS during the development process. In the discussion of the state of the, it is then shown that the currently available test methods in the automotive industry are not sufficient to test DAS in a scenario-based test concept accompanying the product development. As a key challenge, it is elaborated that scenarios are generated by varying different parameters, which means that both the temporal sequence of the generated scenario is unknown in advance and previously unknown scenarios can emerge.

Based on this, a test concept is developed that can be used to evaluate DAS in driving scenarios. The concept enables the verification whether system requirements are fulfilled by a developed DAS in different driving scenarios. Additionally, the test concept includes an analysis of the driving scenarios executed and their content. These results are then used to determine a scenario-specific maturity level of the tested DAS. This is followed by a prototypical implementation in a software-in-the-loop and hardware-in-the-loop test environment. In the software-in-the-loop test environment, for example, an automated emergency braking system is tested in different driving scenarios.

The evaluation of the prototypical implementation shows that the developed test concept is fundamentally functional and enables a system-specific evaluation of DAS in driving scenarios. In comparison with the current state of the art, the developed concept can also reduce the costs for test generation and test execution.

Danksagung

Zunächst möchte ich meinem Doktorvater Prof. Dr.-Ing. Eric Sax für die Möglichkeit der Promotion und seine unermüdliche Unterstützung danken. Darüber hinaus danke ich Prof. Dr.-Ing. Ina Schaefer für die Übernahme des Korreferats und die Diskussion der Arbeitsinhalte.

Ein großer Dank geht an die Kollegen der HiL-Testing Abteilung von Mercedes-Benz, insbesondere Christopher Kober, Jan Lier und Christoph Wohlfahrt, für die Unterstützung meiner Arbeit sowie die vielen Einblicke und anregenden Diskussionen.

Allen Mitarbeitenden des Forschungsbereichs ESS möchte ich für das großartige Arbeitsumfeld danken, in dem die Arbeit auch in den anstrengenden Phasen immer Spaß gemacht hat und natürlich für den überragenden Doktorhut - ihr seid die Besten! Für viele wertvolle Gespräche und Ideen rund um mein Thema danke ich Christian Steinhauser, Julian Fuchs, Thilo Braun und allen weiteren Mitarbeitenden von ESM. Ein ganz besonderer Dank geht auch an Stefan Otten, Jacob Langner und Lennart Ries. Vielen Dank für all die wertvollen und erhellenden Diskussionen sowie dass ihr mit spannenden Projekten einen Rahmen geschaffen habt, in dem ich die Promotion erfolgreich abschließen konnte. Zum Schluss auch noch ein großes Danke an Ulrike Beideck, dass du mir all die Jahre den Rücken freigehalten hast.

Ganz herzlichen Dank an Philipp, Ralph und meine Eltern Ute und Rudibert, dass ihr mich jederzeit unterstützt habt und zum Reden bereit standet. Meiner Freundin Katharina danke ich aus vollem Herzen, vor allem für ihr schier endloses Verständnis und die unglaubliche Unterstützung während der Zeit.

Karlsruhe, im April 2023

Christian King

Inhaltsverzeichnis

Kurzfassung	i
Abstract	iii
Danksagung	v
1 Einleitung und Motivation	1
1.1 Motivation	1
1.2 Herausforderungen beim Testen von Fahrerassistenzsystemen	3
1.3 Forschungsfragen	5
1.4 Gliederung	6
2 Entwicklung und Testen von Fahrerassistenzsystemen in der Automobilindustrie	9
2.1 Automatisierung der Fahrerassistenzsysteme in der Automobilindustrie	9
2.1.1 Herausforderungen des automatisierten Fahrens	11
2.2 Produktentwicklungsprozess und Reifegradabsicherung	14
2.2.1 Produktentwicklung nach dem V-Modell	18
2.3 Entwicklungsbegleitendes Testen von Fahrerassistenzsystemen	21
2.3.1 Funktionale Sicherheit und Betriebssicherheit von hoch automatisierten Fahrzeugen	23
2.3.2 Anforderungsbasiertes Testen von Kraftfahrzeugen	26
2.3.2.1 Testfalldefinition und Testabdeckung	27
2.3.2.2 Integrationsstufen beim Testen von Fahrerassistenzsystemen	30
2.3.2.3 Testplattformen in der Automobilindustrie	33

2.3.3	Szenariobasiertes Testen in der Automobilindustrie	36
2.3.3.1	Definition eines Szenarios	38
2.3.3.2	Beschreibung eines Szenarios	39
2.3.3.3	Testprozess für eine szenariobasierte Freigabe von ADS	43
2.3.3.4	Bewertungskriterien für Fahrscenarien	45
2.4	Zulassung von Kraftfahrzeugen in Deutschland	47
2.4.1	Typprüfung von Fahrzeugen	49
2.4.2	Nutzung von Simulationen für die Zulassung von Fahrzeugen	51
3	Diskussion des Stands der Wissenschaft und Anforderungen an ein szenariobasiertes Testkonzept	53
3.1	Verwandte Arbeiten	53
3.1.1	Bestimmung des Reifegrads	53
3.1.2	Methoden zur Bewertung von Fahrerassistenzsystemen . .	55
3.1.3	Ableitung von Testfällen	58
3.2	Diskussion des Stands der Wissenschaft und Technik	60
3.3	Voruntersuchung zum Einfluss der Parametervariation auf den Inhalt von konkreten Szenarien	64
3.3.1	Versuchsaufbau und Durchführung	64
3.3.2	Ergebnisse der Versuchsdurchführung	67
3.4	Ableitung von Anforderungen an ein Testkonzept für das szenariobasierte Testen	71
3.4.1	Allgemeingültige Anforderungen	71
3.4.2	Anforderungen an die Bestimmung des Reifegrads	72
3.4.3	Anforderungen an die Bewertung von Fahrerassistenzsystemen	73
3.4.4	Anforderungen an die Überprüfung des Testinhalts	74
4	Entwurf eines Konzepts zum Testen von Fahrerassistenzsystemen in Fahrscenarien	75
4.1	Bestimmung des Reifegrads eines Fahrerassistenzsystems	76
4.2	Bewertung von Fahrerassistenzsystemen zur Bestimmung der Anforderungserfüllung	83

4.2.1	Testorakel für die Prädiktion des Systemverhaltens	83
4.2.2	Ansatz zur dynamischen Bewertung von Systemen	86
4.2.3	Evaluation und Vergleich der Ansätze	91
4.3	Validierung von simulierten Szenarien	97
4.3.1	Erkennung von Fahrmanövern	99
4.3.1.1	Regelbasierter Ansatz	101
4.3.1.2	KI-basierter Ansatz	103
4.3.1.3	Vergleich der Ansätze	104
4.3.2	Beschreibung und Vergleich der Szenarien	107
4.3.2.1	Beschreibung mittels Manövermatrizen	107
4.3.2.2	Merkmalsbasierte Beschreibung	113
4.4	Zusammenfassung des Testkonzepts	115
5	Prototypische Umsetzung des Testkonzepts	117
5.1	Umsetzung in einer prototypischen SiL-Testumgebung	117
5.1.1	Aufbau der Testumgebung	117
5.1.2	Automatisiertes Notbremssystem	121
5.1.3	Logische Szenarien für die Testdurchführung	123
5.2	Umsetzung an einem HiL-Prüfstand für Fahrerassistenzsysteme	128
5.2.1	Rahmenbedingungen und Projektkontext	128
5.2.2	HiL-Prüfstand bei Mercedes Benz	129
5.2.3	Getestete Fahrerassistenzsysteme	131
6	Evaluation des Testkonzepts	135
6.1	Bewertung relevanter Aspekte des Testkonzepts	135
6.1.1	Korrektheit der Bewertungsergebnisse	135
6.1.2	Zeitliche Auflösung der Bewertungsergebnisse	137
6.1.3	Praktische Relevanz der Szenariovalidierung	140
6.1.4	Übertragbarkeit der Szenariovalidierung	141
6.1.5	Evaluation der Generalisierungskonfidenz	142
6.1.6	Aussagekraft der aggregierten Testergebnisse und des ermittelten Reifegrads	144
6.1.7	Vergleich des Testkonzepts mit der Bewertung mittels Kritikalitätsmetriken	145
6.2	Theoretische Betrachtung wirtschaftlicher Potentiale	149

6.2.1	Reduktion des Aufwands bei der Testerstellung	149
6.2.2	Effizienzsteigerungen bei der Testdurchführung	151
6.3	Diskussion und Fazit	152
7	Zusammenfassung und Ausblick	157
7.1	Zusammenfassung	157
7.2	Ausblick	160
A	Anhang	163
A.1	Zuordnung und Erfüllung der Anforderungen durch das erarbeitete Konzept	163
A.2	Modellierung der Fahrmanöver	166
A.3	Hyperparameter der KNN	169
A.4	Vorstellung der Abbildungsfunktion	170
A.5	Implementierung einer Systembewertung	173
A.6	Aufbau der Komponente „Bestimmung des Reifegrads“	174
A.6.1	Django-Datenbankmodell	175
A.6.2	Implementierte Sichten auf die Testergebnisse	176
A.7	Vorstellung des automatisierten Notbremssystems	179
A.7.1	Anforderungen an das Notbremssystem	179
A.7.2	Abgeleitetes Systemverhalten	180
A.7.3	Systembewertungen für das betrachtete Fahrerassistenzsystem	182
A.8	Ergebnisse der Testdurchführung	184
A.8.1	Bewertungsergebnisse für die logischen Szenarien	184
A.9	Überprüfung der Systembewertungen	189
A.9.1	KS1 - Szenario ohne Aktivierung der FCW und AEB	189
A.9.2	KS2 - Szenario mit Aktivierung der FCW und ohne AEB	191
A.9.3	KS3 - Szenario mit Aktivierung AEB ohne Kollision	192
A.9.4	KS4 - Szenario mit Aktivierung AEB und Kollision	193
	Abkürzungsverzeichnis	197
	Abbildungsverzeichnis	201
	Tabellenverzeichnis	207

Literaturverzeichnis211

1 Einleitung und Motivation

1.1 Motivation

Wer nur an die Technik denkt, hat noch nicht erkannt, wie das autonome Fahren unsere Gesellschaft verändern wird. (Dr. Dieter Zetsche, Vorstandsvorsitzender der Daimler AG, 2005)

Für unsere heutige Gesellschaft nimmt die Bedeutung von Mobilität, sowohl im beruflichen als auch im privaten Umfeld, stetig zu. Laut statistischen Bundesamt nutzten im Jahr 2020 67 Prozent der Berufspendler einen PKW zur Bewältigung des Arbeitswegs [40]. Die dabei zurückgelegte Strecke nahm im Zeitraum von 2010 bis 2020 von durchschnittlich 14.8 km auf bis zu 16.9 km zu, wobei große Unterschiede zwischen Ballungsgebieten und ländlichen Gegenden existieren. Währenddessen stand ein Pendler 2021 in Deutschland durchschnittlich 40 Stunden im Stau [86], in Großstädten, wie beispielsweise München oder Berlin, sogar 87 beziehungsweise 66 Stunden.

Neben der steigenden Fahrleistung durch Berufspendler, nimmt auch der Güterverkehr stetig zu. Im Jahr 2020 verzeichnet allein die deutsche Post einen Anstieg von knapp 400 Millionen Paketen auf einen Rekordwert von insgesamt über 1.6 Milliarden Paketzustellungen (im Vergleich zu 2016) [108]. In Deutschland stieg die Güterverkehrsleistung von 1999 bis 2019 um 75 Prozent an [202]. Der Anteil von LKWs an der Transportleistung lag 2019 bei 71.6 Prozent und damit hat der Straßengüterverkehr in dem genannten Zeitraum ein Plus von 103 Prozent erreicht [116, 202].

Aktuellen Schätzungen zufolge liegt in Deutschland der Anteil an neu zugelassenen PKW mit einem Fahrerassistenzsysteme (FAS) der SAE-Stufe eins oder höher bei über 80 Prozent [104] (vergl. Abbildung 2.1 auf Seite 11). Darüber hinaus hat die Europäische Union im Jahr 2020 die EU-Verordnung Nr. 2019/2144 erlassen, durch die alle neuen Fahrzeugtypen und Erstzulassungen über verschiedene Fahrerassistenzsysteme verfügen müssen. Vorgeschrieben sind dann beispielsweise Notbremsysteme, Notfall-Spurhalteassistenten oder intelligente Geschwindigkeitsassistenten [71].

Das Thema „automatisiertes Fahren“ ist darüber hinaus einer der wichtigsten Trends und Innovationstreiber der Automobilindustrie, neben der Elektrifizierung von Kraftfahrzeugen und der zunehmenden Digitalisierung [5, 7, 79]. Die Entwicklung von automatisierten Fahrzeugen hat in der zweiten Hälfte der letzten Dekade sowohl aus technologischer Sicht als auch aus regulatorischer Sicht immer weiter an Fahrt gewonnen (siehe Abbildung 1.1). Anfang der 2010er-Jahre waren es überwiegend Technologiekonzerne aus dem Silicon Valley, wie Google (später Waymo) oder Uber, die ihre prototypischen Systeme präsentierten. Die ersten tödlichen Unfälle, ausgelöst durch Fehlfunktionen automatisierter Systeme, wie beispielsweise von Tesla [214] oder Uber [131], haben gezeigt, wie wichtig der

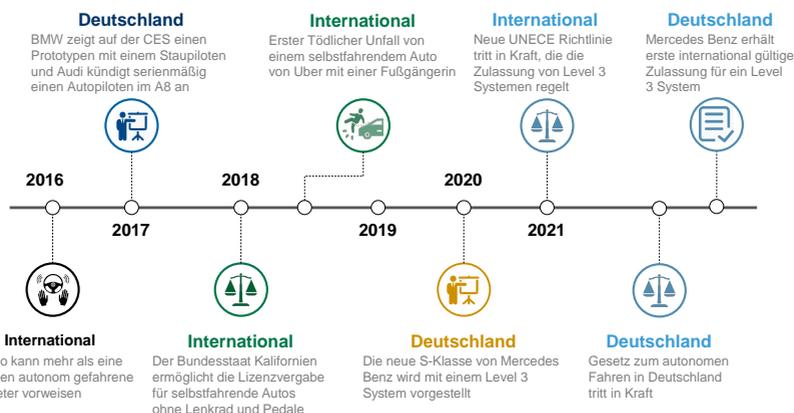


Abbildung 1.1: Übersicht ausgewählter Meilensteine bei der Entwicklung des automatisierten Fahrens

Nachweis der korrekten Funktionalität eines solchen Systems ist. Die Unfallursache war in beiden Fällen eine fehlerhafte Erkennung der Fahrzeugumgebung.

Durch die Schaffung der rechtlichen Rahmenbedingungen für den Betrieb und die Zulassung hoch automatisierter Fahrfunktionen, konnte Mercedes-Benz im Jahr 2021 die erste international gültige Zulassung eines hoch automatisierten Systems erlangen. Das System kann die Fahrzeugführung auf Autobahnen im Stau, oder bei starkem Verkehr, bis zu einer Geschwindigkeiten von 60 km/h übernehmen.

1.2 Herausforderungen beim Testen von Fahrerassistenzsystemen

Bedingt durch seine diverse Umgebung, Vielfalt an möglichen Fahrszenarien und durch unvorhersehbare Interaktionen mit anderen Verkehrsteilnehmern, wie beispielsweise Fußgängern oder Fahrradfahrern, stellt der urbane Raum eine größere Herausforderung für FAS dar als die Autobahn. Sowohl im wissenschaftlichen als auch im regulatorischen und industriellen Umfeld besteht daher Konsens darüber, dass das szenariobasiertes Testen (SBT) ein elementarer Bestandteil des Testens und der Zulassung zukünftiger FAS sein wird (vergl. Abschnitt 2.3.3 ab Seite 36). Die Norm zur Betriebssicherheit von Fahrerassistenzsystemen (ISO-21448), welche sich aktuell in der Finalisierung befindet, stellt ebenfalls Fahrszenarien in den Mittelpunkt ihrer Aktivitäten. Da diese Norm auch für FAS der SAE Stufen eins und zwei gelten wird und den gesamten Entwicklungsprozess begleitet, muss das SBT bereits während der Produktentwicklung eingesetzt werden.

Aktuelle Forschungsprojekte wie PEGASUS¹ oder VV-Methoden² fokussieren sich in ihren Arbeiten auf die Zulassung automatisierter Fahrfunktionen mittels SBT. Während der Zulassung von FAS ist es ausreichend nachzuweisen, dass von

¹ Projekt zur Etablierung von generell akzeptierten Gütekriterien, Werkzeugen und Methoden sowie Szenarien und Situationen zur Freigabe hochautomatisierter Fahrfunktionen

² Verifikations- und Validierungsmethoden automatisierter Fahrzeuge Level 4 und 5

dem FAS in den betrachteten Fahrsczenarien keine Gefährdung für andere Verkehrsteilnehmer ausgeht. Aus diesem Grund wird dabei das allgemeine Fahrzuegverhalten anhand von Kritikalitätsmetriken, wie einer Time-To-Collision (TTC) oder dem Vorhandensein einer Kollision, bewertet (vergl. Abschnitt 2.3.3.4 ab Seite 45).

Im Gegensatz dazu erfolgt die Entwicklung von FAS allerdings auf Grundlage von Systemanforderungen. Der Fortschritt der Systementwicklung wird dabei anhand des Produktreifegrads gemessen, der unter anderem Auskunft darüber gibt, in welchem Umfang das FAS die Systemanforderungen erfüllt. Auf Basis des Produktreifegrads werden dann beispielsweise Entscheidungen über die Freigabe des Systems für Tests im Straßenverkehr oder den Beginn des Zulassungsverfahrens getroffen (vergl. Abschnitt 2.4 ab Seite 47). Bisher erfolgt die Bestimmung des Produktreifegrads noch nicht systematisch für alle möglichen Fahrsczenarien.

Daher ist es möglich, dass auf Grundlage des Produktreifegrads eine Freigabeentscheidung getroffen wird, obwohl das FAS noch nicht in allen Fahrsczenarien richtig funktioniert. Fällt dies erst in einem szenariobasierten Zulassungsprozess auf, kann dies dazu führen, dass die Zulassung nicht erteilt wird. Aus diesem Grund ist die bisherige Bestimmung des Produktreifegrads nicht mehr ausreichend um Freigabeempfehlungen für ein FAS zu geben, welches anschließend einen szenariobasierten Zulassungsprozess durchlaufen muss.

Fahrsczenarien stellen dabei eine abstrakte Beschreibung dessen dar, was im Straßenverkehr auftreten kann. Die Ableitung von Testfällen erfolgt üblicherweise anhand einer automatisierten Variation von verschiedenen Parametern, wie beispielsweise den Geschwindigkeiten von Verkehrsteilnehmern oder von Umweltbedingungen. Daraus resultiert die Herausforderung, dass durch die Parametervariation der genaue Ablauf der erzeugten Szenarien unbekannt ist. Dies verhindert die Verwendung von etablierten Methoden zur Bewertung von Fahrerassistenzsystemen.

Darüber hinaus können durch die Variation der verschiedenen Parameter auch neue Fahrsczenarien entstehen. Dies ist teilweise ein gewollter Effekt des szenariobasierten Testens, da dadurch bisher unbekannte Fahrsczenarien entdeckt werden

können. Um den Produktreifegrad eines FAS bestimmen zu können muss allerdings die Nachverfolgbarkeit von Testergebnissen gewährleistet sein. Dafür muss für jeden durchgeführten Test bekannt sein, welches Fahrscenario getestet wurde.

1.3 Forschungsfragen

Aus den vorgestellten Herausforderungen ergeben sich für die vorliegende Dissertation folgende Forschungsfragen:

Forschungsfrage 1: Wie kann der Produktreifegrad eines Fahrerassistenzsystems in einem szenariobasierten Ansatz ermittelt werden?

Erläuterung: Durch internationale Standards und Regularien, wie beispielsweise die ISO 21448 oder die UNECE Regularie 157, wird das szenariobasierte Testen Grundlage für die zukünftige Zulassung von Fahrerassistenzsystemen werden. Grundlage für die Entscheidung, ob der Freigabe- und Zulassungsprozess gestartet wird, ist unter anderem der Produktreifegrad des Fahrerassistenzsystems. Aktuell existieren noch keine Ansätze, wie dieser in einem szenariobasierten Testansatz bestimmt werden kann.

Forschungsfrage 2: Wie kann bewertet werden, ob die Anforderungen an ein Fahrerassistenzsystem in einem Fahrscenario erfüllt werden?

Erläuterung: Das szenariobasierte Testen verwendet für die Testdurchführung Fahrscenarien, die durch die Variation von Szenarioparametern erzeugt werden. Dies hat zur Folge, dass vor der Ausführung der Tests nicht für jeden Zeitschritt bekannt ist, wie sich das Fahrscenario entwickelt und in welchem Zustand sich das Fahrerassistenzsystem befindet. Dieses Wissen ist allerdings die Grundlage für die Formulierung des erwarteten Verhaltens, mit dem überprüft wird, ob sich das System wie

spezifiziert in dem Fahrscenario verhält. Aus diesem Grund werden neue Methoden benötigt, mit denen Fahrerassistenzsysteme auch ohne Wissen über den genauen Szenarienverlauf bewertet werden können.

Forschungsfrage 3: Wie können durchgeführte Tests dahingehend überprüft werden, dass sie das spezifizierte Fahrscenario enthalten?

Erläuterung: Durch die Variation der Szenarioparameter können auch Fahrscenarien entstehen, die nicht mehr den eigentlich spezifizierten Inhalt der Fahrscenarien enthalten. Um belastbare Aussagen über den Produktreifegrad eines Fahrerassistenzsystems treffen zu können, muss allerdings eine Nachverfolgbarkeit von Testergebnissen zu Fahrscenarien möglich sein.

1.4 Gliederung

Die Arbeit gliedert sich in sieben Kapitel. Nach der Einleitung wird im zweiten Kapitel beschrieben wie in der Automobilindustrie die Entwicklung und die Zulassung von Fahrerassistenzsystemen strukturiert ist. Neben dem allgemeinen Produktentwicklungsprozess, wird auch die Reifegradabsicherung in der Automobilindustrie vorgestellt und detailliert beschrieben, wie aktuelle Fahrerassistenzsysteme während der Entwicklung getestet werden um den Produktreifegrad zu bestimmen. Eine Beschreibung des szenariobasierten Testens, sowie weiterer Testansätze, erfolgt als Erweiterung des aktuellen Stands der Wissenschaft. Anschließend wird im Kapitel drei der Stand der Wissenschaft und Technik diskutiert, sowie Anforderungen an ein szenariobasiertes Testkonzept abgeleitet. In Kapitel vier wird basierend auf den vorgestellten Herausforderungen und Anforderungen ein solches szenariobasiertes Testkonzept erarbeitet. Dieses Konzept wird im Kapitel fünf sowohl an einem Software-in-the-Loop als auch an einem Hardware-in-the-Loop Prüfstand prototypisch umgesetzt und im Kapitel sechs

evaluiert und diskutiert. Die Dissertation wird mit einer Zusammenfassung und einem Ausblick im Kapitel sieben beendet.

2 Entwicklung und Testen von Fahrerassistenzsystemen in der Automobilindustrie

2.1 Automatisierung der Fahrerassistenzsysteme in der Automobilindustrie

Erste Fahrerassistenzsysteme, wie der Tempomat, das Antiblockiersystem (ABS) oder die Antriebsschlupfregelung, sorgten ab den 1970er Jahren für den Beginn einer zunehmenden Automatisierung des Automobils die bis heute anhält [188,205]. Die Anzahl an Steuergeräten (in Englisch *Electronic Control Unit (ECU)*) im Automobil ist von einem Mikrocomputer für die Motorregelung in den 1970er Jahren [26] über 80 Steuergeräte im Jahr 2007 [78] auf bis zu 150 Steuergeräten im Jahr 2015 gestiegen [188]. In einem Premiumklasse-Fahrzeug waren im Jahr 2020 bis zu 100 Millionen Zeilen Code implementiert [66]. Prognosen sagen darüber hinaus zwischen 300 und 500 Millionen Zeilen Software-Code für das automatisierte Fahren voraus [27]. Der Anteil der Software-Entwicklung an der Entwicklung von Fahrerassistenzsystemen gewinnt dadurch zunehmend an Bedeutung [66].

„Fahrerassistenzsystem“ ist ein Überbegriff für Sensor-Aktor-Systeme in Kraftfahrzeugen, die den Fahrer passiv oder aktiv bei der Durchführung der Fahraufgabe unterstützen. Ziel ist es den Fahrkomfort zu erhöhen und durch die Vermeidung

von Unfällen, sowie der Minimierung von Schadenspotential, die Verkehrssicherheit zu erhöhen. Fahrerassistenzsysteme sollen die Beanspruchung des Fahrers reduzieren, Defizite bei der Aufnahme und Verarbeitung der relevanten Fahrerinformation beseitigen, den Fahrer möglichst wenig zusätzlich beanspruchen und sich bei Fahrzeugeingriffen möglichst so verhalten wie der Fahrer selbst [23]. Die Fahraufgabe (in Englisch *Dynamic Driving Task (DDT)*) lässt sich grundsätzlich in drei Ebenen unterteilen: *Stabilisierung*, *Lenkung* und *Navigation* [29]. Diese historische Einteilung aus dem Jahr 1970 lässt sich auf den aktuellen Stand der Technik übertragen, der die Fahraufgabe für das automatisierte Fahren in operative, taktische und strategische Funktionen unterteilt [95].

Die Stabilisierungsebene (operativ) umfasst die grundlegende Bewegungssteuerung des Fahrzeugs in Längs- und Querrichtung, um es beispielsweise in der Fahrspur zu halten oder um mit einer konstanten Geschwindigkeit zu fahren. Auf der Lenkungsebene (taktisch) werden alle Aktivitäten zusammengefasst, welche die Beobachtung der Umgebung, die Interaktion mit anderen Verkehrsteilnehmern oder die Wahl der richtigen Geschwindigkeit beinhalten. Die Ebene der Navigation (strategisch) beinhaltet die Planung und Auswahl der Fahrtroute, sowie andere Aufgaben, die über die eigentliche Fahraufgabe hinaus gehen.

Der „DRIVE PILOT“ von Mercedes Benz ist das erste System, das den Anspruch erhebt ein automatisiertes Fahrersystem (in Englisch *Automated Driving Systems (ADS)*) zu sein [2].

Definition 2.1 (Automatisiertes Fahrersystem) *Hardware und Software die gemeinsam in der Lage sind die gesamte Fahraufgabe dauerhaft durchzuführen. Dies muss auch gewährleistet sein, wenn das System außerhalb der definierten Operational Design Domain (ODD) betrieben wird [95].*

Definition 2.2 (Operational Design Domain) *Definierte Bedingungen, für die ein automatisiertes System entwickelt wurde und in denen es funktionieren muss [95].*

Für die Klassifikation der Automatisierungsgrade von FAS wurden auf nationaler und internationaler Ebene 6 Stufen definiert (siehe Abbildung 2.1) [95, 205]. Für

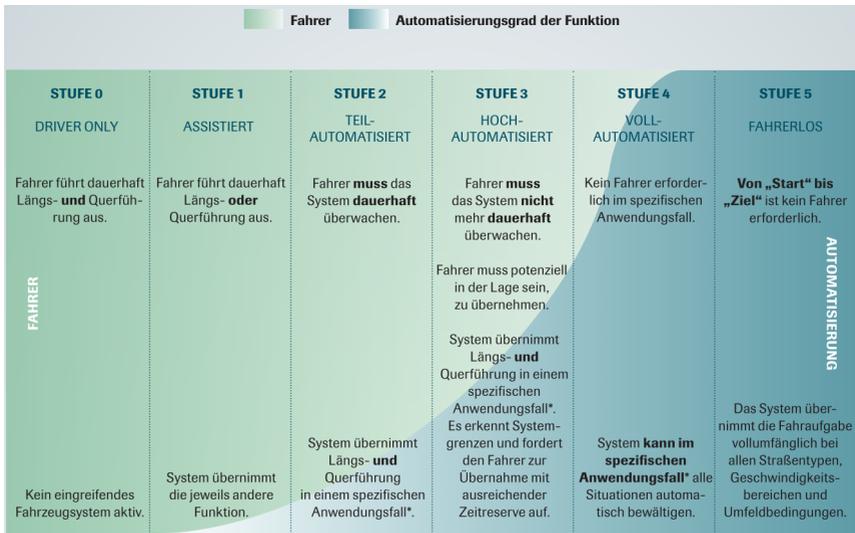


Abbildung 2.1: Automatisierungsgrade des automatisierten Fahrens aus [205]

jede Stufe wird beschrieben, welchen Teil der Fahraufgabe der Fahrer übernehmen muss und welche Aufgabe das automatisierte System übernimmt. Bis einschließlich der Automatisierungsstufe zwei wird von Fahrerassistenzsystemen gesprochen und ab der Automatisierungsstufe drei von automatisierten Fahrsystemen [95].

2.1.1 Herausforderungen des automatisierten Fahrens

Automatisierte Fahrsysteme, wie zum Beispiel ein urbaner Kreuzungsassistent, müssen jederzeit in der Lage sein sich sicher im Straßenverkehr bewegen zu können. Dafür müssen sie ein Modell ihrer Umgebung aufbauen und sich, sowie andere Verkehrsteilnehmer, darin lokalisieren. Um eine zuverlässige Umfeldwahrnehmung bei unterschiedlichen Wetterbedingungen realisieren zu können, werden unterschiedlichen Sensortechnologien miteinander kombiniert. Aktuelle Sensor-konzepte sehen dafür bis zu acht verschiedene Sensorarten, wie beispielsweise

Lidar, Radar, Kamera, Ultraschall, Mikrofone, Drucksensoren, Feuchtigkeitssensoren und hoch genaue Lokalisierungssysteme vor [2].

Im Vergleich zu Fahrerassistenzsystemen bis zur Automatisierungsstufe zwei, reagiert ein ADS nicht mehr nur auf andere Verkehrsteilnehmer, sondern die Interaktion mit der Umgebung und dem umgebenden Verkehr nimmt eine zentrale Rolle für diese Systeme ein.

Definition 2.3 (reagieren) *Auf etwas (besonders einen bestimmten Reiz) in irgendeiner Weise eine Wirkung zeigen, ansprechen. [58]*

Definition 2.4 (Interaktion) *Aufeinander bezogenes Handeln zweier oder mehrerer Personen; Wechselbeziehung zwischen Handlungspartnern. [58]*

Im Fall von Fahrerassistenzsystemen unterscheidet sich die Reaktion von der Interaktion darin, dass bei der Reaktion lediglich das eigene Fahrzeug eine Wirkung auf das Verhalten eines anderen Verkehrsteilnehmers zeigt. Dies ändert sich, wenn ein ADS zusätzlich mit anderen Verkehrsteilnehmern oder der Verkehrsinfrastruktur kommuniziert und interagiert. In diesem Fall muss explizit betrachtet werden, welche Wechselwirkungen durch die Interaktionen entstehen können.

Eine weitere Herausforderung für den Betrieb von ADS ist, dass sie sich in einem offenen Kontext bewegen, der nicht vollständig bekannt und ständigen Veränderungen unterworfen ist [133]. Die Begriffe Kontext und offener Kontext werden für ADS nach [84] definiert als:

Definition 2.5 (Kontext) *Erfassung aller Elemente, die möglicherweise Einfluss auf das Systemverhalten haben können.*

Definition 2.6 (Offener Kontext) *Kontext, der formal nicht vollständig beschrieben werden kann und der auf eine Lücke, zwischen dem beschriebenen Kontext und dem tatsächlich real anzutreffenden Kontext, verweist.*

Ein Kamera-basiertes Erkennungssystem kann beispielsweise nur Objekte erkennen, für die es einen geeigneten Klassifikator besitzt. Neue Verkehrsteilnehmer oder Verkehrszeichen, wie der E-Scooter nach der Zulassung im Jahr 2019 oder



Abbildung 2.2: Neue Verkehrszeichen der StVO Novelle 2020 nach [105]. Überholverbot von Zweirädern (links), Mehrfachbesetzte Pkw (Mitte) und Parkplatz für Carsharing (rechts)

die neuen Verkehrszeichen der StVO Novelle im Jahr 2020 (siehe Abbildung 2.2), können von vorher entwickelten Systemen nicht erkannt werden und es sind Aktualisierungen für das System notwendig. Aktuell besitzen die wenigstens Automobilhersteller sowohl die Technologie, als auch die dafür notwendigen Update-Zyklen, um solche Aktualisierungen über Software Updates over the Air (SOTA) durchzuführen [171] [194].

Im Fehlerfall, oder wenn das ADS außerhalb seiner ODD betrieben wird, muss ein ADS zusätzlich in der Lage sein dies zu erkennen und einen sicheren Fahrzustand zu erreichen. Ein wichtiger Unterschied bei dem Übergang von der Automatisierungsstufe zwei zu drei ist, dass bis zur Stufe drei der Fahrer das System dauerhaft überwachen muss und als Rückfallebene jeder Zeit zur Verfügung steht. Diese Aufgaben werden ab der Stufe drei schrittweise vom ADS übernommen. Damit verbunden steigt der Anspruch an die Zuverlässigkeit der Fahrerassistenzsysteme und auch an die Aussagekraft und Zuverlässigkeit der Funktionsnachweise. Um dies zu gewährleisten steht die Automobilindustrie vor der Herausforderung, dass sie bestehende Entwicklungsprozesse um neuartige Entwicklungs- und Testmethoden erweitern muss.

2.2 Produktentwicklungsprozess und Reifegradabsicherung

Die Entwicklung neuer Fahrerassistenzsysteme oder Fahrzeuge wird in der Automobilindustrie durch einen Produktentwicklungsprozess (PEP) strukturiert [209]. Dieser umfasst verschiedene Entwicklungsphasen, spezifiziert Meilensteine (MS) mit dazugehörigen Liefergegenständen und definiert Prozessketten, sowie beteiligte Akteure und Rollen. Jeder Automobilhersteller (auch Original Equipment Manufacturer (OEM) genannt) verwendet einen eigenen PEP, der auf sein Unternehmen angepasst ist.

Abbildung 2.3 zeigt einen schematischen Produktentstehungsprozess inklusive Meilensteine der Volkswagen AG für Fahrzeug-Neuprojekte. Auch wenn die Phase der Produktentwicklung im PEP mit dem Meilenstein der Beschaffungsfreigabe endet, beinhalten die nachfolgenden Phasen weiterhin Entwicklungsaktivitäten mit unterschiedlichen Freigabestufen [6]. Für die Entwicklung von Fahrerassistenzsystemen können beispielsweise folgende Freigabestufen (FS) vorhanden sein:

- FS1: Freigabe für Teststrecken
- FS2: Experten-Straßenfreigabe
- FS3: Dauerlauf-Freigabe

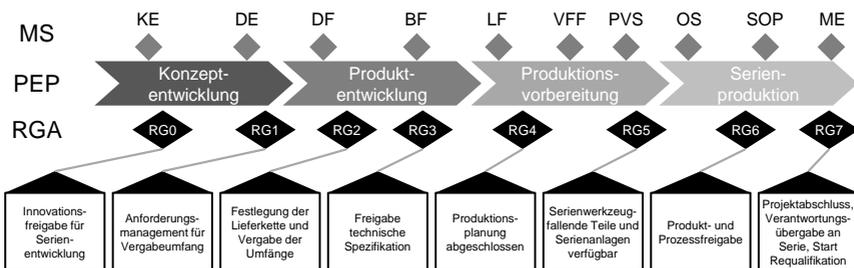


Abbildung 2.3: Schematische Übersicht eines PEP der Volkswagen AG mit Reifegradstufen und exemplarischer PEP-Meilensteine nach [6]

- FS4: Allgemeine Straßenfreigabe
- FS5: Serienfreigabe

Moderne Fahrerassistenzsysteme kombinieren mehrere Funktionalitäten in einem System (siehe Tabelle 2.1). Der neue Abstands-Assistent des Herstellers Mercedes Benz umfasst beispielsweise neben seiner Hauptfunktionalität, der Abstands-

Fahrerassistenzsystem	Hauptfunktionalität	Zusätzliche Funktionalität
Abstandsassistent	Geschwindigkeits- und Abstandsregelung	- Streckenbasierte Geschwindigkeitsanpassung - Stauendefunktion - Geschwindigkeitslimit-Assistent - Stau-Assistent
Bremsassistent	Warnung des Fahrers und Durchführung einer Notbremsung	- Kreuzungsfunktion - Stauende-Notbremsfunktion - Abbiegefunktion
Lenkassistent	Spurhaltefunktion	- Spurwechselassistent - Nothalt-Assistent - Rettungsgassenfunktion
Verkehrszeichenassistent	Erkennung von Geschwindigkeitslimits	- Fußgängerübergangswarnfunktion - Falschfahr-Warnfunktion - Stoppschild-Warnfunktion - Rote-Ampel-Warnfunktion
Totwinkelassistent	Anzeigen von Fahrzeugen im toten Winkel	- Ausstiegswarnfunktion - Aktive Ambientebeleuchtung - MBUX Interieur-Assistent

Tabelle 2.1: Übersicht der Fahrerassistenzsysteme des EQS unterteilt in Hauptfunktionalität und zusätzliche Funktionalität nach [83]

und Geschwindigkeitsregelung, auch zusätzlich die Funktionalitäten (in Englisch Features) [83]:

- Streckenbasierte Geschwindigkeitsanpassung
- Stauendefunktion
- Geschwindigkeitslimit-Assistent
- Stau-Assistent

Während der Systementwicklung wird das Fahrerassistenzsystem inkrementell entwickelt und mit der Zeit um Funktionalitäten ergänzt. Die zeitliche Planung, wann eine Funktionalität während der Entwicklung verfügbar sein soll, wird im sogenannten Feature Roll-Out Plan (FRP) festgehalten (siehe Abbildung 2.4).

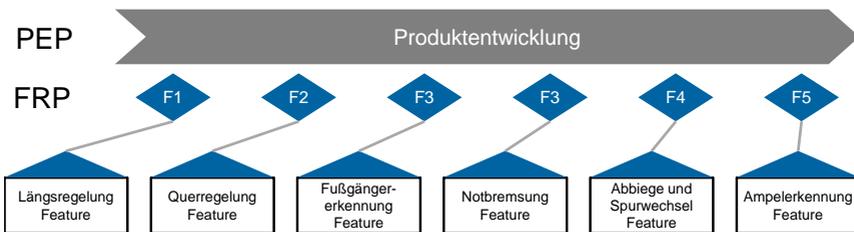


Abbildung 2.4: Beispielhafter Feature Roll-out Plan für eine automatisierte Fahrfunktion

Weiterhin wird zur Sicherstellung der Produktqualität parallel zum PEP auch eine Reifegradabsicherung (RGA) durchgeführt, die den Produktreifegrad (nachfolgend als Reifegrad (RG) bezeichnet) zu festgelegten Zeitpunkten bestimmt [53].

Definition 2.7 (Produktreifegrad) *Der Produktreifegrad ist die objektive Beurteilung eines Produktes anhand von festgelegten Indikatoren [212]. Während der Produktentwicklung gibt der Produktreifegrad an, wie gut ein System funktional die Anforderungen erfüllt, durch die es spezifiziert wurde [174].*

Das Ziel der RGA ist die Harmonisierung von Inhalten und Abläufen in der Lieferkette, die die Produktqualität sicherstellt [53]. Der VDA-Standard „Reifegradabsicherung für Neuteile“ definiert acht Reifegradstufen [53], die im vorgestellten PEP von Volkswagen verwendet werden (siehe Abbildung 2.3).

Die Bewertung des Reifegrads geschieht anhand von festgelegten Indikatoren, für die ein spezifischer Messkriteriensatz definiert ist. Jedes Messkriterium wird mit einer Ampelfarbe (Rot, Gelb, Grün) bewertet und ist so formuliert, dass es mit „ja“ oder „nein“ eindeutig beantwortet werden kann (siehe Tabelle 2.2). Messkriterien für den Reifegrad-Indikator „Produkt-Absicherung“ sind laut VDA-Standard beispielsweise

- Die Simulations- und Testplanung zum Nachweis der Produktqualität ist festgelegt
- Bugs (Software) werden effektiv bis zur Fehlerabstellung verfolgt
- Funktionalität des Produkts ist durch Lieferant (z.B. Zulieferer) nachgewiesen

Bewertung	Definition
Rot	- Das Messkriterium wird mit „Nein“ beantwortet - Mindestens ein Projektziel ist nicht erreichbar und - Maßnahme bedeutet eine Zielanpassung
Gelb	- Das Messkriterium ist mit „Nein“ beantwortet - Eine Maßnahme ist erforderlich und vereinbart - Alle Projektziele werden mit den festgelegten Maßnahmen erreicht
Grün	- Das Messkriterium wird mit „Ja“ beantwortet - Keine zusätzlichen Aktivitäten sind notwendig

Tabelle 2.2: Definition der Reifegradampel nach [53]

- Funktionalität des Produkts ist im Gesamtfahrzeug/Aggregat durch den Kunden (z.B. OEM) nachgewiesen [53]

Die Bestimmung des Messkriteriums erfolgt wiederum anhand von festgelegten Bewertungskriterien. Bewertungskriterien für den Nachweis der Funktionalität im Gesamtfahrzeug/Aggregat sind beispielsweise, dass alle relevanten Testfälle berücksichtigt wurden und dass die Funktionalität anhand von Erprobungsfahrzeugen, Prüfständen oder digitalen Prototypen nachgewiesen wurde. Eine Gesamtbewertung für den Reifegrad ergibt sich aus der Aggregation der verschiedenen Einzelbewertungen der Messkriterien, wobei die Gesamtbewertung der schlechtesten Bewertung eines Einzelkriteriums entspricht.

2.2.1 Produktentwicklung nach dem V-Modell

Die Einführung des V-Modells als Vorgehensmodell stellt einen wichtigen Meilenstein in der Geschichte der Automobilindustrie dar. Die Grundidee des V-Modells, den Entwicklungsprozess als sequenziellen Lebenszyklus darzustellen, wurde allerdings bereits 1956 von H.Benington auf dem Symposium für „advanced programming methods for digital computers“ vorgestellt [113]. Benington beschreibt in seiner Veröffentlichung Probleme, die bei der Erstellung von großen Computerprogrammen vorkommen [25]. Dies sind die Bedienung von Computern, die Zuverlässigkeit von Programmen oder Systemen, der Support und die Dokumentation. Um diesen Problemen zu begegnen, schlägt Benington einen Prozess mit neun Phasen vor. Dieser Entwicklungsprozess wurde dann 1979 von Boehm in die bekannte Form des V-Modells gebracht, um die verschiedenen Integrationsstufen des Entwicklungsprozesses zu visualisieren [37]. Im Jahr 1992 wurde das V-Modell von der deutschen Bundeswehr als Vorgehensmodell für die Softwareentwicklung von öffentlichen IT-Projekten definiert und seit dem mehrfach überarbeitet [13].

Grundlage für den heutigen Entwicklungsprozess bildet das V-Modell 97 aus dem Jahr 1997 [113], dass zu einem Systementwicklungsstandard umgewandelt wurde und seitdem auch die Hardwareentwicklung beinhaltet [33]. Im Gegensatz zu den

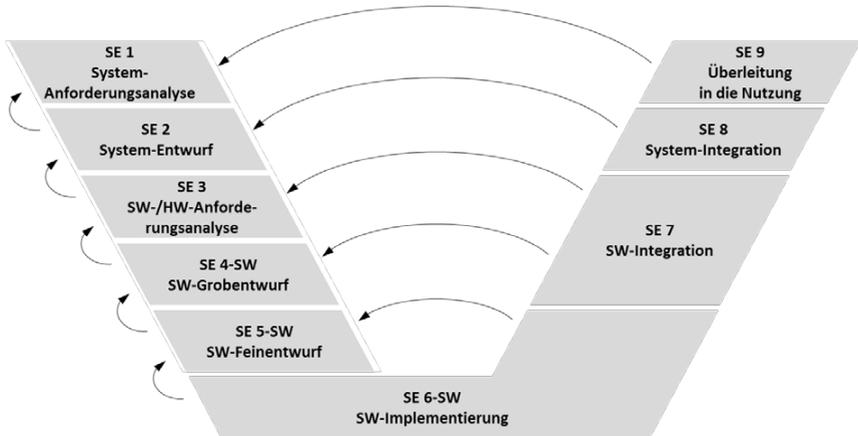


Abbildung 2.5: Struktur des Systementwurfs nach dem V-Modell [34]

Prozessen von Benington und Boehm wird nicht nur die eigentliche Systementwicklung (SE) betrachtet (siehe Abbildung 2.5), sondern auch die Qualitätssicherung (QS), das Konfigurationsmanagement (KM), sowie das Projektmanagement (PM) sind als Submodule ein integraler Bestandteil des Prozesses [34]. Dafür werden im V-Modell 97 zusätzliche Rollen für jedes Submodul definiert. Eine Rolle beschreibt dabei alle notwendigen Fähigkeiten und Kenntnisse, die ein Rolleninhaber für die Erfüllung seiner Aufgaben benötigt [35]. Als Reaktion auf den veränderten Stand der Technik der Informationstechnologie wurde das V-Modell XT 2005 veröffentlicht [54]. Der Namenszusatz XT steht dabei für *extreme tailoring* und soll die hohe Anpassungsfähigkeit des V-Modells an unterschiedliche Projekte und Organisationen verdeutlichen. Zusätzlich wird das V-Modell um Entscheidungspunkte erweitert, die regelmäßige Qualitätsmesspunkte (in Englisch Quality Gates) repräsentieren. Zu diesen Zeitpunkten wird der aktuelle Projektfortschritt evaluiert und wenn nötig die Projektplanung angepasst.

Die Systementwicklung ist so strukturiert, dass die Spezifikation und Zerlegung des zu entwickelnden Systems auf dem linken Schenkel des V-Modells erfolgt. Basierend auf der Analyse von Systemanforderungen wird ein erster technischer Systementwurf erstellt und die Realisierbarkeit untersucht.

Definition 2.8 (Anforderung) *Eine Bedingung oder Fähigkeit, die ein System oder eine Systemkomponente erfüllen oder besitzen muss, um einen Vertrag, eine Norm, eine Spezifikation oder andere formell vorgeschriebene Dokumente zu erfüllen [96].*

In mehreren Top-Down Dekompositionsschritten werden zunächst Anforderungen an die Software (SW) und Hardware (HW) analysiert und darauf aufbauend ein Grob- und Feinentwurf der Software erstellt.

Definition 2.9 (Dekomposition) *Die Dekomposition bezeichnet den Prozess der Auflösung einer Beziehung in ihre Bestandteile. Die ursprüngliche Funktion muss anschließend aus den Bestandteilen vollständig rekonstruiert werden können [137].*

Jedem Dekompositionsschritt auf der linken Seite des V-Modells wird ein entsprechender Integrationsschritt auf der rechten Seite des V-Modells zugeordnet. Parallel zu der Integration finden auf den unterschiedlichen Integrationsstufen Testaktivitäten statt (vgl. Abschnitt 2.3.2.2) [20, 36].

Die Integration des V-Modells in einen Produktentwicklungsprozess der Automobilindustrie ist in Abbildung 2.6 exemplarisch dargestellt. Die Systementwicklung wird in eine Software-, Hardware und Mechanikentwicklung unterteilt, in denen das V-Modell mehrfach durchlaufen wird. Jede abgeschlossene Entwicklung nach dem V-Modell endet mit einem Release.

Definition 2.10 (Release) *Die ausgelieferte Version eines Systems, welche Teile des Systems oder das gesamte System beinhalten kann. (verallgemeinert nach [101])*

Dabei verlaufen die drei Entwicklungsstränge asynchron zueinander und die Zeit für einen Durchlauf des V-Modells variiert zwischen den Entwicklungssträngen und über den PEP. Die Synchronisierung der Software-, Hardware- und Mechanikentwicklung erfolgt anhand von Musterständen zu definierten Zeitpunkten im Produktentstehungsprozess. Die verschiedenen Musterstände werden in Abschnitt 2.3.2.2 detailliert erläutert. [53]

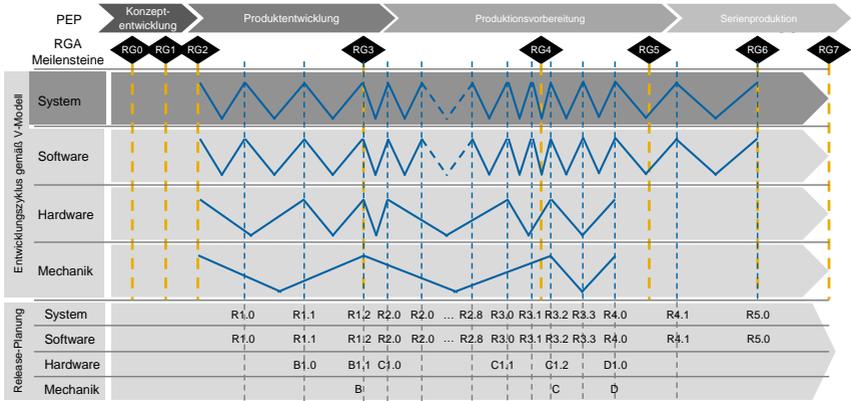


Abbildung 2.6: Integration der Entwicklung nach dem V-Modell in den Produktentstehungsprozess nach [53]

2.3 Entwicklungsbegleitendes Testen von Fahrerassistenzsystemen

Durch die zunehmende Übertragung der Fahraufgabe auf Fahrerassistenzsysteme, nimmt die Bedeutung eines vertrauenswürdigen und aussagekräftigen Nachweises der korrekten Funktionalität eines FAS zu. Dieser Nachweis wird durch die Prozesse Verifikation und Validierung (V&V) erzeugt. Der Begriff Verifikation leitet sich von dem lateinischen Begriff *veritas* ab, der soviel bedeutet wie „Wahrheit“. Formal definiert der Standard ISO/IEC 24765 den Begriff als:

Definition 2.11 (Verifikation) *Die Verifikation beschreibt den Prozess der Bewertung eines Systems oder einer Komponente, um festzustellen, ob die Produkte einer bestimmten Entwicklungsphase die zu Beginn dieser Phase auferlegten Bedingungen erfüllen [101].*

Informell lässt sich die Frage formulieren

Baue ich das Produkt richtig?

Die Validierung dagegen stammt von dem lateinischen *valere* ab, was mit „etwas wert sein“ übersetzt werden kann.

Definition 2.12 (Validierung) *Die Validierung beschreibt den Prozess der Nachweiserbringung, dass ein System oder eine Komponente die gestellten Anforderungen erfüllt. Bei der Validierung wird überprüft, ob das System das beabsichtigte Verhalten besitzt und die Kundenwünsche erfüllt [101].*

Analog zur Verifikation lässt sich für die Validierung die Frage formulieren [37].

Baue ich das richtige Produkt?

Erfüllt ein entwickeltes System die gestellten Anforderungen nicht, kann dies unterschiedliche Ursachen haben. Im Gegensatz zu elektronischen und mechanischen Komponenten, liegt die Ursache für Mängel- oder fehlerbehaftete Software immer in der Entwicklung [13]. Ein Fehler entsteht nach ISO 29119 durch folgende Ursachenkette [99]: Ein fehlerhaftes Verhalten des Entwicklers, beispielsweise die falsche Interpretation von Anforderungen oder handwerkliche Fehler bei der Programmierung, führt zu einem Fehlerzustand oder Defekt in dem Software-Produkt. Dies können unter anderem ein Bug oder eine fehlerhafte Spezifikation sein. Solange dieser Defekt nicht während der Nutzung der Software auftritt, besitzt er noch keinen Einfluss auf den Betrieb der Software. Wird die Software allerdings unter ungünstigen Bedingungen ausgeführt, tritt der Fehlerzustand ein und kann dazu führen, dass die Software die Anforderungen vom Nutzer nicht mehr erfüllt. Damit liegt eine Fehlerwirkung vor [186]. Diese Ursachenkette ist in Abbildung 2.7 dargestellt. Formal werden die Begriffe Fehlverhalten, Fehlerzustand und Fehlerwirkung nach [101] definiert als:



Abbildung 2.7: Darstellung der Ursachenkette von einem Fehler nach ISO29119 [99]

Definition 2.13 (Fehlverhalten) *Der Begriff Fehlverhalten beschreibt eine menschliche Handlung, die zu einem falschen Ergebnis führt, z.B. Software, die einen Fehlerzustand enthält.*

Definition 2.14 (Fehlerzustand) *Der Begriff Fehlerzustand beschreibt eine Manifestation eines Fehlers in der Software oder einen Defekt in einem Hardware-Gerät oder Komponente.*

Definition 2.15 (Fehlerwirkung) *Der Begriff Fehlerwirkung beschreibt die Beendigung der Fähigkeit eines Produkts, eine erforderliche Funktion zu erfüllen, oder seine Unfähigkeit, innerhalb vorher festgelegter Grenzen zu arbeiten.*

2.3.1 Funktionale Sicherheit und Betriebssicherheit von hoch automatisierten Fahrzeugen

Im Bereich des assistierten und automatisierten Fahrens können Fehlerzustände auch durch externe Beeinflussung des Fahrzeugs entstehen. Mögliche Fahrzeugzustände, die während des Fahrens auftreten können, unterteilen sich in sichere Fahrzustände und unsichere Fahrzustände (siehe Abbildung 2.8). Während eines sicheren Fahrzustands geht von dem Fahrzeug keine Gefahr für die Insassen und andere Verkehrsteilnehmer aus. Im Gegensatz dazu stellen unsichere Fahrzustände grundsätzlich ein Risiko dar. Zusätzlich wird zwischen bekannten und unbekanntem Fahrzuständen unterschieden (vgl. Abschnitt 2.1.1). Aus dem Vorhandensein von Kombinationen unbekannter und gleichzeitig unsicherer Fahrzustände folgt, dass es keine 100%ige Sicherheit geben kann und immer ein Restrisiko besteht. Ziel aller Entwicklungs- und Testaktivitäten ist daher, das Restrisiko auf ein noch vertretbares Minimum zu reduzieren und den Nachweis zu erbringen, dass das System sich innerhalb der bekannten Fahrzustände sicher verhält. [189]

Definition 2.16 (Restrisiko) *Das nach dem Einsatz von Sicherheitsmaßnahmen verbleibende Risiko [100].*

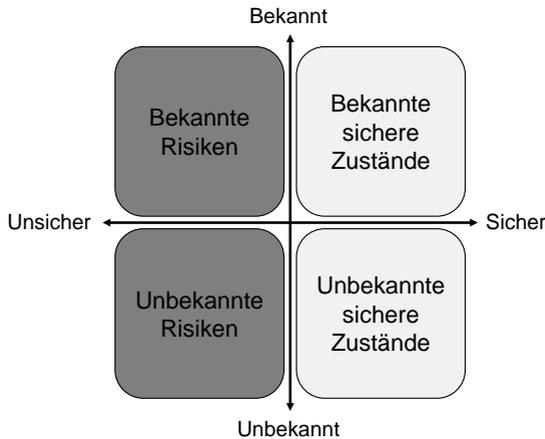


Abbildung 2.8: Klassifikation der möglichen Fahrzustände beim hochautomatisiertem Fahren nach [189]

Definition 2.17 (Risiko) *Kombination aus der Wahrscheinlichkeit des Schadenseintritts und der Schwere des Schadens [100].*

Bei der Betrachtung der Sicherheit wird zwischen einer funktionalen Sicherheit und einer Betriebssicherheit (in Englisch Safety Of The Intended Functionality (SOTIF)) unterschieden.

Definition 2.18 (Funktionale Sicherheit) *Nichtvorhandensein eines unververtretbaren Risikos aufgrund von Gefährdungen im Fehlerfall eines E/E-Systems [100].*

Definition 2.19 (Betriebssicherheit) *Nichtvorhandensein eines unververtretbaren Risikos aufgrund von Gefährdungen, die sich aus funktionalen Mängeln der vorgesehenen Funktionalität oder aus einer angemessenen vorhersehbaren Fehlanwendung durch Personen ergeben [102].*

Definition 2.20 (E/E-System) *System das aus elektrischen und/oder elektronischen Elementen besteht, inklusive programmierbarer elektrischer Elemente [100].*

Die funktionale Sicherheit grenzt sich von der Betriebssicherheit dahingehend ab, dass sie Gefährdungen im Fehlerfall betrachtet, während sich die Betriebssicherheit mit Gefährdungen im Normbetrieb befasst.

Der Umgang mit Risiken im Bereich der funktionalen Sicherheit ist in der ISO 26262 beschrieben [100]. Sie klassifiziert mögliche Fehler, der zu entwickelten Systeme, nach den drei Kriterien „Schwere des Fehlers“, „Eintrittswahrscheinlichkeit“ und „Beherrschbarkeit des Fehlers“ und ordnet ein entsprechendes Automotive Safety Integrity Level (ASIL) zu. In Abhängigkeit der ASIL-Klassifikation werden Sicherheitsanforderungen an das System gestellt, die während des Entwicklungs- und Testprozesses beachtet werden müssen.

Das Thema Betriebssicherheit wird in der Norm ISO 21448 behandelt. Neben externen Faktoren, kann auch das spezifizierte System selbst eine Quelle für Gefahren sein [102]. Der Ansatz der SOTIF versucht daher nachzuweisen, dass das System, so wie es spezifiziert wurde, in der gesamten ODD sicher funktioniert. Die Grundlage für die Beschreibung der ODD und die Identifikation von Gefahren sind Fahrscenarien. Analog zu [189] unterteilt die Norm die möglichen Fahrscenarien in sichere oder unsichere und bekannte oder unbekannte Fahrscenarien.

Mit Hilfe des SOTIF-Prozesses werden sowohl funktionale Unzulänglichkeiten als auch externe Auslösebedingungen identifiziert und evaluiert, die zu einer Gefährdung im Straßenverkehr führen können. Die funktionalen Unzulänglichkeiten umfassen sowohl fehlerhafte Spezifikationen als auch Performancelimitierungen des Systems, wie beispielsweise eine fehlerhafte Klassifikation von Objekten oder das Auftreten von Fehldetektionen bei Regen. Aus den Ergebnissen der Evaluation wird dann das Systemdesign iterativ verbessert, um so die funktionalen Unzulänglichkeiten zu entfernen. Um die Sicherheit der entwickelten Funktionalität in der ODD schlussendlich nachzuweisen, müssen sowohl alle bekannten und unsicheren als auch alle unbekanntes und unsicheren Szenarien evaluiert werden. Wie die Bestimmung des Restrisikos in den unbekanntes Szenarien erfolgen kann, wird durch die Norm allerdings nicht näher erläutert. Es wird lediglich eine Liste an

möglichen Methoden genannt, wie beispielsweise Flottentests, Szenariexploration in der realen Welt oder der zufälligen Erzeugung von Eingangsdaten für das System.

2.3.2 Anforderungsbasiertes Testen von Kraftfahrzeugen

Das Testen von Fahrerassistenzsystemen findet klassischer Weise im rechten Schenkel des V-Modells statt (siehe Abbildung 2.5). Das vorherrschende Testkonzept ist dabei das anforderungsbasierte Testen, bei dem Systemanforderungen die Grundlage für die Testfallgenerierung bilden [89, 199].

Definition 2.21 (Testen) *Eine Tätigkeit, bei der ein System oder eine Komponente unter bestimmten Bedingungen ausgeführt wird, die Ergebnisse beobachtet oder aufgezeichnet werden und eine Bewertung eines Aspekts des Systems oder der Komponente vorgenommen wird [101].*

Definition 2.22 (Regressionstest) *„Erneutes Testen eines bereits getesteten Programms bzw. einer Teilfunktionalität nach deren Modifikation. Ziel ist es nachzuweisen, dass durch die vorgenommenen Änderungen keine Fehlerzustände eingebaut oder (bisher maskierte Fehlerzustände) freigelegt wurden.“ [103]*

Im Rahmen der vorliegenden Dissertation wird der Begriff Bewertung definiert als:

Definition 2.23 (Bewertung) *Die systematische Bestimmung des Umfangs, in dem das beobachtete Verhalten eines Systems oder einer Komponente dem spezifizierten Verhalten entspricht [101].*

Dabei gelten folgende Richtlinien [169]:

Kein Test ohne Anforderung

Zu jeder Anforderung mindestens ein Test

2.3.2.1 Testfalldefinition und Testabdeckung

Die vorgestellten Richtlinien schaffen ein systematisches Testkonzept, welches garantieren soll, dass alle Systemanforderungen getestet werden und somit keine Testlücken entstehen. Bezogen auf die möglichen Fahrzustände eines ADS (vgl. Abschnitt 2.3.1), eignet sich der Ansatz nur für das Testen bekannter und in Anforderungen beschriebener Risiken [189]. Um mögliche Testlücken zu identifizieren und den Reifegrad des Systems bestimmen zu können, wird die Testabdeckung (in Englisch test coverage) als Metrik genutzt.

Definition 2.24 (Testabdeckung) *Der Grad, ausgedrückt als Prozentsatz, bis zu dem bestimmte Testabdeckungselemente (zum Beispiel Anforderungen) durch einen oder mehrere Testfälle abgedeckt werden [99]*

Definition 2.25 (Testabdeckungselement) *Ein Attribut oder eine Kombination von Attributen, die aus einer oder mehreren Testbedingungen abgeleitet wird, indem eine Testdesigntechnik verwendet wird, die die Messung der Vollständigkeit der Testausführung ermöglicht [99]*

Die Wahl des Testabdeckungselementes beeinflusst die Vollständigkeit der resultierenden Testabdeckung maßgeblich. Das NSPLF-Schema bietet die Möglichkeit die notwendige Testabdeckung für eine gewählte Teststrategie zu bestimmen. Jede Kategorie des Schemas definiert die notwendigen Testabdeckungselemente [170].

N None: Das Testobjekt wird zufällig oder implizit getestet

S Sparsely: Für jedes Feature existiert mindestens ein Testfall

P Partly: Für jede Anforderung existiert mindestens ein Testfall

L Largely: Für jede Bedingung existiert mindestens ein Testfall

F Fully: Zusätzliche Testfälle basierend auf der Erfahrung

Das Schema kann nur verwendet werden, wenn die Hauptbestandteile des Testobjekts bekannt und dokumentiert sind.

Definition 2.26 (Testobjekt) *Erzeugtes Arbeitsprodukt, das Gegenstand der Tests ist [99].*

Definition 2.27 (Software Feature) *Ein charakteristisches Merkmal einer Software (z.B. Leistung, Übertragbarkeit oder Funktionalität) [98].*

Aus der erreichten Abdeckung und dem Ergebnis der einzelnen Testfälle kann auf die Erfüllung der Messkriterien für den Reifegrad geschlossen werden (vgl. Abschnitt 2.2).

Grundlage der praktischen Testdurchführung bei Automobilherstellern ist eine Testspezifikation.

Definition 2.28 (Testspezifikation) *Komplette Dokumentation des Testdesigns, der Testfälle und der Testprozeduren für ein bestimmtes Testobjekt [99].*

Sie enthält eine vollständige Beschreibung aller durchzuführender Testfälle. Darüber hinaus können Informationen über Verantwortlichkeiten, den Umfang der Tests oder das Testdesign enthalten sein (siehe Abbildung 2.9) [99]. Testfälle bestehen aus einer Vorbedingung und mehreren Testschritten und sind so strukturiert, dass sie das Testobjekt aus einem initialen Zustand mit mehreren Schritten

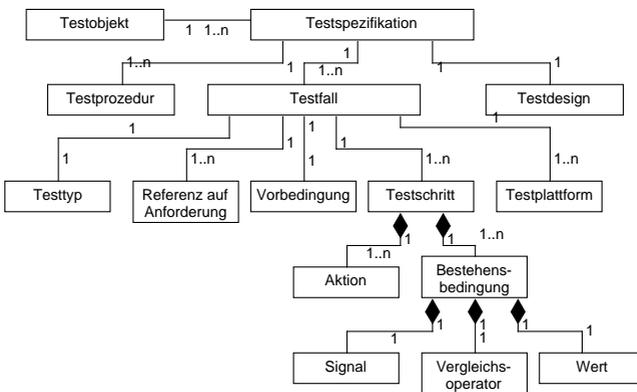


Abbildung 2.9: Grundsätzlicher Aufbau der Testspezifikation eines deutschen Automobilherstellers

in einen gewünschten Zustand führen. In jedem Testschritt wird anhand von Signalüberprüfungen getestet, ob definierte Bestehensbedingungen (in Englisch pass conditions) erfüllt sind und sich das System korrekt verhält. Dadurch wird sichergestellt, dass der Systemzustand des Testobjekts während jedem Testschritt bekannt ist.

Für ein Notbremssystem (in Englisch Autonomous Emergency Braking System (AEB)) beinhaltet ein Testfall beispielsweise die Annäherung an einen Fußgänger und es wird geprüft, ob das System zu definierten Zeitpunkten eine Warnung ausgibt oder eine Notbremsung durchführt (siehe Abbildung 2.10). In dem letzten Testschritt wird dann geprüft, ob eine Kollision mit dem Fußgänger vorgekommen ist oder nicht. Im Fehlerfall erfolgt automatisch eine Lokalisierung und Dokumentation des Fehlers durch die ausgewerteten Bestehensbedingungen. Diese Informationen werden dann zusammen mit einer Messung von ausgewählten Signalen an die Entwicklungsabteilung weitergeleitet. Zusätzlich wird für jeden

<p>Vorbedingung: Action: Ego fährt innerorts mit 30 km/h Bestehensbedingung: Ego_SpeedX >= 30 km/h</p>	
<p>Testschritt 1: Action: Fußgänger überquert die Straße Action: Ego nähert sich Fußgänger ohne zu reagieren Bestehensbedingung: AEBs wird aktiviert und warnt optisch und akustisch AEBs_Acust_Warn = 1 AEBs_Visual_Warn = 1</p>	
<p>Testschritt 2: Action: Ego nähert sich Fußgänger ohne zu reagieren Bestehensbedingung: AEBs vollführt Notbremsung und Warnung bleibt bestehen AEBs_Acust_Warn = 1 AEBs_Visual_Warn = 1 AEBs_Brk_Rq = 1 AEBs_AccelX_Rq = -5.5 m/s²</p>	
<p>Testschritt 3: Action: Situation ist unkritisch durch Systemreaktion Bestehensbedingung: Kollision wurde verhindert, AEBs ist nicht mehr aktiv AEBs_Acust_Warn = 0 AEBs_Visual_Warn = 0 AEBs_Brk_Rq = 0 Ego_SpeedX = 0 km/h Radar_TgtDistX > 0 m</p>	

Abbildung 2.10: Exemplarische Beschreibung und Visualisierung eines Testfalls für ein Notbremssystem

Testfall beschrieben, was der Testtyp ist, zum Beispiel Funktionstest oder Schnittstellentest, auf welcher Testplattform er ausgeführt werden soll (vgl. Abschnitt 2.3.2.3) und welche Anforderungen er überprüft.

Neben der Ableitung von Testfällen aus Systemanforderungen, schreibt die ISO26262 in Abhängigkeit der ASIL-Klassifikation weitere Testdesigntechniken vor, wie die Bildung von Äquivalenzklassen, Grenzwertanalyse oder dem Raten von Fehlern [100].

2.3.2.2 Integrationsstufen beim Testen von Fahrerassistenzsystemen

Zeit und Kosten sind treibende Faktoren in der automobilen Entwicklung. Aus diesem Grund sollen Fehler so früh wie möglich und nicht erst im fertigen Fahrzeug gefunden werden. Um dies zu ermöglichen stellt die Entwicklung nach dem V-Modell jedem Entwurfsschritt einen Integrationsschritt mit Testaktivitäten gegenüber (vgl. Abschnitt 2.2.1). Grundsätzlich kann beim automobilen Testen zwischen vier Teststufen unterschieden werden, den Modul-, Komponenten-, Integrations- und Gesamtsystem-/Fahrzeugtests (siehe Abbildung 2.11) [147]. Der Testinhalt auf jeder Teststufe wird dabei in funktionale Tests und nicht-funktionale

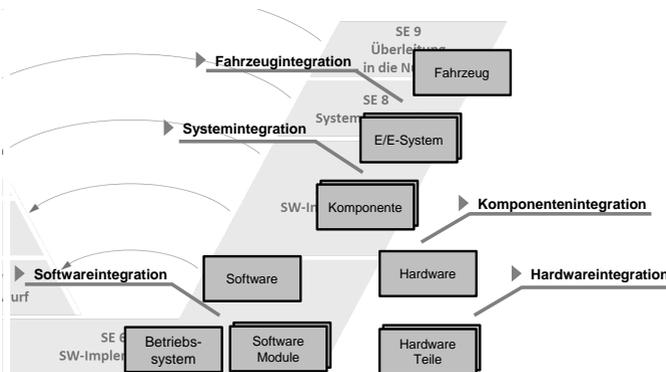


Abbildung 2.11: Systemdekomposition und Integrationsstufen nach [89]

Tests unterteilt. Mit funktionalen Tests wird das eigentliche Funktionsverhalten des Fahrerassistenzsystems überprüft und nicht-funktionale Tests beinhalten beispielsweise Schnittstellentests oder die Überprüfung des Zeitverhaltens von Steuergeräten.

Modul- und Komponententests verifizieren ob Hardware- oder Software Komponenten korrekt entwickelt wurden. Während des Entwurfsschritts „*SE-6 SW-Implementierung*“ des V-Modells (vgl. Abschnitt 2.2.1) überprüfen Modultests die korrekte Implementierung von Funktionsmodellen oder geschriebenen Quellcode [13]. Nach der Integration der Software in ein Steuergerät, im Schritt „*SE-7 SW-Integration*“, dienen die Komponententests unter anderem der Überprüfung der Echtzeitfähigkeit des Systems, dem Finden von elektrischen Fehlern und dem Testen der Ein- und Ausgabeschicht [89].

Definition 2.29 (Echtzeitsystem) *Ein System, das seine Eingangs- und Zustandsgrößen innerhalb eines vorgegebenen Zeitintervalls garantiert verarbeitet und innerhalb des Zeitintervalls die zugehörigen Ausgangs- und Zustandsgrößen erzeugt [89].*

Um einen sicheren Betrieb von FAS gewährleisten zu können, muss die Umgebungserfassung auch bei unterschiedlichen Umwelteinflüssen korrekt funktionieren. Daher nimmt neben klassischen Komponententests der Steuergeräte, die Bedeutung von Sensortests auf der Komponentenebene immer weiter zu [31,32,124].

Während die zuvor beschriebenen Komponententests meistens beim Zulieferer der Komponente durchgeführt werden, führen die Automobilhersteller die Integrationstests selbst durch (vgl. RG-Messkriterien im Abschnitt 2.2) [49]. Das Ziel von Integrationstests ist es, das Verhalten vernetzter Steuergeräte zu testen. Ein Fahrerassistenzsystem stellt eine Wirkkette aus einem Sensor, der Umweltdaten erfasst, einer Funktion, die diese Daten auswertet und eine Reaktion berechnet, und einem Aktor, der diese Reaktion ausführt, dar. Im Rahmen der Systemintegration wird getestet, ob die Komponenten im Zusammenspiel funktionieren und die Kommunikation zwischen diesen wie spezifiziert durchgeführt wird [163, 169]. Weiter lassen sich Integrationstests in Software-, Hardware-, Komponenten- und Systemintegrationstests unterteilen (siehe Abbildung 2.11) [13].



Abbildung 2.12: Zuordnung der Musterphasen zu den einzelnen Reifegraden nach [53]

Musterstände in der Automobilindustrie Nach der Integration aller E/E-Systeme mit der Mechanik im Fahrzeug werden Gesamtsystemtests durchgeführt. Die Testfahrzeuge nutzen ausschließlich reale Sensoren, Bussysteme und Steuergeräte. Damit werden Fehler ausgeschlossen, die bei Labortests durch unvollständige Modelle, Restbus- und Umgebungssimulationen entstehen. Entlang des Entwicklungsprozesses dienen verschiedene Musterstände (A-, B-, C-, D-Muster) zur Bestätigung von Funktionsprinzipien, der Erprobung und schlussendlich zur Erreichung der technischen Freigabe (siehe Abbildung 2.12). Die verschiedenen Musterstände werden nach Vorgaben des Verbandes der Automobilindustrie (VDA) folgendermaßen charakterisiert [53]

A-Muster sind erste, bedingt fahrtaugliche Funktionsmuster, die zur Bestätigung des Funktionsprinzips und zur grundsätzlichen Funktionsdarstellung dienen. Eine Verwendung im öffentlichen Straßenverkehr ist bei fahrsicherheitsrelevanten Systemen nicht möglich und die Verwendung ist auf Versuche im Labor, der Simulation oder einem Prüfstand beschränkt.

Im Gegensatz dazu, sind B-Muster funktionsfähige und fahrtaugliche Muster, die eine ausreichende Betriebssicherheit für erste Erprobungsprototypen gewährleisten. Nach erfolgter Straßenfreigabe durch eine Zulassungsbehörde, können die Prototypen mit einer Sonderzulassung im öffentlichen Straßenverkehr genutzt werden. Damit ist das Testen des gesamten Funktionsumfang, sowie eine Dauererprobung zur Ermittlung und Beseitigung von Schwachstellen möglich.

Definition 2.30 (Erprobung) „Die Erprobung ist eine Analyse des Gesamten, dem eine vergleichsweise geringe Systematik zugrunde liegt. In Erproben steckt das Ausprobieren in einer unbestimmten Situation.“ [169]

B-Musterstände erfüllen alle Hardware- und Funktionsanforderungen, aber sind noch nicht in allen Punkten serientauglich. Zusätzlich beginnt mit dem B-Muster die Design-Validierung durch den Lieferanten und Kunden.

C-Muster werden zur Erreichung der technischen Freigabe verwendet. Wie auch bei den B-Musterständen, ist nach der Erteilung einer Sonderzulassung die Nutzung im öffentlichen Straßenverkehr möglich. Im Gegensatz zum B-Muster, sind keine technischen Einschränkungen zugelassen, die eine Auswirkung auf die Fahr-sicherheit haben. Die Herstellung erfolgt mit Serienwerkzeugen unter serien-nahen Bedingungen, wobei Einschränkungen gegenüber der geplanten Serienproduktion zulässig sind. Sie erfüllen alle Anforderungen, ermöglichen den Abschluss der Design-Validierung und den Start der Fertigungsprozess-Validierung.

Mit dem D-Muster wird eine Produktionsprozessfreigabe und Produktfreigabe durchgeführt. Es dient als Typgenehmigungsmuster zur Freigabe durch Zulassungsstellen (vgl. Abschnitt 2.4). Dafür ist das Muster uneingeschränkt ein-satzfähig, besitzt die Serienfunktionalität und erfüllt alle Qualitätsanforderungen.

2.3.2.3 Testplattformen in der Automobilindustrie

In der Automobilindustsubrie stehen verschiedene Testplattformen zur Verfügung, die für die Ausführung von Testfällen verwendet werden können.

X-in-the-Loop Für den Test von Regelungssystemen werden in der Automobi-lindustrie sogenannte X-in-the-Loop (XiL) Testplattformen eingesetzt.

Definition 2.31 (X-in-the-Loop (XiL)) „X-in-the-Loop bezeichnet die Ver-wendung eines einheitlichen Simulationsmodells für den Test eines Regelungs-systems über alle Phasen des Entwicklungsprozesses hinweg“ [8, 181] zitiert nach [13]

Das Simulationsmodell umfasst dabei die Umgebungssimulation des Testobjekts sowie eine Restbussimulation, welche das Restfahrzeugverhalten sowohl zeitlich, als auch inhaltlich korrekt wiedergibt [93].

Definition 2.32 (Restbussimulation) *Virtuelle Nachbildung des Kommunikationsverhaltens von Steuergeräten und Kopplung zu einem Testobjekt zur Prüfung der Kommunikation im Steuergerätenetzwerk [176]*

In Abhängigkeit des verwendeten Testobjekts werden folgende XiL-Testplattformen unterschieden.

Model-in-the-Loop (MiL) beschreibt eine Testplattform, die ein entwickeltes Funktionsmodell als Testobjekt verwendet. Das Funktionsmodell beschreibt das gewünschte Verhalten eines Systems und wird im nächsten Entwicklungsschritt zur automatischen Generierung von Softwarecode verwendet [89]. MiL-Tests dienen der frühen Verifikation von Algorithmen und der entwickelten logischen Systemarchitektur [13].

Software-in-the-Loop (SiL) beschreibt eine Testplattform, die implementierten Seriencode als Testobjekt verwendet. Der Seriencode ist innerhalb der Software-Gesamtarchitektur eingebettet und wird in einer virtualisierten Laufzeitumgebung betrieben [13]. Die Laufzeitumgebung kann dabei entweder lokal auf einem Desktop-System oder parallelisiert in Rechenzentren oder einer Cloud ausgeführt werden [207]. In Abhängigkeit des Testobjekts und der notwendigen Umgebungssimulation können SiL-Tests auch schneller als in Echtzeit ausgeführt werden und damit die notwendige Testzeit verkürzen.

Hardware-in-the-Loop (HiL) beschreibt eine Testplattform, die eine entwickelte Zielhardware als Testobjekt verwendet. Historisch beinhaltet der Begriff HiL zwingend das Schließen einer Regelschleife durch Streckenmodelle im Testsystem. In der heutigen Entwicklung wird der Begriff auf alle Domänen eines Fahrzeugs ausgedehnt und wird auch für Testobjekte ohne geschlossene Regelschleife verwendet, wie beispielsweise für das Außenlicht, der Sitzverstellung oder dem Innenlicht [89]. Die Verwendung von realen Steuergeräten führt zu einer Beschränkung der Durchführungsgeschwindigkeit von Tests, da Steuergeräte Daten in Echtzeit verarbeiten (vgl. Definition 2.29). Aufgrund von hohen Kosten und langen Auf- und Umrüstzeit von HiL-Systemen, ist eine einfache Parallelisierung der Tests durch mehrere Prüfstände nicht möglich.

Fahrzeugtests Die XiL-Technologie betreibt das Testobjekt in einer simulierten Umgebung, die durch Simulationsmodelle gebildet wird. Da Modelle immer eine Vereinfachung der Wirklichkeit darstellen, müssen Testergebnisse aus Simulationen hinsichtlich ihrer Übertragbarkeit auf die Realität kritisch hinterfragt werden. Aus diesem Grund werden mit den verschiedenen Musterständen auch entwicklungsbegleitende Fahrzeugtests in der realen Welt durchgeführt. In Abhängigkeit des Testziels werden Fahrzeugtests auf dem Prüfgelände und im Realverkehr voneinander unterschieden.

Ab der Freigabestufe zwei (vgl. Abschnitt 2.2) können, nach der Erteilung einer Sonderzulassung, Testfahrten im Straßenverkehr durchgeführt werden. Während die ersten Erprobungen noch von speziell ausgebildeten Testfahrern übernommen werden müssen, dürfen ab der FS4 beispielsweise auch Mitarbeiter des Automobilherstellers im Rahmen der kundennahen Fahrerprobung die neu entwickelten Fahrzeuge im Straßenverkehr verwenden [73]. In den Testfahrzeugen sind Datenlogger verbaut, welche die Buskommunikation aufzeichnen und so umfangreiche Datensätze erzeugen. Auffälligkeiten im Systemverhalten während der Fahrt werden durch den Fahrer mittels Zeitstempel und Anmerkung dokumentiert und im Anschluss der Fahrt analysiert [176].

Die Verkehrssicherheit muss während der Durchführung von Tests dauerhaft gewährleistet sein. Diese Sicherheit kann während früher Entwicklungsphasen im Straßenverkehr nicht garantiert werden. Daher werden auf dem Prüfgelände Fahrversuche mit Kulissen und dynamischen Attrappen von Fußgängern, Fahrradfahrern und anderen Fahrzeugen durchgeführt (siehe Abbildung 2.13). Die Fahrversuche basieren auf definierten Prüf- und Fahrmanöverkatalogen, die für ein Notbremssystem unter anderem durch die European New Car Assessment Programme (Euro NCAP)-Kataloge vorgegeben sind [144]. Die verwendeten Attrappen werden entweder über Schienen, ferngesteuerte Roboter oder Seilsysteme bewegt. Zur Auswertung der Fahrversuche wird die Fahrzeug-interne Buskommunikation aufgezeichnet und externe Referenzsensorik verwendet.



Abbildung 2.13: Möglichkeiten zur Bewegung von Fußgängern und Fahrradfahrern auf dem Prüfgelände (links: Freeboard aus [4], rechts: Seilsystem aus [3])

2.3.3 Szenariobasiertes Testen in der Automobilindustrie

Erste Fahrerassistenzsysteme, wie beispielsweise der Tempomat für die Längsregelung, wurden ausschließlich von dem Fahrzustand des eigenen Fahrzeugs beeinflusst [205]. Die Weiterentwicklung des Tempomats, der adaptive Abstandsregeltempomat, reagiert bereits auf andere Verkehrsteilnehmer und passt die eigene Sollgeschwindigkeit auf die Geschwindigkeit eines vorausfahrenden Fahrzeugs an (siehe Abbildung 2.14). Aktuelle Systeme, wie ein prädiktiver Abstandsregeltempomat, reduzieren die Sollgeschwindigkeit automatisch beim Passieren von Geschwindigkeitslimits, vor Kurven oder wenn Fahrzeuge rechts überholt werden würden. Damit werden sie zunehmend durch die Interaktion mit dem umgebenden Verkehr und der Umgebung charakterisiert (vgl. Abschnitt 2.1.1).

Wie in Abschnitt 2.3.2 beschrieben, verwenden die etablierten Testkonzepte Systemanforderungen als Basis für die Testfalldefinition. Diese sprachlich formulierten Anforderungen beschreiben ausschließlich das bekannte gewünschte Verhalten des eigenen Fahrzeugs und berücksichtigen dabei nur die Reaktion auf andere Verkehrsteilnehmer. Die Herausforderungen bei FAS liegen allerdings nicht mehr

nur im System selbst, sondern in externen Faktoren, wie der Umgebung oder in Interaktionen mit anderen Verkehrsteilnehmern (vgl. Definition 2.4).

Für das Testen leitet sich daraus die Konsequenz ab, dass jede zu testende Interaktion zwischen dem eigenen Fahrzeug und einem anderen Verkehrsteilnehmer zunächst bekannt sein muss und zusätzlich durch eine Anforderung beschrieben wird. So müsste für das in Abschnitt 2.1.1 beschriebene Beispiel eines Erkennungssystems alle möglichen Kombinationen aus Wettereffekten, Lichtverhältnissen und Eigenschaften der Fußgängerbekleidung dokumentiert werden. Betrachtet man den dafür notwendigen Aufwand und die Granularität der entsprechenden Anforderungen, dann kann angenommen werden, dass dieser Beschreibungsaufwand nicht geleistet werden kann. Aus dieser Problematik und dem Umstand, dass nicht alle möglichen Kombinationen a priori bekannt sind, folgt die Konsequenz, dass das anforderungsbasierte Testen nicht mehr ausreicht, um höher automatisierte FAS zu testen und neue Testkonzepte benötigt werden.

Ein vielversprechender Ansatz dafür ist das sogenannte szenariobasiertes Testen. Im Gegensatz zu etablierten Testkonzepten, bilden dabei Fahrscenarien die

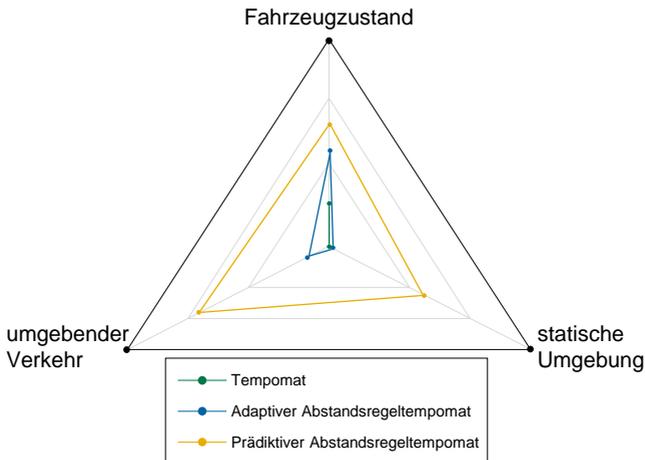


Abbildung 2.14: Exemplarische Visualisierung verschiedener Einflussfaktoren auf unterschiedliche Fahrerassistenzsysteme

Grundlage für die Testfallerstellung und es findet eine Validierung an realen Use-Cases statt. Der Grundgedanke des szenariobasierten Testens ist, dass alle theoretisch relevanten Fahrscenarien aufgezählt werden können und ein FAS in diesen Szenarien getestet wird. Daraus kann dann eine Aussage über Sicherheit des Systems abgeleitet werden [51]. Dabei wird folgende Annahme verwendet: wenn ein Fahrzeug in der Lage ist ein bestimmtes Szenario erfolgreich zu absolvieren, beispielsweise eine Notbremsung auf einen kreuzenden Fußgänger bei 30 km/h, dann kann es auch andere, ähnliche Szenarien erfolgreich absolvieren [182], zum Beispiel eine Notbremsung bei 29 oder 31 km/h. Diese Generalisierung erlaubt eine Reduktion des Testaufwandes, da nicht mehr jedes Szenario getestet werden muss.

Zentrales Element der szenariobasierten Testens ist eine Datenbank mit relevanten Fahrscenarien. Diese Szenarien werden aus realen Fahrzeugdaten extrahiert oder können beispielsweise aus Analysen oder Unfalldaten gewonnen werden [157, 160, 215]. Für die Extraktion von Szenarien aus Realdaten existieren eine Vielzahl von unterschiedlichen Ansätzen [61, 130, 165]. Durch die Extraktion der Szenarien aus realen Fahrdaten, eignet sich das szenariobasierte Testen für die Identifikation von bisher unbekanntem Fahrzuständen und damit zur Reduzierung des Restrisikos (vgl. Abschnitt 2.3.1). Im Entwicklungsprozess werden Szenarien in der Konzeptphase, bei der Spezifikation von technischen Sicherheitsanforderungen und während des abschließenden Testens eingesetzt [100].

2.3.3.1 Definition eines Szenarios

Für den Begriff *Szenario* gibt es im automobilen Umfeld keine einheitliche Definition oder Terminologie. Ulbrich et al. definieren die Begriffe Szenario, Szene, Szenerie und Situation folgendermaßen [201]:

Definition 2.33 (Szenario) *Ein Szenario beschreibt die zeitliche Entwicklung zwischen mehreren Szenen in einer Sequenz aus Szenen. Jedes Szenario startet mit einer Startszene. Aktionen und Events, ebenso wie Ziele und Werte können spezifiziert werden um diese zeitliche Entwicklung innerhalb eines Szenarios*

zu charakterisieren. Im Gegensatz zu einer Szene umfasst ein Szenario eine gewisse Zeitspanne.

Definition 2.34 (Szene) Eine Szene beschreibt eine ideale Momentaufnahme der Umgebung/Umwelt inklusive der Szenerie, der dynamischen Elemente, der Zustände aller Teilnehmer und der Beziehungen zwischen allen statischen und dynamischen Objekten.

Definition 2.35 (Szenerie) Alle räumlich statischen Objekte eines Szenarios, wie geometrische Elemente (z.B. Straßennetz mit Verbindungen und Knoten) und globale Attribute (z.B. das Wetter und die Tageszeit). Der Zustand der räumlich statischen Objekte kann sich mit der Zeit verändern (z.B. Ampeln).

Definition 2.36 (Situation) Alle relevanten Umstände (eingeschlossen die Szene, Ziele + funktionspezifische Aspekte), die für die Auswahl einer geeigneten Verhaltensweise zu einem bestimmten Zeitpunkt notwendig sind. Die Situation ist immer eine subjektive Auswahl von Informationen aus einer Szene, erweitert um eine Intension.

2.3.3.2 Beschreibung eines Szenarios

Im Rahmen des Forschungsprojektes PEGASUS¹ wurde ein 6-Ebenen-Modell entwickelt, welches zur konkreten Beschreibung von Szenarien genutzt werden kann (siehe Abbildung 2.15). Dieses Modell verwendet unterschiedliche Ebenen, um die verschiedenen Aspekte und Eigenschaften eines Szenarios darzustellen [156]. Im Gegensatz zur Definition von Ulbrich et al., liegt der Fokus bei dem 6-Ebenen-Modell auf einer Darstellungsform, die beispielsweise für Simulationen genutzt werden kann.

Die ersten zwei Ebenen L1 und L2 des Modells beschreiben die Basisstrecke, auf dem das Szenario stattfindet. Dazu gehört sowohl die Straßentopologie und deren Beschaffenheit als auch die Leitinfrastruktur wie Begrenzungen, Markierungen

¹ gefördert vom Bundesministerium für Wirtschaft und Energie (BMWi)



Abbildung 2.15: Darstellung des 6-Ebenen-Modells zur Beschreibung von Szenarien nach [156]

und Verkehrszeichen. Temporäre Überlagerungen, wie geänderte Spurführungen oder Leiteinrichtungen in Baustellen, werden in der dritten Ebene L3 beschrieben. Bewegliche Objekte, sowie die manöverbasierte Interaktion zwischen Verkehrsteilnehmern ist Bestandteil der vierten Ebene L4. Die fünfte Ebene L5 wird für die Modellierung von Umweltbedingungen und deren Einfluss auf die Ebenen L1 bis L4 verwendet. Digitale Informationen, wie Vehicle-to-Everything (V2X)-Kommunikation oder digitale Karten, werden in der letzten Ebene L6 beschrieben. Für eine standardisierte, maschinenlesbare Darstellung von Szenarien, finden Aktivitäten im Rahmen der OpenScenario-Initiative statt. Aktuell ist der Standard in der Version OpenScenario 1.1.0 verfügbar [11] und es wird parallel bereits an einer kompletten Überarbeitung des Standards gearbeitet [12].

Der Standard OpenScenario, sowie aktuelle Forschungsarbeiten auf dem Gebiet der Szenariobeschreibung, verwenden Akte für die zeitliche Abstraktion von Szenarien [11, 15]. Diese Beschreibung greift auf eine Terminologie von Theater und Film zurück, welche von [201] und [76] eingeführt wurde. Der Ablauf eines Szenarios wird als Aneinanderreihung oder Sequenz von Akten definiert (siehe Abbildung 2.16). Innerhalb eines Aktes sind alle Manöver der Teilnehmer explizit

definiert und ändern sich nicht. Ändert sich das Manöver eines Teilnehmers, findet ein Übergang in den nächsten Akt statt.

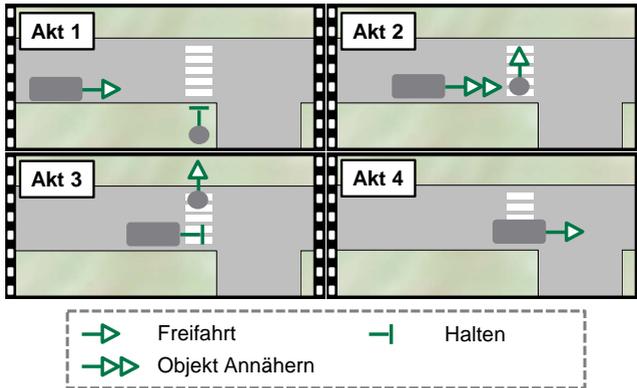


Abbildung 2.16: Akt-basierte Darstellung eines Notbrems-Szenarios mit kreuzendem Fußgänger nach [15]

Bagschik et al. erweitern die Definition eines Szenarios um Abstraktionsebenen [74]. Sie schlagen in ihrer Arbeit folgende drei Abstraktionsebenen für Szenarien vor (siehe Abbildung 2.17).

Funktionale Szenarien stellen die oberste und abstrakteste Ebene von Szenarien dar. Auf dieser Ebene werden Betriebsszenarien des Entwicklungsgegenstands auf semantischer Ebene gesammelt. Die Entitäten und Beziehungen zwischen den Entitäten der Anwendungsdomäne werden in sprachlich gefassten Szenarien ausgedrückt. Die Szenarien sind widerspruchsfrei. Das Vokabular der funktionalen Szenarien ist spezifisch für den Anwendungsfall und die -domäne und kann unterschiedliche Detailgrade aufweisen.

Die nächste Ebene wird durch die sogenannten *Logischen Szenarien* gebildet, welche in der Definition eine Detaillierung der funktionalen Szenarien im physikalischen Zustandsraum darstellen. Sie stellen Betriebsszenarien durch Entitäten und Beziehungen dieser Entitäten mithilfe von Parameterräumen im Zustandsraum dar.

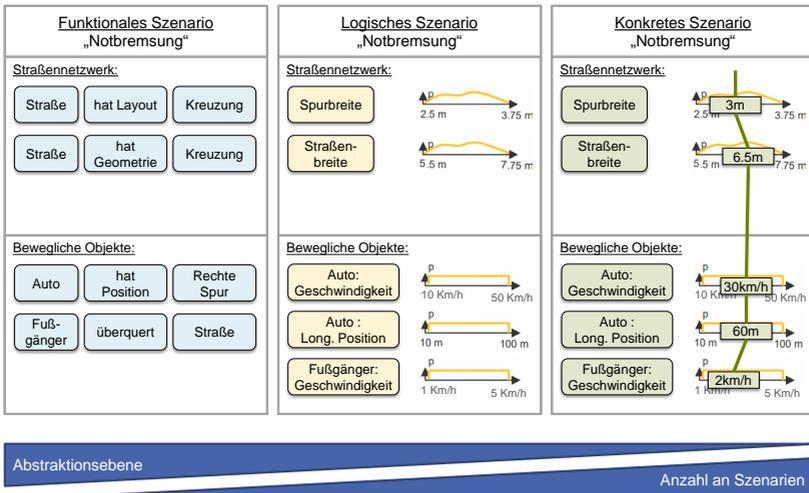


Abbildung 2.17: Darstellung der Abstraktionsebenen von Szenarien nach [74]

Definition 2.37 (Parameterraum) Die Menge aller möglichen Werte eines Parameters oder der Kombination mehrerer Parameter wird Parameterraum genannt (angelehnt an [187]).

Für die einzelnen Parameterräume können optional statistische Verteilungen angegeben werden. Zusätzlich können optional die Beziehungen der Parameterräume zueinander mithilfe von Korrelationen oder numerischen Bedingungen modelliert werden. Logische Szenarien enthalten eine formale Beschreibung von Szenarien.

Als unterste Ebene werden *Konkrete Szenarien* definiert. Sie stellen Fahrscenarien eindeutig durch Entitäten und Beziehungen dieser Entitäten mithilfe von festen Werten im Zustandsraum dar.

Im Gegensatz zu Testfällen, bei denen das gewünschte Systemverhalten und die Beschreibung des Testablaufs eine monolithische Einheit bilden (vgl. Abschnitt 2.3.2.1), beinhalten Szenarien nur eine allgemeine Beschreibung des Verkehrsgeschehens und keine Informationen über das Systemverhalten eines FAS.

2.3.3.3 Testprozess für eine szenariobasierte Freigabe von ADS

Im Rahmen des Forschungsprojekts PEGASUS wurde untersucht, wie die Zulassung von hochautomatisierten Fahrzeugen realisiert werden kann. Dafür wurde ein szenariobasierter Prozess entworfen, der zentral auf einer Datenbank mit allen relevanten logischen Szenarien aufbaut (siehe Abbildung 2.18). Aus den logischen Szenarien werden anhand von Auftretenswahrscheinlichkeiten und definierten Bestehenskriterien logische Testfälle abgeleitet [153]. Anschließend sieht der PEGASUS-Prozess eine Verteilung der logischen Testfälle auf verschiedene Testplattformen, sowie eine Generierung von konkreten Testfällen mittels Parametervariation vor. Konkrete Testfälle bestehen aus einem konkreten Szenario, sowie Szenen- oder Szenariobewertungskriterien (siehe Abbildung 2.19) [190].

Das Testkonzept sieht vor, dass der größte Anteil der Tests in einer SiL-Umgebung durchgeführt werden soll. Das Prüfgelände dient dazu, ausgewählte, besonders relevante Szenarien nachzustellen um die Simulationsergebnisse zu verifizieren.

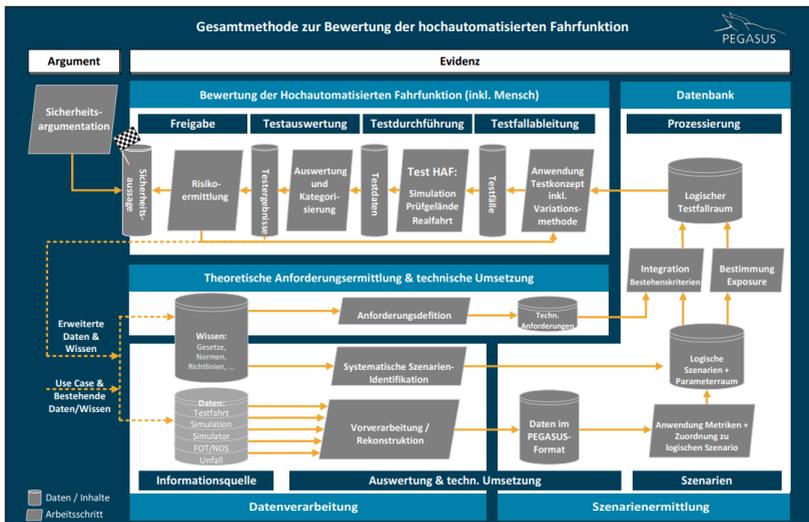


Abbildung 2.18: Szenariobasierter Testprozess zur Freigabe von ADS aus [154]

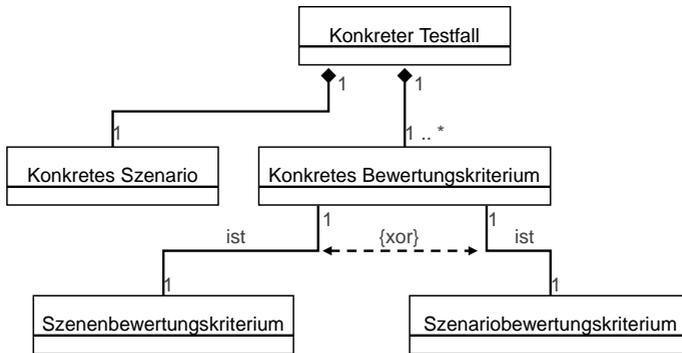


Abbildung 2.19: Definition eines Testfalls für das szenariobasierte Testen nach [190]

Feldtests im realen Straßenverkehr dienen ergänzend vor allem der Aufnahme von Messdaten, die in dem PEGASUS-Prozess zur Identifikation von kritischen Szenarien genutzt werden, sowie der Validierung. Die Ergebnisse der Testdurchführung werden in Form von Testdaten abgelegt und in einem Auswerteschritt mittels Bewertungskriterien, wie einer minimalen Distanz, bewertet. Wenn die Auswertung beispielsweise Kollisionen ergibt, wird über eine Rückführung der Testergebnisse zur Parametervariation im Parameterraum des logischen Szenarios nach weiteren kritischen Parametrisierungen gesucht. Im letzten Schritt der Risikoermittlung wird zur Einordnung der Testergebnisse untersucht, ob sich das Testobjekt an vorher definierte Verhaltensregeln, wie zum Beispiel das Einhalten eines angemessenen Sicherheitsabstands, gehalten hat. Die gesamten Testergebnisse bilden eine Sicherheitsaussage, die dann für eine Sicherheitsargumentation verwendet wird [154]. Dabei folgt die Sicherheitsargumentation von PEGASUS den fünf Paradigmen Struktur, Formalisierung, Kohärenz, Integrität und Relevanz und ist in fünf Ebenen unterteilt [155].

Das Projekt V&V-Methoden (VVM)² führt als Folgeprojekt von PEGASUS die Forschungsarbeiten im Bereich der Zulassung automatisierter Fahrzeuge fort. Der

² gefördert vom Bundesministerium für Wirtschaft und Energie (BMWi)

Schwerpunkt liegt auf der Weiterentwicklung des Testkonzepts und der Anwendung auf den urbanen Raum. Das szenariobasierte Testen bildet auch in VVM die Grundlage für die zu führende Sicherheitsargumentation.

2.3.3.4 Bewertungskriterien für Fahrszenarien

Für die Bewertung von Fahrszenarien werden in der Literatur verschiedene Kriterien und Kritikalitätsmetriken vorgeschlagen, die das allgemeine Fahrzeugverhalten innerhalb eines Szenarios oder einer Szene bewerten. Die Bewertungskriterien können in Abhängigkeit des Zeitraums den sie bewerten in nanoskopisch, mikroskopisch und makroskopisch unterteilt werden [175].

- **Nanoskopische** Bewertungskriterien werden für einen Zeitschritt bestimmt, zum Beispiel die Zeitlücke oder TTC
- **Mikroskopische** Bewertungskriterien werden für ein Zeitintervall bestimmt, zum Beispiel die minimale TTC oder PET
- **Makroskopische** Bewertungskriterien werden für einen Satz an Zeitintervallen bestimmt, eine Kollisionswahrscheinlichkeit

Neben dem räumlichen Abstand (oder Weglücke) (d_x) zwischen zwei Objekten, ist die Zeitlücke τ ein aus der Fahrsicherheit kommendes, oft verwendetes nanoskopisches Bewertungskriterium [91].

Definition 2.38 (Zeitlücke) „Zeitlicher Abstand zwischen dem Passieren eines Querschnitts durch aufeinanderfolgende Fahrzeuge.“ [85]

Für die Bestimmung der Zeitlücke werden Messpunkte entweder an der Vorder- oder Rückkante des Fahrzeugs definiert. Die Bruttozeitlücke (in Englisch Headway [9]) verwendet die beiden Vorderkanten und die Nettozeitlücke (in Englisch Gap Time [175]) die Hinterkante des vorausfahrenden Fahrzeugs und die Vorderkante des Ego-Fahrzeugs [85] (siehe Abbildung 2.20).

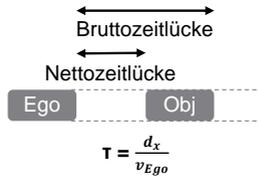


Abbildung 2.20: Darstellung der Definition der Zeitlücke nach [9]

Ein weiteres nanoskopische Bewertungskriterium ist die Time-to-Collision (TTC). Sie beschreibt im Gegensatz zur Zeitlücke eine konkrete Kollisionsgefahr.

Definition 2.39 (Time-to-Collision (TTC)) Beschreibt die Zeit bis zwei Objekte, unter Beibehaltung ihrer Geschwindigkeit, kollidieren.

Sie wird bestimmt durch den Quotienten des Abstands d_x und der Relativgeschwindigkeit. Die minimale Zeitlücke oder TTC in einem Szenario ist dagegen ein mikroskopisches Kriterium, da sie für das Zeitintervall der Szenariendauer bestimmt wird.

Ein weiteres makroskopisches Bewertungskriterium ist die Post Encroachment Time (PET) (siehe Abbildung 2.21).

Definition 2.40 (Post Encroachment Time) Beschreibt den zeitlichen Abstand zwischen zwei Objekten, wenn das erste Objekt einen Konfliktpunkt verlässt und das zweite Objekt in ihn eintritt [151]

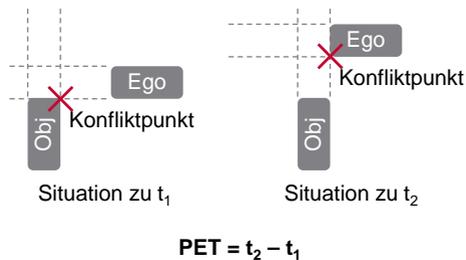


Abbildung 2.21: Darstellung der Definition der PET nach [204]

Die Kollisionsfreiheit ist ein wichtiges makroskopisches Bewertungskriterium. Sie kann durch die Auswertung der minimalen Distanz bestimmt werden.

2.4 Zulassung von Kraftfahrzeugen in Deutschland

Damit ein entwickeltes Fahrerassistenzsystem serienmäßig im Straßenverkehr eingesetzt werden kann, muss der Hersteller eine Typgenehmigung dafür beantragen. Dieser Genehmigungsprozess wird Homologation genannt [200].

Definition 2.41 (Typgenehmigung) *„Bestätigung der Typgenehmigungsbehörde, dass ein serienmäßig in größerer Stückzahl hergestellter Typ gleichartiger Fahrzeuge oder Fahrzeugteile den Vorschriften entspricht“ [123]*

Definition 2.42 (Homologation) *Die Homologation ist der Prozess der Zertifizierung von Fahrzeugen oder eines bestimmten Bauteils eines Fahrzeugs, dass es die von verschiedenen gesetzlichen Stellen festgelegten Anforderungen erfüllt. [200]*

In Deutschland wird die Fahrzeugtypgenehmigung der Europäischen Gemeinschaft (EG) durch das Kraftfahrt-Bundesamt (KBA) unter Berücksichtigung der Straßenverkehrs-Zulassungs-Ordnung (StVZO) erteilt (siehe Abbildung 2.22) [117]. Die Typgenehmigung für ein neues Fahrzeug setzt sowohl einen Nachweis der korrekten Funktionalität des Einzelprodukts als auch einen Nachweis für geeignete Produktionsverfahren voraus. Mit einer Anfangsbewertung durch das KBA wird ermittelt, ob die vom Hersteller verwendeten Verfahren und Prozesse eine genehmigungskonforme Produktion erwarten lassen. Dies geschieht entweder durch eine Begehung der Produktions- und Entwicklungseinrichtungen vor Ort oder durch die Vorlage eines geeigneten Zertifikats nach EN ISO 9001 oder einer anderen EU-Typengenehmigungsbehörde. Die eigentlichen technischen Überprüfungen des Produkts und des Qualitätsmanagement-System (QMS) übernimmt ein sogenannter Technischer Dienst (TD). Dies können beispielsweise der Tüv, die DEKRA oder andere zertifizierte Unternehmen sein [119].

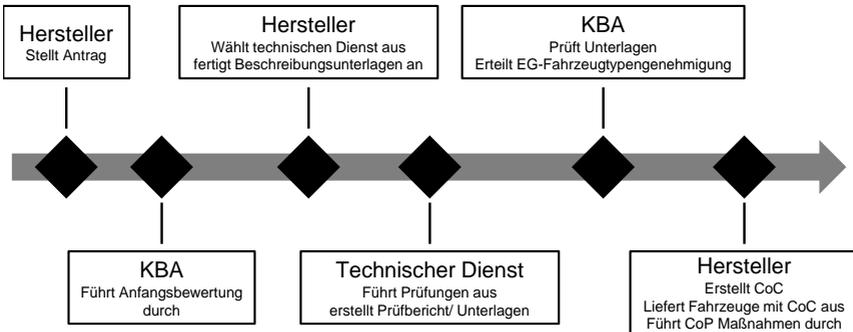


Abbildung 2.22: Ablaufplan zur Erteilung einer EG-Fahrzeugtypgenehmigung nach [117]

Definition 2.43 (Technischer Dienst) „Organisation oder Stelle, die von der Genehmigungsbehörde eines Mitgliedstaates als Konformitätsbewertungsstelle (Prüflabor oder Zertifizierungsstelle) benannt wurde“ [122]

In Prüflaboratorien werden Fahrzeuge sowie einzelne Fahrzeugteile, unter Verwendung der Normen DIN EN ISO/IEC 17025 und DIN EN ISO/IEC 17020, der sogenannten Typprüfung unterzogen und die Ergebnisse in einem Prüfbericht zusammengefasst. Parallel dazu wird durch Zertifizierungsstellen untersucht, inwieweit das QMS den genehmigungsrelevanten Anforderungen entspricht und im Fall einer positiven Überprüfung entsprechende Bescheinigungen ausgestellt. Auf Grundlage dieser Unterlagen erteilt das KBA die EG-Fahrzeugtypgenehmigung [119]. Die Fahrzeugtypgenehmigung ermächtigt den Hersteller für jedes Fahrzeug des zugelassenen Fahrzeugtyps eine Konformitätsbescheinigung (engl. Certificate of Conformity (CoC)) zu erstellen und die Fahrzeuge damit an Kunden auszuliefern. Zusätzlich müssen während der Produktion der Fahrzeuge Konformitätsprüfungen (engl. Conformity of Production (CoP)) durchgeführt werden [117].

Definition 2.44 (Fahrzeugtyp) Fahrzeuge, die sich in wesentlichen Merkmalen nicht unterscheiden, wie beispielsweise: Motortyp, Höchstgeschwindigkeit, Getriebe oder Achslastverteilung.

Definition 2.45 (Konformitätsbescheinigung) „Bescheinigung des Genehmigungsinhabers über die Übereinstimmung eines Fahrzeugs mit dem genehmigten Typ.“ [120]

Definition 2.46 (Konformitätsprüfungen) „Übereinstimmung der Produktion mit dem genehmigten Typ. Die Untersuchung beinhaltet die Bewertung getroffener Vorkehrungen und Prüfverfahren zur Sicherstellung der Produktkonformität sowie die Überprüfung der jeweiligen Produkte (Genehmigungsobjekte).“ [121]

Welche Merkmale den zugelassenen Fahrzeugtyp bestimmen und wie bei einer Änderung des Fahrzeugtyps vorgegangen werden muss, wird in den rechtlichen Regelungen beschrieben.

2.4.1 Typprüfung von Fahrzeugen

Die in Deutschland zur Anwendung kommende StVZO verweist im §19 „Erteilung und Wirksamkeit der Betriebserlaubnis“ auf die Richtlinie 2007/46/EG des Europäischen Parlaments und des Rates vom 5. September 2007 zur Schaffung eines Rahmens für die Genehmigung von Kraftfahrzeugen und Kraftfahrzeuganhängern [55]. Sowohl die StVZO als auch die europäische Richtlinie 2007/46/EG integrierte die United Nations Economic Commission for Europe (UNECE)-Regelungen von 1958, die von der Wirtschaftskommission für Europa der Vereinten Nationen beschlossen wurde (siehe Abbildung 2.23) [55] [203]. Die UNECE-Regelungen beinhalten international vereinbarte, einheitliche Bedingungen für die Genehmigung von Fahrzeugtypen oder Bauteilen. Aktuell beschreiben circa 160 verschiedene Regelungen wie einzelne Bauteile, Komponenten oder Assistenzsysteme zugelassen werden können (siehe Tabelle 2.3).



Abbildung 2.23: Abhängigkeiten und Verweise im Zulassungsrecht nach [55] [203]

Regelung	Inhalt der Regelung
Regelung 89	Fahrzeuge mit Geschwindigkeits-Begrenzungseinrichtungen und einstellbaren Geschwindigkeits-Begrenzungseinrichtungen
Regelung 130	Fahrzeuge mit Spurhaltewarnsystem
Regelung 131	Fahrzeuge mit Notbremsassistentensystem
Regelung 157	Fahrzeuge mit automatischen Spurhalteassistentensystem

Tabelle 2.3: Liste an Regelungen für die Zulassung von ausgewählten FAS nach [69]

In der UNECE-Regelung 157 wurden erstmals Vorschriften für die Zulassung einer automatisierten Fahrfunktion festgelegt, welche die Quer- und Längsbewegung des Fahrzeugs, ohne Eingreifen des Fahrzeugführers, über einen längeren Zeitraum hinweg übernimmt [70]. Als erstes hochautomatisiertes System der Automatisierungsstufe drei, wurde der „DRIVE PILOT“ von Mercedes Benz auf Grundlage der Regelung 157 zugelassen.

Inhalt der Regelungen sind neben der Definition von relevanten Begriffen, Informationen über die Genehmigung, geltende Vorschriften und notwendige Prüfverfahren [67]. Die Regelung 157 enthält darüber hinaus weitere Kapitel, die für einen automatisierten Betrieb relevant sind. Dies sind beispielsweise Kapitel über die Mensch-Maschine-Schnittstelle, einen notwendigen Fahrmodusspeicher oder Cybersicherheit und Softwareaktualisierung. Der Abschnitt „Prüfvorschriften für automatische Spurhalteassistentensysteme (ALKS)“ der UNECE-Regelung 157 beschreibt, wie das Prüfobjekt vom TD auf der Prüfstrecke oder dem Rollenprüfstand getestet werden muss.

Neben einem Katalog mit zu testenden Prüfscenarien und allgemeinen Parameterräumen, enthält die Regelung auch offene Formulierungen wie: *„Bei der Prüfung ist nachzuweisen, dass das ALKS einen Zusammenstoß mit einem stehenden Fahrzeug, Verkehrsteilnehmer oder einem Objekt, das die Fahrspur vollständig oder teilweise blockiert, bis zur angegebenen Höchstgeschwindigkeit des Systems verhindert“*. In welcher Form ein solcher Nachweis zu erbringen ist, wird nicht näher spezifiziert und muss vom Hersteller selbst erarbeitet werden. Zusätzlich fordert

die Regelung auch eine Prüfung unter Realbedingungen, bei der unter anderem auch die Reaktion des Fahrzeugs auf ein ungeplantes Ereignis überprüft werden soll [70].

2.4.2 Nutzung von Simulationen für die Zulassung von Fahrzeugen

Für einige Fahrzeugkomponenten oder Fahrerassistenzsysteme ist es nicht notwendig, dass alle Versionen oder Varianten desselben Fahrzeugtyps am realen Fahrzeug untersucht werden müssen [68]. In diesem Fall muss nur ein Fahrzeug physikalisch geprüft werden. Erfüllt es alle Voraussetzungen, können andere Varianten mittels Rechnersimulation nachgewiesen werden. Ist die Nutzung einer Simulation zur Freigabe möglich, wird dies in der jeweiligen UNECE-Regelung explizit angegeben. Die Anlagen 1 und 2 der UNECE-Regelung 13-H für die Freigabe von Bremssystemen beschreiben beispielsweise für welche Zwecke die Simulation eingesetzt werden kann, welche Faktoren in der Simulation berücksichtigt werden müssen und wie das Simulationsinstrument validiert werden muss [68]. Um die Validität der verwendeten Simulationsinstrumente nachzuweisen, sind Vergleiche mit praktischen Fahrzeugprüfungen für die notwendigen Fahrmanöver durchzuführen.

Definition 2.47 (Fahrmanöver) *Abstrakte Definition des Verhaltens eines Fahrzeugs [15]*

Die UNECE-Regelung beinhaltet Informationen darüber anhand welcher Kriterien der Vergleich durchzuführen ist. Detaillierte Vorgaben, ab wann eine Vergleichbarkeit tatsächlich vorliegt, werden allerdings nicht gegeben. Auch die Regelung 157 erlaubt die Verwendung von Simulationswerkzeugen für Prüfscenarien, die auf einem Prüfstand oder unter realen Fahrbedingungen nur schwer umsetzbar sind. Dafür muss eine Validierung des Simulationswerkzeugs durchgeführt werden, die die Korrelation des Simulationsergebnisses mit physikalischen Prüfungen nachweist, und die Eignung des Simulationswerkzeugs für das dargestellte Szenario muss dargelegt werden [70].

3 Diskussion des Stands der Wissenschaft und Anforderungen an ein szenariobasiertes Testkonzept

3.1 Verwandte Arbeiten

3.1.1 Bestimmung des Reifegrads

Ein Ansatz zur Reifegradbestimmung eines Fahrerassistenzsystems in späteren Entwicklungsphasen ist der Einsatz von statistischer Validierung [206]. Ein distanzbasierter Ansatz berechnet dabei aus der gesamten Fahrleistung in einem Land und der Anzahl an tödlichen Unfällen die durchschnittliche Distanz zwischen zwei tödlichen Unfällen. Aus dieser Distanz lassen sich Wahrscheinlichkeiten für Fehler ableiten. Bezogen auf die gesamte Fahrleistung in Deutschland, ist die Wahrscheinlichkeit für einen tödlichen Unfall, der von einem menschlichen Fahrer verursacht wurde, relativ gering. Statistisch müssten daher 6.62 Milliarden Testkilometer gefahren werden, um nachzuweisen, dass das ADS mindestens so sicher ist wie ein menschlicher Fahrer. Dieser Aufwand ist praktisch nicht leistbar, zumal er bei jeder Änderung des Systems erneut durchgeführt werden müsste. Daher reduzieren „disengagement reports“ die notwendige Kilometeranzahl, indem sie statistisch häufigere Ereignisse als einen tödlichen Unfall verwenden. Hierbei werden Situationen betrachtet, in denen ein menschlicher Sicherheitsfahrer in den Betrieb des Systems eingreifen muss [149, 182]. Dies geschieht entweder

aufgrund eines technischen Versagens oder in Situationen, in denen das System ohne Eingriff eine unsichere Aktion durchführt und möglicherweise einen Unfall verursacht hätte. In beiden Ansätzen liefern die gefahrenen Kilometer allerdings keine Informationen über den Inhalt der Erprobungsfahrten.

Amersbach und Winner schlagen eine Methode zur Bestimmung der benötigten und machbaren Testabdeckung für eine szenariobasierte Validierung von ADS vor [10]. Ihr Ansatz basiert auf der Übertragung des distanzbasierten Ansatz der statistischen Validierung auf das szenariobasierte Testen. Die zu fahrende Distanz wird als Aneinanderreihung von konkreten Szenarien betrachtet und anstelle einer Distanz bestimmen sie die Anzahl zu testender konkreter Szenarien n_{req} . Als Grundlage dafür dient die zurückgelegte Distanz zwischen zwei tödlichen Unfällen \bar{s}_{Ref} . Diese Distanz wird in eine äquivalente Anzahl an konkreten Szenarien umgewandelt, indem sie die durchschnittliche Distanz eines konkreten Szenarios \bar{s}_{sc} aus dessen durchschnittlicher Dauer \bar{t}_{sc} und Geschwindigkeit \bar{v}_{sc} berechnen. Zusätzlich werden zwei Faktoren eingeführt, mit denen die zeitliche Überlappung von konkreten Szenarien, sowie deren Einzigartigkeit berücksichtigt wird. Basierend auf der Annahme, dass für die Validierung eines ADS ein Jahr Zeit zur Verfügung steht, wird eine leistbare Anzahl an konkreten Szenarien n_f berechnet. Mit den selbst gewählten Annahmen kommen die Autoren dennoch zum Schluss, dass die notwendige Anzahl an konkreten Szenarien ($n_{req} = 6 * 10^9$) die leistbare Anzahl an konkreten Szenarien ($n_{req} = 4 * 10^9$) übersteigt.

In den Forschungsprojekten PEGASUS und VV-Methoden wird der Reifegrad eines ADS anhand des verbleibenden Restrisikos nach der Entwicklung bestimmt [154, 173]. Dafür wird im Rahmen vom PEGASUS-Projekt evaluiert, in welchem Rahmen das ADS bestimmte Verhaltensanforderungen erfüllt. Beispielhafte Verhaltensanforderungen sind das Einhalten eines angemessenen Sicherheitsabstands, selbst keine Kollision zu verursachen und wenn möglich Kollisionen zu verhindern [154]. In VV-Methoden hingegen sollen Sicherheitsziele wie zum Beispiel eine positive Risikobilanz nachgewiesen werden. Eine positive Risikobilanz bedeutet, dass beispielsweise die Auftretenswahrscheinlichkeit für eine Kollision, die durch das ADS verursacht wurde, geringer ist als ein statistischer Vergleichswert von menschlichen Fahrern in einem Referenzzeitraum [173].

Klamann et al. stellen eine Methode vor mit der sie Sicherheitsziele auf pass-/fail-Kriterien herunterbrechen können und die eine Bewertung der Sicherheit von ADS ermöglicht [110]. Dafür wird das Sicherheitsziel im ersten Schritt mit Methoden wie der Fehlerbaumanalyse oder der systemtheoretischen Prozessanalyse analysiert und kausale Faktoren identifiziert, die zu der Verletzung des Sicherheitsziels führen. Aus diesen kausalen Faktoren werden Sicherheitsanforderungen an das ADS abgeleitet, welche die Faktoren verhindern sollen. Im letzten Schritt werden dann pass-/fail-Kriterien aus den Sicherheitsanforderungen abgeleitet.

3.1.2 Methoden zur Bewertung von Fahrerassistenzsystemen

Roesener et al. schlagen vor Fahrerassistenzsysteme in Fahrszenarien dahingehend zu bewerten, ob sie von menschlichem Verhalten abweichen [166]. Aus Referenzdaten von menschlichen Fahrern werden Verteilungen für Bewertungsgrößen für jedes zu testende logische Szenario erzeugt. Als Bewertungsgrößen geben Roesener et al. beispielsweise Trajektorien beim Spurwechsel, die Längsbeschleunigung oder die Zeitlücke (vgl. Definition 2.38) an. Mit dem zu testenden System wird anschließend ein Testdatensatz für jedes logische Szenario erzeugt. Die Bewertung des Systems erfolgt dann durch den Vergleich des Referenzdatensatzes mit dem Testdatensatz.

Eine weitere Möglichkeit zur Bewertung von Fahrerassistenzsystemen ist die Verwendung von Kritikalitätsmetriken (vgl. Abschnitt 2.3.3.4) [9, 91, 143, 151, 175]. Anhand der Metriken, wie beispielsweise einer TTC oder minimalem Abstand, wird der Ausgang eines Szenarios bewertet und daraus Aussagen über das Fahrerassistenzsystem abgeleitet. Mahmud et al. haben eine Literaturübersicht der verschiedenen Kritikalitätsmetriken erstellt und diese in vier Kategorien unterteilt [134]. Sie unterscheiden zwischen zeitlichen, räumlichen, verzögerungs-basierten und sonstigen Indikatoren

Bach verwendet in seiner Dissertation statische Gates zur Bewertung eines prädiktiven Abstandsregeltempomaten (vgl. Abschnitt 2.3.3) in simulierten Fahrscenarien [13]. Die Gates stellen dabei eine positionsbezogene Bewertung des FAS dar. Es wird an einer bestimmten Position im Szenario überprüft, ob die Geschwindigkeit des Fahrzeugs im Toleranzbereich um eine Sollgeschwindigkeit liegt. Wenn der spezifizizierte Bereich eingehalten wird, dann wird das Gate als bestanden gewertet. Damit das Fahrscenario als bestanden gewertet wird, müssen alle Gates bestanden werden.

Das Konzept des Shadow-Modes sieht eine Ausführung des Testobjekts während Erprobungsfahrten in einer Open-Loop Schleife vor [107, 208]. Dabei erhält das Testobjekt Eingabedaten von realen Sensoren, aber die Ausgabe des Systems wird nicht an die nachfolgenden Elemente in der Wirkungskette, wie beispielsweise die Lenkung oder Bremse, weitergegeben. Wang nutzt dieses Konzept um ein vollständiges Fahrerassistenzsystem zu validieren [208], während Karpathy es für die Validierung eines neuronalen Netzes zur Prädiktion von Spurwechseln verwendet [107]. Ein wesentlicher Vorteil dieses Ansatzes ist, dass durch den Open-Loop Betrieb keine zusätzlichen Risiken entstehen und das Testobjekt in realen Szenarien im Fahrzeug getestet wird. Darüber hinaus kann die Ausgabe des Testobjekts mit dem Verhalten des Fahrers verglichen werden. Der Vergleich kann dann genutzt werden um Szenarien zu finden, in denen das System anders reagiert hätte als der menschliche Fahrer.

Im Bereich der Software-Entwicklung wird ebenfalls an der automatisierten Erstellung und Durchführung von Tests geforscht. Der Testablauf kann beispielsweise mit modellbasierten Verfahren erstellt werden. Die erwartete Ausgabe des Systems wird mit Hilfe eines Testorakels erzeugt, welches zwischen dem korrektem und nicht korrektem Verhalten eines Testobjekts unterscheiden kann [18].

Definition 3.1 (Testorakel) *Programm oder Verfahren, das das erwartete, spezifikationskonforme Verhalten eines Testobjekts für konkrete Testdaten ermittelt. Testorakel ermitteln das Sollergebnis, das nach der Testdurchführung mit dem Istergebnis verglichen wird. [174]*

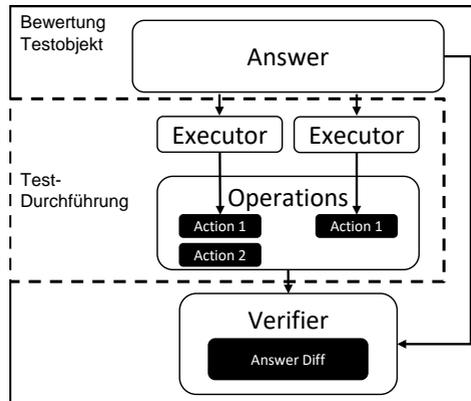


Abbildung 3.1: Darstellung des deklarativen Testens nach [198]

Die Erstellung von Testorakeln ist Bestandteil verschiedener wissenschaftlicher Arbeiten und wurde bereits in mehreren Übersichtsartikeln zusammengefasst [18, 178]. Voraussetzung für die Erstellung eines Testorakels ist die formale Spezifikation des zu testenden Systems [18, 132]. Im automobilen Umfeld gibt es ebenfalls Veröffentlichungen die sich mit der modellbasierten Erstellung von Testfällen beschäftigen [38, 48, 57, 65, 77, 128, 180]. Die automatische Generierung des erwarteten Systemverhaltens für die erzeugten Testfälle mittels Testorakel wird allerdings in keiner dem Autor bekannten Veröffentlichung adressiert.

Das deklarative Testen ist ein weiterer Testansatz aus der Software-Entwicklung, welcher ursprünglich von Microsoft für das Testen von Benutzeroberflächen entwickelt wurde. Benutzeroberflächen bieten die Möglichkeit, dass Systemfunktionen, wie beispielsweise das Einfügen eines Objekts in ein Dokument, über verschiedene Aktionen in der Benutzeroberfläche durchgeführt werden können. Während des Testens müssen dann mehrere Testfälle definiert werden, die sich nur in wenigen Testschritten unterscheiden. Um den Aufwand bei der Testerstellung zu reduzieren, wird beim deklarativen Testen die Systembewertung von der Testdurchführung getrennt (siehe Abbildung 3.1) [198]. Die *Answer* beschreibt das erwartete Systemverhalten, wie zum Beispiel das ein neues Objekt nach dem

Einfügen existiert. Während der Testdurchführung werden durch *Executoren* verschiedene Bediensequenzen zum Einfügen eines Objektes durchgeführt. Anschließend wird in dem *Verifier* überprüft, ob das vorliegende Systemverhalten mit dem erwarteten Systemverhalten übereinstimmt. Die Erstellung von Bediensequenzen ist weniger aufwändig und leichter automatisierbar als die Definition von monolithischen Testfällen. Der Ansatz des deklarativen Testens wurde von Gustafsson erstmals in einem automobilen Kontext eingesetzt, um bei der Ausführung von Tests auf Komponentenebene verschiedene Aspekte einer Lichtsteuerung parallel auszuwerten [82].

3.1.3 Ableitung von Testfällen

Die Ableitung von konkreten Szenarien aus logischen Szenarien mittels Parametervariation ist Inhalt zahlreicher wissenschaftlicher Arbeiten [44, 51, 109, 112, 140]. Üblicherweise ist das Ziel der Parametervariation möglichst sicherheitskritische Szenarien zu generieren, in denen beispielsweise das Risiko für eine Kollision mit anderen Verkehrsteilnehmern besteht [112]. Dabei werden unterschiedliche Methoden zur Parametervariation verwendet, wie zum Beispiel die vollfaktorielle Kombination aller Szenarioparameter [158], Design-of-Experience [22], Methoden der Extremwerttheorie [184] oder evolutionäre Algorithmen [111].

Bei der Generierung von sicherheitskritischen Szenarien werden allerdings auch konkrete Szenarien erzeugt, in denen eine Kollision unvermeidbar ist, weil der Fußgänger beispielsweise direkt vor dem Fahrzeug die Straße betritt. Calo et al. versuchen deshalb in ihrer Arbeit nur konkrete Szenarien zu erzeugen, bei denen die Kollision durch ein ADS auch verhindert werden kann [44]. Dafür wird untersucht, ob Kollisionen in konkreten Szenarien mit einer anderen Parametrisierung des ADS hätte vermieden werden können. Dies kann in einem zweistufigen Verfahren realisiert werden, indem zuerst die Suche nach Kollisionen im Parameterraum stattfindet und im Anschluss nach alternativen Parametrisierungen gesucht wird. Zusätzlich stellen Calo et al. auch eine kombinierte Suche vor, mit

der direkt konkrete Szenarien gefunden werden, die eine vermeidbare Kollision beinhalten.

Bussler et al. stellen einen Ansatz zur Ableitung von Testfällen aus logischen Szenarien vor [43]. Der vorgeschlagene Ansatz leitet zunächst in fünf Schritten konkrete Szenarien aus einem logischen Szenario ab und verknüpft diese anschließend mit Testanforderungen, die Informationen über das Testobjekt, Testziel und Metriken zur Auswertung enthalten. Im ersten Schritt der Ableitung werden zunächst alle möglichen Parameterkombinationen für das betrachtete logische Szenario erstellt. Diese werden dann im zweiten Schritt anhand von physikalischen Grenzen, der ODD-Definition oder Systemgrenzen des Testobjekts gefiltert um so einen reduzierten Satz an konkreten Szenarien zu erhalten. Der reduzierte Satz an konkreten Szenarien wird anschließend anhand von verschiedenen risikobasierten KPI's geclustert um ein besseres Verständnis des Parameterraums zu erhalten. Im vierten Schritt wird dann eine Auswahlstrategie für die erzeugten Cluster erstellt, mit der eine effiziente Abdeckung des Parameterraums erreicht werden soll. Der fünfte Schritt umfasst die iterative Ausführung der Szenarien um einen repräsentativen Satz an konkreten Szenarien zu erhalten.

Daten, die während der Erprobungsfahrten aufgezeichnet werden (vgl. Abschnitt 2.3.2.3), können auch direkt für Tests verwendet werden. Dafür werden die aufgezeichneten Daten in einer Testumgebung wieder abgespielt und liefern so notwendige Eingabeinformationen für das Testobjekt [14, 107, 192, 208]. Für Systeme die keine Rückkopplungsschleife besitzen, kann die erwartete Ausgabe des Systems im voraus definiert werden. Dies ermöglicht eine automatische Bewertung des Systems und eine Wiederholung für nachfolgende Entwicklungsstufen. Während dieser Ansatz auf Systeme mit offenen Regelschleifen begrenzt ist, ermöglicht ein anderer Ansatz zusätzlich die Verifizierung und Validierung eines longitudinalen Regelungssystems [13]. Dies wird ermöglicht indem aufgezeichnete zeitbasierte Daten auf die gefahrene Strecke abgebildet werden. Da der Umfang der aufgezeichneten Daten begrenzt ist, fokussieren sich einige Forschungsarbeiten auf die Modifizierung oder Augmentierung von aufgezeichneten Messdaten. Während der Re-Simulation der Daten werden virtuelle Objekte wie Fußgänger [145, 146] oder Fahrzeuge [216] eingefügt.

3.2 Diskussion des Stands der Wissenschaft und Technik

Im Produktentwicklungsprozess beschreibt der Reifegrad die objektive Beurteilung eines zu entwickelnden Produkts, die durch die Erfüllung verschiedener Messkriterien charakterisiert bestimmt (vgl. Definition 2.7). Ein zentrales Messkriterium während der Produktentwicklung ist, dass die Funktionalität des Produkts nachgewiesen wurde (vgl. Abschnitt 2.2). Dafür wird derzeit überprüft, ob die Systemanforderungen in einem beispielhaften konkreten Szenario durch den aktuellen Entwicklungsstand des Produkts erfüllt werden. Die vorgestellten Arbeiten im Abschnitt 3.1.1 beschäftigten sich ausschließlich mit der Validierung von ADS und der Erbringung eines Nachweises für die Sicherheit eines ADS. Eine entwicklungsbegleitende Bestimmung des Reifegrads des Systems, bei der für die gesamte ODD überprüft wird, ob das System die gestellten Anforderungen erfüllt, ist mit keinem vorgestellten Ansatz möglich.

Um den Reifegrad in logischen Szenarien bestimmen zu können, muss die gesamte Funktionalität von Fahrerassistenzsystemen in den Szenarien bewertet werden können. Die logischen Szenarien repräsentieren dabei laut Definition zunächst nur die abstrakte Beschreibung von Verhalten der verschiedenen Verkehrsteilnehmer (vgl. Definition 2.33). Für das szenariobasierte Testen müssen sie daher in ausführbare Tests überführt werden. Dies beinhaltet sowohl die Ableitung von konkreten Szenarien aus logischen Szenarien durch Parametervariation als auch die Definition von Bewertungskriterien (vgl. Abschnitt 2.3.3.3).

Roesener et al. schlagen vor, dass die Bewertung des Fahrerassistenzsystems anhand eines Vergleichs mit menschlichen Referenzdaten erfolgt (vgl. Abschnitt 3.1.2). Damit lässt sich folglich auch nur Systemverhalten bewerten, das anhand des Gesamtfahrzeugverhaltens beobachtbar ist und für das eine menschliche Referenz erzeugt werden kann. Beispiele dafür wären das Einhalten eines minimalen Abstands oder einer Sollgeschwindigkeit. Die dafür notwendigen Referenzdaten müssen allerdings für jedes logische Szenario in realen Fahrversuchen erzeugt werden. Während der Systementwicklung muss allerdings auch Systemverhalten

bewertet werden, das nicht am Gesamtfahrzeugverhalten beobachtbar ist oder für das keine menschliche Referenz existiert. Beispiele dafür sind die Anforderung einer Warnung oder die Bestimmung der Differenzgeschwindigkeit zu einem vorausfahrendem Fahrzeug. Aufgrund des zu erwartenden Aufwands für die Erstellung der Referenzdaten und der eingeschränkten Anwendbarkeit, ist der Ansatz von Roesener et al. nicht für eine entwicklungsbegleitende Bewertung geeignet.

Die aktuell verwendeten Szenario- oder Szenenbewertungskriterien, wie beispielsweise die Kollisionsfreiheit, die TTC oder der minimale Abstand, ermöglichen nur die Bewertung eines allgemeinen Fahrzeugverhaltens, wie beispielsweise das Einhalten von räumlichen oder zeitlichen Abständen (vgl. Abschnitt 2.3.3.4). Die detaillierte Betrachtung einzelner (Teil-)Funktionalitäten, die nicht auf einzuhaltende Abstände abzubilden sind, wie beispielsweise die Ausgabe einer Warnung oder die Bestimmung einer Differenzgeschwindigkeit, ist damit ebenfalls nicht möglich (siehe Tabelle 3.1).

Wird der in Abbildung 2.10 vorgestellte Testfall in einen szenariobasierten Testfall überführt, dann besteht er aus einem Annäherungsszenario an eine Kreuzung mit einem kreuzendem Fußgänger und dem Bewertungskriterium, dass der Abstand zwischen Fahrzeug und Fußgänger größer als 0 m ist. Die verbleibenden systemspezifischen Signalüberprüfungen des Testfalls, wie die Überprüfung der Ausgabe einer Warnung oder das eine bestimmte Verzögerung angefordert wird, werden nicht abgebildet. Um das szenariobasierte Testen bereits in frühen Entwicklungsphasen effektiv einsetzen zu können, muss die Bewertung von Systemverhalten des

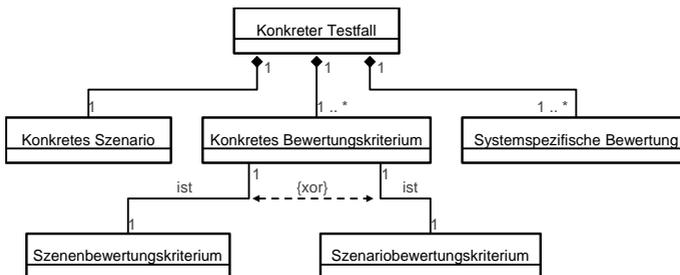


Abbildung 3.2: Erweiterung der Testfalldefinition nach [190] um Systembewertungen

Funktionalität des Fahrerassistenzsystem	Kritikalitätsmetrik			
	Minimaler Abstand	TTC	PET	Kollision
Adaptiver Abstandsregeltempomat				
Geschwindigkeitsregelung	✗	✗	✗	✗
Abstandsregelung	✗	✗	✓	✗
Anzeige im Kombi-Instrument	✗	✗	✗	✗
Übernahme von Geschwindigkeitslimits	✗	✗	✗	✗
Aktiver Bremsassistent				
Warnung des Fahrers	✗	✗	✗	✗
Notbremsung	✓	✓	✗	✓
Aktiver Lenkassistent				
Halten der Fahrspur	✗	✗	✗	✗
Spurwechselunterstützung	✗	✗	✗	✗
Verkehrszeichenassistent				
Erkennung Geschwindigkeitslimits	✗	✗	✗	✗
Totwinkelassistent				
Anzeigen von Fahrzeugen im toten Winkel	✗	✗	✗	✗
Verhindern von Seitkollisionen	✓	✓	✗	✓

Tabelle 3.1: Gegenüberstellung welches Funktionsverhalten der Fahrerassistenzsysteme aus Abbildung 2.1 mit gängigen Kritikalitätsmetriken bewertet werden kann.

Fahrerassistenzsystems in den Szenarien möglich sein. Daher wird im Rahmen der Dissertation die Testfalldefinition nach [190] um systemspezifische Bewertungen (vgl. Definition 2.23) erweitert (siehe Abbildung 3.2).

Um diese systemspezifischen Bewertungen bei der Ableitung von Testfällen erstellen zu können, würde ein Testorakel benötigt, was in Abhängigkeit der Parametrisierung des konkreten Szenarios sowohl den Zeitpunkt einer erwarteten Systemreaktion bestimmt als auch das Systemverhalten vorhersagt. Da ein solches Testorakel noch nicht entwickelt wurde, stellt das deklarative Testen einen alternativen Ansatz dar, mit dem die systemspezifischen Bewertungen umgesetzt werden können. Es lassen sich Parallelen zwischen den beschriebenen Herausforderungen beim Testen von Benutzeroberflächen und dem szenariobasierten Testen erkennen. Die Problemstellung ist insoweit übertragbar, dass das gleiche Systemverhalten, zum Beispiel die Anforderung einer Notbremsung, in unterschiedlichen Fahrscenarien getestet werden soll.

Auch die Ableitung von konkreten Szenarien aus logischen Szenarien erfolgt mehrheitlich mit dem Ziel besonders sicherheitskritische Fahrscenarien zu erzeugen. Dabei wird mit der Annahme gearbeitet, dass jede Parameterkombination aus dem Parameterraum eines logischen Szenarios zu einem konkreten Szenario mit dem gleichen Inhalt führt. Folglich muss jedes konkrete Szenario die gleichen Aktionen und Events besitzen wie das zugrundeliegende logische Szenario (vgl. Definition 2.33). Aus dieser Annahme folgt, dass der Parameterraum von logischen Szenarien in seinen Grenzen als kontinuierlich ist. Ob diese zugrundeliegende Annahme allerdings zutrifft, wurde in keiner dem Autor bekannten Arbeit überprüft. Daher wird eine Voruntersuchung durchgeführt, bei der die Gültigkeit dieser Annahme überprüft wird.

3.3 Voruntersuchung zum Einfluss der Parametervariation auf den Inhalt von konkreten Szenarien

3.3.1 Versuchsaufbau und Durchführung

Durch die Analyse und Auswertung von Verkehrs- und Unfalldaten werden logische Szenarien für die Untersuchung ermittelt. Dafür werden der Unfallatlas vom Statistischen Bundesamt und die German In-Depth Accident Study (GIDAS)-Studie als Datenquelle betrachtet. Der Unfallatlas ist eine öffentlich zugängliche Datenbank, in dem alle Unfälle der Jahre 2016 bis 2020 mit Personenschaden sowie der Ort und die Unfallkategorie aufgelistet sind [139]. Im Gegensatz dazu ist die GIDAS-Studie eine Studie im Bereich der Verkehrsunfallforschung der Bundesanstalt für Straßenwesen und der Forschungsvereinigung Automobiltechnik e.V. Sie dokumentiert detailliert Verkehrsunfälle an zwei Erhebungsstandorten in Hannover und Dresden. Pro Unfall werden bis zu 3500 Einzelinformationen gesammelt und daraus Skizzen des Unfallhergangs erstellt [72]. Da der Unfallatlas keine Informationen über den Unfallhergang bereitstellt, können daraus keine logisches Szenario abgeleitet werden. Für die Auswahl der Szenarien werden daher die Daten aus der Gidas-Studie betrachtet. Die Unfalldaten der GIDAS-Studie wurden im Rahmen des internationalen Forschungsprojekts PROSPECT¹ bereits analysiert und die häufigsten Unfallszenarien zwischen PKW und Fußgängern ermittelt [195].

Für die Untersuchung werden vier häufige Unfallfallszenarien in logische Szenarien überführt und in einer Gesamtfahrzeugsimulationssoftware nachgebaut² (siehe Tabelle 3.2). Laut Definition werden Szenarien durch Events und Aktionen charakterisiert (vgl. Definition 2.33). Alle vier logischen Szenarien besitzen das

¹ <https://cordis.europa.eu/project/id/634149>

² Simulationssoftware: dSPACE VEOS (https://www.dspace.com/en/inc/home/products/sw/simulation_software/veos.cfm) zusammen mit dSPACE ASM ModelDesk (https://www.dspace.com/en/inc/home/products/sw/automotive_simulation_models/produkte_asm/modeldesk.cfm)

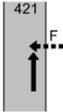
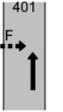
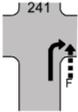
ID	Unfallszenario	Beschreibung
1		Fahrzeug folgt einer geraden Straße Fußgänger überquert die Straße von der Fahrzeugseite aus
2		Fahrzeug folgt einer geraden Straße Fußgänger überquert die Straße von der Gegenseite aus
3		Fahrzeug biegt an einer Kreuzung links ab Fußgänger überquert die Straße von der Fahrzeugseite aus
4		Fahrzeug biegt an einer Kreuzung rechts ab Fußgänger überquert die Straße von der Fahrzeugseite aus

Tabelle 3.2: Übersicht und Beschreibung der betrachteten Unfallszenarien

Event, dass ein Fußgänger vor dem Fahrzeug die Straße überquert und die Aktion, dass das Fahrzeug eine Notbremsung als Reaktion auf den kreuzenden Fußgänger durchführt. Als Szenarioparameter für die logischen Szenarien werden häufige Einflussfaktoren auf Unfälle ausgewählt.

Die Autoren der Studie geben als Einflussfaktoren der Unfälle die Geschwindigkeit des Fahrzeugs und des Fußgängers, sowie das Alter des Fußgängers, die Tageszeit und eine mögliche Verdeckung des Fußgängers an. Für die Analyse werden exemplarisch die Geschwindigkeit des Fahrzeugs v_{Fzg} und die Geschwindigkeit des Fußgängers v_{Ped} als zu variiere Parameter ausgewählt.

Die Geschwindigkeit des Fahrzeugs wird auf Basis der GIDAS Analysen von [195] festgelegt. Die Daten der Studie zeigen, dass über 80 % der Unfälle im Geschwindigkeitsbereich zwischen 10 und 40 km/h geschehen (siehe Abbildung 3.3). Daher wird Geschwindigkeit des Ego-Fahrzeugs in diesem Bereich mit einer

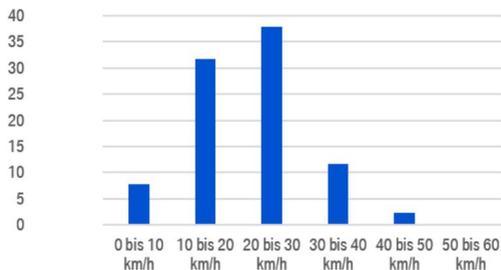


Abbildung 3.3: Initiale Geschwindigkeit des Fahrzeugs bei Unfällen mit Fußgängern nach [195]

exemplarischen Schrittweite von 2 km/h variiert um den gesamten Parameterraum abzudecken.

Die Geschwindigkeit des Fußgängers ist abhängig vom Alter, Geschlecht und der Bewegungsform des Fußgängers [19]. Zur Bestimmung der Geschwindigkeit werden die Bewegungsformen *Gehen*, *Schnell Gehen* und *Laufen* betrachtet. Da das Alter einen signifikanten Einfluss auf die Bewegungsgeschwindigkeit besitzt, wird zusätzlich das Alter von Fußgängern bei Unfällen aus dem GIDAS-Datensatz für die Bestimmung der Minimal- und Maximalwerte berücksichtigt. Laut einer Studie zur Mobilität und Sicherheit von Fußgängern lesen 35 % der Fußgänger Nachrichten auf ihren mobilen Endgeräten oder sehen sich Bilder/Videos an [127]. Darüber hinaus nutzen 45 % der Fußgänger ihre Geräte auch beim Überqueren der Straße. Aus diesem Grund werden zusätzlich zu den Werten aus [19] auch niedrigere Fußgängergeschwindigkeiten von 0.5 m/s und 1 m/s betrachtet. (siehe Tabelle 3.3)

Analysen aus der Unfallforschung zeigen, dass Frontkollisionen die häufigste Ursache für Unfälle zwischen Fahrzeugen und Fußgängern mit Todesfolge oder schwerer Verletzung sind [195]. Um solche Unfälle zu verhindern oder die Unfallschwere zu verringern, werden automatisierte Notbremssysteme eingesetzt. In Abhängigkeit einer Time-to-Collision (vgl. Abschnitt 2.3.3.4) warnt ein Notbremssystem zunächst den Fahrer mittels Frontkollisionswarnung (in Englisch Front Collision Warning (FCW)) oder vollführt eine automatisierte Notbremsung wenn der Fahrer nicht reagiert (in Englisch Autonomous Emergency Brake

Szenario	Geschwindigkeit Fahrzeug [km/h]			Geschwindigkeit Fußgänger [m/s]			Anzahl konkreter Szenarien
	Min	Max	Schrittweite	Min	Max	Schrittweite	
	10	40	2	0.5	4	0.5	

Tabelle 3.3: Parameterbereiche und Schrittweiten für die durchgeführten Szenarien

(AEB)). Die TTC, bei der das System reagiert, ist dabei abhängig von der eigenen Fahrzeuggeschwindigkeit.

Ein solches Notbremssystem wird auch zukünftig ein integraler Bestandteil jeder hochautomatisierten Fahrfunktion sein und daher als exemplarisches Fahrerassistenzsystem für diese Untersuchung ausgewählt. Da sich die Funktionsweise eines solchen Notbremssystems voraussichtlich nicht grundlegend ändern wird, lassen sich die damit erzielten Ergebnisse auch auf zukünftige Systeme übertragen.

3.3.2 Ergebnisse der Versuchsdurchführung

Im Rahmen der Untersuchung wird überprüft, ob die simulierten Szenarien das Event eines kreuzenden Fußgängers vor dem Fahrzeug beinhalten. Entspricht ein konkretes Szenario inhaltlich nicht dem gewünschten logischen Szenario, dann wird nachfolgend von einem „nicht validen Szenario“ gesprochen. Dafür wird die zeitliche und räumliche Position der Objekte am Kreuzungspunkt der Trajektorien des Fahrzeugs und des Fußgängers untersucht. Kreuzen sich die Trajektorien von Fahrzeug und Fußgänger räumlich in einem Punkt, dann wird anhand des Vorzeichens der Post-Enchroachment-Time entschieden, ob der Fußgänger die Straße vor dem Fahrzeug überquert hat (vgl. Definition 2.40). Bei einem negativem Vorzeichen der PET hat der Fußgänger die Straße vor dem Fahrzeug überquert (siehe Algorithmus 1).

Algorithmus 1 : Pseudocode für die Überprüfung eines kreuzenden Fußgängers

Result : Fußgänger kreuzt Fahrzeug

```
for (xVeh,yVeh) in Fahrzeugtrajektorie do
  for (xPed,yPed) in Fußgängertrajektorie do
    if xVeh == xPed && yVeh = yPed then
      tVeh = ZeitstempelAusTrajektorie(xVeh,yVeh);
      tPed = ZeitstempelAusTrajektorie(xPed,yPed);
      PET = tPed - tVeh;
      if PET < 0 then
        return Wahr;
      else
        return Falsch;
      end
    end
  end
end
end
```

Die Ergebnisse der Untersuchung sind in Abbildung 3.4 dargestellt. In dem betrachteten Parameterbereich des zweiten logischen Szenarios, besitzen lediglich 35.7 % der simulierten Szenarien einen kreuzenden Fußgänger vor dem Fahrzeug (siehe Tabelle 3.4). Damit verbunden kann in den restlichen Szenarien auch nicht die Aktion der Notbremsung durchgeführt werden, die zusätzlich für das logische Szenario definiert war. Für die anderen logischen Szenarien liegt der Anteil von validen konkreten Szenarien bei circa 80 %.

Die Zahlen in Tabelle 3.4 zeigen darüber hinaus, dass das betrachtete Notbremsensystem nur in einer verhältnismäßig kleinen Anzahl an konkreten Szenarien aktiv wurde. Im ersten logischen Szenario werden mit 43 von 112 (38.4 %) konkreten Szenarien mit aktiver Warnung die meisten Warnungen ausgegeben. Das logische Szenario mit den wenigsten Warnungen ist das dritte logische Szenario, dort werden lediglich in 6 von 112 (5.4 %) konkreten Szenarien eine Warnung ausgelöst. Notbremsungen werden maximal in 37 von 112 (33.0 %) beziehungsweise

3.3 Voruntersuchung zum Einfluss der Parametervariation auf den Inhalt von konkreten Szenarien

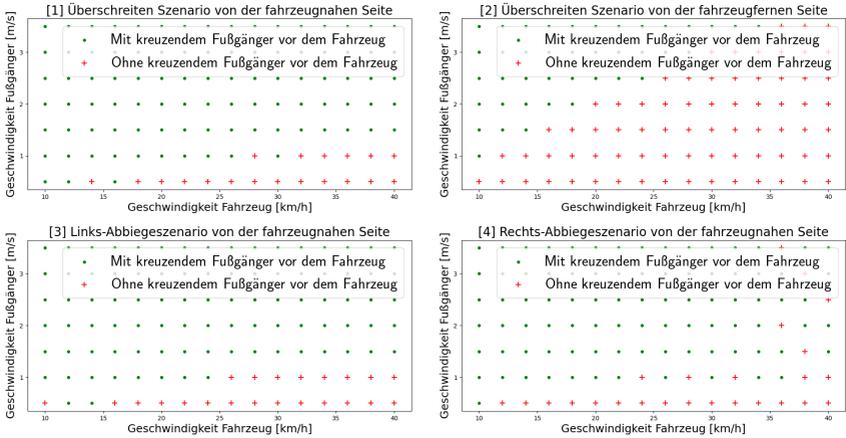


Abbildung 3.4: Visualisierung der validen und nicht validen Parameterkombinationen für der simulierten Szenarien

minimal in 4 von 112 (3.33 %) konkreten Szenarien durchgeführt. Der Grund, warum trotz eines kreuzenden Fußgängers in vergleichsweise wenigen Szenarien eine Warnung oder Notbremsung angefordert wird, ist, dass der Fußgänger in einem so großem Abstand zum Fahrzeug beginnt die Straße zu überqueren, dass es

Szenario ID	Szenarien mit kreuzendem Fußgänger vor dem Fahrzeug				Szenarien ohne kreuzendem Fußgänger vor dem Fahrzeug	
	Absolut	Relativ	mit FCW & ohne AEB	mit FCW & mit AEB	Absolut	Relativ
1	93	83.0 %	43	37	19	17.0 %
2	40	35.7 %	14	12	72	64.3 %
3	90	80.3 %	6	4	22	19.7 %
4	87	77.7 %	27	20	25	22.3 %

Tabelle 3.4: Ergebnisse der Szenarienvvalidierung

zu keiner kritischen Annäherung kommt bei der das Notbremsystem eingreifen muss.

Zusammenfassend lässt sich festhalten, dass die Annahme, dass konkrete Szenarien die gleichen Aktionen und Events besitzen wie die logischen Szenarien aus denen sie abgeleitet wurden, nicht universell gültig zu sein scheint und dass Parameterräume von logischen Szenarien nicht grundsätzlich als kontinuierlich angesehen werden können. Im Rahmen der vorliegenden Dissertation wird daher von einer Diskontinuität von logischen Szenarien gesprochen.

Definition 3.2 (Diskontinuität logischer Szenarien) *Beschreibt das Phänomen, dass einzelne Parameterkombinationen, eines eigentlich als zusammenhängend angenommenen Parameterraums eines logischen Szenarios, zu konkreten Szenarien führen, die inhaltlich nicht mehr dem ursprünglichen logischen Szenario entsprechen.*

Sie können also konkrete Szenarien beinhalten, die inhaltlich einem anderen logischen Szenario zugeordnet werden müssen (siehe Abbildung 3.5). Weiterhin weisen die Ergebnisse der Untersuchung darauf hin, dass einzelne Parametrisierungen für kritische Szenarien zwar aus Unfalldaten extrahiert werden können, aber die Verallgemeinerung in Form einer Ableitung von Parameterbereichen daraus nur eingeschränkt möglich ist.

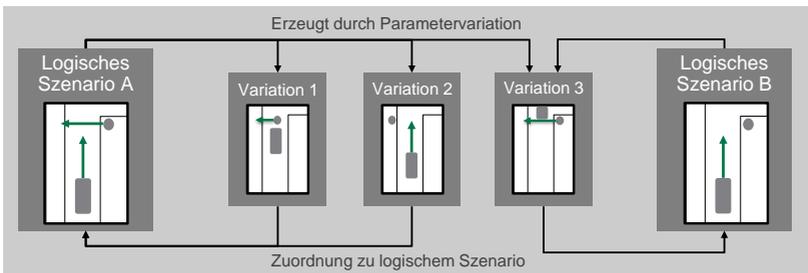


Abbildung 3.5: Visualisierung der Diskontinuität logischer Szenarien

Aus der Analyse lassen sich folgende Erkenntnisse gewinnen, die bei der Definition von Anforderungen an ein Konzept zur Bewertung von Fahrerassistenzsystemen in Fahrszenarien berücksichtigt werden.

- Der Verlauf von konkreten Szenarien nach einer Parametervariation ist variabel.
- Durch die Parametervariation werden konkrete Szenarien erzeugt, in denen das betrachtete Notbremssystem nicht aktiviert wurde. Beim Testen müssen diese Szenarien nicht betrachtet werden.
- Nicht alle konkreten Szenarien, die durch Parametervariation aus einem logischen Szenario erzeugt werden, besitzen die spezifizierten Events und Aktionen. Da das szenariobasierte Testen mit einer Generalisierungsannahme arbeitet, um die korrekte Funktionalität für ein logisches Szenario nachzuweisen, muss nach der Durchführung von Tests überprüft werden, ob das Szenario valide war.

3.4 Ableitung von Anforderungen an ein Testkonzept für das szenariobasierte Testen

3.4.1 Allgemeingültige Anforderungen

Um den szenariobasierten Testansatz bereits entwicklungsbegleitend in den automobilen Entwicklungsprozess einbinden zu können, muss die Bestimmung des Reifegrads eines Fahrerassistenzsystems ermöglicht werden. Nach Definition 2.7 gibt der Reifegrad während der Produktentwicklung an, wie in welchem Umfang ein System die an es gestellten Anforderungen erfüllt. Damit kann der Reifegrad nicht ausschließlich durch die Bewertung von Szenarien mittels Kritikalitätsmetriken erfolgen. Um die Anforderungserfüllung überprüfen zu können, muss das konkrete Funktionsverhalten bewertet werden. Zusätzlich muss auch sichergestellt

werden, dass nur Testergebnisse von konkreten Szenarien verwendet werden, bei denen das gewünschte Verhalten aufgetreten ist.

Damit ergeben sich folgende allgemeine Anforderungen an ein szenariobasiertes Testkonzept:

Allg1: Mit dem Testkonzept müssen Aussagen über den Reifegrad eines Fahrerassistenzsystems erzeugt werden können.

Allg2: Das Testkonzept muss Funktionsverhalten in konkreten Szenarien bewerten können.

Allg3: Jeder ausgeführte Testfall soll nach der Ausführung dahingehend evaluiert werden, ob der spezifizierte Ablauf des logischen Szenarios aufgetreten ist (vgl. Abschnitt 3.3.2).

3.4.2 Anforderungen an die Bestimmung des Reifegrads

Um den Reifegrad eines Fahrerassistenzsystems in einem szenariobasierten Testprozess bestimmen zu können, ist es nicht ausreichend, das korrekte Verhalten in einzelnen Testfällen nachzuweisen. Dementsprechend muss der Funktionsnachweis für jedes logische Szenario der gesamten ODD in ausreichendem Umfang erbracht werden.

Damit ergeben sich folgende allgemeine Anforderungen an die Bestimmung des Reifegrads:

Agg1: Bei der Beurteilung eines logischen Szenarios soll berücksichtigt werden, ob die Generalisierungsannahme des szenariobasierten Testens verwendet werden kann.

Agg2: Es sollen Auswertungen sowohl für ein konkretes Szenario, ein logisches Szenario als auch aggregiert für alle betrachteten logischen Szenarien durchgeführt werden.

3.4.3 Anforderungen an die Bewertung von Fahrerassistenzsystemen

Damit die Erfüllung von Anforderungen in einer ODD überprüft werden kann, müssen die eingesetzten Bewertungsmethoden eine Rückverfolgbarkeit von Testergebnissen auf Systemanforderungen ermöglichen. Aus dem szenariobasierten Ansatz ergibt sich zudem die Notwendigkeit, dass verschiedene logische Szenarien getestet werden müssen. Zusätzlich unterscheidet sich das szenariobasierte Testen vom anforderungsbasierten Testen darin, dass nicht mehr zu jedem Zeitpunkt bekannt ist, in welchem Testschritt sich das FAS befindet und welches Systemverhalten getestet werden muss. Daher ist im Vorfeld nicht bekannt, wann welches Systemverhalten wie lange erwartet und überprüft werden muss.

Damit ergeben sich folgende Anforderungen an die Bewertung von Fahrerassistenzsystemen:

Bew1: Die durchgeführten Bewertungen müssen Systemanforderungen zugeordnet werden können.

Bew2: Fahrerassistenzsysteme müssen in Szenarien bewertet werden können, deren Ablauf variabel ist.

Bew3: Eine kontinuierliche Bewertung der Fahrerassistenzsysteme während der gesamten Zeitspanne eines Szenarios soll möglich sein.

Bew4: Die Dauer und Häufigkeit von Fehlern muss ermittelt werden können.

Bew5: Fahrerassistenzsystemen müssen in unterschiedlichen logischen Szenarien bewertet werden können.

3.4.4 Anforderungen an die Überprüfung des Testinhalts

Der dynamische Inhalt von Fahrscenarien wird durch die auftretenden Aktionen und Events charakterisiert. Dementsprechend muss die Überprüfung des Testinhalts diese Aspekte umfassen. Da das szenariobasierte Testen mit einer Zunahme an durchzuführenden Tests verbunden ist, müssen die verwendeten Methoden skalierbar sein.

Damit ergeben sich folgende Anforderungen an die Überprüfung des Testinhalts:

Val1: Für die Überprüfung des Testinhalts sollen alle relevanten Aspekte eines Szenarios betrachtet werden.

Val2: Die Überprüfung des Testinhalts soll für verschiedene Fahrerassistenzsysteme eingesetzt werden können.

Val3: Die Überprüfung des Testinhalts soll für verschiedene logische Szenarien eingesetzt werden können.

4 Entwurf eines Konzepts zum Testen von Fahrerassistenzsystemen in Fahrszenarien

Aus den abgeleiteten Anforderungen aus Abschnitt 3.4 leiten sich drei übergeordnete Aspekte eines szenariobasierten Testkonzepts ab:

1. die Aggregation von Testergebnissen zur Bestimmung des Reifegrads (vgl. Forschungsfrage 1),
2. die Bewertung des Funktionsverhaltens eines FAS in einem konkreten Szenario zur Überprüfung, ob die Systemanforderungen erfüllt sind (vgl. Forschungsfrage 2) und
3. die Überprüfung der durchgeführten Szenarien (vgl. Forschungsfrage 3)

Um den Reifegrad bestimmen zu können, wird ein Testkonzept erarbeitet, welches je ein Modul zur Adressierung der relevanten Aspekte besitzt (siehe Abbildung 4.1). Aufsatzpunkt des Konzepts ist das Ergebnis der Simulation von konkreten Szenarien im Block Szenariodurchführung. Die konkreten Szenarien werden, wie im vorherigen Kapitel beschrieben, beispielsweise über eine Parametervariation aus logischen Szenarien erzeugt.

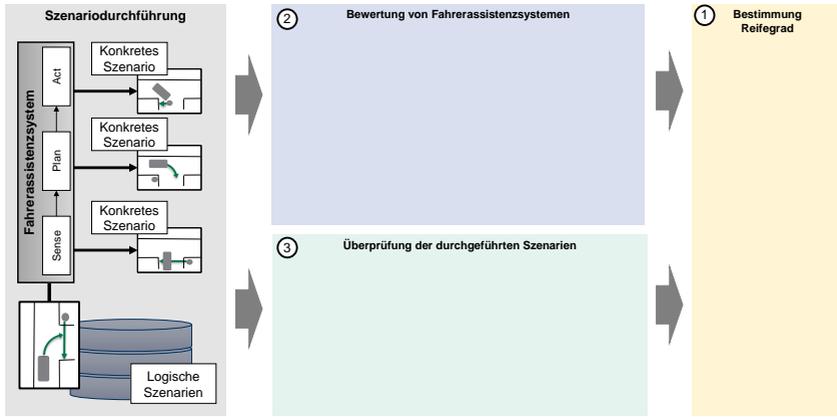


Abbildung 4.1: Strukturierung der notwendigen Module zum Testen von Fahrerassistenzsystemen in Fahrscenarien

4.1 Bestimmung des Reifegrads eines Fahrerassistenzsystems

Für aktuelle Fahrerassistenzsysteme bis zur Automatisierungsstufe zwei (vgl. Abschnitt 2.1) muss der Fahrer im Fehlerfall jederzeit als Rückfallebene zur Verfügung stehen. Während der Entwicklung wird daher die grundlegende Funktionalität des Systems stichprobenhaft in ausgewählten Szenarien getestet [174]. Der Reifegrad des FAS wird beim anforderungsbasierten Testen dann durch das erfolgreiche Absolvieren von Testfällen bestimmt, mit denen die Erfüllung von Systemanforderungen exemplarisch nachgewiesen wird. Ob dabei alle Systemanforderungen mit Tests überprüft werden, hängt von der gewählten Teststrategie ab (vgl. Abschnitt 2.3.2.1).

Im Gegensatz dazu soll beim szenariobasierten Testen das FAS zukünftig in allen logischen Szenarien der ODD getestet werden und in jedem logischen Szenario wird überprüft, ob die Systemanforderungen erfüllt sind (siehe Abbildung 4.2). In diesem Zusammenhang werden die Begriffe „Testbreite“ und „Testtiefe“ definiert als:

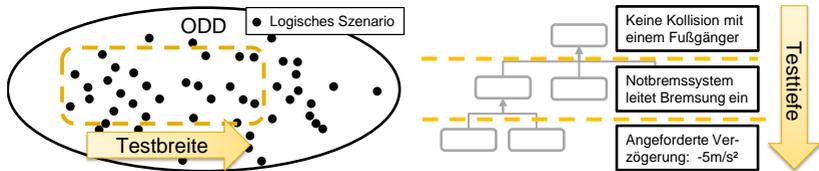


Abbildung 4.2: Darstellung der Begriffe Testbreite und Testtiefe

Definition 4.1 (Testbreite) Die Testbreite gibt an, in welchem Umfang die ODD während des Testens betrachtet wird. Gemessen wird die Testbreite durch die Testabdeckung mit logischen Szenarien als Testabdeckungselement (vgl. Abschnitt 2.3.2.1).

Definition 4.2 (Testtiefe) Die Testtiefe gibt an, bis zu welcher (Sub-) Systemebene Anforderungen während des Testens in einem logischen Szenario betrachtet werden. Gemessen wird die Testtiefe durch die Testabdeckung mit Systemanforderungen als Testabdeckungselement (vgl. Abschnitt 2.3.2.1).

Da beim anforderungsbasierten Testen das FAS in wenigen Szenarien detailliert betrachtet wird, besitzt der Ansatz eine hohe Testtiefe und niedrige Testbreite. Das Testkonzept für szenariobasiertes Testen aus dem Forschungsprojekt PEGASUS, bewertet dagegen das FAS in vielen logischen Szenarien anhand von Kritikalitätsmetriken auf der Fahrzeugebene, wie beispielsweise einer Kollision oder TTC. Damit besitzt es eine hohe Testbreite aber nur eine niedrige Testtiefe. Um zukünftig fundierte Aussagen über den Reifegrad eines FAS treffen zu können, muss sowohl eine hohe Testbreite als auch Testtiefe erreicht werden.

Um dies zu berücksichtigen, wird der Reifegrad Q_O eines FAS in der gesamten ODD durch die Aggregation von szenariospezifischen Reifegraden Q_{S_L} der einzelnen logischen Szenarien bestimmt (vgl. Definition 2.7). Während der Entwicklung eines Fahrerassistenzsystems muss daher nachgewiesen werden, dass das FAS in allen N_{S_K} konkreten Szenarien der N_{S_L} logischen Szenarien der ODD korrekt funktioniert. Die Bestimmung der logischen Szenarien, sowie die

Ableitung von konkreten Szenarien für das Testen, ist bereits Bestandteil verschiedener wissenschaftlicher Arbeiten¹ und wird daher nicht betrachtet. Aus diesem Grund werden für die Bestimmung des Reifegrads folgende Annahmen getroffen:

- Alle zu testende logischen Szenarien sind bekannt (geplante Testbreite)
- Für alle zu testenden logischen Szenarien ist bekannt, welche Systemanforderungen in dem logischen Szenario überprüft werden sollen (geplante Testtiefe)
- Für alle zu testenden logischen Szenarien ist bekannt, wie viele konkrete Szenarien mit welche Parametrisierungen getestet werden müssen

Der Reifegrad in der gesamten ODD Q_O wird berechnet durch:

$$Q_O = \sum_{N_{S_L}} \frac{1}{N_{S_L}} Q_{S_L} \quad (4.1)$$

Der szenariospezifische Reifegrad Q_{S_L} hängt wiederum davon ab, in welchem Umfang die N_A Anforderungen in den konkreten Szenarien erfüllt wurden. Für jede Anforderung muss daher ein Erfüllungsgrad $r_{\text{fulfill},A}$ ermittelt werden.

$$Q_{S_L} = \sum_{N_{S_K}} \frac{1}{N_{S_K}} * \left(\sum_{N_A} \frac{1}{N_A} * r_{\text{fulfill},A} \right) \quad (4.2)$$

Da die szenariobasierte Absicherung mit der Diskretisierung von kontinuierlichen Parameterräumen arbeitet und durch die Generalisierungsannahme nicht alle theoretisch möglichen Parameterkombinationen getestet werden (vgl. Abschnitt 2.3.3), sind Aussagen über Q_{S_L} grundsätzlich mit einer Unsicherheit behaftet. Um eine generalisierte Aussage für den Reifegrad in einem logischen Szenario treffen zu können, muss die Unsicherheit auf ein vertretbares Minimum reduziert werden. Die Fragestellung, ab wann die Unsicherheit ein vertretbares Maß erreicht

¹ zum Beispiel: [10, 44, 61, 88, 130, 160, 195, 208, 210, 215]

hat, ist Bestandteil des wissenschaftlichen Diskurses und wird beispielsweise in Forschungsprojekten wie VV-Methoden diskutiert.

Die Unsicherheit des aggregierten Ergebnisses Q_{S_L} nimmt mit jedem getesteten konkreten Szenario ab und ist unabhängig von dem eigentlichen Testergebnis. Zusätzlich hängt sie auch von der Abdeckung des Parameterraums ab. Wenn beispielsweise der Parameterraum eines logischen Szenarios eine Fahrzeuggeschwindigkeit von 10 km/h bis 50 km/h vorsieht, allerdings nur Tests im Bereich 10 km/h bis 20 km/h durchgeführt werden, dann kann daraus keine generalisierte Aussage für das gesamte logische Szenario getroffen werden. Mit der Generalisierungskonfidenz c_{G,S_L} werden diese Unsicherheit in dem vorgestellten Testkonzept berücksichtigt.

Definition 4.3 (Generalisierungskonfidenz) *Repräsentiert das Vertrauen in den ermittelten Reifegrad für ein logisches Szenario in Abhängigkeit der Anzahl an durchgeführten Tests und des abgedeckten Parameterraums.*

Besitzt ein konkretes Szenario allerdings nicht den spezifizierten Inhalt des logischen Szenarios, wie beispielsweise einen kreuzenden Fußgänger vor dem Fahrzeug, dann kann aus diesem Szenario keine generalisierte Aussage über das logische Szenario abgeleitet werden. Damit werden bei der Ermittlung der Anzahl an durchgeführten Tests nur die validen konkreten Szenarien gezählt (vgl. Abschnitt 3.3.2). Für die Bestimmung der Parameterraumabdeckung werden allerdings alle konkreten Szenarien verwendet. Eine Generalisierungskonfidenz von 1.0 bedeutet, dass jeder Bereich des Parameterraums im vorgegebenen Umfang getestet wurde.

Neue Entwicklungsstände des FAS während der Entwicklung führen zu einem Zurücksetzen der Konfidenz. Der Reifegrad in einem logischen Szenario, in dem N_A Systemanforderungen überprüft werden sollen und für das $N_{S_{K,v}}$ valide konkrete Szenarien getestet wurden (vgl. Abschnitt 3.3.2), wird daher bestimmt durch:

$$Q_{S_L} = \sum_{N_{S_{K,v}}} \frac{1}{N_{S_{K,v}}} * \left(\sum_{N_A} \frac{1}{N_A} * r_{\text{fulfill},A} \right) * c_{G,S_L} \quad (4.3)$$

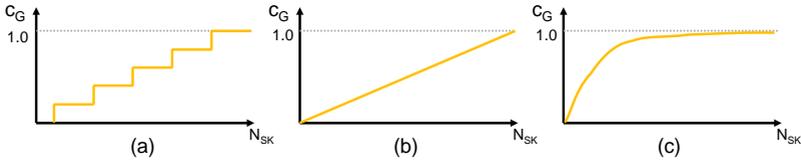


Abbildung 4.3: Mögliche Modelle für die Zunahme der Generalisierungskonfidenz c_{G,S_L} in Abhängigkeit der Anzahl an getesteten konkreten Szenarien N_{SK}

Die Zunahme der Generalisierungskonfidenz während des Testprozesses lässt sich unterschiedlich modellieren. In Abbildung 4.3 sind drei theoretische Modelle für die Zunahme dargestellt. *Modell a* verwendet eine Treppenfunktion, bei der die Konfidenz schrittweise erhöht wird, sobald eine definierte Anzahl an konkreten Tests durchgeführt oder die Abdeckung eines Teils des Parameterraums erreicht wurde. Ebenfalls können damit maximal erreichbare Konfidenzen für die verschiedenen Testplattformen definiert werden. Im Gegensatz dazu bildet das *Modell b* eine lineare Zunahme der Konfidenz ab, bei der jeder ausgeführte Test den gleichen Beitrag zur Konfidenz leistet. Eine weitere Möglichkeit zur Modellierung ist eine asymptotische Annäherung an eine Zielkonfidenz, wie in *Modell c* dargestellt. Der Verlauf der Konfidenz von *Modell c* lässt sich so interpretieren, dass die ersten getesteten Szenarien noch einen hohen Anstieg bewirken. Dieser Anstieg flacht mit zunehmender Anzahl an getesteten Szenarien jedoch ab.

Für die weitere Ausarbeitung wird beispielhaft eine asymptotische Zunahme der Generalisierungskonfidenz angenommen, welche von der Anzahl an getesteten validen Szenarien $N_{SK,v}$, sowie deren Verteilung abhängt. In der Arbeit wird exemplarisch eine Gleichverteilung der Szenarien über den gesamten Parameterraum vorausgesetzt. Die Berechnungsvorschrift für die Generalisierungskonfidenz lautet daher:

$$c_{G,S_L} = \begin{cases} \frac{N_{SK,v}}{\sqrt{N_{SK,v}^2 + n}}, & \text{Szenarien gleichverteilt} \\ 0, & \text{Szenarien nicht gleichverteilt} \end{cases} \quad (4.4)$$

Der Parameter n wird dafür verwendet die Funktion so einzustellen, dass eine gewünschte Zielkonfidenz $c_{G, Dsr}$ mit einer festgelegten Anzahl an Szenarien $N_{K, Dsr}$ erreicht wird. In der vorliegenden Arbeit bedeutet die Notation „ $N_{K, 95\%} = 100$ “, dass 100 konkrete Szenarien getestet werden müssen um eine Konfidenz von 95 % zu erreichen. Die Umformung von Gleichung 4.4 ergibt:

$$n = \frac{N_{K, Dsr}^2}{c_{G, Dsr}^2} - N_{K, Dsr}^2. \quad (4.5)$$

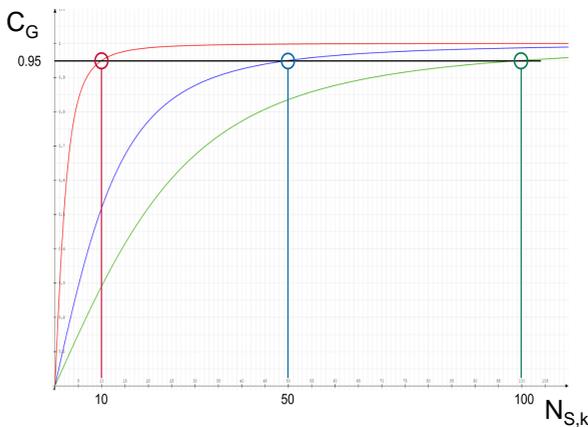


Abbildung 4.4: Zunahme der Generalisierungskonfidenz nach Gleichung 4.4 für verschiedene Parameter n . Rot ($N_{K, 95\%} = 10$), blau ($N_{K, 95\%} = 50$), grün ($N_{K, 95\%} = 100$)

Abbildung 4.4 zeigt den Verlauf von Gleichung 4.4 für eine gewählte Zielkonfidenz von 95 % bei variierender Anzahl an geforderten konkreten Szenarien. Die rote Kurve benötigt 10 Szenarien, die blaue Kurve 50 und die grüne Kurve 100.

Für die Bestimmung des szenariospezifischen Reifegrads wird damit sowohl die Gesamtanzahl an konkreten Szenarien N_{S_K} (valide und nicht valide) als auch die Teilmenge an validen konkreten Szenarien $N_{S_{K,v}}$ benötigt. Daraus folgt, dass während des Testens gegebenenfalls auch nicht valide konkrete Szenarien getestet

werden müssen, um die Gültigkeit der Generalisierungsannahme zu überprüfen. Der Reifegrad für die gesamte ODD berechnet sich dann durch:

$$Q_O = \sum_{N_{SL}} \frac{1}{N_{SL}} * \left(\sum_{N_{SK,v}} \frac{1}{N_{SK,v}} * \left(\sum_{N_A} \frac{1}{N_A} * r_{\text{fulfill},A} \right) * c_{G,SL} \right) \quad (4.6)$$

Damit lässt sich die Bestimmung des Reifegrads in die drei Schritte unterteilen: Bestimmung der Generalisierungskonfidenz, Bestimmung der szenariospezifischen Reifegrade und Aggregation aller logischer Szenarien zur Bestimmung des Reifegrads in der ODD (siehe Abbildung 4.5).



Abbildung 4.5: Vervollständigung des Testkonzepts um die Bestimmung des Reifegrads

4.2 Bewertung von Fahrerassistenzsystemen zur Bestimmung der Anforderungserfüllung

Um den Grad der Anforderungserfüllung $r_{\text{fulfill},A}$ quantifizieren zu können, muss für jedes konkrete Szenario bestimmt werden, in welchem Umfang die Anforderungen erfüllt sind. Für die Bewertung eines FAS in zeitlich variablen Szenarien sind grundsätzlich zwei Herangehensweisen möglich. Entweder es wird mittels Testorakel vor der Durchführung des Szenarios versucht zu präzisieren, wie das Szenario verläuft und zu welchem Zeitpunkt das FAS bewertet werden muss, oder die Bewertung erfolgt sobald das Verhalten auftritt. Im Rahmen der vorliegenden Arbeit werden daher zwei Ansätze untersucht. Der erste Ansatz schätzt den Szenarioverlauf anhand der Parameter und leitet daraus Testfälle ab (vgl. Abschnitt 2.3.2.1). Im Gegensatz dazu, versucht der zweite Ansatz dynamisch das Systemverhalten des FAS zu bewerten.

4.2.1 Testorakel für die Prädiktion des Systemverhaltens

Die Prädiktion des Szenarioverlaufs soll anhand eines Modells erfolgen, welches aus den Szenarioparametern den Zeitpunkt t_{Rel} schätzt, an dem ein bestimmtes Systemverhalten von einem FAS erwartet wird. Grundlage für die Modellierung sind Kenntnisse über den allgemeinen Szenarioverlauf (vgl. Anforderung **Bew2**), welche beispielsweise durch die Simulation von exemplarischen Parametrisierungen erhalten werden können, sowie Systemwissen aus den Anforderungen (vgl. Anforderung **Bew1**).

Nach der Durchführung der konkreten Szenarien, wird dann zum geschätzten Zeitpunkt t_{Rel} überprüft, ob das Systemverhalten des FAS dem spezifizierten Systemverhalten entspricht. Im Folgenden wird eine beispielhafte Modellierung

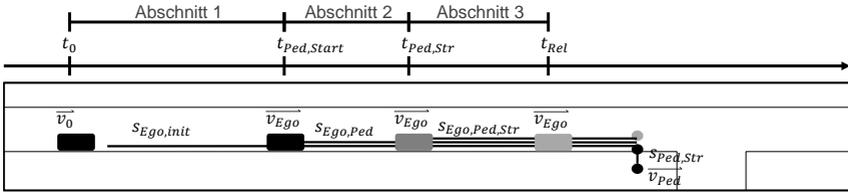


Abbildung 4.6: Modellierung des Notbremsenszenarios zu verschiedenen Zeitpunkten während des Szenarioverlaufs

für das Unfallszenario eins aus Kapitel 3.3.1 durchgeführt. Als FAS wird exemplarisch ein Notbremsensystem betrachtet, für das t_{Rel} den Zeitpunkt darstellt, an dem der Fußgänger für das System relevant wird. Dies ist beispielsweise dann der Fall, wenn der Fußgänger die Straße betritt und die Time-to-Collision einen Grenzwert TTC_{Rel} unterschreitet.

Für die Modellierung werden das Fahrzeug und der Fußgänger jeweils als Massenpunkt betrachtet. Zusätzlich wird die Bewegung vereinfacht als gleichförmig angenommen und fahrdynamische Effekte, wie die Reibung oder der Luftwiderstand, vernachlässigt. Das Szenario lässt sich beispielhaft in drei Abschnitte unterteilen, mit denen ein Prädiktionsmodell aufgebaut werden kann (siehe Abbildung 4.6). Im ersten Abschnitt von $t = t_0$ bis $t = t_{Ped,Start}$ beschleunigt das Fahrzeug auf seine Zielgeschwindigkeit (Szenarioparameter) und fährt mit einer konstanten Geschwindigkeit bis der Abstand zum Fußgänger einen Grenzwert $as_{Fzg,Ped}$ unterschreitet, sodass der Fußgänger beginnt loszulaufen. Der zweite Abschnitt beinhaltet die Annäherung des Fußgängers zur Straße und endet mit dem Betreten der Straße durch den Fußgänger zum Zeitpunkt $t = t_{Ped,Str}$. Der dritte Abschnitt endet mit dem Zeitpunkt t_{Rel} , an dem die TTC zum Fußgänger kleiner ist als TTC_{Rel} .

Im ersten Abschnitt hängt die benötigte Zeit $t_{Ped,Start}$ von dem initialen Abstand zwischen Fahrzeug und Fußgänger $s_{Fzg,init}$, der Zielgeschwindigkeit des Fahrzeugs v_{Fzg} , der Beschleunigung des Fahrzeugs a_{Fzg} und dem Abstand $as_{Fzg,Ped}$ ab.

$$t_{Ped,Start} = t_{Accel} + t_{Const}. \quad (4.7)$$

Die Zeit t_{Const} berechnet sich durch die zurückzulegende Strecke bis der Fußgänger losläuft, nach Abschluss der initialen Beschleunigungsphase.

$$t_{Const} = \frac{(s_{Fzg,init} - a_{s_{Fzg,Ped}} - s_{Fzg,Accel})}{v_{Fzg}}. \quad (4.8)$$

Daraus folgt mit $t_{Accel} = \frac{v_{Fzg}}{a_{Fzg}}$ und $s_{Fzg,Accel} = \frac{a_{Fzg} * t_{Accel}^2}{2}$:

$$t_{PedStart} = \frac{v_{Fzg}}{a_{Fzg}} + \frac{s_{Fzg,init} - a_{s_{Fzg,Ped}} - s_{Fzg,Accel}}{v_{Fzg}} - \frac{v_{Fzg}}{2 * a_{Fzg}}. \quad (4.9)$$

Der zweite Abschnitt dauert an, bis der Fußgänger bei $t = t_{Ped,Str}$ die Straße betritt. Die Zeit ist abhängig von der Geschwindigkeit des Fußgängers v_{Ped} (Szenarioparameter) und dem Abstand vom Fußgänger bis zur Straße $s_{Ped,Str}$.

$$t_{Ped,Str} = \frac{v_{Fzg}}{2 * a_{Fzg}} + \frac{s_{Fzg,init} - a_{s_{Fzg,Ped}}}{v_{Fzg}} + \frac{s_{Ped,Str}}{v_{Ped}}. \quad (4.10)$$

Ab dem Zeitpunkt $t = t_{Ped,Str}$ kann der Fußgänger für das Fahrerassistenzsystem relevant sein, wenn der Abstand zum Fußgänger $s_{Fzg,Ped,Str}$ kleiner ist als ein Grenzwert s_{Rel} . Klassischer Weise wird ein solcher Abstand nicht in Meter angegeben, sondern als TTC in Sekunden. Damit kann der Abstand unabhängig von der Geschwindigkeit des Fahrzeugs definiert werden. Ist der Abstand größer als der Grenzwert, dann folgt ein dritter Abschnitt, indem sich das Fahrzeug weiter an den Fußgänger annähert, bis bei $t = t_{Rel}$ der Abstand $s_{Fzg,Ped,Str}$ kleiner ist als s_{Rel} . Die Abstände $s_{Fzg,Ped,Str}$ und s_{Rel} zum Zeitpunkt $t = t_{PedStreet}$ berechnen sich:

$$s_{Fzg,Ped,Str} = \sqrt{s_{Fzg,Ped,E}^2 - s_{Ped,Str}^2} - \frac{s_{Ped,Str}}{v_{Ped}} * v_{Fzg}. \quad (4.11)$$

$$s_{Rel} = TTC_{Rel} * v_{Fzg}. \quad (4.12)$$

Daraus ergibt sich für den Zeitpunkt, wann der Fußgänger für das System relevant wird, für die Fälle: [1] $s_{Fzq, Ped, Str} \leq s_{Rel}$ und [2] $s_{Fzq, Ped, Str} > s_{Rel}$

$$t_{Rel} = \begin{cases} \frac{v_{Fzq}}{2 * a_{Fzq}} + \frac{s_{Fzq, mit} - a_{Fzq, Ped}}{v_{Fzq}} + \frac{s_{Ped, Str}}{v_{Ped}} & [1] \\ \frac{v_{Fzq}}{2 * a_{Fzq}} + \frac{s_{Fzq, mit} - a_{Fzq, Ped}}{v_{Fzq}} + \frac{s_{Ped, Str}}{v_{Ped}} + \frac{s_{Fzq, Ped, Str}}{v_{Fzq}} & [2] \end{cases} \quad (4.13)$$

Zu den prädierten Zeitpunkten t_{Rel} wird dann in den konkreten Szenarien überprüft, ob das Notbremssystem eine Frontkollisionswarnung ausgibt.

4.2.2 Ansatz zur dynamischen Bewertung von Systemen

In einem zweiten Ansatz wird untersucht, ob eine dynamische Bewertung des Systems möglich ist². Der Ansatz orientiert sich dabei an der Grundidee des deklarativen Testens, die Bewertung des Testobjekts von der Beschreibung des Testablaufs zu entkoppeln (vgl. Abschnitt 3.1.2 auf Seite 57). Die Bewertung setzt sich dabei aus der *Answer*, dem erwarteten Systemverhalten und dem *Verifier*, der Überprüfung des tatsächlichen Systemverhaltens zusammen. Bezogen auf das Testkonzept aus Abbildung 4.1, übernimmt der Block der Szenariodurchführung die Aufgabe der *Executors*.

Für die Übertragung des deklarativen Testens auf die Bewertung von Fahrerassistenzsystemen in Fahrscenarien, ergeben sich daher zwei Fragen:

1. Wie kann das erwartete Systemverhalten abgeleitet und so formalisiert werden, dass es im *Verifier* überprüft werden kann.
2. Wie kann ein *Verifier* konzipiert werden, der die Überprüfung des Systemverhaltens in konkreten Fahrscenarien ermöglicht.

² Veröffentlicht in [King et al., 2019]

Systemanforderungen bilden die Grundlage für die Ableitung des erwarteten Systemverhaltens. Dadurch wird sichergestellt, dass die Anforderung **Bew1** an das Testkonzept erfüllt ist (vgl. Abschnitt 3.4.3). Um das Verhalten in zeitlich variablen Szenarien testen zu können (vgl. Anforderung **Bew2**), muss zusätzlich auch bekannt sein, in welchen Situationen oder welchem Systemzustand das Systemverhalten erwartet wird (vgl. Definition 2.36). Beispielsweise regelt ein adaptiver Abstandsregeltempomat (ACC) auf einen Sollabstand und keine Sollgeschwindigkeit, wenn ein vorausfahrendes Fahrzeug vorhanden ist, was langsamer als die eingestellte Sollgeschwindigkeit fährt (vgl. Abschnitt 2.3.3). Ein Beispiel für einen Systemzustand als Voraussetzung wäre, dass ein ACC-System, während es aktiv ist, die eingestellte Sollgeschwindigkeit im Kombi-Instrument dauerhaft anzeigt.

Das erwartete Verhalten wird in „Systembewertungen“ formalisiert, welche nach einem „wenn-dann“-Schema aufgebaut sind. Das „wenn“ beschreibt die Situation oder den Systemzustand und das „dann“ das gewünschte Systemverhalten. Die obigen Beispiele würden laut Schema beschrieben als:

- **Wenn** der Abstandsregeltempomat aktiv ist und ein vorausfahrendes Fahrzeug vorhanden ist und das vorausfahrende Fahrzeug langsamer fährt als die eingestellte Sollgeschwindigkeit, **dann** befindet sich der Abstandsregeltempomat in der Abstandsregelung.
- **Wenn** Abstandsregeltempomat aktiv ist, **dann** wird die Sollgeschwindigkeit im Kombi-Instrument angezeigt.

Formalisiert setzt sich eine Systembewertung aus einer Aktivierung (beschreibt das „wenn“) und einer Prüfung (beschreibt das „dann“) zusammen (siehe Abbildung 4.7). Sowohl die Aktivierung als auch die Prüfung bestehen aus Bedingungen, die logisch miteinander verknüpft sind.

Um diese Systembewertungen für das Testen einsetzen zu können, wird ein *Verifier* benötigt, der das Systemverhalten, beschrieben durch eine Aktivierung und eine Prüfung, in den konkreten Szenarien überprüft. Um die Anforderungen der kontinuierlichen Bewertung zu erfüllen (vgl. Anforderung **Bew3**), wird die Aktivierung der Systembewertung durch den *Verifier* in jedem Zeitschritt anhand

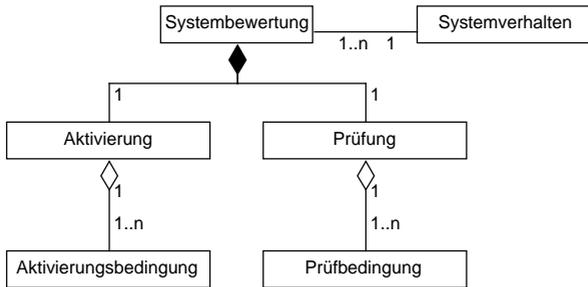


Abbildung 4.7: Formalisierung des erwarteten Verhaltens in Systembewertungen, die sich aus einer Aktivierung und einer Prüfung zusammensetzen

von Aktivierungsbedingungen (AB) evaluiert. Sind die AB erfüllt, erfolgt die Evaluation der Prüfbedingungen (PB). Für das Beispiel des ACC-Systems, ist der grundlegende Ablauf der Systembewertung als Aktivitätsdiagramm in Abbildung 4.8 dargestellt. Der obere Abschnitt des Diagramms repräsentiert die kontinuierliche Auswertung der Systembewertung in jedem Zeitschritt. Im unteren Abschnitt ist der prinzipielle Ablauf dargestellt, wie eine Systembewertung evaluiert wird.

Damit ein FAS über die gesamte Dauer eines Szenarios bewertet werden kann (vgl. Anforderung **Bew3**), erlaubt das Konzept die wiederholte Aktivierung einer Prüfung. Dies wird besonders für Fahrerassistenzsysteme benötigt, die während des gesamten Szenarios aktiv sind, wie zum Beispiel FAS für die Längs- oder Querregelung des Fahrzeugs. Im Fall einer fehlerhaften Querregelung eines Spurhaltesystems, muss jede zu große Abweichung von der Spurmitte erkannt werden (siehe Abbildung 4.9).

Zusätzlich zum Vorhandensein von Fehlern, muss auch die Dauer und Häufigkeit von Fehlern ermittelt werden (vgl. Anforderung **Bew4**). Dafür wird die Auswertung im *Verifier* in einem Zustandsautomaten dargestellt, der die Zustände „passed“ und „failed“ annehmen kann (siehe Abbildung 4.10).

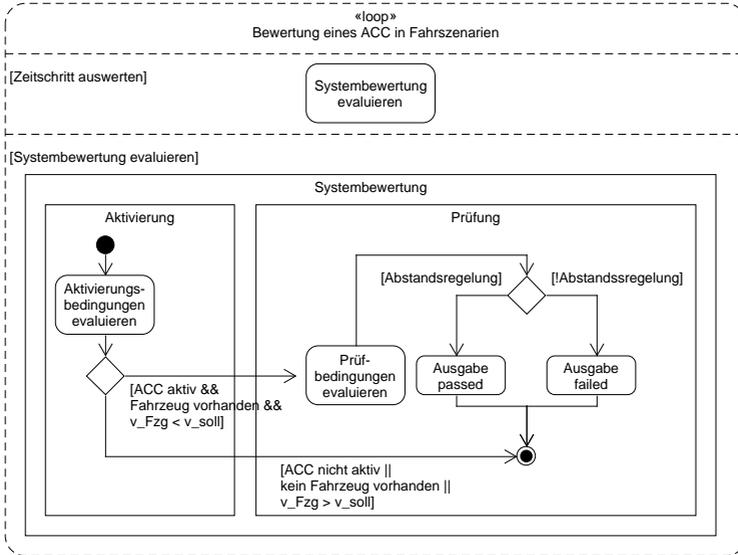


Abbildung 4.8: Aktivitätsdiagramm für die Bewertung eines ACC-Systems in Fahrscenarien

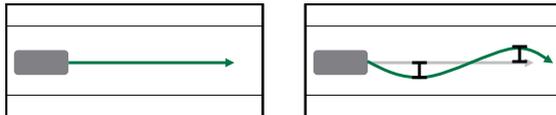


Abbildung 4.9: Beispiel für ein Querregelungssystem mit der Solltrajektorie als grauer Pfeil und der gefahrenen Trajektorie als grüner Pfeil, links: Regelung folgt der vorgegebenen Trajektorie, rechts: Regler schwingt um vorgegebene Trajektorie

Grundsätzlich kann für die Bewertung eines Fahrerassistenzsystems zwischen zwei Arten der Nachweiserbringung unterschieden werden. Entweder muss nachgewiesen werden, dass ein gewünschtes Verhalten auftritt, wie zum Beispiel die Ausgabe einer Warnung, oder ein Nachweis muss erbracht werden, dass ein bestimmtes Verhalten nicht aufgetreten ist. Es darf beispielsweise keine Kollision mit einem Fußgänger geben.

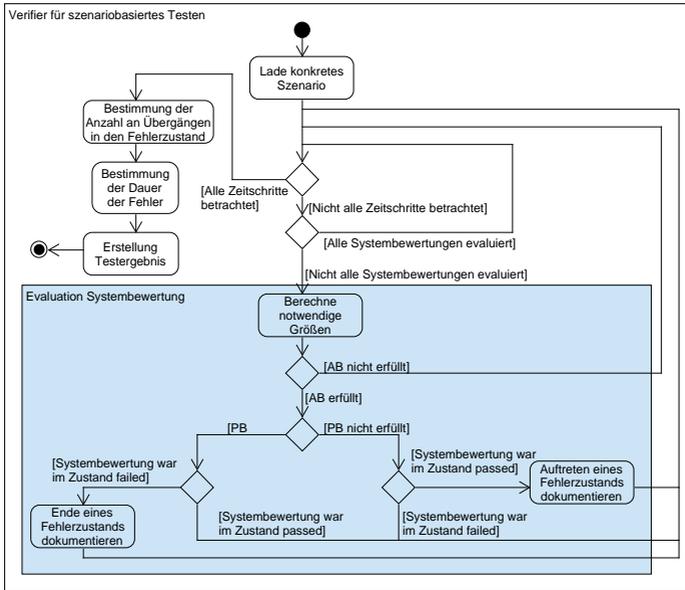


Abbildung 4.10: Aktivitätsdiagramm des Verifiers für die Ausführung der automatisierten Systembewertung

Für die Auswertung des Zustandsautomaten werden daher zwei Arten von Tests definiert, die sich im Startzustand des Zustandsautomaten unterscheiden. Der *Zielerreichungstest* bewertet, ob ein spezifizierter Wert innerhalb einer vorgegebenen Zeit erreicht wird. Im Gegensatz dazu überprüft der *Grenzwerttest*, dass ein spezifizierter Grenzwert nicht erreicht oder unter-/überschritten wird.

Um die Arten von Tests zu veranschaulichen, sollen beide Testarten an einem Beispiel verdeutlicht werden. Nachdem ein Tempomat eingeschaltet wird, muss die eingestellte Sollgeschwindigkeit innerhalb einer bestimmten Zeit erreicht werden. Für dieses Verhalten wird ein Zielerreichungstest verwendet. Bis zum Zeitpunkt, an dem die Sollgeschwindigkeit erreicht wird, ist die Testbedingung nicht erfüllt und der Zustandsautomat befindet sich im Startzustand „failed“. Wird die Sollgeschwindigkeit erreicht, dann soll diese vom Tempomat beibehalten werden.

Der Betrag der Differenz zwischen Soll- und Istgeschwindigkeit darf also einen bestimmten Grenzwert nicht überschreiten (Grenzwerttest). Solange die Testbedingung nicht verletzt wird, befindet sich der Zustandsautomat im Startzustand „passed“.

Um die Anforderung **Bew4** zu erfüllen, wird nach der Evaluation eines konkreten Szenarios sowohl die Anzahl an Übergängen in den Fehlerzustand, als auch die Dauer der Fehlerzustände bestimmt. Das Testergebnis für eine Systembewertung kann dann auf Basis dieser Informationen erstellt werden. Das Konzept ermöglicht es dem Anwender selber Regeln zu definieren, wann ein Test bestanden oder nicht bestanden ist.

4.2.3 Evaluation und Vergleich der Ansätze

Um beide Ansätze evaluieren zu können, werden sie beispielhaft für ein automatisiertes Notbremssystem umgesetzt. Mit beiden Ansätzen soll die Ausgabe einer Warnung durch das Notbremssystem getestet werden. Das System gibt eine Warnung aus, sobald ein Fußgänger in einem geschwindigkeitsabhängigen Abstand vor dem Fahrzeug die Straße betritt. Die Evaluation erfolgt mit dem Versuchsaufbau und der Versuchsdurchführung aus Kapitel 3.3.1.

Für die Evaluation des Testorakels wird für jedes konkrete Szenario der Zeitpunkt t_{Rel} berechnet und der Prädiktionsfehler für die Ausgabe einer Warnung bestimmt. Darüber hinaus werden für den Prädiktionsfehler verschiedene statistische Maßzahlen berechnet (siehe Tabelle 4.1).

Der Vergleich der gemessenen und prädizierten Zeitpunkte zeigt, dass die prädizierten Zeitpunkte von den gemessenen Zeitpunkten abweichen. Der Prädiktionsfehler nimmt mit steigender Geschwindigkeit des Fahrzeugs auf bis zu 50 % zu (siehe Tabelle 4.1). Für die logischen Szenarien eins bis drei wird der Zeitpunkt der Warnung grundsätzlich zu früh geschätzt. Im vierten Szenario tritt auch der Fall auf, dass die Warnung vor dem prädizierten Zeitpunkt ausgegeben

Statistische Maßzahl	Szenario 1	Szenario 2	Szenario 3	Szenario 4
Anzahl Szenarien	43	14	6	27
Mittelwert [s]	2.98	2.80	2.36	2.83
Standardabweichung [s]	0.95	1.48	1.66	1.64
Varianz [s]	0.90	2.19	2.76	2.68
Median [s]	2.95	2.66	2.37	2.61
Minimale Abweichung [s]	1.34/8.8 %	0.51/1.8 %	0.39/1.7 %	-0.20/-0.9 %
Maximale Abweichung [s]	5.27/50.9 %	5.71/46.8 %	5.59/18.2 %	5.61/44.0 %

Tabelle 4.1: Auswertung des Prädiktionsfehlers

wird. Als Ursache für die Prädiktionsfehler werden Abweichungen in der Modellierung angenommen, die beispielsweise durch die vereinfachende Annahme einer gleichförmigen Bewegung, sowie die Vernachlässigung fahrdynamischer Effekte, entstehen. Da eine kommerzielle Simulationsumgebung für die Evaluation verwendet wurde, bestand keine Möglichkeit die verwendete Fahrdynamikmodellierung der Simulationsumgebung mit der eigenen Modellierung abzugleichen. Weiterhin berechnet das Prädiktionsmodell auch für Parameterkombinationen einen Warnzeitpunkt, bei denen keine Warnung durch das Notbremsystem ausgegeben wird (siehe Abbildung 4.11). Diese Parameterkombinationen werden in Abbildung 4.11 durch einen Zeitpunkt der Warnung von 0.0 Sekunden dargestellt und wurden für die Evaluation von Tabelle 4.1 nicht berücksichtigt.

Werden die prädizierten Zeitpunkte für die Erstellung von Testfällen genutzt, dann bedeutet eine maximale Abweichung von 5.27 Sekunden, dass bei einer Geschwindigkeit von 34 km/h die Warnung 49.8 Meter zu früh erwartet wird. Für die Anwendung in einem szenariobasierten Ansatz haben Abweichungen von bis zu 50 Prozent zur Folge, dass so generierte Testfälle nicht für das Testen von Fahrerassistenzsystemen verwendet werden können.

Für die Evaluation der dynamischen Bewertung wird eine Systembewertung für die Ausgabe einer Frontkollisionswarnung implementiert und in einem dafür entwickeltem *Verifier* integriert. Der *Verifier* verwendet die Simulationsdaten, die für die Evaluation des ersten Ansatzes erzeugt wurden. Damit ist sichergestellt, dass beide Ansätze auf der selben Datenbasis arbeiten und die Ergebnisse vergleichbar sind. Für jedes konkrete Szenario wird die Abweichung zwischen dem Zeitpunkt der Warnung und der Aktivierung der Prüfung bestimmt. Analog zur Evaluation des prädiktiven Ansatzes, werden die Abweichungen aggregiert, um statistische Aussagen treffen zu können (siehe Tabelle 4.2).

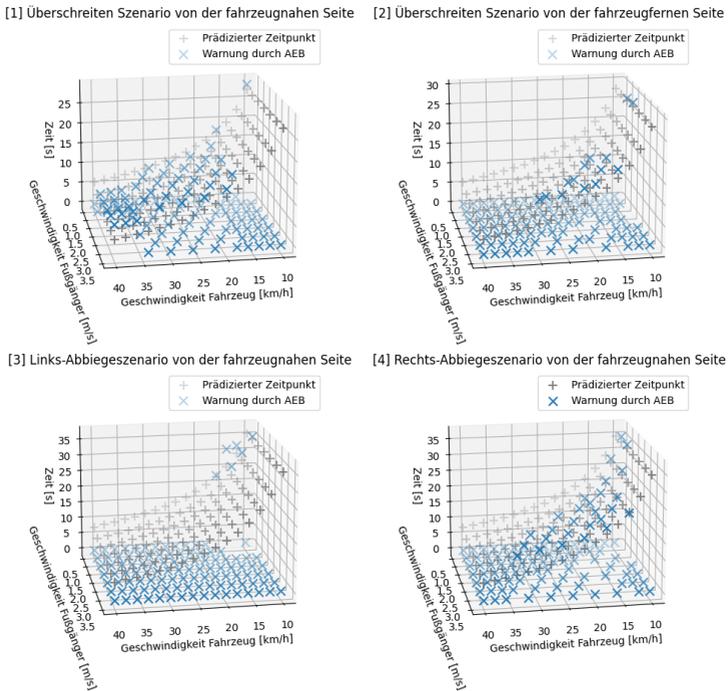


Abbildung 4.11: Visualisierung der Aktivierungszeitpunkte der FCW und der prädizierten Zeitpunkte

Statistische Maßzahl	Szenario 1	Szenario 2	Szenario 3	Szenario 4
Anzahl Szenarien	43	14	6	27
Mittelwert [s]	0.00	0.00	0.00	0.00
Standardabweichung [s]	0.00	0.00	0.00	0.00
Varianz [s]	0.00	0.00	0.00	0.00
Median [s]	0.00	0.00	0.00	0.00
Minimale Abweichung [s]	0.00	-0.08/-0.5 %	0.00	0.00
Maximale Abweichung [s]	0.00	0.00	0.00	0.00%

Tabelle 4.2: Auswertung der zeitlichen Abweichung der Prüfungsergebnisse

Die Ergebnisse der Auswertung zeigen, dass für ein einziges konkretes Szenario die Aktivierung der Prüfung 0.08 Sekunden zu spät erfolgt. Für die anderen konkreten Szenarien wurde die Prüfung im selben Simulationsschritt aktiviert, wie auch die Warnung angefordert wurde. In Abbildung 4.12 liegen daher beide Punkt übereinander.

Vor dem Vergleich der Ansätze soll diskutiert werden, welche Faktoren die Validität der Ergebnisse beeinflussen können. Ein Einflussfaktor ist, dass das Systemverständnis unvollständig oder nicht korrekt ist. In diesem Fall würde die Modellierung oder die Systembewertung nicht das tatsächliche Systemverhalten des Notbremssystems abbilden und die erzeugten Ergebnisse wären nicht vergleichbar. Da der Quellcode des Notbremssystems vorlag und ausführlich analysiert wurde, wird argumentiert, dass das zugrundeliegende Systemwissen vollständig und korrekt ist. Darüber hinaus kann die Validität beispielsweise durch die Wahl nicht aussagekräftiger Szenarien beeinflusst werden. Um dies zu verhindern, werden häufige Unfallszenarien aus der Verkehrsunfallforschung ausgewählt, da von diesen ausgegangen werden kann, dass sie für das betrachtete Notbremssystem relevant sind. Zuletzt wird die Auswahl der statistischen Maßzahlen als weiterer Einflussfaktor diskutiert. Beispielsweise können absolute Zeitfehler, bei unterschiedlich langen Dauern der Szenarien, potentiell unterschiedlich bewertet

werden. Um diesen Einfluss zu minimieren, werden bei der Auswertung sowohl unterschiedliche absolute als auch relative Maßzahlen betrachtet.

Der Vergleich der beiden Ansätze zeigt, dass das Systemverhalten, in dem betrachteten Beispiel, mit dem dynamischen Ansatz präziser bewertet werden konnte als mit dem vorgestellten Testorakel. Eine Verringerung des Prädiktionsfehlers ist zwar vermutlich möglich, erfordert allerdings eine genaue Nachmodellierung der Fahrdynamik der Simulationsumgebung. Das dafür notwendige Wissen ist bei kommerziellen Simulationsumgebungen nicht verfügbar.

Eine weitere Voraussetzung für die Verwendung eines Testorakels ist, dass der Ablauf des Szenarios im Vorfeld bekannt sein muss, damit er modelliert werden

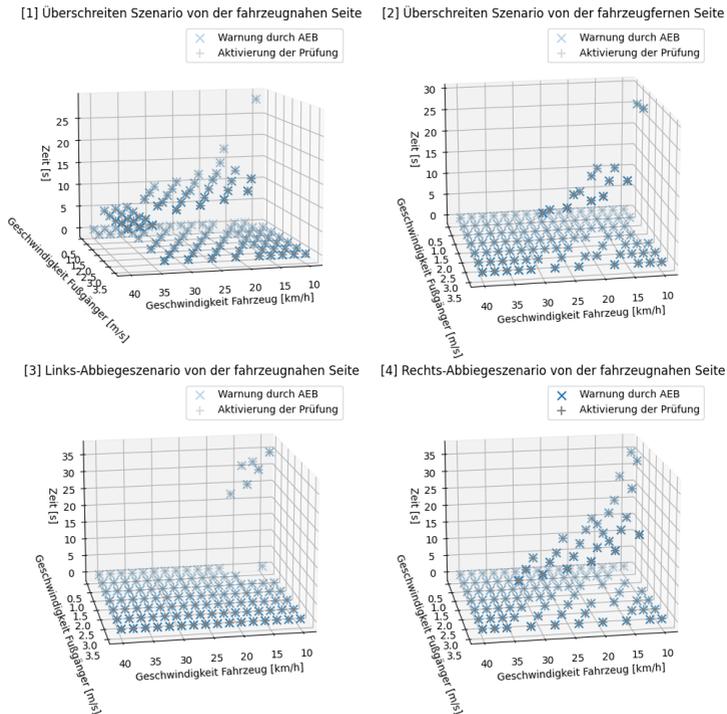


Abbildung 4.12: Visualisierung der Aktivierungszeitpunkte der FCW und der FCW-Prüfung

kann. Szenarien, bei denen mehrere Verkehrsteilnehmer in Abhängigkeit der Parametrisierung unterschiedlich miteinander interagieren und der zeitliche Ablauf dadurch nicht vorhersagbar ist, können daher mit diesem Ansatz nicht getestet werden. Darüber hinaus muss für jedes zu testende logische Szenario ein eigenes Testorakel entworfen werden, womit die Übertragbarkeit auf verschiedene logische Szenarien nicht gegeben ist (vgl. Anforderung **Bew5**). Aus diesen Gründen wird der dynamische Ansatz zur Bewertung von Fahrerassistenzsystemen in Fahrscenarien verwendet (siehe Abbildung 4.13). Der Erfüllungsgrad $r_{\text{fulfill},A}$ einer Anforderung, die mit N_B Systembewertungen getestet wird, berechnet sich folgendermaßen:

$$r_{\text{fulfill},A} = \sum_{N_B} \frac{1}{N_B} * P_{SB}. \tag{4.14}$$

P_{SB} gibt dabei den prozentualen Anteil an erfolgreichen Systembewertungen an.

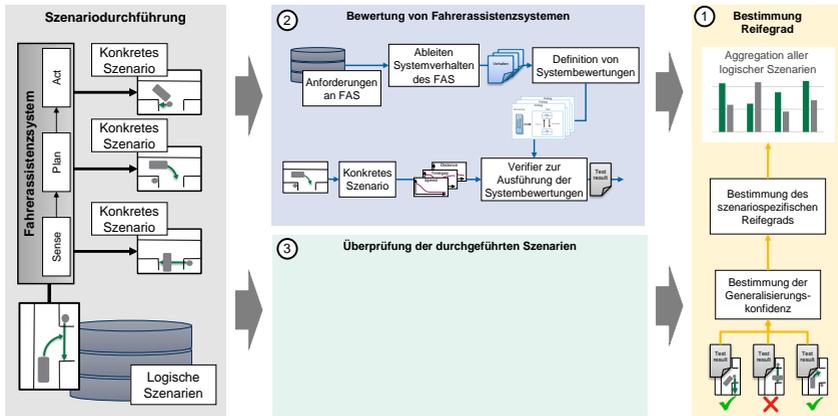


Abbildung 4.13: Vervollständigung des Testkonzepts um die Bewertung Fahrerassistenzsystemen in Fahrscenarien

4.3 Validierung von simulierten Szenarien

Im aktuellen Stand der Wissenschaft existieren keine Ansätze zur Ableitung von konkreten Szenarien, für die formal gezeigt wurde, dass sie nur valide Szenarien erzeugen. Aus diesem Grund beinhaltet das Testkonzept ein Modul zur Überprüfung, ob ein konkretes Szenario inhaltlich dem ursprünglichen logischen Szenario entspricht (vgl. Anforderung **Allg3**). Nach Definition 2.12 wird bei der Validierung überprüft, ob ein System das beabsichtigte Verhalten besitzt. Im Rahmen der vorliegenden Dissertation wird der Begriff „Validierung“ analog auch auf Szenarien übertragen und dabei definiert als:

Definition 4.4 (Szenariovalidierung) *Die Szenariovalidierung beschreibt den Prozess der Nachweiserbringung, dass ein konkretes Szenario die im logischen Szenario beschriebenen Inhalte besitzt. Beispiele für Inhalte sind die Szenerie, dynamische Objekte oder Aktionen und Events (vgl. Definition 2.33).*

Im ersten Schritt wird daher bestimmt, welche Teile eines Szenarios bei der Validierung betrachtet werden müssen (siehe Tabelle 4.3). Die Strukturierung des Inhalts von Szenarien erfolgt üblicherweise anhand des 6-Ebenen Modells (vgl. Abschnitt 2.3.3.2).

Die Ebenen L1 bis L3 beschreiben die verwendete Straße und deren Leitinfrastruktur. Elemente dieser Ebenen können zwar als Szenarioparameter variiert werden, aber sie bestimmen den statischen Aufbau des Szenarios. Daher sind sie während der Durchführung eines konkreten Szenarios fest und müssen bei der Validierung nicht berücksichtigt werden.

Manöver und Interaktionen, die zwischen den Teilnehmern des Szenarios stattfinden sind auf Ebene L4 beschrieben. Da die Manöver und Interaktionen in Abhängigkeit der Parametrisierung variieren können (vgl. Abschnitt 3.3.2), ist die Ebene L4 integraler Bestandteil der Szenariovalidierung.

Die Umwelteinflüsse auf Ebene L5 sind ebenfalls fest definierte Szenarioparameter der Umgebungssimulation und werden daher während der Validierung nicht

Ebene	Änderung während Szenariodurchlauf möglich?	Änderung erfolgt in Abhängigkeit des Szenarioverlaufs?	Für Validierung relevant
Straßenlevel (L1)	Nein	Nein	Nein
Leitinfrastruktur (L2)	Nein	Nein	Nein
Temporäre Beeinflussungen L1 und L2 (L3)	Nein	Nein	Nein
Bewegliche Objekte (L4)	Ja	Ja	Ja
Umweltbedingungen (L5)	Ja	Nein	Nein
Digitale Informationen (L6)	Ja	Ja	Ja

Tabelle 4.3: Ergebnis der Evaluierung, welche Ebenen des 6-Ebenen-Modells für die Validierung betrachtet werden müssen

berücksichtigt. Digitale Informationen auf der Ebene L6, wie beispielsweise Ampeln oder die Kommunikation zwischen Fahrzeugen und der Infrastruktur, sind abhängig von den Elementen auf der Ebene L4 und können dadurch ebenfalls variieren. Da die notwendige Technologie für Kommunikationen zwischen Fahrzeugen und der Infrastruktur, weder in Simulationsumgebungen noch in Fahrerassistenzsystemen verfügbar ist, werden während der Validierung nur Reaktionen auf Lichtsignalanlagen wie Ampeln berücksichtigt.

Während der Szenariovalidierung wird daher betrachtet, ob die spezifizierten Fahrmanöver und Interaktionen der Ebene L4, sowie die Reaktionen auf der Ebene L6 vorhanden sind und auch zeitlich korrekt durchgeführt werden. Zur

besseren Lesbarkeit wird im Folgenden nur noch von Fahrmanövern gesprochen. Die Validierung der Szenarien lässt sich in drei Schritte unterteilen:

1. Die Erkennung der durchgeführten Fahrmanöver in einem konkreten Szenario (siehe Kapitel 4.3.1).
2. Die Zusammenfassung der erkannten Manöver zu einer Beschreibung, die einen Vergleich der Szenarien ermöglicht (siehe Kapitel 4.3.2).
3. Der Vergleich der Beschreibung des konkreten Szenario mit der Beschreibung des logischen Szenarios (siehe Kapitel 4.3.2).

4.3.1 Erkennung von Fahrmanövern

Grundlage für die Erkennung von Fahrmanövern ist das Wissen darüber, welche Fahrmanöver in der ODD auftreten können. Dies wurde für den urbanen Raum bereits in verschiedenen wissenschaftlichen Arbeiten untersucht [64, 88, 142]. Die zu erkennenden Fahrmanöver werden in einem Manöverkatalog zusammengefasst, der aus den Arbeiten von [64, 88, 142] zusammengebaut wird (siehe Tabelle 4.4).

Fahrerassistenzsysteme unterscheiden sich darin, welchen Teil der Fahrzeugführung sie übernehmen und auf welche Elemente der ODD sie reagieren (vgl. Abschnitt 2.1). Daher sind nicht alle Fahrmanöver aus dem erarbeiteten Manöverkatalog für jedes Fahrerassistenzsystem relevant und müssen für eine Validierung erkannt werden. Für ein prädiktives ACC-System können, neben den Fahrmanövern „Freifahrt“, „Objekt folgen“ und „Objekt annähern“, beispielsweise die Manöver „Objekt passieren“, „Geschwindigkeitslimit annähern“ oder „Kreisverkehr überfahren“ relevant sein (vgl. Abschnitt 2.3.3).

	Fahrzeugzustand bezogene Manöver	Infrastruktur bezogene Manöver	Objekt bezogene Manöver
Longitudinale Manöver	<ul style="list-style-type: none"> - Beschleunigen (Sonderfall Losfahren) - Bremsen (Sonderfall Anhalten) - Geschwindigkeit halten (Sonderfall Stillstand) - Rückwärts fahren 	<ul style="list-style-type: none"> - Infrastrukturelement annähern - Infrastrukturelement überfahren (Beispiel Infrastrukturelement: Fußgängerüberweg, Kreuzung, Geschwindigkeitsbegrenzung, Kreisverkehr, Ampel, Kurve) 	<ul style="list-style-type: none"> - Objekt folgen - Objekt annähern - Freifahrt - Objekt passieren - Objekt kreuzen
Laterale Manöver	<ul style="list-style-type: none"> - Lenkeingriff - Kein Lenkeingriff 	<ul style="list-style-type: none"> - Spur folgen - Spur wechseln (links/rechts) 	
Kombinierte Manöver		<ul style="list-style-type: none"> - Abbiegen (links/rechts) - U-Turn - Einparken - Ausparken 	<ul style="list-style-type: none"> - Objekt überholen - Einscheren - Ausscheren

Tabelle 4.4: Fahrmanöverkatalog mit Fahrmanövern nach [142] [64] [88]

Da die Szenariovalidierung für verschiedene Fahrerassistenzsysteme eingesetzt werden soll (vgl. Anforderung **Val2**), muss die Liste an Fahrmanövern entsprechend strukturiert werden, damit die relevanten Fahrmanöver ausgewählt werden können. Um eine Strukturierung des Manöverkatalogs zu erreichen, wird eine Klassifikation der Fahrmanöver nach zwei Kriterien durchgeführt. Mit dem ersten Kriterium werden die Fahrmanöver in longitudinale, laterale oder longitudinal und lateral kombinierte Fahrmanöver unterteilt. Darüber hinaus werden sie dahingehend klassifiziert, ob sie sich auf den eigenen Fahrzeugzustand, die Infrastruktur oder Objekte im umgebenden Verkehr beziehen [88]. Aus dem so strukturierten Manöverkatalog können die relevanten Fahrmanöver eines FAS für die Erkennung ausgewählt werden.

Betrachtet man den Manöverkatalog in Tabelle 4.4, stellt man fest, dass sich Fahrmanöver wie „Beschleunigen“ und „Geschwindigkeit halten“ oder „Spur halten“ und „Spur wechseln“ gegenseitig ausschließen. Sie können nicht gleichzeitig auftreten und werden zu einem Manövertyp zusammengefasst.

Definition 4.5 (Manövertyp) *Ein Manövertyp umfasst eine Menge an konträren Manövern, welche sich implizit untereinander ausschließen, sowie logisch mit einander verknüpft sind.*

Für die eigentliche Erkennung, werden verschiedene Ansätze zur Identifikation von Fahrmanövern erarbeitet und evaluiert.

4.3.1.1 Regelbasierter Ansatz

Der erste Ansatz arbeitet regelbasiert und beinhaltet die Modellierung der Fahrmanöver auf Basis von physikalischen Zusammenhängen³. Zur Erkennung der Fahrmanöver in simulierten Szenarien, wird pro Manövertyp ein Erkennungsmodell erstellt, welches das Auftreten der enthaltenen Manöver überprüft (siehe Tabellen A.5-A.6 im Anhang).

³ Veröffentlicht in [King et al., 2017]

Die longitudinalen Objekt bezogenen Manöver „Freifahrt“, „Objekt folgen“ und „Objekt annähern“ können beispielsweise ausschließlich mit den Geschwindigkeiten und Beschleunigungen der Fahrzeuge, sowie dem Abstand der Fahrzeuge modelliert werden. Ausschlaggebende Parameter für die Manöverbestimmung sind die Schwellwerte für die Differenzgeschwindigkeit v_{th} sowie den Abstand d_{th} . Die Schwellwerte für die Modellierung müssen empirisch ermittelt werden. Die longitudinalen Objektbezogenen-bezogenen Manöver werden modelliert als:

Freifahrt:

$$\begin{aligned} Cruise := & (d > 0) \wedge ((d > d_{th}) \vee ((d < d_{th}) \\ & \wedge ((v_{Obj} - v_{FHg}) > 0) \wedge ((a_{Obj} - a_{FHg}) > 0))) \end{aligned}$$

Objekt annähern: (4.15)

$$Approach := (d < d_{th}) \wedge ((v_{FHg} - v_{Obj}) > v_{th})$$

Objekt folgen:

$$Follow := (d < d_{th}) \wedge (|(v_{FHg} - v_{Obj})| < v_{th})$$

Das Manöver „Objekt kreuzen“ ist so modelliert, dass sich die Trajektorien von zwei Verkehrsteilnehmern sowohl räumlich als auch zeitlich kreuzen (vgl. Algorithmus 1). Die Entscheidung, ob sich die Fahrzeuge auch zeitlich gekreuzt haben, wird anhand der Post-Enchroachment-Time getroffen (vgl. Definition 2.40). Damit ergibt sich die Modellierung für das Manöver:

Objekt kreuzen:

$$(Trajektorien_kreuzen = True) \wedge (0 < PET < PET_{max}) \quad (4.16)$$

Andere Fahrmanöver können auch als Kombination von Fahrmanövern modelliert werden. Das Fahrmanöver „Überholen“ ist beispielsweise die Aneinanderreihung von der Manöver „Spurwechsel nach links“, „Passieren“ und „Spurwechsel nach rechts“. Um dies zu berücksichtigen, können auch Kombination von Fahrmanövern und deren Historie zur Modellierung eines Fahrmanövers verwendet werden.

4.3.1.2 KI-basierter Ansatz

Alternativ wird ein weiterer Ansatz zur Identifikation von Fahrmanövern entwickelt, der künstliche neuronale Netze (KNN) für die Erkennung verwendet. Die Verwendung von KNN hat den Vorteil, dass keine aufwändige manuelle Modellierung und Parametrisierung der Fahrmanöver notwendig ist. Analog zum regelbasierten Ansatz wird jeder Manövertyp mit einem KNN untersucht. Zu jedem Zeitschritt beinhaltet der Ausgangsvektor des neuronalen Netzes die Wahrscheinlichkeiten für die Fahrmanöver eines Manövertyps (siehe Abbildung 4.14). Dafür

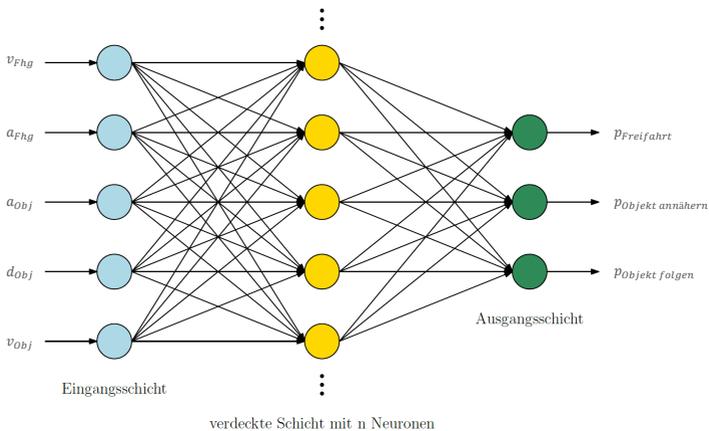


Abbildung 4.14: Konzept zur Identifikation der Manöver „Freifahrt“, „Objekt folgen“ und „Objekt annähern“ mittels künstlicher neuronaler Netze

stehen grundsätzlich verschiedene Netzarchitekturen, wie Feedforward (FNN) oder Recurrent NN (RNN), zur Verfügung, die für bestimmte Anwendungen Vor- und Nachteile besitzen (siehe Tabelle 4.5). Mit welcher Genauigkeit ein trainiertes KNN seine Aufgabe erledigt, hängt maßgeblich von der Qualität der Trainingsdaten ab. Die Qualität bemisst sich dabei unter anderem an der Anzahl an Trainingsdaten und in der Genauigkeit der dazugehörigen Label.

Kriterium	Feedforward NN	Recurrent NN
Notwendige Menge an Trainingsdaten	gering	hoch
Trainingsaufwand	gering	hoch
Zeitliches Gedächtnis	nein	ja

Tabelle 4.5: Vergleich der verschiedenen Netzarchitekturen hinsichtlich relevanter Kriterien für die Erkennung von Fahrmanövern

Für die Erkennung von Fahrmanövern gibt es keine öffentlich verfügbaren Datensätze mit gelabelten Fahrmanövern. Aus diesem Grund müssen die Trainingsdaten in einer Simulationsumgebung manuell erzeugt werden. Die Label für die einzelnen Zeitschritte werden mit dem regelbasierten Ansatz erzeugt und mit den Simulationsdaten zu Trainingsdaten verknüpft. Dadurch kann die Genauigkeit der Manövererkennung mittels KNN maximal der des regelbasierten Ansatzes entsprechen. Da die Trainingsdaten in einem manuellem Prozess erzeugt werden müssen, ist die Anzahl notwendiger Trainingsdaten ein maßgebliches Entscheidungskriterium bei der Auswahl der Netzarchitektur.

Rekurrente neuronale Netze besitzen im Gegensatz zu FNN Rückführungen, womit ein Art Gedächtnis aufgebaut werden kann. Für die Erkennung von Fahrmanövern, die durch eine zeitliche Abfolge von anderen Manövern entstehen, wie beispielsweise das Manöver „Überholen“, ist eine solches Gedächtnis eine Voraussetzung für die Erkennung. Da der Trainingsaufwand und die benötigte Anzahl an Trainingsdaten für FNN geringer ist als für RNN, wird für die Erkennung von Fahrmanövern, die kein zeitliches Gedächtnis benötigen, mit FNN durchgeführt.

4.3.1.3 Vergleich der Ansätze

Um die Ansätze vergleichen zu können, wurden beide Ansätze für die Erkennung von beispielhaften longitudinalen („Objekt folgen“, „Objekt annähern“ und „Freifahrt“) und lateralen („Spur folgen“ und „Spur wechseln“) Fahrmanövern umgesetzt. Der regelbasierte Ansatz verwendet die Modellierung aus Gleichung

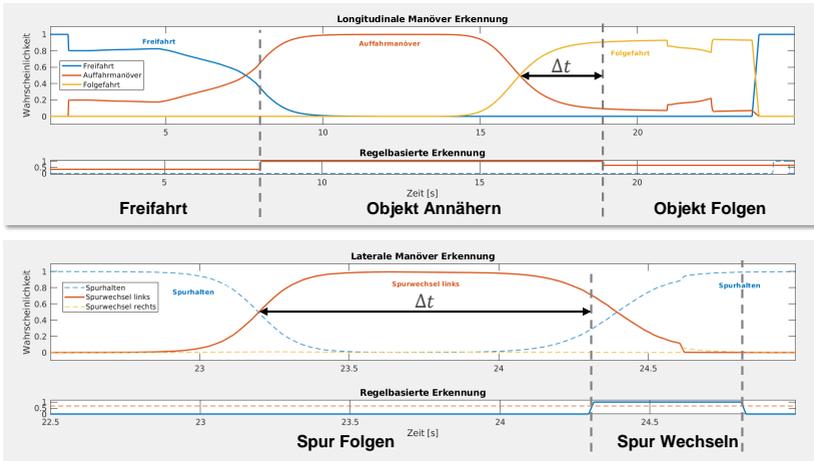


Abbildung 4.15: Vergleich der erkannten Fahrmanöver zwischen einem Ansatz mit einem Feedforward KNN (oben) und dem regelbasierten Modell (unten). Der obere Vergleich zeigt die longitudinalen Fahrmanöver und der Untere die lateralen Fahrmanöver.

4.15 und die gewählten Netzarchitekturen und Hyperparameter sind im Anhang in der Tabelle A.7 dargestellt. Die Evaluation der longitudinalen Fahrmanöver erfolgt anhand eines Szenarios, in dem sich das betrachtete Fahrzeug zunächst in der Freifahrt befindet und anschließend auf ein langsames Fahrzeug auffährt und diesem Fahrzeug folgt. Als Szenarioparameter werden die Startgeschwindigkeiten $v_{Fzg,Start}$ und v_{Obj} der beiden Fahrzeuge variiert. Für die lateralen Manöver wird ein Szenario verwendet, das einen Spurwechsel von der rechten auf die linke Spur bei unterschiedlichen Geschwindigkeiten beinhaltet. Die Szenarien werden in der im Kapitel 3.3.1 beschriebenen Simulationsumgebung simuliert und die Längs- und Querführung des Fahrzeugs erfolgt in beiden Szenarien durch ein Fahrermodell der Simulationsumgebung.

Der Vergleich der beiden Ansätze zeigt, dass Änderungen im Fahrmanöver vom KNN früher erkannt werden können, als beim regelbasierten Ansatz (siehe Abbildung 4.15). Der Grund dafür ist, dass das KNN Wahrscheinlichkeiten für die einzelnen Fahrmanöver ausgibt und das Fahrmanöver mit der höchsten Wahrscheinlichkeit gewählt wird. Der Zeitunterschied Δt ist dabei für laterale Fahr-

manöver, wie zum Beispiel einen Spurwechsel, größer als bei longitudinalen Manövern. Das KNN erkennt bereits während der Annäherung an die Nachbarspur einen Spurwechsel, während der regelbasierte Ansatz, bedingt durch die gewählte Modellierung, nur abgeschlossene Spurwechsel erkennt.

Für die Überprüfung, ob ein Fahrmanöver durchgeführt wurde, kann diese frühere Erkennung allerdings zu falschen Ergebnissen führen. Ein angefangener und dann abgebrochene Spurwechsel kann durch das trainierte KNN fälschlicherweise als Spurwechsel erkannt werden. Dieses Verhalten tritt auf, da in dem verwendeten Trainingsdatensatz keine abgebrochenen Spurwechsel enthalten waren. Um das Problem zu beheben, muss der Trainingsdatensatz um Trainingsdaten erweitert werden, die abgebrochene Spurwechsel beinhalten.

Unabhängig davon besitzt der KNN-basierte Ansatz die Limitierung, dass er, bedingt durch das Fehlen unabhängiger Trainingsdaten, lediglich den regelbasierten Ansatz nachbildet. Dennoch können grundsätzlich beide Ansätze für die Erkennung von Fahrmanövern eingesetzt werden.

Die Validität der Ergebnisse muss insbesondere im Hinblick auf das Training des Netzes und den Umfang des Vergleichs kritisch diskutiert werden. Eine mögliche Einschränkung der Validität für einen solchen Vergleich kann grundsätzlich sein, dass das neuronale Netz andere Regeln für die Erkennung der Fahrmanöver lernt, als sie im regelbasierten Ansatz verwendet werden. Da die Trainingsdaten für den Vergleich mit dem regelbasierten Ansatz erzeugt wurden, stellt dies in diesem Fall keine Einschränkung dar. Die verwendete Datengrundlage stellt einen weiteren Einflussfaktor dar. Die Ergebnisse der Manövererkennung können sowohl von dem betrachteten logischem Szenario als auch von dem zu erkennenden Fahrmanöver abhängen. Da für den Vergleich nur exemplarisch drei longitudinale und zwei laterale Fahrmanöver betrachtet werden und jeweils nur ein logisches Szenario simuliert wird, können aus den Ergebnissen keine verallgemeinerbaren Aussagen für alle Fahrmanöver abgeleitet werden. Dies sollte Bestandteil zukünftiger Forschungsarbeiten sein (siehe Abschnitt 7.2).

4.3.2 Beschreibung und Vergleich der Szenarien

4.3.2.1 Beschreibung mittels Manövermatrizen

Nach der Erkennung in den simulierten Daten, müssen die Fahrmanöver in der korrekten zeitlichen Abfolge zu einer Szenariobeschreibung zusammengefasst werden. Beschreibungssprachen wie OpenScenario erreichen eine zeitliche Abstraktion von Szenarien durch die Aneinanderreihung von Akten (vgl. Abschnitt 2.3.3.1). Jeder Akt wird dabei durch genau ein Manöver pro Manövertyp charakterisiert. Ändert sich ein Manöver, dann erfolgt der Übergang in den nächsten Akt.

Manövertyp		Fahrzeugzustand bezogene Manöver		
ZustandLong		Beschleunigen (a) Losfahren	Geschwindigkeit halten Stillstand	Bremsen (b) Halten
Fahrrichtung		Vorwärts fahren (c)		Rückwärts fahren (d)
Lenken		Kein Lenkeingriff (e)		Lenkeingriff (f)
Manövertyp		Infrastruktur bezogene Manöver		
Infrastruktur		N/A	Infrastrukturelement annähern	Infrastrukturelement überfahren
Abbiegen		N/A		Abbiegen
Spurmanöver		Spur folgen		Spur wechseln
U-Turn		N/A		U-Turn
Parken		N/A	Ausparken	Einparken
Manövertyp		Objekt bezogene Manöver		
Objektmanöver		Objekt folgen	Objekt annähern	Freifahrt
Passieren		N/A		Objekt passieren
Kreuzen		N/A		Objekt kreuzen
Überholen		N/A		Objekt überholen
Einscheren		N/A		Einscheren
Ausscheren		N/A		Ausscheren

Abbildung 4.16: Anordnung der Fahrmanöver in einem morphologischen Kasten zur Beschreibung von Akten

Da jeder Akt durch genau ein Manöver pro Manövertyp bestimmt wird, werden die Fahrmanöver aus Tabelle 4.4 in einem morphologischen Kasten angeordnet (siehe Abbildung 4.16). Alle Fahrmanöver, die in einer Zeile stehen, gehören zum selben Manövertyp. Für Fahrmanöver, die kein konträres Manöver besitzen, wird das Manöver „Not Available (N/A)“ als konträres Manöver im morphologischen Kasten hinzugefügt. Akte können damit durch eine eindeutige Konfiguration aus dem morphologischen Kasten beschrieben werden.

Das Ergebnis der Manövererkennung sind Sequenzen von Fahrmanövern für jeden Manövertyp. Akte werden formalisiert durch einen Manövervektor beschrieben. Eine beispielhafte Manöversequenz für drei Manövertypen ist in Abbildung 4.17 dargestellt. Für den Manövertyp *ZustandLong* wird im Zeitraum $[t_0, t_1]$ das Ma-

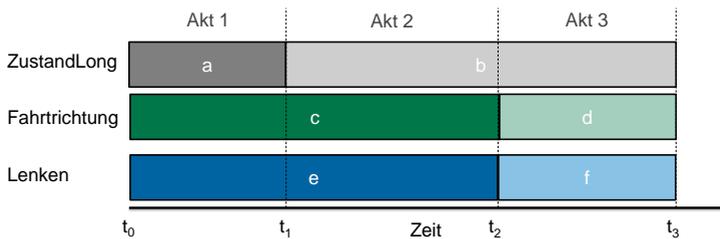


Abbildung 4.17: Einfache Manöversequenz für drei Manövertypen

növer *a* aufgeführt, ab Zeitpunkt t_1 das Manöver *b*. Für Manövertyp *Fahrtrichtung* und *Lenken* findet zum Zeitpunkt t_2 ein Wechsel der Fahrmanöver statt. Zu jedem Zeitpunkt $t \in [t_0, t_3]$ ist der Manövervektor bestimmt als

$$\vec{m} = \begin{cases} \vec{m}_{01} & t_0 \leq t \leq t_1 \\ \vec{m}_{12} & t_1 < t \leq t_2 \\ \vec{m}_{23} & t_2 < t \leq t_3 \end{cases} \quad (4.17)$$

mit

$$\vec{m}_{01} = \begin{pmatrix} a \\ c \\ e \end{pmatrix}, \quad \vec{m}_{12} = \begin{pmatrix} b \\ c \\ e \end{pmatrix}, \quad \vec{m}_{23} = \begin{pmatrix} b \\ d \\ f \end{pmatrix} \quad (4.18)$$

Weiter kann somit die Manövermatrix M erstellt werden, welche das Szenario für einen Teilnehmer repräsentiert. Für eine vollständige Beschreibung des Szenarios, muss für jeden Teilnehmer eine Manövermatrix erstellt werden. Die Spalten der Matrix bestehen jeweils aus den Manövervektoren. Die Ordnung der Manövermatrix ergibt sich durch die Anzahl der Akte, sowie der Anzahl der betrachteten Manövertypen. Für das genannte Beispiel ergibt sich folgende 3×3 Manövermatrix:

$$M = (\vec{m}_{01} \quad \vec{m}_{12} \quad \vec{m}_{23}) = \begin{pmatrix} a & b & b \\ c & d & d \\ e & e & f \end{pmatrix}. \quad (4.19)$$

Da durch jeden Manöverwechsel ein neuer Akt beginnt und sich damit die Ordnung der Manövermatrix ändert, führen schon kleine Variationen im Szenarioablauf zu unterschiedlichen Manövermatrizen.

Erfolgt im Beispiel aus Abbildung 4.17, bedingt durch zeitliche Variationen oder Ungenauigkeiten der Manövererkennung, der Übergang des Manövertyps *Fahrtrichtung* nicht zeitgleich mit dem Manövertyp *Lenken*, dann wird das Szenario mit vier Akten beschrieben (siehe Abbildung 4.18). Damit ergibt sich eine 3×4 Manövermatrix für das konkreten Szenario:

$$M_{var} = (\vec{m}_{01} \quad \vec{m}_{12} \quad \vec{m}_{23} \quad \vec{m}_{34}) = \begin{pmatrix} a & b & b & b \\ c & c & d & d \\ e & e & e & f \end{pmatrix}. \quad (4.20)$$

Um robust gegenüber kurzer Variationen zu sein, werden für die Szenarioerkennung daher nur Akte betrachtet, welche eine minimale Zeit $t_{Akt,min}$ andauern. Ist die Zeit $t_{Akt3} = t_3 - t_2$ aus dem obigen Beispiel kürzer als $t_{Akt,min}$, dann wird

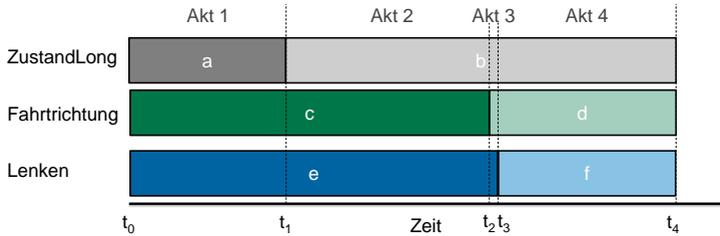


Abbildung 4.18: Geringe zeitliche Variationen in den Manöversequenz führen zur Bildung neuer Akte

das Szenario wieder durch die Manövermatrix aus Gleichung 4.19 beschrieben werden.

Da die Szenariovalidierung für verschiedene logische Szenarien eingesetzt werden soll (vgl. Anforderung **Val3**), erfolgt die Bestimmung der minimalen Aktlänge auf Basis eines Datensatzes, der verschiedene logische Szenarien enthält. Dafür wird eine Szenarioerkennung für eine Kreuzung des INTERACTION-Datensatz durchgeführt⁴ und eine Verteilung der Aktlängen bestimmt (siehe Abbildung 4.19).

Die Ergebnisse der Untersuchung zeigen, dass, in dem betrachteten Datensatz, ungefähr 30 Prozent der Akte kürzer sind als zwei Sekunden. Diese kurzen Akte führen zu einer hohen Anzahl an Aktwechslern, die dann zu unterschiedlichen logischen Szenarien führen. In Abbildung 4.20 ist dargestellt, welcher Zusammenhang zwischen der Anzahl der Akte und der Anzahl an erkannten logischen Szenarien besteht. Besitzt ein konkretes Szenario beispielsweise nur drei Akte, dann führt ein konkretes Szenarien nur zu 0.135 logischen Szenarien (70 logische Szenarien aus 518 konkreten Szenarien). Mit zunehmender Anzahl an Akten steigt diese Zahl, bis bei einer Anzahl von acht Akten jedes konkrete Szenario auch ein neues logisches Szenario erzeugt. Konkrete Szenarien mit zwei Akten treten in dem betrachteten Datensatz nur selten auf, daher führen sie ebenfalls jeweils zu einem neuen logischen Szenario.

⁴ Veröffentlicht in [King et al., 2021]

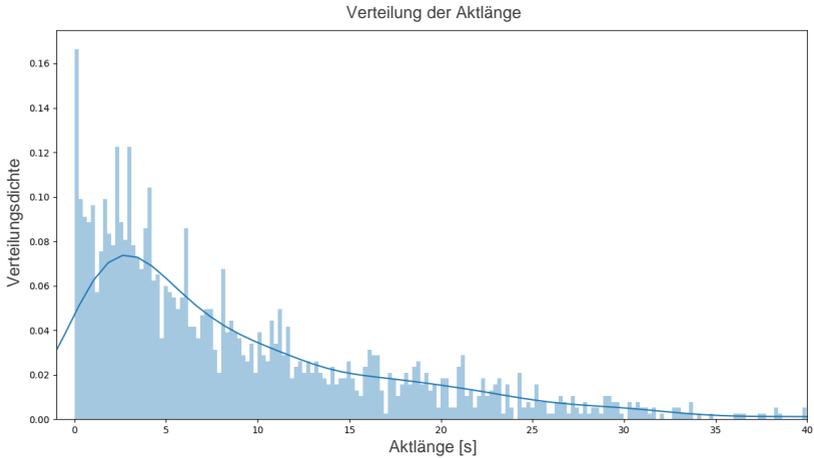


Abbildung 4.19: Verteilung der Aktlängen für die Szenarioerkennung der Kreuzung *DR_USA_Intersection_EP* aus dem INTERACTION Datensatz

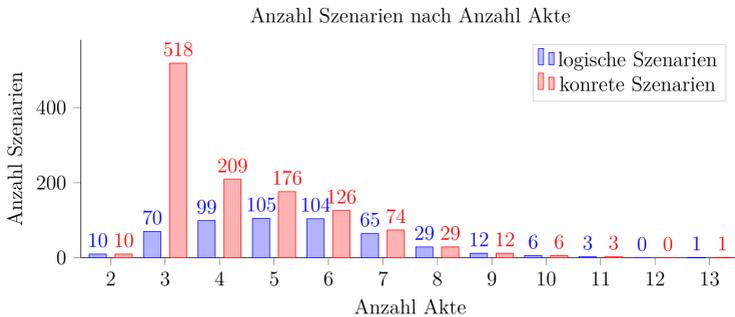


Abbildung 4.20: Einfluss der Anzahl an Akten auf die Anzahl an erkannten logischen Szenarien

Im nächsten Schritt wird daher mittels Sensitivitätsanalyse untersucht, welcher Manövertyp die häufigsten Aktwechsel erzeugt. Dafür wird für jeden Manövertyp die Szenarioerkennung einmal mit und einmal ohne den Manövertyp ausgeführt und untersucht, wie viele logische Szenarien durch die Betrachtung des Manövertyps hinzukommen. Gleichzeitig wird auch eine minimale Aktdauer verwendet, um den quantitativen Einfluss von minimalen Aktdauern auf die Erkennung zu

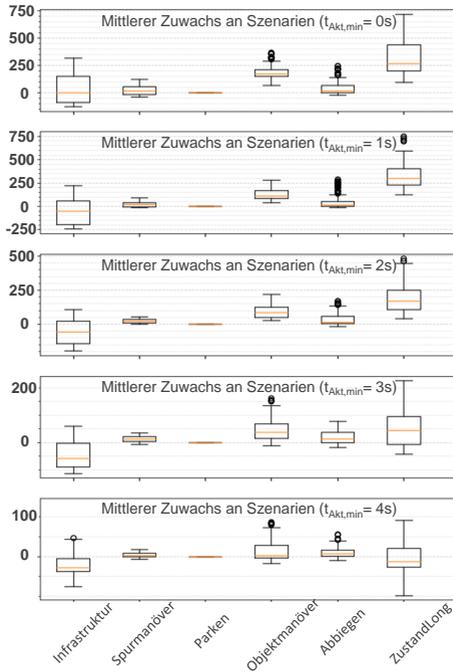


Abbildung 4.21: Ergebnisse der Analyse, welchen quantitativen Einfluss der Manövertyp und die minimale Aktdauer auf die Anzahl an extrahierten logischen Szenarien hat

untersuchen (siehe Abbildung 4.21). Dafür werden minimale Aktdauern zwischen null und vier Sekunden betrachtet.

Die Analyse zeigt, dass insbesondere Übergänge zwischen den Fahrzeugzustand bezogenen Manövern „Beschleunigen“, „Geschwindigkeit halten“ und „Bremsen“ zu dem höchsten Anstieg an logischen Szenarien führen. Weiterhin zeigen die Untersuchungen, dass erst ab einer minimalen Aktdlänge von drei Sekunden ein signifikanter quantitativer Effekt sichtbar wird. Die qualitative Untersuchung der erkannten Szenarien hat allerdings ergeben, dass durch längere minimale Aktdlängen kurze Events entfernt werden, wie beispielsweise eine plötzliche Bremsung auf ein einscherendes Fahrzeug. Daher lässt sich keine allgemeingültige Aussage

über eine optimale minimale Aktlänge treffen, da die quantitativen Einsparungen gegen die qualitativen Einflüsse abgewogen werden müssen.

4.3.2.2 Merkmalsbasierte Beschreibung

Eine weitere Möglichkeit Szenarien zu Beschreiben besteht darin, die Dimension der Zeit vollständig zu abstrahieren. Die Fahrscenarien werden in diesem Fall ausschließlich durch das Vorhandensein von definierten Merkmalen beschrieben. Bei der Validierung wird dann schlussendlich nicht mehr die Äquivalenz zwischen den Manövermatrix eines konkreten und logischen Szenarios überprüft, sondern das Vorhandensein bestimmter relevanter Merkmale überprüft, wie beispielsweise das Auftreten eines kreuzender Fußgänger oder das Warten an einer roten Ampel.

Um diese Merkmale aus den Manövermatrizen zu extrahieren und in eine Beschreibungsform zu bringen, mit der ein Vergleich zwischen konkretem und logischem Szenario möglich ist, wird eine Abbildungsfunktion verwendet, die eine Manövermatrix in einen Merkmalsvektor mit fester Länge umwandelt. Die verwendete Abbildungsfunktion ist im Anhang im Abschnitt A.4 dargestellt.

Beispielsweise werden in der Abbildungsfunktion die vier Objekttypen PKW, Bus, Fußgänger und Fahrradfahrer betrachtet und mit zwei Bit codiert.

- PKW: codiert als '00'
- Bus: codiert als '01'
- Fußgänger: codiert als '10'
- Fahrradfahrer: codiert als '11'

Bis zu vier kreuzende Objekte werden dann in dem Merkmalsvektor in einem Byte zusammengefasst. Die Objekte werden Bit-weise in das Datenfeld geschrieben und nacheinander durch einen Bitshift nach links angehängt. Ein Bitshit nach links um n Bits bedeutet eine Multiplikation des Werts um 2^n . Die Dekodierung wird dementsprechend durch Bitshifts nach rechts realisiert, also einer Division durch

2^n und abschließenden Abrunden. Wenn mehr Objekte oder weitere Objekttypen betrachtet werden sollen, dann kann dies durch die Anpassung der Feldgrößen realisiert werden.

Die Funktionsweise der Abbildungsfunktion soll an einem kurzen Beispiel dargestellt werden. In einem Szenario wird ein betrachtetes Fahrzeug von drei Objekten gekreuzt, einem Auto (Typ = 0), Fußgänger (Typ = 2) und Fahrradfahrer (Typ = 3). Damit berechnet sich der Ausgabewert durch die Abbildungsfunktion folgendermaßen

$$\text{Value} = 0 * 2^0 + 2 * 2^2 + 3 * 2^4 = 0 + 8 + 48 = 56 = 0x38 \quad (4.21)$$

Die Abbildungsfunktion würde in dem Fall die Hexadezimalwerte „0x03“ (Anzahl der Objekte) und „0x38“ (Typ der Objekte) ausgeben. Für eine spätere Dekodierung ergibt sich bei einer Anzahl von drei Objekten und einem Objekttyp Wert von 56:

Typ vom Objekt 3:

$$\text{Typ}_3 = \lfloor \frac{56}{2^4} \rfloor = \lfloor \frac{56}{16} \rfloor = \lfloor 3,50 \rfloor = 3 \quad (4.22)$$

Typ vom Objekt 2: Objekt 3 vom Wert abziehen $56 - 3 * 2^4 = 56 - 48 = 8$

$$\text{Typ}_2 = \lfloor \frac{8}{2^2} \rfloor = \lfloor \frac{8}{4} \rfloor = 2 \quad (4.23)$$

Typ von Objekt 1: Objekt 2 vom Wert abziehen $8 - 2 * 2^2 = 0$

$$\text{Typ}_1 = \lfloor \frac{0}{2^0} \rfloor = \lfloor \frac{0}{1} \rfloor = 0 \quad (4.24)$$

Damit sind die decodierten Objekttypen: 0 (PKW), 2 (Fußgänger), 3 (Fahrradfahrer).

Bei dem merkmalsbasierten Ansatz nimmt die zeitliche Variabilität des Szenarioverlaufs keinen Einfluss mehr auf die Ergebnisse der Szenariobeschreibung und den Szenariovergleich. Im Vergleich zur Beschreibung mittels Manövermatrizen,

führt die Abstraktion allerdings auch zu einem Informationsverlust. Ob durch die Abstraktion relevante Informationen nicht mehr betrachtet werden, ist abhängig von dem Inhalt des logischen Szenarios. Wenn ein logisches Szenario explizit zeitliche Abfolgen beinhaltet, wie beispielsweise einen Überholvorgang vor dem Abbiegen, dann ist der merkmalsbasierte Ansatz nicht ausreichend. Für die Fahrscenarien, die durch Euro NCAP, die Verkehrsunfallforschung oder die Regelung R157 (vgl. Abschnitt 2.4.1) vorgegeben werden, ist dies nicht der Fall und der merkmalsbasierte Ansatz kann verwendet werden.

Es ist allerdings zu erwarten, dass die Zunahme an Verkehrsteilnehmern in logischen Szenarien dazu führt, dass die zeitlichen Abhängigkeiten zukünftig betrachtet werden müssen. Aus diesem Grund sollte der Vergleich von Szenarien mittels Manövermatrizen Inhalt weiterführender Forschungsarbeiten sein (siehe Abschnitt 7.2).

4.4 Zusammenfassung des Testkonzepts

Die vorgestellten Methoden und Ansätze ergeben kombiniert ein Konzept zum Testen von Fahrerassistenzsystemen in Fahrscenarien, mit dem der aktuelle Reifegrad eines Fahrerassistenzsystems bestimmt werden kann (siehe Abbildung 4.22).

Das Testkonzept umfasst die drei Module:

1. Bestimmung des Reifegrads
2. Bewertung des Fahrerassistenzsystems
3. Überprüfung der durchgeführten Szenarien

Im ersten Modul wird das gewünschte Systemverhalten von Fahrerassistenzsystemen aus Anforderungen abgeleitet und mittels Systembewertungen formalisiert. Diese formalisierten Systembewertungen werden dann mittels eines Verifiers genutzt, um das Fahrerassistenzsystem in konkreten Szenarien zu testen.

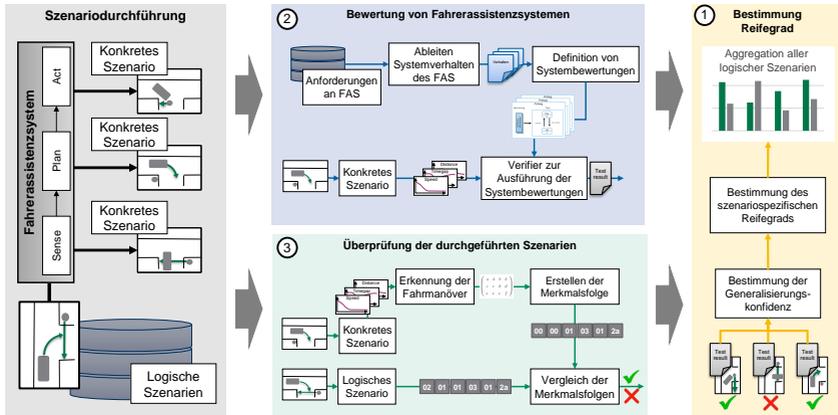


Abbildung 4.22: Konzept zum Testen von Fahrerassistenzsystemen in Fahrszenarien bestehend aus einer Bewertung der FAS, Validierung der Szenarien und der Bestimmung des Reifegrads

Parallel dazu werden die konkreten Szenarien im zweiten Modul validiert. Dabei wird untersucht, ob alle Merkmale aufgetreten sind, die für das logische Szenario ursprünglich spezifiziert wurden. Somit wird sichergestellt, dass die Ergebnisse der Systembewertung für die konkreten Szenarien auch für die Bestimmung des Reifegrads verwendet werden können.

Diese Bestimmung des Reifegrads wird im dritten Modul durchgeführt. Dafür wird sowohl für jedes logische Szenario als auch die gesamte ODD bestimmt, in welchem Umfang die Systemanforderungen durch das Fahrerassistenzsystem erfüllt werden. Grundlage dafür sind die Ergebnisse der Systembewertung aus dem ersten Modul und die Informationen über die Validität der konkreten Szenarien. Ergebnisse der Systembewertung für nicht valide konkrete Szenarien werden für die Bestimmung des Reifegrads nicht berücksichtigt.

5 Prototypische Umsetzung des Testkonzepts

5.1 Umsetzung in einer prototypischen SiL-Testumgebung

5.1.1 Aufbau der Testumgebung

Für die Umsetzung des vorgestellten Konzepts wird eine Testumgebung aufgebaut, die aus vier Komponenten besteht. Die verschiedenen Module des erarbeiteten Testkonzepts werden dabei jeweils in einer Komponente realisiert und in der Programmiersprache Python implementiert (siehe Abbildung 5.1).

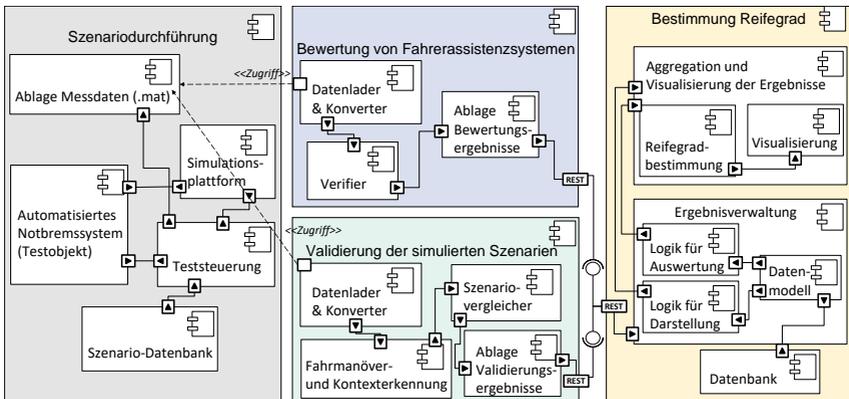


Abbildung 5.1: Komponentendiagramm für die prototypische Umsetzung des vorgestellten Konzepts zum Testen von Fahrerassistenzsystemen in Fahrscenarien

Die erste Komponente „Szenariodurchführung“ ist für die Ausführung des Testobjekts in verschiedenen logischen Szenarien in einer Simulationsumgebung zuständig (vgl. Abbildung 4.22). Als Testobjekt wird exemplarisch ein Notbremssystem betrachtet, welches im Abschnitt 5.1.2 detailliert beschrieben wird. Die Auswahl der logischen Szenarien und Szenarioparameter für die Testdurchführung wird im Abschnitt 5.1.3 vorgestellt. Die Szenariodurchführung erfolgt auf einem Desktop-Rechner¹ in einer Gesamtfahrzeugsimulation in Verbindung mit einer Teststeuerungssoftware². Um logische Szenarien für die Testdurchführung nutzen zu können, müssen sie in der Teststeuerungssoftware aufgebaut werden. Da die verwendete Software den Standard OpenScenario (vgl. Abschnitt 2.3.3.2) noch nicht vollständig unterstützt, verwendet sie dafür eine proprietäre Szenariobeschreibung. Als Ergebnis liefert die Szenariodurchführung Messdaten aus der Simulation, wie beispielsweise Geschwindigkeiten oder Trajektorien der Teilnehmer, sowie interne Signale des Notbremssystems. Diese Messdaten werden als *.mat*-Datei von der Teststeuerung auf einem Desktop-PC abgelegt, sodass sie von den Komponenten „Bewertung von Fahrerassistenzsystemen“ und „Validierung der simulierten Szenarien“ eingelesen und verarbeitet werden können.

Für die Bewertung des Testobjekts werden die Messdaten zunächst eingelesen und aus der proprietären Datenstruktur der Teststeuerungssoftware in die interne Datenstruktur des Verifiers konvertiert. Die Systembewertungen werden als Objekt der Klasse „*AssessmentClass*“ instanziiert und im Verifier ausgeführt (siehe Abbildung 5.2). Damit die Aktivierungs- und Testbedingungen auswertbar sind, werden sie als Funktionen modelliert. Im Algorithmus 2 ist der Pseudo-Code einer FCW-Bewertung dargestellt, die überprüft, ob das automatisierte Notbremssystem eine Frontkollisionswarnung anfordert. Die dazugehörige Implementierung ist im Anhang im Listing A.1 dargestellt.

¹ CPU: Intel Core i9-9900K (3.6 GHz), RAM: 32 GB, GPU: NVIDIA GeForce RTX 2080 Ti

² Simulationssoftware: dSPACE VEOS (https://www.dspace.com/en/inc/home/products/sw/simulation_software/veos.cfm), Teststeuerungssoftware: dSPACE ASM ModelDesk (https://www.dspace.com/en/inc/home/products/sw/automotive_simulation_models/produkte_asm/modeldesk.cfm)

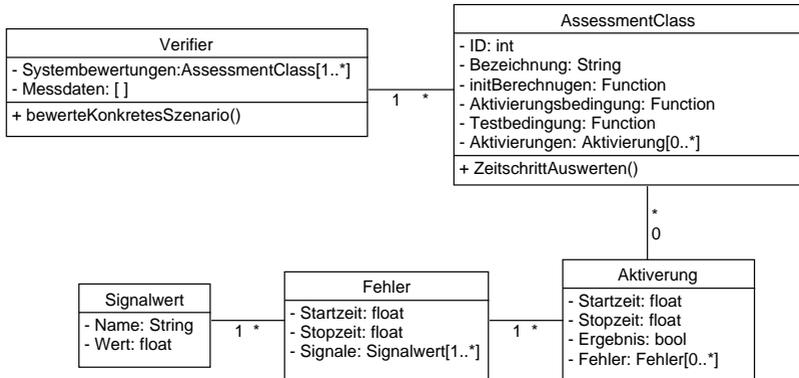


Abbildung 5.2: Klassendiagramm des Verifiers

Algorithmus 2 : Pseudocode von einer Bewertung die überprüft, ob das Notbremssystem eine Frontkollisionswarnung anfordert

for *Zeitschritt* in *Messdaten* **do**

```

  [obereSS, untereSS] = TtcLookupTable(vFzg);
  TtcvergleichswertS = InterpoliereTtc(obereSS, untereSS, vFzg);
  Überprüfe Aktivierungsbedingung
  if TtcMesswertS < TtcvergleichswertS && AEB != deaktivert then
    Überprüfe Testbedingung
    if FCW == aktiv then
      Zustandsautomat(passed);
    else
      log(FCW);
      Zustandsautomat(failed);
    end
  end
end
end

```

Da die TTC, bei der eine Warnung angefordert wird, von der Geschwindigkeit des Ego-Fahrzeugs abhängig ist (vgl. Systemanforderung **E2Anf2**), wird für diese

Bewertung eine Lookup-Tabelle mit Stützstellen, sowie eine Interpolation für die TTC zwischen den Stützstellen benötigt. Die Systembewertung deckt die beiden Bestehensbedingungen „AEBS_Acust_Warn = 1“ und „AEBS_Visual_Warn = 1“ aus dem vorgestellten anforderungsbasierten Testfall aus Kapitel 2.3.2.1 ab (vgl. Abbildung 2.10). Die Ergebnisse der Systembewertung werden über eine Representational State Transfer (REST)-Schnittstelle an die Komponente „Bestimmung des Reifegrads“ gesendet.

Nach dem Laden der Messdaten in der Komponente „Validierung der simulierten Szenarien“, erfolgt die Erkennung der Fahrmanöver und des Fahrkontexts. Da keine unabhängigen Trainingsdaten für den KI-basierten Ansatz zur Verfügung stehen, wird der regelbasierte Ansatz aus Abschnitt 4.3.1.1 für die Erkennung der Fahrmanöver verwendet. Darauf aufbauend wird im Anschluss der Vergleich der Szenarien durchgeführt und die Ergebnisse der Validierung über eine REST-Schnittstelle abgelegt.

Die Komponente „Bestimmung des Reifegrads“ wird mit dem REST-Framework Django³ aufgebaut. Django basiert auf dem Software-Entwurfsmuster „Model View Template“ (MVT) und stellt standardisierte Schnittstellen zur Verfügung, mit denen ein Zugriff über Hypertext Transfer Protocol (HTTP) erfolgen kann. Die zu speichernden Daten, wie beispielsweise Testergebnisse, Systemanforderungen oder Szenarien, werden in einem Datenbankmodell abgebildet (siehe Abbildung A.1) und technisch in einer SQL-Datenbank gespeichert. Um den Reifegrad nach Formel 4.6 bestimmen zu können, müssen die Testergebnisse auf der Ebene der konkreten Szenarien, logischen Szenarien und der gesamten ODD miteinander verknüpft werden. Die dafür notwendige Logik wird in unterschiedlichen Views realisiert. Durch die Erstellung von HTML-Templates können die aggregierten Daten für einen Nutzer visualisiert werden (siehe Abschnitt A.6.2).

³ <https://www.djangoproject.com/>

5.1.2 Automatisiertes Notbremssystem

In der Testumgebung wird ein Fahrerassistenzsystem verwendet, welches die Längsregelung eines Fahrzeugs übernimmt (vgl. Abschnitt 3.3.1). Zusätzlich zu einer Abstands- und Geschwindigkeitsregelung umfasst das FAS auch ein Subsystem für die Durchführung einer automatisierten Notbremsung zur Verhinderung von Frontkollisionen⁴. Für die prototypische Umsetzung wird nur das Notbremssystem als Testobjekt betrachtet. Die Querführung des Fahrzeugs übernimmt ein integriertes Fahrermodell in der Simulationsumgebung.

Die Systemarchitektur des implementierten Notbremssystems ist in Abbildung 5.3 dargestellt. Sie ist dabei entsprechend der Implementierung in drei Dekompositionsebenen unterteilt (vgl. Definition 2.9). Die Darstellung in Abbildung 5.3 beschränkt sich auf die Teile der Systemarchitektur, für die ihm Rahmen der prototypischen Umsetzung Systembewertungen definiert werden. Für das betrachtete Notbremssystem liegen keine Systemanforderungen vor, die Grundlage für die Entwicklung des Systems waren. Aus diesem Grund müssen exemplarische Anforderungen aus der Implementierung des Systems abgeleitet werden (siehe Abschnitt A.7.1).

Dafür wird angenommen, dass an das gesamte Notbremssystem die Anforderung gestellt wird, dass es Kollisionen mit anderen Verkehrsteilnehmern (VT), wie beispielsweise PKW, Fußgängern oder Fahrradfahrern, verhindert. Diese Anforderung wird dann auf die drei Subsysteme „Umfeld erkennen“, „Aktion planen“ und „Aktion umsetzen“ heruntergebrochen und konkretisiert. Das Notbremssystem erkennt Verkehrsteilnehmer, die sich vor dem Fahrzeug befinden und stuft diese als relevant ein, falls sie sich in der Fahrspur des Fahrzeugs befinden oder die Fahrspur kreuzen. Im Falle einer bevorstehenden Kollision (TTC zu relevantem VT unterschreitet einen Grenzwert) warnt das Notbremssystem den Fahrer mit einer Frontkollisionswarnung (FCW) oder führt eine Notbremsung (AEB) durch (siehe Abbildung 5.4).

⁴ Das FAS wird von der Firma dSpace bereitgestellt

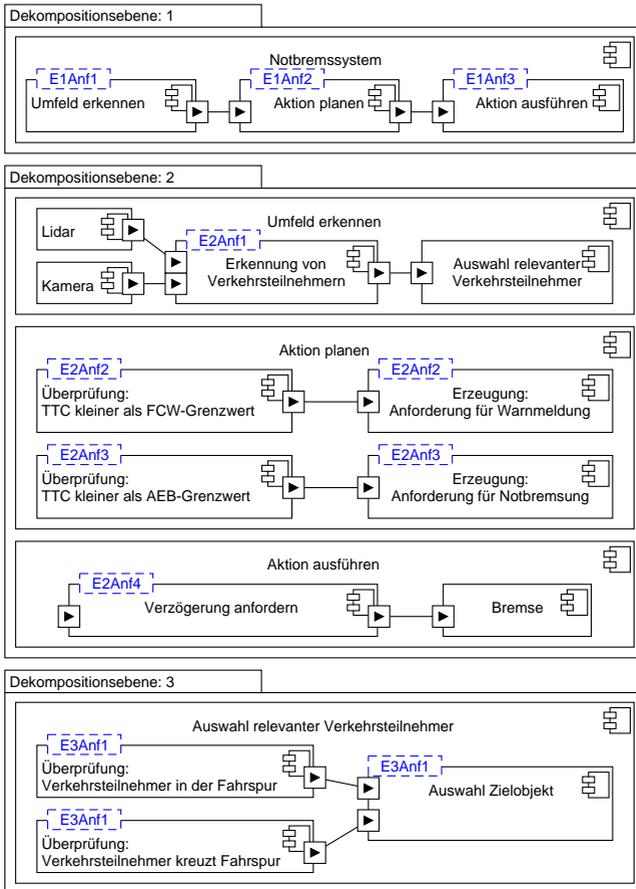


Abbildung 5.3: Systemarchitektur des verwendeten Fahrerassistenzsystems heruntergebrochen auf verschiedene Dekompositionsebenen und mit abgeleiteten Anforderungen

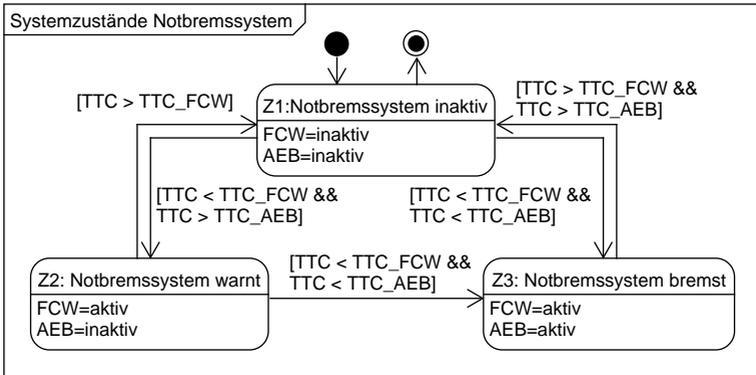


Abbildung 5.4: Zustandsdiagramm mit den Systemzuständen des automatisierten Notbremssystems

Die Bewertung des Notbremssystems erfolgt durch das in Kapitel 4.2.2 vorgestellte Bewertungskonzept. Für die Überprüfung der Systemanforderung wird zunächst das erwartete Systemverhalten abgeleitet (siehe Abschnitt A.7.2). Aus dem Systemverhalten werden Systembewertungen definiert, welche die Erfüllung je einer Anforderung in den simulierten Szenarien evaluieren (siehe Abschnitt A.7.3).

Für die Systemanforderung *E1Anf2* (vgl. Abschnitt A.7.1) wird keine Systembewertung definiert, da die Anforderung allgemein die Ausgabe einer Warnung oder Notbremsung fordert. Dies wird bereits mit den Systembewertungen für die Anforderungen *E2Anf2* und *E2Anf3* überprüft. Daher kann mit den Ergebnissen dieser Systembewertungen auch auf die Erfüllung der Anforderung *E1Anf2* geschlossen werden.

5.1.3 Logische Szenarien für die Testdurchführung

Für die Testdurchführung werden exemplarisch vier urbane Fahrscenarien ausgewählt, die häufige Unfallszenarien mit verschiedenen Verkehrsteilnehmern abdecken. Es werden jeweils zwei Unfallszenarien mit Fußgängern und zwei mit PKW betrachtet (siehe Abbildung 5.5).

Das erste logische Szenario (zukünftig auch mit *LS1* bezeichnet) wird aus der Klasse der *Überschreiten-Unfälle* abgeleitet und beschreibt ein Szenario, in dem es zu einem Unfall zwischen einem Fahrzeug und einem Fußgänger, der die Fahrbahn überschreitet, kommt [150]. Im zweiten Szenario (*LS2*) wird ein Abbiegevorgang nach links betrachtet, bei dem ein Fußgänger die Straße während des Abbiegevorgangs betritt. Damit gehört das *LS2* zur Klasse der *Abbiege-Unfälle*. Um einen *Abbiege-Unfall* handelt es sich, wenn ein Unfall zwischen einem Abbieger und einem aus gleicher oder entgegengesetzter Richtung kommenden Fußgänger auftritt [150]. Als drittes logisches Szenario (*LS3*) wird ein *Einbiegen/Kreuzen-Unfall* ausgewählt, bei dem es zu einem Unfall durch ein kreuzendes wartepflichtiges Fahrzeug und dem Vorfahrtberechtigten kommt [150]. Das automatisierte Notbremssystem wird in dem vorfahrtberechtigtem Fahrzeug getestet. Das vierte

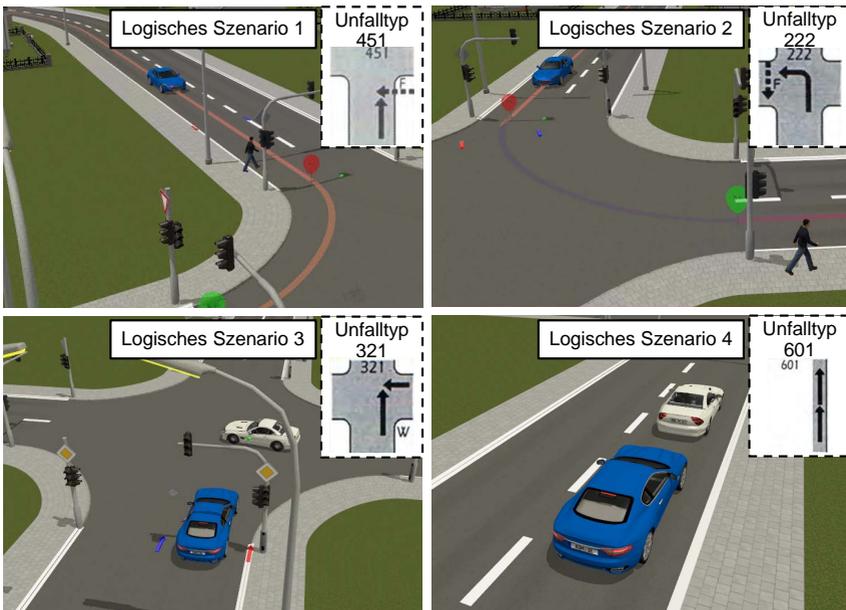


Abbildung 5.5: Visualisierung der ausgewählten Fahrscenarien für die prototypische Umsetzung mit Unfalltypenreferenz nach [150]. Der blaue PKW ist mit dem zu testenden automatisierten Notbremssystem ausgestattet.

logische Szenario (*LS4*) beschreibt einen *Unfall im Längsverkehr*, bei dem ein vorausfahrendes Fahrzeug plötzlich bremst, wodurch es zu einem Auffahrunfall kommt [150].

Mit den ausgewählten logischen Szenarien soll sowohl die grundsätzliche Funktionalität des Testkonzepts, als auch die Übertragbarkeit auf unterschiedliche logische Szenarien evaluiert werden. Aus den Beschreibungen der verschiedenen Unfalltypen werden die zu überprüfenden Szenariomerkmale für die Validierung der Szenarien abgeleitet (siehe Tabelle 5.1). Manövertypen, die in Tabelle 5.1 nicht

Manövertyp	LS1	LS2	LS3	LS4
Infrastruktur	N./A. oder Annähern	Kreuzung überfahren	Kreuzung überfahren	N./A. oder Annähern
Abbiegen	N/A	Abbiegen	N/A	N/A
Spurmanöver	Spur folgen	Spur folgen	Spur folgen	Spur folgen
Objektmanöver	Fußgänger annähern	Fußgänger annähern	PKW annähern	PKW folgen
Kreuzen	Kreuzender Fußgänger	Kreuzender Fußgänger	Kreuzender PKW	N/A

Tabelle 5.1: Zu überprüfende Merkmale der getesteten logischen Szenarien

aufgezählt sind, werden während der Validierung des Szenarios nicht betrachtet.

Als zu variierende Parameter werden für die logischen Szenarien *LS1* und *LS2* die Geschwindigkeit des Fahrzeugs vor dem Konflikt v_{Fzg} , die Geschwindigkeit des Fußgängers v_{Ped} und der euklidische Abstand zwischen Fahrzeug und Fußgänger $s_{Fzg,Ped}$, bei dem der Fußgänger losläuft, betrachtet.

Die Minimal- und Maximalwerte für die Fahrzeug- und Fußgängergeschwindigkeit, sowie die Schrittweite ihrer Variation, werden auf Basis von Studien festgelegt (vgl. Abschnitt 3.3.1). Für den Parameter $s_{Fzg,Ped}$ existieren keine

realen Referenzdaten, die für die Definition des Wertebereichs herangezogen werden können. Da sehr geringe Abstände zu unvermeidbaren Kollision führen und das Notbremssystem bei zu großen Abständen nicht eingreifen muss, wird der Wertebereich anhand exemplarischer Simulationen empirisch ermittelt. Als Parameterverteilung wird für alle Parameter eine Gleichverteilung angenommen und daher eine feste Schrittweite verwendet (siehe Tabelle 5.2).

	Szenario 1			Szenario 2		
	Min	Max	Schrittweite	Min	Max	Schrittweite
v_{Fzg} [km/h]	16	40	2	16	30	2
v_{Ped} [m/s]	0.5	4	0.5	1	4	0.5
$s_{Fzg, Ped}$ [m]	6	26	2	6	26	2
Anzahl konkreter Szenarien / $N_{95\%}$	840/840			528/528		

Tabelle 5.2: Parameterbereiche und Schrittweiten für die logischen Szenarien *LS1* und *LS2*

Für das erste logische Szenario ergibt sich ein Testumfang von 840 konkreten Szenarien. Im zweiten logischen Szenario führen Fahrzeuggeschwindigkeiten von über 30 km/h dazu, dass das Fahrermodell das Fahrzeug beim Abbiegevorgang nicht mehr in der Fahrspur halten kann und es die Fahrspur verlässt. Aus diesem Grund wird der Parameterbereich der Fahrzeuggeschwindigkeit auf 30 km/h reduziert. Damit ergibt die Parametervariation 528 konkrete Szenarien für das zweite logische Szenario.

Für das logische Szenario *LS3* werden die Geschwindigkeiten des Fahrzeugs mit dem automatisierten Notbremssystem $v_{Fzg, AEB}$ und des wartepflichtigen Fahrzeugs $v_{Fzg, Konflikt}$ sowie ein Offset auf den Startpunkt des Fahrzeugs s_{Offset} mit dem Notbremssystem als Szenarioparameter verwendet (siehe Tabelle 5.3). Die Startpunkte beider Fahrzeuge sind so gewählt, dass beide Fahrzeuge die gleiche Strecke bis zu einem Konfliktpunkt auf der Kreuzung zurücklegen müssen.

	Szenario 3			Szenario 4		
	Min	Max	Schrittweite	Min	Max	Schrittweite
$v_{Fzg, AEB}$ [km/h]	25	50	5	25	50	2
$v_{Fzg, Konflikt}$ [km/h]	25	50	5	25	50	2
s_{Offset} [m]	0	5	0.5	N/A	N/A	N/A
s_{Fzg} [m]	N/A	N/A	N/A	20	40	0.5
$a_{Fzg, Konflikt}$ [m/s ²]	N/A	N/A	N/A	-2	-4.5	0.5
Anzahl konkreter Szenarien / $N_{95\%}$	395/395			390/390		

Tabelle 5.3: Parameterbereiche und Schrittweiten für die logischen Szenarien *LS3* und *LS4*

Mit dem Offset wird die zurückzulegende Strecke für das Fahrzeug mit dem Notbremssystem verlängert. Der Testumfang beträgt 396 konkrete Szenarien.

Im vierten logischen Szenario werden ebenfalls die Geschwindigkeiten der Fahrzeuge $v_{Fzg, AEB}$ und $v_{Fzg, Konflikt}$, sowie der Abstand der beiden Fahrzeuge vor der Bremsung des vorausfahrenden Fahrzeugs s_{Fzg} und die Verzögerung $a_{Fzg, Konflikt}$, mit der das vorausfahrende Fahrzeug bremst, variiert (siehe Tabelle 5.3). Die Geschwindigkeiten $v_{Fzg, AEB}$ und $v_{Fzg, Konflikt}$ werden allerdings nicht unabhängig voneinander variiert, sondern es gilt für jedes konkrete Szenario $v_{Fzg, AEB} = v_{Fzg, Konflikt}$. Durch die Parametervariation ergeben sich 390 konkrete Szenarien für das *LS4*.

5.2 Umsetzung an einem HiL-Prüfstand für Fahrerassistenzsysteme

5.2.1 Rahmenbedingungen und Projektkontext

Im Rahmen einer Forschungskooperation mit Mercedes-Benz, wurden während der Promotion Methoden zur Bewertung von Fahrerassistenzsystemen an Hardware-in-the-Loop-Prüfständen erforscht⁵. Dabei konnte die Entwicklung der Baureihe 223 (aktuelle S-Klasse)⁶ von der ersten Inbetriebnahme und Integration der Steuergeräte am HiL-Prüfstand bis zum Produktionsstart des Fahrzeugs begleitet werden (siehe Abbildung 5.6).

Der Fokus der Forschungskooperation lag auf der Bewertung von Fahrerassistenzsystemen der S-Klasse in zufällig erzeugten Fahrszenarien auf der Autobahn.



Abbildung 5.6: HiL-Prüfstand für Fahrerassistenzsysteme von Mercedes-Benz, an dem der vorgeschlagene Bewertungsansatz prototypisch erprobt werden konnte (Bild aus [213])

⁵ Die Zusammenarbeit erfolgte dabei mit einer Abteilung, die bei Mercedes-Benz für das entwicklungsbegleitende Testen von Fahrerassistenzsystemen am Wirkketten-HiL verantwortlich ist.

⁶ Markteinführung Dezember 2020

Das Modul zur Bewertung von FAS in Fahrszenarien wurde dafür an einem HiL-Prüfstand für Fahrerassistenzsysteme prototypisch implementiert und der gesamte Prozess, beginnend mit der Ableitung von Systemverhalten aus realen Systemanforderungen bis zur Auswertung der Testergebnisse, konnte in einem Serenumfeld erprobt werden (vgl. Modul 1 in Abbildung 4.22). Die Module „Validierung der simulierten Szenarien“ und „Bestimmung des Reifegrads“ waren kein Bestandteil der Umsetzung (vgl. Modul 2 und 3 in Abbildung 4.22).

Aus Gründen der Vertraulichkeit können keine detaillierten Informationen über durchgeführte Tests und erzeugte Testergebnisse vorgestellt werden. Die Erkenntnisse aus der Arbeit am HiL-Prüfstand sind dennoch maßgeblich in die prototypische Umsetzung der SiL-Testumgebung aus Abschnitt 5.1 und die im Kapitel 6.1 erfolgte Evaluation des Testkonzepts eingeflossen.

5.2.2 HiL-Prüfstand bei Mercedes Benz

Der HiL-Prüfstand wird von Mercedes-Benz genutzt, um die gesamte Wirkkette (Sensor, Funktion, Aktor) von Fahrerassistenzsystemen gemäß einer Testspezifikation zu testen [213]. Die Erstellung der verwendeten Testfälle erfolgt manuell durch einen Testingenieur und die Struktur der Testfälle entspricht der Struktur in Abbildung 2.10. Üblicherweise werden die Testfälle nachts automatisiert ausgeführt, sodass die Testergebnisse tagsüber von den Testingenieuren ausgewertet werden können.

Der Prüfstand besteht aus einem HiL-Simulator, der einen Echtzeitrechner beinhaltet, auf dem das Fahrzeugmodell und die Restbussimulation ausgeführt werden, einem Simulationsrechner für die Umgebungssimulation, einem Stereokamera-Prüfstand und verschiedenen Steuergeräten, wie beispielsweise dem zentralen FAS-Steuergerät (siehe Abbildung 5.7). Der Kamera-Prüfstand besteht aus der zu testenden Stereokamera und zwei Monitoren, auf denen die simulierte Umgebung visualisiert wird.

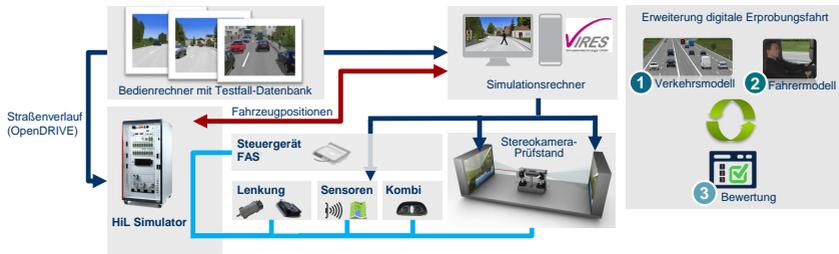


Abbildung 5.7: Architektur des Wirkketten-HiL für Fahrerassistenzsysteme mit der Erweiterung der digitalen Erprobungsfahrt (Abbildung angelehnt an [114])

Weitere real verbaute Komponenten sind unter anderem das Kombi-Instrument oder die Lenkung. Sensorseitig ist nur die Stereokamera als reale Komponente am Prüfstand verbaut. Die anderen verwendeten Sensoren im Fahrzeug, wie beispielsweise Ultraschall-, Radar- oder Lidar-Sensoren, sowie das GPS, werden in der Umgebungssimulation als Sensormodell nachgebildet. Die auszuführenden Testfälle sind in einer Testfall-Datenbank in der Teststeuerungssoftware auf einem Bedienrechner abgelegt⁷.

Darüber hinaus hat Mercedes-Benz den Prüfstand dahingehend erweitert, dass zufällig generierte Erprobungsfahrten erzeugt werden können. In diesen digitalen Erprobungsfahrten wird dem Fahrzeug eine Route auf einer Autobahn vorgegeben und der umgebende Verkehr mittels stochastischer Verkehrsflussimulation auf Basis von Verkehrsmodellen (Nummer 1 in Abbildung 5.7) und Fahrermodellen (Nummer 2 in Abbildung 5.7) erzeugt [115]. Mit den Verkehrsmodellen wird der gesamte Verkehr im betrachteten Streckennetz bestimmt, wie beispielsweise die Verkehrsdichte oder der Anteil an LKWs auf einem Streckenabschnitt. Fahrzeuge, die sich in der unmittelbaren Umgebung des virtuellen Testfahrzeugs befinden, werden durch Fahrermodelle gesteuert. Die Fahrermodelle übernehmen die Längs- und Querverführung des Fahrzeugs und sollen dabei menschliche Fahrer möglichst realistisch nachbilden.

⁷ verwendete Simulationsumgebung: *Virtual Test Drive*, verwendete Teststeuerungssoftware *PRO-VEtech:TA*

Ziel dabei ist es bereits im Labor Erprobungsfahrten durchführen zu können, um sowohl Aussagen über die Dauerlaufstabilität der E/E-Systeme treffen zu können als auch das korrekte Verhalten der Fahrerassistenzsysteme in Fahrscenarien auf der Autobahn zu bewerten [213]. Damit sollen zukünftig Kosten für Erprobungsfahrzeuge und Testfahrer während früher Entwicklungsphasen reduziert werden. Da Prototypen einer neuen Baureihe üblicherweise über eine halbe Million Euro kosten, ergibt sich daraus ein Einsparpotential von mehreren Millionen Euro pro Jahr an Fahrzeug-Prototypen und Personalaufwand.

Da vor Beginn einer digitalen Erprobungsfahrt nicht bekannt ist, welche Fahrscenarien zu welchem Zeitpunkt auftreten, kann die Bewertung der Fahrerassistenzsysteme in digitalen Erprobungsfahrten (Nummer 3 in Abbildung 5.7) mit den vorhandenen Bewertungsmethoden am Prüfstand nicht durchgeführt werden. Zusätzlich müssen die Fahrerassistenzsysteme während der gesamten Fahrt kontinuierlich überwacht werden, damit Aussagen über die Dauerlaufstabilität getroffen werden können. Daher wird die Bewertung der FAS durch den vorgestellten Bewertungsansatz aus Kapitel 4.2.2 umgesetzt. Dabei erfolgt die Bewertung bereits während der digitalen Erprobungsfahrt. Um dies zu realisieren, wird der Verifier auf dem Echtzeitrechner des HiL-Prüfstands ausgeführt und für jede Systembewertung wird ein eigener Thread auf dem Echtzeitrechner angelegt (siehe Abbildung 5.8). Nach der Testdurchführung wird ein Protokoll erzeugt, welche sowohl eine aggregierte Auswertung der Bewertungsergebnisse als auch Detailinformationen für jede Systembewertung enthält.

5.2.3 Getestete Fahrerassistenzsysteme

Im Rahmen der Zusammenarbeit wurden unter anderem folgende Fahrerassistenzsysteme der Baureihe 223 bewertet: ein adaptiver Abstandsregeltempomat,

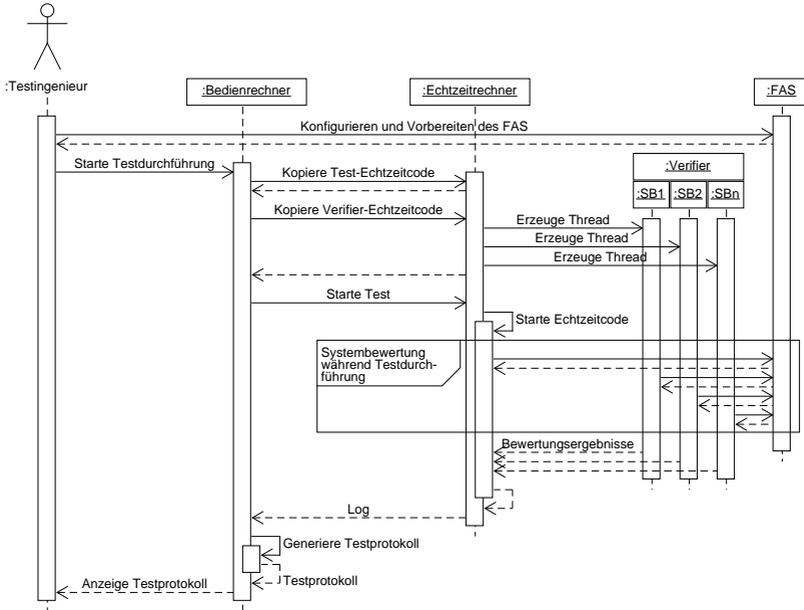


Abbildung 5.8: Workflow für die Testdurchführung am HiL-Prüfstand

ein aktiver Lenk- und Bremsassistent, ein Totwinkelassistent und die hoch automatisierte Fahrfunktion „DRIVE PILOT“⁸. Für die betrachteten Fahrerassistenzsysteme wurden Systembewertungen aus den realen Systemanforderungen abgeleitet und am Prüfstand implementiert (siehe Abbildung 5.9).

Die Anzahl an definierten Systembewertungen variiert dabei in Abhängigkeit von dem überprüften Funktionsumfang (siehe Tabelle 5.4). Während für den Abstandsregeltempomaten eine möglichst hohe Testtiefe erreicht werden sollte, lag der Fokus bei dem Totwinkel- und Spurhalteassistent darauf, die Übertragbarkeit des Ansatzes auf verschiedene FAS zu evaluieren.

⁸ Erstes zugelassenes hoch automatisiertes Fahrerassistenzsystem der SAE-Stufe 3

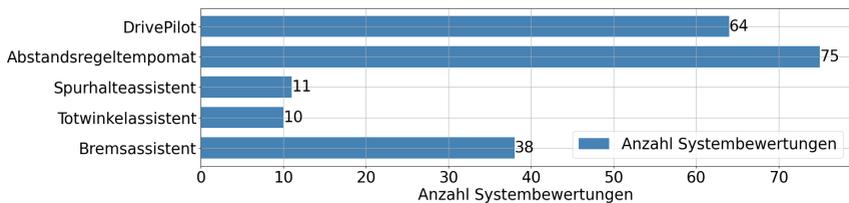


Abbildung 5.9: Anzahl der Systembewertungen für eine Auswahl der betrachteten Fahrerassistenzsysteme im Rahmen der prototypischen Umsetzung am HiL-Prüfstand

In der frühen Phasen des Entwicklungsprozesses, als die Funktionsentwicklung noch nicht so weit fortgeschritten war, dass funktionale Tests durchgeführt werden konnten, wurde der Bewertungsansatz zusätzlich dafür eingesetzt, um die Basisfunktionalität der verbauten Steuergeräte zu testen. Dafür wurden zusätzliche Systembewertungen definiert, welche die Grundfunktionalität der E/E-Systeme anhand von Steuergeräteresets oder der Kommunikation zwischen Steuergeräten bewertet.

Fahrerassistenzsystem	Überprüftes Systemverhalten
Adaptiver Abstandsregeltempomat	<ul style="list-style-type: none"> • Aktivierung- und Deaktivierung des Systems • Funktionalität der Geschwindigkeitsregelung • Funktionalität der Abstandsregelung • Änderung der Setzgeschwindigkeit • Übernahme von Geschwindigkeitslimits • Anzeige im Kombi-Instrument • Automatisches Anfahren aus dem Stillstand
Aktiver Lenkassistent	<ul style="list-style-type: none"> • Aktivierung- und Deaktivierung des Systems • Funktionalität der Querregelung • Hands-Off Erkennung
Aktiver Bremsassistent	<ul style="list-style-type: none"> • Vorbedingungen für einen Eingriff • Durchlauf der verschiedenen Eskalationsstufen • Warnung des Fahrers • Durchführung der Notbremsung
Totwinklassistent	<ul style="list-style-type: none"> • Anzeige von Fahrzeugen im toten Winkel • Eingriff bei drohender Seitkollision
DRIVE PILOT	<ul style="list-style-type: none"> • Aktivierung- und Deaktivierung des Systems • Erfüllung der Vorbedingungen

Tabelle 5.4: Übersicht der bewerteten Fahrerassistenzsysteme mit dem überprüften Systemverhalten

6 Evaluation des Testkonzepts

6.1 Bewertung relevanter Aspekte des Testkonzepts

6.1.1 Korrektheit der Bewertungsergebnisse

Um zu evaluieren ob der erarbeitete Bewertungsansatz korrekte Ergebnisse liefert, werden exemplarisch Bewertungsergebnisse des Notbremssystems in ausgewählten konkreten Szenarien (KS) mit aufgezeichneten Messdaten verglichen. Die konkreten Szenarien werden dafür so gewählt, dass alle Systemzustände des Notbremssystems abgedeckt werden (vgl. Abbildung 5.4).

KS1 Das Notbremssystem muss den Fahrer weder Warnen noch eine Notbremsung durchführen (Z1)

KS2 Das Notbremssystem warnt den Fahrer, aber muss keine Notbremsung durchführen (Z2)

KS3 Das Notbremssystem warnt den Fahrer, führt eine Notbremsung durch und die Kollision wird verhindert (Z3)

KS4 Das Notbremssystem warnt den Fahrer, führt eine Notbremsung durch und die Kollision wird nicht verhindert (Z3)

Für die Evaluation werden aus den Messungen die Zeitpunkte der aufgetreten Systemaktionen bestimmt, wie beispielsweise die Anforderung einer Warnung oder das Auftreten einer Kollision. Im nächsten Schritt werden diese Zeitpunkte

mit den Bewertungsergebnissen abgeglichen um zu überprüfen, ob die Systembewertungen die aufgetretenen Systemaktionen erkannt und korrekt bewertet haben (vgl. Abschnitt A.9). Da das Testkonzept reaktiv das Auftreten einer Systemaktion bewertet, wird zusätzlich noch überprüft, ob das Fahrerassistenzsystem auch zum richtigen Zeitpunkt bewertet wird. Dafür wird die zeitliche Differenz Δt zwischen dem Auftreten einer Systemaktion und der Bewertung durch eine Systembewertung bestimmt (siehe Tabelle 6.1).

Bei dem Vergleich der Bewertungsergebnisse mit den Messdaten werden in den betrachteten konkreten Szenarien KS1 bis KS4 die erwarteten Systemreaktionen, wie beispielsweise die Ausgabe einer Warnung oder Anforderung einer Notbremsung, sowie auftretende Kollisionen durch die implementierten Systembewertungen korrekt erkannt und bewertet. Allerdings ergibt die Auswertung auch, dass die Systembewertung „SB_E2Anf1“ keine korrekten Ergebnisse liefert. Mit der

Systembewertung	KS 1		KS 2		KS 3		KS 4	
	Ergebnis korrekt	Δt						
SB_E0Anf1	✓	0.0s	✓	0.0s	✓	0.0s	✓	0.0s
SB_E1Anf1	✓	0.0s	✓	0.0s	✓	0.0s	✓	0.0s
SB_E2Anf1	✗	0.0s	✗	0.0s	✗	0.0s	✗	0.0s
SB_E3Anf1	✓	0.0s	✓	0.0s	✓	0.0s	✓	0.0s
SB_E2Anf2	✓	0.0s	✓	0.0s	✓	0.0s	✓	0.0s
SB_E2Anf3	✓	0.0s	✓	0.0s	✓	0.0s	✓	0.0s
SB_E1Anf3	✓	0.0s	✓	0.0s	✓	0.0s	✓	0.0s
SB_E2Anf4	✓	0.0s	✓	0.0s	✓	0.0s	✓	0.0s

Tabelle 6.1: Ergebnisse der Überprüfung der definierten Systembewertungen in den konkreten Szenarien KS1-KS4

Systembewertung wird überprüft, ob zum Zeitpunkt der Erkennung des Fußgängers der Abstand zum Fußgänger größer ist als der notwendige Bremsweg bis zum Stillstand. Die Auswertung zeigt, dass die Systembewertung einen theoretischen Bremsweg berechnet, der nicht mit dem tatsächlichen Bremsweg in der Simulation übereinstimmt.

Der Grund für die falsche Berechnung des Bremswegs ist, dass der theoretisch notwendige Bremsweg in der Systembewertung vereinfacht anhand des Weg-Zeit-Gesetzes für die gleichmäßig beschleunigte Bewegung eines Massenpunkts berechnet wird. Da das Fahrzeug in der Simulationsumgebung nicht nur als Massenpunkt modelliert wird und die tatsächliche Verzögerung des Fahrzeugs erst nach einer gewissen Zeit die angeforderte Verzögerung erreicht, stimmt der berechnete Bremsweg der Systembewertung nicht mit dem tatsächlichen Bremsweg in der Simulation überein. Um zuverlässige Bewertungsergebnisse zu bekommen, muss die Berechnung in der Systembewertung an die Simulationsumgebung angepasst werden.

Weiterhin hat die Ermittlung der zeitlichen Differenz Δt in den untersuchten Szenarien ergeben, dass alle Systembewertungen in dem Zeitschritt aktiviert wurden, indem auch die Systemaktion stattgefunden hat. Dieses Ergebnis ist auch plausibel, da die Aktivierungs- und Prüfbedingungen in jedem Simulationsschritt ausgewertet werden und damit eine kontinuierliche Bewertung des Systems stattfindet.

6.1.2 Zeitliche Auflösung der Bewertungsergebnisse

Weicht beim Testen mittels klassischer Testfälle das Systemverhalten eines FAS von dem erwarteten Systemverhalten ab, dann wird der gesamte Testschritt, während dem das fehlerhafte Systemverhalten beobachtet wird, als fehlgeschlagen (in Englisch failed) oder „nicht in Ordnung“ (nio) markiert. Weiterführende Informationen, wie beispielsweise der genaue Zeitpunkt, ab dem die Fehlerwirkung aufgetreten ist oder wie lange sie vorlag, können nur durch eine nachfolgende,

manuelle Analyse der Messdaten erhalten werden (vgl. Definition 2.15). Ein weiteres relevantes Kriterium für die Evaluation des Bewertungskonzepts ist daher die zeitliche Auflösung, mit der das Fahrerassistenzsystem bewertet werden kann. Um diese zu evaluieren, werden neben den Ergebnissen aus dem vorangegangenen Abschnitt auch die Bewertungsergebnisse der Testdurchführung aus Kapitel 5.1.3 betrachtet (siehe Tabelle 6.2).

Für die Evaluation der Genauigkeit, mit der der Auftretenszeitpunkt einer Fehlerwirkung bestimmt werden kann, wird die zeitliche Differenz Δt betrachtet. Die Evaluation aus dem vorherigen Abschnitt hat bereits ergeben, dass in den betrachteten Szenarien das Auftreten einer Fehlerwirkung im selben Zeitschritt identifiziert werden konnte.

Die Bewertungsergebnisse in Tabelle 6.2 zeigen darüber hinaus, dass in mehreren logischen Szenarien Systembewertungen fehlschlagen, bei denen die durchschnittliche Dauer der beobachteten Fehlerwirkung nur wenige Millisekunden beträgt. Im zweiten logischen Szenario erfolgt beispielsweise die Anforderung einer Warnung (Anf. E2Anf2) in 41 konkreten Szenarien zwei Millisekunden zu spät und im ersten logischen Szenario schlägt die Systembewertung *SB_E1Anf3*, welche die Umsetzung der Notbremsung überprüft (Anf. E1Anf3), ebenfalls einmal fehl, weil die Verzögerung eine Millisekunde zu spät angefordert wird.

Systembewertung	Logisches Szenario 1		Logisches Szenario 2		Logisches Szenario 3		Logisches Szenario 4	
	Anzahl fehlgeschl. Bewertungen	Ø Dauer Fehlerwirkung						
SB_E0Anf1	67	3.12s	50	1.65s	12	0.02s	98	0.019s
SB_E1Anf1	1	0	79	0.92s	0	-	0	-
SB_E2Anf1	95	0.001s	27	0.001s	164	0.001s	0	-
SB_E3Anf1	0	-	0	-	0	-	0	-
SB_E2Anf2	1	0.001s	41	0.002s	0	-	1	0.02s
SB_E2Anf3	0	-	0	-	0	-	0	-
SB_E1Anf3	1	0.001s	0	-	21	0.019s	0	-
SB_E2Anf4	0	-	0	-	0	-	0	-

Tabelle 6.2: Ergebnisse der Systembewertungen mit Anzahl an fehlgeschlagenen Bewertungen und der durchschnittlichen Dauer der Fehlerwirkung

6.1.3 Praktische Relevanz der Szenariovalidierung

Um zu evaluieren, ob die Validierung der simulierten Szenarien ein notwendiger Bestandteil eines szenariobasierten Testkonzepts ist, wird untersucht, wie viele simulierte Szenarien aus Abschnitt 5.1.3 ohne die definierten Merkmale der logischen Szenarien erzeugt werden. Die Ergebnisse der Szenariovalidierung in Tabelle 6.3 zeigen, dass auch im Rahmen der Testdurchführung aus Abschnitt 5.1.3 konkrete Szenarien durch die Parametervariation erzeugt wurden, welche nicht die spezifizierten Merkmale aus Tabelle 5.1 beinhalten und damit als nicht valide gekennzeichnet werden (vgl. Tabelle 6.3).

Für das erste logische Szenario beinhalteten beispielsweise nur 584 der 840 (69.5 %) simulierten Szenarien einen kreuzenden Fußgänger vor dem Fahrzeug. In den verbleibenden Szenarien überquert der Fußgänger die Straße entweder nach dem Fahrzeug, läuft seitlich in das Fahrzeug hinein oder mit so viel Abstand vor dem Fahrzeug über die Straße, dass die PET größer ist als der gewählte Grenzwert (siehe Abbildung 6.1).

Die Ergebnisse in Tabelle 6.3 lassen allerdings auch eine Abhängigkeit von logischen Szenario vermuten. Für das Auffahrsszenario (*LS4*) existieren keine konkreten Szenarien, die nicht den spezifizierten Inhalt besitzen. Dies ist auch plausibel,

	Anzahl simulierter Szenarien	Anzahl valider/ nicht valider Szenarien	Anzahl valider Szenarien mit aktivem Notbremssystem
Szenario 1	840	584/256 (69.5 %/30.5 %)	444 (52.9 %)
Szenario 2	528	445/83 (84.3 %/15.7 %)	248 (46.9 %)
Szenario 3	395	302/93 (76.5 %/23.5 %)	162 (41.0 %)
Szenario 4	390	390/0 (100 % / 0.0 %)	389 (99.7 %)

Tabelle 6.3: Ergebnisse der Szenariovalidierung für die vier logischen Szenarien und die zu überprüfenden Merkmale aus der Tabelle 5.1

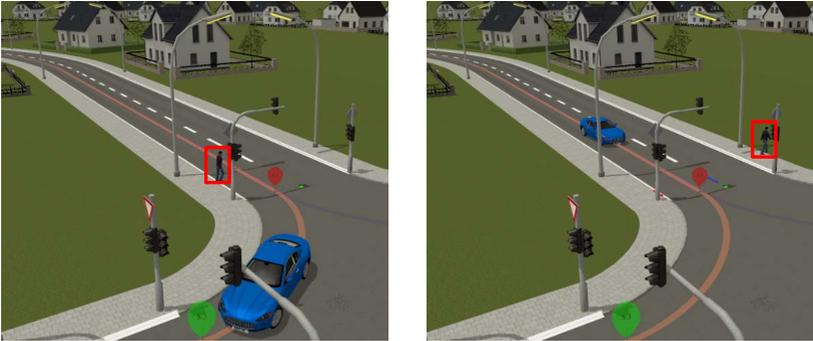


Abbildung 6.1: Visualisierung von zwei nicht validen konkreten Szenarien für das *LS1*

da die Geschwindigkeiten der beiden Fahrzeuge nicht unabhängig voneinander variiert werden (vgl. Abschnitt 5.1.3) und beide Fahrzeuge die selbe Route fahren.

6.1.4 Übertragbarkeit der Szenariovalidierung

Die Ergebnisse der Szenariovalidierung in Tabelle 6.3 zeigen für die logischen Szenarien *LS1*, *LS2* und *LS3* eine Diskrepanz zwischen der Anzahl an Szenarien mit einem kreuzendem Verkehrsteilnehmer und Szenarien mit aktivem Notbremssystem. In den logischen Szenarien *LS2* und *LS3* besitzen beispielsweise nur 52.4 %/53.6 % der konkreten Szenarien mit einem kreuzendem VT auch eine Systemreaktion des Notbremssystems. Dies zeigt, dass nicht nur die zu erkennenden Fahrmanöver, sondern auch die Schwellwerte der Fahrmanövermodellierung an das zu testende Fahrerassistenzsystem angepasst werden müssen (vgl. Abschnitt 4.3.1).

Wenn das Notbremssystem beispielsweise nur auf Verkehrsteilnehmer in einem Abstand bis zu 10m reagiert, dann kann ein Szenario, in dem ein Fußgänger 15m vor dem Fahrzeug die Straße kreuzt, für dieses Systems nicht als valide angesehen werden, da für das FAS kein Fußgänger in diesem Szenario existiert. Daraus folgt, dass die Modellierung der Manövererkennung und damit auch die gesamte Szenariovalidierung nur eingeschränkt auf andere FAS übertragbar ist.

Da die Validierung der Szenarien auf einer regelbasierten Manövererkennung basiert, können folglich nur die logischen Szenarien validiert werden, dessen Manöver und Interaktionen in der Modellierung abgebildet sind. Ist dies nicht der Fall, dann ist eine Anpassung der Modellierung und der Erkennung der Fahrmanöver notwendig. In der praktischen Anwendung hat sich gezeigt, dass vor allem nicht konsistente Modellierungen der Straße zu Problemen bei der Übertragbarkeit der Manövererkennung führen kann. Da OpenDrive viele Freiheiten bei der Modellierung einer Straße zulässt, ist eine robuste Erkennung der Infrastruktur bezogenen Manöver besonders aufwändig.

6.1.5 Evaluation der Generalisierungskonfidenz

Mit der Generalisierungskonfidenz wird bestimmt, ob der Parameterraum eines logischen Szenarios in einem ausreichendem Umfang abgedeckt wird, sodass eine generalisierte Aussage über das logische Szenario möglich ist (vgl. Definition 4.3). Für die Evaluation der Generalisierungskonfidenz wird zum einen überprüft, ob alle relevanten Bereiche des Parameterraums (vgl. Definition 2.37) mit konkreten Szenarien abgedeckt sind und zum anderen, ob die Bereiche mit einer ausreichenden Anzahl an konkreten Szenarien getestet werden.

Die Evaluation wird exemplarisch für das erste logische Szenario aus Kapitel 5.1.3 durchgeführt. In Abbildung 6.2 ist dafür der Parameterraum des ersten logischen Szenarios dargestellt und die konkreten Szenarien sind entsprechend des aufgetretenen Systemverhaltens des Notbremssystems eingefärbt. Die Grafik zeigt, dass die unterschiedlichen Betriebszustände des Notbremssystems über den gesamten Parameterraum verteilt aktiv sind. Daher ist die Vorgabe den gesamten Parameterraum abzudecken grundsätzlich notwendig, um eine generalisierte Aussage treffen zu können (vgl. Abbildung 5.4).

Allerdings hat die Szenariovalidierung auch ergeben, dass ein zusammenhängender Bereich des Parameterraums keinen kreuzenden Fußgänger beinhaltet. In diesem Bereich wäre daher keine gleichmäßige Abdeckung des Parameterraums

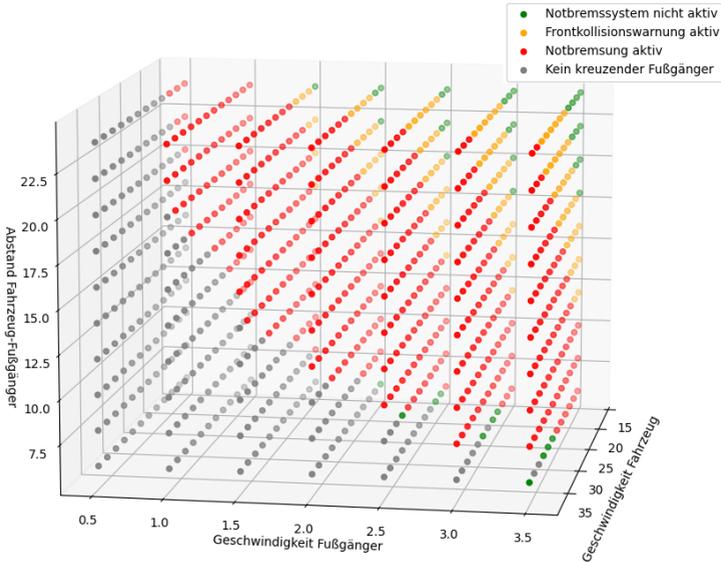


Abbildung 6.2: Parameterraum des ersten logischen Szenarien mit Systemaktivierungen

notwendig gewesen, da das Fahrerassistenzsystem in diesen Szenarien nicht aktiv wird (vgl. Abschnitt 6.1.3).

Ob die gewählte Anzahl an konkreten Szenarien ausreichend ist um eine repräsentative Aussage über das logische Szenario treffen zu können, kann mit den erzeugten Ergebnissen nicht bewertet werden. Um dies final zu beurteilen werden Methoden und Metriken benötigt, mit denen eine Repräsentativität der Parameterraumabdeckung beschrieben und ermittelt werden kann. Im aktuellen Stand der Wissenschaft gibt es noch keinen Konsens darüber, was eine repräsentative Abdeckung eines logischen Szenarios überhaupt bedeutet. Um diese zu bestimmen wird beispielsweise ein geeignetes Ähnlichkeitsmaß benötigt, mit dem eine Parameterraumabdeckung grundsätzlich beurteilt werden kann. Darüber hinaus werden Grenzwerte benötigt, ab welchem Wert eines solchen Ähnlichkeitsmaßes weitere konkrete Szenarien benötigt werden um eine repräsentative Abdeckung zu erreichen. Dies ist allerdings eine Diskussion, die nicht nur auf technischer

Ebene geführt werden kann, sondern bei der auch juristische und ethische Standpunkte betrachtet werden müssen. Diese Fragen werden daher auch im Rahmen von Forschungsprojekten wie VV-Methoden behandelt [43].

6.1.6 Aussagekraft der aggregierten Testergebnisse und des ermittelten Reifegrads

Das erarbeitete Testkonzept bestimmt den Reifegrad eines Fahrerassistenzsystems durch mehrere Aggregationsschritte. Dabei entstehen sowohl für jedes logische Szenario als auch für die gesamte ODD Informationen über den Grad der Anforderungserfüllung und den Reifegrad. Tabelle 6.4 zeigt die Aggregation der Testergebnisse für die vier durchgeführten logischen Szenarien (vgl. Tabellen A.11-A.14). Der Grad der Anforderungserfüllung ist dabei sowohl für die allgemeine Anforderung nach Kollisionsfreiheit als auch heruntergebrochen auf die drei Subsysteme „Umfeld erkennen“, „Aktion planen“ und „Aktion umsetzen“ angegeben (vgl. Abschnitt 5.1.2).

Logisches Szenario	Reifegrad	Generalisierungskonfidenz	Grad der Anforderungserfüllung		
			Kollisionsfrei	Umfeld erkennen	Aktion planen Aktion umsetzen
LS1	92.0 %	95.0 %	88.5 %	94.4 %	99.9 % 99.9 %
LS2	90.3 %	95.0 %	88.5 %	90.5 %	94.5 % 100 %
LS3	88.8 %	95.0 %	91.7 %	69.9 %	100 % 92.8 %
LS4	92.3 %	95.0 %	73.6 %	100 %	99.9 % 100 %
ODD	90.9 %	95.0 %	85.6 %	88.7 %	98.6 % 98.2 %

Tabelle 6.4: Ergebnisse der Reifegradbestimmung für die vier betrachteten logischen Szenarien und die ODD aus den Testergebnissen der Tabellen A.11-A.14

Aus den Testergebnissen in Tabelle 6.4 können dann potentielle Fehlerzustände im Fahrerassistenzsystem identifiziert werden (vgl. Definition 2.14). Für das betrachtete Notbremssystem besitzt beispielsweise das Subsystem „Umfeld erkennen“ in allen vier logischen Szenarien den geringsten Grad der Anforderungserfüllung und bedarf weiterer Entwicklungsaktivitäten. Darüber hinaus zeigen die Ergebnisse, dass die drei Subsysteme im vierten logischen Szenario grundsätzlich korrekt funktionieren, aber dennoch Kollisionen vorkommen. Mögliche Gründe dafür sind, dass die Kollisionen aufgrund der Parametrisierung durch das Notbremssystem nicht vermieden werden konnten oder dass die definierten Systemanforderungen unzureichend sind. In beiden Fällen wären dies Informationen, die im aktuellen Entwicklungsprozess vermutlich nicht schon während der Verifikation auffallen würden, sondern erst bei der Validierung des Systems. Grundlage dafür ist allerdings, dass die Dekomposition der Anforderungen während des Systementwurfs vollständig durchgeführt wurde (vgl. Definition 2.9).

Der ermittelte Reifegrad für das Notbremssystem variiert zwischen den verschiedenen logischen Szenarien und der gesamten ODD. Für eine fundierte Beurteilung des Reifegrads fehlt allerdings eine Referenz, mit der die ermittelten Reifegrade verglichen werden können. Sie können dennoch als Indikator dienen, da sie nicht nur eine Metrik betrachten, wie beispielsweise die Kollisionsfreiheit, sondern alle überprüften Anforderungen mit einbeziehen.

6.1.7 Vergleich des Testkonzepts mit der Bewertung mittels Kritikalitätsmetriken

Der aktuelle Stand der Wissenschaft und Technik verwendet ausschließlich Kritikalitätsmetriken, um FAS in Fahrscenarien zu bewerten (vgl. Abschnitt 2.3.3.4). Aus diesem Grund wird das in dieser Arbeit entwickelte Testkonzept der Bewertung mittels Kritikalitätsmetriken gegenübergestellt. Dafür werden exemplarisch die Ergebnisse der Testdurchführung des Notbremssystems mit den Ergebnissen der Kritikalitätsmetrik „Kollisionsfreiheit“ verglichen. In Abbildung 6.3 wird für die vier logischen Szenarien der prototypischen Umsetzung aufgezeigt, wie viele

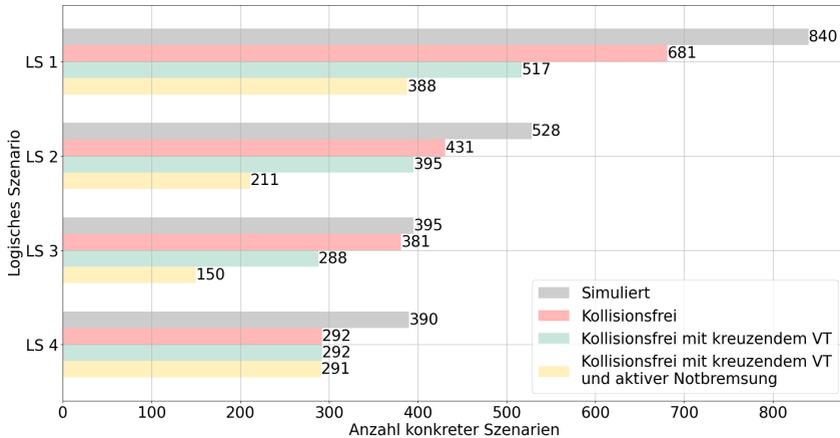


Abbildung 6.3: Gegenüberstellung der simulierten Szenarien (grau) mit den Ergebnissen der Bewertung durch die Kritikalitätsmetrik „Kollisionsfreiheit“ (rot) und dem vorgestellten Testkonzept (vgl. Abbildung 4.22) (grün, gelb)

der simulierten Szenarien aus Abschnitt 5.1.3 kollisionsfrei sind (roter Balken), wie viele der kollisionsfreien Szenarien einen kreuzenden Verkehrsteilnehmer beinhaltet haben (grüner Balken) und in wie vielen Szenarien darüber hinaus das Notbremssystem eingegriffen hat, um die Kollision zu verhindern (gelber Balken).

Betrachtet man in Abbildung 6.3 wie viele kollisionsfreie Szenarien (roter Balken) tatsächlich auch einen kreuzenden Verkehrsteilnehmer beinhaltet (grüner Balken), dann besitzen bis zu 24 % der kollisionsfreien Szenarien für das erste logische Szenario keinen kreuzenden VT vor dem Fahrzeug (siehe Tabelle 6.5). Vergleicht man darüber hinaus die Anzahl an kollisionsfreien Szenarien mit der Anzahl an kollisionsfreien Szenarien mit aktiver Notbremsung (gelber Balken), dann zeigt die Auswertung in Tabelle 6.5, dass bis zu 60.6 % der kollisionsfreien Szenarien keinen Eingriff des automatisierten Notbremssystems beinhalten. Wird die Bewertung des Notbremssystems also nur auf Basis einer Kritikalitätsmetrik durchgeführt, dann kann dies bedeuten, dass der Reifegrad des Systems auch auf Basis von konkreten Szenarien bestimmt wird, in denen das System überhaupt nicht aktiv war.

	LS1	LS2	LS3	LS4
kollisionsfrei/simulierte Szenarien	81.1 %	81.6 %	96.4 %	74.9 %
kollisionsfrei ohne VT/kollisionsfrei	24.1 %	8.4 %	24.4 %	0.0 %
kollisionsfrei ohne AEB/kollisionsfrei	43.1 %	51.1 %	60.6 %	0.4 %

Tabelle 6.5: Vergleich der Bewertung durch Kritikalitätsmetriken mit den Ergebnissen aus dem vorgestellten Testkonzept

Im Gegensatz dazu wird mit dem erarbeiteten Testkonzept sowohl überprüft, dass das Notbremssystem eingegriffen hat (vgl. Systembewertung SB_E2Anf3 auf Seite 182) als auch dass es eine Kollision verhindert hat (vgl. Systembewertung SB_E0Anf1 auf Seite 182).

Darüber hinaus wurde in Abschnitt 3.2 bereits diskutiert, dass die gängigen Kritikalitätsmetriken, wie die TTC, PET, der minimale Abstand oder die Kollisionsfreiheit grundsätzlich nur überprüfen, ob ein räumlicher oder zeitlicher Abstand eingehalten wird. Dies deckt sich auch mit der Klassifikation von Kritikalitätsmetriken in der Literaturübersicht von [134], in der über 70 % der betrachteten Kritikalitätsmetriken zu den zeitabstands-basierten und raumabstands-basierten Indikatoren zugeordnet werden. Die verbleibenden Kritikalitätsmetriken werden den verzögerungs-basierten Indikatoren, welche auf der „Verzögerungsrate zur Verhinderung einer Kollision“ aufbauen und sonstigen Indikatoren, wie beispielsweise dem Ruck oder der Standardabweichung der lateralen Position, zugeordnet.

Aus diesem Grund wird exemplarisch gegenübergestellt, wie umfangreich verschiedene FAS von Mercedes-Benz mit dem erarbeiteten Testansatz und gängigen Kritikalitätsmetriken bewertet werden können. Als Grundlage für die Gegenüberstellung wird das bewertete Systemverhalten am HiL-Prüfstand betrachtet (vgl. Tabelle 5.4). Dem wird gegenübergestellt, welches Systemverhalten theoretisch auch mit gängigen Kritikalitätsmetriken wie der TTC, PET, dem minimalem Abstand oder der Kollisionsfreiheit bewertet werden kann (siehe Tabelle 6.6). Tabelle 6.6 zeigt aus Gründen der Übersichtlichkeit nur eine Auswahl des überprüften Systemverhaltens am HiL-Prüfstand.

Funktionalität des Fahrerassistenzsystem	Kritikalitätsmetrik				Erarbeiteter Testansatz am HiL
	Minimaler Abstand	TTC	PET	Kollision	
Adaptiver Abstandsregeltempomat					
Geschwindigkeitsregelung	✗	✗	✗	✗	✓
Abstandsregelung	✗	✗	✓	✗	✓
Anzeige im Kombi-Instrument	✗	✗	✗	✗	✓
Übernahme von Geschwindigkeitslimits	✗	✗	✗	✗	✓
Aktiver Bremsassistent					
Warnung des Fahrers	✗	✗	✗	✗	✓
Notbremsung	✓	✓	✗	✓	✓
Aktiver Lenkassistent					
Halten der Fahrspur	✗	✗	✗	✗	✓
Spurwechselunterstützung	✗	✗	✗	✗	✓
Verkehrszeichenassistent					
Erkennung Geschwindigkeitslimits	✗	✗	✗	✗	✓
Totwinkelassistent					
Anzeigen von Fahrzeugen im toten Winkel	✗	✗	✗	✗	✓
Verhindern von Seitkollisionen	✓	✓	✗	✓	✓

Tabelle 6.6: Vergleich des testbaren Funktionsverhaltens mittels gängiger Kritikalitätsmetriken und dem erarbeiteten Testkonzept für ausgewähltes Funktionsverhaltens aus Tabelle 5.4

Die Ergebnisse der Bewertung in Tabelle 6.6 zeigen, dass mit den betrachteten Kritikalitätsmetriken fast ausschließlich Funktionalitäten bewertet werden können, die Kollisionen mit anderen Verkehrsteilnehmern verhindern. Sowohl der aktive Bremsassistent, das Notbremssystem von Mercedes-Benz, als auch der Totwinkelassistent können mit den betrachteten Metriken allerdings nicht im selben Umfang bewertet werden, wie es mit dem vorgestellten Bewertungsansatz möglich ist. Interaktionen mit dem Fahrer, wie die Ausgabe einer Warnung oder

die Anzeige eines Fahrzeugs im toten Winkel, werden von den Metriken nicht betrachtet.

Die PET, welche den Zeitabstand beim Passieren eines fixen Punkts durch zwei Fahrzeuge repräsentiert (vgl. Abbildung 2.21), könnte darüber hinaus theoretisch auch dafür verwendet werden, um die Abstandsregelung des adaptiven Abstandsregeltempomaten zu bewerten. Zusammenfassend lässt sich festhalten, dass mit dem erarbeiteten Testkonzept eine höhere Testtiefe für die betrachteten Fahrerassistenzsysteme erreicht werden kann als mit den Kritikalitätsmetriken (vgl. Definition 4.2).

6.2 Theoretische Betrachtung wirtschaftlicher Potentiale

6.2.1 Reduktion des Aufwands bei der Testerstellung

Durch die Einführung des szenariobasierten Testens wird die Anzahl der durchzuführenden Tests während der Entwicklung zukünftig steigen. Aktuell werden Testfälle klassischer Weise von Testingenieuren manuell erstellt und implementiert. Da im Kontext des szenariobasierten Testens noch keine Testorakel existieren, mit denen das erwartete Systemverhalten für verschiedene Parametrisierungen eines logischen Szenarios prädiziert werden kann (vgl. Kapitel 3.1.2), ist keine automatisierte Testfallerstellung möglich. Daher müssen beim klassischen Ansatz die Zeitpunkte, zu denen ein bestimmtes Systemverhalten erwartet wird, für jede Parameterkombination manuell ermittelt werden. Damit ist der Aufwand für die Erstellung von Tests ein wichtiger ökonomischer Faktor und soll an einem theoretischen Beispiel evaluiert werden.

Dafür wird exemplarisch die Erstellung von Testfällen sowohl für die vier betrachteten logischen Szenarien als auch einen vermutlich realistischeren Umfang

Rahmenbedingungen:

- Anzahl logische Szenarien: 100
- Anzahl konkrete Szenarien/Testfälle: 1000
- Anzahl zu überprüfende Anforderungen: 9

Erstellung klassischer Testfälle	Vorgeschlagener Ansatz
<p>Zeitaufwand für die Erstellung von:</p> <ul style="list-style-type: none"> • Testfall: 5 min/Testfall • Testfallkatalog für 1 logisches Szenario: 5 min * 1000 = 83 h 20 min • Testfallkatalog für 100 logisches Szenario: 83 h 20 min * 100 = 8333 h 20 min <p>Zeitaufwand: 8333 h 20 min</p>	<p>Zeitaufwand für die Erstellung von:</p> <ul style="list-style-type: none"> • Systembewertung: 30 min • Systembewertungen für 9 Anforderungen: 30 min * 9 = 4.5 h • Logisches Szenario mit 1000 konkreten Szenarien: 5 min • 10 logische Szenarien 5 min * 100 = 8 h 20 min <p>Zeitaufwand : 12 h 50 min</p>

Abbildung 6.4: Beispielrechnung für die Zeit- und Geldersparnis mit dem erarbeiteten Ansatz im Vergleich zu etablierten Testfällen

von 100 logischen Szenarien mit jeweils 1000 konkreten Szenarien betrachtet¹. In etablierten Testprozessen würde die Erstellung der Testfälle sowohl eine Vorsimulation der konkreten Szenarien als auch die Implementierung der Szenarien als Testfall umfassen (vgl. Abbildung 2.9). Während der Vorsimulation wird analysiert, welches Systemverhalten zu welchem Zeitpunkt auftritt und dies würde dann in Testfällen bestehend aus Testschritten mit Aktionen und Bestehensbedingungen implementiert. Erfahrungsgemäß kann für die manuelle Erstellung der Testfälle am HiL-Prüfstand eine durchschnittliche Dauer von 5 Minuten pro Testfall angenommen werden. Für die Erstellung einer Systembewertung werden 30 Minuten angesetzt. Die Implementierung und automatisierte Parametervariation eines logischen Szenarios dauert circa 5 Minuten.

Um aus den vier logischen Szenarien aus Abschnitt 5.1.3 manuell Testfälle abzuleiten, werden damit ungefähr 179 Stunden benötigt. Mit dem erarbeiteten Ansatz werden stattdessen nur circa fünf Stunden benötigt.

¹ Bei einer Szenarioextraktion für eine urbane Kreuzung wurden über 500 verschiedene logische Szenarien identifiziert [King et al., 2021]

Aufgrund der Tatsache, dass für jede Parameterkombination ein neuer Testfall erstellt werden muss, steigt die insgesamt benötigte Zeit linear mit der Anzahl an logischen Szenarien und konkreten Szenarien pro logisches Szenario an. Im Gegensatz dazu muss für den erarbeiteten Ansatz nur einmalig Zeit aufgewendet werden, um die Systembewertungen zu erstellen und die logischen Szenarien zu implementieren. Die konkreten Szenarien können dann automatisiert mittels Parametervariation in dem Testautomatisierungstool erstellt werden. Dadurch steigt der Zeitaufwand für die manuelle Erstellung des zweiten Falls mit 100 logischen Szenarien stark an und ändert sich für den erarbeiteten Ansatz nur geringfügig (siehe Abbildung 6.4). Bei gleichem Zeitaufwand kann daher mit dem erarbeiteten Ansatz eine höhere Testbreite erzielt werden als mit der klassischen Erstellung von Festfällen (vgl. Definition 4.1).

Gibt es darüber hinaus Modifikationen an der Parametrisierung des Fahrerassistenzsystems, wie beispielsweise eine Änderung der angeforderten Verzögerung des Notbremssystems, dann müssen im klassischen Ansatz alle Testfälle angepasst werden, bei denen die Verzögerung in einer Bestehensbedingung überprüft wird. Im schlechtesten Fall wären das in dem theoretischen Beispiel alle 100.000 erzeugten Testfälle. Im vorgestellten Bewertungsansatz müssen lediglich die Systembewertungen angepasst werden, welche die angeforderte Verzögerung als Aktivierungs- oder Prüfbedingung verwenden. Damit wären dort im schlechtesten Fall Änderungen an 9 Systembewertungen notwendig.

6.2.2 Effizienzsteigerungen bei der Testdurchführung

Betrachtet man darüber hinaus in Tabelle 6.3 in wie vielen validen, konkreten Szenarien das Notbremssystem tatsächlich auch aktiv wird, indem es entweder eine Warnung ausgibt oder eine Bremsung durchführt hat, dann liegt der Anteil an simulierten Szenarien mit Systemaktivierung teilweise unter 50 %. Im dritten logischen Szenario wird das Notbremssystem beispielsweise in 59 % der simulierten Szenarien nicht aktiv und wird daher, aufgrund der verwendeten Systembewertungen, in diesen Szenarien auch nicht getestet.

Dazu kommt, dass der gesamte Testkatalog eines FAS während der Entwicklung erfahrungsgemäß mehr als 10 mal pro Jahr getestet wird. Dabei ist jedes simulierte Szenario sowohl mit einmaligen Kosten für die Erstellung als auch mit fortlaufenden Kosten für die Simulation und Auswertung verbunden. Dies ist besonders für Tests relevant, die an HiL-Prüfständen oder mit Versuchsträgern auf dem Prüfgelände durchgeführt werden. HiL-Prüfstände für Fahrerassistenzsysteme können beispielsweise im Aufbau und Betrieb über 500.000€ pro Prüfstand kosten. Aus den Kosten für den Prüfstand und einen betreuenden Testingenieur, ergibt sich ein Preis von ungefähr 0.44€ pro Minute Simulationszeit in einem Dauerbetrieb des HiLs². Testfälle am HiL dauern, bedingt durch notwendige Aktionen, wie das Initialisieren von Steuergeräten oder das Zurücksetzen von Fehlerspeichern, erfahrungsgemäß länger als zwei Minuten. Daraus ergibt sich eine untere Grenze von circa 1.00€ pro simuliertem konkreten Szenario. Wird beispielsweise von einem vier wöchentlichen Testzyklus ausgegangen, dann ergibt sich für die vier logischen Szenarien aus Abschnitt 5.1.3 ein Einsparpotential von ungefähr 11.000€ pro Jahr, wenn die konkreten Szenarien, in denen das Notbremsystem nicht getestet wird, ausgelassen werden. Für die angenommenen 100 logischen Szenarien aus dem vorherigen Abschnitt, ergibt sich bei durchschnittlich 39 % simulierter Szenarien ohne Aktivierung des Systems (vgl. Tabelle 6.3) ein Betrag von circa 500.000€ pro Jahr, der theoretisch für die Durchführung von Tests aufgewendet werden muss, in denen das Notbremsystem nicht getestet wird und durch die Verwendung der Szenariovalidierung gespart werden kann.

6.3 Diskussion und Fazit

Der Funktionsumfang von Fahrerassistenzsystemen nimmt stetig zu und aktuelle FAS werden daher zunehmend sowohl von der statischen als auch von der dynamischen Fahrzeugumgebung beeinflusst. Dabei übernehmen die FAS Aufgaben wie die Anpassung der Geschwindigkeit an Tempolimits, die Überwachung des toten

² Mit angenommenen Unternehmenskosten von circa 150.000€ pro Jahr für einen nach EG15 eingruppierten Testingenieur.

Winkels oder die Durchführung eines Spurwechsels (vgl. Tabelle 2.1). Aus diesem Grund muss zukünftig neben der funktionalen Sicherheit auch die Betriebssicherheit der FAS betrachtet werden. Die dafür zuständige Norm ISO 21448 steht vor der Finalisierung und da die Norm szenariobasiert arbeitet, wird damit das Thema des szenariobasierten Testens nicht nur für automatisierte Fahrzeuge relevant, sondern auch schon für Systeme bis zum Automatisierungsgrad 2 (vgl. Abschnitt 2.3.1). Damit werden zukünftig potentiell auch Fahrerassistenzsysteme wie beispielsweise ein Abstandsregeltempomat, Lenkassistent oder Totwinkelassistent szenariobasiert getestet werden.

Aus diesem Grund werden zunächst Faktoren diskutiert, die einen Einfluss auf die Übertragbarkeit der Evaluationsergebnisse besitzen. Als erster Einflussfaktor soll diskutiert werden, dass die Evaluation des Testkonzepts nur der Grundlage eines Fahrerassistenzsystems durchgeführt wurde. Wie im Abschnitt 5.2.3 beschrieben, wurde das Bewertungskonzept im Rahmen der Forschungskooperation mit Mercedes Benz für verschiedene Fahrerassistenzsysteme umgesetzt. Auch wenn die dabei erzeugten Ergebnisse aus Gründen der Vertraulichkeit nicht veröffentlicht werden dürfen, stützen sie die Evaluationen aus Abschnitt 6.1.1, 6.1.2 und 6.1.7 und untermauern ihre Übertragbarkeit. Darüber hinaus kann die Auswahl der logischen Szenarien einen Einfluss auf die Ergebnisse der Evaluation und deren Übertragbarkeit haben (vgl. Abschnitt 6.1.3 und 6.1.4). An dieser Stelle wird argumentiert, dass, durch die Verwendung relevanter Unfallszenarien und repräsentativer Parameterverteilungen für das betrachtete Notbremssystem, zumindest für den betrachteten Bereich davon ausgegangen werden kann, dass die Ergebnisse repräsentativ sind. Die Evaluation des Produktreifegrads muss im Bezug auf die betrachteten Anforderungen kritisch diskutiert werden. Da keine vollständige Liste an Systemanforderungen verfügbar ist, kann nicht ausgeschlossen werden, dass Systemanforderungen existieren, die mit dem vorgestellten Testkonzept nicht überprüft werden können. Auch im Rahmen der Umsetzung am HiL-Prüfstand, konnten für kein Fahrerassistenzsystem alle Systemanforderungen überprüft werden. Damit kann nicht davon ausgegangen werden, dass die Ergebnisse verallgemeinerbar sind.

Obwohl die Ergebnisse der vorgestellten Evaluation nicht allgemeingültig sind, konnte gezeigt werden, dass mit dem erarbeiteten Testkonzept die Ermittlung von szenariospezifischen Reifegraden grundsätzlich möglich ist. Damit kann erstmals eine Abschätzung des aktuellen Reifegrads heruntergebrochen auf die ODD eines Fahrerassistenzsystems erfolgen. Dies ermöglicht die Integration des szenariobasierten Testens in den automobilen Produktentwicklungsprozess, indem der ermittelte Reifegrad innerhalb der ODD als Messkriterium für den RG-Indikator „Produkt-Absicherung“ verwendet werden kann (vgl. Abschnitt 2.2).

Welche Aussagekraft der ermittelte Reifegrad allerdings hat, muss noch weiter evaluiert und diskutiert werden. Im vorgestellten Konzept hängt der Reifegrad unter anderem von Testbreite und damit von den getesteten logischen Szenarien ab. Da sich Fahrerassistenzsysteme in einem offenen Kontext bewegen, sind während der Entwicklung potentiell noch nicht alle logischen Szenarien bekannt, die für das Fahrerassistenzsystem relevant sind (vgl. Abschnitt 2.1.1). Werden neue logische Szenarien im Verkehr entdeckt, dann führt dies zunächst zu einer Verringerung des Reifegrads, obwohl sich das Fahrerassistenzsystem selbst nicht verändert hat. Durch die Einführung neuer Verkehrszeichen oder Verkehrsteilnehmer, wie E-Scooter, Lastenräder oder schlussendlich auch hochautomatisierter Fahrsysteme, kann zusätzlich davon ausgegangen werden, dass sich die ODD eines FAS während seines Betriebs zukünftig ändern wird (vgl. Abbildung 2.2). Daraus folgt dann auch, dass es vermutlich keinen finalen und vollständigen Reifegrad eines FAS mehr geben wird, da er immer von dem aktuellen und vermutlich unvollständigem Wissen über die ODD abhängt. Wie mit diesem unvollständigem Wissen über die ODD umgegangen werden kann, wird im Rahmen des Forschungsprojekts VV-Methoden erforscht.

Die Evaluation in Abschnitt 6.1.7 hat ergeben, dass die Auswertung von Kritikalitätsmetriken allein nicht ausreichend ist um Funktionsverhalten von Fahrerassistenzsystemen in Fahrscenarien zu testen. Aus diesem Grund werden alternative Methoden für das szenariobasierte Testen benötigt, mit denen ermittelt werden kann, ob ein System die gestellten Systemanforderungen erfüllt. Mit dem erarbeiteten Bewertungskonzept konnte dies sowohl in gezielt erzeugten Szenarien

im SiL als auch zufällig erzeugten Szenarien während der digitalen Erprobungsfahrt erreicht werden. Im Rahmen der Umsetzung am HiL-Prüfstand und der SiL-Testumgebung waren die Systembewertungen auf die unterschiedlichen logischen Szenarien übertragbar. Damit hat sich die Annahme aus der Diskussion des Stands der Wissenschaft, dass das deklarative Testen in den Rahmenbedingungen des szenariobasierten Testens eingesetzt werden könnte, für die betrachteten Anwendungsfälle bestätigt. Eine universelle Übertragbarkeit des deklarativen Testansatzes kann aus den erzeugten Erkenntnissen allerdings nicht hergeleitet werden.

Soll der erarbeitete Testansatz für eine ganzheitliche Bewertung eines Fahrerassistenzsystems verwendet werden, dann wäre allerdings gewissermaßen eine Nachmodellierung des zu testenden Systems erforderlich. Theoretisch wird eine Art „abstrakter, virtueller Zwilling“ des Systems benötigt, der in Form von Systembewertungen das Sollverhalten abbildet und mit dem das Verhalten des realen Systems verglichen wird. In Abhängigkeit des Detaillierungsgrads, mit dem ein „virtueller Zwilling“ beschrieben und implementiert werden soll, nimmt der dafür notwendige Aufwand zu. Darüber hinaus müsste ein solcher „virtueller Zwilling“ allerdings selbst auch validiert werden, damit er für das Testen verwendet werden kann.

Um das erarbeitete Bewertungskonzept einsetzen zu können sind Systemanforderungen notwendig, die das geforderte Systemverhalten nicht nur qualitativ beschreiben, sondern auch systemspezifische Werte enthalten, wie beispielsweise die angeforderte Verzögerung bei einer Notbremsung oder bei welcher TTC eine Warnung angezeigt wird. Üblicherweise werden solche Werte getrennt von den Anforderungen als Funktionsparameter beschrieben. Um diese Informationen für reale Fahrerassistenzsysteme zu erhalten, sind demnach vermutlich Abstimmungen mit den jeweiligen Fachbereichen wertvoll.

Darüber hinaus haben insbesondere die Ergebnisse der Szenariovalidierung gezeigt, dass die bisher angenommene Generalisierungsannahme des szenariobasierten Testens nicht für jedes logische Szenario gleichermaßen gilt. Durch die Variation von Parametern entstehen zum einen konkrete Szenarien, die nicht den

geplanten Inhalt besitzen und zum anderen auch Szenarien, in denen das zu testende Systemverhalten nicht auftritt (vgl. Tabelle 6.3). Aus diesem Grund stellt die Einführung einer Generalisierungskonfidenz einen notwendigen Schritt dar, um zuverlässige und belastbare Ergebnisse mit einem szenariobasierten Testansatz erzeugen zu können. Die Generalisierungskonfidenz bietet dabei auch die Möglichkeit für einen Eingriffspunkt regulatorischer Vorgaben. So können beispielsweise in der StVZO (auf nationaler Ebene) oder in UNECE-Regularien (auf internationaler Ebene) Regeln definiert werden, wie die Gültigkeit der Generalisierungsannahme nachgewiesen werden muss (vgl. Abschnitt 2.4.1).

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung

Durch den Betrieb von Fahrerassistenzsystemen in einem offenen Kontext ist es unvermeidlich, dass kritische Fahrscenarien entstehen, die vorher unbekannt waren und ein Risiko für die Verkehrssicherheit darstellen. Etablierte Testkonzepte, wie beispielsweise das anforderungsbasierte Testen, dienen dazu die Erfüllung von Systemanforderungen exemplarisch nachzuweisen. Da die neuen Herausforderungen größtenteils durch die Interaktion von Fahrzeugen mit der statischen und dynamischen Fahrzeugumgebung entstehen, sind die etablierten Konzepte nicht mehr ausreichend um die dabei entstehenden Herausforderungen zu bewältigen.

Aktuelle Forschungen fokussieren sich daher auf das szenariobasierte Testen für die Zulassung von hochautomatisierten Fahrsystemen. Zur Bewertung der Szenarien werden dabei Kritikalitätsmetriken wie die TTC oder das Vorhandensein einer Kollision verwendet. Eine Verifikation der Systeme in Fahrscenarien, bei der überprüft wird, ob das System die spezifizierten Anforderungen erfüllt, ist bisher nicht im Fokus der Forschungsarbeiten und daher mit den bisher verfügbaren Methoden des szenariobasierten Testens nicht möglich.

Im Gegensatz zu bisherigen Forschungsschwerpunkten befasst sich die Dissertation nicht mit der szenariobasierten Zulassung von hochautomatisierten Fahrsystemen, sondern mit der Integration des szenariobasierten Testprozesses in den automobilen Entwicklungsprozess. Daher wird im ersten Schritt der etablierte Entwicklungs- und Testprozess in der Automobilindustrie, sowie die Grundlagen des szenariobasierten Testens beschrieben und gegenübergestellt. Daraus ergeben

sich Herausforderungen bei der Bestimmung des Reifegrads in einem szenario-basierten Ansatz, wie beispielsweise das Fehlen von Methoden zur Bewertung von FAS in Fahrszenarien oder die Diskontinuität von logischen Szenarien. Aus den Herausforderungen werden Anforderungen an ein Konzept zum Testen von Fahrerassistenzsystemen in Fahrszenarien abgeleitet.

Anschließend wird ein Konzept entwickelt, welches sowohl die aufgestellten Anforderungen erfüllt als auch Antworten auf die zugrundeliegenden Forschungsfragen liefert. Das vorgestellte Testkonzept besteht aus drei Modulen und ermöglicht die Bestimmung des Reifegrads eines Fahrerassistenzsystems in seiner ODD. Das erste Modul übernimmt die Bewertung des Fahrerassistenzsystems in konkreten Fahrszenarien, das zweite Modul überprüft, ob die durch Parametervariation erzeugten konkreten Szenarien inhaltlich dem logischen Szenario entsprechen und das dritte Modul aggregiert die so erzeugten Informationen, um den Reifegrad zu bestimmen. Die Bewertung der Fahrerassistenzsysteme basiert auf der Grundidee des deklarativen Testens und überprüft das Systemverhalten mit sogenannten Systembewertungen. Diese sind so aufgebaut, dass sie vordefinierte Situationen automatisiert erkennen und das Fahrerassistenzsystem darin evaluieren. Durch die Trennung der Aktivierung und dem eigentlichen Test, kann eine Bewertung auch in a priori unbekanntem Fahrszenarien durchgeführt werden.

Auf Basis des Testkonzepts wird ein prototypisches Testsystem aufgebaut, mit dem ein Notbremssystem in einer Software-in-the-Loop Testumgebung ausgeführt wird. Das Notbremssystem wird exemplarisch in verschiedenen logischen Szenarien getestet und der Reifegrad des Notbremssystems für jedes logische Szenario bestimmt. Zusätzlich wurde der erarbeitete Bewertungsansatz im Rahmen einer Zusammenarbeit mit Mercedes-Benz an einem Hardware-in-the-Loop-Prüfstand implementiert und in einem Serenumfeld erprobt. Dabei wurden verschiedene Fahrerassistenzsysteme von Mercedes-Benz, wie beispielsweise ein Abstandsregeltempomat, ein Notbremssystem und die erste hoch automatisierten Fahrfunktion „DRIVE PILOT“ in digitalen Erprobungsfahrten bewertet.

Bei der anschließenden Evaluation hat sich gezeigt, dass die erreichbare Testtiefe und Testbreite bei dem vorgestellten Testkonzept höher war als bei vergleichbaren Ansätzen aus dem Stand der Wissenschaft und Technik. Darüber hinaus wurde gezeigt, dass mit dem Konzept sowohl bei der Erstellung als auch bei der Durchführung von Tests Kosten eingespart werden können.

Zusammenfassend stellen folgende Ergebnisse den wissenschaftlichen Beitrag der Dissertation dar:

- Durch Gegenüberstellung des szenariobasierten Testansatzes mit den Entwicklungsprozessen in der Automobilindustrie konnten die bestehenden Herausforderungen bei dessen Anwendung herausgearbeitet werden, wie beispielsweise die Diskontinuität von logischen Szenarien (vgl. Definition 3.2) oder mögliche falsch-positive Testergebnisse durch die Verwendung von systemunabhängigen Kritikalitätsmetriken.
- Der vorgeschlagene Ansatz zur dynamischen Bewertung ermöglicht die Bewertung von Fahrerassistenzsystemen in Fahrscenarien. Im Gegensatz zur sequentiellen Bewertung fester Testschritte in etablierten Konzepten, ermöglicht dieser Ansatz eine kontinuierliche Bewertung von Systemen und zeigt, dass der Ansatz des deklarativen Testens grundsätzlich für das szenariobasierte Testen verwendet werden kann. Im Vergleich zur Bewertung mittels Kritikalitätsmetriken, ermöglicht der Ansatz die Erreichung einer höheren Testtiefe.
- Die vorgelegten Ergebnisse zur Diskontinuität von Parameterräumen zeigen, dass die aktuellen Annahmen zu der Beziehung zwischen einem logischen Szenario und einem daraus abgeleiteten konkreten Szenario nicht zwingend gilt.
- Das vorgestellte Gesamtkonzept ermöglicht die Verbindung des szenariobasierten Testens mit der Reifegradbestimmung während des Entwicklungsprozesses.

7.2 Ausblick

Die vorliegende Dissertation liefert einen Beitrag zur Integration des szenariobasierten Testens in den automobilen Entwicklungsprozess, indem sie eine Methode zur Bestimmung des Reifegrads in logischen Szenarien und der ODD vorstellt. Um die erarbeiteten Konzepte in einem industriellen Umfeld anwenden zu können, muss allerdings untersucht werden wie Systembewertungen systematisch abgeleitet und definiert werden können, damit eine möglichst hohe Anforderungsabdeckung erreicht wird. Eine Möglichkeit wäre zu untersuchen, ob sich die Systemanforderungen so formalisieren lassen, dass sie den „abstrakten, virtuellen Zwilling“ bilden. Ein solcher „virtueller Zwilling“ könnte auch eine Art Schnittstelle oder Austauschformat, zwischen einer Zulassungsbehörde und einem Automobilhersteller, für die Freigabe von Fahrerassistenzsystemen oder eines hoch automatisierten Fahrsystems darstellen.

Ein weiteres Anwendungsgebiet für den erarbeiteten Bewertungsansatz wäre die Analyse von aufgezeichneten Fahrzeugdaten während realer Erprobungsfahrten. Dazu wurden bereits erste Versuche mit verschiedenen Datensätzen von Mercedes-Benz durchgeführt und eine grundsätzliche Anwendbarkeit des Ansatzes konnte gezeigt werden (siehe Abbildung 7.1). In einer Forschungs Kooperation mit Mercedes-Benz wird daher die Übertragbarkeit und Skalierbarkeit des Ansatzes auf reale Fahrzeugdaten weiterführend erforscht.

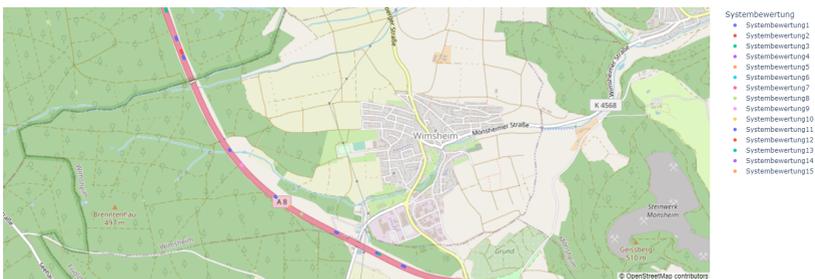


Abbildung 7.1: Kartenvisualisierung der Bewertung des „DRIVE PILOT“ auf Basis von aufgezeichneten Realdaten

Bei der Bestimmung des Reifegrads kann darüber hinaus evaluiert werden, ob jede Anforderung den gleichen Beitrag bei der Reifegradbestimmung besitzt oder ob eine Priorisierung von Anforderungen notwendig oder sinnvoll sein kann. Beispielsweise wäre denkbar, dass das sichere Verhindern von Kollisionen bei der Reifegradbestimmung mehr ins Gewicht fällt als die zuverlässige Ausgabe einer Warnung.

Durch die Betrachtung weiterer Fahrscenarien und Fahrmanöver kann ein fundierter Vergleich des regelbasierten und KI-basierten Ansatzes zur Erkennung der Fahrmanöver durchgeführt werden. In Zukunft könnte zusätzlich untersucht werden, wie ein unabhängiger Trainingsdatensatz mit gelabelten Fahrmanövern erstellt werden kann. Dieser Trainingsdatensatz ermöglicht dann die Verwendung des KI-basierten Ansatzes der Manövererkennung. Dafür müsste sowohl festgelegt werden welche Fahrmanöver für ein Fahrerassistenzsystem relevant sind als auch eine detaillierte Definition und Parametrisierung der Fahrmanöver vorgenommen werden. Diese relevanten Fahrmanöver können dann auch dazu genutzt werden um zu untersuchen, ob eine Priorisierung von Fahrmanövern den Vergleich von Manövermatrizen verbessert, indem nur noch Übergänge relevanter Fahrmanöver zu einem Aktwechsel führen.

Die Bestimmung einer validen Generalisierungskonfidenz ist die Voraussetzung für einen produktiven Einsatz des szenariobasierten Testens. Wenn nicht nachgewiesen werden kann, dass eine repräsentative Parameterraumabdeckung getestet wurde, dann kann von der Systemperformance in einzelnen konkreten Szenarien nicht auf die Performance im gesamten logischen Szenario geschlossen werden. Der aktuelle Entwurf eines Durchführungsrechtsakts der Europäischen Union für die Zulassung automatisierter Fahrzeuge beinhaltet im Abschnitt „2.2.4 Assumptions: Logical to concrete scenarios“ bereits erste Hinweise darauf, dass der Gesetzgeber zukünftig eine Art Generalisierungskonfidenz einfordern könnte. In dem Entwurf wird gefordert: „To ensure that the scenarios [...] are ready to be assessed through simulation or physical testing, the manufacturer may need to coherently parametrise them by applying assumptions. The manufacturer shall

provide evidence supporting the assumptions made [...]. [47]“. Die Generalisierungskonfidenz könnte dabei ein Baustein einer solchen Evidenz sein, mit der die getroffenen Annahmen belegt werden.

A Anhang

A.1 Zuordnung und Erfüllung der Anforderungen durch das erarbeitete Konzept

ID	Kurzbeschreibung	Erfüllt durch	Abschnitt
Allg1	Aussagen über den Entwicklungsstand eines Systems treffen können	Teilmodul für die Bestimmung des Entwicklungsstands	4.1 (S. 76)
Allg2	Bewertung von Funktionsverhalten in konkreten Szenarien	Teilmodul der Qualitätsbewertung	4.2.2 (S. 86)
Allg3	Validierung der konkreten Szenarien	Teilmodul für die Validierung der Szenarien	4.3 (S. 97)

Tabelle A.1: Abdeckung der aufgestellten allgemeinen Anforderungen

ID	Kurzbeschreibung	Erfüllt durch	Abschnitt
Bew1	Zuordnung von Bewertungen zu Systemanforderungen	Ableitung von Systembewertungen aus Anforderungen	4.2.2 (S. 86)
Bew2	Bewertung in zeitlich variablen Szenarien	Bewertung setzt sich durch Aktivierung und Prüfung zusammen	4.2.2 (S. 86)
Bew3	Kontinuierliche Bewertung über gesamtes Szenario	Verifier überprüft jeden Zeitschritt	4.2.2 (S. 86)
Bew4	Ermitteln der Dauer und Häufigkeit von Fehlern	Verwendung von einem Zustandsautomat	4.2.2 (S. 86)
Bew5	Bewertung in unterschiedlichen logischen Szenarien	Bewertung setzt sich durch Aktivierung und Prüfung zusammen	4.2.2 (S. 86)

Tabelle A.2: Abdeckung der aufgestellten Anforderungen an eine Bewertung von Fahrerassistenzsystemen

ID	Kurzbeschreibung	Erfüllt durch	Abschnitt
Val1	Berücksichtigung aller relevanter Aspekte	Analyse der Szenarienebenen	4.3 (S. 97)
Val2	Verwendbar für verschiedene Fahrerassistenzsysteme	Fahrmanöver können FAS zugeordnet werden	4.3 (S. 97)
Val3	Verwendbar für verschiedene logische Szenarien	Manövererkennung ist unabhängig vom logischen Szenario	4.3 (S. 97)

Tabelle A.3: Abdeckung der aufgestellten Anforderungen an eine Szenariovalidierung

ID	Kurzbeschreibung	Erfüllt durch	Abschnitt
Agg1	Berücksichtigung der Generalisierungsannahme	Einführung der Generalisierungskonfidenz	4.1 (S. 76)
Agg2	Aggregation über alle logischen Szenarien	Aggregation der szenario-spezifischen Reifegrade	4.1 (S. 76)

Tabelle A.4: Abdeckung der aufgestellten Anforderungen an eine Bestimmung des Reifegrads

A.2 Modellierung der Fahrmanöver

Manövertyp	Modellierung
ZustandLong	Beschleunigen: $(a > a_{th}) \wedge (a > 0) \wedge (v > 0)$ Geschwindigkeit halten: $(a < a_{th}) \wedge (v > v_{th}) \wedge (v > 0)$ Bremsen: $(a > a_{th}) \wedge (a < 0) \wedge (v > 0)$
Fahrtrichtung	Vorwärts fahren: $(v > v_{th}) \wedge (v \geq 0)$ Rückwärts fahren: $(v > v_{th}) \wedge (v < 0)$
Lenken	Lenkeingriff: Lenkradwinkel $\delta \geq \delta_{th}$ Kein Lenkeingriff: $\delta < \delta_{th}$
Infrastruktur	N/A: Trajektorie des Fahrzeugs überfährt kein Kartensegment, das einem Infrastrukturelement zugeordnet ist Infrastrukturelement annähern: Trajektorie des Fahrzeugs fährt bis zu einem angrenzendem Kartensegment, das einem Infrastrukturelement zugeordnet ist Infrastrukturelement überfahren: Trajektorie des Fahrzeugs überfährt Kartensegment, das einem Infrastrukturelement zugeordnet ist
Abbiegen	NA: Gesamtwinkeländerung der Trajektorie $\theta \leq \theta_{th}$ Abbiegen: $\theta > \theta_{th} \vee \theta < -\theta_{th}$
Spurmanöver	Spur folgen: Trajektorie liegt nur auf Spursegmenten, die einander folgen Spur wechseln: Trajektorie liegt nur auf Spursegmenten, die nebeneinander liegen

Tabelle A.5: Verwendete Modellierung der Fahrmanöver (1 von 2)

Manövertyp	Modellierung
U-Turn	U-Turn: $\theta = 180$ N/A: sonst
Parken	NA: Fahrzeug bewegt sich über den Verlauf der gesamten Trajektorie Einparken: Fahrzeug beendet die Bewegung während des Verlaufs der Trajektorie Ausparken: Fahrzeug beginnt die Bewegung während des Verlaufs der Trajektorie
Objektmanöver	Freifahrt: $(d > 0) \wedge ((d > d_{th}) \vee ((d < d_{th}) \wedge ((v_{Obj} - v_{Fhg}) > 0) \wedge ((a_{Obj} - a_{Fhg}) > 0)))$ Objekt annähern: $(d < d_{th}) \wedge ((v_{Fhg} - v_{Obj}) > v_{th})$ Objekt folgen: $(d < d_{th}) \wedge ((v_{Fhg} - v_{Obj}) < v_{th})$
Passieren	Objekt passieren: $(laneId_{Fzg} \neq laneId_{Obj}) \wedge (v_{Fzg} > v_{Obj})$ N/A: sonst
Kreuzen	Objekt kreuzen: $(Trajektorien_kreuzen = True) \wedge (0 < PET < PET_{max})$ N/A: sonst
Überholen	Objekt überholen: Abfolge der Manöver „Spur wechseln“, „Objekt passieren“ N/A: sonst
Einscheren	Einscheren: Abfolge der Manöver „Spur wechseln“, „Objekt folgen“ N/A: sonst
Ausscheren	Einscheren: Abfolge der Manöver „Objekt folgen“, „Spur wechseln“ N/A: sonst

Tabelle A.6: Verwendete Modellierung der Fahrmanöver (2 von 2)

A.3 Hyperparameter der KNN

Hyperparam.	Manöver	Longitudinale Fahrmanöver	Laterale Fahrmanöver
Gewählte Netzarchitektur		KNN	KNN
Dropout		0.3	0.5
Anzahl Neuronen in der ver- deckten Schichten		128	2048
Anzahl Epochen		4	4
Batchgröße		10	10
Optimierer		adam	adam
Fehlerfunktion		kategorische Kreuzentropie	kategorische Kreuzentropie
Metrik		accuracy	accuracy

Tabelle A.7: Gewählte Hyperparameter der KNN für den Vergleich der Ansätze

A.4 Vorstellung der Abbildungsfunktion

Die verwendete Abbildungsfunktion wird in den Tabellen A.8 bis A.10 dargestellt.

Ebene	Parameter	Wertedefinition
Ebene: L1	Kreuzungskontext	0: no_junction 1: junction
	Kreuzungstyp	0: none 1: T-crossing 2: 4-way crossing
	Anzahl Spuren	1, . . . N
Ebene: L2	Fußgängerüberweg vorhanden	0: false 1: true
	Ampel vorhanden	0: false 1: true
	Vorfahrtsstraße	0: false 1: true
	Tempolimit	0: none 30 50 ..

Tabelle A.8: Verwendete Abbildungsfunktion für die Szenario-Validierung (1 von 3)

Ebene: L4 Abbiegen	0: no turn 1: turn left 2: turn right
Spurwechsel	0: no lane change 1: lane change left 2: lane change right
Zielobjekt	0: no_preceeding 1: approach 2: follow
Typ Zielobjekt	0: car 1: bus 2: pedestrian 3: bicycle 4: truck
Maximale Geschwindigkeit Ego	0, . . . , n
Minimale Geschwindigkeit Ego	0, . . . , n
Kreuzendes Objekt	0: no_crossing 1: crossing
Anzahl Crossings	0, . . . ,4
Typ Objekt	0: car 1: bus 2: pedestrian 3: bicycle
Einscherendes Objekt	0: no_merging 1: merging

Tabelle A.9: Verwendete Abbildungsfunktion für die Szenario-Validierung (2 von 3)

Ebene: L4 Anzahl Mergings	0, . . . , 4
Typ Objekt	0: car 1: bus 2: pedestrian 3: bicycle
Abbiegendes Objekt	0: no_diverging 1: diverging
Anzahl Divergings	0, . . . , 4
Typ Objekt	0: car 1: bus 2: pedestrian 3: bicycle
Minimaler Abstand zum Zielobjekt (absolut)	0, . . . , n
Minimaler Abstand zum Zielobjekt in X-Richtung	0, . . . , n
Minimaler Abstand zum Zielobjekt in Y-Richtung	0, . . . , n
Minimale TTC	0, . . . , n
Minimale PET	0, . . . , n

Tabelle A.10: Verwendete Abbildungsfunktion für die Szenario-Validierung (3 von 3)

A.5 Implementierung einer Systembewertung

Listing A.1: Implementierung einer Systembewertung

```

1 def SB_E2Anf2_ActCond(data):
2     ttc = data['Controller_ACC.TTC[s]']
3     Sw_BrakeMode = data['Controller_ACC.Sw_BrakeMode[1Comfort|2
4     AEBOnly|3AEBOff]']
5     vehSpd = data['Plant.VehicleMovement.Velocities.Vehicle.
6     Linear.v_Total_Vehicle[km|h]']
7     ttc_lookup = {0:0,
8                 5:0.5,
9                 10:0.6,
10                20:0.73,
11                25:0.8,
12                30:0.979,
13                40:1.15,
14                50:1.35,
15                60:1.8}
16    boundaries = find_interval(ttc_lookup, vehSpd)
17    lower_key = boundaries[1]
18    upper_key = boundaries[0]
19    ttc_reference = ((ttc_lookup[upper_key] - ttc_lookup[
20    lower_key])/(upper_key - lower_key)) * (vehSpd - lower_key)
21    + ttc_lookup[lower_key]
22    ttc = round(ttc,4)
23    ttc_reference = round(ttc_reference,4) + 1.2
24    if ttc <= ttc_reference and Sw_BrakeMode != 3:
25        return 1
26    return 0
27
28 def SB_E2Anf2_TestCond(data):
29    FCW = data['Controller_ACC.FCW[0Off|1On]']
30    if FCW == 1:
31        return 1,{}
32    else:
33        return 0,{'FCW':str(FCW)}

```

A.6 Aufbau der Komponente „Bestimmung des Reifegrads“

A.6.2 Implementierte Sichten auf die Testergebnisse

Detail view for concrete scenario Share Export Session

Meta information for concrete scenario

Logical Scenario: Turn_left_with_crossing_p4d_ald
 Scenario Id: 11857
 Date: Aug 20, 2021, 7:41 p.m.
 Scenario valid: True
 Test platform: ESS_Rom_SIL

Assessment results

- SB_EZAnf3: **PASSED**
- SB_EZAnf2: **FAILED**
- SB_EZAnf1: **PASSED**
- SB_E1Anf3: **PASSED**
- SB_EZAnf1: **PASSED**
- SB_E0Anf1: **PASSED**
- SB_E1Anf1: **PASSED**
- SB_EZAnf4: **PASSED**

Simulation Scenario Parameter

EgoVehSpdInKmH: 21.0
 InitEgoVehSpdInKmH: 21.0
 PedSpdInMm: 1.5
 DistVehToPedInM: 18.0

Execute concrete test case

[Execute Test Case Again](#)

Scenario KPI

Abstracted Scenario description

- Crossing context: junction
 - Crossing type: none
 - Lane count: 0
 - Crosswalk passed: true
 - Traffic light passed: false
 - Driving on priority road: false
 - Speed limit: 0
- Turn maneuver: turn left
 - Lane change: no lane change
- Preceding obj maneuver: approach
- Preceding obj type: pedestrian
 - Max ego speed: 31
 - Min ego speed: 0
- Crossing obj context: crossing
- Crossing obj count: 1
- Crossing obj type: [{"Object_1": "pedestrian"}]
 - Merging obj context: no_crossing
 - Merging obj count: 0
 - Merging obj type: []
 - Diverging obj context: no_crossing
 - Diverging obj count: 0
 - Diverging obj type: []
 - Abs distance to obj: 2
 - Longitudinal distance to obj: 0
 - Lateral distance to obj: 0
 - TTC to obj: 0
 - PET to obj: 4

Logical Scenario description

- Crossing context: junction
 - Crossing type: 4-way crossing
 - Lane count: 1
 - Crosswalk passed: true
 - Traffic light passed: false
 - Driving on priority road: true
 - Speed limit: 0
- Turn maneuver: turn left
 - Lane change: no lane change
- Preceding obj maneuver: Free parameter space
- Preceding obj type: Free parameter space
 - Max ego speed: Free parameter space
 - Min ego speed: Free parameter space
- Crossing obj context: crossing
- Crossing obj count: 1
- Crossing obj type: [{"Object_1": "pedestrian"}]
 - Merging obj context: no_crossing
 - Merging obj count: 0
 - Merging obj type: []
 - Diverging obj context: no_crossing
 - Diverging obj count: 0
 - Diverging obj type: []
 - Abs distance to obj: Free parameter space
 - Longitudinal distance to obj: Free parameter space
 - Lateral distance to obj: Free parameter space
 - TTC to obj: Free parameter space
 - PET to obj: Free parameter space

Abbildung A.2: Screenshot der Detailansicht für ein konkretes Szenario

Logisches Szenario - Überblick

Start Zurück Szenario 1

Logisches Szenario: No_turn_with_crossing_car

Szenario Beschreibung

Überqueren der SL Kreuzung mit kreuzendem PKW

Bild: Grobes Bild des Szenarios

🖼️

Liste: Parameterset mit Range

Parameter	Einheit	Min. Werte	Max. Werte
EgWinkelSpehWinkel	km/h	50	25
TypWinkelSpehWinkel	km/h	50	25
StartPositionOrtRoute	m	10	5

Liste: Übersicht über die KPIs

KPI ID	KPI Name	Einheit	KPI Beschreibung
--------	----------	---------	------------------

Szenariospezifischer Reifegrad



Simulierte Szenarien

Absolute Anzahl simulierter Szenarien	Anzahl valider Szenarien	Anzahl valide Szenarien mit aktivem System
395	302	162

Übersicht Systembewertungen

Name	Aggregierter Anteil erfolgreicher Systembewertungen	Anzahl fehlgeschlagener Systembewertungen	Aggregiertes Ergebnis
SB E2aW3	1.0	0	True
SB E2aW2	1.0	0	True
SB E2aW1	1.0	0	True
SB E2aW5	0.855	21	False
SB E2aW7	0.099	164	False
SB E2aW6	0.917	12	False
SB E2aW8	1.0	0	True
SB E2aW4	1.0	0	True

Übersicht Anforderungen

Allgemeine Anforderungen

Name	Grad der Anforderungserfüllung
E2aW1 - Kollisionsfreiheit	0.917

Sense

Name	Grad der Anforderungserfüllung
E2aW1 - Erkennung VRU	1
E2aW1 - Bremsung bei Erkennung VRU	0.099
E2aW2 - Auswahl relevanter Objekte	1.0

Plan

Name	Grad der Anforderungserfüllung
E2aW2 - Anforderung FCW	1.0
E2aW3 - Anforderung Notbremsung	1.0
E2aW1 - Anforderung AEB	1.0
E2aW2 - Vermeidung Kollision	1.0

Act

Name	Grad der Anforderungserfüllung
E2aW4 - Umsetzung AEB	1.0
E2aW5 - Durchführung einer Notbremsung	0.855

Konfidenz der Testergebnisse

Generalisierungskonfidenz



Generalisierungskonfidenz

Testplattform	Aggregierte Testierkonfidenz	Anzahl simulierter Szenarien	Anteil Szenarien an allen getesteten Szenarien
ESS_Rom_SL	1.0	392	100.0%
HL	1.0	0	0.0%
ESS_Linea_SL	1.0	0	0.0%

Abbildung A.3: Screenshot der Detailansicht für ein logisches Szenario

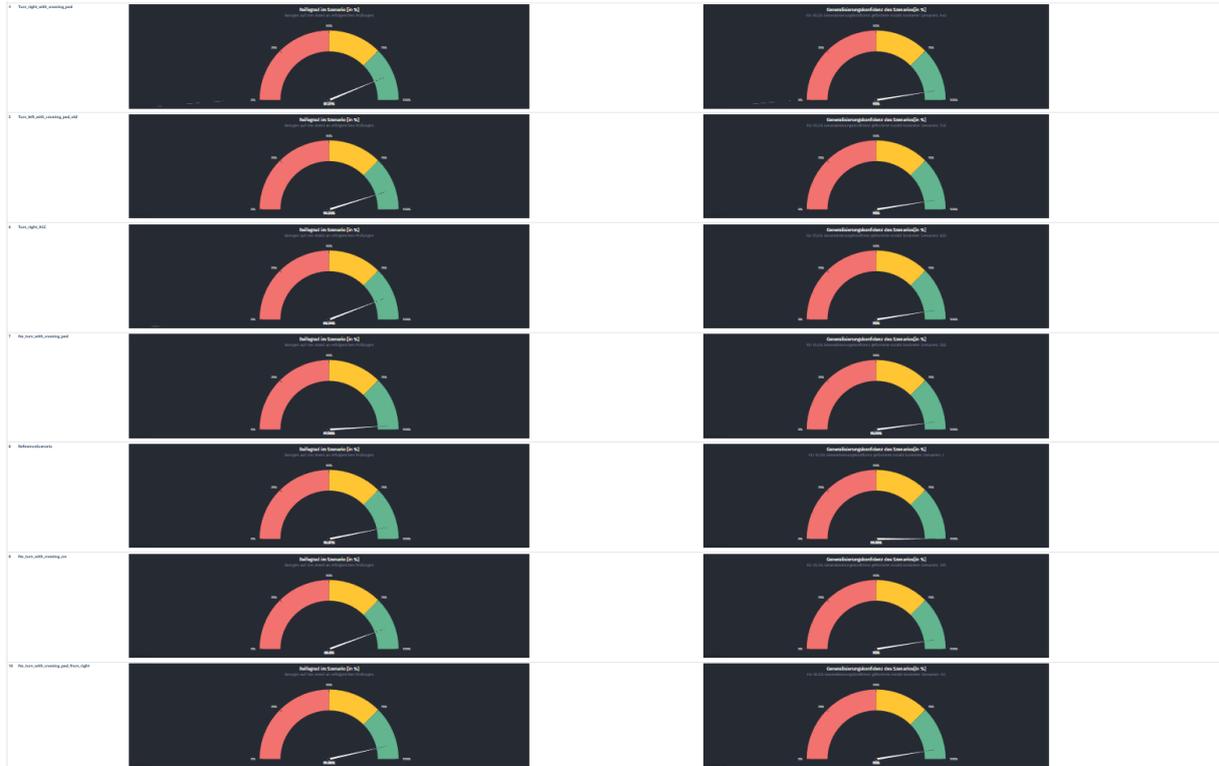


Abbildung A.4: Screenshot der Zusammenfassung der gesamten ODD

A.7 Vorstellung des automatisierten Notbremssystems

A.7.1 Anforderungen an das Notbremssystem

Anforderung an das gesamte Notbremssystem

E0Anf1 Das Notbremssystem muss Frontkollision mit anderen Verkehrsteilnehmern (VT) verhindern

Anforderungen auf Dekompositionsebene 1

E1Anf1 Ein VT, der sich vor dem Fahrzeug befindet, muss erkannt werden

E1Anf2 Im Falle einer bevorstehenden Frontkollision (TTC zum VT unterschreitet einen Grenzwert) muss diese durch eine Aktion (Warnung oder Bremsung) verhindert werden

E1Anf3 Wenn eine Aktion angefordert wird, muss sie auch durchgeführt werden

Anforderungen auf Dekompositionsebene 2

E2Anf1 Im Moment der Erkennung eines VT als relevantes Objekt muss der Abstand zum VT größer sein als der notwendige Bremsweg zum Stillstand

E2Anf2 Bei einer TTC von $1.x$ s (geschwindigkeitsabhängiger Wert) muss eine Warnung (FCW) ausgegeben werden

E2Anf3 Bei einer TTC von $1.x$ s (geschwindigkeitsabhängiger Wert) muss eine Notbremsung (AEB) angefordert werden

E2Anf4 Für eine Notbremsung muss eine Bremsanforderung von mindestens 5.0 m/s^2 an das Betriebsbremssystem des Fahrzeugs gerichtet werden

Anforderungen auf Dekompositionsebene 3

E3Anf1 Wenn ein VT sich vor dem Fahrzeug in der Fahrbahn befindet oder die Fahrbahn kreuzt, muss er als Zielobjekt ausgewählt werden

A.7.2 Abgeleitetes Systemverhalten

Bezeichnung: SV_E0Anf1 (abgeleitet aus Anforderung: E0Anf1)

Wenn: das Notbremssystem aktiv ist

Dann: darf es keine Kollision mit einem VT geben

Bezeichnung: SV_E1Anf1 (abgeleitet aus Anforderung: E1Anf1)

Wenn: der VT sich 10m vor dem Fahrzeug befindet

Dann: muss der VT erkannt werden

Bezeichnung: SV_E2Anf1 (abgeleitet aus Anforderung: E2Anf1)

Wenn: der VT erstmals als relevantes Objekt erkannt wird

Dann: muss der Abstand zum VT größer sein als der theoretisch erforderliche Bremsweg

Bezeichnung: SV_E3Anf1 (abgeleitet aus Anforderung: E3Anf1)

Wenn: sich ein erkannter VT in der Fahrspur befindet oder die Fahrspur kreuzt

Dann: muss der VT als relevant eingestuft werden

Bezeichnung: SV_E2Anf2 (abgeleitet aus Anforderungen: E1Anf2, E2Anf2)

Wenn: die TTC zum VT kleiner als ein geschwindigkeitsabhängiger Grenzwert ist und das Notbremssystem aktiv ist

Dann: muss eine Frontkollisionswarnung ausgegeben werden

Bezeichnung: SV_E2Anf3 (abgeleitet aus Anforderungen: E1Anf2, E2Anf3)

Wenn: die TTC zum VT kleiner als ein geschwindigkeitsabhängiger Grenzwert ist und das Notbremssystem aktiv ist

Dann: muss eine Notbremsung angefordert werden

Bezeichnung: SV_E1Anf3 (abgeleitet aus Anforderung: E1Anf3)

Wenn: eine Notbremsung angefordert wird

Dann: muss das Fahrzeug die Notbremsung durchführen

Bezeichnung: SV_E2Anf4 (abgeleitet aus Anforderung: E2Anf4)

Wenn: eine Notbremsung angefordert wird

Dann: muss die Sollverzögerung des Systems $-5 \frac{m}{s^2}$ betragen

A.7.3 Systembewertungen für das betrachtete Fahrerassistenzsystem

Bezeichnung: SB_E0Anf1 (überprüft Anforderung: E0Anf1)

Aktivierungsbedingung: 'Controller_ACC.AEB[0Off]1On]' == 1

Testbedingung: 'Controller_ACC.Collision[0Off]1On]' == 0

Bezeichnung: SB_E1Anf1 (überprüft Anforderung: E1Anf1)

Aktivierungsbedingung: dist_obj_groundtruth > 0 && dist_obj_groundtruth <= 10 && Object_Prev_Detected == False;

Testbedingung: 'Sensors.Object_Sensor_3D.Sensor_2_Front_Mid.SensorOutput.Dist_Object[m]1' > 0

Bezeichnung: SB_E2Anf1 (überprüft Anforderung: E2Anf1)

Aktivierungsbedingung: 'Target_Selection_ACC.Target_Detected[0]1]' == 1 && Object_Prev_Detected == False

Testbedingung: Dist_Object > Dist_Braking

Bezeichnung: SB_E3Anf1 (überprüft Anforderung: E3Anf1)

Aktivierungsbedingung: 'Target_Selection_ACC.LaneCrossingCheck[0]1]1' == 1 || 'Target_Selection_ACC.InLaneCheck[0]1]1' == 1

Testbedingung: 'Target_Selection_ACC.Target_Detected[0]1]' == 1

Bezeichnung: SB_E2Anf2 (überprüft Anforderungen: E1Anf2, E2Anf2)

Aktivierungsbedingung: 'Controller_ACC.TTC[s]' <= ttc_threshold && 'Controller_ACC.Sw_BrakeMode[1Comfort|2AEBOnly|3AEBOff]' != 3

Testbedingung: 'Controller_ACC.FCW[0Off]1On]' == 1

Bezeichnung: SB_E2Anf3 (überprüft Anforderungen: E1Anf2, E2Anf3)

Aktivierungsbedingung: 'Controller_ACC.TTC[s]' <= ttc_threshold && 'Controller_ACC.Sw_BrakeMode[1Comfort|2AEBOnly|3AEBOff]' != 3

Testbedingung: 'Controller_ACC.AEB[0Off]1On]' == 1

Bezeichnung: SB_E1Anf3 (überprüft Anforderung: E1Anf3)

Aktivierungsbedingung: 'Controller_ACC.AEB[0Off]1On]' == 1

Testbedingung: 'Plant.VehicleMovement.Accelerations.VehicleCoG.Linear.
a_Vehicle_CoG[x;y;z][m|s2].a_x_Vehicle_CoG[m|s2]' < 0.0

Bezeichnung: SB_E2Anf4 (überprüft Anforderung: E2Anf4)

Aktivierungsbedingung: 'Controller_ACC.AEB[0Off]1On]' == 1

Testbedingung: 'Controller_ACC.a_x_Set[m|s2]' < -5.0

A.8 Ergebnisse der Testdurchführung

A.8.1 Bewertungsergebnisse für die logischen Szenarien

Teilfunktion	Systembewertung	Anzahl fehlgeschlagener Bewertungen	Anteil erfolgreicher Bewertungen
Allgemein	SB_E0Anf1	67	88.5%
Umfeld erkennen	SB_E1Anf1	1	99.8%
Umfeld erkennen	SB_E2Anf1	95	83.4%
Umfeld erkennen	SB_E3Anf1	0	100%
Aktion planen	SB_E2Anf2	1	99.8%
Aktion planen	SB_E2Anf3	0	100%
Aktion ausführen	SB_E1Anf3	1	99.8%
Aktion ausführen	SB_E2Anf4	0	100%

Tabelle A.11: Ergebnisse der Systembewertung für das logische Szenario 1

Teilfunktion	Systembewertung	Anzahl fehlgeschlagener Bewertungen	Anteil erfolgreicher Bewertungen
Allgemein	SB_E0Anf1	50	88.8%
Umfeld erkennen	SB_E1Anf1	79	82.2%
Umfeld erkennen	SB_E2Anf1	27	89.2%
Umfeld erkennen	SB_E3Anf1	0	100%
Aktion planen	SB_E2Anf2	41	83.5%
Aktion planen	SB_E2Anf3	0	100%
Aktion ausführen	SB_E1Anf3	0	100%
Aktion ausführen	SB_E2Anf4	0	100%

Tabelle A.12: Ergebnisse der Systembewertung für das logische Szenario 2

Teilfunktion	Systembewertung	Anzahl fehlgeschlagener Bewertungen	Anteil erfolgreicher Bewertungen
Allgemein	SB_E0Anf1	12	91.7%
Umfeld erkennen	SB_E1Anf1	0	100%
Umfeld erkennen	SB_E2Anf1	164	9.9%
Umfeld erkennen	SB_E3Anf1	0	100%
Aktion planen	SB_E2Anf2	0	100%
Aktion planen	SB_E2Anf3	0	100%
Aktion ausführen	SB_E1Anf3	21	85.5%
Aktion ausführen	SB_E2Anf4	0	100%

Tabelle A.13: Ergebnisse der Systembewertung für das logische Szenario 3

Teilfunktion	Systembewertung	Anzahl fehlgeschlagener Bewertungen	Anteil erfolgreicher Bewertungen
Allgemein	SB_E0Anf1	98	73.6%
Umfeld erkennen	SB_E1Anf1	0	100%
Umfeld erkennen	SB_E2Anf1	0	100%
Umfeld erkennen	SB_E3Anf1	0	100%
Aktion planen	SB_E2Anf2	1	99.7%
Aktion planen	SB_E2Anf3	0	100%
Aktion ausführen	SB_E1Anf3	0	100%
Aktion ausführen	SB_E2Anf4	0	100%

Tabelle A.14: Ergebnisse der Systembewertung für das logische Szenario 4

Szenariomerkmal	LS1		LS2		LS3		LS4	
	Vorhanden	Nicht vorhanden						
Fahrtrichtung	840	0	528	0	395	0	390	0
Infrastruktur	840	0	528	0	395	0	390	0
Spurmanöver	N/A	N/A	528	0	N/A	N/A	N/A	N/A
Objektmanöver	840	0	528	0	395	0	390	0
Kreuzen	747	93	488	40	305	90	303	87

Tabelle A.15: Ergebnisse der Szenariovalidierung

A.9 Überprüfung der Systembewertungen

In Abbildung A.5 ist eine Messung der Testdurchführung für ein verwendetes Szenario dargestellt. Der obere Plot zeigt die Signalverläufe der Fahrzeuggeschwindigkeit (blau), des Abstands zum Fußgänger aus der Ground Truth (rot) und der vom Notbremssystem angeforderten Beschleunigung (grün). Im unteren Plot wird der Zeitabschnitt der Annäherung an den Fußgänger detailliert betrachtet. Das rote Signal zeigt die Anforderung der Notbremsung, das grüne Signal die Anforderung der Warnung und das gelbe Signal gibt an, ob eine Kollision vorliegt. Das Kollisions-Signal ist standardmäßig auf dem Wert 1 und wird erst auf den Wert 0 gesetzt, wenn das Notbremssystem ein Zielobjekt erkennt. Erfolgt dann eine Kollision, wechselt das Signal auf den Wert 1. Die Farbcodierung der Signale gilt auch für die Abbildung A.6 bis A.8.

A.9.1 KS1 - Szenario ohne Aktivierung der FCW und AEB

Im ersten Szenario wurden die Parameter so gewählt, dass das Notbremssystem weder eine Warnung ausgibt, noch eine Notbremsung durchführt. Aus der Messung werden folgende Zeitpunkte/Werte ermittelt, die für die Validierung der Bewertungsergebnisse notwendig sind.

- Zeitpunkt/Abstand bei Erkennung: 10.74s / 40.4m
- Zeitpunkt Fußgänger ist näher als 10m vom Fahrzeug entfernt: 14.28s
- Theoretisch berechneter Bremsweg bei 30km/h: 6.94m
- Real benötigter Bremsweg bei 30km/h in der Simulation: 8.80m

In Tabelle A.16 sind die Ergebnisse der verwendeten Systembewertungen dargestellt.



Abbildung A.5: Szenario mit der Parametrisierung Fahrzeuggeschwindigkeit 30 km/h, Abstand 10 m, Fußgängergeschwindigkeit 0.5 m/s

Bezeichnung	Testergebnis	Zeitpunkt/ Dauer Aktivierung	Zeitpunkt/Dauer Fehlerwirkung
SB_E0Anf1	Nicht aktiv	Nicht vorhanden	Nicht vorhanden
SB_E1Anf1	Passed	14.28s / 0.01s	Nicht vorhanden
SB_E2Anf1	Passed	10.74s / 0.01s	Nicht vorhanden
SB_E3Anf1	Passed	10.74s / 0.36s	Nicht vorhanden
SB_E2Anf2	Nicht aktiv	Nicht vorhanden	Nicht vorhanden
SB_E2Anf3	Nicht aktiv	Nicht vorhanden	Nicht vorhanden
SB_E1Anf3	Nicht aktiv	Nicht vorhanden	Nicht vorhanden
SB_E2Anf4	Nicht aktiv	Nicht vorhanden	Nicht vorhanden

Tabelle A.16: Ergebnisse der automatisierten Bewertung für das Szenario ohne aktives AEB und ohne Kollision

A.9.2 KS2 - Szenario mit Aktivierung der FCW und ohne AEB

Das zweite Szenario beinhaltet eine Aktivierung der FCW aber keine Aktivierung des AEB. Aus der Messung werden folgende Zeitpunkte/Werte ermittelt, die für

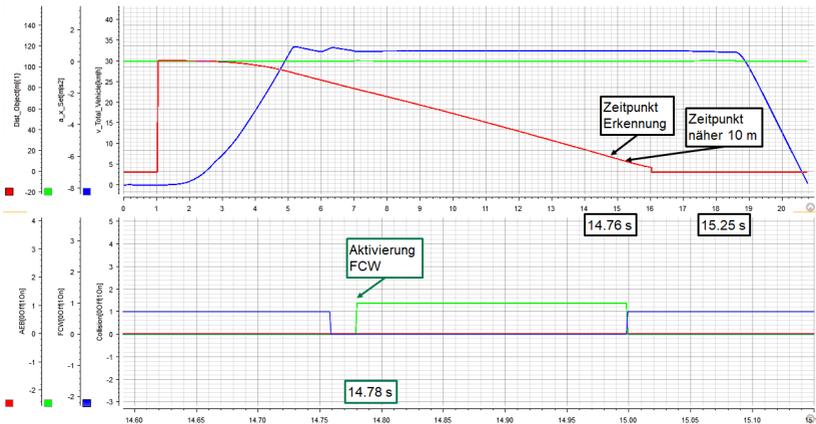


Abbildung A.6: Szenario mit der Parametrisierung Fahrzeuggeschwindigkeit 30 km/h, Abstand 25 m, Fußgängergeschwindigkeit 2.5 m/s

die Validierung der Bewertungsergebnisse notwendig sind.

- Zeitpunkt/Abstand bei Erkennung: 14.76s / 13.9m
- Zeitpunkt Aktivierung FCW: 14.78s, Ende FCW: 14.998s
- Zeitpunkt Fußgänger ist näher als 10m vom Fahrzeug entfernt: 15.25s
- Theoretisch berechneter Bremsweg bei 30km/h: 6.94m
- Real benötigter Bremsweg bei 30km/h in der Simulation: 8.80m

In Tabelle A.17 sind die Ergebnisse der verwendeten Systembewertungen dargestellt.

Bezeichnung	Testergebnis	Zeitpunkt/ Dauer Aktivierung	Zeitpunkt/Dauer Fehlerwirkung
SB_E0Anf1	Nicht aktiv	Nicht vorhanden	Nicht vorhanden
SB_E1Anf1	Passed	15.25s / 0.01s	Nicht vorhanden
SB_E2Anf1	Passed	14.76s / 0.01s	Nicht vorhanden
SB_E3Anf1	Passed	14.78s / 0.22s	Nicht vorhanden
SB_E2Anf2	Passed	14.78s / 0.22s	Nicht vorhanden
SB_E2Anf3	Nicht aktiv	Nicht vorhanden	Nicht vorhanden
SB_E1Anf3	Nicht aktiv	Nicht vorhanden	Nicht vorhanden
SB_E2Anf4	Nicht aktiv	Nicht vorhanden	Nicht vorhanden

Tabelle A.17: Ergebnisse der automatisierten Bewertung für das Szenario ohne aktives AEB und ohne Kollision

A.9.3 KS3 - Szenario mit Aktivierung AEB ohne Kollision

Das dritte Szenario beinhaltet sowohl die Aktivierung der FCW als auch des AEB, sowie die Verhinderung einer Kollision. Aus der Messung werden folgende Zeitpunkte/Werte ermittelt, die für die Validierung der Bewertungsergebnisse notwendig sind.

- Zeitpunkt/Abstand bei Erkennung: 14.16s / 19.29m
- Zeitpunkt Aktivierung FCW: 14.16s, Ende FCW: 17.0s
- Zeitpunkt Aktivierung AEB: 15.2s, Ende AEB: 17.0s
- Angeforderte Beschleunigung durch AEB: -7m/s^2
- Zeitpunkt Fußgänger ist näher als 10m vom Fahrzeug entfernt: 15.26s
- Theoretisch berechneter Bremsweg bei 30km/h: 6.94m

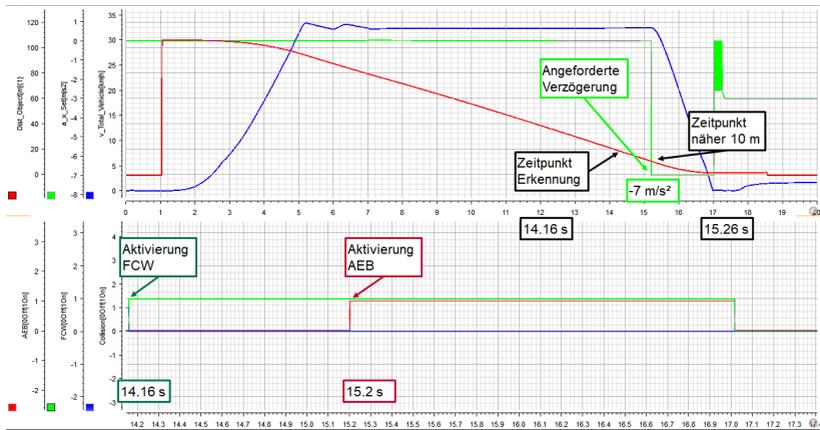


Abbildung A.7: Szenario mit der Parametrisierung Fahrzeuggeschwindigkeit 30 km/h, Abstand 20 m, Fußgängergeschwindigkeit 1 m/s

- Real benötigter Bremsweg bei 30km/h in der Simulation: 8.80m

In Tabelle A.18 sind die Ergebnisse der verwendeten Systembewertungen dargestellt.

A.9.4 KS4 - Szenario mit Aktivierung AEB und Kollision

Das vierte Szenario beinhaltet die Aktivierung des AEB, die eine Kollision nicht verhindern kann. Aus der Messung werden folgende Zeitpunkte/Werte ermittelt, die für die Validierung der Bewertungsergebnisse notwendig sind.

- Zeitpunkt/Abstand bei Erkennung: 15.3s / 8.27m
- Zeitpunkt Aktivierung FCW: 15.3s, Ende FCW: 17.02s
- Zeitpunkt Aktivierung AEB: 15.3s, Ende AEB: 17.02s
- Angeforderte Beschleunigung durch AEB: -7 m/s^2
- Zeitpunkt Kollision: 16.69s, Ende Kollision: 17.01s

Bezeichnung	Testergebnis	Zeitpunkt/ Dauer Aktivierung	Zeitpunkt/Dauer Fehlerwirkung
SB_E0Anf1	Passed	15.2s / 1.6s	Nicht vorhanden
SB_E1Anf1	Passed	15.26s / 0.01s	Nicht vorhanden
SB_E2Anf1	Passed	14.16s / 0.01s	Nicht vorhanden
SB_E3Anf1	Passed	14.16s / 3.4s	Nicht vorhanden
SB_E2Anf2	Passed	14.16s / 1.7s	Nicht vorhanden
SB_E2Anf3	Passed	15.2s / 1.6s	Nicht vorhanden
SB_E1Anf3	Passed	15.2s / 0.1s	Nicht vorhanden
SB_E2Anf4	Passed	15.2s / 0.01s	Nicht vorhanden

Tabelle A.18: Ergebnisse der automatisierten Bewertung für das Szenario mit aktivem AEB und ohne Kollision

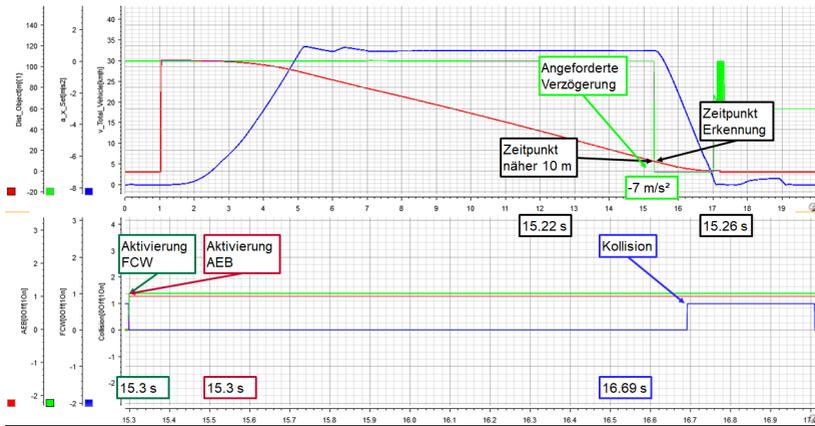


Abbildung A.8: Szenario mit der Parametrisierung Fahrzeuggeschwindigkeit 30 km/h, Abstand 10 m, Fußgängergeschwindigkeit 2 m/s

- Zeitpunkt Fußgänger ist näher als 10m vom Fahrzeug entfernt: 15.22s
- Theoretisch berechneter Bremsweg bei 30km/h: 6.94m

- Real benötigter Bremsweg bei 30km/h in der Simulation: 8.80m

In Tabelle A.19 sind die Ergebnisse der verwendeten Systembewertungen dargestellt.

Bezeichnung	Testergebnis	Zeitpunkt/ Dauer Aktivierung	Zeitpunkt/Dauer Fehlerwirkung
SB_E0Anf1	Fail	15.3s / 1.7s	16.69s / 0.32s
SB_E1Anf1	Passed	15.22s / 0.01s	Nicht vorhanden
SB_E2Anf1	Passed	15.3s / 0.01s	Nicht vorhanden
SB_E3Anf1	Passed	15.3s / 1.7s	Nicht vorhanden
SB_E2Anf2	Passed	15.3s / 1.7s	Nicht vorhanden
SB_E2Anf3	Passed	15.3s / 1.7s	Nicht vorhanden
SB_E1Anf3	Passed	15.3s / 1.7s	Nicht vorhanden
SB_E2Anf4	Passed	15.3s / 0.01s	Nicht vorhanden

Tabelle A.19: Ergebnisse der automatisierten Bewertung für das Szenario mit aktivem AEB und Kollision

Abkürzungsverzeichnis

- AB** Aktivierungsbedingungen
- ACC** adaptiver Abstandsregeltempomat
- ADS** Automated Driving Systems
- AEB** Autonomous Emergency Brake
- AEB** Autonomous Emergency Braking System
- ASIL** Automotive Safety Integrity Level
- CoC** Certificate of Conformity
- CoP** Conformity of Production
- DDT** Dynamic Driving Task
- ECU** Electronic Control Unit
- EG** Europäischen Gemeinschaft
- Euro NCAP** European New Car Assessment Programme
- FAS** Fahrerassistenzsysteme
- FCW** Front Collision Warning
- GIDAS** German In-Depth Accident Study
- HiL** Hardware-in-the-Loop
- HTTP** Hypertext Transfer Protocol

HW	Hardware
KBA	Kraftfahrt-Bundesamt
KNN	künstliche neuronale Netze
MiL	Model-in-the-Loop
MS	Meilensteine
N/A	Not Available
ODD	Operational Design Domain
OEM	Original Equipment Manufacturer
PB	Prüfbedingungen
PEP	Produktentwicklungsprozess
PET	Post Encroachment Time
QMS	Qualitätsmanagement-System
REST	Representational State Transfer
RG	Reifegrad
RGA	Reifegradabsicherung
SBT	szenariobasiertes Testen
SiL	Software-in-the-Loop
SOTA	Software Updates over the Air
SOTIF	Safety Of The Intended Functionality
StVZO	Straßenverkehrs-Zulassungs-Ordnung
SW	Software
TD	Technischer Dienst

TTC Time-to-Collision

UNECE United Nations Economic Commission for Europe

V2X Vehicle-to-Everything

V&V Verifikation und Validierung

VDA Verbandes der Automobilindustrie

VVM V&V-Methoden

XiL X-in-the-Loop

Abbildungsverzeichnis

1.1	Übersicht ausgewählter Meilensteine bei der Entwicklung des automatisierten Fahrens	2
2.1	Automatisierungsgrade des automatisierten Fahrens aus [205]	11
2.2	Neue Verkehrszeichen der StVO Novelle 2020 nach [105]. Überholverbot von Zweirädern (links), Mehrfachbesetzte Pkw (Mitte) und Parkplatz für Carsharing (rechts)	13
2.3	Schematische Übersicht eines PEP der Volkswagen AG mit Reifegradstufen und exemplarischer PEP-Meilensteine nach [6]	14
2.4	Beispielhafter Feature Roll-out Plan für eine automatisierte Fahrfunktion	16
2.5	Sturktur des Systementwurfs nach dem V-Modell [34]	19
2.6	Integration der Entwicklung nach dem V-Modell in den Produktentstehungsprozess nach [53]	21
2.7	Darstellung der Ursachenkette von einem Fehler nach ISO29119 [99]	22
2.8	Klassifikation der möglichen Fahrzustände beim hochautomatisiertem Fahren nach [189]	24
2.9	Grundsätzlicher Aufbau der Testspezifikation eines deutschen Automobilherstellers	28
2.10	Exemplarische Beschreibung und Visualisierung eines Testfalls für ein Notbremsystem	29
2.11	Systemdekomposition und Integrations Schritte nach [89]	30
2.12	Zuordnung der Musterphasen zu den einzelnen Reifegraden nach [53]	32
2.13	Möglichkeiten zur Bewegung von Fußgängern und Fahrradfahrern auf dem Prüfgelände (links: Freeboard aus [4], rechts: Seilsystem aus [3])	36

2.14	Exemplarische Visualisierung verschiedener Einflussfaktoren auf unterschiedliche Fahrerassistenzsysteme	37
2.15	Darstellung des 6-Ebenen-Modells zur Beschreibung von Szenarien nach [156]	40
2.16	Akt-basierte Darstellung eines Notbrems-Szenarios mit kreuzendem Fußgänger nach [15]	41
2.17	Darstellung der Abstraktionsebenen von Szenarien nach [74]	42
2.18	Szenariobasierter Testprozess zur Freigabe von ADS aus [154]	43
2.19	Definition eines Testfalls für das szenariobasierte Testen nach [190]	44
2.20	Darstellung der Definition der Zeitlücke nach [9]	46
2.21	Darstellung der Definition der PET nach [204]	46
2.22	Ablaufplan zur Erteilung einer EG-Fahrzeugtypgenehmigung nach [117]	48
2.23	Abhängigkeiten und Verweise im Zulassungsrecht nach [55] [203]	49
3.1	Darstellung des deklarativen Testens nach [198]	57
3.2	Erweiterung der Testfalldefinition nach [190] um Systembewertungen	61
3.3	Initiale Geschwindigkeit des Fahrzeugs bei Unfällen mit Fußgängern nach [195]	66
3.4	Visualisierung der validen und nicht validen Parameterkombinationen für der simulierten Szenarien	69
3.5	Visualisierung der Diskontinuität logischer Szenarien	70
4.1	Strukturierung der notwendigen Module zum Testen von Fahrerassistenzsystemen in Fahrscenarien	76
4.2	Darstellung der Begriffe Testbreite und Testtiefe	77
4.3	Mögliche Modelle für die Zunahme der Generalisierungskonfidenz c_{G,S_L} in Abhängigkeit der Anzahl an getesteten konkreten Szenarien N_{S_K}	80
4.4	Zunahme der Generalisierungskonfidenz nach Gleichung 4.4 für verschiedene Parameter n. Rot ($N_{K,95\%} = 10$), blau ($N_{K,95\%} = 50$), grün ($N_{K,95\%} = 100$)	81
4.5	Vervollständigung des Testkonzepts um die Bestimmung des Reifegrads	82

4.6	Modellierung des Notbremszenarios zu verschiedenen Zeitpunkten während des Szenarioverlaufs	84
4.7	Formalisierung des erwarteten Verhaltens in Systembewertungen, die sich aus einer Aktivierung und einer Prüfung zusammensetzen	88
4.8	Aktivitätsdiagramm für die Bewertung eines ACC-Systems in Fahrscenarien	89
4.9	Beispiel für ein Querregelungssystem mit der Solltrajektorie als grauer Pfeil und der gefahrenen Trajektorie als grüner Pfeil, links: Regelung folgt der vorgegebenen Trajektorie, rechts: Regler schwingt um vorgegebene Trajektorie	89
4.10	Aktivitätsdiagramm des Verifiers für die Ausführung der automatisierten Systembewertung	90
4.11	Visualisierung der Aktivierungszeitpunkte der FCW und der prädizierten Zeitpunkte	93
4.12	Visualisierung der Aktivierungszeitpunkte der FCW und der FCW-Prüfung	95
4.13	Vervollständigung des Testkonzepts um die Bewertung Fahrerassistenzsystemen in Fahrscenarien	96
4.14	Konzept zur Identifikation der Manöver „Freifahrt, „Objekt folgen“ und „Objekt annähern“ mittels künstlicher neuronaler Netze .	103
4.15	Vergleich der erkannten Fahrmanöver zwischen einem Ansatz mit einem Feedforward KNN (oben) und dem regelbasierten Modell (unten). Der obere Vergleich zeigt die longitudinalen Fahrmanöver und der Untere die lateralen Fahrmanöver.	105
4.16	Anordnung der Fahrmanöver in einem morphologischen Kasten zur Beschreibung von Akten	107
4.17	Einfache Manöversequenz für drei Manövertypen	108
4.18	Geringe zeitliche Variationen in den Manöversequenz führen zur Bildung neuer Akte	110
4.19	Verteilung der Aktlängen für die Szenarioerkennung der Kreuzung <i>DR_USA_Intersection_EP</i> aus dem INTERACTION Datensatz	111
4.20	Einfluss der Anzahl an Akten auf die Anzahl an erkannten logischen Szenarien	111

4.21	Ergebnisse der Analyse, welchen quantitativen Einfluss der Manövertyp und die minimale Aktdauer auf die Anzahl an extrahierten logischen Szenarien hat	112
4.22	Konzept zum Testen von Fahrerassistenzsystemen in Fahrscenarien bestehend aus einer Bewertung der FAS, Validierung der Szenarien und der Bestimmung des Reifegrads	116
5.1	Komponentendiagramm für die prototypische Umsetzung des vorgestellten Konzepts zum Testen von Fahrerassistenzsystemen in Fahrscenarien	117
5.2	Klassendiagramm des Verifiers	119
5.3	Systemarchitektur des verwendeten Fahrerassistenzsystems heruntergebrochen auf verschiedene Dekompositionsebenen und mit abgeleiteten Anforderungen	122
5.4	Zustandsdiagramm mit den Systemzuständen des automatisierten Notbremssystems	123
5.5	Visualisierung der ausgewählten Fahrscenarien für die prototypische Umsetzung mit Unfalltypenreferenz nach [150]. Der blaue PKW ist mit dem zu testenden automatisierten Notbremssystem ausgestattet.	124
5.6	HiL-Prüfstand für Fahrerassistenzsysteme von Mercedes-Benz, an dem der vorgeschlagene Bewertungsansatz prototypisch erprobt werden konnte (Bild aus [213])	128
5.7	Architektur des Wirkketten-HiL für Fahrerassistenzsysteme mit der Erweiterung der digitalen Erprobungsfahrt (Abbildung angelehnt an [114])	130
5.8	Workflow für die Testdurchführung am HiL-Prüfstand	132
5.9	Anzahl der Systembewertungen für eine Auswahl der betrachteten Fahrerassistenzsysteme im Rahmen der prototypischen Umsetzung am HiL-Prüfstand	133
6.1	Visualisierung von zwei nicht validen konkreten Szenarien für das <i>LS1</i> 141	
6.2	Parameterraum des ersten logischen Szenarien mit Systemaktivierungen	143

6.3	Gegenüberstellung der simulierten Szenarien (grau) mit den Ergebnissen der Bewertung durch die Kritikalitätsmetrik „Kollisionsfreiheit“ (rot) und dem vorgestellten Testkonzept (vgl. Abbildung 4.22) (grün, gelb)	146
6.4	Beispielrechnung für die Zeit- und Geldersparnis mit dem erarbeiteten Ansatz im Vergleich zu etablierten Testfällen	150
7.1	Kartenvisualisierung der Bewertung des „DRIVE PILOT“ auf Basis von aufgezeichneten Realdaten	160
A.1	Datenbankmodell der Umsetzung	175
A.2	Screenshot der Detailansicht für ein konkretes Szenario	176
A.3	Screenshot der Detailansicht für ein logisches Szenario	177
A.4	Screenshot der Zusammenfassung der gesamten ODD	178
A.5	Szenario mit der Parametrisierung Fahrzeuggeschwindigkeit 30 km/h, Abstand 10 m, Fußgängergeschwindigkeit 0.5 m/s	190
A.6	Szenario mit der Parametrisierung Fahrzeuggeschwindigkeit 30 km/h, Abstand 25 m, Fußgängergeschwindigkeit 2.5 m/s	191
A.7	Szenario mit der Parametrisierung Fahrzeuggeschwindigkeit 30 km/h, Abstand 20 m, Fußgängergeschwindigkeit 1 m/s	193
A.8	Szenario mit der Parametrisierung Fahrzeuggeschwindigkeit 30 km/h, Abstand 10 m, Fußgängergeschwindigkeit 2 m/s	194

Tabellenverzeichnis

2.1	Übersicht der Fahrerassistenzsysteme des EQS unterteilt in Hauptfunktionalität und zusätzliche Funktionalität nach [83]	15
2.2	Definition der Reifegradampel nach [53]	17
2.3	Liste an Regelungen für die Zulassung von ausgewählten FAS nach [69]	50
3.1	Gegenüberstellung welches Funktionsverhalten der Fahrerassistenzsysteme aus Abbildung 2.1 mit gängigen Kritikalitätsmetriken bewertet werden kann.	62
3.2	Übersicht und Beschreibung der betrachteten Unfallszenarien	65
3.3	Parameterbereiche und Schrittweiten für die durchgeführten Szenarien	67
3.4	Ergebnisse der Szenarivalidierung	69
4.1	Auswertung des Prädiktionsfehlers	92
4.2	Auswertung der zeitlichen Abweichung der Prüfungsergebnisse	94
4.3	Ergebnis der Evaluierung, welche Ebenen des 6-Ebenen-Modells für die Validierung betrachtet werden müssen	98
4.4	Fahrmanöverkatalog mit Fahrmanövern nach [142] [64] [88]	100
4.5	Vergleich der verschiedenen Netzarchitekturen hinsichtlich relevanter Kriterien für die Erkennung von Fahrmanövern	104
5.1	Zu überprüfende Merkmale der getesteten logischen Szenarien	125
5.2	Parameterbereiche und Schrittweiten für die logischen Szenarien <i>LS1</i> und <i>LS2</i>	126
5.3	Parameterbereiche und Schrittweiten für die logischen Szenarien <i>LS3</i> und <i>LS4</i>	127
5.4	Übersicht der bewerteten Fahrerassistenzsysteme mit dem überprüften Systemverhalten	134

6.1	Ergebnisse der Überprüfung der definierten Systembewertungen in den konkreten Szenarien KS1-KS4	136
6.2	Ergebnisse der Systembewertungen mit Anzahl an fehlgeschlagenen Bewertungen und der durchschnittlichen Dauer der Fehlerwirkung	139
6.3	Ergebnisse der Szenariovalidierung für die vier logischen Szenarien und die zu überprüfenden Merkmale aus der Tabelle 5.1	140
6.4	Ergebnisse der Reifegradbestimmung für die vier betrachteten logischen Szenarien und die ODD aus den Testergebnissen der Tabellen A.11-A.14	144
6.5	Vergleich der Bewertung durch Kritikalitätsmetriken mit den Ergebnissen aus dem vorgestellten Testkonzept	147
6.6	Vergleich des testbaren Funktionsverhaltens mittels gängiger Kritikalitätsmetriken und dem erarbeiteten Testkonzept für ausgewähltes Funktionsverhaltens aus Tabelle 5.4	148
A.1	Abdeckung der aufgestellten allgemeinen Anforderungen	163
A.2	Abdeckung der aufgestellten Anforderungen an eine Bewertung von Fahrerassistenzsystemen	164
A.3	Abdeckung der aufgestellten Anforderungen an eine Szenariovalidierung	165
A.4	Abdeckung der aufgestellten Anforderungen an eine Bestimmung des Reifegrads	165
A.5	Verwendete Modellierung der Fahrmanöver (1 von 2)	167
A.6	Verwendete Modellierung der Fahrmanöver (2 von 2)	168
A.7	Gewählte Hyperparameter der KNN für den Vergleich der Ansätze	169
A.8	Verwendete Abbildungsfunktion für die Szenario-Validierung (1 von 3)	170
A.9	Verwendete Abbildungsfunktion für die Szenario-Validierung (2 von 3)	171
A.10	Verwendete Abbildungsfunktion für die Szenario-Validierung (3 von 3)	172
A.11	Ergebnisse der Systembewertung für das logische Szenario 1	184
A.12	Ergebnisse der Systembewertung für das logische Szenario 2	185
A.13	Ergebnisse der Systembewertung für das logische Szenario 3	186

A.14 Ergebnisse der Systembewertung für das logische Szenario 4	187
A.15 Ergebnisse der Szenariovalidierung	188
A.16 Ergebnisse der automatisierten Bewertung für das Szenario ohne aktives AEB und ohne Kollision	190
A.17 Ergebnisse der automatisierten Bewertung für das Szenario ohne aktives AEB und ohne Kollision	192
A.18 Ergebnisse der automatisierten Bewertung für das Szenario mit aktivem AEB und ohne Kollision	194
A.19 Ergebnisse der automatisierten Bewertung für das Szenario mit aktivem AEB und Kollision	195

Literaturverzeichnis

- [1] Mercedes-Benz AG. *First internationally valid system approval for conditionally automated driving*.
- [2] Mercedes-Benz AG. *Introducing DRIVE PILOT: An Automated Driving System for the Highway*. 2019.
- [3] Mercedes Benz AG. *Der Aktive Brems-Assistent reagiert auf vorrausfahrende Fahrradfahrer, zum Beispiel wenn diese plötzlich in die Fahrbahn scheren*. 2020.
- [4] Mercedes Benz AG. *Die Fahrassistenzsysteme der neuen S-Klasse*. 2020.
- [5] Mercedes-Benz AG. *Mercedes-Benz Strategy Update*. 2020.
- [6] Volkswagen AG. *Das gemeinsame Qualitätsmanagement in der Lieferkette Produktentstehung*. 2014.
- [7] Volkswagen AG. *NEW AUTO Strategy Presentation*. 2021.
- [8] Albert Albers, Tobias Düser, Oliver Sander, Christoph Roth, and Josef Henning. X-in-the-Loop-Framework für Fahrzeuge, Steuergeräte und Kommunikationssysteme. *ATZelektronik volume 5*, 2010.
- [9] Jiri Ambros. Free-flow vs car-following speeds: Does the difference matter? In *Advances in Transportation Studies*, 2016.
- [10] Christian Amersbach and Hermann Winner. Defining required and feasible test coverage for scenario-based validation of highly automated vehicles. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019.

- [11] ASAM. *ASAM OpenSCENARIO*. 2021.
- [12] ASAM. *ASAM OpenSCENARIO V2.0*. 2021.
- [13] Johannes Bach. *Methoden und Ansätze für die Entwicklung und den Test prädiktiver Fahrzeugregelungsfunktionen*. PhD thesis, Karlsruher Instituts für Technologie (KIT), 2018.
- [14] Johannes Bach, Marc Holzaepfel, Stefan Otten, and Eric Sax. Reactive-replay approach for verification and validation of closed-loop control systems in early development. In *SAE World Congress Experience*, 2017.
- [15] Johannes Bach, Stefan Otten, and Eric Sax. A model-based scenario specification method to support development and test of automated driving functions. In *IEEE Intelligent Vehicles Symposium*, 2016.
- [16] Johannes Bach, Stefan Otten, and Eric Sax. A taxonomy and systematic approach for automotive system architectures. In *International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS)*, 2017.
- [17] Yvonne Barnard, Satu Innamaa, Sami Koskinen, Helena Gellerman, Erik Svanberg, and Haibo Chen. Methodology for field operational tests of automated vehicles. In *Transportation Research Procedia*, 2016.
- [18] Earl T. Barr, Mark Harman, Phil McMinn, Muzammil Shahbaz, and Shin Yoo. The oracle problem in software testing: A survey. *IEEE Transactions on Software Engineering*, 41(5):507–525, 2015.
- [19] Bettina Bartels and Christian T. Erbsmehl. *Bewegungsverhalten von Fußgängern im Straßenverkehr – Teil 1*. 2014.
- [20] Christian Bartelt. *V-Modell XT - Das deutsche Referenzmodell für Systementwicklungsprojekte Version: 2.0*. 2006.
- [21] Thomas Beck, Clemens Reichmann, and Jörg Schäuuffele. E/E-Entwicklung für zukünftige Fahrzeuginnovationen: Ein integrierter, modellbasierter Ansatz. *ATZ Elektronik*, 2016.

- [22] Halil Beglerovic, A. Ravi, N. Wikström, H.-M. Koegeler, A. Leitner, and J. Holzinger. Model-based safety validation of the automated driving function highway pilot. In Prof. Dr. Peter E. Pfeffer, editor, *8th International Munich Chassis Symposium 2017*, pages 309–329, Wiesbaden, 2017. Springer Fachmedien Wiesbaden.
- [23] Hanna Bellem. *Comfort in Automated Driving: Analysis of Driving Style Preference in Automated Driving*. PhD thesis, Technische Universität Chemnitz, 2018.
- [24] James Bender and Jeff McWherter. *Professional Test Driven Development with C# : Developing Real World Applications with TDD*. John Wiley & Sons, Incorporated, 2011.
- [25] Herbert Benington. Production of large computer programs. *Symposium on advanced programming methods for digital computers*, 1956.
- [26] Jonas Bereisa. Applications of microcomputers in automotive electronics. *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, 1983.
- [27] Roland Berger. *Global Automotive Supplier Study 2018*. 2017.
- [28] Georg Bergweiler, Jan Ole Hansen, and Marc Dörfer. Agile Entwicklung mit physischen Prototypen zur besseren Projektplanung. *ATZ - Automobiltechnische Zeitschrift*, 2019.
- [29] R. Bernotat and KP. Gärtner. *Anthropotechnische Gesichtspunkte bei der Gestaltung der Kommunikation Zwischen Mensch und Hochautomatisierten Systemen*. Springer, 1980.
- [30] Muhammad Wasim Bhatti and Farah Hayat. An investigation of changing requirements with respect to development phases of a software project. *International Conference on Computer Information Systems and Industrial Management Applications*, 2010.

- [31] Mario Bijelic, Tobias Gruber, and Werner Ritter. Benchmarking image sensors under adverse weather conditions for autonomous driving. In *IEEE Intelligent Vehicles Symposium (IV)*, 2018.
- [32] Mario Bijelic, Christian Muench, Werner Ritter, Yuri Kalnishkan, and Klaus Dietmayer. Robustness against unknown noise for raw data fusing neural networks. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018.
- [33] BMI BMVg. *Entwicklungsstandard für IT-Systeme des Bundes*, chapter Zusammenfassende Übersicht über die Änderungen am V-Modell zwischen Version August 1992 und Juni 1997. BMVg, 1997.
- [34] BMI BMVg. *Entwicklungsstandard für IT-Systeme des Bundes*, chapter Einordnung des V Modells in sein Umfeld. BMVg, 1997.
- [35] BMI BMVg. *Entwicklungsstandard für IT-Systeme des Bundes*, chapter Rollenkonzept im V Modell. BMVg, 1997.
- [36] BMI BMVg. *Entwicklungsstandard für IT-Systeme des Bundes*, chapter Anwendung des V-Modells und der ITSEC. BMVg, 1997.
- [37] Barry W. Boehm. Guidelines for verifying and validating software requirements and design specifications. *Euro IFIP*, 1979.
- [38] Eckard Bringmann and Andreas Krämer. Model-based testing of automotive systems. In *2008 1st International Conference on Software Testing, Verification, and Validation*, pages 485–493, 2008.
- [39] Bart Broekman and Edwin Notenboom. *Testing Embedded Software*. Addison-Wesley, 2002.
- [40] Statistisches Bundesamt. *Berufspendler*. 2021.
- [41] Ondrej Burkacky, Johannes Deichmann, and Jan Paul Stein. *Automotive software and electronics 2030*. 2019.
- [42] Ilene Burnstein. *Practical Software Testing*. Springer, 2003.

- [43] Andreas Bussler, Nicolas Wagener, Andreas Pütz, Till Menzel, Thilo Braun, and Julian Fuchs. A story of scenarios – from data and knowledge to scenarios for testing. In *Halbzeitevent VV-Methoden*, 2022. Halbzeitevent VV-Methoden.
- [44] Alessandro Calò, Paolo Arcaini, Shaukat Ali, Florian Hauer, and Fuyuki Ishikawa. Generating avoidable collision scenarios for testing autonomous driving systems. In *2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST)*, 2020.
- [45] Robert N. Charette. This car runs on code. *IEEE Spectrum*, 2009.
- [46] Murali Chemuturi. *Software Design : A Comprehensive Guide to Software Development Projects*. Chapman and Hall/CRC, 2018.
- [47] European Commission. *Automated cars – technical specifications - ANNEXES to the Commission Implementing Regulation*. 2022.
- [48] Mirko Conrad, Ines Fey, and Sadegh Sadeghipour. Systematic model-based testing of embedded automotive software. *Electronic Notes in Theoretical Computer Science*, 111:13–26, 2005. Proceedings of the Workshop on Model Based Testing (MBT 2004).
- [49] Yanja Dajsuren and Mark van den Brand. *Automotive Systems and Software Engineering : State of the Art and Future Trends*, chapter Automotive Software Engineering: Past, Present, and Future. 2019.
- [50] João Carlos R. de Borba, Luís Gonzaga Trabasso, and Marcus Vinicius P. Pessôa. Agile management in product development. *Research-Technology Management*, 62(5):63–67, 2019.
- [51] Erwin de Gelder and Jan-Pieter Paardekooper. Assessment of automated driving systems using real-life scenarios. *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017.
- [52] Verband der Automobilindustrie e.V. (VDA). *Das gemeinsame Qualitätsmanagement in der Lieferkette - Produktentstehung*. 2009.

- [53] Verband der Automobilindustrie e.V. (VDA). *Reifegradabsicherung für Neuteile*. 2021.
- [54] Beauftragter der Bundesregierung für Informationstechnik. *Häufig gestellte Fragen zum V-Modell XT*. 2021.
- [55] Bundesministeriums der Justiz und für Verbraucherschutz. *Straßenverkehrs-Zulassungs-Ordnung*. 2012.
- [56] Bundesverband deutscher Omnibusunternehmer. *Präsentation der Konjunkturumfrage 2014/2015*. 2014.
- [57] Anke Drappa, Peter Huber, and Jon Vollmar. Automated test program generation for automotive devices. In *2010 IEEE International Test Conference*, pages 1–10, 2010.
- [58] Duden. *Duden – Deutsches Universalwörterbuch*. Bibliographisches Institut, 2020.
- [59] Adrian Ebner. *Referenzszenarien als Grundlage für die Entwicklung und Bewertung von Systemen der Aktiven Sicherheit*. PhD thesis, Fakultät V – Verkehrs- und Maschinensysteme der Technischen Universität Berlin, 2014.
- [60] Thomas W. Edgar and David O. Manz. Chapter 2 - science and cyber security. In Thomas W. Edgar and David O. Manz, editors, *Research Methods for Cyber Security*, pages 33 – 62. Syngress, 2017.
- [61] Philip Elspas, Yannick Klose, Simon Isele, Johannes Bach, and Eric Sax. Time series segmentation for driving scenario detection with fully convolutional networks. In *International Conference on Vehicle Technology and Intelligent Transport Systems*, 2021.
- [62] Peter Eyerer. Zukunft erfolgreich gestalten - Inventionen zu Innovationen machen. In *Wirtschaftsforum, IKP, Universität Stuttgart*, 2008.
- [63] Olga Filipova and Rui Vilão. *Software Development From A to Z*. Apress, 2018.

- [64] Jonas Firl. *Probabilistic Maneuver Recognition in Traffic Scenarios*. PhD thesis, Karlsruher Institut für Technologie (KIT), 2014.
- [65] Jannik Fischbach, Andreas Vogelsang, Dominik Spies, Andreas Wehrle, Maximilian Junker, and Dietmar Freudenstein. Specmate: Automated creation of test cases from acceptance criteria. In *2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST)*, pages 321–331, 2020.
- [66] Annett Fischer. *Fakten rund um Software und Elektronik im Fahrzeug*. 2020. Robert Bosch GmbH.
- [67] Economic Commission for Europe. *Regelung Nr. 130 der Wirtschaftskommission der Vereinten Nationen für Europa (UNECE) - Einheitliche Bedingungen für die Genehmigung der Kraftfahrzeuge hinsichtlich ihres Spurhaltesystems*. 2014.
- [68] Economic Commission for Europe. *Regelung Nr. 13-H der Wirtschaftskommission für Europa der Vereinten Nationen (UN/ECE) - Einheitliche Bedingungen für die Genehmigung von Personenkraftwagen hinsichtlich der Bremsen*. 2015.
- [69] Economic Commission for Europe. *Agreement Concerning the Adoption of Uniform Conditions of Approval and Reciprocal Recognition of Approval for Motor Vehicle Equipment and Parts*. 2017.
- [70] Economic Commission for Europe. *UN-Regelung Nr. 157 — Einheitliche Bedingungen für die Genehmigung von Fahrzeugen hinsichtlich des automatischen Spurhalteassistenzsystems (ALKS) [2021/389]*. 2021.
- [71] Bundesministerium für Digitales und Verkehr. *Neue Fahrzeugsicherheitssysteme. Mobilität*, 2022.
- [72] Bundesanstalt für Straßenwesen. *Über GIDAS*. 2022. <https://www.gidas.org/about.html> (Besucht am 08.03.2022).
- [73] Thomas Geiger. *So testet Mercedes die neue C-Klasse*. 2006.

- [74] Bagschik Gerrit, Menzel Till, Reschka Andreas, and Maurer Markus. Szenarien für Entwicklung, Absicherung und Test von automatisierten Fahrzeugen. *Workshop Fahrerassistenzsysteme und automatisiertes Fahren*, 2017.
- [75] Verband der Automobilindustrie e.V. Gesellschaft für Informatik e.V. *Eingebettete Systeme in der Automobilindustrie*. 2015.
- [76] Sebastian Geyer, Marcel Baltzer, Benjamin Franz, and Hermann Winner. Concept and development of a unified ontology for generating test and use-case catalogues for assisted and automated vehicle guidance. In *IET Intelligent Transport Systems*, 2012.
- [77] Clemens Gühmann. Model-based testing of automotive electronic control units. *3rd International Conference on Materials Testing: Test*, 2005.
- [78] Robert Bosch GmbH. *Automotive electrics and automotive electronics*. Springer, 2007.
- [79] BMW Group. *INVESTOR PRESENTATION*. 2021.
- [80] The Standish Group. *CHAOS MANIFESTO 2012*. 2012.
- [81] Song Guo. *Cyber-Physical Systems: Architecture, Security and Application*. Springer, 2019.
- [82] Thomas Gustafsson, Mats Skoglund, Avenir Kobetski, and Daniel Sundmark. Automotive system testing by independent guarded assertions. In *IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, 2015.
- [83] Jochen Haab. Intelligent Drive im neuen EQS. In *TAGUNG AUTOMATISIERTES FAHREN*, 2022.
- [84] Marcel Haiber, Thomas Göppel, Tamara Hoffman, Christian Lalitsch-Schneider, Marcus Nolte, and Julian Pott. Lerngetriebener Produktlebenszyklus für hochautomatisierte Fahrzeuge. In *tdse - Tag des System Engineering - GfSE*, 2021.

- [85] Horst Hanke. *Begriffsbestimmungen - Verkehrsplanung, Straßenentwurf und Straßenbetrieb*. 2012.
- [86] Philipp Hanke. *INRIX Traffic Scorecard: Innenstadtverkehr nimmt zu und erreicht Vorkrisenniveau*. 2021.
- [87] Geir Kjetil Hanssen, Tor Stålhane, and Thor Myklebust. *SafeScrum® – Agile Development of Safety-Critical Software*. Springer, 2018.
- [88] Lukas Hartjen, Robin Philipp, Fabian Schuldt, Falk Howar, and Bernhard Friedrich. Classification of driving maneuvers in urban traffic for parametrization of test scenarios. In *9. Tagung Automatisiertes Fahren*, 2019.
- [89] Nico Hartmann. *Automatisiertes Testen Eingebetteter Systeme in der Automobilindustrie*, chapter Test Automatisierung im Labor. 2008 Carl Hanser Verlag GmbH & Co. KG, 2008.
- [90] J. E. Hopcroft and J. D. Ullman. *Introduction to automata theory, languages, and computation*. Pearson, 1979.
- [91] Dirk Huelsebusch. *Fahrerassistenzsysteme zur energieeffizienten Laengsregelung - Analyse und Optimierung der Fahrsicherheit*. PhD thesis, Karlsruher Institut für Technologie (KIT), 2017.
- [92] Andrea Huetter. *Verkehr auf einen Blick*. 2013. Statistisches Bundesamt.
- [93] Alexander Hutter. *Automatisiertes Testen Eingebetteter Systeme in der Automobilindustrie*, chapter Einsatz von Simulationsmodellen beim Test elektronischer Steuergeräte. 2008 Carl Hanser Verlag GmbH & Co. KG, 2008.
- [94] IEC. *IEC 6108*. International Electrotechnical Commission (IEC), 2001.
- [95] SAE International. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. 2016.
- [96] ISO. *ISO 610.12 - Glossary of Software Engineering*. International Organization for Standardization (ISO), 2002.

- [97] ISO. *ISO 12207 - Systems and software engineering - Software life cycle processes*. International Organization for Standardization (ISO), 2008.
- [98] ISO. *ISO 829 - IEEE Standard for Software and System Test Documentation*. International Organization for Standardization (ISO), 2008.
- [99] ISO. *ISO 29119 - Software and systems engineering - software testing*. International Organization for Standardization (ISO), 2009.
- [100] ISO. *ISO 26262 - Road Vehicles - Functional Safety*. International Organization for Standardization (ISO), 2011.
- [101] ISO. *ISO 24765 - Systems and software engineering - Vocabulary*. International Organization for Standardization (ISO), 2015.
- [102] ISO. *ISO 21448 - Road vehicles - Safety of the intended functionality*. International Organization for Standardization (ISO), 2019.
- [103] International Software Testing Qualification Board (ISTQB). *Istqb glossary*, 2022.
- [104] Matthias Janson. *Der autonome Autofahrer verschwindet*. 2021.
- [105] Christian Jeß. *Diese Schilder bringt die neue StVO*. 2020. <https://www.autobild.de/artikel/stvo-novelle-2020-erklaerung-fuer-neue-verkehrszeichen-und-sinnbilder-16783135.html> (besucht am 12.11.2021).
- [106] Janek Jochheim. *Cloud computing - a key component of adas/ad validation*. In *dSpace World Conference*, 2019.
- [107] Andrej Karpathy. *Tesla autonomy day 2019*, 2019.
- [108] Sarah Keller. *Anzahl der beförderten Pakete durch die Deutsche Post in Deutschland von 2016 bis 2020*. 2021.
- [109] Siddartha Khastgir, Gunwant Dhadyalla, Stewart Birrell, Sean Redmond, Ross Addinall, and Paul Jennings. *Test scenario generation for driving simulators using constrained randomization technique*. 2017.

- [110] Björn Klamann, Moritz Lippert, Christian Amersbach, and Hermann Winner. Defining pass-/fail-criteria for particular tests of automated driving functions. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 169–174, 2019.
- [111] Moritz Klischat and Matthias Althoff. Generating critical test scenarios for automated vehicles with evolutionary algorithms. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 2352–2358, 2019.
- [112] Moritz Klischat, Edmond Irani Liu, Fabian Holtke, and Matthias Althoff. Scenario factory: Creating safety-critical traffic scenarios for automated vehicles. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7, 2020.
- [113] Prof. Dr. Ralf Kneuper. *Eine kleine Geschichte des V-Modells*. 2017.
- [114] Christopher Kober. Aus Fahrern wird Verkehr - Fahrermodelle und Umgebungsverkehr in der digitalen Erprobungsfahrt. In *Workshop Automotive Software Engineering - Virtuelle Absicherung*, 2018.
- [115] Christopher Kober. *Stochastische Verkehrsflusssimulation auf Basis von Fahrerhaltensmodellen zur Absicherung automatisierter Fahrfunktionen*. Springer-Verlag, 2019.
- [116] Martin Kords. *Anteil der Lkw an der Transportleistung im Güterverkehr in Deutschland in den Jahren von 2013 bis 2024*. 2021.
- [117] Kraftfahrt-Bundesamt. *Wegweiser zur EG-Fahrzeugtypgenehmigung nach der Richtlinie 2007/46/EG*. 2007.
- [118] Kraftfahrt-Bundesamt. *Merkblatt zu Genehmigungen für "Neue Techniken oder Konzepte"(MTK)*. 2011.
- [119] Kraftfahrt-Bundesamt. *Benannte Technische Dienste - Zertifizierungsstellen für QM-Systeme*. 2019.

- [120] Kraftfahrt-Bundesamt. *Glossar - Certificate of Conformity*. 2019. <https://www.kba.de/SharedDocs/Glossareintraege/DE/C/CoC.html?view=renderHelp&nn=64> (besucht am 30.09.2019).
- [121] Kraftfahrt-Bundesamt. *Glossar - Conformity of Production*. 2019. https://www.kba.de/SharedDocs/Glossareintraege/DE/C/CoP_P.html?view=renderHelp&nn=64 (besucht am 30.09.2019).
- [122] Kraftfahrt-Bundesamt. *Glossar - Technischer Dienst*. 2019. https://www.kba.de/SharedDocs/Glossareintraege/DE/T/Technischer_Dienst.html?view=renderHelp&nn=456892 (besucht am 30.09.2019).
- [123] Kraftfahrt-Bundesamt. *Glossar - Typgenehmigungen*. 2019. <https://www.kba.de/SharedDocs/Glossareintraege/DE/T/Typgenehmigungen.html?view=renderHelp&nn=456892> (besucht am 30.09.2019).
- [124] Florian Kraus, Nicolas Scheiner, Werner Ritter, and Klaus Dietmayer. Using machine learning to detect ghost images in automotive radar. In *IEEE Intelligent Transportation Systems Conference (ITSC)*, 2020.
- [125] Alexander Kriegisch. *Agiles Projektmanagement, Scrum-Rollen - Product Owner*.
- [126] Peter M. Kruse. Systematisches Testen im agilen Entwicklungsprozess. *ATZ - Automobiltechnische Zeitschrift*, 2020.
- [127] Jörg Kubitzki and Wolfgang Fastenmeier. *Sicher zu Fuß Mobilität und Sicherheit von Fußgängern*. 2019.
- [128] Dae-hyun Kum, Joonwoo Son, Seon-bong Lee, and Ivan Wilson. Automated testing for automotive embedded systems. In *2006 SICE-ICASE International Joint Conference*, pages 4414–4418, 2006.
- [129] Kristofer D. Kusano and Hampton Gabler. *Method for Estimating Time to Collision at Braking in Real-World, Lead Vehicle Stopped Rear-End Crashes for Use in Pre-Crash System Design*. 2011.

- [130] Jacob Langner, Hannes Grolig, Stefan Otten, Marc Holzäpfel, and Eric Sax. Logical scenario derivation by clustering dynamic-length-segments extracted from real-world-driving-data. In *International Conference on Vehicle Technology and Intelligent Transport Systems*, 2019.
- [131] Sam Levin and Julia Carrie Wong. Self-driving uber kills arizona woman in first fatal crash involving pedestrian. *The Guardian*, 2018.
- [132] Shaoying Liu and Shin Nakajima. Automatic test case and test oracle generation based on functional scenarios in formal specifications for conformance testing. *IEEE Transactions on Software Engineering*, 48(2):691–712, 2022.
- [133] et. al. M. Wood. *Safety First for Automated Driving*. 2019.
- [134] S.M. Sohel Mahmud, Luis Ferreira, Md. Shamsul Hoque, and Ahmad Tavassoli. Application of proximal surrogate indicators for safety evaluation: A review of recent developments and research needs. *IATSS Research*, 41(4):153–163, 2017.
- [135] R. Matthaei and M. Maurer. Autonomous driving – a top-down-approach. In G. Bretthauer, editor, *Automatisierungstechnik: Methoden und Anwendungen der Steuerungs-, Regelungs- und Informationstechnik*, volume 3, pages 155–167, 2015.
- [136] Abhishek Mishra. *iOS Code Testing - Test-Driven Development and Behavior-Driven Development with Swift*. Springer, 2017.
- [137] John P.T. Mo, Cees Bil, and Arvind Sinha. *4 - Systems design*. Woodhead Publishing, Oxford, 2015.
- [138] Hrushikesh Mohanty. *Trends in Software Testing*. Springer, 2017.
- [139] Statistische Ämter des Bundes und der Länder. *Unfallatlas | Kartenanwendung*. 2022. <https://unfallatlas.statistikportal.de/> (Besucht am 08.03.2022).
- [140] Galen E Mullins, Paul G Stankiewicz, and Satyandra K Gupta. Automated generation of diverse and challenging scenarios for test and evaluation of

- autonomous vehicles. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 1443–1450. IEEE, 2017.
- [141] n-tv Nachrichtenfernsehen GmbH. *Lastwagen der Zukunft sind rollende Hightech Riesen*. October 2016. IAA Nutzfahrzeuge 2016.
- [142] H.-H. Nagel, W. Enkelmann, and G. Struck. Fhg-co-driver: From map-guided automatic driving by machine vision to a cooperative driver support. In *Mathl. Comput. Modelling Vol. 22*, 1995.
- [143] Demin Nalic, Tomislav Mihalj, Maximilian Bäumler, Matthias Lehmann, Arno Eichberger, and Stefan Bernsteiner. Scenario based testing of automated driving systems: A literature survey. In *FISITA web Congress*, 2020.
- [144] EURO NCAP. *TEST PROTOCOL – AEB VRU systems*. 2020.
- [145] Jonas Nielsson. *Computational verification methods for automotive safety systems*. PhD thesis, Chalmers University of Technology, 2014.
- [146] Jonas Nilsson, Anders C. E. Ödblom, Jonas Fredriksson, and Adeel Zafar. *Handbook of Augmented Reality*, chapter Using Augmentation Techniques for Performance Evaluation in Automotive Safety. Springer, 2011.
- [147] Philipp Obergfell. *Entwurfsmethodik für hybride Software- und Systemarchitektur*. PhD thesis, Karlsruher Institut für Technologie (KIT), 2021.
- [148] Philipp Obergfell, Matthias Traub, Florian Oszwald, and Eric Sax. Agile automotive e/e-systems engineering. In *ELIV MarketPlace 2018*. 2018.
- [149] Department of Motor Vehicles. “Disengagement Reports”. 2020. <https://www.dmv.ca.gov/portal/vehicle-industry-services/autonomous-vehicles/disengagement-reports/> (zugegriffen:09.06.2020).
- [150] Jörg Ortlepp and Petra Butterwegge. *Unfalltypen-Katalog Leitfaden zur Bestimmung des Unfalltyps*. 2016.
- [151] Madhumita Paul. Safety assessment at unsignalized intersections using post-encroachment time’s threshold—a sustainable solution for developing

- countries. In Srinivas Pulugurtha, Indrajit Ghosh, and Sabyasachi Biswas, editors, *Advances in Transportation Engineering*, pages 117–131, Singapore, 2019. Springer Singapore.
- [152] PEGASUS-Projekt. *Functional Decomposition*. 2017.
- [153] PEGASUS-Projekt. *Testen von L3 Systemen – ein Überblick über die aktuelle PEGASUS Vorgehensweise*. 2017.
- [154] PEGASUS-Projekt. *PEGASUS METHOD*. 2019.
- [155] PEGASUS-Projekt. Safety argument. Abschlussveranstaltung, 2019.
- [156] PEGASUS-Projekt. Scenario description and knowledge-based scenario generation. Abschlussveranstaltung, 2019.
- [157] Raphael Pfeffer. *Szenariobasierte simulationsgestützte funktionale Absicherung hochautomatisierter Fahrfunktionen durch Nutzung von Realdaten*. PhD thesis, Karlsruher Institut für Technologie (KIT), 2020.
- [158] Thomas Ponn, Dirk Fratzke, Christian Gnadl, and Markus Lienkamp. Towards certification of autonomous driving: Systematic test case generation for a comprehensive but economically-feasible assessment of lane keeping assist algorithms. pages 333–342, 01 2019.
- [159] Oxford University Press. *Oxford Dictionary*. Oxford University Press, 2020.
- [160] Andreas Pütz, Adrian Zlocki, Julian Bock, and Lutz Eckstein. System validation of highly automated vehicles with a database of relevant traffic scenarios. In *12th ITS European Congress*. 2017.
- [161] Radstand. *Studie: Deutschlands Arbeitnehmer stehen im Stau*. July 2008. Radstand Deutschland.
- [162] Karl-Heinz Rau. *Agile objektorientierte Software-Entwicklung*. Springer, 2016.

- [163] Konrad Reif. *Batterien, Bordnetze und Vernetzung*, chapter Architektur elektronischer Systeme. Vieweg+Teubner, 2010.
- [164] Harald Richter. *Elektronik und Datenkommunikation im Automobil*. 2009.
- [165] Lennart Ries, Philipp Rigoll, Thilo Braun, Thomas Schulik, Johannes Dautbe, and Eric Sax. Trajectory-based clustering of real-world urban driving sequences with multiple traffic objects. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021.
- [166] Christian Roesener, Felix Fahrenkrog, Axel Uhlig, and Lutz Eckstein. A scenario-based assessment approach for automated driving by using time series classification of human-driving behaviour. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1360–1365, 2016.
- [167] Winston W. Royce. Managing the development of large software systems: concepts and techniques. *IEEE WESCON*, 1970.
- [168] Kathrin Sattler. *Methodik für den Systemtest in der integralen Fahrzeugsicherheit*. PhD thesis, Otto-von-Guericke-Universität Magdeburg, 2015.
- [169] Eric Sax. *Automatisiertes Testen eingebetteter Systeme in der Automobilindustrie*. Carl Hanser Verlag, 2008.
- [170] Eric Sax. *Tatort Test - „Test is not the Last“*. 2015.
- [171] Eric Sax, Ralf Reussner, Houssemeddine Guissouma, and Heiko Klare. *A Survey on the State and Future of Automotive Software Release and Configuration Management*. 2017.
- [172] Konrad Schindler. *Photogrammetrie*. ETH Zuerich, 2015.
- [173] Helmut Schittenhelm. Vvm safeguarding automation –how to ensure a safe operation of an automated drivingsystem by a methodological approach? In *Halbzeitevent VV-Methoden*, 2022. Halbzeitevent VV-Methoden.

- [174] Stefan Schmerler. *Softwaretest in der Praxis: Grundlagen, Methoden und Technologien*. ebuli GmBH, 2020.
- [175] Barbara Schütt, Markus Steimle, Birte Kamer, Danny Behnecke, and Eric Sax. *A Taxonomy for Quality in Simulation-based Development and Testing of Automated Driving Systems*.
- [176] Jörg Schäuffele and Thomas Zurawka. *Automotive Software Engineering*. Springer Vieweg, 2016.
- [177] Ken Schwaber and Jeff Sutherland. *The Scrum Guide*™. 2017.
- [178] Seyed Reza Shahamiri, Wan Mohd Nasir Wan Kadir, and Siti Zaiton Mohd-Hashim. A comparative study on automated software test oracle methods. In *2009 Fourth International Conference on Software Engineering Advances*, pages 140–145, 2009.
- [179] Bushra Sharif, Shoab A. Khan, and Muhammad Wasim Bhatti. Measuring the impact of changing requirements on software project cost: An empirical investigation. 2012.
- [180] Ki-Wook Shin and Dong-Jin Lim. Model-based automatic test case generation for automotive embedded software testing. *International Journal of Automotive Technology*, 19:107–119, 02 2018.
- [181] Hesham Shokry and Mike Hinchey. Model-based verification of embedded software. In *Computer*, 2009.
- [182] Shai Shalev Shwartz, Shaked Shammah, and Amnon Shashua. On a formal model of safe and scalable self-driving cars. *CoRR*, abs/1708.06374, 2017.
- [183] Françoise Simonot-Lion and Yvon Trinquet. *Automotive Embedded Systems Handbook*. Tylor & Francis Group, 2009.
- [184] Daniel Åsljung, Jonas Nilsson, and Jonas Fredriksson. Comparing collision threat measures for verification of autonomous vehicles using extreme value theory. *IFAC-PapersOnLine*, 49(15):57–62, 2016. 9th IFAC Symposium on Intelligent Autonomous Vehicles IAV 2016.

- [185] Scott M. Smith. *Determining Sample Size*. 2013.
- [186] Andreas Spillner and Tilo Linz. *Basiswissen Softwaretest*. dpunkt, 2005.
- [187] Evgeny Spodarev. Methoden zur Konstruktion von Schätzern. In *Stochastik I*. Universität Ulm, 2013.
- [188] Mirosław Staron. *Automotive Software Architectures An Introduction*. Springer, 2017.
- [189] Jann-Eve Stavesand. Approval and homologation of autonomous vehicles - with dspace support. In *dSpace World Conference*, 2019.
- [190] Markus Steimle, Gerrit Bagschik, Till Menzel, and Markus Maurer. Ein Beitrag zur Terminologie für den szenarienbasierten Testansatz automatisierter Fahrfunktionen. In *AAET - Automatisiertes und vernetztes Fahren*, 2018.
- [191] Markus Steimle, Till Menzel, and Markus Maurer. *A Method for Classifying Test Bench Configurations in a Scenario-Based Test Approach for Automated Vehicles*. 2019. Preprint.
- [192] Jan Stellet, Matthias Woehrle, Tino Brade, Alexander Poddey, and Wolfgang Branz. Validation of automated driving – a structured analysis and survey of approaches. In *FAS Workshop 2020*, 2020.
- [193] Jan Erik Stellet, Marc René Zofka, Jan Schumacher, Thomas Schamm, Frank Niewels, and J. Marius Zöllner. 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*, 2015.
- [194] Hannes Stoll. *Die (re-)konfigurierbare Fahrzeugarchitektur*. PhD thesis, Karlsruher Instituts für Technologie (KIT), 2021.
- [195] Johann Stoll, Anja Schneider, Marcus Wisch, Patrick Seiniger, and Thomas Schaller. *The addressed VRU scenarios within PROSPECT and associated test catalogue*. 2016.

- [196] Tesla. *Autopilot and Full Self-Driving Capability*. 2021.
- [197] Verena Töpper. In Deutschland gibt es immer mehr Pendler. *Der Spiegel*, 2020.
- [198] Ed Triou, Zafar Abbas, and Sravani Kothapalle. Declarative testing: A paradigm for testing software applications. In *2009 Sixth International Conference on Information Technology: New Generations*, 2009.
- [199] Cumhur Erkan Tuncali, Georgios Fainekos, Danil Prokhorov, Hisahiro Ito, and James Kapinski. Requirements-driven test generation for autonomous vehicles with machine learning components. In *IEEE Transactions on Intelligent Vehicles*, 2020.
- [200] TÜV-Nord. *Homologation: The Test that builds Trust*. 2021.
- [201] Simon Ulbrich, Till Menzel, Andreas Reschka, Fabian Schuldt, and Markus Maurer. Defining and substantiating the terms scene, situation, and scenario for automated driving. *IEEE 18th International Conference on Intelligent Transportation Systems*, 2015.
- [202] Umweltbundesamt. *Fahrleistungen, Verkehrsleistungen und „Modal Split“*. 2021.
- [203] Europäische Union. *RICHTLINIE 2007/46/EG DES EUROPÄISCHEN PARLAMENTS UND DES RATES zur Schaffung eines Rahmens für die Genehmigung von Kraftfahrzeugen und Kraftfahrzeuganhängern sowie von Systemen, Bauteilen und selbstständigen technischen Einheiten für diese Fahrzeuge*. 2007.
- [204] Richard van der Horst, Maartje de Goede, Stefanie de Hair-Buijssen, and Rob Methorst. Traffic conflicts on bicycle paths: A systematic observation of behaviour from video. *Accident Analysis & Prevention*, 62:358 – 368, 2014.
- [205] VDA. Automatisierung - Von Fahrerassistenzsystemen zum automatisierten Fahren. *VDA Magazin*, 2015.

- [206] Walther Wachenfeld and Hermann Winner. *The Release of Autonomous Vehicles*, pages 425–449. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.
- [207] Johannes Wagner and Jürgen Häring. Validierung von hochautomatisierten Fahrfunktionen mit cloud-basierter Simulation. *ATZ - Automobiltechnische Zeitschrift*, 2020.
- [208] Cheng Wang and Hermann Winner. Overcoming challenges of validation automated driving and identification of critical scenarios. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019.
- [209] Julian Weber. *Automotive Development Processes*. Springer, 2009.
- [210] Nico Weber, Christoph Thiem, and Ulrich Konigorski. A needle in a haystack – how to derive relevant scenarios for testing automated driving systems in urban areas. In *30. Aachen Colloquium Sustainable Mobility 2021*, 2021.
- [211] Tim Weilkens, Christian Weiss, Andrea Grass, and Kim Nena Duggen. Chapter 7 - frameworks. In Tim Weilkens, Christian Weiss, Andrea Grass, and Kim Nena Duggen, editors, *OCEB 2 Certification Guide (Second Edition)*, pages 149 – 169. Morgan Kaufmann, second edition edition, 2016.
- [212] Frank Eugen Wißler. *Ein Verfahren zur Bewertung technischer Risiken in der Phase der Entwicklung komplexer Serienprodukte*. PhD thesis, Universität Stuttgart, 2006.
- [213] Christoph Wohlfahrt. Von Systematischer Absicherung zur „Digitalen Erprobungsfahrt“. In *Workshop Automotive Software Engineering - Virtuelle Absicherung*, 2016.
- [214] Danny Yadron and Dan Tynan. Tesla driver dies in first fatal crash while using autopilot mode. *The Guardian*, 2016.

- [215] Ding Zhao, Yaohui Guo, and Yunhan Jack Jia. Trafficnet: An open naturalistic driving scenario library. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017.
- [216] Marc Rene Zofka, Ralf Kohlhaas, Tjomas Schamm, and J. Marius Zöllner. Semivirtual simulations for the evaluation of vision-based adas. In *IEEE Intelligent Vehicles Symposium*, 2014.

Betreute Bachelor- und Masterarbeiten

- [1] Constantin Braess. *Extraktion von manöverbasierten logischen Szenarien im urbanen Verkehr*. 2021.
- [2] Benedikt Dietrich. *Automatisierte Durchführung und Evaluation von Systemtest für Softwarekomponenten mit fahrzeugexternen Schnittstellen*. 2021.
- [3] Daniel Hanselmann. *Methode zur effizienten Absicherung logischer Szenarien durch die Zerlegung von Parameterbereichen*. 2022.
- [4] Lukas Hauber. *Identifikation von Fahrmanövern innerhalb einer virtuellen Testfahrt mittels künstlicher neuronaler Netze*. 2018.
- [5] Jonas Hofmann. *Automated Roadmap Generation and Optimization for Automatic Guided Vehicles*. 2017.
- [6] Matthias Petry. *Übertragbarkeit von IT Software-Testing Methoden auf die automobile Entwicklung*. 2020.
- [7] Anke Pfeiffer. *Definition und Entwicklung von Metriken für eine automatisierte Qualitätsbewertung von Fahrerassistenzsystemen*. 2019.
- [8] Niklas Phieler. *Umgebungssimulation für einen Licht-/Regensensor im HiL-Umfeld*. 2019.
- [9] Niklas Töppel. *Aufbau einer Ground-Truth für die Bewertung von HAF an einem Fahrerassistenzdemonstrator*. 2019.

- [10] Lingguang Wang. *Konzeption und Entwicklung einer personalisierten Gestenerkennung mittels Maschinellern Lernen*. 2018.
- [11] Meng Zhang. *Entwicklung und Test eines Perzeptionssystems für eine automatisierte Fahrfunktion*. 2022.

Eigene Veröffentlichungen

- [King et al., 2017] King, C., Bach, J., Otten, S., and Sax, E. (2017). Identifikation von Fahrscenarien während einer virtuellen Testfahrt. In Eibl, Maximilian; Gadedke, M., editor, *INFORMATIK 2017*, pages 1511–1516.
- [King et al., 2021] King, C., Braun, T., Braess, C., Langner, J., and Sax, E. (2021). Capturing the variety of urban logical scenarios from bird-view trajectories. In *7th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS)*.
- [King et al., 2019] King, C., Ries, L., Kober, C., Wohlfahrt, C., and Sax, E. (2019). Automated function assessment in driving scenarios. In *2019 12th IEEE Conference on Software Testing, Validation and Verification (ICST)*.
- [King et al., 2020] King, C., Ries, L., Langner, J., and Sax, E. (2020). A taxonomy and survey on validation approaches for automated driving systems. In *2020 IEEE International Symposium on Systems Engineering (ISSE)*, pages 1–8.