

Methoden zur Ermittlung der Aussagekraft eines Logischen Szenarios im Absicherungs- und Freigabeprozess einer hochautomatisierten Fahrfunktion

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)

angenommene

DISSERTATION

von

M. Eng. Julian Fuchs

geb. in Ellwangen

Tag der mündlichen Prüfung:

20.03.2026

Hauptreferent:

Prof. Dr.-Ing. Eric Sax

Korreferent:

Prof. Dr.-Ing. Markus Glaser

Inhaltsverzeichnis

Abkürzungen und Symbole	v
Vorwort und Danksagung	vii
1 Einleitung	1
1.1 Einleitung	1
1.2 Motivation	4
1.3 Umfeld	8
1.4 Aufbau	9
2 Automotive Entwicklungs- und Testprozess	11
2.1 Strukturierung und Einordnung der Begriffe	11
2.2 Produktentstehungsprozess in der Automobilindustrie	12
2.2.1 Fahrzeugdomänen	12
2.2.2 Automatisierungsstufen	13
2.2.3 Entwicklungsprozesse	15
2.2.4 Reifegrade in der automotive Produktentwicklung	19
2.2.5 Funktionale Sicherheit und ASIL-Einstufung im Entwicklungsprozess	21
2.3 Testinfrastruktur	23
2.3.1 Testplattform	24
2.4 Szenarien, KPI und Testfälle im automotive Kontext	28
2.4.1 Szenarien	28
2.4.2 KPI als Bewertungsmetrik eines Testfalls	38
2.4.3 Anwendungsfälle von Szenarien im Automotive Kontext	39

3 Ähnlichkeitsmaße in der Wissenschaft und Technik: Die Bewertung der Aussagekraft von Szenarien	45
3.1 Repräsentativität in der Szenarienanalyse	45
3.2 Deterministische Szenarienanalyse	46
3.2.1 Einfache mathematische Distanzmaße	46
3.2.2 Hausdorff-Distanzmaß	47
3.2.3 Dynamic Time Warping	49
3.2.4 Kritikalitätsanalyse	51
3.2.5 Dekomposition	52
3.2.6 Adaptive Ausführung zur Ermittlung von Szenarien	53
3.3 Szenarienanalyse mit Hilfe von maschinellem Lernen	54
3.3.1 Clustering	56
3.3.2 Prädiktions- und Vorhersagemodelle	59
3.3.3 Principal Component Analysis (PCA)	63
4 Anforderungen, Rahmenbedingungen und Schnittstellen für einen Selektionsprozess für Konkrete Szenarien	65
4.1 Anforderungen und Rahmenbedingungen	65
4.1.1 Einordnung in den Gesamtprozess	65
4.1.2 Wiederverwendbarkeit und Übertragbarkeit	67
4.1.3 Ähnlichkeitssaussage	68
4.1.4 Reduktion des Testaufwands	68
4.1.5 Automatisiert	69
4.1.6 Skalierung	70
4.1.7 Modularität	70
4.1.8 Dokumentation und Nachvollziehbarkeit	70
4.1.9 Erweiterbarkeit	71
4.2 Schnittstellen des Selektionsprozesses zum Szenarienkatalog	72
4.2.1 Eigenschaften und Bedingungen an ein Logisches Szenario	72
4.2.2 Bestimmung von relevanten Key-Performance-Indikatoren (KPI) für ein Logisches Szenario	73

5	Methodik zur Selektion von repräsentativen Konkreten Szenarien in einem Logischen Szenario	77
5.1	Basis des Selektionsprozesses	77
5.2	Schritt I: Kombinatorische Exploration des Parameterraums	93
5.2.1	Methodenbaustein I-1: Äquidistante kombinatorische Erstellung	94
5.2.2	Methodenbaustein I-2: Kombinatorische Erstellung auf Basis einer Verteilungsfunktion	96
5.2.3	Methodenbaustein I-3: Stochastische Erstellung	97
5.3	Schritt II: Präexekutive Szenarienanalyse und -reduktion	100
5.3.1	Methodenbaustein II-1: Einschränkung des Parameterraums auf Basis von Restriktionen	101
5.3.2	Methodenbaustein II-2: Zusammenfassung von Parametern	104
5.4	Schritt III: Interexekutive (zyklische) Szenarienanalyse und -reduktion	109
5.4.1	Methodenbaustein III-1: Erste Sampling Strategien	111
5.4.2	Methodenbaustein III-2: Methode zur Einordnung der Konkreten Szenarien in Bezug auf Unfallschwere und Auftrittswahrscheinlichkeit	113
5.4.3	Methodenbaustein III-3: Unsicherheits- und Ähnlichkeitsanalyse	125
5.5	Abdeckungsaussage zum Logischen Szenario mit Abschätzung des Testaufwands	131
6	Design und Realisierung des Selektionsprozesses	137
7	Evaluation am Beispiel eines Logischen Szenarios	145
7.1	Einführung eines beispielhaften Logischen Szenarios	145
7.2	Anwendung der Methodenbausteine auf ein beispielhaftes Logisches Szenario	147
7.2.1	Methodenbaustein I-1: Äquidistante kombinatorische Erstellung	150
7.2.2	Methodenbaustein II-1: Einschränkung des Parameterraums auf Basis von Restriktionen	154
7.2.3	Methodenbaustein II-2: Zusammenfassung von Parametern	161

7.2.4	Methodenbaustein III-1: Erste Sampling Strategien	167
7.2.5	Methodenbaustein III-3: Unsicherheits- und Ähnlichkeitsanalyse	170
7.3	Zusammenfassung und Diskussion der Evaluation des beispielhaften Szenarios	180
7.3.1	Beitrag zur Absicherung und zur Reduktion des Testaufwands	180
7.3.2	Vergleich der Methodenbausteine mit den Anforderungen	181
7.3.3	Praktikabilität und Rolle des Expertenwissens im Selektionsprozess	183
8	Zusammenfassung und Ausblick	187
8.1	Zusammenfassende Ergebnisse und Eigenleistungen	187
8.2	Ausblick auf weitere Forschungsfelder und Schwerpunkte	189
	Abbildungsverzeichnis	193
	Tabellenverzeichnis	199
	Eigene Veröffentlichungen	201
	Abschlussarbeiten	203
	Literaturverzeichnis	205
A	Anhang	223

Abkürzungen und Symbole

Abkürzungen

ADAS	Advanced Driver Assistance System
bspw.	beispielsweise
FMEA	Failure mode and effects analysis (deutsch. Fehlermöglichkeits- und -Einflussanalyse)
HiL	Hardware-in-the-loop
ISO	International Organization for Standardization
SiL	Software-in-the-loop
SOP	Start-Of-Production
SuT	System-under-Test (deutsch. Testsystem)
ViL	Vehicle-in-the-loop
ODD	Operational Design Domain
UNECE	United Nations Economic Commission for Europe
KPI	Key Performance Indicator
PET	Post-encroachment time
TTC	Time-To-Collision (deutsch. Kollisionszeitabstand)
StVO	Straßen-Verkehrs-Ordnung

NKW Nutzkraftwagen

vgl. vergleiche

z.B. zum Beispiel

Vorwort und Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit am Karlsruher Institut für Technologie (KIT) in Zusammenarbeit mit dem FZI Forschungszentrum Informatik sowie dem Institut für Technik der Informationsverarbeitung (ITIV).

Mein besonderer Dank gilt meinem Doktorvater, Herrn Prof. Dr.-Ing. Eric Sax, für die hervorragende wissenschaftliche Betreuung dieser Arbeit. Seine wertvollen Impulse, die zahlreichen fachlichen Diskussionen sowie die stets konstruktive Unterstützung haben maßgeblich zum Gelingen dieser Dissertation beigetragen.

Ebenso möchte ich mich bei meinen Kolleginnen und Kollegen am FZI sowie am ITIV bedanken. Der fachliche Austausch, die inspirierenden Gespräche und die gemeinsame Zeit in Karlsruhe haben nicht nur diese Arbeit bereichert, sondern auch meine persönliche und wissenschaftliche Entwicklung nachhaltig geprägt.

Mein tiefster Dank gilt meiner Frau Louisa. Ihre Geduld, ihr Verständnis und ihre unermüdliche Unterstützung haben mir in allen Phasen dieser Arbeit den nötigen Rückhalt gegeben. Sie war mir stets eine große Stütze – besonders in herausfordernden Zeiten.

Darüber hinaus danke ich meiner Familie und meinen Freunden von Herzen. Ihre Unterstützung, ihr Zuspruch und die vielen wertvollen Momente der Ablenkung haben entscheidend dazu beigetragen, diese Zeit erfolgreich zu meistern.

1 Einleitung

1.1 Einleitung

Fahrerassistenzsysteme sind bereits ein fester Bestandteil der mittleren und oberen Fahrzeugklassen. Seit 2022 müssen neue Fahrzeuge gesetzlich mit einem Spurhalteassistenten und Notbremsassistenten ausgestattet werden [1]. Automatisierte Fahrzeuge mit höheren Automatisierungsgraden stellen seit einigen Jahren einen Schwerpunkt der Forschung und Entwicklung dar [8]. Bisherige Funktionen konnten streng nach geschlossenen und vordefinierten Anforderungen entwickelt werden [68]. Am 17. Mai 2022 erhielt die Mercedes S-Klasse als erstes Fahrzeug eine offizielle Zulassung in Deutschland für einen automatisierten Stau-Piloten, welcher in einem begrenzten Autobahnumfeld die Verantwortung über die Längs- und Querführung an das System übergibt [116]. Der Fahrer hat die Option sich in diesen Situationen anderen Aufgaben als der Fahraufgabe zu widmen. Hochautomatisierte Fahrfunktionen müssen ohne den Fahrer als Rückfallebene und somit in jeder Fahrsituation (Open-World-Kontext) unter den verschiedensten Gegebenheiten und Umwelteinflüssen, wie zum Beispiel bei Schnee, Regen oder der Fahrt auf einer Landstraße, funktionieren. Nicht nur die Entwicklung und Abstimmung der Fahrfunktion stellt die Automobilindustrie vor neue Herausforderungen. Sowohl der Freigabe- als auch der Entwicklungsprozess müssen sich mit neuen Frage- und Problemstellungen beschäftigen. Unter anderem müssen Fragen, wie „Wann wurde genug getestet?“ oder „Wie wird garantiert, dass die relevanten Situationen vom zu testenden System korrekt bewältigt werden?“ beantwortet werden. Ein Lösungsansatz stellt die Methode der szenarienbasierten Absicherung dar. Hierbei wird ein Szenarienkatalog aufgebaut, welcher aus verschiedenen Fahrsituationen und Fahrmanövern besteht. Dieser Trendwandel, von der anforderungsbasierten

Absicherung hin zur szenarienbasierten Absicherung, beschäftigt Fahrzeughersteller und -zulieferer unter anderem in öffentlichen Forschungsprojekten, wie beispielsweise VV-Methoden¹, SetLevel oder Pegasus. Bei der szenarienbasierten Absicherung wird entwicklungsbegleitend ein unabhängiger Szenarienkatalog erstellt. Dieser Szenarienkatalog erlaubt, den kilometerbasierten Absicherungsansatz, bei dem bei jeder neuen Fahrzeugvariante oder Softwarestand erneut 2,1 Milliarden Testkilometer durch Real-Erprobungsfahrten gesammelt werden müssen, zu umgehen [68].

Durch den Einsatz von Simulationen wird der Testaufwand reduziert. Nicht nur der Entwicklungsprozess entlang des V-Modells muss sich an diese neuen Gegebenheiten anpassen. Auch die Testmethoden und Freigabe- und Absicherungsprozesse müssen sich darauf einstellen, dass der Fahrer im Fahrzeug nur noch eine Passagierrolle spielt und sich immer mehr aus dem aktiven Verkehr zurückzieht. Bisher erfolgt die Freigabe von Fahrzeugfunktionen auf Basis von UNECE² Regelungen, welche zusammen mit dem Hersteller und der entsprechenden Zulassungsbehörde vereinbart werden [118]. Aktuelle Homologationsansätze³ für teilautomatisierte Systeme, wie dem Abstandsregeltempomaten, schreiben hier *Konkrete Szenarien* vor, in welcher sich das teilautomatisierte System korrekt verhalten muss [118]. Für Systeme mit einem höheren Automatisierungsgrad, welche im Open-World-Context, unabhängig von den Umgebungsbedingungen funktionieren müssen, lässt sich dieser Ansatz nicht direkt übertragen [118]. Laut Maurer [68] ist die Verwendung von Szenarien in der Simulation zur Absicherung

¹ Das Projekt **VV-M** (Verifikation und Validierung von Methoden zur Absicherung automatisierter Fahrzeuge) ist eine großangelegte Forschungsinitiative in Deutschland, die im Kontext des hochautomatisierten Fahrens und der Fahrzeugsicherheit angesiedelt ist. Das Projekt wird vom Bundesministerium für Wirtschaft und Klimaschutz (BMWK) gefördert und vereint namhafte Industriepartner, wie Mercedes-Benz, sowie Forschungsinstitute und Universitäten

² **UNECE** Abkürzung für United Nations Economic Commission for Europe; Wirtschaftskommission der UN für Europa, im März 1947 vom ECOSOC (UN) als erste regionale Wirtschaftskommission gegründet. Nach [55]

³ **Homologation** beschreibt den Prozess, durch den ein Fahrzeugtyp von einer zuständigen Behörde genehmigt wird, um auf öffentlichen Straßen verwendet werden zu dürfen. Dieser Prozess umfasst in der Regel Tests, um sicherzustellen, dass das Fahrzeug bestimmten Sicherheits- und Umweltstandards entspricht. Es gibt unterschiedliche Homologationsansätze in verschiedenen Ländern und Regionen, die je nach ihren eigenen Gesetzen und Vorschriften variieren. Nach [59]

und Freigabe von hochautomatisierten Fahrfunktionen im Kontext von zukünftigen Trends in der Automobilindustrie eine Lösung des Problems. Beim Erstellen und Zusammenführen der Szenarien zu einem Szenarienkatalog stellt sich schnell die Frage nach der Auswahl und dem Umfang der passenden Szenarien. Welche Szenarien müssen getestet werden, um mit einer entsprechenden Gültigkeitsaussage bzw. Repräsentativität festzustellen, dass sich das zu testende Fahrzeugsystem korrekt verhält? Dabei ist die Sammlung und Identifikation von verschiedenen Szenarien in einem Szenarienkatalog ein erster Schritt. Die Identifikation von Einflussparametern auf das jeweilige Szenario führt zu mehreren Kombinationsmöglichkeiten von verschiedenen Parametern innerhalb einer einzigen Situation. Diese Kombinationsmöglichkeiten sind ohne eine strukturierte Vorgehensweise nicht bewältig- bzw. testbar.

Bei Szenarien auf einer logischen Ebene - sogenannten *Logischen Szenarien* - mit vordefinierten Parametern ergeben sich unterschiedliche Arten und Anzahl von Einflussfaktoren, die das Szenario charakterisieren oder bestimmen. Fahrzeugbezogene Größen, wie Geschwindigkeiten, Abstände zu anderen Verkehrsteilnehmern bzw. verschiedene Arten von Verkehrsteilnehmern, aber auch Wettereffekte, wie Regen, Schnee und Glatteis sind Parameter unter welchen das System im Open-World-Kontext getestet werden muss. Diese Vielzahl an Parameterkombinationen führt zu verschiedenen Systemreaktionen und sind damit unterschiedlich relevant. Gleichzeitig lassen sich aufgrund der Kombinatorik von verschiedenen Parametern mit einem kontinuierlichen Wertebereich nicht alle Parameterkombinationen und damit nicht alle Testfälle testen. Stattdessen muss ein Ansatz entwickelt werden, wie mit Hilfe von verschiedenen Methoden eine Generalisierungsaussage über die Gültigkeit eines *Konkreten Szenarios* getroffen werden kann. Ziel ist es dabei der Testaussage eines *Logischen Szenarios* mit einer Konfidenz zu vertrauen und somit eine Entscheidung zu treffen, ob weitere Konkrete Szenariendurchläufe erforderlich sind oder ob die Repräsentativität der ähnlichen Szenariendurchläufe ausreicht, um den Parameterbereich komplett abzudecken. Es wird mit Hilfe einer Auswahl an repräsentativen Szenarien im Parameterraum eine Aussage über die ausgelassenen, nicht ausgeführten *Konkreten Szenarien* getroffen. Im Rahmen der Dissertation liegt der Fokus auf einem

Prozess zur Ableitung eines validen und repräsentativen Szenarienkatalogs zur Ermittlung von Testfällen, welche im Kontext der Absicherung und Freigabe, aber auch der Ausführbarkeit, am Ende des Entwicklungs- und Testprozesses für eine Sicherheitsargumentation verwendet wird. Hierbei ist das Ziel für ein *Logisches Szenario* eine testbare Anzahl an repräsentativen *Konkreten Szenarien* zu finden, welche eine Abdeckungsaussage über das übergeordnete *Logische Szenario* zulassen.

1.2 Motivation

Für automatisierte Systeme gibt es verschiedenste Anwendungsbereiche und Profiteure. Neben der Reduktion von Schadstoffemissionen durch Carsharingkonzepte, spielt auch der Einsatz im Nutzfahrzeug- und Berufspendler-Bereich eine finanzielle Rolle. In Forschungsprojekten, wie Minga, mit der Stadt München werden Bus-Platoone automatisiert, um dem Busfahrer-mangel entgegenzusteueren. Unter anderem deswegen ist das automatisierte Fahren neben der Elektromobilität das Hauptthema, welches zurzeit alle Automobilkonzerne beschäftigt. Neue technische Optionen der Datenverarbeitung und Sensortechnologien gelten als wichtiger Faktor für das Möglichwerden des automatisierten Fahrens. Neben gesetzlichen und ethischen Rahmenbedingungen gilt die technische Umsetzbarkeit als eine der drei Herausforderungen beim automatisierten Fahren. Betrachtet man dabei den automobilen Entwicklungszyklus ergeben sich bereits bei der Spezifikation eines Systems im Open-World-Kontext Herausforderungen. Bisherige automobiler Systeme arbeiten unter abgesteckten Rahmenbedingungen. Schlechtwetter-Situationen erfordern bisher beispielsweise das Eingreifen des Fahrers als Rückfallebene. Durch die bisherige anforderungsbasierte Vorgehensweise und den abgesteckten Funktionsraum wird das Zielsystem in den weiteren Schritten entsprechend nach Anforderung entwickelt. Diese Anforderungen müssen abschließend validiert und verifiziert werden. Bisherige UNECE Freigaberichtlinien sehen für solche Systeme genau vordefinierte Freigabetests vor, welche entsprechend eng auf die zu testenden Systeme abgestimmt sind.

Bei höheren Automatisierungsgraden im Open-World Kontext entfällt die Option Rahmenbedingungen und Anforderungen aufzustellen. Der Fahrer entfällt als Rückfallebene. Das System muss nun in der Lage sein, sich unabhängig von Umwelteinflüssen korrekt zu verhalten. Es ergeben sich Anforderungen, wie „Das System darf keinen Unfall verursachen“ oder „Das System darf kein Risiko für andere Verkehrsteilnehmer darstellen“. Dies erfüllt nicht die Kriterien für messbare und testbare Anforderungen und sind somit nicht direkt in Testfälle überführbar. Ansätze der Absicherung auf Basis von Anforderungen funktionieren daher im Open-World-Kontext unter sich ändernden Umgebungsbedingungen (vgl. Abbildung 1.1) nicht mehr [62].

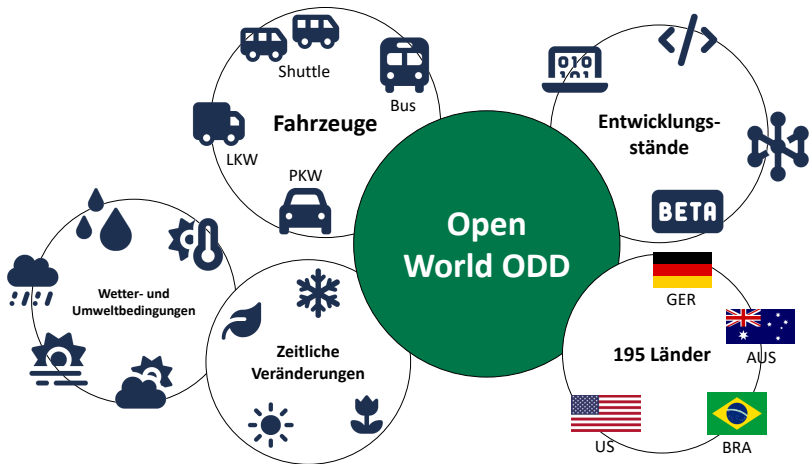


Abbildung 1.1: Herausforderungen in der Entwicklung und Freigabe einer hochautomatisierten Fahrfunktion nach [85]

Als eine Lösung erhält die szenarienbasierte Absicherung, bei welcher ein unabhängiger Katalog von verschiedenen Fahrsituationen, in verschiedenen Umwelt- und Randeinflüssen (Szenarienkatalog) die Testaussagen verifiziert, einen immer größer werdenden Stellenwert. Ziel ist es dabei, einen repräsentativen Satz an Szenarien zu identifizieren, welche dann durch eine konkrete Parametrisierung,

wie die Vorgabe eines Geschwindigkeitsverlaufs, in konkrete testbare Situationen überführt werden. Diese Szenarien werden parallel an simulationsbasierten Testinstanzen virtuell simuliert, getestet und bewertet. Einige konkrete Situationen lassen sich auch im Realfahrzeug im Rahmen von Realerprobungen auf Prüfgeländen testen. Zur Straßenzulassung muss das System Freigabetests bestehen. Diese Freigabetests werden am Ende des Entwicklungsprozesses auf ein System mit einem fortgeschrittenen Reifegrad ausgeführt. Das System muss in unterschiedlichen Situationen getestet werden, um am Ende freigegeben zu werden. Die Anzahl und der Umfang dieser Situationen bzw. Tests sind für hochautomatisierte Fahrfunktionen bisher unbekannt. Am Ende ist es dabei wichtig zum einen den Testaufwand an die verfügbaren Testinstanzen anzupassen und damit Testaufwand zu reduzieren, um nicht jede Kombination testen zu müssen, und zum anderen eine Aussage zur Konfidenz eines Szenarios abzugeben. Dabei wird davon ausgegangen, dass Szenarien gefunden werden, welche repräsentativ für andere Szenarien in ihrer Umgebung stehen (Generalisierungsannahme). Es wird aufgrund des kontinuierlichen Parameterraums immer Parameterkombinationen innerhalb von *Konkreten Szenarien* geben, welche nicht getestet werden konnten. Ziel ist es, eine Aussage zu treffen, dass die bereits ausgewählten und getesteten Kombinationen auch eine Aussage über die ausgelassenen Szenarien treffen.

Hierzu wird im Rahmen dieser Dissertation ein Selektionsprozess entwickelt, welcher es erlaubt, eine notwendige Anzahl an Testfällen auszuwählen.

Die triviale Lösung, bei der versucht wird, jede Parameterkombination durch beispielsweise eine Simulation abzutesten, kommt dabei aufgrund der Anzahl an möglichen Parameterkombinationen an ihre Grenzen. Geht man hierbei beispielhaft von einer konstanten Abtastung des jeweiligen Parameterbereichs mit 20 Stützstellen pro Parameter aus, ergeben sich durch die Kombinatorik bereits bei vier Parametern über 90.000 Parameterkombinationen und damit auch über 90.000 auszuführenden *Konkreten Szenarien*. Nimmt man sich also beispielhaft ein Szenario heraus, in welchem die Parameter, Geschwindigkeit des Egos, die Geschwindigkeit eines Fußgängers, der Startpunkt des Fußgängers und die Regenintensität variiert werden und man das System pro Parameter in 20 Diskretisierungsstufen unterteilt, ergeben sich hierbei für diese einzelne Situation bereits

über 90.000 Situationen, in welchen das System getestet werden muss. Geht man nun davon aus, dass jeder Simulationsdurchlauf 1 min dauert, werden für das Testen des Systems in einer Variante und einem Reifegrad 1.500h Testzeit benötigt. Reale Verkehrssituationen sind noch umfangreicher und besitzen mehr als vier Einflussparameter. Je größer die Anzahl der Einflussparameter ist, desto mehr Parameterkombinationen ergeben sich (vgl. Abbildung 1.2). Zur Absicherung muss

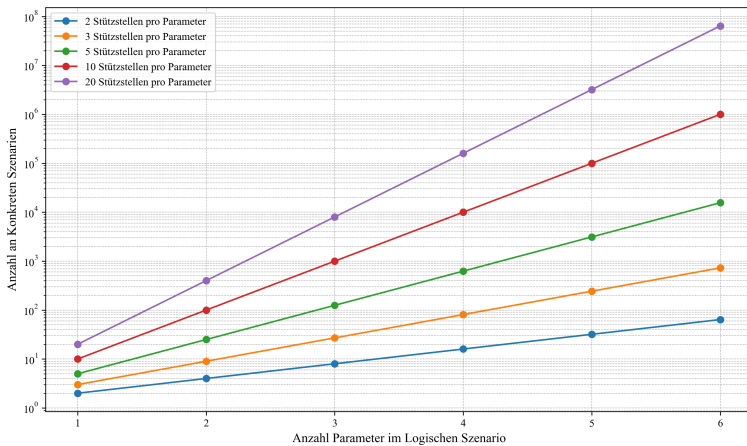


Abbildung 1.2: Zunahme der *Konkreten Szenarien* in Abhängigkeit der Anzahl der Parameter

allerdings nicht nur ein übergeordnetes Szenario getestet werden. Selbst durch eine Verlagerung von realen Tests auf dem Prüfgelände in eine parallelisierbare Simulation lässt sich die Anzahl der Testfälle und Kombinationen nicht in endlicher Zeit mit beschränkten Testressourcen ausführen (vgl. Abbildung 1.3).

Die Anzahl an auszuführenden Testfällen lässt sich nicht pauschal treffen, sondern ist vom System und vom gewählten zugehörigen übergeordneten *Logischen Szenarien* abhängig.

Deshalb muss versucht werden, durch Vorüberlegungen und eine inkrementell gesteuerte Testdurchführung den Parameterraum einzugrenzen, um so nur noch

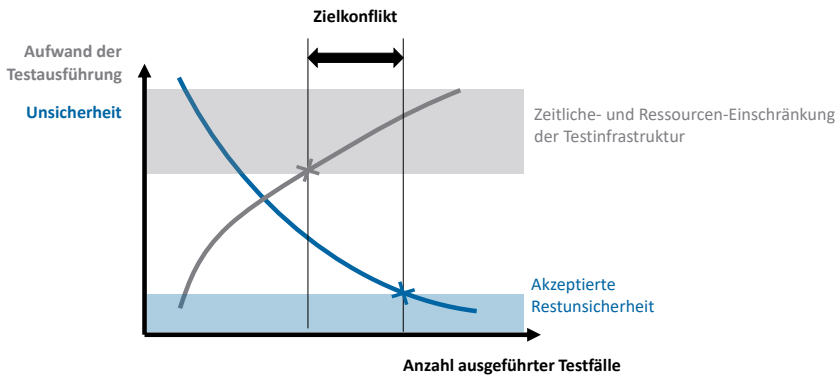


Abbildung 1.3: Beispielhafte qualitative Darstellung des Zielkonflikts zwischen Testaufwand und verbleibender Restunsicherheit

wenige repräsentative Szenarien ausführen zu müssen. Die vorliegende Dissertation adressiert genau diese Problemstellung und versucht die Anzahl der Szenarien mit Hilfe von verschiedenen Methodenbausteinen zu „reduzieren“ und die Ergebnisse dann zu einer Gesamtrepräsentativität des übergeordneten *Logischen Szenarios* zu aggregieren. Diese wird abschließend zur Absicherung der hochautomatisierten Gesamtfahrfunction verwendet.

1.3 Umfeld

Im Forschungsumfeld der szenarienbasierten Absicherung beschäftigen sich viele wissenschaftliche Ausarbeitungen. Diese Dissertation beschäftigt sich mit der Szenarioableitung im Absicherungskontext.

Hierzu wird im Rahmen der vorliegenden Dissertation ein Selektionsprozess eingeführt, welcher bestehende Methoden aufgreift und auf die Problemstellung der Absicherung eines *Logischen Szenarios* überträgt. Hierbei werden ausschließlich simulative Testausführungen betrachtet und keine Orchestrierung der konkreten

Testfälle auf die entsprechende Testinstanz. Das Projekt erfolgt in enger Zusammenarbeit und Abstimmung mit Projekten aus Industrie und Wirtschaft. Es konnten die prinzipiellen Ansätze im Rahmen des VV-Methoden Projekts [120] eingebracht und demonstriert werden. Es konnte unter Mitarbeit von Partnern, wie ZF, Ford und VW, der Methodenbaukasten diskutiert und aufgebaut werden. Ebenso konnten wissenschaftliche Ausarbeitungen und Ansätze aus der Doktorandengruppe, wie z. B. [8], [79], [93], [92], [123], aufgegriffen werden und entsprechend an die Bedürfnisse der Problemstellung angepasst werden. Diese Dissertation gibt einen Überblick über die bereits entwickelten Methoden und stellt gleichzeitig einen Ansatz zur späteren Aggregation der Ergebnisse zu einer Repräsentativitätsmetrik eines *Konkreten Szenarios* innerhalb eines *Logischen Szenarios* vor.

1.4 Aufbau

Die folgende Dissertation gliedert sich in insgesamt acht Kapitel. Das erste Kapitel gibt zunächst eine allgemeine Einführung und Motivation in das Themenfeld. Kapitel 2 analysiert und vergleicht verschiedene automotive Entwicklungs- und Testprozesse, die als Grundlage für den Selektionsprozess dienen.

Kapitel 3 widmet sich dem Stand der Wissenschaft und Technik im Hinblick auf szenarienbasierte Ähnlichkeitsmaße. Dabei werden bestehende Ansätze erörtert, um darauf aufbauend Anforderungen und Rahmenbedingungen für den in dieser Arbeit vorgestellten Selektionsprozess für *Konkrete Szenarien* zu formulieren.

In Kapitel 4 werden die Anforderungen, Rahmenbedingungen und Schnittstellen für einen Selektionsprozess von *Konkreten Szenarien* aufgelistet und der Selektionsprozess in den in Kapitel 2 vorgestellten Entwicklungsprozess eingeordnet.

Anschließend wird in Kapitel 5 ein Methodenbaukasten präsentiert, der die in Kapitel 3 dargestellten Ähnlichkeitsbausteine systematisiert und auf den Selektionsprozess überträgt. Die Ergebnisse der jeweiligen Methodenbausteine werden

zusammengefasst und aggregiert, um eine Gesamtaussage über die Repräsentativität des übergeordneten *Logischen Szenarios* zu treffen.

Kapitel 6 beschreibt das Design und die Realisierung des Selektionsprozesses. Aufbauend auf dem Methodenbaukasten aus Kapitel 4 wird in Kapitel 7 exemplarisch der Selektionsprozess an einem *Logischen Szenario* demonstriert. Gleichzeitig werden die gewonnenen Erkenntnisse zusammengefasst und diskutiert. Abschließend bietet Kapitel 8 eine Zusammenfassung und Ausblick auf potenzielle Weiterentwicklungen (vgl. Abbildung 1.4).



Abbildung 1.4: Gliederung und Struktur der Dissertation

2 Automotive Entwicklungs- und Testprozess

2.1 Strukturierung und Einordnung der Begriffe

Zur besseren Einordnung der zentralen Begrifflichkeiten dieser Dissertation wurde zusätzlich zum Begriffsverzeichnis eine Ontologie erstellt. Diese ordnet die Begriffe eindeutig ein und setzt sie in den entsprechenden Kontext. Mit Hilfe von Verlinkungen werden die Begriffe in Relation zueinander gesetzt. Die Wissensrepräsentation erfolgt in der Darstellungsform eines UML-Diagramms, das die Strukturierung und Visualisierung der Begriffe aus dem Begriffsverzeichnis unterstützt. Auf Basis der in Abbildung 2.1, Abbildung 2.3, Abbildung 2.7 und Abbildung 2.8 dargestellten UML-Diagramme, ergeben sich die Zusammenhänge des automotive Entwicklungs- und Absicherungsprozesses.¹

¹ Die **Ontologie** basiert dabei auf einer Ausarbeitung von [110] und erweitert die bestehende szenarienbasierte Sicht um Begriffe im Testumfeld. Die vollständige Version der Begriffsontologie ist unter abrufbar. Die Begriffe der folgenden Abschnitte befinden sich eingeordnet in der [https://webprotege.stanford.edu/#projects/861200d1-d9d7-4e66-bd34-37301f2baf5d/edit/Classes?selection=Class\(owl:Thing\)](https://webprotege.stanford.edu/#projects/861200d1-d9d7-4e66-bd34-37301f2baf5d/edit/Classes?selection=Class(owl:Thing)) Ontologie.

2.2 Produktentstehungsprozess in der Automobilindustrie

2.2.1 Fahrzeugdomänen

Allgemein ist eine Domäne eine Bezeichnung für ein Fach- oder Wissensgebiet, welches die Gesamtheit des Wissens innerhalb von Themenbereichen beschreibt. Eine Domäne ist Gegenstand einer inhaltlichen Spezialisierung. Bei der Entwicklung von Steuergeräten und Funktionalitäten von Fahrzeugen unterscheidet man ebenfalls zwischen unterschiedlichen Teilbereichen (sogenannte Domänen). Jeder Teilbereich stellt einen abgeschlossenen funktionalen Umfang dar [90]. Die Begrifflichkeiten werden in einem UML-Diagramm in den Kontext eingeordnet (vgl. Abbildung 2.1).

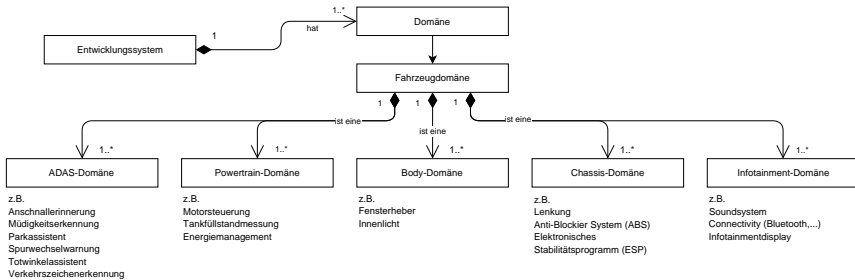


Abbildung 2.1: Begriffliche Einordnung - Fahrzeugdomäne

Body Domäne: Die Fahrzeug Body Domäne beinhaltet alle Passagierfunktionalitäten, welche in einem Fahrzeug erlebt werden. Beispielhafte Systeme sind hierfür das Klimasteuergerät, welches in Abhängigkeit vom Passagierwunsch und der aktuellen Temperatur die Lüftung bzw. Heizung ansteuert [90].

Chassis Domäne: Die Chassis-Domäne ist eine Fahrzeugdomäne, bei der Fahrkomfort und Sicherheit im Vordergrund stehen. In die Chassis-Domäne fallen

vor allem die grundlegenden Sicherheitssysteme, welche die Fahrzeugstabilität sicherstellen [90].

ADAS Domäne: Die Advanced Driver Assistance System (ADAS) -Domäne umfasst alle erweiterten Fahrerassistenzsysteme im Fahrzeug. Fahrerassistenzsysteme unterstützen den Fahrer oder übernehmen die Fahraufgabe, wie beispielsweise ein Abstandsregeltempomat. Diese Systeme überwachen die Umgebung mit Hilfe von Sensoren, die dann je nach System und Situation, Informationen an den Fahrer weitergeben oder selbst die Fahraufgabe übernehmen [90].

Powertrain Domäne: Die Powertrain-Domäne (Antrieb) beinhaltet Steuergeräte, welche bspw. den Motor oder das elektronische Stabilitätsprogramm (ESP) steuern [90].

Infotainment Domäne: Die Infotainmentdomäne ist eine Fahrzeugdomäne, die keinerlei sicherheitsrelevante Bedeutung hat, sondern der Unterhaltung und der Unterstützung des Fahrers dient.

2.2.2 Automatisierungsstufen

Vom amerikanischen Ingenieursverband SAE International wurden sechs Automatisierungsstufen von E/E Funktionen in der Fahrzeugdomäne definiert (vgl. Abbildung 2.2) [97]. Neben der Einordnung in die jeweilige Domäne lassen sich die vom Fahrzeug bereitgestellten Fahrfunktionen anhand dieser Stufen einordnen. Dieses Stufenmodell unterscheidet sich im Grad der Automation in Bezug auf die Übernahme von Längs- und Querregelung, sowie der korrekten Reaktion auf Umgebungseinflüsse und Ereignisse [97]. Mit einer höheren SAE-Stufe nimmt gleichzeitig zur spezifizierten Operational Design Domain (ODD)² auch die Anforderung an Redundanzen im System zu. Mit fortschreitender Automatisierung nimmt die aktive Rolle des Fahrers ab und erlaubt den Passagieren, sich vermehrt fahrfremden Tätigkeiten zu widmen [97]. Auf SAE Stufe 0 er-

² **ODD:** Betriebsumgebung, in welcher die Funktion genutzt wird

Die fünf Stufen zum autonomen Fahren

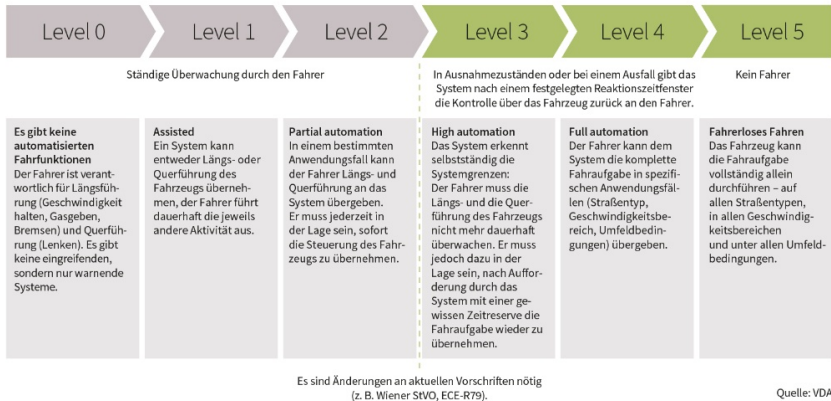


Abbildung 2.2: Automatisierungsstufen zum automatisierten Fahren [97]

folgt kein aktiver Eingriff des Fahrzeugs in die Quer- und Längsführung des Fahrzeugs. Der Fahrer ist allein verantwortlich. In der ersten SAE Stufe finden Basis-Fahrerassistenzsysteme ihre Einordnung. Durch die Übernahme in kritischen Situationen bzw. Fahraufgaben (z. B. Einparken oder Notbremsung) wird dem Fahrer eine Unterstützung gegeben, welche Fahrkomfort- und Sicherheitsaspekte verbessert. Die Fahrerassistenzsysteme auf SAE Stufe 1 müssen vom Fahrer voll überwacht werden. Innerhalb SAE Stufe 2 wird sowohl die Längsführung, als auch die Querführung, vom System übernommen (Teilautomation). Erst ab SAE Stufe 3 spricht man vom hochautomatisierten Fahren, bei welchem der Fahrer nur noch als Rückfallebene dient und das System seine Systemgrenzen selbstständig überwacht. Der Fahrer muss dabei immer in der Lage sein, das System vollständig zu übernehmen. In Deutschland wurde 2022 die Mercedes S-Klasse als erstes Stufe 4 System zugelassen [116]. In der SAE Stufe 4 dem vollautomatisierten Fahren bewegt sich das Fahrzeug in den vorgegebenen Systemgrenzen, z. B. auf spezifisch festgelegten Straßenabschnitten, wie auf Autobahnen (innerhalb der ODD), ohne den Fahrer als Rückfallebene. Der Fahrer muss sich nicht mehr dem Verkehr widmen und hat die Option andere Aufgaben zu übernehmen. Ab SAE Stufe 5

wird der Fahrer zur Erfüllung der Fahraufgabe vollständig durch das System ersetzt. Das System ist in allen Situationen und Betriebszuständen in der Lage, das Fahrzeug zu steuern [17] [97] [8].

2.2.3 Entwicklungsprozesse

Ein Entwicklungsprozess gibt einen Konzeptrahmen von Prozessen und Aktivitäten vor, die sich mit dem Lebenszyklus eines Produktes befassen. Ein Produkt oder System ist dabei beispielsweise die komplette Elektrik- /Elektronikarchitektur eines Fahrzeuges, aber auch eine einzelne Komponente oder Funktion [101] [119]. Der Entwicklungsprozess ist in Stufen organisiert und dient als gemeinsame Re-

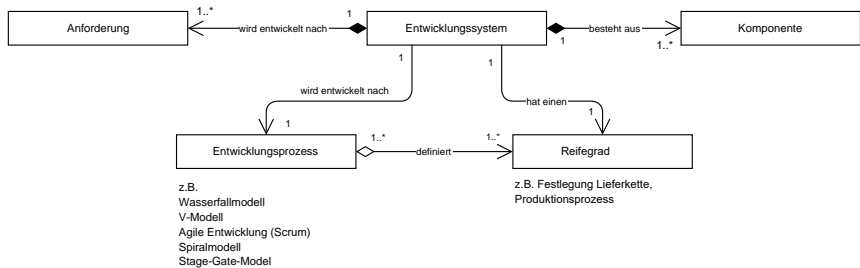


Abbildung 2.3: Begriffliche Einordnung - Entwicklungsprozess

ferenz für die Kommunikation und das Verständnis innerhalb der Entwicklung eines Systems (vgl. Abbildung 2.3). Der aktuelle Stand der Technik zeigt unter anderem folgende Entwicklungsmodelle auf:

- **Wasserfallmodell**
- **V-Modell**
- **Agile Entwicklungsmethoden**, wie Scrum (nähere Informationen siehe [103] und [94])
- **Spiralmodell** (nähere Informationen siehe [23])

- **Stage-Gate-Model** (nähere Informationen siehe [39])

Im automotive Kontext wird hauptsächlich das V-Modell zur Entwicklung des Gesamtsystems und agile Methoden für die Entwicklung von Software verwendet [88] [101].

Wasserfallmodell:

Das Wasserfallmodell ist in seiner Reinform ein lineares, nicht iteratives Entwicklungsmodell. Ursprünglicherweise wurde dieses Modell für die Softwareentwicklung verwendet und strukturiert sich in streng aufeinanderfolgende Projektphasen. „Wie bei einem Wasserfall „fallen“ die Ergebnisse einer Stufe nach unten in die nächste und sind dort verbindliche Vorgaben.“ [83] Die Ein- und Ausgangsdokumente und -ergebnisse sind beim Wasserfallmodell eindeutig definiert. Am Ende jeder Aktivität steht ein Ergebnisdokument. Die praktische Anwendung des Wasserfallmodells eignet sich für Entwicklungen, bei denen Leistungen, Abläufe in der Projektdurchführung und Anforderungen bereits vorab genau bekannt sind. Die Aktivitäten beginnen bei der Anforderungserhebung und der Definition von Lasten- und Pflichtenheft über die Systementwicklung bis hin zum Systemtest bzw. zur Inbetriebnahme. Dabei baut jede Aktivität direkt auf der nächsten auf. Im klassischen Wasserfallmodell sind keine Rücksprünge auf das vorherige Element erlaubt. Die Integration von allen Teilkomponenten erfolgt spät im Entwicklungsprozess. Das Wasserfallmodell ist daher ein sehr starres Entwicklungsmodell. Erst durch die Erweiterung des Modells und durch das Zulassen von Rücksprüngen gewinnt das Wasserfallmodell an Flexibilität und lässt Anpassungen im Systementwurf aufgrund einer Erkenntnis beim Testen zu.

V-Modell:

Das V-Modell ist eine Weiterentwicklung des Wasserfallmodells. Bei der Entwicklung von eingebetteten Systemen in der Automobilindustrie ist das V-Modell eines der am häufigsten angewendeten Entwicklungsmodelle [112]. Der jeweilige Umfang und die Ausprägung des Entwicklungsprozesses hängen vom zu entwickelnden System und den beteiligten Rollen ab.

Das V-Modell unterscheidet sich vom Wasserfallmodell durch die Querbeziehung der Teilprozesse. Die zugehörigen Tests werden beim V-Modell bereits während der Anforderungsanalyse (z. B. während der Erhebung von Anforderungen oder Architekturentwürfen) spezifiziert. Hierdurch werden unvollständige und fehlerhafte Spezifikationen durch die Einbindung mehrerer Akteure frühzeitig erkannt [57]. In Anlehnung an den Automotive Spice Prozess definiert sich der Entwicklungsprozess eines V-Modells im automotive Kontext in verschiedene Teilprozesse (vgl. Abbildung 2.4) [56] [32].

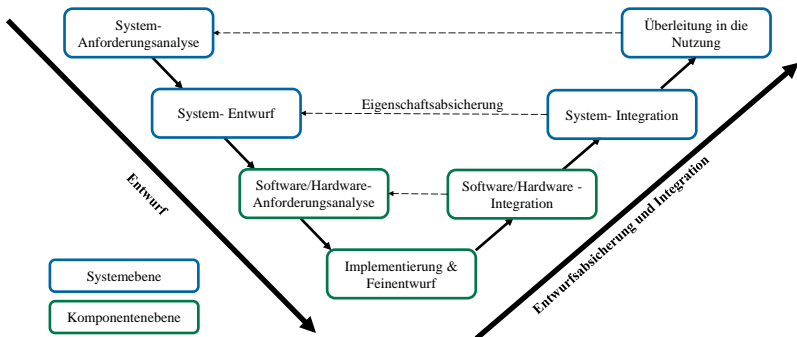


Abbildung 2.4: V-Modell in Anlehnung an [56]

Die einzelnen Teilprozesse sind dabei ähnlich zum Wasserfallmodell aufgebaut und gliedern sich in:

- **System-Anforderungsanalyse:** Im Rahmen der Anforderungsanalyse werden die Anforderungen von verschiedenen Stakeholdern (Kunden, Nutzern, etc.) gesammelt. Die Anforderungen werden in Form von Use Cases oder User Stories abgelegt und dienen als Basis für die Definition von Arbeitsprodukten [101] [56]. Außerdem werden die aufgestellten Anforderungen analysiert und bezüglich der Risiken in Bezug auf die organisatorische und

technische Umgebung bewertet. Dies dient zur Aufwandsabschätzung und zur ersten Strukturierung des Systems in Bezug auf Daten, Funktionen, Schnittstellen und Objekte [56].

- **System-Entwurf:** In diesem Teilprozess wird das Gesamtsystem in einzelne Module und Komponenten aufgeteilt. Jede Komponente bzw. jedes Modul erfüllt unterschiedliche Systemanforderungen [101]. Gleichzeitig werden interne und externe Schnittstellen der einzelnen Komponenten definiert [32].
- **Software/Hardware-Anforderungsanalyse:** Der Systementwurf wird im nächsten Schritt in eine Komponentenarchitektur zerlegt. Das heißt, dass die Gesamtfunktionalität des Systems in einzelne Software- und Hardwarekomponenten und Wirkketten unterteilt wird [56] [32].
- **Software Design:** Im Software Design werden die verschiedenen Anforderungen auf Softwareebene in einen Komponentensoftwareentwurf unter Berücksichtigung der gegebenen Software- und Hardware-Anforderungen und der definierten Schnittstellen überführt.
- **Implementierung und Feinentwurf:** Klassischerweise erfolgt hier die Implementierung und der Feinentwurf der einzelnen Komponenten. Die Implementierungsarbeiten orientieren sich dabei an den gestellten Anforderungen und der Komponentenarchitektur. Ziel dieses Schrittes ist es, einen funktionierenden Code zu entwickeln, welcher alle Anforderungen und damit auch alle Testfälle erfolgreich abdeckt [32].
- **Software- Integration:** Die Komponententests testen die einzelnen Komponenten in einer spezifizierten Anzahl an Testfällen gegen die Komponentenanforderung der Software.
- **System-Integration:** Die einzelnen Komponenten werden zu einer Wirkkette kombiniert. Anschließend erfolgt eine Prüfung der integrierten Komponenten hinsichtlich der definierten Systemanforderungen. Nach der Integration der Steuergeräte in den Systemverbund – z. B. ein Verbund von

Steuergeräten im Fahrzeug – erfolgt der Integrationstest, d. h. der Test des Gesamtsystems gegen den Systementwurf [101] [32].

- **Überleitung in die Nutzung:** Am Ende des Entwicklungsprozesses steht die Freigabe und Systemvalidierung gegen z. B. Kundenanforderungen oder gesetzliche Regularien [32] [101].

Die verschiedenen Entscheidungen und Annahmen, welche in jedem Entwicklungsschritt getroffen werden, haben direkte Konsequenzen für die darauffolgenden Entwicklungsschritte. Fehler in frühen Entwicklungsschritten führen zu teuren und aufwendigen Korrekturen [101] [119].

Das Design, die Entwicklung und der Test auf Komponentenebene findet erneut in einem unterlagerten Entwicklungsprozess statt, welcher oftmals in der Automobilindustrie von externen Zulieferern übernommen wird. Verschiedene Entwicklungs-V-Modelle greifen daher ineinander, sodass beim Fahrzeug die letzte Integration der vom Zulieferer entwickelten Funktionalität im Fahrzeug beim Fahrzeughersteller stattfindet [65].

2.2.4 Reifegrade in der automotive Produktentwicklung

Die Gesamtintegration und der Test der einzelnen elektronischen Steuergeräte in der Automobilindustrie erfolgt in verschiedenen Reifegraden.

Das Reifegradabsicherungs-Modell besteht nach [119] aus acht Reifegraden, welchen den Entwicklungszyklus von der Konzeptphase, bis hin zur laufenden Serie nach dem Start-Of-Production (SOP) begleiten. Diese Reifegrade definieren sich folgendermaßen und dienen als Meilensteine im Produktentwicklungsprozess [119]:

- RG0: Innovationsfreigabe für Serienentwicklung
- RG1: Anforderungsmanagement für Vergabeumfang
- RG2: Festlegung der Lieferkette und Vergabe der Umfänge

- RG3: Freigabe der technischen Spezifikation
- RG4: Produktionsplanung abgeschlossen
- RG5: Serienwerkzeugfallende Teile und Serienanlagen verfügbar
- RG6: Produktionsprozess und Produktfreigabe
- RG7 (nach SOP): Projektabschluss, Verantwortungsübergabe an Serie

Das Gesamtprodukt entsteht inkrementell in mehreren Durchgängen des Entwicklungsprozesses durch das Zusammenführen von verschiedenen Teilkomponenten (vgl. Abbildung 2.5).

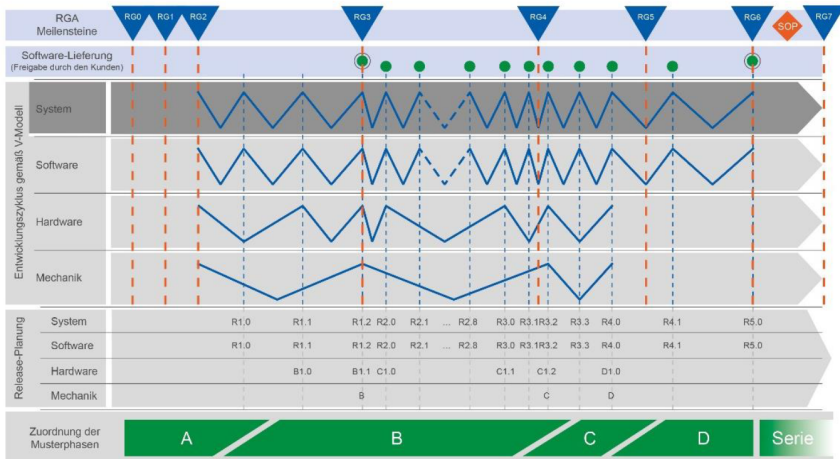


Abbildung 2.5: Beispielhafte Taktung von Hardware, Software und Mechanik nach Reifegrad (entnommen aus [119])

Jeder Durchlauf des Entwicklungsprozesses muss von den entsprechenden Testinstanzen getestet werden. Ziel ist es, unvorhersehbare Integrationsprobleme am Ende zu verhindern, indem durch kleine Produktinkremente schon früh Realfahrzeugtests und Tests an Hardware-in-the-loop Prüfständen durchgeführt werden. In

der Automobilindustrie durchlaufen neue Softwarestände mit einer Iterationszeit von ca. zwei Wochen den kompletten Testprozess [119].

2.2.5 Funktionale Sicherheit und ASIL-Einstufung im Entwicklungsprozess

Die funktionale Sicherheit spielt eine zentrale Rolle im Entwicklungsprozess hochautomatisierter Fahrfunktionen. Mit steigender Komplexität und Automatisierung von Fahrfunktionen nimmt auch die sicherheitsrelevante Verantwortung der Systeme zu. Eine strukturierte Herangehensweise zur Analyse und Bewertung potenzieller Risiken ist daher unabdingbar. Der in der ISO-Norm ISO 26262 [49] definierte Ansatz zur funktionalen Sicherheit von sicherheitsrelevanten elektrischen und elektronischen Systemen in Serienfahrzeugen bildet in diesem Kontext den industriellen Standard.

Ein zentrales Element der ISO 26262 ist die Bestimmung des Automotive Safety Integrity Level (ASIL). ASIL beschreibt die sicherheitsbezogene Kritikalität eines Systems oder einer Funktion und leitet sich aus einer systematischen Gefährdungsanalyse ab. Dabei werden drei Hauptkategorien betrachtet:

- Severity (S) – Wie schwerwiegend sind die möglichen Konsequenzen eines Ausfalls?
- Exposure (E) – Wie häufig ist der entsprechende Betriebszustand, in dem die Gefährdung auftreten kann?
- Controllability (C) – Inwieweit kann das System die Situation noch kontrollieren?

Auf Basis dieser Bewertung wird eine ASIL-Einstufung vorgenommen. Die Stufen reichen von ASIL A (Maßnahme erforderlich) bis ASIL D (umfassende Maßnahmen erforderlich). Funktionen, die keinem nennenswerten Risiko zugeordnet werden können, werden mit dem Status QM (Quality Management) versehen und unterliegen keinen sicherheitsgerichteten Anforderungen der ISO 26262. [49]

Die ASIL-Kategorisierung beeinflusst maßgeblich die Entwicklungstiefe, die Testanforderungen sowie die notwendigen Sicherheitsmaßnahmen innerhalb des V-Modells (vgl. Kapitel 2.2.3). Für Funktionen mit ASIL C oder D sind z.B. umfangreiche Nachweise durch formale Methoden, umfassende Absicherung und Rückverfolgbarkeit erforderlich. Bei automatisierten Fahrfunktionen im SAE Level 3 und höher ist aufgrund der Systemverantwortung und potenziellen Systemgrenzen in vielen Fällen eine ASIL D-Einstufung erforderlich.

Im weiteren Verlauf dieser Dissertation wird der ASIL-Wert genutzt, um die Bedeutung (Severity) innerhalb der FMEA-basierten Bewertung der Methodenbausteine zu bestimmen (vgl. Kapitel 5.4.2). Auch die Aspekte Exposure und Controllability finden dort Eingang in die Gewichtung und Priorisierung der Unsicherheitsfaktoren.

2.3 Testinfrastruktur

Klassischerweise werden Funktionalitäten im automotive Kontext entlang des V-Modells oder durch agile Methoden entwickelt (vgl. Kapitel 2.2). Hierzu muss, zur Verifikation und Validierung von Funktionen, eine Testinfrastruktur aufgebaut werden. Diese Testinfrastruktur hat das Ziel, die konzeptionierten Anforderungen und Funktionalitäten entsprechend zu gewährleisten und abzusichern. Das Ergebnis des Testens ist die Erfüllung des Testziels.

Definition 1. Testinfrastruktur. Die Testinfrastruktur sind die organisatorischen Elemente, die für die Durchführung des Tests benötigt werden, bestehend aus Testumgebung, Testhilfsmitteln, Testmethoden, usw.. Nach [46].

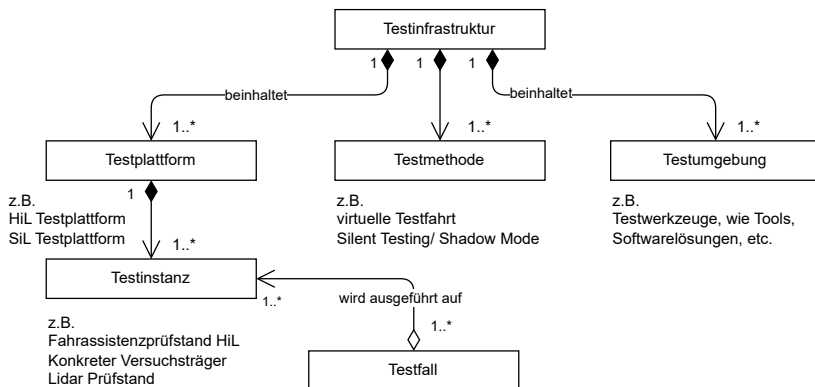


Abbildung 2.6: Zusammenhang der Begrifflichkeiten im Testumfeld

Als relevante Begriffe innerhalb der Testinfrastruktur definieren sich nach [99], [46] und [3] die Begriffe Testplattform, Testumgebung und Testmethode:

Definition 2. Testplattform. Die Testplattform ist Teil der Testinfrastruktur. Sie definiert die Umwelt für das Testobjekt, die benötigt wird, um Tests auszuführen.

Diese Darstellung lässt sich beschreiben durch Parameter, wie technische Infrastruktur, organisatorische Infrastruktur, Interaktion, klimatische Besonderheiten und weitere Gestaltungs- und Störgrößen, z. B. Fahrzeug, Prüfgelände, HiL, SiL, MiL, ViL. Nach [99] [46].

Testplattformen für die Entwicklung von Fahrzeugsystemen zeigt Kapitel 2.3.1.

Definition 3. Testhilfsmittel. *Ein Testhilfsmittel ist ein Element der Testinfrastruktur. Es beinhaltet alle Artefakte, die während des Testprozesses erstellt werden und die erforderlich sind, um die Tests zu planen, zu entwerfen oder auszuführen. Dazu gehören: Dokumente, Skripte, Eingabedaten, erwartete Ergebnisse, Prozeduren zum Aufsetzen und Aufräumen von Testdaten, Dateien, Datenbanken, Umgebungen und weitere zusätzliche Software- und Dienstprogramme, die für das Testen verwendet werden. Nach [99] [46].*

Definition 4. Testmethode. *Die Testmethode ist eine Kombination von Tätigkeiten zum systematischen Erzeugen eines Testproduktes. Man unterscheidet dabei statische Testverfahren, bei denen der Programmcode nicht ausgeführt wird, von dynamischen Testverfahren, bei denen der Programmcode ausgeführt wird. Die gewählte Testmethodik beeinflusst damit die Testfallerstellung und ist Teil des Testfalls (Definition Testfall - siehe Kapitel 2.4). Nach [99] [46].*

2.3.1 Testplattform

Testplattformen lassen sich in Abhängigkeit von dem Grad der Integration in das Gesamtsystem gemäß Steimle et al. [109] in real, semi-real und simuliert (vgl. Abbildung 2.7) unterteilen.

Bei virtuellen Testplattformen werden die fehlenden Teile des Gesamtsystems durch simulative Anteile abgedeckt (z. B. Restbussimulation des fehlenden Fahrzeugbusses). Die Testplattform beschreibt dabei den prinzipiellen Aufbau bzw. Ansatz. Spricht man von einer konkreten Ausprägung einer Plattform, wie zum Beispiel dem Fahrassistenz-Hardware-in-the-loop Prüfstand, nennt man diese

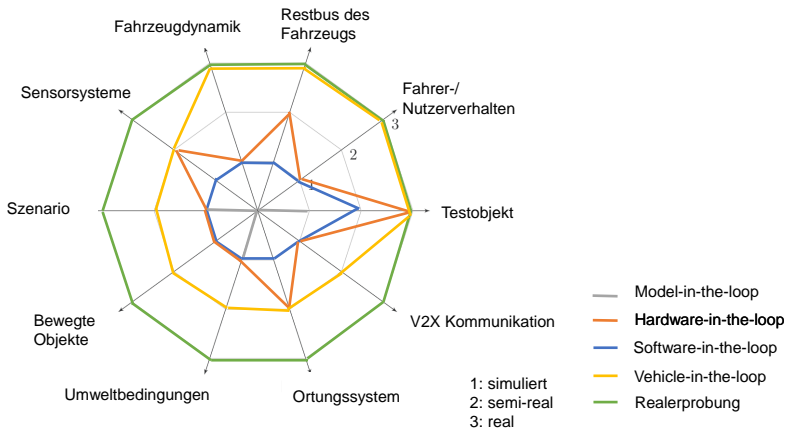


Abbildung 2.7: Einteilung der Testplattformen nach Grad der Integration nach [109]. Dabei variiert die genaue Ausprägung der einzelnen Kategorien einer jeweiligen Testinstanz in Abhängigkeit des Testziels.

konkrete Ausprägung Testinstanz. Die konkreten Testinstanzen werden im rechten Teil des V-Modells zur Verifikation und Absicherung von Funktionalitäten verwendet. Dabei ist zu beachten, dass nicht jede Funktionalität im Fahrzeug auf jeder Testplattform getestet werden kann. Die Ausführung von *Konkreten Szenarien* erfolgt an Testinstanzen (vgl. Kapitel 2.4).

Model-in-the-loop

Model-in-the-loop Testumgebungen dienen zum simulativen Testen von einzelnen Softwarekomponenten in frühen Reifegraden, in denen noch keine Hardware vorhanden ist. Modellbasierte Ansätze sind ein kosteneffizienter Weg, um Fehler in den Algorithmen des Systems zu finden. Hierbei sind alle Teile des eingebetteten Systems und auch die Umwelt simuliert. Es werden physikalische Modelle benötigt. Model-in-the-loop Testplattformen zählen zu den rein virtuellen Absicherungen [109].

Software-in-the-loop

Bei diesem Testverfahren wird das Testsystem auf einem Computer ausgeführt und der entsprechende Softwareprogrammcode in eine mathematische Simulationsumgebung eingebaut (wie z. B. MATLAB/Simulink oder dSpace). Testfälle müssen nicht in Echtzeit ausgeführt werden und lassen sich parallelisieren sowie zu jeder Zeit neu starten. Es wird dabei nicht die Zielhardware des Steuergeräts verwendet [109]. Fehler in der Logik oder der Funktion des Codes werden deshalb oft auf dieser Testplattform gefunden, wohingegen das Laufzeitverhalten und das Verbundverhalten von mehreren Steuergeräten auf dieser Plattform nicht durchgeführt werden kann [108].

Hardware-in-the-loop

Bei einer Hardware-in-the-loop (HiL) Testplattform werden ein oder mehrere eingebettete Systeme (z. B. reale elektronische Steuergeräte oder reale mechatronische Komponenten) über seine Ein- und Ausgänge an ein angepasstes Gegenstück (einen HiL-Simulator) angeschlossen, das als Nachbildung der realen Umgebung des Systems dient. Das zu steuernde System (z. B. Auto) wird über Modelle simuliert, um die korrekte Funktion der zu entwickelnden Steuereinheit (z. B. eine Motorsteuerung) zu verifizieren [101] [14]. Die HiL-Simulation muss in Echtzeit ablaufen und wird im Entwicklungsprozess benutzt, um den Testaufwand zu verteilen, um in frühen Reifegraden, in denen noch kein reales Fahrzeug vorhanden ist, bereits Testaktivitäten zu starten. Insbesondere lassen sich wiederkehrende Abläufe simulieren. Dies hat den Vorteil, dass ein neuer Softwarestand des Testsystems unter den gleichen Rahmenbedingungen, wie Temperatur, getestet wird, wie die Vorgängerversion. Somit wird nachgewiesen, ob ein Fehler beseitigt wurde oder nicht. Dadurch wird auch das Systemverhalten in Grenzsituationen erprobt, ohne dabei Auto und Fahrer zu gefährden. Die HiL-Simulation bedeutet eine Vereinfachung der realen Konfiguration und ersetzt daher den Test am Zielsystem nie ganz. Weicht die HiL-Simulation zu stark von der Realität ab, ist das gewählte HiL-Simulations-Modell für das betrachtete Testziel nicht gültig. Inzwischen wird die HiL-Simulation nicht nur dann angewendet, wenn die Testumgebung

noch nicht vollständig vorhanden ist, sondern als zusätzliche Teststufe bei der Integration der Software [27] [109].

Vehicle-in-the-loop

Die Gültigkeit und damit auch die Verwendbarkeit der Ergebnisse hängt beim simulationsbasierten Ansatz stark von der Qualität der verwendeten Simulationsmodelle ab. In manchen Grenzsituationen reichen die aufgestellten Simulationsmodelle nicht aus. Möchte man den Einfluss von Wettereinflüssen auf die Fahrzeugdynamik für alle Bremssituationen bewerten, kommt ein rein simulationsbasierter Ansatz, aufgrund der Modellgenauigkeit, an seine Grenzen. Der Test von Grenzsituationen auf traditionellen Prüfgeländen ist wiederum mit Herausforderungen und Kosten in Bezug auf Reproduzierbarkeit und Sicherheitsvorkehrungen verbunden. Die Vehicle-in-the-Loop-Technologie (ViL) führt eine Testplattform ein, welche ein reales Testfahrzeug verwendet, bei dem die Sensorinformationen durch eine Simulation ermittelt werden. Dadurch entsteht die Option, virtuelle Objekte in das Fahrzeugumfeld einzufügen [87] [20]. Die Position des Fahrzeugs in der virtuellen Welt wird geschätzt und in die Steuergeräte zurückgeführt. Dadurch lassen sich innerhalb der Simulation sicherheitskritische Szenarien generieren, während das Verhalten des realen Fahrzeugs getestet wird [112].

Realerprobung

Die Realerprobung stellt die realistischste Art der Erprobung dar. Dabei wird das Fahrzeug auf einem Prüfgelände bzw. in späteren Reifegraden auch auf öffentlichen Straßen getestet. Die Realerprobung ist nach [87] gleichzeitig allerdings auch die aufwendigste und kostenintensivste Testplattform. Bei dieser Testplattform wird das Fahrzeug unter realistischen Rahmenbedingungen getestet. Es werden unter anderem verschiedene Wettersituationen auf dem Prüfgelände simuliert. Die Vorbereitung und Herstellung der verschiedenen benötigten Situationen (Wintererprobung, etc.) muss aufgrund der begrenzten Verfügbarkeit von zum Beispiel Tagen mit Schnee gut geplant werden. Die Realerprobung ist aufgrund des zeitlichen und finanziellen Aufwands für eine skalierte Ausführung von Testfällen nicht geeignet [11].

2.4 Szenarien, KPI und Testfälle im automotive Kontext

2.4.1 Szenarien

Szenarienbasierte Ansätze werden in Forschungsarbeiten, wie [11], [100], [50], [92] und Forschungsprojekten, wie Pegasus und VV-Methoden dazu verwendet, bei der Entwicklung und Freigabe von Fahrfunktionen zu unterstützen. Das in Kapitel 2.2 vorgestellte V-Modell wird neben den agilen Ansätzen als Entwicklungsprozess (z. B. nach ISO 26262) für die Entwicklung von automatisierten Fahrzeugen angewendet [82] [120] [49]. Szenarien lassen sich in mehreren Prozessphasen anwenden. Jede Prozessphase stellt unterschiedliche Anforderungen an die jeweilige Darstellung der Szenarien. Ziel ist es entwicklungsbegleitend einen Szenarienkatalog aufzubauen. In der Konzeptphase müssen die Szenarien sprachlich formuliert werden, sodass diese vom Menschen verstanden werden. Zur Konkretisierung und Detaillierung während des Entwicklungsprozesses müssen die Szenarien in Parameterbereiche überführt werden. Im Test- und Validierungskontext müssen die Szenarien ausführbar und genau definiert sein, um sie auf der jeweiligen Testinstanz reproduzierbar zu testen [82] [100].

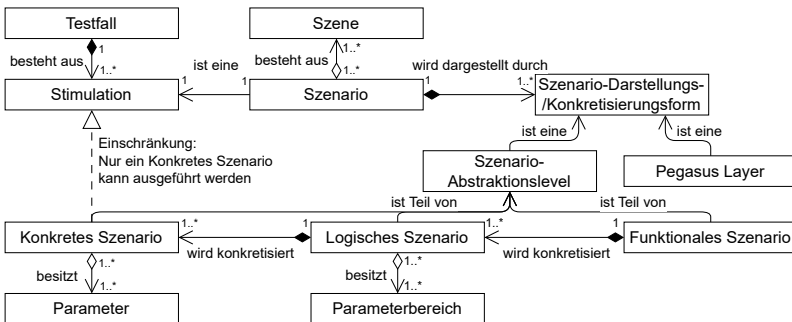


Abbildung 2.8: Begriffliche Einordnung - Szenario

Definition 5. Szenario. *Ein Szenario ist eine Stimulation. Es ist die zeitliche Abfolge oder Entwicklung einer Reihe von Ereignissen oder Aktionen mit jeweiligen Situationen der Teilnehmer in einer Szenerie (alle georäumlich stationären Elemente). Jedes Szenario beginnt mit einer Anfangsszene. Anders als eine Szene erstreckt sich ein Szenario über eine bestimmte Zeitspanne. Szenarien lassen sich in die Abstraktionsebenen Konkret, Logisch und Funktional unterscheiden. Nach [96] [54] [3] [110]*

Das Szenario ist dabei Teil eines Testfalls.

Definition 6. Testfall. *Ein Testfall besteht aus einer Stimulation (z. B. Szenario) und aus einer oder mehreren Bewertungskriterien. Ein Testfall beinhaltet neben der Stimulation und den Bewertungskriterien noch weitere Kontextinformationen zur Testinfrastruktur in Bezug z. B. auf die eingesetzte Testmethode, Testziel, das SuT. Außerdem beinhaltet ein Testfall eine Testinstanz, wie Software-in-the-loop Prüfstand oder Hardware-in-the-loop Prüfstand oder Testfahrt auf einem Testgelände. [110]*

Je nach Abstraktionsebene beinhaltet ein Testfall Informationen über (vgl. Abbildung 2.9):

- **das System-under-Test (SuT)** - Statische, technische und funktionale Beschreibung des zu entwickelnden Systems.
- **das Testziel** - Anforderungen an das SuT im Bezug auf das Verhalten im Szenario. Operationalisierungen des Sollverhaltens.
- **die SuT Performance/ Metriken** - Messvorschriften und Bestehenskriterien zur Bewertung des beobachtbaren SuT Verhaltens in Bezug auf die Erreichung der Testziele.
- **die Testinfrastruktur** - Rahmenbedingungen und Definition der Infrastruktur zur Testdurchführung (vgl. Kapitel 2.3).

Ein Szenario lässt sich nicht direkt an einer Testinstanz ausführen. Hierzu muss das Szenario, welches eine Form der Stimulation ist, durch das Ergänzen von Testanforderungen in einen Testfall überführt werden (vgl. Abbildung 2.9). Die Testanforderungen beinhalten neben Bewertungskriterien für das SuT auch weitere relevante Kontextinformationen für die Testausführung, wie die geforderte Güte der Testumgebung oder die Referenz auf das übergeordnete Testziel. Ein Testfall

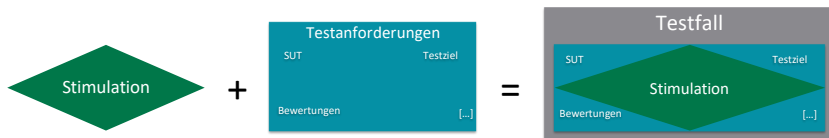


Abbildung 2.9: Der Testfall im Kontext des Szenarios nach [120]

muss im Laufe des Entwicklungsprozesses einer Testinstanz im Rahmen der Testorchestrierung³ zugeordnet werden und wird dort anschließend ausgeführt. Als Ergebnis der Ausführung eines Testfalls stehen zum einen Messreihen und KPIs und zum anderen klassischerweise ein pass/fail Ergebnis, welches die Performance bzw. die Erfüllung der Testanforderungen im Kontext der Stimulation bewertet, zur Verfügung. Die Ergebnisse eines Testfalls werden durch die Veränderung der Stimulation, der Bewertung oder durch eine neue Version des SuTs ungültig und müssen in der Konsequenz erneut ausgeführt werden.

Neben der Verwendung von Szenarien in Testfällen lassen sich Szenarien auf unterschiedliche Art und Weise darstellen. Eine nach dem Pegasus Projekt [82] und Menzel et al. [69] etablierte Darstellungsform ist die Unterteilung von Szenarien in drei Abstraktionslevel (*Funktionale, Logische* und *Konkreten Szenarien*). Wobei jedes *Funktionale Szenario* mehrere *Logische Szenarien* und jedes *Logische Szenario* mehrere *Konkrete Szenarien* beinhaltet. Die Anzahl der Szenarien

³ Die **Testorchestrierung** beschreibt die Zuordnung eines Testfalls auf eine Testinstanz unter Berücksichtigung von Einflussfaktoren, wie der Validität der Testinstanz, der Verfügbarkeit und der Kosten der Ausführung

nimmt zu und das Abstraktionslevel nimmt mit zunehmender Konkretisierung ab (vgl. Abbildung 2.10).

Funktionales Szenario: Ein *Funktionales Szenario* beschreibt den situativen Kontext in der sich das SuT befindet. Innerhalb eines *Funktionalen Szenarios* werden dabei die groben Umweltbedingungen abgesteckt. Jedes *Funktionale Szenario* beinhaltet sprachliche Beschreibungen zum dynamischen und statischen Ablauf des Szenarios. Hierzu wird das im Rahmen von VV-Methoden und Pegasus [120] entwickelte 6-Layer-Modell zur Beschreibung verwendet. *Funktionale Szenarien* sind abstrakt und nicht direkt an Testinstanzen ausführbar. Jedes *Funktionale Szenario* definiert Parametertypen, die der Verständlichkeit des Szenarios dienen. [120] [82]. Jedes *Funktionale Szenario* besteht aus einem oder mehreren Parameterklassen, welche den Variationskontext des Szenarios näher erläutern.

Logisches Szenario: Ein *Logisches Szenario* beinhaltet variable Parameter, welche innerhalb eines Szenarios variiert werden. Sie geben den Variationsbereich vor innerhalb dessen der Parameter angepasst wird. *Logische Szenarien* beschreiben Parameterbereiche im Zustandsraum über eine Parametervariationsvorschrift oder eine Ober- und Untergrenze [120].

Konkretes Szenario: Ein *Konkretes Szenario* hat ein übergeordnetes *Logisches Szenario*. Ein *Konkretes Szenario* besteht aus einem konkreten Parametersatz. Das *Konkrete Szenario* beschreibt dabei genau den Ablauf und die Umweltbedingungen des Szenarios [120]. Das *Konkrete Szenario* ist eine Stimulationsform. Eine Stimulation ist ein Teil eines Testfalls [82].

Betrachtet man den Einsatz der Szenarien und Testfälle im Entwicklungsprozess, ergeben sich für die unterschiedlichen Szenariotypen unterschiedliche Einsatzbereiche. Bei der Produktkonzeption und Anforderungserstellung unterstützen *Funktionale Szenarien*. Sie helfen den Einsatzbereich und die ODD, der zu entwickelnden Funktion, einzuordnen bzw. zu beschreiben. Es werden grobe funktionale Einsatzbereiche in Form von *Funktionalen Szenarien* definiert. Bei der Entwicklung der Funktion werden die *Funktionalen Szenarien* in *Logische Szenarien* überführt. Hierbei werden die abstrakten *Logischen Szenarien* mit Parametern und Parameterräumen konkretisiert. Der Ablauf des Szenarios wird festgelegt

und es werden die Grenzen des Parameterbereichs gemäß der Systemgrenzen der entwickelnden Funktion zum *Logischen Szenario* hinzugefügt. Abschließend werden beim Test des Systems die *Logischen Szenarien* zu *Konkreten Szenarien* konkretisiert und anschließend auf die verfügbaren Testinstanzen orchestriert. *Konkrete Szenarien* sind zusammen mit einer Bewertung, einem Testziel und einem Testsystem an einer Testinstanz ausführbar [102].

Definition 7. Parameter. *Ein Parameter ist ein in der jeweiligen Simulations- oder Ausführungsumgebung direkt veränderbarer Wert, welcher vor der Durchführung eines Testfalls an einer Testinstanz verändert wird und somit das Szenario beeinflusst. Er wird durch einen binären, nominalen, ordinalen, metrischen oder zeitvarianten Wert quantifiziert. Als Parameter zählen die Geschwindigkeit von Verkehrsteilnehmern, der Sonnenstand, die Beschaffenheit des Untergrunds und weitere Einflussfaktoren. Ein Logisches Szenario besitzt mehrere variable Parameter, welche im Konkreten Szenario variiert werden [110] [120].*

Die Gliederung und Strukturierung der Szenarienabstraktionslevel lassen sich mit Hilfe von Abbildung 2.10 darstellen. Die Szenarien werden dabei durch Parameterotypen, Parametervariationsvorschriften und Parameterbereiche beschrieben. Die Szenarienparameter gliedern sich nach Bamberg [12], Watanabe [43] und Wachenfeld et al. [121] je nach Ausprägung und Art des Parameters in unterschiedliche Skalenniveaus:

- binäre Parameter: Parametertyp mit binärer Ausprägung, wie z. B. Objekt auf Straße vorhanden ja/nein
- nominale Parameter: Parametertyp mit diskreten Werten ohne Ordinalität, wie z. B. Wetter mit verschiedenen diskreten Wetterphänomenen sonnig/regnerisch/ nebelig
- ordinale Parameter: Parametertyp mit diskreten Werten mit Ordinalität, wie z. B. Verkehrsaufkommen auf Straße (niedrig, mittel, hoch)

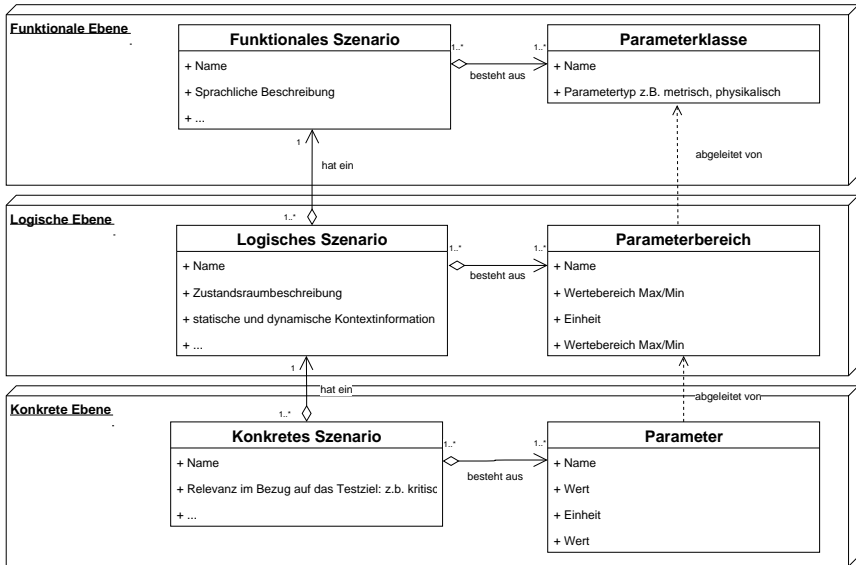


Abbildung 2.10: Aufteilung der Szenarien in die verschiedenen Abstraktionsstufen

- metrische Parameter: Paramtertypen mit einem kontinuierlichen Wertebereich, bei welchen ein max/min Wert oder eine Variationsvorschrift vorgegeben wird, wie z. B. Geschwindigkeit des Ego-Fahrzeugs zu Beginn des Szenarios
- zeitvariante Parameter: Dieser Parametertyp wird für die Beschreibung eines Szenarios nach der Ausführung verwendet, wie z. B. zeitlicher Positionsverlauf eines Fahrzeugs.

Auf jeder Stufe des Pegasus 6-Layer Modell ergeben sich verschiedene Typen von Parametern, welche das Szenario beschreiben. Auf Layer 1 werden Kurvenradien und Spurbreiten vorgeben. In Layer 5 sind Parametertypen enthalten, welche unter anderem Bezug auf das Wetter nehmen. Es wird zwischen konstanten Parametern, welche im Rahmen eines *Logischen Szenarios* nicht verändert werden, und variablen Parametern, welche durch Parameterbereiche (vgl. Abbildung 2.10) variiert werden, unterscheiden. Neben der Zuordnung zum Pegasus 6-Layer Modell

lassen sich die Parameter auch basierend auf die beeinflussende Komponente bzw. Testziel zuordnen.

Szenarien Abstraktionslevel

Ein modernes Fahrzeug besteht aus über 100 Steuergeräten mit über 100 Millionen Zeilen an Programmcode, welche im Rahmen des Entwicklungsprozesses entwickelt und getestet werden müssen [30]. Dabei spielen nicht nur die internen Größen eine Rolle. Der Regelkreis des Gesamtfahrzeugs schließt sich über Umwelteinflüsse. Diese Umwelteinflüsse werden im Rahmen des im Pegasus Projekt entwickelten 6-Layer Szenarienabstraktionsmodell strukturiert und gegliedert [102]. Zur Erweiterung des Modells ergänzen diverse Ansätze [120] eine weitere Sicht, welche die interne Fahrzeugkommunikation als technischen Fahrzeuglayer darstellt (vgl. Abbildung 2.11). Auf den verschiedenen Layern des Modells ergeben sich verschiedene Arten und Typen von Parametern, welche das Szenario bestimmen und beeinflussen.

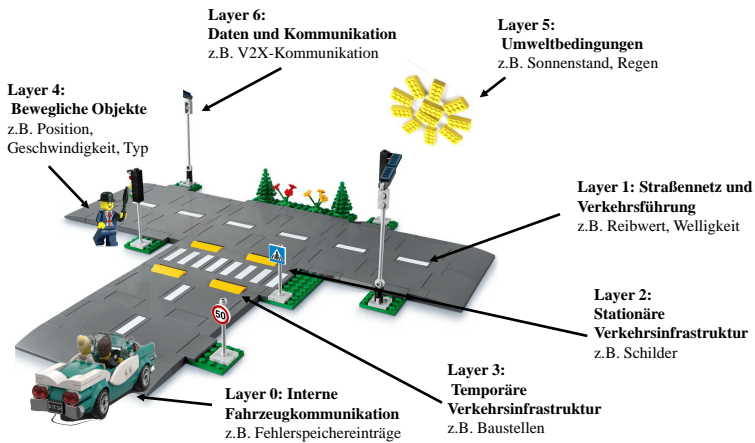


Abbildung 2.11: Pegasus 6+1-Layer-Modell [61] nach [102]

Layer 0: Interne Fahrzeugkommunikation Zum Austausch von funktionalen Nachrichten und Botschaften innerhalb eines Fahrzeuges werden verschiedene Bussysteme verwendet. Jedes Bussystem ist auf die entsprechenden Bedürfnisse in Bezug auf Übertragungsverhalten, Geschwindigkeit, Redundanz, etc. angepasst. Die interne Fahrzeugkommunikation lässt sich in die Unterbereiche *Sense, Plane und Act* unterteilen. Im Layer 0 werden Effekte, wie der Ausfall eines Steuergeräts auf das Gesamtszenario beschrieben [120].

Layer 1: Straßennetz und Verkehrsführung Layer 1 beschreibt das Straßennetz mit allen dazugehörigen statischen Objekten, welche für die Verkehrsführung benötigt werden. Alle Elemente dieses Layers verändern sich nicht innerhalb eines Szenarios [102]. Layer 1 legt die Grenzen fest, in denen sich die Verkehrsteilnehmer bewegen. Er beinhaltet Informationen über die Topologie der Straße, den Aufbau der Straße mit seinen stationären Markierungen, sowie die unterschiedlichen Bereiche im Straßenverkehr (Fußgängerweg, Fahrradweg, Fahrbahn, etc.). Außerdem beschreibt Layer 1 auch stationäre Markierungen auf der Fahrbahn, wie Stoppllinien oder Geschwindigkeitsbegrenzungen. Layer 1 wird mit Hilfe des Standards ASAM OpenDRIVE ⁴ beschrieben. Im Layer 1 beeinflussen Effekte, wie Unebenheiten in der Straßenoberfläche, das Szenario. Diese Effekte dienen als potentielle variable Parameter für die Erstellung von Szenarien [102].

Layer 2: Stationäre Verkehrsinfrastruktur Layer 2 beschreibt stationäre Objekte und Barrieren der Verkehrsinfrastruktur, welche sich während der Laufzeit des Szenarios nicht bewegen bzw. verändern [102]. Dabei liegt der Fokus vor allem auf Objekten neben der Fahrbahn, wie Bäume, Büschen, Hydranten, Straßenlaternen und Zäune. Aber auch auf Objekten, welche die Umgebung der Straße verändern (Brücken, Tunnel, Kreisverkehre). Der Standard ASAM OpenDRIVE

⁴ **OpenDRIVE** ist ein offener Standard für die Beschreibung von Straßennetzen in der Simulation von Automobilsystemen. Es wurde von der OpenDRIVE Association entwickelt und dient als Austauschformat für die Simulation von Fahrzeugen und Verkehrsflüssen in einer virtuellen Umgebung. Mit OpenDRIVE werden Straßennetze in einer einheitlichen und automatisierbaren Weise beschrieben, was die Zusammenarbeit zwischen verschiedenen Simulationstools und -umgebungen erleichtert. OpenDRIVE beschreibt die topographischen und geometrischen Eigenschaften der Straße, sowie die Regeln für Verkehrsteilnehmer und Verkehrssignale [5].

wird dazu verwendet den Pegasus Layer 2 zu beschreiben. Im Layer 2 beeinflussen Effekte, wie Gebäude, welche für Verdeckung sorgen, das Szenario [102].

Layer 3: Temporäre Verkehrsinfrastruktur Layer 3 beschreibt die temporären Änderungen der Verkehrsinfrastruktur, des Straßennetz und der Verkehrsführung (Layer 1 und 2) [102]. Hierfür werden keine neuen Objektklassen eingeführt, sondern lediglich die Eigenschaften und Verhalten der bereits existierenden Klassen verändert und angepasst. Layer 3 bietet die Möglichkeit temporäre Baustellen zu beschreiben, welche die Verkehrsführung und Verkehrszeichen temporär anpassen. Der Standard ASAM OpenDRIVE wird unter anderem dazu verwendet Layer 3 zu beschreiben [102].

Layer 4: Bewegliche Objekte Layer 4 führt bewegliche Objekte ein, welche ihre Position und Eigenschaft über die Zeit verändern. Layer 4 beschreibt den Umgebungsverkehr in dessen Kontext sich das Ego Fahrzeug bewegt. Die beweglichen Objekte werden über Manöver oder Trajektorien vorgegeben, welche sich über die Zeit verändern [102]. Layer 4 wird mit Hilfe des Standards ASAM OpenSCENARIO⁵ beschrieben. Die Parameter, welche sich Layer 4 zuordnen lassen, werden zur Variation der dynamischen Verkehrsteilnehmer in einem Szenario verwendet [102].

Layer 5: Umweltbedingungen Layer 5 stellt Effekte mit Umwelt, Wetter und Beleuchtungseffekten dar. Layer 5 berücksichtigt dabei gleichzeitig den Einfluss von Wettereffekten auf die Sichtweite (Nebel) oder auf den Reibwert der Straße (Glätte). Die Umweltbedingungen lassen sich während des Szenarios auch zeitbasiert ändern [102]. Layer 5 wird mit Hilfe des Standards ASAM OpenSCENARIO beschrieben. Effekte, wie der Sonnenstand beeinflussen das Szenario und lassen sich als Parameter bei der Erstellung von Szenarien variieren [102].

⁵ **OpenSCENARIO** ist ein offener Standard für die Beschreibung von Szenarios in der Simulation von Automobilsystemen. Es wurde von der OpenSCENARIO Association entwickelt und dient als Austauschformat für die Simulation von Fahrzeugen, Verkehrsflüssen und Umgebungen. Mit OpenSCENARIO wird die dynamische Umgebung des Szenarios in einer einheitlichen und automatisierbaren Weise beschrieben, was die Zusammenarbeit zwischen verschiedenen Simulationstools und -umgebungen erleichtert [6].

Layer 6: Daten und Kommunikation Layer 6 fokussiert sich auf jede Art von Informationsaustausch, Kommunikation und Kooperation auf Basis von digitalen Daten. Dabei gehört der Austausch von Informationen zwischen den Fahrzeugen (Vehicle-to-Vehicle Kommunikation (V2V)) und der Austausch von Fahrzeug zur Infrastruktur (Vehicle-to-Infrastruktur (V2X)) dazu [102]. Die Kommunikation erfolgt zum Beispiel direkt über Funk oder über Backend-Dienste. Die Kommunikation mit den GPS-Satelliten ist ein Beispiel für eine Datenkommunikation auf Layer 6, welche dem Fahrzeug zur genauen Positionserfassung dient. Für Layer 6 gibt es zum aktuellen Zeitpunkt noch keinen Standard [102].

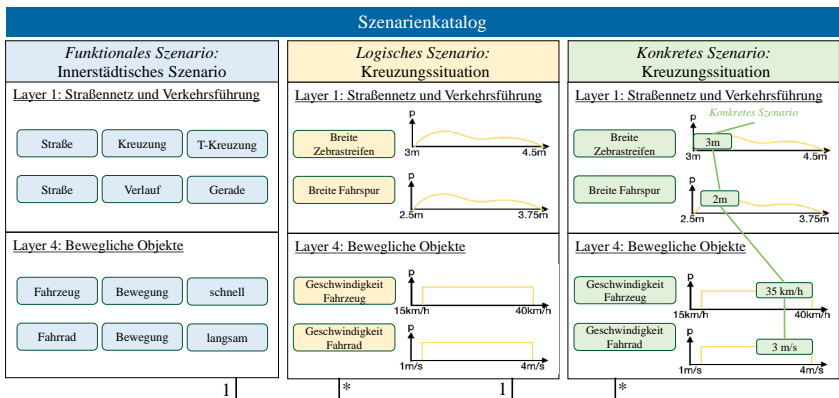


Abbildung 2.12: Zuordnung von Parametern zu den Szenarienabstaktionsklassen nach [69]

Es ist möglich jedem Parameter einene Pegasus Layer zu zuordnen. Ein beispielhaftes *Funktionales Szenario* im Szenarien-katalog (vgl. Abbildung 2.12) wird definiert als Szenario mit zwei beweglichen Objekten (Pegasus Layer 4) und einem groben Straßenaufbau (Pegasus Layer 1). Daraus werden mehrere *Logische Szenarien* abgeleitet. Ein beispielhaftes Szenario wählt zwei Parameterbereiche auf Layer 1 und zwei Parameterbereiche auf Layer 4 aus, welche jeweils eine Eigenschaft des Szenarios variiert. Für das *Logische Szenario* werden dann mehrere *Konkrete Szenarien* abgeleitet, welche eine konkrete Parameterkombination

festsetzen. Aus jedem Parameterbereich wird ein Wert ausgewählt. Das *Konkrete Szenario* dient als Stimulation für einen Testfall.

2.4.2 KPI als Bewertungsmetrik eines Testfalls

Nach der Ausführung eines Testfalls wird dieser Testfall durch verschiedene Metriken - sogenannten Key-Performance Indikatoren (KPIs) - bewertet. Der aktuelle Stand der Technik beschreibt verschiedene Ansätze zur Berechnung von KPIs. Im Rahmen dieser Dissertation wird die Begrifflichkeit KPI - nach ihrer eigentlichen Bedeutung - als Schlüsselmetrik für die Performance eines Szenarios bzw. eines SuT im Kontext einer Stimulation definiert. Prinzipiell sind die Kritikalitätsmetriken eine Untermenge der KPIs. Westhofen, et al. [123] liefert eine Übersicht über verwendbare Kritikalitätsmetriken. Zu den bekanntesten Metriken zählt dabei unter anderem die Time-to-collision, bei welcher nach Formel 2.1 der zeitliche Kollisionsabstand TTC berechnet wird.

$$TTC = \frac{d}{v_{rel}} \quad (2.1)$$

Hierbei stellt d die Entfernung zwischen den beteiligten Objekten und v_{rel} die relative Geschwindigkeit zwischen den beteiligten Objekten dar.

Definition 8. KPI. *Der Key-Performance-Indikator (KPI) ist eine auf Basis von Zeitreihen berechneter Wert, welcher repräsentativ für eine Eigenschaft des Systems steht. Er beschreibt die Performance des Systems im ausgeführten Szenario.*

Durch KPIs werden nicht nur kritische Aspekte (Kritikalitätsmetriken) abgebildet. Zusätzlich zu den Kritikalitätsmetriken werden auch interne Fahrzeugzustände, wie das Auslösen einer Notbremswarnung als KPI für ein Szenario definiert. KPIs nehmen analog zu den Parametern auch binäre, nominale, ordinale, metrische und zeitvariante Werte an (vgl. Kapitel 2.4).

2.4.3 Anwendungsfälle von Szenarien im Automotive Kontext

Szenarien tragen maßgeblich zur Entwicklung und Freigabe von Systemen im Automobilbereich bei. Dazu wird, gemäß [120] und [81], unabhängig vom jeweiligen System ein Szenarienkatalog erstellt (vgl. Abbildung 2.13). Dieser Szenarienkatalog wird entwicklungsbegleitend im V-Modell aufgebaut. Dabei werden die *Funktionalen und Logischen Szenarien* bereits in der Konzeptphase entworfen und im Laufe des Entwicklungsprozesses (vgl. Kapitel 2.2.3) verfeinert und letztendlich zu *Konkreten Szenarien*, welche an einer Testplattform im rechten Schenkel des V-Modells ausgeführt werden.

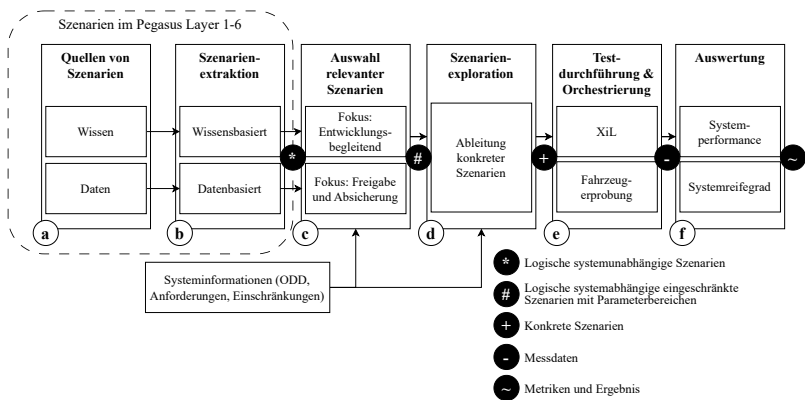


Abbildung 2.13: Anwendungen des Szenarienkatalogs nach [91]

Dieser wird unter anderem datenbasiert auf Basis von Unfalldaten und anderen verfügbaren Datensätzen und Messreihen abgeleitet. Auch die systematische Modellierung von Wissen z. B. durch die Verwendung einer Ontologie nach Mezel et al. [69] oder durch die Formulierung von Einflussfaktoren tragen zur Erstellung eines systemunabhängigen Szenarienkatalogs bei (vgl. Abbildung 2.13 - Teil b). Das Szenarienmodell beinhaltet Parameter der Pegasus Layer 1-6 und berücksichtigt

noch keine fahrzeuginternen bzw. systemabhängigen Parameter, wie z. B. Übergabeszenarien zum Test von Systemredundanzen. In Menzel et al. [69] wird hierfür die Ebene der *Logischen Szenarien* zur Formulierung des systemunabhängigen Szenarienmodells verwendet. Dieses Szenarienmodell dient noch nicht direkt als Stimulation für einen Testfall, da zu dem bereits vorhandenen Szenarienmodell zusätzliches Systemwissen ergänzt werden muss. Durch die Berücksichtigung von Anforderungen und die Ergänzung des Szenarienkatalogs durch Systemwissen lässt sich das erstellte Szenarienmodell entsprechend auf den Systemfokus einschränken und durch weitere systemabhängige Szenarien und Parameter (Pegaus Layer 0) erweitern (vgl. Abbildung 2.13 - Teil c).

Der Szenarienkatalog, der weiterhin auf logischer Ebene formuliert ist, lässt sich je nach Bedarf entweder auf die Entwicklung oder auf die Absicherung fokussieren.

Szenarien im Validierungs- und Freigabeprozess:

Am Ende des Entwicklungsprozesses muss sichergestellt werden, dass das Fahrzeug alle funktionalen und sicherheitsrelevanten Aspekte (ISO 26262, ISO 21448) erfüllt und das Fahrzeug durch die regionale Zulassungsbehörde eine Straßenzulassung nach [118] bekommt.

Definition 9. Validierung. *Die Validierung beschreibt den Prozess mit allen dazugehörigen Dokumenten, Unterlagen, Nachweisen und Tests, welche zeigt, dass sich das System in den spezifizierten Grenzen nach Norm verhält. Hierbei werden durch die Anlehnung an externe Regulatorien, wie der ISO 26262 oder der ISO 21448, Vorgaben für die Absicherung erstellt. Die Absicherung von Fahrzeugen wird in Deutschland von einer unabhängigen Prüf- und Testinstanz durchgeführt. Nach [49]*

Die SOTIF (Safety Of The Intended Functionality) und ISO (International Organization for Standardization) legen dabei Branchenstandards fest, die die Hersteller befolgen müssen, um sicherzustellen, dass ihre Fahrzeuge diese Standards erfüllen. Die SOTIF legt die Anforderungen an die Produktsicherheit in Fahrzeugen fest. Die ISO Normen dienen als Standard für die Qualitätssicherung und Zertifizierung.

Zur Erstellung der Sicherheitsargumentation zur Freigabe eines Fahrzeugs dürfen mehrere Absicherungsansätze verfolgt werden. Die Absicherung von Fahrerassistenzsystemen bis Layer 2 wird auf Basis der anforderungsbasierten Absicherung durchgeführt. Dabei bilden Systemanforderungen die Grundlage der Testfallgenerierung. Hierbei wird zu jeder Anforderung mindestens ein Test gefordert. Das anforderungsbasierte Testen eignet sich besonders zur Identifikation von Risiken. Um Testlücken zu identifizieren und den Reifegrad des Systems zu bestimmen, wird die Testabdeckung als Metrik genutzt. In der Software-Entwicklung werden neben Systemanforderungen auch Äquivalenzklassen, Code-branches oder Bedingungen, als Testabdeckungselemente verwendet, die mit Testdesigntechniken, wie der Anforderungsanalyse, Grenzwertanalyse oder den Fehler-Raten erzeugt werden (siehe [47]). Für die Bewertung des Testablaufs ist a priori Wissen über die genaue Stimulation, System und Testinfrastruktur notwendig. In jedem Testschritt wird auf Signalbasis geprüft, ob eine Testbedingung erfüllt wurde. Die Testfälle werden auf Basis von Systemanforderungen abgeleitet und in Testkatalogen zusammengeführt [101] [117].

Für SAE Stufen mit einem höheren Automatisierungsgrad funktioniert die reine anforderungsbasierte Absicherung nicht mehr [122] [120]. Durch die definierte ODD lassen sich aufgrund des OpenWorld Kontexts nur noch schwierig konkrete Anforderung definieren. Stattdessen findet hier der Szenarien-katalog Anwendung, um das System innerhalb der definierten Grenzen der Operational Design Domain (ODD) abzusichern. Ziel ist es, eine Abdeckung der ODD mit Systemtests zu erreichen. Diese Systemtests bestehen wiederum aus Systembewertungen und einer Stimulation in Form eines Szenarios. Um die Sicherheitsargumentation zu erstellen, müssen durch eine entsprechende Argumentation die repräsentativen Szenarien im Raum der ODD ausgewählt werden (vgl. Abbildung 2.14).

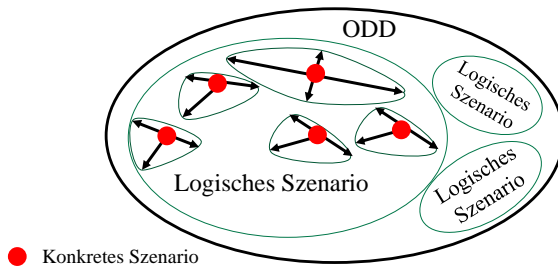


Abbildung 2.14: Abdeckung der ODD mit Szenarien nach [120]

Entwicklungsbegleitender Einsatz:

Die Verwendung von Szenarien im Kontext der Absicherung einer hochautomatisierten Fahrfunktion ist mit Mehraufwand verbunden. Auch im Entwicklungsprozess müssen Szenarien verwendet werden, um den Entwicklungs- und Testaufwand zu reduzieren.

Definition 10. Entwicklungsbegleitend. *Entwicklungsbegleitend beschreibt einen Vorgang, der einen Entwicklungsprozess von Beginn der Produktentwicklung bis zur Serienproduktion begleitet. Nach [67]*

Hierzu wird der systemabhängige Szenarien katalog verwendet, um relevante Stimulationen für das aktuelle entwicklungsbegleitende Testziel festzulegen. Die Entwicklung von automobilen Systemen durchläuft verschiedene Reifegrade (vgl. Kapitel 2.2.4). Jeder neue Reifegrad bzw. jedes entstehende Release muss getestet werden, um so eine Aussage über den aktuellen Reifegrad bspw. der Software zu treffen. Je nach aktuellem Testfokus und Systemreifegrad lassen sich aus dem systemabhängigen Szenarien katalog Stimulationen für Testfälle entnehmen. Hierbei steht nicht die vollständige Abdeckung des Szenarienraums im Fokus, sondern eine relevante Auswahl von Stimulationen, welche das Verhalten des Systems repräsentativ testen. Hierfür ist die zeitliche Verfügbarkeit und Ressourcenbegrenzung aufgrund

der zur Verfügung stehenden Zeit für ein Produktinkrement das begrenzen-
de Element. Ziel des entwicklungsbegleitenden Tests ist die Erkennung und
Analyse von Auffälligkeiten.

Die beiden Fokusse beeinflussen sich gegenseitig, da auch die Erstellung der
Absicherungsargumentation ein entwicklungsbegleitender Prozess ist. Je nach
Reifegrad sind auch während der Entwicklung verschiedene Freigaben für die
Zulassung zum beispielsweise Prüfgelände nötig.

Nach der Auswahl der relevanten Szenarien (vgl. Abbildung 2.13) erfolgt die
Exploration der Szenarien durch eine Überführung der *Logischen Szenarien* in
Konkrete Szenarien (vgl. Abbildung 2.13 - Teil d). Anschließend werden diese
Szenarien einer passenden Testinstanz zugeordnet und mit Hilfe von Bewertungs-
kriterien ausgeführt (vgl. Abbildung 2.13 - Teil e). Anschließend werden die
entstandenen Messdaten ausgewertet (vgl. Abbildung 2.13 - Teil f).

3 Ähnlichkeitsmaße in der Wissenschaft und Technik: Die Bewertung der Aussagekraft von Szenarien

Zum Vergleich von *Konkreten Szenarien* werden in dieser Dissertation verschiedene Vergleichsmetriken benötigt, um eine Aussage über Szenarien und ihre Umgebung (benachbarte Szenarien) zu treffen. In der Wissenschaft existieren hierzu bereits einige Ansätze. Prinzipiell lassen sich diese Ansätze in die Bereiche regelbasierte Szenarienanalyse und Szenarienanalyse mit Hilfe von maschinellem Lernen unterscheiden. Die Vergleichsansätze verwenden nach Braun et al. [24] entweder eine Zeitserie, einen Parametersatz oder KPIs als Vergleichsmaßstab.

3.1 Repräsentativität in der Szenarienanalyse

Die Definition von Repräsentativität ist je nach Anwendungsfall verschieden. In der Softwareentwicklung wird der Begriff Repräsentativität bei der Bildung von Äquivalenzklassen verwendet [29] [85] [21]. Verallgemeinert lässt sich der Begriff Repräsentativität folgendermaßen definieren:

Definition 11. *Repräsentativität.* *Diejenige Menge einer Stichprobe, bei welcher der betrachtete Aspekt das gleiche Verhältnis zur Grundgesamtheit enthält, aus der sie entnommen wurde. Das Repräsentieren bedeutet das Vertreten einer Menge. Nach [72]*

Im Rahmen dieser wissenschaftlichen Ausarbeitung muss die Repräsentativität im Kontext der Szenarienanalyse betrachtet werden. Es wird davon ausgegangen, dass eine endliche Anzahl an *Konkreten Szenarien* gefunden wird, welche den Parameterraum eines *Logischen Szenarios* abbildet und diesen repräsentiert. Der Szenarienraum muss so unterteilt werden, dass wenige Vertreter getestet werden müssen, um dennoch eine Gesamtaussage über das komplette *Logische Szenario* zu treffen.

Zusätzlich ist zu beachten, dass die Repräsentativität im Szenarienkontext, wie sie in dieser Ausarbeitung definiert ist, zunächst unabhängig von der Auswahl der Testinstanz ist. Ob ein simuliertes Ergebnis repräsentativ für eine reale Testfahrt ist, muss durch andere Methoden, wie von Mora et al. [85] und Wachenfeld et al. [122] vorgeschlagenen, überprüft werden. Die Gültigkeit der Testergebnisse einer Testinstanz wird im Rahmen dieser Promotion angenommen. Das heißt, dass die Testinstanz im Kontext des Szenarios und des SuTs gültige und repräsentative Ergebnisse produziert. Die Gültigkeit der Simulationsmodelle und der Umgebungsmodelle muss beispielsweise in Ansätzen nach Steimle et al. [111] sichergestellt werden.

3.2 Deterministische Szenarienanalyse

Die deterministische Szenarienanalyse beschreibt Optionen zum Vergleich von Szenarien, welche auf Basis von vordefinierten Regeln, mathematischen Ansätzen und Vorschriften definiert werden. Die Ähnlichkeit zwischen zwei Szenarien wird dabei über die Distanz definiert. Diese wird mit Hilfe eines Distanzmaßes ermittelt.

3.2.1 Einfache mathematische Distanzmaße

Zu den mathematischen Distanzmaßen zählen der City Block Abstand und die euklidische Distanz. Der mathematische Ansatz lässt sich als Distanzmaß zwischen

zwei Stützstellen verwenden. Die euklidische Distanz stellt die kürzeste Distanz zwischen zwei n -dimensionalen Punkten dar. Sie definiert sich nach Formel 3.1 als die Summe der Abstandskquadrate.

$$(d_{ij}^E) = \sqrt{\sum_{l=1}^n (x_{li} - x_{lj})^2} \quad (i, j = 1, \dots, n) \quad (3.1)$$

Die City-Block-Distanz CB ist die Summe der absoluten Differenzen zwischen Punkten in k Dimensionen. Dieses Distanzmaß ist der einfachste Ansatz und wird für die Berechnung von geometrischen Distanzen zwischen zwei Objekten im multidimensionalen Raum verwendet. [10]

$$d_{ij}^{CB} = \sum_{l=1}^k |x_{li} - x_{lj}| \quad (i, j = 1, \dots, n) \quad (3.2)$$

Anwendung der einfachen mathematischen Distanzansätze im szenarienbasierten Kontext: Die Abstandsmetriken dienen unter anderem dazu, in einer bereits vorhandenen Szenarienlisten von *Konkreten Szenarien*, benachbarte Szenarien zu finden und den Abstand der Parameterwerte zueinander zu berechnen.

3.2.2 Hausdorff-Distanzmaß

Die Hausdorff-Distanz ist ein Maß in der Mathematik, das die Ähnlichkeit zweier Mengen beschreibt. Sie wird verwendet, um die Ähnlichkeit von geometrischen Formen oder Bildern zu messen und ist besonders nützlich in Anwendungen, wie der Mustererkennung und der Computer-Vision. Die Hausdorff-Distanz berechnet den größten Abstand zweier n -dimensionaler Mengen. Je kleiner die Distanz ist, desto ähnlicher sind sich die beiden Mengen. [95].

Es gibt zwei Arten von Hausdorff-Distanzen: die maximale und die minimale Hausdorff-Distanz.

Die maximale Hausdorff-Distanz d zwischen zwei Mengen A und B ist definiert als die größte Distanz zwischen einem Punkt in A und seinem nächsten Nachbarn in B . Die Punkte sind dabei n -dimensional [18]:

$$\begin{aligned}d_{A,B} &= \max\{f_d(A, B), f_d(B, A)\} \\f_d(A, B) &= \max_{a \in A} \min_{b \in B} \|b - a\|, \\f_d(B, A) &= \max_{b \in B} \min_{a \in A} \|b - a\|\end{aligned}\tag{3.3}$$

Die Berechnung der Hausdorff-Distanz erfordert die Verwendung von performanten Algorithmen, um die Distanzen zwischen allen Punkten in den Mengen zu berechnen und dann die maximale oder minimale Distanz zu bestimmen.

Es gibt jedoch auch einige Einschränkungen bei der Verwendung der Hausdorff-Distanz. Zum Beispiel reagiert die Hausdorff-Distanz empfindlich auf Ausreißer und ist daher nicht für alle Situationen als Ähnlichkeitsmetrik geeignet. Auch andere Metriken, wie die Frechet Distance (siehe [33]), werden verwendet, um Trajektorien zu vergleichen.

Anwendung der Hausdorff Distanz im szenarienbasierten Kontext:

In der szenarienbasierten Absicherung von Fahrzeugen wird die Hausdorff-Distanz verwendet, um die Ähnlichkeit von simulierten und realen Szenarien zu bewerten.

Die Hausdorff-Distanz wird verwendet, um die Qualität von simulierten Szenarien im Vergleich zu realen Daten zu bewerten [126]. Durch die Berechnung der Hausdorff-Distanz zwischen den simulierten und realen Szenarien, wird festgestellt, wie gut die Simulationen die realen Ereignisse nachbilden [126]. Atev et al. clustert Fahrzeugtrajektorien, mit Hilfe der Hausdorff Distanz, um neue Trajektorien zu finden [7].

Siebinga et al. [107] clustern den Autobahndatensatz HighD¹ unter Verwendung der Hausdorff-Distanz, indem die Fahrzeugtrajektorien zunächst durch einen mathematischen Ansatz transformiert werden. Dies repräsentiert nach Siebinga et al. [107] den Verkehrskontext besser. Verglichen werden hier immer aufgezeichnete Datensätze, in denen Wegpunkte und Geschwindigkeitsverläufe betrachtet werden [107].

3.2.3 Dynamic Time Warping

Dynamik Time Warping ermöglicht den Vergleich unterschiedlich langer Datenreihen. Es ist ein Maß für die Distanz zwischen zwei Datenreihen und berücksichtigt eventuelle zeitliche Unterschiede in den Messdaten durch die Zuordnung von einem oder mehreren Indizes der anderen Zeitreihe.

Die mathematische Formel für die Berechnung der Dynamic Time Warping (DTW) zwischen zwei Funktionen x und y lautet:

$$DTW(x, y) = \min_{\kappa} \sum_{(i,j) \in \kappa} d(x_i, y_j) \quad (3.4)$$

wobei $d(x_i, y_j)$ die Distanz zwischen den Werten der Funktionen an den Indizes i und j ist und κ eine Abbildung der Indizes von x auf die Indizes von y ist.

Auf diese Weise werden Ähnlichkeiten in Positionssignalverläufen von Personen mit dem DTW-Algorithmus erkannt, selbst wenn eine Person schneller ging als die andere oder wenn es im zeitlichen Verlauf Änderungen der Geschwindigkeit gab [92]. Die DTW wird vor allem bei zeitlichen Sequenzen, wie der Analyse von Video-, Audio- und Grafikdaten angewandt [2] [80]. Prinzipiell benötigt der DTW-Algorithmus zwei lineare (Zeit-)Serien. Eine bekannte Anwendung ist die

¹ Dieser Datensatz wurde mit Hilfe von einer geostationären Drohne über einem Autobahnabschnitt aufgenommen und beinhaltet 147 Stunden an realen Verkehrssituationen im Autobahnkontext.

automatische Spracherkennung, um mit unterschiedlichen Sprechgeschwindigkeiten umzugehen. Weitere Anwendungen sind die Erkennung von Sprachmustern und die eindeutige Erkennung von Unterschriften [2] [80]. Sie wird auch bei Anwendungen zur Formübereinstimmung eingesetzt [80].

Anwendung von Dynamic Time Warping im szenarienbasierten Kontext:

Der DTW-Algorithmus ist unempfindlich gegenüber zeitlichen Verschiebungen. Im Bereich der Analyse von Szenarien werden aufgezeichnete Zeitreihen miteinander verglichen, da sie sowohl unterschiedlich lang sind als auch in ihrer Geschwindigkeit variieren. Trotz dieser Unterschiede ist ein Vergleich zwischen den beiden Zeitreihen möglich. Sakoe et al. [98] nutzen den DTW-Algorithmus direkt, um ein Abstandsmaß für Zeitreihen im szenarienbasierten Umfeld zu definieren. Watanabe [44] verwendet die DTW zum Clustern von Cut-In-Szenarien. Als Signalverlauf für den DTW-Algorithmus wird das Geschwindigkeitssignal des vorausfahrenden Fahrzeugs ausgewählt. Die auf Basis der DTW berechnete Distanz wird als eine Metrik behandelt und mit weiteren KPI-basierten Ähnlichkeiten kombiniert, um das gesamte Szenario zu vergleichen. Ries et al. [92] verwenden DTW, um die Trajektorien mehrerer Verkehrsteilnehmer in Realdaten zu vergleichen. Hierbei müssen zunächst die Metadaten beider Szenarien vorher definierten Übereinstimmungsbedingungen entsprechen, z. B. müssen die gleichen Teilnehmertypen vorhanden sein oder die Szenarien müssen an den gleichen Orten stattfinden. Anschließend wird dann der DTW-Algorithmus auf die Trajektorien angewendet. Unter Verwendung eines definierten Schwellenwerts wird ein Abgleich zwischen verschiedenen Teilnehmenden der Szenarien durchgeführt. Werden beispielsweise zwei Teilnehmende desselben Typs in derselben Situation mit einer hinreichend ähnlichen Trajektorie (d. h. einer Trajektorien-distanz unterhalb eines Schwellenwerts) gefunden, werden diese Situationen als ähnlich angesehen. Hauer et al. [42] verwenden DTW-Algorithmen, um die relative Position zwischen einem Ego und anderen Verkehrsteilnehmern zu vergleichen. Hierfür werden acht Bereiche um das Fahrzeug definiert. In jedem Bereich wird der DTW-Algorithmus auf den Distanzverlauf zu benachbarten Objekten angewandt. Auf die acht sich ergebenden DTW-Distanzen wird eine Min-Max-Normalisierung und eine PCA, um ihre Dimensionalität zu reduzieren, angewandt. Abschließend

werden die Daten mit k-means-Clustering kategorisiert.

3.2.4 Kritikalitätsanalyse

Die Reaktion eines SuT in einer Situation wird auf verschiedene Weise bewertet. Eine Bewertungsmöglichkeit ist die Analyse der Situation in Bezug auf die Kritikalität. Auch einem Szenario lässt sich eine Kritikalität zuschreiben. Diese Kritikalität ist abhängig vom betrachteten System. Für einen Notbremsassistenten sind Situationen, bei denen eine Kollision auftreten könnte oder die Eigensicherheit des Fahrzeugs, sowie andere Verkehrsteilnehmer gefährdet sind, als kritisch einzustufen. In Ansätzen nach Westhofen [123] und Reisgys [36] werden KPIs (vgl. Kapitel 2.4.2) verwendet, um das System in Bezug auf die Kritikalität zu bewerten. Es werden bereits simulierte Daten oder Daten aus öffentlichen Realdatensätzen, wie KITTI² verwendet, um die Kritikalität der Situation im Nachgang zu bewerten. Die Umfrage in Ding et al. [31] zeigt verschiedene Möglichkeiten, einen Szenarienkatalog aus kritischen Szenarien aufzubauen. Rempe et al. [89] analysiert hierfür Unfalldatensätze auf der Basis von KPIs. Hierfür muss ein Grenzwert festgelegt werden. Wird der Grenzwert überschritten, wird die Situation als kritisch eingestuft. Die Einstufung der Kritikalität hängt stark vom zu betrachtenden System ab. Zum Beispiel sind für einen Regensensor andere Situationen kritisch als für einen Notbremsassistenten.

Nach Mora et al. [73] und Abdessalem et al. [4] helfen Entscheidungsbäume (vgl. Abbildung 3.1) kritische Szenarien zu identifizieren und unterstützen bei der Auswahl von kritischen Ausprägungen im multidimensionalen Parameterraum.

² Der **KITTI Datensatz** ist ein Datensatz von Bildern und Videos, der vom Karlsruher Institut für Technologie (KIT) zur Verfügung gestellt wird. Der Datensatz wurde für die Entwicklung und Bewertung von Algorithmen im Bereich der computergestützten Sichtverarbeitung und des hochautomatisierten Fahrens erstellt. Er enthält Daten von verschiedenen Kameras und Sensoren, die an einem fahrenden Fahrzeug angebracht sind. Die Daten wurden unter verschiedenen Bedingungen aufgezeichnet und umfassen Straßenszenen, Städte und Landschaften. Der KITTI Datensatz wird von Wissenschaftlern und Ingenieuren genutzt, um neue Technologien und Algorithmen im Bereich des hochautomatisierten Fahrens zu entwickeln und zu testen.

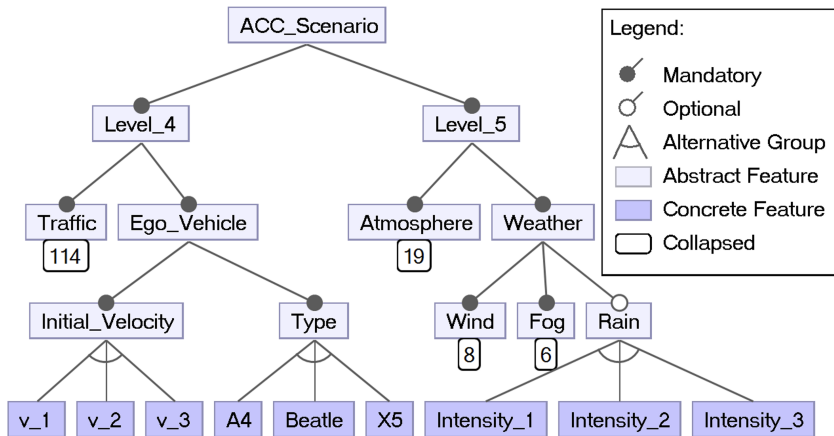


Abbildung 3.1: Beispielentscheidungsbaum für die Szenarienanalyse nach [19]

3.2.5 Dekomposition

In Bezug auf ein mechatronisches System bezieht sich die Dekomposition auf das Zerlegen eines Systems in Teile oder Komponenten, die dann unabhängig voneinander analysiert, entwickelt, getestet und abgesichert werden. Dies erleichtert die Identifizierung und Lösung von Problemen im System, die Fehlerbehebung und die Optimierung der Leistung. Eine Art der Dekomposition in mechatronischen Systemen ist die Trennung von Mechanik, Elektronik und Steuerungstechnologie (vgl. Kapitel 2.2.4) [119]. Dies ermöglicht eine separate Untersuchung und Anpassung jedes Aspekts des Systems. Aus funktionaler Sicht lassen sich die Systeme in einzelne Teilfunktionen dekomponieren.

Anwendung von Dekomposition im szenarienbasierten Kontext:

Das Gesamtsystem Fahrzeug, welches die Funktionalität des hochautomatisierten Fahrens bereitstellt, darf in einzelne Teilsysteme, wie verschiedene Sensoren, zerlegt werden. Jedes Teilsystem wird anschließend für sich separat betrachtet und wird so zum Teil der Sicherheitsargumentation. Nach Frigerio et al. [37] besteht

die Möglichkeit, ein System so zu modularisieren, dass die Schnittstellen der Kamera zur Erkennung verschiedener Fußgängertypen separat getestet werden. In der anschließenden Gesamtfunktionswirkkette wird dann lediglich ein repräsentativer Fußgängertyp geprüft. Diese Dekomposition ist systemabhängig und muss mit entsprechendem Systemwissen vorgenommen werden. Die Sicherheitsargumentation der einzelnen Komponenten wird am Ende in die Gesamtsicherheitsargumentation des Systems eingefügt. Der systemabhängige Teil des Szenarienkatalogs (vgl. Kapitel 2.4.3) führt dabei je nach System zu einer anderen Szenarienauswahl, sodass die Szenarien an einem Subsystem oder an einer einzelnen Komponente, eher ausgeführt werden als im Gesamtsystemverbund.

3.2.6 Adaptive Ausführung zur Ermittlung von Szenarien

Ziel der adaptiven Szenariengenerierung ist eine gezielte Abtastung auf Basis von bereits erfassten Stichproben bzw. ausgeführten Szenarien, um neue Szenarien flexibel an den gewünschten Bereich anzupassen. Hierdurch wird die Auswahl der Szenarien auf die Ergebnisse bereits durchgeführter Testszenarien gestützt [43].

Im Ansatz nach Tatar [114] [115] und Seidel et al. [105] wird die adaptive Szenariengenerierung im Simulationsumfeld eingesetzt. Die Auswahl der neuen Szenarien basiert auf der Kritikalität der bereits ausgeführten Szenarien. Es werden verschiedene Metriken verwendet, welche die Erkenntnisse aus den vorherigen Ausführungsschritten iterativ mit einbeziehen. Nach der Identifikation einer Parameterkombination mit hoher Kritikalität nimmt die Abtastrate im Parameterbereich mit ähnlichen Ausprägungen zu. Für die Entwicklung werden auf diese Weise weitere kritische Szenarien herangezogen und untersucht [114] [115] [105].

Huang et al. [45] verwendet die Kring-Interpolation³ zur simulationsbasierten Wahrscheinlichkeitsvorhersage unbekannter Ereignisse, wie zum Beispiel der Auftrittswahrscheinlichkeit von Spurwechselszenarien mit Geschwindigkeit größer 120km/h.

Evolutionäre Algorithmen helfen Bussler [26], Bühler [25] [77] und Klischat [58] einen automatisierten Prozess zur Generierung von Szenarienkombinationen aufzubauen. Diese Algorithmen bewerten Szenarien auf Basis von Gütekriterien und versuchen den Szenarienraum somit zu optimieren. Kritikalitätsmetriken dienen beispielsweise als Gütekriterien.

3.3 Szenarienanalyse mit Hilfe von maschinellem Lernen

Maschinelles Lernen ist ein Teilgebiet der künstlichen Intelligenz, bei dem es darum geht, Algorithmen das Lernen von Datenmustern, ohne explizite Programmierung zu ermöglichen. Hierbei werden Probleme gelöst, die mit festen Programmen nur mit spezifischen und umfangreichen Algorithmen zu lösen sind. In [71] wird der Begriff des maschinellen Lernens anhand von drei Schlüsselkomponenten definiert: Aufgabe T, Leistungsmaß P und Erfahrung E. „Ein Computerprogramm lernt aus Erfahrung E bei der Lösung von Aufgaben T, wenn sich seine Leistung beim Lösen der Aufgaben, gemessen an P, mit der Erfahrung E verbessert.“ Zu den üblichen Aufgaben des maschinellen Lernens gehören Bildklassifizierung, maschinelle Übersetzung und Erkennung von Anomalien. Um die Fähigkeiten eines maschinellen Lernalgorithmus zu bewerten, ist ein quantitatives Maß für seine Leistung erforderlich. Dieses Leistungsmaß P ist spezifisch für die Aufgabe T, die das System ausführt. Zum Beispiel sind Genauigkeit und Fehlerrate übliche

³ Basierend auf bereits ausgeführten Szenarien und deren Ergebnissen schätzt ein Kriging-Modell die Wahrscheinlichkeit des Eintretens von Ausprägungskombinationen, die in der Nähe der bereits bekannten Ausprägungskombinationen liegenden Szenarien [45].

Leistungsmaße für Klassifizierungsaufgaben. Das Modell lernt mit den gegebenen Daten im Trainingsprozess. Obwohl eine gute Leistung bei den Trainingsdaten wichtig ist, ist dies nicht das endgültige Ziel des maschinellen Lernens. Bei der Bewertung wird auch untersucht, wie gut der Algorithmus für maschinelles Lernen bei Daten abschneidet, die ihm unbekannt sind, da dies ausschlaggebend dafür ist, wie gut er beim Einsatz in der realen Welt funktioniert. In der Praxis ist die Ausbildung eines Modells mit „Underfitting“ und „Overfitting“ verbunden (vgl. Abbildung 3.2). „Underfitting“ bedeutet, dass das Modell keinen ausreichend

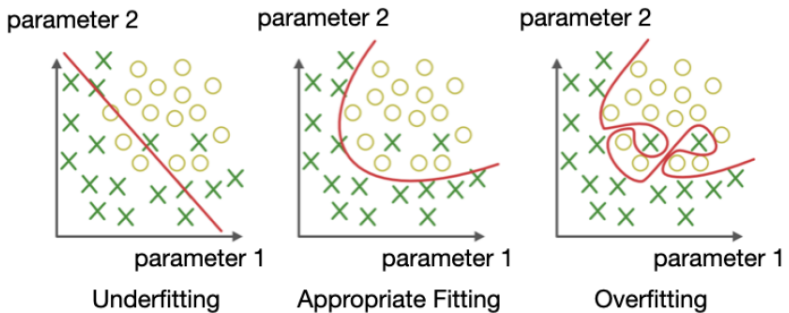


Abbildung 3.2: Underfitting und Overfitting bei der Klassifizierung nach [74]

niedrigen Fehlerwert in der Trainingsmenge erzielt. Lösungen, um das Modell besser an die Daten anzupassen, sind die Erhöhung der Modelltiefe, die Verlängerung der Modelltrainingszeit und das Hinzufügen von Merkmalsdimensionen. „Overfitting“ bedeutet, dass eine zu starke Anpassung der Trainingsdaten an das Modell stattgefunden hat. Die Überanpassung lässt sich durch die Verringerung der Modellkomplexität, die Verwendung von mehr Trainingsdaten und die Entfernung des Rauschens aus den Trainingsdaten reduzieren. Im Idealfall, wenn das Modell die Vorhersagen ohne Fehler trifft, spricht man von einer guten Anpassung an die Daten. Diese Situation ist an einem Punkt zwischen „Underfitting“ und „Overfitting“ zu erreichen.

Man unterscheidet die Algorithmen des maschinellen Lernens in unüberwachte und überwachte Lernalgorithmen. Beim überwachten Lernen wird das Ziel vorab

vorgegeben. Beim unüberwachten Lernen gibt es diese Vorgabe nicht, und der Algorithmus muss lernen, die Daten ohne eine Klassifizierung zu verstehen (vgl. Abbildung 3.3).

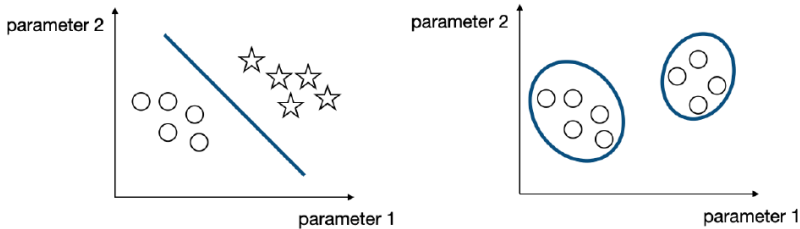


Abbildung 3.3: Unterschied zwischen überwachtem (links) und unüberwachtem (rechts) Lernen

Algorithmen des unüberwachten Lernens erfassen einen Datensatz mit Merkmalen und lernen dann Eigenschaften der Struktur dieses Datensatzes. Beispiele für unüberwachte Lernalgorithmen sind Dimensionsreduzierung (Verringerung der Dimensionen der Daten für die Merkmalsextraktion) und Clustering (Auffinden versteckter Muster in den Daten auf der Grundlage ihrer Ähnlichkeiten oder Unterschiede). Algorithmen des überwachten Lernens versehen einen Datensatz mit Merkmalen, wobei jeder Beispieldatensatz auch mit einem Label oder Ziel verbunden ist. Er lernt mehrere Datenbeispiele eines Zufallsvektors x und eines zugehörigen Werts oder Vektors y und lernt dann, y aus x vorherzusagen, indem er in der $p(y|x)$ schätzt. Beispiele für überwachte Lernalgorithmen sind Klassifizierung (Zuordnung eines Eingabewerts zu einem diskreten Wert) und Regression (Zuordnung eines Eingabewerts zu einem kontinuierlichen Wert).

3.3.1 Clustering

Clustering ist ein Verfahren des unüberwachten Lernens, bei dem ähnliche Datenpunkte in Gruppen (Cluster) zusammengefasst werden. Das Ziel des Clustering ist es, Strukturen in den Daten zu entdecken, die für die weitere Analyse

oder die Entscheidungsfindung von Nutzen sind. Datenpunkte, die unähnlich zueinander sind, werden verschiedenen Clustern zugeordnet. Basierend auf [41] lassen sich die Clusteringansätze in vier Bereiche unterteilen: partitioning Methoden (distance-based), hierarchical Methoden, density-based Methoden, and grid-based Methoden.

Bei k , der Anzahl der zu erstellenden Cluster, erstellt eine Methode ein erstes Clustering. Anschließend wird die Partitionierung durch eine iterative Verschiebetechnik verbessert, indem Objekte von einer Gruppe in eine andere verschoben werden. Basierend auf dem Abstand zwischen den Objekten erfordert ein Clustering, dass Datenpunkte im selben Cluster nahe beieinander liegen und Datenpunkte in verschiedenen Clustern weit voneinander entfernt sind. Eine hierarchische Methode erstellt eine hierarchische Zerlegung der gegebenen Menge von Datenobjekten. Eine hierarchische Methode wird abhängig von der Art der hierarchischen Dekomposition als agglomerativ (von unten nach oben) oder divisiv (von oben nach unten) klassifiziert. Beim agglomerativen Ansatz wird zunächst für jedes Objekt eine eigene Gruppe/Cluster gebildet. Abschließend werden die Objekte oder Gruppen, die einander nahe stehen, sukzessive zusammengeführt. Der divisive Prozess beginnt mit allen Objekten im selben Cluster und teilt ein Cluster nacheinander in kleinere Cluster auf. Bei dichtebasierten Methoden sind Cluster dichte Regionen von Objekten im Raum, die durch Bereiche mit geringer Dichte getrennt sind. Jeder Punkt muss eine Mindestanzahl von Punkten in seiner „Nachbarschaft“ haben. Während Positionierungsmethoden nur kugelförmige Cluster finden, entdecken dichtebasierte Methoden Cluster beliebiger Form. Gitterbasierte Methoden quantisieren den Objektraum in eine endliche Anzahl von Zellen, die eine Gitterstruktur bilden. Die Clustering-Operationen werden auf der Gitterstruktur durchgeführt. Der Hauptvorteil dieses Ansatzes ist seine schnelle Verarbeitungszeit, die unabhängig von der Anzahl der Datenobjekte und nur von der Anzahl der Cluster in jeder Dimension des quantisierten Raums abhängt [41].

Anwendung von Clustering im szenarienbasierten Kontext:

Die Clustering-Analyse wird beispielsweise in der Marktforschung, der Mustererkennung, der Datenanalyse und der Bildverarbeitung eingesetzt. Sie wird auch

zur Analyse von Szenario-Datenbanken eingesetzt, um ein tieferes Verständnis der vorhandenen Daten zu erlangen. Bisherige Clusteringansätze im szenarienbasierten Kontext benutzen ausgeführte *Konkrete Szenarien* aus zum Beispiel Datensätzen, zur Extraktion von *Logischen Szenarien* und Parameterräumen. Es wird dabei versucht, auf Basis von aufgezeichneten Signalverläufen oder aggregierten Metriken *Logische Szenarien* zu extrahieren und in Cluster einzuordnen. Kruber et al. [60] wenden ein Clustering-Verfahren an, um aus Messdaten ähnliche Situationen zu finden und diese zu Clustern zusammenzufassen. Wenn neue Messdaten hinzukommen, werden die Informationen den bestehenden Clustern zugeordnet, wenn sie eine definierte Ähnlichkeit überschreiten. Andernfalls wird ein neues Cluster für die neuesten Messdaten erstellt. Watanabe et al. [44] definieren ein zusammengesetztes Ähnlichkeitsmaß, um den Abstand zwischen Szenariosignalen verschiedener Typen zu quantifizieren und die Cluster-Zentralität zu bestimmen. Auf der Grundlage des Distanz- und Zentralitätsmaßes vergleichen sie den approximativen k-covers-Algorithmus als hierarchischen agglomerativen Clustering-Ansatz mit dem Partitioning Around Medoids von k-medoids als partitionierenden Clustering-Ansatz. Kerber et al. [51] verwenden ein benutzerdefiniertes Distanzmaß, um die Ähnlichkeit zwischen Szenarien zu berechnen, die aus räumlich-zeitlich gefilterten, naturalistischen Straßenverkehrsdaten abgeleitet wurden. Durch die Anwendung von hierarchischem agglomerativem Clustering wurden die Szenarien in Unterräume kategorisiert. Ries et al. [92] nutzen einen Clustering-Algorithmus, um aus realen Daten extrahierte Trajektorien iterativ in semantisch ähnliche Gruppen zu clustern. Eine neue Trajektorie wird entweder zu einem bestehenden Cluster hinzugefügt oder bildet ein neues Cluster auf der Grundlage ihrer Ähnlichkeit mit den Vertretern des Clusters.

Watanabe et al. [44] wenden Clustering auf eine Datenbank mit realen Szenarien an, um relevante und repräsentative Szenarien für die Prüfung auszuwählen. Zur Untersuchung einer bestimmten, in der Entwicklung befindlichen, Fahrfunktion

wurde das Ähnlichkeitsmaß mit einer anwendungsfallbezogenen Gewichtungsfunktion für einen ausgewählten Szenarioparameter oder mit anwendungsfallbezogenen Gewichtungsfaktoren für Szenarioparameter entworfen. Zusammenfassend lässt sich sagen, dass Datensätze aus der realen Welt die am häufigsten verwendeten Quellen für das Clustering von Szenarien im Automobilkontext sind. Die Anwendung von Clustering-Algorithmen zielt darauf ab, einen Katalog zu erstellen, um die Struktur von realen Szenarien besser zu verstehen.

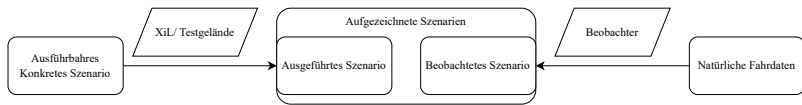


Abbildung 3.4: Verschiedene Arten von Szenarien, je nach Quelle und enthaltenen Informationen. Aufgezeichnete Szenarien bestehen typischerweise aus aufgezeichneten Zeitreihen (z. B. Trajektorien), ausführbare Szenarien enthalten Anweisungen für die Verkehrsteilnehmer (z. B. die auszuführenden Manöver), und naturalistische Fahrdaten enthalten Rohdaten Messungen des Verkehrs nach [24]

Bei den Szenarien im Szenarienkatalog handelt es sich um „ausführbare Szenarien“ (vgl. Abbildung 3.4), welche an einer Testinstanz ausgeführt und in „ausgeführte Szenarien“ im SuT-Kontext mit einer Messung überführt werden. „Aufgezeichnete und beobachtete Szenarien“ im Feld werden dagegen aus „realistischen Fahrdaten“ abgeleitet. Die Szenarien aus öffentlichen Datensätzen sind ohne Kontext eines SuTs [120] [24].

3.3.2 Prädiktions- und Vorhersagemodelle

Ein Ansatz zur Abdeckung des Szenarienraums ist die Monte-Carlo Simulation. Es werden basierend auf einer Verteilung (z. B. Gleichverteilung) Parameter aus dem *Logischen Szenario* zufällig ausgewählt. Nach der Simulationausführung wird überprüft, ob sich das SuT entsprechend den Anforderungen verhält. Dieser Ansatz ist sehr rechenintensiv, da aufgrund der Kombinatorik mehrere hundert bzw. mehrere tausend Szenarien durchlaufen werden müssen, um eine

statistisch signifikante Aussage zu treffen. Anstatt alle Szenarien auszuführen, werden prädiktive Modelle dazu verwendet, einige Simulationsergebnisse durch supervised learning Algorithmen vorherzusagen. Sogenannte Surrogate Models sind Modelle, welche die Funktionsweise eines Systems oder Prozess nachbilden, ohne alle Details und Variablen des Systems zu kennen. Es wird verwendet, um die Leistungsfähigkeit und die Vorhersagegenauigkeit von Black-Box Systemen zu ermitteln, ohne das gesamte System analysieren zu müssen. Surrogate-Modelle sind in verschiedenen Bereichen, wie der Ingenieurwissenschaft, der Mathematik und der künstlichen Intelligenz von Nutzen. Sie tragen dazu bei, die Berechnungszeit und den Ressourcenverbrauch bei der Analyse zu reduzieren. Hierfür werden die verfügbaren ausgeführten Szenarien in einen Trainingsdatensatz und einen Testdatensatz für die Vorhersage unterteilt. Das Modell wird mit Hilfe der verfügbaren Trainingsdaten trainiert und anschließend mit den Testdaten bewertet. Als Eingangsgrößen benötigt das Modell beispielhaft Szenarienmerkmale, auf Basis deren eine Systemperformance vorhergesagt wird [13]. Zur Modellbildung werden verschiedene Algorithmen verwendet. Klassischerweise wird der Gaussian Prozess (GP) (siehe [13]) für die Vorhersage verwendet, weil dieser besonders gut für kleine Trainingsdaten und Unsicherheiten geeignet ist.

Im Folgenden wird der Gaussian Process Regressor [64] [13] [113], der Random Forest Regressor [104] und der Bayesian Ridge [75] [34] näher betrachtet. Der Wertebereich der Unsicherheitsvorhersage ist dabei nicht vergleichbar und muss daher zum direkten Vergleich normiert werden. Die Ergebnisse der Prädiktion und der Unsicherheiten werden an das entsprechende *Konkrete Szenario* angehängt.

Gaussian Process Regressor (GPR)

Der Gaussian Process Regressor ist ein Modell aus dem Bereich des maschinellen Lernens, das insbesondere in der Bayes'schen Optimierung und der nicht-parametrischen Regression verwendet wird. Gauß Prozesse stellen eine Methode zur Modellierung von Unsicherheiten in nicht-linearen Funktionen dar (vgl. Abbildung 3.5) [113].

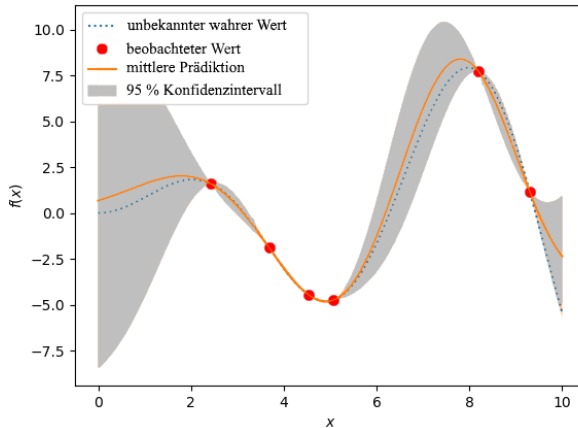


Abbildung 3.5: Unsicherheitsprädiktion beim Gaussian Process Regressor [64]

Für den Einsatz des GPR ist eine umfassende Sammlung von Daten aus bereits durchgeführten Szenarien erforderlich. Diese Daten müssen die Szenarienparameter als Eingangsgrößen sowie die berechneten Metriken als Ausgangsgrößen umfassen. Die gesammelten Daten dienen als Trainingsdaten. Nach dem Training wird beispielsweise das GPR-Modell verwendet, um Metriken für neue, nicht ausgeführte Szenarien vorherzusagen. Im Unterschied zu anderen Regressionsmodellen liefert der Gaussian Process Regressor nicht nur Vorhersagen, sondern berücksichtigt auch Unsicherheiten. Die Unsicherheit beim Gaussian Process Regressor wird in Form der Posterior-Verteilung [13] über die Vorhersagen modelliert. Nach dem Training des Gaussian Process Regressors wird für jede Vorhersage eine Wahrscheinlichkeitsverteilung erstellt. Die Unsicherheit wird durch die Varianz dieser Verteilung repräsentiert.

Die Rückgabe der Standardabweichung σ liefert dabei die Bewertung der Unsicherheit in den Vorhersagen. Dieses nicht-parametrische probabilistische Modell basiert auf der Annahme, dass die unbekannte Funktion, die die Daten generiert, eine Gaußsche Prozessverteilung hat.

Random Forest Regressor

Der Random Forest Regressor repräsentiert ein Ensemble-Lernmodell⁴, das auf einer Sammlung von Entscheidungsbäumen basiert. Diese Ensemble-Methode zeichnet sich durch die Erstellung von mehreren Entscheidungsbäumen während des Trainings aus, wobei die Vorhersagen dieser Bäume kombiniert werden, um robustere und präzisere Ergebnisse zu erzielen. Die Stärken des Random Forest Regressors liegen in seiner Robustheit gegenüber Overfitting und seiner Fähigkeit, mit nicht-linearen Beziehungen in den Daten umzugehen.

Der Ensemble-Ansatz erlaubt dem Random Forest Regressor, verschiedene Interpretationen der Daten zuzulassen, wodurch die Gefahr von Überanpassungen (Overfitting) reduziert wird. Dies trägt dazu bei, dass das Modell auf unterschiedlichen Datensätzen generalisierbar bleibt. Zusätzlich gibt der Random Forest Regressor die Berechnung von Feature-Importance-Werten, die Aufschluss darüber geben, welchen Beitrag einzelne Eingangsvariablen zur Modellvorhersage leisten. Die Feature-Importance-Werte werden durch den Gini-Index oder andere Kriterien (vgl. Han et al. [40]) auf Basis der Anzahl der Entscheidungsbäume berechnet. Die Feature-Importance-Werte dienen als Modell für die Unsicherheit.

Bayesian Ridge

Der Bayesian Ridge ist ein bayesianisches lineares Regressionsmodell, das auf Bayes' Theorem beruht und spezifische Annahmen über die Verteilung der Gewichtungen der Features trifft.

Die Bayesian Ridge-Regression nutzt Bayesianische Methoden, um Unsicherheiten in den Modellparametern zu berücksichtigen.

⁴ Ein **Ensemble-Lernmodell** bezieht sich auf die Idee, mehrere schwache Lernmodelle zu kombinieren, um ein starkes Modell zu erstellen. Der Random Forest Regressor ist eine spezielle Art von Ensemble-Lernmodell, das auf Entscheidungsbäumen basiert.

Anwendung von Prädiktions- und Vorhersagemodellen im szenarienbasierten Kontext:

Bei der Identifikation von kritischen Szenarien aus einer Datenbank, verwendeten Beglerovic et al. [15] ein Surrogate Model mit Hilfe des GP, welches die Leistungsmetriken der betrachteten Szenarien modelliert und vorhersagt. Dieser Prozess wird durch einen adaptiven Algorithmus unterstützt, der den Suchraum iterativ um ein bekanntes Minimum fokussiert. Batch et al. [13] verwendet die GP Klassifikation zur Identifikation von kritischen und unkritischen Situationen in einem Szenarienraum. Hierzu wird die Klassifizierungsunsicherheit dazu verwendet, bisher unbekannte Situationen zu finden. Winkelmann et al. [125] stellt einen iterativen Ansatz zur Präzisierung einer Region of Interest (ROI) in einem Szenarienraum vor. Dabei werden zunächst *Konkrete Szenarien* aus der Region of Interest (ROI) eines *Logischen Szenarios* ausgewählt. Diese Szenarien werden ausgeführt und zum Trainieren eines Metamodells verwendet, das mit einer Erfassungsfunktion kombiniert wird. Das Ergebnis ist eine neue ROI und eine Verteilung *Konkreter Szenarien* über den Eingaberaum. Aus dieser Verteilung wird eine Stichprobe gezogen, wobei die Dichte der *Konkreten Szenarien* dem Wert der Erfassungsfunktion entspricht. Die *Konkreten Szenarien* werden dann ausgeführt und die Ergebnisse dem Datensatz hinzugefügt.

3.3.3 Principal Component Analysis (PCA)

Die Principal Component Analysis (PCA) ist ein Verfahren der multivariaten Statistik, das verwendet wird, um die Struktur von Daten zu erfassen und zu verstehen. Es dient der Dimensionenreduktion und der Visualisierung von hochdimensionalen Daten.

Das Verfahren transformiert die Daten in ein neues Koordinatensystem, in dem es die erste Achse (die erste Hauptkomponente) nach der größtmöglichen Varianz der Daten auswählt (vgl. Abbildung 3.6). Die zweite Achse (die zweite Hauptkomponente) weist die größtmögliche Varianz, die noch orthogonal zur ersten Hauptkomponente steht, usw.

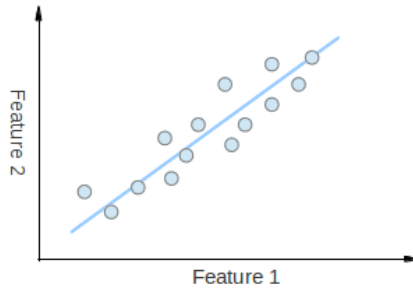


Abbildung 3.6: Hauptkomponente der PCA Analyse nach [78]

Diese neue Darstellung der Daten wird dazu verwendet, um die wichtigsten Merkmale der Daten hervorzuheben und die Unterschiede zwischen verschiedenen Datengruppen zu verdeutlichen.

Die PCA wird auch verwendet, um Daten vor dem Einsatz von Algorithmen wie Klassifizierung oder Clustering zu bereinigen und zu konditionieren, da sie hilft, Rauschen zu entfernen und die Varianz der Klassen zu erhöhen.

Anwendung der PCA im szenarienbasierten Kontext:

Löcklin et al. [66] ermittelten die Hauptkomponenten aus einer synthetisierten Szenariendatenbank und stellten fest, dass einige Aspekte eines Szenarios kritischer sind als andere. Dies erlaubt eine Testgenerierung, die sich auf die wesentlichen Teile konzentriert.

4 Anforderungen, Rahmenbedingungen und Schnittstellen für einen Selektionsprozess für Konkrete Szenarien

Basierend auf dem Stand der Technik, wird ein Selektionsprozess entwickelt. Dieser Selektionsprozess beinhaltet drei Schritte. Jeder Schritt wird über die Anwendung von mehreren verschiedenen Methodenbausteinen (Ähnlichkeits- und Reduktionsmethoden) abgebildet. Ziel des Ansatzes ist eine Selektion von relevanten *Konkreten Szenarien*, welche den gesamten Parameterraum des *Logischen Szenarios* mit einer minimalen Anzahl an Testausführungen, repräsentativ abbildet. An den Selektionsprozess ergeben sich Anforderungen, welche bei der Konzeptionierung des Prozesses und bei der Auswahl der Methoden berücksichtigt werden müssen.

4.1 Anforderungen und Rahmenbedingungen

4.1.1 Einordnung in den Gesamtprozess

Anforderung 1. *Der Selektionsprozess muss Logische Szenarien mit mehreren Parametern mit min/max Werten verarbeiten.*

Anforderung 2. *Der Selektionsprozess muss an einer Testinstanz ausführbare Konkrete Szenarien ausgeben.*

Anforderung 3. *Der Selektionsprozess muss entwicklungsbegleitend entlang des V-Modells anwendbar sein.*

Anforderung 4. *Der Selektionsprozess muss für die Freigabe einer Fahrfunktion geeignet sein.*

Der Testprozess der szenarienbasierten Absicherung von hochautomatisierten Fahrfunktionen mit SAE Stufe 4/5 besteht aus mehreren Stufen (vgl. Abbildung 4.1 - a-f). Der entwickelte Selektionsprozess muss sich in diesen Gesamtprozess einordnen. Er stellt einen Ansatz zur Ableitung und Konkretisierung von *Konkreten Szenarien* aus einem *Logischen Szenario* auf (vgl. Abbildung 4.1 - Szenarienexploration d). Die Entwicklung von Fahrfunktionen im Automobilbereich erfolgt klassischerweise entlang des V-Modells (vgl. Kapitel 2).

Als Ziel des Ansatzes muss eine Repräsentativitätsaussage über ein bisher unbekanntes *Konkretes Szenario* getroffen werden und damit eine Aussage, ob das bisher nicht ausgeführte *Konkrete Szenario* zum Szenarienkatalog hinzugefügt werden muss. Als Endergebnis erhält man einen Szenarienkatalog von *Konkreten*, in ausführbare Testfälle überführbaren, *Szenarien* für jeweils ein *Logisches Szenario* (vgl. Abbildung 4.1). Hierfür wird ein *Logisches Szenario* mit verschiedenen zu variierenden Parametern (z. B. Geschwindigkeit des Ego Fahrzeugs) verwendet. Der Selektionsprozess muss für jedes ermittelte *Logische Szenario* ausgeführt werden. Als Eingangsgröße muss der Ansatz mit einem *Logischen Szenario* mit einem vordefinierten Parameterbereich umgehen. Dieser Parameterbereich muss gemäß Kapitel 2.4 mit Hilfe von Maximal- und Minimalwerten oder einer Variationsvorschrift in Form eines logischen oder mathematischen Ausdrucks gegeben sein. Als Ergebnis muss der Selektionsprozess ein *Konkretes Szenario* mit Parameterwerten, welcher neben dem Bewertungskriterium ein Bestandteil eines Testfalls ist, liefern.

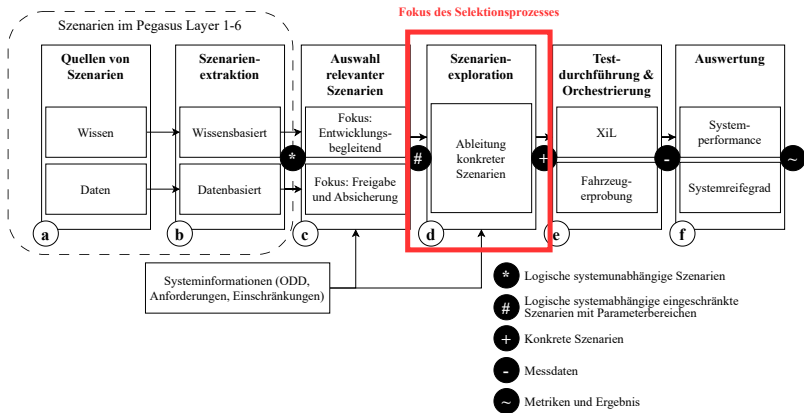


Abbildung 4.1: Einordnung der Methodik in den Gesamtprozess der Erstellung eines Szenarienkatalogs basierend auf Abbildung 2.13

4.1.2 Wiederverwendbarkeit und Übertragbarkeit

Anforderung 5. *Der Selektionsprozess muss aus Methodenbausteinen bestehen, welche auf andere Logische Szenarien übertragbar sind.*

Anforderung 6. *Der Selektionsprozess soll unabhängig von einer Testplattform ausführbar sein.*

Anforderung 7. *Der Selektionsprozess soll aus Methodenbausteinen bestehen, welche auf andere Systeme, Softwarestände und Baureihen übertragbar sind.*

Um nicht für jeden neuen Softwarestand des Systems erneut den kompletten Absicherungsprozess zur Auswahl der *Konkreten Szenarien* durchlaufen zu müssen, muss der Selektionsprozess generalisierbar und soll auf andere Systeme, neue Softwarestände und Baureihen übertragbar sein, sodass der abgeleitete und selektierte Szenarienkatalog auf neue Systemversionen mit Änderungen übertragbar ist. Der Selektionsprozess muss in der Lage sein, zwischen SuT abhängigen und SuT unabhängigen Methoden zu unterscheiden. Gleichzeitig muss das Prinzip des

Selektionsprozesses universell auf alle Arten von *Logischen Szenarien* anwendbar sein.

4.1.3 Ähnlichkeitssaussage

Anforderung 8. *Der Selektionsprozess muss den Parameterraum des Szenarios repräsentativ abtasten und damit die Generalisierungsannahme¹ erfüllen.*

Bei der Selektion der Szenarien müssen *Konkrete Szenarien* ausgewählt werden, welche repräsentativ für andere Szenarien stehen. Dabei lassen sich nicht alle *Konkrete Szenarien* testen. Analog zur Führerscheinprüfung wird der Fahrschüler nur mit einigen ausgewählten Situationen konfrontiert. Anschließend wird davon ausgegangen, dass der Fahrschüler alle Situationen in anderen Kontexten fehlerfrei bewerkstelligt. Es müssen vorab zur Durchführung des Tests ähnliche Szenarien gefunden werden und entsprechend repräsentative Vertreter aus den ähnlichen Bereichen ausgewählt werden, welche repräsentativ für den gesamten Parameterbereich stehen. Ziel ist es, die Generalisierungsannahme für das gesamte *Logische Szenario* mit wenigen repräsentativen *Konkreten Szenarien* zu erfüllen. Gleichzeitig muss der Selektionsprozess eine nachvollziehbare Argumentation für die Auswahl von relevanten Szenarien liefern, um eine Verwendung im Gesamtabsticherungsprozess zu finden.

4.1.4 Reduktion des Testaufwands

Anforderung 9. *Der Selektionsprozess muss den Testaufwand auf Grund der zeitlichen und Ressourcen Einschränkung der Testinfrastruktur auf ein minimales Maß reduzieren.*

¹ Die **Generalisierungsannahme** im Kontext der Absicherung einer hochautomatisierten Fahrfunktion bezieht sich auf die Annahme, dass die Ergebnisse und Erkenntnisse, die durch das Testen in bestimmten Szenarien gewonnen werden, auf eine breitere Palette von Situationen bzw. Szenarien verallgemeinert werden. In diesem Zusammenhang wird angenommen, dass die getesteten Szenarien repräsentativ für eine Vielzahl von Fahrsituationen sind.

Der rein kombinatorische Ansatz, bei welchem alle Parameterkombinationen ausgeführt werden, ist nicht zielführend. Beispielsweise verursacht ein rein kombinatorischer Ansatz in einem Kreuzungsszenario mit angenommenen zehn zu variierenden Parameterbereichen innerhalb des *Logischen Szenarios* mit jeweils 10 Stützstellen pro Parameter bereits 10 Milliarden konkrete Testfälle. Würde man diese Testfälle parallelisiert an 80 Testinstanzen ausführen und würde jede Durchführung nur 1 min². dauern, würde sich aufgrund der Anzahl an Kombinationen immer noch eine beispielhafte Simulationszeit von 230 Jahren ergeben. Auch Pfeffer et al. [88] stellt eine ähnliche Überlegung auf. Diese Simulationszeit beinhaltet dann allerdings nur einen einzigen Stand des SuT, sowie nur ein *Logisches Szenario*. Jeder Testfall, der ausgeführt werden muss, benötigt je nach Testplattform unterschiedlich Zeit. Ziel des Selektionsprozesses ist den Testaufwand auf Grund der zeitlichen und Ressourcen Einschränkung der Testinfrastruktur auf ein minimales Maß zu reduzieren.

Definition 12. Testaufwand. *Der Testaufwand bezieht sich auf die Gesamtheit der Ressourcen, Zeit, Arbeitskraft und Materialien, die für die Planung, Durchführung und Auswertung von Tests benötigt werden. Dies umfasst die Entwicklung von Testfällen, die Bereitstellung von Testumgebungen, die Ausführung von Tests, die Analyse von Ergebnissen, sowie eventuelle Anpassungen und Wiederholungen des Testprozesses.*

Der Testaufwand korreliert mit der Anzahl der durchgeführten Szenarien.

4.1.5 Automatisiert

Anforderung 10. *Der Selektionsprozess soll sich automatisiert ausführen lassen und darf bei erneuter Ausführung keine manuellen Schritte benötigen.*

² Basierend auf Erfahrungswerten angenommene Zeit zur Ausführung eines Testfalls

Zur Analyse und Auswertung von mehreren *Logischen Szenarien* soll die Ausführung der Methoden innerhalb des Selektionsprozesses automatisiert ablaufen, da der manuelle Aufwand für die Anzahl der Szenarien nicht skaliert. Eine Automatisierung bzw. Teilautomatisierung ist gefordert.

4.1.6 Skalierung

Anforderung 11. *Der Selektionsprozess muss sowohl auf Komponentenebene als auch auf Gesamtsystemebene verwendet werden.*

Der Selektionsprozess muss nicht nur für den Test des Gesamtsystems verwendet werden, sondern auch zur Generierung bzw. Selektion von *Konkreten Szenarien* für eine Komponente. Zusätzlich soll der Selektionsprozess auf den Test und die Absicherung von Komponenten und Systemen in der Body- und Komfortdomäne übertragen werden.

4.1.7 Modularität

Anforderung 12. *Der Selektionsprozess muss bestehende Methoden aufgreifen und diese in Form von Methodenbausteinen anwenden.*

Der Selektionsprozess muss aus mehreren Methodenbausteinen bestehen, welche flexibel miteinander verschaltet werden. Hierzu müssen sich weitere Methoden nach dem aktuellen Stand der Wissenschaft und Technik (vgl. Kapitel 3) auf die Problemstellung übertragen lassen, um das Konzept des Selektionsprozesses längerfristig anwendbar zu machen.

4.1.8 Dokumentation und Nachvollziehbarkeit

Anforderung 13. *Der Selektionsprozess muss reproduzierbare und logische Aussagen über die Generalisierungsaussage dokumentieren.*

Anforderung 14. *Der Selektionsprozess darf Konkrete Szenarien nur argumentativ reduzieren.*

Um einen Beitrag zur Absicherung einer automatisierten Fahrfunktion zu leisten, ist es wichtig, den Prozess der Selektion von *Konkreten Szenarien* aus einem *Logischen Szenario* zu dokumentieren. Der Selektionsprozess muss reproduzierbare und logische Aussagen dokumentieren, um diese beispielsweise der zuständigen Freigabebehörde vorzulegen (vgl. ISO26262:6 [49] und SOTIF³ ISO 21448:2022 [48]). Zusätzlich muss die Dokumentation in Haftungsfragen als Referenz herbeigezogen werden. Jedes „ausgeschlossene“ bzw. „reduziert“ Szenario muss mit einer Argumentation nachvollziehbar eingeordnet werden.

4.1.9 Erweiterbarkeit

Anforderung 15. *Der Selektionsprozess muss mit weiteren Ansätzen, in Form von neuen Methodenbausteinen erweiterbar sein, um flexibel auf sich ändernde regulatorische Rahmenbedingungen zu reagieren.*

Aufgrund sich ändernder Randbedingungen und der Entwicklung neuer Methoden und Ansätze, die auf die Problemstellung übertragbar sind, muss der Ansatz erweiterbar gestaltet werden. Neue Methoden sind in Form von neuen Methodenbausteinen in den Gesamtprozess zu integrieren. Der Selektionsprozess und die dazugehörigen Methodenbausteine müssen an die sich ändernden Anforderungen in Bezug auf die Freigabe, durch die entsprechenden Institutionen angepasst und erweitert werden.

³ **SOTIF:** Safety of the intended functionality

4.2 Schnittstellen des Selektionsprozesses zum Szenarienkatalog

4.2.1 Eigenschaften und Bedingungen an ein Logisches Szenario

Als Eingang des Selektionsprozesses wird ein *Logisches Szenario* benötigt, welches aus der statischen Umgebung (vgl. Pegasus Layer 1-3 Kapitel 2.4) und den dynamischen Layern (vgl. Pegasus Layer 4-5 Kapitel 2.4) besteht. Hierbei wird unterschieden zwischen variablen und konstanten Parametern. Konstante Parameter behalten während allen Variationen des *Logischen Szenarios* ihren Wert. Variable Parameter werden im *Logischen Szenario* variiert. Die Variationen bilden die *Konkreten Szenarien*, welche über den Parameterbereich einen Raum (Parameterraum) aufspannen (vgl. Kapitel 2.4.1 - Abbildung 2.10).

Die dynamische Umgebung wird über ein OpenSCENARIO Template und die statische Umgebung über ein OpenDRIVE Template beschrieben. Das Ziel des Methodenbaukastens ist, die variablen Parameter mit konkreten Parameterwerten zu füllen und in ein ausführbares OpenSCENARIO bzw. OpenDRIVE zu überführen (vgl. Kapitel 2.4.1). Das *Logische Szenario* wird nach Langner et al. [63] durch die Variation von verschiedenen Basisszenarien bzw. Manövern erstellt, welche auf Basis von Wissen oder Daten abgeleitet werden. Das *Logische Szenario* definiert verschiedene variable Parameter mit einer Variationsvorschrift (z. B. min/max Werte). Neben den variablen Parametern müssen in Abhängigkeit des SuTs KPIs festgelegt werden, die die Performance des Systems im Szenario beschreiben.

4.2.2 Bestimmung von relevanten Key-Performance-Indikatoren (KPI) für ein Logisches Szenario

Annahme 1. *Ein Szenario wird durch eine endliche Anzahl an KPIs eindeutig beschrieben.*

Schlussfolgerung: Besitzen zwei Konkrete Szenarien die gleichen KPIs sind sie in Bezug auf den zu bewerteten Kontext ähnlich.

Die Auswahl der repräsentativen, für das System relevanten KPIs muss im Kontext des betrachteten *Logischen Szenarios* getroffen werden. Westhofen et al. [124] liefert eine Aufstellung von verschiedenen KPI Arten und vergleicht diese untereinander. Die letztendliche Auswahl der repräsentativen KPIs erfolgt auf Basis von Experten- und Erfahrungswissen. Bewertungskriterien von bestehenden Testfällen sind eine erste Grundlage für die Ermittlung von KPIs sein. Ziel ist es dabei, das Entwicklungssystem mit seinen Systemanforderungen (SUT-Anforderungen) (vgl. Kapitel 2.2.3 - Abbildung 2.3) durch die passende Wahl an KPIs repräsentativ zu beschreiben und anschließend eine Auswahl an KPIs zu treffen, welche im *Logischen Szenario* an der jeweiligen Testinstanz dargestellt bzw. getestet werden müssen. Zur repräsentativen Auswahl der KPIs ist daher das Zusammenspiel aus SuT Anforderung, Testinstanz und *Logischen Szenario* essenziell. Zunächst werden durch die SuT Anforderungen Performancekriterien an das SuT spezifiziert, welche durch KPIs abbildbar sein müssen. Diese Performancekriterien betreffen wiederum verschiedene Pegasus-Layer. Ein Beispiel hierfür ist ein Performancekriterium im Kontext anderer Verkehrsteilnehmer (Pegasus Layer 4), das eine bestimmte minimale Time-to-Collision (TTC) vorschreibt. Alternativ beschreibt die Anforderung ein spezifisches internes Bussignal (Pegasus Layer 0), das in einer bestimmten Situation eine Warnung an den Fahrer ausgeben muss. Unabhängig davon ermöglicht ein *Logisches Szenario* nur eine begrenzte Evaluation an KPIs. Nicht jede KPI Berechnung macht in jedem *Logischen Szenario* Sinn, daher muss

auch hier eine Auswahl an relevanten KPIs im Rahmen des *Logischen Szenarios* getroffen werden. Logischerweise dient die TTC nicht als Bewertungsmetrik, wenn das Szenario nur einen Verkehrsteilnehmer aufweist.

Definition 13. Metrik. Der Term *Metrik* definiert sich als Überbegriff für einen berechneten oder geschätzten Wert, welcher entweder eine Häufigkeit, eine Kritikalität oder ein key-performance Indikator (KPI) ist. Die Metrik eines Konkreten Szenarios ist eindeutig bestimmt.

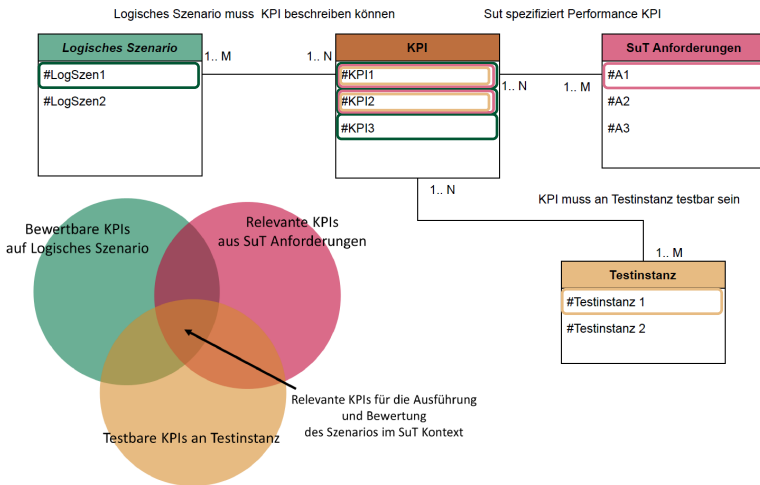


Abbildung 4.2: Auswahl von relevanten KPIs im Kontext des SuT und der Testinstanz für ein *Logisches Szenario*

Der letzte Einflussfaktor auf die Auswahl der repräsentativen KPIs stellt die Testinstanz dar. Die Auswahl der Testinstanz schränkt die Auswahl der betrachteten repräsentativen KPIs zusätzlich ein. Beispielsweise sind bei Real-Erprobungen interne Steuergerätesignale oftmals nicht zugreifbar bzw. messbar. Wittereffekte werden in SiL Umgebungen aufgrund des fehlenden Realismus und von rückwirkenden Einflüssen, wie der Änderung des Reibwerts der Straße in Abhängigkeit des Wetters, nur eingeschränkt nachgebildet. Die KPI muss an der Testinstanz testbar bzw. berechenbar sein (vgl. Abbildung 4.2).

Die Schnittmenge stellt die repräsentative Auswahl an KPIs dar, welche das System mit dem ausgewählten *Logischen Szenario* auf der ausgewählten Testinstanz in den betroffenen Anforderungen testet. Bezogen auf das abstrakte Beispiel in Abbildung 4.2 wird #LogSzen1 mit Hilfe von #KPI1 und #KPI2 an der Testinstanz #Testinstanz1 die SuT Anforderung #A1 getestet. Eine Absicherungsaussage über ein Teilsystem lässt sich nur treffen, wenn alle für die Anforderungen relevanten KPIs abgedeckt wurden. Dies muss nicht im Rahmen von einem *Logischen Szenario* passieren, sondern darf auf mehrere *Logische Szenarien* aufgeteilt werden. Die ermittelten KPIs dienen als Basis für die Selektion der *Konkreten Szenarien* und der Bestimmung der Ähnlichkeiten in den jeweiligen Methodenbausteinen.

5 Methodik zur Selektion von repräsentativen Konkreten Szenarien in einem Logischen Szenario

5.1 Basis des Selektionsprozesses

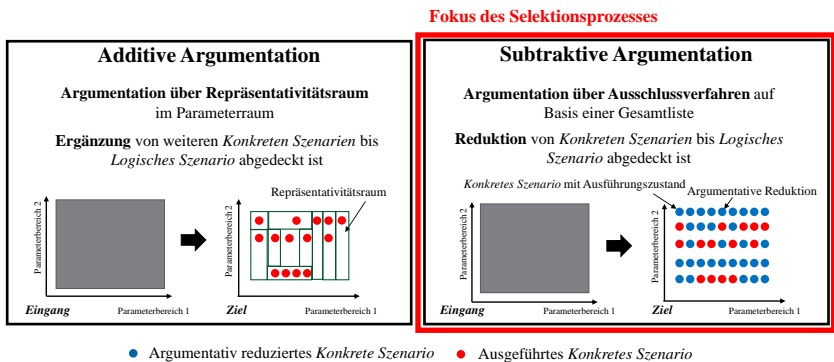


Abbildung 5.1: Argumentative Vorgehensweise bei der Auswahl von repräsentativen Szenarien

Zur Generierung einer Abdeckungsaussage über den Parameterraum und unter Berücksichtigung der genannten Anforderungen (vgl. Kapitel 4) gibt es prinzipiell zwei verschiedene Vorgehensweisen (vgl. Abbildung 5.1). Im Rahmen der additiven Argumentation werden sukzessive *Konkrete Szenarien* dem Parameterraum hinzugefügt, bis eine vollständige Abdeckung des Parameterraums nachgewiesen

wird. Dies erfordert die Bildung von Repräsentativitätsräumen, die das *Logische Szenario* in seiner Gesamtheit abdecken (vgl. Abbildung 5.1 - linker Teil).

Im Gegensatz dazu basiert die subtraktive Argumentation auf einer initialen Gesamtliste von *Konkreten Szenarien*, die noch nicht ausgeführt wurde. Diese Liste wird systematisch analysiert, wobei Szenarien auf Basis argumentativer Überlegungen entweder „argumentativ reduziert“¹ (vgl. blaue Punkte in Abbildung 5.1 - rechter Teil) oder „ausgeführt“ (vgl. rote Punkte in Abbildung 5.1) werden, bis die Liste an *Konkreten Szenarien* vollständig abgearbeitet wurde. Die Argumentation erfolgt auf Basis der Gesamtliste. *Konkrete Szenarien* dürfen nur „argumentative reduziert“ werden (vgl. Anforderung 14).

Der Selektionsprozess, der im Rahmen dieser Promotion vorgestellt wird, folgt dem Ansatz der subtraktiven Argumentation (vgl. Abbildung 5.1 rechter Teil - roter Rahmen), da bei diesem Ansatz jederzeit Transparenz besteht, welche Szenarien noch offen sind und welche bereits analysiert wurden. Hierdurch wird bereits durch die Erstellung der Gesamtliste gewährleistet, dass kein Szenario im Parameterraum übersehen wurde. Im Vergleich dazu beschäftigen sich bestehende Ansätze, wie der von Schütt et al., mit der additiven Argumentation (vgl. Kapitel 3). Der Gesamtprozess der Erstellung (vgl. Anforderung 3) des Szenarienkatalogs lässt sich im V-Modell einordnen (vgl. Abbildung 5.2). Dabei wird der Selektionsprozess sowohl bei der entwicklungsbegleitenden Erstellung von Testfällen auf System- und Komponentenebene als auch bei der Überleitung in die Nutzung bzw. der Freigabe am Ende eingesetzt (vgl. Kapitel 2.4.3).

Aufgrund der Diversität an verschiedenen Parameterarten (vgl. Kapitel 2.4) innerhalb eines *Logischen Szenarios* reicht eine einzelne Methode zur Ermittlung der

¹ **Argumentativ reduziert** bedeutet im Zusammenhang dieser Ausarbeitung eine nachvollziehbare und dokumentierte Argumentation ein *Konkretes Szenario* nicht auszuführen und es in die weiteren Betrachtungen nicht einbeziehen zu müssen, weil es beispielsweise ähnlich zu bereits ausgeführten *Konkreten Szenarien* ist. Diese Argumentation wird zur Erbringung von Sicherheitsnachweisen verwendet.

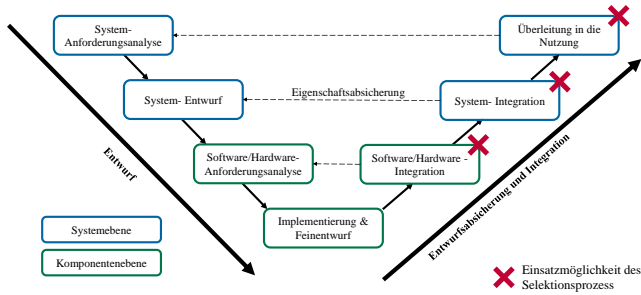


Abbildung 5.2: Verwendung des Selektionsprozesses bei der Entwicklung entlang des V-Modells

Aussagekraft eines *Logischen Szenarios* nicht aus. Als Lösung müssen verschiedene Methoden kombiniert werden. Dies erfolgt in einem Selektionsprozess, der aus einem dreistufigen Vorgehensmodell besteht (vgl. Abbildung 5.3):

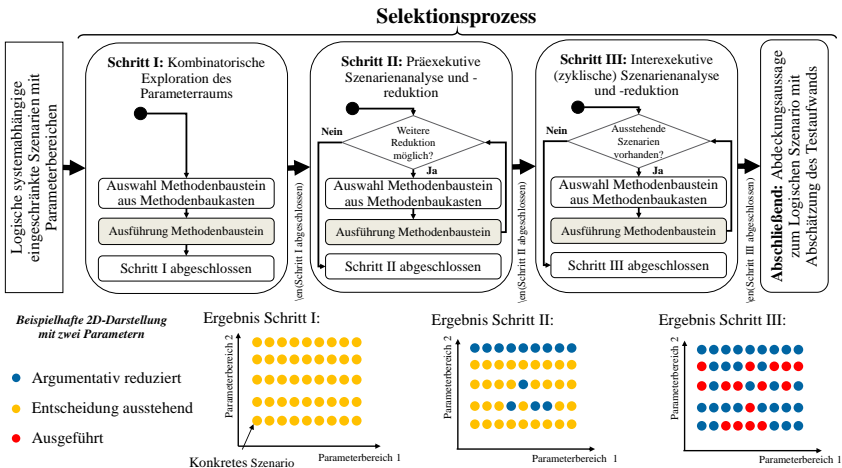


Abbildung 5.3: Schritte des Methodenbaukastens als Ablaufdiagramm inklusive beispielhafte Ergebnisdarstellung im Parameterraum

- **Schritt I:** Kombinatorische Exploration des Parameterraums.

Im ersten Schritt (Schritt I) des Selektionsprozesses werden zunächst alle variablen Parameter des *Logischen Szenarios* mit entsprechenden Vorschriften variiert (vgl. Abschnitt 5.2). Im Schritt I wird daher genau ein sogenannter Methodenbaustein aus dem Methodenbaukasten zur Erstellung der Gesamtliste benötigt (vgl. Abbildung 5.3 - Schritt I).

- **Schritt II:** Präexekutive² Szenarienanalyse und -reduktion.

Anschließend (Schritt II) werden über die Auswahl von Methodenbausteinen aus dem Methodenbaukasten durch präexekutive Analysen *Konkrete Szenarien* „reduziert“. Hierbei wird beispielsweise der Parametersatz durch die Anwendung einer Parameterselektion auf einen reduzierten Satz von Parametern eingeschränkt. Diese Vereinfachungen erfolgen auf der Grundlage physikalischer Zusammenhänge oder Ergebnissen aus Messreihen. Im Schritt II besteht die Option mehrere sogenannte Methodenbausteine aus dem Methodenbaukasten zu kaskadieren, um das Reduktionspotential voll auszuschöpfen (vgl. Abbildung 5.3 - Schritt II). *Konkrete Szenarien*, welche nicht über eine präexekutive Exploration des Parameterraums abgedeckt werden konnten, bleiben im Zustand „Entscheidung ausstehend“ und bilden zusammen mit den „reduzierten“ Szenarien die Eingangsgröße für den nächsten Schritt.

- **Schritt III:** Interexekutive³ (zyklische) Szenarienanalyse und -reduktion.

Im nächsten Schritt (Schritt III) werden nun Ergebnisse von Testausführungen einbezogen. Es lässt sich durch das Testen des Systems in einigen ausgewählten Stützstellen (*Konkreten Szenarien*) eine erste Tendenz über

² **Präexekutiv** steht im Kontext dieser Ausarbeitung für eine Analyse oder Reduktion, die vorab zu einer Ausführung des Szenarios erfolgt. Ob die letztendliche Ausführung simulativ oder im Rahmen einer Realerprobung erfolgt ist dabei irrelevant.

³ **Interexekutiv** steht im Kontext dieser Ausarbeitung für eine Analyse oder Reduktion, die bereits bestehende Ergebnisse einer Ausführung von *Konkreten Szenarien* aufgreift und für die weiteren Analysen verwendet. Ob die letztendliche Ausführung simulativ oder im Rahmen einer Realerprobung erfolgt ist dabei irrelevant.

das Systemverhalten und die Relevanz der konkreten Testausführung im Kontext des Gesamt szenarios ableiten. Durch die iterative Betrachtung von Messergebnissen und der Bewertung einzelner ausgeführter *Konkreten Szenarien* wird zyklisch das Wissen über das *Logische Szenario* verbessert. Dadurch wird versucht, weitere *Konkrete Szenarien* im Parameterraum zu „reduzieren“.

Als Ergebnis liefern alle Schritte eine Liste mit *Konkreten Szenarien* (vgl. Abbildung 5.3). Jedes *Konkrete Szenario* in dieser Liste wird einem von drei Ausführungszuständen eindeutig zugeordnet. Der Ausführungszustand ändert sich während der Ausführung des Selektionsprozess von „Entscheidung ausstehend“ zu „ausgeführt“ oder „argumentativ reduziert“.

Die drei Ausführungszustände werden in den vorgestellten Abbildungen (vgl. Abbildung 5.3) jeweils an einem beispielhaften *Logischen Szenario* mit zwei variablen Parametern (Parameterbereich 1 und 2) und einer exemplarischen Punkte Darstellung visualisiert. Die gelben Punkte stehen dabei für „ausstehende“ *Konkrete Szenarien*, über welche im Rahmen der subtraktiven Argumentation eine Entscheidung getroffen werden muss. Wurde diese Entscheidung getroffen werden diese „ausstehenden“ *Konkreten Szenarien* (gelbe Punkte) entweder „ausgeführt“ (rote Punkte in Abbildungen) oder „argumentativ reduziert“ (blaue Punkte in Abbildungen)

Wurde diese Liste komplett abgearbeitet, wurde die subtraktive Argumentation auf Basis der Gesamtliste durchgeführt (vgl. Kapitel 5.5), um die Generalisierungsannahme (vgl. Anforderung 8) zu erfüllen.

Jedes *Logische Szenario* hat eine bekannte Anzahl an variablen Parametern, welche in den *Konkreten Szenarien* variiert werden.

Am Ende des dritten Schritts steht eine Abdeckungsaussage für ein *Logisches Szenario* mit einer gleichzeitigen Abschätzung des benötigten Testaufwands auf Basis der subtraktiven Argumentation. Alle *Konkreten Szenarien* im Parameterbereich wurden entweder „reduziert“ oder „ausgeführt“.

Der vorgestellte Selektionsprozess ist ein Ansatz, der mehrere Methoden kombiniert und das *Logische Szenario* durch eine subtraktive Argumentation abdeckt. Die drei vorgestellten Schritte benötigen zur Reduktion verschiedene Methodenbausteine. Zur Verwaltung der Methodenbausteine wird ein Methodenbaukasten eingeführt, mit Hilfe dessen verschiedene Methoden unter anderem aus dem aktuellen Stand der Forschung in Abhängigkeit von der Art des Parameters und der Systemarchitektur des zu testenden Systems entwickelt und ausgewählt werden. Die Methodenbausteine erfordern dabei unterschiedliche externe Eingangsgrößen (z. B. externe Daten, Wissen, Erfahrungen, Statistiken) und besitzen die Option eine Testausführung für relevante *Konkrete Szenarien* anzustoßen.

Definition 14. Methodenbaustein. Ein Methodenbaustein im Selektionsprozess stellt eine in sich geschlossene Vorgehensweise dar, deren Eingangsgröße eine Liste mit Konkreten Szenarien innerhalb eines Logischen Szenarios ist. Diese Liste umfasst teilweise bereits „ausgeführte“ oder zuvor „reduzierte“ Konkrete Szenarien, welche in vorhergehenden Methodenbausteinen auf Grundlage von argumentativen Überlegungen „reduziert“ wurden.

Die Anzahl der *Konkreten Szenarien* im Ausführungszustand „Entscheidung ausstehend“ nach der Ausführung des Methodenbausteins muss kleiner sein als davor:

$$|S_{\text{ausstehend, nach}}| < |S_{\text{ausstehend, vor}}| \quad (5.1)$$

Dabei bezeichnet $S_{\text{ausstehend, nach}}$ die Menge der *Konkreten Szenarien* mit dem Zustand „Entscheidung ausstehend“ nach der Ausführung des Methodenbausteins und $S_{\text{ausstehend, vor}}$ die Menge der Szenarien vor der Ausführung. Nicht jedes *Konkrete Szenario* muss demnach durch die Ausführung eines Methodenbausteins seinen Ausführungsstatus verändern. *Konkrete Szenarien*, welche ihren Ausführungsstatus nicht verändern, werden als Don't Care Szenarien (vgl. Abbildung 5.4 - graue Punkte) bezeichnet, da sie durch den Methodenbaustein weder „reduziert“ noch „ausgeführt“ wurden. Jeder Methodenbaustein lässt sich wiederum zu besserer Strukturierung in mehrere kleine Subschritte (0..k) unterteilen.

Jeder dieser Methodenbausteine kann durch weitere Methodenbausteine ergänzt (Kaskadierung) bzw. ersetzt (Austauschbarkeit) werden (vgl. Anforderung 12). Ein Methodenbaustein lässt sich in einem Selektionsprozess mehrmals anwenden. Jeder Methodenbaustein grenzt sich gegenüber seiner Umgebung ab (vgl. Abbildung 5.4).

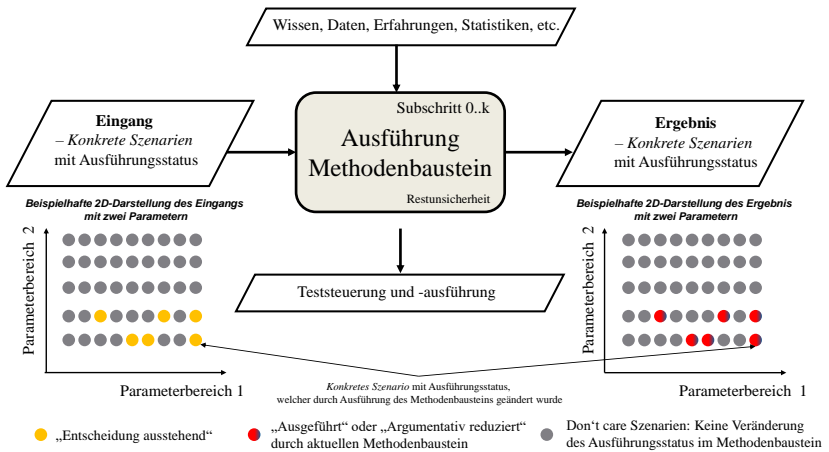


Abbildung 5.4: Schematischer Aufbau eines Methodenbausteins mit Schnittstellen und beispielhafte Entscheidung über Ausführungszustand im Fall mit zwei variablen Parametern

Jeder Methodenbaustein ist mit einer Restunsicherheit behaftet, die in Abhängigkeit der getroffenen Annahmen, der Genauigkeit der Vorhersage und den entsprechenden Rahmenbedingungen getroffen werden muss.

Dabei stellt der einzelne Methodenbaustein Minimalanforderungen an die Testinstanz zur Güte der Simulation. Beispielsweise müssen bei einem Methodenbaustein, welcher auf Basis von KPIs eine Repräsentativität bestimmt, auch sichergestellt werden, dass die entsprechende Testinstanz den benötigten KPI überhaupt mit einer ausreichenden Güte bestimmt. Die Auswahl der Methodenbausteine schränkt somit die Anzahl der Testinstanzen ein. Sofern die Anforderungen an den Selektionsprozess erfüllt sind, ist die im Rahmen der Testorchestrierung gewählte

Testinstanz unabhängig vom Selektionsprozess. Somit werden sowohl simulationsgestützte Testinstanzen, als auch reale Erprobungsfahrten zur Generierung der Testergebnisse verwendet (vgl. Anforderung 6).

Unabhängig von der angewandten Methode wird neben dem Auftrag zur „Ausführung“ oder „Reduktion“ eines *Konkreten Szenarios* gleichzeitig noch eine Argumentation für die getroffene Entscheidung abgespeichert. Dies sorgt für eine spätere Nachvollziehbarkeit und liefert gleichzeitig ein wichtiges Fundament für die Abdeckungsaussage (vgl. Anforderung 4 und Anforderung 13). Gleichzeitig werden zur Vereinfachung der Szenarienliste in einem Methodenbaustein Annahmen (vgl. Abbildung 5.5 - Entität Annahme) getroffen, welche am Ende ebenfalls in die Gesamtaussage einbezogen werden müssen. Die Ermittlung der Annahmen ist abhängig vom Methodenbaustein und muss beim Design des Methodenbausteins dokumentiert werden.

Das Konzept eignet sich zur Übertragung auf weitere *Logische Szenarien* (vgl. Anforderung 5). Gegebenenfalls ändert sich die Reihenfolge und/oder Auswahl der einzelnen Methodenbausteine in Abhängigkeit des *Logischen Szenarios*. Der Selektionsprozess im Kontext der *Logischen Szenarien* und der Methodenbausteine wird als Entitätsdiagramm-Diagramm in Abbildung 5.5 dargestellt.

Die Abbildung des Selektionsprozesses als dreistufiges Vorgehensmodell beschreibt die methodische Gliederung in die Phasen Exploration, präexekutive Reduktion und interexekutive Analyse (vgl. Abbildung 5.3).

Das ergänzend dargestellte Entitätsdiagramm veranschaulicht die strukturellen Beziehungen zwischen den relevanten Entitäten des Selektionsprozesses, insbesondere im Hinblick auf die Verknüpfung von Methodenbausteinen, Parametern, Szenarien und KPIs (vgl. Abbildung 5.5).

Beide Abbildungen beschreiben denselben Selektionsprozess aus unterschiedlichen Perspektiven: Während das Vorgehensmodell den methodischen Ablauf strukturiert, zeigt das Entitätsdiagramm die modulare Zusammensetzung und die zugrunde liegenden Informationsflüsse. Durch die parallele Betrachtung lassen sich sowohl prozedurale als auch strukturelle Aspekte nachvollziehbar darstellen.

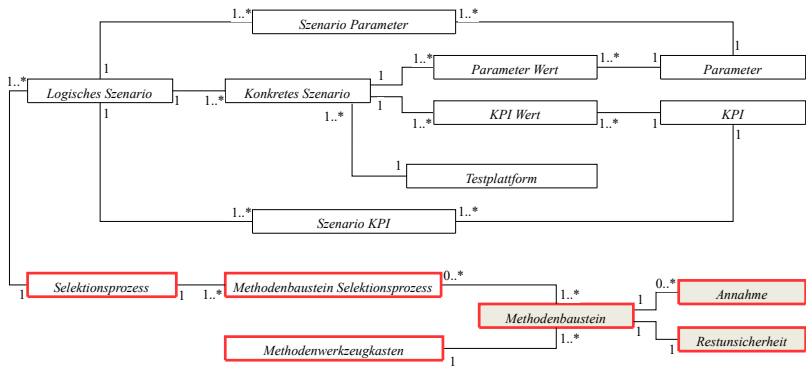


Abbildung 5.5: Entitätsdiagramm zur Szenarienselektion im Kontext der Methodenbausteine und des *Logischen Szenarios* mit dem in rot umrandeten Konzept des Selektionsprozesses zur Strukturierung. Der Selektionsprozess greift das in Abbildung 5.3 dargestellte Vorgehensmodell auf.

Die genaue Ausprägung des Selektionsprozesses variiert je nach *Logischem Szenario*. Jeder Selektionsprozess besitzt eine Auswahl und Reihenfolge verschiedener Methodenbausteine (vgl. Entität Methodenbaustein Selektionsprozess in Abbildung 5.5). Diese Struktur erlaubt die Wiederverwendbarkeit der Methodenbausteine (vgl. Anforderung 11). Durch die Zuordnung des Methodenbausteins zu einer Instanz des Selektionsprozesses mit einer Ausführungsreihenfolge und den konkreten, in Abhängigkeit vom Szenario, relevanten KPIs wird die genaue Reihenfolge der Methodenbausteine bestimmt. Methodenbausteine dürfen auch mehrfach mit unterschiedlichen Parametrisierungen verwendet werden (vgl. Anforderung 5 und Anforderung 7). Jedes *Logische Szenario* hat seinen eigenen Selektionsprozess (vgl. Abbildung 5.5 - Entität Selektionsprozess) der auf die *Konkreten Szenarien* angewandt wird. Die Methodenbausteine bilden die Basis des Selektionsprozesses und werden in einem Methodenbaukasten zusammengefasst (vgl. Abbildung 5.5). Die Entitäten Selektionsprozess, Methodenbaustein, Annahmen, Restunsicherheit und Methodenbaukasten bilden die Basis des Konzepts (vgl. Abbildung 5.5 - roter Rahmen).

Beschreibung von Szenarien mit Parameterkombinationen

Zur einheitlichen Beschreibung der Szenarien wurde für die aufgestellten Anforderungen ein Entitätsdiagramm entwickelt (vgl. Abbildung 5.5). Das Datenmodell beinhaltet das *Konkrete Szenario* als Kernelement. Jedes *Konkrete Szenario* gehört zu einem *Logischen Szenario* und zu einer optionalen Testplattform, auf welcher das Szenario ausgeführt wird, sofern ein Methodenbaustein über den Ausführungszustand den Auftrag zur Ausführung stellt. KPI Berechnungen (vgl. Abbildung 5.5 - Entität KPI Wert) werden entweder durch eine Testausführung auf Basis der Messergebnisse berechnet oder über einen entsprechenden Methodenbaustein prädiert. Die Parameterwerte werden durch eine Generierungsmethode (vgl. Schritt I des Selektionsprozesses) aus dem Methodenbaukasten initial angelegt und dann durch Reduktionsmethoden (vgl. Schritt II und Schritt III des Selektionsprozesses) weiter analysiert. Ein *Konkretes Szenario* besitzt dabei mehrere KPI-Werte und mehrere Parameterwerte unterschiedlicher Typen (vgl. Kapitel 2.4.1).

Das Ergebnis der Ausführung der Restriktions- und Reduktionsmethoden (vgl. Schritt II und Schritt III des Selektionsprozesses) ist jeweils die aktualisierte Liste an *Konkreten Szenarien*, in welcher der Zustand der Szenarienausführung entweder auf „Ausführung benötigt“ oder „reduziert“ gesetzt wurde.

Ermittlung der Restunsicherheit eines Methodenbausteins

Die Restunsicherheit eines Methodenbausteins (vgl. Abbildung 5.5 - Entität Restunsicherheit) muss durch eine quantitative Metrik (z. B. Risikoprioritätszahl) beschrieben werden, die dabei hilft, die Abweichung, Unsicherheit und Einflussfaktoren zu systematisieren (vgl. Abbildung 5.6). Die folgenden Unsicherheitskategorien dienen zur Berechnung und Festlegung eines Unsicherheitswertes:

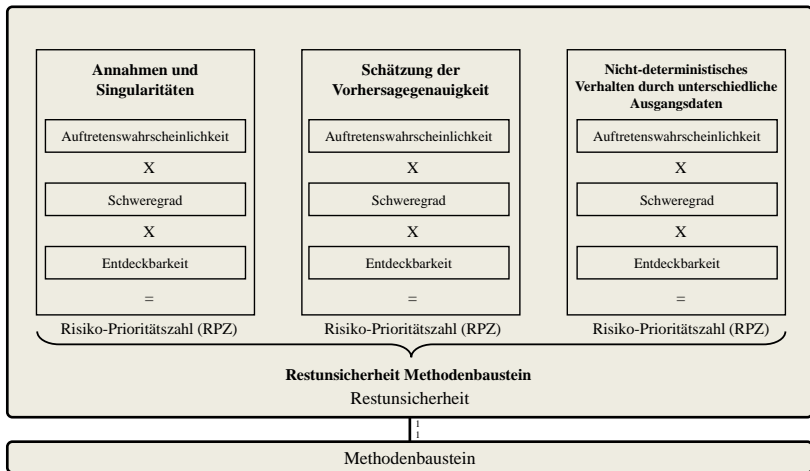


Abbildung 5.6: Ermittlung der Riskoprioritätszahl für jeden Methodenbaustein

- **Annahmen und Singularitäten:**

- Annahmen: Innerhalb des Methodenbausteins werden Annahmen zu bspw. logischen oder physikalischen Zusammenhängen getroffen. Eine Fehlerabschätzung hilft diese Annahmen einzuordnen und ein Unsicherheitsmaß festzulegen.
- Singularitäten: Singularitäten, die auf Basis der gewählten Diskretisierungsschrittweite auftreten, haben Einfluss auf die Restunsicherheit. Diese wird auf Basis von Fehlerabschätzungen oder Abweichungen zwischen realen und idealisierten Bedingungen quantifiziert. Eine hohe Anzahl ignorierte Singularitäten erhöht die Restunsicherheit.

- **Schätzung der Vorhersagegenauigkeit**

- Genauigkeitsmetriken: Wenn der Methodenbaustein eine Vorhersage erzeugt (z. B. durch eine Regression oder ein Prädiktionsmodell für ein KPI), lässt sich die Unsicherheit über die Standardabweichung oder das Vertrauensintervall des Modells abschätzen.

- Vergleich mit Daten: Falls historische Daten vorliegen, erfolgt eine Abweichungsmessung, beispielsweise anhand der mittleren quadratischen Abweichung (MSE) oder der mittleren absoluten Abweichung (MAE), zur Bestimmung der Modellgenauigkeit und Unsicherheit. Das Ergebnis dieser Messung quantifiziert die Unsicherheit des Vorhersagemodells.
- **Nicht-deterministisches Verhalten durch unterschiedliche Ausgangsdaten**
 - Ähnlichkeitsmetrik: Ein Abstandswert zwischen den Ergebnissen verschiedener Ausgangsdaten könnte als Unsicherheitsmaß dienen, etwa die Standardabweichung der Cluster-Ergebnisse. Für Clustering-Methoden, wie K-Means oder DBSCAN könnte die Unsicherheit als Varianz oder Abstand innerhalb der Cluster definiert werden.
 - Varianz über Wiederholungen: Mehrfache Ausführung des Methodenbausteins mit veränderten Ausgangsdaten und anschließender Berechnung der Varianz der Ergebnisse, welche zur Quantifizierung der Restunsicherheit benutzt wird.

Die Restunsicherheit muss für jede Unsicherheitskategorie eines Methodenbausteins bestimmt werden und muss anschließend zu einer Gesamtrestunsicherheit zusammengefasst werden. Als Basis für die Unsicherheitsbewertung dienen die FMEA⁴-typischen Kategorien **Auftretenswahrscheinlichkeit**, **Bedeutung (Schweregrad)** und **Entdeckbarkeit**. Dies erlaubt eine systematische Evaluierung der Unsicherheit und deren Einfluss auf den Methodenbaustein. Die einzelnen Unsicherheitsfaktoren und Annahmen werden dann strukturiert bewertet, was

⁴ Die FMEA (Fehlermöglichkeits- und -einflussanalyse) ist eine strukturierte Methode zur Identifizierung und Analyse potenzieller Fehler in Produkten, Prozessen oder Systemen sowie zur Bewertung ihrer Auswirkungen. Sie dient dazu, Risiken systematisch zu erfassen und priorisieren, um anschließend Maßnahmen zur Risikominderung zu planen und damit die Zuverlässigkeit und Sicherheit zu verbessern.

schließlich zu einer Risiko-Prioritätszahl (RPZ) führt, welche die Restunsicherheit des Methodenbausteins quantitativ beschreibt. Diese berechnet sich aus drei Faktoren (vgl. Abbildung 5.6):

Auftretenswahrscheinlichkeit:

Mit Hilfe der Auftretenswahrscheinlichkeit wird bewertet, wie wahrscheinlich es ist, dass eine Annahme oder Unsicherheit zu einem Problem führt.

- **Bewertungsskala:** Eine Skala von 1 bis 10 (z. B. 1 = sehr unwahrscheinlich, 10 = sehr wahrscheinlich) wird verwendet, um die Wahrscheinlichkeit des Auftretens eines bestimmten Unsicherheitsfaktors einzuschätzen.
- **Methoden zur Bestimmung:** Empirische Analysen, historische Daten oder eine Monte-Carlo-Simulation, falls vorhanden, helfen, die Wahrscheinlichkeit zu quantifizieren. Bei Methodenbausteinen mit nicht-deterministischem Verhalten wird die Anzahl der Fälle, in denen das Verhalten außerhalb eines akzeptablen Bereichs liegt, als Maß für die Auftretenswahrscheinlichkeit genutzt.

Schweregrad (Bedeutung):

Mit Hilfe des Schweregrads wird eingeschätzt, wie gravierend die Konsequenzen der Unsicherheit sind, wenn sie sich tatsächlich auf das System auswirkt.

- **Bewertungsskala:** Auch hier wird eine Skala von 1 bis 10 verwendet (z. B. 1 = sehr geringe Bedeutung, 10 = kritisch).
- **Bestimmung des Schweregrads:** Der Schweregrad wird durch die Auswirkung auf die Systemperformance, die Zuverlässigkeit oder die Sicherheit bewertet. Eine hohe Abweichung führt durch eine Annahme oder ein Modellfehler, der kritische Systemparameter beeinflusst, führt zu einem hohen Schweregrad.

- **Beispiel:** Wenn die Annahme einer Vereinfachung bei einer Singularität zu einem Versagen des Systems führt, hat dies einen hohen Schweregrad zur Folge.

Entdeckbarkeit:

Diese Kategorie bewertet, wie wahrscheinlich es ist, dass die Unsicherheit oder ein daraus resultierender Fehler entdeckt wird, bevor sie zu einem Problem führt.

- **Bestimmung der Entdeckbarkeit:** Die Entdeckbarkeit hängt von den eingesetzten Überwachungssystemen, Qualitätskontrollen und Sensitivitätsanalysen ab, die darauf ausgelegt sind, Unsicherheiten frühzeitig zu identifizieren. Unsicherheiten, die durch Tests oder Simulationen zuverlässig erkannt werden, erhalten einen niedrigen Wert, während schwer oder gar nicht im Voraus identifizierbare Unsicherheiten mit einem hohen Wert bewertet werden.
- **Bewertungsskala:** Eine Skala von 1 bis 10 wird hier ebenfalls eingesetzt (1 = sehr einfach zu entdecken, 10 = schwer zu entdecken).
- **Beispiel:** Eine unsichere Vorhersage, die nur unter sehr speziellen Bedingungen auffällt, führt zu einer schwierigen Entdeckbarkeit.

Berechnung der Risiko-Prioritätszahl (RPZ)

Die RPZ wird berechnet, indem die Werte für Auftretenswahrscheinlichkeit, Schweregrad und Entdeckbarkeit multipliziert werden:

$$\text{RPZ} = \text{Auftretenswahrscheinlichkeit} \cdot \text{Schweregrad} \cdot \text{Entdeckbarkeit} \quad (5.2)$$

Eine hohe RPZ (Maximalwert bei 1000) deutet auf eine hohe Unsicherheit hin. Beispielsweise wird auf diese Weise die Unsicherheit für die Bewertungskategorie Nicht-deterministisches Verhalten durch unterschiedliche Ausgangsdaten ermittelt.

Beispiel: Ein Methodenbaustein wird zur Analyse der Notbremsfunktion eingesetzt. Um den Testaufwand zu reduzieren, wurde der Sonnenstand aus dem Parameterraum ausgeschlossen. Das bedeutet, dass Tests nicht berücksichtigen, wie sich unterschiedliche Lichtverhältnisse (z. B. tiefstehende Sonne, blendendes Licht) auf die Sensordaten der Notbremsfunktion auswirken. Obwohl die Notbremsfunktionen Radar und Kamerasensoren kombiniert, kann die Vernachlässigung des Sonnenstandes dazu führen, dass kritische Szenarien unzureichend getestet werden.

- Auftretenswahrscheinlichkeit: 8 (sehr häufiges Auftreten) Die Sonne steht oft tief (z. B. morgens, abends) und beeinflusst die Kamerasensoren stark.
- Schweregrad: 9 (hohes Unfallrisiko) Falls die Notbremsfunktion Hindernisse nicht erkennt oder falsch interpretiert, kann dies zu schweren Unfällen führen.
- Entdeckbarkeit: 5 (schwierige Entdeckbarkeit) Die Einschränkung des Parameterraums fällt erst auf, wenn reale Tests oder Unfälle Probleme aufzeigen. Allerdings liegen Daten und Messergebnisse vor, dass die Notbremsfunktion auch bei Ausfall der Kamera durch Blendung noch eine Bremsung durchführt.

Die Risikoprioritätszahl für diese Bewertungskategorie ergibt sich beispielsweise zu:

$$\text{RPZ} = \underbrace{7}_{\text{Auftretenswahrscheinlichkeit}} \cdot \underbrace{6}_{\text{Schweregrad}} \cdot \underbrace{4}_{\text{Entdeckbarkeit}} = \underbrace{288}_{\text{RPZ für eine Bewertungskategorie}} \quad (5.3)$$

Zusammenfassende quantitative Bewertung der Metrik

Durch die Kombination der RPZ für die jeweiligen Unsicherheitskategorien ergibt sich eine gewichtete Summe durch das Wurzelquadrat der addierten RPZ. Es

entsteht eine Gesamtunsicherheit für den betrachteten Methodenbaustein (vgl. Formel 5.4).

$$\text{Gesamt Restunsicherheit Methodenbaustein} = \sqrt{\sum_{i=1}^n w_i \cdot (\text{RPZ}_i)^2} \quad (5.4)$$

Hierbei sind w_i die Gewichtungen der einzelnen Unsicherheitsfaktoren und RPZ_i die RPZ der drei Unsicherheitskategorien. Idealisiert wird $w_i = 1$ angenommen. Abhängig vom Fokus der Unsicherheitsbewertung darf dieser Wert angepasst werden.

Die Unsicherheitskategorien in Kombination mit der Bewertung nach Schweregrad, Entdeckbarkeit und Auftretenswahrscheinlichkeit geben eine systematische und quantitative Herangehensweise, um die Gesamt-Restunsicherheit des Methodenbausteins durch einen konkreten Zahlenwert zu beschreiben.

5.2 Schritt I: Kombinatorische Exploration des Parameterraums

Ziel des Schritts: Das Ziel des ersten Schritts des Selektionsprozesses ist die kombinatorische Auflistung einer vollständigen⁵ Liste an *Konkreten Szenarien*, welche in den nächsten Schritten „ausgeführt“ oder „reduziert“ werden. Diese *Konkreten Szenarien* basieren dabei alle auf dem gleichen *Logischen Szenario*. Ziel ist es dabei, die Gesamtliste der subtraktiven Argumentation mit dem Anspruch auf Vollständigkeit aufzustellen, sodass man später davon ausgeht, dass für die Abdeckungsaussage jedes *Konkrete Szenario* betrachtet wurde (vgl. Anforderung 8 und Abbildung 5.7).

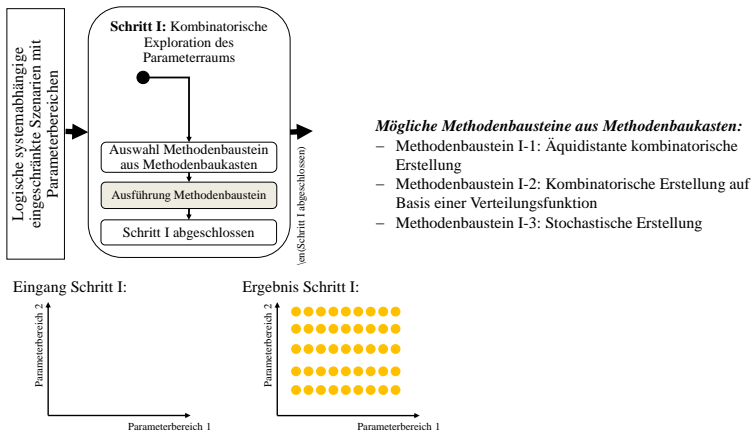


Abbildung 5.7: Schritt I des Selektionsprozesses nach Abbildung 5.3

⁵ Es ist nicht möglich die **Vollständigkeit** empirisch nachzuweisen. Sie muss durch Annahmen und Analysen (vgl. Kapitel 5.5) eingeschränkt und durch Annahmen zum jeweiligen Methodenbaustein bestätigt werden.

Eingang- und Ausgangsgrößen: Die Kombinatorik stellt den ersten Schritt des Selektionsprozesses dar. Hierzu wird als Eingangsgröße ein *Logisches Szenario* mit seinen variablen Parametern aufgegriffen und in eine Liste, welche beispielsweise durch Kombinatorik entsteht, überführt. Diese Liste beinhaltet eine vollständige Übersicht aller Parameterkombinationen. Als Ausgangsgröße wird dem nächsten Schritt eine Liste mit *Konkreten Szenarien* zurückgegeben. Diese *Konkreten Szenarien* beinhalten nach Kapitel 2.4.1 einen konkreten Parametersatz und bilden so den Szenarienteil des ausführbaren Testfalls.

In Kapitel 2.4.1 wurden die vier verschiedenen Parametertypen eingeführt: nominale, ordinale, metrische und zeit-variable Parameter. Nominale und ordinale Parameter werden ohne Festlegung einer Diskretisierungsschrittweite gesampelt, da sie bereits diskret sind und auf alle möglichen Arten kombiniert werden können, um *Konkrete Szenarien* aus ihren zugehörigen *Logischen Szenarien* zu erstellen.

Der Parameterraum wird durch eine lineare, zufallsbasierte oder auf Basis von anderen Verteilungen aufgespannt. Je nach Anzahl der variablen Parameter und gewählter Explorationsmethode entsteht eine andere Liste mit *Konkreten Szenarien* mit dem Ausführungsstatus „Entscheidung ausstehend“ (vgl. Abbildung 5.7). Im ersten Schritt muss nur ein Methodenbaustein zur Generierung angewendet werden.

5.2.1 Methodenbaustein I-1: Äquidistante kombinatorische Erstellung

Ansatz & Vorgehen: Für die gleichmäßige Abtastung des Parameterraums muss vorab eine Diskretisierungsschrittweite pro Parameter, welche als Maß für den Abstand zwischen zwei Parameterwerten gilt, festgelegt werden. Diese unterscheidet sich in Abhängigkeit vom Parametertyp von Parameter zu Parameter und wird auf Basis von Expertenwissen und Erfahrungen ermittelt, indem man die Verteilung der Parameter über das *Logische Szenario* betrachtet oder interne Mess- und Datentypenbeschränkungen berücksichtigt. Die Diskretisierungsschrittweite pro Parameter q_i berechnet sich mit Hilfe der min/max Werte (vgl. Abbildung 2.10)

und der Anzahl an Stützstellen pro Parameterbereich n_i bei einer äquidistanten Abtastung nach Formel 5.5.

$$q_i = \frac{\max_i - \min_i}{n_i} \quad (5.5)$$

Hierbei gibt es bei der initialen Listenerstellung zwei Optionen, welche Einfluss auf die Diskretisierungsschrittweite haben:

- Kombinatorische Liste ohne Subsampling im Selektionsprozess: Die Liste an *Konkreten Szenarien*, welche im Schritt I des Selektionsprozesses erstellt wird, ist final und wird in späteren Schritten nicht durch Subsampling erweitert. Die Diskretisierungsschrittweite muss so klein gewählt werden, dass am Ende eine Abdeckungsaussage über den kontinuierlichen Parameterraum getroffen werden kann, um die Anforderung an die Nachvollziehbarkeit nicht zu verletzen.
- Hinzufügen von weiteren Szenarien im Selektionsprozess durch Subsampling: Erlaubt man das Hinzufügen von weiteren Szenarien zur Liste während des Selektionsprozesses in besonders dynamischen oder kritischen Bereichen, kann die initiale Diskretisierungsschrittweite größer gewählt werden.

Welche der beiden Optionen für das *Logische Szenario* passend ist, hängt von der Anzahl der Parameter und der daraus resultierenden Anzahl an *Konkreten Szenarien* ab. Werden in der Exploration des *Logischen Szenario* weniger als 10 Millionen *Konkrete Szenarien* benötigt, muss kein Subsampling durchgeführt werden, da die Gesamtliste noch eine Verarbeitung zulässt. Bei größeren Gesamtlisten ist das Handling und die Erstellung in den einzelnen Methodenbausteinen der limitierende Faktor. Es muss ein Subsampling durchgeführt werden.

Die Gesamtanzahl der Szenarien lässt sich mit Hilfe der Diskretisierungsschrittweite q_i und der Max- und Min-Werte ermitteln (vgl. Formel 5.6).

$$N_{O_{\text{szenarios}}} = \prod_{i=1}^n k = \prod_{i=1}^n \frac{(max_i - min_i)}{q_i} + 1 \quad (5.6)$$

Annahme 2. Methodenbaustein I-1: Äquidistante kombinatorische Erstellung. Die Diskretisierbarkeit muss ohne Verlust der Gesamtaussage gewährleistet werden. Die Wahl der Diskretisierungsschrittweite kann argumentiert werden, sodass Singularitäten und Auffälligkeiten zwischen den Stützstellen ausgeschlossen werden können. Quereinflüsse zwischen den Parametern bei der Diskretisierung werden nicht betrachtet, da die Diskretisierungsschrittweite pro variablen Parameter gewählt werden muss.

5.2.2 Methodenbaustein I-2: Kombinatorische Erstellung auf Basis einer Verteilungsfunktion

Ansatz & Vorgehen: Ziel des Methodenbausteins ist die Abtastung des Parameterraums durch die Ermittlung der Verteilungsfunktion auf Basis von Messdaten von aufgezeichneten Fahrten, Verkehrsstatistiken oder Risikoanalysen. Die Verteilung ist individuell auf ein Kriterium, wie zum Beispiel Kritikalität oder Auftrittswahrscheinlichkeit, angepasst. Je nach Struktur des *Logischen Szenarios* ist die Verteilungsfunktion vom übergeordneten *Logischen Szenario* vorgegeben. Die Verteilungsfunktion muss in Abhängigkeit des SuTs ausgewählt werden. Die Auswahl der Verteilungsfunktion für die Abtastung des Parameterraums eines SuT, wie einer Notbremsfunktionalität, ist entscheidend für die realistische Simulation von Verkehrssituationen. Die Verteilungsfunktion wird je nach Verkehrsszenario angepasst: In städtischen Umfeldern, mit niedrigen Geschwindigkeiten und häufigen Änderungen, wird eine Normalverteilung verwendet, um die typischen Schwankungen der Fahrzeuggeschwindigkeiten zu modellieren. Auf der Autobahn hingegen, wo höhere Geschwindigkeiten und größere Variationen auftreten, ist eine Log-Normalverteilung sinnvoll, um schnelle Fahrzeuge und gelegentliche langsame Fahrzeuge abzubilden. In Notfallszenarien, wie einem plötzlichen Stau, wird eine Exponentialverteilung genutzt, um die geringe Geschwindigkeit und

seltene Änderungen im Abstand zwischen Fahrzeugen zu reflektieren. Durch die Wahl der passenden Verteilungsfunktion wird bereits die Verteilung im Parameterraum realistisch an das Verhalten der Notbremsfunktionalität angepasst. Eine Verteilungsfunktion ist entweder pro Parameter (vgl. Gleichung 5.7) oder für den gesamten Parameterraum (vgl. Gleichung 5.8) aufgespannt.

$$\vec{p}_{scenario} = \begin{pmatrix} p_1 \\ p_2 \\ \dots \\ p_n \end{pmatrix} = \begin{pmatrix} f_1(\Delta q'_1) \\ f_2(\Delta q'_1) \\ \dots \\ f_n(\Delta q'_n) \end{pmatrix} \quad (5.7)$$

$$\vec{p}_{scenario} = \begin{pmatrix} p_1 \\ p_2 \\ \dots \\ p_n \end{pmatrix} = \vec{f}(\Delta q'_1, \Delta q'_2, \dots, \Delta q'_n) \quad (5.8)$$

Es muss weiterhin eine Diskretisierungsschrittweite q' gewählt werden.

Annahme 3. Methodenbaustein I-2: Kombinatorische Erstellung auf Basis einer Verteilungsfunktion. Die Diskretisierbarkeit muss ohne Verluste der Gesamtaussage gewährleistet werden können. Die Wahl der Diskretisierungsschrittweite kann argumentiert werden, sodass Singularitäten und Auffälligkeiten zwischen den Stützstellen ausgeschlossen werden können. Die Verteilungsfunktionen sind so gewählt, dass auch in bspw. unkritischeren Bereichen noch ausreichend Stützstellen von Konkreten Szenarien vorhanden sind.

5.2.3 Methodenbaustein I-3: Stochastische Erstellung

Ansatz & Vorgehen: Für die stochastische Variation der Parameter muss eine Anzahl an *Konkreten Szenarien* vorgegeben werden. Anschließend wird zufallsbasiert die Liste an *Konkreten Szenarien* erstellt. Zur Ermittlung eines Szenarios wird für jeden variablen Parameter eine Zufallszahl bestimmt und diese dann auf

den vorgegebenen Parameterbereich (min/max-Wert) normiert (vgl. Formel 5.9). Ein *Konkretes Szenario* mit n variablen Parametern erfordert das Ermitteln von n Zufallszahlen.

$$\vec{p}_{\text{szenario}} = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{pmatrix} = \begin{pmatrix} \min_{p_1} + (\max_{p_1} - \min_{p_1}) \cdot \text{Zufallszahl}_1 \\ \min_{p_2} + (\max_{p_2} - \min_{p_2}) \cdot \text{Zufallszahl}_2 \\ \text{esSelektionsprozesses:} \\ \min_{p_n} + (\max_{p_n} - \min_{p_n}) \cdot \text{Zufallszahl}_n \end{pmatrix} \quad (5.9)$$

Die Zufallszahlen sind dabei von einem Algorithmus generierte Zahlen zwischen 0 (inklusive) und 1 (exklusive). Zusätzlich besteht die Option, diesen Methodenbaustein mit einem Abbruchkriterium zu ergänzen. Beispielsweise dient die maximal geforderte Diskretisierungsschrittweite zwischen zwei *Konkreten Szenarien* als Abbruchkriterium. Die Wahl des Abbruchkriteriums (bspw. maximal geforderte Diskretisierungsschrittweite) ist eine zusätzliche Steuermöglichkeit, um Rahmenbedingungen einzuhalten und dennoch eine zufällige Verteilung zu erreichen.

Annahme 4. Methodenbaustein I-3: Stochastische Erstellung. *Die Zahl der zufällig ausgewählten Konkreten Szenarien ist so gewählt, dass der Parameterraum repräsentativ abgebildet wird, ggfs. auch über die Formulierung eines Abbruchkriteriums für die maximal geforderte Diskretisierungsschrittweite. Die Diskretisierbarkeit muss ohne Verluste der Gesamtaussage gewährleistet werden können. Die Anzahl der Konkreten Szenarien kann argumentiert werden, sodass Singularitäten und Auffälligkeiten zwischen den Stützstellen ausgeschlossen werden können.*

Es wird im Schritt I nur ein Methodenbaustein zur Listenerstellung benötigt.

Standardmäßig wird hier eine „Äquidistante kombinatorische Erstellung“ (vgl. Methodenbaustein I-1⁶ in Abschnitt 5.2.1) verwendet. Dieser tastet den Parameterraum gleichmäßig ab, sodass keine relevanten Parameterkombination vergessen werden. Die beiden anderen Methodenbausteine (vgl. Methodenbaustein I-2 in Abschnitt 5.2.2 und Methodenbaustein I-3 in Abschnitt 5.2.3) dürfen allerdings in Spezialfällen, wenn zum Beispiel über den konkreten Parameter genauere Informationen in Form von einer Verteilungsfunktion vorliegen oder mit Hilfe der Methodik schnell zufällige Stützstellen generiert werden müssen, zum Einsatz kommen.

⁶ Die römische Zahl steht dabei für den Schritt indem sich der Methodenbaustein einsetzen lässt. Die arabische Zahl ist eine vorlaufende Nummerierung der Methodenbausteine eines Schrittes. Diese Nummerierung wird für alle Methodenbausteine verwendet.

5.3 Schritt II: Präexekutive Szenarienanalyse und -reduktion

Ziel des Schritts: Ziel des Schrittes ist eine Reduzierung der *Konkreten Szenarien*. Durch die Anwendung des Schritts II werden alle Szenarien „reduziert“, welche sich bereits bei einer Vorabanalyse als unplausibel, unkritisch, unlogisch oder irrelevant identifizieren lassen. Jedes Szenario, welches nicht ausgeführt wird, senkt den Testaufwand. Jedes Szenario, das in Schritt II „reduziert“ wird, trägt somit unmittelbar zur Lösung des in Kapitel 1.2 erwähnten Zielkonflikts bei (vgl. Anforderung 9). Es werden solange Methodenbausteine ausgewählt und ausgeführt bis keine präexekutive Reduktion mehr möglich ist. Hierzu zählen vor allem folgende Reduktions- und Zusammenfassungsmöglichkeiten (siehe auch Abbildung 5.8):

- SuT-funktionslogische Restriktion und Zusammenfassungen (1)
- Szenarien-logische Restriktion und Zusammenfassungen (2)
- Physikalisch bedingte Restriktion und Zusammenfassungen (3)
- ODD-bedingte Restriktion und Zusammenfassungen (4)
- Restriktion und Zusammenfassungen basierend auf Studien- und Messreihen (5)
- Restriktion und Zusammenfassungen auf Basis der Ergebnisse aus vorherigen SuT-Versionen (6)

Eingang- und Ausgangsgrößen:

Je nach verwendetem *Logischen Szenario* bieten sich Methodenbausteine zur präexekutiven Szenarienanalyse an. Die Methodenbausteine lassen sich untereinander kombinieren oder auch einzeln ausführen (vgl. Anforderung 12). Die Auswahl und Reihenfolge der Methodenbausteine sind abhängig von den für das *Logische Szenario* und das SuT ermittelbaren Restriktionen und Zusammenfassungen (vgl. Anforderung 12). Es empfiehlt sich bei den beiden vorgestellten

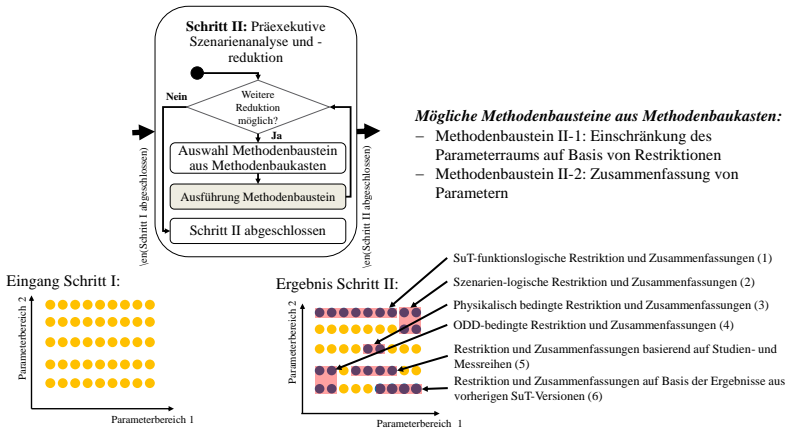


Abbildung 5.8: Schritt II des Selektionsprozesses nach Abbildung 5.3 mit beispielhaften Reduktionen

Methodenbausteinen jedoch zunächst den „Methodenbaustein II-1: Einschränkung des Parameterraums auf Basis von Restriktionen“ auszuführen und erst anschließend den „Methodenbaustein II-2: Zusammenfassung von Parametern“. Das Ergebnis ist dabei unabhängig von der Reihenfolge der Ausführung, jedoch erfordert Methodenbaustein II-2 mehr Rechenleistung, weshalb jedes bereits „reduzierte“ Szenario den Ausführungsaufwand des Methodenbausteins verringert (vgl. Anforderung 9).

5.3.1 Methodenbaustein II-1: Einschränkung des Parameterraums auf Basis von Restriktionen

Ansatz & Vorgehen: Jeder Parameter wird mit seinem jeweiligen Parameterbereich angegeben (Max und Min-Wert). Um den Bereich der Parameter zusätzlich auf Basis von gegenseitigen Einflüssen zu begrenzen, wird eine Liste von Einschränkungen (Restriktionen) erstellt, die der Anwender je nach Szenario, SuT und Reifegrad an die gegebenen Einflussfaktoren und Parameter anpassen und

erweitern muss. Die Liste dient dazu, Quereinflüsse und gegenseitige Abhängigkeiten von Parameterwerten bereits vor der Ausführung auszuschließen. Die Restriktionen stellen dabei mathematische oder logische Beziehungen zwischen verschiedenen variablen Parametern dar und beziehen andere - während der Ausführung - konstante Größen mit ein. Allgemein lässt sich die Restriktion nach Formel 5.10 als Bedingungsausdruck B darstellen:

$$B(p_1, p_2, \dots, p_n) = \text{Wahr/Falsch} \quad (5.10)$$

Nachdem die Restriktionsliste angelegt wurde, wird die Liste der *Konkreten Szenarien* abgearbeitet. Anschließend wird jedes *Konkrete Szenario* auf die Einhaltung aller in der Liste aufgeführten Restriktionen überprüft. Wird ein Verstoß gegen eine Bedingung festgestellt, wird der Ausführungszustand des *Konkreten Szenarios* auf „reduziert“ gesetzt und das nächste *Konkrete Szenario* wird ausgewählt. Für Szenarien, die alle Bedingungen erfüllen, wird das nächste Szenario ausgewählt (vgl. Abbildung 5.9). Nachdem alle *Konkreten Szenarien* auf die Einhaltung der Restriktionen geprüft wurden, werden die geänderten Ausführungszustände abgespeichert und aktualisiert.

Die Restriktionsliste wird auf Basis folgender Zusammenhänge erstellt:

- **Regulatorische Einschränkung:** Der Gesetzgeber definiert bestimmte Abhängigkeiten, wie zum Beispiel die maximale Höchstgeschwindigkeit bei einer Fahrt mit Anhänger.
- **Szenarien-logische Einschränkung:** Das *Logische Szenario* definiert eine Abfolge an Manövern. Wenn hier bereits vorab, aufgrund der gewählten Parameter klar ist, dass das Ergebnis nicht dem spezifizierten Ablauf an Manövern entspricht, muss das Szenario bereits vorab ausgeschlossen werden. Zum Beispiel zum Test eines Notbremsassistenten muss ein Fußgänger die Straße in zeitlicher Reihenfolge vor dem Ego Fahrzeug überqueren, damit das Szenario logisch sinn ergibt.

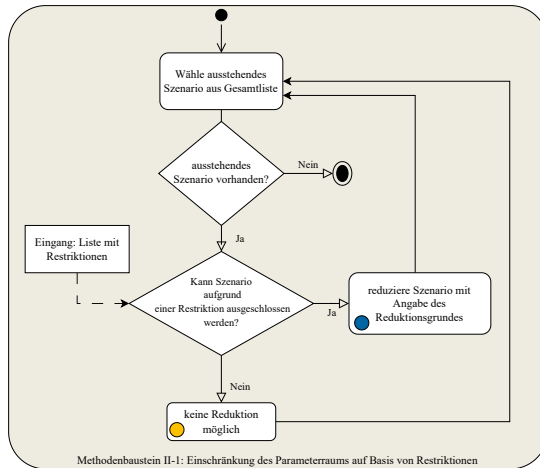


Abbildung 5.9: Anwendung der Liste von Restriktionen auf die Szenarienliste

- **Systembedingte Einschränkung:** In Abhängigkeit der Anforderungen an das SuT werden Szenarien ausgeschlossen. Ein Notbremsassistent ist in der Regel für bestimmte Geschwindigkeitsbereiche ausgelegt. Falls das System beispielsweise nur im Bereich zwischen 10 km/h und 120 km/h aktiv ist, müssen Szenarien, die außerhalb dieses Bereichs liegen, ausgeschlossen werden.

Diese Methode schränkt die Anzahl der Szenarien ein, indem sie die Bandbreite der Werte für die einzelnen Parameter „reduziert“. Durch die Reduktion wird keine Ausführung benötigt, dadurch wird direkt der Testaufwand reduziert (vgl. Anforderung 9). Falls ein Szenario „reduziert“ wird, wird die Reduktionsargumentation mit dem jeweiligen Restriktionsgrund befüllt (vgl. Anforderung 13).

Annahme 5. Methodenbaustein II-1: Einschränkung des Parameterraums auf Basis von Restriktionen. Da in diesem Schritt keine aufgezeichneten Daten im SuT-Kontext von bereits ausgeführten Testfällen verwendet werden, ist mit den formulierten Restriktionen immer ein Modell bzw. eine Vereinfachung verbunden.

Es muss daher angenommen werden, dass die Restriktion im Szenarien- und Systemkontext eine gültige Aussage trifft. Dies kann durch Ausführung einiger „reduzierter“ Szenarien und dem Vergleich der Abweichung der Annahme in der Restriktion überprüft werden.

5.3.2 Methodenbaustein II-2: Zusammenfassung von Parametern

Ansatz & Vorgehen:

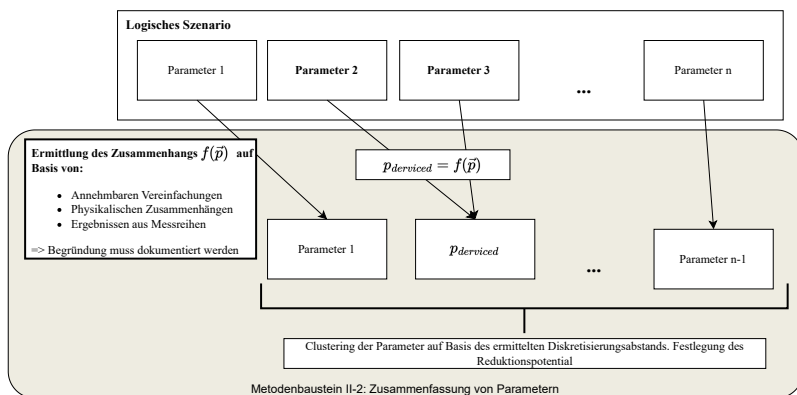


Abbildung 5.10: Prinzipische Skizze zur Zusammenfassung von mehreren Parameter zu einem abgeleiteten Parameter

Dadurch wird die Menge und der Umfang der zu variierenden Parameter direkt reduziert. Geht man davon aus, dass die Gesamtliste an Konkreten Szenarien auf Basis von X Stützstellen pro Parameter des übergeordneten Logischen Szenarios erstellt wurde (vgl. Kapitel 5.2), ergibt sich beim Zusammenfassen von i variablen Parametern das Reduktionspotential red_{pot} nach folgender Formel:

$$red_{pot} = \frac{X^{p-i}}{X^p} \tag{5.11}$$

Der Methodenbaustein besteht dabei aus vier Schritten:

Subschritt α : Formulierung und Anwendung der Zusammenfassung (Korrelation): Bei diesem Subschritt werden Beziehungen zwischen verschiedenen Parametern ermittelt, um abgeleitete Parameter zu erstellen, wobei sowohl lineare als auch nicht lineare Zusammenfassungen berücksichtigt werden. Zur Veranschaulichung wird der Parametervektor in einem *Konkreten Szenario* betrachtet:

$$\vec{p} = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ \dots \\ p_{n-1} \\ p_n \end{pmatrix} \quad (5.12)$$

In Fällen, in denen mehrere variable Parameter (vgl. Definition 7) beispielsweise durch physikalische oder szenarienablauf-logische Zusammenhänge voneinander abhängen und diese Abhängigkeit mit Hilfe einer Formel (vgl. Formel 5.13) in Abhängigkeit der ursprünglichen Parameter ausgedrückt werden kann, werden sie zu einem abgeleiteten Parameter zusammengefasst. Dieser abgeleitete Parameter muss als Formel der variablen Parameter beschrieben werden (vgl. Formel 5.13).

$$p_{derived} = f(\vec{p}) \quad (5.13)$$

Durch Anwendung der Korrelation zwischen beispielsweise dem Parameter p_2 und dem Parameter p_3 (vgl. Formel 5.13) wird ein neuer reduzierter Parametervektor abgeleitet, wie in Formel 5.14 exemplarisch dargestellt. Zur Identifikation der Beziehungen zwischen den Parameter hilft das Aufstellen eine Korrelationsmatrix zwischen den Parametern um die Zusammenhänge zwischen den Parametern zu erkennen. Für das Aufstellen der Korrelationsmatrix sind allerdings bereits ausgeführte Szenarien und Messungen notwendig. Falls diese aus alten Baureihen

oder vorherigen Versionen nicht vorhanden sind, sind die Zusammenhänge auch durch mathematische oder logische Überlegungen erfassbar.

$$\begin{pmatrix} p_1 \\ p_{derived} \\ \dots \\ p_{n-1} \end{pmatrix} = \begin{pmatrix} p_1 \\ f(p_2, p_3) \\ \dots \\ p_{n-1} \end{pmatrix} \quad (5.14)$$

Wurde bei der Auswahl der Parameter des *Logischen Szenarios* der Systemkontext bzw. das Testziel noch nicht betrachtet, muss dies spätestens in diesem Methodenbaustein berücksichtigt werden. Durch die Analyse der Einflüsse der Parameter auf die Systemteile ergeben sich unterschiedlich starke Priorisierungen der Parameter. Zum Beispiel hat eine Veränderung der Sonneneinstrahlung keine Veränderung des Bremswegs zur Folge. Das System Bremse bleibt vom Parameter Sonneneinstrahlung unberührt, wohingegen die kamerabasierte Objekterkennung sehr wohl vom Sonnenstand beeinflusst wird. Die Zuordnung der einzelnen Parameter auf den jeweiligen Systemteil erfolgt mit Hilfe von physikalischen Ausschlussprinzipien, auf Basis von Expertenwissen und Erfahrungswerte bzw. Studien- und Messreihen.

Generell müssen in diesem Subschritt unter anderem physikalische und mathematische Vereinfachungen des Szenarios vorgenommen werden, um verschiedene variable Parameter zu einem relevanten Parameter zusammenzufassen. Beispielsweise werden durch kinematische Beziehungen im Szenario verschiedene Startgeschwindigkeiten von verschiedenen Verkehrsteilnehmern auf einen relevanten Parameter reduziert. Diese Annahmen werden unter der Verwendung von Modellen und Vereinfachungen angenommen.

Anschließend werden die Vereinfachungen auf alle „ausstehenden“ *Konkrete Szenarien* angewendet. Der aggregierte Parameter wird zum Szenario als neuer Parameter hinzugefügt.

Subschritt β : Ermittlung der neuen Diskretisierungsschrittweite auf der Basis der Fehlerfortpflanzung: Anschließend muss ermittelt werden, welche Szenarien ähnliche Parameterkombinationen aufweisen, um so einen repräsentativen Parametersatz auszuwählen. Hierzu wird die Diskretisierungsschrittweite des aggregierten Parameters aus der Diskretisierungsschrittweite der betroffenen Parameter ermittelt. Als Grundlage wird das Prinzip der Fehlerfortpflanzung auf die formulierte Zusammenfassung angewendet. In Abhängigkeit der gewählten Zusammenfassung ergänzen sich die Fehler nach folgenden Prinzipien:

- Additive/ Subtraktive Fortpflanzung (siehe [38])
- Multiplikative Fortpflanzung (siehe [38])

Anschließend wird die aggregierte Diskretisierungsschrittweite dazu verwendet, um die Gesamtanzahl der Cluster zu bestimmen.

$$no_{cluster} = (p_{derived_{max}} - p_{derived_{min}}) / p_{derived_{stepsize}} + 1 \quad (5.15)$$

Subschritt γ : Clustering der Daten auf Basis der vorgegeben Diskretisierungsschrittweite: Auf Basis der berechneten aggregierten Diskretisierungsschrittweiten und der ermittelten Anzahl der Cluster müssen, durch lineare äquidistante Annäherung (vgl. Formel 5.16) oder durch die Verwendung eines Klassifikationsansatzes (bspw. k-Means), die bestehenden Parametersätze in Kategorien eingeteilt werden.

$$cluster_{label} = (p_{derived} - p_{derived_{min}}) / p_{derived_{stepsize}} \quad (5.16)$$

Dabei werden *Konkrete Szenarien* mit den gleichen aggregierten Parametern in dasselbe Cluster eingeordnet.

Subschritt δ : Auswahl von Repräsentanten auf Basis des Clusterings: Im Subschritt δ müssen im Datensatz Duplikate gefunden werden. Hierzu wird das

Clusterlabel zusammen mit den übrigen nicht aggregierten Parametern in Gruppen eingeteilt. Gruppen entstehend indem *Konkrete Szenarien* mit identischen variablen Parameterwerten (übrige variable Parameter und aggregierte Parameter sind identisch) zusammengefasst werden. Aus jeder Gruppe wird eine vordefinierte Anzahl an *Konkreten Szenarien* beibehalten. Alle anderen werden aufgrund ihrer Ähnlichkeit „reduziert“.

Annahme 6. Methodenbaustein II-2: Zusammenfassung von Parametern. *Da in diesem Schritt keine aufgezeichneten Daten im SuT-Kontext von bereits ausgeführten Testfällen verwendet werden, ist mit den formulierten Zusammenfassungen immer ein Modell bzw. eine Vereinfachung verbunden. Es muss daher angenommen werden, dass die Zusammenfassungen im Szenarien- und Systemkontext eine gültige Aussage trifft. Gleichzeitig wird die Diskretisierungsschrittweite des aggregierten Parameters auf Basis der ursprünglichen Parameter festgelegt. Die ursprüngliche Schrittweite der variablen Parameter muss daher auch passend der Annahmen aus Schritt I (vgl. Kapitel 5.2) gewählt worden sein. Eine Validierung der Annahmen lässt sich über eine Referenzausführung von einigen „reduzierten“ Szenarien vornehmen.*

5.4 Schritt III: Interexekutive (zyklische) Szenarienanalyse und -reduktion

Ziel des Schritts: In diesem Schritt werden zum ersten Mal *Konkrete Szenarien* ausgeführt und die Ergebnisse für eine weitere Testpriorisierung verwendet. Der Ansatz zur Bildung einer Repräsentativität nutzt bereits „ausgeführte“ und evaluierte Szenarien⁷. Durch die Variation der variablen Parameter ändert sich der Zeitpunkt der ersten Interaktion mit dem jeweiligen SuT und damit auch der Zeitpunkt, wann ein VRU relevant wird (z. B. VRU betritt die Straße, VRU das erste Mal im Sichtfeld, etc.). Je größer die Distanz zu einem bereits „ausgeführten“ *Konkreten Szenario* ist, desto weiter nimmt die Aussagekraft des „ausgeführten“ Szenarios ab und desto „unähnlicher“ sind diese Szenarien. Zur Ermittlung der Distanz zwischen zwei *Konkreten Szenarien* wird ein Distanzmaß zum Vergleich der Szenarien benötigt, welches in ein Ähnlichkeitsmaß überführt werden muss. Dieses Ähnlichkeitsmaß basiert auf bereits bestehenden Methoden und nutzt die bereits „ausgeführten“ und evaluierten Ergebnisse zur Analyse der „ausstehenden“ *Konkreten Szenarien*. Die Ermittlung der Ähnlichkeit zwischen zwei *Konkreten Szenarien* basiert auf Metriken, Unsicherheiten, Auftrittswahrscheinlichkeiten bzw. Kritikalitäten, welche über Ansätze zur Ähnlichkeitsermittlung verglichen werden (vgl. Kapitel 3). Ziel dieses Schrittes ist es, die restlichen „ausstehenden“ Szenarien zu „reduzieren“ oder „auszuführen“ (vgl. Abbildung 5.11). Hierbei werden bestehende Methoden von Ries et al. [92] [93], Braun et al. [24] und Bachet al. [9] angewandt (vgl. Anforderung 5).

Eingang- und Ausgangsgrößen: Im letzten Schritt (Schritt III) wird die durch vorherige Methodenbausteine reduzierte Gesamtliste an *Konkreten*, noch nicht ausgeführten *Szenarien* iterativ ausgeführt. Nach jedem durchgeführten Methodenbaustein werden die getroffenen Annahmen dokumentiert. Hierbei wird über Methoden zur Bestimmung von Ähnlichkeiten, wie durch den Vergleich von

⁷ **Evaluierte Szenarien** sind Szenarien bei denen auf Basis einer Ausführung an einer Testinstanz bereits eine Auswertung der Zeitreihen und KPIs über Metriken, und Bewertungskriterien durchgeführt wurde.

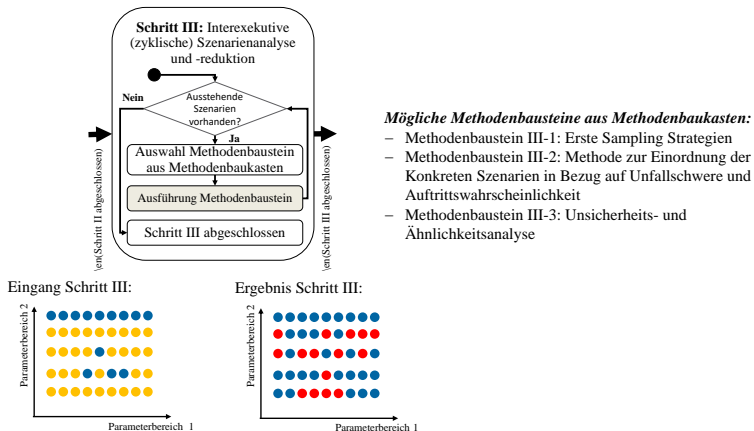


Abbildung 5.11: Schritt III des Selektionsprozesses nach Abbildung 5.3 mit beispielhaften Reduktionen

Zeitreihen mit Hilfe des Dynamic Time Warping (DTW) (vgl. Kapitel 3.2.3), ein Maß für die Ähnlichkeit berechnet. Es werden die Ergebnisse der iterativen Ausführung der Testfälle an der Testinstanz dazu genutzt ähnliche Szenarien zu finden und entsprechend repräsentative Vertreter aus den ähnlichen Bereichen zu selektieren, welche repräsentativ für den gesamten Bereich stehen. Ziel ist es, die Generalisierungsannahme für das gesamte *Logische Szenario* mit wenigen ausgeführten Testfällen zu erfüllen (vgl. Anforderung 8). Dabei müssen zunächst erste *Konkrete Szenarien* „ausgeführt“ werden. „Methodenbaustein III-1: Erste Sampling Strategien“ ist daher zwingend erforderlich, um eine erste Messung zu erzeugen. „Methodenbaustein III-2: Methode zur Einordnung der Konkreten Szenarien in Bezug auf Unfallschwere und Auftretswahrscheinlichkeit“ ist optional und lässt sich anwenden, wenn Informationen über die Unfallschwere und Auftretshäufigkeit aus repräsentativen Daten vorliegen. Anschließend lassen sich weitere Methodenbausteine anwenden. Abschließend muss ein Methodenbaustein folgen, welcher für die restlichen „ausstehenden“ *Konkreten Szenarien* eine Entscheidung trifft. Der „Methodenbaustein III-3: Unsicherheits- und Ähnlichkeitsanalyse“ bietet diese Option.

5.4.1 Methodenbaustein III-1: Erste Sampling Strategien

Zur Bestimmung von ersten Erkenntnissen über den Parameterraum müssen Ausführungsergebnisse generiert werden. Zur Ermittlung, welche *Konkrete Szenarien* aus der Gesamtliste als erstes „ausgeführt“ werden müssen, stellt dieser Methodenbaustein einen Ansatz zur Bestimmung einer ersten Grundmenge an „ausgeführten“ *Konkreten Szenarien* vor. Dieser Ansatz nutzt die Distanz zwischen zwei Parametersätzen. Die Ergebnisse der Ausführung werden dann in weiteren Analyseschritten miteinbezogen. Die Ermittlung des Abstands zwischen zwei *Konkreten Szenarien* basiert beispielsweise auf dem euklidischen Abstand zwischen zwei Parameterkombinationen.

Nachdem im ersten Schritt die Gesamtliste an *Konkreten Szenarien* generiert wurde (vgl. Kapitel 5.2) und im zweiten Schritt bereits einige Szenarien „reduziert“ wurden (vgl. Kapitel 5.3), müssen nun Ausführungsergebnisse generiert werden. Hierfür stehen verschiedene Sampling Methoden zur initialen Abstastung des Parameterraums, zur Verfügung. Jede Methode benötigt einen Samplingabstand pro variablen Parameter als Eingangsgröße.

Grid-basiertes Sampling: Beim Grid-basierten Sampling wird basierend auf der pro Parameter festgelegten Abstastung ein gleichmäßiges Grid über den bestehenden Szenarienraum gelegt. Der Abstand zwischen den Szenarien ist in jeder Parameterrichtung konstant.

City-Block Sampling: Basierend auf dem City-Block Abstand (vgl. Kapitel 3.2.1) lässt sich der normierte Abstand zwischen einer bestehenden ausgewählten Stützstelle i und einer offenen Stützstelle j nach Formel 5.17 berechnen.

$$d_{ijNorm}^{CB} = \sum_{l=1}^k \frac{1}{n_{sample_k}} \cdot |p_{li} - p_{lj}| \quad (i, j = 1, \dots, n) \quad (5.17)$$

Ist der Abstand zwischen den zwei Szenarien größer oder gleich 1, muss das Szenario ausgeführt werden.

$$\begin{cases} \text{Wenn } d_{ijNorm}^{CB} \geq 1, & \text{Szenario ausführen,} \\ \text{sonst,} & \text{Szenario ausstehend.} \end{cases} \quad (5.18)$$

Euklidisches Sampling: Basierend auf dem euklidischen Abstand (vgl. Kapitel 3.2.1) lässt sich der normierte Abstand zwischen einer bestehenden ausgewählten Stützstelle i und einer offenen Stützstelle j nach Formel 5.19 berechnen.

$$d_{ijNorm}^{EK} = \sqrt{\sum_{l=1}^k \frac{1}{n_{sample_k}} * (p_{li} - p_{lj})^2} \quad (i, j = 1, \dots, n) \quad (5.19)$$

Ist der Abstand zwischen den zwei Szenarien größer oder gleich 1, muss das Szenario ausgeführt werden.

$$\begin{cases} \text{Wenn } d_{ijNorm}^{EK} \geq 1, & \text{Szenario ausführen,} \\ \text{sonst,} & \text{Szenario ausstehend.} \end{cases} \quad (5.20)$$

Auch andere Abstandsberechnungsvorschriften (vgl. Kapitel 3.2) lassen sich verwenden.

Annahme 7. Methodenbaustein III-1: Erste Sampling Strategien. *Durch das Sampling werden keine Szenarien „reduziert“. Es müssen daher keine Annahmen getroffen werden. Allerdings ist dieser Methodenbaustein mit einem hohen Ausführungsaufwand verbunden, da ohne Berücksichtigung von Einschränkungen und Messergebnisse Konkrete Szenarien „ausgeführt“ werden.*

5.4.2 Methodenbaustein III-2: Methode zur Einordnung der Konkreten Szenarien in Bezug auf Unfallschwere und Auftrittswahrscheinlichkeit

Zur Ermittlung von relevanten *Konkreten Szenarien* zum Test einer hochautomatisierten Fahrfunktion im Absicherungs- und Entwicklungskontext ist es nicht ausreichend, nur die kritischen Situationen zu testen. Zur korrekten Priorisierung der *Konkreten Szenarien* muss ein kombinierter Ansatz aus Unfallschwere, Systemrelevanz und Häufigkeit gewählt werden, damit nicht nur Grenzfälle und Singularitäten getestet werden, sondern auch häufige Situationen, welche die hochautomatisierten Fahrzeuge täglich meistern müssen. Die Auftrittswahrscheinlichkeit eines *Konkreten Szenarios* in einem *Logischen Szenario* wird dazu verwendet, eine Relevanzbestimmung durchzuführen oder besonders häufige *Konkrete Szenarien* stärker zu priorisieren.

In dieser Methode wird anhand der Auftrittswahrscheinlichkeit (*exposure*) und Unfallschwere (*severity*) bestimmt, ob ein Szenario „ausgeführt“ wird⁸. Diese Berechnungen erfolgen unter der Verwendung von Referenzdaten aus bereits aufgezeichneten Realfahrten, Unfallstatistiken, öffentlichen Fahrzeugdatensätzen oder aus Top-View Drohnenaufnahmen, wie sie in Datensätzen, wie HighD, InD oder RoundD zu finden sind. Für die Auftrittswahrscheinlichkeit und die Unfallschwere wird zunächst eine separate Metrik entwickelt. Diese wird anschließend zu einer Gesamtpriorisierung zusammengefasst (vgl. Abbildung 5.12). Diese wird dafür verwendet, um kritische und häufige *Konkrete Szenarien* intensiver zu testen.

Bewertung der Unfallschwere (*severity*) von Konkreten Szenarien

Zur Bewertung der Unfallschwere werden Metriken nach Kapitel 3.2.4 herangezogen.

⁸ Das Schema lehnt sich an die ASIL (Automotive Safety Integrity Level) Klassifikation [49] zur funktionalen Sicherheit für die Schwere des Fehlers bzw. dem Gefährdungspotential des Nutzers oder der Umgebung an.

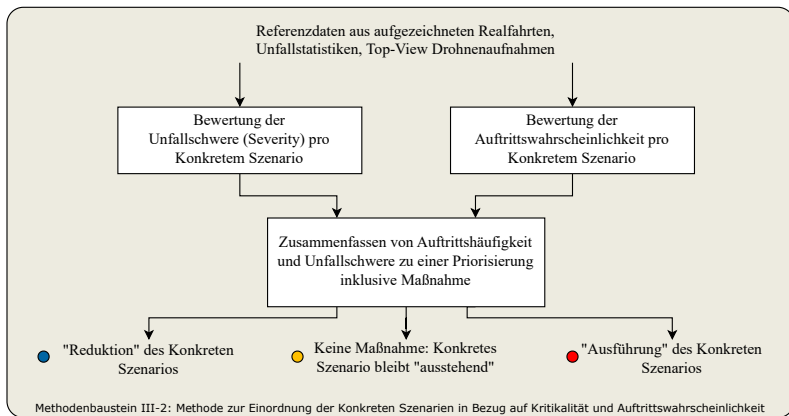


Abbildung 5.12: Zusammenfassen von Unfallschwere und Auftretswahrscheinlichkeit zu einer Gesamtpriorisierung

Diese werden entweder bereits vor der Ausführung auf Parameterbasis berechnet (z. B. je höher die Geschwindigkeit des Ego-Fahrzeugs, desto größer die Unfallschwere bei einem möglichen Unfall) oder nachträglich auf Basis von Metriken für bereits „ausgeführte“ *Konkrete Szenarien* bewertet (z. B. je geringer die Time-to-Collision (TTC), desto kritischer ist das Kollisionsrisiko) (vgl. Kapitel 4.2.2).

Die verschiedenen Metriken m werden anhand biomechanischer Verletzungsskalen bewertet, um die Schwere der Verletzungen realitätsnah abzubilden. Dabei wird insbesondere die Abbreviated Injury Scale (AIS) verwendet, die auf medizinischen Untersuchungen basiert und das Verletzungsrisiko für verschiedene Körperregionen beschreibt. Die Schwere eines Unfalls wird auf Grundlage von Metriken wie Delta-V, HIC (Head Injury Criterion), Aufprallkräfte (g-Kräfte) und Verletzungstoleranzen aus Crashtests bestimmt.

Jedes konkrete Szenario wird anhand der berechneten biomechanischen Indikatoren einer AIS-Kategorie zugeordnet und anschließend in die folgenden Severity-Klassen überführt:

- Severity 0 (S0): Keine Verletzungen – AIS 0
- Severity 1 (S1): Leichte bis mittelschwere Verletzungen, keine bleibenden Schäden – AIS 1–2
- Severity 2 (S2): Schwere Verletzungen, Überleben sehr wahrscheinlich – AIS 3–4
- Severity 3 (S3): Kritische oder lebensbedrohliche Verletzungen – AIS 5–6

Die Einstufung erfolgt auf Basis validierter biomechanischer Daten und Crashtest-Ergebnisse. Beispielsweise wird eine Kollision mit Delta-V von 35 km/h und einem HIC-Wert von 1200 einem AIS 3–4 (Schwere Verletzungen) zugeordnet, was zu einer Severity-Klassifikation S2 führt.

Dieser Ansatz erlaubt eine medizinisch fundierte und nachvollziehbare Bewertung der Unfallschwere und wird unter anderem von der amerikanischen Highway Verkehrsbehörde (NHTSA) dazu verwendet Unfälle nach ihrer Schwere einzuordnen [76].

Die Skala geht dabei von keine Verletzungen bis zu Überleben unwahrscheinlich. Putz et al. [86] nutzen diesen Ansatz bereits für die Zuordnung von Szenarien zu einer Unfallschwere. Die Severity-Klassen lassen sich in späteren Methodenbausteinen dazu benutzen, die Szenarien entsprechend zu priorisieren und die Teststeuerung so zu gewichten, dass Szenarien mit einer höheren Severity-Klasse eher ausgeführt werden als andere. Die Informationen werden als Metadaten an das *Konkrete Szenario* angehängt und später zusammen mit der Auftrittswahrscheinlichkeit für eine Entscheidung verwendet.

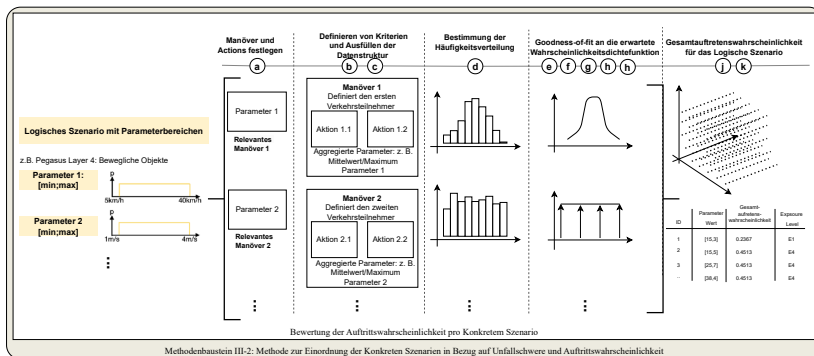


Abbildung 5.13: Grafische Darstellung des Konzepts zur Ermittlung der Auftretenshäufigkeit - analog zu Abbildung 5.14

Bewertung der Auftretenswahrscheinlichkeit (exposure) von Konkreten Szenarien

Das Verfahren ist ein datenbasierter Ansatz zur Definition (a), Ermittlung (b,c) und statistischen Analyse (d,e,f,g,h,i,j,k) spezifischer *Konkreter Szenarien* mit dem Schwerpunkt auf den für die Schwere des Fehlers in der realen Welt auftretenden Parameterwerten (vgl. Abbildung 5.13).

Als Ergebnis liefert der Subschritt einen Wahrscheinlichkeitswert für die geschätzte Auftretenswahrscheinlichkeit jedes spezifischen *Konkreten Szenarios* (vgl. Abbildung 5.13). Abschließend wird diese Auftretenswahrscheinlichkeit dazu verwendet, eine Priorisierung auf Basis der wahrscheinlichsten und relevantesten *Konkreten Szenarien* durchzuführen, indem diese einem Exposure-Level zugeordnet werden. Das *Logische Szenario* gibt die variablen Parameter vor.

Ziel der Methode ist eine datenbasierte Ableitung von repräsentativen Parameterverteilungen für die spezifischen *Konkreten Szenarien* eines *Logischen Szenarios*. Zu diesem Zweck wurde in diesem Subschritt ein Fahrzeugdatensatz schrittweise analysiert und klassifiziert (vgl. Abbildung 5.14 - Subschritt α) und diese Informationen anschließend zur Ermittlung einer Parameterverteilungen genutzt (vgl. Abbildung 5.14 - Subschritt β).

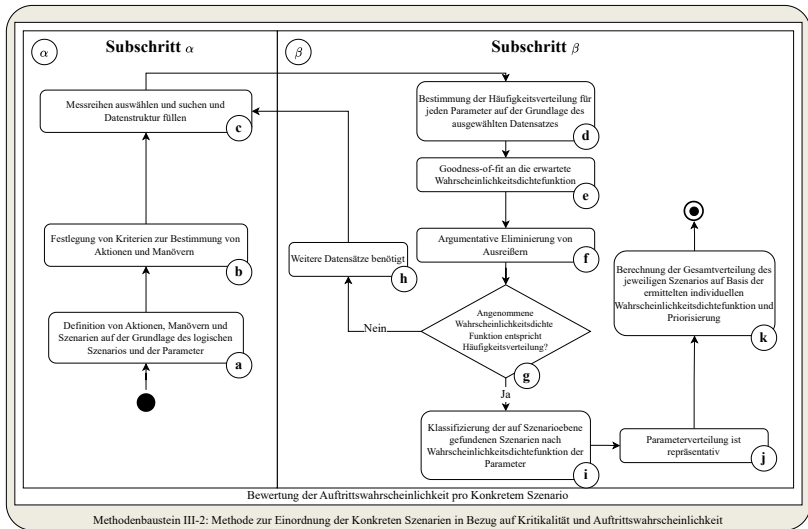


Abbildung 5.14: Aktivitätsdiagramm zur Ermittlung der Häufigkeit eines Konkreten Szenario in einem logischen Szenario

Subschritt α : Datenanalyse und -extraktion

Die Methode nutzt die Einzelverteilungen der variablen Parameter des *Logischen Szenarios*, um diese anschließend zu einer Gesamtverteilung für alle Parameterkombinationen des übergeordneten *Logischen Szenarios* zu aggregieren. Zur Ermittlung der Auftrittswahrscheinlichkeit eines *Logischen Szenarios* werden nach King et al. [53] Manöverbeschreibungen und die Zerlegung in Aktionen verwendet. Hierzu werden Ego-Fahrzeugdatensätze oder Top-View Drohnenaufnahmen benutzt, um eine Auftretenshäufigkeit auf Ebene eines *Logischen Szenarios* zu ermitteln. Dieses Konzept lässt sich auf die Problemstellung der Ermittlung der Häufigkeitsverteilung eines *Konkreten Szenarios* übertragen. Auch hier unterstützen Aktions- und Manöverbeschreibungen dabei, konkrete Parameterausprägungen in Messungen zu finden und diese in ein *Konkretes Szenario* mit Häufigkeiten zu überführen (vgl. Abbildung 5.15). Im Vergleich zur Darstellung nach King et al. [53] werden zusätzlich Parameterwerte als Eigenschaft auf Manöverebene

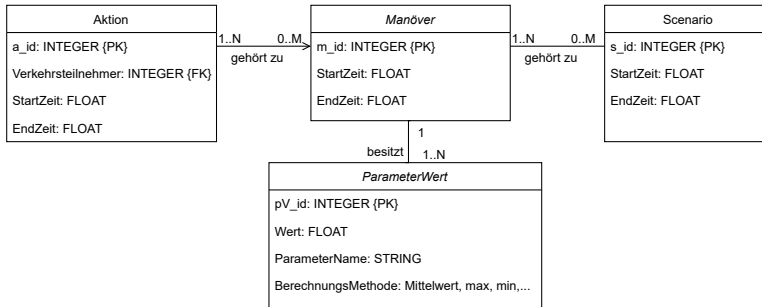


Abbildung 5.15: Datenstruktur zur Zerlegung der Messreihen in eine manöverbasierte Beschreibungsform mit Parameternwerten

an jedes *Konkrete Szenario* angehängt (vgl. Abbildung 5.15). Diese erlauben eine spätere Aggregation der einzelnen Parameterverteilungen zu einem *Konkreten Szenario*. Jedes Element (Aktion, Manöver und Szenario) hat einen Startzeitpunkt und einen Endzeitpunkt.

Teilschritt a - Definition von Aktionen, Manövern und Szenarien auf der Grundlage des Logischen Szenarios und der Parameter.

Als Erstes muss eine Liste relevanter Aktionen in Abhängigkeit vom *Logischen Szenario*, den zugehörigen Parametern des *Logischen Szenarios*, der Verkehrsteilnehmer und der Umweltobjekte definiert werden (vgl. Abbildung 5.14 - Teilschritt a). Die Aktionen werden auf der Grundlage kausaler, logischer und geometrischer Bedingungen für die bestehenden Zeitreihen auf der Grundlage des *Logischen Szenarios* definiert. Wenn alle Bedingungen erfüllt sind, wird eine neue Aktionsinstanz erzeugt. Aktionen werden den verschiedenen Pegasus-Ebenen zugeordnet. Dies bedeutet, dass beispielsweise Wetterbedingungen (Pegasus Layer 5) als Aktionen in die Analyse einbezogen werden. Dieser Schritt wird manuell durch Expertenwissen über das *Logische Szenario* durchgeführt.

Teilschritt b - Festlegung von Kriterien für die Bestimmung von Aktionen und Manövern.

Die Aktionen bilden die Grundlage der Datenstruktur und müssen im nächsten Schritt zu Manövern aggregiert werden. Diese Aggregation erfolgt auf der Grundlage von vordefinierten Regeln für die zeitliche und logische Abfolge von Aktionen. Eine Aktion kann zu mehreren Manövern gehören. Im letzten Schritt der Datenerhebung werden die Manöver auf Basis vordefinierter Regeln, wie z. B. der logischen und zeitlichen Abfolge mehrerer Verkehrsteilnehmer, zu Szenarien auf Szenarioebene zusammengefasst (vgl. Abbildung 5.14 - Teilschritt b). Das betrachtete *Logische Szenario* ist in mehrere Manöver unterteilt, die durch Beziehungen und Bedingungen miteinander verbunden sind (vgl. Abbildung 5.15). Ein Manöver⁹ enthält einen oder mehrere Parameter.

Teilschritt c - Messreihen auswählen, filtern und Datenstruktur füllen.

Im Gegensatz zu Ansätzen nach King et al. [53] werden Parameterwerte, die das Manöver beschreiben, auf Manöverebene ergänzt. Diese Parameterwerte werden auf der Basis von Zeitreihen ermittelt, zum Beispiel durch Berechnung des Mittelwertes über die Aktivierungszeit. Zur Bestimmung des Auftretens wird diese Datenstruktur (vgl. Abbildung 5.15) in mehreren Schritten mit Daten aus Messreihen von z. B. Drohnen-Datensätzen aus der Draufsicht ausgewählt und anschließend analysiert (vgl. Abbildung 5.14 - Teilschritt c). Neben der allgemeinen Verfügbarkeit der Daten spielen Faktoren wie der Umfang der Daten, die Varianz der Datensätze und die Art der Datenerfassung eine Rolle.

Subschritt β : Schätzung einer zuverlässigen Wahrscheinlichkeitsdichtefunktion (PDF)

Teilschritt d - Bestimmung der Häufigkeitsverteilung für jeden Parameter auf der Grundlage des ausgewählten Datensatzes.

⁹ Der Begriff **Manöver** ist hier eine allgemeine Beschreibung einer Unterkategorie und eines Merkmals des Szenarios. Im Zusammenhang mit sich bewegenden Verkehrsteilnehmern wird ein Manöver beispielsweise durch die Bewegung eines Teilnehmers oder eines Fahrzeugs definiert.

In Subschritt β muss die gefüllte Datenstruktur anhand der zugehörigen Parameterwerte des *Logischen Szenarios* analysiert werden. Die Parameterwerte auf Manöverebene werden gezählt und als Häufigkeitsverteilung pro Parameter des *Logischen Szenarios* ausgewertet (vgl. Abbildung 5.14 - Teilschritt d). Anschließend muss aus der vorliegenden Verteilung eine Wahrscheinlichkeitsdichtefunktion geschätzt werden¹⁰. Wird eine Normalverteilung angenommen, wird die Wahrscheinlichkeit mit Hilfe der approximierten Wahrscheinlichkeitsdichtefunktion (Gaußsche Wahrscheinlichkeitsdichtefunktion) [84] ermittelt. Daraus ergibt sich die Einzelauftrittswahrscheinlichkeit für einen Parameterwert w_i in Abhängigkeit von der Standardabweichung σ und dem Mittelwert des jeweiligen Parameterbereichs μ_i :

$$w_i(p_i, \mu_i, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(p_i - \mu_i)^2}{2\sigma^2}} \quad (5.21)$$

Dieses Verfahren funktioniert auf die gleiche Weise für andere Wahrscheinlichkeitsdichtefunktionen.

Teilschritt e - Goodness-of-fit an die erwartete Wahrscheinlichkeitsdichtefunktion.

Die statistische Analyse dient dazu, die Eignung und Signifikanz der ausgewählten Daten zu bewerten und zu überprüfen. Statistische Beschreibungsmittel wie Mittelwert, Median, Varianz und Standardabweichung bieten eine quantitative Grundlage zur Erkennung der zentralen Tendenzen und der Streuung der Daten. Diese Metriken geben einen ersten Einblick in die Struktur der Daten und helfen, Abweichungen von erwarteten Mustern zu erkennen. Die visuelle statistische Analyse mit Hilfe von Tools, wie Histogrammen, Box-Plots und Quantil-Quantil (Q-Q)-Plots ergänzt diese deskriptiven Statistiken, indem sie eine Visualisierung der Datenverteilung erlaubt.

Nach der Analyse der Daten mit Hilfe von deskriptiven Statistiken und visuellen Methoden - wie dem Q-Q-Plot - um einen ersten Eindruck von der Verteilung zu gewinnen und sie mit den ursprünglichen Annahmen zu vergleichen - erfolgt

¹⁰ Diese Annahme wird im Teilschritt e überprüft

eine weitere quantitative Analyse durch Goodness-of-fit Tests (wie Kolmogorov-Smirnov [16] und Shapiro-Wilk [106] [70]). Diese werden verwendet, um zu prüfen, ob die Daten einer bestimmten theoretischen Wahrscheinlichkeitsdichtefunktion entsprechen (vgl. Abbildung 5.14 - Teilschritt e). Sie quantifizieren die Anpassung zwischen der beobachteten Datenverteilung und einer angenommenen theoretischen Verteilung (z. B. normal, exponentiell, log-normal). Die Verwendung dieser Tests erlaubt eine objektive Bewertung der Anpassungsgüte (Goodness-of-fit) und unterstützt die Validierung oder Ablehnung der initialen Annahmen über die Wahrscheinlichkeitsdichtefunktion, was für die anschließende Datenanalyse und -interpretation von wesentlicher Bedeutung ist.

Teilschritt f - Argumentative Eliminierung von Ausreißern.

Bei der Analyse der Ergebnisse der Goodness-of-fit-Tests und der Verteilung wurden diese auf signifikante Abweichungen der erwarteten Wahrscheinlichkeitsdichtefunktion geprüft, die zu Test- und Analysefehlern führen könnten. Ausreißer konnten mit Hilfe des Interquartilsbereichs (IQR) [28] und seiner definierten Intervalle entfernt werden, wenn sie als fehlerhaft oder nicht relevant für die Verteilung angesehen wurden. Auch eine manuelle Analyse hilft dabei Ausreißer zu entfernen (vgl. Abbildung 5.14 - Teilschritt f).

Teilschritt h - Weitere Datensätze benötigt.

Sind diese Bedingungen nicht erfüllt, müssen weitere Datensätze analysiert werden und eine Einordnung erneut durchgeführt werden (vgl. Abbildung 5.14- Teilschritt h). Anpassungen für die Optimierung der Datensatzwahl sind:

- Erweitern des Umfangs der verfügbaren Daten
- Anpassen der Struktur des Datensatzes
- Überprüfung der Qualität und Aufbereitung des Datensatzes
- Ändern des Datensatzes in einen Datensatz, der für das zu analysierende *Logische Szenario* geeignet ist

Die Anwendung einer oder mehrerer der vorgeschlagenen Anpassungen erfordert eine erneute Analyse des Datensatzes und das Füllen der Datenstruktur (Schritt zurück zu Teilschritt c).

Teilschritt g - Angenommene Wahrscheinlichkeitsdichte Funktion entspricht Häufigkeitsverteilung.

Entspricht die angegebene Verteilungsfunktion der datenbasierten Verteilung mit der geforderten Genauigkeit von z. B. 98%, wird die ermittelte Verteilungsfunktion für den Parameter für weitere Analyseschritte verwendet. Die Ungenauigkeit muss mit Hilfe von Hypothesentests, wie Kolmogorov-Smirnov und Shapiro-Wilk für Normalvarianz, bestimmt werden.

Teilschritt i - Klassifizierung der auf Szenarioebene gefundenen Szenarien nach Wahrscheinlichkeitsdichtefunktion der Parameter.

Die für den Parameter beobachtete Verteilung wird anschließend mit den im Datensatz identifizierten Szenarien auf der Ebene der Szenarien korreliert. Hierzu wird jedes gefundene Szenario kategorisiert und die Wahrscheinlichkeit des Auftretens dieser Parameterkombination mit Hilfe von Signifikanztests, wie dem statistischen Hypothesentest, ermittelt (vgl. Abbildung 5.14 - Teilschritt i). Dazu wird jedes Szenario der spezifischen Wahrscheinlichkeitsdichtefunktion auf Szenarioebene zugeordnet und es wird ermittelt, ob die betrachteten Szenarien in die gebildete Verteilung passen.

Teilschritt j - Parameterverteilung ist repräsentativ.

Wenn alle auf Szenarioebene gefundenen Situationen einem Signifikanzintervall kleiner als $6 \cdot \sigma$ zugeordnet wurden, wird davon ausgegangen, dass die Parameterverteilung repräsentativ ist (vgl. Abbildung 5.14 - Teilschritt j).

Teilschritt k - Berechnung der Gesamtverteilung des jeweiligen Szenarios auf Basis der ermittelten individuellen Wahrscheinlichkeitsdichtefunktion und Priorisierung.

Das Verfahren geht davon aus, dass die betrachteten Parameter statistisch unabhängig voneinander sind. Unter dieser Annahme wird die Gesamtwahrscheinlichkeit

$w_{Kombiniert}$ über alle Parameter w_i durch Multiplikation der für das *Konkrete Szenario* gespeicherten Einzelwahrscheinlichkeiten aller Szenarioparameter berechnet. Dies wird mathematisch in Gleichung 5.22 dargestellt:

$$w_{Kombiniert} = \prod_{i=1}^n w_i \quad (5.22)$$

Betrachtet man die vollständigen Bereiche der zu berücksichtigenden Parameter, so werden die Wahrscheinlichkeitsdichtefunktionen im mehrdimensional aufgespannten Parameterraum überlagert (vgl. Abbildung 5.14 - Teilschritt k).

Die aufsteigend sortierten Wahrscheinlichkeitswerte für jedes *Konkrete Szenario* werden anschließend nach folgendem Schema in vier Exposure-Stufen eingeordnet:

- **Exposure 1 (E1):** seltenes Auftreten - Summenwahrscheinlichkeit des Auftretens 0,28% - entspricht $4 \cdot \sigma$ Intervall
- **Exposure 2 (E2):** gelegentliches Auftreten - Summenwahrscheinlichkeit des Auftretens 4,28% - entspricht $3 \cdot \sigma$ Intervall
- **Exposure 3 (E3):** häufiges Auftreten - Summenwahrscheinlichkeit des Auftretens 27,18% - entspricht $2 \cdot \sigma$ Intervall
- **Exposure 4 (E4):** ständiges Auftreten - Summenwahrscheinlichkeit des Auftretens 68,26% - entspricht $1 \cdot \sigma$ -Intervall

Das Ergebnis ist eine Auftrittswahrscheinlichkeit und ein Exposure-Level - analog zu ASIL [49] - für ein bestimmtes *Konkretes Szenario*, das sich aus mehreren Parametern zusammensetzt, die für die Priorisierung von *Konkreten Szenarien* herangezogen werden.

Zusammenfassen von Auftrittswahrscheinlichkeit und Unfallschwere zu einer Priorisierung

Die ermittelte Auftrittswahrscheinlichkeit und Unfallschwere werden analog zur ASIL Einstufung [49] eingeordnet (vgl. Abbildung 5.16).

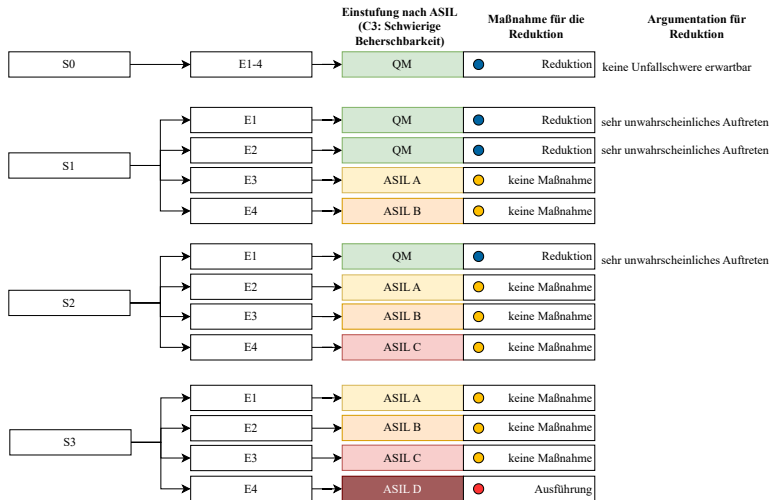


Abbildung 5.16: Ableitung einer Maßnahme aus Auftrittswahrscheinlichkeit und Unfallschwere in Analogie zur ASIL-Klassifikation

Szenarien mit geringen Severity (S0) und geringer Auftrittswahrscheinlichkeit (E1) werden „reduziert“ (vgl. ASIL Klassifikation QM). Szenarien mit einer hohen Severity (S3) und einer hohen Auftrittswahrscheinlichkeit (E4) müssen „ausgeführt“ werden (vgl. ASIL Klassifikation ASIL D). Die anderen *Konkreten Szenarien* mit anderen Kombinationen aus Auftrittswahrscheinlichkeit und Unfallschwere bleiben in diesem Methodenbaustein „ausstehend“ (vgl. ASIL Klassifikation ASIL A, ASIL B, ASIL C). Durch die Reduktion von Szenarien mit einer niedrigen Auftrittswahrscheinlichkeit und keiner Unfallschwere wird der Fokus der Testausführung festgelegt und ggfs. relevante Singularitäten und Edge Cases reduziert. Falls dies nicht gewünscht ist, muss keine Reduktion im Rahmen dieses Methodenbausteins durchgeführt werden.

Durch dieses Auswahlverfahren, das auf einer Kombination von Exposure und Severity basiert, erfolgt eine gezielte Verdichtung der Abtastung in Bereichen

mit hoher Kritikalität und Häufigkeit. Dadurch wird sichergestellt, dass der Parameterraum in sicherheitsrelevanten Regionen detaillierter untersucht wird. Dies erlaubt eine fundiertere Grundlage für nachgelagerte Methodenbausteine, indem eine optimierte Selektion „ausgeführter“ Szenarien gewährleistet wird.

Annahme 8. Methodenbaustein III-2: Methode zur Einordnung der Konkreten Szenarien in Bezug auf Unfallschwere und Auftrittswahrscheinlichkeit. *In diesem Methodenbaustein werden unkritische und unrelevante Szenarien depriorisiert. Die Auftretenshäufigkeit wird auf Basis von Datensätzen berechnet. Es wird davon ausgegangen, dass die Datensätze den Kontext des Logischen Szenarios repräsentativ darstellen. Dies lässt sich zwar durch statistische Tests überprüfen, dennoch muss die Auswahl der Datensätze an das SuT und das geforderte Logische Szenario angepasst werden. Die Annahme zur Verteilung muss zusätzlich mit den Wertebereichen der im Logischen Szenario vorgegebenen variablen Parametern überprüft werden.*

5.4.3 Methodenbaustein III-3: Unsicherheits- und Ähnlichkeitsanalyse

Sind bereits „ausgeführte“ Szenarien vorhanden, müssen die vorliegenden Messdaten für die weitere Bestimmung von ähnlichen Szenarien verwendet werden. Hierzu wird auf Basis von KPIs, welche die Eigenschaften des SuT im Szenarienkontext beschreiben oder auf Basis von Zeitreihen eine Ähnlichkeit zwischen *Konkreten Szenarien* ermittelt werden. Das Ähnlichkeitsmaß (vgl. Kapitel 3) trägt wiederum dazu bei, eine fundierte Aussage über bekannte und unbekannte Szenarien zu liefern und repräsentative *Konkrete Szenarien* anhand der Ähnlichkeit auszuwählen.

Da die meisten berechenbaren Metriken (z.B. KPI-Werte) erst nach der Durchführung des *Konkreten Szenarios* auf Basis von Messungen ermittelt werden können, müssen die Metriken, für die nicht ausgeführten „ausstehenden“ Szenarien, vorhergesagt werden (vgl. Abbildung 5.19). Wird diese Vorhersage mit einer vordefinierten Sicherheit getroffen (Subschritt α), wird der Vorhersage für

die Metrik vertraut. Anschließend werden die Szenarien auf Basis der vorhergesagten und der berechneten Metriken in ähnliche Cluster eingeordnet (Subschritt β).

Subschritt α . Unsicherheitsbasierte Szenarienselektion Im ersten Subschritt dieses Methodenbausteins (vgl. Abbildung 5.19, $\alpha.a$) wird für jede zu betrachtende Metrik ein individuelles Prädiktionsmodell erstellt. Dieses Modell basiert auf den variablen Parametern und Metriken (z.B. KPIs) bereits „ausgeführter“ Szenarien und wird genutzt, um Metriken für „ausstehende“ Szenarien vorherzusagen (vgl. Abbildung 5.17, Subschritte α und β).

Ziel dieses Schritts ist es, das Vorhersagemodell so anzupassen, dass es zuverlässig Metriken für die „ausstehenden“ Szenarien prognostiziert. Hierbei muss der verwendete Algorithmus neben der Vorhersage auch ein quantitatives Maß für die Unsicherheit der Prognose liefern. Entsprechende Regressionsverfahren mit Unsicherheitsabschätzung sind z.B. in [35] beschrieben (vgl. Kapitel 3.3).

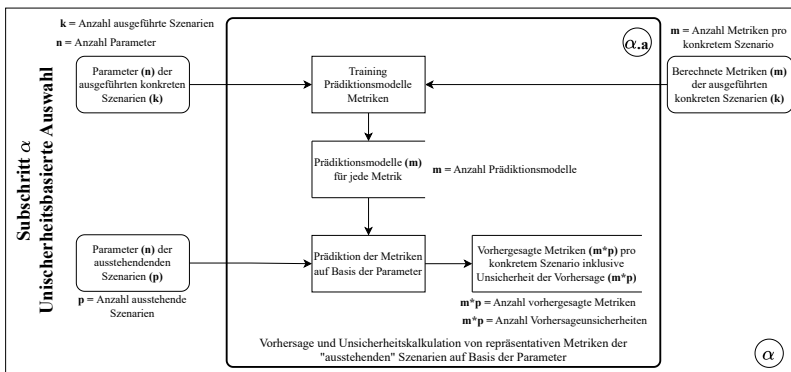


Abbildung 5.17: Datenflussdiagramm zur Prädiktion der Metriken auf Basis der Parameter

Nach dem Training wird das Modell auf alle „ausstehenden“ Szenarien angewendet und berechnet für jedes Szenario eine Vorhersage der Metrik inklusive Unsicherheitsmaß (vgl. Abbildung 5.19, $\alpha.b$ und Abbildung 5.18 – grauer Bereich und Prädiktion des KPI-Werts).

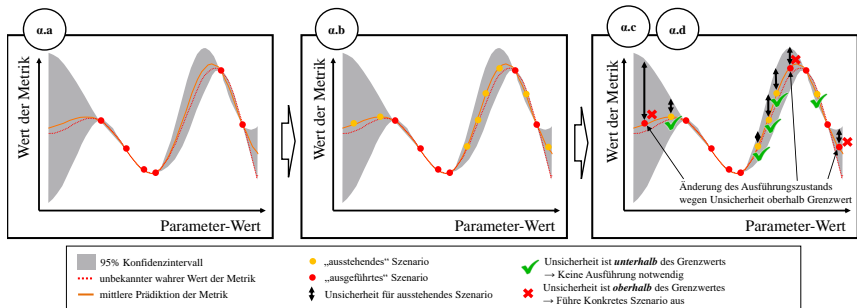


Abbildung 5.18: Grafische Darstellung der Vorgehensweise zur repräsentativen Prädiktion von Metriken in Anlehnung an Abbildung 5.19

Festlegung eines Grenzwerts für Unsicherheit Zur Ableitung eines geeigneten Grenzwerts für die maximal zulässige Unsicherheit werden zunächst die 10% der Szenarien mit der höchsten prognostizierten Unsicherheit selektiert und tatsächlich „ausgeführt“ (vgl. $\alpha.c$ - rotes Kreuz). Die so gewonnenen Messwerte werden mit den zuvor prädizierten Werten verglichen, um die reale Vorhersagegüte zu bestimmen.

Basierend auf diesen Daten wird ein Grenzwert für die maximal zulässige Unsicherheit abgeleitet, bei der die Vorhersagequalität noch als akzeptabel gilt. Dabei werden zusätzlich zwei Szenariencharakteristika berücksichtigt:

- Unfallschwere (Severity): Je höher die potenzielle Gefahr, desto präziser muss die Vorhersage sein.
- Auftretenswahrscheinlichkeit (Exposure): Szenarien mit hoher Eintrittswahrscheinlichkeit erfordern ebenfalls geringere Toleranz gegenüber Unsicherheit.

In besonders sicherheitskritischen oder häufigen Situationen wird folglich ein strengerer Grenzwert angesetzt (vgl. Kapitel 5.4.2).

Iteratives Vorgehen bei Grenzwertüberschreitung Liegen die Unsicherheiten aller prädierten Metriken unterhalb des Grenzwerts, gilt die Vorhersagequalität als hinreichend (vgl. $\alpha.d$), und der nächste Methodenbaustein wird gestartet.

Falls nicht, wird geprüft, ob in den betroffenen (unsicheren) Parameterräumen noch „ausstehende“ Szenarien vorhanden sind (vgl. $\alpha.e$). Falls nicht, muss der betroffene Parameterraum verfeinert werden (Subsampling), indem weitere Szenarien generiert und ausgeführt werden (vgl. $\alpha.f$ und $\alpha.g$).

Dieser Zyklus wiederholt sich so lange, bis die Unsicherheiten aller Vorhersagen innerhalb akzeptabler Grenzen liegen.

Subschritt β . Ähnlichkeitsbasierte Szenarienselektion

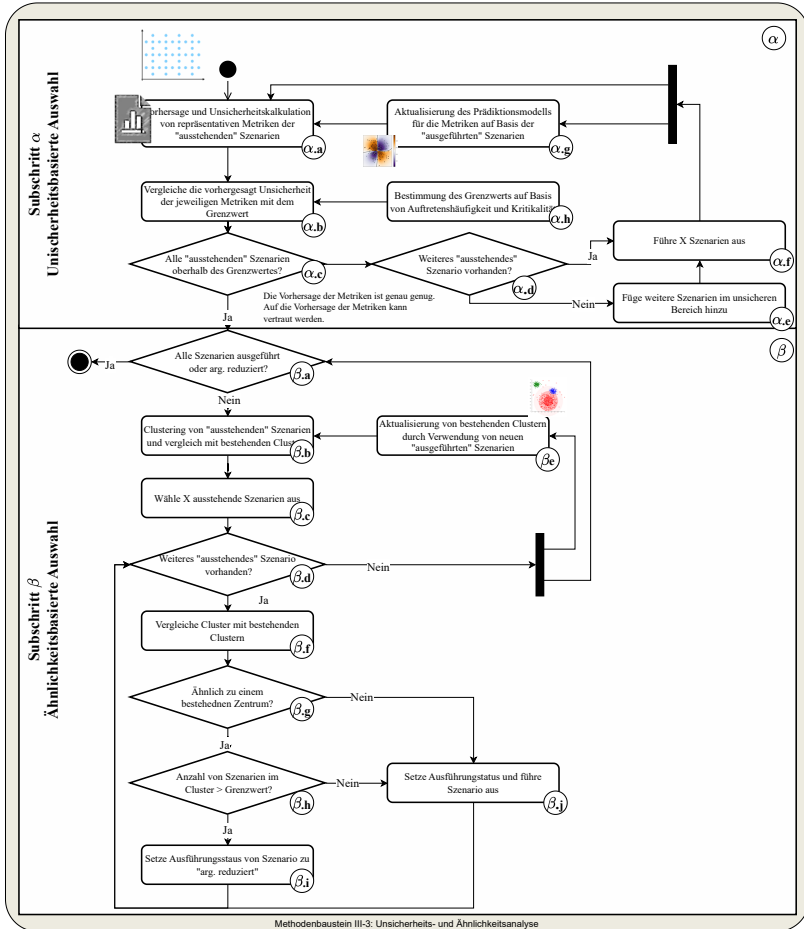
Im Subschritt β werden die „ausstehenden“ Szenarien so lange analysiert, bis alle Szenarien entweder „reduziert“ werden konnten oder „ausgeführt“ wurden (vgl. $\beta.a$ Abbildung 5.19). Sind noch „ausstehende“ Szenarien vorhanden, wird auf Basis der Prädiktion der Metriken und auf Basis der bereits berechneten Metriken, ein auf den Metriken basiertes Clustering durchgeführt (vgl. $\beta.b$ Abbildung 5.19). Das Modell zum Clustering wird auf Basis der Metriken der „ausgeführten“ Szenarien trainiert (vgl. $\beta.e$ Abbildung 5.19). Die Szenarien werden in die Cluster eingeordnet. Hierbei muss eine Clusteringmethode verwendet werden, welche mit einer flexiblen Clusteranzahl umgehen können, da die Anzahl der Cluster vorab unbekannt ist. Ziel ist eine Clusterbildung auf Basis von Metriken, wie Häufigkeit, Unfallschwere, Parametern oder KPI. (siehe Kapitel 3.3). Die K-Means-Elbow-Klassifizierung und die DBScan-Klassifizierung erfüllen diese Anforderung und wurden im Rahmen dieses Methodenbausteins angewendet. Weitere Clusteringansätze finden sich in [10]).

Anschließend wird aus den „ausstehenden“ Szenarien eine Teilmenge zufällig ausgewählt (vgl. $\beta.c$ Abbildung 5.19). Falls dies nicht mehr möglich ist, hat der Methodenbaustein für alle *Konkreten Szenarien* den Ausführungszustand in „reduziert“ oder „ausgeführt“ geändert. Der Selektionsprozess ist beendet. Falls noch Szenarien übrig sind, werden die ausgewählten, „ausstehenden“ Szenarien in die bestehenden Cluster eingeordnet (vgl. $\beta.f$ Abbildung 5.19) und anschließend

mit den bestehenden Clustern der „ausgeführten“ Szenarien verglichen (vgl. *β.g* Abbildung 5.19). Wenn das ausgewählte Szenario einem bekannten Cluster zugeordnet wird und in diesem bereits eine vorbestimmte Anzahl an „ausgeführten“ Szenarien beinhaltet ist, wird das Szenario „reduziert“. Falls dies nicht der Fall ist, wird das Szenario „ausgeführt“. Dieser Prozess wird so lange wiederholt, bis alle Szenarien entsprechend zugeordnet wurden und der Selektionsprozess dadurch beendet wurde

Mit Hilfe dieses Prozesses werden Szenarien auf Basis von prädierten Metriken ausgeschlossen.

Annahme 9. Methodenbaustein III-3: Unsicherheits- und Ähnlichkeitsanalyse. *Der Güte der Vorhersage wird vertraut. Die zu Beginn des Prozesses „ausgeführten“ Szenarien stellen eine ausreichende Trainingsgrundlage dar, indem der Parameterbereich gleichmäßig abgetastet wurde. Es wird außerdem davon ausgegangen, dass Szenarien im selben Cluster, die auf Basis von Metriken bestimmt wurden, ähnlich sind und somit nur eine Stichprobe aus dem Cluster ausgeführt werden muss.*



Methodenbaustein III-3: Unsicherheits- und Ähnlichkeitsanalyse

Abbildung 5.19: Ablaufdiagramm für die Vorhersage und Clustering von Metriken für die Szenarioanalyse

5.5 Abdeckungsaussage zum Logischen Szenario mit Abschätzung des Testaufwands

Nachdem alle *Konkreten Szenarien* in der Gesamtliste durch die Anwendung der vorgestellten Methodenbausteine entweder „reduziert“ oder „ausgeführt“ wurden, müssen abschließend die in den jeweiligen Methodenbausteinen getroffenen Annahmen kritisch hinterfragt und evaluiert werden. Diese Prüfung erfolgt im Rahmen der Qualitätssicherung und Validierung, typischerweise durch Fachexperten, das Qualitätssicherungsteam, sowie das Risikomanagement. Um den Einfluss der Methodenbausteine besser bewerten zu können, wird die Restunsicherheitsbewertung herangezogen. Nur wenn diese Annahmen akzeptiert werden, stehen die ausgeführten *Konkreten Szenarien* repräsentativ für den kompletten Bereich des betrachteten *Logischen Szenarios*. Geht man davon aus, dass für jeden Methodenbaustein eine Risikoprioritätszahl RPZ_k ermittelt wurde (vgl. Kapitel 5.1), lässt sich die Gesamtrestunsicherheit $Restunsicherheit_{Gesamt}$ unter Berücksichtigung aller ermittelten Risikoprioritätszahlen (vgl. Formel 5.23) berechnen:

$$Restunsicherheit_{Gesamt} = \sum_{l=1}^k RPZ_k \quad (5.23)$$

Die Gesamtrestunsicherheit des Selektionsprozesses summiert sich in Abhängigkeit der verschiedenen Restunsicherheiten der einzelnen Methodenbausteine auf (vgl. Abbildung 5.20). Gleichzeitig sinkt die Anzahl der „ausstehenden“ *Konkreten Szenarien*.

Nach Abschluss des Selektionsprozesses muss für die Erstellung einer verlässlichen Freigabeaussage (vgl. Anforderung 4) am Ende des Entwicklungsprozesses jede Annahme hinsichtlich ihrer Validität und Gültigkeit überprüft werden.

Die Überprüfung findet auf Basis der ausgeführten Messpunkte und der zugehörigen Datensätze statt, wobei die getroffenen Annahmen kritisch hinterfragt und

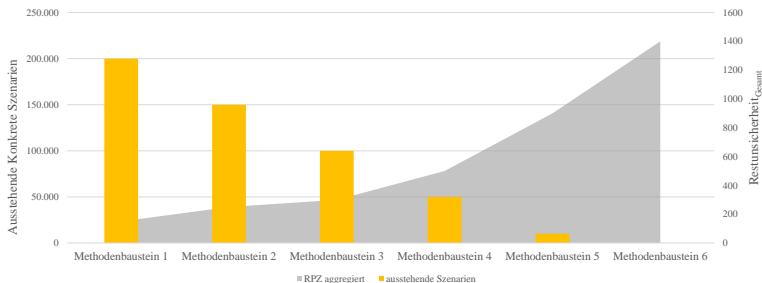


Abbildung 5.20: Beispielhafter Verlauf der Restunsicherheit_{Gesamt} und der Anzahl der „ausstehenden“ Konkreten Szenarien über die verschiedenen Methodenbausteine und Annahmen im Laufe des Selektionsprozesses

validiert werden. Besondere Aufmerksamkeit gilt den betrachteten „reduzierten“ Konkreten Szenarien, die unter Angabe einer Argumentation „reduziert“ wurden (vgl. Anforderung 14). Diese Argumentation, die bereits während der Auswahlprozesse dokumentiert wird, ist essenziell für die spätere Freigabe des Systems unter Test (SuT). Die abschließende Dokumentation, welche alle validierten Annahmen und deren Begründungen umfasst, wird vor der finalen Freigabe geprüft, um die Nachvollziehbarkeit und Regelkonformität sicherzustellen (vgl. Anforderung 13).

Der Selektionsprozess löst das Problem der Abtastung des Logischen Szenarios (Samplingabstand) und der Priorisierung und Auswahl von repräsentativen Konkreten Szenarien auf diesem abgetasteten Bereich. Sowohl die Abtastung als auch die Priorisierung/Auswahl bringen verschiedene Typen von Annahmen mit sich.

- **Annahme zur initialen Samplingstrategie:**

Vor allem in Schritt I des Selektionsprozesses (vgl. Kapitel 5.2) müssen die kontinuierlichen Parameterbereiche abgetastet bzw. diskretisiert werden. Hiermit sind immer Annahmen zur Diskretisierungsschrittweite zu treffen. Ein späteres Subsampling von relevanten Bereichen (vgl. Methode III-2 Kapitel 5.4.3) erlaubt eine zusätzliche Plausibilisierung der Annahme.

- **Annahme zur Vorhersagegenauigkeit von prädierten KPIs:**

Der Grenzwert der geforderten Vorhersagegenauigkeit für KPIs ergibt sich aus der Auftretenswahrscheinlichkeit und der Unfallschwere des *Konkreten Szenarios*. Die Anwendung dieses Grenzwerts basiert auf der Annahme, dass eine Metrik niemals mit absoluter Sicherheit vorhergesagt wird..

- **Annahme zum Zusammenfassen oder Vereinfachen von Parametern:**

Beim Zusammenfassen oder Vereinfachen von Parametern werden auf Basis physikalischer oder mathematischer Zusammenhänge Restriktionen erstellt (vgl. Schritt II Selektionsprozess in Kapitel 5.3). Diese Restriktionen und Annahmen müssen validierte und gültige Annahmen sein, welche keine relevanten System- oder Szenarieneigenschaften reduzieren.

- **Annahme zur Validität der Ausführungsergebnisse:**

Die Validität der Ausführungsinstanz und die damit verbundene Gültigkeit der Messergebnisse, welche als essenzielle Grundlage für den Selektionsprozess (insbesondere für die in Schritt III - vgl. Kapitel 5.4 - vorgestellten Methodenbausteine) dienen, muss im Kontext des betrachteten Szenarios und des betrachteten SuT sichergestellt sein.

- **Annahmen zur Repräsentativität von verwendeten Datensätzen:**

Die analysierten Datensätze müssen repräsentativ für das betrachtete SuT im Szenarienkontext sein, um eine gültige Aussage zur Auftretenshäufigkeit und Unfallschwere zu treffen.

Wird eine Annahme nicht bestätigt, muss die Auswahl der Methodenbausteine oder die Parametrisierung der jeweiligen Methodenbausteine, wie der gewählte Samplingabstand für einen variablen Parameter, geändert werden. Werden alle Annahmen akzeptiert, wurde das *Logische Szenario* auf Basis der subtraktiven Argumentation repräsentativ abgetastet (vgl. Anforderung 8) und für jedes „reduzierte“ *Konkrete Szenario* eine Begründung abgelegt (vgl. Anforderung 14).

Praktische Umsetzung und Überprüfung der Annahmen im Selektionsprozess

Für die praktische Anwendung des Selektionsprozesses ist es erforderlich, die im Rahmen der Methodenbausteine getroffenen Annahmen systematisch zu dokumentieren, zu überprüfen und zu bewerten. Ziel ist es, eine transparente und nachvollziehbare Argumentation für jede getroffene Auswahlentscheidung zu ermöglichen und die Grundlage für eine spätere Freigabeentscheidung abzusichern. Die Umsetzung erfolgt entlang eines dreistufigen Vorgehens:

1. **Dokumentation während der Methodenbausteinausführung:** Alle im Selektionsprozess getroffenen Annahmen – etwa zur Samplingstrategie, zur Parametervereinfachung oder zur KPI-Vorhersage – werden automatisch abgespeichert. Die Dokumentation umfasst den betroffenen Methodenbaustein, den Zeitpunkt der Anwendung, die betroffenen *Konkreten Szenarien* sowie die Begründung der Annahme. Die Begründung erfolgt entweder durch standardisierte Argumentationsbausteine oder ergänzend durch Experteneinschätzungen. Die Informationen dienen als Grundlage für die spätere Bewertung und Validierung.
2. **Validierung nach Testausführung:** Nach der Ausführung der *Konkreten Szenarien* erfolgt die Evaluation der Annahmen durch den Vergleich erwarteter vs. beobachteter KPIs. Dazu wird automatisiert geprüft, ob etwaige Vereinfachungen zu systematischen Abweichungen führen. Zusätzlich erfolgt eine manuelle Prüfung durch Fachpersonal, insbesondere bei sicherheitskritischen Funktionen (vgl. ASIL-Bewertungen in Kapitel 5.5).
3. **Bewertung im Rahmen der Risikobewertung:** Jeder Methodenbaustein wird gemäß dem FMEA-Ansatz hinsichtlich Auftretenswahrscheinlichkeit, Schweregrad und Entdeckbarkeit bewertet. Daraus ergibt sich eine RPZ, welche zur Quantifizierung der Gesamtunsicherheit herangezogen wird (vgl. Kapitel 5.1). Eine zu hohe RPZ einer Annahme führt zu einer Anpassung der Methodenbausteine im Selektionsprozess – etwa durch Anpassung des Samplingabstands oder Austausch eines Methodenbausteins.

Der Selektionsprozess sieht damit explizit vor, dass getroffene Annahmen nicht als statische Vorgaben behandelt werden, sondern im Sinne eines iterativen Entwicklungsprozesses regelmäßig überprüft und angepasst werden können. Die strukturierte und automatisierte Dokumentation sämtlicher Annahmen sowie deren Bewertung im Rahmen des Risikomanagements stellt sicher, dass jede Reduktionsentscheidung auf einer überprüfbaren und nachvollziehbaren Argumentationsgrundlage basiert (vgl. Anforderung 14).

6 Design und Realisierung des Selektionsprozesses

Dieses Kapitel verfolgt das Ziel, den zuvor konzeptionell beschriebenen Selektionsprozess (vgl. Kapitel 5) in eine prototypische technische Umsetzung zu überführen. Dabei steht die Idee einer Bau- und Gebrauchsanleitung im Vordergrund: Es soll nachvollziehbar dargestellt werden, mit welchen Werkzeugen, Datenformaten und algorithmischen Bausteinen der Selektionsprozess praktisch realisiert wurde. Der Fokus liegt auf der Überführung der modularen Methodenbausteine in wiederverwendbare Programmstrukturen, die schrittweise anwendbar sind. Neben der Beschreibung der eingesetzten Technologien wird auch auf die Datenschnittstellen, Parameterabhängigkeiten und typischen Abläufe innerhalb eines Selektionsdurchlaufs eingegangen. Dadurch wird eine Grundlage geschaffen, den Prozess reproduzierbar in bestehende Entwicklungs- und Testumgebungen zu integrieren.

Die Grundlage für das Design des Selektionsprozesses ist ein Datenbankentwurf für die Speicherung des Szenarienkatalogs (vgl. Abbildung 6.1).

Die einzelnen Entitäten des Diagramms (vgl. Abbildung 6.1) werden im Gesamtkontext der Erstellung eines Szenarienkatalogs (vgl. Abbildung 2.13) im Laufe von vier Umsetzungsschritten befüllt. Die Struktur dient zusätzlich zur Protokollierung der Ergebnisse als nachvollziehbare Dokumentation über Reduktion und Ausführung (vgl. Anforderung 13). Die Schritte gliedern sich in den in Kapitel 4 vorgestellten szenarienbasierten Absicherungsprozess ein (vgl. Anforderung 3).

Der Prozess zur Befüllung der Datenstruktur ab Schritt 3 läuft dabei vollautomatisiert ab (vgl. Anforderung 10).

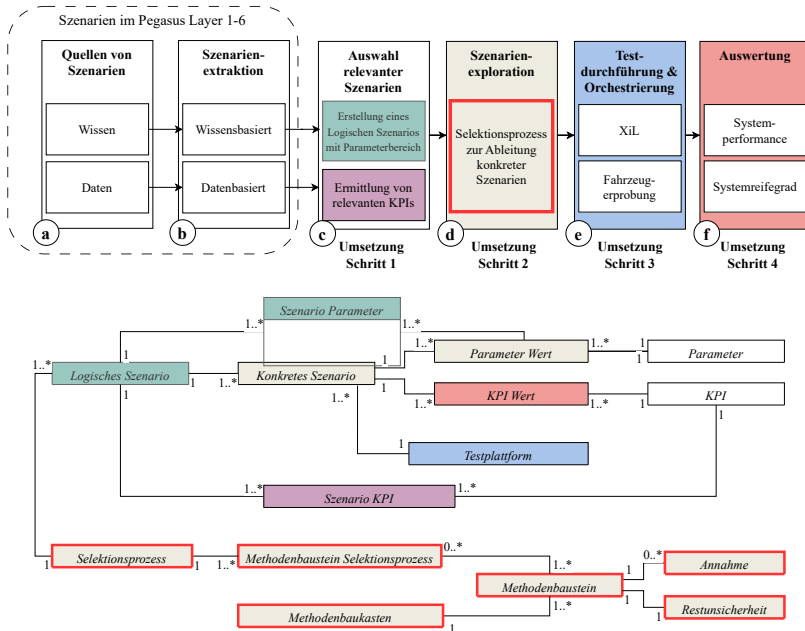


Abbildung 6.1: Einordnung in den Gesamtkontext der Szenarienausführung (angelehnt an Abbildung 2.13) inklusive Zuordnung zur Datenbankstruktur über Farbsemantik (vgl. Abbildung 5.5)

Das aufgebaute Datenbankmodell wurde mit Hilfe einer SQL¹ Datenbank umgesetzt, welche durch ein Backend verwaltet wird. Das Backend bietet eine öffentliche REST-API², über welche sich die relevanten Umsetzungsschritte zur Testplanung, Teststeuerung und Durchführung andocken. Die Kommunikation erfolgt dabei über eine REST-API Schnittstelle. Für die Ausführung der verschiedenen

¹ **SQL**(Structured Query Language) ist eine standardisierte Sprache zur Verwaltung und Abfrage von Daten in relationalen Datenbanksystemen.

² Eine **REST-API** (auch RESTful-API oder RESTful-Web-API genannt) ist eine Programmierschnittstelle (API), die auf den Prinzipien des REST-Architekturstils (Representational State Transfer) basiert. Sie erlaubt eine effiziente und standardisierte Kommunikation zwischen verschiedenen Anwendungen und Systemkomponenten.

Schritte sind folgende Umsetzungsschritte relevant (vgl. Schritte in Abbildung 6.1):

- Umsetzung Schritt 1 - Auswahl relevanter Szenarien
- Umsetzung Schritt 2 - Szenariexploration
- Umsetzung Schritt 3 - Testdurchführung & Orchestrierung
- Umsetzung Schritt 4 - Auswertung

Die jeweiligen Umsetzungsschritte sind dabei modular aufgebaut (vgl. Anforderung 12), sodass sie auf mehreren PCs verteilt ausgeführt werden und nur zentral mit der REST-API-Schnittstelle kommunizieren.

„Umsetzung Schritt 2“ ist dabei der eigentliche Selektionsprozess (vgl. Kapitel 5.1).

Umsetzung Schritt 1.1 - Erstellung eines *Logischen Szenarios* mit Parameterbereich: Die Erstellung des *Logischen Szenarios* ist nicht Teil des Selektionsprozesses, dient aber als Eingang in den Selektionsprozess (vgl. Anforderung 1). Zur weiteren Konkretisierung wird ein *Logisches Szenario* ausgewählt und mit Parameterwerten versehen. Es werden die Entitäten *Logisches Szenario*, Szenario Parameter und Parameter in der Datenbank mit Inhalten befüllt (vgl. Abbildung 5.5). Für jeden variablen Parameter eines *Logischen Szenarios* (vgl. Entität *Logisches Szenario*) wird mindestens ein Parameterbereich (vgl. Entität *Szenario Parameter*) angelegt, welcher auf einen Parameter (vgl. Entität *Parameter*) verlinkt ist. Gleichzeitig wird das *Logische Szenario* in der ausgewählten Testinstanz aufgebaut - mit Hilfe von OpenDRIVE und OpenSCENARIO (vgl. Kapitel 2.4.1). Der Schritt der Auswahl und des Aufbaus des *Logischen Szenarios* erfolgt im Rahmen dieser Ausarbeitung manuell, da der Fokus auf der Auswahl der repräsentativen *Konkreten Szenarien* liegt. Arbeiten nach Braun et al. [24] und Ries et al. [92] zeigen Ansätze zur Identifikation von *Logischen Szenarien* (vgl. Kapitel 3).

Umsetzung Schritt 1.2 - Ermittlung von relevanten KPIs: Für das *Logische Szenario* werden repräsentative KPIs im System, Szenarien und Testinstanzkontext ausgewählt (vgl. Kapitel 2.4.2). Hierbei wurde der Ansatz zur Bestimmung relevanter KPIs für ein *Logisches Szenario* (vgl. Kapitel 4.2.2) angewendet und sowohl der Systemkontext, als auch das *Logisches Szenario*, als auch die Testinstanz berücksichtigt.

Umsetzung Schritt 2 - Szenariexploration mit Hilfe des Selektionsprozess: Dieser Schritt ist der Kern des Selektionsprozesses und damit dieser wissenschaftlichen Ausarbeitung. Hier werden die *Logischen Szenarien* in *Konkrete Szenarien* mit einem Ausführungs- oder Reduktionsauftrag überführt. Beginnend mit der Erstellung der Gesamtliste an Kombinationen (vgl. subtraktiver Ansatz in Kapitel 5.1) und der anschließenden Reduktion, werden hier im Rahmen der vorgestellten drei Schritte (vgl. Kapitel 5.1) verschiedene Methodenbausteine aneinandergereiht und somit relevante *Konkrete Szenarien* für die Ausführung identifiziert. Die Auswahl der eingesetzten Methodenbausteine erfolgt dabei nicht beliebig, sondern basiert auf einer fundierten Abwägung der jeweiligen Stärken und Einsatzbereiche. Ziel ist es, durch die gezielte Kombination unterschiedlicher Methodenbausteine ein möglichst ausgewogenes Verhältnis zwischen Abdeckungsgrad und Reduktionspotenzial zu erzielen. Dabei spielen sowohl theoretische Überlegungen aus Kapitel 5 als auch praktische Aspekte wie Rechenaufwand, Datenverfügbarkeit und Robustheit der Verfahren eine zentrale Rolle. In der Implementierung wurde zudem darauf geachtet, dass sich neue Methodenbausteine mit minimalem Aufwand integrieren lassen. Dies wird durch das konsequente Festhalten an der definierten Schnittstelle (vgl. Abbildung 6.2) ermöglicht, wodurch eine hohe Flexibilität und Erweiterbarkeit des Selektionsprozesses gewährleistet ist. Die konkrete Auswahl und Anordnung der Bausteine in einem Selektionslauf erfolgt aktuell manuell, kann perspektivisch aber durch eine automatisierte, kontextabhängige Selektion ergänzt werden. Die vorgestellten Methodenbausteine lassen sich auf viele andere *Logische Szenarien* anpassen und übertragen. Prinzipiell wird hierbei zwischen zwei Ausprägungen des Methodenbausteins unterschieden. Generierende Methodenbausteine (vgl. Abbildung 6.2 - GenerationMethod) erzeugen neue *Konkrete Szenarien* mit dem Ausführungszustand „Entscheidung ausstehend“.

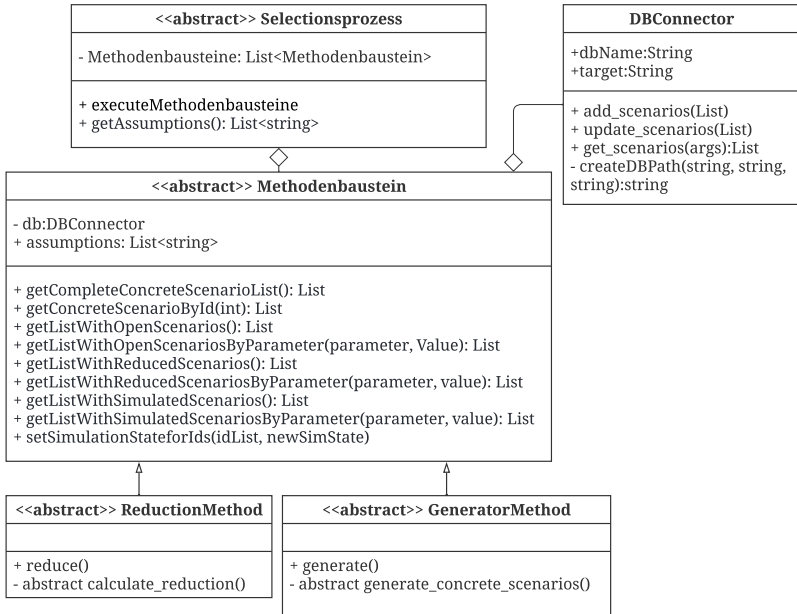


Abbildung 6.2: Klassendiagramm der Kernfunktionalität eines Methodenbausteins mit seinen Schnittstellen zur Datenbank und zum Selektionsprozess

Diese Methodenbausteine kommen vor allem in Schritt I (vgl. Kapitel 5.2) zum Einsatz. Reduzierende Methodenbausteine verändern den Ausführungszustand eines *Konkreten Szenarios* (vgl. Abbildung 6.2 - *ReductionMethod*). Diese kommen in Schritt II (vgl. Kapitel 5.3) und Schritt III (vgl. Kapitel 5.4) des Selektionsprozesses zum Einsatz. Beide abstrakte Klassen erben vom Methodenbaustein, welcher wiederum Teil des Selektionsprozesses ist.

Die Gesamtliste der *Konkreten Szenarien* wird in der Datenbank gespeichert (vgl. Entität *Konkretes Szenario*). Der Ausführungszustand wird durch die einzelnen Methodenbausteine verändert.

Die genauen Implementierungen der in Kapitel 5 vorgestellten Methodenbausteine unterscheiden sich von Baustein zu Baustein. Je nach Methodenbaustein wurden

unterschiedliche, bereits vorhandene Python-Pakete verwendet (vgl. Abbildung 6.3).

Methodenbaustein	Verwendete Packages										
	Pandas	Numpy	FastDTW	euclidean	sklearn - DBSCAN	sklearn - RBF	sklearn KMeans	sklearn - Metrics	sklearn - Random Forest Regressor	sklearn - GPR	
<i>M1-1</i> : Äquidistante kombinatorische Erstellung	X	X		X							
<i>M1-2</i> : Kombinatorische Erstellung auf Basis einer Verteilungsfunktion	X	X									
<i>M1-3</i> : Stochastische Erstellung	X	X									
<i>M2-1</i> : Einschränkung des Parameterraums auf Basis von logischen und physikalischen Zusammenhängen – Restriktion	X	X		X				X			
<i>M2-2</i> : Reduktion und Zusammenfassung der Parameter auf Basis von logischen und physikalischen Zusammenhängen	X	X		X				X			
<i>M3-1</i> : Erste Sampling Strategien	X	X		X							
<i>M3-2</i> : Methode zur Einordnung der Konkreten Szenarien in Bezug auf Kritikalität und Auftrittswahrscheinlichkeit	X	X		X	X						
<i>M3-3</i> : Unsicherheits- und Ähnlichkeitsanalyse	X	X	X	X	X	X	X	X	X	X	X

Abbildung 6.3: Verwendete Bibliotheken für die Implementierung des jeweiligen Methodenbausteins als Matrix Darstellung

Jeder Methodenbaustein hat eine Liste an *Konkreten Szenarien* und eine Visualisierung des Parameterraums in Form eines Diagramms als Ergebnis.

Zusätzliche Informationen über die Anzahl der „reduzierten“ und „auszuführenden“ Szenarien werden vom Selektionsprozess ausgegeben. Alle „reduzierten“ Szenarien werden am Ende mit einer Begründung in der Gesamtliste an *Konkreten Szenarien* abgelegt.

Umsetzung Schritt 3.1 - Testorchestrierung: Alle Szenarien mit dem Ausführungszustand „Ausführung erforderlich “ werden einer Testinstanz zugeordnet. Der Selektionsprozess ist dabei unabhängig von der Testinstanz, an welcher das Szenario ausgeführt wird. Dennoch ist die Orchestrierung des Szenarios zu einer Testinstanz ein wichtiger Schritt im Gesamtkontext der Szenarienausführung. Es muss dabei gewährleistet sein, dass die ausgewählten KPIs an der Testinstanz

valide berechnet werden, genauso muss der Systemkontext zur Auswahl der Testinstanz passen. King et al. [52] liefert Ansätze, wie diese Orchestrierung effizient vorgenommen werden kann.

Umsetzung Schritt 3.2 - Testdurchführung: Das Ziel der Ausführung ist, die im OpenDrive und OpenScenario spezifizierten *Konkreten Szenarien* im Systemkontext auszuführen und relevante Zeitreihen abzuspeichern. Dies erfolgt in dieser Promotion durch die Verwendung des Testautomatisierungstools ModelDesk von der Firma dSpace. An der ausgewählten Testinstanz wird das *Konkrete Szenario* und seine dazugehörigen Parameter zusammen mit dem SuT und weiteren benötigten Umgebungsmodellen (vgl. Kapitel 2.3.1) ausgeführt. Die Simulationsumgebung zeichnet Zeitreihen wie der Geschwindigkeitsverlauf auf und speichert diese für die spätere Auswertung.

Umsetzung Schritt 4 - Auswertung: Basierend auf den aufgezeichneten Zeitreihen wird die Ausführung bewertet. Hierzu wird zum einen evaluiert, ob das spezifizierte Szenario dem ausgeführten Szenario entspricht und ob sich das SuT in dieser Situation nach den Anforderungen verhalten hat. Hierzu werden KPIs zur Beschreibung verwendet (vgl. Kapitel 2.4.2). Das verwendete Backend bietet Zugang zur Ansicht aller *Konkreten Szenarien* und deren jeweiligen Ausführungszustände, Parameterwerte und vorhandenen KPIs.

Die Berechnung und Speicherung der KPIs und Bewertungskriterien in der Datenbank erfolgt automatisiert, sobald ein neues Ausführungsergebnis vorliegt. Die Ergebnisse werden anschließend wieder im Selektionsprozess verwendet.

7 Evaluation am Beispiel eines Logischen Szenarios

7.1 Einführung eines beispielhaften Logischen Szenarios

Der Selektionsprozess wird anhand eines *Logischen Szenarios* mit einem Ego-Fahrzeug und einem VRU (Fahrradfahrer) gezeigt. Betrachtet wird eine innerstädtische Kreuzungssituation mit statischen und dynamischen Objekten. Das *Logische Szenario* besteht neben den zwei dynamischen Verkehrsteilnehmern aus zwei parkenden Fahrzeugen am Straßenrand, welche die Sichtbarkeit der Sensoren auf das Ego-Fahrzeug einschränken. Getestet wird dabei die Kollisionsfreiheit des Systems und damit verbunden die Funktionalität des Notbremssystems bei verschiedenen Parameterkombinationen des *Logischen Szenarios* (vgl. Abbildung 7.1). Im Szenario wird ein Anhalten des Ego-Fahrzeugs am Zebrastreifen erwartet, sofern sich ein VRU im relevanten Bereich befindet. Dieses beispielhafte *Logische Szenario* wurde im Projekt VV-Methoden als repräsentatives Szenario ausgewählt. Sowohl die OpenDRIVE als auch die OpenSCENARIO - Datei wurde vom VV-Methoden Projekt zur Verfügung gestellt. Bei der Kombination von Zebrastreifen und Fahrradfahrer handelt es sich um einen verkehrswidrigen Fall, welcher allerdings dennoch im Projekt als harte Vorgabe gesetzt wurde. Im beispielhaften *Logischen Szenario* werden drei Parameter variiert:

- **Parameter 1** - Ego Fahrzeuggeschwindigkeit zu Beginn der Ausführung v_{ego_0} [km/h]. Dieser Parameter beschreibt die Geschwindigkeit zum Start der Ausführung, welche das Fahrzeug so lange einhält, bis eine Bremsung

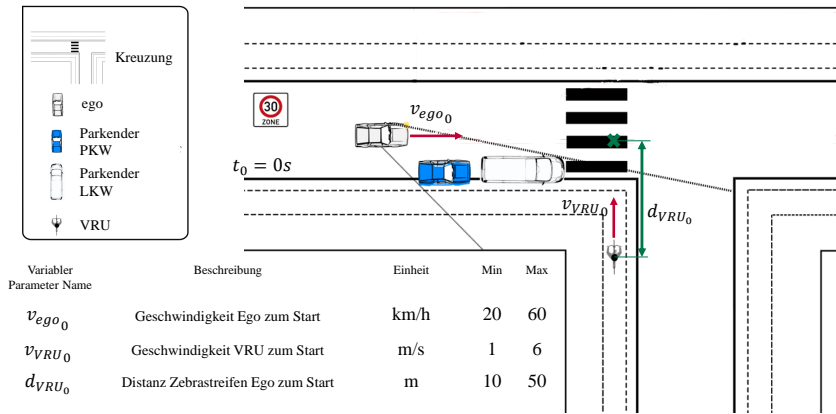


Abbildung 7.1: Aufbau Logisches Szenario - Beispiel

vom entsprechenden SuT getriggert wird. Der im *Logischen Szenario* spezifizierte Parameterbereich umfasst die Werte 20-60 km/h.

- **Parameter 2** - Geschwindigkeit des VRU v_{VRU_0} [m/s]. Der VRU bewegt sich mit einer gleichförmigen Geschwindigkeit über den Zebrastreifen. Er reagiert in diesem Szenario nicht auf das Fahrzeug (Worst-Case Betrachtung). Der im *Logischen Szenario* spezifizierte Parameterbereich umfasst die Werte 1-6 m/s.
- **Parameter 3** - Abstand des VRU zum Zebrastreifen zu Beginn des Szenarios d_{VRU_0} [m]. Der im *Logischen Szenario* spezifizierte Parameterbereich umfasst die Werte 10-50 m.

Das Konzept ist auf weitere Parameter erweiterbar und funktioniert auch für mehr als drei Parameter. Bei bis zu drei Parametern lässt sich der Parameterraum mit Hilfe eines 3D-Plots visualisieren.

7.2 Anwendung der Methodenbausteine auf ein beispielhaftes Logisches Szenario

Die drei vorgestellten Schritte des Selektionsprozesses und die dazugehörigen Methodenbausteine werden zur Evaluation herangezogen (vgl. Kapitel 5).

Zunächst wird die Datenstruktur mit dem *Logischen Szenario* und den ermittelten Parametern gefüllt (vgl. Kapitel 4.2.2). Im Kontext des SuT (Notbremsfunktionalität) wurden drei relevante KPIs im Kontext des *Logischen Szenarios* und des SuT festgelegt:

- **KPI 1** - Abstand des SuT zum VRU zum Zeitpunkt des Betretens der Straße $d_{relevant}[m]$ - Abstand des VRUs zum Ego-Fahrzeug beim Betreten des Zebrastreifens durch den VRU. Der VRU wird zu diesem Zeitpunkt zum ersten Mal durch das Ego-Fahrzeug wahrgenommen und das Notbremsssystem kann eine Bremsung initialisieren.
- **KPI 2** - Minimale Distanz zwischen SuT und VRU $d_{ego \rightarrow VRU}[m]$ - Minimaler euklidischer Abstand zwischen den Ego und VRU während des Szenarios.
- **KPI 3** - Dauer des Bremsvorgangs des SuT $t_{brems}[s]$ - Zeitdauer in der durch das Notbremsystem eine Bremsung durchgeführt wird.

Annahme: Diese KPIs stehen repräsentativ für das betrachtete *Logische Szenario* im Kontext des betrachteten Notbremsassistenten. Unterscheiden sich die KPIs, ändert sich die Systemreaktion deutlich und es handelt sich aus Sicht des SuTs um unterschiedliche Situationen.

Diese aufgeführten KPIs werden auf Basis von Zeitreihen berechnet (vgl. Kapitel 2.4.2). Die KPIs wurden dem *Logischen Szenario* zugeordnet und in der Datenstruktur (vgl. Abbildung 5.5) abgelegt.

Diese Datenstruktur bietet die Basis für die Anwendung des Selektionsprozesses auf das *Logische Szenario*. Für das *Logische Szenario* wurden aus Kapitel 5 fünf

Methodenbausteine ausgewählt und in den Selektionsprozess eingeordnet (vgl. Abbildung 7.2).

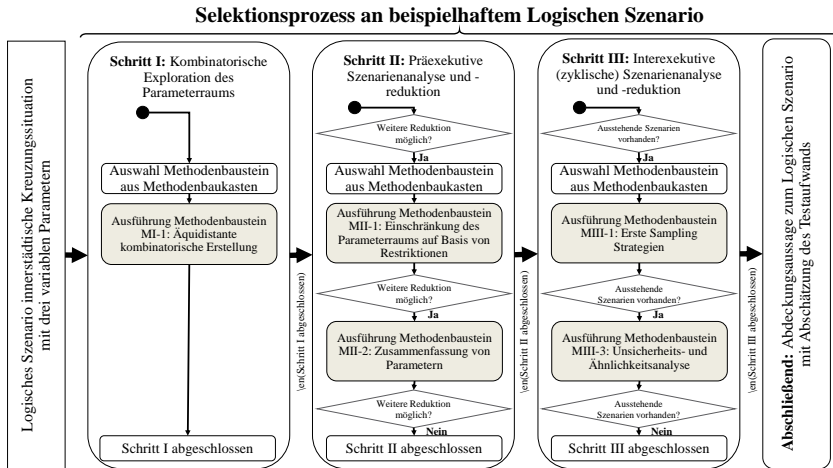


Abbildung 7.2: Ausgewählte Methodenbausteine für das beispielhafte *Logische Szenario* nach Abbildung 5.3

Für Schritt I des Selektionsprozesses wurde „Methodenbaustein I-1: Äquidistante kombinatorische Erstellung“ (vgl. Kapitel 5.2.1) ausgewählt. In Schritt I muss nur ein Methodenbaustein ausgewählt werden (exklusive Auswahl) (vgl. Aufbau Selektionsprozess in Kapitel 5.2).

Für Schritt II des Selektionsprozesses wurde „Methodenbaustein II-1: Einschränkung des Parameterraums auf Basis von Restriktionen“ (vgl. Kapitel 5.3.1) und „Methodenbaustein II-2: Zusammenfassung von Parametern“ (vgl. Kapitel 5.3.2) ausgewählt. In Schritt II wurden für das beispielhafte *Logische Szenario* alle vorgestellten Methodenbausteine ausgewählt.

Für Schritt III des Selektionsprozesses wurde „Methodenbaustein III-1: Erste Sampling Strategien“ (vgl. Kapitel 5.4.1) und „Methodenbaustein III-3: Unsicherheits- und Ähnlichkeitsanalyse“ (vgl. Kapitel 5.4.3) ausgewählt. In Schritt III wurden

für das beispielhafte *Logische Szenario* zwei der drei vorgestellten Methodenbausteine ausgewählt.

Auf die Begründung der Auswahl wird jeweils näher in der Ausführung und Evaluation des Methodenbausteins eingegangen.

7.2.1 Methodenbaustein I-1: Äquidistante kombinatorische Erstellung

Es werden in diesem Methodenbaustein (MI-1) des Schrittes I noch keine Szenarien ausgeführt. Da das ausgewählte *Logische Szenario* drei zu variierende Parameter hat und für die Parameter keine Verteilungsfunktion vorliegt (Ausschluss von anderen Methodenbausteinen in Schritt I), wird die Gesamtliste initial vollständig linear generiert (vgl. Kapitel 5.2.1). Es ist kein Resampling in späteren Schritten erforderlich (vgl. Resampling Kapitel 5.2). Zunächst muss pro Parameter eine Diskretisierungsschrittweite festgelegt werden.

- Für die Fahrzeuggeschwindigkeit zu Beginn des Szenarios (Parameter 1 - v_{ego_0}) wurde eine Diskretisierungsschrittweite von 0,5 km/h gewählt. Im Parameterbereich zwischen 20 km/h und 60 km/h entstehen somit 81 Stützstellen.
- Für die Geschwindigkeit des VRUs zu Beginn des Szenarios (Parameter 2 - v_{VRU_0}) wurde eine Diskretisierungsschrittweite von 0,2 m/s gewählt. Im Parameterbereich zwischen 1 m/s und 6 m/s entstehen somit 26 Stützstellen.
- Für den Abstand des VRUs zu Beginn des Szenarios zum Zebrastreifen (Parameter 3 - d_{VRU_0}) wurde eine Diskretisierungsschrittweite von 0,5 m gewählt. Im Parameterbereich zwischen 10 m und 50 m entstehen somit 75 Stützstellen.

Die Diskretisierungsschrittweiten basieren dabei auf der Implementierung des SuT (Notbremsassistenten) und der funktionspezifischen Verarbeitung der Signale. Zusätzlich unterstützen externe Datensätze [22] die Dynamik des *Logischen Szenarios* und den Umfang der Diskretisierungsschrittweiten zu erfassen.

Durch die Kombinatorik entstehen somit (nach Formel 5.5) 157.950 *Konkrete Szenarien*, welche in der Gesamtliste mit dem Ausführungszustand „ausstehend“ abgelegt werden. Die Aufgabe dieses Methodenbausteins ist die Generierung. Die

Konkreten Szenarien sind nach der Generierung gleichmäßig über den kompletten Parameterraum verteilt (vgl. Abbildung 7.3).

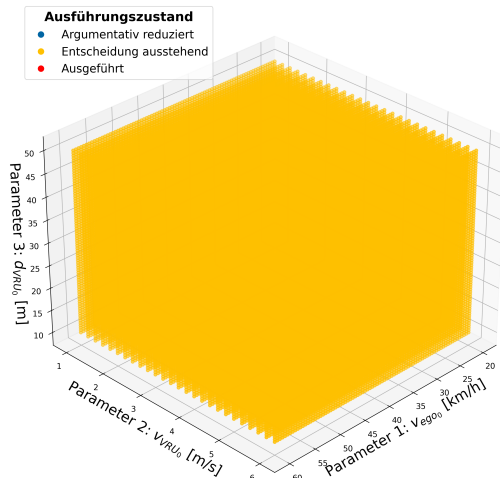


Abbildung 7.3: 3D Plot der *Konkreten Szenarien* im Parameterraum als Ergebnis des ersten Methodenbausteins - 157.950 „ausstehende“ *Konkrete Szenarien*

Ermittlung der Risikoprioritätszahl für den Methodenbaustein (vgl. Methodik Kapitel 5.1):

- **Annahmen und Singularität**

Die Diskretisierungsschrittweite des Methodenbausteins ist auf 0,5 km/h, 0,2 m/s und 0,5 m festgelegt. Die Analyse der Implementierung des dSpace Notbremsassistenten hat ergeben, dass die im Methodenbaustein gewählte Diskretisierungsschrittweite unterhalb der Diskretisierung der Implementierung der Funktion liegt (E3). Die Diskretisierungsschrittweite kann zu Singularitäten und Funktionsfehlern führen, ohne eine feinere Abtastung

(Resampling) werden diese nicht entdeckt (C8). Wird ein Szenario übersehen, bei dem ein Fußgänger plötzlich die Straße betritt, kann dies im realen Betrieb zu einem Unfall mit schwerer Verletzungsfolge führen (S9) - *resultierende aggregierte RPZ 216*.

- **Schätzung der Vorhersagegenauigkeit**

Es werden keine Vorhersagen getroffen (E1) (C1). Ein schwerer Personenschaden ist möglich (S9) - *resultierende aggregierte RPZ 9*.

- **Nicht-deterministisches Verhalten**

Der Methodenbaustein verhält sich deterministisch durch äquidistantes Sampling (E1) (C1). Ein schwerer Personenschaden ist möglich (S9) - *resultierende aggregierte RPZ 9*

- *Ausführliche Begründung der Risikoprioritätszahl siehe Anhang A* -

Unter Anwendung der Formel zur Zusammenfassung der RPZ mit $w_i = 1$ (vgl. Formel 5.3) lässt sich die Gesamtrestunsicherheit des Methodenbausteins zu 216,4 berechnen.



Evaluation des Methodenbausteins:

Eine Evaluation dieses Methodenbausteins ist nicht erforderlich, da keine *Konkreten Szenarien* „reduziert“ oder „ausgeführt“ wurden. Bei einer Ausführungszeit

von 90s pro *Konkretem Szenario*¹ ergibt sich für die vollständige Ausführung aller *Konkreten Szenarien* der Gesamtliste eine Gesamtausführungszeit von 164,5 Tagen. Die Gesamtliste enthält 157.950 *Konkrete Szenarien*, welche in die Datenbank abgespeichert werden.

Für die Anwendung des Methodenbausteins ergeben sich folgende Vor- und Nachteile:

Vorteile:

- Vollständigkeitsargumentation auf Basis der Gesamtliste subtraktiv möglich, da alle² Kombinationen als Basis genutzt werden (vgl. Anforderung 14).
- Annahmen über den Parameterraum werden durch die Festlegung der Diskretisierungsschrittweite explizit dokumentiert (vgl. Anforderung 13).
- Der Methodenbaustein wird vollständig automatisiert ausgeführt und ist daher auch auf andere *Logische Szenarien* und Baureihen ohne Aufwand übertragbar (vgl. Anforderung 5, Anforderung 7 und Anforderung 10).

Nachteile:

- Anzahl der Parameter und Diskretisierungsschrittweite führt zu vielen *Konkreten Szenarien* (vgl. Berechnung Formel 5.6). Dieser Methodenbaustein ist nur sinnvoll, wenn die Anzahl an erstellten *Konkreten Szenarien* $< 10^6$, da sonst eine Bearbeitung und Verarbeitung der erstellten Gesamtliste nicht mehr möglich ist (Widerspruch zu Anforderung 8).

¹ Durchschnittlicher Wert an dSpace SIL Simulator

² Basierend auf der Auswahl relevanter Parameter sowie deren Diskretisierungsschrittweite

7.2.2 Methodenbaustein II-1: Einschränkung des Parameterraums auf Basis von Restriktionen

Die gewählten variablen Parameter stehen in einer geometrischen Abhängigkeit, welche es erlaubt, unplausible Situationen auszuschließen.

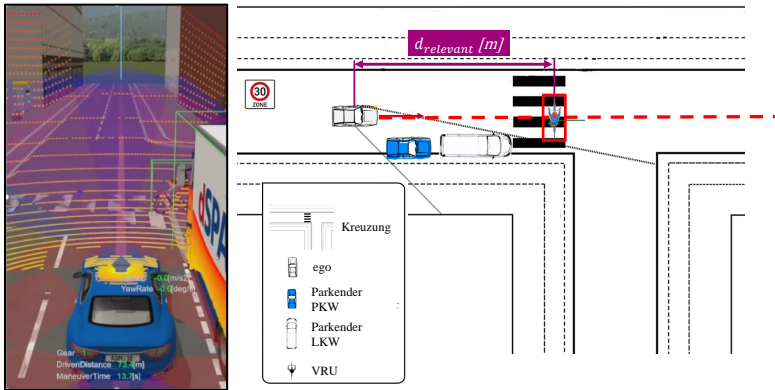


Abbildung 7.4: Relevanter Abstand $d_{relevant} [m]$ ab dem das Ego den VRU wahrnimmt. VRU betritt den Zebrastreifen und kommt aus der Verdeckung hervor

Zur Anwendung des vorgestellten Methodenbausteins (MII-1) wurden durch Analyse des Systems und des *Logischen Szenarios* zwei Restriktionen formuliert (vgl. Kapitel 5.3.1):

- **Restriktion 1:** Der initiale Abstand des Ego zum Zebrastreifen muss ausreichend sein, um mit einer Vollbremsung zum Stehen zu kommen.

$$\underbrace{d_{ego}(0)}_{\text{initialer Abstand}} > \underbrace{\frac{1}{2} \cdot \frac{(v_{ego})^2}{a_{brems}}}_{\text{minimaler Bremsweg}} \quad (7.1)$$

- **Restriktion 2:** Der Abstand zwischen Ego und VRU beim Betreten des Zebrastreifens von VRU muss größer als 0m sein, damit der VRU vor dem Ego-Fahrzeug die Straße betritt (vgl. Abbildung 7.4).

$$\underbrace{d_{ego}(0) - v_{ego} \cdot \frac{d_{VRU}}{v_{VRU}}}_{\text{Abstand des EGO zum Zebra beim Betreten der Straße des VRUs}} > 0m \quad (7.2)$$

Abstand des EGO zum Zebra beim Betreten der Straße des VRUs

Die Parameter a_{brems} und $d_{egoU}(0)$ sind dabei konstante Parameter (vgl. Kapitel 2.4.1) und werden im Rahmen des *Logischen Szenarios* nicht variiert.

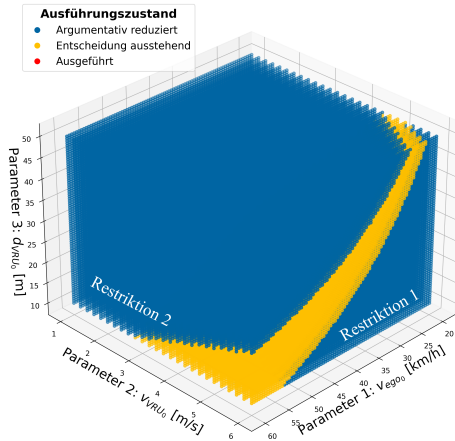


Abbildung 7.5: 3D Plot des Szenarienraums über die Parameter mit den 137.056 „reduzierten“ Konkreten Szenarien, welche auf Basis der beiden Restriktionen ausgeschlossen wurden

Werden beide Restriktionen auf die in Methodenbaustein 1 erstellte Gesamtliste angewendet, werden bereits 137.056 Konkrete Szenarien auf Basis der beiden manuell formulierten Restriktionen „reduziert“. Der Prozess der Anwendung der Restriktionen erfolgt automatisch durch eine sequentielle Abarbeitung der Szenarienliste. Als Argumentation wird jeweils die betroffene Restriktion an das

Konkrete Szenario angehängt. Es verbleiben noch 20.894 *Konkrete Szenarien*. Der Ausführungsaufwand wird damit von 164,5 Tagen auf 21,7 Tage reduziert.

Ermittlung der Risikoprioritätszahl für den Methodenbaustein (vgl. Methodik Kapitel 5.1):

- **Annahmen und Singularität**

Die getroffenen Restriktionen beinhalten potentiell falsche logische Zusammenhänge und führen somit zu einem Ausschluss von eigentlich relevanten Situationen (E7). Die Restriktionen lassen sich im Nachgang durch Ausführung von einigen ausgeschlossenen Szenarien validieren (C2). Wird ein Szenario übersehen, bei dem ein Fußgänger plötzlich die Straße betritt, kann dies im realen Betrieb zu einem Unfall mit schwerer Verletzungsfolge führen (S9) - *resultierende aggregierte RPZ 126*.

- **Schätzung der Vorhersagegenauigkeit**

Es werden keine Vorhersagen getroffen (E1) (C1). Ein schwerer Personenschaden ist möglich (S9) - *resultierende aggregierte RPZ 9*.

- **Nicht-deterministisches Verhalten**

Deterministisches Verhalten des Methodenbausteins durch äquidistantes Sampling (E1) (C1). Ein schwerer Personenschaden ist möglich (S9) - *resultierende aggregierte RPZ 9*.

- Ausführliche Begründung der Risikoprioritätszahl siehe Anhang A -

Unter Anwendung der Formel zur Zusammenfassung der RPZ mit $w_i = 1$ (vgl. Formel 5.3) lässt sich die Gesamtrestunsicherheit des Methodenbausteins zu 126,6 berechnen.



Evaluation des Methodenbausteins:

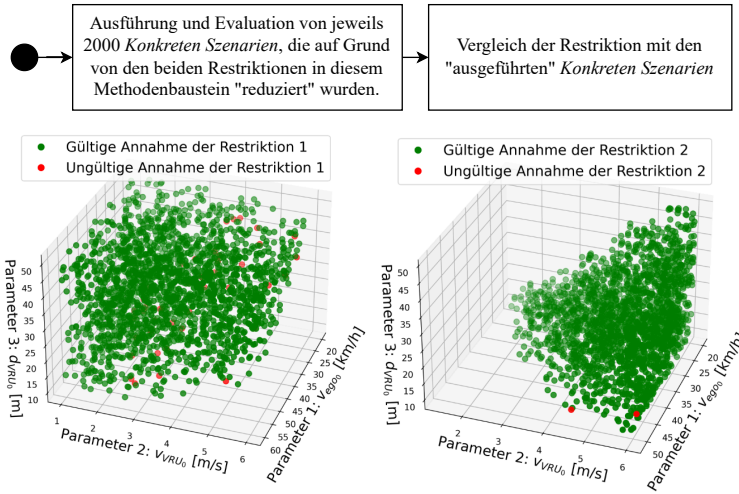


Abbildung 7.6: Evaluation des Erfüllungsgrades der beiden Restriktion und Vergleich zwischen Messung und Berechnung bei einer Stichprobe von 2000 zufälligen Konkreten Szenarien. Restriktion 1 (links - 46 ungültige Annahmen) und Restriktion 2 (rechts - 2 ungültige Annahmen)

Zusätzlich wurde die mathematische Berechnung des Relevanten Abstands $d_{relevant}$ in der Restriktion mit dem tatsächlichen wahren Wert verglichen und die absolute Abweichung berechnet (vgl. Abbildung 7.7).

Das Häufigkeitsdiagramm zeigt, dass die Abweichung maximal $1m$ beträgt (vgl. Abbildung 7.7). Die Annahme zur Restriktion wurde damit valide getroffen, da die vorhergesagte Szenarien-Logik der tatsächlichen Szenarien-Logik entspricht. Lediglich wenige *Konkrete Szenarien* (Restriktion 1 - 46 ungültige Annahmen und Restriktion 2 - 2 ungültige Annahmen - rote Szenarien in Abbildung 7.6) wurden aufgrund einer Fehlannahme „reduziert“. Durch eine Anpassung der Restriktion - indem ein Toleranzbereich eingefügt wurde - wurden die fehlerhafte Annahme vermieden. Restriktion 2 überprüft nun bspw. nicht mehr auf $> 0m$ (vgl. Formel

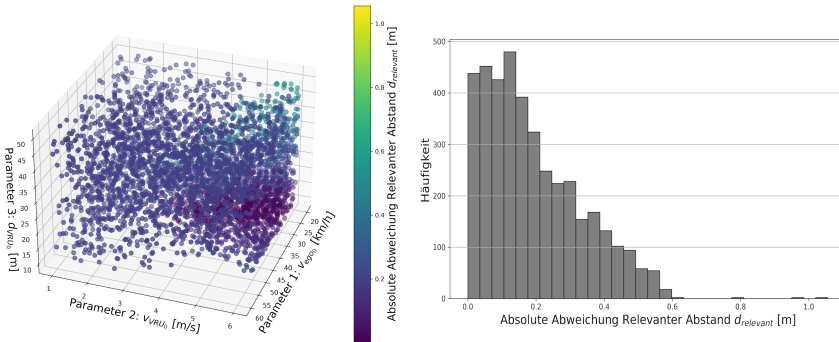


Abbildung 7.7: Absolute Abweichung zwischen wahrer Wert aus Ausführung und durch Restriktion angenommenen Wert im Parameterraum (links) und als Häufigkeitsverteilung (rechts). Durchschnittliche Abweichung: -0.1569m Standardabweichung: 0.1806m

7.2), sondern auf $> -1\text{m}$ und erlaubt somit einen breiteren Gültigkeitsbereich des Szenarios (es werden weniger *Konkrete Szenarien* in diesem Methodenbaustein „reduziert“).

Für die Anwendung des Methodenbausteins ergeben sich folgende Vor- und Nachteile:

Vorteile:

- Abarbeitung der Restriktionen auf Gesamtliste parallelisierbar (vgl. Anforderung 7).
- Reduktion ohne Ausführungsaufwand sorgt direkt für eine Reduktion des Testaufwands (vgl. Anforderung 9).
- Sind die Restriktionen einmal formuliert, wird dieser Methodenbaustein automatisiert ausgeführt (vgl. Anforderung 10).
- Reduktion ist durch die Ergänzung der Reduktionsargumentation nachvollziehbar (vgl. Anforderung 13 und Anforderung 14).

Nachteile:

- Formulierung der Restriktionen ist stark abhängig vom *Logischen Szenario* und dem Kontext des SuTs und den damit verbundenen Restriktionen (Widerspruch zu Anforderung 12).
- Plausibilisierung der Restriktionen ist mit Ausführungsaufwand verbunden (Widerspruch zu Anforderung 5).

7.2.3 Methodenbaustein II-2: Zusammenfassung von Parametern

Alle zu variierenden Parameter im betrachteten *Logischen Szenario* beziehen sich auf geometrische Größen der beiden Verkehrsteilnehmer (Ego und VRU). In diesem Methodenbaustein (MII-2) werden die voneinander abhängigen geometrischen Parameter zusammengefasst, um so nur noch die für die Fahrsituation relevanten Einflüsse zu betrachten (vgl. Methodenbaustein in Kapitel 5.3.2). In dem gewählten innerstädtischen Szenario wird der Abstand des VRUs zum Zebrastreifen, sowie die Geschwindigkeit des VRUs zu einem Zeitpunkt $t_{crossVRU}$ zusammengefasst, an dem der VRU den Zebrastreifen betritt.

Durch die Zusammenfassung von zwei Parametern auf einen Parameter ergibt sich somit theoretisch eine Reduktion der *Konkreten Szenarien* auf zehn Prozent (vgl. Formel 5.11). Zur Reduktion der Unsicherheit wurde in der Anwendung des Methodenbausteins das vollständige Reduktionspotential nicht ausgenutzt.

Der Methodenbaustein gliedert sich in mehrere Subschritte (vgl. Kapitel 5.3.2).

Subschritt α : Formulierung und Anwendung der Korrelation

Unter der Annahme, dass sich der VRU im spezifizierten Szenario mit einer konstanten Geschwindigkeit v_{VRU_0} auf den Zebrastreifen zubewegt, werden die Parameter v_{VRU_0} und d_{VRU_0} zu einem abgeleiteten Parameter $t_{crossVRU}$ zusammengefasst (vgl. Formel 7.3 und 7.4). Dieser Hilfsparameter bestimmt den Zeitpunkt, $t_{crossVRU}$ an dem der VRU den Zebrastreifen betritt, in Abhängigkeit von v_{VRU_0} und d_{VRU_0} .

$$\vec{p} = \begin{pmatrix} v_{ego_0} \\ v_{VRU_0} \\ d_{VRU_0} \end{pmatrix} = \begin{pmatrix} v_{ego_0} \\ v_{VRU_0} \\ d_{VRU_0} \end{pmatrix} \quad (7.3)$$

$$\vec{p} = \begin{pmatrix} p_1 \\ p_{\text{derived}} \end{pmatrix} = \begin{pmatrix} v_{\text{ego}_0} \\ t_{\text{crossVRU}} \end{pmatrix} = \begin{pmatrix} v_{\text{ego}} \\ \frac{dv_{\text{VRU}_0}}{v_{\text{VRU}_0}} \end{pmatrix} \quad (7.4)$$

Subschritt β : Ermittlung der neuen Diskretisierungsschrittweite auf der Basis der Fehlerfortpflanzung

Die neue Diskretisierungsschrittweite für den zusammengefassten Parameter t_{crossVRU} berechnet sich mit Hilfe der Fehlerfortpflanzung für Division (vgl. Formel 7.5) und auf Basis der initial gewählten Diskretisierungsschrittweite für die beiden variablen Parameter.

$$\vec{p} = \begin{pmatrix} p_1 \\ p_{\text{derived}} \end{pmatrix} = \begin{pmatrix} v_{\text{ego}} \\ t_{\text{crossVRU}} \end{pmatrix} = \begin{pmatrix} v_{\text{ego}} \\ \frac{dv_{\text{VRU}}(0)}{v_{\text{VRU}}} \end{pmatrix} \quad (7.5)$$

Auf Basis der relativen Fehlerfortpflanzung³ ergibt sich die neue Diskretisierungsschrittweite zu 0,5s in einem Bereich von $t_{\text{crossVRU}}^{\text{max}} = 10.75\text{s}$ und $t_{\text{crossVRU}}^{\text{min}} = 1.66\text{s}$.

Subschritt γ : Clustering der Daten auf Basis der vorgegeben Diskretisierungsschrittweite

Auf Basis der berechneten Min-Werte und Max-Werte lässt sich nun die Anzahl der Cluster ermitteln. Hierzu wurde das KMeans Clustering Verfahren angewendet (vgl. Kapitel 3.3.1). Es ergeben sich 19 Cluster.

Subschritt δ : Auswahl von Repräsentanten auf Basis des Clusterings

Der nicht-reduzierte variable Parameter wird nun zusammen mit den gebildeten Clustern in Gruppen (386 Gruppen) eingeordnet. Pro Gruppe werden anschließend zufallsbasiert maximal 40 *Konkrete Szenarien* ausgewählt, welche das Cluster repräsentieren. Hierbei wird die Anzahl absichtlich hoch angesetzt, damit sich die Gesamtrestunsicherheit des Methodenbausteins reduziert.

³ Zur Berechnung wurde die Python Library `uncertainties` verwendet

Man erkennt, dass nun einige *Konkrete Szenarien* dasselbe Label für die aggregierte relevante Zeit aufweisen und somit „reduziert“ werden.

Insgesamt gelingt es, die Anzahl der „ausstehenden“ *Konkreten Szenarien* auf 13.381 zu „reduzieren“. Es sind nun in Summe 144.569 *Konkrete Szenarien*, welche durch die Ausführung der beiden Methodenbausteine „reduziert“ wurden. Dabei wurden in diesem Methodenbaustein 7.513 *Konkrete Szenarien* „reduziert“ (vgl. Abbildung 7.8). Dies ergibt eine Reduktion des Testaufwands gegenüber der Vollausführung von weiteren 7,8 Tagen.

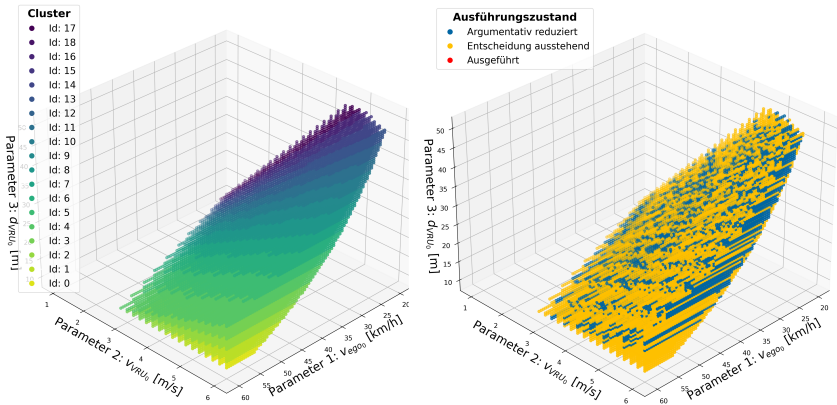


Abbildung 7.8: Cluster zu Ordnung über den Raum der betrachteten „ausstehenden“ Szenarien (links) und Ergebnis 3D Plot der *Konkreten Szenarien* im Parameterraum nach der Ausführung des dritten Methodenbausteins (rechts)

Ermittlung der Risikoprioritätszahl für den Methodenbaustein (vgl. Methodik Kapitel 5.1):

- **Annahmen und Singularität**

Der Ablauf des Szenarios ist möglicherweise fehlerhaft erfasst und führen somit zu falschen Zusammenfassung der Parameter. Es werden Situationen

ausgeschlossen die eigentlich relevant sind. Allerdings ist das ausgenutzte Reduktionspotential dieses Methodenbausteins sehr gering, da pro Cluster 50 Szenarien selektiert wurden (E8). Eine gute Entdeckbarkeit ist garantiert. Es werden nur wenige *Konkrete Szenarien* ausgeschlossen und pro Cluster *Konkrete Szenarien* selektiert (C1). Wird ein Szenario übersehen, bei dem ein Fußgänger plötzlich die Straße betritt, kann dies im realen Betrieb zu einem Unfall mit schwerer Verletzungsfolge führen (S9) - *resultierende aggregierte RPZ 72*.

- **Schätzung der Vorhersagegenauigkeit**

Es werden keine Vorhersagen getroffen (E1) (C1). Ein schwerer Personenschaden ist möglich (S9) - *resultierende aggregierte RPZ 9*.

- **Nicht-deterministisches Verhalten**

DER Methodenbaustein verhält sich deterministisch durch äquidistantes Sampling (E1) (C1). Ein schwerer Personenschaden ist möglich (S9) - *resultierende aggregierte RPZ 9*.

- Ausführliche Begründung der Risikoprioritätszahl siehe Anhang A -

Unter Anwendung der Formel zur Zusammenfassung der RPZ mit $w_i = 1$ (vgl. Formel 5.3) lässt sich die Gesamtrestunsicherheit des Methodenbausteins zu 73,1 berechnen.



Evaluation des Methodenbausteins

Die Zusammenfassung basiert dabei auf der Annahme, dass der Zeitpunkt im Testfall, zu dem der VRU den Zebrastrifen betritt, mit dem erwähnten kinematischen

Zusammenhang berechnet werden kann. Zur Evaluation dieser Annahme wurde ein Testset von *Konkreten Szenarien*, welche eigentlich durch diesen Methodenbaustein „reduziert“ wurden, ausgeführt. Anschließend wurden die Ergebnisse der Ausführung mit der auf Basis von kinematischen Beziehungen ermittelten Zusammenhänge verglichen (vgl. Abbildung 7.9).

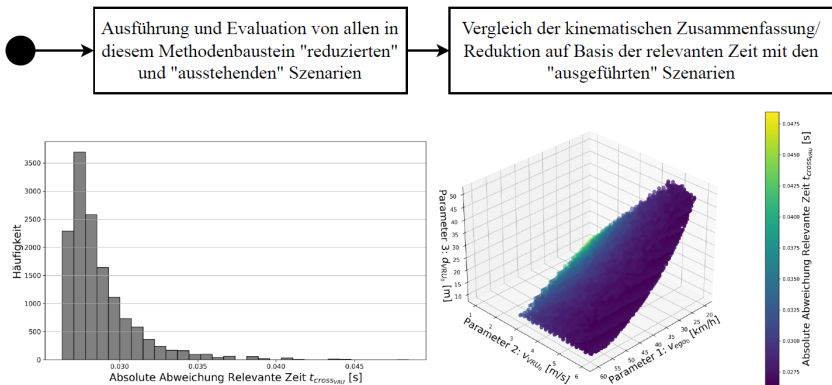


Abbildung 7.9: Evaluation der absoluten Abweichung der relevanten Zeit mit Hilfe von ausgeführten *Konkreten Szenarien*

Die Evaluation zeigt, dass der gewählte Zusammenhang zwischen den beiden variablen Parametern eine hohe Korrelation mit den tatsächlichen Messergebnissen aufweist und dadurch als valide bestätigt wird. Es ergibt sich eine Abweichung von maximal 1,6%.

Für die Anwendung des Methodenbausteins ergeben sich folgende Vor- und Nachteile:

Vorteile:

- Der mehrdimensionale Parameterraum wird „reduziert“, ohne *Konkrete Szenarien* „ausführen“ zu müssen (vgl. Anforderung 9).
- Sind die Zusammenfassungen formuliert, wird dieser Methodenbaustein automatisiert ausgeführt (vgl. Anforderung 10).

- Die ursprüngliche Anzahl der Parameter des *Logischen Szenarios* ändert sich durch die Auswahl der repräsentativen Szenarien nicht (vgl. Anforderung 7).
- Die Reduktion ist durch die Ergänzung der Reduktionsargumentation nachvollziehbar (vgl. Anforderung 13 und Anforderung 14).

Nachteile:

- Die Formulierung der Restriktionen und das damit verbundene Reduktionspotential ist stark abhängig vom *Logischen Szenario* und dem Kontext des SuTs (Widerspruch zu Anforderung 11).
- Eine Plausibilisierung der Restriktionen nur mit Ausführungsaufwand möglich (Widerspruch zu Anforderung 5).
- Der Fehlerfortpflanzungsansatz basiert auf den initial gewählten Diskretisierungsschrittweiten und ist daher nicht vollständig unabhängig (Widerspruch zu Anforderung 12).

7.2.4 Methodenbaustein III-1: Erste Sampling Strategien

Nachdem nun bereits ein Großteil der *Konkreten Szenarien* auf Basis von logischen und physikalischen Zusammenhängen „reduziert“ werden konnte, müssen nun erste Erkenntnisse über das *Logische Szenario* gesammelt werden. Hierzu werden zufallsbasiert 4.000 *Konkrete Szenarien* aus den „ausstehenden“ Szenarien ausgewählt, „ausgeführt“ und die Ergebnisse in die Datenbank gespeichert (vgl. Methodenbaustein MIII-1 in Kapitel 5.4.1). Die Ergebnisse der Ausführung sind die Basis für die Anwendung weiterer Methodenbausteine. Für die Ausführung von 4.000 *Konkreten Szenarien* wurde eine Zeit von 25h benötigt. Es verbleiben noch 9.381 *Konkrete Szenarien* auf dem Ausführungszustand „ausstehend“ (vgl. Abbildung 7.10).

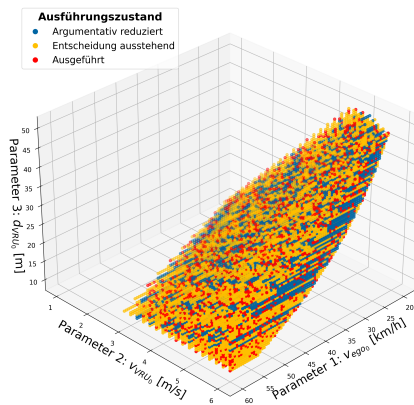


Abbildung 7.10: Ergebnis 3D Plot der *Konkreten Szenarien* im Parameterraum nach der Ausführung des vierten Methodenbausteins

Ermittlung der Risikoprioritätszahl für den Methodenbaustein (vgl. Methodik Kapitel 5.1):

- **Annahmen und Singularität**

Es werden keine Annahmen getroffen (E1) (C1). Wird ein Szenario übersehen, bei dem ein Fußgänger plötzlich die Straße betritt, kann dies im realen Betrieb zu einem Unfall mit schwerer Verletzungsfolge führen (S9) - *resultierende aggregierte RPZ 9.*

- **Schätzung der Vorhersagegenauigkeit**

Es werden keine Vorhersagen getroffen (E1) (C1). Ein schwerer Personenschaden ist möglich (S9) - *resultierende aggregierte RPZ 9.*

- **Nicht-deterministisches Verhalten**

Keine Reduktion im Methodenbaustein (E1) (C1). Ein schwerer Personenschaden ist möglich (S9) - *resultierende aggregierte RPZ 9.*

- Ausführliche Begründung der Risikoprioritätszahl siehe Anhang A -

Unter Anwendung der Formel zur Zusammenfassung der RPZ mit $w_i = 1$ (vgl. Formel 5.3) lässt sich die Gesamtrestunsicherheit des Methodenbausteins zu 15,6 berechnen.



Evaluation des Methodenbausteins:

Eine Evaluation ist nicht erforderlich, da nur *Konkrete Szenarien* „ausgeführt“ und keine Szenarien „reduziert“ wurden. Es sind keine Annahmen mit diesem Schritt verbunden.

Für die Anwendung des Methodenbausteins ergeben sich folgende Vor- und Nachteile:

Vorteile:

- Keine Parametrisierung oder Expertenwissen benötigt. Wiederverwendbarkeit und Automatisierung ohne Probleme möglich (vgl. Anforderung 5, Anforderung 6, Anforderung 7 und Anforderung 10).

Nachteile:

- Direkte Ausführung ohne zyklische Auswertung, ob die Ausführung auch sinnvoll ist (Widerspruch zu Anforderung 9).

7.2.5 Methodenbaustein III-3: Unsicherheits- und Ähnlichkeitsanalyse

Im letzten Methodenbaustein (MIII-3) werden über die übrigen „ausstehenden“ *Konkreten Szenarien* eine Entscheidung getroffen. Hierzu wird basierend auf den bereits „ausgeführten“ *Konkreten Szenarien* und deren berechneten KPIs ein Trainingsmodell für KPIs aufgebaut. Dieses wird verwendet, um für die „ausstehenden“ *Konkreten Szenarien* eine KPI-Vorhersage zu ermitteln (vgl. Methodenbaustein III-3 in Kapitel 5.4.3). Im konkreten Beispiel werden drei Vorhersagemodelle für die Vorhersage der drei vorgestellten KPIs ermittelt (vgl. KPI $d_{relevant}$ - Abbildung 7.11, KPI $d_{ego \rightarrow VRU}$ - Abbildung 7.12 und KPI t_{brems} - Abbildung 7.13).

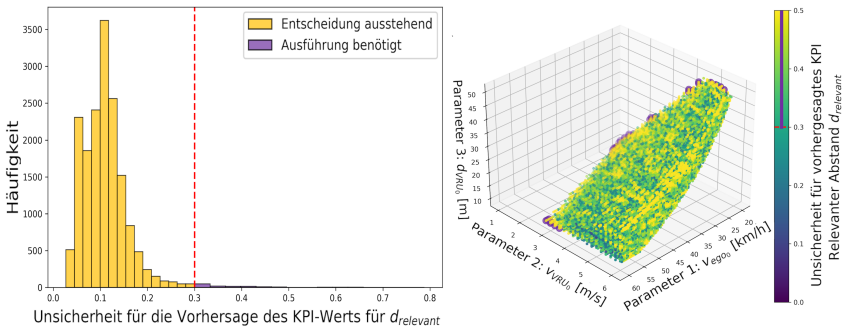


Abbildung 7.11: Unsicherheitsvorhersage für KPI $d_{relevant}$ im Parameterraum (rechts) und als Häufigkeitsdiagramm (links) (Zusätzliche Ausführung von 133 Konkreten Szenarien)

Anschließend wird die Unsicherheitsangabe genutzt, um die Vorhersage zu bewerten und ggfs. *Konkrete Szenarien* an unsicheren Stellen „auszuführen“. Im Beispiel des innerstädtischen Szenarios wurden so besonders in den Randbereichen und Bereichen mit ausgeführten *Konkreten Szenarien* weitere Stützstellen „ausgeführt“ (vgl. Abbildung 7.11, Abbildung 7.12 und Abbildung 7.13 - lila markierte *Konkrete Szenarien* in Häufigkeitsverteilung (links) und Parameterraum (rechts)).

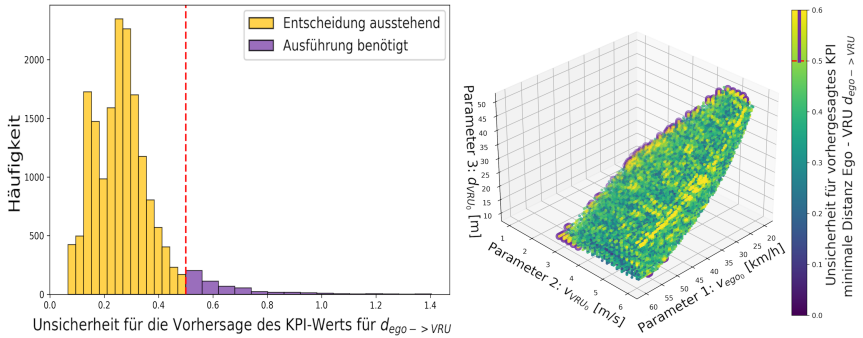


Abbildung 7.12: Unsicherheitsvorhersage für KPI $d_{ego} \rightarrow VRU$ im Parameterraum (rechts) und als Häufigkeitsdiagramm (links) (Zusätzliche Ausführung von 538 Konkreten Szenarien)

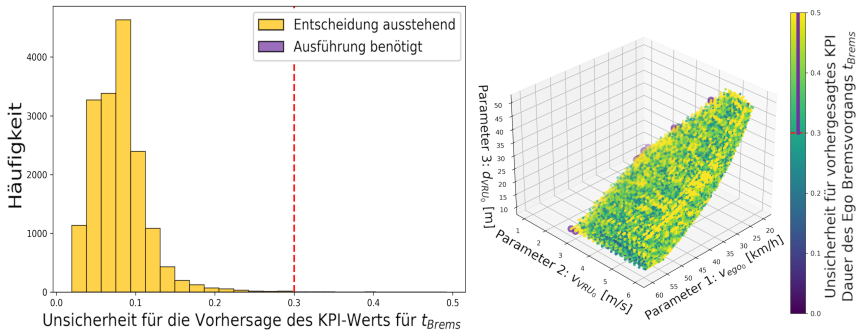


Abbildung 7.13: Unsicherheitsvorhersage für KPI $t_{brem.s}$ im Parameterraum (rechts) und als Häufigkeitsdiagramm (links) (Zusätzliche Ausführung von 35 Konkreten Szenarien)

Insgesamt wurden auf Grund der Unsicherheit in der KPI-Vorhersage zusätzlich 538 Konkrete Szenarien „ausgeführt“, und die KPIs basierend auf den Messergebnissen berechnet. Dabei ist zu beachten, dass sich die zur Analyse herangezogenen Szenarien zwischen den betrachteten KPIs teilweise überlappen. Die Summe der in Abbildung 7.11, Abbildung 7.12 und Abbildung 7.13 genannten Ausführungen (133, 538 bzw. 35) entspricht daher nicht der Gesamtanzahl der tatsächlich zusätzlich simulierten Konkreten Szenarien, sondern gibt die Anzahl pro KPI an, wobei einzelne Szenarien mehrfach berücksichtigt sind. Die Unsicherheit bei

der Vorhersage des KPIs - minimale Distanz zwischen Ego und VRU - weist dabei eine signifikante Abweichung auf. Anschließend wurden die Vorhersagemodelle aktualisiert und die Unsicherheiten und KPI-Vorhersagen systematisch erneut analysiert (vgl. Formel 7.6). Für die ausgewählten drei KPIs wurde eine Gauß-Prozessregression mit einem Matern-Kernel als Kovarianzfunktion ausgewählt, da die Gauß-Prozessregression den niedrigsten Fehlerscore der betrachteten Regressionsmodelle aufweist (vgl. Kapitel 3.3.2).

$$\underbrace{\begin{pmatrix} v_{\text{ego}} \\ v_{\text{VRU}} \\ d_{\text{VRU}}(0) \end{pmatrix}}_{\text{Eingang Vorhersagemodelle}} \rightarrow \underbrace{\begin{pmatrix} \text{Modell}_1 \\ \text{Modell}_2 \\ \text{Modell}_3 \end{pmatrix}}_{\text{Vorhersagemodelle}} \rightarrow \underbrace{\begin{pmatrix} d_{\text{relevant}}(\text{KPI}_1) + \text{Unsich.} \\ d_{\text{ego} \rightarrow \text{VRU}}(\text{KPI}_2) + \text{Unsich.} \\ t_{\text{brems}}(\text{KPI}_3) + \text{Unsich.} \end{pmatrix}}_{\text{Ausgang Vorhersagemodelle}} \quad (7.6)$$

Das neue Vorhersagemodell liefert nun Werte für die Unsicherheit aller drei KPIs, die kleiner sind als die akzeptierte Restunsicherheit (Unsicherheit muss bei dem verwendeten Prädiktionsmodell kleiner als 0.3 bzw. 0.5 sein.⁴). Es ist keine weitere Iteration erforderlich.

Am Ende des ersten Subschritts des Methodenbausteins bleiben 8.843 „ausstehende“ *Konkrete Szenarien* übrig (vgl. Abbildung 7.14), für welche im letzten Subschritt des Methodenbausteins eine Entscheidung getroffen werden muss.

Nun sind für alle *Konkreten Szenarien* der Gesamtliste entweder vorhergesagte oder auf Basis von Messungen berechnete KPIs verfügbar. Allen KPIs wird im Rahmen der zugelassenen Unsicherheit vertraut.

Diese KPIs werden nun dazu verwendet, mit Hilfe eines Clustering-Verfahrens, verschiedene Cluster zu bilden. Die Anzahl der Cluster ist dabei jeweils vorab

⁴ Diese Grenze wurde empirisch auf Basis von Messungen ermittelt und stellt für das ausgewählte KPI-Prädiktionsmodell eine ausreichende Vorhersagegenauigkeit dar.

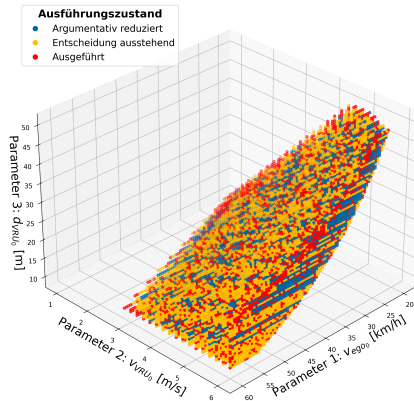


Abbildung 7.14: Ergebnis 3D Plot der *Konkreten Szenarien* im Parameterraum nach der Ausführung des ersten Subschritts des fünften Methodenbausteins

unbekannt und muss sich auf Basis der Varianz der KPIs ergeben. Im ausgewählten innerstädtischen Beispiel wurde die Anzahl und Einordnung der Cluster mit Hilfe der Elbow-Methode und k-Means gebildet. Es ergeben sich 22 Cluster. Pro Cluster werden nun mindestens zehn *Konkrete Szenarien* ausgewählt und auf den Ausführungszustand „Ausführung benötigt“ gesetzt. Da es sich bei diesem Methodenbaustein um den letzten Methodenbaustein handelt, wird die Annahme getroffen, dass alle *Konkrete Szenarien* in einem Cluster ähnlich sind und damit repräsentative Szenarien aus jedem Cluster zufällig ausgewählt werden (hier mind. 10 Szenarien pro Cluster) (vgl. Abbildung 7.15).

Alle „ausstehenden“ *Konkreten Szenarien* werden unter Zuordnung der Cluster Nr. argumentativ „reduziert“. Es werden so zusätzlich weitere 154 *Konkrete Szenarien* „ausgeführt“. In Summe wurden so 153.258 *Konkrete Szenarien* „reduziert“ und 4.692 *Konkrete Szenarien* „ausgeführt“ und somit der komplette Parameterraum abgetastet und argumentativ analysiert (vgl. Abbildung 7.16).

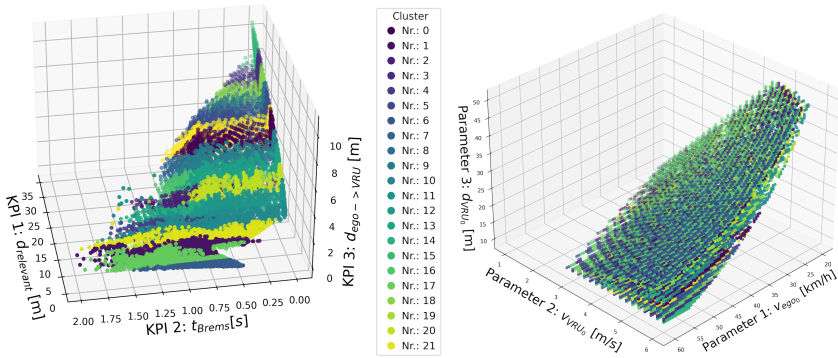


Abbildung 7.15: Cluster gebildet mit k-Means im Parameterraum (rechts) und im KPI-Raum (links)

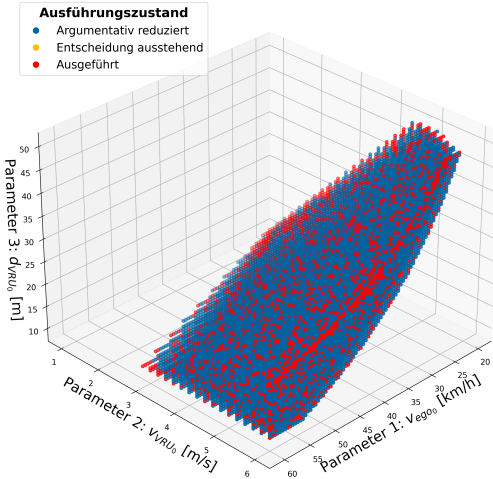


Abbildung 7.16: Gesamtergebnis 3D Plot der Konkreten Szenarien im Parameterraum nach der Ausführung des fünften Methodenbausteins

Ermittlung der Risikoprioritätszahl für den Methodenbaustein (vgl. Methodik Kapitel 5.1):

- **Annahmen und Singularität**

Es wird davon ausgegangen, dass die gewählten KPIs repräsentativ für das Verhalten des SuTs im Szenario stehen (E8). Die Repräsentativität der KPIs im Sut lässt sich nur mit Hilfe der Anforderungen überprüfen (C5). Wird ein Szenario übersehen, bei dem ein Fußgänger plötzlich die Straße betritt, kann dies im realen Betrieb zu einem Unfall mit schwerer Verletzungsfolge führen (S9) - *resultierende aggregierte RPZ 360*.

- **Schätzung der Vorhersagegenauigkeit**

Es muss für jedes „ausstehende“ *Konkrete Szenario* eine Vorhersage getroffen werden. Diese Vorhersage ist mit Unsicherheiten behaftet (E8). Es besteht die Möglichkeit, die getroffenen Vorhersagen zu evaluieren und Fehler in der Vorhersage zu entdecken (C4). Ein schwerer Personenschaden ist möglich (S9) - *resultierende aggregierte RPZ 288*.

- **Nicht-deterministisches Verhalten**

Die Anzahl der Repräsentativitätscluster ist unbekannt. Eine Annahme über die Ähnlichkeit zwischen zwei *Konkreten Szenarien* entscheidet über die Reduktion (E8). Das anschließende Clustering verhält sich nicht deterministisch. Durch die höhere Anzahl an Szenarien pro Cluster wird die Fehlertoleranz verbessert, sodass Fehler besser entdeckt werden (C4). Ein schwerer Personenschaden ist möglich (S9) - *resultierende aggregierte RPZ 288*.

- Ausführliche Begründung der Risikoprioritätszahl siehe Anhang A -

Unter Anwendung der Formel zur Zusammenfassung der RPZ mit $w_i = 1$ (vgl. Formel 5.3) lässt sich die Gesamtrestunsicherheit des Methodenbausteins zu 543,6 berechnen.



Evaluation des Methodenbausteins

Zur Evaluation wurden alle „ausstehenden“ *Konkreten Szenarien* „ausgeführt“ und die Vorhersage der KPI Werte mit dem tatsächlichen, durch Ausführung berechneten, Wert verglichen (vgl. Abbildung 7.17, Abbildung 7.18 und Abbildung 7.19).

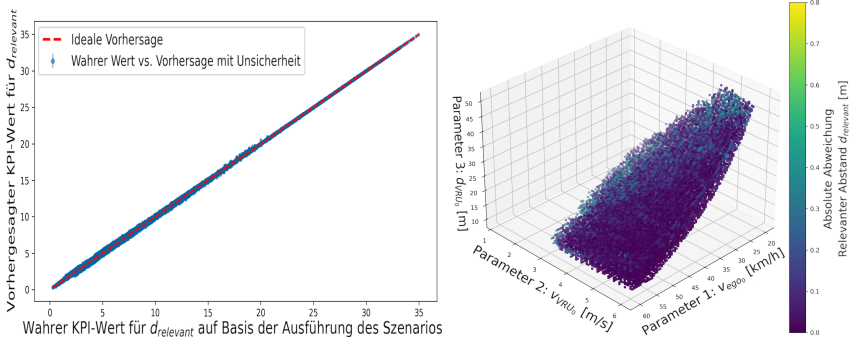


Abbildung 7.17: Gegenüberstellung wahrer KPI Wert mit der Vorhersage auf Basis der Gauß-Prozessregression mit Unsicherheit (links) und der Verteilung des absoluten Fehlers im Parameterraum (rechts) für $d_{relevant}$

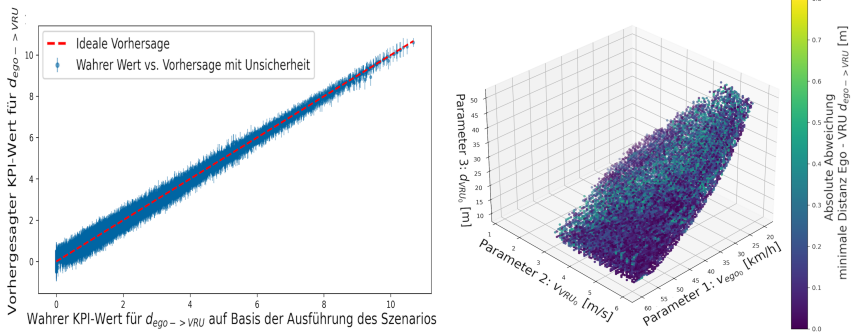


Abbildung 7.18: Gegenüberstellung wahrer KPI Wert mit der Vorhersage auf Basis der Gauß-Prozessregression mit Unsicherheit (links) und der Verteilung des absoluten Fehlers im Parameterraum (rechts) für $d_{ego} \rightarrow VRU$

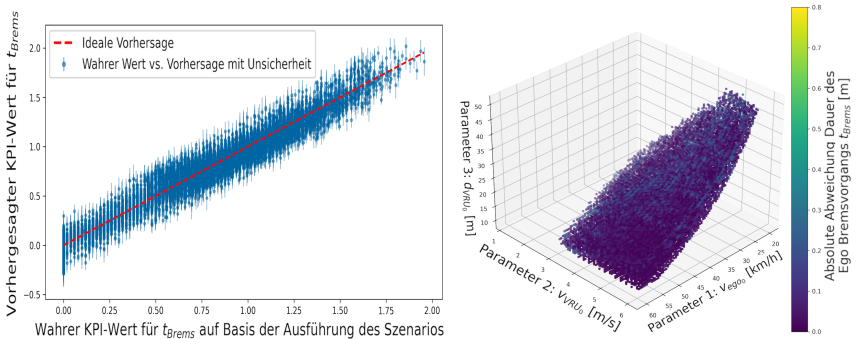


Abbildung 7.19: Gegenüberstellung wahrer KPI Wert mit der Vorhersage auf Basis der Gauß-Prozessregression mit Unsicherheit (links) und der Verteilung des absoluten Fehlers im Parameterraum (rechts) für t_{brems}

Die größten Streuungen und absoluten Fehler ergeben sich dabei beim KPI t_{brems} . Hier streuen die Vorhersagen stark um die ideale Vorhersage. Dies zeigt auch die Evaluation des RMSE⁵ und des mittleren absoluten Fehlers (vgl. Tabelle 7.1).

⁵ Root Mean Squared Error

KPI	RMSE	mittlerer abs. Fehler
$d_{relevant}$	0.1136m	0.0760 m
$d_{ego \rightarrow VRU}$	0.0823m	0.0609 m
t_{brems}	0.1929s	0.1478 s

Tabelle 7.1: Fehler bei der Vorhersage der KPIs.

Im zweiten Subschritt wurden die *Konkreten Szenarien* in 22 Cluster eingeordnet. Durch die Auswahl der Anzahl der Cluster wurden weitere 8.689 *Konkrete Szenarien* „reduziert“. Die Gesamtliste an *Konkreten Szenarien* wurde abgearbeitet. Es sind keine Szenarien mit dem Ausführungszustand „ausstehend“ mehr vorhanden. Die Anzahl der Cluster (22 Stück) wurde mit Hilfe der Elbow Methode ermittelt.

Für die Anwendung des Methodenbausteins ergeben sich folgende Vor- und Nachteile:

Vorteile:

- Im Ergebnis des Methodenbausteins sind keine „ausstehenden“ *Konkreten Szenarien* mehr vorhanden (vgl. Anforderung 2).
- Der Methodenbaustein basiert auf der Berechnung von KPIs und ist - sofern eine Berechnung der KPIs an der Testplattform möglich ist - unabhängig von der Testplattform (vgl. Anforderung 6).
- Ergebnisse von „ausgeführten“ Szenarien verbessern die Vorhersage von Metriken. Die Ergebnisse werden für die nächsten Iterationen direkt wiederverwendet. Die Vorhersage der KPI erlaubt eine Möglichkeit des Vergleichs von „ausstehenden“ *Konkreten Szenarien* (vgl. Anforderung 8).
- Ein Subsampling lässt sich optional ergänzen (vgl. Subschritt $\alpha.e$), um Stellen mit einer hohen Unsicherheit im Vorhersagemodell zu schließen (vgl. Anforderung 8).

- Eine Entscheidung über das Unsicherheitsmaß und die Kategorisierung wird nachvollziehbar dokumentiert und steht für spätere Analysen zur Verfügung (vgl. Anforderung 13 und Anforderung 14).

Nachteile:

- Eine Plausibilisierung der Auswahl ist nur mit Ausführungsaufwand möglich (Widerspruch zu Anforderung 9).
- Wenn die Unsicherheit der KPIs in der unsicherheitsbasierten Auswahl auch nach mehreren Ausführungszyklen nicht sinkt, sind irgendwann alle „ausstehenden“ *Konkreten Szenarien* ausgeführt. Es wird nichts „reduziert“ (Widerspruch Anforderung 9).
- Es werden bereits „ausgeführte“ Szenarien benötigt. Das Reduktionspotential ist abhängig von den bereits „ausgeführten“ *Konkreten Szenarien* und lässt sich vorab nicht bestimmen (Widerspruch Anforderung 12).
- Die Auswahl der KPIs ist für diesen Methodenbaustein relevant. Die Vorhersagemodelle setzen deterministisches Verhalten der KPIs voraus (Widerspruch Anforderung 11).

7.3 Zusammenfassung und Diskussion der Evaluation des beispielhaften Szenarios

7.3.1 Beitrag zur Absicherung und zur Reduktion des Testaufwands

Aus Sicht der Testplanung lässt sich nach Durchführung des Selektionsprozesses der zeitliche Vorteil gegenüber einer Vollaussführung aller betrachteten Kombinationen ermitteln. Der Testaufwand sinkt im exemplarisch ausgewählten *Logischen Szenario* im Vergleich zur Vollaussführung mit 157.950 *Konkreten Szenarien* (7.897,5 Stunden - bei einer Testinstanz) auf 4.692 *Konkrete Szenarien* (234 Stunden - bei einer Testinstanz) nach der Ausführung des Selektionsprozesses mit den ausgewählten fünf Methodenbausteinen (vgl. Abbildung 7.20). Dies stellt eine Reduktion auf ca. drei Prozent dar und spart somit nicht nur Ressourcen an der entsprechenden Testinstanz, sondern erlaubt gleichzeitig auch eine argumentative Selektion. Die $Restunsicherheit_{Gesamt}$ ermittelt sich aus der Summe der RPZ der einzelnen Methodenbausteine zu 975,3 (vgl. Formel 5.23).

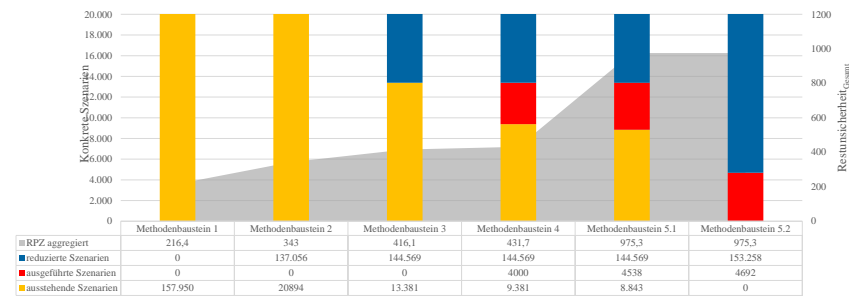


Abbildung 7.20: Verlauf der Restunsicherheit und der *Konkreten Szenarien* über die gewählten Methodenbausteine für das exemplarische *Logische Szenario*

Unter Berücksichtigung der Annahmen und der $Restunsicherheit_{Gesamt}$ von 975,3 erhält nun die verantwortliche Freigabeinstanz eine nachvollziehbare, argumentativ untermauerte Aussage über das *Logische Szenario*. Es wurden 4.692

Konkrete Szenarien für den dargestellten Use-Case gefunden, die repräsentativ für das *Logische Szenario* stehen (vgl. Abbildung 7.20).



Abbildung 7.21: Einordnung der Gesamtestunsicherheit des Selektionsprozesses

7.3.2 Vergleich der Methodenbausteine mit den Anforderungen

In Kapitel 4 wurden Anforderungen an den Selektionsprozess und die entsprechenden Methodenbausteine gestellt. Jeder Methodenbaustein bringt Vor- und Nachteile mit sich (vgl. Kapitel 7.2). Zur Bewertung der Methodenbausteine gegen die Anforderungen wurde ein Bewertungsschema von positiver Auswirkung auf die Anforderung (+ +) bis hin zu negativer Auswirkung auf die Anforderung (- -) in Form einer Matrix auf die einzelnen betrachteten Methodenbausteine angewendet (vgl. Abbildung 7.22).

Die Begründung für die Bewertung der Methodenbausteine beruht dabei auf den genannten Vor- und Nachteile (vgl. Kapitel 5). Die letztendliche Auswahl der Methodenbausteine muss für jedes *Logische Szenario* angepasst werden. Die im Rahmen dieser Promotion vorgestellten Methodenbausteine bieten eine Grundlage für alle *Logische Szenarien*.

Der Selektionsprozess wurde im Rahmen dieser Promotion exemplarisch an einem *Logischen Szenario* angewendet. Zur Absicherung der kompletten ODD (vgl. Kapitel 2.4.3) mit weiteren *Logischen Szenarien* muss der Selektionsprozess auf die anderen *Logischen Szenarien* ausgeführt werden. Dabei müssen die jeweiligen Methodenbausteine an den Kontext der anderen *Logischen Szenarien* angepasst werden. Konkret müssen Restriktionen, Diskretisierungsabstände, die Reihenfolge der Methodenbausteine und der KPIs an das betrachtete *Logische Szenario*

adaptiert werden. Das Reduktionspotential hängt damit stark von dem *Logischen Szenario* und dem betrachteten SuT ab. Expertenwissen der Entwicklungs- und Testabteilungen helfen die richtigen Methodenbausteine auszuwählen. Die Anwendung des erforderlichen Expertenwissens erfordert neben personellen Kapazitäten auch einen zeitlichen Aufwand, insbesondere bei *Logischen Szenarien* mit mehreren variablen Parametern. Vor diesem Hintergrund ist es notwendig, ein angemessenes Verhältnis zwischen dem Aufwand für die Durchführung des Selektionsprozesses und dem zu erzielenden Reduktionspotential herzustellen. Ein unverhältnismäßig aufwändiger Selektionsprozess kann zu einem ineffizienten Ressourceneinsatz führen, während eine unzureichende Selektion dazu führt, dass vorhandene Reduktionspotentiale ungenutzt bleiben. Im Rahmen der exemplarischen Anwendung auf ein *Logisches Szenario* (vgl. Kapitel 7.1) wurde deutlich, dass die Entscheidung über die Anwendung einzelner Methodenbausteine maßgeblich vom Erfahrungswissen der Fachabteilungen geprägt ist. Um eine übergreifende Anwendung auf andere Szenarien sicherzustellen, bedarf es strukturierter Guidelines, die künftig in Form standardisierter Regelwerke oder Templates operationalisiert werden können.

Die gezeigte beispielhafte Reduktion auf vier Prozent ist daher nur exemplarisch. Durch die Ergänzung von weiteren Methodenbausteinen ist eine größere Reduktion möglich, welche wiederum zu einem Anstieg der RPZ führt. Der beispielhafte Aufbau ist dabei ein gutes Grundgerüst für alle weiteren *Logische Szenarien*. Auf der anderen Seite ist ebenso nicht garantiert, dass sich bei jedem *Logischen Szenario* beispielsweise geometrische Reduktionen oder Zusammenfassungen anwenden lassen. Sollten die variablen Parameter des *Logischen Szenarios* keine Reduktion oder Zusammenfassung auf Basis der vorgestellten Kriterien zulassen, entfällt die Anwendung einzelner Methodenbausteine. Das Gesamtkonzept des modularen Selektionsprozess bleibt jedoch erhalten.

Alles in allem bietet der Selektionsprozess eine strukturierte Option, den Zielkonflikt zwischen Testausführungsaufwand und Unsicherheit bei der Auswahl von repräsentativen *Konkreten Szenarien* für ein *Logisches Szenario* zu lösen.

7.3.3 Praktikabilität und Rolle des Expertenwissens im Selektionsprozess

Die Anwendung des vorgestellten modularen Selektionsprozesses setzt in mehreren Phasen domänenspezifisches Expertenwissen voraus. Insbesondere bei der Formulierung von Restriktionen, der Parameterauswahl sowie der Festlegung von Bewertungskriterien (KPIs) ist die Einbindung von Fachexperten erforderlich. Diese domänenspezifische Wissensbasis ist notwendig, um:

- die Relevanz eines *Logischen Szenarios* zu beurteilen,
- geeignete Parameterbereiche und Diskretisierungsabstände zu wählen,
- methodenspezifische Schwellenwerte (z.B. Ähnlichkeitsmetriken) zu definieren.

Der Selektionsprozess ist aktuell teilautomatisiert: Die Anwendung einzelner Methodenbausteine erfolgt mithilfe von Skripten (z.B. Python), jedoch bedarf es für die Parametrierung und Validierung menschlicher Entscheidungen. Eine vollständige Automatisierung ist derzeit nicht realisierbar, da insbesondere semantische Entscheidungen (z.B. ob eine Reduktion zulässig ist) nicht vollständig algorithmisch getroffen werden können.

Um die Wiederholbarkeit und Nachvollziehbarkeit der getroffenen Entscheidungen zu erhöhen, werden alle Schritte im Selektionsprozess dokumentiert. Dies umfasst:

- Die verwendeten Parameterbereiche und Restriktionen
- Die Auswahl- und Gewichtung der KPIs
- Die eingesetzten Algorithmen und Metriken
- Die jeweilige Version des Systems-under-Test (SuT)

Zudem ermöglicht die modulare Struktur des Prozesses eine kontrollierte Wiederverwendung einzelner Schritte bei leicht veränderten Szenarien. Eine vollständige formale Standardisierung ist perspektivisch denkbar, jedoch mit weiterem Forschungsbedarf verbunden.

8 Zusammenfassung und Ausblick

8.1 Zusammenfassende Ergebnisse und Eigenleistungen

Die Promotion behandelt die Entwicklung einer Vorgehensweise zur szenarienbasierten Absicherung (vgl. Kapitel 5), die die Anforderungen der funktionalen Sicherheit und der Dokumentation erfüllt. Der Szenarienselektionsprozess entstand im Kontext des Forschungsprojekts VV-Methoden und wurde zur Unterstützung der Absicherung eines Bus-Plattons im Forschungsprojekt MINGA verwendet. Die grundlegenden Ideen fließen in die Zusammenarbeit mit Mercedes-Benz bei der Analyse von Auffälligkeiten im Hardware-in-the-loop Prüfstandskontext ein und unterstützen hier bei der systematischen Findung von Fehlern und Auffälligkeiten.

Im Fokus steht die Adaption bestehender Methoden und die Anwendung als Methodenbaustein im Selektionsprozess, mit dem Ziel, den Prozess zur Bestimmung repräsentativer *Konkreter Szenarien* zu standardisieren und zu formalisieren. Der entwickelte Selektionsprozess überwindet bisherige Limitationen bei der effizienten Szenarienselektion, indem er erstmals verschiedene Methodenbausteine kombiniert und so eine subtraktive nachvollziehbare Argumentation über den Parameterraum zulässt und gleichzeitig den Ausführungsaufwand reduziert.

Der Beitrag dieser Promotion umfasst unter anderem die Konzeption und Implementierung von Schnittstellen für eine Integration ins V-Modell (vgl. Kapitel

4.1.1), sowie Bewertungen zu Vor- und Nachteilen im Einsatz. Weitere Eigenleistungen liegen in der methodischen Erarbeitung der verschiedenen Methodenbausteine zur Ähnlichkeitsanalyse, statistischen Verteilungen und Unfallschwereinstufung. Ein besonderes Augenmerk liegt auf der Erstellung einer Restriktionsliste und der Entwicklung von Prädiktionsmodellen zur Vorhersage von KPIs zur Ermittlung von ähnlichen Szenarien, die repräsentativ für ein *Logisches Szenario* sind und dabei den aufgestellten Anforderungen und Rahmenbedingungen genügen. Durch die Anwendung des Selektionsprozesses wurde gezeigt, dass eine Reduktion der Testszenarien auf drei Prozent bei gleichzeitiger Beibehaltung der Testtiefe möglich ist (vgl. Kapitel 7). Dennoch wäre durch die Anwendung weiterer Methodenbausteine noch eine weitere Reduktion möglich gewesen, bspw. durch eine Unfallschwere- und Auftretenshäufigkeitsbetrachtung (vgl. Kapitel 5.4.2). Die Anwendung von weiteren Methodenbausteinen führt zu einer Erhöhung der Gesamtrisikoprioritätszahl, reduziert aber gleichzeitig den Testaufwand. Das beispielhafte Szenario dient im Rahmen dieser Promotion vor allem zum Verständnis des Konzepts des Selektionsprozesses.

Die Ähnlichkeitsanalyse konnte eine Genauigkeit von 95 Prozent bei der Identifikation repräsentativer Szenarien gewährleisten. Die Anwendung der entwickelten Werkzeuge bei Mercedes-Benz und im Forschungsprojekt VV-Methoden führte zu einer Reduktion des Testaufwands, wodurch Prüfstandszeit eingespart wurde. Die enge Zusammenarbeit mit Partnern, wie Mercedes-Benz und Ford garantierte eine enge Einbindung in die Entwicklungsmethodik und Vorgehensweise in der Automobilindustrie. Die Einordnung des Selektionsprozesses und die Gegenüberstellung mit den Anforderungen machen die Limitationen und Rahmenbedingungen klar, in welchem sich der Selektionsprozess befindet.

Die wissenschaftliche Ausarbeitung leistet somit einen entscheidenden Beitrag zur Standardisierung und Optimierung des Prozesses zur Selektion *Konkreter Szenarien* und trägt zur Qualitätssicherung sowie Effizienzsteigerung in der sicherheitsorientierten Entwicklung von Fahrzeugfunktionen bei.

8.2 Ausblick auf weitere Forschungsfelder und Schwerpunkte

Parallelisierbarkeit der Ausführung des Selektionsprozesses

Die vorgestellten Methodenbausteine im Selektionsprozess wurden bisher sequentiell nacheinander abgearbeitet. Diese sequentielle Abarbeitung und Analyse führten dazu, dass für das in Kapitel 8.1 vorgestellte *Logische Szenario* eine Gesamtbearbeitungszeit von 234 Stunden benötigt wurde, um eine Entscheidung über alle *Konkreten Szenarien* zu treffen. Eine parallelisierte Abarbeitung der Methodenbausteine sowie der einzelnen Subschritte innerhalb dieser Bausteine steigert die Effizienz erheblich. Moderne Parallelisierungsansätze, wie verteiltes Rechnen auf Cluster-Architekturen oder GPU-basierte Berechnungen, erlaubt eine Beschleunigung der Entscheidungsprozesse und eine Verbesserung der Skalierbarkeit, insbesondere bei einer zunehmenden Anzahl von Szenarien.

Erstellung des Gesamtszenarienkatalogs zur subtraktiven ODD Analyse für eine Gesamtabdeckungsaussage

Die Kombination des Selektionsprozesses mit der wissens- oder datenbasierten Generierung von *Logischen Szenarien* stellt eine zentrale Erweiterung dar, um den Gesamtaufwand einer Absicherung fundiert abzuschätzen. Hierbei müssen Mechanismen entwickelt werden, die es erlaubt, die Anzahl relevanter *Logischer Szenarien* für eine vollständige Abdeckung der ODD systematisch zu reduzieren, ohne dabei die Aussagekraft der Analyse zu beeinträchtigen. Ein besonderer Fokus sollte auf die Ableitung von *Logischen Szenarien* aus bestehenden Wissens- und Datenbasen gelegt werden. Die Herausforderung hierbei liegt nicht nur in der logischen Ableitung von Szenarien, sondern auch in der intelligenten Strukturierung der zugrundeliegenden Wissensbasis, um Redundanzen zu vermeiden und den Aufwand zu minimieren.

Erweiterung um weitere Methodenbausteine

Die bisher entwickelten Methodenbausteine stellen lediglich eine erste Sammlung an Bausteinen dar. In der aktuellen Forschung gibt es weitere vielversprechende Ansätze, die sich in den bestehenden Methodenbaukasten integrieren lassen. Ein bedeutender Aspekt ist die Anwendung von KI-gestützten Verfahren. Diese erweiterten Methodenbausteine könnten dazu beitragen, eine verbesserte Überprüfung und Validierung, der im Selektionsprozess angewandten Kriterien, zu ermöglichen.

Abbruchkriterium zur weiteren Effizienzsteigerung bei der entwicklungsbegleitenden Anwendung

Ein weiterer wesentlicher Punkt ist die Definition von Abbruchbedingungen. Im derzeitigen Konzept wird der Selektionsprozess sequenziell durchlaufen, ohne dass explizite Unterbrechungskriterien für schwerwiegende Fehlerzustände implementiert sind. Sollte während des Selektionsprozesses ein gravierender Fehler in dem zu testenden System identifiziert werden, muss der Selektionsprozess unterbrochen werden, bis dieser Fehler behoben ist. Hierfür müssen geeignete Mechanismen entwickelt werden, die eine frühzeitige Fehlererkennung erlauben und adaptive Teststrategien vorschlagen, um Ressourcen effizienter zu nutzen.

Validität der Testinstanz

Ein entscheidender Aspekt für die Aussagekraft des Selektionsprozesses ist die Validität der Testinstanz. Die entwickelte Methodik muss sicherstellen, dass die Testumgebung sowie die durchgeführten Analysen tatsächlich die Realweltbedingungen adäquat widerspiegeln. Hierbei spielen Faktoren wie die Qualität der Simulationsmodelle, die Korrektheit der Annahmen sowie die Reproduzierbarkeit der Testergebnisse eine wesentliche Rolle. Die Integration von Methoden zur Validitätsprüfung in den Selektionsprozess stellt ein wichtiges Forschungsfeld dar, um Verzerrungen oder ungewollte Einschränkungen zu vermeiden.

Übertragbarkeit auf andere Domänen

Die entwickelten Methoden des Selektionsprozesses könnten nicht nur im Bereich der hochautomatisierten Fahrzeuge Anwendung finden, sondern auch in anderen sicherheitskritischen Domänen wie der Luftfahrt oder der Medizintechnik. In der Luftfahrt könnten beispielsweise Sicherheitsanalysen für autonom agierende Drohnensysteme oder Assistenzsysteme für Piloten nach einem ähnlichen Prinzip erfolgen. In der Medizintechnik könnten Selektionsprozesse zur Bewertung der Sicherheit von KI-gestützten Diagnoseverfahren oder roboterassistierten chirurgischen Eingriffen genutzt werden. Eine weiterführende Forschung sollte untersuchen, welche Anpassungen erforderlich sind, um die entwickelten Methoden auf verschiedene Anwendungsfälle zu übertragen.

Flexible Skalierung der Methodenbausteine mit Hilfe von maschinellem Lernen

Ein weiterer vielversprechender Ansatz ist die Nutzung von maschinellem Lernen zur flexiblen Skalierung der Methodenbausteine. Abhängig von den vorliegenden Daten und der Komplexität der betrachteten Szenarien könnten adaptive Algorithmen eingesetzt werden, die die Auswahl und Kombination der Methodenbausteine optimieren. Hierzu könnten Reinforcement-Learning-Ansätze oder evolutionäre Algorithmen genutzt werden, um den Selektionsprozess kontinuierlich zu verbessern und an neue Bedingungen anzupassen.

Anwendung des Konzepts des Selektionsprozesses auf die Additive Vollständigkeitsanalyse

Das Konzept des Selektionsprozesses lässt sich auch auf die additive Vollständigkeitsargumentation anwenden, um eine systematische und reproduzierbare Überprüfung der Vollständigkeit von Testfällen sicherzustellen. Dies könnte beispielsweise durch eine iterative Erweiterung des Szenarienraums erfolgen, wobei gezielt Lücken identifiziert und geschlossen werden. Die Nutzung heuristischer Methoden oder KI-gestützter Optimierungsverfahren könnte hier dazu beitragen, den Selektionsprozess effizient zu gestalten und eine umfassende Testabdeckung zu gewährleisten. Ein weiterer Aspekt ist die Integration von Unsicherheitsmodellen,

um die Auswirkungen unbekannter oder schwer modellierbarer Einflussfaktoren zu berücksichtigen.

Skalierung des Testaufwands auf den Gesamtabsicherungsprozess für eine hochautomatisierte Fahrfunktion

Alles in allem ist nach dem aktuellen Stand der Technik unbekannt, wieviele übergeordnete *Logische Szenarien* zur Absicherung einer hochautomatisierten Fahrfunktion nötig sind. Daher lässt sich abschließend nicht sagen, welche Reduktion benötigt wird, um den Gesamttestausführungsaufwand ausführbar zu halten.

Dennoch konnte nachgewiesen werden, dass der Selektionsprozess auf ein *Logisches Szenario* angewendet werden konnte. Neben der Vorstellung verschiedener Methodenbausteine wurde gezeigt, dass eine deutliche Reduktion gegenüber der Vollauführung möglich ist. Die jeweils benötigten Restriktionen und Einschränkungen sind dabei in Abhängigkeit des betrachteten *Logischen Szenarios* anzupassen und zu adaptieren.

Abbildungsverzeichnis

1.1	Herausforderungen in der Entwicklung und Freigabe einer hochautomatisierten Fahrfunktion nach [85]	5
1.2	Zunahme der <i>Konkreten Szenarien</i> in Abhängigkeit der Anzahl der Parameter	7
1.3	Beispielhafte qualitative Darstellung des Zielkonflikts zwischen Testaufwand und verbleibender Restunsicherheit	8
1.4	Aufbau	10
2.1	Begriffliche Einordnung - Fahrzeugdomäne	12
2.2	Automatisierungsstufen zum automatisierten Fahren [97]	14
2.3	Begriffliche Einordnung - Entwicklungsprozess	15
2.4	V-Modell in Anlehnung an [56]	17
2.5	Beispielhafte Taktung von Hardware, Software und Mechanik nach Reifegrad (entnommen aus [119])	20
2.6	Zusammenhang der Begrifflichkeiten im Testumfeld	23
2.7	Einteilung der Testplattformen nach Grad der Integration nach [109]. Dabei variiert die genaue Ausprägung der einzelnen Kategorien einer jeweiligen Testinstanz in Abhängigkeit des Testziels.	25
2.8	Begriffliche Einordnung - Szenario	28
2.9	Der Testfall im Kontext des Szenarios nach [120]	30
2.10	Aufteilung der Szenarien in die verschiedenen Abstraktionsstufen	33
2.11	Pegasus 6+1-Layer-Modell [61] nach [102]	34
2.12	Zuordnung von Parametern zu den Szenarienabstraktionsklassen nach [69]	37
2.13	Anwendungen des Szenarienkatalogs nach [91]	39
2.14	Abdeckung der ODD mit Szenarien nach [120]	42

3.1	Beispielentscheidungsbaum für die Szenarienanalyse nach [19]	52
3.2	Underfitting und Overfitting bei der Klassifizierung nach [74]	55
3.3	Unterschied zwischen überwachtem (links) und unüberwachtem (rechts) Lernen	56
3.4	Verschiedene Arten von Szenarien, je nach Quelle und enthaltenen Informationen. Aufgezeichnete Szenarien bestehen typischerweise aus aufgezeichneten Zeitreihen (z. B. Trajektorien), ausführbare Szenarien enthalten Anweisungen für die Verkehrsteilnehmer (z. B. die auszuführenden Manöver), und naturalistische Fahrdaten enthalten Rohdaten Messungen des Verkehrs nach [24]	59
3.5	Unsicherheitsprädiktion beim Gaussian Process Regressor [64]	61
3.6	Hauptkomponente der PCA Analyse nach [78]	64
4.1	Einordnung der Methodik in den Gesamtkontext der Erstellung eines Szenarienkatalogs basierend auf Abbildung 2.13	67
4.2	Auswahl von relevanten KPIs im Kontext des SuT und der Testinstanz für ein <i>Logisches Szenario</i>	74
5.1	Argumentative Vorgehensweise bei der Auswahl von repräsentativen Szenarien	77
5.2	Verwendung des Selektionsprozesses bei der Entwicklung entlang des V-Modells	79
5.3	Schritte des Methodenbaukastens als Ablaufdiagramm inklusive beispielhafte Ergebnisdarstellung im Parameterraum	79
5.4	Schematischer Aufbau eines Methodenbausteins mit Schnittstellen und beispielhafte Entscheidung über Ausführungszustand im Fall mit zwei variablen Parametern	83
5.5	Entitätsdiagramm zur Szenarienselektion im Kontext der Methodenbausteine und des <i>Logischen Szenarios</i> mit dem in rot umrandeten Konzept des Selektionsprozesses zur Strukturierung. Der Selektionsprozess greift das in Abbildung 5.3 dargestellte Vorgehensmodell auf.	85
5.6	Ermittlung der Riskoprioritätszahl für jeden Methodenbaustein	87
5.7	Schritt I des Selektionsprozesses nach Abbildung 5.3	93

5.8	Schritt II des Selektionsprozesses nach Abbildung 5.3 mit beispielhaften Reduktionen	101
5.9	Anwendung der Liste von Restriktionen auf die Szenariensliste	103
5.10	Prinzipskizze zur Zusammenfassung von mehreren Parameter zu einem abgeleiteten Parameter	104
5.11	Schritt III des Selektionsprozesses nach Abbildung 5.3 mit beispielhaften Reduktionen	110
5.12	Zusammenfassen von Unfallschwere und Auftrittswahrscheinlichkeit zu einer Gesamtpriorisierung	114
5.13	Grafische Darstellung des Konzepts zur Ermittlung der Auftretenshäufigkeit - analog zu Abbildung 5.14	116
5.14	Aktivitätsdiagramm zur Ermittlung der Häufigkeit eines Konkreten Szenario in einem logischen Szenario	117
5.15	Datenstruktur zur Zerlegung der Messreihen in eine manöverbasierte Beschreibungsform mit Parameternwerten	118
5.16	Ableitung einer Maßnahme aus Auftrittswahrscheinlichkeit und Unfallschwere in Analogie zur ASIL-Klassifikation	124
5.17	Datenflussdiagramm zur Prädiktion der Metriken auf Basis der Parameter	126
5.18	Grafische Darstellung der Vorgehensweise zur repräsentativen Prädiktion von Metriken in Anlehnung an Abbildung 5.19	127
5.19	Ablaufdiagramm für die Vorhersage und Clustering von Metriken für die Szenarienanalyse	130
5.20	Beispielhafter Verlauf der Restunsicherheit _{Gesamt} und der Anzahl der „ausstehenden“ <i>Konkreten Szenarien</i> über die verschiedenen Methodenbausteine und Annahmen im Laufe des Selektionsprozesses	132
6.1	Einordnung in den Gesamtkontext der Szenarienausführung (angelehnt an Abbildung 2.13) inklusive Zuordnung zur Datenbankstruktur über Farbsemantik (vgl. Abbildung 5.5)	138
6.2	Klassendiagramm der Kernfunktionalität eines Methodenbausteins mit seinen Schnittstellen zur Datenbank und zum Selektionsprozess	141
6.3	Verwendete Bibliotheken für die Implementierung des jeweiligen Methodenbausteins als Matrix Darstellung	142

7.1	Aufbau <i>Logisches Szenario</i> - Beispiel	146
7.2	Ausgewählte Methodenbausteine für das beispielhafte <i>Logische Szenario</i> nach Abbildung 5.3	148
7.3	3D Plot der <i>Konkreten Szenarien</i> im Parameterraum als Ergebnis des ersten Methodenbausteins - 157.950 „ausstehende“ <i>Konkrete Szenarien</i>	151
7.4	Relevanter Abstand $d_{relevant}[m]$ ab dem das Ego den VRU wahrnimmt. VRU betritt den Zebrastreifen und kommt aus der Verdeckung hervor	154
7.5	3D Plot des Szenarienraums über die Parameter mit den 137.056 „reduzierten“ <i>Konkreten Szenarien</i> , welche auf Basis der beiden Restriktionen ausgeschlossen wurden	155
7.6	Evaluation des Erfüllungsgrades der beiden Restriktion und Vergleich zwischen Messung und Berechnung bei einer Stichprobe von 2000 zufälligen <i>Konkreten Szenarien</i> . Restriktion 1 (links - 46 ungültige Annahmen) und Restriktion 2 (rechts - 2 ungültige Annahmen)	158
7.7	Absolute Abweichung zwischen wahrer Wert aus Ausführung und durch Restriktion angenommenen Wert im Parameterraum (links) und als Häufigkeitsverteilung (rechts). Durchschnittliche Abweichung: -0.1569m Standardabweichung: 0.1806m	159
7.8	Cluster zu Ordnung über den Raum der betrachteten „ausstehenden“ Szenarien (links) und Ergebnis 3D Plot der <i>Konkreten Szenarien</i> im Parameterraum nach der Ausführung des dritten Methodenbausteins (rechts)	163
7.9	Evaluation der absoluten Abweichung der relevanten Zeit mit Hilfe von ausgeführten <i>Konkreten Szenarien</i>	165
7.10	Ergebnis 3D Plot der <i>Konkreten Szenarien</i> im Parameterraum nach der Ausführung des vierten Methodenbausteins	167
7.11	Unsicherheitsvorhersage für KPI $d_{relevant}$ im Parameterraum (rechts) und als Häufigkeitsdiagramm (links) (Zusätzliche Ausführung von 133 <i>Konkreten Szenarien</i>)	170
7.12	Unsicherheitsvorhersage für KPI $d_{ego->VRU}$ im Parameterraum (rechts) und als Häufigkeitsdiagramm (links) (Zusätzliche Ausführung von 538 <i>Konkreten Szenarien</i>)	171

7.13	Unsicherheitsvorhersage für KPI t_{brems} im Parameterraum (rechts) und als Häufigkeitsdiagramm (links) (Zusätzliche Ausführung von 35 Konkreten Szenarien)	171
7.14	Ergebnis 3D Plot der <i>Konkreten Szenarien</i> im Parameterraum nach der Ausführung des ersten Subschritts des fünften Methodenbausteins	173
7.15	Cluster gebildet mit k-Means im Parameterraum (rechts) und im KPI-Raum (links)	174
7.16	Gesamtergebnis 3D Plot der <i>Konkreten Szenarien</i> im Parameterraum nach der Ausführung des fünften Methodenbausteins	174
7.17	Gegenüberstellung wahrer KPI Wert mit der Vorhersage auf Basis der Gauß-Prozessregression mit Unsicherheit (links) und der Verteilung des absoluten Fehlers im Parameterraum (rechts) für $d_{relevant}$	176
7.18	Gegenüberstellung wahrer KPI Wert mit der Vorhersage auf Basis der Gauß-Prozessregression mit Unsicherheit (links) und der Verteilung des absoluten Fehlers im Parameterraum (rechts) für $d_{ego \rightarrow VRU}$	177
7.19	Gegenüberstellung wahrer KPI Wert mit der Vorhersage auf Basis der Gauß-Prozessregression mit Unsicherheit (links) und der Verteilung des absoluten Fehlers im Parameterraum (rechts) für t_{brems}	177
7.20	Verlauf der Restunsicherheit und der <i>Konkreten Szenarien</i> über die gewählten Methodenbausteine für das exemplarische <i>Logische Szenario</i>	180
7.21	Einordnung der Gesamtrestunsicherheit des Selektionsprozesses . . .	181
7.22	Bewertung der Methodenbausteine gegen die aufgestellten Anforderungen	185
A.1	Einstufung und Begründung der RPZ für das ausgewählte <i>Logische Szenario</i>	223

Tabellenverzeichnis

7.1 Fehler bei der Vorhersage der KPIs.	178
---	-----

Eigene Veröffentlichungen

- [1] Birkenmeyer, L. ; Fuchs, J. ; Gambi, A. ; Schaefer, I. : SOTIF-Compliant Scenario Generation Using Semi-Concrete Scenarios and Parameter Sampling. In: *IEEE ITSC 2023*, 2023
- [2] Braun, T. ; Fuchs, J. ; Resigys, F. ; Ries, L. ; Plaum, J. ; Schütt, B. : A Review of Scenario Similarity Measures for Validation of Highly Automated Driving. In: *IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)* (2024)
- [3] Fuchs, J. : Automated 3D Object Reference Generation for the Evaluation of Autonomous Vehicle Perception. In: *International Conference on System Reliability and Safety (ICSRS)* (2021)
- [4] Fuchs, J. : Nutzung der Ähnlichkeitsanalyse konkreter Szenarien für die Entwicklung von Testplänen im Rahmen der virtuellen szenariobasierten Validierung von Fahrzeugen. In: *VDI Tagung hochautomatisiertes Fahren*, 2024
- [5] Fuchs, J. ; Kober, C. : Testfall-Erkennung in Erprobungsfahrten. In: *VDI Tagung Autotest Stuttgart*, 2024
- [6] Fuchs, J. ; Luentzel, V. ; Kaag, K. ; Sax, E. : Simulation of realistic vehicle battery voltage profiles on hardware-in-the-loop test benches. In: *Shanghai-Stuttgart-Symposium Automotive and Powertrain Technology* (2022)
- [7] Fuchs, J. ; Ries, L. ; Sax, E. : Estimating Scenario Occurrence for Virtual Validation of Automated Vehicles. In: *Stuttgart International Symposium on Automotive and Engine Technology*, 2024

- [8] Fuchs, J. ; Ries, L. ; Sax, E. : Occurrence estimation for the Classification and Prioritization of Concrete Scenarios in the Context of Virtual Scenario-Based Validation of Vehicles. In: *Stuttgart International Symposium on Automotive and Engine Technology* (2024)
- [9] Fuchs, J. ; Steinhauser, C. ; Ries, L. ; Sax, E. : Usage of stimulation-independent evaluation in the context of vehicle body domain. In: *Journal of Tongji University* (2023)

Abschlussarbeiten

- [1] Alhayani, H. : *Modellbasierte Testfallerstellung zur automatisierten HiL-Absicherung von Sitzmodulen.* 2022
- [2] Dürr, J. : *Bestimmung von Häufigkeitsverteilungen konkreter Szenarien auf Grundlage von Manövern am Beispiel von Top-View-Kameraaufnahmen.* 2024
- [3] Fabos, V. : *Konzeption und Evaluation von Technologien und Methoden zur Optimierung einer automatischen Lüftersteuerung für Dunstabzüge.* 2022
- [4] Hahn, C. : *Entwicklung einer Methode zur systematischen, presimulativen Auswahl konkreter Szenarien auf Basis physikalischer und logischer Einflussfaktoren.* 2023
- [5] Heisterklaus, T. : *Automated Driving Analytics – KI-basierte Auswertung von Fehlerfällen in automatisierten Fahrsystemen.* 2022
- [6] Hörter, M. : *Präsimulative Ähnlichkeitsanalyse auf Basis von Zeitreihen zur Priorisierung von Szenarien im Kontext der virtuellen, szenariobasierten Absicherung von Fahrzeugen.* 2024
- [7] Lüntzel, V. : *Entwicklung eines Prozesses zur Ableitung von Bewertungskriterien auf Grundlage der Zerlegung von Anforderungen für das automatisierte Fahren.* 2021
- [8] Pan, L. : *Ähnlichkeitsanalyse konkreter Szenarien im Kontext der virtuellen, szenariobasierten Validierung von Fahrzeugen.* 2022

- [9] Phielers, N. : *Entwicklung eines adaptiven GUI-Frameworks zur Stimulation von End-to-End-Wirkketten im Hardware-in-the-Loop-Umfeld.* 2024
- [10] Siknas, A. : *Selektion und Priorisierung von Top-View-Fahraufnahmen im szenariobasierten Testen hochautomatisierter Fahrfunktionen.* 2023
- [11] Steiner, H. : *Probabilistische Modellierung der statischen Umgebung auf der Grundlage von Radardaten in Nutzfahrzeugen.* 2025
- [12] Varon, L. M. L.: *Zukunftsfähigkeit von individuellen App Lösungen und Mobilitätsanwendungen.* 2022
- [13] Yujia, L. : *Entwicklung eines Prozesses zur Ableitung von Bewertungskriterien auf Grundlage der Zerlegung von Anforderungen für das automatisierte Fahren.* 2021
- [14] Zhu, Z. : *Automatisierte, objektbasierte Referenzgenerierung zur Bewertung der Umgebungswahrnehmung hochautomatisierter Fahrsysteme.* 2020

Literaturverzeichnis

- [1] *Fahrassistenzsysteme - Pflicht ab 2022 | TÜV NORD*. <https://www.tuev-nord.de/de/privatkunden/ratgeber-und-tipps/technik/fahrassistenzsysteme/>. Version: 07.10.2022
- [2] Dynamic Time Warping. Version: 2007. http://dx.doi.org/10.1007/978-3-540-74048-3_4. In: *Information Retrieval for Music and Motion*. Springer, Berlin, Heidelberg. – DOI 10.1007/978-3-540-74048-3_4, 69–84
- [3] Version: 2008. <http://dx.doi.org/10.1524/9783486710236.195>. In: Bacher, J. (Hrsg.): *Clusteranalyse*. München : Oldenbourg R, 2008. – DOI 10.1524/9783486710236.195. – ISBN 978-3-486-58457-8, S. 195–231
- [4] Abdessalem, R. B. ; Nejati, S. ; Briand, L. C. ; Stifter, T. : Testing vision-based control systems using learnable evolutionary algorithms. In: Crnkovic, I. (Hrsg.): *Proceedings of the 40th International Conference on Software Engineering*. New York, NY : ACM, 2018 (ACM Conferences). – ISBN 9781450356381, S. 1016–1026
- [5] Association for Standardization of Automation and Measuring Systems: *ASAM OpenDRIVE*. <https://www.asam.net/standards/detail/openscenario/>. Version: 2021
- [6] Association for Standardization of Automation and Measuring Systems: *ASAM OpenSCENARIO*. <https://www.asam.net/standards/detail/openscenario/>. Version: 2022
- [7] Atev, S. ; Miller, G. ; Papanikolopoulos, N. P.: Clustering of Vehicle Trajectories. In: *IEEE Transactions on Intelligent Transportation Systems* 11 (2010),

- Nr. 3, S. 647–657. <http://dx.doi.org/10.1109/TITS.2010.2048101>.
– DOI 10.1109/TITS.2010.2048101. – ISSN 1524–9050
- [8] Bach, J. : *Methoden und Ansätze für die Entwicklung und den Test prädiktiver Fahrzeugregelungsfunktionen*. Karlsruhe. <http://dx.doi.org/10.5445/IR/1000085654>. <http://dx.doi.org/10.5445/IR/1000085654>
- [9] Bach, J. ; Holzäpfel, M. ; Otten, S. ; Sax, E. : Reactive-Replay Approach for Verification and Validation of Closed-Loop Control Systems in Early Development
- [10] Bacher, J. (Hrsg.): *Clusteranalyse: Anwendungsorientierte Einführung*. 3rd edition. München : Oldenbourg R, 2008. <http://dx.doi.org/10.1524/9783486710236>. <http://dx.doi.org/10.1524/9783486710236>. – ISBN 978–3–486–58457–8
- [11] Bagschik, G. ; Menzel, T. ; Andreas, R. ; Markus, M. : Szenarien für Entwicklung, Absicherung und Test von automatisierten Fahrzeugen. In: *Workshop Fahrerassistenzsysteme und automatisiertes Fahren (2017)*
- [12] Bamberg, G. ; Baur, F. ; Krapp, M. : *Statistik*. 17., überarb. Aufl. München : Oldenbourg, 2012 (Oldenbourgs Lehr- und Handbücher der Wirtschafts- und Sozialwissenschaften). – ISBN 978–3–486–71651–1
- [13] Batsch, F. ; Daneshkhah, A. ; Cheah, M. ; Kanarachos, S. ; Baxendale, A. : Performance Boundary Identification for the Evaluation of Automated Vehicles using Gaussian Process Classification. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, 102019. – ISBN 978–1–5386–7024–8, S. 419–424
- [14] Bayer, S. ; Wolf, M. ; Kreuzinger, T. ; Oka, D. K.: *Effektive Security-Tests am HiL-System* https://www.researchgate.net/publication/318210163_Effektive_Security-Tests_am_HiL-System
- [15] Beglerovic, H. ; Stolz, M. ; Horn, M. : Testing of autonomous vehicles using surrogate models and stochastic optimization. In: *2017 IEEE 20th*

-
- International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 102017. – ISBN 978–1–5386–1526–3, S. 1–6
- [16] Berger, V. W. ; Zhou, Y. : *Kolmogorov–Smirnov Test: Overview*. John Wiley & Sons, Ltd, 2014 (Wiley StatsRef: Statistics Reference Online). <http://dx.doi.org/10.1002/9781118445112.stat06558>. <http://dx.doi.org/10.1002/9781118445112.stat06558>
- [17] BFU ; Smartrider (Hrsg.): *5 Stufen der Automatisierung*. <https://smartrider.ch/de/aktuelles/5-stufen-der-automatisierung-1>. Version: 2019
- [18] Bian, J. ; Tian, D. ; Tang, Y. ; Tao, D. : *A survey on trajectory clustering analysis*. <https://arxiv.org/pdf/1802.06971>
- [19] Birkemeyer, L. ; Fuchs, J. ; Gambi, A. ; Schaefer, I. : *SOTIF-Compliant Scenario Generation Using Semi-Concrete Scenarios and Parameter Sampling*. Zenodo. <http://dx.doi.org/10.5281/ZENODO.7940901>. <http://dx.doi.org/10.5281/ZENODO.7940901>
- [20] Blissing, B. ; Bruzelius, F. ; Ölvander, J. : *Augmented and Mixed Reality as a tool for evaluation of Vehicle Active Safety Systems*
- [21] BMW Group: *BMW Group Safety Assessment Report: SAE Level 3 Automated Driving System*. (2020). <https://lindseyresearch.com/wp-content/uploads/2020/06/BMW.pdf>
- [22] Bock, J. ; Krajewski, R. ; Moers, T. ; Runde, S. ; Vater, L. ; Eckstein, L. : *The inD Dataset: A Drone Dataset of Naturalistic Road User Trajectories at German Intersections*. <http://arxiv.org/pdf/1911.07602>
- [23] Boehm, B. W.: *A spiral model of software development and enhancement*. In: *Computer* 21 (1988), Nr. 5, S. 61–72. <http://dx.doi.org/10.1109/2.59>. – DOI 10.1109/2.59. – ISSN 0018–9162

- [24] Braun, T. ; Fuchs, J. ; Reisgys, F. ; Ries, L. ; Plaum, J. ; Schütt, B. ; Sax, E. : Scenario Similarity Measures for Validation of Highly Automated Driving. (2023)
- [25] Bühler, O. ; Wegener, J. : Evolutionary functional testing. In: *Computers & Operations Research* 35 (2008), Nr. 10, S. 3144–3160. <http://dx.doi.org/10.1016/j.cor.2007.01.015>. – DOI 10.1016/j.cor.2007.01.015. – ISSN 03050548
- [26] Bussler, A. : Einsatz Evolutionärer Algorithmen zur Generierung relevanter Verkehrsszenarien zur Absicherung hochautomatisierter Fahrfunktionen. Version: 2018. <http://dx.doi.org/10.51202/9783181023358-71>. In: *Fahrerassistenzsysteme und automatisiertes Fahren 2018*. Düsseldorf : VDI Verlag, 2018 (VDI-Berichte). – DOI 10.51202/9783181023358-71. – ISBN 9783181023358, S. 71–78
- [27] C. Neurohr ; L. Westhofen ; T. Henning ; T. de Graaff ; E. Möhlmann ; E. Böde: Fundamental Considerations around Scenario-Based Testing for Automated Driving. In: *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020. – ISBN 2642–7214, S. 121–127
- [28] Cleff, T. : *Deskriptive Statistik und Explorative Datenanalyse: Eine computergestützte Einführung mit Excel, SPSS und STATA* <https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=3109170>. – ISBN 9783834947482
- [29] Daigl, M. ; Glunz, R. : *ISO 29119 - Die Softwaretest-Normen verstehen und anwenden*. Heidelberg : dpunkt.verlag, 2016. – ISBN 9783864917721
- [30] Dajsuren, Y. (Hrsg.) ; van Brand, M. den (Hrsg.): *Automotive systems and software engineering: State of the art and future trends*. Cham : Springer, 2019. – ISBN 978–3–030–12156–3
- [31] Ding, W. ; Xu, C. ; Arief, M. ; Lin, H. ; Li, B. ; Zhao, D. : *A Survey on Safety-Critical Driving Scenario Generation – A Methodological Perspective*. <https://arxiv.org/pdf/2202.02215>

- [32] Dröschel, W. (Hrsg.); Wiemers, M. (Hrsg.): *Das V-Modell 97: Der Standard für die Entwicklung von IT-Systemen mit Anleitung für den Praxiseinsatz*. De Gruyter Oldenbourg. <http://dx.doi.org/10.1515/9783486800265>. <http://dx.doi.org/10.1515/9783486800265>. – ISBN 3–486–25086–8
- [33] Eiter, T. ; Mannila, H. : *Computing-Discrete-Frechet-Distance*. https://www.researchgate.net/profile/Thomas-Eiter-2/publication/228723178_Computing_Discrete_Frechet_Distance/links/5714d93908aebda86c0d1a7b/Computing-Discrete-Frechet-Distance.pdf
- [34] Faber, F. A. ; Hutchison, L. ; Huang, B. ; Gilmer, J. ; Schoenholz, S. S. ; Dahl, G. E. ; Vinyals, O. ; Kearnes, S. ; Riley, P. F. ; Lilienfeld, O. A.: *Prediction Errors of Molecular Machine Learning Models Lower than Hybrid DFT Error*. Bd. 13. 2017. <http://dx.doi.org/10.1021/acs.jctc.7b00577>. <http://dx.doi.org/10.1021/acs.jctc.7b00577>
- [35] Fahrmeir, L. (Hrsg.) ; Kneib, T. (Hrsg.) ; Lang, S. (Hrsg.): *Regression: Modelle, Methoden und Anwendungen*. Berlin and Heidelberg : Springer, 2007 (Statistik und ihre Anwendungen). <http://dx.doi.org/10.1007/978-3-540-33933-5>. <http://dx.doi.org/10.1007/978-3-540-33933-5>. – ISBN 978–3–540–33932–8
- [36] Felix Reisgys ; Mohamed Elgharbawy ; Andreas Schwarzhaupt ; Eric Sax: Argumentation on ADAS Simulation Validity using Aleatory and Epistemic Uncertainty Estimation. In: *Proceedings of the Driving Simulation Conference 2021 Europe VR (2021)*, 25–32. <https://proceedings.driving-simulation.org/proceeding/dsc-2021/argumentation-on-ad-as-simulation-validity-using-aleatory-and-epistemic-uncertainty-estimation/>
- [37] Frigerio, A. ; Vermeulen, B. ; Goossens, K. : Component-Level ASIL Decomposition for Automotive Architectures. In: *DSN 2019 workshop volume*. Piscataway, NJ : IEEE, 2019. – ISBN 978–1–7281–3030–9, S. 62–69

- [38] Gräber, P.-W. : *Messfehler: Automatisierungstechnik in der Wasserwirtschaft*. Dresden, 2009
- [39] Grönlund, J. ; Sjödin, D. R. ; Frishammar, J. ; University of California (Hrsg.): *Open Innovation and the Stage-Gate Process: A Revised Model for New Product Development*. <https://journals.sagepub.com/doi/pdf/10.1525/cmr.2010.52.3.106>. Version: 2010
- [40] Han, H. ; Guo, X. ; Yu, H. : Variable selection using Mean Decrease Accuracy and Mean Decrease Gini based on Random Forest. In: Babu, M. S. P. (Hrsg.): *Proceedings of 2016 IEEE 7th International Conference on Software Engineering and Service Science*. Piscataway, NJ : IEEE, 2016. – ISBN 978–1–4673–9904–3, S. 219–224
- [41] Han, J. ; Kamber, M. ; Pei, J. : *Data mining: Concepts and techniques*. 3rd ed (Online-Ausg.). Elsevier professional <https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=729031>. – ISBN 978–0–12–381479–1
- [42] Hauer, F. ; Pretschner, A. ; Holzmüller, B. : Fitness Functions for Testing Automated and Autonomous Driving Systems. Version: 2019. http://dx.doi.org/10.1007/978-3-030-26601-1_{_}5. In: *Computer Safety, Reliability, and Security*. Springer, 2019. – DOI 10.1007/978–3–030–26601–1_5. – ISBN 978–3–030–26601–1
- [43] Hiroki Watanabe: *Methodik zur Determinierung repräsentativer und relevanter Testszenarien für prädiktive Sicherheitsfunktionen*. Dresden : TU Dresden, 2022
- [44] Hiroki Watanabe ; Tomáš Malý ; Johannes Wallner ; Tobias Dirndorfer ; Günther Prokop: *Methodology of Scenario Clustering for Predictive Safety Functions*
- [45] Huang, Z. ; Lam, H. ; Zhao, D. : Towards affordable on-track testing for autonomous vehicle — A Kriging-based statistical approach. In: *2017 IEEE*

-
- 20th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 102017. – ISBN 978–1–5386–1526–3, S. 1–6
- [46] IEEE: IEEE Standard Glossary of Software Engineering Terminology. In: *IEEE Std 610.12-1990* (1990). <http://dx.doi.org/10.1109/IEEESTD.1990.101064>. – DOI 10.1109/IEEESTD.1990.101064
- [47] International Organization for Standardization: *ISO 29119: Software and systems engineering - software testing*
- [48] International Organization for Standardization: *ISO 21448:2022*. <https://www.iso.org/standard/77490.html>. Version: 10.12.2024
- [49] International Organization for Standardization: *ISO 26262-6:2018: Road vehicles — Functional safety*. <https://www.iso.org/standard/68388.html>. Version: 2018
- [50] Junietz, P. ; Schneider, J. ; Winner, H. : Metrik zur Bewertung der Kritikalität von Verkehrssituationen und -szenarien. (2017)
- [51] Kerber, J. ; Wagner, S. ; Groh, K. ; Notz, D. ; Kuhbeck, T. ; Watzenig, D. ; Knoll, A. : Clustering of the Scenario Space for the Assessment of Automated Driving. In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. – ISBN 978–1–7281–6673–5, 578–583
- [52] King, C. : *Bewertung von Fahrerassistenzsystemen im Umfeld des szenario-basierten Testens*
- [53] King, C. ; Braun, T. ; Braess, C. ; Langner, J. ; Sax, E. : Capturing the Variety of Urban Logical Scenarios from Bird-view Trajectories. In: *Proceedings of the 7th International Conference on Vehicle Technology and Intelligent Transport Systems*, SCITEPRESS - Science and Technology Publications, 2021
- [54] King, C. ; Ries, L. ; Langner, J. ; Sax, E. : A Taxonomy and Survey on Validation Approaches for Automated Driving Systems. In: *IEEE ISSE 2020*. Piscataway, NJ, USA : IEEE, 2020. – ISBN 978–1–7281–8602–3, S. 1–8

- [55] Klein, M. : Definition: UNECE. In: *Springer Fachmedien Wiesbaden GmbH* (19.02.2018). <https://wirtschaftslexikon.gabler.de/definition/unece-33531>
- [56] Klendauer, R. ; Hoffmann, A. ; Leimeister, J. M. ; Berkovich, M. ; Krcmar, H. : Using the IDEAL software process improvement model for the implementation of Automotive SPICE. In: Charp, H. (Hrsg.): *2012 5th International Workshop on Co-Operative and Human Aspects of Software Engineering (CHASE 2012)*. Piscataway, NJ : IEEE, 2012. – ISBN 978–1–4673–1824–2, S. 66–72
- [57] Klessascheck, M. : V-Modell vs. Wasserfallmodell für Hardware- und Softwareentwicklung. In: *Johner Institut GmbH* (11.07.2018). <https://www.johner-institut.de/blog/iec-62304-medizinische-software/v-modell/>
- [58] Klischat, M. ; Althoff, M. : Generating Critical Test Scenarios for Automated Vehicles with Evolutionary Algorithms. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2019. – ISBN 978–1–7281–0560–4, S. 2352–2358
- [59] Koller, B. ; Düser, T. ; ATZ elektronik (Hrsg.): *Homologation und Validierung von automatisierten Fahrfunktionen*. <https://www.springerprofessional.de/en/homologation-und-validierung-von-automatisierten-fahrfunktionen/17791026>.
Version: 2020
- [60] Kruber, F. ; Wurst, J. ; Morales, E. S. ; Chakraborty, S. ; Botsch, M. : Unsupervised and Supervised Learning with the Random Forest Algorithm for Traffic Scenario Clustering and Classification. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2019. – ISBN 978–1–7281–0560–4, S. 2463–2470
- [61] Kurth, L. : *Straßenkreuzung mit Ampel*. <https://www.stonewars.de/>.
Version: 2022

- [62] Langner, J. ; Grolig, H. ; Otten, S. ; Holzäpfel, M. ; Sax, E. : Logical Scenario Derivation by Clustering Dynamic-Length-Segments Extracted from Real-World-Driving-Data. In: *5th International Conference on Vehicle Technology and Intelligent Transport Systems*, S. 458–467
- [63] Langner, J. ; Grolig, H. ; Otten, S. ; Holzäpfel, M. ; Sax, E. : Logical Scenario Derivation by Clustering Dynamic-Length-Segments Extracted from Real-World-Driving-Data. In: *Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems*, SCITEPRESS - Science and Technology Publications, 2019
- [64] learn scikit: *Gaussian Process Regression*. https://scikit-learn.org/stable/modules/gaussian_process.html. Version: 2007
- [65] Lehmann, V. ; QZ-online (Hrsg.): *Automotive SPICE für Einsteiger*. <https://www.qz-online.de/a/grundlagenartikel/automotive-spice-fuer-einsteiger-312480>. Version: 2020
- [66] Löcklin, A. ; Müller, M. ; Schuster, A. ; Vietz, H. ; Weyrich, M. ; TÜV Süd (Hrsg.): *Robo-Test: Kognitive Assistenz zur Validierung autonomer Fahrzeuge*
- [67] Maibaum, O. : *Entwicklungsbegleitender Test mechatronischer Systeme*. (2022). <https://elib.dlr.de/51061/1/071022Pr%C3%A4sentation.pdf>
- [68] Maurer, M. ; Gerdes, J. C. ; Lenz, B. ; Winner, H. : *Autonomes Fahren*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2015. <http://dx.doi.org/10.1007/978-3-662-45854-9>. <http://dx.doi.org/10.1007/978-3-662-45854-9>. – ISBN 978–3–662–45853–2
- [69] Menzel, T. ; Bagschik, G. ; Maurer, M. : *Scenarios for Development, Test and Validation of Automated Vehicles*. IEEE Intelligent Vehicles Symposium, 2018
- [70] Mishra, P. ; Pandey, C. M. ; Singh, U. ; Gupta, A. ; Sahu, C. ; Keshri, A. : Descriptive statistics and normality tests for statistical data. In: *Annals of Cardiac Anaesthesia* 22 (2019), Nr. 1, S. 67–72. <http://dx.doi.org/10.>

4103/aca.ACA_{_}157_{_}18. – DOI 10.4103/aca.ACA_157_18. – ISSN 0971-9784

- [71] Mitchell, T. M.: *Machine learning*. International ed., [Reprint.]. New York, NY : McGraw-Hill, 20]10 (McGraw-Hill series in computer science). – ISBN 0070428077
- [72] Moosbach, D. : *Repräsentativität - Wortbedeutung*. <https://www.wortbedeutung.info/Repr%C3%A4sentativit%C3%A4t/>. Version: 2023
- [73] Mora, L. ; Wu, X. ; Panori, A. : Mind the gap: Developments in autonomous driving research and the sustainability challenge. In: *Journal of Cleaner Production* 275 (2020), 124087. <http://dx.doi.org/10.1016/j.jclepro.2020.124087>. – DOI 10.1016/j.jclepro.2020.124087. – ISSN 0959-6526
- [74] Nautiyal, D. : ML | Underfitting and Overfitting. In: *GeeksforGeeks* (2017). <https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/>
- [75] Neal, R. M.: *Lecture notes in statistics*. Bd. 118: *Bayesian learning for neural networks: Zugl.: Toronto, Univ., Diss., 1995*. New York and Berlin and Heidelberg : Springer, 1996. <http://dx.doi.org/10.1007/978-1-4612-0745-0>. <http://dx.doi.org/10.1007/978-1-4612-0745-0>. – ISBN 978-0-387-94724-2
- [76] NHTSA: *Investigation Report*. <https://www.nts.gov/investigations/AccidentReports/Pages/Reports.aspx>. Version: 01.04.2025
- [77] Oliver BUEHLER ; Joachim WEGENER: *Evolutionary Functional Testing of an Automated Parking System* https://www.researchgate.net/publication/246122616_Evolutionary_Functional_Testing_of_an_Automated_Parking_System
- [78] OpenCV: *Introduction to Principal Component Analysis (PCA)*. https://docs.opencv.org/4.x/d1/dee/tutorial_introduction_to_pca.html. Version: 18.01.2024

- [79] Otten, S. ; Bach, J. ; Wohlfahrt, C. ; King, C. ; Lier, J. ; Schmid, H. ; Schmerler, S. ; Sax, E. : Automated Assessment and Evaluation of Digital Test Drives. In: Zachäus, C. (Hrsg.) ; Müller, B. (Hrsg.) ; Meyer, G. (Hrsg.): *Advanced Microsystems for Automotive Applications 2017*. Cham : Springer International Publishing, 2017 (Lecture Notes in Mobility). – ISBN 978–3–319–66972–4, S. 189–199
- [80] Pavel Senin: *Dynamic Time Warping Algorithm Review* https://www.researchgate.net/publication/228785661_Dynamic_Time_Warping_Algorithm_Review
- [81] PEGASUS: *PEGASUS METHOD*. 2020
- [82] Pegasus Projekt ; Bundesministerium für Wirtschaft und Energie (Hrsg.): *Szenarienbeschreibung*. 2017
- [83] Peterjohann, H. : *Das Wasserfallmodell - Peterjohann Consulting*. <https://www.peterjohann-consulting.de/wasserfallmodell/>. Version: 20.01.2023
- [84] Prof. Dr. Wolfgang Ludwig-Mayerhofer: *Vorlesung Statistik (Master): Punkt- und Intervallschätzung*. Siegen, Universität Siegen, Diss., 2024. https://www.uni-siegen.de/phil/sozialwissenschaften/soziologie/mitarbeiter/ludwig-mayerhofer/statistik/statistik_downloads/statistik_ii_4.pdf
- [85] Pukallus, S. ; Knobel, C. ; Wood, M. ; Wittmann, D. : Safety First for Automated Driving. <https://group.mercedes-benz.com/documents/innovation/other/safety-first-for-automated-driving.pdf>
- [86] Pütz, A. ; Fuchs, J. : *Risk-based selection of concrete (test) scenarios for a logical scenario*. https://www.vvm-projekt.de/fileadmin/user_upload/Final_Event/Poster/VVM_FE_Poster_06.04_Risk_Based_Selection_of_Concrete_Test_Scenarios_web.pdf. Version: 21.11.2023

- [87] Pütz, A. ; Zlocki, A. ; Bock, J. ; Eckstein, L. : System validation of highly automated vehicles with a database of relevant traffic scenarios. (2017). <https://www.semanticscholar.org/paper/System-validation-of-highly-automated-vehicles-with-P%C3%BCtz-Zlocki/85d664ad3238a96065c77465c9b7f9e16b300df1>
- [88] R. Pfeffer ; G. N. Basedow ; N. R. Thiesen ; M. Spadinger ; A. Albers ; E. Sax: Automated Driving - Challenges for the Automotive Industry in Product Development with Focus on Process Models and Organizational Structure. In: *2019 IEEE International Systems Conference (SysCon)*, 2019. – ISBN 2472–9647, S. 1–6
- [89] Rempe, D. ; Pillion, J. ; Guibas, L. J. ; Fidler, S. ; Litany, O. : Generating Useful Accident-Prone Driving Scenarios via a Learned Traffic Prior. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2022
- [90] Richter, H. ; Dieter, Z. : *Einführung in die Konzepte und Techniken im Automotiven Bereich - Free Download PDF*. Koblenz-Landau, Universität Koblenz-Landau, Seminar, 09.12.2021. <https://silo.tips/download/einfhrung-in-die-konzepte-und-techniken-im-automotiven-bereich>
- [91] Riedmaier, S. ; Ponn, T. ; Ludwig, D. ; Schick, B. ; Diermeyer, F. : Survey on Scenario-Based Safety Assessment of Automated Vehicles. In: *IEEE Access* 8 (2020), S. 87456–87477. <http://dx.doi.org/10.1109/ACCESS.2020.2993730>. – DOI 10.1109/ACCESS.2020.2993730. – ISSN 2169–3536
- [92] Ries, L. ; Rigoll, P. ; Braun, T. ; Schulik, T. ; Sax, E. : Trajectory-Based Clustering of Real-World Urban Driving Sequences with Multiple Traffic Objects, 2021, S. 1251–1258
- [93] Ries, L. ; Stumpf, M. ; Bach, J. ; Sax, E. : Semantic Comparison of Driving Sequences by Adaptation of Word Embeddings, 2020, S. 1–7
- [94] Rogowski, C. : Scrum Booklet. (2020)

- [95] Rote, G. : Computing the minimum Hausdorff distance between two point sets on a line under translation. In: *Information Processing Letters* 38 (1991), Nr. 3, S. 123–127. [http://dx.doi.org/10.1016/0020-0190\(91\)90233-8](http://dx.doi.org/10.1016/0020-0190(91)90233-8). – DOI 10.1016/0020–0190(91)90233–8. – ISSN 00200190
- [96] S. Ulbrich ; T. Menzel ; A. Reschka ; F. Schuldt ; M. Maurer: Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving. In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015. – ISBN 2153–0017, S. 982–988
- [97] SAE International: *J3016B: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. https://www.sae.org/standards/content/j3016_201806/. Version: 15.12.2021
- [98] Sakoe, H. ; Chiba, S. : Dynamic Programming Algorithm Optimization for Spoken Word Recognition. Version: 1978. <http://dx.doi.org/10.1016/B978-0-08-051584-7.50016-4>. In: Waibel, A. (Hrsg.) ; Lee, K.-F. (Hrsg.): *Readings in Speech Recognition*. Elsevier Science. – DOI 10.1016/B978–0–08–051584–7.50016–4. – ISBN 9781558601246, 159–165
- [99] Saubert, P. : Testing-Glossar – Fahrzeugprüfung. In: *Fahrzeugprüfung.org* (20.05.2018). <https://www.fahrzeugpruefung.org/testing-glossar/>
- [100] Sauerbier, J. ; Bock, J. ; Weber, H. ; Eckstein, L. : Definition von Szenarien zur Absicherung automatisierter Fahrfunktionen. In: *ATZ - Automobiltechnische Zeitschrift* 121 (2019), Nr. 1, 42–45. <http://dx.doi.org/10.1007/s35148-018-0205-2>. – DOI 10.1007/s35148–018–0205–2. – ISSN 2192–8800
- [101] Sax, E. (Hrsg.): *Automatisiertes Testen eingebetteter Systeme in der Automobilindustrie*. Hanser. <http://dx.doi.org/10.3139/9783446419018>. <http://dx.doi.org/10.3139/9783446419018>. – ISBN 9783446419018

- [102] Scholtes, M. ; Westhofen, L. ; Turner, L. R. ; Lotto, K. ; Schuldes, M. ; Weber, H. ; Wagener, N. ; Neurohr, C. ; Bollmann, M. ; Körtke, F. ; Hiller, J. ; Hoss, M. ; Bock, J. ; Eckstein, L. ; IEEE Access (Hrsg.): *6-Layer Model for a Structured Description and Categorization of Urban Traffic and Environment*. <https://arxiv.org/pdf/2012.06319.pdf>
- [103] Schwaber, K. ; Sutherland, J. : Scrum-Guide. (2020). <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf#zoom=100>
- [104] Segal, M. : *Machine learning benchmarks and random forest regression*. San Francisco, 2004 <https://escholarship.org/uc/item/35x3v9t4>
- [105] Seidel, D. ; Heppner, N. ; Kohlhoff, M. : Concept in the Loop: Validation in Early Development Process. In: *5. International Symposium fo Future Active Safety Technology toward Zero Accidents*. 2019
- [106] Shaphiro, S. ; Wilk, M. : *An analysis of variance test for normality*. 1965
- [107] Siebinga, O. ; Zgonnikov, A. ; Abbink, D. : *Automatic extraction of similar traffic scenes from large naturalistic datasets using the Hausdorff distance*. <https://arxiv.org/pdf/2206.13386>
- [108] Steimle, M. ; Bagschik, G. ; Menzel, T. ; Wendler, J. T. ; Maurer, M. : Anwendung eines Grundvokabulars für den szenarienbasierten Testansatz automatisierter Fahrfunktionen anhand eines Beispiels
- [109] Steimle, M. ; Menzel, T. ; Maurer, M. : *A Method for Classifying Test Bench Configurations in a Scenario-Based Test Approach for Automated Vehicles*. <https://arxiv.org/pdf/1905.09018>
- [110] Steimle, M. ; Menzel, T. ; Maurer, M. : *Toward a Consistent Taxonomy for Scenario-Based Development and Test Approaches for Automated Vehicles: A Proposal for a Structuring Framework, a Basic Vocabulary, and its Application*. <http://dx.doi.org/10.1109/ACCESS.2021.3123504>

- [111] Steimle, M. ; Weber, N. ; Maurer, M. : Toward Generating Sufficiently Valid Test Case Results: A Method for Systematically Assigning Test Cases to Test Bench Configurations in a Scenario-Based Test Approach for Automated Vehicles. In: *IEEE Access* 10 (2022), S. 6260–6285. <http://dx.doi.org/10.1109/ACCESS.2022.3141198>. – DOI 10.1109/ACCESS.2022.3141198. – ISSN 2169–3536
- [112] Strasser, B. ; Strasser, P. S. B.: *Audi-Dissertationsreihe*. Bd. 69: *Vernetzung von Test- und Simulationmethoden für die Entwicklung von Fahrerassistenzsystemen: Zugl.: München, Techn. Univ., Diss., 2012*. 1. Aufl. Göttingen : Cuvillier, 2012. – ISBN 978–3–95404–263–0
- [113] Swiler, L. P. ; Gulian, M. ; Frankel, A. L. ; Safta, C. ; Jakeman, J. D.: A SURVEY OF CONSTRAINED GAUSSIAN PROCESS REGRESSION: APPROACHES AND IMPLEMENTATION CHALLENGES. In: *Journal of Machine Learning for Modeling and Computing* 1 (2020), Nr. 2, S. 119–156. <http://dx.doi.org/10.1615/JMachLearnModelComput.2020035155>. – DOI 10.1615/JMachLearnModelComput.2020035155. – ISSN 2689–3967
- [114] Tatar, M. : Chasing critical situations in large parameter spaces. In: *Autonomous Vehicle Test & Development Symposium 2018*. https://www.pegasusprojekt.de/files/tmpl/pdf/Tatar_AVTD_Symposium_2018.pdf
- [115] Tatar, M. : Test and Validation of Advanced Driver Assistance Systems Automated Search for Critical Scenarios. In: *ATZelektronik worldwide* 11 (2016), Nr. 1, 54–57. <http://dx.doi.org/10.1007/s38314-015-0574-1>. – DOI 10.1007/s38314-015-0574-1. – ISSN 2192–9092
- [116] Theile, G. : Für S-Klasse und EQS: Mercedes bringt automatisiertes Fahren auf den Markt. In: *Frankfurter Allgemeine Zeitung* (06.05.2022). <https://www.faz.net/aktuell/wirtschaft/digitec/mercedes-bietet-als-erster-hersteller-autonomes-fahren-auf-level-3-an-18009871.html>

- [117] Tuncali, C. E. ; Fainekos, G. ; Prokhorov, D. ; Ito, H. ; Kapinski, J. : Requirements-Driven Test Generation for Autonomous Vehicles With Machine Learning Components. In: *IEEE Transactions on Intelligent Vehicles* 5 (2020), Nr. 2, S. 265–280. <http://dx.doi.org/10.1109/TIV.2019.2955903>. – DOI 10.1109/TIV.2019.2955903. – ISSN 2379–8858
- [118] UNECE ; UNECE (Hrsg.): *Vehicle Regulations*. <https://unece.org/transport/vehicle-regulations>. Version: 2023
- [119] Verband der Automobilindustrie e.V.: *Reifegradabsicherung für Neuteile*. Berlin, 2021
- [120] VV-Methoden Projekt: *Strukturierung und Aufbau von Szenarien*. Karlsruhe,
- [121] Wachenfeld, W. ; Junietz, P. ; Winner, H. ; Themann, P. ; Pütz, A. : *Safety Assurance Based on an Objective Identification of Scenarios*. San Francisco, 2016
- [122] Wachenfeld, W. ; Winner, H. : Die Freigabe des autonomen Fahrens. Version: 2015. http://dx.doi.org/10.1007/978-3-662-45854-9_{_}21. In: *Autonomes Fahren*. Springer Vieweg, Berlin, Heidelberg. – DOI 10.1007/978-3-662-45854-9_21, 439–464
- [123] Westhofen, L. ; Neurohr, C. ; Koopmann, T. ; Butz, M. ; Schütt, B. ; Utesch, F. ; Kramer, B. ; Gutenkunst, C. ; Böde, E. : *Criticality Metrics for Automated Driving: A Review and Suitability Analysis of the State of the Art*. <http://arxiv.org/pdf/2108.02403v1>
- [124] Westhofen, L. ; Neurohr, C. ; Koopmann, T. ; Butz, M. ; Schütt, B. ; Utesch, F. ; Neurohr, B. ; Gutenkunst, C. ; Böde, E. : Criticality Metrics for Automated Driving: A Review and Suitability Analysis of the State of the Art. In: *Archives of Computational Methods in Engineering* 30 (2023), Nr. 1, 1–35. <http://dx.doi.org/10.1007/s11831-022-09788-7>. – DOI 10.1007/s11831-022-09788-7. – ISSN 1134–3060

- [125] Winkelmann, M. ; Kohlhoff, M. ; Tadjine, H. H. ; Müller, S. : *Probabilistic Metamodels for an Efficient Characterization of Complex Driving Scenarios*.
<https://arxiv.org/pdf/2110.02892>
- [126] Zhang, Z. ; Huang, K. ; Tan, T. : Comparison of Similarity Measures for Trajectory Clustering in Outdoor Surveillance Scenes. In: Tang, Y. Y. (Hrsg.): *18th International Conference on Pattern Recognition, 2006*. Los Alamitos, Calif. : IEEE Computer Society, 2006. – ISBN 0–7695–2521–0, S. 1135–1138

A Anhang

Ausgewählte Methodenbausteine für Selektionsprozess		Ermittlung der Risikoprioritätszahl												
		Annahmen und Singularität				Schätzung der Vorhersagegenauigkeit				Nicht-deterministisches Verhalten				RPZ Methodenbausteins
		Exposure E	Controllability C	Severity S	resultierende RPZ	Exposure E	Controllability C	Severity S	resultierende RPZ	Exposure E	Controllability C	Severity S	resultierende RPZ	
M1-1: Äquidistante kombinatorische Erstellung	Begründung	5) E3	C8	S9	216	3) E1	C1	S9	9	4) E1	C1	S9	9	216,4
M2-1: Einschränkung des Parameterraum auf Basis von logischen und physikalischen Zusammenhängen - Restriktion	Begründung	7) E7	C2	S9	126	3) E1	C1	S9	9	4) E1	C1	S9	9	126,6
M2-2: Reduktion und Zusammenfassung der Parameter der Parameter auf Basis von logischen und physikalischen Zusammenhängen	Begründung	9) E8	C1	S9	72	3) E1	C1	S9	9	4) E1	C1	S9	9	73,1
M3-1: Erste Sampling Strategie	Begründung	2) E1	C1	S9	9	3) E1	C1	S9	9	4) E1	C1	S9	9	15,6
M3-3: Unsicherheits- und Ähnlichkeitsanalyse	Begründung	11) E8	C5	S9	360	13) E8	C4	S9	288	15) E8	C4	S9	288	543,6

Restunsicherheit Gesamt Selektionsprozess 975,3

Abbildung A.1: Einstufung und Begründung der RPZ für das ausgewählte Logische Szenario

Ausführung zu den Begründungen:

- 1) **S9:** Wird ein Szenario übersehen, bei dem ein Fußgänger plötzlich die Straße betritt, kann dies im realen Betrieb zu einem Unfall mit schweren bis tödlichen Verletzungen führen. Da der Fußgänger im Gegensatz zu Fahrzeuginsassen keinerlei Schutzsysteme wie Rückhalteeinrichtungen oder Knautschzonen besitzt, ist seine Verletzungsanfälligkeit im Kollisionsfall besonders hoch. Unfallstatistiken aus Quellen wie GIDAS, BAST oder internationalen Datenbanken (z.B. EU CARE, NHTSA) belegen, dass Zusammenstöße zwischen Fahrzeugen und Fußgängern insbesondere bei Geschwindigkeiten über 30 km/h mit hoher Wahrscheinlichkeit zu schweren oder tödlichen Verletzungen führen. Zudem wird in regulatorischen Rahmenwerken wie UN/ECE R152 implizit davon ausgegangen, dass das Nichterkennen eines Fußgängers ein sicherheitskritisches Versäumnis darstellt, das mit schwerwiegenden Folgen verbunden ist. Vor diesem Hintergrund ist die Bewertung des Szenarios mit einer hohen Severity gemäß ISO 26262 nachvollziehbar.
- 2) **C1/E1:** In diesem Methodenbaustein werden keine Annahmen getroffen.
- 3) **C1/E1:** In diesem Methodenbaustein werden keine Schätzungen zur Vorhersagegenauigkeit vorgenommen.
- 4) **E1:** Der Methodenbaustein verhält sich deterministisch.
- 5) **E3:** Die Diskretisierungsschrittweite des Methodenbausteins ist auf 0,5 km/h, 0,2 m/s und 0,5 m festgelegt.

Die betrachtete Funktion des dSpace-Notbremsassistenten ist in einem Simulink-Modell implementiert, das die Messgrößen mit einer gröberen oder mindestens gleichwertigen Auflösung verarbeitet. Somit ist sichergestellt, dass die Abtastung keine Genauigkeitsverluste gegenüber der gewählten Diskretisierung aufweist und die Reduktion valide für die weitere Analyse genutzt werden kann.

- 6) **C8:** Eine zu grob gewählte Diskretisierungsschrittweite kann dazu führen, dass kritische Zustandsänderungen oder relevante Signalverläufe innerhalb des Modells nicht adäquat erfasst werden. Insbesondere in sicherheitskritischen Funktionen wie dem Notbremsassistenten, die auf Schwellenwertbedingungen oder nichtlinearen Zustandsübergängen basieren, besteht die Gefahr, dass durch das Überspringen einzelner Zustände unerwünschte Effekte wie numerische Singularitäten oder logische Fehlreaktionen auftreten. So kann es beispielsweise vorkommen, dass ein Eingriff des Notbremsassistenten aufgrund einer unzureichend aufgelösten Time-to-Collision-Berechnung nicht ausgelöst wird, obwohl objektiv eine Gefahrensituation vorliegt. Solche Fehlfunktionen bleiben bei grober Diskretisierung häufig unentdeckt, da der Simulationsverlauf keine Anzeichen für unstetige oder fehlerhafte Systemreaktionen zeigt. Erst durch eine feinere Abtastung, etwa im Rahmen eines gezielten Resamplings, werden diese potenziellen Schwachstellen im Modell sichtbar. Eine hinreichend feine Diskretisierung ist daher essenziell, um die Korrektheit und Robustheit des modellbasierten Systems sicherzustellen und die vollständige Abdeckung kritischer Szenarien im Rahmen der Absicherung zu gewährleisten.
- 7) **E7:** Die manuell formulierten Restriktionen dienen der Reduktion des Parameterraums durch den Ausschluss offensichtlich unplausibler Szenarien. Dabei besteht jedoch die Gefahr, dass durch vereinfachte Annahmen auch sicherheitskritische, aber realistische Situationen entfernt werden. In solchen Fällen erfolgt die Sichtbarkeitsänderung abrupt, und die Annahmen aus Restriktion 2 (z.B. positiver Abstand beim Betreten) greifen nicht mehr. Ebenso kann die vereinfachte Bremswegberechnung aus Restriktion 1 bei komplexeren Reaktionsdynamiken oder Systemverzögerungen nicht mehr adäquat angewendet werden. Daraus ergibt sich die Notwendigkeit, die Restriktionen kritisch zu hinterfragen und ihre Auswirkungen auf die Repräsentativität und Vollständigkeit der Szenarienabdeckung zu evaluieren. Andernfalls besteht die Gefahr, dass für die Absicherung relevante Extremsituationen systematisch ausgeschlossen werden.

- 8) **C2:** Die Validität der formulierten Restriktionen auf die *Konkreten Szenarien* können durch eine nachgelagerte qualitative Validierung überprüft werden. Hierzu werden ausgewählte, durch die Restriktionen ausgeschlossene *Konkrete Szenarien* exemplarisch ausgeführt (vgl. C2). Ziel ist es, die Annahmen der jeweiligen Restriktion an der Testplattform zu verifizieren und potenziell fälschlich ausgeschlossene zu identifizieren. Erkenntnisse aus dieser Validierung können zur iterativen Verbesserung der Restriktionen beitragen und erhöhen somit die methodische Absicherung der Reduktionsschritte.
- 9) **E8:** Zwar besteht die Möglichkeit, dass durch eine fehlerhafte Modellierung des Szenarioablaufs – etwa infolge unvollständiger oder vereinfachter Annahmen (z.B. konstante Geschwindigkeit des VRU) – Situationen falsch zusammengefasst und somit ausgeschlossen werden. Allerdings wurde im Rahmen der Umsetzung dieses Methodenbausteins das theoretisch mögliche Reduktionspotenzial bewusst nicht vollständig ausgereizt. Pro Cluster wurden bis zu 40 Repräsentanten zufällig selektiert, wodurch auch atypische Randfälle erhalten bleiben. In Summe wurden nur 7.513 *Konkrete Szenarien* ausgeschlossen (von 20.894 ausstehenden), was einer vergleichsweise konservativen Reduktion entspricht. Das Risiko, durch die Parametrisierung sicherheitskritische Situationen zu verlieren, ist daher als sehr gering einzuschätzen, insbesondere im Vergleich zum gewonnenen Reduktionsnutzen im Testaufwand.
- 10) **C1:** Eine gute Entdeckbarkeit sicherheitskritischer Situationen ist in diesem Methodenbaustein (MII-2) weiterhin gewährleistet. Trotz der Anwendung einer dimensionsreduzierenden Parametrierung wurden nur 7.513 *Konkrete Szenarien* ausgeschlossen, während aus jedem der resultierenden 19 Cluster bis zu 40 Szenarien zufällig ausgewählt wurden. Diese bewusste Überrepräsentation stellt sicher, dass auch Rand- und Sonderfälle eines Clusters erfasst werden. Durch die Anwendung der Fehlerfortpflanzung zur Wahl der Diskretisierung und die begrenzte Reduktion ergibt sich eine hohe Abdeckung des Szenarienraums, was eine sehr gute Entdeckbarkeit gewährleistet.

- 11) **E8:** Die Auswahl und Bewertung der *Konkreten Szenarien* im letzten Methodenbaustein erfolgt auf Basis der vorhergesagten bzw. gemessenen KPIs. Um die Aussagekraft der Reduktionsentscheidung zu legitimieren, muss sichergestellt sein, dass die verwendeten KPIs das sicherheitsrelevante Verhalten des Systems unter Test (SuT) im betrachteten Szenarioraum ausreichend repräsentieren. Nur unter dieser Voraussetzung kann davon ausgegangen werden, dass Szenarien mit ähnlichen KPI-Ausprägungen auch ein vergleichbares Verhalten des SuT aufweisen.
- 12) **C5:** Die Repräsentativität der ausgewählten KPIs für das Verhalten des SuT lässt sich nur anhand der Anforderungen überprüfen. Diese Anforderungen definieren die relevanten Szenarien und Qualitätskriterien, an denen sich das System messen lassen muss. Ohne eine solche Anforderungsbasis kann nicht sichergestellt werden, dass die KPIs tatsächlich alle sicherheits- und funktionsrelevanten Aspekte des Systems abbilden. Daher ist die Validierung der KPIs im Kontext der Anforderungen notwendig, um eine aussagekräftige Bewertung und Interpretation der Testergebnisse zu gewährleisten.
- 13) **E8:** Die Anzahl der Repräsentativitätscluster ist vorab unbekannt. Daher entscheidet eine Annahme über die Ähnlichkeit zwischen zwei *Konkreten Szenarien* über die Reduktion der Szenarien.
- 14) **C4:** Das anschließende Clustering-Verfahren verhält sich nicht deterministisch, was bedeutet, dass sich die Clusterbildung bei unterschiedlichen Läufen leicht unterscheiden kann. Durch die Erhöhung der Anzahl an Szenarien pro Cluster wird die Fehlertoleranz verbessert, da so mehr potenziell relevante Fälle berücksichtigt werden. Dies ermöglicht eine robustere Erkennung von Fehlern und sorgt dafür, dass auch seltene, aber kritische Szenarien nicht übersehen werden.
- 15) **E8:** Die genaue Anzahl der Repräsentativitätscluster ist im Vorfeld nicht bekannt, da die Verteilung der *Konkreten Szenarien* im KPI-Raum variabel und von der zugrundeliegenden Szenariendynamik abhängig ist. Um dennoch eine sinnvolle Reduktion durchzuführen, muss eine Annahme über die

Ähnlichkeit zwischen zwei *Konkreten Szenarien* getroffen werden. Diese Annahme legt fest, ab welchem Maß an Ähnlichkeit Szenarien als repräsentativ für ein Cluster angesehen und somit zusammengefasst werden können. Die Annahme über die Ähnlichkeit ist deshalb entscheidend, da sie direkt die Anzahl und Struktur der Cluster beeinflusst und damit maßgeblich bestimmt, wie stark die Reduktion der Szenarien ausfällt, ohne die Repräsentativität zu verlieren.

- 16) **C4:** Das Clustering-Verfahren ist aufgrund seiner stochastischen Initialisierung (z. B. bei k-Means zufällige Startzentren) nicht deterministisch, was zu unterschiedlichen Clusterzuordnungen bei wiederholter Ausführung führen kann. Durch die Auswahl einer höheren Anzahl von *Konkreten Szenarien* pro Cluster wird die Stichprobe innerhalb des Clusters repräsentativer und weniger anfällig für Ausreißer oder zufällige Varianz. Dadurch erhöht sich die Robustheit gegenüber zufälligen Fehlern in der Clusterzuordnung. Dies verbessert die Fehlertoleranz im Selektionsprozess und erhöht die Wahrscheinlichkeit, potenzielle Fehler oder Randfälle im Parameterraum zuverlässig zu entdecken.