

Methodik für den Systemtest in der integralen Fahrzeugsicherheit

Dissertation

zur Erlangung des akademischen Grades

**Doktoringenieurin
(Dr.-Ing.)**

von Dipl.-Ing. (FH) Kathrin Sattler

geboren am 09. September 1986

in Regensburg

genehmigt durch die Fakultät für Elektrotechnik und Informationstechnik
der Otto-von-Guericke-Universität Magdeburg

Gutachter Prof. Dr.-Ing. Christian Diedrich
Prof. Dr.-Ing. Birgit Vogel-Heuser
Prof. Dr.-Ing. Thomas Brandmeier

Promotionskolloquium am 15. Juli 2015

Ehrenerklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die Hilfe eines kommerziellen Promotionsberaters habe ich nicht in Anspruch genommen. Dritte haben von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen. Verwendete fremde und eigene Quellen sind als solche kenntlich gemacht.

Ich habe insbesondere nicht wissentlich:

- Ergebnisse erfunden oder widersprüchliche Ergebnisse verschwiegen,
- statistische Verfahren absichtlich missbraucht, um Daten in ungerechtfertigter Weise zu interpretieren,
- fremde Ergebnisse oder Veröffentlichungen plagiiert,
- fremde Forschungsergebnisse verzerrt wiedergegeben.

Mir ist bekannt, dass Verstöße gegen das Urheberrecht Unterlassungs- und Schadensersatzansprüche des Urhebers sowie eine strafrechtliche Ahndung durch die Strafverfolgungsbehörden begründen kann.

Ich erkläre mich damit einverstanden, dass die Dissertation ggf. mit Mitteln der elektronischen Datenverarbeitung auf Plagiate überprüft werden kann.

Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form als Dissertation eingereicht und ist als Ganzes auch noch nicht veröffentlicht.

Ingolstadt, 11.03.2015

Kathrin Stephanie Sattler

Danksagung

Die vorliegende Arbeit entstand im Rahmen meiner Tätigkeit als wissenschaftliche Mitarbeiterin am Institut für Fahrzeugsicherheitssysteme (AUREA) am Zentrum für Angewandte Forschung (ZAF) der Technischen Hochschule Ingolstadt.

Hiermit möchte ich mich bei allen bedanken, die mich während der Promotion begleitet und unterstützt und somit zum Gelingen dieser Arbeit beigetragen haben.

Bei Herrn Prof. Dr.-Ing. Thomas Brandmeier, Forschungsprofessor für Fahrzeugsicherheit, Stellvertretender wissenschaftlicher Leiter des Zentrums für Angewandte Forschung (ZAF) und wissenschaftlicher Leiter von CARISSMA, möchte ich mich für das entgegengebrachte Vertrauen, die zahlreichen Diskussionen, die Betreuung und seine fortlaufende Unterstützung bedanken. Mein Werdegang wurde durch die durch ihn gegebene Möglichkeit zur Promotion einschlägig positiv beeinflusst.

Großen Dank möchte ich Herrn Prof. Dr.-Ing. Christian Diedrich, dem Leiter des Lehrstuhls Integrierte Automation an der Otto-von-Guericke-Universität Magdeburg und stellvertretendem Institutsleiter des ifak e.V. in Magdeburg, für die Ermöglichung einer kooperativen Promotion und die Betreuung der Arbeit seitens der Otto-von-Guericke-Universität Magdeburg. Auch bei ihm möchte ich mich für die interessanten Diskussionen und den sehr wertvollen Beitrag bedanken.

Frau Prof. Dr.-Ing. Birgit Vogel-Heuser, der Ordinaria des Lehrstuhls für Automatisierung und Informationssysteme an der Technischen Universität München, danke ich herzlich für ihr Interesse an meiner Arbeit und der Übernahme des Gutachtens.

Bei meinen Kollegen am Zentrum für Angewandte Forschung bedanke ich mich für die angenehme Arbeitsatmosphäre und die hervorragende Zusammenarbeit. Zudem möchte ich allen Mitarbeitern und Studenten der Technischen Hochschule Ingolstadt danken, die mich in irgendeiner Art und Weise während meiner Arbeit unterstützt haben.

Mein Dank gilt auch allen Projektpartnern für das große entgegengebrachte Vertrauen, die intensive, fruchtbare und kollegiale Zusammenarbeit, die Bereitstellung von Versuchsdaten und generell das Beisteuern von wichtigen Erfahrungen und Impulsen aus der industriellen Anwendung.

Nicht zuletzt möchte ich meinem Partner Daniel, meiner Familie und meinen Freunden für die Unterstützung und Motivation bei der Entstehung dieser Arbeit danken. Durch ihren Beitrag waren sie mir ein ständiger Rückhalt und trugen somit wesentlich zum Gelingen dieses Lebensabschnitts bei.

*Wenn man etwas nicht weiß, so kann man fragen;
wenn man etwas nicht kann, so kann man es lernen.*

Lü Buwei, chinesischer Philosoph

Zusammenfassung

Im Rahmen dieser Dissertation wird eine Methodik für den Systemtest von Sicherheitssteuergeräten der integralen Fahrzeugsicherheit erarbeitet. Ziel der Arbeit ist es, durch Erprobung innovativer Testmethoden und einer Testmetrik Impulse für den Systemtest in der integralen Fahrzeugsicherheit zu setzen.

Durch die steigende Integration und Vernetzung kontemporärer Sicherheitssysteme im Automobil stellt das Testen in diesem Bereich eine große Herausforderung dar. Da die Kombination aller möglichen Fehlerfälle und die Ausführung aller Testfälle aus Zeit- und Kapazitätsgründen nicht möglich ist, müssen Methoden intelligent eingesetzt und kombiniert werden. Gerade beim anforderungsbasierten Systemtest von Sicherheitssteuergeräten existiert eine Lücke zwischen realen Testfahrten und dem Modul- und Integrationstest. Der für den Automobilbereich rahmengebende Standard ISO 26262 „Road vehicles - Functional safety“ liefert hierfür keine detaillierten Lösungsansätze. Durch den Einsatz von simulativem, manöverbasiertem Testen und somit der synthetischen Dateneinspeisung kann mithilfe von Hardware-in-the-Loop-Prüfständen der Systemtest von Steuergeräten flexibler, genauer sowie zeit- und kostensparend gestaltet werden.

Der erarbeitete Ansatz der Methodik der Dissertation ist, eine Bewertungsmetrik durchzuführen, anhand derer verschiedene Testmethoden und deren Einsatzumfang für den anforderungsbasierten Systemtest ausgewählt werden. Betrachtet wird auch die Korrelation aus Komplexität eines Projekts und der zugehörigen, aufgetretenen Fehler im Systemtest. Die Testmethode simulatives, manöverbasiertes Testen wird in dieser Arbeit zum ersten Mal in Kombination mit den ergänzenden Analyse- und Optimierungsmethoden statistische Testfallanalyse aus Design of Experiments und dem evolutionären Algorithmus Differential Evolution eingesetzt und die Testfallerstellung anhand mehrerer Beispiele erprobt. Durch Priorisierung der Testfälle kann anschließend ein Effektivitätsgewinn hinsichtlich der Durchführung auf einem geeigneten Testsystem erfolgen. Abschließend kann anhand des sogenannten Vertrauensgrades, der eine Testmetrik darstellt und ein Expertenaudit zur Testqualitätsbewertung sowie Fehlerschätzung, Fehlerdichte und Testüberdeckung beinhaltet, eine Aussage über den aktuellen Testfortschritt getroffen werden. Dieser Ansatz wurde anhand mehrerer Projekte aus der Automobilbranche erprobt.

Die der Arbeit zugrunde liegenden Projekte wurden mit Mitteln des Bundesministeriums für Wirtschaft und Technologie aufgrund eines Beschlusses des Deutschen Bundestages unter den Förderkennzeichen KF2122303RA9 (Projekt „Vernetzung und Integration von Sicherheitssystemen der Aktiven und Passiven Sicherheit VISAPS“, 2009 bis 2011) und KF2122306PR1 (Projekt VISAPS², 2012 bis 2014) gefördert.

Abstract

This dissertation suggests a method for the system test of safety control units in integrated safety. The objective is to introduce new approaches for the system test of integrated safety systems by proving innovative test methods and a test metric.

Testing of contemporary automotive safety systems is a huge challenge due to increasing integration and networking. Since all combinations of potential errors and the execution of all possible test cases is not feasible due to time and capacity constraints, methods have to be used and combined intelligently. Especially in the requirements-based system test exists a gap between real test drives and the module and integration test. The for the automotive field very important standard ISO 26262 „Road vehicles - Functional safety“ does not deliver any detailed approaches. With the adoption of simulative maneuver-based testing and therefore the feeding of synthetic data via Hardware-in-the-Loop-test-systems the system test of control units can be devised more flexible, accurate and time and cost saving.

The in the dissertation developed approach is to execute an assessment metric that is the basis for selecting different test methods and the extent of their use for the requirements-based system test. Also considered is the correlation of the complexity of a project and the related occurred errors during the system test. The test method simulative maneuver-based testing is here used in combination with the additional analysis and optimization methods statistical test case analysis Design of Experiments and the evolutionary algorithm Differential Evolution for the first time. The test case creation is proved by several examples. Afterwards with prioritization of test cases an efficiency advantage can be reached at test execution with an adequate test system. Concluding the so called confidence level, a test metric consisting of an experts audit for test quality assessment, error estimation, error concentration and test coverage, can state an overview of the current test progress. This approach was proved with several projects of the automotive field.

This research was funded by the Federal Ministry of Economics and Technology (Grant Number KF2122303RA9, Project „Vernetzung und Integration von Sicherheitssystemen der Aktiven und Passiven Sicherheit VISAPS“, 2009 to 2011 and Grant Number KF2122306PR1, Project VISAPS², 2012 to 2014) based on a decision of the German Bundestag.

Inhaltsverzeichnis

Zusammenfassung	ix
Abstract	xi
Inhaltsverzeichnis	xv
Symbolverzeichnis	xix
1. Einleitung und Aufbau der Arbeit	1
1.1. Verkehrs- und Unfallgeschehen	2
1.2. Zielsetzung und Aufbau der Arbeit	4
2. Grundlagen und Stand der Technik	7
2.1. Integrale Fahrzeugsicherheit	8
2.2. Steuergeräte	10
2.2.1. Sensoren	12
2.2.2. Aktoren	13
2.2.3. Aktuelle Entwicklung	14
2.3. ISO 26262	14
2.3.1. Überblick	15
2.3.2. Testmethoden nach ISO 26262	18
2.4. Der Systemtest im Entwicklungs- und Testprozess der Automobilbranche .	20
2.4.1. Vorgehensmodelle	22
2.4.2. Anforderungen	27
2.4.3. Teststrategie	31
2.4.4. Testdokumentation	33
2.4.5. Testmethoden	35
2.4.6. Testsysteme	39
2.5. Notwendigkeit der Arbeit	42
Fazit	47
3. Komplexitätsbewertung und Fehlerschätzung	49
3.1. Warum Kennzahlen und Indikatoren?	50
3.2. Expertenaudit zur Projektkomplexitätsbewertung	51
3.2.1. Die Projektkomplexität beeinflussende Faktoren	52
3.2.2. Gewichtung der Faktoren	54

3.2.3. Berechnung der Projektkomplexität	54
3.3. Fehlerschätzung	56
3.4. Beispiel	56
Fazit	59
4. Testauswahl, -methoden und -priorisierung	61
4.1. Testauswahl	63
4.2. Testmethode manöverbasiertes Testen	65
4.2.1. Einleitung	65
4.2.2. Notwendigkeit und Anwendung	66
4.2.3. Beispiel	68
4.2.4. Fazit	71
4.3. Testmethode statistische Analyse von Testfällen	72
4.3.1. Einleitung	72
4.3.2. Prinzipielle Vorgehensweise	74
4.3.3. Anwendung	77
4.3.4. Fazit	87
4.4. Testmethode Differential Evolution	88
4.4.1. Einleitung	88
4.4.2. Prinzipielle Funktionsweise der DE	90
4.4.3. Mathematische Abbildung der DE und Pseudo-Code	90
4.4.4. Anwendung und Problemstellung	95
4.4.5. Parametrierung	97
4.4.6. Beispiel 1: Eisbremsung	98
4.4.7. Beispiel 2: μ -Split-Bremsung	105
4.4.8. Fazit	112
4.5. Testfallpriorisierung	112
4.5.1. Ziele	112
4.5.2. Maßnahmen	113
Fazit	115
5. Testsystem	117
5.1. Anforderungen an die Testumgebung	118
5.2. Kombination Fahrdynamiksimulation und Crashdateneinspeisung	119
5.3. Beispielkonfiguration für ein Hardware-in-the-Loop-Testsystem	120
Fazit	122
6. Metrik zur Testfortschrittsbeurteilung	123
6.1. Testqualitätsbewertung	124
6.1.1. Die Testqualität beeinflussende Faktoren	125
6.1.2. Gewichtung der Faktoren	128
6.1.3. Berechnung der Testqualität	129
6.1.4. Beispiel	129

6.2. Testüberdeckung	129
6.2.1. Funktions- bzw. Funktionalitätsüberdeckung	130
6.2.2. Anforderungsüberdeckung	131
6.2.3. Fehlerdichte und Fehlerüberdeckung	131
6.2.4. Beispiele für Testüberdeckungen	132
6.3. Der Vertrauensgrad als Testmetrik	134
6.3.1. Zusammenführung der Maße	134
6.3.2. Beispiel	136
Fazit	136
7. Zusammenfassung und Ausblick	139
7.1. Zusammenfassung	140
7.2. Wissenschaftlicher Beitrag dieser Arbeit	141
7.3. Ausblick	142
A. Anhang - Statistische Versuchsplanung: Einzelauswertungen der Faktoren nach Faktorstufen	145
B. Anhang - Differential Evolution	151
B.1. Beispiel 1: Vergleich Ausführung mit und ohne vorhergehende statistische Analyse	151
B.2. Beispiel 1: Vergleich Variation der Kontrollparameter Cr und F	154
B.3. Beispiel 2: Wiederholung des Versuchs	160
Definitionsverzeichnis	164
Abbildungsverzeichnis	167
Tabellenverzeichnis	170
Literaturverzeichnis	171

Symbolverzeichnis

Abkürzungen

<i>Abkürzung</i>	<i>Bedeutung</i>
ADAS	Advanced Driver Assistance System
ASIL	Automotive Safety Integrity Level
AUTOSAR	Automotive Open System Architecture
CAN	Controller Area Network
CMM	Capability Maturity Model Integration
DE	Differential Evolution
DoE	Design of Experiments
EA	Evolutionäre Algorithmen
E/E-System	Elektrik- und Elektronik-System
EU	Europäische Union
FAS	Fahrerassistenzsystem
FIT	Failure in Time
HiL	Hardware-in-the-Loop
IC	Integrated Circuit
IEC	International Electrotechnical Commission
HW	Hardware
ISO	International Organization for Standardization
LIN	Local Interconnect Network
MiL	Model-in-the-Loop
MOST	Media Oriented Systems Transport
PSI5	Peripheral Sensor Interface 5
OEM	Original Equipment Manufacturer
QM	Quality Management
SiL	Software-in-the-Loop
SPI	Serial Peripheral Interface
SPICE	Software Process Improvement and Capability Determination
ASPICE	Automotive Software Process Improvement and Capability Determination
SUT	System Under Test
SW	Software
ViL	Vehicle-in-the-Loop

Formelzeichen

<i>Formelzeichen</i>	<i>Einheit</i>	<i>Bedeutung</i>
$B_{K,i}$	-	Bewertung eines Komplexitätseinflussfaktors
$B_{Q,i}$	-	Bewertung eines Qualitätseinflussfaktors
C	-	Controllability
Cr	-	Crossoverrate
D	-	Dimension
E	-	Exposure
F	-	Skalar oder Fehleranzahl
f	-	Funktion
f	-	frequency
G	-	Generation
G_{max}	-	maximale Anzahl der Generationen
$G_{K,i}$	-	gewichteter Einfluss eines Faktors auf die Projektkomplexität
$G_{Q,i}$	-	gewichteter Einfluss eines Faktors auf die Testqualität
g	m/s ²	Erdbeschleunigung
i	-	Zähler der Dimension der Generation
i	-	Zähler der Faktoren
j	-	Zähler der Vektoren der aktuellen Generation
j_{rand}	-	zufällig gewählter Index
K	-	Komplexitätszahl
N	-	Größe einer Population
N_+	-	Anzahl der Versuchsnummern mit oberer Faktorstufe +
N_0	-	Anzahl der Versuchsnummern mit mittlerer Faktorstufe 0
N_-	-	Anzahl der Versuchsnummern mit unterer Faktorstufe -
n_K	-	Anzahl der Einflussfaktoren für die Komplexität
n_Q	-	Anzahl der Einflussfaktoren für die Qualität
Q	-	Qualitätszahl
R	-	Risk
\mathfrak{R}^D	-	D -dimensionaler, realer Parameterraum
r_1^i, r_2^i, r_3^i	-	zufällig gewählte ganzzahlige Werte
$rand$	-	gleichmäßig verteilte Zufallsfunktion
S	-	Severity
SP	-	Schwerpunkt
v	m/s	Geschwindigkeit im Schwerpunkt
$\vec{U}_{i,G}$	-	i -ter Trial-Vektor der Generation G
$\vec{V}_{i,G}$	-	i -ter Donor-Vektor der Generation G

$\vec{X}_{i,G}$	-	i -ter Parametervektor der Generation G , Target-Vektor
\vec{X}_{max}	-	maximaler Parametervektor
\vec{X}_{min}	-	minimaler Parametervektor
$\vec{X}_{r_1^i}, \vec{X}_{r_2^i}, \vec{X}_{r_3^i}$	-	Parametervektoren
$\vec{X}_{r_1^i,G}, \vec{X}_{r_2^i,G}, \vec{X}_{r_3^i,G}$	-	Parametervektoren der Generation G
$u_{j,i,G}$	-	j -tes Element des i -ten Trial-Vektors der Generation G
$v_{j,i,G}$	-	j -tes Element des i -ten Donor-Vektors der Generation G
x_i	-	i -ter Faktor der statistischen Versuchsplanung
$x_{j,i,G}$	-	j -tes Element des i -ten Parametervektors der Generation G
$x_{j,min}$	-	j -tes Element des maximalen Parametervektors
$x_{j,min}$	-	j -tes Element des minimalen Parametervektors
$-\ddot{x}$	m/s ²	longitudinale Verzögerung
$y(x_i)$	-	Versuchsergebnis des Faktors x_i
z	-m/s ²	Abbremsung
β	rad	Schwimmwinkel
$\dot{\beta}$	rad/s	Schwimmwinkelgeschwindigkeit
μ	-	Reibwert
ψ	rad	Gierwinkel
$\dot{\psi}$	rad/s	Giergeschwindigkeit oder Gierrate

1. Einleitung und Aufbau der Arbeit

Dieses Kapitel leitet zum Themengebiet anhand des aktuellen Verkehrs- und Unfallgeschehens ein. Es folgt überblickend die Zielsetzung der vorliegenden Arbeit und deren Aufbau.

1.1. Verkehrs- und Unfallgeschehen

Verkehrs- und Fahrzeugsicherheit ist ein Thema, das weltweit diskutiert wird und als so brisant und wichtig erachtet wird, dass Ban Ki-moon als Generalsekretär der Vereinten Nationen am 11. Mai 2011 das „Decade of Action for Road Safety 2011 - 2020“, also das Jahrzehnt für Maßnahmen in der Verkehrssicherheit, ausgerufen hat. Ungefähr 1,3 Millionen Menschen sterben jährlich weltweit im Straßenverkehr - dies entspricht einem Verkehrstoten alle sechs Sekunden [95]. Insgesamt sollen durch diesen Aktionsplan 5 Millionen Menschenleben gerettet und 50 Millionen ernsthafte Verletzungen verhindert werden. Die meisten Opfer weltweit treten in Ländern auf, in denen überlastete Infrastruktur, wenig Fahrzeugsicherheit und Verkehrserziehung vorherrschen. Doch auch im technologisch weiterentwickelten Europa, in dem diese Bedingungen wesentlich fortschrittlicher sind, gibt es immer noch großes Potential zur weiteren Reduzierung der Verkehrstoten. Aus diesem Grund hat auch die Europäische Union (EU) eine Aktion mit dem Ziel gestartet, die Anzahl der tödlich Verunglückten von 2010 bis zum Jahr 2020 zu halbieren. Abbildung 1.1 zeigt die Anzahl der Verkehrstoten in der EU seit 2001 und gibt die Zielvorgabe wieder. Deutlich wird, dass das Bestreben der Halbierung durch die positiven Entwicklungen der letzten Jahre realistisch, aber auch durchaus ambitioniert ist. In den ersten drei Jahren der Aktion konnte das Ziel kein einziges Mal eingehalten werden.

Zur Aufschlüsselung, auf welches EU-Land welcher Anteil der tödlich Verunglückten fällt, zeigt Abbildung 1.2 die Verkehrstoten pro EU-Mitgliedsstaat auf eine Million Einwohner genormt. Hier wird deutlich, dass Deutschland im Vergleich zu anderen EU-Mitgliedsstaaten bereits auf einem guten Weg ist. Noch wirkungsvoller wird Deutschlands Status, wenn man in Betracht zieht, dass auf Deutschland 16% aller EU-Einwohner, 18% aller PKW der

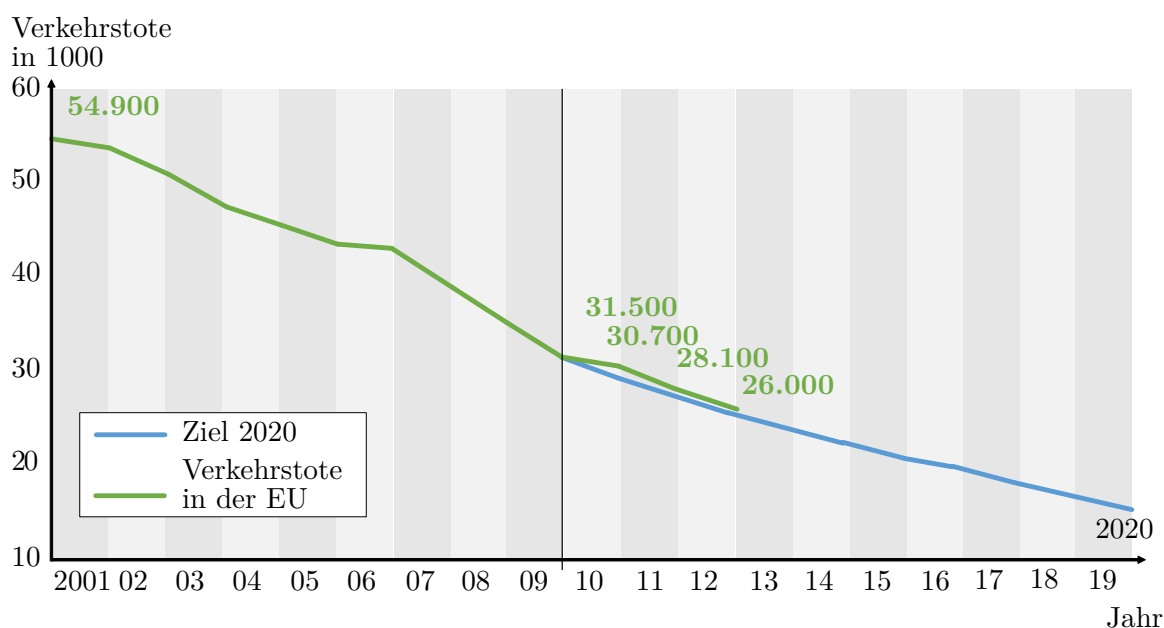


Abbildung 1.1. – Verkehrstote in der EU seit 2001 und Zielsetzung der Reduzierung der tödlich Verunglückten nach [31]

EU und 19% aller EU-weit gefahrenen Personenkilometer fallen, aber nur 13% der EU-Verkehrstoten (Jahr 2011, [29]). Woher könnte dieser Sicherheitsvorsprung kommen? Ein interessanter Aspekt, der zur Beantwortung der Frage beitragen kann, ist, dass jedes vierte Neufahrzeug der EU (26%) in Deutschland zugelassen wird. Insgesamt sind in Deutschland ein Drittel (33,1%) der 43,4 Millionen Fahrzeuge, die zum 1. Januar 2013 gemeldet waren, unter vier Jahre und 62,8% unter 10 Jahre alt ([62], [63]). Dies lässt darauf schließen, dass die meisten Fahrzeuge in Deutschland mit Fahrzeugsicherheitssystemen neuer Generationen ausgestattet sind. Gleichbleibend gering sind auch die in der Hauptuntersuchung als verkehrsuntüchtig eingestuften Fahrzeuge mit 0,0006% Anteil (im Jahr 2012 18.542 von 26.963.897), was auf eine gute qualitative Verfassung der Gesamtflotte schließen lässt [61],

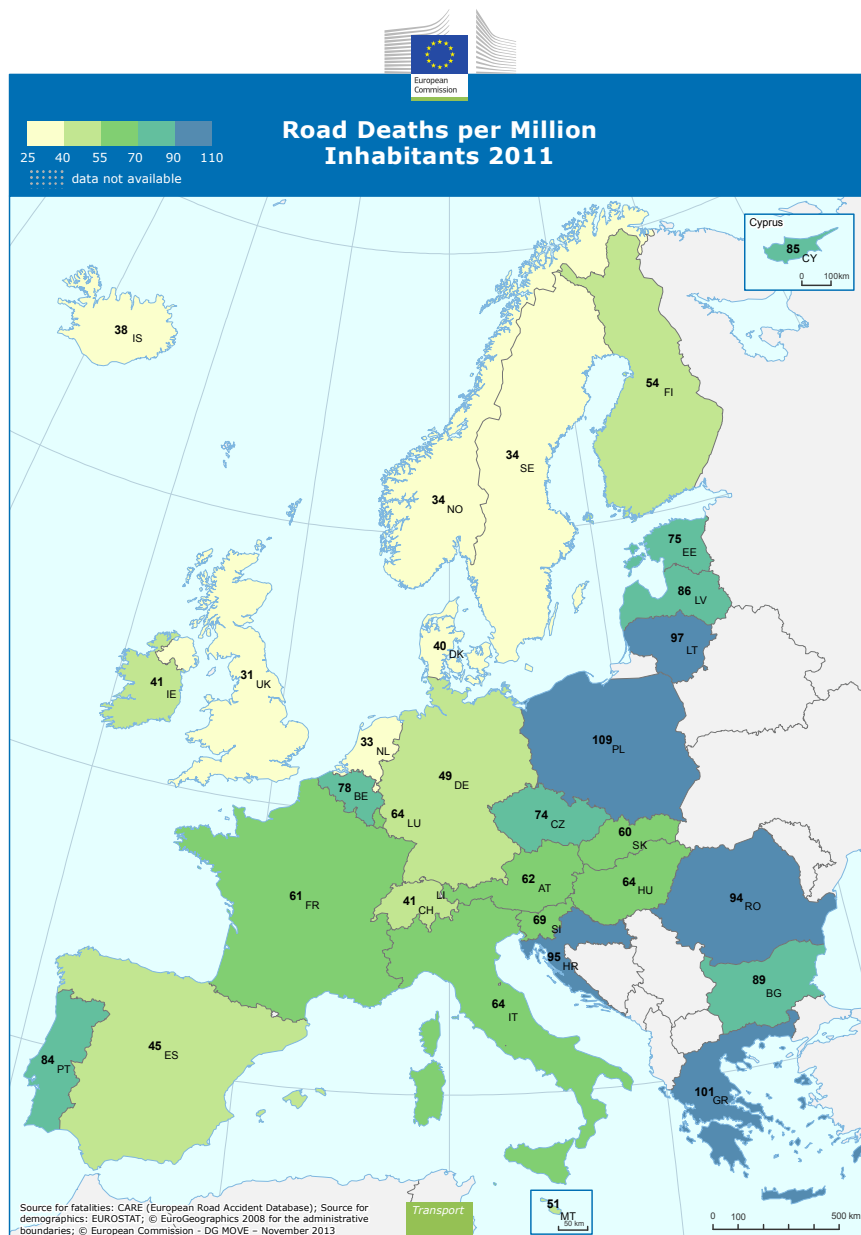


Abbildung 1.2. – Verkehrstote in Europa pro eine Million Einwohner im Jahr 2011 [30]

[25]. Dies führt zusammen mit gut ausgebauter Infrastruktur und Verkehrserziehung zu dem recht positiven Gesamtresümee für Deutschland. Doch da selbst in Deutschland jedes Jahr noch über tausend Menschen im Straßenverkehr tödlich verunglücken und Hunderttausende bei Unfällen verletzt werden (im Jahr 2013 insgesamt 377.397 Personenschäden, davon 3.338 Getötete, 64.045 Schwerverletzte und 310.014 Leichtverletzte [123]), wird deutlich: die Entwicklung der Fahrzeugsicherheitssysteme ist noch nicht am Ziel. Dies führt unweigerlich neben einem gesteigerten Sicherheitspotential auch zu erhöhter Komplexität der Systeme. Durch diese Verkomplizierung entstehen Fragen, die die Gewährleistung der richtigen Funktionsweise der Sicherheitssysteme, betreffen. Die Notwendigkeit neuartiger, effektiver Testmethodiken und -methoden ist eine wichtige aktuelle Forschungsfrage zur Absicherung von Sicherheitssystemen, um nicht tolerierbare Risiken, die von Systemen wie z. B. Airbags im Fehlverhalten ausgehen können.

1.2. Zielsetzung und Aufbau der Arbeit

Zur Verifizierung eines Fahrzeugsicherheitssystems im Systemtest muss mithilfe von Black-Box-Tests die Funktionalität sichergestellt werden. Probleme stellen hierbei die Vernetzung zwischen Steuergeräten und Funktionen dar, die schier unendliche Anzahl an möglichen Fehlerkombinationen und die Herausforderungen der nicht tolerierbaren Fehlfunktion in einem System der funktionalen Sicherheit.

Die Zielsetzung der vorliegenden Arbeit liegt darin, neuartige Testmethoden für den Systemtest integraler Sicherheitssysteme zu diskutieren und erproben, um Testfälle flexibler zu erstellen, genauer zu analysieren und zu optimieren. Durch die Anwendung simulativer, manöverbasierter Dateneinspeisung soll die Lücke im Systemtest zwischen Modul- und Integrationstest und realen Testfahrten geschlossen werden. Außerdem soll eine Testmetrik erarbeitet werden, die die Projekt- und Testfortschrittsbewertung erleichtert und Vorhersagen und Bewertungen zulässt. Somit sollen Impulse im Bereich des Systemtests integraler Fahrzeugsicherheitssysteme gegeben und neuartige Methodiken zum effizienteren und effektiveren Test erprobt werden.

Der Aufbau der Arbeit beginnt zunächst mit der Einleitung zum Thema in Kapitel 1, das einen Abriss über das aktuelle Verkehrs- und Unfallgeschehen sowie der Notwendigkeit der Arbeit gibt.

In Kapitel 2 werden anschließend die Grundlagen und Rahmenbedingungen des Themengebiets dargestellt. Hier wird ein Überblick über die integrale Fahrzeugsicherheit, den Systemtestprozess in der Automobilbranche und die Fahrzeugsicherheitsnorm ISO 26262 gegeben. Außerdem wird die Notwendigkeit der Arbeit anhand der aktuellen Entwicklungen erörtert.

Der Kern der Arbeit, die Methodik für den Systemtest integraler Fahrzeugsicherheitssysteme, ist in Abbildung 1.3 dargestellt. Analog gibt die Abbildung den Aufbau der Arbeit

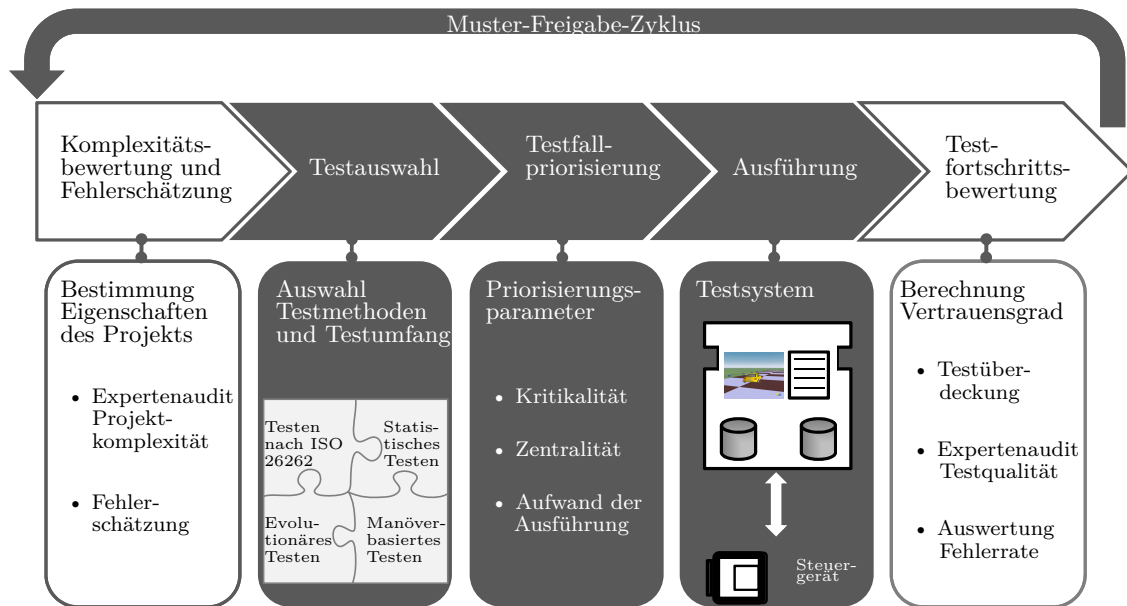


Abbildung 1.3. – Aufbau der Methodik und analog Aufbau der Arbeit. Außen hell dargestellt ist die rahmende Testmetrik, die die Testaktivitäten quantitativ und qualitativ bewertet. Mittig dunkel dargestellt sind die Schritte der Testaktivitäten.

wieder.

Kapitel 3 erarbeitet einen neuartigen Ansatz für die Komplexitätsbewertung und Fehler-ratenschätzung von Projekten in der Fahrzeugsicherheit. Hier wird anhand von Experten-audits eine objektive Aussage über die Komplexität eines Projekts getroffen und daraus abgeleitet Fehler, die im Laufe des Systemtestprozesses auftreten, abgeschätzt. Resultierend aus den Ergebnissen der Komplexitätsbewertung werden Testmethoden und -umfang ausgewählt. Hierfür wird in Kapitel 4 ein neuer Ansatz für die Auswahl von Testmethoden dargestellt, der die Methode manöverbasiertes Testen in den Mittelpunkt stellt und die Kombination der Methode mit statistischer Analyse von Testfällen und Optimierung durch Evolutionäre Algorithmen beinhaltet. Vor der Ausführung der Testfälle ist eine Testfallpriorisierung sinnvoll. Kapitel 4.5 zeigt einen solchen Priorisierungsvorgang exemplarisch anhand der Parameter Kritikalität, Zentralität und Aufwand der Ausführung. Zur Ausführung der Testfälle und der verschiedenen Methoden wird ein auf die Gegebenheiten speziell zugeschnittenes Testsystem benötigt, das in Kapitel 5 dargestellt wird. Abschließend zeigt Kapitel 6 eine Methode zur Testfortschrittsbewertung, die auf der Berechnung eines sogenannten Vertrauensgrades beruht. Durch die Einflussfaktoren Testüberdeckung, Expertenaudit zur Testqualität, Fehlerdichte und der Auswertung der

Fehlerschätzung soll ein objektiver Überblick über den Testfortschritt gegeben werden. Kapitel 7 fasst die Arbeit zusammen und gibt einen Ausblick über offene Punkte und zukünftige Forschungsmöglichkeiten.

2. Grundlagen und Stand der Technik

Dieses Kapitel gibt die aktuellen Rahmenbedingungen hinsichtlich der Arbeit wieder. Betrachtet wird konkret die Entwicklung der integralen Fahrzeugsicherheit, der aktuelle Aufbau von Sicherheitssteuergeräten inklusive des Trends der Vernetzung. Anschließend folgt eine kurze Betrachtung des für die Automobilbranche so wichtigen und rahmengebenden Standards ISO 26262. Der in der Arbeit im Fokus stehende Systemtest im Entwicklungs- und Testprozess soll durch Darstellung der kontemporären Vorgehensmodelle abgegrenzt werden und die Testbasis, die durch Anforderungen bzw. Spezifikationen gegeben ist, dargestellt werden. Außerdem soll der Systemtest hinsichtlich Teststrategie, -methoden und -systemen von anderen Testphasen abgegrenzt werden. Ausgehend aus diesen Grundlagen soll die Notwendigkeit der Arbeit begründet werden.

2.1. Integrale Fahrzeugsicherheit

Die integrale Fahrzeugsicherheit versteht sich als Vernetzung von Fahrzeugsicherheitssystemen der aktiven und passiven Sicherheit. Zur Verdeutlichung und Abgrenzung der geläufigen Begriffe, wird zunächst die wichtigste Terminologie definiert.

Die Definition von aktiver Sicherheit erfolgt nach [73] wie folgt:

Definition 2.1 (Aktive Sicherheit)

Unter aktiver Sicherheit versteht man Systeme, die

- unfallvermeidende Maßnahmen treffen,
- aktiv in das Fahrgeschehen eingreifen,
- zur Fahrsicherheit, Bedienungssicherheit, Wahrnehmungssicherheit und Konditionssicherheit beitragen.

Die passive Sicherheit wird definiert durch (nach [73]):

Definition 2.2 (Passive Sicherheit)

Unter passiver Sicherheit versteht man Systeme, die

- unfallfolgenmindernde Maßnahmen treffen,
- erst nach einem detektierten Unfall eingreifen,
- zum Selbstschutz und Kontrahentenschutz beitragen.

Die Definition von Fahrerassistenzsystemen (FAS) wird nach [110] und [132] wie folgt verwendet:

Definition 2.3 (Fahrerassistenzsystem)

Fahrerassistenzsysteme oder ADAS (Advanced Driver Assistance Systems) sind komplexe Systeme, die den Fahrer bei der Fahrzeugführung unterstützen, indem sie

- bei Routinesituationen die Tätigkeiten überwacht und so Unfälle durch Übermüdung oder Ablenkung vermeidet,
- Informationen zur Verfügung stellen, die über den Wahrnehmungsbereich des Fahrers hinausgehen,
- bei komplexen Situationen wichtige Informationen zur Verfügung stellen und bei Fehlverhalten warnt,
- vor kritischen Situationen durch Signale warnen,
- in kritischen Situationen so in das Fahrgeschehen eingreifen, dass ein Unfall verhindert oder die Unfallfolgen abgemildert werden.

Den Einsatz der unterschiedlichen Sicherheits- und Assistenzsysteme zeigt Abbildung 2.1. Durch integrale Fahrzeugsicherheit, d. h. die Vernetzung der Fahrzeugsicherheits- und Assistenzsysteme können die Insassen und andere Verkehrsteilnehmer über alle Fahrphasen hinweg unterstützt und geschützt werden [94]. Während einer normalen Fahrt wird der Fahrer durch Fahrerassistenzfunktionen und Aktive Sicherheitssysteme entlastet. Im Falle einer Gefahrensituation werden Warnungen ausgegeben und die Auslösung autonomer Sicherheitsfunktionen vorbereitet. Durch aktive Eingriffe in der Pre-Crash-Phase können Unfälle verhindert oder zumindest die Unfallfolgen abgemildert werden. Die Informationen aus der Pre-Crash-Phase werden zur Unfallschwereschätzung herangezogen, so dass eine optimierte Auslösung der passiven Rückhaltesysteme während der In-Crash-Phase möglich ist. In der Post-Crash-Phase können Verkehrsteilnehmer gewarnt und ein Notruf abgesetzt werden (nach [28]). Hierbei wird deutlich, dass die Grenzen zwischen Fahrerassistenz, akti-

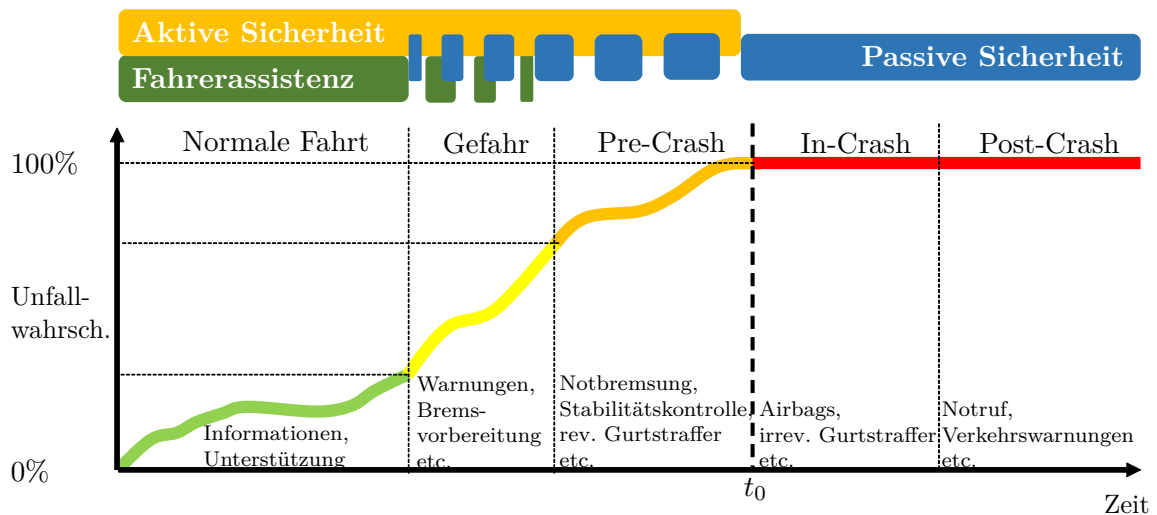


Abbildung 2.1. – Ablauf eines Unfalls und zugehörige Sicherheitssysteme (nach [28])

ver und passiver Fahrzeugsicherheit langsam verschwimmen und nicht genau abgegrenzt werden können.

Von diesen immer weiter zerfließenden Grenzen rührt auch der Ansatz der integralen Fahrzeugsicherheit her: durch intelligente Kombination und Integration der aktiven und passiven Sicherheitssysteme soll eine Erhöhung des absoluten Schutzniveaus erreicht werden [111]. Abbildung 2.2 zeigt die zeitliche Entwicklung der Integration inklusive beispielhafter Anführung von entsprechenden Systemen. Deutlich wird hier, dass die Integration heute noch nicht vollständig abgeschlossen ist und noch einige Jahre andauern wird. Das allumfassende Ziel der integralen Fahrzeugsicherheit ist das verletzungs- und unfallvermeidende Fahrzeug, das durch intensiven Einsatz von Assistenz den Fahrer entlastet und langfristig als Autopilot sogar ersetzt. Dies resultiert in einem sehr hohen Sicherheitspotential, führt aber auch einen extremen Anstieg der Systemkomplexität und damit des Entwicklungs- und Testaufwandes mit sich.

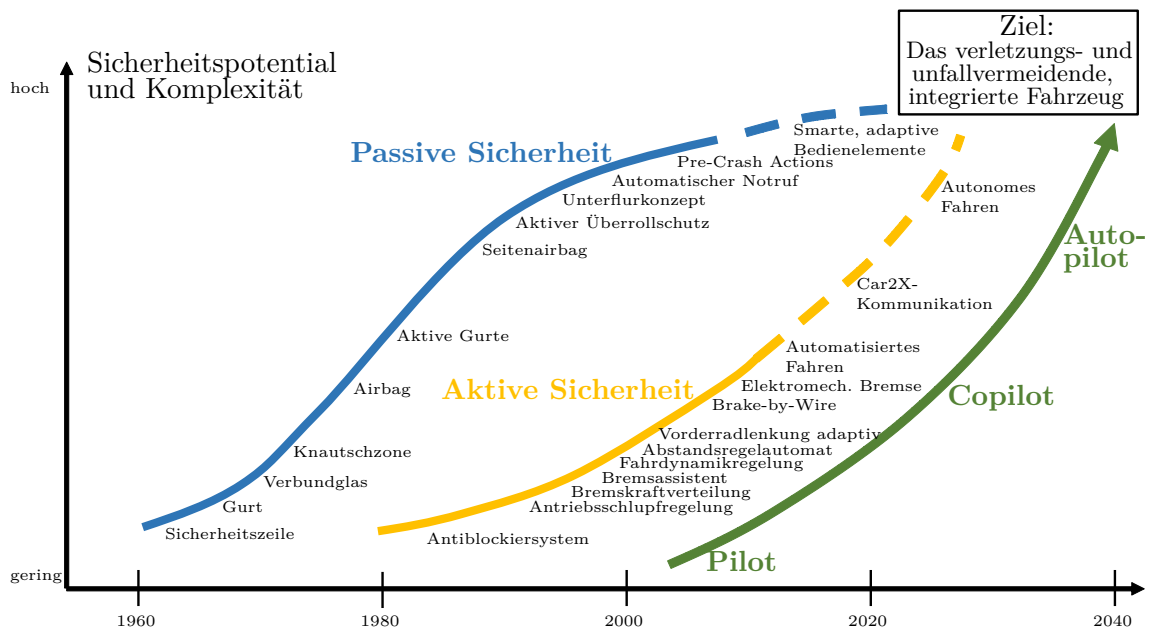


Abbildung 2.2. – Entwicklung der Zunahme der Assistenz hin zu einem vollständig integrierten Sicherheitssystem (nach [74])

2.2. Steuergeräte

Die Ausprägung der Recheneinheiten im Fahrzeug, genannt Steuergeräte, kann aufgrund der dynamischen Weiterentwicklung in diesem Bereich nur als kurzzeitig gültige Momentaufnahme dargestellt werden. Zunächst soll hierfür der Begriff des Steuergeräts allgemein definiert werden (nach [112]):

Definition 2.4 (Steuergerät)

Ein Steuergerät, engl. Electronic Control Unit (ECU), ist die physische Umsetzung eines eingebetteten Systems. Es ist die Kontrolleinheit eines mechatronischen Systems, das Sensoren und Aktoren als Peripherie besitzen kann.

Der schematische Aufbau eines Steuergeräts ist in Abbildung 2.3 dargestellt. Der algorithmenverarbeitende Mikrocontroller bekommt durch angeschlossene Sensoren und Bussysteme Sensorinformationen und Daten von anderen Steuergeräten. Typische Bussysteme für Sensoren im Fahrzeugsicherheitsbereich sind die digitalen Protokolle SPI (Serial Peripheral Interface) und PSI5 (Peripheral Sensor Interface 5). Für die steuergeräteübergreifende Vernetzung im Automobil werden typischerweise die Bussysteme CAN (Controller Area Network), LIN (Local Interconnect Network), FlexRay und MOST (Media Oriented Systems Transport) verwendet.

Zu unterscheiden sind die logische und technische Systemarchitektur bei der Vernetzung von Steuergeräten im Fahrzeug (siehe Abbildung 2.4). Logische Kommunikationsbeziehungen, d. h. der Austausch von Informationen, wird durch die Notwendigkeit und Verwendung der Daten von anderen Steuergeräten bzw. Netzknoten begründet. Hierbei können zum

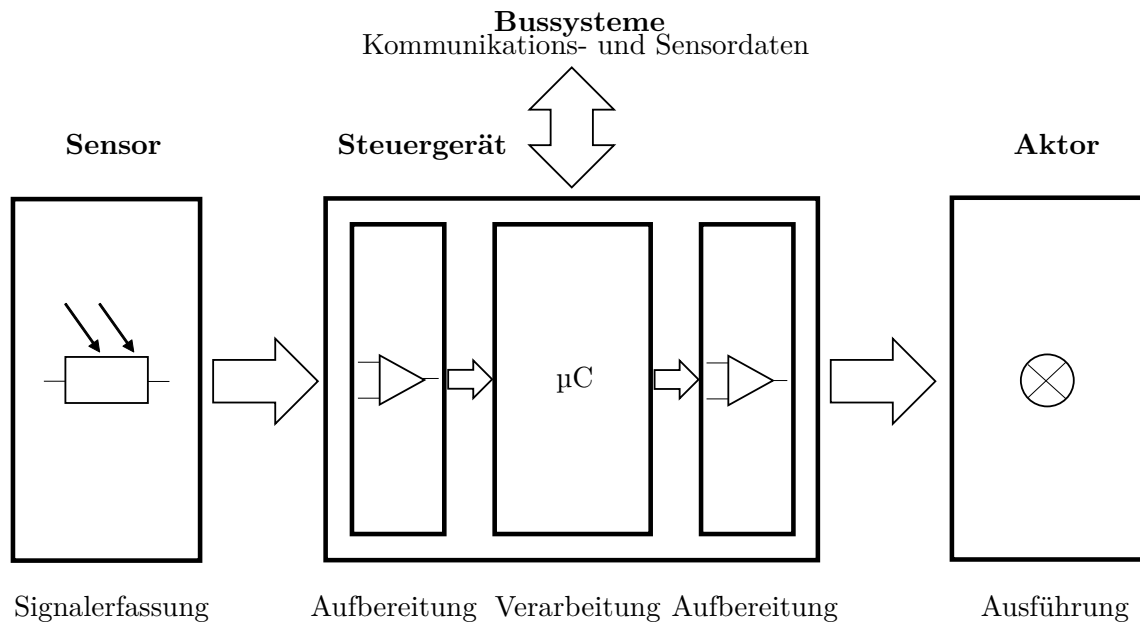


Abbildung 2.3. – Der schematische Aufbau eines Steuergeräts enthält neben dem Steuergerät selbst, das Signalaufbereitungsinstanzen und einen Mikrocontroller beinhaltet, Sensoren zur Signalerfassung, Aktoren zur Signalausführung und Bussysteme zum Datenaustausch mit anderen Steuergeräten im Fahrzeug (nach [83] und [109]).

einen bidirektionale, also in beide Richtungen austauschende Verbindungen vorhanden sein und zum anderen unidirektionale, informierende Verbindungen existieren. Möglich und häufig vorkommend ist auch die Abwesenheit von Informationsaustausch zwischen zwei Netzknoten, weil kein Datenfluss notwendig ist. So ergibt sich eine Verflechtung, die die logische Systemarchitektur widerspiegelt.

Die technische, physische Systemarchitektur kann ganz anders realisiert sein, als es die logische Systemarchitektur ist. Punkt-zu-Punkt-Verbindungen für den Austausch von Informationen sind aufgrund stark reduzierter Protokollkomplexität die schnellste Lösung, allerdings kommt es aufgrund der verflochtenen logischen Architektur schnell zu einer sehr hohen Anzahl an Verbindungen. Dies führt zu Problemen in den Aspekten Kosten, Zuverlässigkeit, Gewicht und Wartung [114]. Dies lässt sich durch den Einsatz von Bussystemen lösen, führt aber zu einer Loslösung von der logischen Systemarchitektur bei verteilten und vernetzten Systemen. Die Kommunikation zwischen Netzknoten findet so über ein gemeinsames Medium oder gar über Gateways über verschiedene Bussysteme hinweg statt. Dies kann zu Zeitproblemen führen, da weitere Wege und Konkurrenzsituationen auf dem Bus entstehen können. Die Herausforderungen für die technische Systemarchitektur bestehen also in der Lösung von Konfliktsituationen beim Buszugriff und den Zeitanforderungen.

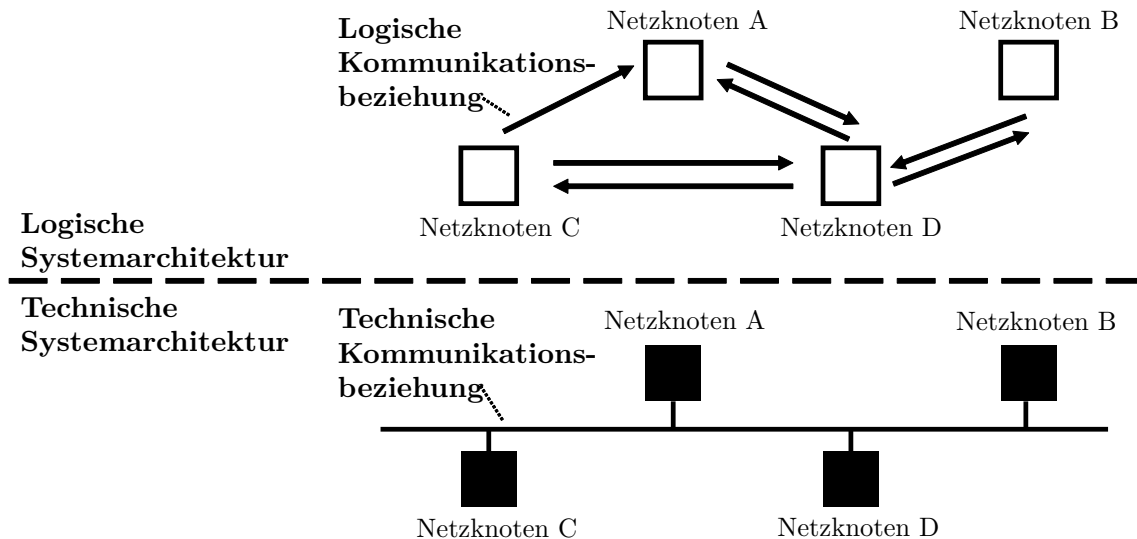


Abbildung 2.4. – Darstellung des Unterschieds zwischen logischer und technischer/physischer Systemarchitektur (nach [114]).

2.2.1. Sensoren

Sensoren werden möglichst für einen breiten Anwenderkreis und in gestaffelten Messbereichen entwickelt, ohne dass der Hersteller oft die Anwendung kennt. Im Gegensatz dazu werden Sensoren im Automobilbereich sehr speziell für eine Anwendung spezifiziert und optimiert, was wegen der erhöhten Ansprüche häufig zu längeren Entwicklungszeiten führt, die parallel zur Systementwicklung läuft [93]. Die Aufgabe der Sensoren im Fahrzeug ist die Aufnahme und Messung der Umgebung und Weitergabe an das Steuergerät. Dies ist vergleichbar mit den Sinnesorganen von Mensch und Tier und der Weiterleitung der Informationen an das Gehirn. Mit ihrer Hilfe kann die ECU die Umwelt wahrnehmen und anhand der Signale Entscheidungen treffen. Grundsätzlich ist ein Sensor nach [91] wie folgt definiert:

Definition 2.5 (Sensor)

Ein Sensor (lat. *sentire* - fühlen), ist ein Fühler oder Aufnehmer, der eine physikalische oder chemische (meist nichtelektrische) Größe in eine elektrische Größe umsetzt. Anforderungen an Sensoren im Automobilbereich sind niedrige Herstellungskosten, hohe Zuverlässigkeit, harte Betriebsbedingungen, kleine Bauweise und hohe Genauigkeit.

Abbildung 2.5 zeigt den aktuellen Einsatz von Sensoren im Bereich Fahrzeugsicherheit. Um noch einmal auf die Analogie der Sinnesorgane zurückzukommen, wird hierbei deutlich, dass ein Fahrzeug durch verschiedene Sensoren wie z. B. Beschleunigungs-, Druck- oder Drehratensensoren fühlen, durch vorausschauende Sensoren wie Radar, Kamera oder Lidar sehen, z. B. durch Körperschallsensoren hören und zukünftig durch Funkmodule im Car2X-Bereich auch mit anderen Verkehrsteilnehmern und Infrastruktur kommunizieren kann. All diese Informationen werden über Bussysteme an das Steuergerät gesendet, verarbeitet und als Entscheidungsgrundlage zur Handlungsausführung verwendet. Hierbei sind aus

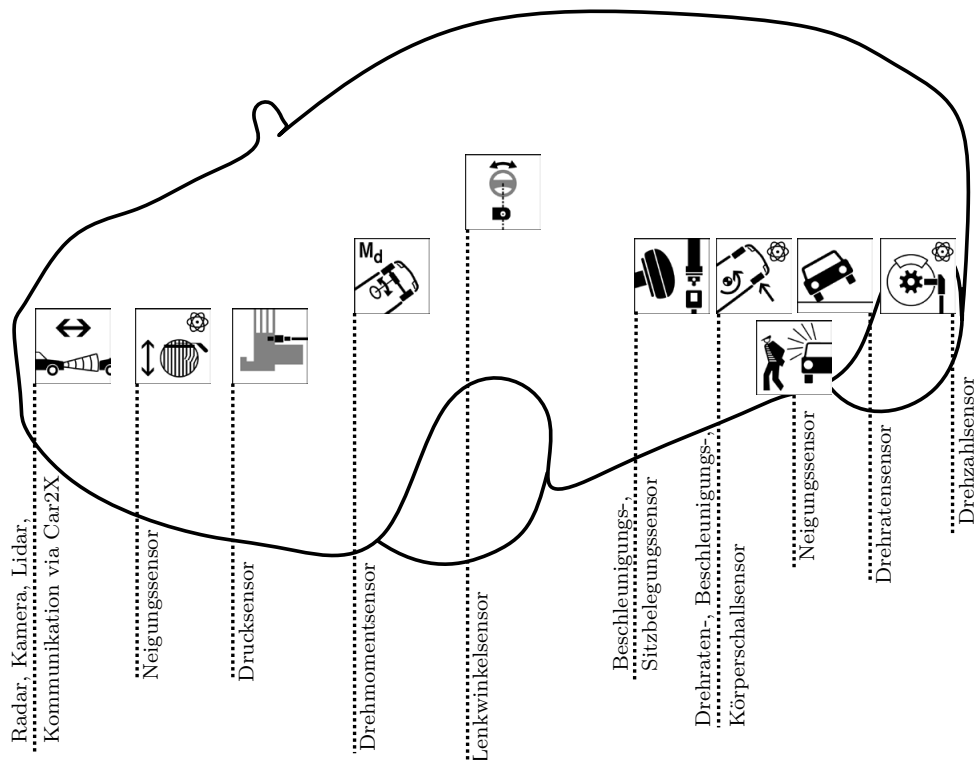


Abbildung 2.5. – Sensoren im Bereich Fahrzeugsicherheit (nach [91])

Gründen der Redundanzvermeidung nicht alle Sensoren direkt an das Sicherheitssteuergerät angeschlossen, sondern werden von anderen Steuergeräten wie z. B. der Bremse aufbereitet und via CAN- oder FlexRay-Bus gesendet. Häufig sind Sensoren bereits selbst sehr aufwändige Systeme: sie beinhalten häufig Signalaufbereitung, Analog-Digital-Wandlung, Selbstkalibrierung und kleine Mikrocomputer. Dies birgt die Vorteile, dass im Steuergerät weniger Rechenleistung benötigt wird, einheitliche, busfähige Schnittstellen für alle Sensoren bereitgestellt werden können, über den Datenbus eine Mehrfachnutzung des Sensors erfolgen kann, kleine Messeffekte erfasst werden können und ein einfacher Abgleich des Sensors ermöglicht wird [90].

2.2.2. Aktoren

Aktoren stellen die Handlungsmedien des Steuergerätes dar. Definiert sind Aktoren nach [91] wie folgt:

Definition 2.6 (Aktor oder Aktuator)

Ein Aktor oder Aktuator (lat. *agere* - handeln), setzt Signale geringer Leistung, die Stellinformationen beinhalten, in leistungsbefahene Signale um, die in einer zur Prozessbeeinflussung notwendigen Energieform vorliegen. Sie bilden die Schnittstelle zwischen elektrischer Signal- bzw. Informationsverarbeitung und dem Prozess, also der Mechanik.

Die Ausprägungen der Aktoren in der Fahrzeugsicherheit sind häufig Elektromotoren (z. B. beim Aufstellen der Motorhaube im Fußgängerschutz oder reversiblen Gurtstraffern) und pyrotechnischer Art (z. B. beim Airbag und dem nichtreversiblen Gurtstraffer). Die Kommunikation des Steuergeräts mit den Aktoren erfolgt dementsprechend hauptsächlich über elektrische, analoge Schnittstellen und nur am Rande über Bussysteme.

2.2.3. Aktuelle Entwicklung

Die Komplexitätsentwicklung des Steuergeräts, das für Fahrzeugsicherheitsfunktionen verwendet wird, ist enorm: während vor ein paar Jahren noch vom Airbagsteuergerät gesprochen wurde und das Steuergerät eine relativ kleine Komponente mit Hauptfunktionalität der Airbagauslösung im Fahrzeug darstellte, so spricht man heute vom Sicherheitssteuergerät, das ganze Sensorverbunde, mannigfaltige Pre-Crash- und Post-Crash-Funktionen und stark erweiterte Airbagfunktionalität enthält. Der aktuelle Trend zeigt eine weitere Integration und Vergrößerung: man spricht von einem Sicherheitsdomänensteuergerät, das erweiterte Rechenleistung beinhalten wird und viele vorausschauende, komplexe ADAS- und vielseitig kommunizierende Car2X-Funktionen integrieren wird. Aufgrund der Vernetzung der Steuergeräte im Fahrzeug kann die Begrenzung der Funktionalität auf ein System aufgehoben werden. Ohne die Aufnahme weiterer Steuergeräte können neue Funktionalitäten eingeführt werden, indem die Kooperation verschiedener Systeme genutzt wird [109]. Ein Beispiel hierfür ist die vorausschauende Notbremsfunktion Fahrzeug: Wenn z. B. durch Radar eine Vollbremsung des vorausfahrenden Fahrzeugs erkannt wird, meldet das Sicherheitssteuergerät eine Bremsanforderung an das Bremsensteuergerät. Weiterführend kann anschließend noch über Car2X-Funktionalität die Bremsinformation an dahinter fahrende Fahrzeuge gemeldet werden.

Zur Absicherung dieser vernetzten Funktionen ist effektive und eng abgestimmte Testarbeit notwendig. Es wird versucht, die Bewältigung dieser Aufgaben durch Standardisierung, Automatisierung und domänenübergreifende Zusammenarbeit erreichen. Der wichtigste Standard der Fahrzeugsicherheit, der einen Beitrag hierzu leistet, ist ISO 26262 [56].

2.3. ISO 26262

ISO 26262 Road vehicles - Functional safety wurde 2011 als Richtlinie zur Entwicklung von funktionalen Sicherheitsfunktionen für die Automobilindustrie veröffentlicht. Sie stellt die Spezialisierung der generischen Sicherheitsnorm IEC 61508 [51] dar und ist eng mit dem bestehenden Kernprozess ISO/TS 16949 [56] und dem SPICE Framework respektive ISO 15504 [53] verwandt [37]. Die Anwendung des Standards IEC 61508 für die Automobilindustrie hätte diverse Schwierigkeiten mit sich gebracht, die durch die Interpretation dieser Norm und Modifizierung in ISO 26262 gelöst werden konnten [136]. Insgesamt umfasst

ISO 26262 1269 für die Entwicklung und den Test eines sicherheitsrelevanten Systems zu berücksichtigende Vorgaben [43].

2.3.1. Überblick

ISO 26262 verändert aktuell die Fahrzeugentwicklung derart, wie man es sich noch vor zehn Jahren nicht hätte vorstellen können. Entstanden ist die Norm aufgrund der Notwendigkeit der Absicherung der Produktsicherheit. Sie beinhaltet einen detaillierten Sicherheitslebenszyklus (safety life cycle), der die früheren Validierungs- und Verifizierungstechnik der Automobilbranche nun als industrieweite Methodik zusammenträgt [17]. Da sie nicht als Richtlinie geschrieben wurde, macht sie keine genauen Angaben zur Umsetzung, sondern stellt nur Anforderungen an Aktivitäten und Methoden, die bei den jeweiligen Sicherheitsaktivitäten berücksichtigt werden sollen [97]. Dennoch stellt ISO 26262 einen wichtigen Rahmen für die Entwicklung und den Test in der Funktionalen Sicherheit dar. Die wichtigsten Ziele von ISO 26262 sind [125]:

- Definition einer guten Sicherheitskultur
- „Top Down“-Ableitung von Sicherheitsfunktionen
- Definition einer quantitativen Produktzuverlässigkeit oder einer Methodik zu Ausfall pro Zeit (Failure in Time, FIT)

Nachfolgend sollen kurz die wichtigsten Begriffe und Ansätze von ISO 26262 dargestellt werden.

ISO 26262 findet ihren Anwendungsbereich in der Funktionalen Sicherheit. Die allgemeine Definition von Funktionaler Sicherheit ist nach [57]:

Definition 2.7 (Funktionale Sicherheit)

Funktionale Sicherheit ist die Abwesenheit von nicht tolerierbarem Risiko aufgrund von Gefahren, die durch Fehlfunktion eines Elektrik- und Elektronik-Systems (E/E-System), das Software und Hardware beinhaltet, entstehen können.

Das Ziel der Funktionalen Sicherheit ist die Abwesenheit von unakzeptablen Risiken, um Schäden zu vermeiden [97]. Der Zusammenhang zwischen Risiko und Schaden wird durch ISO 26262 wie folgt definiert (nach [57] und [97]):

Definition 2.8 (Zusammenhang zwischen Risiko und Schaden)

Ein Risiko ($R = \text{Risk}$) wird durch Kombination der Häufigkeit ($f = \text{frequency}$) und dem Ausmaß des Schadens ($S = \text{Severity}$) beschrieben:

$$R = f \cdot S$$

Die Auftretenshäufigkeit f wiederum wird durch die Dauer und Häufigkeit bestimmt, in der sich Personen in einer Situation befinden ($E = \text{Exposure}$) sowie durch die Nicht-Abwendbarkeit von Schäden durch rechtzeitige Reaktion von Personen ($C = \text{Controllability}$):

$$f = E \cdot C$$

Risiko ist also wie folgt definiert:

$$R = E \cdot C \cdot S$$

Aus dieser Definition ergibt sich ein wichtiger, zentraler Punkt aus ISO 26262: die Spezifikation von Sicherheitsanforderungsstufen für sicherheitsrelevante Systeme, sogenannte Automotive Safety Integrity Level (ASIL). Definiert sind ASIL wie folgt [97]:

Definition 2.9 (ASIL - Automotive Safety Integrity Level)

ASIL bezeichnet eine von insgesamt fünf Stufen (QM (Quality Management), A, B, C, D, wobei QM die niedrigste und D die höchste Stufe ist), die auf Sicherheitsmaßnahmen und notwendige Anforderungen für das Fahrzeugsystem oder einzelne Elemente verweist. ASIL dient der Vermeidung von nicht tolerierbaren Risiken.

Ermittelt werden die ASIL mithilfe eines qualitativen Verfahrens. Hierbei werden analog zu Definition 2.8 die Kriterien Severity, Exposure Time und Controllability verwendet, um das Risiko der Funktion zu bestimmen [37]. Das Vorgehen wird in Tabelle 2.1 dargestellt. Zur Bewertung der Einstufung einer Funktion werden häufig Felddaten, Erfahrungswerte, Simulationen und Audits herangezogen. Die Eingruppierung ist häufig auch Grund zur Diskussion und nicht immer eindeutig. Grundsätzlich existiert das Bestreben möglichst niedrige ASIL zu vergeben, um Zeit- und Kostenaufwand einzusparen. Jedoch muss sich der Komponentenhersteller im Falle eines Fehlverhaltens seines Systems für die Einstufung rechtfertigen und seine Absicherung darlegen. Außerdem sind OEMs (Original Equipment Manufacturers, also Automobilhersteller) immer auf größtmögliche Absicherung bestrebt, da Rückrufaktionen und Fehlfunktionen zu Imageschädigung und Rechtsstreitigkeiten führen. Aus diesem Grund wirken die OEMs einer zu niedrigen Einstufung durch Zulieferern entgegen. Beispiele für die einzelnen Einstufungen können in ISO 26262 oder einschlägiger Literatur nachgelesen werden. Zur Reduktion des Aufwandes gibt es allerdings noch die sogenannte ASIL-Decomposition, die es bei redundanten Strukturen erlaubt ein hohes ASIL durch Anwendung verschiedener Maßnahmen zu reduzieren. So kann z. B. ein ASIL D aufgeteilt werden in ein ASIL C(D) und ein ASIL A(D) oder in ein ASIL B(D) und ein ASIL B(D).

Ausgehend von der ASIL-Bewertung einer Funktion werden Testaktivitäten und -methoden empfohlen. Die Empfehlung beruht auf drei Stufen: *highly recommended*, *recommended*

S: Severity (Schwere/Ausmaß des Schadens)					
	S0	S1	S2	S3	
Beschreibung	Keine Verletzungen	Leichte und mäßige Verletzungen	Schwere und lebensbedrohliche Verletzungen	Lebensbedrohliche, fatale Verletzungen	
E: Exposure Time (Eintrittswahrscheinlichkeit bzw. Häufigkeit der Situation)					
	E0	E1	E2	E3	E4
Beschreibung	Unvorstellbar	Sehr unwahrscheinlich	Unwahrscheinlich	Mittlere Wahrscheinlichkeit	Hohe Wahrscheinlichkeit
Anteil an der Betriebszeit	-	< 0,1%	< 1%	1 - 10 %	> 10 %
Ab-schätzung	-	seltener als 1x pro Jahr	wenige Male pro Jahr	monatlich o. häufiger	bei fast jeder Fahrt
C: Controllability (Beherrschbarkeit durch den Durchschnittsfahrer)					
	C0	C1	C2	C3	
Beschreibung	Im Allgemeinen beherrschbar	Einfach beherrschbar	Normalerweise beherrschbar	Schwierig o. nicht beherrschbar	
Ab-schätzung	< 0,1% der durchschnittl. Fahrer können den Schaden nicht vermeiden	< 1% der durchschnittl. Fahrer können den Schaden nicht vermeiden	< 10% der durchschnittl. Fahrer können den Schaden nicht vermeiden	Durchschnittl. Fahrer können den Schaden nicht vermeiden	
Abgeleiteter ASIL aus den Teilbewertungen					
Summe S + E + C					
Summe	<= 6	7	8	9	10
ASIL	QM	ASIL A	ASIL B	ASIL C	ASIL D

Tabelle 2.1. – Risikobewertung nach ISO 26262 (nach [37] und [57])

und *no recommendation for or against*. Wie zu erwarten ist, ist z. B. für eine ASIL-D-Funktion fast jede Aktivität oder Testmethode *highly recommended*, während für eine ASIL-A-Funktion weniger empfohlen wird. Somit korreliert der über den kompletten Entwicklungs- und Testzyklus hinweg zu betreibende Aufwand sehr stark mit der ASIL-Einstufung.

2.3.2. Testmethoden nach ISO 26262

Die Testmethoden, die nach ISO 26262 für den Systemtest in Teil „4-8 Item integration and testing“ (siehe auch Abbildung 2.10) empfohlen werden, sind in Tabelle 2.2 dargestellt. Die Testziele im Systemtest zielen in erster Linie auf Verifizierung, Effektivität und Robustheit ab. Eine kurze Erläuterung der einzelnen Testmethoden wird nachfolgend wiedergegeben (nach [57]).

Testziele auf Systemebene	Methoden	ASIL			
		A	B	C	D
Korrekte Implementierung von funktionalen und technischen Sicherheitsanforderungen	Requirements-based test	++	++	++	++
	Back-to-back test	o	+	+	++
	Fault injection test	+	+	++	++
Korrekte funktionale Performance, Genauigkeit und Timing von Sicherheitsmechanismen	Back-to-back test	o	+	+	++
	Performance test	o	+	+	++
Konsistente und korrekte Implementierung von externen und internen Interfaces	Tests of external interfaces	+	++	++	++
	Tests of internal interfaces	+	++	++	++
	Interface consistency check	o	+	++	++
	Test of interaction/communication	++	++	++	++
Effektivität der Diagnoseüberdeckung eines Sicherheitsmechanismusses	Fault injection test	+	+	++	++
	Error guessing test	+	+	++	++
	Test derived from field experience	o	+	++	++
Robustheitslevel	Resource usage test	o	+	++	++
	Stress test	o	+	++	++
	Tests for interference and robustness and under certain environmental conditions	++	++	++	++

Legende: o no recommendation for or against, + recommended, ++ highly recommended

Tabelle 2.2. – Projektbewertung für fünf abgeschlossene und drei laufende Beispielprojekte

- **Requirements-based test:** Test gegen funktionale und nichtfunktionale Anforderungen
- **Back-to-back test:** Vergleich der Reaktion des Testobjekts gegen die Reaktion eines Simulationsmodells mit denselben Stimuli, um Unterschiede zwischen dem Verhalten des Modells und seiner Implementierung zu erhalten
- **Tests of external interfaces:** Tests von analogen und digitalen Inputs und Outputs, Grenzfällen und Äquivalenzklassen zum vollständigen Test der spezifizierten Interfaces, Kompatibilität, Timings und anderen Auslegungen des Testobjekts
- **Tests of internal interfaces:** Tests von analogen und digitalen Inputs und Outputs, Grenzfällen und Äquivalenzklassen zum vollständigen Test der spezifizierten Schnittstellen, Kompatibilität, Timings und anderen Auslegungen des Testobjekts;

Test der internen Schnittstellen eines Steuergeräts durch statische Tests für die Kompatibilität zur Software und Hardware und durch dynamische Tests hinsichtlich der Serial Peripheral Interface- (SPI-) oder Integrated Circuit- (IC-) Kommunikation

- **Interface consistency check:** Tests von analogen und digitalen Inputs und Outputs, Grenzfällen und Äquivalenzklassen zum vollständigen Test der spezifizierten Interfaces, Kompatibilität, Timings und anderen Auslegungen des Testobjekts
- **Test of interaction/communication:** Tests der Kommunikation zwischen Systemelementen, dem System Under Test (SUT) und anderen Fahrzeugsystemen zur Laufzeit gegen funktionale und nichtfunktionale Anforderungen
- **Fault injection test:** Methode, um Fehler während der Laufzeit einzuspeisen und die Testüberdeckung der Sicherheitsanforderungen zu erhöhen, da im normalen Operationszustand die Sicherheitsmechanismen nicht angesprochen werden; Einspeisung entweder über spezielle Testinterfaces in der Software oder speziell präparierte Hardware möglich
- **Error guessing test:** Verwendung von Expertenwissen und gesammelten Lessons-Learned-Daten um Fehler im Testobjekt vorauszusehen
- **Test derived from field experience:** Verwendet Erfahrungen und Daten aus dem Feldgebrauch; Falsches Fahrzeugverhalten oder neu entdeckte Situationen werden analysiert und Tests werden spezifiziert um das Fahrzeug hinsichtlich der neuen Bedingungen zu testen
- **Resource usage test:** Ressourcengebrauch im dynamischen Umfeld (z. B. Prototypen) um z. B. Stromverbrauch und Buslast zu testen
- **Performance test:** Verifikation der Leistungsfähigkeit (z. B. Aktorgeschwindigkeit oder -kraft, Reaktionszeiten des Gesamtsystems) der Sicherheitsmechanismen des Systems
- **Stress test:** Verifikation der korrekten Lauffähigkeit des Systems unter hohen Lasten oder hohen Anforderungen durch das Umfeld; z. B. können Tests unter hohen Lasten, extremem Userinput, Anfragen von anderen Systemen, extremen Temperaturen, Luftfeuchtigkeit oder mechanischen Schocks durchgeführt werden
- **Tests for interference resistance and robustness and under certain environmental conditions:** Spezieller Fall des Stresstests; enthält Tests zur elektromagnetischen Verträglichkeit und elektrostatischer Entladung

Grundsätzlich werden diese Testmethoden inzwischen im Systemtest von Fahrzeugsicherheitssystemen eingesetzt. Als problematisch kann sich jedoch der große Spielraum erweisen, den ISO 26262 zulässt. ISO 26262 beinhaltet diverse Freiheitsgrade, die berücksichtigt werden müssen (nach [84]):

- **Freiheit der Interpretation:**

Die genaue Bedeutung, Interpretation und Umsetzung von vorgegebenen Testmethoden ist nicht fix vorgegeben, d. h. nur eine Orientierung und nicht bindend. Die Ausprägungen sind selbst festlegbar. Die Testmethoden werden nicht detailliert beschrieben, sondern es besteht die Möglichkeit der Interpretation und der Argumentation der Auslegung im Kontext des eigenen Umfelds.

- **Freiheit der Selektion:**

Die Auswahl der Testmethoden ist frei wählbar (siehe auch Abbildung 2.2, die aufzeigt, dass lediglich *recommendations* also Empfehlungen gegeben werden). Das bedeutet, es werden nur Vorschläge hinsichtlich der adäquaten Selektion gegeben und keine festen Vorgaben, was eine Rechtfertigung zur Ausklammerung von Testmethoden jederzeit beinhaltet.

- **Freiheit der Kombination:**

Die Kombination der Testmethoden ist nicht festgelegt. Dadurch, dass manche Kompositionen zum Beispiel aufgrund der Ähnlichkeit der Methoden nicht sinnvoll sind, kann diese Freiheit leicht missbraucht werden, um Testaufwand einzusparen und die Testqualität zu mindern.

Daraus ergibt sich die große Verantwortung, die ein Testingenieur bei der Testplanung trägt: Ohne sorgfältige Testmethoden- und Testumfangwahl, kann trotz der rahmengebenden Norm nicht ausreichend getestet werden. Diese Arbeit behandelt in erster Linie den Requirements-based test, d. h. die Ableitung von Testfällen aus Anforderungen. Genaue Angaben zur Bedeutung, Ausprägung und Umsetzung dieser Tests werden in ISO 26262 nicht behandelt. Tiefergehende Analysen, der Grad der Testabdeckung und spezielle Testfälle zum gründlichen Test von Grenzfällen werden ebenfalls nicht explizit gefordert.

2.4. Der Systemtest im Entwicklungs- und Testprozess der Automobilbranche

Der Begriff des Systems wird häufig in der Regelungstechnik verwendet und ist nach [133] wie folgt definiert:

Definition 2.10 (System)

Ein dynamisches System ist eine Funktionseinheit, die Signale (z. B. Energie, Material, Information, Kapital) verarbeitet und überträgt. Systemeingangsgrößen werden als Ursache und Systemausgangsgrößen als zeitliche Auswirkung zueinander in Relation gebracht.

Der Systemtest beschäftigt sich also mit Signalen und deren Werten und zeitlicher Abfolge. Die Testaktivitäten im Entwicklungsprozess eines Systems sind in verschiedene Testschritte untergliedert. Grundsätzlich unterscheidet man zwischen den Verifikationsmaßnahmen

Modul-/ Komponententest, Integrationstest und Systemtest und den Akzeptanz-/ Abnahmetests als Validierungsmaßnahmen (siehe auch Kapitel 2.4.1.1). Zur Abgrenzung des Systemtests von den anderen Phasen, sollen die genannten Testschritte zunächst definiert werden (nach [114]):

Definition 2.11 (Modul-/ Komponententest)

Ein Modul bzw. eine Komponente wird gegen die Spezifikation des Moduls bzw. der Komponente getestet.

Definition 2.12 (Integrationstest)

Das System wird gegen die Spezifikation der technischen Funktionsanforderung und -architektur getestet

Definition 2.13 (Systemtest)

Das System wird gegen die Spezifikation der logischen Systemanforderung und -architektur getestet

Definition 2.14 (Akzeptanz-/ Abnahmetest)

Das System wird gegen die Benutzeranforderungen bzw. das Lastenheft getestet

Durch die Testmaßnahmen können Fehler nachgewiesen werden - nicht jedoch die Abwesenheit von Fehlern, wie Dijkstra bereits 1972 in seinem berühmten Zitat „Program testing can be a very effective way to show the presence of bugs, but is hopelessly inadequate for showing their absence“ festgestellt hat [21]. Durch die Identifikation von Fehlern wird jedoch kontinuierlich zur Produktqualität beigetragen, weshalb Testen möglichst früh auf allen Systemebenen durchgeführt werden sollte [114].

Der Systemtest ist also eingebettet zwischen dem Integrationstest eines Moduls und dem Akzeptanztest. D. h. es handelt sich um die Betrachtung und den Test des zusammengefügt Modulglomerats. Der Systemtest zielt dementsprechend darauf ab, das Gesamtsystem abzusichern, bevor es anschließend zum ersten Mal in ein Fahrzeug integriert wird. Die Hauptziele des Systemtests sind nach [118] wie folgt definiert:

Definition 2.15 (Ziele des Systemtests)

Die Hauptziele des Systemtests sind

- Fehler entdecken, bevor wirtschaftlicher oder körperlicher Schaden entsteht,
- Vertrauen in das System aufbauen,
- Nachweis der Funktionsfähigkeit des Systems für bestimmte Randbedingungen (Anforderungen),
- Erreichen einer vorgegebenen Testüberdeckung.

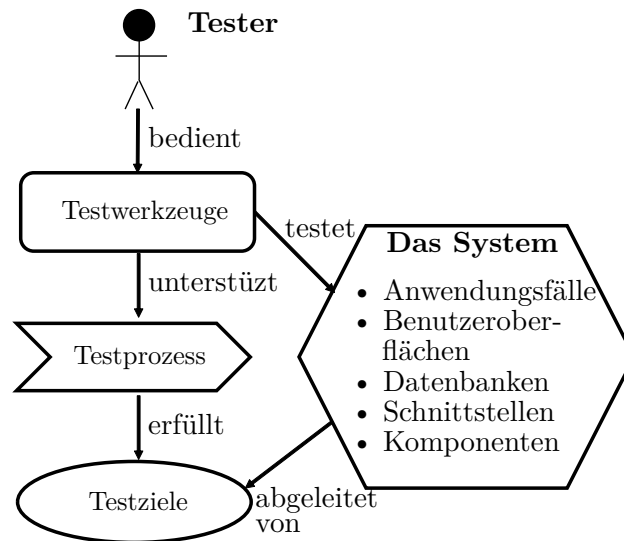


Abbildung 2.6. – Die Systemtestkette (nach [118]).

Der Ablauf der Systemtestkette ist in Abbildung 2.6 dargestellt. Ein Tester bedient ein Testwerkzeug, das das zu testende System integriert und den Testprozess unterstützen muss, welcher wiederum die Testziele erfüllen soll. Die Testziele werden aus dem System bzw. den Anforderungen an das System abgeleitet.

Weiterführend soll nun der Systemtest anhand verschiedener Vorgehensmodelle, der Darstellung von Anforderungen und der beinhalteten Teststrategien und -methoden genauer beschrieben und abgegrenzt werden.

2.4.1. Vorgehensmodelle

Die Beschreibung des Entwicklungs- und Testprozesses und somit des Ablaufs der Entstehung eines Produkts und der Einbettung der verschiedenen Entstehungsphasen erfolgt durch Vorgehensmodelle. Diese Modelle stellen ideale Abläufe dar. Nach [48] sind Produktlebenszyklen bzw. Vorgehensmodelle wie folgt definiert:

Definition 2.16 (Vorgehensmodell)

Ein Vorgehensmodell beschreibt die Zeitspanne, die mit der Konzipierung eines Produkts beginnt und mit der Fertigstellung oder Außerbetriebnahme endet. Der Lebenszyklus enthält typischerweise die Phasen Konzept, Anforderung, Entwurf, Implementierung, Test, Installierung, Produktionsstart, Betrieb, Wartung und eventuell Außerbetriebnahme. Diese Phasen können überlappen oder iterativ durchgeführt werden.

In der Realität herrscht oft auch aus historischen oder praktischen Gründen eine Mischform aus verschiedenen Vorgehensmodellen vor. Häufig werden die Modelle auch in Teilschritten oder iterativ durchlaufen, um wichtige Änderungen zu realisieren oder Fehler zu beheben [114]. Andererseits ist der chronologische und strenge Ablauf eines Modells oft nicht realisierbar, da nicht zu jedem Zeitpunkt bereits alle Informationen wie Anforderungen oder

technische Gegebenheiten zur Verfügung stehen [92]. Die drei gängigsten Vorgehensmodelle der Automobilbranche sollen nun exemplarisch vorgestellt werden.

2.4.1.1. Das V-Modell

Das bekannteste und am weitesten verbreitetste Vorgehensmodell in der Automobilbranche ist das sogenannte V-Modell. Das V-Modell wird in verschiedenen Quellen wie der ISO 9001 [54] (Qualitätsmanagementsysteme) und ISO 12207 [55] (Systems and software engineering — Software life cycle processes) sowie CMMI [14] (Capability Maturity Model Integration) und SPICE bzw. ASPICE [53] (ISO/IEC 15504, Software Process Improvement and Capability Determination) beschrieben. Es existiert auch eine Erweiterung des V-Modells zum V-Modell XT, dass das V nur für die Entwicklung einzelner Produkte beschreibt [97]. Es herrschen sporadisch auch andere V-Modell-Derivate vor, die statt einer V- eine W-Struktur beschreiben oder eine mehrdimensionale Anordnung des Vs verwenden [6], [83]. Das in Abbildung 2.7 aufgezeigte V-Modell hat sich zu einem Quasi-Standard im Automobilbereich entwickelt, da die in der Automobilbranche intensiv genutzten Tests integriert und ausführlich und übersichtlich dargestellt werden. Die Unterteilung des V-Modells erfolgt grundsätzlich in einen linken Entwurfs- und Entwicklungsast und einen rechten Verifizierungs- und Validierungsast. Die Begriffe Verifizierung und Validierung sind wie folgt definiert:

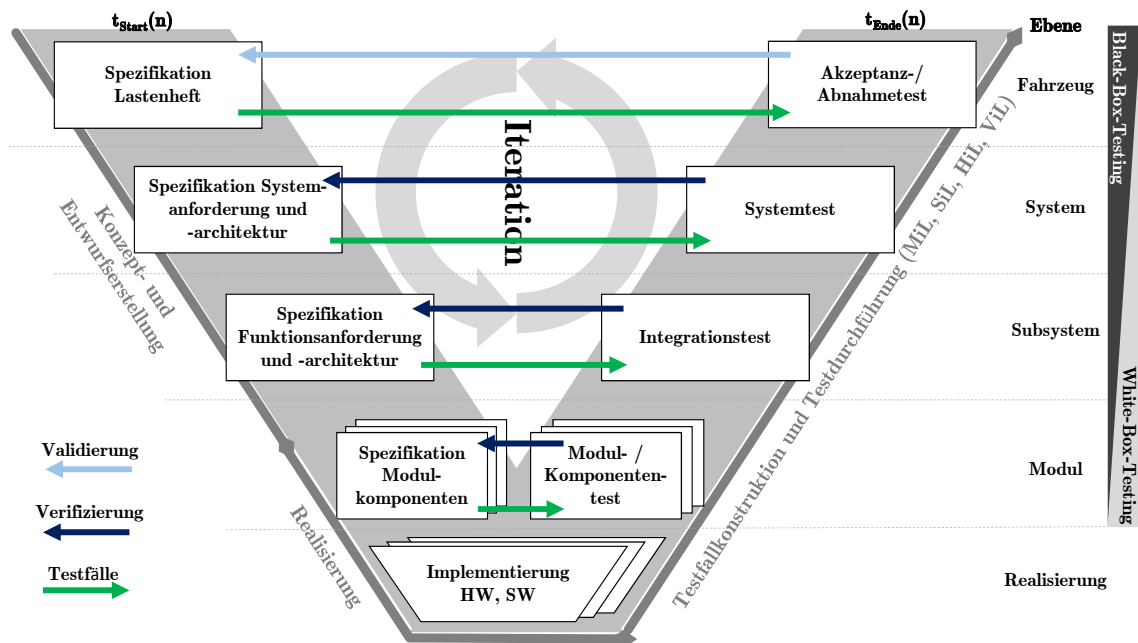


Abbildung 2.7. – Das V-Modell (nach [8], [10], [83] und [122])

Definition 2.17 (Verifizierung)

Verifizierung (lat. *verificare* - überprüfen) ist eine Bestätigung, dass durch eine Untersuchung und einen bereitgestellten Nachweis festgelegte Forderungen (z. B. Anforderungen einer Spezifikation) erfüllt wurden [83], [52]. Es ist der formale Nachweis der Korrektheit des Programms [48]. Wurde das Produkt „richtig“ entwickelt [10]?

Definition 2.18 (Validierung)

Validierung (lat. *validus* - wirksam, stark) ist eine Aktivität zur Überprüfung der Ergebnisse einer Entwurfsphase bezüglich ihrer Vorgaben durch initiale Anforderungen [83], [48]. Wurde das „richtige“ Produkt entwickelt [10]?

Das zu entwickelnde System wird in einzeln lösbare Teilproblematiken zerlegt und Funktionen und Komponenten zugeordnet. Für den Test jeder Entwurfsphase werden Testfälle abgeleitet, sodass die entsprechenden Anforderungen überprüft werden können. Die Testausführung wird in umgekehrter Reihenfolge zur Entwicklung durchgeführt. Die Verifikation beginnt somit mit der Überprüfung der Softwareimplementierung, die aus Anwendungssoftware, Betriebssystem, hardwareabhängigen Treibern, Diagnosesoftware und Schnittstellenmodulen besteht. Häufig werden viele Teile davon als fertige Softwarekomponenten realisiert, wie zum Beispiel durch das Baukastensystem von AUTOSAR [4]. Die Implementierung einer Komponente wird durch die Hardwareintegration getestet. Die Funktionalität der Komponente ist dann nach der Integration der Software gegeben, die durch die Hardware-Software-Integration überprüft wird. Dieser Teil ist im Normalfall noch Aufgabe des Zulieferers und der eigentliche Komponententest kann - je nachdem ob noch weitere Systeme für den Test notwendig sind - dann entweder vom OEM oder noch vom Zulieferer durchgeführt werden. Anschließend erfolgt der Systemtest, der alle Komponenten integriert und im Verbund die Systemfunktionalität verifiziert. Dieser Teil erfolgt meist beim OEM, kann aber bei einem Gesamtsystemlieferanten auch beim Zulieferer erfolgen. Im Automobilbereich wird das V-Modell abschließend noch um die Fahrzeugintegration erweitert, bei der die Funktionalität des Systems im Gesamtfahrzeug überprüft wird. Die Validierung des Systems erfolgt durch den OEM durch Testfahrten, Fahrzeug- und Dauerlauf tests [83]. Die Trennung des Entwicklungs- und Testprozesses erfolgt zum einen aufgrund von Zeitmangel auf Seiten der Entwickler systematisch und gründlich zu testen und andererseits aufgrund der Existenz psychologischer Barrieren, die das Auffinden eigener Fehler erschweren [118].

2.4.1.2. Das Spiralmodell

Auch andere Vorgehensmodelle sind neben dem V-Modell möglich. Ein in der Automobilbranche traditionsreiches Vorgehensmodell ist das sogenannte Spiralmodell (siehe Abbildung 2.8). Da in der Automobilindustrie vorwiegend Musterphasen die Entwicklungsaktivitäten bestimmen, ist ein Vorgehensmodell, das sich an diesen Phasen orientiert, ebenfalls möglich [97]. Nach [48] ist das Spiralmodell wie folgt definiert:

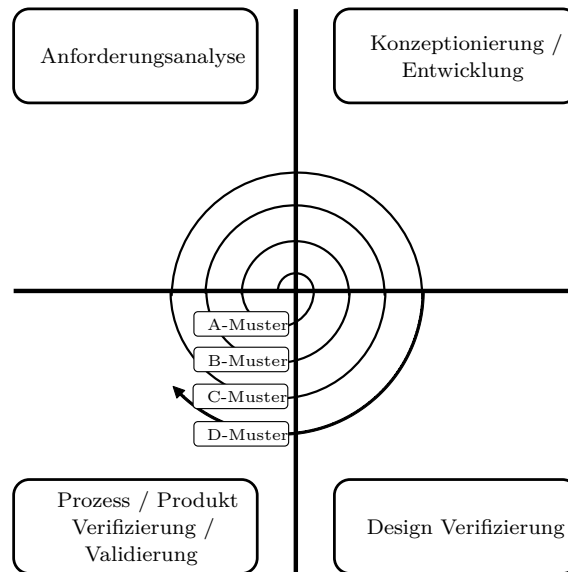


Abbildung 2.8. – Das Spiralmodell (nach [97])

Definition 2.19 (Spiralmodell)

Das Spiralmodell ist ein Modell des Entwicklungsprozesses, in dem die beinhalteten Aktivitäten - typischerweise Anforderungsanalyse, vorläufiger und detaillierter Entwurf, Implementierung, Integration und Test - iterativ bis zur Fertigstellung des Produkts durchgeführt werden.

Wie der Name Spiralmodell schon aussagt, ist der Zeitstrahl als Spirale aufgetragen, der als Meilensteine sogenannte A-, B-, C- und D-Muster enthält. Diese entstehen jeweils aus der Anforderungsanalyse, Konzeptionierung und Entwicklung, Design Verifikation und anschließender Prozess- und Produktverifizierung und -validierung. Die Musterstände sind wie folgt definiert [107], [109]:

Definition 2.20 (A-Muster)

A-Muster bieten die Möglichkeit der grundsätzlichen Funktionsdarstellung und -erprobung mit Spezifikationseinschränkungen. Die Implementierung der Grundfunktionalität ist entsprechend dem Lastenheft erfolgt und wird unter realen Ersatzbedingungen erprobt, evaluiert und optimiert, wobei die eingesetzte Hardware noch nicht mit der Zielhardware übereinstimmen muss.

Definition 2.21 (B-Muster)

B-Muster verwenden bereits die Zielhardware und die Funktionalität gewährleistet eine ausreichende Betriebssicherheit für Erprobung auf dem Prüfstand und im Fahrzeug.

Definition 2.22 (C-Muster)

C-Muster beinhalten die vollständige Software und die Hardware wird unter seriennahen Bedingungen gefertigt. Dies ermöglicht eine seriennahe Gesamterprobung.

Definition 2.23 (D-Muster)

D-Muster werden mit Serienwerkzeugen unter Serienbedingungen gefertigt. Die Einsatzfähigkeit ist vollständig gewährleistet.

Häufig herrscht in der Praxis trotz Nutzung des V-Modells immer noch die Bezeichnung der Musterstände vor. Dies ist ein Beispiel dafür, dass auch eine gemischte Verwendung von Vorgehensmodellen möglich ist.

Es existieren noch weitere generische Vorgehensmodelle wie z. B. das Wasserfallmodell nach [99], die jedoch in der Automobilbranche eher selten verwendet werden.

2.4.1.3. Das Vorgehensmodell für Sicherheitsfunktionen nach ISO 26262

Neben den vorhergehend beschriebenen allgemeinen Vorgehensmodellen existiert auch ein spezielles, genormtes Vorgehensmodell zur Gestaltung von sicherheitskritischen Fahrzeugfunktionen [134]. Basierend auf dem Vorgehen in der internationalen Norm IEC 61508 [51] definiert auch ISO 26262 [57] (siehe auch Kapitel 2.3) ein solches Vorgehensmodell für den Lebenszyklus von Sicherheitsfunktionen (siehe Abbildung 2.9). Das Vorgehensmodell beinhaltet explizit Phasen für Risikoanalyse, Konzept für die Funktionale Sicherheit, Entwicklung und auch spätere Realisierung, Produktion und Nutzung von Sicherheitsfunktionen. Parallel zur Entwicklung wird die Produktion und die Verwendung geplant [134].

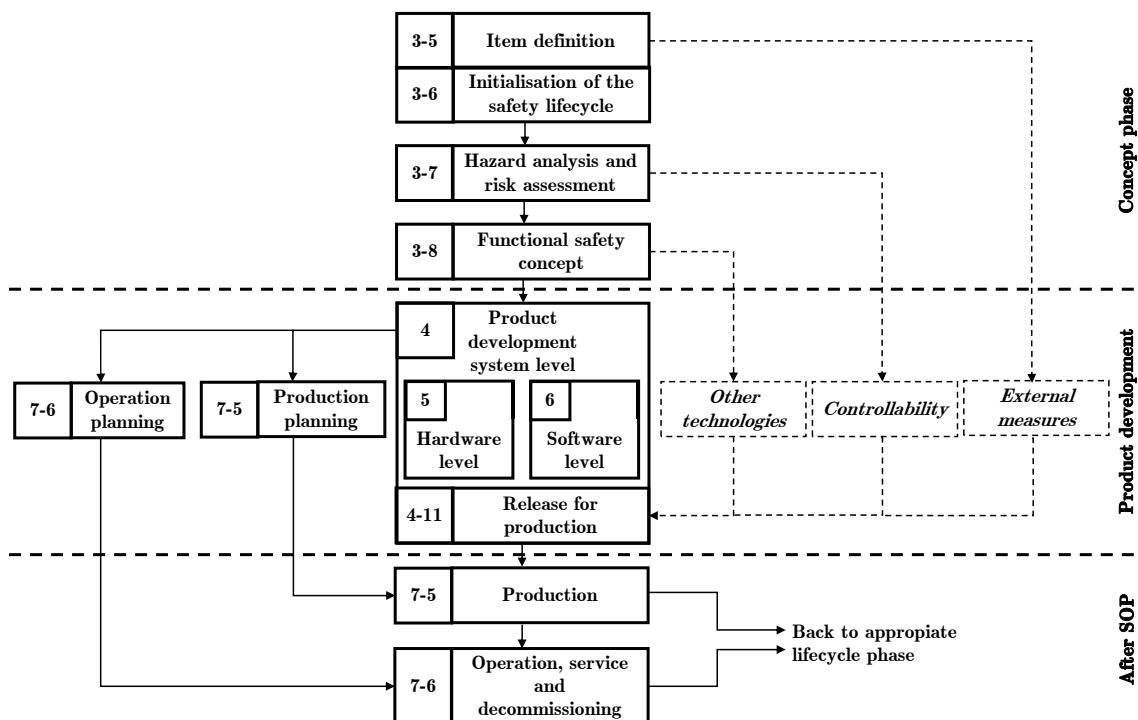


Abbildung 2.9. – Der Safety Lifecycle nach ISO 26262 [57]. Die Nummern beziehen sich auf die einzelnen Teile von ISO 26262.

Ersichtlich wird vor allem der integrierte Systemansatz von ISO 26262: Hardware- und Softwareentwicklung sind eng in die Systementwicklung eingebunden. Für ein genaueres Bild der Systemebene, stellt Abbildung 2.10 das Referenzphasenmodell für das System Level dar. Der hierbei beschriebene Ablauf ist im Kern durchaus ähnlich zum V-Modell aus

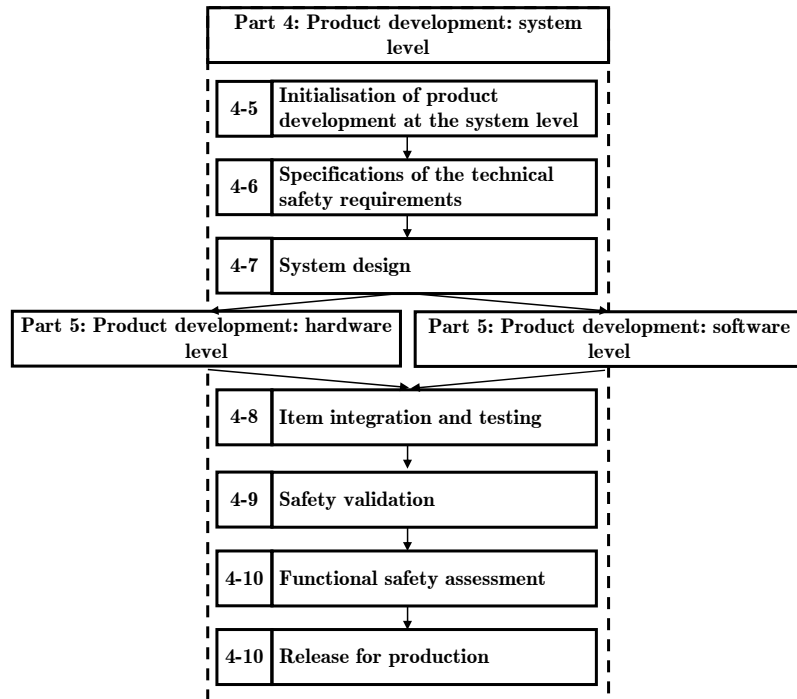


Abbildung 2.10. – Das Referenzphasenmodell für die Systemebene nach ISO 26262 [57]. Die Nummern beziehen sich auf die einzelnen Teile von ISO 26262.

Kapitel 2.4.1.1, wenn auch nicht in dem charakteristischen V dargestellt. Die Aktivitäten des V-Modells finden sich in den verschiedenen Teilen von ISO 26262 wieder und spezifische Aktivitäten und Arbeitsprodukte werden in der Norm durch Anforderungen festgelegt [137]. Der Systemtest ist in „4-8 Item integration and testing“ eingebettet. Im späteren Kapitel 2.3.2 werden die darin enthaltenen Testmethoden genauer diskutiert.

2.4.2. Anforderungen

Qualitativ hochwertige Anforderungen und Anforderungs-Management sind wichtige Bausteine im Entwicklungsprozess. Ohne Anforderungen gegen die getestet wird, können Tester nur aus Routine oder Annahme wissen, was das erwartete Ergebnis sein soll. Eine Anforderungsspezifikation stellt also die Testbasis für den Systemtest dar. Man könnte sogar sagen: ohne Anforderungen, die den Erwartungswert festlegen, ist es kein wirklicher Test [41].

Wie aus Abbildung 2.7 ersichtlich ist, werden im Systemtest Testfälle aus Systemanforderungen und -architektur zur Verifizierung abgeleitet. Allgemein werden Anforderungen, häufig auch mit dem englischen Begriff *Requirements* verwendet, nach [48] wie folgt definiert:

Definition 2.24 (Anforderungen)

Anforderungen sind

- ein Zustand oder eine Fähigkeit, die von einem Benutzer zur Problemlösung benötigt wird.
- ein Zustand oder eine Fähigkeit, die von einem System oder einer Systemkomponente erfüllt oder beherrscht werden muss, um einen Vertrag, Standard, Spezifikation oder andere formelle Dokumente zu erfüllen.
- eine dokumentierte Repräsentation eines Zustands oder einer Fähigkeit.

Anforderungen verbalisieren also gewünschte Fähigkeiten eines Systems. Sie bilden die Rahmenbedingungen und den ersten Schritt hin zu einem entwickelten System. Anforderungen können mannigfaltiger Art sein. Abbildung 2.11 zeigt einerseits, welche Benutzergruppen ein Fahrzeug verwenden und andererseits, welche Anforderungen sich daraus ergeben. Neben

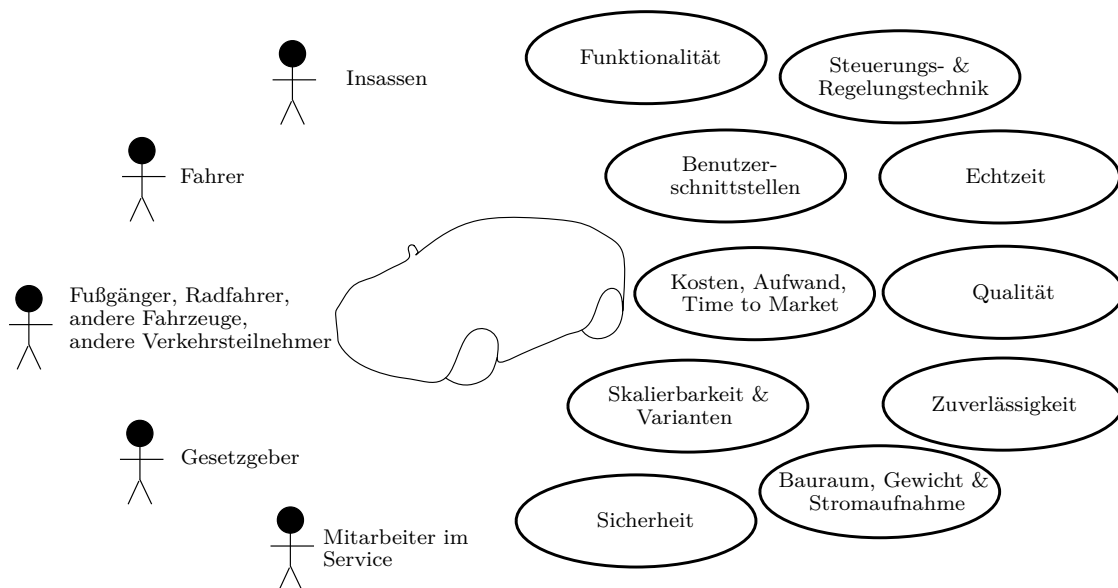


Abbildung 2.11. – Darstellung der Benutzergruppen eines Fahrzeugs und die sich ergebenden Anforderungsklassen an das System Fahrzeug (nach [114]).

Fahrer und Insassen haben auch andere Verkehrsteilnehmer Kontakt mit einem Fahrzeug. Auch der Gesetzgeber hat Interesse daran, Fahrzeugentwicklungen zu beeinflussen. Ebenfalls wichtig ist die Wartbarkeit des Fahrzeugs durch Servicemitarbeiter. Aus den Interessen dieser Benutzergruppen ergeben sich verschiedene Anforderungsarten an ein Fahrzeug: Von Funktionalität über Sicherheit bis hin zu Bauraum, Gewicht und Stromaufnahme werden unzählige Dinge festgelegt, um ein System zu beschreiben.

Es werden verschiedene Arten von Anforderungen unterschieden. Nach [47] sind sie wie folgt definiert:

Definition 2.25 (Arten von Anforderungen)

Anforderungen sind untergliederbar in

- Funktionale Anforderungen: Was soll das System/Produkt tun?
- Nichtfunktionale Anforderungen: Welche Randbedingungen und Qualitätsanforderungen muss das System erfüllen?
 - Randbedingungen: Ressourcen, Wiederverwendung, Zukauf, Technologie
 - Qualitätsanforderungen: Wie gut? Wie schnell? Wie zuverlässig?

Wichtig ist auch die Qualität der Anforderungen, d. h. es gibt Anforderungen an Anforderungen. Nach [5] und [49] erfüllen gute Anforderungen folgende Kriterien:

Definition 2.26 (Kriterien für gute Anforderungen)

Gute Anforderungen sind

- korrekt (*correct*),
- eindeutig (*unambiguous*),
- vollständig (*complete*),
- konsistent (*consistent*),
- klassifizierbar nach Wichtigkeit (*ranked for importance*),
- klassifizierbar nach Stabilität (*ranked for stability*),
- überprüfbar (*verifiable*),
- modifizierbar (*modifiable*),
- verfolgbar (*traceable*).

Ein wichtiges Gebiet, das den Umgang mit Anforderungen verwaltet, ist das sogenannte Requirements-Management oder Anforderungsmanagement. Abbildung 2.12 zeigt die einmaligen und laufenden Tätigkeiten des Requirements-Managements. Während die einmaligen Tätigkeiten vor allem definierende Maßnahmen wie Rollenfestlegung oder Werkzeugauswahl umfassen, stellen die laufenden Tätigkeiten permanente Aktionen dar, die während des Entwicklungsprozesses ablaufen. Wichtig sind hier Attribute für Anforderungen, die z. B. Prioritäten, Zustand oder Bearbeiter sein können [47]. Auch das sogenannte Change Management, also die Änderung und Veränderung von Anforderungen ist eine häufig wiederkehrende Aufgabe während des Projekts. Das Anforderungsmanagement befasst sich auch mit der Verfolgung und Umsetzung der Anforderungen (siehe Abbildung 2.13). Es ist notwendig, die Zusammenhänge zwischen den Benutzeranforderungen sowie der logischen und technischen Systemarchitektur zu verwalten, was der Verbindung aller

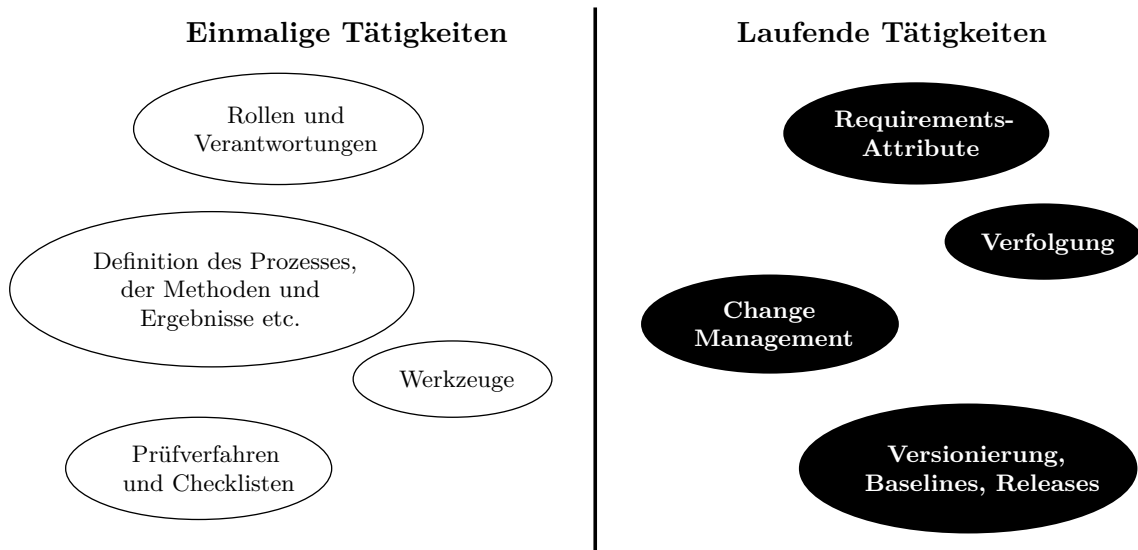


Abbildung 2.12. – Einmalige und laufende Tätigkeiten des Requirements-Managements (nach [47])

Systemkomponenten mit den entsprechenden Anforderungen entspricht [114]. Die Anforderungen werden also Funktionen zugewiesen, die wiederum aus verschiedenen Teilen der technischen Systemarchitektur bestehen. Somit entsteht ein komplexes Geflecht, das oft nur durch leistungsfähige Werkzeuge effektiv verwaltet werden kann.

Ein weiterer wichtiger Punkt in den laufenden Requirements-Management-Tätigkeiten ist

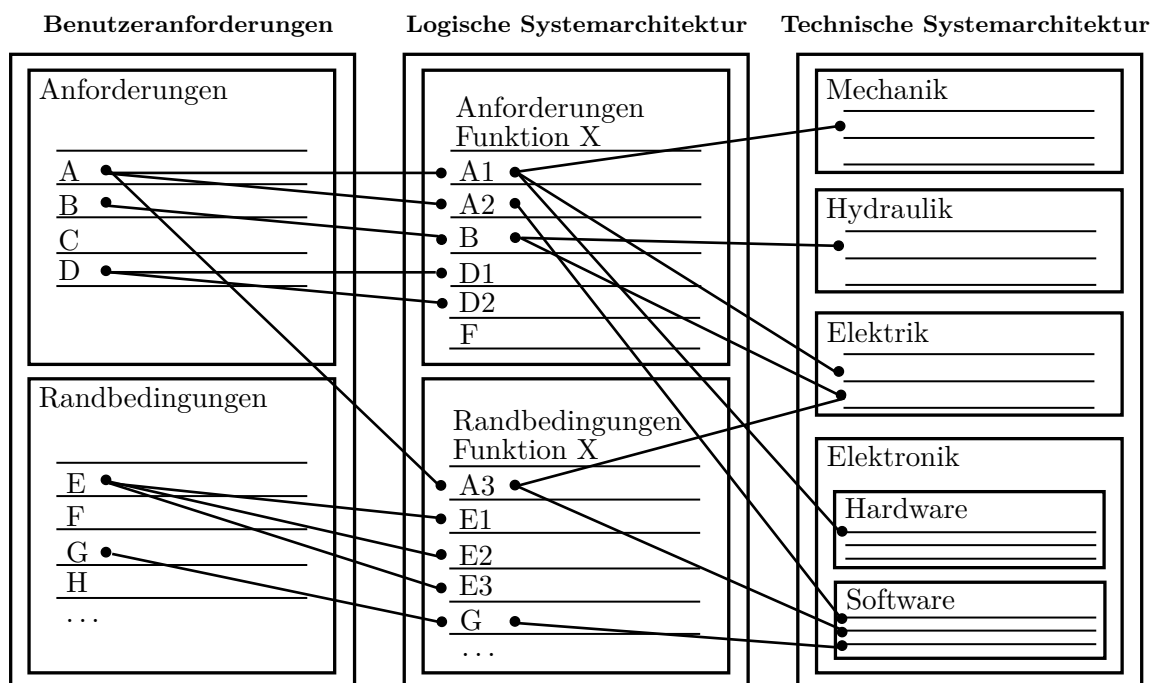


Abbildung 2.13. – Darstellung der Verfolgung von Anforderungen von Benutzeranforderungen über die logische hin zur technische Systemarchitektur (nach [114] und [124]).

die Versionierung sowie die Erstellung von Baselines und Releases. Diese sind nach [47] wie folgt definiert:

Definition 2.27 (Baseline)

Eine Baseline ist eine festgelegte Menge von Anforderungen. Diese Anforderungen werden gebündelt, eingefroren und an die Entwicklung übergeben. Man kann auch von einer Konfiguration von Anforderungen sprechen.

Definition 2.28 (Release)

Bei einer iterativen, inkrementellen Vorgehensweise spricht man bei dem nach einem völligen Durchlaufen der Kette Anforderungen, Design, Implementierung und Test entstandenen Systemversion von einem Release.

Häufig stellen Releases fertige Stände oder Meilensteine dar, die dann ausgeliefert oder veröffentlicht werden.

2.4.3. Teststrategie

Der Einsatz von Teststrategien und verschiedenen Testmethoden ist für eine zuverlässige Absicherung unerlässlich. Zur Darstellung sollen zunächst die verschiedenen, in diesem Bereich geläufigen Begrifflichkeiten definiert werden.

Der Begriff der Methodik ist nach [23] wie folgt definiert:

Definition 2.29 (Methodik)

Eine Methodik [Griechisch *methodikós* - methodisch] ist die Wissenschaft bzw. Lehre von Verfahrensweisen einer Wissenschaft, Wissenschaft von Lehr- und Unterrichtsmethoden und von festgelegten Arten des Vorgehens.

Ähnlich, aber dennoch abgegrenzt zu einer Methodik ist der Begriff Methode zu verstehen, der nach [22] wie folgt definiert ist:

Definition 2.30 (Methode)

Eine Methode [Griechisch *méthodos* - Weg oder Gang einer Untersuchung] ist ein auf einem Regelsystem aufbauendes Verfahren, das zur Erlangung von wissenschaftlichen Erkenntnissen oder praktischen Ergebnissen dient bzw. eine Art und Weise eines Vorgehens.

Abbildung 2.14 zeigt den fundamentalen Testprozess. Hierbei wird das Testen in kleine Teilabschnitte aufgeteilt, die nicht unbedingt sequentiell abgearbeitet werden, sondern sich auch durchaus überschneiden oder parallelisiert sein können [120]. Die Planung und Steuerung der Testaktivitäten wird häufig durch eine sogenannte Teststrategie festgelegt. Die Teststrategie stellt das Rahmengerüst für die kompletten Testaktivitäten dar. Bei der Festlegung der Strategie sollte also besondere Sorgfalt geboten sein. Hierbei wird deutlich, dass Testen wesentlich mehr beinhaltet als reine Testdurchführung: die Planung und

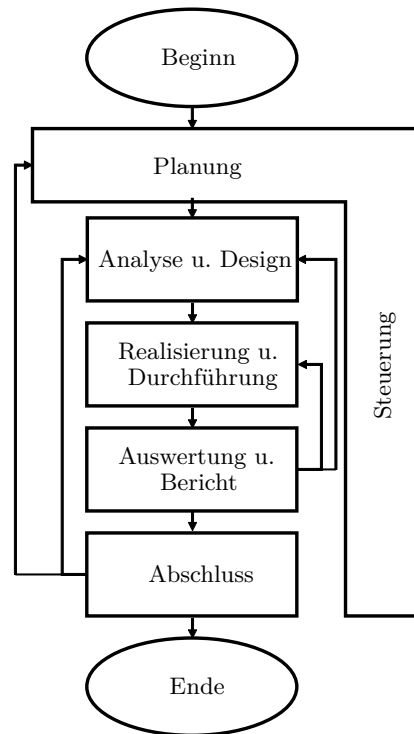


Abbildung 2.14. – Der fundamentale Testprozess [120].

Steuerung der Testaktivitäten spielt während des gesamten Testprozesses eine existenzielle Rolle. Ausgehend von der gründlichen Planung und Analyse und Design können Testfälle realisiert und durchgeführt werden. Wichtig sind auch die Auswertung und Dokumentation der Ergebnisse. Bevor ein Fehler beseitigt werden kann, muss er analysiert und lokalisiert werden [1]. Diese Aktivitäten beeinflussen dann wiederum die erneute Analyse, Design, Realisierung und Durchführung. Zum effektiven Testen ist also eine gründliche Planung und somit Teststrategie unerlässlich. Nach [72] und [83] ist der Begriff *Teststrategie* folgendermaßen definiert:

Definition 2.31 (Teststrategie)

Eine Teststrategie ist die Beschreibung der Methodik zur Erreichung der Testziele. Dies beinhaltet

- die Festlegung der für das zu testende Objekt geforderten Testabdeckungen und -stufen,
- die Testintensität pro Merkmal, Teilobjekt und Teststufe,
- die einzusetzenden Testmethoden,
- die Zuordnung der Testobjekte auf Testumgebungen.

Bei der Festlegung der Teststufen werden häufig analog zum V-Modell (siehe Abbildung 2.7) Modul-, Integrations- und Systemtest festgelegt [72]. Folgende Aspekte sollten in die Abgrenzung und Festlegung der Teststufen einfließen [72]:

- Gewählte Entwicklungsmethode
- Organisationsstruktur der Abteilungen
- Getrennte Verantwortlichkeiten
- Verschiedene Interessensvertreter
- Risikostufe
- Reifegrad von Testen und Entwicklung
- Vertragliche Vereinbarungen
- Verfügbarkeit der Infrastruktur
- Integrationsstufe
- Kauf oder Abnahme

Die Testintensität wird häufig anhand Risikoanalysen und -bewertungen wie z. B. dem ASIL (siehe Kapitel 2.3) festgelegt. Kritische Funktionen oder Teilobjekte benötigen somit eine höhere Testintensität als niedrig eingestufte [72]. Auch die Festlegung der einzusetzenden Testmethoden und -umgebungen beruht häufig auf der Risikoanalyse von Funktionen und Teilobjekten. Die geläufigsten Vertreter im Systemtest der Testmethoden und -systeme sollen nun nachfolgend kurz dargestellt werden.

2.4.4. Testdokumentation

Ein wichtiger Aspekt des Testens ist die Dokumentation der Ergebnisse. Eine gut gepflegte Dokumentation führt zu Reproduzierbarkeit und Übersichtlichkeit. Ein wichtiger Standard für die Dokumentationsaktivitäten ist *IEEE Std 829-2008: IEEE Standard for Software and System Test Documentation* [50]: Er beschreibt die Vorgänge, Inhalte und Abläufe, die für qualitativ hochwertige Dokumentation notwendig sind. Abbildung 2.15 zeigt den von dem Standard vorgeschriebenen Ablauf. Hierbei soll nach IEEE 829 [50] der Master Test Plan die Integritätslevelwahl und das Schema, die übergreifenden Testprozesse, -aktivitäten und -aufgaben sowie die ausgewählten Testlevel und -dokumente beinhalten. Der System Test Plan beinhaltet den Bereich des Testlevels, die Ressourcen und die Methoden. Das System Test Design ist detaillierter und beinhaltet Aktualisierungen für die Testmethoden. Die System Test Cases beinhalten Input und Output und die System Test Procedures dokumentieren den Testaufbau und die Ausführungsinstruktionen. Nach der Testausführung wird zum einen ein Anomaly Report erzeugt, der inkorrekte und unerwartete Ergebnisse beinhaltet und zum anderen ein System Test Report generiert, der alle Ergebnisse detailliert beinhaltet. All diese Ergebnisse werden im Master Test Report gesammelt. Auch ISO 26262 [57] beschäftigt sich in Teil 8 mit der Dokumentation. Die Absichten der Dokumentation sind:

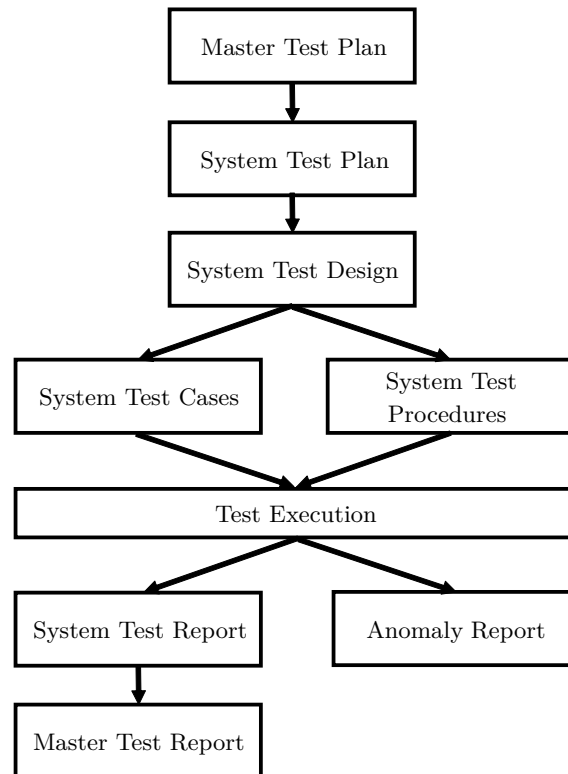


Abbildung 2.15. – Testdokumentation im Systemtest nach IEEE Std 829-2008 [50])

- Effektive Fertigstellung der Phasen und Verifikationsaktivitäten während jeder Phase des gesamten Safety Lifecycles (siehe Abbildung 2.9)
- Management der Funktionalen Sicherheit
- Ausführung des Assessments für die Funktionale Sicherheit

Zur Sicherstellung, dass diese Punkte auch erfüllt werden, stellt ISO 26262 folgende Anforderungen an die Qualität der Dokumente:

Die Dokumente sollen

- präzise und prägnant,
- eindeutig und klar strukturiert,
- einfach von der adressierten Zielgruppe verständlich und
- wartbar

sein.

Wichtig ist in ISO 26262 auch das Format der Dokumentation. Jedes Dokument sollte folgende formale Elemente enthalten:

- Titel, der sich auf den Bereich des Inhalts bezieht
- Autor und Freigebenden

- Eindeutige Identifikation jeder einzelnen Revision bzw. Version
- Änderungshistorie
- Status

Ein wichtiges Testdokument ist die sogenannte Testspezifikation. Nach [118] ist sie wie folgt definiert:

Definition 2.32 (Testspezifikation)

Eine Testspezifikation beschreibt Testfälle für den Test der fachlichen Funktionen und den Test nichtfunktionaler Anforderungen bzw. der technischen Funktionen.

Sie ist somit rahmengebend für alle Testaktivitäten.

Die Abbildung und Dokumentation des aktuellen Zustands erfolgt häufig auch über Metriken. Der Begriff Metrik ist nach [48] wie folgt definiert:

Definition 2.33 (Metrik)

Eine Metrik stellt eine quantitative Bewertung dar, zu welchem Grad ein System, eine Komponente oder ein Prozess ein Attribut erfüllt. Sie kann auch als Funktion verstanden werden, bei der Softwaredaten als Input eingespeist werden und ein numerischer Wert als Output generiert wird, der als Grad dafür interpretiert werden kann, inwieweit die Software ein Qualitätsattribut besitzt.

Deutlich wird hier, dass sehr viel Sorgfalt und Kontinuität hinsichtlich der Dokumentation gefordert wird. Sie wird häufig aufgrund des Aufwands vernachlässigt, was langfristig Probleme wie Inkonsistenzen, doppelte oder vernachlässigte Testausführung und verminderte Nachvollziehbarkeit nach sich zieht. Eine qualitativ hochwertige Dokumentation bildet daher die Grundlage für qualitativ hochwertiges Testen.

2.4.5. Testmethoden

In diesem Kapitel sollen die gängigsten Testmethoden kurz erläutert und abgegrenzt werden. Die grundsätzliche Unterscheidung zwischen statischen und dynamischen Testmethoden soll ebenso aufgezeigt werden wie die Unterschiede zwischen Black-, White- und Grey-Box-Testtechniken.

2.4.5.1. Statischer Test

Statische Testverfahren finden vor allem in der Softwareanalyse - also vor dem Systemtest - ihre Anwendung. Die Definition von statischem Testen ist nach [75]:

Definition 2.34 (Statischer Test)

Ein statischer Test ist ein Verifizierungs- und Analyseverfahren, bei dem

- keine Ausführung der zu prüfenden Software erfolgt,
- alle Analysen prinzipiell ohne Computerunterstützung durchgeführt werden können, meist jedoch durch entsprechende Softwareunterstützung bewerkstelligt werden,
- keine Testfälle ausgewählt werden,
- vollständige Aussagen über die Korrektheit und Zuverlässigkeit nicht getroffen werden können.

Es handelt sich hierbei hauptsächlich um formale (mit Hilfe eines Rechners) Verfahren wie z. B. Syntax-, Kontrollfluss-, Datenflussanalyse sowie Metriken und Maße und informelle Verfahren (ohne Hilfe eines Rechners) wie z. B. Inspektions- und Reviewtechniken [109]. Da beim Systemtest die innere Struktur nicht mehr einsehbar ist, sind statische Testverfahren nicht einsetzbar.

2.4.5.2. Dynamischer Test

Der dynamische Test ist ein Testverfahren, das die Ausführung der Software bzw. des Systems mit konkreten Eingaben, also Testdaten, verlangt [75]. Die Definition von dynamischem Testen ist nach [75]:

Definition 2.35 (Dynamischer Test)

Ein dynamischer Test ist ein struktur- oder funktionsorientiertes, diversifizierendes, bereichstestendes, statistisches, grenzwertanalysierendes oder zusicherungstechnisches Testverfahren, bei dem

- das System bzw. die übersetzte, ausführbare Software mit konkreten Eingaben versehen und ausgeführt wird,
- in der realen Betriebsumgebung getestet werden kann,
- stichprobenartig getestet wird,
- vollständige Aussagen über die Korrektheit und Zuverlässigkeit nicht getroffen werden können.

Das Ziel der dynamischen Testtechniken ist es, repräsentative, fehlersensitive, redundanzarme und ökonomische Testfälle zu erzeugen [75]. Die nachfolgend beschriebenen Testtechniken des Black-, White- und Grey-Box-Testens sind dynamische Testverfahren.

2.4.5.3. Black-Box-Testtechniken

Beim Black-Box-Test handelt es sich um spezifikations- bzw. anforderungsorientiertes Testen, d. h. die Eingaben in das System werden aus Spezifikationen abgeleitet und die innere Struktur des Systems ist nicht einsehbar (schwarze Box, Black-Box, siehe Abbildung 2.16). Nach [46] definiert sich Black-Box-Testen wie folgt:

Definition 2.36 (Black-Box-Test)

Ein Black-Box-Test ist eine Testtechnik, bei der für die Konstruktion von Testfällen ausschließlich das Ein- und Ausgabeverhalten des Systems herangezogen wird. Die innere Struktur bleibt gänzlich unberücksichtigt. Der Vorteil hierbei ist, dass der Tester sich rein auf die Angabe aus Anforderungen und Spezifikationen bezieht und die Implementierung der Funktion nicht mit in die Definition der Testfälle einbeziehen kann. Somit kann die Funktionalität ohne Berücksichtigung der Implementierung getestet werden.

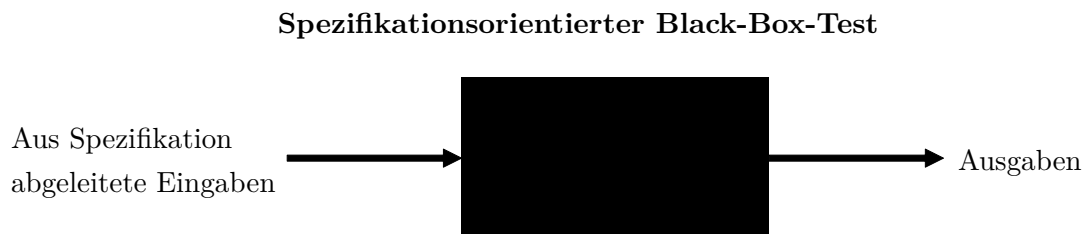


Abbildung 2.16. – Prinzip des Black-Box-Testens (nach [109])

Die gängigsten Black-Box-Test-Verfahren sind [46]:

- Äquivalenzklassentests:
Durch eine geringe Anzahl an Testfällen soll eine möglichst hohe Testabdeckung erreicht werden. Hierfür wird der Wertebereich aller Eingangsvariablen in Äquivalenzklassen, d. h. disjunkte Partitionen aufgeteilt. Variablen sind in derselben Äquivalenzklasse, wenn sie durch das System durch die gleiche Art und Weise bearbeitet werden. Anschließend wird eine beliebige Variablenbelegung ausgewählt und ein konkreter Testfall definiert.
- Grenzwertbetrachtung:
Ähnlich dem Äquivalenzklassentest wird eine Partition gebildet, die der Definition von Testfällen dient. Die Ableitung der Testfälle wird jedoch durch Randbereiche, d. h. Grenzwerte, bestimmt.
- Paarweises Testen:
Statt der semantischen Eigenschaften des Systems werden rein kombinatorische Überlegungen zur Testfalldefinition angestellt. Hierbei werden Ausprägungspaare von Faktoren paarweise getestet, sodass Doppelfehler aufgedeckt werden können.

Deutlich wird hier, dass man sich Methoden bedienen muss, um die große Anzahl der Testkombinationen und Einzeltests einzudämmen. Aus diesem Grund werden Grenzen und Äquivalenzklassen gebildet, die einen größeren Datenraum effektiv mit wenigen Testfällen abdecken können.

2.4.5.4. White-Box-Testtechniken

Im Gegensatz dazu existieren früher im Entwicklungsprozess auch White-Box-Tests, die kurz der Vollständigkeit halber dargestellt werden sollen. Hierbei wird die innere Struktur des Systems einbezogen und es handelt sich um implementationsorientiertes Testen, d. h. die Eingaben werden aus der Programmlogik bzw. der Implementierung abgeleitet. Das System ist also gläsern, einsehbar (weiße Box, White-Box, siehe Abbildung 2.17). Nach [46] definiert sich White-Box-Testen wie folgt:

Definition 2.37 (White-Box-Test)

Ein White-Box-Test ist eine Testtechnik, bei der für die Konstruktion von Testfällen die innere Struktur des Systems herangezogen wird.

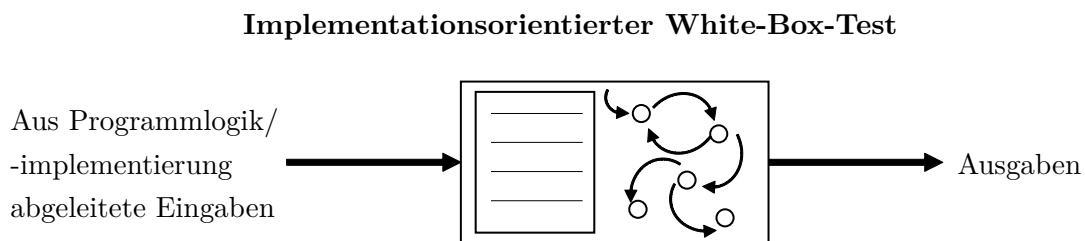


Abbildung 2.17. – Prinzip des White-Box-Testens (nach [109])

Die gängigsten White-Box-Test-Verfahren sind kontrollflussorientiertes und datenflussorientiertes Testen, die hier aber nur der Vollständigkeit halber genannt werden sollen. Für weiterführende Informationen sei auf einschlägige Literatur wie z. B. [46] verwiesen.

2.4.5.5. Grey-Box-Testtechniken

Als Zwischenstufe zwischen White-Box und Black-Box existieren auch sogenannte Grey-Box-Testtechniken. Die Testfälle werden hierbei aus der Anforderungs- und Schnittstellenbeschreibung unter Kenntnis der inneren Programmstruktur hergeleitet [46]. Definiert sind Grey-Box-Tests nach [46] wie folgt:

Definition 2.38 (Grey-Box-Test)

Ein Grey-Box-Test ist eine Testtechnik, bei der die Konstruktion von Testfällen nach den Prinzipien des Black-Box-Tests konstruiert werden, jedoch im Vorfeld wie beim White-Box-Test die innere Struktur des Systems betrachtet und dieses Wissen für die Testerstellung herangezogen wird. Das System ist also einerseits gläsern und einsehbar, beim Test selbst ist es jedoch nicht mehr einsehbar (graue Box, Grey-Box, siehe Abbildung 2.18).

Spezifikations- und implementationsorientierter Grey-Box-Test

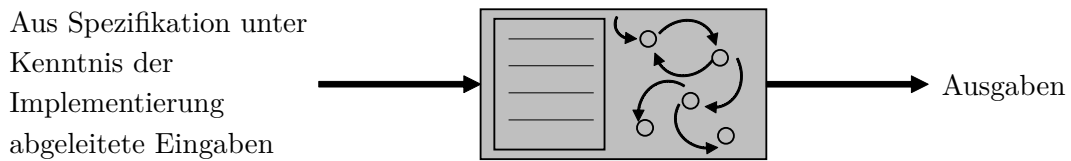


Abbildung 2.18. – Prinzip des Grey-Box-Testens (nach [109])

Deshalb werden die Testfälle somit nicht systematisch wie beim White-Box-Test aus der Programmstruktur hergeleitet [46].

2.4.6. Testsysteme

2.4.6.1. Testen in-the-Loop

In-the-Loop-Tests sind Testverfahren, die ein eingebettetes elektronisches System über Schnittstellen mit einer realen (z. B. Sensoren und Aktoren) oder virtuellen Umgebung (z. B. mathematische Modelle) im Labor verbindet. Dadurch kann die Reaktion des Systems analysiert und in das System zurückgespielt werden (d. h. im Regelkreis, also in-the-loop) [91]. Abbildung 2.19 zeigt die gängigsten in-the-loop-Verfahren schematisch. Die grundsätzliche Definition der einzelnen Verfahren ist nach [91]:

Definition 2.39 (Model-in-the-Loop, MiL)

Model-in-the-Loop testet das Funktionsmodell der Software, welches auf einem Entwicklungsrechner läuft.

Definition 2.40 (Software-in-the-Loop, SiL)

Software-in-the-Loop testet den Softwarecode, der auf einem Entwicklungsrechner läuft.

Definition 2.41 (Hardware-in-the-Loop, HiL)

Hardware-in-the-Loop testet das vollständige Steuergerät über die Ein- und Ausgabeschnittstellen über Simulationsrechner mit Bedienschnittstellen.

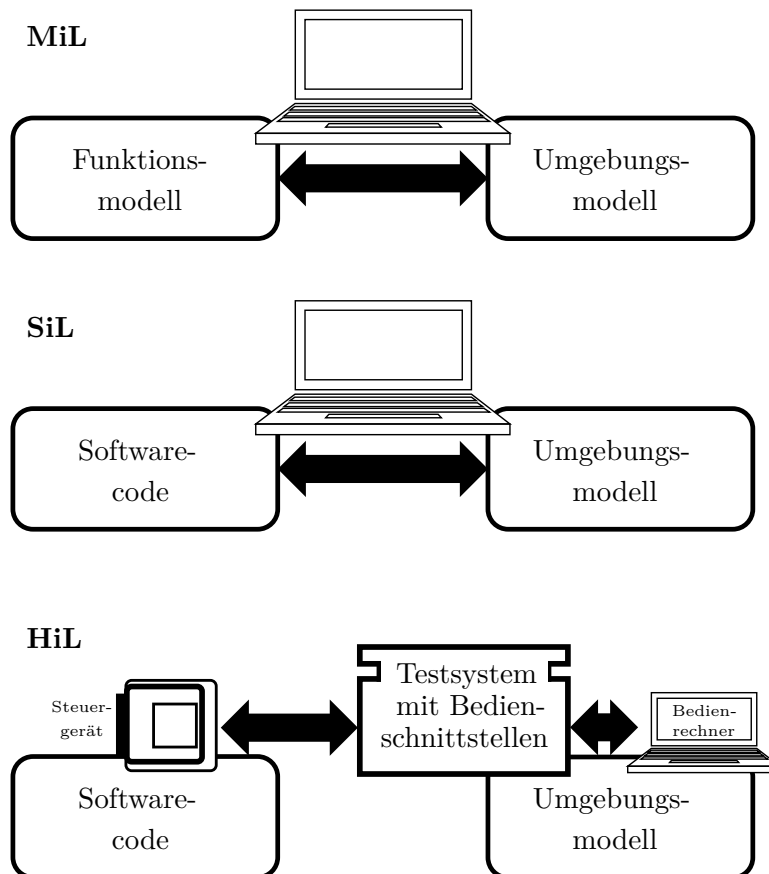


Abbildung 2.19. – Schematische Darstellung des MiL-, SiL- und HiL-Testens (nach [91])

Aufgrund des fortgeschrittenen Entwicklungsstandes handelt es sich beim Systemtest hauptsächlich um HiL-Testverfahren. Das Steuergerät wird über Bus-, Sensor-, Aktor und physikalische Schnittstellen an ein Testsystem angeschlossen und über Datenbasen, Restbussimulation und Stromversorgung genauso bedient, als wäre es bereits im realen Fahrzeug.

Häufig existieren je nach Entwicklungs- und Testprozess und den existenten Integrationsstufen noch weitere in-the-Loop-Verfahren wie z. B. Function-in-the-loop, Processor-in-the-Loop oder Vehicle-in-the-Loop.

2.4.6.2. Testumgebungen

Zum Test von komplexen Systemen sind leistungsfähige Testumgebungen notwendig, die effektive, durchgängige und konsistente Testverfahren ermöglichen. Grundsätzlich sind Testsysteme nach [44] wie folgt definiert:

Definition 2.42 (Testsystem)

Ein Testsystem ist ein rechnergestütztes Werkzeug, das den automatisierten Test von elektronischen Komponenten und Systemen gestattet.

Testsysteme lassen sich nach verschiedenen Merkmalen klassifizieren. Diese Merkmale sind in Tabelle 2.3 aufgeführt. Für den Systemtest, bei dem hauptsächlich HiL-Testsysteme

	Rapid- Prototyping	SiL- Testsystem	HiL- Testsystem	Onboard- Testsystem
Entwicklungsphase (siehe Spiralmodell in Abbildung 2.8)	A-Muster	A-Muster	B-/C-/D- Muster	C-/D-Muster
Realitätsnähe	hoch	gering	hoch	maximal
Elektrische Fehlersimulation	bedingt möglich	nicht möglich	möglich	bedingt möglich
Fehlerlokalisierung / Testtiefe	gering (White-Box, Unvollständigkeit)	hoch (White-Box)	gering (Grey-/ Black-Box)	gering (Grey-/ Black-Box)
Zeitanforderungen	Echtzeit	zeitsynchron	Echtzeit	Echtzeit
Automatisierungsgrad	gering	hoch	hoch	niedrig

Tabelle 2.3. – Testsystemmerkmale (nach [109])

(siehe Abbildung 2.20) eingesetzt werden, bedeutet dies eine hohe Realitätsnähe, elektrische Fehlersimulation sowie Echtzeitanforderung und einen hohen Automatisierungsgrad. Diese Merkmale zeigen, dass hohe Anforderungen an Testsysteme für den Einsatz im Systemtest gestellt werden. Gerade bei Betrachtung des Sicherheitssteuergeräts, das nicht nur im normalen Fahrbetrieb, sondern auch in Grenzbereichen wie der Pre-Crash-Phase und extremen Bedingungen wie der Crash-Phase fehlerfrei funktionieren muss, werden an ein Testsystem hohe Anforderungen gestellt. Es reicht nicht aus, aufgezeichnete Fahrdynamikdaten einzuspeisen, sondern Fehlersimulation, Crashdateneinspeisung und genaue Auswertung von Auslösezeiten sind notwendig. Allgemeine Anforderungen an HiL-Testsysteme sind nach [109]:

- **Nachvollziehbarkeit:** Die Nachvollziehbarkeit der Ergebnisse muss gewährleistet und durch Dokumentation belegt werden.
- **Reproduzierbarkeit:** Bei gleichen Bedingungen müssen gleiche Testergebnisse entstehen.
- **Fehlerfreiheit:** Das Testsystem muss fehlerfrei funktionieren und darf keine Fehler in das Testobjekt einbringen.
- **Vollständigkeit:** Das Testsystem muss die gesamte Peripherie der Komponente vollständig simulieren können, was die Vorgabe der Eingangsgrößen, Bewertung der Ausgangssignale, Buskommunikation und Diagnose beinhaltet [71].
- **Fehlergenerierbarkeit:** Das Testsystem muss spezifiziertes (Gutfälle) und nicht spezifiziertes Systemverhalten (Schlechtfälle) erzeugen können, indem die Simulationsmodelle und Schnittstellen Manipulationen ermöglichen.

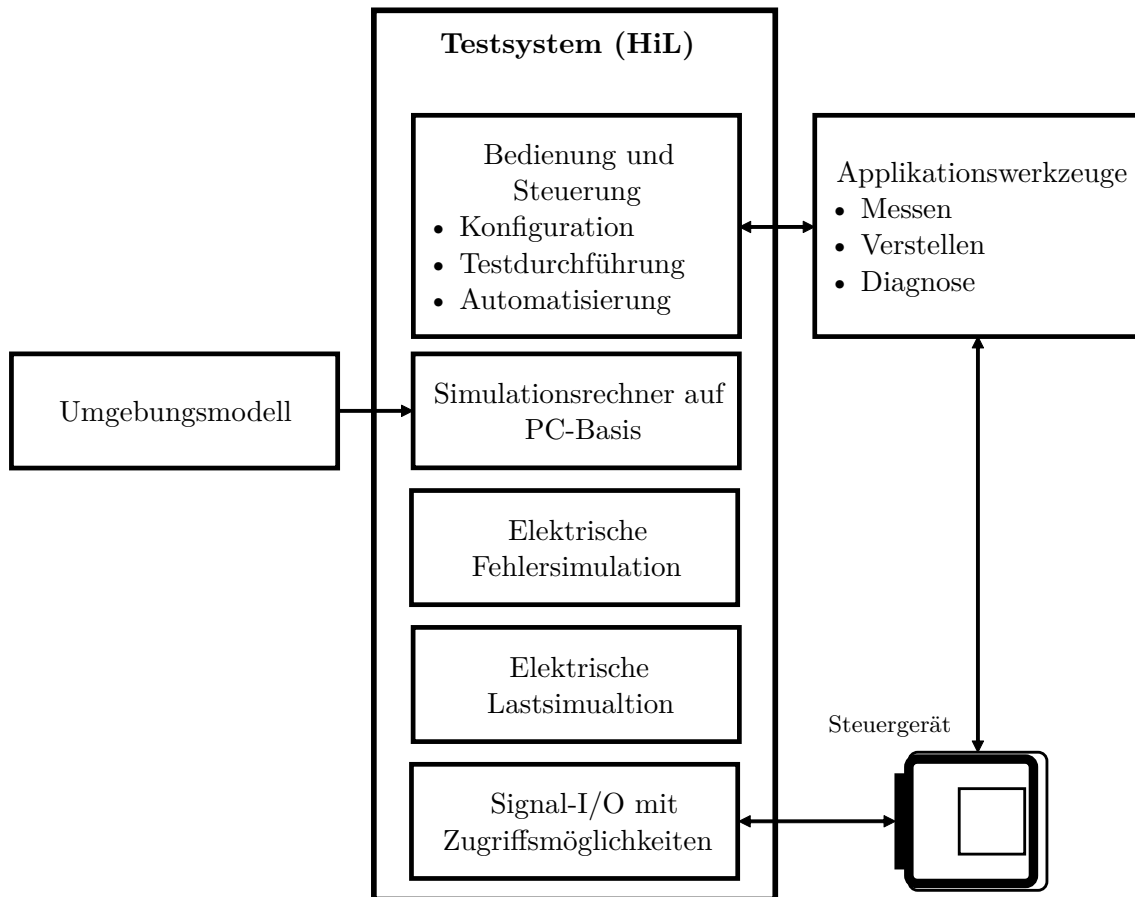


Abbildung 2.20. – Prinzipieller Aufbau eines Hardware-in-the-Loop-Testsystems (nach [91])

Die Universalität, Leistungsfähigkeit, Modularität und Durchgängigkeit von Testsystemen ist heute noch ein großes Forschungs- und Entwicklungsthema. Gerade die bereits vorgestellte Integration der Einzelsysteme und die Vernetzung der Komponenten stellt eine große Herausforderung für die Testausführung und -automatisierung dar.

2.5. Notwendigkeit der Arbeit

Die zentrale Bedeutung der Fahrzeugsicherheit im Alltag ist unverkennbar. Durch neuartige Sicherheitssysteme kann die Mobilität des Einzelnen sowie anderer Verkehrsteilnehmer wesentlich weniger gefährdend gestaltet werden. In der Entwicklung und dem Test von kontemporären Fahrzeugsicherheitssystemen herrschen sehr harsche Randbedingungen vor, was das Erreichen dieser Sicherheitsziele erschwert. Abbildung 2.21 zeigt, wie sich die Gegebenheiten in der Automobilbranche über Jahrzehnte hinweg verändert bzw. zugespitzt haben. Da sehr hohe Innovationszyklen kurzen Entwicklungszeiten gegenüberstehen, in der Fahrzeugsicherheit aber kein Fehlverhalten wie z. B. die Falschauslösung eines Airbags geduldet wird, gilt es diese Herausforderungen durch effiziente und effektive Methodiken für Entwicklung und Test zu meistern, um die Systemkomplexität beherrschbarer zu machen

gestern	Entwicklungsfaktoren	heute
Geringer Wettbewerb im Premium-Segment	Marktentwicklung	Großer Wettbewerb im Premium-Segment
Wenig Modellvarianten und Sonderausstattungen	Komplexität	Viele Modellvarianten, Nischen und Sonderausstattungen
Bewährte Technologie	Technologie	Kurze Innovationszyklen
Sieben Jahre und mehr	Entwicklungszeit	Vier Jahre und weniger
Geringe Vernetzung, überschaubar	Arbeitsweise	Hohe Vernetzung, viele Beteiligte
Geringer Kostendruck	Kosten	Hoher Kostendruck

Abbildung 2.21. – Veränderung der Entwicklungsfaktoren in der Automobilbranche (nach [73])

und die Funktionen abzusichern. Vor allem die Forderung nach Effizienzsteigerung, die durch Kostenreduktion, hohen Automatisierungsgrad sowie Standardisierung und Wiederverwendbarkeit gekennzeichnet ist, steht im Vordergrund der Optimierung der Prozesse. Auch die hohen Erwartungen an Produkt- und Systemreife sowie Qualität sind nicht leicht zu befriedigen. Die Forderung nach 100%iger Anforderungsüberdeckung und Prozesskonformität stellen Herausforderungen dar, die durch Standardmethoden nur schwer gemeistert werden können.

Einen Rahmen zur Entwicklung und dem Test von Steuergeräten im Automobilbereich geben die Normen *ISO/IEC 15504*, oder auch *SPICE (Software Process Improvement and Capability Determination)* genannt [58], und die Einführung der *ISO 26262: Road vehicles - Functional safety* im Jahr 2011 [57]. Durch die Vielfältigkeit der Problemstellungen und vielen Freiheiten, die diese Normen hinsichtlich Testmethodiken zulassen (siehe auch Kapitel 2.3), gibt es allerdings immer noch viel Raum für die Erforschung und Anwendung innovativer Methodiken.

Auch die steigende Vernetzung der Steuergeräte im Fahrzeug über verschiedene Bussysteme hinweg, stellt eine große Herausforderung für Entwicklung und Test dar. Abbildung 2.22 zeigt schematisch die Steuergeräteverflechtung in einem kontemporären Automobil. Es wird deutlich, dass eine Vielzahl an Steuergeräten - in einem aktuellen Oberklassefahrzeug

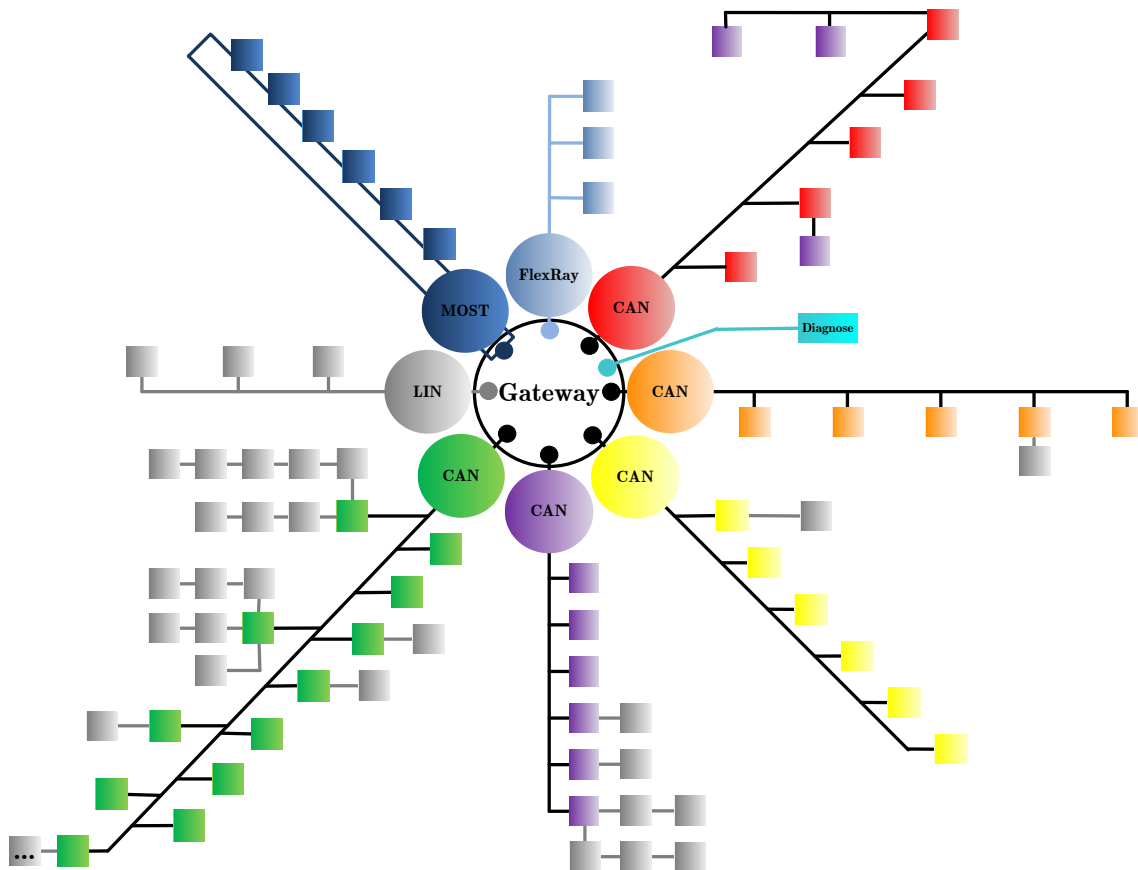


Abbildung 2.22. – Schematische, exemplarische Darstellung der Steuergerätevernetzung über verschiedene Bussysteme und ein zentrales Gateway hinweg in einem kontemporären Automobil (nach [64], [103], [104] und [105])

können es durchaus über 100 Stück sein - durch unterschiedliche Bussysteme verbunden sind [139]. Zum Austausch sicherheitskritischer Situationen müssen also Wege abgesichert und Zeitkriterien eingehalten werden. Auch die verteilte Entwicklung der einzelnen Steuergeräte durch verschiedene Zulieferer und OEMs und die Realisierung von Funktionen über Steuergerätegrenzen hinweg (z. B. Anforderung eines Bremsmanövers durch das Sicherheitssteuergerät an das Bremsensteuergerät), führen zu einer System- und Testkomplexität, die nur durch die Anwendung neuartiger Methodiken und Ansätze handhabbar ist. Auch aktuelle Veröffentlichungen wie [116] stellen heraus, dass die Komplexität von integralen Fahrzeugsicherheitssystemen unterschätzt wird und noch immer kein allgemein akzeptierter Freigabeprozess besteht.

Ein weiterer Aspekt, der zu einem gesteigerten Schwierigkeitsgrad in der Fahrzeugsicherheit führt, ist der Einsatz neuartiger Sensorsysteme wie zum Beispiel Radar, Lidar oder Kamera und deren mannigfaltige Möglichkeiten. Durch die Horizonterweiterung des Fahrzeugs durch vorausschauende Systeme steigt der Absicherungs- und Testaufwand dramatisch an. Ein Ziel ist es z. B. zukünftig auch durch vorausschauende Sensordaten Airbags zünden zu können, um schon vor Aufprall des Fahrzeugs einen kommenden Unfall detektieren zu können.

können und so einen wichtigen Zeitvorteil zu gewinnen. Auch der Trend hin zur Sensordatenfusion, also die Kombination verschiedener Umfeld- und Intertialsensorinformationen zu einer Informationsgesamtheit, die eine wesentlich bessere Abbildung der Realität als die Einzelinformationen wiedergibt, führt zu gesteigerter Komplexität. Die grundsätzliche Idee entspringt dem Themengebiet des autonomen oder automatischen Fahrens, spielt allerdings in der Fahrzeugsicherheit aufgrund des gesteigerten Informationsgewinns eine immer größere Rolle (siehe zum Beispiel [35], [36], [59], [69] oder [117]). Durch die komplexeren Sensorsysteme, mehr Hardware und Software sowie umfangreiche, vernetzte Funktionen entstehen mehr Fehlermöglichkeiten, die den Testaufwand erhöhen. All diese Tendenzen ziehen ebenfalls die Notwendigkeit zur Anwendung neuartiger Testmethodiken und -ansätze nach sich.

Die für die Fahrzeugsicherheit rahmengebende Norm ISO 26262 ist ein relativ junger Standard, dessen Integration in alle Prozesse noch nicht erfolgt ist. Die Umsetzenden sind immer noch im Prozess, die Prinzipien und Methodiken vollständig zu verstehen und in die Prozesse zu integrieren [17]. Auch bildet ISO 26262 die aktuell bereits eingesetzten Methodiken und Methoden ab. D. h. durch den Einsatz dieser Prozesse und Abläufe kann zwar ein gewisser Qualitätsstandard gewährleistet werden, ein Wettbewerbsvorteil durch innovative Methoden ist jedoch nicht zu erwarten. Dies zeigt deutlich, dass noch großes Potential in der tiefergehenden Erforschung und Erprobung der in ISO 26262 beinhalteten Methodiken existiert.

Eine weitere Problematik stellt die sehr große Anzahl an möglichen Signalzuständen, die z. B. über Bussysteme in ein Steuergerät eingespeist werden können, dar. Durch die Kommunikation eines Steuergeräts mit anderen Steuergeräten, Sensoren und Aktoren entsteht ein komplexes Geflecht, das sowohl Datenplausibilität als auch strenges Zeitverhalten voraussetzt, gleichzeitig aber bei Fehlverhalten eine abgesicherte Rückfallebene beinhalten muss. Häufig ist nicht ein einzelner Fehler verantwortlich für falsches Systemverhalten, sondern die Verkettung von Fehlern und der Zeitpunkt ihres Auftretens. Da aber die Kombination jedes Fehlers mit jedem anderen möglichen zu einer quasi unendlichen Anzahl an Testfällen führt, müssen intelligente Methodiken dazu eingesetzt werden, die relevanten Kombinationen zu identifizieren.

Durch bloßes Einspeisen aufgezeichneter realer Testdaten, wie es heute hauptsächlich praktiziert wird, kann eine vollumfängliche Testabdeckung nie erreicht werden. Testfahrten können keine gefährlichen Szenarien abdecken und auch aufgrund der Gefahr für Testfahrer Unfälle nicht einbeziehen. Des Weiteren ist die Konsistenz der Daten ein wichtiger Aspekt. Das sprungartige Verändern von fahrdynamischen Größen ist aufgrund von Plausibilisierung durch das Steuergerät nur begrenzt möglich. Dies begründet die Notwendigkeit simulativer Datenerzeugung, die die synthetische Szenarienerzeugung durch Simulation und Dateneinspeisung durch Testsysteme ermöglicht. Ein konsistenter Datenstrom ist notwendig, um ein Steuergerät möglichst realitätsnah zu testen. Durch diesen Ansatz können flexibel und in großem Umfang Testdaten erzeugt werden, die auch Szenarien beinhalten, die aufgrund von Gefahr oder hohen Kosten von realen Testfahrten nicht erzeugt werden können. Simulative Manöver können reproduziert und variiert werden und

durch Kombination mit mathematischen Verfahren und Algorithmen genau analysiert und optimiert werden. Durch die exakte Festlegung von Manövern ergeben sich dann neue Testmöglichkeiten: ein fahrdynamisches Szenario kann von normaler Fahrt über die Pre-Crash-Phase hin zu einem Unfall, d. h. Aufprall simuliert werden. Zum Aufprallzeitpunkt können dann passende Crashdaten eingespeist werden, sodass ein durchgängiges Unfallmanöver entsteht. Da Crashdaten aufgrund ihrer aufwändigen Erzeugung durch Crashtests oder Finite-Elemente-Methode nur in sehr begrenzter Zahl vorhanden sind, muss sich das Fahrdynamikmanöver den Crashdaten anpassen, um einen konsistenten Datenstrom, d. h. die Fahrdynamikdaten müssen zu den Crashdaten passen, zu erzeugen. Dies ist nur durch Analyse und Optimierung der Simulationsszenarios möglich. Ein möglicher Ansatz hierfür soll in dieser Arbeit gezeigt werden.

Testingenieure im Systemtest besitzen in erster Linie Knowhow im Bereich des Testens und der zugehörigen Rahmenbedingungen und wenig oder kaum Wissen über Fahrdynamik. Da sie jedoch im Systemtest mit dem Test von Funktionen konfrontiert werden, die sich häufig auf fahrdynamische Daten stützen, ist bei der Testfallerstellung oft eine enge Abstimmung mit Algorithmenentwicklern notwendig, die die Expertise im Bereich Fahrdynamik haben. Um diese häufig zeitraubende Abhängigkeit zwischen Test und Funktionsentwicklung zu lockern, sind Methoden hilfreich, die den Testingenieuren erlauben, fahrdynamische Testfälle auch ohne tiefgehendes Knowhow im Bereich der Fahrdynamik zu genauen, definierten Datenzuständen zu erzeugen, zu analysieren und in das System einzuspeisen. Eine Analyse- und eine Optimierungsmethode für manöverbasierte, simulative Testfälle wird in dieser Arbeit vorgestellt.

Durch Testmetriken können zusätzlich Aussagen über Testabdeckung und -fortschritt getroffen werden, um eine objektive Einschätzung zur Testqualität und -quantität zu geben. Aktuell werden solche Aussagen häufig rein durch den Erfahrungsschatz langjähriger Test- und Projektmitarbeiter getroffen, die den Projektstand beurteilen können. Durch eine zahlen- und bewertungsgestützte Metrik kann diese Erfahrung nachvollziehbar festgehalten und dokumentiert werden, was Überblick für den Projektverlauf verschafft und als Grundlage für die Bemessung des Testumfangs dienen kann.

All diese offenen Fragen und Herausforderungen führen zu der Prämisse, weiter an Testmethodiken, -methoden und -metriken im Bereich der integralen Fahrzeugsicherheit zu forschen. Die heute noch bestehende Lücke zwischen Software- bzw. Modultest und realer Testfahrt kann durch die Einspeisung aufgezeichneter Daten über ein HiL-Testsystem allein nicht bewältigt werden. Durch relevante Testfälle, die als Szenarien durch Simulation synthetisch erzeugt und über HiL-Testsysteme in Steuergeräte eingespeist werden, können sicherheitsrelevante Systeme umfassender getestet werden. Diese Arbeit erarbeitet eine rahmengebenden, bewertenden Testmetrik und die Erprobung von Testmethoden und leistet so einen Beitrag zur Bewältigung dieser Aufgaben.

Fazit

In diesem Kapitel wurden die Rahmenbedingungen für die Eingruppierung der Dissertation dargelegt. Die Entwicklung der integralen Fahrzeugsicherheit wurde ebenso dargestellt wie der Aufbau kontemporärer Steuergeräte. Der für die Automobilbranche wichtige Sicherheitsstandard ISO 26262 wurde kurz vorgestellt und die wichtigsten Prinzipien erklärt. Außerdem wurde der Systemtest im Entwicklungs- und Testprozess anhand gängiger Vorgehensmodelle und Anforderungen beschrieben sowie hinsichtlich der Teststrategien, -methoden und -systeme erörtert. Abschließend wurde die Notwendigkeit der Arbeit anhand der aktuellen Gegebenheiten in der Fahrzeugsicherheit begründet.

3. Komplexitätsbewertung und Fehlerschätzung



Abbildung 3.1. – Teil 1 der in der Arbeit vorgeschlagenen Methodik (siehe auch Abbildung 1.3).

In diesem Kapitel wird eine neue Methode zur Komplexitätsbewertung für Projekte in der Fahrzeugsicherheit erarbeitet. Anhand der Einführung von Expertenaudits, die mit verschiedenen, gewichteten Kennzahlen und Faktoren ein Projekt abbilden, kann eine objektive Einschätzung über die Projektkomplexität gegeben werden. Durch diese Komplexitätsbewertung können die Anzahl der zu erwartenden Fehler abgeschätzt werden, die wiederum nach dem Test ein Indiz für die Testqualität bzw. ein Testendekriterium sein können.

3.1. Warum Kennzahlen und Indikatoren?

Die zentrale Frage des Testens lautet seit jeher: Wann ist das Testen zu Ende? Wann kann man aufhören zu testen? Habe ich die richtigen Testfälle ausgewählt? Sind gefährliche Fehler gefunden worden? Grundsätzlich könnte man jederzeit aufhören oder gar nicht erst damit beginnen, da man die Abwesenheit von Fehlern nicht beweisen kann [118]. Da jedoch wie in Definition 2.15 dargestellt die Ziele des Systemtests die Vermeidung von Schaden und das Schaffen von Vertrauen in das System sind, ist es natürlich keine Lösung, nicht zu testen. Wichtig ist es, Testmethoden und -fälle so auszuwählen, dass das System guten Gewissens verwendet werden kann und sich Metriken zu bedienen, die abschätzen können, wie qualitativ und quantitativ getestet wurde.

Ein wichtiges Medium von Metriken sind sogenannte Kennzahlen. Kennzahlen sind nach [135] und [39] definiert zu:

Definition 3.1 (Kennzahlen)

Kennzahlen fassen messbare, relevante Daten zusammen und stellen sie in einen größeren Zusammenhang. Kennzahlen sind quantitative Informationen, die für Bedürfnisse der Analyse und Steuerung aufbereitet werden. Kennzahlen leisten

- die Messung von betrieblichen Vorgängen,
- die Beurteilung von betrieblichen Sachverhalten,
- die kurze und prägnante Darstellung komplexer Sachverhalte,
- die Festlegung zukünftiger Maßstäbe und
- die Festlegung kritischer Erfolgsfaktoren.

Des Weiteren existieren neben Kennzahlen noch Indikatoren. Sie sind nach [39] definiert zu:

Definition 3.2 (Indikatoren)

Indikatoren sind quantitative Informationen, die als Ersatzgrößen Schlüsse auf die Ausprägung oder Veränderung anderer wichtiger Größen zulassen. Indikatoren sind häufig nicht direkt messbar und werden auch als weiche Faktoren bezeichnet.

Kennzahlen und Indikatoren finden häufig Anwendung zur Entscheidungsfindung, Strategieplanung und Absicherung [2]. Grundsätzlich wird zwischen absoluten und relativen Kennzahlen unterschieden, die jedoch meist aus Bilanzen und anderen Unterlagen entnommen werden [135]. Durch Kennzahlen und Indikatoren werden also Werte und Prozesse abgebildet und dargestellt, die dann eine Basis zur weiteren Entscheidung bilden.

Neben der Anwendung für Management, Controlling etc. kann diese Methode auch auf Entwicklungsprojekte übertragen werden. Eine wichtige Frage, die sich Projektmanager und Tester vor dem Testen stellen ist, welche Eigenschaften ein Projekt hat. Die Komplexität von Projekten wird in erster Linie durch Erfahrung und „Gefühl“ ausgedrückt. Die

Komplexität eines Projekts bestimmt aber maßgeblich den Testumfang und die Auswahl der Testmethoden. Um dies jedoch auch objektiv und nachvollziehbar festzuhalten, kann man sich Kennzahlen und einem sogenannten Expertenaudit bedienen, um die Erfahrung der Projektmitarbeiter darzustellen. Weiterführend wird klar werden, dass man hier nicht ausschließlich von dem Begriff Kennzahlen sprechen kann, da nicht alle Faktoren für die Komplexität eines Projekts auf messbaren Größen beruhen, sondern häufig auch Schätzung, Einschätzungen und Meinungen einbezogen werden. Es werden also auch Indikatoren verwendet. Es ist wird sich jedoch zeigen, dass es dennoch sinnvoll ist, Projekte nicht nur technisch mit Kennzahlen messbar, sondern auch auf soziotechnischer Ebene durch Indikatoren zu bewerten, um die realen Einflüsse abzubilden.

3.2. Expertenaudit zur Projektkomplexitätsbewertung

Um eine Bewertung von Komplexität durchzuführen, stellt sich zunächst die Frage, was *Komplexität* eigentlich bedeutet. Der Duden definiert „Komplexität“ zu *Gesamtheit aller Merkmale oder Möglichkeiten und Vielschichtigkeit* [22]. Nach [119] kann Komplexität einerseits als strukturell, also als Relation der Zahl verknüpfter Elemente und der Anzahl der Verbindungen gesehen werden. Andererseits kann die Betrachtung semantisch abgeleitet von der Kommunikationstheorie, d. h. die Relation der unterschiedlichen Wörter zur Gesamtheit der Wörter, erfolgen. Deutlich wird also, dass es viele verschiedene Ansätze zur Komplexitätsdefinition gibt [9]. Zur Anwendung der Komplexität auf Projekte, muss ein gesamtheitlicher Ansatz gefunden werden, der die unterschiedlichen Gegebenheiten in Projekten abbilden kann. Gerade die Anwendung von Kennzahlen und Indikatoren stellt hier ein Maß dar, das das Projektumfeld gut beschreiben kann. Zusammengefasst kann man hier von beeinflussenden Faktoren für die Komplexität sprechen, die die Gesamtheit der Projektkennzahlen und -indikatoren abbilden.

Zur Eruiierung und Bewertung der die Projektkomplexität beeinflussenden Faktoren soll ein sogenanntes Expertenaudit herangezogen werden. Der Begriff Audit stammt aus dem Lateinischen *audire* und bedeutet *zuhören, erfahren*. Ein Audit ist allgemein nach [38] definiert zu:

Definition 3.3 (Audit)

Ein Audit stellt eine objektive Untersuchung dar, die regelmäßig Aktivitäten, Ziele und Vorgaben hinterfragt.

Das hier nachfolgend verwendete Expertenaudit soll ebenso objektiv und regelmäßig durchgeführt werden, um die aktuellen Gegebenheiten im Projekt zu hinterfragen und jeweils aufs Neue festzuhalten. Am wichtigsten ist die Bewertung jedoch zu Beginn eines Projekts, um schon am Anfang geeignete Maßnahmen zu treffen. Die regelmäßige Wiederholung stellt dann die Aktualität der Bewertung sicher. Die operative Person des Audits soll ein Experte für das Projekt sein. Optimalerweise handelt es sich um eine Person, die Überblick über das gesamte Projektgeschehen hat, um die Vergleichbarkeit der Bewertungen zwischen

den Projekten zu gewährleisten.

Die Komplexitätsbewertung eines Projekts soll während des Testprozesses als Grundlage für Testfallauswahl, Testumfang und Testqualitätsbewertung dienen.

3.2.1. Die Projektkomplexität beeinflussende Faktoren

Für die Projektkomplexitätsbewertung sollen Faktoren, also entsprechende Kennzahlen und Indikatoren zur Bestimmung der Komplexität, dienen. Durch Analyse von Projekten und Befragung von langjährigen Projektmitarbeitern können diese eruiert werden. Mögliche Faktoren für Projekte in der Fahrzeugsicherheit sind in Tabelle 3.1 dargestellt. Jeder $n_{K,i}$ Faktor wird jeweils mit einer Bewertungsskala von 0 bis 5 belegt, mithilfe derer jeweils eine Bewertungszahl $B_{K,i}$ zugewiesen wird.

Faktor $n_{K,i}$	Bewertung $B_{K,i}$	Beschreibung
Anzahl der Entwickler	5 - Zu wenig 2,5 - Durchschnittlich 0 - Ausreichend	Eine geringe Anzahl an Entwicklern bezogen auf den Projektaufwand führt aufgrund von wachsendem Zeitdruck und Stress zu vermehrten Fehlern.
Anzahl der an der Entwicklung involvierten Standorte	5 - Ein Standort 2,5 - Durchschnittliche Anzahl, zwei Standorte 0 - Weit mehr als zwei Standorte	Je mehr Standorte beteiligt sind, desto wahrscheinlicher werden Fehler aufgrund von Kommunikationsproblemen und Missverständnissen.
Anzahl und Qualität der Anforderungen	5 - Unvollständige, unklare Anforderungen; zusätzliche Anforderungen während der Projektlaufzeit 2,5 - Durchschnittlich in Umfang und Qualität 0 - Ausreichende, eindeutige Anforderungen; wenige zusätzliche Dokumente.	Sind Anforderungen schlecht formuliert kommt es zu Missverständnissen und Fehlentwicklungen. Gute Anforderungen sind genau und eindeutig sowie gut verständlich und nachvollziehbar formuliert.
Bekanntheit des Auftraggebers	5 - Erster Auftrag 2,5 - Bekannt 0 - Häufige Aufträge	Bei der Zusammenarbeit mit einem neuen Auftraggeber sind Missverständnisse oft Ursache vermehrter Fehler.

Faktor $n_{K,i}$	Bewertung $B_{K,i}$	Beschreibung
Entwicklungszeit	5 - Sehr kurz 2,5 - Durchschnittlich 0 - Ausreichend	Bei einer knapp bemessenen Entwicklungszeit bezogen auf den Projektaufwand werden Fehler aufgrund von Zeitdruck und Stress wahrscheinlicher.
Erfahrung der Entwickler	5 - Sehr wenig Erfahrung im Fachgebiet; nicht alle sind im Entwicklungsprozess geschult 2,5 - Durchschnittliche Erfahrung im Fachgebiet; alle sind im Entwicklungsprozess geschult 0 - Experten im Fachgebiet; alle sind im Entwicklungsprozess geschult	In einem Projektteam deren Mitarbeiter über wenig Erfahrung verfügen kommt es häufiger zu Problemen und Fehlern aufgrund von weniger effizientem und effektivem (Zusammen-)Arbeiten.
Komplexität der Funktionen	5 - Sehr hohe Komplexität durch starke Vernetzung, Zentralität, ASIL und Beeinflussung 2,5 - Durchschnittlich 0 - Geringe Komplexität	Aufgrund der stärkeren Vernetzung, Zentralität, ASIL und geringeren Überschaubarkeit anspruchsvoller Funktionen treten vermehrt Fehler auf.
Pilotierung von Funktionen	5 - Sehr viele neue Funktionen 2,5 - Durchschnittlich 0 - Keine neuen Funktionen	Bei der Implementierung (vieler) neuer Funktionen werden oft neue Themengebiete eröffnet und neue Herangehensweisen sind gefragt, was zu einer größeren Fehleranzahl führen kann.
Reuse aus früheren Projekten	5 - Kein/Kaum Reuse möglich 2,5 - Reuse in Teilen möglich 0 - Sehr viel Reuse möglich	Je weniger aus früheren Projekten wiederverwendet werden kann, desto größer ist der verbundene Arbeitsaufwand, woraus mehr Fehler resultieren.

Faktor $n_{K,i}$	Bewertung $B_{K,i}$	Beschreibung
Umfang des Projekts	5 - Sehr viele Funktionen 2,5 - Durchschnittlich 0 - Sehr wenige Funktionen	Eine hohe Anzahl an Funktionen bezogen auf den Vergleich mit anderen Projekten erhöht die Fehlerwahrscheinlichkeit aufgrund des wachsenden Umfangs.

Tabelle 3.1. – Mögliche beeinflussende Faktoren für die Komplexität von Entwicklungsprojekten der Fahrzeugsicherheit (nach [9]). Die Bewertung erfolgt jeweils in fünf Stufen, wobei höhere Zahlen nahe fünf für kritischere Bewertung stehen.

Natürlich können spezielle Faktoren ergänzt werden, die je Entwicklungsgegebenheiten spezifisch sein können. Die eruierten Faktoren können anhand eines Expertenaudits je Projekt bewertet werden und so einen Überblick über die Komplexität des Projekts geben. Zur realistischeren Bewertung sollten die Komplexitätsfaktoren jedoch noch gewichtet werden, da nicht alle denselben Einfluss haben.

3.2.2. Gewichtung der Faktoren

Zur Gewichtung der die Projektkomplexität beeinflussenden Faktoren kann man sich dem sogenannten paarweisen Vergleich bedienen. Mithilfe des paarweisen Vergleichens können durch systematisches Gegenüberstellen Kriterien in eine nach Wichtigkeit sortierte Reihenfolge gebracht werden. Jedes Kriterium wird mit jedem anderen Kriterium verglichen und bewertet, welches der beiden dominiert. Dominanzaspekte können z. B. Wichtigkeit, Kosten oder Erfolg sein [11], [65].

Tabelle 3.2 zeigt den paarweisen Vergleich anhand der Befragung eines Experten und somit auch eine mögliche Gewichtung der Faktoren für die Projektkomplexität. Hierbei wird deutlich, dass die beeinflussenden Faktoren gravierende Unterschiede in ihrem Beitrag zur Komplexität aufweisen können. Während einige wenige Faktoren einen Großteil der Komplexität ausmachen, haben andere einen sehr geringen Einfluss. Es zeigt sich in dieser Bewertung sogar, dass ein Faktor (die involvierten Standorte) ganz vernachlässigt werden kann.

3.2.3. Berechnung der Projektkomplexität

Die Berechnung der Projektkomplexität kann nun anhand der Faktoren und der zugehörigen Gewichtung erfolgen. Durch ein Expertenaudit wird das Projekt hinsichtlich der Faktoren mithilfe der Bewertungsskala begutachtet. Die Komplexitätszahl K eines Projekts errechnet

Kriterium 1 → Kriterium 2 ↓	Erfahrung der Entwickler	Anzahl u. Qualität der Anforderungen	Komplexität der Funktionen	Reuse aus früheren Projekten	Pilotierung von Funktionen	Umfang des Projekts	Anzahl der Entwickler	Bekanntheit des Auftraggebers	Entwicklungszeit	Anzahl der Entwicklungs-Standorte	
Erfahrung der Entwickler	■	1	0	0	0	0	0	0	0	0	
Anzahl u. Qualität der Anforderungen	1	■	0	0	0	0	0	0	0	0	
Komplexität der Funktionen	2	2	■	1	0	1	0	0	0	0	
Reuse aus früheren Projekten	2	2	1	■	1	1	0	0	0	0	
Pilotierung von Funktionen	2	2	2	1	■	1	0	0	0	0	
Umfang des Projekts	2	2	1	1	1	■	2	1	1	0	
Anzahl der Entwickler	2	2	2	2	2	0	■	1	1	0	
Bekanntheit des Auftraggebers	2	2	2	2	2	1	1	■	0	0	
Entwicklungszeit	2	2	2	2	2	1	1	2	■	0	
Anzahl der Entwicklungs-Standorte	2	2	2	2	2	2	2	2	2	■	
Summe der Punkte	17	17	12	11	10	7	6	6	4	0	90
Rang	1	1	2	3	4	5	6	6	7	8	
Gewichtung in Prozent $G_{K,i}$	18,9	18,9	13,3	12,2	11,1	7,8	6,7	6,7	4,4	0,0	100

Legende:

- Kriterium 1 ist wichtiger als Kriterium 2 2:0
- Kriterium 1 ist weniger wichtig als Kriterium 2 0:2
- Kriterium 1 ist genauso wichtig als Kriterium 2 1:1

Tabelle 3.2. – Paarweiser Vergleich und somit Gewichtung der Faktoren für die Projektkomplexität.

sich für jeden der n_K Faktoren mit dem gewichteten Einfluss $G_{K,i}$ und der zugehörigen Bewertung $B_{K,i}$ für die Komplexität zu:

$$K = \sum_{i=1}^{n_K} G_{K,i} \cdot B_{K,i} \tag{3.1}$$

Je näher die Komplexitätszahl K für ein Projekt an fünf ausfällt, umso kritischer ist dies zu betrachten. Es kann also als Schwierigkeitsgrad des Entwicklungsprozesses betrachtet werden.

Die nachfolgende Verwendung der Komplexität ist in Abbildung 3.2 dargestellt. Einerseits wird nachfolgend in Kapitel 3.3 eine Fehlerschätzung durchgeführt, die für ein laufendes Projekt anhand der Erfahrungswerte vergangener Projekte die Anzahl der zu erwartenden

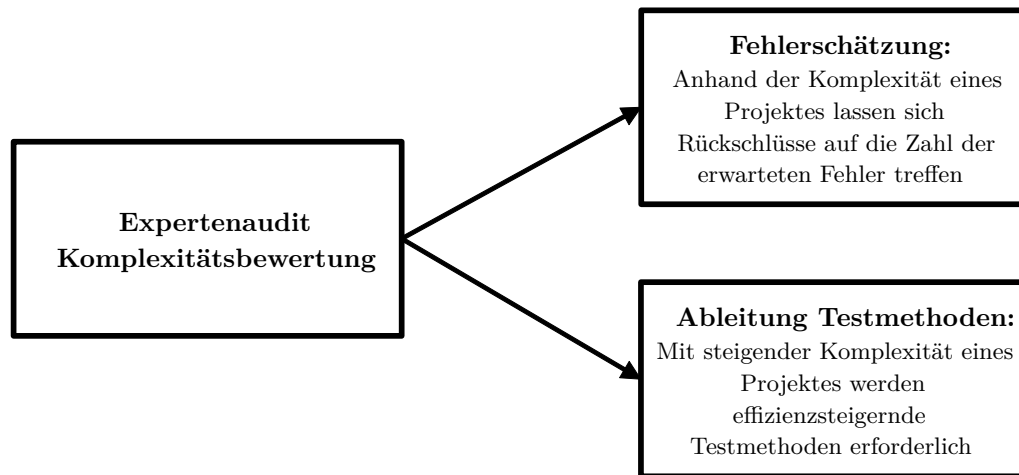


Abbildung 3.2. – Ableitung von Fehlerschätzung und Testmethoden aus der Projektkomplexität [9].

Fehler schätzt. Andererseits können anhand der Komplexität Maßnahmen hinsichtlich Testmethoden getroffen werden. Je komplexer ein Projekt ist, desto notwendiger ist der Einsatz effizienzsteigernder Testmethoden und die Erhöhung und Intensivierung des Testumfangs. Dies wird in Kapitel 4.1 genauer ausgeführt.

3.3. Fehlerschätzung

Eine mögliche Verwendung der Komplexitätsbewertung von Projekten kann die Vorhersage von zu erwartenden Fehlern sein. Je komplexer ein Projekt ist, also je höher die Komplexitätszahl K ausfällt, desto mehr Fehler sind im Systemtest zu erwarten. Die genaue Vorhersage der Fehler ist natürlich nicht möglich, da jedes Projekt andere Gegebenheiten hat, aber zumindest eine grobe Abschätzung der Erwartung der Fehler ist dadurch möglich. Um die Schätzung zu visualisieren ist die Auswertung von einigen Projekten notwendig. Nach der Projektkomplexitätsschätzung können retrospektiv die Systemtestfehler dieser Projekte über ihren Verlauf aufgetragen und analysiert werden.

3.4. Beispiel

Nachfolgend soll eine beispielhafte Bewertung und Berechnung der Komplexität für fünf abgeschlossene und drei aktuelle, parallel laufende Projekte aus der Fahrzeugsicherheit durchgeführt werden. Tabelle 3.3 zeigt die Ergebnisse der Bewertungen anhand des Expertenaudits. Deutlich wird, dass die Ergebnisse für die Komplexität K unterschiedlich ausfallen und die Projekte somit nicht demselben Schwierigkeitsgrad entsprechen: während bei den abgeschlossenen Projekten zwei höhere Werte haben, sind zwei Projekte im Durchschnitt und ein Projekt hat einen niedrigen Wert. Es existieren also gravierende

	Erfahrung der Entwickler, $E_{K,1} = 18,9\%$	Anzahl u. Qualität der Anforderungen, 18,9%	Komplexität der Funktionen, 13,3%	Reuse aus früheren Projekten, 12,2%	Pilotierung von Funktionen, 11,1%	Umfang des Projekts, 7,8%	Anzahl der Entwickler, 6,7%	Bekanntheit des Auftraggebers, 6,7%	Entwicklungszeit, 4,4%	Anzahl der Standorte, 0,0%	K
Abgeschlossen											
$B_{K,i}$ Projekt 1	2	2	4,5	2	3,5	3,5	3,5	1	5	5	2,78 ↑
$B_{K,i}$ Projekt 2	1,5	2	3	2,5	3,5	3	2,5	0	2,5	2,5	2,27 →
$B_{K,i}$ Projekt 3	2	2,5	2,5	3	1	2,5	1	2,5	2,5	2,5	2,20 →
$B_{K,i}$ Projekt 4	2	1,5	2,5	2	1	3	1	0,5	2,5	2,5	1,79 ↓
$B_{K,i}$ Projekt 5	2,5	2,5	4	2,5	3	3	2,5	2	3,5	2,5	2,80 ↑
Laufend											
$B_{K,i}$ Projekt 6	1,5	1,5	4,5	3,5	4,5	4,5	1	2,5	1,5	2,5	2,82 ↑
$B_{K,i}$ Projekt 7	1,5	1,5	4	2	2,5	4	1	2,5	0,5	5	2,24 →
$B_{K,i}$ Projekt 8	1,5	1,5	3	2,5	1	3,5	2,5	2,5	0,5	2,5	1,94 ↓

Tabelle 3.3. – Projektbewertung für fünf abgeschlossene und drei laufende Beispielprojekte

Unterschiede zwischen parallel laufenden Projekten, was auch unterschiedliche Maßnahmen in der Testplanung, im Umfang und den Methoden erfordert. Durch die retropektivische Bewertung der fünf abgeschlossenen Projekten wird auch eine Basis für zukünftige Bewertungen geschaffen. Durch Vergleich mit den vergangenen Projekten wird eine Bewertung für neu anlaufende Projekte wesentlich erleichtert. Bei den drei aktuell laufenden Projekten wird deutlich, dass es sich ebenfalls um drei sehr unterschiedliche Projekte handelt: Projekt 6 ist sehr komplex, Projekt 7 eher durchschnittlich und Projekt 8 unterdurchschnittlich. Projekt 6 ist durch die ähnliche Komplexität mit den Projekten 1 und 5, Projekt 7 mit den Projekten 2 und 3 und Projekt 8 mit Projekt 4 vergleichbar. Zu erwarten ist also in etwa ähnlicher Ablauf des Systemtests.

Um nun eine Verbindung zwischen der Komplexitätsschätzung und dem realen Testverlauf herzustellen, wurden für die fünf vergangenen Projekte jeweils die im Projektverlauf gefundenen Systemtestfehler untersucht. Abbildung 3.3 zeigt den zeitlichen Verlauf der Fehler. Es zeigt sich, dass die Höhe der Komplexität auch mit der Anzahl der gefundenen Systemtestfehler einhergeht: Projekte 5 und 1 haben die höchste Komplexität und auch die meisten Fehler. Die Projekte 2 und 3 haben durchschnittliche Komplexität und wesentlich

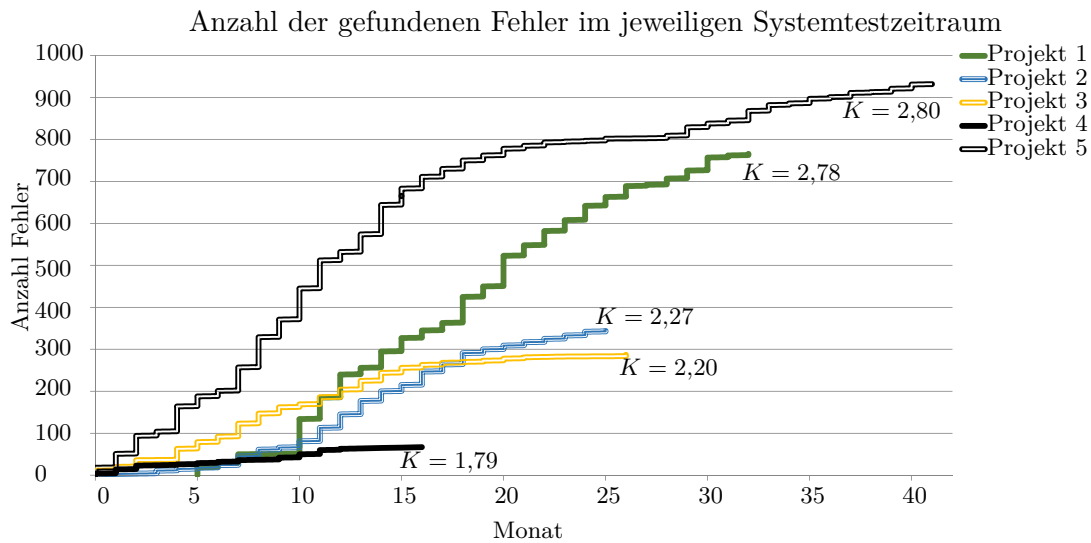


Abbildung 3.3. – Im Systemtest aufgetretene Fehler der Projekte aus Tabelle 3.3. Deutlich wird, dass die Komplexität mit der Anzahl der Fehler korreliert. Das kritischste Projekt 5 hat auch die meisten Systemtestfehler und das unkritischste Projekt 4 die wenigsten.

weniger Fehler. Projekt 4, das eine niedrige Komplexität aufweist hat auch mit Abstand die wenigsten Systemtestfehler. Abbildung 3.4 zeigt den Zusammenhang zwischen der Komplexität K und den gefundenen Fehlern noch einmal genauer. Deutlich wird durch diese Darstellung und die eingefügte Trendlinie, dass ein Zusammenhang zwischen der Komplexität und der Anzahl der gefundenen Fehler besteht. Die Trendlinie wurde durch folgende Formel approximiert, wobei x durch die Komplexität K und y durch die Fehleranzahl F repräsentiert wird:

$$y = 425,7x^2 - 1167,3x + 794,05 \text{ bzw. } F = 425,7K^2 - 1167,3K + 794,05 \quad (3.2)$$

Eine Vorhersage durch die Komplexitätszahl K für die aktuellen Projekte ist also möglich. Für zukünftig anlaufende Projekte kann nun durch die Komplexitätsschätzung eine erste grobe Schätzung für die Anzahl der Systemtestfehler gegeben werden. Zu erwarten sind für die drei laufenden Projekte die Fehlerwerte aus Tabelle 3.4.

Deutlich wird, dass die Projekte jeweils vergangen in Komplexität und vorhergesagter

Projekt	K	$F = 425,7K^2 - 1167,3K + 794,05$
Projekt 6	2,82	888
Projekt 7	2,24	315
Projekt 8	1,94	132

Tabelle 3.4. – Vorhersage der erwarteten Fehler im Systemtest für drei laufende Projekte.

Fehleranzahl ähnlich sind. Es können für den Systemtest deshalb ähnliche Testverhalten

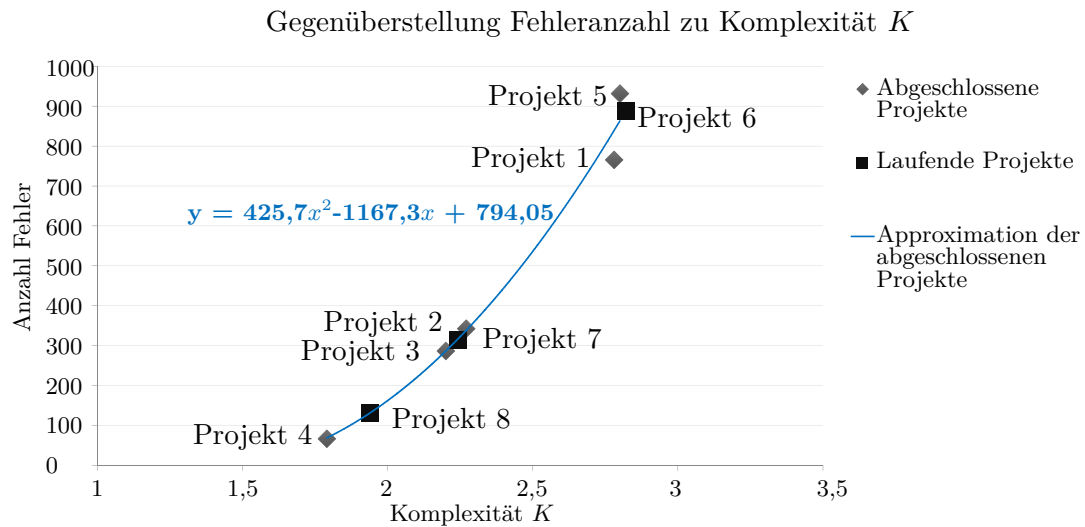


Abbildung 3.4. – Gegenüberstellung der Komplexität K und der zugehörigen Fehler inklusive einer Trendlinie für die abgeschlossenen Projekte, die einen approximierten Zusammenhang aufzeigt.

erwartet werden.

Es ist also möglich durch die Komplexitätsbewertung - zumindest in einem gewissen Maße - den Verlauf bzw. zumindest die Anzahl der zu erwartenden Systemtestfehler schon am Projektanfang zu prognostizieren. Durch weiteres Fortführen der Methode wird eine immer größere Datenbasis geschaffen, was die Vorhersage graduierlich weiter präzisiert.

Fazit

In diesem Kapitel wurde eine neue Methode gezeigt, wie Projekte anhand verschiedener Einflussgrößen charakterisiert, bewertet und ihre Komplexität bestimmt werden können. Zusätzlich wurde gezeigt, dass mithilfe dieser Bewertung eine grobe Prognose für die zu erwartenden Systemtestfehler abgegeben werden kann.

4. Testauswahl, -methoden und -priorisierung



Abbildung 4.1. – Teil 2 und 3 der in der Arbeit vorgeschlagenen Methodik (siehe auch Abbildung 1.3).

In diesem Kapitel wird die untersuchte Testauswahl und zugehörige Testmethoden dargestellt. Zunächst wird der erarbeitete Ansatz der Testauswahl in Abhängigkeit der Komplexitätsbewertung von Projekten dargelegt. In Kapitel 4.1 wird eine mögliche Auswahl der Testmethoden und Testfälle nach einem hier vorgeschlagenen Konzept erarbeitet. Es wird der Prozess des Übergangs von aktuellen Testmethoden, wie z. B. in ISO 26262 vorgegeben, hin zu effizienzsteigernden, ergänzenden Methoden aufgezeigt.

Anschließend werden drei Testmethoden ausformuliert, die ihren Einsatz im Systemtest integraler Fahrzeugsicherheitssysteme finden können.

Kapitel 4.2 stellt die Testmethode des simulativen, manöverbasierten Testens vor, die aktuell sporadisch eingesetzt wird. Nach einer Einleitung sollen die Anwendung sowie die Erstellung und Parametrierung von Testfällen dargelegt und somit der Einsatz der Methode für den Systemtest begründet werden. Anhand des Beispiels des reversiblen Gurtstraffers wird ein relevantes Szenario erarbeitet.

Darauf aufbauend werden die mit manöverbasiertem, simulativem Testen kombinierbaren Testmethoden der evolutionären Algorithmen und der statistischen Analyse von Testfällen erarbeitet. Der Einsatz dieser Methoden wird in dieser Arbeit neu eingeführt. Die Testmethoden werden jeweils an einem Beispiel erläutert. In Kapitel 4.3 wird nach einer Einleitung die prinzipielle Funktionsweise der statistischen Versuchsplanung sowie die Anwendung der Methode beispielhaft durch Kombination des Algorithmus mit manöverbasiertem Testen aus Kapitel 4.2 aufgezeigt. Kapitel 4.4 stellt die Testmethode der evolutionären Algorithmen anhand der Methode Differential Evolution dar. Nach einer Einleitung wird die prinzipielle Funktionsweise sowie ihre mathematische Abbildung und der Ablauf in Pseudo-Code erarbeitet. Die Anwendung der Methode wird abschließend beispielhaft durch Kombination des Algorithmus mit manöverbasiertem Testen anhand des bereits analysierten Beispiels aus Kapitel 4.2 und Kapitel 4.3 sowie einem vorher nicht analysierten Beispiels aufgezeigt.

Abschließend stellt Kapitel 4.5 zusammen, welche Möglichkeiten es zur Testfallpriorisierung gibt. Die Hauptziele der Priorisierung sind schnelleres Auffinden von Fehlern und schnelleres Schaffen von Vertrauen in das Gesamtsystem.

4.1. Testauswahl

Aktuelle Testmethoden für den Systemtest integraler Fahrzeugsicherheitssysteme stützen sich vorwiegend auf ISO 26262 (siehe auch Kapitel 2.3). Diese Methoden schaffen für den momentanen Stand der Technik auf Komponentenebene eine gute Grundlage für den Test. Da jedoch zukünftig die Systemkomplexität weiter steigen und wie bereits dargelegt in ein integrales Sicherheitssystem münden wird, ist der Einsatz effizienzsteigernder, ergänzender Testmethoden unerlässlich. Abbildung 4.2 zeigt einen neuen Ansatz für die Zusammenstellung und die Kombination solcher Methoden, die zukünftig Anwendung im Test solcher komplexer Sicherheitssysteme finden können.

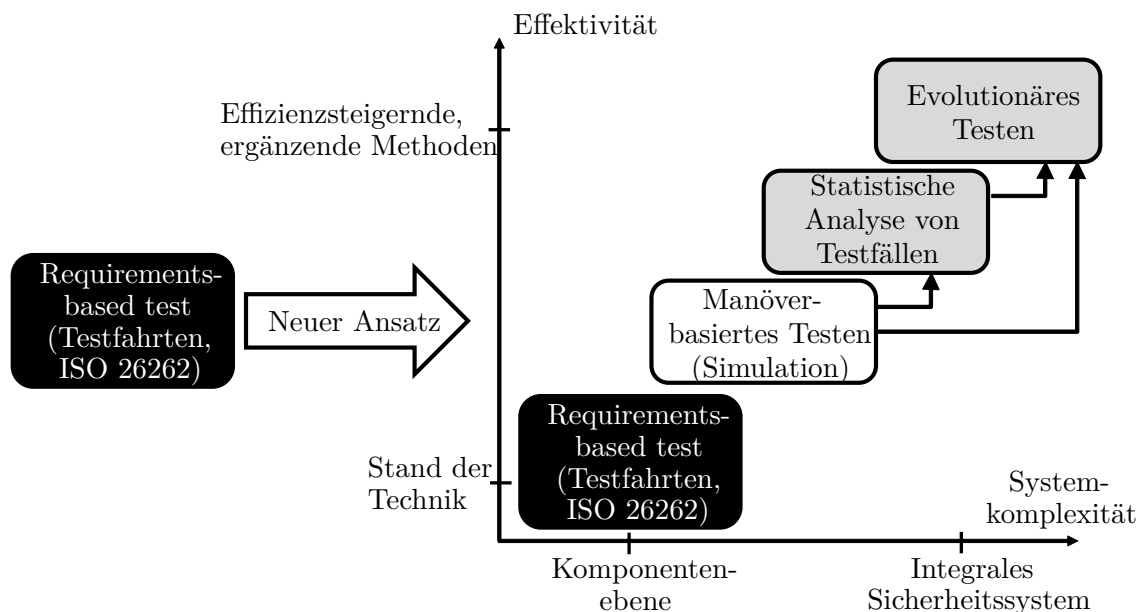


Abbildung 4.2. – Mithilfe des durch diese Arbeit vorgeschlagenen Ansatzes bei der Auswahl der Testmethoden kann ein Effizienzgewinn im Test von komplexen Systemen erreicht werden. Bei Bedarf - z. B. wenn die Komplexitätsbewertung des Projekts bzw. der Funktion dies ergibt - kann durch den Einsatz ergänzender, innovativer Testmethoden eine Testqualitätssteigerung und somit langfristig ein Wettbewerbsvorteil durch Produktqualitätssteigerung erreicht werden. Zukünftig können die hier vorgeschlagenen Testmethoden noch erweitert werden.

Die Systemkomplexität kann z. B. durch den in Kapitel 3 dargestellten Ansatz der Komplexitätszahl K , die anhand eines Expertenaudits mithilfe von gewichteten und bewerteten Einflussfaktoren ermittelt wird, abgebildet werden. Während für Projekte mit geringer Komplexität aktuelle Testmethoden wie das Requirements-based testing durch Einspeisung von aufgezeichneten Testfahrten durchaus ausreichend sind, sollten für Projekte mit hoher Komplexität weitere Maßnahmen ergriffen werden: durch effizienzsteigernde, ergänzende Methoden wie beispielsweise simulatives, manöverbasiertes Testen in Kombination mit statistischer Analyse von Testfällen und/oder evolutionären Algorithmen kann die Effektivität des Testens gesteigert und eine höhere Testtiefe erreicht werden. Diese zusätzlichen Methoden können auf bestimmte Testfälle und Szenarien angewandt werden und so zu einem verbesserten Systemverständnis und einer besseren Absicherung führen. Abbildung

4.3 zeigt schematisch den Testraum einer Funktion. Der rechnerische Testraum stellt eine fast unendliche Größe dar: durch Kombination jedes Wertes jeder Variable mit jedem anderen Wert jeder Variable entsteht schnell eine riesige Anzahl von Testfällen. Durch die stark begrenzte Testzeit und beschränkte Ressourcen können diese natürlich nicht alle ausgeführt werden. Durch Analyse des Wirkfeldes einer Funktion, also des relevanten Umfeldes und der möglichen Auswirkungen, kann der Testraum bereits stark eingeschränkt werden. Das Wirkfeld selbst kann durch Nutzung von Symmetrien und somit durch Ausschluss äquivalenter Fälle weiter eingeschränkt werden. Symmetrien können z. B. entstehen, weil eine Funktion Sensordaten aus der rechten und linken Fahrzeughälfte erhält und so das Ergebnis gleich ausfallen muss, wenn ein Signal rechts oder links eingespeist wird. Der verbleibende Testraum kann nun mit Testfällen gefüllt werden. Hierbei sollten besonders relevante und wichtige Testfälle wie zentrale Funktionalitäten, sicherheitskritische Szenarios und Grenzfälle genauer betrachtet werden. In diesen Bereichen ist eine genaue Kenntnis des Systemverhaltens notwendig, sodass eine vertiefte Analyse und genauere Betrachtung dieser Fälle unerlässlich ist. Durch Anwendung analytischer Methoden können vertiefte Aussagen über die Systemgüte getroffen werden. Nachfolgend sollen neben der Anwendung von ISO 26262 Methoden für den Systemtest vorgestellt werden, die eine tiefergehende Analyse von wichtigen Testfällen ermöglichen und ein tieferes Verständnis für das Systemverhalten liefern.

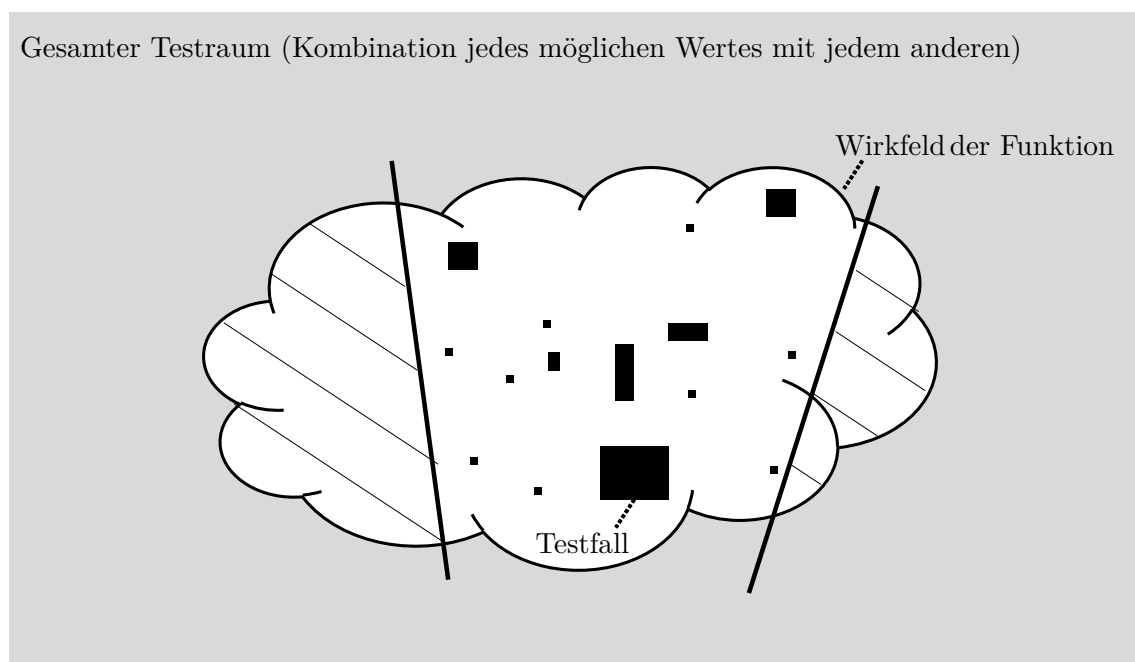


Abbildung 4.3. – Schematische Darstellung des Testraums (nach [116], [9]). Der gesamte Testraum stellt die Kombination jedes möglichen Wertes von internen und externen Variablen dar mit jeder anderen Variable dar. Die Anzahl dieser Testfälle geht schnell in die Milliarden. Durch Analyse des Wirkfeldes einer Funktion kann der Testraum schon stark eingeschränkt werden (Wolke). Das Wirkfeld kann durch Ausnützen von Symmetrien und Ausschluss nicht relevanter und äquivalenter Fälle weiter verkleinert werden. Im verbleibenden Testraum müssen dann entsprechende Testfälle und genauer zu untersuchende Teilräume ermittelt werden.

4.2. Testmethode manöverbasiertes Testen¹

4.2.1. Einleitung

Das Wort Manöver stammt aus dem Lateinischen *manu operari* und bedeutet wörtlich „mit der Hand bewerkstelligen“ oder freier übersetzt „Handhabung“. Diese Wortherkunft zeigt, dass allgemein bezogen auf ein System manöverbasierte Abläufe durch den Benutzer gesteuert werden. Im Bereich des Automobils kann ein Manöver als eine abgeschlossene Operation gesehen werden, die durch den Fahrer initiiert und durch das Fahrzeug ausgeführt wird. Die reale Verwendung eines Fahrzeugs entspricht viel eher dem zielorientierten Denken des Fahrers (z. B. „ich möchte das vorausfahrende Fahrzeug überholen“ oder „ich möchte 100 km/h fahren“) als das reine Setzen von Eingangsvariablen von Steuergeräten wie z. B. Radgeschwindigkeiten oder Beschleunigungen wie es häufig im Systemtest praktiziert wird [60]. Den Grad der Einflussnahme des Fahrers auf das Fahrzeug und indirekt die Anhängigkeit von der Umwelt zeigt Abbildung 4.4. Der Fahrer nimmt aufgrund der gewählten Navigation sowie der Führung und Stabilisierung des Fahrzeugs Einfluss auf die Längs- und Querdynamik des Vehikels. Dies beeinflusst wiederum den genauen Ort auf der Straßenoberfläche (z. B. den lokalen Reibwert), die Verkehrssituation im Straßennetzwerk und tangiert rückwirkend die Entscheidungen des Fahrers, wodurch eine geschlossene

¹Teile dieses Kapitels sind wörtlich in Veröffentlichung [101] enthalten

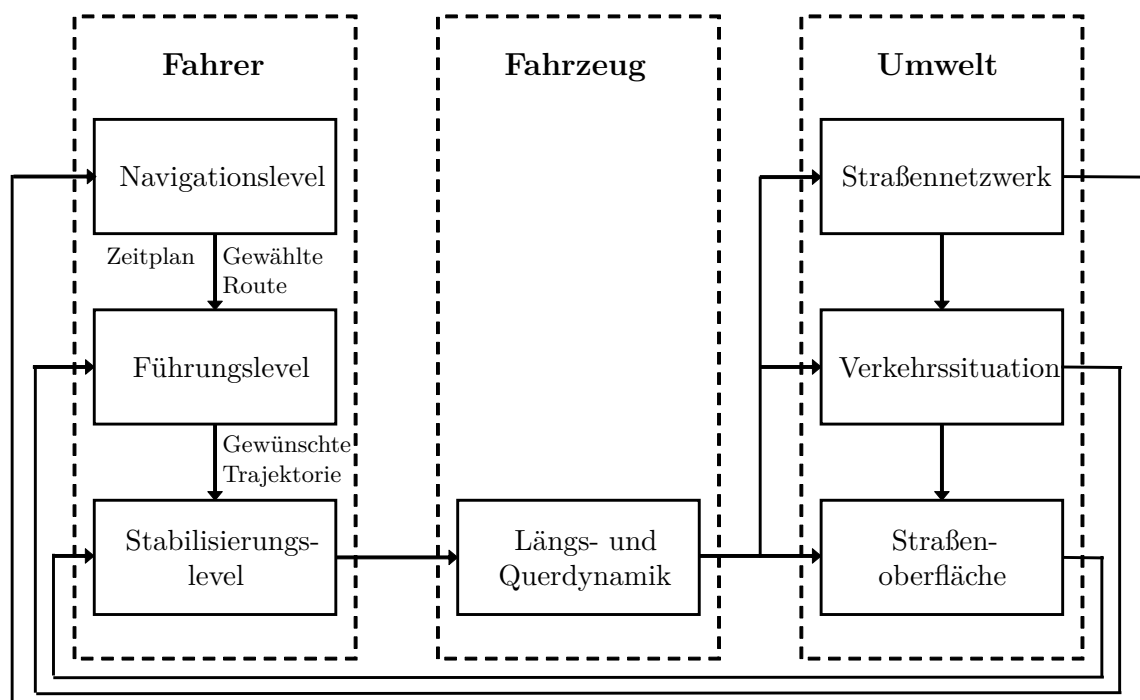


Abbildung 4.4. – Diese Abbildung zeigt eine closed-loop Drei-Level-Hierarchie von Fahrmanövern (nach [7], [101]). Deutlich werden die mögliche Einflussnahme des Fahrers auf das Fahrzeug und indirekt die Umwelt und die Rückwirkungen.

Wirkkette entsteht. Deutlich wird also, dass der aktuelle Fahrzustand eines Fahrzeugs mehr ist als das bloße Setzen von Eingangsvariablen.

4.2.2. Notwendigkeit und Anwendung

Für die Anwendung von manöverbasiertem Testen für den Systemtest von Fahrerassistenz- und Sicherheitssystemen im Fahrzeug ist eine Fahrdynamik- und Umfeldsimulation notwendig, die synthetische Daten erzeugt und die in das System eingespeist werden können. Natürlich ist auch die althergebrachte Methode des Einspeisens von real aufgezeichneten Testfahrten möglich, was aber aufgrund der fehlenden Flexibilität (keine spätere Modifikation der Manöver möglich) und der Notwendigkeit eines Testfahrzeugs als alleinige Methode inzwischen zu wenig ist. Das bloße Setzen von Variablenwerten über ein Testsystem und die Einspeisung über Bussysteme ist nur sehr begrenzt möglich, da die Steuergeräte Signale plausibilisieren, d. h. nicht physikalisch logische Sprünge werden erkannt und als Fehler registriert. So würde z. B. erkannt werden, wenn die Geschwindigkeit von 0 auf 50km/h springt. Des Weiteren sind nach [24] folgende Maßnahmen notwendig, um ein vernetztes, komplexes System abzusichern:

- Verkehrsflussimulation, manöver- und ereignisbasiertes Testen für neuartige Assistenzsystemen und Funktionalität
- Bereitstellung von Methoden, die Validierung unter der Berücksichtigung der Systeme Fahrer und Umwelt bereits in frühen Phasen der Produktentwicklung ermöglichen
- Einsatz von Umwelt- und Fahrermodellen aufgrund Sicherheit, Reproduzierbarkeit und Objektivierung

Fahrdynamik- und Umfeldsimulation gekoppelt mit MiL-, SiL- und HiL-Testsystemen ist eine sehr mächtige Lösung dieser Prämissen.

Abbildung 4.5 zeigt beispielhaft den Ablauf eines Auffahrunfalls und die zugehörigen Fahrerassistenz- und Sicherheitssysteme. Hier wird deutlich, dass der Unfallablauf mehr ist



Abbildung 4.5. – Aufgezeigt wird die manöverbasierte Einstufung der Phasen eines Unfalls inklusive Fahrerassistenz- und entsprechend auslösender Sicherheitssysteme (nach [101], [88] und [102]).

als das bloße Auftreffen des Ego-Fahrzeugs auf einen anderen Verkehrsteilnehmer. Wichtig ist nicht nur der Fahrzeugzustand zum Unfallzeitpunkt, der z. B. durch Geschwindigkeit, Winkel und Überdeckung gekennzeichnet ist, sondern auch die Vorgeschichte des Vehikels. Durch die vorhergehenden Phasen des normalen Fahrzustandes und der Pre-Crash-Phase werden intern in Systemen Variablen und Abläufe gespeichert, die zu anderen Systemreaktionen führen können, als wenn das Steuergerät nur den unmittelbaren Unfall durch Setzen von Eingangsvariablen erlebt.

Des Weiteren ist es wünschenswert, möglichst früh manöverbasierte Testfälle einspeisen zu können. Mithilfe der Simulation von verschiedensten Situationen und der Erzeugung synthetischer Daten schon im frühen Entwicklungs- und Teststadium, können viele Fehler, die sonst erst in der realen Testfahrt am Ende des Testzyklusses aufgetreten wären, frühzeitig aufgedeckt und beseitigt werden. Die Durchgängigkeit und Wiederverwendbarkeit der Testfälle ist wünschenswert, um im Model-in-the-Loop-Stadium (MiL) genauso testen zu können wie im späteren Software-in-the-Loop- (SiL) und abschließenden Hardware-in-the-Loop-Fall (HiL). Um nun solche Szenarien simulieren zu können, sind verschiedene Anforderungen an eine Simulationsumgebung zu erfüllen:

- Genaue, parametrierbare Modelle, die realistisch die dynamischen Interaktionen zwischen Fahrzeug, eventuell Anhänger, Fahrer, Straße und Verkehr abbilden
- Definition und Parametrierung von manöver- und ereignisbasierten Testfällen
- Möglichkeit des durchgängigen Testens von Modell-, Software- und Systemstufe (MiL, SiL, HiL) und der Wiederverwendung von Testfällen über den Lebenszyklus des Systems hinweg

Eine mögliche Auswahl eines solchen Simulationstools stellt die Software CarMaker[®] der IPG Automotive GmbH dar, die zur Simulation aller in dieser Arbeit nachfolgender manöverbasierter Testfälle verwendet wurde. In Abbildung 4.6 werden einige wenige Beispiele für das manöver- und eventbasierte Testen mit CarMaker[®] aufgezeigt. Deutlich wird, dass neben dem Ego-Fahrzeug und dem zugehörigen Manöver, das durch den Fahrer ausgeführt wird, auch die gesamte Umwelt inklusive anderer Verkehrsteilnehmer definiert werden kann. Somit steht dem Tester eine mächtige Simulationsumgebung zur Verfügung und es können alle denkbaren Szenarien simuliert und variiert werden. Abbildung 4.7 zeigt das Prinzip des manöverbasierten Testens mithilfe einer Simulationsumgebung im HiL-Fall des Systemtests von Steuergeräten. Die an ein Testsystem angebundene Fahrdynamik- und Umfeldsimulation speist die manöverbasierten Testdaten eines Szenarios ein. Im Testsystem werden die Daten in die für das Steuergerät notwendigen Dateiformate und Protokolle wie z. B. FlexRay oder CAN überführt und versorgen das Steuergerät über entsprechende Schnittstellen mit den Eingabedaten. Zusätzlich wirkt das Steuergerät über Brems- und Lenkeingriffe auf das Fahrzeugverhalten zurück. Das Steuergerät erlebt den Ablauf somit genauso, als wäre es in einem Fahrzeug verbaut und würde das entsprechende Szenario erfahren. Es handelt sich somit um eine Teststufe, die dem realen Einsatz des Steuergeräts so nahe kommt wie es über Simulation möglich ist.

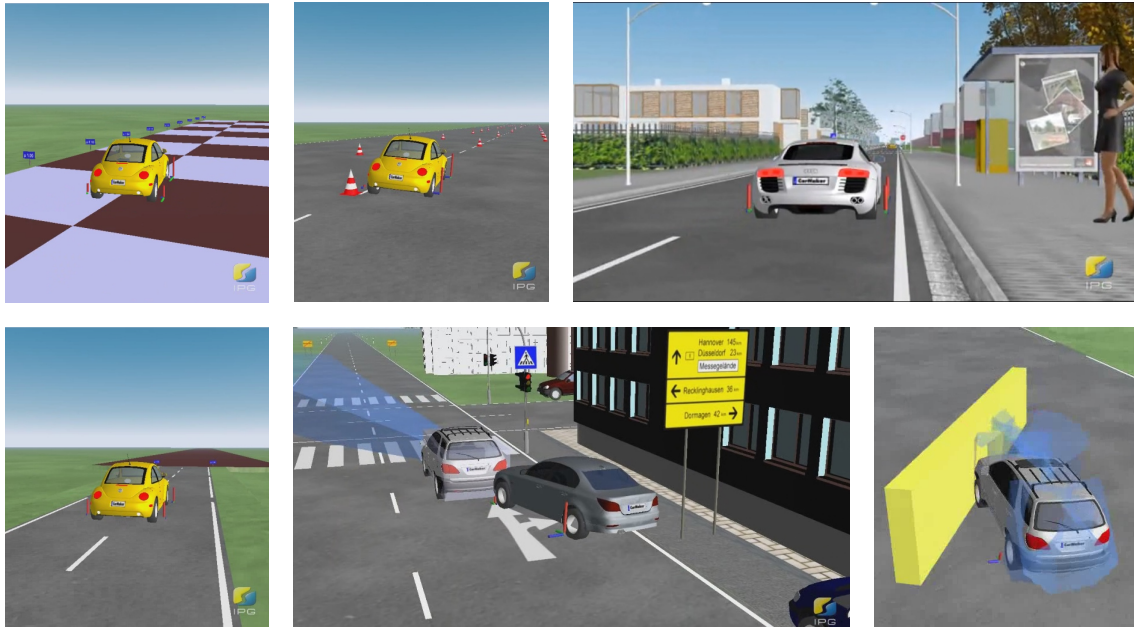


Abbildung 4.6. – Beispiele für manöver- und eventbasiertes Testen mit CarMaker®

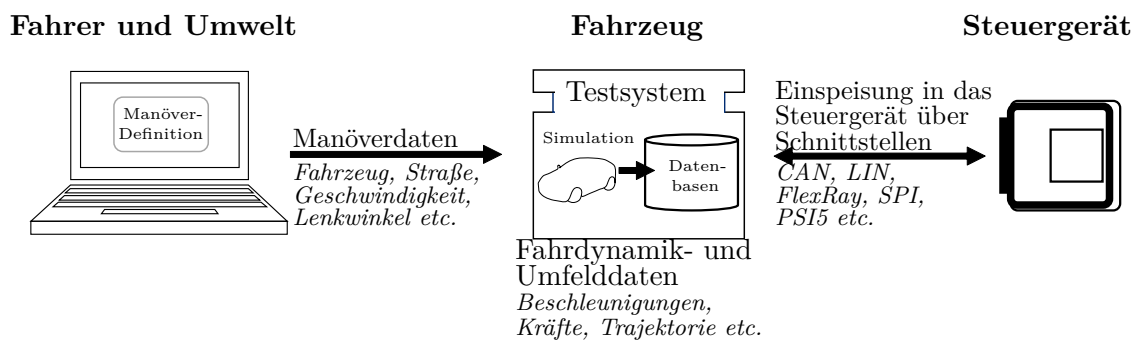


Abbildung 4.7. – Das Prinzip des manöverbasierten Testens im Bereich des Systemtests von Steuergeräten umfasst die Fahrdynamik- und Umfelddaten von Szenarien sowie die Fusion von verschiedenen Datenbanken und den Simulationsdaten in einem Testsystem, das wiederum das Steuergerät mit den benötigten Eingabedaten versorgt [101].

4.2.3. Beispiel

Als Beispieldestobjekt bzw. -funktion soll nachfolgend die Auslösung eines reversiblen Gurtstraffers dienen. Der reversible Gurtstraffer stellt ein Sicherheitssystem dar, das anhand von Fahrdynamik- und Umfelddaten ausgelöst wird. Abbildung 4.8 zeigt den Aufbau und das Einsatzgebiet des Gurtstraffers. Im Gegensatz zum pyrotechnischen Gurtstraffer, der nach einem Aufprall kurz vor oder gemeinsam mit dem Airbag gezündet wird und auch nicht wiederverwendet werden kann, findet der reversible Gurtstraffer seinen Einsatz in der Pre-Crash-Phase anhand von Fahrdynamik- oder vorausschauenden Umfelddaten. Er ist somit ein Beispiel für das Zusammenwachsen der passiven und aktiven Sicherheit: ein eigentlich passives Sicherheitssystem (es vermeidet keine Unfälle, sondern mindert Verletzungen) wird

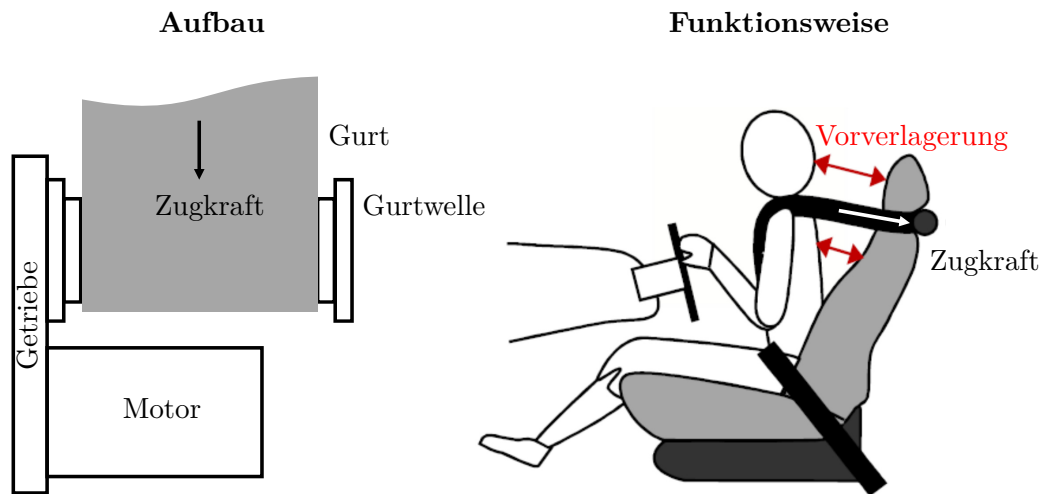


Abbildung 4.8. – Der prinzipielle Aufbau eines reversiblen Gurtstraffers zeigt die Komponenten Motor und Getriebe, die über eine Gurtwelle den Gurt aufrollen und somit eine Zugkraft auf den Insassen ausüben. Die prinzipielle Funktionsweise zeigt, dass die Vorverlagerung des Insassen vermindert werden kann (nach [77], [82] und [101]).

vor dem Unfalleintritt, also in der Domäne der aktiven Sicherheitssysteme, gezündet. Da er seine Energie nicht über Pyrotechnik, sondern einen Elektromotor erhält, ist er mehrfach aktivierbar, die auf den Insassen wirkenden Kräfte jedoch auch deutlich geringer. Das Ziel der Auslösung eines reversiblen Gurtstraffer ist es, den Insassen in einer kritischen Situation auf einen möglichen Unfall vorzubereiten und ihn über den aktuellen Fahrzustand zu warnen [74]. Hierfür wird die Gurtlose reduziert und der Fahrer in den Sitz gezogen. Die Unterschiede zwischen dem pyrotechnischen und dem reversiblen Gurtstraffer werden in Tabelle 4.1 noch einmal zusammengefasst.

Reversibler Gurtstraffer	Pyrotechnischer Gurtstraffer
elektromechanischer Aktor	pyrotechnische Zündkapsel
ca. 250N Zugkraft in 120ms [73]	ca. 2-4kN Zugkraft in 12-15ms [70]
Auslösung in Pre-Crash-Phase	Auslösung in In-Crash-Phase
mehrmalige Auslösung möglich	einmalige Auslösung, danach Austausch in Werkstatt
Auslösung auch ohne nachfolgenden Unfall möglich	Auslösung nur bei Unfall

Tabelle 4.1. – Vergleich reversibler und pyrotechnischer Gurtstraffer

Eine Auslösesituation des reversiblen Gurtstraffers ist eine Gefahrenbremsung, die durch eine starke Erhöhung des Bremspedaldrucks aus hohen Geschwindigkeiten auf Eis detektiert wird [130]. Dies bedeutet, dass das Fahrzeug kaum verzögert, obwohl der Fahrer starken

Druck auf das Bremspedal ausübt. Die für die Auslösung des Algorithmus verwendeten Signale könnten also Fahrzeuggeschwindigkeit, der Bremsdruck und die resultierende Längsbeschleunigung des Fahrzeugs sein. Mögliche quantitative Anforderungen an die Auslösung dieses Falls werden in Tabelle 4.2 dargestellt. Es müssen also drei Kriterien

Auslösekriterium	Wert
Geschwindigkeit	$\geq 50\text{km/h}$
\wedge Bremsdruck	$\geq 90\text{bar}$
\wedge longitudinale Verzögerung	$[-1,9\text{m/s}^2, 0\text{m/s}^2]$

Tabelle 4.2. – Mögliche Auslösekriterien für den reversiblen Gurtstraffer bei der Gefahrenbremsung (siehe [130], [101]).

für die dargestellten Variablen erfüllt sein, damit eine Auslösung erfolgt. Wenn eine der Anforderungen nicht mehr erfüllt ist, ist die Gefahrensituation grundsätzlich vorbei, jedoch bleibt der Gurtstraffer häufig der Gurtstraffer noch eine gewisse Zeit aktiviert bzw. gestrafft. Das Auslösekriterium selbst ist aber dann nicht mehr erfüllt.

Zum Aufbau des Testfalls in der Fahrdynamik- und Umfeldsimulation müssen in diesem Fall verschiedene Maßnahmen getroffen werden:

- Auswahl oder Definition des Fahrzeugs
- Definition der Straße
- Definition des Ablaufs des Manövers
- Definition der Variablen bzw. Eingabe

Als Beispiel für den Aufbau dieses manöverbasierten Testfalls in CarMaker[®] soll auch in nachfolgenden Kapiteln Tabelle 4.3 dienen. Die Eingabevariablen ergeben sich aus den Anforderungen: die Anfahrtsgeschwindigkeit, die Betätigung der Bremse und der Reibwert sind entscheidend für die Ausgabevariable Verzögerung, die ja die Auslösung des reversiblen Gurtstraffers mitbestimmt. Ohne Vorkenntnisse über die Ergebnisse des Ablaufs des Testfalls können die Werte für die Eingabeparameter nur erraten werden, um eine gewünschte longitudinale Verzögerung zu erhalten. Dieses Problem soll nachfolgend durch statistische Analyse und anschließende Optimierung des Testfalls hin zu gewünschtem Verhalten gelöst werden. Das Ziel des Beispiels soll es sein, einen Grenzfall zu parametrieren, bei dem der reversible Gurtstraffer gerade auslösen muss. Der Wert der longitudinalen Verzögerung soll $-1,9\text{m/s}^2$ betragen, damit der Grenzbereich der Funktionskalibrierung getestet werden kann. Auch die anderen beiden Anforderungen hinsichtlich Geschwindigkeit und Bremsdruck sollen erfüllt sein. Dies ist jedoch trivial zu ermitteln, da die beiden Variablen als direkte Eingabe in die Simulation vorgegeben werden können und nicht wie die Verzögerung ein Resultat des Manövers sind. Nach Analyse und Optimierung sollen die Fahrdynamikdaten aus der Simulation dann zur Demonstration und Erprobung in das Testobjekt eingespeist werden.

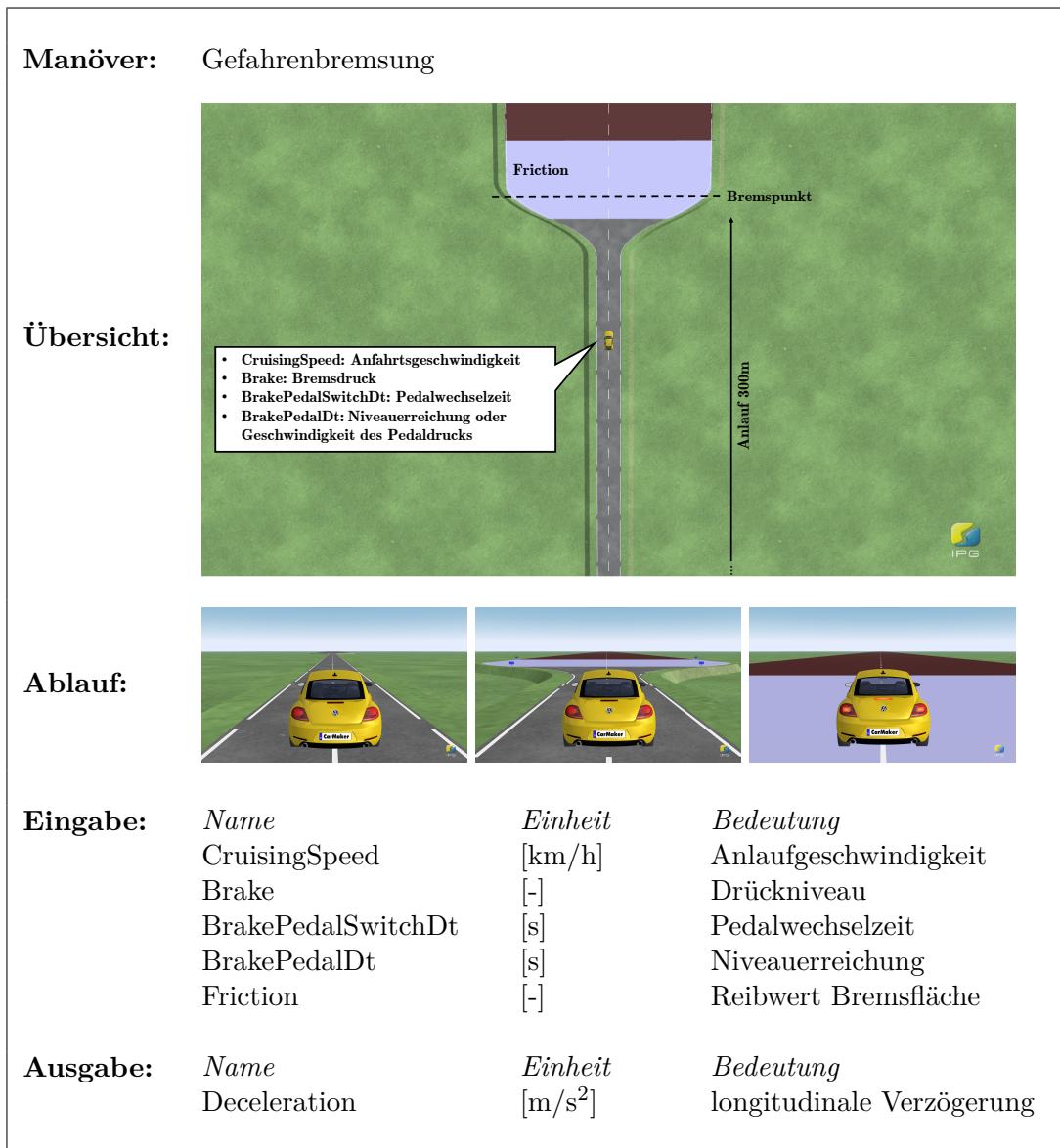


Tabelle 4.3. – Aufbau Beispiel manöverbasierter Testfall: Es wurde ein Bremsmanöver ausgewählt, das einen longitudinal kritischen Fall darstellt. Nach der Beschleunigung auf CruisingSpeed innerhalb eines Anlaufs, wird auf einer Fläche gebremst, die den Reibwert Friction hat. Hierbei werden für den Bremsvorgang selbst die Variablen Brake, BrakePedalSwitchDt und BrakePedalDt angegeben, die die Stärke des Bremsens, die Reaktionszeit bzw. Pedalwechselzeit und die Geschwindigkeit des Bremsens angeben. Das Fahrzeug bremst mit der longitudinalen Verzögerung Deceleration. Kritisch ist hierbei, dass der Fahrer stark bremsen möchte, aber die Verzögerung des Fahrzeugs aufgrund des niedrigen Reibwerts der Straße durch Eis, Schnee oder Aquaplaning nur sehr gering ausfällt [130], [101].

4.2.4. Fazit

Die Vorteile des manöverbasierten Testens können folgendermaßen zusammengefasst werden:

- Schon weit bevor ein Testfahrzeug bzw. real aufgezeichnete Fahrmanöver zur Verfügung stehen, können verschiedenste Situationen eingespeist werden
- Die Simulation und mögliche Automatisierung ist erheblich kostengünstiger als reale Testfahrten.
- Jegliche denkbare Situation kann simuliert werden; auch Unfallhergänge können durchgängig simuliert werden, die sonst in Testfahrten und Crashtests - wenn überhaupt - nur separat getestet werden können
- Die Wiederverwendbarkeit und Durchgängigkeit der Simulationsumgebung über MiL, SiL und HiL hinweg ist gewährleistet und erbringt sowohl Zeitersparnis als auch Vergleichbarkeit der Ergebnisse über den gesamten Lebenszyklus des Systems
- Die Reproduzierbarkeit der Ergebnisse ist uneingeschränkt gewährleistet.
- Simulationen können einfach und schnell reproduziert und variiert werden.
- Der Einsatz bzw. die Kombination mit verschiedenen Testmethoden und -metriken ist zur Erstellung und Auswertung der Szenarien möglich.

Manöverbasiertes, simulatives Testen ist also eine Methode um zusätzlich zu signalbasierter Einspeisung und der Einspeisung von aufgezeichneten Testfahrten flexibel, zeit- und kostenarm testen zu können. Durch die Erweiterung um die Einspeisung synthetischer Daten wird die Testabdeckung erhöht. Das Testobjekt wird genauso gespeist als wäre es im Fahrzeug und kann deshalb früh im Testprozess sehr realitätsnah getestet werden.

4.3. Testmethode statistische Analyse von Testfällen

4.3.1. Einleitung

Seit den 1980er Jahren wird verstärkt an Methoden geforscht, die die Planung von Experimenten und Versuchen strukturiert und bestmögliche Informationen über ein System liefert [40]. Vor allem die Arbeit des Japaners Genichi Taguchi stellt eine wichtige Grundlage dar, um Produkte, Prozesse und Systeme robust zu entwerfen und zu verbessern (siehe [131]). Die sogenannte Statistische Versuchsplanung oder Design of Experiments (DoE) ist ein Werkzeug, das bei der Erstellung von Versuchsreihen eine wichtige Rolle spielt. Nach [67] ist die Statistische Versuchsplanung definiert zu:

Definition 4.1 (Statistische Versuchsplanung)

Statistische Versuchsplanung oder Design of Experiments (DoE) beschreibt ein Vorgehen zur versuchsbasierten, systematischen Analyse von Systemverhalten. Durch Kombination von Faktorstufen, d. h. Wertstufen von relevanten Variablen, und der Ausführung von Einzelversuchen können Zusammenhänge zwischen Faktoren und Zielgrößen quantitativ beschrieben, Maßnahmen definiert und Verbesserungen erzielt werden.

Nach [67] sind Versuche, Statistik und Versuchsplanung aus verschiedenen Gründen wichtig:

Gründe für den Einsatz von Versuchen:

- Steigender Funktionsumfang von Produkten
- Gewährleistung der Erfüllung von Anforderungen von Kunden
- Kostensenkung
- Verkürzung von Entwicklungs- und Produktionszeiten

Gründe für den Einsatz von Statistik:

- Trotz Sorgfalt erhält man bei Wiederholung eines Versuchs meist nicht genau dieselben Ergebnisse: die Versuchsergebnisse streuen aufgrund zufälliger Unterschiede bei Ausgangsbedingungen
- Absicherung gegen Fehlentscheidungen: Zwei Versuchsergebnisse werden dann als unterschiedlich akzeptiert, wenn der Unterschied so groß ist, dass er nur mit ausreichend kleiner Wahrscheinlichkeit zufällig auftritt, obwohl in Wahrheit keiner besteht
- Mittelwerte können ermittelt werden
- Ermittlung quantitativer Unterschiede
- Zusammenhänge sind nicht klar, aber nicht alle Kombinationsmöglichkeiten können ausgeführt werden
- Analyse von komplexen Zusammenhängen

Gründe für den Einsatz von Versuchsplanung:

- Festlegung des richtigen Versuchsumfanges: zu viele Versuche kosten unnötig Zeit und Geld, zu wenige führen zu nicht erkannten relevanten Unterschieden
- Mehr Informationen durch Versuchsplanung (mehrere Faktoren werden auf einmal verändert und kombiniert) als bei One-factor-at-a-time (ein Faktor wird pro Versuch verändert), obwohl weniger Versuche benötigt werden
- Systematische Vorgehensweise, die erzwingt, dass man sich genau mit der Festlegung der Versuchsobjekte beschäftigt: „Ausprobieren“ wird vermieden
- Aufwand kann von Anfang an gut abgeschätzt werden
- Systematische Darstellung der Ergebnisse erlaubt bessere Rückschlüsse und Ableitung von Maßnahmen
- Erleichterung der Dokumentation der Ergebnisse und Übertragung der Erfahrungen an zukünftige Entwicklungen und Tests

Wie in Abbildung 4.9 dargestellt, hilft DoE bei der Optimierung von drei Teilen des Wissensakquiseprozesses. Die Auswahl der Experimentmethode, die Ergebnisanalyse und die progressive Wissensakquise kann durch statistische Versuchsplanung gewährleistet werden. Nachfolgend soll nun zunächst die prinzipielle Vorgehensweise bei der Statistischen Versuchsplanung erläutert und diese dann anhand eines Beispiels aus dem simulativen, manöverbasierten Testen erprobt werden. Die Untersuchung von manöverbasierten Testfällen birgt den Vorteil, dass die gewonnenen Erkenntnisse über die Zusammenhänge und Einflüsse eines Testfalls wiederum für die Definition und Planung von evolutionären Testfällen verwendet werden können.

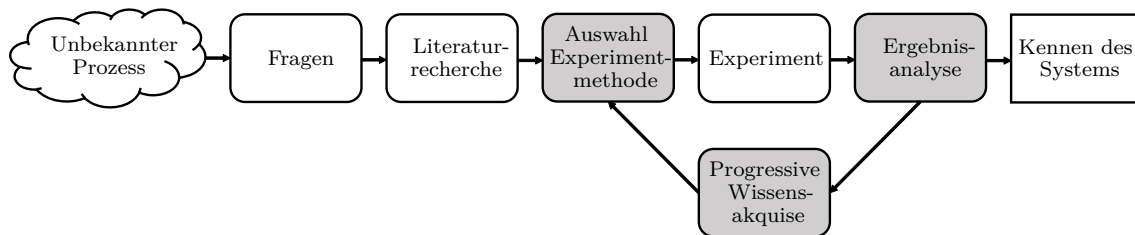


Abbildung 4.9. – DoE optimiert drei Teile des Wissensakquiseprozesses nach [40]

4.3.2. Prinzipielle Vorgehensweise

Die prinzipielle Vorgehensweise der statistischen Versuchsplanung ist Abbildung 4.10 gezeigt. Für detaillierte Informationen und Ausformulierungen sei auf einschlägige Fachliteratur

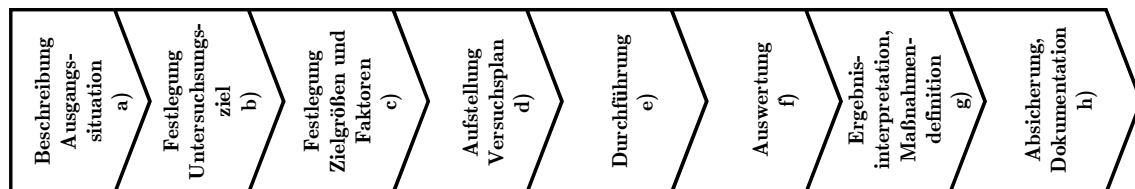


Abbildung 4.10. – Dargestellt wird die prinzipielle Vorgehensweise des Design of Experiments nach [67]

verwiesen (siehe neben [40] und [68] z. B. auch [3], [15], [66], [81] oder [115]). Die prinzipiellen Einzelschritte der Statistischen Versuchsplanung sollen nun kurz erläutert werden (nach [68]):

a) Beschreibung Ausgangssituation:

Zur Vorbereitung der Untersuchung ist wichtig, das Umfeld zu definieren. Folgende Fragen können dabei eine Rolle spielen: Für wen wird die Untersuchung gemacht? Was ist die Zielsetzung? Welches Problem soll gelöst werden? Wie viel Zeit und Geld steht zur Verfügung?

b) Festlegung Untersuchungsziel:

Verschiedene Möglichkeiten zur Festlegung des Untersuchungsziels sind möglich. Grundsätzlich gibt es die folgenden Alternativen:

- Optimale Lage des Mittelwerts: Das Ergebnis soll einen bestimmten Wert annehmen und somit ist das Ergebnis selbst eine geeignete Zielgröße. Aus statistischen Gründen ist dies meist das arithmetische Mittel. Die Untersuchung soll ergeben, wie der Mittelwert von den Faktoren abhängt.
- Reduzierung der Streuung/Robustheit: Vor allem bei der Fertigung ist oft eher die Streuung des Ergebnisses problematisch als dessen Mittelwert. Die Streuung soll dann vermindert werden und stellt die Zielgröße dar.
- Erkennen der wichtigsten Störgrößen: Die Identifikation und Überwachung von Störgrößen ist ein wichtiges Maß zur Reduzierung von Streuungen.
- Experimentieren und Lernen: Durch systematisches Verändern von Parametern können Verbesserungsmöglichkeiten erkannt werden.
- Funktion und Zuverlässigkeit nachweisen: Das Testobjekt soll beweisen, dass es unter allen spezifizierten Einsatzbedingungen bestimmungsgemäß funktioniert und ausreichende Zuverlässigkeit erreicht.

c) Festlegung Zielgrößen und Faktoren:

- Zielgrößen: Beschreiben das Ergebnis eines Versuchs. Wichtig bei der Auswahl der Zielgrößen sind die Relevanz, die Quantifizierung, die Vollständigkeit und Verschiedenheit.
- Einflussgrößen: Beeinflussen die Versuchsergebnisse und somit die Zielgrößen. Essentiell ist, keine wichtige Einflussgröße zu vergessen. In bewährter Weise sollten zunächst alle möglichen Einflussgrößen gesammelt werden und anschließend die wichtigsten ausgewählt werden, um eine handhabbare Anzahl zu erreichen, die dann die Faktoren darstellen. Nicht ausgewählte Einflussgrößen werden konstant gehalten.
- Faktorstufen: Für die aus den Einflussgrößen ausgewählten Faktoren werden Stufen festgelegt (quantitativ oder qualitativ), die in den Versuchsplan übernommen werden. Im Normalfall werden die Faktoren 2- oder 3-stufig gewählt (z. B. niedrig, mittel, hoch). Wichtig ist, den Abstand der Stufen weder zu klein noch zu groß zu wählen, um die Verdeckung des Effekts von Zufallsstreuungen und eine schlechte Beschreibung des Zusammenhangs zu vermeiden.

d) Aufstellung Versuchsplan (siehe Abbildung 4.11):

- Festlegung Faktorstufenkombinationen: Festlegung, welche Kombinationen der Faktorstufen ausgeführt werden sollen. Eine vollständige Kombination aller Faktorstufen heißt vollständig faktorieller Versuchsplan oder full factorial design. Eine teilweise Kombination der Faktorstufen heißt fraktioneller faktorieller Versuchsplan, Teilfaktorplan oder fractional factorial design.

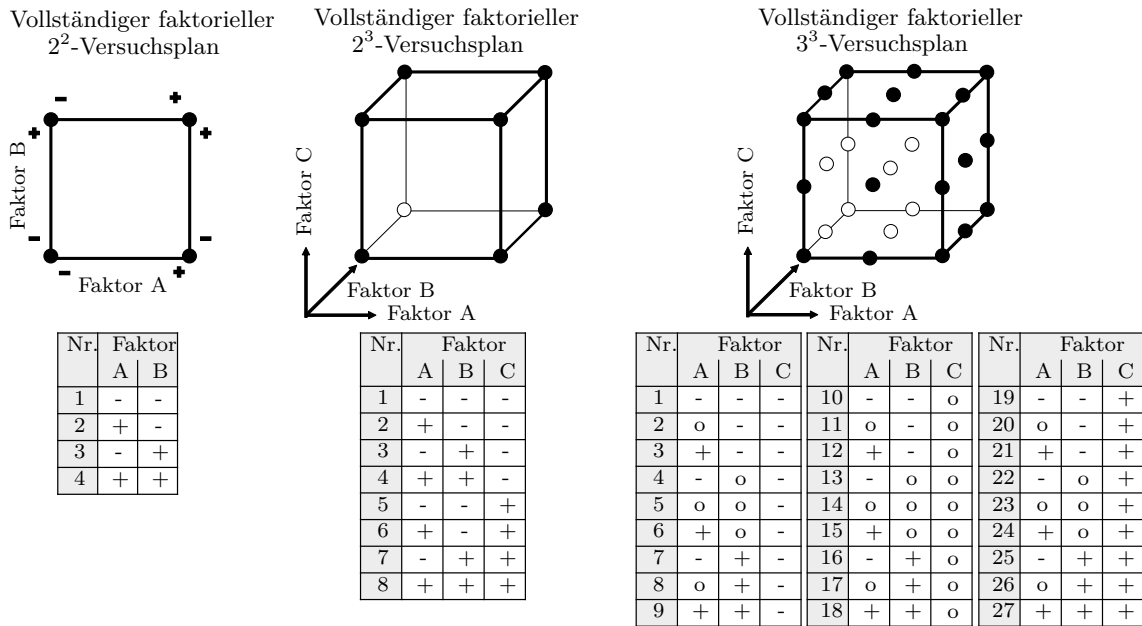


Abbildung 4.11. – Dargestellt werden verschiedene Arten von Versuchsplänen nach [68]. Die einfachste Stufe, ein vollständiger faktorieller 2²-Versuchsplan besitzt zwei Faktoren A und B mit je zwei Faktorstufen + und -, was in vier Versuchen resultiert. Ein vollständiger faktorieller 2³-Versuchsplan hingegen hat drei Faktoren A, B und C auf je zwei Stufen + und -, was 8 Versuche ergibt. Ganz rechts ist ein vollständiger faktorieller 3³-Versuchsplan dargestellt, der drei Faktoren A, B und C mit drei Stufen +, o und - (meist hoch, mittel, niedrig), wodurch 27 Versuche entstehen.

- Anzahl der Realisierungen: Je nach Streuung der Ergebnisse und gewünschter Untersuchungsgenauigkeit der Effekte der Faktoren sind mehr oder weniger Einzelversuche für jede Kombination erforderlich.
- Blockbildung: Zur Minimierung der Zufallsstreuung können Einzelversuche in Gruppen eingeteilt werden, damit die zufälligen Unterschiede möglichst klein sind und die Faktorstufenkombinationen möglichst gleich häufig auftreten.
- Randomisierung: Die Einzelversuche jedes Blockes sollten in zufälliger Reihenfolge ausgeführt werden, damit die Ergebnisse nicht verfälscht werden.
- Aufwandsschätzung: Das Ergebnis der Schätzung sollte in einem angemessenen Verhältnis zum erhofften Ergebnis stehen.

e) Durchführung:

- Vorbereitung: Wichtig sind Ressourcenplanung, Überprüfung der Geräte und Verfahren, Festlegung des Versuchsablaufs, Durchführung eines Pilotversuchs
- Ausführung: Sorgfältige Überwachung der Durchführung, Messung, Übertragung und Speicherung der Ergebnisse ist unerlässlich.

f) Auswertung:

Mögliche statistische Auswerteverfahren der Versuchsergebnisse werden in Kapitel 4.3.3 anhand eines Beispiels erläutert.

g) Ergebnisinterpretation, Maßnahmendefinition:

Ist die Wechselwirkung zwischen zwei Faktoren signifikant, so ist der Effekt eines Faktors vom Wert des anderen Faktors abhängig. Wichtig ist die technische Plausibilität der Ergebnisse. Aus quantitativen und technischen Betrachtungen können Maßnahmen für Verbesserungen resultieren.

h) Absicherung, Dokumentation:

Die getroffenen Maßnahmen werden bei Bedarf durch Bestätigungsversuche abgesichert. Die Ergebnisse müssen dokumentiert werden.

Die Anwendung und Auswertung dieser Methode soll nun anhand des Beispiels aus dem manöverbasierten Testen genauer erläutert und aufgezeigt werden.

4.3.3. Anwendung

Die statistische Versuchsplanung ist für die Analyse von manöverbasierten Szenarien sehr gut geeignet. Aufgrund der schnellen Simulationszeit können auch große Versuchspläne in kurzer Zeit und komfortabel durchgeführt werden. Anhand eines Beispiels, das bereits aus dem vorhergehenden Kapitel 4.2.3 bekannt ist, soll nun die vorhergehende Vorgehensweise im Praxisfall dargestellt werden.

a) Beschreibung Ausgangssituation:

Es liegt ein manöverbasierter Testfall in der Simulationsumgebung CarMaker® vor. Dieser Testfall kann beliebig parametrisiert werden. Um diesen Testfall entsprechend von Anforderungen als Testfall in ein Steuergerät einzuspeisen, müssen gewisse Eigenschaften (z. B. bestimmte Werte für Variablen, die sich aus den Anforderungen ergeben) erfüllt werden. Diese genauen Charakteristika sind jedoch händisch schwer herbeizuführen (nur die Eingangsgrößen für die Simulation wie Geschwindigkeit, Lenkwinkel etc. sind vorgebar; Beschleunigungen, Kräfte etc. entstehen aus der Fahrdynamiksimulation heraus als Ausgangsgrößen, die nur indirekt über die Eingangsgrößen beeinflusst werden können), weshalb eine Analyse des Testfalls notwendig ist.

b) Festlegung Untersuchungsziel:

Das Ziel der Untersuchung soll sein, Wechselwirkungen zwischen den Einflussgrößen (Faktoren) aufzuzeigen und den Einfluss der Faktoren auf eine Zielgröße darzustellen. Der Testfall soll die Eigenschaft erfüllen, eine longitudinale Verzögerung von $-1,9\text{m/s}^2$ zu haben. Der Punkt des Experimentierens und Lernens spielt ebenfalls eine Rolle. Aufgrund der optimalen Simulation sind Streuungen und Störgrößen in diesem Fall keine Gesichtspunkte. Auch die Zuverlässigkeit des Testobjekts steht hier nicht im Fokus.

c) Festlegung Zielgrößen und Faktoren:

- *Zielgröße:*
Deceleration Es ist eine möglichst genaue Verzögerung von $-1,9\text{m/s}^2$ während des Bremsens gewünscht (Gefahrenbremsung auf Eis, Schnee oder Aquaplaning)
- *Faktoren:*
CruisingSpeed Anlaufgeschwindigkeit
Brake Drückniveau des Bremspedals
Friction Reibwert Bremsfläche
BrakeSwitchPedalDt Zeitpunkt, an dem das Drückniveau aufgegeben wird (Pedalwechsel bzw. Reaktionszeit)
BrakePedalDt Zeitdauer, die bis zur Erreichung des Niveaus benötigt wird
- *Faktorstufen:*
Aufgrund des breiten Wertebereichs der Faktoren wird ein 3-stufiger Versuchsplan ausgewählt.
CruisingSpeed – : 25km/h ○ : 50km/h + : 75km/h
Brake – : 0,5 ○ : 0,75 + : 1,0
Friction – : 0,1 ○ : 0,3 + : 0,5
BrakePedalSwitchDt – : 0,1s ○ : 0,3s + : 0,5s
BrakePedalDt – : 0,1s ○ : 0,3s + : 0,5s

d) Aufstellung Versuchsplan:

Der Versuchsplan wird anhand der Faktorstufen aufgestellt. Der vollständige faktorielle 3^5 -Versuchsplan zum manöverbasierten Testen ist in Tabelle 4.4 aufgeführt. Jede Faktorstufe wird einmal mit jeder anderen Faktorstufe kombiniert, was insgesamt $3^5 = 243$ Einzelversuche ergibt.

In diesem Fall reicht eine Realisierung, da eine ideale Simulation vorliegt, die bei

Versuchs- nummer	CrSpeed [km/h]	Brake [-]	Friction [-]	BrPSwDt [s]	BrakePDt [s]
1	– : 25,0	– : 0,5	– : 0,1	– : 0,1	– : 0,1
2	○ : 50,0	– : 0,5	– : 0,1	– : 0,1	– : 0,1
3	+ : 75,0	– : 0,5	– : 0,1	– : 0,1	– : 0,1
4	– : 25,0	○ : 0,75	– : 0,1	– : 0,1	– : 0,1
5	○ : 50,0	○ : 0,75	– : 0,1	– : 0,1	– : 0,1
6	+ : 75,0	○ : 0,75	– : 0,1	– : 0,1	– : 0,1
7	– : 25,0	+ : 1,0	– : 0,1	– : 0,1	– : 0,1
8	○ : 50,0	+ : 1,0	– : 0,1	– : 0,1	– : 0,1
9	+ : 75,0	+ : 1,0	– : 0,1	– : 0,1	– : 0,1
10	– : 25,0	– : 0,8	○ : 0,3	– : 0,1	– : 0,1
...
241	– : 25,0	+ : 1,0	+ : 0,5	+ : 0,5	+ : 0,5
242	○ : 50,0	+ : 1,0	+ : 0,5	+ : 0,5	+ : 0,5
243	+ : 75,0	+ : 1,0	+ : 0,5	+ : 0,5	+ : 0,5

Tabelle 4.4. – Vollständiger faktorieller 3^5 -Versuchsplan

denselben Eingangsgrößen bei jedem Versuch dieselben Ausgangsgrößen erzeugt, Reproduzierbarkeit ist also gegeben. Auch Blockbildung und Randomisierung sind aufgrund der vorhandenen Nachvollziehbarkeit in diesem Fall nicht notwendig.

Die Aufwandsschätzung zur Durchführung der Testreihe ist abhängig von der gewählten Anzahl der Versuche und von der Simulationsdauer der einzelnen Testfälle. In diesem Fall werden 243 Einzelversuche mit je circa 5s Simulationszeit ausgeführt, was sich insgesamt zu ungefähr 20min aufsummiert. Die Vorbereitungszeit für die Versuchsreihe hängt davon ab, ob der Testfall bereits vorhanden ist oder noch in der Simulationsumgebung aufgebaut werden muss. Als grober Richtwert für die Vorbereitung können ungefähr 60min angegeben werden.

e) Durchführung:

Zur Vorbereitung des Experiments muss der Testfall zunächst in einem Pilotversuch getestet werden sowie die Testreihe als Inputdatei für die Simulation erzeugt werden. Die Durchführung wird mithilfe des sogenannten Testmanagers automatisiert, der nacheinander alle Einzeltestfälle aus dem Versuchsplan simuliert (siehe Abbildung 4.12).

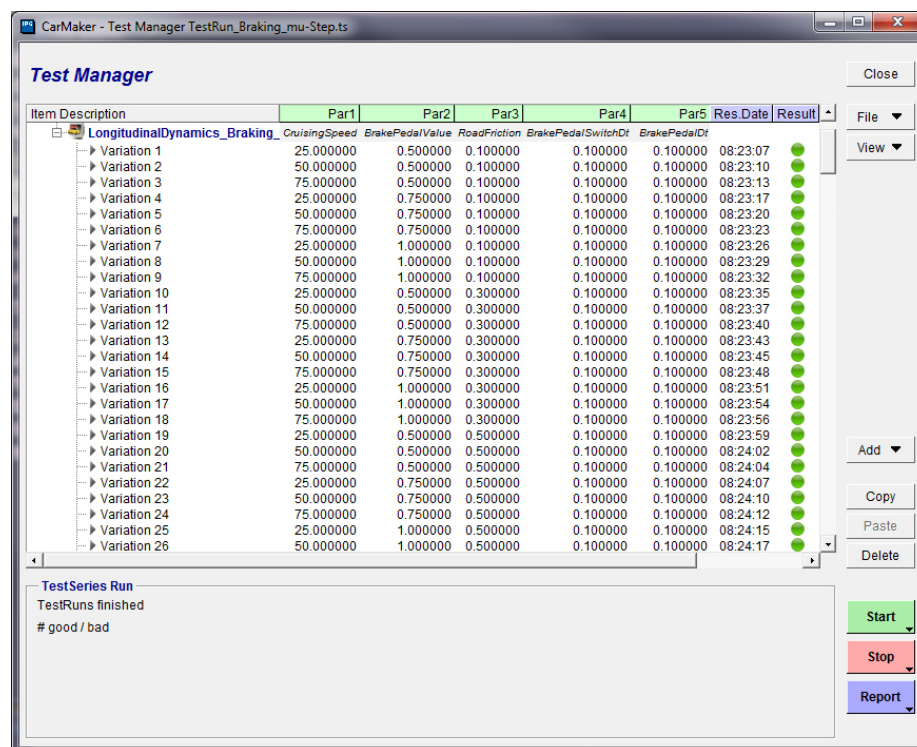


Abbildung 4.12. – Dargestellt wird die Durchführung der Testreihe mithilfe des Testmanagers

f) Auswertung:

Die Auswertung und Interpretation kann mithilfe verschiedener statistischer Methoden erfolgen.

Zunächst wird das Gesamtergebnis der Versuchsreihe in Abbildung 4.13 dargestellt. Aus dieser Abbildung kann der in der Versuchsreihe minimal erreichte Wert für Deceleration abgelesen werden. Dieser traf bei Versuchsnummer 88 ein und beträgt $-0,6720\text{m/s}^2$ bei

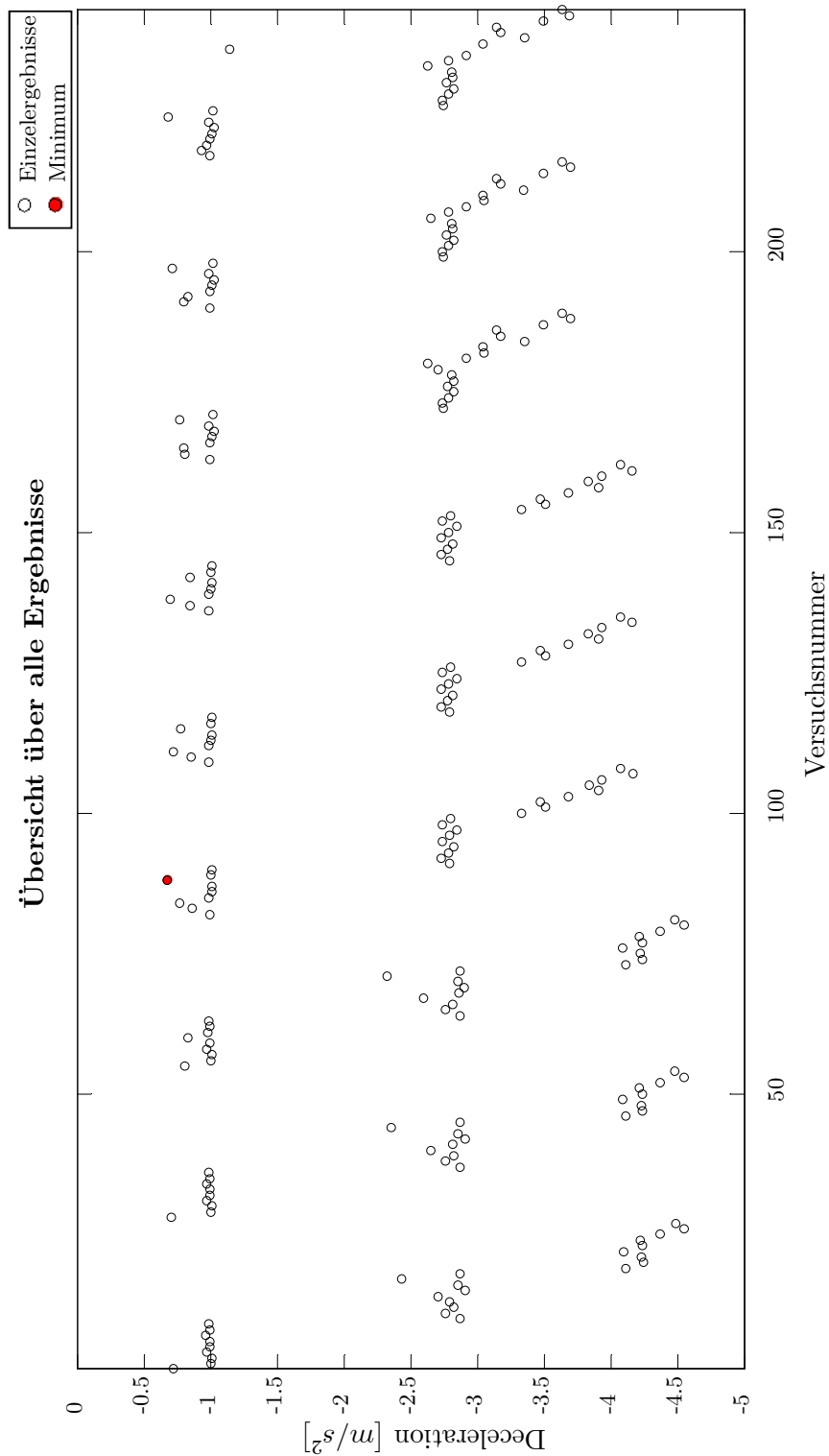


Abbildung 4.13. – Das Gesamtergebnis des ausgeführten Versuchsplans zeigt eine große Bandbreite für die Ergebnissen. Jedoch wird schnell ersichtlich, dass durchaus Muster und wiederkehrende Verhaltensweisen auftreten, die es zu analysieren gilt. Das minimale Ergebnis wurde ebenfalls dargestellt, da in dieser Reihe ein möglichst niedriges Ergebnis (möglichst wenig Verzögerung, da auf niedrigem Reibwert gebremst wird) gewünscht ist.

folgenden Faktorstufen:

CruisingSpeed	+ : 25km/h
Brake	- : 1,0
Friction	- : 0,1
BrakePedalSwitchDt	+ : 0,1s
BrakeDt	+ : 0,3s

Zur besseren Übersichtlichkeit und zur Verdeutlichung, dass unterschiedliche Muster in den Ergebnissen vorhanden sind, werden in Anhang A noch die Ergebnisse der einzelnen Faktorstufen gesondert dargestellt.

Die Auswirkungen bzw. die Schlussfolgerungen aus diesen Ergebnissen kann eine sogenannte Effektanalyse mathematisch aufzeigen, die sowohl die Haupteffekte der einzelnen Faktoren, aber auch die Wechselwirkungseffekte zwischen den Faktoren bezogen auf die Zielgröße aufzeigt.

Zunächst berechnet man für die einzelnen Faktoren die Mittelwerte der Versuchsergebnisse $y(x_i)$ auf den drei Stufen +, o und - (nach [68]):

$$\text{Mittelwert der Versuchsergebnisse bei } x_i^+ : \frac{\sum_{i=1+}^{N_+} y(x_i)}{N_+} \quad (4.1)$$

$$\text{Mittelwert der Versuchsergebnisse bei } x_i^o : \frac{\sum_{i=1o}^{N_o} y(x_i)}{N_o} \quad (4.2)$$

$$\text{Mittelwert der Versuchsergebnisse bei } x_i^- : \frac{\sum_{i=1-}^{N_-} y(x_i)}{N_-} \quad (4.3)$$

Der Haupteffekt oder Faktoreffekt eines Faktors stellt die Differenz zwischen oberem und unterem Mittelwert dar. Nach [42] kann dies mit folgender Formel dargestellt werden:

$$\text{Effekt}(x_i) = \frac{\sum_{i=1+}^{N_+} y(x_i)}{N_+} - \frac{\sum_{i=1-}^{N_-} y(x_i)}{N_-} \quad (4.4)$$

Je größer der jeweilige Effekt der Faktoren ist, desto größer ist die Wirkung des Faktors auf die Zielgröße. Das Vorzeichen des Effekts zeigt die Richtung des Effekts, also erhöhend oder verringernd [3].

Die Ergebnisse der Mittelwerte und der Effekte aus dem angeführten Beispiel sind in Tabelle 4.5 aufgeführt. Abbildung 4.14 zeigt zusätzlich das Ergebnis graphisch. Deutlich wird die Höhe des Einflusses sowie die Richtung der Wirkung. Zu interpretieren sind die Ergebnisse folgendermaßen: z. B. bei Friction bedeutet ein Effekt von $-2,8096\text{m/s}^2$, dass der Mittelwert der Ergebnisse bei einer Friction von 0,5 (+-Wert) um $-2,8096\text{m/s}^2$ niedriger ist als bei 0,1 (--Wert). Während der Faktor Friction einen sehr großen

Faktor	Mittelwert x_i^-	Mittelwert x_i^0	Mittelwert x_i^+	Effekt x_i
CruisingSpeed	-2,4798m/s ²	-2,4686m/s ²	-2,5204m/s ²	-0,0405m/s ²
Brake	-2,3833m/s ²	-2,5094m/s ²	-2,5761m/s ²	-0,1929m/s ²
Friction	-0,9434m/s ²	-2,7724m/s ²	-3,7530m/s ²	-2,8096m/s ²
BrakePedalSwitchDt	-2,4976m/s ²	-2,4965m/s ²	-2,4747m/s ²	0,0229m/s ²
BrakePedalDt	-2,6700m/s ²	-2,4933m/s ²	-2,3056m/s ²	0,3644m/s ²

Tabelle 4.5. – Mittelwerte und Ergebnisse der Haupteffekte für die einzelnen Faktorstufen

verringern den Effekt hat, haben die Faktoren CruisingSpeed und Brake einen geringen verringern den Einfluss, der wesentlich geringer ist als der von Friction. Die Faktoren BrakePedalSwitchDt und BrakePedalDt haben einen geringen erhöhenden Einfluss. Des Weiteren sollten auch die Wechselwirkungs- oder Interaktionseffekte zwischen Faktoren untersucht werden, die ein Maß dafür sind wie stark die Wirkung eines Faktors von einem anderen abhängt. Anders gesagt, hängt der Effekt des einen Faktors davon ab, welchen Wert der andere Faktor hat. Nach [42] definiert sich eine allgemeingültige Berechnungsformel der Abhängigkeit der Wirkung des Faktors x_i von x_j zu:

$$\text{Effekt}(x_i x_j) = \frac{\sum_{i=1+}^{N_+} \sum_{j=1+}^{N_+} y(x_i x_j)}{N_+} - \frac{\sum_{i=1-}^{N_-} \sum_{j=1-}^{N_-} y(x_i x_j)}{N_-} \quad (4.5)$$

Berechnet wird zunächst der Effekt von x_i , wenn x_j die obere Stufe + hat und anschließend wird der Effekt von x_i , wenn x_j auf der niedrigen Stufe – steht, abgezogen. Dasselbe Ergebnis wird erzielt, wenn die Abhängigkeit des Faktors x_j zu x_i berechnet wird. Somit kommt man im Beispiel auf insgesamt 15 Ergebnisse, die in Tabelle 4.6 dargestellt sind.

Ermittelt werden können die Wechselwirkungen zwischen den Faktoren auch graphisch.

	Brake	Friction	BrPedalSwitchDt	BrakePedalDt
CruisingSpeed	-0,0197m/s ²	-0,0359m/s ²	0,00064m/s ²	0,0462m/s ²
Brake		-0,2616m/s ²	-0,0313m/s ²	-0,0969m/s ²
Friction			0,0405m/s ²	0,5282m/s ²
BrPedalSwitchDt				0,0248m/s ²

Tabelle 4.6. – Ergebnisse der Wechselwirkungseffekte

Das Verfahren hierzu wird in Abbildung 4.15 exemplarisch für die Faktoren Friction und BrakePedalDt aufgezeigt. Allgemein graphisch dargestellt sind mögliche Ausprägungen von Wechselwirkungseffekten in Abbildung 4.16. Graphisch interpretiert kann man also sagen, dass die Wechselwirkungseffekte umso größer sind, je weiter die beiden Linien

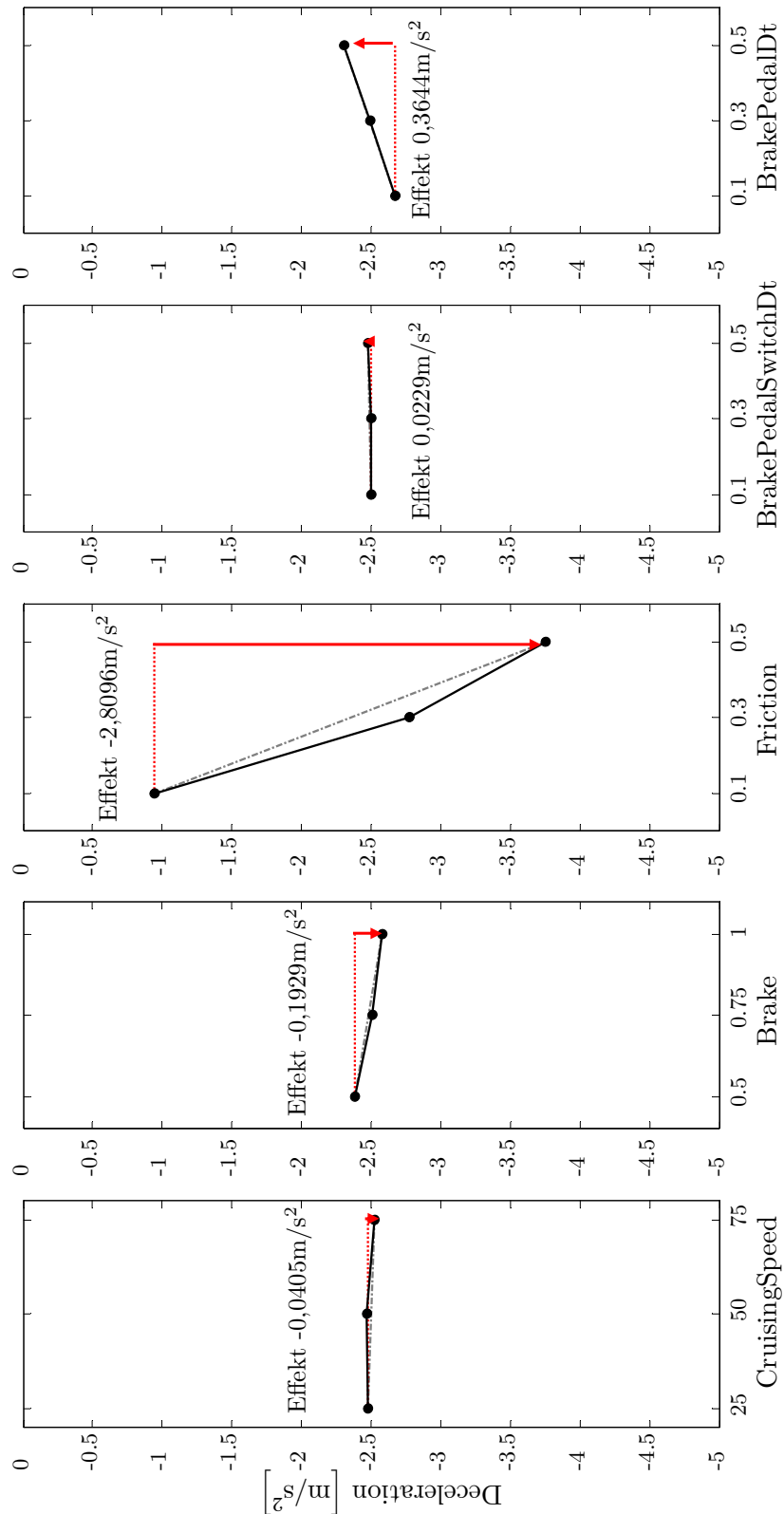


Abbildung 4.14. – Die graphische Darstellung der berechneten Haupteffekte zeigt die errechneten Mittelwerte aus Tabelle 4.5 als Punkte. Die daraus errechneten Effekte der einzelnen Faktoren sind hier als rote Pfeile gekennzeichnet.

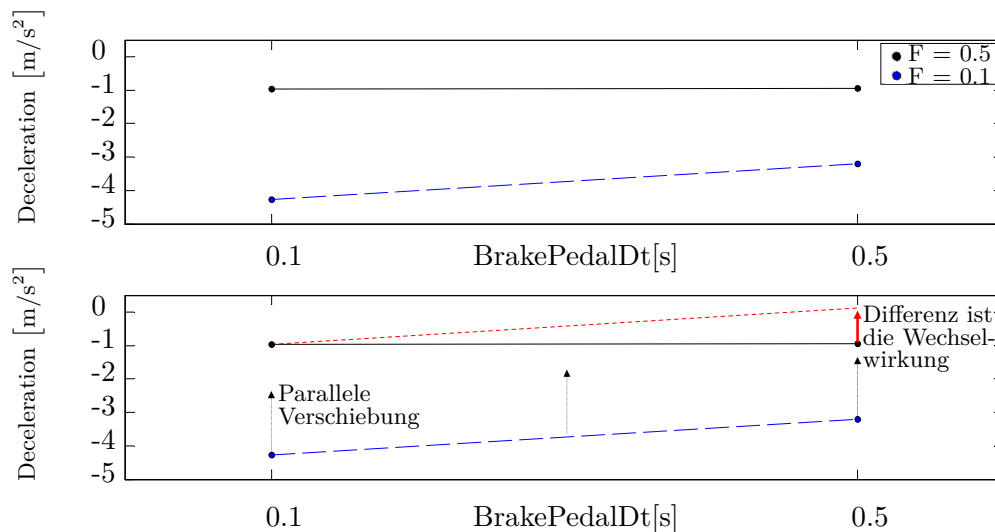


Abbildung 4.15. – Die graphische Auswertung des Wechselwirkungseffekts zwischen Friction und BrakePedalDt erfolgt mithilfe der vier Mittelweltergebnisse aus den Versuchskombinationen (Friction₋, BrakePedalDt₋), (Friction₋, BrakePedalDt₊), (Friction₊, BrakePedalDt₋) und (Friction₊, BrakePedalDt₊). Die Parallelität der beiden Linien ist ein Maß für die Wechselwirkungseffekte zwischen den Faktoren: je parallel die Linien sind, desto weniger Effekte gibt es zwischen den Faktoren. Wenn man nun wie im unteren Bild dargestellt die Linie mit den beiden Werten für Friction₊ parallel auf den Anfangspunkt der Linie der Werte Friction₋ verschiebt, ergibt sich als Differenz der Endpunkte der Wechselwirkungseffekt von 0,5282m/s² (hier als roter Pfeil dargestellt).

von der Parallelität entfernt sind.

Die graphische Auswertung der Ergebnisse aller Wechselwirkungseffekte des Beispiels sind in Abbildung 4.17 dargestellt.

Hier wird auf den ersten Blick deutlich, dass im Vergleich zu Abbildung 4.16 zwischen den Faktoren Friction und Brake sowie Friction und BrakePedalDt Wechselwirkungseffekte vorherrschen. Somit ergibt sich dasselbe Resultat wie in Tabelle 4.6.

Die Kombination aus den Einzeleffekten und den Wechselwirkungseffekten wird in Abbildung 4.18 aufgezeigt. Hier wird deutlich, dass oftmals das Ergebnis durch die Wechselwirkungseffekte stärker beeinflusst werden als durch die Einzeleffekte allein. Des Weiteren wird hier auch deutlich, dass die Höhe, also die Stärke des Einflusses des Effekts auf die Zielgröße, sehr unterschiedlich ausgeprägt ist. Es ist offensichtlich, dass Maßnahmen zur Verringerung der Komplexität des Versuchsaufbaus möglich sind, indem Faktoren reduziert werden, indem sie auf einen fixen Wert gesetzt werden. Wechselwirkungen zwischen bestimmten Faktoren sollten berücksichtigt werden.

g) Ergebnisinterpretation, Maßnahmendefinition:

Aus den erhaltenen Ergebnissen können nach Durchführung des Versuchs verschiedene Schlüsse gezogen werden:

- Der Effekt der fünf Faktoren ist unterschiedlich hoch. Die Faktoren CruisingSpeed

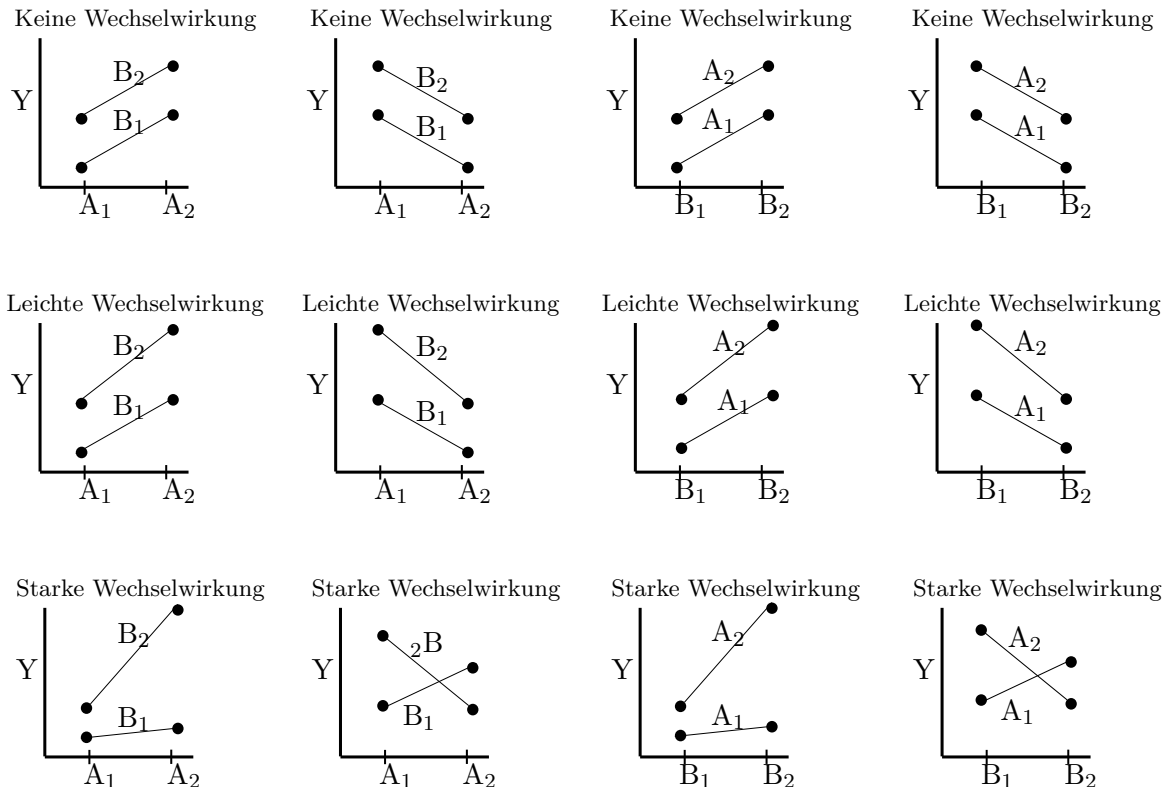


Abbildung 4.16. – Darstellung von Beispielen für Wechselwirkungseffekte zwischen zwei Faktoren A und B mit je zwei Werten mit der Zielgröße Y (nach [85])

und BrakePedalSwitchDt haben einen so geringen Einfluss auf das Ergebnis, dass sie vernachlässigt werden können \Rightarrow Reduktion der Komplexität des Versuchsaufbaus

- Der Effekt von Friction ist der höchste, sodass das Hauptaugenmerk auf diesem Faktor liegen muss.
- Die Faktoren sind nicht unabhängig voneinander betrachtbar. Die Wechselwirkungseffekte sind teilweise größer als die Einzeleffekte, was impliziert, dass man die Faktoren in Kombination betrachten muss.

h) Absicherung, Dokumentation:

Da in diesem Beispiel aufgrund der reproduzierbaren Simulation keine Ergebnisstreuung zu erwarten ist, ist keine erneute Durchführung des Versuchs notwendig. Die gewonnenen Ergebnisse und Erkenntnisse können nun als Basis für andere Methoden wie zum Beispiel die evolutionären Algorithmen genutzt werden. Zum einen kann die Parameterzahl des Versuchs auf drei verringert werden, indem die Effektanalyse ausgenutzt wird in der der Einfluss von zwei Parametern als gering aufgezeigt wurde. Des Weiteren können die 243 Versuchsergebnisse als Basis für die Initialisierung von evolutionären Algorithmen genutzt werden, wie nachfolgend in Kapitel 4.4 aufgezeigt wird: je nach Suchwunsch kann der Suchalgorithmus bereits an einer Stelle initialisiert werden, an der er sich bereits in der Nähe des gewünschten Resultats befindet. So kann der Algorithmus bereits an einer voroptimierten Stelle starten und wesentlich schneller weiterarbeiten.

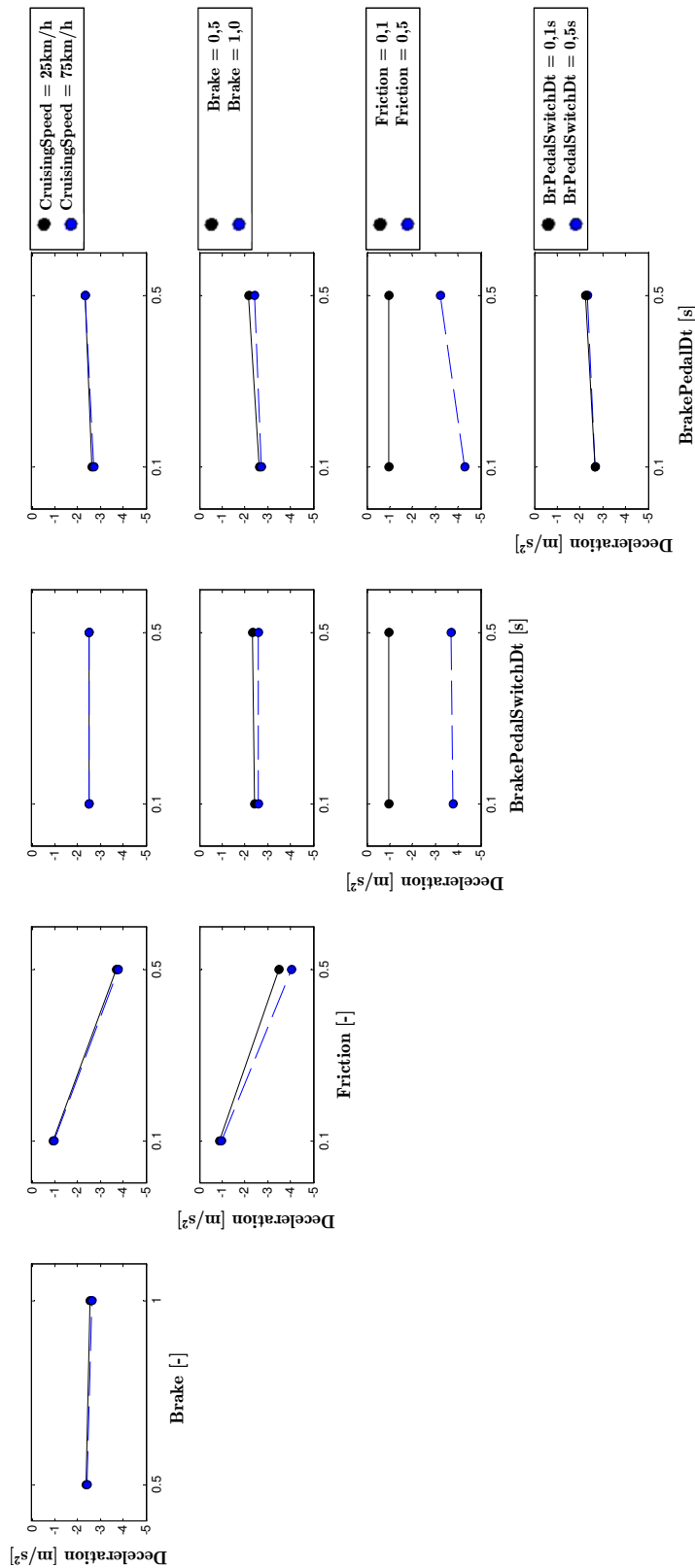


Abbildung 4.17. – Die graphische Auswertung der Wechselwirkungseffekte aus Tabelle 4.6 ergibt sich aus der Parallelität der Linien (gestrichelt ist jeweils der höhere Parameterwert): je parallel die Linien, desto weniger Wechselwirkungen sind zwischen den Faktoren vorhanden.

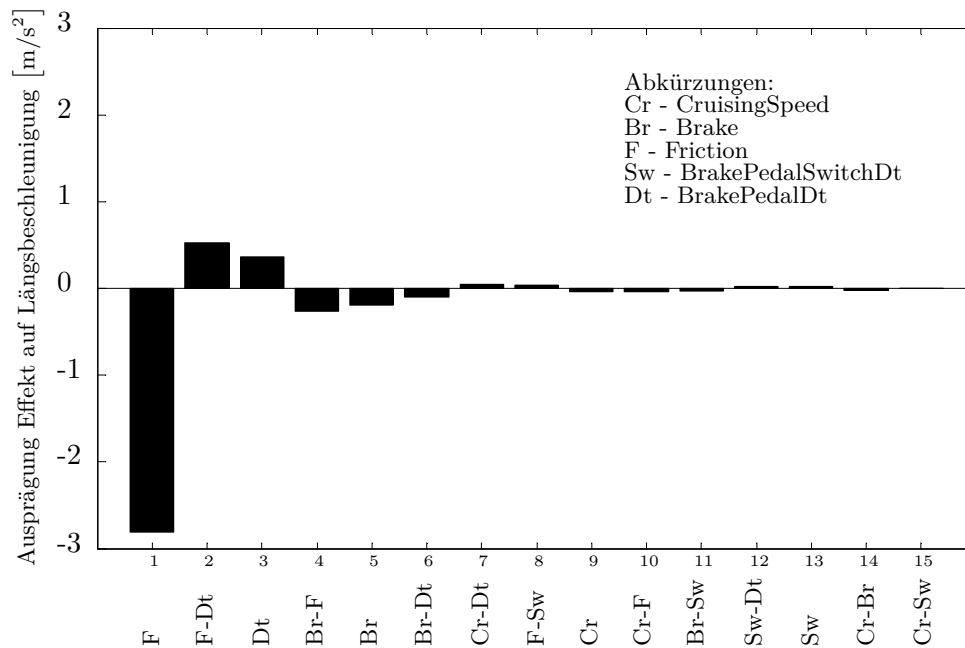


Abbildung 4.18. – Die Einzel- und Wechselwirkungseffekte werden hier nach ihrer absoluten Höhe, d. h. nach dem Maß ihrer Einflussnahme auf die Zielgröße, sortiert. Deutlich wird, dass oft Wechselwirkungseffekte von mehreren Faktoren gewichtiger sind als Einzeleffekte von Faktoren.

4.3.4. Fazit

Vorgelegt wurde eine Methode zur Analyse von Testfällen. Anhand eines Beispiels aus dem manöverbasierten Testen wurde ein Testfall analysiert, Einflussgrößen und Wechselwirkungen qualitativ und quantitativ wiedergegeben und interpretiert. Die Vorteile der Methode sind folgende:

- Es können durch die Analyse Schlüsse auf die Notwendigkeit der Verwendung von Parametern gezogen werden, d. h. die Parameterdimension eines Testfalls kann gegebenenfalls reduziert werden, was die Simulationszeit verkürzt.
- Der Einfluss der einzelnen Parameter sowie die Wechselwirkungen und der Einfluss der kombinierten Betrachtung von Parametern kann ermittelt werden. Dadurch lassen sich Parametergrenzen hinsichtlich des gewünschten Systemverhaltens festlegen.
- Der Einsatz bei Funktionsentwicklung und in der Designphase kann zur Anforderungsdefinition und zum Algorithmestest erfolgen.
- Ergänzend oder auch tiefergehender Analyseeinsatz ist möglich.
- Die Analyse kann als Initialisierung für evolutionäre Algorithmen dienen, wie es im nachfolgenden Kapitel 4.4 dargelegt wird.

Die Analyse eines Testfalls kann durch diese Methode ohne viel Aufwand und schnell realisiert werden. Das Wissen über die Einflüsse und das Verhalten des Testfalls kann

vielfältig eingesetzt und als Zugewinn bei Design, Kalibrierung und Test genutzt werden.

4.4. Testmethode Differential Evolution²

4.4.1. Einleitung

Seit den 1950er Jahren existieren evolutionäre Optimierungsverfahren, die die Idee umsetzen, darwinistische Prinzipien aus der Natur für die automatisierte Problemlösung zu verwenden. Als Vorbild der evolutionären Optimierungsmethoden dient die darwinistische Evolutionstheorie der Selektion der „fittesten“, respektive bestangepassten Individuen. Die Entstehung der optimierten Anpassung an die Umgebung resultiert aus der Mutation, Rekombination und Selektion der Individuen, die dem Ideal der notwendigen Charakteristika, die zum Überleben und zum Einnehmen einer noch freien Nische in der Natur erforderlich sind, am nächsten kommen [19], [18].

Der geläufige Oberbegriff für diese Verfahren lautet Evolutionäre Algorithmen (EA). Nach [121] sind Evolutionäre Algorithmen definiert zu:

Definition 4.2 (Evolutionäre Algorithmen)

Evolutionäre Algorithmen (EA) stellen probabilistische Optimierungsverfahren zur Parameteroptimierung dar, die dem Vorbild der natürlichen Evolution folgen. Eine Punktmenge im Suchraum wird als Population von Lebewesen in einer künstlichen Umwelt betrachtet und durch Reproduktion, Mutation, Rekombination und Selektion so verändert, dass neue Generationen von Suchpunkten entstehen, die mit hoher Wahrscheinlichkeit durchschnittlich besser sind als ihre Vorfahren.

Wichtig ist noch den Begriff der Optimierung genau abzugrenzen. Optimierung ist nach [121] definiert zu:

Definition 4.3 (Optimierung)

Optimierung bezeichnet die Verbesserung eines gegebenen Systems anhand festgelegter Gütekriterien. Es werden dabei vereinfachende Modelle der Systeme verwendet, um die Komplexität der Aufgabe, Kosten und Gefahren zu reduzieren.

Abbildung 4.19 zeigt das prinzipielle Vorgehen des Prozesses in der Natur. Zufällige Mutationen können sowohl vorteilhaft als auch nachteilig für das einzelne Individuum sein. Im Falle des Nachteils gegenüber den anderen Individuen wird sich diese Eigenschaft nicht langfristig durchsetzen, z. B. weil das Individuum aufgrund von Problemen bei der Jagd verhungert. Ist eine Mutation jedoch nutzbringend, so kann sich die Mutation durch Rekombination und Selektion in der Fortpflanzung durchsetzen und nach einigen Generationen erfolgt eine Optimierung bzw. bessere Anpassung oder Fitness der gesamten

²Teile dieses Kapitels sind wörtlich in Veröffentlichung [101] enthalten

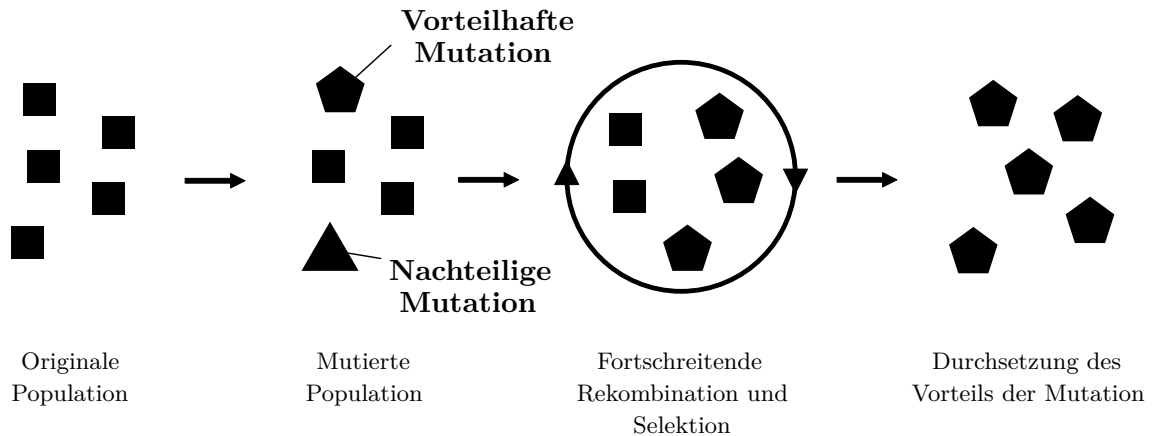


Abbildung 4.19. – Dargestellt wird die Vererbung und Durchsetzung von vorteilhaften Mutationen. Aus einer Originalpopulation entwickeln sich durch Mutationen neue Individuen. Vorteilhaft Mutationen, die zu einer besseren Anpassung an die Umwelt führen, werden über mehrere Generationen durch Rekombination und Selektion immer dominanter, bis hin zur vollständigen Durchsetzung der Eigenschaft [106], [101].

Population an die individuelle Problemstellungen der Umgebung. Für eine exakte Beschreibung der genetischen Vorgänge in der Natur sei auf einschlägige Fachliteratur wie [13] oder [100] verwiesen.

Grundsätzlich werden EA weiter untergliedert. Nach [138] unterscheidet man grundsätzlich in zwei verschiedene Standardalgorithmen: Genetische und Evolutionäre Verfahren. Der Hauptunterschied liegt darin, dass genetischen Verfahren für binärcodierte Suchräume eingesetzt werden, d. h. nicht für reellwertige Probleme. Inzwischen gibt es auch reellwertige genetische Algorithmen, die jedoch in der Leistungsfähigkeit nicht an die evolutionären Ansätze reichen. Evolutionäre Verfahren dagegen beschäftigen sich grundsätzlich mit reellwertigen Problemstellungen. Weitere Unterschiede liegen u. a. in der Geschwindigkeit und der Betonung von Mutation oder Rekombination. Grundsätzlich existieren unzählige Verfahren, die jeweils Vor- und Nachteile mit sich bringen. Für die Lösung verschiedener Probleme können also unterschiedliche Algorithmen zum Einsatz kommen. Für grundlegende Informationen sei hierfür auf tiefere Literatur wie [138] verwiesen. Eine Untergruppe der evolutionären Verfahren ist Differential Evolution (DE). DE stellt eine Weiterentwicklung von evolutionären Optimierungsmethoden dar, die zum ersten Mal 1995 von R. Storn und K. V. Price vorgeschlagen wurde [128]. Sie ist aktuell einer der mächtigsten stochastischen Algorithmen zur Optimierung von reellwertigen Parametern. Neben dem guten Konvergenzverhalten und der Möglichkeit der Parallelisierung, ist das Verfahren außerdem konzeptionell einfach aufgebaut und zu verstehen und es ist einfach um- und einzusetzen. Der Algorithmus hat auch nur wenig Kontrollparameter, die ihn kalibrieren, sodass wenig Initialaufwand besteht [127]. Außerdem kann nachfolgend noch gezeigt werden, dass der Algorithmus sehr robust gegenüber der Einstellung der Kontrollparameter ist und somit falsche Anwendung und Handhabung minimiert werden kann. Eine übersichtliche Zusammenfassung über die aktuellen Entwicklungen im Bereich der DE gibt das Survey-Paper von S. Das und P. N. Suganthan [20].

Der nachfolgende Einsatz der DE in Kombination mit manöverbasiertem Testen begründet sich darin, dass bereits mehrfach gezeigt wurde, dass DE der beste bzw. schnellste EA zur Lösung reellwertiger Probleme ist [20]. Da das manöverbasierte Testen auf Fahrdynamik- und Umfeldsimulation beruht, die physikalischen Prinzipien folgt und somit ein reellwertiges Problem darstellt, ist es naheliegend, den momentan leistungsfähigsten Algorithmus zur Problemlösung zu verwenden.

4.4.2. Prinzipielle Funktionsweise der DE

Die Methode DE nimmt sich ein Beispiel am Vorgehen aus der Natur, um die Optimierung einer Problemstellung zu bewältigen. Durch einen iterativen Optimierungsprozess durch ebenfalls Mutation, Rekombination und Selektion der Testindividuen hin zum Ziel der Anpassung an eine sogenannte Fitnessfunktion können vielfältigste Suchprobleme bewältigt werden. Abbildung 4.20 zeigt das grundsätzliche Prinzip der DE, das der Natur sehr ähnlich ist. Das Vorgehen der DE auf Datenebene, das auf verschiedenste Probleme angewendet

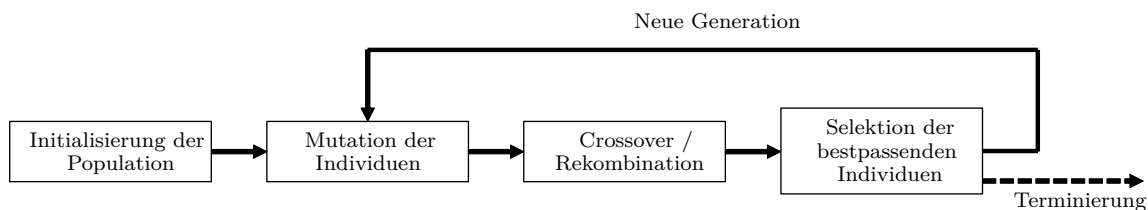


Abbildung 4.20. – Dargestellt wird das Prinzip der DE durch Mutation, Rekombination und Selektion ([20], [106], [87]).

werden kann, wird in Abbildung 4.21 aufgezeigt. Begonnen wird mit der Initialisierung der Population. Durch Mutation einzelner Bits wird die Eltern-Generation verändert. Anschließend werden durch Crossover bzw. Rekombination neue Individuen gebildet. Als letzter Schritt werden aus dem Pool der neu erzeugten und der Eltern-Generation die bestpassenden Individuen selektiert und zu einer neuen Generation zusammengefasst. Begonnen wird mit der Initialisierung der Population. Durch Mutation einzelner Bits wird die Eltern-Generation verändert. Anschließend werden durch Crossover bzw. Rekombination neue Individuen gebildet. Als letzter Schritt werden aus dem Pool der neu erzeugten und der Eltern-Generation die bestpassenden Individuen selektiert und zu einer neuen Generation zusammengefasst.

4.4.3. Mathematische Abbildung der DE und Pseudo-Code

Nachfolgend soll das Vorgehen der DE mathematisch und anschließend in Pseudo-Code erarbeitet werden [20].

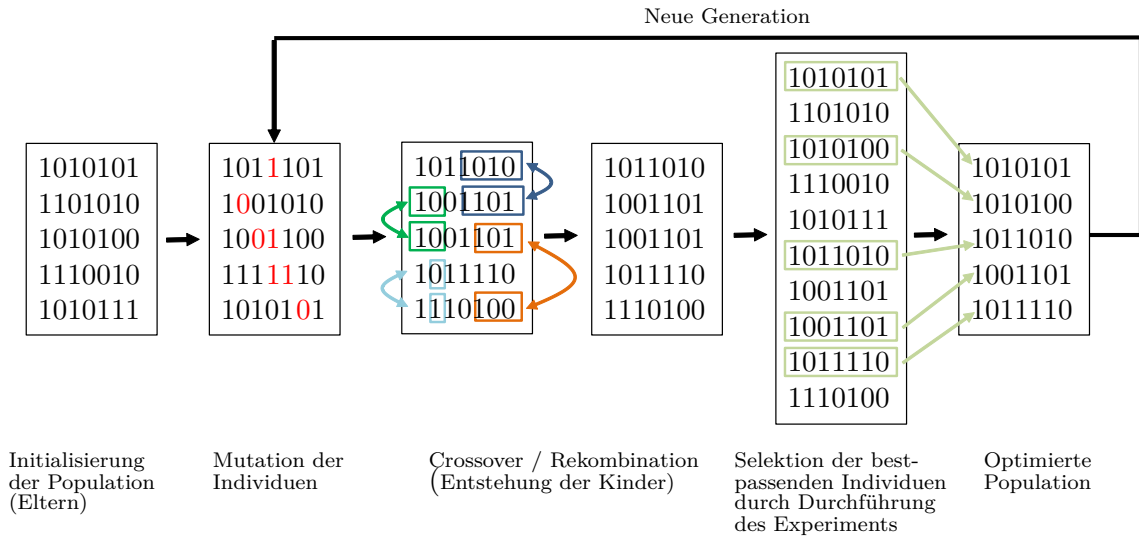


Abbildung 4.21. – Dargestellt wird das Prinzip der DE durch Mutation, Rekombination und Selektion auf Datenebene. Dieses Beispiel verfolgt keine Logik hinsichtlich der Mutation, Rekombination und Selektion, sondern ist eine schematische Darstellung.

- **Initialisierung:**

Die DE sucht nach einem globalen Optimum in einem D -dimensionalen realen Parameterraum \mathfrak{R}^D . Die Initialisierung der Population der Größe N erfolgt durch N D -dimensionale realwertige Parametervektoren mit zufälligen Werten. Die Parametervektoren werden durch folgende Notation repräsentiert:

$$\vec{X}_{i,G} = [x_{1,i,G}, x_{2,i,G}, x_{3,i,G}, \dots, x_{D,i,G}] \quad (4.6)$$

Hierbei stellt der erste Index $j = 1, 2, \dots, D$ den Zähler für die Dimension der Parameter dar, der zweite zeigt den i -ten Vektor der Population, d. h. das Individuum der aktuellen Generation mit $i = 1, 2, \dots, N$ an und der dritte Index repräsentiert die aktuelle Generation mit $G = 0, 1, 2, \dots, G_{max}$.

Da bei physikalischen Größen oft eine Beschränkung der Parameterwerte vorherrscht (z. B. dass ein Wert nicht negativ werden kann), werden die minimalen und maximalen Grenzen der einzelnen Parameter folgendermaßen festgelegt: $\vec{X}_{min} = \{x_{1,min}, x_{2,min}, \dots, x_{D,min}\}$ und $\vec{X}_{max} = \{x_{1,max}, x_{2,max}, \dots, x_{D,max}\}$.

Die Initialpopulation $G = 0$ sollte die Bandbreite der festzulegenden Parametergrenzen $[x_{j,min}, x_{j,max}]$ möglichst umfassend abdecken, weshalb eine gleichmäßig-zufällige Verteilung über den Wertebereich angestrebt werden sollte. Der Wertebereich der Parametergrenzen kann sich sowohl aus physikalischen Gegebenheiten sowie aus manuell festgelegten Grenzen, die durch die eingrenzende Festlegung aus dem Interessensgebiet des Testfalles heraus entstehen, ergeben. Dies wird durch folgende Formel gewährleistet:

$$x_{j,i,0} = x_{j,min} + rand_{i,j} [0; 1] \cdot (x_{j,max} - x_{j,min}) \quad (4.7)$$

Hierbei stellt $rand_{i,j} [0; 1]$ eine gleichmäßig verteilte Zufallszahl zwischen 0 und 1 dar, die für jede Komponente des i -ten Vektors einzeln und unabhängig berechnet wird.

- **Differenzvektorbasierte Mutation:**

Mutation wird in DE als Veränderung oder Perturbation eines Individuums der aktuellen Generation mithilfe eines zufälligen Elements verstanden. Die Notation der hauptsächlich englischsprachigen DE-Literatur bezeichnet einen Eltern-Vektor der aktuellen Generation als *Target* („Ziel“), ein mutierter Vektor der aktuellen Generation heißt *Donor* („Spender“) und der Abkömmling, der durch die Rekombination des Targets und des Donors entsteht, wird *Trial* („Erprobung“) genannt.

Eine Form der DE-Mutation sieht die Erzeugung eines Donor-Vektors für jeden i -ten Target-Vektor der aktuellen Generation durch drei andere unterschiedliche Parametervektoren $\vec{X}_{r_1^i}$, $\vec{X}_{r_2^i}$ und $\vec{X}_{r_3^i}$ aus der aktuellen Population vor. Neu erzeugte Kind-Vektoren werden nicht einbezogen, d. h. es wird für jedes Eltern-Individuum separat und sequentiell ein Kind-Individuum unter Einbezug der anderen Eltern-Individuen erzeugt. Die Indizes r_1^i , r_2^i und r_3^i sind hierbei drei unterschiedliche ganzzahlige Werte zwischen $[1; N]$, die nicht den Wert i des Target-Vektors haben dürfen. Die Indizes werden für jeden zu mutierenden Vektor einzeln berechnet. Nun werden zwei beliebige der drei ausgewählten Vektoren subtrahiert und mit dem Skalar F multipliziert. Anschließend wird die skalierte Differenz zu dem dritten ausgewählten Vektor addiert, wodurch der Donor-Vektor $\vec{V}_{i,G}$ entsteht. Das Vorgehen wird in folgender Formel zusammengefasst:

$$\vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F \cdot (\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G}) \quad (4.8)$$

Abbildung 4.22 zeigt der Übersichtlichkeit halber den Mutationsprozess für $D = 2$.

- **Crossover / Rekombination:**

Um die potentielle Diversität der Gesamtpopulation zu erhöhen, wird nach der Mutation noch ein Crossover bzw. Rekombination durchgeführt. Grundsätzlich existieren zwei Möglichkeiten zur Rekombination, eine exponentielle und eine binominale (uniforme) Methode. Da nach [141] der Unterschied der beiden Methoden hauptsächlich in der Distribution der mutierten Komponenten liegt und dies nur geringe Auswirkungen auf die Optimumlösung hat und auch durch einen Kontrollparameter gesteuert werden kann, wird aufgrund der besseren Anschaulichkeit der Methode das binominale Verfahren gewählt.

Bei der Rekombination tauscht der durch die Mutation erhaltene Donor-Vektor $\vec{V}_{i,G}$ Komponenten mit dem Target-Vektor $\vec{X}_{i,G}$ aus, um den Trial-Vektor $\vec{U}_{i,G} = [u_{1,i,G}, u_{2,i,G}, u_{3,i,G}, \dots, u_{D,i,G}]$ zu formen. Beim binominalen Crossover werden zwischen den D Parametern immer dann Komponenten ausgetauscht, wenn eine zufällig generierte Zahl zwischen 0 und 1 kleiner oder gleich der sogenannten Crossoverrate Cr ist. Cr stellt genau wie F einen Kontrollparameter der DE dar. Das Vorgehen

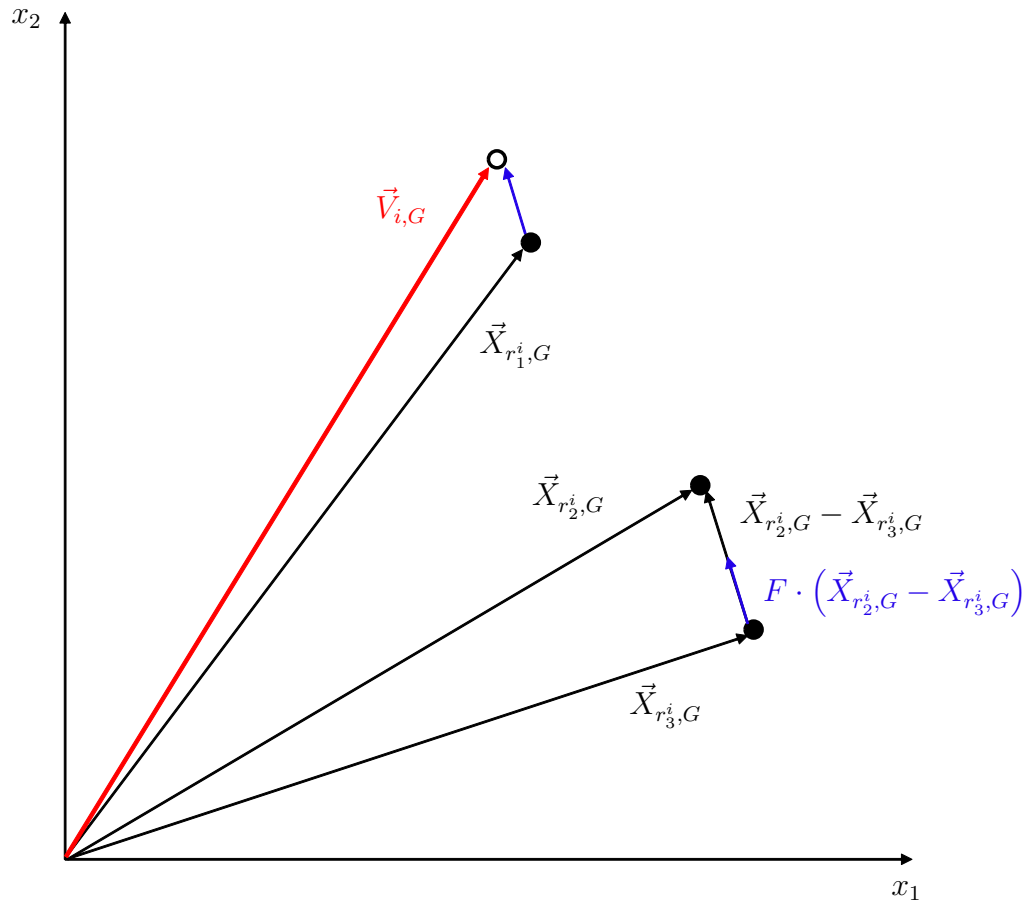


Abbildung 4.22. – Dargestellt wird das DE-Mutationsverfahren im 2- D -Parameterraum. Ein Donor-Vektor $\vec{V}_{i,G}$ wird durch Addition eines skalierten Differenzvektors $F \cdot (\vec{X}_{r_2,G} - \vec{X}_{r_3,G})$ mit einem dritten, ausgewählten Vektor $\vec{X}_{r_1,G}$ gebildet. Die Vektoren $\vec{X}_{r_j,G}$ sind jeweils zufällig gewählte Individuen der Generation G [101], [106], [87].

lässt sich folgendermaßen zusammenfassen:

$$\vec{u}_{j,i,G} = \begin{cases} \vec{v}_{j,i,G} & \text{falls } (\text{rand}_{i,j}[0;1] \leq Cr \text{ oder } j = j_{rand}) \\ \vec{x}_{j,i,G} & \text{sonst} \end{cases} \quad (4.9)$$

$\text{rand}_{i,j}[0;1]$ ist eine gleichförmig verteilte Zufallszahl, die für jede der j Komponenten des i -ten Parameters einzeln berechnet wird. $j_{rand} \in [1, 2, \dots, D]$ ist ein zufällig gewählter Index, der sicherstellt, dass $\vec{U}_{i,G}$ zumindest eine Komponente von $\vec{V}_{i,G}$ erhält. Dieser Index wird für jeden Vektor einmal pro Generation ermittelt. Abbildung 4.23 zeigt das Vorgehen des binominalen Crossovers anschaulich.

- **Selektion:**

Da die Populationsgröße N über mehrere Generationen hinweg konstant zu halten ist, muss eine Selektion erfolgen. Hierbei wird ermittelt, welcher Trial-Vektor in die nachfolgende Generation $G = G+1$ übernommen wird bzw. ob das Eltern-Individuum weitergeführt wird oder ob das zugehörige Kind-Individuum das Eltern-Individuum

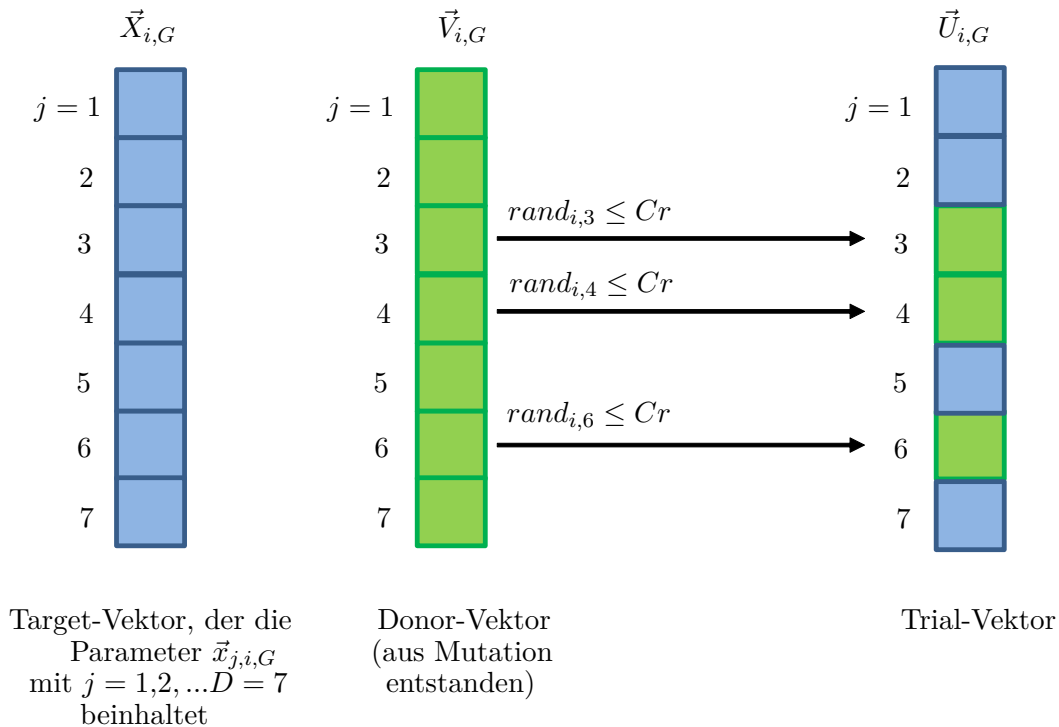


Abbildung 4.23. – Dargestellt wird das DE-Crossoververfahren im 7- D -Parameterraum zur Erzeugung des Trial-Vektors [101], [106], [129].

ersetzt. Für die Selektion wird das folgende Schema angewendet:

$$\vec{X}_{i,G+1} = \begin{cases} \vec{U}_{i,G} & \text{falls } f(\vec{U}_{i,G}) \leq f(\vec{X}_{i,G}) \\ \vec{X}_{i,G} & \text{falls } f(\vec{U}_{i,G}) > f(\vec{X}_{i,G}) \end{cases} \quad (4.10)$$

Hier stellt $f(\vec{X})$ die zu minimierende Funktion dar. Man spricht auch von Fitness- oder Zielfunktion, die z. B. ein zu minimierender physikalischer Wert sein kann. Es werden also dann Individuen aus dem Pool der vorhandenen und neu generierten Vektoren ausgewählt, wenn sie eine Verbesserung darstellen. Somit kann die Population nur besser werden oder gleich bleiben, sich aber nie verschlechtern. Man kann auch sagen, dass sich die Fitness der Individuen verbessert oder gleich bleibt.

Das beschriebene Vorgehen wird kompakt in Tabelle 4.7 durch Pseudo-Code dargestellt [20]. Das hier beschriebene Verfahren wird auch kurz als DE/rand/1/bin bezeichnet [20]. Die generelle Namenskonvention ist DE/x/y/z, wobei DE für Differentielle Evolution steht, x stellt die Auswahlweise des Basisvektors dar (in unserem Fall zufällig), y ist die Anzahl der Differenzvektoren, die für die Perturbation von x verwendet wird und z bezeichnet die Methode, die für das Crossover verwendet wird (exp: exponentiell, bin: binomial). In dem hier verwendeten Fall DE/rand/1/bin wird die DE also durch einen zufällig gewählten Vektor \vec{X}_{r_1} und einem skalierten Differenzvektor $F \cdot (\vec{X}_{r_2,G} - \vec{X}_{r_3,G})$ und durch binomiales Crossover ausgeführt.

Schritt 1: Festlegen der Kontrollparameter F und Cr und der Populationsgröße N .

Schritt 2: Setzen der ersten Generation $G = 0$ und Initialisierung der Population von N Individuen $P_G = \{\vec{X}_{1,G}, \dots, \vec{X}_{N,G}\}$ mit $\vec{X}_{i,G} = [\vec{x}_{1,i,G}, \vec{x}_{2,i,G}, \dots, \vec{x}_{D,i,G}]$ und davon jedes Individuum i gleichförmig verteilt im Bereich $[\vec{X}_{min}; \vec{X}_{max}]$ mit $\vec{X}_{min} = \{\vec{x}_{1,min}, \vec{x}_{2,min}, \dots, \vec{x}_{D,min}\}$ und $\vec{X}_{max} = \{\vec{x}_{1,max}, \vec{x}_{2,max}, \dots, \vec{x}_{D,max}\}$ und $i = [1, 2, \dots, N]$.

Schritt 3: WHILE Stop-Kriterium nicht erfüllt
DO
FOR $i = 1$ TO N für jedes Individuum sequenziell

Schritt 3.1: Mutation
Generieren eines Donor-Vektors $\vec{V}_{i,G} = \{\vec{v}_{1,i,G}, \vec{v}_{2,i,G}, \dots, \vec{v}_{D,i,G}\}$ für den i -ten Target-Vektor $\vec{X}_{i,G}$ durch das differentielle Mutationsschema der DE:

$$\vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F \cdot (\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G})$$

Schritt 3.2: Crossover
Generieren eines Trial-Vektors $\vec{U}_{i,G} = \{\vec{u}_{1,i,G}, \vec{u}_{2,i,G}, \dots, \vec{u}_{D,i,G}\}$ für den i -ten Target-Vektor $\vec{X}_{i,G}$ durch binominales Crossover:

$$\vec{u}_{j,i,G} = \begin{cases} \vec{v}_{j,i,G} & \text{falls } (rand_{i,j}[0,1] \leq Cr \text{ oder } j = j_{rand}) \\ \vec{x}_{j,i,G} & \text{sonst} \end{cases}$$

END FOR

SIMULATION
FOR $i = 1$ TO N für jedes Individuum sequenziell

Schritt 3.3: Selektion
Auswerten des Trial-Vektors $\vec{U}_{i,G}$:
IF $f(\vec{U}_{i,G}) \leq f(\vec{X}_{i,G})$
THEN $\vec{X}_{i,G+1} = \vec{U}_{i,G}$
ELSE $\vec{X}_{i,G+1} = \vec{X}_{i,G}$
END IF

END FOR

Schritt 3.4: Erhöhen des Generationszählers
 $G = G + 1$

END WHILE

Tabelle 4.7. – Pseudo-Code zur Darstellung des DE-Algorithmus (nach [20], [101])

4.4.4. Anwendung und Problemstellung

Die DE wird klassischerweise für Such- und Optimierungsprobleme verschiedenster Arten angewendet. Nach [12] ist die Wahl einer geeigneten Lösungsstruktur häufig ein manueller,

wissensintensiver Prozess, der durch das iterative Vorgehen eines Experten mit Systemwissen und Versuch und Irrtum erreicht wird. Der Einsatz von EA und hier speziell DE stellt ein robustes Such- und Optimierungsverfahren dar und übertrifft die Leistungsfähigkeit lokaler Verfahren. Besonders bei komplexen, multimodalen, diskontinuierlichen, kombinatorischen und verrauschten Problemen sowie Problemstellungen mit eingeschränktem analytischem Systemwissen sind EA eine sehr effiziente und effektive Methode. Das hier verfolgte Ziel ist die Optimierung und das Auffinden von Parameterwerten gewünschter Szenarien im Bereich des manöverbasierten Testens (siehe Abbildung 4.24). Die manuelle Ermittlung von

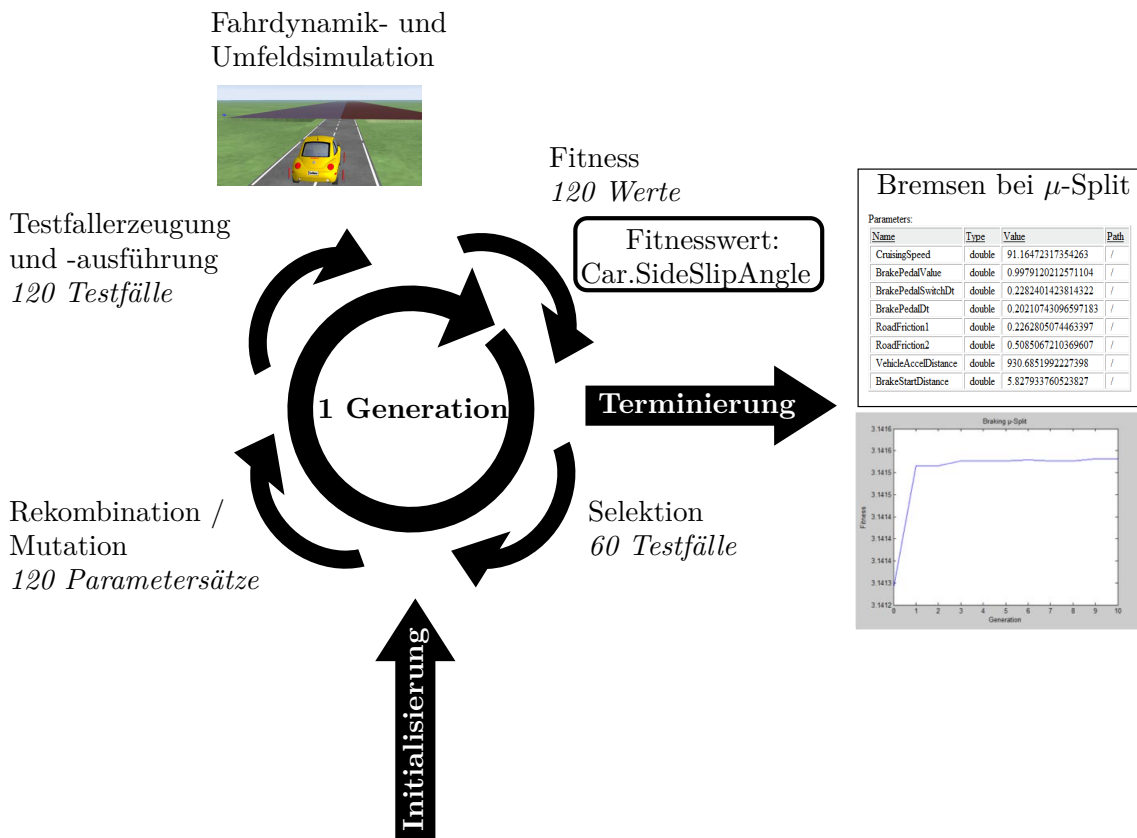


Abbildung 4.24. – Nach der Initialisierung werden durch Mutation und Rekombination neue Parametersätze erstellt, die anschließend in der Simulationssoftware ausgeführt werden. Die Testergebnisse werden mit dem zu optimierenden Fitnesswert bzw. -formel verglichen und anschließend werden die Parametersätze nach den bestpassenden selektiert. Nach einigen Generationen bzw. wenn eine Stagnation der Fitnesswerte eintritt, kann der Simulationsablauf terminiert werden und man erhält einen optimierten Parametersatz.

Parameterwerten zur Eingabe in die Simulation, um ein gewünschtes Testfall- bzw. Szenarienverhalten zu erreichen, ist eine mühsame Angelegenheit, die sehr viel fundiertes Wissen im Bereich der Fahrdynamik- und Umfeldsimulation erfordert. Auch eine mögliche Modellsimulation stellt einen komplexen und aufwändigen Weg dar, der viel Vorbereitungszeit und Kenntnis des Systems herausfordert. Die nachfolgende Kombination des manöverbasierten Testens bzw. der manöverbasierten Fahrdynamik- und Umfeldsimulation mit DE zeigt einen schnellen und vergleichsweise einfachen Weg, wie Parametersätze zur Szenariendefinition ermittelt werden können. Anders als in vergleichbaren Veröffentlichungen wie [76] werden

in dieser Arbeit nicht Testsequenzen für Automaten oder Auszeichnungssprachen optimiert, sondern die Anwendung auf Fahrdynamik- und Umfeldsimulation propagiert. Des Weiteren ist der Einsatz hier im Systemtest angesiedelt.

Das manöverbasierte Testen, das durch Fahrdynamik- und Umfeldsimulation durchgeführt wird, stellt einen wichtigen Schritt im Bereich des Systemtests von Steuergeräten im Fahrzeug dar. Durch die Simulation kann in Kombination mit einem Hardware-in-the-Loop-Prüfstand, der das Steuergerät in eine fahrzeugähnliche Umgebung versetzt, das System in seiner späteren Einsatzumgebung getestet werden bereits bevor ein Erprobungsfahrzeug zur Verfügung steht. Somit dient diese Methode zur frühen und kostengünstigen Auffindung von Fehlern. Wichtig ist beim manöverbasierten Testen, dass kritische Szenarien sowie Grenzfälle getestet werden, um die Hauptfehlerquellen schnell zu identifizieren. Des Weiteren ist es z. B. beim Test von integralen Fahrzeugsicherheitssystemen häufig sinnvoll, Fahrdynamiksimulation und Crashdateneinspeisung zu kombinieren. Dies bedeutet, dass der Zustand der Daten der Fahrdynamiksimulation zum Aufprallzeitpunkt genau zu den vorhandenen Crashdaten passt, sodass keine Inkonsistenz entsteht, die das Steuergerät in einen Fehlerzustand versetzt. Dieser Ablauf wird in Kapitel 5 noch genauer dargelegt. Beim Aufsetzen der Testszenarien in der Simulationsumgebung müssen also genau solche essentiellen Fälle mit präzisen Simulationsdaten zu exakten Zeitpunkten erstellt werden, was ein nicht einfaches Unterfangen ist. Die hier vorhandene Problemstellung resultiert daraus, dass durch die schiere Unendlichkeit der Simulationsmöglichkeiten der Fahrdynamik- und Umfeldsimulation das „Raten“ oder „Ausprobieren“ von Parametern zur Eingabe in die Simulationssoftware äußerst ineffektiv und ungenau zur Auffindung von gewünschten Szenarien ist. Des Weiteren ist auch eine sehr gute Kenntnis im Bereich der Fahrdynamik Voraussetzung zur Parametrierung von Testszenarien, um die Erstellung eines solchen Szenarios zu bewältigen. Aus diesem Grund ist die Kombination von evolutionären Optimierungsalgorithmen wie der DE eine effektive und sinnvolle Ergänzung zum manöverbasierten Testen. Abbildung 4.25 zeigt den prinzipiellen Vorgang der Parameteroptimierung und anschließende Einspeisung in ein Steuergerät exemplarisch.

4.4.5. Parametrierung

Die Festlegung der Kontrollparameter hat einen wichtigen Einfluss auf die Effektivität und Genauigkeit des Optimierungsergebnisses. Insgesamt gibt es im hier beschriebenen Fall drei Kontrollparameter: den Mutationsskalierungsfaktor F , die Crossoverrate Cr und die Populationsgröße N der Generationen. Nach [128] ist bei einem Problem im Bereich Dimensionsbereich D von 5-10, wie es hier vorliegt, ein Wert von $F = 0,5$ ein guter Initialwert. Der effektive Bereich von F liegt normalerweise im Intervall $[0,4; 1]$ [20]. Eine Parameterstudie nach [34] schlägt für ein 5- D -Problem einen Wert von 0,4 für Cr vor. Begründet wird die Aussage dadurch, dass ein kleinerer Wert (nahe 0) dazu führt, dass das Optimum gar nicht oder erst nach sehr vielen Generationen gefunden wird. Ein sehr hoher Wert (nahe 1) für Cr dagegen führt zu großer Diversität. Durch die vorherrschende sehr hohe Crossoveraktivität kann die fortfolgende Veränderung der Population nicht in

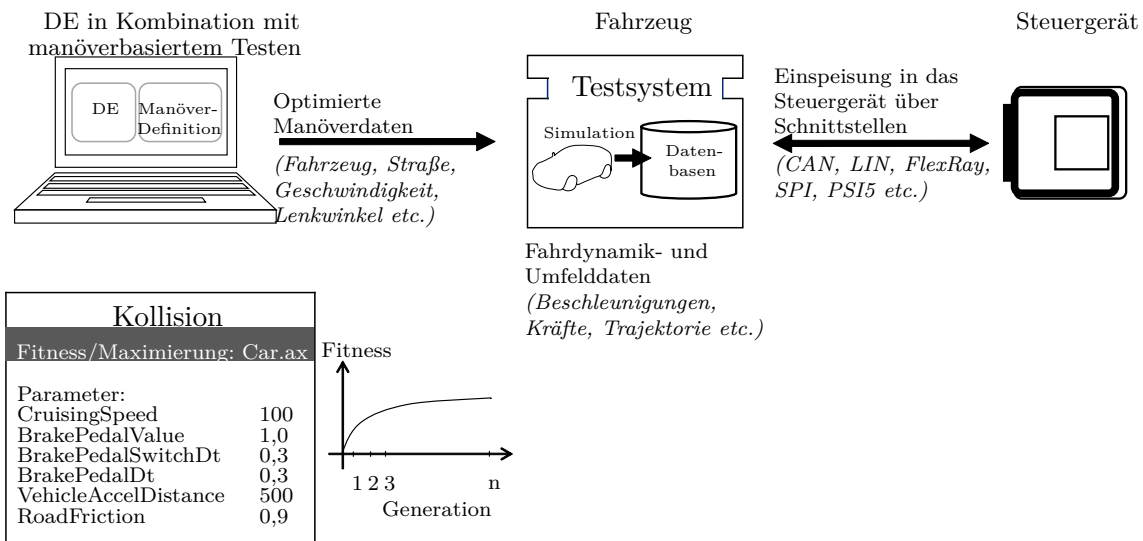


Abbildung 4.25. – Die Kombination aus DE und manöverbasiertem Testen wird zunächst als reine Office-Simulation auf einem normalen Rechner durchgeführt. Die zum gewünschten Verhalten führenden, optimierten Parameter werden anschließend auf ein Testsystem transferiert und die Simulationsdaten über verschiedene Schnittstellen in das Steuergerät eingespeist [106].

Richtung des Optimums konvergieren, da die Individuen mutieren bevor sie das Minimum erreichen. Die Auswahl der Populationsgröße stellt selbst heute noch eine schwierige Frage dar. Eine Grundregel, die in [127] vorgeschlagen wird, ist die Festlegung nach der Größe der Dimension mit $N = 10 \cdot D$. In Anhang B wird ein Vergleich zwischen der Ausführung ohne und mit vorheriger statistischer Analyse (siehe auch Kapitel 4.3) und der Einfluss von der Kontrollparameter Cr und F diskutiert. Dies wird anhand des nachfolgenden Beispiels gezeigt. Dort zeigt sich, dass der Algorithmus im hier angewandten Fall in erster Linie empfindlich gegen einen zu hohen Wert für F und für einen zu niedrigen Wert für Cr ist. Nachfolgend sollen nun zwei Beispiele aufzeigen, wie Experimente im Bereich der DE in Kombination mit manöverbasiertem Testen verlaufen. Ziel der Beispiele ist, durch Optimierung mit der DE schon nach wenigen Generationen Tendenzen für die Werte der einzelnen Parameter aufzuzeigen, um das gewünschte Simulationsergebnis bzw. Fitnesswert zu erreichen.

4.4.6. Beispiel 1: Eisbremsung

Dieses Beispiel wurde schon in Kapitel 4.2.3 vorgestellt (siehe auch Tabelle 4.3) und durch statistische Versuchsplanung in Kapitel 4.3.3 analysiert. Der Aufbau von Beispiel 1 für die Optimierung durch DE ist in Tabelle 4.8 aufgezeigt. Es handelt sich um ein Bremsmanöver, das auf einem Straßenabschnitt mit variablem, niedrigem Reibwert begonnen wird. Somit handelt es sich um eine Eisbremsung. Veränderbare Parameter sind ebenfalls die Betätigung des Bremspedals hinsichtlich der Pedaldruckgeschwindigkeit und der Druckstärke. Als feste Werte werden wie in Kapitel 4.3.3 begründet, die Anfahrtsgeschwindigkeit und die Pedalwechselzeit vorgegeben. Das Ziel des Experiments soll es nun sein, ein möglichst


Manöver:	Gefahrenbremsung			
Ablauf:				
Parameter:	<i>Name</i>	<i>Einheit</i>	<i>Grenzen</i>	<i>Bedeutung</i>
	Brake	[-]	0,5-1,0	Drückniveau
	BrakePedalDt	[s]	0,1-0,6 s	Niveauerreichung
	RoadFriction	[-]	0,1-0,4	Reibwert Bremsfläche (Eis)
Fitness:	Verzögerung, Deceleration von $-1,9\text{m/s}^2$ IF $ (\text{Deceleration}(\vec{U}_{i,G}) + 1,9) \leq (\text{Deceleration}(\vec{X}_{i,G}) + 1,9) $ THEN $\vec{X}_{i,G+1} = \vec{U}_{i,G}$ (Kindindividuum) ELSE $\vec{X}_{i,G+1} = \vec{X}_{i,G}$ (Elternindividuum) END IF			
Kontrollparameter:	Parameteranzahl	$D = 3$		
	Skalierungsfaktor	$F = 0,5$		
	Mutationsrate	$Cr = 0,4$		
	Populationsgröße	$N = 30$		
	Versuchsgröße	$2 \cdot N = 60$ (je 30 Eltern und 30 Kinder)		
	Anzahl Generationen	$G_{max} = 100$		

Tabelle 4.8. – Aufbau Experiment 1

genaues Ergebnis zu erhalten, das einen Grenzfall im Algorithmus darstellt. Gewünscht ist eine Verzögerung von genau $1,9\text{m/s}^2$, da dies die Abgrenzung einer normalen Bremsung und einer Eisbremsung darstellt.

Zunächst soll aber anhand des ausgewählten Experiments dargestellt werden, wie die Mutation und Rekombination konkret vonstatten geht. Der erste Schritt ist das Generieren des Donor-Vektors durch drei zufällig ausgewählte Elternparametersätze durch das differentielle Mutationsschema der DE aus Abbildung 4.22. Es werden drei verschiedene Zufallszahlen r_1^i , r_2^i und r_3^i erzeugt, die als Index für die Auswahl von drei Parametersätzen aus der Elterngeneration dienen. Diese drei Parametersätze erzeugen dann den Donor-Vektor $\vec{V}_{i,G}$ nach folgender exemplarischer Berechnung:

$$\begin{array}{rclclcl}
 \vec{X}_{r_1^i, G}^+ & F \cdot & (\vec{X}_{r_2^i, G}^- & \vec{X}_{r_3^i, G}) & = & \vec{V}_{i, G} \\
 \text{Brake [-]} & 0,8 + & 0,5 \cdot & (0,5 - & 0,7) & = 0,7 \\
 \text{BrakePedalDt [s]} & 0,1 + & 0,5 \cdot & (0,2 - & 0,2) & = 0,1 \\
 \text{RoadFriction [-]} & 0,1 + & 0,5 \cdot & (0,3 - & 0,2) & = 0,15
 \end{array}$$

Dieser entstandene Donor-Vektor wird nun nach dem Schema aus Abbildung 4.23 so mit dem Target-Vektor $\vec{X}_{i,G}$ kombiniert, dass der Trial-Vektor entsteht. Pro Parameter wird eine Zufallszahl $rand_{i,j} [0; 1]$ erzeugt, die dann mit Cr verglichen wird. Ist die Zufallszahl kleiner oder gleich, so wird der Parameterwert aus $\vec{V}_{i,G}$ übernommen, ist er größer, bleibt der ursprüngliche Wert von $\vec{X}_{i,G}$ bestehen:

	$rand_{i,j} [0; 1]$	Cr	$\vec{X}_{i,G}$	$\vec{V}_{i,G}$	$\vec{U}_{i,G}$
Brake [-]	0,4	$\leq 0,4 \Rightarrow$	0,5	<u>0,7</u>	0,7
BrakePedalDt [s]	0,9	$> 0,4 \Rightarrow$	<u>0,1</u>	0,1	0,1
RoadFriction [-]	0,3	$> 0,4 \Rightarrow$	<u>0,3</u>	0,15	0,3

Das zu erwartende Ergebnis für den Parameter RoadFriction ist in diesem Testfall trivial und leicht zu errechnen. Dies soll zur Verifizierung des Algorithmus herangezogen werden. Nach [80] ist die Abbremsung eines Fahrzeugs mit folgender Formel zu berechnen:

$$\text{Abbremsung } z = \frac{-\ddot{x}}{g} \quad (4.11)$$

Hierbei stellt $-\ddot{x}$ die longitudinale Verzögerung und g die Erdbeschleunigung dar. Nach [80] ist höchste von einem Fahrzeug mit idealer Bremsanlage erreichbare Abbremsung:

$$\text{Abbremsung } z_{\text{grenz}} = \mu \quad (4.12)$$

Im hier dargestellten Fall, in dem der der Reibwert unter einer Abbremsung von $-1,9\text{m/s}^2$ gesucht wird, bedeutet dies:

$$\mu = \frac{-\ddot{x}_{\text{grenz}}}{g} = \frac{-1,9\text{m/s}^2}{-9,81\text{m/s}^2} = 0,19 \quad (4.13)$$

Die Simulationsergebnisse des gesamten Testablaufs sind in Tabelle 4.9 dargestellt. Es zeigt sich, dass die Simulation den in Formel 4.13 erwarteten Wert für RoadFriction liefert. Insgesamt wurden 100 Generationen mit je 60 Individuen durchgeführt. Deutlich wird, dass bereits nach 16 Generationen ein Ergebnis sehr nahe null, d. h. am gewünschten Wert von $-1,9\text{m/s}^2$ für die Verzögerung eintritt. Des Weiteren ist hier anzumerken, dass 100 durchgeführte Generationen keine hohe Anzahl sind. Um das globale Optimum für die Parameter zu finden, wäre eine wesentlich höhere Anzahl an Generationen notwendig. Hervorzuheben ist, dass selbst nach einer relativ geringen Anzahl von Generationen bereits ein guter Fitnesswert bzw. zumindest ein lokales Optimum erreicht wird und eine Tendenz der einzelnen Parameter deutlich wird, die ebenfalls in Tabelle 4.9 dargestellt wird. Mithilfe der Tendenzen können die Szenarien noch einfacher analysiert und erstellt werden. Auch ein Abbruchkriterium ist definierbar, das bei einer gewünschten Ergebnisgüte wie zum Beispiel einem vorgegebenen Wert für die Längsbeschleunigung nach jeder Generation das

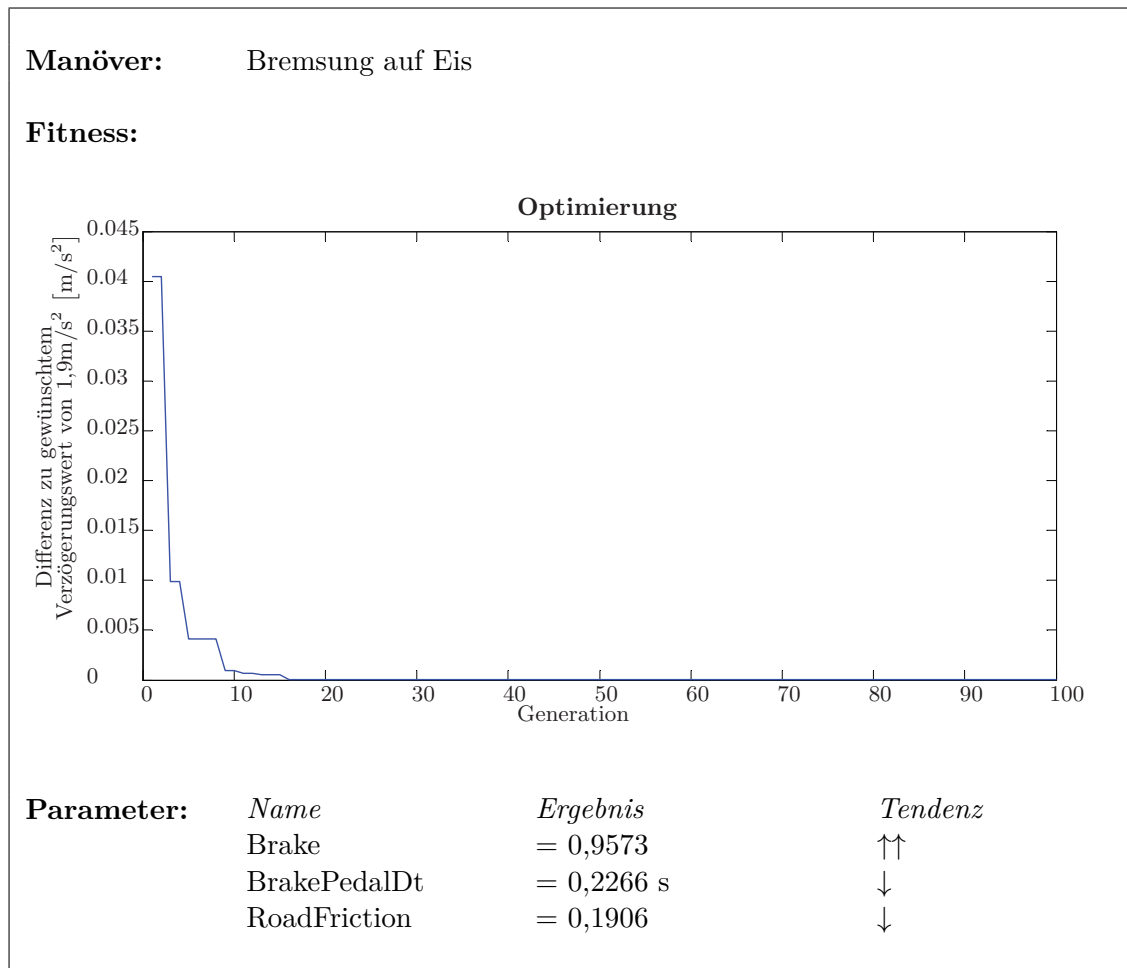


Tabelle 4.9. – Ergebnisse des Experiments Bremsen auf Eis nach 100 Generationen [101]

aktuelle Ergebnis mit dem Gütewert vergleicht und dann bei Erreichen die Simulation beendet, um Zeit zu sparen.

Zum besseren Verständnis der Dynamik des Algorithmus sollen nun noch Auswertungen hinsichtlich der Entstehung der Ergebnisse erfolgen. Hierfür wird zunächst in Abbildung 4.26 aufgezeigt, welche Parameterwerte jeweils das bestangepasste Individuum darstellen und wie häufig das bestangepasste Individuum (Minimum der Fitnessfunktion) gewechselt wurde. Es zeigt sich in Abbildung 4.26a, dass die Dynamik der Parameter nach circa 50 Generationen stark abnimmt. Die Werte der Parameter haben sich quasi eingependelt. Es zeigt sich auch, dass vor allem der Wert für RoadFriction, der ja nach Kapitel 4.3.3 den größten Einfluss hat, sich bereits innerhalb der ersten fünf Generationen so einpendelt, dass er sich kaum noch verändert. Danach erfolgt noch die Optimierung der Parameter BrakePedalDt und Brake. Abbildung 4.26b zeigt, dass auch nach Generation 50 noch besser angepasste Individuen gefunden werden, d. h. es werden immer noch Eltern durch ihre Kinder ersetzt.

Die Dynamik der Parameter zeigt sich auch in Abbildung 4.27, wo sowohl der Wertekorridor, in dem sich die Parameter bewegen, als auch die durchschnittlichen Werte für die

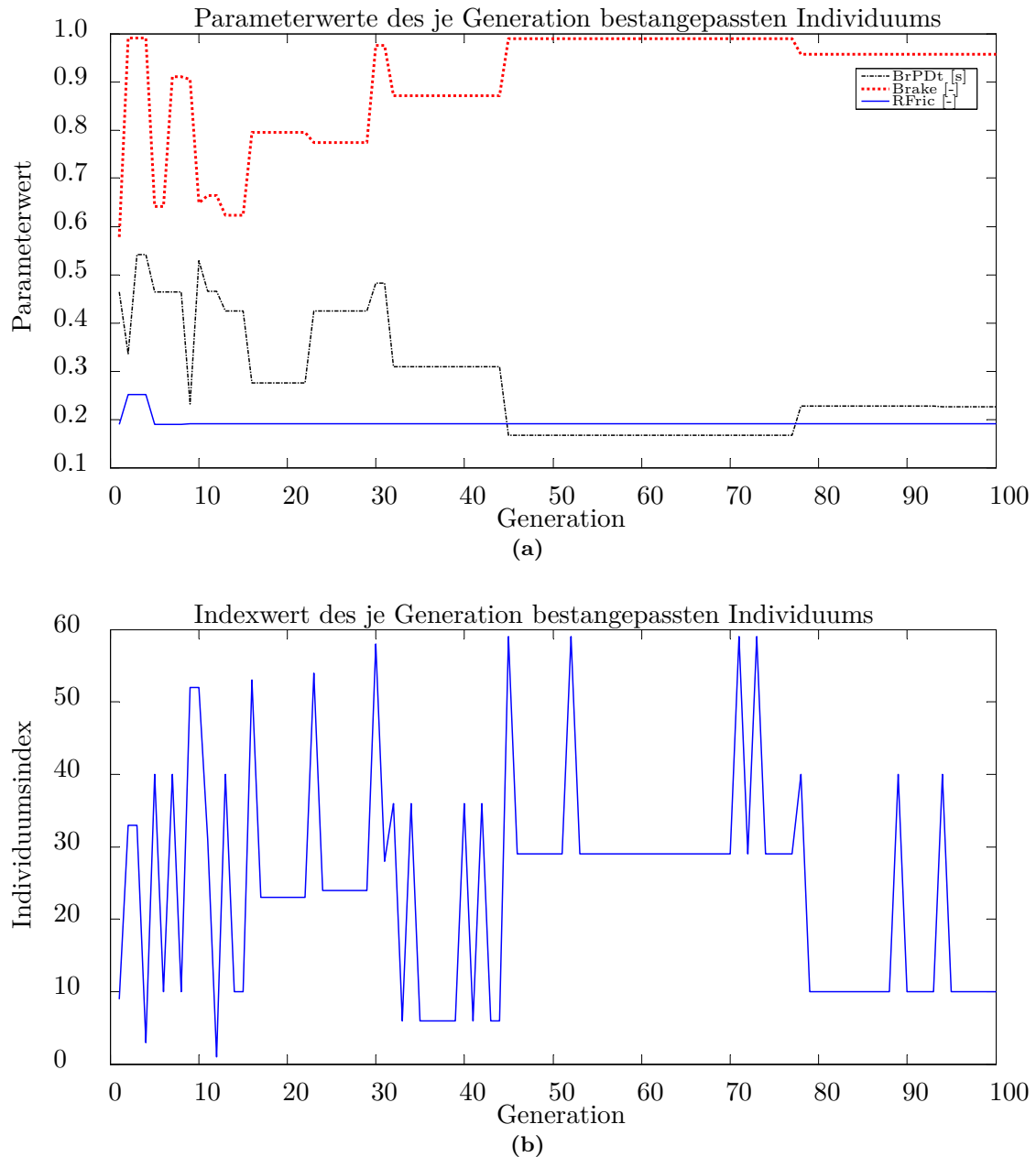


Abbildung 4.26. –

(a) Darstellung der Veränderung der Parameterwerte der jeweils minimalen Individuen hinsichtlich der Fitnessfunktion je Generation.

(b) Darstellung der Indizes der jeweils minimalen Individuen hinsichtlich der Fitnessfunktion je Generation mit 60 Einzelindividuen (30 Eltern, 30 Kinder). Jede Spitze stellt einen Wechsel von Eltern- zu Kindindividuum, also eine Optimierung, dar.

Deutlich wird, dass auch nach der bereits sehr gut optimierten Generation 16 (siehe Tabelle 4.9) weiterhin eine Optimierung stattfindet. Die Dynamik der Parameter nimmt jedoch nach circa 50 Generationen stark ab [101].

Parameter aufgezeigt werden. Abbildung 4.27a verdeutlicht, dass die Parameterwertkorridore, d. h. der maximale und minimale Wert der drei Parameter für eine Generation mit 60 Individuen für BrakePedalDt und Brake weit geöffnet bleibt, für RoadFriction jedoch nach circa 23 Generationen sehr eng wird. Dies gibt wieder, dass der optimale

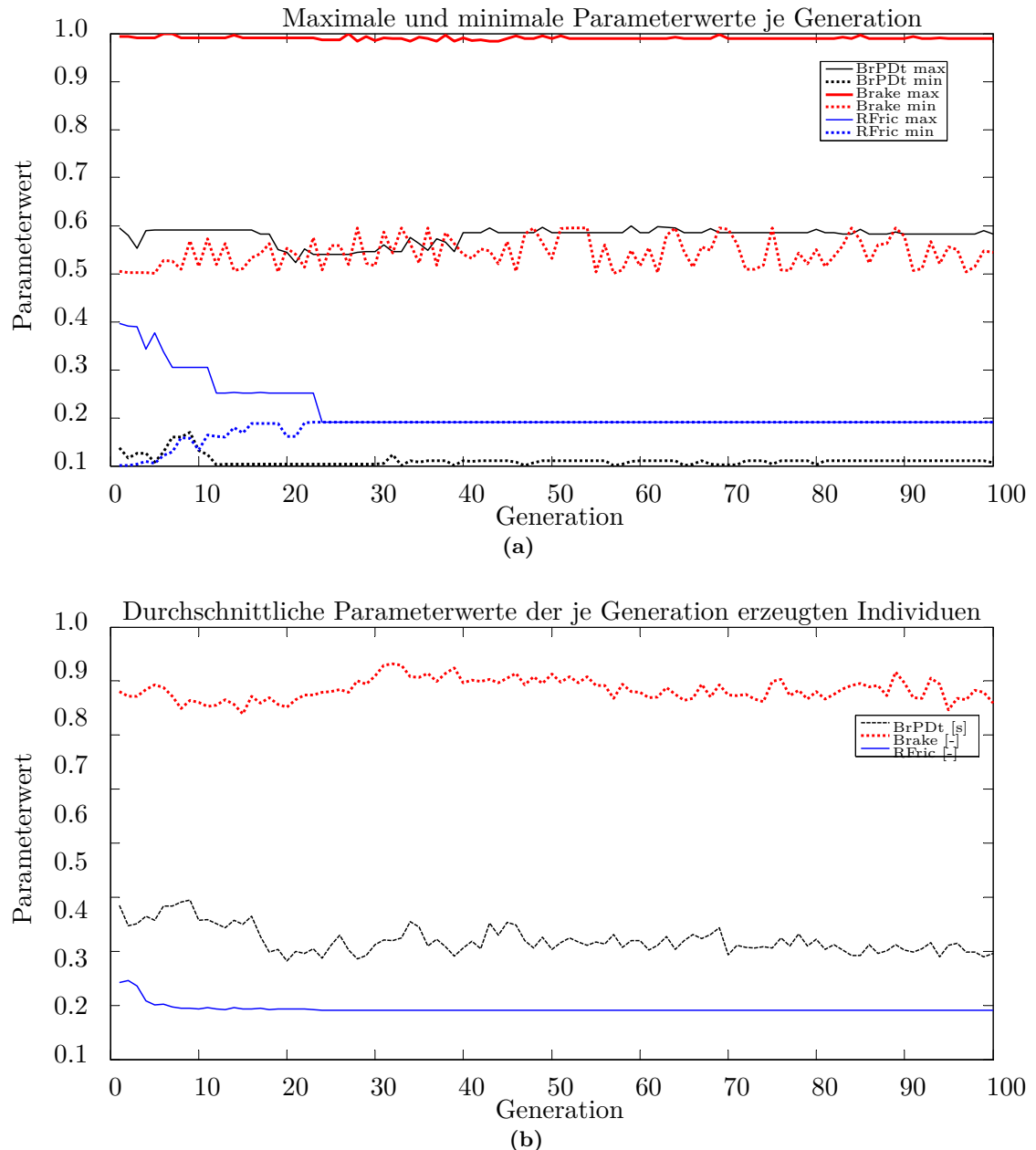


Abbildung 4.27. –

(a) Darstellung der jeweils maximalen und minimalen Parameterwerte, die in den 60 Versuchssindividuen pro Generation auftraten.

(b) Darstellung der durchschnittlichen Parameterwerte aller Individuen pro Generation mit 60 Einzelindividuen (30 Eltern, 30 Kinder).

Deutlich wird auch hier, dass sich der Wert für RoadFriction sehr schnell einpendelt und kaum noch variiert. Die Parameterwerte für BrakePedalDt und Brake dagegen verändern sich auch in Generation 100 noch stark. Dies lässt darauf schließen, dass ein optimierter Wert für RoadFriction relativ leicht zu finden ist, die Werte für BrakePedalDt und Brake jedoch nicht einfach auffindbar und auf einander abstimmbare sind. Da der Einfluss der beiden Größen jedoch relativ gering ist, kann auch ohne optimales Zusammenspiel der Parameter ein sehr gutes Ergebnis erzielt werden.

Wert für RoadFriction relativ einfach zu finden ist, die beiden anderen Parameter jedoch nicht so einfach optimiert werden können. Da der Einfluss von RoadFriction jedoch sehr

groß und der von BrakePedalDt und Brake nach Kapitel 4.3.3 verhältnismäßig klein ist, kann trotzdem ein sehr gutes Fitnessergebnis erzielt werden. Abbildung 4.27b verdeutlicht diese Schlussfolgerung weiter: während der Durchschnittswert über alle 60 Individuen einer Generation für RoadFriction sich nach Generation 23 kaum noch verändert, variieren die Durchschnittswerte für BrakePedalDt und Brake auch bei Generation 100 noch.

Ein interessanter Aspekt ist noch die Zeitdauer des Suchalgorithmus. Bei 100 Generationen und einer Versuchsgröße von 60 Individuen ergibt sich eine Testfallanzahl von 6000 Einzelversuchen. Die Simulationsdauer pro Einzelversuch beträgt circa drei Sekunden, was sich auf eine Simulationsdauer von fünf Stunden kumuliert. Da die Einzelindividuen einer Generation unabhängig von einander berechnet werden können, kann die Simulationszeit im besten auf 5s pro Generation verringert werden, was eine Gesamtdauer von $5s \cdot 100 = 500s$ bzw. rund 8 Minuten ergibt.

Der optimierte Testfall kann nun in das Testobjekt reversibler Gurtstraffer (siehe Tabelle 4.3) eingespeist werden. Das Ziel ist, die richtige Umsetzung der Anforderungen (siehe Tabelle 4.2) zu testen. Hierfür wird der Simulationsablauf eingespeist und die Auslösezeiten, also die Reaktion des Systems, gemessen. Abbildung 4.28 zeigt diesen Vorgang. Anhand

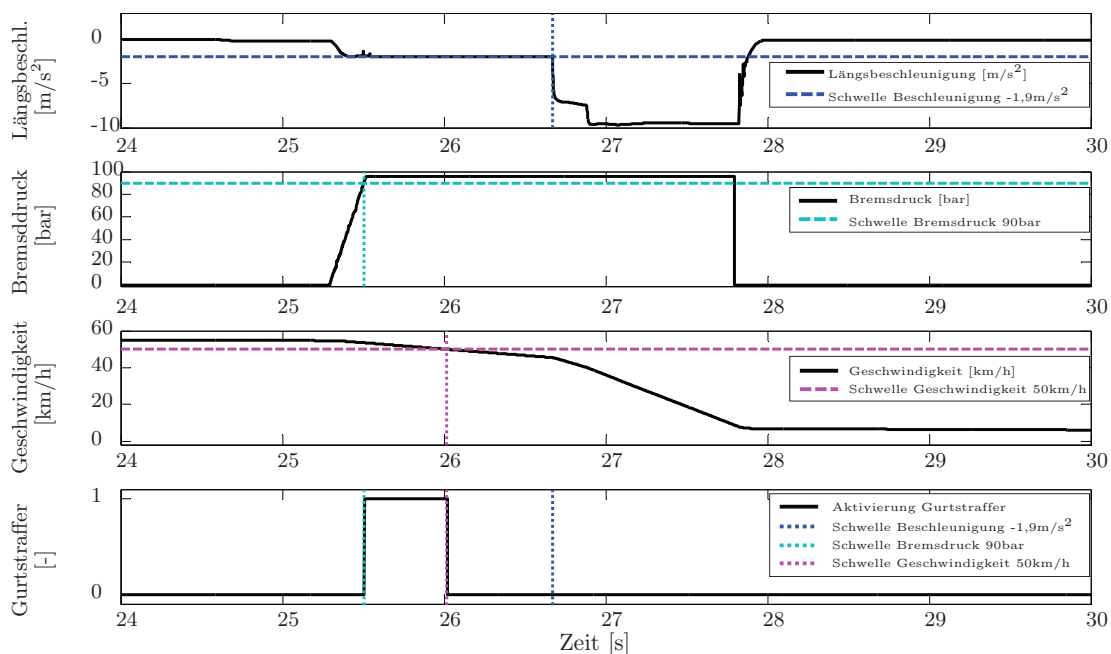


Abbildung 4.28. – Einspeisung der optimierten Simulationsparameter in das System zum Test der Funktion reversibler Gurtstraffer. Die Auslöseschwelle ergibt sich aus den Anforderungen in Tabelle 4.2. Die Über- oder Unterschreitung einer Anforderung ist jeweils durch eine vertikale Linie dargestellt. Deutlich wird, dass der reversible Gurtstraffer richtig auslöst: Solange die Anforderungen hinsichtlich Längsbeschleunigung, Bremsdruck und Geschwindigkeit erfüllt sind, löst er aus. Sobald eine Anforderung nicht mehr erfüllt ist (in diesem Fall fällt durch den Bremsvorgang die Geschwindigkeit unter 50km/h), geht das Auslösekriterium des Gurtstraffers auf null. Die Funktion ist also richtig implementiert [101].

der Einspeisung der optimierten Simulationsparameter kann also ein Grenzfall getestet werden, der die richtige Implementierung der Funktionsanforderungen aufzeigt.

Abschließend kann als Fazit für das Beispiel gezogen werden, dass für eine tiefgehende Optimierung weitere Generationen möglich wären, jedoch aufgrund des geringen Einflusses von BrakePedalDt und Brake ein ausreichend gutes Ergebnis bereits nach unter 25 Generationen erreicht wird. Durch die Vorgabe eines Abbruchkriteriums, d. h. der Vorgabe eines geeigneten Gütekriteriums, kann hier also 75% der Simulationszeit eingespart werden. Durch die vorhergehende Analyse durch die Statistische Versuchsplanung konnte bereits an einem voroptimierten Punkt gestartet werden, sodass hier Zeit und Ressourcen eingespart werden konnten. Die Simulationsdauer pro Einzelversuch beträgt circa fünf Sekunden, was sich sequentieller Ausführung auf eine Simulationsdauer von acht Stunden und 20 Minuten kumuliert. Da die Individuen einer Generation in der Simulation jedoch unabhängig von einander berechnet werden können, lassen sich die 60 Individuen parallelisieren, was die Berechnungszeit für eine Generation von $5s \cdot 60 = 300s$ eine Reduktion um ein Vielfaches bzw. im besten Fall auf 5s Simulationszeit pro Generation reduziert. Für 100 Generationen wäre so theoretisch eine Gesamtsimulationszeit von $5s \cdot 100 = 500s$ möglich. Grundsätzlich hätte die Simulationszeit mithilfe eines gütebezogenen Abbruchkriteriums weiter reduziert werden können. Das heißt, wenn die Qualität des Ergebnisses für den Einsatzzweck ausreichend ist, wird die Simulation beendet. Die Methode der DE konnte hier in kurzer Zeit einen Testfall erzeugen, der ein per Hand schwer zu erstellendes Szenario darstellt. Neben dem Parametersatz erhält man auch weiteren Einblick in die Dynamik und das Verhalten des Testfalls selbst und könnte nun ohne große Probleme weitere Testfälle mit gewünschten Eigenschaften wie z. B. der Maximierung oder Minimierung einer Größe oder der Erstellung von Grenzfällen erzeugen.

4.4.7. Beispiel 2: μ -Split-Bremmung

Ein zweites Beispiel soll nun die Optimierung mit einer größeren Anzahl an Parametern und ohne vorherige Analyse durch statistische Versuchsplanung aufzeigen. Das Ziel des Experiments soll es nun sein, eine möglichst lateral kritische Situation hervorzurufen, die zum Beispiel durch eine hohe Schwimmwinkelgeschwindigkeit ausgezeichnet ist. Der Schwimmwinkel ist der Winkel zwischen der Geschwindigkeit des Fahrzeugschwerpunkts und der Fahrzeugmittellinie [79]. Abbildung 4.29 zeigt den Gierwinkel, den Schwimmwinkel und das Kurvenverhalten von Fahrzeugen graphisch anhand des Linearen Einspurmodells. Die x-Achse des ortsfesten Koordinatensystems und die Geschwindigkeit v im Schwerpunkt SP des Fahrzeugs bilden den Gierwinkel ψ und die Geschwindigkeit v im Schwerpunkt und die Fahrzeuglängsachse bilden den Schwimmwinkel β . Der Gierwinkel und der Schwimmwinkel zusammen bilden den Kurswinkel, d.h. $\text{Kurswinkel} = \beta + \psi$. Grundsätzlich kann man sagen, dass das vom Fahrer durch Lenkradeinschlag gewünschte Kurvenverhalten durch die Gierrate bzw. Giergeschwindigkeit $\dot{\psi}$ abgebildet wird. Sie ist ein Maß für die Drehwinkelgeschwindigkeit des Fahrzeugs um die Hochachse [45]. Folgt das Fahrzeug dem Lenkradeinschlag bzw. dem Trajektorienwunsch des Fahrzeugs perfekt, so spricht man von einem neutralsteuernden Verhalten. Untersteuern bezeichnet, wenn das Fahrzeug eine kleinere Gierrate umsetzt, als der Fahrer wünscht. Das Fahrzeug dreht also weniger als es

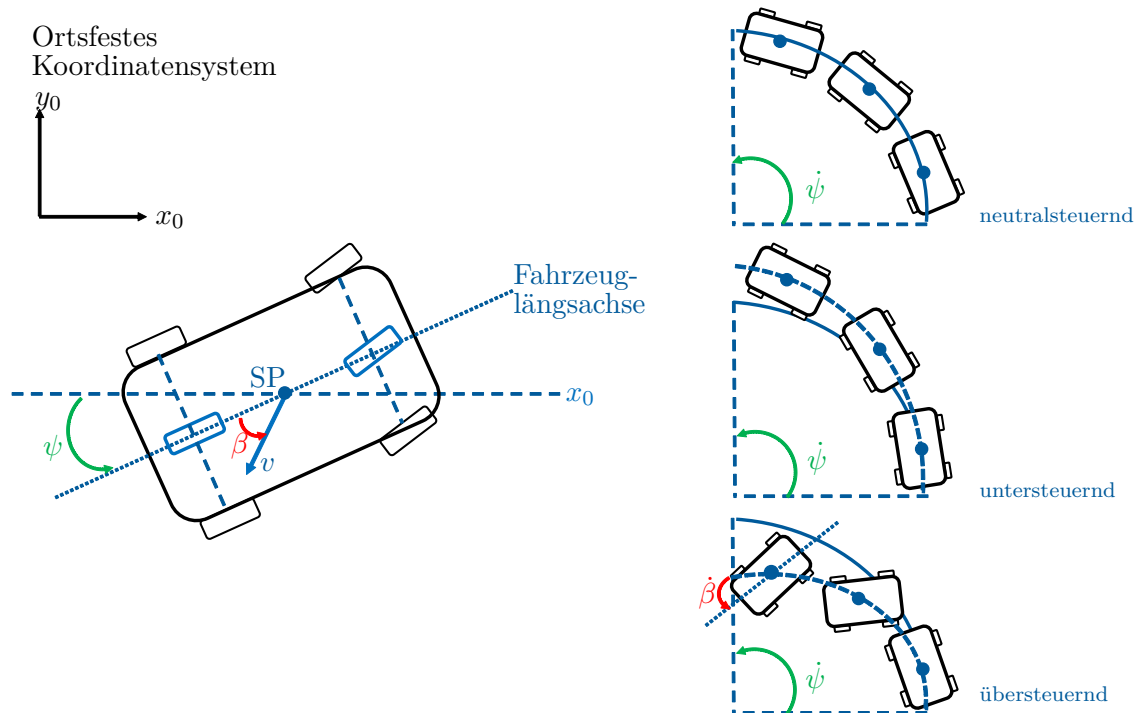


Abbildung 4.29. – Darstellung des Gierwinkels ψ , des Schwimmwinkels β und des Kurvenverhaltens von Fahrzeugen anhand des Linearen Einspurmodells unter Berücksichtigung der Giergeschwindigkeit $\dot{\psi}$ und der Schwimmwinkelgeschwindigkeit $\dot{\beta}$ (nach [79] und [113]).

soll. Das am schwersten beherrschbare Verhalten ist das sogenannte Übersteuern. Hierbei dreht sich das Fahrzeug mehr als der Fahrer möchte, es „bricht aus“. Hierbei entsteht der Schwimmwinkel β bzw. die Schwimmwinkelgeschwindigkeit $\dot{\beta}$: sie gibt an, um wie viel bzw. wie schnell sich das Fahrzeug um die Hochachse dreht, ohne dass dies durch Lenkradeinschlag vorgegeben ist. Der Schwimmwinkel bzw. die Schwimmwinkelgeschwindigkeit kann also als Maß für die querdynamische Kritikalität eines Manövers dienen: je größer, desto kritischer.

Ein Schwimmwinkel entsteht also dann, wenn sich das Fahrzeug in eine Richtung bewegt, ohne dass die Fahrzeuglängsachse schnell genug folgen kann. Dies kann z. B. durch das in Tabelle 4.10 beschriebene Beispiel hervorgerufen werden: durch Bremsen auf Flächen mit unterschiedlichen Reibwerten entsteht ein Drehmoment auf das Fahrzeug, das eine Schwimmwinkelgeschwindigkeit erzeugt. Ein interessantes Ziel ist nun, die Schwimmwinkelgeschwindigkeit zu maximieren und somit die Frage zu stellen: was ist der lateral kritischste Fall, der in diesem Szenario entstehen kann? Hier handelt es sich um ein Bremsmanöver, das auf einem Straßenstreifen mit unterschiedlichen Reibwerten links (RoadFriction1) und rechts (RoadFriction2) ausgeführt wird. Veränderbare Parameter sind ebenfalls die Anfahrtsgeschwindigkeit (CruisingSpeed) sowie die Betätigungsstärke des Bremspedals (Brake), die Pedalwechselzeit von Gas zu Bremse (BrakePedalSwitchDt) und die Zeitdauer, die zur Erreichung des Drückniveaus Brake erreicht wird (BrakePedalDt). Des Weiteren sollen auch hier wie in Kapitel 4.4.5 vorgeschlagen $10 \cdot D = 60$ Eltern und somit auch 60 Kindern ausgeführt werden, was eine Versuchsgröße von 120 Individuen ergibt. Die Anzahl

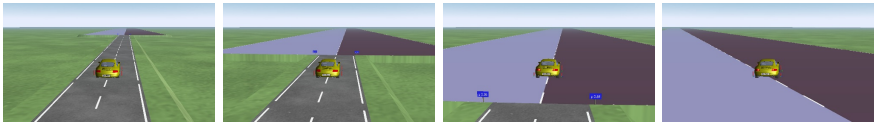
Manöver:	Bremsen auf μ -Split			
Ablauf:				
Parameter:	<i>Name</i>	<i>Einheit</i>	<i>Grenzen</i>	<i>Bedeutung</i>
	CruisingSpeed	[km/h]	5-50 km/h	Anlaufgeschwindigkeit
	Brake	[-]	0,3-1,0	Drückniveau
	BrakePedalSwitchDt	[s]	0,1-0,3 s	Pedalwechselzeit
	BrakePedalDt	[s]	0,1-0,3 s	Niveauzeitpunkt
	RoadFriction1	[-]	0,2-1,0	Reibwert links
	RoadFriction2	[-]	0,2-1,0	Reibwert rechts
Fitness:	Maximale Schwimmwinkelgeschwindigkeit $\dot{\beta}$, Car.SideSlipAngleVel [rad/s] IF Schwimmwinkelgeschw. ($\vec{U}_{i,G}$) \geq Schwimmwinkelgeschw. ($\vec{X}_{i,G}$) THEN $\vec{X}_{i,G+1} = \vec{U}_{i,G}$ (Kindindividuum) ELSE $\vec{X}_{i,G+1} = \vec{X}_{i,G}$ (Elternindividuum) END IF			
Kontrollparameter:	Parameteranzahl	$D = 6$		
	Skalierungsfaktor	$F = 0,5$		
	Mutationsrate	$Cr = 0,4$		
	Populationsgröße	$N = 60$		
	Versuchsgröße	$2 \cdot N = 120$ (je 60 Eltern und 60 Kinder)		
	Anzahl Generationen	$G_{max} = 100$		

Tabelle 4.10. – Aufbau Experiment 2

der Generationen soll 100 betragen.

Das Optimierungsergebnis des Szenarios ist in Tabelle 4.11 dargestellt. In diesem Testfall wurden 100 Generationen mit je 120 Individuen erzeugt. Es zeigt sich, dass bis Generation 32 eine gravierende Optimierung stattfindet, anschließend stagniert das Ergebnis. Auch hier wäre es durchaus möglich, dass in höheren Generationen noch höhere Fitnesswerte auftreten können. Aufgrund des Zieles in diesem Fall, möglichst schnell eine Tendenz für die Parameter zu erkennen, ist es jedoch nicht notwendig, mehr Simulationszeit aufzuwenden. Die Tendenzen für die Parameter sind nach den 100 Generationen bereits deutlich sichtbar. Bezüglich der Gesamtsimulationszeit ist hier die Versuchsgröße von 120 Individuen mit 100 Generationen zu multiplizieren, was eine Gesamttestfallanzahl von 12000 ergibt. Ein Testfall hat eine Simulationszeit von vier Sekunden, was kumuliert eine Simulationszeit von 13 Stunden und 20 Minuten ergibt. Durch Parallelisierung, d. h. Parallelberechnung der Individuen, könnte im besten Fall die Simulationszeit auf $100 \cdot 4s = 400s$ reduziert werden.

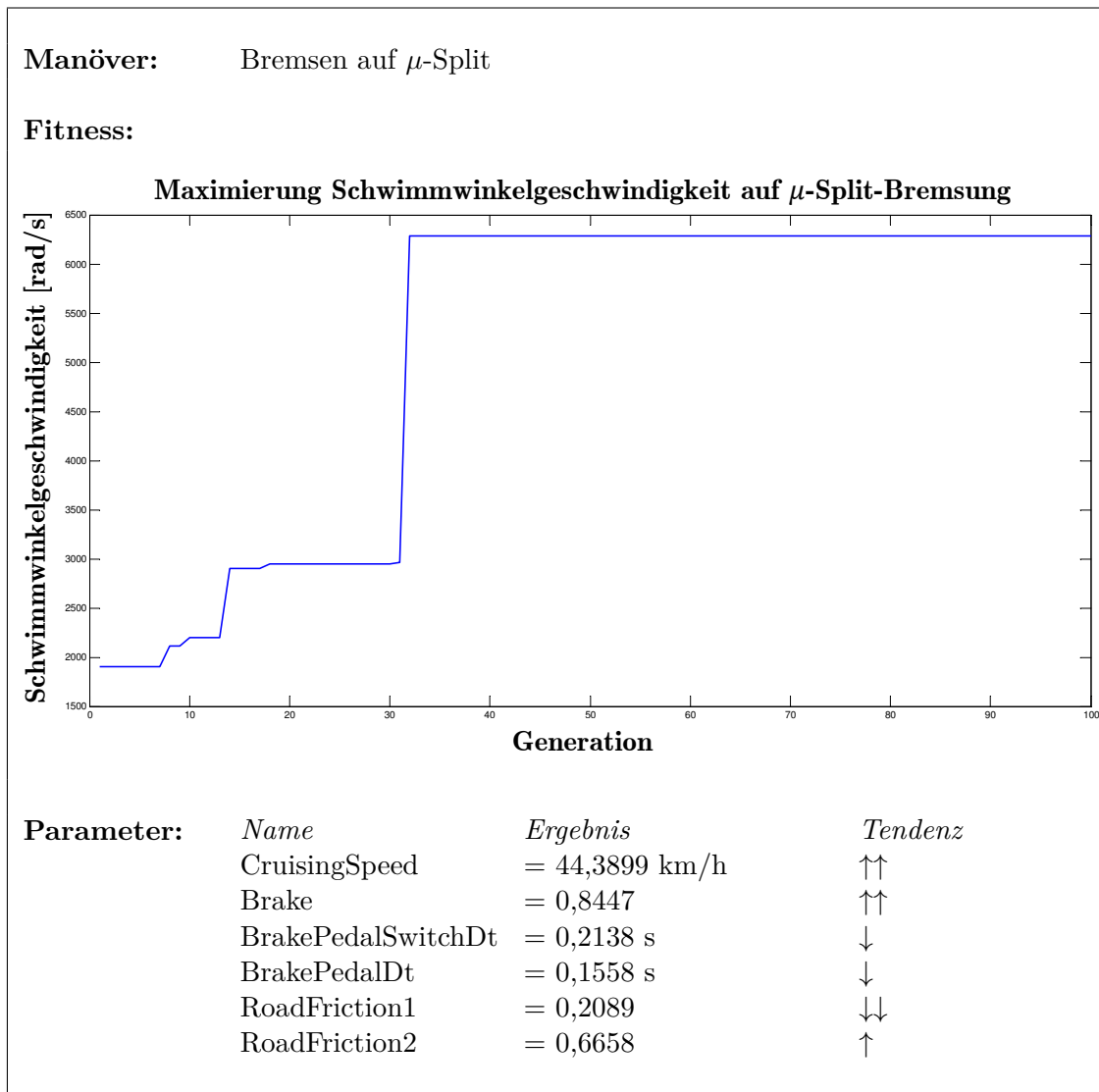


Tabelle 4.11. – Ergebnisse der μ -Split-Bremmung nach 100 Generationen

Auch hier hätte die Simulationszeit um weitere 2/3 reduziert werden können, wenn das Optimierungsergebnis dann bereits für den Einsatzzweck ausreichend gewesen wäre. Das bis zur Generation 100 beste Ergebnis, d.h. die Parameterkombination, die die höchste Schwimmwinkelgeschwindigkeit erzeugt, weist eindeutige Tendenzen für die Parameter auf. Während CruisingSpeed und Brake sehr hoch und RoadFriction2 hoch ausfallen, ist der Wert für RoadFriction1 besonders niedrig und BrakePedalSwitchDt und BrakePedalDt fallen ebenfalls niedrig aus. Dies bedeutet, dass sich das Fahrzeug besonders dann sehr stark ungewollt und unkontrolliert dreht, wenn eine sehr hohe Anfahrgeschwindigkeit mit starkem, schnellen Bremsengriff auf einer Fläche durchgeführt wird, die rechts und links sehr unterschiedliche Reibwerte hat. Dieses Ergebnis entspricht auch dem subjektiven Erwartungswert dieses Szenarios. In Anhang B.3 wird der Versuch wiederholt und diskutiert. Zur weiteren Diskussion und Auswertung des Versuchs werden in Abbildung 4.30 die Parameterwerte der je Generation bestangepassten Individuen und der zugehörige Indexwert des

jeweiligen Individuums dargestellt. Abbildung 4.31 zeigt die Parameterwertkorridore und die durchschnittlichen Parameterwerte pro Generation. Es zeigt sich in Abbildung 4.30a, dass die grundsätzliche Tendenz der Parameter sich bereits in den ersten Generationen herauskristallisiert. Bis Generation 20 finden starke Schwankungen statt, anschließend bewegen sich die Parameter in einem kleineren Rahmen weiter, um dann nach Generation 32 lange zu stagnieren. In Generation 90 findet noch eine Verbesserung des Ergebnisses statt (siehe auch die Spitze in Abbildung 4.30b), die allerdings keinen nennenswert großen Einfluss mehr auf das Gesamtergebnis Schwimmwinkelgeschwindigkeit hat (siehe Ergebnis in Tabelle 4.11). Im Vergleich zum ersten Experiment aus Kapitel 4.4.6 zeigt sich durch die Gegenüberstellung von Abbildung 4.26b mit Abbildung 4.30b, dass trotz der höheren Anzahl an Parametern und keiner vorherigen Analyse durch statistische Versuchsplanung, weniger Optimierungsschritte, die als Spitzen in den Abbildungen dargestellt sind, auftreten. Dies zeigt, dass allein von der Anzahl der Parameter nicht auf die Komplexität der Optimierungsfrage geschlossen werden kann. Eine statistische Analyse kann Sinn machen, ist aber nicht zwingend notwendig. Durch eine breite Aufstellung der Parametergrenzen kann der evolutionäre Algorithmus der DE auch ohne voroptimierten Startpunkt schnell zum Ziel kommen.

Durch Betrachtung von Abbildung 4.31a wird weiterhin deutlich, dass jedoch eine eindeutige Parametertendenz nicht sichtbar wird. Bei allen Parametern bleibt der Minimum-Maximum-Korridor der Werte weit geöffnet: das Problem ist also nicht trivial und eindeutig zu lösen. In Abbildung 4.31b zeigt sich jedoch, dass der Durchschnittswert für CruisingSpeed über die 100 Generationen hinweg stark ansteigt und somit eine eindeutige Richtung hat. Für alle anderen Parameter gilt dies nicht, sie bewegen sich über die komplette Simulation hinweg mehr oder weniger stetig in einem gewissen Bereich. Dies lässt - ohne eine statistische Analyse des Testfalls durchzuführen - darauf schließen, dass CruisingSpeed den höchsten Einfluss bzw. Effekt auf die Zielgröße Schwimmwinkelgeschwindigkeit aller Parameter hat. D. h. dass auf die Auslegung der Anfahrtsgeschwindigkeit der größte Fokus gelegt werden muss.

Abschließend kann als Fazit für das Beispiel gezogen werden, dass auch hier für eine tiefgehende Optimierung weitere Generationen möglich wären, jedoch aufgrund der bereits nach 32 Generationen eintretenden, langen Stagnation des Optimierungswertes Schwimmwinkelgeschwindigkeit ein ausreichend gutes Ergebnis erreicht wird. Auch ohne vorhergehende Analyse durch statistische Versuchsplanung konnte mit wenig Zeit- und Ressourcenaufwand ein befriedigendes Ergebnis erzielt werden. Grundsätzlich hätte die Simulationszeit um über die Hälfte reduziert werden können, wenn die Qualität des Ergebnisses bereits für den Einsatzzweck ausreichend ist. Die Methode der DE konnte auch hier in kurzer Zeit einen Testfall erzeugen, der ein per Hand schwer zu erstellendes Szenario darstellt. Neben dem Parametersatz selbst erhält man auch weiteren Einblick in die Dynamik und das Verhalten des Testfalls selbst und könnte nun ohne große Probleme weitere Testfälle mit gewünschten Eigenschaften wie z. B. der Maximierung oder Minimierung einer Größe oder der Erstellung von Grenzfällen erzeugen.

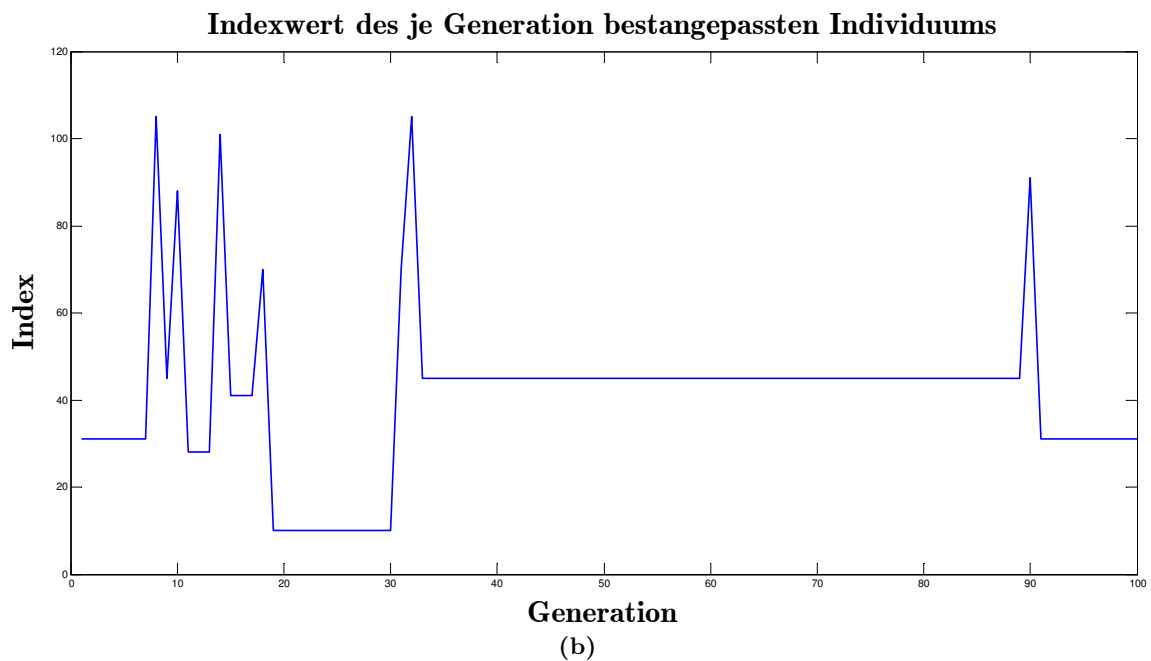
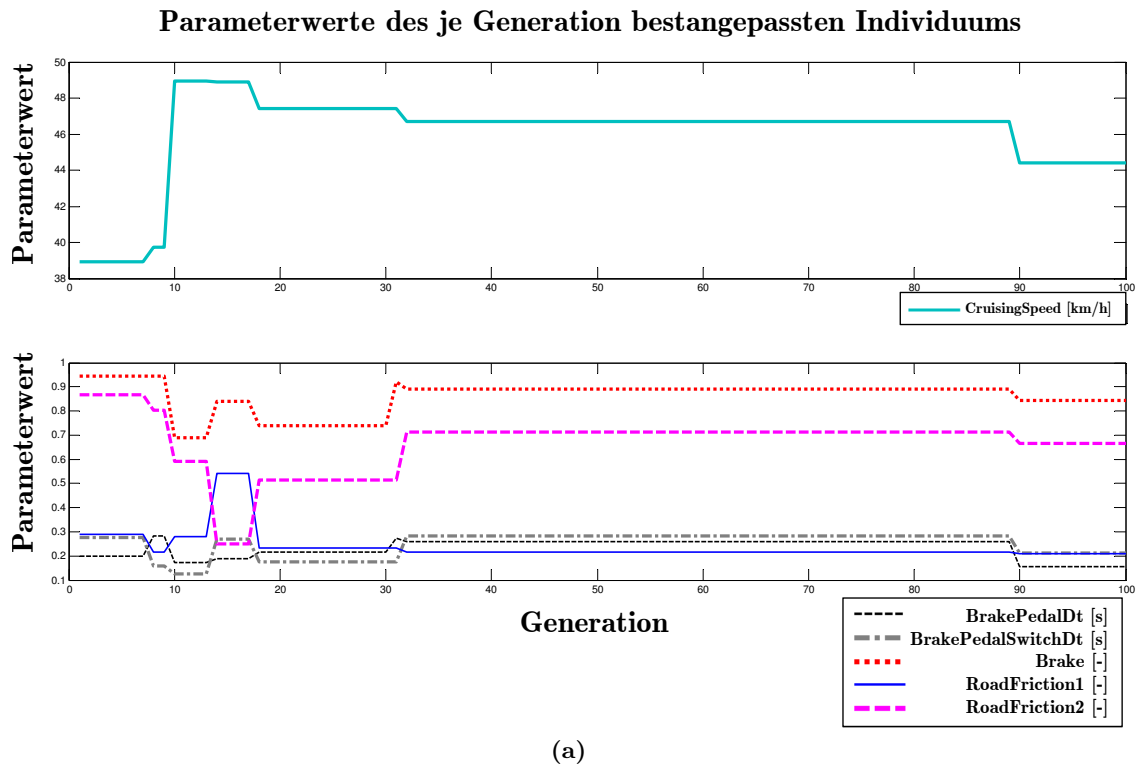


Abbildung 4.30. –

(a) Darstellung der Veränderung der Parameterwerte der jeweils minimalen Individuen hinsichtlich der Fitnessfunktion je Generation.

(b) Darstellung der Indizes der jeweils minimalen Individuen hinsichtlich der Fitnessfunktion je Generation mit 120 Einzelindividuen (60 Eltern, 60 Kinder). Jede Spitze stellt einen Wechsel von Eltern- zu Kindindividuum, also eine Optimierung, dar.

Deutlich wird, dass nach der bereits sehr gut optimierten Generation 32 (siehe Tabelle 4.11) kaum noch Optimierung stattfindet. Die Dynamik der Parameter nimmt ebenfalls nach Generationen 32 stark ab.

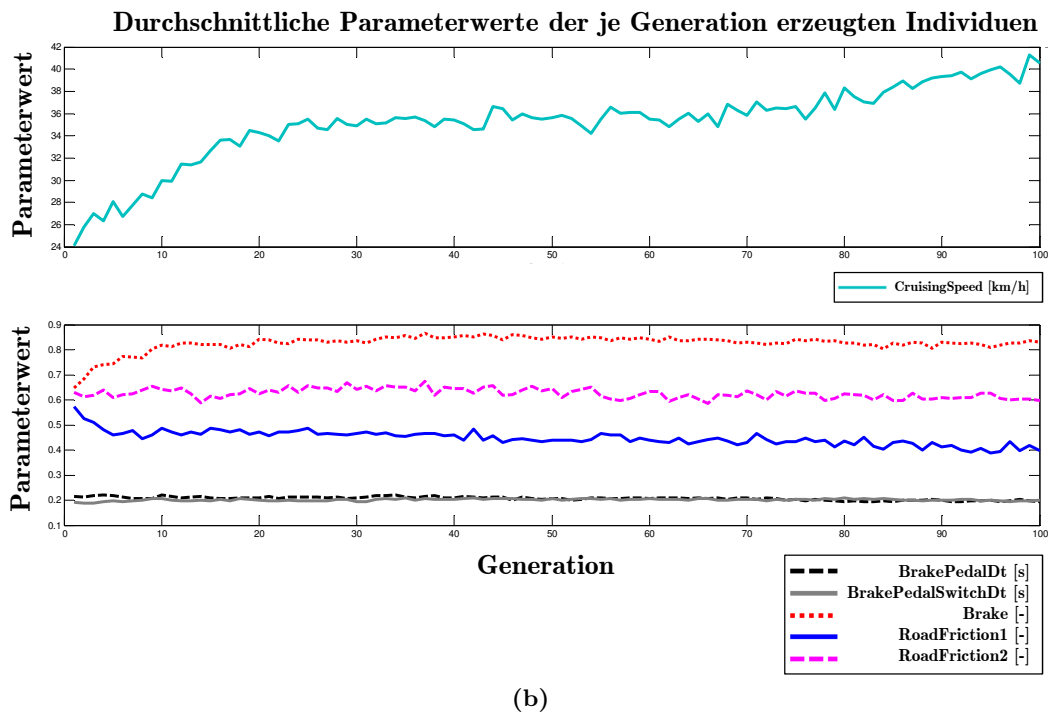
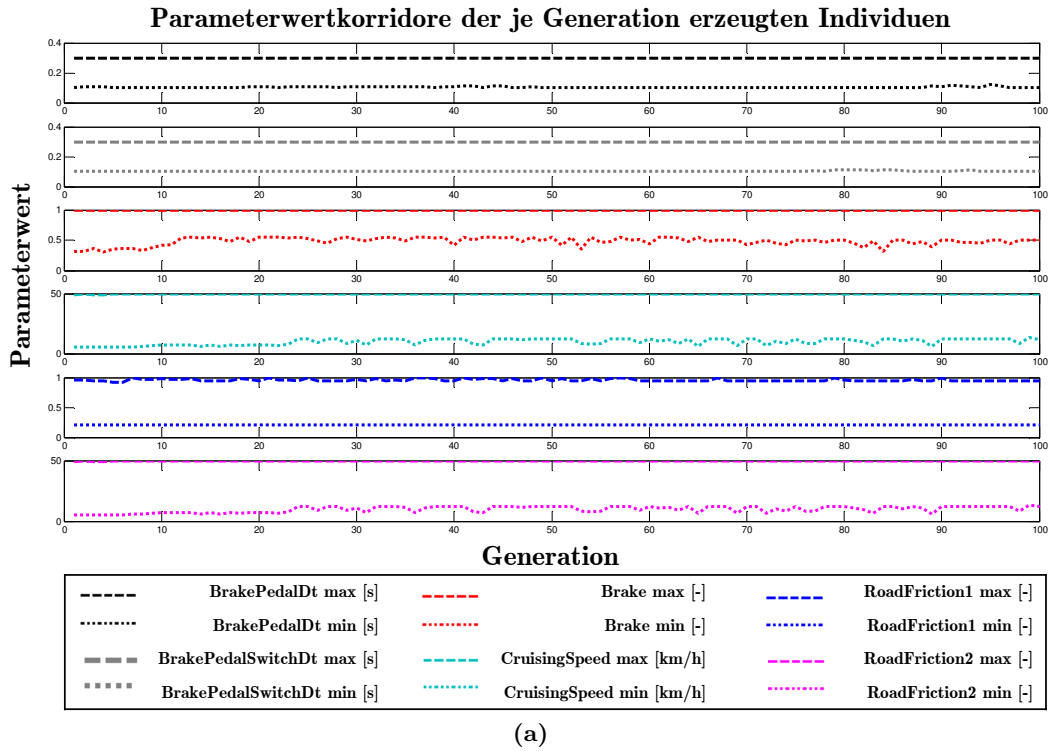


Abbildung 4.31. –

(a) Darstellung der jeweils maximalen und minimalen Parameterwerte, die in den 120 Versuchssindividuen pro Generation auftraten.

(b) Darstellung der durchschnittlichen Parameterwerte aller Individuen pro Generation mit 120 Einzelindividuen (60 Eltern, 60 Kinder).

Deutlich wird hier, dass im Gegensatz zum ersten Beispiel (siehe analog Abbildung 4.27), sich kein Parameterwertkorridor nennenswert verkleinert. Außer CruisingSpeed, die sich kontinuierlich im Durchschnitt erhöht, bleiben alle Parameter weitestgehend konstant.

4.4.8. Fazit

Die Vorteile der vorgestellten Methode zur Optimierung von Fahrdynamiksszenarien sind folgende:

- Es handelt sich um eine Methode zum schnellen Auffinden von optimierten Manövern (Vorbereitungs- und Simulationszeit im Bereich weniger Stunden, je nach Komplexität des Szenarios und der Anzahl der Parameter)
- Die Tendenz der Parameter wird schon nach wenigen Generationen sichtbar
- Aus den Versuchsergebnissen können zusätzlich zur statistischen Versuchsplanung Schlüsse zur Analyse und dem Verhalten des jeweiligen Testszenarios gezogen werden.
- Jegliches Szenario und jegliche Optimierung (Maximierung, Minimierung etc.) sind möglich
- Wenig Knowhow im Bereich Fahrdynamik ist notwendig, um ein Testszenario aufzubauen
- Keine Hardware außer einem normalen Rechner wird benötigt
- Die Simulation kann schneller als in Echtzeit erfolgen, da das Szenario erst nach der Optimierung auf das Testsystem und somit auf das Echtzeitsystem transferiert wird

Die Kombination aus manöverbasierten, simulativen Testfällen und einem evolutionären Algorithmus bietet also eine gute Möglichkeit, um schnell, ohne viel Modellierungsaufwand und ohne exploratives Suchen ein Testszenario mit gewünschten Eigenschaften zu erzeugen.

4.5. Testfallpriorisierung

4.5.1. Ziele

Testfallpriorisierung ist eine effektive und praktische Methode zur Steigerung der Erfüllung von Testzielen wie z. B. das Auffinden von Fehlern [142]. Wie in Kapitel 2.3.2 dargestellt, ergeben sich durch ISO 26262 zahlreiche Methoden, die in eine große Anzahl an Einzeltestfällen münden. Um diese Vielzahl an Einzeltests möglichst effizient auszuführen, sind verschiedene Ansätze zur Testfallpriorisierung möglich.

Testfallpriorisierungsmethoden wie z. B. in [27], [98] oder [140] dargestellt, finden in allen Bereichen des Software- und Systemtests ihre Anwendung. Die Hauptziele der Priorisierung sind nach [98]:

- Steigerung der Anzahl der gefundenen Fehler
- Beschleunigung des Auffindens von sehr kritischen Fehlern
- Beschleunigung der Testfallüberdeckung

- Beschleunigung der Überdeckung von kritischen Funktionen (z. B. mit hohem ASIL)
- Beschleunigung des Vertrauens in das System
- Verbesserte Auslastung von Ressourcen

In erster Linie soll also durch Maßnahmen der Ordnung der Testaktivitäten und Testfälle ein Effizienzgewinn erreicht werden.

4.5.2. Maßnahmen

Viele Metriken zur Testfallpriorisierung wie z. B. in [26] oder [89] schlagen eine eingehende Analyse und Bewertung der Anforderungen, des Codes und des Systems vor. Da im Systemtest jedoch nur Black-Box-Verfahren Anwendung finden können, sind tieferegehende Analysen nicht möglich und auch äußerst zeitraubend. Aus diesem Grund sollen einige Kriterien zur Testfallpriorisierung vorgeschlagen werden, die ihre Anwendung in der Testplanung relativ einfach und unaufwändig finden können:

- **ASIL:**
Nach dem Motto „hardest first“ sollten die Funktionen mit dem höchsten ASIL zuerst getestet werden.
- **Kritikalität:**
Funktionen, deren Fehlverhalten sehr kritisch wäre, sollten priorisiert getestet werden. Häufig geht dies auch mit dem ASIL einher.
- **Zentralität:**
Zentrale Funktionen, deren Ausführung unerlässlich für das Gesamtsystemverhalten sind, sollten früh abgesichert werden.
- **Neuheitsgrad:**
Neue Funktionen sind fehleranfälliger und sollten priorisiert getestet werden.
- **Aufwand:**
Testvorbereitungszeit, Equipment und Kosten der Durchführung (Zeitaufwand für den Ingenieur, Durchführungs- und Auswertungsaufwand) müssen für die Testfallpriorisierung berücksichtigt werden. Häufig ist es sinnvoll, Tests mit wenig Aufwand vorzuziehen, um die Testabdeckung schneller zu erreichen.

Besonders die Priorisierung des Tests der ASIL-Anforderungen ist ein wichtiges Maß. Abbildung 4.32 zeigt für ein beispielhaftes Projekt aus der Fahrzeugsicherheit die ASIL-Verteilung der Anforderungen. Besonders diese Anforderungen sollten möglichst priorisiert getestet werden, um schnell Vertrauen in die Funktionen der Funktionalen Sicherheit zu erhalten.

Weitere mögliche Regeln zur Priorisierung werden von [86] vorgeschlagen:

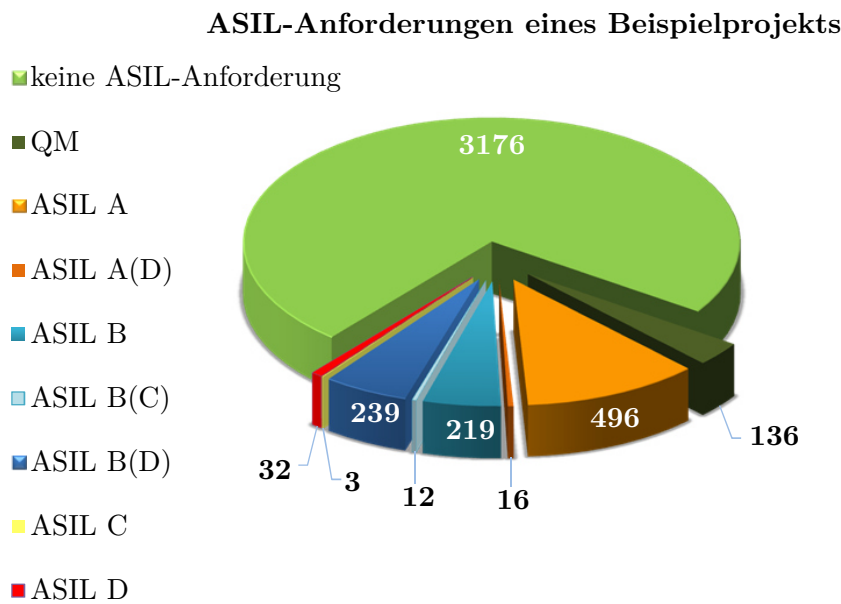


Abbildung 4.32. – Anteil der ASIL-Anforderungen in einem kontemporären Fahrzeugsicherheitsprojekt: Von insgesamt 4329 Anforderungen haben 1153, also 27%, eine Relevanz hinsichtlich ISO 26262.

- **Der Test der Gesamtsystemfähigkeiten ist genauso wichtig wie der Test der Einzelkomponenten**
Begründung: Tester tendieren dazu, Einzelkomponenten gründlicher zu testen als das Zusammenspiel der Komponenten.
- **Der Test alter Fähigkeiten ist genauso wichtig wie der Test neuer Fähigkeiten**
Begründung: Tester testen neue Funktionen gründlicher als bereits bestehende, was dazu führen kann, dass bereits abgesicherte Funktionen nicht mehr ausreichend getestet werden.
- **Der Test typischer Situationen ist genauso wichtig wie der Test von seltenen Ereignissen**
Begründung: Systematisches Grenzwerttesten unterscheidet nicht zwischen häufigen und seltenen Situationen, weshalb zusätzlich Fokus auf den ausführlichen Test typischer Situationen gelegt werden sollte.

In der Praxis können nicht immer alle Priorisierungskriterien berücksichtigt werden. Manche Regeln stehen sogar fast in Konflikt zueinander. Die Testplanung sollte nichtsdestotrotz eine grundlegende Priorisierung der Testaktivitäten beinhalten, um Synergien auszunutzen.

Fazit

In diesem Kapitel wurde ein neues Vorgehen zur Testfallauswahl und zugehörige Testmethoden dargestellt. Es wurde der Ansatz der Testauswahl in Abhängigkeit der Komplexitätsbewertung von Projekten dargelegt. Anschließend wurden drei Testmethoden ausformuliert, die ihren Einsatz im Systemtest integraler Fahrzeugsicherheitssysteme finden bzw. finden können. Es wurde die Testmethode des simulativen, manöverbasierten Testens vorgestellt, die aktuell sporadisch verwendet wird. Darauf aufbauend wurden die damit kombinierbaren Testmethoden der evolutionären Algorithmen und der Statistischen Analyse von Testfällen dargestellt, die in dieser Arbeit neu eingeführt wurden. Die Testmethoden wurden jeweils an einem Beispiel erläutert. Abschließend wurde ein kurzer Einblick in die Ziele der Testfallpriorisierung gegeben. Anhand verschiedener Kriterien können Testfälle geordnet werden und so Synergien hinsichtlich der Erreichung der Testziele ausgenutzt werden.

5. Testsystem



Abbildung 5.1. – Teil 4 der in der Arbeit vorgeschlagenen Methodik (siehe auch Abbildung 1.3).

In diesem Kapitel wird ein neuartiges Testsystem vorgestellt, das den aktuellen und zukünftigen Anforderungen an den Systemtest in der integralen Fahrzeugsicherheit gerecht wird. Durch die Integration von durchgängigen Testmethoden, automatischer Auswertung und der Wiederverwendbarkeit von Testfällen kann der Test von neuartigen Sicherheitssystemen abgesichert werden. Zunächst werden die allgemeinen Anforderungen an eine Testumgebung in der Fahrzeugsicherheit dargestellt. Anschließend wird der Testansatz der Kombination aus Fahrdynamiksimulation und Crashdateneinspeisung kurz vorgestellt. Abschließend wird eine Beispielkonfiguration für ein HiL-Testsystem aufgezeigt, die für den Systemtest in der integralen Fahrzeugsicherheit eingesetzt werden kann.

5.1. Anforderungen an die Testumgebung

Für den Test von integralen Fahrzeugsicherheitssystemen ist ein leistungsfähiges Testsystem notwendig, das sowohl den Lebenszyklus des Systems durchgängig begleitet als auch die Wiederverwendbarkeit von Testfällen und -methoden beinhaltet. Grundsätzliche Anforderungen an ein umfassendes Testsystem in der Fahrzeugsicherheit sind:

- Durchgängigkeit über MiL, SiL und HiL
- Wiederverwendbarkeit der Testfälle
- Modularer Aufbau zum flexiblen Einsatz
- Automatisierter Ablauf
- Anbindung von Auswertung und Datenbanken
- Einfache Integration neuer Methoden, Erweiterbarkeit
- Abbildung aller Schnittstellen zum System (Bussysteme, Sensoren, Aktoren, Diagnose)
- Zeitsynchronität und Echtzeitfähigkeit
- Möglichkeit zur Anbindung von Modellen und Simulation
- Einfache Modifikation bei Änderungen der Datenbasen
- Einfache Modifikation bzw. Anpassung an unterschiedliche Projekte
- Möglichkeit zur Stimulation, Fehlereinspeisung, Manipulation
- Einfache Bedienung und Schulbarkeit
- Ausführliche Dokumentation
- Nachweis der Fehlerfreiheit (z. B. durch Zertifizierung)
- Wartbarkeit

Deutlich wird, dass zum einen die physischen Gegebenheiten wie z. B. die Bereitstellung aller Schnittstellen erfüllt sein müssen, aber auch der Umgang, die Erweiterung und die Modifikation des Testsystems eine wichtige Rolle spielen. Nur durch ein flexibles und leistungsfähiges Testsystem können zukünftige, vernetzte und integrierte Sicherheitssysteme ausreichend abgesichert werden.

5.2. Kombination Fahrdynamiksimulation und Crashdateneinspeisung

Um einen umfangreichen Test von Sicherheitssystemen durchzuführen ist auch einzubeziehen, dass Fahrdynamik- und Unfalldaten nicht nur getrennt von einander, sondern auch in Kombination eingespeist werden können. Lange Zeit wurden die aktive und die passive Sicherheit separat betrachtet und entwickelt. Bis heute besteht die Trennung der beiden Welten im Test: einerseits werden Fahrdynamikdaten aus aufgezeichneten Testfahrten eingespeist (aktive Sicherheit) und andererseits werden Crashdaten in das vorinitialisierte System eingebracht, die die Zündzeiten der passiven Sicherheitssysteme testen. Crashdaten sind z. B. die hochdynamischen Beschleunigungswerte, die bei einem Aufprall entstehen. Die integrale Fahrzeugsicherheit macht es notwendig, diese beiden Welten zu kombinieren und in einem durchgängigen Datenstrom für das System abzubilden. Außerdem bedingt die getrennte Betrachtung des Tests auch, dass ein Sicherheitssteuerggerät den vollständigen Ablauf eines Unfalls von normaler Fahrt über Pre-Crash- und In-Crash-Phase zum ersten Mal in einem realen Unfall auf einer realen Straße beim Fahrer des Vehikels erfährt. Dies stellt ein Risiko dar, das sich durch die Kombination aus Fahrdynamik- und Umfeldsimulation und Crashdateneinspeisung abmildern lässt, indem zum einen durch Simulation ein Szenario realistisch in das Steuergerät eingespeist wird und zum anderen dazu passende Crashdaten nahtlos abgespielt werden. Abbildung 5.2 zeigt ein Beispiel für diese Kombination. Durch Simulation oder Einspeisung von aufgezeichneten Testfahrten wird das Steuergerät mit zunächst mit einer normalen Fahrt und dann durch eine kritische Situation in die Pre-Crash-Phase versetzt. Der Aufprallzeitpunkt wird durch virtuelle Sensoren, die am virtuellen Fahrzeug angebracht sind, detektiert. Zu diesem Zeitpunkt werden die zugehörigen Unfallparameter wie z. B. die Geschwindigkeit, der Aufprallwinkel und die Überdeckung aus der Simulation ermittelt. Anschließend wird in einer Crashdatenbank nach passenden Crashdaten gesucht und diese werden über die entsprechenden Sensorschnittellen wie z. B. Beschleunigungssensoren abgespielt. Natürlich kann aufgrund von Echtzeitanforderungen an die Einspeisung die Berechnung der Parameter bereits im Vorfeld erfolgen, sodass keine Zeitverzögerung beim Umschalten zwischen Fahrdynamik- und Crashdaten entsteht. Aufgrund der Reproduzierbarkeit der Einspeisung ist dies ohne Probleme möglich.

Hier zeigt sich ein Ansatzpunkt für die Testfalloptimierung aus Kapitel 4: um zu den vorhandenen Crashdaten, die häufig aufgrund des hohen Zeit- und Kostenaufwands nur sehr spärlich und in ganz bestimmten Testfällen vorhanden sind, eine passende Fahrdynamik- und Umfeldsimulation zu erzeugen, muss großer Aufwand betrieben werden. Zum Aufprallzeitpunkt müssen die Simulationsparameter genau zu den Crashdaten passen, da sonst das Steuergerät die Inkonsistenz erkennt und in einen Fehlerzustand geht. Somit kann die Auslösung von z. B. Airbags nicht getestet werden. Um nun z. B. einen Testfall mit einer bestimmten Geschwindigkeit, einem bestimmten Winkel und/oder einer bestimmten Überdeckung zu erzeugen, kann man sich der statistischen Testfallanalyse und evolutionären Algorithmen bedienen, um so einen Testfall zu erhalten, der genau auf die Crashdaten passt.

Somit können die Zündzeiten und die Reaktion anderer Sicherheitssysteme, die nach dem Aufprallzeitpunkt aktiviert werden, durchgängig getestet werden.

5.3. Beispielkonfiguration für ein Hardware-in-the-Loop-Testsystem

Nachfolgend wird eine Beispielkonfiguration für ein Hardware-in-the-Loop-Testsystem, das für den umfassenden Systemtest eines Sicherheitssteuergeräts eingesetzt werden kann, aufgezeigt. Abbildung 5.3 zeigt hierfür einen möglichen Aufbau. Für den Test muss zum einen die Möglichkeit der Programmierung, Kalibrierung und des Zugriffs auf den Fehlerspeicher gegeben sein. Dies lässt die Modifikation und Analyse des Steuergeräteverhaltens zu. Die flexible Anwendung und Anbindung von Testmethoden wie z. B. in Kapitel 4 dargestellt, ermöglicht einen umfassenden Test und die Erreichung einer hohen Testtiefe. Durch eine automatisierte Auswertung und die Anbindung an eine Testdatenbank können die Testergebnisse übersichtlich dargestellt und zurückgespielt werden. Das Testsystem selbst muss

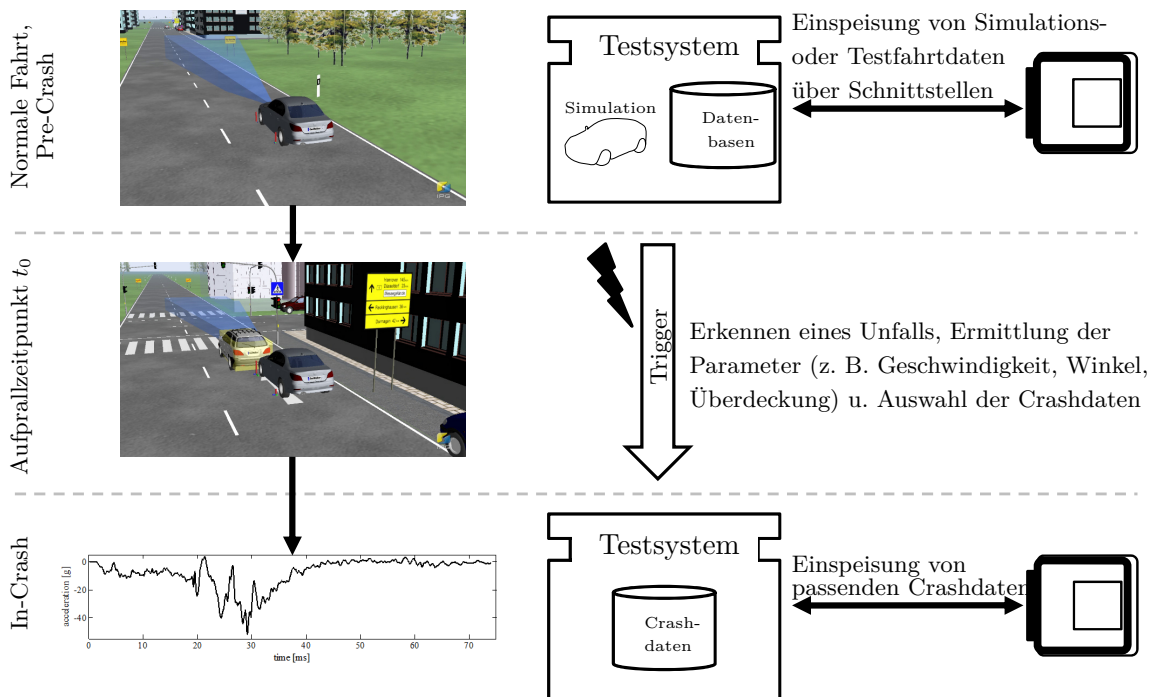


Abbildung 5.2. – Schematische Darstellung des Ablaufs des durchgängigen Testens. Als erster Schritt wird ein normales Fahrscenario simuliert oder per aufgezeichneter Testfahrt eingespeist. Kurz vor dem Unfall können Systeme der Pre-Crash-Phase getestet werden. Zum Aufprallzeitpunkt wird der Unfall detektiert, die zugehörigen Parameter wie Geschwindigkeit, Winkel und Überdeckung ermittelt und passende zugehörige Crashdaten über die Schnittstellen in das Steuergerät eingespeist [88].

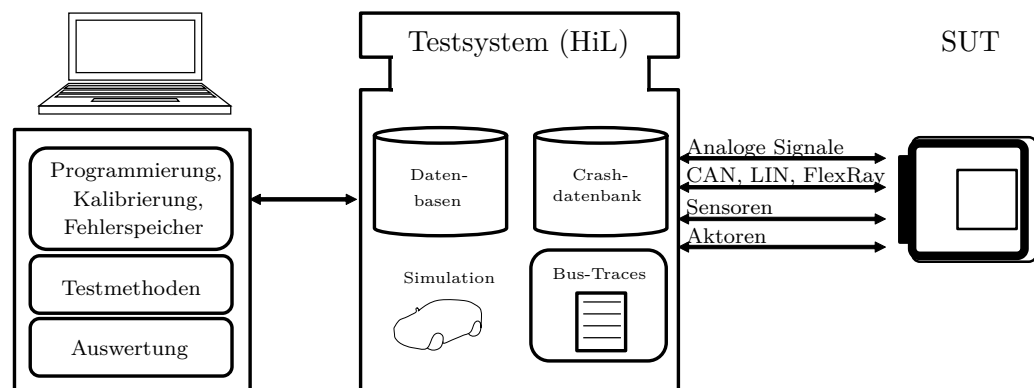


Abbildung 5.3. – Schematische Abbildung eines HiL-Testsystems für das Sicherheitssteuerggerät im Fahrzeug. Ein Bedienrechner übernimmt Programmierung, Kalibrierung, Auslesen des Fehlerspeichers sowie die Testmethoden und die Auswertung. Das Testsystem selbst beinhaltet eine Crashdatenbank, Datenbasen, die Fahrdynamik- und Umfeldsimulation sowie aufgezeichnete Bus-Traces. Das Steuergerät wird dann über verschiedene Schnittstellen genauso versorgt als wäre es in einem realen Fahrzeug.

alle notwendigen Datenbasen hinsichtlich Bussystemen, Sensoren und Aktoren beinhalten. Die Anbindung einer Crashdatenbank ist für die Einspeisung von Unfalldaten notwendig und wie in Kapitel 5.2 dargestellt für den Test von Sicherheitssteuergeräten bedeutend. Durch die Anbindung von Fahrdynamik- und Umfeldsimulation wird manöverbasiertes Testen möglich, das hohe Flexibilität bietet. Durch die Freiheit der Szenarien- und Fahrzeugwahl kann bereits im Vorfeld zu realen Testfahrten jeglicher Testfall simuliert werden. Natürlich ist jedoch auch die Möglichkeit zur Wiedergabe von realen aufgezeichneten Daten notwendig, wie sie durch die Einspeisung von Bus-Traces realisiert wird. Das Steuergerät selbst wird durch verschiedene physische Schnittstellen gespeist.

Wichtige Punkte sind nicht nur die Simulation und die Dateneinspeisung selbst, sondern auch die Fehlereinbringung. Folgende Möglichkeiten zur Fehlereinspeisung sollten gegeben sein:

- Zeitfehler von Bussystemen
- Signalfehler von Bussystemen
- Protokollfehler von Bussystemen
- Simulation von Leitungsabrissen
- Manipulation von elektrischen Signalen
- Simulation von Peripherie (z. B. Gurtschlösser)

Bussysteme bezieht sich hier sowohl auf Kommunikationsbusse wie CAN, LIN und FlexRay als auch auf analoge und digitale Sensor- und Aktorbussysteme. Zur Überprüfung des richtigen Verhaltens des Steuergeräts müssen auch dessen Ausgangssignale abgegriffen

werden. Gerade Zündströme für Airbags und Auslösungsanforderungen für andere Sicherheitssysteme wie z. B. den reversiblen Gurtstraffer müssen genau messbar sein.

Eine große Herausforderung ist auch die notwendige Zeitsynchronität, die durch eine gemeinsame Zeitbasis auf Seite des Testsystems erreicht werden muss. Auch der automatisierte Ablauf von Testfällen hintereinander muss durch vorangestellte Neuinitialisierung des Steuergeräts gewährleistet werden.

Deutlich wird durch dieses Beispiel, dass nicht nur die integralen Fahrzeugsicherheitssysteme selbst immer komplexer werden, sondern auch die zugehörigen Testsysteme an Umfang, Möglichkeiten und Leistungsfähigkeit gewinnen müssen. Nur durch neuartige, effektive und effiziente Ansätze im Testsystembereich können zukünftige Systeme ausreichend abgesichert werden.

Fazit

In diesem Kapitel wurden kurz die Anforderungen an ein Testsystem für den Einsatz in der integralen Fahrzeugsicherheit aufgezeigt. Der Ansatz der Kombination von Fahrdynamik- und Umfeldsimulation mit Crashdateneinspeisung wurde dargelegt und eine Beispielkonfiguration für ein Hardware-in-the-Loop-Testsystem für den Einsatz im Systemtest der Fahrzeugsicherheit beschrieben.

6. Metrik zur Testfortschrittsbeurteilung



Abbildung 6.1. – Teil 5 der in der Arbeit vorgeschlagenen Methodik (siehe auch Abbildung 1.3).

In diesem Kapitel wird eine neuartige Metrik zur Testfortschrittsbeurteilung dargestellt werden. Zunächst wird kurz auf die Notwendigkeit einer solchen Metrik eingegangen. Anschließend wird eine Möglichkeit zur Testqualitätsbewertung dargestellt. Als wichtiger Punkt wird danach die Testüberdeckung diskutiert, die anhand verschiedener Überdeckungsmaße definiert wird. Abschließend werden die Testqualitätsbewertung, die Testüberdeckung und die Fehlerschätzung aus Kapitel 3.3 zum sogenannten Vertrauensgrad zusammengefasst, der eine übersichtliche Metrik zur Testfortschrittsbeurteilung darstellen soll.

Zur Kontrolle, Nachvollziehbarkeit und Rechtfertigung des Testens ist es notwendig, den Testfortschritt zu beurteilen. Häufig wird dies durch Dokumentation und das subjektive Gefühl von erfahrenen Testern abgebildet, was jedoch nicht als objektives Kriterium bzw. Begründung verwendet werden kann. Allgemein sind nach [118] folgende Punkte zur Testauswertung wichtig:

- Auswertung der Testergebnisse
- Fehleranalyse
- Messung der Testüberdeckung
- Testmetrik

Die Auswertung der Testergebnisse erfolgt häufig automatisiert und die Erwartungswerte beruhen im Systemtest auf den Anforderungen aus System- und Architekturspezifikation. Eng verknüpft mit der Auswertung ist die Fehleranalyse. Hierbei können in schwierigen oder kritischen Fällen ergänzende Methoden, wie sie in Kapitel 4 dargestellt wurden, herangezogen werden. Wichtig ist allerdings auch die Frage, wann ausreichend getestet wurde. Allein an der Testüberdeckung ist dies noch nicht festzuhalten, da die Qualität des Testens nicht allein an der Anzahl der Tests bzw. der Testüberdeckung sichtbar ist. Qualität ist nicht allein durch Quantität erreichbar. Auch andere Faktoren wie z. B. Testdokumentation und Reviews von Tests sind wichtige Indikatoren für Testqualität. Häufig bedient man sich zur Lösung dieses Problems Testmetriken. Nach [33] ist eine Testmetrik definiert zu:

Definition 6.1 (Testmetrik)

Eine Testmetrik stellt die messbare Eigenschaft eines Testfalls oder Testlaufs mit Angabe einer zugehörigen Messvorschrift wie z. B. eines Testabdeckungsgrades dar. Sie definiert Testendekriterien.

Nachfolgend soll eine Testmetrik, der sogenannte Vertrauensgrad vorgestellt werden, der neben der Testüberdeckung und der Fehlerauswertung aus Kapitel 3.3 auch eine Testqualitätsbewertung anhand eines Expertenaudits (siehe auch Kapitel 3.2) mit einbezieht. Die Metrik soll die vielen einzelnen Teile des Systemtests zusammentragen und übersichtlich darstellen.

6.1. Testqualitätsbewertung

Die Forderung nach hoher Testtiefe ist im Fahrzeugsicherheitsbereich allgegenwärtig. Durch qualitativ hochwertiges Testen muss eine Absicherung von Systemen erreicht werden, ohne alle theoretisch möglichen Testfälle ausführen zu können, da die schiere Anzahl an Kombinationsmöglichkeiten ins Unendliche geht [9]. Zur Bewertung der Testqualität sind grundsätzlich mehrere komplementäre Ansätze möglich und zur aussagekräftigen Beurteilung sind sowohl objektive als auch subjektive Methoden notwendig. Objektive Methoden bedienen sich hierbei Kennzahlen und subjektive funktionalen Gesamtbeurteilungen [108].

Um dies in einem Prozess abzubilden, kann man sich analog zu Kapitel 3 einem Expertenaudit mit relevanten Einflussgrößen auf die Testqualität bedienen und so den aktuellen Stand des Testens in einem objektiven Verfahren abbilden.

6.1.1. Die Testqualität beeinflussende Faktoren

Für die Testqualitätsbewertung sollen analog zur Komplexitätsbewertung aus Kapitel 3 beeinflussende Faktoren, also Kennzahlen und Indikatoren zur Bestimmung, dienen. Auch hier können diese durch Analyse von Projekten und Befragung von langjährigen Projektmitarbeitern eruiert werden. Mögliche n_Q Faktoren für die Bewertung des Testfortschritts von Projekten in der Fahrzeugsicherheit sind in Tabelle 6.1 mit zugehöriger Bewertungsskala $B_{K,i}$ von 0 bis 5 dargestellt.

Faktor $n_{Q,i}$	Bewertung $B_{Q,i}$	Beschreibung
Anzahl und Verfügbarkeit der Tester und Tools	5 - Ausreichend	Für die zeitnahe und korrekte Ausführung der Testaktivitäten müssen auf den Testaufwand bezogen ausreichend Tester und Tools vorhanden sein.
	2,5 - Durchschnittlich	
	0 - Zu wenig	
Automatisierungsgrad der Testausführung	5 - Hoch ($\gg 70\%$)	Je höher der Automatisierungsgrad der Testaktivitäten ist, desto besser wird Reproduzierbarkeit gewährleistet und menschliche Fehler werden vermieden.
	2,5 - Durchschnittlich (60-70%)	
	0 - Gering ($\ll 60\%$)	
Dokumentation der Testaktivitäten	5 - Umfassend: Dokumente zur Nachvollziehbarkeit sind in den entsprechenden Qualitätstools fast vollständig hinterlegt	Durch Dokumentation werden die Testaktivitäten reproduzierbar, nachvollziehbar, eindeutig und detailliert abgelegt.
	2,5 - Durchschnittlich: Dokumente sind überwiegend hinterlegt	
	0 - Wenig: kaum Dokumente hinterlegt.	

Faktor $n_{Q,i}$	Bewertung $B_{Q,i}$	Beschreibung
Erfahrung der Tester	<p>5 - Hoch: Experten mit umfassenden Prozess- und Technologieverständnis</p> <p>2,5- Durchschnittlich: mindestens ein Projekt wurde komplett begleitet</p> <p>0 - Wenig: Berufsanfänger oder wenig Seniorerfahrung am Standort</p>	Erfahrene Tester können durch ihr Knowhow effizienter und effektiver testen.
Reifegrad vorhergehender Tests (SW, HW)	<p>5 - Hoch: alle Tests sind standardisiert, wurden definiert, durchgeführt und freigegeben</p> <p>2,5 - Durchschnittlich: Tests wurden spezifiziert, durchgeführt, aber nicht umfassend dokumentiert oder freigegeben</p> <p>0 - Gering: Tests nicht oder nur teilweise durchgeführt oder nicht definiert oder nur teilweise bestanden, Ergebnisse wurden nicht dokumentiert und nicht freigegeben</p>	Je mehr Fehler im Vorfeld bei der Entwicklung der Algorithmen, Software und Hardware durch gründliches und hochwertiges Testen gefunden werden, desto weniger beeinträchtigen sie den Systemtest.
Reviews der Tests	<p>5- Umfassend: die Testaktivitäten wurden fast vollständig gereviewt</p> <p>2,5 - Durchschnittlich: die Testaktivitäten wurden überwiegend gereviewt</p> <p>0 - Gering: kaum Testaktivitäten wurden gereviewt</p>	Durch Reviews werden Testaktivitäten kontrolliert und Fehler in der Testausführung vermieden.

Faktor $n_{Q,i}$	Bewertung $B_{Q,i}$	Beschreibung
Qualität der Testspezifikation	<p>5 - Hoch: Testspezifikation basiert auf frühzeitigem Requirements Engineering durch den Testarchitekten, es wurden aus allgemeinen Anforderungen spezielle für Testtools abgeleitet</p> <p>2,5 - Durchschnittlich: Testspezifikation evtl. aus vergangenen Projekten vorhanden, aber noch nicht auf aktuelles Projekt erweitert</p> <p>0- Gering: Kein Requirements Management, Testarchitekt schlecht oder nicht eingebunden, kaum Dokumentation</p>	<p>Die Qualität einer Testspezifikation ist gut, wenn sie detailliert, eindeutig, nachvollziehbar, reproduzierbar und wiederverwendbar ist.</p> <p>Durch die Vermeidung von Missverständnissen kann so die Qualität des Systemtests erhöht werden.</p>
Zuverlässigkeit des Testsystems	<p>5 - Hoch: Testtool ist langjährig bekannt und bewährt, ingenieurmäßig berechnet und beurteilt, Wartung vorhanden, sicher, realitätsnah, Aussagekraft der Tests bewiesen</p> <p>2,5 - Durchschnittlich: Testtool aus Vorgängerprojekt vorhanden oder Methode sehr einfach oder bekannt</p> <p>0 - Gering: Testtool neu und unbekannt, nicht abgesichert, Nachbesserungen notwendig</p>	<p>Ein zuverlässiges Testsystem soll einfach in der Benutzung, ausfallsicher, regelmäßig gewartet und korrekt sein, um qualitativ hochwertiges Testen zu gewährleisten.</p>

Tabelle 6.1. – Mögliche beeinflussende Faktoren für die Testqualitätsbewertung von Entwicklungsprojekten der Fahrzeugsicherheit (nach [9]). Die Bewertung erfolgt jeweils in fünf Stufen, wobei höhere Zahlen nahe fünf für bessere Bewertung stehen.

Natürlich können auch hier spezielle Faktoren ergänzt werden, die je Entwicklungsgegebenheiten spezifisch sein können. Die eruierten Faktoren können anhand eines Expertenaudits je Projekt bewertet werden und so einen Überblick über die Testqualität des Projekts geben. Zur realistischeren Bewertung sollten diese Faktoren noch gewichtet werden, da nicht alle denselben Einfluss haben.

6.1.2. Gewichtung der Faktoren

Zur Gewichtung der beeinflussenden Faktoren soll auch hier analog zu Kapitel 3 der paarweise Vergleich verwendet werden. Tabelle 6.2 zeigt den paarweisen Vergleich anhand der Befragung eines Experten und somit auch eine mögliche Gewichtung der Faktoren für die Projektkomplexität. Hierbei wird deutlich, dass auch die Faktoren für die Test-

Kriterium 1 → Kriterium 2 ↓	Qualität der Testspezifikation	Reifegrad vorhergehender Tests (SW, HW)	Erfahrung der Tester	Reviews der Tests	Dokumentation der Testaktivitäten	Automatisierungsgrad der Testausführung	Anzahl und Verfügbarkeit der Tester und Tools	Zuverlässigkeit des Testsystems	
Qualität der Testspezifikation	■	1	1	1	1	0	0	0	
Reifegrad vorhergehender Tests (SW, HW)	1	■	1	1	1	1	0	0	
Erfahrung der Tester	1	1	■	1	1	1	0	0	
Reviews der Tests	1	1	1	■	1	1	1	0	
Dokumentation der Testaktivitäten	1	1	1	1	■	1	1	0	
Automatisierungsgrad der Testausführung	2	1	1	1	1	■	0	1	
Anzahl und Verfügbarkeit der Tester und Tools	2	2	2	1	1	2	■	0	
Zuverlässigkeit des Testsystems	2	2	2	2	2	1	2	■	
Summe der Punkte	10	9	9	8	8	7	4	1	56
Rang	1	2	2	3	3	4	5	6	
Gewichtung in Prozent $G_{Q,i}$	17,9	16,1	16,1	14,3	14,3	12,5	7,1	1,8	100

Legende:

Kriterium 1 ist wichtiger als Kriterium 2	2:0
Kriterium 1 ist weniger wichtig als Kriterium 2	0:2
Kriterium 1 ist genauso wichtig als Kriterium 2	1:1

Tabelle 6.2. – Paarweiser Vergleich und somit Gewichtung der Faktoren für die Testqualität.

bewertung gravierende Unterschiede in ihrem Beitrag aufweisen. Im Vergleich zu den Faktoren für die Komplexitätsbewertung zeigt sich aber, dass die Unterschiede nicht so hoch sind: sechs der acht ausgewählten Faktoren haben jeweils über 10% Beitrag zur Gesamtbewertung.

6.1.3. Berechnung der Testqualität

Die Berechnung der Testqualität kann nun anhand der beeinflussenden Faktoren und der zugehörigen Gewichtung erfolgen. Durch ein Expertenaudit wird das Projekt hinsichtlich der Faktoren mithilfe der Bewertungsskala begutachtet. Die Qualitätszahl Q eines Projekts errechnet sich für jeden der n_Q Faktoren mit dem gewichteten Einfluss $G_{Q,i}$ und der zugehörigen Bewertung $B_{Q,i}$ für die Qualität zu:

$$Q = \sum_{i=1}^{n_Q} G_{Q,i} \cdot B_{Q,i} \quad (6.1)$$

Je näher die Qualitätszahl Q für ein Projekt an fünf ausfällt, als umso qualitativ hochwertiger ist der Test zu betrachten. Grundsätzlich ist ein überdurchschnittlicher Wert ($>2,5$) erstrebenswert.

6.1.4. Beispiel

Nachfolgend wird anhand der drei bereits in Kapitel 3.4 aktuell laufenden Projekte der integralen Fahrzeugsicherheit die Testqualitätsbewertung durchgeführt. Hierfür wurden zwei Expertenaudits im Abstand von 14 Monaten durchgeführt. Die erste Bewertung fand jeweils circa nach einem Drittel der Laufzeit und die zweite Bewertung nach zwei Dritteln statt. Tabelle 6.3 zeigt die Ergebnisse der Expertenaudits. Deutlich wird, dass sich alle drei Projekte über den Zeitraum zwischen den beiden Expertenaudits verbessert haben. Zwar haben sich vereinzelt die Bewertungen verschlechtert (z. B. Erfahrung der Tester durch Personalwechsel), insgesamt ist die Testqualität jedoch erhöht worden, sie ist bei allen drei Projekten überdurchschnittlich gut ($>2,5$) und somit zufriedenstellend.

6.2. Testüberdeckung

Testüberdeckungen werden häufig als Testendekriterium verwendet. Während im White-Box-Softwareentwicklungsstadium von ISO 26262 z. B. Metriken für Anweisungs-, Zweig-, Entscheidungs-, Funktions- und Aufrufüberdeckung gefordert werden, wird keine Aussage für eine Metrik hinsichtlich des Systemtests gemacht [57], [78]. Die Überdeckungen aus dem Softwarebereich können im Systemtest aufgrund der Blackbox-Eigenschaft des Tests nicht mehr oder kaum verwendet werden. Allgemein gibt [118] für den Systemtest folgende Überdeckungen als mögliche Ansätze an: Code-, Architektur-, Daten-, Funktions- und Anforderungsüberdeckung. Code-, Architektur- und Datenüberdeckung sind im Systemtest eines Steuergeräts aus technischen Gründen nur sehr beschränkt möglich (Black-Box). Falls Grey-Box-Testtechniken möglich sind, können Funktions- und Anforderungsüberdeckung je nach Projektbeschaffenheit verwendet werden. Des Weiteren können Maße wie Fehlerdichte oder Fehlerüberdeckung definiert werden, die auf der Anzahl der aktuell im Projekt gefundenen Fehlern beruhen und so Rückschlüsse auf den Stand des Testens zulassen.

6.2.1. Funktions- bzw. Funktionalitätsüberdeckung

Bei der Funktions- bzw. Funktionalitätsüberdeckung wird das System in seine Funktionen gegliedert und es werden so lange Testfälle ausgeführt, bis die Funktion vollständig getestet wurde. Häufig wird es dabei mit der Funktionalität eines vorherigen Systems verglichen, die das neue System ebenfalls oder weiterhin erfüllen muss. Dieses Kriterium wird jedoch aufgrund des Aufwands zur Kapselung der Funktionen hauptsächlich für Regressionstests verwendet [96]. Grundsätzlich lässt sich das die Funktions- bzw. Funktionalitätsüberdeckung nach [118] abbilden:

$$\text{Funktionsüberdeckung} = \frac{\text{Erfolgreich getestete Anwendungsfallvarianten}}{\text{Gesamtzahl Anwendungsfallvarianten}} \quad (6.2)$$

Betrachtet werden hier also die Anwendungsfälle des Systems [126]. Ein Anwendungsfall besteht aus Aktionen, Bedingungen und erwarteten Ergebnissen. Z. B. könnte ein Anwendungsfall ein Bremsmanöver auf Eis sein, bei dem bei bestimmten Signalzuständen ein

	Qualität der Testspezifikation, $E_{Q,i} = 17,9\%$	Reifegrad vorhergehender Tests (SW,HW), 16,1%	Erfahrung der Tester, 16,1%	Reviews der Tests, 14,3%	Dokumentation der Testaktivitäten, 14,3%	Automatisierungsgrad der Testausführung, 12,5%	Anzahl und Verfügbarkeit der Tester und Tools, 7,1%	Zuverlässigkeit des Testsystems, 1,8%	Q
Audit 1									
$B_{Q,i}$ Projekt 6	4	2,5	3,5	1	4	4	4	4	3,27
$B_{Q,i}$ Projekt 7	4	2,5	3,5	2	4	4	4	4	3,41
$B_{Q,i}$ Projekt 8	4	3,5	4	1	4	4	5	4	3,58
Audit 2									
$B_{Q,i}$ Projekt 6	4	3,5	3,5	2,5	4	4	4	4	3,64 ↑
$B_{Q,i}$ Projekt 7	4	3,5	3	3	4	4	4	4	3,63 ↑
$B_{Q,i}$ Projekt 8	4	3,5	3	3	4	4	5	4	3,70 ↑

Tabelle 6.3. – Testqualitätsaudits für drei aktuell laufende Projekte der integralen Fahrzeugsicherheit im Abstand von 14 Monaten.

System auslösen muss (Fall des reversiblen Gurtstraffers aus Kapitel 4). Als Testmethode kann hierfür das manöverbasierte Testen auf Basis von Fahrdynamik- und Umfeldsimulation aus Kapitel 4.2 dienen. Durch den Test des Systems hinsichtlich seiner Anwendungsfälle kann ein realitätsnahes Abbild des Systemverhaltens geschaffen werden.

Aufgrund des aktuell geringen Einsatzes des Tests der Anwendungsfälle und der somit kaum erfassten Funktionsüberdeckung in der integralen Fahrzeugsicherheit, kommt dieses Maß noch nicht umfassend zum Einsatz. Zukünftig wird diese Methode aufgrund der fortschreitenden Integration von vorausschauender Sensorik und von Fahrerassistenzsystemen an Bedeutung gewinnen, da diese Systeme hauptsächlich anwendungsfallbasiert getestet werden.

6.2.2. Anforderungsüberdeckung

Die Anforderungsüberdeckung stellt eine funktionale Überdeckung dar, die auf der Anforderungsdokumentation beruht [16]. Jede Anforderung muss hierfür mit einem oder mehreren Testfällen verknüpft sein, um ihre Erfüllung zu testen. Die Überdeckung misst dann, welche dieser Testfälle ausgeführt wurden und welches Ergebnis erzielt wurde. Eine Anforderung gilt erst als getestet, wenn mindestens ein Testfall für sie erfolgreich ausgeführt wurde. Das Maß für die Anforderungsüberdeckung lässt sich folgendermaßen abbilden:

$$\text{Anforderungsüberdeckung} = \frac{\text{Erfolgreich getestete Anforderungen}}{\text{Gesamtzahl Anforderungen}} \quad (6.3)$$

Der Anteil der ausgeführten funktionalen Testfälle entspricht dann der funktionalen Testüberdeckung. Das System wird hier gegen die Systembeschreibung getestet [118].

Zur Festlegung eines Testendekriteriums eignet sich die Anforderungsüberdeckung aktuell am besten, da sie etabliert ist und leicht über Tools erfassbar ist. Hierbei wird gegen die Anforderungen getestet und die Funktionalität des Systems sichergestellt. Sie stellt ein gut nachvollziehbares und leicht dokumentierbares Maß dar. Nachfolgend soll aus diesem Grund die Anforderungsüberdeckung als Testüberdeckungsmaß dienen.

6.2.3. Fehlerdichte und Fehlerüberdeckung

Die Fehlerdichte stellt ein Maß zur Fehleranalyse dar. Hierbei wird nach [118] folgende Abbildung verwendet:

$$\text{Fehlerdichte} = \frac{\text{Fehleranzahl}}{\text{Gesamtzahl ausgeführte Testfälle}} \quad (6.4)$$

Je weniger Fehler durch die Testfälle aufgedeckt werden, desto besser ist die Systemqualität. Man kann auch von geringer Fehlerdichte sprechen. Hierbei kann die Fehleranzahl noch gewichtet werden, d. h. schwerwiegende Fehler gehen stärker ein als unkritische. Hierbei muss allerdings auf eine exakte und vollständige Dokumentation des Fehlerfindungsprozesses geachtet werden.

Eine weitere Möglichkeit der Fehlerüberdeckung ist durch die Fehlerschätzung aus Kapitel 3.3 gegeben. Durch die Bewertung und den Vergleich mit vorherigen Projekten kann ein erwarteter Fehlerwert für ein Projekt abgeschätzt werden. Darstellen lässt sich dieses Maß zu:

$$\text{Fehlerüberdeckung} = \frac{\text{Fehleranzahl}}{\text{Schätzwert Fehleranzahl}} \quad (6.5)$$

Je größer der Wert für die Fehlerüberdeckung wird, desto vergleichbarer ist die Rate der Fehleraufdeckung zu anderen Projekten. D. h. es wurden bei einer Fehlerüberdeckung von 1 entsprechend der ermittelten Komplexität alle erwarteten Fehler gefunden. Bei einer Zahl größer als 1 wurde das Ziel im Vergleich zu anderen Projekten übertroffen.

6.2.4. Beispiele für Testüberdeckungen

Als Beispiel für die Testabdeckung soll nachfolgend die Anforderungsüberdeckung, die Fehlerdichte und die Fehlerüberdeckung für die bereits bekannten aktuell laufenden Projekte aus den Kapiteln 3.4 und 6.1.4 dienen. Die Funktions- bzw. Funktionalitätsüberdeckung kann aufgrund der nicht vorhandenen Datenlage nicht eingesetzt werden, sollte aber zukünftig aufgrund ihrer Leistungsfähigkeit in Betracht gezogen werden.

Anforderungsüberdeckung Abbildung 6.2 zeigt für die drei ausgewählten Beispielprojekte den Stand der Anforderungstests. Deutlich wird hier die Trennung zwischen den

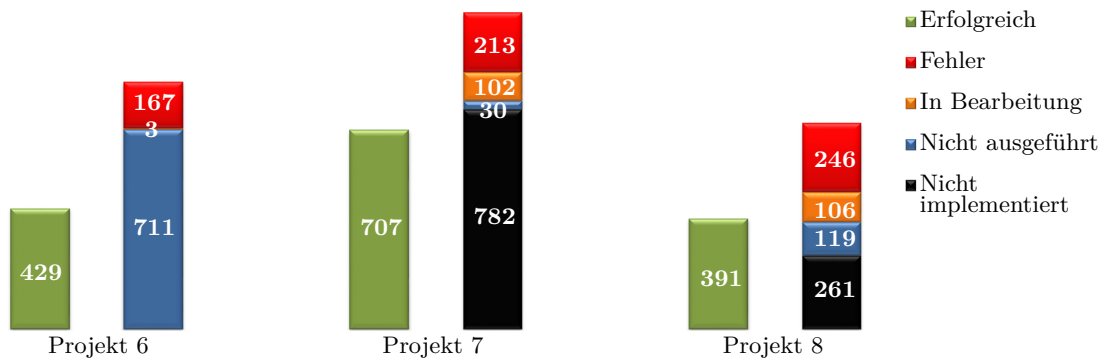


Abbildung 6.2. – Stand der Anforderungstests der drei ausgewählten Projekte nach zwei Dritteln der Projektlaufzeit.

erfolgreich getesteten und nicht getesteten oder fehlerhaften Anforderungen. Die Berechnung der prozentualen Anforderungsüberdeckungen erfolgt in Tabelle 6.4. Es zeigt sich, dass bei allen drei Projekten ungefähr ein Drittel der Anforderungen erfolgreich getestet wurden.

Projekt:	Erfolgreich getestete Anforderungen	/	Gesamtzahl Anforderungen	=	Anforderungs- überdeckung
Projekt 6:	429	/	1310	=	33%
Projekt 7:	707	/	1834	=	39%
Projekt 8:	391	/	1123	=	35%

Tabelle 6.4. – Aktueller Stand der Anforderungsüberdeckung für die drei ausgewählten Projekte nach circa zwei Dritteln der Projektlaufzeit.

Fehlerdichte Die Fehlerdichte gibt die Testeffektivität wieder. Für die drei laufenden Projekte wurden nach circa zwei Dritteln der Projektlaufzeit die Fehlerdichten aus Tabelle 6.5 ermittelt. Die Fehlerdichte sagt aus, dass bei Projekt 6 in 15% aller ausgeführten Testfälle, bei Projekt 7 in 21% aller ausgeführten Testfälle und bei Projekt 8 in 45% aller ausgeführten Testfälle Fehler gefunden wurden. Die Testeffektivität ist somit in Projekt 8 höher als bei den Projekten 7 und 6, jedoch werden pro ausgeführtem Testfall noch sehr viele Fehler gefunden, was die Notwendigkeit weiterer Tests begründet.

Projekt:	Fehleranzahl	/	Gesamtzahl ausgeführte Testfälle	=	Fehlerdichte
Projekt 6:	264	/	1736	=	15%
Projekt 7:	269	/	1296	=	21%
Projekt 8:	142	/	313	=	45%

Tabelle 6.5. – Stand der Fehlerdichte für die drei ausgewählten Projekte nach circa zwei Dritteln der Projektlaufzeit.

Fehlerüberdeckung Die Fehlerüberdeckung hängt eng mit der Komplexitätsbewertung und der daraus resultierenden Fehlerschätzung aus Kapitel 3 zusammen. Für die laufenden Projekte wurden nach circa zwei Drittel der Projektlaufzeit die Fehlerüberdeckungen aus Tabelle 6.6 ermittelt. Deutlich wird, dass Projekt 7 bereits nah an den Schätzwert heranreicht, während Projekt 6 erst 30% des erwarteten Wertes erreicht hat. Projekt 8 hingegen hat den Vergleich mit anderen Projekten bereits übertroffen: es wurden mehr Fehler als erwartet gefunden, d. h. dass mehr Fehler aufgedeckt wurden als es in einem vergleichbar komplexen Projekt bisher der Fall war. Dies zeigt, dass der Verlauf der Fehlerfindung sehr unterschiedlich ausfallen kann. Die Ergebnisse der Fehlerüberdeckung sollten nach Abschluss der Projekte verwendet werden, um die Fehlerschätzung aus Kapitel 3.3 zu präzisieren.

Wichtig für die spätere Bewertung ist, dass die Fehleranzahl zum Ende des Projekts hin zumindest nah an die erwartete Schätzzahl herankommt oder sie übertrifft. Nach Ende des Projekts können die absoluten Fehlerzahlen dann zur Verbesserung der Kurve aus Abbildung 3.4 bzw. der Formel 3.2 verwendet werden.

Projekt:	Fehleranzahl	/	Schätzwert Fehleranzahl	=	Fehler- überdeckung
Projekt 6:	264	/	888	=	30%
Projekt 7:	269	/	315	=	85%
Projekt 8:	142	/	132	=	108%

Tabelle 6.6. – Stand der Fehlerüberdeckung für die drei ausgewählten Projekte nach circa zwei Dritteln der Projektlaufzeit.

6.3. Der Vertrauensgrad als Testmetrik

Wie bereits in Kapitel 2.4 erwähnt, ist nach [120] eines der wichtigsten Ziele des Systemtests Vertrauen zu schaffen. Um nun das Vertrauen zu „messen“ schlägt [118] eine Metrik vor, die Vertrauensgrad genannt wird. Für den Vertrauensgrad nach dieser Definition wird die Fehlerdichte und die Testüberdeckung, wie sie durch Anforderungs- oder Funktionsüberdeckung ausgedrückt wird, verwendet:

$$\begin{aligned} \text{Vertrauensgrad nach Sneed} &= \left(1 - \frac{\text{Fehleranzahl}}{\text{Gesamtzahl ausgeführte Testfälle}}\right) \cdot \text{Testüberdeckung} \\ &= (1 - \text{Fehlerdichte}) \cdot \text{Testüberdeckung} \end{aligned} \quad (6.6)$$

Hier ist der Vertrauensgrad das Verhältnis der gefundenen Fehler zur Anzahl der ausgeführten Testfälle multipliziert mit der Testüberdeckung. Der Vertrauensgrad ist also dann hoch, wenn die Fehlerdichte niedrig und die Testüberdeckung hoch sind. Diese Definition des Vertrauensgrades soll nachfolgend modifiziert und um weitere Aspekte zu einer noch leistungsfähigeren und für die Fahrzeugsicherheit angepassten Testmetrik erweitert werden. Anstatt der Berechnung durch Multiplikation, die die Einzelmaße nicht gleichförmig einbringt, soll nachfolgend eine getrennte Betrachtung der Fehlerdichte und der Testüberdeckung erfolgen.

6.3.1. Zusammenführung der Maße

Anders als bei [118] vorgeschlagen soll der Vertrauensgrad nicht nur aus Fehlerdichte und Testüberdeckung bestehen, sondern aus der getrennten Betrachtung von Fehlerdichte (Kapitel 6.2.3) und Testüberdeckung (in erster Linie Anforderungsüberdeckung aus Kapitel 6.2.2) sowie durch die Fehlerüberdeckung aus der Komplexitätsbewertung (Kapitel 3.3) und dem Expertenaudit zur Testqualitätsbewertung (Kapitel 6.1) erweitert werden.

Abbildung 6.3 zeigt die vorgeschlagene Testmetrik. Der Vertrauensgrad besteht aus vier Einzelmaßen: der Fehlerüberdeckung aus der Komplexitätsbewertung, dem Expertenaudit zur Testqualitätsbewertung, der Fehlerdichte und der Testüberdeckung. Den vier Komponenten wird jeweils eine Ampel zugewiesen, die drei mögliche Status haben können: rot (nicht in Ordnung, kritisch), gelb (Achtung, Gründe sollten analysiert werden) und grün (in Ordnung). Vorgeschlagen wird die Ermittlung der Gesamtbewertung des Vertrauensgrades nach folgendem Schema:

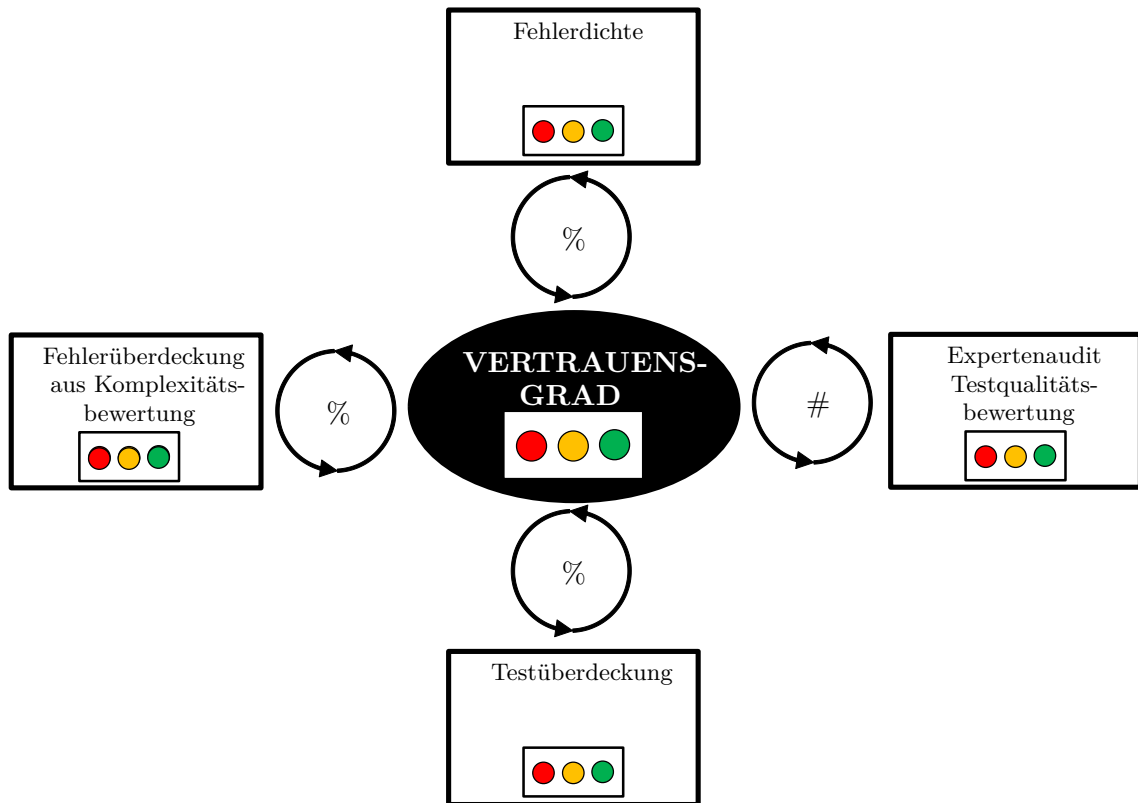


Abbildung 6.3. – Ermittlung des Vertrauensgrad anhand der Fehlerüberdeckung aus Komplexitätsbewertung, Expertenaudit Testqualitätsbewertung und Fehlerdichte sowie Testüberdeckung. Vorgeschlagen wird ein Ampelsystem, das als Vertrauensgrad eine Gesamtbewertung wiedergibt, die aus den vier Einzelbewertungen der Maße besteht.

- Sobald ein Einzelmaß rot ist, ist der Vertrauensgrad rot
- Nur wenn alle Einzelmaße grün sind, ist der Vertrauensgrad grün
- Bei einer Mischung aus gelb und grün bei den Einzelmaßen ist der Vertrauensgrad gelb

Die Farbbewertungen der Einzelmaße erfolgt nach Tabelle 6.7. Mithilfe der Metrik des

Maß	rot ●	gelb ●	grün ●
Expertenaudit	$0 < Q < 2$	$2 \leq Q \leq 3$	$3 < Q < 5$
Testqualitätsbewertung			
Fehlerüberdeckung aus Komplexitätsbewertung	$< 70\%$	$70\% - 90\%$	$> 90\%$
Fehlerdichte	$> 50\%$	$35\% - 50\%$	$< 35\%$
Testüberdeckung	$< 85\%$	$85\% - 95\%$	$> 95\%$

Tabelle 6.7. – Vorschlag zur Ermittlung der Status der Einzelmaße. Je nach Umfeld und Projektgegebenheiten können die Grenzen anders festgelegt werden.

erweiterten Vertrauensgrades lässt sich auf einen Blick der aktuelle Stand des Testens

ablesen. Durch die Ampeldarstellung ist die Metrik leicht verständlich und durch die dahinterstehenden Einzelmaße lässt sich die Ursache einer schlechten Vertrauensbewertung leicht lokalisieren und eruieren. Da die Testüberdeckungen und Fehleranzahl auch heute schon leicht und automatisiert ermittelt werden können, erzeugt der Einsatz des Vertrauensgrades bis auf die Expertenaudits keinen Zusatzaufwand. Der Vertrauensgrad ist auch leicht bei Bedarf um weitere Einzelmaße erweiterbar und kann sich so ohne großen Aufwand neuen Gegebenheiten anpassen.

6.3.2. Beispiel

Nachfolgend soll anhand der drei laufenden Projekte jeweils der Status des Vertrauensgrades dargestellt werden. In Tabelle 6.8 sind die Ergebnisbewertungen der Projekte dargestellt. Auf einen Blick ist erkennbar, dass alle drei Projekte noch keinen ausreichenden Vertrau-
















Maß	Projekt 6	Projekt 7	Projekt 8
Expertenaudit	3,64 	3,63 	3,70 
Testqualitätsbewertung			
Fehlerüberdeckung aus Komplexitätsbewertung	30% 	85% 	108% 
Fehlerdichte	15% 	21% 	45% 
Testüberdeckung	33% 	39% 	35% 
Vertrauensgrad			

Tabelle 6.8. – Der Vertrauensgrad für die drei ausgewählten laufenden Projekte nach circa zwei Dritteln der Projektlaufzeit.

ensgrad erreicht haben. Dies liegt bei allen drei Projekten an der noch nicht erfüllten Testüberdeckung. Projekt 6 hat zusätzlich die Fehlerüberdeckung aus der Komplexitätsbewertung noch nicht erfüllt. Projekt 7 ist bei diesem Einzelmaß im gelben Bereich, d. h. es hat die Anforderungen fast erreicht. Die Fehlerdichte bzw. Testeffizienz sowie die Testqualität ist bei den Projekten jeweils im grünen Bereich. Projekt 8 hat eine zu hohe Fehlerdichte, was die Notwendigkeit weiterer Tests begründet. Die Fehlerüberdeckung wurde bei Projekt 8 hingegen als einziges bereits erfüllt. Die erwarteten Fehler wurden übertroffen.

Es zeigt sich hier anschaulich, dass durch den Vertrauensgrad eine Übersicht und ein Vergleich zwischen Projekten auf einen Blick möglich ist. Auch die Ursachenlokalisierung und -eruierung ist sehr schnell möglich. Gerade wenn noch die gesamte Testqualitätsbewertung hinzugezogen wird, ergibt sich ein übersichtliches Gesamtbild über den aktuellen Projekt- und Systemteststand.

Fazit

In diesem Kapitel wurde eine Metrik zur Testfortschrittsbewertung vorgestellt. Die Testqualität kann durch ein Expertenaudit ermittelt werden, das anhand gewichteter Faktoren den

aktuellen Stand des Testens abbildet. Des Weiteren stellen Testüberdeckungen ein wichtiges Maß zur Testbeurteilung dar. Um einen umfassenden Eindruck vom momentanen Testfortschritt zu erhalten, werden die Testfortschrittsbewertung, die Testüberdeckungen und die Fehlerüberdeckung aus Kapitel 3 zum sogenannten Vertrauensgrad zusammengefasst. Der Vertrauensgrad gibt durch eine Ampeldarstellung kurz, präzise und leicht verständlich den aktuellen Stand des Testens wieder. Er wurde als flexibel erweiterbares Maß gestaltet, das auch die Ursachenlokalisierung und -erweiterung einfach ermöglicht und so als leistungsfähige Metrik im Systemtest eingesetzt werden kann.

7. Zusammenfassung und Ausblick

In diesem Kapitel wird der Inhalt zusammengefasst, der Beitrag der Arbeit zum Stand der Technik dargestellt und ein Ausblick für zukünftige Arbeiten gegeben.

7.1. Zusammenfassung

In dieser Arbeit wurde eine Methodik für den Systemtest in der integralen Fahrzeugsicherheit erarbeitet und anhand aktueller Projekte und Systeme der Automobilindustrie erprobt. Hierfür wurde zunächst in Kapitel 1 das aktuelle Verkehrs- und Unfallgeschehen dargestellt sowie die Zielsetzung und der Aufbau der Arbeit festgelegt. Nachfolgend wurden in Kapitel 2 die Grundlagen für die Arbeit festgehalten. Hierfür wurde zunächst auf die integrale Fahrzeugsicherheit eingegangen und anschließend das Testobjekt Steuergerät beschrieben. Der wichtige Standard ISO 26262 wurde ebenfalls diskutiert. Anschließend wurde der Systemtest in den Entwicklungs- und Testprozess der Automobilbranche eingegliedert. Anhand dieser Grundlagen wurde dann die Notwendigkeit der Arbeit diskutiert.

Ausgehend aus den Grundlagen wurde in Kapitel 3 eine Methode zur Projektkomplexitätsbewertung vorgestellt. Hierfür wurden ein sogenanntes Expertenaudit entworfen, das eine Projektbewertung anhand von gewichteten Einflussfaktoren ermöglicht. Dadurch konnte nachfolgend anhand der Analyse vergangener Projekte eine Fehlerschätzung für laufende Projekte abgegeben werden. Es wurde also eine Grundlage geschaffen, um Projekte untereinander zu vergleichen und somit das Testverhalten vorherzusagen. Durch die Komplexität eines Projekts können zwei Maßnahmen abgeleitet werden: eine Vorhersage der zu erwartenden Fehler im Systemtest und die Auswahl von Testmethoden. Durch Erprobung anhand abgeschlossenen und laufenden Projekten wurde einer Fehlerapproximation aufgezeigt. Anschließend wird in Kapitel 4 die Testauswahl anhand der Projektkomplexität diskutiert. Durch den Einsatz von effizienzsteigernden, ergänzenden Methoden können komplexe Systeme effektiver getestet werden. Es wurden dann neben den aktuellen Testmethoden aus ISO 26262 insgesamt drei neuartige Testmethoden vorgestellt: manöverbasiertes Testen durch Fahrdynamik- und Umfeldsimulation, statistische Analyse von Testfällen und der Einsatz eines Evolutionären Algorithmus. Der Einsatz der drei Testmethoden wurde anhand des integralen Fahrzeugsicherheitssystem des reversiblen Gurtstraffers erprobt und ihre Wirksamkeit aufgezeigt. Durch die Methoden können Testfälle flexibler erstellt, ihr Verhalten genauer analysiert und hin zu gewünschten Testeigenschaften optimiert werden. Somit kann ein verbesserter Test von komplexen Funktionen gewährleistet werden.

Kapitel 4.5 gab anschließend eine Zusammenfassung über mögliche Testfallpriorisierungen im Bereich der Fahrzeugsicherheit. Anhand verschiedener Kriterien lassen sich Testfälle ordnen und so Fehler schneller gefunden und das Erreichen einer hohen Testüberdeckung beschleunigt werden. Außerdem kann so rascher Vertrauen in das System geschaffen und Ressourcen können besser ausgelastet werden.

Zur Ausführung der Testmethoden bzw. -fälle wurde in Kapitel 5 ein Testsystem vorgestellt, das zunächst anhand von Anforderungen, die an eine Testumgebung in der integralen Fahrzeugsicherheit diskutiert wurde. Durch den Einsatz von Fahrdynamiksimulation in Kombination von Crashdateneinspeisung können durchgängige Testfälle erstellt werden, die alle Unfallphasen umfassen. Der neuartige Ansatz lässt es zum ersten Mal zu, dass ein System im Laborbetrieb am HiL sowohl die Phasen der Normalfahrt und der Pre-Crash-Phase als auch die Unfallphase durchlebt. Abschließend wurde eine Beispielkonfiguration

eines Testsystems aufgezeigt.

Zur Testfortschrittsbewertung wurde in Kapitel 6 eine Testmetrik vorgestellt. Die Metrik umfasst zum einen ein Expertenaudit zur Testqualitätsbewertung, das durch gewichtete Einflussfaktoren eine Bestimmung der aktuellen Testqualität erlaubt. Des Weiteren werden Testüberdeckungen inkludiert, die sowohl Funktions- als auch Anforderungsüberdeckung umfassen. Außerdem werden noch die Fehlerdichte und die Fehlerüberdeckung, die auf der Fehlerschätzung basierend auf der Komplexitätsbewertung basiert, vorgestellt. All diese Maße werden mithilfe eines Ampelbewertungssystems zum sogenannten Vertrauensgrad zusammengefasst, der zulässt, den aktuellen Testfortschritt auf einen Blick zu erfassen. Die Metrik wurde aktuell laufender Projekte der integralen Fahrzeugsicherheit erprobt.

7.2. Wissenschaftlicher Beitrag dieser Arbeit

Nachfolgend wird in Tabelle 7.1 dargelegt, welchen Beitrag zum Stand der Technik die jeweiligen Kapitel liefern. Die einzelnen Teile beziehen sich jeweils auf die Schritte der vorgeschlagenen Methodik in Abbildung 7.1.



Abbildung 7.1. – Übersicht über die in der Arbeit vorgeschlagene Methodik (siehe auch Abbildung 1.3).

Kapitel	Delta und Beitrag zum Stand der Technik
Gesamt	<ul style="list-style-type: none"> • Methodik für den Systemtest integraler Fahrzeugsicherheitssysteme • Gesamtheitlicher Ansatz über den ganzen Testprozess hinweg • Analyse des Testfortschritts mithilfe von Testmetriken • Erprobung von innovativen Methoden zur Testfallanalyse und -optimierung
Kapitel 3 / Teil 1	<ul style="list-style-type: none"> • Einführung des Expertenaudits zur Komplexitätsbewertung von Projekten der integralen Fahrzeugsicherheit • Abschätzung der zu erwartenden Fehler im Systemtest anhand der Komplexität und Fehler vergangener Projekte • Erprobung des Vorgehens anhand realer Projekte der integralen Fahrzeugsicherheit

Kapitel	Delta und Beitrag zum Stand der Technik
Kapitel 4 / Teile 2 u. 3	<ul style="list-style-type: none"> • Neuartige Ableitung von Testmethoden anhand der Projektkomplexität • Erstmalige Kombination von simulativem, manöverbasiertem Testen mit analytischen und optimierenden Methoden zur Erleichterung der Testfallerstellung bei wenig Knowhow im Bereich Fahrdynamik • Manöverbasierte Testfälle werden erstmals durch statistische Methoden analysiert, um ihr Verhalten vorherzusagen • Evolutionäres Testen wird erstmals in Kombination mit manöverbasiertem Testen bzw. Fahrdynamik- und Umfeldsimulation eingesetzt, um optimierte bzw. charakterisierte Testfälle zu erhalten • Evolutionäres Testen wird erstmals für den Systemtest hochvernetzter Sicherheitssysteme im Automobilbereich eingesetzt • Erprobung der Methoden anhand des integralen Fahrzeugsicherheitssystems des reversiblen Gurtstraffers • Umfassende Zusammenfassung zur möglichen Testfallpriorisierung zur schnelleren Erreichung von Testüberdeckung und Fehleraufdeckung • Adaption der Priorisierungsparameter an die integrale Fahrzeugsicherheit
Kapitel 5 / Teil 4	<ul style="list-style-type: none"> • Umfassende Darstellung der notwendigen Komponenten eines Testsystems für die integrale Fahrzeugsicherheit • Kombination aus Fahrdynamik- und Umfeldsimulation mit Crashdateneinspeisung • Möglichkeit der Erweiterung und Integration neuer, leistungsfähiger Methoden
Kapitel 6 / Teil 5	<ul style="list-style-type: none"> • Testmetrik „Vertrauensgrad“ mit verschiedenen Testüberdeckungsmaßen und Bewertung des Testfortschritts anhand eines Expertenaudits • Analyse und Erprobung der Metrik anhand acht Projekten

Tabelle 7.1. – Beitrag der Arbeit im Vergleich zum Stand der Technik.

7.3. Ausblick

Ausgehend von den Ergebnissen dieser Arbeit entstehen Ansatzpunkte für die weitere Forschung und Erprobung für den Systemtest in der integralen Fahrzeugsicherheit. Hinsichtlich der Testmetrik könnte sowohl bei den Expertenaudits als auch bei der Testüberdeckungs- und Fehlerauswertung eine länger und intensiver angelegte Untersuchung genauere Ergebnisse liefern. Hierbei könnte vor allem der Zusammenhang zwischen der Komplexität und den Fehlern im Systemtest genauer erforscht werden. Durch die Einbeziehung der Fehlerschwere und somit der Möglichkeit der Gewichtung von Fehlern, könnte eine verbesserte Vorhersage für die Anzahl und den Verlauf der Fehler im Systemtest getroffen werden.

Im Bereich der Testmethoden können statistische Testverfahren einen weiteren Beitrag

zum Test komplexer Systeme liefern. Durch den Einsatz von Zufallstests und Monte-Carlo-Simulation könnte eine weitere Effizienzsteigerung des Testens erreicht werden.

Hinsichtlich der Evolutionären Algorithmen kann neben Differential Evolution eine genauere Analyse und Erprobung weiterer Algorithmen erfolgen. Eine Untersuchung des Testfallverhaltens für mehrmalige Wiederholung der evolutionären Optimierung wäre ebenfalls erstrebenswert, um den Optimierungsvorgang genauer zu verstehen. Des Weiteren ist eine Parallelisierung der Simulation der Einzelindividuen sinnvoll, um die Optimierungszeit zu verkürzen. Da die Individuen unabhängig von einander simuliert werden, kann eine parallele Ausführung die Erstellungszeit für eine Generation stark verkürzen. Gerade bei umfangreichen Optimierungen mit vielen Parametern liefert eine Parallelisierung einen großen Nutzen.

Der Einsatz von modellbasiertem Entwickeln und Testen ist eine wichtige Modifikation, die im Bereich des durchgängigen Testens viele Vorteile in sich birgt. Durch die Verwendung eines System- und eines Testmodells lassen sich Testfälle anschaulicher ableiten und über MiL-, SiL- und HiL-Stadium einfach wiederverwenden.

Die Adaption der Testmethoden auf vorausschauende Sensortechniken und den damit verbunden Herausforderungen wird ebenfalls zukünftig aufgrund des verstärkten Einsatzes notwendig sein. Hierbei spielt das manöverbasierte Testen eine große Rolle, da z. B. für Kamerasysteme nur manöverbasiert getestet werden kann. Auch hier würde eine Umstellung von der Einspeisung real aufgezeichneter Fahrten hin zu simulierten Manövern die Vorteile der Flexibilität, Analysier- und Optimierbarkeit liefern.

A. Anhang - Statistische Versuchsplanung: Einzelauswertungen der Faktoren nach Faktorstufen

Nachfolgend werden die Einzelauswertungen der Faktoren von Abbildung 4.13 aus Kapitel 4.3 nach Faktorstufen dargestellt.

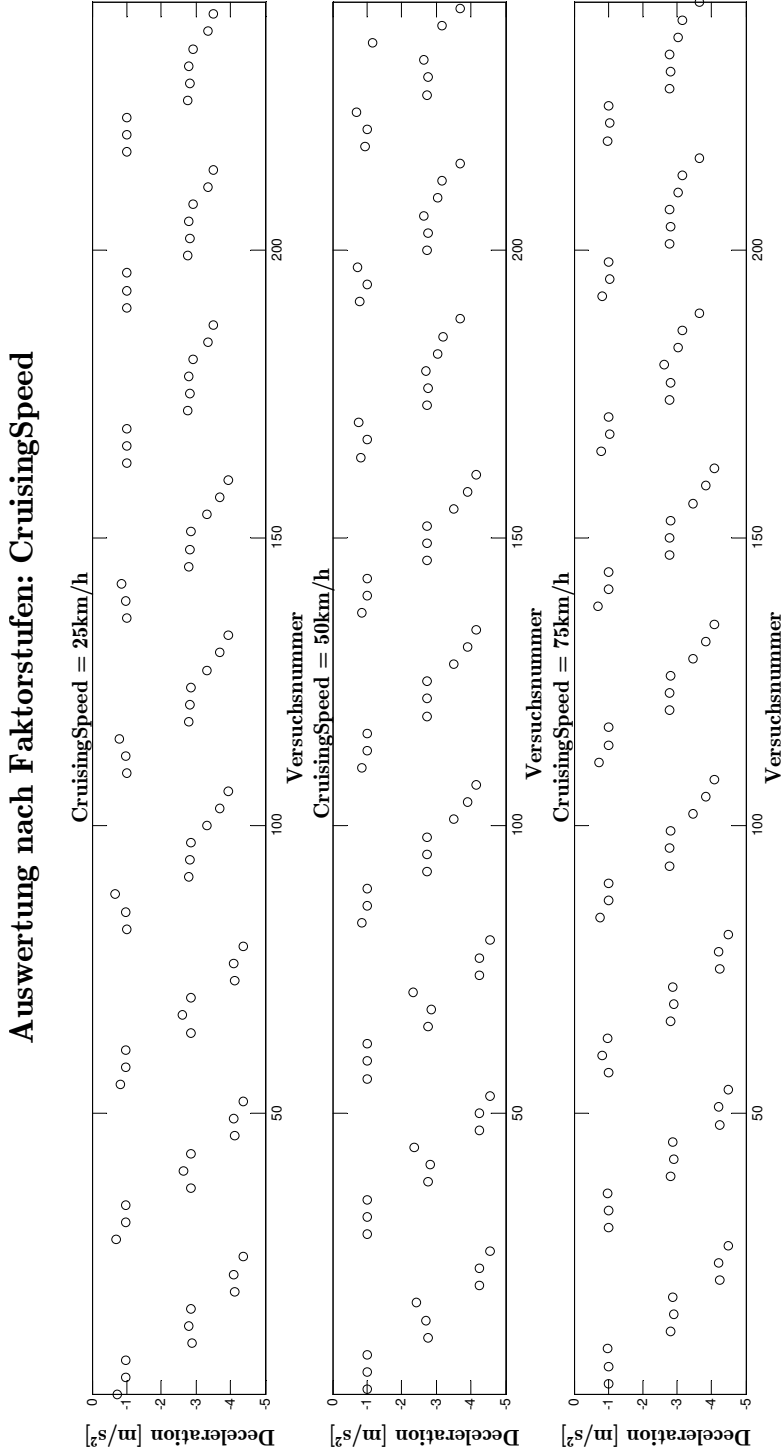


Abbildung A.1. – Einzelauswertung des Faktors CruisingSpeed nach Faktorstufen

Auswertung nach Faktorstufen: Brake

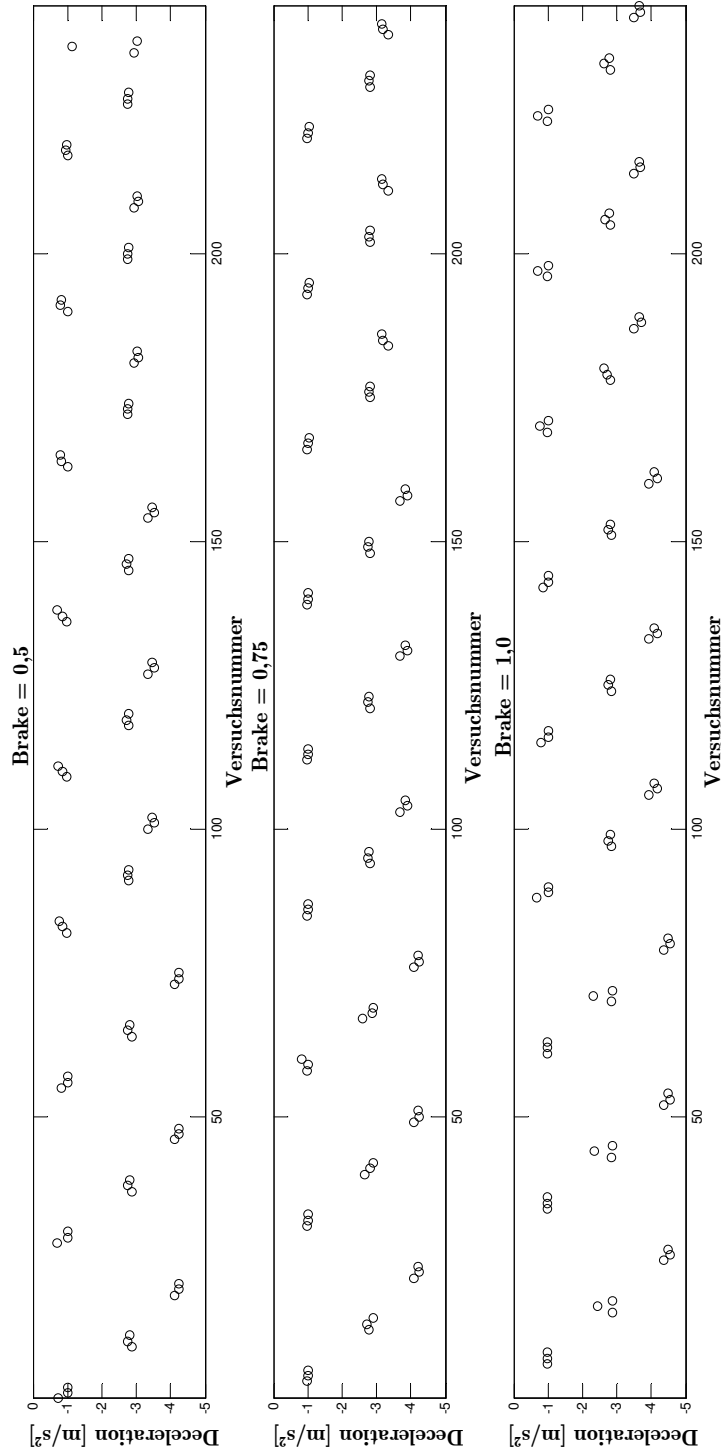


Abbildung A.2. – Einzelauswertung des Faktors Brake nach Faktorstufen

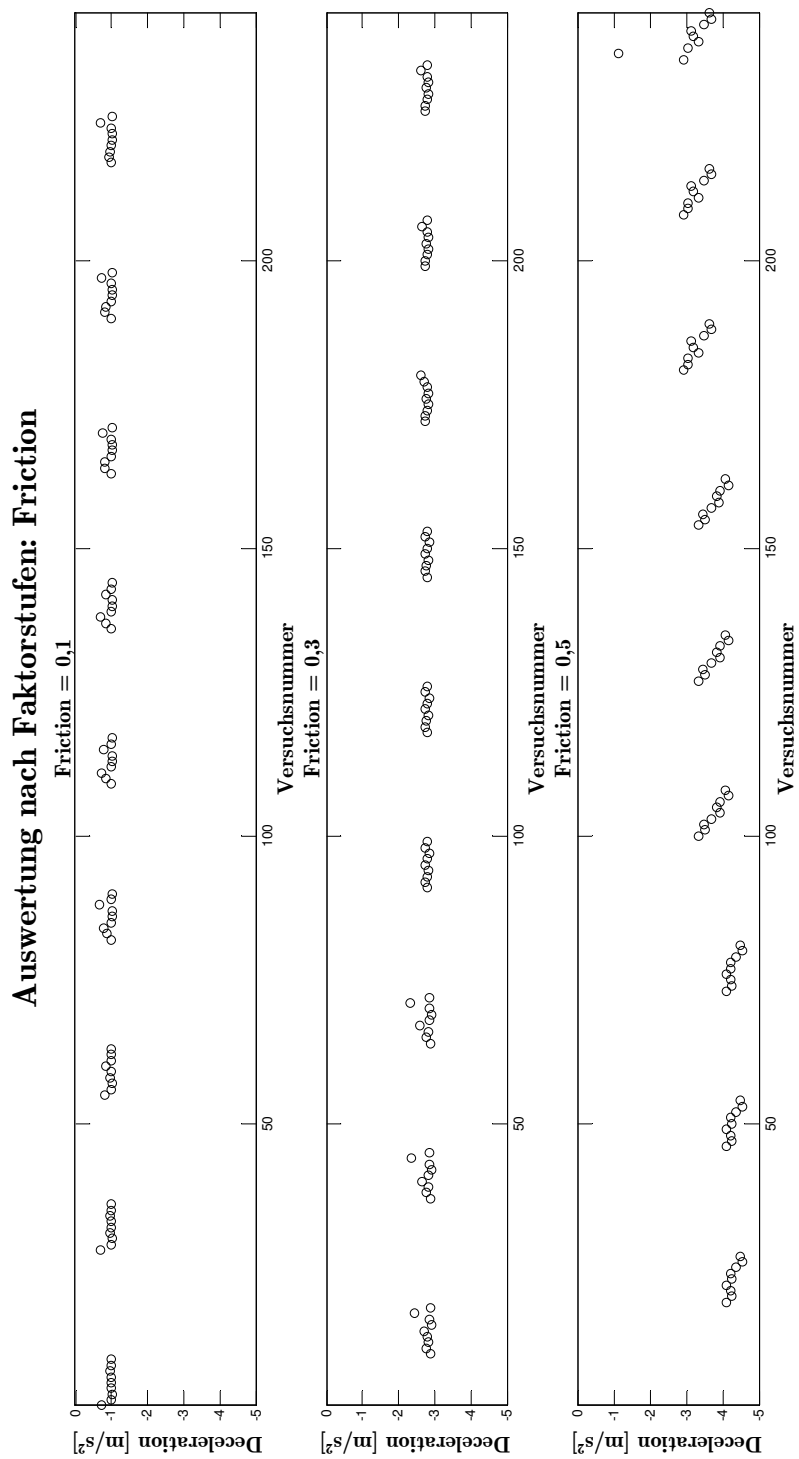


Abbildung A.3. – Einzelauswertung des Faktors Friction1 nach Faktorstufen

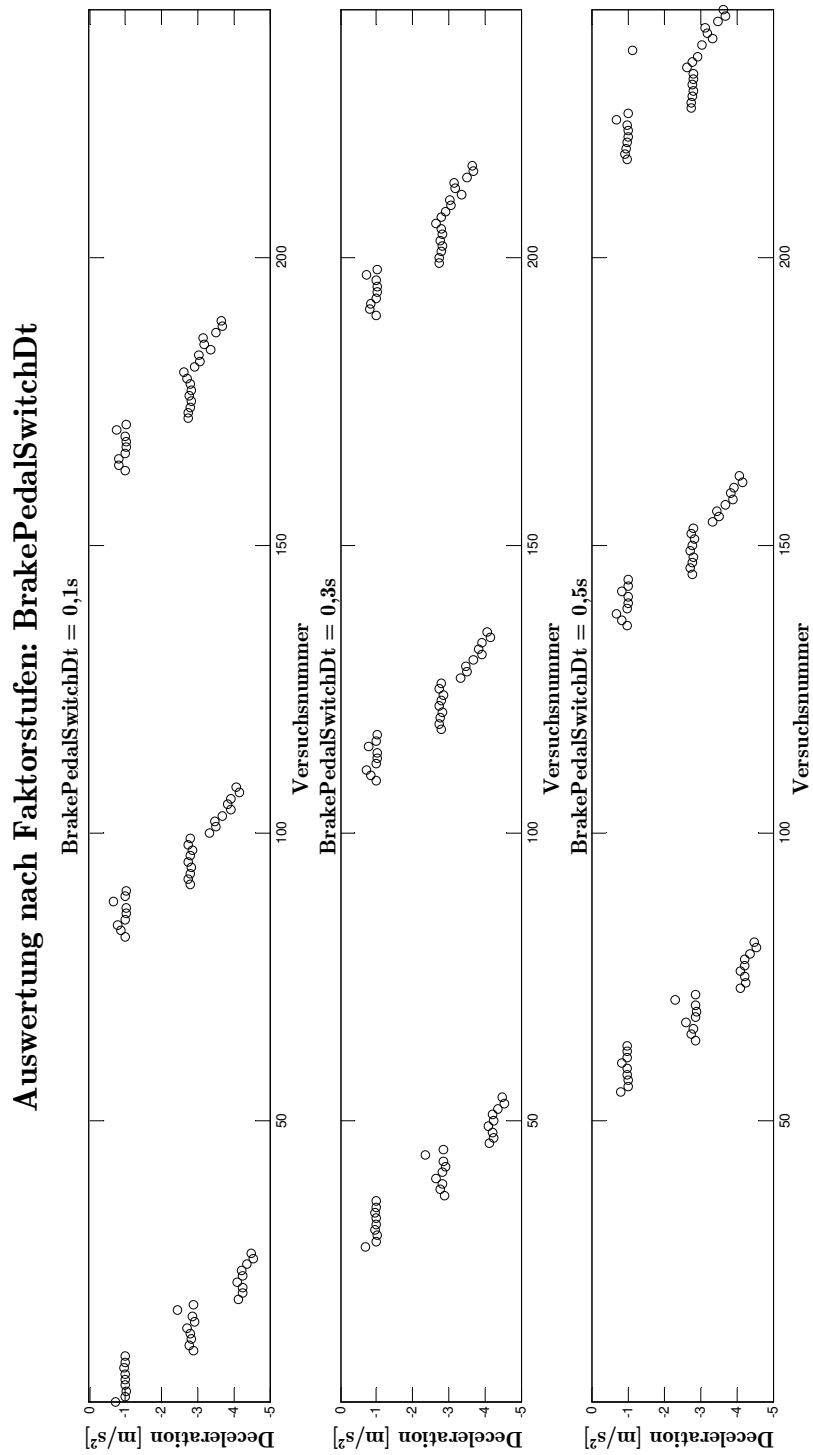


Abbildung A.4. – Einzelauswertung des Faktors BrakePedalSwitchDt nach Faktorstufen

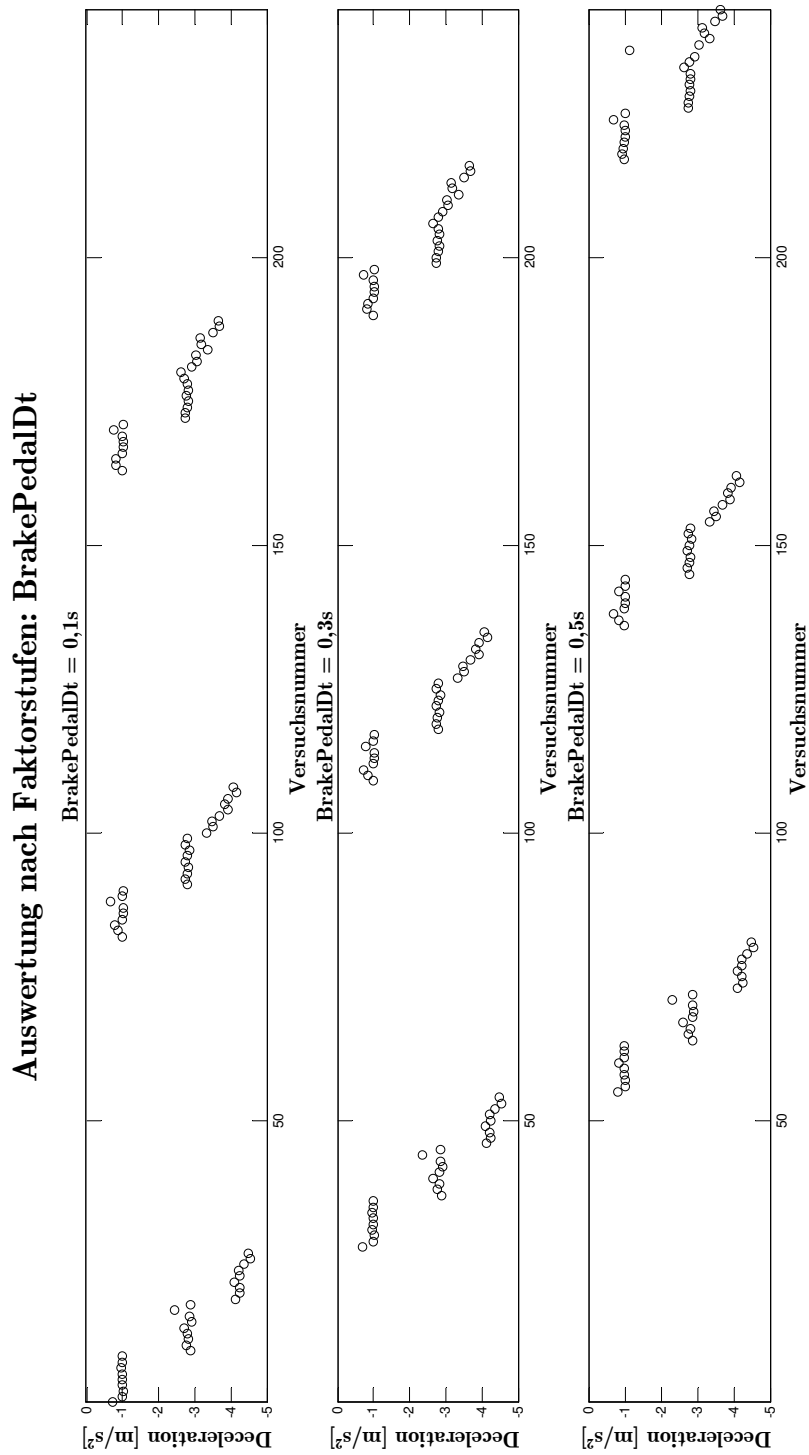


Abbildung A.5. – Einzelauswertung des Faktors BrakePedalDt nach Faktorstufen

B. Anhang - Differential Evolution

B.1. Beispiel 1: Vergleich Ausführung mit und ohne vorhergehende statistische Analyse

Nachfolgend werden die Ergebnisse von Beispiel 1 aus Kapitel 4.4.6 mit und ohne vorherige statistische Analyse verglichen. Die Unterschiede im Versuchsaufbau sind in Tabelle B.1 festgehalten. Der Unterschied liegt vor allem in der Anzahl der Parameter D und der Para-

	Mit statistischer Analyse (Abb. B.1, analog zu Tabelle 4.8)	Ohne statistische Analyse (Abb. B.2)
Parameteranzahl	$D = 3$	$D = 5$
Skalierungsfaktor	$F = 0,5$	$F = 0,5$
Mutationsrate	$Cr = 0,4$	$Cr = 0,4$
Populationsgröße	$N = 30$	$N = 50$
Versuchsgröße	$2 \cdot N = 60$	$2 \cdot N = 100$
Anzahl Generationen	$G_{max} = 50$	$G_{max} = 50$
<i>Parameter Grenzen:</i>		
Brake [-]	0,5 - 1,0	0,2 - 1,0
BrakePedalDt [s]	0,1 - 0,6	0,2 - 1,0
RoadFriction [-]	0,1 - 0,4	0,1 - 1,0
CruisingSpeed [km/h]	konstant 55km/h	50,0 - 100,0
BrakePedalSwitchDt [s]	konstant 0,2s	0,2 - 1,0

Tabelle B.1. – Vergleich des Versuchsaufbaus von Beispiel 1 mit und ohne vorherige statistische Analyse

metergrenzen. Mit vorhergehender statistischer Analyse können zwei Parameter reduziert werden, indem sie auf einen konstanten Wert gesetzt werden. Dies reduziert wiederum auch die Populationsgröße, die mit $N = 10 \cdot D$ direkt von der Anzahl der Parameter abhängt. Auch die Versuchsgröße von $2 \cdot N$ reduziert sich entsprechend. Die Berechnung einer Generation ist mit statistischer Analyse bei einer Versuchsgröße von $2 \cdot N = 60$ Individuen im Gegensatz zur Berechnung einer Generation ohne vorherige Analyse bei einer Versuchsgröße von ($2 \cdot N = 100$) Individuen um 40% geringer als ohne vorhergehende Analyse. Bei einer Simulationsdauer von 3 Sekunden pro Testfall ergibt dies bei 60 Individuen im vorher analysierten Fall eine Zeitdauer von $60 \cdot 3s = 180s = 3min$ pro Generation. Beim nicht analysierten Fall mit 100 Individuen resultiert dies in eine Simulationszeit von $100 \cdot 3s = 300s = 5min$ pro Generation. Bei der Erzeugung von 50 Generationen ergibt sich so ein Zeitunterschied und somit eine Zeitersparnis von $(50 \cdot 5min) - (50 \cdot 3min) = 100min$.

Des Weiteren sind die Parametergrenzen bei vorhergehender statistischer Analyse stärker eingrenzbar, da der Einfluss bzw. der Effekt der Einzel- und Wechselwirkungen bekannt ist. Ein Parameter kann somit bereits voroptimiert werden.

Die Abbildungen B.1 und B.2 zeigen die Ergebnisse mit und ohne vorherige statistische Analyse für Beispiel 1 mit jeweils fünf durchgeführte Versuchen. Eingezeichnet sind dort als Punkte jeweils die Generation, an der das Ergebnis für die Verzögerung das erste Mal den Wert $1,9\text{m/s}^2 \pm 0,0001\text{m/s}^2$ erreicht. In Abbildung B.1 wird dies bei den fünf Versuchen nach $(16 + 14 + 25 + 19 + 13)/5 = 17,4$ Generationen erreicht. Dies entspricht einer Simulationszeit von $17,4 \cdot 3\text{min} = 52,2\text{min}$, um die gewünschte Ergebnislösung zu erreichen. Im Fall der Optimierung ohne vorhergehende Analyse in Abbildung B.2 wird die gewünschte Güte bei den fünf Versuchen nach $(3 + 29 + 28 + 31 + 31)/5 = 24,4$ Generationen erreicht. Dies entspricht einer durchschnittlichen Simulationszeit von $24,4 \cdot 5\text{min} = 122\text{min}$. Es zeigt sich somit, dass die vorhergehende Analyse die Simulationszeit mehr als halbieren kann.

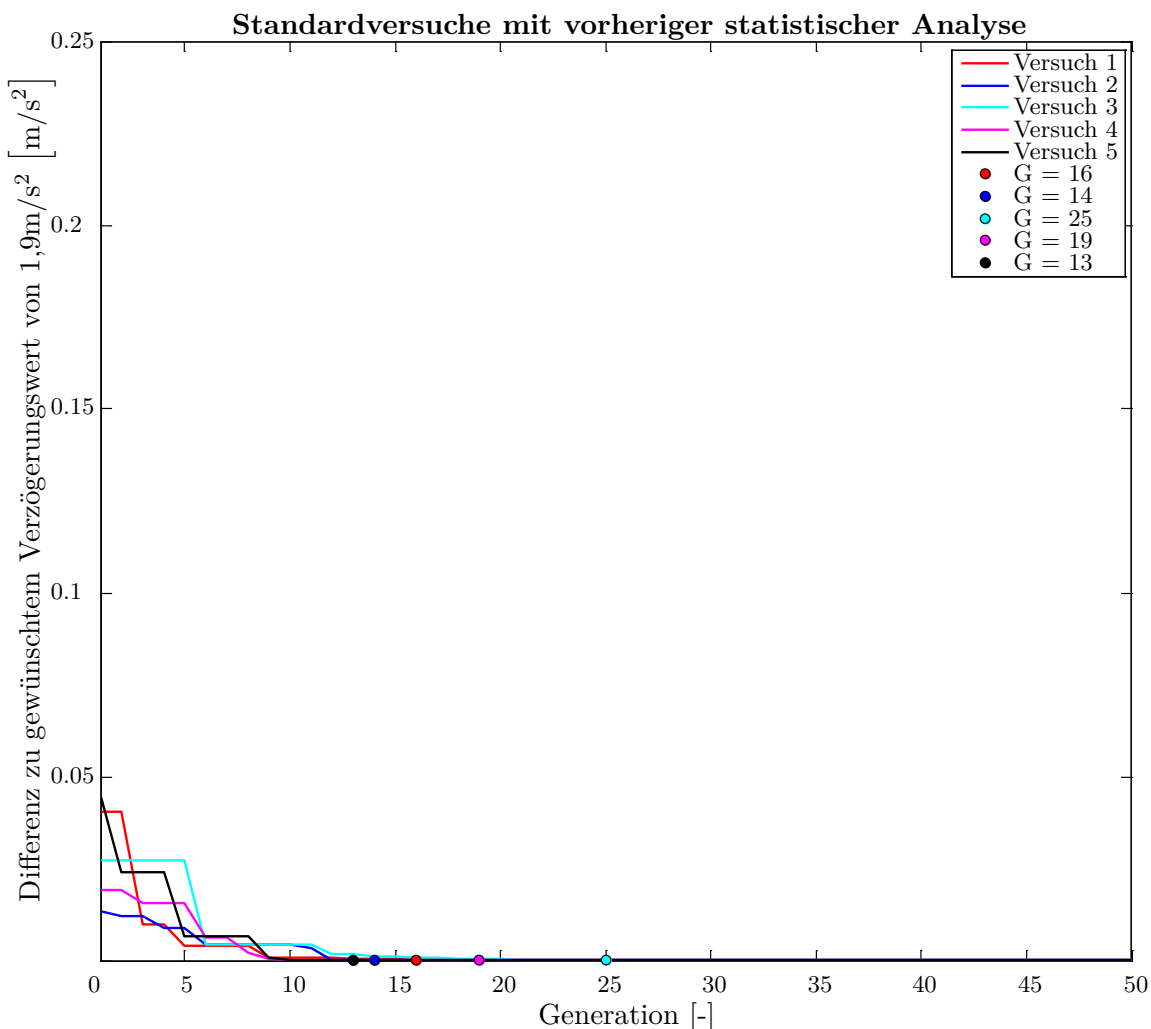


Abbildung B.1. – Fünfmalige Durchführung von Beispiel 1 mit vorheriger statistischer Analyse. Versuch 1 entspricht den Versuchsdaten aus Tabelle 4.9.

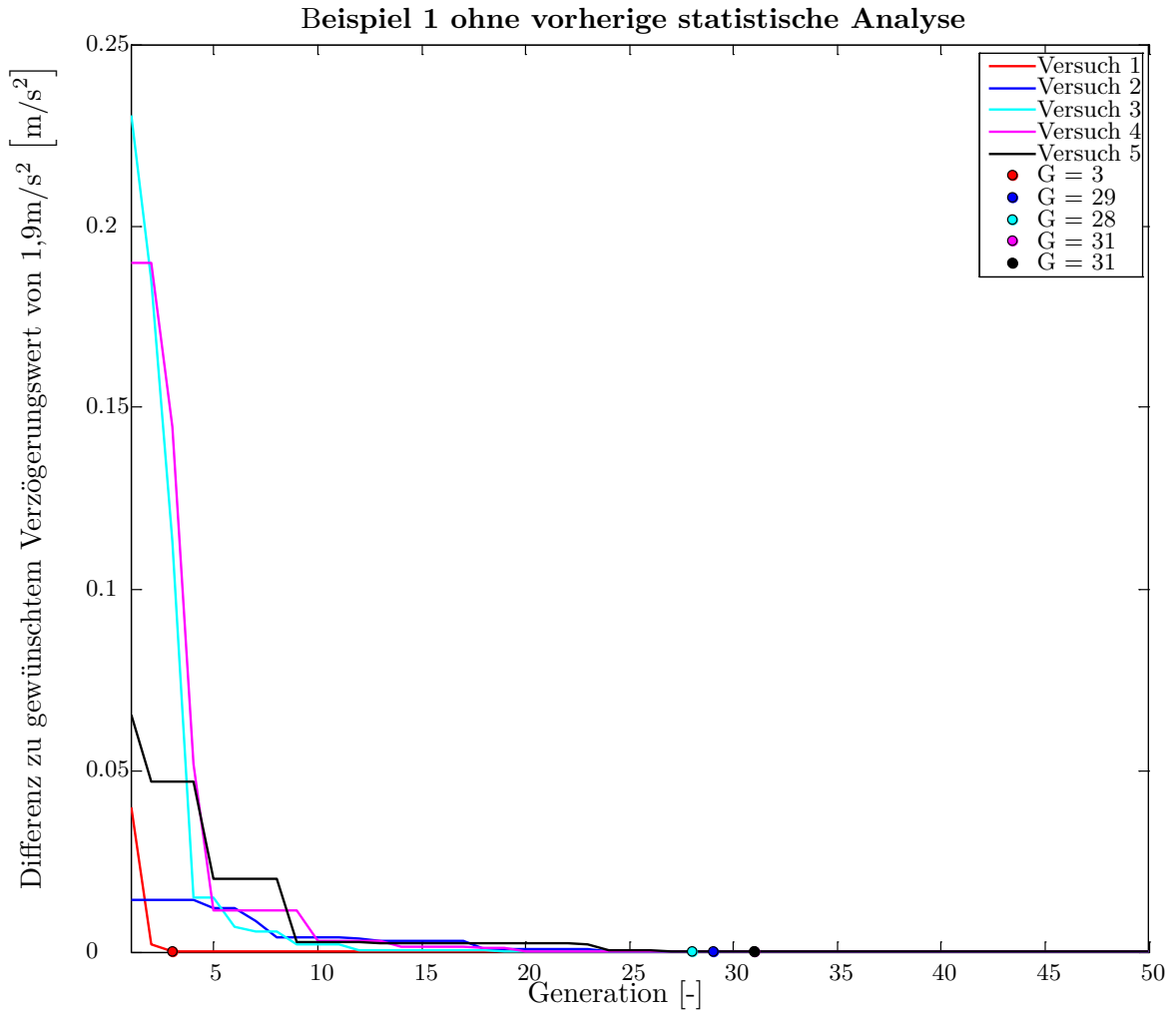


Abbildung B.2. – Fünfmalige Durchführung von Beispiel 1 ohne vorherige statistische Analyse

<i>Mit statistischer Analyse</i>					
Versuch	BrakePedalDt	Brake	RoadFriction	Br.P.SwitchDt	CruisingSpeed
1	0,27525s	0,79464	0,19072	konstant	konstant
2	0,57436s	0,61429	0,1894	konstant	konstant
3	0,14254s	0,85537	0,18977	konstant	konstant
4	0,42578s	0,98714	0,1893	konstant	konstant
5	0,51304s	0,58068	0,18937	konstant	konstant
<i>Ohne statistische Analyse</i>					
Versuch	BrakePedalDt	Brake	RoadFriction	Br.P.SwitchDt	CruisingSpeed
1	0,94969s	0,774027	0,188278	0,654703s	65,3312km/h
2	0,345868s	0,553683	0,189227	0,234479s	58,4717km/h
3	0,744593s	0,631872	0,18949	0,357936s	52,4683km/h
4	0,253678s	0,735511	0,1893	0,298912s	84,9235km/h
5	0,205699s	0,577093	0,186151	0,793537s	79,6657km/h

Tabelle B.2. – Parameterwerte für Beispiel 1 für das erste Individuum, das die Güte von $1,9\text{m/s}^2 \pm 0,0001\text{m/s}^2$ erreicht.

B.2. Beispiel 1: Vergleich Variation der Kontrollparameter Cr und F

Nachfolgend werden die Ergebnisse von Beispiel 1 aus Kapitel 4.4.6 bei Variation der Kontrollparameter der DE Crossoverrate Cr und Skalierungsfaktor für die Mutation F verglichen. Die Kontrollparameter werden jeweils sehr hoch (0,9) und sehr niedrig (0,1) gesetzt und jeweils fünf gleiche Versuche durchgeführt [101]. Der Versuchsaufbau ist ausgenommen der jeweiligen Variation von Cr und F analog zur linken Spalte „Mit statistischer Analyse“ in Tabelle B.1. Die Ergebnisse der jeweils fünf Versuche sind in den Abbildungen B.3 bis B.6 dargestellt. Analog zur Auswertung in Kapitel B.1 wurden jeweils wieder die Generationen als Punkte gekennzeichnet, in denen jeweils das Ergebnis zum ersten Mal einen Gütewert von $1,9\text{m/s}^2 \pm 0,0001\text{m/s}^2$ erreicht hat. Tabelle B.3 zeigt die durchschnittlichen Generationswerte für die Güteerreichung. Es zeigt sich, dass der

Variation	Durchschnittsgeneration Güteerreichung ($\sum_{n=1}^5 \text{Generation_Güteerreichung_Versuch}_n$)/5
$F = 0,9$	$(25 + 24 + 24 + 23 + 22)/5 = 23,6$
$F = 0,1$	$(10 + 23 + 15 + 20 + 14)/5 = 16,4$
$Cr = 0,9$	$(15 + 6 + 8 + 25 + 24)/5 = 15,6$
$Cr = 0,1$	$(27 + 27 + 15 + 26 + 49)/5 = 28,8$

Tabelle B.3. – Durchschnittliche Generation in der der vorgegebene Gütewert von $1,9\text{m/s}^2 \pm 0,0001\text{m/s}^2$ je Variation erreicht wird.

Algorithmus sehr unempfindlich ist, wenn F verkleinert oder Cr vergrößert wird. Es werden ähnliche Ergebnisse wie in Abbildung B.1 erreicht, wo die Werte $F = 0,5$ und $Cr = 0,4$ vorgegeben waren. Eine kleinere Schrittweite bei der Mutation oder ein häufigeres Crossover sind also nicht nachteilig. Wenn F allerdings sehr groß oder Cr sehr klein gesetzt werden, wird der vorgegebene Gütewert erst sehr viel später erreicht. Eine große Schrittweite bei der Mutation oder wenig Crossover führen also dazu, dass der Algorithmus länger nach lokalen Optima sucht [101].

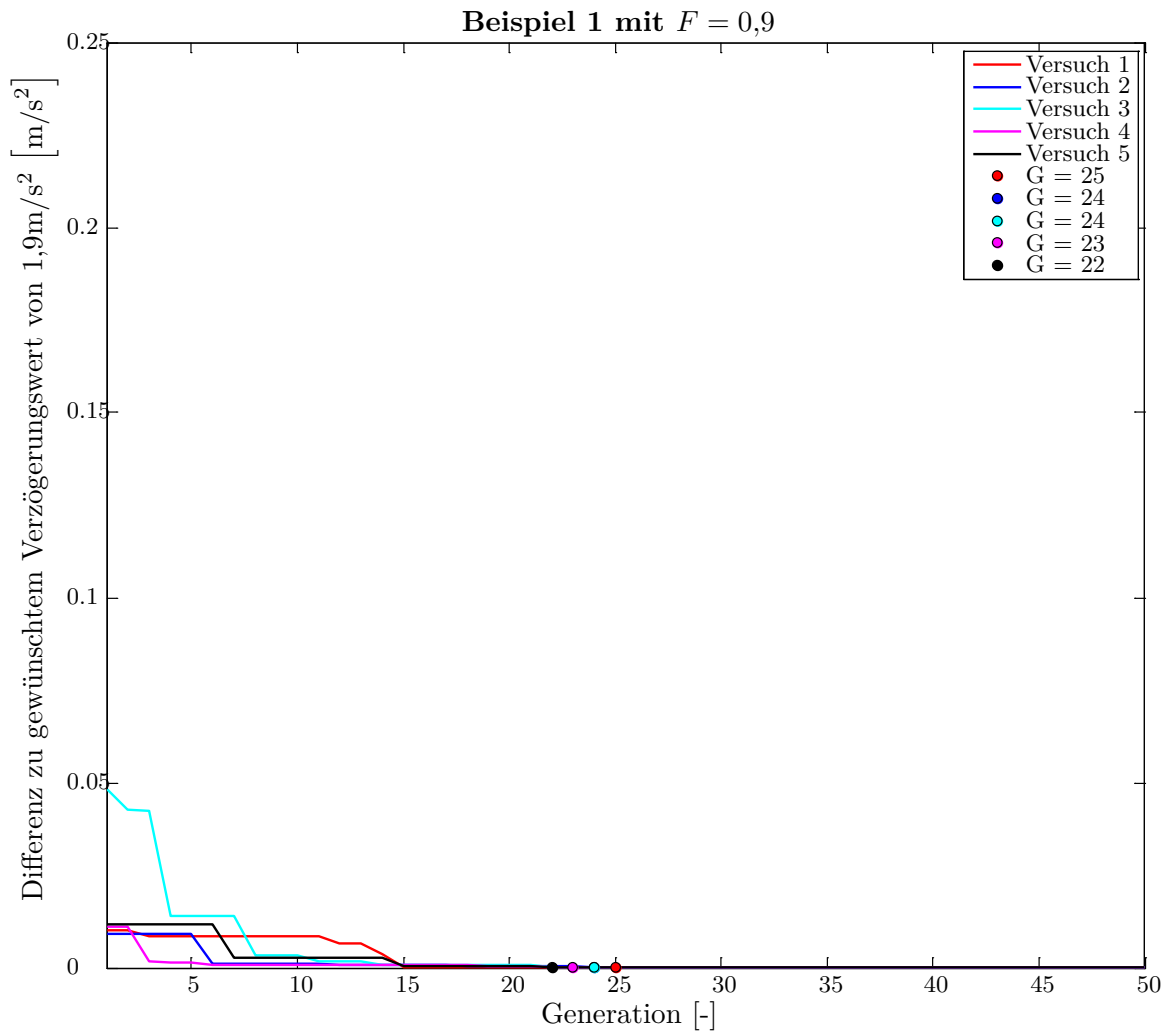


Abbildung B.3. – Fünfmalige Durchführung von Beispiel 1 mit $F = 0,9$

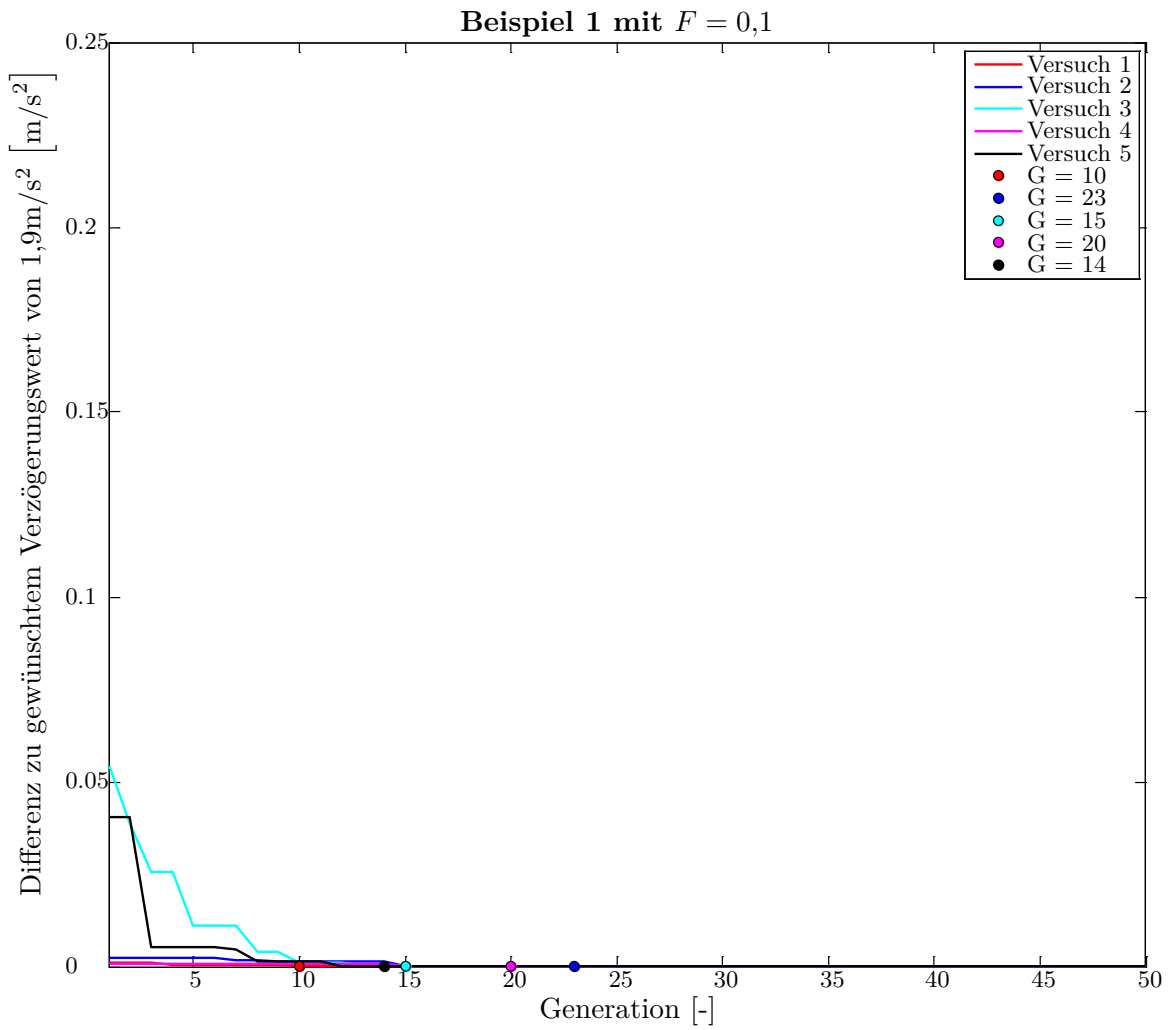


Abbildung B.4. – Fünfmalige Durchführung von Beispiel 1 mit $F = 0,1$

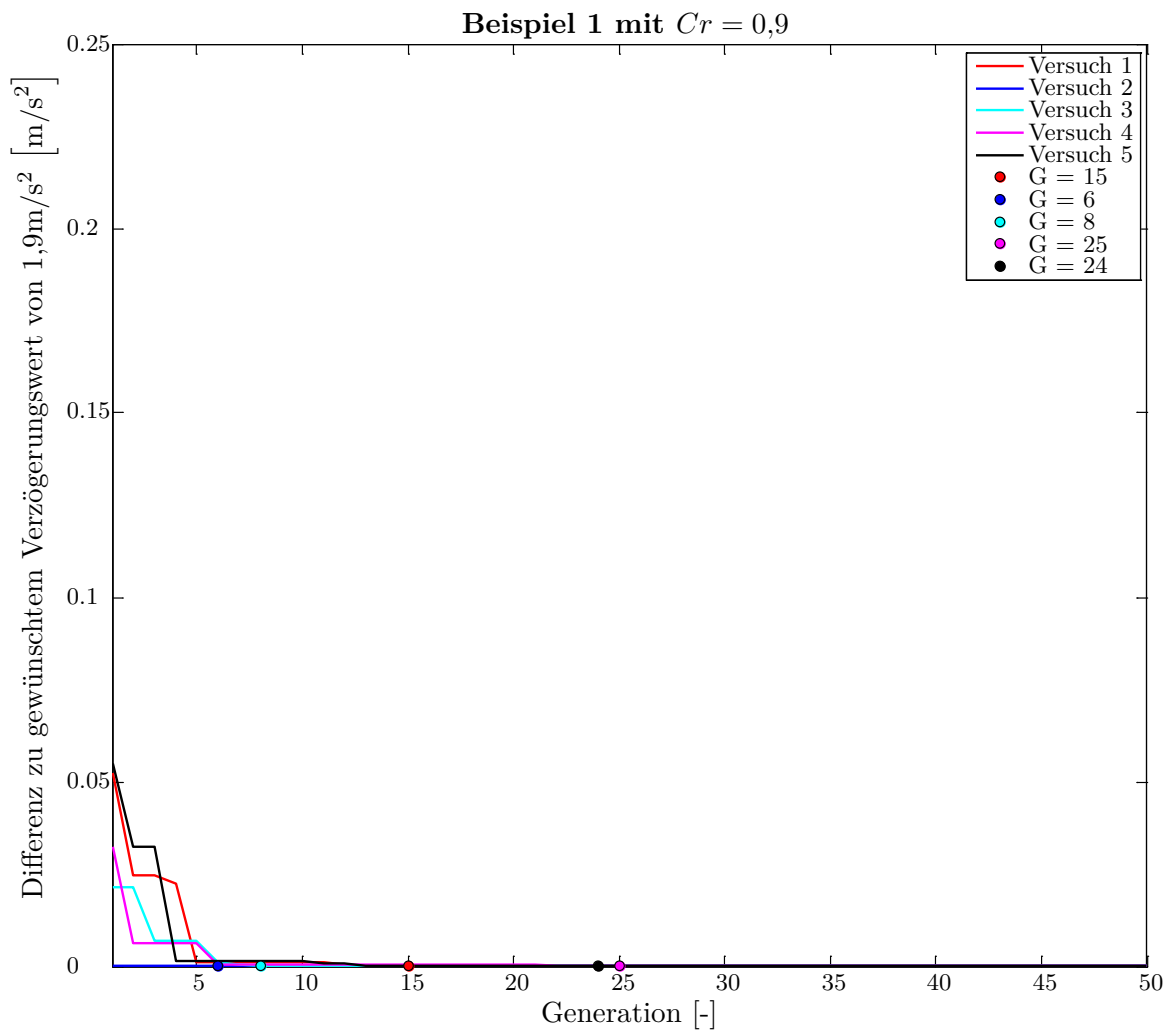


Abbildung B.5. – Fünfmalige Durchführung von Beispiel 1 mit $Cr = 0,9$

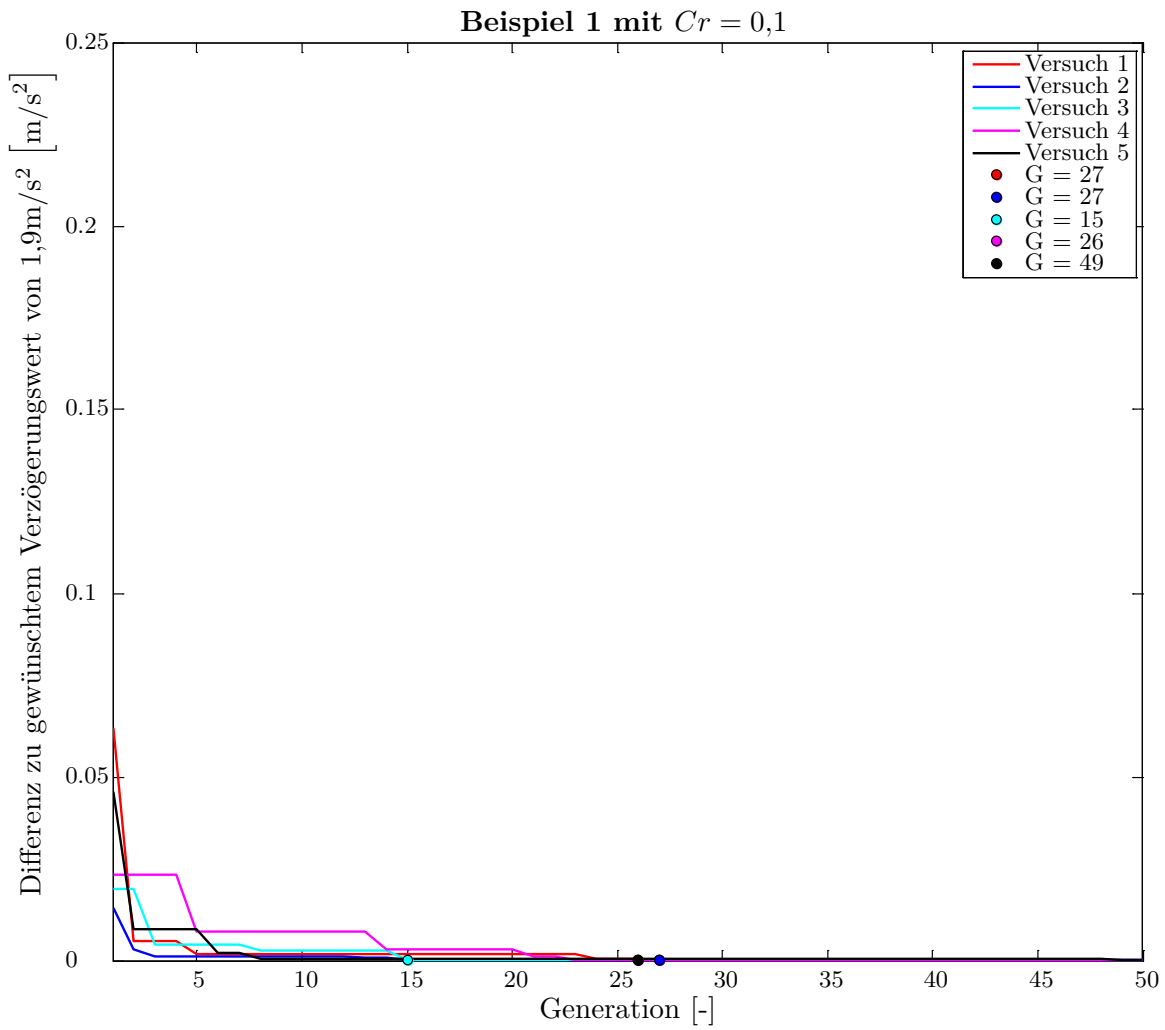


Abbildung B.6. – Fünfmalige Durchführung von Beispiel 1 mit $Cr = 0,1$

$F = 0,9$			
Versuch	BrakePedalDt	Brake	RoadFriction
1	0,21283s	0,78674	0,18935
2	0,5015s	0,65273	0,18945
3	0,47639s	0,82115	0,18938
4	0,34039s	0,81641	0,18939
5	0,48297s	0,7945	0,18934
$F = 0,1$			
Versuch	BrakePedalDt	Brake	RoadFriction
1	0,5273s	0,76979	0,1893
2	0,57519s	0,67206	0,18924
3	0,53696s	0,6962	0,18924
4	0,2219s	0,76873	0,18933
5	0,55017s	0,90066	0,18941
$Cr = 0,9$			
Versuch	BrakePedalDt	Brake	RoadFriction
1	0,35323s	0,84416	0,18941
2	0,12361s	0,59761	0,19049
3	0,38666s	0,8177	0,18938
4	0,5082s	0,65266	0,18944
5	0,24648s	0,81129	0,18939
$Cr = 0,1$			
Versuch	BrakePedalDt	Brake	RoadFriction
1	0,22523s	0,64471	0,18945
2	0,50429s	0,61892	0,1894
3	0,49436s	0,7714	0,18932
4	0,22667s	0,54472	0,18933
5	0,3445s	0,50145	0,18924

Tabelle B.4. – Parameterwerte bei der Variation von Cr und F für Beispiel 1 für das erste Individuum, das die Güte von $1,9\text{m/s}^2 \pm 0,0001\text{m/s}^2$ erreicht [101]t.

B.3. Beispiel 2: Wiederholung des Versuchs

Beispiel 2 aus Kapitel 4.4.7 wird hier zwei Mal mit denselben Einstellungen wie in Tabelle 4.10 wiederholt. Anstatt 100 Generationen wurden jeweils 420 Generationen durchgeführt, um zu zeigen, dass keine weiteren ausschlaggebenden Optimierungen stattfinden. Es zeigt sich beim Vergleich der Ergebnisse in Tabelle B.5, dass die Parameterwerte sich weiterhin optimieren, jedoch der Zielwert für die Schwimmwinkelgeschwindigkeit sich kaum noch verändert. Für eine gründliche Untersuchung sind also langfristige Versuche sinnvoll, für einen relativ guten Zielwert reicht in diesem Fall auch ein Viertel der Generationen (100 statt 420). Des Weiteren zeigt sich im Vergleich von Versuch 1 mit Versuch 2, dass auch verschiedene Parameterwerte zu einem sehr ähnlichen Zielwert führen können.

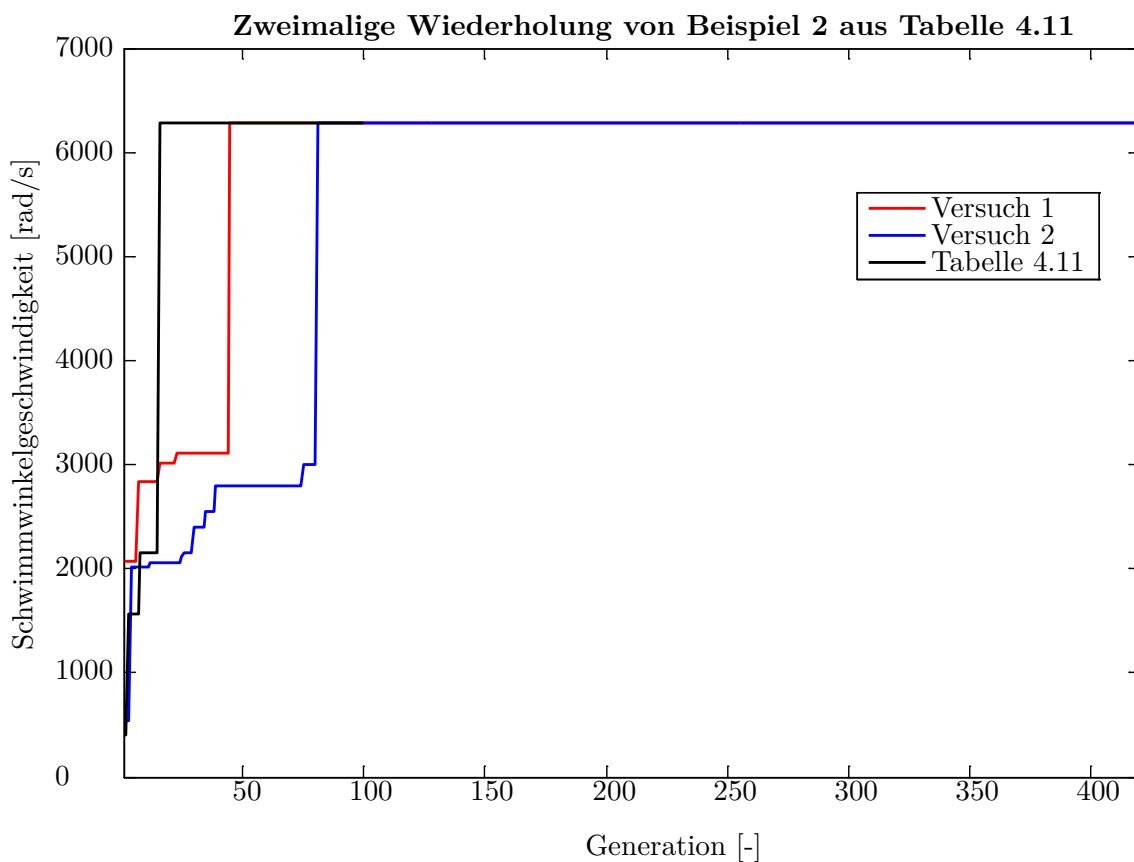


Abbildung B.7. – Zweimalige Wiederholung von Beispiel 2 im Vergleich mit den Ergebnissen aus Tabelle 4.11.

	Versuch 1 $G_{max} = 420$	Versuch 2 $G_{max} = 420$	Tabelle 4.11 $G_{max} = 100$
CruisingSpeed	49.8196km/h	49.503km/h	44,3899km/h
Brake	0,85892	0,884868	0,8447
BrakePedalSwitchDt	0,191737s	0,181894s	0,2138s
BrakePedalDt	0,227239s	0,124798s	0,1558s
RoadFriction1	0,208582	0,395704	0,2089
RoadFriction2	0,360351	0,217648	0,6658
max. Schwimmwinkelgeschwindigkeit	6282,5586rad/s	6282,4595rad/s	6281,8521rad/s

Tabelle B.5. – Parameterwerte des bestangepassten Individuums für zweimalige Wiederholung von Beispiel 2 im Vergleich mit den Ergebnissen aus Tabelle 4.11.

Definitionsverzeichnis

Aktive Sicherheit	8
Passive Sicherheit	8
Fahrerassistenzsystem	8
Steuergerät	10
Sensor	12
Aktor oder Aktuator	13
Funktionale Sicherheit	15
Zusammenhang zwischen Risiko und Schaden	16
ASIL - Automotive Safety Integrity Level	16
System	20
Modul-/ Komponententest	21
Integrationstest	21
Systemtest	21
Akzeptanz-/ Abnahmetest	21
Ziele des Systemtests	21
Vorgehensmodell	22
Verifizierung	24
Validierung	24
Spiralmodell	25
A-Muster	25
B-Muster	25
C-Muster	25
D-Muster	26
Anforderungen	28
Arten von Anforderungen	29
Kriterien für gute Anforderungen	29
Baseline	31
Release	31
Methodik	31
Methode	31
Teststrategie	32
Testspezifikation	35
Metrik	35
Statischer Test	36
Dynamischer Test	36

Black-Box-Test	37
White-Box-Test	38
Grey-Box-Test	39
Model-in-the-Loop, MiL	39
Software-in-the-Loop, SiL	39
Hardware-in-the-Loop, HiL	39
Testsystem	40
Kennzahlen	50
Indikatoren	50
Audit	51
Statistische Versuchsplanung	72
Evolutionäre Algorithmen	88
Optimierung	88
Testmetrik	124

Abbildungsverzeichnis

1.1. Verkehrstote in der EU seit 2001 und Zielsetzung der Reduzierung der tödlich Verunglückten nach [31]	2
1.2. Verkehrstote in Europa pro eine Million Einwohner im Jahr 2011 [30]	3
1.3. Aufbau der Methodik und analog Aufbau der Arbeit	5
2.1. Ablauf eines Unfalls und zugehörige Sicherheitssysteme (nach [28])	9
2.2. Entwicklung der Zunahme der Assistenz hin zu einem vollständig integrierten Sicherheitssystem (nach [74])	10
2.3. Schematischer Aufbau eines Steuergeräts (nach [83] und [109])	11
2.4. Darstellung des Unterschieds zwischen logischer und technischer/physischer Systemarchitektur (nach [114]).	12
2.5. Sensoren im Bereich Fahrzeugsicherheit (nach [91])	13
2.6. Die Systemtestkette (nach [118]).	22
2.7. Das V-Modell (nach [8], [10], [83] und [122])	23
2.8. Das Spiralmodell (nach [97])	25
2.9. Der Safety Lifecycle nach ISO 26262 [57].	26
2.10. Das Referenzphasenmodell für die Systemebene nach ISO 26262 [57].	27
2.11. Benutzergruppen eines Fahrzeugs und die sich ergebenden Anforderungsklassen an das System Fahrzeug (nach [114]).	28
2.12. Einmalige und laufende Tätigkeiten des Requirements-Managements (nach [47])	30
2.13. Verfolgung von Anforderungen von Benutzeranforderungen über die logische hin zur technischen Systemarchitektur (nach [114] und [124])	30
2.14. Der fundamentale Testprozess [120].	32
2.15. Testdokumentation im Systemtest nach IEEE Std 829-2008 [50])	34
2.16. Prinzip des Black-Box-Testens (nach [109])	37
2.17. Prinzip des White-Box-Testens (nach [109])	38
2.18. Prinzip des Grey-Box-Testens (nach [109])	39
2.19. Schematische Darstellung des MiL-, SiL- und HiL-Testens (nach [91])	40
2.20. Prinzipieller Aufbau eines Hardware-in-the-Loop-Testsystems (nach [91])	42
2.21. Veränderung der Entwicklungsfaktoren in der Automobilbranche (nach [73])	43
2.22. Schematische, exemplarische Darstellung der Steuergerätevernetzung in einem kontemporären Automobil (nach [64], [103], [104] und [105])	44
3.1. Teil 1 der in der Arbeit vorgeschlagenen Methodik (siehe auch Abbildung 1.3).	49

3.2. Ableitung von Fehlerschätzung und Testmethoden aus der Projektkomplexität [9].	56
3.3. Im Systemtest aufgetretene Fehler der Projekte.	58
3.4. Gegenüberstellung der Komplexität K und der zugehörigen Fehler.	59
4.1. Teil 2 und 3 der in der Arbeit vorgeschlagenen Methodik (siehe auch Abbildung 1.3).	61
4.2. Durch diese Arbeit vorgeschlagener Ansatz bei der Auswahl der Testmethoden	63
4.3. Schematische Darstellung des Testraums (nach [116], [9])	64
4.4. Closed-loop Drei-Level-Hierarchie von Fahrmanövern nach [7]	65
4.5. Manöverbasierte Einstufung vom Ablauf eines Unfalls inklusive entsprechend auslösender Sicherheitssysteme (nach [88] und [102])	66
4.6. Beispiele für manöver- und eventbasiertes Testen	68
4.7. Prinzip des manöverbasierten Testens im Bereich des Systemtests von Steuergeräten	68
4.8. Aufbau und Prinzip des reversiblen Gurtstraffers (nach [77] und [82])	69
4.9. DoE optimiert drei Teile des Wissensakquiseprozesses nach [40]	74
4.10. Prinzipielle Vorgehensweise des Design of Experiments nach [67]	74
4.11. Verschiedene Arten von Versuchsplänen nach [68]	76
4.12. Durchführung der Testreihe mithilfe des Testmanagers	79
4.13. Gesamtergebnis des ausgeführten Versuchsplans inklusive Maximum	80
4.14. Darstellung der Mittelwerte und der zugehörigen Haupteffekte der Faktoren	83
4.15. Graphische Auswertung des Wechselwirkungseffekts von Friction und BrakePedalDt	84
4.16. Darstellung von Beispielen für Wechselwirkungseffekte zwischen zwei Faktoren nach [85]	85
4.17. Graphische Auswertung der Wechselwirkungseffekte aus Tabelle 4.6	86
4.18. Einzel- und Wechselwirkungseffekte sortiert nach ihrer absoluten Höhe	87
4.19. Darstellung des Prinzips der Evolutionslehre durch Mutation, Rekombination und Selektion	89
4.20. Darstellung des Prinzips der DE durch Mutation, Rekombination und Selektion	90
4.21. Darstellung des Prinzips der DE durch Mutation, Rekombination und Selektion auf Datenebene	91
4.22. Darstellung des DE-Mutationsverfahrens im 2- D -Parameterraum	93
4.23. Darstellung des DE-Crossoververfahrens nach binominaler Methode für einen 7- D -Parameterraum	94
4.24. Ablauf der Kombination von DE und manöverbasiertem Testen	96
4.25. Prinzip der Kombination von DE und manöverbasiertem Testen im Bereich des Systemtests von Steuergeräten	98
4.26. Veränderung der Parameterwerte und Indizes der minimalen Individuen je Generation, Experiment 1	102

4.27. Maximale und minimale sowie durchschnittliche Parameterwerte je Generation, Experiment 1	103
4.28. Einspeisung der optimierten Simulationsparameter in das System zum Test der Funktion reversibler Gurtstraffer.	104
4.29. Gierwinkel ψ , Schwimmwinkel β und des Kurvenverhalten von Fahrzeugen anhand des Linearen Einspurmodells	106
4.30. Veränderung der Parameterwerte und Indizes der minimalen Individuen je Generation, Experiment 2	110
4.31. Maximale und minimale sowie durchschnittliche Parameterwerte je Generation, Experiment 2	111
4.32. Anteil der ASIL-Anforderungen in einem kontemporären Fahrzeugsicherheitsprojekt	114
5.1. Teil 4 der in der Arbeit vorgeschlagenen Methodik (siehe auch Abbildung 1.3).	117
5.2. Schematische Darstellung des Ablaufs des durchgängigen Testens.	120
5.3. Schematische Abbildung eines HiL-Testsystems für das Sicherheitssteuergerät im Fahrzeug.	121
6.1. Teil 5 der in der Arbeit vorgeschlagenen Methodik (siehe auch Abbildung 1.3).	123
6.2. Stand der Anforderungstests der drei ausgewählten Projekte nach zwei Dritteln der Projektlaufzeit.	132
6.3. Ermittlung des Vertrauensgrad anhand der Fehlerüberdeckung aus Komplexitätsbewertung, Expertenaudit Testqualitätsbewertung und Fehlerdichte und Testüberdeckung.	135
7.1. Übersicht über die in der Arbeit vorgeschlagene Methodik (siehe auch Abbildung 1.3).	141
A.1. Einzelauswertung des Faktors CruisingSpeed nach Faktorstufen	146
A.2. Einzelauswertung des Faktors Brake nach Faktorstufen	147
A.3. Einzelauswertung des Faktors Friction nach Faktorstufen	148
A.4. Einzelauswertung des Faktors BrakePedalSwitchDt nach Faktorstufen	149
A.5. Einzelauswertung des Faktors BrakePedalDt nach Faktorstufen	150
B.1. Fünfmalige Durchführung von Beispiel 1 mit vorheriger statistischer Analyse	152
B.2. Fünfmalige Durchführung von Beispiel 1 ohne vorherige statistische Analyse	153
B.3. Fünfmalige Durchführung von Beispiel 1 mit $F = 0,9$	155
B.4. Fünfmalige Durchführung von Beispiel 1 mit $F = 0,1$	156
B.5. Fünfmalige Durchführung von Beispiel 1 mit $Cr = 0,9$	157
B.6. Fünfmalige Durchführung von Beispiel 1 mit $Cr = 0,1$	158
B.7. Zweimalige Wiederholung von Beispiel 2 im Vergleich mit den Ergebnissen aus Tabelle 4.11.	160

Tabellenverzeichnis

2.1. Risikobewertung nach ISO 26262 (nach [37] und [57])	17
2.2. Projektbewertung für fünf abgeschlossene und drei laufende Beispielprojekte	18
2.3. Testsystemmerkmale (nach [109])	41
3.1. Mögliche beeinflussende Faktoren für die Komplexität von Entwicklungsprojekten der Fahrzeugsicherheit (nach [9]). Die Bewertung erfolgt jeweils in fünf Stufen, wobei höhere Zahlen nahe fünf für kritischere Bewertung stehen.	54
3.2. Paarweiser Vergleich und somit Gewichtung der Faktoren für die Projektkomplexität.	55
3.3. Projektbewertung für fünf abgeschlossene und drei laufende Beispielprojekte	57
3.4. Vorhersage der erwarteten Fehler im Systemtest für drei laufende Projekte.	58
4.1. Vergleich reversibler und pyrotechnischer Gurtstraffer	69
4.2. Mögliche Auslösekriterien für den reversiblen Gurtstraffer bei der Gefahrenbremsung (siehe [130], [101]).	70
4.3. Aufbau Beispiel manöverbasierter Testfall	71
4.4. Vollständiger faktorieller 3^5 -Versuchsplan	78
4.5. Mittelwerte und Ergebnisse der Haupteffekte für die einzelnen Faktorstufen	82
4.6. Ergebnisse der Wechselwirkungseffekte	82
4.7. Pseudo-Code zur Darstellung des DE-Algorithmus (nach [20], [101])	95
4.8. Aufbau Experiment 1	99
4.9. Ergebnisse des Experiments Bremsen auf Eis nach 100 Generationen [101] .	101
4.10. Aufbau Experiment 2	107
4.11. Ergebnisse der μ -Split-Bremsung nach 100 Generationen	108
6.1. Mögliche beeinflussende Faktoren für die Testqualitätsbewertung von Entwicklungsprojekten der Fahrzeugsicherheit (nach [9]). Die Bewertung erfolgt jeweils in fünf Stufen, wobei höhere Zahlen nahe fünf für bessere Bewertung stehen.	127
6.2. Paarweiser Vergleich und somit Gewichtung der Faktoren für die Testqualität.	128
6.3. Testqualitätsaudits für drei aktuell laufende Projekte der integralen Fahrzeugsicherheit im Abstand von 14 Monaten.	130
6.4. Aktueller Stand der Anforderungsüberdeckung für die drei ausgewählten Projekte nach circa zwei Dritteln der Projektlaufzeit.	133

6.5. Stand der Fehlerdichte für die drei ausgewählten Projekte nach circa zwei Dritteln der Projektlaufzeit.	133
6.6. Stand der Fehlerüberdeckung für die drei ausgewählten Projekte nach circa zwei Dritteln der Projektlaufzeit.	134
6.7. Vorschlag zur Ermittlung der Status der Einzelmaße.	135
6.8. Der Vertrauensgrad für die drei ausgewählten laufenden Projekte nach circa zwei Dritteln der Projektlaufzeit.	136
7.1. Beitrag der Arbeit im Vergleich zum Stand der Technik.	142
B.1. Vergleich des Versuchsaufbaus von Beispiel 1 mit und ohne vorherige statische Analyse	151
B.2. Parameterwerte für Beispiel 1 für das erste Individuum, das die Güte von $1,9\text{m/s}^2 \pm 0,0001\text{m/s}^2$ erreicht.	153
B.3. Durchschnittliche Generation in der der vorgegebene Gütewert von $1,9\text{m/s}^2 \pm 0,0001\text{m/s}^2$ je Variation erreicht wird.	154
B.4. Parameterwerte bei der Variation von C_r und F für Beispiel 1 für das erste Individuum, das die Güte von $1,9\text{m/s}^2 \pm 0,0001\text{m/s}^2$ erreicht [101]t.	159
B.5. Parameterwerte des bestangepassten Individuums für zweimalige Wiederholung von Beispiel 2 im Vergleich mit den Ergebnissen aus Tabelle 4.11.	161

Literaturverzeichnis

- [1] ABRAN, A., J.W. MOORE, P. BOURQUE und R. DUPUIS: *Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society, Los Alamitos, Kalifornien, USA, 2004.
- [2] ANONYM: *Methodik und Aussagefähigkeit des Kennzahlen-Controllings für die Geschäftsführung*. GRIN Verlag GmbH, Bachelorarbeit, Hochschule Niederrhein in Mönchengladbach, München, 2011.
- [3] ANTONY, J.: *Design of Experiments for Engineers and Scientists*. Butterworth-Heinemann, Oxford, United Kingdom, 2010.
- [4] AUTOSAR-KONSORTIUM: *Automotive Open System Architecture*. <http://www.autosar.org/>, 2013.
- [5] BALZERT, H.: *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering*. Spektrum Akademischer Verlag, Heidelberg, 3. Auflage, 2009.
- [6] BENZ, S.: *Eine Entwicklungsmethodik für sicherheitsrelevante Elektroniksysteme im Automobil*. Universität Karlsruhe (TH), Dissertation, Karlsruhe, 2004.
- [7] BERGMEIER, U.: *Methode zur kontaktanalogen Visualisierung von Fahrerassistenzinformationen unter automotive-tauglichen Gesichtspunkten*. Produkt- und Produktions-Ergonomie: Aufgabe für Entwickler und Planer; Bericht zum 54. Kongress der Gesellschaft für Arbeitswissenschaft, Seiten 125-128, TU München, 2008.
- [8] BERTRAM, T., P. TORRE FLORES, J. SCHIRMER, J. PETERSEN, A. LAPP, D. KRAFT und W. HERMSEN: *Software-Entwicklung für vernetzte Steuergeräte - Von der Cartronic-Domänenstruktur zum Steuergerätecode*. ATZ/MTZ/Automotive Engineering Partners-Sonderausgabe Automotive Electronics, Seiten 32-41, 2002.
- [9] BIANK, S.: *Der Vertrauensgrad als Testmetrik im Systemtest vorausschauender Fahrerassistenzsysteme*. Bachelorarbeit, Technische Hochschule Ingolstadt, 2014 (Betreute Bachelorarbeit, nicht öffentliche Quelle).
- [10] BOEHM, B.W.: *Verifying and Validating Software Requirements and Design Specifications*. Software, IEEE, Vol.1, Issue 1, Seiten 75-88, 1984.
- [11] BORTZ, J. und N. DÖRING: *Forschungsmethoden und Evaluation*. Springer Verlag, Berlin, 4. Auflage, 2006.

- [12] BRAUN, J., J. KRETTEK, F. HOFFMANN, T. BERTRAM, H. LAUF und G. SCHOPPEL: *Struktur- und Parameteridentifikation mit Evolutionären Algorithmen*. at - Automatisierungstechnik, 59 (2011) 6, Seiten 340-353, 2011.
- [13] CAMPBELL, N.A. und J.B. REECE: *Biologie*. Addison-Wesley Verlag, München, 8. Auflage, 2009.
- [14] CMMI INSTITUTE: *Capability Maturity Model Integration*. 2010.
- [15] COBB, G.W.: *Introduction to Design and Analysis of Experiments*. Springer-Verlag, New York, USA, 1998.
- [16] COPPIT, D., J. YANG, S. KHURSHID, W. LE und K. SULLIVAN: *Software Assurance By Bounded Exhaustive Testing*. IEEE Transactions on Software Engineering, Volume 31, Issue 4, Seiten 328-339, 2005.
- [17] DA SILVA AZEVEDO, L., D. PARKER, M. WALKER, Y. PAPADOPOULOS und R. ESTEVES ARAUJO: *Assisted Assignment of Automotive Safety Requirements*. IEEE Software, Volume 31, Number 1, Seiten 62-68, 2014.
- [18] DARWIN, C.: *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. John Murray, London, United Kingdom, 1. Auflage, 1859.
- [19] DARWIN, C. und A.R. WALLACE: *On the Tendency of Species to form Varieties, and on the Perpetuation of Varieties and Species by Natural Means of Selection*. Journal of the Proceedings of the Linnean Society of London, Zoology 3, London, United Kingdom, 1858.
- [20] DAS, S. und P.N. SUGANTHAN: *Differential Evolution: A Survey of the State-of-the-Art*. IEEE Transactions on Evolutionary Computation, Seiten 4-31, Vol. 15, No. 1, 2011.
- [21] DIJKSTRA, E.W.: *The Humble Programmer*. Communications of the ACM, Vol. 15, Issue 10, Seiten 859–866, 1972.
- [22] DUDEN: *Das große Fremdwörterbuch*. Bibliographisches Institut, Mannheim, 4. Auflage, 2007.
- [23] DUDEN: *Deutsches Universalwörterbuch*. Bibliographisches Institut, Mannheim, 7. Auflage, 2011.
- [24] DÜSER, T.: *X-in-the-Loop – ein durchgängiges Validierungsframework für die Fahrzeugentwicklung am Beispiel von Antriebsstrangfunktionen und Fahrerassistenzsystemen*. Forschungsberichte, Band 47, IPEK - Institut für Produktentwicklung, Karlsruher Institut für Technologie (KIT), Dissertation, Karlsruhe, 2010.

-
- [25] ECKSTEIN, L.: *Aktive Fahrzeugsicherheit und Fahrerassistenz*. fka - Forschungsgesellschaft Kraftfahrwesen mbH Aachen, Aachen, 2011.
- [26] ELBAUM, S., A.G. MALISHEVSKY und G. ROTHERMEL: *Incorporating varying test costs and fault severities into test case prioritization*. Proceedings of the 23rd International Conference on Software Engineering, ICSE 2001, Seiten 329-338, Toronto, Kanada, 2001.
- [27] ELBAUM, S., A.G. MALISHEVSKY und G. ROTHERMEL: *Test Case Prioritization: A Family of Empirical Studies*. IEEE Transactions on Software Engineering, Volume 28, Issue 2, Seiten 159-182, 2002.
- [28] ERTLMEIER, R.: *Modellbasierte und fahrdynamikunterstützte Überschlagserkennung*. Otto-von-Guericke-Universität Magdeburg, Dissertation, Magdeburg, 2013.
- [29] EUROPEAN COMMISSION: *EU transport in figures*. Statistical Pocketbook 2013, Brüssel, Belgien, 2013.
- [30] EUROPEAN COMMISSION: *Road Deaths per Million Inhabitants 2011*. CARE (European Road Accident Database) und EUROSTAT, Brüssel, Belgien, 2013.
- [31] EUROPEAN COMMISSION: *Road fatalities in the EU since 2001*. CARE (European Road Accident Database), Brüssel, Belgien, 2014.
- [32] FISCHER, W. und W. DANGELMAIER: *Produkt- und Anlagenoptimierung*. Springer-Verlag, Berlin, 2000.
- [33] FRANZ, K.: *Handbuch zum Testen von Web-Applikationen*. Springer-Verlag, Berlin, 2007.
- [34] GÄMPERLE, R., S.D. MÜLLER und P. KOUMOUTSAKOS: *A Parameter Study for Differential Evolution*. 3rd WSEAS International Conference on Evolutionary Computation (EC '02), Seiten 4841-4846, Interlaken, Schweiz, 2002.
- [35] GARCIA, F., A. DE LA ESCALERA und J.M. ARMINGOL: *Enhanced obstacle detection based on Data Fusion for ADAS applications*. 16th International IEEE Conference on Intelligent Transportation Systems - (ITSC), Seiten 1370-1375, Den Haag, Niederlande, 2013.
- [36] GAVRIILIDIS, A., T. SCHWERDTFEGGER, J. VELTEN, S. SCHAULAND, L. HOHMANN, A. HASELHOFF, F. BOSCHEN und A. KUMMERT: *Multisensor data fusion for advanced driver assistance systems - the Active Safety Car project*. 7th International Workshop on Multidimensional (nD) Systems (nDs), Seiten 1-5, Poitiers, Frankreich, 2011.
- [37] GEBHARDT, V., G.M. RIEGER, J. MOTTOK und C. GIESSELBACH: *Funktionale Sicherheit nach ISO 26262*. dpunkt.verlag GmbH, Heidelberg, 2013.

- [38] GIETL, G. und W. LOBINGER: *Qualitätsaudit*. Carl Hanser Verlag, München, 3. Auflage, 2014.
- [39] GLADEN, W.: *Performance Measurement*. Springer Fachmedien, Wiesbaden, 6. Auflage, 2014.
- [40] GOUPY, J. und L. CREIGHTON: *Introduction to Design of Experiments*. SAS Institute Inc., Cary, North Carolina, USA, 2. Auflage, 2008.
- [41] GRAHAM, D.: *Requirements and Testing: Seven Missing-Link Myths*. IEEE Software, Volume 19, Number 5, Seiten 15-17, 2002.
- [42] GUNDLACH, C.: *Entwicklung eines ganzheitlichen Vorgehensmodells zur problemorientierten Anwendung der statistischen Versuchsplanung*. Kassel University Press, Dissertation, Kassel, 2004.
- [43] HALANG, W.A.: *Funktionale Sicherheit - Echtzeit 2013*. Springer Vieweg, Berlin, 2013.
- [44] HARTMANN, N.: *Automation des Tests eingebetteter Systeme am Beispiel der Kraftfahrzeugelektronik*. Universität Karlsruhe Dissertation, Karlsruhe, 2001.
- [45] HEISSING, B., M. ERSOY und S. GIES: *Fahrwerkhandbuch*. Springer Vieweg, Berlin, 4. Auflage, 2013.
- [46] HOFFMANN, D.W.: *Software-Qualität*. Springer Vieweg, Berlin, 2. Auflage, 2013.
- [47] HRUSCHKA, P.: *Business Analysis und Requirements Engineering*. Carl Hanser Verlag, München, 2014.
- [48] IEEE STANDARDS BOARD: *IEEE Standard Glossary of Software Engineering Terminology*. New York, USA, 1990.
- [49] IEEE STANDARDS BOARD: *IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications*. IEEE Press, 1998.
- [50] IEEE STANDARDS BOARD: *IEEE Std 829-2008: IEEE Standard for Software and System Test Documentation*. IEEE Press, 2008.
- [51] INTERNATIONAL ELECTROTECHNICAL COMMISSION: *IEC 61508: Functional Safety: safety related systems, parts 1 to 7*. 1998.
- [52] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *EN ISO 9000: Grundlagen und Begriffe*. 2005.
- [53] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *ISO/IEC 15504: SPICE (Software Process Improvement and Capability Determination)*. 2006.
- [54] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *EN ISO 9001: Qualitätsmanagementsysteme – Anforderungen (ISO 9001:2008)*. 2008.

- [55] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *ISO/IEC 12207: Systems and software engineering - Software life cycle processes*. 2008.
- [56] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *ISO/TS 16949: Quality management systems*. 2009.
- [57] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *ISO 26262: Road vehicles - Functional safety*. 2011.
- [58] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *ISO/IEC 15504, SPICE (Software Process Improvement and Capability Determination)*. 2012.
- [59] JIANG, L., C. BAI-GEN, W. YUN-PENG und W. JIAN: *Location-based Cooperative Vehicle Collision Avoidance for Unsignalized Intersections: A Multi-sensor Integration Approach*. International Conference on Connected Vehicles and Expo (ICCVE), Seiten 246-251, Beijing, China, 2012.
- [60] KAUER, M., M. SCHREIBER und R. BRUDER: *How to conduct a car? A design example for maneuver based driver-vehicle interaction*. 2010 IEEE Intelligent Vehicles Symposium, Seiten 1214-1221, University of California, San Diego, Kalifornien, USA, 2010.
- [61] KBA (KRAFTFAHRT-BUNDESAMT): *Fahrzeuguntersuchungen (FU), Hauptuntersuchungen und Einzelabnahmen nach Überwachungsinstitutionen, Jahr 2012*. Flensburg, 2013.
- [62] KBA (KRAFTFAHRT-BUNDESAMT): *Anzahl der gemeldeten Pkw in Deutschland in den Jahren 1955 bis 2014*. Statista 2014, Flensburg, 2014.
- [63] KBA (KRAFTFAHRT-BUNDESAMT): *Verteilung des Pkw-Bestandes in Deutschland nach Alter der Fahrzeuge (Stand 1. Januar 2013)*. Statista 2014, Flensburg, 2014.
- [64] KIESEWETTER, T.: *Der VW Touareg - Karosserielektronik und Infotainment*. ATZ extra, 2010-02, Seiten 40-47, 2010.
- [65] KILLICH, S. und H. LUCZAK: *Unternehmenskooperation für kleine und mittelständische Unternehmen*. Springer Verlag, Berlin, 2003.
- [66] KLEIN, B.: *Versuchsplanung - DoE*. Oldenbourg Verlag, München, 3. Auflage, 2011.
- [67] KLEPPMANN, W.: *Taschenbuch Versuchsplanung*. Carl Hanser Verlag, München, 7. Auflage, 2011.
- [68] KLEPPMANN, W.: *Versuchsplanung*. Carl Hanser Verlag, München, 8. Auflage, 2013.
- [69] KLOEDEN, H., N. DAMAK, R.H. RASSHOFER und E.M. BIEBL: *Sensor resource management with cooperative sensors for preventive vehicle safety applications*. Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF), Seiten 1-5, Bonn, 2013.

- [70] KOCK, H.O.: *Gegenwärtiger Stand und Trends bei Gurtsystemen*. ATZ Automobiltechnische Zeitschrift 101 (1999) 10, Seiten 780-785, 1999.
- [71] KOHRT, J.P., M. KELLER, H. SCHÜTTE und J. RICHTERT: *Automatisierter Test und Evaluierung von Steuergeräten mit Hilfe von Hardware-in-the-Loop-Simulation*. 8. Aachener Kolloquium Fahrzeug- und Motorentechnik, Seiten 569-588, Aachen, 1999.
- [72] KOOMEN, T., L. VAN DER AALST, B. BROEKMAN und M. VROON: *TMap Next. Ein praktischer Leitfaden für ergebnisorientiertes Softwaretesten*. dpunkt.verlag GmbH, Heidelberg, 2008.
- [73] KRAMER, F.: *Passive Sicherheit von Kraftfahrzeugen*. Vieweg + Teubner Verlag, Wiesbaden, 3. Auflage, 2009.
- [74] KRAMER, F.: *Integrale Sicherheit von Kraftfahrzeugen*. Springer Vieweg, Wiesbaden, 4. Auflage, 2013.
- [75] LIGGESMEYER, P.: *Software-Qualität*. Spektrum Akademischer Verlag, Heidelberg, 2. Auflage, 2009.
- [76] LINDLAR, F.: *Modellbasierter evolutionärer Funktionstest*. Technische Universität Berlin, Dissertation, Berlin, 2012.
- [77] LUCHT, A., G.H. WITTENBERG, F. MATZEN, E. ZIEL, P. GROSS und Y. PATEL: *Sicherheitsgurtaufroller mit einer Vorstraffeinrichtung*. Patent WO 03/099619 A2, 2003.
- [78] LÖW, P., R. PABST und E. PETRY: *Funktionale Sicherheit in der Praxis*. dpunkt.verlag GmbH, Heidelberg, 2010.
- [79] MITSCHKE, M. und H. WALLENTOWITZ: *Dynamik der Kraftfahrzeuge*. Springer-Verlag, Berlin, 4. Auflage, 2004.
- [80] MITSCHKE, M. und H. WALLENTOWITZ: *Dynamik der Kraftfahrzeuge*. Springer Vieweg, Wiesbaden, 5. Auflage, 2014.
- [81] MONTGOMERY, D.C.: *Design and Analysis of Experiments*. John Wiley & Sons, Singapur, 8. Auflage, 2013.
- [82] MÜHLFELD, F.: *Fahrstiladaptive Auslegung integraler Sicherheitssysteme am Beispiel des reversiblen Gurtstraffers*. Otto-von-Guericke-Universität Magdeburg, Dissertation, Magdeburg, 2013.
- [83] NÖRENBERG, R.: *Effizienter Regressionstest von E/E-Systemen nach ISO 26262*. KIT Scientific Publishing, Karlsruhe, 2012.

-
- [84] NÖRENBERG, R., R. REISSING, J. WEBER und K.D. MÜLLER-GLASER: *ISO 26262 Conformant Verification Plan*. Proceedings of 8th Workshop Automotive Software Engineering, Seiten 515-520, Leipzig, 2010.
- [85] PETERSEN, H.: *Grundlagen der statistischen Versuchsplanung*. ecomed verlagsgesellschaft, Landsberg am Lech, 1991.
- [86] PETSCHENIK, N.H.: *Practical Priorities in System Testing*. IEEE Software, Volume 2, Number 5, Seiten 18-23, 1985.
- [87] PRICE, K.V., R.M. STORN und J.A. LAMPINEN: *Differential Evolution - A Practical Approach to Global Optimization*. Springer-Verlag, Berlin, 1. Auflage, 2005.
- [88] RAITH, A., K. SATTLER, R. ERTLMEIER und T. BRANDMEIER: *Networking and Integration of Active and Passive Safety Systems*. 2011 Proceedings of the Ninth Workshop on Intelligent Solutions in Embedded Systems WISES 2011, IEEE Conference Proceedings, Hochschule Regensburg, Seiten 75-80, Regensburg, 2011.
- [89] RAMASAMY, K. und S.A. MARY: *Incorporating varying requirement priorities and costs in test case prioritization for new and regression testing*. International Conference on Computing, Communication and Networking, ICCCN 2008, Seiten 1-9, St. Thomas, United States Virgin Islands, 2008.
- [90] REIF, K.: *Fahrstabilisierungssysteme und Fahrerassistenzsysteme*. Vieweg + Teubner Verlag, Wiesbaden, 2010.
- [91] REIF, K.: *Bosch Autoelektrik und Autoelektronik*. Vieweg + Teubner Verlag, Wiesbaden, 6. Auflage, 2011.
- [92] REIF, K.: *Automobilelektronik*. Vieweg + Teubner Verlag, Wiesbaden, 4. Auflage, 2012.
- [93] REIF, K.: *Sensoren im Kraftfahrzeug*. Vieweg + Teubner Verlag, Wiesbaden, 2. Auflage, 2012.
- [94] RIETH, P.E.: *Derzeitiger Stand der intelligenten Fahrerassistenz*. Intelligenz im Automobil, FSD-Symposium 2010, Radeberg, 2010.
- [95] ROAD SAFETY FUND: *Decade of Action for Road Safety 2011-2020*. Guide, London, United Kingdom, 2011.
- [96] ROSENBLUM, D.S. und E.J. WEYUKER: *Using coverage information to predict the cost-effectiveness of regression testing strategies*. IEEE Transactions on Software Engineering, Vol. 23, Issue 3, Seiten 146-156, 1997.
- [97] ROSS, H.L.: *Funktionale Sicherheit im Automobil*. Carl Hanser Verlag, München, 2014.

- [98] ROTHERMEL, G., R.H. UNTCH, C. CHU und M.J. HARROLD: *Test Case Prioritization: An Empirical Study*. IEEE International Conference on Software Maintenance, ICSM '99, Seiten 179-188, Oxford, United Kingdom, 1999.
- [99] ROYCE, W.: *Managing the Development of Large Software Systems*. Institute of Electrical and Electronics Engineers, Proceedings of IEEE WESCON, Vol. 26, Seiten 328-338, 1970.
- [100] SADAVA, D., G.H. ORIAN, H.C. HELLER, D.M. HILLIS und M.R. BERENBAUM: *Purves, Biologie*. Spektrum Akademischer Verlag, Heidelberg, 9. Auflage, 2012.
- [101] SATTLER, K., C. DIEDRICH und T. BRANDMEIER: *Manöverbasiertes Testen in Kombination mit evolutionären Algorithmen*. at - Automatisierungstechnik, Band 63, Heft 6, Seiten 450-464, ISSN (Online) 2196-677X, ISSN (Print) 0178-2312, DOI: 10.1515/auto-2014-1156, 2015.
- [102] SATTLER, K., A. RAITH, D. SADOU und C. SCHYR: *Testsystem für integrierte, hochvernetzte Sicherheitssysteme*. ATZ Automobiltechnische Zeitschrift, 2012-12, Springer Vieweg, Seiten 980-987, Wiesbaden, 2012.
- [103] SATTLER, K., R. RAITH und T. BRANDMEIER: *Crashfähiges Testsystem für vernetzte Sicherheitssysteme im Automobil*. 10. Magdeburger Maschinenbau-Tage, Otto-von-Guericke-Universität Magdeburg, Magdeburg, 2011.
- [104] SATTLER, K., R. RAITH und T. BRANDMEIER: *Efficient test methods for the system test of highly networked safety systems*. 2012 Proceedings of the Tenth Workshop on Intelligent Solutions in Embedded Systems, WISES 2012, IEEE Conference Proceedings, Alpen-Adria Universität, Seiten 69-75, Klagenfurt, Österreich, 2012.
- [105] SATTLER, K., R. RAITH, C. SCHYR, D. SADOU und T. BRANDMEIER: *Maneuver-based testing of integrated, highly interconnected safety systems*. FISITA 2012 World Automotive Congress, Proceedings of the FISITA 2012 World Automotive Congress, Volume 8: Vehicle Design and Testing (II), Springer-Verlag Berlin Heidelberg, Seiten 1677-1689, Beijing, China, 2012.
- [106] SATTLER, K., D. STADLER, T. BRANDMEIER, J. REINER, S. HAKULI und D. SADOU: *Maneuverbased Testing in Combination with Evolutionary Algorithms for the System Test of Safety Systems*. FISITA World Automotive Congress, Paper-Nummer F2014-AST-003, Maastricht, Niederlande, 2014.
- [107] SAX, E.: *Automatisiertes Testen Eingebetteter Systeme in der Automobilindustrie*. Carl Hanser Verlag, München, 2008.
- [108] SCHICK, B., R. BÜTTNER, K. BALTRUSCHAT, G. MEIER und H. JAKOB: *Bewertung der Funktion und Güte von Fahrerassistenzsystemen bei aktivem Bremsengriff*. ATZ worldwide, Ausgabe 05/2007, Seiten 14-18, 2007.

-
- [109] SCHMID, H.: *Konzeption einer pragmatischen Testmethodik für den Test von eingebetteten Systemen*. Universität Ulm, Dissertation, Ulm, 2003.
- [110] SCHMIDT, C.: *Hardware-in-the-Loop gestützte Entwicklungsplattform für Fahrerassistenzsysteme - Analyse und Generierung kritischer Verkehrsszenarien*. kassel university press GmbH, Dissertation, Kassel, 2010.
- [111] SCHMITZ, F.: *Entwicklung einer Methodik zur Bewertung der integralen Sicherheit von Kraftfahrzeugen*. Shaker Verlag, Dissertation, Aachen, 2013.
- [112] SCHOLZ, P.: *Softwareentwicklung eingebetteter Systeme: Grundlagen, Modellierung, Qualitätssicherung*. Springer-Verlag, Berlin, 2005.
- [113] SCHRAMM, D., M. HILLER und R. BARDINI: *Modellbildung und Simulation der Dynamik von Kraftfahrzeugen*. Springer Vieweg, Berlin, 2. Auflage, 2013.
- [114] SCHÄUFFELE, J. und T. ZURAWKA: *Automotive Software Engineering*. Springer Vieweg, Wiesbaden, 5. Auflage, 2013.
- [115] SIEBERTZ, K., D. VAN BEBBER und T. HOCHKIRCHEN: *Statistische Versuchsplanung*. Springer-Verlag, Heidelberg, 2010.
- [116] SIGL, S., T. GOLLEWSKI, T. MIEHLING und A. KUHN: *About Development Process And Accompanying Performance Evaluations Of Integral Automotive Safety Systems*. FISITA World Automotive Congress, Paper-Nummer F2014-AST-084, Maastricht, Niederlande, 2014.
- [117] SKUTEK, M., D.T. LINZMEIER, N. APPENRODT und G. WANIELIK: *A precrash system based on sensor data fusion of laser scanner and short range radars*. 8th International Conference on Information Fusion, Seiten 1287-1294, Philadelphia, Pennsylvania, USA, 2005.
- [118] SNEED, H.M., M. BAUMGARTNER und R. SEIDL: *Der Systemtest*. Carl Hanser Verlag, München, 3. Auflage, 2012.
- [119] SNEED, H.M., R. SEIDL und M. BAUMGARTNER: *Software in Zahlen*. Carl Hanser Verlag, München, 2010.
- [120] SPILLNER, A., T. ROSSNER, M. WINTER und T. LINZ: *Praxiswissen Softwaretest - Testmanagement*. dpunkt.verlag GmbH, Heidelberg, 2008.
- [121] SPRAVE, J.: *Evolutionäre Algorithmen zur Parameteroptimierung*. at - Automatisierungstechnik, 43 (1995) 3, Seiten 110-117, 1995.
- [122] STADLER, D.: *Analyse und Optimierung des Testprozesses im Systemtest vorausschauender, kamerabasierter Fahrerassistenzsysteme*. Masterarbeit, Hochschule Landshut, 2013 (Betreute Masterarbeit, nicht öffentliche Quelle).

- [123] STATISTISCHES BUNDESAMT: *Verkehr - Verkehrsunfälle*. Destatis, Fachserie 8, Reihe 7, Dezember 2013, Wiesbaden, 2014.
- [124] STEVENS, R., P. BROOK, K. JACKSON und S. ARNOLD: *Systems Engineering. Coping with Complexity*. Prentice Hall Europe, Harlow, United Kingdom, 1998.
- [125] STIRGWOLT, P.: *Effective management of functional safety for ISO 26262 standard*. Reliability and Maintainability Symposium (RAMS), 2013 Proceedings - Annual, Seiten 1-6, Orlando, Florida, USA, 2013.
- [126] STOCKS, P. und D. CARRINGTON: *A framework for specification-based testing*. IEEE Transactions on Software Engineering, Volume 22, Issue 11, Seiten 777-793, 1996.
- [127] STORN, R.M.: *On the usage of Differential Evolution for Function Optimization*. Fuzzy Information Processing Society, NAFIPS, 1996 Biennial Conference of the North American Fuzzy Information Processing Society, Seiten 1-5, Berkeley, Kalifornien, USA, 1996.
- [128] STORN, R.M. und K.V. PRICE: *Differential evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces*. ICSI, USA, Tech. Rep. TR-95-012, University of California, Berkeley, Seiten 1-15, Berkeley, Kalifornien, USA, 1995.
- [129] STORN, R.M. und K.V. PRICE: *Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces*. Journal of Global Optimization, Vol. 11, No. 4, Kluwer Academic Publishers, Seiten 341-359, Dordrecht, Niederlande, 1997.
- [130] STRUTZ, T. und A. ROLL: *Innovativ und sicher dank modularem Querbaukasten*. ATZ extra, Volume 17, Number 6, Seiten 34-37, 2012.
- [131] TAGUCHI, G., S. CHOWDHURY und Y. WU: *Taguchi's Quality Engineering Handbook*. John Wiley & Sons, Hoboken, New Jersey, USA, 2005.
- [132] TELLMANN, D.: *Hardware-in-the-Loop gestützte Entwicklungsplattform für Fahrerassistenzsysteme - Modelle der Umfeldsensorik und angepasste Fahrermodelle*. kassel university press GmbH, Dissertation, Kassel, 2012.
- [133] UNBEHAUEN, H.: *Regelungstechnik I*. Vieweg + Teubner Verlag, Wiesbaden, 15. Auflage, 2008.
- [134] VOGEL-HEUSER, B.: *Systems Software Engineering*. Oldenbourg Industrieverlag, München, 2003.
- [135] VOLLMUTH, H. und R. ZWETTLER: *Kennzahlen*. Haufe Verlag, Freiburg, 2008.
- [136] WARD, D.D. und S.E. CROZIER: *The uses and abuses of ASIL decomposition in ISO 26262*. 7th IET International Conference on System Safety and Cyber Security 2012, Seiten 1-6, Edinburgh, United Kingdom, 2012.

- [137] WARD, D.D. und I. IBARRA: *Development phase in accordance with ISO 26262*. 8th IET International Conference on System Safety and Cyber Security 2013, Seiten 1-6, Cardiff, United Kingdom, 2013.
- [138] WEICKERT, K.: *Evolutionäre Algorithmen*. B.G. Teubner Verlag, Wiesbaden, 2. Auflage, 2007.
- [139] WINTERHAGEN, J.: *Das Superhirn im Auto*. Pictures of the Future, Siemens AG, Ausgabe Herbst, Seiten 64-65, 2012.
- [140] WONG, W.E. und J.R. HORGAN: *A Study of Effective Regression Testing in Practice*. The Eighth International Symposium on Software Reliability Engineering, ISSRE '97, Seiten 264-274, 1997.
- [141] ZAHARIE, D.: *Influence of crossover on the behavior of Differential Evolution Algorithms*. Applied Soft Computing, Seiten 1126–1138, Vol. 9, No. 3, 2009.
- [142] ZHANG, X., C. NIE, B. XU und B. QU: *Test Case Prioritization based on Varying Testing Requirement Priorities and Test Case Costs*. Seventh International Conference on Quality Software, QSIC '07, Seiten 15-24, 2007.

