Technische Universität München

TUM School of Engineering and Design

Model Validation and Uncertainty Aggregation for Safety Assessment of Automated Vehicles

Stefan Riedmaier, M.Sc.

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:	Prof. Dr. phil. Klaus Bengler
Prüfer der Dissertation:	1. Prof. Dr.-Ing. Markus Lienkamp
	2. Prof. Bernhard Schick

Die Dissertation wurde am 11.08.2021 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 15.02.2022 angenommen.

Abstract

Automated vehicles have the potential to prevent numerous traffic accidents with human causes. Since a proof of safety is essential but costly, testing is shifting more and more from the real to the virtual world. However, virtual safeguarding must be accompanied by model validation to ensure the applicability of the safety case and enable the benefits of simulation. Unfortunately, this is rarely achieved in the current state of the art of automated vehicles. This thesis provides a comprehensive survey about safety assessment and type approval, as well as about model validation theory and applications across several engineering fields. It concludes that there is a research gap in validation methods that quantify modeling uncertainties by means of real comparisons and aggregate them to the purely virtual decision making of the application.

This thesis introduces a novel validation framework that connects a verification, calibration, validation, and application domain through individual process steps. The framework is developed in several manifestations to cover varying types of simulation models, such as (non-)deterministic, hierarchical, time-(in)variant, or formal simulations. The framework is first presented in its generic form with several configuration options and then configured for the specific use case of type approval of automated vehicles. The first step begins with the independent design of validation and application scenarios. The former are executed in real experiments and subsequently re-simulated for comparison, whereas the latter are used exclusively for model predictions. After an assessment of the results, a validation metric quantifies the modeling uncertainties at the validation scenarios. However, these are not neglected as usual but modeled via statistical regression so that they can be inferred to new application scenarios. Finally, the uncertainties are added as bounds around the nominal model predictions to obtain additional confidence in decision making of the application.

The validation methodology is itself validated by intentionally injecting modeling errors to determine if it can identify and correct them. This is analyzed by means of a binary classifier that relates the approval decisions of the bounded model predictions with the true ones. The classifier results show a perfect recall rate at a high precision, indicating that no safety-critical cases are missed by the methodology. Then, the validation framework is applied to the actual type approval of an automated vehicle, including real experiments on the road and simulations in a hybrid test environment. The validation methodology successfully uncovers modeling errors that are particularly valuable to the developers for future improvements. The thesis concludes with a comprehensive discussion of the results and the original research objectives. The validation framework closes an important gap for virtual safeguarding of automated vehicles and beyond.

Kurzfassung

Automatisierte Fahrzeuge haben das Potential, zahlreiche Verkehrsunfälle mit menschlicher Ursache zu verhindern. Da ein Sicherheitsnachweis unerlässlich, aber aufwendig ist, verlagert sich das Testen immer mehr von der realen in die virtuelle Welt. Allerdings muss die virtuelle Absicherung von einer Modellvalidierung begleitet werden, um die Anwendbarkeit des Sicherheitsnachweises zu gewährleisten und die Vorteile der Simulation zu nutzen. Leider wird dies im aktuellen Stand der Technik von automatisierten Fahrzeugen nur selten erreicht. Diese Arbeit gibt einen umfassenden Überblick über die Sicherheitsbewertung und Typgenehmigung sowie über die Theorie und Anwendung der Modellvalidierung in verschiedenen technischen Feldern. Sie kommt zu dem Schluss, dass es eine Forschungslücke bei Validierungsmethoden gibt, die Modellierungsunsicherheiten durch reale Vergleiche quantifizieren und diese zur rein virtuellen Entscheidungsfindung der Anwendung aggregieren.

In dieser Arbeit wird ein neuartiges Validierungs-Framework vorgestellt, das eine Verifikations-, Kalibrierungs-, Validierungs- und Anwendungsdomäne durch einzelne Prozessschritte miteinander verbindet. Das Framework wird in mehreren Ausprägungen entwickelt, um unterschiedliche Arten von Simulationsmodellen wie (nicht-)deterministische, hierarchische, zeit-(in)variante oder formale Simulationen abzudecken. Das Framework wird zunächst in seiner generischen Form mit mehreren Konfigurationsmöglichkeiten vorgestellt und dann für den spezifischen Anwendungsfall der Typgenehmigung von automatisierten Fahrzeugen konfiguriert. Der erste Schritt beginnt mit dem unabhängigen Entwurf von Validierungs- und Anwendungsszenarien. Erstere werden in realen Experimenten durchgeführt und anschließend zum Vergleich re-simuliert, während letztere ausschließlich für Modellvorhersagen verwendet werden. Nach einer Bewertung der Ergebnisse quantifiziert eine Validierungsmetrik die Modellierungsunsicherheiten an Validierungsszenarien. Diese werden jedoch nicht wie üblich vernachlässigt, sondern mittels statistischer Regression modelliert, sodass sie auf neue Anwendungsszenarien prädiziert werden können. Schließlich werden die Unsicherheiten als Schranken um die nominalen Modellvorhersagen addiert, um zusätzliche Entscheidungssicherheit in der Anwendung zu erhalten.

Die Validierungsmethodik wird validiert, indem absichtlich Modellierungsfehler injiziert werden, um festzustellen, ob sie diese identifizieren und korrigieren kann. Dies wird mit Hilfe eines binären Klassifikators analysiert, der die Zulassungsentscheidungen der beschränkten Modellvorhersagen mit den wahren in Beziehung setzt. Die Ergebnisse des Klassifikators zeigen eine perfekte Sensitivität bei hoher Präzision, was darauf hindeutet, dass keine sicherheitskritischen Fälle von der Methodik übersehen werden. Anschließend wird das Validierungsframework auf die tatsächliche Typgenehmigung eines automatisierten Fahrzeugs angewendet, einschließlich realer Versuche auf der Straße und Simulationen in einer hybriden Testumgebung. Die Validierungsmethodik deckt erfolgreich Modellierungsfehler auf, die für die Entwickler besonders wertvoll für zukünftige Verbesserungen sind. Die Arbeit endet mit einer umfassenden Diskussion der Ergebnisse und der ursprünglichen Forschungsziele. Das Framework schließt eine wichtige Lücke für die virtuelle Absicherung von automatisierten Fahrzeugen und darüber hinaus.

All models are wrong,
but some are useful.

George E. P. Box

Acknowledgement

This thesis was written during my time from 2019 to 2021 as research assistant at the Institute of Automotive Technology at the Technical University of Munich and my time from 2017 to 2019 as research assistant at the Adrive LivingLab at the Kempten University of Applied Sciences. The entire time over 3 years and 10 months was characterized by a joint cooperation between the two universities and the project partner TÜV SÜD Auto Service GmbH.

First and foremost, I would like to thank my two supervisors Prof. Dr.-Ing. Markus Lienkamp and Prof. Bernhard Schick for giving me this great opportunity and for all their trust and support during this formative period. The start-up of the Adrive LivingLab in Kempten was very exciting for me, characterized by flexible work in a growing team and building up research and education from scratch. The following time at FTM was a completely different but equally exciting experience learning from an renowned institute and its team built up over decades. In addition, I would like to thank Prof. Dr. phil. Klaus Bengler for taking over the chairmanship.

I would like to express my gratitude to TÜV SÜD Auto Service GmbH for financing the research project. My special thanks go to my supervisors Christian Gnandt, Houssem Abdellatif, and Emmeram Klotz, who gave me the opportunity and confidence to work on an important research topic. I would also like to say thank you to all my colleagues from the TÜV SÜD HAD department, who were always available for valuable discussions. Furthermore, my special gratitude goes to the Bavarian Academic Forum (BayWiss) and in particular the Joint Academic Partnership "Mobility and Transport", which supports the indispensable cooperation between universities.

I wish to express my sincere appreciation to my former FTM and Adrive colleagues for an exciting, instructive, and unforgettable time. Thank you very much for your commitment, your support, and the time spent together. My special thanks go to my colleagues Thomas Ponn, Benedikt Danquah, Tim Stahl, Andreas Schimpe, Jakob Schneider, Stefan Schneider, Daniel Schneider, Jonas Nesensohn, Johann Haselberger, and Kmeid Saad, who have worked closely with me in projects, teaching, and publications. In addition, I would especially like to thank my FTM group leader Dr.-Ing. Frank Diermeyer, who accompanied me during the entire time with professional advice. Many thanks to all my students, for their interest in my topics and their confidence in my supervision, as well as to my proofreaders Thomas Ponn, Benedikt Danquah, and Tim Stahl.

My family, especially my parents and brother, deserve my deepest gratitude. You have always encouraged and supported me in my plans. Without you this would not have been possible.

Garching, January 2021

Stefan Riedmaier

Contents

List of Abbreviations ..... **V**

Formula Symbols ..... **VII**

1 Introduction **1**

1.1 Research Motivation **1**

1.2 Research Objectives **2**

1.3 Structure of the Thesis **3**

1.4 Publications **4**

2 State of the Art **7**

2.1 Safety Assessment of Automated Vehicles **7**

2.1.1 Terms and Definitions 7

2.1.2 Overview of Safety Assessment Approaches 8

2.1.3 Scenario-Based Approach 10

2.2 Virtual-Based Homologation **13**

2.2.1 Regulation 140: Electronic Stability Control 13

2.2.2 Regulation 79: Automatically Commanded Steering Function 15

2.2.3 Regulation 157: Automated Lane Keeping System 15

2.3 Model Validation Theory **16**

2.3.1 Terms and Definitions 16

2.3.2 Error and Uncertainty Types 18

2.3.3 Error and Uncertainty Sources 19

2.3.4 Model and Simulation Types 21

2.4 Model Validation across Engineering Fields **23**

2.4.1 Automotive Model Validation 23

2.4.2 Railway Model Validation 27

2.4.3 Aircraft Model Validation 28

2.4.4 Model Validation in Numerical Fields 28

2.5 Criticism of the State of the Art **31**

2.5.1 Safety Assessment of Automated Vehicles 31

2.5.2 Virtual-Based Homologation 32
2.5.3 Model Validation 33
2.6 Research Gaps 36
2.7 Research Questions 37
3 Model Validation Methodology 39
3.1 Requirements for the Methodology 39
3.2 Overall Validation Framework 40
3.2.1 Framework Overview Based on Continuous Example 40
3.2.2 Structure of this Chapter 43
3.2.3 Framework Domains 43
3.2.4 Framework Blocks 44
3.2.5 Framework Notation 45
3.2.6 Framework Manifestations 46
3.2.7 Framework Configuration for Automated Vehicle Approval 49
3.3 Scenario Design 51
3.3.1 Selection of Scenario Design Methods 51
3.3.2 Data-driven Application Scenarios via Event Finding 52
3.3.3 Coverage-based Validation Scenarios via Map Planning 53
3.3.4 Scenario Design with Nested Sampling of Uncertainties 55
3.4 System and Model Assessment 58
3.5 Validation Metric 59
3.5.1 Overview and Selection of Validation Metrics 59
3.5.2 Absolute Deviation and Area Validation Metric 61
3.6 Error Learning and Inference 62
3.6.1 Ensemble Validation versus Point-by-Point Validation 63
3.6.2 Overview and Selection of Learning Techniques 63
3.6.3 Linear Regression with External Prediction Intervals 64
3.7 Aggregation of Errors and Uncertainties 65
3.7.1 Overview and Selection of Aggregation Techniques 65
3.7.2 Uncertainty Expansion of Model Responses 66
3.8 Decision Making 68
4 Validation Results 69
4.1 Validation of the Methodology via the Method of Manufactured Universes 70
4.1.1 Introduction into the Method of Manufactured Universes 70

4.1.2 Binary Classification of Type-Approval Decisions 70
4.1.3 Creation of the Manufactured Universe 72
4.1.4 Approval Results of the (Non-)Deterministic Manifestation 73
4.1.5 Classification Results of the (Non-)Deterministic Manifestation 75
4.1.6 Discussion of the Classification Results 77
4.2 Application of the Validation Methodology Based on Real Driving Tests 78
4.2.1 Final Framework Configuration 78
4.2.2 Coverage-Based Validation Scenarios 79
4.2.3 Validation Experiments on the Real Road 80
4.2.4 Re-Simulation of the Validation Experiments 81
4.2.5 Data-Driven Application Scenarios 82
4.2.6 Assessment 83
4.2.7 Validation Metric 84
4.2.8 Error Learning and Inference 85
4.2.9 Uncertainty Expansion 86
4.2.10 Type Approval 86
4.2.11 Discussion of the Results 88
5 Discussion 89
5.1 Fulfillment of the Requirements 89
5.2 Response to the Research Questions 90
5.3 Fulfillment of the Research Objectives 92
5.4 Limitations and Outlook 93
5.4.1 Framework Blocks 93
5.4.2 Use Case and Extension 95
6 Summary 97
List of Figures i
List of Tables v
Bibliography vii
Prior Publications xxxvii
Supervised Student's Thesis xxxix
Appendix xli

List of Abbreviations

AAVM	Asymmetrical Area Validation Metric
ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance System
AV	Automated Vehicle
AVM	Area Validation Metric
CDF	Cumulative Distribution Function
FN	False Negative
FP	False Positive
HiL	Hardware-in-the-Loop
KPI	Key Performance Indicator
LKA	Lane Keeping Assist
MAVM	Modified Area Validation Metric
MiL	Model-in-the-Loop
MMU	Method of Manufactured Universes
ODD	Operational Design Domain
PBA	Probability Bound Analysis
p-box	probability box
PDF	Probability Density Function
PI	Prediction Interval
PoC	Proof of Concept
SBA	Scenario-Based Approach
SiL	Software-in-the-Loop
TN	True Negative
TP	True Positive
UQ	Uncertainty Quantification
V&V	Verification & Validation
VEHiL	Vehicle-Hardware-in-the-Loop
ViL	Vehicle-in-the-Loop
VV&UQ	Verification, Validation & Uncertainty Quantification
XiL	X-in-the-Loop

Formula Symbols

Symbol[1]	Unit	Domain	Description
α	–	$\mathbb{R}$	Placeholder symbol[2]
$\boldsymbol{\alpha}$	–	$\mathbb{R}^N$	Vector representation of a symbol[3]
A	–	—	Random variable of a symbol
$\mathbf{A}$	–	$\mathbb{R}^{N_1 \times N_2}$	Matrix representation of a symbol
α^a	–	$\mathbb{R}$	Symbol within the application domain
α^c	–	$\mathbb{R}$	Symbol within the calibration domain
α^n	–	$\mathbb{R}$	Symbol within the (numerical) verification domain
α^v	–	$\mathbb{R}$	Symbol within the validation domain
α_m	–	$\mathbb{R}$	Symbol of the simulation model
α_s	–	$\mathbb{R}$	Symbol of the physical system
$\hat{\alpha}$	–	$\mathbb{R}$	Estimator of a symbol
$\underline{\alpha}$	–	$\mathbb{R}$	Lower interval bound of a symbol
$\overline{\alpha}$	–	$\mathbb{R}$	Upper interval bound of a symbol
$\langle\alpha\rangle$	–	$\mathbb{R}$	Mean value of a symbol
$B(\alpha)$	–	—	P-box representation of a symbol
$D(\alpha)$	–	—	Data set representation of a symbol[9]
$f(\alpha)$	–	—	PDF representation of a symbol
$F(\alpha)$	–	—	CDF representation of a symbol
$\underline{F}(\alpha)$	–	—	Left CDF edge of p-box representation of a symbol
$\overline{F}(\alpha)$	–	—	Right CDF edge of p-box representation of a symbol
$I(\alpha)$	–	—	Interval representation of a symbol
α	%	$\mathbb{R}$	Statistical confidence
$\mathbf{a}_y$	$\mathrm{m\,s^{-2}}$	$\mathbb{R}^{N_t}$	Measured lateral acceleration signal
$a_{y,\mathrm{ref}}$	$\mathrm{m\,s^{-2}}$	$\mathbb{R}$	Reference lateral acceleration parameter
$\mathbf{a}_{y,\mathrm{ref}}$	$\mathrm{m\,s^{-2}}$	$\mathbb{R}^{N_t}$	Reference lateral acceleration signal
$a_{y,\mathrm{smax}}$	$\mathrm{m\,s^{-2}}$	$\mathbb{R}$	Specified maximum lateral acceleration
$\mathbf{b}$	–	$\mathbb{B}^{N_t}$	Binary event signal
$\mathbf{b}_{a,i}$	–	$\mathbb{B}^{N_t}$	Binary acceleration event signal of i-th bin
$\tilde{\mathbf{b}}_{a,i}$	–	$\mathbb{B}^{N_t}$	Binary connected acceleration event signal of i-th bin
$\mathbf{b}_{\mathrm{add}}$	–	$\mathbb{B}^{N_t}$	Binary additional condition event signal
$\mathbf{b}_i$	–	$\mathbb{B}^{N_t}$	Binary AND-conjunct event signal of i-th bin
$\mathbf{b}_v$	–	$\mathbb{B}^{N_t}$	Binary velocity event signal
C_{abs}	–	$\mathbb{N}$	Conservativeness measure as absolute scenario count

Symbol[1]	Unit	Domain	Description
C_{rel}	%	$\mathbb{R}$	Conservativeness measure relative to N^a
d	–	$\mathbb{B}$	Binary decision of a single KPI and a single scenario
d^a	–	$\mathbb{B}^{N_y}$	True application decision at a single scenario
$\hat{d}^a$	–	$\mathbb{B}$	Estimated application decision at a single scenario
$\hat{\mathbf{d}}^a$	–	$\mathbb{B}^{N_y}$	Estimated application decision at multiple scenarios
$\hat{d}^a_{\text{mac}}$	–	$\mathbb{B}$	Estimated macroscopic application decision
d^v	–	$\mathbb{B}$	Validation decision at a single scenario
$\mathbf{d}^v$	–	$\mathbb{B}^{N_y}$	Validation decision at multiple scenarios
d^v_{mac}	–	$\mathbb{B}$	Macroscopic validation decision
e	–	$\mathbb{R}$	Error
E	–	—	Random variable of the error
e^n	–	$\mathbb{R}$	Numerical error
$\hat{e}^{na}$	–	$\mathbb{R}$	Estimated numerical error at a single application scenario
$I(\hat{e}^{na})$	–	—	Numerical error interval at a single application scenario
e^c	–	$\mathbb{R}$	Calibration error
$\hat{e}^{ca}$	–	$\mathbb{R}$	Estimated calibration error at a single application scenario
e^v	–	$\mathbb{R}$	Validation error at a single validation scenario
$\mathbf{e}^v$	–	$\mathbb{R}^{N^v}$	Validation error at multiple validation scenarios
$\hat{e}^v$	–	$\mathbb{R}$	Estimated validation error at a single scenario
$I(e^v)$	–	—	Validation error interval at a single validation scenario (model-form uncertainty)
$I(\mathbf{e}^v)$	–	—	Validation error interval at multiple validation scenarios
e^v_l	–	$\mathbb{R}$	Left validation error at a single validation scenario
e^v_r	–	$\mathbb{R}$	Right validation error at a single validation scenario
e^{va}	–	$\mathbb{R}$	True validation error at a single application scenario
$\hat{e}^{va}$	–	$\mathbb{R}$	Estimated validation error at a single application scenario
$\underline{e}^{va}$	–	$\mathbb{R}$	Lower validation error at a single application scenario
$\overline{e}^{va}$	–	$\mathbb{R}$	Upper validation error at a single application scenario
$I(\hat{e}^{va})$	–	—	Validation error interval at a single application scenario
$\hat{e}^{va}_l$	–	$\mathbb{R}$	Estimated left validation error at single application scenario
$\hat{e}^{va}_r$	–	$\mathbb{R}$	Estimated right validation error at single application scenario
$\hat{e}^{vv}$	–	$\mathbb{R}$	Estimated validation error at a single validation scenario
$\hat{\mathbf{e}}^{vv}$	–	$\mathbb{R}^{N^v}$	Estimated validation error at multiple validation scenarios
e_h	–	$\mathbb{R}$	Error due to numerical discretization
e_m	–	$\mathbb{R}$	Error due to the model-form
e_θ	–	$\mathbb{R}$	Error due to model parameters
e_x	–	$\mathbb{R}$	Error due to scenario parameters / inputs
$e_{y,\text{obs}}$	–	$\mathbb{R}$	Error due to observation/measurement
$\mathbb{E}[\cdot]$	–	—	Expected value operator
g	–	—	Function, mapping
g_{kpi}	–	—	KPI extraction function
g_m	–	—	Simulation model function (model-form)
$\underline{g}_m$	–	—	Lower interval model function

Symbol[1]	Unit	Domain	Description
$\overline{g}_m$	–	—	Upper interval model function
g_{mat}	–	—	Mathematical model function (analytical)
g_p	–	—	Prediction interval function
$g_{p,l}$	–	—	Prediction interval function of the left error
$g_{p,r}$	–	—	Prediction interval function of the right error
g_s	–	—	Physical system function
g^a_{dec}	–	—	Application decision making function
g^a_{int}	–	—	Application error integration function
g^a_{maa}	–	—	Macroscopic application assessment function
g^a_{mad}	–	—	Macroscopic application decision making function
g^c_{met}	–	—	Calibration metric function
g^n_{met}	–	—	Numerical verification metric function
g^v_{dec}	–	—	Validation decision making function
g^v_{mad}	–	—	Macroscopic validation decision making function
g^v_{met}	–	—	Validation metric function
g^v_{lea}	–	—	Validation error learning function
g^{va}_{inf}	–	—	Validation to application error inference function
h	s	$\mathbb{R}$	Step size of the simulation
i	–	$\mathbb{N}$	Index of the acceleration ranges/bins
j	–	$\mathbb{N}$	Index of the number of events per bin
k	–	$\mathbb{N}$	Index of the velocity bins (only coverage-based algorithm)
κ	m^{-1}	$\mathbb{R}$	Road curvature
μ	–	$\mathbb{R}$	Mean value
N	–	$\mathbb{N}$	Number of elements, cardinality
N_t	–	$\mathbb{N}$	Number of time steps
N_x	–	$\mathbb{N}$	Number of scenario parameters / inputs
N_y	–	$\mathbb{N}$	Number of response KPIs / outputs
N^a	–	$\mathbb{N}$	Number of application scenarios
N^v	–	$\mathbb{N}$	Number of validation scenarios
N^v_r	–	$\mathbb{N}$	Number of validation experiment repetitions
$P(\cdot)$	–	—	Probability of an event
P	%	$\mathbb{R}$	Precision of a binary classifier
R	%	$\mathbb{R}$	Recall of a binary classifier
$\mathbf{r}_l$	%	$\mathbb{R}^9$	Lower limits of the acceleration ranges/bins
$\mathbf{r}_u$	%	$\mathbb{R}^9$	Upper limits of the acceleration ranges/bins
R	m	$\mathbb{R}$	Curve radius parameter
σ	–	$\mathbb{R}$	Standard deviation
s	–	$\mathbb{R}$	Sample standard deviation
s_r	°	$\mathbb{R}$	Road slope parameter
θ, θ_m	–	$\mathbb{R}^{N_\theta}$	Simulation model parameters

Symbol[1]	Unit	Domain	Description
θ_s	–	$\mathbb{R}^{N_\theta}$	System parameters
Θ	–	—	Random variable of simulation model parameters
$f(\theta)$	–	—	PDF of simulation model parameters
$I(\theta)$	–	—	Interval of simulation model parameters
$B(\theta)$	–	—	P-box of simulation model parameters
$t_N^{\alpha/2}$	–	—	Function of the t-distribution
t	s	$\mathbb{R}$	Continuous time
t_e	s	$\mathbb{R}$	End time of an event
t_s	s	$\mathbb{R}$	Start time of an event
Δt	s	$\mathbb{R}$	Time duration of an event
t_l	kg	$\mathbb{R}$	Tank load parameter
t_y^a	–	$\mathbb{R}$	Decision making threshold of the application KPI
t_e^v	–	$\mathbb{R}$	Decision making threshold of the validation error
v_x	$\mathrm{m\,s^{-1}}$	$\mathbb{R}$	Longitudinal velocity parameter
$\mathbf{v}_x$	$\mathrm{m\,s^{-1}}$	$\mathbb{R}^{N_t}$	Longitudinal velocity signal
$v_{x,\mathrm{smin}}$	$\mathrm{m\,s^{-1}}$	$\mathbb{R}$	Specified minimum longitudinal velocity
$v_{x,\mathrm{smax}}$	$\mathrm{m\,s^{-1}}$	$\mathbb{R}$	Specified maximum longitudinal velocity
v_w	$\mathrm{m\,s^{-1}}$	$\mathbb{R}$	Wind velocity parameter
x	–	$\mathbb{R}$	Single scenario with a single input parameter (SI)
$\mathbf{x}$	–	$\mathbb{R}^{N_x+1}$	Single scenario with multiple input parameters (MI)
X	–	—	Random variable of a scenario
$f(\mathbf{x})$	–	—	PDF of scenario parameters
$I(\mathbf{x})$	–	—	Interval of scenario parameters
$B(\mathbf{x})$	–	—	P-box of scenario parameters
x^a	–	$\mathbb{R}$	Single application scenario with a single parameter
$\mathbf{x}^a$	–	$\mathbb{R}^{N_t}$	Single application scenario with a single signal
$\mathbf{x}^a$	–	$\mathbb{R}^{N_x+1}$	Single application scenario (with multiple parameters)
$\mathbf{X}^a$	–	$\mathbb{R}^{N^a\times(N_x+1)}$	Multiple application scenarios
$\mathbf{x}^v$	–	$\mathbb{R}^{N_x+1}$	Single validation scenario
$\mathbf{X}^v$	–	$\mathbb{R}^{N^v\times(N_x+1)}$	Multiple validation scenarios
$D(\mathbf{x}_s^c, y_s^c), D^c$	–	—	Calibration data set
x, y, z	m	$\mathbb{R}$	Euclidian world coordinates
y	–	$\mathbb{R}$	Single KPI at a single scenario
Y	–	—	Random variable of a KPI
y_{exact}	–	$\mathbb{R}$	Exact mathematical model KPI
y_m	–	$\mathbb{R}$	Simulation model KPI
Y_m	–	—	Random variable of a model KPI
$f(y_m)$	–	—	PDF of simulation model KPI
$I(y_m)$	–	—	Interval of simulation model KPI
$B(y_m)$	–	—	P-box of simulation model KPI
y_s	–	$\mathbb{R}$	System KPI

Symbol[1]	Unit	Domain	Description
Y_s	–	—	Random variable of a system KPI
y_{true}	–	$\mathbb{R}$	Nature's true KPI (without measurement errors)
y_m^a	–	$\mathbb{R}$	Model KPI at a single application scenario
$\mathbf{y}_m^a$	–	$\mathbb{R}^{N_t}$	Model output signal at a single application scenario
$B(y_m^a)$	–	—	Model KPI p-box at a single application scenario
$\hat{y}_{\text{mac}}^a$	–	$\mathbb{R}$	Estimated macroscopic application assessment KPI
y_s^a	–	$\mathbb{R}$	True system KPI at a single application scenario
$B(y_s^a)$	–	—	True system KPI p-box at a single application scenario
$\hat{y}_s^a$	–	$\mathbb{R}$	Estimated system KPI at a single application scenario
$I(\hat{y}_s^a)$	–	—	Estimated system KPI interval at a single appl. scenario
$B(\hat{y}_s^a)$	–	—	Estimated system KPI p-box at a single application scenario
y_m^c	–	$\mathbb{R}$	Model KPI at a single calibration scenario
y_s^c	–	$\mathbb{R}$	System KPI at a single calibration scenario
y_m^n	–	$\mathbb{R}$	Model KPI at a single numerical scenario
y_{exact}^n	–	$\mathbb{R}$	Exact mathematical KPI at a single numerical scenario
y_m^v	–	$\mathbb{R}$	Model KPI at a single validation scenario
$F(y_m^v)$	–	—	Model KPI CDF at a single validation scenario
$B(y_m^v)$	–	—	Model KPI p-box at a single validation scenario
y_s^v	–	$\mathbb{R}$	System KPI at a single validation scenario
$F(y_s^v)$	–	—	System KPI CDF at a single validation scenario
y	m	$\mathbb{R}$	Minimum distance to line KPI
$y_{l,\min}$	m	$\mathbb{R}$	Minimum distance to left line KPI
$\mathbf{y}_l$	m	$\mathbb{R}^{N_t}$	Minimum distance to left line signal
$y_{r,\min}$	m	$\mathbb{R}$	Minimum distance to right line KPI
$\mathbf{y}_r$	m	$\mathbb{R}^{N_t}$	Minimum distance to right line signal
$\text{Var}[\cdot]$	–	—	Variance operator
$\mathbf{w}$	–	$\mathbb{R}^{N_x+1}$	Regression weights

[1]The entries are sorted first by the symbol, then the decorator, then the upper index, and then the lower index.

[2]The symbol α is used as a placeholder to demonstrate the notation at the start of the list of symbols and in Chapter 3.2.5. Afterwards, there can emerge a multitude of combinations, from which the ones used in this thesis are included. The origin of the symbols is highlighted by underlining the words at the corresponding letters. If no letter is underlined, the symbol was set for other reasons like the usual x and y for inputs and outputs. The domains are given to highlight the dimensions and types. In case of functions, there is no domain given, since they perform a mapping between domains. In case of percentages, the domain is actually $[0, 1] \subset \mathbb{R}$.

[3]The framework symbols used in this thesis focus on a Multiple-Input-Single-Output constellation. Therefore, the inputs $\mathbf{x}$ are usually bold, while the output y, as well as the derived error e and decision d are italic.

1 Introduction

Every year, 25,000 people lose their lives on our roads. The vast majority of these accidents are caused by human error. We can and must act to change this. [...] Now we raise the safety level across the board, and pave the way for connected and automated mobility of the future.

European Commissioner Elżbieta Bieńkowska [1]

1.1 Research Motivation

Automated driving is one of the biggest trends in transportation, as it raises great hopes among society, politics, and vehicle manufacturers that it will make traffic safer and more comfortable [1]. In 2018, more than one million people died in road accidents worldwide [2] and more than 25,000 of them in the 28 member states of the European Union [3]. Governments are striving to reduce these figures by increasing the automation of modern vehicles. The European Commission has started to make the Electronic Stability Control mandatory in new vehicles in 2014 and will continue to make the Emergency Braking Assist and the Lane Keeping Assist (LKA) mandatory in 2022 [1]. This strategy of mandatory introduction of an Advanced Driver Assistance System (ADAS) — ranked as Level 1 according to SAE [4] — will transition to higher automation levels in the future.

Before Automated Vehicles (AVs) of Level 3 upwards enter the market, a thorough safety assessment is required, since the driving responsibility transfers from the human driver to the automation [5]. Therefore, an AV should drive at least as safe as a human driver, or preferably even safer [6]. Investigating the reaction of media and society on rare accidents of prototype AVs on public roads gives the impression that society's expectations of an AV are significantly higher than of itself. The automotive industry and academia have recognized the relevance of this topic and have spent a considerable amount of resources. However, the safety assessment of AVs remains a huge challenge [7]. Proofing that an AV is at least as safe as a human driver with regards to fatal accident rates on highways requires billions of kilometers, since accidents are rare events [8]. For sure, this high mileage is not feasible via classical real-world testing. This is especially true if one considers that the proof is only valid for one software version of one vehicle variant. These challenges also put governments in a difficult spot, as regulatory requirements for type approval are lacking and ultimately cause an approval trap. It refers to the current situation where the first prototypes of Level 3 exist, but no approvals are available [5].

There are several safety assessment approaches currently addressed by the AV research community to solve the approval trap [9]. The most prominent is the Scenario-Based Approach (SBA), since it is frequently used in the literature and the core of large research projects such

as the German PEGASUS [10] or the European ENABLE-S3 [11]. It argues that no actions and events occur during most of the driving time [12]. Therefore, it concentrates on interesting traffic situations, mainly by means of computer simulation. Both the restriction of traffic situations and the use of simulation tackle the efficiency and scalability of the safety assessment. These factors become more and more important with higher automation levels [13]. Nevertheless, even if not mandatory, it is still helpful to apply new safeguarding methods using simulation already to simpler ADAS to increase their safety. There are further safeguarding approaches using, for example, traffic simulations [14], standards [15], or formal methods [16]. Most of them have in common that they use mathematical models.

However, the model-based safety assessment approaches are currently rarely accompanied by model validation activities [9]. The British statistician George Box stated once that “all models are wrong, but some are useful” [17]. This reflects the fact that a model is per definition a simplified abstraction of reality and is developed with a certain use case in mind. Therefore, it is the task of model validation to assess the quality of simulation models in comparison to physical experiments as ground truth [18]. This is of enormous relevance, in particular for such safety-critical systems as AVs. Without any knowledge about the validity of simulation models, the model-based statements about the safety of AVs are of limited value at best, heavily misleading at worst. Imagine the reaction of society if the AV causes an accident after having passed all test scenarios in simulation. This shows that model validation is indispensable for reliable model-based safeguarding and the latter in turn for a safe market introduction of AVs. Thus, the core of this thesis is the quality assessment of simulation models for AV safeguarding.

1.2 Research Objectives

The insight into the current safeguarding of AVs shows that simulation is the central hope for solving the approval trap, but the simulation itself brings new challenges. Therefore, the main objective of the present work is the

O1) **Development and application of an overall framework that covers the quality assessment of the simulation models on the one hand, in order to enable the actual safety assessment of AVs on the other hand.**

On closer examination, individual pillars can be identified, each of which represents an essential contribution to achieving the overall objective. These represent early goals not yet based on a deep literature research:

O1.1) **Development of taxonomies for classification of the state of the art:**

Since models have a long-standing history and are used in numerous application fields, there exists a myriad of references on model validation. However, since the validation methods have developed in different communities, they have historically diverged greatly over the decades [19]. This is currently reflected in a heterogeneous research landscape that lacks a uniform understanding and procedure. The terms and definitions of the individual communities already differ in what model validation is [18, Sec. 2.1.2] and thus represent a good indicator of the heterogeneity. Therefore, a taxonomy that provides a uniform classification scheme is needed to structure the references. Even though the literature on safeguarding is still young, a similar effect and demand is evident because research shows a dynamic behavior due to the large input of resources.

O1.2) **Extension of the taxonomies to an overall framework:**

However, one taxonomy for model validation and one for safeguarding constitutes only the first pillar. The individual classes of both taxonomies have to be combined into an overall framework that offers several options for the user. The framework shall be designed in a modular, uniform, and generic way so that users from different communities can benefit from it. Although safeguarding AVs is the central application in this thesis, the framework shall consider it a preferred configuration, but still be flexible enough to cover further ones. The framework must connect the individual classes or blocks to a continuous process with several steps. They range from selecting test scenarios to quantifying modeling errors and to making decisions about the safety of the system.

O1.3) **Validation of the framework itself through simulative preliminary studies:**

In other research fields such as object recognition, it is natural to apply new methods to a defined data set to evaluate their recognition rate against comparable approaches [20]. In contrast, new test and validation procedures are often only demonstrated by a simple Proof of Concept (PoC), which states, for example, that a system is safe or that a model is valid. However, it is difficult to judge whether the system is actually safe or whether there would have been alternative test methods that would have revealed safety gaps. Therefore, this work shall not only aim to use the framework for the validation of a model, but also to validate the framework itself. On the one hand, the configuration procedure of the framework for its specific use case shall be as sound as possible based on the current state of the art. On the other hand, actual ground truth data shall be used to validate the framework results. Since physical validation experiments require enormous resources and are limited in their scope, the validation of the framework itself shall be based on extensive preliminary studies using dedicated simulations.

O1.4) **First application of the framework for model-based type approval of AVs:**

After configuration and validation of the framework, it shall be applied to the actual simulation-based safety assessment of an AV involving physical validation experiments, re-simulations, and new predictive simulations. Selecting a use case from the type approval of AVs is a promising example, since it has a neutral, public, and standardized character, and it is important for vehicle manufacturers, technical services, and regulators. This will make the findings relevant for a broad audience. The perspective of the technical service on type approval has shaped this work the most. Nevertheless, it shall serve as a representative PoC and blueprint for the general safety assessment of the manufacturer as well. The physical experiments always involve additional challenges such as noisy signals, large measurement files, or natural variability. Therefore, the real PoC shall demonstrate how the framework can take these effects into account.

1.3 Structure of the Thesis

Figure 1.1 illustrates the structure of this work and assigns the four research objectives to the respective chapters. This chapter gave a short introduction into the safety assessment of AVs, the relevance of simulation in it, and the objectives of this thesis to overcome the challenges of model validation. Chapter 2 examines the current state of the art. It provides a compact overview of safety assessment approaches including a taxonomy according to the first research

objective. It continues the overview with the type approval, as well as with model validation theory and its application across several engineering fields. It concludes with the criticism of the current state of the art and the derivation of research gaps and questions. Chapter 3 develops a novel validation framework according to the second research objective. It first describes it theoretically in its generic form and then derives a specific configuration of the framework for AV type approval by systematically selecting suitable methods. Chapter 4 validates the framework in a simulative preliminary study according to the third research objective. It makes a final method selection based on the study results and takes this validated framework configuration to apply it to a real PoC. The latter includes physical experiments and simulations for the type approval of the lane keeping behavior of a vehicle. This use case lends itself to the fourth research objective. Chapter 5 discusses this work and presents future improvements for current limitations. It checks whether all research objectives, research questions, and requirements are fulfilled. Finally, Chapter 6 summarizes this work.

1.4 Publications

Major parts of this dissertation have already been published in the four peer-reviewed journal papers shown in Table 1.1. This assures in advance that the dissertation content has undergone a peer-review process and that it has been made available to the public. This section briefly discusses the relationship of the papers to this dissertation in order to clarify their role at the very beginning and to refer to further information.

Table 1.1: Overview about the main papers of this dissertation.

	Reference	Journal	Title
P1)	Riedmaier et al. [9]	IEEE Access 2020	Survey on Scenario-Based Safety Assessment of Automated Vehicles
P2)	Riedmaier et al. [21]	Springer ACME 2021	Unified Framework and Survey for Model Verification, Validation and Uncertainty Quantification
P3)	Riedmaier et al. [22]	Elsevier SIMPAT 2021	Non-deterministic model validation methodology for simulation-based safety assessment of automated vehicles
P4)	Riedmaier et al. [23]	MDPI Applied Sciences 2021	Model Validation and Scenario Selection for Virtual-Based Homologation of Automated Vehicles

These papers were aligned along the entire scientific structure of this dissertation from the state of the art to the methodology to the results and their discussion. Figure 1.1 contains a mapping between papers and thesis sections that is made based on the main paper content. Nevertheless, there can be a small overlap with other sections. We dedicate a separate chapter in the appendix for more detailed information. The list in Chapter A.1 includes a short summary of each paper, classifies them into this thesis, quotes the author contributions, and quotes the copyright statements. Table A.1 in Chapter A.2 presents a mapping between paper sections and thesis sections at the lowest numbered layer. These central overviews are convenient because of the strong dependence of this dissertation on previous publications. At designated points, reference will be made again to a respective paper to explicitly point the reader to additional information. However, if there is no more reference in a chapter of choice, the interested reader can return to Figure 1.1 or Table A.1 to extract the references to the author‘s publications.

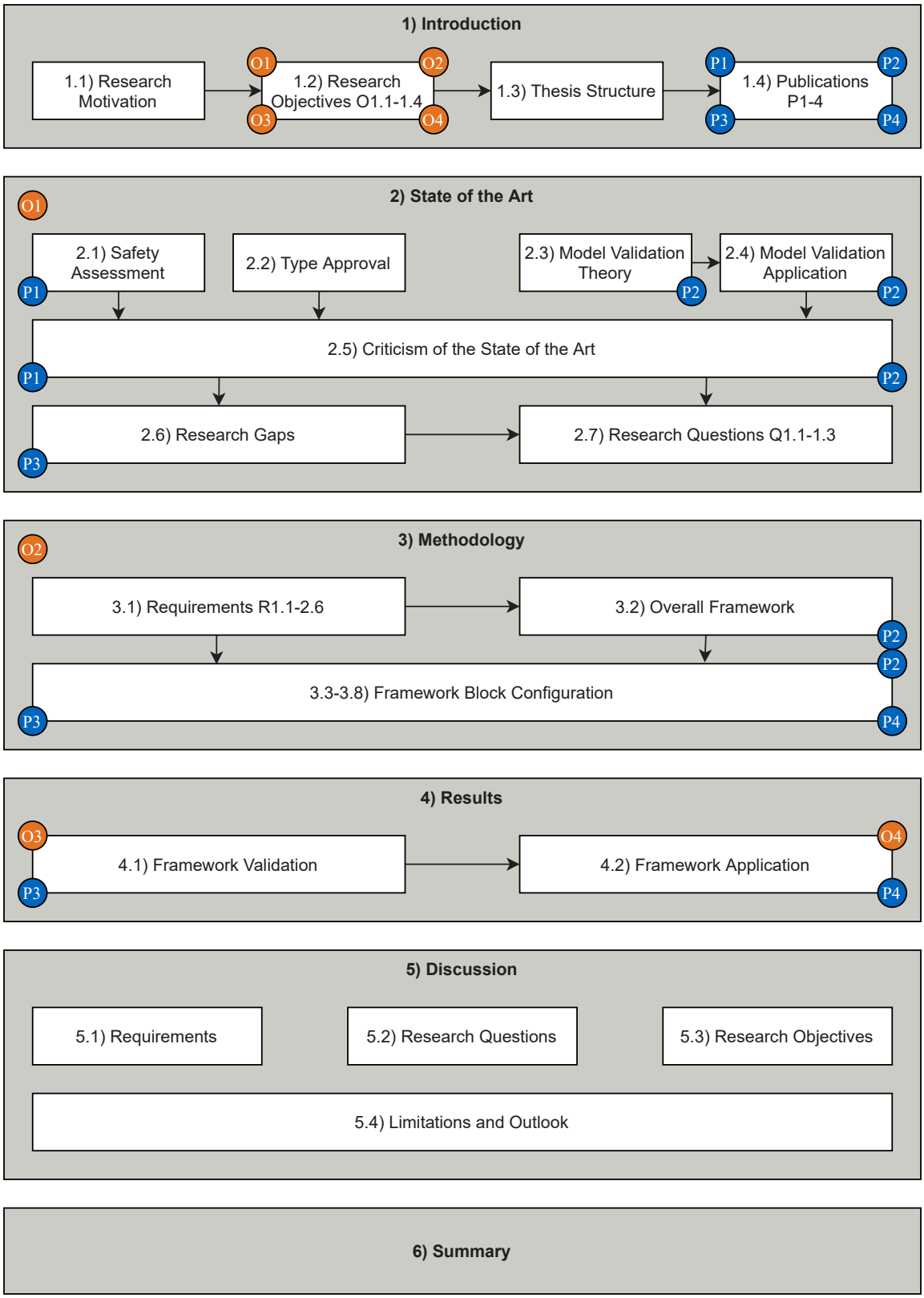


Figure 1.1: Structure of the thesis with assignment of research objectives and papers.

The role of this dissertation is to describe a coherent thread through the papers with additional insights into the scientific process. On the one hand, there are dissertation sections that are based on a related section of a publication. They can be a compact summary of the related section, a revised and rewritten form, or an extended version with additional information. The decision in each case depended on which was better suited to telling the reader a coherent story. On the other hand, there are entirely new sections. They sometimes include additional content or illustrations, but focus mainly on a systematic reappraisal of key procedures from the scientific process, which had to be omitted from the publications for reasons of space. This starts with a comprehensive research motivation and research objectives to work out the problem statement, goals, and relevance of the topic. It continues with a systematic derivation of research questions from the criticism of the state of the art and in turn with requirements for the methodology. It addresses the selection procedure within each framework block to record why the framework was configured in a certain way for our use case. It concludes with a comprehensive discussion of the results and referring back to the requirements, research questions, and research objectives at the end to determine whether they are fulfilled. Finally, it contains an outlook with major directions for future improvements.

In addition, there are journal and conference papers by the author of this dissertation that are related to the dissertation topic, but are not a direct part of it. They are given in Table 1.2. The early paper [24] focuses more on implementation. The author supported Danquah et al. [25–27] as the second author of three other peer-reviewed publications. These papers pick up methods of this dissertation, but apply them to the vehicle parameters in consumption simulations instead of the scenario conditions in AV type approval. The list in Chapter A.3 includes, in turn, a short summary, thesis classification, and author contribution.

Table 1.2: Overview about papers that are not a direct part of this thesis but are related in content.

	Reference	Journal/Conference	Title
P5)	Riedmaier et al. [24]	GSVF 2018	Validation of X-in-the-Loop Approaches for Virtual Homologation of Automated Driving Functions
P6)	Danquah et al. [25]	Taylor & Francis VSD 2020	Potential of statistical model verification, validation and uncertainty quantification in automotive vehicle dynamics simulations: a review
P7)	Danquah et al. [26]	Elsevier Procedia CIRP 2020	Statistical Model Verification and Validation Concept in Automotive Vehicle Design
P8)	Danquah et al. [27]	MDPI Applied Sciences 2021	Statistical Validation Framework for Automotive Vehicle Simulations using Uncertainty Learning

2 State of the Art

This chapter is divided into three groups of sections. The first section pair deals with the safety assessment of AVs, first in general and then in the specific context of type approval. The second pair deals with model validation, first by introducing general principles and then by giving an overview about several engineering fields. The first two pairs neutrally describe the related work, while the third group analyses it to derive the research gaps and questions of this thesis. This chapter contains the state of the art that is either important for the derivation of research gaps or the fundamental understanding of the work. In contrast, literature that targets the specific content of an individual thesis section will be introduced later directly before the respective section.

2.1 Safety Assessment of Automated Vehicles

This section starts with common terms and definitions for safety assessment of AVs. It continues with a survey of safety assessment approaches and dedicates a separate section to the SBA.

2.1.1 Terms and Definitions

In order to avoid misunderstandings caused by a different language, this subsection introduces common terms and definitions in the field of safeguarding:

Automation Levels: The standardization body SAE initially released the standard J3016 [4] in 2014 including a taxonomy and definitions for six levels of driving automation. Starting from classical human driving at Level 0, an ADAS assists the driver with either longitudinal or lateral control at Level 1, or with both in parallel as Partial Automation at Level 2. This is the range of production vehicles currently seen on public roads. There is a large gap to Level 3 and higher, since the responsibility transitions from the human driver to the vehicle. Conditional Automation at Level 3, High Automation at Level 4, and Full Automation at Level 5 differ in the fallback to the human driver and increasing operating conditions.

AV: Strictly speaking, an AV refers to the Automation Level 3 and higher [4]. From the perspective of model validation in this thesis, higher levels require more tests in simulation and higher accuracy requirements, as the implications become larger. This motivates a more rigorous configuration of the model validation methodology, but it does not change the methodology per se that will be developed in this thesis. The generic methodology will equally affect all levels, which is why we will often use the term AV as a placeholder for all levels.

Scenario: The term scenario may sound similar to scene, situation, or test case, but there are subtle differences. According to [12], a scene characterizes the environment in only one time step. A situation is a limited representation of a scene from a certain perspective. In contrast, a scenario describes the entire chronology of several sequential scenes. The

scenario transitions from the initial scene due to actions, events, aims, and objectives of the individual traffic participants [12]. A test case goes beyond a scenario by adding pass/fail criteria to the test scenarios [28]. In addition, a scenario can be further categorized into functional, logical, and concrete scenarios [29]. The former describes a scenario such as car following or cut-in in form of a text description. A logical scenario specifies parameter ranges and possibly parameter distributions based on real-world exposure. Finally, a concrete scenario represents a specific parameter combination within the scenario space.

6-Layer Environment Model: The environment as scenario input was initially structured by means of five layers [30] and extended afterwards to six layers [31]. Layer 1 describes the static road topology, Layer 2 the traffic infrastructure such as speed limits, and Layer 3 temporary manipulations such as construction sites. Layer 4 represents dynamics traffic objects and Layer 5 varying weather conditions. Layer 6 is an extension to Vehicle-to-X communication. Beyond the 5-level and 6-level environment model, the driving behavior of the AV is affected by interval vehicle states and possible human driver inputs depending on the automation level.

Operational Design Domain (ODD): The ODD restricts the operating conditions of the AV to its intended scope specified by the vehicle manufacturer [4]. Possible restrictions are the layers from the environment model such as road classes or weather conditions.

Safety Assessment: There are several terms in the AV community to refer to the safety aspect such as safeguarding, safety assessment, verification, or validation. This thesis uses the first two interchangeably, but excludes the terms verification and validation from systems and software engineering to avoid confusion with the corresponding modeling terms. A microscopic assessment focuses on safety within individual scenarios, whereas a macroscopic assessment aims to make an overall safety statement about a large amount of scenarios [32]. The type approval does also address the AV safety, but it is only a subset of the overall safety assessment that is required in the last step to get the regulatory permission for market introduction. Nevertheless, both are similar from the perspective of model validation. We should always keep in mind when reading the two terms in this thesis that the type approval serves us as a blueprint for safeguarding in general and, conversely, the overall safeguarding contains the type approval.

Key Performance Indicator (KPI): For an efficient safety assessment, it is important to post-process the results by means of KPIs, characteristic values, or in this context by criticality metrics [32, 33]. The minimum time-to-collision from the AV to other traffic vehicles is an illustrative example. These KPIs are the analogon to the scenario parameters on the input side. They are associated with pass/fail criteria specifying the permissible thresholds such as a minimum time of one second.

2.1.2 Overview of Safety Assessment Approaches

There are various approaches for assessing the Safety Of The Intended Functionality [34]. The following list is a compact summary, except for the SBA, to which a separate section is dedicated in more detail. White papers [35] and standards [36] such as ISO 26262 [37] focusing on functional safety or UL 4600 [15] focusing on safety cases are out of scope of this thesis.

Real-World Testing: Classical vehicles and ADAS are currently released after real world tests. Their mileage is extensive but still feasible, since the driver is available as a fallback level.

Therefore, the focus is mainly on avoiding false-positive interventions so that, for example, a braking assist does not brake without reason. However, this limits the testing scope of ADAS, and the mileage rises with higher automation levels to impracticable regions [5, 8].

Function-Based Approach: In function-based testing, the functionality of a system is compared with requirement specifications based on proving ground tests or simulations [38]. It is currently used for ADAS, since they have clear specifications in a restricted ODD. However, with higher automation levels it is getting more and more difficult to specify the requirements and to link them to pre-defined test scenarios.

Formal Verification: Formal verification is a mathematical technique to guarantee that a system satisfies its requirements. In contrast to testing, it does not select individual test cases, but it creates a proof for the entire scenario space [39]. Several verification methods have been developed for AVs. The first category is based on theorem proving, which uses a formal model consisting of axioms and lemmas, as well as techniques such as induction to verify the system safety. This includes the well-known Responsibility-Sensitive Safety approach [6] and further ones such as [40–43]. The second category uses reachable set calculations. The idea is to determine the states a system can reach given initial states and possible inputs and parameters. It can guarantee safety online during run-time if the reachable set of the AV does not intersect the predicted ones of the other traffic participants. A reachable set is usually over-approximated by means of non-deterministic models with set-based uncertainties. Reachability analysis is addressed in the European research project UnCoVerCPS [44] and several publications such as [16, 45–48]. Since it is computationally expensive, it is possible to speed up the process by a robustness-guided verification that uses a simulation-based optimization upstream to localize interesting areas for verification [49, 50]. The third category automatically synthesizes correct-by-construction controllers from formal specifications [51–53]. There are several languages available to formalize traffic rules [54–56] such as Signal Temporal Logic [57].

Shadow Mode: An interesting and safe idea to test new software systems is the open-loop integration in the background of the driver without actual intervention [58]. This is sometimes called shadow mode or Trojan horse. It requires simulation to test a trajectory planner in shadow mode as opposed to only the perception system, since the control loop must be closed in the virtual world. However, this requires model validation, and the other road users do not see and react to the AV behavior in the virtual world. Nevertheless, it provides knowledge, albeit limited, and car manufacturers like Tesla [59] use this approach.

Staged Introduction: Since the introduction of AVs from Level 3 upwards involves risks and enormous resources, it is a promising way to start with a small ODD and safety drivers, as well as to successively expand them and discard the drivers. Vehicle manufacturers and suppliers such as Daimler and Bosch use this procedure on defined road sections [60].

Traffic-Simulation-Based Approach: Most of the safety assessment approaches currently focus on making a microscopical statement on individual scenarios from the perspective of a single AV. In contrast, traffic simulations with several agents [61] are capable of making macroscopic statements about the impact of AVs on traffic. They can model complex interactions and analyze the accident impact of certain factors such as the ratio of AVs and human drivers [62] or the failure of components [63]. Some of these approaches stem from the effectiveness analysis of ADAS [64].

2.1.3 Scenario-Based Approach

In recent years, a large amount of literature has been created with the aim of selecting representative scenarios to enable effective safeguarding [65, 66]. The individual approaches cover a wide range of methods and focus on different aspects within the SBA. The high dynamics in this field of research testifies to the relevance of the topic, but for the observer it seems disorderly and makes comparability difficult. Therefore, a taxonomy was developed in the author's previous publication [9] to classify 183 references into the blocks of the framework in Figure 2.1. In order to enable a clear assignment, the framework builds on the current state of the art and large research projects such as PEGASUS [10]. The assignment can be ambiguous for some references if they address edge topics or extend over several blocks. Therefore, the taxonomy aims to identify the central topic of each reference and highlights further edge cases. The framework blocks and pillars are structured based on the process of safeguarding. The scenario database is the central pillar separating scenario methods that fill the database from methods that take scenarios out of the database for subsequent execution and assessment. The following paragraphs give an overview about each of the six pillars with a certain focus on the knowledge-based, data-driven, testing-based, and falsification-based scenario methods, since they will be taken up later in Chapter 3.3.1. Only selected references are shown in the text to illustrate the principle for the fundamental understanding of this work. Further references are summarized in the respective block diagrams.

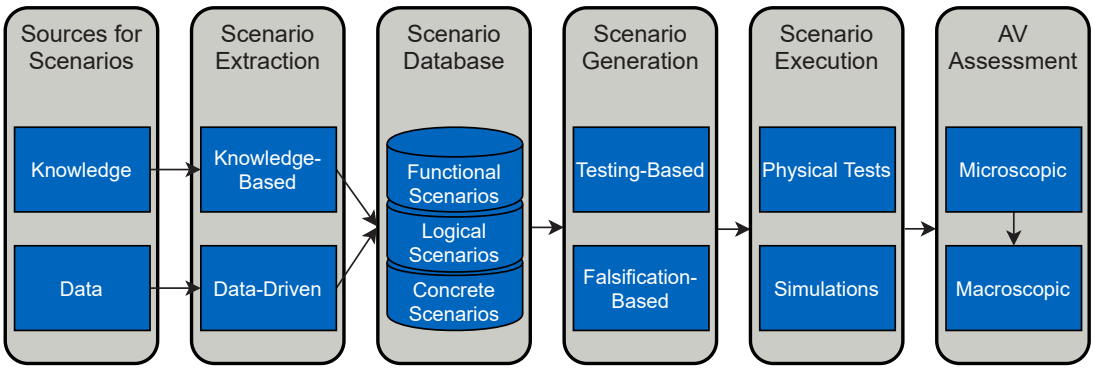


Figure 2.1: Framework of the SBA based on [9, Fig. 2].

Sources for Scenarios

There are two categories of scenario sources available: abstract knowledge on the one hand and data on the other. Typical knowledge comes from experts, standards, or guidelines such as the German guidelines for the construction of highways [67]. Driving data are usually collected with fleet vehicles during Field Operational Tests. Many car manufacturers have their own private data sets. Nevertheless, several organizations have recently published data sets [68–70]. An important factor is the measurement equipment used to collect the data. Depending on the application, the vehicle's internal sensors may be sufficient, an extended setup can be mounted on the vehicle roof, or external sensors can be positioned on the infrastructure or a drone [71].

Knowledge-Based Scenario Extraction

In analogy to the distinction of knowledge and data as scenario sources, the taxonomy also distinguishes between knowledge-based and data-driven approaches to fill the database. They are summarized under the term scenario extraction from the highlighted perspective of the sources. The knowledge-based methods often represent the described knowledge in the form

of ontologies [72]. They structure and characterize knowledge by means of properties and relationships. They automatically derive scenarios by combining the knowledge and ensure those are valid thanks to the modeled relationships. It is both possible to derive functional, logical, and concrete scenarios as shown in Figure 2.2a. For example, Bagschik et al. [30] represent all layers of their environment model in the form of an ontology.

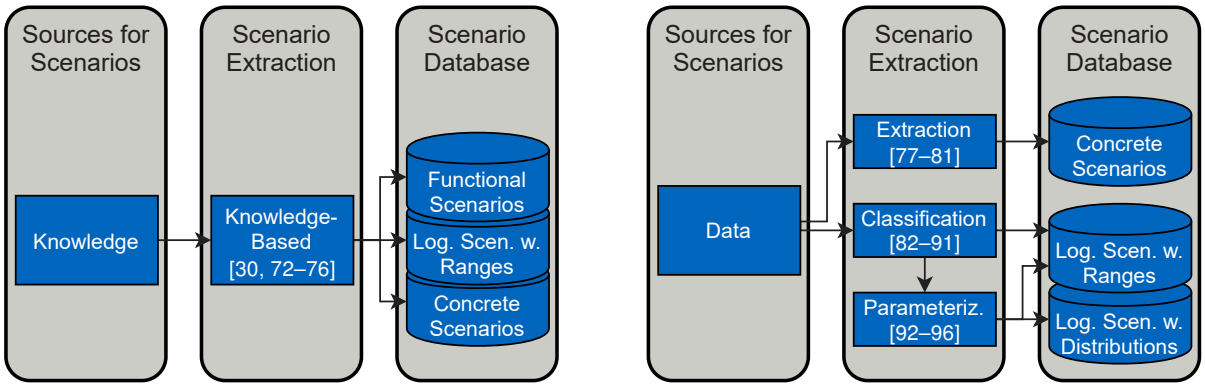


Figure 2.2: Scenario extraction methods including literature based on [9, Fig. 3-4].

Data-Driven Scenario Extraction

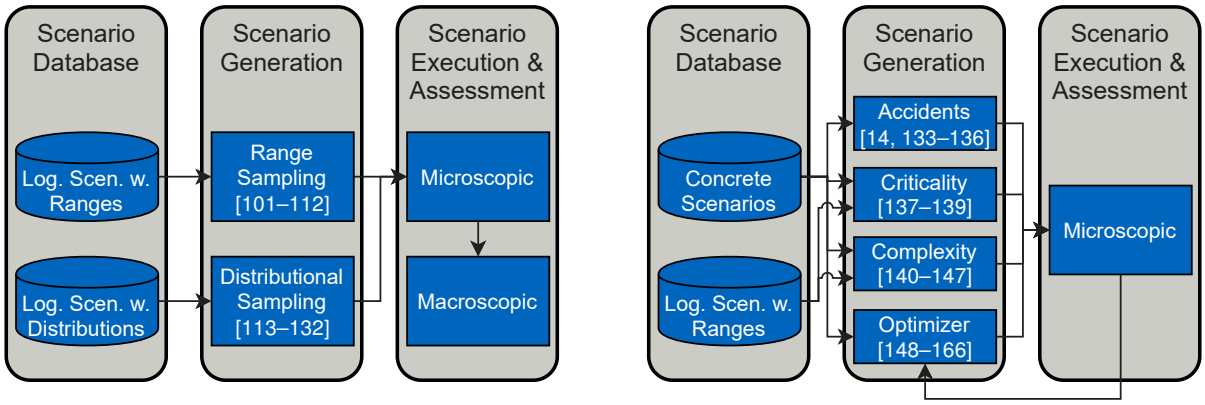
The data-driven methods apply machine learning and pattern recognition techniques to extract interesting scenarios. One subcategory directly extracts concrete scenarios, for example, by detecting high novelty values compared to already seen data [81]. The second subcategory of methods first groups the data. They are using supervised learning methods that classify the data into pre-defined scenario classes [83] or unsupervised learning methods that cluster the data based on similarity [89]. Erdogan et al. [85] perform a comparison and get the best results with supervised learning. The grouped data can be further processed by the publications of the third subcategory. They intend to parameterize the scenario space and describe the scalar parameters of each group and logical scenario either via ranges of minimum and maximum values or via probability distributions [93].

Scenario Database

The scenario database is the central element of the SBA. Its primary goal is to specify a standardized interface for reading different data sources and to process them into a machine-readable format. The PEGASUS project [10] developed a scenario database for the ODD highway [97]. It forms the basis for further enhancements such as [98, 99]. Althoff et al. [100] introduce the Commonroad framework that combines the scenario database with the corresponding models and cost functions to fully reproduce the virtual assessment of trajectory planners.

Testing-Based Scenario Generation

The taxonomy classifies publications that focus on the generation of concrete scenarios into the scenario generation pillar of the framework. As highlighted in Figure 2.3, it offers both testing-based and falsification-based approaches. The former aim at a good coverage of the entire scenario space in order to be able to make a fair statement about the safety of the AV. Within the former, the taxonomy distinguishes between methods that take logical scenarios with parameter ranges or distributions from the database.



(a) Testing-based (b) Falsification-based

Figure 2.3: Scenario generation methods including literature based on [9, Fig. 5-6].

Exemplary algorithms that have been used to generate samples within parameter ranges are Design of Experiments techniques [102] or Rapidly Exporing Random Trees [16]. Distributional sampling techniques are based on the classical Monte Carlo sampling. However, they are inefficient because accidents are rare events. Therefore, many publications accelerate the process by using Extreme Value Theory [117] or Importance Sampling [130]. Both sampling types are similar in regards to a microscopic assessment of individual scenarios. Regarding a macroscopic assessment, however, the distributional sampling excels since it weights the scenarios with their occurrence probabilities from the real world.

Falsification-Based Scenario Generation

In contrast to testing-based approaches aiming at a good coverage of the entire space, the falsification-based approaches search for sub-spaces where the AV violates the safety requirements. They are particularly interesting for the system developer to find counterexamples within a short time. The downside is that they are only suitable for a microscopic assessment, but not for a fair macroscopic assessment about the totality of all scenarios. The taxonomy further divides the falsification-based literature according to the four blocks in Figure 2.3b. Concrete scenarios from accident databases are reasonable candidates to falsify the AV behavior, since they already led to accidents in the past [136]. However, the informative value is limited, as they concentrate exclusively on accidents that can be avoided, but not on new ones caused by the AV itself. Another subcategory of approaches specifies criticality metrics to detect dangerous scenarios. Klischat and Althoff [138] calculate the free area for the AV and arrange the scenarios so that the area is minimal. Similar publications define a measure of scenario complexity to generate complex scenarios that will probably lead to critical situations in the future [142, 167]. Finally, there are methods that do not only optimize the scenarios beforehand, but include the scenario execution and assessment into the optimization loop [154]. Therefore, the actual assessment results can be used to direct the subsequent scenarios into more and more critical areas.

Scenario Execution

The generated concrete scenarios can be assigned to several test environments [168]. The target environment is the real road, where the AV will drive after approval. Proving grounds are closest to it and allow the execution of physical tests in a comparatively safe manner. In return, there are a variety of X-in-the-Loop (XiL) environments with increasing degree of virtualization from Vehicle-in-the-Loop (ViL) to Hardware-in-the-Loop (HiL), Software-in-the-Loop (SiL), and

Model-in-the-Loop (MiL). The Vehicle-Hardware-in-the-Loop (VEHiL) environment constitutes a mixture of ViL and HiL, since the entire vehicle is mounted on a chassis dynamometer or powertrain test bed [24]. ViL, HiL, and VEHiL are hybrid environments, since they contain both real and virtual components. The virtual traffic environment can be injected into the physical sensors, for example, by positing a monitor in front of the camera. The XiL environments excel regarding costs, effort, and safety, but they get further and further away from reality. Nevertheless, almost all references use a certain type of simulation for their PoC, either commercial ones, free ones, or self-developed environments [169, 170].

AV Assessment

The assessment paragraph builds on the definitions given in Chapter 2.1.1. The microscopic assessment of individual scenarios uses criticality metrics such as the time-to-collision [171] to quantify safety. Further variants can be found in [172–174]. The macroscopic assessment is the motivation for large research projects such as PEGASUS to compare the AV with human drivers. However, most of the current references focus on the microscopic assessment. There are just a few publications [175–177] that target the transition to the macroscopic assessment.

2.2 Virtual-Based Homologation

The car manufacturers carry out a thorough safety assessment internally to ensure the safety of their product. In addition, they need official approval to release a vehicle onto the market. There are two main procedures to achieve this. On the one hand, nations such as the United States [178, 179] apply a self-certification procedure, where the car manufacturer performs the tests and creates the certificates on his own. The authority can purchase individual vehicles to carry out random checks on the requirements. On the other hand, nations such as the members of the European Union apply a type-approval procedure, where the authority certifies that the vehicle satisfies the administrative and technical requirements. It usually delegates the task to a technical service as independent organization. The type approval refers to a vehicle type as a group of vehicles that share specific properties. Homologation is the major procedure for type approval by showing the equality of many vehicles with one vehicle type. Both terms are used as synonyms in this thesis. There are a variety of regulations that address individual components of the vehicle ranging from the door latches and hinges to functions of an AV. The following subsections address three relevant regulations for the further course of this thesis.

2.2.1 Regulation 140: Electronic Stability Control

Regulation 140 [180], formerly R-13H, addressing stability control systems is the first major example for a virtual-based homologation process [181]. Its relevance for this thesis is based rather on its process than its vehicle dynamics requirements. The following citation contains all regulatory statements relating to simulation and skips the irrelevant passages in between:

> *Where a vehicle has been physically tested in accordance with [...], the compliance of versions or variants of that same vehicle type may be demonstrated by a computer simulation [...]. The simulation shall be carried out with a validated modelling and simulation tool and using the dynamic manoeuvres [...]. A typical model may include the following vehicle parameters [...]. The Vehicle Stability Function shall be added to*

> *the simulation model by means of: (a) A subsystem (software model) of the simulation tool; or (b) The electronic control box in a hardware-in-the-loop configuration. [...] The validity of the applied modelling and simulation tool shall be verified by means of comparisons with practical vehicle tests. The tests utilised for the validation shall be the dynamic manoeuvres [...]. During the tests, the following motion variables, as appropriate, shall be recorded [...]. The simulator shall be deemed to be validated when its output is comparable to the practical test results produced by a given vehicle type during the dynamic manoeuvres [...].* [180, Sec. 7, Annex 3, Annex 4].

In summary, the manufacturer can use the simulation after demonstrating its validity in comparison to physical tests. The regulation allows the controller design as SiL or HiL and gives recommendations on which vehicle parameters, motion variables, and dynamic maneuvers to use. The maneuvers for model validation are the same as they are subsequently used for the actual type approval. However, the regulation defines neither a validation methodology nor the permissible tolerances for the comparison. Since it leaves a lot of room for interpretation in this respect, ISO developed the two standards ISO 19364 [182] and ISO 19365 [183]. They decompose the model validation into the validation of the pure vehicle dynamics model, referred to as passive vehicle model or open-loop model here, and the validation of the overall vehicle model including stability controller, referred to as active vehicle model or closed-loop model. The idea is to first validate the component model before increasing the scope to the system-level. Concatenating the individual tasks yields the virtual-based multi-stage process illustrated in Figure 2.4. The synonymous terms virtual-based, model-based, and simulation-based indicate that the process focuses on virtual tests based on a few real ones. According to this definition, a virtual-based homologation contains both the model validation and the actual virtual homologation as its last process step.

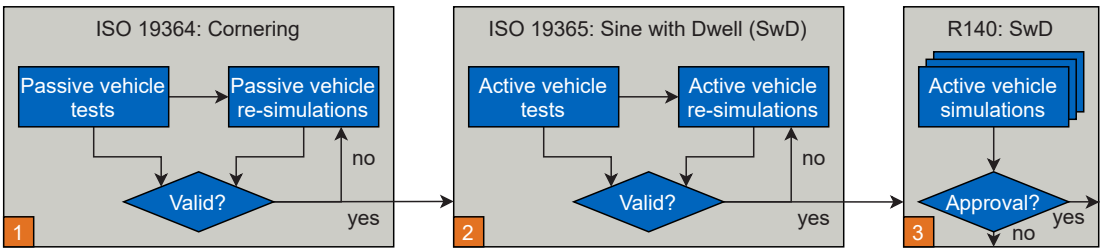
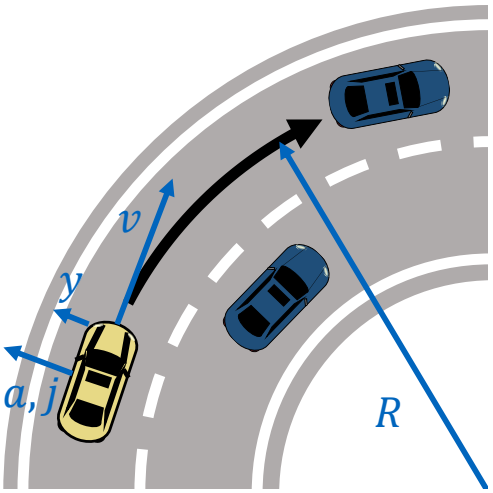


Figure 2.4: Virtual-based homologation process of stability control systems. It consists of a passive and active vehicle model validation followed by the actual virtual type approval. If all approvals tests are passed (final yes-arrow), the vehicle can be sold on the market. Otherwise (final no-arrow), the manufacturer must make internal improvements to the vehicle.

Whereas the passive vehicle model validation in the first step is based on a steady-state cornering maneuver, the active vehicle model validation in the second step uses the dynamic sine-with-dwell maneuver from the type approval in the third step. Each of the three steps results in a binary decision. The passive and active vehicle model validation assess whether the respective models are valid by comparing their deviations from reality with permissible tolerances. This approach will be described in Chapter 2.4.1. The virtual homologation checks whether the vehicle satisfies the requirements of the dynamic maneuver and ultimately whether it is approved.

2.2.2 Regulation 79: Automatically Commanded Steering Function

The UNECE developed regulations for the approval of ADAS. Regulation 79 [184] in its fourth revision, briefly referred to as R-79, is a representative example that addresses the lane-keeping behavior of production vehicles of Level 1 and Level 2 (its lateral part). It is supplemented by further amendments that specify selected aspects such as signal filtering [185]. It distinguishes several categories of lane-keeping functions. The Automatically Commanded Steering Function of category B1 describes "a function which assists the driver in keeping the vehicle within the chosen lane, by influencing the lateral movement of the vehicle" [184, Sec. 2.3.4.1.2.]. R-79 specifies four types of tests to assess the safety of those vehicles. Three of these relate to characteristic limitations of an ADAS such as maximum lateral acceleration, maximum steering wheel force, or acoustic and visual warning signals for the driver. The Lane Keeping Functional Test, however, focuses on the intended lane-keeping behavior of the vehicle with the driver taken out of the loop, as it is equally relevant for higher automation levels. It is illustrated in Figure 2.5 with key facts about the scenario and pass/fail criteria.



Scenario description:

- "constant speed on a curved track with lane markings at each side"
- "necessary lateral acceleration to follow the curve shall be between 80 and 90 per cent of the maximum lateral acceleration specified by the vehicle manufacturer $a_{y,\mathrm{smax}}$"
- "vehicle speed shall remain in the range from v_{smin} up to v_{smax}"
- "whole lateral acceleration and speed range"

Pass/fail criteria:

- "vehicle does not cross any lane marking"
- "lateral jerk does not exceed $5\,\mathrm{m\,s^{-3}}$"

Figure 2.5: Lane-keeping scenario with illustration from [22] and quotes from [184].

The regulation focuses on stationary conditions of the two scenario parameters velocity and "lateral acceleration to follow the curve" [184, Annex 8, Sec. 3.2.1.1.]. The latter is not the actual lateral acceleration of the vehicle depending on its trajectory. It is characterized in advance by the curve and can be achieved by driving with constant velocity through a curve with a constant radius. It will be referred to as reference lateral acceleration in this thesis. Thus, the regulation addresses driving states of the vehicle and the road layer from the 6-layer environment model. It targets one band from $80\,\%$ to $90\,\%$ of $a_{y,\mathrm{smax}}$ with higher priority, since it lies close to the maximum lateral acceleration $a_{y,\mathrm{smax}}$ specified by the car manufacturer. Nevertheless, the regulation requires proof for the whole range of velocities and reference lateral accelerations. There are two pass/fail criteria for the vehicle to pass in each concrete scenario. The limitation of the lateral jerk is mainly a comfort criterion. From a safety perspective, it is essential that the vehicle can stay within its lane without crossing the lane markings.

2.2.3 Regulation 157: Automated Lane Keeping System

The UNECE further developed the Regulation 79 for Level 1 and 2 vehicles to the Regulation 157 [186] for Level 3 vehicles. The latter addresses an Automated Lane Keeping Systems that "keeps the vehicle within its lane for travelling speed of $60\,\mathrm{km\,h^{-1}}$ or less by controlling

the lateral and longitudinal movements of the vehicle for extended periods without the need for further driver input" [186, Sec. 2.1]. The commission recently accepted the new proposal, assigned it the number 157, and decided that it is binding from early 2021. Afterwards, it has to be transitioned to national law of the individual member countries of the UNECE that are responsible for performing the type approval. Since this regulation was subject to major changes and no vehicles were available during this work that already complied with it, it does not lend itself as a use case for this thesis. Nevertheless, it shows the future trend for higher automation levels. Thus, we do not look at the test descriptions and criteria, but take an important statement from it. Whereas R-79 does not explicitly allow computer simulation, R-157 allows it again:

> *Simulation tool and mathematical models for verification of the safety concept may be used [...], in particular for scenarios that are difficult on a test track or in real driving conditions. Manufacturers shall demonstrate the scope of the simulation tool, its validity for the scenario concerned as well as the validation performed for the simulation tool chain (correlation of the outcome with physical tests).* [186, Annex 4]

2.3 Model Validation Theory

This section introduces principles of model validation that are important for the fundamental understanding of this thesis. It starts with common terms and definitions for modeling and simulation and introduces several types of simulation models. It describes the nature of errors and uncertainties and their major sources that are inherent in every simulation and experiment. The following explanations build on book content from [18, 187, 188].

2.3.1 Terms and Definitions

In analogy to Chapter 2.1.1 on safety terms, this section introduces common terms and definitions in the field of Modeling & Simulation. It builds on Oberkampf and Roy [18, Chap. 2.1, 2.2, 3.2.2], who unify terms across several engineering communities:

Model: It refers to the "representation of a physical system or process intended to enhance our ability to understand, predict, or control its behavior" [18, p. 92]. We must always have in mind that a model is a simplified abstraction that focuses on selected system characteristics with a certain accuracy depending on its use case. Errors and uncertainties are inherent in every model by definition [17].

Simulation: It refers to the "exercise or use of a model to produce a result" [18, p. 92]. For computer models, the user normally accesses a simulation tool or tool chain that provides a solver, graphical user interface, and further functionalities to perform the simulation. It should be noted that the qualification of a tool chain [189] is not part of model validation and is beyond the scope of this thesis. There are a variety of XiL simulations that constitute hybrid environments with physical components. They are still referred to as simulation in this thesis, since they must likewise be compared against reality by means of validation methods, the test bed itself is also a model, just no computer model, and they are treated the same in type approval [180]. Nevertheless, the hybrid nature of a simulation is emphasized for understanding when necessary in this work.

Domain: This term is used in combination with the following four model-based activities to specify the current use of a model or system. For example, the validation domain indicates that certain tests are executed for the purpose of model validation, while the application domain indicates that they are executed for the actual model predictions. The latter is a more general term than the ODD from the safety definitions in Chapter 2.1.1. To highlight the concrete test conditions within a domain, we use the scenario terminology such as validation or application scenarios. Finally, we refer to the scenario space if we intend to emphasize the entire parameter range. The definitions are both in line with safety terms such as concrete scenarios and with the modeling terms [190, 18, Fig. 2.10].

Model verification: It refers to the "process of determining that a model implementation accurately represents the developer's conceptual description of the model" [18, p. 25]. For computer models, it assesses their numerical properties. Since numerical errors either arise from software bugs or from the solver, model verification can be further separated into code verification followed by solution verification. The former is "the process of determining that the numerical algorithms are correctly implemented in the computer code and of identifying errors in the software" [18, p. 32]. The latter is "the process of determining the correctness of the input data, the numerical accuracy of the solution obtained, and the correctness of the output data for a particular simulation" [18, p. 34].

Model calibration: It refers to the "process of adjusting physical modeling parameters in the computational model to improve agreement with experimental data" [18, p. 44]. In the strict sense, calibration targets inverse methods that optimize agreement of output quantities by iteratively adapting model parameters. In contrast, parameter estimation and parameter measurement directly measure the parameters without dependency on the outputs. The former, however, requires a mathematical relationship between the measured quantity and the parameter. Thus, it is recommended to use parameter measurements before resorting to parameter estimates and again before resorting to calibration [18, p. 45].

Model validation: It refers to the "process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model" [18, p. 25]. By definition, it contains a comparison between simulation and reality and refers to a certain use case of the model. Therefore, there is no generic model validity. In the strict sense of this definition, validation yields a degree of conformity and not only the two binary answers valid and invalid. We will sometimes use the term model validation in a wider sense as a representative of further model-based activities. For example, when we refer to our validation framework, the model validation in the strict sense is a key framework pillar, but the framework goes beyond it by incorporating new model predictions.

Model prediction: It refers to "interpolating or extrapolating the model beyond the specific conditions tested in the validation domain to the conditions of the intended use of the model" [18, p. 36]. According to this definition, model prediction does not cover all simulations but its subset of unseen application scenarios. The simulation of conditions observed during physical calibration and validation experiments is usually referred to as re-simulation.

Uncertainty Quantification (UQ): It refers to the "process of identifying, characterizing, and quantifying those factors in an analysis that could affect the accuracy of the computational results" [18, p. 14]. This should not be confused with a sensitivity analysis. The latter is the "process of determining how the simulation results, i.e., the outputs, depend on all of the factors that make up the model" [18, p. 15]. Nevertheless, there is a relationship between

the two. If a parameter has a low sensitivity and hardly affects the simulation results, it has little influence on their accuracy. For individual sources of uncertainty, such as the model inputs, the term UQ can be further refined as Input UQ. A subsequent uncertainty propagation takes the input uncertainties and propagates them through the model to derive the corresponding output uncertainties. The combination of both is called Uncertainty Quantification & Propagation.

Verification & Validation (V&V): The two tasks of model verification and model validation are jointly abbreviated as V&V to summarize the traditional assessment of simulation models. Whereas verification focuses on the correct implementation of the model, validation focuses on the correct behavior of the simulation model compared to the real world.

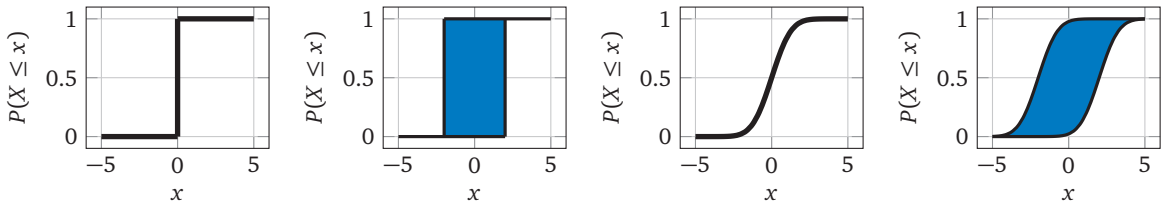
Verification, Validation & Uncertainty Quantification (VV&UQ): Adding the task of uncertainty quantification to the V&V definition yields the acronym VV&UQ. It is used to emphasize the modern assessment of simulation models including their uncertainties and will be described in more detail in the further course of this thesis. If a joint term for V&V and VV&UQ approaches is needed, we use the more general term VV&UQ.

2.3.2 Error and Uncertainty Types

Since a model is a simplified representation of reality, it contains errors by definition. However, it is challenging to precisely quantify those errors. The deterministic errors are represented in form of scalar point values. If this cannot be achieved, they must be replaced by non-deterministic uncertainties. Whereas this section distinguishes errors and uncertainties by their type, the subsequent one distinguishes them by their sources. Both subsections are a collection of fundamental knowledge from literature references such as [18, 187, 191, 192].

There are two types of uncertainties, which differ in their nature. If a phenomenon contains natural variability and stochastic effects, aleatory uncertainties arise. They are represented as probability distributions, either as Probability Density Functions (PDFs) or Cumulative Distribution Functions (CDFs). If knowledge about the phenomenon is missing, epistemic uncertainties arise. In contrast to aleatory uncertainties that can only be quantified, epistemic uncertainties can either be quantified or reduced by gaining more knowledge about the phenomenon. The remaining epistemic uncertainties are usually quantified by intervals, since this reflects the fact that no knowledge is available within the interval boundaries. Thus, there is a significant difference between aleatory uncertainties with uniform probability and epistemic uncertainties in form of intervals. Knowing, that all values within the borders are equally likely, yields quite an advantage over knowing nothing in between. Besides deterministic errors and aleatory and epistemic uncertainties, there is a fourth category of mixed uncertainties. They contain both aleatory and epistemic uncertainties at the same time. They are either described as a family of probability distributions or as imprecise probabilities. The probability box (p-box) is an imprecise probability [193, Chap. 4.6.4] that combines probabilities with intervals by extending a CDF to a box with an interval width. It still contains the aleatory uncertainty in the CDF form of the p-box edges, but also the missing epistemic knowledge via the box that encloses an infinite amount of possible CDFs [194, p. 13]. Figure 2.6 illustrates the mathematical structures of all four categories. A CDF and p-box with its two CDF edges can either be continuous and smooth, or discrete and empirical with several steps. While the smooth versions are illustrated here, we will work with the empirical versions later, as actual tests are always limited in their number. In the following, we will often refrain from mentioning the whole term errors and uncertainties and use one of the

two as representative for the sake of brevity. Nevertheless, we should keep in mind that both transition into each other with rising or falling precision.



(a) Deterministic point (b) Epistemic interval (c) Aleatory CDF (d) Mixed p-box

Figure 2.6: Mathematical structures to describe deterministic errors as well as epistemic, aleatory, and mixed uncertainties based on [195, Fig. 2, 21, Fig. 2]. All structures are visualized as cumulative probabilities $P(X \leq x)$ for consistency. The point value can be seen as a degenerate probability and the interval as a degenerate p-box. The horizontal lines are drawn once here for correctness, but omitted for simplicity as we proceed.

2.3.3 Error and Uncertainty Sources

Besides the distinction of errors and uncertainties according to their nature, there are various sources of errors and uncertainties. They occur both when comparing physical experiments g_s and simulations g_m to the true value of nature [187, eq. (1-5-6)]

$$y_{\text{true}} = g_s(x) - e_{y,\text{obs}} \tag{2.1}$$
$$= g_m(x, \theta, h) - (e_m + e_x + e_\theta + e_h)\,. \tag{2.2}$$

The individual error sources with their respective symbols will be introduced and explained step-by-step in the following paragraphs. Model calibration and validation are affected by all error sources, since they perform a comparison between the experiment in Equation (2.1) and the (re-)simulation in Equation (2.2). In contrast, only the modeling errors of Equation (2.2) affect a new model prediction. It is important to highlight again that these equations hold true if the errors can be quantified precisely. This assumption is usually made in traditional V&V. Otherwise, variances replace the deterministic errors to reflect the arising uncertainties. The law of total variance decomposes the variance [196, eq. (3, 4)]

$$\text{Var}[Y_s] = \text{Var}[\mathbb{E}[Y_s \mid X]] + \mathbb{E}[\text{Var}[Y_s \mid X]] \tag{2.3}$$
$$= \text{Var}[Y_m + \mathbb{E}[E]] + \mathbb{E}[\text{Var}[E]] \tag{2.4}$$

of the random variable Y_s of the experimental result y_s into two summands. Whereas the first one includes the variance of the random variable Y_m of the simulation result y_m, the second one includes the variance of the random variable E of the total error e. $\text{Var}[\cdot]$ represents the variance, $\mathbb{E}[\cdot]$ the expected value, and $\mid$ conditioning. Therefore, when considering uncertainties, it is no longer sufficient to concentrate exclusively on the error term, as is the case with traditional V&V. It has to be combined with UQ as overall VV&UQ framework in order to quantify the first summand and to not under-approximate the total prediction uncertainty.

Numerical Errors e_h – Model Verification

The first source of errors and uncertainties lies in the computational aspect of a model. The developers of a simulation tool have to implement the mathematical models in form of computer code. During this process, coding errors might occur. Therefore, code verification and software testing

activities are crucial to identify and fix these bugs. Even if this is fully achieved, numerical errors remain due to the nature of a computer. The simulation tool saves model inputs, parameters, and outputs in variables with a data type of finite precision leading to rounding errors. The solver uses a discrete step size h leading to discretization errors e_h. Various solution verification techniques such as Richardson extrapolation [192] quantify them. The numerical errors — symbolized by e_h as representative — are of different importance for different communities. In traditional numerical fields such as Computational Fluid Dynamics, they are of crucial importance due to complex calculations. In the automotive field or more general in systems simulation, however, they are often negligible [197].

Input and Parameter Errors e_x and e_θ – Input Uncertainty Quantification

The second source of errors and uncertainties lies in the model inputs and parameters. At this point, we distinguish between re-simulations and model predictions. For re-simulations and ultimately for model validation, the aim is to measure and use the conditions of the physical experiments. Traditional V&V methods of deterministic simulations assume the conditions are precisely known, for example, a-priori from the test plan or by perfect measurements. In contrast, VV&UQ methods of non-deterministic simulations remove this assumption and decide separately for each input and parameter, whether it is deterministic, aleatory, epistemic, or mixed, and quantify it accordingly. This decision is made for the re-simulation and thus determines the accuracy for which the model is validated. The desired accuracy of a model prediction may differ. There are cases where organizations devote enormous resources to physical experiments in order to quantify the errors as precisely as possible, so that the uncertainties of model validation are much smaller than in actual model prediction. In contrast, it is not advisable to use model predictions with higher accuracy than specified during model validation. Suppose the extreme case with large validation uncertainties and precise model predictions in form of point values. This would contradict the nature of VV&UQ and lead to an erroneous trust in the model. We can employ the variance decomposition in Equation (2.4) for illustration purposes. Large validation uncertainties prevent the appropriate quantification of the error in the second summand and point predictions lead to zero variance in the first summand so that both parts are played against each other. In summary, if uncertainties are considered, it is crucial to perform input UQ and non-deterministic simulations both within the validation and application domain as well as to ensure that the prediction accuracy does not exceed the validation accuracy. As stated earlier, it is recommended to use parameter measurements before parameter estimations and again before calibration methods. Several measurement repetitions offer the possibility to quantify input uncertainties. For inverse methods, Bayesian calibration has the advantage over maximum likelihood estimators that it provides uncertainties in form of probability distributions.

Model-Form Errors e_m – Model Validation

The third source of errors and uncertainties lies in the underlying equations, assumptions, and simplifications of the model, usually referred to as model-form. The only reasonable option to quantify model-form uncertainties is by means of comparisons to physical experiments during model validation. The model-form uncertainty stems from a lack of knowledge and is therefore epistemic in nature.

Observation Errors $e_{y,\mathbf{obs}}$

Since physical experiments are executed during model validation and Input UQ, observation errors are inevitable due to the measurement process. The measured quantities can contain

both epistemic bias errors and aleatory noise. On the one hand, they cause observation errors of output quantities $e_{y,\mathrm{obs}}$ of the physical experiments. On the other hand, they affect the quantification of model inputs and parameters used for the re-simulation. Since these in turn propagate to the simulation outputs, the measurement errors affect both the experimental results and the re-simulation results compared during model validation.

Interpolation and Extrapolation Errors

Additional errors beyond the comparison of Equation (2.1) and (2.2) might occur during the transition from model validation to model prediction, since the validation conditions might significantly vary from the prediction conditions. If this is the case, the quantification of the model-form error at a validation scenario is not representative for an application scenario. Depending on whether an application scenario lies inside or outside the validation space, interpolation or extrapolation uncertainties arise. The latter are usually larger due to lack of knowledge.

Error Aggregation

The previous paragraphs and the Equations (2.1) and (2.2) already indicate that the individual sources of errors and uncertainties are strongly interdependent. Different methods deal with this situation in different ways. Pure calibration approaches are extremely risky. The danger is that they compensate for an incorrect model-form by adapting the model parameters far beyond their physical meaning. Unfortunately, this often happens unconsciously in vehicle simulations without the engineer being aware of the consequences. A simulation model might look appealing from the perspective of the calibration conditions because it is optimized for them and the model quality is not apparent for the prediction conditions. Therefore, model calibration is no replacement for model validation. The latter must be performed with an independent set of experiments. Traditional V&V methods consider the modeling error but analyze it in its entirety. This is less risky compared to pure calibration, but the individual error sources might still compensate or magnify each other. This might cause a misleading trust in the predictive capability of the simulation model. Modern VV&UQ approaches intend to separate the error sources to quantify them individually and to aggregate them in the final step. However, this is challenging because they are strongly interdependent, and it sometimes leads to overly conservative uncertainty estimations due to redundant considerations of the same source. Exemplary approaches will follow in Chapter 2.4 in more detail.

2.3.4 Model and Simulation Types

We can distinguish several types of simulation models that are important for the fundamental understanding of this work and will be taken up later, for example, in Chapter 3.2.6:

Conceptual, mathematical, and computer models: Oberkampf and Roy [18, p. 38] distinguish conceptual models, mathematical models, and computational or computer models. They arise step-by-step when designing a model, formalizing it by means of mathematical equations, and implementing them as a computer code.

Physical and data-driven models: Durst et al. [19] separate traditional computational models that solve complex numerical equations from physics-based models. Furthermore, we can distinguish the physical models, which are deduced from the laws of nature and whose parameters have a physical meaning, from black-box models such as artificial neural networks that are trained inductively from data.

Hierarchical models: Physics-based models are usually hierarchical models that couple individual components to model physical phenomena. Mahadevan [191] proposes the four architectures in Figure 2.7 for the system hierarchy. Besides the degenerate case of a single-component model, a multi-level model includes a vertical hierarchy that connects the system-level with the component-level. A time-varying model contains a horizontal sequence, in which the inputs are passed through the individual sub-models one after the other. A multi-physics model has simultaneous connections between the sub-models and goes beyond the unidirectional flow. An AV, whose architecture can be found in Figure 2.8, shows many similarities with these architectures. From the top-level it is one system. Inside, it contains a sequence of sense-plan-act that controls the classical vehicle. This, in turn, has multiple levels with components such as the powertrain, brake, steering system, or tires, and their components such as the engine or clutch inside the powertrain. Lastly, the AV simultaneously interacts with the traffic environment.

Domain-specific models: Furthermore, there are domain-specific model types, for example, in the field of sensor modeling with ground truth models, idealized models, and phenomenological models, further subdivided into stochastic and physical models [198].

Formal models: In computer science, formal models are used to proof the correctness of systems by means of formal methods. They can be represented as hybrid automata [199] or differential inclusions that extend differential equations with set-based uncertainty [101].

(Non-)parametric models: Models can be separated based on their model-form into non-parametric models such as Gaussian Processes and parametric models.

Time-(in)variant models: If a model reacts instantaneous to a new input, it is a static or time-invariant model. If the model-form includes inputs with a dependency on time, for example, in form of a differential equation, it is a dynamic or time-variant model [200].

(Non-)deterministic models: A model is called deterministic or non-deterministic to indicate whether its inputs and parameters are precisely known or subject to uncertainty.

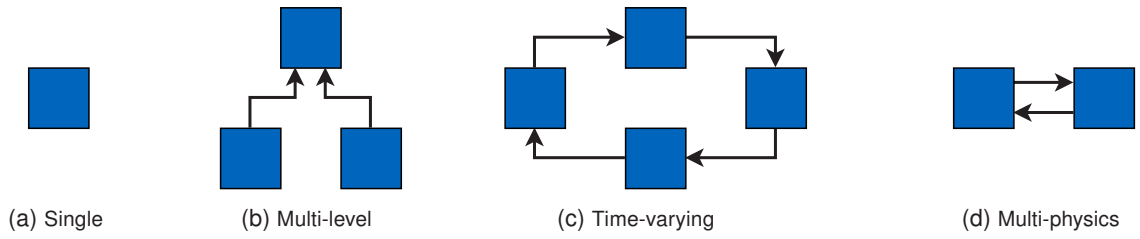


Figure 2.7: Hierarchical model types based on [191].

The previous differentiation of models can be transferred to the simulation to emphasize certain aspects. In computer simulation, briefly referred to as simulation here, a solver is used to obtain the solution of a computational model numerically. In system simulation, the emphasis is on physics-based models to represent systems such as cars, trains, or aircraft. In addition, there are different types of simulation depending on their prediction properties. A deterministic simulation predicts a point value for a single, completely-specified scenario [196]. A non-deterministic simulation considers input and parametric uncertainties and propagates them through the model to obtain the corresponding output uncertainties [188]. Non-deterministic simulation is an umbrella term for interval and probabilistic simulations. We have seen four mathematical structures in Figure 2.6: deterministic point values, epistemic intervals, aleatory probabilities,

and mixed p-boxes. A simulation preserves the higher structure on the output side, if at least one of its inputs has this structure. This yields the following combinations:

- **(General) non-deterministic simulation:**
  If at least one input is mixed or alternatively at least one input is aleatory and at least one epistemic, the output is mixed.
- **Interval simulation:**
  If at least one input is epistemic and the others deterministic, the output is epistemic.
- **Probabilistic simulation:**
  If at least one input is aleatory and the others deterministic, the output is aleatory.
- **Deterministic simulation:**
  If all inputs are deterministic, the output is also deterministic.

V&V methods have developed successively with the emergence of new model types. However, the development went in different communities with different speeds in different directions. For this reason, a heterogeneous research landscape can be seen in model validation today. The interested reader is referred to the review papers [19, 25] for the historical development.

2.4 Model Validation across Engineering Fields

This section provides a survey of model validation methods across several engineering fields. The focus is on highlighting major approaches and trends using exemplary references. A comprehensive collection of references will be given at the end of the state of the art chapter in order to derive the research gaps in Table 2.2. We select the automotive field due to its congruence with the research objectives of this thesis. We accompany it with the railway and the aircraft fields because they have many parallels as system simulations. Finally, we add numerical engineering fields, since they have a leading edge in VV&UQ, from which the other fields can benefit. Within the following subsections, we dedicate separate paragraphs to major model validation approaches that are frequently used or standout by their treatment of uncertainties. They can be recognized by paragraph headings with the special numbering from A1) to A6). We will pay particular attention to them in the analysis in Chapter 2.5.3.

2.4.1 Automotive Model Validation

We start with Figure 2.8 to understand the model validation communities within the overall automotive field. It contains a typical architecture of an AV, its interaction with the external environment, and possibly (dotted arrows) with a human driver depending on the automation level. An AV consists of sensor hardware, a software stack with perception, planning, and control, and the actuators in the vehicle. The software components do not require model validation because there are proven compilers that translate the same software for a new target hardware. There may be some latency effects due to the lower computing power and bus connections in the vehicle, but there is no model development and validation as with the mechatronic components. Therefore, there are two communities that focus on model validation of the mechatronic sensor and vehicle dynamics as isolated components. Nevertheless, component-level validation is never a substitute for system-level validation. The entire AV model, including all hardware and software components that propagate the modeling errors, either amplified or attenuated, must

still be validated to cover the component interactions. On system-level, it is either possible to just compare the final system outputs such as the vehicle trajectory or to additionally compare intermediate component outputs such as the sensor object lists. The latter provides additional insights about the location of the dominant errors within the system that are especially interesting for the developer. Nevertheless, in the end, it is primarily important that the final outputs match, especially for a technical service during the approval of the entire vehicle.

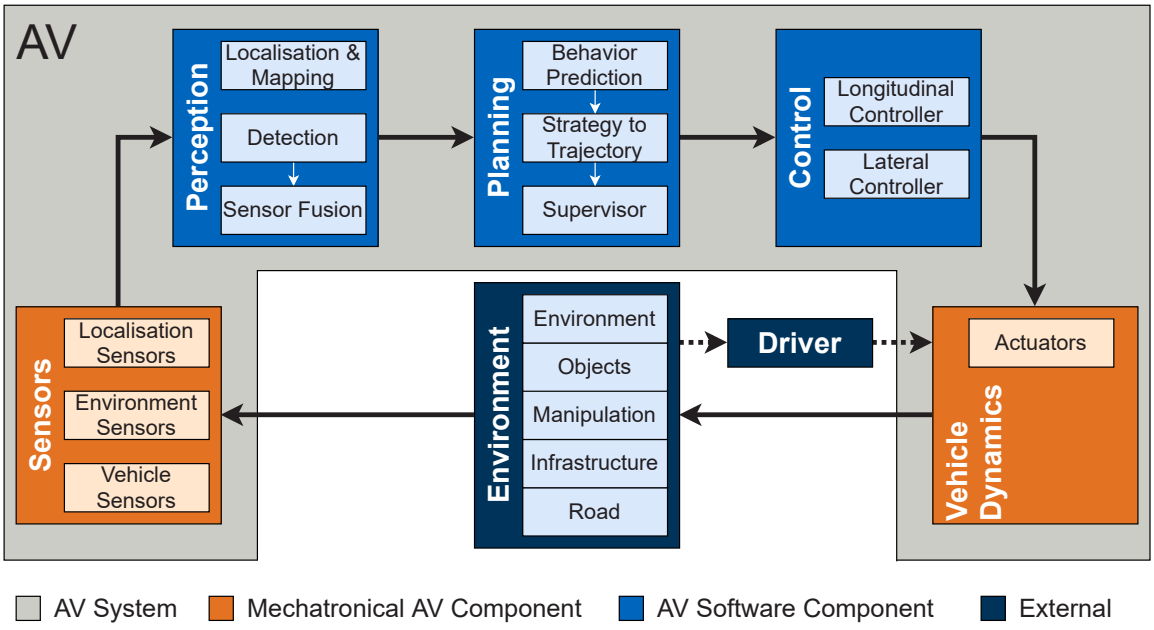


Figure 2.8: Architecture of an AV adapting [201] and five environment layers from [30].

Since model validation targets a use case, there are different references [199, 202, 203] addressing the validation of AV models for different safety assessment approaches. The SBA performs simulations with the entire AV and thus requires a validation of the entire AV model from Figure 2.8. Since the SBA assesses a single AV, it often involves an exact re-simulation of the environment instead of using and validating models of the environment itself. The opposite holds true for the traffic-simulation-based approach. It does not focus on individual AVs, but on macroscopic traffic behavior generated by traffic models that have to be validated. The reachability analysis is an online safety assessment approach in the vehicle during driving. The supervisor performing the online monitoring is located after the trajectory planner within the planning module of Figure 2.8. It relies on the internal behavior prediction using models. It does not require the entire AV model, but the models for the subsequent components that likewise require validation. This includes the behavior prediction models of other traffic participants and the own vehicle dynamics model including the controller that implements the desired trajectory.

The following subsections present exemplary references from model validation of the isolated vehicle dynamics and sensor components, the online vehicle model used within reachability analysis, the entire AV model of the SBA, and the overall traffic model.

Sensor Model Validation

Environment sensor models are a key enabler for virtual-based safety assessment as they influence how realistically the AV perceives the environment. The same holds true for the training of supervised machine learning methods for object detection and recognition based on virtual data [20]. Therefore, sensor models are currently evolving rapidly and the importance of sensor

model validation increases. As indicated earlier, there are different types of sensor models and different types of sensor principles such as camera, lidar, and radar sensors. A physical camera model provides images as raw data and a lidar provides point clouds, each of which is later converted into object lists using algorithms. A phenomenological sensor model provides object lists directly. The three examples illustrate that the sensor type and principle have different requirements for model validation.

The current state of the art in sensor model validation focuses on characteristic properties of a sensor. The first one is that it might require complex environmental representations as its input. For example, Schaermann et al. [204] create a sophisticated reference scenario to provide detailed material characteristics for validation of physical lidar models. They survey an area with houses, streets, parking lots, and cars by means of 3D laser scanning with 120 million points as well as terrestrial and aerial photogrammetry with 3800 images. The second sensor property are the underlying physical phenomena during the perception of the environment inputs. For example, Gruyer et al. [205] analyze the lighting, blur, glow effects, noise, color management, and lens distortion of a camera sensor and its model. They do not perform actual driving tests, but force specific effects in an isolated laboratory setup with dot and retro-lighting charts. The third property are the complex raw data as outputs of physical sensors. They require special validation metrics for the comparison on raw data level that are capable of handling the multidimensional structures. Schaermann et al. [204] present an overall error, Barons and Pearson [206] cross correlation coefficient to compare the occupancy grids of simulation and reality after the low-level fusion of radar and lidar raw data. In addition, the current literature analyzes the influence of the sensor model on the subsequent algorithm to derive requirements for the development and validation of sensor models. For example, Holder et al. [207] investigate the influence of removing features of object recognition algorithms on their classification performance so that they know which features are elementary for the sensor model to include.

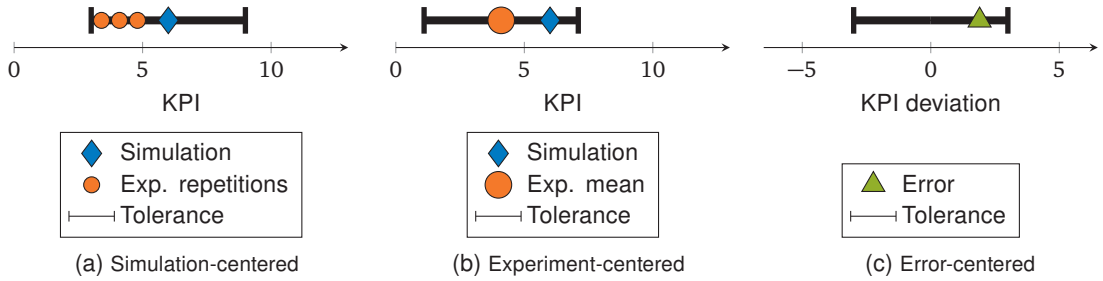
Vehicle Dynamics Model Validation

There are many standards and regulations that describe driving maneuvers such as steady-state cornering [208] or sine with dwell [180], and their evaluation procedures with signals and KPIs. The corresponding standards for model validation [182, 183] and the literature in general [197, 209] reuse these maneuvers and procedures. In addition, they compare simulation and reality using allowable tolerances or model accuracy requirements [18, p. 478]. We refer to it as the tolerance approach [25]. We consider it a major validation approach and dedicate a separate paragraph to it, since it occurs several times in vehicle dynamics and beyond [210].

A1) Tolerance Approach

The tolerances transform the model validation into binary results that postulate either a valid model if the model deviations are below the tolerance thresholds or an invalid one if they exceed them. The literature differs in how the tolerances are applied. The first category takes the deterministic simulation as baseline and adds the tolerances around it so that multiple experimental repetitions have to lie within them. This principle is both possible as tolerance bands across entire time signals [182] or as tolerance values around characteristic KPIs [183]. The second category inverts the principle. It takes the experimental mean as baseline and adds the tolerances around it so that the simulation mean must lie within them [209]. The third category calculates the deviation between simulation and reality and checks whether its absolute value lies within a tolerance [211]. We summarize all of them under the umbrella term of tolerance

approaches. Figure 2.9 illustrates the three categories and Table 2.1 ranks literature sources from the history of vehicle dynamics model validation accordingly.



(a) Simulation-centered (b) Experiment-centered (c) Error-centered

Figure 2.9: Categories of the tolerance approach checking whether (a) the experimental repetitions, (b) the simulation, or (c) the error lies within the correspondingly centered tolerance.

Table 2.1: Classification of references from [25, Tab 4–6] into the categories of the tolerance approach.

Simulation-centered	Experiment-centered	Error-centered
[182, 183]	[197, 209, 212–216]	[211, 217–229]

Recent publications introduce uncertainties into vehicle dynamics model validation. Kutluay [209] extracts confidence intervals from the experimental repetitions based on Student's t-distribution and combines them with the second category of tolerance approaches. He adds both the confidence intervals and the tolerances — either as absolute values or relative percentages — around the experimental mean. He introduces an averaged-input case, which takes the mean of the input conditions of all experimental repetitions and performs one deterministic re-simulation to obtain one propagated output mean. Alternatively, he introduces an averaged-output case, which re-simulates the input conditions of all repetitions separately and takes the mean value of all outputs afterwards. Depending on the non-linearity of the model, there might be a large difference between the two cases [18, p. 492]. Viehof [197] takes up the second case and performs one re-simulation per experiment. For users with lower requirements, he offers a comparison of the simulation output mean with the tolerances around the experimental mean. In addition, he offers a statistical t-test to check whether the scatter of the simulation lies within the scatter of the experiment. He argues that this should be used for the highest requirements because the simulation can never be validated with higher accuracy than the experimental scatter. If the t-test is passed, the model is assumed valid. Rhode [230] goes one step beyond and does not only re-simulate each experimental repetition, but performs Latin Hypercube Sampling to obtain several samples within the input uncertainties of the experiment. He uses a non-deterministic simulation to propagate the uncertainties. He offers four types of confidence intervals based on Gaussian process regression and assumes the non-deterministic model is valid if its output uncertainty lies within the confidence interval of the experiment.

Online Vehicle Model Validation

Models used for online safety assessment in the vehicle likewise require model validation in advance. Researchers in the field of reachability analysis [199] as a formal method are developing conformance testing approaches [44]. They aim at the transfer of formal properties from the model to the physical system by testing whether they conform and are the analogue of classical model validation. Since reachability analysis over-approximates the states a vehicle can reach, its conformance testing approach checks for behavioral inclusion [101]. This means

it ensures that all measured trajectories lie within the set-based trajectory bounds of the non-deterministic simulation model. If this is achieved, the simulation includes the real behavior so that the guarantees obtained from simulation are also valid for reality. Conformance testing consists of three steps [199]. The first step defines a formal notion of conformance [231] such as trace [48, 232] or reachset conformance [199, 233]. The second step formulates an inverse optimization problem that searches for the set representation of each model parameter resulting in the tightest inclusion of the measured trajectories [44]. Thus, it can be rather seen as a model calibration approach with inverse methods. The third step is responsible for the design of experiments to select the calibration scenarios. Similar work addressing the online validation of vehicle dynamics models can be found in [234, 235].

Automated Vehicle Model Validation

A research group from Munich [203, 236–238] focuses on AV model validation for the SBA. They perform physical driving tests and record ground truth measurements of the environment as well as signals from the vehicle bus along the processing pipeline of the AV. They configure the environment in the simulation tool once via the reference information and once via the sensor outputs of the car. Then, they execute the re-simulation twice and investigate the influence of the environment representation on deviations between simulation and experiment along the further processing pipeline. They compare signals such as the AV velocity, its trajectory, and an overall risk measure, and illustrate the respective modeling errors by means of box-plots. Matute-Peaspan et al. [239] perform model calibration using open-loop tests with the vehicle dynamics model and model validation using closed-loop tests with a traffic jam assist. They apply graphical comparisons, confidence intervals, and box-plots. Fremont et al. [240] compare simulation and reality in a crossing scenario with a pedestrian. For the comparison, they use the Skorokhod metric and Dynamic Time Warping as time series metrics, characteristic values such as the minimum distance to the pedestrian, and a formal definition of safety. Similar work can be seen in [51] for a parking lot scenario after formal controller synthesis.

Traffic Model Validation

In contrast to SBA, the traffic-simulation-based approach does not require an exact re-simulation of the environment from the perspective of one AV, but a separate model for the traffic itself that has to be validated. The same holds true for the other layers of the environment model, for example, to model the course of the sun. The interested reader is referred to [195, 202, 241–245] for the traffic model validation and to [246] for the validation of environmental effects.

2.4.2 Railway Model Validation

The literature on the validation of rail vehicle models shows many parallels to the validation of automotive vehicle models. There is also a camp of researchers [247, 248] who focus on deterministic simulations and the tolerance approach. They have also captured it in a corresponding standard [210]. On a closer look, there are some differences. These researchers offer several types of tolerances such as relative, constant, or decreasing ones. They define KPIs such as quasi-static values or maxima and apply tolerances for both the mean values and standard deviations. Besides, there is a second camp of researchers focusing on UQ methods. Funfschilling et al. [249] quantify input and parametric uncertainties of railway vehicle simulations. They present an approach how to model the environment of the train such as its track geometry

and how to propagate it through the simulation model. The interested reader is referred to [250] for a detailed overview about UQ in rail vehicle simulations.

2.4.3 Aircraft Model Validation

The current state of the art in the validation of aircraft models similarly offers the extraction of KPIs from flying data. Hällqvist et al. [251] distinguish stationary and transient properties of the aircraft and represent the latter by means of overshoot values or rise times. In addition, Eek et al. [252] develop the Output Uncertainty Approach in a series of publications. We dedicate it a separate paragraph, since it aggregates errors and uncertainties.

A2) Output Uncertainty Approach

Eek et al. [252] do not focus on the assessment of the entire aircraft system, but on an early concept phase where only prototypes of components are available. Nevertheless, they are interested in making predictions on system-level. Therefore, they intend to propagate the findings from component-level experiments to the system-level without the need for actual system-level tests. They start with verification, calibration, and validation of component models. They also re-simulate each experimental repetition, but they derive an error histogram across the entire scenario space and take its interval boundaries as representative for the output uncertainty of each component. They use the component output uncertainties as input uncertainty for the system and propagate it through its model to get a system-level uncertainty. Finally, they accompany the system-level UQ with model verification. Thus, they state that they consider various sources of errors and uncertainties, but do not strictly separate them and lack model-form uncertainties on system-level due to missing validation experiments with the entire aircraft [253].

2.4.4 Model Validation in Numerical Fields

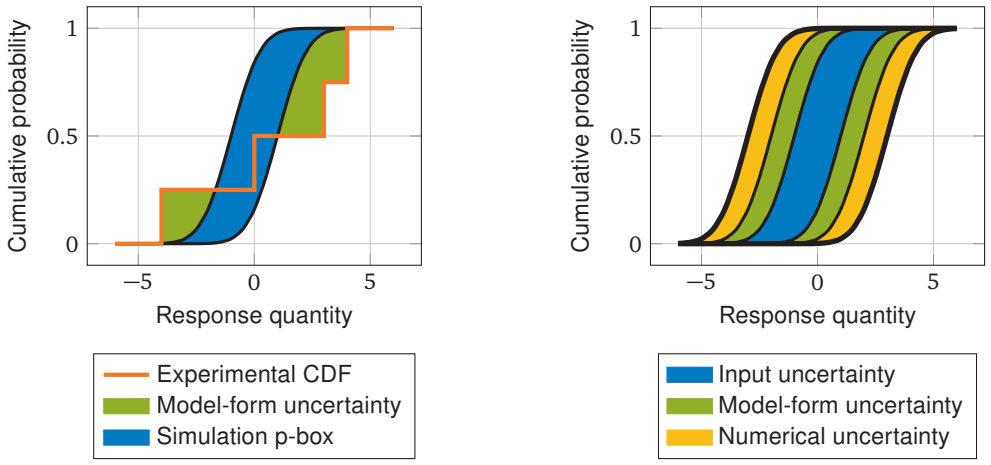
Numerical fields that use complex Finite Element Method simulations have a rich history in VV&UQ. We present four main approaches in the following four paragraphs that stand out by their treatment of errors and uncertainties. They were frequently applied across several engineering fields such as Reynolds-averaged Navier–Stokes equations [254], manufacturing [255, 256], civil engineering [257–259], wind energy [260], watershed modeling [261], naval engineering [262], power electronics [263, 264], nuclear reactor safety [265, 266], or crash simulations [267–270].

A3) Probability Bound Analysis

With frequentist and Bayesian statistics, there are two types that are coexisting for decades. Probability Bound Analysis (PBA) is a VV&UQ approach that extents frequentist statistics [271]. Its origins go back to the work of Ferson et al. [272] and beyond, and its concepts are summarized in detail by Oberkampf and Roy [18]. This paragraph gives a compact overview, while details will follow in the further course of this thesis. As shown in Figure 2.6, they represent deterministic quantities as point values, aleatory uncertainties as probability distributions, epistemic uncertainties as intervals, and mixed uncertainties as p-boxes. PBA contains three pillars for the sources of uncertainty in numerics, model-form, and inputs.

They use parameter measurements and estimations to quantify the input and parametric uncertainties of both the validation domain in Figure 2.10a and the application domain in Figure 2.10b. However, they completely refrain from using inverse calibration methods. They apply a nested uncertainty propagation with epistemic uncertainties in the outer loop and aleatory uncertainties

in the inner loop. They use intensive sampling of uncertainties to perform fully non-deterministic simulations. Aggregating all aleatory sampling results from the inner loop yields one CDF per epistemic sample. Combining all epistemic sampling results from the outer loop yields a p-box with its edges determined by the outer CDFs (blue area in Figure 2.10a and 2.10b).



(a) Model validation (b) Model prediction

Figure 2.10: PBA. It determines the model-form uncertainty during model validation in (a). It then combines the model-form with the input and numerical uncertainty in (b). The green shift to the left and right in (b) corresponds to the size of the green area from (a), respectively.

According to Figure 2.10a, they compare the p-box of the non-deterministic re-simulation during model validation with a CDF of the experiment including its repetitions. They introduce and apply a probabilistic metric referred to as Area Validation Metric (AVM) that focuses on the area between both mathematical structures. The idea of this approach is to better isolate the model-form uncertainty by only calculating the (green) area outside the p-box edges, which symbolize the uncertainties due to inputs and parameters (blue area). They consider measurement errors during the experimental design and extrapolation uncertainties by means of polynomial regression and external Prediction Intervals (PIs).

According to Figure 2.10b, they take the non-deterministic model predictions (blue) in the application domain as baseline and combine them with numerical uncertainties (yellow) and model-form uncertainties (green). The numerical uncertainties originate from quantifying numerical errors by means of verification techniques and from converting them to numerical uncertainties. They add the numerical and model-form uncertainty both to the left and right p-box edge of the input uncertainty to obtain the total prediction uncertainty. This adds conservatism to the simulation due its sources of errors and uncertainties by increasing the p-box width to both sides.

A4) Bayesian Network Approach

A research group led by Professor Mahadevan at Vanderbilt University [191] has developed a Bayesian network approach over the past decade. It also accounts for numerical, model-form, and input uncertainties. They apply model verification techniques to quantify numerical errors, directly correcting the model-form so that its influence is isolated and no longer affects the subsequent quantification of input and model-form uncertainties [192]. They integrate Bayesian calibration methods to estimate model parameters and inputs with corresponding uncertainties. Nevertheless, they perform separate model validation with a Bayesian hypothesis test [192, 273] or via a reliability metric [274, 275] to quantify the model-form uncertainty. Finally, they use a Bayesian network to aggregate all sources of uncertainty. It integrates the model-form

uncertainty into the posterior distributions of the model responses [192] or parameters [276]. They extend the Bayesian network approach to dynamic models with time-varying behavior [200, 277, 278] and to hierarchical models to flexibly incorporate verification, calibration, and validation data at both the component and system levels into the Bayesian network [192].

A5) Interval Predictor Models

Crespo et al. [279] further develop the Interval Predictor Model from [280]. In contrast to a deterministic simulation making point predictions, it maps scalar inputs and set-valued parameters to set-valued outputs. It is represented as a data-driven model without physical meaning, whose boundaries follow a polynomial or radial basis function. The idea is to incorporate the modeling uncertainties into the parameter sets so that the resulting boundaries enclose the system behavior as shown in Figure 2.11. Crespo et al. [279] do not perform model verification and validation but pure calibration. Nevertheless, they formulate the inverse calibration as an interval-valued optimization problem that finds the set of model parameters that bounds the system behavior. Afterwards, they assume that no errors are left and use the Interval Predictor Model directly for model prediction.

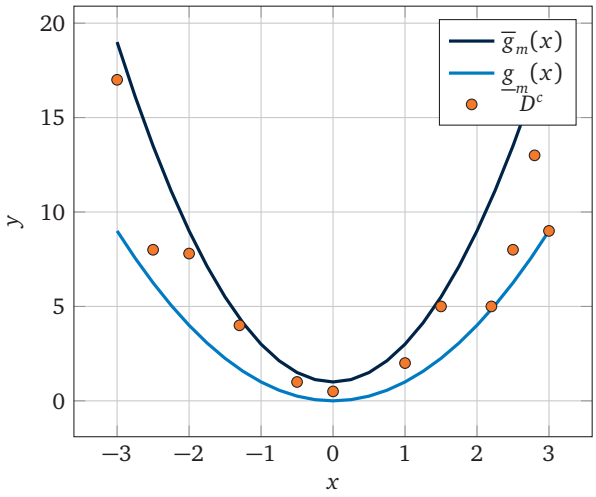


Figure 2.11: Exemplary Interval Predictor Model from [21, Fig. 5] with upper and lower boundaries $\overline{g}_m(x)$ and $\underline{g}_m(x)$ enclosing the calibration data D^c after inverse optimization.

A6) Meta-Model Approach

Hamilton and Hills [281, 282] and Hills [283] present a meta-model approach that focuses in particular on interpolation and extrapolation uncertainties. The task of the meta-model is to relate the behavior of the re-simulation at validation scenarios with the behavior of the model prediction at application scenarios. They sample within neighborhoods around the nominal conditions to reflect the local model behavior. They quantify the errors between simulation and experiment during model validation and consider parameter uncertainties and measurement uncertainties via sampling and bootstrapping. Afterwards, they use the meta-model to infer the quantified errors and uncertainties to the application scenarios and combine them with the nominal model predictions to account for these sources of uncertainty.

2.5 Criticism of the State of the Art

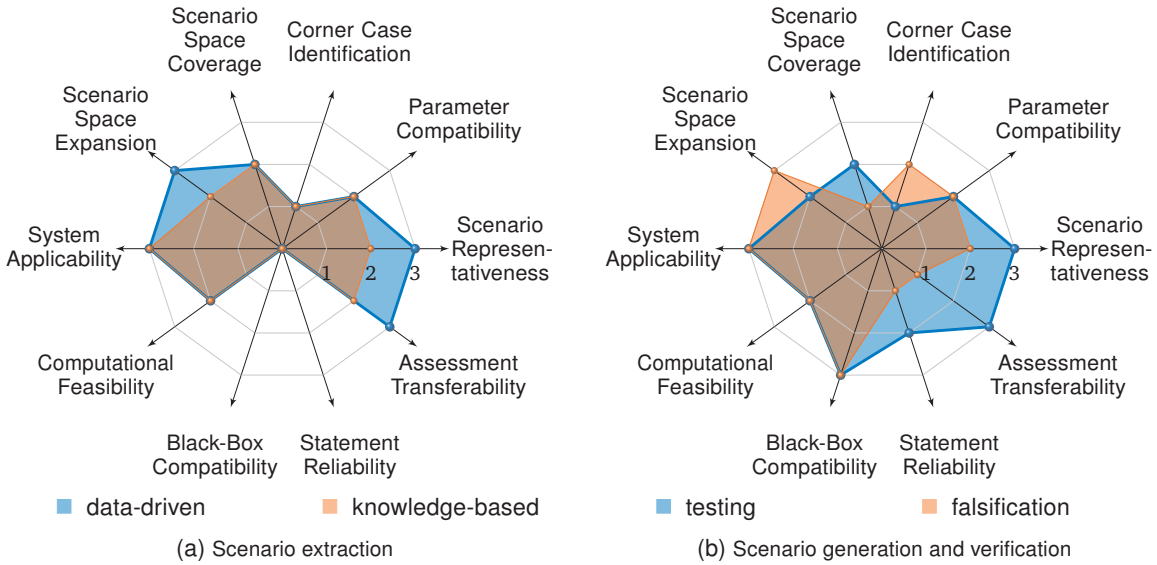
After presenting the current state of the art, this section analyses it and provides constructive criticism. It is divided into two parts for the two section pairs addressing the safety assessment of AVs and the model validation. The first part concentrates on the safeguarding literature from Chapter 2.1 and the three type-approval regulations from Chapter 2.2. The second part concentrates exclusively on the model validation across engineering fields from Chapter 2.4, since Chapter 2.3 introduced fundamental principles. We have to keep in mind that the research objective of this thesis is not to develop a new safeguarding method but to integrate model validation to obtain reliable safety statements. Therefore, the first part serves the purpose of evaluating the safety assessment approaches and the type-approval regulations in order to select the most promising one as the use case to which the model validation refers by definition. For example, it will be important for the configuration of the validation methodology whether an online vehicle model has to be validated for formal verification or whether the entire AV model has to be validated for scenario-based testing. The purpose of the second part is twofold. It reveals a research gap in the model validation of AVs. In addition, it analyses all engineering fields and their main approaches, since they form the basis for the validation methodology of this thesis. Obviously, with the extensive history of model validation, it makes no sense to completely reinvent the wheel, but to specifically address the open issues.

2.5.1 Safety Assessment of Automated Vehicles

The persistently large amount of literature on safeguarding AVs over the last five years indicates that the topic is of particular relevance, but that a consensus has not yet been found. Many challenges are still remaining [7, 13, 284, 285]. There is a variety of safety assessment approaches available, but all of them have strengths and weaknesses in different areas. The shadow mode and the staged introduction already contribute to the collection of experience on the road, but they do not include an entire safety analysis. Traffic simulations will help to reach macroscopic statements, but the current focus is still on single AVs. The role of real-world testing transitions more and more from directly validating system safety to validating the models and assumptions of alternative safety assessment approaches [286]. Formal methods are capable of providing an actual proof of safety, but they lack scalability beyond the trajectory planning module. In general, the perception module is rarely addressed [287–289]. Function-based testing reaches its limits at higher automation levels due to the hardly possible description of requirements. From the perspective of large research projects [10, 11] and the amount of literature, the SBA is currently the most promising approach. It is also finding its way into standardization such as the ISO Working Group 9 and into new type-approval regulations [186]. Thus, this thesis focuses particularly on the SBA as the use case to which the model validation relates.

However, the SBA has its own limitations and leaves questions open in terms of concrete implementation. For example, it is not yet obvious how to identify all relevant scenarios to guarantee completeness. The Kiviat diagrams in Figure 2.12 analyze the data-driven and knowledge-based approach for scenario extraction and the testing-based and falsification-based approach for scenario generation from Chapter 2.1.3. Without taking a closer look at the criteria and ratings at this point, which are described in detail in [9], none of the approaches covers the entire area and impresses with regard to all criteria. The data-driven approach slightly outperforms the knowledge-based approach and the testing-based approach slightly outperforms the falsification-based approach. We will come back to this in Chapter 3.3.1 for the selection of a

scenario method based on our requirements. Instead, we continue with overall remarks on open challenges and research directions.



(a) Scenario extraction

(b) Scenario generation and verification

Figure 2.12: Evaluation of scenario approaches. Comparison of (a) the data-driven and knowledge-based approach for scenario extraction from [9, Fig. 7] and (b) the testing-based and falsification-based approach for scenario generation from [9, Fig. 8]. Details regarding the derivation of the criteria and ratings can be found in [9, Sec. 8].

Some of the limitations can be compensated by combining approaches. For example, formal verification techniques can be applied for the planning module to obtain a formal proof of safety before using the SBA for final system-level tests. Current publications demonstrate new methods by means of simple PoCs. This is suitable for the purpose of illustration but makes it hard to judge how well the methods scale to complex constellations with many scenario parameters. To finally reach industrialization, the complexity has to be extended to stress test the research methods. The basic idea of the SBA is to leave out tedious situations. Nevertheless, a large scenario space remains in the complex traffic environment. It will require further reduction techniques such as functional decomposition [290, 291] to make the safety assessment feasible. This will be of particular importance for the type approval of AVs, since it focuses on ensuring that a system meets a set of minimum requirements. There are yet almost no approaches that are tailor-made for the characteristics of homologation. Whereas research projects such as PEGASUS started with the aim of comparing AV safety to human drivers and maximum mortality rates, the current literature assesses the AV in single scenarios. There is a huge gap left between the microscopic assessment of individual scenarios and macroscopic statements about the impact of AVs on traffic. However, as mentioned earlier, the research objective of this thesis does not correspond to these challenges, but is aligned with model validation. Most safety assessment approaches rely on models and computer simulation, but they are rarely validated. This does not only hold true for the SBA, but also for formal methods, traffic simulation, and the shadow mode. We dedicate Chapter 2.5.3 to a detailed analysis of the validation literature.

2.5.2 Virtual-Based Homologation

The purpose of this subsection is not to criticize the type-approval requirements, since they are given by the UNECE. It is to analyze the regulations regarding their suitability as the PoC of this thesis. The initial research motivation lies in the safety assessment in the context of type approval from the perspective of a technical service. A type-approval regulation lends itself as

a use case, since it contains clearly defined logical scenarios and pass/fail criteria, is publicly accessible, has a neutral character, and is binding for series vehicles. The newest regulation 157 from Chapter 2.2.3 could not be taken as use case during this scientific work, since it came up towards the end of this work, was under heavy development, and no vehicles and functions were available that already complied with it. Nevertheless, it motivates the application of its predecessor R-79 from Chapter 2.2.2 on LKAs. It is already in force and a corresponding vehicle was available during this work. In theory, simulation is not explicitly allowed for the homologation of this specific Level 1 function. However, it serves us as a blueprint for a general virtual-based safety assessment process. Simulation is already used by the manufacturer for ADAS testing and will occur in the future for the type approval of Level 3 vehicles [186] and beyond. R-79 focuses on quasi-stationary cornering behavior. This does not cover complex traffic scenarios of higher automation levels. However, it makes sense to start with a simpler use case and extend it afterwards. The necessary steps for extension will be discussed extensively in Chapter 5.4.

In contrast to regulation 140 on stability control, which is vague with regards to the model validation methodology, but gives recommendations and is supplemented by the ISO standards 19364 and 19365, the newest regulation 157 only provides one statement allowing simulation. Thus, there is a large regulatory gap on the model validation methodology in particular for AVs legitimizing the research motivation of this thesis. This gap will probably not be closed in the regulation itself to offer the manufacturer and technical service flexibility in implementing it. This means that if all parties agree, a simple validation methodology could be applied. However, this is not advisable from a safety perspective, as it leaves available information unused. This dissertation does not aim to maneuver through the type-approval process with as little effort as possible, but uses the LKA type approval as a blueprint for general safeguarding and further engineering fields in order to develop and evaluate a reliable validation methodology.

2.5.3 Model Validation

This subsection begins with general remarks on model validation before examining the six validation approaches, which were highlighted by the paragraph numbers A1-6) in Chapter 2.4, and the engineering fields. The landscape of model validation across all fields is heterogeneous and does not follow a unified methodology. Origins date back to the historical development of methods in different communities focusing on different aspects. Non-deterministic simulations and quantification of errors and uncertainties have a long-standing tradition in numerical fields, but they have only recently been addressed in system simulations. They tend to focus on specific properties of the systems such as complex sensor phenomena or driving behavior. As a rule, users are often only familiar with the corresponding methods from their field of expertise and take them for granted. Even if this is not the case, it is difficult to judge which validation methodology to use for a specific application, since a multitude of methods exists and they have not been compared with each other. Therefore, the Kiviat diagrams in Figure 2.13 contrast the six validation approaches with regard to specific criteria and rate them. Without taking a closer look at the criteria and ratings at this point, which can be found in detail in [21], none of the approaches covers the entire area and impresses with regard to all criteria. The standalone tolerance approach, which is frequently used in the automotive and railway field, performs comparatively poorly. In particular the PBA, the Bayesian network approach, and the meta-model approach stand out by covering the largest area. We will come back to this in Chapter 3.2.7 for the specific configuration of our validation methodology for the use case of LKA type approval. We continue with a short summary and the criticism of the engineering fields.

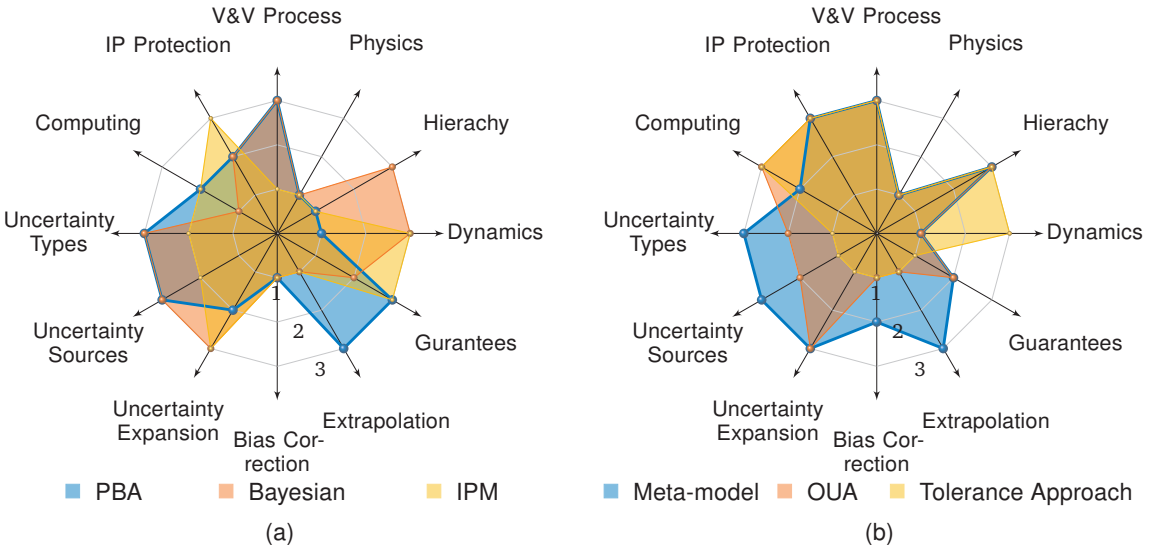


Figure 2.13: Comparison of validation methods from [21, Fig. 11-12]. The split is for visualization purposes only. Details regarding the criteria and ratings can be found in [21, Sec. 8.1]. PBA stands for Probability Bound Analysis, IPM for Interval Predictor Model, and OUA for Output Uncertainty Approach.

Sensor Model Validation: Its references concentrate on the representation of the environment with material properties, on characteristic sensor phenomena, on validation metrics for complex sensor raw data, and on the influence of a physical sensor model on the perception algorithm and the sensor fusion. They interpret the sensor as an isolated component. This is understandable from the perspective of a sensor model developer who wants to ensure the basic functionality of his model. However, it is not sufficient for the use of the sensor model in safety assessment, since there is no generic model validity and component validation must always be accompanied by system-level validation.

Vehicle Dynamics Model Validation: Its references focus on characteristic vehicle dynamics maneuvers, KPIs, and the tolerance approach. The tolerances are based on expert knowledge and should be derived top-down from the use case. They transform the continuous modeling errors into binary decisions about model validity. This is understandable from the model developer's point of view. However, they are often quite subjective, simply set as default values such as $10\,\%$ or $15\,\%$, and sometimes even derived bottom-up from the own model quality. Moreover, it is hardly possible to transfer them to the closed-loop behavior of an AV. Imagine a robust controller that can compensate for a poor vehicle dynamics model, while a sensitive controller cannot. This demonstrates that an isolated component validation may be misleading, since the interactions on system-level are missing. Recent publications introduce non-deterministic simulations into the field of vehicle dynamics. This is a significant contribution, but reveals problems in the actual implementation. They assume that a non-deterministic simulation is valid if its scatter is smaller than the one from the experiment. However, this is not in line with the spirit of non-deterministic simulations [18, p. 490]. As can be seen in conformance testing and reachability analysis, it is almost the opposite. In a perfect world, the non-deterministic simulation should have exactly the same scatter as the experiment. For practical reasons, the scatter of the non-deterministic simulation should enclose the one from the experiment as tightly as possible so that the virtual statements hold for reality. If a model would be valid according to this binary formu-

lation of model validity, it indicates that the input uncertainties are under-approximated and discourages quantification of model-form uncertainty.

Online Vehicle Model Validation: Its references targeting formal methods running online in the AV present conformance testing approaches that bound the system behavior by means of non-deterministic calibration methods. The non-deterministic models match the properties of reachability analysis, but they rely on the assumption that the errors observed during calibration bound all errors across the entire scenario space.

Automated Vehicle Model Validation: Its references focus on the validation of the entire AV model via exemplary quantitative and qualitative comparisons. However, they do not consider different sources of errors and uncertainties, do not systematically describe what scenarios and validation metrics to select for model validation, and do not state how to determine whether the models are ultimately good enough for virtual safety assessment.

Traffic Model Validation: There are early references on the related subject of traffic model calibration. The hype around virtual safety assessment with intelligent traffic models and detailed sensor models does also affect the validation of environmental models in general. There are still many open questions about what these models must be able to deliver.

Railway Model Validation: Its references are mainly split into two camps. The first one addresses the validation of deterministic models via the tolerance approach. The second one addresses the quantification and propagation of input and parametric uncertainties. However, both camps were not yet brought together to an overall VV&UQ framework that addresses several sources of errors and uncertainties.

Aircraft Model Validation: A series of publications introduce a VV&UQ approach into the field of aircraft simulations by presenting the Output Uncertainty Approach. It considers numerical, model-form, and input uncertainties of individual components, but without a strict separation of the uncertainty sources. It represents the component output uncertainties as intervals and infers them to the system-level. It targets cases where a system prototype is not yet available or where system-level tests are impossible due to cost or safety reasons. Therefore, Eek et al. [253] state that it is not a complete VV&UQ approach. They intended to use PBA, but found that the complexity of the aircraft and its amount of parameters is too high. For practical reasons, they developed the Output Uncertainty Approach.

Model Validation in Numerical Fields: It has been the pioneer in the development of VV&UQ approaches. It contains several references addressing verification methods, statistical validation metrics, sampling and propagation techniques, Bayesian calibration, and many more. Nevertheless, there are still limitations and open questions. As stated by Eek et al. [253], its application to system simulations involves challenges due to the rising complexity. This is especially true for dynamical systems with time-variant behavior. Roy [292] identifies three unanswered questions in the VV&UQ pillars. The verification question refers to automatic solution adaption, the validation question on how to split a fixed data set between quantification of model-form uncertainties during validation and model improvements during calibration, and the UQ question on how to aggregate the errors and uncertainties to ultimately obtain a final prediction uncertainty. The latter requires an interpolation or extrapolation in the scenario space or in the system hierarchy space from components to system-level.

2.6 Research Gaps

We first show a map of the state of the art and point out gaps, before stating the research questions. The map in Table 2.2 embeds the references of the presented engineering fields. It distinguishes between deterministic simulations and non-deterministic ones considering modeling uncertainties, as well as whether the validation methods aggregate errors and uncertainties to the final application decision making or not.

Table 2.2: Classification of references dealing with model validation based on [22, Tab 1], highlighting their focus and the focus of this thesis. The gray areas emphasize main contributions within the respective field. This does not exclude references on the left side of a gray area, for example, simpler approaches without error aggregation in numerical fields. The table supplements previous sections focusing on exemplary references with additional ones [293–344] belonging to the same respective category.

	Without error aggregation		With error aggregation	
	Deterministic simulation	Non-deterministic simulation	Deterministic simulation	Non-deterministic simulation
Sensor	[20, 49, 198, 204, 205, 207, 246, 293–305]			
Dynamics	[182, 183, 306]	[197, 209, 230, 344]		
Online	[234, 235]			[44, 48, 199, 231–233]
AV	[51, 203, 236–240]			
Traffic	[202, 242–244]	[241, 245]		[195]
Railway	[210, 247, 248, 307–313]	[249, 250, 314–319]		
Aircraft		[320, 321]		[251–253, 322–328]
Numerics				[18, 81, 191, 192, 195, 200, 272, 273, 275, 277–279, 281–283, 292, 329–343]

Literature focus Thesis method focus Thesis application focus

The non-deterministic simulations show advantages over the deterministic simulations, as they fairly record their state of knowledge in the form of uncertainties and they can separate varying sources. In return, this involves additional effort in quantifying the uncertainties during the experiments and in performing many non-deterministic simulations. The deterministic simulations are often more practical for complex applications. Therefore, we should not yet make a selection between the two simulation types and consider both of them in our validation methodology. Nevertheless, the uncertainty aggregation methods show a clear advantage over the ones without it, since the latter neglect information leading potentially to wrong decisions about the safety of the system. Thus, we concentrate in the remainder of this dissertation on aggregation methods in combination with both deterministic and non-deterministic simulations.

After the table columns, we take a closer look at the engineering fields in the table rows. Whereas the focus in the sensor and AV model validation field is on deterministic simulations without error aggregation, the automotive and railway vehicle dynamics model validation field are in a transition to non-deterministic simulations. However, they do not aggregate the quantified uncertainties.

First approaches with uncertainty aggregation can be seen in the online vehicle, traffic, and aircraft model validation field. Those approaches are the clear focus of the pioneering numerical fields. The gaps in the right half of the table along all engineering fields show a research demand that cannot be covered by one dissertation but by several communities. Nevertheless, we want to contribute by developing a generic validation framework that targets the flexibility of integrating several validation approaches so that several engineering fields can benefit from it. A generic validation process that takes into account the aggregation of uncertainties, different types of simulation models, and several validation approaches is far from existing. The current research landscape is heterogeneous with a lot of small islands. The corresponding research gap is highlighted in Table 2.2 by means of the blue box symbolizing the dissertation focus from the methodological point of view. Thereby, this thesis aims to unify, combine current approaches to new ones, and transfer them in particular to the automotive research community. The aim is not to completely start from scratch, when there is already a myriad of literature.

Since our research objective is aligned with the type approval of AVs, we aim to configure the generic validation framework for a PoC in the AV field. This research gap is highlighted in Table 2.2 by means of the orange box. While there is at least a long history in the vehicle dynamics literature and a high dynamic in the sensor literature, there are hardly any publications dealing with the validation of the entire AV model at all. The validation of isolated component models is helpful but not sufficient, since the selected SBA inspects the entire closed-loop behavior of the vehicle by means of interacting simulation models. Without system-level validation, the statements about AV safety are an indicator at best, completely misleading at worst. Therefore, we focus in the PoC of this thesis on the final system-level validation from the perspective of a technical service during the type approval. We only consider the inputs and outputs of the overall system to be independent of internal variables from the car manufacturer. This is the most important step at the end that includes all component interactions. Nevertheless, it is recommended and usual that car manufacturers validate their vehicle dynamics and sensor models in advance. They can also draw on the state of the art and the generic validation framework of this thesis.

2.7 Research Questions

The research questions of this thesis emerge from the criticism of the state of the art and the research gaps, and they reflect the initial research motivation and objective. The corresponding answers will be provided in the remainder of this work and summarized in the discussion Chapter 5.2. The main research question is:

Q1) **How should a validation methodology be designed to assess the quality of simulations for type approval of AVs using scenario-based testing?**

The first part of the question addressing the methodology is kept universal in order not to limit the solution approaches in advance. We do not intend to rely on predefined tolerances for simulation quality, as this depends heavily on the individual scenarios. Ultimately, the simulation must be accurate enough to lead to the same decisions as if real tests would have been performed. We will specify requirements in the following section that reflect the specific criticism of the state of the art. The second part of the question addressing the use case is kept concise, since model validation refers to a use case by definition. The safety assessment in the context of type approval is of outstanding importance in this work and relies on the SBA as a promising strategy.

We will target the extensibility from LKA type approval to general safety assessment and further use cases in the requirements of the next section.

Based on the major steps of the model validation process, we can specify subquestions under the umbrella of the main research question. The first one aims at the experimental design. The current state of the art in safety assessment focuses on methods to select safeguarding scenarios, but does not contain methods to select scenarios for model validation. There is one paper [345] that determines the optimal amount of real and virtual tests based on assumptions and budget constraints, but it does not address the distribution of the validation scenarios. Even the state of the art in model validation does rarely target the concrete experimental design. It includes general remarks on Design of Experiments [18] and the optimal split between calibration and validation data [190, 346] to reduce costs and prediction uncertainty [276]. However, it does not address the distribution of validation scenarios. Thus, it remains the subquestion:

Q1.1) **What is the best method for selecting scenarios for model validation?**

The second subquestion aims at the comparison between simulation and reality by means of validation metrics. Whereas the literature on sensor model validation introduced complex raw data metrics, the literature on vehicle dynamics model validation often uses differences between KPIs. The other domains offer further metrics, for example, time series metrics or probabilistic metrics such as the area metric [18]. This yields the subquestion:

Q1.2) **Which validation metric suits the comparison between experiments and re-simulations for the type approval of AVs?**

The third subquestion targets the connection between the findings from model validation and the actual type approval of AVs. While vehicle dynamics simulations interpret model validation as a binary problem and neglect errors if they are below a tolerance value, the numerical simulations additionally aggregate errors and uncertainties to final decision making. They regard the aggregation as one of the major challenges in VV&UQ. Therefore, the third subquestion is of outstanding importance and deserves the highest priority among all subquestions. It asks:

Q1.3) **How to integrate modeling uncertainties into the type approval of AVs?**

The three subquestions cover the major steps along the validation process of the main research question. In summary, the first subquestion targets the scenario design for the experiments and re-simulations, the second one the comparison of the results to derive the modeling errors, and the third one the integration of the errors into the type-approval use case. Of course, there are further open questions in the fields of safety assessment and VV&UQ that go beyond these research questions: How to target the trade-off between calibration and validation? How to assign the test scenarios to different XiL environments? How to aggregate time-variant errors and uncertainties? How to deal with an entire fleet of vehicles of the same type? This are only exemplary questions in order to indicate that it involves the efforts of an entire research community. This thesis focuses on the presented questions since they cover the validation process and are in line with the initial research objective. This is immediately reflected in the requirements at the beginning of the subsequent chapter.

3 Model Validation Methodology

This chapter presents the methodological part of this thesis. It specifies requirements for the validation methodology based on the current state of the art. It presents an overall validation framework and dedicates one section to each framework block. It offers several configuration options, selects one for the use case of LKA type approval, and illustrates it with examples. The novelty of the framework arises from its totality including all steps in the given sequence. This matches the conclusions from the state of the art and does not mean that excerpts of it have not already been applied in the literature. On the contrary, it is even the intention to integrate several approaches so that new combinations emerge and all engineering fields benefit.

3.1 Requirements for the Methodology

This section derives requirements for the validation methodology using expert judgment and the findings from the criticism of the state of the art. On the one hand, this work shall develop a generic validation framework that can benefit multiple users and engineering fields. On the other hand, it shall use the validation framework to transfer suitable validation methods to the use case of AV type approval. Thereby, it can add further improvements to the current validation methods. The following requirements have been considered during the development process of this work. Their importance will get more and more obvious in the further course of this thesis and will be discussed in Chapter 5.1. We start with the requirements for a generic methodology:

R1.1) **Modularization:** This thesis shall develop a modular validation framework by identifying individual steps along the validation process.

R1.2) **Unification:** This thesis shall develop a unified validation framework to target the heterogeneous landscape across the engineering fields. The idea is to bring together major validation methods so that all fields can benefit.

R1.3) **Formalization:** The framework shall be mathematically formalized, with clear interface descriptions and input-output mappings. This ensures interchangeability of approaches within a single block without impacting the other ones.

R1.4) **Composition:** The framework shall follow a building block principle like a construction kit. It shall offer major validation approaches by integrating them into the framework blocks and be flexible enough for alternative ones. This adds considerable added value for users with different system complexity and levels of requirements. In addition, this design allows for new combination of approaches that compensate for their individual drawbacks.

R1.5) **Aggregation:** The framework shall emphasize the aggregation of errors and uncertainties, as it is of key importance but still one of the biggest challenges.

This thesis configures the generic validation framework according to the building block principle for the specific use case of AV type approval from the perspective of a technical service. The type approval has a public and independent character and is therefore particularly suitable as a PoC. Nevertheless, the validation methodology also fits the internal safety assessment of car manufacturers thanks to the generic design. However, the specific configuration of the validation framework can differ due to deviating requirements from the manufacturer perspective. We state specific requirements for the framework configuration originating from our case of AV type approval from Chapter 2.2. These will be taken up throughout this chapter to perform the configuration of the framework. The following list contains the specific requirements:

R2.1) **Objective:** The validation method shall fit the objective character of the type approval from the perspective of a technical service.

R2.2) **Unchanged:** The physical vehicle and the corresponding models are provided by the car manufacturer and must not be modified during the type approval.

R2.3) **Protected:** The access to internal system information such as component signals from the vehicle bus might be restricted for the technical service. Thus, the validation method shall focus on important and accessible quantities.

R2.4) **Regulatory:** The validation method shall target the decision making about the AV safety by means of pass/fail criteria from the type approval regulation.

R2.5) **Safety First:** False Negatives (FNs), where the simulation passes the regulation, but the physical vehicle would not, shall be avoided. False Positives (FPs), where the simulation fails, but the physical vehicle would pass, shall also be avoided. However, they shall by far not get the same weight, since they can be corrected by further experiments. In contrast, the FN suggests erroneous trust in the simulation and is therefore particularly dangerous.

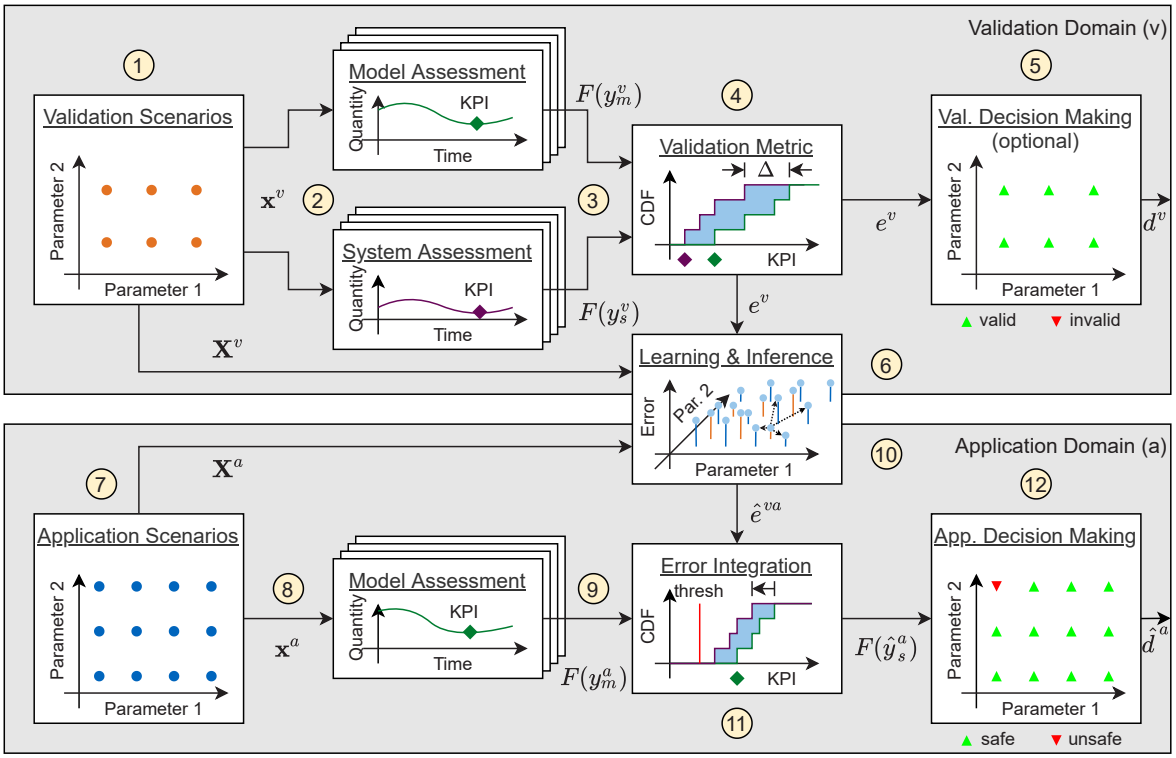
R2.6) **Trustworthy:** The validation method shall increase the trustworthiness of the simulation compared to the tolerance approach as automotive baseline.

3.2 Overall Validation Framework

This section presents the generic validation framework satisfying the generic requirements. It focuses on the overall framework before the subsequent sections continue with the individual framework blocks. We start with an illustration and explain the chapter structure afterwards.

3.2.1 Framework Overview Based on Continuous Example

The goal of this subsection is to introduce the generic framework and to illustrate it briefly using a continuous example. Therefore, the framework is presented in Figure 3.1 as a combination of a block diagram with illustrative plots. The block names and axes labels are kept generic. Nevertheless, the specific points and curves in the plots are unique to each use case and are chosen as examples to demonstrate the principle. Figure 3.1 includes a legend with several steps that are explained in the following paragraphs. More information about these steps follows in the remaining sections of this chapter. The continuous illustration of the framework should give a first impression and later make it easier to understand the relationships between the individual sections and to place them in the overall concept.



1. Create a set of validation scenarios $\mathbf{X}^v$ to assess the quality of the simulation model
2. Execute each validation scenario $\mathbf{x}^v$ multiple times in experiments and re-simulations
3. Extract a KPI y from each repetition to assess the model and system and aggregate them to a CDF $F(y)$
4. Calculate the error area e^v between the model CDF $F(y_m^v)$ and system CDF $F(y_s^v)$
5. Make the decision d^v whether the model is valid by checking whether the error e^v is within a tolerance
6. Train a data-driven model to cover the error e^v across the validation scenarios $\mathbf{X}^v$
7. Create a set of application scenarios $\mathbf{X}^a$ to assess the safety of the system
8. Execute each application scenario $\mathbf{x}^a$ multiple times in simulation
9. Extract a KPI y from each repetition to assess the model and aggregate them to a CDF $F(y)$
10. Infer the error $\hat{e}^{va}$ from the validation scenarios $\mathbf{X}^v$ to the application scenarios $\mathbf{X}^a$
11. Estimate (or bound) the system CDF $F(\hat{y}_s^a)$ by shifting the model CDF $F(y_m^a)$ by the error $\hat{e}^{va}$
12. Make the decision $\hat{d}^a$ whether the system is safe by comparing the system CDF $F(\hat{y}_s^a)$ with a threshold

Figure 3.1: VV&UQ framework representing a virtual-based process of model validation and prediction. While the former compares model and system, the latter purely relies on the erroneous model. This gap is targeted by the vertical error pipeline consisting of the validation metric, error learning and inference, and error integration. They aggregate errors and uncertainties to the application domain so that they are reflected in a reliable decision making.

The framework starts with model validation before proceeding to the actual application of the simulation model. At the beginning, we select a set of validation scenarios. They are illustrated as a coarse full-factorial grid of six orange points in a 2D space. Each scenario is executed in experiments with the physical system and re-simulated using the model. The system and model blocks are stacked to indicate that each scenario is usually repeated several times to cover errors and uncertainties. The resulting output behavior is shown as a green time signal for the simulation model and a purple time signal for the system. We assess each of them equally, for example, by calculating the minimum of the signal as worst-case KPI. Aggregating the KPIs from each repetition of a validation scenario yields a green CDF for the model and a purple CDF for the system. Both CDFs have four steps due to the four stacked blocks or repetitions. The purple CDF lies further on the left as the green CDF, since its KPIs had smaller minima. The exemplary

validation metric quantifies the modeling errors as the light-blue area between both CDFs. We can optionally define permissible tolerances to decide whether the areas or errors are sufficiently small or not. The latter is symbolized by a red triangle pointing down and would suggest a need for improvement. Even if the errors of all validation scenarios are deemed sufficiently small as in this example, we do still not neglect them but learn them in a data-driven error model. We use the light-blue balls at the top of each orange validation scenario as training data set to create a response surface across the 2D space.

The application of the simulation model starts with similar blocks and steps. We select a set of application scenarios, which are represented as a fine full-factorial grid of twelve blue points. They are executed only in simulation but no longer in experiments. Assessing the model results and aggregating the KPIs yields a green model CDF per application scenario. However, the true system CDF is unknown in the application domain. The role of the data-driven error model is to compensate for this by inferring the modeling errors to the new application scenarios. The estimated errors can be seen as the light-blue balls at the top of each blue application scenario. We can integrate an error or uncertainty by shifting or extending the model CDF by means of the light-blue area to estimate or bound the true system CDF. This estimation is shown as the purple CDF lying again to the left of the green CDF. For decision making, we can compare whether the outer purple CDF entirely exceeds a red threshold line. Since this is the case here, the corresponding decision will be symbolized by a green triangle pointing up. In total, the system would in this example be safe for all scenarios except one. The big advantage of the error and uncertainty pipeline is that we do not have to rely on the erroneous simulation results for the final decision making of the application. Instead, we consider the modeling errors and uncertainties to obtain reliable decisions.

From a generic structural point of view, the framework consists of multiple pillars:

1. It is structured into domains that represent model-based activities such as model validation for the validation domain and model prediction for the application domain.
2. Each domain consists of several blocks, ranging from scenario design to decision making, representing the process steps of the respective model-based activity.
3. There are several manifestations of the framework. They originate from the types of simulation models and affect major parts of the framework. For example, the illustration shows a probabilistic manifestation based on CDFs from model and system that in return require a distributional validation metric.

The framework in Figure 3.1 is a reduced version for the purpose of illustration. It contains all generic elements that will be relevant for the specific configuration of our use case. The complete version includes extensions in all pillars. The validation and application domain may be preceded by a verification and calibration domain. The probabilistic manifestation can be extended to a fully non-deterministic manifestation. There are further framework blocks such as a macroscopic assessment or decision making that target not only individual scenarios but all scenarios at once. A block diagram for the complete version of the framework can be found in the appendix in Figure B.1. In this section, we address the complete framework to fulfill our generic requirements. Selected aspects that require comprehensive elaborations but are not core to this work will be mentioned briefly and outsourced to the appendix. However, this should not diminish their relevance to other engineering fields. In the subsequent sections, we focus on the reduced version to ensure a continuous story throughout this dissertation.

3.2.2 Structure of this Chapter

This chapter dedicates one section to the overall framework and one section to each framework block. There are various generic options within a block for various specific use cases. Thus, we define the following three tasks for each framework block:

1. We give an overview about several methods to fulfill the generic requirements.
2. We select one method for AV type approval based on the specific requirements.
3. We illustrate the selection of a specific method with an example.

The framework blocks primarily determine the section headings of the second level, whereas the three tasks primarily determine the subsection headings of the third level. In summary, we first provide options for a framework block, then selected one option, and finally demonstrate it. This is not a strict rule but a rule of thumb, since each framework block is unique and there are blocks such as the assessment that are already specific by definition. We can imagine the chapter structure like a pair of scissors that keeps opening and closing. Table 3.1 takes the framework blocks and the tasks as axes of a matrix to provide the corresponding chapter for each combination. The chapters that deal with the method overview contain literature. However, this literature is not intended as a wide survey to derive research gaps, but to provide specific configuration options to individual framework blocks. Similarly, the exemplary illustrations for each selected method are not intended as results and discussion yet, but should make the theory more tangible. Therefore, the majority of the figures is located in the chapters that deal with the method illustrations. These figures from the penultimate column in Table 3.1 correspond to the plots within the overall framework in Figure 3.1. If we connect these figures, we get the same continuous story, but for the actual use case of LKA type approval.

Table 3.1: Classification of the sections, figures, and tables of this chapter into the methodological selection process for each framework block.

	Method Overview			Method Selection			Method Illustration		
	Chapter	Figure	Table	Chapter	Figure	Table	Chapter	Figure	Table
Overall	3.2.3–6	3.2	3.2	3.2.7		3.3			
Scenarios	2.1.3			3.3.1			3.3.2–4	3.3–5	3.4
Assessment	3.4			3.4			3.4	3.6	
Metric	3.5.1		3.5	3.5.1			3.5.2	3.7	
Learning	3.6.1			3.6.2			3.6.3	3.8	
Aggregation	3.7.1	3.9	3.6	3.7.1			3.7.2	3.10	
Decision Making	3.8			3.8			3.8	3.10	

The principle of the method overview and selection is not only valid for the framework blocks of the remaining sections, but also for the overall framework in this section itself. The following subsections address the framework domains, blocks, and manifestations with the corresponding mathematical notation. The last subsection targets the specific configuration of the framework domains and manifestations for the use case of AV type approval.

3.2.3 Framework Domains

The complete version of the framework contains four domains that symbolize a model-based process. The verification domain is responsible for model verification to identify coding errors and quantify numerical errors. The calibration domain is responsible for determining the model parameters, in the strict sense via inverse calibration methods or in a wider sense via all param-

eterization techniques including the preferred parameter measurements. The validation domain is responsible for quantifying the model-form errors and uncertainties. The application domain includes the actual model predictions for the intended use case. The four domains reflect the order of a model-based process starting from verification, calibration, and validation to the actual application. During a practical implementation, the process is usually not entirely straightforward from start to finish. It may require iterations, for example, back to the parameterization if the modeling errors are deemed too large. The domains are connected to allow the aggregation of modeling errors and uncertainties to the application. The connections are located at the layer of the framework blocks and will be described in Chapter 3.7.

3.2.4 Framework Blocks

This subsection gives a short overview about the framework blocks. Details will follow in the remaining sections. The model-based activities show synergies in their process steps. Therefore, multiple blocks occur several times in several domains. They are highlighted in the reduced framework in Figure 3.1 and the complete framework in Figure B.1 by aligning them in the same column. We will go now step-by-step through the framework from left to right and top to bottom:

Scenarios: On the left side, each domain requires a set of concrete test scenarios. The data sets should be independent of each other to ensure the separate quantification of the uncertainty sources and to avoid optimization towards a set of scenarios.

Model and System: In the second column, the scenarios are executed in physical experiments or in simulation. During model verification, the computational model should be compared with the mathematical model to isolate the effects of the computer and solver. If the mathematical model is not available, for example, due to a black-box simulation tool, code-to-code comparisons should be used instead. During model calibration and validation, the simulation is compared with the physical experiment as reference. The experiments are usually executed first so that the real conditions can be re-simulated afterwards to separate the impact of input and parameter errors on the quantification of model-form errors. The actual model prediction takes place in simulation. Therefore, there is no real system block available in the application domain in Figure 3.1 and Figure B.1.

Assessment: It offers a post-processing of the experiment and simulation results. The transition between the model or system and its assessment can be seamless, depending on the application. An example from the use case of safeguarding AVs are the criticality metrics. They are sometimes part of the functionality of the simulation tool. Otherwise, they have to be calculated externally. Therefore, we can either combine them to the model assessment and system assessment block in the second column in Figure 3.1 or keep them separate such as in the second and third column of the complete framework in Figure B.1.

Errors and Uncertainties: The penultimate column is responsible for the modeling errors and uncertainties. Verification, calibration, and validation metrics calculate the respective errors and uncertainties. These are distance operators relating the simulation to the reference. During verification and calibration, the calculated errors are fed back into the model-form or the model parameters, indicated by the reverse orange arrows in Figure B.1. In addition, the framework includes a vertical pipeline connecting the verification, calibration, and validation domain to the application domain. Its purpose is the direct aggregation of errors and uncertainties. An error model can cover the dependency of the modeling errors on

the scenario inputs. It is trained with the data from model verification, calibration, and validation, as well as inferred to the new application scenarios to consider interpolation and extrapolation uncertainties. The error integration block incorporates the inferred errors into the actual model predictions in the application domain.

Decision Making: The last column addresses the final decision making that converts the continuous values to binary results via thresholds or tolerances. Verification, calibration, and validation decisions make statements about the magnitude of the errors. The application decision making targets the use case, in this thesis the safety of the AV. Lastly, the complete version of the framework in Figure B.1 offers the option to transfer microscopical decisions of individual scenarios to macroscopic statements about a multitude of scenarios. However, this goes beyond the current SBA and will not be relevant for the specific framework configuration.

3.2.5 Framework Notation

This subsection introduces a mathematical notation to formalize the framework. The notation ensures consistency with previous publications [21–23] of this thesis. There are only subtle differences such as the use of the letter D for data sets in [21], whereas they are represented in matrix notation in this thesis. We show some representations for a placeholder symbol α and support it with examples. The framework notation covers the following aspects:

Variables: The arrows in the framework represent variables. We denote the scenario inputs as x, the assessment outputs as y, the errors as e, and the binary decisions as d. We refrain from introducing a separate symbol for the direct model or system outputs, since we work with the assessment outputs after post-processing and want to stick to the established y. For brevity, we often call the model or system assessment outputs just model or system outputs. Within the model, θ refers to the model parameters and h to the step size, and within the decision making, t refers to a threshold for binarization.

Mappings: The framework blocks perform mappings between the variables. We denote each function mapping as g and specify three characters in the subscript for distinction and quick recognition. met stands for metric, lea for learning, inf for inference, int for integration, dec for decision making, maa for macroscopic assessment, and mad for macroscopic decision making. An exception are the simulation model m and the physical system s with just one character, since they occur frequently.

Model and System: Since both the model and the system have an assessment output y, we reuse the m and s in the subscript of the variable symbols y_m and y_s.

Domains: Since some blocks and corresponding arrows appear in multiple domains, we add the domain information to the superscript. We denote the verification domain as n (from numerical), the calibration domain as c, the validation domain as v, and the application domain as a. This yields, for example, g^{v}_{met} for the validation metric or y^{a}_{m} for the model output in the application domain. If two domain letters appear, this indicates a domain transition, for example, $\hat{e}^{va}$ for the inferred validation error in the application domain. If the superscript is missing, the equation is valid for all domains.

Dimensions: We denote scalar point values as italic symbols α, vectors as bold lower case symbols $\boldsymbol{\alpha}$, and matrices as bold upper case symbols $\mathbf{A}$.

Bounds and Estimators: We use an underline and overline decorator to denote lower and upper bounds $\underline{\alpha}$ and $\overline{\alpha}$, as well as a hat decorator $\hat{\alpha}$ to emphasize an estimated value like $\hat{e}^{va}$ of the true value.

(Non-)Determinism: We denote deterministic point values as standalone symbols and add a letter for non-deterministic variables with the symbol it refers to in parenthesis. F represents a CDF, f a PDF, I an interval, and B a p-box. Thus, $F(y_s^v)$ symbolizes the CDF of the system output from validation experiments with multiple repetitions or $B(y_m^a)$ the p-box after propagating aleatory and epistemic uncertainties through the model in the application domain. For brevity, we refrain from stating the random variable in the index of a probability distribution like $F_{Y_s^v}(y_s^v)$ and from specifying conditional probabilities such as $F(y_s^v|g_s, \mathbf{x}^v)$, since it is mostly obvious. The distributions are actual mathematical functions. For consistency, we specify an interval in bracket and set-builder notation as

$$I(\alpha) = [\underline{\alpha}, \overline{\alpha}] = \{\alpha \mid \underline{\alpha} \leq \alpha \leq \overline{\alpha}\} \tag{3.1}$$

with the quantity it refers to as the running variable in parenthesis. A dependency on inputs can be given again via conditions like $I(e^v|g_m, g_s, \mathbf{x}^v)$. In analogy, we specify a p-box as

$$B(\alpha) = [\underline{F}(\alpha), \overline{F}(\alpha)] = \{F(\alpha) \mid \underline{F}(\alpha) \leq F(\alpha) \leq \overline{F}(\alpha)\}\,. \tag{3.2}$$

with the only exception that its boundaries are CDFs instead of point values [194, p. 13].

Figure 3.1 and Figure B.1 include an overview of the framework symbols and Table 3.2 summarizes the block mappings. Strictly speaking, Figure B.1 and Table 3.2 refer to the deterministic framework manifestation to demonstrate the complete version with the simplest setup. In contrast, Figure 3.1 refers to a probabilistic manifestation that suits the illustration with regard to our use case. All of them are aligned to a Multiple-Input-Single-Output case, recognizable by the bold input vectors and italic outputs. It can be easily extended to multiple outputs, since each step of the framework stays the same. The generic framework notation can be filled with the respective symbols of a specific use case. For the LKA type-approval, we define

$$\mathbf{X}^v = \begin{bmatrix} 1 & v_{x,1} & a_{y,\mathrm{ref},1} \\ 1 & v_{x,2} & a_{y,\mathrm{ref},2} \\ \vdots & \vdots & \vdots \\ 1 & v_{x,N^v} & a_{y,\mathrm{ref},N^v} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^{\mathrm{T}} \\ \mathbf{x}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_{N^v}^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{N^v \times (N_x+1)} \quad \text{with} \quad \mathbf{x} = \begin{bmatrix} 1 & v_x & a_{y,\mathrm{ref}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{N_x+1} \tag{3.3}$$

for the set of validation scenarios $\mathbf{X}^v$. The scenario inputs $\mathbf{x}$ contain $N_x = 2$ scenario parameters for the longitudinal velocity v_x and the reference lateral acceleration $a_{y,\mathrm{ref}}$. The homogeneous coordinates are introduced in line with [347] for its statistical calculations applied later in Chapter 3.6. We define the single output as the scalar y directly representing the distance to line. The dimensions are preserved for the subsequent variables such as the error e or the decision d that depend on the single output. For example, the error variable also contains one item for the difference of the distance to line between simulation and reality.

3.2.6 Framework Manifestations

We presented several types of simulation models in Chapter 2.3.4. These types do not only affect the simulation model itself but further blocks of the framework. We can easily imagine that a validation metric must have different characteristics if a deterministic simulation predicts a point

Table 3.2: Framework block mappings in accordance with Figure B.1. The validation error learning, inference, and decision making are shown as representatives for their verification and calibration counterparts.

Framework block	Mapping		Domain		Co-Domain	
System	g_s	:	$\mathbf{x}$	$\mapsto$	y_s	(3.4)
Mathematical Model	g_{mat}	:	$(\mathbf{x}, \boldsymbol{\theta})$	$\mapsto$	y_{exact}	(3.5)
Computer Model	g_m	:	$(\mathbf{x}, \boldsymbol{\theta}, h)$	$\mapsto$	y_m	(3.6)
Verification Metric	g^n_{met}	:	$(y^n_m, y^n_{\text{exact}})$	$\mapsto$	e^n	(3.7)
Calibration Metric	g^c_{met}	:	(y^c_s, y^c_m)	$\mapsto$	e^c	(3.8)
Validation Metric	g^v_{met}	:	(y^v_s, y^v_m)	$\mapsto$	e^v	(3.9)
Validation Error Learning	g^v_{lea}	:	$(\mathbf{X}^v, \mathbf{e}^v)$	$\mapsto$	g^{va}_{inf}	(3.10)
Validation Error Inference	g^{va}_{inf}	:	$\mathbf{x}^a$	$\mapsto$	$\hat{e}^{va}$	(3.11)
Error Integration	g^a_{int}	:	$(y^a_m, \hat{e}^{na}, \hat{e}^{ca}, \hat{e}^{va})$	$\mapsto$	$\hat{y}^a_s$	(3.12)
Validation Decision Making	g^v_{dec}	:	(e^v, t^v_e)	$\mapsto$	$d^v = \begin{cases} 1 & \text{if } e^v \leq t^v_e \\ 0 & \text{else} \end{cases}$	(3.13)
Application Decision Making	g^a_{dec}	:	$(\hat{y}^a_s, t^a_y)$	$\mapsto$	$\hat{d}^a = \begin{cases} 1 & \text{if } \hat{y}^a_s \leq t^a_y \\ 0 & \text{else} \end{cases}$	(3.14)

value, a probabilistic simulation predicts a probability distribution, or a time-variant simulation predicts a time series. Therefore, we distinguish several manifestations of the framework based on different types of simulation models. We select (non-)deterministic models, time-(in)variant models, hierarchical models, and formal models from Chapter 2.3.4, since they are of relevance for systems simulations. At this point, we present the (non-)deterministic manifestation as a representative with high importance for safeguarding AVs. The other manifestations can be found in the appendix in Chapter B.2. We show the basic idea of the manifestation, assign central references from the state of the art, and explain how it fits into the overall framework. The interested reader is referred to [21] for more detailed information and theory.

(Non-)Deterministic Manifestation

We start with deterministic and non-deterministic simulations, which both have been identified as valuable during the derivation of research gaps in Chapter 2.6. The deterministic simulations were often used for the complex systems of the automotive and railway field. The non-deterministic simulations were primarily used by the pioneering numerical fields. For example, the Bayesian network approach [192] builds on probabilistic simulations, the Interval Predictor Models [279] on a type of interval simulation, or PBA [18] on general non-deterministic simulations with p-boxes. Hybrid simulations including both models and hardware components are a mixture between deterministic and non-deterministic simulations. They also exhibit non-deterministic behavior, but it does not arise from the quantification and propagation of input uncertainties but from the intrinsic non-determinism of the hardware itself. The same holds true for software components with random algorithms as sometimes used in machine learning. This makes hybrid simulations different from deterministic simulations, where repeating the same inputs always results in

the same outputs. The hybrid simulations lead to scatter in the outputs in the form of a CDF, despite multiple provision of identical inputs. It would theoretically be possible to combine input uncertainty quantification and propagation with a hybrid environment, but this can be treated the same as a general non-deterministic simulation. In the end, the hybrid simulation still counts as simulation and needs to be validated against reality.

A crucial idea of the modular framework is to keep the interface of each block consistent to ensure interchangeability. The simulation model block always maps model inputs and parameters to outputs, just in form of varying mathematical structures. The following list shows mappings for the different types of simulation models:

$$\text{Deterministic simulation} \qquad g_m : (\ \mathbf{x}\ ,\ \ \theta) \ \mapsto \ \ y_m \tag{3.15}$$

$$\text{Hybrid simulation} \qquad g_m : (\ \mathbf{x}\ ,\ \ \theta) \ \mapsto f(y_m) \tag{3.16}$$

$$\text{Probabilistic simulation} \qquad g_m : (f(\mathbf{x}), f(\theta)) \mapsto f(y_m) \tag{3.17}$$

$$\text{Interval simulation} \qquad g_m : (I(\mathbf{x}), I(\theta)) \mapsto I(y_m) \tag{3.18}$$

$$\text{Interval predictor simulation} \qquad g_m : (\ \mathbf{x}\ , I(\theta)) \mapsto I(y_m) \tag{3.19}$$

$$\text{Non-deterministic simulation} \qquad g_m : (B(\mathbf{x}), B(\theta)) \mapsto B(y_m)\,. \tag{3.20}$$

To achieve this consistency of the interface, the simulation model block is extended internally. An example is shown in Figure 3.2 for a probabilistic simulation. In advance, the uncertainties have to be quantified. Then, the integral [192, Eq. (8)]

$$f_Y(y) = \int f_Y(y \mid \mathbf{x}, \theta) f_X(\mathbf{x}) f_\Theta(\theta) d\theta d\mathbf{x} \tag{3.21}$$

should be solved analytically. However, since this is usually not possible, Figure 3.2 propagates the uncertainties by a multitude of deterministic simulations. It starts sampling the input and parameter uncertainties, propagates each sample through the model, and aggregates all results to an output uncertainty afterwards [192, 343]. Thus, multiple deterministic simulations can approximate a non-deterministic simulation. The important point now is that the interface is placed on the external probabilities and not on the internal deterministic simulation. This ensures that we can insert the block diagram from Figure 3.2 into the model assessment block of the overall framework in Figure 3.1.

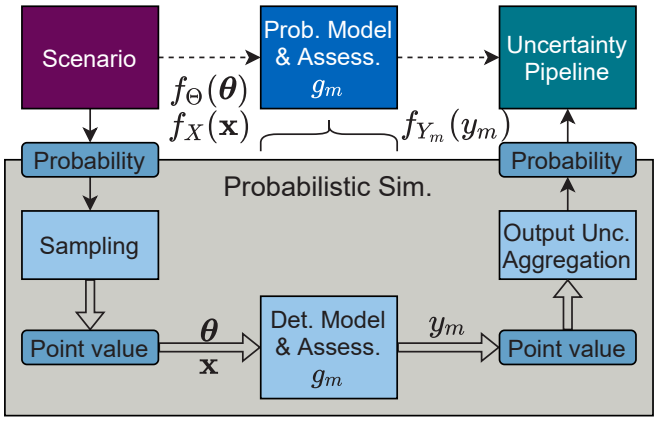


Figure 3.2: Probabilistic simulation propagating uncertainties via several deterministic simulations based on [21, Fig. 6]. The double arrow emphasizes multiple samples.

The simulation type has an influence on the subsequent error and uncertainty pipeline of the framework in Figure 3.1. For example, a deterministic simulation requires a validation metric between two scalar outputs from simulation and experiment, while a probabilistic simulation requires a metric between two probability distributions. The term manifestation refers to the entire framework and contains the simulation as its original name-giving subset. Thus, we should not confuse the deterministic manifestation with a state of the art approach. It symbolizes the combination of a classical deterministic simulation with novel aggregation techniques. They can even contain non-deterministic elements as shown later in Chapter 3.7.2. For brevity and clarity, we continue with the deterministic and non-deterministic manifestation, since the latter is an umbrella for probabilistic and interval simulations. The hybrid simulations are also covered by a mixture of the two. Details and corresponding illustrations will follow in the remaining thesis.

3.2.7 Framework Configuration for Automated Vehicle Approval

This subsection aims at making decisions about the framework configuration for the use case of AV type approval. It examines the framework domains, manifestations, and VV&UQ approaches from the state of the art, while the framework blocks follow in the remaining chapter. The final selections from this subsection are summarized in Table 3.3.

Table 3.3: Selected domains, manifestations, and VV&UQ approaches for LKA type approval. We select two configurations that differ in the (non-)determinism and will be briefly referred to by the distinguishing factor as deterministic and non-deterministic manifestation.

Domains	Hierarchical	Formal	Time-(in)variant	(Non-)deterministic	VV&UQ
Validation & Application	Entire system	–	Time-invariant	Deterministic	–
Validation & Application	Entire system	–	Time-invariant	Non-deterministic	PBA

Framework Domains

From the four domains, we focus on the validation and application domain. The numerical effects are often negligible in automotive simulations as analyzed for vehicle dynamics by Viehof [197] and for LKA type approval by a previous publication of this thesis [22]. Therefore, we skip the verification domain in the following. Nevertheless, this should not be automatically assumed granted when working with complex co-simulations that couple multiple physical processes of an AV with varying step sizes. In accordance with requirement R2.2), we refrain from model calibration in the strict sense, since it would enhance the nominal simulation model from the car manufacturer. Instead, we directly work with the nominal model and possibly accompany it by UQ for the validation and application domain.

Framework Manifestations

After making decisions about the framework domains, we continue with the manifestations:

Hierarchical Manifestation: The AV is a hierarchical system with a sequence of sense-plan-act and multiple levels of vehicle dynamics components. We decided in Chapter 2.6 to perform model validation exclusively on the system-level with the entire AV and dispense with component-level validation. This decision is in line with the requirements R2.3) and R2.4) from Chapter 3.1 that state that the validation methodology should focus on accessible quantities without intellectual property issues for a technical service and on the type-

approval regulation. The validation of components such as the sensor would require access to the vehicle bus and is not addressed by the type approval regulation of the AV. Despite first approaches, it is still an open research question how to aggregate errors and uncertainties in the system hierarchy from component to system level. In contrast to nuclear reactors or aircraft, system-level tests are possible with AVs. Therefore, we focus in the remaining thesis on the system-level validation independent of the internal components. Nevertheless, a car manufacturer should validate the respective component models at least on a qualitative level to gain trustworthiness in advance.

Formal Manifestation: In Chapter 2.5.1, we selected the SBA as the most promising safety assessment approach. Therefore, we do not continue with the formal manifestation in the further course of this thesis. Nevertheless, it is of interest for formal verification methods that aim to prove the safety of AVs.

Time-(In)variant Manifestation: The AV is a dynamic system with time-variant behavior. We can either directly work with the time signals or extract important static features from them. We choose the latter, since this simplification fits the characteristics of safeguarding AVs with scenario parameters and KPIs and allows the majority of VV&UQ approaches. Details can be found in the appendix in Chapter B.2.3. Conversely, we dispense with the general dynamic principle because it involves many open research questions. Its advantages of no limitations or no information loss are not of weight in this thesis. They have little influence for the predefined cornering scenarios with an expected lane-keeping behavior. This may become more relevant for complex trajectory planners that make erratic decisions in situations with obstacles, for example, using machine learning components. However, this is a general challenge in safeguarding [284] and will be discussed later.

(Non-)Deterministic Manifestation: According to Chapter 2.6, a deterministic and a non-deterministic simulation have both strength and weaknesses. Therefore, we decided to compare the deterministic and non-deterministic manifestation of the framework in the further course of this work. Thus, we do not have to perform a further selection here and continue with both options in the remaining sections of this chapter.

VV&UQ Approaches

In the current state of the art in Chapter 2.4, we have seen six major approaches that account for modeling uncertainties. They were analyzed in Chapter 2.5.3 by means of Kiviat diagrams. We evaluate the six approaches with respect to the non-deterministic manifestation of the framework and the use case of AV type approval. The following list contains this evaluation by checking whether the requirements from Chapter 3.1 are satisfied ($\models$) or not ($\not\models$):

Tolerance Approach: The tolerances are subjective expert estimates and not objective ($\not\models$ R2.1), they lack an aggregation of errors and uncertainties ($\not\models$ R1.5), and have no direct link to the type approval regulation ($\not\models$ R2.4). This can be seen in the framework in Figure 3.1 by the fact that there is no connection of the validation decision making block to the application domain. We offer the tolerances within this framework block as an option for the model developer and only use it as a baseline for the results chapter.

Output Uncertainty Approach: It focuses on extrapolation from component to system level and does not separate uncertainties. Thus, it does not match our requirements ($\not\models$ R2.3).

Bayesian Network Approach: It uses subjective Bayes probabilities ($\not\models$ R2.1) and enhances the simulation model via calibration ($\not\models$ R2.2). Therefore, it is not suitable from the neutral perspective of the technical service in this thesis, but from the perspective of the car manufacturer. Then, it would offer comprehensive functionalities.

Interval Predictor Models: It is not based on an actual physical simulation model from the car manufacturer and it is a pure calibration approach without model validation ($\not\models$ R2.2). Thus, it does not fit our requirements.

Meta-Model Approach: It does not learn an error model in the input and parameter space. Instead, a meta-model relates the output behavior in the neighborhood of nominal validation scenarios to the output behavior in the neighborhood of application scenarios for a bias correction ($\not\models$ R2.2). Therefore, it is not entirely in line with our framework and use case.

Probability Bound Analysis: It extends objective frequentist statistics ($\models$ R2.1), does not enhance the simulation model via calibration ($\models$ R2.2), targets pass/fail criteria ($\models$ R2.4), avoids FNs ($\models$ R2.5), and adds statistical guarantees ($\models$ R2.6). Thus, it perfectly fits the nature of our use case of AV type approval and satisfies all requirements. The only exception would be if the internal model parameters are subject to uncertainty and could not be accessed due to intellectual property issues. This is, however, not the case in the PoC of this work ($\models$ R2.3) and solutions could be found to allow partial access. Therefore, we use PBA as starting base for the configuration of the VV&UQ framework.

3.3 Scenario Design

This section addresses the scenario design from the first column of the framework in Figure 3.1. We have already introduced and evaluated scenario methods when presenting the literature on safeguarding AVs. In the first subsection, we take them up according to the building block principle and perform a selection for LKA type approval. In the following subsections, we illustrate the selected scenario methods to design concrete validation and application scenarios. At the end, we can expect a scenario space with two set of points in analogy to the blue and orange ones from the initial framework illustration in Figure 3.1.

3.3.1 Selection of Scenario Design Methods

There is a large research community addressing the selection of scenarios for the application of safeguarding AVs. We divided the related work in Chapter 2.1.3 into knowledge-based, data-driven, coverage-based, and falsification-based methods, and compared them in Chapter 2.5.1 with regards to certain criteria. We do not create a scenario database with parameter ranges, since they are usually given by the type-approval regulation. These application scenarios are not the focus of this thesis, but required to pursue the entire model-based process from the framework in Figure 3.1. Therefore, we select an exemplary method for a first PoC. We develop a new data-driven method that suits the LKA type approval and contains randomness for a fair safety assessment without behavior optimization towards the test cases.

The validation scenarios are an important block of this thesis but rarely addressed. Oberkampf and Smith [340] give a general advice on Design of Experiments to deal with experimental uncertainties. Böde et al. [345] find an optimal amount of real and virtual test scenarios in terms of budget constraints. However, none of these references focus on the distribution of validation

scenarios across the scenario space. Therefore, we analyze the suitability of the extensive scenario literature from safeguarding AVs with respect to model validation. We investigate which of the four categories also match the characteristics of real and virtual validation tests:

Data-Driven Approach: It contains randomness from real-world driving that is desired for safety assessment. However, it is counterproductive for model validation, since the latter builds on the reproducibility of multiple test repetitions.

Knowledge-Based Approach: It is complex to construct an ontology for the infinite traffic environment, and it was never applied to insert knowledge from model validation.

Falsification-Based Approach: It targets challenging scenarios with potential violation of AV safety requirements and often uses a large number of iterations in the virtual world. The focus of model validation is not directly on identifying deviations from safety, but on deviations from reality. The principle could be transferred by adjusting the optimizer's objective function from safety to model errors. However, this will hardly be applicable in the real world due to high effort and limited control over the scenario parameters.

Coverage-Based Approach: It explores the entire scenario space, is applicable in the real and virtual world, and fits the reproducibility of a test scenario with multiple repetitions. Thus, we select the coverage-based category for the design of validation scenarios.

3.3.2 Data-driven Application Scenarios via Event Finding

We start with the application scenarios, despite them being executed after the validation scenarios, since the data-driven algorithm is the foundation. It has the objective of extracting scenarios, which suit the LKA type approval, from a random driving data set in post-processing. The data set originates from the simulation environment, since the algorithm is applied in the application domain to extract application scenarios. It concentrates on stationary conditions of the velocity and lateral acceleration highlighted in R-79. We also call these conditions events and the algorithm event finder to emphasize the data-driven nature. The algorithm is described on the basis of R-79, but can be applied to other use cases by adapting the conditions. It consists of the following steps, which can be tracked successively in Figure 3.3 for one cornering scenario:

D1) Partitioning of the scenario space into 1D acceleration bins:

R-79 [184] contains the two scenario parameters $\mathbf{x} = \begin{bmatrix} 1 & v_x & a_{y,\mathrm{ref}} \end{bmatrix}^{\mathrm{T}}$ of the velocity v_x and "lateral acceleration to follow the curve", referred to as reference lateral acceleration $a_{y,\mathrm{ref}}$ here and usually given normalized to the maximum lateral acceleration $a_{y,\mathrm{smax}}$ from the car manufacturer. We partition the lateral acceleration dimension into 1D bins

$$r_{l,i} \in \mathbf{r}_l = [0.1, 0.2, \ldots, 0.9] \ \forall \ i \in \{1, \ldots, 9\}, \tag{3.22}$$

$$r_{u,i} \in \mathbf{r}_u = \mathbf{r}_l + 0.1 \ \forall \ i \in \{1, \ldots, 9\} \tag{3.23}$$

ranging from the lower boundary $r_{l,i}$ to the upper boundary $r_{u,i}$. The bins extend in $10\,\%$ steps of $a_{y,\mathrm{smax}}$ in the style of the $80\,\%$ to $90\,\%$ band from R-79, highlighted as red area in Figure 3.3a. In contrast, we do not require bins for the longitudinal velocity, since it will be kept almost constant during each cornering simulation.

D2) Butterworth filtering [185] of the measured lateral acceleration signal $\mathbf{a}_y$.

D3) Calculation of the reference lateral acceleration:

Exemplary signals of $\mathbf{v}_x$, $\mathbf{a}_y$, and $\mathbf{a}_{y,\mathrm{ref}}$ can be seen in Figure 3.3a. The measured acceleration depending on the vehicle trajectory shows more oscillations compared to the reference acceleration. The latter depends on the recorded velocity and the curve radius R or curvature κ, which can be obtained by localization on a map, as follows:

$$a_{y,\mathrm{ref}} = \frac{v_x^2}{R} = v_x^2 \kappa \,. \tag{3.24}$$

D4) Binary thresholding:

We use the acceleration bins and additional conditions from R-79 and its amendments [185], highlighted as horizontal lines in Figure 3.3a, to transform the continuous time signals of Figure 3.3a to binary mask signals of Figure 3.3b:

$$\mathbf{b}_v = (\mathbf{v}_x \geq v_{x,\mathrm{smin}}) \wedge (\mathbf{v}_x \leq v_{x,\mathrm{smax}}), \tag{3.25}$$

$$\mathbf{b}_{\mathrm{add}} = (\mathbf{a}_y \leq 1.4 \cdot a_{y,\mathrm{smax}}) \wedge (\mathbf{a}_y < 3.3\,\mathrm{m\,s^{-2}}), \tag{3.26}$$

$$\mathbf{b}_{a,i} = (\mathbf{a}_{y,\mathrm{ref}} \geq r_{l,i} a_{y,\mathrm{smax}}) \wedge (\mathbf{a}_{y,\mathrm{ref}} \leq r_{u,i} a_{y,\mathrm{smax}}). \tag{3.27}$$

D5) Processing of the mask $\mathbf{b}_{a,i}$ to $\tilde{\mathbf{b}}_{a,i}$ via a connected components algorithm [348].

D6) AND conjunction of all binary masks:

$$\mathbf{b}_i = \mathbf{b}_v \wedge \tilde{\mathbf{b}}_{a,i} \wedge \mathbf{b}_{\mathrm{add}} \;\forall\; i \in \{1, \ldots, 9\}\,. \tag{3.28}$$

D7) Event extraction:

We extract sequences of ones, called events, from the resulting binary mask $\mathbf{b}_i$ if they exceed a minimal length. Each event is characterized by its start time $t_{s,ij}$ and its end time $t_{e,ij}$, as well as highlighted as gray area in Figure 3.3.

D8) Extraction of the scenario parameters:

From each event, we select the two scenario parameters mean velocity and bin-centered lateral acceleration:

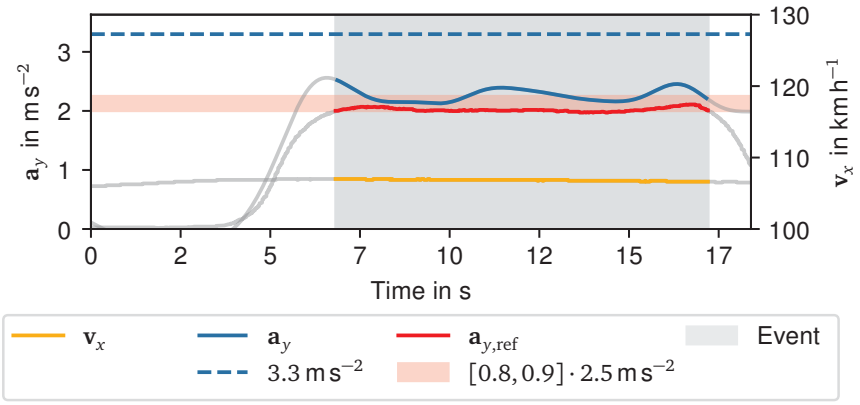
$$v_{x,ij} = \frac{1}{t_{e,ij} - t_{s,ij}} \int_{t_{s,ij}}^{t_{e,ij}} v_x(t) dt \,, \tag{3.29}$$

$$a_{y,ref,ij} = (r_{l,ij} + r_{u,ij})/2 \cdot \mathrm{m\,s^{-2}}\,. \tag{3.30}$$

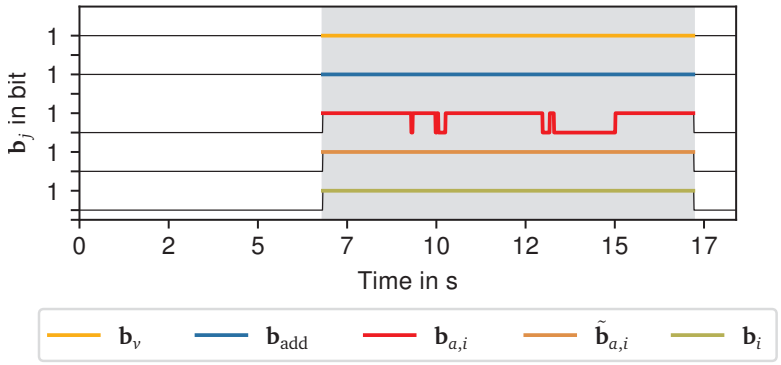
This yields a mean of $v_x = 107\,\mathrm{km\,h^{-1}}$ and the bin center of $a_{y,\mathrm{ref}} = 85\,\% \cdot a_{y,\mathrm{smax}} = 0.85 \cdot 2.5\,\mathrm{m\,s^{-2}} = 2.125\,\mathrm{m\,s^{-2}}$ for the gray event in Figure 3.3a. This scenario point can be seen in Figure 3.4 as part of an entire scenario design.

3.3.3 Coverage-based Validation Scenarios via Map Planning

While the data-driven algorithm post-processes an already executed driving data set, the coverage-based algorithm plans the validation scenarios in advance. They will be executed afterwards in real experiments with multiple repetitions and in corresponding re-simulations. The algorithm relies on road maps with radius and curvature information. The principle idea is to search for long curves with a constant radius and to combine them with varying velocities to obtain scenarios that have a meaningful length and cover the entire space. The search builds on



(a) Continuous time signals before conditions checks



(b) Binary mask signals after conditon checks

Figure 3.3: Data-driven condition checks based on [23, Fig. 3a-b]. The time signals and condition thresholds are shown in (a) and the resulting binary masks after thresholding in (b). The flat yellow $\mathbf{v}_x$ signal is always within the large velocity band of D4) (not shown), the blue $\mathbf{a}_y$ signal is always below the blue dashed threshold line, and the red $\mathbf{a}_{y,\mathrm{ref}}$ signal is mostly within the red acceleration band. The glitches in the red mask are shorter than two seconds [185] and thus pulled-up in the brown mask via the connected components algorithm of D5). The AND conjunction of the yellow, blue, and brown masks yields the final green mask of D6). It determines the position of the gray event area of D7), within which the scenario parameters of D8) are extracted from the yellow $\mathbf{v}_x$ and red $\mathbf{a}_{y,\mathrm{ref}}$ signals in (a).

similarities with the data-driven processing. The algorithm description is kept to a minimum to avoid redundancies. The coverage-based algorithm contains the following steps:

C1) Partitioning of the scenario space into 2D bins:

In contrast to the data-driven pipeline, we create 2D bins for the longitudinal velocity and reference lateral acceleration here. The bins extend in $10\,\mathrm{km\,h^{-1}}$ and $10\,\%$ of $a_{y,\mathrm{smax}}$ steps in the style of R-79.

C2) Full-factorial sampling of the velocity in each bin in $1\,\mathrm{km\,h^{-1}}$ steps.

C3) Calculation of $a_{y,\mathrm{ref}}$ for each velocity sample.

C4) Binary tresholding similar to D4) from Chapter 3.3.2.

C5) Processing of the masks similar to D5) from Chapter 3.3.2.

C6) AND conjunction of all binary masks similar to D6) from Chapter 3.3.2.

C7) Event extraction similar to D7) from Chapter 3.3.2.

C8) Selection of the longest event per 2D bin:

In contrast to the data-driven pipeline with 1D bins, the coverage-based algorithm with 2D bins requires an additional index for the velocity. This results in three indices (i, j, k) for the overall bin index along the acceleration dimension, the velocity dimension, and for a consecutive number of events per bin. At this point, we maximize

$$\arg\max_{k}(t_{e,ijk} - t_{s,ijk}) \tag{3.31}$$

over the event duration to select only the longest event per bin.

C9) Manual selection:

To further reduce the number of scenarios and ultimately the testing effort, we analyze the event length for the selected bins and their coverage of the entire scenario space. If multiple bins are close to each other or if an event has a short duration compared to its neighbors, we discard them.

C10) Extraction of the scenario parameters as center of the selected 2D bins.

We can roughly imagine the coverage-based algorithm as synthetically generating multiple combinations of signals from road maps so that the data-driven algorithm illustrated in Figure 3.3 can be reused internally for planning ahead. The design of the data-driven application scenarios and the coverage-based validation scenarios can be seen in Figure 3.4 for an exemplary driving data set and road map from the German highway A7. Figure 3.4 is intended as an insight into the algorithms and to show that the amount of application scenarios significantly exceeds the amount of validation scenarios to legitimize the virtual-based process. The concrete distribution of scenarios will be analyzed in Chapter 4.1.4.

3.3.4 Scenario Design with Nested Sampling of Uncertainties

For a deterministic simulation, there is just a sampling within the scenario space. For a non-deterministic simulation, however, each sample in the scenario space has to be combined with a sampling of its respective uncertainties. If there are both aleatory and epistemic uncertainties, the uncertainty sampling itself is nested according to Oberkampf and Roy [18]. This yields the following loops in total:

1. Sampling of scenarios within the scenario space,
2. Sampling of epistemic uncertainties for each scenario,
3. Sampling of aleatory uncertainties for each epistemic sample.

We demonstrate this using an exemplary setup from Table 3.4. It contains five parameters consisting of the velocity v_x and reference acceleration $a_{y,\mathrm{ref}}$ from R-79, as well as the wind speed v_w, road slope s_r, and tank load l_t to add further layers of the 6-layer environment model. We treat all of the five parameters both as uncertain parameters and as $N_x = 5$ scenario parameters leading to the extended vector $\mathbf{x} = \begin{bmatrix} 1 & v_x & a_{y,\mathrm{ref}} & v_w & s_r & l_t \end{bmatrix}^{\mathrm{T}}$. This is not mandatory, but usually recommended, since it matches the characteristics of model validation to precisely quantify the input uncertainties around a nominal scenario. If a parameter is only assumed uncertain, but not part of the scenario design, its global uncertainties have to be taken into account instead of the local ones around each scenario. This might reduce the testing effort, but it causes an

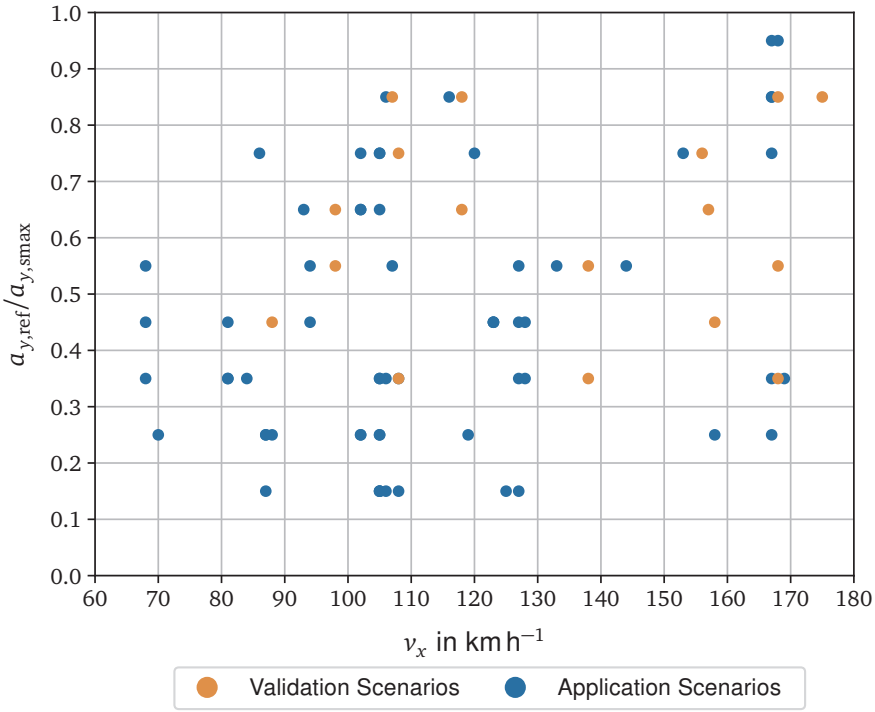


Figure 3.4: Coverage-based validation scenarios and data-driven application scenarios from [23, Fig. 4]. The exemplary application scenario at $v_x = 107\,\mathrm{km\,h^{-1}}$ and $a_{y,\mathrm{ref}} = 0.85 \cdot a_{y,\mathrm{smax}}$, which was extracted from the data-driven algorithm in Figure 3.3, can be seen as one of the blue points. While the orange points belong to the 2D bins, only the 1D acceleration bins affect the blue points. All points are centered vertically within the respective acceleration bin. Whereas the blue points can take each value horizontally along the velocity dimension due to the mean value calculation from D8), the orange points have the resolution of $1\,\mathrm{km\,h^{-1}}$ from C2). Due to the maximum operator from C8), there is a maximum of one orange point per 2D bin. Not all bins are filled due to the availability of curves in the map and to reduce the testing effort. The x axis starts at the specified minimum velocity $v_{x,\mathrm{smin}} = 60\,\mathrm{km\,h^{-1}}$.

inflation of uncertainties. Conversely, a scenario parameter must count as uncertain unless it can be shown to exhibit deterministic behavior.

Table 3.4: Parameter ranges and uncertainties from [22, Tab. 2]. The mean μ and variance σ^2 characterize the normal distribution $\mathcal{N}(\mu, \sigma^2)$. The determination of values is described in [22].

		Validation			Application			Uncertainty (around each scenario)			
		space		scen-	space		scen-	space		samples	samples
Parameter	Unit	min	max	arios	min	max	arios	type	size	(nested)	(rep.)
Velocity	km/h	90	170	3	80	180	6	Alea.	$\mathcal{N}(0, 0.5)$		
Lat. Accel.	-	0.4	0.8	3	0.35	0.85	5	Alea.	$\mathcal{N}(0, 0.01)$	$\sum 10$	$\sum 10$
Wind Speed	km/h	-5	5	2	-5	5	2	Alea.	$\mathcal{N}(0, 2)$		
Tank Load	kg	-20	20	2	-20	20	2	Alea.	$\mathcal{N}(0, 0.5)$		
Road Slope	°	-1	1	2	-1	1	2	Epis.	$[-0.1, 0.1]$	3	
$\prod$			$N^v =$	72		$N^a =$	240			30	$N_r^v = 10$

Table 3.4 specifies the validation and application scenario space by min-max ranges. The systematic determination of the specific values can be found in [22] based on real-world analogies. Briefly summarized, these ranges contain cornering scenarios at highway speeds with typical driving influences of wind, tank, and slope. The validation space is chosen slightly smaller so that the application space encloses it. For illustration, we use a simple full-factorial grid

as coverage-based or space-filling design for both the validation and application scenarios in the outermost loop. The number of validation scenarios $N^v = 72$ is smaller than the number of application scenarios $N^a = 240$ to legitimize the model-based process. This is where the sampling of a deterministic simulation ends. The resulting scenario design is shown in Figure 3.5 in form of crosses and a 2D scenario excerpt. The latter arises by projection from the 5D space onto the original velocity and reference lateral acceleration dimension from R-79.

For a non-deterministic simulation, Table 3.4 specifies the local uncertainties around each scenario. The values were derived in [22] from real-world analogies. In summary, the velocity, acceleration, and tank load can be precisely measured and thus exhibit tight normal distributions. The wind speed can be extracted from measurement stations and shows a wide normal distribution. The road slope must be estimated from road maps and is thus described by an epistemic interval. The uncertainties are assumed constant across the scenario space and between the validation and application domain. We apply full-factorial sampling with 3 steps in the epistemic loop and random sampling with 10 samples in the aleatory loop. This yields in total $72 \cdot 3 \cdot 10 = 2160$ re-simulations and $240 \cdot 3 \cdot 10 = 7200$ model predictions. The resulting design of the non-deterministic simulation is shown in Figure 3.5. We can see local point clouds around the nominal scenarios. In addition to the simulations, the validation domain includes physical validation experiments. They are also affected by the scenario design in the outermost loop. However, they do not require two further loops for nested uncertainty sampling as the non-deterministic simulation does, but one further loop for several experimental repetitions. For $N_r^v = 10$ experimental repetitions and the same uncertainties from Table 3.4, the respective (yellow-green) point clouds can be examined in Figure 3.5. As expected, they are slightly smaller than their (dark blue) counterparts from the non-deterministic simulation with more samples.

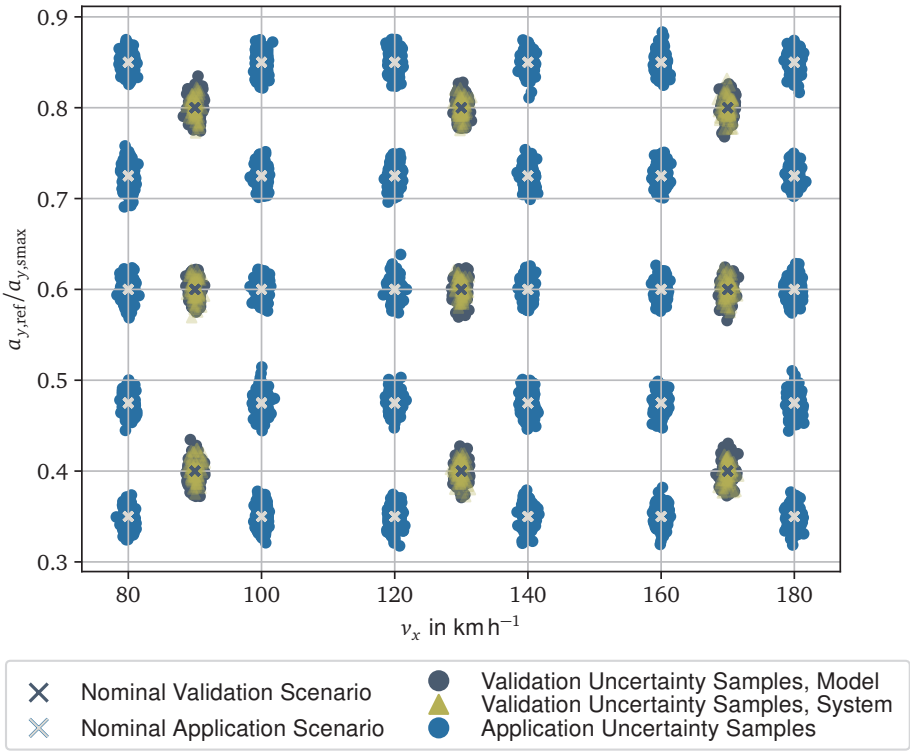


Figure 3.5: Validation and application scenarios based on [22, Fig. 4]. The non-deterministic uncertainty samples form local points clouds around the nominal deterministic scenarios. While each validation scenario has samples for both the model and system, the application scenarios refer exclusively to the model.

3.4 System and Model Assessment

Each concrete scenario has to be executed in the respective real or virtual test environment. There is one exception of the data-driven scenarios, where the order is inverted and the tests are already executed in advance. Either way, the responses from system and model are assessed in post-processing with respect to safety. The focus of this section is primarily on the assessment, since the test executions of the model and system are less interesting from a methodological point of view. The application assessment happens not only in the actual application domain but also in the validation domain. The reason is that model validation always refers to a use case. Thus, it makes sense to validate the same quantities that will be used afterwards during the use case of AV safeguarding. This is in case of R-79 the distance to line from the vehicle edges to the lane markings. The trustworthiness into the simulation model can be further enhanced by validating additional quantities, but it is hard to relate them quantitatively to the actual application quantities. Therefore, we concentrate on the distance to line during the PoC of this thesis. It can be interpreted as a representative for other criticality metrics from AV safeguarding and for general post-processing of arbitrary applications.

The framework in Figure 3.1 illustrated the system and model assessment by extracting minima from time signals. The same principle can be seen in Figure 3.6 for the distances to line in an exemplary cornering scenario. Since R-79 states that the vehicle must not cross any lane marking, the plot contains both the distance to left line signal $\mathbf{y}_l$ and the distance to right line signal $\mathbf{y}_r$. For better understanding of the distance to line values in this thesis, we can relate them to the lane and vehicle width. The outer highway lanes have a width of 3.75 m and the inner ones of 3.5 m according to German construction guidelines [67]. Some scenarios have to be executed on an outer lane, others on an inner lane, depending on their velocity and the traffic situation. The vehicle has a width of roughly 1.90 m. If we assume that the vehicle drives exactly in the center of a lane, we get the same distance from the right vehicle edge to the respective right lane marking and the left vehicle edge to the left lane marking. Subtracting half the widths yields 0.925 m for both the left and right distance when driving on an outer lane and 0.8 m when driving on an inner lane. The latter can be roughly confirmed with the measured distance of 0.75 m at the entrance of the (gray) curve in Figure 3.6. We can use these numbers to establish a rule of thumb for the remaining thesis. If we observe a distance of, for example, 50 cm only less than $50\,\%$ of the maximum available area is left. So we must not be deceived by small-sounding distance values. This is a challenging quantity with high accuracy requirements.

We focus on the minimum distance to line as relevant KPI from a worst-case safety perspective, since a minimum value greater than zero ensures that the entire time signal is greater than zero and ultimately that the vehicle trajectory does not cross the lane marking. We extract the minimum from both the left signal

$$y_{l,\min} = \min_{t \in [t_{s,ij}, t_{e,ij}]} y_l(t) \tag{3.32}$$

and analogously the right signal within the start and end time indices of the scenario event. In addition, we combine both minima to an overall minimum

$$y := \min\{y_{l,\min}, y_{r,\min}\} \tag{3.33}$$

highlighted as red dot in Figure 3.6. This transforms the system and model description

$$g : (v_x, a_{y,\mathrm{ref}}) \mapsto y \tag{3.34}$$

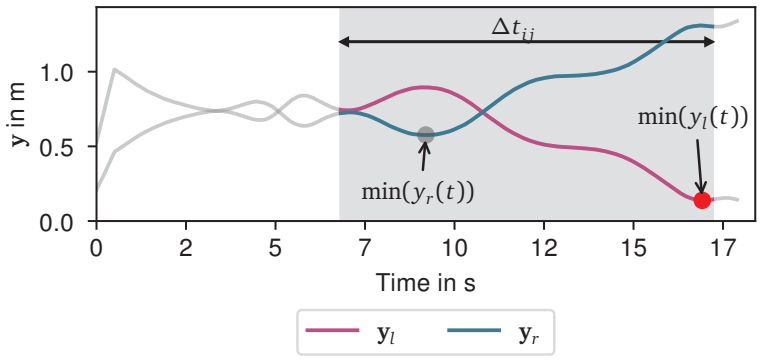


Figure 3.6: Assessment of the distance to line in an exemplary cornering scenario from [23, Fig. 3c]. The gray area refers to the same event from Figure 3.3, which was extracted by the data-driven algorithm. The distance to left line signal $\mathbf{y}_l$ and distance to right line signal $\mathbf{y}_r$ run inversely, since the lane width was constant during the highway section. At the beginning of the right turn, highlighted in gray, the vehicle moves slightly towards the inside of the curve as a preventive measure, before it gets pushed outwards more and more by the cornering forces. The overall minimum (red dot) occurs at the end of the curve near the left lane marking.

to the Multiple-Input-Single-Output case with the two scenario parameters velocity and reference lateral acceleration, as well as the overall minimum distance to line y as assessment output. This has the advantage of a compact PoC for the purpose of illustration. We could alternatively perform the entire model-based process twice for the left and right minimum as two KPIs. This would not have any disadvantages for decision making, but it would mean that we have to show and analyze all result figures and tables twice. Thus, we calculate the overall minimum distance to line for both the model and system, both the application and validation domain, all experimental repetitions, and all uncertainty samples in the case of the non-deterministic simulation. We aggregate the non-deterministic results to probability distributions for the subsequent validation metric calculation. The aggregation yields one CDF $F(y_s^v)$ for the experimental repetitions of each validation scenario and generally one p-box for the propagated uncertainty samples of each validation scenario $B(y_m^v)$ and each application scenario $B(y_m^a)$.

3.5 Validation Metric

In the initial framework illustration in Figure 3.1, we got both a CDF from the system and model assessment and compared them by calculating the area in-between. We have a similar constellation here for the LKA type approval. This section starts again with an overview and selection of validation metrics in terms of the building block principle. The second part presents and demonstrates the theory for one validation metric each for the deterministic and the non-deterministic manifestation.

3.5.1 Overview and Selection of Validation Metrics

Table 3.5 presents a taxonomy of validation metrics that distinguishes different types of metric inputs and outputs. The metric inputs — as outputs from the model and system assessment — may be deterministic or non-deterministic, and in turn static or dynamic depending on the respective framework manifestations. This can be stationary or transient behavior in the context of a vehicle. The output of the validation metric may be Boolean, probabilistic, or real-valued

with a physical unit, and in turn static or dynamic. This results in 24 table cells from which not all are meaningful or already covered by the current state of the art. Table 3.5 includes exemplary references for demonstration.

Table 3.5: Taxonomy of validation metrics from [21, Tab. 2] with examples like the hypothesis test (HT) or area metric with Principal Component Analysis (PCA). The lines with Boolean outputs are included for completeness, but strictly speaking do not indicate the degree of matching of a metric. For example, we can compare single static or stationary values against a tolerance [183] or compare entire time signals with a tolerance band [182] to derive a Boolean output.

	Inputs	Deterministic		Non-deterministic	
Outputs		Static	Dynamic	Static	Dynamic
Boolean	Static	Tol. [183]	Tol. band [182]	HT [274]	HT with KPIs [197]
	Dynamic	-	-	-	-
Probabilistic	Static	-	-	Bayes. HT [274]	Bay. HT with wavelets [273]
	Dynamic	-	-	-	Dyn. reliability metric [200]
Real-valued	Static	Deviation	Vector metric [349]	Area metric [18]	Area metric with PCA [350]
	Dynamic	-	Difference vector	-	-

Further information is provided in the following list and in [21, Sec. 5.2]:

Validation Metrics with Boolean Output: According to Oberkampf and Roy [18, p. 69], the model accuracy requirements or tolerances should not be part of the actual validation metric. Therefore, we created a separate framework block for the validation decision making. This statement argues against the classical hypothesis test as a validation metric, since it integrates the tolerances and yields a Boolean result.

Validation Metrics with Probabilistic Inputs and Output: A Bayesian hypothesis test [274] and the reliability metric [274, 275] also integrate the tolerances. However, they do not provide a Boolean result, but a continuous probabilistic value. Therefore, they match the nature of the Bayesian network approach from Chapter 2.4.4.

Validation Metrics with Real-Valued Output: In contrast to the Bayesian approach, there are deterministic approaches and frequentist VV&UQ approaches that require the result of the validation metric in the unit of the response quantity. In the case of deterministic-static inputs, a simple absolute deviation preserves the physical unit. Mathematical time-series metrics [216, 349, 351] are the analogue for deterministic-dynamic inputs. In the extended frequentist case, the AVM [18] quantifies the area between CDFs or p-boxes and thus also preserves the unit. Further area metrics can be found in [352–355] and general probabilistic metrics in [356–359]. Caution is advised with divergences such as the Kullback-Leibler divergence [360], since they do not correspond to the axioms of a mathematical metric. They satisfy the identify of indiscernibles and the triangle inequality but not the symmetry axiom. This means with regard to validation metrics that the distance from the experiment to the simulation would not be the same as vice versa.

Validation Metrics with Dynamic Inputs and Output: There are two principles how to deal with dynamic behavior. The first one is to extract characteristic values to apply the static approaches. There are a couple of combinations between feature extraction and time-invariant validation metrics in the literature such as Principal Component Analysis with area metric [350], Karhunen-Loeve expansion with area metric [361], wavelet decomposition with Bayesian hypothesis test [273], or window functions with classical hypothesis test [197]. The second principle directly works with dynamic errors and uncertainties. There is one publication [200] that extends the reliability metric to output an actual time series.

In Chapter 3.2.7, we decided to compare the deterministic and non-deterministic framework manifestations, to apply PBA based on frequentist statistics, and to work with scenario parameters and real-valued KPIs for time-invariant behavior. These decisions direct us to select two metrics for our first PoC from the list item on validation metrics with real-valued outputs. We select the absolute deviation for the case of deterministic-static inputs. This metric is signed and thus contains the information whether the simulation or reality has a smaller distance to line value. For the non-deterministic case based on frequentist statistics, we select an area metric, since the area calculation takes into account the entire course of the probability distributions. The exact implementation of the validation metrics follows in the next subsection.

3.5.2 Absolute Deviation and Area Validation Metric

Starting with the deterministic validation metric, the decision fell on an absolute deviation

$$e^v = y_m^v - y_s^v \tag{3.35}$$

between simulation and reality. If there are multiple experimental repetitions, they have to be averaged first as in [209] before calculating the absolute deviation

$$e^v = g_m(\langle \mathbf{x}^v \rangle, \theta_m) - \left\langle g_s(\mathbf{x}^v, \theta_s) \right\rangle . \tag{3.36}$$

The angle brackets as mean operator are required twice, since not only the outputs from the experiment have to be averaged in the second term, but also its inputs inside the first term to obtain only one deterministic re-simulation.

For the non-deterministic framework manifestation, we decided to use an area metric. This type of metric relies on a separate left and right area

$$e_l^v = \int_{F(y_s^v) \leq F(y_m^v)} |F(y_m^v) - F(y_s^v)| dy \, , \tag{3.37}$$

$$e_r^v = \int_{F(y_s^v) \geq F(y_m^v)} |F(y_m^v) - F(y_s^v)| dy \tag{3.38}$$

between simulation and reality. The principle is illustrated in Figure 3.7 for two CDFs. A rectangle counts as left area, if the experimental CDF $F(y_s^v)$ lies on the left side ($\leq$) of the simulation CDF $F(y_m^v)$ or more general the left edge $\underline{F}(y_m^v)$ of the simulation p-box. Conversely, a rectangle counts as right area, if the experiment lies on the right side ($\geq$) of the simulation. Figure 3.7 shows a case with both a left and right area. Nevertheless, it is also possible that one of the two is zero if the experiment lies completely on one side of the simulation. It is even possible that both areas are zero if the experiment lies within a p-box from the simulation.

There are different types of area metrics. One differentiating factor is how they represent the epistemic model-form uncertainty. In contrast to the deterministic validation metric with one real-valued result, this non-deterministic metric yields an epistemic interval with two real-valued interval boundaries. This leads to an advantage, since the latter contains more information that is otherwise lost, for example, by averaging. We do not apply the original AVM [18], since it symmetrically adds the average of the left and right area to both the left and right interval boundary. This leads in turn to a loss of information in cases with varying left and right areas.

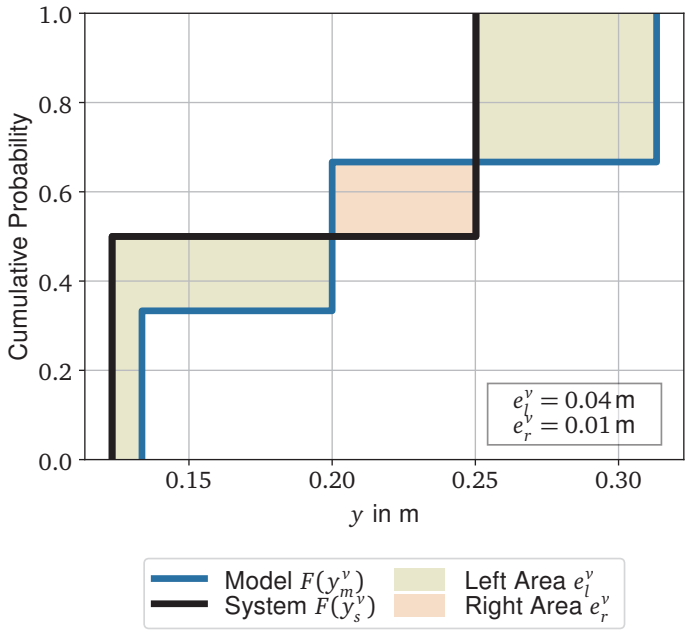


Figure 3.7: Exemplary area metric based on [23, Fig. 6]. The distinction in right and left refers to the location of the system from the perspective of the model. This yields, for example, the orange right area $e_r^v = (0.25\,\mathrm{m} - 0.2\,\mathrm{m}) \cdot (0.67 - 0.5) \approx 0.01\,\mathrm{m}$. Since it is not about the absolute location, the axis limits do not start from zero but highlight the relevant range.

Instead, we treat both areas separately to get the asymmetrical model-form uncertainty

$$I(e^v) := \{e^v \mid \underline{e}^v \leq e^v \leq \overline{e}^v\} := [\underline{e}^v, \overline{e}^v] = [-e_l^v, e_r^v]\,. \tag{3.39}$$

We abbreviate this metric by the acronym Asymmetrical Area Validation Metric (AAVM). Whereas both areas are positive due to the integration, the left interval boundary is negative and the right one positive to represent the orientation between simulation and reality. AAVM can be interpreted as a degenerate case of the Modified Area Validation Metric (MAVM) [354]. The latter additionally interleaves both areas by means of a safety factor, which includes a buffer depending on the number of real test repetitions.

3.6 Error Learning and Inference

This section deals with the error learning and inference in the framework in Figure 3.1. It lies at the interface between the validation and application domain. We initially illustrated it by drawing the errors as points across the validation and application scenarios. At the end of this section, we can expect a similar set of points and a response surface representing the trend of the errors across the scenario space. The error learning is responsible for training a data-driven error model based on the validation metric results. The error inference is responsible for predicting the errors to unseen scenarios in the application domain. In a similar structure to the previous sections, this one first gives a general overview about techniques from the state of the art, before presenting a selected technique in detail for the type approval use case. At the beginning, however, it introduces the two fundamental concepts of ensemble and point-by-point validation. The error learning and inference thereby relate to the point-by-point validation.

3.6.1 Ensemble Validation versus Point-by-Point Validation

If a validation metric compares simulation and reality separately for each validation scenario, it is referred to as point-by-point validation [359]. If all simulation and all experimental results are aggregated first so that the validation metric is only applied once, it is referred to as ensemble validation [359]. A mixture of both is possible when grouping the data into multiple ensembles for comparison [18, p. 656]. The point-by-point validation has the advantage that it distinguishes the magnitude of errors and uncertainties across the scenario space. It offers the possibility to take interpolation and extrapolation uncertainties into account by learning the modeling errors as a function of the scenario inputs. This is particularly important if the modeling errors and the quality of the simulation model vary across the space. Alternatively, the point-by-point comparisons can be aggregated afterwards using a macroscopic validation metric, for example, in form of an error histogram [253] or an integral with the joint probability density across the application domain [359]. The point-by-point validation requires, however, well-characterized experiments with multiple repetitions. If this is not given, the ensemble validation can be helpful. A promising technique is the statistical u-pooling [18, Chap. 12.8.3]. It basically transforms the experimental data into the probability space to aggregate them to one point independent of their physical unit. The resulting CDF can be compared by means of an area metric and back-transformed to the original space. However, it looses physical significance and does not perfectly match the VV&UQ framework from Figure 3.1. It would require an additional block before the validation metric, which was hidden for clarity.

3.6.2 Overview and Selection of Learning Techniques

We use the point-by-point validation in combination with error learning to take interpolation and extrapolation uncertainties as well as dependencies on the scenario space into account. There are a variety of data-driven modeling techniques available in literature. Some of them have already been applied to model an error or discrepancy in different contexts. A linear regression is implemented in [18], a bi-linear regression in [362], a polynomial regression in [18, 363, 364], and Gaussian process regression in [365]. An Interval Predictor Model has not only been implemented as meta-model of the physical system as presented in Chapter 2.4.4, but also to model errors with uncertainty bounds [334]. Further techniques can be found in [366–371]. The traditional linear and polynomial regression techniques perform point predictions, but do not provide information about their prediction uncertainty. This means that they can model the error across the scenario space so that it does not have to be neglected as with the pure tolerance approach from Chapter 2.4.1. In return, however, this causes a usually smaller uncertainty of the learning of the error model itself and of its interpolation and extrapolation to unseen application scenarios. This is already an improvement, but some uncertainties remain. The Gaussian process can provide information about its uncertainty thanks to its composition of normal distributions. Similarly, the Interval Predictor Model contains uncertainty information, but directly as interval bounds instead of a point prediction with variance. The linear and polynomial regression techniques themselves do not have this capability, but they can be extended by means of external PIs [347]. The latter suit our case of error modeling well [18, p. 657], since they contain both the uncertainty of the model learning and the uncertainty of interpolation and extrapolation to new scenarios [347]. Therefore, we select a linear regression with PIs for the PoC. The linear regression can only cover linear dependencies of the error, but the PIs cover nonlinearities. It requires little data and has a low risk of over-fitting.

3.6.3 Linear Regression with External Prediction Intervals

Since the use case of this thesis contains multiple scenario parameters $\mathbf{x} = \begin{bmatrix} 1 & v_x & a_y \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{N_x+1}$, we require a multiple linear regression

$$\hat{e}^v = \mathbf{w} \cdot \mathbf{x}, \quad \mathbf{w} = \begin{bmatrix} w_0 & \cdots & w_{N_x} \end{bmatrix} \in \mathbb{R}^{N_x+1} \tag{3.40}$$

with the scenario parameters as regressor $\mathbf{x}$ and the data-driven weights $\mathbf{w}$. The latter are usually learned by means of a least square estimation

$$\arg\min_{\mathbf{w}} \{s^2\} \quad \text{with} \quad s^2 = \frac{1}{N^v - (N_x + 1)} \sum_{i=1}^{N^v} \left(e_i^v - \hat{e}_i^{vv}\right)^2 \tag{3.41}$$

to minimize the sample variance or mean squared error s^2 between the true validation errors $\mathbf{e}^v \in \mathbb{R}^{N^v}$ and the predicted validation errors $\hat{\mathbf{e}}^{vv} \in \mathbb{R}^{N^v}$. All N^v point-by-point comparisons from the validation domain serve as the training data set. A linear regression surface can be examined in the middle of Figure 3.8 with the orange validation errors as training labels.

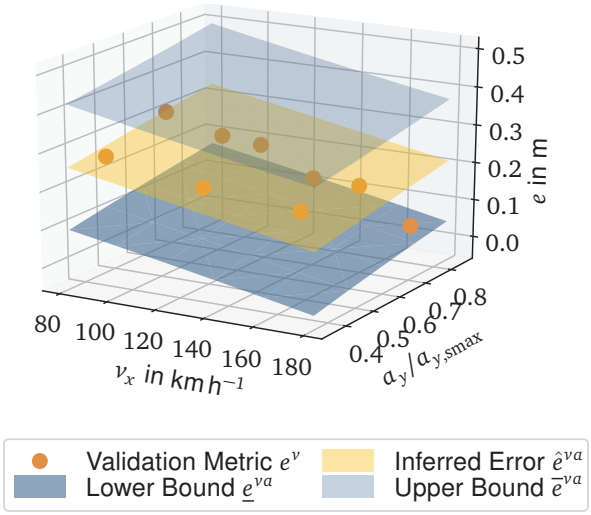


Figure 3.8: Error inference across the application space with linear regression and external PIs from [22, Fig. 7b]. It is shown for the total error of the deterministic manifestation, but looks analogously for the left and right error of the non-deterministic manifestation as in Figure 4.8.

In addition, we use a non-simultaneous Bonferroni-type PI function [347, p.115]

$$g_p(\mathbf{x}^a) = t_{N^v-(N_x+1)}^{\alpha/2} s \sqrt{1 + \mathbf{x}^{aT} (\mathbf{X}^{vT} \mathbf{X}^v)^{-1} \mathbf{x}^a} \tag{3.42}$$

with a t-distribution and a confidence of $\alpha = 95\,\%$, since it can provide an interval estimate for the mean value of a random variable. It contains the uncertainty of both the data-driven model itself and the prediction to future observations $\mathbf{x}^a$ in the application domain. Figure 3.8 contains the corresponding PI to the linear regression estimate. In case of the deterministic framework manifestation, the validation error is a point value and signed. Thus, the PI extends the point value to both sides and thereby transforms it to an interval-valued error estimate

$$I(\hat{e}^{va}) = [\underline{e}^{va}, \overline{e}^{va}] = [\hat{e}^{va} - g_p(\mathbf{x}^a), \hat{e}^{va} + g_p(\mathbf{x}^a)] \tag{3.43}$$

including prediction uncertainties. In the non-deterministic case, the validation uncertainty is already an epistemic interval with two boundaries. Thus, we require two separate linear

regression models and two separate PIs for the left and right areas. The new interval

$$I(\hat{e}^{va}) = [\underline{e}^{va}, \overline{e}^{va}] = [-\hat{e}_l^{va} - g_{p,l}(\mathbf{x}^a), \hat{e}_r^{va} + g_{p,r}(\mathbf{x}^a)]\,. \tag{3.44}$$

is determined by shifting the left interval boundary with the left PI to the left side and by shifting the right interval boundary with the right PI to the right side. These outer shifts enclose the inner ones. At the end, the intention is that the true validation error in the application domain

$$e^{va} \in I(\hat{e}^{va}) \tag{3.45}$$

is embedded inside the interval estimation.

3.7 Aggregation of Errors and Uncertainties

This section targets the aggregation of errors and uncertainties to the application domain. In the initial framework illustration in Figure 3.1, we illustrated the error integration by shifting the CDF of the model response to the left side. This section starts with an overview of aggregation techniques, selects the uncertainty expansion that expands a distribution to the sides, and describes it in more detail.

3.7.1 Overview and Selection of Aggregation Techniques

We classify aggregation approaches based on certain criteria of the taxonomy in Table 3.6.

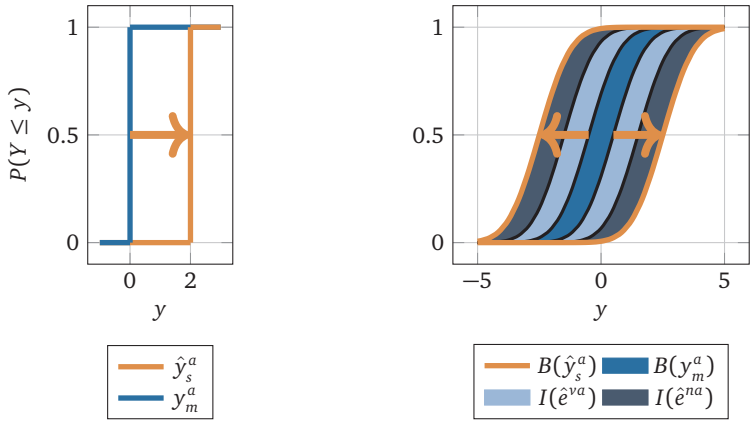
Table 3.6: Taxonomy of error and uncertainty aggregation from [21, Tab. 3].

Aggregation techniques	Aggregation stages	Aggregation source domains	Aggregation target domains
Bias correction	Model parameters	Verification domain	Application domain
Uncertainty expansion	Model-form	Calibration domain	All after the source domain
	Model responses	Validation domain	

Aggregation techniques: The taxonomy distinguishes two fundamental techniques. Whereas the deterministic bias correction uses the quantified modeling errors to actually perform a correction of the nominal simulation model, the non-deterministic uncertainty expansion adds conservatism by increasing the uncertainties of the simulation model. Both techniques are illustrated in Figure 3.9. The bias correction papers [283] argue why the knowledge about the errors should be wasted. The uncertainty expansion papers argue that all sources of uncertainties should be quantified and aggregated [18]. While the bias correction might be overly risky and lose credibility of simulation, the uncertainty expansion might be overly conservative with an inflation of uncertainties and prohibit realistic decision making.

Aggregation stages: Both aggregation techniques can be applied at several stages. It is possible to correct or expand the internal model parameters, internal model-form, or resulting model responses.

Aggregation source domains: The errors and uncertainties are quantified in varying source domains. Numerical uncertainties emerge in the verification domain, parametric uncertainties in the calibration domain, and model-form uncertainties in the validation domain.



(a) Bias correction (b) Uncertainty expansion

Figure 3.9: Bias correction versus uncertainty expansion from [21, Fig. 9]. The former example contains a correction to the right, while the latter example expands the area to both sides as in PBA.

Aggregation target domains: The framework in Figure B.1 offers two directions for the aggregation of the error and uncertainty sources. Either they are directly integrated into the application domain by means of the fourth column of the framework. This is the focus within this thesis. Alternatively, they are fed back via the inverse orange arrows of the framework so that they affect all subsequent domains in the model-based process.

The taxonomy provides the criteria in Table 3.6 to classify the VV&UQ approaches from the state of the art regarding their aggregation properties. This is done comprehensively in the previous publication [21, Sec. 6]. For example, the Bayesian network approach [192] can correct the numerical errors internally in the model-form, can expand the parametric uncertainties via calibration, and finally expand the posterior distributions of the model responses with an additional uncertainty from the Bayesian hypothesis test. At this point, however, we will not go into any further detail and concentrate on the uncertainty expansion of model responses from the validation to the application domain due to the requirements from Chapter 3.1 and our framework configuration from Chapter 3.2.7. The bias correction, the internal model parameters, and the internal model-form would not match the type approval from the perspective of a technical service without modification of the simulation model ($\not\models$ R2.2).

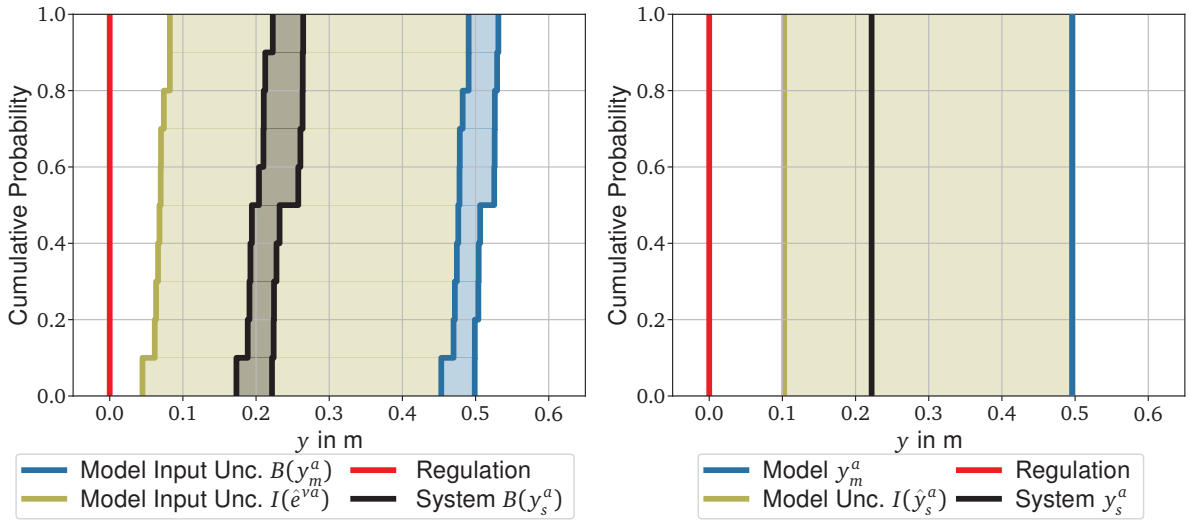
3.7.2 Uncertainty Expansion of Model Responses

The uncertainty expansion is a non-deterministic technique. Thus, it directly fits into the non-deterministic manifestation of the framework. In contrast to the implementation of PBA in [18] with a constant numerical uncertainty and a symmetrical model-form uncertainty due to the original AVM, we neglect the numerical uncertainty for our automotive application and use our AAVM with varying left and right values. This uncertainty expansion can be formalized according to our notation as

$$\begin{aligned} B(\hat{y}_s^a) = B(y_m^a) + I(\hat{e}^{va}) &= \{F(\hat{y}_s^a) \mid \underline{F}(y_m^a - \underline{e}^{va}) \leq F(\hat{y}_s^a) \leq \overline{F}(y_m^a - \overline{e}^{va})\} \\ &= \{F(\hat{y}_s^a) \mid \underline{F}(y_m^a + \hat{e}_l^{va} + g_{p,l}(\mathbf{x}^a)) \leq F(\hat{y}_s^a) \leq \overline{F}(y_m^a - \hat{e}_r^{va} - g_{p,r}(\mathbf{x}^a))\} \end{aligned} \tag{3.46}$$

by substituting Equation (3.44). Care must be taken with the signs, since the CDF is a function and thus a plus leads to a shift to the left, as opposed to the interval notation. The shift of the p-box is illustrated graphically in Figure 3.10a. Unlike adding a simple offset that would be

subjective and fixed for all scenarios, this shift is objectively derived based on regression and statistics and adjusts for each application scenario.



(a) Non-deterministic (b) Deterministic

Figure 3.10: Uncertainty expansion and decision making from [22, Fig. 8] for (a) the non-deterministic and (b) deterministic framework manifestation. The blue simulation prediction is expanded via the green uncertainty bounds and compared against the red decision-making threshold. The black system results as ground truth are normally not available in the application domain, but added for the analysis in Chapter 4.1 to show that they lie within the uncertainty bounds. The green shift in (b) can be traced back exemplarily by means of the error surfaces in Figure 3.8. Intersecting the upper PI surface at the coordinates $v_x = 100\,\mathrm{km\,h^{-1}}$ and $a_{y,\mathrm{ref}} = 0.35 \cdot a_{y,\mathrm{smax}}$ from this application scenario yields the $0.4\,\mathrm{m}$ from the green area.

The deterministic errors would fit to the bias correction. However, we do not want to enhance the original simulation model from the car manufacturer according to requirement R2.2) from Chapter 3.1. It would ultimately mean to trust the data-driven error model more than the physical simulation model. Therefore, we convert the deterministic error to a non-deterministic uncertainty. This yields an uncertainty expansion

$$I(\hat{y}_s^a) = \begin{cases} [y_m^a - \overline{e}^{va}, y_m^a] & \underline{e}^{va}, \overline{e}^{va} \geq 0 \\ [y_m^a - \overline{e}^{va}, y_m^a - \underline{e}^{va}] & \underline{e}^{va} \leq 0, \overline{e}^{va} \geq 0 \\ [y_m^a, y_m^a - \underline{e}^{va}] & \underline{e}^{va}, \overline{e}^{va} \leq 0 \end{cases} \tag{3.47}$$

with three cases to ensure the expansion, but prohibit an improvement of the simulation model so that its value is always included within the interval bounds. This is a new combination that was only made possible by the modular building block principle of the framework. The deterministic framework manifestation unifies classical deterministic simulations with modern non-deterministic uncertainty bounds. It can therefore be interpreted as a pragmatic compromise that is executable even for complex systems and still accounts for a total uncertainty. However, it does not quantify all sources of uncertainty separately as with the more elaborate non-deterministic framework manifestation. The respective graphical illustration can be seen in Figure 3.10b and will be taken up again in the next section.

3.8 Decision Making

The final step in the framework in Figure 3.1 is the application decision making. We initially illustrated it by checking whether a CDF lies above a safety threshold and by plotting the final decisions as green and red triangles across the application domain. We take up the thresholding in this section and demonstrate it analogously. However, we yet refrain from providing the decisions across the scenario space, since they would anticipate the results of the next chapter. The type-approval regulation ultimately states the vehicle must not cross the lane. This means that the minimum distance to line from the vehicle edges to the lane markings must be greater than zero. This lower threshold is included in Figure 3.10b as vertical red line. In the deterministic case, the entire filled area between the outer uncertainty bounds exceeds the threshold and thus passes the regulation requirements in this exemplary scenario. For sure, it is harder to achieve this with the entire area instead of the point value from the nominal simulation. Therefore, passing it anyway gives additional guarantees and trustworthiness into the simulation model.

The non-deterministic manifestation offers further options, since it separately considers aleatory input uncertainties leading to several steps of the p-box edges. The highest statistical guarantee is obtained when ensuring that all steps exceed the zero threshold. This is recommended from the safety perspective. For the example with 10 steps, this corresponds to a confidence of $90\,\%$. This confidence refers to the CDF steps and originates from the sampling uncertainty depending on the number of repetitions. It cannot be equated with a $90\,\%$ guarantee of passing, since this is impossible without the real system. There are further factors influencing the p-boxes such as the $95\,\%$ confidence of the PI calculation or implicit assumptions of the error model that the assessment behavior at the application scenarios is similar to the behavior at the validation scenarios. If less confidence is sufficient, less steps have to pass. If more confidence is required, the number of aleatory samples has to be increased. The width of the p-box emerges from the separate quantification of epistemic model input, model-form, and prediction uncertainties. Since more uncertainties are considered and they are treated separately, it yields higher guarantees that the entire p-box from the non-deterministic manifestation exceeds the zero threshold in Figure 3.10a, compared to the deterministic manifestation.

4 Validation Results

This chapter presents the results from the virtual-based LKA type approval. The examples from the previous methodology chapter have preempted selected result figures for practical illustrations and because the initial scenario algorithms require specific data sets. These figures will not be shown here again to avoid duplicates. However, the text will summarize the relevant information so that these figures are not necessary for the further understanding. Many publications select a validation method without being fully aware of the assumptions and consequences. A few go beyond this by systematically describing their selection procedure. Similarly, we build the development of our validation methodology on requirements. This is already an improvement, but it does not yet validate the validation methodology itself based on data as the strongest form of evidence. We close this gap to support the abstract requirements, before we get to the actual application of the validation methodology. We summarize the entire process as follows:

1. **Configuration of the framework based on a systematic selection procedure:**

   We addressed the first step in Chapter 3. It contains a variety of options for users from several engineering fields according to the building block principle. We selected two overall framework manifestations with the deterministic and non-deterministic one, and configured each framework block for both of them.

2. **Validation of the configured framework itself based on data:**

   Chapter 4.1 targets the second step by validating both framework configurations. The term validation refers in this sense to the validation of a methodology during the scientific process and not to model validation. Nevertheless, both have the same origins in the general validation of scientific theories. This is only relevant in Chapter 4.1 and is linguistically emphasized so that the reference of the validation to the models or the validation methodology itself is directly evident. In contrast to the actual framework application in the fourth step, the framework validation is purely done in simulation and does not yet contain real experiments. From a process point of view, however, it goes beyond the framework application, since the results will additionally be compared with ground truth. Since the second step comes first and we want to avoid repetition, we briefly address the actual application in Chapter 4.1 and focus on the additional validation aspects. If a reader is only interested in the application, he can skip this insertion.

3. **Final selection of the best performing framework configuration:**

   At the transition from Chapter 4.1 to Chapter 4.2, there will be a final selection between the deterministic and non-deterministic configuration based on the results. This always occurs if more than one configuration was initially selected in the first step.

4. **Application of the framework configuration to the actual type approval:**

   Chapter 4.2 continues with the fourth step by applying the final framework configuration to the actual type-approval use case including real and virtual tests.

4.1 Validation of the Methodology via the Method of Manufactured Universes

This section concentrates on the validation of the VV&UQ methodology itself. It begins with a brief excursion into the literature to introduce the Method of Manufactured Universes (MMU). The latter forms the basis for the validation of the VV&UQ framework. Then, this section extends and implements the MMU approach for the R-79 use case. Finally, it presents the type approval results and their validation results using MMU.

4.1.1 Introduction into the Method of Manufactured Universes

In code verification, the aim is to demonstrate that the code does not contain software bugs and that it approaches the exact mathematical solution. To analyze and compare verification methods, the research community has recognized decades ago that benchmark solutions are pivotal. They should be complex but still have an exact analytical solution to precisely quantify the discretization errors. Therefore, the verification community developed the Method of Manufactured Solutions [372] as a procedure to automatically generate the benchmark solutions for robust code verification. Inspired by it, Stripling et al. [373] introduces MMU as an analogue for VV&UQ. The idea is to create a manufactured universe that imitates reality and serves as an environment for benchmarking. In contrast to reality, it is fully controllable, can be scaled to many simulations without enormous effort and costs, and the true values are known. This are major requirements for validation of VV&UQ methods. Thus, the actual simulation model is compared against the manufactured universe as a reference model so that the modeling errors are precisely known. In contrast to the Method of Manufactured Solutions, where the focus is on mathematics and the benchmark solutions do not require any physical meaning, the manufactured universe should closely imitate reality as the focus of model validation. The underlying physics, the modeling errors, and the experimental uncertainties should be as representative as possible so that the findings within the universe are also valid for reality.

Stripling et al. [373] demonstrate MMU by means of a particle transport universe. They have a low-fidelity model for their application and a high-fidelity model as a reference. They vary the modeling parameters and perform multiple comparisons. Whiting et al. [374] apply MMU in a fluid dynamics universe to evaluate four calibration and validation approaches. They compare PBA with AVM [18], PBA with MAVM [354], a Bayesian approach by Kennedy and O'Hagan [365], and the V&V 20 standard [187] from the fluid dynamics and heat transfer community. To judge the VV&UQ results under various validation and prediction conditions, they define the two measures conservativeness and tightness. The former indicates the frequency with which the true value lies within the uncertainty bound. The latter depends on the ratio between the true error and the uncertainty. They propose two separate definitions for the tightness of the validation and calibration methods and weight the conservativness and tightness to an overall score. Lastly, they state that PBA with MAVM performs particularly well under sparse data, whereas the Bayesian approach excels under big data.

4.1.2 Binary Classification of Type-Approval Decisions

While the Method of Manufactured Solutions is commonly used in verification, MMU is underrepresented in validation. The majority of publications simply rely on a particular VV&UQ method to validate a simulation model without validating the method itself. The original introduction of

MMU [373] is a promising starting point that has been extended by Whiting et al. [374] to include the evaluation measures conservativeness and tightness for comparing VV&UQ approaches. We further develop it and embed it in the VV&UQ framework. The contribution of this thesis is twofold. With respect to the creation of the manufactured universe, we do not use a space-filling design for the model parameters, but intentionally inject modeling errors that are characteristic of the use case and must be identified by the VV&UQ method. This is demonstrated in the following subsection using the LKA type approval. With respect to the evaluation measures, we propose to perform the comparisons in several stages along the VV&UQ process. The resulting framework is shown in Figure 4.1. It revives the unavailable system block in the application domain from Figure 3.1 and Figure B.1 using the manufactured universe. Therefore, the true values, indicated by the orange arrows, are available during the process steps.

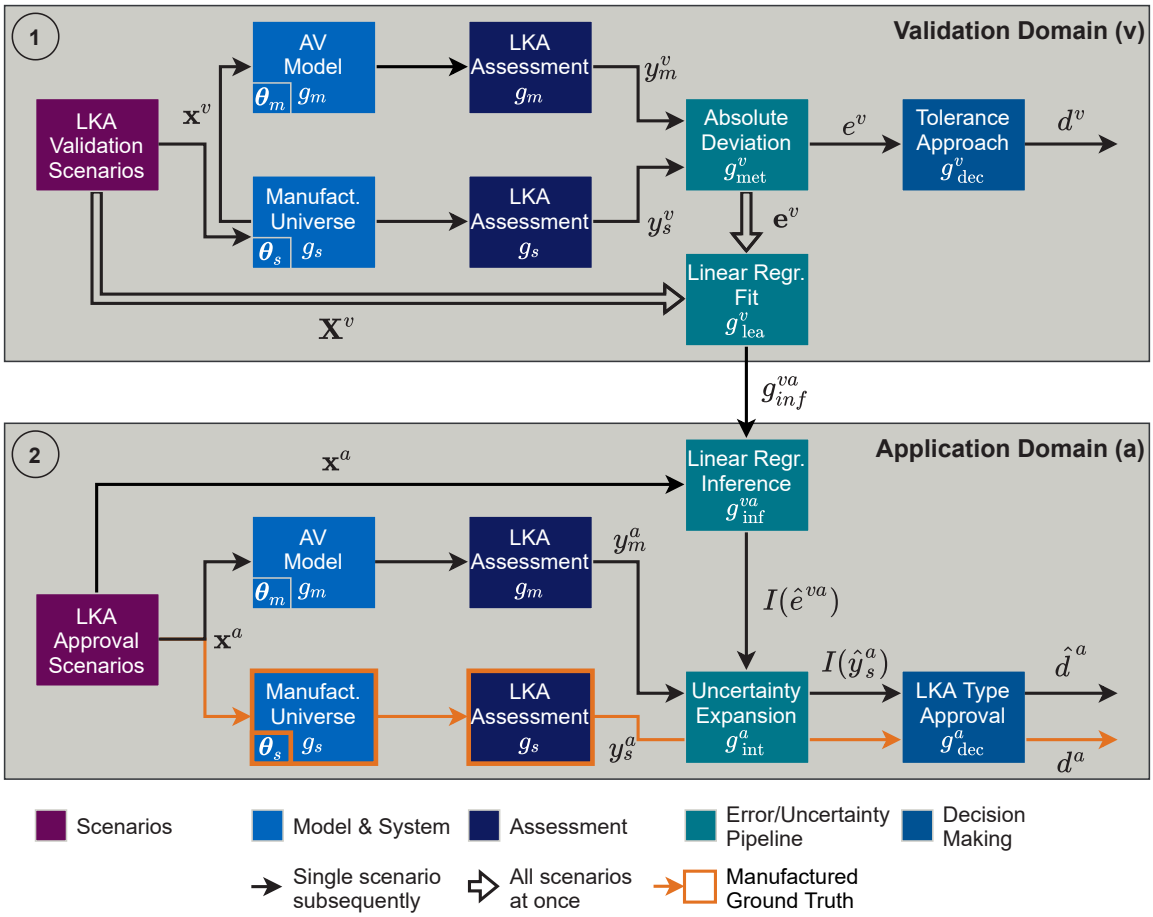


Figure 4.1: VV&UQ framework in MMU configuration for LKA type approval. It shows the non-deterministic manifestation based on the deterministic one from [22, Fig. 4]. They differ in the validation metric and the mathematical structures of the KPI y and error e. The orange ground truth is made available by the manufactured universe in analogy to [21, Fig. 13].

This yields the following four stages for comparison, each for the deterministic and non-deterministic manifestation, if they differ:

1. Error inference stage: $I(\hat{e}^{va})$ vs. e^{va}
2. Assessment stage: y_m^a vs. y_s^a or rather $B(y_m^a)$ vs. $B(y_s^a)$
3. Error integration stage: $I(\hat{y}_s^a)$ vs. y_s^a or rather $B(\hat{y}_s^a)$ vs. $B(y_s^a)$
4. Type-approval stage: $\hat{d}^a$ vs. d^a

The two measures conservativeness and tightness from [374] can be classified into the error inference stage. We concentrate in this thesis primarily on the last stage, since it contains the influences of the previous ones and is ultimately decisive. We accompany it with the penultimate stage to gain additional insight into the location of the expanded uncertainties independent of the regulation threshold. Starting from the type-approval stage backwards, we compare the final type-approval decisions. Since the decisions are binary values, we introduce a binary classifier as often used in object detection or medical drug studies. It distinguishes the four combinations True Positive (TP), True Negative (TN), FP, and FN. The decision of the nominal simulation model affects the terms positive and negative and the decision of the manufactured universe as system imitation the terms true and false. A TP represents a correctly failed type approval, an FP an incorrectly failed type approval, an FN an incorrectly passed type approval, and a TN a correctly passed type approval. This ensures that FPs are Type I errors that can be illustrated by the conviction of an innocent (AV) and that FNs are Type II errors than can be illustrated by the acquittal of a criminal (unsafe AV). The four combinations within the cells of a confusion matrix can be partially combined to overall measures such as the precision P and recall R

$$P = \frac{\sum \mathrm{TP}}{\sum \mathrm{TP} + \sum \mathrm{FP}} \quad \text{and} \quad R = \frac{\sum \mathrm{TP}}{\sum \mathrm{TP} + \sum \mathrm{FN}}. \tag{4.1}$$

The recall rate R defines the ratio of all correctly failed type approvals to all failed type approvals according to the manufactured universe as ground truth. It is therefore particularly important from the safety perspective, as it states how many of the actual fails are detected. The precision rate P defines the ratio of all correctly failed type approvals to all failed type approvals according to the simulation model. It is relevant to avoid many Type I errors. At the end, there is a trade-off between Type I and Type II errors and between precision and recall. Suppose we widen the uncertainty bounds to increase the conservativeness. This reduces Type II errors and improves the recall rate on the cost of Type I errors and the precision. In the case of type approval, the Type II errors and the recall rate are more important from a safety perspective, since not detecting fails can potentially lead to accidents after market launch.

At the error integration stage, we check whether the true values lie within the uncertainty bounds. This is similar to the conservativeness measure from [374], but with the responses instead of the errors. It yields for the deterministic and non-deterministic case the measures

$$C_{\mathrm{abs}} = \sum \quad y_s^a \quad \in I(\hat{y}_s^a), \qquad C_{\mathrm{rel}} = C_{\mathrm{abs}}/N^a, \tag{4.2}$$

$$C_{\mathrm{abs}} = \sum F(y_s^a) \in F(\hat{y}_s^a), \qquad C_{\mathrm{rel}} = C_{\mathrm{abs}}/N^a. \tag{4.3}$$

It can either be given as the absolute amount of bounded scenarios or as a percentage relative to the total number of application scenarios N^a. These measures provide further insights about the quality of the error model, since the type-approval decision can be correct, despite the true value not falling within the uncertainty bounds. This occurs if there is a sufficiently large distance to the regulation threshold so that the distance from the true value to the uncertainty bounds does not matter yet. In this case, the quality of the error model is of less importance, but these measures can still increase the overall reliability of the error model.

4.1.3 Creation of the Manufactured Universe

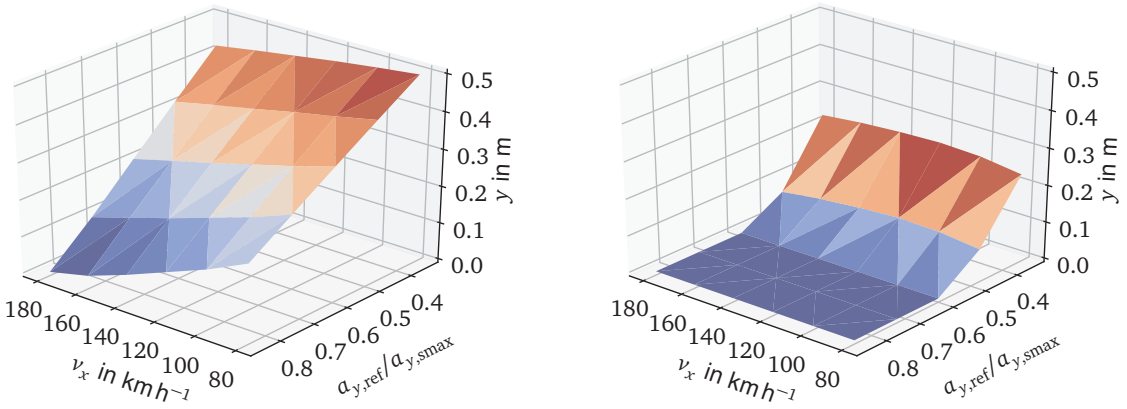
In complex systems, there are an infinite number of possible modeling errors. The differential equation might have the wrong order, important parameters might have been neglected, or the

model parameters might have wrong values. Since vehicle models in common simulation tools often have more than a hundred parameters, and each of them can differ from the true value by different orders of magnitude, there are an infinite number of combinations. For sure, we cannot validate VV&UQ methods under every conceivable constellation. Instead, we need to focus on key features that are of particular relevance to the use case. Therefore, we shift the attention from model parameters to responses. From an abstract point of view, either the nominal simulation model behaves more safely with a higher distance to line than the physical system, or vice versa, or both behave approximately equally safely. The former has the highest relevance from a safety perspective, since the car manufacturer believes in the safety of his system based on the simulation results, but it would actually cause accidents in the real world. If the model and system behave exactly the same, the model would be perfect and there would be no modeling errors that the VV&UQ method could detect. In the reverse case, there are modeling errors again, but the simulation model is a conservative estimate of the real system. If the simulation model would still pass the type approval, there is a high probability that the system would do so. This leaves performance potential unused, but it is nowhere near as safety critical as the first case. Therefore, we focus on creating a manufactured universe that behaves less safe than the simulation model. This constellation and further ones will be discussed later in Chapter 4.1.6.

We have a parameterized vehicle dynamics model of a sports car, a proportional-integral controller for lane keeping, and an ideal sensor model perceiving the virtual environment in the simulation tool. We use this setup for both the nominal simulation model and the manufactured universe as ground truth. We achieve the desired behavior of the manufactured universe driving less safe than the nominal simulation by injecting a modeling error into the vehicle mass parameter, since it is a central parameter with a global influence. The nominal simulation model has a weight of $1377\,\mathrm{kg}$, whereas the manufactured universe has a weight of $1577\,\mathrm{kg}$. This systematic error is a realistic failure case where the simulation model does only contain the actual vehicle weight but not the additional one from the loading. The value of $200\,\mathrm{kg}$ corresponds roughly to the weight of the test driver, an instructor on the passenger seat, and measurement equipment. This modeling errors generate the desired constellation with some failed cases in the manufactured universe but none in the nominal simulation. The resulting relative behavior is shown in the two surface plots in Figure 4.2. It plots the minimum distance to line across the scenario space of the velocity and lateral acceleration for both the nominal simulation model and the manufactured universe. The surface of the manufactured universe is flatter and contains a plateau of zeros. This means that the manufactured universe fails in some of the application scenarios and passes in the others, whereas the nominal simulation model passes in all scenarios. This is exactly the safety-critical constellation where the simulation model erroneously suggests a safe system behavior. The two surfaces meet the expectations in terms of lane keeping. Increasing the lateral acceleration or increasing the velocities for constant curve radii leads to smaller distance to lines, since the vehicle is pushed outwards during cornering. The higher vehicle mass results in a higher centripetal force and ultimately in smaller distance to lines of the manufactured universe. Lastly, we do not add random influences like noise, since they distort the true values and would be counterproductive at this point.

4.1.4 Approval Results of the (Non-)Deterministic Manifestation

In this subsection, we walk through the results along the entire framework process from Figure 4.1. It can be seen as the analogue to the framework application in Chapter 4.2, but for the purely virtual MMU setup instead of the actual one including real and virtual tests. As mentioned earlier, we will keep the framework application within a short subsection here, since it will



(a) Simulation model (b) Manufactured universe

Figure 4.2: Minimum distance to line y across the scenario space based on [22, Fig. 3]. The surfaces refer to the application scenarios of velocity and acceleration for a fixed wind speed of $-5\,\mathrm{km\,h^{-1}}$, tank load of $-20\,\mathrm{kg}$, and road slope of $-1°$. The simulation vehicle drives in all scenarios within its lane, while the manufactured vehicle crosses the line in individual scenarios. The corresponding distances are zero by definition (like a crash barrier), since this has implementation advantages over negative values without affecting decision making.

be the focus of the entire Chapter 4.2. Thus, the purpose of this subsection is to compactly summarize the results of each framework block to provide brief insights. The methodology of each block was described extensively in Chapter 3 and illustrated with exemplary figures. For the interested reader who wants deeper insights at this point, we provide the figure references within each summary so that they can be traced back. However, they are not required for the further understanding. Only the final type-approval results form the basis to validate the framework by means of the binary classifier in the next subsection. The following list contains the individual block summaries for both the deterministic and non-deterministic manifestation:

Scenario Design: The scenario design hyper-parameters were given in Table 3.4 and resulted in the point distribution in Figure 3.5, where the system represents the manufactured universe. The distribution follows a simple full-factorial grid directly used for the deterministic manifestation. It can indirectly be seen in the final type approval plot of this subsection in Figure 4.3 by interpreting the rectangles as validation scenarios and the union of circles and crosses as application scenarios. The nominal grid is extended by the sampling of uncertainties in local neighborhoods for the non-deterministic manifestation. It ensures the same baseline for a fair comparison between both manifestations.

Assessment: The assessment of an exemplary scenario can be found in [22, Fig. 6], similar to the time series plot in Figure 3.6. The minimum value of the distance to line appears at the entrance of the curve and is corrected afterwards by the LKA. The two assessment surfaces across the entire scenario space are shown in Figure 4.2 for the deterministic simulation and the corresponding manufactured universe. The non-deterministic simulation propagates the local uncertainties leading to scatter around the surface plots with more extreme values. This yields a few non-deterministic assessment results for high velocities and accelerations were the distance to line is zero for the nominal simulation model.

Validation Metric: The absolute deviation is used as deterministic validation metric. It can be imagined as the difference between the two assessment surfaces at the location of the validation scenarios. The resulting deviations were included as orange points in Figure 3.8. They have a positive sign, since the surface of the nominal simulation model

lies above the one of the manufactured universe. The distribution of errors across the scenario space shows higher errors for higher accelerations and lower velocities. This information is of particular interest for the model developer to guide improvements. In the non-deterministic case, we used an area metric in form of the AAVM. An illustration can be found in [22, Fig. 7a] for an exemplary scenario, similar to Figure 3.7 but extended to p-boxes. The modeling errors manifest themselves in the left area metric, since the manufactured universe has smaller distances to line than the nominal simulation model. The resulting distribution of left areas looks similar to the deterministic case, since both manifestations are dominated by the injected systematic error.

Tolerance Approach: We can optionally apply tolerances for validation decision making. However, it is challenging to select meaningful tolerance values for the lateral driving behavior of an AV. A fixed value of 0.1 m has a much stronger effect if the vehicle comes within 0.01 m of the lane marking compared to if it keeps 1 m distance. This can be compensated to some extent by more sophisticated types of tolerances [22, Sec. 6.2]. We integrated the tolerances in the VV&UQ framework as an additional option for the model developer. However, as described later in the discussion, the tolerance approach must be used with with particular caution. Figure 4.3 demonstrates the dangers of the tolerance approach, since they can easily lead to contradictions without the developer knowing it.

Error Learning and Inference: We learned an error model by means of linear regression with external PIs for both the deterministic errors and the left area metric. The regression surface and the lower and upper prediction bounds were shown in Figure 3.8 for the deterministic manifestation. The regression surface matches the metric results as its training data well. This is quantitatively confirmed by a small mean squared error within the PI calculation. The prediction surfaces are almost planar and bound the metric results of all validation scenarios. The same holds true for the non-deterministic manifestation.

Uncertainty Expansion: The expansion of uncertainties was illustrated in Figure 3.10 for an exemplary application scenario of both manifestations. The uncertainty shifts the nominal results as expected to the left to compensate for the modeling errors. In these two cases, the outer bounds of the results are still larger than the regulation threshold of zero.

Type Approval: Therefore, both cases count as passed test scenarios. Performing the decision making for all application scenarios yields 123 passed cases and 117 failed cases for the deterministic manifestation, as well as 97 passed cases and 143 failed cases for the non-deterministic manifestation. The amount of passed cases is lower for the non-deterministic manifestation due to the additional consideration of input uncertainties. The relation of these VV&UQ decisions to the true universe decisions follows in the next subsection.

4.1.5 Classification Results of the (Non-)Deterministic Manifestation

The previous subsection concluded with the type-approval decisions of the VV&UQ method including uncertainty bounds. This is the end of the normal virtual-based process. However, since we are interested in the validation of the VV&UQ method itself at this point, the question arises what these decisions say about the quality of the methodology. To set them into relation and to answer this question, the actual ground truth values from the manufactured universe are required. The uncertainty expansion example in Figure 3.10 included not only the nominal simulation results and the uncertainty bounds but also the true values. For both the deterministic and

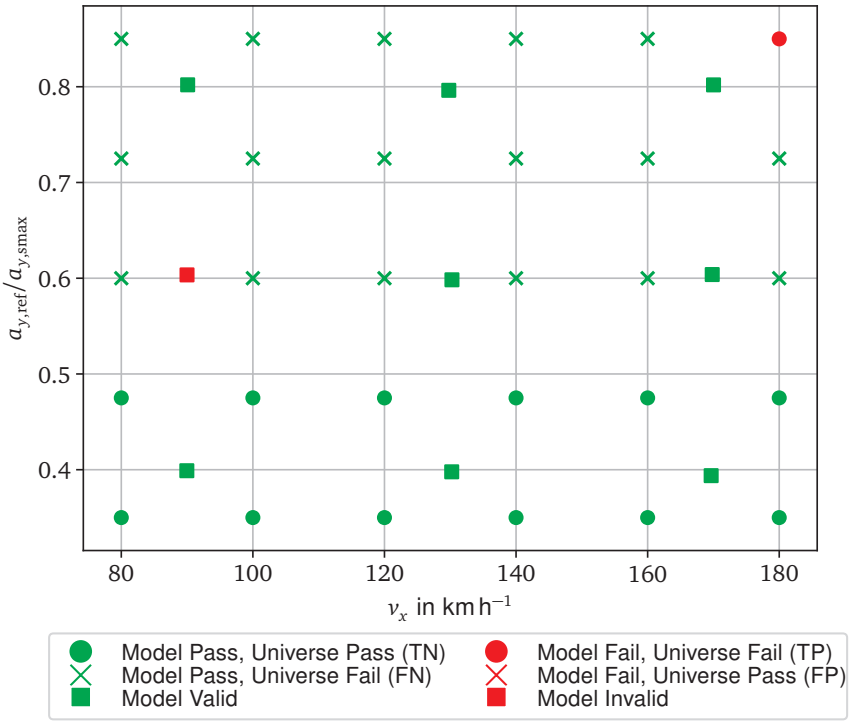


Figure 4.3: Dangers of the tolerance approach based on [22, Fig. 9]. The rectangular validation decisions originate from the tolerance approach. The application decisions of circles and crosses relate the non-deterministic nominal model and the true universe. We selected an exemplary tolerance of $0.3\,\mathrm{m}$ to demonstrate how easily multiple contradictions arise. A green rectangle claiming a valid model contradicts each cross in the neighborhood, since a deviation between the model and universe decision indicates an invalid model.

non-deterministic manifestation, the uncertainty bounds enclosed the true values. If we extend the analysis to all 240 application scenarios, we obtain 238 of these bounded cases for the deterministic manifestation and even 240 bounded cases for the non-deterministic manifestation. This corresponds to a conservativeness of $C = 99.17\,\%$ and $C = 100\,\%$, respectively. It indicates that the linear regression estimate with PIs is wide enough to cover the true values.

At the type-approval stage, the binary classifier can be applied twice to relate both the nominal simulation results and the uncertainty bounds of the VV&UQ method to the true values. The former represents the initial situation generated by the design of the manufactured universe. The latter represents the actual validation of the VV&UQ framework. Nevertheless, the latter has to be seen in relation to the former as its starting point. Therefore, Table 4.1 summarizes the binary classification results of the nominal simulation model in the first row and of the VV&UQ method in the second row, each for the deterministic and non-deterministic manifestation. The initial situation is characterized by a majority of TNs but accompanied by a significant amount of FNs. The nominal model fails only in 2 or rather 5 application scenarios. However, 88 or rather 92 of the passed cases from the simulation would actually fail in the manufactured universe. This corresponds to a recall rate of $2\,\%$ or rather $5\,\%$ and to a precision of $100\,\%$. They result from the large systematic error injected into the manufactured universe. The low recall rates are particularly safety critical and must be avoided by the VV&UQ method.

The VV&UQ method successfully removes all FNs and transforms them into TPs, despite their challenging numbers. This means that the uncertainty bounds are always large enough to shift the nominal simulation over the regulation threshold of zero to the true values. This completely avoids Type II errors. However, it comes at the cost of a few Type I errors by transforming some

TNs into FPs. At the end, the recall rate is $100\,\%$ and the precision $77\,\%$ or rather $86\,\%$. Thus, the VV&UQ method increases the recall rate from a small single-digit number to a perfect $100\,\%$. This means that no unsafe AV is released to the market anymore. In return, the VV&UQ method decreases the precision from $100\,\%$ to $77\,\%$ or rather $86\,\%$. This leaves potential unused, but it is hardly completely avoidable with such a large systematic error.

Table 4.1: Binary classification results from [22, Tab. 3].

(a) Deterministic results — Nominal model

		Universe		
		Fail	Pass	
Model	Fail	2 TPs	0 FPs	$P = 100\,\%$
	Pass	88 FNs	150 TNs	
		$R \approx 2\,\%$		

(b) Non-deterministic results — Nominal model

		Universe		
		Fail	Pass	
Model	Fail	5 TPs	0 FPs	$P = 100\,\%$
	Pass	92 FNs	143 TNs	
		$R \approx 5\,\%$		

(c) Deterministic results — VV&UQ method

		Universe		
		Fail	Pass	
VV&UQ	Fail	90 TPs	27 FPs	$P \approx 77\,\%$
	Pass	0 FNs	123 TNs	
		$R = 100\,\%$		

(d) Non-deterministic results — VV&UQ method

		Universe		
		Fail	Pass	
VV&UQ	Fail	123 TPs	20 FPs	$P \approx 86\,\%$
	Pass	0 FNs	97 TNs	
		$R = 100\,\%$		

4.1.6 Discussion of the Classification Results

This subsection discusses the classification results as the basis for the final selection of the deterministic or non-deterministic manifestation at the beginning of the next section. The overall discussion follows in the separate Chapter 5. Both manifestations show similar classification rates. The non-deterministic manifestation has a slightly better precision of $86\,\%$ over $77\,\%$ at the same recall of $100\,\%$. The perfect recall rate provides high confidence for decision makers, since it states that all safety-critical FNs were successfully recognized and corrected. The precision rates are meaningful for such a large systematic error and provide arguments to the decision maker against the model developer. In order to better understand and discuss these numbers, we have to take into account the creation of the manufactured universe. It goes back to Chapter 4.1.3 and to the uncertainties in Table 3.4 in the non-deterministic case. All values were systematically selected in advance of this study, but they still influence the results.

The large systematic error is the most important constellation from the safety perspective. However, it decreases the relative importance of both the true and quantified input uncertainties. The true ones directly affect the manufactured universe of both manifestations as the common ground truth. However, they hardly affect the deterministic simulation due to its averaging of inputs, whereas they are quantified and propagated by the non-deterministic simulation. Thus, large true input uncertainties penalize the deterministic manifestation, since the averaging cannot cover them. In return, they can be a preference or disadvantage for the non-deterministic manifestation, depending on how precisely the uncertainties were quantified. If the precision is low due to practical issues, the quantified input uncertainties inflate leading to more FPs and a smaller precision. In our universe, we have a large systematic error, relatively small true input uncertainties, and a perfect precision so that the true and quantified input uncertainties coincide. While the second aspect favors the deterministic manifestation, the third aspect favors

the non-deterministic one. However, the first aspect reduces the significance of the other two. In this constellation, the high classifier rates show the validity of both manifestations alike.

Therefore, the user has the possibility to include further factors beyond the classifier results in the final selection of the framework configuration. The non-deterministic manifestation considers multiple uncertainty sources separately, covers more scenario space due to the local point clouds, and has a higher and adjustable confidence. In return, it relies on a precise UQ and requires more effort and budget. Thus, if the resources are available and input uncertainties matter, the non-deterministic manifestation is recommended. Otherwise, the deterministic manifestation represents a good trade-off between confidence and effort. The analysis can be extended to further constellations in the future by generating more universes or, to a certain extent, by transferring the available results. For example, a reduction of the safety-critical systematic error would yield less FNs for the VV&UQ method to detect, but it would give more weight to the precision of the input uncertainties.

4.2 Application of the Validation Methodology Based on Real Driving Tests

The previous section presented and discussed the results from the MMU study to validate the VV&UQ framework itself. It addressed both the deterministic and the non-deterministic manifestation. This section continues with the final selection between the two considering the available test environments. The selected one will then be applied to the actual type approval. We will go step-by-step through the results from the entire type-approval process including physical experiments in the field, re-simulations, and new model predictions.

4.2.1 Final Framework Configuration

For the PoC of this section, we have a prototype vehicle available for field tests and a hybrid simulation environment that contains both the corresponding models and hardware components. It initially goes back to the methodology in Chapter 3.2.6. Whereas the MMU study in Chapter 4.1 relies per definition on the comparison of two virtual environments, the actual PoC has to compare either a virtual or hybrid environment with a real one. They are, however, part of the system under test and do not fundamentally affect the validation methodology. We have a vehicle and hybrid environment available in this thesis that must be seen as given from the car manufacturer for the purpose of homologation and only limited knowledge is provided to the technical service according to our requirements. The vehicle is complex, contains black-box components, and drives in a complex traffic environment where many conditions are unknown. We have an Inertial Measurement Unit and geo-referenced road maps to measure the velocity and acceleration input of R-79 and the distance to line output. However, we have no data about scenario parameters beyond R-79 such as weather conditions or other road users. The available hybrid environment has to run in real-time due to the hardware components and requires a transition time between the scenarios to ensure the laws of physics and to avoid errors in the control units. Therefore, it can not be accelerated by changing the speed of a pure simulation or by using a computing cluster. The hybrid environment is currently under development and not yet at a final stage of maturity. Thus, it will probably be dominated by systematic errors. However, this thesis concentrates on the development of a validation methodology and not on the development of a simulation environment or of an AV. They should be viewed as the system

under test. Therefore, the maturity of the simulation does not negatively affect the PoC in any way. On the contrary, it is interesting for the validation methodology to identify modeling errors. These can then be sent back to the developers for improvement.

From experience, there is likely to be both input errors and a systematic model-form error, with the latter dominating. We can verify the correctness of this assumption as we proceed. The systematic errors fit the previous MMU study, so we can build on its main finding. It states that both the deterministic and non-deterministic manifestation excel with high classification rates in the same order of magnitude, so that other factors can be included in the selection process. We finally choose a hybrid manifestation due to the intrinsic non-determinism of the hybrid environment. This is convenient, since it can be interpreted as a mixture of the deterministic and non-deterministic manifestation according to Chapter 3.2.6 and includes properties from both worlds. On the one hand, it takes into account a total deterministic error and accordingly leads to the desired balance between low effort and yet high confidence from the uncertainty bounds. On the other hand, it can exploit synergies to the non-deterministic manifestation in the framework configuration. For example, we can reuse the area metric to consider the scatter from a few repetitions of the hybrid simulation. The full non-deterministic manifestation is hardly applicable for this PoC due to two limiting factors. First, the lack of data from the complex traffic environment makes a precise UQ impossible. Second, the nested uncertainty sampling generates thousands of samples for a non-deterministic simulation. The hybrid environment is not capable of running them within a reasonable time. We will later discuss the challenges that need to be solved to enable its real application. The final framework configuration is summarized in Table 4.2 and Figure 4.4. The theory of each block was introduced in the methodology chapter and the associated results will follow in the remaining subsections.

Table 4.2: Selected domains, manifestations, and VV&UQ approaches for the final PoC.

Domains	Hierarchical	Formal	Time-(in)variant	(Non-)deterministic	VV&UQ
Validation & Application	Entire system	–	Time-invariant	Hybrid	–

4.2.2 Coverage-Based Validation Scenarios

There are certain requirements for the physical experiments that have to be considered during the design of the validation scenarios. First and foremost, the tests have to be executed in the field to cover a wide variety of elongated curve radii with lane markings on both sides. No proving ground in the world can offer this. Furthermore, the road information of the experiments has to be precisely known to enable an accurate re-simulation in the virtual world. The quality of the experiments is particularly important for the category of validation experiments. To fulfill both requirements, we use measured roads that are available in a road map format supported by most simulation tools. During this thesis, measured roads were available in the region around the German city Kempten. The measurements were carried out in advance by an external company. They use multiple runs with a vehicle equipped with camera and lidar sensors and process them semi-automatically with algorithms and manual corrections. The resulting road maps have an accuracy in the range of a few centimeters, are geo-referenced with GPS coordinates, and are available in the OpenDrive file format. The latter is standardized by the ASAM organization, pushed forward by the automotive community in research projects such as PEGASUS, and supported by most simulation tools. The road maps cover segments of the German highway A7 and the rural road B19 in the Kempten region.

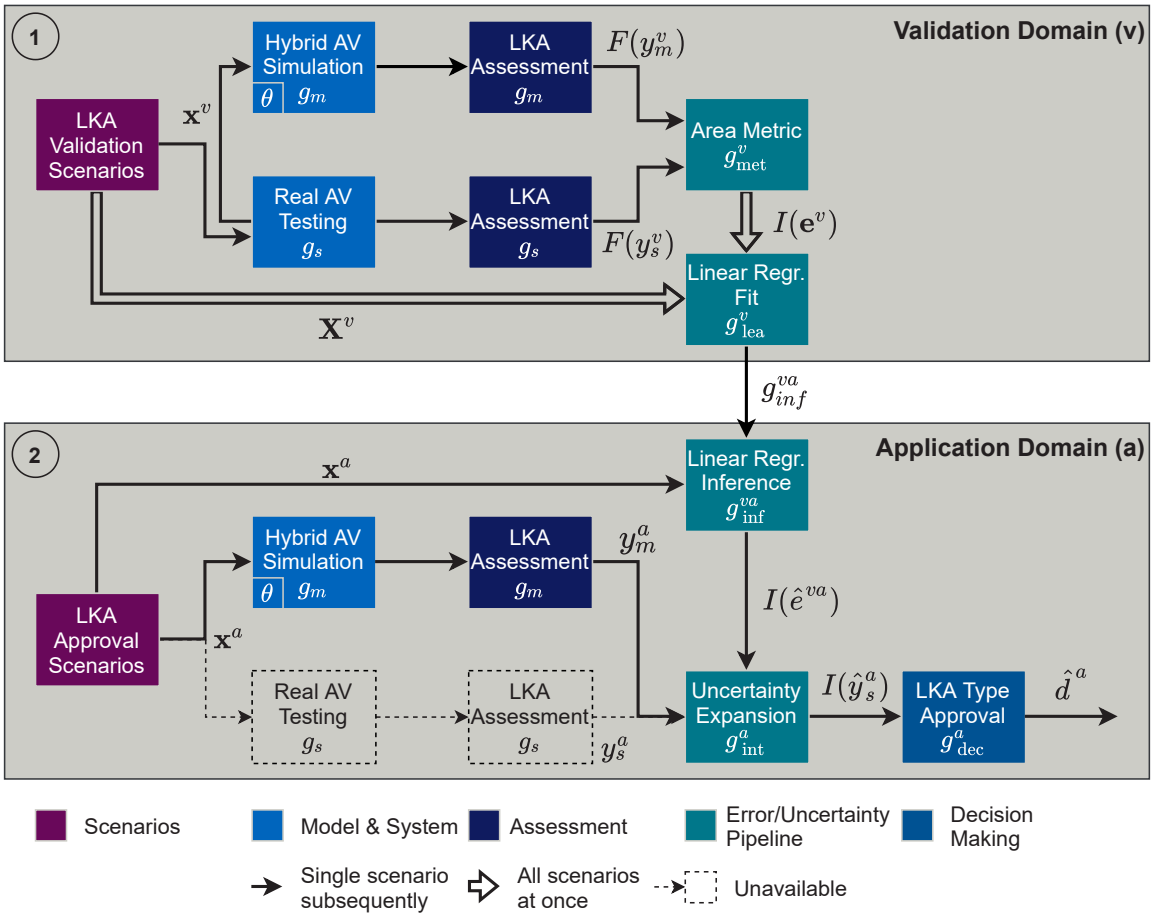


Figure 4.4: VV&UQ framework for actual LKA type approval based on [23, Fig. 2].

The described road maps were used by the coverage-based algorithm from Chapter 3.3.3. In addition, it obtains the hyper-parameter $a_{y,\text{smax}} = 2.5\,\text{m}\,\text{s}^{-2}$ from the vehicle specification to calculate the normalized reference acceleration from R-79. The focus with respect to the scenario coverage was set on the range from 30 % to 90 % of $a_{y,\text{smax}}$. This includes the range from 80 % to 90 % of $a_{y,\text{smax}}$ as upper bound, which the regulation emphasizes from a safety-critical point of view. The low lateral accelerations were omitted because the experiments are associated with enormous effort and the lower accelerations lead to easier driving situations similar to a straight line. Nevertheless, the application scenarios can go beyond this range, since the error model in the framework considers extrapolation uncertainties. The resulting test plan was initially shown in Figure 3.4 and can be seen indirectly in superimposed figures of the remaining section. It represents a good trade-off between effort and scenario coverage for a PoC.

4.2.3 Validation Experiments on the Real Road

The selected curve radii are distributed across the entire road map. They were transferred into an efficient plan for the test driver and were approached several times for a statistical analysis. The intention was to repeat each scenario at least three times. Nevertheless, if a curve is conveniently located on the map and practically has to be driven off anyway to get to a more distant curve, significantly more repetitions were performed on this one. This corresponds to the recommendations given by Viehof [197] based on the t-distribution to repeat most tests at least three times and individual ones between ten and fifteen times.

The vehicle was equipped with a high-precision Inertial Measurement Unit that combines D-GPS localization with gyroscope sensors. It excels with an accuracy of a few centimeters and provides reference measurements that are independent of the in-vehicle sensors and thus of particular importance for validation experiments. It directly provides the velocity and lateral acceleration signals. The radius and distance to line information is obtained by localizing its GPS coordinates within the geo-referenced road maps. An exemplary projection of the vehicle trajectory onto a OpenStreeMap of the German highway A7 is shown in Figure 4.5. Details about the distance to line calculation including equidistant interpolation can be found in [375]. The combination of the radius and velocity yields the reference lateral acceleration according to Equation (3.24). In contrast to the measured lateral acceleration, it does not depend on the actual vehicle trajectory but on the curve as scenario input. Therefore, both the input quantities and the output quantities of R-79 can be determined from data of the Inertial Measurement Unit.

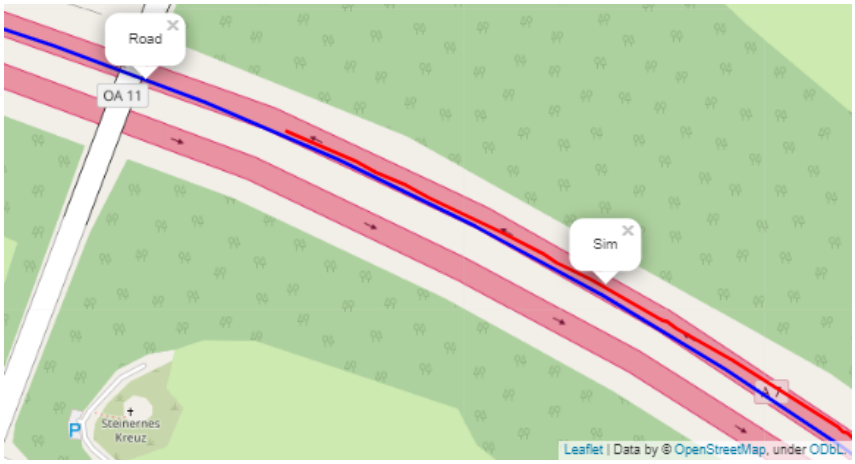


Figure 4.5: Exemplary projection of a real and virtual vehicle trajectory from the same scenario onto a OpenStreetMap from [23, Fig. 5].

After performing the experiments, we build on the event finder from Chapter 3.3.2 to check in post-processing whether the required conditions from R-79 are satisfied. If they are violated, we discard the respective test. Since we have multiple repetitions of the same concrete scenario, this did only reduce the number of repetitions in single cases, but the 17 planned validation scenarios are preserved with at least two repetitions. If the conditions are satisfied, the event finder does help to automatically localize the planned scenario within a large measurement file.

4.2.4 Re-Simulation of the Validation Experiments

The digital road maps in the OpenDrive format are imported into the simulation tool for a realistic re-simulation. Strictly speaking, we cut the selected curves out of the entire road network and connect them by means of straight lines so that we do not have to drive a longer distance to reach the target curves. This exploits the efficiency advantage of the simulation compared to reality. The new route can then simply be run several times to obtain test repetitions of the hybrid simulation. The respective straight line segment before the actual curve is used to set the target speed from the test plan and as a stable starting position for the LKA. The OpenStreetMap illustration in Figure 4.5 does not only contain an exemplary vehicle trajectory from real-world driving but also the corresponding one from the simulation. Both trajectories end up at the same curve of the A7 highway. This indicates that the localization pipeline works properly. Thus, we have an accurate re-simulation of the velocity by setting it in cruise control, and the appropriate curve radius by locating it on the road map. They determine the reference lateral acceleration

so that we have a re-simulation of both scenario parameters from R-79. In addition, the map provides information about the road such as its inclination. However, we cannot re-simulate influences beyond R-79 such as weather conditions or traffic. Their parasitic effects must be mitigated as part of the error model.

After performing the re-simulations, we can analyze the repeatability of the test conditions within the hybrid environment. We look at the measured and at the reference lateral acceleration. The former depends on the driven trajectory and the latter on the set speed and the road radius. They are all provided by the simulation environment. The reference lateral acceleration does almost coincide across all repetitions of the same scenario, since the radius is fixed and the speed is set precisely. Therefore, Figure 4.6 presents multiple measured accelerations $\mathbf{a}_{y,i}$ from the test repetitions but only one averaged reference acceleration $\langle \mathbf{a}_{y,\mathrm{ref}} \rangle$ as representative. The reference lateral acceleration in Figure 4.6 rises at the curve entrance and stabilizes slightly above $2\,\mathrm{m\,s^{-2}}$. This refers to the R-79 band from $80\,\%$ to $90\,\%$ of $a_{y,\mathrm{smax}} = 2.5\,\mathrm{m\,s^{-2}}$. The measured signals show an overshoot at the curve entrance and some oscillations around the convergence value, since the vehicle trajectory does not exactly coincide with the perfect curve center-line. Nevertheless, they reflect the trend of the reference lateral acceleration well.

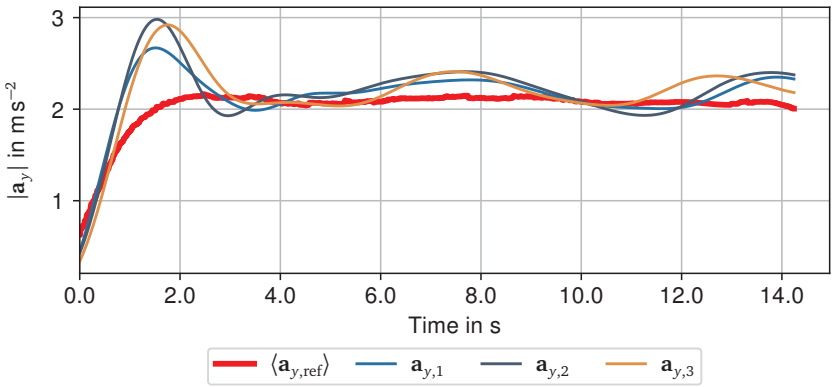


Figure 4.6: Repeatability analysis from [23, Fig. 8a]. The repetitions of the measured acceleration show a similar signal form with oscillations around the smooth reference acceleration signal.

4.2.5 Data-Driven Application Scenarios

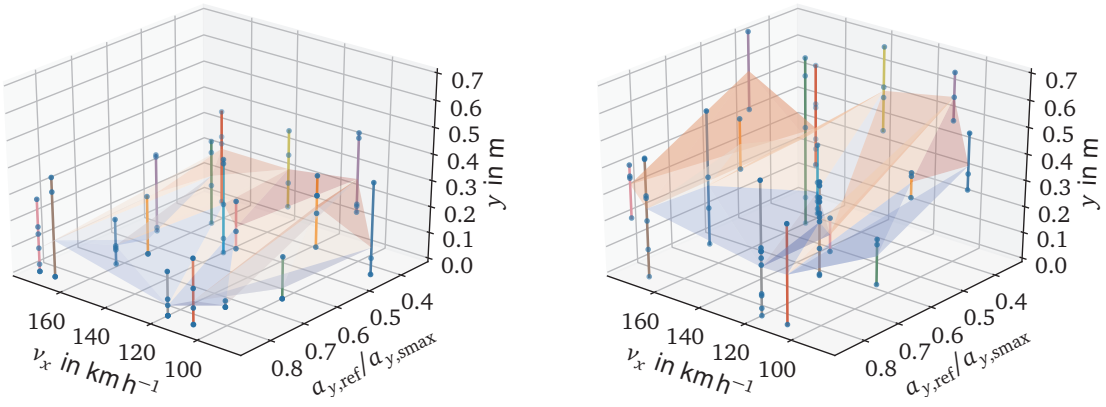
We also use the road network of stacked curves and straight lines from the previous subsection as the basis for model prediction in the application domain. However, we drive it several times at arbitrary speeds, obtaining a random 2D scenario design. The randomness and ultimately the data-driven design was intended for a fair type approval, but it is not the focus of this work. The recorded data set is passed to the event finder of Chapter 3.3.2. It has a total length of $153.86\,\mathrm{km}$ accumulated over all runs. The minimum duration of an extracted event is configured as $4.5\,\mathrm{s}$ to avoid short scenarios where the LKA would hardly need to prove itself. This constitutes a reasonable trade-off between the amount and meaningfulness of the scenarios. The shortest scenario found by the event finder has a duration of $4.55\,\mathrm{s}$ and is thus just above the lower threshold. The longest event has a duration of $40.38\,\mathrm{s}$, which indicates a steady-state cornering through an elongated curve. The average event length is $10.01\,\mathrm{s}$ and lies in the order of magnitude of a typical scenario duration [9].

The event finder extracts an amount of 62 application scenarios in post-processing from the given data set. This corresponds to a frequency of 0.88 events per kilometer. The number of 62 application scenarios is significantly larger compared to the 17 validation scenarios to legitimize

the virtual-based process. The distribution of the scenarios can be found in Figure 3.4. The application scenarios are closer together due to the higher number and are distributed across the entire space. They cover also the range of lower velocities and accelerations saved during the validation experiments due to cost reasons. Therefore, the relationship between validation and application scenarios does not only include interpolation, but also extrapolation constellations. Whereas the validation scenarios were repeated several times as an important characteristic of model validation, the application scenarios were not. This is because of the random design and since we focused the availability of the hybrid environment on the exploration of new application scenarios instead of repeating the same ones. Nevertheless, we cover the scatter of the hybrid environment to some extent by the entirety of all application scenarios, since they are numerous and can come arbitrarily close due to the data-driven design. This is not the best we can do in terms of safety, but it is a sufficient compromise between safety and efficiency for this PoC.

4.2.6 Assessment

Each repetition of each scenario in each test environment and each domain goes through the same assessment. In case of R-79, we calculate the minimum distance to line over time and both sides of the lane. Figure 4.7 presents it in dependence of both scenario parameters. It includes each repetition of each validation scenario of both the hybrid simulation environment and the real world. We select the assessment in the validation domain as representative, since it includes both test environments. Nevertheless, the simulation vehicle behaves similarly in the application domain as in the presented validation domain. Each dot represents one repetition and each vertical line of the same color one distinct scenario with several repetitions.



(a) Simulation (b) Real world

Figure 4.7: Minimum distance to line as a function of the validation scenarios from [23, Fig. 9]. The colors of the averaged surface planes and the vertical lines connecting the repetitions of the same scenario are only selected for differentiation.

We can extend the repeatability analysis from the last subsection from the test conditions of the hybrid environment within one scenario to the assessment results of both the hybrid environment and the real experiments across the entire scenario space. First of all, the position of the dots along the vertical lines provides distributional information. In case of a zero distance to line, multiple dots might coincide at the ground plane. Whereas the dots are spaced roughly equidistantly in some cases, they are grouped at the line ends in other cases. Thus, there is no clear tendency in the distribution so that we focus on the length of the vertical lines. It represents the total uncertainty of one scenario due to its repetitions. There are only isolated scenarios with a short line and ultimately a small repetition uncertainty. Most scenarios have a

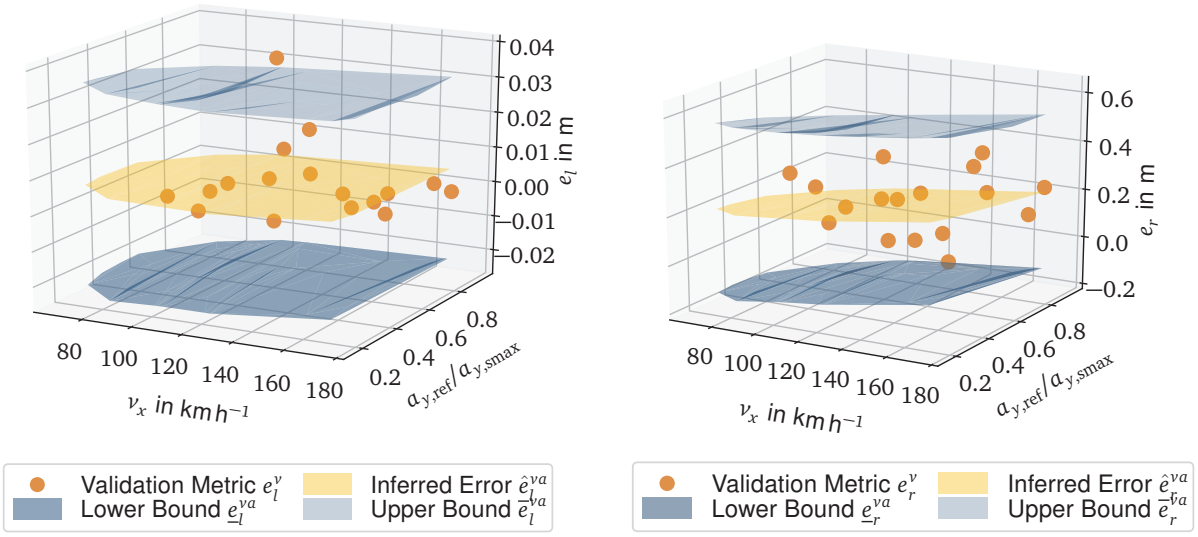
line length or repetition uncertainty in the order of magnitude of 30 cm. For better understanding, this corresponds according to the rule of thumb from Chapter 3.4 to more than 30 % of the maximum available lane area, which is covered only from repeating the same scenario.

A scenario is characterized by the two parameters velocity and reference lateral acceleration from R-79. In addition, there are slightly different start conditions before a curve, since the vehicle cannot always be perfectly centered and due to further scenario parameters beyond R-79 such as side wind that might affect the LKA behavior. Thus, the repetition uncertainty in the real world does not only arise from the intrinsic non-determinism of the system but also from further external conditions. In contrast, the repetition uncertainty in the hybrid environment arises purely from the hardware components, since the scenario conditions are precisely set in the virtual world. When comparing the repetition uncertainty between the simulation and reality, it is in the same order of magnitude. The fact that the repetition uncertainty is comparable, but the reality is additionally influenced by external conditions, suggests that the joint intrinsic non-determinism is more strongly elicited in the simulation environment.

After analyzing the repeatability of the same scenarios, we can analyze the trend of the distance to line across the scenario space. Figure 4.7 shows the trend in the form of a transparent surface that connects the averaged distance to line values of each scenario by means of colored planes. We can see that both planes fall from high velocities and in particular high accelerations towards lower values of the scenario parameters. This suits the typical cornering behavior of a vehicle. The surface gradient is much more pronounced in reality. Its highest distance to line value is 55 cm and the highest dot value of a single repetition 70 cm. Its lowest value is 10 cm and the lowest dot lies at the ground plane. In comparison, the highest value of the simulation surface is 20 cm, the highest dot value 40 cm, the lowest surface value lies slightly above the ground plane and the lowest dot value on the ground plane. Thus, there are many cases where the real vehicle has a large buffer to the lane markings and only individual cases where it crosses the lane. However, the virtual vehicle drives mostly close to the lane edge and crosses it often. In summary, the assessment finds that both test environments have a comparatively large scatter, but the tendency can be recognized anyway, and that both have individual fails, but the real vehicle is in average significantly safer than the virtual vehicle. This dominant systematic error is not optimal, but its direction towards a safer real vehicle is preferable from a safety perspective, since it is more conservative and does not lead to an erroneous trust in the AV.

4.2.7 Validation Metric

The location of the two assessment surfaces relative to each other is reflected in the validation metric results. All repetitions of one scenario are aggregated in form of a CDF. The area metric quantifies the area between the CDF of the hybrid simulation and the CDF of the experiment. It exploits the knowledge gained from all the test repetitions. There is one exceptional case where the simulation CDF and the experimental CDF lie close to each other. This was used earlier in Figure 3.7 for demonstration and ultimately resulted in a left area of 4 cm and a right area of 1 cm. In most cases, however, the simulation CDF lies fully on the left side of the experimental CDF, since its assessment surface is flatter and has smaller distance to line values. This leads to left areas of zero and mostly to right areas in the range from 10 cm to 30 cm. The left and right areas of the 17 validation scenarios are summarized in Figure 4.8 as orange points. Figure 4.8a contains the outlier of the left area at 4 cm, 14 points on the zero plane, and two more points in-between. The right area values in Figure 4.8b are more scattered. The lowest orange point refers to the 1 cm from above, but most points have significantly higher values.



(a) Left area

(b) Right area

Figure 4.8: Validation metric results, error inference via linear regression, and external PIs for both the (a) left areas from [23, Fig. 10] and (b) the right areas. Care should taken with the different z-axis scaling. This is a necessary exception, because otherwise the factor of more than 10 would make it hard to see anything on the left side. Instead, we adjusted the scaling and perspective so that the position of the points and planes to each other is clearly visible.

4.2.8 Error Learning and Inference

The task of the two error models is to represent the left and right area results shown as the orange points in Figure 4.8, respectively. The multiple linear regression technique yields a plane in the 2D scenario space. The left regression plane does almost coincide with the zero plane, since most of the left areas have a value of zero and dominate the training data set. Thus, the left regression plane matches the majority of the orange points well. The outliers can not directly be covered by the regression plane itself due to order of the linear model. Nevertheless, this is intentional to some extent to avoid over-fitting and covered by the PI. The task of the right regression plane is more challenging due to the scattering of the respective orange points. The plane still reflects the trend and passes through the center of the points at about 20 cm.

The task of the PI is to cover the uncertainty of the error model and the one associated with an interpolation or extrapolation from the validation scenarios to an application scenario. The PIs are characterized by a lower and upper interval bound and jointly lead to the lower and upper surfaces in Figure 4.8. The left upper surface lies at about 3 cm and the left lower surface slightly above -3 cm. They enclose all left area results used for training of the linear model with the exception of the outlier at 4 cm. The right upper surface lies at about 50 cm and the right lower surface at about -10 cm. There we can see that the scatter has led to a widening of the prediction uncertainties. A value of 50 cm constitutes more than half of the available domain of the distance to line from the vehicle edges to the lane markings. This value is influenced by the number and distribution of validation scenarios, as well as the magnitude and scattering of the modeling error depending on the simulation quality. In this PoC, the large systematic error is the dominating factor leading to the wide right uncertainty. Both right surfaces enclose in turn the validation results from 16 out of the 17 validation scenarios. The negative values would not be possible as areas, which are mathematically defined positive. Thus, the shift to the bottom could be cut at the zero plane. Nevertheless, this is not of relevance, since we focus on a worst-case consideration and use only the upper bounds for the uncertainty expansion.

4.2.9 Uncertainty Expansion

Executing the application scenarios in the simulation tool yields the nominal simulation results. They are erroneous by definition, since a model is always a simplified abstraction of reality. Therefore, we use the left upper bounds from the previous subsection to shift the nominal results to the left and the right upper bounds to shift them to the right. This uncertainty expansion is shown in Figure 4.9 for an exemplary application scenario. The location of the nominal simulation result differs between the application scenarios. However, the shift to the left and right is similar, since both upper surfaces from Figure 4.8 are close to horizontal. There is a small shift of 3 cm to the left and a large shift of roughly 50 cm to the right. These values match the ones from the previous subsection. In this application scenario, the resulting uncertainty bounds still have positive distance to line values. In scenarios where the nominal result is close enough to zero so that the shift to the left would yield negative values, we can in turn cut it at zero without affecting the decision making. In the extreme case where the virtual vehicle already crosses the lane and has a distance to line of zero, there is no need for a shift to the left anymore.

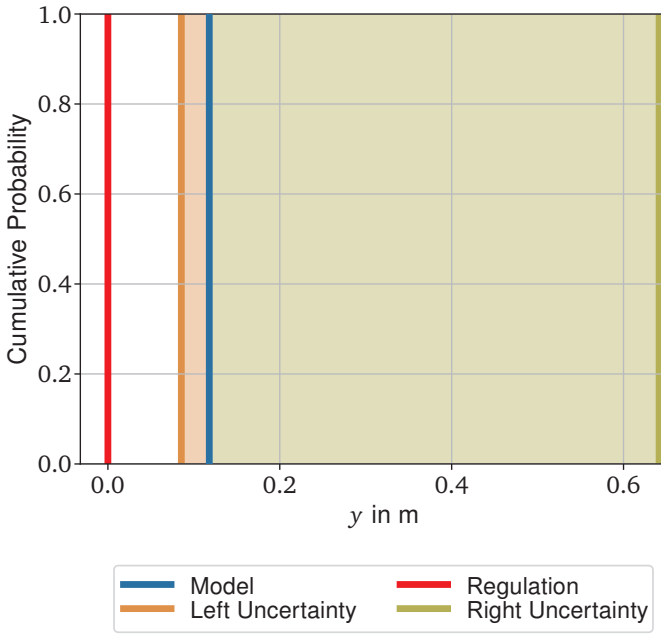


Figure 4.9: Uncertainty expansion for an exemplary application scenario from [23, Fig. 7]. It contains a small shift to the left and a large shift to the right due to small left and large right errors.

4.2.10 Type Approval

The type approval builds on the uncertainty expansion results by relating it to the regulation threshold. The distance to line has a lower threshold of zero but no upper threshold due to the combined minimum over the left and right distance. Therefore, we are only interested in the left uncertainty bound from a worst-case safety perspective. For completeness, we illustrated the results from the validation metric, error learning, and uncertainty expansion not only for the left but also the right side. There are quantities in other use cases that must not exceed an upper threshold and require the right side, or quantities that contain both thresholds and require both sides. Since shifts to the left of 3 cm are small, they rarely affect the type-approval decisions in this specific PoC. Their distribution is shown in Figure 4.10 across the entire scenario space. The decisions of both the nominal simulation and the uncertainty bounds for all 62 application scenarios are coded into the cross and triangle symbol and the green and orange color. There

are 29 green triangles where both the nominal simulations and the uncertainty bounds pass, 32 orange crosses where both fail, and one orange triangle where only the simulation passes. There can be no green cross by definition, since the uncertainty bounds include the simulation and are always more conservative. Thus, we obtain 30 passed and 32 failed cases for the simulation, and 29 passed and 33 failed cases considering the uncertainty bounds. There is only one case that toggles due to the uncertainties, since the virtual vehicle behaves less safe in simulation than in reality leading to the small left errors.

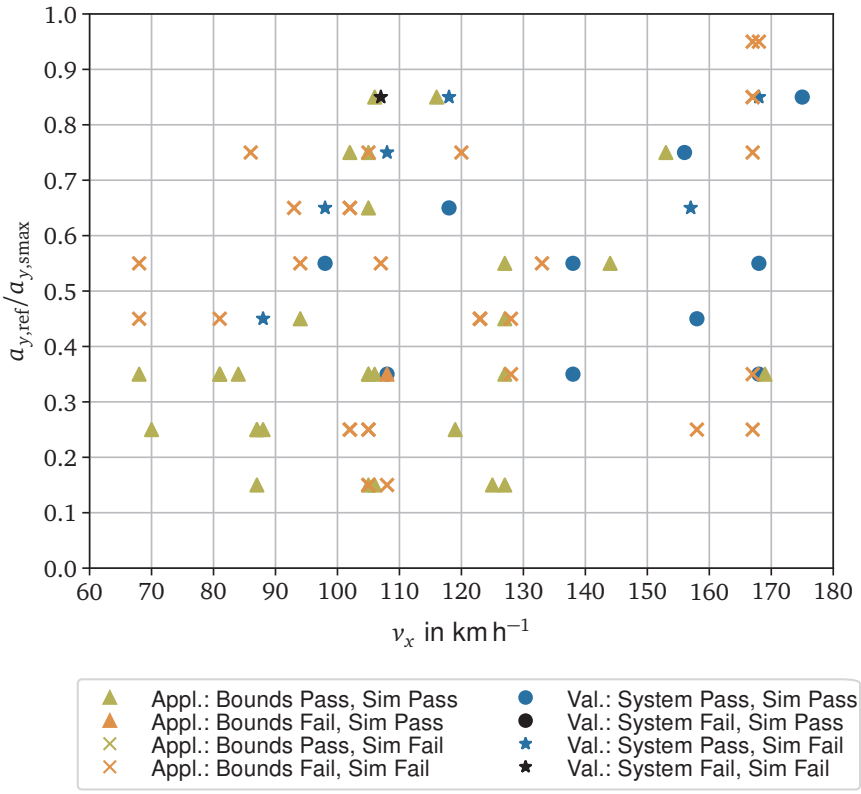


Figure 4.10: Distribution of type-approval decisions across all application and validation scenarios from [23, Fig. 11]. The former contain the nominal simulation decisions and the ones including the uncertainty bounds. The latter contain the nominal simulation decisions and the actual ones from the real vehicle. Each of them can either pass or fail. Thus, there are eight combinations that are coded into four symbols and four colors for clarity.

In addition, Figure 4.10 presents the type-approval results for all validation scenarios. The primary purpose of the validation scenarios is the assessment of the model quality. Nevertheless, they provide further insights about the vehicle safety in particular from the real system. We must be aware that the validation scenarios have repetitions to allow accurate model validation. In contrast, we treated the application scenarios separately for a balance between safety and efficiency. To use the repetitions in form of a CDF for decision making, we must choose how many steps have to exceed the zero threshold. From a purely safety point of view, all steps should pass the regulation. At this point, however, we want to relate the type-approval decisions at the validation scenarios with the actual ones at the application scenarios. Requiring all steps to pass would bias the comparison, since it is much harder to pass only one application scenario compared to all repetitions of one validation scenario. Therefore, we position the threshold right in the middle of the probability range to make the comparison between the simulation across the validation and application domain as fair as possible without a bias to either side. A confidence

of $\alpha = 50\,\%$ means that at least half of the steps have to pass so that the overall validation scenario counts as passed case. If more than half fail, it counts as failed case.

Figure 4.10 codes the type-approval decisions of the real system and the nominal simulation of all 17 validation scenarios into the circle and star symbol and the blue and black color. There are 10 blue circles where both pass, one black star where both fail, and 6 blue stars where only the real vehicle passes. Thus, the simulation passes in 10 out of the 17 validation scenarios, while the real vehicle passes in 16 out of the 17 scenarios. The pass rate of $10/17 = 58.8\,\%$ is higher for the simulation at the coverage-based validation scenarios compared to its rate of $30/62 = 48.4\,\%$ at the data-driven application scenarios. Nevertheless, both are in the same order of magnitude of roughly $50\,\%$. The real vehicle has a pass rate of $16/17 = 94.1\,\%$ and is significantly safer compared to the virtual vehicle. When considering each repetition separately, there are more failed cases also for the real vehicle. This corresponds to the maximum confidence and can be traced backed to the individual assessment dots in Figure 4.7b by the interested reader. In summary, the amount of failed cases after the virtual-based process would be too high to allow for a market launch, as the regulation requires all test scenarios to be passed.

4.2.11 Discussion of the Results

We keep this discussion short, as several aspects are taken up by the broader discussion in the following chapter. Our focus is not on the type approval results per se as with the manufacturer, but on the perspective of the validation methodology. We identified during the assessment that the vehicle behaves significantly less safe in simulation than in reality. This large systematic error testifies to the demand of model validation. However, the direction of the error towards a less safe simulation with many line crosses already indicated that the hybrid vehicle cannot pass all scenarios. Thus, we do not have the target constellation to obtain an approval. This would require a nominal simulation passing all scenarios with a buffer that is sufficiently large in relation to the quantified modeling errors so that it still passes after adding the uncertainty bounds. At the end, the role of the validation methodology in this PoC cannot be to increase the confidence in a successful approval. Thus, this PoC does not exploit its largest strength in form of the error and uncertainty aggregation. This fact can be seen in the previous subsection, since there was only one of the 30 passed scenarios from the nominal simulation that toggled, and none of the 32 failed scenarios was allowed to be improved by definition. Instead, the role of the validation methodology in this PoC is to identify the systematic error and thereby show the developers the way for future improvements. This role has been successfully taken by the methodology. It provides not only the final binary decisions but many intermediate results, from which many insights should be extracted as in the previous subsections. Lastly, there are two possibilities for improvements for the manufacturer. They are still basic directions without internal knowledge of the system. For virtual-based safeguarding, the driving functions must not only hold the line in reality but provide additional buffers for the modeling errors. In addition, the virtual environments must be enhanced so that the modeling errors are as small as possible. These two directions will pave the way for all model-based safety assessment approaches in the future.

5 Discussion

We have already discussed the respective results of the MMU study and the real PoC in Chapter 4.1.6 and Chapter 4.2.11. This chapter focuses on a discussion of the overall validation methodology. In the first part, we discuss the requirements, research questions and gaps, and research objectives. This is an integral element of the scientific process to ensure that all aspects are fulfilled. We will thereby specifically address the main aspects of the methodology in compact summaries. In the second part, we will go through the individual framework blocks and overall aspects to discuss future improvements for current limitations.

5.1 Fulfillment of the Requirements

This section checks whether the requirements for the validation methodology from Chapter 3.1 are fulfilled. It starts with the first half addressing the generic validation framework:

R1.1) **Modularization:** The framework has a modular structure by means of the domains, manifestations, and blocks representing the individual steps of a virtual-based process. This is the key enabler for further requirements such as the unification and composition.

R1.2) **Unification:** The framework brings together techniques from several engineering fields. During the automotive PoC, we configured the framework not only with automotive approaches such as the data-driven scenario method but also with the non-deterministic PBA from fluid dynamics. The framework provides such unification for further fields.

R1.3) **Formalization:** We have presented a mathematical notation in Chapter 3.2.5 to formalize all framework domains, blocks, and manifestations. It provided clear descriptions and interfaces that allowed for the interchangeability of single approaches.

R1.4) **Composition:** The framework covers a variety of techniques and offers several configuration options thanks to the building block principle. They were summarized for each framework block at the beginning of the respective section before making a selection and describing it in more detail. There are only a few explicitly mentioned exceptions such as the meta-model approach or ensemble validation, where a graphical representation has been omitted for clarity. The framework provides, for example, the non-deterministic manifestation for users with high requirements and medium system complexity, or the deterministic manifestation for users with medium requirements and high system complexity. The latter is a novel combination between deterministic simulations and uncertainty expansion that was only made possible due to building block principle.

R1.5) **Aggregation:** The aggregation of errors and uncertainties is the core of the framework that connects the domains by means of the vertical error pipeline and the inverse arrows in Figure B.1. It is responsible for the increased confidence in decision making.

Thus, all generic requirements were successfully taken into account during the development of the framework. They provided considerable added value to both the methodology and the results of this work. They are initially responsible for the increased confidence and novel combinations improving the state of the art. The second half of the requirements addresses the specific configuration of the framework for AV type approval. They were explicitly considered at the beginning of the configuration procedure in Chapter 3.2.7. Therefore, their fulfillment should already be given in advance. Nevertheless, we will briefly go through them step by step:

R2.1) **Objective:** We configured the validation framework exclusively with objective techniques such as frequentist statistics. The generic framework itself is aligned towards objectivity by means of the error aggregation. This makes the subjective tolerances only an option for the model developer, whereas they are currently the main approach in the automotive field. There are a few hyper-parameters such as the statistical confidence of the PI that the user selects. However, the user should define sound values such as a $95\,\%$ confidence in advance and not adjust them afterwards until a desired result appears.

R2.2) **Unchanged:** We avoided calibration approaches that would modify the simulation.

R2.3) **Protected:** We avoided calibration and component-level validation with access of internal quantities and focused on accessible quantities instead.

R2.4) **Regulatory:** We expanded the nominal model predictions with the estimated uncertainties so that they are reflected in the binary decision making during type approval.

R2.5) **Safety First:** The safety-critical FPs are avoided whenever possible without neglecting FNs. The fulfillment of this requirement is confirmed in the results from Chapter 4.1.5, as the recall rate is $100\,\%$ and the precision rate still $77\,\%$ or rather $86\,\%$.

R2.6) **Trustworthy:** The validation method increases the trustworthiness of the simulation compared to the tolerance approach by providing uncertainty bounds. The results in Figure 4.3 support this by showing several cases where the tolerance approach estimates the model to be valid even though the true values and uncertainty bounds contradict this.

5.2 Response to the Research Questions

The research questions from Chapter 2.7 raise methodological aspects about the configuration of the validation methodology for the AV type approval. They were answered mainly in the course of Chapter 3 by selecting the most suitable configuration options based on our requirements. The main question relates to the overall validation methodology. It is divided into three sub-questions, the third of which was given the highest priority based on the analysis of the state of the art. The following list answers each question in the form of a compact summary:

Q1.1) **What is the best method for selecting scenarios for model validation?**

We dedicated Chapter 3.3 to the answer of this sub-question. We analyzed four categories of scenario approaches from the field of safeguarding AVs with respect to their suitability for model validation. We finally selected coverage-based scenarios, since they fit to the nature of validation experiments. Individual physical tests must cover the entire scenario space and they must be repeatable to precisely quantify the test conditions for a fair re-simulation. We developed a coverage-based scenario approach based on map planning. It is tailor-made for the real-world validation experiments from Chapter 4.2.

Q1.2) **Which validation metric suits the comparison between experiments and re-simulations for the type approval of AVs?**

We addressed this sub-question in Chapter 3.5. It contains a taxonomy of validation metrics for different types of input and output quantities. We decided against metrics with Boolean outputs, since they interpret model validity as a binary problem and thus loose information. We also decided against metrics with probabilistic outputs, since they match the subjective Bayesian approach. Instead, we selected metrics that preserve the units of the response quantities. For the deterministic framework manifestation, we calculated an absolute deviation between KPIs from simulation and reality. This emphasizes characteristic values that are of key importance during the type approval and contains not only a magnitude but also a sign information. For the non-deterministic manifestation, we introduced the AAVM. It takes into account the entire shape of the non-deterministic structures by means of an area calculation, and it has an asymmetric nature that can differ between left and right areas.

Q1.3) **How to integrate modeling uncertainties into the type approval of AVs?**

We dedicated the sequence of sections from Chapter 3.6 to Chapter 3.8 to this sub-question, as it deserves the highest priority. The aggregation of errors and uncertainties to the type approval is crucial for such safety-critical systems as AVs to avoid false decisions with high impact. The current literature often neglects the modeling errors after applying subjective tolerances or visual comparisons. Instead, we preserve the knowledge gained during the validation experiments by means of an error model. It can be applied afterwards to infer the validation errors to unseen application scenarios. We implemented a multiple linear regression with external PIs to not only cover the trend of the errors, but also the additional uncertainties associated with the error modeling itself and the extrapolation to new scenarios. We integrated the modeling uncertainties before making the actual type-approval decisions. We refrained from correcting nominal model predictions, since a bias correction is risky. Instead, we expanded the nominal model predictions with the modeling uncertainties to obtain further statistical guarantees.

Q1) **How should a validation methodology be designed to assess the quality of simulations for type approval of AVs using scenario-based testing?**

We started the framework configuration in Chapter 3.2.7 with the overall domains and manifestations. We focused on the validation and application domain, since numerical effects turned out to be negligible during verification and since the models are already calibrated before reaching the technical service. The validation and application domain are connected by an error and uncertainty aggregation. We decided to compare a deterministic and non-deterministic manifestation and to apply PBA within the latter. The block details were summarized in the answers to the sub-questions. The MMU study from Chapter 4.1 demonstrated that both manifestations can recognize all safety-critical modeling errors. The non-deterministic manifestation showed slightly better results but requires additional effort. Ultimately, the user has two good options at this point. We selected a hybrid manifestation for the real PoC in Chapter 4.2, as it represents a good compromise between confidence and effort for the complex test environment. In the end, we closed the research gap from Chapter 2.6 by combining deterministic and non-deterministic simulations with uncertainty aggregation. We successfully applied it for AVs and paved the way for further applications in the future.

5.3 Fulfillment of the Research Objectives

This section checks whether the initial research objectives from the introduction are fulfilled. The following list discusses the main objective with its four pillars distributed across the entire thesis:

O1) **Development and application of an overall framework that covers the quality assessment of the simulation models on the one hand, in order to enable the actual safety assessment of AVs on the other hand:**

The main research objective is represented in the core of the validation framework, since it consists of a separate validation and application domain. The validation enables the safeguarding application by connecting the two domains via an uncertainty pipeline.

O1.1) **Development of taxonomies for classification of the state of the art:**

We introduced several taxonomies in this work to tackle the dynamic research in safeguarding AVs and the heterogeneous research landscape in model validation. We distinguished six major safety assessment approaches in Chapter 2.1.2. Within the SBA in Chapter 2.1.3, we differed between four categories of scenario methods and assigned the references based on certain criteria. In the course of Chapter 3, we presented major pillars of model validation such as validation metrics or error aggregation approaches. At the beginning of a section, we introduced the respective taxonomy to structure the literature and to ultimately make systematic selections. For example, Table 3.5 classifies validation metrics according to their input and output type, or Table 3.6 structures aggregation approaches based on the technique, stage, source, and target domain.

O1.2) **Extension of the taxonomies to an overall framework:**

We developed the VV&UQ framework in Figure 3.1. It connects single steps to an overall validation process with a hierarchical structure of framework domains, blocks, and manifestations. The individual taxonomies were then inserted into the appropriate place in the framework. We initially presented a safety framework in Figure 2.1 that could be inserted into the general application domain of the VV&UQ framework. However, several of its scenario processing steps must be imagined within the application scenario block.

O1.3) **Validation of the framework itself through simulative preliminary studies:**

The majority of the state of the art that develops a new test or validation method only applies it to a simple PoC without validating the method itself. In contrast, we conducted an extensive preliminary study in Chapter 4.1 to validate the validation framework itself. We build on MMU to obtain the true values from a reference simulation. We evolved it by intentionally injecting modeling errors to stress test the framework and by evaluating it using a binary classifier. The combination of a perfect recall rate with high precision gives us quantitative confidence in the framework that the state of the art lacks.

O1.4) **First application of the framework for model-based type approval of AVs:**

We applied the validation framework in Chapter 4.2 to the actual LKA type approval according to R-79. We enhanced the algorithms to tackle real-world artifacts such as noisy signals. We successfully went through the entire validation process and identified systematic errors in the simulation environment. We derived two areas of improvement for developers to enable virtual-based safeguarding in the future.

5.4 Limitations and Outlook

In the previous three sections, we ensured that important aspects of the validation methodology are fulfilled. In this section, we will discuss limitations and link them to possible improvements. This section starts with the individual framework blocks and continuous with the overall use case and its extension. Further aspects that go beyond the techniques we actually applied during the PoC are outsourced to the appendix in Chapter C. This includes, for example, the verification and calibration domain, the fully non-deterministic, time-variant, and hierarchical manifestation, or the macroscopic application assessment.

5.4.1 Framework Blocks

This subsection discusses the framework blocks independently before the higher-level discussion follows. It incorporates student theses that were supervised by the author of this dissertation. One of these contributed to the publication [22] of the MMU study. However, the vast majority prepared new approaches that are not yet applied in this dissertation. Nevertheless, they fit well into the framework building blocks, such as in particular the scenario design and the error modeling. They can be applied to a common use case, resulting in further new combinations of the framework thanks to the building block principle.

Scenario Design

The scenario method determines how the scenarios are distributed across the space. The data-driven application scenarios were chosen to include randomness but were not the focus of this work. The coverage-based validation scenarios were chosen to ensure repeatability. The specific map planning algorithm was tailor-made for the characteristics of R-79. There are alternative space-filling designs such as Latin Hypercube Sampling that can generally be used and are more efficient than an equidistant full-factorial design. There are even designs beyond that, some of which were assigned to the following student theses:

- Stadler [376] worked on an adaptive design that adds scenario points in regions where the results from the initial design suggest a better resolution.
- Martin [377] aimed to find regions with high modeling errors by comparing two simulation models based on several scenario designs. He implemented Monte Carlo sampling, Latin Hypercube Sampling, the genetic algorithm, and evolutionary strategies, and obtained the best results for Latin Hypercube Sampling. They could still be improved by feeding the Latin Hypercube Sampling design as initialization into the genetic algorithm.
- Zhou [378] worked on assigning driving data to predefined scenario classes using rule-based and machine learning algorithms.
- Schneider [379] clustered preprocessed data based on similarities.

The adaptive falsification techniques are suitable for simulation but hardly applicable as validation scenarios in the real world. The data-driven scenario classification and clustering algorithms should be used for validation scenarios only as a backup solution if the coverage-based planning does not work. This can be interesting if environment conditions such as other road users can only be observed but not controlled to force test repetitions. Then, the algorithms can extract

similar conditions as replacement of actual test repetitions. However, they contain usually more scatter and lead ultimately to a less precise quantification of the model-form uncertainties.

Experiment and Simulation

We focused on one vehicle in one virtual test environment being validated against reality before being applied to the virtual safety assessment. There are a few extensions possible in the AV field. We have seen that there are several types of virtual environments including hybrid ones and that the real validation experiments can be reused for safeguarding. In the most flexible setup, we could validate all virtual environments in parallel against the same reality, reuse the real data for safeguarding, and assign each new application scenario to the most suitable environment. This would cause additional validation effort, but it would leverage the strengths of each environment in safeguarding and make better use of resources. We can imagine that a MiL environment is fast and can be parallelized in the cluster, but its modeling errors might be too high in certain regions of the scenario space. In contrast, a ViL environment is slow and hardly applicable to non-deterministic simulations, but it is closer to reality due to more hardware components and could provide a detailed analysis of important scenarios. The scenario assignment to the test environments is rarely addressed in the current literature [168]. The advances in the validation methodology can be an enabler for this.

Application Assessment

The assessment depends fully on the application. In this thesis, we used the minimum distance to line of R-79. It is important that the KPI follows a functional dependence on the scenario inputs, such as higher speeds and accelerations leading to smaller distances to line. If the behavior contains unexpected jumps or outliers, this poses a problem for safety assessment, since the sampling of application scenarios would need to hit these singular blind spots exactly to reveal them. This is normally not the case, but it may occur with complex trajectory planners using machine learning and is an open research question [380]. Similarly, this poses a problem for the sampling of validation scenarios and model validation in general. The error model with PIs can compensate for unsteady behavior but only to some extent. We will discuss the consistency in the KPI extraction from time signals in a separate aspect addressing the dynamic behavior.

Validation Metrics

The absolute deviation and area metric were systematic choices based on our requirements. Nevertheless, it might be interesting for the future to implement multiple validation metrics and compare them via a MMU study in analogy to the comparison of the deterministic and non-deterministic manifestation.

Error Learning and Inference

The linear regression with PIs was a systematic choice for a first PoC. Nevertheless, there are alternative meta-modeling techniques, several of which were assigned to student theses:

- Schneider [381] combined ensemble validation via u-pooling with linear regression to find an optimal data split for the two approaches.
- Wang [382] applied Gaussian Process and polynomial regression.
- Wirth [383] applied Interval Predictor Models.

- Wang [384] applied Multi-Layer Perceptrons.
- Freier [385] applied Long Short-Term Memories.

After implementing all these techniques, they can be transferred to a common use case and compared via a MMU study. The u-pooling is interesting for scenarios with other road users that cannot be repeated. It aggregates distinct scenarios to the same scenario point by means of a probabilistic transformation so that multiple repetitions are generated to enable non-deterministic validation metrics. The ensemble validation in general is interesting for uncharacterized or partially characterized experiments as they occur in the complex traffic environment, but it loses the dependency from the scenario space.

Error Integration and Decision Making

We selected the uncertainty expansion technique, since a conservative approach fits to the type approval. We directly applied it for the non-deterministic manifestation and converted the total error to an uncertainty for the deterministic manifestation. The size of the uncertainty bounds was proven meaningful by our results. If they would be too large, the number of validation scenarios and the measurement precision should be increased. The bias correction is a risky alternative that, if applied, should at least by combined with uncertainties [386].

5.4.2 Use Case and Extension

This final subsection is dedicated to the overall discussion of the use case and its scalability. We developed a generic VV&UQ framework and applied it to the R-79 use case from the perspective of a technical service. It is characterized by steady-state cornering behavior with two scenario parameters and the distance to line output. This use case served its purpose as an objective type approval regulation, for which a corresponding vehicle and data were available. This allowed the entire work to be demonstrated on a consistent and illustrative use case. However, strictly speaking, the specific framework configuration and the obtained results are only valid for this use case. Nevertheless, future users can take this as a blueprint and define their own requirements to configure the framework for their application. This might be another type-approval regulation, safeguarding in general, or other use cases in the automotive field and beyond. These requirements will sometimes motivate for the same framework configuration but sometimes for new ones. On the one hand, a change of perspective from a technical service to a car manufacturer can remove limitations such as data protection, no calibration, or no subjective priors. This enables alternatives within the framework such as Bayesian calibration. On the other hand, scaling the complexity of the use case can reach the limits of current techniques.

The current safeguarding literature demonstrates new approaches by means of illustrative examples. The same holds true for the R-79 use case in this dissertation. However, the car manufacturers have to scale these approaches from academia to industrialization to reach a full safety assessment in the entire traffic environment. This will be demanding for many techniques and motivate to improve or combine them. The scaling factors are a combination of many aspects discussed so far. First and foremost, a full safety assessment goes significantly beyond the R-79 type approval with two scenario parameters and generally beyond any scientific paper. The SBA focuses on interesting traffic situations executed in simulation, and it decomposes the scenario space into multiple logical scenarios and the critical driving behavior into multiple KPIs. This decomposition does not only simplify safeguarding but also our framework configuration. It is not required that the uncertainty pipeline including the error model can cover the entire scenario

space with all parameters at once. Instead, it is possible to configure the framework several times for varying logical scenarios and KPIs. The reduced effort due to reduced complexity should significantly exceed the increased effort due to multiple framework applications. The individual framework configurations could even use the same techniques, such as the same validation metric or learning approach, just with new data provided. Despite the SBA focusing on interesting traffic situations, the complexity and testing effort is still too large. Companion approaches such as functional decomposition [291] or sensitivity analysis [301, 381] are gaining importance to further exploit redundancies in the scenario space. In addition, there are further aspects such as dynamic behavior that correlate with the scalability to higher automation levels.

In the end, this thesis and its main framework add significant added value to safeguarding AVs and beyond. Nevertheless, there are and will always be many open issues for several dissertations and the entire research community.

6 Summary

Automated driving has the potential to reduce accident numbers in the long term, but it comes with the short-term risk of causing additional accidents due to the introduction of a new system. An extensive safety assessment is required to keep these risks as low as possible. Since real-world testing reaches its limits due to high mileage requirements, simulation is a promising candidate to execute the tests in a safe and scalable virtual world. However, this raises a large gap between the real vehicle that will be introduced to the market and the virtual vehicle that is used for safety assessment. If the quality of the simulation models is insufficient, the safety statements can lead to a deceptive trust and ultimately to unexpected accidents in the real world. Therefore, it is of particular relevance to integrate a model validation methodology into the safety assessment of the vehicle. This represents the main research objective of this dissertation. Major parts of it were already published in previous peer-reviewed publications and connected and extended to an entire thesis here. Its chapters cover a scientific process from the state of the art to the methodology to the results and discussion.

The state of the art affecting this thesis is twofold. On the one hand, we require knowledge about the safety assessment as the use case of this work. On the other hand, we present model validation methods that inspire the core of this work. We introduce several safety assessment approaches, with an emphasis on the frequently used scenario-based approach that aims for safety-relevant traffic situations. We distinguished four categories of scenario methods: knowledge-based, data-driven, coverage-based, and falsification-based methods. A special emphasis was placed on the type approval from the perspective of a technical service as motivation of this work. We presented regulations such as R-79 addressing the lane-keeping behavior of vehicles. The model validation state of the art begins with fundamental theory including types of simulation models and sources of their errors and uncertainties. It continues with validation methods across several engineering fields. This includes the automotive field with automated vehicle models and component models of the sensor and vehicle dynamics. Nevertheless, it goes beyond by introducing literature from railway, aircraft, and numerical simulations. The state of the art chapter concludes with criticism of the literature to derive research gaps and questions. This work intends to close a major gap by aggregating errors and uncertainties from model validation to the actual decision making of the application. The research questions focus on how such a validation methodology shall be designed for the type-approval of automated vehicles.

The methodology chapter starts with the derivation of requirements. The purpose is to first develop a generic validation framework, from which all fields can benefit, before configuring it to the special type-approval use case. The generic framework consists of domains, blocks, and manifestations. The domains represent a whole process covering model verification, calibration, validation, and prediction to new application scenarios. Each domain consists of several blocks for the individual process steps. There are multiple manifestations to cover different types of simulation models such as (non-)deterministic, time-(in)variant, hierarchical, or formal simulations.

Based on our requirements, we decided to focus on the validation and application domain, and on a comparison of the deterministic and non-deterministic manifestation. Both domains contain a block for the scenario design. We developed a coverage-based approach to generate validation scenarios that ensure the repeatability of the experiments. In contrast, we developed a data-driven approach that extracts application scenarios from a virtual driving data set to ensure randomness. The validation scenarios are executed in physical experiments followed by the corresponding re-simulations. The application scenarios, however, are executed only in simulation to exploit the advantages of the virtual-based safeguarding process. The two blocks for the experiment and simulation can be accompanied by an assessment block to post-process the results. In case of the type approval, a minimum distance to line is calculated for each scenario. The real and virtual assessment results in the validation domain are compared by means of validation metrics. We selected an absolute deviation and an area metric between probability distributions for this block. The resulting modeling errors are not neglected as usual but captured in form of a data-driven error model. We applied a multiple linear regression with external prediction intervals to learn the errors from validation. These can then be inferred to new application scenarios. We added the inferred errors as bounds around the nominal application predictions. These uncertainty bounds can be imagined as adding systematically derived buffers around the vehicle edges. They offer additional guarantees during the type-approval decision making if not only the nominal simulation but the entire bounds pass the regulation.

The results chapter consists of two parts. In the first one, we validated the validation framework itself. In the second one, we applied the framework to the actual type approval including physical experiments. For the framework validation, we replaced the physical system with a reference model to intentionally inject an error and know the true values. We used a binary classifier to relate the type-approval decisions of our methodology to the true ones. A perfect recall rate indicates that the methodology successfully identified all scenarios where the erroneous simulation would have caused false decisions. This holds true for both the deterministic and the non-deterministic manifestation. Afterwards, we applied a hybrid manifestation as a combination between the two to the actual type approval. For this proof of concept, we were given both a real vehicle and a hybrid simulation environment, containing both models and hardware components. Our validation methodology revealed systematic errors between the hybrid vehicle and the real vehicle on the road. The hybrid vehicle came closer to the lane markings and often crossed them so that an approval would not be possible. This is not the intended outcome from the perspective of the developer. However, it demonstrates that the validation methodology successfully identified weaknesses that were provided to the developers for future improvement.

The discussion chapter analyses both the overall validation framework and individual aspects within it. It directly suggests possible solutions for future improvement of current limitations. In the first part, we ensured that all research objectives, gaps, questions, and requirements of the overall validation methodology are fulfilled. In the second part, we discussed individual aspects and gave an outlook for future enhancement. The generic validation framework can be configured for new applications beyond the type approval such as the internal safeguarding of the car manufacturer or even to further engineering fields. It will be interesting to see which configurations prevail there in the long term. Finally, we encourage the use of the model-based safeguarding process. When it is accompanied by stable driving functions, high-quality simulation environments, an extensive validation methodology, and supporting safety assessment approaches, we can make our traffic more safe due to automated driving.

List of Figures

Figure 1.1: Structure of the thesis with assignment of research objectives and papers. 5

Figure 2.1: Framework of the SBA based on [9, Fig. 2]. 10

Figure 2.2: Scenario extraction methods including literature based on [9, Fig. 3-4]. 11

Figure 2.3: Scenario generation methods including literature based on [9, Fig. 5-6]. 12

Figure 2.4: Virtual-based homologation process of stability control systems. It consists of a passive and active vehicle model validation followed by the actual virtual type approval. If all approvals tests are passed (final yes-arrow), the vehicle can be sold on the market. Otherwise (final no-arrow), the manufacturer must make internal improvements to the vehicle. 14

Figure 2.5: Lane-keeping scenario with illustration from [22] and quotes from [184]. 15

Figure 2.6: Mathematical structures to describe deterministic errors as well as epistemic, aleatory, and mixed uncertainties based on [195, Fig. 2, 21, Fig. 2]. All structures are visualized as cumulative probabilities $P(X \leq x)$ for consistency. The point value can be seen as a degenerate probability and the interval as a degenerate p-box. The horizontal lines are drawn once here for correctness, but omitted for simplicity as we proceed. 19

Figure 2.7: Hierarchical model types based on [191]. 22

Figure 2.8: Architecture of an AV adapting [201] and five environment layers from [30]. 24

Figure 2.9: Categories of the tolerance approach checking whether (a) the experimental repetitions, (b) the simulation, or (c) the error lies within the correspondingly centered tolerance. 26

Figure 2.10: PBA. It determines the model-form uncertainty during model validation in (a). It then combines the model-form with the input and numerical uncertainty in (b). The green shift to the left and right in (b) corresponds to the size of the green area from (a), respectively. 29

Figure 2.11: Exemplary Interval Predictor Model from [21, Fig. 5] with upper and lower boundaries $\overline{g}_m(x)$ and $\underline{g}_m(x)$ enclosing the calibration data D^c after inverse optimization. 30

Figure 2.12: Evaluation of scenario approaches. Comparison of (a) the data-driven and knowledge-based approach for scenario extraction from [9, Fig. 7] and (b) the testing-based and falsification-based approach for scenario generation from [9, Fig. 8]. Details regarding the derivation of the criteria and ratings can be found in [9, Sec. 8]. 32

Figure 2.13: Comparison of validation methods from [21, Fig. 11-12]. The split is for visualization purposes only. Details regarding the criteria and ratings can be found in [21, Sec. 8.1]. PBA stands for Probability Bound Analysis, IPM for Interval Predictor Model, and OUA for Output Uncertainty Approach. 34

Figure 3.1: VV&UQ framework representing a virtual-based process of model validation and prediction. While the former compares model and system, the latter purely relies on the erroneous model. This gap is targeted by the vertical error pipeline consisting of the validation metric, error learning and inference, and error integration. They aggregate errors and uncertainties to the application domain so that they are reflected in a reliable decision making. 41

Figure 3.2: Probabilistic simulation propagating uncertainties via several deterministic simulations based on [21, Fig. 6]. The double arrow emphasizes multiple samples. 48

Figure 3.3: Data-driven condition checks based on [23, Fig. 3a-b]. The time signals and condition thresholds are shown in (a) and the resulting binary masks after thresholding in (b). The flat yellow $\mathbf{v}_x$ signal is always within the large velocity band of D4) (not shown), the blue $\mathbf{a}_y$ signal is always below the blue dashed threshold line, and the red $\mathbf{a}_{y,\mathrm{ref}}$ signal is mostly within the red acceleration band. The glitches in the red mask are shorter than two seconds [185] and thus pulled-up in the brown mask via the connected components algorithm of D5). The AND conjunction of the yellow, blue, and brown masks yields the final green mask of D6). It determines the position of the gray event area of D7), within which the scenario parameters of D8) are extracted from the yellow $\mathbf{v}_x$ and red $\mathbf{a}_{y,\mathrm{ref}}$ signals in (a). 54

Figure 3.4: Coverage-based validation scenarios and data-driven application scenarios from [23, Fig. 4]. The exemplary application scenario at $v_x = 107\,\mathrm{km\,h^{-1}}$ and $a_{y,\mathrm{ref}} = 0.85 \cdot a_{y,\mathrm{smax}}$, which was extracted from the data-driven algorithm in Figure 3.3, can be seen as one of the blue points. While the orange points belong to the 2D bins, only the 1D acceleration bins affect the blue points. All points are centered vertically within the respective acceleration bin. Whereas the blue points can take each value horizontally along the velocity dimension due to the mean value calculation from D8), the orange points have the resolution of $1\,\mathrm{km\,h^{-1}}$ from C2). Due to the maximum operator from C8), there is a maximum of one orange point per 2D bin. Not all bins are filled due to the availability of curves in the map and to reduce the testing effort. The x axis starts at the specified minimum velocity $v_{x,\mathrm{smin}} = 60\,\mathrm{km\,h^{-1}}$. 56

Figure 3.5: Validation and application scenarios based on [22, Fig. 4]. The non-deterministic uncertainty samples form local points clouds around the nominal deterministic scenarios. While each validation scenario has samples for both the model and system, the application scenarios refer exclusively to the model. 57

Figure 3.6: Assessment of the distance to line in an exemplary cornering scenario from [23, Fig. 3c]. The gray area refers to the same event from Figure 3.3, which was extracted by the data-driven algorithm. The distance to left line signal $\mathbf{y}_l$ and distance to right line signal $\mathbf{y}_r$ run inversely, since the lane width was constant during the highway section. At the beginning of the right turn, highlighted in gray, the vehicle moves slightly towards the inside of the curve as a preventive measure, before it gets pushed outwards more and more by the cornering forces. The overall minimum (red dot) occurs at the end of the curve near the left lane marking. 59

Figure 3.7: Exemplary area metric based on [23, Fig. 6]. The distinction in right and left refers to the location of the system from the perspective of the model. This yields, for example, the orange right area $e_r^v = (0.25\,\mathrm{m} - 0.2\,\mathrm{m}) \cdot (0.67 - 0.5) \approx 0.01\,\mathrm{m}$. Since it is not about the absolute location, the axis limits do not start from zero but highlight the relevant range. 62

Figure 3.8: Error inference across the application space with linear regression and external PIs from [22, Fig. 7b]. It is shown for the total error of the deterministic manifestation, but looks analogously for the left and right error of the non-deterministic manifestation as in Figure 4.8. 64

Figure 3.9: Bias correction versus uncertainty expansion from [21, Fig. 9]. The former example contains a correction to the right, while the latter example expands the area to both sides as in PBA. 66

Figure 3.10: Uncertainty expansion and decision making from [22, Fig. 8] for (a) the non-deterministic and (b) deterministic framework manifestation. The blue simulation prediction is expanded via the green uncertainty bounds and compared against the red decision-making threshold. The black system results as ground truth are normally not available in the application domain, but added for the analysis in Chapter 4.1 to show that they lie within the uncertainty bounds. The green shift in (b) can be traced back exemplarily by means of the error surfaces in Figure 3.8. Intersecting the upper PI surface at the coordinates $v_x = 100\,\mathrm{km\,h^{-1}}$ and $a_{y,\mathrm{ref}} = 0.35 \cdot a_{y,\mathrm{smax}}$ from this application scenario yields the $0.4\,\mathrm{m}$ from the green area. 67

Figure 4.1: VV&UQ framework in MMU configuration for LKA type approval. It shows the non-deterministic manifestation based on the deterministic one from [22, Fig. 4]. They differ in the validation metric and the mathematical structures of the KPI y and error e. The orange ground truth is made available by the manufactured universe in analogy to [21, Fig. 13]. 71

Figure 4.2: Minimum distance to line y across the scenario space based on [22, Fig. 3]. The surfaces refer to the application scenarios of velocity and acceleration for a fixed wind speed of $-5\,\mathrm{km\,h^{-1}}$, tank load of $-20\,\mathrm{kg}$, and road slope of $-1°$. The simulation vehicle drives in all scenarios within its lane, while the manufactured vehicle crosses the line in individual scenarios. The corresponding distances are zero by definition (like a crash barrier), since this has implementation advantages over negative values without affecting decision making. 74

Figure 4.3: Dangers of the tolerance approach based on [22, Fig. 9]. The rectangular validation decisions originate from the tolerance approach. The application decisions of circles and crosses relate the non-deterministic nominal model and the true universe. We selected an exemplary tolerance of 0.3 m to demonstrate how easily multiple contradictions arise. A green rectangle claiming a valid model contradicts each cross in the neighborhood, since a deviation between the model and universe decision indicates an invalid model. 76
Figure 4.4: VV&UQ framework for actual LKA type approval based on [23, Fig. 2]. 80
Figure 4.5: Exemplary projection of a real and virtual vehicle trajectory from the same scenario onto a OpenStreetMap from [23, Fig. 5]. 81
Figure 4.6: Repeatability analysis from [23, Fig. 8a]. The repetitions of the measured acceleration show a similar signal form with oscillations around the smooth reference acceleration signal. 82
Figure 4.7: Minimum distance to line as a function of the validation scenarios from [23, Fig. 9]. The colors of the averaged surface planes and the vertical lines connecting the repetitions of the same scenario are only selected for differentiation. 83
Figure 4.8: Validation metric results, error inference via linear regression, and external PIs for both the (a) left areas from [23, Fig. 10] and (b) the right areas. Care should taken with the different z-axis scaling. This is a necessary exception, because otherwise the factor of more than 10 would make it hard to see anything on the left side. Instead, we adjusted the scaling and perspective so that the position of the points and planes to each other is clearly visible. 85
Figure 4.9: Uncertainty expansion for an exemplary application scenario from [23, Fig. 7]. It contains a small shift to the left and a large shift to the right due to small left and large right errors. 86
Figure 4.10: Distribution of type-approval decisions across all application and validation scenarios from [23, Fig. 11]. The former contain the nominal simulation decisions and the ones including the uncertainty bounds. The latter contain the nominal simulation decisions and the actual ones from the real vehicle. Each of them can either pass or fail. Thus, there are eight combinations that are coded into four symbols and four colors for clarity. 87
Figure B.1: VV&UQ framework representing a virtual-based process based on [21, Fig. 1]. The stacked blocks indicate that the same block appears for the verification, calibration, and validation domain, respectively. This holds true with the exception of a mathematical model instead of a system in the verification domain. The inferred errors from the three domains are merged in the error integration block of the application domain. I
Figure B.2: Transformation of a dynamic simulation to a simplified static representation from [21, Fig. 8]. liii

List of Tables

Table 1.1: Overview about the main papers of this dissertation. 4

Table 1.2: Overview about papers that are not a direct part of this thesis but are related in content. 6

Table 2.1: Classification of references from [25, Tab 4–6] into the categories of the tolerance approach. 26

Table 2.2: Classification of references dealing with model validation based on [22, Tab 1], highlighting their focus and the focus of this thesis. The gray areas emphasize main contributions within the respective field. This does not exclude references on the left side of a gray area, for example, simpler approaches without error aggregation in numerical fields. The table supplements previous sections focusing on exemplary references with additional ones [293–344] belonging to the same respective category. 36

Table 3.1: Classification of the sections, figures, and tables of this chapter into the methodological selection process for each framework block. 43

Table 3.2: Framework block mappings in accordance with Figure B.1. The validation error learning, inference, and decision making are shown as representatives for their verification and calibration counterparts. 47

Table 3.3: Selected domains, manifestations, and VV&UQ approaches for LKA type approval. We select two configurations that differ in the (non-)determinism and will be briefly referred to by the distinguishing factor as deterministic and non-deterministic manifestation. 49

Table 3.4: Parameter ranges and uncertainties from [22, Tab. 2]. The mean μ and variance σ^2 characterize the normal distribution $\mathcal{N}(\mu, \sigma^2)$. The determination of values is described in [22]. 56

Table 3.5: Taxonomy of validation metrics from [21, Tab. 2] with examples like the hypothesis test (HT) or area metric with Principal Component Analysis (PCA). The lines with Boolean outputs are included for completeness, but strictly speaking do not indicate the degree of matching of a metric. For example, we can compare single static or stationary values against a tolerance [183] or compare entire time signals with a tolerance band [182] to derive a Boolean output. 60

Table 3.6: Taxonomy of error and uncertainty aggregation from [21, Tab. 3]. 65

Table 4.1: Binary classification results from [22, Tab. 3]. 77

Table 4.2: Selected domains, manifestations, and VV&UQ approaches for the final PoC. 79

Table A.1: Mapping between thesis and table sections. The first column mostly contains the lowest numbered sections as the content container. The introduction, discussion, and summary chapters are excluded. The second column contains not only the actual references, but additional hints based on the author's judgment. A summary indicates a short form, an extended version a long form, and a rewritten version a form of roughly the same length that potentially shifts the emphasis. Thus, the latter can be a combination of a summary in one part and an extended version in another part. A structured version indicates the preparation of the content in a list, and a restructured version that the content was heavily rearranged. Finally a "–" indicates that the section has no noteworthy origin. xlv

Bibliography

[1] European Commission. „*Road safety: Commission welcomes agreement on new EU rules to help save lives*,“ Mar. 26, 2019. Available: https://ec.europa.eu/commission/presscorner/detail/en/IP_19_1793 [visited on 11/03/2020].

[2] World Health Organization. „*Global status report on road safety 2018*,“ 2018. Available: https://www.who.int/publications/i/item/9789241565684 [visited on 11/03/2020].

[3] Eurostat. „*Persons killed in road accidents by road user (CARE data) [tran_sf_roadus]*,“ Apr. 14, 2020. Available: https://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=tran_sf_roadus&lang=en [visited on 02/02/2021].

[4] SAE J3016. „*Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*,“ 2018.

[5] W. Wachenfeld and H. Winner, „The Release of Autonomous Vehicles,“ in *Autonomous Driving: Technical, Legal and Social Aspects*, M. Maurer, J. C. Gerdes, B. Lenz and H. Winner, ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 425–449, ISBN: 978-3-662-48847-8.

[6] S. Shalev-Shwartz, S. Shammah and A. Shashua, „On a Formal Model of Safe and Scalable Self-driving Cars,“ *arXiv preprint arXiv:1708.06374*, 2017. Available: http://arxiv.org/abs/1708.06374 [visited on 01/23/2020].

[7] P. Koopman and M. Wagner, „Toward a Framework for Highly Automated Vehicle Safety Validation,“ in *WCX World Congress Experience*, 2018.

[8] N. Kalra and S. M. Paddock, „Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?,“ *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182–193, 2016.

[9] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick and F. Diermeyer, „Survey on Scenario-Based Safety Assessment of Automated Vehicles,“ *IEEE Access*, vol. 8, pp. 87456–87477, 2020.

[10] PEGASUS Project Office: German Aerospace Center, ed. „*PEGASUS Method: An Overview*,“ 2019. Available: https://www.pegasusprojekt.de/files/tmpl/Pegasus-Abschlussveranstaltung/PEGASUS-Gesamtmethode.pdf.

[11] A. Leitner et al. „*ENABLE-S3: Testing & Validation of Highly Automated Systems: Summary of Results*,“ 2019. Available: https://enable-s3.eu/media/dissemination-material/.

[12] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt and M. Maurer, „Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving,“ in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 982–988.

[13] P. Koopman and M. Wagner, „Challenges in Autonomous Vehicle Testing and Validation,“ *SAE International Journal of Transportation Safety*, vol. 4, no. 1, pp. 15–24, 2016.

[14] F. Fahrenkrog, L. Wang, T. Platzer, A. Fries, F. Raisch and K. Kompaß, „Prospective Effectiveness Safety Assessment of Automated Driving Functions – From The Method to the Results," in *26th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, 2019.

[15] Underwriters Laboratories. „*UL 4600: Standard for Safety for the Evaluation of Autonomous Products*," 2019.

[16] M. Althoff and J. M. Dolan, „Online Verification of Automated Road Vehicles Using Reachability Analysis," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 903–918, 2014.

[17] G. Box, „Robustness in the Strategy of Scientific Model Building," in *Robustness in Statistics* Elsevier, 1979, pp. 201–236, ISBN: 9780124381506.

[18] W. L. Oberkampf and C. J. Roy, *Verification and Validation in Scientific Computing*, Cambridge, Cambridge University Press, 2010, ISBN: 9780511760396.

[19] P. J. Durst, D. T. Anderson and C. L. Bethel, „A historical review of the development of verification and validation theories for simulation models," *International Journal of Modeling, Simulation, and Scientific Computing*, vol. 08, no. 02, p. 1730001, 2017.

[20] A. Gaidon, Q. Wang, Y. Cabon and E. Vig, „Virtual Worlds as Proxy for Multi-object Tracking Analysis," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4340–4349, ISBN: 978-1-4673-8851-1.

[21] S. Riedmaier, B. Danquah, B. Schick and F. Diermeyer, „Unified Framework and Survey for Model Verification, Validation and Uncertainty Quantification," *Archives of Computational Methods in Engineering*, vol. 28, pp. 2655–2688, 2021.

[22] S. Riedmaier, J. Schneider, B. Danquah, B. Schick and F. Diermeyer, „Non-deterministic model validation methodology for simulation-based safety assessment of automated vehicles," *Simulation Modelling Practice and Theory*, vol. 109, pp. 1–19, 2021.

[23] S. Riedmaier, D. Schneider, D. Watzenig, F. Diermeyer and B. Schick, „Model Validation and Scenario Selection for Virtual-Based Homologation of Automated Vehicles," *Applied Sciences*, vol. 11, pp. 1–24, 2021.

[24] S. Riedmaier, J. Nesensohn, C. Gutenkunst, T. Düser, B. Schick and H. Abdellatif, „Validation of X-in-the-Loop Approaches for Virtual Homologation of Automated Driving Functions," in *11th Graz Symposium Virtual Vehicle (GSVF)*, 2018.

[25] B. Danquah, S. Riedmaier and M. Lienkamp, „Potential of statistical model verification, validation and uncertainty quantification in automotive vehicle dynamics simulations: a review," (in press), *Vehicle System Dynamics*, pp. 1–30, 2020.

[26] B. Danquah, S. Riedmaier, J. Rühm, S. Kalt and M. Lienkamp, „Statistical Model Verification and Validation Concept in Automotive Vehicle Design," *Procedia CIRP*, vol. 91, pp. 261–270, 2020.

[27] B. Danquah, S. Riedmaier, Y. Meral and M. Lienkamp, „Statistical Validation Framework for Automotive Vehicle Simulations using Uncertainty Learning," *Applied Sciences*, vol. 11, pp. 1–23, 2021.

[28] J. E. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, F. Niewels and J. M. Zöllner, „Testing of Advanced Driver Assistance Towards Automated Driving: A Survey and Taxonomy on Existing Approaches and Open Questions," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 1455–1462.

[29] T. Menzel, G. Bagschik and M. Maurer, „Scenarios for Development, Test and Validation of Automated Vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[30] G. Bagschik, T. Menzel and M. Maurer, „Ontology based Scene Creation for the Development of Automated Vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1813–1820, DOI: 10.1109/IVS.2018.8500632.

[31] J. Sauerbier, J. Bock, H. Weber and L. Eckstein, „Definition of Scenarios for Safety Validation of Automated Driving Functions," *ATZ worldwide*, vol. 121, no. 1, pp. 42–45, 2019, DOI: 10.1007/s38311-018-0197-2.

[32] P. Junietz, U. Steininger and H. Winner, „Macroscopic safety requirements for highly automated driving," *Transportation Research Record*, vol. 2673, no. 3, pp. 1–10, 2019.

[33] S. Wagner, K. Groh, T. Kühbeck, M. Dörfel and A. Knoll, „Using Time-to-React based on Naturalistic Traffic Object Behavior for Scenario-Based Risk Assessment of Automated Driving," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[34] International Organization for Standardization. „*ISO/PAS 21448: Road vehicles — Safety of the intended functionality*," 2019.

[35] M. Wood et al. „*Safety First for Automated Driving*," 2019.

[36] Á. Takács, D. A. Drexler, P. Galambos, I. J. Rudas and T. Haidegger, „Assessment and Standardization of Autonomous Vehicles," in *2018 IEEE 22nd International Conference on Intelligent Engineering Systems (INES)*, 2018, pp. 000185–000192.

[37] International Organization for Standardization. „*ISO 26262: Road vehicles — Functional safety*," 2018.

[38] United Nations Economic Commission for Europe (UNECE). „*Addendum 130 - Regulation 131 — Uniform provisions concerning the approval of motor vehicles with regard to the Advanced Emergency Braking Systems (AEBS)*," 2013.

[39] J. Kapinski, J. V. Deshmukh, X. Jin, H. Ito and K. Butts, „Simulation-Based Approaches for Verification of Embedded Control Systems: An Overview of Traditional and Advanced Modeling, Testing, and Verification Techniques," *IEEE Control Systems*, vol. 36, no. 6, pp. 45–64, 2016.

[40] N. Aréchiga, S. M. Loos, A. Platzer and B. H. Krogh, „Using theorem provers to guarantee closed-loop system properties," in *2012 American Control Conference (ACC)*, 2012, pp. 3573–3580, ISBN: 978-1-4577-1096-4.

[41] S. M. Loos, A. Platzer and L. Nistor, „Adaptive Cruise Control: Hybrid, Distributed, and Now Formally Verified," in *FM 2011*, M. Butler and W. Schulte, ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, ISBN: 978-3-642-21436-3.

[42] J. Nilsson, A. C. E. Odblom and J. Fredriksson, „Worst-Case Analysis of Automotive Collision Avoidance Systems," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 1899–1911, 2016.

[43] H. Täubig, U. Frese, C. Hertzberg, C. Lüth, S. Mohr, E. Vorobev and D. Walter, „Guaranteeing functional safety: Design for provability and computer-aided verification," *Autonomous Robots*, vol. 32, no. 3, pp. 303–331, 2012.

[44] M. Hartung, D. Hess, R. Lattarulo, J. Oehlerking, J. Perez and A. Rausch. „*Report on Conformance Testing of Application Models*," ed. by Unifying Control and Verification of Cyber-Physical Systems (UnCoVerCPS) Project. 2017.

[45] M. Althoff, „Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars," PhD thesis, Technical University of Munich, Munich, 2010.

[46] F. Gruber and M. Althoff, „Anytime Safety Verification of Autonomous Vehicles," in *2018 IEEE 21th International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 1708–1714.

[47] M. E. O'Kelly, H. Abbas, S. Gao, S. Kato, S. Shiraishi and R. Mangharam, „APEX: Autonomous Vehicle Plan Verification and Execution," in *SAE 2016 World Congress and Exhibition*, 2016.

[48] B. Schürmann, D. Heß, J. Eilbrecht, O. Stursberg, F. Koster and M. Althoff, „Ensuring drivability of planned motions using formal methods," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–8, ISBN: 978-1-5386-1526-3.

[49] H. Abbas, M. E. O'Kelly, A. Rodionova and R. Mangharam, „A Driver's License Test for Driverless Vehicles," *Mechanical Engineering*, vol. 139, no. 12, S13, 2017.

[50] M. O'Kelly, H. Abbas and R. Mangharam, „Computer-aided design for safe autonomous vehicles," in *2017 Resilience Week (RWS)*, 2017, pp. 90–96, ISBN: 978-1-5090-6055-9.

[51] B. Johnson, F. Havlak, H. Kress-Gazit and M. Campbell, „Experimental Evaluation and Formal Analysis of High-Level Tasks with Dynamic Obstacle Anticipation on a Full-Sized Autonomous Vehicle," *Journal of Field Robotics*, vol. 34, no. 5, pp. 897–911, 2017.

[52] P. Nilsson, O. Hussien, A. Balkan, Y. Chen, A. D. Ames, J. W. Grizzle, N. Ozay, H. Peng and P. Tabuada, „Correct-by-Construction Adaptive Cruise Control: Two Approaches," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1294–1307, 2016.

[53] T. Wongpiromsarn, U. Topcu and R. M. Murray, „Receding Horizon Temporal Logic Planning," *IEEE Transactions on Automatic Control*, vol. 57, no. 11, pp. 2817–2830, 2012.

[54] C. Pek, P. Zahn and M. Althoff, „Verifying the safety of lane change maneuvers of self-driving vehicles based on formalized traffic rules," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1477–1483, ISBN: 978-1-5090-4804-5.

[55] A. Rizaldi and M. Althoff, „Formalising Traffic Rules for Accountability of Autonomous Vehicles," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 1658–1665, ISBN: 978-1-4673-6596-3.

[56] B. Vanholme, D. Gruyer, B. Lusetti, S. Glaser and S. Mammar, „Highly Automated Driving on Highways Based on Legal Safety," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 333–347, 2013.

[57] N. Aréchiga, „Specifying Safety of Autonomous Vehicles in Signal Temporal Logic," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 58–63.

[58] C. Wang and H. Winner, „Overcoming Challenges of Validation Automated Driving and Identification of Critical Scenarios," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 2639–2644, ISBN: 978-1-5386-7024-8.

[59] Tesla. „*Tesla Autonomy Day*," 2019. Available: https://youtu.be/Ucp0TTmvqOE?t=10540 [visited on 11/16/2020].

[60] Daimler. „*Daimler and Bosch: Start of the San José pilot project for automated ride-hailing service*," 2019. Available: https://www.daimler.com/innovation/case/autonomous/pilot-city-san-jose.html [visited on 11/16/2020].

[61] S. Kitajima, K. Shimono, J. Tajima, J. Antona-Makoshi and N. Uchida, „Multi-agent traffic simulations to estimate the impact of automated technologies on safety," *Traffic injury prevention*, vol. 20, no. sup1, pp. 58–64, 2019.

[62] C. Roesener, M. Harth, H. Weber, J. Josten and L. Eckstein, „Modelling Human Driver Performance for Safety Assessment of Road Vehicle Automation," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 735–741.

[63] M. Saraoglu, A. Morozov and K. Janschek, „MOBATSim: MOdel-Based Autonomous Traffic Simulation Framework for Fault-Error-Failure Chain Analysis," *IFAC-PapersOnLine*, vol. 52, no. 8, pp. 239–244, 2019.

[64] Y. Page et al., „A Comprehensive and Harmonized Method for Assessing the Effectiveness of Advanced Driver Assistance Systems by Virtual Simulation: The PEARS Initiative," in *24th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, 2015.

[65] W. Huang, K. Wang, Y. Lv and F. Zhu, „Autonomous vehicles testing methods review," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 163–168.

[66] P. Junietz, Wachenfeld Walther, K. Klonecki and H. Winner, „Evaluation of Different Approaches to Address Safety Validation of Automated Driving," in *21st IEEE International Conference on Intelligent Transportation Systems, November 4-7, 2018, Maui, Hawaii*, 2018, pp. 491–496.

[67] Forschungsgesellschaft für Straßen- und Verkehrswesen. „*Richtlinien für die Anlage von Autobahnen*," 2008.

[68] J. Guo, U. Kurup and M. Shah, „Is It Safe to Drive? An Overview of Factors, Metrics, and Datasets for Driveability Assessment in Autonomous Driving," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–17, 2019.

[69] Y. Kang, H. Yin and C. Berger, „Test Your Self-Driving Algorithm: An Overview of Publicly Available Driving Datasets and Virtual Testing Environments," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 171–185, 2019.

[70] J. Zhu, W. Wang and D. Zhao, „A Tempt to Unify Heterogeneous Driving Databases using Traffic Primitives," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2052–2057.

[71] R. Krajewski, J. Bock, L. Kloeker and L. Eckstein, „The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems," in *21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2118–2125.

[72] S. Geyer et al., „Concept and development of a unified ontology for generating test and use-case catalogues for assisted and automated vehicle guidance," *IET Intelligent Transport Systems*, vol. 8, no. 3, pp. 183–189, 2014.

[73] W. Chen and L. Kloul, „An Ontology-based Approach to Generate the Advanced Driver Assistance Use Cases of Highway Traffic," in *10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 2018. Available: https://hal.archives-ouvertes.fr/hal-01835139.

[74] F. Klueck, Y. Li, M. Nica, J. Tao and F. Wotawa, „Using Ontologies for Test Suites Generation for Automated and Autonomous Driving Functions," in *2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 2018, pp. 118–123.

[75] Y. Li, J. Tao and F. Wotawa, „Ontology-based test generation for automated and autonomous driving functions," *Information and Software Technology*, vol. 117, p. 106 200, 2020.

[76] F. Wotawa and Y. Li, „From Ontologies to Input Models for Combinatorial Testing," in *Testing Software and Systems*, 2018, pp. 155–170, ISBN: 978-3-319-99927-2.

[77] J.-A. Bolte, A. Bär, D. Lipinski and T. Fingscheidt, „Towards Corner Case Detection for Autonomous Driving," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 438–445.

[78] I. R. Jenkins, L. O. Gee, A. Knauss, H. Yinz and J. Schroeder, „Accident Scenario Generation with Recurrent Neural Networks," in *2018 IEEE 21th International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 3340–3345.

[79] S. Jesenski, J. E. Stellet, F. Schiegg and J. M. Zöllner, „Generation of Scenes in Intersections for the Validation of Highly Automated Driving Functions," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 502–509.

[80] R. Krajewski, T. Moers, D. Nerger and L. Eckstein, „Data-Driven Maneuver Modeling using Generative Adversarial Networks and Variational Autoencoders for Safety Validation of Highly Automated Vehicles," in *2018 IEEE 21th International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2383–2390.

[81] J. Langner, J. Bach, L. Ries, S. Otten, M. Holzäpfel and E. Sax, „Estimating the Uniqueness of Test Scenarios derived from Recorded Real-World-Driving-Data using Autoencoders," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[82] J. Bach, J. Langner, S. Otten, E. Sax and M. Holzäpfel, „Test scenario selection for system-level verification and validation of geolocation-dependent automotive control systems," in *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, 2017, pp. 203–210, ISBN: 978-1-5386-0774-9.

[83] H. Beglerovic, T. Schloemicher, S. Metzner and M. Horn, „Deep Learning Applied to Scenario Classification for Lane-Keep-Assist Systems," *Applied Sciences*, vol. 8, no. 12, p. 2590, 2018.

[84] B. Dávid, G. Láncz and Hunyady Gergely, „Highway Situation Analysis with Scenario Classification and Neural Network based Risk Estimation for Autonomous Vehicles," in *2019 IEEE 17th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 2019, pp. 375–380.

[85] A. Erdogan, B. Ugranli, E. Adali, A. Sentas, E. Mungan, E. Kaplan and A. Leitner, „Real-World Maneuver Extraction for Autonomous Vehicle Validation: A Comparative Study," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 267–272.

[86] R. Gruner, P. Henzler, G. Hinz, C. Eckstein and A. Knoll, „Spatiotemporal representation of driving scenarios and classification using neural networks," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1782–1788, ISBN: 978-1-5090-4804-5.

[87] F. Kruber, J. Wurst, E. S. Morales, S. Chakraborty and M. Botsch, „Unsupervised and Supervised Learning with the Random Forest Algorithm for Traffic Scenario Clustering and Classification," in *30th IEEE Intelligent Vehicles Symposium*, 2019, pp. 2463–2470, ISBN: 978-1-7281-0560-4.

[88] F. Kruber, J. Wurst and M. Botsch, „An Unsupervised Random Forest Clustering Technique for Automatic Traffic Scenario Categorization," in *21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2811–2818.

[89] W. Wang and D. Zhao, „Extracting Traffic Primitives Directly From Naturalistically Logged Data for Self-Driving Applications," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1223–1229, 2018.

[90] H. Watanabe, T. Maly, J. Wallner, T. Dirndorfer, M. Mai and G. Prokop, „Methodology of Scenario Clustering for Predictive Safety Functions," in *9. Tagung Automatisiertes Fahren*, 2019.

[91] H. Weber, J. Bock, J. Klimke, C. Roesener, J. Hiller, R. Krajewski, A. Zlocki and L. Eckstein, „A framework for definition of logical scenarios for safety assurance of automated driving," *Traffic injury prevention*, vol. 20, no. sup1, S65–S70, 2019.

[92] E. de Gelder and J.-P. Paardekooper, „Assessment of Automated Driving Systems using real-life scenarios," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 589–594.

[93] E. de Gelder, J.-P. Paardekooper, O. Op den Camp and B. de Schutter, „Safety assessment of automated vehicles: how to determine whether we have collected enough field data?," *Traffic injury prevention*, vol. 20, no. sup1, S162–S170, 2019.

[94] L. Hartjen, R. Philipp, F. Schuldt, F. Howar and B. Friedrich, „Classification of Driving Maneuvers in Urban Traffic for Parametrization of Test Scenarios," in *9. Tagung Automatisiertes Fahren*, 2019.

[95] J. Zhou and L. del Re, „Identification of critical cases of ADAS safety by FOT based parameterization of a catalogue," in *2017 11th Asian Control Conference (ASCC)*, 2017, pp. 453–458, ISBN: 978-1-5090-1573-3.

[96] M. R. Zofka, F. Kuhnt, R. Kohlhaas, C. Rist, T. Schamm and J. M. Zöllner, „Data-driven simulation and parametrization of traffic scenarios for the development of advanced driver assistance systems," in *18th International Conference on Information Fusion*, 2015.

[97] A. Pütz, A. Zlocki, J. Bock and L. Eckstein, „System validation of highly automated vehicles with a database of relevant traffic scenarios," in *12th ITS European Congress*, 2017.

[98] S. Feng, Y. Feng, H. Sun, Y. Zhang and H. X. Liu, „Testing Scenario Library Generation for Connected and Automated Vehicles: An Adaptive Framework," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[99] D. Zhao, Y. Guo and Y. J. Jia, „TrafficNet: An open naturalistic driving scenario library," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–8.

[100] M. Althoff, M. Koschi and S. Manzinger, „CommonRoad: Composable benchmarks for motion planning on roads," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 719–726, ISBN: 978-1-5090-4804-5.

[101] M. Althoff and J. M. Dolan, „Reachability computation of low-order models for the safety verification of high-order road vehicle models," in *2012 American Control Conference (ACC)*, 2012, pp. 3559–3566, ISBN: 978-1-4577-1096-4.

[102] H. Beglerovic, A. Ravi, N. Wikström, H.-M. Koegeler, A. Leitner and J. Holzinger, „Model-based safety validation of the automated driving function highway pilot," in *8th International Munich Chassis Symposium 2017* (Proceedings), P. P. E. Pfeffer, ed. Wiesbaden: Springer Fachmedien Wiesbaden, 2017, pp. 309–329, ISBN: 978-3-658-18458-2.

[103] L. Huang, Q. Xia, F. Xie, H.-L. Xiu and H. Shu, „Study on the Test Scenarios of Level 2 Automated Vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 49–54.

[104] S. Khastgir, G. Dhadyalla, S. Birrell, S. Redmond, R. Addinall and P. Jennings, „Test Scenario Generation for Driving Simulators Using Constrained Randomization Technique," in *WCX™ 17: SAE World Congress Experience*, 2017.

[105] B. Kim, A. Jarandikar, J. Shum, S. Shiraishi and M. Yamaura, „The SMT-based automatic road network generation in vehicle simulation environment," in *Proceedings of the 13th International Conference on Embedded Software - EMSOFT '16*, 2016, pp. 1–10, ISBN: 9781450344852.

[106] B. Kim, T. Masuda and S. Shiraishi, „Test Specification and Generation for Connected and Autonomous Vehicle in Virtual Environments," *ACM Transactions on Cyber-Physical Systems*, vol. 4, no. 1, pp. 1–26, 2019.

[107] I. Majzik, O. Semeráth, C. Hajdu, K. Marussy, Z. Szatmári, Z. Micskei, A. Vörös, A. A. Babikian and D. Varró, „Towards System-Level Testing with Coverage Guarantees for Autonomous Vehicles," in *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS)*, 2019, pp. 89–94.

[108] T. Ponn, D. Fratzke, C. Gnandt and M. Lienkamp, „Towards Certification of Autonomous Driving: Systematic Test Case Generation for a Comprehensive but Economically-Feasible Assessment of Lane Keeping Assist Algorithms," in *Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2019)*, 2019, pp. 333–342.

[109] E. Rocklage, H. Kraft, A. Karatas and J. Seewig, „Automated scenario generation for regression testing of autonomous vehicles," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2018-March, 2018.

[110] C. E. Tuncali and G. Fainekos, „Rapidly-exploring Random Trees for Testing Automated Vehicles," in *2019 IEEE 22th International Conference on Intelligent Transportation Systems (ITSC)*, 2019.

[111] F. Xie, T. Chen, Q. Xia, L. Huang and H. Shu, „Study on the Controlled Field Test Scenarios of Automated Vehicles," in *SAE Technical Paper*, 2018. Available: https://doi.org/10.4271/2018-01-1633.

[112] J. Zhou and L. d. Re, „Reduced Complexity Safety Testing for ADAS & ADF," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5985–5990, 2017. Available: http://www.sciencedirect.com/science/article/pii/S2405896317317755.

[113] Y. Akagi, R. Kato, S. Kitajima, J. Antona-Makoshi and N. Uchida, „A Risk-index based Sampling Method to Generate Scenarios for the Evaluation of Automated Driving Vehicle Safety *," in *IEEE Intelligent Transportation Systems Conference - ITSC*, 2019, pp. 667–672, ISBN: 978-1-5386-7024-8.

[114] M. Arief, P. Glynn and D. Zhao, „An Accelerated Approach to Safely and Efficiently Test Pre-Production Autonomous Vehicles on Public Streets," in *2018 IEEE 21th International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2006–2011.

[115] D. Åsljung, J. Nilsson and J. Fredriksson, „Comparing Collision Threat Measures for Verification of Autonomous Vehicles using Extreme Value Theory," *IFAC-PapersOnLine*, vol. 49, no. 15, pp. 57–62, 2016.

[116] D. Åsljung, J. Nilsson and J. Fredriksson, „Using Extreme Value Theory for Vehicle Level Safety Validation and Implications for Autonomous Vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 4, pp. 288–297, 2017.

[117] D. Åsljung, M. Westlund and J. Fredriksson, „A Probabilistic Framework for Collision Probability Estimation and an Analysis of the Discretization Precision," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 52–57.

[118] Z. Huang, D. Zhao, H. Lam, D. J. LeBlanc and H. Peng, „Evaluation of automated vehicles in the frontal cut-in scenario — An enhanced approach using piecewise mixture models," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 197–202.

[119] Z. Huang, H. Lam and D. Zhao, „Sequential experimentation to efficiently test automated vehicles," in *2017 Winter Simulation Conference (WSC)*, 2017, pp. 3078–3089.

[120] Z. Huang, H. Lam and D. Zhao, „Towards affordable on-track testing for autonomous vehicle — A Kriging-based statistical approach," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–6.

[121] Z. Huang, Y. Guo, M. Arief, H. Lam and D. Zhao, „A Versatile Approach to Evaluating and Testing Automated Vehicles based on Kernel Methods," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 4796–4802.

[122] Z. Huang, H. Lam, D. J. LeBlanc and D. Zhao, „Accelerated Evaluation of Automated Vehicles Using Piecewise Mixture Models," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2017.

[123] Z. Huang, H. Lam and D. Zhao, „An accelerated testing approach for automated vehicles with background traffic described by joint distributions," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 933–938, ISBN: 978-1-5386-1526-3.

[124] Z. Huang, M. Arief, H. Lam and D. Zhao, „Synthesis of Different Autonomous Vehicles Test Approaches," in *2018 IEEE 21th International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2000–2005.

[125] Z. Huang, M. Arief, H. Lam and D. Zhao, „Evaluation Uncertainty in Data-Driven Self-Driving Testing," in *2019 IEEE 22th International Conference on Intelligent Transportation Systems (ITSC)*, 2019.

[126] M. O'Kelly, A. Sinha, H. Namkoong, R. Tedrake and J. C. Duchi, „Scalable end-to-end autonomous vehicle testing via rare-event simulation," in *Advances in Neural Information Processing Systems*, 2018, pp. 9827–9838.

[127] S. P. Olivares, N. Rebernik, A. Eichberger and E. Stadlober, „Virtual stochastic testing of advanced driver assistance systems," in *Advanced Microsystems for Automotive Applications 2015* Springer, 2016, pp. 25–35.

[128] X. Wang, H. Peng and D. Zhao, eds. „*Combining Reachability Analysis and Importance Sampling for Accelerated Evaluation of Highly Automated Vehicles at Pedestrian Crossing*,“ vol. 3. Dynamic Systems and Control Conference. 2019.

[129] S. Zhang, H. Peng, D. Zhao and E. Tseng, „Accelerated Evaluation of Autonomous Vehicles in the Lane Change Scenario Based on Subset Simulation Technique,“ in *2018 IEEE 21th International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 3935–3940.

[130] D. Zhao, „Accelerated Evaluation of Automated Vehicles,“ PhD thesis, University of Michigan, Michigan, 2016.

[131] D. Zhao, H. Lam, H. Peng, S. Bao, D. J. LeBlanc, K. Nobukawa and C. S. Pan, „Accelerated Evaluation of Automated Vehicles Safety in Lane-Change Scenarios Based on Importance Sampling Techniques,“ *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, pp. 595–607, 2017.

[132] D. Zhao, X. Huang, H. Peng, H. Lam and D. J. LeBlanc, „Accelerated Evaluation of Automated Vehicles in Car-Following Maneuvers,“ *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 733–744, 2018.

[133] P. Feig, V. Labenski, T. Leonhardt and J. Schatz, „Assessment of Technical Requirements for Level 3 and Beyond Automated Driving Systems Based on Naturalistic Driving and Accident Data Analysis,“ in *26th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, 2019.

[134] J. So, I. Park, J. Wee, S. Park and I. Yun, „Generating Traffic Safety Test Scenarios for Automated Vehicles using a Big Data Technique,“ *KSCE Journal of Civil Engineering*, 2019.

[135] L. Stark, M. Düring, S. Schoenawa, J. E. Maschke and C. M. Do, „Quantifying Vision Zero: Crash avoidance in rural and motorway accident scenarios by combination of ACC, AEB, and LKS projected to German accident occurrence,“ *Traffic injury prevention*, vol. 20, no. sup1, pp. 126–132, 2019.

[136] L. Stark, S. Obst, S. Schoenawa and M. Düring, „Towards Vision Zero: Addressing White Spots by Accident Data based ADAS Design and Evaluation,“ in *2019 IEEE International Conference of Vehicular Electronics and Safety (ICVES)*, 2019, pp. 1–6.

[137] M. Althoff and S. Lutz, „Automatic Generation of Safety-Critical Test Scenarios for Collision Avoidance of Road Vehicles,“ in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[138] M. Klischat and M. Althoff, „Generating Critical Test Scenarios for Automated Vehicles with Evolutionary Algorithms,“ in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 2352–2358.

[139] A. Pierson, W. Schwarting, S. Karaman and D. Rus, „Learning Risk Level Set Parameters from Data Sets for Safer Driving,“ in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 273–280.

[140] F. Gao, J. Duan, Y. He and Z. Wang, „A test scenario automatic generation strategy for intelligent driving systems,“ *Mathematical Problems in Engineering*, vol. 2019, 2019.

[141] P. Koopman and F. Fratrik, „How Many Operational Design Domains, Objects, and Events?,“ in *SafeAI*, 2019.

[142] T. Ponn, C. Gnandt and F. Diermeyer, „An Optimization-Based Method to Identify Relevant Scenarios for Type Approval of Automated Vehicles," in *26th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, 2019.

[143] Y. Qi, Y. Luo, K. Li, W. Kong and Y. Wang, „A Trajectory-Based Method for Scenario Analysis and Test Effort Reduction for Highly Automated Vehicle," in *SAE Technical Paper Series*, 2019.

[144] J. Wang, C. Zhang, Y. Liu and Q. Zhang, „Traffic Sensory Data Classification by Quantifying Scenario Complexity," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1543–1548.

[145] Q. Xia, J. Duan, F. Gao, T. Chen and C. Yang, „Automatic Generation Method of Test Scenario for ADAS Based on Complexity," in *SAE Technical Paper*, 2017. Available: https://doi.org/10.4271/2017-01-1992.

[146] Q. Xia, J. Duan, F. Gao, Q. Hu and Y. He, „Test Scenario Design for Intelligent Driving System Ensuring Coverage and Effectiveness," *International Journal of Automotive Technology*, vol. 19, no. 4, pp. 751–758, 2018.

[147] C. Zhang, Y. Liu, Q. Zhang and Wang Le, „A Graded Offline Evaluation Framework for Intelligent Vehicle's Cognitive Ability," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 320–325.

[148] H. Abbas, M. O'Kelly, A. Rodionova and R. Mangharam, „Safe At Any Speed: A Simulation-Based Test Harness for Autonomous Vehicles," in *Seventh Workshop on Design, Modeling and Evaluation of Cyber Physical Systems (CyPhy'17)*, 2017.

[149] R. B. Abdessalem, S. Nejati, L. C. Briand and T. Stifter, „Testing advanced driver assistance systems using multi-objective search and neural networks," in *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering - ASE 2016*, 2016, pp. 63–74, ISBN: 9781450338455.

[150] H. Beglerovic, M. Stolz and M. Horn, „Testing of autonomous vehicles using surrogate models and stochastic optimization," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–6, ISBN: 978-1-5386-1526-3.

[151] A. Corso, Du Peter, K. Driggs-Campbell and M. J. Kochenderfer, „Adaptive Stress Testing with Reward Augmentation for Autonomous Vehicle Validation," in *2019 IEEE 22th International Conference on Intelligent Transportation Systems (ITSC)*, 2019, pp. 163–168.

[152] H. Felbinger, F. Klück and M. Zimmermann, „Comparing two systematic approaches for testing automated driving functions," in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE) Proceedings*, 2019.

[153] B. Gangopadhyay, S. Khastgir, S. Dey, P. Dasgupta, G. Montana and P. Jennings, „Identification of Test Cases for Automated Driving Systems Using Bayesian Optimization," in *2019 IEEE 22th International Conference on Intelligent Transportation Systems (ITSC)*, 2019.

[154] M. Koren, S. Alsaif, R. Lee and M. J. Kochenderfer, „Adaptive Stress Testing for Autonomous Vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1898–1904.

[155] M. Koschi, C. Pek, S. Maierhofer and M. Althoff, „Computationally Efficient Safety Falsification of Adaptive Cruise Control Systems," in *IEEE Intelligent Transportation Systems Conference - ITSC*, 2019, pp. 2879–2886, ISBN: 978-1-5386-7024-8.

[156] R. Lee, M. J. Kochenderfer, O. J. Mengshoel, G. P. Brat and M. P. Owen, „Adaptive stress testing of airborne collision avoidance systems," in *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*, 2015, pp. 1–24, ISBN: 978-1-4799-8940-9.

[157] R. Lee, O. J. Mengshoel, A. Saksena, R. Gardner, D. Genin, J. Brush and M. J. Kochenderfer, „Differential adaptive stress testing of airborne collision avoidance systems," in *2018 AIAA Modeling and Simulation Technologies Conference*, 2018.

[158] G. E. Mullins, P. G. Stankiewicz and S. K. Gupta, „Automated generation of diverse and challenging scenarios for test and evaluation of autonomous vehicles," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1443–1450.

[159] G. E. Mullins, P. G. Stankiewicz, R. C. Hawthorne and S. K. Gupta, „Adaptive generation of challenging scenarios for testing and evaluation of autonomous vehicles," *Journal of Systems and Software*, vol. 137, pp. 197–215, 2018.

[160] G. E. Mullins, „Adaptive Sampling Methods for Testing Autonomous Systems," PhD thesis, Department of Mechanical Engineering, University of Maryland, College Park, 2018.

[161] M. Nabhan, M. Schoenauer, Y. Tourbier and H. Hage, „Optimizing coverage of simulated driving scenarios for the autonomous vehicle," in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE) Proceedings*, 2019.

[162] C. E. Tuncali, T. P. Pavlic and G. Fainekos, „Utilizing S-TaLiRo as an Automatic Test Generation Framework for Autonomous Vehicles," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1470–1475, ISBN: 978-1-5090-1889-5.

[163] C. E. Tuncali, S. Yaghoubi, T. P. Pavlic and G. Fainekos, „Functional gradient descent optimization for automatic test case generation for vehicle controllers," in *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, 2017, pp. 1059–1064, ISBN: 978-1-5090-6781-7.

[164] C. Tuncali, G. Fainekos, H. Ito and J. Kapinski, „Simulation-based Adversarial Test Generation for Autonomous Vehicles with Machine Learning Components," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[165] C. E. Tuncali, G. Fainekos, D. Prokhorov, H. Ito and J. Kapinski, „Requirements-driven Test Generation for Autonomous Vehicles with Machine Learning Components," *IEEE Transactions on Intelligent Vehicles*, 2019.

[166] C. E. Tuncali, „Search-based Test Generation for Automated Driving Systems: From Perception to Control Logic," PhD thesis, Arizona State University, Tempe, 2019.

[167] T. Ponn, A. Schwab, F. Diermeyer, C. Gnandt and J. Záhorský, „A Method for the Selection of Challenging Driving Scenarios for Automated Vehicles Based on an Objective Characterization of the Driving Behavior," in *9. Tagung Automatisiertes Fahren*, 2019.

[168] F. Schuldt, T. Menzel and M. Maurer, „Eine Methode für die Zuordnung von Testfällen für automatisierte Fahrfunktionen auf X-in-the-Loop Simulationen im modularen virtuellen Testbaukasten," in *Workshop Fahrerassistenzsysteme*, 2015, pp. 1–12.

[169] S. Hallerbach, Y. Xia, U. Eberle and F. Koester, „Simulation-Based Identification of Critical Scenarios for Cooperative and Automated Vehicles," *SAE International Journal of Connected and Automated Vehicles*, vol. 1, no. 2, pp. 93–106, 2018.

[170] J. C. Kirchhof, E. Kusmenko, B. Rumpe and H. Zhang, „Simulation as a Service for Cooperative Vehicles," in *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, 2019, pp. 28–37.

[171] J. C. Hayward, „Near-Miss Determination Through Use of a Scale of Danger," *Highway Research Record*, no. 384, 1972.

[172] S. M. S. Mahmud, L. Ferreira, M. S. Hoque and A. Tavassoli, „Application of proximal surrogate indicators for safety evaluation: A review of recent developments and research needs," *IATSS Research*, vol. 41, no. 4, pp. 153–163, 2017.

[173] M. Schreier, V. Willert and J. Adamy, „An Integrated Approach to Maneuver-Based Trajectory Prediction and Criticality Assessment in Arbitrary Road Environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2751–2766, 2016.

[174] W. Wachenfeld, P. Junietz, R. Wenzel and H. Winner, „The worst-time-to-collision metric for situation identification," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 729–734.

[175] C. Amersbach and H. Winner, „Defining Required and Feasible Test Coverage for Scenario-Based Validation of Highly Automated Vehicles*," in *IEEE Intelligent Transportation Systems Conference - ITSC*, 2019, pp. 425–430, ISBN: 978-1-5386-7024-8.

[176] F. Hauer, T. Schmidt, B. Holzmuller and A. Pretschner, „Did We Test All Scenarios for Automated and Autonomous Driving Systems?," in *IEEE Intelligent Transportation Systems Conference - ITSC*, 2019, pp. 2950–2955, ISBN: 978-1-5386-7024-8.

[177] P. Junietz, „Microscopic and Macroscopic Risk Metrics for the Safety Validation of Automated Driving," PhD thesis, Lehrstuhl für Fahrzeugtechnik, TU Darmstadt, Darmstadt, 2019.

[178] NHTSA. „*Federal Automated Vehicles Policy: Accelerating the Next Revolution In Roadway Safety: September 2016*," 2016.

[179] NHTSA. „*A Framework for Automated Driving System Testable Cases and Scenarios*," 2018.

[180] United Nations Economic Commission for Europe (UNECE). „*Addendum 139 - Regulation No. 140 — Uniform provisions concerning the approval of passenger cars with regard to Electronic Stability Control (ESC) Systems*," 2017.

[181] A. Lutz, B. Schick, H. Holzmann, M. Kochem, H. Meyer-Tuve, O. Lange, Y. Mao and G. Tosolin, „Simulation methods supporting homologation of Electronic Stability Control in vehicle variants," *Vehicle System Dynamics*, vol. 55, no. 10, pp. 1432–1497, 2017.

[182] International Organization for Standardization. „*ISO 19364: Passenger cars — Vehicle dynamic simulation and validation — Steady-state circular driving behaviour*," 2016.

[183] International Organization for Standardization. „*ISO 19365: Passenger cars — Validation of vehicle dynamic simulation — Sine with dwell stability control testing*," 2016.

[184] United Nations Economic Commission for Europe (UNECE). „*Addendum 78: UN Regulation No. 79 — Uniform provisions concerning the approval of vehicles with regard to steering equipment*," 2018.

[185] United Nations Economic Commission for Europe. „*Proposal for amendments to ECE/-TRANS/WP.29/GRVA/2019/19*," ed. by United Nations Economic Commission for Europe. 2019.

[186] United Nations Economic Commission for Europe (UNECE). „*Proposal for a new UN Regulation on: Uniform provisions concerning the approval of vehicles with regard to Automated Lane Keeping Systems: GRVA-06-02-Rev.4*,“ 2020.

[187] American Society of Mechanical Engineers, *Standard for verification and validation in computational fluid dynamics and heat transfer: An American national standard*, (ASME V&V). vol. 20-2009, Reaffirmed 2016, New York, NY, The American Society of Mechanical Engineers, 2009, ISBN: 9780791832097.

[188] H. W. Coleman and W. G. Steele, *Experimentation, Validation, and Uncertainty Analysis for Engineers*, Hoboken, NJ, USA, John Wiley & Sons, Inc, 2009, ISBN: 9780470485682.

[189] H. Abdellatif and C. Gnandt, „Use of Simulation for the Homologation of Automated Driving Functions,“ *ATZ Electronics Worldwide (ATZelectronics worldwide)*, vol. 14, no. 12, pp. 68–71, 2019.

[190] G. Terejanu, „Predictive Validation of Dispersion Models Using a Data Partitioning Methodology,“ in *Model Validation and Uncertainty Quantification, Volume 3*, H. S. Atamturktur, B. Moaveni, C. Papadimitriou and T. Schoenherr, ed. Cham: Springer International Publishing, 2015, pp. 151–156, ISBN: 978-3-319-15223-3.

[191] S. Mahadevan, „Uncertainty Aggregation Variability, Statistical Uncertainty, and Model Uncertainty,“ in *École Thématique sur les Incertitudes en Calcul Scientifique (ETICS)*, 2018.

[192] S. Sankararaman and S. Mahadevan, „Integration of model verification, validation, and calibration for uncertainty quantification in engineering systems,“ *Reliability Engineering & System Safety*, vol. 138, pp. 194–209, 2015.

[193] T. Augustin, F. P. A. Coolen, G. d. Cooman and M. C. M. Troffaes, *Introduction to Imprecise Probabilities*, Chichester, UK, John Wiley & Sons, Ltd, 2014, ISBN: 9781118763117.

[194] S. Ferson, V. Kreinovich, L. Ginzburg, D. S. Myers and K. Sentz. „*Constructing Probability Boxes and Dempster-Shafer Structures*,“ 2003.

[195] S. Ferson and K. Sentz, „Epistemic Uncertainty in Agent-based Modeling,“ in *7th international workshop on reliable engineering computing*, 2016, pp. 65–82.

[196] R. G. Easterling. „*Measuring the Predictive Capability of Computational Models: Principles and Methods, Issues and Illustrations*,“ 2001.

[197] M. Viehof, „Objektive Qualitätsbewertung von Fahrdynamiksimulationen durch statistische Validierung,“ PhD thesis, Technische Universität Darmstadt, Darmstadt, 2018.

[198] P. Rosenberger, J. T. Wendler, M. Holder, C. Linnhoff, M. Berghöfer, H. Winner and M. Maurer, „Towards a Generally Accepted Validation Methodology for Sensor Models - Challenges, Metrics, and First Results,“ in *Graz Symposium Virtual Vehicle*, 2019, pp. 1–13.

[199] H. Roehm, J. Oehlerking, M. Woehrle and M. Althoff, „Reachset Conformance Testing of Hybrid Automata,“ in *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control - HSCC '16*, 2016, pp. 277–286, ISBN: 9781450339551.

[200] D. Ao, Z. Hu and S. Mahadevan, „Dynamics Model Validation Using Time-Domain Metrics,“ *Journal of Verification, Validation and Uncertainty Quantification*, vol. 2, no. 1, p. 011004, 2017.

[201] J. Bach, S. Otten and E. Sax, „A Taxonomy and Systematic Approach for Automotive System Architectures - From Functional Chains to Functional Networks," in *Proceedings of the 3rd International Conference on Vehicle Technology and Intelligent Transport Systems*, 2017, pp. 90–101, ISBN: 978-989-758-242-4.

[202] W. Daamen, C. Buisson and S. Hoogendoorn, *Traffic simulation and data: Validation methods and applications*, Hoboken and Boca Raton, Fla., Taylor and Francis and CRC Press, 2015, ISBN: 978-1-4822-2871-7.

[203] K. Groh, S. Wagner, T. Kuehbeck and A. Knoll, „Simulation and Its Contribution to Evaluate Highly Automated Driving Functions," in *WCX SAE World Congress Experience*, 2019.

[204] A. Schaermann, A. Rauch, N. Hirsenkorn, T. Hanke, R. Rasshofer and E. Biebl, „Validation of vehicle environment sensor models," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 405–411, ISBN: 978-1-5090-4804-5.

[205] D. Gruyer, M. Grapinet and P. de Souza, „Modeling and validation of a new generic virtual optical sensor for ADAS prototyping," in *2012 IEEE Intelligent Vehicles Symposium*, 2012, pp. 969–974, ISBN: 978-1-4673-2118-1.

[206] I. Guyon, M. Nikravesh, S. Gunn and L. A. Zadeh, *Feature Extraction*. vol. 207, Berlin, Heidelberg, Springer Berlin Heidelberg, 2006, ISBN: 978-3-540-35487-1.

[207] M. Holder, P. Rosenberger, F. Bert and H. Winner, „Data-driven Derivation of Requirements for a Lidar Sensor Model," in *11th Graz Symposium Virtual Vehicle (GSVF)*, 2018.

[208] International Organization for Standardization. „*ISO 4138: Passenger cars — Steady-state circular driving behaviour — Open-loop test methods*," 2012.

[209] E. Kutluay, „Development and Demonstration of a Validation Methodology for Vehicle Lateral Dynamics Simulation Models," PhD thesis, Technische Universität Darmstadt, Darmstadt, 2012.

[210] Deutsches Institut für Normung, European Committee for Standardization. „*Railway applications — Testing and simulation for the acceptance of running characteristics of railway vehicles — Running behaviour and stationary tests*," 2019.

[211] M. Krausz, „Methode zur Abschätzung der Ergebnisqualität von modularen Gesamtfahrzeugsimulationsmodellen," PhD thesis, Universität Stuttgart, Stuttgart, 2016.

[212] J. E. Bernard and C. L. Clover, „Validation of Computer Simulations of Vehicle Dynamics," in *SAE Technical Paper Series*, 1994.

[213] W. R. Garrott, P. A. Grygier, J. P. Chrstos, G. J. Heydinger, K. Salaani, J. G. Howe and D. A. Guenther, „Methodology for Validating the National Advanced Driving Simulator's Vehicle Dynamics (NADSdyna)," in *SAE Technical Paper Series*, 1997.

[214] G. J. Heydinger, C. Schwarz, M. K. Salaani and P. A. Grygier, „Model Validation of the 2006 BMW 330i for the National Advanced Driving Simulator," in *SAE Technical Paper Series*, 2007.

[215] J. Klemmer, J. Lauer, V. Formanski, R. Fontaine, P. Kilian, S. Sinsel, A. Erbes and J. Zäpf, „Definition and Application of a Standard Verification and Validation Process for Dynamic Vehicle Simulation Models," *SAE International Journal of Materials and Manufacturing*, vol. 4, no. 1, pp. 743–758, 2011.

[216] H. Sarin, M. Kokkolaras, G. Hulbert, P. Papalambros, S. Barbat and R.-J. Yang, „A Comprehensive Metric for Comparing Time Histories in Validation of Simulation Models With Emphasis on Vehicle Safety Applications," in *Volume 1: 34th Design Automation Conference, Parts A and B*, 2008, pp. 1275–1286, ISBN: 978-0-7918-4325-3.

[217] A. Alasty and A. Ramezani, „Genetic Algorithm Based Parameter Identification of a Nonlinear Full Vehicle Ride Model," in *SAE Technical Paper Series*, 2002.

[218] R. Wade-Allen, J. P. Chrstos, G. Howe, D. H. Klyde and T. J. Rosenthal, „Validation of a non-linear vehicle dynamics simulation for limit handling," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 216, no. 4, pp. 319–327, 2002.

[219] S. J. Cassara, D. C. Anderson and J. M. Olofsson, „A Multi-Level Approach for the Validation of a Tractor-Semitrailer Ride and Handling Model," in *SAE Technical Paper Series*, 2004.

[220] I. Dettwiller, M. Rais-Rohani, F. Vahedifard, G. L. Mason and J. D. Priddy, „Bayesian calibration of Vehicle-Terrain Interface algorithms for wheeled vehicles on loose sands," *Journal of Terramechanics*, vol. 71, pp. 45–56, 2017.

[221] L. Horlbeck, „Auslegung elektrischer Maschinen für automobile Antriebsstränge unter Berücksichtigung des Überlastpotentials," PhD thesis, Institute of Automotive Technology, Technical University Munich, Munich, 2018.

[222] H.-X. Hu, „Experimental Validation of a Half-Vehicle Suspension Model," in *SAE Technical Paper Series*, 1993.

[223] P. J. Mcnaull, D. A. Guenther, G. J. Heydinger, P. A. Grygier and M. K. Salaani, „Validation and Enhancement of a Heavy Truck Simulation Model with an Electronic Stability Control Model," in *SAE Technical Paper Series*, 2010.

[224] E. P. Milich, N. F. Fife and D. A. Guenther, „A Validation Study of Vehicle Dynamics Simulations for Heavy Truck Handling Maneuvers," in *SAE Technical Paper Series*, 2001.

[225] F. Netter, „Komplexitätsadaption integrierter Gesamtfahrzeugsimulationen," PhD thesis, Karlsruher Institut für Technologie (KIT), Karlsruhe, 2015.

[226] B. Ozan, P. Sendur, M. E. Uyanik, Y. Oz and S. I. Yilmaz, „A Model Validation Methodology for Evaluating Rollover Resistance Performance of a Ford Commercial Vehicle," in *SAE Technical Paper Series*, 2010.

[227] A. Pazooki, S. Rakheja and D. Cao, „Modeling and validation of off-road vehicle ride dynamics," *Mechanical Systems and Signal Processing*, vol. 28, pp. 679–695, 2012.

[228] S. Schmeiler, „Enhanced insights from vehicle simulation by analysis of parametric uncertainties," PhD thesis, Lehrstuhl für Fahrzeugtechnik, Technische Universität München, Munich, 2020.

[229] M. K. Tschochner, „Comparative Assessment of Early Development Phase Powertrain Concepts," PhD thesis, Institute of Automotive Technology, Technical University Munich, Munich, 2018.

[230] S. Rhode, „Non-stationary Gaussian process regression applied in validation of vehicle dynamics models," *Engineering Applications of Artificial Intelligence*, vol. 93, p. 103716, 2020.

[231] N. Khakpour and M. R. Mousavi, „Notions of Conformance Testing for Cyber-Physical Systems: Overview and Roadmap,“ in *26th International Conference on Concurrency Theory* (Leibniz international proceedings in informatics), L. Aceto and D. d. Frutos Escrig, ed. Saarbrücken/Wadern, Germany: Schloss Dagstuhl - Leibniz-Zentrum für Informatik GmbH Dagstuhl Publishing, 2015, pp. 18–40, ISBN: 978-3-939897-91-0.

[232] O. Stursberg, D. Kontny, Z. Liu, A. Rausch, J. Oehlerking, M. Prandini and G. Frehse. „*Report on modelling of networked cyber-physical system for verification and control*,“ ed. by Unifying Control and Verification of Cyber-Physical Systems (UnCoVerCPS) Project. 2017.

[233] S. B. Liu and M. Althoff, „Reachset Conformance of Forward Dynamic Models for the Formal Analysis of Robots,“ in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 370–376, ISBN: 978-1-5386-8094-0.

[234] M. Nolte, R. Schubert, C. Reisch and M. Maurer, „Sensitivity Analysis for Vehicle Dynamics Models – An Approach to Model Quality Assessment for Automated Vehicles,“ in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1162–1169, ISBN: 978-1-7281-6673-5.

[235] S. Rhode and J. v. Keler, „Online validity monitor for vehicle dynamics models,“ in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE) Proceedings*, 2019.

[236] K. Groh, T. Kuehbeck, B. Fleischmann, M. Schiementz and C. C. Chibelushi, „Towards a Scenario-Based Assessment Method for Highly Automated Driving Functions,“ in *8. Tagung Fahrerassistenzsysteme*, 2017.

[237] D. Notz, M. Sigl, T. Kühbeck, S. Wagner, K. Groh, C. Schütz and D. Watzenig, „Methods for Improving the Accuracy of the Virtual Assessment of Autonomous Driving,“ in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE) Proceedings*, 2019.

[238] S. Wagner, K. Groh, T. Kuhbeck and A. Knoll, „Towards Cross-Verification and Use of Simulation in the Assessment of Automated Driving,“ in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 1589–1596, ISBN: 978-1-7281-0560-4.

[239] J. A. Matute-Peaspan, A. Zubizarreta-Pico and S. E. Diaz-Briceno, „A Vehicle Simulation Model and Automated Driving Features Validation for Low-Speed High Automation Applications,“ *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[240] D. J. Fremont et al., „Formal Scenario-Based Testing of Autonomous Vehicles: From Simulation to the Real World,“ in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–8, ISBN: 978-1-7281-4149-7.

[241] S. Detering, L. Schnieder and E. Schnieder, „Two-Level Validation and Data Acquisition for Microscopic Traffic Simulation Models,“ *International Journal on Advances in Systems and Measurements*, vol. 3, no. 1-2, 2010.

[242] Y. Hollander and R. Liu, „The principles of calibrating traffic microsimulation models,“ *Transportation*, vol. 35, no. 3, pp. 347–362, 2008.

[243] L. Rao and L. Owen, „Validation of High-Fidelity Traffic Simulation Models,“ *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1710, no. 1, pp. 69–78, 2000.

[244] T. Toledo and H. N. Koutsopoulos, „Statistical Validation of Traffic Simulation Models," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1876, no. 1, pp. 142–150, 2004.

[245] L. Zheng, T. Sayed, M. Essa and Y. Guo, „Do Simulated Traffic Conflicts Predict Crashes? An Investigation Using the Extreme Value Approach," in *2019 IEEE 22th International Conference on Intelligent Transportation Systems (ITSC)*, 2019.

[246] M. Nentwig, M. Miegler and M. Stamminger, „Concerning the applicability of computer graphics for the evaluation of image processing algorithms," in *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*, 2012, pp. 205–210, ISBN: 978-1-4673-0993-6.

[247] G. Götz and O. Polach, „The influence of varying input parameters on model validation results," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 232, no. 2, pp. 529–541, 2016.

[248] O. Polach and A. Böttcher, „A new approach to define criteria for rail vehicle model validation," in *23rd International Symposium on Dynamics of Vehicles on Roads and Tracks*, 2013.

[249] C. Funfschilling, G. Perrin, Kraft and Sönke, „Propagation of variability in railway dynamic simulations: Application to virtual homologation," *Vehicle System Dynamics*, vol. 50, no. sup1, pp. 245–261, 2012.

[250] C. Funfschilling and G. Perrin, „Uncertainty quantification in vehicle dynamics," *Vehicle System Dynamics*, vol. 229, no. 6, pp. 1–25, 2019.

[251] R. Hällqvist, M. Eek, I. Lind and H. Gavel, „Validation Techniques Applied on the Saab Gripen Fighter Environmental Control System Model," in *Proceedings of the 56th Conference on Simulation and Modelling (SIMS 56)*, 2015, pp. 199–210.

[252] M. Eek, S. Steinkeller, H. Gavel and J. Ölvander, „Enabling Uncertainty Quantification of Large Aircraft System Simulation Models," in *4th CEAS conference, CEAS2013: "Innovative Europe", Air & Space Conference*, 2013.

[253] M. Eek, H. Gavel and J. Ölvander, „Definition and Implementation of a Method for Uncertainty Aggregation in Component-Based System Simulation Models," *Journal of Verification, Validation and Uncertainty Quantification*, vol. 2, no. 1, p. 011006, 2017.

[254] H. Xiao and P. Cinnella, „Quantification of Model Uncertainty in RANS Simulations: A Review," *Progress in Aerospace Sciences*, vol. 108, pp. 1–31, 2019.

[255] O. H. Díaz-Ibarra, J. Spinti, A. Fry, B. Isaac, J. N. Thornock, M. Hradisky, S. Smith and P. J. Smith, „A Validation/Uncertainty Quantification Analysis for a 1.5 MW Oxy-Coal Fired Furnace: Sensitivity Analysis," *Journal of Verification, Validation and Uncertainty Quantification*, vol. 3, no. 1, p. 011004, 2018.

[256] G. Tapia, W. King, L. Johnson, R. Arroyave, I. Karaman and A. Elwany, „Uncertainty Propagation Analysis of Computational Models in Laser Powder Bed Fusion Additive Manufacturing Using Polynomial Chaos Expansions," *Journal of Manufacturing Science and Engineering*, vol. 140, no. 12, p. 121006, 2018.

[257] S. Atamturktur, F. M. Hemez and J. A. Laman, „Uncertainty quantification in model verification and validation as applied to large scale historic masonry monuments," *Engineering Structures*, vol. 43, pp. 221–234, 2012.

[258] M. Kumar and A. S. Whittaker, „Cross-platform implementation, verification and validation of advanced mathematical models of elastomeric seismic isolation bearings," *Engineering Structures*, vol. 175, pp. 926–943, 2018.

[259] X. Lin, Z. Zong and J. Niu, „Finite element model validation of bridge based on structural health monitoring—Part II: Uncertainty propagation and model validation," *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 2, no. 4, pp. 279–289, 2015.

[260] K. L. van Buren, M. G. Mollineaux, F. M. Hemez and S. Atamturktur, „Simulating the dynamics of wind turbine blades: Part II, model validation and uncertainty quantification," *Wind Energy*, vol. 16, no. 5, pp. 741–758, 2013.

[261] H. Yen, X. Wang, D. G. Fontane, R. D. Harmel and M. Arabi, „A framework for propagation of uncertainty contributed by parameterization, input data, model structure, and calibration/validation data in watershed modeling," *Environmental Modelling & Software*, vol. 54, pp. 211–221, 2014.

[262] L. Eça, G. Vaz, A. Koop, F. Pereira and H. Abreu, „Validation: What, Why and How," in *Volume 2: CFD and VIV*, 2016, ISBN: 978-0-7918-4993-4.

[263] N. Rashidi Mehrabadi, B. Wen, R. Burgos, D. Boroyevich and C. Roy, „Verification, Validation and Uncertainty Quantification (VV & amp;UQ) Framework Applicable to Power Electronics Systems," in *SAE 2014 Aerospace Systems and Technology Conference*, 2014.

[264] N. Rashidi Mehrabadi, R. Burgos, D. Boroyevich and C. Roy, „Modeling and design of the modular multilevel converter with parametric and model-form uncertainty quantification," in *2017 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2017, pp. 1513–1520, ISBN: 978-1-5090-2998-3.

[265] J. Baccou, J. Zhang and E. Nouy, „Towards a Systematic Approach to Input Uncertainty Quantification Methodology," in *The 17th International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH-17)*, 2017.

[266] N. W. Porter, V. A. Mousseau and M. N. Avramova, „Quantified Validation with Uncertainty Analysis for Turbulent Single-Phase Friction Models," *Nuclear Technology*, vol. 2008, no. 5, pp. 1–11, 2018.

[267] J. Yang, Z. Zhan, C. Chen, Y. Shu, L. Zheng, R.-J. Yang, Y. Fu and S. Barbat, „Development of a Comprehensive Validation Method for Dynamic Systems and Its Application on Vehicle Design," *SAE International Journal of Materials and Manufacturing*, vol. 8, no. 3, 2015.

[268] X. Yang, Z. Zhan, Q. Wang, P. Wang, Y. Fang and L. Zheng, „An Integrated Deformed Surfaces Comparison Based Validation Framework for Simplified Vehicular CAE Models," in *WCX World Congress Experience*, 2018.

[269] Z. Zhan, J. Yang, Y. Fu, R.-J. Yang, S. Barbat and L. Zheng, „Research on Validation Metrics for Multiple Dynamic Response Comparison under Uncertainty," *SAE International Journal of Materials and Manufacturing*, vol. 8, no. 2, 2015.

[270] Z. Zhan, J. Yang, X. Chen and Z. Shen, „An Integrated Validation Method for Nonlinear Multiple Curve Comparisons," *SAE International Journal of Materials and Manufacturing*, vol. 9, no. 2, 2016.

[271] W. L. Oberkampf and M. S. Balch, „Closure on the Discussion of "Models, Uncertainty, and the Sandia V&V Challenge Problem"," *Journal of Verification, Validation and Uncertainty Quantification*, vol. 5, pp. 1–3, 2020.

[272] S. Ferson, W. L. Oberkampf and L. Ginzburg, „Model validation and predictive capability for the thermal challenge problem," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 29-32, pp. 2408–2430, 2008.

[273] X. Jiang and S. Mahadevan, „Bayesian wavelet method for multivariate model assessment of dynamic systems," *Journal of Sound and Vibration*, vol. 312, no. 4-5, pp. 694–712, 2008.

[274] R. Rebba and S. Mahadevan, „Computational methods for model reliability assessment," *Reliability Engineering & System Safety*, vol. 93, no. 8, pp. 1197–1207, 2008.

[275] S. Sankararaman and S. Mahadevan, „Assessing the reliability of computational models under uncertainty," in *54th AIAA/ASME/ASCE/AHS/ASC structures, atructural dynamics and materials conference*, 2013.

[276] J. Mullins, S. Mahadevan and A. Urbina, „Optimal Test Selection for Prediction Uncertainty Reduction," *Journal of Verification, Validation and Uncertainty Quantification*, vol. 1, no. 4, p. 041002, 2016.

[277] Z. Hu, C. Hu, Z. P. Mourelatos and S. Mahadevan, „Dynamic Model Discrepancy Quantification in Simulation-Based Design of Dynamical Systems," in *Volume 2B: 44th Design Automation Conference*, 2018, V02BT03A052, ISBN: 978-0-7918-5176-0.

[278] Z. Hu, C. Hu, Z. P. Mourelatos and S. Mahadevan, „Model Discrepancy Quantification in Simulation-Based Design of Dynamical Systems," *Journal of Mechanical Design*, vol. 141, no. 1, p. 011401, 2019.

[279] L. G. Crespo, S. P. Kenny and D. P. Giesy, „Interval Predictor Models With a Linear Parameter Dependency," *Journal of Verification, Validation and Uncertainty Quantification*, vol. 1, no. 2, p. 021007, 2016.

[280] M. C. Campi, G. Calafiore and S. Garatti, „Interval predictor models: Identification and reliability," *Automatica*, vol. 45, no. 2, pp. 382–392, 2009.

[281] J. R. Hamilton and R. G. Hills, „Relation of Validation Experiments to Applications," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 57, no. 5, pp. 307–332, 2010.

[282] J. R. Hamilton and R. G. Hills, „Relation of Validation Experiments to Applications: A Nonlinear Approach," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 57, no. 6, pp. 373–395, 2010.

[283] R. G. Hills. „*Roll-up of Validation Results to a Target Application*," 2013.

[284] P. Koopman and M. Wagner, „Autonomous Vehicle Safety: An Interdisciplinary Challenge," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 90–96, 2017.

[285] S. E. Shladover and C. Nowakowski, „Regulatory challenges for road vehicle automation: Lessons from the California experience," *Transportation Research Part A: Policy and Practice*, vol. 122, pp. 125–133, 2019.

[286] W. Wachenfeld and H. Winner, „The New Role of Road Testing for the Safety Validation of Automated Vehicles," in *Automated Driving*, D. Watzenig and M. Horn, ed. Cham: Springer International Publishing, 2017, pp. 419–436, ISBN: 978-3-319-31893-6.

[287] M. Berk, O. Schubert, H. M. Kroll, B. Buschardt and D. Straub, „Reliability Assessment of Safety-Critical Sensor Information: Does One Need a Reference Truth?," *IEEE Transactions on Reliability*, vol. 68, no. 4, pp. 1227–1241, 2019.

[288] H. Martin and D. Watzenig, „Identification of performance limitations of sensing technologies for AD," in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE) Proceedings*, 2019.

[289] T. Ponn, F. Müller and F. Diermeyer, „Systematic Analysis of the Sensor Coverage of Automated Vehicles Using Phenomenological Sensor Models," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 1000–1006.

[290] C. Amersbach and H. Winner, „Functional Decomposition: An Approach to Reduce the Approval Effort for Highly Automated Driving," in *8. Tagung Fahrerassistenz*, 2017.

[291] C. Amersbach and H. Winner, „Functional decomposition—A contribution to overcome the parameter space explosion during validation of highly automated driving," *Traffic injury prevention*, vol. 20, no. sup1, pp. 52–57, 2019.

[292] C. J. Roy, „Unanswered Questions in 1) Verification, 2) Validation and 3) Uncertainty Quantification," in *ASME 2018 Verification and Validation Symposium*, 2018, ISBN: 978-0-7918-4079-5.

[293] T. Hanke, A. Schaermann, M. Geiger, K. Weiler, N. Hirsenkorn, A. Rauch, S.-A. Schneider and E. Biebl, „Generation and validation of virtual point cloud data for automated driving systems," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–6, ISBN: 978-1-5386-1526-3.

[294] M. Holder et al., „Measurements revealing Challenges in Radar Sensor Modeling for Virtual Validation of Autonomous Driving," in *2018 IEEE 21th International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2616–2622.

[295] M. Holder, J. R. Thielmann, P. Rosenberger, C. Linnhoff and H. Winner, „How to evaluate synthetic radar data? Lessons learned from finding driveable space in virtual environments," in *13. Workshop Fahrerassistenzsysteme und automatisiertes Fahren*, 2020, pp. 1–11.

[296] M. Jasinski, „A Generic Validation Scheme for real-time capable Automotive Radar Sensor Models integrated into an Autonomous Driving Simulator," in *2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR)*, 2019, pp. 612–617, ISBN: 978-1-7281-0933-6.

[297] M. Nentwig and M. Stamminger, „A method for the reproduction of vehicle test drives for the simulation based evaluation of image processing algorithms," in *13th International IEEE Conference on Intelligent Transportation Systems*, 2010, pp. 1307–1312, ISBN: 978-1-4244-7657-2.

[298] M. Nentwig and M. Stamminger, „Hardware-in-the-loop testing of computer vision based driver assistance systems," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 339–344, ISBN: 978-1-4577-0890-9.

[299] M. Nentwig, „Untersuchungen zur Anwendung von computergenerierten Kamerabildern für die Entwicklung und den Test von Fahrerassistenzsystemen," PhD thesis, Friedrich-Alexander-Universität, Erlangen-Nürnberg, 2013.

[300] K. von Neumann-Cosel, „Virtual Test Drive," PhD thesis, Technische Universität München, München, 2013.

[301] A. Ngo, M. P. Bauer and M. Resch, „A Sensitivity Analysis Approach for Evaluating a Radar Simulation for Virtual Testing of Autonomous Driving Functions," in *2020 5th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, 2020, pp. 122–128, ISBN: 978-1-7281-9818-7.

[302] P. Rosenberger, M. Holder, M. Zirulnik and H. Winner, „Analysis of Real World Sensor Behavior for Rising Fidelity of Physically Based Lidar Sensor Models," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[303] E. Roth, T. J. Dirndorfer, Knoll Alois, K. von Neumann-Cosel, T. Ganslmeier, A. Kern and M.-O. Fischer, „Analysis and validation of perception sensor models in an integrated vehicle and environment simulation," in *22nd Enhanced Safety of Vehicle Conference*, 2011.

[304] A. Schaermann, „Systematische Bedatung und Bewertung umfelderfassender Sensormodelle," PhD thesis, Technische Universität München, Munich, 2020.

[305] E. L. Zec, N. Mohammadiha and A. Schliep, „Statistical Sensor Modelling for Autonomous Driving Using Autoregressive Input-Output HMMs," in *2018 IEEE 21th International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 1331–1336.

[306] E. Kutluay and H. Winner, „Validation of vehicle dynamics simulation models – a review," *Vehicle System Dynamics*, vol. 52, no. 2, pp. 186–200, 2014.

[307] Y. Bezin, C. Funfschilling, S. Kraft and L. Mazzola, „Virtual testing environment tools for railway vehicle certification," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 229, no. 6, pp. 755–769, 2015.

[308] G. Götz and O. Polach, „Influence of varying the input parameters on the results of model validation," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 231, no. 5, pp. 598–609, 2017.

[309] G. Götz and O. Polach, „Verification and validation of simulations in a rail vehicle certification context," *International Journal of Rail Transportation*, vol. 6, no. 2, pp. 83–100, 2017.

[310] G. Götz and O. Polach, „Verification and Validation of Simulations for Rail Vehicle Certification," in *ICRT 2017*, 2018, pp. 130–141, ISBN: 9780784481257.

[311] O. Polach and J. Evans, „Simulations of Running Dynamics for Vehicle Acceptance: Application and Validation," *International Journal of Railway Technology*, vol. 2, no. 4, pp. 59–84, 2013.

[312] O. Polach and A. Böttcher, „A new approach to define criteria for rail vehicle model validation," *Vehicle System Dynamics*, vol. 52, no. sup1, pp. 125–141, 2014.

[313] O. Polach et al., „Validation of simulation models in the context of railway vehicle acceptance," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 229, no. 6, pp. 729–754, 2015.

[314] N. Bogojević and V. Lučanin, „The proposal of validation metrics for the assessment of the quality of simulations of the dynamic behaviour of railway vehicles," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 230, no. 2, pp. 585–597, 2014.

[315] C. Funfschilling, G. Perrin, M. Sebes, Y. Bezin, L. Mazzola and M.-L. Nguyen-Tajan, „Probabilistic simulation for the certification of railway vehicles," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 229, no. 6, pp. 770–781, 2015.

[316] S. Kraft, J. Causse and F. Coudert, „An approach for the validation of railway vehicle models based on on-track measurements," *Vehicle System Dynamics*, vol. 53, no. 10, pp. 1480–1499, 2015.

[317] S. Kraft, Q. van Clooster and J. Causse, „Validation of railway vehicle models considering measurement uncertainty," in *19th International Conference on Railway Engineering (ICRE 2017)*, 2017.

[318] R. Licciardello, E. Grappein and A. Rueter, „On the accuracy of the assessment of open-air pressure loads due to passing trains: Part 2: Assessment by simulation," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 229, no. 6, pp. 657–667, 2015.

[319] R. Licciardello, C. Funfschilling and G. Malavasi, „Accuracy of the experimental assessment of running dynamics characteristics quantified through an uncertainty framework," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 231, no. 8, pp. 945–960, 2016.

[320] A. Dorobantu, P. J. Seiler and G. J. Balas, „Validating Uncertain Aircraft Simulation Models Using Flight Test Data," in *AIAA Atmospheric Flight Mechanics (AFM) Conference*, 2013.

[321] A. Dorobantu, G. J. Balas and T. T. Georgiou, „Validating Aircraft Models in the Gap Metric," *Journal of Aircraft*, vol. 51, no. 6, pp. 1665–1672, 2014.

[322] M. Carlsson, S. Steinkellner, H. Gavel and J. Ölvander, „Utilizing Uncertainty Information in Early Model Validation," in *AIAA/AAS Astrodynamics Specialist Conference*, 2012.

[323] M. Carlsson, „Methods for early model validation: Applied on simulation models of aircraft vehicle systems," PhD thesis, Department of Management and Engineering, Linköping University, Linköping, 2013.

[324] M. Eek, S. Kharrazi, H. Gavel and J. Ölvander, „Study of industrially applied methods for verification, validation and uncertainty quantification of simulator models," *International Journal of Modeling, Simulation, and Scientific Computing*, vol. 06, no. 02, p. 1550014, 2015.

[325] M. Eek, J. Karlén and J. Ölvander, „A Framework for Early and Approximate Uncertainty Quantification of Large System Simulation Models," in *Proceedings of the 56th Conference on Simulation and Modelling (SIMS 56)*, 2015, pp. 91–104.

[326] M. Eek, „On Credibility Assessment in Aircraft System Simulation," PhD thesis, Department of Management and Engineering, Linköping University, Linköping, Sweden, 2016.

[327] M. Eek, R. Hällqvist, H. Gavel and J. Ölvander, „Development and Evaluation of a Concept for Credibility Assessment of Aircraft System Simulators," in *AIAA Journal of Aerospace Information Systems*, 2016.

[328] M. Eek, R. Hällqvist, H. Gavel and J. Ölvander, „A Concept for Credibility Assessment of Aircraft System Simulators," *Journal of Aerospace Information Systems*, vol. 13, no. 6, pp. 219–233, 2016.

[329] A. Choudhary, I. T. Voyles, C. J. Roy, W. L. Oberkampf and M. Patil, „Probability Bounds Analysis Applied to the Sandia Verification and Validation Challenge Problem," *Journal of Verification, Validation and Uncertainty Quantification*, vol. 1, no. 1, p. 011003, 2016.

[330] L. G. Crespo, E. A. Morelli, S. P. Kenny and D. P. Giesy, „A Formal Approach to Empirical Dynamic Model Optimization and Validation," in *AIAA Guidance, Navigation, and Control Conference*, 2014.

[331] L. G. Crespo, S. P. Kenny and D. P. Giesy, „Random Predictor Models for Rigorous Uncertainty Quantification," *International Journal for Uncertainty Quantification*, vol. 5, no. 5, pp. 469–489, 2015.

[332] L. G. Crespo, S. P. Kenny and D. P. Giesy, „Random Predictor Models for Rigorous Uncertainty Quantification: Part 1," in *Proceedings of the 25th European Safety and Reliability Conference*, 2015.

[333] L. G. Crespo, S. P. Kenny and D. P. Giesy, „Random Predictor Models for Rigorous Uncertainty Quantification: Part 2," in *Proceedings of the 25th European Safety and Reliability Conference*, 2015.

[334] L. G. Crespo, S. P. Kenny, D. P. Giesy, Y, R. B. Norman and S. R. Blattnig, „Application of Interval Predictor Models to Space Radiation Shielding," in *18th AIAA Non-Deterministic Approaches Conference, AIAA SciTech Forum*, 2016.

[335] L. G. Crespo, S. P. Kenny and D. P. Giesy, „A Comparison of Metamodeling Techniques via Numerical Experiments," in *18th AIAA Non-Deterministic Approaches Conference*, 2016, p. 1, ISBN: 978-1-62410-397-1.

[336] L. G. Crespo, S. P. Kenny and D. P. Giesy, „Staircase predictor models for reliability and risk analysis," *Structural Safety*, vol. 75, pp. 35–44, 2018.

[337] S. Ferson and W. L. Oberkampf, „Validation of imprecise probability models," *International Journal of Reliability and Safety*, vol. 3, no. 1/2/3, p. 3, 2009.

[338] Ferson et al., „Bounding Uncertainty Analyses," in *Application of Uncertainty Analysis to Ecological Risks of Pesticides*, 2010, ISBN: 978-1-4398-0734-7.

[339] M. J. Lacerda and L. G. Crespo, „Interval predictor models for data with measurement uncertainty," in *2017 American Control Conference (ACC)*, 2017, pp. 1487–1492, ISBN: 978-1-5090-5992-8.

[340] W. L. Oberkampf and B. L. Smith, „Assessment Criteria for Computational Fluid Dynamics Model Validation Experiments," *Journal of Verification, Validation and Uncertainty Quantification*, vol. 2, no. 3, p. 031002, 2017.

[341] C. J. Roy and W. L. Oberkampf, „A Complete Framework for Verification, Validation, and Uncertainty Quantification in Scientific Computing (Invited)," in *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 2010.

[342] C. J. Roy and W. L. Oberkampf, „A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing," *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 25-28, pp. 2131–2144, 2011.

[343] C. J. Roy and M. S. Balch, „A Holistic Approach to Uncertainty Quantification with Application to Supersonic Nozzle Thrust," *International Journal for Uncertainty Quantification*, vol. 2, no. 4, pp. 363–381, 2012.

[344] M. Viehof and H. Winner, „Research methodology for a new validation concept in vehicle dynamics," *Automotive and Engine Technology*, 2018.

[345] E. Böde, M. Büker, Ulrich Eberle, M. Fränzle, S. Gerwinn and B. Kramer, „Efficient Splitting of Test and Simulation Cases for the Verification of Highly Automated Driving Functions," in *Computer Safety, Reliability, and Security*, B. Gallina, A. Skavhaug and F. Bitsch, ed. Springer International Publishing, 2018, pp. 139–153, ISBN: 978-3-319-99129-0.

[346] R. E. Morrison, C. M. Bryant, G. Terejanu, S. Prudhomme and K. Miki, „Data partition methodology for validation of predictive models," *Computers & Mathematics with Applications*, vol. 66, no. 10, pp. 2114–2125, 2013.

[347] R. G. Miller, *Simultaneous Statistical Inference*, New York, NY, Springer New York, 1981, ISBN: 978-1-4613-8124-2.

[348] M. B. Dillencourt, H. Samet and M. Tamminen, „A general approach to connected-component labeling for arbitrary image representations," *Journal of the ACM*, vol. 39, no. 2, pp. 253–280, 1992.

[349] H. Sarin, M. Kokkolaras, G. Hulbert, P. Papalambros, S. Barbat and R.-J. Yang, „Comparing Time Histories for Validation of Simulation Models: Error Measures and Metrics," *Journal of Dynamic Systems, Measurement, and Control*, vol. 132, no. 6, p. 061401, 2010.

[350] Z. Xi, H. Pan, Y. Fu and R.-J. Yang, „Validation Metric for Dynamic System Responses under Uncertainty," *SAE International Journal of Materials and Manufacturing*, vol. 8, no. 2, pp. 309–314, 2015.

[351] C.-J. Kat and P. S. Els, „Validation metric based on relative error," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 18, no. 5, pp. 487–520, 2012.

[352] M. Tanaka, „Application of Area Validation Methods for Uncertainty Quantification in Validation Process of Thermal-Hydraulic Code for Thermal Fatigue Issue in Sodium-cooled Fast Reactors," in *ASME 2016 Verification & Validation Symposium*, 2016.

[353] I. T. Voyles and C. J. Roy, „Evaluation of Model Validation Techniques in the Presence of Uncertainty," in *16th AIAA Non-Deterministic Approaches Conference*, 2014, ISBN: 978-1-62410-312-4.

[354] I. T. Voyles and C. J. Roy, „Evaluation of Model Validation Techniques in the Presence of Aleatory and Epistemic Input Uncertainties," in *17th AIAA Non-Deterministic Approaches Conference*, 2015, ISBN: 978-1-62410-347-6.

[355] N. Wang, W. Yao, Y. Zhao, X. Chen, X. Zhang and L. Li, „A New Interval Area Metric for Model Validation With Limited Experimental Data," *Journal of Mechanical Design*, vol. 140, no. 6, 2018.

[356] S. Bi, S. Prabhu, S. Cogan and S. Atamturktur, „Uncertainty Quantification Metrics with Varying Statistical Information in Model Calibration and Validation," *AIAA Journal*, vol. 55, no. 10, pp. 3570–3583, 2017.

[357] Y. Ling and S. Mahadevan, „Quantitative model validation techniques: New insights," *Reliability Engineering & System Safety*, vol. 111, pp. 217–231, 2013.

[358] Y. Liu, W. Chen, P. Arendt and H.-Z. Huang, „Toward a Better Understanding of Model Validation Metrics," *Journal of Mechanical Design*, vol. 133, no. 7, p. 071005, 2011.

[359] J. Mullins, Y. Ling, S. Mahadevan, L. Sun and A. Strachan, „Separation of aleatory and epistemic uncertainty in probabilistic model validation," *Reliability Engineering & System Safety*, vol. 147, pp. 49–59, 2016.

[360] P. Gardner, C. Lord and R. J. Barthorpe, „An Evaluation of Validation Metrics for Probabilistic Model Outputs," in *ASME 2018 Verification and Validation Symposium*, 2018, V001T06A001, ISBN: 978-0-7918-4079-5.

[361] Z. Wang, Y. Fu, R.-J. Yang, S. Barbat and W. Chen, „Validating Dynamic Engineering Models Under Uncertainty," *Journal of Mechanical Design*, vol. 138, no. 11, p. 111402, 2016.

[362] B. M. Rutherford, „Computational modeling issues and methods for the “regulatory problem” in engineering – Solution to the thermal problem,“ *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 29-32, pp. 2480–2489, 2008.

[363] I. Farajpour and S. Atamturktur, „Error and Uncertainty Analysis of Inexact and Imprecise Computer Models,“ *Journal of Computing in Civil Engineering*, vol. 27, no. 4, pp. 407–418, 2013.

[364] R. Shinn, F. M. Hemez and S. W. Doebling, „Estimating the Error in Simulation Prediction Over the Design Space,“ in *Proceedings of the 44th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Conference*, 2003.

[365] M. C. Kennedy and A. O’Hagan, „Bayesian calibration of computer models,“ *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 3, pp. 425–464, 2001.

[366] S. Atamturktur, F. Hemez, B. Williams, C. Tome and C. Unal, „A forecasting metric for predictive modeling,“ *Computers & Structures*, vol. 89, no. 23-24, pp. 2377–2387, 2011.

[367] S. Atamturktur, M. C. Egeberg, F. M. Hemez and G. N. Stevens, „Defining coverage of an operational domain using a modified nearest-neighbor metric,“ *Mechanical Systems and Signal Processing*, vol. 50-51, pp. 349–361, 2015.

[368] R. Feeley, P. Seiler, A. Packard and M. Frenklach, „Consistency of a Reaction Dataset,“ *The Journal of Physical Chemistry A*, vol. 108, no. 44, pp. 9573–9583, 2004.

[369] F. Hemez, H. S. Atamturktur and C. Unal, „Defining predictive maturity for validated numerical simulations,“ *Computers & Structures*, vol. 88, no. 7-8, pp. 497–505, 2010.

[370] W. Li, S. Chen, Z. Jiang, D. W. Apley, Z. Lu and W. Chen, „Integrating Bayesian Calibration, Bias Correction, and Machine Learning for the 2014 Sandia Verification and Validation Challenge Problem,“ *Journal of Verification, Validation and Uncertainty Quantification*, vol. 1, no. 1, p. 011004, 2016.

[371] V. Romero, „Real-Space Model Validation and Predictor-Corrector Extrapolation applied to the Sandia Cantilever Beam End-to-End UQ Problem,“ in *AIAA Scitech 2019 Forum*, 2019, ISBN: 978-1-62410-578-4.

[372] P. J. Roache, „The Method of Manufactured Solutions for Code Verification,“ in *Computer Simulation Validation* (Simulation Foundations, Methods and Applications). vol. 23, C. Beisbart and N. J. Saam, ed. Cham: Springer International Publishing, 2019, pp. 295–318, ISBN: 978-3-319-70765-5.

[373] H. F. Stripling, M. L. Adams, R. G. McClarren and B. K. Mallick, „The Method of Manufactured Universes for validating uncertainty quantification methods,“ *Reliability Engineering & System Safety*, vol. 96, no. 9, pp. 1242–1256, 2011.

[374] N. W. Whiting, C. J. Roy, E. P. Duque and S. Lawrence, „Assessment of Model Validation and Calibration Approaches in the Presence of Uncertainty,“ in *AIAA Scitech 2019 Forum*, 2019, ISBN: 978-1-62410-578-4.

[375] D. Schneider, B. Huber, H. Lategahn and B. Schick, „Measuring method for function and quality of automated lateral control based on high-precision digital ”Ground Truth” maps,“ in *34. VDI/VW-Gemeinschaftstagung Fahrerassistenzsysteme und Automatisiertes Fahren 2018* (VDI-Berichte) Düsseldorf: VDI Verlag GmbH, 2018, pp. 3–16, ISBN: 9783180923352.

[376] M. Stadler, „Effiziente Ermittlung konkreter Fahrszenarien für die virtuelle Sicherheitsbewertung eines Fahrerassistenzsystems," Master's Thesis, Kempten University of Applied Sciences, Kempten, Germany, 2019.

[377] J. Martin, „Scenario Generation for the Comparison of Automated Vehicle Variants," Master's Thesis, Technical University of Munich, Munich, Germany, 2019.

[378] C. Zhou, „Maschinelles Lernen und Mustererkennung zur Klassifikation von Testszenarien automatisierter Fahrzeuge," Semesterarbeit, Technical University of Munich, Munich, Germany, 2019.

[379] J. Schneider, „Datengetriebene Manöverselektion zur Validierung eines Fahrdynamikmodells für das automatisierte Fahren," Semesterarbeit, Technical University of Munich, Munich, Germany, 2019.

[380] C. E. Tuncali, J. Kapinski, H. Ito and J. V. Deshmukh, „Reasoning about Safety of Learning-Enabled Components in Autonomous Cyber-physical Systems," in *2018 Design Automation Conference (DAC)*, 2018.

[381] J. Schneider, „Unsicherheiten in der Modellvalidierung zur Absicherung automatisierter Fahrzeuge," Master's Thesis, Technical University of Munich, Munich, Germany, 2020.

[382] J. Wang, „Statistische Prädiktion maximaler Modellabweichungen für die virtuelle Freigabe automatisierter Fahrzeuge," Semesterarbeit, Technical University of Munich, Munich, Germany, 2019.

[383] R. Wirth, „Datengetriebene Verhaltensmodellierung automatisierter Fahrzeuge durch Punkt- und Intervallmodelle," Bachelor's Thesis, Technical University of Munich, Munich, Germany, 2019.

[384] J. Wang, „Maschinelles Lernen zur Metamodellierung von Fehlern in der Simulation automatisierter Fahrzeuge," Master's Thesis, Technical University of Munich, Munich, Germany, 2020.

[385] F. Freier, „Methode zur datengetriebenen Modellierung automatisierter Fahrfunktionen für die virtuelle Absicherung," Bachelor's Thesis, Kempten University of Applied Sciences, Kempten, Germany, 2019.

[386] J. Mullins, B. Schroeder, R. Hills and L. Crespo, „A Survey of Methods for Integration of Uncertainty and Model Form Error in Prediction," in *Probabilistic Mechanics & Reliability Conference (PMC)*, 2016.

[387] Elsevier. „*Permissions*," 2021. Available: https://www.elsevier.com/about/policies/copyright/permissions [visited on 02/24/2021].

[388] K. Neal, C. Li, Z. Hu, S. Mahadevan, J. Mullins, B. Schroeder and A. Subramanian, „Confidence in the Prediction of Unmeasured System Output Using Roll-Up Methodology," in *Model Validation and Uncertainty Quantification, Volume 3*, R. Barthorpe, ed. Cham: Springer International Publishing, 2019, pp. 105–107, ISBN: 978-3-319-74792-7.

[389] I. Babuška, F. Nobile and R. Tempone, „A systematic approach to model validation based on Bayesian updates and prediction related rejection criteria," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 29-32, pp. 2517–2539, 2008.

[390] T. A. Oliver, G. Terejanu, C. S. Simmons and R. D. Moser, „Validating predictions of unobserved quantities," *Computer Methods in Applied Mechanics and Engineering*, vol. 283, pp. 1310–1335, 2015.

[391] M. Panesi, K. Miki, S. Prudhomme and A. Brandis, „On the assessment of a Bayesian validation methodology for data reduction models relevant to shock tube experiments," *Computer Methods in Applied Mechanics and Engineering*, vol. 213-216, pp. 383–398, 2012.

[392] M. N. Avramova and K. N. Ivanov, „Verification, validation and uncertainty quantification in multi-physics modeling for nuclear reactor design and safety analysis," *Progress in Nuclear Energy*, vol. 52, no. 7, pp. 601–614, 2010.

[393] G. Stevens, S. Atamturktur, R. Lebensohn and G. Kaschner, „Experiment-based validation and uncertainty quantification of coupled multi-scale plasticity models," *Multidiscipline Modeling in Materials and Structures*, vol. 12, no. 1, pp. 151–176, 2016.

[394] G. N. Stevens, „Experiment-Based Validation and Uncertainty Quantification of Partitioned Models: Improving Predictive Capability of Multi-Scale Plasticity Models," PhD thesis, Clemson University, Clemson, South Carolina, USA, 2016.

[395] G. Stevens and S. Atamturktur, „Mitigating Error and Uncertainty in Partitioned Analysis: A Review of Verification, Calibration and Validation Methods for Coupled Simulations," *Archives of Computational Methods in Engineering*, vol. 24, no. 3, pp. 557–571, 2017.

[396] K. L. van Buren, M. Ouisse, S. Cogan, E. Sadoulet-Reboul and L. Maxit, „Effect of model-form definition on uncertainty quantification in coupled models of mid-frequency range simulations," *Mechanical Systems and Signal Processing*, vol. 93, pp. 351–367, 2017.

[397] F. Harirchi, S. Z. Yong and N. Ozay, „Passive Diagnosis of Hidden-Mode Switched Affine Models with Detection Guarantees via Model Invalidation," in *Diagnosability, Security and Safety of Hybrid Dynamic and Cyber-Physical Systems*, M. Sayed-Mouchaweh, ed. Cham: Springer International Publishing, 2018, pp. 227–251, ISBN: 978-3-319-74961-7.

[398] S. Prajna, „Barrier certificates for nonlinear model validation," *Automatica*, vol. 42, no. 1, pp. 117–126, 2006.

[399] N. Ozay, M. Sznaier and C. Lagoa, „Convex Certificates for Model (In)validation of Switched Affine Systems With Unknown Switches," *IEEE Transactions on Automatic Control*, vol. 59, no. 11, pp. 2921–2932, 2014.

[400] S. Streif, D. Henrion and R. Findeisen, „Probabilistic and Set-based Model Invalidation and Estimation Using LMIs," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 4110–4115, 2014.

[401] H. Abbas, B. Hoxha, G. Fainekos, J. V. Deshmukh, J. Kapinski and K. Ueda, „Conformance Testing as Falsification for Cyber-Physical Systems," in *2014 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, 2014, p. 211, ISBN: 978-1-4799-4930-4.

[402] J. V. Deshmukh, R. Majumdar and V. S. Prabhu, „Quantifying conformance using the Skorokhod metric," *Formal Methods in System Design*, vol. 50, no. 2-3, pp. 168–206, 2017.

[403] A. Halder and R. Bhattacharya, „Probabilistic model validation for uncertain nonlinear systems," *Automatica*, vol. 50, no. 8, pp. 2038–2050, 2014.

[404] K. Karydis, I. Poulakakis, J. Sun and H. G. Tanner, „Probabilistically valid stochastic extensions of deterministic models for systems with uncertainty," *The International Journal of Robotics Research*, vol. 34, no. 10, pp. 1278–1295, 2015.

[405] Z. Wei, K. G. Robbersmyr and H. R. Karimi, „An EEMD Aided Comparison of Time Histories and Its Application in Vehicle Safety," *IEEE Access*, vol. 5, pp. 519–528, 2017.

[406] R. Allemang, M. Spottswood and T. Eason, „A Principal Component Analysis (PCA) Decomposition Based Validation Metric for Use with Full Field Measurement Situations," in *Model Validation and Uncertainty Quantification, Volume 3*, H. S. Atamturktur, B. Moaveni, C. Papadimitriou and T. Schoenherr, ed. Cham: Springer International Publishing, 2014, pp. 249–264, ISBN: 978-3-319-04551-1.

[407] H. G. Pasha, R. J. Allemang and M. Agarkar, „Application of PCA-SVD Validation Metric to Develop Calibrated and Validated Structural Dynamic Models," in *Model Validation and Uncertainty Quantification, Volume 3*, S. Atamturktur, T. Schoenherr, B. Moaveni and C. Papadimitriou, ed. Cham: Springer International Publishing, 2016, pp. 213–226, ISBN: 978-3-319-29753-8.

[408] P. L. Green, „Towards the Diagnosis and Simulation of Discrepancies in Dynamical Models," in *Model Validation and Uncertainty Quantification, Volume 3*, S. Atamturktur, T. Schoenherr, B. Moaveni and C. Papadimitriou, ed. Cham: Springer International Publishing, 2016, pp. 271–277, ISBN: 978-3-319-29753-8.

[409] C. L. Denham, M. Patil and C. J. Roy, „Estimating Uncertainty Bounds for Modified Configurations from an Aerodynamic Model of a Nominal Configuration," in *2018 AIAA Atmospheric Flight Mechanics Conference*, 2018.

[410] M. Feilhauer, „Simulationsgestützte Absicherung von Fahrerassistenzsystemen," PhD thesis, Institut für Höchstleistungsrechnen, Universität Stuttgart, Stuttgart, 2018.

[411] J. Schote, „Methode des Künstlichen Universums für Simulatoren des autonomen Fahrens," Bachelor's Thesis, Technical University of Munich, Munich, Germany, 2020.

Prior Publications

During the development of this dissertation, publications and student theses were written in which partial aspects of this work were presented.

Journals; Scopus/Web of Science listed (peer-reviewed)

[9] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick and F. Diermeyer, „Survey on Scenario-Based Safety Assessment of Automated Vehicles," *IEEE Access*, vol. 8, pp. 87456–87477, 2020.

[21] S. Riedmaier, B. Danquah, B. Schick and F. Diermeyer, „Unified Framework and Survey for Model Verification, Validation and Uncertainty Quantification," *Archives of Computational Methods in Engineering*, vol. 28, pp. 2655–2688, 2021.

[22] S. Riedmaier, J. Schneider, B. Danquah, B. Schick and F. Diermeyer, „Non-deterministic model validation methodology for simulation-based safety assessment of automated vehicles," *Simulation Modelling Practice and Theory*, vol. 109, pp. 1–19, 2021.

[23] S. Riedmaier, D. Schneider, D. Watzenig, F. Diermeyer and B. Schick, „Model Validation and Scenario Selection for Virtual-Based Homologation of Automated Vehicles," *Applied Sciences*, vol. 11, pp. 1–24, 2021.

[25] B. Danquah, S. Riedmaier and M. Lienkamp, „Potential of statistical model verification, validation and uncertainty quantification in automotive vehicle dynamics simulations: a review," (in press), *Vehicle System Dynamics*, pp. 1–30, 2020.

[26] B. Danquah, S. Riedmaier, J. Rühm, S. Kalt and M. Lienkamp, „Statistical Model Verification and Validation Concept in Automotive Vehicle Design," *Procedia CIRP*, vol. 91, pp. 261–270, 2020.

[27] B. Danquah, S. Riedmaier, Y. Meral and M. Lienkamp, „Statistical Validation Framework for Automotive Vehicle Simulations using Uncertainty Learning," *Applied Sciences*, vol. 11, pp. 1–23, 2021.

Conferences, Periodicals; Scopus/Web of Science listed (peer-reviewed)

Patents

Journals, Conferences, Periodicals, Reports, Conference Proceedings and Poster, etc.; not Scopus/Web of Science listed

[24] S. Riedmaier, J. Nesensohn, C. Gutenkunst, T. Düser, B. Schick and H. Abdellatif, „Validation of X-in-the-Loop Approaches for Virtual Homologation of Automated Driving Functions,“ in *11th Graz Symposium Virtual Vehicle (GSVF)*, 2018.

Non-thesis-relevant publications; Scopus/Web of Science listed (peer-reviewed)

Thesis-relevant open-source software

Supervised Student's Thesis

The following student theses were written within the framework of the dissertation under the supervision of the author in terms of content, technical and scientific support as well as under relevant guidance of the author. In the following, the bachelor, semester and master theses relevant and related to this dissertation are listed. Many thanks to the authors of these theses for their extensive support within the framework of this research project.

[376] M. Stadler, „Effiziente Ermittlung konkreter Fahrszenarien für die virtuelle Sicherheitsbewertung eines Fahrerassistenzsystems," Master's Thesis, Kempten University of Applied Sciences, Kempten, Germany, 2019.

[377] J. Martin, „Scenario Generation for the Comparison of Automated Vehicle Variants," Master's Thesis, Technical University of Munich, Munich, Germany, 2019.

[378] C. Zhou, „Maschinelles Lernen und Mustererkennung zur Klassifikation von Testszenarien automatisierter Fahrzeuge," Semesterarbeit, Technical University of Munich, Munich, Germany, 2019.

[379] J. Schneider, „Datengetriebene Manöverselektion zur Validierung eines Fahrdynamikmodells für das automatisierte Fahren," Semesterarbeit, Technical University of Munich, Munich, Germany, 2019.

[381] J. Schneider, „Unsicherheiten in der Modellvalidierung zur Absicherung automatisierter Fahrzeuge," Master's Thesis, Technical University of Munich, Munich, Germany, 2020.

[382] J. Wang, „Statistische Prädiktion maximaler Modellabweichungen für die virtuelle Freigabe automatisierter Fahrzeuge," Semesterarbeit, Technical University of Munich, Munich, Germany, 2019.

[383] R. Wirth, „Datengetriebene Verhaltensmodellierung automatisierter Fahrzeuge durch Punkt- und Intervallmodelle," Bachelor's Thesis, Technical University of Munich, Munich, Germany, 2019.

[384] J. Wang, „Maschinelles Lernen zur Metamodellierung von Fehlern in der Simulation automatisierter Fahrzeuge," Master's Thesis, Technical University of Munich, Munich, Germany, 2020.

[385] F. Freier, „Methode zur datengetriebenen Modellierung automatisierter Fahrfunktionen für die virtuelle Absicherung," Bachelor's Thesis, Kempten University of Applied Sciences, Kempten, Germany, 2019.

[411] J. Schote, „Methode des Künstlichen Universums für Simulatoren des autonomen Fahrens," Bachelor's Thesis, Technical University of Munich, Munich, Germany, 2020.

B. Wolf, „Methode zur Validierung von Sensormodellen für das automatisierte Fahren," Master's Thesis, Kempten University of Applied Sciences, Kempten, Germany, 2019.

Appendix

A Relation between Papers and Thesis ..... **xliii**

A.1 Overview about Main Papers ..... **xliii**

A.2 Mapping between Paper and Thesis Sections ..... **xlv**

A.3 Overview about Related Papers ..... **xlvii**

B Framework ..... **xlix**

B.1 Complete Framework ..... **xlix**

B.2 Framework Manifestations ..... **l**

B.2.1 Hierarchical Manifestation li

B.2.2 Formal Manifestation li

B.2.3 Time-(In)variant Manifestation lii

C Extended Discussion ..... **lv**

C.1 Further Discussion Points ..... **lv**

C.2 Framework Domains and Manifestations ..... **lvi**

A Relation between Papers and Thesis

A.1 Overview about Main Papers

This section is devoted to four peer-reviewed journal papers on which this dissertation is mainly based. The author of this dissertation is the first author of all four publications, with an equal contribution in two of them [9, 23]. The author primarily developed, implemented, and evaluated the entire relevant content of this dissertation. This is not to diminish the role of the co-authors of the publications, as cutting-edge research cannot occur without a team. Their support is gratefully acknowledged, despite their individual contributions not listed explicitly at this point. The following list contains a short summary of each paper, classifies them into the chapters of this thesis according to Figure 1.1, quotes the contribution segments belonging to the author of this thesis, and quotes the respective copyright statements:

P1) **IEEE Access 2020 — Survey on Scenario-Based Safety Assessment of Automated Vehicles [9]:**

Summary: This survey paper presents safety assessment approaches for AVs. It focuses in particular on the SBA with four types of scenario selection methods and on formal verification. It classifies 183 references and concludes with an analysis of the state of the art to identify open research topics.

Classification: It forms the basis for the state of the art in Chapter 2.1 and Chapter 2.5 of this thesis.

Contribution: “Stefan Riedmaier and Thomas Ponn contributed equally to this work.” “Stefan Riedmaier and Thomas Ponn (corresponding author) initiated and wrote this paper. They were involved in all stages of development, and primarily developed the concept as well as the whole content of this work.”

Copyright: “This work is licensed under a Creative Commons Attribution 4.0 License.”

P2) **Springer ACME 2020 — Unified Framework and Survey for Model Verification, Validation and Uncertainty Quantification [21]:**

Summary: This survey paper develops a novel validation framework. It unifies a multitude of validation approaches from several engineering communities in a modular and generic framework. It starts with fundamental theory and the framework description, before embedding 201 references and giving an overview of the engineering fields. It concludes with an analysis of the state of the art.

Classification: It forms the basis for the state of the art in Chapter 2.3–2.5, the generic framework in Chapter 3.2, and its configuration options in Chapter 3.3–3.8.

Contribution: “Stefan Riedmaier initiated and wrote this paper. He was involved in all stages of development and primarily developed the concept as well as the whole content of this work.”

Copyright: “© The Author(s) 2020”. “Open Access: This article is licensed under a Creative Commons Attribution 4.0 International License”.

P3) **Elsevier SIMPAT 2021 — Non-deterministic model validation methodology for simulation-based safety assessment of automated vehicles [22]:**

Summary: This research paper configures the validation framework for the specific use case of LKA type approval. It performs a simulative study to validate the validation framework itself. It develops a novel procedure based on a binary classifier to relate the type-approval decisions to their true counterparts. Finally, it analyses and discusses the type-approval and classification results.

Classification: It forms the basis for the research gaps in Chapter 2.6, for the framework configuration in Chapter 3.3–3.8, and the corresponding results in Chapter 4.1.

Contribution: “Stefan Riedmaier initiated and wrote this paper. He was involved in all stages of development and primarily developed the concept and content of this work.”

Copyright: “As an Elsevier journal author, you have the right to include the article in a thesis or dissertation (provided that this is not to be published commercially) whether in full or in part, subject to proper acknowledgment” [387]. Here, the article is included in the dissertation according to the classification above. The acknowledgment and copyright belongs to Elsevier according to the quotation given in [22].

P4) **MDPI Applied Sciences 2021 — Model Validation and Scenario Selection for Virtual-Based Homologation of Automated Vehicles [23]:**

Summary: This research paper configures the validation framework for its actual application to the LKA type approval. It includes physical validation experiments, re-simulations and new model predictions. It develops two algorithms to design test scenarios for safeguarding and in particular for model validation experiments. Finally, it analyses and discusses the results from LKA type approval.

Classification: It forms the basis for the framework configuration in Chapter 3.3–3.8 and the corresponding results in Chapter 4.2.

Contribution: “S.R. and D.S. contributed equally to this publication. S.R. initiated this work and wrote a large part of it. D.S. developed the coverage-based and data-driven scenario methods. S.R. improved and formalized them and developed the presented validation and homologation methodology. D.S. was responsible for the data acquisition. Both S.R. and D.S. wrote the corresponding software parts, brought them together and improved the results in many joint discussions.”

Copyright: “© 2020 by the authors. LicenseeMDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the CreativeCommonsAttribution (CCBY) license”.

A.2 Mapping between Paper and Thesis Sections

As mentioned previously in the text, own publications are a significant part of this work. However, their individual contents have been strongly restructured and interlinked in order to maintain the best possible thread for the dissertation. Therefore, the mapping between thesis and paper sections is complex. Nevertheless, Table A.1 aims to provide the interested reader with references of main section contents as a starting point for further information.

Table A.1: Mapping between thesis and table sections. The first column mostly contains the lowest numbered sections as the content container. The introduction, discussion, and summary chapters are excluded. The second column contains not only the actual references, but additional hints based on the author's judgment. A summary indicates a short form, an extended version a long form, and a rewritten version a form of roughly the same length that potentially shifts the emphasis. Thus, the latter can be a combination of a summary in one part and an extended version in another part. A structured version indicates the preparation of the content in a list, and a restructured version that the content was heavily rearranged. Finally a "–" indicates that the section has no noteworthy origin.

Thesis section	Paper section
2.1.1	Extended and structured version of [9, Sec. II-A-1]
2.1.2	Summary of [9, Sec. II-B, VII]
2.1.3	Summary of [9, Sec. II-C, III–VI] and extended version of [23, Sec. 2.4]
2.2.1	Rewritten version of [24, Sec. II-B]
2.2.2	Rewritten version of [22, Sec. 4.1] and [23, Sec. 2.1]
2.2.3	–
2.3.1	Extended and structured version of [21, Sec. 2.1]
2.3.2	Extended and restructured version of [21, Sec. 2.2, 2.3]
2.3.3	Extended and restructured version of [21, Sec. 2.4, 2.5]
2.3.4	–
2.4.1	Rewritten and restructured version of [21, Sec. 7.2] and [22, Sec. 2.2]
A1)	Rewritten and restructured version of [21, Sec. 7.2.1] and [25, Sec. 4.1]
2.4.2	Summary of [21, Sec. 7.3]
2.4.3	Summary of [21, Sec. 7.4]
A2)	Rewritten version of [21, Sec. 6.6.4]
2.4.4	Summary of [21, Sec. 7.1]
A3)	Rewritten and restructured version of [21, Sec. 5.2.3, 6.2.2, 6.4.5, 7.1.1] and extended version of [22, Sec. 2.3]
A4)	Summary of [21, Sec. 5.2.2, 6.2.1, 6.3.2, 6.4.3, 6.6.1, 6.7.1, 7.1.2]
A5)	Summary of [21, Sec. 4.1.2, 6.3.4]
A6)	Summary of [21, Sec. 5.4.3, 6.4.2]
2.5.1	Rewritten and restructured version of [9, Sec. 8]
2.5.2	–
2.5.3	Rewritten and restructured version of [21, Sec. 8] and extended version of [22, Sec. 3.1]
2.6	Extended version of [22, Sec. 3.1]
2.7	–
3.1	–
3.2.1	Extended version of the introduction of [21, Sec. 2]
3.2.2	–

Table A.1: Continuation

Thesis section	Paper section
3.2.3	Summary of [21, Sec. 2.5]
3.2.4	Summary of [21, Sec. 3]
3.2.5	Extended and structured version of [21, Sec. 2.3]
3.2.6	Summary and restructured version of [22, Sec. 4-6]
3.2.7	–
3.3.1	Extended version of [23, Sec. 2.5]
3.3.2	Rewritten version of [23, Sec. 3.2]
3.3.3	Summary of [23, Sec. 3.3]
3.3.4	Rewritten version of [22, Sec. 5.4]
3.4	Extended version of [23, Sec. 3.4]
3.5.1	Rewritten and restructured version of [21, Sec. 5.2]
3.5.2	Extended version of [22, Sec. 5.6] and rewritten and restructured version of [23, Sec. 3.5]
3.6.1	Summary of [21, Sec. 5.4]
3.6.2	Rewritten version of [21, Sec. 5.4.2]
3.6.3	Rewritten version of [22, Sec. 5.7] and rewritten and restructured version of [23, Sec. 3.5]
3.7.1	Extended and structured version of [21, Sec. 6.1]
3.7.2	Rewritten and restructured version of [22, Sec. 5.8]
3.8	Extended version of [22, Sec. 5.9] and rewritten and restructured version of [22, Sec. 3.6]
4	Rewritten version of [22, Sec. 1]
4.1.1	Rewritten version of [21, Sec. 8.4] and rewritten and restructured version of [22, Sec. 4.2]
4.1.2	Extended and restructured version of [22, Sec. 6.1] and of introduction of [22, Sec. 5]
4.1.3	Rewritten and restructured version of [22, Sec. 4.2]
4.1.4	Summary of [22, Sec. 5, 6.2]
4.1.5	Rewritten and restructured version of [22, Sec. 5.9, 6.3]
4.1.6	Rewritten version of [22, Sec. 6.4]
4.2.1	Extended version of [23, Sec. 3.1]
4.2.2	Rewritten and restructured version of [23, Sec. 4.2]
4.2.3	Rewritten and restructured version of [23, Sec. 3.3, 4.2]
4.2.4	Rewritten and restructured version of [23, Sec. 4.2]
4.2.5	Rewritten version of [23, Sec. 4.1]
4.2.6	Extended version of [23, Sec. 4.3]
4.2.7	Rewritten and restructured version of [23, Sec. 4.4]
4.2.8	Rewritten and restructured version of [23, Sec. 4.4]
4.2.9	Rewritten and restructured version of [23, Sec. 4.5]
4.2.10	Rewritten and restructured version of [23, Sec. 4.5]
4.2.11	–

A.3 Overview about Related Papers

There are publications by the author of this thesis that share the validation topic with the dissertation but do not form its main content. The first one focuses on implementation. The other three use methods of this dissertation for vehicle parameters in consumption simulations instead of the scenario inputs in safeguarding AVs. The following list includes a short summary of each paper, its classification into the thesis, and the contributions from the author of this thesis:

P5) **GSVF 2018 — Validation of X-in-the-Loop Approaches for Virtual Homologation of Automated Driving Functions [24]:**

Summary: This paper starts in an early stage of the development process, where an Adaptive Cruise Control (ACC) function is introduced and driving maneuvers are performed to calibrate a vehicle dynamics model. Then, this setup is used to execute test scenarios such as car following or emergency braking on a proving ground, at a test bed (DrivingCube), and in pure computer simulation (Model-in-the-Loop). The results from the simulation and test bed are compared against reality to validate them by means of graphical comparisons and statistical measures.

Classification: It forms the basis for complex test executions across several environments from an implementation point of view. However, from a methodological point of view, the applied techniques were superseded by the novel validation framework of this dissertation. Since the implementation is not the focus of this dissertation, this paper is not within its scope.

Contribution: “Stefan Riedmaier is the initiator and main author of this paper. He contributed the methodology and accompanied the whole research. Stefan Riedmaier created the virtual world from the real measurement data for the DrivingCube and Model-in-the-Loop approaches and performed the simulations.”

P6) **Taylor & Francis VSD 2020 — Potential of statistical model verification, validation and uncertainty quantification in automotive vehicle dynamics simulations: a review [25]:**

Summary: This survey paper gives an historical overview about vehicle dynamics model validation with 113 references and compares them to uncertainty methods.

Classification: This paper feeds into the vehicle dynamics literature in Chapter 2.4.1, but it only includes one engineering field with its methods.

Contribution: “S. R. introduced the initial idea of using uncertainty frameworks and statistical validation in the automotive domain. B. D. and S.R. further analysed uncertainty quantification methods and deduced their potential in the automotive domain.”

P7) **Elsevier Procedia CIRP 2020 — Statistical Model Verification and Validation Concept in Automotive Vehicle Design [26]:**

Summary: This research paper quantifies parametric, numerical and model-form uncertainties for the application of vehicle consumption simulations.

Classification: This paper applies uncertainty techniques that are also used in this dissertation. However, the consumption example differs significantly from the LKA type approval. In addition, this paper does not contain an extrapolation from validation con-

ditions to new application conditions, which will be a key property of the validation framework of this dissertation.

Contribution: “Stefan Riedmaier: Conceptualization, Methodology, Validation, Writing — Review & Editing.”

P8) **MDPI Applied Sciences 2021 — Statistical Validation Framework for Automotive Vehicle Simulations using Uncertainty Learning [27]:**

Summary: This research paper quantifies several sources of uncertainty in vehicle consumption simulations and extrapolates them to new parameter constellations.

Classification: This research paper extends [26] to extrapolation by applying the validation framework from [21] and this dissertation to the vehicle consumption example. Thus, it can be seen as a further use case focusing on vehicle parameters instead of scenario conditions of the LKA type approval.

Contribution: S.R.: methodology, writing — review & editing.

B Framework

B.1 Complete Framework

We presented the framework in Figure 3.1 of the main methodology chapter as a reduced version. It focuses on illustrating major framework blocks that are important for AV type approval by means of example plots. In addition, Figure B.1 shows the complete version of the framework. It also includes the verification and calibration domain and special blocks for macroscopic assessment and decision making across multiple scenarios.

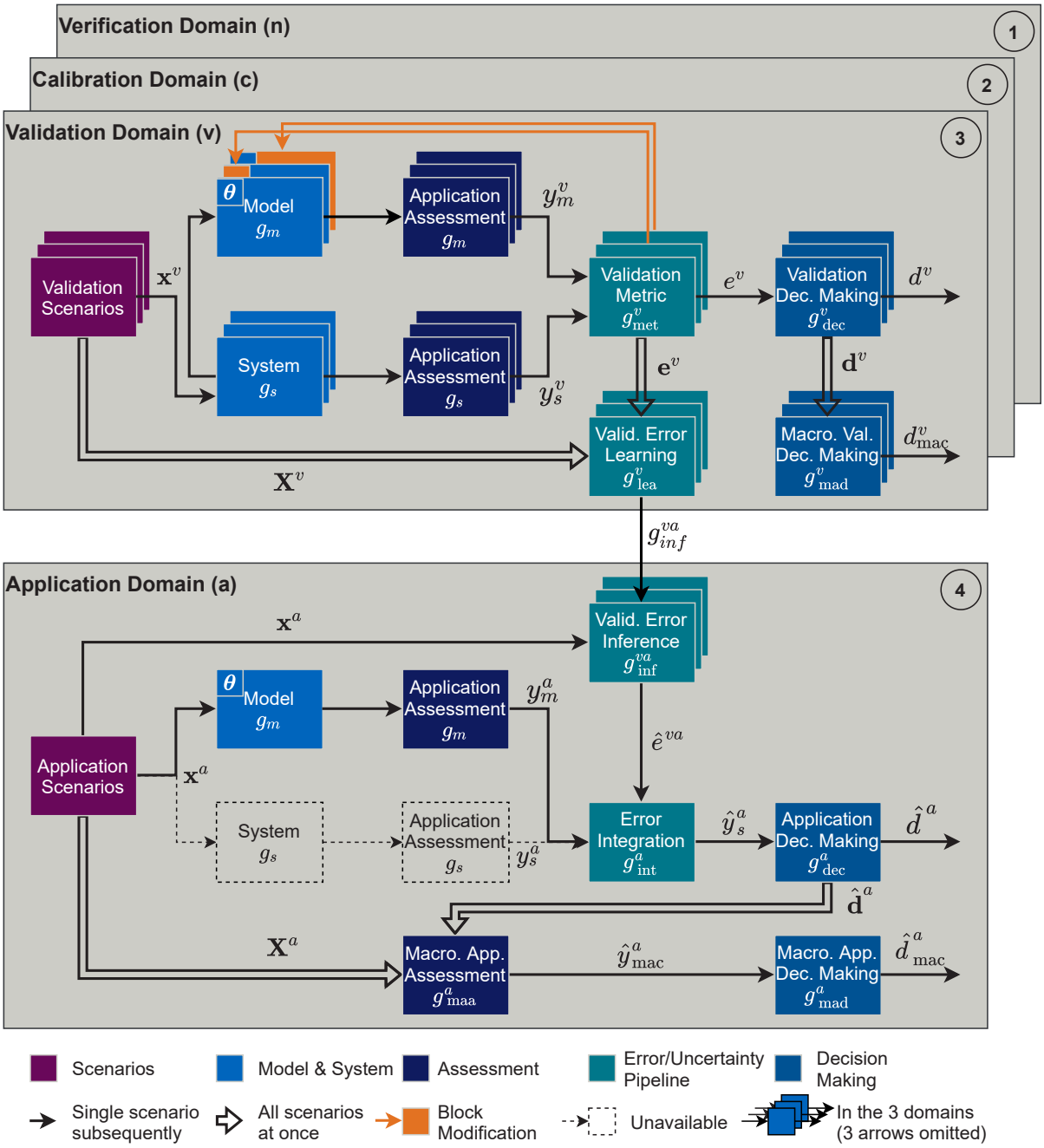


Figure B.1: VV&UQ framework representing a virtual-based process based on [21, Fig. 1]. The stacked blocks indicate that the same block appears for the verification, calibration, and validation domain, respectively. This holds true with the exception of a mathematical model instead of a system in the verification domain. The inferred errors from the three domains are merged in the error integration block of the application domain.

B.2 Framework Manifestations

This section extends the main part on the framework manifestations in Chapter 3.2.6. It adds the hierarchical, formal, and time-(in)variant manifestations to the (non-)deterministic manifestation.

B.2.1 Hierarchical Manifestation

Complex systems with multiple components offer model validation at varying levels of the system hierarchy. We have already stated that component-level validation is valuable, but that it should be combined with system-level validation to cover interactions. However, there are extreme cases where no physical experiments are possible at the system-level at all. This may be too dangerous as in nuclear reactors, too expensive as in aircraft and especially spacecraft, or generally not possible at an early stage of development. Sometimes there is a limited amount of system-level validation accompanied by thorough component-level validation. Nevertheless, these applications still aim to make statements about the safety of the overall system. This opens a large gap that requires extrapolation in the system hierarchy. In a single-component model, all domains of the framework address the same model and system. In a hierarchical model, there may be an independent verification, calibration, and validation of individual component models, while the actual model prediction takes place at the system level. This rises a challenge for the aggregation of errors and uncertainties in the framework in Figure B.1. It must connect separate verification, calibration, and validation domains of each component and possibly the entire system with one application domain on system level. The components might have a completely different set of input and output quantities than the overall system.

We presented three major approaches in Chapter 2.4 that address VV&UQ from component to system level: Output Uncertainty Approach [253], meta-model approach [283], and Bayesian network approach [192]. The latter, for example, solves the extrapolation challenge by defining joint model parameters or outputs as linking variables between the components and the system. They use both the vertical uncertainty pipeline from Figure B.1 during model validation and the inverse orange arrow during model calibration. They incorporate the uncertainties into the posterior distributions of the joint parameters or outputs so that they are reflected afterwards in the application domain. The principle can be extended by weighting the relevance of the component data via a sensitivity analysis [370] or an optimization problem [388]. Details regarding the embedding of the three approaches into our framework can be taken from [21]. Further information regarding Bayesian calibration can be found in [389–391] and regarding multi-physics coupling in [392–396].

B.2.2 Formal Manifestation

We have seen formal verification methods in Chapter 2.1.2 that rely on formal models. Reachability analysis [45] was an example from safeguarding AVs. Similar methods can be found in computer science, robotics, or control theory. There are approaches for formal model invalidation [397]. They invert the intention by checking, for example, by means of barrier certificates [398], whether the simulation and experimental data can be proven inconsistent. Finding a barrier is a proof of model invalidity, but finding no barrier is not a proof of model validity. The latter is impossible. Therefore, most researchers relax the hard invalidation and focus on model validation in the positive sense [399, 400]. They combine the formal theory with deterministic or non-deterministic simulations and are thereby related to the corresponding manifestations.

Deterministic simulations can be combined with time-series validation metrics, often referred to as closeness notions in this context, that yield a continuous distance between simulation and experimental trajectories. Among them are the (τ, ϵ)-closeness [401] or the Skorokhod metric [402]. Non-deterministic simulations can be combined with probabilistic closeness notions such as the Wasserstein metric [403]. The formal aspect can be integrated into the validation decision making block of the framework by calculating in each time step the probability of violating

a tolerance value [404], referred to as Probabilistically Robust Validation Certificate in [403]. Another possibility are the non-deterministic conformance testing approaches [44] presented in Chapter 2.4.1. They apply conformance notions [231] that directly yield a binary decision instead of the continuous distance from the closeness notions. This can be interpreted as a combination of the calibration metric and the calibration decision making blocks of the framework. It skips the validation domain with the vertical uncertainty pipeline of the framework and uses the inverse orange arrow within the calibration domain. It incorporates the uncertainties into the simulation model so that they are reflected afterwards in the application domain.

B.2.3 Time-(In)variant Manifestation

There are time-invariant simulations that predict static or quasi-stationary values. The majority of VV&UQ theory has been developed to target this type of simulation. In addition, there are complex dynamic systems that require time-variant simulations. Since these are more difficult for VV&UQ methods, often qualitative comparisons or the tolerance approach [182] from Chapter 2.4.1 are applied in the automotive and railway field. However, there are two principles how to deal with dynamic behavior to enable more sophisticated VV&UQ methods. The first principle is to represent the dynamic simulation by a set of stationary features so that the vast majority of classic VV&UQ approaches can be applied. Similar ideas can be seen, for example, when linearizing a non-linear system to apply mature linear control theory. The second principle is to actually calculate with time-variant errors. This does not suffer from any limitations, but it comes with additional challenges for the error aggregation within the framework. Both principles can take dynamic behavior into account, but they differ in the complexity of the dynamics and the corresponding VV&UQ approaches.

The first principle is illustrated in Figure B.2. The scenario, model, and assessment blocks look like time-invariant from the outside so that they keep a consistent interface and can be inserted into the overall framework in Figure B.1. The current state of the art contains several methods to represent time signals via scalar parameters and to extract characteristic values from time signals. The former is important on the side of the scenario inputs (parameter conversion) and the latter on the side of the assessment outputs (KPI assessment). For example, the test scenarios describing the environment as input of an AV are parameterized in most publications [9]. The target vehicle of an emergency braking scenario typically drives with a constant velocity until it starts braking from a defined point in time with a constant deceleration. Thus, its trajectory is represented by three scenario parameters. This can be extended to further ones via the 6-layer environment model [31]. The parameterization structures the infinite scenario space to enable logical scenarios with parameter ranges or distributions. On the output side, experts often extract characteristic values or KPIs such as minima, maxima, rise times, overshoot values, or gradients. Examples can be found in the standardized evaluation of vehicle dynamics maneuvers [183]. Automatic data reduction techniques are an alternative that extract low-dimensional features from the high-dimensional time signals. Many techniques [273, 405] such as Principal Component Analysis [350, 406, 407] or Karhunen-Loeve expansion [361] have already been applied in the VV&UQ literature to enable classical approaches. Time-series validation metrics are a further alternative to obtain the scalar results at the subsequent framework block. In any case, from the error learning on, the time-invariant representation continues as usual.

Recent papers following the second principle are in the minority. Ao et al. [200] and Hu et al. [277, 278] extend the Bayesian network approach to certain types of dynamic systems such as discrete time state-space models. In this case, the validation metric block of the framework relies

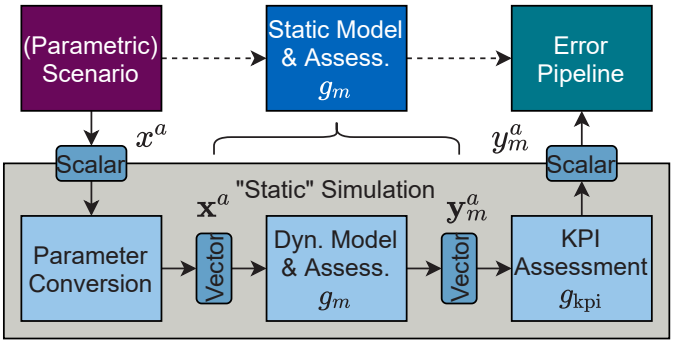


Figure B.2: Transformation of a dynamic simulation to a simplified static representation from [21, Fig. 8].

on full time-series metrics that convert the two time-series outputs of simulation and experiment to one resulting time series. The dependency of the current state on the previous time history and on hidden states significantly complicates the error learning pipeline of the framework. It can be solved by means of a combined error modeling with state estimation [277]. Further publications addressing VV&UQ approaches for dynamic systems can be found in [408].

C Extended Discussion

C.1 Further Discussion Points

This section targets aspects that were not the primary focus of this thesis and thus beyond the main discussion in Chapter 5.4.

Number of Scenarios

The number of application scenarios is a trade-off between the confidence in the AV safety and the simulation effort. The number of validation scenarios is a trade-off between the confidence in the simulation models and the testing effort dominated by the physical experiments. Both numbers depend heavily on the use case including the number of scenario parameters, their ranges, the distribution of scenarios within the space, the requirements for the confidence, and the available resources in terms of budget and road constraints. Additionally, the validation scenarios depend on the application scenarios themselves. In this thesis, we used expert judgment to define numbers that represent a reasonable balancing of all these factors for a first PoC. They are, however, by no means the result of an optimization procedure. The safeguarding and model validation literature rarely addresses the topic of the scenario count. Nevertheless, the work of [175, 176, 276, 345] can be seen as a starting base to derive optimal numbers.

Number of Repetitions

The number of repetitions should provide sufficient samples for the scatter of the experiment. The applied recommendation by Viehof [197] to always use at least 3 repetitions and 10 to 15 only for single scenarios originates from fitting a t-distribution to normally distributed data for a t-test as validation metric. We do not have this assumption, since we use all CDF steps for the area metric. Nevertheless, the principle of investigating single scenarios in detail is generally applicable. Following the recommendation, we ensured a minimum number for all scenarios and executed individual ones with a good location on the map multiple times. This can be improved in the future by first only executing the individual scenarios to analyze the scatter. The findings about the optimal number of repetitions and whether certain disturbances must be kept constant can then be used in a extensive second run of experiments.

Macroscopic Assessment

While most applications concentrate on a separate assessment of single scenarios, there are applications that extend this to an overall assessment about multiple scenarios. This is interesting for safety assessment approaches that make statements about the impact of AVs on traffic. However, this extension is rarely addressed in the current literature of the SBA [175–177]. Moreover, this type of application is not the subject of model validation research. Therefore, we have taken it into account in the design of the general framework in Figure 3.1

but not for its specific configuration for LKA type approval. The general framework highlights a constellation where the macroscopic assessment is based on all binary microscopic decisions. An example would be to weight the binary decisions with the real-world exposure of the application scenarios to combine them to an overall risk score. This constellation has the advantage that the macroscopic decisions do not have to be validated directly if the validation methodology can ensure the correctness of the microscopic decisions [21, Sec. 3.3.3]. If the macroscopic assessment relies on continuous microscopic assessment results instead of the binary decisions, the uncertainty pipeline must be extended to the macroscopic assessment.

Vehicle Variants

Another extension is to scale the validation process from one vehicle to several vehicle variants or types in homologation. It is not feasible to repeat the entire safety assessment each time but overly risky to safeguard only one variant. There is still no reasonable solution in between. Similar challenges exist in model validation. We can apply the validation framework of this thesis not only for external scenario parameters but also for internal vehicle parameters. The latter was done by Danquah et al. [27] for a consumption simulation. Combining both parameter types yields a joint space that allows sampling of validation and application scenarios, as well as interpolation and extrapolation across all dimensions. This can save a lot of effort, but it is still more than with only one vehicle. We might require additional strategies such as [409] to gain even more efficiency.

C.2 Framework Domains and Manifestations

This section targets the discussion of higher-level aspects that were not applied in the final PoC. This includes the verification and calibration domain, as well as the fully non-deterministic, time-variant, and hierarchical manifestations.

Verification and Calibration Domain

We integrated the verification and calibration domain into the general framework, since they are part of the entire VV&UQ process. However, we discarded them from the specific framework configuration, since numerical effects were negligible in the LKA use case and calibration contradicted the final viewpoint of the technical service. Nevertheless, the numerical effects should be assessed for a new use case involving, in particular, complex co-simulations [410, Fig. 4.5]. The calibration is interesting from the perspective of the car manufacturer. We recommend to use parameter measurements before inverse calibration methods, to collect independent calibration and validation data sets, and to apply Bayesian calibration to consider parametric uncertainties. Nevertheless, there are open research questions such as the optimal split between the calibration and validation data [292].

Non-Deterministic Manifestation

We have used a hybrid manifestation for the real PoC in this work. It quantifies a total error that includes both the actual model-form errors and the parasitic input errors. This is already an achievement, but it can be improved by separating the sources of uncertainty in the non-deterministic manifestation. However, applying the latter to the real traffic environment in the future poses many challenges. The environment consists of multiple influences within the 6-layer

model. Some of them may be extremely complex, such as a road friction surface, a car's radar reflection surface, or scenarios with multiple traffic objects of different types. These may no longer be represented with scalar values, but require time signals for speed profiles or maps for the surfaces. Similarly, vehicle models often contain more than a hundred parameters. The non-deterministic manifestation requires that they all be quantified and represented in a mathematical structure. The non-deterministic representations become more difficult because, for example, a time series input can no longer be described by a probability distribution but requires a stochastic process. A parameter can only be quantified as deterministic if it does not scatter and can be measured with highest precision, or if its influence can be proven negligible. The latter can be supported by a sensitivity analysis [381, 411]. Nevertheless, many uncertain parameters will remain. Several of these cannot be quantified by measurements because they may be deep inside the vehicle or, conversely, part of external road users that are not equipped with measurement devices. It should be noted again that the in-vehicle sensors should not be used as a reference, as they are part of the system under test. If a parameter cannot be measured, it requires the estimation of a conservative epistemic interval. If there are too many or too wide uncertainties, they do not lend themselves for decision making anymore. In the end, the non-deterministic manifestation is only applicable if the use case is restricted as in type approval and if there are extensive pre-analyses and extensive measurement campaigns as usual in numerical fields. It might be interesting to apply novel measurement approaches such as drone observations [71].

For the application of the non-deterministic manifestation, the second limiting factor after the quantification of uncertainties is the computational demand for their propagation. This was hardly feasible for the hybrid environment in our PoC. This would require MiL simulations, preferably running several times faster than real-time and parallelized to multiple clusters. Then, it might even be possible to increase the number of aleatory and epistemic samples. While we used 10 aleatory samples for the MMU study, they could be increased to 100 if a resolution of $99\,\%$ instead of $90\,\%$ is desired. Moreover, the comprehensive safeguarding of AVs will require a high mileage in simulation. It will hardly be possible to combine a large amount of distinct scenarios with a large amount of uncertainty samples for each of them. This motivates to develop intelligent strategies that adjust the number of uncertainty samples as needed. In the end, it is crucial to have the deterministic and hybrid manifestation as novel combinations until sometime all open issues of the non-deterministic manifestation are solved for complex systems.

Time-Variant Manifestation

We decided in Chapter 3.2.6 to apply a time-invariant framework manifestation by transforming dynamic to static behavior, since it offers significantly more literature and is in line with current safeguarding. However, highly dynamic constellations such as complex trajectory planners handling complex scenarios will become more important with higher automation levels. This will reach the limits of many safeguarding approaches that rely on a few scalar parameters and KPIs. When safeguarding evolves to tackle these challenges, there is also a demand for the specific framework configuration to evolve. An extension to time-variant scenario signals affects according to the framework in Figure 3.1 both the simulation and experiment, as well as the error learning and inference. First of all, the simulation environment and test equipment such as steering robots must offer the dynamic representation of the respective scenario parameters. With respect to the error learning and inference, there are two possibilities how to deal with arbitrary scenario signals. Automatic reduction techniques such as Principal Component Analysis can extract a compact set of input features that enable the traditional meta-modeling techniques.

The alternative is to apply more sophisticated techniques such as Long Short-Term Memories that can directly process time signals and perform the feature extraction and error modeling in one step. On the output side, we extracted the minimum distance to line as worst case KPI. This shows robustness due to the worst-case consideration and the PIs. Nevertheless, they can also cover erratic behavior only up to a certain degree. An extension to time-variant errors affects the entire error aggregation pipeline. Most validation metrics, in particular the non-deterministic ones such as the area metric, do not yet have a counterpart that calculates a time-variant error. Furthermore, the error learning and inference requires sophisticated techniques such as the combined error modeling and state estimation from [200]. In summary, the generic framework with its manifestations is prepared for the fully dynamic case, but it needs extensions in the specific configuration of individual blocks. It remains to be seen to what extent the robust transformation to time-invariant behavior is still applicable at higher automation levels or whether dynamic elements become necessary.

Hierarchical Manifestation

It is recommended to perform an isolated validation of components, such as the sensor and the vehicle dynamics model in the AV field. However, this is based on subjective expert tolerances, not quantitatively linked to the system level, and does not contain component interactions. It is possible to extrapolate component validation data to the system level if actual system-level tests are impossible, too risky, or lack enough data [192]. However, this must be performed with caution and is one of the largest VV&UQ challenges [292]. In the automotive field, we can execute validation experiments with the entire system including all component interactions. This was the focus of our PoC for the final system output from the perspective of a technical service. It can be extended to additionally validate internal component outputs within the system loop. This is covered by our methodology and interesting for developers to locate the modeling errors.