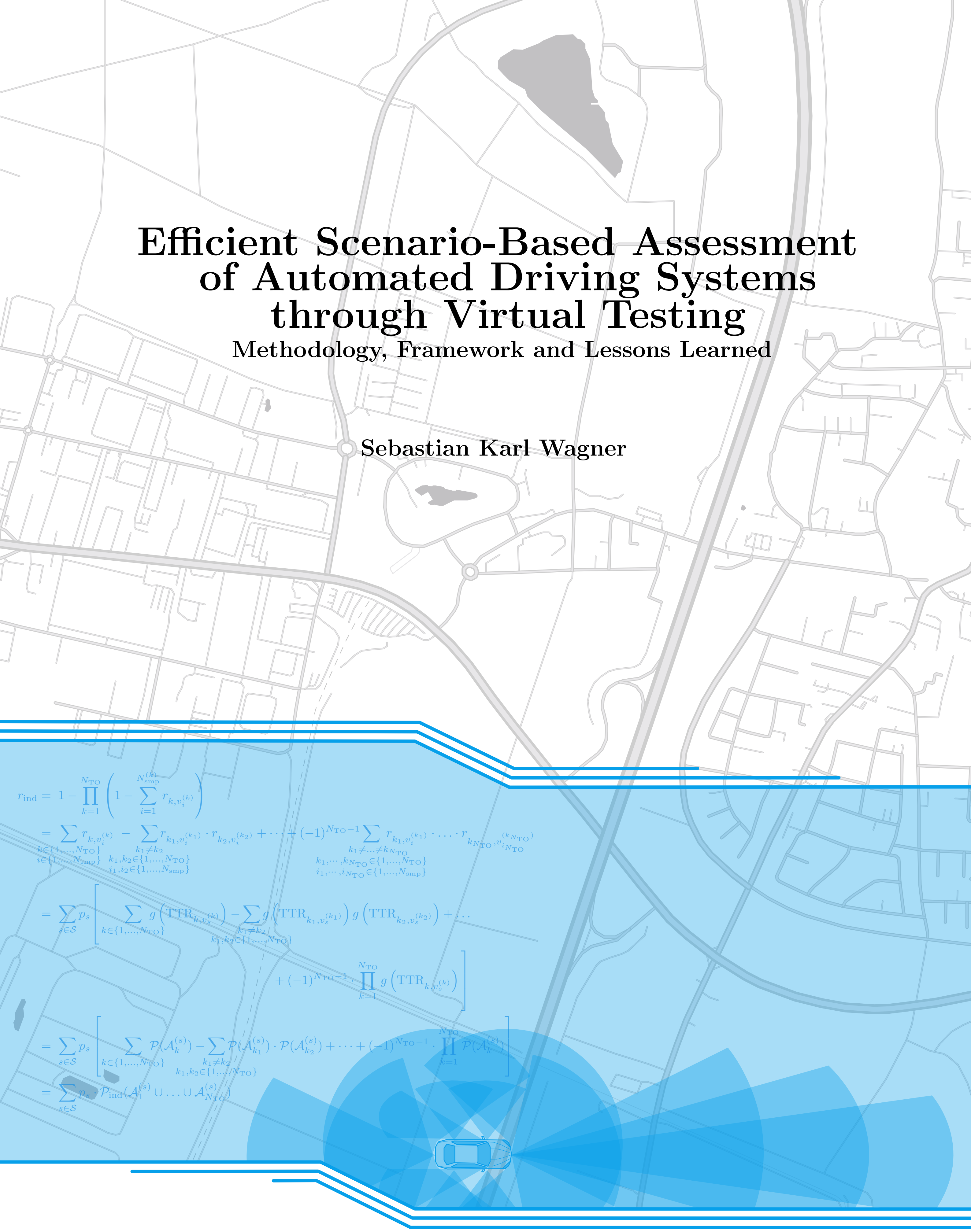

$r_{\text{ind}} = 1 - \prod_{k=1}^{N_{\text{TO}}} \left(1 - \sum_{i=1}^{N_{\text{smp}}^{(k)}} r_{k,v_i^{(k)}}\right)$
$= \sum_{\substack{k\in\{1,\dots,N_{\text{TO}}\}\\ i\in\{1,\dots,N_{\text{smp}}\}}} r_{k,v_i^{(k)}} - \sum_{\substack{k_1\neq k_2\\ k_1,k_2\in\{1,\dots,N_{\text{TO}}\}\\ i_1,i_2\in\{1,\dots,N_{\text{smp}}\}}} r_{k_1,v_i^{(k_1)}} \cdot r_{k_2,v_i^{(k_2)}} + \dots + (-1)^{N_{\text{TO}}-1} \sum_{\substack{k_1\neq\dots\neq k_{N_{\text{TO}}}\\ k_1,\cdots,k_{N_{\text{TO}}}\in\{1,\dots,N_{\text{TO}}\}\\ i_1,\cdots,i_{N_{\text{TO}}}\in\{1,\dots,N_{\text{smp}}\}}} r_{k_1,v_i^{(k_1)}} \cdot \dots \cdot r_{k_{N_{\text{TO}}},v_{i_{N_{\text{TO}}}}^{(k_{N_{\text{TO}}})}}$
$= \sum_{s\in\mathcal{S}} p_s \left[\sum_{k\in\{1,\dots,N_{\text{TO}}\}} g\left(\text{TTR}_{k,v_s^{(k)}}\right) - \sum_{\substack{k_1\neq k_2\\ k_1,k_2\in\{1,\dots,N_{\text{TO}}\}}} g\left(\text{TTR}_{k_1,v_s^{(k_1)}}\right) g\left(\text{TTR}_{k_2,v_s^{(k_2)}}\right) + \dots \right.$
$\left. + (-1)^{N_{\text{TO}}-1} \cdot \prod_{k=1}^{N_{\text{TO}}} g\left(\text{TTR}_{k,v_s^{(k)}}\right) \right]$
$= \sum_{s\in\mathcal{S}} p_s \left[\sum_{k\in\{1,\dots,N_{\text{TO}}\}} \mathcal{P}(\mathcal{A}_k^{(s)}) - \sum_{\substack{k_1\neq k_2\\ k_1,k_2\in\{1,\dots,N_{\text{TO}}\}}} \mathcal{P}(\mathcal{A}_{k_1}^{(s)}) \cdot \mathcal{P}(\mathcal{A}_{k_2}^{(s)}) + \dots + (-1)^{N_{\text{TO}}-1} \cdot \prod_{k=1}^{N_{\text{TO}}} \mathcal{P}(\mathcal{A}_k^{(s)}) \right]$
$= \sum_{s\in\mathcal{S}} p_s \cdot \mathcal{P}_{\text{ind}}(\mathcal{A}_1^{(s)} \cup \dots \cup \mathcal{A}_{N_{\text{TO}}}^{(s)})$

Efficient Scenario-Based Assessment of Automated Driving Systems through Virtual Testing

Methodology, Framework and Lessons Learned

Sebastian Karl Wagner

TECHNISCHE UNIVERSITÄT MÜNCHEN
Fakultät für Informatik
Lehrstuhl für Robotik, Künstliche Intelligenz und Echtzeitsysteme

TECHNISCHE UNIVERSITÄT MÜNCHEN
Fakultät für Informatik
Lehrstuhl für Robotik, Künstliche Intelligenz und Echtzeitsysteme

Efficient Scenario-Based Assessment of Automated Driving Systems through Virtual Testing

Methodology, Framework and Lessons Learned

Sebastian Karl Wagner

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Prof. Dr.-Ing. Jörg Ott
Prüfer der Dissertation: 1. Prof. Dr.-Ing. habil. Alois C. Knoll
2. Prof. Dr. techn. Daniel Watzenig

Die Dissertation wurde am 18.08.2020 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 26.01.2021 angenommen.

Abstract

Even though Automated Driving (AD) is among the most anticipated future technologies, the task of delivering safe solutions to future automation is the automotive Original Equipment Manufacturers (OEMs) primary problem child. The race towards the first truly automated production-ready vehicle started already a decade ago and is still ongoing due to technical difficulties. While it promises the revolution of mobility, improved traffic efficiency and safety, the latter is the hardest to achieve and proof. Attached to the safety are significant financial risks for car manufactures caused by liability for accidents and customer acceptance. Even more importantly, ethical concerns in the form of human harm arise. Existing methods for proofing the safety of Advanced Driver Assistance System (ADAS) are stretched to their limits, and research towards new assessment concepts for AD is necessary.

This work shows that a scenario-based assessment concept can put relief on the infeasible amount of real-world testing that would be required by a traditional concept. Therefore, a methodology is elaborated and transformed into a novel proof of concept framework, which includes virtual testing for better test coverage and robustness. While virtual testing is not considered for safety testing prior due to validity concerns, this framework enables its use through re-simulation of real-world tests and local cross-verification. The inclusion of virtual testing opens up new possibilities, such as test case variation to increase the test amount. Further, the framework provides tooling for variation along with a definition of a scenario space for test coverage estimation as well as an automated evaluation of test criteria with a novel measure of risk.

Given such a framework, a much more safety efficient and less risky assessment of AD functions is possible. While there are still challenges to overcome, this methodology provides the proof that scenario-based testing can escape the approval trap the industry is currently facing. Furthermore, through the proof of concept implementation of all necessary modules for this approach, the remaining work necessary for a production-ready application is well known. Consequently, the release of AD in the near future is one step closer.

Zusammenfassung

Auch wenn sich automatisiertes Fahren in den Reihen der am meisten erwarteten Zukunftstechnologien befinden, so ist die Aufgabe sichere Lösungen zukünftiger Automatisierungstechnologien zu finden auch das größte Sorgenkind der Fahrzeughersteller. Das Rennen um das erste komplett automatisierte Fahrzeug startete bereits vor einem Jahrzehnt, ist aber aufgrund technischer Hindernisse bis heute noch nicht abgeschlossen. Während es die Revolution der Mobilität sowie Verbesserung der Verkehrseffizienz und -sicherheit verspricht, ist vor allem letzteres schwer zu erreichen und zu beweisen. Mit dieser Sicherheit einhergehend sind große finanzielle Risiken für Fahrzeughersteller durch Haftbarkeit für Unfälle und mangelnde Kundenakzeptanz. Aber noch viel wichtiger, es kommen auch ethische Bedenken durch die Gefahr für den Menschen auf. Aktuelle Methoden zur Absicherung von fortgeschrittenen Fahrassistenzsystemen stoßen an ihre Limits und Forschung in Richtung neuer Methoden für das automatisierte Fahren ist notwendig. Diese Arbeit zeigt, dass ein szenarienbasiertes Absicherungskonzept Erleichterung bei den ansonsten durch traditionelle Vorgehensweisen geforderten Menge an Realfahrttests schaffen kann. Dafür wird eine Methodik unter Miteinbeziehung von virtuellen Tests, die die Testabdeckung und Robustheit erhöhen, erarbeitet und in Form eines Machbarkeitsnachweises als Framework umgesetzt. Obwohl virtuelles Testen bisher nicht für sicherheitsrelevantes Testsen aufgrund von Zweifeln an der Validität herangezogen wurde, ermöglicht dieses Framework die Einbindung durch Resimulation realer Tests und lokaler Kreuzverifikation. Diese Miteinbeziehung virtueller Tests eröffnet neue Möglichkeiten wie zum Beispiel Testfallvariation, um die Menge an Tests zu erhöhen. Des Weiteren beinhaltet das Framework die Mittel zur Durchführung einer Variation zusammen mit der Beschreibung eines Szenarien Raumes zur Abschätzung der Testabdeckung und eine neuartige Metrik zur automatisierten Auswertung der Tests.

Mit Hilfe eines solchen Frameworks ist eine viel effizientere und risikofreie Absicherungsbewertung einer automatisierten Fahrfunktion möglich. Obwohl es immer noch Herausforderungen zu bewältigen gibt, stellt diese Methodik den Beweis dar, dass szenarienbasiertes Testen der Freigabefalle, vor der die Industrie im Moment steht, entkommen kann. Des Weiteren ist durch die Implementierung aller nötigen Module als Machbarkeitsnachweis die übrige Arbeit, die für einen serienmäßigen Einsatz noch von Nöten ist, bekannt. Folglich rückt die Freigabe des automatisierten Fahrens ein Stück näher.

Acknowledgements

As given trough the declaration under oath, a dissertation is meant to be written by oneself. However, during the long journey towards this final document, one is never alone. There are several people throughout this way who directly and indirectly have their influence on both the scientific work and the personal living conditions that are crucial for the progress. Here I want to thank those people who stood out during this period and made this dissertation possible as it is.

First, I want to thank my supervisor Prof. Knoll. He always knew the right way to support the scientific work and keep me motivated and focused. Discussions with him were always productive and helped to push the next step towards this document.

Also, students writing master and bachelor thesis constitute a valuable contribution to this work. Their brilliant ideas and motivated execution of those always brought the research a step further to the goal.

A comfortable socioenvironment is as essential as technical discussions to keep oneself motivated and healthy. Especially all my colleagues at our Parkring lab knew well how to form a friendly community that always helped each other. Without you, these years would not have been that fun.

The project behind this scientific research was funded entirely by the Bayerische Motorenwerke AG (BMW) Group, which I am particularly grateful for. Behind this funding is a group of colleagues that were especially helpful with tips, scientific exchange, and providing tooling. The acknowledgment here goes in particular to the colleagues at LT-6 and Mohammad as the bridgehead, who provided the fund, necessary tooling, and valuable technical exchange. Then to the whole EV department for providing an endpoint for this research, discussion, and dialogue, as well as conducting test drives for real-world evaluation of this work.

Special thanks goes to the two doctoral candidates working on related topics, Korbi and Martin. Together we formed a team that could always benefit from each other and helps with the elaboration of publications like no other. Also, without the friendship, we build in our self-named ”tech office,” this time would not have been the fun it was.

A way too underrated position of a dissertation is the mentor. During the last years, I got to know how vital this role really is and that I got assigned the perfect person for this. Tom has always been there for me, even from the remotest of all locations. He helped not only with countless sessions discussing methodologies and results but also solving the minor everyday problems in life was no problem for him.

The last bulk of thanks is dedicated to my closest proximity, my family. My sister, with her own growing family, my father, and mother, and my grandparents always managed to keep me motivated over the last years in their own particular ways. Troublesome times come and go, but the family remains. Last but not least, one person has proven to become more and more a member of this family. Thank you, Katharina, for going through this at my side, always accepting that I put so much of my time into this that I could otherwise spend with you and for the assistance also in troublesome times. I know this can not be taken for granted, and I am grateful to have you as my girlfriend.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

8VM	Eight-Vehicle-Model	114, 115, 121, 122
ACC	Adaptive Cruise Control	7, 15, 18, 21, 39, 94
AD	Automated Driving	i, ix, 1, 3–5, 7–23, 25, 28–31, 34–48, 50, 57, 69–71, 74–82, 85, 87, 91, 92, 94, 97–100, 102, 111, 128–131, 133–137
ADAS	Advanced Driver Assistance System	i, ix, 3, 4, 7, 8, 10, 11, 13–15, 19, 21, 34, 36–42, 44, 46, 77, 134
ASIL	Automotive Safety Integrity Level	37
AWGN	Additive White Gaussian Noise	174
BA	Brake Assist	3
BMW	Bayerische Motorenwerke AG	iii, 1, 18, 91
CMU	Carnegie Mellon University	18
CSV	Comma-Separated Values	113
CTRA	Constant Turn Rate and Acceleration	ix, 50, 51, 53
DARPA	Defense Advanced Research Projects Agency	17
dGPS	Differential GPS	82, 83
DIL	Driver-in-the-Loop	22, 23, 41, 42
EGO	Vehicle under Test	ix, xxii, xxiii, 19, 20, 25, 27, 31, 33, 34, 48, 49, 57–62, 66, 68–75, 79, 81, 83–85, 87, 89–93, 97–99, 101, 110, 111, 113–124, 126, 127
EPS	Electronic Power Steering	3
ETTC	Enhanced Time-to-Collision	ix, 31, 32, 48
EU	European Union	3
EVT	Extreme Value Theory	41
FCW	Forward Collision Warning	21

FOT	Field Operational Test	21, 23, 24, 41, 42, 78, 79
GM	General Motors Company LLC	16
GPS	Global Positioning System	xvii, 19, 82, 83
GT	Ground Truth	xix, xxi, 83, 84, 88–94, 96–101, 109–111, 134, 135, 137, 163, 166
GUI	Graphical User Interface	96, 166
HAC	Hierarchical Agglomerative Clustering	x, xi, 124–126, 131, 177–181
HD	High-Definition	19
HIL	Hardware-in-the-Loop	22, 24, 25, 38, 41, 79
HOL	High Order Logic	35
IEC	International Electrotechnical Commission	36
IMU	Inertial Measurement Unit	83
ISO	International Organization for Standardization	1, 36–38
JSCEN	Scenario description in JSON file format	x, xiii, 87, 89–94, 96, 97, 104, 109, 117, 163
JSON	JavaScript Object Notation	xviii, 87
KDE	Kernel Density Estimation	42
KPI	Key Performance Indicator	47, 48, 50, 69–76, 79, 80, 94–96, 103, 111
Lidar	Light Detection and Ranging	19, 82–84
LKAS	Lane Keeping Assistance System	15, 18
LTL	Linear Temporal Logic	35
NCC	Nonlinear Correlation Coefficient	xi, xxiii, 106–108, 171–174
NDS$_1$	Naturalistic Driving Study	ix, x, 21, 23, 41, 44, 51, 53, 57, 71–74, 76, 78, 79, 105, 108, 112, 131
NDS$_2$	Navigation Data Standard	28
ODD	Operational Design Domain	15, 16, 21, 39, 43, 46, 77, 101, 112, 129, 131, 137
ODE	Ordinary Differential Equation	50
OEM	Original Equipment Manufacturer	i, 1, 4, 7, 18, 28, 36, 37, 42, 43, 47, 78, 137
OSI	Open Simulation Interface	92

PCA	Principal Component Analysis	x, xi, xxi–xxiii, 104–106, 108, 109, 111, 131, 134, 135, 170–172
PCC	Pearson Correlation Coefficient	xi, 106–108, 172–175
PEGASUS	Projekt zur Etablierung von generell akzeptierten Gütekriterien, Werkzeugen und Methoden sowie Szenarien und Situationen zur Freigabe hochautomatisierter Fahrfunktionen	12, 42, 78, 114
PMD	Predicted-Minimum-Distance	xx, 34
PROMETHEUS	PROgraMme for a European Traffic of Highest Efficiency and Unprecedented Safety	17
Radar	Radio Detection and Ranging	18, 19, 82, 84
rGT	Resimulated GT	xxi, 92–94, 97–99, 110, 111
RPC	Remote Procedure Call	97
rSD	Resimulated SD	xxi, 92–94, 97–99
SAE	Society of Automotive Engineers	ix, 15, 17, 18, 45, 134, 137
SD	Sensor Data	xix, xxi, 83, 84, 88, 90–94, 96–101, 135, 137, 166
SIL	Software-in-the-Loop	22, 24, 38, 79
SNR	Signal to Noise Ratio	174
SOTIF	Safety of the Intended Functionality	37
SRS	Supplementary Restraint System	21
STA	Situation Threat Assessment	31, 34, 57
StVO	Straßenverkehrsordnung	33, 35, 71
SUMO	Simulation of Urban Mobility	29
SVD	Singular Value Decomposition	170
TD	Test Drive	x, 21, 23, 24, 38, 42, 71, 74–76, 78, 79, 83, 86, 87, 92
TDD	Test Driven Development	38
THW	Time-Headway	ix, 31, 33, 34, 48, 70–73, 75
TO	Traffic Object	ix, xxi–xxiv, 18–20, 23, 25–27, 31–34, 48–53, 56–62, 64–76, 79, 81–85, 87, 89–94, 97–99, 104, 109–112, 114–117, 122–124, 126, 127

TPMD	Time-to-PMD	34
TRL	Technology Readiness Level	1, 18
TTB	Time-to-Break	33, 72, 73, 75
TTC	Time-to-Collision	ix, 31–35, 48, 57, 59–63, 70–73, 75
TTK	Time-to-Kickdown	34, 72, 73, 75
TTR	Time-to-React	ix, xxiii, 33, 34, 48, 49, 57, 61–66, 69–73, 75
TTS	Time-to-Steer	34, 62, 63, 72, 73, 75
UniBW	Universität der Bundeswehr München	17
USA	United States of America	8, 34, 40
V2V	Vehicle-to-Vehicle Communication	19, 27, 28
V2X	Vehicle-to-X Communication	19, 27, 28
VaMP	Versuchsfahrzeug für autonome Mobilität PKW	17
VIL	Vehicle-in-the-Loop	23, 24, 38, 42, 79
WHO	World Health Organization	3
WTTC	Worst-Time-to-Collision	32, 35, 70, 72, 73, 75
WTTR	Worst-Time-to-React	70, 72, 73, 75
XML	Extensible Markup Language	28, 29, 86, 97

Symbols

H	Information Entropy	172
I	Mutual Information	173
N_C	Number of components for the PCA	106, 108, 111, 170
N_F	Number of features for the PCA	106, 170
N_{TO}	Number of TOs in a scene	57, 58
N_{col}	Number of possible collisions in a scene	61
$N_{S,\theta}$	Number of spline segments in θ_S	88
$N_{S,v}$	Number of spline segments in v_S	88
$N_{\mathrm{scenarios}}$	Number of scenarios	125
N_{scenes}	Number of scenes in a scenario	122
N_{pt}	Number of data points	52
N_{smp}	Number of samples	54, 56, 88, 170, 172, 178
$N_{smp}^{(k)}$	Number of predicted trajectories for the kth TOs	57, 70
T_{PNR}	Point of no return, where a collision is unavoidable	63, 64, 69–71
T_{max}	Maximal time until a collision is of interest	64, 70
T_{pred}	Prediction time	55, 58, 70
T_{step}	Prediction step size	55, 58, 61, 70
$\Delta\mathrm{FIT}_{\mathrm{GT}}$	GT fitting error	91, 98
$\Delta\mathrm{FIT}_{\mathrm{SD}}$	SD fitting error	91, 97–99
$\Delta\mathrm{FIT}_{\mathrm{rGT}}$	rGT fitting error	92, 98
$\Delta\mathrm{FIT}_{\mathrm{rSD}}$	rSD fitting error	94, 98
$\Delta\mathrm{PIPE}_{\mathrm{GT}}$	GT pipeline error	97, 98, 100, 101
$\Delta\mathrm{PIPE}_{\mathrm{SD}}$	SD pipeline error	97, 98
$\Delta\mathrm{SIM}_{\mathrm{GT}}$	GT simulation error	92, 97–99, 101
$\Delta\mathrm{SIM}_{\mathrm{SD}}$	SD simulation error	92, 97–99
$\Delta\mathrm{S}$	Sensor error	84, 91, 98–100
Δc_i	Variation range in one direction of the ith component	109, 111
Δt	Time difference	99
$\Delta t_{\mathrm{react}}$	Time difference in reaction to and incident in a scenario	98, 99
$\boldsymbol{\theta}_S$	Base point value of the heading spline θ_S	88

$\mathbf{D}$	Pairwise distance matrix of the clustering input data	178, 179
$\mathbf{X}$	A data set in the feature space of the PCA	170
$\mathbf{a}_\theta$	First internal parameters of the heading splines	89
$\mathbf{a}_v$	First internal parameters of the velocity splines	88, 89
$\mathbf{b}_\theta$	Second internal parameters of the heading splines	89
$\mathbf{b}_v$	Second internal parameters of the velocity splines	88, 89
$\mathbf{c}_\theta$	Third internal parameters of the heading splines	89
$\mathbf{c}_v$	Third internal parameters of the velocity splines	88, 89
$\mathbf{d}_\theta$	Fourth internal parameters of the heading splines	89
$\mathbf{d}_v$	Fourth internal parameters of the velocity splines	88, 89
$\mathbf{h}$	List of hierarchical dendrogram entries	178
$\mathbf{l}$	labels of the clustering input data	178, 179
$\mathbf{p}$	Parameter vector for both velocity and heading splines	88, 89
$\mathbf{p}_\theta$	Parameter vector for the heading spline θ_S	88
$\mathbf{p}_v$	Parameter vector for the velocity spline θ_S	88
$\mathbf{t}_M$	Time data from a Measurements	88
$\mathbf{t}_{S,\theta}$	Base point time of the heading spline θ_S	88
$\mathbf{t}_{S,v}$	Base point time of the velocity spline v_S	88
$\mathbf{v}_S$	Base point value of the velocity spline v_S	88
$\mathbf{x}_M$	Position data on the x-axis from a measurement	88
$\mathbf{y}_M$	Position data on the y-axis from a measurement	88
$\mathcal{A}_k^{(s)}$	Event of having an accident with the kth TO in scene $s \in \mathcal{S}$	67, 68
$\mathcal{A}_{\mathrm{acc}^{(s)}}$	Event of having an accident with any TO in scene $s \in \mathcal{S}$	68
$\mathcal{S}$	Permutation of as possible evolutions of a scene	xxii, xxiii, 65–68
$\mathcal{V}$	Set of samples	54, 55
$\mathcal{V}_k$	Set of predicted trajectories for the kth TOs	57, 58, 65
TO_k	Index for the kth TO in the scene	58, 59, 61
μ	Mean of a distribution	83, 90, 93
μ_ω	Mean of the yaw rate ω	52
μ_a	Mean of the acceleration a	52
$\mu_{\mathrm{EGO},\theta}$	Mean of the EGO vehicles's heading error	83
$\mu_{\mathrm{EGO},d}$	Mean of the EGO vehicles's distance error	83
$\mu_{\mathrm{EGO},v}$	Mean of the EGO vehicles's velocity error	83
$\mu_{\mathrm{TO},\theta}$	Mean of a TO's heading error	84
$\mu_{\mathrm{TO},d}$	Mean of a TO's distance error	84
$\mu_{\mathrm{TO},v}$	Mean of a TO's velocity error	84

ω	Change of heading θ or yaw rate	xxii, xxiii, 50, 51, 53–55, 94
ω_0	Change of heading θ or yaw rate at $t = 0$	51, 52, 54
σ	Standard deviation of a distribution	83, 90, 93
σ_ω	Standard deviation of the yaw rate ω	52
σ_a	Standard deviation of the acceleration a	52
$\sigma_{\mathrm{EGO},\theta}$	Standard deviation of the EGO vehicles's heading error	83
$\sigma_{\mathrm{EGO},d}$	Standard deviation of the EGO vehicles's distance error	83
$\sigma_{\mathrm{EGO},v}$	Standard deviation of the EGO vehicles's velocity error	83
$\sigma_{\mathrm{TO},\theta}$	Standard deviation of a TO's heading error	84
$\sigma_{\mathrm{TO},d}$	Standard deviation of a TO's distance error	84
$\sigma_{\mathrm{TO},v}$	Standard deviation of a TO's velocity error	84
θ	Heading or yaw angle around the objects z-axis	xxiii, 50, 51, 83, 88, 90, 93, 97, 99
θ_S	Spline representation of the heading θ	xxi, xxii, 88
a	Acceleration	xxii, xxiii, 31, 32, 50, 51, 53–55
a_0	Acceleration at $t = 0$	51, 52, 54
b	Rank for the computation of the NCC; Basis of the Halton Sequence	173, 175–177
c	A component calculated through the PCA	171, 172
$c_{i,\max}$	Maximum value of components i	108, 109, 111
$c_{i,\min}$	Minimum value of components i	108, 109, 111
d	Distance between two further specified objects	32, 58, 59, 83, 90, 93, 97, 121
f	A feature used for the PCA	171, 172
f_ω	Distribution of the yaw rate ω	52
f_a	Distribution of the acceleration a	52
$f_{a,\omega}$	Combined distribution of acceleration a and yaw rate ω	53
m	Parameter for controlling the steepness of the TTR weighting function	64, 70
n	Counter variable in sums and iterations	52
n_{col}	Index for a predicted collision in the scene	61
p	Probability	172
p_s	Probability of a scene permutation $s \in \mathcal{S}$	65, 67, 68
$p_{k,v_i^{(k)}}$	Probability of the trajectory $v_i^{(k)}$ from the kth TO in the scene	64, 65
r	Radius of some further specified object	58, 59
r	Scene or scenario risk	69, 71, 73–75, 94, 128

r_{dep}	Scene risk considering dependencies between TOs	65
r_{ind}	Scene risk assuming independency between TOs	66
r_{TO_k}	Risk imposed from the kth TO in the scene	65
s-Offset	Distance along a road since its start	87
t	Time	xxiii, xxiv, 50, 51, 58, 59, 61, 88, 106, 114, 174
t-Offset	Deviation to a lane's center	87
v	Velocity at $t = 0$	xxiv, 31–33, 50, 51, 83, 88, 90, 93, 97, 99
v_0	Velocity	51–54, 56
v_S	Spline representation of the velocity v	xxi, xxii, 88
v_{lat}	Lateral velocity v of a TO with respect to the road lane	xxiv, 114
v_{thresh}	Threshold of the lateral velocity v_{lat} used for the lane change maneuver detection	114
$v_i^{(k)}$	The ith predicted trajectory for the kth TOs	57–59
x	Position along the x-axis in a Cartesian coordinate frame	xxii, xxiv, 50, 51, 56, 105, 106, 172–174, 176, 181
y	Position along the y-axis in a Cartesian coordinate frame	xxii, xxiv, 50, 51, 106
z	Position along the z-axis in a Cartesian coordinate frame	xxiii, xxiv

1. Introduction

The urge for automation accompanies the advancement of humanity already since ancient times [1]. Several impulses in history, such as the industrial revolution [2] and electrification [3], kept this drive alive until today. Ever since the first gasoline-powered production automobile in 1996 [4], also the automotive industry is not spared from this trend. This not only affects the production of vehicles itself by the famous introduction of the assembly line in 1912 [5] but also the way a human interacts with the car.

Thus about 80 years ago arose the idea of automating the driving task [6]. Several spikes in research did not result in fully automated vehicles until this day because of missing technical prerequisites. Concrete research towards automated vehicles started again in the 80s [7, 8, 9]. With the achievement of Technology Readiness Level (TRL) 4 [10, 11], meaning experts consider it as ready for future market release, also the race among Original Equipment Manufacturers (OEMs) towards the release of production-ready Automated Driving (AD) started. This race between renowned companies such as Google [12, 13], Tesla [14, 15], Bayerische Motorenwerke AG (BMW) [16, 17], and Toyota [18], to name a few can be observed in media frequently.

Indeed, the urge to automate driving is not only impelled by the laziness of humanity. One can expect multiply benefits from this functionality [19]. AD is expected to reduce traffic jams and increase fuel efficiency through optimized and cooperative traffic flow control. Also, it can oppose the demographic bias in mobility by allowing age groups into traffic that are not yet or not anymore able to drive themselves. Lastly and most important is the aspect of traffic safety. The claim of accident reduction and improved safety is unanimously AD's most prominent benefit and most significant challenge.

Up to this date, it is unclear how one can ensure that AD is safer than a human driver. Experts agree that a statistical series of tests as done for previous generations in the automotive industry is not feasible [19]. Some sources state the necessity of $5 \cdot 10^9 km$ [20] under test for every development lifecycle derived from the International Organization for Standardization (ISO) norm 26262 [21]. Since the appearance of this issue, whole new research towards alternate assessment methods for AD arises. Likewise, this thesis aims to contribute to the answer to the assessment problem.

1.1. Background and Motivation

To motivate the safety aspect of road traffic in general, an elaboration of some statistics follows. Contrary to popular belief, one is almost 1000 times more likely to have an accident on the road than in air traffic [22]. Still, about 1 million people are killed worldwide in traffic [23]. For the age group of 15 to 29, it is the leading cause of mortality [24]. Given that this age group is partly below the legal age of driving implicates that also passive participants are included in these numbers. The share of these vulnerable road users, such as pedestrians or cyclists, steadily increases [25], making up half of the deaths today [26]. In general, the number of fatalities related to road traffic is a good indicator of traffic safety [27].

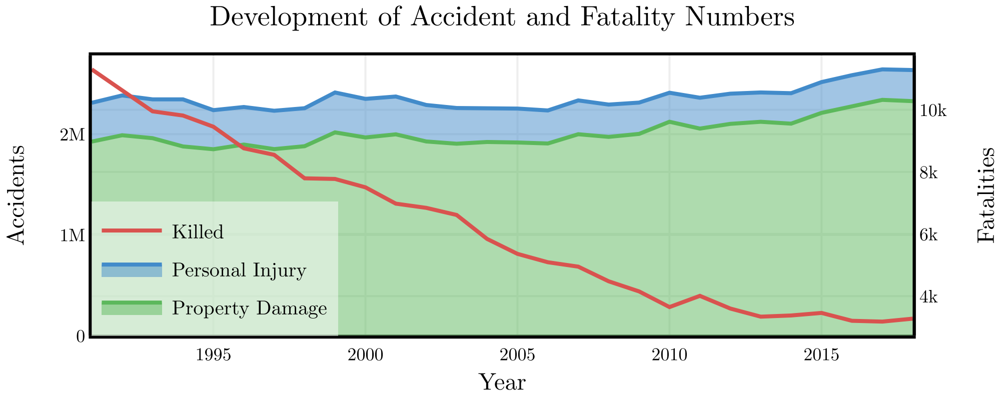


Figure 1.1.: Statics of traffic accidents and fatalities from 1991 to 2018 in Germany [28, 29, 30]. The blue and green plots stack the accidents with personal and property damage, respectively. In red the number of fatalities is shown.

Figure 1.1 displays the evolution of accidents and fatalities based on accumulated official numbers in Germany from [28, 29, 30]. Blue and green stack the count of traffic accidents for property damage only and accidents involving personal injury. The red line indicates the number of fatalities. Note that the number of accidents and the number of deaths are bound to a different vertical axis. In general, the number of accidents slowly increases over the years due to the increasing number of road vehicles. Concomitantly, the share of accidents involving personal injury as well as the number of fatalities decreases. However, the reduction of fatalities seems to have stagnated over the past decade. Both the decline and the stagnation are now considered more carefully.

The number of accidents increases not as much as vehicles on the road and their driven milage [31]. Additionally, the number of fatalities even decreases. Consequently, there must have been

countermeasures that show effectiveness. In the automotive context, these are passive and active safety features. The most prominent and effective members of passive safety are seat belts and airbags. However, their mandatory inclusion in production vehicles started before the scope of Figure 1.1. This limits the expressiveness of their effect on the decreasing fatality in this scope. Therefore, active safety systems seem to be the primary cause of improved traffic safety in recent years. The positive influence of systems such as Electronic Power Steering (EPS) and Brake Assist (BA) is proven [32, 33, 34]. Over time, many such active safety systems summarized under the term Advanced Driver Assistance System (ADAS) contribute to increased traffic safety.
The last decade (2011-2020) is labeled the *"decade of action for road safety"* by the World Health Organization (WHO) [35]. More specifically, the European Union (EU) goal is to halve the number of death in traffic during this period [36]. However, stagnation in the last decade is observable. With 4009 fatalities in 2011 and 3275 in 2018, this goal is not achieved yet and hard to accomplish in Germany with the remaining two years, where the data is not available yet. Hence, a new advancement is necessary to achieve this goal retroactively. And the most promising candidate is AD.

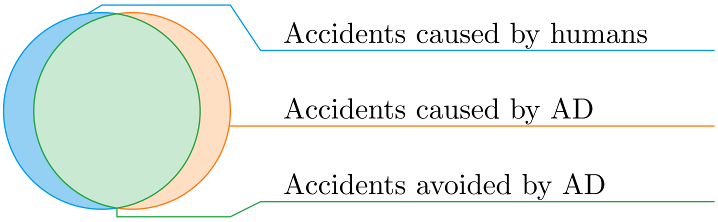


Figure 1.2.: Venn diagram of accidents that are caused by humans (blue) can be avoided by AD (green) and are caused by failures of AD (orange) adapted from [37].

To motivate the potential of AD to improve traffic safety, consideration of accountability of traffic accidents follows. For 90% to 99% of all accidents, a human is partially or fully responsible, with only a small share left for technical failures of the vehicles or its systems [38, 39, 40]. Hence, it appears logical to take the driving task away from humans through automation. However, this only benefits if the AD system does not introduce new accident potential exceeding human drivers. This wishful thinking is illustrated in Figure 1.2 [37]. The blue circle of the Venn diagram shows the hypothetical set of accidents that humans cause. A green area mostly overlaps it, indicating accidents that are potentially avoided by AD. The share of accidents that AD introduces is shown as an orange half-moon. A benefit of introducing AD concerning traffic safety is only present if the green area is greater than the orange.

This benefit is the case for any ADAS. However, it is well known that proving the difference for AD is infeasible with current methods [20]. Hence, it is necessary to define new ways of concluding to that proof. It is essential for many reasons besides guaranteeing the safety of human passengers in AD equipped vehicles. For the OEMs, it is crucial to minimize the economic risk of paying for caused accidents [41]. Moreover, to increase public acceptance of such systems, because 57% of German people do not believe in the reliability of AD [42]. These reasons motivate the central question of this thesis: *How can the safety benefit of AD be assessed?*

1.2. Aim and Objectives

The general aim of this thesis is to contribute the answer to the central question just stated. In particular, it shall elaborate a methodology that overcomes the limitations of the currently practiced ADAS assessment approach and is applicable to AD. While certainly, the goal can not be to provide the complete safety proof of a concrete AD function due to missing such a production-ready system and sufficient data, it is instead to provide the necessary tooling to achieve this task.

To achieve this goal, three objectives are formalized:

- **Make the safety of an AD function measurable:** Certainly, driving tests are part of an assessment and need to be evaluated for the function's safety performance. A novel measure for safe driving is required as a distinction between crashing and not crashing is not suitable.

- **Elaborate a methodology for the assessment of AD:** The task is to overcome the infeasibility of the current ADAS assessment's application to AD. Hence, a suitable assessment approach shall be elaborated and verified on a proof of concept base of all its components.

- **Provide tooling to show the completeness of the assessment:** An essential factor for an assessment method is its completeness concerning test coverage. This work shall provide tooling that can measure and improve the coverage of tests with respect to possible real-world driving scenarios.

1.3. Structure and Notation of the Thesis

The rest of this thesis is structured as follows. The problem stated in the introduction is refined in Chapter 2. Together with the objectives, it provides the research questions and hypotheses and designs a research methodology. Chapter 3 encapsulates all aspects of the state of the art that are necessary for the subsequent research. This includes assessment methods and general aspects of AD and related work for all components of the elaborated assessment methodology. The following three chapters constitute the doctoral studies' main research results grouped by the three objectives, research questions, and hypotheses. Chapter 4 provides a novel measure for the AD function's safety performance within a driving scenario. The assessment methodology elaboration with all its core modules, a proof of concept implementation, and verification follows in Chapter 5. Chapter 6 provides tooling for measuring and improving the completeness of the assessment methodology. Lastly, Chapter 7 revisits the fulfillment of the objectives, concludes this work, and finalizes with the suggestion for future work.

In the remainder of this work, uppercase bold $\mathbf{A}$ denotes matrices, lower case bold $\mathbf{a}$ vectors, and regular letters A or a scalars. Variables to functions are given with round brackets $a(\,\cdot\,)$ while elements of matrices and vectors index with square brackets $\mathbf{A}[\,\cdot\,]$ or $\mathbf{a}[\,\cdot\,]$, respectively. Deviations from this notation are mentioned explicitly in the text.

2. Problem, Hypothesis, and Methodology

The introduction already stated the general problem and why research in the field of AD assessment is necessary. This chapter refines the challenge to extract the core aspects of the required research. Thereby also the three research questions can be formalized, resulting in three hypotheses. An explanation follows with the research methodology stating how this thesis aims to answer the research question and confirm the hypotheses. Lastly, the contribution to knowledge achieved in this work is anticipated.

2.1. Description of the General Assessment Problem

An essential and critical step during the creation of any safety-relevant product is its release. This applies to an ADAS or AD function. The release itself is the transition from development to series production [43]. In that process, an OEM must ensure the product's safety, which is indispensable for two reasons. On the one side, human and property harm due to the product's use should be avoided at all costs. The OEM is responsible for any product failures and thereby caused accidents. As a consequence, it must fear substantial economic damage. On the other side, the product's safety must be made clear to the customer to achieve market acceptance. Especially for AD, Germany's public trust is with 57% of the interviewed people not believing in its reliability remarkably low [42].

The release process of an ADAS can briefly be explained as follows. In the beginning, its scope or use cases are defined. For example, Adaptive Cruise Control (ACC) needs to follow a preceding vehicle and keep a safe distance on highways, thus controlling acceleration. Then, a series of parameterized proving ground tests are conducted to ensure basic safety. Afterward, a more significant amount of real-world test cases on public roads, so-called endurance testing, stochastically hardens the safety argument [44]. Typically, these involve a magnitude of $10^6 - 10^7$km. If these stochastic guarantees safety with a significantly low margin of error, release can proceed.

This process is not feasible for AD functions due to several reasons. Starting with the definition of its scope and use cases already defines the problem. AD driving should be able to handle any kind of driving situation it can observe throughout its lifetime. Whether it is only on highways, rural, and urban environments, the number of use cases is enormous and hard to define. Additionally, it has to handle those use cases without a human driver's availability as a fallback. While this fallback is there for an ADAS as a human driver is still responsible for supervising the system, AD should be able to operate without any human onboard or distracted passengers. A cumulative result is the approval trap [37, 44]. Various work derives a dimension of real-world public road testing volumes from providing stochastic proof of safety. Examples are $5 \cdot 10^9$km for German highways [20] and $8.8 \cdot 10^9$km for the United States of America (USA) [45]. These numbers, whether they are precise or not, state that the effort required is infeasible. Hence, novel approaches are needed. However, even if one can conduct a number of tests that comes close to this volume, the assessment of the gathered data results in additional challenges. Consequently, also novel measures for the safety performance of the AD function are required [44].

Currently, these problems result in a wave of research trying to provide a remedy. New release processes and methods are proposed, with none being market-ready. An overview follows in the literature review from Chapter 3. Generally, two concepts recently draw attention. The first tries to leverage the number of required tests by conducting more targeted scenario-based testing instead of unplanned driving on real roads [46, 47, 48, 49]. However, therein also remain some unanswered questions. How can the catalog of scenarios be defined? How can one measure the tests' coverage and catalog concerning the possible driving scenarios in reality? The second concept tries to leverage the amount of necessary real-world testing through the inclusion of virtual testing, such as simulation [50, 51]. Undoubtedly, virtual testing always leaves the question of validity behind. Moreover, validity is crucial for safety-related testing. It is unclear how representative virtual tests must be to participate in the assessment.

This thesis aims to participate in answering those open questions. Indeed, a single thesis can not give a global answer to the assessment problem. Hence, the concrete research questions dealt with in this work are defined in the following.

2.2. Research Questions

The general scope of this thesis is to answer three open questions in the field of AD assessment. The first relates to the absence of suitable assessment metrics that enable the automated evaluation of a vast amount of driving kilometers. The second deals with the scenario-based assessment itself and the inclusion of verified virtual testing. The last research question concerns the problem of defining a suitable set of scenarios, the traversal of this set, and the estimation of its coverage. Consequently, this thesis formulates its research questions as follows:

1. How can the safety of automated vehicles be measured?
2. How can efficient virtual testing be verified and included in a scenario-based assessment process, and what is the impact of virtual testing?
3. How can the set of test cases be defined, traversed, and coverage be estimated?

2.3. Hypothesis

Based on the research questions, three hypotheses are formalized:

1. The performance and safety performance of an AD functions' behavior can be measured.
2. An assessment methodology can be formalized that locally verifies virtual test domains against real-world test drives through reprocessing of scenarios. The deviation of sensor information from ground truth in a measured scenario significantly impacts the reprocessing quality.
3. A scenario space and its coverage can be defined with local clusters and traversed through local variation with verified results.

These relate to the questions in their respective order. The three main content chapters (Chapter 4, Chapter 5, and Chapter 6) aim to answer the research question and confirm these hypotheses in this exact order.

2.4. Methodology

In order to answer the research questions and confirm the hypotheses, a designated strategy is chosen. Figure 2.1 illustrates this general research methodology. In this bottom-up approach, the three hypotheses erect as pillars on comprehensive literature research as a foundation. Then, an overall conclusion completes the structure as a roof.

The foundation or literature research in Chapter 3 splits into three major parts and includes all relevant prior knowledge for the subsequent three main contribution chapters. The first part summarizes general aspects of AD, such as scope, development progression, and functionality. The following compound explains how a single test is conducted in the automotive domain by listing test domains, elaborating the definition of test cases, and existing evaluation criteria.

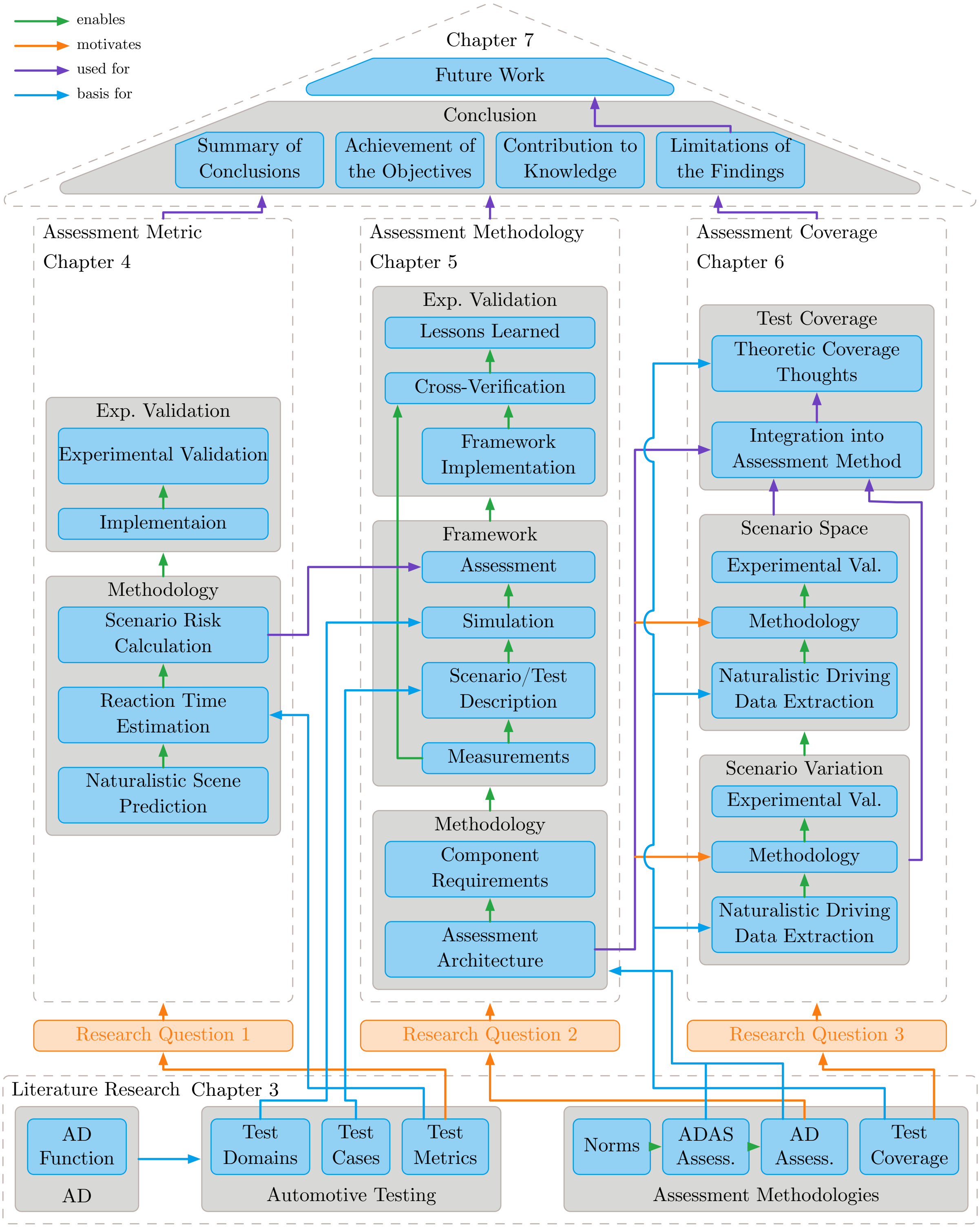


Figure 2.1.: A flowchart of the research methodology applied in this thesis. The literature research forms the foundations for the three pillars, answering one research question each. The conclusion and future work complete the structure as a roof.

Nevertheless, tests alone do not suffice for the release of new automotive functions. Instead, a whole assessment concept is necessary, which defines the last part of the literature research. Hereto belong norms that need to be applied, previous concepts for ADAS, and new approaches to AD that emerge current research. Lastly, also the crucial question about test coverage is considered.

From an analysis of existing test metrics as evaluation criteria emerges the need for new measures that are applicable to AD. Hence the first pillar builds upon the first research questions suggests such a criterion in Chapter 4. The first part describes the methodology of this metric and its mathematical background. Experimental validation of the metric follows in the second half.

The central pillar dedicates to answering the second research question on an efficient assessment concept for AD in Chapter 5. Therefore, a scenario-based concept, including virtual as well as real test domains, consolidates based on available approaches from the literature. Yet theory alone is well present in the literature, and actual proof of concept is missing. That is why such a proof of concept is provided for every part of the methodology. This results in a whole assessment framework, also utilizing the metric from the first pillar. This framework is, in turn, implemented and evaluated experimentally with a focus on cross-verification of virtual testing against real-world tests. An in-depth evaluation of the influence of virtualization and the used data basis for the tests in the framework is the core part of the experimental validation. Those form impediments that are identified and dealt with through solution approaches in the lessons learns section.

The last important part, the third pillar, and the dedicated Chapter 6 answer the research question about test coverage and ensures that the assessment concept is also representative. This question also arises from the assessment concept of the first chapter. The two building blocks for this are presented and evaluated separately. First, a proof of concept for a scenario or test case variation shows how coverage can be increased. Second, a concept for the definition of the test space is presented. These two consolidate in a fundamental understanding of test coverage for the assessment of AD. However, a validation remains reasoned theory as the vast amount of data necessary for an experimental validation is not available and can not be collected within the scope of this work.

The conclusion and future work in Chapter 7 completes the structure. Therefore, the individual conclusions of the three main contribution chapters summarize first. Revisiting this thesis's objectives and the promised contribution to knowledge following next is dedicated to checking the work results' completeness concerning the given task. Indeed, remaining impediments, caveats, and limitations of several aspects in this work are present and therefore summarized.

These and the collected solution approaches from this work form the tip of the roof, the future work.

2.5. Contribution to Knowledge

The research questions are motivated by the lack of knowledge in specific fields of the assessment of AD. Likewise, this thesis aims to fill gaps and contribute to knowledge. This contribution shall now be anticipated, aligned with publications, and later revisited in the conclusion from Chapter 7.

First, a gap of missing suitable assessment metrics for AD can be closed through a novel proposed metric resembling accident risk derived from traffic participants' naturalistic driving behavior. This metric is published in [Paper1].

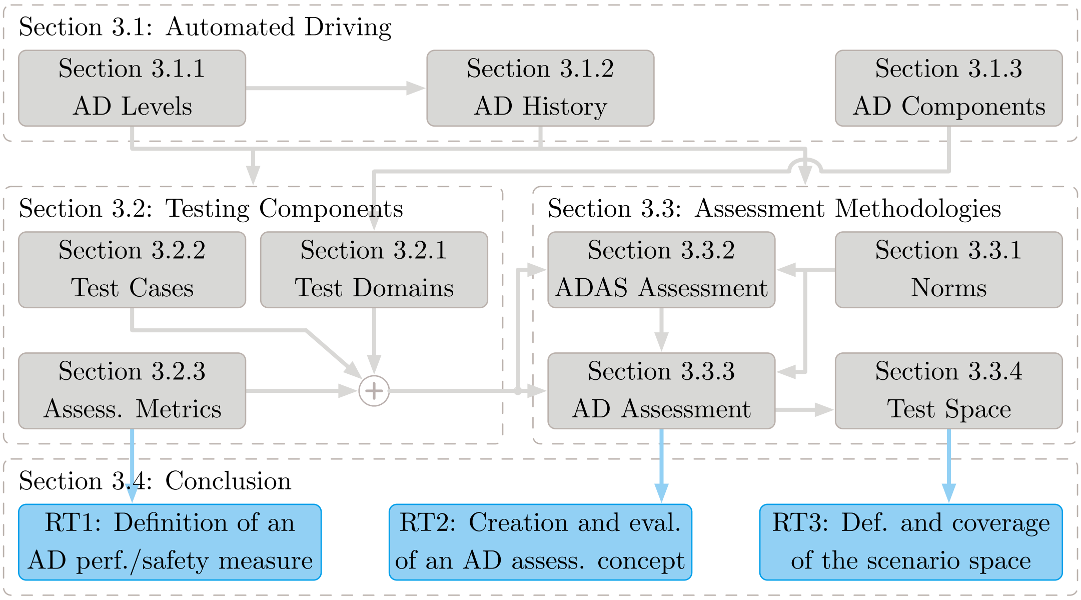
A significant contribution of the work is elaborating and realizing a complete assessment concept on a proof of concept level. This concept's theoretical foundation is not new [47], but is completed and realized in this work. A complete description, including all components, is summarized in [Journal3]. This work identifies and addresses the two most significant impediments of scenario-based assessment, which are the data basis for test case generation [Journal1] and the influence of virtualized tests [Paper2]. The whole concept and realization of its individual components also benefit the research project Projekt zur Etablierung von generell akzeptierten Gütekriterien, Werkzeugen und Methoden sowie Szenarien und Situationen zur Freigabe hochautomatisierter Fahrfunktionen (PEGASUS) [Poster1][Presentation1].

This thesis also extends the knowledge asked for in the last part and clarifies the test coverage methods. A novel local scenario variation algorithm providing naturalistic results promises an improvement of coverage [Journal3]. Further, a data-driven approach to the definition of a scenario space extends the knowledge and the possibilities to measure test coverage [Paper5].

Further improvements next to the core of the thesis have been made. These are in the field of map validation [Paper3], perception modeling [Paper4], and influence identification for driving controller performance [Journal2]. As these do not directly contribute to this thesis's goal, they are not further explained and not revisited in the conclusion.

With the problems stated, the task, and the research questions refined, the main central of this thesis follows next. As described in the methodology, the foundation through comprehensive literature research follows first.

3. State of the Art in Automated Driving and its Assessment



Section 3.1: Automated Driving
Section 3.1.1 AD Levels
Section 3.1.2 AD History
Section 3.1.3 AD Components
Section 3.2: Testing Components
Section 3.2.2 Test Cases
Section 3.2.1 Test Domains
Section 3.2.3 Assess. Metrics
Section 3.3: Assessment Methodologies
Section 3.3.2 ADAS Assessment
Section 3.3.1 Norms
Section 3.3.3 AD Assessment
Section 3.3.4 Test Space
Section 3.4: Conclusion
RT1: Definition of an AD perf./safety measure
RT2: Creation and eval. of an AD assess. concept
RT3: Def. and coverage of the scenario space

Figure 3.1.: The structure of Chapter 3. The sections are shown in grey boxes. Connections between these sections show informational dependencies. The three research tasks (RT) are derived from the state of the art and summarized in the chapter's conclusion.

The research in the field of AD dates back to almost a century [6] and has drawn industrial focus with more intense research since a decade. Thereby, the amount of completed work forming the state of the art is significant. This is true for the research in the driving function of AD itself and its assessment, which is the main focus of this work. Then again, the studies and development in the field of AD are by far not completed, and a lot of open questions remain. This chapter introduces the work relevant to automated driving assessment and highlights knowledge gaps that justify the research questions stated in Section 2.2. The structure of this chapter is shown in Figure 3.1 and described in the following.

The state of the art for AD itself is introduced in a fundamental scope in Section 3.1 to form the basis of the thesis's intrinsic content. In the beginning, the levels of driving automation are defined in Section 3.1.1 and serve as the milestones for the history of driving automation shown in

Section 3.1.2. An exemplary AD driving function with its functional principle and components outlines in Section 3.1.3. Section 3.2 introduces the necessary tools to perform a test in the automotive domains. Different test domains are explained in Section 3.2.1, and their capabilities to assess the previously introduced modules of the AD function are analyzed. The remaining two components necessary for driving tests are the definition of test cases and metrics for performance and safety. They are elaborated in Section 3.2.2 and Section 3.2.3, respectively. In combination, those three modules serve as a foundation for superordinate assessment methodologies that are discussed in Section 3.3. This section starts with the introduction of the applicable norms to such methodologies in Section 3.3.1. A direct result from norms is the established method to assess ADAS elaborated in Section 3.3.2. This subsection also addresses the problems arising with its application to AD and the necessity to improve or work on new methods. An extensive literature review on approaches to the assessment of ad AD follows in Section 3.3.3 as a consequence. Next, the question on the completeness of the assessment approach is raised and addressed Section 3.3.4. Finally, the literature review reinforces the three research tasks in assessment metrics, methodologies, and test space coverage as a conclusion in Section 3.4.

3.1. Automated Driving

Before discussing the assessment of AD, it is crucial to understand the basics and principles of driving automation itself. For that reason, this section first introduces the definition of automated driving in the context of road vehicles. Afterward, a short recap of the history of AD concludes the current stage of development. Finally, a general structure of such a state of the art system and its components is explained.

3.1.1. Levels of Driving Automation

The term *automation* itself indicates a system that takes over human responsibilities to control a process. Concerning road vehicles, the system would take responsibility for parts or the whole driving task. This thesis intentionally avoids the commonly used term *autonomous*. In contrast to *automation*, it implies a degree of intelligence within the system that is able to make decisions in the absence of any human. Technically, this autonomy would include the car's ability to decide on a control strategy on the road and the path towards the goal itself and the destination [52]. It remains arguable if this can be considered the highest form of automation or as too much automation. A vehicle deciding where to go by itself might not be in the interest of customers. However, as the development of full driving automation is still in progress, it is more desirable to

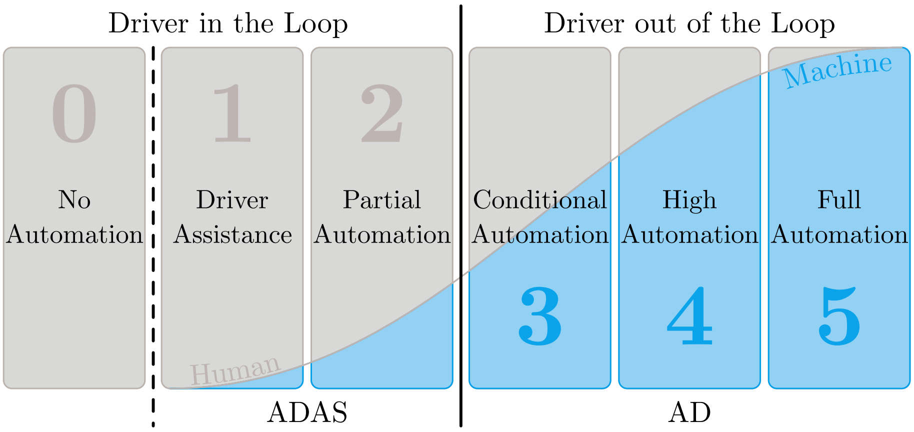


Figure 3.2.: The six levels of driving automation defined by SAE. The blue fill within the respective pillar implicates the share of the driving task transferred to the driving function. A categorization of those levels into ADAS and AD is given.

define automation levels instead of only full autonomy. Well-established and generally accepted levels emerge the Society of Automotive Engineers (SAE).

The range from no automation at all to the full extent is divided into six levels [53] and depicted in Figure 3.2. The driving task is split between a human driver's indicated as a grey surface and the machine's responsibility as a blue surface through these levels. Additionally, the levels distinguish with the environments the function can operate in, the so-called Operational Design Domain (ODD). These levels are given as follows:

- **SAE level 0 – No Automation:** This characterizes vehicles without any automation at all. The human driver is responsible for all aspects of the driving task.
- **SAE level 1 – Driver Assistance:** As soon as modern driving assistance systems like ACC or Lane Keeping Assistance System (LKAS) can take over longitudinal or lateral control, respectively, the automation reaches level 1. A significant limitation is that either of these systems is allowed to be active simultaneously. The human driver is still responsible for the system and environment supervision and acts as a fallback. Also, ODD for such systems is limited.
- **SAE level 2 – Partial Automation:** Vehicles able to take over both lateral and longitudinal control simultaneously qualify for level 2. However, the human driver is still responsible for monitoring the driving task and intervening at any time. The ODD is still limited to, e.g., highways. Vehicles that offer these capabilities for a limited time range are already available on the market.

- **SAE level 3 – Conditional Automation:** With level 3, the human driver is not responsible for supervising the environment and system anymore, while the machine is responsible for lateral and longitudinal control. However, he must still be present as a fallback by request, and the ODD is still limited.
- **SAE level 4 – High Automation:** Level 4 extends the previous layer's capabilities by removing the need for a human driver as a fallback. The limitation on the ODD still applies.
- **SAE level 5 – Full Automation:** Lastly, the ODD becomes unlimited in level 5. A fully automated vehicle is capable of driving in any environment without the need for a passenger at all.

Current market-released vehicles are capable of up to level 2 automation, while higher levels are subject to ongoing research and development. The history, current progress, and an outlook into the future of AD are elaborated in more detail.

3.1.2. History and Progress in Automated Driving

The idea and dream of driving automation have come a long way since documented in history nearly 100 years ago [6]. Ever since, the awaited technology seems to be 20 years away [54]. This section aims to give an insight into the timely progression of AD from the past into the future. It does not claim completeness but instead outlines some important milestones. The information mainly bases [6] and [55].

Figure 3.3 provides a timeline of said progression featuring the significant milestones. A first driverless vehicle raised people's astonishment by driving around the streets of Dayton, Ohio, in 1921 [56][1]. Despite being remote control by a human and hence not automated, it started building attention for the matter. The public interest rose even more through the introduction of the remote-controlled "American Wonder" from Houdina Radio Control as a first full-sized driverless car in 1925 [58][2]. The Great Depression from 1929 to 1941 caused the vision of AD to take a backseat. One step further from the remote control to automatically guided vehicles was made by General Motors Company LLC (GM) through the development of a driver-less automatic guided vehicle in 1958 [60][3]. Sensors attached to the car's lower front detected cables buried into the streets and followed them. The first steps towards automation extended through

[1] First mentioned in the Washington Herald [56], image obtained from the Daily Ardmoreite [57].

[2] First mentioned in the TIME magazine [58], image obtained from the Discover magazine [59].

[3] Image obtained from the magazine article [60].

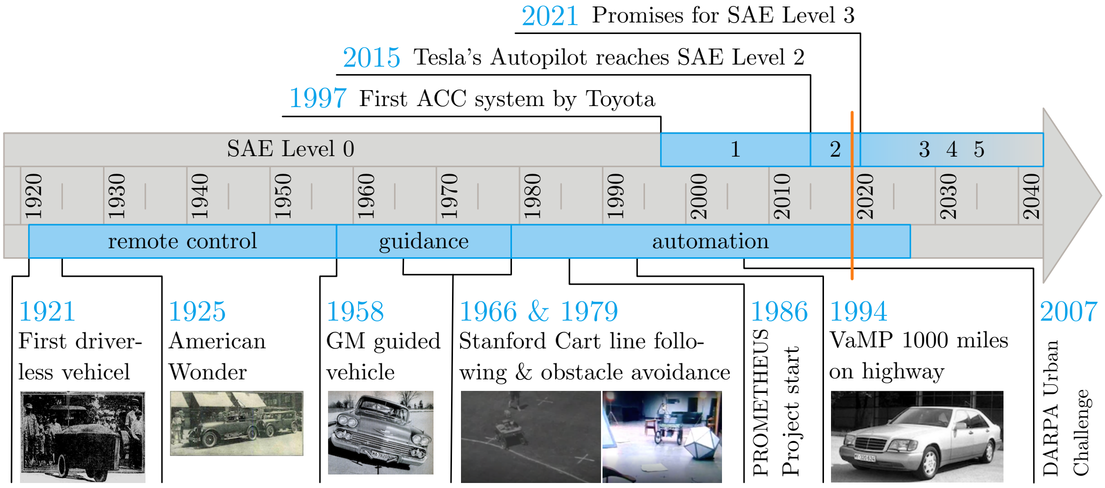

2021 Promises for SAE Level 3
2015 Tesla's Autopilot reaches SAE Level 2
1997 First ACC system by Toyota
SAE Level 0
1
2
3 4 5
1920
1930
1940
1950
1960
1970
1980
1990
2000
2010
2020
2030
2040
remote control
guidance
automation
1921 First driver-less vehicel
1925 American Wonder
1958 GM guided vehicle
1966 & 1979 Stanford Cart line follo-wing & obstacle avoidance
1986 PROMETHEUS Project start
1994 VaMP 1000 miles on highway
2007 DARPA Urban Challenge

Figure 3.3.: A timeline of AD history featuring the research milestones on the lower and industrial milestones on the upper side. Within the timeline, the phases of research and SAE levels are shown as blue bars, respectively. The orange line indicates the time of writing.

the "Stanford Cart" [61] research project from 1960. This small vehicle managed to follow a printed white line in 1966 [62][1] through a vision-based algorithm running remotely on a mainframe computer. The latter two approaches technically automate the driving task; however, they are labeled as guidance as it is not what we expect as AD today. Rudimentary signs of our today's expectancy of automation can be found in extended work on the "Stanford Cart" from the same researchers in 1979 [63][2]. Now the vehicle was able to find a path on its own through obstacles to reach a defined goal. A rail-mounted horizontally moving camera enabled stereo vision and advanced vision algorithms to achieve path planning. However, the 20 meter long obstacle course took the cart five hours to complete. Major milestones in the automation of full-sized vehicles on real roads emerged the PROgraMme for a European Traffic of Highest Efficiency and Unprecedented Safety (PROMETHEUS) starting in 1986 [64]. The research vehicle of the Universität der Bundeswehr München (UniBW) named Versuchsfahrzeug für autonome Mobilität PKW (VaMP) [65][3] completed 1000 kilometers during the final presentation in 1994 on Autoroute 1 near Charles-de-Gaulle airport in Paris. During this highway drive, the vehicle performed various tasks such as lane changing and overtaking automatically. VaMP is cited as one of the first truly autonomous cars [67]. As a milestone for urban driving automation serves the Defense Advanced Research Projects Agency (DARPA) urban challenge 2007 [68], where

[1] Image extracted from the video [62].
[2] Image extracted from the video [63].
[3] Image obtained from the research report [66].

competitors from various institutions competed with their research vehicles on parkours in an urban-like environment. The Carnegie Mellon University (CMU) team "Tartan Racing" [69] won the challenge [70].

What begun early on the side of research takes more effort and time for the industry or, more specifically, the OEMs. Market release for such technology usually requires a long route of development, testing, and verification. Speaking of automation in the automotive sector, the first vehicle that can be considered as SAE level 1 was developed by Toyota and equipped with the first marked-ready laser-based ACC in 1997 [71]. Mercedes answered with a nowadays more common Radio Detection and Ranging (Radar) based system in 1999 [72]. The first LKAS was presented by Nissan in 2001 [73]. While the use of either of these systems for longitudinal or lateral guidance was established two decades ago, a combination qualifying for SAE level 2 reached consumer vehicles from 2015 on. Since then, Tesla's autopilot is able to drive automatically on highways taking over lateral and longitudinal control [74]. In the following year, other OEMs such as Audi [75], BMW [76], and Daimler [77] released level 2 production cars. Up to the day of writing, this is the progress of the industry. Until regulatory circumstances are not fully solved and the human driver remains a fully responsible system supervisor, the release of level 3 and beyond manifests in announcements and promises. Exemplarily, Audi's 2019 model of the A8 is said to be capable of level 3, but the system remains undelivered [75]. Also, BMW announced the next step for 2021 [76]. The levels 4 and 5 are still a long way off and hence not further discussed at this point.

The groundwork for the ambitious goal of fully AD is advancing. Its TRL [10] surpassed level 4 [11], meaning it is in the state of industrial development and in need of standards and regulations. Hence, as one of the many subtopics of AD that are unclear, also its final assessment for market release remains an open question for current research. Some other topics of research interest are outlined in the next section while discussing the fundamental functional principle of an AD system.

3.1.3. Functional Principle of Automated Driving

While an AD's functional scope depends on its level defined in Section 3.1.1, the functional principle and its components are similar. It can be roughly divided into the five components [78] depicted in the upper row in Figure 3.4[1].

For the function to interact with its environment, it must first perceive it. For that task, a wide range of sensors is used to gain knowledge about static and Traffic Objects (TOs). In contrast

[1] Satellite icon made by mavadee from www.flaticon.com.

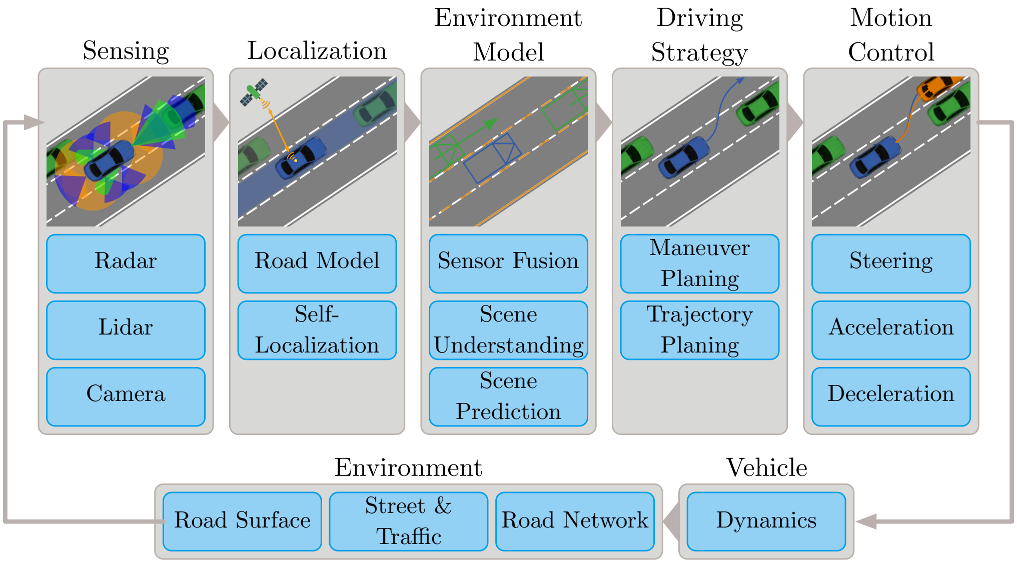


Figure 3.4.: A block chart of the five components of an AD system in closed-loop with the physical worlds. The top row enlists sensing, localization, environment model, driving strategy, and motion control with their subcomponents in their order of execution in the function. The feedback loop in the lower part defines the movement of the vehicle and the evolution of the environment.

to market-released ADAS, the number of sensors is increased [79] to provide an area-covering perception and redundancy. Together with established sensor systems such as Radar, ultrasonic, and camera, new technology such as Light Detection and Ranging (Lidar) are used in ongoing research to increase the accuracy of sensing. Additionally, investigations of data aggregation through Vehicle-to-X Communication (V2X) or Vehicle-to-Vehicle Communication (V2V) for improvement of gathered information or redundancy are ongoing [80]. Besides information about the environment, the function also needs to extend its self-understanding and its place within the environment. For that, the task of localization creates a road model and estimates the Vehicle under Test (EGO) vehicle position within this model. Literature lists different approaches to this. Road models can either be created from previously sensed information [81][Paper3], derived from prior-knowledge in the form of High-Definition (HD) maps [82] of from a combination of both [83]. The EGO vehicle's localization follows afterward through sensed information and Global Positioning System (GPS) measurements. All of this data further needs to be processed into a single environment model. Again sensor information is used to detect and parameterize dynamic TOs on the road model. Current research focuses on different approaches to this problem. One can either fuse the information of various sensors on a low level and perform object detection and classification afterward [84], or the other way around to use detected objects of different sensors providing redundancy [85]. This environment model is then amplified with an understanding and prediction of the scene's future. Especially the accurate prediction of other TOs such as human-controlled vehicles [86] or pedestrians [87] still attracts the focus of current research. Based on the predicted environment, the EGO vehicles decides on its future driving strategy in the next block. This task can be divided into a high-level maneuver planning and low-level trajectory planning [78] although other approaches are possible. Lastly, the planned trajectory needs to be executed by the motion controller through the usage of control inputs such as steering, acceleration, and braking.

For a complete description of AD as a system, these five components embed in a feedback loop in the lower half of Figure 3.4. These are two layers extracted from the three-layer-model of Donges [88] the AD function substitutes the driver. The vehicle takes the controller inputs of the function and moves as a result within the environment. The latter consists of the road network, local street, traffic, and surface of the road, according to Donges. Of course, this list can be extended by further influences such as weather conditions. The feedback of the environment serves as input for the AD function to close the loop. Certainly, the interconnections and influences between these blocks are more complex in reality. For the scope of this work, this level of complexity suffices.

The correct assessment of such a system is the main focus of this thesis. Hence, a more detailed description of the five modules is substituted by a black box consideration in favor of system level testing. Existing tools and methods for testing such black boxes are elaborated in the following.

3.2. Components of a Driving Test

The testing of driving functions is crucial during development and the final assessment for market release. The primary purpose is to ensure the quality and safety of a system that meets expected and regulated requirements. Depending on the functional scope and safety relevance, volume and type of required testing vary. This section aims at giving an introduction to testing in the automotive domain. Available test domains are introduced, followed by the definition of test cases. Finally, assessment metrics are discussed as they are an essential component of testing.

3.2.1. Automotive Test Domains

The first questions arising with the planning of a test involve where and how to conduct the test. Several domains of execution are available for testing, both in general and unique to the automotive domain. Their abstraction levels range from real-world tests with prerelease functions on real roads to purely simulated tests within a computer. In the following, the available domains are explained with their gradation of abstraction.

Test Drive

A generic test on either real roads or enclosed environments, such as test tracks with a prototype vehicle, is named Test Drive (TD). Usually, within this kind of test, the procedure and the tested cases are pre-planned. It aims at testing a specific system against defined scenarios. Hence, results are valid as no abstraction or virtualization is used at all. However, the reproduction of a pre-defined test case, in reality, is only possible to a certain degree as reality is neither deterministic nor fully schedulable. Additionally, real-road tests are comparatively expensive, time-consuming, and put the test driver into a certain amount of risk. Furthermore, the test accuracy is strongly dependent of the measurement system's accuracy.

Field Operational Test

Similar to TDs, the Field Operational Test (FOT) is also performed without any abstraction or virtualization on real roads. However, the conducted test cases are unspecified and aimed at testing a specific system under real conditions [89]. While driving a larger amount of test kilometers on public roads with real and uninitiated traffic, the tester gains predictions on the system's performance after market-release in its intended ODD. FOTs also help to establish a stochastic proof of safety through defect rates over driven distances. This type of testing is also often used for studying the impact of aftermarket devices [90] or ADAS, such as ACC, Supplementary Restraint System (SRS), or Forward Collision Warning (FCW) on traffic and driver's behavior [91, 92]. Still, those tests are costly, time-consuming, and at a certain level of risk for test drivers.

Naturalistic Driving Study

Similar to the last domain Naturalistic Driving Studies (NDS_1s) are conducted at a large scale on public roads. In contrast to FOT, the focus is here not on a specific system but rather on the data obtained from measuring traffic and the environment [93]. Hence, this domain is not suitable assessing AD but listed for completeness and its importance as data set generator. Such data find application through various research fields of AD, such as traffic prediction and driving scenario generation. Popular member projects of this domain are Cityscapes [94], UDRIVE [95], KITTI [96], and highD [97]. As FOT can also suffice for data generation, and the intention being the only difference, FOT and NDS_1 are closely related. Hence, they share the same drawbacks of being costly, time-consuming, and risky for test drivers.

Simulation and Software-in-the-Loop

In contrast to real-world testing is the simulation. Hereby, every piece of the system is virtualized and simulated in the loop. Hence, this test domain finds its most significant application in the early development stages [98], where hardware is not present yet. Closely related is Software-in-the-Loop (SIL). The only distinction is that the function is translated into the language and code intended to run on the real hardware. Still, it is executed on development computers. Simulation is the cheapest and fastest testing domain and bears no risk for drivers. It is fully controllable in terms of test cases due to the determinism that can be one of the design requirements for the simulation itself. However, due to its level of abstraction, it generally lacks validation, and the actual test outcome on real roads can only be estimated [99]. Also, depending on the required level of accuracy, complex models the system's modules need to be developed separately.

Hardware-in-the-Loop

Between purely virtual and real domains resides Hardware-in-the-Loop (HIL) [100]. Similar to SIL, the function under test is translated into target code, but now runs on the target hardware. Hence the domain is targeted on testing the interaction of a component's software with its intended hardware target. Still, all other modules are simulated and adapt to the benefits and drawbacks of the simulation domain. A complete validation of test results is still missing. However, the coupling of multiple HILs is possible and can reduce the degree of abstraction. Certainly, constraints due to such machines' size and flexibility prohibit the linkage of all AD system modules through HILs. Its usage is thereby also limited to the development-supporting test in the automotive context [98]. Although the higher cost of HILs hardware compared to solely computers, it is still cheaper than the real road test in prototype vehicles. Other benefits and drawbacks align with SIL.

Driver-in-the-Loop

A domain unique in the automotive domain is the Driver-in-the-Loop (DIL) [101]. Depending on the configuration level, a single seat with a steering wheel and paddles or a whole car suffices in a virtual world displayed on screens. To compensate for motion sickness [102], moving platforms are often used. As stated in its name, the test environment requires a human driver. While this might not be suitable for the assessment of AD in the first place, this technology can still be utilized to address the takeover process [103] or the effect on distracted passengers [104]. While a DIL acquisition cost is usually high, test execution is less costly than a real road test. The test cases are plannable and put the test driver at no risk. Still, the simulated world prohibits

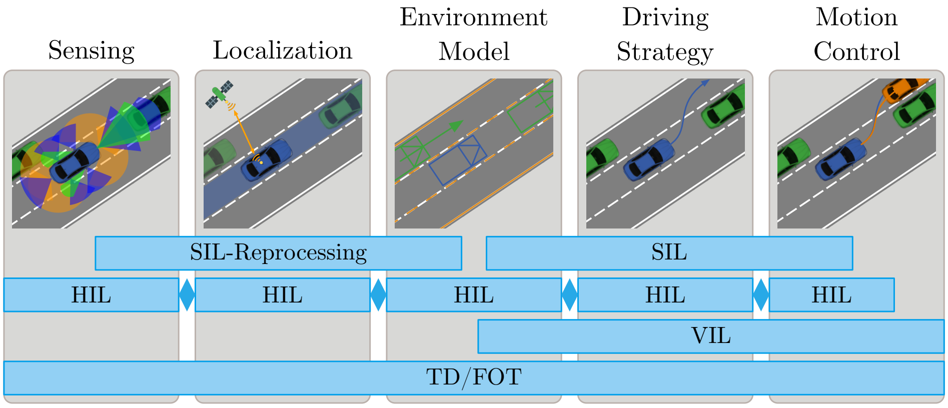


Figure 3.5.: The capabilities of the test domains in terms of assessing the five components of the AD function from Figure 3.4. The coverage of the blue bars indicates the applicability of a domain to test a particular module.

validation of the test results. For AD, this is a crucial component to test the passengers' experience at no risk.

Vehicle-in-the-Loop

Another domain exclusively found with an automotive application is Vehicle-in-the-Loop (VIL). It is much closer to reality as a real vehicle is used in the real world, but the environment information is manipulated [105]. Thereby, one can trick the vehicle under test into seeing TOs and other environmental aspects that are not really there. This decreases the driver's risk compared to real-road testing and makes the test plannable to a certain degree. It can be used in later development stages [98] to test the function's interaction with realistic vehicle behavior. Due to the simulated environment, full validation is still not possible. VIL tests are more costly than regular real-world tests due to the need for additional hardware and software for sensor manipulation.

The Capability of the Test Domains

Depending on their level of abstraction and virtualization, the introduced test domains are suitable for assessing different modules of the introduced AD function. An overview supporting this paragraph is given in Figure 3.5[1] [Journal3]. The two domains NDS_1 and DIL are neglected, as they are not intended to test a specific function or system without a driver, respectively. The range of the blue bars indicates the assessment abilities of the test domains. The only domain able to assess all five modules of the AD function ad TD and FOT. As nothing is virtual and no

[1] Satellite icon made by mavadee from www.flaticon.com.

Test Domain	Expense	Risk	Validity	Plannable	Testing capabilities				
					Sen	Loc	Env	Pla	Con
TD/FOT	−−	−−−	+++		✓	✓	✓	✓	✓
SIL/Simulation	+++	+++	−−−	✓			(✓)	✓	(✓)
HIL	+	+++	−−	✓	[✓]	[✓]	[✓]	[✓]	[✓]
VIL	−−−	−	++	(✓)			(✓)	✓	✓

+ better − worse ✓true (✓) partially true [✓] individually true

Table 3.1.: Summary of benefits and drawbacks of the test domains introduced in this section. The table also indicated where tests are plannable and which components of the driving function they can assess.

abstraction of either hardware or software applies, the results are valid by default. Within a HIL, the interfaces and components around the modules are simulated. As long as the module itself is not abstracted, its inner functionality is assessable. Combinations of HILs are possible, and connections indicate such compounds with blue diamonds. Hence, the HIL testing capability concerning the components is always individually given, but never as a whole. Still, chaining too many modules is infeasible in reality. Also, a HIL typically resides in a lab environment, limiting its capability for motion control due to the simulated environment. A VIL is a real vehicle in the real world with manipulated information about its environment. Hence, sensors and localization are not assessable, and the environment model only partly, as predictions are still in the responsibility of the function. When replacing the real vehicle with a simulation, the accessibility of the motion controller is limited. Therefore, the bar of SIL shortens. Apart from the assessment of the functional behavior, SIL-reprocessing can be used to test partly sensing, localization, and the environment model by feeding raw recorded sensor data into the chain. This raw data reconstructs a pre-recorded environment. Hence, this method is named reprocessing and primarily useful for debugging faulty behavior. Therefore, it is excluded in the following.

All results of this section finally summarize in Table 3.1 concerning benefits, drawbacks, and capabilities. The table weights the benefits and drawbacks of the test domains through a plus and minus scheme. Further, capabilities and plannability are given through checkmarks, with round brackets indicating the aforementioned limitations and square brackets the individual capabilities of a HIL. Pure simulation is the most economical and risk-free, while real driving tests preserve the results' general validity. HIL and VIL increase validity at the cost of increased expenses. Completely plannable test domains are simulation and HIL, while VIL has limitations due to uncertainties in the vehicle's physical feedback. Also, the previously described capabilities are transferred to the table. Partial assessment abilities indicated with (✓) and [✓] stand for the interfacing boundaries of a HIL.

A testing engineer can choose from a variety of methods for the assessment. Due to the drawbacks, benefits, and applicability limits, the choice is dependent on the level of development and the associated demand for accuracy and validity. The research in the field of assessment of AD focusing on optimizing for benefits while preserving validity. Besides the test domain itself, a possible working point is the combination of domains in the assessment method itself. This work as well focuses on this matter. The following section discusses the definition of a test case that executes in the presented domains.

3.2.2. Scenarios as Test Cases

To execute a planned test, first, the specification of an associated test case is necessary. In the automotive context, such a test case commonly refers to a driving scenario. This section aims to elaborate on the definition of relevant terms, a scenario's contents, and available descriptions standards.

3.2.2.1. Definition of the Terms Scenario, Scene, and Situation

The term *scenario* already appears in this work without a precise definition. While it might seem reasonable to specify this as a traffic occurrence period, it should be clarified further concerning its contents. Widely accepted definitions of the relevant terms *scene*, *situation*, and scenario in the automotive context are given by Ulbrich [106] and elaborated in the following.

Scene

> *"A scene describes a snapshot of the environment including the scenery and dynamic elements, as well as all actors' and observers' self-representations, and the relationships among those entities."* [106, p. 2]

A scene contains information about the environment, dynamic TOs, and EGO at a certain point in time. Ulbrich further clarifies that this can be either objective and from a global point of view or subjective from a single vehicle's point of view. This not only concerns the amount of information stored in a scene, but usually also its accuracy. While a global point of view can generally only be obtained in simulation and hence is all-encompassing and accurate, a particular vehicle derives this information from error-prone sensors with a limited field of view. The exact subelements of the environment and dynamic elements are discussed later in Section 3.2.2.2. Still, a single point in time is insufficient to make up a test case.

Situation

> *"A situation is the entirety of circumstances, which are to be considered for the selection of an appropriate behavior pattern at a particular point of time. It entails all relevant conditions, options and determinants for behavior. A situation is derived from the scene by an information selection and augmentation process based on transient (e.g. mission-specific) as well as permanent goals and values. Hence, a situation is always subjective by representing an element's point of view."* [106, p. 4]

Hence, a situation is considered partly as a subset of a subjective scene. However, the augmentation process does add information that is not available in a scene such as TO predictions. Containing solely necessary information decision making, it serves as input for the driving strategy module from Figure 3.4. For the definition of a test case, the situation inconsiderable as the creation of this information is part of the functions' responsibilities that should be tested.

Scenario

> *"A scenario describes the temporal development between several scenes in a sequence of scenes. Every scenario starts with an initial scene. Actions & events as well as goals & values may be specified to characterize this temporal development in a scenario. Other than a scene, a scenario spans a certain amount of time."* [106, p. 5]

The concatenation of several scenes forms a scenario. This adds the temporal aspect to the situation. Ulbrich states that this evolution of scenes describes alternatively through actions and events. Through the derivation of this definition from the scene, a scenario can be subjective and objective. Further, the exact contents in terms of subelements are the same and discussed in Section 3.2.2.2.

In consideration of these definitions, it seems convenient to use scenarios as test cases. The temporal aspect allows for exposing the vehicle or function under test to the evolutionary surrounding and observe its reaction. We further conclude that for testing the scenario ideally contains cause and effect of some traffic incident. A scenario is then in the temporal range of seconds [107]. Thus, the function's reaction to this incident can be assessed and compared to other functions, previous development iterations, or human drivers.

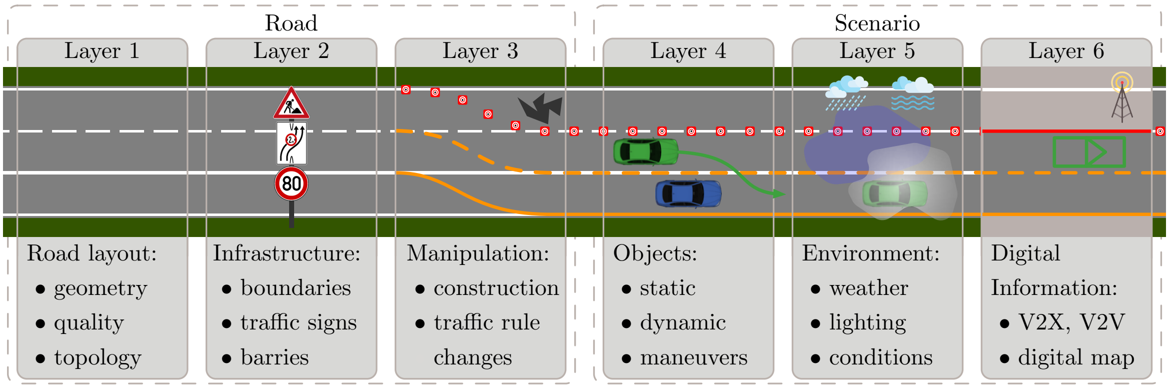


Figure 3.6.: The six layers of a traffic scenario [49]. Layout, infrastructure elements, and temporary manipulations define the road. The dynamic part of the scenario consists of TOs, environmental conditions, and information provided to the vehicle through V2V or V2X communication.

3.2.2.2. Contents of a Scenario

After defining the term scenario, further considerations about its contents are necessary. Despite the temporal component, the content of this section applies to a scene as well. Its primary purpose is to describe a test case that can execute in any of the in Section 3.2.1 mentioned test domains. Hence, it must be all-encompassing, not leaving out any necessary information for the proper test conduction.

The essential components for a complete description can be grouped into the six layers [49] shown in Figure 3.6[1]. The first layer defines the underlying road. It contains the overall topology of the relevant map section as well as the geometries of lanes and markings, the surface quality, and physical parameters. On top of the road, infrastructure elements form layer two. Among these are structural boundaries of the road, traffic signs, and static barriers, for example. Temporal manipulation of the first two layers can be specified in the third layer. For example, construction sites frequently result in closure, construction, or relocation of driving lanes as well as changes in the local traffic rules such as a limited allowed velocity and restriction of overtaking. The dynamic and static objects that are not part of the road and infrastructure are added in layer four. To that also account the EGO vehicle and TOs, including the movement and dynamic properties over time. Layer five adds weather, lighting, and other environmental conditions to the scenario. Finally, the digital information provided to the function under test forms the sixth layer. Such data can come in the form of digital maps and environment measurements sent

[1] Weather and antenna icons made by Freepik from www.flaticon.com.

over V2X or V2V. As the quality and availability of such information are not dependent on the function itself, it is part of the scenario description.

There is generally a larger quantity of dynamic scenario part configuration possible on a single road topology. Thus, the road and the road and the dynamic part of the scenario are naturally described independently. This makes the road elements reusable and avoids the storage of redundant data. Referred to the six-layer-model, this means a separation of the layers into the road and actual scenario. Layer one to three belong to the road and four to six to the actual scenario. For simplicity, the latter three layers are referred to as scenario from now on. Tangible implementations of roads and scenarios are presented in the next section.

3.2.2.3. Available Road and Scenario Description Standards

Several standards are already available as scenarios for testing and benchmarking are essential for the development and assessment in the automotive context. This section aims to give a short insight into the availability of road and scenario description format down to the level of computer-readable files.

Road Descriptions

An exemplary open standard for the detailed description of road networks and topologies is *openDRIVE* [108]. The since 2006 developed file format is based on Extensible Markup Language (XML) and is intended to store a local area map. Its structure defines the elements of layers one to three extensively. Roads form with sections and lanes along with different geometric reference line types, and infrastructure can be placed on top afterward. Elevation profiles and an optional third dimension can be added to the description. The *openDRIVE* facilitates a road-based modeling approach.

Another approach frequently found in recent research are descriptions of roads with *lanelets* [109]. These atomic elements define a lane segment through left and right bounds as polygons. Interconnected lanelets represented as a graph form a road. Infrastructure elements are limited to regulatory elements such as traffic signs linked to the lanelets. Likewise, this description stores in an XML file.

An industrial description is available with Navigation Data Standard (NDS_2) [110]. This format is developed by a non-profit consortium with OEMs and used in market-ready products such as navigation systems since 2012. Since then, it continuously improves to match the precision and detail requirement of AD. Similar to lanelets, this standard bases on lanes as atomic elements. The data is stored in database formats to increase the scalability and availability of the information.

Scenario Descriptions

One can define the flow of a scenario in two ways. The first is a linear description of the procedure similar to a recording. Thereby it is strictly defined what happens when and with whom in a time series. It is especially useful for a rigorously planned test or for replaying a recorded scenario in virtual test domains. Alternatively, an event-based description is possible, where the flow is not defined with time but with a series of actions and triggers. This may mitigate minor non-deterministic behavior in the scenario but can also result in deadlocks. It enables the creation of scenarios with a logical flow.

An open standard for the description of scenarios on top of roads is *openSCENARIO* [111]. Originating from the same developers as *openDRIVE*, also this format is stored as XML. Objects can be defined with their properties before assigning deterministic routings or paths to them. Predefined driving paths for planned tests can be specified as time series of position points. The format also offers sub-nodes for the specification of environmental conditions or the link to a road, for example. *openSCENARIO* is supported by the open-source simulator CARLA for instance [112].

Other than this standard, only specific solutions related to a product are available. For example, the Simulation of Urban Mobility (SUMO) uses its own event based description in XML [113]. Groh suggests the usage of velocity and heading splines for the description of time series in predefined paths in [47] instead of a sequence of coordinates. This reduces the amount of data and redundancies but does not constitute a standard yet.

Summarizing, there are different description formats for both roads and scenarios available. However, none of them defines a de facto standard that is globally accepted. The available descriptions also face other issues that are discussed later in this work when further discussing the requirements on these for the assessment of AD.

3.2.3. Existing Assessment Metrics

After the conduction of a test case follows the assessment of the results. An important part of the assessment of AD is the definition of metrics. On the one hand, it is not easy to decide on a test's outcome by visual inspection of the raw data. On the other hand, the vast amount of

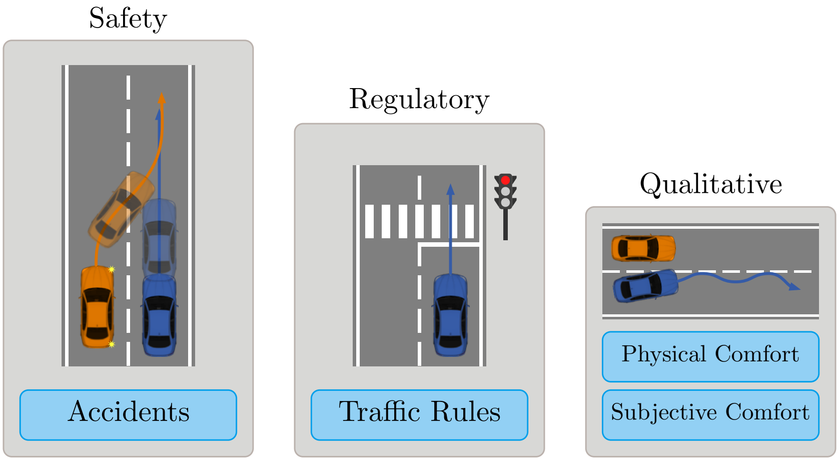


Figure 3.7.: The three categories of assessment metrics, safety, compliance, and comfort. The heights of the respective pillars implicate their order of importance for the assessment.

necessary tests makes manual inspection infeasible. Hence, metrics for the performance of the AD function within such a test are required.

Literature provides numerous quantities for such an assessment. They can roughly be categorized into three groups, as shown in Figure 3.7. Most important is traffic safety. An accident causes human and property damage that must be avoided at all costs. Also, compliance with regulations and laws apply to AD in the same way as to human drivers. However, it might be acceptable to violate a traffic rule to avoid an accident. Exemplarily, exceeding the speed limit to escape a critical situation that would otherwise lead to an accident is tolerable. Lastly, the AD function's quality manifests in the passengers' comfort within the vehicles and plays a vital role in the assessment. The safest AD vehicle that complies with all rules and regulations will not reach market acceptance if the occupants do not feel comfortable. But again, situations may occur, where the comfort zone may be trespassed to avoid an accident or to comply with traffic rules. The quality itself divides into two subgroups. Physical comfort concerns about the driver not getting sick or other physical sufferings. Subjective comfort encompasses the driver's perception of safety. Arguably, there is a fourth category of metrics encompassing internal states of the function and degradation on a system level. However, as these are primarily in the developers' interest for debugging and do not directly reflect risk, they are excluded from this view.

A critical requirement applies to metrics used for the assessment of the risk a driving function imposes. In general, a metric should recognize low risk and high risk as such. These are called true negatives and true positives [51], respectively, and are shown in Table 3.2. Measuring high risk when low risk is present, the functional behavior gets falsely rejected by the assessment.

		real risk	
		low	high
estimated risk	low	true negative	false negative
	high	false positive	true positive

Table 3.2.: False positives and False negatives in the context of metrics assessing the risk of driving functions.

Hence, such false positives are undesirable. In contrast, false negatives underestimate the risk present in reality. The consequent acceptance of faulty functional behavior unacceptable for the assessment. Therefore, uncertainty in the metrics should be kept small and biased slightly towards false positives.

In the following, existing metrics from the literature for the assessment of driving tests are elaborated. As accident risk is the most important for the assessment of an AD system's safety, this thesis focuses on that kind of metric. For completeness, brief insights on the other two categories complete this section.

3.2.3.1. Safety Related Assessment Metrics

Many assessment metrics related to traffic safety emerge from the literature. Most of them measure the time until an upcoming incident in the scenario. In the following, various existing metrics from the Situation Threat Assessment (STA) field are presented.

Time-to-Collision (TTC)

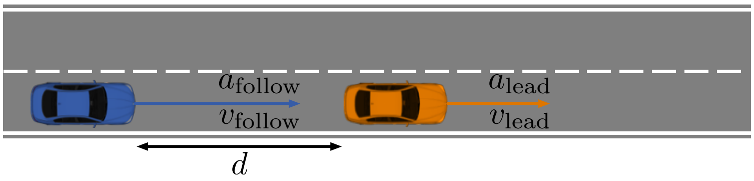


Figure 3.8.: Description of TTC, THW, and ETTC alongside a vehicle following scenario. The EGO vehicle shows in blue and the TO in orange. Variables relevant for the calculations are indicated.

The first and most common metric measures the time left until an imminent collision between two vehicles [114]. For explanation, a vehicle following scenario is illustrated in Figure 3.8. Herein, the blue vehicle with velocity v_{follow} and acceleration a_{follow} trails and orange vehicle with velocity v_{lead} and acceleration a_{lead}. The Time-to-Collision (TTC) now calculates to [88]:

$$v_{\text{rel}} = v_{\text{follow}} - v_{\text{lead}} \tag{3.1}$$

$$\text{TTC} = \frac{d}{v_{\text{rel}}} \qquad v_{\text{rel}} > 0\frac{m}{s} \tag{3.2}$$

Notably, the TTC is only defined for a positive relative velocity v_{rel}, as there will never be a collision if the following vehicle is slower. The simplified formula in Equation (3.2) only applies to this exact scenario. If the distance d calculates along the vehicles' predicted paths to the collision point, the formula applies to arbitrary scenes. Still, only a single future path for TOs is considered, and uncertainties in their driving intention are ignored.

Enhanced Time-to-Collision (ETTC)

As the standard TTC only incorporates the participating vehicles' velocity, a logical extension is the inclusion of the acceleration to increase the precision of the metric. The resulting Enhanced Time-to-Collision (ETTC) [115, 88] calculates with the help of the standard TTC through:

$$a_{\text{rel}} = a_{\text{follow}} - a_{\text{lead}} \tag{3.3}$$

$$\text{ETTC} = \frac{\sqrt{v_{\text{rel}}^2 + 2a_{\text{rel}}d} - v_{\text{rel}}}{a_{\text{rel}}} \qquad v_{\text{rel}}^2 > 2a_{\text{rel}}d \tag{3.4}$$

Again, a condition on the relative velocity v_{rel} and acceleration a_{rel} must be satisfied for the ETTC to be defined. Despite being more accurate, the same drawbacks as for TTC apply.

Worst-Time-to-Collision (WTTC)

Another extension to the standard TTC is its worst-case estimation, Worst-Time-to-Collision (WTTC). Herein, not only a single path of the vehicles is considered – a whole set of physically drivable trajectories in predicted by the use of a simple vehicle model. As a result, a broader set of TTCs emerges from collision checking, and its worst case is selected to represent the scene [116]. It can be interpreted as the time until a collision, while every traffic participant tries to achieve a collision as fast as possible. This metric is the first that considers multiple possible TO behaviors and hence includes uncertainties in human intention. However, as it overestimates the scene's risk due to the worst-case consideration, it is useful to filter large data sets for potentially critical scenes rather than for assessment.

Time-Headway (THW)

The Time-Headway (THW) metric can be considered as a simplification of the TTC. It only contemplates the velocity v_{follow} follow of the following vehicle and is hence defined for a broader range of velocities. The calculation is done as follows [117]:

$$\text{THW} = \frac{d}{v_{\text{follow}}} \qquad v_{\text{follow}} > 0 \tag{3.5}$$

The THW is used in the German Straßenverkehrsordnung (StVO) to calculate fines for tailgating [118]. Being a simplification of TTC, the same drawbacks apply. It is a good measure for traffic congestion and flow rather than for safety.

Time-to-React (TTR)

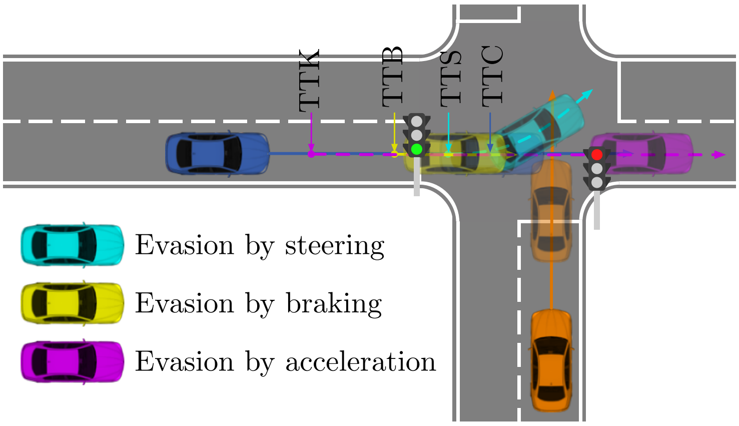


Figure 3.9.: Description of TTR alongside a crossroad scenario. The EGO vehicle pictures in blue and the TO in orange. Evasion plans are shown in light blue for steering, yellow for braking, and purple for acceleration.

As the time until an upcoming incident is not the only indicator of criticality, [119] includes information about the EGO vehicle's evasion options in the Time-to-React (TTR) measure. For upcoming collisions, three basic evasion options are defined and illustrated in Figure 3.9. In a crossroad scenario, the blue vehicle's path is endangered by an orange vehicle ignoring the red light. Without intervention, a crash would happen after the expiration of the time defined by the TTC. As a first avoidance option, full braking is investigated. Thereby, the orange vehicle can pass before the blue vehicle enters the critical area. The maneuver shows in yellow. The time left until full braking avoids the accident estimates as Time-to-Break (TTB). Algebraic solutions for the TTB are proposed in [120]. A second option is passing the critical area before the orange vehicle through full acceleration. As depicted through the purple maneuver in Figure 3.9, acceleration is not as powerful as braking, and the maneuver needs to be initiated earlier, resulting in a lower Time-to-Kickdown (TTK). As the last option steering to the left or the right remains. In the exemplary scenario, evasion by full left steering indicates in light blue. It leaves the most time to avoid defined by the Time-to-Steer (TTS).

Now, as all of the above-mentioned evasion strategies reveal different reaction times, the TTR is defined as the maximum of those, hence the evasion strategy leaving the most reaction time for the EGO vehicle:

$$\mathrm{TTR} = \max\left(\mathrm{TTB}, \mathrm{TTK}, \mathrm{TTS}\right) \tag{3.6}$$

The TTR identifies an improved understanding of risk as it further incorporates the EGO's options to avoid an upcoming accident. However, the uncertainties in TO driving intentions are still ignored.

Predicted-Minimum-Distance (PMD) and Time-to-PMD (TPMD)

After stating the insufficiency of static STA calculations such as TTC and THW for next-generation ADAS, [121] introduces Predicted-Minimum-Distance (PMD) together with Time-to-PMD (TPMD). For this value, a future collision on predicted vehicle paths is not mandatory. Instead, the algorithm first searches for the shortest distance between the EGO and other TOs or obstacles in the scene, resulting in the PMD. Thereafter, the TPMD is set as the time until this minimum distance is reached. Improvements of this metric are present as risk can also be measured for a so-called ”close call” situation, where vehicles pass each other too close to be considered safe. Still, only a single future trajectory for uncertain TOs incorporates in this approach.

3.2.3.2. Regulatory Metrics

Another critical aspect of the assessment of AD is compliance with traffic rules. Despite being inferior to the avoidance of accidents, they must not be neglected as compliance is mandatory in the same as for human drivers. Following rules directly correlates with safety due to the restriction of risky driving behavior.

A measurement of respective compliance is dependent on the particular rule itself. For example, an exceedance of the currently applicable speed limit is a simple comparison, while yielding the right of way for other traffic participants requires a more profound knowledge of the road topology. Additional difficulties arise from the distinctions in the rule sets between countries. While in Germany, overtaking on the right is prohibited, it is allowed in parts of the USA and advisable to maintain the traffic flow.

Some research exists formulating traffic rules with High Order Logic (HOL) [122] or Linear Temporal Logic (LTL) [123]. While this generally covers the applied rules, only a few simple examples are given. However, rules are numerous and sometimes fuzzily defined. Alone in Germany, the code of law called StVO [118] defines its rules within 53 paragraphs on more than 200 pages. Due to the increased focus on risk metrics in this thesis, regulatory metrics are disregarded.

3.2.3.3. Qualitative Metrics

Finally, a complete assessment should address the driving quality for passengers. Subsidiary to the last two categories, it must not be neglected as the passengers' subjective wellbeing is crucial for market acceptance [104]. As previously stated, one can divide this category into physical driver comfort and subjective perception of safety.

In terms of physical comfort, the well-known motion sickness [124] becomes an additional problem for the driver for AD that usually only exists for co-drivers. Here, the mismatch between perceived motion through distraction by non-driving-related activities and the vehicle's real motion causes the indisposition. Studies [125, 104] quantify this effect through user surveys in exemplary test cases. While longitudinal and lateral acceleration is the leading cause of motion sickness, the degree of distraction identifies as a significant influence. Despite this work targeting limits of acceleration for comfortable driving depending on the passengers' distraction, the user survey's interim results are still valuable for quantifying the discomfort.

A metric for subjective safety perception emerges [126]. Again through user surveys, the influence of relative acceleration and velocity to close by vehicles could be mapped the passengers feeling of driving safety.

As this category is less important than actual driving safety for the preliminary study of AD assessment methodologies, the rest of this thesis will not focus on qualitative metrics.

There exists a variety of metrics for the assessment of driving scenarios. Simple metrics suffer from restriction to a specific scene. For example, the standard definition of the TTC is designed for vehicle following scenarios. More complex algorithms exist but rely on assumptions that are not met with the requirements of assessments. For instance, WTTC overestimates the risk, which results in a larger quantity of false positives and possible rejection of a well-behaving AD function. No found metric addresses the rate of false negatives that need to be avoided at any cost.

Hence, the first research question addresses the definition of a scenario assessment metric satisfying all those requirements. Having defined test cases, domains, and assessment metrics, it follows how this information can consolidate into the assessment of a whole driving function. The next section introduces regulatory boundary conditions, established methods, and approaches to AD.

3.3. Assessment Methodology and Approaches to Automated Driving

Having defined how a single test in the automotive domain executes, the next part embeds this into the superordinate assessment methodology. A short outline of applicable rules and standards for testing forms the basis for a review of established present methods for the assessment of ADAS. Problems that arise when applying these to AD derive from the literature review of assessment approaches to AD. Finally, concerns about the coverage of the scenarios these methods can provide can follow.

3.3.1. Norms for the Assessment

AD takes, depending on its level, more or less driving responsibilities from human drivers. Malfunction can cause harm to human health and property. Hence, it is unarguably a safety-critical system. There exist norms that specify the requirements on safety during development, market release, and lifecycle safety management. It is worth mentioning that norms are not rules from regulatory bodies and do not imply punishments through violation. Nevertheless, norms are usually elaborated by collaboration between the OEMs and normative institutions such as ISO, putting relief on a company's responsibility to define safety requirements. Additional, the certification and compliance with norms are beneficial when dealing with legal issues due to safety issues after marker release.
For the development and assessment of ADAS, two particularly important norms are widely accepted. ISO 26262 defines requirements on system malfunction and failure rates. These are completed by ISO 21448, where the limitation and misbehavior of a driving function itself.

ISO 26262 - Road Vehicles – Functional Safety

ISO 26262 [21] is the automotive adaption of International Electrotechnical Commission (IEC) 61508 [127] for general safety concerns with electronic and electrical systems. While it applies for road vehicles with four wheels or more, an extra chapter for motorcycles is added since the second revision. It defines four Automotive Safety Integrity Levels (ASILs) as a system's risk level by three factors [128]. The first is the severity of a potential hazard for human health or property. Then the exposure to this hazard of the frequency of its occurrence is incorporated. Lastly, controllability is important as there are possibilities for a system to avoid or mitigate hazards. The ASIL classification form A to D then directly results in safety goals and then again requirements on the function. According to [129], AD qualifies for the most critical ASIL D and is subject to the highest safety demands.

Other than that, this norm supports the collaboration between OEMs and suppliers. It is continuously developed and improved with the last revision published in 2018. A direct consequence of ISO 26262 is the development alongside the V-Model, which will be further addressed later in Section 3.3.2.

ISO 21448 - Road Vehicles – Safety of the Intended Functionality (SOTIF)

While the previous norm concerns the fault of a system, ISO 21448 [130] treats functional safety in the absence of system failures, particularly if the function behaves as expected and within its limitations [131]. It also helps to find unknown limitations, analyses the effect of sensor and actuator limitation, and the faulty by humans of the intended functionality. Finally, validation and verifications measures to achieve SOTIF are given.

Indeed, a lot more norms are applicable, especially when considering the submodules of AD isolated. For this work, the short prospect into the matter suffices for later argumentations. Based upon norms establish strategies and methods for the assessment. A state of the art of established approaches for ADAS and suggested approaches for AD is discussed in the following.

3.3.2. Assessment of Advanced Driver Assistance System and its Problems with Automated Driving

Market-released vehicles ship with ADAS for several years. Thus they have already been assessed and certified with the norms through an entrenched process. This process generally follows the V-Model for the design, development, and assessment of such systems, as mentioned before. This section explains a simplified version of this model. Additionally, the use of test domains within this process elaborates, followed by a discussion about the problems arising with this assessment and AD.

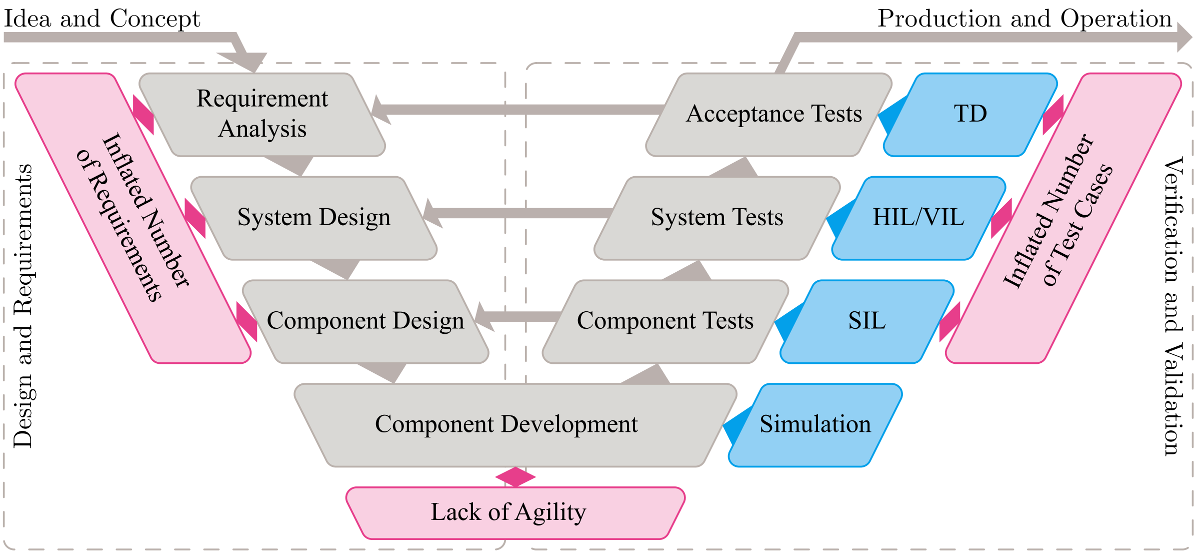


Figure 3.10.: A simplified V-Model for the assessment of ADAS shows in grey [132]. The domains commonly used for testing on different levels are shown in blue and are adapted from [132, 133]. Lastly, arising problems when applying this strategy to AD are displayed in red.

The V-Model in the Context of Automotive Assessment

A simplified version of the complex V-Model defined in ISO 26262 [21] is shown in grey in Figure 3.10. The idea and concept of the desired function result in a requirement analysis first. Following the model's left side downwards, the whole function is refined in its system design and partitioned into components that can be designed and developed independently. Thereby, also the requirements are inherited and further refined to the components level. Predefined requirements allow for Test Driven Development (TDD) [134], where the test cases are defined before the actual function development starts and are continuously tested during programming. This stage creates and checks the proof of concept functionality using simulation. Thereby, errors in the code or function logic can be detected early on where corrections are fast. Eventually, the components assemble, and the function's code is tested in its intended language with SIL in the component testing stage. Later on, when dedicated hardware for the component is available, HIL and VIL provide a more realistic assessment of single or multiple components together through system tests. Finally, acceptance tests follow when the final product with all its components assembles. As those are relevant for the market release of a safety-critical system, real-world TDs ensure the test results' validity. Usually, $1 \cdot 10^5$km to $1 \cdot 10^6$km of continuous operation is necessary for an ADAS [20]. The classification of the test domains shown here at different levels of the V-Model bases on [132, 133]. On every layer, the V-Models allows for redesigns and adaptions of the function through feedback loops if tests fail.

Validation and Verification

The right half of the V-Model describes *verification* and *validation.* These are not synonyms, and their distinction is essential for the assessment. As definitions are not consistent across literature [135], the statement used in this work follows.

Verification is the process of ensuring that "*... a simulation computer program performs as intended*" [136]. This can be in the form of ensuring that implemented code itself is correct or examination of the underlying models. In the scope of AD, assessment *verification* can be implemented by checking if test results concur between real-world and virtual tests with sufficient accuracy [137]. Hence, such a *cross-verification* of test domains can help with the trustworthiness of simulative or virtual results.

In contrast, the purpose of *validation* is to make statements of systems properties such as safety as a whole.. For the assessment method *validation* "*... aims to build the statistical argument to confirm the safety across both known and unknown scenarios with enough confidence*" [129].

By means of those definitions, the term *verification* is used in the following to describe the process of ensuring the correctness of the assessment method itself. In particular, *cross-verification* makes sure that virtual test domains provide adequate results by comparison with already verified tests. *Validation* is then the process to ensure the system's correctness and safety under test, thus the AD function. It is intentionally not used to call a virtual test domain *validated*, as a simulation to any degree of abstraction can never be validated as a whole [99]. For the AD function, a proven 100% validity is never possible due to the complexity of the system under test. Hence, the *validation* builds a statistical argument with a quantified residual risk, a concept that is accepted by society for a long time [129]. Those definitions will become more important later on while discussing assessment approaches to AD.

Problems with the V-Model and the Assessment of AD

The presented V-Model successfully assists the market release of ADAS for several years. This is possible as the ODD for an ADAS is generally limited. For example, ACC operates only while following a road lane behind another vehicle. Thereby continuous operation of about $1 \cdot 10^5$km to $1 \cdot 10^6$km [20] on real roads offers sufficient stochastic proof of safety on the testing branch's highest level. However, the requirements for AD concerning the ODD exceed the scope of ADAS. Depending on the level of automation, the function needs to operate in any possible driving scenario. Different sources state that this raises the number of necessary driving distance to $5 \cdot 10^9$km for Germany [20] and $8.8 \cdot 10^9$km for the USA [45] to statistically prove that automated driving is safer than a human driver. It is to mention that these numbers are not directly comparable as the fundamental conditions for the calculation, such as the confidence margin and the type of accidents for the reference vary. However, the conclusion that the number of required testing distance exceeds that of ADAS by far is undeniable. Additionally, these vast distances do not guarantee the coverage of every possible scenario with tests [20]. Rare corner cases will never be revealed. Therefore, scenario-based approaches for ADAS define a number of scenarios that the function needs to handle instead of or additionally to continuous operation as requirements [46, 138]. However, for AD, these are infeasible to define due to its unlimited scope. It is impossible to define every single possible permutation of a scenario, and part of the requirements remains unknown [129]. This problem emerges on the left side of the V-Model from Figure 3.10. Additionally, even if the set of requirements or scenarios could be defined in its entirety, testing those on the right side is expensive, time-consuming, and imperiling for test drivers. Lastly, the amount of requirements and tests makes the procedure slow, and iterations through V-Model feedback loops when tests fail may result in unwanted delays of market release. The approach lacks agility when applied to AD.

3.3.3. Assessment Approaches for Automated Driving Functions

To overcome the above-mentioned issues, several approaches occur in the research literature lately. These can be categorized into virtual assessment, reduction of necessary testing, scenario-based testing, stepwise introduction, and formal verification. In the following, such approaches are presented and discussed within the respective categories.

Virtual Assessment

Due to their lack of validity, virtual test domains are generally not designated for the V-Model's highest level's final market release assessment. However, due to their favorable cost and safer nature, they contain the potential to put relief on the required testing amount. Hence, research in this direction is justified.

Virtual testing environments and test automation are used in [50] to fill the test space with samples progressively. This approach formulates for the general assessment of systems, but an example in an automotive context is present. With this, intelligent search algorithms traverse the possible test space to explore it widely. While this method is suitable for finding system faults early on in the development, there is no guarantee for complete exploration of all vulnerabilities. Additionally, the definition of the test space for AD is an uncovered but necessary part of making a statement about its coverage. Also, the impact of using virtual testing environments on the concept's validity is not examined.

The benefits of virtual continuous operation complementary to the real-world are examined in [51]. Here, Monte-Carlo methods randomize traffic scenarios. A stochastic statement about safety can thereby be accelerated. The responsibility of verifying used simulation models refers to neutral, independent, scientifically accredited institutions without further elaboration.

Both approaches improve the use of virtual test domains for the early detection of faults. Still, the lack of verification prohibits the use in market release assessment. Variations of scenarios still need to be validated against their occurrence probability and meaningfulness.

Reduction of the Necessary Test Volume

Another working point to address the infeasible test amount is the number of required testing itself.

To reduce the mileage on real roads [138] suggests testing against critical scenarios. While this work targets the assessment of increasingly complex ADAS, the result can be mapped to AD ad the underlying problems with the assessment are the same. Critical scenarios are extracted from NDS_1 or FOT and feed into a database. Due to the testing against those, faulty behavior is observant early in virtual test domains. For example, scenarios that appear to be significantly challenging for the function within the simulation are again tested in HIL or DIL. This can be repeated upwards in the V-Model until a small set of scenarios eventually remains for real road testing. Still, the verification remains questionable. Additionally, testing against uncritical scenarios is still essential, as the function could also fail because of implementation errors.

Another approach to reducing the test volume bases on Extreme Value Theory (EVT) [139]. This technique extrapolates the assessment results from insufficient measurements or tests and hence lowers the milage. It is especially useful for measuring the occurrence of rare events such as traffic accidents. The theory is heavily reliant on a measure for accident criticality within test scenarios. This work could only obtain reasonable results for a rear-end collision criticality measure. Hence, an assessment for any kind of accident is missing.

Scenario-Based Assessment

A remaining concern with continuous operation testing is the coverage of possible driving situations. If one records a vast amount of driving data, most of it reveals common scenarios. Corner cases are hard to find and to cover with that approach [140]. Hence, scenario-based approaches aim to achieve uniform coverage of the test space by a priori definition of the cases to test.

A similar method has already been suggested for ADAS [46]. In the first step, the relevant parameters of a scenario, which affect the considered function, are manually analyzed. From this basis, test cases are generated from four layers of the scenario contents, namely road topology, temporary road manipulation, traffic objects, and environmental condition. The test cases generate using combinatoric practices to explore the scenario space densely while avoiding redundant tests. The tests' conduction is carried out mostly in simulation but partly in DIL, VIL, and TD as well. While a test in TD cannot be exactly conducted as intended, the other domains lack verification due to their virtualization. Also, the manual examination of relevant scenario parameters is infeasible for AD.

Another scenario-based approach extracts scenarios from large-scale FOT or with the AD function [47]. Thereby the coverage of the scenarios is as good as the coverage of the data set is. The scenarios are then reprocessed in virtual test domains such as pure simulation. By comparing the results of the simulation with their respective real-world counterpart, the simulation itself can be locally cross-verified. Variations of verified scenarios can then increase the test space coverage. Still, the approach is missing to derive concrete descriptions of the scenario space itself and methods for said variations.

Also, [48] uses real-world measurements for the basis of a scenario space that is covered with variations afterward. In this approach, a parameterized description of a specific type remains manual work. A Kernel Density Estimation (KDE) enables fitting parameter distributions from the real-word measurements for the use in the following importance sampling. An increased number of test cases can then be simulated. Still, the manual definition of scenario parameters is infeasible for the whole test space of AD. Additionally, it states that KDE is unreliable for complex scenarios with many parameters. Again, the verification of the simulation misses.

Another simulative approach emerges from PEGASUS [49]. The project encourages the OEMs to maintain a common scenario database to draw test cases. Through the joint ingestion of measurements, the scenario space coverage increases in comparison to individual efforts. However, due to the different sensor modalities of the OEMs, scenarios might get perceived differently. The gain in data, test case conduction, and evaluation can result in further requirements on the database's test cases. Hence, in an iterative approach, the coverage in resulting safety argumentation is stepwise increased. A detailed elaboration of the concept, especially with respect to the test case conduction and verification of used domains, is missing to this date.

Stepwise Introduction

The number of use cases and thus requirements and test cases necessary for the AD assessment raises the question if gradual fulfillment of those is a solution. This approach termed *seed* *automation* [141] plans the release of automated vehicles with certain restrictions and thereby limits applicable use cases. For example, this can happen on a designated highway section at a limited maximum velocity under good weather conditions [142]. The released vehicles can, in turn, collect data for the safety argumentation. By softening the constraints stepwise and repeating this cycle, AD can eventually cover the original intended use case and requirements. However, the customer acceptance of an expensive system which only operates in a minimal ODD remains questionable. Further, it is unclear how the collected data's safety argument can be extrapolated and validated for the softened constraints.

Formal Verification

Apart from stochastic and scenario-based approaches, *formal verification* replaces classical testing with direct proofs of system safety. Herein mathematical methods are used to explore every state a system can reach and prove their correctness and safety [143]. Certainly, this cannot be done on real roads and requires formalized models of all involved parts, including development tools such as compilers [144]. Hence, for AD, formal verification is based on the assumptions of verified models as well as perfect and correct sensor information [145]. While it might be possible to prove the correctness of the system's software functionality, a complete assessment of AD is infeasible with formal verification. In contrast to the other, this method can be used online for verifying the currently planned trajectory [146].

Although the assessment of AD is in the focus of OEMs and subject to intensive research for several years, its problems and the strategy towards a safety argument are not solved yet. All of the discussed approaches indeed contribute to the answer, but all suffer from unproven assumptions, missing validation, verification, or individual limitations. This motivates the second research question of this thesis, where a scenario-based approach, including locally verified simulation, is examined to put relief on the cast amount of necessary testing.

3.3.4. Coverage of the Assessment

Besides an assessment methodology itself, it is desirable to make a statement if it covers the AD function's ODD. More precisely, if the possible driving scenarios a function can come across during its lifetime are included in the conducted tests. Hence, this thesis ultimately focuses on the coverage of the assessment.

This last research field partitions into three subfields. First, scenario variations are investigated to add to the test volume and therefore increase coverage. Second, the definition of a scenario or test space is necessary to form a basis for coverage assertions. Finally, this work provides an outlook on how this scenario space and variations can be combined to measure the AD assessment's desired test coverage. Therefore, available literature for these topics is also discussed here.

3.3.4.1. Scenario Variation

The variation enables the generation of test cases adding to the coverage of the scenario space. Literature targeting the variation of driving scenarios is sparse. Only a few publications for global scenario variation are available. A combinatorial method pairs with a trajectory planner for automatic scenario generation in [147]. Next, [148] fills a scenario catalog through vehicle behavior models and simulation. Another approach uses a structural model for the generation of the discretely defined parts of a scenario and a behavioral vehicle model on top to generate dynamic elements [149]. Lastly, in [150], search algorithms are used to find a set of critical scenarios in structural data. This work deliberately does not go into further detail of these works as all of them are test case generation approaches or global variations. In the further course, it becomes apparent that due to the lack of validity in simulated global variations, these approaches are not suitable for the assessment of AD. For the scenario-based methodology used in this work, local variations as alterations of locally verified scenarios are necessary. This motivates the research in this vacant area.

3.3.4.2. Scenario Space as Test Space

The definition of a scenario space is the foundation for coverage statements. Prior approaches with ADAS relying on a vast amount of driven kilometers could merely assume sufficient coverage. This section presents literature directly or indirectly addressing the scenario space problem. Some sources indirectly consider the manual definition of the scenario space [46, 48, 151, 152] through a logical and structured process of defining scenarios by human imagination. Then the parameters defining these logical scenarios are specified manually [153, 154], and bounds are extracted from real data [48]. However, this is an exhaustive approach that is further limited to the human imagination. Complete coverage of the specified scenarios with respect to possible scenarios in the real world can never be guaranteed. Further, it is unclear if the logical scenarios' manually specified parameters are enough to describe the complete characteristic of the associated scenario in reality.

In contrast to manual definition is the use of large-scale NDS_1 data [47]. The extraction of scenarios covers the reality as good as the data covers it. However, there is no structure in those scenarios with respect to a space, and the extraction itself is unclear. Without such a relation to a scenario space, the coverage is still not measurable.

Lastly, an approach directly related to the definition of scenario space establishes through clustering [155, 156]. Therein, a large amount of driving data is clustered with supervised and unsupervised random forest techniques based on manually defined scenario features. However, the approach is only applied to simulated data, which does not resemble possible occurrences in the real world. Furthermore, manual feature definition is again an exhaustive task limited to human imagination.

3.3.4.3. Test Space Coverage

The ultimate question of AD assessment is the performance and safety of the AD function and if the coverage of the test within the assessment is a sufficient representation of the reality. Thereby it can be guaranteed that the function does not expose any traffic participant to more than a reasonable amount of risk. Literature directly addressing the test coverage is sparse.

A purely stochastic approach consolidates on a large number of driven kilometers with the AD function [20]. Section 3.3.2 already states that the required test volume for this approach is infeasible. Additionally, coverage can only be assumed and not measured.

For scenario-based assessment approaches, an approach to the necessary test coverage exists in [157]. They conclude that a total number of $6 \cdot 10^{10}$ scenarios is the necessary test coverage for German highways, which is received through by adapting the required driving distance in [20] to scenario-based assessment. However, these thoughts are only theoretical and not further validated. It is stated that for empirical analysis, the required data is not available.

This work's research focus does not challenge the given number of $6 \cdot 10^{10}$ scenarios in [157]. Instead, it aims to provide a proof of concept and outlook on how to achieve the empirical evidence of enough coverage. Certainly, it is out of scope to collect the necessary data for empirical analysis. Here, the aim is to provide a proof of concept in how scenario variation and a well-defined scenario space can be combined to a sufficient and measurable coverage, motivating the third research question.

3.4. Conclusion

Research towards AD has come a long way. In this chapter, the history and state of the art are introduced in Section 3.1 as detailed as relevant for the remainder. After identifying SAE levels 3 to 5 as to be released levels of automation, it addresses the issues of existing assessment methodology with the new ODD of an AD function compared to level 1 to 2 ADAS systems. Identified as the main scope of this thesis, new approaches to the assessment of AD existing in resent literature are analyzed, and drawbacks and caveats are identified.

Section 3.2 examines the components of a driving test in general. Concomitantly, it presents different test domains and their capabilities to test different components of the AD function. Driving scenarios constitute test cases for these domains, and an explanation follows regarding their contents and required terminology. The last part of a driving test elaborates assessment metrics that measure the AD function's performance and safety and decide over a test being passed or failed. The state of the art reveals a lack of such assessment measures applicable to AD, and the first research question is motivated: *How can the safety of automated vehicles be measured?*

In Section 3.3, assessment methodologies extend single driving tests to an understanding of larger test concepts. For that purpose, the section introduces applicable norms, existing and well-established test concepts for ADAS, and the problems arising with their application to AD. A deficiency of a single comprehensive solution arises from a state of the art overview of existing assessment approaches to AD. With scenario-based assessment as the most promising but mostly immature candidate, the second research question is motivated: *How can efficient virtual testing be verified and included in a scenario-based assessment process, and what is the impact of virtual testing?*

Furthermore, a new test concept also bears additional challenges. Consequently, switching from a stochastic and driving distance defined to a scenario-based assessment approach requires the definition of a test space and its coverage. Therefore, the state of the art in scenario variation, scenario space definition, and coverage criteria is elaborated. As this field of research remains unsolved today, it motivated the third and last research question: *How can the set of test cases be defined, traversed, and coverage be estimated?*

In summary, there is a lot of remaining work and research required for the release of AD to the general public. This thesis aims to contribute to a subset of these challenges, as presented above. Consequently, the stated research questions are addressed in each of the next three chapters in their respective order.

4. A Scenario Safety Assessment Metric for Automated Driving

Metrics for the assessment of risk are a vital component for testing procedures. No matter how much relief new assessment methodologies can provide for the necessary amount of test cases, their evaluation through human observers is infeasible and inaccurate. Hence, it is essential to apply several such measures to determine a test's outcome automatically. The measure is named Key Performance Indicators (KPIs) in the following, as they assess the AD function's performance in driving tests.

Whether one observes the risk of releasing AD from the OEMss' perspective or the risk of using them from the customers' viewpoint, both pursue the same goals. As introduced by Section 3.2.3, there are three types of risk. The avoidance of accidents is the highest priority for both stakeholders, followed by compliance with traffic rules. While they are essential from a customer's perspective in favor of their safety, OEM will most likely be held responsible for incidents caused by solely the AD function. Last but not least, the passengers' comfort in such a vehicle is even more important when being driven than when driving themselves. Here, additional effects, such as the explained motion sickness, play an important role [124]. For the core topic of this work, the methodology of the assessment, such KPIs are required. The state of the art in Section 3.2.3 already concluded that available metrics do not fulfill the requirements for the function's assessment. Hence, this chapter addresses the first research question: How can the safety of automated vehicles be measured?

The following dedicates to the design of a KPI that is independent of the scenario, free of false negatives, and not too biased towards false positives. For the scope of this work, the focus is set on a KPI for accident risk. The chapter is based on the findings of the publication [Paper1] and structured as follows. The methodology and theoretic thoughts behind the KPI are presented in Section 4.1. An algorithm implementation and parameterization, as well as experiments for its evaluation, are shown and reasoned in Section 4.2. Finally, the chapter's conclusion is drawn in Section 4.3.

4.1. Methodology of the Accident Risk Metric

According to [128], the desired argument for safety is the absence of unreasonable risk. Further, risk in the scope of accidents is defined as a "*combination of the probability of occurrence of harm and severity of harm*" [128, p. 21]. This implicates that not only the harm itself needs to be analyzed, but also its probability. Further, this aligns well with the common procedure from the literature for the assessment of accident risk. Whether only for the evaluation of planned trajectories or offline for the assessment of test cases, it is done on scene level in two steps [158, 37, 159]:

1. **Scene prediction:** Based on the actual TOs' states in the scene, their future evolution is predicted. In some KPIs presented in Section 3.2.3, such as TTC, ETTC, or THW, this hides in the analytic solution of movement equations. In contrast, other values directly perform numeric predictions of the models. Further approaches distinguish themselves in whether a single trajectory or a probabilistic set is predicted. As the probability of harm is important for risk, a probabilistic prediction is desirable for the KPI.

2. **Risk assessment:** After prediction, a possible threat must be found. In the scope of accident risk, a threat is an imminent collision with stationary or other TOs. As a binary classification of a specific path colliding or not is not sufficient, the severity of the crash is quantified. Finally, the gathered information consolidates into the actual measure of risk.

The KPI presented in this work also follows this rough outline. Figure 4.1 displays a more comprehensive procedure of the algorithm. In the beginning, a stochastic TO prediction using Monte-Carlo simulation provides a set of possible scene evolutions in Section 4.1.1. A subsequent collision detection isolates the trajectories imposing danger to the EGO vehicle. For those trajectories, the TTR estimates the imminence and severity of the threat. The process follows in Section 4.1.2. The consolidation into the final KPI is explained at a single TO level and, eventually, the scene level in Section 4.1.3 and Section 4.1.4, respectively.

4.1.1. Monte-Carlo Scene Prediction

The first step or the presented algorithm predicts possible future outcomes of a scene. One might ask why a prediction is necessary when the complete trajectory of traffic objects is already available in a downstream assessment of a driving test. However, AD has to perform well in an environment with human drivers and their behavioral uncertainty. Proper functional planning must incorporate these uncertainties, which in turn needs to be assessed. Therefore, the

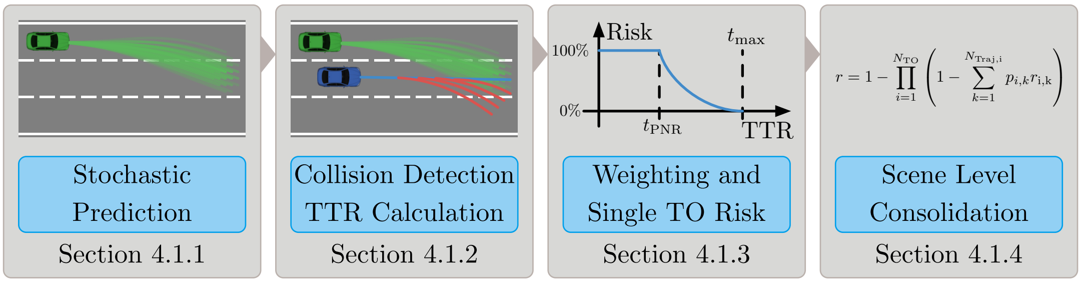

Risk
$t_{\max}$
100%
0%
t_{PNR}
TTR
$r = 1 - \prod_{i=1}^{N_{\mathrm{TO}}} \left(1 - \sum_{k=1}^{N_{\mathrm{Traj,i}}} p_{i,k} r_{\mathrm{i,k}}\right)$
Stochastic Prediction
Section 4.1.1
Collision Detection TTR Calculation
Section 4.1.2
Weighting and Single TO Risk
Section 4.1.3
Scene Level Consolidation
Section 4.1.4

Figure 4.1.: The methodological flow of the accident risk metric shown for a traffic scene in four steps. First, possible future evolutions of the scene are predicted, followed by collision detection with the EGO vehicles trajectory plan and TTR calculation. Through the trajectory probabilities and a weighting function for the TTR, the risk for a single TO is calculated. Finally, consolidation of all TOs leads to the scene risk measure.

stochastic prediction of TOs, incorporating the distribution of human behavior, is necessary and presented in the following. This can either be achieved through Monte-Carlo simulation of object models [159, 87] or the computation of reachable regions in a discretized spatio-temporal domain [160, 161]. While the latter is computationally more efficient and suitable for online assessment, it suffers from an additional source of errors caused by the spatial domain's discretization [160]. As computational constraints are not as hard for offline function assessment, this work uses Monte-Carlo simulation. For that, TO models are required.

4.1.1.1. Vehicle Model

Figure 4.2 presents the three types of TO models [158] that can be used for the prediction. For each of the categories, the same scenario at a traffic intersection is given. Additionally, the figure shows a hypothetical prediction of the green vehicle achieved through the model's respective category. The leftmost starts with physics-based models. Kinematic models of the TO of arbitrary complexity exist. Commonly, a numeric integration of the model's movement equation results in the desired prediction. As only the vehicle's kinematics are modeled, it ignores the environment and other TOs. This results in a prediction, as indicated on the left side of Figure 4.2. In the long-term, the prediction would simply run straight over the intersection towards the embankment and crash into the crossing vehicle, which has the right of way. Hence, physics-based models are suitable for short-term predictions only. Raising the awareness of the road topology can prevent overrunning the intersection and results in maneuver-based models. However, as other TOs are still ignored, the prediction violates the right of way and crashes. Maneuver-based models qualify for midterm predictions. Lastly, the consideration of nearby TOs and their action creates interaction-aware models. In the right scenario of Figure 4.2,

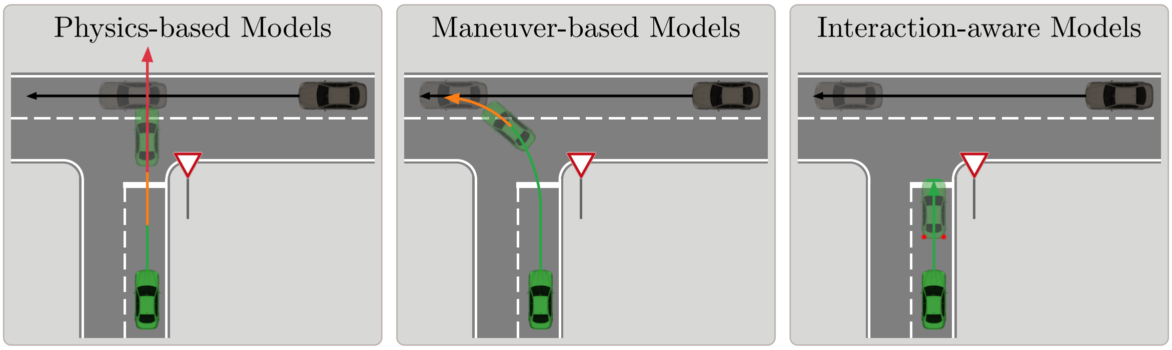


Figure 4.2.: The three types of vehicle models, according to [158]. The physics-based modes are simple but ignore the environment. Maneuver-based models incorporate the road topology but ignore interactions with other TOs. Interaction-aware models incorporate the whole environment, but are complex and require a lot of information about the scene to be calculated.

it shows that the prediction with this category of models would stop for the crossing vehicle. Thereby, profound long-term predictions are possible. From left to right, the predictions using the respective models are valid for a longer period, but more complex, computational demanding, more difficult to implement and require more information for the prediction itself.

In contrast to online assessment methods supporting the AD function's decision making, a shorter period of prediction is sufficient for a post-processing method. The online planning module of the driving functions needs to predict a longer period to prevent accidents in the long run. Certainly, for a critical situation itself, only a short time from the scenario becoming critical until the actual incident is relevant [162]. Situations that are critical in the long run are identified as such anyways by iteration over time in the scenario. Hence, a physics-based model suffices for the proof of concept of the KPI.

There are a variety of kinematic models available for the task of prediction [158]. This work refrains from providing a detailed list and comparison of those, as it exceeds the scope, and such models can be arbitrarily detailed and complex. A comprehensive overview can be gathered from [162]. For the presented KPI, the Constant Turn Rate and Acceleration (CTRA) model [159] is used and described alongside Figure 4.3. Let $x(t)$, $y(t)$, and $\theta(t)$ define the vehicle's Cartesian position fixed at its center of gravity and orientation towards the x-axis at the time t. Further, its current velocity and acceleration in the direction of the heading are $v(t)$ and $a(t)$, respectively. Together with turn rate $\omega(t)$ the movement of the vehicles can be described with the following set of Ordinary Differential Equations (ODEs):

Through the CTRA model, the movement on the TO is constrained to translational change

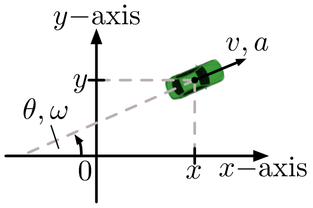


Figure 4.3.: CTRA model description with all variables used in Equation (4.1)

$$\begin{aligned}\dot{x}(t) &= v(t) \cdot \cos(\theta(t))\\ \dot{y}(t) &= v(t) \cdot \sin(\theta(t))\\ \dot{\theta}(t) &= \omega(t)\\ \dot{v}(t) &= a(t)\end{aligned} \tag{4.1}$$

in the direction of the heading $\theta(t)$ through acceleration $a(t)$ and change of the heading over time through $\omega(t)$. Hence, the model is suited for TOs, such as cars, trucks, and motorcycles. Different models are required for other types, such as pedestrians. For this proof of concept, the consideration of cars suffices.

4.1.1.2. Dimensioning of the Parameter Variation

A Monte-Carlo simulation varies the model's parameters. Now, the central question is, which are these parameters and how to define the variation's extent and pattern itself. The movement model from Equation (4.1) defines a state-space system with four states x, y, θ, and v, and two inputs a and ω. As there are no additional parameters in the model, the only variables qualifying for variation are the inputs ω and a. Variating system states would violate the equations themselves.

In any given scene, the current inputs initial values a_0 and ω_0 at the scenes time set to $t = 0$ are known. Most likely, a human driver will keep this intention for a short period. Certainly, deviations from that are possible. The farther away from the inputs, the less likely they are. To find the distribution of a and ω, this work utilizes the NDS$_1$ euroFOT [92]. A first glance has shown that they are not only dependent on a_0 and ω_0, but also from the initial velocity v_0. Hence, representative scenes are extracted for a wide range of TO parameter combinations a_0 and v_0 or ω_0 and v_0, respectively. Additionally, the scenes are chosen to be at minimum $5s$ apart from each other to avoid correlations. Exemplary findings for $v_0 = 14\frac{m}{s^2}$ ($\sim 50\frac{km}{h}$) are shown in Figure 4.4 for a on the left and ω on the right. Only the left plot for a is explained in the following as the right presents the data for ω analogously. The gray scatter points show the raw data based on the initial values on the horizontal axis and the average evolution of the variable over the next second on the vertical axis. At the three exemplary initial conditions, $0.0\frac{m}{s^2}$, $1.2\frac{m}{s^2}$, and $2.4\frac{m}{s^2}$ for a_0, histograms for a are shown in the lower half of the plot in red, blue, and green, respectively. A Gaussian-like distribution of the histograms is observable. To be able to sample from the respective distributions, functions for their descriptions are necessary. Hence, the distributions f_a and f_ω for the parameter variations are formulated as

$$\begin{aligned} f_a(a|v_0,a_0) &= \frac{1}{\sigma_a(v_0,a_0)\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{a-\mu_a(v_0,a_0)}{\sigma_a(v_0,a_0)}\right)^2} \\ f_\omega(\omega|v_0,\omega_0) &= \frac{1}{\sigma_\omega(v_0,\omega_0)\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\omega-\mu_\omega(v_0,\omega_0)}{\sigma_\omega(v_0,\omega_0)}\right)^2}, \end{aligned} \tag{4.2}$$

with μ_a and μ_ω being the mean and σ_a and σ_ω the standard deviation of the distribution. Fitting the distribution functions is achieved through maximization of the logarithmic likelihood

$$\begin{aligned} \mathcal{L}(a|\mu_a,\sigma_a) &= -\frac{N_{\text{pt}}(v_0,a_0)}{2}\log(2\pi\sigma_a^2) - \sum_{n=1}^{N_{\text{pt}}(v_0,a_0)} \frac{(a_n-\mu_a)^2}{2\sigma_a^2} \\ \mathcal{L}(\omega|\mu_\omega,\sigma_\omega) &= -\frac{N_{\text{pt}}(v_0,\omega_0)}{2}\log(2\pi\sigma_\omega^2) - \sum_{n=1}^{N_{\text{pt}}(v_0,\omega_0)} \frac{(\omega_n-\mu_\omega)^2}{2\sigma_\omega^2}, \end{aligned} \tag{4.3}$$

with $N_{pt}(v_0,a_0)$ and $N_{pt}(v_0,\omega_0)$ being the number of parsed data points of the euroFOT data set for the respective initial parameter configurations. Additionally, a_n and ω_n denote the nth samples of said sets.

Repeating the progress for every parameter combination where enough data is present, results in parameter surfaces for μ_a, μ_ω,σ_a, and σ_ω that are shown in Figure 4.5. Therein, the top row shows the means μ_a and μ_ω. One can observe, that the means are roughly diagonal with respect to the initial conditions a_0 and ω_0. This enforces the statement that a human driver will most likely retain his driving intentions in the near future. However, the farther away from $0\frac{m}{s^2}$ or $0\frac{rad}{s}$ the initial inputs are, an increasing tendency of the mean towards zero is observable in the fitted distributions from Figure 4.4. That is because it is more likely, that a human driver tends towards a neutral driving state after performing an action such as acceleration or turning. The initial velocity v_0 has less influence on the means themselves than on the range where they are defined. The faster a vehicle is, the less range for acceleration or turning rate is available. This is reasonable due to the vehicle's limited power of the engine and traction of the tires. As a consequence, this parametrization automatically includes the physical boundaries of TOs. The lower two plots show the same surfaces for σ_a and σ_ω. Those parameters' availability is the same as for their mean counterpart as mean and standard deviation are estimated together. One can observe that the standard deviation is greater the farther away from the initial condition it is. This can be explained by the nature of human drivers again. While driving neutrally with constant velocity and direction, a driver is less likely to change that behavior than while driving at a certain acceleration of while turning. A driver will not accelerate or turn at the same rate

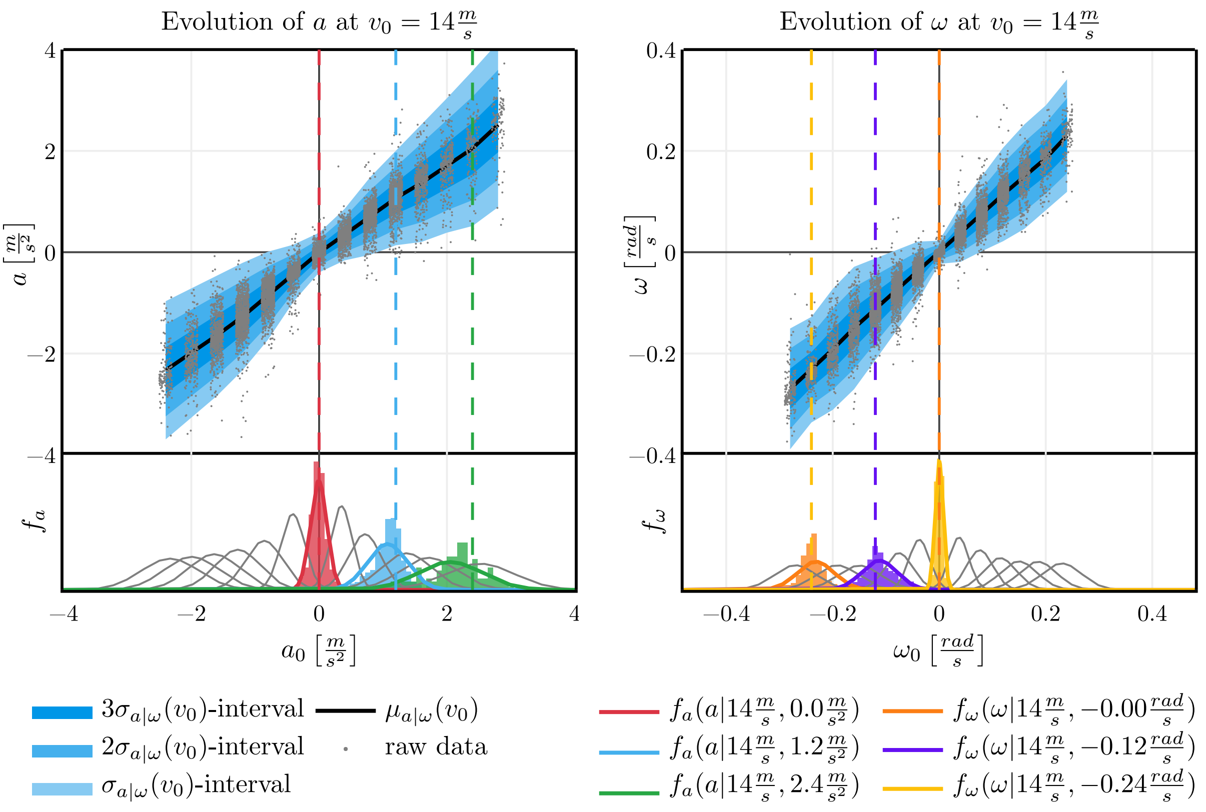


Figure 4.4.: The evolution of the inputs a and ω for the CTRA model in Equation (4.1) at $v_0 = 14\frac{m}{s}$. The raw data extracted for various initial values of the inputs against their future values are shown as gray scatter dots in the upper halves. Histograms and fitted Gaussian distribution functions are shown in the lower halves. The intervals for the σ, 2σ and 3σ ranges of the distribution functions are pictured as different shades of blue in the upper halves for the whole range of initial parameters.

as long as he would stay in neutral driving. Summarizing, the derivation of the model input parameters from NDS$_1$ delivers naturalistic, explainable, and realistic results that can be used for Monte-Carlo short-term prediction of scenes.

Lastly remains the question, how these distributions can be used for scene prediction of TOs. A sampled (a,ω) pair is necessary for each prediction. Hence, the probability distributions are combined for the sampling process. Under the assumption of independence between the distribution, the joint distribution function $f_{a,\omega}$ calculates as follows:

$$f(a, \omega|v_0, a_0, \omega_0) = f_a(a|v_0, a_0) \cdot f_\omega(\omega|v_0, \omega_0) \tag{4.4}$$

For the assumption of independence, bivariate Gaussian distributions incorporating dependency were fitted, and no remarkable improvements were observable. The sampling itself can now

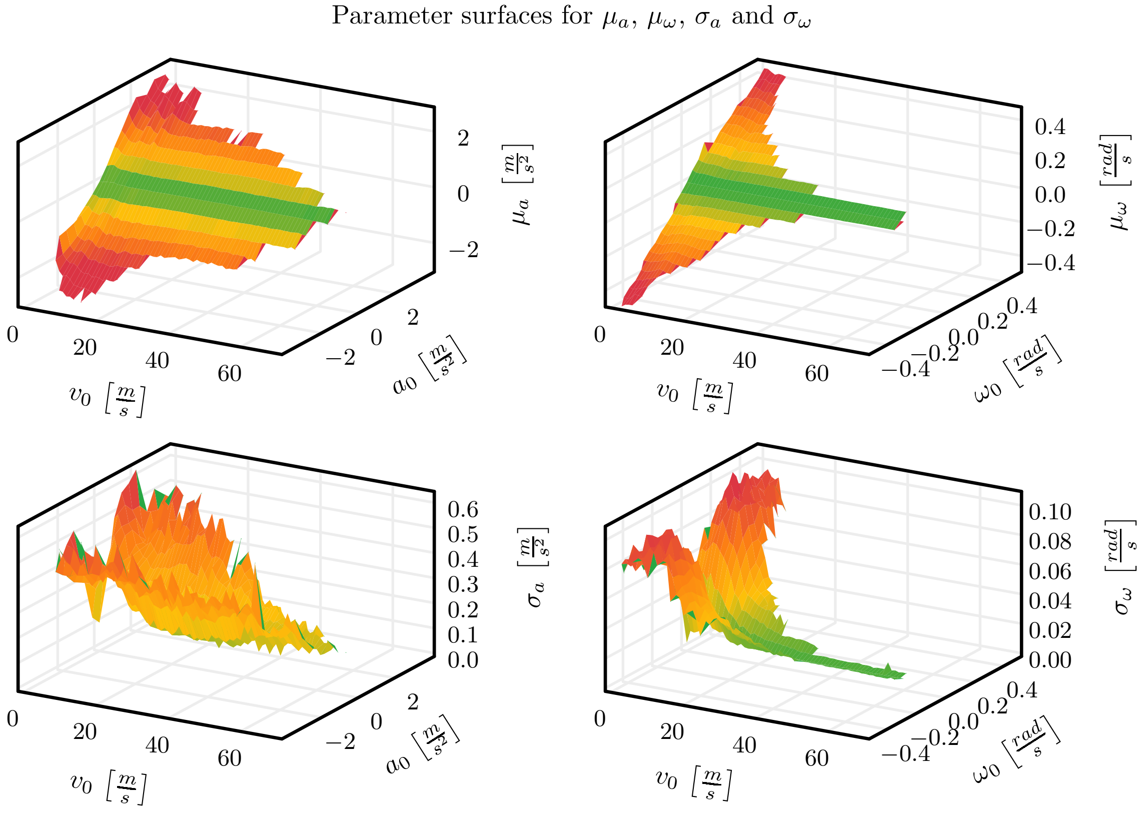


Figure 4.5.: The values for μ_a, μ_ω, σ_a, and σ_ω of the fitted Gaussian distribution functions are plotted for the whole range of initial inputs a_0, ω_0 and velocity v_0. The upper row shows the means and the lower row the standard deviations. The left displays the two parameters for the acceleration a and the right for yaw rate ω.

make use of the joint distribution. Two different techniques can achieve this. The first is sampling uniformly and transforming the uniform distribution into the parameterized Gaussian distribution through its inverse cumulative distribution function. Thereby the sample points assemble densely around the distribution's center just as the raw data used for the fitting. The second method called importance sampling [163] samples uniformly but assigns probabilities to the samples. With the first method, many samples are necessary to reach less probable parameter configurations. By contrast, the second method can achieve rare cases with fewer samples through the uniform distribution. Hence, importance sampling is applied. As the distribution functions' magnitudes are related to the physical quantity they represent and not the probability, they need to be normalized. With a given sampled set $\mathcal{V}$ containing N_{smp} tuples of (a, ω) for the prediction, the probabilities calculate from the distribution through

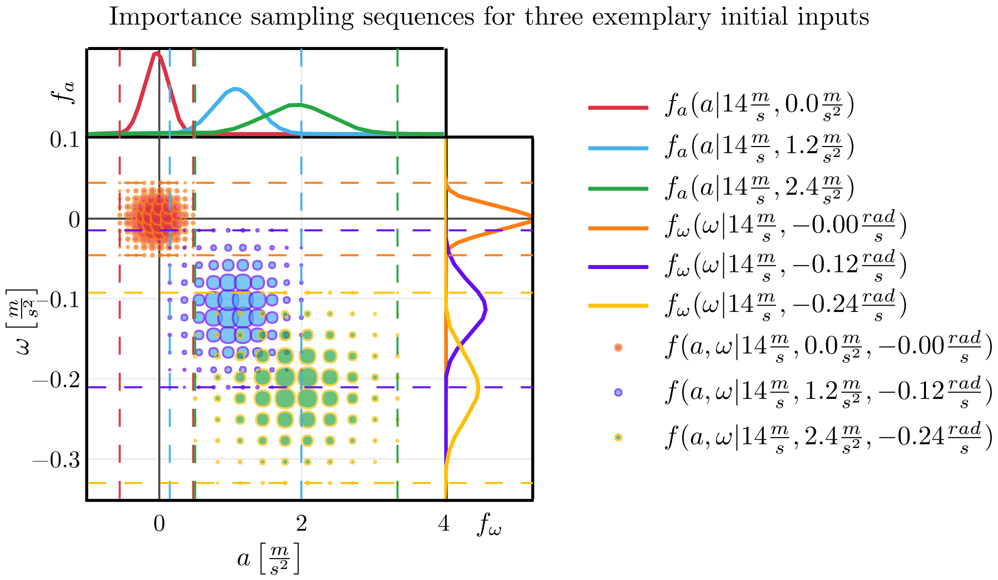


Figure 4.6.: The three exemplary shown distributions from Figure 4.4 for each a, and ω are combined in this plot to their joint distributions, and samplings are presented. With an increasing probability of the sampling points from the importance sampling method, the respective scatter point increases in size.

normalization:

$$p(a, \omega|v_0, a_0, \omega_0) = \frac{f(a, \omega|v_0, a_0, \omega_0)}{\sum_{\{a', \omega'\} \in \mathcal{V}} f(a', \omega'|v_0, a_0, \omega_0)}, \tag{4.5}$$

Figure 4.6 now shows three exemplary input distributions. Here, the selected exemplary distributions from Figure 4.4 combine to their joint distributions, and $\mathcal{V} = 100$ samples are uniformly generated for each. The uniform distributions sample the extent of the 3σ-interval around the respective mean. Thereby, they cover 99.73% of the possible inputs through the empirical 68-95-99.7 rule [164]. The size of the scatter-point scales with the calculated probability of the assigned model input combination. Through importance sampling of the 3σ-intervals, a broad range of possible inputs can be covered with a reasonable amount of samples. In the next step, those samples are used for the Monte-Carlo prediction.

4.1.1.3. Monte-Carlo Prediction

With a model and the distribution of its inputs available, the Monte-Carlo prediction follows next. Due to the simplicity of the model from Equation (4.1) and the short-term character of the prediction, a forward simulation through *scipy*'s *odeint()* solver suffices [165]. Every trajectory is predicted for a period of $T_{pred} = 3s$ with integration step size $T_{step} = 0.1s$.

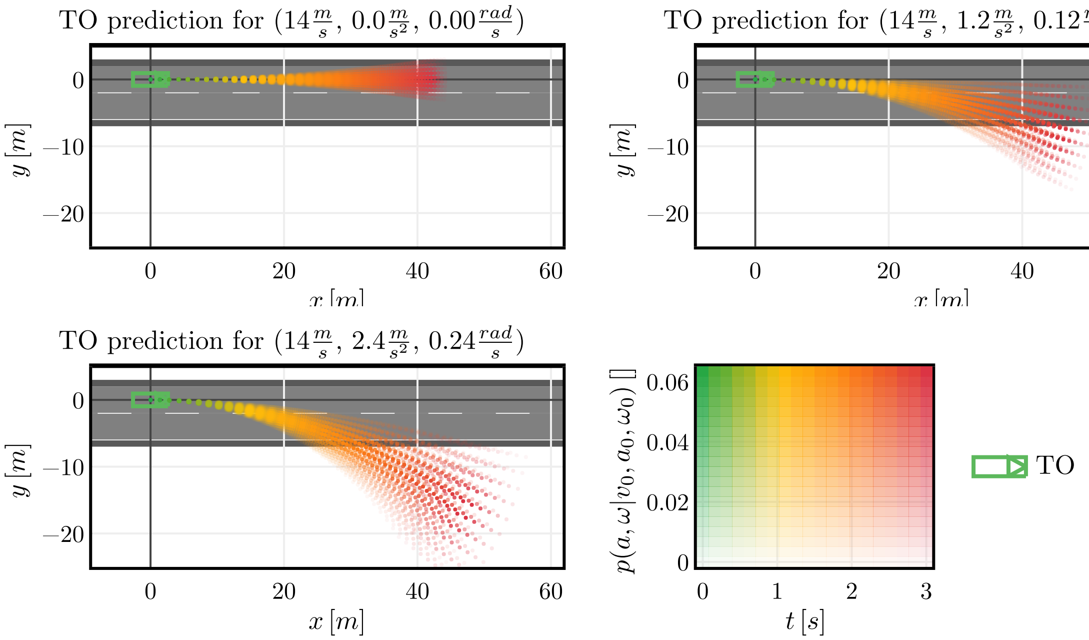


Figure 4.7.: Monte-Carlo predictions of a TO for the three tuples of $(v_0,\ a_0,\ \omega_0)$. In the three shown situations, the predictions display as scatters with changing color dependent on the prediction time and fading opacity with decreasing trajectory probability.

For the three exemplary tuples $(v_0,\ a_0,\ \omega_0)$ from Figure 4.6 the $N_{smp} = 100$ predicted trajectories each display in Figure 4.7. For each plot, the TO is shown at the coordinate systems origin and headed into positive x-direction. A road is deposed into the background to emphasize the spatial extent of the prediction. It is to mention that this road does not serve any other purpose. Neither it participates in the prediction itself, nor it is a realistic candidate for the present initial conditions of the TO. Each predicted path shows as a sequence of scatter points colored depending on the related time step of the prediction. The colors are intentionally chosen to show the validity of the prediction similar to Figure 4.2, but no actual scale for the validity is applied. Lastly, the paths fade out with lower probability as calculated from Equation (4.5). The upper left plot shows neutral driving at $v_0 = 14\frac{m}{s}$. Its prediction corridor is narrow and remains on the same lane. A slightly more dynamic scene with initial condition $(14\frac{m}{s},\ 1.2\frac{m}{s^2},\ 0.12\frac{rad}{s})$ displays at the top right. This could be an initiated lane change to the right, for instance. This time, the prediction corridor is wider, as a dynamic scene is more likely to change than a static such as in the previous plot. With passing time, some of the predictions leave the road. However, this invalidity might not only result from the prediction, but also the made-up road topology or initial state of the TO. Lastly, a highly dynamic scene with $(14\frac{m}{s},\ 2.4\frac{m}{s^2},\ 0.24\frac{rad}{s})$ is shown in the lower left plot. This time, all of the TO's predictions leave the road. Certainly, initial parameters for this experiment locate at the physical boundaries of the vehicle to show the variance in predictions at their limits. Compliance with the road topology is thus not expectable.

The predicted paths with the presented method provide naturalistic future behavior of TOs as they derive from NDS_1. It is valid for a short-term prediction as neither road topologies nor vehicle interaction is taken into account. However, the results gather with simple models, a basic set of information needed about the respective TO, and short-term prediction suffices for the task of STA. An investigation of road-dependent prediction models could still provide benefit for the short-term prediction, as shown in the examples from Figure 4.7. However, for the scope of this work, it is skipped. Having the predictions ready, the actual threat assessment follows.

4.1.2. Collision Detection and TTR Calculation

From naturalistic path predictions alone, a conclusion for the risk of the situation can not be drawn. Hence, the second part of STA involves the calculation of risk-related measures. A variety of measures provides Section 3.2.3, with most of them relating to imminent collisions. As mentioned in the introduction of Section 4.1, the risk is the combination of an incident's occurrence probability and severity [128]. The severity of human harm calculation is an entirely different research scope [162] and out of the scope of this work. Nevertheless, the severity of a situation also correlated with the time to the upcoming incident (TTC), which leaves time to avoid an accident or mitigate the severity of human harm. Therefore, this section explains the collision detection and the applicability of the existing STA measures TTC and TTR on the available predictions.

4.1.2.1. Collision Detection and the TTC

The collisions of interest are those, which involve the EGO vehicle and any TO. For that, the EGO vehicle inserts into the scene with its path from trajectory planning. Therefore, only a single future path for the EGO is given and assumed certain[1].

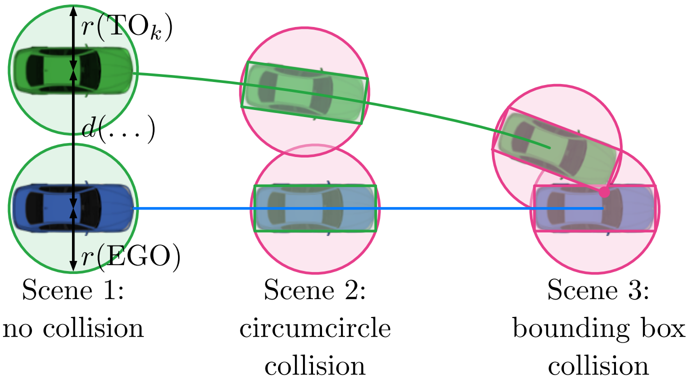
The functional principle of collision detection itself follows alongside Algorithm 4.1. To find any possible collision utilizing the stochastic scene predictions, the algorithm has to iterate over all N_{TO} TOs in the scene, all $v_i^{(k)}$ predicted trajectories $N_{smp}^{(k)}$ from the set $\mathcal{V}_k$ of the current TO k

[1] Indeed, the exact future path of the ego vehicle is not entirely certain as errors arise while the motion control follows the planned path and the EGO vehicle is allowed to change the planned path in future time steps. However, motion control errors are small compared to human driver uncertainty in TOs. Additionally, the assessment metric intentionally measures the safety of the AD function's current plan in a specific scene. Replanning will affect future scenes.

```
    Input: predictions of all $\mathrm{TO}_k$ TOs
    Output: list of collisions between TOs and the EGO vehicle
 1  Function collision_detection(predictions) : collisions is
 2      List collisions = [ ];
 3      for $\mathrm{TO}_k \leftarrow 1$ to $N_{\mathrm{TO}}$ do
 4          for $v_i^{(k)} \in \mathcal{V}_k$ do
 5              for $t \in [0.0, T_{step}, \ldots, T_{pred}]$ do
 6                  if vehicle_intersection($\mathrm{TO}_k$, $v_i^{(k)}$, $t$) then
 7                      collisions.append(($\mathrm{TO}_k$, $v_i^{(k)}$, $t$));
 8                      break;
 9  return collisions;
10  Function vehicle_intersection($\mathrm{TO}_k$, $v_i^{(k)}$, $t$) : collided is
11      if $d(\mathrm{TO}_k, v_i^{(k)}, t) \leq r(\mathrm{TO}_k) + r(EGO)$ then
12          Bool collided = false;
13          for corners of $\mathrm{TO}_k$ do
14              if ray_cast(corner, EGO) then
15                  collided = true;
16          for corners of EGO do
17              if ray_cast(corner, $\mathrm{TO}_k$) then
18                  collided = true;
19  return collided;
```

Algorithm 4.1: Pseudocode definition of the collision detection algorithm and its two-step sub-algorithm for checking the intersection of two vehicles.

and over every time step of the prediction $t \in [0.0, T_{step}, \ldots, T_{pred}]$ in lines 3, 4 and 5 respectively. This nesting is described inside the function *collision_detection()* and results in a large number of necessary collision checks. Therefore, the collision check from function *vehicle_intersection()* called every iteration efficiently operates in two steps. As depicted in figure Figure 4.8, the first step checks if the bounding circles intersect in line 11 of Algorithm 4.1. For that, the radius from a vehicle's geometric center to any corner calculates as $r(\mathrm{TO}_k)$ or $r(\mathrm{EGO})$ for the current TO or the EGO vehicle, respectively. If their sum is less then the distance $d(\mathrm{TO}_k, v_i^{(k)}, t)$ between the two vehicles' geometrical centers, the current constellation is eligible for a more sophisticated collision check. In scene 1, they do not intersect, and the function returns *false* immediately. The intersection is present in scene 2, but the two vehicles did not collide yet. Therefore, the



Scene 1: no collision

Scene 2: circumcircle collision

Scene 3: bounding box collision

Figure 4.8.: Visualization for the *vehicle_intersection()* algorithm from Algorithm 4.1 with a single TO, a singe trajectory prediction, and three scenes for simplicity. In the first scene, the circumcircles of the vehicles do not intersect, and further checks for a collision are skipped. While the second reveals this intersection, a more computational demanding bounding box intersection check is performed, which finally reveals a collision in scene 3.

intersection of the vehicle's bounding boxes is checked next. Two rectangles intersect if at least one corner of any of the vehicles is inside the bounding box of another vehicle. This case is present in scene 3. In the algorithm, this happens within the iteration over the available corners in lines 13 and 16. Whether a point is within a polygon or not can be determined by the *ray_cast()* algorithm [166] called within the two iterations. If the latter returns *true*, a valid collision is detected and saved to the list of collisions in line 7 of function collision_detection(). The stored information contains the TO index TO_k, its predicted path index $v_i^{(k)}$ and the TTC as the current time t. On collision detection, the iteration over the time steps aborts on line 8 as the path and future possible collisions are invalid and of no interest.

The delineated algorithm is used to find the possible collisions in the three exemplary scenes accompanying this chapter. Figure 4.9 shows the scenes in their familiar sequence, one per row. In each case, the left side displays the scene, including predicted paths and possible collisions, both fading out with less probability of occurrence. The right side contains a histogram of the possible collisions weighted with the probability and colored with the TTC. As the differences in magnitude between the paths probabilities are significant, the histogram's vertical axis scales logarithmically. In the first scene, the vehicles are driving in parallel. The probability of a collision calculated as the sum over all the trajectory probabilities that collide is $\sim 2\%$. What might seem too high for parallel driving, is calculated on the assumption that the EGO will not change its planned trajectory if the TO decides to take on one of the colliding paths. Also, the histograms show high values for the TTC that leave plenty of time for changing the trajectory

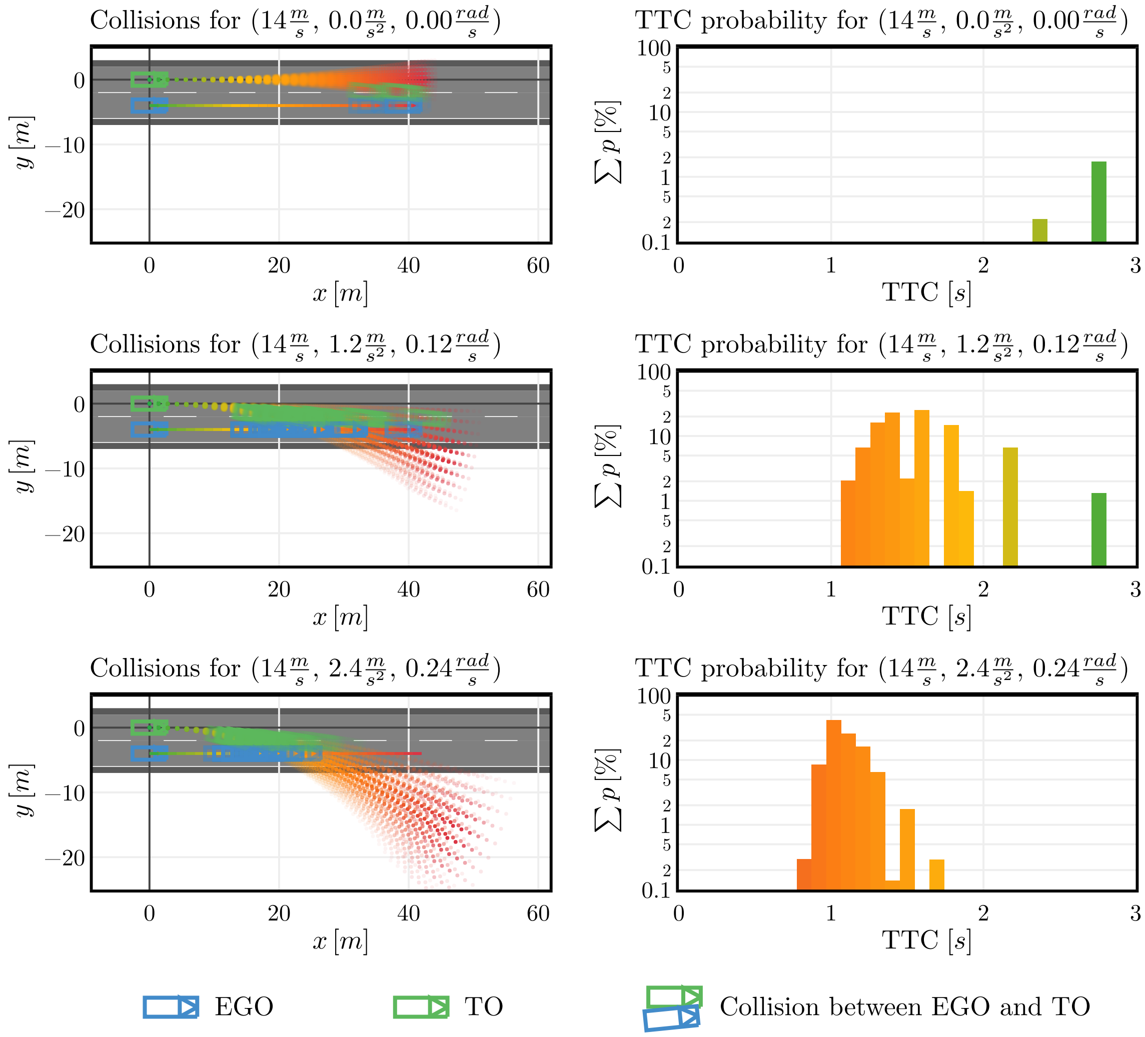


Figure 4.9.: Collision with an EGO vehicle driving parallel to the TO for the three tuples of (v_0, a_0, ω_0). One scene shows in each row with the scene's overview plot on the left and a weighted histogram of the collision probabilities over the TTC on the right.

plan. Hence, scene one can be considered safe. In scene two, the collision probability with respect to the path predictions rises to $\sim 99\%$. Also, the TTCs reveal lower values. A change of the planned path is recommended to avoid a possible accident. Lastly, in scene three, all predicted paths reveal a collision, and therefore the probability is 100%. Additionally, the TTC's distribution reaches values lower than $1s$. This scene can be considered critical.

In conclusion, for a sufficient statement about risk, colliding paths' cumulative probability is not enough. One has to take into account the time left until the incident. To further improve that, the following section considers the possibilities of the EGO to avoid the collision.

```
    Input: predictions of all $TO_k$ TOs
    Input: list of collisions between TOs and the EGO vehicle
    Output: same list of collisions with appended TTR
 1  Function ttr_calculation(collisions) : collisions is
 2      for $n_{col} \leftarrow 1$ to $N_{col}$ do
 3          for $t \in [0.0, T_{step}, \ldots, collisions[n_{col}].get(t)]$ do
 4              tts_left_path = predict_tts_left();
 5              if not path_collides(tts_left_path, $TO_k$) then
 6                  collisions[$n_{col}$].set(TTR) = $t$;
 7                  break;
 8              tts_right_path = predict_tts_right();
 9              if not path_collides(tts_right_path, $TO_k$) then
10                  collisions[$n_{col}$].set(TTR) = $t$;
11                  break;
12              ttb_path = predict_ttb();
13              if not path_collides(ttb_path, $TO_k$) then
14                  collisions[$n_{col}$].set(TTR) = $t$;
15                  break;
16              ttk_path = predict_ttk();
17              if not path_collides(ttk_path, $TO_k$) then
18                  collisions[$n_{col}$].set(TTR) = $t$;
19                  break;
20  return collisions;
```

Algorithm 4.2: Pseudocode definition of the TTR calculation algorithm.

4.1.2.2. TTR Calculation

In contrast to the TTC, the TTR reveals the time left until a collision is avoidable. This relates more to risk, as there can be scenes with the same TTC that are distinctly hard to avoid and therefore differ in their criticality. For instance, a full-frontal collision happening in the next second is harder to avoid than a rear-end collision with the same TTC and much more severe. Hence, the TTR functionally described in Section 3.2.3.1 is applied to the previously revealed collisions.

Algorithm 4.2 outlines the realization of the functional principle. For every detected collision, an iteration over all time steps until the collision follows in lines 2 and 3. Herein, paths for full

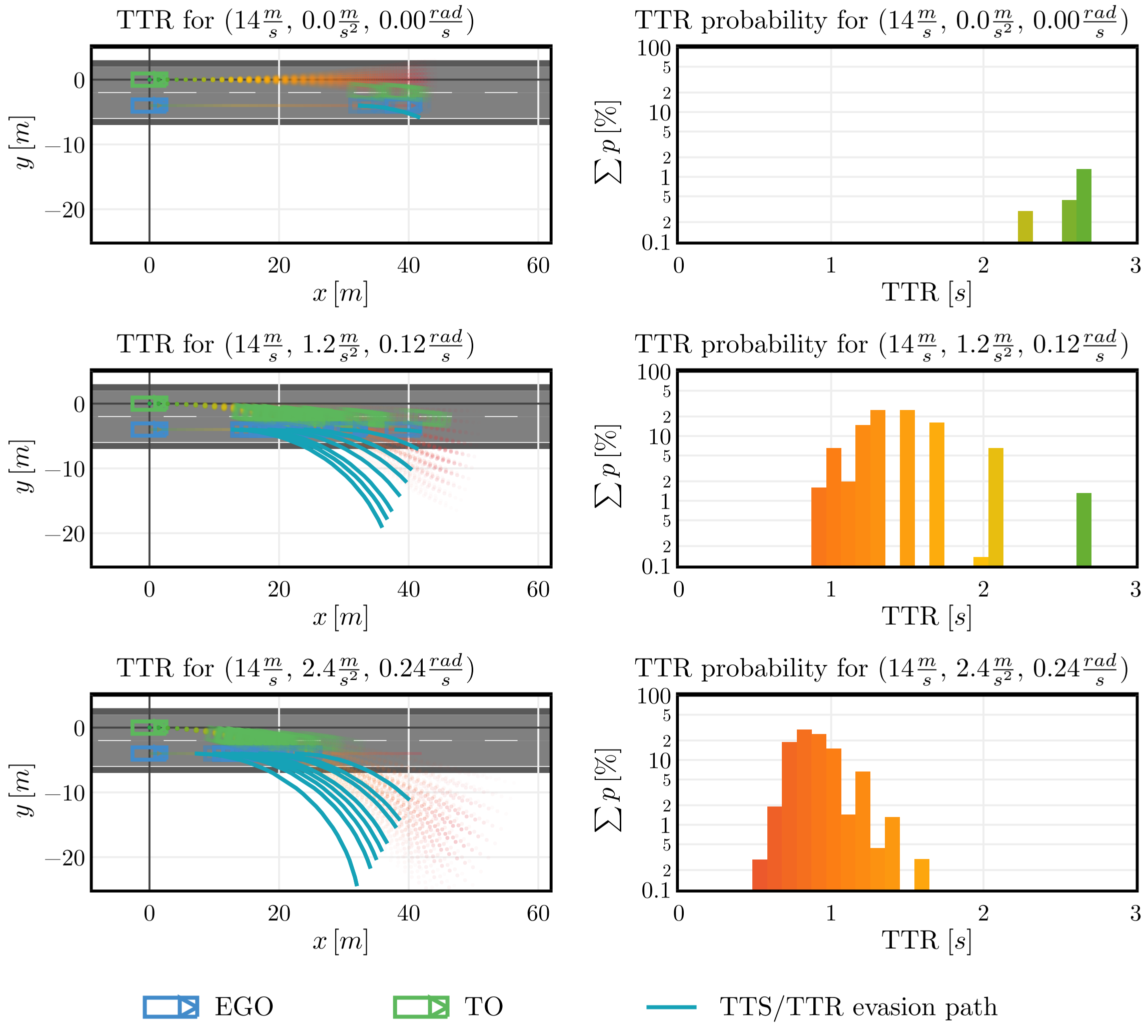


Figure 4.10.: TTRs for collisions with an EGO vehicle driving parallel to the TO for the three tuples of (v_0, a_0, ω_0). One scene is shown in each row with the scene's overview plot on the left and a weighted histogram of the collision probabilities over the TTC on the right.

left steering, full right steering, full braking, and full acceleration are predicted in that order in lines 4, 8, 12, and 16, respectively. In between, these paths are checked for the avoidance of the collision with the TO in function *path_collides()*. This works similar to *collision_detection()* from Algorithm 4.1 and is hence not further discussed. The first evasion strategy not colliding anymore sets the current time step as TTR and breaks the current time step iteration. Later throughout the time, when no strategy provides avoidance of the collision anymore, the TTR remains as the last found possible evasion in the timeline.

Again the three sample scenes are considered and shown in Figure 4.10 the same way as in Figure 4.9. In the scene overview plots, the predictions fade out for clarity, and the calculated evasion paths for the last possible evasion maneuvers are plotted. The right histograms show the TTRs' distribution over time with the probabilities on the vertical axis this time. In the selected scene, evasion by steering to the right reveals the most leftover time for reaction. Hence, only TTS paths are plotted in cyan in the scenes. Certainly, full breaking can also avoid the collisions but must happen earlier, so steering is chosen. The paths leave the road by far, but first, not the whole path is relevant for evasion, and second, the road is only for size comparisons. The chosen scene might not be realistic for that road, as previously stated. If the road were incorporated, the collision checking on the evasion paths would need to incorporate road boundaries as well and limit the right steering evasion maneuvers. The histograms show that TTR reveals lower values than TTC. The distributions moved to the left and changed slightly in their appearance because the same TTCs can have different TTRs depending on the scene. The shortest revealed reaction time is now $0.8s$.

The applied TTR is a measure that relates to a scene's riskiness but is more critical the lower it is. Hence, a transformation into a measure directly proportional to risk follows next.

4.1.3. Weighting and Single Traffic Object Risk

So far, a list of collisions, including their probability of occurrence and their TTR, is available. The scene's risk might be visible through a more in-depth inspection of the available data, but for convincing and automated testing, a measure directly related to risk based on this data follows.

4.1.3.1. Weighting

As already mentioned, the TTR is indirectly proportional to risk. Lower TTRs entail higher risk. Hence, a weighted transformation is applied. Further, leaving time to react to a particular incident does not necessarily mean the vehicle is able to perform this reaction. A certain time is necessary for a vehicle to observe the situation, plan a reaction, and conducted the movement, which is called the point of no return T_{PNR}.

$$\begin{aligned} g(\text{TTR}) &= \min\left(1, \max\left(0, \bar{g}(\text{TTR})\right)\right) \\ \bar{g}(\text{TTR}) &= \begin{cases} \frac{e^{-m(\text{TTR}-T_{\text{PNR}})}-e^{-m(T_{\max}-T_{\text{PNR}})}}{1-e^{-m(T_{\max}-T_{\text{PNR}})}} & m > 0 \\ \frac{T_{\max}-\text{TTR}}{T_{\max}-T_{\text{PNR}}} & m = 0 \end{cases} \\ T_{\text{PNR}} &< T_{\max}, \quad m \geq 0, \end{aligned} \tag{4.6}$$

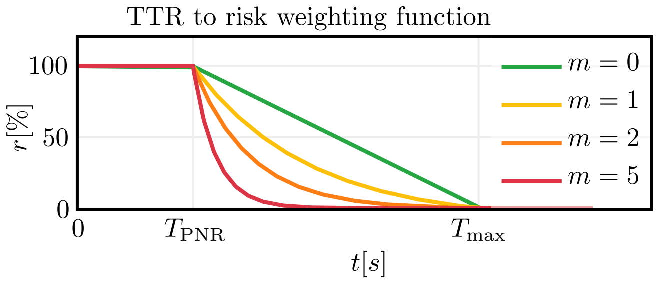


Figure 4.11.: The TTR weighting function from Equation (4.6) plotted for steepness parameter $m = \{0, 1, 2, 5\}$.

A function incorporating these aspects is presented in Equation (4.6) and plotted in Figure 4.11. A monotonic decreasing exponential function is used on the raw TTR to make the measure directly proportional to risk. The parameter m controls the monotonic descent's steepness, as shown in the center part of Figure 4.11. For $m = 0$, a linear decline is visible. The T_{PNR} is also incorporated through the exponential function's displacement and clipping to 100% before this point. Lastly, $T_{\max}$ is the time after a TTR is not considered risky anymore as it is too far in the future and the prediction is not reliable anymore. Therefore. the weighting clips to 0 afterwards.

The presented function maps the TTR to a risk related value between 0% and 100%. There are three parameters for the choice to tune the function. T_{PNR} is system dependent and cannot be chosen freely. The steepness m is a matter of choice and can be freely set greater than zero from a linear descent to a hard cutoff at T_{PNR}. Lastly, $T_{\max}$ should be specified to weight long TTR risk-less and consider the validity of the prediction. The latter is scene dependent. For example, the used simple prediction is valid for longer periods on less dynamic highway drives in contrast to highly dynamic inner-city crossings.

Still, having a larger amount of predictions per TO available, including their risks, is not an adequate measure. Therefore, the next step consolidated the information into a risk value arising from a single TO, and finally from the whole scene.

4.1.3.2. Single TO Risk

The consolidation of the gathered data to the risk imposed from a single TO is a stochastic problem. Given $g\left(\text{TTR}_{k,v_i{}^{(k)}}\right)$ as the risk of a single trajectory $v_i^{(k)}$ from the kth TO calculated through Equation (4.6) and $p_{k,v_i^{(k)}}$ the trajectories probability from Equation (4.5), the wanted risk is

$$r_{\mathrm{TO}_k} = \sum_{i=1}^{N_{\mathrm{smp}}^{(k)}} p_{k,v_i^{(k)}} \cdot g\left(\mathrm{TTR}_{k,v_i^{(k)}}\right). \tag{4.7}$$

Due to the normalized probabilities from $p_{k,v_i^{(k)}}$ and the constraint target domain in the weighting function from Equation (4.6), the risk value r_{TO_k} always unveils a value in the interval $[0,1]$ and hence represents a probability.

This risk value assesses only a single TO at a time. Further consolidation is necessary to reach the scene level risk.

4.1.4. Scene Risk Consolidation

It takes further considerations to calculate a risk value on scene level. The reason for this is that possible accidents with different TOs are not independent of each other. For example, an accident occurring with one TO earlier in the prediction timeline can block off accidents with other TOs later in time. This section states both dependent and independent consolidation and provides mathematical proof on why independence can be assumed in favor of computational feasibility.

4.1.4.1. Consolidation Considering TO Dependencies

In the case of dependency between TOs, one has to consider all permutations of the scene's evolution. The set of predicted trajectories $\mathcal{V}_k$ for all TOs spans this scene set $\mathcal{S}$ [167]

$$\mathcal{S} = \mathcal{V}_1 \times \mathcal{V}_2 \times \cdots \times \mathcal{V}_{N_{\mathrm{TO}}}, \tag{4.8}$$

and contains $|\mathcal{S}| = N_{\mathrm{smp}}^{N_{\mathrm{TO}}}$ elements under the assumption that each TO has the same number of predictions N_{smp} in its set V_k. A definition of the scene's risk r_{dep} considering the dependency is given by

$$\begin{aligned} r_{\mathrm{dep}} &= \sum_{s \in \mathcal{S}} p_s \cdot g\left(\mathrm{TTR}_{\mathrm{acc}}\right) \\ p_s &= \prod_{k \in \{1,\ldots,N_{\mathrm{TO}}\}} p_{k,v_s^{(k)}} \\ \mathrm{TTR}_{\mathrm{acc}} &= \min\left(\left\{\mathrm{TTR}_{k,v_s^{(k)}} \mid {v_s}^{(k)} \in \mathcal{V}_k \wedge k \in \{1,\ldots,N_{\mathrm{TO}}\}\right\}\right) \end{aligned} \tag{4.9}$$

with p_s being the current scene permutations probability and $\mathrm{TTR}_{\mathrm{acc}}$ the minimal TTR in this permutation.

Taking only the minimal TTR, hence the one imposing the highest risk, into account can be reasoned by the reflection of the differences between dependencies and independencies in the scene. First, there might be a TO path colliding in the independent case, but in the dependent case, another TO collided earlier with the EGO, and the collision point for the first is never reached. Hence, this collision is ruled out and becomes infinite. The same behavior implements by taking the minimum. Secondly, there also might be a TO that does not collide in the independent case, but in the dependent. For example, a collision with one TO would lock the EGO vehicle in the place of the accident. A second traffic object could then cross this place later in time and collide too. However, its TTR is undoubtedly higher than the first one, also revealing less risk. In terms of this risk assessment, it is irrelevant how many vehicles collide as accidents a KO criterion anyways. Therefore, the minimum function always chooses the riskiest TTR by default.

Although the consolidation can be mathematically described, and computational cost decreases by calculating the predictions and the TTR values independently, the computational demand of the consolidation increases quadratically with the number of vehicles in a scene. Notably, the complexity of Equation (4.9) is $\mathcal{O}\left(N_{\text{smp}}^{N_{\text{TO}}}\right)$ due to the permutation of the scene outcomes $\mathcal{S}$. Tests have shown that this becomes infeasible to calculate with three or more TOs in the scene.

4.1.4.2. Consolidation Assuming Independency of TOs

When ignoring the dependencies of accidents with TOs, individual risks from Equation (4.7) can be used to calculate the scene risk r_{ind} with

$$r_{\text{ind}} = 1 - \prod_{k=1}^{N_{\text{TO}}} \left(1 - r_{\text{TO}_k}\right). \tag{4.10}$$

The computational complexity of this equation is $\mathcal{O}\left(N_{\text{smp}} \cdot N_{\text{TO}}\right)$ and hence already with two TOs by far more efficient than the dependent consolidation.

The reasoning in Section 4.1.4.1 already states that ignoring dependencies includes accidents in the risk calculation, that are either not possible due to paths that cannot be reached anymore or not of interest of the risk assessment. This suggests the assumption that the risk calculated using the formula in this section reveals a greater risk than its counterpart considering the dependencies. Overestimation is favorable in risk assessment retrospecting the avoidance of false negatives as a requirement defined in Section 3.2.3. The suspicion of this inequality between the two consolidation methods is mathematically proven in the next section.

4.1.4.3. Proof that Assumption of Independence Overestimates Risk

The event of having an accident with the kth TO is now defined as in a specific scene outcome $s \in \mathcal{S}$ now defines as $\mathcal{A}_k^{(s)}$. Corresponding, the risk of the event is then

$$\mathcal{P}(\mathcal{A}_k^{(s)}) = g\left(\mathrm{TTR}_{k,v_s^{(k)}}\right). \tag{4.11}$$

Furthermore, the definition is the risk of a single trajectory incorporating its probability of appearance $r_{k,v_i^{(k)}}$.

$$r_{k,v_i^{(k)}} = p_{k,v_i^{(k)}} \cdot g\left(\mathrm{TTR}_{k,v_i^{(k)}}\right). \tag{4.12}$$

Using these definitions while inserting the risk of a single TO from Equation (4.7) into the independent risk consolidation from Equation (4.10), the following reshaping applies:

$$\begin{aligned}
r_{\mathrm{ind}} &= 1 - \prod_{k=1}^{N_{\mathrm{TO}}} \left(1 - \sum_{i=1}^{N_{\mathrm{smp}}^{(k)}} r_{k,v_i^{(k)}}\right) \\
&= \sum_{\substack{k\in\{1,\ldots,N_{\mathrm{TO}}\}\\ i\in\{1,\ldots,N_{\mathrm{smp}}\}}} r_{k,v_i^{(k)}} - \sum_{\substack{k_1\neq k_2\\ k_1,k_2\in\{1,\ldots,N_{\mathrm{TO}}\}\\ i_1,i_2\in\{1,\ldots,N_{\mathrm{smp}}\}}} r_{k_1,v_i^{(k_1)}} \cdot r_{k_2,v_i^{(k_2)}} + \cdots + (-1)^{N_{\mathrm{TO}}-1} \sum_{\substack{k_1\neq\ldots\neq k_{N_{\mathrm{TO}}}\\ k_1,\cdots,k_{N_{\mathrm{TO}}}\in\{1,\ldots,N_{\mathrm{TO}}\}\\ i_1,\cdots,i_{N_{\mathrm{TO}}}\in\{1,\ldots,N_{\mathrm{smp}}\}}} r_{k_1,v_i^{(k_1)}} \cdot \ldots \cdot r_{k_{N_{\mathrm{TO}}},v_{i_{N_{\mathrm{TO}}}}^{(k_{N_{\mathrm{TO}}})}} \\
&\overset{\substack{(4.9)\\(4.12)}}{=} \sum_{s\in\mathcal{S}} p_s \left[\sum_{k\in\{1,\ldots,N_{\mathrm{TO}}\}} g\left(\mathrm{TTR}_{k,v_s^{(k)}}\right) - \sum_{\substack{k_1\neq k_2\\ k_1,k_2\in\{1,\ldots,N_{\mathrm{TO}}\}}} g\left(\mathrm{TTR}_{k_1,v_s^{(k_1)}}\right) g\left(\mathrm{TTR}_{k_2,v_s^{(k_2)}}\right) + \ldots \right. \\
&\qquad \left. + (-1)^{N_{\mathrm{TO}}-1} \cdot \prod_{k=1}^{N_{\mathrm{TO}}} g\left(\mathrm{TTR}_{k,v_s^{(k)}}\right) \right] \\
&\overset{(4.11)}{=} \sum_{s\in\mathcal{S}} p_s \left[\sum_{k\in\{1,\ldots,N_{\mathrm{TO}}\}} \mathcal{P}(\mathcal{A}_k^{(s)}) - \sum_{\substack{k_1\neq k_2\\ k_1,k_2\in\{1,\ldots,N_{\mathrm{TO}}\}}} \mathcal{P}(\mathcal{A}_{k_1}^{(s)}) \cdot \mathcal{P}(\mathcal{A}_{k_2}^{(s)}) + \cdots + (-1)^{N_{\mathrm{TO}}-1} \cdot \prod_{k=1}^{N_{\mathrm{TO}}} \mathcal{P}(\mathcal{A}_k^{(s)}) \right] \\
&= \sum_{s\in\mathcal{S}} p_s \cdot \mathcal{P}_{\mathrm{ind}}(\mathcal{A}_1^{(s)} \cup \ldots \cup \mathcal{A}_{N_{\mathrm{TO}}}^{(s)}).
\end{aligned} \tag{4.13}$$

The independent risk now expresses as the sum over all scene permutations $\mathcal{S}$ of its permutation probability p_s multiplied by the risk of any of the TOs having an accident in the scene. Herein, the event is mathematically the union of all the single TO accident events $\mathcal{A}_k^{(s)}$.

The dependent risk consolidation from Equation (4.9) also reshapes using the above definitions as

$$r_{\text{dep}} = \sum_{s \in \mathcal{S}} p_s \cdot g(\text{TTR}_{\text{acc}}) = \sum_{s \in \mathcal{S}} p_s \cdot \mathcal{P}_{\text{dep}}(\mathcal{A}^{(s)}_{\text{acc}}), \tag{4.14}$$

with $\mathcal{A}_{\text{acc}^{(s)}}$ being the event of a unique accident in the scene, or in particular, the timely first one happing ignoring subsequent.

The two reshaped equations can now be compared. For that, the whole set of possible scene outcomes $\mathcal{S}$ divides into two types. The first one contains only permutations that contain no possible collisions. A member of this set denotes as $s' \in \mathcal{S}'$. These do neither contribute to dependent nor independent consolidation and therefore require no further consideration.

$$\mathcal{P}_{\text{dep}}(\mathcal{A}^{(s')}_{\text{acc}}) = \mathcal{P}_{\text{ind}}(\mathcal{A}^{(s')}_{1} \cup \ldots \cup \mathcal{A}^{(s')}_{N_{\text{TO}}}) = 0 \;\; \forall s' \in \mathcal{S}' \tag{4.15}$$

The second type is its complement and contains all permutations where at minimum one possible collision is present, and its members denote as $s'' \in \mathcal{S}''$. That is either at minimum one accident $\mathcal{A}^{(s)}_{k}$ in the independent case or the unique accident $\mathcal{A}_{\text{acc}^{(s)}}$ in the dependent case. Within that set, permutations containing only a single possible collision contribute equally to the risk, while those with multiple are handled differently. In the independent case, the contribution is that of the union of the two or more accident events, while in the independent case, the first appearing is dominant. Thus the unique accident can be seen as a subset of the union of the single accidents for the second type of permutations:

$$\mathcal{A}^{(s'')}_{\text{acc}} \subseteq \mathcal{A}^{(s'')}_{1} \cup \ldots \cup \mathcal{A}^{(s'')}_{N_{\text{TO}}} \;\; \forall s'' \in \mathcal{S}''. \tag{4.16}$$

Going one level higher and combining this information with the permutations from the first group, that do not appear as they do not contribute, the accident risk of a single permutation is described as inequality between the two consolidations methods:

$$\mathcal{P}_{\text{dep}}(\mathcal{A}^{(s)}_{\text{acc}}) \leq \mathcal{P}_{\text{ind}}(\mathcal{A}^{(s)}_{1} \cup \ldots \cup \mathcal{A}^{(s)}_{N_{\text{TO}}}) \;\; \forall s \in \mathcal{S}. \tag{4.17}$$

As both methods reshape as a sum over all permutations $\mathcal{S}$ multiplied with their respective probabilities p_s in Equation (4.13) and Equation (4.14), the final statement establishes:

$$r_{\text{dep}} \leq r_{\text{ind}} \tag{4.18}$$

Hence, independent risk consolidations always result in equal or higher risk values. Additionally, accidents classify as rare events [139]. Thus, in scenes with two or more TOs on a collision course with the EGO vehicle, are even more rare. Hence, the contribution of the overestimation made by assuming independency is minimal and infrequently present. This results in an overall slight overestimation that is computationally feasible with many TOs in the scene. It favors the task of risk assessment, as the result is always over or equal to the correct result avoiding false negatives while keeping the number of false positives due to greater overestimations at a minimum.

The methodology of the risk assessment metric is now fully explained. Finally, the risks of the three exemplary scenes accompanying this section can be evaluated. In the first scene, where both vehicles are driving next to each other with the same velocity, the calculated risk is $r \approx 0.01\%$. The few possible collisions are on paths with low probabilities and far enough in the future that they can be easily avoided. Likewise, by human inspection, such a scene frequently happens in traffic and bears almost no risk. In the next scene, the TO tries to cut into the right lane ignoring the EGO vehicle. Nearly all possible paths reveal a collision now, in which the reaction time reaches just below $1s$, and the risk rises to $r \approx 31.17\%$. Without any reaction, an accident occurs, but there is still enough time for avoidance. The measured result seems reasonable. Lastly, scene 3 shows the same cut-in but with the TO at its physical limits. Despite all paths now colliding, the difference results mostly from the shorter reaction times. There are now paths with a TTR is below the point of no return T_{PNR}. Nevertheless, most of them are above $1.6s$. Avoiding this scene with a risk of $r \approx 64.82\%$ requires either good driving skills or a performant reaction of the AD function.
The second part of this chapter covers the implementation as well as experimental results on scene and scenario level from both simulations and real test drives.

4.2. Implementation and Experimental Results

So far, the algorithm is explained alongside three exemplary scenes. For the assessment of AD, the measure's purpose is the evaluation of whole driving scenarios. Therefore, this section gives short insights into the algorithm's actual implementation and parameterization to fulfill this task, followed by representative scenario evaluations.

4.2.1. Implementation and Parameterization

The previously only in theory described algorithm is implemented in the Python programming language for evaluation. It follows, in general, the four steps from Figure 4.1 in every iteration over time. Thereby a timely sequence of the KPI is generated for a scenario. Internally, the

Parameter Name	Symbol	Value
Prediction Horizon	T_{pred}	$4.0s$
Prediction Strep	T_{step}	$0.1s$
Trajectories per TO	$N_{smp}^{(k)}$	100
Point of no Return	T_{PNR}	$0.5s$
Maximum Risk Time	$T_{\max}$	3.5
Weighting Slope	m	2.0

Table 4.1.: Parameterization of the accident risk calculation algorithm for the following scenarios.

algorithm offers options for parallelization by design. Specifically, the stochastic object prediction, collision detection, and TTR calculation are parallelized to increase efficiency. However, the current implementation is not real-time capable at a sufficient sampling frequency. The computation time necessary for a single scene iteration is strongly dependent on the number of TOs within the scene. As it is only used in post-processing for the assessment of AD, this characteristic is acceptable. In addition to the accident risk related KPI, additional measures calculate on the fly. The collision detection reveals TTC as the time up to a collision on a TO's most probable path and WTTC as its worst-case over the whole set of TO paths directly. Likewise, TTR and Worst-Time-to-React (WTTR) can be extracted from the third step of the algorithm[1]. Lastly, the THW implements by checking for collisions of the EGO path with the TO's initial position in the scene.

The parameterization used in the experiments summarizes in Table 4.1. Choosing prediction horizon T_{pred}, maximum risk time $T_{\max}$ and the weighting slope m offers to main tuning possibilities of the method. As the following examples are scenarios on a straight road such as a highway, the simple prediction model is valid for a longer period, and a rather long prediction horizon is set. Adapting this method for city scenarios with highly dynamic traffic flow requires a shorter horizon or better prediction models. Possible accidents far in the future are not risky in the current scene. Hence a rather steep weighting parameter m compensates for the prediction horizon and maximum risk time. In contrast to these, the remaining parameters can not be set freely. The prediction step should be small enough for the model to be predicted correctly in the integration and to provide a dense timeline of future TO positions. Significant timely and spatial distance between prediction steps could cause the measure to miss possible collisions. The number of predicted trajectories must be large enough to map the naturalistic TO behavior

[1] Note that this implementation reveals the worst case with respect to the naturalistic TO prediction of this measure. In contrast, [116] uses the physical boundaries of a vehicle.

onto the risk value and again fill the predictions' spatial resolution. Convergence experiments have shown that the risk measure safely converges for 100 predictions. Lastly, T_{PNR} is dependent on the AD function response time to incidents. For the following experiments, it is assumed to be $0.5s$.

With the implementation and parameterization completed, experiments in the three test domains simulation, NDS_1, and TD follow.

4.2.2. Experimental Results and Evaluation

In this section, the presented KPI evaluates with three scenarios from different test domains. As there is no ground truth for such a measure, a complete validation is not possible. Parts of the method can be reasoned or mathematically proven, as done in Section 4.1. However, the risk measure's outcome can only be evaluated against cornerstones of the expectations on such a measure and compared to existing KPIs. First, a simulated collision scenario exhibits the measure at full risk. Every day driving from NDS_1 is used to show the method's capabilities of measuring low risk. Lastly, a critical planned TD experiment shows the behavior in critical scenarios without an accident.

4.2.2.1. Experiment on a Simulated Rear-End Collision

The first experiment analyzes a rear-end collision. To put no human driver at risk, it is simulated. In the initial scene, as depicted on the first road of the upper plot in Figure 4.12, a TO drives $\sim 52m$ ahead on the EGO vehicle. Both are driving at the constant velocities $10\frac{m}{s}$ and $21\frac{m}{s}$, respectively. Due to the negative relative velocity between the two vehicles, a rear-end collision will happen, which the EGO vehicle's driving function is designed to ignore.

While two additional scenes locate in the upper plot of Figure 4.12, the whole risk time-series alongside the common KPIs are shown in the lower half. In scene (2), the distance between both vehicles is already reduced to $\sim 30m$, leaving a TTC of $2.7s$. The risk value for this scene is $r \approx 4\%$, indicating that risk is not too high at the moment, but action might be necessary shortly. Further, in scene (3), when the TTC decreases to $1.1s$, the risk value is already at $r \approx 100\%$. For any predicted trajectory, the TTR falls below the vehicle's necessary reaction time T_{PNR}, and the collision is unavoidable anymore. This is only apparent from the presented accident risk KPI. For example, the THW of $0.6s$ would only be fined with 25€, according to the German StVO [118].

In this scenario, the presented KPI shows a plausible mapping of accident risk. It converges exponentially towards 100% with less time for collision avoidance. In contrast to the other KPIs,

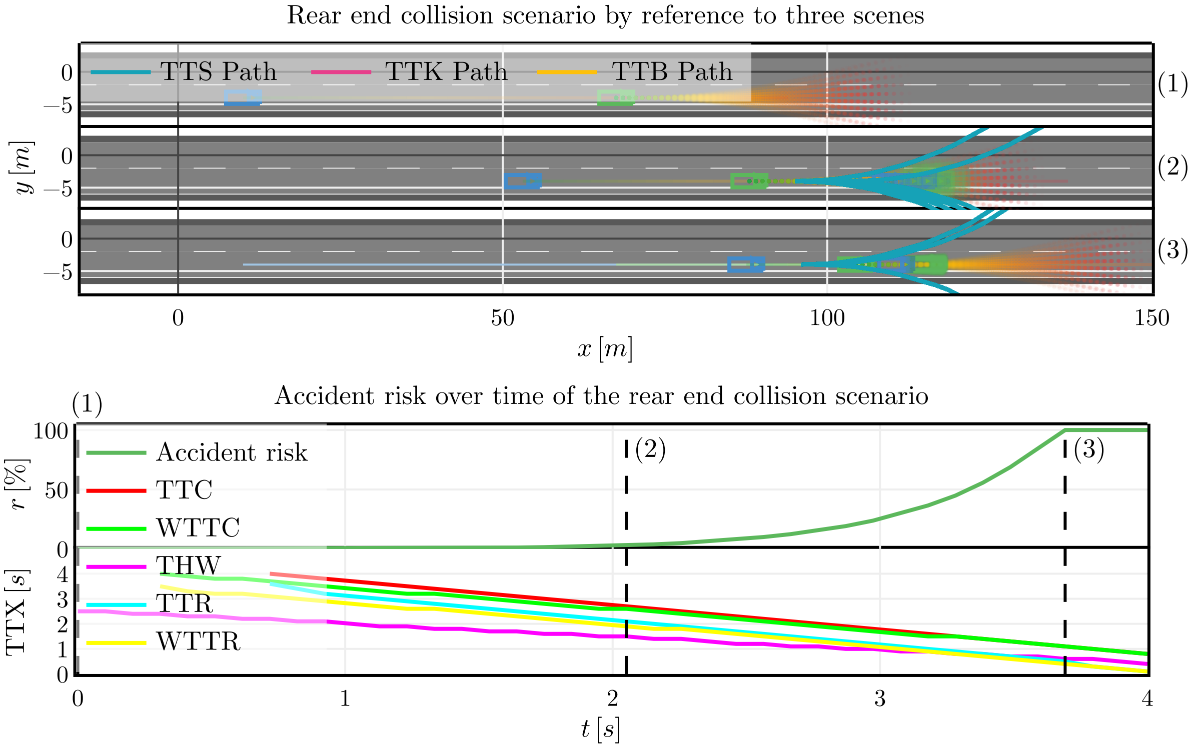


Figure 4.12.: Risk of a simulated rear-end collision. The upper plot shows three scenes of the scenario, each on a separate road. The risk over time is shown alongside common KPIs in the lower half with dashed black lines indicating the location of the scenes.

the level of accident risk is directly observable from the measure. However, for this scenario, the stochastic prediction of multiple traffic objects does not contribute to the benefits enough to justify the increased computational demand. Hence, experiments on other scenarios follow.

4.2.2.2. Experiments on Lane Change Scenarios from NDS$_1$

The next scenario emerges from the NDS$_1$ dataset *highD* [97]. This dataset contains naturalistic traffic flow on german highways. In order to obtain scenarios with cause and effect of some traffic incident according to the definition in Section 3.2.2, it is parsed for time periods where at minimum one lane change occurs. This extraction is part of another scope of this work, the scenario space definition, and can be found in Section 6.2.1. Within the extracted scenarios, one vehicle is set as EGO vehicle from whose perspective risk is measured, and the others are TOs. Figure 4.13 shows an exemplary scenario with a close overtaking maneuver involving two TOs from this NDS$_1$. Thereby, the consolidation of risk from multiple vehicles comes into use. In the first scene, one TO drives behind the EGO while another is in the process of overtaking it. Also, the first intends to overtake the EGO but must wait for the second to free the left lane.

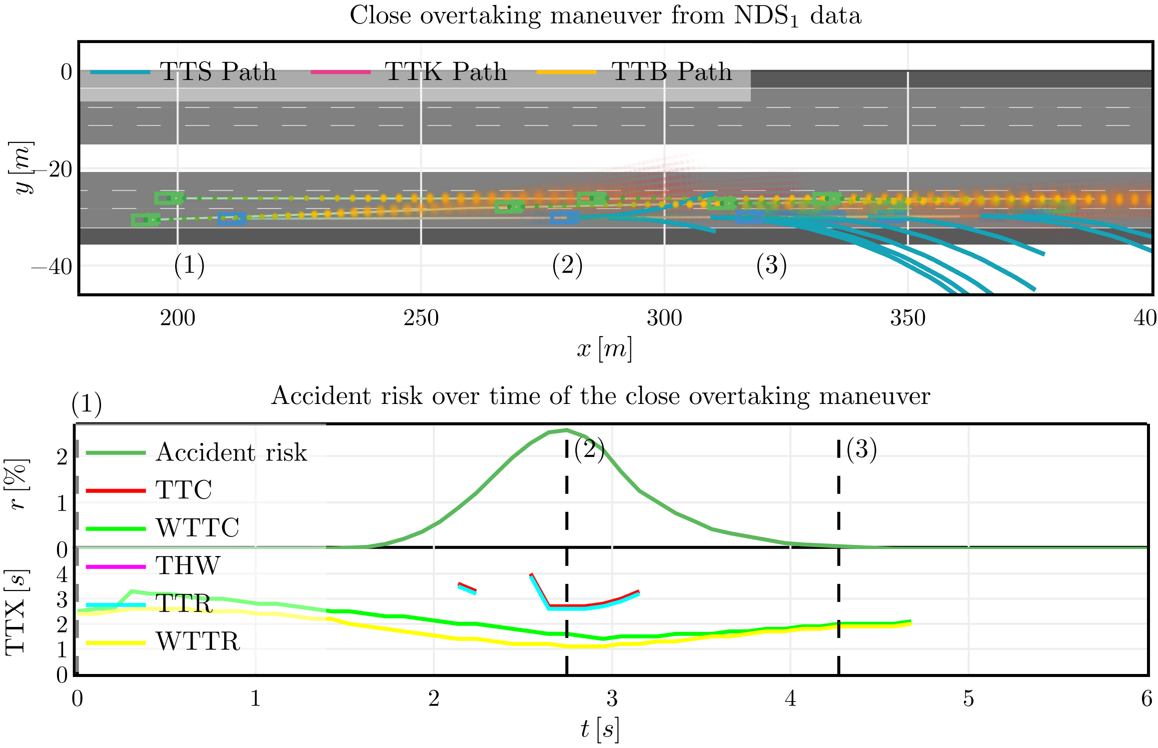


Figure 4.13.: Risk of a close overtaking scenario from the highD NDS$_1$. The upper plot shows three scenes of the scenario on the same road. The risk over time is shown alongside common KPIs in the lower half with dashed black lines indicating the location of the scenes.

This is a common highway situation that usually results in closer overtaking by the following vehicle than with a free left lane. Also, here the overtaking in scene (2) happens with less than a vehicle's length for the distance, resulting in $r \approx 2.6\%$ risk for the EGO. In scene (3), the overtaking's closeness is more apparent, but an accident is not probable anymore, and the risk vanishes.

Comparing these results to the common KPIs emphasizes the benefits of the stochastic measure. As no TO is ever in the EGO vehicle's pathway, there is no value for the THW at all. Also, TTC and TTR appear only sparsely as for the first TO, the prediction of the main path assumes a cutting into the right lane again after overtaking. Only their worst-case estimations from the stochastic prediction are available over a longer period. However, the presented accident risk measure provides data over the whole time and a smooth progression throughout the scenario. Also, its result seems reasonable. Such a close overtaking is common on highways and may be risky but does not result in an accident frequently.

One receives an impression from the everyday risk obtained through the measure by processing

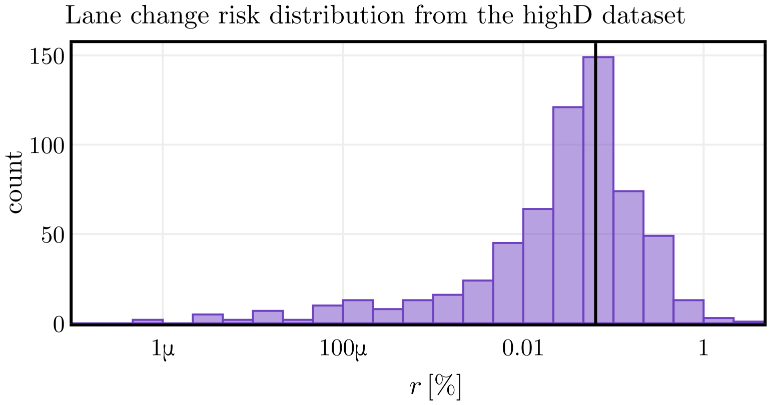


Figure 4.14.: Risk distribution from 1000 lane change scenarios in the highD NDS$_1$. 379 scenarios are not included as their risk of 0.0% can not be displayed on a logarithmic scale. The mean of the distribution is indicates a black line.

a more significant amount of such scenarios. Therefore, the risk is calculated for 1000 scenarios that contain at minimum one lane change. The result presents in the histogram from Figure 4.14. Due to the nature of the measure, multiplying probabilities with an exponential descending weighting function, its output usually diverges in magnitudes. Hence, a logarithmic scale is chosen for risk. As a consequence, 379 scenarios that reveal $r \approx 0.0\%$ risk are not shown in the histogram. It is observable that the scenario chosen above is from the upper bound of riskiness. The remainder of the lane change scenarios resembles around $r \approx 0.063\%$ risk, indicated with a black line. These results are feasible, as everyday driving does not put the driver at exceedingly high risk. Accidents remain rare events, which is confirmed by this distribution.

4.2.2.3. Experiment on a Prepared Critical TD

Lastly, this TD resembles an actual test conducted with an active AD function on a closed test track with a planned scenario. This test drive is from the set of measurements generated for a later part of this work, found in Section 5.2.1. The scenario from this measurement displays in Figure 4.15. The EGO vehicle drives at initially $\sim 24\frac{m}{s}$ on the right lane while being overtaken by a faster TO on its left in scene (1). The TO driven by a trained test driver cuts into the EGO driving lane and brakes strongly at $\sim -4\frac{m}{s^2}$ deceleration rate, imposing risk to the EGO. As the AD function is free to react to this incident in this test, its performance is directly assessable.
In scene (2), the TO is in the process of cutting into the driving lane of the EGO vehicle. Despite the spatial closeness of this maneuver observable from the scenario plot in the upper half of Figure 4.15, the relative velocity causing the two vehicles to gain distance from each other, which makes a collision physically impossible. This is correctly recognized by all shown KPIs in the lower half of the figure. Solely at the hard braking maneuver, the EGO vehicle is

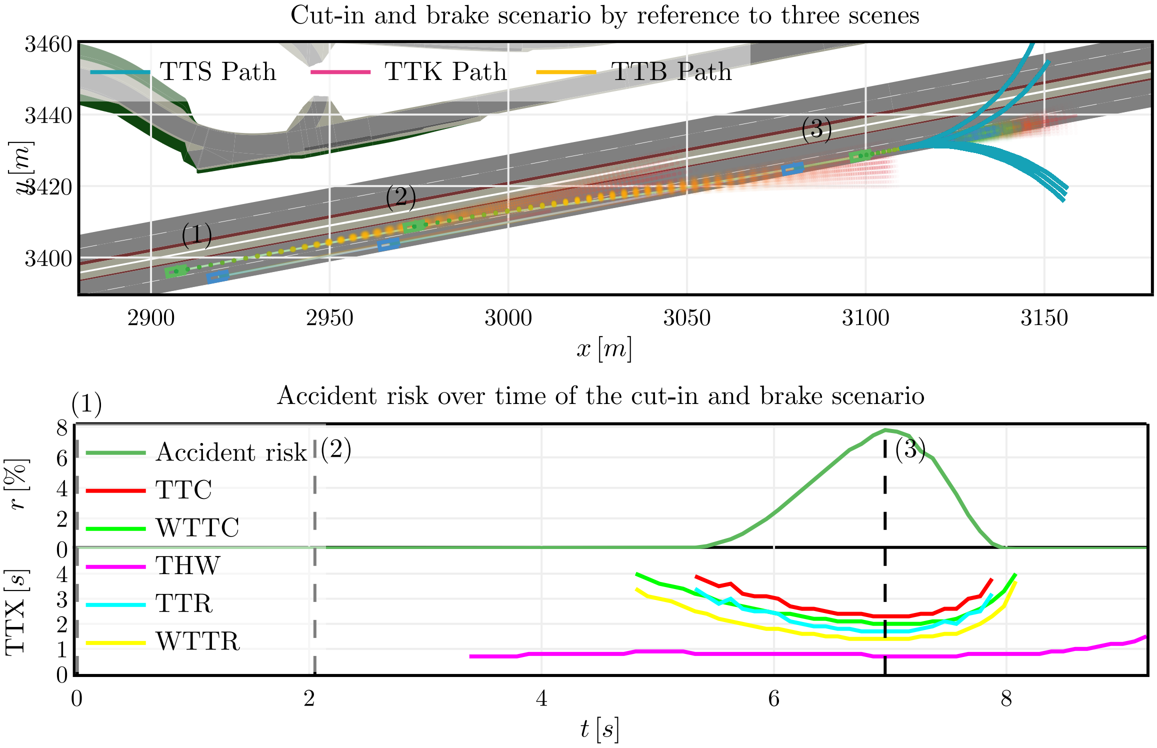


Figure 4.15.: Risk of a critical cut-in and brake scenario from a TD. The upper plot shows three scenes of the scenario on the same road. The risk over time is shown alongside common KPIs in the lower half with dashed black lines indicating the location of the scenes.

exposed to an accident risk peaking in scene (3). The AD function is already applying strong braking to avoid the recognized accident. As it keeps the risk at rather low $r \approx 7.8\%$ in this scene, its reaction can be considered successful.

Although this scenario is riskier than the highway overtaking from the previous section as an EGO action is required to avoid an accident, the common KPIs remain in the same magnitudes. For example, the TTC is $2.3s$ in comparison to $2.7s$ in the previous scenario. Again, the presented accident risk measure is the only one available over the whole period of time and provides smooth and reasonable results.

In this experiment, it becomes obvious that the simple TO model used for the prediction can result in future trajectories leaving the road in the long run. This is compensated by the vanishing weighting for longer prediction times but raises the question of whether road-dependent models might benefit the measure, especially for higher dynamic scenes in urban environments with more significant orientation changes.

4.3. Conclusion

At the beginning of this chapter, the necessity of evaluation metrics for the assessment of AD motivates this research. For safety-critical systems, a measure for the risk of human harm and material damage is the most important and leads to the accident risk measure presented in this chapter. After a recap of the requirements, the measure with its four steps prediction, collision detection, weighting, and consolidation is explained in detail.

In contrast to common KPIs from literature, this risk-based measure directly reveals values between 0 and 100% resembling the threat level of a driving scene instead of time-based values which require interpretation. Additionally, it always reveals a value regardless of the number of TOs in a scene and their constellations. The incorporation of naturalistic traffic object behavior ensures a realistic assessment of the accident risk. Additionally, it is mathematically proven that it overestimates the risk slightly to avoid malicious false-negatives in the assessment. Following experiments on both simulated and real-world drives from NDS_1 and TD show comprehensible results for evaluating the measure against expectations, as validation is not possible due to the absence of a ground truth reference.

Despite the reasoning that simple TO models for the short-term prediction suffice the measure, more complex models might be beneficial, especially for a more dynamic urban scene. In particular, the incorporation of road topologies could improve predictions also in the short term but comes at the cost of extra information necessary for the assessment. In the presented implementation, only scenarios containing cars solely can be assessed. For use in production, additional effort must be made for modeling and parameterizing all kinds of traffic participants with their physical boundaries and naturalistic behavior. As this task is not scientifically relevant and necessary for the proof of concept delivered in this chapter, it is omitted.

With the presented risk assessment measure, a driving test in the assessment can be evaluated. Hence, the first hypothesis, *The performance and safety performance of an AD functions' behavior can be measured.*, is confirmed. Indeed, this is only one of several necessary build blocks of a successful assessment. The next chapter presents the overall assessment methodology incorporating this KPI.

5. Testing Methodology, Framework and its Impact on Test Results

A measure for the performance of AD in a driving test alone is not enough for its assessment for market release. This task compromises a whole concept. When, where, and how many tests need to be conducted to prove such systems' safety is the question. The state of the art from Section 3.3.2 qualifies the V-Model as de facto development and testing methodology used previously to AD to assess and release ADAS. However, also the problems arising with its application to AD are addressed in this section. Regarding the limited ODD of ADAS, it is feasible to define and test the necessary amount of real-world scenarios [20]. In its highest level of automation, AD is planned to take control in any kind of scenario possible on public roads [53]. This inevitably results in a vase amount of testing [20, 45] that is neither feasible to define on the requirements side nor feasible to conduct on the test side. Additionally, the V-Model is limited in terms of agility during development when testing modules against a vast amount of requirements.

Section 3.3.3 discusses existing approaches to overcome those limitations and provide whole new solution approaches. Despite all efforts, a single de facto assessment standard has not been agreed on yet. The literature agrees that the use of virtual test domains both in the early stages of the assessment as well as for market release testing seems unavoidable to cope with the massive amount of testing required. Their integration into the testing concept promises an acceleration of both development and assessment. However, virtualization is always accompanied by limitations of the test results validity [99]. Especially for safety-critical systems, validity of test results is crucial. This motivates the second research question of this thesis: *How can efficient virtual testing be verified and included in a scenario-based assessment process, and what is the impact of virtual testing?* This chapter's findings are mainly funded by the findings of the publications [Journal1, Journal3][Paper2].

The rest of this chapter is structured as follows. The theory behind the suggested methodology elaborates in Section 5.1, together with the requirements on its modules. These modules are then described in Section 5.2 alongside individual evaluation against a critical corner case scenario.

Section 5.3 summarizes the findings of the whole pipeline's implementation, the results of a cross-verification between simulation and real-world tests, and the derivation and discussion of the lessons learned. Finally, Section 5.4 provides the chapter's conclusion.

5.1. Scenario-Based Assessment Methodology

Rethinking the approaches to asses AD from Section 3.3.3, this thesis focuses on scenario-based approaches for the following reasons:

1. The use of scenarios as requirements does not decrease the number of requirements itself but makes the generation of those automatable. Scenarios can be derived from NDS_1, for example. This results in a single significant requirement: AD needs to navigate through all scenarios safely.

2. The collection of scenarios can be shared among OEMs and used for centralized approval by legislative institutions [49].

3. If scenarios define the test space, a statement about the coverage with respect to the traffic space can be made (this is addressed by the third research question in Chapter 6).

The methodology presented in the following founds on the theoretical thoughts of two notable works from the literature. First, the PEGASUS approach initiates the idea of a centralized scenario database that iteratively fills with scenarios generated by the OEMs. Consequently, a dense representation of the traffic space develops over time and can be used to test the AD function against. However, the domain of testing and the concurrent validity of the test results remains unclear. Furthermore, the comparability of scenarios extracted from different sensor sources of different OEMs requires analysis of their accuracy. Second, the approach is extended by [47]. Herein, scenarios from real-world domains such as FOT, NDS_1, or TD form the basis of the traffic space. Virtual domains are then used to reprocess the available scenarios, and their results are verified against the available ground truth locally for this exact scenario. A locally verified virtual test domain can then enable the extension of the traffic space through local variation. More precise explanations of this concept, a description of the individual steps, and evaluation are still missing.

At that point, this thesis extends the state of the art with respect to the assessment methodology. The whole method is introduced in detail in the following. Subsequently, the requirements on the methods' individual steps are discussed before actual implementations and evaluations of the concept follow.

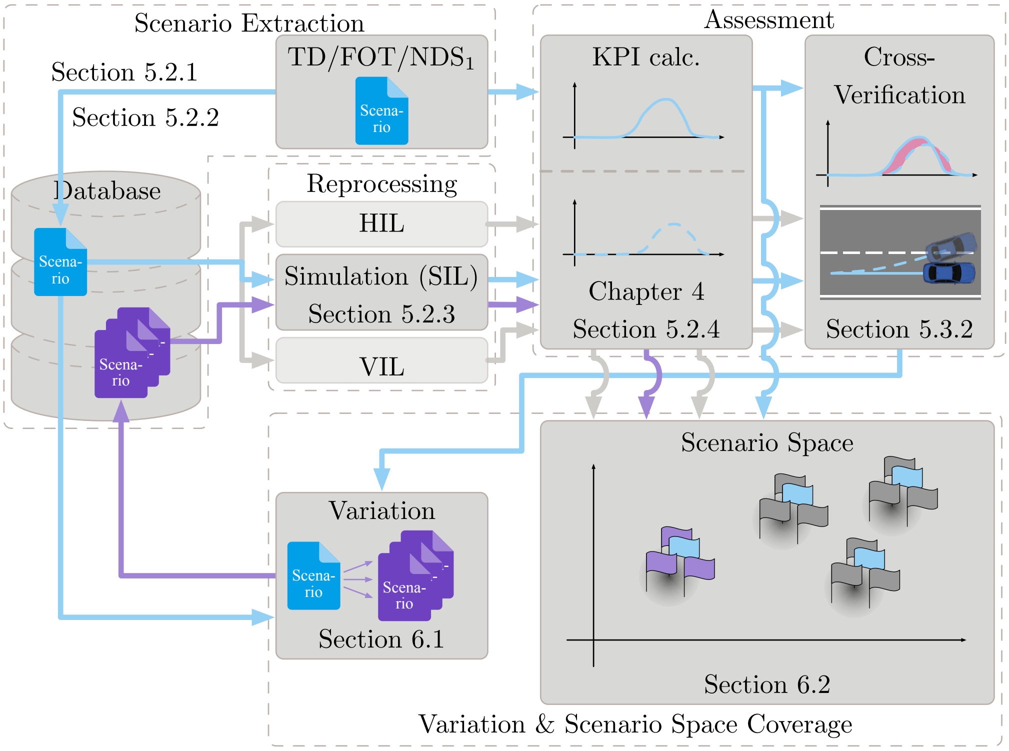


Figure 5.1.: The architecture of the assessment methodology divided into four parts. The relevant sections describing the components in detail are given implicitly.

5.1.1. Architecture of the Assessment Framework

The methodology presented in this work outlines in Figure 5.1 [Journal3] and divides into four major parts. Similar to previous approaches, scenarios from real-world test drives such as TD, FOT, and NDS$_1$ to fill a database in the first step. The acquisition of driving data in the TD domain is explained in Section 5.2.1. It is important that the EGO vehicle for recording the scenarios runs with the AD function under test. The reason for this becomes apparent later. Section 5.2.2 provides a specialized scenario description format and the conversion into it. In the following reprocessing step, the scenarios are recreated in virtual test domains as accurately as possible. Therein, static scenario, and the TOs are fixed for re-simulation. The EGO vehicle facilitates the exact same AD function as being used in the real-world test out of which the scenario was generated and is free to perform. The proof of concept explained within this thesis chooses simulation as it can be considered the opposite of real-world tests through pure virtualization of all components. Simulation and its results are explained in more detail in Section 5.2.3. Since the same AD function is used both in the TD and the simulation, two comparable test results oppose each other. Both are assessed in the third part with the KPI calculation from Chapter 4. Section 5.2.4 compares the results. The following cross-verification assesses the virtual test domain's accuracy by comparing both the results on a KPI basis and the tests' raw data. Further detail follows in Section 5.3.2. Based on the cross verifications results, the following assumption can be made:

> *"If a virtual test domain can reprocess a real-world experiment's result accurately enough, it is able to replace the real-world test for this exact experiment. It is further assumed that this assertion holds for a close and continuous range around said experiment."* [p. 4 Journal3]

This justifies the following variation of scenarios solely tested in the virtual domain and contributes to filling the scenario space and, hence, the required test volume. Additionally, they receive verification to some degree depending on the cross-verification results and the variations' distance to the original scenario.

This chapter dedicates to the framework from the recording of real-world measurements to the assessment, hence incorporating the reprocessing of real-world test drives with all sub-steps and the received precision in detail. The question of how the actual scenario space can be defined and how the variation can build up upon it is the subject of Chapter 6. Hence, the discussion of the method's main components, namely measurement generation, scenario description and extraction, simulation, and assessment, follow.

5.1.2. Components and their Requirements

Before starting the details about all components, this sub-section dedicates to the collection of all applicable requirements.

First, the scenario description and the extraction from the available data define a key component to the method. The following reprocessing accuracy is strongly dependent on the precision and usability of the scenario description as input. Therefore, three requirements are given [Journal1]. A scenario description must be unambiguous to define a single fixed point in the scenario space. There must be no degree of freedom or room for interpretation that leaves the simulation underdetermined. Thereby it is guaranteed that the simulation's error can attribute to the test domain, and the cross-verification applies to it. A description should also not be overdetermined. It is best practice to include only so much information that there are no redundancies. For example, velocity calculates from the change in positions. Hence, the inclusion of both into the description is redundant. If redundancies must be included for some reason, they need to be consistent. Lastly, the description should be as accurate as possible, not to affect the AD function's behavior.

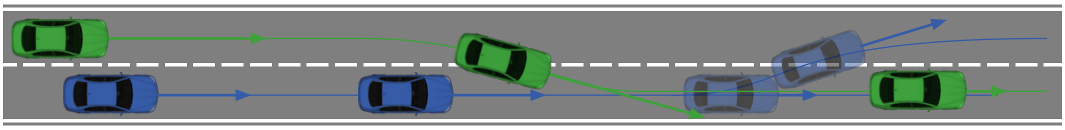

Scene (1): Pass-by
$v_{\mathrm{TO}} > v_{\mathrm{EGO}}$
Scene (2): Cut-in
$v_{\mathrm{TO}} > v_{\mathrm{EGO}}$
Scene (3): Brake
$v_{\mathrm{TO}} \leq v_{\mathrm{EGO}}$

Figure 5.2.: Description of the corner case scenario to challenge the assessment framework. A TO overtakes the EGO vehicle cutting into its front and brakes in the further course of this scenario

Next, the simulation must use the same AD function that is used for generating real-world measures. Solely through that, a cross-verification is meaningful. Then again, also the simulation must be able to recreate the scenario given the description accurately. Otherwise, the results can not be used for cross-verification.

An assessment metric should be representative. This connotes that the desired information is directly observable from its value. Together with the avoidance of false-negatives, the KPI presented in Chapter 4 already fulfills the requirements. Hence, further proofing is not necessary anymore.

These are the fundamental requirements on the components of the method dealt with in this chapter. As the variation follows in Chapter 6, its requirements do not follow here. Now the actual realization of the modules can be discussed.

5.2. Modules of the Framework

This section explains the necessary modules for the reprocessing and assessment of real-world scenarios in the simulation detail. To illustrate each of the modules' functioning, a representative scenario accompanies the whole section. This scenario is intentionally chosen to challenge the framework and highlight current weak spots, which Section 5.3.2 addresses in the subsequent cross verification.

A challenging scenario for reprocessing locates close to a decision threshold of the AD function. Such a corner case scenario is presented in Figure 5.2. In scene (1), the EGO vehicle is overtaken by a TO with higher velocity on the lane to the left. After overtaking, the TO proceeds with a cut-in maneuver into the front of the EGO in scene (2). Lastly, it breaks strongly in scene (3). Thereby, the AD function is exposed to a certain risk and required to take action to avoid a collision. Depending on the situation's criticality, two actions are possible and pictured in scene (3). If the threat induced by the TO is too high, emergency breaking provides remedy at the cost of passenger comfort. This can be avoided if there is still enough space and time left

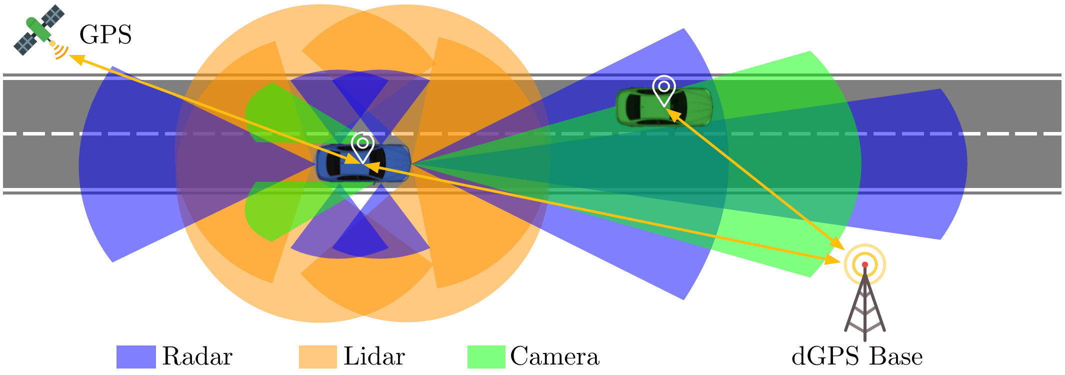


Figure 5.3.: Measurement setup of a prototype vehicle. Radars are shown as blue, Lidars as orange, and cameras as green cones. The extent of these only show the schematic arrangement of the sensors, their range and their field of view and are not true on scale. Further, the interaction of GPS and dGPS with the two vehicles is shown.

to overtake the TO safely, defining the second possible and more comfortable reaction to that situation. If one records such a scenario close to the explained decision threshold, the errors made while reprocessing can change the scenario's outcome.

5.2.1. Measurements

The first module required for the reprocessing and assessment frame is acquiring measurements from real-world scenario data. For that, a vehicle equipped with the AD function under test and a sensory setup is necessary. This setup from the used prototype vehicle shows in Figure 5.3[1]. It comes equipped with two types of sensors.

The first is sensors that also equip with the planned production vehicle. These collect environmental information under the same conditions as in a consumer car. A wide variety of sensors are key components to the AD system to provide a solid understanding of the vehicle's surroundings [79]. Among these are Radars, which are part of modern production vehicles for several years [168] and sense the surroundings by reflecting radio waves. Through the time of flight of the radio waves, the distance to the object is estimated. Different radars are equipped with the prototype vehicle, differ in their detection range and field of view, and depict as blue cones in Figure 5.3. Radars collect their information in point clouds associated with the measured attributes, which can further suffice for object detection and parameterization. They tend to be robust against environmental conditions but generally are either limited in resolution or their field of view [168]. Similar to a Radar is the functional principle of a Lidar, albeit it uses laser

[1] Satellite icon made by mavadee from www.flaticon.com. Antenna and location marker icons made by Freepik from www.flaticon.com.

rays instead of radio waves. Lidars generally allow for higher resolutions at larger fields of view [169] but are costly [170] and can be easily manipulated [171]. In the prototype vehicle, Lidars illuminate the midsize proximity densely as depicted as orange cones. The sensing of the environment is enlarged by cameras located at the front and the rearview mirrors to cover blind spots. They are cheap but subordinate to environmental conditions such as visibility and lighting. The detection of objects from these three sensor types can either be achieved on a lower level with each sensor's data individually and the fusion performed afterward [85] or by fusing the data first and performing the detection on the combined data on a higher level [84]. Besides recognizing the environment, the sensor information also suffices for the self-localization of the EGO vehicle. Supported by standard onboard sensors and GPS, also the EGO is parameterized. All the so far mentioned data is referenced as Sensor Data (SD) from now on.

As the raw sensor information and the object detection and parameterization are error-prone and can differ more or less from reality, the prototype vehicle also equips with special reference sensors that can not be used in production. The integrated *OxTS RT3000* combines a high precision Inertial Measurement Unit (IMU) with Differential GPS (dGPS) to parameterize the vehicle [172]. The specified errors are as low as $1cm$ for the position, $0.05\frac{km}{h}$ for velocity, and $0.1\frac{rad}{\pi}$ for the heading. Certainly, still errors are present, but as they are negligibly low in comparison to the information provided by onboard sensors, this data is considered as Ground Truth (GT). Further, the TO used in this TD also equips with this system to provide its GT wirelessly to the EGO. However, such a system is dependent on a base station with an exactly known position as pictured in Figure 5.3 and can hence not be used in production vehicles.

A measurement of the representative scenario from TD illustrates in Figure 5.4. The GT data depicts in blue for the EGO and green for the TO while the SD counterparts show in pink and red, respectively. For each, plot the lower half shows the error between SD and GT in the color of the respective SD. It extends with a ribbon indicating the σ-interval of the errors' distribution around its mean μ. In the top plot, the trace of the scenario superimposes to the proving ground road. The error calculates from the distance d between SD and GT. The lower left shows the velocities v and the lower right the headings θ.

First of all, it is observable that the EGO vehicle's self-measurement is by far more precise than the detection and parameterization of objects. This manifests in the indistinguishable pink and blue lines in all three plots and the narrow error margins around zero. The means and standard deviations of these errors are $\mu_{\text{EGO},d} = 0.226m$, $\sigma_{\text{EGO},d}$ $0.0513m$, $\mu_{\text{EGO},v} = 0.0363\frac{m}{s}$, $\sigma_{\text{EGO},v} = 0.0420\frac{m}{s}$, $\mu_{\text{EGO},\theta} = 2.24 \cdot 10^{-5}rad$, $\sigma_{\text{EGO},\theta} = 1.81 \cdot 10^{-4}rad$ for the respective physical

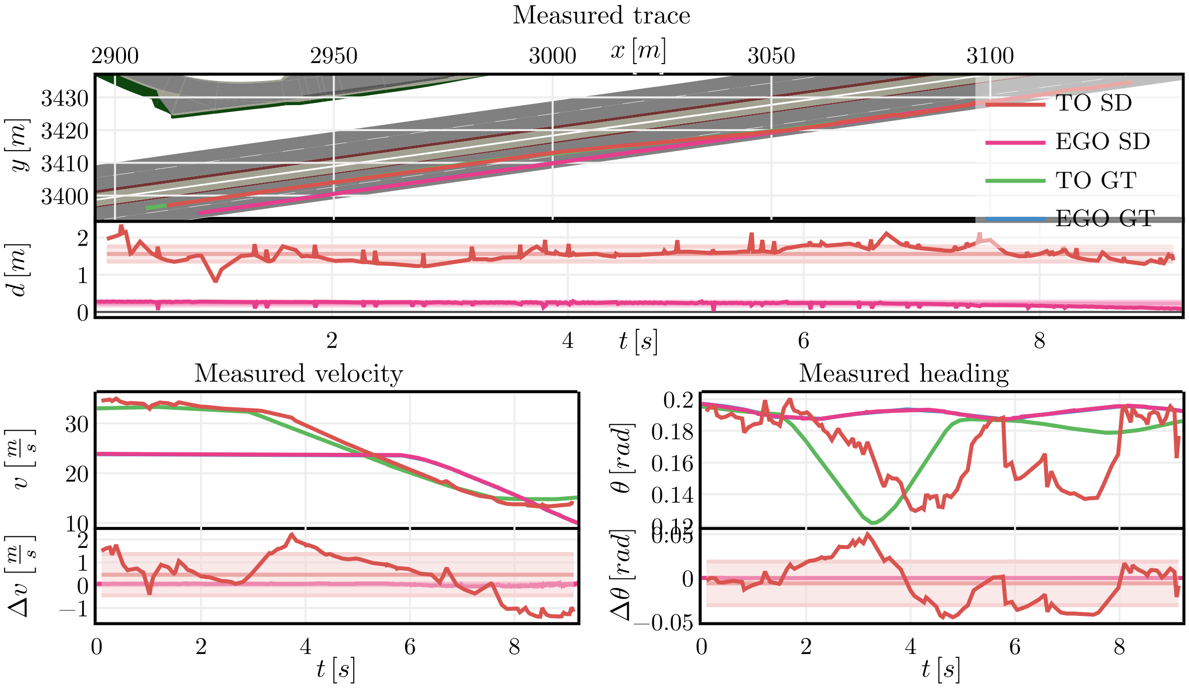


Figure 5.4.: Measurement of the corner-case scenario. The spatial traces are shown in the upper plot with the error between GT and SD plotted in its lower half. Bottom left and right show velocity and heading, including their errors over time, respectively.

values. Consequently, in the following this work concentrates on the error caused by the detection of the TO.

In contrast, the TO shows more significant errors. With $\mu_{\text{TO},d} = 1.56m$ and $\sigma_{\text{TO},d}$ $0.202m$, a rather large bias for the distance error is present. In this scenario, the TO is farther away to the front in SD than in GT. Likewise the errors of the velocity and yaw rate are significantly higher than for the EGO vehicle with $\mu_{\text{TO},v} = 0.456\frac{m}{s}$, $\sigma_{\text{TO},v} = 0.904\frac{m}{s}$, $\mu_{\text{TO},\theta} = -0.00612rad$, $\sigma_{\text{TO},\theta} = 0.0249rad$. It is important to notice that the derivation of velocity and heading directly from Lidar or Radar point clouds is difficult. Therefore, both calculate as derivatives of the position. As a consequence, these two values commonly inherit a time delay from the causal derivative that can be well observed in both of the lower plots from Figure 5.4. Additionally, this time delay causes the SD to be inconsistent with itself. At a certain point in time, the derivatives do not match the current change in position as the calculation delays them. This has a decisive impact on the scenario description presented in the next section. The entirety of errors caused by the sensors, object detection, and modeling of the environment references as ΔS in the following.

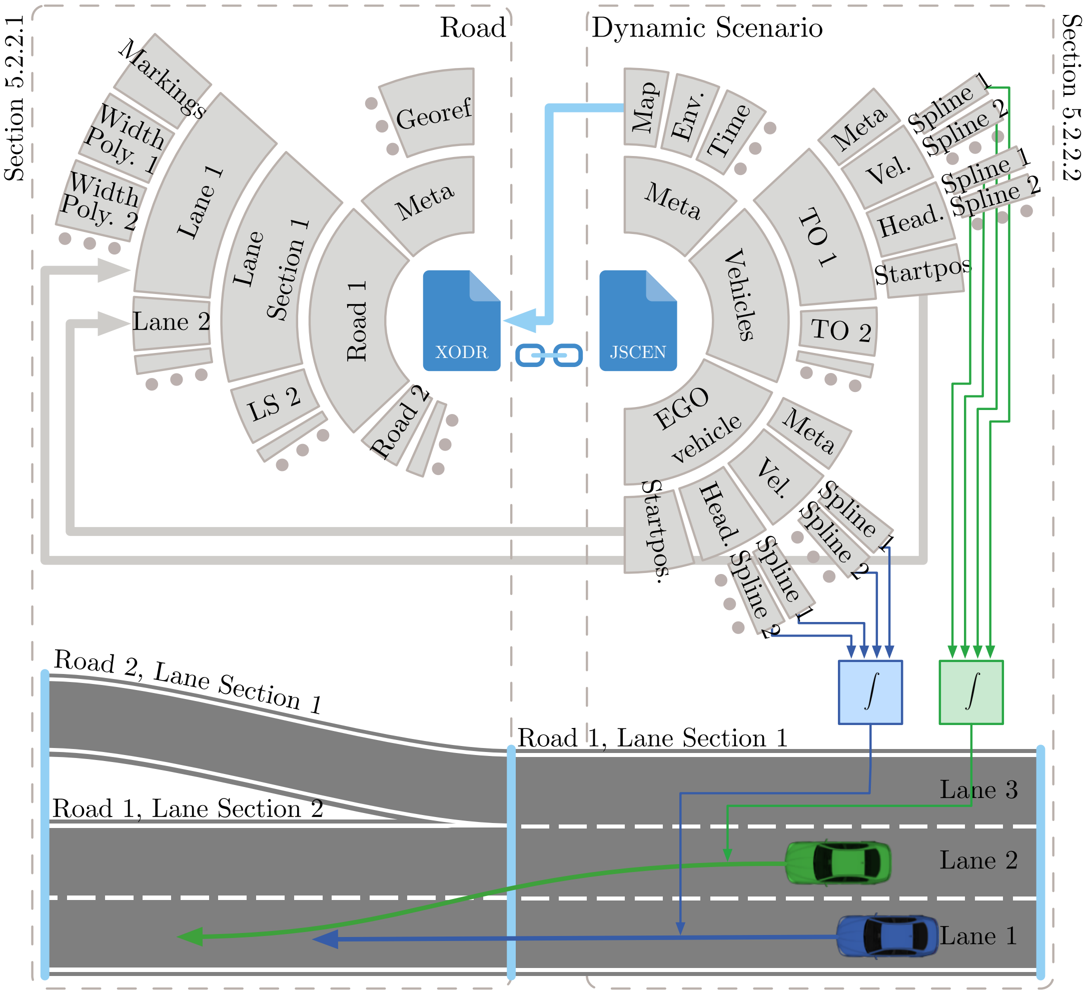


Figure 5.5.: The structure of the scenario description with its two components road network on the left and dynamic scenario on the right. The hierarchical structure of both descriptions is given in the embedded sunburst charts and explained in detail in Section 5.2.2.1 and Section 5.2.2.2. Arrows indicate how the traffic references to the road. The road with the scenario shown at the bottom of the figure explains how the two formats work together to describe the complete scenario.

5.2.2. Scenario Description

A scenario defines a test case for the assessment method. A test case, in turn, is a point in the test space or scenarios space. To match such a representation, a parameterized description language is necessary. Recapitulating the requirements, this description needs to be unambiguous and consistent to qualify as a distinct test case that the test domain cannot misinterpret. Further, a representation accurate enough with respect to the measurement is necessary not to affect the AD function's decision making.

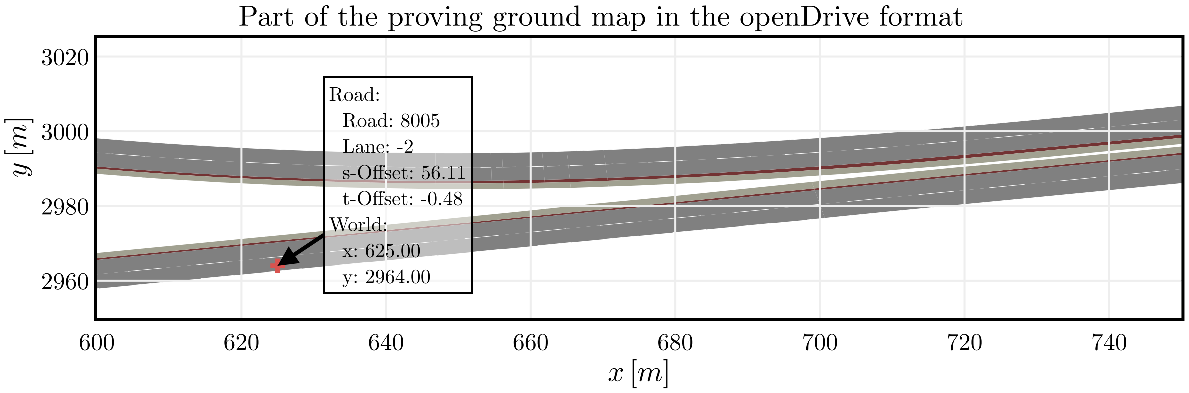


Figure 5.6.: Section of the proving ground road in *openDrive* used for the TDs with an arbitrary position point in both cartesian and road coordinates.

As already described in Section 3.2.2.2, the description splits into a static road and a dynamic environment containing the traffic. Figure 5.5 shows the schematic structure of these two components. In the following two subsections, the road description and the dynamic scenario description used in this work are described in detail.

5.2.2.1. Static Road Description

The static part of the scenario or the road network is described using the existing and approved *openDrive* format [108]. Its schematic structure shows in the left half of Figure 5.5. This schematic does not aim for completeness, but rather covers the essential components that are required for the rest of this work. Additionally, an exemplary road that matches the elements shown in the schematic presents at the lower end of the figure. A road network is structured in several increasingly detailing layers in its hierarchical structure embedded in an XML file format. On top of that, some meta-data such as the geological reference of the local map is given. On the highest level, roads define the structure of the network based on line shapes alongside which the road careers along. These can then be subdivided into parts called lane sections that make it easier to describe the lanes on the segment if, for example, the number of lane changes as in the shown road. For each section, multiple lanes are defined through polygons facilitating the road width and their lane marking type on the respective outer border. The lane indexing convention states that lane indices are negative decreasing for ongoing and positive increasing for oncoming traffic lanes on a road. Hence, the shown road only shows the oncoming road part. Additionally, by definition, there is a zero-width lane with index zero that is equal to the line shape that the road careers along and contains the missing lane marking between oncoming and ongoing road. Thereby, arbitrary road structures and networks can be defined.

As the format itself is already well defined in [108], only a tooling for building the road network form the available data is necessary. Such tooling is implemented and used throughout this work to visualize *openDrive* maps and perform transformations between the cartesian coordinate frame and local road coordinates. Figure 5.6 shows a section of the proving ground track used for the TDs in this work and calculated using the implemented tooling. In addition, an arbitrary point on the road is shown with both its cartesian and road coordinates. The latter consists of the road and lane index as well as the distance s-Offset from the road's start and the deviation to the lanes center t-Offset. Since the s-Offset starts at the beginning of a road, the lane section index can be calculated, and its inclusion in the road coordinates would be redundant. The use of this notation defines a road map based Frenet coordinate system that is used for localizing and positioning TOs in the second part of the scenario description.

5.2.2.2. Dynamic Scenario Description

The dynamic part of the scenario description concerns mainly the TOs, including their traces on the road. Section 3.2.2.3 lists various description languages. While description languages bound to a specific simulation tool are not suitable for the definition of generic applicable test cases, only one open standard named *openScenario* is available. As this standard defines traces as series of points over time accompanied by the inclusion of a vast amount of partially redundant data, this work follows the suggestion of [47] to use splines [173] for the definition of object traces. In the following, the Scenario description in JSON file format (JSCEN) description language based on the JavaScript Object Notation (JSON) file format, which is first invented in [47] and further refined in [Journal3], is introduced.

The right side of Figure 5.5 also lists the schematic structure of this format. On the highest level, meta-information such as the time of generation, environmental conditions, and the reference to the previous map is given. Despite the embedding of object traces is the same, the format differentiates the EGO vehicle from other traffic objects in the list on the highest level. That is because the EGO vehicle distinguishes from the other objects as the unit under test equipped with the AD function on the one hand. On the other hand, strictly speaking, the trace of the EGO is not part of the definition of a test case as it needs to be free to perform rather than being guided along a predefined path. This is the only aspect where the set definitions of scenario and test case differ. However, the EGO trace is still included in the format for completeness of the scenario description. It allows the calculation of assessment metrics later on. Therefore, the description of the traces and the conversion is now explained independently from the object type.

Every embedded trace originates in its start position that is a reference to the previously introduced *openDrive* road, as shown exemplarily in Figure 5.6. From this point, the traces are given relatively in a local cartesian coordinate frame through splines. To avoid data redundancies as stated in the requirements, using either position over time or velocity and heading over time is sufficient as one of these representations can always be calculated from the other. The chosen data basis is velocity v and heading θ over time t as multi-segment cubic splines. Let $v_S(t)$ and $\theta_S(t)$ be the spline representations of those two quantities with $N_{S,v}$ and $N_{S,\theta}$ segments, respectively. In between those segments, the coordinate vectors $(\mathbf{t}_{S,v}[i],\ \mathbf{v}_S[i])\ \forall i \in [1,\dots,N_{S,v}+1]$ and $(\mathbf{t}_{S,\theta}[i],\ \boldsymbol{\theta}_S[i])\ \forall i \in [1,\dots,N_{S,\theta}+1]$ define the connection points of the spline segments

$$v_S(t) = \begin{cases} v_{S,1}(t) = \mathbf{a}_v[1] + \mathbf{b}_v[1]t + \mathbf{c}_v[1]t^2 + \mathbf{d}_v[1]t^3 & \mathbf{t}_{S,v}[1] \le t < \mathbf{t}_{S,v}[2] \\ v_{S,2}(t) = \mathbf{a}_v[2] + \mathbf{b}_v[2]t + \mathbf{c}_v[2]t^2 + \mathbf{d}_v[2]t^3 & \mathbf{t}_{S,v}[2] \le t < \mathbf{t}_{S,v}[3] \\ \vdots & \\ v_{S,N_{S,v}}(t) = \mathbf{a}_v[N_{S,v}] + \mathbf{b}_v[N_{S,v}]t + \mathbf{c}_v[N_{S,v}]t^2 + \mathbf{d}_v[N_{S,v}]t^3 & \mathbf{t}_{S,v}[N_{S,v}] \le t \le \mathbf{t}_{S,v}[N_{S,v}+1], \end{cases} \tag{5.1}$$

with $(\mathbf{a}_v[i],\ \mathbf{b}_v[i],\ \mathbf{c}_v[i],\ \mathbf{d}_v[i])\ \forall i \in [1,\dots,N_{S,v}+1]$ being the internal spline parameters for every segment. Those for the heading spline are defined analogously. Note that there is always one more connection point than segments in a spline. The definition of splines, as well as the underlying data, demand the spline segments to preserve continuity between the segments. More precisely, C^2-continuity [174] ensures through the following boundary conditions:

$$\begin{aligned} v_{S,i}(\mathbf{t}_{S,v}[i]) &= \mathbf{v}_{S,v}[i] && \forall i \in [1,\dots,N_{S,v}] \\ v_{S,i}(\mathbf{t}_{S,v}[i+1]) &= \mathbf{v}_{S,v}[i+1] && \forall i \in [1,\dots,N_{S,v}] \\ v_{S,i}(\mathbf{t}_{S,v}[i+1])' &= v_{S,i+1}(\mathbf{t}_{S,v}[i+1])' && \forall i \in [1,\dots,N_{S,v}-1] \\ v_{S,i}(\mathbf{t}_{S,v}[i+1])'' &= v_{S,i+1}(\mathbf{t}_{S,v}[i+1])'' && \forall i \in [1,\dots,N_{S,v}-1] \\ v_{S,1}(\mathbf{t}_{S,v}[1])'' &= v_{S,N_{S,v}}(\mathbf{t}_{S,v}[N_{S,v}+1])'' = 0. \end{aligned} \tag{5.2}$$

Again these conditions exist analogously for the heading splines. The internal $4N_{S,v}$ polynomial parameters of the cubic spline segments are dependent on each other. Though the boundary condition, they are unambiguously determined through the coordinates of the connections points $(\mathbf{t}_{S,v}[i],\ \mathbf{v}_S[i])$. Hence, we define the parameter vectors $\mathbf{p}$, $\mathbf{p}_v$, and $\mathbf{p}_\theta$ for both of the splines as:

$$\begin{aligned} \mathbf{p}_v &= [\mathbf{t}_{S,v}[1], \mathbf{v}_S[1], \dots, \mathbf{t}_{S,v}[N_{S,v}+1], \mathbf{v}_S[N_{S,v}+1]] \\ \mathbf{p}_\theta &= [\mathbf{t}_{S,\theta}[1], \boldsymbol{\theta}_S[1], \dots, \mathbf{t}_{S,\theta}[N_{S,\theta}+1], \boldsymbol{\theta}_S[N_{S,\theta}+1]] \\ \mathbf{p} &= [\mathbf{p}_v, \mathbf{p}_\theta]\,. \end{aligned} \tag{5.3}$$

Under these circumstances, the fitting on actual data can be performed. In Section 5.2.1, it shows that the position data is the most accurate in both GT and SD measurement. Therefore, we use the time series data $\mathbf{x}_M$, $\mathbf{y}_M$, and $\mathbf{t}_M$ with N_{smp} as the basis for the fitting. Then, the optimization problem of the fitting defines as

$$\begin{aligned}
&\underset{\mathbf{p}}{\operatorname{argmin}} \left(\frac{1}{N_{\text{smp}}} \sum_{n=1}^{N_{\text{smp}}} \sqrt{\left(\mathbf{x}_{\text{M}}[n] - \mathbf{x}_{\text{S}}(\mathbf{t}_{\text{M}}[n])\right)^2 + \left(\mathbf{y}_{\text{M}}[n] - \mathbf{y}_{\text{S}}(\mathbf{t}_{\text{M}}[n])\right)^2} \right) \\
&\mathbf{x}_{\text{S}}(t) = \int_{\mathbf{t}_{\text{M}}(1)}^{t} v_{\text{S}}(\tau|\mathbf{p}_v) \cos(\theta_{\text{S}}(\tau|\mathbf{p}_\theta)) d\tau \\
&\mathbf{y}_{\text{S}}(t) = \int_{\mathbf{t}_{\text{M}}(1)}^{t} v_{\text{S}}(\tau|\mathbf{p}_v) \sin(\theta_{\text{S}}(\tau|\mathbf{p}_\theta)) d\tau.
\end{aligned} \tag{5.4}$$

As position points compare to the velocity and heading splines, integration of the splines is necessary. This is achieved through a simple movement model embedded into Equation (5.4) and trapezoidal integration. These two approximations do not cause any error in the framework, as the conversion between velocity and heading to position point is performed the same way every time it is necessary. To ensure the stability of the optimization and guarantee the correct mapping of the full trace, the following conditions are defined:

$$\begin{aligned}
\mathbf{t}_{S,v}[i+1] - \mathbf{t}_{S,v}[i] - 0.1 \geq 0 \qquad & \forall i \in [1, \dots, N_{S,v} - 1] \\
\mathbf{t}_{S,\theta}[i+1] - \mathbf{t}_{S,\theta}[i] - 0.1 \geq 0 \qquad & \forall i \in [1, \dots, N_{S,\theta} - 1] \\
\mathbf{t}_{S,v}[1] = \mathbf{t}_{S,\theta}[1] = \min(\mathbf{t}_{\text{M}}) & \\
\mathbf{t}_{S,v}[N_{S,v} + 1] = \mathbf{t}_{S,\theta}[N_{S,\theta} + 1] = \max(\mathbf{t}_{\text{M}}) &
\end{aligned} \tag{5.5}$$

They ensure that the segment connection points' time components are always in the right order with a small safety margin, and the first and last connection point enframe the whole trace in a timely manner. The *optimize.minimize()* function from *scipy* [165] with the L-BFGS-B algorithm [175] performs the optimization of the parameter vector $\mathbf{p}$. After a successful optimization, the internal parameters $(\mathbf{a}_v[i], \mathbf{b}_v[i], \mathbf{c}_v[i], \mathbf{d}_v[i])$ $\forall i \in [1, \dots, N_{S,v} + 1]$ for velocity $(\mathbf{a}_\theta[i], \mathbf{b}_\theta[i], \mathbf{c}_\theta[i], \mathbf{d}_\theta[i])$ $\forall i \in [1, \dots, N_{S,\theta} + 1]$ for heading are calculated through the boundary conditions from Equation (5.2) and *scipy*'s *interpolate.CubicSpline()* [165] class. Finally, the internal parameters embed into the JSCEN file format as a parameterized and unambiguous representation of both the EGO and TOs traces. Certainly, choosing the number of spline segments used for fitting the traces is crucial for accuracy. In general, more segments allow for smaller errors but bear the risk of overfitting into signal noise. Therefore, a superordinate algorithm repeats the whole fitting process with an increasing number of segments until a certain threshold of relative improvement towards its predecessor is not reached anymore. A threshold of 2% has shown to be suitable and is used in the following. For reference, an example of the JSCEN file from the GT information of the representative scenario is available in Appendix A.1. It is to mention that this fitting is only suitable for traffic participants that follow a bicycle-like movement. Other participants, such as pedestrians, could make the fitting of the heading difficult due to volatile changes in direction but are considered to be out of scope in this proof of concept.

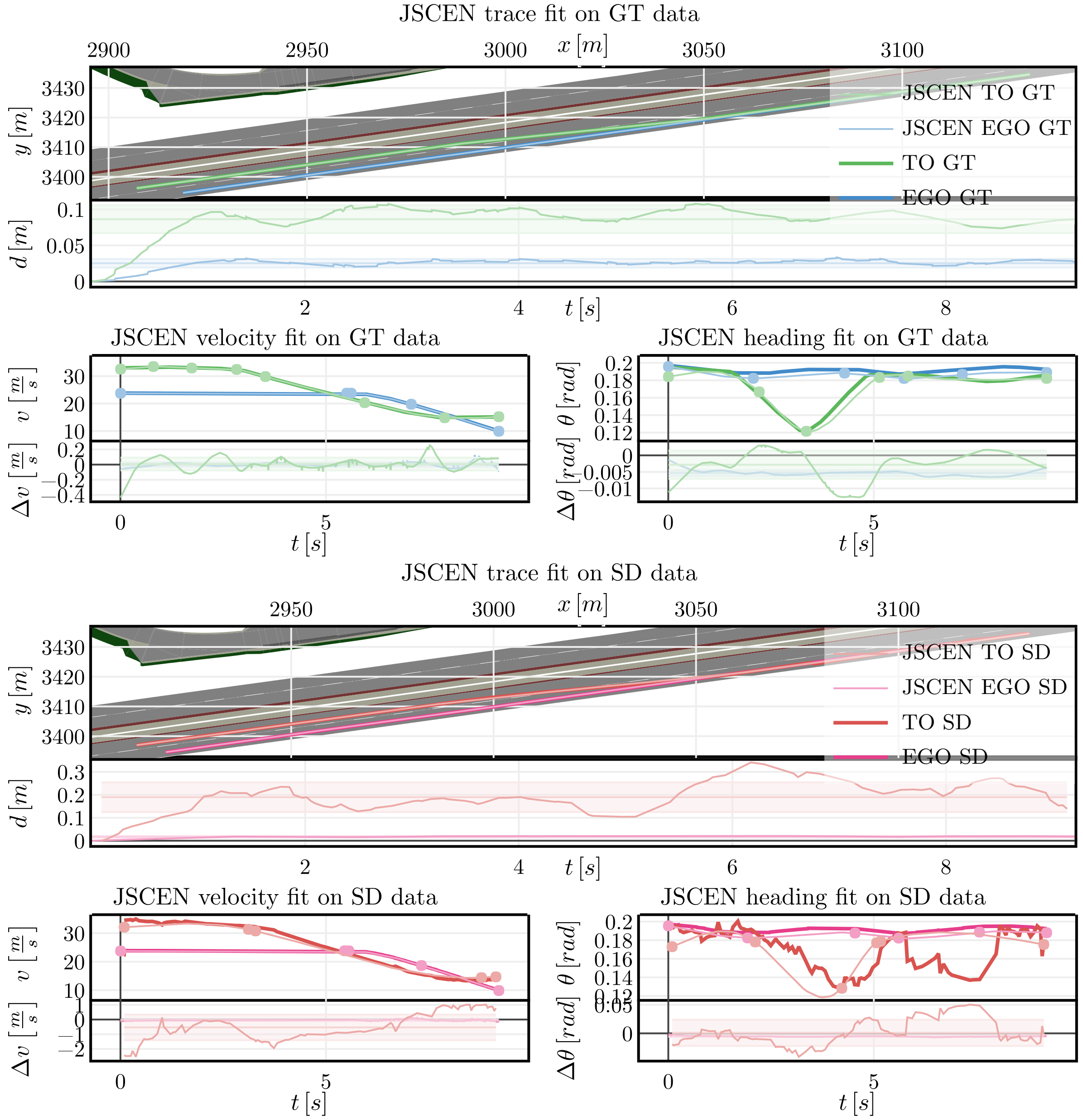


Figure 5.7.: Fitted JSCEN scenario on both the GT data in the upper half and the SD data in the lower half.

	EGO GT		TO GT		EGO SD		TO SD	
	μ	σ	μ	σ	μ	σ	μ	σ
$d\ [m]$	0.0252	0.00602	0.0863	0.0193	0.0170	0.00401	0.190	0.0648
$\Delta v\ \left[\frac{m}{s}\right]$	−0.000469	0.0287	0.00785	0.0848	−0.0567	0.0432	−0.533	0.870
$\Delta\theta\ [rad]$	−0.00536	0.000883	−0.00283	0.00426	−0.00540	0.000820	0.000902	0.0228

Table 5.1.: Error of the JSCEN fit for the representative scenario. The most significant errors highlight in red and lower errors in green.

Presented JSCEN file format and fitting now applies to both GT and SD measurement data of the representative scenario. The results, including the errors arising, are shown in Figure 5.7, quantified in Table 5.1, and explained in the following. The figure shows the trace, velocity, and heading including the fitting errors in the same order as previously with the measurement. But this time, both GT in the upper and SD in the lower half. The fittings are superimposed to the measurements with thinner and lighter colored lines. Additionally, the respective velocity and heading plots show dots where the fitting algorithm chooses the spline segment connection points.

The fitting error on a position trace level is lower for GT data than for the more noise SD, but both remain in an acceptable centimeter range. The table highlights the error quantities in green. It also highlights the ability of spline with their smooth nature to filter noise in the data. For GT, the error also remains low for velocity and heading. In contrast, those two quantities reveal more significant errors in SD for the TO. That is because the measurement data of velocity and heading is not used in the fitting process in favor of the more accurate position data. Thereby, the fitting is able to cancel the time delay that is observable in the SD in Section 5.2.1. This is observable in the velocity plot by the slight backward shift of the fit, but even more evident in the heading with a larger backward shift. Additionally, the fitting cancels the wrongly sensed second dip in the heading. Hence, the fitting can improve the errors in ΔS towards GT by removing inconsistencies. The JSCEN fit of the SD does not represent the world as the AD function has seen it, but does fulfill the consistency requirement. How this affects the remaining reprocessing is discussed in the next section, where an actual re-simulation is performed. For the remainder of this work, the error induced by fitting on GT is reference as $\Delta\mathrm{FIT}_{\mathrm{GT}}$ and SD as $\Delta\mathrm{FIT}_{\mathrm{SD}}$ analogously.

5.2.3. Simulation

The simulation's task is to recreate what is written in the scenario description as accurate as possible. This includes the guidance of the described TOs along their traces and the EGO vehicle's initial setup. During the simulation, the EGO must be free to perform to observe the simulated world's AD function reaction. Otherwise, a comparison with its real-world counterpart would be meaningless.

This work uses the BMW internal simulation framework called SPIDER [176]. Its use in the presented framework is possible with a few modifications. First, the compatibility with the presented JSCEN scenario file format is established. For that, a position-based trace for TOs calculates through the same model and integration used within the fitting of Equation (5.4). Thereby, errors due to the simple model and integration technique are avoided, as explained in the last section. SPIDER utilized the position points to guide the TOs along the predefined trajectory. Unfortunately, it cannot be avoided that a controller uses the position points as a guideline, resulting in control errors. Second, the EGO needs to be initialized with its starting conditions. A few seconds before the actual scenario starts, the EGO is guided towards its starting position defined in the JSCEN description. The physical vehicle model reaches a state that complies with the scenario's initial conditions, such as initial velocity. After that, the scenario can begin, and the EGO vehicle's control hands over to the AD function, which is embedded in closed-loop with the simulation. Of course, the used driving function is the same piece of software that is used in the TDs before. Lastly, SPIDER's capability to stream the simulated data through Open Simulation Interface (OSI) [177] is exploited to record the results in the same format as the TD measurements. The received, re-simulated traces denote in the following as Resimulated GT (rGT) and Resimulated SD (rSD) for GT and SD.

Using those modifications, a re-simulation of the representative scenario based on both the GT and SD JSCEN description is possible. The results are presented in Figure 5.8, and its errors are quantified in Table 5.2. The plots' arrangement remains as before with the GT branch filling the upper half and SD the lower. Received data from the simulation shows dashed in the plots this time to distinguish from the respective measurement data counterpart, while the colors remain the same. In both trace plots, similar results are received. The TO s affected by a position error caused by the mentioned position controller of the simulated TO. This error is not negligible, but still significantly lower than the position error of the EGO. Hence, it is highlighted orange in the error table. The EGO vehicle reveals a significant error, especially in the last two seconds of the scenario. On closer inspection of the trace itself, it becomes apparent that the AD function decided to change a lane to the left rather than just breaking as in the measurement data. This is the mentioned specialty of the corner case scenario that is intentionally chosen in Section 5.2. The errors in the framework so far caused the AD function to make a different decision than in the TD. Also, the velocity and heading plots reveal this issue and provide further information about the behavior change. While the velocity decreases early in the re-simulation, observable by the dent in the dashed line before the solid line, its descent is also less steep, meaning softer breaking. Hence, the AD is able to react earlier in the simulation and perform a softer reaction. That this is in conjunction with a lane change becomes apparent in the heading plots by elevating the EGO vehicle's heading value towards the end of the scenario that is not present in the scenario description. Altogether, the errors arising in the re-simulation denote as $\Delta\mathrm{SIM}_{\mathrm{GT}}$ and $\Delta\mathrm{SIM}_{\mathrm{SD}}$ for GT and SD in the following.

Of course, the observed behavior change is unacceptable for the simulation to be cross-verified for that scenario. This section intentionally chose a scenario that triggers such an effect in the re-simulation to highlight the difficulties in achieving an accurate simulative result. Therefore, this issue is addressed again in Section 5.3.2, analyzed and traced back to the errors in the chain, followed by solution suggestions in Section 5.3.3.

After receiving the rGT and rSD traces, they are converted into the JSCEN file format again, receive their scenario space representation, and perform the assessment, which follows in the next section. It is shown that the error caused by this fitting process, namely $\Delta\mathrm{FIT}_{\mathrm{rGT}}$ and

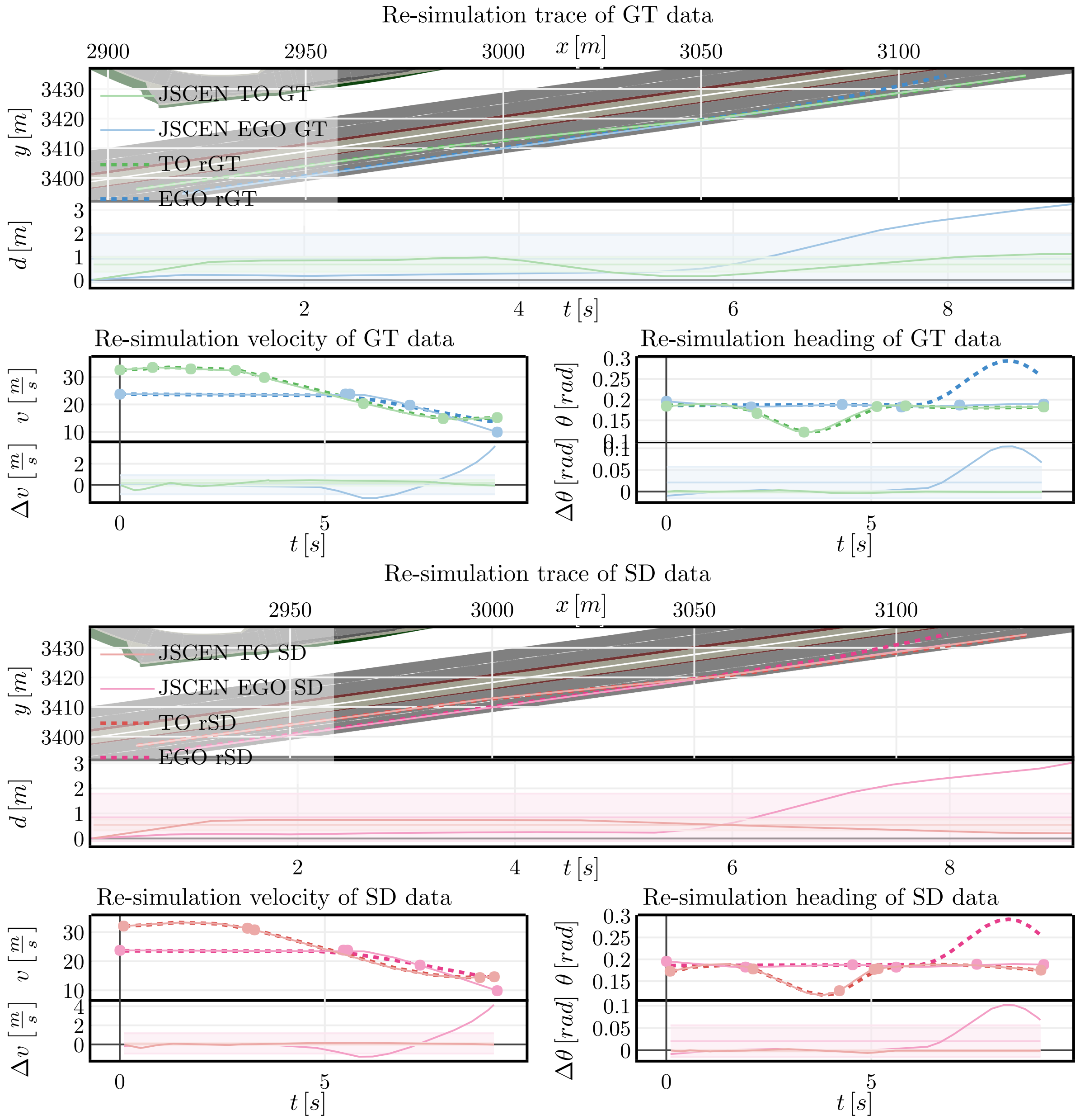


Figure 5.8.: Re-simulated scenario based on both the GT data in the upper half and the SD data in the lower half.

	EGO GT		TO GT		EGO SD		TO SD	
	μ	σ	μ	σ	μ	σ	μ	σ
$d\ [m]$	0.913	1.02	0.665	0.309	0.844	0.947	0.541	0.217
$\Delta v\ \left[\frac{m}{s}\right]$	0.00464	0.903	0.183	0.256	0.108	1.06	0.0511	0.122
$\Delta\theta\ [rad]$	0.0210	0.0365	-0.000670	0.00171	0.0205	0.0355	-0.000960	0.00179

Table 5.2.: Error of the re-simulation for the representative scenario. The most significant errors highlight in red, lower errors in green, and medium in orange.

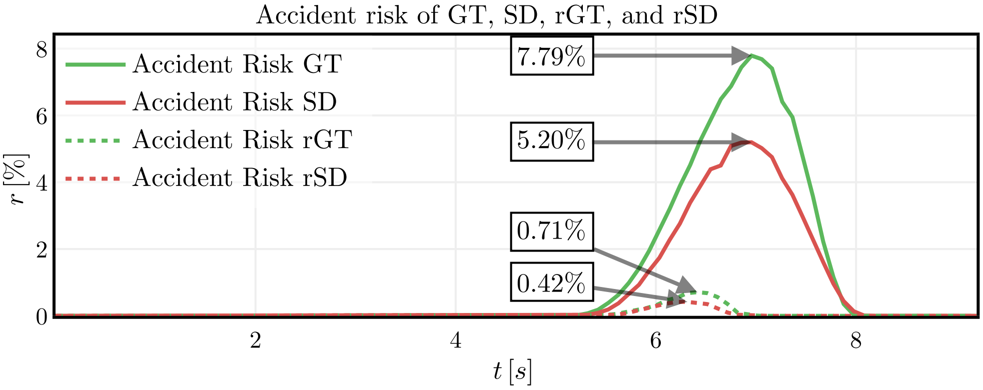


Figure 5.9.: Results of the risk measure for all four representations of the scenario. GT and SD are drawn in solid and rGT and rSD in dashed green and red, respectively. The maxima of each risk time series are marked and quantified.

$\Delta\text{FIT}_{\text{rSD}}$, is negligible low as the simulated data is smooth and well reproducible through splines [Paper2].

5.2.4. Assessment

The assessment module of the presented framework is based on the accident risk metric from Chapter 4. Therefore, further discussions about its theoretic background are not necessary at this point. The parameterization remains the same as specified in Table 4.1. Certainly, a production-ready framework requires a whole set of assessment KPIs, especially supplementary members from the regulatory and comfort categories, as explained in Section 3.2.3. However, for the proof of concept, this single representative value suffices. As the quality of the underlying data is crucial for the assessment metric, the consistent and smooth representation of the JSCEN file format is chosen as foundations. Not only it allows for a noise-less calculation of the required derivatives acceleration ACC and heading rate ω through analytic derivatives of the splines, but it has also revealed the ability to improve the erroneous and inconsistent SD. Using the noisy acceleration and heading rate data from SD would either require heavy filtering or would render the assessment results unusable. Hence, JSCEN is the best fit as a data foundation.

The accident risk is now calculated for all four available representations of the scenario, thus GT, SD, rGT, and rSD. The results of those are compared in Figure 5.9. First, it is observable that the risk for GT is higher than for SD with $r_{\text{GT}} \approx 7.79\%$ and $r_{\text{SD}} \approx 5.20\%$, respectively. Due to the TO appearing farther away and decelerating later in SD, this manifestation is reasonable. The true risk that needs to be assigned to the measured scenario in terms of the assessment is r_{GT}, as it pictures what happened in reality. The second observation arises from the risk of the re-simulation in dashed lines. Both rGT and rSD are significantly lower than their measured counterparts. That is because of the explained improved reaction of the AD function to the idealized data in simulation, as shown in Section 5.2.3. Hence, the errors and flaws in the current re-simulation framework that lead to the behavior change are also well observable in the KPI. The upcoming lessons learned discussion further controverts this topic.

For now, the functional principle behind all necessary modules for re-simulation is introduced. They define the necessary elements to perform cross-verification between reality and a virtual test domain. With the aid of the representative corner case scenario, it is shown where the major errors arise during the re-simulation framework. Certainly, the errors prevent the virtual domain from being called cross-verified. Therefore, Section 5.3.2 traces back those errors to their sources, leading to solution approaches in Section 5.3.3. Nevertheless, the next section explains the assembly of those modules in a global automated assessment framework prior to that.

5.3. Implementation, Results, and Lessons Learned

This section first explains the assembly of the previously introduced modules to the re-simulation pipeline and assessment framework. As the representative scenario already unveiled the major errors along the re-simulation steps, the following error analysis dedicates to the search of causes and working points for improvements. The suggestion and elaboration of such improvements finalize this section.

5.3.1. Implementation of the Assessment Supervisor

So far, the major modules are explained individually. In the case of assessment, a lot of required test conductions result in even more invocations of the modules. Certainly, an assembly into an integrated framework with automation and scaling capabilities is required. Such an implementation forms with the name *Assessment Supervisor*. Its structure is shortly explained alongside Figure 5.10[1] in the following.

Three key reasons lead to individual containerization of the module's software rather than running the whole framework directly on a computer. This means incorporating the program code into an isolated operating system, including its dependencies, that is then virtualized on operating system level. First, when the modules are embedded in their individual containers with defined interfaces to the outside world, the framework is highly modular by default, and modules can be exchanged in run-time. Second, isolation avoids dependency conflicts. Lastly, the framework can then be easily scaled to fit the current demand for tasks. For instance, if multiple simulations need to run, multiple simulation module containers can be executed in parallel, even on different computers across a network. The containerization framework of choice for the *Assessment Supervisor* is *Docker* [178]. Blue boxes visualize containers in Figure 5.10. Above all is the central controlling application allowing the measurement data upload, the controlling and automatizing data pipeline, and visualization of results. This is managed through a com-

[1] Docker and cogwheel icons made by Freepik from www.flaticon.com.

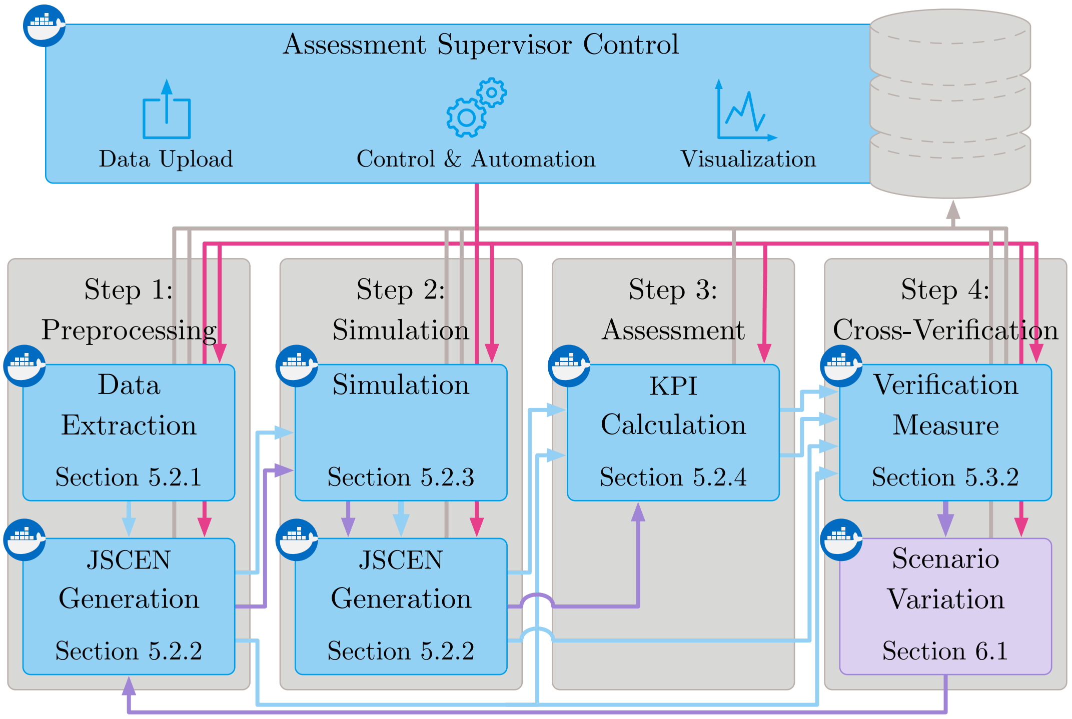


Figure 5.10.: Structure of the Assessment Supervisor. Modules embedded in *Docker* containers are shown as blue boxes, while gray boxes indicate the steps of the framework pipeline. Blue arrows depict the propagation of data in the reprocessing and assessment of real-world scenarios, magenta arrows the controlling of the control unit, gray arrows the data access, and purple the data propagation for the variation part that follows in Section 6.1.

bination of *Docker* virtualized volumes for files and a relational database for meta-information and file references. Additionally, the control modules offer a web-based Graphical User Interface (GUI) as the user's interface to the whole framework. A selected number of exemplary views of the GUI can be found in Appendix A.2. In the background, the modules and their respective containers group into the four steps of the pipeline. Step one is responsible for preprocessing the measured data by extracting either GT or SD information and converting it into the JSCEN file format. In a second step, the scenarios are simulated and converted into JSCEN using the same container as in the first step. The assessment in terms of KPI calculation takes place in the third step. The fourth and last step finally measures the difference between measured and simulated scenarios based on the KPIs and scenarios itself, which is discussed in Section 5.3.2. The data flow in the pipeline shows with blue arrows. In the current configuration the pipeline is set to run twice, both for GT and SD, to support discussions in Section 5.3.2. Figure 5.10 further depicts the usage of the cross-verified scenario for variation as an addition to the test volume, as discussed in Section 5.1.1. The respective container and the following data flow are shown in purple. More detail on the variation follows in the next chapter in Section 6.1.

The control module's governing influence over all others depicts as magenta-colored arrows and realizes through XML-based Remote Procedure Calls (RPCs). Further, the data access of the modules to the database indicates through gray arrows.

This implementation is used for both the data generation of the representative scenario in Section 5.2 and the cross verification and error analysis following in the next sub-section.

5.3.2. Towards Cross-Verification and Error Analysis

The aforementioned behavior change in the representative corner case scenario prevents the simulation's cross-verification for that exact scenario. Thus, it is necessary to trace back the roots of this phenomenon and resolve the respective errors. The content of this section is an extended prospect of the findings in [Paper2].

Therefore, the errors arising throughout the pipeline of Section 5.2 summarize in Figure 5.11. The errors of the three physical quantities d, Δv, and $\Delta\theta$ define the rows of this plot's table-like structure. The columns resembled the pipeline's individual steps for either GT or SD, depending on the colors of the embedded error box plots. Two of those boxes within each cell indicate the step's error for EGO on the left and TO on the right. A box plot consists of a rectangle embodying the errors range from the lower (25%) to the upper (75%) quantile of its distribution, whiskers marking the whole range and a superimposed diamond with mean in the center and the standard deviation extending to both sides as the tips. Each error pipeline starts from the true measured scenario, thus GT, and ends after the fitted re-simulated scenario. Therefore, the pipeline errors $\Delta\text{PIPE}_{\text{GT}}$ and $\Delta\text{PIPE}_{\text{SD}}$ summarize to

$$\begin{aligned}\Delta\text{PIPE}_{\text{GT}} &= \Delta\text{FIT}_{\text{GT}} + \Delta\text{SIM}_{\text{GT}} + \Delta\text{FIT}_{\text{rGT}}\\ \Delta\text{PIPE}_{\text{SD}} &= \Delta\text{S} + \Delta\text{FIT}_{\text{SD}} + \Delta\text{SIM}_{\text{SD}} + \Delta\text{FIT}_{\text{rSD}},\end{aligned} \tag{5.6}$$

for GT and SD, respectively. Both the introduction of the modules in Section 5.2 and the box plots show that some of those errors are more or less significant for the final results. For instance, the JSCEN fitting works well for GT, rGT, and rSD, while the underlying data's inconsistency causes a significant error in the SD case. Albeit this is only the case for the TO with assigned error $\Delta\text{FIT}_{\text{SD,TO}}$. Also, the sensor error is significant for the TO. Solely both simulation errors are dominated by the EGO vehicle followed by a lower but not negligible error for the TO. Altogether, considering only the significant errors reduces the error equations from Equation (5.6) to

$$\begin{aligned}\Delta\text{PIPE}_{\text{GT}} &\approx \Delta\text{SIM}_{\text{GT,EGO}} + \Delta\text{SIM}_{\text{GT,TO}}\\ \Delta\text{PIPE}_{\text{SD}} &\approx \Delta\text{S},\text{TO} + \Delta\text{FIT}_{\text{SD,TO}} + \Delta\text{SIM}_{\text{SD,EGO}} + \Delta\text{SIM}_{\text{SD,TO}}.\end{aligned} \tag{5.7}$$

Yet alone, this information does not help to find solution strategies. A more in-depth analysis of the causes follows. Certainly, different behavior of the AD function in the simulation, that manifest in $\Delta\text{SIM}_{\text{GT,EGO}}$ and $\Delta\text{SIM}_{\text{SD,EGO}}$, is triggered by differences in the virtual world

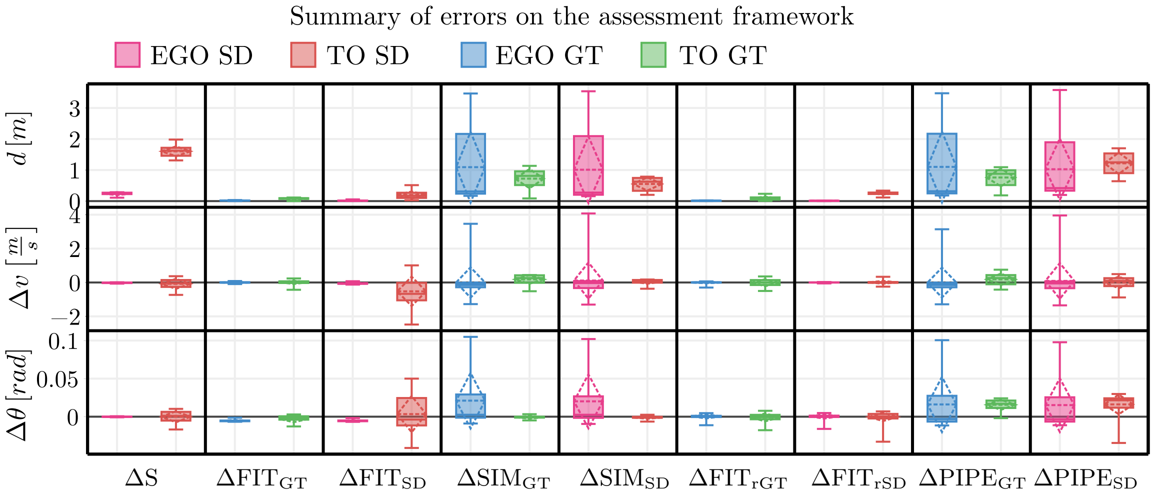


Figure 5.11.: Summary of the errors in the framework as box plots. The rows indicate the physical quantities and the columns the steps of the frameworks pipeline separated by GT and SD. Within each cell, a box plots for EGO and one for TO show the error boundaries, upper and lower quantile, and the mean and standard deviation.

compared to reality. Whether it is hard braking or soft braking with a lane change, the reaction itself is caused by the braking of the TO in front of the EGO vehicle. Then again, the TO reveals significant differences in the virtual world towards its real-world counterpart. Speaking of the GT path in the pipeline, the function is presented with information that it has not seen in reality. On the one side, the difference to the sensor information ΔS, on the other side, the error the simulation induces to the TO $\Delta\mathrm{SIM}_{\mathrm{GT,TO}}$. In the SD path, the significant TO errors arise in $\Delta\mathrm{FIT}_{\mathrm{SD,TO}}$, where the data improves towards GT, and $\Delta\mathrm{SIM}_{\mathrm{SD,TO}}$ as the simulations internal error similar to the GT case. The reason, why ΔS is listed in the SD pipeline, is that even if the framework is perfectly able to recreate the scenario in its SD perspective, it would still assess erroneous SD data that is not suitable for safety testing. Summarizing, in both SD and GT branches of the pipeline, the major differences are that the simulation is initially presented with data that is improved towards reality and the internal simulation error of the TO.

To find which of those two is responsible for the behavior change, both re-simulations' reaction time is compared to the GT reality in Figure 5.12. This comparison is the simplest by inspecting the velocity profile of the scenario representations. The figure contains the profiles of GT, rGT, and rSD for the EGO with the kink significant for the braking maneuver, and the TO GT for reference. A vertical indicator line shows the point in time where the deceleration falls below $-0.5\frac{m}{s^2}$, considered as braking, and hence the time, the AD function realized its reaction to the incident. It becomes apparent that the difference in reaction time $\Delta t_{\mathrm{react\,GT}\leftrightarrow\mathrm{rGT}}$ and

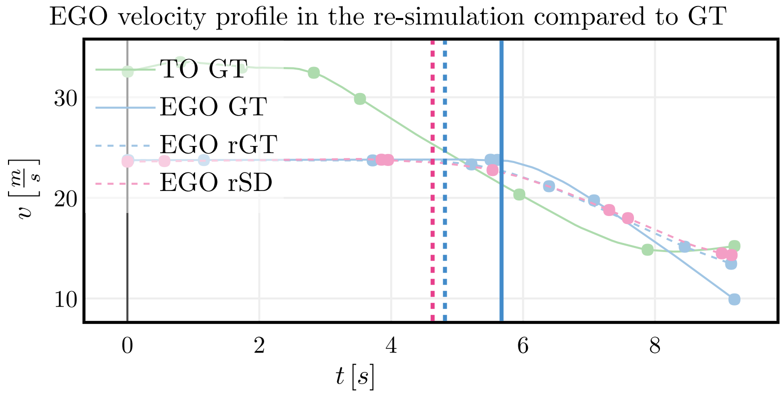


Figure 5.12.: Comparison of the EGO vehicle's reaction time in rGT and rSD towards GT among their velocity profiles. The velocity of the GT TO is shown for reference. Vertical indicators mark the time the AD function realizes its reaction to the incident.

$\Delta t_{\mathrm{react\,GT\leftrightarrow rSD}}$ between the two re-simulations and the GT are significant:

$$\begin{aligned}\Delta t_{\mathrm{react,GT\leftrightarrow rGT}} &= 0.86s \\ \Delta t_{\mathrm{react,GT\leftrightarrow rSD}} &= 1.05s.\end{aligned} \tag{5.8}$$

The timing differences $\overline{\Delta t}$ caused by the positional error in $\Delta \mathrm{S_{TO}}$, $\Delta \mathrm{SIM_{GT,TO}}$ and $\Delta \mathrm{SIM_{SD,TO}}$ are

$$\begin{aligned}\overline{\Delta t_{\Delta \mathrm{S_{TO}}}} &= \frac{\mu_{\mathrm{TO},d,\Delta \mathrm{S_{TO}}}}{v_{\mathrm{EGO}}} \approx \frac{1.56m}{23.7\frac{m}{s}} \approx 0.066s \\ \overline{\Delta t_{\Delta \mathrm{SIM_{GT,TO}}}} &= \frac{\mu_{\mathrm{TO},d,\Delta \mathrm{SIM_{GT,TO}}}}{v_{\mathrm{EGO}}} \approx \frac{0.665m}{23.7\frac{m}{s}} \approx 0.028s \\ \overline{\Delta t_{\Delta \mathrm{SIM_{SD,TO}}}} &= \frac{\mu_{\mathrm{TO},d,\Delta \mathrm{SIM_{SD,TO}}}}{v_{\mathrm{EGO}}} \approx \frac{0.541m}{23.7\frac{m}{s}} \approx 0.023s,\end{aligned} \tag{5.9}$$

with the approaching speed of the EGO $v_{\mathrm{EGO}} = 23.7\frac{m}{s}$ and the respective mean distance errors. If one compares those to the difference in reaction time, it becomes apparent that solely distance can not be the leading cause of the behavior change. The influence of errors in v and θ with its time delay must be more important. Figure 5.4 revealed that the time delays within those two quantities reside in the magnitude of the reaction time differences.

Thereby, $\Delta \mathrm{SIM_{GT,TO}}$ and $\Delta \mathrm{SIM_{SD,TO}}$ are eliminated as the main root of the change, as they do not contain errors in those two quantities. Hence, the choice is limited to $\Delta \mathrm{S_{TO}}$ and $\Delta \mathrm{FIT_{SD,TO}}$, those two that show a different world in the GT path or correct sensor errors in the SD. This exposes the necessity of a correct representation of the sensor errors in the simulation. How this can produce relief on the frameworks flaws is discussed in the next section. At this point arises the question, why not to use raw sensor information and stream it into the simulation. First, raw sensor data does not represent a scenario as a valid test case and cannot be mapped into a test space as it does not reflect what really happens and is not consistent. An assessment

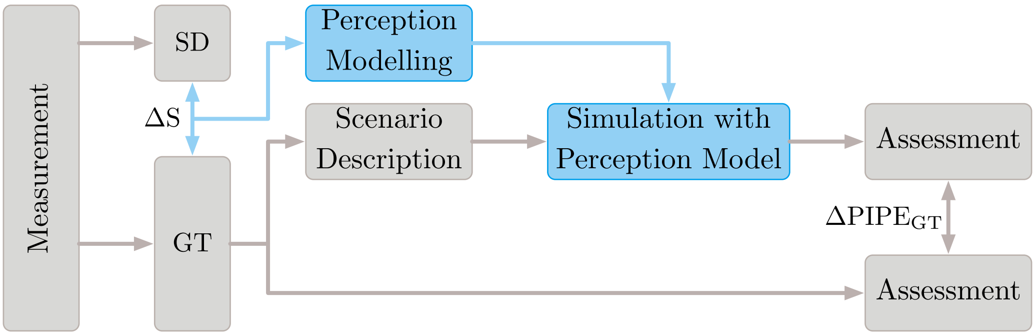


Figure 5.13.: Integration of a perception model into the re-simulation pipeline [Paper4]. The model is trained from ΔS data and injected into the simulation of the GT path to recreate the perception of the AD function from reality while simulating the true GT scenario.

based on this information would be erroneous. Second, a meaningful variation on erroneous and inconsistent data is infeasible, and without variation, a pure reprocessing of real-world scenarios does not add to the test volume and hence does not benefit the assessment problem itself.

5.3.3. Lessons Learned and Suggested Improvements

The leading cause can be isolated to the sensor error ΔS, which is not recreated accurately in both GT and SD reprocessing. Hence, a cross-verification without the inclusion of the offsets and sensors' delays is not possible for now. The streaming of raw sensor data into simulation is already ruled out in the previous section due to methodological reasons. Therefore, modeling of the sensor inaccuracies inside the simulation is required. A suggestion on such a model and its inclusion into the pipeline follows.

A variety of existing sensor modeling techniques is discussed and evaluated against its use in this pipeline in [Paper4]. Classical sensor models map the differences in the GT measurements on an individual sensor's raw data level. However, the reprocessing pipeline considers the information on the level of an already fused environmental model, ruling out classical sensor modeling. To put relief on the errors arising in this pipeline, a model mapping the differences on this level is required. As it describes the world's perception through the eyes of the AD function based on GT information, the term *perception model* is a better fit than sensor model. Further, it shows that the constellation and occurrences of and within a scenario positively impact the sensor errors. In conclusion, a maneuver-based perception model based on a reasonable amount of collected GT and SD data is suggested in [Paper4].

How such a model integrates into the presented re-simulation pipeline is shown in Figure 5.13. From a reasonable amount of ΔS data, a maneuver-based perception model is trained, which in turn is injected into the simulation of the GT path. There, it recreates the AD function's perception as close as in reality as possible, while still simulating the true GT scenario. A matching perception in simulation and reality promises to reduce $\Delta\text{PIPE}_{\text{GT}}$ significantly. Still, the assessment result is valid, as GT information is present.

While this promises relief on the more significant part of $\Delta\text{PIPE}_{\text{GT}}$, still a not negligible error $\Delta\text{SIM}_{\text{GT,TO}}$ remains. An improvement at this point can only be achieved by improving the simulation accuracy itself. However, the realization of both improved simulation and perception models is out of this work's scope. With provided results and evidence so far, we believe that the cross-verification becomes feasible by including those two improvements, even for corner case scenarios. That, in turn, enables the subsequent variation and pure virtual conduction of scenarios adding to the test volume, as explained in Section 5.1.

5.4. Conclusion

The task to complete in this chapter is to formalize a methodology that enables virtual domains to participate in the assessment through cross-verification. Implicitly, the impact of choice for a data basis in reprocessing real-world scenarios has to be analyzed.

First, the architecture of the assessment methodology is presented. Besides the pipeline for reprocessing and cross-verification, it also hints at the use of a locally verified test domain that is addressed later in Chapter 6. The following explanation of the pipeline's individual modules is accompanied by a test against a critical corner case scenario. The results present after every step of the pipeline. As this scenario is intentionally chosen to challenge the methodology and its implementation, more significant differences between real-world and simulation are revealed in the form of a behavior change of the EGO vehicle. A more in-depth analysis of the error's cause reveals the impact of the chosen data basis for the re-simulation. While a correct assessment of the scenario requires GT, also an SD representation is required to trigger the same behavior in simulation as observed in the measurement. This leads to the inclusion of perception models in the simulation, which is suggested and also described.

For production use, still, extensions are necessary at some points. For example, the scenario fitting is not suitable for all kinds of traffic participants, and an evaluation in other ODDs than highways is pending. While the presented methodology promises to enable the simulation's cross-verification with real-world measurements, the current development state of the simulation impedes success. Hence, the remedy in the form of the perception models and improved simulation accuracy is described. Evidence for its success derives from the error analysis. However, the actual improvement of the simulation and development of perception models is out of the scope. As the evidence of the suggested solutions' success is clear through the error and impact analysis of GT over SD, the second hypothesis confirms at this point: *An assessment methodology can be formalized that locally verifies virtual test domains against real-world test drives through reprocessing of scenarios. The deviation of sensor information from ground truth in a measured scenario significantly impacts the reprocessing quality.*

On a side note, the presented methodology is designed to assess the AD function and not the whole system, as defined in Section 3.1.3. This is due to the test cases already defining a perceived environment. The sensor's consideration of the environment model part in the assessment is a whole other research topic, for example, addressed in [179].

Nevertheless, reprocessing and cross-verification alone does not help to cope with the vast amount of necessary real-world testing for the assessment of AD. Therefore, the next chapter considers scenario variation and scenario space definitions.

6. Scenario Variation and the Scenario Space

The assessment methodology overview Figure 5.1 lists two components that remain uncovered so far. Those are the local variation, without which the methodology would not add to the test volume and the analysis of the scenario or test space. Therefore, the third research task is addressed: *How can the set of test cases be defined, traversed, and coverage be estimated?* This chapter expands both topics individually.

The variation is supposed to start from a single, through the previous methodology locally verified scenarios, to generate similar test cases in its local proximity. Thereby, additional test kilometers add to the assessment, illuminating the test space around a real-world scenario and verified through the cross-verification of the original scenario. Additionally, similar results on the KPI side for similar scenarios strengthen the assessment results' credibility [Journal3]. For variation already, Section 3.3.4.1 concluded that research about local scenario variation is missing. Hence Section 6.1 suggests a methodology for that.

Afterward, an insight into the topic of scenario space definition is given in Section 6.2. As scenarios do not easily map to an orthogonal space where coverage through samples can be estimated, a more elaborate definition of such a test space is required. The findings extend the state of the art from Section 3.3.4.2.

Lastly, Section 6.3 discusses the synergy between those two topics and the previous chapter's overall method. On a theoretical basis, it is shown how coverage of the scenario space can be reached and which current limitations hinder this task's empirical execution.

6.1. Local Scenario Variation

A few requirements need to be fulfilled to make use of local scenario variation within the methodology from Chapter 5. First, the variations need to be local. Otherwise, the cross-verification of local scenarios can not suffice to gain trust in the simulation of variated scenarios. Thereby, existing methods from the state of the art in Section 3.3.4.1 drop out a priori. Then, variations must be both naturalistic and physically possible. If they cannot possibly be driven or never occur in reality, their inclusion into the assessment is not advantageous. Lastly, the controllability of the variations' range is desirable. The further away from the original scenario, the less it can profit from the local verification of the simulation's original scenario.

The first question arising asks what should actually be variated. Especially for local variation, not all of the aspects defined in Section 3.2.2.2 make sense. Some of these are discrete, such as the number of road lanes, and a variation would violate locality. For example, adding another lane to an existing road opens a whole lot of new maneuver options that can make the prior critical situation easily avoidable. Hence, the focus is drawn on continuous parameters. This proof of concept only considers the variation of TOs.

Two approaches are investigated. Due to the smooth nature of naturalistic driving trajectories, first, the application of cubic Bèzier curves is considered in [Paper2]. More precisely, as a vehicle's velocity and heading cannot jump instantaneously from one value to another, $\mathcal{C}^2$-continuity of the spatial domain's trajectory is desirable. Despite similar trajectories that satisfy the smoothness constraint that emerges the method, there is no guarantee that they are naturalistic or physically drivable. Therefore, this work does not go any further into the detail presented in [Paper2]. A second approach, first presented in [Journal3] and based on Principal Component Analysis (PCA), shows more convenient results and is presented in the following.

6.1.1. Use of PCA in the Variation of Scenario Trajectory Data

The general problem of any variation is how to variate the parameters. This involves the type of alteration and its extent. For driving trajectories, the problem of dependence of the parameters exists. Whether one varies the base points of the JSCEN splines as described in Section 5.2.2.2 or the trajectory's coordinates, the parameters can not be altered freely. Otherwise, unrealistic, physically impossible, and or non-naturalistic paths are created. However, conventional sampling methods used for variation, such as random, full factor sets, Sobol sets [180], Halton sets [181], or Latin Hypercube Sampling [182], operate on an orthogonal space and hence create uncorrelated sample data. That is where PCA can fill the gap. Based on enough available correlated data, it can find a mapping to an orthogonal, lower-dimensional subspace of that data and resolves correlation. If applied correctly, it is possible without the loss of too much information about the original data. An explanation of the PCA from a mathematical point of view can be found in Appendix B.1. The following explains first the PCA training based on trajectory data in Section 6.1.2 and the application of them in local trajectory variation in Section 6.1.3.

6.1.2. Training of the PCA Kernels

A PCA training based on existing trajectory data is required to find the mapping function from the correlated trajectory space to the aforementioned uncorrelated subspace. Hence, naturalistic driving data is required. As the trajectory's shapes among possible driving scenarios are versatile and differ significantly in their length, it seems convenient to split them into maneuvers. Therefore, extraction and data preparation follow first.

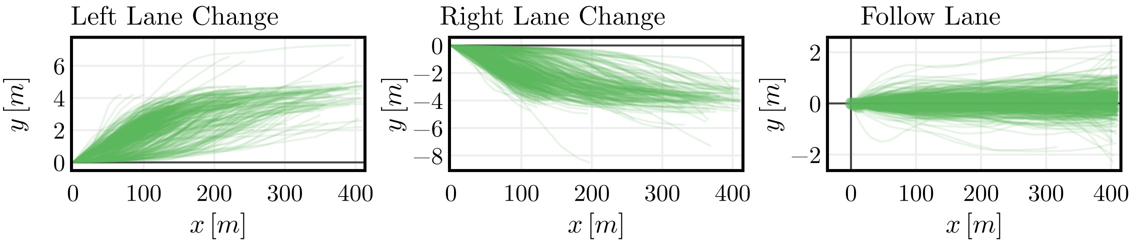


Figure 6.1.: Extracted maneuver traces from the *highD* dataset [97]. In each case, left lane changes on the left, right lane changes in the middle, and lane following maneuvers on the right, 500 traces from the extracted data are plotted.

6.1.2.1. Extraction of Maneuvers form Naturalistic Driving Data

For the naturalistic trajectory data again, the *highD* dataset [97] is used as a substitute for NDS_1. Hence, this proof of concept is limited to highway maneuvers. Every trajectory from the dataset splits into the three maneuvers *left lane change*, *right lane change*, and *follow lane*. This can be easily achieved by defining thresholds on the lateral velocity of a vehicle with respect to its lane to trace sections containing a lane change. A more detailed description of the maneuver detection follows in Section 6.2.1.1. Consequently, the remaining parts of the trajectory are lane following maneuvers.

In Figure 6.1, for each of the three maneuvers, 500 trajectory segments achieved through this splitting are shown. The trajectories are shifted and rotated to always start at the origin in the direction of the x-axis. Thereby it is guaranteed that the subsequent PCA solely trains the mapping of the maneuver's characteristics without any side effects as arbitrary start positions. The left lane changes on the left and right on the middle show a median lateral shift that matches the lane width on German highways, which is either $3.75m$ or $4m$ [183]. This emphasizes the correctness of the splitting. However, individual double lane changes remain in the data sets. Whether to count them into the lane change maneuver sets or as individual maneuvers is a matter of opinion, and as they are inferior in terms of numbers, the sets remain this way. Lane following maneuvers are shown on the right and contain all trajectories that do not leave the lane. Naturalistic behavior, such as driving with an offset to the lane's center or swerving around it, is contained. In the next step, a PCA trains on each of these three sets.

6.1.2.2. PCA Kernel Fitting

Training individual PCAs on maneuvers instead of whole trajectories has several benefits. First, the trajectories to learn in a single PCA are similar in terms of their characteristics, reducing the training's complexity significantly. Then, the trajectories remain in a limited scope with respect to their length. The longer a trajectory is, the mode features might be necessary to contain the whole characteristic. This is not the case for the segments. Lastly, individual PCAs

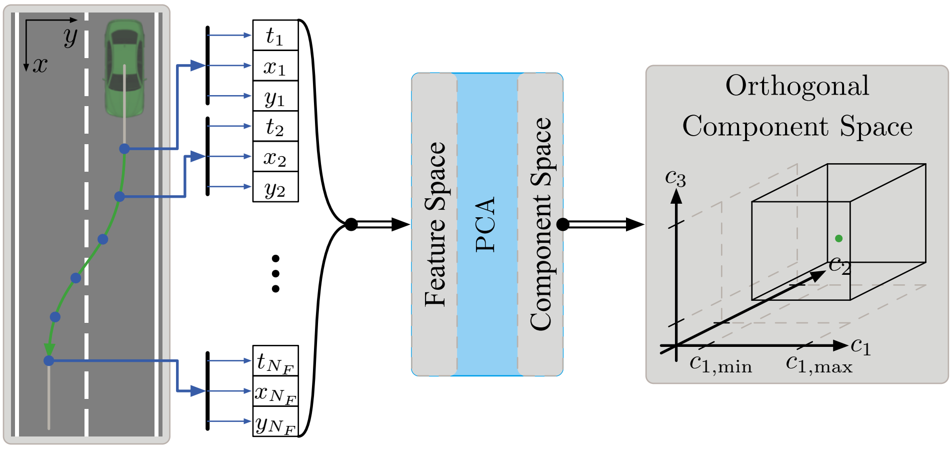


Figure 6.2.: Training of the PCA maneuver kernels. The feature extraction and vectorization from an exemplary right lane change are shown on the left. The right side shows its representation in the orthogonal component space.

allow an individual variation of a traces segment. Hence, for a cut-in, the typical lane change can be variated without altering the remaining segments. The trained PCAs are referred to as maneuver kernels from now on.

However, still, trajectories of different lengths are to be trained. Additionally, it does not make sense to push every single sampling point into the PCA. Hence features are selected as a fixed number of equitemporal points on the path, where each point consists of the time components t and the coordinates x and y. Figure 6.2 depicts this assembly to the feature vector from a lane change trajectory on the left. To make sure that the features also contain smaller swerving of offsets to the characteristic shape of a lane change, $N_F = 50$ points are extracted from the trajectory resulting in 150 features. The strong correlation of those features is observable in Figure 6.3. In this figure, the columns are for lane change left, right, and follow lane maneuvers, respectively. The upper row shows the standard Pearson Correlation Coefficient (PCC) in-between the features, while the lower row shows the Nonlinear Correlation Coefficient (NCC) [184]. Throughout both measures and all maneuvers, a strong correlation is observable as red and orange colors, and the application of the PCA is motivated. A description of the NCC can be found in Appendix B.2.

The training itself is accomplished through *scikit-learn*'s *PCA()* [185] class. This implementation's parameterization consists of setting the number of desired components, which is not a trivial task. Choosing too many components results in unwanted correlations between them, while too few cause loss of information from the features. The right amount is a balanced compromise between those two properties. A short parameter search reveals that $N_C = 20$ components are a good fit for the three present maneuvers. On the one hand, Figure 6.4 confirms this choice as a very low correlation is observable throughout all matrices. At this point, also the

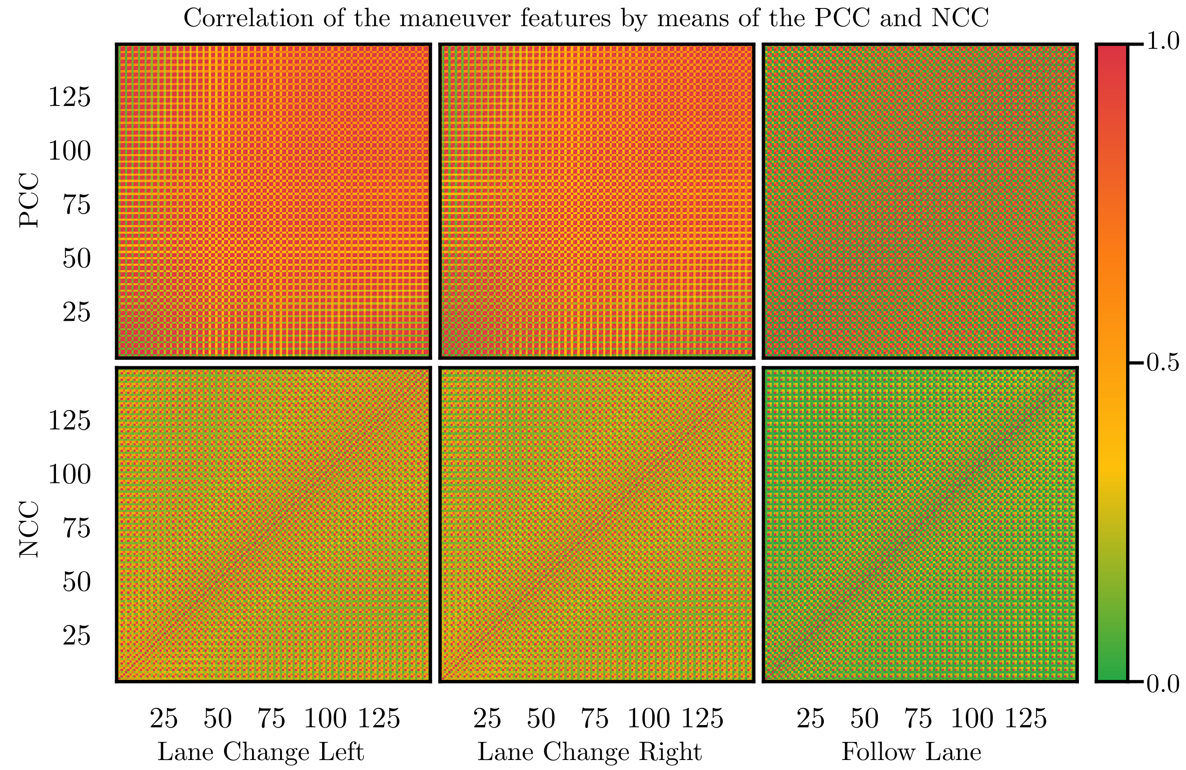


Figure 6.3.: Correlation of the features from the extracted maneuvers. Each column belongs to the maneuver category. The top row shows correlations employing the PCC while the lower row uses the NCC. Orange to red color throughout the matrices reveals a strong correlation between the features.

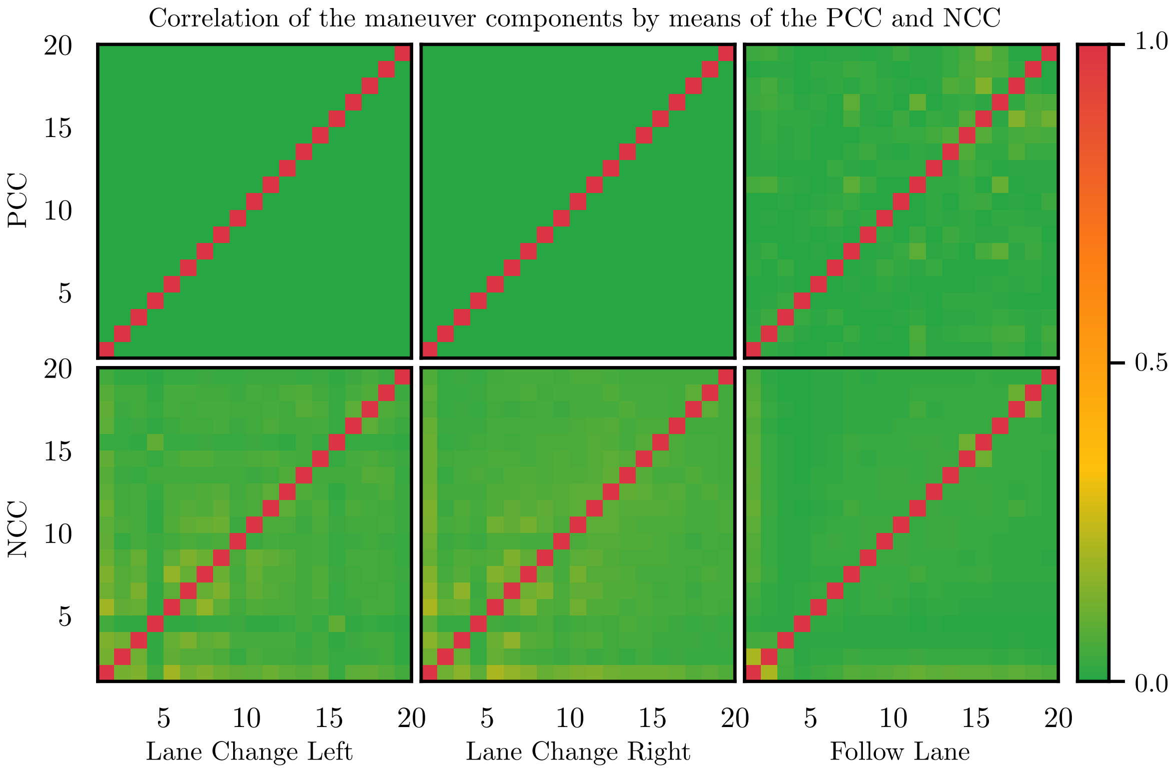


Figure 6.4.: Correlation of the components from the transformed maneuvers features. Low correlation as green color throughout all maneuvers and both correlation measures. Solely the diagonal shows strong correlation as indeed a component always correlates with itself.

choice of an additional nonlinear correlation measure is justified. As PCA is a method to resolve linear correlations in high order data and there are no nonlinear correlations present in its results, there is no need to applicate nonlinear component analysis techniques for the decorrelation of these maneuver features. On the other hand, the relative error of the self-mapping, hence the back and forth conversion of features to components, is lower than 0.1% throughout all three maneuvers. Hence, the trained PCAs provide a good mapping of the chosen features into a lower-dimensional, orthogonal subspace. As shown on the right of Figure 6.2, the representation of a scenario within this subspace is a point contained in an N_C-dimensional hyper-rectangle, shown pictorially as three dimensions in the figure. The boundaries $c_{i,\text{min}}\ \forall i \in N_C$ and $c_{i,\text{max}}\ \forall i \in N_C$ can be estimated by transforming the whole set of maneuvers from the NDS$_1$ into the components space. The next step explains how the trained transformation can be exploited for scenario variation.

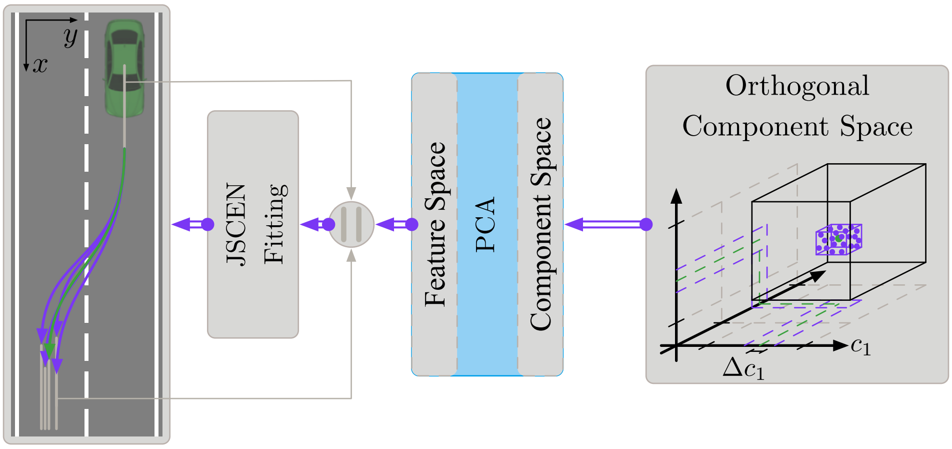


Figure 6.5.: Trajectory variation using the PCA maneuver kernels. On the right, a local hyper-rectangle around the original scenario defines the variations range in the component space. The generated samples show in purple. The left side concatenates the back-transformed features with the remaining maneuvers to TO traces that convert to JSCEN again.

6.1.3. Variation of Trajectories

The orthogonal nature of the component spaces and the components' uncorrelated property can be exploited for the variation. To preserve locality, also the representation as a point of the original maneuver is altered locally. Also, the extent defines through the determined boundaries $c_{i,\text{min}}$ and $c_{i,\text{max}}$ from the previous steps. It is convenient to choose a percental portion Δc_i of the enclosed interval as a variation range. In Figure 6.5, the range shows as a purple enclosure on the right. The parameter Δc_i is defined in the direction of component i. Hence, the variation interval is twice as big. Next, samples have to be generated within this enclosure. These should be uniformly distributed throughout the framed space while the number of samples may be chosen freely. Hence, the Halton set algorithm [181] with Reverse Radix Scrambling [186] is chosen. Additional benefits of this technique are determinism and reproducibility, which are mandatory for safety assessment. A mathematical description of this sampling method can be found in Appendix B.3. After sampling, the inverse transformation from the PCA generates the respective feature representation or coordinate points in time of the variated maneuver. Concatenated with the TO's other maneuvers, the whole trajectory serves as input for the JSCEN fitting from Section 5.2.2.2. In contrast to fitting raw data, this fitting optimization starts from the original trajectory, ensuring fast convergence into the same minima.

6.1.4. Results

The presented variation method now applies to the overall assessment method and the representative scenario from Section 5.1. As the last chapter concluded, GT is the correct foundation

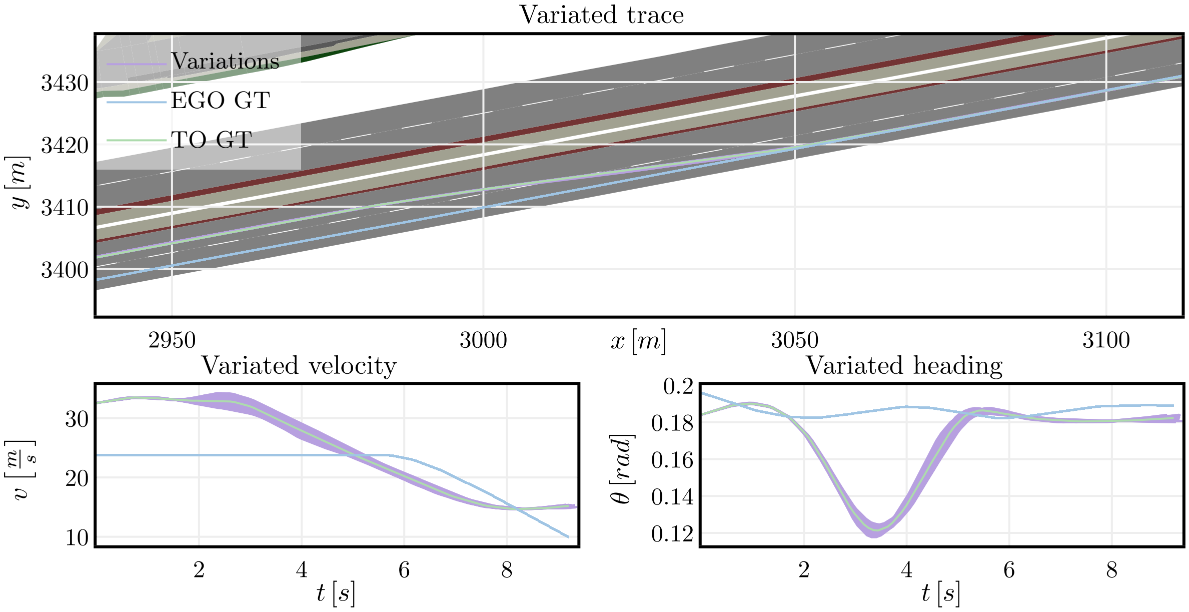


Figure 6.6.: Variations of the representative scenario. The traces are shown at the top while the lower row presents velocity and heading on the left and right, respectively. As a reference, the GT scenario is plotted with familiar colors. Variations are shown in purple.

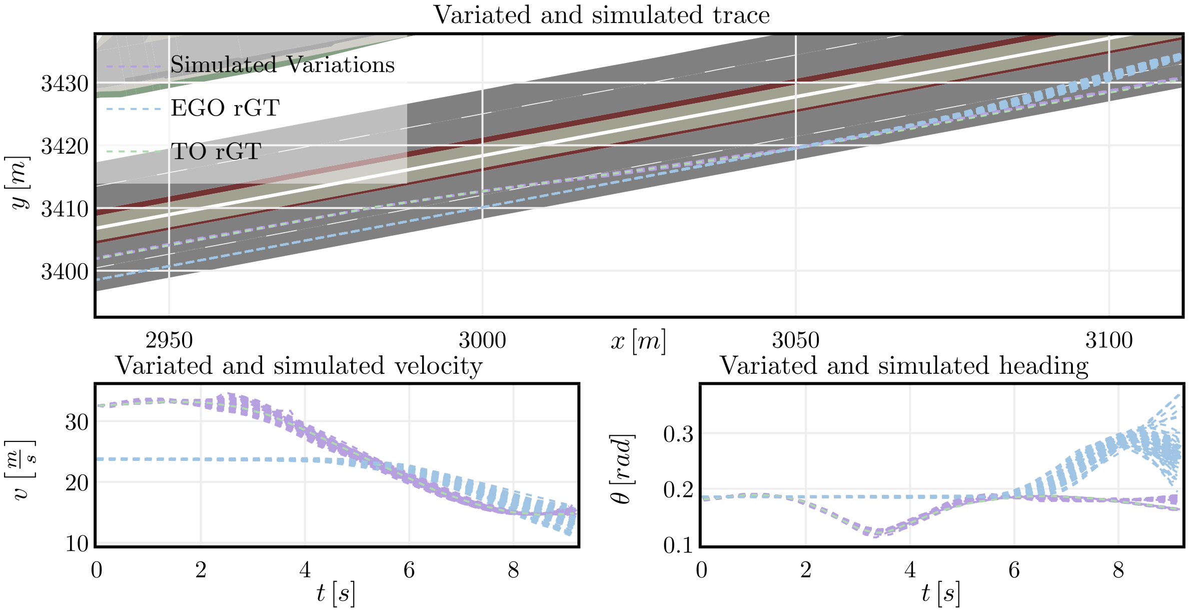


Figure 6.7.: Simulations of the scenario variations. The order of the plots and the coloring of the contents remains the same as in Figure 6.6

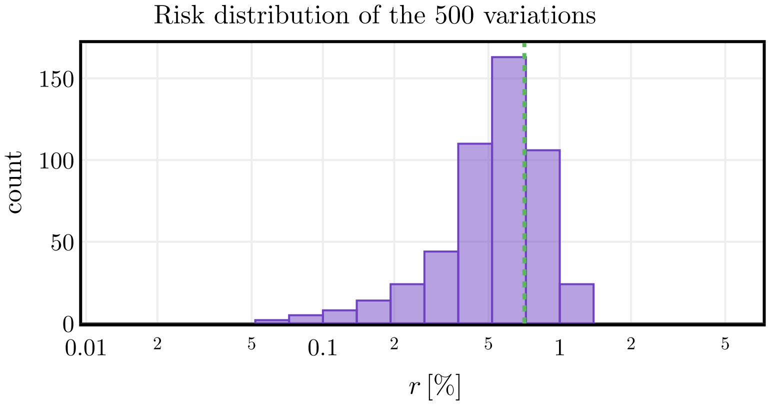


Figure 6.8.: Risk distribution of the 500 variations. A dashed green line indicates the risk of the original rGT scenario.

for an assessment, and it also forms the variation. The representative scenario characterizes through a right lane change maneuver of the cutting-in and braking TO. Hence, this maneuver is the variation target, and the trained right lane change PCA kernel applies to it. Within the components space the relative variation range of $\Delta c_i = 0.5\% \cdot (c_{i,\max} - c_{i,\min})\ \forall i \in [1, \ldots, N_C]$ is chosen.

Altogether, 500 variations are generated and simulated for this experiment. The results limited to 100 traces are shown in Figure 6.6 with the traces at the top, velocity, and heading on the bottom left and right, respectively. The locality of this small variation is best observable in the trace plot, in which the set of purple variation trajectories spans a small ribbon around the original trajectory in green. This ribbon is more apparent in the velocity and heading profiles than in position. Furthermore, the scenario did not change drastically, and the locality of the variation is present. The blue EGO traces are only shown for reference as the scenario must be simulated first to observe the AD function's reaction. Therefore, the simulation results of those 100 variations show in Figure 6.7 with a similar structure as before. The EGO vehicle, presented as blue dashed traces in all three plots, adapts its behavior to the original scenario's variations. It is observable that also the ego variates only locally. Within the traces velocity and heading plots, a small ribbon of EGO traces appears. Again the locality of the variation is confirmed.

The last step to ensure locality throughout the whole pipeline is calculating the KPI on the variations. Figure 6.8 shows a logarithmic histogram over all 500 variations' risk values. The dashed green line is the rGT risk of the representative scenario for reference. The same simulation, including its issues discussed in Chapter 5, is used to perform the variated scenarios' tests, and its expectable risk results resemble around rGT rather than GT. Indeed, the purple distribution builds around the dashed green line with its peak, including the reference. Thereby also, the last instance confirms the locality of the variation.

The presented variation has shown to provide local results as required by the assessment methodology from Chapter 5. Additionally, the outcome provides naturalistic and physically drivable trajectories motivated by training with NDS$_1$ data. This is crucial for the assumption that the variations can add to the test volume. Only for similar local scenarios, similar performance in terms of simulation accuracy can be expected. Then again, the variation's results can inherit thrust in the form of usability in the assessment from the local cross-verification. Finally, purely simulated results can add to the test volume. However, not all aspects are addressed with this prototype. The remaining questions include the variation of other components of a scenario than the TO trajectories. Additionally, the influence between multiple TOs arising with individual variation remains to be analyzed. Next follows the definition of a scenario space which is the second part necessary for coverage considerations made at the end of this chapter.

6.2. Defining the Scenario-Space

One of the most significant advantages of a scenario-based assessment approach is the ability to map scenarios or test cases to a scenario or test space. Then again, a lot of mapped scenarios enable the coverage statements based on these spaces. In contrast, a kilometer-based approach is not mappable to a space, and hence, a statement on how many possible scenarios are covered is not possible. Prior to a coverage statement, the definition of a scenario space is required. This section dedicates to answering the question, how such a space could be defined.

For seamless integration into the whole test concept, the scenario space should be naturalistic and all-encompassing. This means that it must be able to hold near to every scenario that is possible in reality. The overview of related work in Section 3.3.4.2 reveals that research in this direction is sparse. The given approaches are either not all-encompassing because of the manual definition of the space or not naturalistic as they base on simulation. Therefore, additional research is necessary in this field.

This work presents a clustering approach based on large-scale driving data. As scenarios consist of a large amount of both discrete parameters such as the number of TOs and dynamic parameters such as the vehicle's trajectories, clustering directly on this data is not feasible. Hence, a two-step approach is presented. First, scenarios are extracted and partitioned into buckets of similar scenarios based on their discrete parameters, as explained in Section 6.2.1. Afterwards, Section 6.2.4 explains how a novel distance measure between scenarios within a bucket enables clustering.

This section's findings are a consolidated explanation of the work elaborated in [Thesis6] and published in [Paper5].

6.2.1. Extraction and Partitioning of Large Scale Scenario Data

A crucial foundation for the scenario space is a large amount of data. For the proof of concept, again, the *highD* data set is used [97]. Therefore, the results are limited to the ODD of German

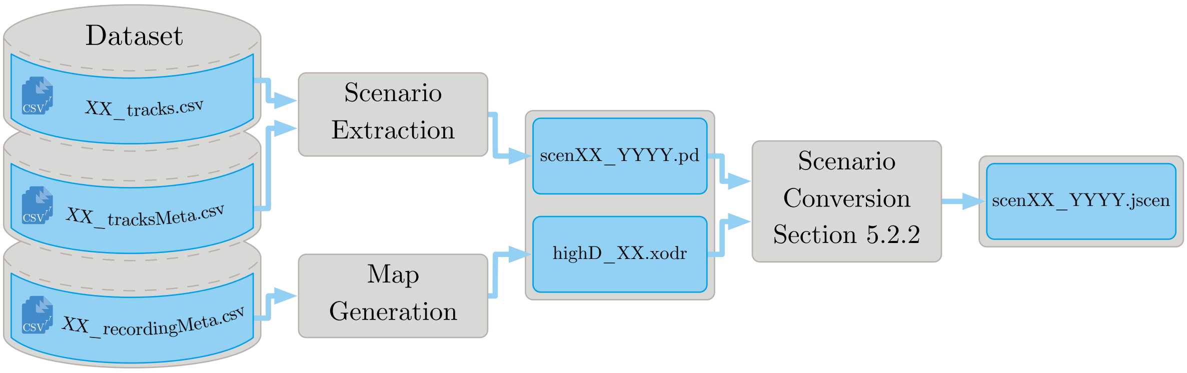


Figure 6.9.: The workflow of the large scale scenario extraction. Scenarios and maps are based on the information contained in the provided CSV files. Together they are fed into the known scenario fitting from Section 5.2.2 to reveal the scenarios in the desired format.

highway scenarios with two or three road lanes and highway traffic participants.

The process of the scenario extraction is now explained alongside Figure 6.9. The starting point is based on the Comma-Separated Values (CSV) files provided by the dataset. From top to bottom, they contain information on the recorded vehicle traces in the *tracks* files, meta-information about vehicles in the *tracksMeta* files, and the road topology in the *recordingMeta* files. Indeed, the recordings contain more than one hour of driving data containing hundreds of vehicles each and do not qualify as scenarios themselves. Hence, The extraction of scenarios follows next as the main focus of this subsection. The dataset contains more than 100000 vehicles, leading to a further limitation of this proof of concept. Only scenarios containing at minimum one lane change are considered in the following. This decision is motivated by the fact that lane changes are one of the primary sources of risk in highway traffic. Section 6.2.1.1 explains the detection of maneuvers as a first step of the scenario extraction. The next step focuses on the spatial context of a scenario that determines the vehicles that are relevant for the EGO in Section 6.2.1.2. Lastly, based on the vehicles' temporal appearance, the scenario's temporal context is extracted in Section 6.2.1.3. Several thousand extracted scenarios are stored with reference *XX* to the recording number and *YYYY* as an assigned scenario ID with these steps. Additionally, they are sorted into the mentioned buckets based on discrete parameters, which is explained in Section 6.2.1.4. Besides the vehicles' traces, the road is required in the *openDrive* file format [108]. The data relating to the road geometry translates into a simple straight road. As this process is trivial, this work does not go into further detail. Lastly, to make the scenarios readable by the same simulation tooling used in Section 5.2.3, both the vehicle traces and the road are fed into the scenario fitting algorithm from Section 5.2.2. The explanation of the important extraction step in detail follows next.

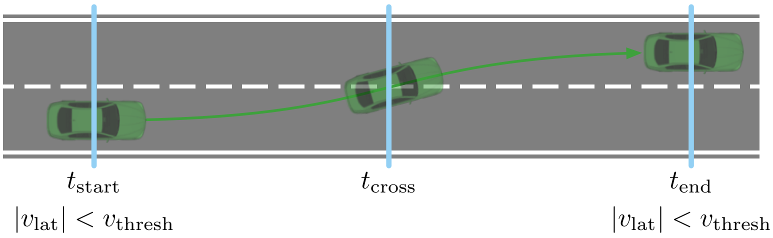

t_{start}
$|v_{\text{lat}}| < v_{\text{thresh}}$
t_{cross}
t_{end}
$|v_{\text{lat}}| < v_{\text{thresh}}$

Figure 6.10.: Description of the lane change maneuver detection. After identifying the crossing time in the middle, the vehicle is tracked back and forth in time until its lateral velocity falls below a certain threshold.

6.2.1.1. Maneuver Detection

The detection of lane change maneuvers is already explained in Section 6.1.1 for a basic understanding and detailed within the following. The characteristic scene of this maneuver type is when a vehicle crosses the border between two lanes. In Figure 6.10, this happens at time t_{cross} in the middle. Once such an event is detected while iterating over a vehicle's trajectory, tracking back and forth in time reveals the start t_{start} and end t_{end} of the maneuver. In both directions, the algorithm continuous its search until the TO's lateral velocity falls below a certain threshold v_{thresh}. Lateral velocities below this threshold are considered as the vehicle's neutral driving state within a lane. Manual tuning of the threshold parameter reveals $v_{\text{thresh}} = 0.03\frac{m}{s^2}$ as a good fit. Additionally, through the direction and extent of the identified maneuver, it can be tagged as *left*, *right*, *double left*, or *double right* lane change.

The presence of lane changes within recordings becomes essential in the next step of the scenario extraction process in which a spatial and temporal filter is applied to find the context of a scenario.

6.2.1.2. Spatial Scenario Context

A scenario assembles around the perspective of a particular vehicle, namely the EGO. All other vehicles are then TOs. Certainly, not all objects appearing in an hour-long recording on a $400m$ highway section as in highD are relevant to the EGO. Furthermore, including all of them in the scenario description would explode the parameter space. Hence, a spatial context filter is required to identify the TOs relevant to the EGO driving and decisions.

The Eight-Vehicle-Model (8VM)[1] can be exploited for the construction of such a filter. Figure 6.11 illustrates a scene on a three-lane highway section with nine vehicles, including the

[1] There is no exact source for the 8VM as used here. Instead, the idea that only these eight vehicles are relevant for the own driving behavior extracts from the traffic challenger description used in PEGASUS. A publicly available description can be found on slide 8 of [187].

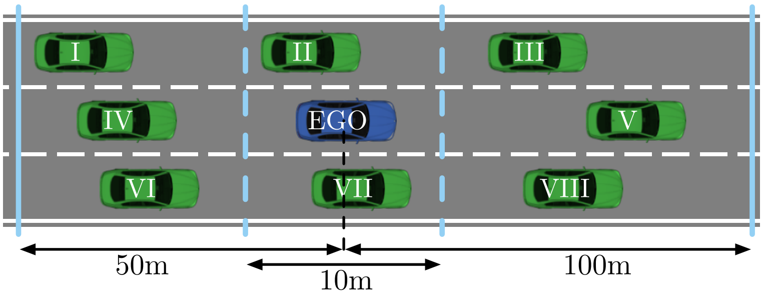


Figure 6.11.: The Eight-Vehicle-Model (8VM) used in the spatial scenario context filter. The identified TOs are numbered and assemble in green around the EGO in blue. The boundaries of the longitudinal cells are displayed as dashed and solid blue lines. Laterally the cells are defined by the lane boundaries.

EGO in the center. The eight TOs shown are the vehicles considered by this filter. They locate on the EGO's lane (V and IV), one to the left (I, II, and III) and one to the right (VI, VII, and VIII) as well as behind (I, IV, and VI), in-front (III, V, and VIII) and roughly on the same level as the EGO (II and VII). While the lateral boundaries are well-defined through the lanes, the longitudinal delimitation requires three parameters. First is the range equally apportioned in-front and behind the EGO, where TOs are considered to be on the same level. The figure shows through the dashed blue delimiters, and a width of $10m$ has shown to be a good fit for reliably identifying adjacent vehicles and minimizing the chance of two vehicles ending up in that space. Next, the ranges beyond the dashed borders are necessary. A distance of $100m$ in-front between the EGO vehicle's center and the solid blue line and $50m$ to the back enframe the search range for preceding and following vehicles. At those ranges, it is certainly possible that more than one TO is within that space. In this case, always the one closed to the EGO is chosen by the filter.

Yet unresolved is the determination of the EGO vehicle, as the *highD* dataset contains TOs solely. Every TO within a measurement is once chosen to be the EGO. Concomitantly, it is made sure that at minimum one of the surrounding vehicles identified by the 8VM performs a lane change. The latter becomes more evident with the temporal context filter in the next subsection. By choosing every TO to be the EGO once, the number of extracted scenarios increases greatly.

The 8VM, as explained, does identify the relevant TOs for a single scene. Therefore, the process must repeat for every time step of the EGO's alive time. The maximum of eight vehicles only applies to a scene. TOs can leave and enter the cells during a whole scenario, resulting in more than eight possible relevant TOs.

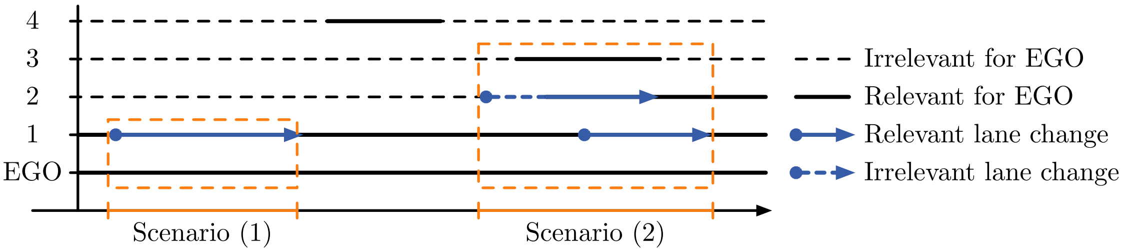


Figure 6.12.: Example of the functionality of the temporal context filter. A measurement with four TOs and the EGO vehicle is given. Dashed lines indicate irrelevancy and solid relevancy from the perspective of the EGO. Lane change maneuvers show in blue. Due to the two rules, the filter extracts two scenarios shown as dashed orange boxes.

6.2.1.3. Temporal Scenario Context

After the spatial filter, all TOs relevant for the EGO over the whole time of its activity are known. However, a scenario ideally is limited to some traffic behavior's cause and effect [106]. Taking the EGO's whole lifetime into account results in unnecessarily long scenarios with multiple events and again exploding parameter spaces. A subsequent temporal filter solves this problem.

In this proof of concept, lane change maneuvers are considered to be the cause. The effect is whether or not the ego vehicle reacts to this behavior. This reaction is expected to happen within the lane change time frame. Otherwise, it might not be related to the lane change or is not critical. Therefore, a temporal filter proposing the following to rules is applied:

Rule A: A scenario starts with the first maneuver start event and ends with the last ongoing maneuver end event in the relevant environment determined by the spatial filter.

Rule B: A maneuver that is only partially in the relevant environment is always fully included.

These rules shall be briefly explained along with Figure 6.12. The horizontal axis relates to the measurement track's time, while vertically, the TOs are stacked. For each object, dashed lines indicate irrelevancy time intervals and solid relevancy intervals from the EGO's perspective. Additionally, the lines are blue during a lane change maneuver with a bullet indicating its start and an arrow at its end. The first TO remains relevant for the whole track and performs two lane changes. During the first lane change, no other maneuver happens, and the time frame for scenario (1) is set through rule A. The second lane change happens in a more complex environment. Accompanied by a second lane change from TO 2, the start and end time is defined by the start of the first maneuver and end of the last maneuver through rule A. Additionally, the lane change of TO 2 is fully included despite being irrelevant for the EGO at the beginning through rule B. Furthermore, TO 3 is included in scenario (2) as it is relevant during that time. Only TO 4 is not in any scenario as its relevancy interval does not intersect with any maneuver.

With the proposed temporal filter, multiple scenarios can be extracted from a single vehicle's perspective. Based on the spatial filter that considers each recorded vehicle as EGO once, a total of 46677 lane change scenarios can be extracted from the highD dataset. Extracted scenarios only contain what is relevant for the EGO's behavior by limiting the spatial and temporal context. The short scenarios contain the cause and effect of this behavior, satisfying the definitions explained in Section 3.2.2.1 [106].
After converting the scenarios into the JSCEN file format by the use of *openDrive* maps, the extraction process is completed. Before continuing with the actual clustering, the mentioned partitioning into buckets follows next.

6.2.1.4. Partitioning into Scenario Buckets

The definition of a distance measure for scenario clustering enormously benefits from a presorting of all the extracted scenarios into so-called buckets. This is done by dividing them based on their discrete parameters. First, the road offers distinctions such as the number of road lanes and its general topology. As the dataset originates from only six different highway sections, partitioning based on these is a suitable discrete distinction. Furthermore, as every location contains structurally separated ongoing and oncoming lanes that do not influence each other, a further distinctions raises the number to 12 possible track locations. Due to the nature of the coordinate frame in the data set, oncoming and ongoing roads are referred to as upper (u) and lower (l) in the following. The last sorting rule defines through the number of vehicles in a scenario. Thereby, only the paths of the objects remain as dynamic components of the scenario. The partitioning results are now given in Table 6.1. The columns specify the twelve different roads grouped by the six recording locations. First, the upper half of the table provides general information about the locations. The number of road lanes is the first used divergence parameter. The number of recordings performed on a specific location is given below. Most time is spent on location one (loc1), resulting in the most vehicles given in the third row. The following rows divide the buckets below the general information by the number of vehicles, including the EGO ending up in an extracted scenario. According to the temporal filter rules, two vehicles are the minimum, as a TO performing a lane change needs to be present in the scenario. In general, three-lane roads tend to reveal scenarios with more vehicles than two-lane roads. For each of the twelve locations, the number of scenarios is roughly normally distributed. Some buckets are filled with more than a thousand scenarios, while others remain sparsely populated. The few scenarios containing up to 30 vehicles can either be considered outliers or are a result of traffic jams. For clustering, sufficient data is required. Certainly, with more degrees of freedom in a scenario, even more data is required to obtain reasonable results. The bucket on the lower half of location 5 with three vehicles offers a good compromise. While three vehicles on only two road lanes reveal relatively low degrees of freedom in the scenario, the amount of 260 scenarios is both a sufficient start for clustering and overseeable for visual inspections. Hence, this bucket

		loc1		loc2		loc3		loc4		loc5		loc6	
		u	l	u	l	u	l	u	l	u	l	u	l
lanes		3		2		3		3		2		3	
tracks		37		3		3		4		10		3	
total vehicles		69751		2400		2710		3799		8192		2287	
Number of cars in bucket including the EGO vehicle	2	72	76	26	30	46	27	23	23	75	64	11	7
	3	443	447	78	92	170	123	100	82	271	260	59	31
	4	1403	1340	132	117	242	189	182	188	398	452	130	90
	5	2640	2441	124	74	245	208	246	230	327	422	192	119
	6	3695	3594	88	34	180	133	243	207	173	238	190	143
	7	4055	3859	30	6	86	83	180	186	41	79	162	109
	8	3424	2959	3	1	35	35	87	84	11	17	107	59
	9	2222	1582	1	-	6	10	50	31	2	2	42	25
	10	1127	629	1	-	5	3	24	14	1	-	19	8
	11	472	185	-	-	-	-	1	3	-	-	11	-
	12	216	67	-	-	-	-	5	2	-	-	3	1
	13	96	12	-	-	-	-	2	-	-	-	2	-
	14	60	10	-	-	-	-	-	1	-	-	1	-
	15	37	2	-	-	-	-	-	-	-	-	1	-
	16	27	4	-	-	-	-	-	-	-	-	-	-
	17	16	4	-	-	-	-	-	-	-	-	-	-
	18	14	-	-	-	-	-	-	-	-	-	-	-
	19	3	-	-	-	-	-	-	-	-	-	-	-
	20	-	-	-	-	-	-	-	-	-	-	-	-
	21	2	-	-	-	-	-	-	-	-	-	-	-
	22	2	-	-	-	-	-	-	-	-	-	-	-
	23	-	-	-	-	-	-	-	-	-	-	-	-
	24	2	-	-	-	-	-	-	-	-	-	-	-
	25	-	-	-	-	-	-	-	-	-	-	-	-
	26	-	-	-	-	-	-	-	-	-	-	-	-
	27	-	-	-	-	-	-	-	-	-	-	-	-
	28	-	-	-	-	-	-	-	-	-	-	-	-
	29	2	-	-	-	-	-	-	-	-	-	-	-
	30	1	-	-	-	-	-	-	-	-	-	-	-

Table 6.1.: Overview of the number of extracted scenarios sorted into buckets. The columns indicate the locations with the numbers of vehicles depicted by the rows. On the table's top, some general information about the recorded tracks is given.

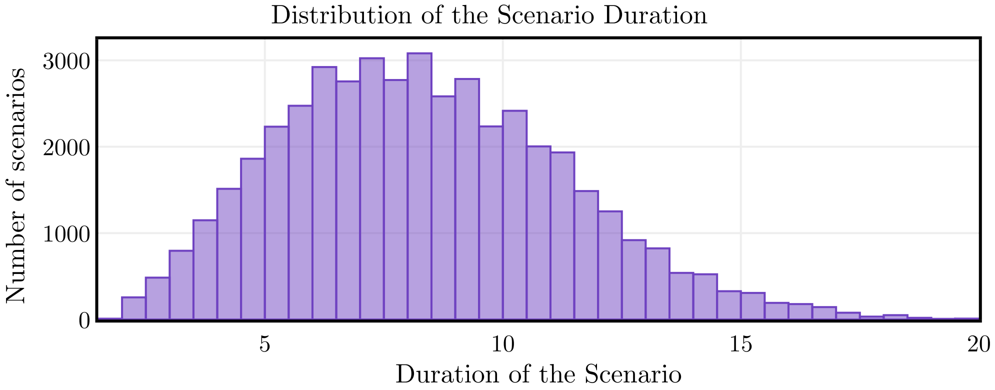


Figure 6.13.: Distribution of the lengths of all extracted scenarios from the *highD* dataset.

is used in the remainder of this section. In the table, it is marked in green. Before starting with the distance measure and the clustering, a few statistics on the extracted data follow.

6.2.2. Analysis of Scenario Data

Based on the extracted data, interesting statistics about scenarios can be derived. These are not necessary for the method in general but can serve as a plausibility and meaningfulness check of the applied extraction and filters.

First, considering the whole recorded trace of the scenario, any given object identified as EGO is accompanied by an average of 30.26 other vehicles during its lifetime. When applying the suggested filter to the data, the number of vehicles is reduced to a relevant share of 6.51 on average. Thereby, the dimensionality of the parameter space for the scenarios is already significantly reduced.

Figure 6.13 shows a distribution of the extracted scenarios length. With a maximum at $\approx 20s$ the average scenario is only about $8.36s$ long. This shows how short a scenario in the median is including cause and effect of some traffic behavior. In contrast, considering an EGO vehicle's whole lifetime as scenario length results in an average $14.38s$ and therefore, significantly bigger scenarios with more parameters.

While iterating over the EGO vehicles, zero to many scenarios extract for a single EGO. Roughly 59.8% do not reveal a scenario as no lane change is present. One scenario is present for 38.1% of the vehicles. Only 2.0% see two and 0.1% more than two scenarios during their lifetime.

Lastly, the tagging of the scenarios can be analyzed. Figure 6.13 displays a distribution of tags over the number of cars within a scenario. Two main statements can be drawn from this diagram. On the one hand, the number of cars in a scenario is normally distributed, as mentioned before. On the other hand, the left and right lane changes are equally present as a vehicle usually enters and leaves a highway on the rightmost lane. Therefore, it must perform an equal number of

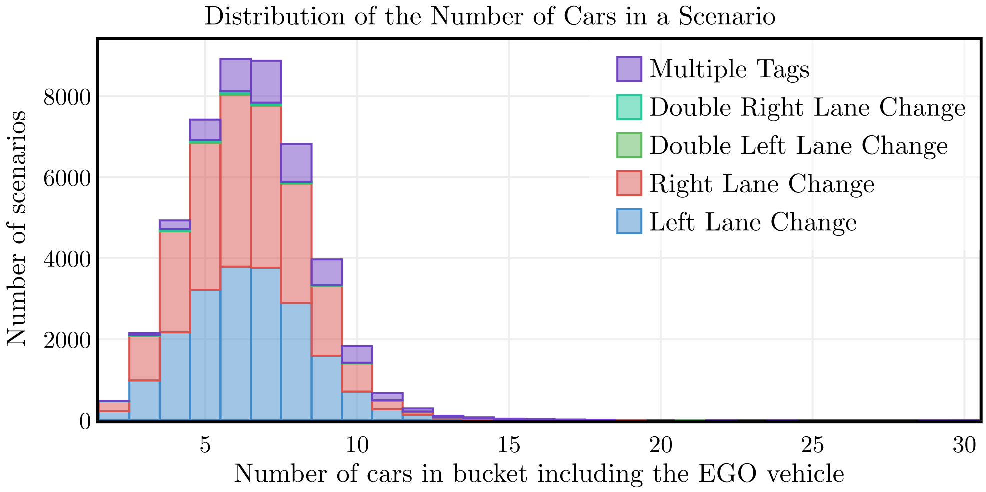


Figure 6.14.: Distributions of the maneuvers in all scenarios extracted from the *highD* dataset.

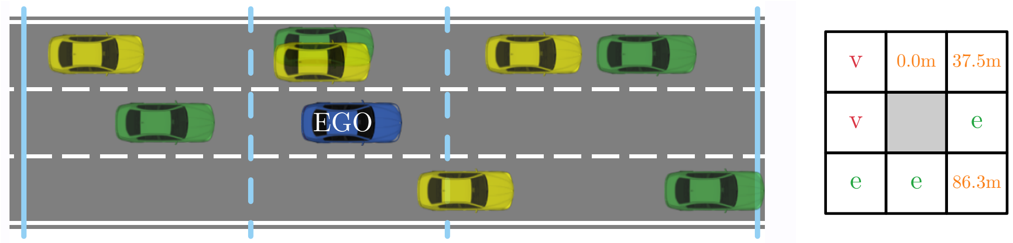
lane changes over its whole highway drive. With a significant amount of observations, this also levels on smaller highway sections as in the dataset tracks. Double lane changes are rare events, observable by the relatively small green and cyan bars. A non-negligible amount of scenarios features multiple different maneuvers, as shown in purple.

Altogether, the suggested scenario extraction reveals scenarios with a minimal number of necessary vehicles and a short period that includes traffic behavior's cause and effect according to a scenario's definition. Also, the tagging of the scenarios reveals plausible results. With the present data, all preconditions for the following distance measure and clustering are set.

6.2.3. Definition of a Scenario Distance Measure for Clustering

The main goal of clustering is finding groups within a large amount of data by some criterion of similarity [188]. In other words, a metric for this similarity is required prior to the actual clustering. This metric is heavily dependent on the type and structure of the underlying data. The definition of such a metric is trivial in an Euclidean feature space as distance. In contrast, it is a non-trivial task for the present scenarios as they contain multiple vehicles with many coordinates on their trajectories of different lengths. Hence, an approach to the definition of a scenario distance measure follows.

Let there be two scenarios S_1 and S_2, that are due to be compared in terms of similarity or distance. As scenarios are time series, the first step is to start this comparison on a scene level or a single step in time.



v	0.0m	37.5m
v		e
e	e	86.3m

Figure 6.15.: Exemplary scenes of scenario S_1 in yellow and S_2 in green within the 8VM for the explanation of the slot distance. On the right, the eight values fo the slots yielded by comparison of those scenes are given.

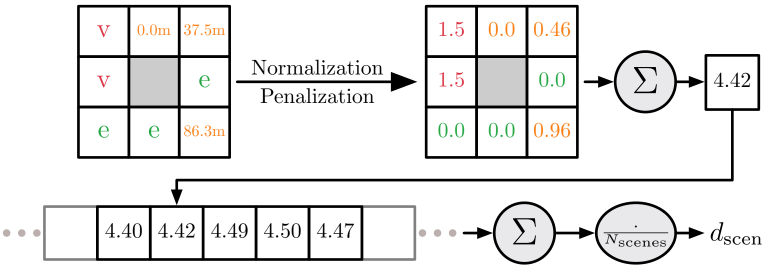


Figure 6.16.: In the upper half, the slot distances from Figure 6.15 are normalized, penalized, and summarized to the scenes distance measure. All scene distance measures consolidate to the scenarios distance measure in the lower half.

6.2.3.1. Slot Distance in the 8-Vehicle-Model

On a scene level, a slot distance defines with respect to the 8VM introduced in Section 6.2.1.2. The model parameterizes in the same way as before. Figure 6.15 shows two exemplary scenes of scenarios S_1 and S_2 in the context of the 8VM in yellow and green, respectively. As the 8VM is EGO-centered, a relative view of the scene from the EGO vehicle's perspective is used in this measure, and scenarios from whole different locations can be compared. This model results in eight slots containing a vehicle for none, one, or both scenes. If none of the scenes yields a vehicle in a particular slot, it is considered empty, as indicated by a green e in the respective field on the figure's right. A red v for vacant denotes the case where only one scenario contains a vehicle in the respective slot. In the last case, where both of the scenarios occupy a slot, the absolute longitudinal distance between the two vehicles with respect to their center denotes.

6.2.3.2. Normalization and Penalization

As the current values entered into the slots do not relate intuitively to the scene's distance due to the empty and vacant labeled cells, normalization and penalization apply. Figure 6.16 shows this process in the upper half. The normalized and penalized values on the right arise from the slot distances from Figure 6.15 on the left as follows. First, the cells containing an actual distance in meters are normalized to the bounds on $[0, 1]$. Therefore, each value is devided by the longest distance possible in the 8VM. This is $95m$ for the chosen parameterization as $100m$ front distance reduces by $5m$ reserved for the adjacent slot right around the EGO. Note that this value normalizes all cells despite the adjacent and rear cells being shorter to avoid different weighting of distances in different cells.

Then remains the question on how to deal with cells labeled as empty or vacant. Indeed, an empty cell is not occupied in any of the two scenes and bears no difference. A value of zero is assigned to those cells. In contrast, a cell where only one scene provides a vehicle is considered as a structural difference in the scene and unwanted in terms of similarity. Therefore, it penalizes with a value of 1.5 even higher than the maximal value achieved by longitudinal distant vehicles. Lastly, a single value of the distance between the two scenes is required. This is simply achieved by adding up all eight cells. Hence, two scenes can be distant on a scale of $[0, 12]$.

6.2.3.3. Consolidation to the Scenario Distance

The distance between scenes is only half of the goal. For the clustering of scenarios, consolidation is required and pictured in the lower half of Figure 6.16. Scenarios pairs are sampled at $5Hz$ while calculating scene distance for every sample. Summing up and dividing through the number of sampled scenes N_{scenes} yields the scenario's distance value. This averaging process is necessary to avoid assigning larger distance values to longer scenario pairs. Again the resulting value range is $[0, 12]$.

A question unanswered to this point is how to deal with the comparison of scenarios of different lengths. As the fixed sampling rate applies to both scenarios, one of them yields more scenes than the other in this case. At the state of the current implementation, the surplus of the longer scenario is ignored in the calculation. However, this approach's effect and if a penalty for time difference benefits the results should be considered further. As for this proof of concept, comparing only the common time frame should suffice.

6.2.3.4. Scenario Distance Matrix for a Bucket and Exemplary Results

The presented scenario distance measure shall now briefly be evaluated with two exemplary scenario pairs. All of these scenarios originate from the chosen bucket of Table 6.1. Therefore, they contain three vehicles, including the EGO, on a two-lane highway. In the upcoming visualizations, the scenarios are superimposed by longitudinally aligning the EGO start position to the arbitrarily chosen marker of $200m$. The driving direction is from left to right.

A pair of quite similar scenarios is shown in the upper half of Figure 6.17. Scenario S_1 marks with a cross at the beginning and end of the associated vehicle's trajectories and S_2 with a bullet. The colors indicate the vehicle with blue for the EGO, green for TO_1, and red for TO_2,

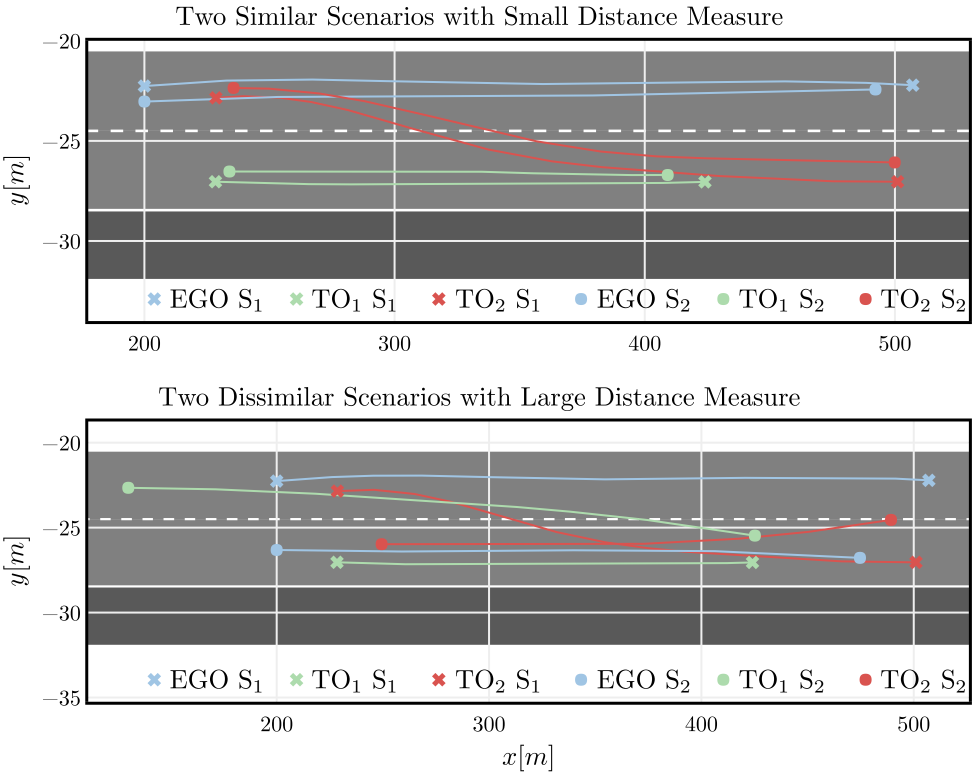


Figure 6.17.: Example scenario tuples with low distance in the upper plot and high distance in the lower plot. Scenario S_1 indicates with crosses at the beginning and end of the associated trajectories and S_2 with bullets. While the EGO is shown in blue, the objects TO$_1$ and TO$_2$ are shown in green and red, respectively. In the upper plot, both scenarios show visually high similarity and are measured with a low distance value of 0.14. In contrast, the lower plot shows two structurally very different scenarios in various manners. Also, the provided measure assigns a significantly higher value of 2.0.

respectively. Note that it is irrelevant for comparing similarities of scenarios if the TOs have the same index for the matching trajectories of both scenarios. In the figure, this is just done manually for visual clarity. The scenario observant in this figure is a TO$_2$ driving in-front of the EGO in the left lane while overtaking another TO$_1$ located in the right lane. After overtaking, TO$_2$ cuts into the right lane in-front of TO$_1$, making room for the EGO to pass. This is the case for both S_1 and S_2. In fact, both scenarios' trajectories are reasonably similar in both their characteristic shape and extent within the common time frame. The presented distance measure assigns a value 0.14, which is at the percentile of 0.77% of most similar scenarios within this bucket.

In contrast, the lower half of Figure 6.17 shows a pair of very dissimilar scenarios. In comparison

TO_2 changes lanes from left to right in S_1 while it initiates a lane change in the opposite direction in S_2. A similar structural difference applies to TO_2 where a lane change is only present in S_2. Additionally, the EGO vehicle drives in different lanes. As expected, the presented distance measure reveals a significantly higher value of 2.0.

Note that this is still at the lower end of the theoretic value range of $[0, 12]$ provided by the distance measure. However, reaching a value of 12 with the provided measure would actually require two scenarios with four vehicles, each residing in distinct cells for the whole time range. Hence, the value must be considered relatively, and upcoming clustering thresholds need to be adapted accordingly.

These two example comparisons show that the presented scenario distance measure assigns a significantly lower value to a visually similar scenario pair than to a pair of structural different scenarios. Certainly, this is only an indicator of correctness and not a proof. However, the overall performance of this measure becomes apparent when applied to clustering in the next step.

6.2.4. Approach to Define a Scenario Space through Hierarchical Agglomerative Clustering

The main goal of this section is the definition of a scenario space through clustering. Therefore, scenarios are extracted and a measure of similarity or distance between them is established. Finally, the actual clustering can be applied.

A vast amount of algorithms and methods [189] for this task are suitable for different types of data and information. The goal of this work is not to find the most suitable method providing universally best results. Instead, it shall show that reasonable results can be achieved through clustering in general, and the suggested procedure is able to provide a definition of the scenario space. Hence, this work chooses Hierarchical Agglomerative Clustering (HAC) [190] without the claim of it being the best fit.

A brief introduction into clustering in general and HAC in detail is given in Appendix B.4. In summary, the approach starts by grouping the available data into one cluster each and joining those clusters iteratively based on the given distance measure. This bottom-up approach builds a hierarchy of links with increasing distance while going upwards. By defining a threshold on this distance, the merging of clusters stops and identifies the desired clusters.

6.2.4.1. Preconditions for Hierarchical Agglomerative Clustering

For the algorithm to work, two contemplations need to be made. First, the definition of distance is heavily data-dependent and crucial for the success of the clustering. Indeed, the previously defined measure from Section 6.2.3 is applied. Second, the algorithm requires a linkage criterion that defines a newly joined cluster's distance to all others (see Appendix B.4 for more information on this). This again imposes requirements on the distance measure used. Ward, centroid, and median drop out as they require the measure to be Euclidean, which certainly is not true for the proposed measure. Other than that, all linkage criteria require non-negativity and symmetry. While evaluating the slot distance in Section 6.2.3.1, the absolute longitudinal distance is considered, which ensures non-negativity. For the same reason, it does not matter if scenario S_1 is compared to S_2 or vice versa, presenting symmetry. Therefore, single, complete, average, and weighted linkage may be used.

6.2.4.2. Distance Matrix as Input for the Clustering

The inputs to HAC are the pre-computed distances between all pairs of scenarios within a bucket. A scenario distance matrix is calculated by comparing all possible combinations of scenarios within this bucket. This matrix has a square and diagonally symmetric character with row and column indices being the scenarios of the bucket as labels. Hence a total number of

$$\binom{N_{\text{scenarios}}}{k} = \binom{260}{2} = \frac{260!}{2!(260-2)!} = 33670 \tag{6.1}$$

measures need to be evaluated with $N_{\text{scenarios}} = 260$ as the number of scenarios in the buckets and $k = 2$ as the number of scenarios in the pair, based on the Binomial Coefficient. Hence, the computation of the distance matrix has $\mathcal{O}(N^2_{\text{scenarios}})$ complexity.

6.2.4.3. Application to the Scenario Bucket

The precomputed distance matrix now serves as input to the HAC algorithm. The dendrogram visualizes the outcome in Figure 6.18. On the vertical axis of this plot, the cluster distance is calculated with complete linkage while the clusters locate along the horizontal axis. At the bottom-most level, each scenario resides within its own cluster. Moving upwards clusters are merged depending on their distance. At a topmost level, all scenarios end up in the same cluster. The trick is no to define a threshold on the cluster distance for which merges are not accepted anymore. Setting this threshold too high causes scenarios of structurally different types remaining in a cluster while a too low threshold can produce multiple clusters of the same type. Through an experimental process of setting a threshold and visually examining the scenarios residing in the resulting clusters, a value of 0.7 for this bucket can be determined. The threshold indicates as a dashed horizontal line in Figure 6.18. Additionally, the 77 resulting clusters are colored and marked along the horizontal axis. A visualization of exemplary clusters follows next.

6.2.4.4. Cluster Evaluation

As mentioned in the last section, a suitable threshold can be determined by visual inspection of the members within a cluster at a certain level. For the chosen value of 0.7, the two largest

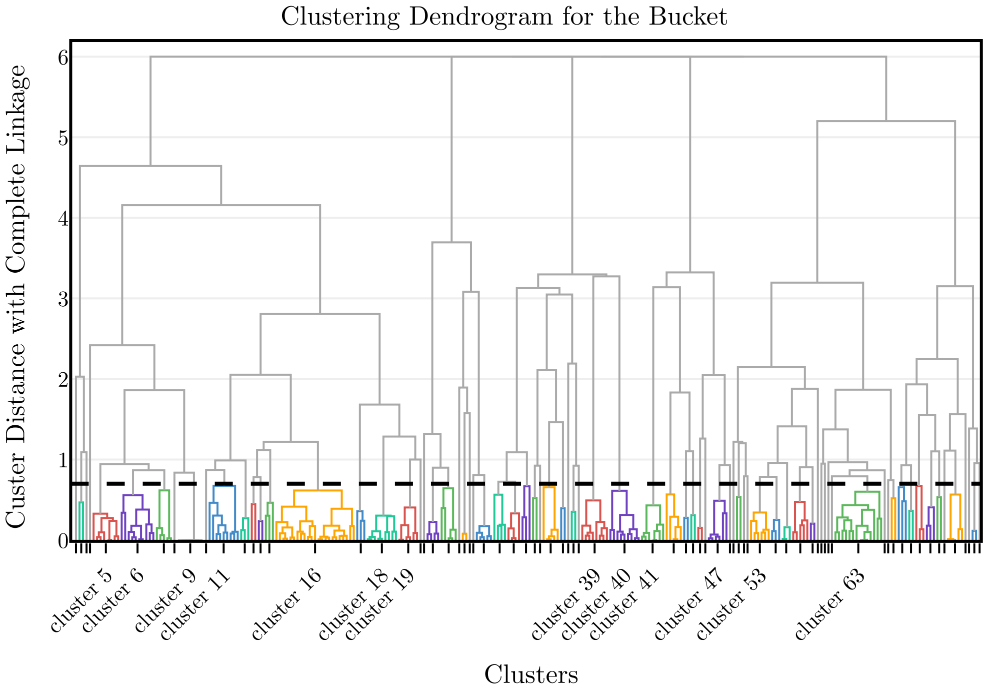


Figure 6.18.: Dendrogram of the HAC of the bucket on lower location five with three vehicles on a two-lane highway. The 260 member scenarios of this bucket are hierarchically merged based on their distance from bottom to top. The threshold of 0.7 up to which cluster merges are accepted is shown as a dashed horizontal line.

clusters 16 and 63 are plotted in Figure 6.19 at the top and bottom, respectively. Within both plots, the EGO vehicle shows in blue and both TOs in green. Having both TOs in the same color is due to the fact that neither the distance measure nor the structure of a scenario cares about the indexing of TOs.

The upper plot shows the familiar scene. The EGO on the left lane overtakes a TO on the right while following another to TO cutting into the right lane. Previously, two members of this cluster are used to evaluate the distance measure. All members show the same scenario structure, with the most variation being observable in one of the TOs' cut-in. The lower plot shows the same structural scenario but with the EGO vehicle switching place with the TO being overtaken. In both cases, the method is successfully able to find clusters of scenarios with a similar structure. It is important to notice that in both of the shown clusters, technically, the same happens. However, for the definition of scenarios as test cases within a scenario space, the EGO perspective matters. Hence, the differentiation of those two clusters is valuable.

From both the dendrogram in Figure 6.18 and the visualizations in Figure 6.19, it can be seen that the method is able to cluster the given bucket and that the cluster's members are indeed

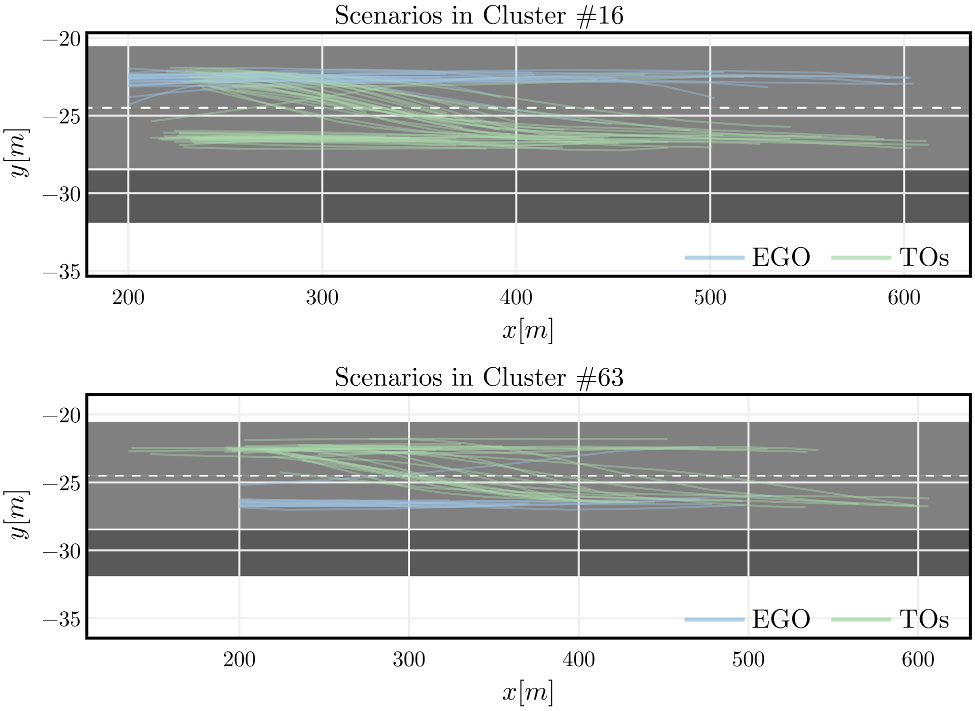


Figure 6.19.: Scenarios in cluster #16 and #63 plotted above each other. The EGO shows in blue, while all other TOs are green.

similar. However, the fact that there are 77 clusters grouped from 260 scenarios and some of them even containing only one member is not the result that defines the scenario space of a two-lane highway with three vehicles as a whole. This only shows how much combinations are still possible and how much data is actually required to achieve the goal. The vast amount of 44.500 driven kilometers in the *highD* dataset is not enough data to cope with the permutations possible in both static parameters as road geometry and the number of vehicles and the dynamic parameters within the vehicle's trajectories. This is a paramount example of the Curse of Dimensionality [191]. Moreover, that is only for highway scenarios as of now.

6.3. Final Thoughts about the Test Space and its Coverage

So far, a proof of concept for both a scenario space definition and variation of scenarios is available with a statement about test coverage to be made. Already the state of the art from Section 3.3.4.3 concludes, that there is not enough data available yet to conclude on a test coverage in a scenario-based assessment [157]. Unfortunately, the previous section confirms this statement. Even a very limited two or three lanes highway environment still allows for so many structurally different scenarios that only a two-digit number of the 46677 extracted scenarios end up on average in a cluster. Hence, this section does not aim to challenge the number of

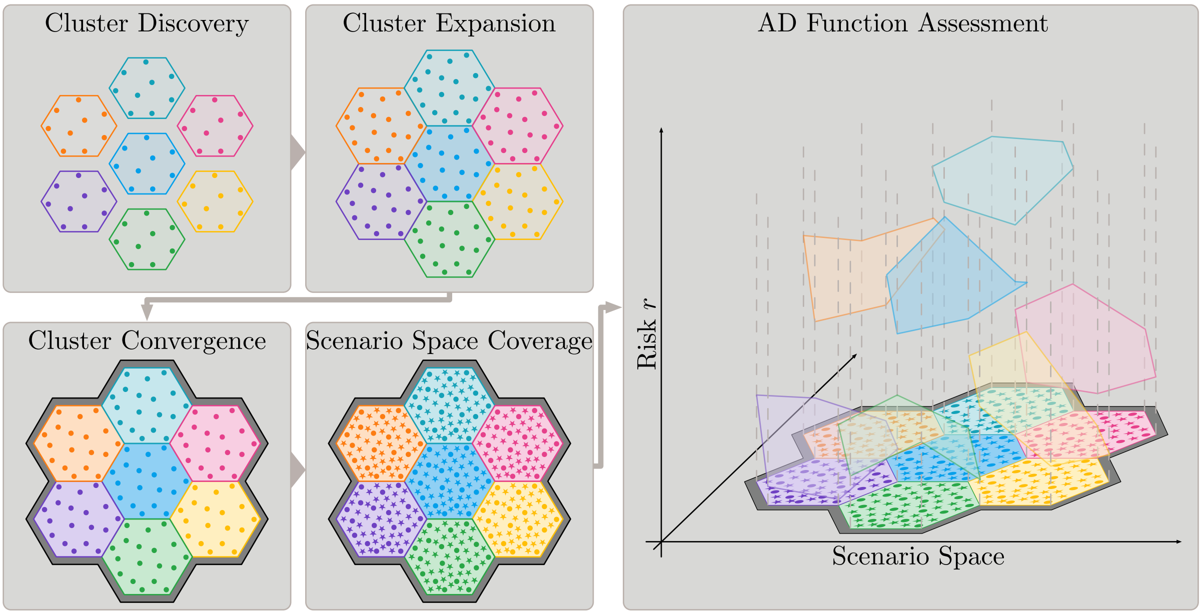


Figure 6.20.: The exploration and coverage of the scenario space with presented methods in five steps. A scenario cluster is visualized as individually colored hexagon and its member scenarios as bullets and starts with the same color depending on their origin in real or simulated data, respectively.

$6 \cdot 10^{10}$ scenarios necessary for a scenario-based assessment approach on german highways given in [157]. Instead, this section aims to explain how both the scenario space and variation can be used together for the coverage of the scenario space within the assessment method presented in this thesis. Further, the influence of the data basis on the coherence of the scenario space itself is discussed. Due to data insufficiency for an empirical study, this section's remainder is based on theoretical thoughts and remains unvalidated.

6.3.1. Integration with the Assessment Method

The whole process shall be explained along with Figure 6.20. We assume a separate scenario space for each bucket, hence for every structural difference of road geometry or scenario components such as the number of vehicles. For the sake of simplicity, a few simplifications are made in the illustrations. First, the scenario space is shown in two dimensions while certainly, it is of much higher dimensionality. Then, clusters are shown as simple hexagons within the bucket scenario space. In reality, those geometries are more complex. Also, the transition between clusters in this space displays as continuous and gapless with respect to the scenario parameters. However, if this holds with real data is questionable. Lastly, there are only seven clusters shown in a bucket, while there are certainly more. Nevertheless, all the simplifications enable

an explanation of the following concept.
Figure 6.20 is structured as follows. The four steps on the left explain the scenario space's discovery and coverage, and the right its integration with the presented assessment methodology. Each of the seven clusters is given a unique color, while its members' density rises with increasing opacity. The members themselves are shown as bullets for real-world measurements and stars for variations in a virtual test domain. The building blocks explain as follows:

- **Cluster Discovery**: In the beginning, there are only real-world test drives with the AD function. With this data, new clusters are discovered and sparsely filled with members.
- **Cluster Expansion**: With a growing amount of driven kilometers and therefore scenarios, the clusters expand, and density increases.
- **Cluster Convergence**: Eventually, the expansion stops, and the clusters converge. At this stage, the local space within a bucket and its extent is known. The grey outline in the cluster convergence block indicates this total scenario space. On a side note, this already allows for occurrence calculation of scenarios and outlier detections in the form of often referenced edge case scenarios.
- **Scenario Space Coverage**: At this point, the assessment methodology from Chapter 5 comes into the picture. The available real-world scenarios can be re-evaluated in a virtual test domain, such as simulation. Suppose a local cross-verification of those is established. In that case, variations of those scenarios and test execution in simulation can dramatically increase the test coverage and, hence, the scenario space's density to the desired level.
- **AD Function Assessment**: The only thing remaining with sufficient locally verified and tested scenarios available is the assessment with respect to the AD function's performance and safety. The right half of Figure 6.20 shows a risk as exemplarily calculated through the measure in Chapter 4 as a third dimension pointing upwards. A risk hyper-surface forms above the scenario space and shows the risk a passenger is exposed to while using the AD function separated by the clustered scenarios. In this hypothetical outcome, the function performs well in the purple cluster with overall low risk and worse in the light blue cluster with high risk.

Consequently, the release can then be achieved in two ways. Either the AD function improves iteratively until the risk is low across all scenario spaces relevant for the ODD, or the ODD must be limited to spaces with acceptable risk. Suppose it is eventually possible to reach that level of information. In that case, the final statement can be made: *The AD function is tested against a set of scenarios we trust to represent the entirety of scenarios within the ODD at enough confidence, and the overall risk across these scenarios is low enough to release the function with enough confidence.*

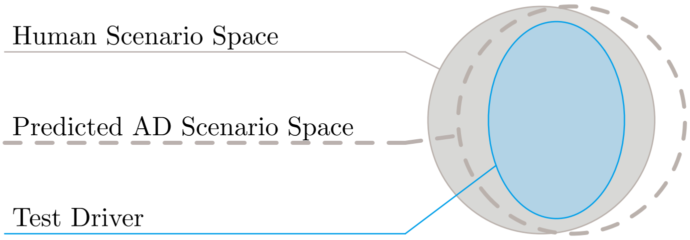


Figure 6.21.: Influence of Driving Data on the Scenario Space explained along with a Venn diagram. The light circle encompasses the entirety of traffic scenarios a human driver can be exposed to while the dashed gray circle explains the same for an AD function. The blue subset shows the limited space that can be explored by a test driver in a prototype vehicle.

6.3.2. Influence of the used Driving Data on the Scenario Space

In Section 6.2, naturalistic driving data is used for the derivation of a scenario space. In contrast, the integration of the method in Section 6.3.1 describes the use of closed-loop data of a vehicle equipped with the AD function under test. Indeed, as in the first case through ordinary fleet vehicles, data acquisition is more straightforward, less expensive, and less risky as there are no prototypes involved. Therefore, the question is raised if there is actually a difference in the scenario space when derived by either of these sources.

The Venn diagram in Figure 6.21 helps with answering this question. Let the light grey-filled circle be the entirety of driving scenarios a human can observe when driving in any kind of environment for an infinite amount of time. It is expected that an AD function has an impact on the scenario space itself. In fact, the effect of AD on traffic flow is in the focus of current research [192, 193, 194, 195]. Hence, the traffic space, including automated vehicles, is expected to shift, as illustrated through a grey dashed circle. Certainly, this is the space where AD functions should be assessed, as they are exposed to real-world traffic. Thus, the integration in the prior section starts with data collection from closed-loop drives. However, it is more expensive, time-consuming, and riskier to go this path as prototype vehicles and test drivers are involved. Furthermore, due to special training and time limitation of test drivers, it remains questionable if they are able to explore the whole space. For that reason, their traffic space shown in blue is only a subset. Which way to go for the definition of a scenario space ultimately remains a question on how big the difference in human and AD traffic space really is, and further research is necessary.

6.4. Conclusion

This chapter provides insight into the two missing components, scenario variation and scenario space definition, from Chapter 5.

First, a scenario variation technique based on PCA and data from NDS_1 is proposed. This approach varies individual traffic objects on a maneuver level and is therefore independent of the general scenario structure. Due to the derivation of the PCA scenario kernels from naturalistic data, also the variations are naturalistic and physically drivable. The method can produce local variations based on an original scenario, as expected by the assessment method. Validation of the results is presented experimentally by generating, simulating, and assessing 500 variations of a proving ground scenario. Both the inspection of the position, velocity, and heading traces, combined with the resulting risk distribution, show that the approach preserves the locality. However, due to the input data being only recorded on highways, an extension of this variation method to other ODDs is postponed to future work. Also, a limitation to only highway traffic participants is bound to the limits of this data set. Furthermore, the variation of other scenario components than vehicle trajectories is to be elaborated.

The second section addresses the definition of a scenario space. Primarily, the NDS_1 data splits into a large set of highway driving scenarios through a custom proposed spatio-temporal filter. After presorting those into buckets based on structural differences, a custom distance measure calculates the similarity between any scenario pair of a bucket. This can finally suffice as an input to HAC. Visual inspection of the scenarios shows that this method is successfully able to group similar scenarios together. In particular, the applied extraction can gather comparable instances as scenarios from a large amount of data. The distance measure is able to assess similarity, and HAC is suitable for the final clustering. However, this approach confirms the assumption that to this date, there is not enough data for the definition of a global scenario space and, therefore, scenario-based assessment available [157]. This embodies through low populations within a cluster and many singleton clusters. Also, this presented method is currently limited to the ODD of highway scenarios to the chosen data basis. Future work should also address the usage of more advanced clustering methods, as currently, much manual parameterization is necessary.

The third and last part of this chapter covers the synergy between presented scenario variation and scenario space definition and the assessment methodology explained in Chapter 5. It shows how real-world drives gradually explore the scenario space and variations conducted in virtual test domains increase coverage to the desired level. Due to the absence of sufficient data, this remains a theory. Validation of the findings remains infeasible and out of scope at this point. Further, the influence of the databasis on the scenario space is elaborated. The NDS_1 data can either be from human drivers or closed-loop test drives with AD-equipped prototypes. Depending on this, the outcome of the scenario space may vary, and different caveats arise. The answer which to way to go is part of future work as soon as sufficient data is available.

Despite the constraints given and the lack of data, this chapter shows that the derivation and traversing of a scenario space are possible on a proof of concept level. Further, it can be well integrated into the assessment methodology from the previous chapter. Therefore, local cross-verification and scenario variation can leverage the data demand for sufficient test coverage. Hence, this chapter confirms the last hypothesis: *A scenario space and its coverage can be defined with local clusters and traversed through local variation with verified results.*

7. Conclusion and Future Work

The preceding pages of this thesis elaborate on some novel concepts that aim to resolve issues regarding AD assessment. More precisely, the infeasible amount of testing required when applying previous assessment methodologies to AD shall diminish through the application of scenario-based testing as well as the inclusion of locally cross-verified virtual testing. For the latter's achievement, the current state of the art misses several modules that this work addresses. This chapter summarizes the conclusion of the individual topics addressed by revisiting the contribution to knowledge, the aim and objectives, and the findings' limitations. Suggestions for continuative research topics finalize the thesis.

7.1. Conclusion

Each of the last three chapters represents a pillar in this thesis's structure (see Section 2.4) and answers one research question featuring its conclusion. At this point, the summary shall present a roundup of the contributions to knowledge and its limitations. This summary then serves to justify the achievement of the objectives that constitute this thesis. It aims at the bigger picture, while for a more detailed and closer-to-theory conclusion, the individual chapters' conclusions suffice.

7.1.1. Summary of the Contribution to Knowledge

The contribution to knowledge is anticipated in Section 2.5 and revisited here with the gained technical detail.

Among the present literature, a lack of suitable assessment metrics for AD's safety performance is observable. Existing metrics are either oversimplified, suffer from greater overestimations or do not meet the requirement of avoiding false negatives. By answering the thereby motivated research question, this thesis introduces a novel data-driven scenario risk assessment metric. A naturalistic scene prediction ensures a realistic view of the risk in a given scene. Further, the stochastic consolidation on reaction times to possible accidents offers a percental value of risk closer to real accident risk than given metrics. An only slight overestimating characteristic for the avoidance of false negatives is mathematically proven. A subsequent experimental validation rounds up the development of an applicable accident risk metric for the AD assessment.

Simple statistical considerations reveal that current assessment approaches applied to ADAS are stretched to their limits. Recent research puts much effort into new scenario-based approaches, but none of them pushes beyond mere theory. Motivated by the second research question, this thesis pushes a novel concept utilizing both scenarios as test cases and virtualized testing towards a proof of concept state. It is achieved by building a re-simulation pipeline that extracts scenarios from real-world data and enables re-simulation as well as assessment with the introduced metric from the first research question. The availability of an end-to-end pipeline for scenario-based assessment constitutes the enabler for research on scenario-based assessment beyond mere theory. Consequently, the influence of GT and different error sources in such a pipeline on the cross-verification of simulation can be investigated. An in-depth investigation of these errors reveals the deal-breaker of cross-verification and concludes in feasible future approaches. Notably, the absence of accurate perception modeling motivates further research beyond this thesis.

The necessity of a scenario space definition and scenario variation tooling for measurability and improvement of coverage is a direct result of the suggested assessment methodology and forms the third research question. Literature research in both fields reveals sparse results that further are not adequate for this method. Hence, this work first provides a novel concept for scenario-based variation. Again a data-driven approach utilizes a large amount of naturalistic driving trajectories to create PCA-based mapping functions for driving maneuvers. This approach enables the generation of naturalistic and drivable local variations of a given trajectory and fits well into the overall assessment concept. Further, the definition of the scenario space is addressed. A custom distance measure on a large amount of naturalistic driving data enables the clustering of scenarios into subspaces, where parameterization is then possible. Finally, both the scenario space and the variation combine in a novel concept for defining and increasing coverage. Thereby, completeness of the assessment on a theory level due to a shortage of data is observant.

In general, this thesis provides a concept for the assessment of AD from theory to proof of concept implementation. The findings help to cope with the assessment problem existing today. Indeed, a proof of concept state has its limitations, which follow next.

7.1.2. Limitations of the Findings

The limitations are pointed out throughout this thesis's body but are summarized concisely within this chapter. As several limitations appear throughout the whole work, they are categorized by type rather than by research question.

As for this proof of concept, all considerations narrow down to a highway scope. For the first iteration of AD, this might be sufficient, but further development towards SAE level 5 requires an extension to rural and urban areas as well. Within the assessment metric, the prediction model is simple enough to suit highway scenarios. An extension would involve the inclusion of more complex, maybe maneuver-based or interaction-aware models with a different parameter set. Also, the methodology itself is only tested against highway scenarios. Lastly, both the data-driven scenario space and variation are based on a highway-only dataset. Therefore they are only applicable to the same extent. As rural and urban areas are by far more complex than highways, additional challenges are predictable with these extensions. However, these do not directly relate to the assessment methodology itself. Hence, they remain out of scope for this thesis.

Another limitation of the current implementation is the choice of represented road users. Currently, throughout the whole work, only cars are considered. Other traffic participants, such as trucks and motorcycles that also follow this bicycle-like movement, can easily be integrated by training a different set of parameters and PCA kernels in the assessment metric and variation, respectively. For traffic participants that follow a less restricted movement, such as pedestrians, also the spline representation in the scenario description needs to be reviewed in terms of representability. These extensions count towards the completeness of the implementation. Therefore, they are beyond the proof of concept and claim of this work.

For now, the mismatch between GT and SD representation due to missing perception models is a deal-breaker for the local cross-verification of the simulation against real-world test drives. It causes the simulation to fail reaching the desired precision in re-simulation. Hence, improvements to the simulation itself are necessary. However, both the simulation itself and the modeling of AD perception are not in this thesis's focus.

Lastly, all data-driven approaches used in this work are heavily reliant on the underlying data regarding their success. This is particularly crucial in the definition of the scenario space and the variation. Limitations arise not only from the data source restricted to the highway-scope, as mentioned earlier, but also the amount of data is currently a problem. Even with the seemingly large amount of extracted scenarios, the immense diversity of possible structural different scenarios on highways causes sparse and event singleton clusters. Completeness of the clusters with respect to possible scenarios can currently not be expected. Consequently, the subsequent concept for coverage can only remain a theory. However, the applied methods have shown their principal functionality on the given data, and the acquisition of a larger scale of data can not be achieved within such a thesis.

A proof of concept always has its limitations. On the contrary, it is one goal of such concepts to detect such limitations early on and suggest remedy. All of the limitations listed here are not impossible to solve. Hence, the general concept is still viable.

7.1.3. Achievement of the Objectives

Having the contributions to knowledge and their limitations confronted is the ideal place for checking the given objectives' achievement. These objectives are fundamental to this thesis and given in Section 1.2.

- **Make the safety of an AD function measurable:** Motivated through the first research question, Chapter 4 treats this task. For that, the aforementioned assessment metric is created and provides reasonable results with respect to accident risk in the given test scenarios. It is limited to highway scenarios in the current implementation, but possible extensions and their feasibility are stated. For the proof of concept scope of this thesis, it suffices to achieve the first objective.

- **Elaborate a methodology for the assessment of AD:** A complete assessment methodology elaborates and consolidates from existing scenario-based theories in Chapter 5. It does not remain on a theoretical level but instead results in a complete framework allowing further research. All modules are implemented as proof of concept and experimentally validated. The remaining caveats are investigated, and solution suggestions follow. The deal-breakers identified are solvable but out of the scope of this work. Hence, the second objective is achieved.

- **Provide tooling to show the completeness of the assessment:** Lastly, the thesis should provide tooling to show the test coverage. Chapter 6 splits this task into the definition of a scenario space and the variation of scenarios to make coverage measurable and improve it, respectively. For both, a promising method is elaborated and experimentally validated. The synergy between both to measure and extend coverage and therefore show completeness of the assessment is discussed. Due to the lack of a sufficient amount of data, the latter remains a profound theory resulting from prior findings. As the collection of this data can not be the scope of such a thesis, the last objective can be seen as achieved with the inexpugnable limitations.

Given three objectives linked to the problems of the assessment of AD, this thesis derives three research questions and transforms them into three hypotheses. It shows that the addressed problems are indeed solvable, while subsequent tasks and challenges remain. While the given objectives are considered to be achieved, the remainder to achieve the superior objective of ultimately assessing an AD shall be summarized next.

7.2. Future Work

Research is, especially in recent upcoming topics, an ongoing process. Every new step achieved results in new research topics and open questions until the ultimate goal of a field can be solved. The situation is no different with this work. While the findings bring scenario-based assessment one step closer to success, additional questions arise, and future work remains. The more significant part of future work arises throughout this work's development and is a direct result of the limitations of the findings summarized in Section 7.1.2.

For the highest maturity level of AD (SAE 5), an extension of the modules and unlimited ODD with rural and urban environments is inevitable. This involves the prediction models used in the assessment metric, the scenario space definition, and variation. For completeness, not only all environments have to be considered, but also all kinds of traffic participants. In addition to the aforementioned modules, this also involves revisiting the description of the scenarios. Conditioned through the modularity of the framework and the modules itself' constitution, a deal-breaker for those extensions is not observant, and feasibility is given.

To include virtual test domains such as simulation into the assessment process, the re-simulation accuracy is crucial. Without sufficient accuracy, local cross-verification is not achievable. Concerning this, the development of perception models bridging the gap between GT and SD views in the simulation is inevitable. Other than that, the simulation raises the question, how deep modeling depth needs to be achieved cross-verification.

The last and presumably most time-consuming and costly future work originating from this work is the acquisition of a sufficient data basis for the data-driven modules. For the completeness of the assessment, this data basis must fulfill the same condition. Only with this precondition met, the definition of a scenario space and its coverage through variation can provide profound statements about the AD's global safety. Including urban, rural, and highway environments, this can be the most challenging task. It is evident that this objective remains to be achieved by the OEMs as only they possess the infrastructure and carpool for such a large-scale data collection.

The suggested scenario-based methodology promises to put relief on the assessment of AD through the inclusion of virtual test domains. This statement reinforces by a whole proof of concept framework, experimental validation, and analysis. This work shows that such an approach is indeed possible by providing an implementation of all necessary modules. Still, the achievement of the ultimate goal of an ironclad assessment methodology that is both efficient and reliable is not done yet. Likewise, every scientific research and industrial endeavor, also this thesis makes its contribution to this achievement. Together with all ongoing efforts, this task is to the author's best belief, possible in the foreseeable future.

Bibliography

[1] M. Guarnieri, "The roots of automation before mechatronics [historical]," *IEEE Industrial Electronics Magazine*, vol. 4, no. 2, pp. 42–43, 2010.

[2] J. Horn, L. N. Rosenband, and M. R. Smith, *Reconceptualizing the Industrial Revolution.* MIT press, 2010.

[3] W. D. Devine, "From shafts to wires: Historical perspective on electrification," *The Journal of Economic History*, vol. 43, no. 2, pp. 347–372, 1983.

[4] K. F. Benz, "Fahrzeug mit Gasmotorenbetrieb," Patent DE37435C, Nov. 1886.

[5] D. F. Prindle and D. F. Prindle, *The Paradox of Democratic Capitalism: Politics and Economics in American Thought.* JHU Press, 2006.

[6] F. Kröger, "Das automatisierte Fahren im gesellschaftsgeschichtlichen und kulturwissenschaftlichen Kontext," in *Autonomes Fahren*, pp. 41–67, Springer, 2015.

[7] A. Bartels, "Roadmap Automatisches Fahren," in *Braunschweiger Symposium Automatisierungs-, Assistenzsysteme Und Eingebettete Systeme Für Transportmittel*, (Braunschweig, Deutschland), pp. 350–364, Gesamtzentrum für Verkehr (GZVB), Feb. 2008.

[8] C. Thorpe, M. H. Hebert, T. Kanade, and S. A. Shafer, "Vision and navigation for the Carnegie-Mellon Navlab," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 3, pp. 362–373, 1988.

[9] T. Kanade, C. Thorpe, and W. Whittaker, "Autonomous Land Vehicle Project at CMU," in *Proceedings of the 1986 ACM Fourteenth Annual Conference on Computer Science*, CSC '86, (New York, USA), pp. 71–80, ACM, 1986.

[10] "Technology Readiness Levels Handbook for Space Applications," tech. rep., European Space Agency, Sept. 2008.

[11] "ERTRAC - Automated Driving Roadmap," tech. rep., European Road Transport Research Advisory Council, July 2015.

[12] E. Guizzo, “How google’s self-driving car works.” https://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works, Oct. 2011.

[13] R. J. Rosen, “Google’s Self-Driving Cars: 300,000 Miles Logged, Not a Single Accident Under Computer Control.” https://www.theatlantic.com/technology/archive/2012/08/googles-self-driving-cars-300-000-miles-logged-not-a-single-accident-under-computer-control/260926/, Aug. 2012.

[14] D. Muoio, “Bosch’s self-driving car prototype could give us a glimpse of Tesla’s Autopilot plans.” http://www.businessinsider.de/bosch-driverless-tesla-autopilot-2016-10, Oct. 2016.

[15] F. Lambert, “Tesla CEO Elon Musk: self-driving will encompass all modes of driving by the end of next year.” https://electrek.co/2018/03/11/tesla-ceo-elon-musk-self-driving-next-year/, Mar. 2018.

[16] R. Grosspietsch, “Investor Presentation: BMW Highly Automated Driving,” *Presentation at Barclays Investor Conference, London, England*, Sept. 2015.

[17] G. Prodhan, “BMW says self-driving car to be Level 5 capable by 2021.” https://www.autonews.com/article/20170316/MOBILITY/170319877/bmw-says-self-driving-car-to-be-level-5-capable-by-2021, Mar. 2017.

[18] D. Lee, “Toyota sneak previews self-drive car ahead of tech show.” http://www.bbc.com/news/technology-20910769, Jan. 2013.

[19] J. Becker, “Toward fully automated driving,” *Presentation at TRB Second Workshop on Road Vehicle Automation, Stanford, USA*, 2013.

[20] H. Winner and W. Wachenfeld, “Absicherung automatischen Fahrens,” *Presentation at 6. Tagung Fahrerassistenz, Munich, Germany*, 2013.

[21] International Organization for Standardization, “ISO/DIS 26262:2018 - Road vehicles - Functional safety,” Dec. 2018.

[22] I. Vorndran, “Unfallstatistik - Verkehrsmittel im Risikovergleich,” tech. rep., Statistisches Bundesamt, Dec. 2010.

[23] M. Peden, R. Scurfield, D. Sleet, D. Mohan, A. A. Hyder, E. Jarawan, and C. D. Mathers, *World Report on Road Traffic Injury Prevention*. World Health Organization Geneva, 2004.

[24] R. Lozano, M. Naghavi, K. Foreman, S. Lim, K. Shibuya, V. Aboyans, J. Abraham, T. Adair, R. Aggarwal, S. Y. Ahn, *et al.*, “Global and regional mortality from 235 causes of death for 20 age groups in 1990 and 2010: A systematic analysis for the Global Burden of Disease Study 2010,” *The Lancet*, vol. 380, no. 9859, pp. 2095–2128, 2012.

[25] "ERTRAC Road Transport Scenario 2030+," tech. rep., European Road Transport Research Advisory Council, Oct. 2009.

[26] "Decade of Action for Road Safety, 2011–2020: Saving millions of lives," tech. rep., World Health Organization, 2011.

[27] E. Papadimitriou, G. Yannis, F. Bijleveld, and J. a. L. Cardoso, "Exposure data and risk indicators for safety performance assessment in Europe," *Accident Analysis & Prevention*, vol. 60, pp. 371–383, 2013.

[28] "Verkehrsunfälle - Fachserie 8 Reihe 7 - 2010," tech. rep., Statistisches Bundesamt, July 2011.

[29] "Verkehrsunfälle - Zeireihen 2015," tech. rep., Statistisches Bundesamt, Oct. 2016.

[30] "Verkehrsunfälle - Zeitreihen 2018," tech. rep., Statistisches Bundesamt, July 2019.

[31] U. Kunert, S. Radke, B. Chlond, and M. Kagerbauer, "Auto-mobilität: Fahrleistungen steigen 2011 weiter," *DIW-Wochenbericht*, vol. 79, no. 47, pp. 3–14, 2012.

[32] T. Unselt, J. Breuer, L. Eckstein, and P. Frank, "Avoidance of "loss of control crashes" through the benefit of ESP," in *Proceedings of the 30th FISTA World Automotive Congress*, (Barcelona, Spain), May 2004.

[33] N. Giesen, "20 Prozent weniger Auffahrunfälle durch DISTRONIC PLUS und Bremsassistent PLUS," tech. rep., Mercedes Unfallforschung, June 2010.

[34] S. Karush, "They're working: Insurance claims data show which new technologies are preventing crashes," *Status Report (Insurance Institute for Highway Safety)*, vol. 47, no. 5, 2012.

[35] "WHO — Global Plan for the Decade of Action for Road Safety 2011-2020," tech. rep., World Health Organization, Mar. 2010.

[36] "Public Consultation on an EU Strategy to Reduce Injuries Resulting from Road Traffic Accidents," tech. rep., European Commission, 2012.

[37] W. Wachenfeld and H. Winner, "Die Freigabe des Autonomen Fahrens," in *Autonomes Fahren*, pp. 439–464, Springer, 2015.

[38] D. L. Hendricks, M. Freedman, and J. C. Fell, "The relative frequency of unsafe driving acts in serious traffic crashes," tech. rep., United States National Highway Traffic Safety Administration, Jan. 2001.

[39] "National Motor Vehicle Crash Causation Survey," tech. rep., National Highway Traffic Safety Administration, U.S. Department of Transportation, Springfield, Virginia, July 2008.

[40] S. Singh, "Critical reasons for crashes investigated in the national motor vehicle crash causation survey," tech. rep., National Highway Traffic Safety Administration, U.S. Department of Transportation, Feb. 2015.

[41] Bundesanstalt für Straßenwesen, "Volkswirtschaftliche Kosten durch Straßenverkehrsunfälle in Deutschland 2004," *BASt Info*, vol. 2, no. 6, 2010.

[42] "Where are we heading? Paths to mobility of tomorrow. The 2018 Continental Mobility Study," tech. rep., Continental AG, Dec. 2018.

[43] R. Felkai and A. Beiderwieden, "Schaffen allgemeiner Voraussetzungen der Projektabwicklung," in *Projektmanagement Für Technische Projekte*, pp. 4–44, Springer, 2011.

[44] H. Winner, M. Graupner, and W. Wachenfeld, "How to Address the Approval Trap for Autonomous Vehicles," Nov. 2015.

[45] 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.

[46] F. Schuldt, F. Saust, B. Lichte, M. Maurer, and S. Scholz, "Effiziente systematische Testgenerierung für Fahrerassistenzsysteme in virtuellen Umgebungen," in *Automatisierungssysteme, Assistenzsysteme Und Eingebettete Systeme Für Transportmittel*, ITS mobility e.V, 2013.

[47] K. Groh, T. Kuehbeck, M. Schiementz, and C. Chibelushi, "Towards a Scenario-Based Assessment Method for Highly Automated Driving Functions," in *8th Conference on Driver Assistance*, (Munich, Germany), Sept. 2017.

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

[49] T. Form, "PEGASUS Method for Assessment of Highly Automated Driving Function," *Presentation*, Nov. 2018.

[50] M. Tatar and J. Mauss, "Systematic test and validation of complex embedded systems," in *Embedded Real Time Software and Systems*, (Toulouse, France), Feb. 2014.

[51] T. Helmer, L. Wang, K. Kompass, and R. Kates, "Safety Performance Assessment of Assisted and Automated Driving by Virtual Experiments: Stochastic Microscopic Traffic Simulation as Knowledge Synthesis," in *Proceedings of the 2015 IEEE International Conference on Intelligent Transportation Systems (ITSC)*, (Las Palmas de Gran Canaria, Spain), pp. 2019–2023, Sept. 2015.

[52] D. Levinson, "On the Differences between Autonomous, Automated, Self-Driving, and Driverless Cars." https://transportist.org/2017/06/29/on-the-differences-between-autonomous-automated-self-driving-and-driverless-cars/, June 2017.

[53] "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," Tech. Rep. J3016, SAE international, Sept. 2016.

[54] J. Wetmore, "Driving the dream. The history and motivations behind 60 years of automated highway systems in America," *Automotive History Review*, vol. 7, pp. 4–19, 2003.

[55] J. Schmidhuber, "Prof. Schmidhuber's highlights of robot car history." http://people.idsia.ch/˜juergen/robotcars.html, 2019.

[56] "Driverless Auto, Guided by Radio, Navigates Street," *The Washington Herald*, p. 5, Aug. 1921.

[57] "Driverless Radio Auto in Detroits Busiest Traffic," *Daily Ardmoreite*, p. 1, Aug. 1921.

[58] "Science: Radio Auto," *TIME*, Aug. 1925.

[59] C. Engelking, "The Driverless Car Era Began in 1925." http://blogs.discovermagazine.com/d-brief/2017/12/13/driverless-car-houdina-houdini/, Dec. 2017.

[60] M. Mann, "The Car That Drives Itself," *Popular Science*, vol. 172, pp. 75–79, May 1958.

[61] H. Moravec, "The stanford cart and the cmu rover," in *Autonomous Robot Vehicles* (I. J. Cox and G. T. Wilfong, eds.), pp. 407–41, Springer-Verlag, January 1990.

[62] H. P. Moravec, "Stanford Mech Eng Cart." http://archive.org/details/sailfilm_cart, 1966.

[63] H. P. Moravec, "Stanford Cart - Moravec 1979." https://www.youtube.com/watch?v=ypE64ZLwC5w, Oct. 1979.

[64] H. Zimmer, "PROMETHEUS - Ein europäisches Forschungsprogramm zur Gestaltung des künftigen Straßenverkehrs," Tech. Rep. 34/1, Forschungsgesellschaft für Straßen- und Verkehrswesen, 1990.

[65] M. Maurer and E. D. Dickmanns, “A system architecture for autonomous visual road vehicle guidance,” in *Proceedings of the 1997 IEEE Conference on Intelligent Transportation Systems*, (Boston, USA), pp. 578–583, IEEE, Nov. 1997.

[66] E. D. Dickmanns, “Forschungsbericht - Fakultät für Luft- und Raumfahrttechnik,” tech. rep., Universität der Bundeswehr München, Sept. 2002.

[67] E. D. Dickmanns, *Dynamic Vision for Perception and Control of Motion.* Springer, 2007.

[68] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, vol. 56. Springer, 2009.

[69] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, J. Dolan, D. Duggins, D. Ferguson, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. Howard, A. Kelly, D. Kohanbash, M. Likhachev, N. Miller, K. Peterson, R. Rajkumar, P. Rybski, B. Salesky, S. Scherer, Y. Woo-Seo, R. Simmons, S. Singh, J. Snider, A. Stentz, J. Ziglar, H. Bae, B. Litkouhi, J. Nickolaou, V. Sadekar, S. Zeng, J. Struble, and M. Taylor, “Tartan Racing: A Multi-Modal Approach to the DARPA Urban Challenge,” p. 25, May 2007.

[70] “DARPA Urban Challenge.” https://www.darpa.mil/about-us/timeline/darpa-urban-challenge.

[71] “TOYOTA Motor — 75 Years of TOYOTA — Technical Development — Electronics Parts.” http://www.toyota-global.com/company/history_of_toyota/75years/data/automotive_business/products_technology/technology_development/electronics_parts/index.html.

[72] “Stichworte der Zukunft: Integrierte Sicherheit.” https://media.daimler.com/marsMediaSite/de/instance/ko/Stichworte-der-Zukunft-Integrierte-Sicherheit.xhtml?oid=9272000, June 2009.

[73] “Nissan Demos New Lane Keeping Products.” https://web.archive.org/web/20050110073214/http://ivsource.net/archivep/2001/feb/010212_nissandemo.html, Feb. 2001.

[74] G. Nelson, “Tesla beams down ’autopilot’ mode to Model S.” https://www.autonews.com/article/20151014/OEM06/151019938/tesla-beams-down-autopilot-mode-to-model-s, Oct. 2015.

[75] B. Wasef, “2019 Audi A8 L with Level 3 autonomy driving review.” https://www.autoblog.com/2018/10/16/2019-audi-a8-l-review-first-drive, Oct. 2018.

[76] M. Porschenrieder, “Automatisiertes Fahren bei der BMW Group.,” tech. rep., BMW Group, May 2017.

[77] Mercedes-Benz, "The 2016 E-Class on the road to autonomous driving – Mercedes-Benz original." https://www.youtube.com/watch?time_continue=6&v=thG8mNuk1Sg, Feb. 2016.

[78] M. Aeberhard, S. Rauch, M. Bahram, G. Tanzmeister, J. Thomas, Y. Pilat, F. Homm, W. Huber, and N. Kaempchen, "Experience, results and lessons learned from automated driving on Germany's highways," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 42–57, 2015.

[79] G. Girardin and E. Mounier, "Sensors and Data Management for Autonomous Vehicles," tech. rep., Yole Dévelopement, 2015.

[80] "Providentia - Die intelligente Autobahn auf dem Digitalen Testfeld A9." http://testfeld-a9.de/.

[81] J. Thomas and R. Rojas, "Sensor-based road model estimation for autonomous driving," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, (Redondo Beach, USA), pp. 1764–1769, IEEE, June 2017.

[82] J. Ziegler, M. Werling, and J. Schroder, "Navigating car-like robots in unstructured environments using an obstacle sensitive cost function," in *Proceedings of the 2008 IEEE Intelligent Vehicles Symposium (IV)*, (Eindhoven, Netherlands), pp. 787–791, IEEE, June 2008.

[83] T. Rakowski, *Informationstheoretische Änderungserkennung von Hochgenauen Straßenmodellen Als Grundlage Für Automatisierte Fahrfunktionen.* Masters Thesis, Freie Universität Berlin, July 2013.

[84] K. Banerjee, D. Notz, J. Windelen, S. N. Gavarraju, and M. He, "Online Camera LiDAR Fusion and Object Detection on Hybrid Data for Autonomous Driving," in *Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV)*, (Changshu, China), IEEE, June 2018.

[85] M. Aeberhard and N. Kaempchen, "High-level sensor data fusion architecture for vehicle surround environment perception," in *Proceedings of the 8th International Workshop on Intelligent Transportation*, (Hamburg, Germany), Mar. 2011.

[86] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, "Vehicle trajectory prediction based on motion model and maneuver recognition," in *Proceedings of the 2013 IEEE/RSJ Intelligent Robots and Systems (IROS) Conference*, (Tokyo, Japan), pp. 4363–4369, IEEE, Nov. 2013.

[87] A. Eidehall and L. Petersson, "Statistical threat assessment for general road scenes using Monte Carlo sampling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 137–147, 2008.

[88] H. Winner, S. Hakuli, and G. Wolf, *Handbuch Fahrerassistenzsysteme: Grundlagen, Komponenten Und Systeme Für Aktive Sicherheit Und Komfort.* Springer, 2011.

[89] W. Höfs, M. Fukushima, P. Potters, A. Etemad, P. Mononen, J. Ference, and D. Allekotte, "Field Operational Tests - testing ITS applications in the real world," brochure, FOTNET, Sept. 2010.

[90] P. Mononen, "TeleFOT - Use and Impacts of Aftermarket & Nomadic Devices in Vehicles," *Presentation at VTT Technical Research Centre of Finland, Espoo, Finnland*, Apr. 2009.

[91] C. Kessler and A. Etemad, "European Large-Scale Field Operational Tests on In-Vehicle Systems - SP 6 D6.8 FOT Data," tech. rep., Ford Research & Advanced Engineering Europe, June 2012.

[92] M. Benmimoun, A. Pütz, A. Zlocki, and L. Eckstein, "Eurofot: Field operational test and impact assessment of advanced driver assistance systems: Final results," in *Proceedings of the FISITA 2012 World Automotive Congress*, pp. 537–547, Springer, 2013.

[93] H. Lietz, T. Petzoldt, M. Henning, J. Haupt, G. Wanielik, J. Krems, H. Mosebach, J. Schomerus, M. Baumann, and U. Noyer, "Methodische und technische Aspekte einer Naturalistic Driving Study," *FAT-Schriftenreihe*, no. 229, 2011.

[94] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Las Vegas, USA), pp. 3213–3223, June 2016.

[95] R. Eenink, Y. Barnard, M. Baumann, X. Augros, and F. Utesch, "UDRIVE: The European naturalistic driving study," in *Proceedings of 2014 Transport Research Arena*, (Paris, France), IFSTTAR, Apr. 2014.

[96] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[97] 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 *Proceedings of the 2018 IEEE International Conference on Intelligent Transportation Systems (ITSC)*, (Hawaii, USA), Nov. 2018.

[98] K. von Neumann-Cosel, M. Dupuis, and C. Weiss, "Virtual test drive-provision of a consistent tool-set for [d, h, s, v]-in-the-loop," in *Proceedings of the Driving Simulation Conference*, (Monaco), 2009.

[99] D. J. Murray-Smith, *Testing and Validation of Computer Simulation Models.* Springer, 2015.

[100] K. Borgeest, *Elektronik in Der Fahrzeugtechnik.* Springer, 2010.

[101] K. Cammaerts, P. Morse, and K. Kidera, "Leistungssteigerung durch Driver-in-the-Loop-Simulation," *ATZ - Automobiltechnische Zeitschrift*, vol. 121, pp. 52–57, Jan. 2019.

[102] A. J. Benson, "Motion Sickness," in *Medical Aspects of Harsh Environments*, vol. 2, pp. 1048–1083, Washington, USA: Borden Institute, 2002.

[103] C. Gold, D. Damböck, L. Lorenz, and K. Bengler, ""Take over!" How long does it take to get the driver back into the loop?," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 57, (Los Angeles, USA), pp. 1938–1942, SAGE Publications, 2013.

[104] M. Festner, A. Eicher, and D. Schramm, "Beeinflussung der Komfort-und Sicherheitswahrnehmung beim hochautomatisierten Fahren durch fahrfremde Tätigkeiten und Spurwechseldynamik," in *Uni-DAS eV Workshop Fahrerassistenz Und Automatisiertes Fahren*, (Walting Im Altmühltal, Germany), Mar. 2017.

[105] T. Bock, *Vehicle in the Loop: Test-Und Simulationsumgebung Für Fahrerassistenzsysteme.* Cuvillier Verlag, 2008.

[106] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer, "Defining and substantiating the terms scene, situation, and scenario for automated driving," in *Proceedings of the 2015 IEEE International Conference On Intelligent Transportation Systems (ITSC)*, (Las Palmas de Gran Canaria, Spain), pp. 982–988, IEEE, Sept. 2015.

[107] H. Elrofai, D. Worm, and O. O. den Camp, "Scenario identification for validation of automated driving functions," in *Advanced Microsystems for Automotive Applications 2016*, pp. 153–163, Springer, 2016.

[108] M. Dupuis, M. Strobl, and H. Grezlikowski, "OpenDRIVE 2010 and Beyond–Status and Future of the de facto Standard for the Description of Road Networks," in *Proceedings of the Driving Simulation Conference DSC Europe*, (Paris, France), pp. 231–242, Aug. 2010.

[109] P. Bender, J. Ziegler, and C. Stiller, "Lanelets: Efficient map representation for autonomous driving," in *Proceedings of the 2014 IEEE Intelligent Vehicles Symposium (IV)*, (Ypsilanti, USA), pp. 420–425, IEEE, June 2014.

[110] P. Hubertus, "The Benefits of a Common Map Data Standard for Autonomous Driving," *White Paper from Navigation Data Standard e.V.*, p. 11, 2019.

[111] P. Wang, M. Dupuis, U. Wössner, and A. F. Walser, "OpensScenario - Open File Format for the Cross-Functional Description of Dynamic Elements of a Virtual Driving Test," in *Proceedings of the Driving Simulation Conference DSC Europe*, (Tübingen, Germany), Sept. 2015.

[112] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," *Published at the 1st Conference on Robot Learning (CoRL)*, Nov. 2017.

[113] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO–simulation of urban mobility: An overview," in *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*, (Barcelona, Spain), ThinkMind, Oct. 2011.

[114] J. C. Hayward, "Near miss determination through use of a scale of danger," *Highway Research Record*, vol. 384, pp. 24–34, 1972.

[115] H. Winner, S. Geyer, and M. Sefati, "Maße für den Sicherheitsgewinn von Fahrerassistenzsystemen," *Maßstäbe des sicheren Fahrens*, vol. 6, 2013.

[116] W. Wachenfeld, P. Junietz, R. Wenzel, and H. Winner, "The worst-time-to-collision metric for situation identification," in *Proceedings of the 2016 IEEE Intelligent Vehicles Symposium (IV)*, (Gothenburg, Sweden), pp. 729–734, June 2016.

[117] K. Vogel, "A comparison of headway and time to collision as safety indicators," *Accident analysis & prevention*, vol. 35, no. 3, pp. 427–433, 2003.

[118] R. Schurig, *Straßenverkehrs-Ordnung StVO - mit Anlagen zur StVO (jeweils bei den §§ 40 bis 43) und Allgemeiner Verwaltungsvorschrift zur Straßenverkehrs-Ordnung (VwV-StVO) - Kommentar.* Reihe Verkehrsrecht, Bonn: Kirschbaum Verlag, 15 ed., 2015.

[119] J. Hillenbrand, A. M. Spieker, and K. Kroschel, "A multilevel collision mitigation approach—Its situation assessment, decision making, and performance tradeoffs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 528–540, 2006.

[120] J. Hillenbrand, K. Kroschel, and V. Schmid, "Situation assessment algorithm for a collision prevention assistant," in *Proceedings of the 2005 IEEE Intelligent Vehicles Symposium*, (Las Vegas, USA), pp. 459–465, IEEE, June 2005.

[121] A. Polychronopoulos, M. Tsogas, A. Amditis, U. Scheunert, L. Andreone, and F. Tango, "Dynamic situation and threat assessment for collision warning systems: The EUCLIDE approach," in *Proceedings of the 2004 IEEE Intelligent Vehicles Symposium*, (Parma, Italy), pp. 636–641, IEEE, June 2004.

[122] A. Rizaldi and M. Althoff, "Formalising Traffic Rules for Accountability of Autonomous Vehicles," in *Proceedings of the 2015 IEEE International Conference on Intelligent Transportation Systems (ITSC)*, (Las Palmas de Gran Canaria, Spain), Sept. 2015.

[123] K. Esterle, V. Aravantinos, and A. Knoll, "From Specifications to Behavior: Maneuver Verification in a Semantic State Space," in *Proceedings of the 2019 IEEE Intelligent Vehicle Symposiym*, (Paris, France), pp. 2140–2147, IEEE, June 2019.

[124] C. Diels and J. E. Bos, "Self-driving carsickness," *Applied ergonomics*, vol. 53, pp. 374–382, 2016.

[125] M. Festner, H. Baumann, and D. Schramm, "Der Einfluss fahrfremder Tätigkeiten und Manöverlängsdynamik auf die Komfort-und Sicherheitswahrnehmung beim hochautomatisierten Fahren: Ein Argument für die Adaptivität automatisierter Fahrfunktionen," *VDI/VW Gemeinschaftstagung Fahrerassistenz und Integrierte Sicherheit 2016*, 2016.

[126] T. Hierlmeier, *Anwendung von Scrum in der Konzeptentwicklung einer Evaluationsmetrik für die Absicherung des hochautomatisierten Fahrens aus Kundensicht.* Masters Thesis, Technische Universität München, Aug. 2017.

[127] International Electrotechnical Commission, "IEC 61508 - Functional safety of electrical/-electronic/programmable electronic safety-related systems," Apr. 2010.

[128] B. J. Czerny, J. D'Ambrosio, R. Debouk, and K. Stashko, "Functional Safety Draft International Standard for Road Vehicles: Background, Status, and Overview," *Presentation at the ISSC 2010, Minneapolis, USA*, 2010.

[129] M. Wood, Philipp Robbel, M. Maass, R. D. Tebbens, M. Meijs, M. Harb, J. Reach, K. Robinson, C. Knobel, D. Boymanns, M. Löhning, B. Dehlink, D. Kaule, R. Krüger, J. Frtunikj, F. Raisch, M. Gruber, J. Steck, J. Meija-Hernandez, D. Wittmann, T. Srivastava, M. E. Bouyouraa, S. Liu, Y. Wang, S. Syguda, P. Blüher, K. Klonecki, P. Schnarz, T. Wiltschko, S. Pukallus, K. Sedlaczek, N. Garbacik, D. Smerza, D. Li, A. Timmons, M. Bellotti, M. O'Brien, M. Schöllhorn, U. Dannebaum, J. Weast, A. Tatourian, B. Dornieden, P. Schnetter, P. Themann, T. Weidner, and P. Schlicht, *Safety First for Automated Driving (White Paper).* Aptiv Services US, LLC; AUDI AG; Bayrische Motoren Werke AG; Beijing Baidu Netcom Science Technology Co., Ltd; Continental Teves AG & Co oHG; Daimler AG; FCA US LLC; HERE Global B.V.; Infineon Technologies AG; Intel; Volkswagen AG, July 2019.

[130] International Organization for Standardization, "ISO/PAS 21448:2019 - Road vehicles - Safety of the intended functionality," Jan. 2019.

[131] W. Wendorff, "Quantitative SOTIF analysis for highly automated driving systems," *Presentation at Safetronic.2017, Stuttgart, Germany*, Nov. 2017.

[132] A. Weitzel, H. Winner, C. Peng, S. Geyer, L. Lotz, and M. Sefati, "Absicherungsstrategien für Fahrerassistenzsysteme mit Umfeldwahrnehmung," tech. rep., Bundesanstalt für Straßenwesen, Nov. 2014.

[133] J. Börcsök, "Funktionale Sicherheit," *Grundzüge sicherheitstechnischer Systeme*, vol. 2, 2006.

[134] K. Beck, *Test-Driven Development: By Example.* Addison-Wesley Professional, 2003.

[135] J. P. Kleijnen, "Verification and validation of simulation models," *European Journal of Operational Research*, vol. 82, no. 1, pp. 145–162, 1995.

[136] A. M. Law, W. D. Kelton, and W. D. Kelton, *Simulation Modeling and Analysis*, vol. 3. McGraw-Hill New York, 2000.

[137] U. Steininger, H.-P. Schöner, and M. Schiementz, "Validation of Assisted and Automated Driving Systems," *Presentation at crash.tech, Munich, Germany*, Apr. 2016.

[138] L. Eckstein and A. Zlocki, "Safety potential of ADAS–Combined methods for an effective evaluation," in *Proceedings of the 23rd International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, (Seoul, South Korea), May 2013.

[139] D. Asljung, J. Nilsson, and J. Fredriksson, "Using Extreme Value Theory for Vehicle Level Safety Validation and Implications for Autonomous Vehicles," *IEEE Transactions on Intelligent Vehicles*, no. 99, pp. 288–297, 2017.

[140] D. Anguelov, "MIT Self-Driving Cars." https://www.youtube.com/watch?v=Q0nGo2-y0xY, Feb. 2019.

[141] W. H. K. Wachenfeld, *How Stochastic Can Help to Introduce Automated Driving.* PhD thesis, Technische Universität Darmstadt, 2017.

[142] T. Victor, M. Rothoff, E. Coelingh, A. Ödblom, and K. Burgdorf, "When Autonomous Vehicles Are Introduced on a Larger Scale in the Road Transport System: The Drive Me Project," in *Automated Driving: Safer and More Efficient Future Driving* (D. Watzenig and M. Horn, eds.), pp. 541–546, Springer International Publishing, 2017.

[143] A. Sanghavi, "What is formal verification?," *EE Times-Asia*, p. 2, May 2010.

[144] 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.

[145] S. Mitsch, K. Ghorbal, and A. Platzer, “On provably safe obstacle avoidance for autonomous robotic ground vehicles,” in *Robotics: Science and Systems IX*, (Berlin, Germany), June 2013.

[146] M. Althoff, O. Stursberg, and M. Buss, “Online Verification of Cognitive Car Decisions,” in *Proceedings of the 2007 IEEE Intelligent Vehicles Symposium (IV)*, (Istanbul, Turkey), pp. 728–733, June 2007.

[147] E. Rocklage, H. Kraft, A. Karatas, and J. Seewig, “Automated scenario generation for regression testing of autonomous vehicles,” in *Proceedings of the 2017 International Conference on Intelligent Transportation Systems (ITSC)*, (Yokohama, Japan), pp. 476–483, Oct. 2017.

[148] C. Sippl, F. Bock, D. Wittmann, H. Altinger, and R. German, “From simulation data to test cases for fully automated driving and ADAS,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9976 LNCS, pp. 191–206, 2016.

[149] A. Andrews, M. Abdelgawad, and A. Gario, “Towards world model-based test generation in autonomous systems,” in *2015 3rd International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*, (Angers, France), pp. 1–12, Feb. 2015.

[150] R. Abdessalem, S. Nejati, L. Briand, and T. Stifter, “Testing vision-based control systems using learnable evolutionary algorithms,” in *Proceedings of the 2018 International Conference on Software Engineering*, vol. Part F137142, (Gothenburg, Sweden), pp. 1016–1026, May 2018.

[151] F. Schuldt, *Ein Beitrag für den methodischen Test von automatisierten Fahrfunktionen mit Hilfe von virtuellen Umgebungen.* Dissertation, TU Braunschweig, Apr. 2017.

[152] G. Bagschik, T. Menzel, and M. Maurer, “Ontology based scene creation for the development of automated vehicles,” in *Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV)*, (Changhsu, China), pp. 1813–1820, IEEE, June 2018.

[153] C. Amersbach and H. Winner, “Functional Decomposition: An Approach to Reduce the Approval Effort for Highly Automated Driving,” in *8. Tagung Fahrerassistenz*, (Munich, Germany), Nov. 2017.

[154] 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.

[155] F. Kruber, J. Wurst, and M. Botsch, "An Unsupervised Random Forest Clustering Technique for Automatic Traffic Scenario Categorization," in *Proceedings of the 2018 International Conference on Intelligent Transportation Systems (ITSC)*, (Hawaii, USA), pp. 2811–2818, Nov. 2018.

[156] 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 *Proceedings of the 2019 IEEE Intelligent Vehicle Symposiym*, (Paris, France), pp. 2463–2470, IEEE, June 2019.

[157] C. Amersbach and H. Winner, "Defining Required and Feasible Test Coverage for Scenario-Based Validation of Highly Automated Vehicles," in *Proceedings of the 2019 IEEE International Conference on Intelligent Transportation Systems*, (Auckland, NZ), pp. 425–430, IEEE, Nov. 2019.

[158] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH Journal*, vol. 1, p. 1, July 2014.

[159] A. Broadhurst, S. Baker, and T. Kanade, "Monte Carlo road safety reasoning," in *Proceedings of the 2005 IEEE Intelligent Vehicles Symposium (IV)*, (Las Vegas, USA), pp. 319–324, IEEE, June 2005.

[160] M. Althoff and A. Mergel, "Comparison of Markov Chain Abstraction and Monte Carlo Simulation for the Safety Assessment of Autonomous Cars," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1237–1247, 2011.

[161] M. Althoff, O. Stursberg, and M. Buss, "Erreichbarkeitsanalyse von Verkehrsteilnehmern zur Verbesserung von Fahrerassistenzsystemen," in *Proceedings of the 3. Tagung Aktive Sicherheit durch Fahrerassistenz*, (Garching, Germany), Apr. 2008.

[162] T. Kühbeck, *Pre-Crash Extraction of the Constellation of a Frontal Collision between Two Motor Vehicles.* PhD thesis, Staffordshire University, Sept. 2017.

[163] D. P. Kroese and R. Y. Rubinstein, "Monte Carlo methods," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 4, pp. 48–58, Jan. 2012.

[164] F. Pukelsheim, "The three sigma rule," *The American Statistician*, vol. 48, no. 2, pp. 88–91, 1994.

[165] E. Jones, T. Oliphant, and P. Peterson, "SciPy: Open source scientific tools for Python," 2001.

[166] M. Shimrat, "Algorithm 112: Position of point relative to polygon," *Communications of the ACM*, vol. 5, no. 8, p. 434, 1962.

[167] A. Lawitzky, D. Althoff, C. F. Passenberg, G. Tanzmeister, D. Wollherr, and M. Buss, “Interactive scene prediction for automotive applications,” in *Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV)*, (Gold Coast City, Australia), pp. 1028–1033, IEEE, June 2013.

[168] J. Hasch, “Driving towards 2020: Automotive radar technology trends,” in *Proceedings of the 2015 IEEE MTT-S International Conference On Microwaves for Intelligent Mobility (ICMIM)*, (Heidelberg, Germany), pp. 1–4, IEEE, apr 2015.

[169] P. Lindner and G. Wanielik, “3D LIDAR processing for vehicle safety and environment recognition,” in *Proceedings of the 2009 IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems*, (Nashville, USA), pp. 66–71, IEEE, Mar. 2009.

[170] E. Musk, “The future we’re building – and boring.” https://www.ted.com/talks/elon_musk_the_future_we_re_building_and_boring, Sept. 2018.

[171] S. Gibbs, “Hackers can trick self-driving cars into taking evasive action,” *The Guardian*, Sept. 2015.

[172] “OxTS RT3000 Datasheet,” tech. rep., Oxford Technical Solutions Ltd. (OxTS), June 2017.

[173] C. de Boor, *A Practical Guide to Splines*. Applied Mathematical Sciences, New York: Springer-Verlag, 1978.

[174] A. Wolberg, “Monotonic cubic spline interpolation,” in *Proceedings of the 1999 Computer Graphics International*, (Alberta, Canada), pp. 188–195, IEEE, June 1999.

[175] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, “Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 23, no. 4, pp. 550–560, 1997.

[176] M. H. Strobl, “SPIDER - The innovative software framework of the BMW driving simulation,” *VDI-Berichte*, no. 1745, 2003.

[177] T. Hanke, “Open Simulation Interface - Introduction and Overview,” *Presentation at Technical University of Munich*, Jan. 2017.

[178] D. Merkel, “Docker: Lightweight Linux Containers for Consistent Development and Deployment,” *Linux J.*, vol. 2014, Mar. 2014.

[179] M. Berk, *Safety Assessment of Environment Perception in Automated Driving Vehicles*. Dissertation, Technische Universität München, München, 2019.

[180] I. M. Sobol', "On the distribution of points in a cube and the approximate evaluation of integrals," *USSR Computational Mathematics and Mathematical Physics*, vol. 7, pp. 86–112, Jan. 1967.

[181] J. H. Halton, "Algorithm 247: Radical-inverse Quasi-random Point Sequence," *Communications of the ACM*, vol. 7, pp. 701–702, Dec. 1964.

[182] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 42, no. 1, pp. 55–61, 2000.

[183] Bundesministerium für Verkehr und digitale Infrastruktur, "Richtlinien fuer die Anlage von Autobanen (RAA)," 2008.

[184] Q. Wang, Y. Shen, and J. Q. Zhang, "A nonlinear correlation measure for multivariable data set," *Physica D: Nonlinear Phenomena*, vol. 200, pp. 287–295, Jan. 2005.

[185] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[186] L. Kocis and W. J. Whiten, "Computational investigations of low-discrepancy sequences," *ACM Transactions on Mathematical Software (TOMS)*, vol. 23, no. 2, pp. 266–294, 1997.

[187] J. Hiller, "Experiences and Takeaways of Ika's Handling and Processing of Field Data in Various Federal and European Projects for Automated Driving," *Presentation ASAM Ideation Workshop, Höhenkirchen, Germany*, Dec. 2019.

[188] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, vol. 3. Wiley New York, 1973.

[189] V. Estivill-Castro, "Why so many clustering algorithms: A position paper," *ACM SIGKDD explorations newsletter*, 2002.

[190] D. Müllner, "Modern hierarchical, agglomerative clustering algorithms," *arXiv preprint, arXiv:1109.2378*, Sept. 2011.

[191] R. Bellman, *Dynamic Programming.* Princeton University Press, 1957.

[192] L. C. Davis, "Effect of adaptive cruise control systems on traffic flow," *Physical Review E*, vol. 69, no. 6, 2004.

[193] M. Guériau, R. Billot, N.-E. El Faouzi, J. Monteil, F. Armetta, and S. Hassas, "How to assess the benefits of connected vehicles? A simulation framework for the design of cooperative traffic management strategies," *Transportation research part C: emerging technologies*, vol. 67, pp. 266–279, 2016.

[194] A. Talebpour and H. S. Mahmassani, "Influence of connected and autonomous vehicles on traffic flow stability and throughput," *Transportation Research Part C: Emerging Technologies*, vol. 71, pp. 143–163, 2016.

[195] R. E. Stern, S. Cui, M. L. Delle Monache, R. Bhadani, M. Bunting, M. Churchill, N. Hamilton, R. Haulcy, H. Pohlmann, F. Wu, B. Piccoli, B. Seibold, J. Sprinkle, and D. B. Work, "Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments," *Transportation Research Part C: Emerging Technologies*, vol. 89, pp. 205–221, 2018.

[196] K. Pearson, "On lines and planes of closest fit to systems of point in space," *Philosophical Magazine*, vol. 2, no. 11, pp. 559–572, 1901.

[197] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[198] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.

[199] J. Shlens, "A tutorial on principal component analysis," *arXiv preprint, arXiv:1404.1100*, 2014.

[200] J. G. van der Corput, "Verteilungsfunktionen," *Proceedings. Akadamie van Wetenschappen Amsterdam*, vol. 38, pp. 813–821, 1935.

[201] B. Vandewoestyne and R. Cools, "Good permutations for deterministic scrambled Halton sequences in terms of L2-discrepancy," *Journal of Computational and Applied Mathematics*, vol. 189, pp. 341–361, May 2006.

[202] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[203] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, (Oakland, USA), pp. 281–297, 1967.

[204] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[205] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD)*, vol. 96, no. 34, pp. 226–231, 1996.

[206] E. Elhamifar and R. Vidal, “Sparse subspace clustering: Algorithm, theory, and applications,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[207] T. Kohonen, “The self-organizing map,” *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[208] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, vol. 344. John Wiley & Sons, 2009.

Own Publications

This list contains publications the author of this work participated in. The entries are categorized by the type of publication and sorted chronologically. An asterisk (*) indicates equal contribution of the marked authors to the respective publication.

Conference Papers

[Paper1] **S. Wagner**, K. Groh, T. Kühbeck, M. Doerfel, and A. Knoll, "Using Time-To-React Based on Naturalistic Traffic Object Behavior for Scenario-Based Risk Assessment of Automated Driving," in *Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV)*, (Changhsu, China), IEEE, June 2018.

[Paper2] **S. Wagner***, K. Groh*, T. Kühbeck, and A. Knoll, "Towards Cross-Verification and Use of Simulation in the Assessment of Automated Driving," in *Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV)*, (Paris, France), IEEE, June 2019.

[Paper3] T. Salzmann, J. Thomas, T. Kühbeck, J.-c. Sung, **S. Wagner**, and A. Knoll, "Online Path Generation from Sensor Data for Highly Automated Driving Functions," in *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, (Auckland, NZ), IEEE, Oct. 2019.

[Paper4] 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 *Proceedings of the 2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, (Graz, Austria), IEEE, Nov. 2019.

[Paper5] J. Kerber*, **S. Wagner***, K. Groh, D. Notz, T. Kühbeck, D. Watzenig, and A. Knoll, "Clustering of the Scenario Space for the Assessment of Automated Driving," in *Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, Oct. 2020.

Journal Articles

[Journal1] K. Groh*, **S. Wagner***, T. Kühbeck, and A. Knoll, "Simulation and its Contribution to evaluate Highly Automated Driving Functions," *SAE International Journal of Advances and Current Practices in Mobility*, vol. 1, pp. 539–549, Apr. 2019.

[Journal2] J.-D. Korus, M. Bullinger, C. Schütz, P. Garcia Ramos, **S. Wagner**, and S. Müller, "A Method for Identifying Most Significant Vehicle Parameters for Controller Performance of Autonomous Driving Functions," *SAE International Journal of Advances and Current Practices in Mobility*, vol. 1, pp. 996–1005, Apr. 2019.

[Journal3] **S. Wagner***, K. Groh*, T. Kühbeck, D. Watzenig, L. Eckstein, and A. Knoll, "Virtual Assessment of Automated Driving – Methodology, Challenges and Lessons Learned," *SAE International Journal of Connected and Automated Vehicles*, vol. 2, pp. 263–277, Dec. 2019.

Scientific Posters

[Poster1] K. Groh*, **S. Wagner***, T. Kühbeck, and A. Knoll, "PEGASUS - Verification," PEGASUS Symposium 2019, May 2019.

Scientific Presentations

[Presentation1] M. Schiementz, **S. Wagner**, K. Groh, and T. Kühbeck, "PEGASUS - Test Case Variation and Execution." PEGASUS Symposium 2019, May 2019.

Supervised Student Theses

[Thesis1] P. Hagemann, *Befähigung einer Gaming Engine zur Darstellung prototypischer Anzeigen und Bedienfunktionen im Versuchsfahrzeug.* Bachelor Thesis, Technische Universität München, Feb. 2018.

[Thesis2] M. Sigl, *Development of a Method for Deriving a Vehicle Dynamic Model from Real World Experiments for Highly Automated Driving Simulation.* Masters Thesis, Technische Universität München, Apr. 2018.

[Thesis3] T. Salzmann, *Deriving a Neuronal Architecture for Scenario Based Multi-Sensor Input Intelligent Road Models for Automated Driving Functions.* Masters Thesis, Technische Universität München, Dec. 2018.

[Thesis4] A. Šaljić, *Systematic Variation of Driving Scenarios for the Assessment of Autonomous Driving.* Masters Thesis, Technische Universität München, Feb. 2019.

[Thesis5] F. Drost, *Online Modelling and Validation of Map Data for Autonomous Driving in Urban Scenarios.* Masters Thesis, Technische Universität München, Nov. 2019.

[Thesis6] J. Kerber, *Scenario Clustering to Leverage the Estimation of the Required Test Coverage for Automated Vehicles.* Masters Thesis, Technische Universität München, München, Mar. 2020.

Appendices

A. Testing Framework

This appendix contains supplementary material for Chapter 5.

A.1. The Representative Scenario in the JSCEN File Format

The following code shows the representative scenario from Section 5.2 extracted from GT data and converted into the JSCEN file format, as presented in Section 5.2.2.2. It is shortened at a few locations noted by `...` to decrease the space demand. Its purpose is to show the actual structure of the file format rather than the embedded variables.

```
{
    "map": "proving_ground.xodr",
    "description": "Representative cut-in and break scenario",
    "environment": {
        "fogDensity": "ExcellentVisibility",
        "precipitationDensity": "None",
        "dateTime": "2019-03-12T17:30:27.077079-07:00"
    },
    "objectPool": {
        "vehicles": [
            {
                "maneuver": {
                    "startPosition": {
                        "tOffset": -0.37117342098455985,
                        "referenceObj": {
                            "odrLane": -1,
                            "odrRoad": 8005
                        },
                        "sOffset": 2378.7244738907502
                    },
                    "velocity": [
                        {
                            "endTime": 0.8018612868509138,
                            "polynomParameter": [
                                -0.7943151516043908,
                                0.0,
                                1.685027119965802,
                                32.549377788836615
                            ],
                            "startTime": 0.004978895187377793
                        },
                        {
                            "endTime": 1.7336867786663803,
                            "polynomParameter": [
                                1.171807985028049,
                                -1.898927273235274,
                                0.1718054128749607,
                                33.49019298585831
                            ],
                            "startTime": 0.8018612868509138
                        },
                          ...
                    ],
                    "heading": [
                        {
                            "endTime": 2.2119508618748274,
                            "polynomParameter": [
                                -0.003544152010206214,
                                1.734723475976807e-18,
                                0.009364022675574405,
                                0.18418863294385404
                            ],
                            "startTime": 0.004978895187377 93
                        },
                        {
                            "endTime": 3.3589012772611615,
                            "polynomParameter": [
                                0.022716847306762 44,
                                -0.02346553239661224,
                                -0.042423749507145,
                                0.16675671467611974
                            ],
                            "startTime": 2.2119508618748274
                        },
                          ...
                    ]
                },
                "model": {
                    "color": "#000000",
                    "type": "Unknown"
                },
                "spawningTime": 0.004978895187377793,
                "description": "some vehicle description",
                "objectId": 1
            }
        ],
        "ego": {
            "maneuver": {
                "startPosition": {
                    "tOffset": -0.2974222258353594,
                    "referenceObj": {
                        "odrLane": -2,
                        "odrRoad": 8005
                    },
                    "sOffset": 2389.9265704435356
                },
                "velocity": [
                    ...
                ],
                "heading": [
                    ...
                ]
            }
        }
    },
    "duration": 9.199949741363525,
    "fileVersion": {
        "major": 0,
        "minor": 9
    }
}
```

A.2. Views of the Assessment Supervisor

This appendix presents selected views of the web-based *Assessment Supervisor* GUI as presented in Section 5.3.1. The selected views describe as follows:

- Figure A.1: Upon arrival, the user is presented with a dashboard containing all the relevant information on the system the framework is currently running on. This includes besides general information also the current workload of system's components.

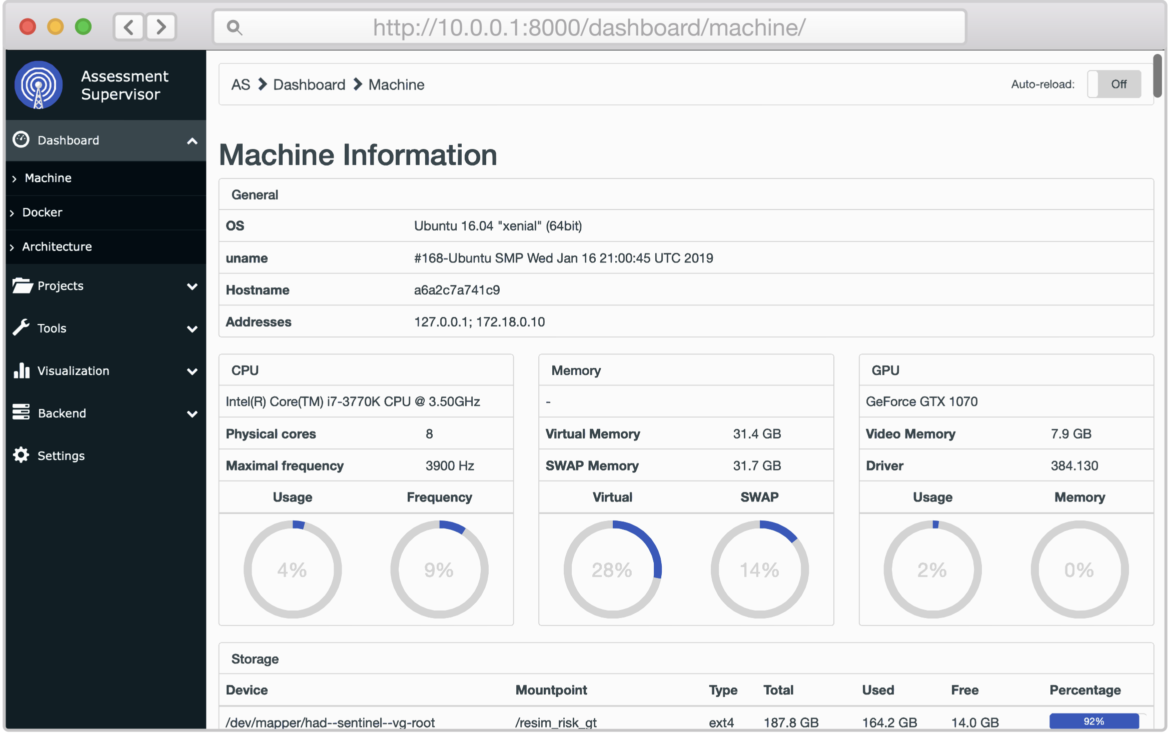
- Figure A.2: Each measurement defines an experiment in the *Assessment Supervisor*. All associated files that exist for an experiment are shown. Steps of the framework that succeeded display in green and failed in red. The reason that files and steps list in parallel in two rows is that the framework is configured to run twice for both GT and SD information.

- Figure A.3: A click on a file icon from the previous view leads to the respective visualization. In this example, the GT trace from a measurement is visualized on the road. The remainder of this page shows various physical quantities of this measurement over time.

- Figure A.4: The progress of the simulation module can be viewed live inside the browser in this view, accompanied by the queue of pending simulations. The simulation desktop itself is the graphical output of the simulation tool SPIDER [176]

- Figure A.5: The user's options to interact with the control and automation process within the framework is displayed in this view. On the top, the output of the control processes terminal is printed. Below are various buttons for starting, stopping, and pausing tasks, among others.

- Figure A.6: The remaining steps that need to be conducted for the experiments list in the task manager. Besides the information about the task itself, it also provides options to cancel a task or rearrange priorities.



Machine Information

General	
OS	Ubuntu 16.04 "xenial" (64bit)
uname	#168-Ubuntu SMP Wed Jan 16 21:00:45 UTC 2019
Hostname	a6a2c7a741c9
Addresses	127.0.0.1; 172.18.0.10

CPU	
Intel(R) Core(TM) i7-3770K CPU @ 3.50GHz	
Physical cores	8
Maximal frequency	3900 Hz
Usage	Frequency
4%	9%

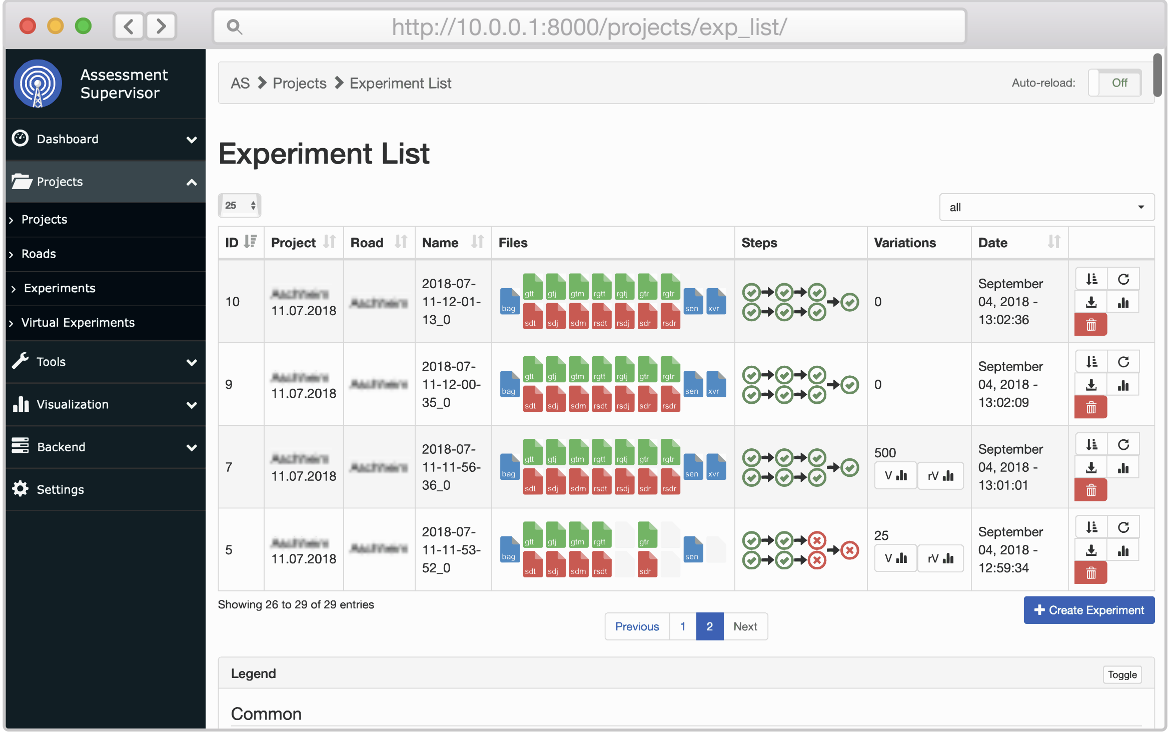
Memory	
-	
Virtual Memory	31.4 GB
SWAP Memory	31.7 GB
Virtual	SWAP
28%	14%

GPU	
GeForce GTX 1070	
Video Memory	7.9 GB
Driver	384.130
Usage	Memory
2%	0%

Storage

Device	Mountpoint	Type	Total	Used	Free	Percentage
/dev/mapper/had--sentinel--vg-root	/resim_risk_gt	ext4	187.8 GB	164.2 GB	14.0 GB	92%

Figure A.1.: Dashboard of the Assessment Supervisor.



Experiment List

ID	Project	Road	Name	Files	Steps	Variations	Date
10	[redacted] 11.07.2018	[redacted]	2018-07-11-12-01-13_0	bag gtt gtj gtm rgtt rgtj gtr rgtr sdt sdj sdm rsdt rsdj sdr rsdr sen xvr	✓→✓→✓ ✓→✓→✓ →✓	0	September 04, 2018 - 13:02:36
9	[redacted] 11.07.2018	[redacted]	2018-07-11-12-00-35_0	bag gtt gtj gtm rgtt rgtj gtr rgtr sdt sdj sdm rsdt rsdj sdr rsdr sen xvr	✓→✓→✓ ✓→✓→✓ →✓	0	September 04, 2018 - 13:02:09
7	[redacted] 11.07.2018	[redacted]	2018-07-11-11-56-36_0	bag gtt gtj gtm rgtt rgtj gtr rgtr sdt sdj sdm rsdt rsdj sdr rsdr sen xvr	✓→✓→✓ ✓→✓→✓ →✓	500 V rV	September 04, 2018 - 13:01:01
5	[redacted] 11.07.2018	[redacted]	2018-07-11-11-53-52_0	bag gtt gtj gtm rgtt gtr sdt sdj sdm rsdt sdr sen	✓→✓→✗ ✓→✓→✗ →✗	25 V rV	September 04, 2018 - 12:59:34

Showing 26 to 29 of 29 entries

Figure A.2.: Experiment list of the Assessment Supervisor.

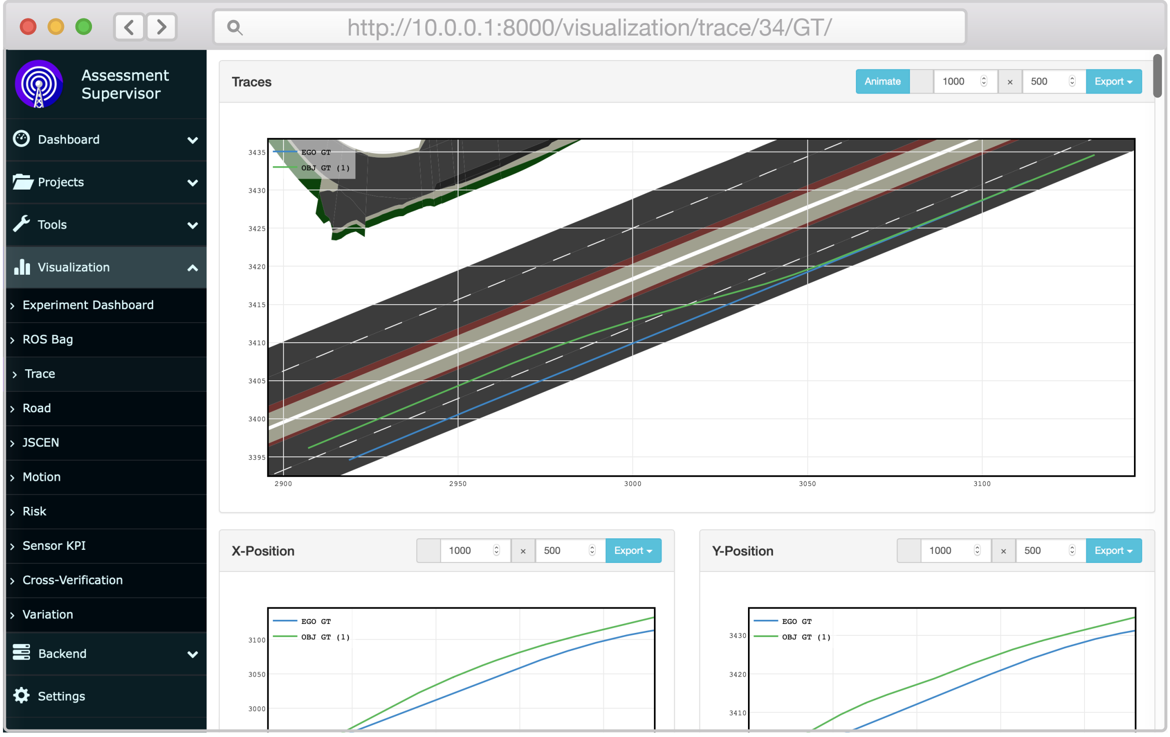
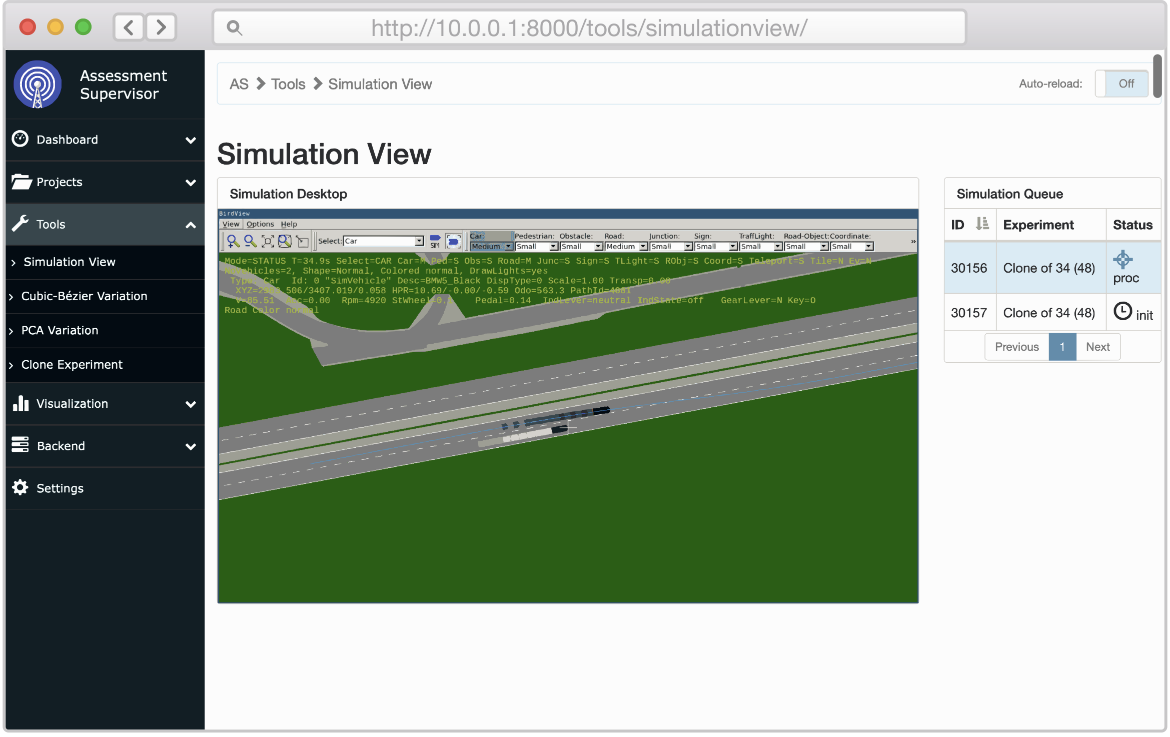


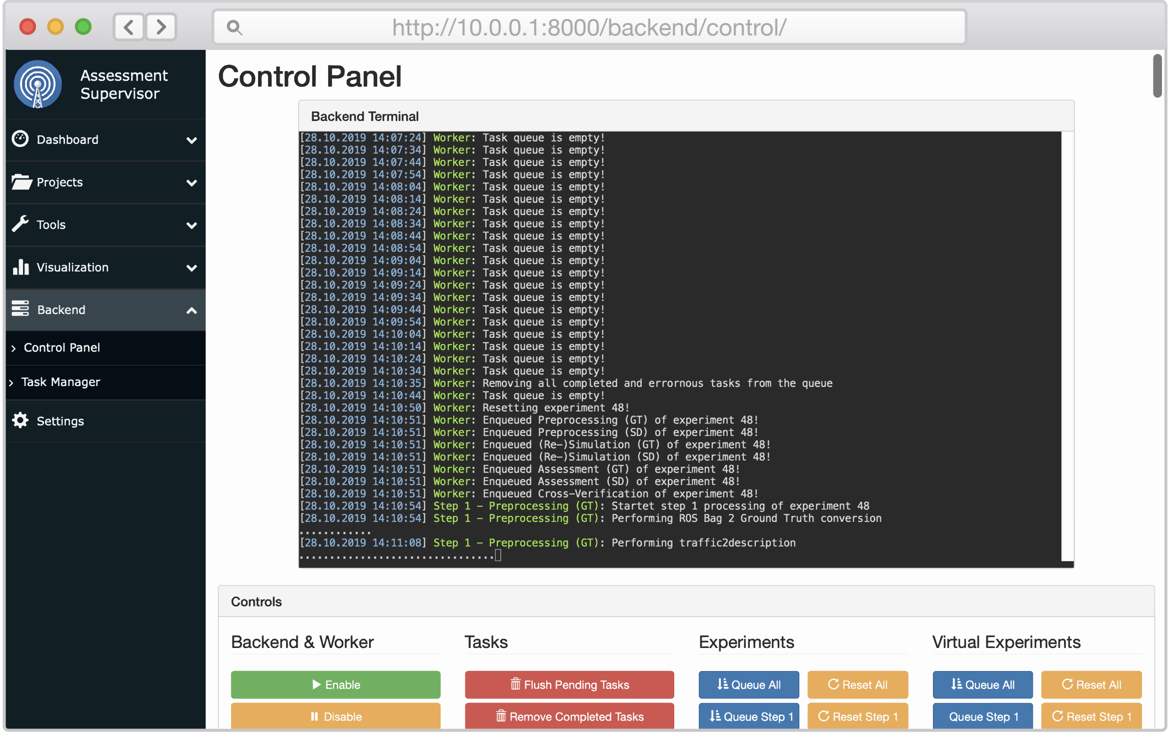
Figure A.3.: Trace visualization of the Assessment Supervisor.



```
Mode=STATUS T=34.9s Select=CAR Car=M Ped=S Obs=S Road=M Junc=S Sign=S TLight=S RObj=S Coord=S Teleport=S Tile=N Ev=N
NoVehicles=2, Shape=Normal, Colored normal, DrawLights=yes
 Type: Car  Id: 0 "SimVehicle" Desc=BMW5_Black DispType=0 Scale=1.00 Transp=0.00
  XYZ=2983.506/3407.019/0.058 HPR=10.69/-0.00/-0.59 Odo=563.3 PathId=4001
  V=85.51  Acc=0.00  Rpm=4920 StWheel=0.1    Pedal=0.14  IndLever=neutral IndState=off   GearLever=N Key=0
Road Color normal
```

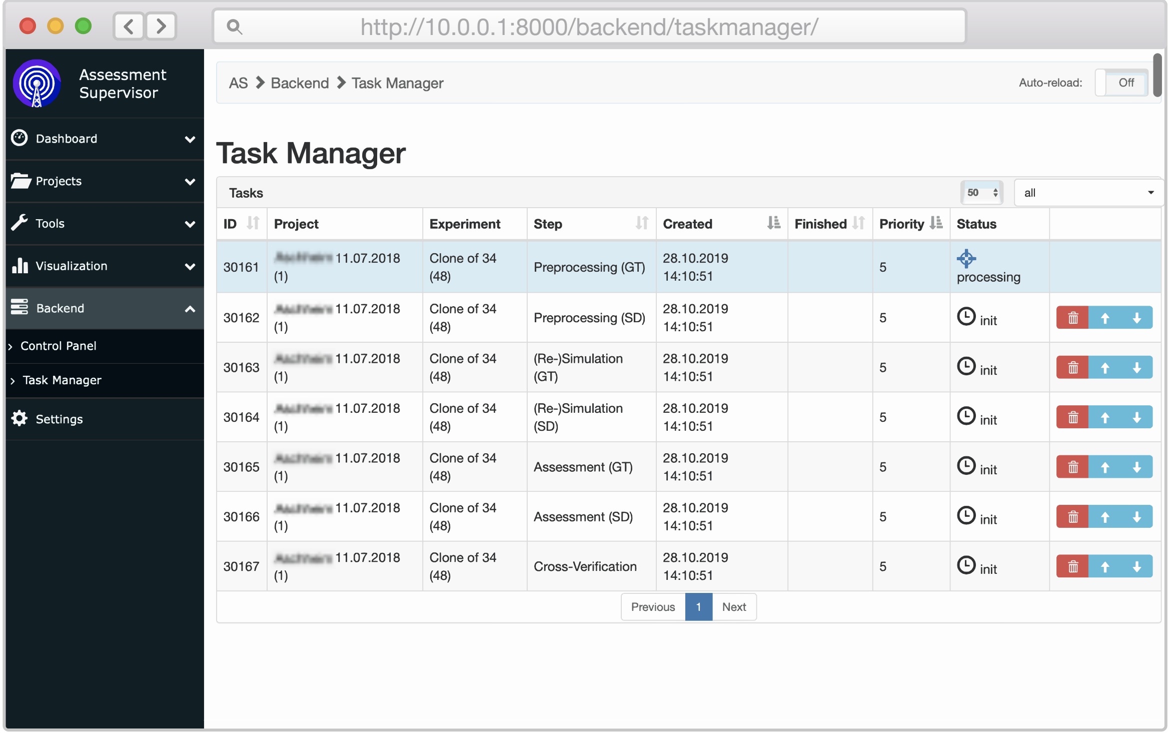
ID	Experiment	Status
30156	Clone of 34 (48)	proc
30157	Clone of 34 (48)	init

Figure A.4.: Simulation within the Assessment Supervisor using SPIDER [176].



```
[28.10.2019 14:07:24] Worker: Task queue is empty!
[28.10.2019 14:07:34] Worker: Task queue is empty!
[28.10.2019 14:07:44] Worker: Task queue is empty!
[28.10.2019 14:07:54] Worker: Task queue is empty!
[28.10.2019 14:08:04] Worker: Task queue is empty!
[28.10.2019 14:08:14] Worker: Task queue is empty!
[28.10.2019 14:08:24] Worker: Task queue is empty!
[28.10.2019 14:08:34] Worker: Task queue is empty!
[28.10.2019 14:08:44] Worker: Task queue is empty!
[28.10.2019 14:08:54] Worker: Task queue is empty!
[28.10.2019 14:09:04] Worker: Task queue is empty!
[28.10.2019 14:09:14] Worker: Task queue is empty!
[28.10.2019 14:09:24] Worker: Task queue is empty!
[28.10.2019 14:09:34] Worker: Task queue is empty!
[28.10.2019 14:09:44] Worker: Task queue is empty!
[28.10.2019 14:09:54] Worker: Task queue is empty!
[28.10.2019 14:10:04] Worker: Task queue is empty!
[28.10.2019 14:10:14] Worker: Task queue is empty!
[28.10.2019 14:10:24] Worker: Task queue is empty!
[28.10.2019 14:10:34] Worker: Task queue is empty!
[28.10.2019 14:10:35] Worker: Removing all completed and errornous tasks from the queue
[28.10.2019 14:10:44] Worker: Task queue is empty!
[28.10.2019 14:10:50] Worker: Resetting experiment 48!
[28.10.2019 14:10:51] Worker: Enqueued Preprocessing (GT) of experiment 48!
[28.10.2019 14:10:51] Worker: Enqueued Preprocessing (SD) of experiment 48!
[28.10.2019 14:10:51] Worker: Enqueued (Re-)Simulation (GT) of experiment 48!
[28.10.2019 14:10:51] Worker: Enqueued (Re-)Simulation (SD) of experiment 48!
[28.10.2019 14:10:51] Worker: Enqueued Assessment (GT) of experiment 48!
[28.10.2019 14:10:51] Worker: Enqueued Assessment (SD) of experiment 48!
[28.10.2019 14:10:51] Worker: Enqueued Cross-Verification of experiment 48!
[28.10.2019 14:10:54] Step 1 - Preprocessing (GT): Startet step 1 processing of experiment 48
[28.10.2019 14:10:54] Step 1 - Preprocessing (GT): Performing ROS Bag 2 Ground Truth conversion
............
[28.10.2019 14:11:08] Step 1 - Preprocessing (GT): Performing traffic2description
................................
```

Figure A.5.: Control panel of the Assessment Supervisor.



ID	Project	Experiment	Step	Created	Finished	Priority	Status	
30161	[redacted] 11.07.2018 (1)	Clone of 34 (48)	Preprocessing (GT)	28.10.2019 14:10:51		5	processing	
30162	[redacted] 11.07.2018 (1)	Clone of 34 (48)	Preprocessing (SD)	28.10.2019 14:10:51		5	init	
30163	[redacted] 11.07.2018 (1)	Clone of 34 (48)	(Re-)Simulation (GT)	28.10.2019 14:10:51		5	init	
30164	[redacted] 11.07.2018 (1)	Clone of 34 (48)	(Re-)Simulation (SD)	28.10.2019 14:10:51		5	init	
30165	[redacted] 11.07.2018 (1)	Clone of 34 (48)	Assessment (GT)	28.10.2019 14:10:51		5	init	
30166	[redacted] 11.07.2018 (1)	Clone of 34 (48)	Assessment (SD)	28.10.2019 14:10:51		5	init	
30167	[redacted] 11.07.2018 (1)	Clone of 34 (48)	Cross-Verification	28.10.2019 14:10:51		5	init	

Figure A.6.: Task manager of the Assessment Supervisor.

B. Scenario Variation and the Scenario Space

This appendix contains supplementary material for Chapter 6.

B.1. The Principal Component Analysis

The PCA is a method for the orthogonalization and dimensionality reduction of data that was introduced by Pearson in 1901 [196]. Despite it being over 100 years old, it is still a popular technique for this matter [197]. The name PCA was assigned later in history [198]. Its modern implementation through Singular Value Decomposition (SVD) is explained in the following [199]. Let $\mathbf{X} \in \mathbb{R}^{N_{smp} \times N_F}$ be a data set with N_F features and N_{smp} samples and correlations between the features. First, the data has to be normalized to be zero mean through the transformation

$$\tilde{\mathbf{X}}_i = \mathbf{X}_i - \text{mean}(\mathbf{X}_i) \cdot \mathbb{1}_{N_{smp} \times 1} \qquad \forall i \in [1, N_F], \tag{B.1}$$

where i denotes the ith column or feature of $\mathbf{X}$ and $\mathbb{1}_{N_{smp} \times 1}$ is a column vector of ones. With the normalization

$$\hat{\mathbf{X}} = \frac{1}{\sqrt{N}} \tilde{\mathbf{X}}^T, \tag{B.2}$$

the computation of the covariance matrix

$$\mathbf{C}_{\tilde{\mathbf{X}}} = \hat{\mathbf{X}}^T \hat{\mathbf{X}} \left(= \frac{1}{N} \tilde{\mathbf{X}} \tilde{\mathbf{X}}^T \right). \tag{B.3}$$

is possible. This matrix contains the variances of the features on the diagonal elements. The covariances between the features locate on the off-diagonal elements. Large covariance values indicate strong correlations between the features, whose removal is the target of the PCA. For this matter, the properties of the SVD can be exploited. Let the decomposition of the normalized feature data set $\hat{\mathbf{X}}$ be

$$\hat{\mathbf{X}} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T. \tag{B.4}$$

Thereby much new information generates. First, the columns of the $\mathbf{V}^{N_F \times N_F}$ are the principal directions or the axes of the new orthogonal basis of the data. Second, the elements of the diagonal matrix $\mathbf{\Sigma}^{N_{smp} \times N_F}$ contains the Eigenvalues of the data ordered by their magnitude. Hence, together with the unitary matrix $\mathbf{U}^{N_{smp} \times N_{smp}}$, the so-called score matrix $\mathbf{U\Sigma}$ can be calculated the third piece of information. It is a measure of the importance of each component to the representation of the original data. The less important a component is, the less information about the original data is represented through it. At this point, the dimensionality reduction of the PCA can follow. Deleting less important components through dropping respective rows or columns in the N_F direction of the matrices, the dimensionality can be reduced without the loss of too much information. In general, for a PCA, the inequality $N_C \leq N_F$ holds.

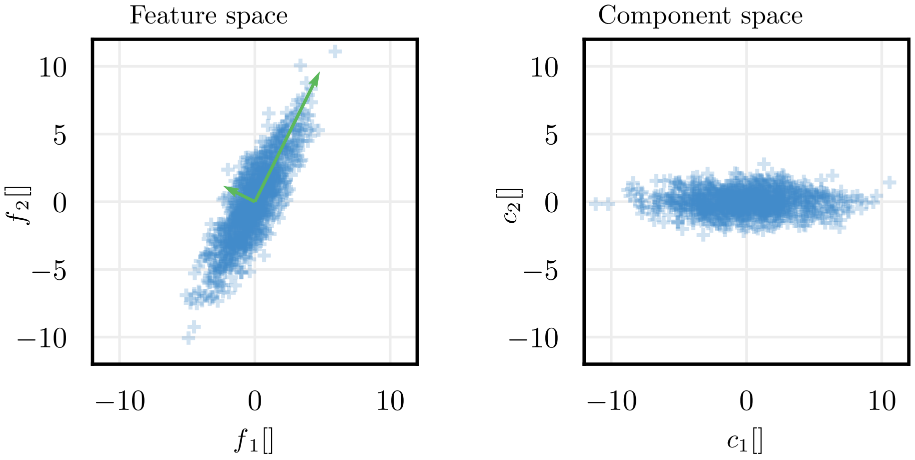


Figure B.1.: Exemplary PCA on two-dimensional data. The left side shows the original data with the principal directions, and the right contains the component space representation of the data.

Hence, the PCA is interpreted as a multidimensional rotation of the normalized feature data into a component coordinate frame, reducing correlations and covariances in $\mathbf{C}_{\tilde{\mathbf{X}}}$ the most. Certainly, this is only possible for linear correlations with the presented method. Nonlinear extensions exist but are not necessary for the application in this work. This statement can be confirmed by proving the absence of nonlinear correlations in the component data by the NCC from Appendix B.2, for example. Also, strongly correlated feature data requires fewer components for its representation. Even further, using too many components results in remaining correlations in the component space. Therefore, choosing the right number of components is a tradeoff between avoiding loss of information and correlations in-between the components.

Two examples follow to state the cases. In both cases, the implementation of the *PCA()* class from *scikit-learn* [185] is used to obtain the presented results. First, linearly correlated data shows on the left side of Figure B.1. It seems evident that this data can be shown in an uncorrelated fashion by merely rotating the coordinate frame. The principal directions found through the PCA indicate with green arrows whose lengths resemble the importance. On the right of this figure, the data is presented in its component space. The more important axis orientates in $c_1[]$ direction and the less important along $c_2[]$. The data loss caused by the PCA can be measured through the mean square error between the feature data and its forth and back transformation through the PCA. For this example, a value of $1.4 \cdot 10^{-31}$ shows that almost no information is lost, most probably only through numerical operations. This is reasonable as no dimensionality reduction is performed in this example.

The second example on three-dimensional data shows in Figure B.2 on the left side. This data is linearly correlated between the f_1 and f_2 axes and skewed with little added noise in the f_3 direction. Hence, two components are chosen this time, and two principal directions are shown as

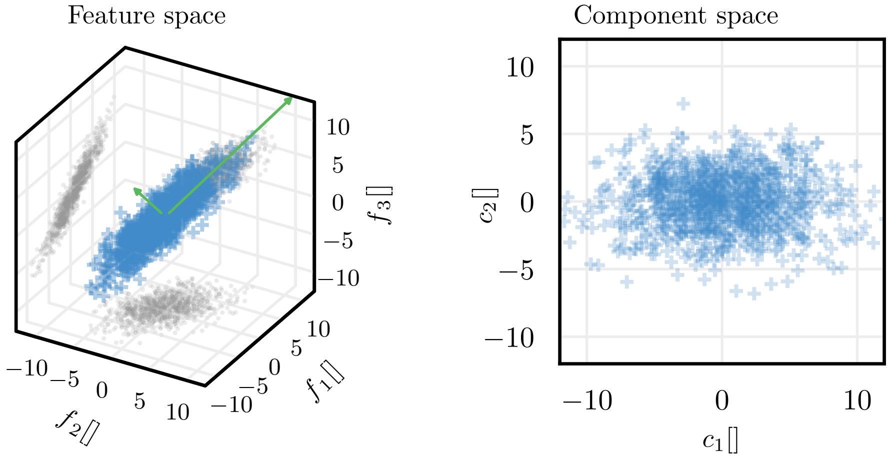


Figure B.2.: Exemplary PCA on three-dimensional data with dimensionality reduction. The left side shows the original data with the principal directions and 2D projections of the data in gray, and the right contains the component space representation of the data.

green arrows in the figure. The component representation of the data again shows on the right. As its dimensionality is one lower than the data, a higher error of 0.009 is present. However, it is still low and traceable to the little added noise in f_3 direction that is lost in the transformation. Other than that, the data is planar and can be well explained in two dimensions.

B.2. The Nonlinear Correlation Coefficient

The widely used PCC is able to find linear correlations between two samples of two variables by design. However, in some cases, information about nonlinear correlations are necessary. In this work, it suffices to show that standard PCA is sufficient to orthogonalize maneuver features in Section 6.1.2 by showing that no nonlinear correlations are left in the components space. Therefore, the NCC, as defined by [184], is presented in the following.

If x_1 and x_2 are two discrete signals with N_{smp} samples each and N_1 and N_2 possible different values respectively, the contained information in the signals can be measured through their entropy $H(\,\cdot\,)$

$$H(x_i) = -\sum_{n=1}^{N_i} p_n \ln(p_n) \qquad\qquad i = 1, 2, \tag{B.5}$$

with p_n being the probability of the specific value in a signal. The joint entropy of both signals is defined with the respective joint probability $p_{n,m}$ as

$$H(x_1, x_2) = -\sum_{n=1}^{N_1}\sum_{m=1}^{N_2} p_{n,m} \ln(p_{n,m})\,. \tag{B.6}$$

Accordingly, the mutual information $I(x_1, x_2)$ both signals share defines as

$$I(x_1, x_2) = H(x_1) + H(x_2) - H(x_1, x_2) \tag{B.7}$$

Hence, it defines as the exclusively shared information, more precisely the joint information minus the individual information of the signals, between both signals indicating correlations. The signs in this formula are inverted in contrast to this literal interpretation, as the entropies from Equation (B.5) and Equation (B.6) are defined negatively. Finally, the normalization

$$\frac{I(x_1, x_2)}{\sqrt{H(x_1)H(x_2)}}. \tag{B.8}$$

reveals a correlation that can be compared to the PCC but in a nonlinear fashion.
However, especially for continuous variables, the numbers of individual values N_1 and N_2 are not determinable. Hence a binned alternative suitable for sampled signals follows. First, all values in the signals x_1 and x_2 are ordered ascending and named x_{s1} and x_{s2}. Then, each is divided into b intervals called ranks, each containing an equal number $\frac{N_{smp}}{b}$ of samples. The value ranges of each rank defines a $b \times b$ grid of two-dimensional intervals. Next, sample pairs $\{x_1(n),x_2(n)\}_{1 \leq n \leq N_{smp}}$ of the original signals x_1 are distributed into the bins fitting the intervals, and the number of members $n_{i,j}$ of each bin is counted. The altered joint entropy now defines as

$$H^b(x_1, x_2) = -\sum_{i=1}^{b}\sum_{j=1}^{b} \frac{n_{i,j}}{N} \log_b \frac{n_{i,j}}{N}. \tag{B.9}$$

Taking Equation (B.7) and $H^b(x_i) = 1 \quad \forall i$ for the altered entropy [184], the NCC finally computes as

$$\text{NCC}(x_1, x_2) = 2 + \sum_{i=1}^{b}\sum_{j=1}^{b} \frac{n_{i,j}}{N} \log_b \frac{n_{i,j}}{N}. \tag{B.10}$$

To show its performance compared to the commonly known and used PCC, the correlation of the following five test signals is examined:

$$\begin{aligned} x_1(t) &= 0.75 \cdot (\sin(t) + 1) \\ x_2(t) &= \text{awgn}(x_1(t), 30) \\ x_3(t) &= 0.75 \cdot (\cos(t) + 1) \qquad\qquad t \in [0; 100]. \\ x_4(t) &= \text{awgn}(x_3(t), 30) \\ x_5(t) &= e^{\left(\frac{t}{100}\right)} - 1 \end{aligned} \tag{B.11}$$

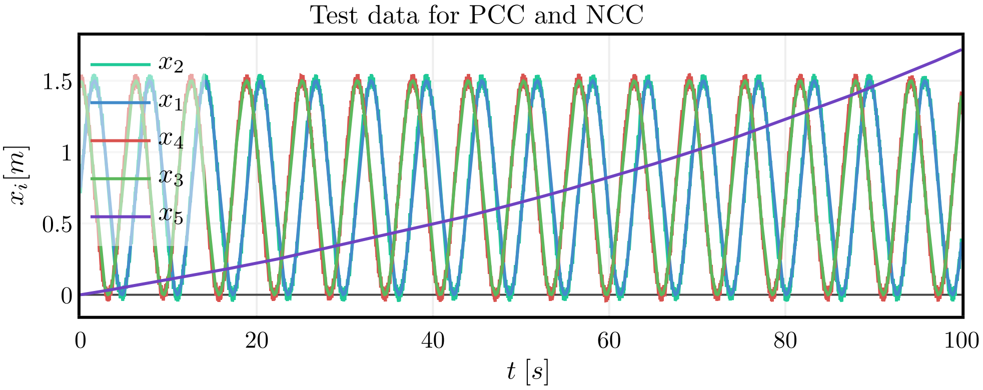


Figure B.3.: Example signals from Equation (B.11) to test the NCC. The order of the signals is changed for visibility of the non-noisy signals.

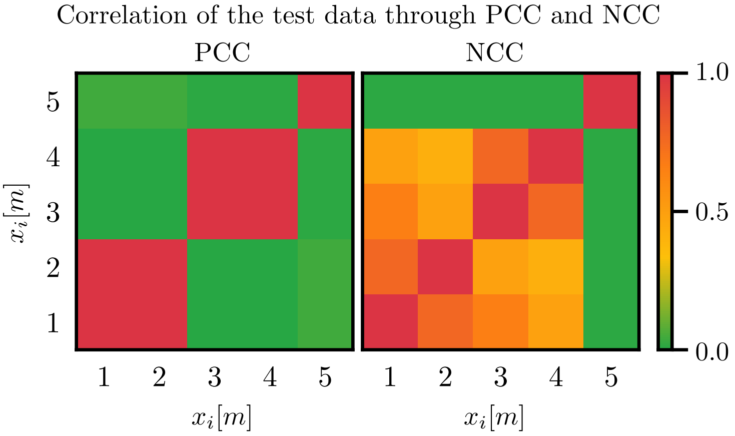


Figure B.4.: Correlation through PCC and NCC on the example data.

These are plotted for $t \in [0s, 100s]$ in Figure B.3. As the sine and cosine in x_1 and x_2 are timely shifted periodic signals, they share a nonlinear correlation. In contrast, x_2 and x_4 are x_1 and x_2 with Additive White Gaussian Noise (AWGN) with $30dB$ Signal to Noise Ratio (SNR), respectively, and therefore linearly correlated. For completeness, the exponential behavior of x_5 does not correlate with any other signal.

The results of both the PCC and NCC display in Figure B.4. These are achieved through a custom implementation of the NCC in *Python*. First, it is observable that the PCC reveals a strong correlation between a sinusoid and its noisy counterpart, but almost none in-between the sinusoidal signals. The NCC improves these insights in two ways. First, the noise is recognized, and the relevant signals are less correlated. Second, the nonlinear correlation between the time-shifted sine and cosine is recognized as mediocre correlation. Both measures correctly show x_5 as uncorrelated to any other.

The presented measure for nonlinear correlation has shown to be able to extend the information

```
Input: index $n$
Input: base $b$
Output: result $r$
Function halton(index $n$, base $b$) : result $r$ is
    $r \leftarrow 0$;
    $c \leftarrow 0$;
    while $n > 0$ do
        $c \leftarrow c + 1$;
        $f \leftarrow \frac{1}{b^c}$;
        $r \leftarrow r + f \cdot (n \bmod b)$;
        $n \leftarrow \lfloor \frac{n}{b} \rfloor$;
return result;
```

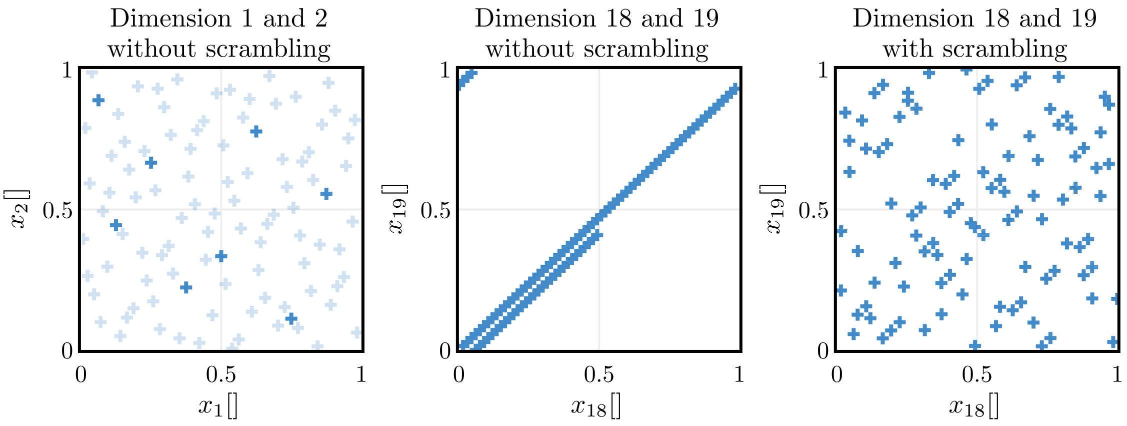
Algorithm B.1: Algorithm for the generation of the nth Halton Sequence sampling point r in dimension with base b, according to [181].

gained through the commonly used PCC. Hence, it is suitable to demonstrate the absence of nonlinear correlations in a data set.

B.3. The Halton Sequence with Reverse Radix Scrambling

Many Monte-Carlo based methods require uniform sampling methods in multiple dimensions. Also, for the local scenario variation introduced in Section 6.1, such a technique is required. The most common and also uniform generation of sample points in the full fact set. Hereby, a discretization for every dimension is chosen individually, and permutations of the dimensions fill the space. However, this method does not allow a free choice of the number of samples and becomes infeasible for higher dimensions as in the variation task of this work. Additionally, it is not straight forward to add additional points after a completed sampling. In contrast, the Halton Sequence [181] is a technique that overcomes these limitations. Reverse Radix Scrambling [186] suffices to improve the uniformity of the data further. Its implementation shall be explained briefly in the following.

The Halton Sequence founds on the Van der Corput Sequence [200], meaning it bases on dividing the interval $[0, 1]$ repeatedly for every dimension individually. First, for every dimension, a prime number is defined as its basis b. For example, the first three dimensions have the bases 2, 3, and 5. Now the nth coordinate in a dimension is calculated by converting the decimal number of n into the representation of the base, inverting the digits, pushing them beyond the decimal point, and converting the number back. This process calculates for the first eight points in the


Dimension 1 and 2 without scrambling

Dimension 18 and 19 without scrambling

Dimension 18 and 19 with scrambling

Figure B.5.: Exemplary Halton Sequences. The left shows the eight samples for the first two dimensions from Equation (B.14) extend to 100 by faded out scatters. The middle exhibits the correlation problem in-between higher dimensions that is solved on in the right plot through scrambling.

first two dimensions as:

$$b = 2 \begin{cases} n=1 & =1_2 & \Rightarrow & 0.1_2 & =1\cdot 2^{-1} & =\frac{1}{2} \\ n=2 & =10_2 & \Rightarrow & 0.01_2 & =1\cdot 2^{-2} & =\frac{1}{4} \\ n=3 & =11_2 & \Rightarrow & 0.11_2 & =1\cdot 2^{-1}+1\cdot 2^{-2} & =\frac{3}{4} \\ n=4 & =100_2 & \Rightarrow & 0.001_2 & =1\cdot 2^{-3} & =\frac{1}{8} \\ n=5 & =101_2 & \Rightarrow & 0.101_2 & =1\cdot 2^{-1}+1\cdot 2^{-3} & =\frac{5}{8} \\ n=6 & =110_2 & \Rightarrow & 0.011_2 & =1\cdot 2^{-2}+1\cdot 2^{-3} & =\frac{3}{8} \\ n=7 & =111_2 & \Rightarrow & 0.111_2 & =1\cdot 2^{-1}+1\cdot 2^{-2}+1\cdot 2^{-3} & =\frac{7}{8} \\ n=8 & =1000_2 & \Rightarrow & 0.0001_2 & =1\cdot 2^{-4} & =\frac{1}{16} \end{cases} \tag{B.12}$$

$$b = 3 \begin{cases} n=1 & =1_3 & \Rightarrow & 0.1_3 & =1\cdot 3^{-1} & =\frac{1}{3} \\ n=2 & =2_3 & \Rightarrow & 0.2_3 & =2\cdot 3^{-1} & =\frac{2}{3} \\ n=3 & =10_3 & \Rightarrow & 0.01_3 & =1\cdot 3^{-2} & =\frac{1}{9} \\ n=4 & =11_3 & \Rightarrow & 0.11_3 & =1\cdot 3^{-1}+1\cdot 3^{-2} & =\frac{4}{9} \\ n=5 & =12_3 & \Rightarrow & 0.21_3 & =2\cdot 3^{-1}+1\cdot 3^{-2} & =\frac{7}{9} \\ n=6 & =20_3 & \Rightarrow & 0.02_3 & =2\cdot 3^{-3} & =\frac{2}{9} \\ n=7 & =21_3 & \Rightarrow & 0.12_3 & =1\cdot 3^{-1}+2\cdot 3^{-2} & =\frac{5}{9} \\ n=8 & =22_3 & \Rightarrow & 0.21_3 & =2\cdot 3^{-1}+2\cdot 3^{-2} & =\frac{8}{9} \end{cases} \tag{B.13}$$

$$(x_1(n), x_2(n)) = \left\{ \left(\frac{1}{2}, \frac{1}{3}\right), \left(\frac{1}{4}, \frac{2}{3}\right), \left(\frac{3}{4}, \frac{1}{9}\right), \left(\frac{1}{8}, \frac{4}{9}\right), \left(\frac{5}{8}, \frac{7}{9}\right), \ldots \right\}. \tag{B.14}$$

A computer-executable description of the generation is present in Algorithm B.1 in pseudo-code. It basically works by checking which share of the different powers of the basis b needs to be added according to the representation of the base beyond the decimal point. The first eight points of the sequence from Equation (B.14) are illustrated in Figure B.5 in the far left plot. Uniformity of the samples is better observable the more samples are plotted. Therefore, this number extends to 100 with faded out points. While uniformity is excellent along lower dimensions, this algorithm exposes highly linear correlations between higher dimensions disturbing the uniformity [201]. As an example, the middle plot of Figure B.5 demonstrates this effect between the 18th and 19th dimension.

A common method to overcome this issue is scrambling the bases. Thereby, line 7 of Algorithm B.1 replaces with $r \leftarrow r + f \cdot \text{scramble}(n \bmod b, b)$. Basically, the share that assigns to a specific power of the basis is shuffled. For example, if a 2 is present at the second digit beyond the decimal point for $b = 3$, the original algorithm would add $\frac{2}{9}$, but the scramble maybe $\frac{5}{9}$ depending on the used scrambling. Hence, the 2 is always exchanged with a 5 for $b = 3$. As the scrambling differs depending on the basis b and therefore the dimension (the second argument to the function), this method breaks the linear correlation. A popular scrambling method is Reverse Radix Scrambling [186]. Herein, the number provided by the argument $n \bmod b$ is converted to a binary number with enough digits to store integers up to b. Then, the digit order is inverted and converted back to decimal to reveal the new share. A more detailed explanation is intentionally left out at this point and can be received from [186]. The most right plot of Figure B.5 shows the dimensions 18 and 19 again but with the scrambling method. It can be seen that the correlation brakes and the uniformity improves.

The presented Halton Sampling Sequence, including the Reverse Radix scrambling, implements in a custom *Python* class to provide the presented results. It can provide an arbitrary number of uniformly distributed samples across any dimensionality. Additionally, it is deterministic by design, and additional points can be added at any time. Hence, it is suitable for the application in the scenario variation from Section 6.1.

B.4. A brief Introduction into Hierarchical Clustering

This appendix aims at giving a brief introduction into connectivity-based or hierarchical clustering [202]. There exist many other approaches to the clustering problem, and the number is steadily growing [189]. Among these are centroid-based [203], distribution-based [204], density-based [205], subspace [206] and neural [207] approaches. However, the goal of this work is not to find the best fitting and most advanced algorithm that can be applied to the problem of scenario space clustering. Rather it is to show that clustering is the right path to the definition of a scenario space in a proof of concept manner.

Therefore, hierarchical clustering is chosen as a well-established method with preconditions that can be fulfilled by the underlying data and the distance measure presented in Section 6.2.3. This approach again divides into two subcategories. First, HAC [190] is known as the bottom-up approach, starts with all entities in their own cluster, and iteratively merges them based on their distance. In contrast, the top-down approach divisive hierarchical clustering [208] all

```
Input: labels $\mathbf{l}$
Input: dist_matrix $\mathbf{D}$
Output: dendrogram $\mathbf{h}$
Function primitive_clustering(labels $\mathbf{l}$, dist_matrix $\mathbf{D}$) : dendrogram $\mathbf{h}$ is
    $N \leftarrow |\mathbf{l}|$;
    $\mathbf{h} \leftarrow []$;
    $\text{size}[x] \leftarrow 1 \forall x \in \mathbf{l}$;
    for $i \leftarrow 1$ to $N-1$ do
        $(a, b) \leftarrow \text{argmin}(\mathbf{D})$;
        Append $(a, b, \mathbf{D}[a, b])$ to $\mathbf{h}$;
        $\mathbf{l} \leftarrow \mathbf{l}/\{a, b\}$;
        Add new label $n$;
        foreach $x \in \mathbf{l}$ do
            $\mathbf{D}[x, n] \leftarrow \mathbf{D}[n, x] \leftarrow \text{LINKAGE}(\mathbf{D}[a, x], \mathbf{D}[b, x], \mathbf{D}[a, b], \text{size}[a], \text{size}[b], \text{size}[x])$;
        $\text{size}[n] \leftarrow \text{size}[a] + \text{size}[b]$;
        $\mathbf{l} \leftarrow \mathbf{l} \cup \{n\}$;
return dendrogram;
```

Algorithm B.2: A primitive HAC algorithm according to [190] with $\mathcal{O}(N^3)$ complexity.

entities start in a single cluster and are divided iteratively in subsequent steps. Again the agglomerative approach is chosen in the following without the intention to evaluate if it works better or worse than divisive.

B.4.1. Hierarchical Agglomerative Clustering

The chosen clustering algorithm shall now be explained briefly. The presented procedure is the simplest given in [190], which is most suitable for the understanding of this method. It is now explained with the help of Algorithm B.2.

Let $\mathbf{x}$ be data of arbitrary dimension with N_{smp} samples. As input to the clustering algorithm serve a list of labels $\mathbf{l}$ containing a label for each sample and a distance matrix $\mathbf{D}$. The matrix $\mathbf{D} \in \mathbb{R}^{N_{\text{smp}} \times N_{\text{smp}}}$ is of diagonal symmetric nature and contains the distances between any pair of data points. There exist many different definitions of distance. Its purpose is to quantify similarity and dissimilarity between data points. For coordinates, the most common measure is the Euclidean distance. The output of the algorithm is a dendrogram $\mathbf{h}$ as a list of links between hierarchically structured clusters. From bottom to top in this hierarchy, those clusters

Linkage Criterion	Calculation Formula
Single	$\min\left(\mathbf{D}[a,x], \mathbf{D}[a,x]\right)$
Complete	$\max\left(\mathbf{D}[a,x], \mathbf{D}[a,x]\right)$
Average	$\frac{\text{size}[a]\mathbf{D}[a,x]+\text{size}[b]\mathbf{D}[b,x]}{\text{size}[a]+\text{size}[b]}$
Weighted	$\frac{\mathbf{D}[a,x]+\mathbf{D}[b,x]}{2}$
Ward	$\sqrt{\frac{(\text{size}[a]+\text{size}[x])\mathbf{D}[a,x]+(\text{size}[b]+\text{size}[x])\mathbf{D}[b,x]-\text{size}[x]\mathbf{D}[a,b]}{\text{size}[a]+\text{size}[b]+\text{size}[x]}}$
Centroid	$\sqrt{\frac{\text{size}[a]\mathbf{D}[a,x]+\text{size}[b]\mathbf{D}[b,x]}{\text{size}[a]+\text{size}[b]} - \frac{\text{size}[a]\text{size}[b]\mathbf{D}[a,b]}{(\text{size}[a]+\text{size}[v])^2}}$
Median	$\sqrt{\frac{\mathbf{D}[a,x]}{2} + \frac{\mathbf{D}[b,x]}{2} - \frac{\mathbf{D}[a,b]}{4}}$

Table B.1.: Overview of different linkage criteria for HAC with name and formula [190].

contain more and more data points at the cost of increasing distance between them. The present primitive clustering algorithm calculates these hierarchy links as follows. In the beginning, the list of labels contains a label for each individual data point. Hence, there exist N singleton clusters, indicated at line 4. Next, the algorithm performs one iteration less than available samples as there can be only one less than data points links between those data points. In every iteration, the labels (a, b) for the clusters with minimal distance are selected with the help of $\mathbf{D}$ in line 6. The link between those two is established and saved in the dendrogram with the originating cluster labels and the distance between them in line 7. Line 8 removes the two linked clusters from the list, and following a new label n for the new cluster is created. The distance matrix $\mathbf{D}$ needs to extend so that all clusters have a distance entry to the new cluster, which is done iteratively in lines 10 and 11. Calculating the new distance is not a trivial task without knowledge of the data and used distance measure. Different so-called linkage criteria follow later. At the end of every iteration, lines 12 and 13 calculate the size of the newly created cluster and append its label to the list $\mathbf{l}$. The result of this primitive algorithm is a list of hierarchical links connecting all the available data at different levels of distance. The actually desired cluster can now be chosen by interactively defining a threshold on the links distances that fits the needs of the desired outcome. A minimum working example following later visualizes the process more intuitively. As the algorithm has to evaluate the linkage criterion $\binom{N-1-i}{2}$ times in the ith iteration of line 6, it has a time complexity of $\mathcal{O}(N^3)$. A choice of available linkage criteria follows next.

B.4.2. Available Linkage Criteria

The method defined in Algorithm B.2 still misses a representative for the LINKAGE function in line 11. An overview of different available criteria is given in Table B.1 [190]. Each time two clusters are joined in Algorithm B.2, their new distance to all other clusters needs to be determined. Which of these available linkage function reveals the best result is both dependent on the data itself and the distance metric used. A single best cannot be given at this point. Ward, centroid, and median require the data to be given in a Euclidean space, and the Euclidean distance measures similarity. Other than that, all criteria in the table require the distance measure to be non-negative and symmetric.

B.4.3. Minimum Working Example

Lastly, HAC shall be evaluated briefly with a minimum working example. Therefore, two-dimensional data $\mathbf{x} \in \mathcal{R}^2$ generates from the bivariate normal distribution, spatially distributed into the (x_1, x_2)-plane. These distributions are chosen to be without covariance and are hence defined by two means and two standard deviations as $\mathcal{N}([\mu_1, \mu_2], [\sigma_1 \sigma_2])$, one for each coordinate direction. The data is given as follows

$$\mathbf{x} \sim \begin{cases} \mathcal{N}\left([1.0, 2.0], [0.3, 0.5]\right) \\ \mathcal{N}\left([-1.5, 1.75], [0.3, 0.2]\right) \\ \mathcal{N}\left([-1.0, 0.0], [0.4, 0.2]\right), \end{cases} \tag{B.15}$$

and displayed as scatter in Figure B.6 on the left. In the background, three ovals of vanishing opacity indicate the σ, 2σ, and 3σ ranges for each of the three distributions in green, purple, and red, respectively.

This data is now processed with the HAC algorithm implemented in *scipys*'s *cluster.hierarchy()* function [165]. The chosen distance measure is the Euclidean distance, and complete linkage is applied. A dendrogram on the right of Figure B.6 displays the result. From the bottom to the top, clusters are hierarchically joined based on their distance until all sampled reside in a single cluster. Each join or link indicates through a ⊓ shape with the height as distance between the joined pair. If a threshold is set to 2.5, three clusters remain and show in green, red, and purple colors. Also, the associated scatter points in the left figure are colored accordingly. The links display as lines in between. Certainly, the three colors are set manually to match the ovals in the background. However, it is observable that each determined cluster assembles around a source distribution with no foreign elements present.

The algorithm is successfully able to regroup the samples to their source distribution. Manual tuning and experimenting with different linkage criteria or even distance measures are necessary

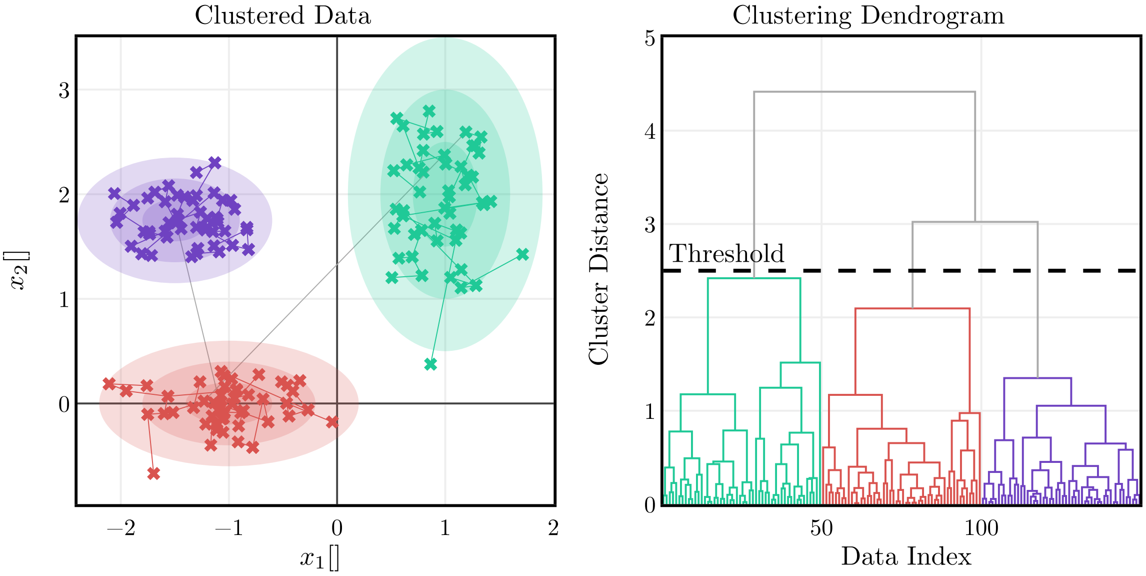


Figure B.6.: Example of HAC on data from three spatially separated bivariate normal distributions. The left plot shows the extend of the distributions by three vanishing ovals indicating the σ, 2σ and 3σ ranges. The data assigned to these clusters is scattered in the same colors, and hierarchical links are shown with thin lines. These links are also visible in the right dendrogram together with the chosen threshold as a dashed horizontal line.

to achieve good results. An application to more complex driving scenario data is discussed in Section 6.2.4.