Fakultät II - Informatik, Wirtschafts- und Rechtswissenschaften
Department für Informatik

Simulation-Based Testing of Cooperative and Automated Vehicles

Von der Fakultät für Informatik, Wirtschafts- und Rechtswissenschaften
der Carl von Ossietzky Universität Oldenburg
zur Erlangung des Grades und Titels eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

angenommene Dissertation

von Herrn Sven Hallerbach

geboren am 26. Februar 1987 in Wiesbaden-Dotzheim

Gutachter:
Prof. Dr. Frank Köster
Prof. Dr. Eric Sax
Tag der Disputation:
03. Juni 2020

Contents

Abstract

The automotive industry faces the major challenge of testing and releasing cooperative and automated vehicles. Among other issues, immense driving distances need to be covered considering state-of-the-art methodologies for test and validation of these vehicles. In practice, which considers besides level 3 systems also level 4 and 5 vehicle technologies, a demand for new methodologies arises.

Additionally, interactions with manual driven vehicles will frequently occur because of the limited market penetration in the beginning. In this context, the handling of non-normative driving behavior caused by human drivers appears to be an important field of research. Further, precision maps, which are used for localization as well as for driving strategy decisions, can be error-prone and differ in a larger, heterogeneous fleet of cooperative and automated vehicles.

In order to support development accompanying tests while simultaneously considering the effects of cooperative and automated vehicles on traffic systems and vice versa, a generic simulation-based toolchain is proposed in this dissertation. These vehicle systems should reach a certain level of maturity before the validation process is initiated and, therefore, only a limited amount of complications should preferably appear. The proposed methodology allows for assessing and identifying critical scenarios that can be tested even during early development stages. Various automation risks, disturbances, and the impact of human driving behavior are investigated and can be parameterized when applying the methodology.

A co-simulation framework is proposed for the verification and identification process. Therefore, a comprehensive simulation environment that combines a vehicle dynamics simulation, which consists of a digital prototype, a traffic simulation providing other test participants, and a cooperation simulation that includes cooperative features, is established. Metrics are used to determine the criticality of a scenario. In this dissertation, safety-related metrics are investigated as well as the impact of cooperative and automated vehicles on traffic quality. Further, the critical scenarios serve as an input for X-in-the-Loop methods, test benches, mixed reality approaches, and driving tests. Finally, a novel approach coupling a real-world prototype with the afore mentioned traffic simulation, referred to as Prototype-in-the-Loop, is proposed.

Kurzfassung

Für die Automobilindustrie bestehen aktuell in den Bereichen des Testens und der Freigabe vernetzter (kooperativer) und automatisierter Fahrzeuge große Herausforderungen. Unter anderem wäre eine extrem hohe Anzahl an gefahrenen Testkilometern für eine Validierung dieser Fahrzeuge mit etablierten Methoden notwendig. Für die Praxis, die heute neben Level 3 Systemen auch bereits auf Level 4 und 5 Fahrzeugsysteme schaut, ist deshalb eine Methodenerneuerung unabdingbar.

Zudem ergibt sich durch die anfänglich geringe Marktdurchdringung automatisierter Fahrzeuge eine erhöhte Interaktion mit manuell gefahrenen Fahrzeugen. In diesem Zusammenhang ist zum Beispiel der Umgang mit nicht-normativem Verhalten menschlicher Fahrer ein wichtiges Handlungsfeld. Auch die digitalen Karten, die zur Lokalisierung und Fahrstrategieentscheidung der Fahrzeuge verwendet werden, können fehlerhaft sein bzw. innerhalb einer größeren heterogenen Flotte voneinander abweichende Sachverhalte darstellen.

In dieser Dissertation wird eine generische simulationsbasierte Werkzeugkette vorgestellt, welche das entwicklungsbegleitende Testen unterstützt sowie die in direkter Wechselwirkung stehenden Effekte von kooperativen und automatisierten Fahrzeugen und Verkehrssystemen berücksichtigt. Fahrzeugsysteme sollten schon vor dem Validierungsprozess so weit ausgereift sein, dass bei der eigentlichen Validierung möglichst wenige Komplikationen auftreten. Die vorgeschlagene Methodik ermöglicht die Identifikation und Prüfung kritischer Szenarien, welche bereits entwicklungsbegleitend untersucht werden können. Verschiedene Automatisierungsrisiken, Störungen und menschliches Fahrverhalten werden in der Methodik mitberücksichtigt und können sehr gut parametrisiert werden.

Der Prüf- und Identifikationsprozess wird durch ein gekoppeltes Simulationsframework umgesetzt. Eine Kombination aus einer Fahrdynamiksimulation, welche einen digitalen Prototyp enthält, einer Verkehrssimulation, die andere Verkehrsteilnehmer simuliert und einer Kooperationssimulation, welche kooperative Funktionen beinhaltet, wird genutzt, um eine umfassende Simulationsumgebung zu etablieren. Die Kritikalität der Szenarien wird durch Metriken bestimmt. In dieser Dissertation werden sicherheitskritische Metriken verwendet und zusätzlich die Auswirkungen von kooperativen und automatisierten Fahrzeugen auf die Verkehrsqualität untersucht. Diese kritischen Szenarien dienen als Input für X-in-the-Loop Methoden, Prüfstände, gemischt-reale Ansätze und Versuchsfahrten. Zuletzt wird eine ergänzende neue Methode (Pro-

totype-in-the-Loop), welche die Kopplung eines realen Versuchsträgers mit der vorher genannten Verkehrssimulation ermöglicht, vorgestellt.

List of Figures

List of Tables

Abbreviations & Nomenclature

Abbreviation	Definition
ADAS	Advanced Driver Assistance Systems
ASIL	Automotive Safety Integrity Level
AV	Automated Vehicle
CAE	Computer Aided Engineering
CAV	Cooperative and Automated Vehicle
COG	Center of Gravity
DOI	Domain of Interest
EH/LH	Enter Highway / Leave Highway
FPR	False Positive Rate
HiL	Hardware-in-the-Loop
HCM	Highway Capacity Manual
KL	Keep Lane
LCL/LCR	Lane Change Left / Lange Change Right
LOS	Level of Service
MiL	Model-in-the-Loop
NA	Not Active
ODD	Operational Design Domain
OOP	Object Oriented Programming
PC	Passenger Car
PiL	Prototype-in-the-Loop
ROC	Receiver Operator Characteristics
SAE	Society of Automotive Engineers
SiL	Software-in-the-Loop
SLAM	Simultaneous Localization and Mapping
SOTIF	Safety of the Intended Functionality
TPR	True Positive Rate
TTB/TTC	Time to Brake / Time to Collision
V2X/V2V	Vehicle to X / Vehicle to Vehicle
V&V	Verification and Validation
VAAFO	Virtual Assessment of Automation in Field Operation
ViL	Vehicle-in-the-Loop

Symbol/ Acronym	Definition	Unit
β	Vehicle Dynamics Model: Slip Angle	n.a.
$\beta_1, \beta_2, \beta_3, \beta_4$	Metrics Optimization: Weighting Parameters	n.a.
$\boldsymbol{\beta}$	Metrics Optimization: Weighting Parameter Vector	n.a.
δ	Vehicle Dynamics Model: Wheel Angle	n.a.
Δp	Quality Metrics Ego-Vehicle: Difference of Vehicle Positions	m
σ_{a}	Quality Metrics Traffic: Individual Standard Deviation of Vehicle Acceleration	m/s^2
$\sigma_{\mathrm{a,ref}}$	Quality Metrics Traffic: Reference Value for Individual Standard Deviation of Vehicle Acceleration	m/s^2
$\sigma_{v_{\mathrm{Circle,j}}}$	Quality Metrics Traffic: Nanoscopic Velocity Standard Deviation	m/s
σ_{vj}	Quality Metrics Traffic: Microscopic Velocity Standard Deviation	m/s
τ	Multi-Agent Simulation: Reaction Time of the Driver Behavior Model	s
Ψ	Vehicle Dynamics Model: Yaw Angle	n.a.
a	Multi-Agent Simulation: Acceleration of the Driver Behavior Model	m/s^2
$a_{\mathrm{ego,max}}$	Quality Metrics Ego-Vehicle: Maximum Deceleration of the Ego-Vehicle	m/s^2
a_{obj}	Quality Metrics Ego-Vehicle: Object Acceleration	m/s^2
a_{req}	Quality Metrics Ego-Vehicle: Required Deceleration	m/s^2
b	Multi-Agent Simulation: Braking Function of the Car Following Model	n.a.
$\boldsymbol{b}_{\mathrm{l}}$	Metrics Optimization: Vector of Lower Bound Constraints	n.a.
$\boldsymbol{b}_{\mathrm{u}}$	Metrics Optimization: Vector of Upper Bound Constraints	n.a.
$\overline{CV}$	Quality Metrics Traffic: Microscopic Mean Coefficient of Variation	n.a.
CV_{j}	Quality Metrics Traffic: Microscopic Coefficient of Variation	n.a.
CV_{ref}	Quality Metrics Traffic: Reference Value for the Microscopic Mean Coefficient of Variation	n.a.
D	Quality Metrics Traffic: Traffic Density	pc/m/lane
$\overline{DV}$	Quality Metrics Traffic: Nanoscopic Mean Coefficient of Variation	n.a.

Symbol/ Acronym	Definition	Unit
DV_{j}	Quality Metrics Traffic: Nanoscopic Coefficient of Variation	n.a.
DV_{ref}	Quality Metrics Traffic: Reference Value for the Nanoscopic Mean Coefficient of Variation	n.a.
D_{veh}	Quality Metrics Traffic: Traffic Density Obtained from Simulation	veh/m/lane
E_{T}	Quality Metrics Traffic: Passenger Car Equivalents (Trucks, Busses)	n.a.
E_{R}	Quality Metrics Traffic: Passenger Car Equivalents (RVs)	n.a.
$F_{\mathrm{xf}}, F_{\mathrm{xr}}, F_{\mathrm{yf}}, F_{\mathrm{yr}}$	Vehicle Dynamics Model: Longitudinal Tire Force (front), Longitudinal Tire Force (rear), Lateral Tire Force (front), Lateral Tire Force (rear)	kgm/s^2
f_{p}	Quality Metrics Traffic: Driver Population Factor	n.a.
f_{HV}	Quality Metrics Traffic: Heavy-Vehicle Adjustment Factor	n.a.
$G_{\mathrm{mac}}, G_{\mathrm{mic}}$ $G_{\mathrm{nan}}, G_{\mathrm{ind}}$	Quality Metrics Traffic: Macroscopic Grade, Microscopic Grade, Nanoscopig Grade, Individual Grade	n.a.
$\overline{G}_{\mathrm{final}}$	Metrics Optimization: Mean Final Grad	n.a.
G_{final}	Metrics Optimization: Weighted Final Grade	n.a.
$g(t)$	Multi-Agent Simulation: Gap for Car Following Model	m
I_{z}	Vehicle Dynamics Model: Moment of Inertia	kgm^2
l_{f}	Vehicle Dynamics Model: Distance to the Vehicle's Front from the COG	m
l_{r}	Vehicle Dynamics Model: Distance to the Vehicle's Rear from the COG	m
LOS	Quality Metrics Traffic: Level of Service	n.a.
m	Vehicle Dynamics Model: Mass	kg
N_{L}	Quality Metrics Traffic: Number of Lanes	n.a.
P_{T}	Quality Metrics Traffic: Proportion of Trucks/Busses	n.a.
P_{R}	Quality Metrics Traffic: Proportion of RVs	n.a.
P_{HF}	Quality Metrics Traffic: Peak-Hour Factor	n.a.
S	Quality Metrics Traffic: Average Travel Velocity	m/s
S_{N}	Parameter Variation Module: Number of Possible Scenarios	n.a.
T	Multi-Agent Simulation: Time Step Size	s

Symbol/ Acronym	**Definition**	**Unit**
TTB	Quality Metrics Ego-Vehicle: Time to Brake	s
TTC	Quality Metrics Ego-Vehicle: Time to Collision	s
$\boldsymbol{u}$	Vehicle Dynamics Model: Input Vector	n.a.
$\bar{v}_\mathrm{j}$	Quality Metrics Traffic: Microscopic Mean Velocity of Every Vehicle	m/s
$\bar{v}$	Quality Metrics Traffic: Mean Velocity of Every Vehicle with Respect to Time Interval	m/s
v_ref	Quality Metrics Traffic: Reference Value Mean Velocity of Every Vehicle	m/s
v	Vehicle Dynamics Model: Velocity	m/s
$\bar{v}_\mathrm{Circle,j}$	Quality Metrics Traffic: Nanoscopic Mean Velocity of Every Vehicle	m/s
v_des	Multi-Agent Simulation: Desired Velocity of the Driver Behavior Model	m/s
v_safe	Multi-Agent Simulation: Safe Velocity of the Driver Behavior Model	m/s
v_max	Multi-Agent Simulation: Maximum Velocity of the Driver Behavior Model	m/s
v_l	Multi-Agent Simulation: Velocity of the Leading Vehicle	m/s
v_f	Multi-Agent Simulation: Velocity of the Following Vehicle	m/s
v_P	Quality Metrics Traffic: Passenger Car Equivalent Traffic Flow Rate	pc/s/lane
v_ego	Quality Metrics Ego-Vehicle: Ego-Vehicle Velocity	m/s
v_obj	Quality Metrics Ego-Vehicle: Object Velocity	m/s
v_rel	Quality Metrics Ego-Vehicle: Relative Velocity	m/s
v_i	Parameter Variation Module: Number of Values Selected for Individual Parameters	n.a.
$\bar{v}_\mathrm{ego}$	Quality Metrics Traffic: Average Velocity of the Ego-Vehicle with Respect to Time Interval	m/s
V	Quality Metrics Traffic: General Traffic Flow Rate	veh/s/lane
$\boldsymbol{x}$	Vehicle Dynamics Model: State Vector	n.a.
x_1, x_2, x_3, x_4	Metrics Optimization: Individual Grades for Parameter Optimization (Calculated by Metrics)	n.a.
$\boldsymbol{X}$	Metrics Optimization: Matrix of Grades (Calculated by Metrics)	n.a.
$\boldsymbol{y}$	Metrics Optimization: Vector of Grades (Evaluated by Experts)	n.a.

1 Introduction

1.1 Motivation

The development of Cooperative and Automated Vehicles (CAVs) is progressing rapidly. Automotive manufacturers, suppliers as well as start-up companies push to release CAVs to the market as soon as possible, because the benefits of operating these systems are tremendous. Next to the economic aspects CAVs pledge to reduce emissions, increase traffic efficiency, raise driver comfort, productivity, and most notably diminish accidents.

It is assumed that over 90 percent of accidents are caused by human driver insufficiencies [1, pp. 11-12]. Thus, the potential of increasing safety is one of the major reasons to deploy CAV technologies. At this point, one of the biggest challenges for the development of CAVs arises: It is expected that a vast driving distance has to be covered to confirm that CAVs operate at least as safe as a human driver [2]. Further, to increase traffic safety CAVs have to perform even better than human drivers. This line of argumentation on its own does not consider important aspects in detail. There are complex interactions between CAVs and other traffic participants that have an impact on safety. For example, non-normative driving behavior of other traffic participants poses a challenge for CAV systems. Further, the incorporation of automation causes so-called automation risks [3], which currently are not accounted for in traffic safety considerations. Another important aspect is the general acceptance of CAVs in society. At present, the acceptance of these systems is not clearly foreseeable and the responses of society due to accidents caused by CAVs are not predictable. Therefore, ensuring an accepted degree of CAV safety is still a challenging task up to this point, which triggers controversial discussions in the scientific community. It can be agreed upon that established test and validation methods are not sufficient to ensure CAV safety. New approaches have to be elaborated to enhance current methods for testing and validation.

The problem that arises lies in the change of the responsibility shifting from human drivers to the automation modules. It is clearly stated by the Society Of Automotive Engineers (SAE) that CAVs can be categorized as level 3 or higher [4]. This means that the automation must be able to handle critical scenarios relying on their implemented driving functions.

Due to the fact that these systems are very complex themselves and the domain they are supposed to operate in is even more complex, comprehensive testing is mandatory. More details are illustrated in Figure 2.4.

The complexity that arises on one hand is caused by different algorithms that have to operate properly on their own and in collaboration with each other. For example, the vehicle guidance controller relies strongly on sufficient localization results. Therefore, if the localization does not work properly, the controller functionality is confined as well. Currently, the embedded sensors of CAVs are extended by the use of precision maps. These maps are supposed to provide a larger prevision and detailed information on road properties, such as lane geometries and markings. Precision maps supply helpful information regarding driving strategy, localization, trajectory planning, etc.

On the other hand, a major challenge for CAVs is caused by the domain they are operating in. CAVs are supposed to operate in public traffic systems, e.g., highways, rural roads, urban areas, etc. Assuming that the implemented algorithms are working properly, the interactions between CAVs and human drivers are quite difficult to handle. Especially in crowded traffic systems human drivers tend to disregard traffic regulations, which aggravates the driving task for CAVs even further. Additionally, the driving behavior of some human drivers, which ranges from defensive to aggressive, poses another challenge for CAVs to operate safely.

Even though the validation process is very demanding, the CAVs that are handed over from the development departments have to be tested intensively to a certain maturity as well. Too many feedback loops from validation back to the development departments need to be avoided. Therefore, even before the validation process begins, comprehensive testing of CAV driving functions during development is absolutely necessary. It is not possible to endanger safety of public traffic systems for the validation of CAVs that have not been tested intensively during the development process. Basically, the CAV needs to be in a mature stage, before it can be tested in public traffic systems exposing humans. For the acceptance of CAVs in general, tests with immature driving functions causing accidents are unacceptable. Furthermore, CAVs driving through public traffic systems should contribute to the improvement of traffic quality. It is generally expected that CAVs are able to rectify or at least do not worsen traffic quality.

All of these aspects require new concepts for performing sufficient structured testing of CAVs even during development phases. Additionally, the interactions between CAVs and traffic systems need to be considered in the testing procedure. CAVs and traffic systems are influencing each other simultaneously, which is an important aspect that needs to be accounted for during development accompanying tests even in early stages.

1.2 Problem Statement

As already indicated, comprehensive testing during the development process is necessary for many reasons. One of the major issues for testing in general occurs because of the complexity of the CAV algorithms and the domain they are supposed to operate in. It is quite difficult to find out which tests have to be performed and how tests can be evaluated properly. Nowadays, so-called scenario catalogs are developed for CAV testing [5]. Depending on the CAV specification a lot of effort is spent on deriving scenario catalogs that contain tests that have to be performed. This approach relies largely on expert knowledge and experience. It can be easily foreseen, that this proceeding is not sufficient and even the most experienced experts are not able to think of every scenario that could occur in such a complex system. Additionally, with that approach the amount of tests that have to be performed is tremendous and can not be conducted economically. The tests need to be shifted to different test environments, for example, simulation or test benches, in order to make the test effort feasible. Anyway, the rather more difficult challenge is the identification of scenarios experts did not think of in the first place.

A structured approach is needed for a systematical identification of scenarios that have to be tested. One could say that the expert knowledge needs to be enhanced with other methods to achieve comprehensive testing.

As already mentioned, CAVs rely on many different algorithms working together in a complex domain. One method to test CAV systems is the concept of digital prototypes and digital twins as shown in Figure 1.1.

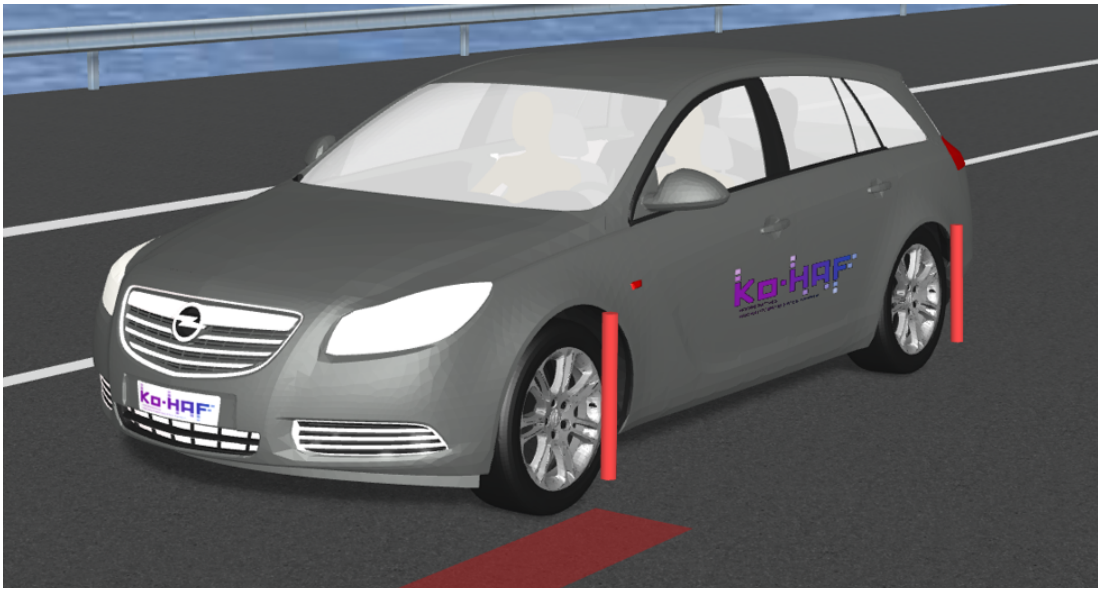

Figure 1.1: Digital prototype of a CAV.

The difference between a digital prototype and a digital twin depends on the development stage of the CAV. In this dissertation, a CAV is called digital prototype until the development of the CAV is in a mature stage. After reaching this development stage the digital prototype becomes a digital twin.

Digital prototypes and digital twins are widely used to create models of technical systems, for example, in manufacturing, aerospace, robotics, etc. The motivation for this course of action lies in the advantage to be able to test technical systems without causing damage to hardware as well as finding errors early and, preferably, even before the system is built [6].

In this dissertation the digital prototype is a model of the CAV that needs to be tested. Based on the specifications of the CAV, features are included in the digital model. The automation spectrum begins with so-called Advanced Driver Assistance Systems (ADAS) that aim to support the driver performing driving tasks. Systems, such as adaptive cruise control, lane keeping assist, or combinations, can be categorized as assisting features (SAE level 1-2), while the responsibility remains by the driver. Systems that are categorized SAE level 3 and higher are called CAVs or Automated Vehicles (AVs) if cooperative aspects are left out. In this case, the automation relies on the implemented functions even in dangerous situations and the responsibility shifts from human drivers to the automation modules [4].

The digital prototype contains the vehicle dynamics of the CAV as well as the sensor setup including the driving functions that are implemented. This digital model operates in a digital world containing roads, road markings, traffic signs, etc. Broadly speaking, the world the digital prototype is driving in can be considered as a digital twin of the surrounding environment. Next to the afore mentioned static attributes the digital world consists of many dynamic aspects as well, for example, vehicles, cyclists, pedestrians, traffic lights, etc.

Additionally, if cooperative aspects are considered, the CAVs are connected with each other as well as with the infrastructure surrounding them. In this case, there arises a need for digital prototypes and digital twins for cooperative features. This means that the digital world becomes cooperative and controllable. Thus, this raises the demand for new testing methods as well.

The algorithms controlling the CAV are complex automation modules that have to operate together. Thereby, the CAV senses its environment by different sensors simultaneously and fusions the measurements to one environment model. Based on this model, driving decisions and trajectory planners plan the vehicle motion accordingly. The vehicle guidance is carried out by a trajectory following controller that controls the actuators of the CAV. All in all it is a difficult task to operate and synchronize these automation modules properly.

The error sources are manifold. Even helpful enhancements, e.g., using precision maps, can lead to problems, if the maps contain errors. It is very likely that the used precision maps are not always accurate. Figure 1.2 shows a map error and its aftereffects on traffic quality.

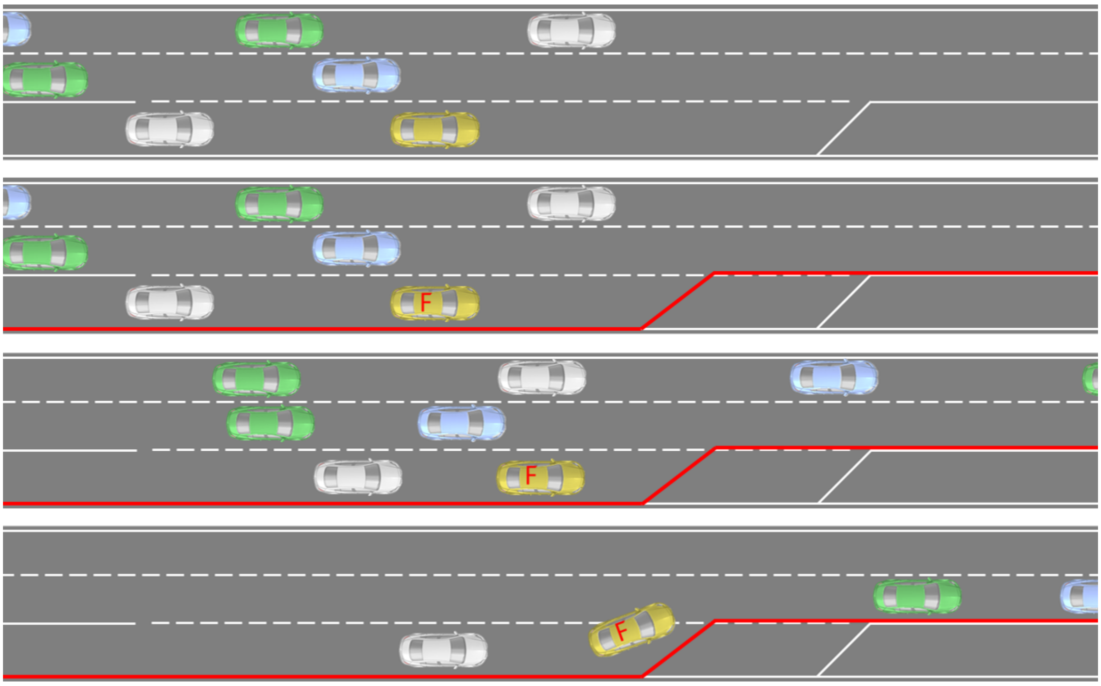

Figure 1.2: Critical scenario caused by a map error (incorrect lane end) of the CAV marked in yellow. Initial state shown at the top. Final state shown at the bottom [7, p. 371].

In this scenario, all vehicles are CAVs possessing a correct precision map, except for the yellow one. Normally, the CAV marked in yellow would have overtaken the CAV on the left lane marked in blue. The map error causes the yellow CAV's driving functions to brake, because the trajectory planner falsely predicts that there is not enough space to overtake the blue CAV. The blue CAV, which possesses a correct map, also brakes to let the yellow CAV pass. In this moment, both CAVs brake and block the entrance ramp and the right lane of the highway, which causes other vehicles to brake, too. Depending on the driving strategy the blue CAV will decide to pass the yellow one, but for a certain time interval both vehicles are braking. In crowded traffic systems such a behavior could lead to a serious congestion decreasing the traffic quality significantly.

This is one example of scenarios that have to be tested. In general, map errors can lead to safety critical scenarios as well as a significant decrease in traffic quality.

The next example aims to illustrate human driving behavior as a potential risk for CAV safety. Figure 1.3 shows a CAV that enters the highway, while two human drivers possessing aggressive and defensive driving behavior participate as well.

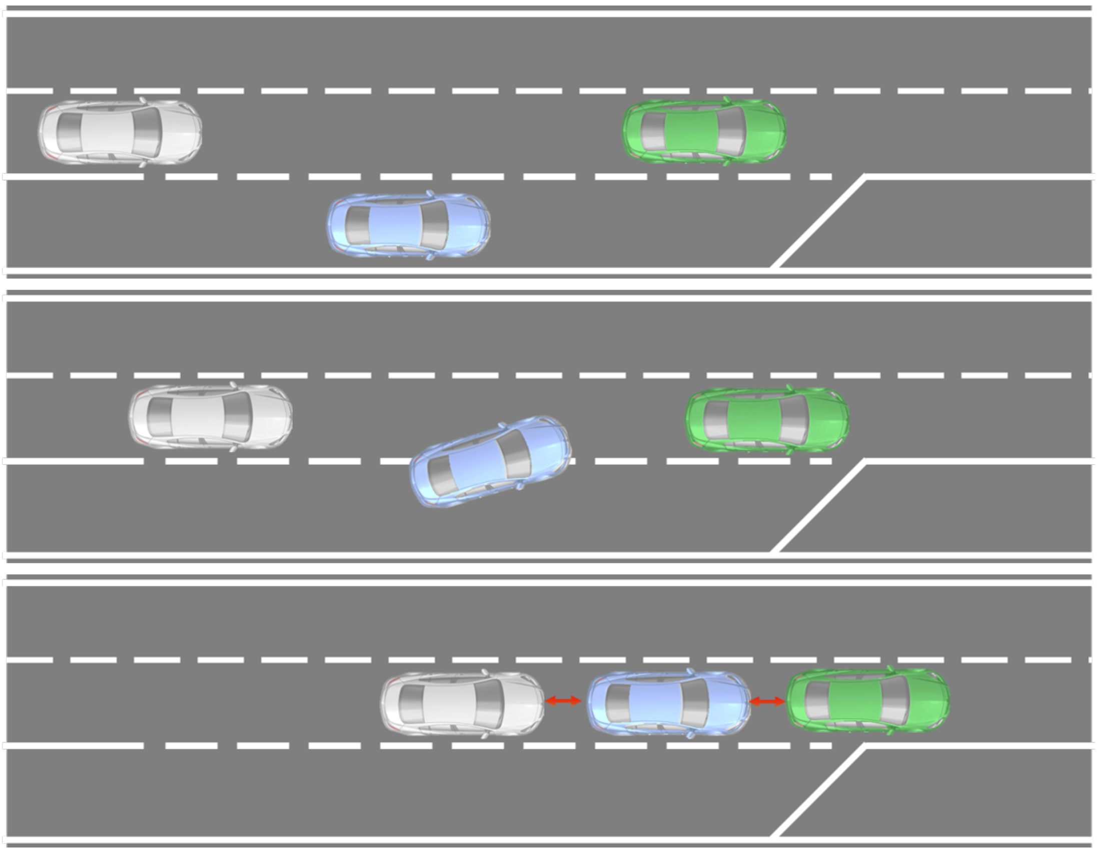

Figure 1.3: Critical scenario caused by human driving behavior. CAV colored in blue. Aggressive driver colored in gray. Defensive driver colored in green. Initial state shown at the top. Final state shown at the bottom [8, p. 94].

Even though the CAV enters the highway because of a sufficient gap size in the beginning, the aggressive driver marked in gray tailgates the CAV, while the defensive driver marked in green drives slowly. That leads to a critical scenario, where the CAV has limited space in both directions caused by human drivers. This represents also a critical scenario that requires testing.

These so-called critical scenarios need to be investigated closely and the behavior of the CAV has to be precisely observed. The afore mentioned types of critical scenarios are likely to happen and even if the CAV is not always the reason for the occurrence, it needs to be ensured that the CAV operates appropriately.

1.3 Outline

This dissertation is structured in five main chapters. It starts with a general introduction, motivation, and problem statement elaborating on the main aspects and challenges in the field of CAV testing.

After that, the Chapter State of the Art and Research Objectives states current test methodologies as well as challenges for testing CAVs in general. Fundamentals considering simulation-based testing are provided, where the aim is to explain how simulation methodologies constitute a major benefit for this enormous task. To state the motivation for this dissertation research objectives are defined and the current need for improvement is clarified based on the state-of-the-art and related work in the scientific and industrial domain. Therefore, the research objectives are compared to related work and current research projects that are working in the same field of research.

Chapter 3 contains the methodology proposed in this dissertation. The main focus of this chapter is the proposal of a generic simulation-based toolchain for development accompanying tests of CAVs. In order to explain the toolchain, each aspect is accounted for and specific details are provided on how to use the methodology. Throughout the chapter a co-simulation approach, coupling a vehicle dynamics simulation that contains a digital prototype of a CAV with a traffic simulation, is applied while the simulation runs are evaluated by metrics. The developed metrics are considering safety related issues as well as CAVs interacting with traffic systems.

Further, a mixed reality approach for enhancing current test methods is introduced, implemented, and elaborated. The major benefit of the so-called Prototype-in-the-Loop approach aims for coupling a real-world CAV prototype driving on a proving ground with virtual participants that are controlled by a traffic simulation tool. After the methodologies are explained, the manifold contributions of this dissertation are stated.

Chapter 4 contains the results of the performed tests based on a run-through the entire toolchain. First, the results of the co-simulations are presented proving that the methodology is applicable and functional. Further, the results of the mixed reality approach are illustrated by implementing the method in a real-world CAV prototype. The tests are performed on a proving ground and the results are analyzed based on the driving test data of the CAV.

Finally, Chapter 5 concludes by summarizing the dissertation. Additionally, an outlook on future work and recommended proceedings to enhance and apply this methodology is provided.

2 State of the Art and Research Objectives

This chapter contains the state-of-the-art development and test methods for ADAS as well as for CAVs. Starting with current test methods, the chapter advances to specific test environments and their advantages as well as their constraints. Furthermore, this chapter contains the content of tests based on expert opinions and the specific challenges of testing CAVs. An overview is provided on the difficulties arising through the complexity of CAV driving functions and the demanding methods required to test and assess the functionality of these systems.

Especially, the assessment of the performed tests can be considered as a challenging task. Usually, the evaluation of a test is determined by metrics. In the case of CAV testing, a vast variety of metrics need to be considered. The different metrics correlate with each other. This means that it is not always possible to satisfy each single metric sufficiently and simultaneously. A brief overview on that topic is provided and the main aspects are asserted.

A brief introduction on simulation-based development and testing is provided covering fundamentals for simulation schemes in general. Thus, the standard methodology for performing simulation-based tests is given. Fundamentals regarding vehicle dynamics and multi-agent simulation approaches aim for a better understanding of the used simulation methods in this dissertation.

The simulation fundamentals are followed by defining the research objectives and comparing these with related work and state-of-the-art research projects that are contributing to this field of interest. Specific requirements are deduced from the research objectives and compared with those of the state-of-the-art research projects. Based on that process, gaps and additional research benefits are stated and the scope of the dissertation is specified. Furthermore, an additional overview about related work throughout academic and industrial research is provided.

Finally, the chapter ends with a conclusion on the current state-of-the-art and elaborates on the specific requirements that are going to be fulfilled by this dissertation aiming to establish a comprehensive simulation-based toolchain for development accompanying tests of CAVs.

2.1 Development and Test Methods for Cooperative and Automated Vehicles

ADAS are a partial set of automotive electronics and therefore, the established development and test methods from this field serve as a basis for CAVs. The development process for automotive electronics is based on a V-model providing specific proceedings for development and validation [9, 10]. Figure 2.1 shows the V-model for the development process of automotive electronics.

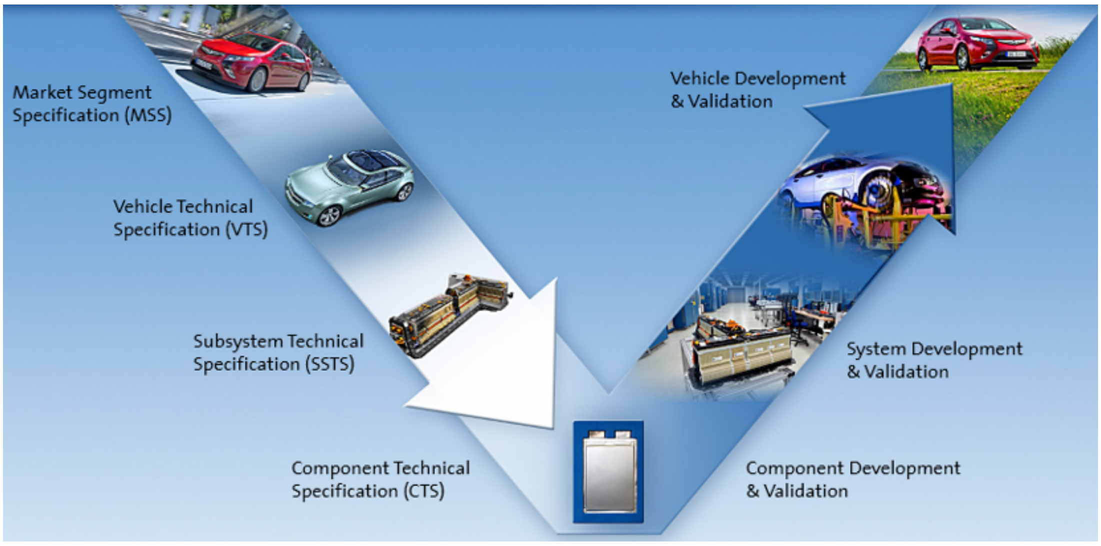

Figure 2.1: V-model for the development and validation of automotive electronics [10, p. 22].

The V-model can be divided into two parts. The left part of the V describes the vehicle development and the right part the validation process. Usually, these parts are divided to avoid that the development engineers validate their own functions. Proceeding like this ensures that the validation process is led by engineers that are not involved in the development process. Thus, the performed validation tests can be considered as unbiased. However, the safety validation of CAVs is challenging. According to [2], a vast driving distance has to be covered to ensure that CAVs perform at least as good as human drivers. Even though the validation poses a difficult challenge, it is necessary to improve the development process of the cooperative and automated functions as well. Hence, the CAV's development status should be in a mature state before the validation process is initiated.

Typically, the used test methods and test environments are dependent on the development status of the CAV. Even though the actual vehicle has not been built yet, it is possible to perform Model-in-the-Loop (MiL) [11, 12] tests based on vehicle models.

Further, Software-in-the-Loop (SiL) [13] tests are feasible, before the components are developed. Basically, on the left side of the V-model tests can be performed before the actual vehicle production has started. On the right side of the V-model, parts, components, or even the vehicle development has to be in a mature state. If developed and built, Hardware-in-the-Loop (HiL) [14, 15, 16], Vehicle-in-the-Loop (ViL) [17, 18, 19] and actual driving tests can be performed. Of course, there are feedback loops from validation to development indicated by the arrows in Figure 2.2.

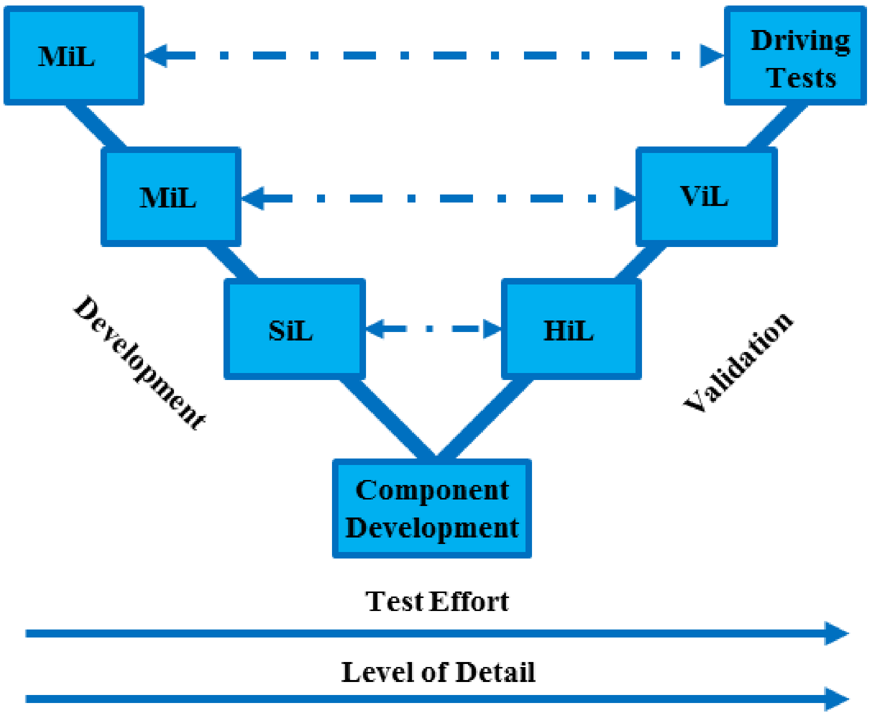

Figure 2.2: Virtual V-model for the development and validation of automotive electronics [8, p. 96], see also [20].

Awareness of the validation process is used to enhance and improve development tests. For example, the mathematical models used in the MiL environment can be improved by data gathered by driving tests. During the walk through of the V-model the level of detail increases. The gap between reality and test environment decreases, while the test effort increases simultaneously. The more the level of detail increases, the efforts and costs to perform a test are increasing as well [2, p. 460]. Taking this knowledge into account, it is obvious that a poor developed function results in a vast test effort on the right side of the V-model as well as an increase of the required feedback loops from right to left. Usually, MiL tests are solely virtual and based on simulation tools. SiL tests are used to run the compiled software without considering the hardware, where the software is implemented on. This provides the possibility of software testing in early development stages.

HiL tests are performed with actual components integrated into a test bench. The test procedure starts by dismantling components of the vehicle. These components are mounted on specifically designed test benches, where component tests are performed. The ViL environment requires the actual vehicle and is used either for human factor purposes or for performing tests with virtual test participants, while the actual vehicle-under-test operates. This approach is a trade-off between driving tests and virtual testing. The advantages are manifold, for example, dangerous tests can be performed, tests can be reproduced without using test robots. The disadvantages lie in the neglect of the sensor setup. The virtual test participants' attributes are usually known and do not have to be captured by the sensors. Therefore, a reality gap between ViL and actual driving tests exists. Driving tests are the last and most expensive step in the V-model. The test effort is very high and for that reason these tests should be reduced to a necessary minimum.

The test requirements are usually developed based on expert opinions. Based on requirements engineering [21, 22], test catalogs are prepared. For CAVs the state-of-the-art proceeding for testing and validation starts with a use case description. A use case depends on the CAV function that needs to be tested. For example, if the evaluated driving function is a highway chauffeur [23, p. 53], the general use cases could be: enter the highway, leave the highway, follow another vehicle, overtake, etc. These use cases are then divided into different tests. For a highway chauffeur possible different examples are shown in Figure 2.3. The CAV's trajectory remains unknown until the test is completed, because it depends on the implemented driving functions. Only the trajectories of the test participants are predefined.

The test evaluation is based upon metrics that are developed by experts. Standard evaluation metrics can be manifold. Foremost, safety critical and regulatory metrics are used for CAV test evaluation. For example, time gap [24] and time to collision [25] are often used for automated driving test cases. On the other hand, metrics, such as driver comfort, traffic efficiency, travel time, can be applied for CAVs as well [26]. Another question that arises is conformance with regulatory requirements. Regulations define traffic rules that have to be obeyed. This poses a lot of difficulties for the CAV, because human drivers sometimes ignore these regulations, which can be challenging for CAV driving functions. Problems, such as close gap cut-ins pose a problem that has to be considered by tests. Even though human drivers are generally not allowed to do this, it can be seen that in highly populated traffic systems these cut-ins often occur. This aspect can be considered as a problematic constraint especially for driving tests. If driving tests are performed, the safety of test drivers has to be ensured. This means that performing these tests is too dangerous even on proving grounds. These tests are then shifted to simulations in which harmful behavior does not affect test drivers.

Another challenge is the reproducibility of tests driven by human test drivers. It is almost impossible to guarantee that a test is performed in a reproducible manner, since human test drivers are not able to repeat tests accurately.

Here, one solution is the usage of test robots. Test robots are at least able to guarantee reproducibility. Furthermore, there are test robots that are equipped with soft targets. The purpose of soft targets is to run safety critical tests without causing harm to test drivers and the CAV itself [27]. The usage of test robots is quite expensive and constitutes an economic constraint.

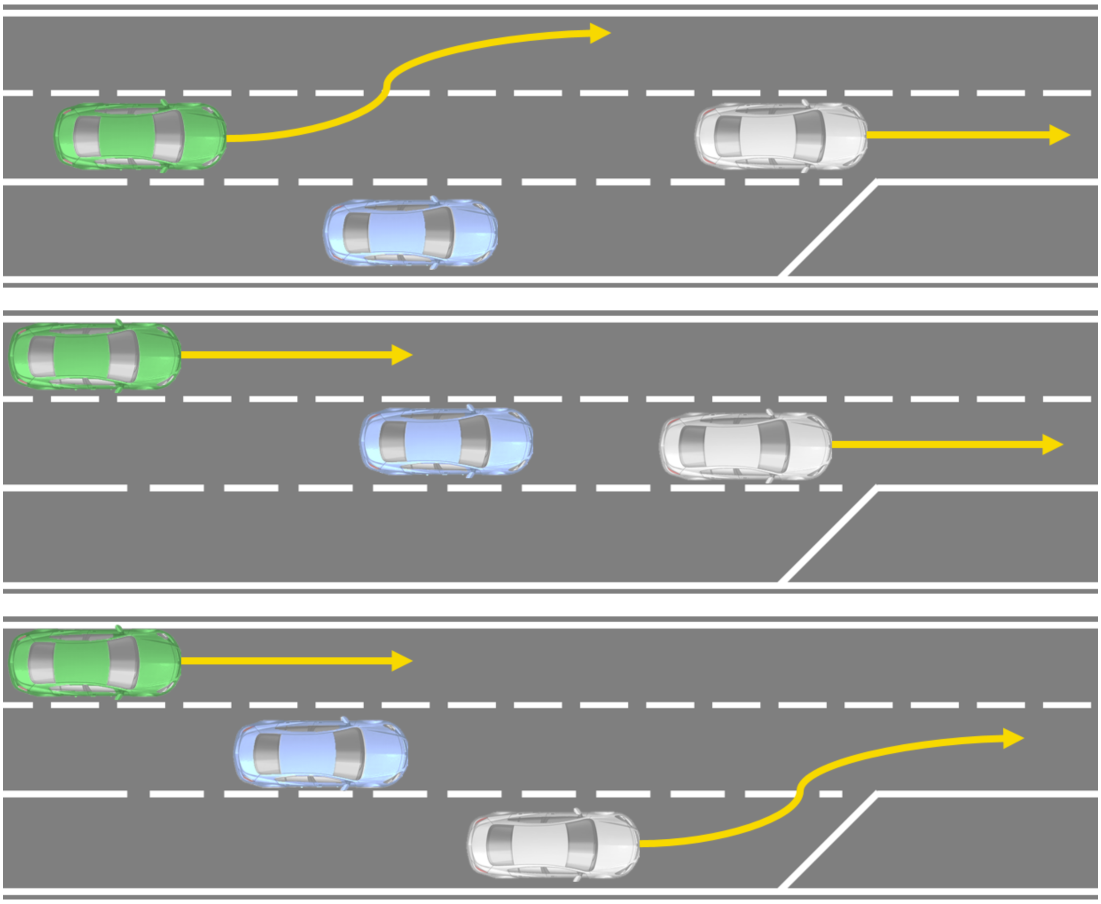

Figure 2.3: Different example scenarios for CAV testing. CAV colored in blue. Other test participants colored in green and gray.

Furthermore, the tests captured in a catalog have to be varied. Moreover, due the necessity of test variations as well as the complexity and multiplicity of possible parameters the test quantity is high. By varying the parameters additional test cases are being generated. Figure 2.4 shows possible parameter variations of a specific test. It is obvious that the variety increases with every additional test participant and every parameter that is accounted for. The performed tests have to be executed again for every new developed driving function. In addition, if a test case is not passed, the test has to be repeated. All of this points to the fact that testing of CAVs is complex and demanding. Thus, a well-studied research question is how to optimally distribute the identified test set across the various test environments.

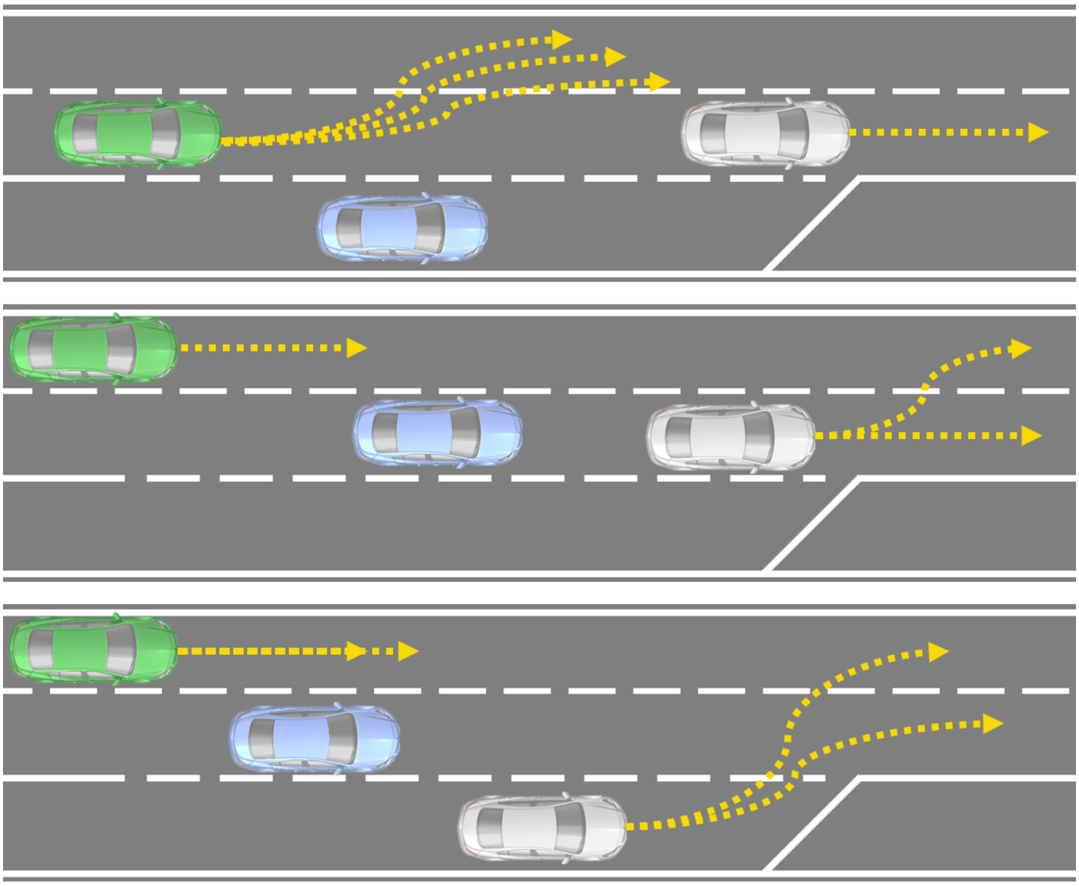

Figure 2.4: Different example scenarios for CAV testing including parameter and trajectory variations. CAV colored in blue. Other test participants colored in green and gray.

In order to get an impression on the complexity and test variety, which depend strongly on the number and step sizes of varied scenario parameters, a brief example is presented. Imagine that two parameters for each test participant are varied, for example, the initial conditions of the position and velocity in 14 and 12 steps, respectively. This summarizes to 28 224 variants just for the initial conditions of the scenario [28, ch. 7.2.2]. The vast variety of test cases enforces a distribution of tests to different test environments. Some tests are too dangerous for actual driving tests executed by test drivers and need to be performed by simulation tools. Other tests, which are of high relevance, need to be performed by actual driving tests. For some tests vehicle components can be mounted on a bench and therefore, the tests can be performed with less test effort. The decision depends on the specific requirements and constraints of a test. Additionally, existing data bases are used to find relevant test cases. Accident data [29] or driving studies [30] are accessed to enhance catalogs. The usage of data is important to identify tests experts did not think of in the first place. These tests are collected and added to the catalog and again, the parameters can be varied to derive supplemental tests. This method is also used to define critical areas and function boundaries based on metrics. It is important to find the boundaries of a driving function for an assessment of the CAV's capabilities. A realistic estimation of the CAV's capabilities will be crucial to get an approval for releasing these vehicles. Particularly, the safety confirmation is indispensable for launching CAVs into public traffic systems. The existing guideline for this confirmation is stated by the ISO-26262 [31]. The ISO-26262 is derived from the basic norm IEC 61508 for functional safety of electrical/electronic/programmable electronic safety-related systems [32] that aims to confirm safety goals. The norm aims to reduce the risk that originates from a system to a tolerable risk.

In general, the term risk is defined in engineering related fields as the product of the probability of occurrence and severity of damage. The ISO-26262 provides a methodology on how to confirm that the potential risk of a system is reduced to a tolerable risk called residual risk. Tolerable risk orients itself on the technical and scientific state of the art [33, ch. 2.5]. The proceeding leans on the V-model for development and validation of automotive electronics and is initiated by a concept phase starting with an item definition. Therefore, functional requirements, safety-related requirements, and boundaries are described [34]. Following that, the safety life cycle is initiated clarifying what development category applies. Thus, there can be categories, such as existing item or newly developed item. After that, the hazard analysis and risk assessment is carried out. This is done by determining the Automotive Safety Integrity Level (ASIL) with help of a determination matrix that consists of the factors severity, exposure, and controllability [34]. The determination matrix is shown in Table 2.1.

Table 2.1: ASIL determination matrix [35]. S1 (light and moderate), S2 (severe and life-threatening), S3 (life-threatening to fatal), E1 (very low), E2 (low), E3 (medium), E4 (high), C1 (easy), C2 (normal) , C3 (difficult), QM (Quality Management).

		Controllability		
Severity	**Exposure**	C1	C2	C3
S1	E1	QM	QM	QM
S1	E2	QM	QM	QM
S1	E3	QM	QM	ASIL A
S1	E4	QM	ASIL A	ASIL B
S2	E1	QM	QM	QM
S2	E2	QM	QM	ASIL A
S2	E3	QM	ASIL A	ASIL B
S2	E4	ASIL A	ASIL B	ASIL C
S3	E1	QM	QM	ASIL A
S3	E2	QM	ASIL A	ASIL B
S3	E3	ASIL A	ASIL B	ASIL C
S3	E4	ASIL B	ASIL C	ASIL D

The ISO-26262 describes in detail how to determine the ASIL of a system. Based on that assessment a functional safety concept is developed. With this proceeding safety requirements are derived and safety mechanisms, such as fault detection as well as redundancy concepts, are developed.

In order to illustrate the relation of the ASIL determination factors and the terms potential risk, tolerable risk, and residual risk, Figure 2.5 shows an example on how to interpret them.

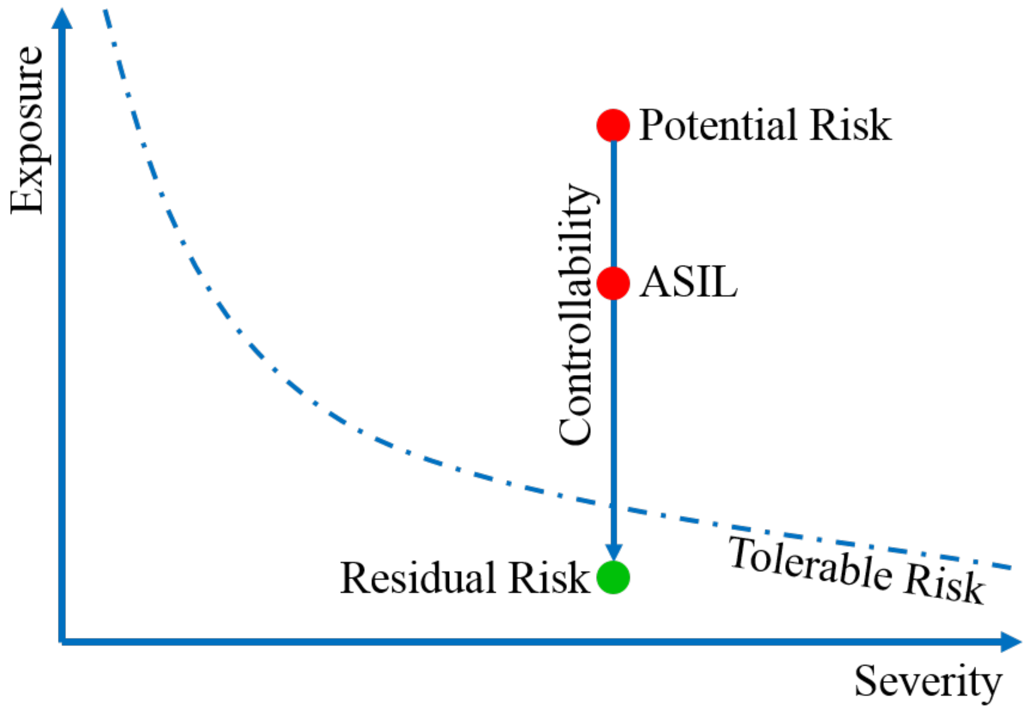

Figure 2.5: Illustration of the dependencies between ASIL determination factors and potential, tolerable, and residual risk [36, p. 90].

The potential risk of a system is defined by the product of probability of occurrence and severity. With an increase of these factors the potential risk results in higher values. The ASIL determination considers the controllability of a risk situation. In general, the potential risk can be decreased by controllability measures. Therefore, controllability mechanisms are elaborated to reduce the potential beneath the tolerable risk. The remaining risk, which can never be reduced to zero, is called residual risk [36].

After the development of the functional safety concept the product development including system, hard- and software level is conducted. On the system level the functional safety requirements are specified in detail, before the actual system design begins.

Based on those requirements the hardware and software development is initiated. The safety requirements are refined to the demands of the functional safety concept with respect to the hard- and software that is supposed to be developed. Subsequently, hard- and software tests are performed, where failures can lead to safety requirement violations. At last an installation plan is developed that ensures functional safety is guaranteed for production and operation of the vehicle [31].

It is worth to mention that there are a lot of supporting processes for the overall system development as listed in [31]:

- Interfaces within distributed developments
- Overall management of safety requirements
- Configuration management
- Change management
- Verification
- Documentation
- Qualification of software tools
- Qualification of software components
- Qualification of hardware components
- Proven in use argument

All of these aspects are mandatory to ensure functional safety for road vehicles according to the ISO-26262. For CAVs the ISO standard is not sufficient anymore. CAVs are clearly ranked as an ASIL D system, because the exposure to dangerous situations is high, the severity on public roads can result in fatalities, and the controllability depends on the CAVs' implemented driving functions. The only influence to reduce the potential risk of a CAV is to improve the controllability of the vehicle itself. Due to the complexity of implemented functions and their required collaboration it is challenging to derive comprehensive functional safety requirements to ensure functional safety. Another difficulty is the environment the CAV has to operate in. Often, these influences are not controllable at all. As already mentioned, the CAV is exposed to human driving behavior which can be hard to predict. Further, other influences, such as disturbances caused by weather conditions or map errors can cause behavior that is difficult to summarize in functional safety requirements. Additionally, it is difficult to state a tolerable risk for CAVs. At this time, there is no generally accepted state of the art on how safe CAVs have to operate. This depends also strongly on the public acceptance of CAVs in general. One thing is clear at this point, society will not accept CAVs that deteriorate traffic safety and cause more accidents than human drivers.

2.2 Challenges of Testing Cooperative and Automated Vehicles

The major challenge of testing CAVs lies in the complexity of the individual modules necessary to enable automated driving and their collaboration as well as the difficulties that are posed by the domain CAVs have to operate in. In order to understand why this is a more sophisticated task than testing of established ADAS, e.g., adaptive cruise control or lane keeping assist, it is required to understand the levels of automation, illustrated in Figure 2.6, introduced by the SAE [4].

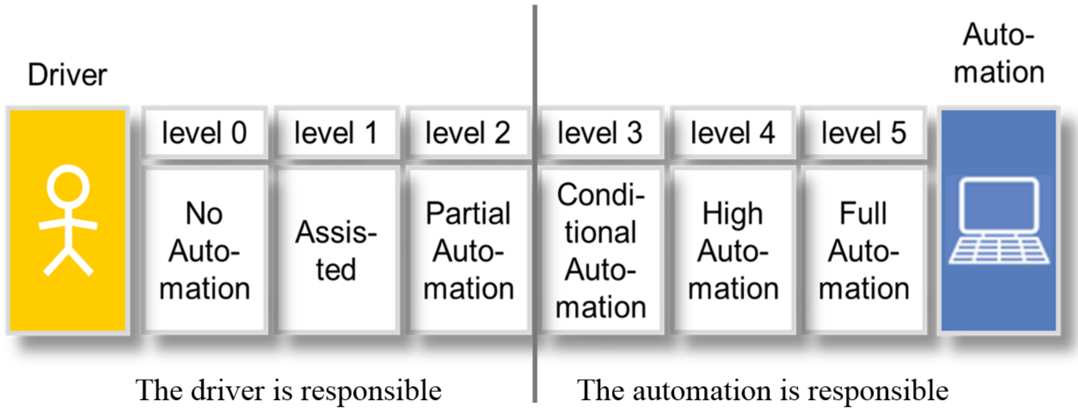

Figure 2.6: SAE levels of automation [23, p. 5], see also [4].

The important difference lies between level 2 and 3. Up to level 2 the responsibility to ensure safety remains by the driver, while for levels 3 to 5 the automation has to fulfill that demand. CAVs are clearly allocated to levels 3 to 5. The first generation of CAVs are level 3 systems, where the driver is kept in the loop and has to take control over the vehicle after a certain amount of time. The time span in which the driver has to take control is not agreed upon yet, but it is certain that the automation has to handle safety critical scenarios on its own. Especially, because these scenarios can occur unforeseeable, there is not always enough time to shift the vehicle control back to the driver. At the latest for level 4 and 5 systems the driver is kept out of the loop completely. The difference between level 4 and 5 is that level 4 systems are supposed to drive in a certain operating domain, such as highways or rural roads, while level 5 systems are supposed to be able to drive in every domain.

The shift of responsibility from driver to automation raises the need for new testing strategies as well as the question on how safe CAVs have to operate in comparison to human drivers.

One way to approach this question is proposed by Wachenfeld [37] by statistically determining the assurance for safety and use human drivers as a metric for comparison of CAV capabilities. In this consideration, accident data is used to determine the comparable abilities of human drivers. The average distance between two accidents can be considered as a measure for human driver capabilities. Now, the basic idea is to prove that a CAV performs as well as a human driver or preferably even better. Winner et al. [38, ch. 62.3.1] stated that the driving distance that has to be covered to statistically prove this, is 10-20 times longer than the average distance between two accidents. Considering that the distance between 2 accidents with injuries on German highways is around 12 million km [9, p. 23], it can be clearly stated that this proof is demanding. In fact, this proof is not feasible for CAV manufacturers. Another interesting aspect regarding this issue is that for every new developed CAV this proof has to be performed again. Further, considering that CAVs aim to increase the distance between accidents, the driving distance that has to be covered becomes even larger [2]. Note, since the opening of the Opel proving ground in 1966 until 2016, an overall distance of 200 million km has been accumulated on this test facility [39], only to get an impression on the required efforts.

Another attempt to tackle this challenge is the Safety Of The Intended Functionality (SOTIF) standard [40]. Although originally constructed for ADAS up to SAE level 2, the extension of the ISO-26262 standard is expected to become a potential guideline to ensure CAV safety as well. The SOTIF approach considers safety violations that are caused by limitations of the intended functionality. Key aspects, such as environment influences, object detections, and performance limits, are considered in the safety risk identification and evaluation. Based on that, measures to mitigate the risk are taken and a verification as well as a validation plan is elaborated [40]. However, this method also relies strongly on expert opinions, which can be considered as incomplete due to the complexity of the domain CAVs have to operate in. All expert knowledge based approaches face the obstacle, that the implemented algorithms, their collaboration, and the domain they are supposed to operate in are too complex to think of every possible safety violation.

Besides CAV safety, many non-functional requirements need to be considered in the development process. In fact, these requirements become more and more important, because the driving task is shifted to automation modules and therefore, aspects, such as traffic efficiency, travel time, driver comfort, and fuel efficiency, can be influenced directly. These non-functional requirements are significantly important for the acceptance of CAVs in general. The incorporation into current test methods is another challenge that needs to be considered. Requirements often correlate and can not be tested separately. For example, the travel time directly influences the fuel efficiency. To achieve a short travel time, high velocities and accelerations are necessary, which increases the fuel consumption [26]. Figure 2.7 shows an example of correlations between non-functional requirements.

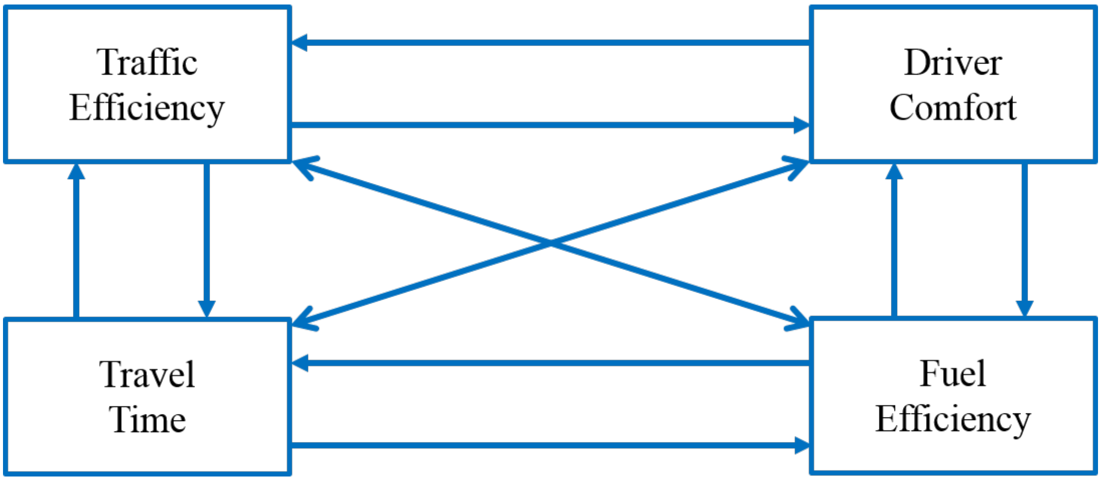

Figure 2.7: Correlations between non-functional requirements that need to be considered in the development process, based on [26].

As already indicated, if one non-functional requirement is considered, the influences on other non-functional requirements need to be considered as well. Starting with the requirement of a short travel time, the fuel consumption is going to rise and the driver comfort is directly influenced as well. Strong accelerations cause a lot of jerks that increase the forces affecting the driver. Such driving behavior of the CAV can be considered as uncomfortable for the driver. Additionally, a driving strategy minimizing travel time can cause safety related problems and decrease the traffic efficiency as well. For example, other drivers could be provoked or unsettled by such a driving strategy.

Another aspect is the fact that what increases the driver comfort of one CAV is not always the best strategy for other drivers involved in traffic. Considering the end of a traffic jam, where one lane is blocked and the best way to dissolve the situation for each involved traffic participant is alternate merging. The minimization of travel time for the CAV would be in contrast to that. Reducing travel time for the CAV results in a driving strategy that does not adopt alternate merging. The CAV suspends that concept and therefore, decreases the traffic efficiency, which will deteriorate the acceptance of CAVs significantly.

All in all, non-functional requirements can not always be fulfilled without influencing other non-functional requirements. Figure 2.8 shows this dependency. The optimal solution would be to improve the driver comfort and the traffic efficiency simultaneously, indicated in the first quadrant in Figure 2.8. As stated before, this is not always possible because of the correlation between those requirements. It is up to the manufacturer to decide which requirements are more important for the development process of the CAV. In any case, indicated through the shaded area, if one of the requirements falls below a certain threshold, the CAV's acceptance will downgrade significantly [7].

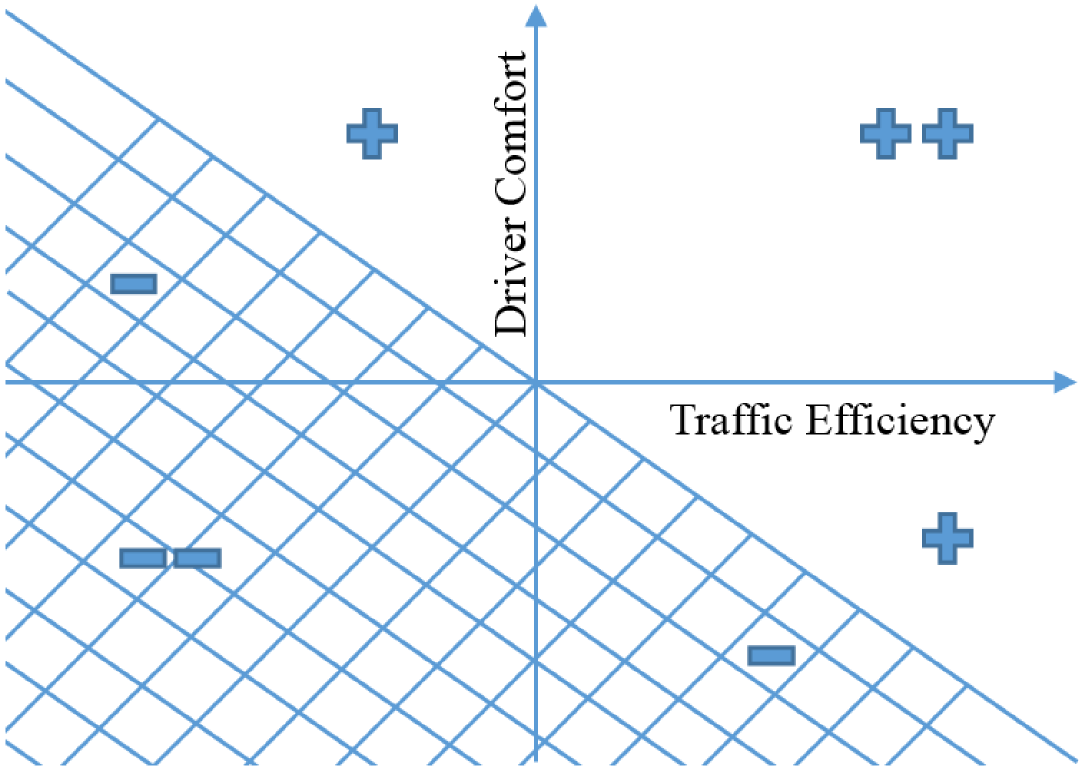

Figure 2.8: Non-functional requirements for CAVs. Optimal solution in the first quadrant. Critical region indicated by crossed area in quadrants 2-4 [7, p. 379].

In summary, all of these aspects highlight that conventional test methods (see Fig. 2.2) are not sufficient anymore. The prospects of simulation-based testing can provide helpful enhancements to approach this challenge. In general, simulations possess useful benefits, e.g., tests can be performed faster than real time, dangerous tests can be conducted, an omniscient view on all state space variables can be obtained, and reproducibility can be assured. Especially, the impact of CAVs on traffic efficiency and other non-functional requirements can be evaluated by simulation-based approaches. Simulation offers the possibility to observe large scale traffic systems without the need to set up traffic observation systems.

Thus, it can be stated that the effort of conventional methods to test CAVs is not feasible anymore. Especially, the complexity and variety of scenarios (see Fig. 2.4) as well as requirements that need to be tested and evaluated properly strongly exceeds the abilities of conventional test methods. The need for new test methods is obvious at this point and simulation-based approaches are a valuable asset to improve CAV testing in general, regardless if the tests are performed during development or validation.

2.3 Simulation-Based Development and Testing

Simulation tools are widely used to develop algorithms and test them in early development stages as well as in the validation phase. The tremendous advantages of the so-called Computer Aided Engineering (CAE) is utilized in almost every industry branch. Especially for control systems, vehicle development, and aerospace applications simulation used very frequently. The ability to test developed algorithms without the need of existing hardware and the possibility to test these systems without endangering users and components is one of the major advantages of simulation-based approaches in general. Thus, if a test under real circumstances is not possible, for example, when developing spacecrafts, simulation becomes a necessity. It is simply not possible to test spacecrafts under realistic conditions, because the components are too expensive and can not be deployed into space, before they are tested to a certain degree in simulation environments. Similar to that, CAVs should not be deployed to Operational Design Domains (ODDs) before comprehensive testing has been performed. The reasons are quite similar: CAV components are complex, the ODD poses safety related issues regarding traffic participants, and the testing in public traffic systems is too expensive as stated in Section 2.2.

First of all, it is important to understand how simulations operate and what the specific challenges are when utilizing CAE methods. Common practices use a so-called simulation pipeline to structure the simulation methodology [41, p. 13]. Figure 2.9 shows a simulation pipeline with the individual steps that need to be considered when applying simulation-based schemes.

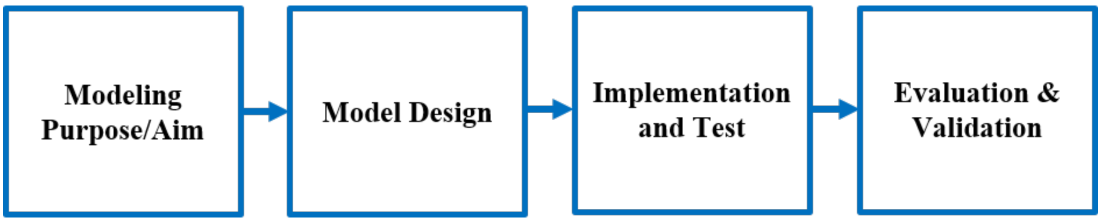

Figure 2.9: Pipeline to structure simulation-based approaches, based on [41, 42, 43].

In the beginning, the modeling purpose and aim need to be clarified. This means that the purpose of the simulation regarding the evaluation requirements is defined. For example, a vehicle dynamics simulation aims to describe certain aspects of the vehicle behavior under specific circumstances. In order to do that, the model that describes the vehicle behavior based on mathematical modeling needs to be chosen properly. For that matter, it is necessary to define the model properties and the view on the system. A system is always the part that is described and is usually only a part of the entire original system. This partial system is confined and described by mathematical models.

Based on the requirements for the simulation and the confined system a model type is chosen. There exists a vast variety of model types such as, differential equations, partial differential equations, stochastic equations, just to name a few. Additionally, discrete models, e.g., cellular automatas or hidden markov models, can be utilized.

In this dissertation, the focus lies in part on a vehicle dynamics simulation. Therefore, a brief introduction of a vehicle dynamics model based on differential equations is provided. This is part of the second aspect of the simulation pipeline called model design. The model purpose aims at characterizing the dynamical behavior of a CAV. Hence, the main goal for this model is to describe the vehicle behavior that is later on controlled by CAV driving functions. Figure 2.10 shows a mathematical model called non-linear single track model.

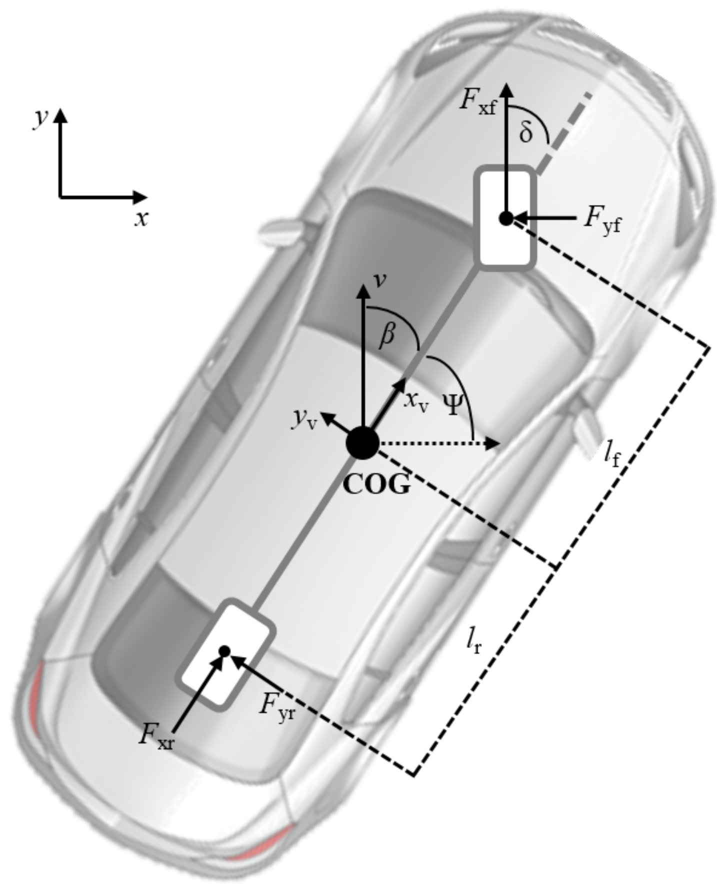

Figure 2.10: Non-linear single track model, see also [44, 45].

The model represents a simplified part of the original system. The vehicle motion model neglects the vertical vehicle dynamics and transforms the two front and rear tires in each case to one tire that are positioned along the longitudinal axis. The differential equations are now derived by force and momentum equations, which leads to a non-linear state space model, choosing the states to $\boldsymbol{x} = \begin{bmatrix} x & y & \Psi & \dot{\Psi} & \beta & v \end{bmatrix}^{\mathrm{T}}$ and inputs to $\boldsymbol{u} = \begin{bmatrix} \delta & F_{\mathrm{xf}} & F_{\mathrm{xr}} \end{bmatrix}^{\mathrm{T}}$, which can be expressed by [44, 45]:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} v\cos(\Psi + \beta) \\ v\sin(\Psi + \beta) \\ \dot{\Psi} \\ \frac{-F_{\mathrm{yr}} l_{\mathrm{r}} + F_{\mathrm{yf}} l_{\mathrm{f}} \cos(\delta) + F_{\mathrm{xf}} l_{\mathrm{f}} \sin(\delta)}{I_{\mathrm{z}}} \\ -\dot{\Psi} + \frac{F_{\mathrm{yf}} \cos(\delta - \beta) + F_{\mathrm{xf}} \sin(\delta - \beta) + F_{\mathrm{yr}} \cos(\beta)}{mv} \\ \frac{-F_{\mathrm{yf}} \sin(\delta - \beta) + F_{\mathrm{xf}} \cos(\delta - \beta) + F_{\mathrm{yr}} \sin(\beta)}{m} \end{bmatrix} . \tag{2.1}$$

The states x and y are the vehicle positions with respect to the reference coordinate frame, Ψ the yaw angle, $\dot{\Psi}$ the yaw rate, β the slip angle, and v is the velocity vector. The input vector consists of the steering wheel angle δ, the longitudinal tire forces F_{xf} and F_{xr}, the lateral tire forces F_{yf} and F_{yr} in the vehicle's front and rear, respectively. With respect to the Center Of Gravity (COG), the lengths l_{f} and l_{r} describe the distances to the vehicle's front and rear. At last, m describes the mass of the vehicle and I_{z} the moment of inertia of the yaw movement [45].

This mathematical model is usually solved numerically by simulation tools. The parameters can be obtained by examining data, measurements, and experiments. The integrators for the simulation model are usually solved by a numerical solver, e.g., Euler or the Runge-Kutta method [46], just to name a few. Important considerations need to be made regarding numerical step sizes, precision, singularities, input and output value ranges, etc. For further information the reader is kindly referred to [11], [47], and [48].

One of the major issues regarding simulation-based techniques, which causes a lot of dispute, are the uncertainties between the mathematical model and the real system. Thus, the model quality is important to determine the reliability of the performed simulations and the obtained results. Predictions about model reliability are usually determined by statistical experiments and robustness analysis. The applied methods rely on the specific models and their individual properties. The decision by which method these aspects can be determined depends on the beforehand stated requirements as well as the resulting mathematical model, which is obtained by the afore mentioned steps. Based on the validation of a model by assessing the model quality, the evaluation of the simulation results is performed. Note, the mathematical model does not have to be as valid as possible in general, rather the model needs to suit the purpose for what it is designed in the first place.

Simulation techniques require the user to be aware of model uncertainties and imperfections. Modeling of dynamical systems can not always be achieved as mentioned before. In some cases, the system that is described is too complex to be modeled in this way or it is easier to model aspects by using another point of view. Especially, if there are various models that interact with each other. It is rather difficult to describe a variety of interacting models in a single differential equation. This difficulty can be overcome by using multi-agent simulations [49], which model the individual agents and the interactions with each other. This means, each agent possesses a model describing its individual behavior. Then, these individual models that appear multiple times are equipped with rules and equations on how to behave and interact with each other. For example, these techniques are suited for large traffic systems, risk assessment of maritime operations, and applications in medicine.

In this dissertation, the focus lies on CAVs operating in their ODD, which is a public traffic system. In order to perform simulations of traffic systems multi-agent-system approaches are eligible. As already indicated, these systems possess agents that include an individual parameterizable model that describes the behavior of each agent in a spatially distributed environment. In the case of a traffic system, the spatial domain is modeled as a road system. The agents travel on that road system according to their individual models. Agents can be cars, buses, pedestrians, etc. The road system infrastructure consists of lanes, intersections, speed limits, traffic lights, etc. A comprehensive list can be found in [50].

To model the behavior of the individual agents the traffic simulation tool Simulation Of Urban Mobility (SUMO) [51] is an option. The target of the agents is to behave as a human driver and therefore, so-called driver behavior models are implemented in each agent of the simulation tool. Among many other existing models, the two important ones, car following and lane changing, are briefly explained. Figure 2.11 shows two agents on a highway, where one agent follows another.

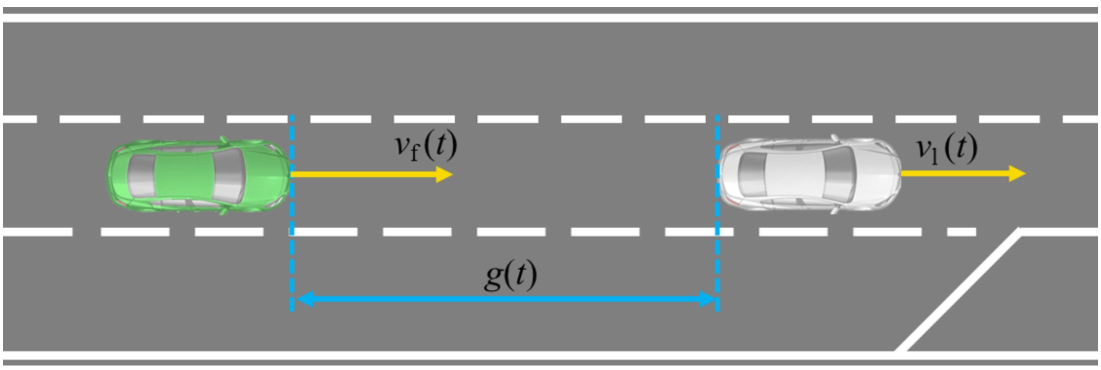

Figure 2.11: Car following situation of a multi-agent traffic simulation [52]. Agent 1: following vehicle colored in green. Agent 2: leading vehicle colored in gray.

Agents are configured with a specific routing before a simulation run is initiated. The agents' behavior during the simulation run is determined by the driver behavior models and their parametrization. Basically, the models determine the exact behavior of an agent, which is unknown in the beginning of the simulation run, because the behavior depends on other agents and the interaction between each other. The car following model in SUMO can be considered defensive and accident free. There are three velocity modes that are described by

$$v_{\mathrm{des}} = \min\{v_{\mathrm{safe}}, v(t-1) + aT, v_{\mathrm{max}}\} \tag{2.2}$$

and calculated simultaneously [52, p. 2]. The minimum value of each calculation is chosen to be the desired velocity. Either the desired value is determined by the so-called safe velocity relationship [52, p. 2]

$$v_{\mathrm{safe}} = v_{\mathrm{l}}(t) + \frac{g(t) - v_{\mathrm{l}}(t)\tau}{\frac{\overline{v}}{b(\overline{v})} + \tau} \quad , \tag{2.3}$$

or the agent accelerates and decelerates as parameterized, or the agent drives with the maximum velocity that is allowed on the particular lane. The safe velocity depends on the velocity of the leading v_{l} and following vehicle v_{f}, respectively, the gap $g(t)$ between the vehicles, a braking function $b(\overline{v})$, and the so-called reaction time of the driver expressed by τ.

The second aspect to be considered is that the lane changing model possesses four different modes: strategic, cooperative, tactical, and regulatory. First of all, there are some rules that prevent a lane change, for example, the gap between a leading or a following agent on the desired lane is too small. Lane changes are executed in a discrete way, which means that by default, the agents jump from one lane to another. There are possibilities to change the lane change duration to achieve a continuous lane change behavior. The decision whether a lane change should be performed depends on the chosen lane change model. Strategic lane changes occur due to the following set of rules: a lane is ending, the occupation on the current lane is higher than on an adjacent lane, the offset to the next desired lane according to the route should be small. Cooperative lane changes occur when a desired lane change is blocked by another agent. If there is no current reason for the blocking, one agent has to free the desired lane for the other agent and a cooperative lane change is performed. Tactical lane changing is applied when the desired velocity can not be reached because of a blocking agent driving slowly on the same lane. In this case, the agent changes the lane to overtake the other agent. Note, traffic rules, such as no overtaking on the right side can be applied.

At last, regulatory lane changing enables to enforce that the agents should drive on the right side, if they are not overtaking [53, ch. 4-7]. A comprehensive overview about car following models, lane change models, and adjustable parameters of the driver behavior models provided by SUMO can be found at [54].

2.4 Research Objectives and Related Work

In this section, the research objectives for this dissertation, based on previous sections 1.2, 2.1, 2.2, and 2.3, are shown. A detailed description of related work as well as partial overlaps of the methodologies and required research contributions are identified. First of all, the research objectives are clarified and can be stated as:

- Establish a systematic method for improving simulation-based CAV development and testing,
- Enable the assessment and identification of critical scenarios systematically and automatically,
- Evaluate the effects of CAV driving functions on ODDs and vice versa for different DOIs simultaneously,
- Feature the investigation of disturbance influences on CAVs and ODDs.

These are the research objectives that motivated this dissertation. Based on these objectives requirements are derived more precisely and compared to existing methods and approaches. It is worth to mention that the fulfillment of these requirements is achieved by the methodology introduced in Section 3. Of course, other approaches fulfill different research objectives, which is caused by the general scope of those methodologies. Although all of these approaches are focusing on CAV or AV testing, the specific objectives differ quite strongly. On the other hand, the methodologies possess some overlapping aspects as well. The aim in this section does not lie on an evaluation of the different methodologies, but rather focusing on the research fields that possess gaps and how this dissertation contributes to establishing a novel methodology as well as the enhancement and improvement of existing approaches. In many ways, the methodologies that are compared can co exist and complement each other.

The general research objectives need to be specified and transferred to requirements. Therefore, the objectives are represented in more detail. In this way, the resulting requirements can be compared to other methodologies. The comparison is applied by focusing on state-of-the-art research projects that are aiming for the improvement of CAV or AV testing as well. Later on, supplementary related work is presented mainly containing partial auxiliary solutions from individual industry and academic players.

The main contributions are delivered by current research projects in which industrial and academic partners work together to approach this challenge. For that reason, three research projects, which play a major role for CAV testing are chosen: PEGASUS [55], Ko-HAF [56], and Enable-S3 [57]. While Ko-HAF and Enable-S3 possess additional objectives besides CAV testing, the PEGASUS project's focus lies solely on AV testing and validation. Ko-HAF and Enable S3 are focusing on CAV testing in sub-projects. PEGASUS, while being very comprehensive on AV testing and validation, does not focus on cooperative features. The specified requirements regarding the methodology developed in this dissertation are compared with those developed by the afore mentioned research projects. Table 2.2 shows the requirements in the first column and the comparison result in the following columns. Note, that the requirements stated in this table are derived from the stated research objectives of this dissertation.

Table 2.2: Comparison of requirements (derived from research objectives) regarding methodology and comparison with respect to state-of-the-art research projects. "+" fulfilled, "-" gap, "n.a." not available.

Requirements	**PEGASUS**	**Ko-HAF**	**Enable-S3**
Systematic simulation-based methodology for development accompanying tests of automated driving functions	-	-	+
Consideration of cooperative features	-	+	+
Aspects of the methodology are generic and exchangeable	+	+	+
Simulation tool independent	+	+	+
Usability across different test platforms	+	-	+
Exchangeability of simulation models	+	n.a.	+
Exchangeability of CAV driving functions	-	-	+
Features all SAE automation levels	-	-	+
Compatible with state-of-the-art scenario modeling	+	-	-
Automated identification of critical scenarios	+	-	-
Assessment of critical scenarios	+	+	+
Continuous evaluation, improvement, and documentation of maturity level for CAVs	-	-	-
Simultaneous evaluation of CAV effects on ODDs and vice versa, considering different DOIs	-	-	-
Features the investigation of disturbance influences on CAVs and ODDs	-	-	+

Table 2.2 lists the specified requirements as well as the fulfillment and gaps identified by analyzing the selected research projects. It is worth to mention that the research projects fulfill other requirements that are not within the scope of the methodology developed in this dissertation. In the following, the projects are briefly introduced and the aims and resulting related work is examined. The subject matter and comparison to the stated requirements are supported by expert reviews with researchers participating in the projects.

PEGASUS

The PEGASUS project was funded by the German government and focuses on efficient securing of automated driving. The main contributions of this initiative can be divided in four sub-projects: scenario analyses and quality measures, implementation processes, testing, and result reflection and embedding [58, 59]. The first sub-project deals with defining design criterion, deriving of functional requirements [60], and the important question: "how good is good enough [61]?" AV driving functions are compared to human driver capabilities in critical scenarios, aiming for a determination of a benchmark required for AV driving function quality [62]. Further, the second sub-project focuses on designing development and test processes and the embedding into already established methods [10]. The third sub-project focuses on testing AVs and possesses the most overlapping aspects regarding this dissertation. Especially, the PEGASUS database [63] is developed in this sub-project as well as test strategies, ensuring that all safety relevant scenarios within the scope of the AV driving function are tested, determination of functional limits, and the verification and validation of test methods and results. Additionally, a toolchain for different test domains, such as simulation, proving ground, and field tests, is established [64]. Finally, sub-project four ensures that the developed processes and results are reflected and can be used in companies [65].

The overall PEGASUS methodology starts with gathering data from various sources, e.g., regulations, standards, guidelines, driving tests, simulation, recorded data, and accident studies. This data is used to derive technical requirements, identify scenarios, and store and evaluate them in the PEGASUS database. Further, the scenarios are assigned to scenarios containing parameter spaces. After the determination of exposure for the individual scenarios a resulting space of test cases is obtained. Based on this, specific test cases are derived by stochastic variation through the space of test cases. The test cases are divided and assigned to simulations, proving grounds, and driving tests. The obtained test data is assessed, categorized, and used for test result evaluation. At the end, a risk assessment is carried out and a safety argumentation is established [58].

The database of relevant scenarios is one relevant result of the PEGASUS project. The database approach gathers relevant scenarios from different domains, such as dynamic driving simulators, field operational tests, and real world traffic.

Based on generic criteria the scenarios are classified. If the scenario is relevant, it is stored in the database and can be extracted for other parts of the V-model for testing purposes.

The approach enables the CAV manufacturer to transfer gathered data into the database. A methodology called data processing chain provides a data transformation and a generation of deduced signals, where the data format is checked and the raw data is enriched as well as deduced signals are calculated. Enriched and deduced signals can be stated as quantities that are not directly obtained from the raw data, for example, the derived evaluation metric Time To Collision (TTC) [25]. The following step contains the calculation of scenario likelihoods, which is necessary to cluster the scenarios to predefined parameter space-based scenario models. Further, the scenarios are equipped with events, where the start as well as the end time of a scenario are determined. The scenarios are clustered with respect to the predefined scenario models and enhanced with a probability distribution of the parameters. The last step contains the selection of scenarios for testing and additional information on exposure, severity, and controllability. Finally, the test specifications can be derived based on the previous step [63].

A test concept was proposed to assign tests from the database to different test domains. The collaborators propose to deviate test cases automatically by stochastic variations of the parameters. The proving ground tests are manually chosen either from the database or based on the varied simulation data. Field tests are used to find unknown scenarios, data for the database, and scenarios for a replay to simulation approach [66, p. 22]. The results are leading to an evaluation of the test cases based on a pass/fail criterion, leading to a systematic test concept for CAVs [67].

Further, details for the different test domains are elaborated on, containing sensor simulation models [68], SiL [69] approaches, HiL simulations [70], and how to perform tests on a proving ground by implementing a control center [71].

Another aspect that is covered in PEGASUS is the identification of risks that occur because of the vehicle automation. The potential to increase safety is a major asset of CAVs. In order to stay accurate, there are risks and safety issues that are induced by CAVs. There remains a certain impact either from traffic participants on the CAV or vice versa. Especially, insufficiencies of the driving functions, limitations of the CAV's perception as well as misinterpretation of behavioral prediction of other traffic participants can cause unsafe situations. PEGASUS aims to address and identify so-called automation risks [3, 72].

A structured methodology on how to embed the PEGASUS approaches and concepts into the vehicle development process is also in the scope of the research project. Further, the enhancement of the classical V-model is part of research. The PEGASUS methodology as well as SOTIF need to be included in the V-model [10].

Ko-HAF

The Ko-HAF project is funded by the German government as well and possesses a sub-project where the center of interest lies on validation and test of CAVs. In comparison to PEGASUS, Ko-HAF features only one sub-project paying attention to resolve this issue. Main aspects of the sub-project are development of a test methodology, a test procedure as well as a test specification and the assessment of real world data [5].

The Ko-HAF approach for validation and test starts by deriving use cases based on the implemented CAV driving functions. These use cases are assembled in a so-called scenario catalog. In accordance with the Ko-HAF scenario modeling, use cases are an abstract description, e.g., enter the highway, overtaking, leave the highway, etc. The consortium focuses on a SAE level 3 CAV equipped with driving functions for German highways. The use cases are divided into basic scenarios and extended by generic parameters. After the specification of the generic parameters test cases are derived. Note, the test cases possess another input called test specifications. The test specification includes aspects, such as initial values, test participant trajectories as well as the test environment. The methodology concludes with a test execution and evaluation [5].

The AV cooperation in Ko-HAF includes a safety server, where the individual CAVs are enabled to notify other CAVs on road closures, broken-down vehicles, and other hazardous events. Notifying other CAVs within a fleet about dangerous situations beforehand aims at increasing safety. The cooperation features in Ko-HAF do not contain collaborative trajectory planning or merging based on direct Vehicle-To-Vehicle (V2V) communication [73].

One of the main contributions of the sub-project is the development of test tools for the advisory of test drivers on a proving ground. Each test participant is equipped with a localization system as well as a display and therefore, guided on how to drive a certain scenario that is derived by the afore mentioned methodology [74, 75].

A simulation approach for software testing is proposed, where the developed software can be tested and evaluated. This approach aims at an agile development and early feedback for the feature developers. Further, the methodology considers different architecture levels starting on the source code level, progressing to software and system component levels up to the system level, aiming at continuous testing of CAV software [76]. Additionally, effort is spent on generic validation schemes for the validation of vehicle environment sensor models [77].

The overall methodology for test and validation was published by the TU Brunswick's chair of automotive engineering [78, 79] as well as on the Ko-HAF final event on the Opel proving ground in Dudenhofen near Frankfurt. Especially, the efforts spent on this methodology, test tools, and CAV testing were showcased by driving demonstrations, presentations, and posters.

Enable-S3

The European initiative Enable-S3's focal interest lies on enabling the validation of highly automated safe and secure systems [57]. In contrast to PEGASUS and Ko-HAF the industrial domains are manifold. Enable-S3 examines, besides automotive, industry sectors, such as aerospace, rail, maritime, health, and farming [80]. The main contribution of the project aims for developing a methodology combining simulation-based approaches with real-world tests for highly automated safe and secure systems.

The consortium inaugurated the system-under-test and test requirements for each covered domain [81]. Further, the test requirements and domains are specified and prioritized. The evaluation report delivers the objectives of Enable-S3 to provide a test and validation framework, reducing test effort, novel testing with physical sensor signal stimuli generators, reducing malfunctions, reuse of validation scenarios, establish standards, functionality across different domains, and the consideration of a verification and validation ecosystem for the European union's industry [82]. The afore mentioned objectives are evaluated by comprehensive quality measurements and metrics [83]. A generic test methodology is proposed containing the main aspects: Verification And Validation (V&V) management, test management, and test platform. The V&V management includes all reusable aspects that are valid during different test phases. The test management takes care of the generation and execution of test cases as well as the evaluation regarding performance and correctness of tests and a qualification for release and residual risks. Finally, the test platform is elaborated covering all relevant aspects. These aspects are abstract and generic not specifying a test environment, e.g., MiL, SiL, HiL, etc. The methodology is set up to be compatible with standard test environments and can be instantiated using different abstraction levels [84].

The methodology features a scenario-based approach for verifying scenarios including a criticality analysis. Furthermore, scenario reconstruction from recorded real-world data, safety and security analysis, an abstract scenario design, representative scenario checking, and labeling of recorded data. In particular, virtual testing and test benches are elaborated on followed by a test plan specification [85].

Additionally, a test platform specification is developed within the research project. The chosen use cases from all domains are used for fitting test platforms and the applied simulation models, simulation tools, and use case relevant technical requirements are stated. Broadly speaking, the Enable-S3 methodology is applied and the test platform is specified for use cases containing the different domains as mentioned before [86].

Within Enable-S3's publications a co-simulation framework for the supporting verification and validation of ADAS, such as adaptive cruise control, lane keep assist, and traffic jam chauffeur, are developed and applied. Certain scenarios, for example, platooning, cut-ins, target lane change, are implemented and tested [87].

Zofka et al. [88] proposed a mixed reality framework for training and evaluation of autonomous vehicles. The focus lies on testing AVs interacting with vulnerable road users in shared spaces on parking facilities. A supplementary paper proposes to fill the gap between MiL and driving tests by augmenting the sensor perception of a real AV prototype with static and dynamic augmented virtual objects [89].

2.5 Complementary Work

Alongside the afore mentioned state-of-the-art research projects an overview of supplementary related work and contributions on this topic by academia and industrial research is provided.

As already indicated, besides contributing methodologies and implementation of the PEGASUS database, the research Institute For Automotive Engineering (IKA) of the RWTH Aachen provides a traffic trajectories dataset for German highways recorded by drones [90, 91]. This approach is suitable to investigate traffic participant behavior and to identify critical scenarios based on real traffic data.

The chair of traffic engineering and control located at the TU Munich published studies concerning the impact of CAVs on traffic capacities. Foremost, the CAV driving functions are modeled following assumptions of typical lane change and car following behavior. The studies are conducted and evaluated by using microscopic traffic simulation tools [92, 93]. Moreover, the chair proposed to combine the advantages of vehicle dynamics and traffic simulations for ADAS evaluation in general [94].

Research associates of the TU Darmstadt's chair of automotive engineering proposed a method to virtually assess AVs by implementing driving functions in a manually driven vehicle, using the sensor perception and simulating the AV behavior based on the currently gained information, called Virtual Assessment Of Automation In Field Operation (VAAFO) [95, 96].

Schuldt et al. introduced a test case generation methodology for ADAS including the identification of factors that influence the system significantly. The test case generation is carried out in a virtual environment, where the scenario generation is conducted with a 4-layer model to separate different elements of a scenario [97].

Researchers of the Johannes Kepler University from the institute for design and control of mechatronical systems located in Linz (Austria) proposed a virtual development and evaluation framework for ADAS. The idea of simultaneously using different simulation tools, such as vehicle dynamics and microscopic traffic simulation, for the virtual development and evaluation of ADAS systems is proposed. A case study for determining the performance of ADAS functions including cooperative features was carried out [98].

Among other researchers of the Karlsruhe Institute Of Technology (KIT), the research center for information technology published several papers on the identification as well as assessment and evaluation methods for driving scenarios through virtual test drives and test methods for automotive systems in general. The research tackles the rising complexity of soft- and hardware as well as an identification and mapping of driving scenarios using simulation-based tests and machine learning-based clustering algorithms [14, 99, 100].

A distributed simulation platform for cooperative ADAS development, testing, and validation is elaborated on in the paper from Jemaa et al. and aims at using co-simulation concepts for testing cooperative ADAS features. The platform consists of a data processing management, simulation for sensors and environment as well as a simulation for communication [101].

Researchers from the Carnegie Mellon University also persevere on the challenges of ensuring AV safety. Koopman et al. claim that a vast amount of testing is necessary for this formidable challenge [102]. The suggestions to tackle this issue are to separate requirement validation from design validation, use high-fidelity simulations to reduce residual risks, ensure observability to assure test evaluation, and enhance the safety argument by uncertainties [103]. Koopman elaborates on safety argument strategies, potential pitfalls, formal proofs of correctness, and further safety argument observations [104]. Additionally, a comprehensive overview of simulation-based testing and the limitations of these approaches is given in [105].

Leading companies in the field of CAV development are publishing safety reports. Waymo provides insights on system safety programs, testing and validation methods, and safely interacting with the public. Further, details on software and hardware testing as well as testing fully integrated self-driving vehicles are disclosed [66].

General Motors and their AV development company Cruise Automation located in the Silicon Valley also published a safety report for AVs elaborating on deployment, system diversity and redundancy, and how to design a safe AV. A first hand comprehensive overview can be found on elements of safety covering aspects, such as ODD, object and event detection and response, minimal risk conditions, validation methods, data recording, just to name a few [106].

The international research and development center Virtual Vehicle located in Graz works on advanced virtualization methods considering automated driving, safety & security, efficiency & comfort, digital operation, efficient development, and advanced testing [107].

The German Aerospace Center announced to launch a research initiative to transform the city of Brunswick into a laboratory for CAV development and testing as well as intelligent mobility called AIM. The focus lies on traffic flow optimization, safety, intermodal mobility, and future mobility concepts.

The initiative aims at enhancing the existing test infrastructure that consists of simulation frameworks, test benches, proving grounds, among others [108]. Currently, this research infrastructure is extended by the Test Bed Lower Saxony initiative. One major outcome will be ground-truth data for many modeling and V&V procedures. In contrast to AIM the Test Bed Lower Saxony focuses on highways [109].

Additionally, another test field with a similar focus is established in Karlsruhe called Test Area Autonomous Driving Baden-Württemberg. The initiative aims at providing an infrastructure for companies to develop and test CAVs under real conditions. ODDs include highways, rural roads, and urban areas. The concept consists of, similar to AIM, equipping the city's infrastructure with sensors for CAV testing, providing precision maps, and featuring the combination with test infrastructure systems, such as test benches [110].

2.6 Conclusions on the State of the Art

The afore mentioned related work includes a vast variety of methodologies and approaches for CAV testing in general. This dissertation aims at developing a systematic simulation-based toolchain for development accompanying tests for CAVs. The toolchain covers all SAE automation levels including cooperative features. Another major asset is the generic and exchangeable composition that allows for the use of different simulation models and tools, CAV driving functions, evaluation methods, and the combination with other test platforms, such as SiL, HiL, ViL, mixed reality, test benches, and driving tests. Additionally, the approach enables to automatically assess and identify critical scenarios in conformance with state-of-the-art scenario modeling. Especially, the identification of yet unknown critical scenarios poses a major challenge that requires a systematic methodology. Basically, the scenario catalogs developed by requirements engineering and expert opinions are incomplete and require enhancement with additional critical scenarios that are unknown at the point in time of the current development stage.

Another gap that arises is the need for a continuous evaluation, improvement, and documentation of the maturity level of the CAV during development. While a CAV driving function is evolving, developers require a methodology that provides information on the current maturity level as well as a continuous evaluation and documentation through a systematic proceeding. The evaluation needs to be adaptable to the requirements of the CAV driving function developers. Further, the interdependency of the CAV and its ODD demands for a systematic evaluation procedure, simultaneously covering the impact of CAVs on the ODD and vice versa. Specifically, the domains of interactions between CAVs and ODDs can be considered as versatile. Disturbances that occur either in the CAV driving function or through the ODD are supposed to be covered by the toolchain.

The toolchain is usable throughout the entire development process for different CAV prototypes as well as mitigates manual human intervention and operates automatically during the majority of the approach. Further, the compatibility with other existing methods, as mentioned before, can be considered as an important asset of the toolchain. Foremost, the toolchain is applicable for CAV developers and their individual objectives independent of the development stage.

Finally, the assessed and identified scenarios can be used for validation purposes as well. Therefore, the toolchain delivers input, for example, to the PEGASUS database or scenario catalogs for the validation and release of CAVs. The toolchain allows for enhancing current validation schemes by augmenting the knowledge base of current state-of-the-art approaches, such as introduced in PEGASUS, Ko-HAF, and Enable-S3.

3 Methodology

This chapter contains the generic methodology for the verification and identification of critical scenarios. It is worth to mention that this approach aims for the improvement of testing CAV driving functions in general. In order to solve this challenge systematically, a simulation-based toolchain is proposed. First of all, an overview on the capabilities and the prospects of the toolchain is given. Thereby, every step of the toolchain is explained in detail. Further, advantages, disadvantages and aspects that have to be considered when applying this toolchain are elaborated.

The chapter holds the structure to perform a walk through the entire methodology. So, the toolchain can be considered as a guide for reading this chapter. Every aspect is dedicated in a separate section containing the specific content for this step. Briefly speaking, the chapter starts by introducing the simulation-based toolchain. After that, the scenario modeling is explained based on which the simulation runs are derived. The derivation is either performed by conventional methods, such as expert opinions or automatically by parameter variations. The simulation environment is set up as a co-simulation scheme combining the advantages of different simulation tools. In advance, the toolchain possesses a vehicle dynamics, traffic and cooperation simulation covering the main parts required for comprehensive CAV testing. The automated evaluation of the simulation runs is performed by metrics classifying the criticality of a scenario. Subsequently, the data processing is elaborated on following an aspect to increase the validity of the results. Finally, an analyses step is conducted providing test results to improve the CAV's driving function.

After walking through the toolchain a novel approach called Prototype-in-the-Loop is presented. This methodology aims for the combination of a real-world CAV prototype with virtual test participants. In principal, the vehicle dynamics simulation is substituted by a real-world CAV prototype driving on a proving ground, while other virtual test participants are controlled by the traffic simulation. This approach enables to achieve reproducible results while providing an omniscient view on the test execution and due to the fact that the test participants are virtual, even dangerous tests can be performed.

This chapter concludes with an in-depth discussion on the contributions of this methodology. Furthermore, a review about the fulfilled research objectives and the derived requirements for this dissertation is provided.

3.1 Simulation-Based Toolchain

In order to verify and identify critical scenarios systematically, a generic simulation-based toolchain is introduced. The toolchain shown in Figure 3.1 provides a structured guideline to approach this challenge.

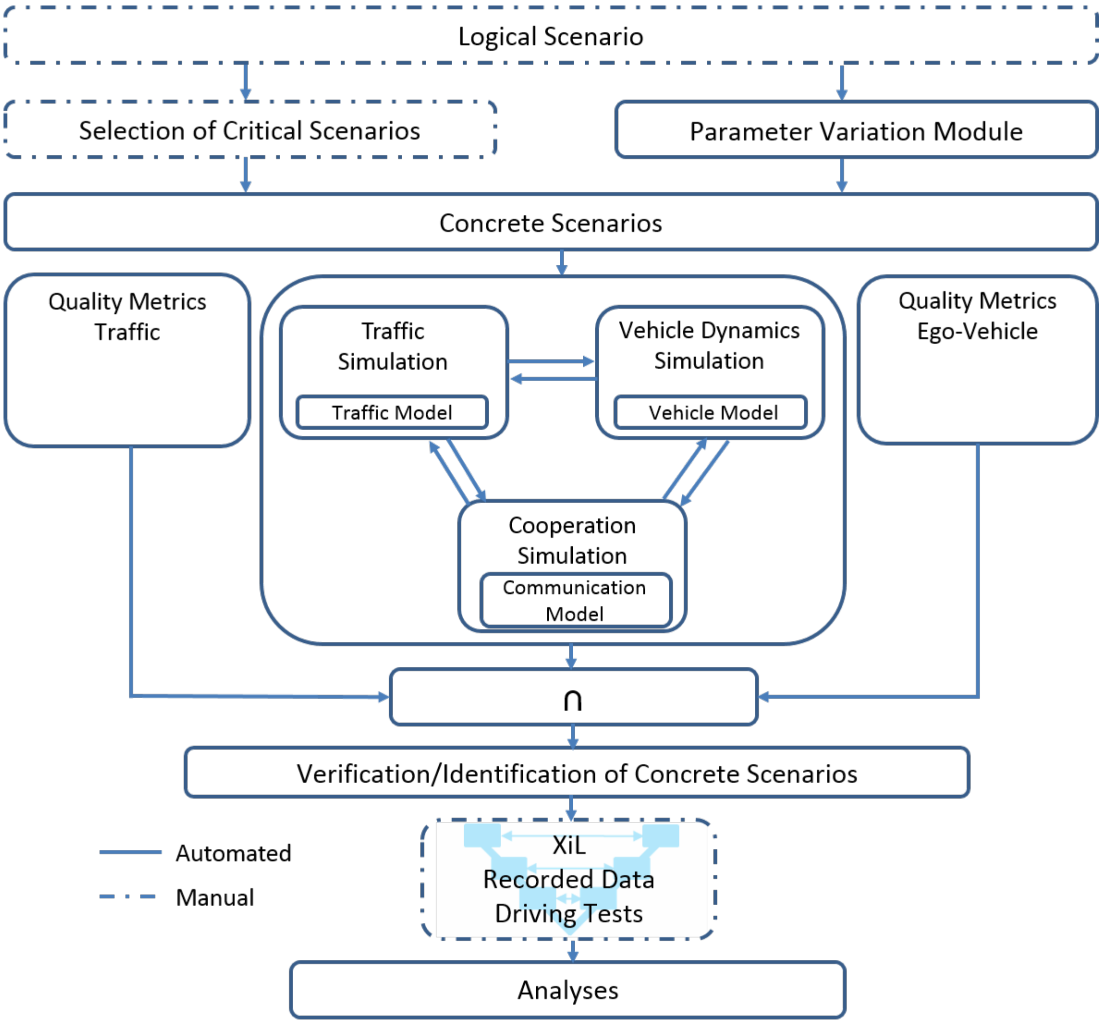

Figure 3.1: Simulation-based toolchain for the verification and identification of critical scenarios for CAVs [8, p. 95].

The methodology aims to verify and identify critical scenarios that need to be tested for improving cooperative and automated driving functions. This is an essential challenge that has to be taken care of before the CAV's validation process begins. As described in Chapter 2, the validation of these vehicle functions is challenging and requires a lot of effort. In a way, the CAVs that are used for validation should be in a mature state.

This means that the functions have to be tested intensively in the development phase as well. Additionally, the influences of disturbances and the impact on traffic systems should be considered in early development stages. Occurring disturbances can be manifold due to the complexity of these systems, e.g., sensors and actuators, the implemented functions, the interaction of the individual modules are error-prone. Fortunately, the toolchain is able to inject these influences and in case of the MiL approach without endangering test drivers or prototypes. In the special case of CAVs, disturbances can be caused due to influences of the world they are supposed to operate in.

Here, human driving behavior is still one of the most challenging problems. Anxious drivers are difficult to predict, while aggressive drivers pose a huge problem for the safe operation of CAVs. Considering that the new generation of CAVs are equipped with precision map data for localization, driving strategy decisions, and the support of motion planning, map errors can occur and cause significant issues. Especially, if different manufacturers use their own maps, these maps differ in accuracy, level of detail, up-to-dateness, up to errors that have been made during the map data acquisition.

Depending on the scope of the implemented functions a defined parameter space can be derived. This parameter space can be described in a certain scenario description explained in Section 3.1.1. Based on these parameter spaces it is possible to select critical scenarios determined by scenario catalogs, expert opinions, and examination of data. This is covered by the left-hand side of the toolchain and provides the possibility to verify critical scenarios. On one hand it is useful to be able to test and improve the CAV behavior for known critical scenarios. On the other hand, early tests provide useful information for actual driving tests performed later on. Furthermore, the scenario catalog can be extended and expert opinions can be evaluated before a CAV driving test has to be performed. In some cases, scenarios based on expert opinions are not critical at all and can be neglected.

On the right-hand side the focus lies on the identification process. A parameter variation module gives the contingency to vary parameters with a certain step size with respect to their parameter spaces. Proceeding like this aims for the identification of yet unknown critical scenarios. This is a major advantage of this method, because many critical scenarios are unknown and not accounted for in the scenario catalog itself. Even though requirements engineering has been established and successfully applied for a long time, the complexity of CAV functions and the variety of operation conditions resulting from the domains the vehicles have to operate in seems to push established methods to their limits. Therefore, it is inevitable to find new solutions for identifying critical scenarios, preferably supported by automated methods, where human intervention is kept at a necessary minimum. It is worth mentioning, that most parts of the toolchain can be operated automatically. Only the parameter spaces, the selection of critical scenarios on the left-hand side and the validation step need to be carried out by experts.

This differentiation between fully automated aspects and aspects, where manual human intervention is needed, is illustrated in Figure 3.1 by solid and dash-dotted lines, respectively. The resulting scenarios possess a defined description to start simulation runs performed by a coupled simulation environment. For a simulation run it is required that the parameters are preliminary defined. Either the trajectories of each test participant can be specified completely or test participants with driver behavior models can be generated. The CAV's trajectory is unknown at this point, because it is controlled by driving functions.

Running simulations is enabled by a coupled simulation environment that consists of a vehicle dynamic, traffic, and cooperation simulation. The goal of operating different simulation tools simultaneously is to use the advantages of each individual tool and combine them into a comprehensive simulation environment. The vehicle dynamics simulation contains a digital prototype of the CAV (see Fig. 1.1). The surrounding environment for the digital prototype is provided by the traffic simulation. The basic idea presented here aims at testing a digital prototype in a dynamically reacting world provided by a microscopic multi-agent traffic simulation [51]. Each test participant reacts to the CAV's driving functions based on the implemented driver behavior models. These models can be modified to represent a variety of different driving behaviors. Cooperative features like V2V, V2X and server communication can be included in the cooperation simulation. All tools used are executed as a co-simulation, which means that all tools run in parallel and exchange the required data through interfaces. Note, the toolchain provides the inclusion of a cooperation simulation to contribute an overall generic methodology, while the implementation in this dissertation is carried out for the vehicle dynamic and the traffic simulation.

Each simulation run is evaluated by the use of metrics depending on specific test requirements. In this dissertation the metrics are developed for a, SAE level 3 [4] or higher, driving function called highway chauffeur [23, p. 53]. Figure 3.1 shows exemplary traffic and ego-vehicle related metrics, which can be exchanged. Metrics are used for evaluating the simulation runs systematically and automatically. Again, human intervention is kept at a minimum and the evaluation is based on mathematical examination, where the results are represented by a grade and a binary threshold classifying the scenarios as either critical or not.

After evaluating a simulation run by the use of metrics the scenario verification or identification is concluded. On the basis of the steps carried out so far it is possible to extend the scenario catalog and store the gathered data into a database. The function development department can access the simulation runs for the improvement of the CAV. Scenarios that are classified as not critical are discarded or reported if the scenario was originally classified as critical by experts. This approach contributes to exclude scenarios that may have been wrongly included in the scenario catalog. Thus, experts are enabled to review their opinion based on new insights gathered by this methodology.

Additionally, a validation step can be applied depending on the required accuracy of the test and the development status of the digital prototype. Using a MiL approach offers the advantage of executing many virtual test runs in a short amount of time without the risk of endangering test drivers or cause damage to hardware. In early development stages the non-existence of the CAV forces developers to use only MiL or SiL approaches. The further the vehicle development progresses through the V-model the more test methods become available. As soon as the components are built it is possible to enhance this toolchain by HiL, ViL, mixed reality, and driving tests. Even though these methods are located on the right side of the V-model, the knowledge of these tests can provide helpful information and feedback loops for the toolchain. Data gathered by driving tests helps to evaluate the quality of the simulation environment. Thus, the gap between the simulation models and reality can be reduced. Furthermore, simulation can support driving tests by providing virtual test participants while operating a CAV on the proving ground. CAV functions can be tested without sensor weaknesses, which is very useful to assess a functionality without sensor uncertainties. In addition, dangerous tests can be performed without destroying hardware and even corner cases are testable, e.g., the functional behavior if another test participant causes a rear-end collision with high relative velocity.

The toolchain concludes with the analyses and test result preparation for the function development department. Providing the simulation data along with the used parameters of the complete run through the toolchain, including the metric calculation results, used thresholds and time points of violation a test result sheet is generated. This enables the developer to obtain a detailed evaluation of the implemented function and provides the opportunity to run the simulation again to identify weaknesses.

One of the major advantages is the generic composition of the methodology. Each aspect can be adapted for a specific purpose. Generally, the toolchain possesses the ability to use different digital prototypes, spatial domains, and metrics. Therefore, the toolchain allows for verifying and identifying critical scenarios also for more advanced automation levels and challenging applications, such as automated driving in urban areas.

Even though the toolchain is mainly prepared for development testing, the approach could be used on the right side of the V-model as well. In that case the implementation is in a mature state, the principles for verifying and identifying critical scenarios even in the validation process are similar. Limitations of the CAV can be located and the information gathered in the validation process provides helpful insights for the next generation of CAVs. The evolution of simulation tools will continue on and will improve further regarding complexity, validity, provided features, and calculation effort. The toolchain introduced in this dissertation offers a structured way to support the overall complex challenge posed by developing, testing, and releasing CAVs for customers.

3.1.1 Logical Scenario

The simulation-based toolchain starts with a logical scenario following the scenario modeling of the research project PEGASUS [111, pp. 132-133]. In order to obtain an accurate understanding of use case and scenario modeling in general, a brief introduction to this topic is given in this section. Basically, the description provided by Ulbrich et al. [112] contains three main terms for scenario modeling: Scene, Situation, and Scenario. According to Ulbrich et al. "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 [112, p. 983]." "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 [112, p. 985]." "A **Scenario** describes the temporal development between several scenes in a sequence of scenes. Every scenario starts with an initial scene [112, p. 986]." Figure 3.2 illustrates these definitions.

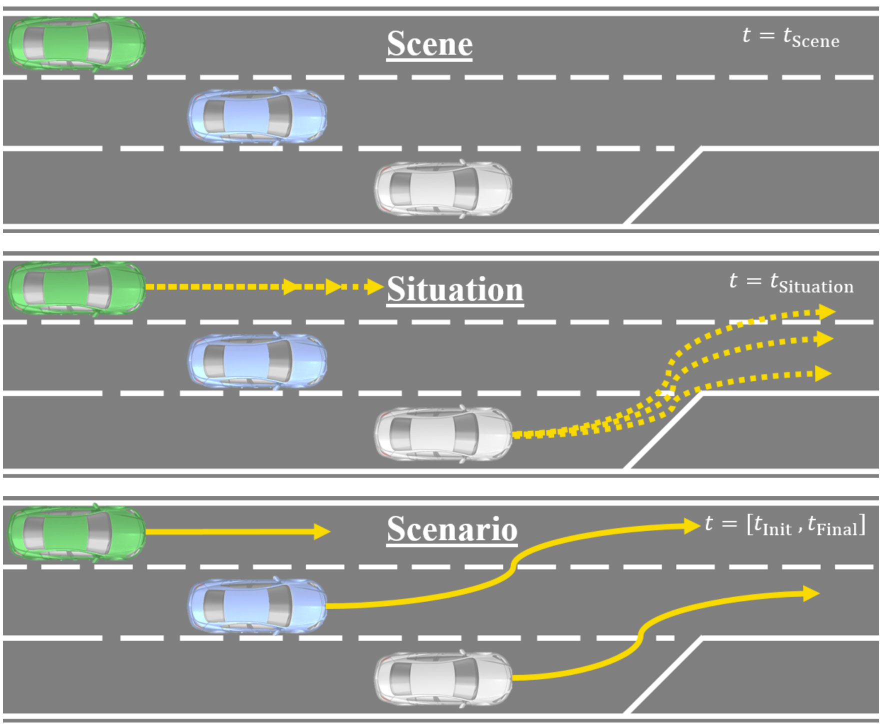

Figure 3.2: Scenario modeling according to Ulbrich et al. [112]. CAV indicated by blue. Other test participants indicated by green and gray. Defined trajectories marked by solid yellow lines. Possible future trajectories marked by dashed yellow lines.

The Scene in Figure 3.2 is defined by a certain time point of a scenario, while a scenario comprises a time interval. A scenario could be represented as a sequence of scenes.

The situation covers possible future interactions at a certain point in time that are necessary to consider for the CAV's driving functions. In general, CAV functions can be divided into several use cases. For a highway chauffeur the use cases can be, for example, entering the highway, leaving the highway, following another vehicle, etc. Hence, these use cases can be modeled by a parameter space based description called logical scenario [111, pp. 132-133]. A logical scenario is a representation for capturing parameter spaces that occur due to different possible static and dynamic attributes. Figure 3.3 exemplarily shows the scenario parameter space of a CAV driving function.

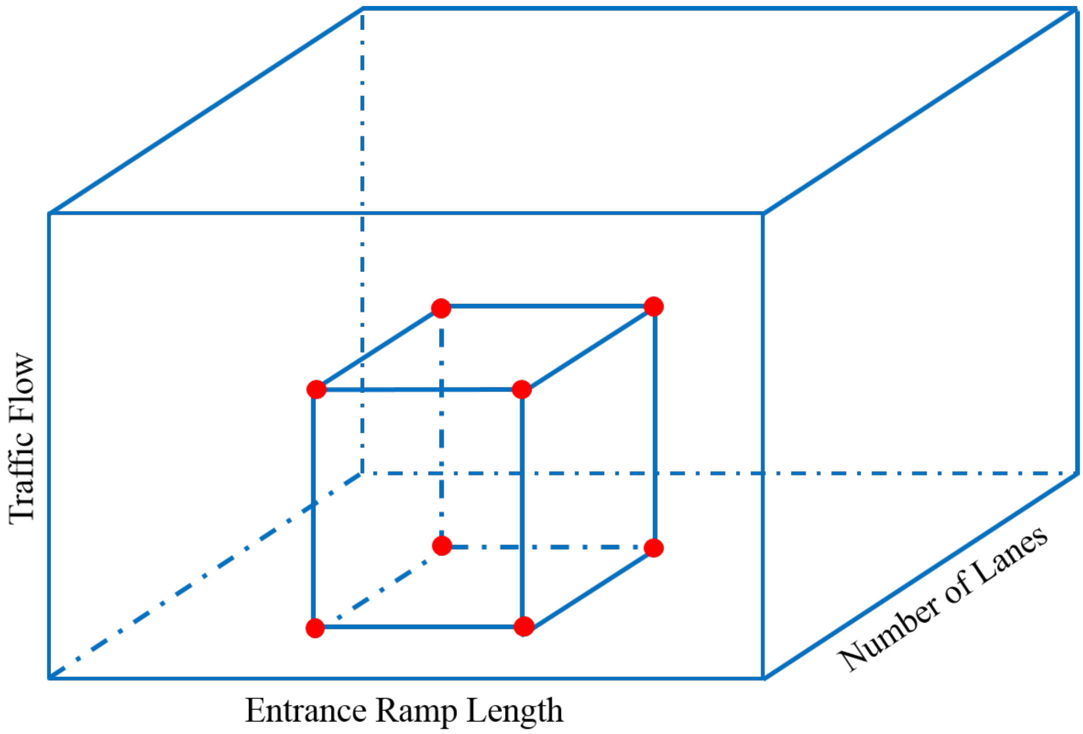

Figure 3.3: Logical scenario with a 3-dimensional parameter space for the exemplary scenario of entering the highway.

The parameter space can be represented as a multi-dimensional graphic. Figure 3.3 represents a visual example of parameter spaces that can be varied accordingly. The larger cube represents every possible parameter variation. The smaller cube shows the parameter spaces depending on the use case of the CAV's driving function. In reality the physical constraints, for example, the spatial domain or the capability of the vehicle restrict parameter spaces to a certain size. Due to the fact that there are more than three parameters for a logical scenario, the representation can be transformed into tables. Table 3.1 shows an exemplary logical scenario with parameter spaces for the use case of entering the highway.

Table 3.1: Logical scenario with an n-dimensional parameter space is represented by a table [8, p. 95].

Attribute	Parameter Space	Unit
Entrance Ramp Length	$l_{\min} - l_{\max}$	m
Number of Lanes	$N_{\min} - N_{\max}$	n.a.
Speed Limit Highway	$v_{\min} - v_{\max}$	m/s
Traffic Flow	$Q_{\min} - Q_{\max}$	veh/s
Driver Behavior	Defensive − Aggressive	n.a.
Curve Radius	$r_{\min} - r_{\max}$	m
Coefficient of Friction	$\mu_{\min} - \mu_{\max}$	n.a.

Of course, there are some uncertainties within the parameter spaces itself, but to keep the representation in Figure 3.3 and Table 3.1 vivid, uncertainties are neglected.

Parameter space based scenario modeling is the first step to narrow down the almost infinite variety of parameters for CAVs. It provides the contingency to determine constraints of the driving function's parameter space usually by expert knowledge and statistical analysis. Parameter spaces, such as "Entrance Ramp Length", "Speed Limits", etc., can be identified by statistical data examination. For example, a highway in a certain country possesses a finite number of entrance ramps that are usually known from existing maps. Based on that data the boundaries of a logical scenario can be detected. Additionally, statistical distributions can be derived to gather more insight on parameter combinations that appear more frequently than others. Regarding the toolchain, a logical scenario provides the potential to capture and vary disturbances as well. Especially, driver behavior and map errors can be described by parameter spaces. As already indicated in Table 3.1 driver behavior can be characterized, ranging from defensive to aggressive.

In summary, the toolchain models use cases as logical scenarios to tailor and limit the parameters of the investigated CAV. Later on, the simulation-based proceeding allows for varying these parameters. Simulation tools offer the possibility to change static and dynamic attributes with minimal effort. On account of the advantages that MiL is performed in a completely virtual fashion, there are almost no boundaries for varying parameters without causing harm to the CAV's prototype. The further the test environments become non-virtual the more constraints appear regarding parameter variations. Test track availability, physical limitations as well as safety aspects set stringent boundaries for testing in existing test centers. Another advantage virtual testing offers lies in the contingency that tests can be performed faster than real time and that tests do not rely on human limitations. Simulations can operate day and night without interruptions. Therefore, a lot of tests can be simulated and evaluated in a short amount of time, considering that this toolchain does not rely on human intervention if this first step is concluded and the tests are solely performed virtually.

3.1.2 Selection of Critical Scenarios

As already indicated in Chapter 2, using the state-of-the-art method critical scenarios are often derived by expert knowledge and experience. Usually, these scenarios are captured in a scenario catalog. This catalog contains use cases and is based on the scenarios that have to be tested. For automated driving, additional data is used to extend the knowledge of experts. For example real-world driving data, e.g., NGSIM [30], PEGASUS [90], GIDAS database [29], are used to identify critical scenarios.

This toolchain is able to review and evaluate these scenarios and verify their actual criticality (see Sec. 3.1.11 and 3.1.15). The scenarios declared as critical by the already mentioned methods are prepared for this approach and re-evaluated. The proceeding can be stated as follows: The critical scenarios are mapped from use cases to logical scenarios, which represents the first step of the toolchain. These scenarios are converted into a concrete scenario, where Section 3.1.4 describes the next step of the toolchain. This approach allows for the simulation-based verification of scenario catalogs containing already known critical scenarios.

One of the main benefits of this approach is that it enables developers to test their driving functions in early development stages and to obtain an understanding of the maturity of the current implementation. This provides an actual course of action for improving the CAV functions, while having the convenience to run a set of simulation-based tests, which the functions should be able to pass properly. For example, if a function developer wants to test a trajectory planner, he is able to test this function by this proceeding and improve it to a state where these tests are passed satisfactorily. The developer is able to implement the function and run a set of simulations and thereby, achieves a verification of the current level of maturity. Tests that are classified as critical demand for corrections. Thus, the developer knows exactly in which scenarios the driving function has flaws and obtains a guideline where the function has room for improvement.

Further, the evolution of the functions over time can be documented. While freezing the current state of the CAV functions the increased functionality can be seen, because the implemented functionalities should usually evolve over time. Even the deterioration of added functionalities can be captured. Sometimes, an already implemented method can be proven as a wrong approach to start with and other approaches may be better suited for this particular challenge. Other approaches can be implemented, tested and compared, which enables to select an entirely different method or rather, the choice of the best approach if more than one is implemented. In conclusion, the toolchain enables with this aspect a structured way for testing, improving, and choosing CAV functions fitting to requirements determined by experts and data stored in a scenario catalog. Another benefit is the documentation of implementation improvements over time and providing the possibility to get back to earlier implementations if the CAV functions' quality deteriorates.

3.1.3 Parameter Variation Module

Based on the parameter spaces of a logical scenario it is possible to vary parameters with a certain step size through the entire space. This allows for the start of a systematic identification of critical scenarios within the parameter spaces of the CAV's driving function. For every parameter combination a concrete scenario is generated. By varying the parameters by a certain step size one concrete scenario after another is created automatically. The step size is an important factor for the resulting amount of concrete scenarios. Grindal et al. [113] stated that the scenarios resulting from parameter combinations can be stated as [114, p. 29]:

$$S_{\mathrm{N}} = \prod_{i=1}^{N} v_{\mathrm{i}} \quad , \tag{3.1}$$

where S_{N} is the number of possible scenarios, v_{i} is the number of values selected for parameter i and N the actual number of parameters. Every parameter and chosen point by the parameter variation module increases the amount of possible scenarios. Therefore, strategies have to be applied when using this toolchain for industrial practices later on.

The parameter combination strategies can be divided into non-deterministic, deterministic and compound. Each of these strategies possesses advantages and disadvantages and are often used concurrently. [113] provides a comprehensive survey on this issue and is recommended for further reading.

In early development stages it is recommended to use a larger step size, because the CAV function is not yet in a mature stage. In this way, the developer is able to obtain a better understanding on which parameter combinations are more challenging than others. Another strategy that can be applied uses the information of the scenario catalog as starting points and varies the parameters around these particular chosen scenarios. This proceeding enables findings by varying the parameters of the critical scenario and covering the area nearby. Thus, a local criticality space can be determined and tendencies of criticality evolvements can be explored.

An initial practical approach for defining appropriate step sizes utilizes statistical data. If data of, e.g., highway entrance ramp lengths for a CAV function is available, next to a general distribution, the mean of length differences between those can be found out. This helps to identify a step size based on statistical analyses. Moreover, the smallest difference can be utilized as a worst case for discretization purposes. Figure 3.4 shows the distribution of highway entrance ramp lengths for the highways around Frankfurt (Germany). This distribution is provided within the context of the funded project Ko-HAF [56, 78].

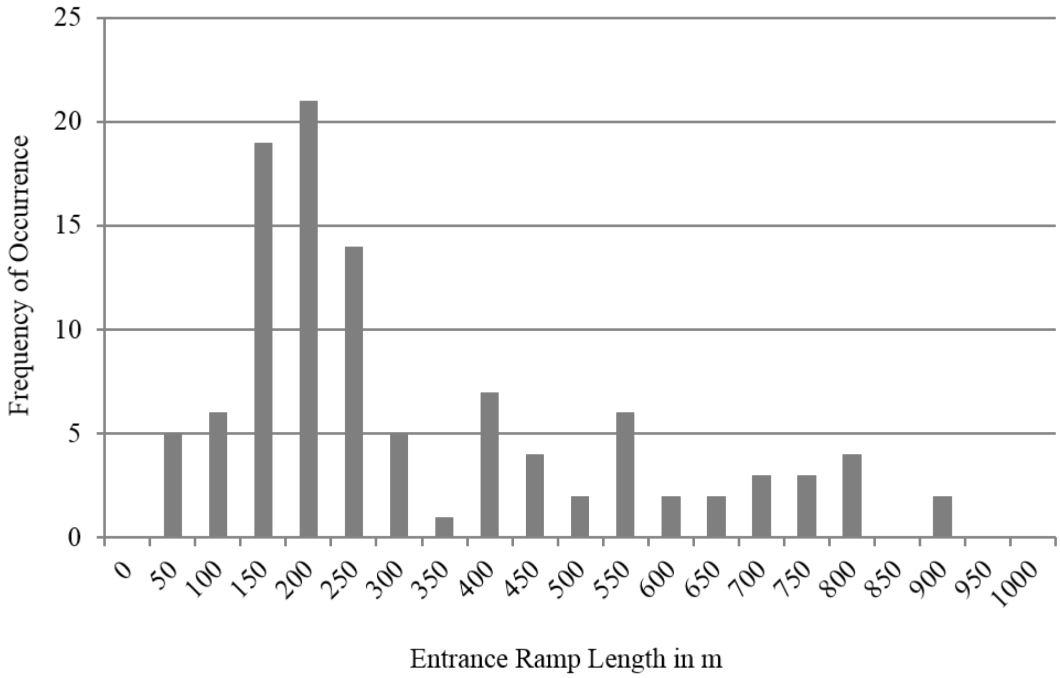

Figure 3.4: Distribution of entrance ramp lengths (Frankfurt) [78, ch. 4.1].

Most highway entrances are around 200 m long, which gives the developer an idea of the frequency of occurrence. The shortest entrance ramps are around 50 m while the longest are more than 1000 m long. The distribution shows the differentiation between highway entrances and intertwining roads occurs roughly at 350 m. For that reason, the frequency for the length of 350 m is almost zero [56, 78].

The mean value of the differences between the entrance ramp lengths lies around 5 m and the smallest difference is 1 m excluding the entrances with equal ramp lengths. This information can be used for setting up the parameter variation module. To achieve an applicable sampling the step size of 2.50 m can be considered a practical choice.

At this particular area some knowledge from experiences has to be gathered to make a more well-grounded statement. For the first applications using this toolchain in practice, a straightforward proceeding as previously described can serve as a starting point. The more the toolchain is utilized the more useful data is gathered for determining appropriate step sizes.

Another important aspect that has to be considered is the uncharted parameter combinations that trigger critical scenarios. One way that can be used to find these combinations is the so-called criticality heat map [115]. Based on criticality metrics a heat map is generated depending on the parameters chosen for a scenario. This requires prior knowledge of either existing data, simulations or the scenario catalog. A heat map is used to find regions in the parameter space that caused critical scenarios.

This method is suited to find, for example, safety critical scenarios and even worst cases. Figure 3.5 shows an exemplary criticality heat map depending on two parameters.

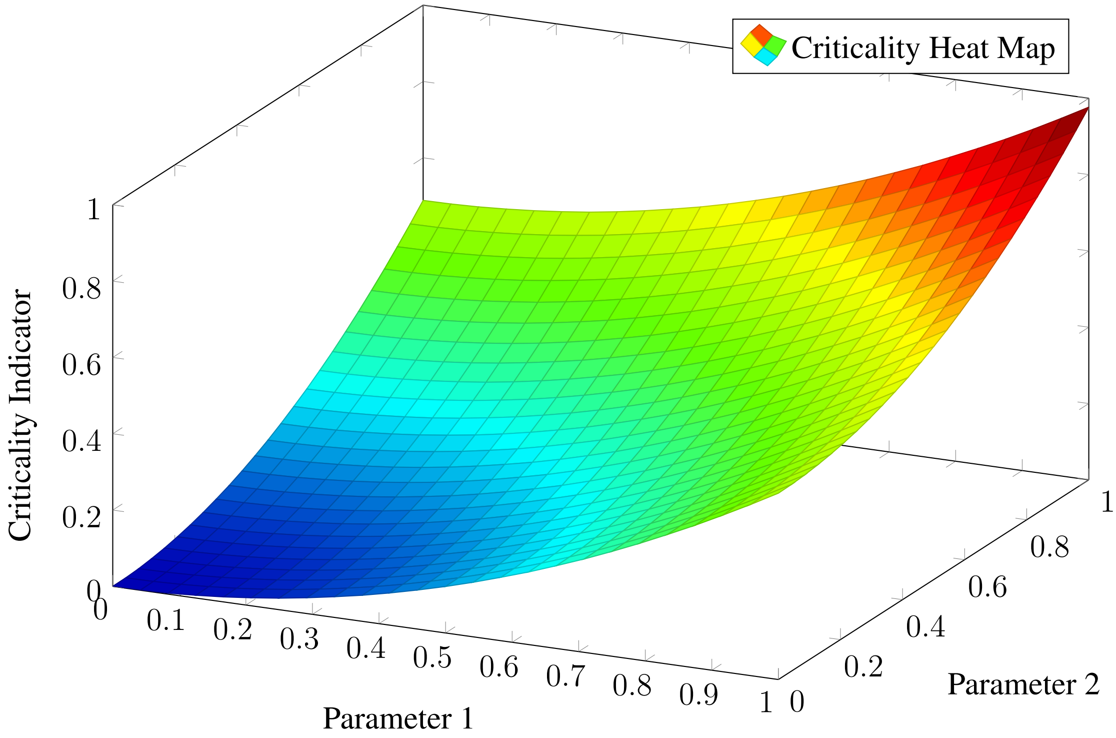

Figure 3.5: Criticality heat map, based on [115, p. 10].

In summary, the toolchain on the right-hand side facilitates the identification of critical scenarios based on a parameter variation module. This allows for the enhancement of scenario catalogs, examination of databases and expert reviews. In this way, the developer is enabled to find scenarios that are unknown at this point. It reveals weaknesses of the CAV function that have not yet been accounted for in the development phase and provides a lot of insights about insufficiencies of the current implementation. Due to the reason that the parameter variation module operates automatically it is prevented that the developer implements the CAV function only considering the requirements of the scenario catalog. This increases the robustness of the function and the identified scenarios can be considered unbiased. Even on the left side of the V-model it is important to perform robustness tests that are independent from the developers own requirements. This proceeding provides the contingency to reach a certain CAV function maturity, before the challenging task of validation even begins. Furthermore, the identified scenarios can provide helpful insights for the validation part. At least, the parameter space regions that are critical in general can be passed on to the validation engineers.

3.1.4 Concrete Scenario

A concrete scenario [111, p. 133], following the PEGASUS scenario modeling, is a detailed description that is sufficient to serve as an input for the coupled simulation environment. Static and dynamic parameters of the logical scenario are predefined for the simulation run, which is based on the selection of critical scenarios or set by the parameter variation module. Table 3.2 shows an exemplary concrete scenario, listing a few examples of predefined parameters.

Table 3.2: Exemplary concrete scenario with defined parameters.

Attribute	**Parameter**	**Unit**
Entrance Ramp Length	l	m
Number of Lanes	N	n.a.
Speed Limit Highway	v	m/s
Traffic Flow	Q	veh/s

It is possible to predefine trajectories of the test participants for the entire simulation run. This proceeding is shown on the left hand side of the toolchain, where the scenarios are mostly derived by scenario catalogs and experts. Figure 3.6 shows an exemplary concrete scenario with predefined trajectories for the involved test participants.

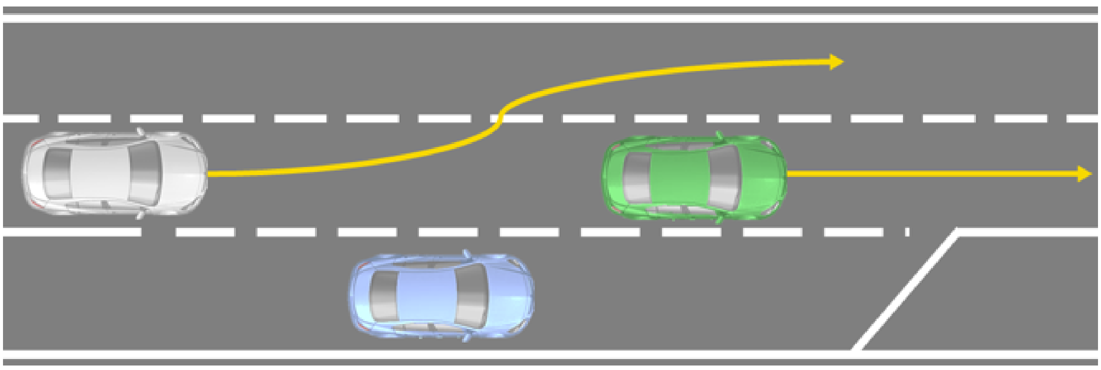

Figure 3.6: Concrete scenario with predefined trajectories for test participants. Trajectories marked by yellow lines.

The trajectory of the CAV is unknown at this point, because this trajectory is controlled by the driving functions. The traffic simulation provides the possibility to define an overall traffic flow instead of predefined trajectories. In this case, the trajectories of the involved test participants are not predefined anymore. The trajectories are determined by the driver behavior models of the traffic simulation. Basically, the driver behavior models of the traffic simulation respond to the CAV‘s behavior and the other way around.

Using this approach, the CAV is inserted into a virtual traffic environment in which each test participant possesses its own behavior. This allows for determining a concrete scenario by defining just a certain traffic flow instead of predefining trajectories for the entire simulation run. The properties of the driver behavior models can be adjusted in the traffic simulation.

A more detailed description of these models is stated in Section 2.3. To conclude the scenario description used in this dissertation an illustration of the relationship between logical and concrete scenarios is provided in Figure 3.7.

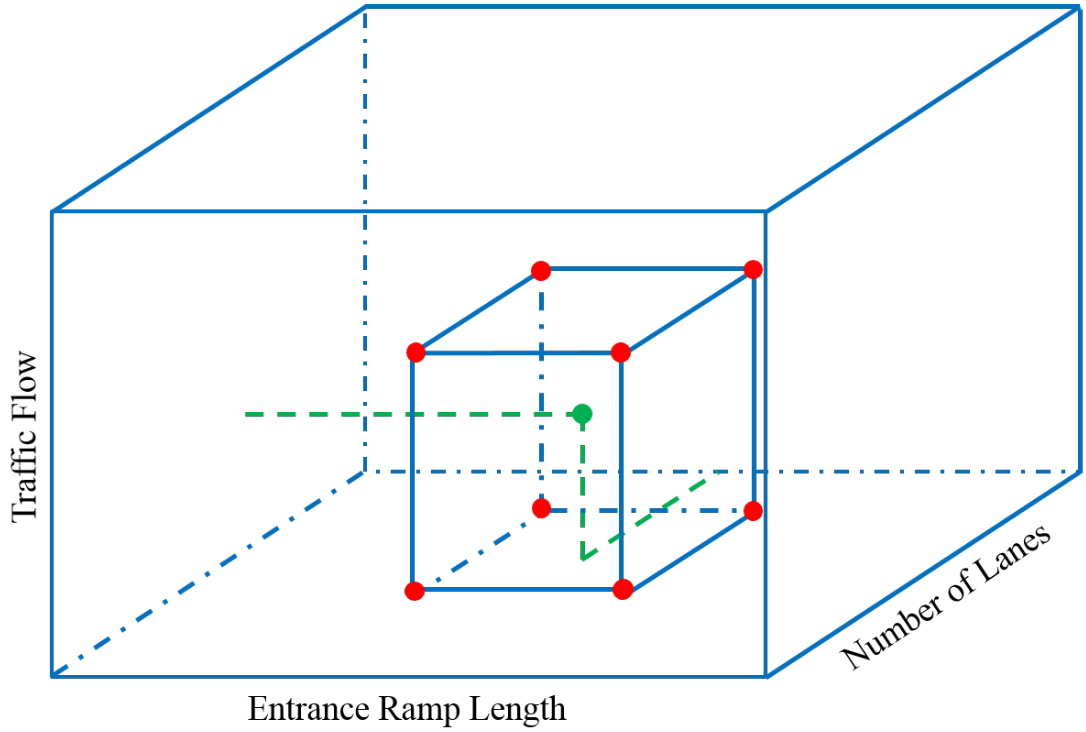

Figure 3.7: Relationship between logical and concrete scenarios. Concrete scenario indicated by green point.

The logical scenario for a certain use case represented by the small cube contains three exemplary parameters and illustrates the parameter space-based description. By determining the entrance ramp length, the number of lanes, and the traffic flow, a concrete scenario is generated indicated by the green point. Based on this description all the inputs for a simulation run are gathered.

Subsequently, the next aspect of the toolchain is the coupled simulation environment and the metrics-based evaluation. Before that a proposal is made for another scenario modeling approach that appears to be a potential alternative. Especially, the modeling of scenarios should be completely readable and conducted by machines. The goal is to exclude human intervention as much as possible.

3.1.5 Class-Based Scenario Modeling

In this dissertation the scenario modeling developed in the PEGASUS project is sufficient for the toolchain. In any case, the scenario description needs to be machine readable. For this purpose an approach for a class-based scenario description is briefly explained. The toolchain is not extended by this approach, but still the basic idea is worth mentioning. Generally, the number of scenarios are dependent on the abstraction level, which is illustrated in Figure 3.8.

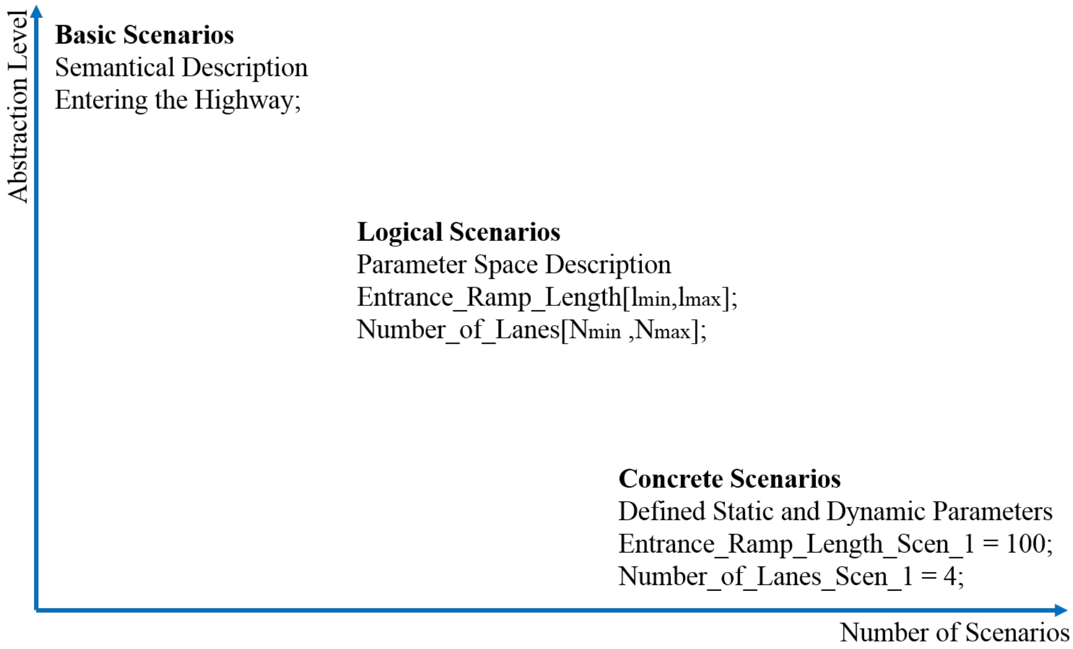

Figure 3.8: Number of scenarios and abstraction level, see also [111, p. 132]. Example: Entering the highway.

Starting with a semantic scenario description the abstraction level is high and the number of scenarios is low. By introducing parameter spaces the number of scenarios increases while the scenario modeling becomes more specific. The derivation of concrete scenarios based on these parameter spaces results in a high number of scenarios that are completely defined [111, p. 132].

This coherence resembles the principals of Object Oriented Programming (OOP). OOP is based on classes and provides the possibility for inheritance. This means that basic classes are defined and other classes can be derived from that. Considering the example of a basic scenario for entering the highway presented in Figure 3.8, the principal becomes obvious. The base class with the highest abstraction level possesses only a semantic description and maybe a few objects that are universal.

In this class universal parameters, such as country and its traffic regulations can be predefined, because these parameters are valid for every derived class as well. Next, there are logical scenarios that can be considered as a derived class that inherits the properties of the base class and adds objects that include the parameter spaces. Every single object of the derived class can now be defined by parameter spaces as defined in the PEGASUS scenario description [111]. Additionally, if necessary every derived class can access the universal objects of the base class, respectively. Concrete scenarios can be considered as a derived class of the base class and the logical scenario class, which is also derived from the basic scenario. Thus, the concrete scenario inherits the properties of both the base and the logical scenario class. Then, the concrete scenario class is extended with other objects that are important for the defined parameter description. The same principal of abstraction is valid for this modeling approach, too. The more the class is derived from other classes the more objects and properties are available. Figure 3.9 shows the concept of class inheritance for the scenario modeling approach in more detail. It should be noted that the figure is deliberately cut off on the right side indicating that many more classes exist in an ongoing scheme progressing to the right.

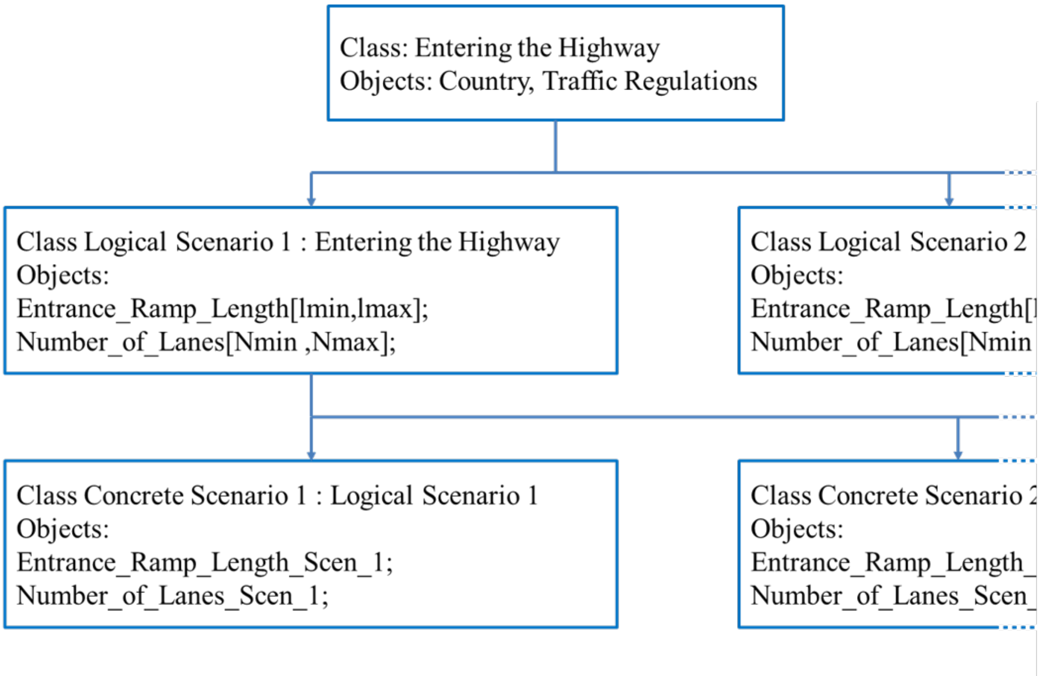

Figure 3.9: Class structure and inheritance for class-based scenario modeling.

A base class is defined as for example entering the highway. This class is declared with universally valid objects, such as country and traffic regulations as indicated in Figure 3.9. Derived from that base class other classes are generated that inherit these properties.

In this case, new objects with parameter spaces are added, e.g., entrance ramp length and number of lanes. Depending on the highways that are investigated there can be more than one logical scenario with different parameter spaces. Finally, multiple derived logical scenario classes are obtained. From each of those classes multiple concrete scenario classes can be derived. The concrete description requires the definition of the parameters to become usable for a test. Of course, the classes and derived classes possess way more objects than illustrated in Figure 3.9, but this is just an example to present the general idea.

Now, the advantages of this proceeding correlate with the general benefits OOP possesses. First of all, inheritance can be used as a useful feature. Classes can be derived from other classes, they can be copied and modified individually. The structure of a class can be adapted to the requirements of the scenario modeling. Objects and methods are implemented for this purpose. Even templates could be generated for different scenario types. Thereby, a standard template for every scenario class could be used without worrying about the basic setup of a class while using the toolchain. The allocation of data storage can be performed dynamically. Classes of concrete scenarios are added or deleted whenever necessary.

All kinds of existing software that was developed for OOP can be used to handle scenarios. For example, sorting, searching, dynamically enhancing are just a few issues that have already been solved by OOP. Broadly speaking, every open source software that already exists for OOP can be used and adapted for this purpose. Especially, because of the widespread application of OOP, a lot of solutions for many different problems already exist and do not need to be implemented again.

Considering the actual need of a standardized machine-readable scenario modeling it seems obvious to make use of existing and established approaches, such as OOP. This method has been used for a long time and proven to be very useful. Almost every software tool possesses an interface that supports C++, which is based on OOP. If established, this scenario modeling framework can be standardized and used for different software tools, or even as an interface across different test environments.

As a matter of fact, the existing scenario modeling needs a practical and intuitive way to access, store, and handle the arising vast amount of data. All in all, this idea is not implemented yet and needs further research and experience on how applicable it is in practice. But generally, it seems to be an approach worth to consider for scenario modeling of CAVs, especially because no standard has been established yet.

Regarding future work, the idea should be implemented and the usability in practice should be investigated. If the method proofs to be applicable, it could be used for handling scenarios in a standardized fashion providing all the advantages of already mentioned OOP methods. Finally, it would ease the process of making the toolchain machine-readable and automated.

3.1.6 Coupled Simulation Environment

The main goal of this approach is to use the advantages of each individual simulation tool and combine these into a coupled simulation environment for the verification and identification of critical scenarios. Figure 3.10 shows the coupled simulation environment and the features of the tools separately.

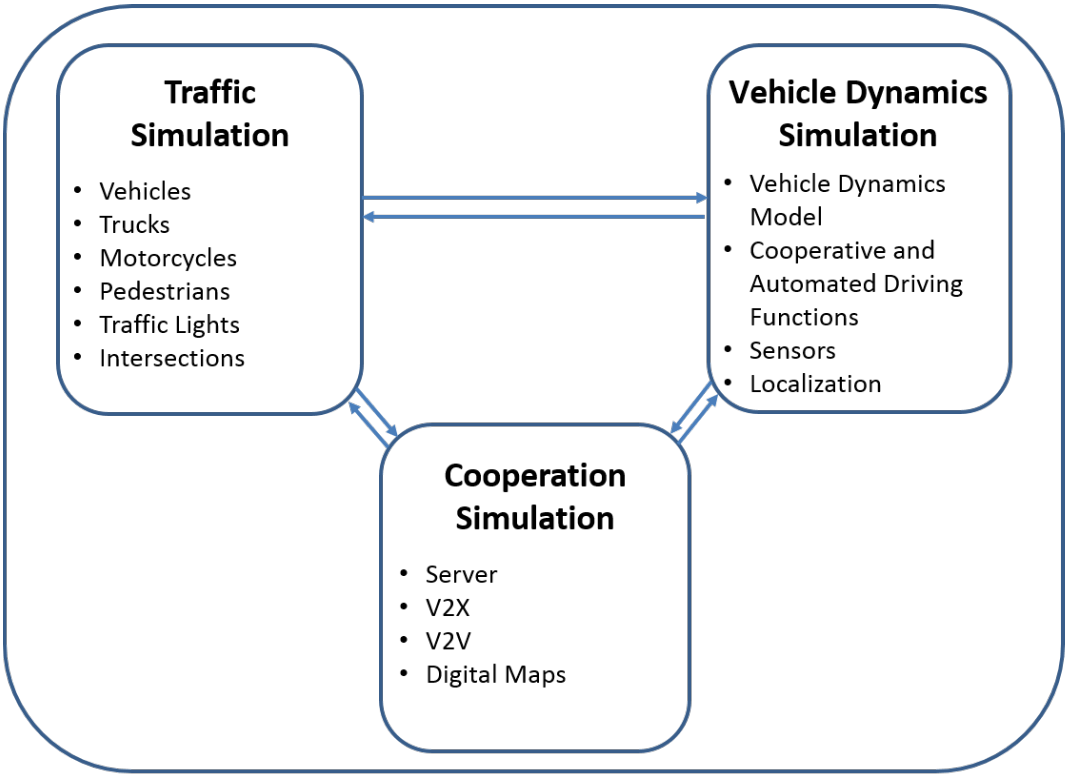

Figure 3.10: Coupled simulation tools and their individual features, based on [8].

The vehicle dynamics simulation includes a detailed simulation model of the CAV, the so-called digital prototype. As already stated in Section 1.2, a digital prototype consists of a vehicle dynamics model, virtual sensors, automated driving functions, etc. The traffic simulation provides the dynamical surroundings for the digital prototype, which can be other test participants, for example, vehicles, trucks, pedestrians, etc. Additionally, traffic light and intersection simulations are feasible. Using a cooperation simulation, cooperative features, such as servers, V2V, V2X, digital maps, etc., can be applied. To stay focused on the toolchain's capabilities only the concept and some aspects of the implementation are briefly explained in this section.

The simulation environment used for this dissertation is based on two simulation tools: a vehicle dynamics and a traffic simulation tool. These tools are coupled and are executed as a so called co-simulation.

Basically, the two tools are running in parallel and exchange information, such as positions, velocities, orientations, etc., through their interfaces. The vehicle dynamics simulation aims at modeling the virtual CAV's behavior. This tool simulates the CAV using the implemented driving functions provided by the function development department. The vehicle model is a complex and detailed virtual representation of the CAV (see Sec. 2.3). The vehicle model can be equipped with virtual sensors driving in a virtual world consisting of roads, lane markings, guard rails, buildings, test participants, etc. The CAV functions are developed in Matlab/Simulink [116], which possesses an interface for controlling the digital prototype. Driving functions, such as trajectory planning, motion control, maneuver prediction, driving strategy, are developed and provide guidance for the digital prototype through the interface of the vehicle dynamics simulation.

The microscopic traffic simulation tool is a multi-agent simulation providing other test participants for the virtual testing of the CAV. Each agent is equipped with a driver behavior model that can be parameterized individually. This means that each agent reacts according to its own behavior model, which is also dependent on other agents' behavior. Simulating and parameterizing each agent individually provides an intuitive way for changing the driving behavior of other test participants surrounding the CAV. Parameters, such as maximum acceleration and deceleration, reaction time, and the minimum gap for car following purposes, can be adjusted. The digital prototype's odometry can be retrieved from the vehicle dynamics simulation and communicated to the traffic simulation, as the digital prototype being an agent itself. The other agents within the traffic simulation react to the digital prototype using each individual agent's driver behavior model. The agents of the traffic simulation are transferred to the vehicle dynamics simulation as traffic objects. The digital prototype's virtual sensor setup detects the traffic objects controlled by the multi-agent traffic simulation and uses this information as an input for the implemented driving functions.

A cooperation simulation can be used to enhance the framework even further. Future CAVs will feature more and more cooperative functions, e.g., cooperative decision making, trajectory planning, mapping etc. To incorporate these features the toolchain provides the contingency of a cooperation simulation enhancement. In this dissertation the inclusion of the cooperation simulation will not be performed, but the methodology comprises this functionality and can be elaborated in subsequent research.

Co-simulations are useful tools to tackle these challenges, and for that matter, an important part of the entire methodology. This approach aims at testing a digital prototype in a virtual environment with individual test participants reacting to the prototype's driving functions. The driving behavior of the test participants can be changed individually and therefore, the CAV has to be able to operate in a virtual environment with changing driving behavior. It is anticipated that this proceeding allows for gaining insights into the current implementation of the CAV and to identify critical scenarios no one has thought of up to this point in time.

3.1.7 Static Coupling

The foundation of the coupled simulation environment is a geometrically consistent representation of the digital environment in both simulation tools. In other words: the the digital prototype and the other test participants have to drive in the same digital world. Therefore, the maps must be geometrically consistent in both simulation tools. Based on a map data acquisition of a road network provided in OpenDrive [117], the map data needs to be converted to the static representations required by each simulation tool. Figure 3.11 shows the map data conversion scheme.

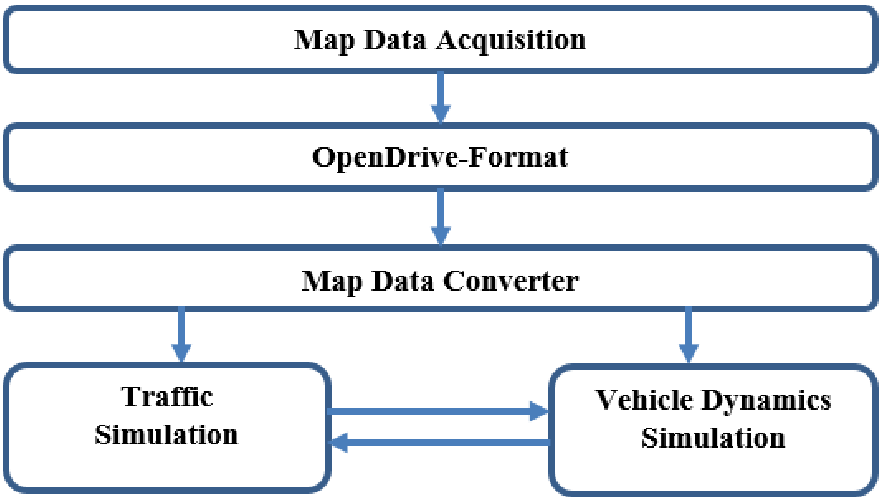

Figure 3.11: Map data conversion scheme [8, p. 97].

The map data acquisition is performed by measuring a road network using high precision sensor technology. Therefore, a measurement vehicle is equipped with a sensor setup including DGPS and laser scanners. These sensors allow for gathering of road data, such as road geometry, lane markings, traffic signs, etc. The measurement data is processed and converted to the OpenDrive-format [117]. Based on the provided data the static coupling is performed by converting the OpenDrive data to the road network representations of the individual simulation tools. The road network representations of the simulation tools differ to the provided OpenDrive standard. To keep this section sufficiently compact a short description is given to explain how the map data conversion is done and what the main differences are. Afterwards, a list of road properties is provided to demonstrate which aspects are converted.

The OpenDrive standard is setup by an inertial reference frame and sub-frames from which splines are calculated. Starting with a coordinate frame transformation the subframes are located insight the map pointing in the tangential direction of the road shape. From this point a third order spline represents the so-called reference line to the next sub-frame, where a new sub-frame is added.

Dependent on that reference line the corresponding lanes are calculated. The elevation profile is also represented by cubic splines adding the shape in the z-direction. Lateral slope or rather road banking along the reference line is given by a cubic spline as well. Additionally, traffic signs are represented as objects that consist of features, for example, position, speed limit, etc.

The vehicle dynamics' static representation is similar to OpenDrive. In contrast to OpenDrive the road shapes have to be given as a point list, while the interpolation is done by the simulation tool itself. In order to do this, a discretization of the spline is performed to obtain necessary points. The required attributes for the vehicle dynamics simulation are converted sequentially, providing a suitable digital environment. Of course, the data conversion is set to required attributes.

For the traffic simulation the road shape representation is less complicated. The road geometry can be converted by calculating points along the road center line. Properties, such as elevation profile and road banking are not supported. Every road is equipped with an ID, an index, a speed limit, length, and shape. The shape is stored as a polyline, while the index is counted upwards from the rightmost lane. Additionally, logical connections are stored in the traffic simulation to ensure that the lanes are connected correctly. The decision to include driving lanes only was made to keep the road network of the traffic simulation as small as necessary and roads that can not be driven on can be excluded for this purpose. Figure 3.12 shows the resulting maps for the highways around Frankfurt (Germany).

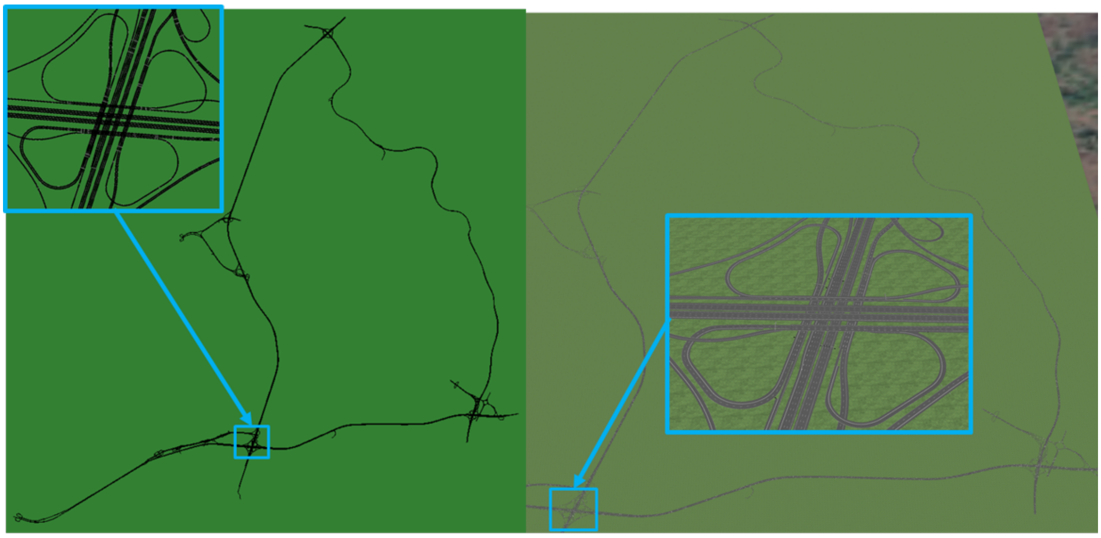

Figure 3.12: Static coupling for the highways around Frankfurt (Germany). Left side: traffic simulation. Right side: vehicle dynamics simulation [118].

Based on the map data conversion it is possible to implement the dynamic coupling as described in Section 3.1.8.

The attributes that are converted for each simulation tool are listed in Table 3.3. For more details, the interested reader is referred to the master's thesis of Lukas Zaruba [118].

Table 3.3: Road properties that are converted for the static coupling of the simulation tools [118, p. 33].

Road Property	**Vehicle Dynamics Simulation**	**Traffic Simulation**
Logical Connections	no	yes
Traffic Signs	yes	no
Road Geometry	Cubic Splines	Polylines
Road Surface	yes	no
Road Elevation Profile	yes	no
Road Banking	yes	no
Lane Types	All Types	Driving Lanes
Lane-Specific Information	no	yes
Road Markings	yes	no
Road Objects	yes	no

The static coupling is a necessary step to build a comprehensive simulation environment. The map standards differ in some cases quite strongly, which does not allow for a conversion of all properties. Tool providers assure that their tools are going to support OpenDrive in the near future, but currently most of the tools do not support the OpenDrive standard. Furthermore, one of the most important issue is the quality of the map data acquisition itself. A lot of map errors where found along this conversion process, which are easy to spot in a simulation tool. Most errors occurred in road markings, logical connections, lane types, discontinuity in the elevation profile. These errors had to be corrected mostly by hand, which is a challenging task depending on the size of the map. Another problem is the absence of validity of the acquired maps. It is simply not possible to examine every road of a highway network. Some help could be provided by Simultaneous Localization And Mapping (SLAM) algorithms [119], which aim at creating map updates using the CAV's sensor setup.

All in all it is important to keep these issues in mind when setting up the static coupling. Further, it underlines the fact that map errors exist in practice and if not detected can lead to problems even in the setup of a simulation tool. Considering that map information is used for the development of driving strategies, it is obvious that these errors can lead to difficulties for the CAV. Simulation is a powerful tool for performing at least a plausibility check of the acquired map data. At least some errors, such as large discontinuities in the elevation profile can be detected by the vehicle dynamics simulation. Regarding usability of the toolchain, the static coupling represents the foundation of the co-simulation framework, which is also capable of exchanging dynamic attributes, as described in the next section.

3.1.8 Dynamic Coupling

In order to couple the dynamical attributes properly, the simulation tools' state space variables have to be exchanged through their interfaces. In order to do this, it is worth to mention that the simulation tools significantly differ regarding granularity of models, sample time, and computational effort. While the vehicle dynamics simulation requires a lot of detail calculating the specific dynamical models, the agents of the traffic simulation are less detailed. This is not particular problematic, because the digital prototype's behavior has to be more precise due to the fact that this is the model for which the driving functions are implemented. Basically, the driving functions that are developed are controlling the digital prototype. The traffic simulation provides the dynamical surroundings based on driver behavior models for each agent. An overview of the tools in particular is provided in Section 2.3. In order to illustrate the dynamic coupling Figure 3.13 shows the geometrically consistent test track of the long straight in both tools located on the Opel proving ground in Dudenhofen near Frankfurt. Beneath, the digital prototype and other test participants are illustrated.

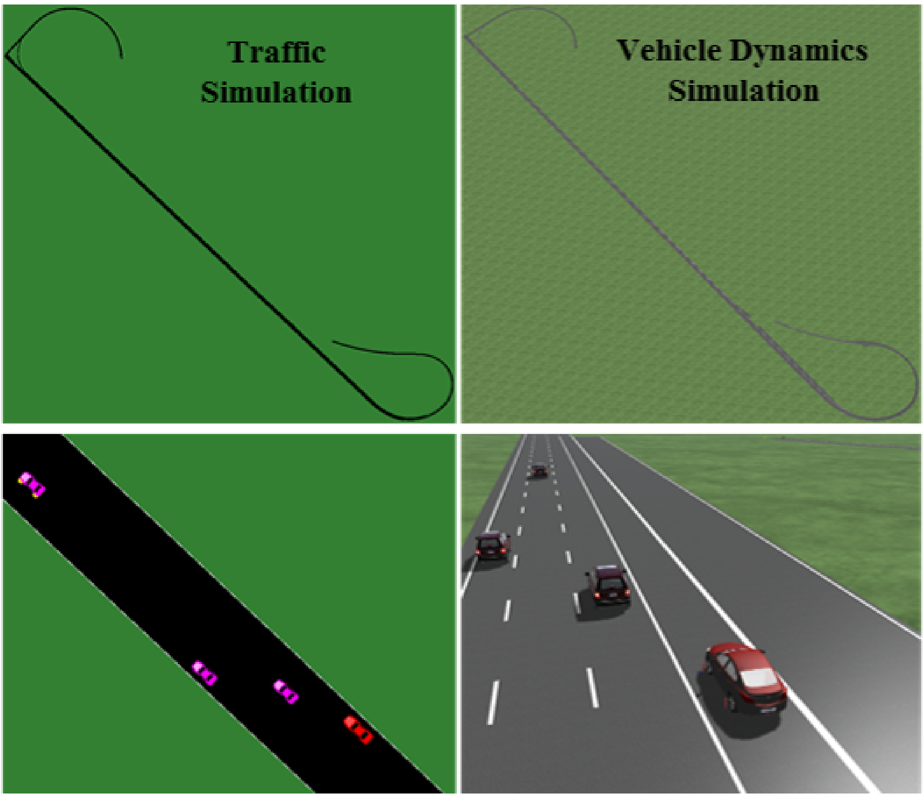

Figure 3.13: Proving ground representation in different simulation environments. Left-hand side: Traffic simulation. Right-hand side: Vehicle dynamics simulation. Top segments: Proving ground overview. Bottom segments: CAV surrounded by other test participants [8, p. 97].

Both tools run in parallel as a so-called co-simulation. Due to the fact that the traffic simulation possesses another sample time than the vehicle dynamics simulation, a motion prediction needs to be implemented. The vehicle dynamic is calculated with a sample time of 0.001 s, while the traffic simulation uses a sample time of 0.1 s. This would result in a loss of information and poses a problem for the digital prototype that relies on this information gathered by its virtual sensor setup. In order to solve this issue a constant velocity prediction [120] is implemented as shown in Figure 3.14.

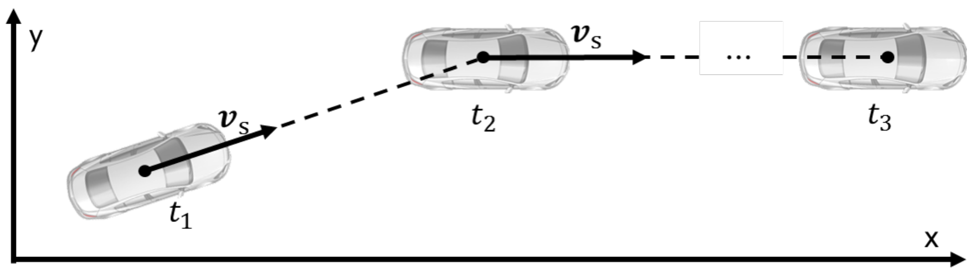

Figure 3.14: Motion prediction is required due to the difference in sample times [8, p. 98].

The directed velocity vector of every test participant is predicted to be constant until new information is available. Therefore, the driving function, which operates on its own sample rate, will always be able to receive information about the current state space variables of the test participants whenever required. In this way, it can be prevented that the test participants vanish and appear in an uncontrolled manner. This is particularly important for motion prediction algorithms and the development of driving strategies.

In addition, the computational effort should be limited to an acceptable minimum. The traffic simulation tool is able to quickly perform simulations with a vast amount of test participants, while the vehicle dynamics simulation's performance decreases significantly when more test participants are added. Hence, the test participants simulated in the vehicle dynamics simulation should be kept as small as necessary. In this case, the restrictions of the CAV's sensor setup help to reduce the computational effort. Usually, the sight of a CAV possesses boundaries due to limited sensor ranges. This means that only test participants within a region of interest have to be considered. Figure 3.15 shows the concept in more detail. Only the test participants that are within this region of interest, illustrated by the black rectangle and implied by the sensor restrictions of the CAV, are taken from the traffic simulation and considered in the vehicle dynamics simulation. Broadly speaking, this approach enables the simulation of large traffic systems without increasing the computational effort of the vehicle dynamics simulation tool. The test participants that are not inside the region of interest, but still positioned in a larger area, are calculated by the much less demanding traffic simulation tool.

The traffic simulation tool already provides an interface to control and retrieve aspects of simulated test participants. This interface is used to place the CAV into the traffic environment and retrieve the state space variables of the test participants inside the region of interest. Thus, the CAV reacts to the test participants according to the implemented driving functions and the test participants react to the CAV according to the driver behavior models provided by the traffic simulation tool.

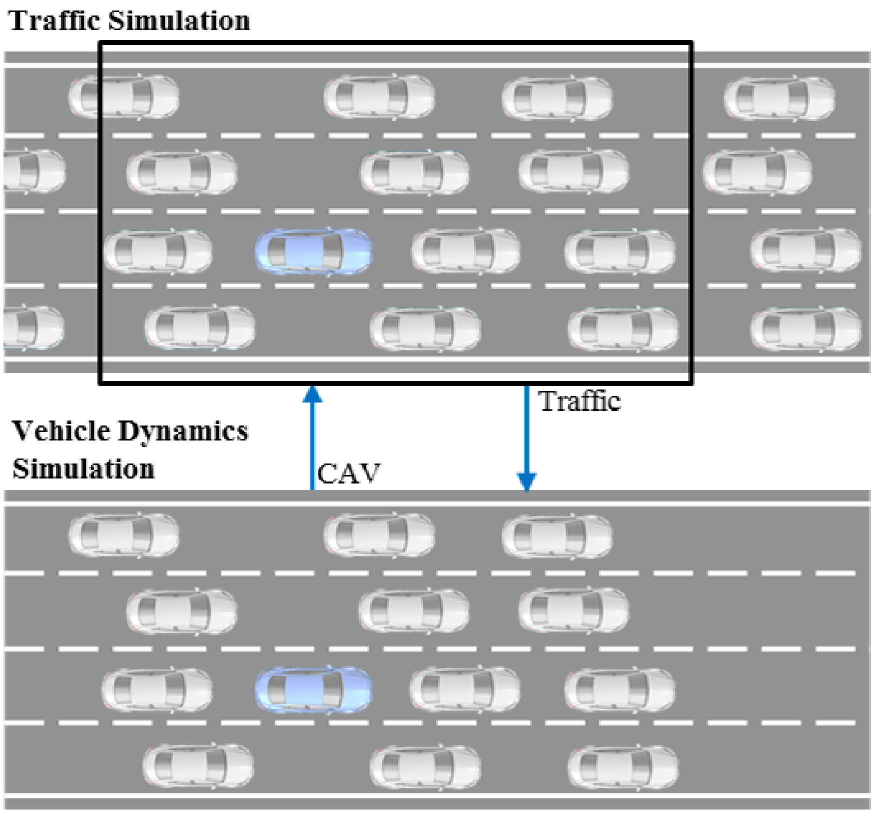

Figure 3.15: Region of interest for the dynamic coupling [8, p. 98].

The cooperation simulation is not elaborated in this dissertation. Nevertheless, the toolchain still provides an enhancement for these features and the capability to include cooperative simulation tools, which is part of current research that is built on this methodology. In particular, cooperative decision making algorithms will be evaluated by the use of this toolchain [121].

In summary, the dynamic coupling facilitates to simulate the CAV inside a traffic environment. This provides a MiL framework to verify and identify critical scenarios even in early development stages. Obviously, to perform a lot of concrete scenario simulations in a short amount of time without endangering test drivers or components is a major benefit of this simulation-based approach.

3.1.9 Quality Metrics Traffic

In order to show the capabilities of the toolchain to evaluate the impact of CAVs on traffic quality, metrics are introduced in this section. In general, the toolchain possesses the ability to use different kinds of metrics. In this dissertation, two possible metrics are elaborated, where the focus of this section lies on traffic quality. It is worth mentioning that the developed metrics are only sensitive for this evaluation purpose. Furthermore, it is not the objective to develop metrics that are universally valid. The metrics are rather used to illustrate the capabilities of the simulation-based toolchain.

To evaluate traffic quality a literature-based assessment of traffic quantities is performed. This approach aims at combining different traffic quality related sub-metrics. Therefore, traffic state space variables, such as density, mean velocity, velocity fluctuations, etc., are investigated. The requirements of the combined metrics can be stated as follows [8, p. 99]:

- All critical scenarios need to be identified.
- The combined metrics evaluation should aim for a low False Positive Rate (FPR).
- An overall assessment grading system should be used.
- The classification for the combined metrics should be binary.

The first sub-metric evaluates the change of the macroscopic traffic quality following the Highway Capacity Manual (HCM) [122]. The HCM is an internationally accepted guideline for traffic planning. According to HCM the evaluation of macroscopic traffic quality can be obtained by investigating the traffic flow rate and the average travel velocity on a certain highway segment over a time interval of 15 minutes by calculating the traffic density

$$D = \frac{v_\mathrm{P}}{S} \quad , \tag{3.2}$$

and compare this value to assessment Table 3.4, where v_P is the passenger car equivalent traffic flow rate and S the average travel velocity [122, p. 23-12].

Table 3.4: Level Of Service (LOS) assessment table for freeway/highway [122, p. 23.3].

Criteria	**LOS A**	**LOS B**	**LOS C**	**LOS D**	**LOS E**	**LOS F**
Max Density (pc/km/ln)	0-7	>7-11	>11-16	>16-22	>22-28	>28
Min Speed (km/h)	120	120	114.6	99.6	85.7	n.a.
Max (v/c) ratio	0.35	0.55	0.77	0.92	1.00	n.a.
Max Flow Rate (pc/h/ln)	840	1320	1840	2200	2400	n.a.

Based on this established proceeding, the evaluation method is tailored to assess the impact of CAVs on traffic quality for highways. The LOS method allows for using different adjustment factors, for example the heavy-vehicle adjustment factor

$$f_{\mathrm{HV}} = \frac{1}{1 + P_{\mathrm{T}}(E_{\mathrm{T}} - 1) + P_{\mathrm{R}}(E_{\mathrm{R}} - 1)} \quad , \tag{3.3}$$

where each E_{T} ,E_{R} are passenger car equivalents, P_{T}, P_{R} are the proportion for trucks/busses and Recreational Vehicles (RVs) in the current traffic flow. Additionally, a peak-hour factor P_{HF} and a driver population factor f_{P} can be adjusted in the assessment [122, p. 23-8]. The passenger car equivalent traffic flow rate can be calculated by

$$v_{\mathrm{P}} = \frac{V}{P_{\mathrm{HF}} N_{\mathrm{L}} f_{\mathrm{HV}} f_{\mathrm{P}}} \quad , \tag{3.4}$$

where V is the general traffic flow rate without differentiating vehicle types and N_{L} is the number of lanes [122, p. 23-7].

Adjustments made compared to the original approach are as follows:

- The time interval is changed to 15 seconds.
- The traffic density is obtained directly from simulation data.
- A unified Domain Of Interest (DOI) for all highway segments is proposed.
- The change in traffic quality between two time intervals is considered.

The DOI is chosen to be 450 m following the suggestions of the HCM. In order to evaluate the impact of CAVs on the current traffic situation, the time interval of 15 minutes to an hour, which is originally proposed by the HCM, is insufficient for this purpose. The time interval is changed to 15 seconds for the following reasons: If the time interval is too long, the direct effects of the automated vehicle can not be evaluated. If the interval is too short, there are just a few sample points that can be used for the evaluation. The time interval of 15 seconds is chosen following the suggestions of [123]. Further, Zhu Weihua et al. also used a time interval of 16 seconds for evaluating traffic quality of a vehicle [124]. Considering this, the choice of a time interval in this range seems reasonable. One of the major advantages of simulation-based approaches is the possibility of having omniscient view of all state space variables. For that reason, the traffic density can be obtained directly and calculated by

$$D = \frac{D_{\mathrm{veh}}}{P_{\mathrm{HF}} f_{\mathrm{HV}} f_{\mathrm{P}}} \quad , \tag{3.5}$$

where D_{veh} is the general traffic density obtained from the simulation [125, p. 37]. The heavy vehicle adjustment factor, the peak-hour factor, and the driver population factor are set following the recommended values of the HCM. Table 3.5 shows the recommended factors.

Table 3.5: Factors for the LOS assessment of traffic quality recommended by the HCM [122, p. 12-10].

Factor	**Value**
f_{HV}	$\frac{1}{1+0.1(1.5-1)+0}$
P_{HF}	1
f_{P}	1

To investigate the change in macroscopic traffic quality between two time intervals the following grading equation is used and normalized [8, p. 100]:

$$G_{\text{mac}} = \frac{LOS_{\text{t}} - LOS_{\text{t}-1}}{5} \quad . \tag{3.6}$$

The denominator of equation 3.6 results from the maximal change of the LOS grade from A to F, which is represented by the value 5. To exclude an improvement of the macroscopic traffic quality by the CAV, which can be stated clearly as not critical, for example a change from LOS grade F to A, negative values are taken into account as $G_{\text{mac}} = 0$. This value represents the best grade for the traffic evaluation. Based on the preparatory steps the macroscopic grade varies from 0 to 1 and can be used later on for the combination with further traffic quality evaluation metrics that are also prepared to vary in the same range. Figure 3.16 shows the concept of the macroscopic metric.

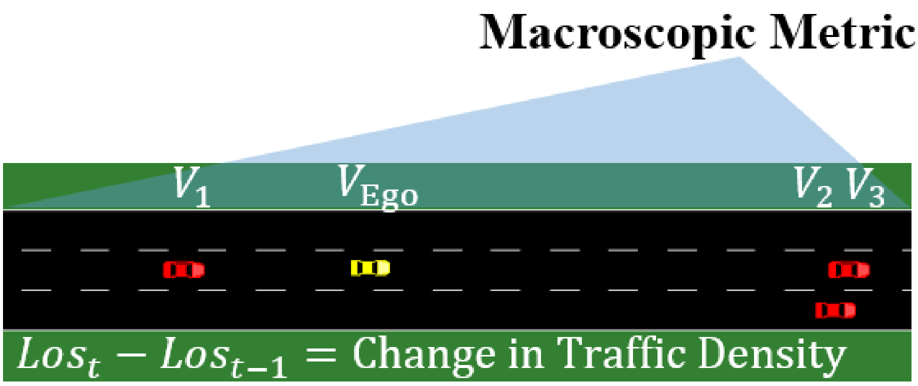

Figure 3.16: Concept of the macroscopic metric [8, p. 100].

The second sub-metric is called microscopic traffic quality and follows the research of [124]. Zhu Weihua et al. stated that the traffic quality of a vehicle can be described by the fraction of velocity fluctuations and the average travel velocity of a vehicle within a certain time interval. The basic idea is explained by the following example. Imagine a vehicle is driving on a highway without any traffic. In this scenario, the average travel velocity is going to be high and the velocity fluctuations small. If more traffic is added to this scenario the average travel velocity is going to decrease and due to the interactions with other traffic participants, the velocity fluctuations are going to increase. In this dissertation the following extensions of this basic principal are carried out:

- The time interval is also set to 15 s.
- The indicators are extended from one vehicle to all vehicles within the DOI.
- A reference value is determined to make the grade comparable to the other sub-metrics.

The so-called coefficient of variation can be calculated by [124, p. 49]

$$CV_{\mathrm{j}} = \frac{\sigma_{\mathrm{vj}}}{\bar{v}_{\mathrm{j}}} \quad , \tag{3.7}$$

where σ_{vj} is the standard deviation and $\bar{v}_{\mathrm{j}}$ is the mean velocity of every vehicle traveling through the DOI [8, p. 100]. The values of equation 3.7 are sampled every second and used along with the average velocity to evaluate the microscopic traffic quality and can be stated as:

$$G_{\mathrm{mic}} = \frac{\frac{\overline{CV}}{CV_{\mathrm{ref}}} + (1 - \frac{\bar{v}}{v_{\mathrm{ref}}})}{2} \quad , \tag{3.8}$$

where $\overline{CV}$ is the mean coefficient of variation and $\bar{v}$ is the mean velocity, both with respect to the DOI and the time interval [8, p. 100]. The coefficient of variation is now combined with the mean velocity, which provides further information that is used for the evaluation. The metric should only deflect if both values indicate that the traffic quality is poor. If a congestion occurs, usually the velocity fluctuations are high and the average velocity is low. Figure 3.17 illustrates the concept of the microscopic metric.

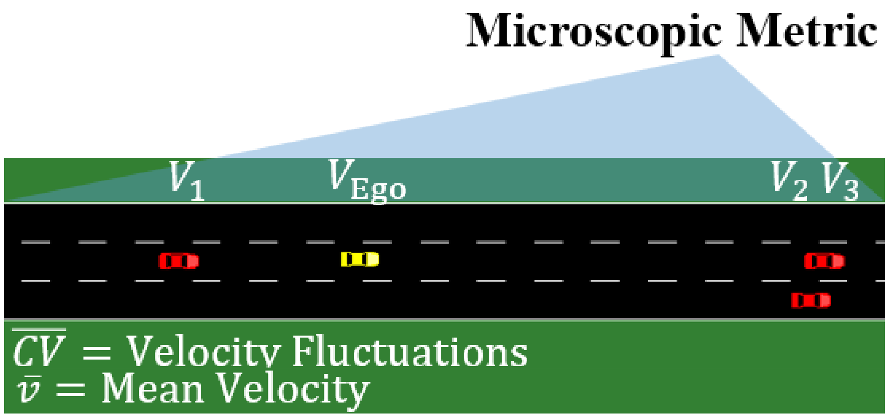

Figure 3.17: Concept of the microscopic metric [8, p. 100].

The variables with the subscript "ref" are introduced for the following reason: In order to combine the sub-metrics later on, the individual grades need to range from 0 to 1. Considering that, for example $\overline{CV}$ varies from 0 to 0.1 during the evaluation of different typical traffic scenarios, one might consider a reference value of 0.1 for normalizing the equation. This proceeding would be insufficient for the determination of the reference values, because the metrics should be sensitive enough to evaluate smaller changes within the full range. The values 0 and 0.1 are the minimum and maximum values for this indicator and therefore rare to find. To obtain a better fitting reference value a trade-off between sensitivity and robustness is chosen. Using the traffic simulation tool 75 typical traffic scenarios are generated, the criticality is evaluated individually, and the indicator is observed. The resulting reference value is obtained from a Receiver Operator Characteristics (ROC) graph [126], where the FPR is zero, which represents a trade-off between sensitivity and robustness. The used reference values in this dissertation and additional information can be found at [125, pp. 37-45]. Further, the reference value for the mean of the velocity can be extracted according to suggestions of the HCM [122, p. 23-3].

The third sub-metric is called nanoscopic traffic quality and captures close range interactions of the CAV. To achieve this, a moving circular DOI following the position of the CAV is proposed. Only the vehicles that are surrounded by the circle are used to calculate the nanoscopic metric by using

$$DV_{\mathrm{j}} = \frac{\sigma_{v_{\mathrm{Circle,j}}}}{\bar{v}_{\mathrm{Circle,j}}} \tag{3.9}$$

as the coefficient of variation, where $\sigma_{v_{\mathrm{Circle,j}}}$ is the velocity standard deviation of all vehicles inside the circle and $\bar{v}_{\mathrm{Circle,j}}$ is the mean velocity of all vehicles within the circular DOI [8, p. 100].

The overall nanoscopic metric can be calculated by

$$G_{\text{nan}} = \frac{\frac{\overline{DV}}{DV_{\text{ref}}} + (1 - \frac{\bar{v}_{\text{Circle}}}{v_{\text{ref}}})}{2} \quad , \tag{3.10}$$

where $\overline{DV}$ is the mean coefficient of variation and $\bar{v}_{\text{Circle}}$ is the mean velocity with respect to the circular DOI [8, p. 100]. The reference values are determined by the same method mentioned in the microscopic metric analysis. Figure 3.18 illustrates the nanoscopic metric using the circular DOI following the CAV.

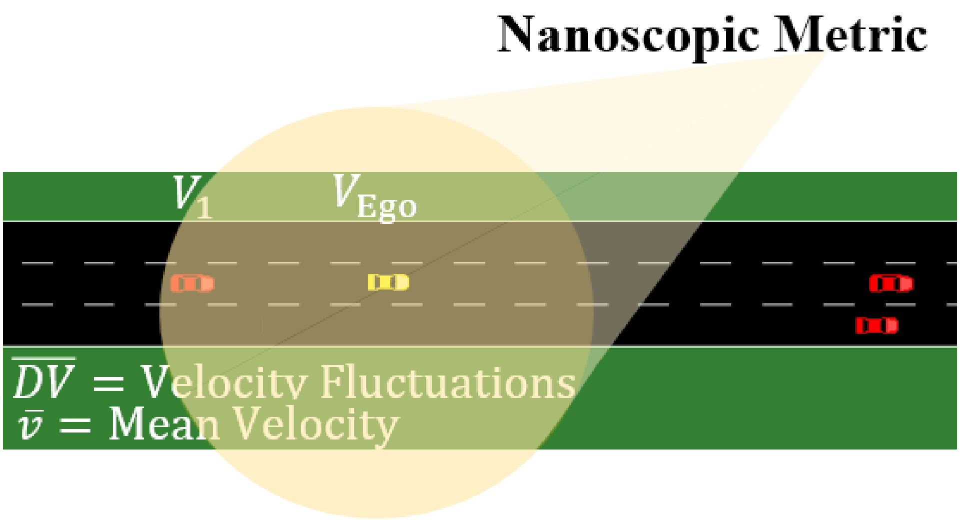

Figure 3.18: Concept of the nanoscopic metric [8, p. 101].

The last sub-metric used for the evaluation of traffic quality is called individual metric [127]. This metric aims at assessing the CAV's state space variables. Due to the assumption that large acceleration changes are usually not desired for the CAV, this indicator can be used to gather further information for the traffic quality evaluation. The standard deviation of the acceleration within the time interval provides additional information on the traffic conditions. The individual metric is calculated by

$$G_{\text{ind}} = \frac{\frac{\sigma_a}{\sigma_{a,\text{ref}}} + (1 - \frac{\bar{v}_{\text{ego}}}{v_{\text{ref}}})}{2} \quad , \tag{3.11}$$

where σ_a is the acceleration standard deviation of the CAV and $\bar{v}_{\text{ego}}$ is the mean velocity of the CAV [8, p. 100]. Figure 3.19 shows the concept of the individual metric.

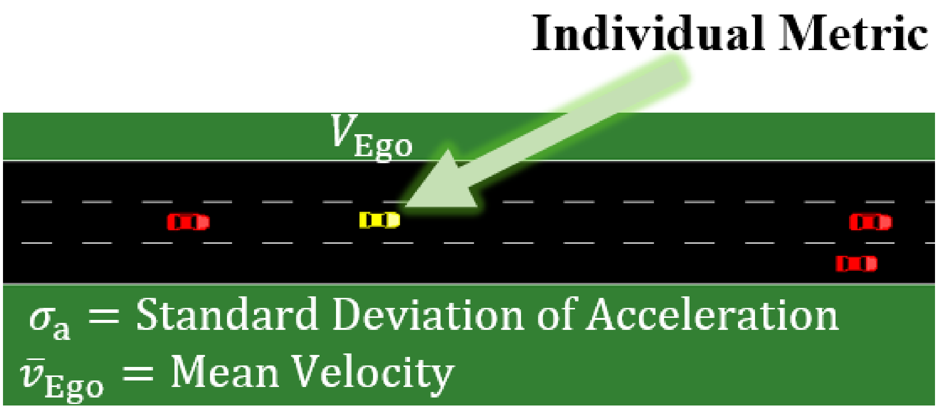

Figure 3.19: Concept of the individual metric [8, p. 101].

Similarly, the reference value is obtained as mentioned in the microscopic traffic analysis. Figure 3.20 illustrates the overall concept including all sub-metrics and their different DOIs.

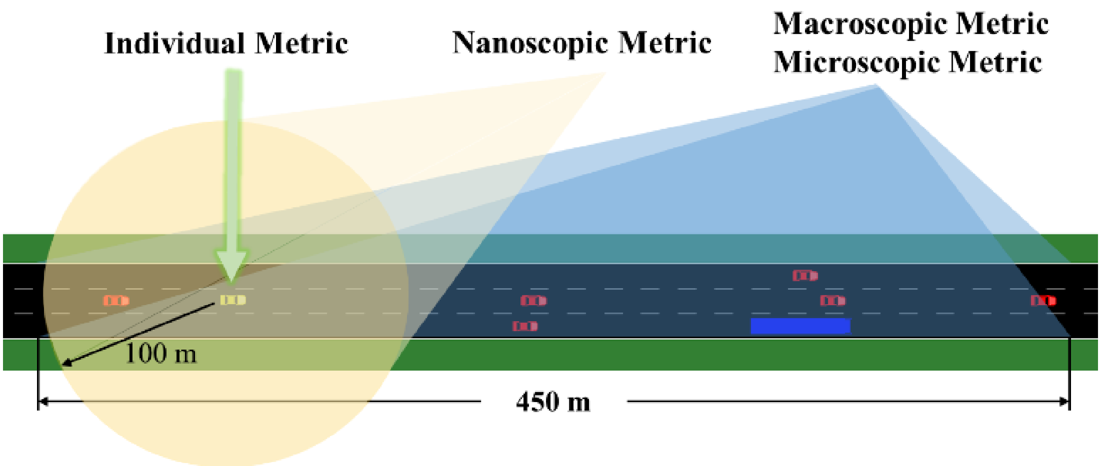

Figure 3.20: Concept of the overall metric combination. Illustration of metric DOIs [8, p. 99].

Using a constant time interval poses the question which DOI and time interval should be used for the traffic quality evaluation, when the CAV travels through different consecutively stacked DOIs, while the time duration within a DOI is not always 15 seconds. In this dissertation the DOI that is assigned depends on the CAV's time duration within the DOI and has to be greater than 5 seconds. Otherwise, there are not enough sample points for the evaluation and calculation of the metrics. This approach is based on observations and assigned empirically.

Time durations of less than 5 seconds can be considered as insufficient due to the lack of sample points. Based on empirical observations the assignment using this time period seems appropriate.

The use of combined metrics demands an overall grading system. The easiest way for determining an overall grade is to calculate the mean of all sub-metrics. This means that each sub-metric is weighted equally as stated by Equation 3.12.

$$\overline{G}_{\text{final}} = \frac{1}{4}(G_{\text{mac}} + G_{\text{mic}} + G_{\text{nan}} + G_{\text{ind}}) \tag{3.12}$$

Since this is not necessarily the best way to combine the sub-metrics, an optimization approach is proposed to determine the sensitivity of the individual metrics. For that reason, a cross-validation [128] is performed. First of all, Equation 3.12 can be rewritten in parameter form:

$$G_{\text{final}} = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 \quad , \tag{3.13}$$

where $x_1, ..., x_4$ are the grades of the individual sub-metrics and $\beta_1, ..., \beta_4$ are the weighting factors, respectively. Equation 3.12 and 3.13 are equivalent, if all weighting factors are chosen equal to 1/4 [8, p. 101]. To determine the sensitivity of each sub-metric, a training and a test data set is generated using the traffic simulation tool. The training data set consists of 826 and the test data set of 498 typical traffic scenarios simulated on a highway. The highway is equipped with an entrance ramp, an exit ramp, and a basic highway element. The highway possesses 3 lanes and the traffic flow is varied to create typical traffic situations. Figure 3.21 shows the highway that is used for the data generation.

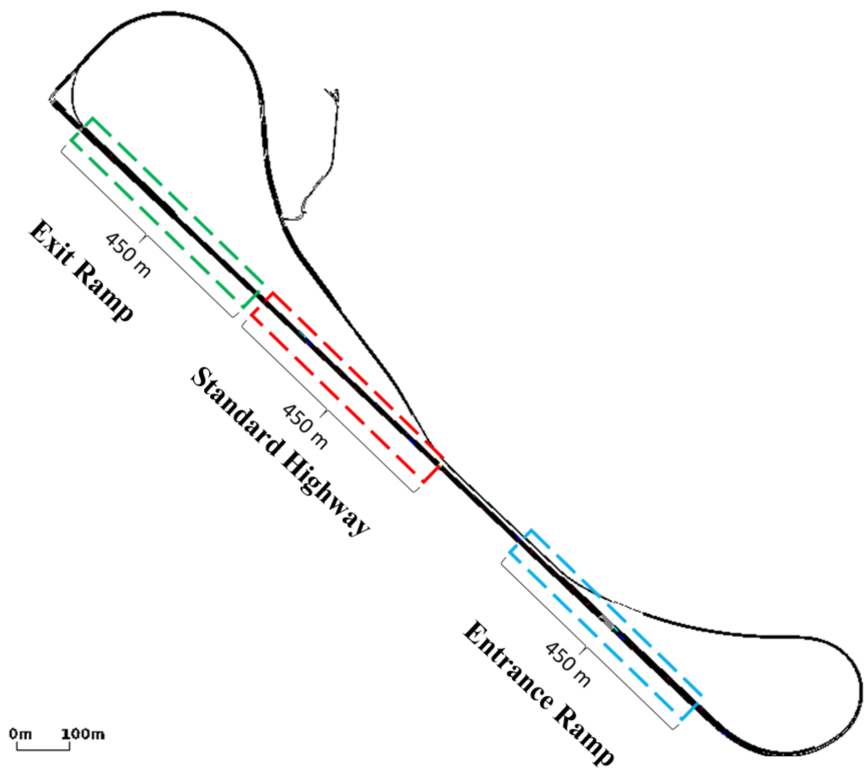

Figure 3.21: Data generation for the cross-validation on the test track in Dudenhofen performed by the traffic simulation tool [125, p. 15].

The simulations are conducted on the Opel test track in Dudenhofen located near Frankfurt. The three DOIs are positioned as shown in Figure 3.21. The traffic simulation tool is used to simulate a CAV traveling through the test track. Thereby, typical scenarios are generated and evaluated by experts. The metric results are calculated and used for the weighting factor optimization shown in Figure 3.22.

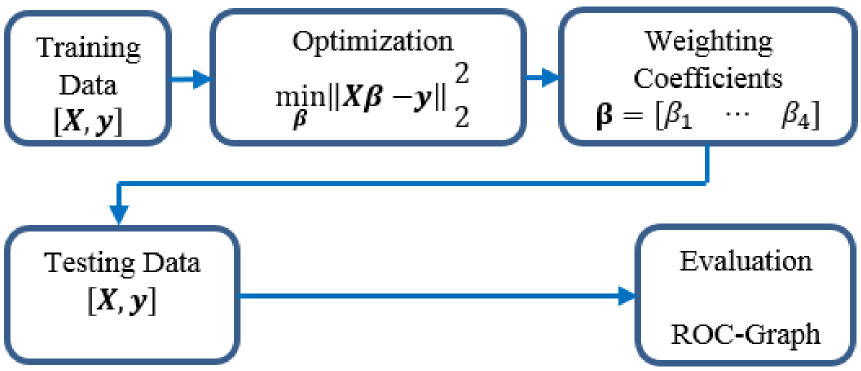

Figure 3.22: Weighting factor optimization and evaluation [8, p. 101].

Basically, the training data is used to determine the weighting parameters and to minimize the difference between the metric results and the evaluation performed by experts. The weighted equation is then tested by the test data. The weighting factor identification is solved by the following optimization problem.

$$\min_{\boldsymbol{\beta}} ||\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}||_2^2$$
$$\text{s.t.} \boldsymbol{b}_\text{l} \leq \boldsymbol{X}\boldsymbol{\beta} \leq \boldsymbol{b}_\text{u}, \sum_{i=1}^{4} \beta_\text{i} = 1, i = 1, \ldots, 4 \quad , \tag{3.14}$$

where $\boldsymbol{X}$ is a matrix containing the metric results, $\boldsymbol{\beta}$ a vector containing the weighting factors and $\boldsymbol{y}$ a vector with the expert opinion based grades. $\boldsymbol{b}_\text{l} = \begin{bmatrix} 0 & \ldots & 0 \end{bmatrix}^\text{T}$ and $\boldsymbol{b}_\text{u} = \begin{bmatrix} 1 & \ldots & 1 \end{bmatrix}^\text{T}$ are the lower and upper bound constraints. The constraints ensure that the values of the optimized individual grades range between 0 and 1 and the sum of all weighting factors equals to 1 [8, p. 101]. The optimization problem is solved by the method of linear least squares [129]. Figure 3.23 shows the results of the identification process based on training data.

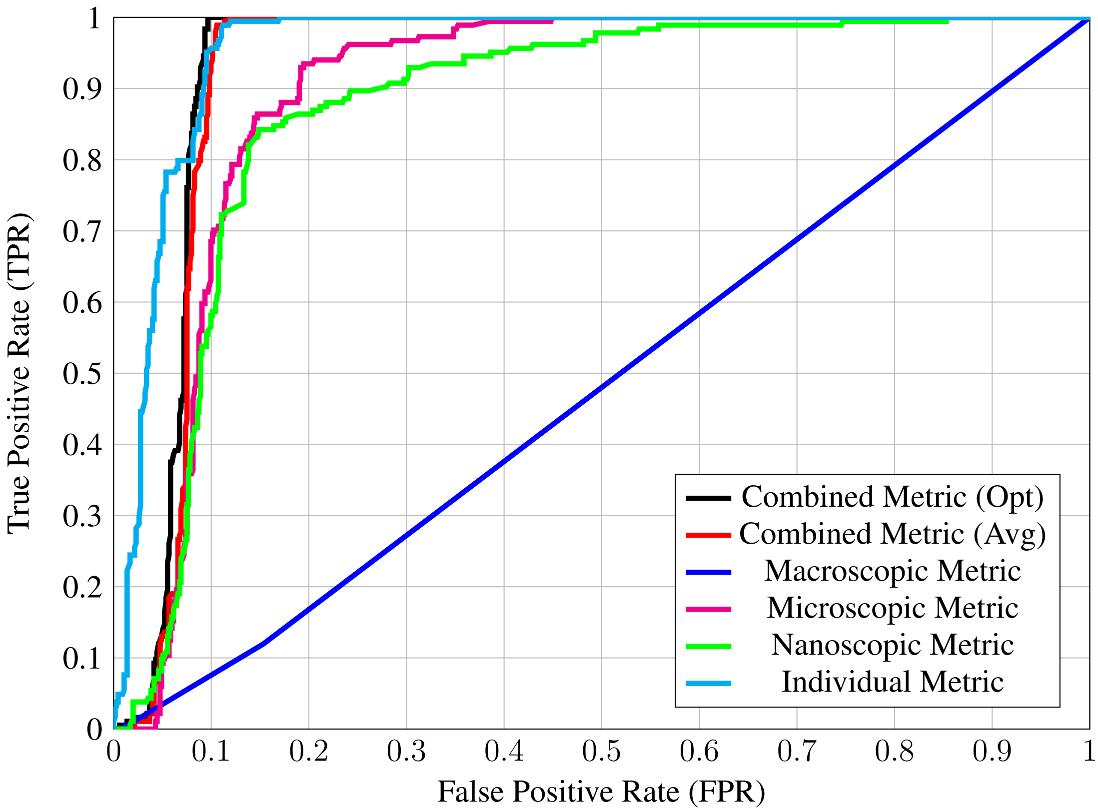

Figure 3.23: Metric performance results (Training Data) [8, p. 102].

The ROC graph shows the relation between the True Positive Rate (TPR) over the FPR of a criticality evaluation. The upper left corner would be ideal, because at this point the TPR would be one while the FPR is zero, i.e., a 100 percent correct critical scenario identification without a single misclassification. Based on the parameter identification and threshold determination the overall grading system is verified by test data. The results of the metrics performance evaluated by the testing data is illustrated in Figure 3.24.

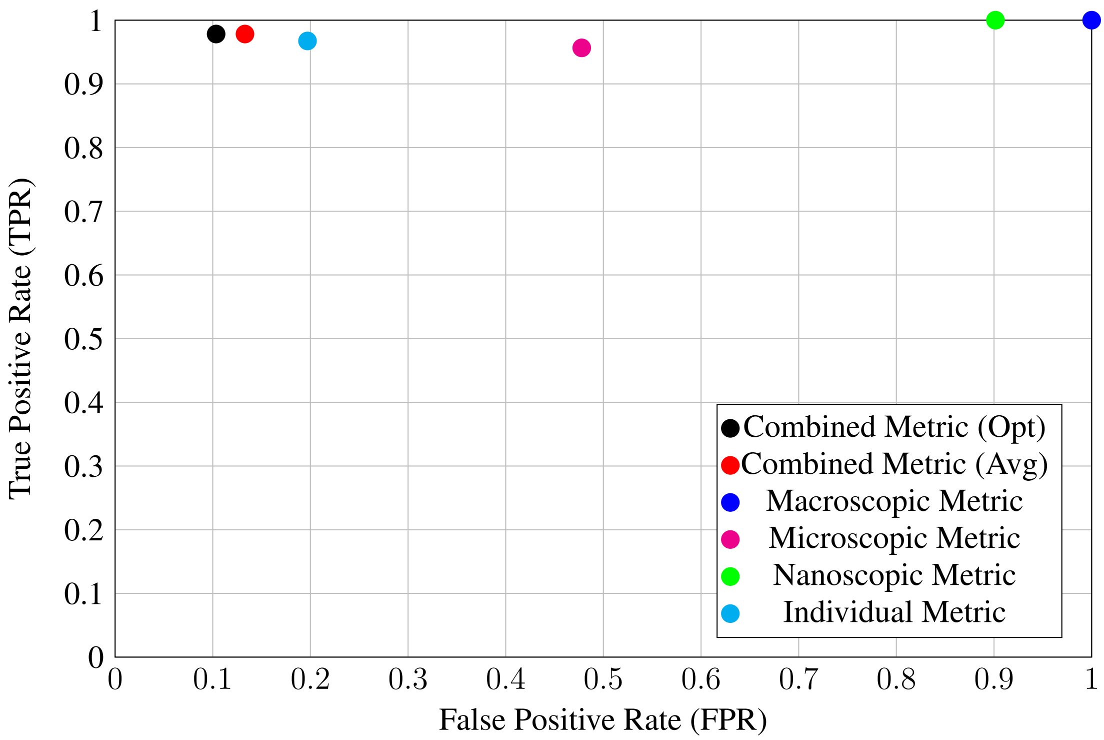

Figure 3.24: Metric performance results (Testing Data) [125, p. 68].

The performance results of the metrics verified by test data results slightly differ from the performance of the identification process based on training data. This mainly results from the independence of the testing data. This data was just constructed for evaluation purposes. Still, the combined and optimized metric show the best results. The TPR slightly decreases to 97.8 percent, while the FPR measures around 10 percent. All of this points to the fact that a cross-validation can improve the metric results. This general idea can be adapted to other more complex metric applications, for example, CAVs driving in urban areas. Furthermore, the metrics parameter optimization could be enhanced with even more training data or it would be possible to train metrics by setting up machine learning algorithms in order to learn how to classify traffic quality. For further details, the reader is kindly referred to the master's thesis of Yiqun Xia [125].

3.1.10 Quality Metrics Ego-Vehicle

The second quality metrics presented in this dissertation are safety related. The usage of these type of metrics aims at evaluating safety critical scenarios. These metrics are usually well-known and used for decades in the automotive industry. Nevertheless, they provide useful insights in the context of this toolchain. Improved safety is one of the most emphasized goals and reasons to develop CAVs in the first place. Reducing accidents is maybe the most beneficial aspect of these systems. Consequently, safety critical scenarios have to be tested intensively. Thus, it is reasonable to apply this kind of metrics within the scope of this toolchain.

The metrics presented here are as before literature-based. The first safety related metric is called Time To Collision (TTC) [25]. This is as already pointed out a popular and often used metric. It decribes the time until a collision would occur, if the velocities of the involved vehicles are kept constant. The TTC can be stated as [130, p. 153]:

$$TTC = \frac{\Delta p}{v_{\text{ego}} - v_{\text{obj}}} = \frac{\Delta p}{v_{\text{rel}}} \quad , \tag{3.15}$$

where Δp is the distance between the two vehicles, v_{ego} and v_{obj} are the ego-vehicle and the object velocity, respectively [8, p. 98]. The difference can also be expressed by the relative velocity indicated by v_{rel}. The reason for using the variable names ego and object is to be consistent with the notations in the literature.

The second metric is called Time To Brake (TTB) and gives the time the ego-vehicle has left to brake until a collision with 0 m/s is unavoidable, given its maximum deceleration [131]. This indicator allows for the use of another metric for evaluation purposes. TTB can be stated as [130, p. 153]:

$$TTB = \frac{\Delta p + \frac{v_{\text{rel}}^2}{2a_{\text{ego,max}}}}{v_{\text{rel}}} = TTC + \frac{v_{\text{rel}}}{2a_{\text{ego,max}}} \quad , \tag{3.16}$$

where Δp is again the distance between the vehicles, v_{rel} is the relative velocity and $a_{\text{ego,max}}$ is the maximum deceleration [8, p. 99].

The third safety related metric indicates the required deceleration that is needed to generate a collision with 0 m/s [132]. This metric is defined by [130, p. 153]:

$$a_{\text{req}} = a_{\text{obj}} - \frac{v_{\text{rel}}^2}{2\Delta p} \quad , \tag{3.17}$$

where Δp and v_{rel} are the distance and the relative velocity between the vehicles, respectively, and a_{obj} is the current object acceleration [8, p. 99].

These three indicators are now used to assess the criticality of the simulated scenario. Again, the usage of more than one metric aims at gathering more information on the evaluated scenario. A metrics combination as described before is deliberately not carried out for the following reason: Safety, in comparison to traffic quality, is an aspect that has to be looked at more carefully. If one of the safety metrics deflects, it is absolutely necessary to take a closer look. There is no room for errors regarding safety related issues. If a CAV is not able to drive safely, the trust in these systems would decrease significantly. So in this regard, the evaluation has to be conservative and observed very strictly.

Another aspect is that these metrics provide a guideline on which tests can be performed by actual driving tests. In simulation, a safety critical maneuver can be performed without causing damage to components or harm to test drivers. So, the evaluation can be considered as a criticality indicator for driving tests as well. If a collision occurs in the simulation, consequently, a driving test should be avoided. Of course, it is possible to perform driving tests by mixed reality approaches, where the other test participants are virtual. For that reason, using mixed reality approaches for safety critical tests is elaborated on in Section 3.2. In addition, using these metrics can be considered as a way for finding corner cases and constraints of the CAV. The evaluation of this aspect is important to gain insights about the general performance and limitations of the CAV. Even if the critical scenario is caused by another test participant, it is essential to know how the CAV is going to react. This means that the CAV has to perform adequately even if another test participant causes a dangerous situation. Adequately means in this case that the CAV should not behave in a way that the criticality increases.

Regarding the toolchain, the metrics elaborated in this dissertation are used for the evaluation of the simulated scenarios. This concept applies for the verification as well as for the identification process. By formalizing the metrics the verification and identification can be performed automatically. No human intervention is necessary at this point. The metrics are implemented and can be calculated automatically for each simulation run. This proceeding constantly calculates the criticality at every point of the simulation and if a metric deflects the simulation run is stored into a database. That way, the function developers can access the simulation data, search for the error sources, and improve the implemented functions. Due to the generic setup of the toolchain, it is possible to exchange the metrics for other evaluation purposes. Basically, the toolchain is not dependent on a specific evaluation method or requirement, providing the benefit that a vast majority of evaluation metrics can be included in this methodology. Especially, non-functional requirements need to be tested after assuring safety. A comprehensive survey on non-functional requirements for CAVs can be found at [26].

3.1.11 Verification/Identification of Concrete Scenarios

All the aspects described up to this point lead to the systematical verification and identification of critical scenarios for CAVs. This step contains the data preparation either for the next aspect, which consists of another test execution in different test environments, or rather for storing the simulation data in a database. Thus, by walking through the toolchain, a digital scenario catalog is generated that contains verified and identified critical scenarios. The simulation data including the metric results containing a criticality classification is stored in a database. Afterwards, the function developers are able to reuse the simulation data to improve the CAV driving functions. By re-running a simulation and observing the behavior of the CAV and the corresponding metrics, the function developer is able to find errors and weaknesses of the implemented function.

If the approach is chosen to route the CAV through a road network while populating the traffic system with a certain amount of test participants, the metrics can be used to identify a critical scenario. If a critical scenario is detected, the simulation data around this time can be stored in a database. By taking a defined time before and after the criticality threshold is violated, the simulation data is extracted and added to the virtual scenario catalog containing already verified and newly identified scenarios. In essence, the CAV is navigated through a populated traffic network for a long time and every critical scenario that is detected can be saved and replayed by the function developers.

Another advantage is the possibility to retain the maturity of the function under test. Depending on how many scenarios included in the scenario catalog are classified as critical, the function developer is able to assess on how well-engineered the driving function is at this point. In general, it does not seem to make sense to increase the validity of the test environment if a lot of tests failed in simulation. Normally, if the tested function does not work in the simulation, it is very unlikely that this function would produce better results during driving tests. On the contrary, functions that fail in simulation could even endanger prototypes and test drivers in a real driving test.

Additionally, the storage of the used simulation models and parameters, especially the configuration of traffic agents' is important. As a developer it is crucial to know if the critical scenario occurred in a very populated traffic system and furthermore, the information on how aggressive or defensive the agents parametrization is. Usually, in the beginning of the development even ordinary conditions, e.g., low populated traffic systems and normative driving behavior of agents can cause critical scenarios. It is not necessarily constructive to test the first implementations in a highly difficult parameterized environment. If disturbances are added, the construction of the disturbance influencing the CAV is also stored in the database providing an indication of the origin of the critical behavior. Furthermore, a list of used simulation models provides necessary information for the function development.

For example, it is not productive to search for errors caused by sensor imperfections, if an ideal sensor setup was used in the simulation run.

The increase of validity provided in the next step of the toolchain raises the question on how to preprocess the simulation data accordingly. The answer depends on one hand on the abilities and constraints of the test environment. This part of the toolchain generally needs human intervention, unless all test environments are known and a strategy can be developed to do this automatically. Based on the assumption that the test environments available in a company are known, the preprocessing can be performed by defining interfaces for XiL methods, test benches, and driving tests. Of course, constraints have to be taken into account. The dismantling of components is not always feasible. For example, a CAV sensor setup is difficult to mount on a test bench, especially considering, that the surrounding environment a CAV has to drive in is very complex. The reflections of the real world, e.g., urban areas or highways are complicated to replicate on a test bench. Nevertheless, if the availability of test environments is known the toolchain provides the possibility to use these in the next step.

In summary, this step contains the database that is passed on to the function development department. Notably, it is vital that the function developers are able to access this data and replay the verified and identified scenarios including detailed information on parametrization, disturbances, used models, etc. In practice, the data memory should be kept as small as necessary due to the fact, that virtual scenario catalogs contain a vast amount of scenarios. Especially, when the developed function is in an early development stage. If the scenarios are also used for validation, the number increases further. For this reason, the introduction of a minimal data set should be considered. A minimal data set that just stores the absolutely necessary information possesses another advantage. This data set definition can be used for different platforms. The data conversion for each test environment can be set up based on this data set and is the foundation for exchanging data between different platforms. Of course a lot of effort has to be spent on the requirements for this data set. It is a challenging task, considering all test environments with their specific demands and interfaces, but this work is very beneficial, especially because the stored data possesses a lot of value. For example, the data can be used for training of machine learning algorithms or statistical analyses as well.

All in all, the toolchain's data storage should be well-arranged and easy to access. Otherwise, the acceptance and usability of this methodology decreases significantly. Function developers should be able to use this toolchain without any barriers. In general the usability has to be intuitive and easy to handle. Regarding future work, it is thinkable that additional recommendations can be included that provide a suggestion on why the developed function caused critical behavior. The used parameters, disturbances, function under test, and metrics can be used as a foundation for suggestions of that manner. This would ease the search for errors and weaknesses for the function development department on a large scale.

3.1.12 XiL Methods

This aspect of the toolchain provides the possibility to increase the validity and plausibility of the simulation results. Whether a MiL result should be enhanced by this step depends on the test requirements. In general XiL methods have certain advantages and disadvantages that are mainly affected by two aspects: The level of detail and the test effort. Figure 3.25 shows the relation between these aspects.

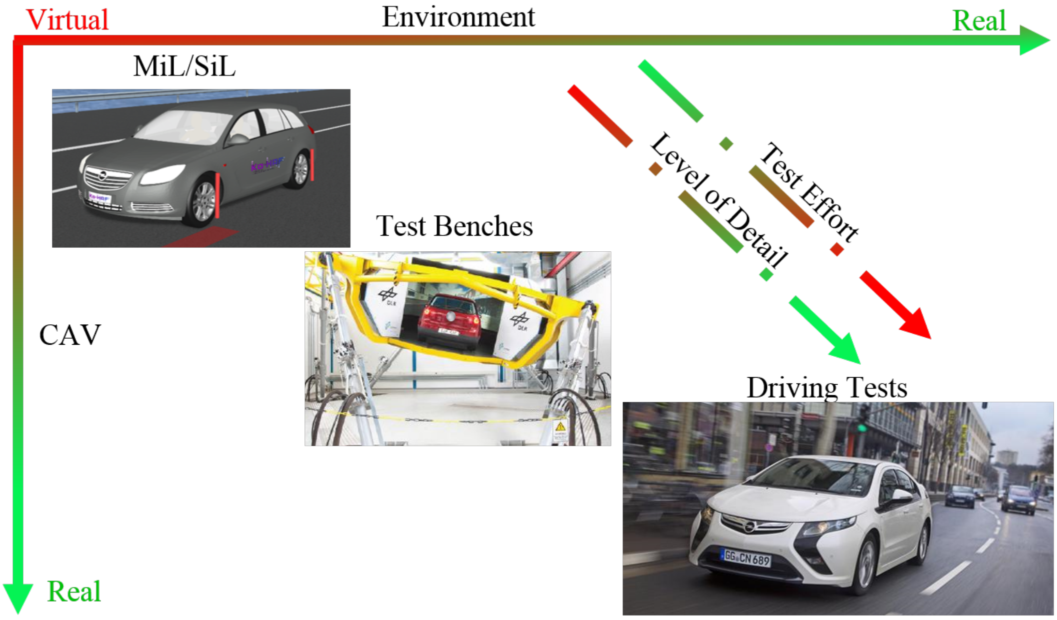

Figure 3.25: XiL methods considering virtual and real test environments, based on [2, p. 460].

If tests are performed solely virtually, the level of detail and the test effort are low. By increasing the validity through test benches and driving tests both aspects increase significantly. When reviewing this correlation, a conflict of objectives arises. On one hand, the test should be valid, on the other hand, the test effort poses a constraint.

There are some important factors that have to be taken into consideration. In early development stages it is not necessary or even not feasible to perform driving tests, because the actual vehicle and components are absent. Additionally, when taking first steps to implement a CAV function, MiL approaches are the right way to go. For more mature stages of the development, MiL approaches can be enhanced by more valid test environments. Furthermore, the toolchain facilitates the use of a step that increases the validity of tests. Even more advanced methods, such as ViL, mixed reality approaches, and soft-target test robots can be applied. ViL is widely used for human factor studies.

The main idea is to drive in a test vehicle on a proving ground while looking through virtual reality glasses. Virtual objects and test participants can be added and the influences on the human driver are investigated. Mixed reality approaches advance this idea by combining driving simulators with actual CAV prototypes, virtual and real world test participants, etc. Even traffic lights, pedestrians, and animals can be included by this approach. All in all, these are useful tools for increasing the validity of the performed tests and should be added depending on the requirements that are defined.

3.1.13 Recorded Data

Recorded data poses an important part to identify parameters for the toolchain. The applications for recorded data are manifold. To begin with, recorded data is used for the model parameter identification of the simulation. Independent of the model in scope, for example, vehicle dynamics, virtual sensors, digital roads, etc., the parameters can be identified by recorded data. Fortunately, a majority of the parameters are already determined by function developers. It is a standard approach that parameters are identified, if a control concept is developed. In that case, it is worth to closely work together with function developers working on different applications. The recorded data is gathered and used for the determination of model parameters for the simulation tools. Additionally, recorded data can be utilized for test benches. For example, video data from manual driving tests can be used to penetrate a camera test bench. This video data is also a valuable source for understanding how disturbances influence the sensors of the CAV. The characteristics of these disturbances can be injected and modeled in the simulation environment as well.

Recorded data is also used for so-called replay to simulation approaches [66]. This aspect features a feedback loop for the toolchain. Available data provides the foundation for concrete scenarios for the coupled simulation environment. This means that the data recorded while driving on public roads is directly converted and replayed in simulation later on, after the test drives are concluded. The main question, which data should be replayed, can be answered by metrics. If a criticality metric, for example, deflects, the data should be stored and replayed later on. Replaying the critical scenario by simulation provides then again the possibility of an in-depth investigation.

Even the correction of digital maps that are used for the static coupling can be achieved by the use of recorded data. SLAM [119] are methods to localize the CAV's position on the road while creating complementary map data. The map data acquired through these algorithms is used to improve the quality of digital maps and also constructional modifications of roads are updated. Constructional modifications are, for example, new traffic signs, road markings, additional lanes, etc.

Data raised by driving studies provide the convenience to apply statistical analysis on the frequency of verified and identified critical scenarios.

In order to reach a high safety level, the driving function should be able to handle all the critical scenarios that are known. Nevertheless, more frequently occurring critical scenarios have to be tested more intensively, before very rare scenarios have to be taken into account. Driving studies enable to find a distribution of the probability of occurrence of critical scenarios that are experienced more often than others.

A lot of effort is spent on setting up traffic observation systems. For example, there are some test facilities, such as the "DLR-AIM" in Brunswick [108], the "Test Bed Lower Saxony" [109], or the "Test Area Autonomous Driving" in Karlsruhe [110], that aim at gathering data of public traffic systems. The goal of these initiatives is to convert entire cities into research laboratories. Additionally, the areas around the research labs are incorporated, too. This proceeding gives the opportunity to include highways and rural roads. In the case of DLR-AIM, inner city traffic observation and control systems are installed, for example, on a crossroad. In this way, the crossroad data is gathered and intelligent control approaches can be tested. A large part of the surrounding highways is observed to acquire realistic traffic data.

Limitations for the usage of recorded data are usually the disparity in quality. At this particular time, there is no standard for collected data. This means that in many cases the accuracy is not specified and required state space variables are not always available. This happens due to the reason that the operating companies specify the requirements of their data set by themselves. Nowadays, research initiatives, such as PEGASUS [55], try to define a minimal data set. As already mentioned before, this data set contains a minimum representation of state space variables to enable interfaces between different test environments. In the future, the providers of measurement data should reach an agreement on how to standardize recorded data. Thus, every manufacturer, supplier, etc., would be able to access data sets in the same way. Anyhow, every data set used for this toolchain the specific requirements, measurement accuracy, availability of state space variables, etc., has to be checked as well as closely observed and evaluated. Otherwise, the usage of the recorded data could be a source of failure and worsen the results of the simulation environment. For example, if a data source is faulty, the utilization for identifying model parameters decreases the quality of simulation results.

Regarding the toolchain, all of these data sources can be used to compare and improve the plausibility of performed simulations. This increases the validity of the entire approach and can be considered a useful enhancement for the whole methodology. Unfortunately, the data sets are not standardized yet, but in the near future it is probable that a unified specification is going to be developed. In that case, the toolchain's interfaces can be easily adapted and the effort spent on accuracy and availability can be omitted. Within the framework of this toolchain the created data gained by simulation, driving tests, etc., are defined in a compatible fashion. Hence, the data created by this approach should be exchangeable between the different test environments.

3.1.14 Driving Tests

A driving test usually contains an actual real-world prototype of a CAV driving on a test track with other test vehicles driven by human test drivers or in a public traffic system. Even though driving tests can be considered as a valid test method, there are some setbacks. As already indicated, driving tests are expensive and require a lot of time and effort.

First of all, it is not a given to find a so-called ground truth measurement system. On public roads, the ground truth is usually not achievable, except in some rare cases where a whole traffic system is observed by a measurement system [108, 110]. The gained data of the driven distance on public roads relies on the CAV's own sensor setup and is imperfect. Particularly, the other test participants are hard to track correctly. The same issue arises for test tracks as well. A ground truth measurement system is usually not available and has to be installed. Manufacturers aim at equipping test tracks with these systems nowadays. If an independent measurement system is in place, the obtained data is useful for comparison with other test environments.

Another challenge that has to be faced is the reproducibility of tests driven by human test drivers. There is a lot of endeavor to develop test tools to instruct test drivers on how to drive certain tests exactly as planned [74, 75]. But in reality it is difficult to repeat a test driven by test drivers without fluctuations. Meanwhile, some solutions, such as automated test robots are available. The potential danger test drivers are exposed to can be far more problematic. A malfunction causing an unexpected behavior of the CAV can lead to accidents and therefore, harm to test drivers and components. Test robots equipped with soft targets are considered a helpful solution for that issue. These robots can be overrun and driven against without destroying the CAV. Test drivers are not required, because the robots are equipped with a driving function itself, which reduces the risk for humans and ensures reproducibility as well [27].

Driving tests on public roads are used to acquire driving data. Of course, the CAV has to ensure a certain maturity, before these tests are performed. Especially in environments, where interactions with human drivers are wanted and necessary, safety is important to protect, for example, human drivers, pedestrians, etc. Normally, only specially trained test drivers are allowed to perform tests on public roads and in addition, a permit has to be obtained from governmental administrations. Then again, the ground truth and reproducibility lack in this test environment. Only the CAV sensor setup can be used for data acquisition. However, the data gathered while driving on public roads is important due to the following reasons. First of all, the tests performed are intrinsically true. This means that this is the domain the CAV is going to operate in. All developments and tests should aim for this application. Secondary, unexpected behavior of other test participants can be observed and analyzed. Even risky behavior or security problems can be studied in this test environment.

3.1.15 Analyses

The last aspect of the toolchain is the analyses and transfer to the function development department. While using this toolchain, the results, properties, and parameters need to be clearly arranged. Therefore, a test result sheet is produced and contains significant information. Further, links to the data sets generated while going through the toolchain are provided. Figure 3.26 shows a generic test result sheet. This is an example how a test sheet could look like. The content can be adapted with respect to the evaluation goals.

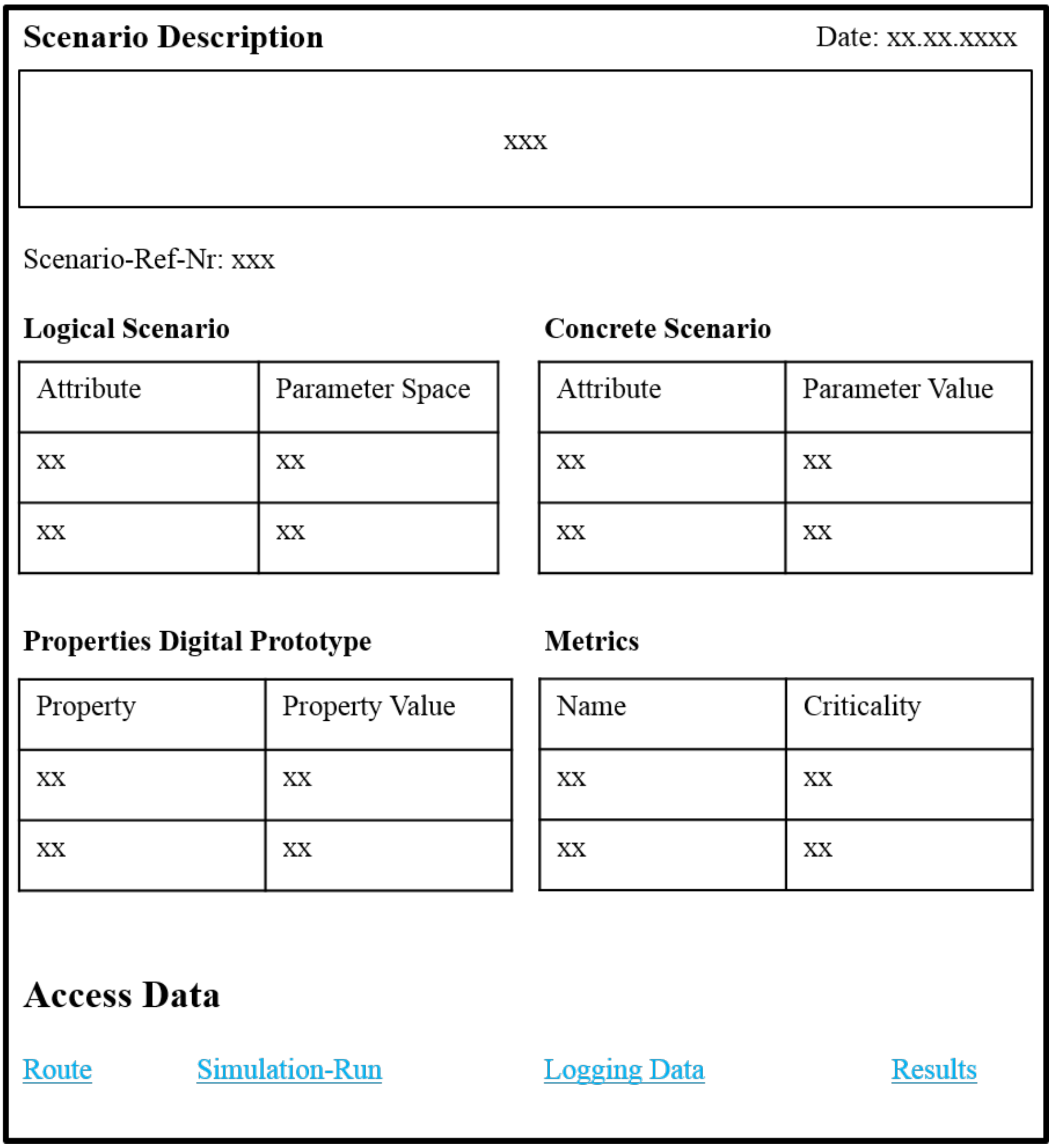

Figure 3.26: Template for a test result sheet.

Starting with a semantic scenario description, the test result sheet explains what has been simulated and tested. This provides helpful information for a developer. Additionally, a date and a time stamp when the test was performed can be added. To synchronize the scenarios with the existing virtual scenario catalog the performed test or scenario is referenced by a scenario reference number. This helps to separate and organize the scenarios, for example, in a database.

Important aspects, such as the attributes and parameter spaces of the logical scenario (see Sec. 3.1.1), can be obtained directly from the test result sheet. Of course, the determined parameters resulting in a concrete scenario are provided as well. Thus, function developers can quickly and easily look at this information. The next table on the test result sheet contains the properties of the digital prototype. This is particularly crucial, because this table provides information on which simulation models, software version, etc., was used. This data can be synchronized with software repositories. For that reason, the software development department is enabled to keep track on which software version was tested and in which environment the test was performed. It is possible to provide information, if the test was conducted by a MiL, SiL, HiL approach or even by a driving test.

The last table contains the evaluation metrics used for the methodology. The metrics used in the toolchain are exchangeable and for that reason, it is important to know which metrics were used and how the metrics deflected. The scenario is usually only added, if a metric classified the scenario as critical. The toolchain provides the possibility to use different metrics, for example, safety, traffic quality and others, simultaneously. The test result sheet allows for reviewing metric results and why the scenario was classified as critical. It is worth mentioning that only the metrics concerning the function developer's focus are shown. On the other hand, every elaborated metric can be calculated and stored in a database. In this way, the function developers can perform a search of, for example, every scenario tested with a particular software version that was classified as critical regarding safety issues.

To conclude the test result sheet several links are provided for accessing data directly. First of all, the driven route can be observed. The routing is shown on a digital map. The next link leads to the simulation run. The function developer is able to replay the scenario in the MiL environment. Re-running the simulation aims at investigating of the critical scenario that occurred. Thus, it is possible to observe what happened and it provides the possibility to improve the driving function afterwards. If changes in the implementation have been made, the simulation run can be started again to show advancements. Basically, the developer is able to review if the changes in the driving function shows a better result than before. The third link guides to logging data. Sometimes, only the logging data is investigated to improve driving functions. It is a choice the function developer has to make. At last, an overview of the results can be given for determining which metric deflected at a certain point in time. This concludes the walk-trough the entire toolchain.

Finally, remarks considering the proposed toolchain‘s compatibility with existing testing frameworks and the application in practice are given. First of all, the newly developed approach is already used by Opel beyond the direct context of this thesis. Projects to industrialize the methodology and incorporate major aspects into the PSA simulation framework AXIOM are underway [133]. The toolchain is to be applied for advanced virtual development and testing of ADAS systems as well as for CAVs. Further activities investigating the individual aspects of the toolchain more detailed are launched preparing the methodology for development and validation [134].

The concept is also picked up by funded research projects that focus on testing of CAVs with higher SAE levels 4 and 5 in complex ODDs, such as urban areas. The method can be incorporated in the project Verification & Validation Methods and the related simulation project Set Level 4 to 5.

All in all, the proposed methodology provides an advancement in the field of CAV testing that is beneficial and applicable in practice. The fact that this concept is already industrialized and used for CAV and ADAS testing as well as utilized in further research activities, can be considered as a confirmation that this approach is going to play a role in CAV testing frameworks.

3.2 Novel Prototype-in-the-Loop Approach

In this section, a novel approach to combine a real-world CAV prototype driving on a proving ground with the already used traffic simulation is presented. To stay connected to the toolchain this methodology could be used in the validation step or substitute the vehicle dynamics simulation in the simulation environment. In this dissertation, the focus is on the substitution of the vehicle dynamics simulation. Therefore, the aspect simulation environment looks different, while the rest of the toolchain remains as is. Figure 3.27 shows the modified simulation environment (see Fig. 3.10) extended by the real-world CAV prototype. This approach is called Prototype-in-the-Loop (PiL) derived by the inclusion of a real-world CAV prototype in the simulation environment.

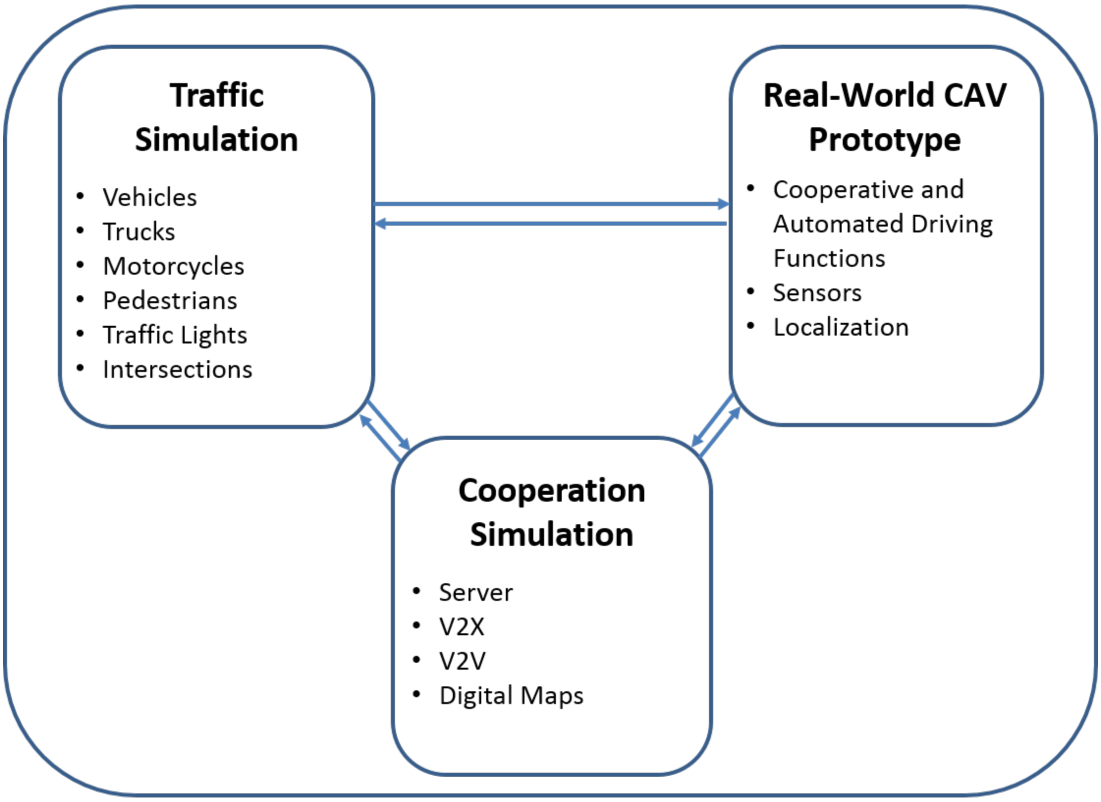

Figure 3.27: Toolchain's simulation environment extended by the Prototype-in-the-Loop approach.

Now, the walk through the toolchain can be performed the same way as before. Only the vehicle dynamics simulation is replaced (see Fig. 3.1). Again, in this dissertation the focus lies on the interaction of CAV prototypes with the traffic simulation. Thus, the cooperation simulation is not elaborated on at this point. Nevertheless, the toolchain can be further enhanced by including a cooperation simulation using this approach as well. Cooperative features can be included either by simulation or it would even be possible to elaborate on these features within the real-world CAV prototype. Some cooperative features, such as precision map updates are included in the CAV prototypes anyway and therefore, do not need to be simulated. Figure 3.28 shows a snapshot of a test performed with the PiL approach.

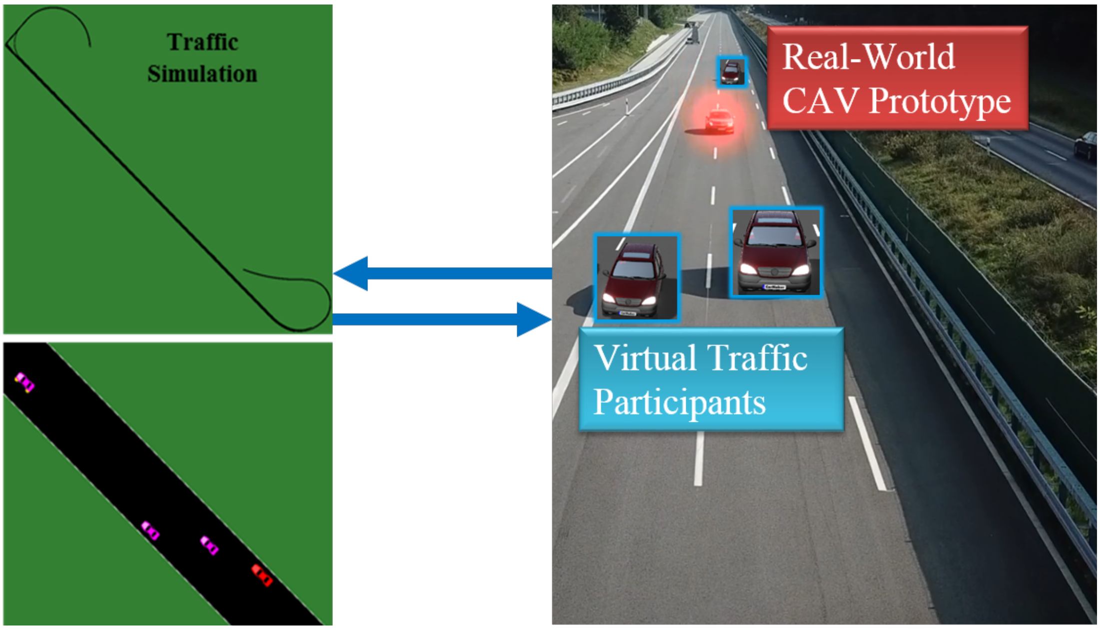

Figure 3.28: Prototype-in-the-Loop snapshot on the Opel proving ground. Real-world CAV prototype colored in red. Virtual traffic participants marked by blue boxes [135, p. 8].

The application of the PiL approach starts by using the localization implemented in the CAV to position the prototype in the traffic simulation. In this way, the traffic simulation possesses the CAV prototype's state space variables and is similar to the MiL approach able to control the other test participants accordingly. The behavior of the test participants needs to be communicated to the CAV prototype. Therefore, the dynamical properties of the object list, which usually originate from an environmental modeling approach relying on the sensor setup, is overwritten. Thus, the dynamic attributes can be controlled by the traffic simulation. Attributes, such as lane marking detection, traffic sign detection, etc., are not overwritten. Only the dynamic objects should be virtual due to the fact that the remaining CAV functions should be as realistic as possible. Figure 3.29 shows the PiL approach.

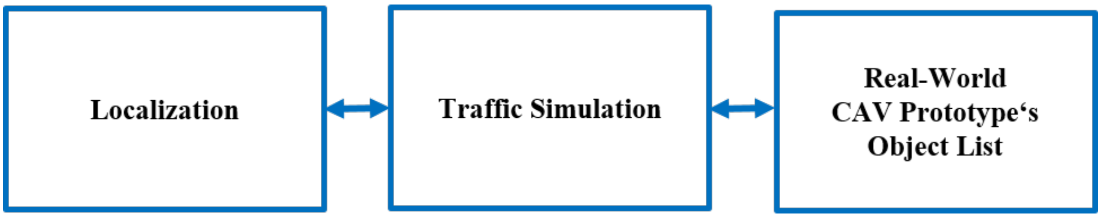

Figure 3.29: Prototype-in-the-Loop application.

When applying this method, there are some issues that need to be taken into account. First of all, the implemented code has to perform in real time and needs to be synchronized with the time of the localization and the function development. Fortunately, by just changing a part of the object list, the basic message remains the same. Thus, the motion prediction and calculation of the environment model between two messages is performed by the function development themselves. The only major difference is that the dynamical attributes are substituted by the test participants calculated in the traffic simulation tool.

In general, the implemented software has to be fast enough to cause only a small delay between receiving a message, changing the dynamical attributes, and forward the message to the driving functions. The traffic simulation is controlled via the interface TraCI. TraCI provides the possibility to control the traffic simulation SUMO-Simulation of Urban Mobility by external software. There are a vast amount of functions available. In this case, it is important to change the CAV’s state space variables, read the test participant‘s state space variables, and perform simulation steps when required with respect to the time synchronization. The scenario modeling can be performed as already mentioned in the toolchain. The scenario generation in the CAV prototype is generated with the traffic simulation tool. SUMO enables to control the trajectory of test participants in a predefined manner or the test participants act as agents driving according to driver behavior models.

Logging and preparing data is more expensive than in a pure MiL approach. The data in the CAV prototype is often distributed over many different systems. Therefore, the logging concept requires effort on where the data can be obtained from which source, sample time, and accuracy. In special cases where the data is stored in an asynchronous manner, the data has to be synchronized afterwards with respect to the message time stamps. On the other hand, even if the data is time synchronized it is sometimes necessary to interpolate between sampling points to obtain a differentiable result. A suited method for this is the usage of splines. Splines are differentiable even at the sampling points itself and therefore, differentiated variables, e.g., velocity, can be calculated at the logged position values. Another advantage of interpolating the asynchronous data is that the discretization can be done afterwards with the required sampling rate best suited for the individual purpose.

Before examining the potential of this approach a more detailed overview on the setup is given. Considering a CAV prototype, there are a lot of modules necessary to let the vehicle drive by itself. Briefly speaking, the CAV operates by the robotic paradigm: Sense, Plan, Act [136, p. 189]. The sensing part is done by different sensors the CAV is equipped with. These sensor signals are consolidated and processed to an environmental model as well as a localized position. Based on that information and prediction algorithms the second term plan becomes relevant. The planning part contains the driving strategy and trajectory planning. The difference between these aspects can be separated as follows: A driving strategy possesses information, such as desired velocity, when the vehicle should overtake, etc. The trajectory planner is able to implement the driving strategy with respect to physical constrains, comfort, safety-related issues and so on. The last part, act, contains the trajectory following controller that guides the CAV. Figure 3.30 shows the Opel Insignia prototype enhanced with the PiL approach in detail.

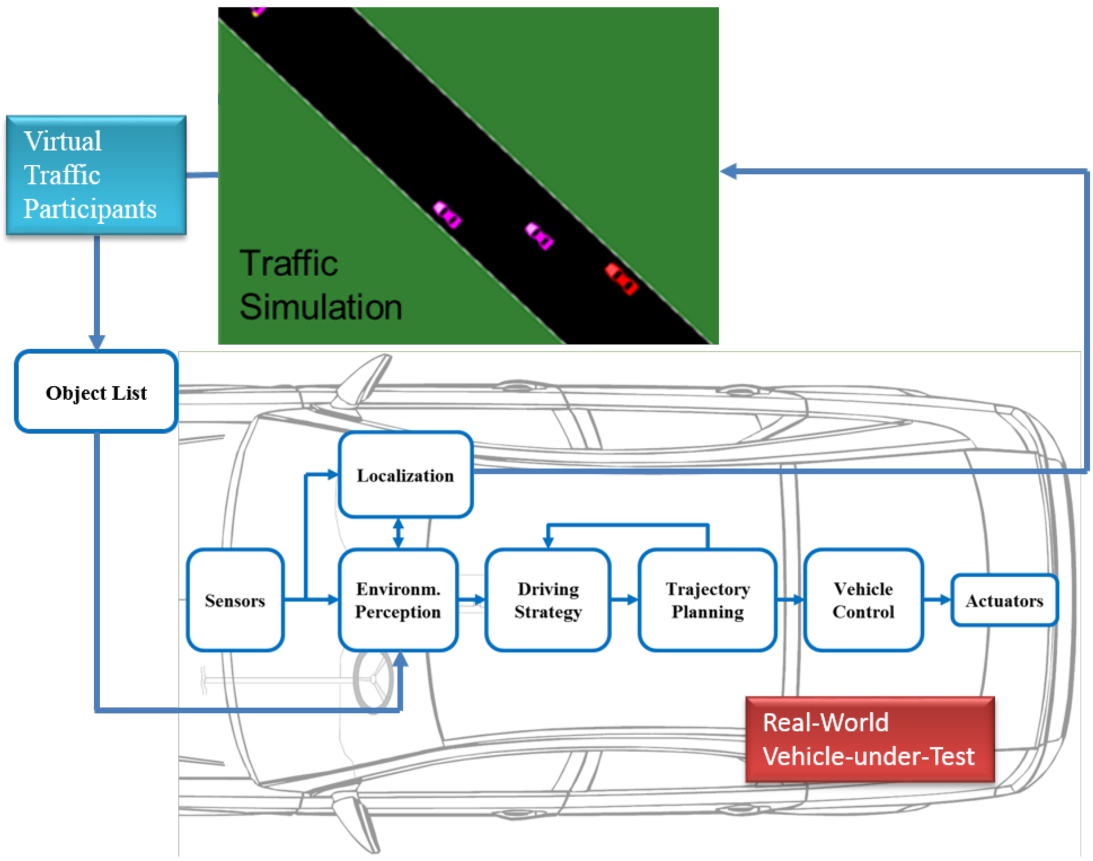

Figure 3.30: Prototype-in-the-Loop method in collaboration with the CAV prototype [135, p. 7].

As mentioned before, the localization provides the CAV's state space variables that are stored in the traffic simulation tool.

After calculating a simulation step, the state space variables of the test participants are replaced in the object list of the CAV. The CAV uses this information in the environment model and executes the "Sense, Plan, Act" scheme before the next localization data is sent to the traffic simulation. Afterwards, this scheme is continuously repeated. By doing so, the real-world CAV drives on a real proving ground using the implemented functions on the installed hardware, but reacting to virtual test participants.

To keep track of what is actually happening while the test is performed, the visualization of the traffic simulation tool can be used. If a human machine interface exists that shows other test participants, the overview is further improved. Otherwise, it is difficult to monitor what the CAV is doing and whether those actions are plausible. In any case, even if the other test participants are virtual, safety needs to be guaranteed. There are other hard targets on a proving ground than test participants only. Experienced test drivers are absolutely necessary while performing these kind of tests. For example, a performed test can cause the CAV's driving function to fail in some way. If that happens, the test driver does not know how the CAV is going to react. So, the test driver needs to handle this tool carefully. It is thinkable, that an extension of this method could be an enhancement of virtual reality glasses to investigate human factor aspects. The behavior of drivers could be studied while the CAV is actually driving, but the dynamic surroundings are virtual. This would be an addition to the existing ViL approach, controlling the other test participants by driver behavior models instead of predefined trajectories. Virtual reality glasses could also be used for test drivers to get a more intuitive understanding of the current test situation. This enhancement would provide many benefits for the usability of this methodology.

In general, one of the major advantages of this approach is the possibility to perform critical scenarios without causing damage either to the CAV prototype or the test driver. Scenarios, such as risky cut ins or strong braking maneuvers of the vehicle driving in front of the CAV, can be tested. Even if the driving function causes a collision with the other test participants, no damage or harm occurs. Furthermore, the CAV testing can be done at its physical and functional limit. The behavior at these limits is an important part of investigating the CAV's behavior. Before, these so-called corner cases could only be tested with pure simulation approaches and due to the importance, these tests should be as accurate as possible. Mathematical models possess uncertainties that can make a huge difference in corner cases. Thus, these tests are obligated to be performed with a real-world CAV.

Another aspect that can be tested with this methodology are requirements regarding other test participants. For example, if a CAV enters the highway, investigating the impact on other involved vehicles becomes feasible. Do other participants have to perform strong braking maneuvers because of the CAV‘s behavior? Is the CAV able to handle different types of human drivers? How does the CAV impact traffic quality? Even misuse cases, caused by other participants, can be evaluated.

In addition, due to the possibility of including a huge amount of other test participants provided by the traffic simulation, long term impacts on traffic systems can be studied. This helps to test even large traffic jams for traffic jam chauffeur [23, pp. 52-53] evaluation. It is almost impossible to reproduce and choreograph a traffic jam on a proving ground with human test drivers. Using the traffic simulation tool, this should not longer be an issue. The tests become reproducible. Reproducibility and ground truth are difficult to achieve on a proving ground. The lack of data available poses a challenge for vehicle manufacturers. The presented PiL approach eases the process to achieve that target. The use of simulation enables to gather important data in a convenient way. Creation of a ground truth largely depends on the localization quality. If the localization implemented in the CAV is accurate, this approach gives an omniscient view on the test quantities. The trajectories, state space variables of each test participant, even in tests with a lot of participants, can be observed and evaluated. No test observation system nor ground truth setup is necessary anymore.

Regarding future work, the comparison between simulation only and proving ground tests can be achieved easily. If the scenarios are reproducible, the difference between MiL and PiL is caused by the discrepancy of mathematical models. A statement on the accuracy of the performed MiL tests is easy to make. Just performing the same test in both environments allows for the investigation of an error model. Properties, such as robustness, probability of error, and accuracy are achieved by just comparing data of the test results of both approaches. This enables a statement on the confidence level of the simulation methods introduced before.

There is one drawback of this methodology as well: the lack of sensors used for perception. By changing the object list the sensors are left out of the equation. Sensor errors and sensor imperfections are excluded and can not be evaluated directly. Therefore, every test should be replayed in simulation with an accurate sensor model. Another way out of this dilemma would be to insert a failure injection into the sensor signals. Anyway, sensor errors and imperfections have to be considered while using the PiL approach.

As one of the next steps, the motion control of test robots through traffic simulation can be obtained. The agents and their driver behavior models can be used as an input for soft target test robots. This enables to create test robots with different driving behavior that move according to the traffic simulation. In this way, the sensor perception can be included into testing. PiL provides a beneficial enhancement to the toolchain's capabilities and testing of CAVs in general, irrespective of whether the method is used for identification, verification, or as a validation step. The possibility to have reproducible and ground truth testing is a huge advantage. The most crucial benefit is the testing of critical scenarios without posing risk to test drivers and prototype. Last but not least, the approach is economically interesting, because performing these tests requires less test drivers and vehicles.

3.3 Contributions

In this section the contributions of this dissertation are elaborated. Based on the afore defined research objectives in Chapter 2, the research advancements, innovations, and benefits of the proposed simulation-based toolchain are stated.

The primary motivation for composing this dissertation was to make a contribution in the challenging field of CAV testing in general. Further, the development of a tool, or rather a toolchain contributing to this challenge, became the focal point of the conducted research. Broadly speaking, the testing methodology for CAVs should be structured, eased, and applicable for driving function developers. Encouraged through the participation in the research project Ko-HAF, the challenges of CAV development accompanying tests arose rather quickly. The buildup of a CAV prototype as well as the task to implement a toolchain for testing of progressing implementations of CAV driving functions compelled the demand for a systematic and, particularly, an applicable approach.

The introduced toolchain provides a systematic methodology for improving simulation-based CAV development and tailored testing for CAV driving function developers as well as for validation engineers. The main focus lies on rectifying development accompanying tests, but the developed approach can also be utilized for validation of CAV driving functions. By automatically assessing and identifying critical scenarios with respect to the current implementation status, the methodology provides helpful guidance for improvement as well as a systematic way to test the driving functions according to the developers demands. The centerpiece of the approach consists of a co-simulation, which includes a vehicle dynamics, a traffic, and a cooperation simulation. By combining the advantages of each simulation tool, the approach developed in this dissertation provides a comprehensive simulation environment. Further, the evaluation or assessment is automatically carried out by metrics, while simultaneously considering the effects of the CAV on its ODD and vice versa. Thus, different DOIs are investigated to enable a concurrent evaluation of the afore mentioned effects. Another aspect that is incorporated into the framework is the injection of disturbances that can originate from either the CAV itself or its ODD.

The first requirement derived from the research objectives in Section 2.4 is to establish a "systematic simulation-based methodology for development accompanying tests of automated driving functions." The proposed toolchain elaborated on in Chapter 3 accomplishes the requirement. The CAV driving function developer is enabled to utilize this toolchain based on a co-simulation environment for systematically testing and improving the implemented functions. The toolchain provides a guideline for testing considering the demands of the developer and assists in the continuous improvement of the current implementation.

The toolchain allows for embedding cooperative features that can also be automatically tested and evaluated. The co-simulation can be enhanced with these features and cooperative aspects can be tested as well.

One of the major assets of the simulation-based toolchain can be considered the generic composition of the entire methodology. Exchangeability of each aspect is key to support the developer demands for testing. The parameters can be set up according to the required test objectives, while the toolchain includes a scenario model that allows for defining parameter spaces and varying the important parameters based on a parameter variation module. In this way, the developer is able to decide which parameters are of major importance and which parameter changes pose a severe challenge. Further, the parameters can be automatically changed for finding parameters that possess severe impact on the driving function under test.

Simulation tool independence needs to be ensured for applying this approach independent of tool providers or available simulation models. The proposed methodology is unattached of the simulation tool the developer wants to apply. This is particularly important for users throughout industry, research, etc. Each user can incorporate the simulation tool of his choosing while operating the toolchain. Even the models applied in the tools can be customized based on the requirements of the developer. This toolchain is able to support different simulation models as well as different digital prototypes, virtual sensors, etc.

In order to deploy more validity to the concept the toolchain makes a usability across different test platforms available. The aspect can be applied, if a test is required to be more valid than solely relying on a simulation run. Therefore, the performed tests can be enhanced by test benches, mixed reality approaches as well as driving tests.

Exchanging CAV driving functions, or rather the digital prototype and its ODD allows for featuring all SAE automation levels. The toolchain is not dependent on a certain automation level, ODD, or implemented function. Tests for more challenging tasks, such as automation level 4 or higher and difficult ODDs, for example, rural roads and even urban areas, can be tested by applying this toolchain.

In order to stay compatible with the state of the art scenario modeling, the toolchain leans on the scenario modeling of the PEGASUS project, which is probably to become a standard for scenario modeling for future applications of CAVs. In this way, the toolchain can be combined with, or support the PEGASUS approach. Thus, additional input can be provided, for example, for the PEGASUS database and many other aspects that are investigated within the project, or for consecutive projects that are assembled based on the findings of this research initiative.

A major benefit of the introduced methodology and eventually the most important asset is the automated identification of critical scenarios. The extension of requirements engineering based on experts can be considered as a tremendous accomplishment that concerns one of the major challenges for CAV testing. However, the complexity that is imposed by CAV development can not be covered single handedly by requirements engineering. It is simply not possible to cover every critical scenario that can occur by only relying on expert knowledge. CAV driving functions themselves and, particularly, the interactions with the ODD consisting of human drivers, pedestrians, infrastructure, etc., are by far too complex, to ensure safety, for example, solely based on requirements engineering. The toolchain provides a systematic simulation-based proceeding to identify critical scenarios. Thereby, the term "critical" can be defined by the developer himself, considering if the test requirements are safety related or the focus lies on some other aspect which is in the current field of interest.

The assessment or rather verification of scenarios that are already classified as critical is also supported by the toolchain. Consider a scenario that is added to the scenario catalog by requirements engineering. This scenario can be verified by the simulation-based approach and the actual criticality with respect to the applied metrics is evaluated. This proceeding provides additional information and a plausibility check, whether the scenario is actually critical or not. Sometimes, scenarios that are intended for a test based on expert knowledge can be revoked because of the insights the toolchain provides. Another advantage of a re-evaluation is the determination, if a test is too dangerous to be performed by driving tests and, therefore, harm to test drivers and hardware can be prevented.

CAV testing requires continuous evaluation, improvement, and documentation of maturity level of the currently implemented software features. The toolchain enables that by generating and saving data while operating. Implementation stages are recorded as well as documented and, therefore, even previous development stages can be restored and accessed. This proceeding addresses the afore stated requirements and is considered a valuable improvement compared to state-of-the-art approaches for CAV testing.

Simultaneous evaluation of CAV effects on ODDs and vice versa, considering different DOIs, is a valuable extension to existing methodologies. CAVs have a direct impact on their ODDs, which consist of human driven vehicles, pedestrians, cyclists, etc. Particularly, the impact of CAVs on traffic systems and quality is investigated and evaluated in this dissertation. Developed traffic quality metrics enable to assess this issue. Simultaneous evaluation for different DOIs provides helpful insight for the assessment of traffic quality influences caused by operating CAVs. Of course, the ODD affects the CAV as well. Further, the impact of ODDs on CAVs is evaluated by this approach, too. Besides traffic quality aspects, the investigation on safety related concerns are addressed by the toolchain. Imperfections of CAV driving functions or disturbances are incorporated into the toolchain. The disturbance sources can be manifold due to the complexity of the CAV driving functions. On the other hand, disturbances can be caused by the ODD as well. For example, human driving behavior is still one of the major challenges when operating CAVs in public traffic systems.

In a way, the CAV and the ODD are influencing each other because of the direct interaction between them. These interactions can be investigated, evaluated, and analyzed by operating the simulation-based toolchain.

While designing the methodology, a key requirement was the applicability of the approach without needing to disrupt the toolchain for different applications and demands. The generic composition allows for exchanging each aspect, while the proceeding and the structure remains unchanged. For example, the simulation tools and models can be exchanged without changing the structure of the toolchain. Parameter spaces, metrics, disturbances, driving functions, etc., can be tailored to the demands of the developer. Depending on the simulation tool availability accessible to the developer an overview of compatible tools that can be used when applying the toolchain are, for example, Apollo Simulation [137], AVL Assisted and Automated Driving [138], Carla-Open-source simulator for autonomous driving research [139], IPG - Carmaker [140], Nvidia Drive Constellation [141], PTV-Vissim [142], DLR-SUMO [51], Tass-Prescan [143], TESIS-Virtual Test Driving [144], Unity [145], and Vires-VTD [146].

Finally, the PiL approach can be considered as an additional effort to enhance existing CAV testing methodologies. In comparison to existing mixed reality approaches, such as ViL, PiL possesses the afore mentioned advantages of using a multi-agent traffic simulation similar to the co-simulation aspect of the toolchain. Substituting the vehicle dynamics by an actual real world CAV prototype, while the other test participants are solely virtual and controlled by the traffic simulation, provides a major benefit. Other test participants are equipped with driver behavior models and do not need to use predefined trajectories. In this way, similar to the co-simulation, the real world CAV prototype reacts to the other test participants and vice versa. All the applications that are covered by the co-simulation approach become feasible for testing real world CAV prototypes driving on proving grounds. This method allows for testing of critical scenarios without causing harm to test drivers and hardware. Because of the absence of sensor uncertainties the actual driving function can be tested. Use cases, such as traffic jams, can be applied simply by adding a vast amount of agents to the traffic simulation. Therefore, it is possible to test CAVs in large scale traffic systems without requiring any test drivers. Of course, the entire toolchain can be applied as well, by replacing the vehicle dynamics simulation with a real world CAV prototype. Further, the overall advantages and benefits provided by the toolchain are valid for the PiL approach as well.

In conclusion, the research objectives and requirements that motivated this dissertation are fulfilled, providing a step forward in the research field of CAV testing. Regarding other approaches, the developed toolchain is set up to be compatible and support other research efforts, such as PEGASUS. The overall contributions show that this dissertation accomplishes an advancement in this particular field and provides an applicable and generic methodology, which can be applied for development accompanying tests as well as for validation of CAVs.

4 Results

The focus in this dissertation lies more on the application of the toolchain itself, rather than changing the parameters as efficient as possible. For that reason, the Chapter Results contains some specifically chosen examples on how the toolchain can be utilized.

Beginning with a description of the CAV prototype's setup and the implemented driving functions, results of the utilized toolchain are presented. Therefore, the CAV prototype that is built for the research project Ko-HAF is used. This prototype is tested on the one hand solely virtually and on the other hand by the PiL approach. Figure 4.1 shows the test methods used in this chapter with respect to virtual and real test environments (see Fig. 3.25).

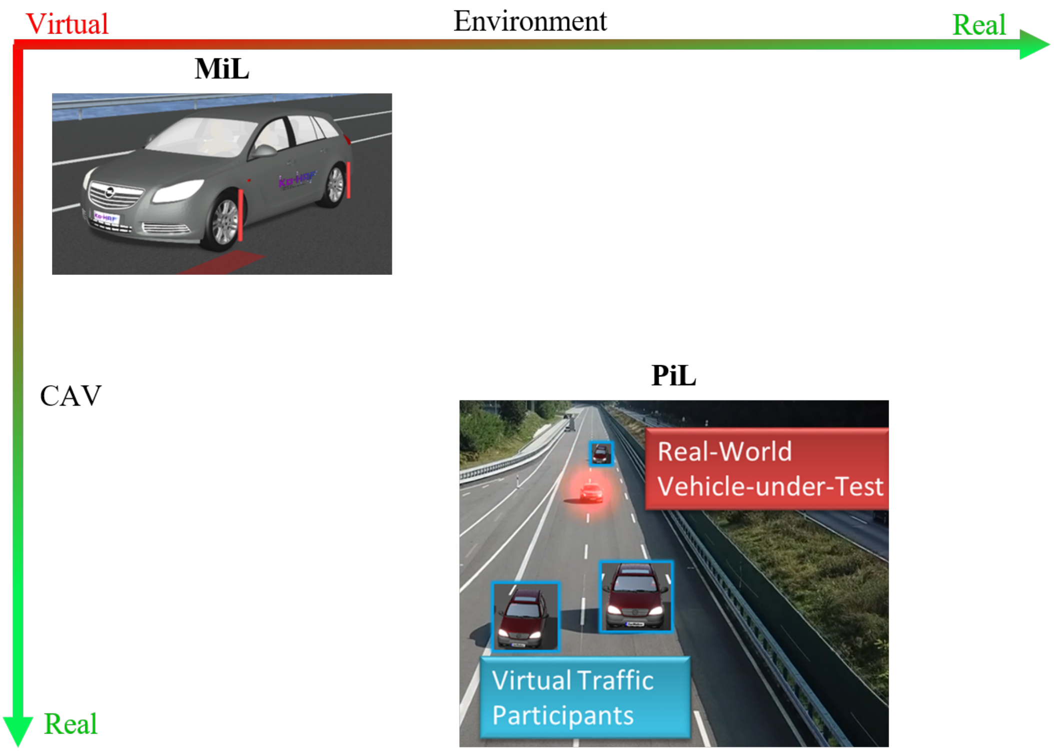

Figure 4.1: MiL and PiL considering virtual and real test environments.

The developed toolchain introduced in Chapter 3 builds the foundation for the presented results. The proceeding and the designed aspects can be considered as the foundation for this chapter. Thus, the metrics introduced before are used to illustrate the evaluation abilities of the toolchain. Aspects, such as safety and traffic quality, are investigated and used to identify critical scenarios following the methodology of the toolchain. The aim is to prove that the toolchain is applicable and can be utilized for the purpose it was designed for. In the evaluation, disturbances are added to increase the criticality of the investigated critical scenarios. As mentioned before, disturbances, such as map errors or aggressive driving behavior, are very common factors that a CAV has to deal with. The first part of the presented results are produced by the coupled simulation environment and evaluated by the metrics introduced in Chapter 3.

Additionally, results for the PiL approach are shown. The application and usability of this method is presented by utilizing one of the major advantages this approach provides. This advantage lies in the testing of safety critical scenarios with a real-world CAV prototype without endangering test drivers. First, a scenario that was designed for the Ko-HAF final event is tested to show that the PiL approach works in general. After that, two safety critical scenarios are configured and investigated. It will be shown that the CAV reacts to the virtual test participants and performs maneuvers according to its implemented driving functions. With this approach the driving functions of the real-world CAV prototype driving on a real-world test track are tested and evaluated.

It should be noted that the implementation of the MiL and the PiL environment as well as the performed tests and results are contributions of this dissertation, while the implementations of the driving functions (see Sec. 4.1) are carried out by a function development team.

This chapter concludes with a discussion of the presented results and how to put them into perspective. Aspects, such as validity, constraints, and operating experience as well as the resilience are discussed and clarified. Furthermore, benefits, further activities, and possible enhancements are elaborated.

4.1 Cooperative and Automated Driving Function

The utilization of the toolchain is presented for a SAE-Level 3 [4] CAV prototype feature highway chauffeur [23, p. 53]. The CAV prototype's driving functions are implemented for the research project Ko-HAF performed by an Opel development team. An Opel Insignia is equipped with components that allow for the development of CAV driving functions. Figure 4.2 shows the prototype that is used for the function development.

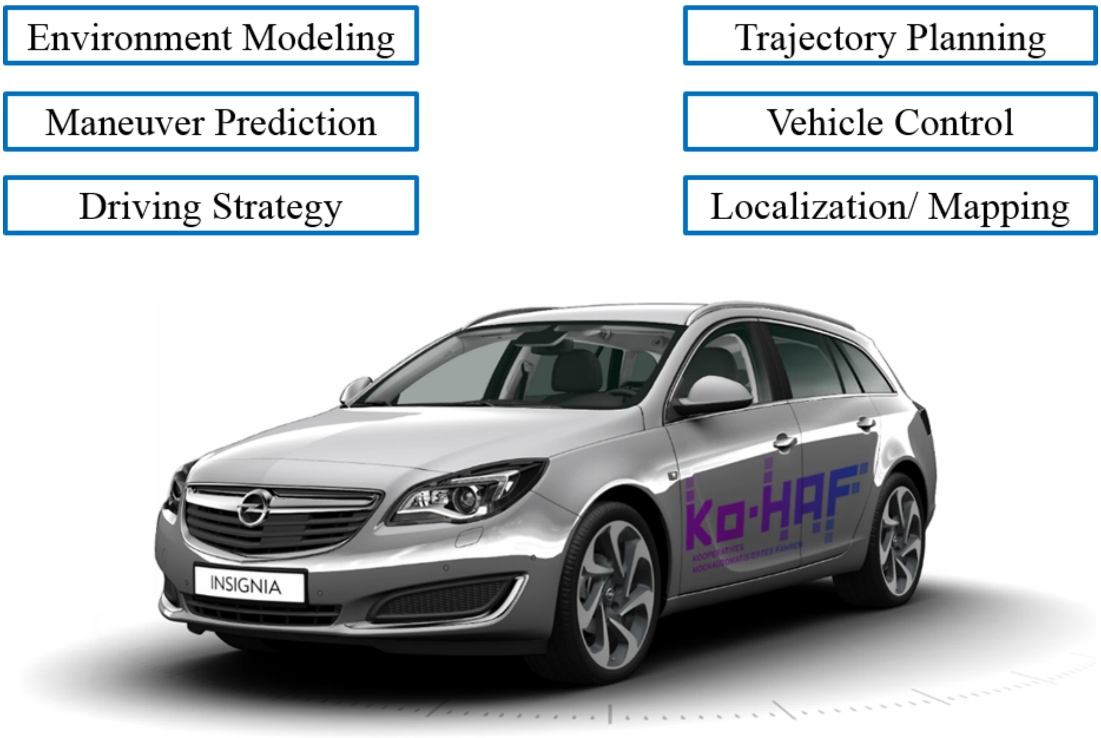

Figure 4.2: Opel Insignia CAV prototype developed for the research project Ko-HAF.

The implementation of the CAV's driving functions follows the robotics paradigm: Sense, Plan, Act [136, p. 189]. Using the equipped sensors the surroundings of the CAV are captured. The CAV prototype is equipped with various sensor systems shown in Figure 4.3. These measurements are consolidated and converted into an environment model as well as a localization determining the current position. The environment model is based on an unscented Kalman Filter fusing the sensor inputs. A SLAM [119] algorithm is used for localization based on a pose graph optimization scheme. A special asset of the prototype is the extension of a precision digital map. The prototype is positioned on that map while providing useful information on road shapes, lane markings, traffic signs, etc.

Thus, the digital map can be considered as an additional asset providing more information for the CAV. One could say, the digital map is used to extend the sensor range of the CAV at least for static attributes.

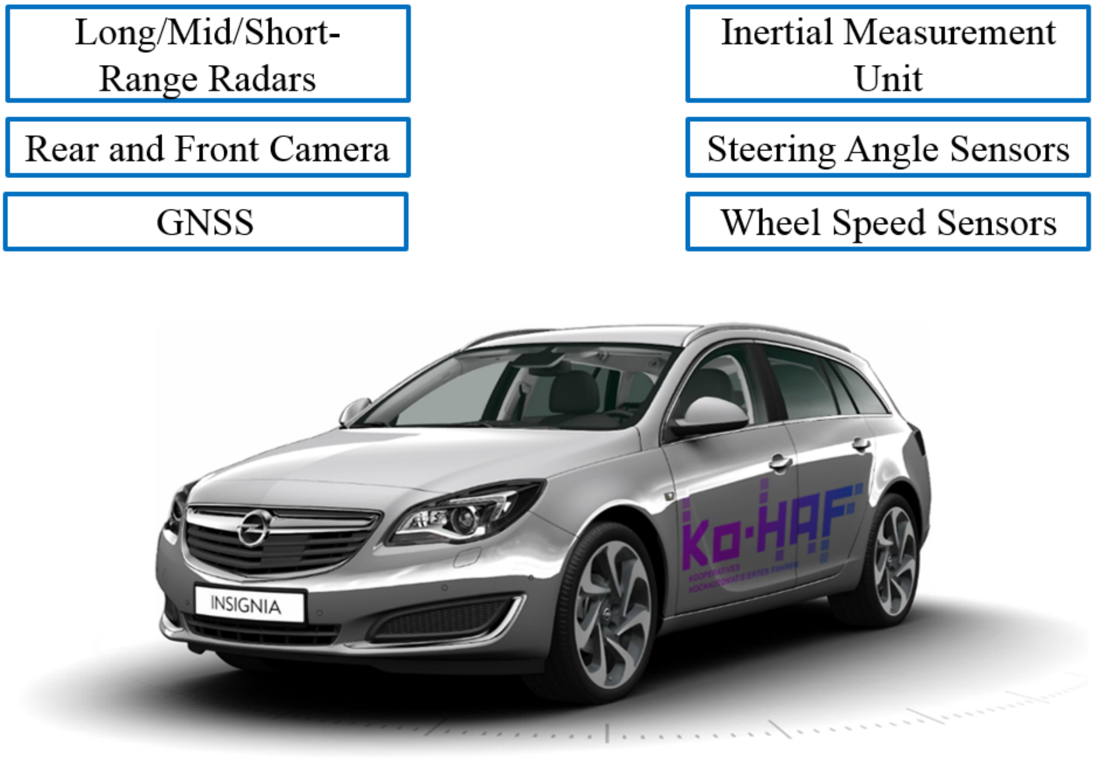

Figure 4.3: Opel Insignia CAV prototype's sensor setup [147, p. 26].

Considering the different sensor perceptions a surround view of the current driving situation is established. Figure 4.4 shows the sensor range and the resulting surround view in detail. The last part of the "Sense" scheme is the maneuver prediction. On the basis of the environment model every other test participant's motion is predicted and out of that the next probable maneuver is calculated. The implementation is performed by statistical motion pattern recognition detecting the maneuver and predicting prototype trajectories for the participants' movements. For further reading [148] is recommended.

The "Plan" part is derived from the calculations of the "Sense" part. First of all, a driving strategy is developed. This strategy decides which reference velocity the vehicle should drive with, when to overtake, and which route the CAV has to take. The routing is performed based on the digital map. The reference velocity can be adjusted and is configured to overtake, whenever another participant forces the CAV to drive with a lower velocity.

Further, aspects, such as leaving the highway for example are considered in the driving strategy. The procedure to leave the highway is planned beforehand to avoid overtaking while the highway exit is approaching soon.

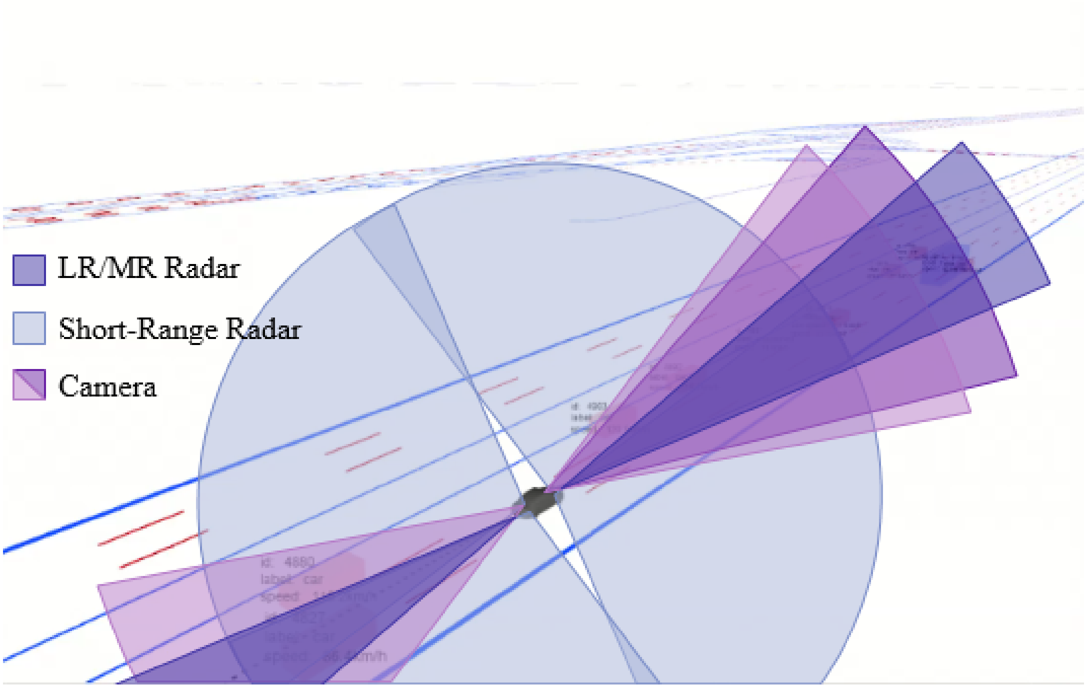

Figure 4.4: Opel Insignia CAV prototype's sensor view [147, p. 26].

The final step is to plan the trajectory of the CAV. This enhances the "Plan" by not only planning of what to do, but with a strategy on how to do it. This means that a trajectory planner considers constraints and limits of the vehicle motion. All of this is captured in a constrained nonlinear optimization problem. To keep this short, just the goals are briefly elaborated on. Separate trajectories for longitudinal and lateral motion as well as an emergency maneuver are calculated simultaneously. The optimization problem considers physical constraints, comfort aspects, traffic regulations, etc.

Based upon the driving strategy and trajectory planning the next step is to "Act", meaning to perform the vehicle motion. For the CAV's guidance the concept of feedback linearizing control is used. The concept aims at compensating nonlinearities of the vehicle model and therefore, the system is transformed to a linear integrator chain. For this system, a linear trajectory following controller can be designed using the entire potential of linear control theory. While implementing the control concepts the parameters of the CAV prototype are identified. The digital prototype used in the MiL results is parameterized with the identified parameters of the function developers designing the control concepts. For further reading [45, 149] is recommended. This concludes the definition of the CAV prototype's technical specification and driving functions.

After elaborating on the CAV prototype the use cases are illustrated in Figure 4.5. Basically, the CAV should be able to handle highway use cases, such as entering, leaving, and overtaking properly.

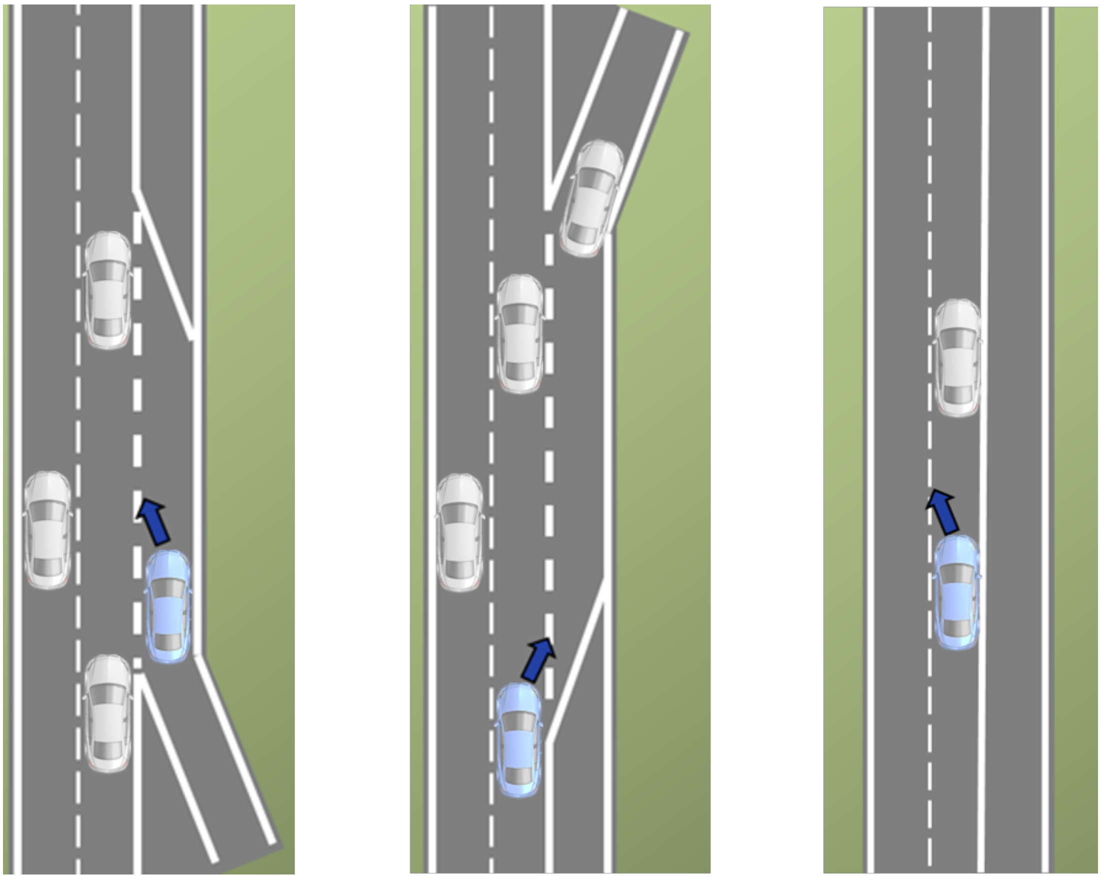

Figure 4.5: Use cases for the Opel Insignia CAV prototype‘s driving function development. CAV colored in blue. Other test participants colored in gray.

In the course of the research project Ko-HAF, the function developers implemented these driving functions completely from scratch. Thus, the functions are built up from bottom to top, which gives us the contingency to utilize the toolchain with driving functions that are in early development stages. The functions are not in a condition for series development and therefore perfect to show the capabilities of the toolchain to support early testing in the development phase of the CAV.

The tests presented in this chapter are performed on the "long straight" located on the Opel proving ground. This test track enables to test highway functionalities providing around 1.6 kilometers of a three lane highway including an entrance and an exit ramp. The function development and testing is mainly performed on this test track for the reasons that confined development and testing is possible.

The precision map data of the proving ground is more accurate and could be evaluated in comparison to the map of highways around Frankfurt. Specific testing of scenarios can be performed with other test vehicles driven by test drivers. Basically, all the functions implemented for this CAV are tested intensively on the long straight before deploying the CAV to public traffic systems. Figure 4.6 shows the long straight constructed for testing highway use cases.

Figure 4.6: Long straight on the Opel proving ground located in Dudenhofen near Frankfurt [150, p. 16].

It is worth mentioning that the test track is equipped with a DGPS transmitter to provide accurate positioning for test evaluation purposes. Localization, trajectory planning, and the vehicle guidance quality can be evaluated. A system to equip other test participants with a DGPS-based ground truth measurement system was developed as well, which enables to replay tests in simulation environments or to provide guidance for test drivers.

At last, the precision map data acquired for this test track lays the foundation for the digital world used for the MiL and PiL results. The parameter spaces of the logical scenarios are derived from the conditions of this track. Selected exemplary concrete scenarios for the PiL results are implemented on the long straight. The capabilities of the CAV and the simulation-based toolchain were presented on the final event of the research project Ko-HAF, which also took place on this test track.

4.2 Disturbances

For evaluation purposes, disturbances are included in the scenarios to increase the criticality of the performed simulation runs. This is done for the reason that the capabilities of the toolchain can be shown at its best, when the CAV's driving functions are stressed to a certain degree. At first, it will be shown that if a normative scenario is applied, the toolchain is not going to classify this scenario as critical. By injecting disturbances, such as sensor errors, map errors, and aggressive driving behavior of other test participants, critical scenarios occur. This proceeding is chosen to present the toolchain's capability of identifying critical scenarios and to prove on an exemplary level that the metrics are sensitive to these critical scenarios.

4.2.1 Sensor Errors

The first disturbance aims at illustrating that sensor errors can lead to critical scenarios. The CAV's velocity profile shows fluctuations caused by a disturbance that emulates a sensor error. Thus, the CAV's trajectory following controller is not performing well due to the emulated disturbance.

Sensor errors are very common for CAVs. Many different aspects, such as weather conditions, are influencing the CAV's perception. Even if the sensors are working properly, a lot of problems can occur caused by the implemented algorithms processing the input data. In general, the trajectory following controller requires a precise position and state space variable observation. Otherwise, the control output is going to guide the CAV in an unsatisfactory fashion. Further, the trajectory planning strongly relies on accurate measurements. For example, the motion of the vehicle may be guided wrongly and the CAV could pass over road boundary lines or disregard safety distances, etc. Another important aspect for controllers is the smoothness of measurement inputs. If the measurement possesses strong fluctuations, the controller tends to destabilize the CAV. Especially, delays in the processing of the measurements are destabilizing factors. Additionally, when applying linearizing control methods, e.g., feedback linearization, the transformation relies on accurate sensor data. Otherwise, the nonlinearities are falsely compensated, which can result in major problems when guiding the vehicle motion.

Regarding the toolchain it is expected that such behavior is classified as a critical scenario that needs further investigation. Generally, it is not that important where the error source lies, instead it is of major interest how the CAV is going to behave in these particular situations. After examining these scenarios the function developers have to decide how to enhance the current implementations to decrease the magnitude of faulty CAV behavior or even find a way to detect these errors and apply a strategy if they occur.

4.2.2 Map Errors

The second disturbance applied to the CAV is a map error. Again, CAVs use precision maps to support driving functions and localization. Map errors can cause various problems regarding the driving behavior of CAVs and it is not improbable that these errors occur. Flaws in precision maps are likely to occur during the map data acquisition, the conversion into the manufacturer's map standard, or even by the function development converting the map for their purposes. Figure 4.7 shows an exemplary map error.

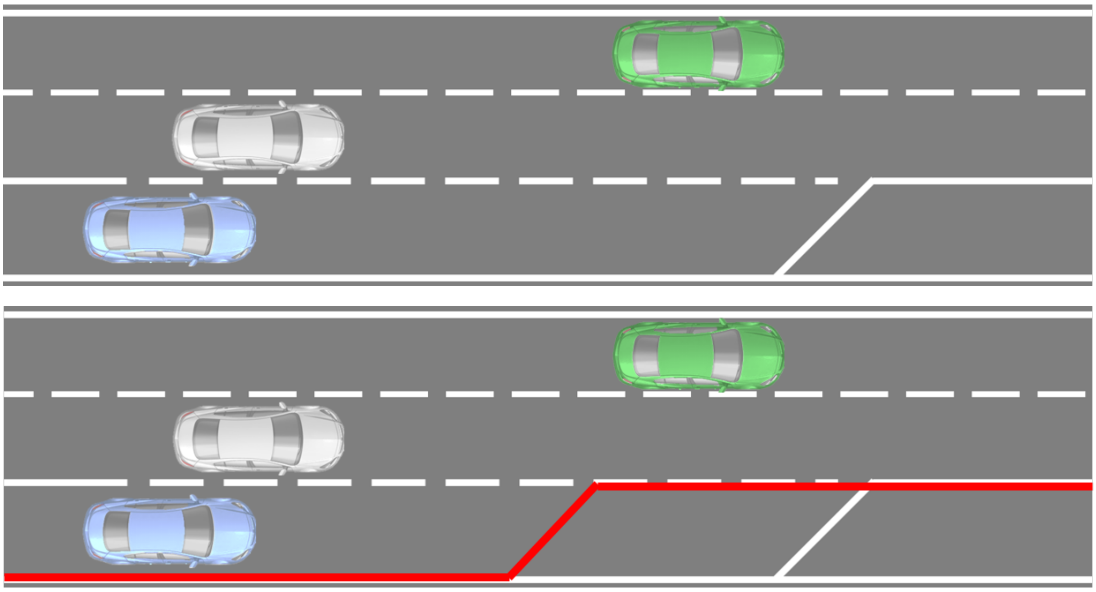

Figure 4.7: Map error shortening the highway entrance ramp (Example). CAV colored in blue. Other test participants colored in green and gray. Correct map above. Map error beneath.

The highway entrance ramp length is falsely shortened and used as an input for the CAV's driving functions. Thus, it is now more probable that the CAV stops on the highway entrance causing poor traffic quality. The shortened entrance constraints the vehicle guidance to a certain degree. There is not much room left to overtake or to find a safe gap while driving on this entrance ramp. In general, short highway entrances are difficult to handle for a CAV. The map error strongly amplifies that issue. It is expected that these kinds of errors are likely and therefore, good examples to present the capabilities of the toolchain identifying critical scenarios.

Additionally, regarding cooperative maneuver planning, these errors can cause major implications. If cooperative trajectories are planned and the maps differ from one another, the planning could lead to serious mistakes and accidents could be caused. Hence, keeping the precision maps accurate and consistent, even between different manufacturers is an important aspect for CAV safety in general.

4.2.3 Driving Behavior of Other Test Participants

The third and last disturbance is also a common one. It is planned to incorporate CAVs in public traffic systems. This means that CAVs have to be able to handle interactions with human drivers properly. It would be easier to let CAVs drive in domains where only CAVs are permitted in, but that is not the actual plan of vehicle manufacturers or the government. It is not intended to build road systems where only CAVs are driving, although this would make the world for function developers way easier, because CAVs are going to drive normatively up to defensively and follow to traffic regulations. Instead, CAVs have to deal with insecure, defensive, and aggressive human drivers. Figure 4.8 shows an example of an aggressive test participant tailgating the CAV without regard for safety regulations.

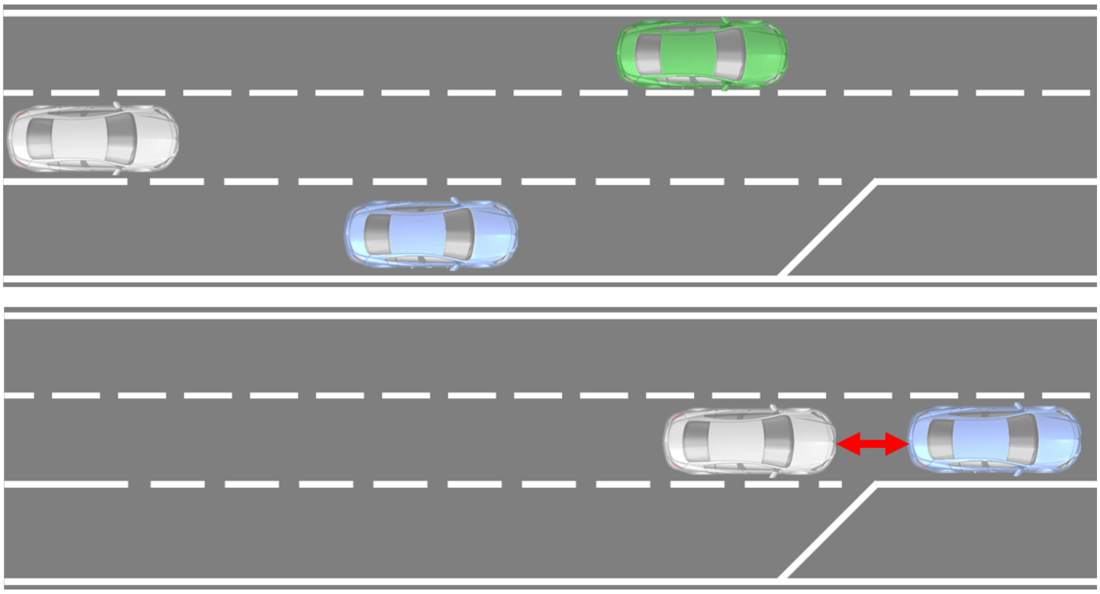

Figure 4.8: Critical scenario caused by human driving behavior. CAV colored in blue. Aggressive test participant colored in gray. Initial state above. Final state beneath.

Even though the CAV is not causing this critical scenario, at least it needs to be ensured that the driving function reacts properly and does not worsen the situation due to a rapid unforeseen reaction. A strong deceleration of the CAV, for example, would cause a collision, because the human driver is not expecting such a behavior from another human driver.

In general, crowded public traffic systems especially during rush hours possess their own characteristics and it is not unlikely that regulations are disregarded by human drivers. When deploying CAVs to public traffic systems, it is absolutely necessary to test these scenarios intensively.

4.3 Model-in-the-Loop Results

The MiL results are presented on the following pages. It is worth mentioning that the results are deliberately presented in a compact fashion. The toolchain and the metrics are designed to classify, if a scenario is critical or not, and precisely that capability is presented here. It is one objective to reduce the results to a few metric outputs that can be interpreted intuitively. At first, the results of the coupled co-simulations are presented. These results are only virtual. Figure 4.9 shows the general setup of the MiL environment.

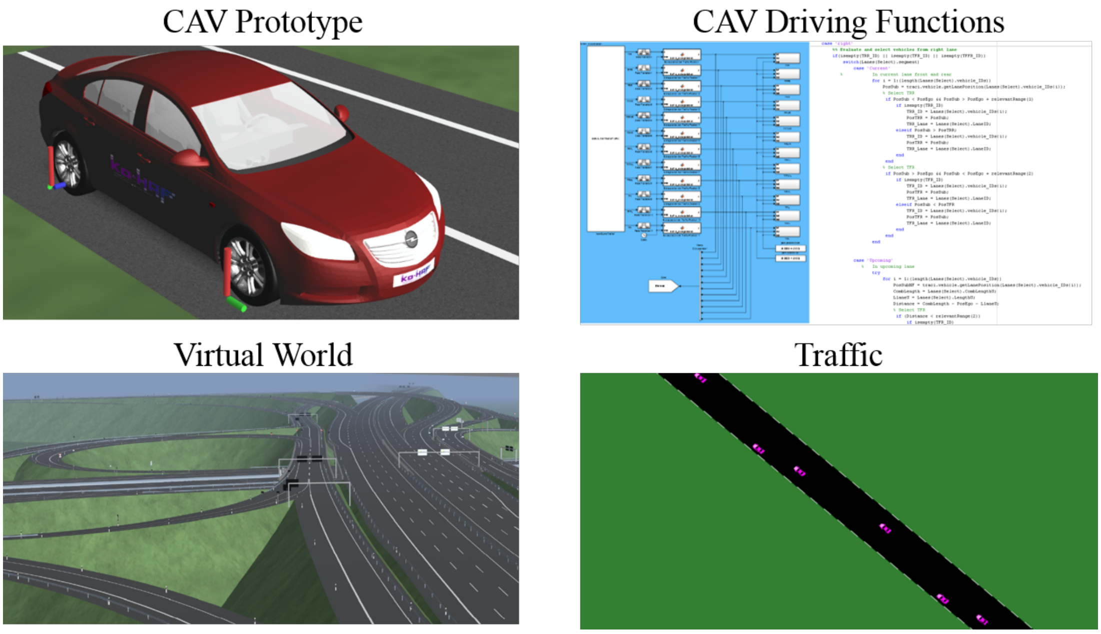

Figure 4.9: MiL environment setup containing the driving functions, a CAV prototype, a virtual world, and the traffic provided by the traffic simulation tool.

The MiL environment consists of the CAV's digital prototype in which the driving functions are implemented. This prototype drives in a virtual world, while both a vehicle dynamics and a traffic simulation run simultaneously exchanging the necessary information between each other. The digital prototype is controlled by the CAV driving functions and the other test participants by the traffic simulation's driver behavior models. Thus, the CAV reacts to other test participants and the other test participants to the CAV. It is worth to mention, that the CAV is a prototype in an early development stage. The functions are developed in a research project and are not compliant with series production standards. Furthermore, this is a research effort that does not reflect the Opel autonomous driving series production requirements.

The use cases presented are for a CAV entering the highway. Table 4.1 shows the most important parameter spaces of the logical scenario for the presented run-through the toolchain, which is performed on the Opel test track.

Table 4.1: Logical scenario for the performed run-through the toolchain. [8, p. 102]

Attribute	**Parameter Space**	**Determined by Example**
Entrance Ramp Length	$l_{\text{min}} - l_{\text{max}}$	410 m
Number of Lanes	$N_{\text{min}} - N_{\text{max}}$	4
Speed Limit Highway	$v_{\text{min}} - v_{\text{max}}$	36.1 m/s
Traffic Flow	$Q_{\text{min}} - Q_{\text{max}}$	1 veh/s

The parameters are mostly defined by static conditions of the test track. Of course it is possible to capture many more parameters, but to keep it short, only some of major importance are presented here. The traffic flow is set to a value where the highway is crowded but not congested. The initial conditions of the simulation runs are shown in Figure 4.10.

Figure 4.10: Initial conditions of the performed simulation runs. CAV colored in light red. Other test participants colored in dark scarlet [8, p. 102].

The thresholds for the criticality evaluation are $G_{\text{final}} \geq 0.279$, $TTC \leq 3.9$ s, $TTB \leq 3.8$ s, $a_{\text{req}} \leq -2\ \text{m/s}^2$ [8, p. 102]. Safety related thresholds are obtained by a paper [130, p. 157] that determined these by observing natural driving data. The thresholds for the traffic metrics are determined in Section 3.1.9 by the usage of training data.

The CAV is supposed to enter the highway in all four presented concrete scenarios. The first scenario is a so-called normative scenario, where the CAV enters the highway without any critical incidents. This scenario is shown to illustrate that the toolchain does not classify normative behavior as critical. Table 4.2 shows the resulting simulation run's metric calculations.

Table 4.2: Results of the concrete scenario without disturbances [8, p. 103].

Scenario Characteristics	**Results**
Disturbance	None
Safety Metrics	$TTC_{\text{krit}} = \emptyset$, $TTB_{\text{krit}} = \emptyset$, $a_{\text{req,krit}} = \emptyset$
Traffic Metrics	$G_{\text{final}} = 0.15$
Criticality	Not Critical

Correctly, the scenario is classified as not critical indicated by the empty set $\emptyset$ for the safety related metrics. Furthermore, the traffic quality metrics do not deflect either, because the CAV showed normative behavior. Thus, concrete scenarios, which are not critical, are not further considered.

In the second concrete scenario, the motion planner shows fluctuating behavior. This causes the CAV to control the velocity in an unsteady manner, which can be considered as an analog to sensor errors, e.g., disturbances in localization algorithms, perception systems, etc. Table 4.3 shows the simulation results, clearly classifying this scenario as critical.

Table 4.3: Results of the concrete scenario with sensor errors [8, p. 103].

Scenario Characteristics	**Results**
Disturbance	Sensor Errors
Safety Metrics	$TTC_{\text{krit}} = 2.9$ s, $TTB_{\text{krit}} = 1.2$ s, $a_{\text{req,krit}} = -6\ \text{m/s}^2$
Traffic Metrics	$G_{\text{final}} = 0.29$
Criticality	Critical

The CAV possesses a strong fluctuating velocity profile and passes slightly over the lane marking at the end of the highway entrance. Due to that fact, the motion planner reduces the velocity further, which causes the traffic related metric to deflect. Thereby, another test participant slightly tailgates the CAV and therefore, the safety related metrics deflect as well. So, this scenario is identified as a critical scenario and needs further investigation. Thus, it is added to the scenario catalog and passed on to the function development department.

The third simulation run accounts for occurring map errors. These errors are very likely and were observed during the setup of the simulation environment's static coupling. In this case, a map error shortens the length of the highway entrance ramp and reduces the movement range of the CAV. Actually, it poses a lot of constraints for accelerating, overtaking maneuvers, and shortens the time to make a well-founded driving strategy decision. Table 4.4 shows the metric results of the performed simulation run.

Table 4.4: Results of the concrete scenario with map errors [8, p. 103].

Scenario Characteristics	**Results**
Disturbance	Map Errors
Safety Metrics	$TTC_{\text{krit}} = \emptyset$, $TTB_{\text{krit}} = \emptyset$, $a_{\text{req,krit}} = \emptyset$
Traffic Metrics	$G_{\text{final}} = 0.46$
Criticality	Critical

First of all, there are no safety related issues, because no other test participant drove in front or behind the CAV. This is again indicated by the empty set $\emptyset$ in Table 4.4. The shortening of the highway entrance causes the CAV to perform a full stop on the entrance. A SAE level 3 driving function would now transfer the driving task back to the driver. Clearly, regarding the function's scope, this is an acceptable behavior. For driver convenience and the acceptance of CAVs in general, this poses a major problem. If a CAV stops on the highway entrance ramp, it forces the driver into an uncomfortable situation. Furthermore, if this happens too often, the driver will loose trust in the abilities of CAVs. For promoting CAVs to be a helpful extension for customers, such behavior can be considered as very poor and needs to be avoided. Tests on these issues have to be performed intensively and strategies to avoid this behavior need to be developed. It is worth mentioning, that these errors can also cause safety critical scenarios. If the CAV stops on the highway entrance ramp, other human drivers are annoyed and thus, it could lead them to perform safety critical maneuvers as a result of the poor driving behavior of the CAV. Now, the traffic related metrics deflect strongly as expected. Regarding traffic quality this behavior can be clearly stated as critical. The consideration of the individual quantities of the CAV in the overall metrics combination pushes the criticality value of the calculation way above the threshold. This scenario also needs further investigation, which is correctly classified by the toolchain.

The fourth and last presented concrete scenario includes the disturbance of aggressive driving behavior. The ability to change the driving behavior of other test participants is one of the major assets of the toolchain. Human drivers can cause a lot of problems for the CAV's driving functions and the probability that these issues occur is very high. In this simulation run the driver behavior models of the traffic simulation tool are changed to aggressive. This means that the other test participants are driving fast and do not maintain safety distances. Table 4.5 shows the results of the simulation run.

Table 4.5: Results of the concrete scenario with aggressive driving behavior[8, p. 103].

Scenario Characteristics	**Results**
Disturbance	Aggressive Driving Behavior
Safety Metrics	$TTC_{\text{krit}} = 1.9$ s, $TTB_{\text{krit}} = 1.2$ s, $a_{\text{req,krit}} = -5.7\ \mathrm{m/s^2}$
Traffic Metrics	$G_{\text{final}} = 0.19$
Criticality	Critical

Shortly after entering the highway another test participant tailgates the CAV. This is noticed by the safety critical metrics and therefore, classified correctly as a critical scenario by the toolchain. The driving function decreases the velocity of the CAV due to the behavior of the other test participant. It could be argued that this motion worsens the situation, because it further reduces the safety gap. The other test participant comes even closer. Out of the domain of interest of this scenario, the safety distance decreased further and the safety related metrics showed even stronger criticality values. Regarding the use case of entering the highway, these values are not in scope and the criticality for the highway entrance are sufficient to fall below the criticality threshold. The traffic related metrics did not deflect, because there was no significant decrease in traffic quality in this simulation run.

Regarding the toolchain, the performed simulations are not enhanced by the aspects of increasing the validity provided. That is not done for the reason that in this stage the implemented driving functions are not yet ready for test benches or driving tests. The digital prototype model was chosen in cooperation with the function development department and includes a highly complex nonlinear two track model for the vehicle dynamics, which can be considered as a bit too exact for this purpose. At this stage, a simpler model would have done the job as well. Broadly speaking, it is not always necessary to use a highly complex vehicle model. Regarding calculation efforts, it is thinkable to use a less complicated vehicle model, in particular, when the driving functions are in this early development stage. Anyways, the function development department uses this model and identifies the parameters for the digital prototype. Hence, it was decided to equally use these simulation models for development and testing.

In summary, the driving functions presented here are in an early development stage. This is particularly beneficial for illustrating the toolchain capabilities and what it was designed for: The improvement of systematical testing during the development phase of a CAV. It is a major asset to use this toolchain for early finding flaws in the implementations and performing comprehensive testing before handing over the CAV for validation. This is necessary to avoid too many feedback loops between validation and development, because the validation is so demanding by itself.

Lastly, the simulation runs and the overall data are saved to a database. Now, the function developers can re-run the simulations and improve their driving functions. A test result sheet is created and passed on. This concludes an exemplary run-through the toolchain, which proved on an exemplary basis that the toolchain is able to identify critical scenarios and classify them correctly. The calculated metrics deflected whenever something critical occurred. Regarding the metric of required deceleration, it should be noted that it is necessary to observe the object acceleration closely, because it is determined by differentiating the velocity of the other test participants numerically and therefore, some outliers can occur. The thresholds can be adapted by the manufacturers' guidelines themselves. Each manufacturer possesses its own views of what criticality threshold should be used.

It is thinkable to subdivide the safety related metrics into different criticality classes achieving a hierarchical clustering on how severe the critical scenarios are. In the next section, the results of a useful enhancement of the toolchain called Prototype-in-the-Loop are presented. For this demonstration, the functions are in a more mature stage and even tests that are potentially dangerous can be performed using a real-world CAV prototype driving on the Opel test track.

4.4 Prototype-in-the-Loop Results

In this section the results of the novel approach of Prototype-in-the-Loop are presented. Referring to Section 3.2, the real-world CAV prototype is coupled with a traffic simulation tool driving on a real-world test track. The other test participants are virtual and controlled by the driver behavior models of the traffic simulation. Regarding the toolchain, some exemplary selected concrete scenarios will be shown. Basically, the left hand side of the toolchain is used to perform some dangerous tests that can not be conducted by driving tests involving test drivers. Now, this section aims at demonstrating the capabilities of this method, which has to be elaborated further in future research. The CAV driving functions are in a more mature stage now and were presented at the Ko-HAF final event. Furthermore, the use case that was demonstrated is illustrated at first and then tested with the PiL approach. Of course, everything presented in Section 4.3, such as traffic and safety related metrics, could be applied for this method, too. In order to show how the PiL method works we disregard these aspects and illustrate some tests that are performed out of the scenario catalog. It should be noted that this is just an illustration of an enhancement that is useful for the overall methodology. A whole run-through the toolchain is not performed at this point and needs to be elaborated on in future work.

The first use case aims at demonstrating how the CAV handles usual highway scenarios and starts by entering the highway, overtaking, due to a slow driving test participant, and leaving the highway afterwards. One of the main characteristics of the scenario is that the CAV wants to overtake another vehicle, but has to wait, because another test participant blocks the left lane. So, the CAV is waiting until this participant overtakes the CAV and then the CAV overtakes the slower driving test participant. After the overtaking maneuver, the CAV changes back to the right lane and leaves the highway. Figure 4.11 shows this Ko-HAF use case. The CAV colored in blue enters the highway and drives behind a slow driving test participant colored in gray. Therefore, the driving strategy decides to overtake, but in this moment another test participant, colored in green, approaches on the left lane with high velocity. The CAV decreases its own velocity, keeps its lane, and waits until the fast driving test participant overtakes. Then changes the lane, overtakes the slow driving test participant, changes back to the right lane, and leaves the highway.

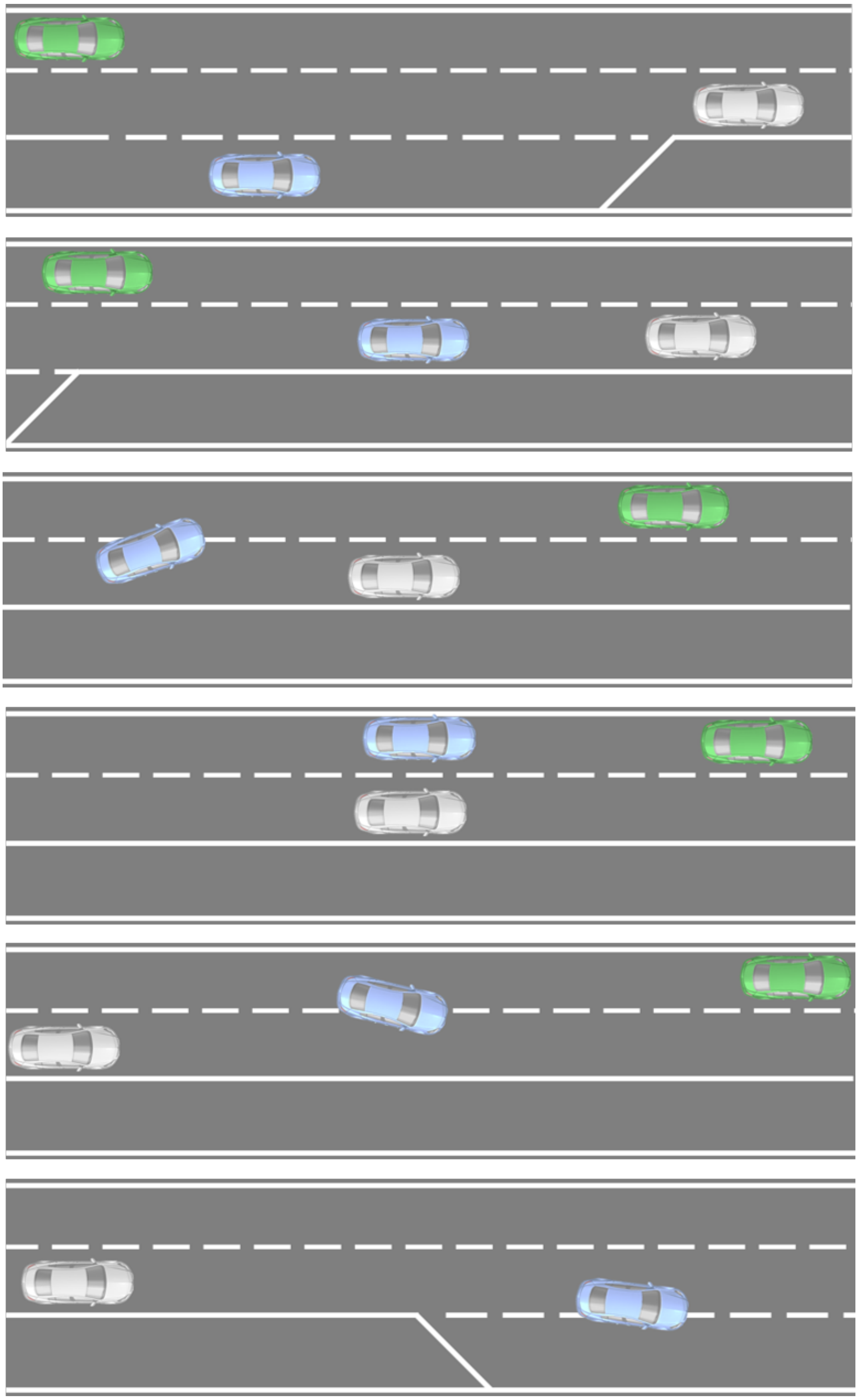

Figure 4.11: Illustration of the first test scenario performed by the PiL approach (Ko-HAF use case). Initial state at the top. Steps in between in the middle. Final state at the bottom.

To get an intuitive overview a tool is implemented to present the current driving situation and the sensor perception synchronized with a webcam recording a video out of the CAV while driving on the test track. Of course, the other test participants are virtual and not visible for the webcam, but it is helpful to have this video data for plausibility checks. Additionally, this illustrates that the CAV is actually driving on a test track performing tests. In the middle of Figure 4.12 the virtual test participants are plotted to track what the driving functions received as input from the traffic simulation tool. The other test participants' velocities are indicated by a color code providing an intuitive overview of the current situation. Figure 4.12 shows a snapshot of the tool for monitoring the performed tests.

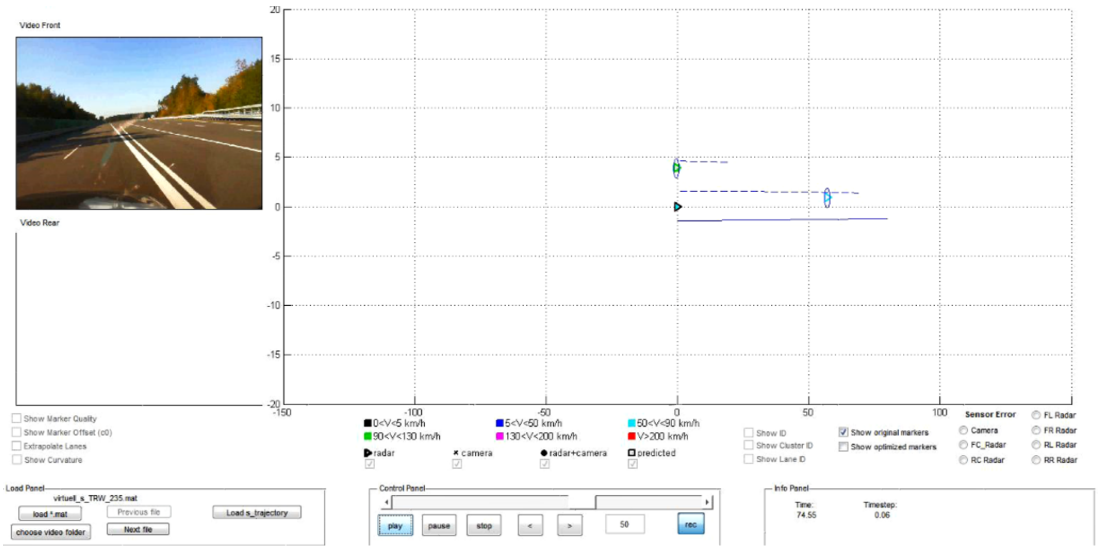

Figure 4.12: Test monitoring tool to replay and observe performed tests [147, p. 42].

The Ko-HAF use case is now evaluated with the PiL approach. Therefore, the other test participants' trajectories are implemented as illustrated in Figure 4.11 accordingly. The test was passed without any incidents. The most important part of the test, which is the moment the CAV wants to overtake the vehicle, but the left lane is blocked is illustrated in Figure 4.13. Deliberately, the rest of the test is not illustrated for clarity purposes and to keep this section sufficiently brief. The test shows that the CAV wants to overtake test participant 2, but keeps the lane (KL) waiting until the other test participant 1 passes the CAV on the left lane. Then, the lane change (LCL) is initiated and the overtaking maneuver is performed correctly. Note, that the longitudinal and lateral distances are presented with respect to the CAV's fixed reference frame. This frame travels with the CAV's velocity while the longitudinal axis points in the the CAV's traveling direction and the lateral axis is orthogonal to that. Figure 4.13 illustrates that the CAV wants to overtake test participant 2, indicated by the driving strategy (at approx. $t = 65$ s), but the left lane is blocked by the approaching test participant 1.

The CAV reduces its velocity until the left lane is not occupied anymore and after that (at approx. $t = 75$ s), it performs a lane change and overtakes the slower driving test participant 2, indicated by the velocity profile as well as the relative longitudinal and lateral distances. Further, the CAV performs a lane change back to the right lane and leaves the highway after that. The highway exit is not shown for clarity reasons.

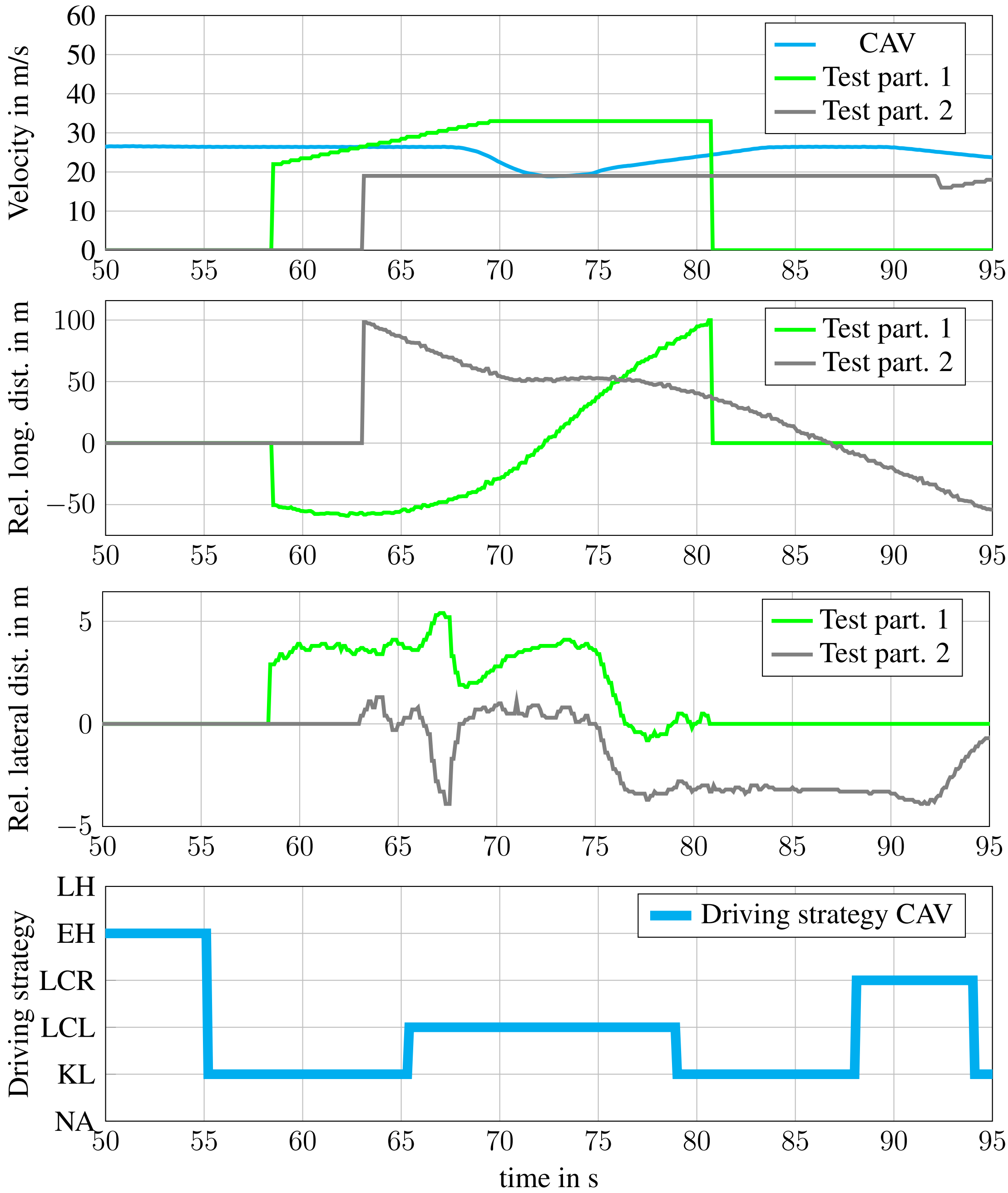

Figure 4.13: Test results of the first performed test scenario (Ko-HAF use case).

This proves that it is in general possible to perform tests with the PiL approach in a real-world CAV driving on a real-world test track. The CAV reacts to the virtual test participants while the driving functions that are implemented are actually controlling the motion. The logging data is obtained from the hardware included in the CAV prototype and partially retrieved from the function development system and the CAN bus. To show the prospects of the methodology some dangerous tests are presented. Due to the reason that the other test participants are virtual it is possible to test scenarios that are too dangerous to perform with human test drivers. The second scenario illustrates the CAV's behavior, if another test participant, driving on the same lane ahead of the CAV, performs a strong braking maneuver. This represents common behavior that occurs very likely, for example, when approaching the end of a traffic jam. Figure 4.14 shows the scenario that is tested.

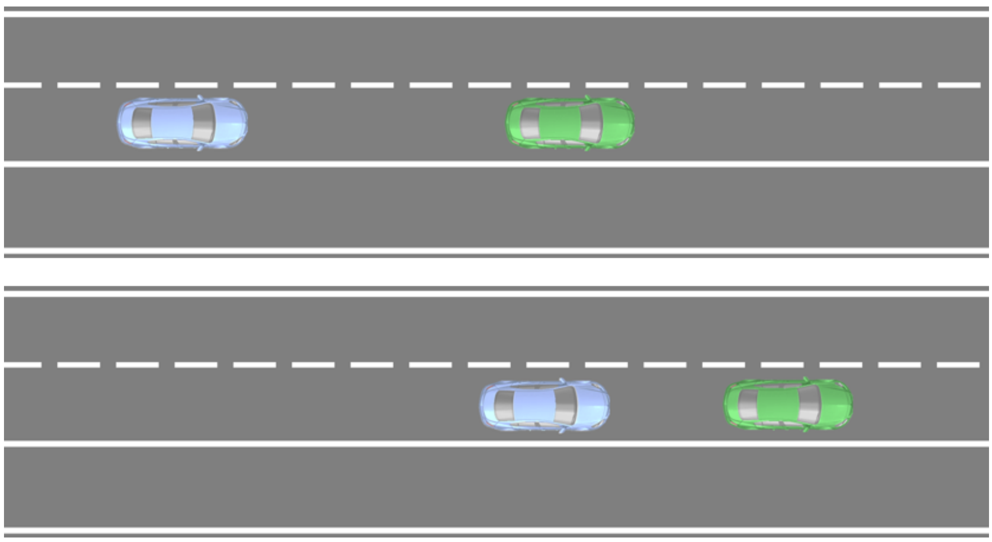

Figure 4.14: Illustration of the second test scenario: test participant performs a strong braking maneuver. CAV colored in blue. Other test participant colored in gray. Initial state at the top. Final state at the bottom.

Figure 4.15 illustrates the CAV behavior. First of all, after the activation of the driving function the CAV enters the highway (at approx. $t = 47$ s), indicated by the driving strategy. It can be seen that the CAV reduces its velocity (at approx. $t = 73$ s), illustrated by the velocity profile, due to the braking maneuver of test participant 1. Simultaneously, the CAV performs a lane change and overtakes test participant 1 and adjusts its velocity to the reference value again. After that, the CAV leaves the highway. The CAV avoided the collision by reducing the velocity and performing a lane change. On the other hand, the maneuver could be considered as safety critical, because the CAV came very close to a collision. This means that the function development should implement a more secure driving function to handle these scenarios even safer.

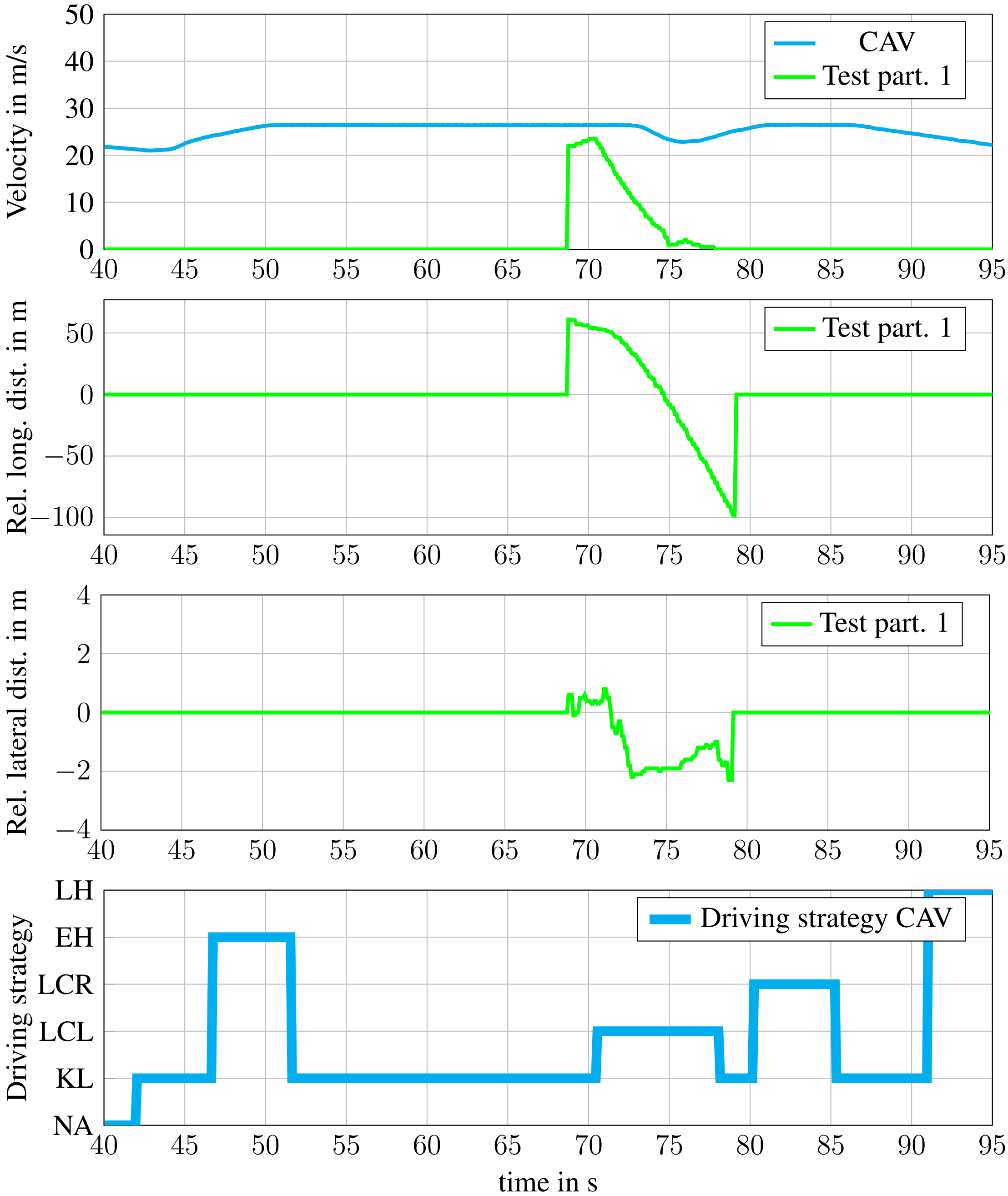

Figure 4.15: Test results of the second performed test scenario: test participant performs a strong braking maneuver.

The third test is also a very common one: another test participant performs a cut-in maneuver disregarding the safety distance. Figure 4.16 shows the scenario based on which upon the test is built. The other test participant cuts in right in front of the CAV disregarding safety regulations. It should be noted that this behavior can be observed frequently, especially in densely populated traffic systems.

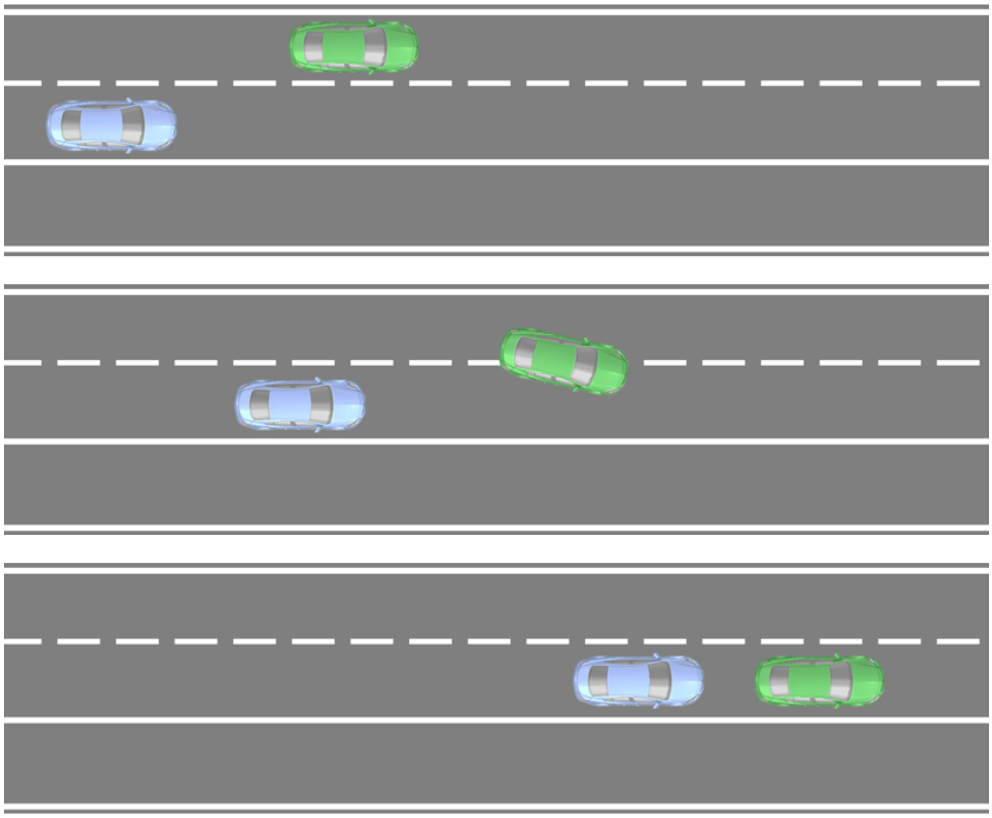

Figure 4.16: Illustration of the third test scenario: test participant performs a dangerous cut-in. CAV colored in blue. Other test participant colored in gray. Initial state at the top. Final state at the bottom.

Again, the trajectory of the other test participant is set up by the traffic simulation tool. The CAV reduces its velocity due to the cut-in maneuver, performs a lane change and overtakes the other test participant. It can be observed that the CAV's behavior is appropriate for the concrete scenario. Figure 4.17 shows the logging data results of the performed test. Test participant 1 is driving on the left lane with a slower velocity than the CAV, indicated by the velocity profiles. Then, test participant 1 changes the lane (at approx $t = 32$ s), which results in a velocity reduction of the CAV, and simultaneously, the CAV performs a lane change (LCL) to pass test participant 1, illustrated by the relative lateral distance profile and the driving strategy. After overtaking, the CAV moves back to the right lane and sets its velocity to the desired reference value. All in all, this could be considered as appropriate behavior of the CAV.

Even considering that these tests are designed to evaluate dangerous scenarios that are not easy to handle, the CAV did not cause a collision. That speaks for the implemented driving functions, although it does not guarantee that the CAV reacted in the best way possible. Further investigation of these tests and the current implementation is necessary to improve the performance.

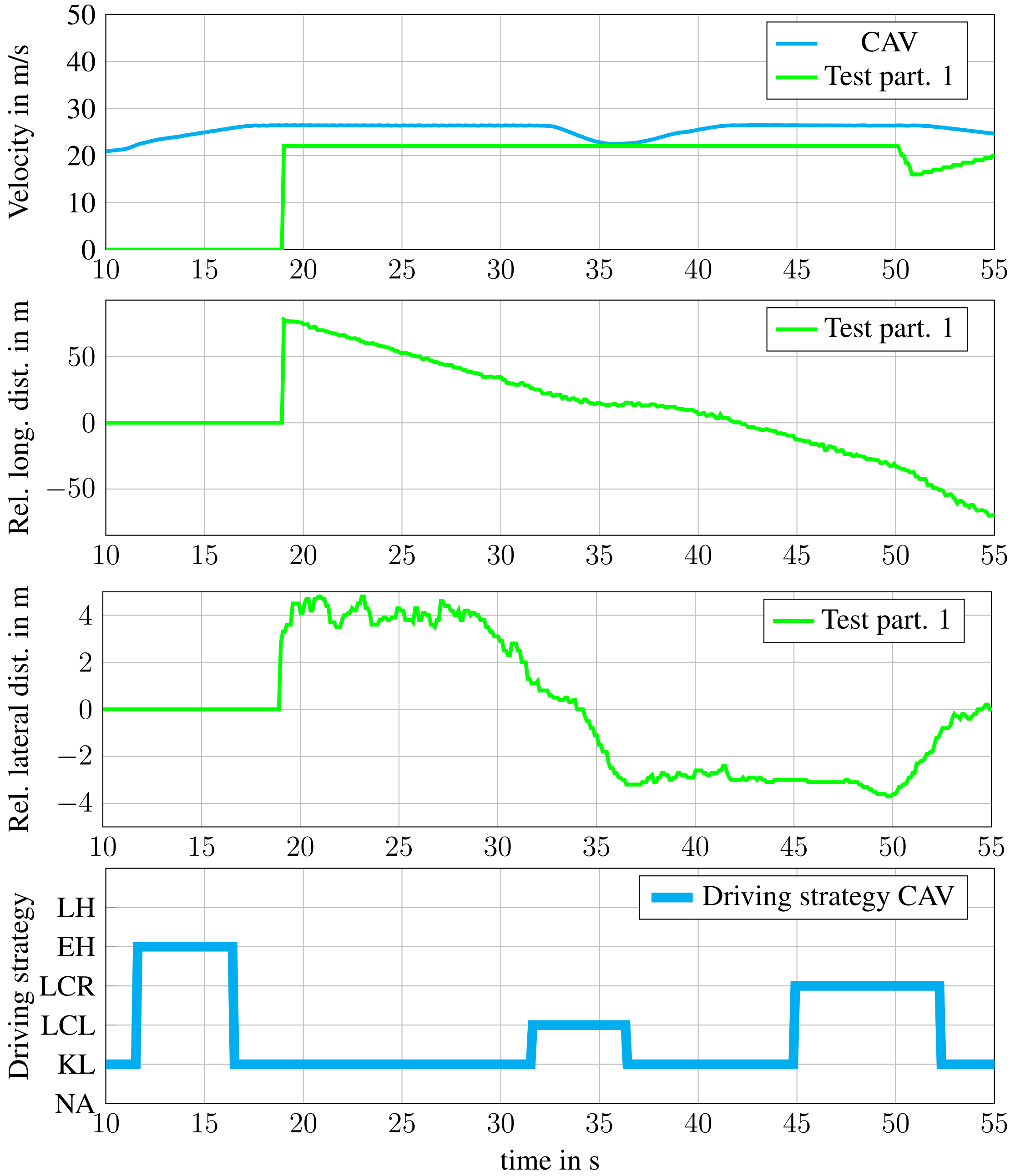

Figure 4.17: Test results of the third performed test scenario: test participant performs a dangerous cut-in.

Anyway, the methodology is proven to be a valid enhancement of existing test environments. The function development department can greatly benefit from the advantages of the PiL approach and efficiently improve their driving function implementations. All in all, these critical scenarios have to be tested and the behavior of the CAV needs close observation. In collaboration with the function development department, an automated test result evaluation can be created. The logging data that was aligned with the function developers is used to evaluate the test. The requirements for the tests developed with the function development department are stated as a first draft as follows:

- Disregard of speed limits.
- Obeying to maximum longitudinal/lateral acceleration limits due to comfort aspects of the CAV.
- Sufficient relative velocity during overtaking maneuvers.
- Out-braking of other test participants due to performed lane changes or when entering the highway.
- Disregard of safety gaps.
- Occurrence of collisions between the CAV and virtual test participants.

As already indicated, this method is in an early stage and needs further elaboration. Anyway, the approach can be applied and used to improve the real-word CAV prototype's driving functions. This proceeding played a major role in testing the CAV prototype for the Ko-HAF project. Implementation changes could be tested with this method before involving other test drivers. The procedure is to first run the scenarios in the simulation, then with the PiL approach, and after that with test drivers. In this way, the validity increases with every test method and therefore, immature implementations do not have to be tested by driving tests. Thus, if a test already fails in simulation it does not make sense to perform driving tests with higher test effort.

Regarding this section, the results presented are only to show the capabilities of the PiL approach. A lot of test effort can be reduced using this approach. In fact, these tests possess another huge advantage, because they can be performed in a reproducible manner. It can be assured that the scenarios are always the same. Additionally, the scenarios that are set up in the traffic simulation tool provide the possibility to run the same scenarios in the simulation as well. This works in both directions. Scenarios tested in the simulation can be transferred to the proving ground as well. Further, tests that are performed by the PiL approach can be replayed in the simulation environment. The function developers consider this a helpful enhancement for the development and implementation of the CAV prototype. Especially for the Ko-HAF project, this enhancement can be considered as very beneficial for the final presentation and supporting the showcase of the CAV capabilities to the general public.

4.5 Discussion

In this chapter, it was shown that the toolchain and the PiL approach are useful tools for improving testing of CAVs in general. The results are on an exemplary basis and more tests have to be performed in the future. But in principle the framework shows the capabilities that are required for the enhancement and improvement of existing CAV test methods.

Now, to put the shown results into perspective, it is clear that the presented toolchain was applied for a driving function that is able to perform highway use cases. Further, the metrics were trained for these scenarios. At least for the traffic quality metrics the training has to be performed for other use cases, for example, rural roads or urban areas as well. Although the safety metrics are well-established and have been used for a long time in the automotive industry, they need close observation during the evaluation process. Particularly, the required deceleration shows outliers that occur because of numerical differentiation in the calculation. Further, it is necessary to investigate the thresholds for these metrics more closely. With more experience in using the toolchain, the thresholds can be adapted to the needs of CAV manufacturers. Additionally, recorded data can be used to determine the thresholds more precisely and an analysis applying the ROC method can be performed. The training of metrics could be furthermore enhanced by experiments combining simulation, PiL, test benches, and driving tests. A statistical analysis considering the level of significance should be taken into consideration. It should be noted that the traffic quality metrics presented here are not universally valid, but sensitive to the use cases shown in this dissertation.

Another consideration that is worth to mention is that the results depend on the used models and their validity. Even the way the coupling was implemented needs to be considered for evaluation in the future. These aspects need to be closely observed when applying these methodologies. In this dissertation, the digital prototype is based on a highly complex and comprehensive vehicle dynamics model. Therefore, the behavior can be considered close to the real-world CAV, but especially the virtual sensors that are used differ from the real-world sensors. These differences influence the validity of the performed tests. Thus, the tests have to be performed in other test environments to increase the validity of the results. The future development of simulation tools in this direction is going to be helpful for targeting this issue. Nevertheless, the results show the capabilities of the toolchain and the PiL approach. The metrics that are used are sensitive to these tests and it could be confirmed that the methodology has proven its suitability. Furthermore, the fact that the methodology was used to test the driving functions of the Ko-HAF CAV prototype by the function developers demonstrates the usefulness and applicability of the entire approach. It is worth mentioning that the presented results originate from testing with an existing CAV prototype including actual driving functions, which were recently developed by a function development team.

Currently, further refinements on the individual aspects of the toolchain are in progress or soon to be approached. The toolchain provides a systematic and generic guideline for testing of CAVs. Further, each individual aspect provides a wide range of possible analyses and elaborations. Subsequently, the methodology can be extended for a broad series of applications. The parameter variation module opens up a research field on its own. It is a challenging task to set up this module properly for automatically generating concrete scenarios based on the parameter spaces of the logical scenarios. Further, the application and the demands of the development team are important for the proposed setup. Close interactions between function development and the operator of the toolchain, if they are not in the same department, are necessary. Additionally, if the approach is applied for validation, a vast amount of arrangements have to be taken into account. Highly dimensional parameter spaces tend to result in a so-called parameter space explosion [114], meaning that the parameter spaces as well as the identification of relevant ones is a difficult task. Furthermore, the sampling within these spaces largely depends on the specific applications, ODDs, and requirements.

Further activities are required in the field of data handling and applicability. A lot of interfaces and automated data processing are necessary to apply this toolchain for CAV developers. Especially, because of the availability of XiL, test benches, and test centers individual adjustments are required. Existing simulation infrastructures need to be adapted to ensure compatibility with the proposed methodology.

The toolchain is already used to evaluate cooperative features within another research project. By simply exchanging a few aspects of the methodology, the toolchain enables testing of cooperative functions as well. This is provided due to the generic setup of the approach and constitutes a major advantage of the methodology in general. Because of this research, further experience on operating the toolchain is gathered [121].

Recent activities applying the toolchain for training and testing machine learning algorithms are performed. The co-simulation environment enables to automatically generate training data without the need to specify predefined scenarios. In this way, the CAV's machine learning modules learn within a comprehensive environment while neither requiring driving tests nor gathering real-world test data. Basically, the toolchain is set up as described in Chapter 3, the CAV prototype drives in the digital environment and produces training data. Because of the omniscient view that can be ensured by simulation, the data can be prepared according to the requirements of the machine learning algorithm. Further, the trained algorithms can be tested and embedded in the CAV prototype that includes implemented driving functions. Consequently, the performance of the machine learning algorithms can be evaluated by the toolchain as well. Thus, based on the fact that the simulation can be performed in parallel and faster as real time, the approach can be considered as a helpful enhancement for these applications.

Additionally, the PiL approach gives us the ability to test these algorithms incorporated in a real-world CAV prototype, particularly, without endangering test drivers or hardware. This proceeding provides a closer step to realistic evaluation, but still, possesses all the advantages of the PiL approach introduced in Section 3.2.

Within the Ko-HAF project, the collaboration was limited to a development team and the author of this dissertation. Therefore, the application was tailored to the demands of the project. Certainly, the toolchain and the PiL approach, contributed to the successful demonstration of the CAV prototype at the final event. The advantage to constantly test newly implemented functions, first in the simulation, then in by the PiL approach can be considered as a helpful tool. Aspects throughout the entire "Sense, Plan, Act" scheme can be evaluated and improved. The discrepancy between simulation and driving tests is reduced by the PiL approach. Thus, the leap between solely virtual testing and actual driving tests can be extended by this helpful approach, building a bridge between these test environments.

In summary, the presented results in this chapter are exemplary, but the methodology is already used for further applications. Selected results, which were presented, rest upon the requirements that motivated this dissertation. Especially, disturbance injection and, particularly, the impacts of map errors and differences of human driving behavior were emphasized on. Further, the strengths of the PiL approach in testing critical scenarios that are dangerous to perform with test drivers were pointed out. In collaboration with the function development department, in particular, the developers that implemented the localization as well as the driving functions, the approach was implemented. The benefit of testing reproducible scenarios, which was provided by the PiL approach, can be considered a huge advantage. Additionally, the capability of testing the same reproducible scenarios in the virtual environment as well as on the proving ground provided a helpful asset during the development of the CAV prototype's driving functions. It is nearly impossible to achieve reproducible results by performing driving tests with test drivers. These slight, but non-negligible differences in the test execution pose a challenge that can be solved by the PiL approach. The tests can be parameterized and equally triggered, which guarantees that PiL tests do not vary significantly from solely virtually performed test executions. It should be noted that the PiL approach distinguishes itself from ViL by using the multi-agent traffic simulation for controlling other test participants' driving behavior. Of course, the method is able to operate with predefined trajectories for test participants as well. But the major advantage of the approach is the interaction between the CAV prototype and the agents. Only the parameters of the agents need to be predefined. After that, the agents operate according to their driver behavior models. The approach can be seen as an enhancement of the ViL approach. Additionally, if an initial situation is predefined for a test, the variation of the agents' parameters allows for further investigations. In conjunction with the co-simulation environment and the systematic approach enabled by the toolchain, this overall research effort significantly contributes to the field of CAV testing.

5 Conclusion and Future Work

This dissertation was motivated by the challenges that arise in the field of CAV development and testing. The automotive industry faces major obstacles regarding this topic, where a vast driving distance would need to be covered to test and validate these systems considering state-of-the-art methodologies. It should be noted that this problem originates from shifting the responsibility from human drivers to automation modules. While the current focus mostly lies on SAE level 3 systems, level 4 and 5 CAV technologies will be the center of interest in the near future.

The general complexity of these systems as well as the challenges that occur, due to the domain CAVs are supposed to operate in, raise the demand for new testing methodologies. In particular, the interactions between CAVs and manually driven vehicles pose a difficult obstacle that requires intensive investigation. Thus, non-normative driving behavior caused by human drivers is an important field of research. Additionally, used precision maps for localization, extension of the perception range, and driving strategy decisions, can contain errors and differ in a larger, heterogeneous CAV fleet.

State-of-the-art methodologies for testing and validating CAVs rely strongly on expert opinions and requirements engineering. Certainly, these systems are far too complex for this course of action. Anyway, approaches, such as the ISO-26262 and SOTIF, can be considered as helpful fundamentals for future testing of CAVs. In addition, state-of-the-art research projects, such as PEGASUS, Ko-HAF, and ENABLE-S3 work on methodologies to overcome these challenges. Despite of these fundamentals, the requirements for CAV testing are demanding and the vast driving distance that would need to be covered for validation urge for simulation-based approaches. Simulation is widely used for development and testing in almost every industry branch because of its manifold advantages including testing without requiring hardware or endangering users and components. Additionally, simulations can be performed faster than real time, which enables to run a vast amount of tests in a short amount of time and provides an omniscient view on all state space variables.

For all of the afore mentioned reasons, a generic simulation-based toolchain that supports CAV development accompanying tests is proposed. When considering the demanding validation process, the CAV needs to be tested up to a certain maturity level even before this process is initiated. Otherwise, there are going to be too many complications with immature driving functions, which causes a lot of supplementary effort in the validation process.

The proposed methodology aims at providing an applicable proceeding that supports the CAV driving function developer according to his specific demands. The toolchain's setup is generic as well as parameterizable and can be utilized for a variety of CAV applications that require comprehensive testing. The toolchain provides a systematic approach for automatically verifying and identifying critical scenarios. Particularly, the identification of yet unknown scenarios is a major contribution of the methodology. In order to allow the verification and identification of critical scenarios, use cases are modeled by a parameter space based scenario description and, thereby, concrete scenarios are generated, which are either selected by experts or chosen by a parameter variation module. Concrete scenarios are sufficient as an input for performing simulation runs. The toolchain allows for injecting and parameterizing disturbances, such as sensor errors, map errors, non-normative human driving behavior, etc.

The centerpiece of the toolchain consists of a coupled simulation environment combining the advantages of a vehicle dynamics, a traffic, and a cooperation simulation. The simulations are coupled and operate as a so-called co-simulation. The vehicle dynamics simulation includes a digital CAV prototype. The test participants are controlled by the traffic simulation. Cooperative features can be included in the cooperation simulation. The simulation runs are evaluated by metrics tailored for this dissertation, which aim at assessing safety related issues and impacts on traffic quality. The verified and identified scenarios can be extended by further test environments, such as XiL methods, test benches, mixed reality, and driving tests, to increase the validity of the test results. An analysis step consists of a test result sheet that enables the CAV developer to access the data and review the test results.

A novel approach has been developed in the course of this thesis. Prototype-in-the-Loop combines the traffic simulation with a real-world CAV prototype driving on real-world proving ground. This approach aims at using the advantages provided by the traffic simulation and combines them with a real-world CAV prototype.

The first part of the Chapter 4 shows the MiL results considering different disturbances, e.g., sensor errors, map errors, and aggressive driving behavior of other test participants. The second part‘s focal point presents the abilities of the PiL method. The PiL tests were performed on the proving ground in Dudenhofen near Frankfurt (Germany). The traffic simulation was incorporated in the Ko-HAF CAV prototype and, especially, safety critical tests were performed to show one of the major advantages of the PiL approach. All in all, the proposed methodology provides a lot of contributions in the field of CAV testing. The research objectives for this dissertation to establish a systematic methodology for improving simulation-based CAV development and testing, enable the assessment and identification of critical scenarios systematically and automatically to simultaneously evaluate the effects of CAV driving functions on ODDs and vice versa for different DOIs, and to feature the investigation of disturbance influences on CAVs and ODDs, could be comprehensively met.

Regarding future work, more application and operating experience with the simulation-based toolchain should be gathered. It will be interesting to see how the toolchain supports the driving function development departments in the future. SAE level 3 systems, such as a highway chauffeur can be tested by this approach, which is currently in the focus of automotive manufacturers. Because of the generic setup of the toolchain, the testing of level 4 and 5 systems can be performed as soon as these systems become the focal point of CAV development.

Another aspect that requires further investigation are studies about the validity of vehicle dynamics and driver behavior models. Certainly, the granularity as well as the validity of the models need to be considered for the assessment of the simulation results. Further, boundaries, frequency of occurrence, and plausibility of parameters as well as parameter combinations require investigation and operating experience. An estimation of the confidence level for the run through the toolchain should be performed, where the afore mentioned aspects need to be considered. Additionally, valuable information can be gathered by comparing the tests performed by the co-simulation and the PiL approach. The major difference between the co-simulation and the PiL approach is the substitution of the vehicle dynamics model with a real-world CAV prototype. Thus, it is possible to gain insights on the simulation model quality comparing the test runs with each other.

Moreover, the utilization of the toolchain enables to perform statistical studies about the influences of CAVs on large scale traffic systems. Studies on traffic quality improvement, benefits of CAVs operating in traffic systems, and the influences of CAV fleet routing can be investigated. Further, knowledge about the impacts of certain disturbances and errors can be studied. For example, map errors that differ in their characteristics can be evaluated and the significance of specific errors could be determined by operating the toolchain. Additionally, the errors can be parameterized and varied to analyze influences of errors on traffic systems.

The toolchain is also capable of testing cooperative features, such as cooperative merging and trajectory planning. The assessment of these systems becomes more and more important and the toolchain is already applied for evaluating vehicles with cooperative features in other research projects. While the general proceeding of operating the toolchain remains the same, the specific aspects are exchanged and applied for the respective purpose.

In addition, each aspect of the toolchain opens a research topic of its own. For example, the parameter variation module allows for automatically generating concrete scenarios. The method, how the parameters are sampled properly, is a difficult task. The use of criticality heat maps is one proceeding that can be applied. The toolchain could be used to create criticality heat maps, as well. By identifying critical scenarios for certain parameter combinations with respect to the evaluation metrics, data for criticality heat maps can be generated.

The traffic quality metrics combination applied in this dissertation is based on training and test data generated by the traffic simulation. The general proceeding could be used for other evaluation metrics in different ODDs, such as rural roads and urban areas as well. Broadly speaking, the approach is feasible for training various evaluation metrics that are utilized for CAVs and other industrial domains. New metric concepts, e.g., Mobileye's "Responsibility-Sensitive Safety [151]" or Nvidia's "Safety Force Field [152]", can be applied within the framework of this toolchain.

Generating training data for Artificial Intelligence (AI) approaches is another aspect the toolchain and, particularly, the co-simulation environment can be utilized for. Having the opportunity to guide a vehicle equipped with AI modules through an interacting traffic environment enables to produce training data. Further, the trained modules can be tested by the simulation-based toolchain as well, or rather, test data can be generated. Thus, by using the co-simulation framework, the AI algorithms can be trained, tested, and evaluated. If the AI module, for example, aims at learning a driving strategy, this module can be inserted in a digital prototype and, thereby, assessed with the methodology proposed in this dissertation. In addition, the PiL approach allows for testing this AI module in a more realistic environment without endangering test drivers or hardware. Currently, this proceeding is applied in our work group. The driving strategy module is trained by data generated by the traffic simulation tool and learns how to guide a CAV on highways. Due to the fact that the proceeding is generic, this course of action can be used for other AI modules and applications as well. Even for reinforcement learning, where the AI guides the CAV by exploring alternative actions, the co-simulation framework can be applied without damaging the CAV prototype or harming test drivers. After the AI module has reached a certain maturity level, the PiL approach is used to train this algorithm in a more valid environment and, still, without endangering other test drivers. If the AI algorithm shows a satisfying performance, the tests can be conducted with soft target robots or other test drivers.

At last, the co-simulation framework could be applied for predicting future scenario developments, if the CAV was overruled by a test driver because of a dangerous situation. Considering CAV tests in a public traffic system, usually a trained test driver observes the CAV's actions and takes over in dangerous situations. At this particular point in time, the CAV driving function does not control the vehicle motion anymore and the scenario progresses differently as if it was controlled by the automation modules. The recorded data of this scenario can be used and serves as an initial condition for the co-simulation environment. The future motion of the CAV without the overtaking of the driver could be simulated, while the other test participants' future motions are controlled according to the driver behavior models of the traffic simulation. Thus, the behavior of the CAV in critical scenarios and ongoing evolvements can be tested and assessed.

Bibliography

[1] Statistisches Bundesamt, Wiesbaden. *Unfallentwicklung auf deutschen Straßen 2017. Begleitmaterial zur Pressekonferenz am 12. Juli 2018 in Berlin*. Berlin, Germany, 2018.

[2] W. Wachenfeld and H. Winner. Die Freigabe des autonomen Fahrens. In M. Maurer, J. C. Gerdes, B. Lenz, and H. Winner, editors, *Autonomes Fahren - Technische, rechtliche und gesellschaftliche Aspekte* , chapter 21, pages 439–464. Springer-Verlag GmbH, Berlin Heidelberg, Germany, 1st edition, 2015.

[3] M. Bücker, B. Kramer, E. Böde, S. Vander Maelen, and M. Fränzle. Identifikation von Automationsrisiken hochautomatisierter Fahrfunktionen in PEGASUS. In *AAET - Automatisiertes und vernetztes Fahren*, pages 315–329, Brunswick, Germany, 2019.

[4] SAE International. *Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems*. SAE Standard J3016.201401, 2014.

[5] J. Vetter. *Highly Automated Driving - Validation and Test*. Presented at the Ko-HAF final event, Dudenhofen, Germany, 2018.

[6] S. Boschert and R. Rosen. Digital Twin - The Simulation Aspect. In P. Hehenberger and D. Bradley, editors, *Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and their Designers*, chapter 5, pages 59–74. Springer International Publishing, Switzerland, 1st edition, 2016.

[7] S. Hallerbach, U. Eberle, and F. Köster. Absicherungs- und Bewertungsmethoden für kooperative und hochautomatisierte Fahrzeuge. In *AAET - Automatisiertes und vernetztes Fahren*, pages 368–384, Brunswick, Germany, 2017.

[8] S. Hallerbach, Y. Xia, U. Eberle, and F. Köster. Simulation-Based Identification of Critical Scenarios for Cooperative and Automated Vehicles. *SAE International Journal of Connected and Automated Vehicles*, 1(2):93–106, 2018.

[9] U. Eberle. *Automated Driving - Status and Challenges*. Presented at Automotive Photonics, Stuttgart-Ditzingen, Germany, 2016.

[10] U. Eberle, O. Schädler, S. Hallerbach, M. Büker, S. Gerwinn, T. Brade, C. Amersbach, and PEGASUS. *Automated Driving and Development Processes - A Brief Look Beyond*. Presented at the PEGASUS-Symposium, Aachen, Ger-

many, 2017.

[11] R. Isermann, editor. *Fahrdynamik-Regelung - Modellbildung, Fahrerassistenzsysteme, Mechatronik*. Friedr. Vieweg und Sohn Verlag, GWV Fachverlage GmbH, Wiesbaden, Germany, 1st edition, 2006.

[12] A. R. Plummer. Model-in-the-Loop Testing. *Proceedings of the Institution of Mechanical Engineers Part I: Journal of Systems and Control Engineering*, 220(3):183–199, 2006.

[13] J. Schäuffele and T. Zurawka. *Automotive Software Engineering - Grundlagen, Prozesse, Methoden und Werkzeuge effizient einsetzen*, chapter 5.5.1, pages 298–300. Springer Vieweg, Wiesbaden, Germany, 6th edition, 2016.

[14] S. Otten, J. Bach, C. Wohlfahrt, C. King, J. Lier, H. Schmid, S. Schmerler, and E. Sax. Automated Assessment and Evaluation of Digital Test Drives. In C. Zachäus, B. Müller, and G. Meyer, editors, *Advanced Microsystems for Automotive Applications 2017 - Smart Systems Transforming the Automobile*, pages 189–199. Springer International Publishing AG, Cham, Switzerland, 1st edition, 2018.

[15] H. K. Fathy, Z. S. Filipi, J. R. Hagena, and J. L. Stein. Review of Hardware-in-the-Loop Simulation and Its Prospects in the Automotive Area. In K. Schum and A. F. Sisti, editors, *Modeling and Simulation for Military Applications*, volume 6228, pages 117 – 136. International Society for Optics and Photonics, SPIE, 2006.

[16] M. Feilhauer and J. Haering. Current Approaches in HiL-Based ADAS Testing. *SAE Int. J. Commer. Veh.*, 9(2):63–69, 2016.

[17] T. Bock, M. Maurer, F. v. Meel, and T Müller. Vehicle in the Loop - Ein innovativer Ansatz zur Kopplung virtueller mit realer Erprobung. *ATZ - Automobiltechnische Zeitschrift*, 110(2):10–16, 2008.

[18] T. Bock, K.-H. Siedersberger, and M. Maurer. Vehicle in the Loop - Augmented Reality Application for Collision Mitigation Systems. In *Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'05)*, Wien, Austria, 2005.

[19] G. Berg. *Das Vehicle in the Loop - Ein Werkzeug für die Entwicklung und Evaluation von sicherheitskritischen Fahrerassistenzsystemen*. PhD thesis, Universität der Bundeswehr, Munich, Germany, 2014.

[20] S. Hakuli and M. Krug. Virtuelle Integration. In H. Winner, S. Hakuli, F. Lotz, and C. Singer, editors, *Handbuch Fahrerassistenzsysteme - Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*, chapter 8, pages 125–138. Springer Fachmedien, Wiesbaden, Germany, 3rd edition, 2015.

[21] J. Dick, E. Hull, and K. Jackson. *Requirements Engineering*. Springer International Publishing, Cham, Switzerland, 4th edition, 2017.

[22] K. M. Adams. *Non-functional Requirements in Systems Analysis and Design - Topics in Safety, Risk, Reliability and Quality*, volume 28. Springer International Publishing, Cham, Switzerland, 1st edition, 2015.

[23] A. Bartels, U. Eberle, and A. Knapp. *Deliverable D2.1. - System Classification and Glossary - Version 1.2*. Adaptive Consortium, Wolfsburg, Germany, 2015.

[24] U.S. Software System Safety Working Group. Adaptive Cruise Control System Overview. In *5th Meeting of the U.S. Software System Safety Working Group*, Anaheim, California, USA, 2005.

[25] J.C. Hayward. Near miss determination through use of a scale of danger. In *51st Annual Meeting of the Highway Research Board*, pages 24–34, Washington D.C., USA, 1972.

[26] S. Hallerbach, U. Eberle, and F. Köster. The challenges of releasing cooperative and highly automated vehicles - A look beyond functional requirements. In *AmE - Automotive meets Electronics, GMM-Fachbericht, VDE*, pages 102–106, Dortmund, Germany, 2017.

[27] Anthony Best Dynamics Ltd. *ADAS development using ABD driving robots*. ABD Application Note AN6311 - Issue 3. Accessed: 12.01.2020 `https://www.abdynamics.com/resources/files/AN-6032-ADAS-Testing.pdf`.

[28] J. Kramer. *Kollektive Energieeffizienz auf mehrspurigen Straßen*. PhD thesis, Karlsruher Institut für Technologie (KIT), Karlsruhe, Germany, 2018.

[29] GIDAS - German In-Depth Accident Study. Accessed: 12.01.2020. `https://www.gidas.org/`.

[30] Next Generation SIMulation Fact Sheet. Accessed: 12.01.2020. `https://www.fhwa.dot.gov/publications/research/operations/its/06135/index.cfm`.

[31] ISO-26262 - Road vehicles - Functional safety. Standard, International Organization for Standardization, 2011.

[32] IEC 61508 - Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems (E/E/PE, or E/E/PES). Standard, International Electrotechnical Commission, 2010.

[33] M. Hillenbrand. *Funktionale Sicherheit nach ISO 26262 in der Konzeptphase der Entwicklung von Elektrik/Elektronik Architekturen von Fahrzeugen*. PhD thesis, Karlsruher Institut für Technologie (KIT), Karlsruhe, Germany, 2012.

[34] B.J. Czerny, J. D'Ambrosio, and Debouk R. *Introduction to the ISO CD 26262 Functional Safety Standard*. Presented at the 26th International System Safety Conference, Vancouver, Canada, 2008.

[35] Embitel - Understanding How ISO 26262 ASIL is Determined for Automotive Applications. Accessed: 12.01.2020. https://www.embitel.com/blog/embedded-blog/understanding-how-iso-26262-asil-is-determined- for-automotive-applications.

[36] U. Wilhelm, S. Ebel, and A. Weitzel. Funktionale Sicherheit und ISO 26262. In H. Winner, S. Hakuli, F. Lotz, and C. Singer, editors, *Handbuch Fahrerassistenzsysteme - Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*, chapter 6, pages 85–103. Springer Fachmedien, Wiesbaden, Germany, 3rd edition, 2015.

[37] W. Wachenfeld. *How Stochastic can Help to Introduce Automated Driving*. PhD thesis, Technische Universität Darmstadt, Darmstadt, Germany, 2017.

[38] H. Winner. Quo vadis, FAS? In H. Winner, S. Hakuli, F. Lotz, and C. Singer, editors, *Handbuch Fahrerassistenzsysteme - Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*, chapter 62, pages 1167–1186. Springer Fachmedien, Wiesbaden, Germany, 3rd edition, 2015.

[39] B. Milke. *Welcome Address*. Presented at the Ko-HAF final event, Dudenhofen, Germany, 2018.

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

[41] F. Köster. *Analyse von Simulationsmodellen mit Methoden des Knowledge Discovery in Databases*. PhD thesis, Carl von Ossietzky Universität Oldenburg, Oldenburg, Germany, 2001.

[42] U. Vogel and M. Sonnenschein. *Lecture Notes: Modellbildung und Simulation ökologischer Systeme*, page 33. Carl von Ossietzky Universität Oldenburg - Department of Environmental Informatics, Oldenburg, Germany, WS2016/17.

[43] H.-J. Bungartz, S. Zimmer, M. Buchholz, and D. Pflüger. *Modellbildung und Simulation - Eine anwendungsorientierte Einführung*, chapter 1.1. Springer Verlag, Berlin Heidelberg, Germany, 2nd edition, 2013.

[44] R. Rajamani. *Vehicle Dynamics and Control - Lateral Vehicle Dynamics*, chapter 2, pages 15–46. Springer US, Boston, MA, USA, 2nd edition, 2012.

[45] J. Schucker and P. Hoffmann. Nonlinear control methods for combined lateral and longitudinal control. In *AmE - Automotive meets Electronics, GMM-Fachbericht, VDE*, pages 20–25, Dortmund, Germany, 2017.

[46] M. Glöckler. *Simulation mechatronischer Systeme - Grundlagen und technische Anwendung- Kapitel 4 - Numerische Integrationsverfahren*, pages 85–108. Springer Vieweg, Wiesbaden, Germany, 1st edition, 2014.

[47] M. Rabe, S. Spieckermann, and S. Wenzel. *Verifikation und Validierung für die Simulation in Produktion und Logistik - Vorgehensmodelle und Techniken*. VDI-Buch. Springer-Verlag, Berlin Heidelberg, Germany, 1st edition, 2008.

[48] M. Glöckler. *Simulation mechatronischer Systeme - Grundlagen und technische Anwendung*. Springer Vieweg, Wiesbaden, Germany, 1st edition, 2014.

[49] J. S. Sichman, R. Conte, and N. Gilbert. *Multi-Agent Systems and Agent-Based Simulation - First International Workshop, MABS '98, Paris, France, July 4-6, 1998, Proceedings*. Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin Heidelberg, Germany, 1st edition, 1998.

[50] German Aerospace Center - Institute of Transportation Systems. SUMO - Simulation of Urban MObility - Traffic Management and Other Structures. Accessed: 12.01.2020. `https://sumo.dlr.de/userdoc/SUMO_User_Documentation.html#network_building`.

[51] German Aerospace Center - Institute of Transportation Systems. SUMO - Simulation of Urban MObility. Accessed: 12.01.2020. `https://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/`.

[52] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner. SUMO (Simulation of Urban MObility) An open-source traffic simulation. In *Proceedings of the 4th Middle East Symposium on Simulation and Modelling (MESM2002)*, Sharjah, UAE, 2002.

[53] J. Erdmann. Lane-Changing Model in SUMO. In *SUMO2014 - Proceedings of the SUMO2014 Modeling Mobility with Open Data*, Reports of the DLR-Institute of Transportation Systems Proceedings Vol. 24, pages 77–88, Berlin, Germany, 2014.

[54] DLR-SUMO Publications. Accessed: 12.01.2020. `https://sumo.dlr.de/wiki/Publications`.

[55] PEGASUS - Project for the establishment of generally accepted quality criteria, tools and methods as well as scenarios and situations for the release of highly-automated driving functions. Accessed: 12.01.2020. `https://www.pegasusprojekt.de/`.

[56] Ko-HAF - Kooperatives und hochautomatisiertes Fahren. Accessed: 12.01.2020. `https://www.ko-haf.de/`.

[57] ENABLE-S3 - European Initiative to Enable Validation for Highly Automated Safe and Secure Systems. Accessed: 12.01.2020. https://www.enable-s3.eu/.

[58] T. Form. *General introduction to PEGASUS and Opening of the exhibition*. Presented at the PEGASUS - Symposium, Aachen, Germany, 2017.

[59] J. Mazzega, F. Köster, K. Lemmer, and T. Form. Absicherung hochautomatisierter Fahrfunktionen. *ATZ - Automobiltechnische Zeitschrift*, 10/2016:48–52, 2016.

[60] D. Lipinski. *Scenario analysis and quality measures*. Presented at the PEGASUS - Symposium, Aachen, Germany, 2017.

[61] F. Köster, T. Form, K. Lemmer, and J. Plättner. Wie gut müssen automatisierte Fahrzeuge fahren - PEGASUS. In *AAET - Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme für Transportmittel*, pages 292–300, Brunswick, Germany, 2016.

[62] K. Preuk and Caroline Schießl. Menschliche Leistungsfähigkeit als Gütekriterium für die Zulassung automatisierter Fahrzeuge: Methode zur Ermittlung der Grenzen menschlicher Leistungsfähigkeit. In *VDI-Fachtagung - Der Fahrer im 21. Jahrhundert*, Brunswick, Germany, 2017.

[63] A. Pütz, A. Zlocki, J. Bock, and L. Eckstein. System validation of highly automated vehicles with a database of relevant traffic scenarios. In *12th ITS European Congress*, Strasbourg, France, 2017.

[64] H. Schittenhelm. *Testing of level-3 Systems - stepping through the current PEGASUS approach*. Presented at the PEGASUS - Symposium, Aachen, Germany, 2017.

[65] W. Wachenfeld. *Ergebnisreflexion: Der Weg zu generell akzeptierten Ergebnissen*. Presented at the PEGASUS - Symposium, Aachen, Germany, 2017.

[66] Waymo Safety Report - On the Road to Fully Self-Driving. Accessed: 12.01.2020. https://storage.googleapis.com/sdc-prod/v1/safety-report/Safety%20Report%202018.pdf.

[67] PEGASUS Consortium. *Test Concept and Test Case Allocation*. Presented at the PEGASUS - Symposium, Aachen, Germany, 2017.

[68] PEGASUS Consortium. *Sensor Simulation Models*. Presented at the PEGASUS - Symposium, Aachen, Germany, 2017.

[69] PEGASUS Consortium. *Software-in-the-Loop*. Presented at the PEGASUS - Symposium, Aachen, Germany, 2017.

[70] PEGASUS Consortium. *Hardware-in-the-Loop Simulation*. Presented at the PEGASUS - Symposium, Aachen, Germany, 2017.

[71] PEGASUS Consortium. *Proving Ground - Generic Approach and Control Center*. Presented at the PEGASUS - Symposium, Aachen, Germany, 2017.

[72] PEGASUS Consortium. *Critical Scenarios for and by the HAD (L3)*. Presented at the PEGASUS - Symposium, Aachen, Germany, 2017.

[73] A. Hohm. *Ko-HAF - Safety by Cooperation*. Presented at the Ko-HAF final event, Dudenhofen, Germany, 2018.

[74] J. Vetter. *Coordination Tools for HAD-System Testing*. Presented at the Ko-HAF final event, Dudenhofen, Germany, 2018.

[75] H. Znamiec. *Functional Testing of Automated Driving on Highways*. Presented at the Ko-HAF final event, Dudenhofen, Germany, 2018.

[76] M. Burg. *Simulation as Tool*. Presented at the Ko-HAF final event, Dudenhofen, Germany, 2018.

[77] A. Schaermann, A. Rauch, N. Hirsenkorn, T. Hanke, R. Rasshofer, and E. Biebl. Validation of vehicle environment sensor models. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 405–411, Los Angeles, CA, USA, 2017.

[78] H. Znamiec, B. Reuber, R. Henze, and F. Kücükay. A Method For The Efficient Testing OF New Automated Driving Functions. In *Future Active Safety Technology Toward zero traffic accidents - FAST-Zero*, Gothenburg, Sweden, 2017.

[79] H. Znamiec, B. Reuber, R. Henze, and F. Kücükay. Method for the efficient testing of future generations of automated driving functions. In *IEDAS - Activesafety and Automated Driving, 3. Interdisciplinary Expert Dialogue*, Essen, Germany, 2017.

[80] ENABLE-S3 - European initiative to Enable Validation for Highly Automated Safe and Secure Systems. Accessed: 12.01.2020. `https://www.enable-s3.eu/domains/`.

[81] S. Settelmeier, R. Hettel, J. Ibanez-Guzman, G. Russo, B. Goncalves, and B. Boeddeker. *D2 D1.1.2 v1 - System under test requirements and test system requirements from the automotive domain*. Enable-S3 Consortium, 2016.

[82] W. Griggs, G. Russo, H. Esen, A. Molin, E.M. Fialho, B. Goncalves, F. Soares, R. Hettel, J. Ibanez-Guzman, J. Holzinger, M.v.d. Gaar, and S. Roubtsov. *D4 (D1.4) - D1.1.3 v1 - Evaluation result of the automotive domain*. Enable-S3 Consortium, 2018.

[83] S. Roubtsov, R. Ekkel, and J. Stevens. *D2.2.1 - Metrics and measurements for*

evaluation report. Enable-S3 Consortium, 2017.

[84] D. Nickovic, W. Herzner, E. Moehlmann, S. Gerwinn, G. Garcia Padilla, F. Diegmann, S. Hakki, and C. Snook. *D3.2.2 v1 - V&V Methodology*. Enable-S3 Consortium, 2017.

[85] D. Nickovic et al. *D46 (D3.10) - D3.2.2 v2 - V&V-Methodology*. Enable-S3 Consortium, 2018.

[86] C. Schwarzl et al. *D3.4.1 - Platform specification*. Enable-S3 Consortium, 2017.

[87] R. Kastner. Modellbildung und Simulation eines kooperativen Stauassistenten über ein Co-Simulationsframework. Master's thesis, TU Graz - Institute of Automotive Engineering, Graz, Austria, 2017.

[88] M. R. Zofka, S. Ulbrich, R. Kohlhaas, A. Rönnau, R. Dillmann, and J. M. Zöllner. Towards an Mixed-Reality Multi-User Framework for the Evaluation and Training of Autonomous Vehicles. In *RSSRail 2017: International Conference on Reliability, Safety, and Security of Railway Systems. Modelling, Analysis, Verification, and Certification. vol. 10598*, Pistoia, Italy, 2017.

[89] M. R. Zofka, M. Essinger, T. Fleck, R. Kohlhaas, and J. M. Zöllner. The Sleepwalker Framework: Verification and Validation of Autonomous Vehicles by Mixed Reality LiDAR Stimulation. In *2018 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, pages 151–157, Brisbane, Australia, 2018.

[90] A. Pütz, A. Zlocki, J. Küfen, J. Bock, and L. Eckstein. Database Approach for the Sign-Off Process of Highly Automated Vehicles. In *25th International Technical Conference on the Enhanced Safety of Vehicles (ESV) National Highway Traffic Safety Administration*, Detroit, USA, 2017.

[91] 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 *2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*, Hawaii, USA, 2018.

[92] S. Krause, N. Motamedidehkordi, S. Hoffmann, and F. Busch. Mikroskopische Simulation von automatisierten Fahrzeugen zur Ermittlung der Wirkungen auf die Kapazität von Autobahnen. *Straßenverkehrstechnik*, 10/2016, 2017.

[93] F. Busch, S. Krause, N. Motamedidehkordi, S. Hoffmann, P. Vortisch, and M. Hartmann. *Auswirkungen des teil- und hochautomatisierten Fahrens auf die Kapazität der Fernstraßeninfrastruktur*. Forschungsvereinigung Automobiltechnik e.V. - FAT-Schriftenreihe 296, VDA - Verband der Automobilindus-

trie, 2017.

[94] J. Kaths and S. Krause. *Integrated simulation of microscopic traffic flow and vehicle dynamics*. IPG Apply and Innovate 2016, Karlsruhe, Germany, 2016.

[95] P. Junietz, W. Wachenfeld, V. Schönemann, K. Domhardt, W. Tribelhorn, and H. Winner. Gaining Knowledge on Automated Driving's Safety - The Risk-Free VAAFO Tool. In H. Waschl, I. Kolmanovsky, and F. Willems, editors, *Control Strategies for Advanced Driver Assistance Systems and Autonomous Driving Functions - Development, Testing and Verification*, volume 476, chapter 3, pages 47–65. Springer International Publishing, Cham, Switzerland, 1st edition, 2019.

[96] H. Winner, W. Wachenfeld, and P. Junietz. Validation and Introduction of Automated Driving. In H. Winner, G. Prokop, and M. Maurer, editors, *Automotive Systems Engineering II*, chapter 8, pages 177–196. Springer International Publishing, Cham, Switzerland, 1st edition, 2018.

[97] F. Schuldt, A. Reschka, and M. Maurer. A Method for an Efficient, Systematic Test Case Generation for Advanced Driver Assistance Systems in Virtual Environments. In H. Winner, G. Prokop, and M. Maurer, editors, *Automotive Systems Engineering II*, chapter 7, pages 147–175. Springer International Publishing, Cham, Switzerland, 1st edition, 2018.

[98] H. Waschl, R. Schmied, D. Reischl, and M. Stolz. A Virtual Development and Evaluation Framework for ADAS - Case Study of a P-ACC in a Connected Environment. In H. Waschl, I. Kolmanovsky, and F. Willems, editors, *Control Strategies for Advanced Driver Assistance Systems and Autonomous Driving Functions - Development, Testing and Verification*, volume 476, chapter 6, pages 107–131. Springer International Publishing, Cham, Switzerland, 1st edition, 2019.

[99] A. Lauber, H. Guissouma, and E. Sax. Virtual Test Method for Complex and Variant-Rich Automotive Systems. In *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pages 1–8, Madrid, Spain, 2018.

[100] C. King, J. Bach, S. Otten, and E. Sax. Identifikation von Fahrszenarien während einer virtuellen Testfahrt. In M. Eibl and M. Gaedke, editors, *INFORMATIK 2017*, pages 1511–1516. Gesellschaft für Informatik, Bonn, Germany, 2017.

[101] I.B. Jemaa, D. Gruyer, and S Glaser. Distributed simulation platform for cooperative ADAS testing and validation. In *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 77–82, Rio de Janeiro, Brazil, 2016.

[102] P. Koopman and M. Wagner. Challenges in Autonomous Vehicle Testing and

Validation. *SAE Int. J. Trans. Safety*, 4(1):15–24, 2016.

[103] P. Koopman and M. Wagner. Toward a Framework for Highly Automated Vehicle Safety Validation. In *SAE International, World Congress Experience*, pages 1–13, Detroit, USA, 2018.

[104] P. Koopman. Practical Experience Report: Automotive Safety Practices vs. Accepted Principles. In B. Gallina, A. Skavhaug, and F. Bitsch, editors, *Computer Safety, Reliability, and Security*, pages 3–11. Springer International Publishing, Cham, Switzerland, 2018.

[105] P. Koopman, A Kane, and J. Black. Credible Autonomy Safety Argumentation. In *Safety-Critical Systems Club, 27th Safety-Critical Systems Symposium*, pages 1–27, Bristol, UK, 2019.

[106] General Motors 2018 Self-Driving Safety Report. Accessed: 12.01.2020. `https://www.gm.com/content/dam/company/docs/us/en/gmcom/gmsafetyreport.pdf`.

[107] Virtual Vehicle - Passion for electrified, automated and highly-connected vehicles. Accessed: 12.01.2020. `https://www.v2c2.at/`.

[108] German Aerospace Center - Anwendungs-plattform Intelligente Mobilität. Accessed: 12.01.2020. `https://www.dlr.de/ts/Portaldata/16/Resources/aim/DLR-TS_AnwendungsplattformIntelligenteMobilitaet.pdf`.

[109] Testfeld Niedersachsen für automatisierte und vernetzte Mobilität. Accessed: 12.01.2020. https://verkehrsforschung.dlr.de/de/projekte/testfeld-niedersachsen-fuer-automatisierte-und-vernetzte-mobilitaet.

[110] Testfeld Autonomes Fahren Baden-Württemberg - Reallabor für Forschung und Innovation. Accessed: 12.01.2020. `https://taf-bw.de/`.

[111] G. Bagschik, T. Menzel, A. Reschka, and M. Maurer. Szenarien für Entwicklung, Absicherung und Test von automatisierten Fahrzeugen. In *11. Workshop Fahrerassistenzsysteme und automatisiertes Fahren, FAS 2017, Uni-DAS e.V.*, pages 125–135, Walting, Germany, 2017.

[112] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer. Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 982–988, Las Palmas, Spain, 2015.

[113] M. Grindal, J. Offutt, and S. F. Andler. Combination Testing Strategies: A Survey. *Journal of Softw. Test., Verif. Reliab.*, 15(3):167–199, 2005.

[114] C. Amersbach and H. Winner. Funktionale Dekomposition - Ein Beitrag zur

Überwindung der Parameterraumexplosion bei der Validation von höher automatisiertem Fahren. In *12. Workshop Fahrerassistenzsysteme und automatisiertes Fahren, FAS 2018, Uni-DAS e.V.*, pages 27–36, Walting, Germany, 2018.

[115] T. Mugur. *Chasing critical situations in large parameter spaces*. Presented at Autonomous Vehicle Test and Development Symposium, Stuttgart, Germany, 2018.

[116] MathWorks - Matlab/Simulink. Accessed: 12.01.2020. `https://www.mathworks.com/products/simulink.html`.

[117] M. Dupuis et al. *OpenDrive Format Specification, Rev. 1.4*. VIRES-Simulationstechnologie GmbH, 2015.

[118] L. Zaruba. Development and evaluation of a simulation environment for co-operative and highly automated vehicles. Master's thesis, HAN University of Applied Sciences - Faculty of Engineering - Institute of Automotive, Nijmegen, Netherlands, 2017.

[119] M. Harr, J. Janosovits, S. Wirges, and C. Stiller. Fast and Robust Vehicle Pose Estimation by Optimizing Multiple Pose Graphs. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1707–1714, Cambridge, UK, 2018.

[120] R. Schubert, E. Richter, and G. Wanielik. Comparison and Evaluation of Advanced Motion Models for Vehicle Tracking. In *11th International Conference on Information Fusion*, pages 1–6, Cologne, Germany, 2008.

[121] V. Lizenberg, S. Knapp, R. Mannale, V. Wendel, and F. Köster. Simulations-basierte Bewertungs- und Vergleichsmethodik für Abstimmungsverfahren in kooperativen Fahrfunktionen. In *AAET - Automatisiertes und vernetztes Fahren*, pages 49–65, Brunswick, Germany, 2019.

[122] Transport Research Board National Research Council. *Highway Capacity Manual*. Washington, USA, 2000.

[123] C. Oh, A. Tok, and S.G. Ritchie. Real-Time Freeway Level of Service Using Inductive-Signature-Based Vehicle Reidentification System. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):138–146, 2005.

[124] W. Zhu, K. Boriboonsomsin, and M. Barth. Microscopic Traffic Flow Quality of Service from the Drivers' Point of View. In *IEEE, Intelligent Transportation Systems Conference*, pages 47–52, Seattle, USA, 2007.

[125] Y. Xia. Simulationsbasierte automatisierte Identifikation kritischer Fahrsituationen für kooperative und hochautomatisierte Fahrzeuge. Master's thesis,

TU Darmstadt - Institute for Mechatronic Systems in Mechanical Engineering, Darmstadt, Germany, 2017.

[126] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[127] J. Ko, R. Guensler, and M. Hunter. Variability in Traffic Flow Quality Experienced by Drivers - Evidence from Instrumented Vehicles. *Transportation Research Record Journal of the Transportation Research Board*, 1988(1):1–9, 2006.

[128] M. Kuhn and K. Johnson. *Applied Predictive Modeling*. Springer Science + Business Media, New York, USA, 1st edition, 2013.

[129] C. Radhakrishna Rao, H. Toutenburg, Shalabh, and C. Heumann. *Linear Models and Generalizations - Least Squares and Alternatives*. Springer Series in Statistics. Springer-Verlag, Berlin Heidelberg, Germany, 3rd edition, 2008.

[130] P. Junitz, J. Schneider, and H. Winner. Metrik zur Bewertung der Kritikalität von Verkehrssituationen und -szenarien. In *11. Workshop Fahrerassistenzsysteme und automatisiertes Fahren, FAS 2017, Uni-DAS e.V.*, pages 149–160, Walting, Germany, 2017.

[131] J. Hillenbrand, K. Kroschel, and V. Schmid. Situation Assessment Algorithm for a Collision Prevention Assistant. In *IEEE Proceedings Intelligent Vehicles Symposium*, pages 459–465, Las Vegas, USA, 2005.

[132] R. Karlsson, J. Jansson, and F. Gustafsson. Model-based Statistical Tracking and Decision Making for Collision Avoidance Application. In *Proceeding of the 2004 American Control Conference*, pages 3435–3440, Boston, USA, 2004.

[133] D. Frerichs. *Managing Quality in a Virtual World - Application of Quality Assurance for Virtual Vehicle Development* . Presented at IPG - Apply and Innovate, Karlsruhe, Germany, 2018.

[134] N. Weber, D. Frerichs, and U. Eberle. Preprint: A simulation-based, statistical approach for the derivation of concrete scenarios for the release of highly automated driving functions. In *AmE - Automotive meets Electronics, GMM-Fachbericht, VDE*, pages 1–7, Dortmund, Germany, 2020.

[135] S. Hallerbach and U. Eberle. *Simulation-Based Testing of Cooperative and Automated Vehicles*. Presented at the Ko-HAF final event, Dudenhofen, Germany, 2018.

[136] B. Siciliano and O. Khatib, editors. *Springer Handbook of Robotics*. Springer-Verlag, Berlin Heidelberg, Germany, 1st edition, 2008.

[137] Baidu Artificial Intelligence. Apollo Simulation. Accessed: 12.01.2020. `http:`

//apollo.auto/platform/simulation.html.

[138] AVL List GmbH. AVL Autonomous Driving and ADAS. Accessed: 12.01.2020. https://www.avl.com/adas.

[139] CARLA - Open-source simulator for autonomous driving research. Accessed: 12.01.2020. http://carla.org/.

[140] IPG Automotive GmbH. Automatisiertes und autonomes Fahren: Innovationen für die Mobilität der Zukunft. Accessed: 12.01.2020. https://ipg-automotive.com/de/anwendungsgebiete/fahrerassistenz-automatisiertes-fahren/automatisiertes-autonomes-fahren/.

[141] Nvidia Cooperation. Nvidia Drive Constellation. Accessed: 12.01.2020. https://www.nvidia.com/de-de/self-driving-cars/drive-constellation/.

[142] Die PTV Planung Transport Verkehr AG. PTV Vissim. Accessed: 12.01.2020. https://www.ptvgroup.com/en/solutions/products/ptv-vissim/.

[143] TASS International. Prescan - Simulation of ADAS and active safety. Accessed: 12.01.2020. https://tass.plm.automation.siemens.com/prescan.

[144] TESIS. Virtual Test Driving. Accessed: 12.01.2020. https://www.tesis.de/en/?r=1.

[145] Unity Technologies. Automotive, Transportation & Manufacturing. Accessed: 12.01.2020. https://unity.com/solutions/automotive-transportation-manufacturing.

[146] VIRES Simulationstechnologie GmbH. VTD - Virtual Test Drive. Accessed: 12.01.2020. https://vires.com/vtd-vires-virtual-test-drive/.

[147] U. Eberle. *Automatisiertes Fahren - Ein alter Traum oder die Zukunft der Mobilität*. Presented at the European School of Business and School of Theology, Reutlingen, Germany, 2018.

[148] D. Augustin, M. Hofmann, and U. Konigorski. Motion Pattern Recognition for Maneuver Detection and Trajectory Prediction on Highways. In *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pages 1–8, Madrid, Spain, 2018.

[149] J. Schucker and U. Konigorski. Nonlinear Vehicle Trajectory Guidance for Automated Driving on Highways. *IFAC - PapersOnLine - 12th IFAC Symposium*

on Robot Control, 51(22):262–267, 2018.

[150] S. Berger. *Development of Automated Driving Functions*. Presented at the Ko-HAF final event, Dudenhofen, Germany, 2018.

[151] S. Shalev-Shwartz, S. Shammah, and A. Shashua. Mobileye: On a Formal Model of Safe and Scalable Self-driving Cars. 2017. Accessed: 12.01.2020. https://arxiv.org/pdf/1708.06374.pdf.

[152] D. Nister, H-L. Lee, J. Ng, and Y. Wang. Nvidia: The Safety Force Field. 2019. Accessed: 12.01.2020. https://www.nvidia.com/content/dam/en-zz/Solutions/self-driving-cars/safety-force-field/the-safety-force-field.pdf.

Publications, Patents, Invited Presentations, and Supervised Theses

Publications

S. Hallerbach, U. Eberle, and F. Köster. Absicherungs- und Bewertungsmethoden für kooperative und hochautomatisierte Fahrzeuge. In *AAET - Automatisiertes und vernetztes Fahren*, pages 368-384, Brunswick, Germany, 2017.

S. Hallerbach, U. Eberle, and F. Köster. The challenges of releasing cooperative and highly automated vehicles - A look beyond functional requirements. In *AME - Automotive meets Electronics, GMM-Fachbericht, VDE*, pages 102-106, Dortmund, Germany, 2017.

D. Augustin and S. Hallerbach. Interaction-aware Motion Prediction for Highly-Automated Driving Functions on Highways. In *AME - Automotive meets Electronics, GMM-Fachbericht, VDE*, pages 127-132, Dortmund, Germany, 2017.

S. Hallerbach , Y. Xia, U. Eberle, and F. Köster. Simulation-Based Identification of Critical Scenarios for Cooperative and Automated Vehicles. *WCX - World Congress Experience*, SAE Technical Paper 2018-01-1066, Detroit, USA, 2018.

S. Hallerbach, Y. Xia, U. Eberle, and F. Köster. Simulation-Based Identification of Critical Scenarios for Cooperative and Automated Vehicles. *SAE International Journal of Connected and Automated Vehicles*, 1(2): 93-106, 2018.

Patent Applications

S. Hallerbach, D. Augustin, and U. Eberle. Patentanmeldung: Verfahren zum Trainieren selbstlernender Algorithmen für ein automatisiert fahrbares Fahrzeug. 2017.

S. Hallerbach, S. Berger, and U. Eberle. Patentanmeldung: Verfahren zur verkehrssimulationsbasierten Erprobung und Validierung von Komponenten und Fahrzeugen. 2018.

S. Hallerbach, J. Kammerer, and U. Eberle. Patentanmeldung: Methode zum Testen und Absichern von KI-basierten HAF-Systemen. 2018.

U. Eberle and S. Hallerbach. Patentanmeldung: Nutzung von Verkehrssimulation zur Durchführung und Steuerung von Entwicklung und Test automatisierter Fahrzeuge mittels eines angekoppelten Mixed-Reality-Ansatzes. 2019.

U. Eberle and S. Hallerbach. Patentanmeldung: Verfahren zum Trainieren wenigstens eines Algorithmus für ein Steuergerät eines Kraftfahrzeugs, Computerprogrammprodukt, Kraftfahrzeug sowie System. 2019.

U. Eberle and S. Hallerbach. Patentanmeldung: Analyse und Validierung eines neuronalen Netzes für ein Fahrzeug. 2019.

U. Eberle and S. Hallerbach. Patentanmeldung: Analyse eines neuronalen Netzes für ein Fahrzeug. 2019.

U. Eberle and S. Hallerbach. Patentanmeldung: Automatisierte Erkennung eines anormalen Verhaltens eines Verkehrsteilnehmers. 2019.

Invited Presentations

S. Hallerbach, U. Eberle and F. Köster. A Simulation-Based approach for testing and validation of cooperative and automated vehicles. International Symposium - ADFD - Automated Driving, Future Mobility and Digitalization. Hanover, Germany, 2017.

S. Hallerbach, Y. Xia, U. Eberle, and F. Köster. Simulation-Based Identification of Critical Scenarios for Cooperative and Automated Vehicles. DLR-Symposium "Testen - Automatisiertes und vernetztes Fahren." Brunswick, Germany, 2018.

Supervised Theses

L. Zaruba. Development and evaluation of a simulation environment for cooperative and highly automated vehicles. Master's thesis, HAN University of Applied Sciences - Faculty of Engineering - Institute of Automotive, Nijmegen, Netherlands, 2017.

Y. Xia. Simulationsbasierte automatisierte Identifikation kritischer Fahrsituationen für kooperative und hochautomatisierte Fahrzeuge. Master's thesis, TU Darmstadt - Institute for Mechatronic Systems in Mechanical Engineering, Darmstadt, Germany, 2017.

K. Hansel. Kritikalitätsanalyse für kooperative und hochautomatisierte Fahrzeuge auf der Autobahn. Bachelor's thesis, RheinMain University of Applied Sciences, Rüsselsheim, Germany, 2017.

D. Navarro Garcia. Erzeugung simulations-basierter Daten für automatisierte Fahrzeuge. Bachelor's thesis, Frankfurt University of Applied Sciences, Frankfurt, Germany, 2018.

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Die Dissertation hat weder in ihrer Gesamtheit noch in Teilen einer anderen wissenschaftlichen Hochschule zur Begutachtung in einem Promotionsverfahren vorgelegen.

Hiermit erkläre ich, dass mir die Leitlinien guter wissenschaftlicher Praxis der Carl von Ossietzky Universität Oldenburg bekannt sind und von mir befolgt wurden.

Hiermit erkläre ich, dass im Zusammenhang mit dem Promotionsvorhaben keine kommerziellen Vermittlungs- oder Beratungsdienste (Promotionsberatung) in Anspruch genommen worden sind.

Oldenburg, 20.01.2020

Sven Hallerbach