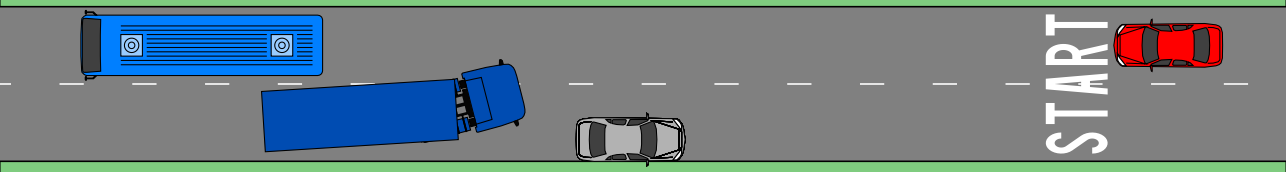
Safety assessment of automated vehicles using real-world driving scenarios

Erwin de Gelder



Safety assessment of automated vehicles using real-world driving scenarios

Safety assessment of automated vehicles using real-world driving scenarios

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen
Chair of the Board for Doctorates
to be defended publicly on
Friday 9 September 2022 at 12:30 o'clock

by

Erwin DE GELDER

Master of Science in Systems & Control,
Delft University of Technology, Delft, the Netherlands,
born in Zwijndrecht, the Netherlands.

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus	Chairperson
Prof. dr. ir. B. De Schutter	Delft University of Technology, promotor
Dr. J.-P. Paardekooper	TNO Integrated Vehicle Safety and Radboud University, copromotor
Independent members:	
Prof. dr. D.M. Gravila	Delft University of Technology
Prof. dr. ir. J.W.C. van Lint	Delft University of Technology
Prof. dr. H. Nijmeijer	Eindhoven University of Technology
Prof. dr. L. del Re	Johannes Kepler University Linz
Prof. dr. G.P. van Wee	Delft University of Technology, reserve member
Other members:	
Dr. O. Op den Camp	TNO Integrated Vehicle Safety

Printed by: Ridderprint, www.ridderprint.nl

ISBN 978-94-6384-362-1

An electronic version of this dissertation is available at
http://repository.tudelft.nl.

I have no special talents.
I am only passionately curious.

Albert Einstein

Contents

Summary

Automated Vehicles (AVs) have a great potential to change transport fundamentally by making it safer, by reducing travel time, and by increasing mobility and accessibility for all. The level of automation of these vehicles determines the extent to which the driver's task is accomplished by the AV. With the increasing number of AVs entering the market, the level of automation of these vehicles is increasing. The increasing level of automation will cause a paradigm shift: traditionally, human drivers are responsible for the behavior of the vehicle, even if the vehicle is momentarily controlled by an Automated Driving System (ADS), but with increasing levels of automation, the human driver will no longer be solely responsible. So, the accountability and liability shift from the driver to the vehicle manufacturer, the operator of the vehicle (fleet), and/or the (vehicle) authorities. Due to this paradigm shift, for higher levels of automation, it can no longer be assumed that the human driver intervenes whenever the ADS does not respond appropriately. To guarantee that these ADSs respond appropriately in nearly all situations, new methods for assessing ADSs are required.

Scenario-based assessment is an approach for assessing AVs that is broadly supported by the automotive field. With a scenario-based assessment, the AV under test is subjected to many different test scenarios. These test scenarios resemble situations that may be encountered in real-world traffic, to see whether the AV responds appropriately to these scenarios. One of the main challenges with scenario-based assessment of an AV with a high level of automation is to come up with a set of test scenarios that provides enough confidence that the AV responds appropriately in nearly all situations. One popular approach is to use real-world data that contain scenarios from real-world traffic as a source to automatically generate test scenarios. This dissertation describes new methods for improving this data-driven scenario-based assessment of AVs.

The first contribution of this dissertation is a comprehensive and operable definition of the term scenario in the context of scenario-based assessment of AVs. We define a scenario as a quantitative description of the relevant characteristics and activities and/or goals of the ego vehicle(s), the static environment, the dynamic environment, and all events that are relevant to the ego vehicle(s) within the time interval between the first and the last relevant event. A scenario category is defined as the qualitative counterpart of a scenario and can be regarded as an abstraction of a scenario. To enable a computer to store, communicate, interact with, and interpret scenarios, an Object-Oriented Framework (OOF) is proposed in which scenarios, scenario categories, and their building blocks are defined as classes of objects having attributes, methods, and relationships. The advantage of the OOF is that it promotes clarity, modularity, and reusability of the objects that constitute a scenario.

The second contribution is a novel metric for quantifying the degree of completeness of the collected data that are used for the data-driven scenario-based assessment of AVs. The data are used to estimate unknown probability density functions (pdfs) of the important parameters that are used to describe scenarios. The proposed completeness metric is based on the expected approximation error, which is the discrepancy between the real pdf and the estimated pdf: a lower approximation error indicates a higher degree of completeness.

The third contribution is a novel method for capturing scenarios of a specific scenario category from a data set. For example, the provided method can capture all cut-in scenarios from a data set. One of the benefits of the method is that characteristics of a scenario that are shared among different scenario categories need to be identified only once. As a result, the provided method is easily applied to a wide range of scenario categories, such that a wide variety of scenarios can be obtained from the data.

The fourth contribution is the proposal of two complementary methods for generating test scenarios for AVs. The first method automatically determines the parameters that best describe the scenarios of a specific scenario category. The underlying, unknown pdf of the parameters is estimated and scenarios are generated by sampling parameter values from the estimated pdf. The second method enables the conditional sampling of parameter values, which can be used to, e.g., generate scenarios with predefined starting conditions. The benefits of the presented methods are that the generated scenarios are representative of real-world scenarios, they cover the actual variety found in real-world traffic, and they extend the variety found in the collected data. To measure the extent to which the generated scenarios indeed represent real-world scenarios while covering the actual variety found in real-world traffic, the novel Scenario Representativeness metric is proposed.

The fifth contribution is the proposal of two novel methods for quantifying the risk of an AV. Both methods calculate the risk by combining the outcome of virtual simulations of scenarios generated using the aforementioned methods and the estimated likelihood of these scenarios. The first method quantifies the risk prospectively, i.e., before the actual deployment of the AV on public roads. The quantified risk supports the risk assessment activities of ISO 26262 and ISO 21448, the leading standards in automotive safety. These standards decompose the risk into three aspects: exposure, severity, and controllability. Whereas safety experts' opinions are traditionally used to provide qualitative, subjective ratings for each of these three aspects, our proposed method computes these aspects in a data-driven, quantitative manner. The second method is the novel data-driven Probabilistic RISk Measure derivAtion (PRISMA) method, which is used to derive Surrogate Safety Measures (SSMs) that estimate the probability of a specific event (e.g., a crash) in real time. As opposed to existing SSMs, which are only applicable in specific types of scenarios, the PRISMA method can be used to derive multiple SSMs for different types of scenarios.

The work presented in this dissertation thus makes a substantial contribution to the full integration of a scenario-based assessment for the type approval of AVs. This, in turn, brings us closer to the large-scale deployment of AVs on public roads.

Samenvatting

Geautomatiseerde voertuigen (Automated Vehicle, AV) hebben een groot potentieel om het vervoer fundamenteel te veranderen door het veiliger te maken, de reistijd te verkorten en de mobiliteit en toegankelijkheid voor iedereen te vergroten. De mate van automatisering van deze voertuigen bepaalt in hoeverre de bestuurder de taak van het AV uitvoert. Met het toenemende aantal AV's dat op de markt komt, neemt het niveau van automatisering van deze voertuigen toe. De toenemende mate van automatisering zal een paradigmaverschuiving veroorzaken: van oudsher zijn menselijke bestuurders verantwoordelijk voor het gedrag van het voertuig, zelfs als het voertuig tijdelijk wordt bestuurd door een geautomatiseerd rijsysteem (Automated Driving System, ADS), maar met toenemende mate van automatisering zal de menselijke bestuurder niet langer verantwoordelijk zijn. De verantwoordelijkheid en aansprakelijkheid verschuiven dus van de bestuurder naar de voertuigfabrikant of de exploitant van het voertuig of de voertuigvloot. Vanwege deze paradigmaverschuiving kan voor hogere automatiseringsniveaus niet langer worden aangenomen dat de menselijke bestuurder ingrijpt wanneer het ADS niet adequaat reageert. Om te garanderen dat een ADS in bijna alle situaties adequaat reageert, zijn nieuwe methoden nodig om deze systemen te beoordelen.

De beoordeling van AV's op basis van scenario's is een door de automobielindustrie breed gedragen aanpak voor het beoordelen van AV's. Bij een op scenario's gebaseerde beoordeling wordt de te testen AV onderworpen aan veel verschillende testscenario's. Deze testscenario's lijken op situaties die zich in het echte verkeer kunnen voordoen, om te zien of het AV adequaat op deze scenario's reageert. Een van de grootste uitdagingen bij deze op scenario's gebaseerde beoordeling van AV's met een hoge mate van automatisering is het bedenken van een set testscenario's die voldoende zekerheid biedt dat het AV in bijna alle situaties adequaat reageert. Een populaire benadering is het gebruik van echte data die scenario's van het echte verkeer bevatten als bron om automatisch testscenario's te genereren. Dit proefschrift beschrijft nieuwe methoden voor het verbeteren van de datagedreven, scenariogebaseerde beoordeling van AV's.

De eerste bijdrage van dit proefschrift is een uitgebreide en bruikbare definitie van het begrip scenario in de context van scenariogebaseerde beoordeling van AV's. We definiëren een scenario als een kwantitatieve beschrijving van de relevante eigenschappen en activiteiten en/of doelen van het (de) ego-voertuig(en), de statische omgeving, de dynamische omgeving en alle gebeurtenissen die relevant zijn voor het (de) ego-voertuig(en) binnen het tijdsinterval van de eerste en de laatste relevante gebeurtenis. Een scenariocategorie is gedefinieerd als de kwalitatieve tegenhanger van een scenario en kan worden beschouwd als een abstractie van een scenario. Om een computer in staat te stellen scenario's op te slaan, te communiceren en te interacteren met scenario's is een object georiën-

teerd raamwerk (Object-Oriented Framework, OOF) voorgesteld waarin scenario's, scenariocategorieën en hun bouwstenen gedefinieerd zijn als klassen van objecten met attributen, methoden en relaties. Het voordeel van het OOF is dat het duidelijkheid, modulariteit en herbruikbaarheid van de objecten die een scenario vormen bevordert.

De tweede bijdrage is een nieuwe maat voor het kwantificeren van de mate van volledigheid van de data die gebruikt worden voor de datagedreven, scenariogebaseerde beoordeling van AV's. De data worden gebruikt voor het schatten van onbekende kansdichtheidsfuncties (probability density function, pdf) van de belangrijke parameters die worden gebruikt om scenario's te beschrijven. De voorgestelde volledigheidsmaat is gebaseerd op de verwachte schattingsfout, welke het verschil is tussen de echte pdf en de geschatte pdf: een lagere schattingsfout duidt op een hogere mate van volledigheid.

De derde bijdrage is een nieuwe methode om scenario's van een specifieke scenariocategorie uit een dataset te halen. De voorgestelde methode kan bijvoorbeeld invoegscenario's uit een dataset halen. Een van de voordelen van de methode is dat kenmerken van een scenario die worden gedeeld met andere scenariocategorieën slechts eenmaal hoeven te worden geïdentificeerd. Hierdoor is de voorgestelde methode eenvoudig toepasbaar op een breed scala aan scenariocategorieën zodat een grote variëteit aan scenario's uit de data kan worden verkregen.

De vierde bijdrage is het voorstel van twee complementaire methoden voor het genereren van testscenario's voor AV's. De eerste methode bepaalt automatisch de parameters die de scenario's van een specifieke scenariocategorie het beste beschrijven. De werkelijke onderliggende, onbekende pdf van de parameters wordt geschat en scenario's worden gegenereerd door parameterwaarden te bemonsteren uit de geschatte pdf. De tweede methode maakt de voorwaardelijke bemonstering van de parameters mogelijk. Dit kan bijvoorbeeld worden gebruikt om scenario's met vooraf gedefinieerde startvoorwaarden te genereren. De voordelen van de gepresenteerde methoden zijn dat de gegenereerde scenario's representatief zijn voor echte scenario's, dat ze de werkelijke variatie in het echte verkeer dekken en dat ze de variatie van de verzamelende data uitbreiden. Om te meten in hoeverre de gegenereerde scenario's inderdaad echte scenario's vertegenwoordigen en tegelijkertijd de werkelijke variatie in het echte verkeer dekken, wordt de nieuwe scenario representativiteit (Scenario Representativeness, SR) maat voorgesteld.

De vijfde bijdrage is het voorstel van twee nieuwe methoden voor het kwantificeren van het risico van een AV. Beide methoden berekenen het risico door de uitkomst van virtuele simulaties van scenario's die zijn gegenereerd met de bovengenoemde methoden te combineren met de geschatte waarschijnlijkheid van deze scenario's. De eerste methode kwantificeert het risico prospectief, dus vóór de daadwerkelijke inzet van het AV op de openbare weg. Het gekwantificeerde risico ondersteunt de risicobeoordelingsactiviteiten van ISO 26262 en ISO 21448, de toonaangevende normen op het gebied van autoveiligheid. Deze normen splitsen het risico op in drie aspecten: blootstelling, ernst en beheersbaarheid. Waar traditioneel de meningen van veiligheidsexperts worden gebruikt voor kwalitatieve, subjectieve beoordelingen voor elk van deze drie aspecten, berekent onze voor-

gestelde methode deze aspecten op een datagedreven, kwantitatieve manier. De tweede methode is een nieuwe manier voor datagedreven probabilistische afleiding van risicostatistieken (Probabilistic RISk Metric derivAtion, PRISMA), die wordt gebruikt voor het afleiden van surrogate veiligheidsstatistieken (Surrogate Safety Metric, SSM) die in realtime de waarschijnlijkheid van een specifieke gebeurtenis (bijvoorbeeld een crash) schatten. In tegenstelling tot bestaande SSM's, welke alleen toepasbaar zijn in specifieke soorten scenario's, kan de PRISMA-methode worden gebruikt voor het afleiden van meerdere SSM's voor verschillende soorten scenario's.

Het werk gepresenteerd in dit proefschrift levert dus een substantiële bijdrage aan de volledige integratie van een scenariogebaseerde beoordeling voor de typegoedkeuring van AV's.

Acronyms

ACC	Adaptive Cruise Control
ADS	Automated Driving System
AMISE	Asymptotic Mean Integrated Squared Error
ASIL	Automotive Safety Integrity Level
ASV	approaching slower vehicle
AV	Automated Vehicle
CC	Cruise Control
CPI	Crash Potential Index
DRAC	Deceleration Rate to Avoid Collision
EVT	Extreme Value Theory
FCW	Forward Collision Warning
FMEA	Failure Mode and Effect Analysis
GAN	Generative Adversarial Network
HARA	Hazard Analysis and Risk Assessment
IDM+	Intelligent Driver Model plus
KDE	Kernel Density Estimation
LVD	leading vehicle decelerating
MADR	Maximum Available Deceleration Rate
MAIS	Maximum Abbreviated Injury Scale
MISE	Mean Integrated Squared Error
MTTC	Modified TTC
NGSIM	Next Generation SIMulation
NLP	Natural Language Processing
NW	Nadaraya-Watson

ODD Operational Design Domain

OOF Object-Oriented Framework

PCA Principal Component Analysis

PET Post-Encroachment Time

PICUD Potential Index for Collision with Urgent Deceleration

pdf probability density function

PRISMA Probabilistic RISk Measure derivAtion

PSD Proportion of Stopping Distance

QM Quality Management

SOTIF Safety Of The Intended Functionality

SR Scenario Representativeness

SSM Surrogate Safety Measure

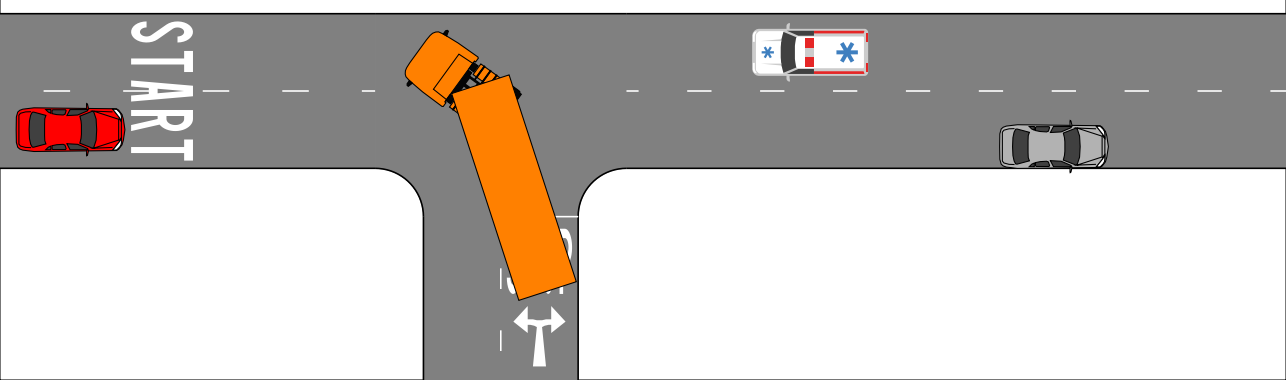
STPA Systems-Theoretic Processes Analysis

SVD Singular Value Decomposition

THW Time Headway

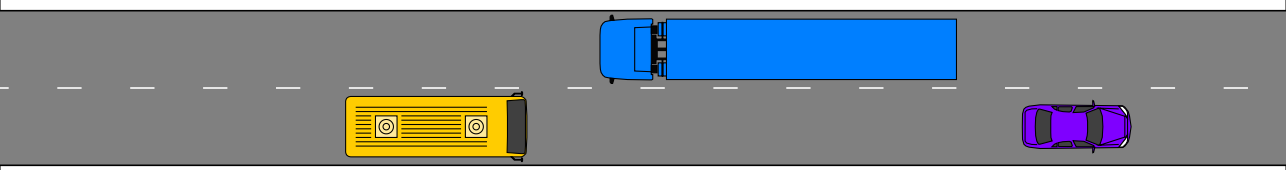
TIT Time Integrated TTC

TTC Time to Collision


START

1

Introduction



1

The aim of the study described in this dissertation is to develop methods for the safety assessment of Automated Vehicles (AVs) using real-world driving scenarios. This introductory chapter starts with explaining the need for scenario-based testing to address the challenge of the assessment of vehicles with a high automation level. Next, it explains why real-world data should be used as a basis for the scenarios and it describes the overall approach of a scenario-based assessment. Then the introduction outlines the research goals that are addressed in this work. This chapter concludes with explaining the structure and organization of the remaining chapters.

1.1. Automated vehicles

Essentially, vehicles are machines that transport people and goods from A to B. Traditionally, a human is in charge of controlling road vehicles by means of steering, accelerating, and braking. In recent years, more and more vehicles have been entering the market that are equipped with multiple systems that aid the human in the driving task or even to take over the driving task for a part of a trip. These systems have a great potential to change transport fundamentally by making it safer, by reducing travel time, energy consumption, and pollution, and by increasing mobility, accessibility, and availability for all [227].

Examples of systems that aid the human in the driving task range from systems that only act in an emergency, systems that warn a driver for a potential collision, and comfort systems. For example, an autonomous emergency braking system [252] typically uses a camera, a radar, or a combination of these sensors to detect an upcoming collision and activates the brakes upon imminent collision to avoid or mitigate the consequences in case a collision is unavoidable. Warning systems include forward collision warning [175], blind-spot detection [186], and lane departure warning [152]. Well-known examples of comfort systems are Adaptive Cruise Control (ACC) [191], which keeps the vehicle at a safe distance behind a preceding vehicle while not exceeding a set speed, and lane keeping assist system [193], which steers the vehicle in order to avoid potential lane departures.

To classify the level of automation of an Automated Driving System (ADS), the SAE J3016 standard [243] defines six levels of automation, ranging from no automation to full automation. Figure 1.1 shows a summary of these levels. The six levels are defined as follows:

- With level 0 (no driving automation), the ADS may issue warnings and may temporarily intervene, but the driver is in charge of the sustained motion control.
- A level 1 system (driver assistance) takes over part of the motion control, e.g., the longitudinal motion control using an ACC.
- With a level 2 system (partial driving automation), the ADS takes over the motion control. It is important to note that with a level 2 system, just like a level 1 system, the human driver is still in charge of supervising the system and responsible for intervening immediately if the system fails to respond

Level	Name	Motion control	Fallback	ODD
0	No driving automation	Driver	Driver	Not applicable
1	Driver assistance	Driver and system	Driver	Limited
2	Partial driving automation	System	Driver	Limited
3	Conditional driving automation	System	Fallback-ready user	Limited
4	High driving automation	System	System	Limited
5	Full driving automation	System	System	Unlimited

Figure 1.1: Summary of levels of driving automation according to SAE J3016 [243]. Note that ODD refers to operating conditions under which a given Automated Driving System (ADS) is designed to function. Adapted from [243].

properly. So, the driver is the fallback in case the system fails to respond properly.

- Level 3 (conditional driving automation) differs from level 2 in that the system is able to cope with situations that require an immediate response. Still, the driver must be prepared to intervene within some limited time in case the system asks the driver to do so.
- A level 4 system (high driving automation) is responsible for both the motion control and the fallback. Therefore, no driver attention is required. The driving automation, however, is still only available for a particular Operational Design Domain (ODD), where the ODD refers to the operating conditions under which the ADS is designed to function. For example, the ADS is only available in some designated area during daytime.
- Level 5 (full driving automation) is not bound to a specific ODD and, thus, offers a self-driving functionality in all operating conditions.

For a vehicle with no driving automation system, the driver is accountable in case

of any damage caused by the vehicle. An AV refers to a vehicle equipped with an ADS of level 1 or higher. Since the driver is still responsible for intervening if a level 1 or level 2 ADS fails, the driver is also accountable for AVs with driver assistance (level 1) or partial driving automation (level 2). From level 3 and onward, the driver cannot fully control the vehicle in each single situation and is neither required nor expected to do so. Hence, the accountability shifts away from the driver [189]. As a result, according to a German ethics committee for automated and connected driving [189], the companies who built the vehicle or who are operating its relevant systems are liable for damage caused by activated ADSs of level 3 and higher. In addition, the authority that approves the AVs for usage on the public roads has the responsibility to thoroughly assess the AVs to verify whether the assessed AVs can be deployed safely on the public roads.

The development of AVs with conditional, high, or full driving automation requires extensive testing. For lower levels of automation (level 0, 1, and 2), it is assumed that the driver responds appropriately to situations that the AV cannot handle. For ADSs of level 3, 4, and 5, the driver, if any, may not be fully responsible in case of an accident while the ADS is activated. Therefore, it must be assured that such ADSs can cope with all kinds of scenarios, even with scenarios that are rare, without relying on a successful intervention of a human driver. It is even expected that ADSs of level 3 and above must be at least 5 times safer than a human driver in order to be publicly accepted [188]. To prove that ADSs of level 3 and above are safe enough, the more traditional methods [144, 164], used for evaluation of driver assistance systems, are no longer sufficient [165, 167, 291]. This is mainly because the traditional methods assume that a human can take over control in case the ADS fails to respond appropriately and because the traditional methods assume that it is possible to predict nearly all (potentially dangerous) situations during the design process [168]. Also, safety validation through test drives with prototypes is infeasible as this requires billions of kilometers of driving [155]. Therefore, as a widely used alternative in the automotive field, a scenario-based assessment approach is adopted in this PhD dissertation.

1.2. Scenario-based assessment

With a scenario-based assessment of an AV, the AV is tested in a large number of individual, distinct traffic scenarios. In this context, informally, a scenario represents a description of the AV and its surroundings over a limited period of time (a more rigorous definition follows in Chapter 2). One challenge with scenario-based assessment is to come up with the scenarios that are used during the assessment. In this thesis, as will be motivated next, we focus on a data-driven approach for generating the scenarios for the assessment. In a data-driven approach, scenarios are collected from real-world driving data and these collected scenarios are used to generate test scenarios.

Instead of adopting a data-driven approach for the generation of test scenarios, alternative approaches are adopted as well. One such an alternative approach is a knowledge-driven approach. In this approach, typically, expert knowledge is structured using one [23, 185] or multiple [51] ontologies. An ontology is a

way to formalize properties of and relations between different concepts. The developed ontologies are automatically converted into a set of test scenarios. Another method is to adapt the test scenarios based on the ADS-under-test, such that the scenarios are challenging for or even show failures of the system-under-test [58, 149, 169, 205, 272]. A third alternative for scenario generation is functionality-based testing where test scenarios are based on a specific functionality of the ADS [137, 138]. Although each of these three alternatives has its own advantages and drawbacks, which are not further discussed here, they all share one drawback: no quantitative evaluation can be made of the performance of the ADS-under-test once deployed on the road because it is unknown how realistic and likely the generated test scenarios are. With a data-driven approach, the likelihood of the scenarios can be estimated from the data. Therefore, a data-driven approach is a valuable, complementary, and necessary extension to the aforementioned approaches for generating test scenarios.

The objective of the data-driven scenario-based assessment is to quantify the risk of an AV once the AV would be deployed on the public roads. Figure 1.2 presents a schematic overview of scenario-based assessment using real-world data that is followed in the study described in this dissertation. The process consists of 9 steps:

1. Data are acquired, e.g., using a vehicle equipped with sensors [221].
2. Activities of the road participants are detected, where activities refer to elements of a scenario. An activity can be, e.g., a particular lane change of a vehicle or a braking action of a vehicle.
3. Based on the detected activities in the previous step and other information, e.g., the road layout, scenarios are mined, i.e., identified and extracted, from the data.
4. Parameters are defined for characterizing the mined scenarios. Values of the parameters are computed for each of the mined scenarios.
5. Based on the parameterized scenarios, scenario statistics are obtained. These scenario statistics contain, among others, the likelihood of encountering scenarios with specific parameter values.
6. Test scenarios are generated using the scenario statistics from the previous step.
7. Tests are conducted in which the response of the AV in the test scenarios is measured. Often virtual simulations are used for conducting part of the tests.
8. Aggregating all test results, typically with some selected key performance indicators, leads to the actual evaluation of the AV performance.
9. After approval of an AV, the AV may be deployed on the public road. The scenario-based assessment does not end as the AV is still monitored during its deployment. The so-called in-service monitoring may be required by road

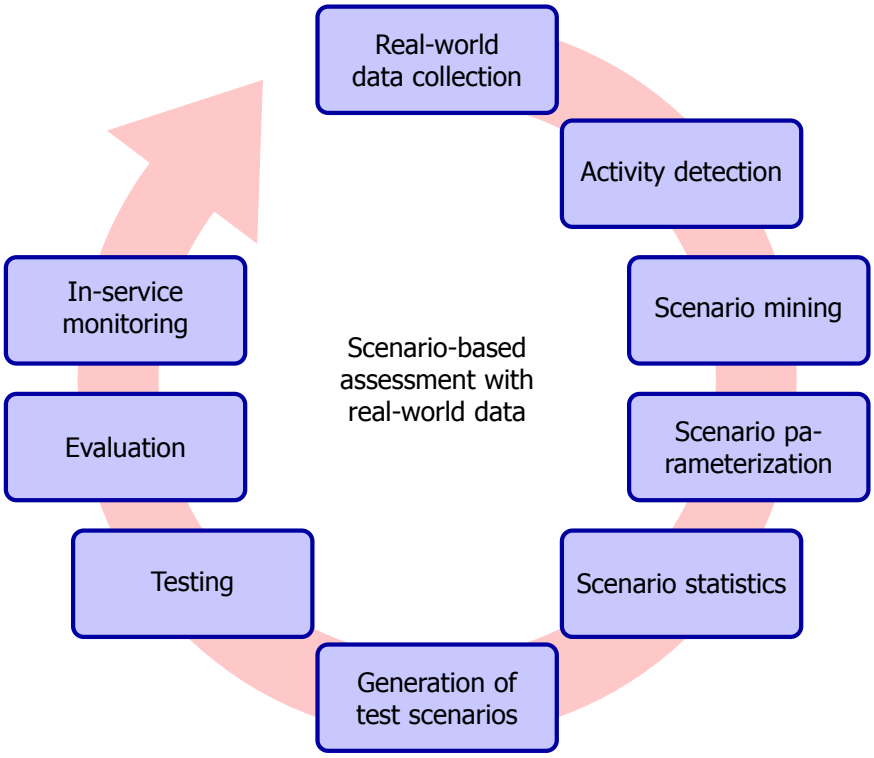


Figure 1.2: Schematic overview of the process of scenario-based assessment using real-world data that is followed in this dissertation.

and/or vehicle authorities in order to assess the safety continuously during the deployment.

The acquired data during in-service monitoring can be used to improve the generation of tests as it is possible that some scenarios have been overlooked during the initial assessment process. Furthermore, the situations on the road will gradually change with changes in traffic, e.g., because of the introduction of new mobility systems. Therefore, new data need to be collected and the process as indicated in Figure 1.2 needs to be repeated with the new data. The acquired data during the in-service monitoring may be used for this purpose. The arrow in Figure 1.2 indicates the continuous repetition of the scenario-based assessment approach for the continuous improvement of the scenario-based assessment, accommodating both tests that have been overlooked and changes to the traffic.

1.3. Research questions

This dissertation intends to contribute to the development of the data-driven, scenario-based assessment as outlined in Figure 1.2. To do so, the thesis endeavors to answer the research questions that are presented in this section.

The objective of scenarios for scenario-based assessment of AVs is to represent the actual phenomena in the real-world traffic. Given the complexity of the reality that is being modeled, it is a challenge to define a structure for capturing these scenarios. Nevertheless, a proper specification of the scenarios is crucial in order to arrive at an unambiguous description of scenarios that is required for providing repeatable and reproducible tests [16]. Furthermore, properly specified scenarios enable us to translate the result of a test into an assessment of the AV performance with regards to a particular ODD [125, 302]. On the one hand, existing definitions of the notion of a scenario are unclear because of ambiguities and the use of other undefined terms. On the other hand, existing file formats for specifying scenarios unambiguously lack the definitions and justifications of each of the terms. The following research question aims to address this:

Research question 1.1. *What is a scenario and how to specify a scenario in the context of the scenario-based assessment of AVs?*

As a source of information for generating test scenarios, real-world data are used. Because data collection is time-consuming and requires high investments and resources, questions like "Do we have enough data?," "How much more information can we gain when obtaining more data?," and "How far are we from obtaining completeness?" are highly relevant. In fact, deducing safety claims based on a scenario-based assessment that uses collected data as a basis for the test scenarios requires knowledge about the degree of completeness of the data used. Therefore, this thesis also aims to answer the following research question:

Research question 1.2. *How to quantify whether we have collected enough field data?*

In the data-driven scenario-based assessment approach, the scenarios that are captured from the real-world data form the basis of the test scenarios. Therefore, different techniques for capturing scenarios from real-world data are proposed in the literature [45, 157, 171, 221, 248, 313]. These techniques all focus on specific types of scenarios, e.g., a cut-in scenario. This research aims to provide a method for extracting the scenarios from the data that can easily be extended to multiple types of scenarios. Hence, this thesis also considers the following research question:

Research question 1.3. *How to extract a specific type of scenario from a given data set with real-world traffic data and how to easily extend this approach to other types of scenarios?*

The observed scenarios that are extracted from the data can directly be used as test scenarios [176]. In this case, however, the total variety of scenarios that is found in real life will not be covered unless unrealistic amounts of data are gathered. As an alternative to directly using the observed scenarios as test scenarios, test scenarios can be generated based on the statistical information that underlies the observed scenarios [94]. In the existing literature, scenario generation methods are typically

- oversimplifying scenarios, e.g., assuming a vehicle's speed is constant [70, 277], and/or
- simplifying the distributions of the scenario parameters, e.g., assuming Gaussian distributions [113] or independent distributions [102].

To address this, this thesis also aims to answer the following research question:

Research question 1.4. *Based on a set of observed scenarios, how to generate test scenarios for the assessment of AVs without oversimplifying the scenarios?*

ISO 26262 [144] and ISO 21448 [143], the leading standards in automotive safety, provide an approach to estimate the risk where risk is defined as the combination of the probability of occurrence of harm and the severity of that harm. The former standard focuses on risks due to potential malfunctioning of components and the latter standard focuses on risks due to possible functional insufficiencies. The main shortcomings of the approach provided in ISO 26262 are that it depends on subjective judgments of safety experts and that only a qualitative risk estimation is performed. ISO 21448 addresses these shortcomings partially by providing statistical methods to guide the safety validation, but no complete method is provided to quantify the risk. To address the lack of a complete method for quantifying the risk of an AV, this thesis also considers the following research question:

Research question 1.5. *How to quantify the risk of an AV in real-world scenarios?*

1.4. Organization of the dissertation

This thesis is presented as a collection of papers, either published, accepted for publication, or under review. Except for this chapter and the last chapter, each chapter is based on a single publication and, therefore, the chapters are standalone and can be read separately. The structure of the thesis is shown in Figure 1.3. Although each chapter can be read independently, the arrows in Figure 1.3 indicate a convenient reading order. The contents of each chapter can be summarized as follows:

Chapter 2: Object-oriented framework for scenarios

This chapter addresses Research question 1.1 by presenting a comprehensive and operable definition of the notion of scenario for the context of scenario-based assessment of AVs while considering existing definitions in the literature. This is achieved by proposing an Object-Oriented Framework (OOF) in which scenarios and their building blocks are defined as classes of objects having attributes, methods, and relationships with other objects. The object-oriented approach promotes clarity, modularity, reusability, and encapsulation of the objects. This chapter also provides definitions and justifications of each of the terms. Furthermore, the framework is used to translate the terms in a coding language that is publicly available.

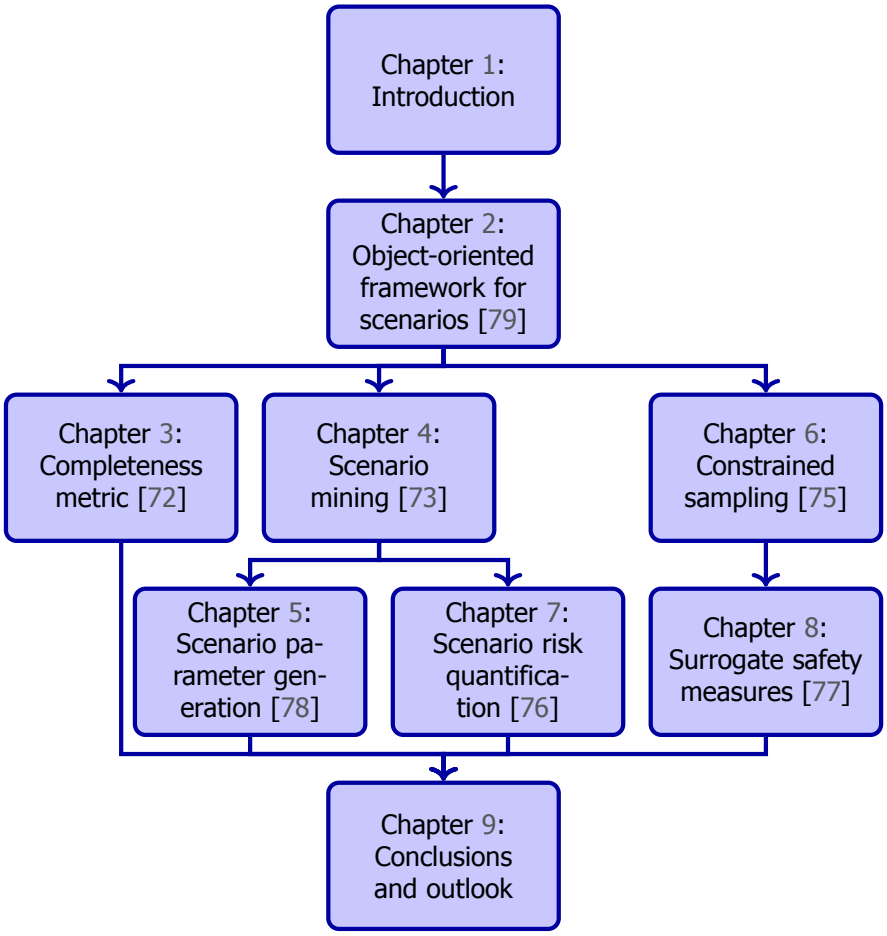


Figure 1.3: Organization of the chapters of this dissertation.

Chapter 3: Quantifying whether we have collected enough field data

To answer Research question 1.2, Chapter 3 proposes a method for quantifying the completeness of the so-called activities in a data set. For every activity, we create a parameterization that encodes the information in the data of this activity. For each type of activity, we estimate a probability density function (pdf) of the associated parameters. Our proposed method approximates the degree of completeness of a data set using the estimated uncertainty of the estimated pdf. The smaller this uncertainty, the higher the degree of completeness and vice versa.

Chapter 4: Real-world scenario mining

Chapter 4 tackles Research question 1.3 by introducing a new method to capture scenarios from real-world data using a two-step approach. The first step consists in automatically labeling the data with tags. These tags typically describe activities,

e.g., a vehicle that changes lane. Second, we mine the scenarios, represented by a combination of tags, based on the labeled tags. One of the benefits of our approach is that the tags can be used to identify characteristics of a scenario that are shared among different type of scenarios. In this way, these characteristics need to be identified only once. Furthermore, the method is not specific for one type of scenario and, therefore, it can be applied to a large variety of scenarios.

Chapter 5: Generation and evaluation of test scenarios

Chapter 5 addresses Research question 1.4. The contribution of this chapter is twofold. First, we propose a method to automatically determine the parameters that describe the scenarios. Because the proposed method determines the parameters automatically, the chosen parameterization relies less on strong assumptions that are typically made when the parameters that characterize the scenarios are manually selected. By estimating the pdf of these parameters, realistic parameters values can be generated. Second, we present the Scenario Representativeness (SR) metric based on the Wasserstein distance, which quantifies to what extent the scenarios with the generated parameter values are representative of real-world scenarios while covering the actual variety found in real-world scenarios. A comparison of our proposed method with methods relying on assumptions of the scenario parameterization and pdf estimation shows that the proposed method outperforms the latter. The presented method is promising because the parameterization and pdf estimation can directly be applied to already available importance sampling strategies for accelerating the evaluation of AVs.

Chapter 6: Constrained sampling to generate scenario parameters

Chapter 6 continues addressing Research question 1.4. One way to generate the required scenario-based test descriptions is to parameterize the scenarios and to draw these parameters from a pdf that is fitted to the data. In some cases, it might be useful to sample the parameters of a scenario such that the samples satisfy a linear equality constraint, e.g., in case we want to generate scenarios in which a vehicle has a predetermined starting speed. In this chapter, we propose a method to sample from a pdf estimated using Kernel Density Estimation (KDE), such that the samples satisfy a linear equality constraint.

Chapter 7: Risk quantification in driving scenarios

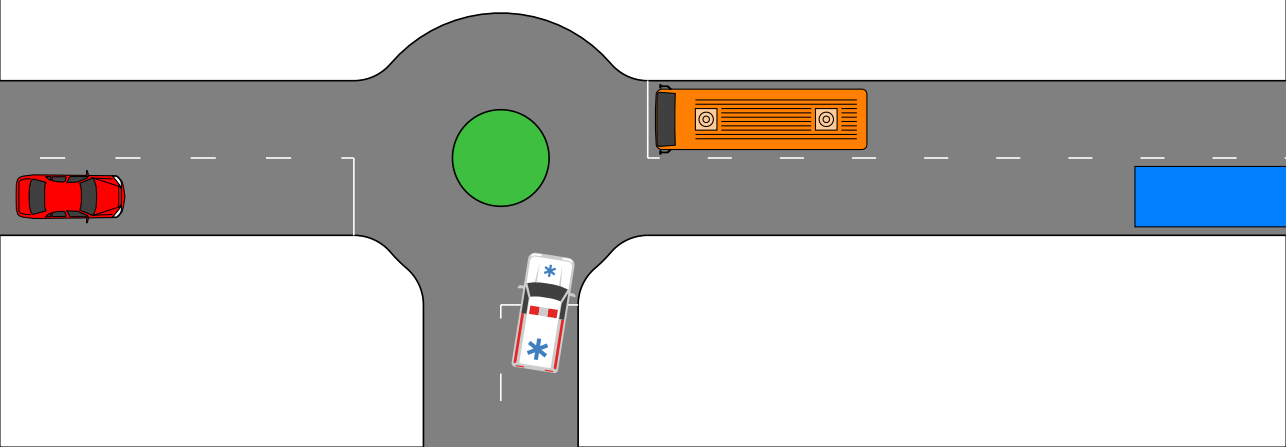
To address Research question 1.5, the first objective of Chapter 7 is to propose a method to estimate the risk of an ADS in a more quantitative and objective manner. A data-driven approach is used to rely less on subjective judgments of safety experts. The output of the method is the expected number of injuries in a potential crash. Thus, the method is quantitative, the result is easily interpretable, and the result can be compared with road crash statistics. The second objective is to provide a method that supports the risk assessment as stipulated by the ISO 26262 and ISO 21448 standards by decomposing the quantified risk into the three aspects of risk considered in these standards: exposure, severity, and controllability.

Chapter 8: Probabilistic risk measure derivation method

Research question 1.5 is also addressed in Chapter 8. Surrogate Safety Measures (SSMs) are used to express road safety in terms of the safety risk in traffic conflicts. Typically, SSMs rely on assumptions regarding the future evolution of traffic participant trajectories to generate a measure of risk. As a result, they are only applicable in scenarios where those assumptions hold. To address this issue, Chapter 8 presents a novel data-driven Probabilistic RISk Measure derivAtion (PRISMA) method. The PRISMA method is used to derive SSMs that can be used to calculate in real time the probability of a specific event (e.g., a crash). Because we adopt a data-driven approach to predict the possible future evolutions of traffic participant trajectories, compared to traditional SSMs, less assumptions on these trajectories are needed. Since the PRISMA method is not bound to specific assumptions, multiple SSMs for different types of scenarios can be derived.

Chapter 9: Conclusions and outlook

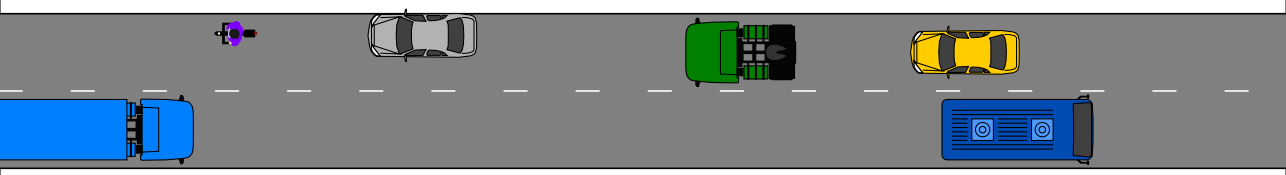
Finally, this chapter concludes this dissertation with conclusions regarding the contributions of this work and an outlook with directions for future work.



2

Object-oriented framework for scenarios

This chapter is based on:
E. de Gelder, J.-P. Paardekooper, A. Khabbaz Saberi, H. Elrofai, O. Op den Camp, S. Kraines, J. Ploeg, and B. De Schutter, *Towards an ontology for scenario definition for the assessment of automated vehicles: An object-oriented framework,* IEEE Transactions on Intelligent Vehicles **7**, 300 (2022).



The development of new assessment methods for the performance of automated vehicles is essential to enable the deployment of automated driving technologies, due to the complex operational domain of automated vehicles. One contributing method is scenario-based assessment in which test cases are derived from real-world road traffic scenarios obtained from driving data. Given the complexity of the reality that is being modeled in these scenarios, it is a challenge to define a structure for capturing these scenarios. An intensional definition that provides a set of characteristics that are deemed to be both necessary and sufficient to qualify as a scenario assures that the scenarios constructed are both complete and intercomparable.

In this chapter, we develop a comprehensive and operable definition of the notion of scenario while considering existing definitions in the literature. This is achieved by proposing an object-oriented framework in which scenarios and their building blocks are defined as classes of objects having attributes, methods, and relationships with other objects. The object-oriented approach promotes clarity, modularity, reusability, and encapsulation of the objects. We provide definitions and justifications of each of the terms. Furthermore, the framework is used to translate the terms in a coding language that is publicly available.

2.1. Introduction

An essential aspect in the development of Automated Vehicles (AVs) is the assessment of quality and performance aspects of the AVs, such as safety, comfort, and efficiency [27, 114, 132, 229, 233, 235, 270, 291]. For legal and public acceptance of AVs, a clear definition of system performance is important, as well as quantitative measures for system quality. According to Wachenfeld and Winner [291], traditional methods for evaluating driver assistance systems, such as [144, 164], cannot sufficiently assess quality and performance aspects of an AV because they would require too many resources. A scenario-based approach could be a viable way to perform the AV assessment [94, 229, 233]. For a scenario-based assessment, proper specification of scenarios is crucial because

- scenarios provide the basis and justification for the tests used for the scenario-based assessment [16, 112, 229, 270, 284, 327],
- it helps to arrive at an unambiguous description of scenarios that is crucial for providing standardized, repeatable, and reproducible tests [16],
- standardized descriptions of scenarios can be more easily compared and classified automatically [74],
- properly specified scenarios are the basis for evaluating the coverage of the assessment [229], and
- properly specified scenarios enable us to translate the result of a test into an assessment of the AV performance with regards to a particular Operational Design Domain (ODD) [125, 302].

Although the notion of scenario is frequently used in the context of automated driving [89, 114, 140, 212, 229, 231, 235, 259, 314, 327], only rarely is an explicit definition actually given. Furthermore, even those definitions are unclear because of ambiguities and the use of other undefined terms. From the implementation perspective, describing scenarios unambiguously becomes more important given the many simulators that are recently being introduced [50, 158, 211, 238, 306]. To this end, there are several file formats and methods for defining scenarios for the assessment of AVs, such as OpenSCENARIO [20] and CommonRoad [12]. Because the focus of these implementations is on scenarios that can be simulated, these implementations describe scenarios at a quantitative level and, consequently, they do not provide concepts for a qualitative description of a scenario. Furthermore, these implementations and other object-oriented approaches used in the field of the assessment of AVs [282, 286, 307, 328] mostly lack the definitions and justifications of each of the terms.

In this work, as a starting point for developing a full ontology of scenarios, we propose a novel Object-Oriented Framework (OOF) that addresses the aforementioned shortcomings. To avoid ambiguities in the definitions, we provide intensional definitions for concepts corresponding to scenarios and all of their essential building blocks (such as activities, actors, and events). These intentional definitions give

the meaning of the concepts by specifying necessary and sufficient conditions for when the concepts should be used. We base the definitions of each of the components on definitions that are commonly used in the field of the safety assessment of AVs [20, 104, 112, 245, 268, 284]. While being broadly consistent with existing definitions [93, 112, 284], this framework aims to be sufficiently explicit to enable the formalization of a scenario description. More specifically, because we give the characteristics of the concepts corresponding to scenarios and specify how those concepts interrelate, we can define the scenario components as objects of classes having attributes, methods, and relationships with objects that are members of other classes. In addition to the definition of a scenario, we introduce the concept of a *scenario category* that is used to qualitatively describe scenarios, i.e., an abstraction of a scenario. Scenario categories enable the categorization of scenarios in terms of the categories of their typical components. The presented OOF provides explicit guidelines for the construction of scenario descriptions that are able to effectively assess the AV performance.

The proposed approach brings several benefits. First, we provide concepts for a qualitative description of a scenario, which is useful because it enables to classify scenarios and to interpret scenarios. Second, the OOF allows for reusing and maintaining (the building blocks of) a scenario as well as performing operations on and interacting with (the building blocks of) a scenario. Third, our framework is supported with the definitions and justifications of each of the concepts. Fourth, the framework enables the translation of the concepts and their relationships into object-oriented code. This, in turn, is used to describe scenarios in a coding language that can be understood by various software agents, such as simulation tools, and that can be ported to already available formats like OpenSCENARIO [20].

To illustrate how to use the presented OOF, we have implemented the framework in a coding language that is publicly available at `https://github.com/ErwindeGelder/ScenarioDomainModel`[1]. This link contains real-life applications of the presented OOF, such as describing scenarios extracted from data [73]. The framework is also used as a schema for a database system for storing scenarios and scenario categories. Such a database can be used to perform scenario-based assessment of AVs[2] [76]. To further illustrate the use of the OOF, this chapter provides an example with a real-world case in which a vehicle approaches a pedestrian crossing. The proposed OOF provides a first step towards an ontology [263] for scenarios for the assessment of AVs. In a subsequent study, the formalized concepts presented in this chapter will be used to design an ontology with logical constraints that enable a computer to perform reasoning on scenarios.

The outline of the chapter is as follows. In Section 2.2, we explain why an OOF is useful and what the context is. We define the notions of *scenario*, *event*, *activity*, and *scenario category* in Section 2.3. The OOF that formalizes these definitions is presented in Section 2.4. In Section 2.5, an application example is provided

[1] As a coding language, Python is used. The code implementation also contains more methods than presented in this chapter.

[2] An illustration of such an assessment is publicly available at:
`https://github.com/ErwindeGelder/ScenarioRiskQuantification`.

to illustrate the use of the framework with a real-world scenario. The chapter is concluded in Section 2.6.

2.2. Background

In Section 2.2.1, we explain why we want to present an OOF for describing scenarios and scenario categories. Section 2.2.2 provides information on the context for which we want to define scenarios.

2.2.1. Why an object-oriented framework?

According to Johnson and Foote [150], an OOF is a "set of classes that embodies an abstract design for solutions to a family of related problems." The object orientation is used for "a representation, modeling, and abstraction formalism" [303], which is why it is considered "not only useful but also fundamental" [303]. In addition, Patridge [223] notes that object-oriented modeling can provide a bridge from traditional entity-relation-based data modeling to data modeling that is fully grounded in a formalized ontology. An OOF offers the following benefits:

- *Clarity:* It provides "a common vocabulary for designers to communicate, document, and explore design alternatives" [109].
- *Modularity:* By decomposing a scenario into components, the complexity of a scenario itself is reduced. Thus, "modularity makes it easier to understand the effect of changes" [150].
- *Reusability:* An OOF promotes reusability [150, 197, 265]. For example, if two classes share certain procedures and/or properties, these procedures and/or properties could be provided by a so-called superclass from which these two classes inherit the procedures and properties, such that these procedures and properties need to be defined only once.
- *Encapsulation:* Encapsulation assures "that compatible changes can be made safely, which facilitates program evolution and maintenance" [265].
- *Possibility to translate to object-oriented programming languages:* As the OOF consists of a set of classes, it can be directly used in an object-oriented coding language. The OOF then specifies the relationships between the different classes and provides information on the properties of a class and the possible values.

2.2.2. Context of a scenario

Because the notion of scenario is used in many different contexts outside of the domain of road traffic, a wide diversity in definitions of this notion exists. For an overview, see [31, 288]. Therefore, it is reasonable to assume that "there is no [generally] 'correct' scenario definition" [288]. As a result, to define the notion of scenario, it is important to consider the context in which it will be used.

In this dissertation, the context of a scenario is the assessment of AVs, where AVs refer to vehicles equipped with a driving automation system[3]. It is assumed that the assessment methodology uses scenario-based test cases. The ultimate goal is to build a database with all relevant scenarios that an AV has to cope with when driving in the real world [229]. Hence, a scenario should be a description of a potential use case of an AV.

Note that a scenario typically describes *what* happens. *How* it happens is typically not included in the scenario descriptions. For example, if a scenario contains a description of a complex trajectory of a vehicle, then the sudden reaction from the driver controlling the vehicle, the vehicle dynamics, and tire mechanics that lead to the complex trajectory do not need to be included in the content of a scenario.

2.3. Definitions

One of the main reasons to introduce an OOF is to enable sharing of knowledge between researchers, developers, and users. Therefore, it is important that the terms we use are clearly defined. When presenting the OOF in Section 2.4, we will formalize the terms such that they can be used by software agents. In this section, we define the terms *scenario*, *event*, *activity*, and *scenario category*, thereby providing insight into the terms used in the next section. We aim to provide intensional definitions that are in accordance with the common use of these terms in the literature and to provide clarity on what are the necessary and sufficient conditions for when the term should be used.

We first define the concept of a scenario in Section 2.3.1. Next, we define two important components of a scenario: events and activities, in Sections 2.3.2 and 2.3.3, respectively. Lastly, we present the definition of a scenario category in Section 2.3.4. Each of the Sections 2.3.1 to 2.3.4 starts with background information. Next, we draw conclusions that lead to our proposed definition of the corresponding term. After proposing a definition, each section finishes with remarks and implications of the proposed definition. For the definitions provided in Sections 2.3.1 to 2.3.4, use is made of the terms listed in Table 2.1. The definitions in Table 2.1 are mostly based on literature; see Section 2.A for more details.

2.3.1. Scenario

Go and Carroll [115] describe a scenario within the field of system design. They define a scenario as "a description that contains (1) actors, (2) background information on the actors and assumptions about their environment, (3) actors' goals or objectives, and (4) sequences of actions and events. Some applications may omit one of the elements or they may simply or implicitly express it. Although, in general, the elements of scenarios are the same in any field, the use of scenarios is quite different."

[3]According to [243], a driving automation system is "the hardware and software that are collectively capable of performing part or all of the dynamic driving task on a sustained basis. This term is used generically to describe any system capable of level 1-5 driving automation." Here, level 1 driving automation refers to "driver assistance" and level 5 refers to "full driving automation". For more details, see [243].

Table 2.1: Terms and definitions that are used in Section 2.3. For more details, see Section 2.A.

Term	Definition
Ego vehicle	Vehicle from which the world is perceived and/or vehicle that must perform a certain task during a test
Physical element	Object that exists in the three-dimensional space
Actor	Physical element that experiences change Note: An actor is a physical element, but a physical element is not necessarily an actor.
Static environment	Part of the environment that does not change
Dynamic environment	Part of the environment that does change and that comprises all actors
Act	Combination of an actor and an activity
State variables	Description of the present configuration of a system that can be used to determine the future response, given the excitation inputs and the equations describing the dynamics
State vector	Vector containing all n state variables
Model	Equations that describe the dynamics
Mode	Period in which a system does not exhibit a sudden change in an input, a model parameter, or the model

Geyer *et al.* [112] describe a scenario within the context of automated driving. They use the metaphor of a movie or a storybook for describing a scenario and state that "a scenario includes at least one situation within a scene including the scenery and dynamic elements. However, [a] scenario further includes the ongoing activity of one or both actors." Geyer *et al.* [112] define a scene "by a scenery, dynamic elements, and optional driving instructions." In [112], the meaning of activity is not detailed.

Ulbrich *et al.* [284] define a scenario as "the temporal development between several scenes in a sequence of scenes. Every scenario starts with an initial scene. Actions & events as well as goals & values may be specified to characterize this temporal development in a scenario. Other than a scene, a scenario spans a certain amount of time." The authors of [284] state that actions and events link the different scenes. A further description of actions and events is not given in [284].

Another definition of a scenario in the context of automated driving is given by Elrofai *et al.* [93]. They define a scenario as "the combination of actions and maneuvers of the host vehicle in the passive [i.e., static] environment, and the ongoing activities and maneuvers of the immediate surrounding active [i.e., dynamic] environment for a certain period of time."

Saigol *et al.* [245] define a scenario as "a description of a short interaction between an AV and other road users and/or road infrastructure".

In a concept paper on OpenSCENARIO 2.0 [19], a scenario is defined as "a 'description of the temporal development' of road users (actor entities) defined by

their actions, where temporal activation (defining when) 'is regulated by' conditional 'triggers'. A scenario comprises both scenery and dynamic elements."

As a basis for constructing a comprehensive definition for the concept of scenario, we list the major characteristics contained in the above definitions as follows:

1. *A scenario corresponds to a time interval.* The aforementioned definitions [93, 112, 115, 284] state that a scenario corresponds to a time interval. Van Notten *et al.* [288] call such a scenario a chain scenario ("like movies"), as opposed to a snapshot scenario, i.e., a scenario that describes the state at a given time instant ("like photos").

2. *A scenario consists of two or more events [112, 115, 154, 284, 288].* It can be helpful to develop scenarios using events [31]. Thus, a scenario could be defined as a particular sequence of events or, as Kahn [154, p. 143] writes, "a scenario results from an attempt to describe in more or less detail some hypothetical sequence of events". Furthermore, Geyer *et al.* [112] and Ulbrich *et al.* [284] use the notion of event for describing a scenario, although they do not provide a definition of the term *event*. Because a scenario contains at least a start event and an end event, the minimum number of events is two. In Section 2.3.2, we will elaborate on the notion of *event*.

3. *Real-world traffic scenarios are quantitative scenarios.* Regarding the nature of the data, a scenario can be either qualitative or quantitative [288]. For a real-world traffic scenario to be suitable for simulation purposes, it must be described quantitatively. A scenario, however, can also be described qualitatively, such that it is readable and understandable for human experts. Providing a qualitative description of a quantitative scenario has become known as a story-and-simulation approach [6]. Note that a qualitative description of a scenario does not uniquely define a quantitative scenario. A qualitative description can be regarded as an abstraction of the quantitative scenario, see also Section 2.3.4.

4. *The time interval of a scenario contains all relevant events.* According to Geyer *et al.* [112], "the end of a scenario is defined by the first irrelevant situation with respect to the scenario". In a similar manner, we require that the time interval of a scenario should contain all relevant events. Note that 'relevant' is subjective and, therefore, an event is considered to be relevant with respect to the perspective of one or more of the participating actors, often called the "ego vehicle".

5. *A scenario includes the description of the environment.* A scenario should include the description of the static and dynamic environment. Although the description of the static environment is not a general prerequisite of a scenario, this is often included when speaking about traffic scenarios [12, 89, 93, 112, 284]. The static environment consists of all relevant[4] physical elements

[4]The term 'relevant' is subjective and depends on the use of the scenario. The composer of a scenario typically judges whether something might be relevant for the scenario.

that do not undergo relevant changes with respect to the ego vehicle(s) within the time interval between the start and the end of the scenario. The dynamic environment consists of all relevant actors that undergo changes that are relevant to the ego vehicle(s). For example, the road may be part of the static environment, but if the change in the road temperature is relevant to the ego vehicle(s), the road is part of the dynamic environment.

6. *A scenario includes at least one ego vehicle [93, 112].* Because of the two previously mentioned characteristics, a scenario is required to include at least one ego vehicle. Note that an ego vehicle is often regarded as the device under test. In this chapter, however, this is not necessary because the ego vehicle is just the vehicle whose perspective is used to define what is relevant in the scenario.

7. *A scenario describes the goals or activities of the actors.* Either the activities, the goals, or a combination of activities and goals are required to determine how each actor in a scenario responds to specific events. Note that this also holds for an ego vehicle since an ego vehicle is an actor. When describing a scenario using real-world data, goals do not need to be given; e.g., Elrofai *et al.* [93] mention the activities of the actors rather than the goals. When describing a scenario that an AV has to cope with, however, the ego vehicle's goals (i.e., its driving mission [112]) could be specified rather than its activities [284]. Note that if the activities of an actor are described rather than its goals, an observer might not be able to determine whether the actor has successfully responded to the scenario.

Hence, we define a scenario as follows:

Definition 2.1 (Scenario)**.** *A scenario is a quantitative description of the relevant characteristics and activities and/or goals of the ego vehicle(s), the static environment, the dynamic environment, and all events that are relevant to the ego vehicle(s) within the time interval between the first and the last relevant event.*

When applying Definition 2.1 in an OOF, it is possible to give the "description" of a component of a scenario simply by providing a reference to that component. A reference could be, e.g., the full name of a file, a pointer pointing to a specific part of the computer memory, or an identifier that addresses a specific entry in a database. The advantage of references is that these parts of the scenario can be exchanged across different scenarios, as these scenarios can use the same references. As an example, an OpenSCENARIO file allows to provide a reference to an OpenDRIVE file that describes a road network [88]. As we will see in Section 2.4, in our proposed framework, a scenario may contain references to physical elements, activities, actors, and events.

2.3.2. Event

As mentioned in Definition 2.1, a scenario consists of events. The term event is used in many different fields, e.g.:

- In computing [38], an event is an action or occurrence recognized by software. A common source of events are inputs by the software users. An event may trigger a state transition.
- In probability theory, an event is an outcome or a set of outcomes of an experiment [226]. For example, a thrown coin landing on its tail is an event.
- In the field of hybrid systems theory, "the continuous and discrete dynamics interact at 'event' or 'trigger' times when the continuous state [vector] hits certain prescribed sets in the continuous state space" [37]. Moreover, "a hybrid system can be in one of several modes, [...], and the system switches from one mode to another due to the occurrence of events" [80].
- In the ISO 15926-2 standard, an ontology for long-term data integration, access, and exchange is specified in which an event is defined as "a *possible_individual*[5] with zero extent in time, which means that it occurs at an instant in time" [25].
- In event-based control, a control action is computed when an event is triggered, as opposed to the more traditional approach where a control action is periodically computed [131]. In event-based control, the event is triggered at the moment at which the system (is about to) reach a certain threshold.

Before providing the definition of an event, the following is concluded about an event, based on the aforementioned literature:

1. *An event corresponds to a time instant.* For the definition of event, we consider a hybrid-systems setting with a linear-time model [13]. Therefore, an event happens at some time instant.
2. *An event marks a mode transition or the moment a system reaches a threshold.* A mode transition may be induced by either an abrupt change of an input signal, a change of a parameter, a change in the model, or an external cause. It is also possible that the event marks the moment that a system reaches a threshold.

Hence, we define an event as follows:

Definition 2.2 (Event)**.** *An event corresponds to a moment at which a mode transition occurs or a system reaches a specified threshold, where the former can be induced by both internal and external causes.*

Definition 2.2 indicates that the moment of an event can be defined in two different ways: (1) by a mode transition or (2) by the system reaching a threshold. The first type could be a mode transition caused by a sudden driver input. An event might also be induced by an external cause, such as an environmental change. The second type of event, i.e., related to the system reaching a threshold, is especially

[5] "An entity that exists in space and time" [25].

useful when describing test scenarios. For example, consider the ego vehicle approaching a pedestrian that is about to cross the road [256]. Here, the event marks the moment that the distance between the vehicle and pedestrian is less than $d_{\mathrm{v,p}}$ meters. At the moment of this event, the pedestrian starts to cross the road such that the vehicle would impact with the pedestrian if it would not change its speed or direction [256]. By using a variable threshold $d_{\mathrm{v,p}}$, the value is flexible and can be set differently to define multiple scenarios.

For the practical implementation of events, a set of conditions may be specified. In that case, the event occurs at the moment that the conditions are met. In [20], an extensive list of possible conditions that can be used to define an event is given. For example, a condition could be that the distance between the vehicle and the pedestrian is below a certain threshold.

Remark 2.1. Geyer *et al.* [112] and Ulbrich *et al.* [284] use the term *scene* to define a scenario. Like an event, we consider a scene to correspond to a temporal snapshot of the entire scenario. A scene can be obtained by taking a temporal cross-section of the entire scenario as described in Definition 2.1. ◊

2.3.3. Activity

To describe the dynamic environment of a scenario, activities are used. A scenario may also describe the activities of an ego vehicle.

Both the terms activity [54, 94, 104, 112, 268] and action [23, 112, 284] are used in the context of automated driving. Although, strictly speaking, the terms action and activity have a slightly different meaning, they are often used for the same purpose:

- According to Ulbrich *et al.* [284], actions may be specified for characterizing the temporal development in a scenario.
- Elrofai *et al.* [94] consider an activity as a building block of the dynamic part of the scenario: "An activity is a time evolution of state variables such as speed and heading to describe for instance a lane change, or a braking-to-standstill."
- In a glossary for scenario catalog development [104], an activity is defined as "the state [vector] of an object over an interval of time. An activity starts with an event and ends with another event."
- In the ISO 15926-2 standard, an activity is defined as "a *possible_individual* that brings about change by causing the *event* that marks the *beginning*, or the *event* that marks the *ending* of a *possible_individual*" [25].

Before providing the definition of an activity, the following is concluded about an activity based on the aforementioned literature:

1. *An activity corresponds to an inter-event time interval.* As opposed to an event, an activity spans a certain time interval. Furthermore, the start and the end of an activity are marked by an event.

2. *An activity quantitatively describes the time evolution of one or more state variables.* Because activities are building blocks of a scenario and a scenario corresponds to a quantitative description, the activities themselves need to be quantitative as well. Therefore, an activity describes the time evolution of one or more state variables, i.e., the trajectory of one or more state variables over an inter-event time interval that corresponds to the activity, where the term state variable is defined in Table 2.1.

3. *An activity is performed by an actor.* An activity describes the time evolution of one or more state variables and a state variable corresponds to an actor, e.g., the acceleration of a vehicle.

Hence, we define an activity as follows:

Definition 2.3 (Activity)**.** *An activity is a quantitative description of the time evolution of one or more state variables of an actor between two events.*

As an example, an activity could describe the longitudinal acceleration (or, e.g., speed) during an acceleration or deceleration of an actor. Activities describing the lateral position of a vehicle with respect to the center of the corresponding lane might, e.g., be labeled with "driving straight" or "changing lane".

2.3.4. Scenario category

According to Definition 2.1, a scenario in the context of the performance assessment of an AV needs to be quantitative. However, in the literature, the term scenario is also used to refer to a collection of scenarios, where this collection of scenarios is described qualitatively. For example, in [209], a typology of pre-crash scenarios is proposed. Here, each of the pre-crash scenarios is an abstraction of many quantitative scenarios. Similar studies have been performed to describe scenarios that lead to highway accidents [98], car-cyclist accidents [216], and car-pedestrian accidents [180]. In [280], a taxonomy of scenarios is proposed to qualitatively describe challenging scenarios for automated driving. In [212], a distinction is made between so-called functional scenarios, abstract scenarios, logical scenarios, and concrete scenarios. These four types of scenario descriptions represent different levels of abstraction with functional scenarios referring to non-formal human-readable scenarios, abstract scenarios referring to formalized declarative descriptions, logical scenarios referring to parameterized scenarios with ranges and distributions of the parameters, and concrete scenarios referring to parameterized scenarios with fixed parameters values.

The aforementioned references [98, 180, 209, 212, 216, 280] show that the term *scenario* is also used to address qualitative descriptions. Since we define a scenario as a quantitative description, we need to introduce a different term to address the qualitative description. We propose to use the term *scenario category* to refer to the qualitative description of a scenario. A qualitative description can be regarded as an abstraction of a quantitative scenario, whereas a quantitative description can be regarded as a concretization of a qualitative description.

We thus define a scenario category as follows:

Definition 2.4 (Scenario category)**.** *A scenario category is a qualitative description of the relevant characteristics and activities and/or goals of the ego vehicle(s), the static environment, and the dynamic environment.*

Introducing the concept of scenario categories brings the following benefits:

- For a human, it is often easier to interpret a qualitative description than a quantitative description.
- Scenarios that have something in common can be grouped together, which enables characterization of types of scenarios and facilitates discussion of scenarios.
- The completeness of a set of scenarios can be assessed by considering the completeness of scenario categories (see, e.g., [129]) and the completeness of scenarios in each category (see, e.g., Chapter 3 [72]).

We describe the formal relation between a scenario and a scenario category with the verb "to comprise", denoted by $\ni$. If a specific scenario category $\mathcal{C}$ is an abstraction of a specific scenario S, then we say that $\mathcal{C}$ comprises S, or simply $\mathcal{C} \ni S$. A given scenario category can comprise multiple scenarios and multiple scenario categories can comprise a specific scenario. As a consequence, as opposed to the proposed categorization of scenarios in [177, 180, 209, 216], scenario categories do not need to be mutually exclusive.

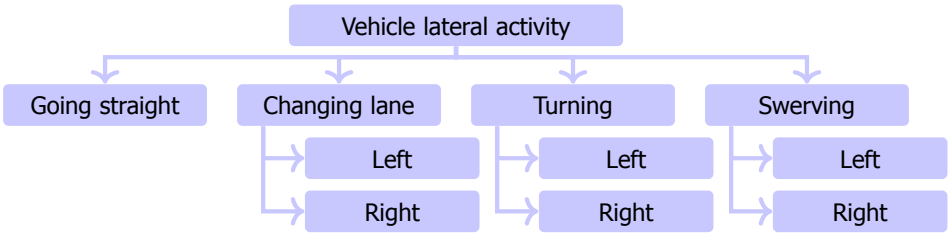
The verb "to include" is used to describe the relation between two scenario categories. A scenario category $\mathcal{C}_2$ is said to include a scenario category $\mathcal{C}_1$ if $\mathcal{C}_2$ comprises all scenarios that are comprised in $\mathcal{C}_1$. In that case, we can write $\mathcal{C}_2 \supseteq \mathcal{C}_1$. Thus we have

$$\mathcal{C}_2 \supseteq \mathcal{C}_1 \text{ if } \mathcal{C}_2 \ni S \ \forall \ \{S : \mathcal{C}_1 \ni S\}. \tag{2.1}$$

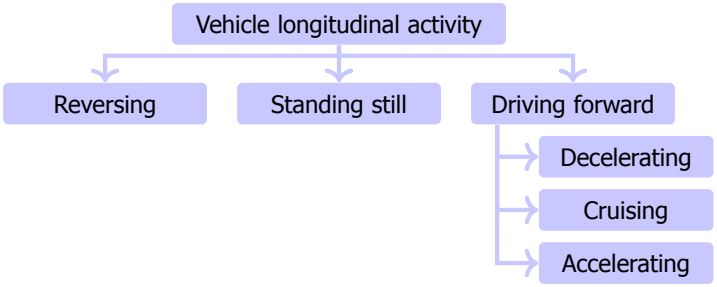
We propose to provide scenarios and scenario categories with additional information in the form of tags. A tag is a keyword or a keyphrase that provides extra information on a piece of data [264]. For example, items in a database can contain some tags that enable users to quickly retrieve several items that share a certain characteristic described by a tag [60]. The use of these tags brings several benefits:

- The tags of a scenario can be helpful in determining which scenario categories do and do not comprise the scenario.
- It is easy to select scenarios from a scenario database or a scenario library by using tags or a combination of tags.

There is a balance between having generic scenario categories — and thus a wide variety among the scenarios comprised by the scenario category — and having specific scenario categories without much variety among the scenarios comprised by the scenario category. For some systems, one may be interested in a very specific set of scenarios, while for another system one might be interested in a set of scenarios with a high variety. To accommodate this, tags can be structured



(a) Lateral activities of a vehicle.



(b) Longitudinal activities of a vehicle.

Figure 2.1: Tags for lateral and longitudinal activities of a vehicle [74]. The lateral activity is relative to the lane in which the corresponding vehicle is driving.

in hierarchical trees [201]. The different layers of the trees can be regarded as different abstraction levels [35].

Figure 2.1 shows two examples of trees of tags taken from [74]. These tags describe possible activities of a vehicle, i.e., the lateral motion control (via steering) and longitudinal motion control (via acceleration and deceleration). The tags may refer to the objective of an actor in case no activities are defined. For example, a test case in which the ego vehicle's objective is to make a left turn, the tags "Turning" and "Left" are applicable. Note that tags may be used not only to classify vehicle behavior, but also traffic and environment situations, e.g., "cut-in" or "heavy rain". For more examples of tags, see [68].

2.4. Object-oriented framework for scenarios

We have already explained the use of an OOF in Section 2.2.1. In this section, we present our OOF for scenarios for the assessment of AVs. The overview of the framework is formally represented through class diagrams that are briefly presented in Section 2.4.1. Next, Section 2.4.2 explains how a scenario category is formally represented in our framework. Similarly, in Section 2.4.3, we describe how a scenario is formally represented. The OOF can be implemented straightforwardly in object-oriented languages such as C++ and Python, since these languages support the definition of classes, the instantiation of objects from those classes, and concepts such as inheritance and aggregation. An actual implementation of the OOF in a coding language is publicly available at

https://github.com/ErwindeGelder/ScenarioDomainModel. This link also contains tutorials for the technical application of the OOF.

2.4.1. Class diagrams

In Figures 2.2 and 2.3, the blue blocks represent the classes[6] that are used to describe a scenario category according to Definition 2.4 and the red blocks represent the classes that are used to describe a scenario according to Definition 2.1. The green blocks represent so-called abstract classes. Abstract classes cannot be instantiated. Each class serves as a template for creating objects whereas an object of a particular class is referred to as the instance of that particular class.

Figure 2.2 shows the class-level relationships and Figure 2.3 shows the instance-level relationships. In Figure 2.2, the arrow from, e.g., *Scenario* to *Time interval*, denotes that *Scenario* is a subclass of *Time interval*. Therefore, all properties of the *Time interval* are inherited by *Scenario*. The arrow with the diamond in Figure 2.3 denotes an aggregation. This means that, e.g., an *actor*, which is an instance of the *Actor* class, has an *actor category* as an attribute. Here, the "1" at the start of the arrow from *Actor category* to *Actor* indicates that an *actor* has exactly one *actor category*. Similarly, "2" at the aggregation arrow from *Event* to *Time interval* indicates that a *time interval* contains two *events*, i.e., the events that define the start and the end of the time interval. A "$0, 1, \ldots$" at the start of an aggregation arrow indicates that an object has zero, one, or multiple objects of the corresponding class. The arrow with the text "comprises" and "includes" represent methods that are explained in Section 2.3.4. Here, "comprises" can be denoted by $\ni$ and "includes" can be denoted by $\supseteq$, see (2.1).

2.4.2. Scenario category and its attributes

Because all other classes in Figure 2.2 are subclasses of *Scenario element*, these classes inherit the attributes and procedures of *Scenario element*. In our framework, a *scenario element* has a human-interpretable name, a unique ID, and possibly predefined tags that are also interpretable by a software agent. So, all other classes in Figure 2.2 also have these attributes. In addition to these attributes, the *Qualitative element* class has a human-interpretable description.

The static environment is qualitatively described by one or more *physical element categories*. Because *physical element categories* qualitatively describe the static environment, they contain a human-interpretable description of the physical things they describe.

The ego vehicle(s) and the dynamic environment are qualitatively described by *activity categories* and *actor categories*. In line with Definition 2.3, *Activity category* includes the state variable(s). The *Model* that is used to describe the time evolution of the state variable(s) is specified. Note that *Model* is an abstract class that serves as a template for different models.

A *Model* may be a differential equation of the form $\dot{z}(t) = f_\theta(z(t), u(t), t)$ [214],

[6] In the remainder of this chapter, when referring to (an instance of) a class, italic font is used. Additionally, class names start with capital letters and instance names with lowercase letters.

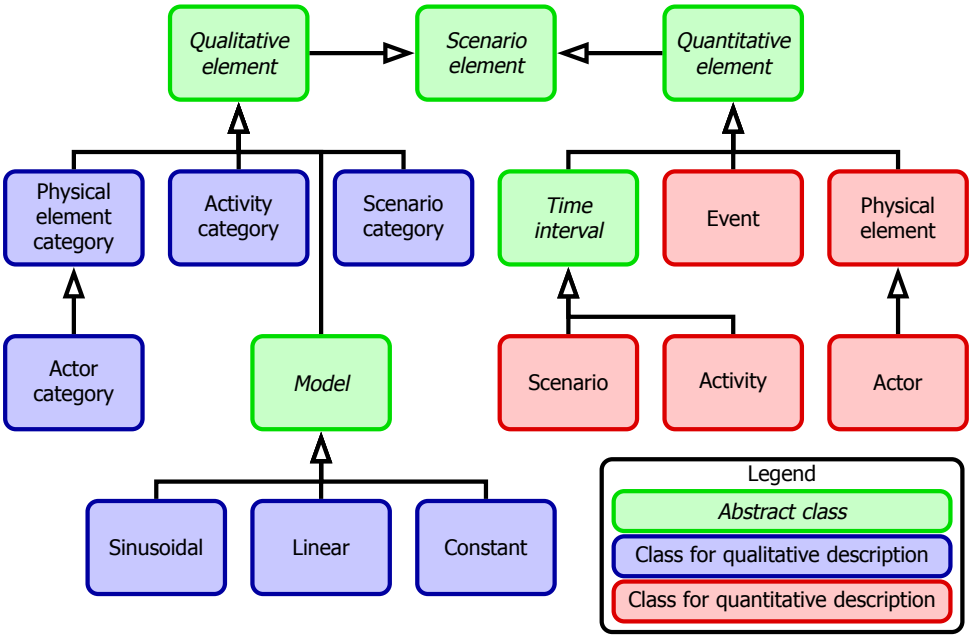


Figure 2.2: Class-level relationships of most classes of our Object-Oriented Framework (OOF).

where $z(t)$ represents the state vector at time t, $u(t)$ represents an external input vector, and the function $f_\theta(\cdot)$ is parameterized by θ. Note that, technically speaking, $z(\cdot)$, $u(\cdot)$, t, and θ are inputs of the function f, but θ is assumed to be constant for a certain time interval.

For illustration purposes, this work considers three examples of models, which are, as shown in Figure 2.2: *Sinusoidal*, *Linear*, and *Constant*. Note that more complex models are also possible, but since these models are not the focus of the current work, this is out-of-scope. The *Sinusoidal* model is defined as follows:

$$\dot{z}(t) = \frac{\pi A}{2T} \sin\left(\frac{\pi(t - t_0)}{T}\right), \; t \in [t_0, t_0 + T], \tag{2.2}$$

$$z(t_0) = z_0. \tag{2.3}$$

Here, the amplitude (A), duration (T), initial time (t_0), and initial state (z_0) are parameters. The *Linear* and *Constant* models are described by the following equations, respectively:

$$\dot{z}(t) = s, \; z(t_0) = z_0, \tag{2.4}$$

$$z(t) = z_0. \tag{2.5}$$

The *Linear* model contains three parameters, i.e., the slope (s), initial time (t_0), and initial state (z_0). The *Constant* model only has the parameter z_0. Since an *activity*

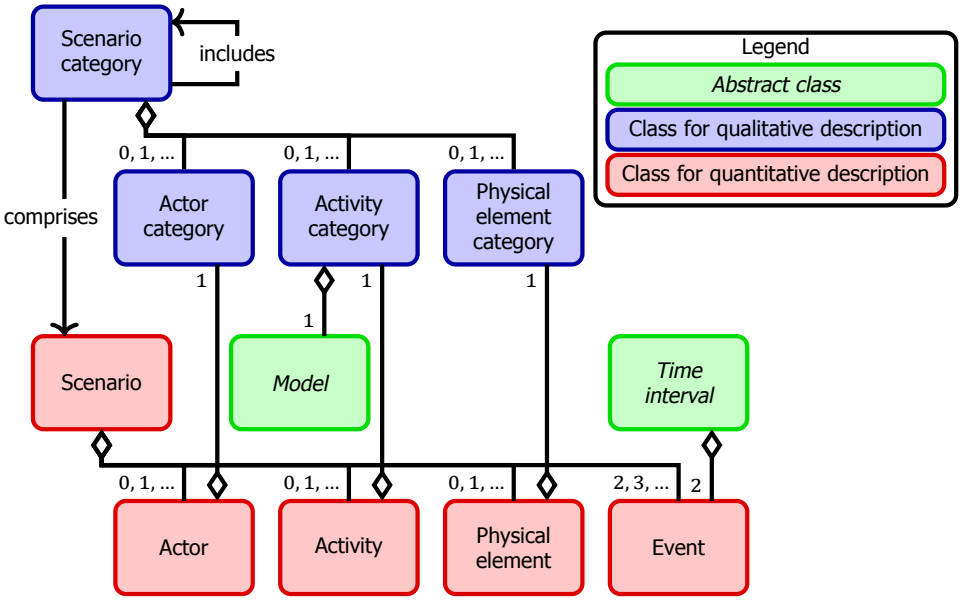


Figure 2.3: Instance-level relationships of most classes of our Object-Oriented Framework (OOF).

category is a qualitative description, the values of the parameters of its *model* are not part of the *activity category*. Note that this chapter only considers the models *Sinusoidal*, *Linear*, and *Constant*, but more complex models may be necessary to describe complex behavior. More complex models are out-of-scope of this chapter, but it is straightforward to extend the OOF with such models.

The *Actor category* is a subclass of *Physical element category* so *Actor category* inherits the properties of *Physical element category*. In addition, *Actor category* has an attribute that specifies the type of object. To indicate that an actor is an ego vehicle, the tag "Ego vehicle" is added to the list of tags of the *actor category*.

The *Scenario category* has *physical element categories*, *activity categories*, and *actor categories* as attributes. Another attribute of the *Scenario category* is the list of acts. These acts describe which actors perform which activities. Note that it is possible that one actor performs multiple activities and that one activity is performed by multiple actors.

The reader might wonder why we introduce the different classes for describing a scenario category, i.e., the blue blocks, instead of only one class for modeling a scenario category. The main advantage of the different classes is the reusability of the instances of the classes because these instances can be exchanged among different *scenario categories*. For example, if two *scenario categories* have the same *actor categories*, we only need to define the *actor categories* once, whereas if the *actor categories* would not be instances of a class but only properties of the scenario category, we would need to define the *actor categories* twice.

2.4.3. Scenario and its attributes

To distinguish objects that are directly used to compose a *scenario*, these objects are instantiated from subclasses of the *Quantitative element* class. The class *Scenario* is a subclass of *Time interval* and, therefore, it has *events* that define the start and the end of the scenario. The *Scenario* also has *physical element*, *activities*, *actors*, and *events* as attributes. The *physical elements*, *activities*, and *actors* are the quantitative counterparts of the *physical element categories*, *activity categories*, and *actor categories*, just as a *scenario* is the quantitative counterpart of a *scenario category*. As with the *Scenario category*, the *Scenario* contains a list of acts that describe which actors perform which activities.

A *physical element* has a *physical element category* and it may have multiple properties that quantitatively define the object, such as its size, weight, color, radar cross section, etc. Physical elements can be used to define, e.g., the road layout, static weather and lighting conditions, and infrastructural elements.

According to Definition 2.3, an activity quantitatively describes the evolution of one or more state variables during a time interval. The state variable(s) are defined by the *activity category* that the *activity* has as an attribute. Together with the *Model* that is contained by the *activity category*, the time evolution of the state variable(s) is described by a set of parameters. The values of the parameters are part of the *activity*.

Following Definition 2.2, an *event* contains conditions that describe the threshold or mode transition at the time of the *event*.

Similar to a *physical element* and an *activity*, an *actor* has its qualitative counterpart — an *actor category* — as an attribute. Additionally, the *Actor* contains an initial state vector and a desired state vector, that can be used to specify the intent, as attributes. Describing the intent is especially useful for defining a test scenario in terms of the objective of the ego vehicle rather than its activities.

An advantage of having the qualitative counterparts of the *Physical element*, *Activity*, and *Actor* is that the qualitative description can be reused and exchanged. For example, there can be many different braking activities, but there needs to be only one *activity category* for qualitatively defining the braking activity. Here, it is assumed that all braking activities are modeled with the same model and that similar tags apply. If this is not the case, multiple *activity categories* need to be defined, but the number of *activity categories* will still be substantially lower than the number of *activities*.

2.5. Example: pedestrian crossing

To illustrate the use of the OOF, we describe a scenario using objects of the classes presented in Section 2.4. The scenario is schematically shown in Figure 2.4. The ego vehicle is driving on the right lane of a two-lane road and a pedestrian is walking on a footway that intersects the road the ego vehicle is driving on. Both the ego vehicle and the pedestrian are initially approaching the pedestrian crossing. The ego vehicle brakes and comes to a full stop in front of the pedestrian crossing. While the ego vehicle is stationary, the pedestrian crosses the road using the pedestrian crossing. When the pedestrian has passed the ego vehicle, the ego

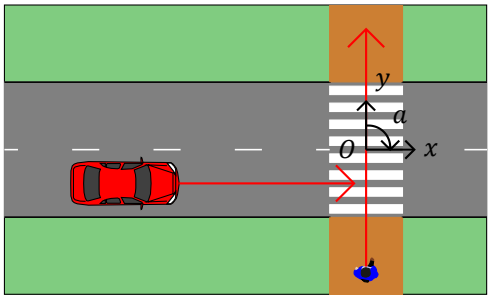


Figure 2.4: Schematic overview of a scenario where both the ego vehicle and a pedestrian are approaching a non-signalized pedestrian crossing. The pedestrian has priority.

vehicle accelerates. The code of this example is publicly available[7].

This particular scenario can be used to formulate a test scenario for the assessment of an AV. For example, when assessing a pedestrian automatic emergency braking system [256], we are interested in the behavior of the system in case the driver or automation system of the ego vehicle does not brake.

We first describe the scenario qualitatively using our proposed framework. Next, the scenario is described quantitatively in Section 2.5.2. In Section 2.5.3, we show which objects are reused and which objects are different if we consider an actual test scenario with a crossing pedestrian.

2.5.1. Qualitative description of the pedestrian crossing

To describe the scenario according to the presented OOF, objects are instantiated from the classes presented in Figures 2.2 and 2.3. Figure 2.5 shows the objects for describing the scenario qualitatively. There are two *actor categories*: one for the ego vehicle and one for the pedestrian. Four different *activity categories* are defined: *braking*, *stationary*, *accelerating*, and *walking straight*. The *braking*, *stationary*, and *accelerating activity categories* contain the state variable v_{ego}, i.e., the speed of the ego vehicle, and use the *Sinusoidal* model of (2.2) and (2.3), the *Constant* model of (2.5), and the *Linear* model of (2.4), respectively. The *activity category walking straight* has the position of the pedestrian (y_{ped}) as its state variable and uses the *Linear* model of (2.4).

The two *actor categories*, the four *activity categories*, and the *physical element category* that represents the crosswalk are used by the *scenario category*. The *scenario category* has four acts. The first three acts assign the first three *activity categories* to the ego vehicle. The last act assigns the *activity category walking straight* to the pedestrian.

[7] See `https://github.com/ErwindeGelder/ScenarioDomainModel`. The repository also contains other examples.

Crossing pedestrian::Scenario category
description: A pedestrian is crossing the road on a zebra crossing in front of the ego vehicle
physical element: Pedestrian crossing qualitative
actors: Ego qualitative, Pedestrian qualitative
activities: Braking, Stationary, Accelerating, Walking straight
acts: (Ego qualitative, Braking), (Ego qualitative, Stationary), (Ego qualitative, Accelerating), (Pedestrian qualitative, Walking straight)
tags:

Ego qualitative::Actor category
type: Vehicle
tags: Ego vehicle, Passenger car

Pedestrian qualitative::Actor category
type: Pedestrian
tags: Pedestrian

Braking::Activity category
model: Sinusoidal
state variable: Speed (v_{ego})
tags: Decelerating

Stationary::Activity category
model: Constant
state variable: Speed (v_{ego})
tags: Stationary

Accelerating::Activity category
model: Linear
state variable: Speed (v_{ego})
tags: Accelerating

Walking straight::Activity category
model: Linear
state variable: Position (y_{ped})
tags: Walking straight

Pedestrian crossing qualitative::Physical element category
description: Straight road with two lanes and a pedestrian crossing
tags: Non-signalized zebra crossing

Figure 2.5: The objects that are used to qualitatively describe the scenario that is schematically shown in Figure 2.4. The first line of each block shows the name (before the double colon) and the class from which the object is instantiated. The following lines show the attributes of the object with the name and value of the attribute before and after the colon, respectively. For the sake of brevity, the unique ID of each object is omitted.

2.5.2. Quantitative description of the pedestrian crossing

The objects to describe the scenario quantitatively are shown in Figure 2.6. The two *actors* refer to the quantitative counterparts of the *actor categories* in Figure 2.5. Initial state vectors are listed for each *actor* using the coordinate frame that is shown in Figure 2.4. Since we are describing a real-world scenario, there is no need to define goals or intents for the actors.

There are four *events* defined. These *events* mark the time instants that define the start and the end of the *activities*. For simplicity, it is assumed that the start of the scenario occurs at $0\,\mathrm{s}$.

There are four *activities* defined and each of these *activities* refers to its qualitative counterpart. The *activities* contain the values of the parameters as well as events that mark the start and the end of the *activities*. As described by the first *activity* (*ego braking*), the ego vehicle starts with a speed of $8\,\mathrm{m/s}$ and brakes in $4\,\mathrm{s}$ to come to a full stop. By integrating the sinusoidal function of (2.2) twice, it can be shown that the ego vehicle stops at $4\,\mathrm{m}$ from the center of the pedestrian crossing. After waiting for $3\,\mathrm{s}$ as described by the second *activity* (*ego stationary*), the ego vehicle accelerates with $1.5\,\mathrm{m/s^2}$ towards a speed of $7.5\,\mathrm{m/s}$ as described by the third *activity* (*ego accelerating*). The fourth *activity* describes the position

Ego brakes for pedestrian::Scenario
physical element: Pedestrian crossing
actors: Ego, Pedestrian
activities: Ego braking, Ego stationary, Ego accelerating, Pedestrian walking
acts: (Ego, Ego braking), (Ego, Ego stationary), (Ego, Ego accelerating), (Pedestrian, Pedestrian walking)
start event: Start scenario
end event: End scenario

Ego::Actor
actor category: Ego qualitative
properties: {width=1.8 m, length=4.5 m}
initial state vector: $x_{\text{ego}} = -20\,\text{m}$, $y_{\text{ego}} = -1.5\,\text{m}$, $a_{\text{ego}} = 90°$
desired state vector:

Pedestrian::Actor
actor category: Pedestrian qualitative
properties: {width=0.5 m, color=blue}
initial state vector: $x_{\text{ped}} = 0\,\text{m}$, $a_{\text{ped}} = 0°$
desired state vector:

Start scenario::Event
time: 0 s

End braking::Event
time: 4 s

Start accelerating::Event
time: 7 s

End scenario::Event
time: 12 s

Ego braking::Activity
activity category: Braking
parameters: $A = -8\,\text{m/s}$, $T = 4\,\text{s}$, $z_0 = 8\,\text{m/s}$, $t_0 = 0\,\text{s}$
start event: Start scenario
end event: End braking

Ego stationary::Activity
activity category: Stationary
parameters: $z_0 = 0\,\text{m/s}$
start event: End braking
end event: Start accelerating

Ego accelerating::Activity
activity category: Accelerating
parameters: $s = 1.5\,\text{m/s}^2$, $z_0 = 0\,\text{m/s}$, $t_0 = 7\,\text{s}$
start event: Start accelerating
end event: End scenario

Pedestrian walking::Activity
activity category: Walking
parameters: $s = 1\,\text{m/s}$, $z_0 = -6\,\text{m}$ $t_0 = 0\,\text{s}$
start event: Start scenario
end event: End scenario

Pedestrian crossing::Physical element
physical element category: pedestrian crossing qualitative
properties: {road: {lanes: 2, lanewidth: 3 m, xy: [(−60, 0), (60, 0)]}, footway: {width: 3 m, xy: [(0, 6), (0, −6)]}}

Figure 2.6: The objects that are used to quantitatively describe the scenario that is schematically shown in Figure 2.4. For the sake of brevity, the tags and the unique ID of each object are omitted.

and speed of the pedestrian.

The *pedestrian crossing* describes the entire static environment, including the main road the ego vehicle is driving on and the footway the pedestrian is walking on. The example in Figure 2.6 shows some properties of the road layout to illustrate how the static environment can be described. Note that, in practice, the quantitative description of the static environment may contain many more facets than the ones mentioned in Figure 2.6. As mentioned in Section 2.3.1, it is possible to refer to another source that contains a description of (part of) the static environment, see, e.g., [88].

The *scenario* has the previously defined *physical element*, *actors*, and *activities* as attributes. The acts are used to assign the first three *activities* to the ego vehicle and the last *activity* to the pedestrian. The *scenario* also has *events* marking the

Crossing pedestrian::Scenario category
description: A pedestrian is crossing the road on a zebra crossing in front of the ego vehicle
physical element: Pedestrian crossing qualitative
actors: Ego qualitative, Pedestrian qualitative
activities: Walking straight
acts: (Pedestrian qualitative, Walking straight)

Ego must brake for pedestrian::Scenario
physical element: Pedestrian crossing
actors: Ego, Pedestrian
activities: Pedestrian walking
acts: (Pedestrian, Pedestrian walking)
events: Start walking, Stop walking
start event: Start scenario
end event: End scenario

Start walking::Event
condition: $|x_{\text{ego}}/v_{\text{ego}}| \leq 2.5\,\text{s}$

End walking::Event
contition: $y_{\text{ped}} = 6\,\text{m}$

Ego::Actor
actor category: Ego qualitative
properties: {width=1.8 m, length=4.5 m}
initial state vector: $x_{\text{ego}} = -60\,\text{m}$, $y_{\text{ego}} = -1.5\,\text{m}$, $a_{\text{ego}} = 90°$, $v_{\text{ego}} = 8\,\text{m/s}$
desired state vector: $x_{\text{ego}} = 20\,\text{m}$, $y_{\text{ego}} = -1.5\,\text{m}$, $a_{\text{ego}} = 90°$, $v_{\text{ego}} = 8\,\text{m/s}$

Pedestrian walking::Activity
activity category: Walking
parameters: $s = 1\,\text{m/s}$, $z_0 = -6\,\text{m}$, t_0 = At Start walking
start event: Start walking
end event: End walking

End scenario::Event
condition: $x_{\text{ego}} \geq 20\,\text{m}$ OR *collision* OR $y_{\text{ego}} \leq -2\,\text{m}$ OR $y_{\text{ego}} \geq -1\,\text{m}$ OR $t > 100\,\text{s}$.

Figure 2.7: The objects that, together with the objects *Ego qualitative*, *Pedestrian qualitative*, *Walking straight*, and *Pedestrian crossing qualitative* from Figure 2.5 and *Start scenario*, *Pedestrian*, and *Pedestrian crossing* from Figure 2.6, describe a test scenario that is schematically shown in Figure 2.4. For the sake of brevity, the tags and the unique ID of each object are omitted.

start and the end of the *scenario*. A different *scenario* can be defined by, e.g., changing the parameter values. This illustrates that the *scenario category* in Figure 2.5 comprises multiple *scenarios*, including the *scenarios* that only differ from the *scenario* in Figure 2.6 because of different parameter values.

2.5.3. Test scenario of the pedestrian crossing

In this example, we consider a test scenario based on the previously illustrated real-world scenario, see Figure 2.4. To describe the test scenario, we reuse the two *actor categories* from Figure 2.5 (*ego qualitative* and *pedestrian qualitative*) and the *actor* describing the pedestrian from Figure 2.6 (*pedestrian crossing*). Figure 2.7 shows the other objects that are used to describe this test scenario.

The *scenario category* only differs from the *scenario category* shown in Figure 2.5 in that it does not contain *activity categories* that describe the activity of the ego vehicle.

Two attributes of the quantitative description of the ego vehicle are different. First, the initial state vector also includes the speed, denoted by v_{ego}, at the start of the scenario and the initial position is further away from the pedestrian crossing, such that the ego vehicle's driver or automation system has more time to perceive the pedestrian. Second, because there are no activities defined for the ego vehicle, the desired state vector is defined. The goal is to reach the point $80\,\text{m}$ in front of

the ego vehicle while driving with a speed of $v_{\text{ego}} = 8\,\text{m/s}$.

The *event* that marks the start of the walking activity of the pedestrian is triggered if the ego vehicle is $2.5\,\text{s}$ away from the center of the footway, assuming that the speed of the ego vehicle is constant. In case the ego vehicle drives with a speed of $v_{\text{ego}} = 8\,\text{m/s}$, this is at a distance of $20\,\text{m}$, similar to the scenario described in Figure 2.6.

As with the *scenario category*, the *scenario* does not contain *activities* of the ego vehicle. Furthermore, the end event of the scenario is defined differently: now the scenario ends if the ego vehicle either reaches its destination ($x_{\text{ego}} \geq 20\,\text{m}$), collides with the pedestrian, deviates too much from its path ($y_{\text{ego}} \leq -2\,\text{m}$ or $y_{\text{ego}} \geq 1\,\text{m}$), or takes too long to reach the destination ($t > 100\,\text{s}$).

Note that this example considers a pedestrian that crosses the road at a fixed speed ($1\,\text{m/s}$) regardless of the proximity of the ego vehicle. To model, e.g., the case where the pedestrian notices the ego vehicle and accelerates if a collision is about to happen, an activity can be added that describes the increased speed (e.g., $2\,\text{m/s}$) of the pedestrian. The start of this activity is at a predefined event with, e.g., the condition $|x_{\text{ego}}/v_{\text{ego}}| \leq 1\,\text{s}$ AND $y_{\text{ped}} < 0\,\text{m}$.

2.5.4. Remarks on the example

The example illustrates the benefits of the object-oriented approach for defining a scenario, which are:

- clarity regarding the content of the scenario,
- modularity, which makes it easy to understand the changes from the real-world scenario in Figure 2.6 to the test scenario in Figure 2.7, and
- reusability, as is illustrated by the objects that are used more than once.

Furthermore, each object listed in Figures 2.5 to 2.7 is directly translatable to an object in an object-oriented programming language. As a further illustration that the presented OOF is practical to use in real life, the framework is used by TNO's StreetWise program for storing real-world scenarios in a database [94][8].

In the example, two different actors are considered: the ego vehicle and the pedestrian. These are examples of traffic participants, but an actor is not necessarily a traffic participant. For example, road side units that transmit messages in an infrastructure-to-vehicle communication setting can also be actors. In this case, the transmission of messages can be considered as an activity. Another example of an actor is the road surface in case it is important for the scenario to model the changing surface temperature.

Note that the main purpose of the current example is to illustrate the use of the OOF. For the sake of brevity, the current example is a simplified description of a typical description of a scenario. Relevant aspects that are not considered in the current example include, but are not limited to, weather conditions, lighting conditions, visibility conditions, road friction, road surface markings, road edge, and height differences.

[8] See also `https://www.tno.nl/streetwise`.

2.6. Conclusions

The performance assessment of Automated Vehicles (AVs) is essential for the legal and public acceptance of AVs as well as for the technology development of AVs. Because scenarios are crucial for the assessment, a clear definition of a scenario is required. In this work, we have proposed a new definition of the concept scenario in the context of the performance assessment AVs.

While our definition is consistent with other definitions from the literature, it is more concrete such that it can directly be implemented using code. We have further defined the notions of event, activity, and scenario category. To formalize the concepts of scenario, event, activity, and scenario category, an Object-Oriented Framework (OOF) has been proposed. Using the proposed framework, it is possible to describe a scenario in both a qualitative and quantitative manner. The framework, represented using class diagrams, can be directly translated into a class structure for an object-oriented software implementation. This allows us to translate scenarios into code, such that both domain experts and software programs, such as simulation tools, are able to understand the content of the scenarios. To demonstrate this, we have made our implementation in the coding language Python publicly available.

The OOF has been illustrated with an example of an urban scenario with a pedestrian crossing. We have also demonstrated how this particular scenario can be used to define a test scenario using the proposed framework. In the publicly available[9] coding implementation of the presented OOF, we have shown how to use the proposed OOF from a real application's perspective.

The presented OOF is applicable for scenario mining [73, 221] and scenario-based assessment [94, 229] and, therefore, this framework provides a step towards scenario-based performance assessment of AVs. The next step is to define scenarios and scenario categories[10] that are relevant for an AV in a specific deployment area. Other future work is the expansion of the OOF, for example, by including distributions and/or uncertainties of the parameters of the activities or the properties of the physical elements. Future work also includes creating an ontology for scenarios for the assessment of AVs. The presented OOF could be a good starting point for this [263]. An ontology allows, among others, to add properties to relationships that enable automated reasoning. In this way, an ontology enables automated classification of scenarios, thereby helping to overcome problems of data ambiguity [19].

2.A. Nomenclature

For the definition of *scenario*, several notions are adopted from the literature. In this section, the concepts of *ego vehicle*, *physical element*, *actor*, *static environment*, *dynamic environment*, *act*, *state variable*, *state vector*, *model*, and *mode*, which are adopted from the literature, are detailed.

[9] https://github.com/ErwindeGelder/ScenarioDomainModel
[10] As a starting point, the 67 scenario categories in [74] can be used.

2.A.1. Ego vehicle

The ego vehicle is the main subject of a scenario. In particular, the ego vehicle refers to the vehicle that is perceiving the world through its sensors (see, e.g., [35]). When performing tests, the ego vehicle also refers to the vehicle that must perform a specific task (see, e.g., [12, 104]). In this case, the ego vehicle is often referred to as the system under test [270], the vehicle under test [114], or the host vehicle [114].

2.A.2. Physical element

A physical element refers to an object that exists in the three-dimensional space.

2.A.3. Actor

According to Frost [104], "actors are all dynamic components of a scenario, excluding the ego vehicle itself." Note that, in contrast to [104], in this work, the ego vehicle's driver, and/or automation system are considered as actors, similar to [112], because they have the same properties as another driver or automation system. While the aforementioned definition of Frost [104] provides a good idea of what an actor could be, we use another definition in order to avoid a circular definition: an actor is a dynamic physical element, i.e., a physical element that experiences change.

Remark 2.2. An actor is also a physical element whereas a physical element is not necessarily an actor. For example, a static road sign is considered a physical element, but because it does not change during the course of a scenario, it is not an actor. ◊

2.A.4. Static environment

The static environment refers to the part of the environment that does not experience change during a scenario. This includes geo-spatially stationary elements [284], such as the road network.

2.A.5. Dynamic environment

As opposed to the static environment, the dynamic environment refers to the part of the environment that changes during the time frame of a scenario. In practice, the dynamic environment mainly consists of the moving actors (other than the ego vehicle) that are relevant to the ego vehicle. For example, the primary use case of OpenSCENARIO [20], a file format for the description of the dynamic content of driving simulations, is to describe "complex, synchronized maneuvers that involve multiple entities like vehicles, pedestrians, and other traffic participants" [20]; so for OpenSCENARIO, these maneuvers represent the dynamic environment. Roadside units that communicate with vehicles within the communication range [5] are also part of the dynamic environment. Furthermore, changing (weather) conditions are part of the dynamic environment.

Remark 2.3. Note that it might not always be obvious whether an element of the environment belongs to the static or dynamic environment. Most important, how-

ever, is that all parts of the environment that are relevant to the assessment of an AV are described in either the static or the dynamic environment. ◊

2.A.6. Act

We define an act as a combination of an actor and the activity that is performed by the actor or the combination of actors and the activities they are subjected to. This is in accordance with the use of the term *act* in [20].

2.A.7. State variable

Dorf and Bishop [83, p. 163] write that "the state variables describe the present configuration of a system and can be used to determine the future response, given the excitation inputs and the equations describing the dynamics." In our case, "the system" could refer to an actor, a component, or a simulation. For example, a state variable could be the acceleration of an actor.

2.A.8. State vector

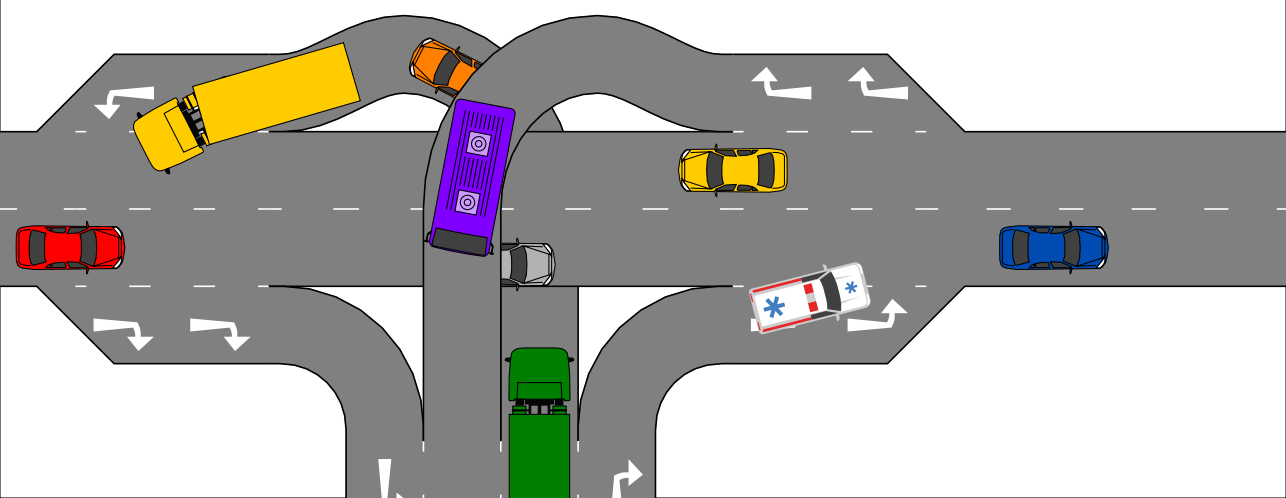
A state vector refers to "the vector containing all n state variables" [83, p. 233].

2.A.9. Model

The model refers to the equations that describe the dynamics. This may be algebraic equations, ordinary or partial differential equations, time-delay models, etc.

2.A.10. Mode

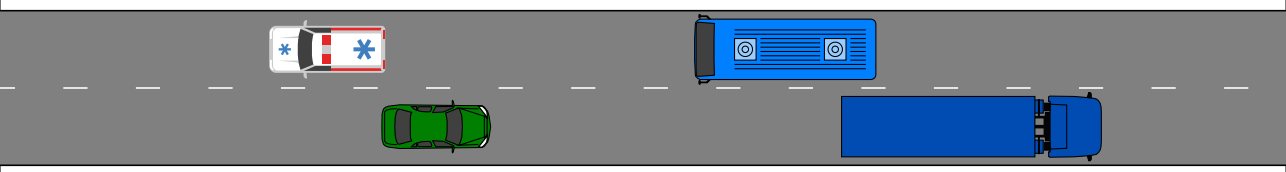
In some systems, the behavior of the system may suddenly change abruptly, e.g., due to a sudden change in an input, a model parameter, or the model. Such a transition is called a mode switch. In each mode, the behavior of the system is described by a model with a fixed function f_θ and smooth input $u(\cdot)$ [80].



3

Quantifying whether we have collected enough field data

This chapter is based on:
E. de Gelder, J.-P. Paardekooper, O. Op den Camp, and B. De Schutter, *Safety assessment of automated vehicles: How to determine whether we have collected enough field data?* Traffic Injury Prevention **20**, 162 (2019).



The amount of collected field data from naturalistic driving studies is quickly increasing. The data are used for, among others, developing automated driving technologies (such as crash avoidance systems), studying driver interaction with such technologies, and gaining insights into the variety of scenarios in real-world traffic. Because data collection is time consuming and requires high investments and resources, questions like "Do we have enough data?," "How much more information can we gain when obtaining more data?," and "How far are we from obtaining completeness?" are highly relevant. In fact, deducing safety claims based on collected data — for example, through testing scenarios based on collected data — requires knowledge about the degree of completeness of the data used. We propose a method for quantifying the completeness of the so-called activities in a data set. This enables us to partly answer the aforementioned questions.

In this chapter, the (traffic) data are interpreted as a sequence of different so-called scenarios that can be grouped into a finite set of scenario categories. The building blocks of scenarios are the activities. For every activity, there exists a parameterization that encodes the information in the data of each recorded activity. For each type of activity, we estimate a probability density function (pdf) of the associated parameters. Our proposed method quantifies the degree of completeness of a data set using the estimated pdfs.

To illustrate the proposed method, two different case studies are presented. First, a case study with an artificial data set, of which the underlying pdfs are known, is carried out to illustrate that the proposed method correctly quantifies the completeness of the activities. Next, a case study with real-world data is performed to quantify the degree of completeness of the acquired data for which the true pdfs are unknown.

The presented case studies illustrate that the proposed method is able to quantify the degree of completeness of a small set of field data and can be used to deduce whether sufficient data have been collected for the purpose of the field study. Future work will focus on applying the proposed method to larger data sets. The proposed method will be used to evaluate the level of completeness of the data collection on public roads, aimed at defining relevant test cases for the autonomous vehicle road approval procedure.

3.1. Introduction

The amount of collected field data from driving studies is increasing rapidly and these data are extensively used for the research, development, assessment, and evaluation of driving-related topics [40, 70, 81, 94, 163, 171, 228, 229, 242, 304, 327]. For any work that depends on data, it is important to know how complete the data are. As mentioned by various authors [14, 112, 270], especially when deducing safety claims based on collected data, e.g., through testing scenarios based on collected data, we require knowledge about the degree of completeness of the data set used. Hence, questions like "do we have enough data?" are highly relevant when our work and conclusions depend on the data. Furthermore, since the collection of data is time-consuming and requires high investments and resources, we should ask ourselves "how much more data do we need?" or "how much more information can we gain when obtaining more data?"

The aforementioned questions are already explored in other fields [32, 122, 194, 298, 316], but the question of how much data are enough regarding traffic-related applications is less frequently answered. Wang *et al.* [298] appear to be the first in the literature to point out and discuss issues concerning the amount of data needed to understand and model driver behaviors. They propose a statistical approach to determine how much naturalistic driving data are enough for understanding driving behaviors. For scenario-based assessments [14, 94, 112, 228, 270], however, the approach of Wang *et al.* [298] might not be applicable because they only consider the individual measurements at consecutive time instants instead of taking into account the whole driving scenario. Hence, there is a need for a quantitative measure for the completeness of a data set that takes into account the different scenarios a vehicle encounters in real-world traffic.

We describe a method for quantifying the completeness of a data set. The data are interpreted as a sequence of different scenarios that can be grouped into a finite set of scenario categories. Activities, such as "braking" and "lane change" form the building blocks of the scenarios [94]. For every activity, we create a parameterization that encodes the information in the data of this activity. For each type of activity, we estimate a probability density function (pdf) of the associated parameters. Our proposed method approximates the degree of completeness of a data set using the expected error of the estimated pdf. The smaller this error, the higher the degree of completeness.

To illustrate the proposed method, two different case studies are presented. The first case study involves artificial data of which the underlying distributions are known. Because the underlying distributions are known, we can show that the proposed method correctly quantifies the degree of completeness. Next, a case study with real-world data is performed to quantify the degree of completeness of the acquired data for which the underlying distributions are unknown. Additionally, we show how we can estimate the required amount of data to meet a certain requirement.

This chapter is structured as follows. Section 3.2 describes in more detail the problem for which a solution is proposed in Section 3.3. The two case studies are presented in Section 3.4. After a discussion in Section 3.5, this chapter is concluded

in Section 3.6.

3.2. Problem definition

The required amount of data depends on the use of the data [298]. For example, when investigating (near)-accident scenarios from naturalistic driving data, more data might be required compared to studying nominal driving behavior because of the low probability of having a (near)-accident scenario in naturalistic driving data. Therefore, in this chapter, the goal is to define a quantitative measure for the completeness of the data that can be used to determine whether the data are enough.

To define the problem of quantifying the completeness of the data, few assumptions are made:

1. The data are interpreted as an infinite number of possible scenarios, where scenarios might overlap in time [94]. Several definitions of the term scenario in the context of traffic data have been proposed in the literature, e.g., by Elrofai *et al.* [93, 94], Geyer *et al.* [112], Ulbrich *et al.* [284]. Because we want to differentiate between quantitative and qualitative descriptions, the definition of the term scenario is adopted from Elrofai *et al.* [94] as it explicitly defines a scenario as a quantitative description: "A scenario is a quantitative description of the ego vehicle, its activities and/or goals, its dynamic environment (consisting of traffic environment and conditions) and its static environment. From the perspective of the ego vehicle, a scenario contains all relevant events." Extracting scenarios from data received significant attention and the applied methods are very diverse. For example, Elrofai *et al.* [93] use a model-based approach to detect scenarios in which the ego vehicle is changing lane, whereas Kasper *et al.* [157] use Bayesian networks to detect scenarios with lane changes of other vehicles around the ego vehicle. Xie *et al.* [313] use a random forest classifier for extracting various scenarios and Paardekooper *et al.* [221] employ rule-based algorithms for scenario extraction.

2. Just as Elrofai *et al.* [94], we assume that a scenario consists of activities: "An activity is considered [to be] the smallest building block of the dynamic part of the scenario (maneuver of the ego vehicle and the dynamic environment)." An activity describes the time evolution of state variables. For example, an activity can be "braking", where the activity describes the evolution of the speed over time. Furthermore, "the end of an activity marks the start of the next activity" [94]. For more information, see Section 2.3.3.

3. Though a scenario refers to a quantitative description, these scenarios can be abstracted by means of a qualitative description, referred to as scenario category; see also Section 2.3.4 and [79, 94, 228]. An example of a scenario category could have the name "ego vehicle braking"; that is, this scenario qualitatively describes a scenario in which the ego vehicle brakes. An actual (real-world) scenario in which the ego vehicle is braking would fall into this

scenario category. It is assumed that all scenarios can be categorized into these scenario categories. This assumption does not limit the applicability of this chapter, though it might require a large number of scenario categories to describe all scenarios that are in the data.

4. It is assumed that all scenarios that fall into a specific scenario category can be parameterized similarly. As a result, the specific activities that constitute the scenario are also parameterized similarly. As with the previous assumption, this does not limit the applicability of this chapter. However, it might constrain the variety of scenarios that fall into a scenario category.

Using these assumptions, we can describe the problem of quantifying the completeness of a data set into three subproblems:

1. How to quantify the completeness regarding the scenario categories?
2. How to quantify the completeness regarding all scenarios that fall into a specific scenario category?
3. How to quantify the completeness regarding the activities?

The first step towards quantification of the completeness of the data is to assess the completeness of the activities. The next step is to quantify the completeness of the scenarios, i.e., the combinations of activities. The final step is to quantify the completeness of the scenario categories. In this chapter, the first step, i.e., the third subproblem, is addressed. Because of the different approach required for answering the first and second subproblem, those will be addressed in a forthcoming paper.

3.3. Method

In this section, we present how to quantify the completeness regarding the activities. As explained in Section 3.2, all scenarios that fall into a specific scenario category are parameterized similarly. Therefore, all similar types of activities are also parameterized similarly. For example, all activities labeled "braking" are parameterized similarly. In the remainder of this section, we assume that all activities that we are dealing with are a similar type of activities, such that they are parameterized similarly.

Considering one type of activity, let n denote the number of recorded activities of that type of activity. Each activity is described by a parameter vector. Let the parameters of the i-th activity be denoted by $X_i \in \mathbb{R}^d$ with $i \in \{1, \dots, n\}$ and d denoting the number of parameters for one activity. We will estimate the underlying distribution of X_i. Let $f(\cdot)$ denote the true pdf and let $f(x)$ denote the probability density evaluated at x. Similarly, let $\hat{f}(\cdot; n)$ denote the estimated pdf using n parameter vectors.

To quantify the completeness of the collection of the n activities, we use the estimated pdf $\hat{f}(\cdot; n)$. For example, suppose that $\hat{f}(x; n)$ equals $f(x)$ for all $x \in \mathbb{R}^d$. In this case, it would be reasonable to say that the n activities give a complete view of the variety and the distribution of the different activities that are labeled

similarly. On the other hand, when $\hat{f}(x;n)$ is very different from $f(x)$, it would be reasonable to say that the opposite is the case, i.e., the n scenarios do not give a complete view. One common measure for comparing the estimated pdf with the true pdf is the Mean Integrated Squared Error (MISE):

$$\mathrm{MISE}_f(n) = \mathbb{E}\left[\int_{\mathbb{R}^d} \left(f(x) - \hat{f}(x;n)\right)^2 \mathrm{d}x\right]. \tag{3.1}$$

The index f indicates that the MISE is calculated with respect to the pdf $f(\cdot)$.

A low MISE indicates a high degree of completeness whereas a high MISE indicates a low degree of completeness because the expected integrated squared error is high. Therefore, the MISE can be used to quantify the completeness of set of activities that are of a similar type. The problem is, however, that (3.1) depends on the true pdf $f(\cdot)$ which is unknown. So the MISE of (3.1) cannot be evaluated.

In the remainder of this section, we will explain how the MISE of (3.1) can be estimated when Kernel Density Estimation (KDE) is employed. First, KDE will be explained. Next, in Section 3.3.2, a method is presented for estimating the MISE when assuming that the d parameters are correlated. Section 3.3.3 shows how the MISE can be approximated when some of the d parameters are independent from each other.

3.3.1. Estimating the distribution using Kernel Density Estimation

The shape of the pdf $f(\cdot)$ is unknown beforehand. Furthermore, the shape of the estimated pdf might change as more data are acquired. Assuming a functional form of the pdf and fitting the parameters of the pdf to the data may therefore lead to inaccurate fits unless a lot of hand-tuning is applied. We employ a non-parametric approach using KDE [222, 237] because the shape of the pdf is automatically computed and KDE is highly flexible regarding the shape of the pdf.

In KDE, the estimated pdf is given by

$$\hat{f}(x;n) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right). \tag{3.2}$$

Here, $K(\cdot)$ is an appropriate kernel function and h denotes the bandwidth. The choice of the kernel $K(\cdot)$ is not as important as the choice of the bandwidth h [283]. We use a Gaussian kernel because it will simplify some of our calculations. The Gaussian kernel is given by

$$K(u) = \frac{1}{(2\pi)^{d/2}} \exp\left\{-\frac{1}{2}\|u\|_2^2\right\}, \tag{3.3}$$

where $\|u\|_2^2 = u^{\mathsf{T}}u$ denotes the squared 2-norm of u.

The bandwidth h controls the amount of smoothing. For the kernel of (3.3), the same amount of smoothing is applied in every direction, although our method can

easily be extended to a multi-dimensional bandwidth, see, e.g., [52, 255]. There are many different ways of estimating the bandwidth, ranging from simple reference rules like, e.g., Scott's rule of thumb [254] or Silverman's rule of thumb [262] to more elaborate methods; see [24, 55, 151, 283] for reviews of different bandwidth selection methods.

3.3.2. Estimating the Mean Integrated Squared Error for dependent parameters

As an approximation of the MISE of (3.1), the Asymptotic Mean Integrated Squared Error (AMISE) is often used. With the KDE of (3.2) employed, the AMISE is defined as follows [195]:

$$\mathrm{AMISE}_f(n) = \frac{h^4}{4}\sigma_K^4 \int_{\mathbb{R}^d} \left(\nabla^2 f(x)\right)^2 \mathrm{d}x + \frac{\mu_K}{nh^d}. \tag{3.4}$$

Here, σ_K and μ_K are constants that depend on the choice of the kernel $K(\cdot)$:

$$\sigma_K^2 = \int_{\mathbb{R}^d} \|u\|_2^2 K(u)\,\mathrm{d}u, \tag{3.5}$$

$$\mu_K = \int_{\mathbb{R}^d} \left(K(u)\right)^2 \mathrm{d}u. \tag{3.6}$$

Because we use the Gaussian kernel of (3.3), we have $\sigma_K = 1$ and $\mu_K = \left(2\sqrt{\pi}\right)^{-d}$. In (3.4), $\nabla^2 f(x)$ denotes the Laplacian of $f(x)$, i.e.,

$$\nabla^2 f(x) = \sum_{l=1}^{d} \frac{\partial^2 f(x)}{\partial x_l^2}, \tag{3.7}$$

where x_l denotes the l-th entry of x. Note that the Laplacian equals the trace of the Hessian. Assuming that $h \to 0$ and $nh^d \to \infty$ as $n \to \infty$, the AMISE only differs from the MISE by higher-order terms under some mild conditions[1] [262].

The influence of the bandwidth h is demonstrated in an illustrative way by the AMISE of (3.4). The first term of the AMISE of (3.4) corresponds to the asymptotic bias introduced by smoothing the pdf. Therefore, this term approaches zero when $h \to 0$. However, when $h \to 0$, the variance goes to infinity, as can be seen by the second term of the AMISE, which corresponds to the asymptotic variance.

As with the MISE, we cannot evaluate the AMISE because it depends on the true pdf $f(\cdot)$. As suggested by Chen [52] and Calonico *et al.* [44], we can estimate the quantity $\nabla^2 f(x)$ by $\nabla^2 \hat{f}(x;n)$, with $\hat{f}(x;n)$ defined in (3.2). Substituting $f(x)$ in (3.4) with $\hat{f}(x;n)$ gives the measure that we will use to quantify the completeness:

$$J_f(n) = \frac{h^4}{4}\sigma_K^4 \int_{\mathbb{R}^d} \left(\nabla^2 \hat{f}(x;n)\right)^2 \mathrm{d}x + \frac{\mu_K}{nh^d}. \tag{3.8}$$

[1] The pdf $f(\cdot)$ needs to comply with the regularity conditions, $K(u) \geq 0, \forall u$, $\int_{\mathbb{R}^d} K(u)\,\mathrm{d}u = 1$ and σ_K from (3.5) is not infinite.

In summary, the measure (3.8) is an estimation of the MISE of (3.1) given that the pdf is estimated using the KDE of (3.2). Because the MISE cannot be directly evaluated, the asymptotic MISE is used with the estimated pdf substituted for the real pdf.

3.3.3. Estimating the Mean Integrated Squared Error for independent parameters

As explained in Section 3.3.1, KDE is employed because the KDE is highly flexible regarding the shape of the pdf. However, when a large number of parameters are used, i.e., for large values of d, the KDE becomes unreliable due to the curse of dimensionality [254]. One way to overcome this, is to assume that certain parameters are independent. In that case, the joint distribution is not modeled using only one multivariate KDE, but using a combination of KDEs.

Without loss of generality, consider the parameter vector x that can be decomposed into two parts:

$$x = \begin{bmatrix} y \\ z \end{bmatrix}, \tag{3.9}$$

such that $y \in \mathbb{R}^{d_y}$ and $z \in \mathbb{R}^{d_z}$ with $d_y + d_z = d$. If the parameter vectors y and z are independent, the probability density of x equals

$$f(x) = g(y) \cdot l(z), \tag{3.10}$$

where $g(\cdot)$ and $l(\cdot)$ are pdfs. Because y and z have a lower dimensionality than x, the estimated pdfs of $g(\cdot)$ and $l(\cdot)$ will be more reliable. However, we cannot use the measure of (3.8) to quantify the completeness anymore. Therefore, we will show in this section how $J_f(n)$ can be computed in case the real distribution is assumed to take the form (3.10).

The first step is to estimate $g(\cdot)$ and $l(\cdot)$ using $\hat{g}(\cdot;n)$ and $\hat{l}(\cdot;n)$, respectively, where $\hat{g}(\cdot;n)$ and $\hat{l}(\cdot;n)$ are also estimated using KDE, see (3.2). Note that the bandwidths of $\hat{g}(\cdot;n)$ and $\hat{l}(\cdot;n)$ are generally different. Now let the MISE of $g(\cdot)$ and $l(\cdot)$ be defined similarly as the MISE of $f(\cdot)$ in (3.1). It can be shown[2] that if (3.10) holds, then the MISE of $f(x)$ approximately equals

$$\begin{aligned} \mathrm{MISE}_f(n) \approx & \mathrm{MISE}_g(n) \int_{\mathbb{R}^{d_z}} \left(l(z)\right)^2 \mathrm{d}z + \mathrm{MISE}_l(n) \int_{\mathbb{R}^{d_y}} \left(g(y)\right)^2 \mathrm{d}y + \\ & \mathrm{MISE}_g(n) \cdot \mathrm{MISE}_l(n). \end{aligned} \tag{3.11}$$

We can estimate the MISE of $g(\cdot)$ and $l(\cdot)$ in a similar manner as we did for the MISE of $f(\cdot)$ in Section 3.3.2, such that we obtain $J_g(n)$ and $J_l(n)$. Since we cannot evaluate the integrals of (3.11), we estimate them by substituting the estimated pdfs. As a result, we have

$$J_f(n) = J_g(n) \int_{\mathbb{R}^{d_z}} \left(\hat{l}(z;n)\right)^2 \mathrm{d}z + J_l(n) \int_{\mathbb{R}^{d_y}} \left(\hat{l}(y;n)\right)^2 \mathrm{d}y + J_g(n) \cdot J_l(n). \tag{3.12}$$

[2] For the sake of brevity, the proof is omitted. The main idea is based on the variance of the product of two independent variables, see [118], and the assumptions $\mathbb{E}[\hat{g}(y;n)] \approx g(y)$ for all y and $\mathbb{E}[\hat{l}(z;n)] \approx l(z)$ for all z.

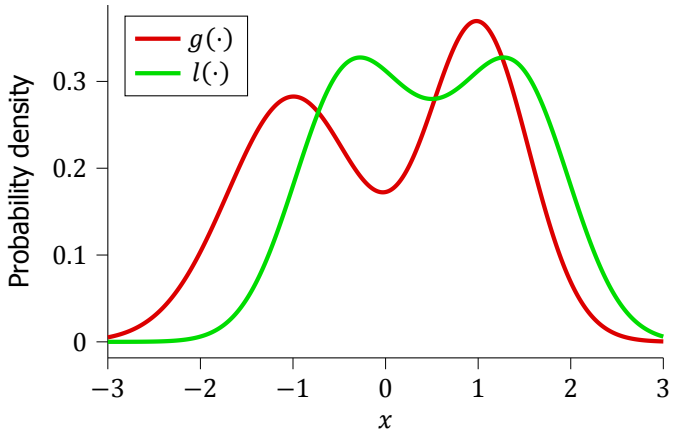


Figure 3.1: The true pdfs $g(\cdot)$ (red line) and $l(\cdot)$ (green line) that are used to illustrate the quantification of the completeness.

In this section, we assumed that the parameters x can be split into two partitions that are independent. It is straightforward to extend the result of (3.12) in case that the parameters x can be split into three of more partitions.

3.4. Examples

In this section, the proposed method of Section 3.3 is illustrated by means of two examples. The first example applies the method with data generated from a known distribution. Because the distribution is known, the real MISE can be accurately approximated and compared with the results from (3.8) and (3.12). Secondly, in Section 3.4.2, the proposed method is applied on a data set containing naturalistic driving data.

3.4.1. Example with known underlying distribution

In this example, the data samples Y_i with $i \in \{1, \dots, n\}$ are independently and identically distributed random variables that are distributed according to the pdf $g(\cdot)$. Each data sample Y_i corresponds to a scalar, i.e., $d_y = 1$. Similarly, the data samples Z_i with $i \in \{1, \dots, n\}$ and $d_z = 1$ are independently and identically distributed random variables that are distributed according to the pdf $l(\cdot)$. The data samples are combined, similar to (3.9), such that the likelihood of $X_i \in \mathbb{R}^d$ with $d = 2$ is $f(X_i) = g(Y_i) \cdot l(Z_i)$.

Figure 3.1 shows the distributions $g(\cdot)$ (red line) and $l(\cdot)$ (green line). Both distributions are Gaussian mixtures, i.e., both pdfs equal the sum of multiple weighted Gaussian distributions. The pdf $g(\cdot)$ corresponds to the average of two Gaussian distributions with means of -1 and 1 and standard deviations 0.5 and 0.3, respectively. The pdf $l(\cdot)$ corresponds to the average of three Gaussian distributions with means -0.5, 0.5, and 1.5, and standard deviations 0.3, 0.5, and 0.3, respectively.

The expectation $\mathbb{E}[\cdot]$ of (3.1) is estimated by repeating the estimation of the pdf

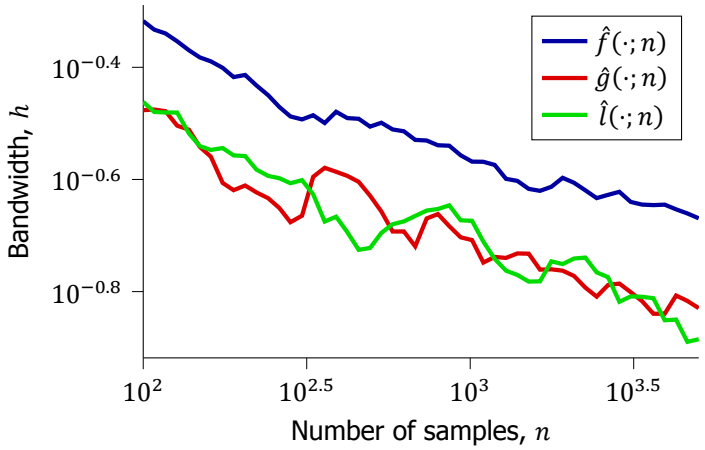


Figure 3.2: The bandwidths of $\hat{f}(\cdot;n)$, $\hat{g}(\cdot;n)$, and $\hat{l}(\cdot;n)$ for the example of Section 3.4.1. The bandwidths are computed using leave-one-out cross validation for different number of samples n.

200 times, such that the real MISE is approximated:

$$\text{MISE}_f(n) \approx \frac{1}{m}\sum_{j=1}^{m}\int_{\mathbb{R}^d}\left(f(x)-\hat{f}_j(x;n)\right)^2 \mathrm{d}x, \tag{3.13}$$

where $\hat{f}_j(x;n)$ is the j-th estimate and $m = 200$.

All three pdfs are estimated using (3.2). We use leave-one-out cross validation to compute the bandwidth h [86] because this minimizes the Kullback-Leibler divergence between the real pdf $f(\cdot)$ and the estimated pdf $\hat{f}(\cdot;n)$ [283, 319]. Note that although the estimation of the pdfs is repeated 200 times to accurately approximate the MISE using (3.13), the bandwidth is only determined once for a specific number of samples. All the other 199 times, the same bandwidths are adopted. The resulting bandwidths are shown in Figure 3.2. The bandwidth of $\hat{f}(\cdot;n)$ is significantly larger than the bandwidths of $\hat{g}(\cdot;n)$ and $\hat{l}(\cdot;n)$. This result is not surprising: because $\hat{f}(\cdot;n)$ represents a bivariate distribution, it requires more data to have a similar bandwidth compared with a univariate distribution [255].

Figure 3.3 shows the results of this example. The blue lines show the real MISEs, approximated using (3.13), where the solid line represents the MISE when $f(\cdot)$ is directly estimated and the dashed line represented the MISE when use is made of (3.10). The MISE is significantly lower when it is correctly assumed that the two parameters are independent. One way to look at this is that the degree of freedom of $f(\cdot)$ is reduced when assuming that the two parameters are independent and this lower degree in freedom leads to a more certain estimate. Hence, the MISE is lower.

The red lines in Figure 3.3 show the measures to quantify the completeness of the data. The solid line shows the result of applying (3.8) and the dashed line shows the result of applying (3.12). Both lines follow the same trend as the blue solid line

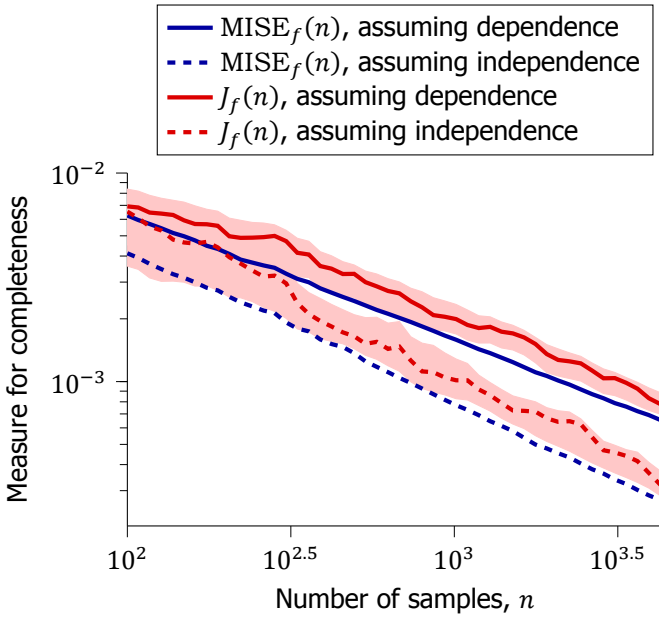


Figure 3.3: The real MISEs (blue lines) of the example of Section 3.4.1, approximated using (3.13), and the measures that are used to quantify the completeness (red lines). The solid lines show the result of estimating a bivariate pdf, so here (3.8) is used to quantify the completeness. The dashed lines show the result of estimating two univariate pdfs and combining them according to (3.10) to create a bivariate pdf, so (3.12) is used to quantify the completeness. The red areas show the interval $[\mu - 3\sigma, \mu + 3\sigma]$, where μ and σ denote the mean and standard deviation, respectively, of the measures of (3.8) and (3.12) when repeating the experiment 200 times.

and the blue dashed line, respectively. This illustrates that the measures (3.8) and (3.12) are applicable for estimating the real MISE of (3.1). To show that this is not a mere coincidence, the red areas in Figure 3.3 show the interval $[\mu - 3\sigma, \mu + 3\sigma]$, where μ and σ denote the mean and standard deviation, respectively, of the measures of (3.8) and (3.12) when repeating the experiment 200 times. Note that the measures of completeness are consistently higher than the real MISE. This can be explained from the fact that the measures of completeness are approximations of the AMISE and the AMISE itself is always higher than the real MISE under some mild conditions, see Theorem 4.2 of [195].

3.4.2. Example with real data

In this example, 60 hours of naturalistic driving data from 20 different drivers (see [70]) are used to extract approximately 2800 braking activities. Three parameters are used to describe each braking activity: the average deceleration, the total speed difference, and the end speed. A histogram of each of these parameters is shown in Figure 3.4. Note that these braking activities do not include full stops, i.e., activities where the end speed is zero, because the distribution of the end speed will have a large peak at zero. The AMISE of (3.4) deviates more from the real

MISE of (3.1), especially for larger bandwidths, when such peaks are present in the underlying distribution [195]. Because the measure (3.8) we use for quantification of completeness is based on the AMISE of (3.4), we want to avoid these peaks as much as possible. Therefore, the full stops are excluded. Note, however, that the method can be applied separately for the full stops. In fact, the analysis for full stops will be simpler because a full stop activity can be parameterized using only two parameters because the end speed always equals zero.

The three parameters are correlated so this advocates the use of a multivariate KDE. However, as we have seen in the first example, the higher the dimension, the higher the measure for completeness will generally be. So there is a trade-off: Assuming that certain parameters are independent results in an error of the estimated pdf but the resulting MISE, and hence the measure of completeness, will be lower. To illustrate this, we estimate the pdf while assuming all parameters to be dependent and we estimate the pdf while assuming that the average deceleration is independent from the other two parameters. Note that the correlation between the average deceleration and the other parameters is fairly low, so this justifies this choice. The speed difference and end speed are highly correlated, so we will not assume that these two parameters are independent. Before estimating the pdfs, the parameters are translated and rescaled such that each parameter has a sample mean of zero and a sample variance of one. In this example, $\hat{f}(\cdot;n)$ denotes the estimated 3-dimensional pdf using all three parameters, $\hat{g}(\cdot;n)$ denotes the estimated univariate pdf of the average deceleration, and $\hat{l}(\cdot;n)$ denotes the estimated bivariate pdf of the speed difference and the end speed.

Figure 3.5 shows the bandwidths of the three estimated pdfs for different number of samples, starting from $n = 600$ samples to approximately $n = 2800$ samples. As opposed to the bandwidths of our previous example, see Figure 3.2, the bandwidth of $\hat{f}(\cdot;n)$ is not larger than the bandwidth of $\hat{g}(\cdot;n)$ for low values of n. This is caused by some outliers of the average deceleration because these outliers have a large influence on the bandwidth of $\hat{g}(\cdot;n)$ [127]. These outliers also influence the bandwidth of $\hat{f}(\cdot;n)$, but this influence is less because the bandwidth of $\hat{f}(\cdot;n)$ is also influenced by the other parameters.

The measures of completeness of the data of the braking activities are shown in Figure 3.6. The solid blue line results from the estimated 3-dimensional pdf, i.e., $\hat{f}(\cdot;n)$, where (3.8) is used to quantify the completeness. The solid red line results from the estimated univariate and bivariate pdfs $\hat{g}(\cdot;n)$ and $\hat{l}(\cdot;n)$, where (3.12) is used to quantify the completeness. The measure for the completeness is much lower for the latter case, indicating that the uncertainty of the pdf is much lower when it is assumed that the average deceleration is independent from the other two parameters.

Whether it is better to assume that all parameters are dependent or not depends on the threshold that defines the desired measure and the amount of data. If the threshold is not met, the result can be used to guess how much more data is required by extrapolating the result. To illustrate this, the straight dashed lines in Figure 3.6 represent the least squares logarithmic fits of the corresponding solid lines that can be used for extrapolation. These straight blue and red lines are

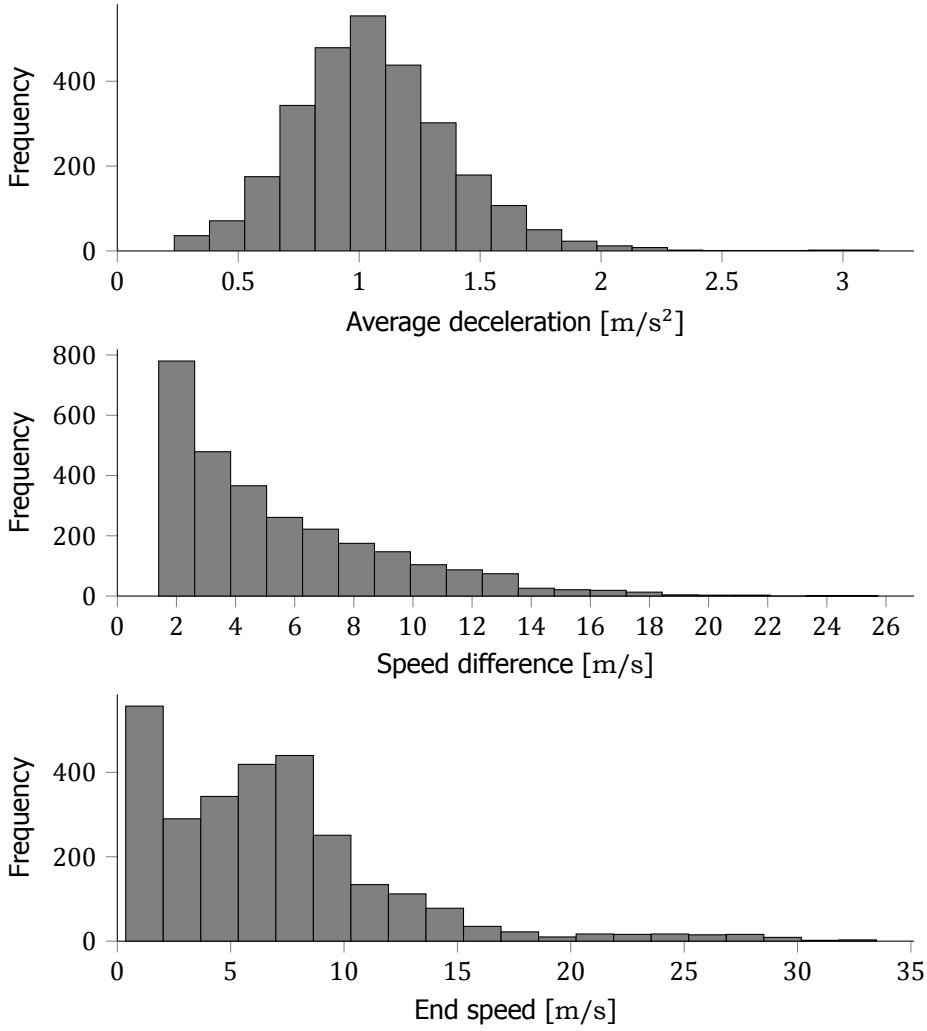


Figure 3.4: Histograms of the data that are used for the example of Section 3.4.2.

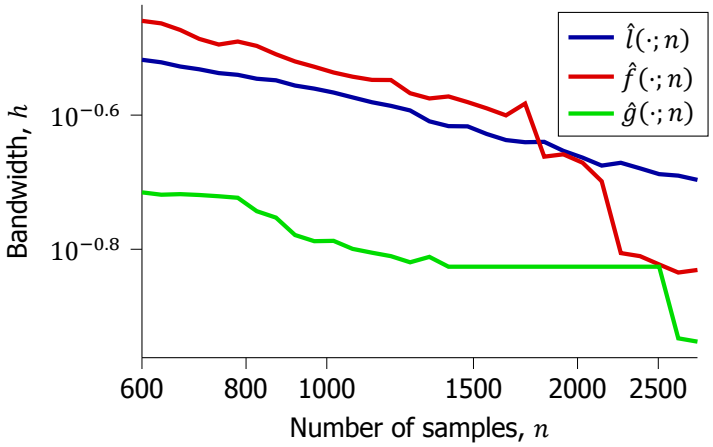


Figure 3.5: The bandwidths of $\hat{f}(\cdot;n)$, $\hat{g}(\cdot;n)$, and $\hat{l}(\cdot;n)$ for the example of Section 3.4.2. The bandwidths are computed using leave-one-out cross validation for different number of samples n.

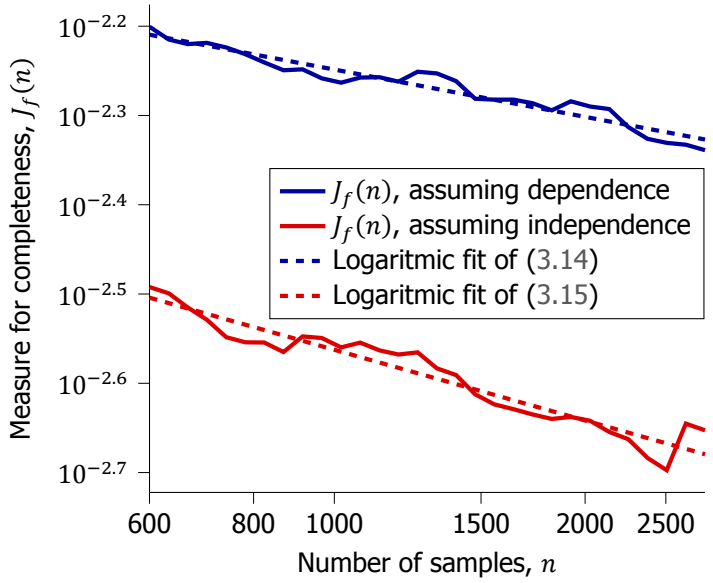


Figure 3.6: The measures of completeness for the example of Section 3.4.2 with the assumption that all three parameters depend on each other (blue solid line) and with the assumption that the first parameter, i.e., the average deceleration, does not depend on the other two parameters (red solid line). The corresponding dashed lines represent the least squares logarithmic fits given by (3.14) and (3.15).

described by the formulas

$$0.019 \cdot n^{-0.18}, \tag{3.14}$$
$$0.017 \cdot n^{-0.26}, \tag{3.15}$$

respectively. As an example, let us assume that the threshold equals 0.003. In that case, $n \approx 800$ would suffice if we assume that the average deceleration is independent from the speed difference and end speed, see the dashed red line in Figure 3.6 and (3.15). This threshold, however, is not yet reached when assuming that all parameters are dependent, see the blue lines in Figure 3.6. Extrapolating the result using (3.14) provides a rough estimate of the required number of samples: $n \approx 28000$, i.e., ten times as many samples as we have used in this example.

3.5. Discussion

The measure for quantification of completeness of the set of activities that is presented in this chapter is based on the amount of data and the chosen parameterization. More data might be used to achieve a certain threshold. However, it might also be possible to adapt the parameterization to achieve a certain threshold if a parameterization exists that achieves a certain threshold. Hence, the presented method can be used to determine an appropriate parameterization of activities.

The method for quantifying the completeness of a set of activities depends on a threshold that needs to be chosen. Only in case of an infinite set of data, the measure for completeness approaches zero, so this threshold needs to be larger than zero. This threshold might be different for different applications. For example, when the data are used for determining test scenarios [94, 228], the desired threshold might be lower than when the data are used for determining driver models [242, 298]. Furthermore, the threshold depends on the number of parameters for one activity, denoted by d in Section 3.3. Based on experience with the data set used in Section 3.4.2, assuming that the data set is normalized such that the standard deviation equals one, a threshold between 0.01 and 0.001 gives good results. When a threshold of 0.01 is reached, a reasonable reliable pdf can be constructed to analyze nominal driving behavior, whereas a threshold of around 0.001 is required to also accurately analyze the edge cases.

When using our measure for completeness, the following might be considered. As explained in Section 3.3, the measure for completeness is based on the AMISE. It is also mentioned that the AMISE only differs from the MISE by higher-order terms under some mild conditions. This requires that the real pdf is smooth, i.e., without large spikes [195]. Marron and Wand [195] also state that the AMISE is strictly higher than the MISE under some mild conditions[3]. As a result, it is likely that the measure for completeness, which is an approximation of the AMISE, is higher than the MISE. This could lead to an overestimation of the number of required samples.

The measure for completeness that is proposed in this chapter can be regarded as an approximation of the MISE of (3.1). To minimize the MISE, the approximated

[3]The Laplacian of $f(\cdot)$ needs to be continuous and square-integrable and $K(u) \geq 0, \forall u$.

pdf should be similar to the real pdf. It might be, however, that one is not interested in the exact likelihoods of certain values of the parameters, but in all possible values that the parameters can have. In this case, one might be interested in the support of the real pdf because the support of the pdf defines all possible values for which the likelihood is larger than zero, see, e.g., [250].

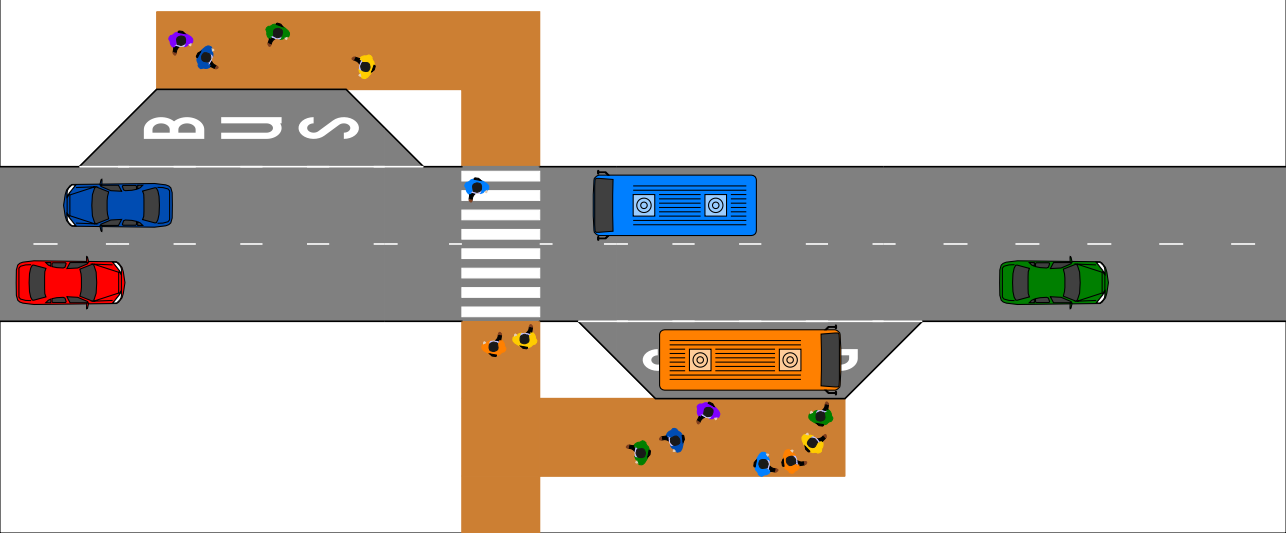
As mentioned in Section 3.2, our problem of quantifying the completeness of a data set can be divided into three subproblems. The first subproblem, i.e., how to quantify the completeness regarding the scenario categories, can be regarded as the so-called unseen species problem [42, 110] or species estimation problem [316]. In case of the unseen species problem, the entire population is partitioned into C categories and the objective is to estimate C given only a part of the entire population. To continue the analogy, the second subproblem, i.e., how to quantify the completeness regarding all scenarios that fall into a specific scenario category, relates to quantifying whether we have a complete view on the variety among one species, given the number of individuals that we have seen. The third subproblem addresses a part of the scenarios, i.e., the activities. In line with the previous analogy, this can be seen as quantifying whether we have a complete view of the parts of the species, e.g., its limbs or organs.

Our proposed method answers the third subproblem, i.e., how to quantify the completeness regarding the activities. The advantage of using the activities for determining the completeness is that there is only a limited number of types of activities. As a result, for each type of activity, it is expected that there is no need for an extremely large data set to obtain a fair number of similar activities. On the other hand, however, it is not known how much data are required to obtain the desired threshold because, e.g., this depends on the parameterization that is chosen. The next step is to quantify the completeness regarding all scenarios that fall into a specific scenario category. Here, the joint probability of the parameters of different activities in the same scenario category might be considered. Although the presented method can be applied, this might be impractical because the number of parameters will be higher than for the activities. The problems of quantifying the completeness regarding all scenarios that fall into a specific scenario category and quantifying the completeness regarding the scenario categories remain future work.

The more parameters are considered, the lower the rate at which the measure for completeness decreases with more data. This is a direct result from the curse of dimensionality [254]. Therefore, if more parameters are considered, the need for more data increases exponentially if no further assumptions are done regarding the dependence of the parameters. The presented measure for completeness shows that if all aspects of a scenario, e.g., all vehicles, weather conditions and road properties, are dependent, then one needs a practically infeasible amount of data to estimate the statistics reliably. In other words, to estimate the statistics of all such aspects of a scenario, it will be important to carefully examine which assumptions can be made regarding the dependence of the aspects of a scenario.

3.6. Conclusions

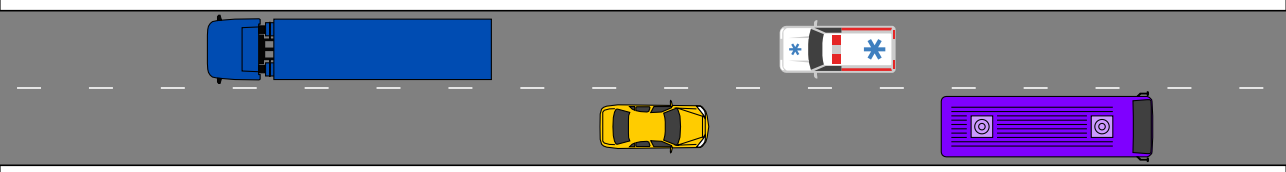
More and more field data from (naturalistic) driving data become available. The data are used for all kinds of driving-related research, developments, assessments, and evaluations. When deducing claims based on the collected data, we require knowledge about the degree of completeness of the data. We considered the data as a sequence of scenarios, whereas activities are the building blocks of these scenarios. To obtain knowledge about the degree of completeness of the data, we propose a measure to quantify the completeness of the activities. This measure allows to partly answer questions like "have we collected enough field data?" We illustrated the method using an artificial data set, for which the underlying distributions are known. These results show that the proposed method correctly quantifies the completeness of the activities. We also applied the method on a data set with naturalistic driving to show that the method can be used to estimate the required number of samples. In future work, we will extend the method to whole traffic scenarios and scenario categories and we will investigate the appropriate thresholds for the measure to quantify completeness in different applications. Furthermore, the proposed method will be used to evaluate the level of completeness of the data collection aimed at defining relevant test cases for the assessment of automated vehicles.


BUS

4

Real-world scenario mining

This chapter is based on:
E. de Gelder, J. Manders, C. Grappiolo, J.-P. Paardekooper, O. Op den Camp, and B. De Schutter, *Real-world scenario mining for the assessment of automated vehicles,* in *IEEE International Transportation Systems Conference (ITSC)* (2020) pp. 1073–1080.



Scenario-based methods for the assessment of Automated Vehicles (AVs) are widely supported by many players in the automotive field. Scenarios captured from real-world data can be used to define the scenarios for the assessment and to estimate their relevance. Therefore, different techniques are proposed for capturing scenarios from real-world data. In this chapter, we propose a new method to capture scenarios from real-world data using a two-step approach. The first step consists in automatically labeling the data with tags. Second, we mine the scenarios, represented by a combination of tags, based on the labeled tags. One of the benefits of our approach is that the tags can be used to identify characteristics of a scenario that are shared among different types of scenarios. In this way, these characteristics need to be identified only once. Furthermore, the method is not specific for one type of scenario and, therefore, it can be applied to a large variety of scenarios. We provide two examples to illustrate the method. This chapter is concluded with some promising future possibilities for our approach, such as automatic generation of scenarios for the assessment of AVs.

4.1. Introduction

The development of Automated Vehicles (AVs) has made significant progress in the last years and it is expected that AVs will soon be introduced on our roads [29, 190] and become an integral part of intelligent transportation systems [49, 95]. An essential aspect in the development of AVs is the assessment of quality and performance aspects of the AVs, such as safety, comfort, and efficiency [27, 270]. Among other methods, a scenario-based approach has been proposed [94, 229]. For scenario-based assessment, proper specification of scenarios is crucial since they are directly reflected in the test cases used for the assessment [270]. One approach for specifying these test cases is to base them on captured scenarios from real-world data collected on the level of individual vehicles [70, 94, 229, 234].

Different techniques for capturing scenarios and driving maneuvers have been proposed in the literature. Kasper *et al.* [157] use object-oriented Bayesian networks for the recognition of 27 type of driving maneuvers. Krajewski *et al.* [171] detect lane changes using lane crossings and Schlechtriemen *et al.* [248] detect lane changes using a naive Bayes classifier and a hidden Markov model. Xie *et al.* [313] use random forest classifiers for detecting accelerating, braking, and turning with features extracted using principal component analysis, stacked sparse auto-encoders, and statistical features. Cara and de Gelder [45] extract safety-critical car-cyclist scenarios from data collected by a vehicle using several machine-learning techniques, among which support vector machines and multiple instance learning.

In this chapter, we propose a new method for mining scenarios from real-world driving data using automated tagging and searching for combination of tags. Our method consists of two steps. First, the data are automatically tagged with relevant information. For example, a tag "lane change" is added to a vehicle at the time this vehicle is performing a lane change. Second, the scenarios are mined based on the aforementioned tags. To do this, we represent a scenario using a combination of tags and we search for this combination of tags in the tagged data from the previous step.

The proposed method brings several benefits. First, by tagging the data, characteristics that are shared among different type of scenarios need to be identified only once, whereas these characteristics would be identified multiple times if each type of scenarios would be identified completely independently. For example, a characteristic could be the presence of a leading vehicle, so if we independently identify two different types of scenarios that consider a leading vehicle, we would identify the leading vehicle two times. Second, by splitting the process in two parts, i.e., the tagging and the scenario mining, the scenario mining can be applied to different types of data, e.g., data from a vehicle [221] or a measurement unit above the road [170, 171]. It is also possible to have manually tagged data, e.g., see [103]. Third, our approach is easily scalable because additional types of scenarios can be mined by describing them as a combination of (sequential) tags. Fourth, the approach reveals promising future possibilities, such as the generation of scenarios based on the mined scenarios. The generated scenarios can be used to define the test cases for the assessment of intelligent vehicles [70, 94, 229, 234, 270, 323].

In Section 4.2, we formulate the problem of scenario mining. Sections 4.3

and 4.4 describe the two steps of our proposed method, i.e., the tagging of the data and the scenario mining based on these tags, respectively. We illustrate the proposed scenario mining approach with few examples in Section 4.5. In Section 4.6, we discuss the approach, results, and some possible future improvements. We end this chapter with conclusions and discuss next steps in Section 4.7.

4.2. Problem formulation

To formulate the scenario mining problem, we distinguish quantitative scenarios from qualitative scenarios, using the definitions of *scenario* and *scenario category* of Chapter 2 [79]:

Definition 4.1 (Scenario)**.** *A scenario is a quantitative description of the relevant characteristics and activities and/or goals of the ego vehicle(s), the static environment, the dynamic environment, and all events that are relevant to the ego vehicle(s) within the time interval between the first and the last relevant event.*

Definition 4.2 (Scenario category [79])**.** *A scenario category is a qualitative description of the relevant characteristics and activities and/or goals of the ego vehicle(s), the static environment, and the dynamic environment.*

A scenario category is an abstraction of a scenario and, therefore, a scenario category comprises multiple scenarios [79]. For example, the scenario category "cut-in" comprises all possible cut-in scenarios. Given such a scenario category, our goal is to find all corresponding scenarios in a given data set. Hence, we define the scenario mining problem as follows:

Problem (Scenario mining)**.** *Given a scenario category, how to find all scenarios that correspond to this scenario category in a given data set?*

4.3. Data tagging

Our method of scenario mining is divided into two steps. The first step consists in describing the data using *tags*, where a tag is a "label attached to a scenario for the purpose of categorization" [147]. The second step involves extracting the scenarios based on these tags. In this section, we explain how the *tags* are determined. The scenario mining based on these tags is explained in the next section.

As described in Definition 4.1, events and activities are constituents of a scenario. Part of the tags that we consider describe activities of the vehicles. Therefore, we will use the definition of the term *activity* of Chapter 2 [79]:

Definition 4.3 (Activity)**.** *An activity is a quantitative description of the time evolution of one or more state variables of an actor between two events.*

Because an activity starts and ends with an event, for describing the activities, we will first describe how we detect the events.

As an illustration, Table 4.1 lists the tags that are considered for the case study in this chapter. We distinguish between tags that describe the behavior of the ego

Table 4.1: Tags that are considered in this chapter.

Subject	Description	Section	Possible tags
Ego vehicle	Longitudinal activity	Section 4.3.1	Accelerating, decelerating, cruising
Ego vehicle	Lateral activity	Section 4.3.2	Changing lane left, changing lane right, following lane
Any other vehicle	Longitudinal activity	Section 4.3.3	Accelerating, decelerating, cruising
Any other vehicle	Lateral activity	Section 4.3.4	Changing lane left, changing lane right, following lane
Any other vehicle	Longitudinal state	Section 4.3.5	In front of ego, behind ego
Any other vehicle	Lateral state	Section 4.3.6	Left of ego, right of ego, same lane as ego, unclear
Any other vehicle	Leading vehicle	Section 4.3.7	Leader, no leader
Static environment	On highway	Section 4.3.8	Highway, no highway

vehicle, tags that describe the behavior and the state of any other vehicle, and tags that describe the static environment.

Remark 4.1. When other scenario categories are considered than the ones in our case study, other tags might be necessary. The approach for mining the scenarios using the tags, however, is general and also applies when other tags are used. For example, for this chapter we do not consider other road users than vehicles, but the proposed method also works if cyclists or pedestrians are considered. ◊

In the remainder of this section, we explain how the tags of Table 4.1 are assigned. Here, we assume that the data are sampled with a sample time of t_s.

4.3.1. Longitudinal activity of the ego vehicle

We distinguish between three different longitudinal activities: "accelerating", "decelerating", and "cruising". The ego vehicle is performing either one of these activities. An acceleration activity starts at an acceleration event, so we will first describe how we detect an acceleration event.

To extract the longitudinal events, we might simply examine whether the acceleration of the vehicle is above or below a certain threshold. This approach, however, would be prone to sensor noise. That is why we use the speed difference within a certain sample window of length $k_\mathrm{h} > 0$, where k_h is an integer. Let $v(k)$ denote the speed of the ego vehicle at sample step k. Next, let us define the

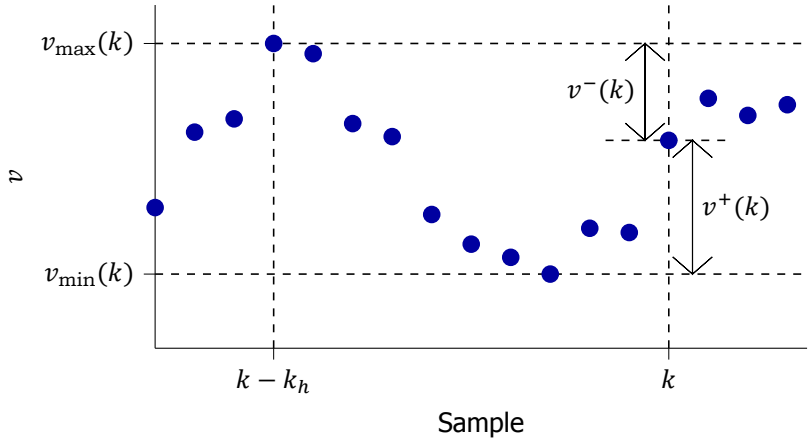


Figure 4.1: An example of a speed profile and how the variables $v_{\min}(k)$, $v_{\max}(k)$, $v^+(k)$, and $v^-(k)$ are calculated at a certain sample step k with $k_{\mathrm{h}} = 10$.

minimum and maximum speed between the current sample step k and $k - k_{\mathrm{h}}$:

$$v_{\min}(k) \equiv \min_{\tau \in \{k-k_{\mathrm{h}},\ldots,k\}} v(\tau), \tag{4.1}$$

$$v_{\max}(k) \equiv \max_{\tau \in \{k-k_{\mathrm{h}},\ldots,k\}} v(\tau). \tag{4.2}$$

For detecting acceleration and decelerating events, the difference between the current speed and $v_{\min}(k)$ and $v_{\max}(k)$ are used:

$$v^+(k) \equiv v(k) - v_{\min}(k), \tag{4.3}$$

$$v^-(k) \equiv v(k) - v_{\max}(k). \tag{4.4}$$

Figure 4.1 illustrates how $v_{\min}(k)$, $v_{\max}(k)$, $v^+(k)$, and $v^-(k)$ are calculated.

First, we assume that the event at the start of the data set is a cruising event at $k = k_0$. Next, we go chronologically through the data set. An acceleration event is happening at sample k if all of the following conditions are true:

- The vehicle is not performing an acceleration activity, i.e., the last event is not an acceleration event.
- There has been a substantial speed increase between sample step $k - k_{\mathrm{h}}$ and k, i.e.,

  $$v^+(k) \geq a_{\mathrm{cruise}} k_{\mathrm{h}} t_{\mathrm{s}}, \tag{4.5}$$

  where $a_{\mathrm{cruise}} > 0$ is a parameter that describes the maximum average acceleration within the time window $k_{\mathrm{h}} t_{\mathrm{s}}$ for a cruising activity.
- There is no lower speed in the near future, i.e.,

  $$v_{\min}(k + k_{\mathrm{h}}) = v(k). \tag{4.6}$$

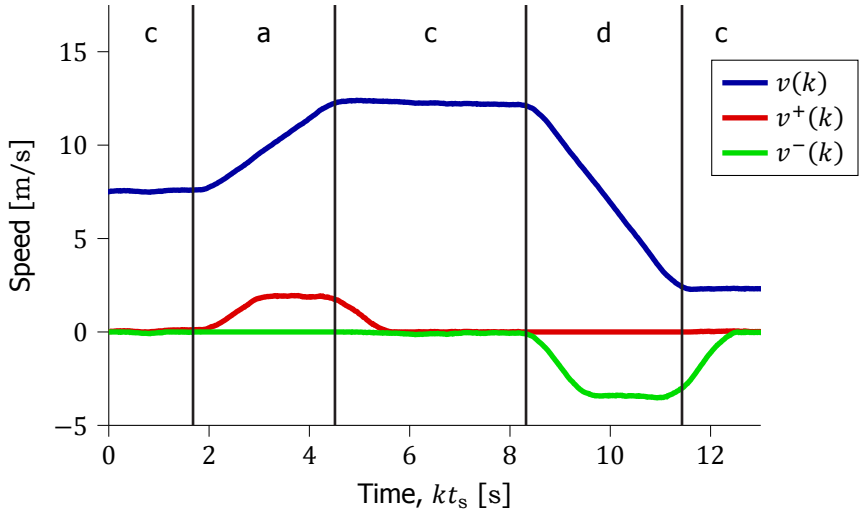


Figure 4.2: Hypothetical speed profile and the corresponding activities cruising (c), accelerating (a), and decelerating (d). The vertical lines represent the events times.

- There is a substantial speed difference during the activity, i.e.,

$$|v(k_{\text{end}}(k)) - v(k)| > \Delta_v, \tag{4.7}$$

where $\Delta_v > 0$ is the minimum speed increase and $k_{\text{end}}(k)$, i.e., the last sample of the acceleration activity, is controlled by the parameter a_{cruise} and equals the first sample at which the speed increase is below a threshold:

$$k_{\text{end}}(k) \equiv \min_{\tau > k} \{\tau : v^+(\tau + k_h) < a_{\text{cruise}} k_h t_s\}. \tag{4.8}$$

A deceleration event is detected in a similar manner as an acceleration event. Now that we know the start and the end of the acceleration and deceleration activities, we simply label the remaining samples as "cruising".

Figure 4.2 illustrates the longitudinal activities given a hypothetical speed profile. The algorithm above described produces the activities "cruising", "accelerating", "cruising", "decelerating", and "cruising".

A result of the activity detection could be very short cruising activities, especially when the acceleration is around a_{cruise} or $-a_{\text{cruise}}$. Therefore, all cruising activities shorter than k_{cruise} sample steps are removed as well as the two events that define the start and the end of the cruising activity. Here, we consider three possibilities:

1. Before and after the cruising activity, the vehicle performs the same activity. In that case, these activities are merged.

2. The vehicle decelerates before the cruising activity and accelerates afterwards. In that case, an acceleration event is defined at the lowest speed of the vehicle within the original cruising activity.

3. The vehicle accelerates before the cruising activity and decelerates afterwards. In that case, a deceleration event is defined at the highest speed of the vehicle within the original cruising activity.

4.3.2. Lateral activity of the ego vehicle

We distinguish between three different lateral activities: "following lane", "changing lane left", and "changing lane right". To detect the lane changes, the lateral distances toward the left and right lane lines are used. These distances are estimated from camera images. The estimation is outside the scope of this chapter. We refer the interested reader to [92]. Let $l(k)$ and $r(k)$ denote the distance toward the left and right lane line, respectively. We use the ISO coordinate system[1] [145], so $l(k) \geq 0$ and $r(k) \leq 0$. At the moment the vehicle crosses the line, the distances to the lines will change substantially. For example, during a lane change to the left, the left lane line becomes the right lane line. Hence, we detect a left lane change if the change in the lane line distances is more than the threshold $\Delta_l > 0$:

$$l(k) - l(k-1) > \Delta_l, \tag{4.9}$$
$$r(k) - r(k-1) > \Delta_l. \tag{4.10}$$

Similarly, a right lane change is detected when the following conditions are satisfied:

$$l(k) - l(k-1) < -\Delta_l, \tag{4.11}$$
$$r(k) - r(k-1) < -\Delta_l. \tag{4.12}$$

Once a lane change is detected using (4.9) to (4.12), the moment at which the lane change starts is determined. To do this, we make use of $l^+(k)$ and $l^-(k)$, which are similarly defined as $v^+(k)$ and $v^-(k)$, i.e.,

$$l^+(k) \equiv l(k) - \min_{\tau \in \{k-k_h,\ldots,k\}} l(\tau), \tag{4.13}$$
$$l^-(k) \equiv l(k) - \max_{\tau \in \{k-k_h,\ldots,k\}} l(\tau). \tag{4.14}$$

Similarly, $r^+(k)$ and $r^-(k)$ are defined. If a lane change is detected at sample step k using (4.9) and (4.10) or (4.11) and (4.12), the start of this lane change is estimated. The start of the lane change is at the last sample step before sample step k at which there was not a change in either of the line distances larger than a threshold controlled by the parameter v_{lat}. For example, for a right lane change detected at sample step k, the start is at:

$$\max_{\tau<k}\{\tau : l^+(\tau) < v_{\text{lat}} k_h t_s \vee r^+(\tau) < v_{\text{lat}} k_h t_s\}, \tag{4.15}$$

where $\vee$ indicates that either one of the two conditions needs to be satisfied.

[1] In the ISO coordinate system, the x-axis points to the front of the vehicle and the y-axis points to the left of the vehicle. The origin of the coordinate system is often at the ground below the midpoint of the rear axle.

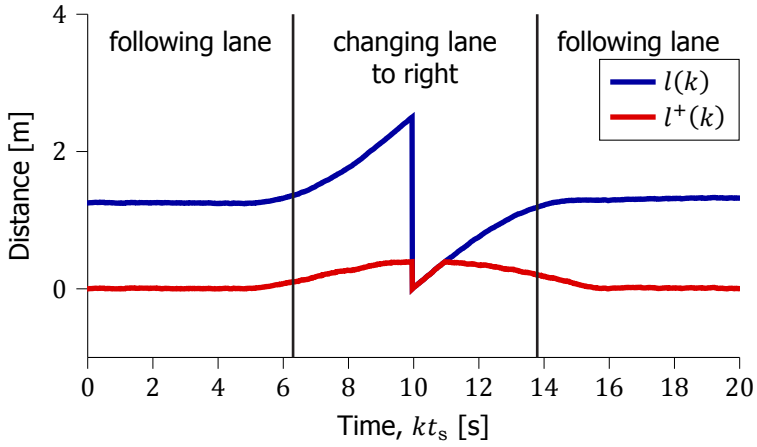


Figure 4.3: The hypothetical distance toward the left lane line $l(k)$ during a right lane change of the ego vehicle. The vertical lines indicate the time instants of the events at the start and the end of the lane change.

The end of a lane change is at the sample step at which either of the lane line distances increase or decrease is below a threshold. For a right lane change, this is at

$$\min_{\tau>k}\{\tau\colon l^+(\tau+k_\mathrm{h}) < v_\mathrm{lat}k_\mathrm{h}t_\mathrm{s} \vee r^+(\tau+k_\mathrm{h}) < v_\mathrm{lat}k_\mathrm{h}t_\mathrm{s}\} \tag{4.16}$$

For a left lane change, the start and end is defined by substituting $-l^-(\cdot)$ and $-r^-(\cdot)$ for $l^+(\cdot)$ and $r^+(\cdot)$, respectively, in (4.15) and (4.16).

Figure 4.3 illustrates a hypothetical lane change of the ego vehicle. It shows that the distance towards the left lane line changes when the ego vehicle crosses the line, such that the conditions (4.11) and (4.12) are satisfied. In Figure 4.3, events at the start and the end of the lane change are denoted by the black vertical lines.

Remark 4.2. It might happen that there is no accurate measurement of the lane line distances available at a certain sample step k. For example, in Figure 4.4, there is no line information while the ego vehicle performs a lane change. By using the next available line distances instead of $l(k)$ and $r(k)$ and the previous available line distances instead of $l(k-1)$ and $r(k-1)$ in (4.9) to (4.12), our algorithm is still able to detect lane changes. ◊

4.3.3. Longitudinal activity of other vehicle

The longitudinal activities of other vehicles are estimated in a similar manner as for the ego vehicle. However, instead of the speed of the ego vehicle, $v(k)$, the speed of the other vehicles is used. The ego vehicle measures the relative speed of other vehicles. Let $v_i^\mathrm{rel}(k)$ denote the relative speed of the i-th vehicles at sample k. The absolute speed of other vehicles is estimated by adding $v(k)$ to the estimated

Figure 4.4: The ego vehicle passes a flyover during daytime while performing a lane change. This causes glare such that the distance to the lane lines are temporarily unavailable.

relative speed:

$$v_i^{\text{abs}}(k) = v_i^{\text{rel}}(k) + v(k). \tag{4.17}$$

To compute the longitudinal activities of the i-th vehicle, the approach outlined in Section 4.3.1 is used with $v_i^{\text{abs}}(k)$ substituted for $v(k)$.

Remark 4.3. Typically, $v_i^{\text{rel}}(k)$ is obtained by fusing the outputs of several sensors [92]. If $v_i^{\text{rel}}(k)$ is not available, e.g., because the vehicle moved out of the view of the ego vehicle's sensors, there are no activities estimated for the i-th vehicle at sample step k. Consequently, no tags are applied for the i-th vehicle at sample step k. This applies for all tags of the other vehicles that are mentioned in Table 4.1. ◊

4.3.4. Lateral activity of other vehicle

For the lane changes of other vehicles, only the lane changes to and from the ego vehicle's lane are considered. To detect a lane change of the i-th vehicle, we use the distance of the i-th vehicle toward the ego vehicle's left and right lane lines, denoted by $l_i(k)$ and $r_i(k)$, respectively. To determine $l_i(k)$ and $r_i(k)$, we subtract the estimated lane line positions from the estimated lateral position of the i-th vehicle. The lane line positions are based on the estimated shape of the lane lines. For more details, we refer the reader to [92]. We define $l_i^+(k)$, $r_i^+(k)$, $l_i^-(k)$, and $r_i^-(k)$ similarly as $l^+(k)$ and $l^-(k)$ in (4.13) and (4.14).

A lane change is detected if the vehicle crosses either of the two lane lines. There are four possible ways this can happen. For now, we consider a right lane change toward the ego vehicle's lane. A right lane change of the i-th vehicle toward the ego vehicle's lane is detected at sample step k if the vehicle is not already changing lane and

$$l_i(k-1) \leq 0 \wedge l_i(k) > 0, \tag{4.18}$$

where $\wedge$ indicates that both of the two conditions need to be satisfied.

To determine the start of the lane change, the lateral speed should be below the threshold v_{lat} or — in case the vehicle changes several lanes — the lateral

Table 4.2: Lateral state based on $l_i(k)$ and $r_i(k)$.

	$l_i(k) < 0$	$l_i(k) \geq 0$
$r_i(k) < 0$	Left of ego	Same lane as ego
$r_i(k) \geq 0$	Unclear	Right of ego

movement should be above a certain threshold (controlled by α_1). Because it might happen that the lateral speed is below the threshold during the whole lane change, a minimum lateral movement is considered as well (controlled by α_2). As a result, the start of a right lane change toward the ego vehicle's lane is estimated to occur at sample step

$$\max_{\tau<k}\{\tau : l_i(\tau) < -\alpha_1 w_i(k) \vee \left(l_i^+(\tau) < v_{\text{lat}} k_{\text{h}} t_{\text{s}} \wedge l_i(\tau) < -\alpha_2 w_i(k)\right)\}. \tag{4.19}$$

Here, $w_i(k) = l_i(k) - r_i(k)$ is the estimated lane width. The end of the same lane change is estimated, in a similar way, to occur at sample step:

$$\min_{\tau>k}\{\tau : l_i(\tau) > \alpha_1 w_i(k) \vee \left(l_i^+(\tau + k_{\text{h}}) < v_{\text{lat}} k_{\text{h}} t_{\text{s}} \wedge l_i(\tau) > \alpha_2 w_i(k)\right)\}. \tag{4.20}$$

A right lane change from the ego vehicle's lane and a left lane change from or to the ego vehicle's lane are determined in a similar manner.

4.3.5. Longitudinal state of other vehicle

For the longitudinal state of any other vehicle, two possibilities are considered: in front of the ego vehicle or behind the ego vehicle. Let the longitudinal position at sample step k of the i-th vehicle relative to the ego vehicle be denoted by $x_i(k)$. The tag "in front of ego" applies when $x_i(k) > 0$ and the tag "behind ego" applies when $x_i(k) \leq 0$.

4.3.6. Lateral state of other vehicle

Four different possibilities are considered for the lateral state of any other vehicle. The lateral state is based on the estimated distance of the other vehicle toward the ego vehicle's lane lines, see Table 4.2. The situation of $l_i(k) < 0$ and $r_i(k) \geq 0$ would mean that the other vehicle is left of the left lane line and right of the right lane line, so it is unclear in which lane the vehicle is.

4.3.7. Leading vehicle

Two possibilities are considered: a vehicle is a leading vehicle (the tag "leader" applies) or not (the tag "no leader" applies). A vehicle i is considered as a leading vehicle at sample step k if all of the following conditions are satisfied:

- The vehicle is in front of the ego vehicle, i.e., $x_i(k) > 0$.
- The vehicle drives in the same lane as the ego vehicle, i.e., $l_i(k) \geq 0$ and $r_i(k) < 0$.

- The time headway of the ego vehicle toward the other vehicle, i.e., $x_i(k)/v(k)$ is less than the parameter $\tau_{\mathrm{h}} > 0$.
- There is no other vehicle that is closer to the ego vehicle while satisfying the above conditions, i.e., $x_i(k) \leq x_j(k)$ for all j-th vehicles that satisfy the above conditions.

4.3.8. Static environment

The aspect of the static environment that is considered in this chapter is whether the ego vehicle drives on the highway or not. The location of the ego vehicle, based on GPS measurements, is used to determine the road the ego vehicle is driving on based on OpenStreetMaps[2]. If the road is classified as "motorway" (see [218] for all possibilities), the tag "highway" is applied. Otherwise, the tag "no highway" is used.

4.4. Mining scenarios using tags

For the scenario mining, we formulate a scenario category using a combination of tags. As an example, Figure 4.5 shows how the scenario category "cut-in" can be formulated using tags. To further structure the tags, we formulate a scenario category as a sequence of *items* where each *item* corresponds to a combination of tags for all relevant subjects. The number of items may vary from scenario category to scenario category. The scenario category "cut-in" in Figure 4.5 contains two items and considers a vehicle other than the ego vehicle that changes lane (other vehicle, item 1 and 2) and becomes the leading vehicle (other vehicle, item 2). In the meantime, the ego vehicle follows its lane (ego vehicle, items 1 and 2) and the scenario category only considers highway driving (static environment, items 1 and 2). When describing the tags for each item, logical AND, OR, or NOT rules may be used. For example, for the other vehicle in Figure 4.5, either the tag "changing lane left" *or* the tag "changing lane right" needs to apply.

The scenarios are mined by searching for matches of the defined items within the tags of the data set. This searching is subject to two rules:

1. For each item, there needs to be a match for all relevant subjects *at the same sample time*.
2. The different items need to occur *right after each other*.

To continue the example of the scenario category "cut-in", Figure 4.6 shows a part of labeled data in which a cut-in scenario is found. The two vertical dashed lines indicate the start and the end of the cut-in that is defined in Figure 4.5.

4.5. Case study

Here, we illustrate the proposed method by applying it to the data set described in [221]. The data have been recorded from a single vehicle in which different

[2]See https://www.openstreetmap.org/.

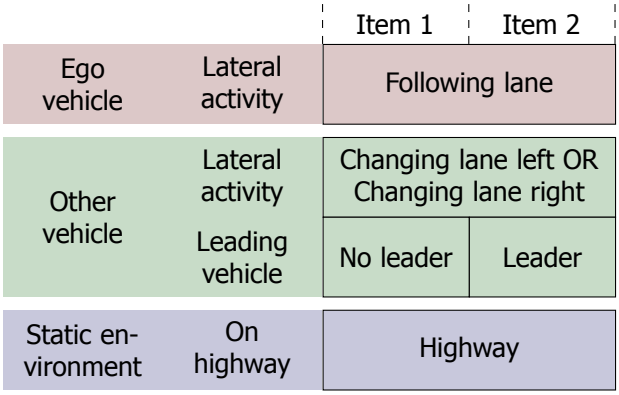


Figure 4.5: Formulation of the scenario category "cut-in" using tags.

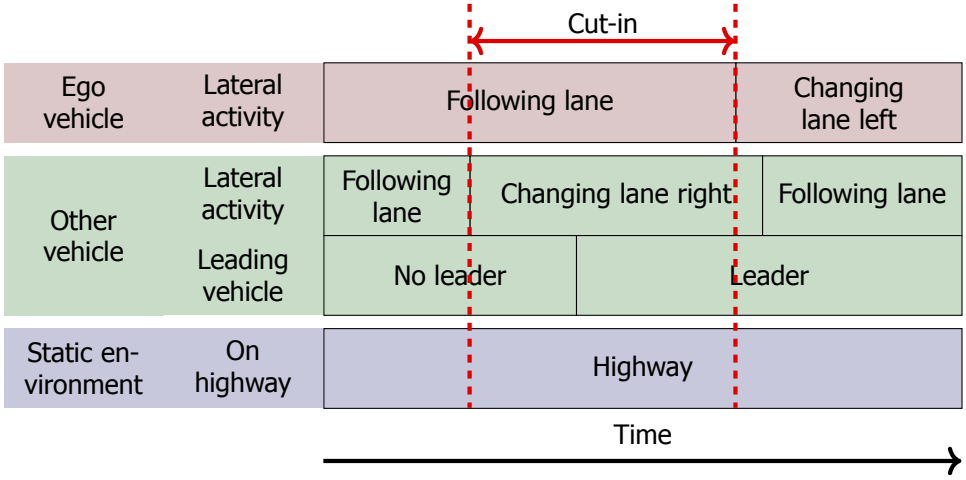


Figure 4.6: Example of tags describing a cut-in. Note that only the tags that are relevant for the cut-in, as defined in Figure 4.5, are shown. Furthermore, whereas there are multiple other vehicles around the ego vehicle, only the other vehicle that performs the cut-in is shown.

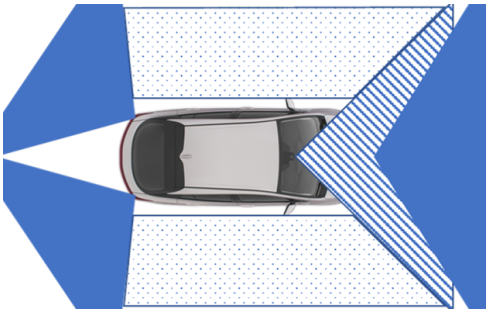


Figure 4.7: Schematic representation of the field of view of the three radars (solid area) and the camera (area filled with lines) that the ego vehicle is equipped with. The positions of vehicles on the left or the right of the ego vehicle (dotted area) are predicted based on previous measurements.

		Item 1	Item 2	Item 3	Item 4
Ego vehicle	Lateral activity	Following lane		Changing lane left	
Other vehicle	Lateral state	Left of ego			Same lane
	Longitudinal state	Behind ego	In front of ego		
Static environment	On highway	Highway			

Figure 4.8: Formulation of the scenario category "overtaking before lane change" using tags.

drivers were asked to drive a prescribed route. The majority of the route is on the highway. To measure the surrounding traffic, the vehicle is equipped with three radars and one camera, as shown in Figure 4.7. The images from the camera are used to estimate the lane line distances [92]. Furthermore, the surrounding traffic is measured by fusing the data of the radars and the camera [92]. While fusing the data of the different sensors, the position of the vehicles that disappear from the sensors' field of view on the left and right of the ego vehicle, see the dotted areas in Figure 4.7, are predicted until the vehicles appear again in the sensors' field of view. In total, four hours of driving are analyzed.

To illustrate the proposed scenario mining approach, two different scenario categories are considered: "cut-in" and "overtaking before lane change". Figures 4.5 and 4.8 show the formulation of these scenario categories using tags. Table 4.3 lists the values of the parameters that are used for the tagging of the data.

The results of the scenario mining are presented in Table 4.4. A false negative (FN) means that a scenario that occurred is not detected and a false positive (FP) means that the scenario mining detects a scenario whereas this scenario does not

Table 4.3: Values of parameters used in the case study.

Parameter	Description	Value
t_s	Sample time	$0.01\,\mathrm{s}$
k_h	Sample window	100
a_{cruise}	Threshold determining the start and end of an acceleration or deceleration activity	$0.1\,\mathrm{m/s^2}$
Δ_v	Minimum speed increase/decrease for an acceleration/deceleration activity	$1\,\mathrm{m/s}$
k_{cruise}	Minimum number of samples for a cruising activity	400
Δ_l	A lane change is detected when the difference between consecutive lane line distances is larger than this threshold	$1\,\mathrm{m}$
v_{lat}	Threshold determining the start and end of a lane change	$0.25\,\mathrm{m/s}$
α_1	Maximum factor of the lane width for a lane change of any other vehicle	0.5
α_2	Minimum factor of the lane width for a lane change of any other vehicle	0.1

Table 4.4: Results of the scenario mining.

Scenario category	FN	FP	TP	Recall	Precision	F1
Cut-in	3	3	33	92 %	92 %	92 %
Overtaking before lane change	1	0	18	95 %	100 %	97 %

occur. The true positives (TP) are the scenarios that are correctly detected. The recall is the ratio of the number of true positives (TP) and the total number of scenarios that occur (TP + FN) and the precision is the ratio of the number of true positives (TP) and the total number of detected scenarios (TP + FP). The F1 score is the harmonic mean of the recall and the precision:

$$\mathrm{F1} = 2 \cdot \frac{\mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}. \tag{4.21}$$

As listed in Table 4.4, 33 out of 36 cut-ins are correctly detected and 3 out of the 36 detected cut-ins are incorrect. This results in an F1 score of 92 %. For the scenario category "overtaking before lane change", 18 out of 19 scenarios are correctly detected and there are no scenarios incorrectly detected. This results in an F1 score of 97 %.

4.6. Discussion

The false detections are a result of inaccurate or missing data. For example, in case of the four false negatives, the other vehicle is not detected at the

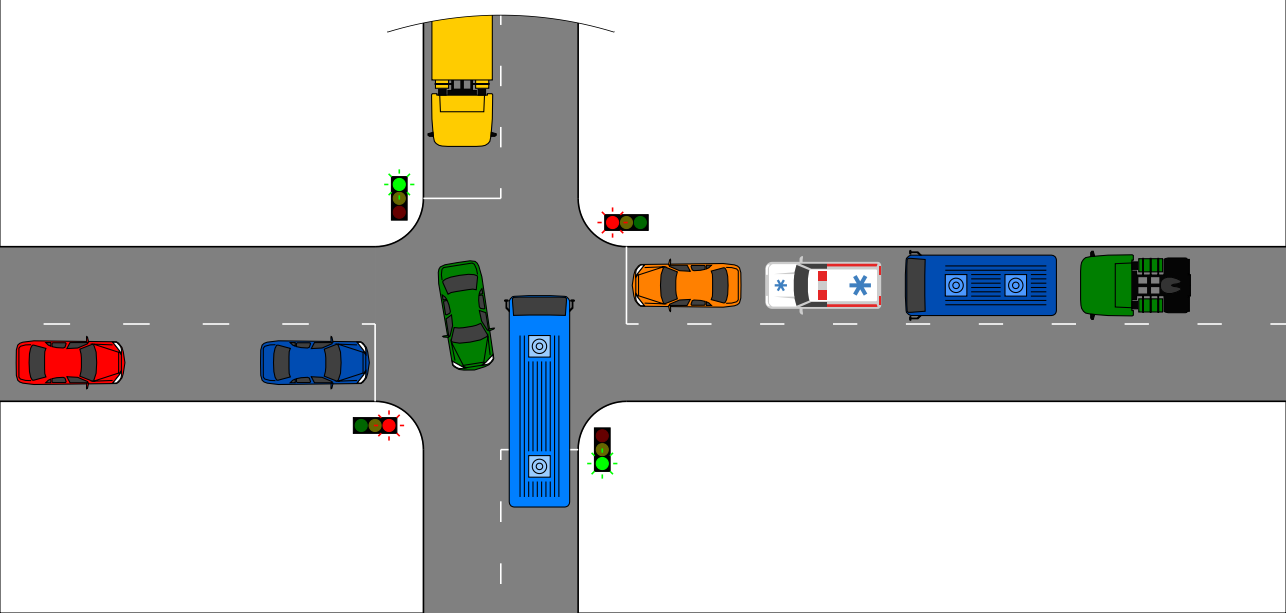
time of the cut-in or overtaking. For one cut-in, this is because another vehicle obstructs the view toward the vehicle at the moment of the cut in. For the other three false negatives, the other vehicles appear from the sensor's blind spot (dotted area in Figure 4.7). The three false positives of the cut-in scenario are a result of inaccurate measurements of the lane line distances. On the one hand, it might be interpreted as that the false detections are due to limitations of the data. On the other hand, for future work, we can expand our work to deal with these limitations of the data. For example, using techniques used for correcting the interpretation of natural language [139], we might be able to correct wrong tags or to add missing tags.

To mine scenarios from a scenario category, the scenario category needs to be represented by a a certain combination of tags, such as shown in Figures 4.5 and 4.8. Provided that there are no new tags required, there are no new algorithms required for mining scenarios from new scenario categories. As a result, it is relatively straightforward to apply the proposed approach for mining scenarios from other scenario categories than the ones presented in our case study. Future work includes more tags, e.g., "turning left" or "turning right", and to consider more actors, e.g., pedestrians and cyclists. This will enable the mining of many more scenarios.

For future research, the analogy between the proposed scenario mining and Natural Language Processing (NLP) could be explored. In NLP, natural language is analyzed by searching for certain combination of words or syllables. Similarly, we are searching for certain combinations of tags. In NLP, n-gram models are successfully used to correct [139] and predict [41] words and to generate text [215]; so n-gram models might be used to correct and predict tags and to generate new scenarios for the assessment of AVs.

4.7. Conclusions

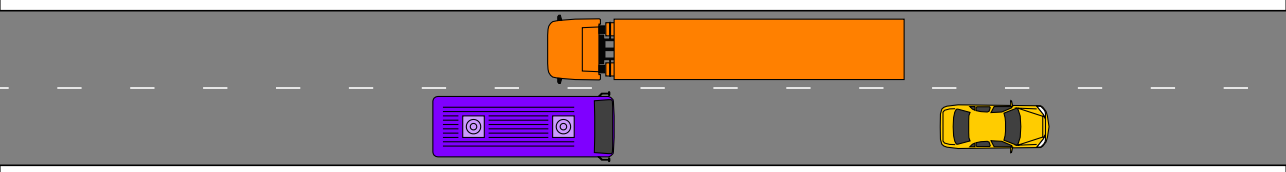
For the scenario-based assessment of Automated Vehicles (AVs), scenarios captured from real-world data collected on the level of individual vehicles can be used to define the tests. We have proposed a two-step approach for mining real-world scenarios from a data set. The first step consists in labeling the data with tags that describe, e.g., the lateral and longitudinal activities of the different actors. The second step mines the scenarios by searching for particular combinations of tags. We have illustrated the approach with two examples, a cut-in and an overtaking before a lane change. These examples demonstrated that the proposed approach is suitable for mining scenarios from real-world data. Future work includes labeling the data with more tags and exploring the possibilities of using techniques that are used in the field of Natural Language Processing (NLP).



5

Generation and evaluation of test scenarios

This chapter is based on:
E. de Gelder, J. Hof, E. Cator, J.-P. Paardekooper, O. Op den Camp, J. Ploeg, and B. De Schutter, *Scenario parameter generation method and scenario representativeness metric for scenario-based assessment of automated vehicles,* IEEE Transactions on Intelligent Transportation Systems (Early access).



The development of assessment methods for the performance of Automated Vehicles (AVs) is essential to enable the deployment of automated driving technologies, due to the complex operational domain of AVs. One candidate is scenario-based assessment, in which test cases are derived from real-world road traffic scenarios obtained from driving data. Because of the high variety of the possible scenarios, using only observed scenarios for the assessment is not sufficient. Therefore, methods for generating additional scenarios are necessary.

Our contribution is twofold. First, we propose a method to determine the parameters that describe the scenarios to a sufficient degree while relying less on strong assumptions on the parameters that characterize the scenarios. By estimating the probability density function (pdf) of these parameters, realistic parameter values can be generated. Second, we present the Scenario Representativeness (SR) metric based on the Wasserstein distance, which quantifies to what extent the scenarios with the generated parameter values are representative of real-world scenarios while covering the actual variety found in the real-world scenarios.

A comparison of our proposed method with methods relying on assumptions of the scenario parameterization and pdf estimation shows that the proposed method can automatically determine the optimal scenario parameterization and pdf estimation. Furthermore, it is demonstrated that our SR metric can be used to choose the (number of) parameters that best describe a scenario. The presented method is promising because the parameterization and pdf estimation can directly be applied to already available importance sampling strategies for accelerating the evaluation of AVs.

5.1. Introduction

An essential aspect in the development of Automated Vehicles (AVs) is the assessment of the quality and performance of AV behavior with respect to safety, comfort, and efficiency [27, 167, 270]. Because public road tests are expensive and time consuming [155, 323], a scenario-based approach has been proposed [15, 70, 94, 171, 229, 233, 270]. With a scenario-based approach, the response of the system-under-test is assessed in many scenarios and for the variations of these scenarios that occur in the real world. Here, a scenario describes the situation the system-under-test is in and how this situation develops over time (in Section 5.3.1, a precise definition of the term scenario is provided). One of the advantages of a scenario-based approach is that the assessment can focus on the more challenging situations by selecting scenarios that are challenging for the system-under-test. As a source of information for the assessment scenarios, real-world driving data has been proposed, thereby guaranteeing that the scenarios represent real-world driving conditions [94, 171, 229].

For the scenario-based assessment approach, it is important that the generated scenarios are representative of scenarios that could happen in real life. In other words, the scenarios should be a representation of the real world [233]. Only then, the results of the assessment can be generalized to the performance of the system-under-test when operating in real life [70]. Furthermore, it is essential that the generated scenarios cover the same variety that is found in real life. Riedmaier *et al.* [233] argue that since an infinite number of situations occur in the real world, the scenario generation methods must provide a large number of variations in order to cover this infinite number of situations.

Our data-driven approach uses observed scenarios to generate parameter values that describe new scenarios. Instead of relying on a predetermined functional form of the signals, such as a vehicle's speed, and fitting parameters to this functional form, we employ a Singular Value Decomposition (SVD) [116] to determine in a data-driven manner the parameters that best describe the scenarios. Next, the probability density function (pdf) of the parameters is estimated, such that the pdf can be used to sample the parameters to generate similar scenarios. To not assume a particular shape of the pdf, Kernel Density Estimation (KDE) [222, 237] is used for the pdf estimation. Furthermore, with KDE, the correlations that might exist among the parameters is modeled. This chapter also proposes a novel metric called the Scenario Representativeness (SR) metric for quantifying to what extent the generated scenarios are representative and cover the actual variety of real-world scenarios. More specifically, this metric uses the Wasserstein distance [241] to compare a set of generated scenarios with a set of observed scenarios.

This chapter is organized as follows. Section 5.2 reviews related works. In Section 5.3, the approach for generating scenarios for the assessment of AVs is explained. Next, Section 5.4 presents the SR metric, i.e., a novel metric for quantifying the performance of the scenario-generation method. A case study is performed in Section 5.5. Section 5.6 discusses relevant implications of our approach and some directions for future research. Conclusions of this chapter are provided in Section 5.7.

5.2. Related works

In this section, first, works concerning the generation of scenarios for the assessment of AVs are reviewed. Next, works related to the SR metric are reviewed.

5.2.1. Scenario generation

The approaches to determine scenarios for the assessment of AVs can be categorized into three kinds [183]: scenarios based on observations of real-world traffic, scenarios based on the functionality that is being assessed, and a combination of these two approaches. The current chapter focuses on the first approach.

In the literature, several methods are proposed to generate scenarios for the assessment based on real-world driving data. Lages *et al.* [176] proposed a method to construct scenarios in a virtual simulation environment by reconstructing the real-world scenarios observed by laser scanners. Zofka *et al.* [327] presented how recorded sensor data can be exploited to create scenarios that might lead to critical situations by modifying parameters of the recorded parameterized scenarios. Stepien *et al.* [271] generate scenarios by sampling scenario parameter values from generalized extreme value distributions, where the distribution parameters are fitted using scenario parameter values extracted from safety-critical scenarios observed in naturalistic driving data. In [70, 100–102, 184, 277], also parameterized scenarios were generated and, in addition, importance sampling techniques were presented that automatically generate scenarios in which the system-under-test shows (safety-) critical behavior. Other approaches to generate scenarios in which the system-under-test shows (safety-) critical behavior are Monte Carlo tree search [169] and genetic programming [58]. Schuldt *et al.* [253] provided a method to generate scenarios using combinatorial algorithms that should ensure that the test cases cover the variety of the possible situations the system-under-test could encounter in real life. More recently, Spooner *et al.* [269] presented a Generative Adversarial Network (GAN) to generate pedestrian crossing scenarios.

In the existing literature, the scenario generation methods for the assessment of AVs have either one or more of the following shortcomings:

- Observed scenarios are replayed without adding more variations [176]. In this case, the total variety of scenarios that is found in real life will not be covered unless unrealistic amounts of data are gathered.
- The scenarios are oversimplified. For example, a vehicle's speed profile follows a predetermined functional form [70, 271, 277].
- Assumptions regarding the scenario parameter distributions are made that potentially compromise the quality of the scenarios. For example, the parameters are assumed to originate from a Gaussian [113] or generalized extreme value [271] distribution, and/or it is assumed that (some of) the parameters are uncorrelated [102].
- Because no pdf of the scenario parameters is known [327], no evaluation can be made of the performance of the system once deployed on the road because it is unknown how realistic and likely the scenarios are.

In Section 5.3, a method is proposed that overcomes these shortcomings.

5.2.2. Scenario representativeness metric

The generated scenarios should represent scenarios that could happen in real life. Whereas different approaches exist in the literature regarding the generation of scenarios for the assessment of AVs, less is known about the comparison of the generated scenarios with real-life traffic. From the mentioned sources in Section 5.2.1, only Feng *et al.* [102] compared their generated scenarios with the ground truth from naturalistic driving data. Feng *et al.* [102] compared the distributions of vehicle speeds and bumper-to-bumper distances between the constructed scenarios and the ground truth. To quantify the similarity between the distributions, the Hellinger distance [47] and the mean absolute error were used. The disadvantages of this approach are that:

1. the generated scenarios may still be substantially different even though the distributions of the vehicle speeds and bumper-to-bumper distances are similar, and
2. only the marginal distributions are considered while the correlation between the vehicle speeds and bumper-to-bumper distances might be completely different.

Whereas little is known about comparing generated scenarios for the scenario-based assessment of AVs with ground truth data, many similarity metrics for comparing two pdfs are known [47]. Well-known metrics are the Minkowski metric [47], which is a generalized version of the Euclidean distance, the f-divergence, which is a generalized version of both the Kullback-Leibler divergence [173] and the Hellinger distance [47], and the Wasserstein metric [241]. For practical reasons, this work uses the Wasserstein metric. As is shown in Section 5.4.2, the Wasserstein distance can be estimated using empirical distributions, i.e., without the need to estimate and evaluate a pdf. The other mentioned metrics require integration over the domain of the pdfs, which will give computational issues since the considered pdfs will have a high dimensionality.

5.3. Scenario generation

To generate realistic scenarios for the assessment of AVs, we use a data-driven approach: observed scenarios are used to generate new scenarios. To do this, the scenarios are parameterized, i.e., parameters are defined that characterize a scenario. For example, the duration of a scenario could be a parameter. Next, the pdf of the parameters is estimated. This pdf can be used to generate parameter values for new scenarios. In addition, the pdf contains the statistical information of the parameters so that the performance of AVs can be estimated [70, 322]. Choosing the parameters that describe a scenario, however, is not trivial:

- Choosing too few parameters might lead to an oversimplification of the actual scenarios. As a result, not all possible variations of a scenario are modeled.

- Too many parameters lead to problems with estimating the pdf, due to the curse of dimensionality [254].

To overcome this problem, we first consider as many parameters as needed for a complete description of the scenarios to avoid the oversimplification of the scenarios. Next, using an SVD, a new set of parameters is created using a linear mapping of the original scenario parameters. Because this new set of parameters is ordered according to the contribution of each of these parameters in describing the variation that exists among the original scenario parameters, we will consider only the most important parameters without losing too much information. In this way, the curse of dimensionality is avoided without relying on a predetermined choice of parameters.

Below, we first explain how to describe a scenario using many parameters. Next, Section 5.3.2 proposes the use of the SVD to reduce the number of parameters. Section 5.3.3 describes how KDE is used to estimate the pdf of the reduced set of parameters and how the estimated KDE can be used to generate scenario parameter values.

5.3.1. Parameterization of scenarios

The first step of our approach is the parameterization of scenarios. There is no single best way to parameterize the scenarios considering the wide variety of scenarios. To deal with this variety, this work distinguishes quantitative scenarios from qualitative scenarios, using the definitions of *scenario* and *scenario category* of [79] (Chapter 2):

Definition 5.1 (Scenario)**.** *A scenario is a quantitative description of the relevant characteristics and activities and/or goals of the ego vehicle(s), the static environment, the dynamic environment and all events that are relevant to the ego vehicle(s) within the time interval between the first and last relevant event.*

Definition 5.2 (Scenario category)**.** *A scenario category is a qualitative description of relevant characteristics and activities and/or goals of the ego vehicle(s), the static environment, and the dynamic environment.*

A scenario category is an abstraction of a scenario and, therefore, a scenario category comprises multiple scenarios [79]. For example, the scenario category "cut-in" comprises all possible cut-in scenarios. The goal of our approach is to determine the optimal parameterization of scenarios of a given scenario category based on a set of observed scenarios of the same scenario category and to estimate the pdf of these parameters that can be used to generate parameter values for new scenarios.

The observed scenarios are described using a time series for the content of the scenario that changes within the time window of the scenario (e.g., the speed of a vehicle) and some additional parameters for the content that is fixed (e.g., the lane width and the duration of the scenario). Here, $y(t) \in \mathbb{R}^{n_y}$ denotes the time series of a scenario with $t \in [t_0, t_1]$, where n_y denotes the dimension of the time series

and t_0 and t_1 denote the start and end time of the scenario, respectively. The n_θ additional parameters are represented by $\theta \in \mathbb{R}^{n_\theta}$.

To deal with the time series, the continuous time interval $[t_0, t_1]$ is discretized, such that two consecutive time instants are $(t_1 - t_0)/(n_\mathrm{t} - 1)$ apart. This gives:

$$\mathbf{y} = \begin{bmatrix} y(t_0) \\ y\left(t_0 + \frac{t_1 - t_0}{n_\mathrm{t} - 1}\right) \\ y\left(t_0 + 2\frac{t_1 - t_0}{n_\mathrm{t} - 1}\right) \\ \vdots \\ y(t_1) \end{bmatrix} \in \mathbb{R}^{n_\mathrm{t} n_y}. \tag{5.1}$$

Note that n_t must be chosen such that no important information is lost during the discretization, i.e., all relevant frequencies must be captured. It depends on the application what the relevant frequencies are. Because in practice, due to the discrete nature of sensor readings, the time series $y(t)$ is obtained at certain specific times rather than on a continuous time interval, it may be required to use interpolation techniques, such as splines [66], to evaluate $\mathbf{y}$.

Let us assume that N_x observed scenarios can be used to generate new scenarios. To indicate that the scenario parameters $\mathbf{y}$ and θ belong to a specific scenario, the index $i \in \{1, \ldots, N_x\}$ is used, i.e., the parameters of the i-th scenario are $\mathbf{y}_i$ and θ_i. To further ease the notation, $\mathbf{y}_i$ and θ_i are combined into one vector x_i:

$$x_i = \begin{bmatrix} \mathbf{y}_i \\ \theta_i \end{bmatrix} \in \mathbb{R}^{n_\mathrm{t} n_y + n_\theta}. \tag{5.2}$$

5.3.2. Parameter reduction using Singular Value Decomposition

As shown in (5.2), $n_x = n_\mathrm{t} n_y + n_\theta$ parameters describe a scenario. Even for small numbers of n_t, n_y, and n_θ, the total number of parameters becomes too large to reliably estimate the joint pdf. One way to avoid this curse of dimensionality is to assume that the parameters are independent, but especially the parameters $y(t_0)$ till $y(t_1)$ in (5.1) are obviously correlated, so assuming that the parameters are independent is not a good solution.

In the field of machine learning, Principal Component Analysis (PCA) is commonly used for dimensionality reduction [3]. As PCA uses the SVD [116], this work uses the SVD to transform the parameters x_i into a lower-dimensional vector of parameters. Before applying the SVD, the parameters are weighted with $\alpha \in \mathbb{R}^{n_x}$ in order to give more or less importance to the n_x parameters. This is particularly useful to compensate for the imbalance in the parameter vector, where the imbalance is caused by the fact that the parameter vector considers the time series $y(t)$ at n_t different times and the additional parameters θ only once. Let us define a matrix that contains the parameters of the N_x scenarios:

$$X = \begin{bmatrix} (\alpha \odot x_1) - \mu & \cdots & \left(\alpha \odot x_{N_x}\right) - \mu \end{bmatrix} \in \mathbb{R}^{n_x \times N_x}, \tag{5.3}$$

where $\odot$ denotes the element-wise product of vectors and $\mu \in \mathbb{R}^{n_x}$ denotes the mean of the weighted scenario parameters:

$$\mu = \frac{1}{N_x} \sum_{i=1}^{N_x} \alpha \odot x_i. \tag{5.4}$$

Using the SVD of X, we obtain:

$$X = U \Sigma V^{\mathsf{T}}. \tag{5.5}$$

Here, both $U \in \mathbb{R}^{n_x \times n_x}$ and $V \in \mathbb{R}^{N_x \times N_x}$ are orthonormal matrices. Therefore, both matrices can be interpreted as rotation matrices in $\mathbb{R}^{n_x}$ and $\mathbb{R}^{N_x}$, respectively. The matrix $\Sigma \in \mathbb{R}^{n_x \times N_x}$ takes the same shape as X. This matrix has only zeros except on (part of) the diagonal. The diagonal contains the so-called singular values, denoted by σ_j with $j \in \{1, \ldots, \bar{N}\}$, $\bar{N} = \min(n_x, N_x)$. These singular values are in decreasing order, i.e.,

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\bar{N}} \geq 0. \tag{5.6}$$

Because of the decreasing singular values, rotating the matrix X from the left with U^{T} transforms the data to a new coordinate system such that the first coordinate has the largest variance compared to the other coordinates. This variance equals σ_1^2. Similarly, the second largest variance equals σ_2^2 and lies on the second coordinate, etc. Because of the decreasing variance, the scenario parameters can be approximated using only the first d coordinates of the new coordinate system, as these d coordinates describe the majority of the variations. So, the scenario parameters of the i-th scenario are approximated by setting $\sigma_j = 0$ for $j > d$:

$$\alpha \odot x_i = \mu + \sum_{j=1}^{\bar{N}} \sigma_j v_{ij} u_j \approx \mu + \sum_{j=1}^{d} \sigma_j v_{ij} u_j, \tag{5.7}$$

where v_{ij} is the (i,j)-th element of V, u_j is the j-th column of U, and d is the number of parameters that are retained. Thus, the n_x parameters of the i-th scenario are approximated using the d parameters $v_{i1}, \ldots, v_{id}$. The singular values $\sigma_1, \ldots, \sigma_d$, the vectors $u_1, \ldots, u_d$, and μ are used to map the new scenario parameters, $v_{i1}, \ldots, v_{id}$, to an approximation of the weighted original scenario parameters, $\alpha \odot x_i$. In Section 5.4, a metric is proposed for evaluation, among others, whether the approximation of (5.7) is acceptable or not.

Remark 5.1. Using the approximation of (5.7), it is not necessary to evaluate the complete SVD of (5.5). Only the first d columns of U and V need to be computed and only the first d singular values. In practice, $d \ll \bar{N}$, so this saves a substantial amount of computation time. ◊

The choice of $d < \bar{N}$ is not trivial. Choosing d too small results in too much loss of detail. Choosing d too large will give problems when estimating the pdf of the new parameters. One method to choose d is to look at the amount of overall

variance of $\alpha \odot x_i$ explained by the first d singular values. The overall variance scales with the sum of the squared singular values [116, p. 77], i.e.,

$$\sum_{i=1}^{N_x} ((\alpha \odot x_i) - \mu)^\mathsf{T} ((\alpha \odot x_i) - \mu) = \sum_{j=1}^{\bar{N}} \sigma_j^2. \tag{5.8}$$

Thus, the first d singular values explain

$$\frac{\sum_{j=1}^{d} \sigma_j^2}{\sum_{j=1}^{\bar{N}} \sigma_j^2} \tag{5.9}$$

of the overall variance. One approach would be to set d such that (5.9) exceeds a certain threshold, such as 0.95. Another way to choose d is by inspecting the actual approximation error in (5.7) and keep increasing d until the approximation error is not too large. Section 5.4 proposes an alternative way to determine d using a metric that quantifies the goal of our generated scenarios, i.e., that the generated scenarios are representing real-world scenarios and cover the actual variety of real-world scenarios.

5.3.3. Estimating the probability density function

Using the approximation of (5.7) based on the SVD, the i-th scenario is described by the vector $\tilde{v}_i$:

$$\tilde{v}_i^\mathsf{T} = \begin{bmatrix} v_{i1} & \cdots & v_{id} \end{bmatrix}. \tag{5.10}$$

Note that the d entries of $\tilde{v}_i$ are linearly uncorrelated with the d entries of $\tilde{v}_m$ $(m \neq i)$[1]. Despite the linear independence, the different entries of $\tilde{v}_i$ may still be dependent due to higher-order correlations; so we treat these d entries as dependent variables.

To estimate the pdf of $\tilde{v}_i$, we propose to use KDE. KDE [222, 237] is often referred to as a non-parametric way to estimate the pdf because KDE does not rely on the assumption that the data are drawn from a given parametric family of probability distributions. Because KDE produces a pdf that adapts itself to the data, it is flexible regarding the shape of the actual underlying distribution of $\tilde{v}_i$. In KDE, the pdf is estimated as:

$$\hat{f}_H(v) = \frac{1}{N_x} \sum_{i=1}^{N_x} K_H(v - \tilde{v}_i). \tag{5.11}$$

Here, $K_H(\cdot)$ is the so-called scaled kernel with a positive definite symmetric bandwidth matrix $H \in \mathbb{R}^{d \times d}$. The kernel $K(\cdot)$ and the scaled kernel $K_H(\cdot)$ are related

[1] This is assuming that $\sigma_d > 0$. With this assumption and because X in (5.3) is defined such that the sum of each row of X equals zero, it is easy to verify that $\frac{1}{N_x} \sum_{m=1}^{N_x} v_{mj} = 0$ for $j \in \{1, \ldots, N_x\}$. Therefore, $\sum_{i=1}^{N_x} \left(v_{ij} - \frac{1}{N_x} \sum_{m=1}^{N_x} v_{mj}\right)\left(v_{ik} - \frac{1}{N_x} \sum_{m=1}^{N_x} v_{mk}\right) = \sum_{i=1}^{N_x} v_{ij} v_{ik} = 0$ for $j \neq k$, where the latter equality follows from the orthonormality of V.

using

$$K_H(u) = |H|^{-1/2} K\left(H^{-1/2}u\right), \tag{5.12}$$

where $|\cdot|$ denotes the matrix determinant. The choice of the kernel function is not as important as the choice of the bandwidth matrix [87, 283]. This article considers the Gaussian kernel[2], which is given by

$$K(u) = \frac{1}{(2\pi)^{d/2}} \exp\left\{-\frac{1}{2}\|u\|_2^2\right\}, \tag{5.13}$$

where $\|u\|_2^2 = u^{\mathrm{T}}u$ denotes the squared 2-norm of u.

A bandwidth matrix of the form $H = h^2 I_d$ is used, where I_d denotes the d-by-d identity matrix. The bandwidth h is determined with leave-one-out cross-validation [86] because this minimizes the difference between the real pdf and the estimated pdf according to the Kullback-Leibler divergence [283, 319].

To sample scenario parameters using $\hat{f}_H(\cdot)$, first, an integer $i \in \{1, \dots, N_x\}$ is randomly chosen with each integer having equal likelihood. Next, a random sample is drawn from a Gaussian with covariance H and mean $\tilde{v}_i$. Then, using the approximation in (5.7), the scenario parameters are calculated.

As far as the computational effort is concerned, sampling the scenario parameters from a KDE is efficient because there is no need to actually evaluate the pdf. Determining the optimal bandwidth matrix requires more computational effort, but this only has to be done once per data set. The computational complexity of cross-validation methods for the bandwidth estimation typically scales with N_x^2 [119].

5.4. Scenario Representativeness metric

Ideally, the parameters of the generated scenarios are sampled from the same distribution that underlies the real-world scenario parameters. The problem is that this distribution is unknown. Nevertheless, it is possible to define a metric that quantifies the similarity of the distribution that is used to generate scenario parameters and the distribution that underlies the real-world scenario parameters. Section 5.4.1 further explains the goal of this metric, which we call the Scenario Representativeness (SR) metric. Next, Section 5.4.2 explains the Wasserstein distance [241], which is then applied to derive our metric in Section 5.4.3.

5.4.1. Scenario comparison problem

The set of observed scenarios, described using the parameters x_i, $i \in \{1, \dots, N_x\}$, are used for generating the scenario parameters. To ease the notation, let us denote the set of observed scenarios by $\mathcal{X} = \{x_1, \dots, x_{N_x}\}$. This chapter assumes that these scenarios — that are comprised by the same scenario category — are independently and identically distributed according to the distribution $f(\cdot) : \mathbb{R}^{n_x} \to \mathbb{R}$. Let us denote the set of generated scenario parameter vectors by $\mathcal{W} = \{w_1, \dots, w_{N_w}\}$ where

[2]The advantage of the Gaussian kernel is that it gives the possibility to calculate a metric that quantifies the completeness of the data [72] (Chapter 3) and to apply conditional sampling when generating scenario parameters [75] (Chapter 6). Both these topics are out of scope of this chapter.

$w_i \in \mathbb{R}^{n_x}$, $i \in \{1, \dots, N_w\}$ are similarly parameterized as in (5.2) and N_w is the number of generated scenario parameter vectors. Let $\hat{f}(\cdot) : \mathbb{R}^{n_x} \to \mathbb{R}$ denote the pdf of the generated scenario parameter vectors, which is obtained from $\hat{f}_H(\cdot) : \mathbb{R}^d \to \mathbb{R}$ under a change of variable according to the approximation in (5.7). As later appears, it is not needed to have an explicit definition for $\hat{f}(\cdot)$. Ideally, $\hat{f}(\cdot)$ is equal to $f(\cdot)$. So our metric aims to quantify the similarity of $\hat{f}(\cdot)$ and $f(\cdot)$.

To estimate the similarity between $\hat{f}(\cdot)$ and $f(\cdot)$, we cannot simply compare $\mathcal{W}$ with $\mathcal{X}$. In that case, taking $\mathcal{W} = \mathcal{X}$ would give us the best result, but this is undesirable because, ideally, the scenarios of the generated parameters cover the whole variety of real-world scenarios and not just the variety that have been observed in $\mathcal{X}$. Therefore, another set of scenarios is needed that can be used to test. Let us assume that such a set of scenarios is available, denoted by $\mathcal{Z} = \{z_1, \dots, z_{N_z}\}$ where $z_i \in \mathbb{R}^{n_x}$, $i \in \{1, \dots, N_z\}$ are independently and identically distributed according to $f(\cdot)$. Thus, $\mathcal{X}$ and $\mathcal{Z}$ can be regarded as a training and test set, respectively.

In summary, the goal is to find a metric that quantifies the similarity of $\hat{f}(\cdot)$ and $f(\cdot)$ using the sets of observed scenario parameters $\mathcal{X}$ and $\mathcal{Z}$ and the set of scenario parameters $\mathcal{W}$, generated based on $\mathcal{X}$.

5.4.2. Empirical Wasserstein metric

The p-th Wasserstein metric ($p \geq 1$) [241] is used to compare two pdfs $\xi(\cdot)$ and $\eta(\cdot)$ defined on the set $\mathcal{U}$. This metric is defined as follows:

$$W_p(\xi, \eta) = \left(\inf_{\gamma \in \Gamma(\xi,\eta)} \left\{ \int_{\mathcal{U}\times\mathcal{U}} (\Delta(u,v))^p \, \mathrm{d}\gamma(u,v) \right\} \right)^{1/p}. \tag{5.14}$$

Here, $\Delta(u, v)$ denotes the distance from u to v, which will be defined below, and $\Gamma(\xi, \eta)$ denotes the set of joint distributions of (u, v) that have marginal distributions $\xi(\cdot)$ and $\eta(\cdot)$. Intuitively, if the pdfs $\xi(\cdot)$ and $\eta(\cdot)$ are seen as two piles of earth having a different shape with mass 1, then (5.14) calculates the minimum cost of converting one pile of earth with shape $\xi(\cdot)$ into a pile of earth with shape $\eta(\cdot)$. Therefore, the Wasserstein metric is also referred to as the earth mover's distance [240].

In our case, the goal is to have a metric to compare $f(\cdot)$ and $\hat{f}(\cdot)$. Because $f(\cdot)$ is unknown, its approximation based on $\mathcal{Z}$ is considered:

$$f(z) \approx \frac{1}{N_z} \sum_{i=1}^{N_z} \delta(z - z_i), \; z \in \mathbb{R}^{n_x}, \tag{5.15}$$

where $\delta(\cdot)$ denotes the Dirac delta function. Considering the high dimension of z, numerical approximation of the integral of the Wasserstein metric (5.14) using this approximation and $\hat{f}(\cdot)$ would require so many evaluations of $\hat{f}(\cdot)$ that it becomes computationally infeasible. Therefore, the empirical estimation of the Wasserstein metric (5.14) is considered, which makes use of the empirical estimation of $\hat{f}(\cdot)$:

$$\hat{f}(w) \approx \frac{1}{N_w} \sum_{i=1}^{N_w} \delta(w - w_i), \; w \in \mathbb{R}^{n_x}. \tag{5.16}$$

Substituting the empirical estimations of (5.15) and (5.16) for $\xi(\cdot)$ and $\eta(\cdot)$, respectively, into (5.14), leads to the so-called empirical Wasserstein metric [266], which is defined as:

$$\tilde{W}_p(\mathcal{Z},\mathcal{W}) = \left(\inf_{T} \sum_{i=1}^{N_z} \sum_{j=1}^{N_w} \big(\Delta(z_i,w_j)\big)^p T_{ij} \right)^{1/p}, \tag{5.17}$$

where T_{ij} is the (i,j)-th element of the transportation matrix T that is subject to the following conditions:

$$\sum_{i=1}^{N_z} T_{ij} = \frac{1}{N_w} \quad \forall\, j \in \{1,\dots,N_w\}, \tag{5.18}$$

$$\sum_{j=1}^{N_w} T_{ij} = \frac{1}{N_z} \quad \forall\, i \in \{1,\dots,N_z\}, \tag{5.19}$$

$$T_{ij} \geq 0 \quad \forall\, i \in \{1,\dots,N_z\},\ j \in \{1,\dots,N_w\} \tag{5.20}$$

For the distance function, we will use the 2-norm of the difference of the scenario parameters after scaling the scenario parameters according to the weights α that we also used in Section 5.3.2:

$$\Delta(z,w) = \|(\alpha \odot z) - (\alpha \odot w)\|_2. \tag{5.21}$$

5.4.3. Metric for testing scenario representativeness

The empirical Wasserstein metric $\tilde{W}_p(\mathcal{Z},\mathcal{W})$ is an approximation of the Wasserstein metric $W_p(f,\hat{f})$. As one might expect, using an infinite number of scenarios, i.e., for $N_z \to \infty$ and $N_w \to \infty$, the empirical Wasserstein metric approaches the Wasserstein metric with probability 1 [266]. The problem is that N_z and N_w are not infinite. In addition, whereas a fairly large number for N_w can be chosen, as it is only limited by the available computational resources, to increase N_z, more data are needed and this is generally expensive. Therefore, this section proposes a metric that is different from (5.17).

Our proposed SR metric is based on the following intuition: Suppose that $\hat{f}$ is indeed an approximation of f. Because $\mathcal{X}$ and $\mathcal{Z}$ are based on the same underlying pdf, i.e., f, it is expected that $\tilde{W}_p(\mathcal{X},\mathcal{W})$ is similar to $\tilde{W}_p(\mathcal{Z},\mathcal{W})$. If, however, $\tilde{W}_p(\mathcal{X},\mathcal{W})$ is significantly smaller than $\tilde{W}_p(\mathcal{Z},\mathcal{W})$, it suggests overfitting of the training data because the generated scenario parameters are too much skewed towards the training data $\mathcal{X}$. To penalize overfitting of the training data, our SR metric includes a penalty in case $\tilde{W}_p(\mathcal{Z},\mathcal{W})$ is larger than $\tilde{W}_p(\mathcal{X},\mathcal{W})$. Thus, the SR metric becomes:

$$M_p(\mathcal{W},\mathcal{Z},\mathcal{X}) = \tilde{W}_p(\mathcal{Z},\mathcal{W}) + \beta\big(\tilde{W}_p(\mathcal{Z},\mathcal{W}) - \tilde{W}_p(\mathcal{X},\mathcal{W})\big). \tag{5.22}$$

Here, β is the weight of the penalty. The case study in Section 5.5 demonstrates empirically that $M_p(\mathcal{W},\mathcal{Z},\mathcal{X})$ of (5.22) better correlates with the Wasserstein metric

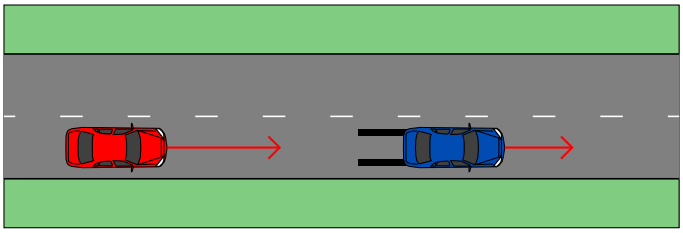


Figure 5.1: Schematic representation of the scenario category "leading vehicle decelerating (LVD)". The left vehicle is the ego vehicle.

of (5.14) than the empirical Wasserstein metric of (5.17) and a method to choose β.

5.5. Case study

To illustrate the proposed method for generating the scenario parameters (Section 5.3) and the SR metric (Section 5.4), these are applied in a case study. Section 5.5.1 explains the scenario categories that are considered in the case study and describes the choices that are made regarding the scenario parameterization. Section 5.5.2 illustrates the approximation of the original parameterization using the SVD. Next, the scenario parameter generation method is demonstrated in Section 5.5.3. Section 5.5.3 also shows that the SR metric (5.22) can be used to choose d. Our method for generating scenario parameters is compared with other methods in Section 5.5.4. Section 5.5.5 demonstrates that the SR metric (5.22) better correlates with the Wasserstein metric (5.14) than the empirical Wasserstein metric (5.17).

5.5.1. Scenario categories and parameterization

In this case study, two scenario categories are considered. The first scenario category, labeled leading vehicle decelerating (LVD), involves an ego vehicle that is following another vehicle that decelerates, see Figure 5.1. As a result, the ego vehicle might need to brake or change direction to avoid contact with the vehicle that decelerates. The second scenario category considers a vehicle that performs a cut-in, such that this vehicle becomes the leading vehicle of the ego vehicle, see Figure 5.2. Depending on the speed and timing of the vehicle that performs a cut-in, the ego vehicle might need to brake or change direction to avoid a crash.

To obtain the scenarios, the data set described in [221] is used. The data were recorded from a single vehicle in which 20 drivers were asked to drive a prescribed route, resulting in 63 hours of data containing 1150 LVD scenarios and 289 cut-in scenarios. The majority of the route was on the highway. To measure the surrounding traffic, the vehicle was equipped with three radars and one camera. The surrounding traffic was measured by fusing the data of the radars and the camera as described in [92]. To extract the LVD and cut-in scenarios from the data set with the fused data, we searched for particular (combinations of) activities in

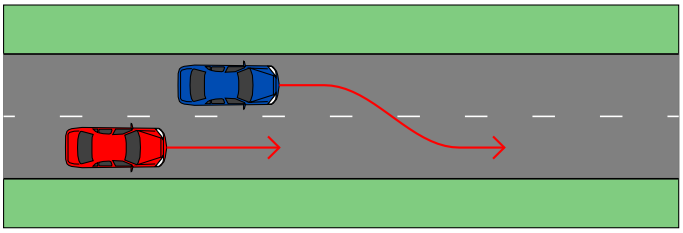


Figure 5.2: Schematic representation of the scenario category "cut-in". The left vehicle is the ego vehicle.

the data: a deceleration activity of a leading vehicle indicates an LVD scenario and a lane change of another vehicle that becomes the leading vehicle indicates a cut-in scenario. For more information on the process of extracting the scenarios, see [73] (Chapter 4).

From the 1150 LVD scenarios, the training uses $80\,\%$ (so $N_x = 920$) and the testing uses the remaining $20\,\%$ (so $N_z = 230$) as this 80/20 ratio is commonly used for splitting the data into a training set and a test set. The training data are used for generating $N_w = 10000$ new scenario parameter vectors. To describe the decelerating behavior of the leading vehicle, the acceleration of the leading vehicle at $n_\mathrm{t} = 50$ time instants is used ($n_y = 1$). As additional parameters, the duration of the scenario, $t_1 - t_0$, the initial speed of the leading vehicle, and the initial time gap between the leading vehicle and the ego vehicle are considered ($n_\theta = 3$). Thus, $n_x = 53$. In Figure 5.3, the speed of the leading vehicle of 100 randomly-selected observed LVD scenarios are shown. The k-th weight, α_k, is obtained by dividing a chosen constant β_k by the standard deviation of the k-th parameter:

$$\alpha_k = \frac{\beta_k}{\sqrt{\frac{1}{N_x}\sum_{i=1}^{N_x}\left((x_i)_k - \bar{x}_k\right)^2}}, \tag{5.23}$$

with $(x_i)_k$ denoting the k-th element of x_i and $\bar{x}_k = \frac{1}{N_x}\sum_{i=1}^{N_x}(x_i)_k$. In this way, the contribution of the k-th parameter to the overall variance (see (5.8)) only depends on β_k. When choosing $\beta_1 = \ldots = \beta_{53}$, the acceleration of the leading vehicle would contribute 50 times more to the overall variance of (5.8) because $n_\mathrm{t} = 50$ elements are used to describe the acceleration. For the LVD scenarios, we want to give the acceleration the same importance as each of the other parameters, so we choose $\beta_1 = \ldots = \beta_{50} = 1/\sqrt{n_\mathrm{t}}$ and $\beta_{51} = \beta_{52} = \beta_{53} = 1$.

From the 289 cut-in scenarios, $80\,\%$ are used for training (so $N_x = 231$) and $20\,\%$ are used for testing (so $N_z = 58$). Both cut-in scenarios from the left and from the right are considered. The training data are used for generating $N_w = 10000$ parameter vectors that describe cut-in scenarios. A cut-in scenario is described using the speed of the vehicle that performs the lane change and its lateral position with respect to the center of the ego vehicle's lane (so $n_y = 2$) at $n_\mathrm{t} = 50$ time instants. In case of a cut-in scenario from the left, the lateral position is positive when the cutting-in vehicle is on the left of the center of ego vehicle's lane and vice

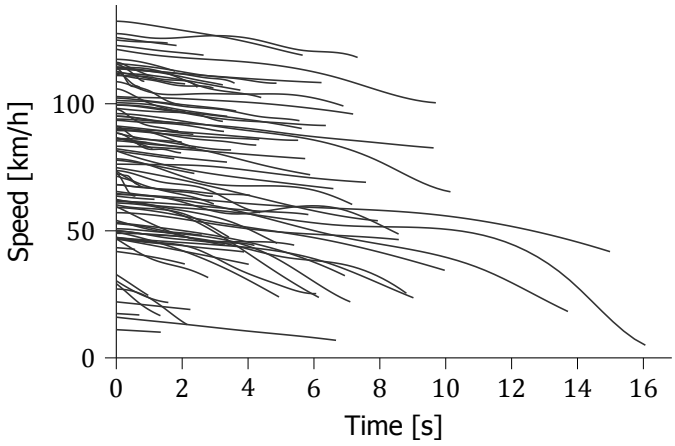


Figure 5.3: Speed of the leading vehicle during 100 randomly-selected observed LVD scenarios. For plotting purposes, the starting time of each scenario is set to 0.

versa for a cut-in scenario from the right. Furthermore, $n_\theta = 3$ extra parameters are used to describe a cut-in scenario: the duration of the scenario, the initial speed of the ego vehicle, and the initial longitudinal position of the cutting-in vehicle with respect to the ego vehicle. Thus, $n_x = 103$. To give the same importance to the speed of the vehicle that performs the lane change, its lateral position, and the 3 extra parameters, the weights are calculated using (5.23) with $\beta_1 = \ldots = \beta_{100} = 1/\sqrt{n_t}$ and $\beta_{101} = \beta_{102} = \beta_{103} = 1$.

5.5.2. Approximation of scenarios with SVD

As explained in Section 5.3.3, using too many parameters will lead to poor estimations of the pdf of the parameters. We use an SVD to obtain a reduced number of parameters that best describe the original scenario parameters. This section illustrates the approximation of the original scenario parameters using the parameters obtained after applying the SVD.

Following the approximation of (5.7), the scaled parameter vector, $\alpha \odot x_i$, is approximated using a linear combination of the first d columns of U, i.e., $u_1, \ldots, u_d$. In Figure 5.4 and Table 5.1, μ and the first four columns of U are shown for the LVD scenarios. For an easier interpretation, the original scaling of the parameters by α is undone via the element-wise division by α. Figure 5.4 shows that the average scenario starts with a deceleration of about $0.4\,\mathrm{m/s^2}$ and ends with a deceleration of about $0.8\,\mathrm{m/s^2}$. Table 5.1 shows that the average scenario duration is $4.73\,\mathrm{s}$, the average initial speed of the leading vehicle is $22.11\,\mathrm{km/h}$, and the average initial time gap is $1.49\,\mathrm{s}$. Since each scenario is estimated by combining the curves in Figure 5.4 and values in Table 5.1, it can be seen that the approximations do not contain complex acceleration curves. In other words, the accelerations will be smoothed and the details may get lost. The amount of smoothing depends on d, i.e., the number of vectors of U that are used to approximate the original

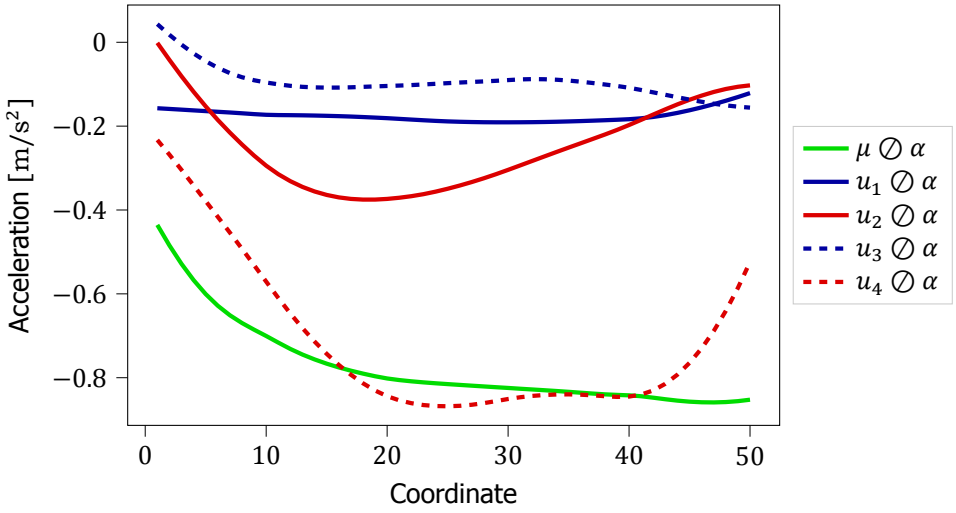


Figure 5.4: The first $n_t = 50$ coordinates of μ and the first four columns of U after scaling with α for the LVD scenarios. Note that $\oslash$ denotes element-wise division.

Table 5.1: The last $n_\theta = 3$ coordinates of μ and the first four columns of U after scaling with α for the LVD scenarios. Note that $\oslash$ denotes element-wise division.

	Coordinate 51 Scenario duration	Coordinate 52 Initial speed	Coordinate 53 Initial time gap
$\mu \oslash \alpha$	4.73 s	22.11 km/h	1.49 s
$u_1 \oslash \alpha$	−1.50 s	−15.17 km/h	0.28 s
$u_2 \oslash \alpha$	−3.09 s	12.22 km/h	−0.06 s
$u_3 \oslash \alpha$	1.15 s	−16.52 km/h	0.29 s
$u_4 \oslash \alpha$	1.32 s	−2.88 km/h	−0.08 s

parameter vector. Choosing the value of d is a trade-off: a higher value of d leads to less smoothing and, therefore, a smaller approximation error, but choosing d too large leads to problems when estimating the pdf of the new parameters.

Figure 5.5 shows five LVD scenarios. These selected LVD scenarios correspond to the five LVD scenarios that require the highest average deceleration of the following vehicle. The line with the "1" denotes the LVD scenario that requires the highest average deceleration. Table 5.2 lists the values of $\sigma_j v_{ji}$ for $j \in \{1, \dots, d\}$ with $d = 4$ that are used to approximate the original scenarios according to the approximation in (5.7). The red lines in Figure 5.5 show the approximated speed of the five LVD scenarios. Table 5.2 shows the initial time gaps of the five scenarios shown in Figure 5.5. These five scenarios illustrate that the accelerations are smoothed, but the main characteristics of the scenarios are captured by the approximations: the average deceleration, the scenario duration, the initial speed, and the initial time gap are well approximated.

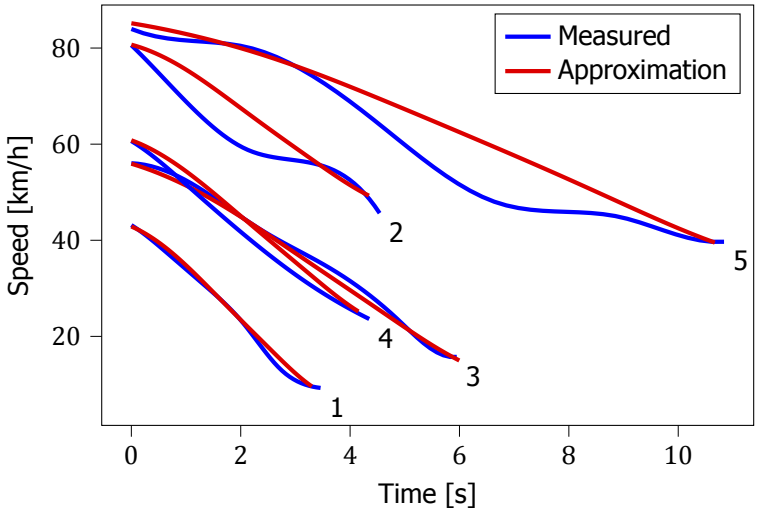


Figure 5.5: Five scenarios that require the highest average deceleration of the following vehicle. The blue lines denote the observed scenarios and the red lines denote their approximations based on the $d = 4$ new parameters. The corresponding initial time gaps are listed in Table 5.2.

5.5.3. Generating scenario parameters

An important parameter for the generation of the scenario parameter vectors is the number of reduced parameters (d). One approach is to look at the so-called explained variance of (5.9) of the first d singular values, see Table 5.3. The first four singular values already explain $90.4\,\%$ of the variance for the LVD scenarios, so $d = 4$ might be a suitable choice. In Figure 5.6, the speed of the leading vehicle of 100 generated LVD scenarios is shown using $d = 4$.

Another way to determine d is to use the SR metric $M_p(\mathcal{W}, \mathcal{Z}, \mathcal{X})$ defined in (5.22). In Figure 5.7, the result is shown when applying this metric with $p = 1$, alongside with the empirical Wasserstein metric $\tilde{W}_1(\mathcal{Z}, \mathcal{W})$ of (5.17) and the penalty $\tilde{W}_1(\mathcal{Z}, \mathcal{W}) - \tilde{W}_1(\mathcal{X}, \mathcal{W})$. Each point in Figure 5.7 represents the median[3] when applying the metric 200 times, each time with a different (random) partition of the training data $\mathcal{X}$ and test data $\mathcal{Z}$. The standard deviation of the medians in Figure 5.7, estimated using bootstrapping [91], is less than 0.005. For the SR metric, the penalty is weighted using $\beta = 0.25$. The choice of $\beta = 0.25$ is justified in Section 5.5.5.

The most left points in Figure 5.7 represent the metric in case the set of training data $\mathcal{X}$ is directly used to sample the scenario parameters instead of the approach of Section 5.3. Here, $\mathcal{W}$ is a selection with replacement of N_w scenarios from $\mathcal{X}$, i.e.:

$$w_i = x_{\lfloor u \rfloor},\ u \sim U(1, N_x + 1),\ \forall i \in \{1, \dots, N_w\}, \tag{5.24}$$

where $U(1, N_x + 1)$ denotes the continuous uniform distribution with boundaries 1

[3]We preferred to use the median instead of the mean, such that the result is less influenced by outliers [82].

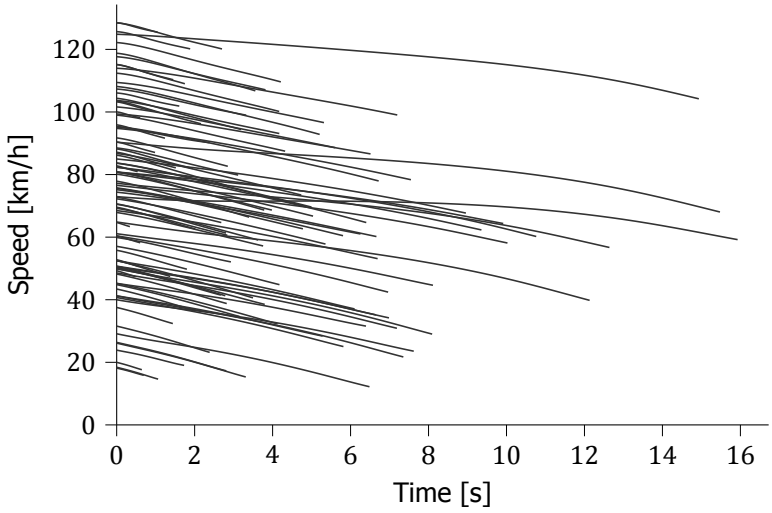


Figure 5.6: Speed of the leading vehicle during 100 generated LVD scenarios.

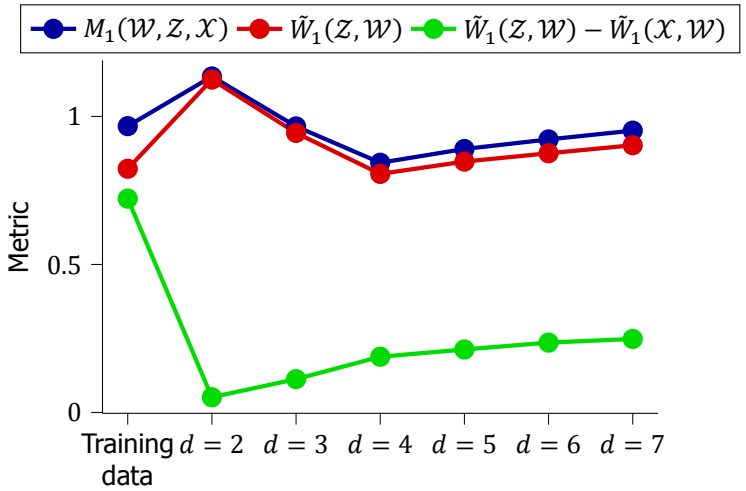


Figure 5.7: Medians of the metrics for the set of generated LVD scenario parameters. Note that $d = 1$ is excluded because its metrics are an order of magnitude higher than for $d = 2$ and would, therefore, not be visible with the current scaling of the y-axis.

Table 5.2: Initial time gaps of the five scenarios that require the highest average deceleration of the following vehicle. The corresponding speeds are shown in Figure 5.5. The values $\sigma_j v_{ij}$, $j \in \{1, 2, 3, 4\}$, are used to approximate the corresponding scenarios.

#	$\sigma_1 v_{i1}$	$\sigma_2 v_{i2}$	$\sigma_3 v_{i3}$	$\sigma_4 v_{i4}$	Initial time gap	
					Original	Approximated
1	2.21	0.30	−0.03	2.16	1.91 s	1.91 s
2	−0.28	0.75	−0.46	1.56	1.08 s	1.11 s
3	0.61	−0.21	−0.42	1.53	1.43 s	1.43 s
4	1.74	0.25	0.58	1.63	2.00 s	2.00 s
5	−1.74	−0.71	−0.49	1.30	0.81 s	0.80 s

Table 5.3: Explained variance according to (5.9).

d	Leading vehicle decelerating	Cut-in
1	36.9 %	36.9 %
2	63.0 %	63.3 %
3	78.0 %	84.5 %
4	90.4 %	94.1 %
5	94.5 %	96.9 %
6	96.7 %	99.0 %
7	98.2 %	99.6 %
8	99.2 %	99.8 %

and $N_x + 1$, and $\lfloor \cdot \rfloor$ denotes the floor function. Using the training data directly for "generating scenarios" leads to a low empirical Wasserstein metric. The downside is that there is not much variation among the generated scenarios. Therefore, the penalty is also the highest, which results in $M_1(\mathcal{W}, \mathcal{Z}, \mathcal{X}) \approx 0.967$. Looking at $d = 4$, the empirical Wasserstein metric is approximately similar compared to when the training set is directly used. Due to the sampling of the scenario parameters from the KDE, the generated scenarios contain more variation than the training set, resulting in a lower penalty and, therefore, a lower metric evaluation of $M_1(\mathcal{W}, \mathcal{Z}, \mathcal{X}) \approx 0.843$. Increasing d even further results in higher metric evaluations. So based on the proposed metric, $d = 4$ seems the right choice.

Figure 5.8 shows the results of the generation of the cut-in scenario parameters in a similar way as Figure 5.7. The standard deviation of all points in Figure 5.8 is less than 0.008. The lowest penalty is obtained with $d = 2$, but the higher empirical Wasserstein distance suggests that too much information is lost. The best result, i.e., where the SR metric, $M_1(\mathcal{W}, \mathcal{Z}, \mathcal{X})$, is minimal, is obtained at $d = 3$.

5.5.4. Comparison with other approaches

Our proposed method utilizes an SVD to obtain the scenario parameters and multivariate KDE to estimate the pdf of these parameters. To illustrate the advantages of

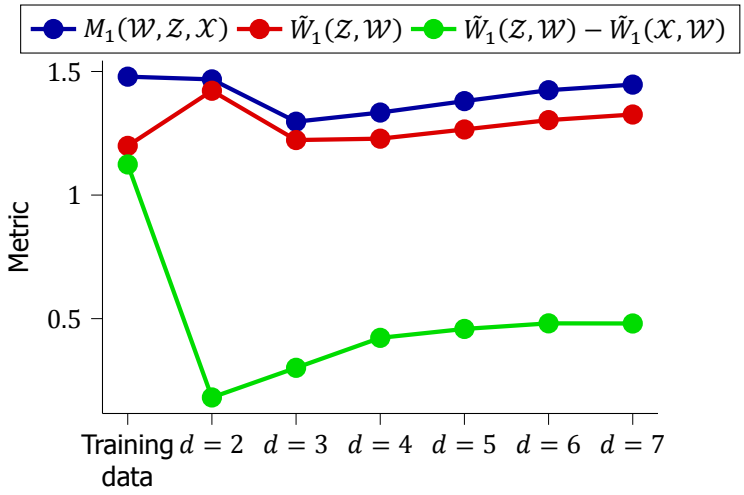


Figure 5.8: Medians of the metrics for the set of generated cut-in scenario parameters.

5

these choices, the results of our method are compared with alternative approaches. First, instead of using an SVD for obtaining the parameters, a fixed parameterization is used, such as in [70, 277, 327]. Second, instead of using KDE to estimate the pdf of the parameters, a Gaussian distribution like in [113] is assumed. Third, the parameters are assumed to be independent.

When using a fixed parameterization for the LVD scenario, four parameters describe the scenario [70]: the speed reduction of the leading vehicle, the final speed of the leading vehicle, the duration of the scenario, and the initial time gap between the leading vehicle and the ego vehicle. The speed of the leading vehicle is assumed to follow a sinusoidal function, such that the acceleration at the start and at the end of the scenario equals zero. In case of the cut-in scenario, five parameters describe the scenario: the mean speed of the vehicle cutting in, its initial lateral position with respect to the center of the ego vehicle's lane, the duration of the scenario, the initial speed of the ego vehicle, and the initial longitudinal position of the vehicle cutting in with respect to the ego vehicle. The speed of the vehicle cutting in is assumed to be constant. Its lateral position is assumed to follow a sinusoidal function, such that the vehicle ends at the center of the ego vehicle's lane. For estimating the pdf of these parameters, the comparison considers four possibilities: multivariate KDE, multiple univariate KDEs, a multivariate Gaussian distribution, and multiple univariate Gaussian distributions.

Table 5.4 shows the results of the different approaches for generating scenario parameters. For the LVD scenarios, our proposed approach (top row in Table 5.4) resulted in the lowest $M_1(\mathcal{W},\mathcal{Z},\mathcal{X})$. For the cut-in scenarios, it is interesting to note that the scores are not very different as long as SVD is used to obtain the parameters. This is partly explained by the smaller data set because this results

Table 5.4: Medians of the metric $M_1(\mathcal{W}, \mathcal{Z}, \mathcal{X})$ with different approaches for generating scenario parameter values.

Parameters	Distribution	Dependency	LVD	Cut-in
SVD	KDE	Dependent	0.84	1.30
SVD	Gaussian	Dependent	1.00	1.33
SVD	KDE	Independent	0.99	1.28
SVD	Gaussian	Independent	1.00	1.33
Fixed	KDE	Dependent	2.65	1.70
Fixed	Gaussian	Dependent	2.58	1.71
Fixed	KDE	Independent	2.31	1.67
Fixed	Gaussian	Independent	4.76	1.69

in a higher bandwidth[4] that makes the KDE result with the Gaussian kernel look more like a Gaussian distribution. Using SVD and KDE while assuming that the parameters are independent, results in an even better result: 1.28 instead of 1.30 (with a standard deviation of 0.005). This indicates that assuming that the three parameters obtained with the SVD are independent, is acceptable.

5.5.5. Evaluating the scenario representativeness metric

To determine whether our proposed metric (5.22) correlates better with the Wasserstein metric (5.14) than the empirical Wasserstein metric (5.15), the Wasserstein metric (5.14) needs to be known. This is not possible because the true underlying distribution of the data is unknown. To estimate the Wasserstein metric (5.14), the empirical Wasserstein metric (5.15) can be used with large numbers of test scenarios and generated scenario parameters, i.e., with large values of N_z and N_w, respectively. Since a large number of test scenarios is not available to us, we assume a certain distribution for $f(\cdot)$ from which the training data and the test data are generated. The approach is as follows (the numbers are for the LVD scenarios and, in parenthesis, the numbers for the cut-in scenarios are shown):

1. Based on the original 1150 (289) scenarios, obtained from the data, the following sets of scenario parameters are generated using the proposed approach explained in Section 5.3 with $d = 4$ ($d = 3$):
   - A new set of training data $\mathcal{X}^*$ of size $N_x = 920$ ($N_x = 231$);
   - A new set of test data $\mathcal{Z}^*$ of size $N_z = 230$ ($N_z = 58$); and
   - A large set of test data $\mathcal{Z}^*_{\text{large}}$ of size $N_z = 10000$ ($N_z = 10000$).
2. Based on $\mathcal{X}^*$, $N_w = 10000$ ($N_w = 10000$) scenario parameters are generated and collected in a set $\mathcal{W}^*$.

[4] On average, the bandwidth is about 1.5 to 2 times larger for the cut-in scenarios compared to the LVD scenarios.

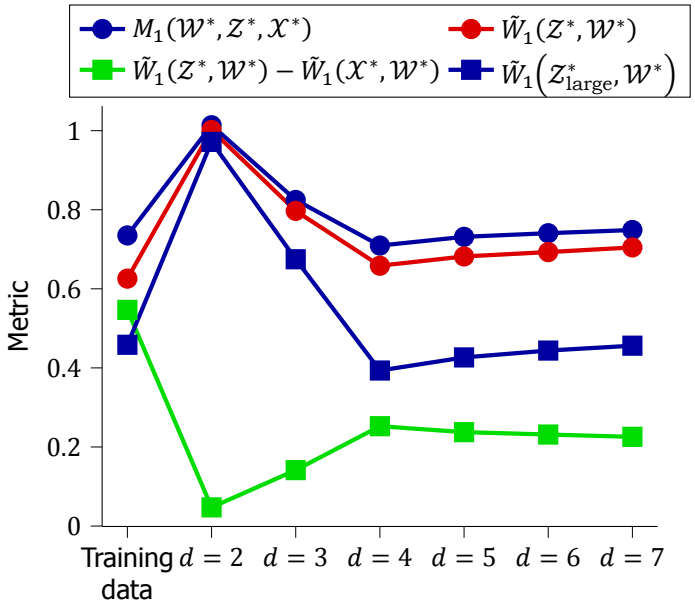


Figure 5.9: Medians of the metrics for the set of $N_w = 10000$ generated LVD scenario parameter vectors. In this case, the $N_x = 920$ scenarios of $\mathcal{X}^*$ are sampled from $\hat{f}_H(\cdot)$ of (5.11), where $\hat{f}_H(\cdot)$ is based on the original data set $\mathcal{X}$.

3. Our proposed metric is computed using $\mathcal{W}^*$, $\mathcal{Z}^*$, and $\mathcal{X}^*$: $M_1(\mathcal{W}^*, \mathcal{Z}^*, \mathcal{X}^*)$ with $\beta = 0.25$.
4. The Wasserstein metric of (5.14) is estimated using the empirical Wasserstein metric of (5.17) with $\mathcal{W}^*$ and $\mathcal{Z}^*_{\text{large}}$. Note: to approximate the Wasserstein metric of (5.14) using the empirical Wasserstein metric of (5.17), both $\mathcal{W}^*$ and $\mathcal{Z}^*_{\text{large}}$ need to be large (but not necessarily the same) in size.

We have repeated this approach 200 times, each time with a different (random) partition of the training data $\mathcal{X}$ and test data $\mathcal{Z}$. Figures 5.9 and 5.10 show the result of this approach for the LVD scenarios and cut-in scenarios, respectively. In both cases, the empirical Wasserstein metric $\tilde{W}_1(\mathcal{Z}^*, \mathcal{W}^*)$ is minimal when the training data are directly used for the generated scenario parameters. Thus, the empirical Wasserstein metric suggests that the best approach for generating new scenario parameters is to simply sample parameters from the training data. The actual Wasserstein metric, estimated using $\tilde{W}_1\left(\mathcal{Z}^*_{\text{large}}, \mathcal{W}^*\right)$, shows that using our proposed method outperforms sampling parameters directly from the training data.

To justify the choice of $\beta = 0.25$, Figure 5.11 shows the correlation between the medians of the proposed metric $M_1(\mathcal{W}^*, \mathcal{Z}^*, \mathcal{X}^*)$ and $\tilde{W}_1\left(\mathcal{Z}^*_{\text{large}}, \mathcal{W}^*\right)$ for different values of β. With $\beta = 0$, i.e., $M_1(\mathcal{W}^*, \mathcal{Z}^*, \mathcal{X}^*) = \tilde{W}_1(\mathcal{Z}^*, \mathcal{W}^*)$, the correlation is 0.974 for the LVD scenarios and 0.824 for the cut-in scenarios. The correlation increases with increasing β until the maximum is obtained at $\beta \approx 0.21$ for the

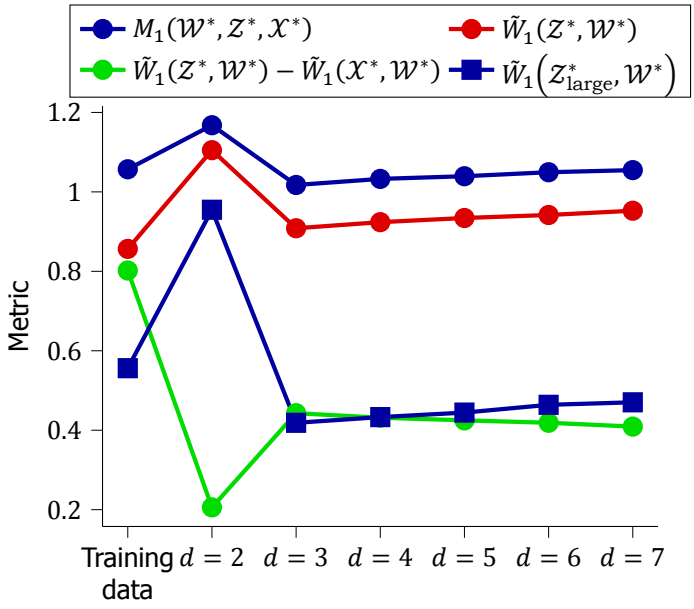


Figure 5.10: Medians of the metrics for the set of $N_w = 10000$ generated cut-in scenario parameter vectors.

LVD scenarios and at $\beta \approx 0.27$ for the cut-in scenarios. The correlations at these maxima are 0.992 and 0.987, respectively. Increasing β further results in a lower correlation, which suggests that a choice of $\beta = 0.25$ seems appropriate.

The experiment described in this section can be used to determine both d and β in an iterative manner given an initial choice for β (denoted by β_0):

1. Set $i = 0$.
2. Determine d_i such that $M_1(\mathcal{W}, \mathcal{Z}, \mathcal{X})$ with $\beta = \beta_i$ and $d = d_i$ is minimized.
3. Generate $\mathcal{X}^*$, $\mathcal{W}^*$, $\mathcal{Z}^*$, and $\mathcal{Z}^*_{\text{large}}$ using the approach described in this section with $d = d_i$.
4. Increase i by 1.
5. Determine β_i by maximizing the correlation between $M_1(\mathcal{W}^*, \mathcal{Z}^*, \mathcal{X}^*)$ with $\beta = \beta_i$ and $\tilde{W}_1\left(\mathcal{Z}^*_{\text{large}}, \mathcal{W}^*\right)$ (e.g., see Figure 5.11).
6. Repeat step 2.
7. Stop if $d_i = d_{i-1}$. Otherwise, return to step 3.

As an initial choice, $\beta_0 = 0.25$ seems appropriate. More specifically, when choosing $\beta_0 \in [0.1, 1]$, the optimal choice of d is found after one iteration.

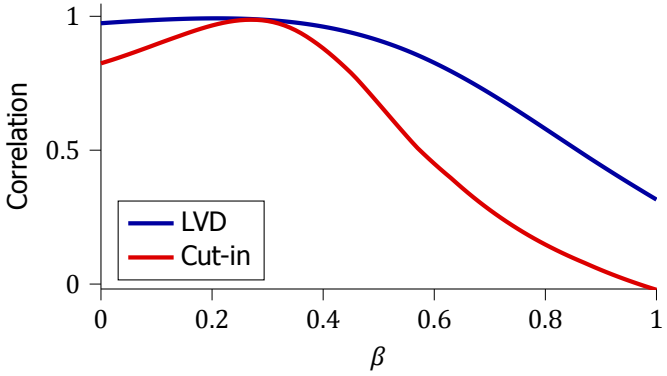


Figure 5.11: Correlation between the medians of $M_1(\mathcal{W}^*, \mathcal{Z}^*, \mathcal{X}^*)$ and $\tilde{W}_1\left(\mathcal{Z}^*_{\text{large}}, \mathcal{W}^*\right)$ for different values of β. The blue line shows the result for the LVD scenarios with a maximum correlation of 0.992 at $\beta \approx 0.21$. The red line shows the result for the cut-in scenarios with a maximum correlation of 0.987 at $\beta \approx 0.25$.

5

5.6. Discussion

One of the advantages of our proposed method for generating scenario parameters is that less assumptions are needed regarding the parameterization of the scenarios:

- There is no assumption needed on a predetermined functional form of the time series data. For example, in an LVD scenario, the speed is often assumed to follow a polynomial function [70], a sinusoidal function, or a linear function [277]. In case of a predetermined functional form, parameters are fitted to the functional form. In our case, the SVD automatically determines the optimal choice of parameterization without relying on a predetermined functional form.
- There is no assumption needed for the shape of the distribution of the parameters. For example, a particular distribution, such as a Gaussian distribution [113] or a uniform distribution, may be assumed for which parameters are fitted. Alternatively, assumptions are made regarding the independence of the parameters [102]. In our case, the KDE automatically adapts its shape to the data and also considers the dependence among the different parameters.

It should be noted, however, that if there is a reason to believe that one or more of the assumptions are valid, then alternative methods for generating scenario parameters that make use of such assumptions might perform equally or better than the presented method [261]. In most cases, it will be difficult to provide a proper justification of the assumptions regarding the functional form of, e.g., the vehicle speed, and the pdf of the scenario parameters and the presented method will outperform methods relying on such assumptions. In any case, the presented SR metric provides an opportunity to verify the applicability of any assumptions regarding the scenario parameterization and parameter distributions.

The generated scenario parameters represent scenarios that could happen in real life and cover the same variety that is found in real-world traffic. Most likely, the majority of these scenarios are straightforward for an AV to deal with. To do an efficient assessment, the focus should be on scenarios that might lead to critical situations in which the probability of a crash is high. That is why so-called *importance sampling* [239, Chapter 5.6] is often used for the assessment of AVs, e.g., see [70, 149, 315, 323]. With importance sampling, a different pdf, $g(\cdot)$, is used to sample scenario parameters, such that more emphasis is put on scenarios that might lead to critical situations. To get unbiased results, the result of a test with scenario parameters x is weighted by the ratio of the original probability density, $\hat{f}_H(x)$, and the probability density of the pdf used for importance sampling, $g(x)$ [70, 149, 239, 315]. Note that the importance sampling techniques explained in [70, 149, 315] can be directly applied on the estimated pdf $\hat{f}_H(\cdot)$ in (5.11) of the reduced set of parameters. In future work, our method for generating scenarios will be combined with importance sampling [70, 149, 315] for an assessment of an AV.

In some cases, one might want to sample from a conditional pdf, e.g., in case of sampling the scenario parameters for the LVD scenario such that the initial time gap equals a specified value. Sampling from a KDE such that one or more parameters are predetermined is straightforward [134]. In our case, sampling from $\hat{f}_h(\cdot)$ such that the time gap equals a specified value results in a linear constraint on the samples because the reduced parameter vector $\tilde{v}_i$ of (5.10) results from a linear mapping of the original parameters x_i of (5.2). In other words, one might want to sample v from $\hat{f}_H(\cdot)$ of (5.11), such that v is subject to the linear constraint

$$Av = b, \tag{5.25}$$

where A and b are a matrix and vector, respectively. In Chapter 6 [75], an algorithm is provided for sampling from a pdf estimated using KDE such that the generated sample is subject to the constraint of (5.25). The main idea of [75] is to weight each parameter vector v_i, $i \in \{1, \dots, N_x\}$ in the KDE based on how closely the v_i matches the constraint (5.25).

Our method for generating scenarios employs SVD to reduce the number of parameters. It must be noted that the parameters that result from the SVD are linear combinations of the original parameters. Therefore, the SVD only captures the linear relationships of the original parameters. To capture non-linear relationships, the original data can be first mapped using a non-linear kernel [249]. Choosing an appropriate kernel is, however, not straightforward and may involve much trial and error.

The presented case study considers a vehicle for which the full trajectory is predetermined. For the presented scenarios, this works well, but the full trajectory is not predetermined in scenarios where the actor's behavior depends on the behavior of the ego vehicle [12]. To deal with such scenarios, one option is to use a driver behavior model (e.g., [159, 281]) with predefined parameters instead of describing the full trajectory. The parameters of the driver behavior model may be part of θ. The proposed method for generating scenario parameter values still applies in these

kind of scenarios. Our ongoing research focuses on the assessment of AVs using scenarios in which driver behavior models are used for vehicles that may respond to the ego vehicle's behavior.

Since KDE is used, the generated scenario parameters represent variations of the data. Nevertheless, if the data do not contain scenarios that might lead to critical situations, such as an emergency braking maneuver or a reckless cut-in scenario, it is unlikely that such scenarios are generated, even if importance sampling [70, 149, 315] is used. Therefore, when using the generated scenarios for the (safety) assessment of AVs, it is important that there is enough data such that the data contain such scenarios. Although there is no consensus yet on the required amount of data, some metrics have been proposed (Chapter 3 [72] and [298]) for determining whether enough data have been collected when using the data for the assessment of AVs.

This work employs the Wasserstein metric to propose the SR metric for evaluating the generated scenario parameters. It is illustrated how our proposed metric could be used to determine the appropriate number of parameters (d) and the type of distribution that is used to model the pdf of the scenario parameters. Also, the bandwidth h or bandwidth matrix H could also be determined by optimizing the proposed metric. In case of the bandwidth estimation, the disadvantage is that it would require more computational resources compared to, e.g., leave-one-out cross-validation.

More research is needed to determine the influences on the optimal choice for the penalty weight β. The case study has demonstrated one way to verify whether the initial choice of β is appropriate, but we do not yet know *why* a weight of $\beta \approx 0.25$ is an appropriate choice. The actual choice might depend on, among others, N_x, N_z, N_w, and the shape of the underlying distribution of the scenario parameters. Future research with a larger data set will allow us to better determine the optimal β and how this optimal value is influenced.

Future work involves researching the use of the proposed metric in combination with alternative methods for generating scenarios for the assessment of AVs. For example, Spooner *et al.* [269] have used a GAN [117] to create pedestrian crossing scenarios. One of the difficulties with GANs is to know when the GAN truly replicates the underlying distribution. Several metrics have been proposed [36] to evaluate the performance of GANs, among which a metric based on the Wasserstein metric that compares the generated data with test data. Alternatively, our proposed metric, which also considers the training data, could be considered for evaluating GANs. To judge the potential of our proposed metric in this application, more research is needed.

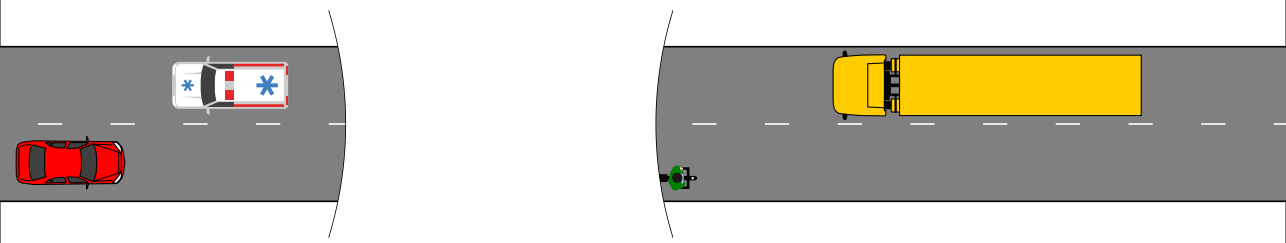
5.7. Conclusions

It is essential for the deployment of Automated Vehicles (AVs) to develop assessment methods. Scenario-based assessment in which test cases are derived from real-world road traffic scenarios is regarded as a viable approach for assessing AVs. This work has presented a method to generate parameterized scenarios for the use in test case descriptions for the assessment of AVs. To not rely on a small set of

parameters, we have used Singular Value Decomposition (SVD) to reduce the parameters. Parameter values for the scenarios are generated by drawing samples from the estimated probability density function (pdf) of the reduced set of parameters. To deal with the unknown shape of the pdf, it has been proposed to estimate the pdf using Kernel Density Estimation (KDE). This work has also presented a novel metric, the so-called Scenario Representativeness (SR) metric, based on the Wasserstein metric, for evaluating whether the generated scenario parameters represent realistic scenarios while covering the same variety that is found in real-world traffic.

A case study has illustrated the proposed method for generating scenario parameter values using scenarios with a leading vehicle that decelerates and scenarios with a vehicle that performs a cut-in. The case study has also illustrated that the proposed SR metric correctly quantifies the degree to which the generated scenario parameter values represent real-world scenarios and, at the same time, cover the same variety of scenarios that is found in real life.

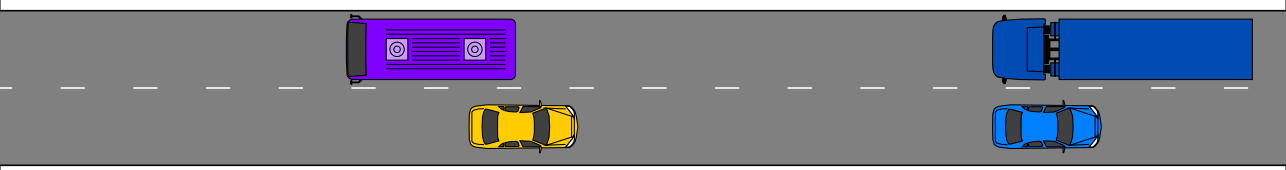
Future work involves applying the proposed method for more complex scenarios, e.g., scenarios that contain several different actors, to generate scenario-based test cases for the safety assessment of AVs. Additionally, it would be of interest to apply importance sampling for AV assessment in combination with the proposed method for generating scenarios. Other future work involves investigating the use of the proposed metric in combination with alternative methods for generating scenarios for the assessment of AVs.



6

Constrained sampling to generate scenario parameters

This chapter is based on:
E. de Gelder, E. Cator, J.-P. Paardekooper, O. Op den Camp, and B. De Schutter, *Constrained sampling from a kernel density estimator to generate scenarios for the assessment of automated vehicles,* in *IEEE Intelligent Vehicles Symposium Workshops (IV Workshop)* (2021) pp. 203–208.



The safety assessment of Automated Vehicles (AVs) is an important aspect of the development cycle of AVs. A scenario-based assessment approach is accepted by many players in the field as part of the complete safety assessment. A scenario is a representation of a situation on the road to which the AV needs to respond appropriately. One way to generate the required scenario-based test descriptions is to parameterize the scenarios and to draw these parameters from a probability density function (pdf). Because the shape of the pdf is unknown beforehand, assuming a functional form of the pdf and fitting the parameters to the data may lead to inaccurate fits. As an alternative, Kernel Density Estimation (KDE) is a promising candidate for estimating the underlying pdf because it is flexible with the underlying distribution of the parameters. Drawing random samples from a pdf estimated with KDE is possible without the need of evaluating the actual pdf, which makes it suitable for drawing random samples for, e.g., Monte Carlo methods. Sampling from a KDE while the samples satisfy a linear equality constraint, however, has not been described in the literature, as far as the authors know.

In this chapter, we propose a method to sample from a pdf estimated using KDE, such that the samples satisfy a linear equality constraint. We also present an algorithm of our method in pseudo-code. The method can be used for generating scenarios that have, e.g., a predetermined starting speed or for generating different types of scenarios. This chapter also shows that the method for sampling scenarios can be used in case a Singular Value Decomposition (SVD) is used to reduce the dimension of the parameter vectors.

6.1. Introduction

An essential facet in the development of Automated Vehicles (AVs) is the assessment of quality and performance aspects of the AVs, such as safety, comfort, and efficiency [27, 167, 270]. Because public road tests are expensive and time consuming [155, 323], a scenario-based approach has been proposed [15, 70, 94, 229, 233, 270]. As a source of information for the scenarios for the assessment, real-world driving data has been proposed, such that the scenarios relate to real-world driving conditions [94, 171, 229].

When using scenarios extracted from real-world driving data as a direct source for describing scenario-based tests, two problems arise. First, not all possible variations of the scenarios might be found in the data. Therefore, the failure modes of the AVs might not be reflected in the tests that are based on the scenarios that are extracted from real-world driving data [323]. Second, using scenarios extracted from real-world driving data might not reduce the actual testing load because the set of extracted scenarios is largely composed of non-safety critical scenarios [323]. As a solution to this, so-called importance sampling has been introduced in order to put more emphasis on scenarios that are likely to trigger safety-critical situations [70, 149, 315, 323]. These methods [70, 149, 315, 323] have in common that they describe scenarios using parameters for which a probability density function (pdf) is estimated.

As already presented in [70, 71], we propose to estimate the pdf using Kernel Density Estimation (KDE). KDE [222, 237] is often referred to as a non-parametric way to estimate the pdf because no use is made of a predefined functional form of the pdf for which certain parameters are fitted to the data. Because KDE produces a pdf that adapts itself to the data, it is flexible regarding the shape of the actual underlying distribution of the parameters.

Sampling from a KDE is straightforward. In some cases, however, one wants to sample from the estimated pdf while a part of the random sample is fixed. For example, one may want to assess the performance of an AV when it is driving at its maximum allowable speed. Conditional sampling allows to generate scenario-based test cases in which, e.g., the ego vehicle has a fixed speed. One approach to performing conditional sampling is to evaluate the conditional pdf and to use this for sampling. This method, however, would be highly cumbersome, especially with a higher-dimensional pdf, because marginal integrals of codimension 1 of the conditional pdf must be evaluated.

We will propose an algorithm to sample parameters from a KDE while the parameters are subject to a linear equality constraint. Our work differs from [128, 308] because these works consider (shape) constraints on the estimated pdf. The proposed algorithm can be regarded as a generalization of the conditional density estimation in [134]. Our proposed method is efficient because the actual (conditional) pdf does not need to be evaluated. We illustrate the proposed sampling technique and its practical usefulness using an example. Furthermore, we will explain the usefulness of sampling with linear equality constraints in case parameter reduction techniques are used to avoid the curse of dimensionality that pdf estimation techniques, such as KDE, are suffering from.

This chapter is organized as follows. In Section 6.2, we first describe the problem in more detail. The proposed method is presented in Section 6.3. Through an example, we illustrate the correct performance of the algorithm in Section 6.4. We also apply the proposed method to sample different types of scenarios in Section 6.4. In Section 6.5, some implications and limitations of this work are discussed. Conclusions of this chapter are provided in Section 6.6.

6.2. Problem definition

With KDE, the pdf $f(\cdot)$ is estimated as follows:

$$\hat{f}(x) = \frac{1}{N}\sum_{i=1}^{N} K_H(x - x_i). \tag{6.1}$$

Here, $x_i \in \mathbb{R}^d$ represents the i-th data point of dimension d. In total, there are N data points, so $i \in \{1, \ldots, N\}$. In (6.1), $K_H(\cdot)$ is the so-called scaled kernel with a positive definite symmetric bandwidth matrix $H \in \mathbb{R}^{d\times d}$. The kernel $K(\cdot)$ and the scaled kernel $K_H(\cdot)$ are related using

$$K_H(u) = |H|^{-1/2} K\left(H^{-1/2}u\right), \tag{6.2}$$

where $|\cdot|$ denotes the matrix determinant. The choice of the kernel function is not as important as the choice of the bandwidth matrix [87, 283]. Often, a Gaussian kernel is opted and this chapter is no exception. The Gaussian kernel is given by

$$K(u) = \frac{1}{(2\pi)^{d/2}} \exp\left\{-\frac{1}{2}\|u\|_2^2\right\}, \tag{6.3}$$

where $\|u\|_2^2 = u^{\mathsf{T}}u$ denotes the squared 2-norm of u. Substituting (6.2) and (6.3) into (6.1) gives

$$\hat{f}(x) = \frac{1}{N(2\pi)^{d/2}|H|^{1/2}} \sum_{i=1}^{N} \exp\left\{-\frac{1}{2}(x - x_i)^{\mathsf{T}} H^{-1}(x - x_i)\right\}. \tag{6.4}$$

The bandwidth matrix is an important parameter of the KDE. Several methods have been proposed to estimate the bandwidth matrix based on the available data. Perhaps the simplest method is Silverman's rule of thumb [262], in which $H = h^2 I_d$ with I_d denoting the d-by-d identity matrix[1] and

$$h = 1.06 \min\left(\sigma, \frac{R}{1.34}\right) N^{-\frac{1}{5}}. \tag{6.5}$$

[1] If $H = h^2 I_d$, the same smoothing is applied in every direction. Therefore, the data are often normalized before applying KDE with $H = h^2 I_d$.

Here, σ denotes the standard deviation of the data and R is the interquartile range of the data. Another strategy is to use cross validation. As a special case, with one-leave-out cross validation, the bandwidth matrix equals

$$\arg\max_{H} \prod_{i=1}^{N} \left(\frac{1}{N-1} \sum_{j=1, j\neq i}^{N} K_H(x_i - x_j) \right). \tag{6.6}$$

Selecting the bandwidth matrix by this method minimizes the Kullback-Leibler divergence between $f(\cdot)$ and $\hat{f}(\cdot)$ [283]. Another often-used strategy is to use plug-in methods. The idea of plug-in methods is to select an initial H and then plug $\hat{f}(\cdot)$ into an equation that calculates the optimal[2] bandwidth based on a given pdf. This process is then iterated until H converges. We refer the interested reader to [87, 120, 151, 283] for details on the estimation of H. In this chapter, we assume that H is given.

Sampling new data points from $\hat{f}(\cdot)$ of (6.4) is straightforward. First, an integer $j \in \{1, \ldots, N\}$ is randomly chosen with each integer having equal likelihood. Next, a random sample is drawn from a Gaussian with covariance H and mean x_j.

In this chapter, we want to sample from (6.4) such that the samples satisfy the linear equality constraint:

$$Ax = b. \tag{6.7}$$

Here $A \in \mathbb{R}^{d_c \times d}$ and $b \in \mathbb{R}^{d_c}$ denote the constraint matrix and the constraint vector, respectively. It is assumed that the constraint matrix A has full rank. Note that if A has not full rank, the constraint of (6.7) can easily be reformulated using Gaussian elimination, resulting in a similar constraint with a constraint matrix that has full rank. In total, there are $d_c < d$ constraints.

6.3. Method

To deal with the constraint (6.7), we will perform a rotation of x such that a part of the resulting vector is fixed by the constraint (6.7), while the other part of the resulting vector can be freely chosen. To perform the rotation, we employ a Singular Value Decomposition (SVD) [116] of A:

$$A = U \begin{bmatrix} \Sigma & 0 \end{bmatrix} V^T = U \begin{bmatrix} \Sigma & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U \Sigma V_1^T. \tag{6.8}$$

Here, $U \in \mathbb{R}^{d_c \times d_c}$ and $V \in \mathbb{R}^{d \times d}$ are orthonormal matrices, i.e., $U^{-1} = U^T$ and $V^{-1} = V^T$. The first d_c columns of V are denoted by V_1 while V_2 denotes the remaining $d_u = d - d_c$ columns of V. Moreover, $\Sigma \in \mathbb{R}^{d_c \times d_c}$ is a diagonal matrix with its so-called singular values on its diagonal. Because A has full rank and $d_c < d$, all singular values are strictly positive. As such, evaluating Σ^{-1} is straightforward. Now, let $\bar{x} \in \mathbb{R}^{d_c}$ and $\tilde{x} \in \mathbb{R}^{d_u}$ such that

$$x = V_1 \bar{x} + V_2 \tilde{x} = \begin{bmatrix} V_1 & V_2 \end{bmatrix} \begin{bmatrix} \bar{x} \\ \tilde{x} \end{bmatrix} = V \begin{bmatrix} \bar{x} \\ \tilde{x} \end{bmatrix}. \tag{6.9}$$

[2] Optimal in the sense that it minimizes a specific value, which is usually the asymptotic mean integrated squared error.

Note that because $V^{-1} = V^{\mathsf{T}}$, we have $\bar{x} = V_1^{\mathsf{T}} x$ and $\tilde{x} = V_2^{\mathsf{T}} x$. Moreover, $V_1^{\mathsf{T}} V_1 = I_{d_\mathrm{c}}$ and $V_1^{\mathsf{T}} V_2 = 0$, such that substituting (6.8) and (6.9) into (6.7), gives

$$U\Sigma V_1^{\mathsf{T}}(V_1\bar{x} + V_2\tilde{x}) = U\Sigma\bar{x} = b. \tag{6.10}$$

This means that in order to satisfy the constraint (6.7), $\tilde{x}$ can take any value whereas $\bar{x}$ is fixed:

$$\bar{x} = \Sigma^{-1} U^{\mathsf{T}} b. \tag{6.11}$$

Similar as $\bar{x}$ and $\tilde{x}$, let $\bar{x}_i = V_1^{\mathsf{T}} x_i$ and $\tilde{x}_i = V_2^{\mathsf{T}} x_i$. Using this and (6.9), we can rewrite (6.4):

$$\hat{f}(x) = \frac{1}{N(2\pi)^{d/2}|H|^{1/2}} \sum_{i=1}^{N} \exp\left\{-\frac{1}{2}\begin{bmatrix}\bar{x}-\bar{x}_i\\ \tilde{x}-\tilde{x}_i\end{bmatrix}^{\mathsf{T}} V^{\mathsf{T}} H^{-1} V \begin{bmatrix}\bar{x}-\bar{x}_i\\ \tilde{x}-\tilde{x}_i\end{bmatrix}\right\}. \tag{6.12}$$

To ease the notation, let us use the following notation:

$$V^{\mathsf{T}} H^{-1} V = \begin{bmatrix}\Lambda_{11} & \Lambda_{12}\\ \Lambda_{21} & \Lambda_{22}\end{bmatrix}, \tag{6.13}$$

with $\Lambda_{11} \in \mathbb{R}^{d_\mathrm{c}\times d_\mathrm{c}}$, $\Lambda_{12} \in \mathbb{R}^{d_\mathrm{c}\times d_\mathrm{u}}$, $\Lambda_{21} \in \mathbb{R}^{d_\mathrm{u}\times d_\mathrm{c}}$, and $\Lambda_{22} \in \mathbb{R}^{d_\mathrm{u}\times d_\mathrm{c}}$. Using the Schur complement [320]

$$\Lambda_\mathrm{S} = \Lambda_{11} - \Lambda_{12}\Lambda_{22}^{-1}\Lambda_{21}, \tag{6.14}$$

we can write (6.14) as

$$V^{\mathsf{T}} H^{-1} V = \begin{bmatrix} I_{d_\mathrm{c}} & \Lambda_{12}\Lambda_{22}^{-1}\\ 0 & I_{d_\mathrm{u}}\end{bmatrix}\begin{bmatrix}\Lambda_\mathrm{S} & 0\\ 0 & \Lambda_{22}\end{bmatrix}\begin{bmatrix} I_{d_\mathrm{c}} & 0\\ \Lambda_{22}^{-1}\Lambda_{21} & I_{d_\mathrm{u}}\end{bmatrix}. \tag{6.15}$$

Substituting this in the exponent of (6.12) gives

$$\begin{aligned}
&\begin{bmatrix}\bar{x}-\bar{x}_i\\ \tilde{x}-\tilde{x}_i\end{bmatrix}^{\mathsf{T}} H^{-1} \begin{bmatrix}\bar{x}-\bar{x}_i\\ \tilde{x}-\tilde{x}_i\end{bmatrix}\\
&= \begin{bmatrix}\bar{x}-\bar{x}_i\\ \tilde{x}-\tilde{x}_i + \Lambda_{22}^{-1}\Lambda_{12}^{\mathsf{T}}(\bar{x}-\bar{x}_i)\end{bmatrix}^{\mathsf{T}}\begin{bmatrix}\Lambda_\mathrm{S} & 0\\ 0 & \Lambda_{22}\end{bmatrix}\begin{bmatrix}\bar{x}-\bar{x}_j\\ \tilde{x}-\tilde{x}_j + \Lambda_{22}^{-1}\Lambda_{21}(\bar{x}-\bar{x}_j)\end{bmatrix} \qquad (6.16)\\
&= (\bar{x}-\bar{x}_i)^{\mathsf{T}}\Lambda_\mathrm{S}(\bar{x}-\bar{x}_i) +\\
&\qquad (\tilde{x}-\tilde{x}_i + \Lambda_{22}^{-1}\Lambda_{12}^{\mathsf{T}}(\bar{x}-\bar{x}_i))^{\mathsf{T}}\Lambda_{22}(\tilde{x}-\tilde{x}_i + \Lambda_{22}^{-1}\Lambda_{12}^{\mathsf{T}}(\bar{x}-\bar{x}_i)). \qquad (6.17)
\end{aligned}$$

Using this, (6.12) can be written as

$$\hat{f}(x) = \frac{1}{N(2\pi)^{d/2}|H|^{1/2}} \sum_{i=1}^{N} w_i \exp\left\{-\frac{1}{2}(\tilde{x}-\tilde{x}_i')^{\mathsf{T}}\Lambda_{22}(\tilde{x}-\tilde{x}_i')\right\}, \tag{6.18}$$

Algorithm 6.1: Sampling with linear equality constraints and full bandwidth matrix.

Input : $x_1, \dots, x_N$, A, b, H
Output: Sample x from (6.4) while satisfying $Ax = b$

1 U, Σ, V_1, $V_2 \leftarrow$ Perform an SVD of A; see (6.8)
2 $\bar{x}_1, \dots, \bar{x}_N \leftarrow$ Map the data points using $\bar{x}_i = V_1^\mathsf{T} x_i$
3 $\tilde{x}_1, \dots, \tilde{x}_N \leftarrow$ Map the data points using $\tilde{x}_i = V_2^\mathsf{T} x_i$
4 $\bar{x} \leftarrow$ Compute $\bar{x}$ using (6.11)
5 Λ_{11}, Λ_{12}, Λ_{21}, Λ_{22}, $\Lambda_\mathrm{S} \leftarrow$ Compute $V^\mathsf{T} H^{-1} V$ according to (6.13) and Λ_S according to (6.14)
6 $w_1, \dots, w_N \leftarrow$ Compute the weights according to (6.19)
7 $j \leftarrow$ Generate a random integer $j \in \{1, \dots, N\}$ with the likelihood of j proportional to w_j
8 $\tilde{x}_j' \leftarrow$ Compute the mean of the Gaussian to generate a sample from according to (6.20)
9 $\tilde{x} \leftarrow$ Generate a random sample from a Gaussian with covariance Λ_{22}^{-1} and mean $\tilde{x}_j'$
10 $x \leftarrow$ Compute x according to (6.9)

with

$$w_i = \exp\left\{-\frac{1}{2}(\bar{x} - \bar{x}_i)^\mathsf{T} \Lambda_\mathrm{S} (\bar{x} - \bar{x}_i)\right\}, \ \forall i \in \{1, \dots, N\}, \tag{6.19}$$

$$\tilde{x}_i' = \tilde{x}_i - \Lambda_{22}^{-1} \Lambda_{21} (\bar{x} - \bar{x}_i), \ \forall i \in \{1, \dots, N\}. \tag{6.20}$$

To generate samples from (6.4) that satisfy (6.7), two random numbers need to be generated. First, an integer $j \in \{1, \dots, N\}$ is randomly chosen with the likelihood of the integer j proportional to the weight w_j of (6.19). Next, a random sample is drawn from a Gaussian with covariance Λ_{22}^{-1} and mean $\tilde{x}_j'$. Finally, this random sample is mapped according to (6.9) to obtain the final random sample. The procedure for sampling is summarized in Algorithm 6.1.

6.4. Example

To illustrate the proposed method, we have applied it to generate a new set of parameters that describe scenarios. The Next Generation SIMulation (NGSIM) data set is used as a data source. The NGSIM data set contains vehicles' trajectories obtained from video footage of cameras that were located at several motorways in the US [170]. In total, 18182 longitudinal interactions between two vehicles are analyzed. In each of these longitudinal interactions, we look at the speed profile of the leading vehicle. The speed profile is split into parts of $5\,\mathrm{s}$, resulting in $N = 99840$ data samples.

We first apply the method of conditional sampling in a straightforward example to illustrate that Algorithm 6.1 produces correct results. The second example

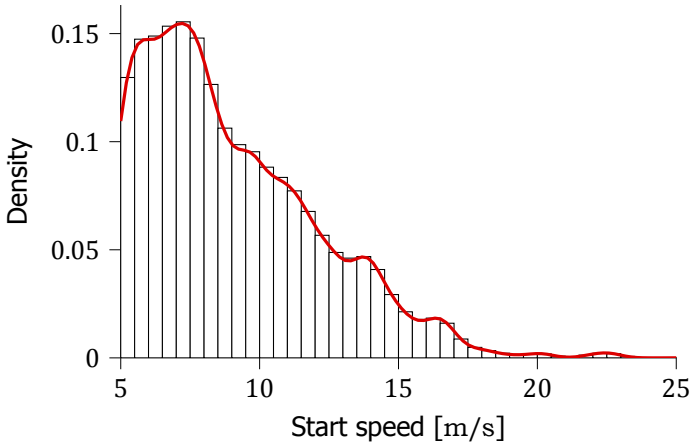


Figure 6.1: The histogram shows the result of 10^6 samples obtained by the conditional sampling according to Algorithm 6.1. The red line represents the true pdf.

explains the usefulness of sampling with a linear equality constraint in case a parameter reduction technique is used.

6.4.1. Sampling with a linear equality constraint

In this first example, two parameters describe a scenario:

1. The speed of the leading vehicle at a certain time t and
2. The speed of the same vehicle at time $t + \Delta t$ with $\Delta t = 5\,\mathrm{s}$.

The bandwidth matrix is estimated using the plug-in selector of Wand and Jones [294]. We want to sample the initial speed in case the speed is reduced by $5\,\mathrm{m/s}$. To achieve this, we use

$$A = \begin{bmatrix} 1 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 5 \end{bmatrix}. \tag{6.21}$$

In Figure 6.1, the result of Algorithm 6.1 is shown. In total, 10^6 samples are generated and shown by the histogram. Because the histogram follows the same pattern as the actual density of the speed difference according to the KDE, it illustrates that the provided algorithm correctly samples from the KDE.

6.4.2. Applying conditional sampling with parameter reduction

Typically, more than two parameters are needed to describe a scenario. If the number of parameters is too high, however, the pdf estimation using the KDE suffers from the curse of dimensionality [254]. To avoid the curse of dimensionality, a reduction of parameters can be obtained. In this example, we use an SVD to reduce the number of parameters.

Instead of using the speed at only two time instances, we consider the following scenario parameters:

$$x_i = \begin{bmatrix} v(t_i) \\ v(t_i + \Delta t) \\ \vdots \\ v(t_i + n_\mathrm{t}\Delta t) \end{bmatrix} \in \mathbb{R}^{n_\mathrm{t}+1}, \tag{6.22}$$

where $v(\cdot)$ denotes the speed of the leading vehicle, t_i denotes the time of the i-th data point, Δt denotes the time step, and n_t denote the number of time steps. We use $\Delta t = 0.1\,\mathrm{s}$ and $n_\mathrm{t} = 50$, so this results in 51 parameters. To reduce these 51 parameters to d parameters, consider the following matrix:

$$X = \begin{bmatrix} x_1 - \mu & \cdots & x_N - \mu \end{bmatrix} \in \mathbb{R}^{(n_\mathrm{t}+1)\times N}, \tag{6.23}$$

with $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$ and the following SVD of this matrix:

$$X = \begin{bmatrix} \bar{U}_1 & \bar{U}_2 \end{bmatrix} \begin{bmatrix} \bar{\Sigma}_1 & 0 \\ 0 & \bar{\Sigma}_2 \end{bmatrix} \begin{bmatrix} \bar{V}_1^\mathsf{T} \\ \bar{V}_2^\mathsf{T} \end{bmatrix} \approx \bar{U}_1 \bar{\Sigma}_1 \bar{V}_1^\mathsf{T}, \tag{6.24}$$

with $\bar{U}_1 \in \mathbb{R}^{(n_\mathrm{t}+1)\times d}$, $\bar{U}_2 \in \mathbb{R}^{(n_\mathrm{t}+1)\times(n_\mathrm{t}+1-d)}$, $\bar{\Sigma}_1 \in \mathbb{R}^{d\times d}$, $\bar{\Sigma}_2 \in \mathbb{R}^{(n_\mathrm{t}+1-d)\times(N-d)}$, $\bar{V}_1 \in \mathbb{R}^{N\times d}$, and $\bar{V}_2 \in \mathbb{R}^{N\times(N-d)}$. Using this approximation, it follows that

$$x_i \approx \bar{U}_1 \bar{\Sigma}_1 \bar{v}_i + \mu, \ \forall i \in \{1, \ldots, N\}, \tag{6.25}$$

where $\bar{v}_i \in \mathbb{R}^d$ is the i-th column of $\bar{V}_1^\mathsf{T}$.

Instead of using the original data points x_i, the vectors $\bar{v}_i$ are used for the estimation of the KDE. For each sample of this KDE, the mapping of (6.25) is then applied to obtain the scenario parameters according to (6.22). In this example, $d = 4$ is used. As with the first example, the bandwidth matrix is estimated using the plug-in selector of Wand and Jones [294].

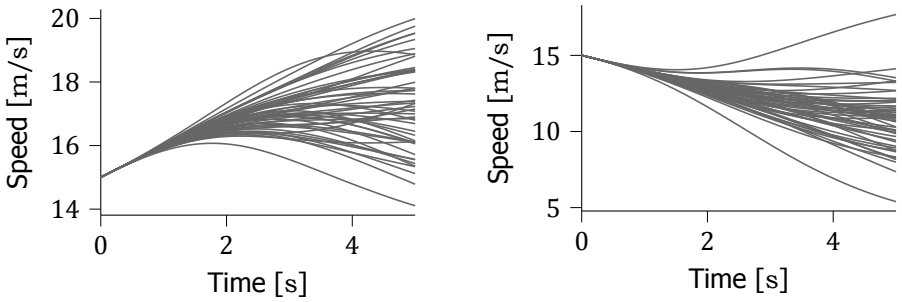
In Figure 6.2, 50 scenarios are sampled from the KDE in which the initial speed v_init and initial acceleration a_init are fixed by a linear equality constraint. To achieve this, the following constraint matrix and constraint vector are used:

$$A = \begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{bmatrix} \bar{\Sigma}_1, \quad b = \begin{bmatrix} v_\mathrm{init} - \mu_1 \\ v_\mathrm{init} + \Delta t \cdot a_\mathrm{init} - \mu_2 \end{bmatrix}, \tag{6.26}$$

where $\bar{u}_1$ and $\bar{u}_2$ denote the first and second row of $\bar{U}_1$, respectively, and μ_1 and μ_2 denote the first and second entry of μ, respectively. Figure 6.2a shows the result with $v_\mathrm{init} = 15\,\mathrm{m/s}$ and $a_\mathrm{init} = 1\,\mathrm{m/s^2}$. In Figure 6.2b, $a_\mathrm{init} = -1\,\mathrm{m/s^2}$ is used instead.

In Figure 6.3, 50 scenarios are sampled from the KDE in which the initial speed v_init and end speed v_end are fixed by a linear equality constraint. To achieve this, the following constraint matrix and constraint vector are used:

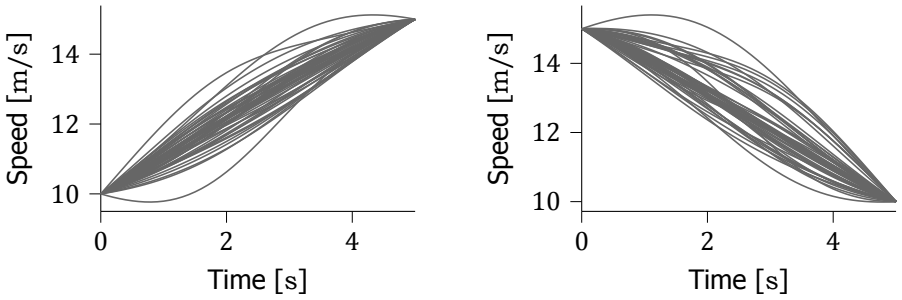
$$A = \begin{bmatrix} \bar{u}_1 \\ \bar{u}_{n_\mathrm{t}+1} \end{bmatrix} \bar{\Sigma}_1, \quad b = \begin{bmatrix} v_\mathrm{init} - \mu_1 \\ v_\mathrm{end} - \mu_{n_\mathrm{t}+1} \end{bmatrix}. \tag{6.27}$$



(a) $v_{\text{init}} = 15\,\text{m/s}$ and $a_{\text{init}} = 1\,\text{m/s}^2$.

(b) $v_{\text{init}} = 15\,\text{m/s}$ and $a_{\text{init}} = -1\,\text{m/s}^2$.

Figure 6.2: 50 scenarios sampled from the KDE with a constraint on the initial speed and the initial acceleration.



(a) $v_{\text{init}} = 10\,\text{m/s}$ and $v_{\text{end}} = 15\,\text{m/s}$.

(b) $v_{\text{init}} = 15\,\text{m/s}$ and $v_{\text{end}} = 10\,\text{m/s}$.

Figure 6.3: 50 scenarios sampled from the KDE with a constraint on the initial speed and the end speed.

This illustrates that the conditional sampling can be used to generate scenarios in which a leading vehicle is accelerating (Figure 6.3a) or decelerating (Figure 6.3b). Figure 6.3a shows the result with $v_{\text{init}} = 10\,\text{m/s}$ and $v_{\text{end}} = 15\,\text{m/s}$. In Figure 6.3b, the start and end speed are opposite: $v_{\text{init}} = 15\,\text{m/s}$ and $v_{\text{end}} = 10\,\text{m/s}$.

Note that all speed profiles in Figures 6.2 and 6.3 are all drawn from the same KDE. The only difference between these scenarios is that different conditions are used.

6.5. Discussion

This chapter has provided a method for sampling from a KDE such that the samples satisfy a linear equality constraint. The provided algorithm calculates weights that are used to weigh the contribution of each input data point to the overall pdf. Samples are drawn by randomly picking a random data point with the likelihood proportional to the calculated weights and adding an offset to this data point using a zero-mean multivariate Gaussian with a covariance matrix that equals the original bandwidth matrix (pre)multiplied with a rotation matrix.

The provided algorithm can be regarded as a generalization of the conditional

sampling in [134]. With conditional sampling, few parameters are fixed. This is the same as considering the linear equality constraint of (6.7) with

$$A = \begin{bmatrix} I_{d_c} & 0 \end{bmatrix}. \tag{6.28}$$

In this particular case, the rotation of the data is not needed, so steps 2, 3, 4, and 10 of Algorithm 6.1 can be skipped. Note that this way of sampling becomes impractical if a parameter reduction is used as shown in Section 6.4.2 because the set of reduced parameters have no physical meaning anymore.

The computational cost of the provided algorithm scales quadratically with respect to number of parameters that are used to describe a single scenario (d) and linearly with the number of data points (N). Because the number of data points is generally much larger than the dimension of the data points, let us assume that $N \gg d$. Looking at Algorithm 6.1 and considering $N \gg d$, steps 2, 3, and 6 are the most time consuming because these steps contain a loop over the data points. It is easy to see that the number of computations of these steps scales linearly with N. Since these computations contain a multiplication of a d-by-d matrix and a vector with d rows, the computational cost scales quadratically with d.

If we want to sample multiple times using the same linear constraint, it suffices to perform steps 1 till 6 of Algorithm 6.1 only once. Step 7 of Algorithm 6.1 does not depend on d and scales linearly with N [290]. Steps 8 till 10 of Algorithm 6.1 do not depend on N and scale quadratically with d. Because these steps do not depend on N and $N \gg d$, the computational cost of these steps is minor compared to step 7 of Algorithm 6.1. Therefore, if we want to draw many samples, i.e., more than N, the computational cost is dominated by the computational cost of step 7, which means that, in that case, the computational cost scales linearly with N.

Another application of the conditional sampling is to predict how the future will develop based on some initial conditions. For example, Figures 6.2a and 6.2b each show 50 possibilities for the speed of the leading vehicle in the next $5\,\mathrm{s}$ given an initial speed and an initial acceleration. This could be used, for instance, to determine real-time a worst-case scenario, such that an AV could proactively respond to such a scenario. Similarly, when using Bayesian networks for predicting continuous variables [30], our algorithm provides a way to sample from the conditional densities.

Note that for an efficient scenario-based assessment of an AV, scenarios that might lead to critical behavior need to be emphasized. Several techniques are proposed in the literature to emphasize these scenarios, such as reachability analysis [11], boundary searching [325], and importance sampling [70, 149, 315, 323]. With importance sampling, a different pdf is used to sample scenario parameters, such that more emphasis is put on scenarios that might lead to critical behavior. Our proposed method for conditional sampling can be combined with the importance sampling techniques explained in [70, 149, 315].

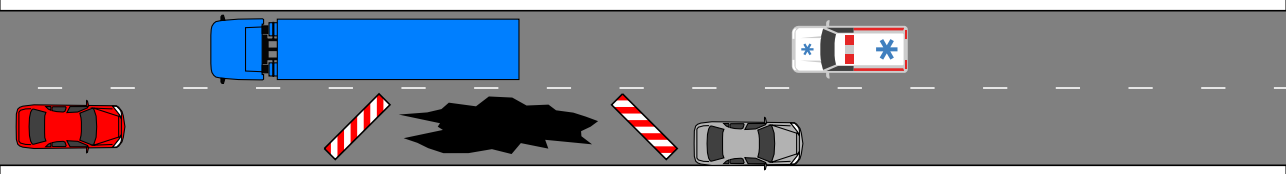
This work comes with limitations. It should be emphasized that the method only works when using a Gaussian kernel for the KDE. In practice, this is usually not a problem because the choice of the kernel is not crucial [87]. Another limitation is that our method cannot be extended to deal with (linear) inequality constraints. If

these inequality constraints are not too severe, however, straightforward rejection sampling could be used in that case, i.e., sample data points until a data point satisfies the linear inequality constraint. It should also be noted that in practice, more parameters for describing a scenario might need to be considered. For example, the state of neighboring vehicles (instead of only the leading vehicle), lane curvature, etc. Although these parameters are not considered in the example of this chapter, a parameter reduction technique as explained in Section 6.4.2 can still be useful in case these parameters are considered.

6.6. Conclusions

It is expected that scenario-based test descriptions become more and more important for the assessment of Automated Vehicles (AVs). One way to generate these scenario-based test descriptions is to sample the scenario parameters from a probability density function (pdf). To deal with the unknown shape of the pdf, it is proposed to estimate the pdf using Kernel Density Estimation (KDE). In this chapter, we have shown how these parameters can be drawn from the estimated pdf while these parameters are subject to a linear equality constraint. Through an example, we have illustrated the effectiveness of our method by generating different scenarios of a longitudinal interaction with a leading vehicle.

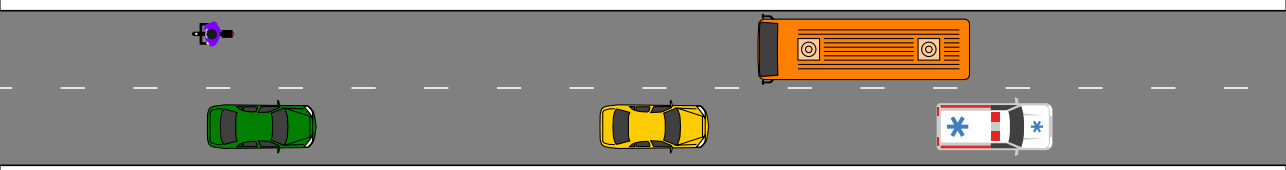
Future work involves applying this method using more complex scenarios, e.g., scenarios that contain several different actors, to generate scenario-based test cases for the safety assessment of AVs.



7

Risk quantification in driving scenarios

This chapter is based on:
E. de Gelder, H. Elrofai, A. Khabbaz Saberi, O. Op den Camp, J.-P. Paardekooper, and B. De Schutter, *Risk quantification for automated driving systems in real-world driving scenarios,* IEEE Access **9**, 168953 (2021).



The development of safety validation methods is essential for the safe deployment and operation of Automated Driving Systems (ADSs). One of the goals of safety validation is to prospectively evaluate the risk of an ADS dealing with real-world traffic. ISO 26262 and ISO 21448, the leading standards in automotive safety, provide an approach to estimate the risk where the former focuses on risks due to potential malfunctioning of components and the latter focuses on risks due to possible functional insufficiencies. The main shortcomings of the approach provided in ISO 26262 are that it depends on subjective judgments of safety experts and that only a qualitative risk estimation is performed. ISO 21448 addresses these shortcomings partially by providing statistical methods to guide the safety validation, but no complete method is provided to quantify the risk.

The first objective of this chapter is to propose a method to estimate the risk of an ADS in a more quantitative and objective manner. A data-driven approach is used to rely less on subjective judgments of safety experts. The output of the method is the expected number of injuries in a potential crash. Thus, the method is quantitative, the result is easily interpretable, and the result can be compared with road crash statistics. The second objective is to provide a method that supports the risk assessment as stipulated by the ISO 26262 and ISO 21448 standards by decomposing the quantified risk into the three aspects of risk as mentioned in these standards: exposure, severity, and controllability. The proposed methods are illustrated by means of a case study in which the risk is quantified for a longitudinal controller in three different types of scenarios. The code of the case study is publicly available.

7.1. Introduction

It is expected that Automated Driving Systems (ADSs) will make traffic safer by eliminating human errors, enable more comfortable rides, and reduce traffic congestion [48]. Lower levels of automation systems, such as Adaptive Cruise Control (ACC) [191] and lane keeping assist systems [193], are already widely deployed in modern cars and trucks. Since the development of ADSs has made significant progress, it is expected that ADSs addressing higher levels of automation and covering the full dynamic driving task, i.e., SAE level 3 or higher [243], are soon to be introduced on public roads [29, 190, 198]. Before deploying an ADS on public roads, it is of paramount importance to ensure that there is no negative impact on the traffic safety. It has even been suggested [188] that vehicles controlled by an ADS should be at least 4 to 5 times as safe as human-driven vehicles in order to be accepted by the general public.

Safety validation of an ADS is essential to guarantee that the ADS is safe enough to be allowed on public roads. Retrospective safety validation alone, i.e., through test drives with prototypes, requires millions of kilometers of driving [155], which makes this practically infeasible. Therefore, prospective safety validation, i.e., before performing (test) drives in public traffic, is required. Scenario-based safety validation is an approach for prospective safety validation that is broadly supported by the automotive field [12, 15, 94, 196, 206, 212, 213, 228, 229, 233, 235, 251, 278].

As a consequence of the broad support for scenario-based safety validation, significant research progress has been made. Recent research focuses, among others, on scenario terminology [79, 284], scenario-based requirement verification [156], virtual simulation of scenarios [85, 158, 236, 238, 306], generation of scenarios [78, 101, 253, 269], and scenario databases [12, 94, 229]. All these components are vital parts for estimating the risk of the deployment of an ADS, where risk is the combination of the probability of occurrence of harm and the extent of that harm. This risk estimation itself, however, has received less attention in the literature despite its essential role in the well-known ISO 26262 [144] and ISO 21448 [143] standards, which capture the state of the art in automotive safety.

The contribution of this work is twofold. First, we propose a novel data-driven method for assessing and quantifying the risk of an ADS considering real-world driving scenarios. To provide a more objective method for risk quantification compared to existing approaches, the proposed method uses real-world data and relies less on the judgments of experts. Second, we show how the risk quantification can be decomposed into the terms exposure, severity, and controllability, such that the method can be applied to assess risks based on the ISO 26262 and ISO 21448 standards. Therefore, our method supports the risk assessment activities of these standards. In the remainder of this section, we elaborate more on the need for our work and we present the research questions that are addressed by the current chapter.

7.1.1. Risk quantification of automated driving systems

In [53, 59, 136, 181, 182], risk quantification is proposed for real-time use, e.g., to support the path planning of a self-driving vehicle, so this is not intended to be

used for a prospective risk assessment. In [172], a method for quantifying the risk of scenarios is proposed based on the probability of occurrence of so-called environmental conditions and the probability that an error propagates in the fault tree given a specified environment condition. However, they [172] do not provide methods to estimate or specify these probabilities. Furthermore, the role of a back-up operator, e.g., a human driver that supervises the ADS, is not considered, whereas the influence of the back-up operator is an important aspect that is considered in the ISO 26262 and ISO 21448 standards. Another method to quantify the risk of a driving scenario is proposed in [292], but this method does not consider the likelihood of encountering the scenario and the role of the back-up operator is not explicitly considered. A quantitative assurance framework is proposed in [28, 299], but this framework assumes that the frequency of accidents is known, whereas this is unknown in a prospective assessment. Furthermore, similar to [172], [28] and [299] do not consider the role of a back-up operator.

To address the aforementioned shortcomings, the current work aims to answer the following question:

Research question 7.1. *How to quantify the risk of an ADS in real-world driving scenarios?*

To answer Research question 7.1, this chapter proposes a novel method for quantifying the risk of an ADS. The first step of the presented method is to identify the scenarios that the ADS encounters in real life. Next, the exposure, i.e., the likelihood of encountering these scenarios, is estimated. Using simulations, the probability that a scenario leads to a harmful event is calculated. Combining this probability with the exposure and the probability that a harmful event leads to an injury results in the estimated risk.

7.1.2. Risk quantification in relation with ISO 26262 and ISO 21448

ISO 26262 is the state-of-the-art standard in automotive functional safety that offers a framework for measuring risk in a qualitative manner. This standard concerns hazards that are the result of malfunctioning behavior of components. In the Hazard Analysis and Risk Assessment (HARA), an Automotive Safety Integrity Level (ASIL) is determined for each hazardous event based on the classification of three aspects: exposure, severity, and controllability (the definitions of these terms will be provided in Section 7.2). The classification has two limitations. First, the classification relies on the judgments of experts of the three risk aspects, which renders the classification subjective [161, 276]. Second, the classification is qualitative and offers only five different levels of risk.

As an addition to the ISO 26262 standard, the ISO 21448 standard, also known as "Safety Of The Intended Functionality (SOTIF)", concerns hazardous behavior due to a functional insufficiency of the intended functionality at the vehicle level as opposed to malfunctioning behavior of components. A functional insufficiency may refer to a failure due to technological limitations of a sensor or an actuator. The hazard in the context of SOTIF is initiated by a so-called triggering condition.

For the risk assessment, the ISO 21448 standard does not determine an ASIL, but the exposure, severity, and controllability can be used to determine the required validation effort. In [143, Annex C.6.4], the exposure, severity, and controllability are quantitatively expressed as probabilities, but no method is provided for determining these probabilities other than that the probabilities can be checked for consistency with the functional safety HARA of the ISO 26262 standard. Therefore, the risk estimation of the ISO 21448 standard inherits the limitations of the ISO 26262 standard, which means that there is no explicit risk estimation method that considers hazardous behavior caused by a triggering condition.

To address the lack of a quantitative and objective approach to assess risks related to the ISO 26262 and ISO 21448 standards, the current work also aims to answer the following question:

Research question 7.2. *How to quantify the risk of an ADS using the three aspects of risk as defined by the ISO 26262 standard: exposure, severity, and controllability?*

To answer Research question 7.2, we decompose the quantified risk into the terms exposure, severity, and controllability. Note that it is not our objective to provide a method that replaces parts of the ISO 26262 and ISO 21448 standards, but rather to provide a method that supports the risk assessment activities of these standards.

7.1.3. Organization of this chapter

This chapter is organized as follows. We first elaborate on the risk assessment in the ISO 26262 and ISO 21448 standards in Section 7.2. Section 7.3 provides a method for risk quantification to answer Research question 7.1. In Section 7.4, Research question 7.2 is answered by describing how the proposed risk quantification method relates to the risk assessment according to the ISO 26262 and ISO 21448 standards. To illustrate the proposed method, a case study[1] involving the risk quantification of an ACC is presented in Section 7.5 and the results are reported in Section 7.6. This chapter ends with a discussion in Section 7.7 and conclusions in Section 7.8.

7.2. ISO 26262 and ISO 21448

In Section 7.2.1, the risk assessment approach provided in the ISO 26262 standard [144] and its shortcomings are described. Section 7.2.2 elaborates on the risk assessment approach described in the ISO 21448 standard [143].

7.2.1. ISO 26262

The ISO 26262 standard [144] captures the state of the art in automotive functional safety. It defines the safety lifecycle and the related safety activities such as the HARA. Other methodologies, such as Systems-Theoretic Processes Analysis (STPA) [4] and Failure Mode and Effect Analysis (FMEA) [279], give guidelines on safety

[1]The code is publicly available at:
https://github.com/ErwindeGelder/ScenarioRiskQuantification.

Table 7.1: Definitions of exposure, severity, and controllability according to the ISO 26262 standard [144].

Term	Definitions
Exposure	State of being in an operational situation that can be hazardous if coincident with the failure mode under analysis
Severity	Estimate of the extent of harm to one or more individuals that can occur in a potentially hazardous event
Controllability	Ability to avoid a specified harm or damage through the timely reactions of the persons involved, possibly with support from external measures

engineering based on systems theory. Unlike the ISO 26262 standard, STPA and FMEA do not offer a framework for measuring risk.

The ISO 26262 standard gives guidelines to assess risk based on hazardous events. A hazardous event is the combination of an operational situation (or a scenario) with a potential source of harm caused by malfunctioning behavior of system components. The standard requires analyzing the risk of each hazardous event based on three aspects: exposure, severity, and controllability; see Table 7.1 for the definitions according to the ISO 26262 standard. In this framework, each aspect is classified in 4 or 5 levels:

- Exposure is classified as 0 ("incredible"), 1, 2, 3, or 4 ("high probability");
- Severity is classified as 0 ("no injuries"), 1, 2, or 3 ("life-threatening injuries, fatal injuries"); and
- Controllability is classified as 0 ("controllable in general"), 1, 2, or 3 ("difficult to control or uncontrollable").

The combination of these aspects contributes to constructing the ASIL ranking A, B, C, D. With an ASIL D, the most stringent requirements on system design, verification, and testing apply while an ASIL A requires the least additional safety measures. If the sum of the aspects is 10, an ASIL D is assigned, representing the most critical level. ASIL B or C is assigned when the sum equals 8 or 9, respectively. If the sum is 7 and no aspect has scored 0, then ASIL A is assigned. In all other cases, there are no requirements to comply with the ISO 26262 standard and the classification "Quality Management (QM)" applies because it is assumed that the QM system of the manufacturer suffices for reducing the risk.

A shortcoming of the ASIL ranking is that the results are subjective. Teuchert [276] mentions that the classification of the ASIL depends very much on the engineers that perform the classification. Also Khastgir *et al.* [161] mention the subjectivity of the ranking: "The two distinct short-comings of the current ISO 26262-2011 standard are guided by the subjective nature of the experts' mental models leading to unreliable ratings and the ability to identify a hazard (including the black swan

events)." Through an experiment, Khastgir *et al.* [161] demonstrated the low (intra-rater) reliability due to the subjectivity of the HARA process. Another disadvantage is the qualitative nature of the ASIL ranking. Because only five different levels of risk are considered (ASIL A to D and QM), only large changes in the overall risk are captured.

Since our proposed method for estimating the risk, presented in Section 7.3, is based on data, it is more objective by nature. In addition, the results are quantitative. Therefore, our proposed method can be used to determine an ASIL-like indicator in a *quantitative* and *objective* manner.

7.2.2. ISO 21448

Whereas the ISO 26262 standard focuses on possible hazards caused by the malfunctioning behavior of system components, the ISO 21448 standard [143] addresses hazardous situations caused by the intended functionality, despite the systems being free from the faults addressed in the ISO 26262 standard. The absence of unreasonable risk due to these hazardous situations is defined as the Safety Of The Intended Functionality (SOTIF). Although no ASIL is determined for SOTIF-related hazards, the aspects exposure, severity, and controllability are still used to adjust the required evidence of the safe operation, including the number validation scenarios used for testing. For determining these aspects, the ISO 21448 standard refers to the ISO 26262 standard, which — as reasoned earlier — provides a subjective and qualitative approach.

To address the qualitative nature of the ISO 26262 standard, Annex C of the ISO 21448 standard provides statistical methods to guide the SOTIF verification and validation. A method to quantify the validation targets is provided in [143, Annex C.2], while [143, Annex C.3] proposes to use on-road data or simulations to validate whether systems meet these targets. The use of importance sampling to lessen the amount of simulation testing is discussed in [143, Annex C.5], but no quantification of, e.g., the risk, is provided. In [143, Annex C.6], a statistical approach is presented for arguing that a safety criterion is met while considering the performance of the constituent components. Also in [143, Annex C.6], the risk aspects exposure, severity, and controllability are quantitatively expressed as probabilities. No method is provided, however, to estimate these probabilities. Thus, where Annex C of the ISO 21448 standard provides a step towards making the risk evaluation more quantitative, the current work aims to be more explicit in quantifying the risk. More specifically, in Section 7.4, we show how our proposed method for risk quantification can be used to quantify the aspects exposure, severity, and controllability. Furthermore, Section 7.4 describes how to combine these aspects to quantify the risk.

7.3. Method for risk quantification

To answer Research question 7.1, this section proposes a novel method for quantifying the risk of an ADS in real-world driving scenarios. To structure the risk quantification, the method consists of six steps:

Table 7.2: The terms and definitions.

Term	Definition
Risk	Combination of the probability of occurrence of harm and the severity of that harm [144]
ODD	Operating conditions under which a given driving automation system or feature thereof is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics [243]
Scenario	Quantitative description of the relevant characteristics and activities and/or goals of the ego vehicle(s), the static environment, the dynamic environment, and all events that are relevant to the ego vehicle(s) within the time interval between the first and the last relevant event [79] (Definition 2.1)
Scenario category	Qualitative description of the relevant characteristics and activities and/or goals of the ego vehicle(s), the static environment, and the dynamic environment [79] (Definition 2.4)
Triggering condition	Specific conditions of a scenario that [may] serve as an initiator for a subsequent system reaction leading to hazardous behavior [143]

1. Identify the scenarios that are part of the so-called Operational Design Domain (ODD) of the ADS.
2. Determine the exposure of these scenarios, i.e., the expected number of occurrences per hour of driving.
3. Simulate the response of the ADS in these scenarios.
4. Calculate the expected number of harmful events.
5. Calculate the expected injury rate.
6. Calculate the risk by combining the exposure and the injury rate.

These steps are schematically shown in Figure 7.1. Figure 7.1 also shows how the risk aspects exposure, severity, and controllability relate to these steps. This relation will be further explained in Section 7.2.

Table 7.2 presents the definitions of the terms that are used in our proposed method. In this work, the probability of u is denoted by $\mathbb{P}(u)$, while $\mathbb{E}[u]$ denotes the expectation of u. In the following subsections, the 6 steps for quantifying the risk are described.

7.3.1. Identification of scenarios

An ADS is designed to operate within its ODD, which is defined by the ADS developer and typically consists of a geofence and some known operational conditions, e.g.,

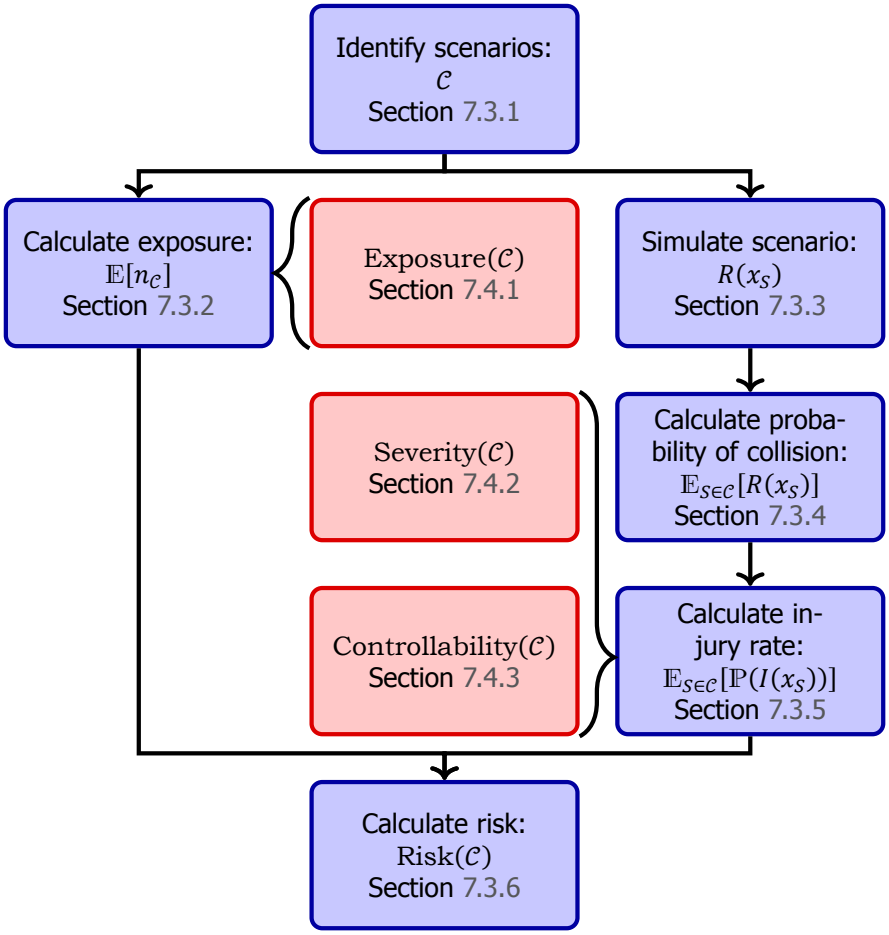


Figure 7.1: Schematic overview of the risk quantification method presented in this chapter. The blue blocks represent the six steps of the risk quantification method explained in Section 7.3. Each of these steps is further explained in Sections 7.3.1 to 7.3.6, respectively. The red blocks show the relation between our proposed method and the three aspects of risk as proposed in the ISO 26262 standard. Each of these aspects is further explained in Sections 7.4.1 to 7.4.3, respectively.

see [22, 111, 301]. The ODD is used to confine the risk analysis [125]. To quantify the risk of an ADS in driving scenarios, the ODD of the ADS must be known. Once deployed, the ADS needs to deal with many scenarios and the ODD in which the ADS is operating determines the variety of these scenarios. It is our goal to provide a method for determining the risk for the ADS in these scenarios.

Considering the wide variety of scenarios, we propose to distinguish between quantitative scenarios and qualitative scenarios, where scenario categories refer to the latter, see Table 7.2. It is assumed that all possible scenarios within a given ODD can be categorized into one or more scenario categories. This assumption does not limit the applicability of the methodology proposed in this work, though it might require many scenario categories to describe all these scenarios. In the remainder of this section, we propose a method to calculate the risk for an ADS in all scenarios that are categorized by the same scenario category, i.e., for all $S \in \mathcal{C}$ [79], where S and $\mathcal{C}$ denote a scenario and a scenario category, respectively. For example, the scenario category "cut-in" comprises all possible cut-in scenarios in the ODD of the ADS. See [74] for more examples of scenario categories.

Remark 7.1. As part of the scenarios, some factors may cause hazardous behavior. These factors are called *triggering conditions* because they may trigger some specific behavior [143]. Typically, triggering conditions may happen rarely, so the impact on safety may not be known. In our case, one or more triggering conditions could be part of a scenario category. Examples of triggering conditions are heavy rain, low road friction, poor lighting, or dirty sensor(s). For more examples, see [143, Annex B.2]. ◊

7

7.3.2. Probability of exposure

To determine the exposure, we estimate the expected number of encounters of a scenario $S \in \mathcal{C}$. Let $n_{\mathcal{C}}$ denote the number of encounters per unit of time of a scenario belonging to scenario category $\mathcal{C}$. We express the exposure as $\mathbb{E}[n_{\mathcal{C}}]$, i.e., the expected number of encounters per unit of time of a scenario belonging to scenario category $\mathcal{C}$:

$$\mathbb{E}[n_{\mathcal{C}}] = \sum_{n=1}^{\infty} n\mathbb{P}(n_{\mathcal{C}} = n), \tag{7.1}$$

where $\mathbb{P}(n_{\mathcal{C}} = n)$ denotes the probability of encountering n scenarios belonging to scenario category $\mathcal{C}$.

We propose to determine $\mathbb{P}(n_{\mathcal{C}} = n)$, $n = 0, 1, 2, ...$, based on data because the data provide a quantitative way to estimate $\mathbb{P}(n_{\mathcal{C}} = n)$, $n = 0, 1, 2, ...$ Furthermore, assuming that the data are collected with the same conditions as specified by the ODD of the ADS, the data provides an objective way to estimate $\mathbb{P}(n_{\mathcal{C}} = n)$. The probability can be estimated by counting the number of occurrences of the scenarios in the data. The method to find the scenarios belonging to $\mathcal{C}$ is explained in Chapter 4 [73]: First, tags are used to describe activities, such as lane changing and braking, and statuses, such as "leading vehicle" and "driving slower". Note that a tag is typically associated with an object and has a start time and an end time.

Second, by searching for a particular combination of these tags that describes the scenario category $\mathcal{C}$, the start and end time of the scenarios are found.

Remark 7.2. It is not uncommon to assume that

- the occurrence of a scenario $S_1 \in \mathcal{C}$ does not affect the probability that a second scenario $S_2 \in \mathcal{C}$ occurs,
- the expected rate at which a scenario belonging to $\mathcal{C}$ occurs is constant, and
- two scenarios belonging to $\mathcal{C}$ cannot occur at exactly the same time.

In that case, the probability $\mathbb{P}(n_{\mathcal{C}} = n)$ simplifies to a Poisson distribution:

$$\mathbb{P}(n_{\mathcal{C}} = n) = \frac{\lambda^n}{n!} \exp\{-\lambda\}. \tag{7.2}$$

Here, $\lambda > 0$ is a parameter that determines the rate at which a scenario belonging to $\mathcal{C}$ occurs. Assuming (7.2), (7.1) simplifies to $\mathbb{E}[n_{\mathcal{C}}] = \lambda$. ◊

7.3.3. Simulation of scenarios

The next step of the risk quantification is to simulate how the ADS behaves in the scenarios belonging to scenario category $\mathcal{C}$. Let $x_S \in \mathbb{R}^d$ denote the d-dimensional parameter vector that describes scenario S. The (stochastic) outcome of a simulation of scenario S is denoted by $R(x_S)$, where $R(x_S) = 1$ means that the simulation of the scenario with parameters x_S ends with an unsuccessful outcome and otherwise $R(x_S) = 0$. An unsuccessful outcome may be a crash or a situation where the ego vehicle ends off the road. In addition to $R(x_S)$, the output of a simulation run provides information to rank multiple scenarios from "most critical" to "least critical" (see Section 7.3.4) and information to estimate the extent of harm in case of a crash (see Section 7.3.5).

To enable the simulation of the scenarios, a simulation framework is set up. Figure 7.2 shows the scheme of the simulation framework, which is represented by the following five blocks:

- World: the relevant information about the environment of the system under test. This includes other vehicles.
- Sensors: mapping of the global information to sensor data that can be used by the ADS.
- ADS: the logic and control laws to perform an automated function. The ADS uses the information of the sensors to determine the input to the vehicle. The ADS provides input to the actuators of the vehicle and information to the operator, e.g., a beep in case of a crash warning.
- Operator: the actual driver that is behind the steering wheel or a remote driver.

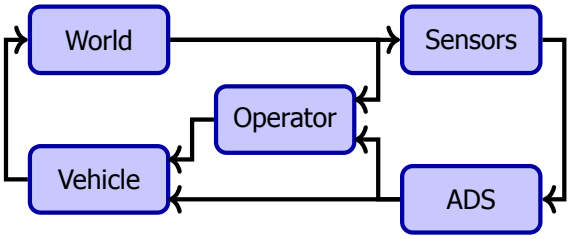


Figure 7.2: Scheme of the simulation framework.

- Vehicle: the system consisting of actuators for translating the inputs generated by the ADS and the operator into vehicle motions and subsystems for enhancing conspicuity via, e.g., lighting, signaling, and sounding the horn.

Note that the simulation framework is easily extended to consider multiple ADSs, operators, and vehicles. This could be of interest for testing, e.g., (cooperative) ACC systems because one might be interested in the performance of a platoon of vehicles rather than the performance of a single vehicle (see, e.g., [217, 287]).

7.3.4. Probability of a crash

Instead of estimating the risk of an ADS in a specific scenario with parameters x, the goal is to calculate the risk for all scenarios from scenario category $\mathcal{C}$. The first step toward the calculation of the risk is to compute the expected outcome while averaging over all scenarios in $\mathcal{C}$: $\mathbb{E}_{S\in\mathcal{C}}[R(x_S)]$. Here, the subscript $S \in \mathcal{C}$ indicates that the expectation is computed while averaging over all scenarios belonging to scenario category $\mathcal{C}$.

To provide inputs to the simulation, scenarios are identified and characterized from real-world driving data, e.g., collected in field operational tests or naturalistic driving studies. In this way, the scenarios are most likely to represent real-world driving conditions [94, 171, 229]. One approach would be to simply simulate the scenarios that are recorded from data, but this gives two problems. First, because not all possible variations of the scenarios might be found in the data, the failure modes of the ADS might not be reflected in the simulations [323]. Second, because the set of extracted scenarios is largely composed of non-safety critical scenarios, many scenarios may be required to obtain the required statistical accuracy of rare events such as crashes [149, 323]. To overcome these problems, so-called importance sampling can be applied in order to put more emphasis on scenarios that are likely to trigger safety-critical situations [70, 149, 315, 323].

In this section, we propose a nonparametric importance sampling method without requiring a-priori knowledge of what might be scenarios that are likely to trigger safety-critical situations given the ADS under test. First, crude Monte Carlo sampling is used. Next, the nonparametric importance sampling method based on the simulation results of the crude Monte Carlo sampling is proposed.

Crude Monte Carlo sampling

With crude Monte Carlo sampling, parameters are sampled from a probability density function (pdf). Let $f_{\mathcal{C}}(\cdot)$ denote the pdf of the parameters of the scenarios from scenario category $\mathcal{C}$. Typically, the pdf $f_{\mathcal{C}}(\cdot)$ is unknown, so it needs to be estimated. To estimate the pdf, we use the observed scenarios that have also been used to estimate the exposure (Section 7.3.2). Since the shape of the pdf is also unknown beforehand, assuming a predefined functional form of the pdf for which certain parameters are fitted to the data may lead to inaccurate fits unless a lot of hand-tuning is applied. As an alternative, we propose to estimate $f_{\mathcal{C}}(\cdot)$ using Kernel Density Estimation (KDE) [222, 237]. Let $x_{S_1}, \ldots, x_{S_{N_x}}$ denote the parameters of the observed scenarios $S_i \in \mathcal{C}$, $i \in \{1, \ldots, N_x\}$, then the density $f_{\mathcal{C}}(\cdot)$ is estimated by

$$\hat{f}_{\mathcal{C}}(x) = \frac{1}{N_x h^d} \sum_{i=1}^{N_x} K\left(\frac{1}{h}\left(x - x_{S_i}\right)\right). \tag{7.3}$$

Here, $K(\cdot)$ is the so-called kernel and h is the bandwidth. The choice of the kernel function is not as important as the choice of the bandwidth [87, 283]. We use the often-used Gaussian kernel[2], but our method does not depend on the choice of kernel. The Gaussian kernel is given by

$$K(u) = \frac{1}{(2\pi)^{d/2}} \exp\left\{-\frac{1}{2}\|u\|_2^2\right\}, \tag{7.4}$$

where $\|u\|_2^2 = u^{\mathsf{T}}u$ denotes the squared 2-norm of u.

The bandwidth $h > 0$ is a free parameter that controls the width of the kernel. A larger bandwidth results in a smoother pdf, but choosing h too large may result in loss of details in the pdf. Methods for estimating the bandwidth range from simple reference rules like Silverman's rule of thumb [262] to more elaborate methods (for reviews, see [24, 151, 283]). We use leave-one-out cross validation to determine the optimal bandwidth because this minimizes the Kullback-Leibler divergence between the estimated pdf, $\hat{f}_{\mathcal{C}}(\cdot)$, and the unknown pdf that underlies the data, $f_{\mathcal{C}}(\cdot)$ [283]. With leave-one-out cross validation, the bandwidth equals:

$$\arg\max_h \prod_{i=1}^{N_x} \left(\frac{1}{(N_x - 1)h^d} \sum_{j=1, j\neq i}^{N_x} K\left(\frac{1}{h}\left(x_{S_j} - x_{S_i}\right)\right)\right). \tag{7.5}$$

Note that with the one-dimensional bandwidth h, the same amount of smoothing is applied in every direction. Therefore, the parameters are first scaled, such that they have the same standard deviation in each dimension. Our method can easily be extended to a multi-dimensional bandwidth [87, 120].

Sampling from $\hat{f}_{\mathcal{C}}$ is straightforward. First, an integer $j \in \{1, \ldots, N_x\}$ is chosen randomly with each integer having equal likelihood. Next, a random sample is

[2]The advantage of the Gaussian kernel is that it gives the possibility to calculate a metric that quantifies the completeness of the data (Chapter 3 [72]) and to apply conditional sampling when generating scenario parameters (Chapter 6 [75]). Both these topics are out of scope of this chapter.

drawn from a Gaussian with covariance $h^2 I_d$ and mean x_{S_j}, where I_d denotes the d-by-d identity matrix.

With crude Monte Carlo, the probability of a crash is calculated by taking the mean of $R(x_S)$ over a large number, N_{MC}, of different realizations of x_S:

$$\mu_{\mathrm{MC}} = \frac{1}{N_{\mathrm{MC}}} \sum_{j=1}^{N_{\mathrm{MC}}} R(x_j), \quad x_j \sim \hat{f}_{\mathcal{C}}. \tag{7.6}$$

It is easy to see that the crude Monte Carlo result is unbiased, i.e., $\mathbb{E}[\mu_{\mathrm{MC}}] = \mathbb{E}_{S\in\mathcal{C}}[R(x_S)]$. To estimate the potential approximation error, $\mu_{\mathrm{MC}} - \mathbb{E}_{S\in\mathcal{C}}[R(x_S)]$, the estimated standard deviation of (7.6) is commonly used:

$$\sigma_{\mathrm{MC}} = \frac{1}{N_{\mathrm{MC}}} \sqrt{\sum_{j=1}^{N_{\mathrm{MC}}} \left(\mu_{\mathrm{MC}} - R(x_j)\right)^2}. \tag{7.7}$$

Nonparametric importance sampling

In general, it can be expected that the probability of a crash, $\mathbb{E}_{S\in\mathcal{C}}[R(x_S)]$, is small. As a result, none or few of the N_{MC} scenario simulations may end with a crash and the relative uncertainty, i.e., $\sigma_{\mathrm{MC}}/\mu_{\mathrm{MC}}$, will be high or undefined (in case none of the scenario simulations ends with a crash). With importance sampling, the scenario parameters are sampled from a different distribution — the so-called importance density — such that the simulation runs focus more on scenarios in which the probability of a crash is high. This will lead to a lower relative uncertainty of the estimated probability of a crash. We use nonparametric importance sampling, which means that a nonparametric method is used to estimate the unknown optimal importance density [321]. More specifically, as proposed in [321], the nonparametric importance sampling method employs KDE to construct the importance density.

Let $g(\cdot)$ denote the importance density for sampling the scenario parameters. If N_{NIS} scenarios are simulated in nonparametric importance sampling, the estimated probability of a crash is

$$\mu_{\mathrm{NIS}} = \frac{1}{N_{\mathrm{NIS}}} \sum_{j=1}^{N_{\mathrm{NIS}}} R(x_j) \frac{\hat{f}_{\mathcal{C}}(x_j)}{g(x_j)}, \quad x_j \sim g. \tag{7.8}$$

The weight $\hat{f}_{\mathcal{C}}(x_j)/g(x_j)$ is used to correct for the bias introduced by sampling from $g(\cdot)$ instead of $\hat{f}_{\mathcal{C}}(\cdot)$. If $g(x) > 0$ whenever $R(x)\hat{f}_{\mathcal{C}}(x) > 0$, then (7.8) gives unbiased results [219]. The estimated standard deviation of (7.8) is

$$\sigma_{\mathrm{NIS}} = \frac{1}{N_{\mathrm{NIS}}} \sqrt{\sum_{j=1}^{N_{\mathrm{NIS}}} \left(\frac{R(x_j)\hat{f}_{\mathcal{C}}(x_j)}{g(x_j)} - \mu_{\mathrm{NIS}} \right)^2}. \tag{7.9}$$

In the ideal case, we choose $g(\cdot)$ such that the actual standard deviation of μ_{NIS} is minimized. This ideal $g(\cdot)$ is, however, unknown because it depends on $R(\cdot)$, for which no functional form is available, and on the unknown $\mathbb{E}_{S\in\mathcal{C}}[R(x_S)]$. Therefore, we propose to first conduct crude Monte Carlo sampling and to base $g(\cdot)$ on this result. Let $\{x_{(1)}, \ldots, x_{(N_{\text{MC}})}\}$ denote the ordered set of scenario parameters from the crude Monte Carlo sampling, such that $x_{(1)}$ and $x_{(N_{\text{MC}})}$ are the parameter vectors corresponding to the "most critical" scenario and "least critical" scenario, respectively. Then, we use the KDE technique described earlier to construct $g(\cdot)$ using the $N_{\text{C}} < N_{\text{MC}}$ "most critical" scenarios:

$$g(x) = \frac{1}{N_{\text{C}} h_{\text{NIS}}^d} \sum_{i=1}^{N_{\text{C}}} K\left(\frac{1}{h_{\text{NIS}}}\left(x - x_{(i)}\right)\right). \tag{7.10}$$

Using the Gaussian kernel of (7.4), it follows that $g(x) > 0$ for all x, such that (7.8) gives an unbiased result. The bandwidth h_{NIS} is estimated using leave-one-out cross validation, similar to h in (7.5).

To order the scenarios from "most critical" to "least critical", a metric for quantifying the (maximum) risk during a single simulation run can be used. For illustration purposes, this work uses the minimum absolute Time to Collision (TTC) [130] during each simulation run. Note that the presented method can easily be applied with other metrics. The TTC is defined as the ratio of the distance toward an approaching object and the speed difference with that object. In case the simulation of a scenario ends in a crash, the minimum TTC is 0. Note that N_{C} must be larger than the number of simulation runs of the crude Monte Carlo sampling that ended in a crash and N_{C} must be substantially smaller than N_{MC}, e.g., an order of magnitude, to really benefit from the nonparametric importance sampling.

7.3.5. Calculation of severity

Besides the probability of a crash, risk also includes the extent of the harm in a potential crash. We express the severity as the expectation of the probability of a moderate injury or worse, corresponding to a Maximum Abbreviated Injury Scale (MAIS) [225] level of 2 or higher: $\mathbb{E}_{S\in\mathcal{C}}[\mathbb{P}(I(x_S))]$.

Typically, two different approaches are considered for predicting the extent of harm in a crash. The first approach involves simulation of the crash. The simulations as explained in Section 7.3.3 and as used in Section 7.3.4 consider the pre-crash phase and are used to determine initial conditions and boundary conditions for the simulation of the in-crash phase. The extent of harm is predicted by simulations of the in-crash phase using biomechanical models; see [39, 46, 224, 230, 317] for an overview. The second approach makes use of phenomenological injury risk functions based on the research of the relationships between crash parameters and the probability of an injury [153]. The most commonly used crash parameter is the impact velocity change [21, 106, 107, 174, 318]. Other factors for determining the probability of an injury are belt usage [106, 174, 318], occupant age [21, 318], peak acceleration during the crash [105], airbag deployment [21], and seating position of the occupants [21]. Typically, logistic regression is used to model the relationship

between crash parameters and the probability of an injury.

In this chapter, we assume that a method to estimate the probability of an injury given a parameterized scenario, i.e., $\mathbb{P}(I(x_S))$, is known. In the case study in Section 7.5, an example is provided. To estimate $\mathbb{E}_{S\in\mathcal{C}}[\mathbb{P}(I(x_S))]$, the same approach for estimating the expectation of $R(x)$ in (7.8) is used:

$$\mathbb{E}_{S\in\mathcal{C}}[\mathbb{P}(I(x_S))] \approx \mu_{\text{injury}} = \frac{1}{N_{\text{NIS}}}\sum_{j=1}^{N_{\text{NIS}}} w(x_j), \quad x_j \sim g, \tag{7.11}$$

with

$$w(x_j) = \mathbb{P}(I(x_j))R(x_j)\frac{\hat{f}_{\mathcal{C}}(x_j)}{g(x_j)}. \tag{7.12}$$

The estimated standard deviation of μ_{injury} is

$$\sigma_{\text{injury}} = \frac{1}{N_{\text{NIS}}}\sqrt{\sum_{j=1}^{N_{\text{NIS}}}\left(w(x_j) - \mu_{\text{injury}}\right)^2}. \tag{7.13}$$

7.3.6. Risk quantification

The risk associated with a scenario category $\mathcal{C}$ can be defined as the combination of the probability of occurrence of a scenario of $\mathcal{C}$ and the probability of an injury given such a scenario. Thus, the risk is mathematically defined as:

$$\text{Risk}(\mathcal{C}) = \mathbb{E}[n_{\mathcal{C}}] \cdot \mathbb{E}_{S\in\mathcal{C}}[\mathbb{P}(I(x_S))], \tag{7.14}$$

where $\mathbb{E}[n_{\mathcal{C}}]$ is defined in (7.1) and $\mathbb{E}_{S\in\mathcal{C}}[\mathbb{P}(I(x_S))]$ is estimated in (7.11).

Remark 7.3. The risks of two scenario categories can be combined as follows:

$$\text{Risk}(\mathcal{C}_1 \cup \mathcal{C}_2) = \text{Risk}(\mathcal{C}_1) + \text{Risk}(\mathcal{C}_2) - \text{Risk}(\mathcal{C}_1 \cap \mathcal{C}_2). \tag{7.15}$$

In general, it is sufficient to estimate the upper bound of the risk, so in case it is practically difficult to evaluate $\text{Risk}(\mathcal{C}_1 \cap \mathcal{C}_2)$, one can use

$$\text{Risk}(\mathcal{C}_1 \cup \mathcal{C}_2) \leq \text{Risk}(\mathcal{C}_1) + \text{Risk}(\mathcal{C}_2), \tag{7.16}$$

where equality applies if two scenario categories, $\mathcal{C}_1$ and $\mathcal{C}_2$, do not overlap. ◊

7.4. Relation with ISO 26262 and ISO 21448

In this section, we propose how to quantify the exposure, severity, and controllability in Sections 7.4.1 to 7.4.3, respectively. Finally, in Section 7.4.4, we show that combining these aspects results in the earlier calculated risk of (7.14).

7.4.1. Exposure

We consider the likelihood of being in a scenario that is comprised by the scenario category $\mathcal{C}$. Hence, the exposure is similar to (7.1):

$$\text{Exposure}(\mathcal{C}) = \mathbb{E}[n_{\mathcal{C}}]. \tag{7.17}$$

Note that the ISO 26262 standard [144] also mentions the "failure mode", see Table 7.1. The ISO 21448 standard [143], however, does not consider a specific "failure mode", which is why we focus on the likelihood of "being in an operational situation". Here, the "operational situation" is described by the scenario category $\mathcal{C}$.

7.4.2. Severity

In Section 7.3.5, a method to compute the severity has been proposed. Following the reasoning of the ISO 26262 standard, however, the severity is defined slightly differently, as it is assumed that there is no operator that can avoid harm or damage. Therefore, we define severity as follows:

$$\text{Severity}(\mathcal{C}) = \mathbb{E}_{S\in\mathcal{C}}[\mathbb{P}(I(x_S))|\text{no operator}], \tag{7.18}$$

where "no operator" indicates that the backup function of the operator is not considered in the simulations.

Remark 7.4. An operator might still be considered in the simulation. For example, if the ADS only controls the vehicle in the longitudinal direction, an operator is still in charge of lateral control. In this example, the notation "no operator" indicates that the driver is not a backup for the longitudinal control. ◊

Note that similar to the exposure, the severity in (7.18) is calculated with respect to a scenario category, whereas the ISO 26262 standard determines the severity with respect to a "hazardous event". Our method still allows to evaluate the severity considering such malfunctioning behavior by simply injecting this malfunctioning behavior as part of the sensor(s), ADS(s), and/or vehicle(s). In a similar manner, any triggering conditions (see Remark 7.1) that may cause hazardous behavior can be included in the simulations.

7.4.3. Controllability

To determine the controllability, we compare the probability of an injury with and without a backup operator. Hence,

$$\text{Controllability}(\mathcal{C}) = \frac{\mathbb{E}_{S\in\mathcal{C}}[\mathbb{P}(I(x_S))]}{\mathbb{E}_{S\in\mathcal{C}}[\mathbb{P}(I(x_S))|\text{no operator}]}. \tag{7.19}$$

If $\text{Controllability}(\mathcal{C}) = 1$, then a backup operator cannot avoid any potential harm. The ISO 26262 standard calls this "difficult to control or uncontrollable". If, on the other hand, $\text{Controllability}(\mathcal{C}) = 0$, then a backup operator is able to avoid any potential harm.

Remark 7.5. It might be counterintuitive that a higher score for $\mathrm{Controllability}(\mathcal{C})$ indicates that the situation is less controllable. The ISO 26262 standard also assigns higher scores for the controllability in case the situation is less controllable, see Section 7.2.1. We have chosen to prefer consistency with respect to the ISO 26262 standard, which is why $\mathrm{Controllability}(\mathcal{C}) = 1$ means that the situation is difficult to control and $\mathrm{Controllability}(\mathcal{C}) = 0$ means that the situation is easy to control. ◊

7.4.4. Combining the risk aspects to compute the risk

To compute the risk, the scores for the exposure, severity, and controllability are multiplied:

$$\mathrm{Risk}(\mathcal{C}) = \mathrm{Exposure}(\mathcal{C}) \cdot \mathrm{Severity}(\mathcal{C}) \cdot \mathrm{Controllability}(\mathcal{C}). \tag{7.20}$$

Substituting (7.17) to (7.19) into (7.20) results in the risk of (7.14).

Remark 7.6. The ASIL ranking is obtained by summing the scores for the exposure, severity, and controllability, while the risk in (7.20) is obtained by multiplying the scores for the exposure, severity, and controllability. Following the ISO 26262 standard, the scoring for the exposure, severity, and controllability is such that one point difference corresponds to an order in magnitude. Therefore, loosely said, the ASIL is proportional to the log of the risk of (7.20) while the individual scores of the exposure, severity, and controllability are proportional to the log of (7.17), (7.18), and (7.19), respectively. Since we can rewrite (7.20) as

$$\log \mathrm{Risk}(\mathcal{C}) = \log \mathrm{Exposure}(\mathcal{C}) + \log \mathrm{Severity}(\mathcal{C}) + \log \mathrm{Controllability}(\mathcal{C}), \tag{7.21}$$

(7.20) is consistent with the way the risk is determined in the ISO 26262 standard. ◊

7.5. Case study

This section explains the details of the case study; the results are reported in the next section. In Section 7.5.1, the models for the ADS under test and the human driver that serves as a backup operator are described. The scenario categories and triggering conditions are listed in Sections 7.5.2 and 7.5.3, respectively. Section 7.5.4 describes the data set. This section ends with details on the simulation (Section 7.5.5) and the severity estimation (Section 7.5.6).

7.5.1. Automated driving system under test

For the ADS under test, we consider the ACC model described by Xiao *et al.* [312]. This ACC model is based on the ACC model proposed by Milanés and Shladover [199] but also includes a safety distance d_0. The ACC function adjusts the ego vehicle speed such that the ego vehicle maintains a safe distance from a leading vehicle. If there is no vehicle ahead or the distance between the ego vehicle and the leading vehicle is large, then the ACC acts like a Cruise Control (CC) that maintains a constant speed set by the driver. The acceleration of the ACC is based on the

speed of the ego vehicle, $v_e(t)$, the speed of the leading vehicle, $v_l(t)$, and the gap between the leading vehicle's back and the ego vehicle's front, $g(t)$, at time t. If the ACC function is active, the acceleration of the ego vehicle at time t is described by the following equations [312]:

$$a_e(t) = \max(\min(a_{ACC}(t), a_{CC}(t)), -d_{max}), \tag{7.22}$$

$$a_{ACC}(t) = \begin{cases} k_1 g_{error}(t) + k_2 v_{error}(t) & \text{if } g(t) < d_{ACC}, \\ a_{CC}(t) & \text{otherwise,} \end{cases} \tag{7.23}$$

$$g_{error}(t) = g(t) - d_0(v_e(t)) - \tau_h v_e(t), \tag{7.24}$$

$$v_{error}(t) = v_l(t) - v_e(t), \tag{7.25}$$

$$d_0(u) = \begin{cases} 5\,\text{m} & \text{if } u \geq 15\,\text{m/s}, \\ 7\,\text{m} & \text{if } u < 10.8\,\text{m/s}, \\ \frac{75\,\text{m}^2/\text{s}}{u} & \text{otherwise,} \end{cases} \tag{7.26}$$

$$a_{CC}(t) = k_{CC}(v_{set} - v_e(t)). \tag{7.27}$$

The values and descriptions of the parameters d_{max}, d_{ACC}, k_1, k_2, τ_h, and k_{CC} are provided in Table 7.3. The parameter v_{set} is the desired speed, which is assumed to be the same as the initial speed of the ego vehicle in each simulation run, i.e., $v_{set} = v_e(0)$.

Following Xiao *et al.* [312], the human driver takes over control from the ACC in the following circumstances:

- In case of a Forward Collision Warning (FCW). If the FCW is at time t, then the human driver takes over at $t + t_r$, where t_r is the reaction time of the human driver.
- In case the ego vehicle approaches the leading vehicle with a relative speed of $15\,\text{m/s}$ and the gap between the ego vehicle and the leading vehicle is less than the perception range, d_{view}.

Similar as in [312], the FCW model is taken from [162] and is triggered when $1/(1 + \exp\{-\beta\}) > 0.75$, with

$$\beta = \begin{cases} -6.09 + 18.82\frac{v_e - v_l}{g} + 0.12 v_e & \text{if } v_l > 0 \wedge a_l < 0, \\ -6.09 + 12.58\frac{v_e - v_l}{g} + 0.12 v_e & \text{if } v_l > 0 \wedge a_l \geq 0, \\ -9.07 + 24.23\frac{v_e - v_l}{g} + 0.12 v_e & \text{otherwise,} \end{cases} \tag{7.28}$$

where a_l denotes the acceleration of the leading vehicle. The reaction time of the driver, t_r, is distributed according to the log-normal distribution with a mean of $0.92\,\text{s}$ and a standard deviation of $0.28\,\text{s}$ [121].

Similar to Xiao *et al.* [312], we use the Intelligent Driver Model plus (IDM+) [247] to model the human driver behavior. If the human driver is controlling the

Table 7.3: Parameters of the system under test and human driver behavior model. If the parameter value is taken from literature, the corresponding reference is mentioned after the value. For the parameters $d_{\max}$ and d_{view}, different values are considered as so-called triggering conditions that are included in the scenarios, see Section 7.5.3.

Parameter	Description	Value
$d_{\max}$	Maximum deceleration (value with triggering condition "limited braking capacity")	$6\,\text{m/s}^2$ ($3\,\text{m/s}^2$)
d_{ACC}	Maximum sensor range of ACC	$150\,\text{m}$
k_1	Distance gain of ACC	$0.23\,\text{s}^{-2}$ [312]
k_2	Speed gain of ACC	$0.07\,\text{s}^{-1}$ [312]
τ_{h}	Time-gap setting, also known as desired time headway	$1.1\,\text{s}$ [312]
k_{CC}	Speed gain of CC	$0.4\,\text{s}^{-1}$ [312]
v_{set}	Desired speed	Variable
t_{r}	Reaction time	Variable [121]
d_{view}	Perception range of human driver (value with triggering condition "poor visibility")	$150\,\text{m}$ ($60\,\text{m}$) [312]
k_{a}	Maximum acceleration for human driver	$0.73\,\text{m/s}^2$ [281]
k_{d}	Maximum deceleration for human driver	$1.67\,\text{m/s}^2$ [281]
δ	Acceleration exponent for human driver	4 [281]
s_0	Safety distance for human driver	$2\,\text{m}$ [281]

7

vehicle and $g(t) \leq d_{\text{view}}$, then the acceleration of the ego vehicle at time $t + t_{\text{r}}$ is described by the following equations [247]:

$$a_{\text{e}}(t + t_{\text{r}}) = \max(a_{\text{IDM}}(t), -d_{\max}), \tag{7.29}$$

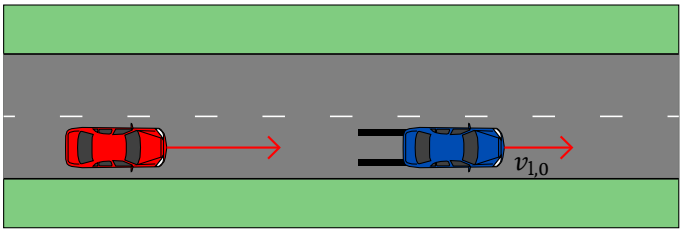
$$a_{\text{IDM}}(t) = k_{\text{a}} \min\left(1 - \left(\frac{v_{\text{e}}(t)}{v_{\text{set}}}\right)^{\delta}, 1 - \left(\frac{g^*(v_{\text{e}}(t), v_{\text{e}}(t) - v_{\text{l}}(t))}{g(t)}\right)^2\right), \tag{7.30}$$

$$g^*(u, v) = s_0 + \tau_{\text{h}} u + \frac{uv}{2\sqrt{k_{\text{a}} k_{\text{d}}}}. \tag{7.31}$$

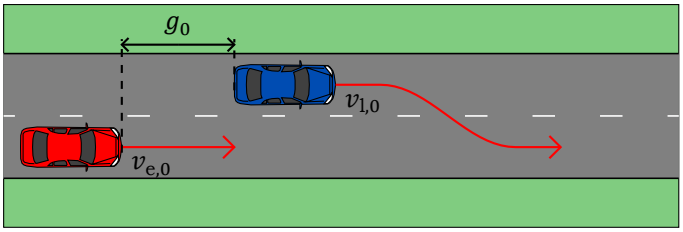
Table 7.3 provides the descriptions of the constants k_{a}, k_{d}, δ, s_0, and τ_{h}. If $g(t) > d_{\text{view}}$, then $a_{\text{e}}(t + t_{\text{r}}) = 0\,\text{m/s}^2$.

7.5.2. Scenario categories

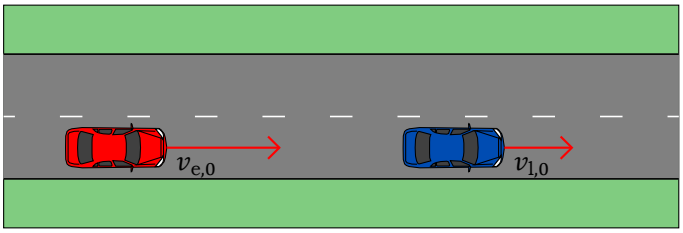
This case study considers three scenario categories named "leading vehicle decelerating (LVD)" ($\mathcal{C}_{\text{LVD}}$), "cut-in" ($\mathcal{C}_{\text{cut-in}}$), and "approaching slower vehicle (ASV)" ($\mathcal{C}_{\text{ASV}}$), see Figure 7.3. The relevance of the first two scenario categories is exemplified by the proposed regulation for automated lane-keeping systems in which these two scenario categories are mentioned as "critical scenarios" [90]. The scenario category ASV, which also includes scenarios in which the leading vehicle is stationary, accounts for more than $25\,\%$ of all crashes that involve two vehicles



(a) Leading vehicle decelerating (LVD).



(b) Cut-in.



(c) Approaching slower vehicle (ASV).

Figure 7.3: Schematic representation of the scenario categories considered in the case study. The left vehicle is the ego vehicle. All parameters are shown except Δ_v and $\tilde{a}$ of the LVD scenario.

[209]. For each scenario category, we will list the parameters that describe the scenarios. Furthermore, we will describe how the speed of the leading vehicle is modeled. Since the ego vehicle is controlled by the ACC or the human driver, only its initial state at $t = 0$ is provided.

$\mathcal{C}_{\text{LVD}}$: Leading vehicle decelerating

In an LVD scenario, the ego vehicle is following another vehicle, which is referred to as the leading vehicle. The LVD scenario starts as soon as the leading vehicle starts to decelerate. The simulation of an LVD scenario ends if the distance between the ego vehicle and the leading vehicle is no longer decreasing or if the ego vehicle and the leading vehicle collide. To describe an LVD scenario, three parameters are used:

- $v_{l,0} > 0$: The initial speed of the leading vehicle;

- $\Delta_v \in (0, v_{l,0}]$: The speed difference of the leading vehicle obtained through decelerating; and
- $\bar{a} > 0$: The average deceleration of the leading vehicle.

To model the speed of the leading vehicle during its deceleration activity, a sinusoidal shape is used:

$$v_l(t) = \begin{cases} v_{l,0} - \frac{\Delta_v}{2}\left(1 - \cos\left(\frac{\pi \bar{a} t}{\Delta_v}\right)\right) & \text{if } t < \frac{\Delta_v}{\bar{a}}, \\ v_{l,0} - \Delta_v & \text{otherwise.} \end{cases} \tag{7.32}$$

It is assumed that the ego vehicle is following the leading vehicle at the same speed, i.e., $v_e(0) = v_{l,0}$. This is also the desired speed, thus $v_{set} = v_{l,0}$. The initial gap is such that the ego vehicle is initially driving at a constant speed, so $g(0) = d_0(v_e(0)) + \tau_h v_e(0)$, such that the initial distance error, $g_{error}(0)$, is zero, see (7.24).

$\mathcal{C}_{cut-in}$: Cut-in

In a cut-in scenario, another vehicle changes lane such that it becomes the leading vehicle of the ego vehicle. The reason for the other vehicle to change lane is not considered, i.e., it may change lane because the driver of the vehicle assumes it is safe and appropriate to change lane or because a lane change is mandatory. A cut-in scenario starts as soon as the other vehicle enters the lane of the ego vehicle. Similarly as for an LVD scenario, the simulation of a cut-in scenario ends as soon as the distance between the ego vehicle and the leading vehicle is no longer decreasing or if there is a crash. To describe a cut-in scenario, three parameters are used:

- $g_0 > 0$: The gap between the leading vehicle and the ego vehicle at the moment of the cut-in;
- $v_{l,0} > 0$: The initial speed of the leading vehicle; and
- $v_{e,0} > 0$: The initial speed of the ego vehicle.

It is assumed that the leading vehicle drives at a constant speed, thus $v_l(t) = v_{l,0}$. The initial gap and the initial speed of the ego vehicle are provided by the parameters: $g(0) = g_0$ and $v_e(0) = v_{e,0}$. The initial speed of the ego vehicle is also the desired speed: $v_{set} = v_{e,0}$.

$\mathcal{C}_{ASV}$: Approaching slower vehicle

In an ASV scenario, another vehicle, referred to as the leading vehicle, is driving in front of the ego vehicle. Furthermore, the leading vehicle is driving slower than the ego vehicle, such that the ego vehicle is approaching this vehicle. The ASV scenario starts if the ego vehicle is at a safe distance and ends if the distance between the two vehicle is no longer decreasing or if the two vehicles collide. To describe an ASV scenario, two parameters are used:

Table 7.4: Information of the participants that drove the $50\,\mathrm{km}$ route six times.

	Minimum	Maximum	Mean
Age [years]	29	60	52.3
Experience [years]	8	42	32.4
Mileage [km/year]	$10 \cdot 10^3$	$40 \cdot 10^3$	$16.1 \cdot 10^3$

- $v_{\mathrm{l},0} \geq 0$: The initial speed of the leading vehicle; and
- $v_{\mathrm{e},0} > 0$: The initial speed of the ego vehicle.

It is assumed that the leading vehicle drives at a constant speed, thus $v_{\mathrm{l}}(t) = v_{\mathrm{l},0}$. The initial speed of the ego vehicle is $v_{\mathrm{e}}(0) = v_{\mathrm{e},0}$, which is also the desired speed: $v_{\mathrm{set}} = v_{\mathrm{e},0}$. The initial gap is $g(0) = \tau_{\mathrm{h},0} v_{\mathrm{e}}(0)$ with $\tau_{\mathrm{h},0} = 4\,\mathrm{s}$, such that the initial distance is safe, considering a time headway of $4\,\mathrm{s}$.

7.5.3. Triggering conditions

To illustrate the application of the proposed method for the risk quantification for the validation of the SOTIF, triggering conditions are included in the scenarios that may trigger hazardous situations, see Remark 7.1. For comparison, the risk is calculated without a triggering condition and with a triggering condition.

The first triggering condition is a limited braking capacity of the ego vehicle. As a result, the maximum deceleration of the ego vehicle is $d_{\max} = 3\,\mathrm{m/s^2}$. The reason for a limited braking capacity is not further specified here, but it could be caused by a loss of road friction, e.g., due to an adverse road condition.

The second triggering condition is a poor visibility. It is assumed that the human driver can only see up to $d_{\mathrm{view}} = 60\,\mathrm{m}$. It is further assumed that the maximum sensor range of the ACC, d_{ACC}, is unaffected. This assumption is reasonable in case the poor visibility is caused by fog because fog has a limited effect on automotive radars that are typically used for an ACC.

7.5.4. Data set

To estimate the exposure and the pdfs of the scenario parameters, the data set described in [221] is used. The data were recorded from a single vehicle in which 20 experienced drivers were asked to drive a prescribed route. Table 7.4 lists more information about the drivers. Each driver drove the $50\,\mathrm{km}$ route, as shown in Figure 7.4, 6 times, which resulted in $63\,\mathrm{h}$ of data. The route contains urban roads, rural roads, and highways. To measure the surrounding traffic, the vehicle was equipped with three radars and one camera. The surrounding traffic was measured by fusing the data of the radars and the camera as described in [92].

To extract the scenarios from the data set, the approach described in [73] is used. LVD scenarios are found by querying for a vehicle that has the tags "leading vehicle" and "braking" at the same time. Cut-in scenarios are found by querying for a vehicle that initially has the tags "changing lane" and "no leading vehicle" which

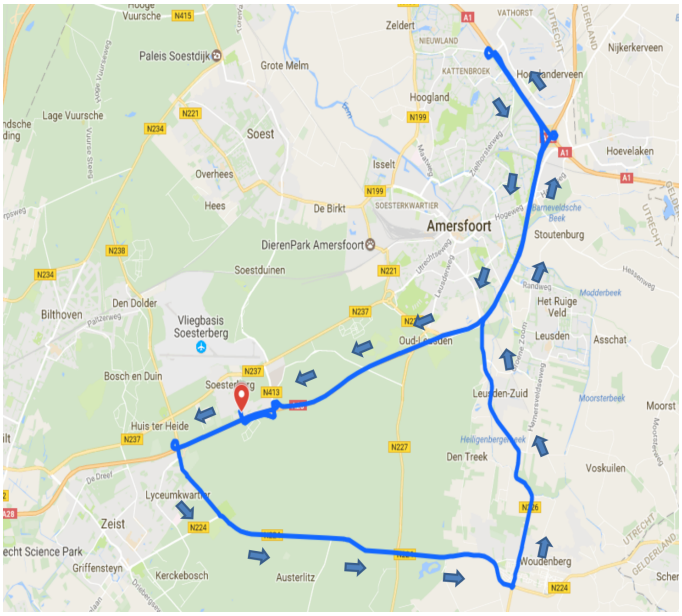


Figure 7.4: The route, including the driving direction, that was driven by the drivers.

changes into the tags "changing lane" and "leading vehicle" [73]. ASV scenarios are found by querying for a vehicle that has the tags "leading vehicle" and "driving slower", where the tag "driving slower" indicates that the vehicle has at most $90\,\%$ of the speed of the ego vehicle. In 63 hours of driving, 1300 LVD scenarios, 297 cut-in scenarios, and 291 ASV scenarios has been found.

7.5.5. Simulations

For the simulation, we use the forward Euler method with a step size of $0.01\,\mathrm{s}$ to compute the positions of the ego vehicle and the leading vehicle. We used Python as the coding language. The code is available on a public repository[3]. Initially, the crude Monte Carlo sampling with $N_{\mathrm{MC}} = 10000$ simulation runs is performed where the scenario parameters are drawn from $\hat{f}_{\mathcal{C}}(\cdot)$ of (7.3). Additionally, if applicable, the driver reaction time, t_{r}, is sampled from the log-normal distribution described in Section 7.5.1.

The importance density $g(\cdot)$ of (7.10) is constructed using the $N_{\mathrm{C}} = 200$ "most critical" scenarios of the crude Monte Carlo simulation: the scenarios with the lowest TTC. If applicable, the driver reaction time is also part of the multivariate pdf $g(\cdot)$. Note that these N_{C} "most critical" scenarios always include scenarios that ended with a crash because $\mu_{\mathrm{MC}} < N_{\mathrm{C}}/N_{\mathrm{MC}} = 0.02$.

[3] https://github.com/ErwindeGelder/ScenarioRiskQuantification

7.5.6. Probability of an injury

For the calculation of the injury rate, see (7.11), it has been assumed that the probability of an injury given a parameterized scenario, $\mathbb{P}(I(x_S))$, is known. The presented case study uses the model from Kusano and Gabler [174] to determine $\mathbb{P}(I(x_S))$, since this model is also used in [175, 323]. Kusano and Gabler [174] model the relationship between impact velocity change and belt usage with the probability of an injury with MAIS≥2 in rear-end crashes:

$$\mathbb{P}(I(x_S)) = \frac{1}{1 + \exp\{-(\beta_0 + \beta_1 \Delta v(x_S) + \beta_2 b)\}}. \tag{7.33}$$

Here, $\beta_0 = -6.068$, $\beta_1 = 0.100\,\mathrm{s/m}$, and $\beta_2 = 0.6234$ are parameters that are fitted to data of 1406 rear-end crashes [174]. For the velocity change during a crash, $\Delta v(x_S)$, which depends on the masses of the two objects colliding, we assume equal masses such that $\Delta v(x_S)$ is half of the impact speed (i.e., the speed difference at the start of the crash). If the belt is not used, then $b = -1$. In this case study, it is assumed that the belt is always used, so $b = 1$.

7.6. Results

This section provides the results of the case study described in Section 7.5. First, the exposures of the scenarios are listed. Next, the severity and controllability are reported. Finally, we explain the results of the simulations that include the triggering conditions.

7.6.1. Exposure

The bar graph in Figure 7.5 shows the estimated probability that the respective scenario is found n times per hour of driving. Given the number of encounters of the scenarios, we have the following exposures:

$$\mathrm{Exposure}(\mathcal{C}_{\mathrm{LVD}}) = \mathbb{E}\left[n_{\mathcal{C}_{\mathrm{LVD}}}\right] = 20.6\,\mathrm{h}^{-1}, \tag{7.34}$$

$$\mathrm{Exposure}(\mathcal{C}_{\mathrm{cut-in}}) = \mathbb{E}\left[n_{\mathcal{C}_{\mathrm{cut-in}}}\right] = 4.71\,\mathrm{h}^{-1}, \tag{7.35}$$

$$\mathrm{Exposure}(\mathcal{C}_{\mathrm{ASV}}) = \mathbb{E}\left[n_{\mathcal{C}_{\mathrm{ASV}}}\right] = 4.62\,\mathrm{h}^{-1}. \tag{7.36}$$

As mentioned in Remark 7.2, it is not uncommon to assume that the probability of encountering n scenarios belonging to a specific scenario category is Poisson-distributed. To compare the results with the Poisson distribution of (7.2), the probability of the Poisson distribution is also shown in Figure 7.5. The parameter λ is estimated by maximizing the likelihood. This gives the same results as (7.34) to (7.36), i.e., $\lambda = 20.6\,\mathrm{h}^{-1}$ for the LVD scenarios, $\lambda = 4.71\,\mathrm{h}^{-1}$ for the cut-in scenarios, and $\lambda = 4.62\,\mathrm{h}^{-1}$ for the ASV scenarios. According to the Chi-square goodness of test, the probability that $\mathbb{P}(n_{\mathcal{C}_{\mathrm{LVD}}} = n)$, $\mathbb{P}(n_{\mathcal{C}_{\mathrm{cut-in}}} = n)$, and $\mathbb{P}(n_{\mathcal{C}_{\mathrm{ASV}}} = n)$ are distributed according to the Poisson distribution is very small ($p < 0.001$). Hence, it can be concluded that the assumption that the number of encounters of the LVD, cut-in, and ASV scenarios is Poisson-distributed is unrealistic.

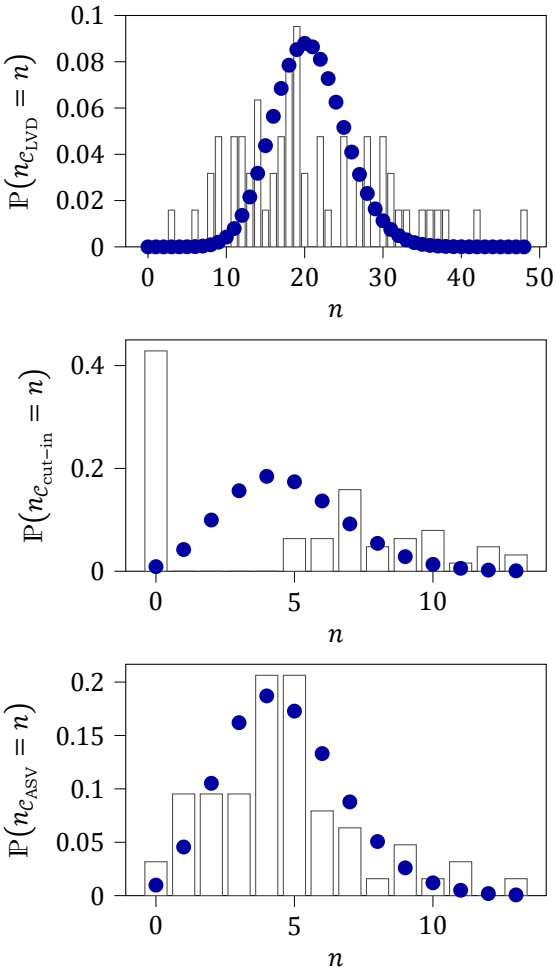


Figure 7.5: The bar graph shows how often a scenario of scenario category $\mathcal{C}_{\text{LVD}}$, $\mathcal{C}_{\text{cut-in}}$, or $\mathcal{C}_{\text{ASV}}$ is encountered n times per hour of driving divided over the total number of hours, 63. For comparison, the dots denote the Poisson distribution of (7.2) with the maximum likelihood estimate of λ.

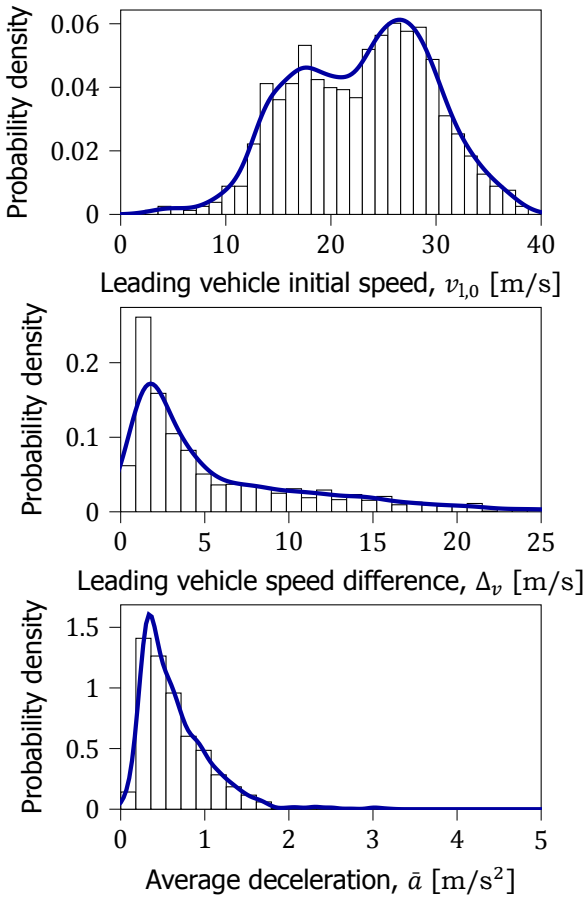


Figure 7.6: Three parameters of the LVD scenarios. The histograms show the original data and the solid lines represent the marginal probability distributions of the estimated pdf.

7.6.2. Severity and controllability

As explained in Section 7.3.4, to estimate the probability of a crash, the pdfs of the scenario parameters need to be estimated. In Figures 7.6 to 7.8, the results of the pdf estimation are shown. The histograms show the original data that are used to estimate the pdfs. The multivariate pdfs are estimated using KDE, see (7.3). To account for the infinite support of the Gaussian kernel of (7.4), the resulting pdfs are set to 0 if the parameters are outside the valid range of parameters as mentioned in Section 7.5.2. Next, the pdfs are scaled such that they integrate to 1. The solid lines in Figures 7.6 to 7.8 represent the marginal distributions of the resulting pdfs.

Tables 7.5 and 7.6 show the results of the simulation runs. It shows that the estimated probability of an injury with MAIS≥2 in LVD scenarios is $2.77 \cdot 10^{-7}$ with an estimated uncertainty of $1.27 \cdot 10^{-7}$. If, however, there would be no backup from an

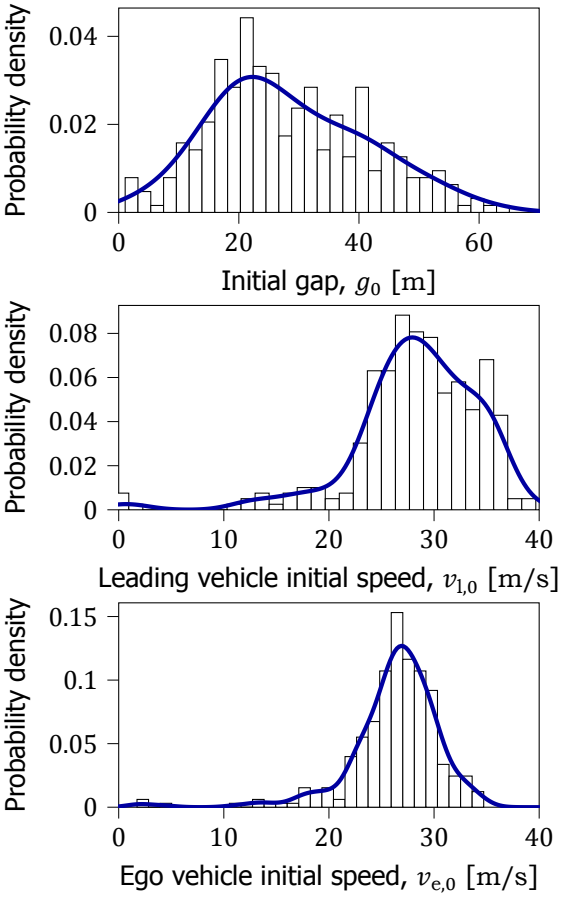


Figure 7.7: Three parameters of the cut-in scenarios. The histograms show the original data and the solid lines represent the marginal probability distributions of the estimated pdf.

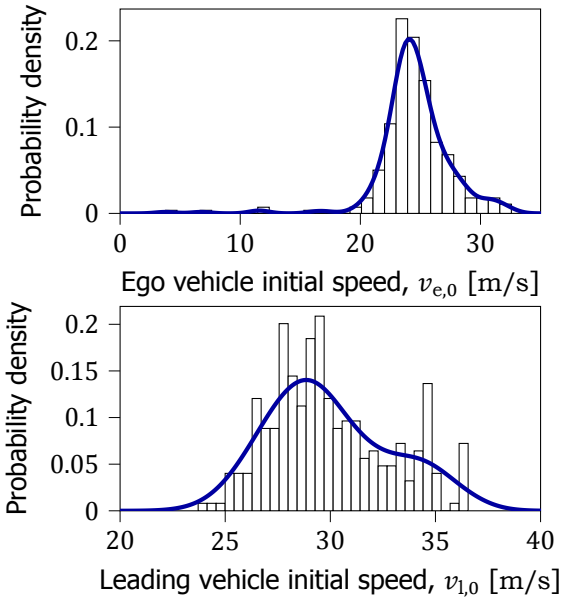


Figure 7.8: Two parameters of the ASV scenarios. The histograms show the original data and the solid lines represent the marginal probability distributions of the estimated pdf.

operator, then this probability is substantially higher: $\text{Severity}(\mathcal{C}_{\text{LVD}}) = 1.12 \cdot 10^{-5}$. Therefore, the controllability score is small ($\text{Controllability}(\mathcal{C}_{\text{LVD}}) = 2.47 \cdot 10^{-2}$), resulting in $\text{Risk}(\mathcal{C}_{\text{LVD}}) = 5.72 \cdot 10^{-6}\,\text{h}^{-1}$. In other words, it is expected that, on average, one crash in an LVD scenario with a moderate injury or worse happens in $1.75 \cdot 10^{5}\,\text{h}$ of driving.

For the cut-in scenario category, the controllability score is almost 1, indicating that it is unlikely that a human driver can avoid any potential harm. Although the severity and exposure are lower than for the LVD scenario category, the higher controllability score results in a higher overall risk ($\text{Risk}(\mathcal{C}_{\text{cut-in}}) = 1.75 \cdot 10^{-5}\,\text{h}^{-1}$).

Out of the three scenario categories, the ASV scenario category has the highest severity ($\text{Severity}(\mathcal{C}_{\text{ASV}}) = 2.12 \cdot 10^{-5}$). According to the simulations, however, the human driver is able to avoid any harm in almost all cases. Therefore, the controllability score is low ($\text{Controllability}(\mathcal{C}_{\text{ASV}}) = 7.83 \cdot 10^{-6}$) and the estimated risk is also low ($\text{Risk}(\mathcal{C}_{\text{ASV}}) = 7.68 \cdot 10^{-10}\,\text{h}^{-1}$). Note that for the ASV scenario category, σ_{injury} is almost equal to μ_{injury}, indicating that the relative uncertainty is high. This indicates that the actual risk could be an order of magnitude lower than the estimated risk.

7.6.3. Triggering conditions

Tables 7.5 and 7.6 also show the results of the simulations that include a triggering condition. The risk in LVD scenarios is approximately 50 times higher when considering the triggering condition "limited braking capacity"; so this triggering condition

Table 7.5: First part of the results of the case study. μ_{NIS} of (7.8): estimated crash probability. σ_{NIS} of (7.9): estimated standard deviation of μ_{NIS}. μ_{injury} of (7.11): estimated probability of an injury with MAIS≥2. σ_{injury} of (7.13): estimated standard deviation of μ_{NIS}.

$\mathcal{C}$	Trig. cond.	μ_{NIS}	σ_{NIS}	μ_{injury}	σ_{injury}
$\mathcal{C}_{\text{LVD}}$	None	$1.61 \cdot 10^{-4}$	$5.95 \cdot 10^{-5}$	$2.77 \cdot 10^{-7}$	$1.27 \cdot 10^{-7}$
$\mathcal{C}_{\text{LVD}}$	Lim. braking	$7.22 \cdot 10^{-3}$	$1.58 \cdot 10^{-4}$	$1.16 \cdot 10^{-5}$	$2.70 \cdot 10^{-7}$
$\mathcal{C}_{\text{LVD}}$	Poor visibility	$1.61 \cdot 10^{-4}$	$5.95 \cdot 10^{-5}$	$2.77 \cdot 10^{-7}$	$1.27 \cdot 10^{-7}$
$\mathcal{C}_{\text{cut-in}}$	None	$1.95 \cdot 10^{-3}$	$1.32 \cdot 10^{-4}$	$3.71 \cdot 10^{-6}$	$2.71 \cdot 10^{-7}$
$\mathcal{C}_{\text{cut-in}}$	Lim. braking	$2.20 \cdot 10^{-3}$	$1.35 \cdot 10^{-4}$	$4.79 \cdot 10^{-6}$	$3.23 \cdot 10^{-7}$
$\mathcal{C}_{\text{cut-in}}$	Poor visibility	$1.95 \cdot 10^{-3}$	$1.32 \cdot 10^{-4}$	$3.71 \cdot 10^{-6}$	$2.71 \cdot 10^{-7}$
$\mathcal{C}_{\text{ASV}}$	None	$1.03 \cdot 10^{-7}$	$1.02 \cdot 10^{-7}$	$1.66 \cdot 10^{-10}$	$1.64 \cdot 10^{-10}$
$\mathcal{C}_{\text{ASV}}$	Lim. braking	$4.51 \cdot 10^{-3}$	$1.72 \cdot 10^{-4}$	$9.78 \cdot 10^{-6}$	$4.15 \cdot 10^{-7}$
$\mathcal{C}_{\text{ASV}}$	Poor visibility	$3.57 \cdot 10^{-3}$	$1.41 \cdot 10^{-4}$	$1.01 \cdot 10^{-5}$	$3.97 \cdot 10^{-7}$

Table 7.6: Second part of the results of the case study. Severity: estimation of (7.18). Controllability: estimation of (7.19). Risk: estimation of (7.20). Note that for the risk estimation, the exposure of the triggering condition is not considered.

$\mathcal{C}$	Trig. cond.	Severity	Controllability	Risk
$\mathcal{C}_{\text{LVD}}$	None	$1.12 \cdot 10^{-5}$	$2.47 \cdot 10^{-2}$	$5.72 \cdot 10^{-6}\,\text{h}^{-1}$
$\mathcal{C}_{\text{LVD}}$	Limited braking	$1.52 \cdot 10^{-5}$	0.763	$2.39 \cdot 10^{-4}\,\text{h}^{-1}$
$\mathcal{C}_{\text{LVD}}$	Poor visibility	$1.12 \cdot 10^{-5}$	$2.47 \cdot 10^{-2}$	$5.72 \cdot 10^{-6}\,\text{h}^{-1}$
$\mathcal{C}_{\text{cut-in}}$	None	$3.92 \cdot 10^{-6}$	0.947	$1.75 \cdot 10^{-5}\,\text{h}^{-1}$
$\mathcal{C}_{\text{cut-in}}$	Limited braking	$4.54 \cdot 10^{-6}$	1.06	$2.26 \cdot 10^{-5}\,\text{h}^{-1}$
$\mathcal{C}_{\text{cut-in}}$	Poor visibility	$3.92 \cdot 10^{-6}$	0.947	$1.75 \cdot 10^{-5}\,\text{h}^{-1}$
$\mathcal{C}_{\text{ASV}}$	None	$2.12 \cdot 10^{-5}$	$7.83 \cdot 10^{-6}$	$7.68 \cdot 10^{-10}\,\text{h}^{-1}$
$\mathcal{C}_{\text{ASV}}$	Limited braking	$3.70 \cdot 10^{-5}$	0.265	$4.52 \cdot 10^{-5}\,\text{h}^{-1}$
$\mathcal{C}_{\text{ASV}}$	Poor visibility	$2.12 \cdot 10^{-5}$	0.476	$4.66 \cdot 10^{-5}\,\text{h}^{-1}$

has a significant impact on the safety during such scenarios. For cut-in scenarios, the risk is of the same order of magnitude when considering the triggering condition "limited braking capacity". The estimated risk in ASV scenarios is the highest when considering a limited braking capacity.

The triggering condition "poor visibility" does not influence the risk in LVD scenarios. This is because the leading vehicle is always within the viewing range of the driver, i.e., $s_0 + v_e \tau_h < d_{\text{view}}$. The risk in cut-in scenarios is also not influenced by this triggering condition. It may be possible that the vehicle cutting in is at a larger distance than d_{view}, but in potential risky scenarios that may result in harm, the vehicle cutting in is at a smaller distance than d_{view}. Hence, the vehicle cutting in is still in the limited viewing range of the human driver. For ASV scenarios, the risk is significantly higher when considering the triggering condition "poor visibility". The severity is the same because the limited viewing range of the human driver does not affect the ACC, but the controllability score is significantly higher. This indicates

that it is less likely that the human driver will prevent harm in ASV scenarios when its view toward the leading vehicle is limited.

7.7. Discussion

Quantifying the risk in driving scenarios is an important component of the overall risk assessment of an ADS. Research question 7.1 addresses this by asking how to quantify the risk of an ADS. The proposed method in Section 7.3 answers this question. With answering Research question 7.2, we have also shown how the proposed method for risk quantification contributes to the risk assessment as proposed in the ISO 26262 [144] and ISO 21448 [143] standards by decomposing the risk into the components exposure, severity, and controllability. This section provides further interpretations of the results and discusses limitations of the presented research that are to be addressed in future research.

Whereas the ISO 26262 standard addresses functional safety and the ISO 21448 standard addressed SOTIF, for the safe deployment and operation of an ADS, cybersecurity [61] needs to be addressed as well. The cybersecurity ensures that an ADS is well protected against security attacks [62, 246]. This includes, e.g., mitigating privacy-related risks [160, 187, 297, 309–311, 326]. Note that cybersecurity is out-of-scope of the ISO 26262 and ISO 21448 standards and, thus, it is also out-of-scope of the current work. We refer to the ISO 21434 standard [142] for more information regarding cybersecurity of ADSs.

The proposed risk quantification serves two purposes. One purpose is to support the evaluation of whether it is safe to actually deploy an ADS in real-world traffic. In fact, the output of the proposed method, i.e., the expected number of injuries per hour of driving, can be compared with road crash statistics. Another purpose is to facilitate the design decisions during the development of an ADS. For example, based on the high controllability score of the ACC in ASV scenarios with poor visibility conditions (Table 7.6), it might be decided that a subsystem needs to be in place to detect poor visibility conditions such that the speed is reduced under these conditions. Also, the severity score might be lowered by adapting the control logic of the ADS.

The exposure of a scenario category is estimated by counting the number of occurrences of the corresponding scenarios and dividing this number by the number of hours of driving. As pointed out in Remark 7.2, it is not uncommon to assume that the rate of occurrence is constant such that the occurrence is Poisson-distributed. Looking at the results in Figure 7.5, however, the data do not support the assumption that the occurrence is Poisson-distributed. This suggests that the rate of occurrence is not constant and depends on other factors, e.g., the road layout, the environment (a cut-in is less likely on a rural road than on the highway), the driver, and the time of the day. If such factors are different during the deployment of the ADS, the estimated exposure needs to be reconsidered.

The presented method for risk quantification comes with limitations that are to be addressed in future research. While we advertised the use of data such that the risk estimation relies less on possibly subjective opinions from safety experts,

it might be difficult to justify the adequacy of the data. First, it is important that we have enough data to accurately determine the pdf of the scenario parameters[4]. Second, the data need to match the Operational Design Domain (ODD) (see Table 7.2 for the definition) of the ADS. For example, the data that have been used in the case study were obtained from driving a specific route multiple times during daytime and good weather conditions. If the ODD of the ADS considers the same route during daytime and good weather conditions, then the data are representing this ODD. If, however, the ODD is substantially different, extra arguments are needed to justify that the data still represent the ODD. The estimated exposure of the scenarios and the estimated parameter pdfs might not be accurate for the specified ODD in case the data have been recorded under different circumstances. As a result, the estimated risk might not be accurate enough.

Another difficulty involves accounting for the exposure of the triggering conditions. If the occurrence rate of the triggering condition can be assumed to be independent of the scenario category, the calculated risk of (7.20) — which does not consider the exposure of the triggering condition — can simply be multiplied with the estimated exposure of the triggering condition. It may be difficult, however, to justify this independence. More research is needed to investigate the possible triggering conditions, their occurrence rates, their relations with the occurrence of different scenarios, and the (possible) dependency between the scenario parameter values and the triggering conditions.

The proposed method for risk quantification employs simulations of the ADS response in driving scenarios. As a proof of concept, we have implemented simulations using simple models for the system-under-test and the driver behavior model. On the one hand, using these simple models contributes to the explainability of the results, ensures short simulation run times, and facilitates the reproducibility of the case study. On the other hand, the fidelity of the simulation results may be compromised by the simplicity of the simulations. When using the proposed method to assess the risk of deploying an ADS in the real world, evidence is needed to justify the fidelity of the simulation results. Therefore, the development of high-fidelity simulators for ADSs is an important research topic; see [158, 238] for an overview. More research is needed to actually verify the fidelity of such simulators. Note that it might be possible to combine virtual simulations with, e.g., hardware-in-the-loop tests, vehicle-in-the-loop tests, and proving ground tests [232]. For example, proving ground tests may be used to verify the virtual simulation results and virtual simulations are used to alleviate the required resources as no longer all tests have to be performed physically.

The proposed risk quantification is performed with regards to a given scenario category, possibly including one or more triggering conditions. To thoroughly review the risk of deploying an ADS in real-world traffic, many scenario categories and triggering conditions need to be considered. It remains an open question how many scenario categories and triggering conditions are to be considered. The 67 scenario categories described in [74] might be a good starting point, but it is expected that

[4]To determine whether enough data have been collected to estimate the pdf accurately, the metric proposed in Chapter 3 [72] can be used.

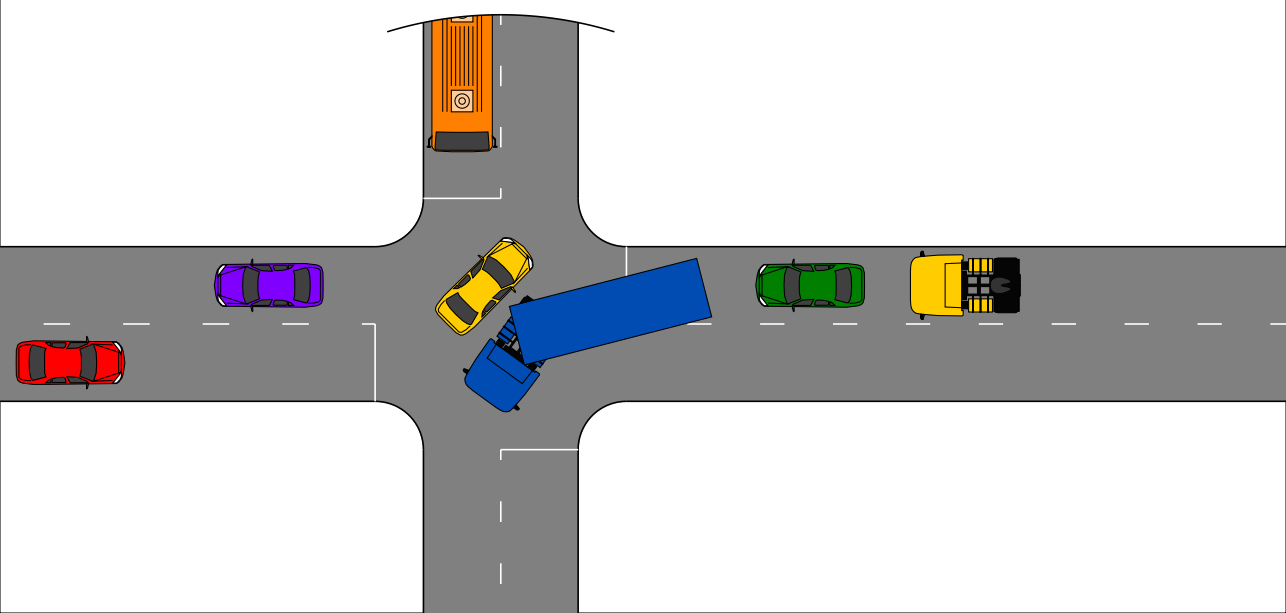
more scenario categories are needed for an ADS that aims for deployment in a complex ODD. The ISO/CD 34502 standard [148], currently under development, provides a process to determine the relevant scenario categories and triggering conditions for the safety validation of an ADS that is active on highways. Thus, once published, this standard may help to answer the question of which and how many scenario categories and triggering conditions are needed for a thorough risk assessment.

7.8. Conclusions

Validating the safety of an Automated Driving System (ADS) is essential to enable the safe deployment of an ADS. Part of the safety validation is the estimation of the risk of an ADS when dealing with real-world driving scenarios. The current work has presented a method to quantify this risk given a set of driving scenarios that are comprised by a scenario category. Since a data-driven approach has been proposed, the method relies less on the possibly subjective inputs from safety experts. Simulations are employed such that risks can be assessed prospectively, i.e., before real-life testing. The method supports the decomposition of the risk into the three aspects of risk, i.e., exposure, severity, and controllability, as mentioned by the leading standards in automotive safety, i.e., the ISO 26262 and ISO 21448 standards. Thus, the current chapter provides a method that can help engineers when designing an ADS in compliance with these standards and when evaluating the compliance of the ADS to these standards.

The risk quantification method has been illustrated by means of a case study. In the case study, the risk of a moderate injury or worse per hour of driving of an ADS is estimated for three scenario categories: leading vehicle decelerating (LVD), cut-in, and approaching slower vehicle (ASV). In addition to these scenarios, the scenarios have been complemented with so-called triggering conditions, such as poor visibility conditions. The results have indicated, e.g., that the role of a human driver as a back-up is essential to reduce the risk in ASV scenarios and that, therefore, conditions that cause a poor visibility of the driver's environment have a significant impact on the risk.

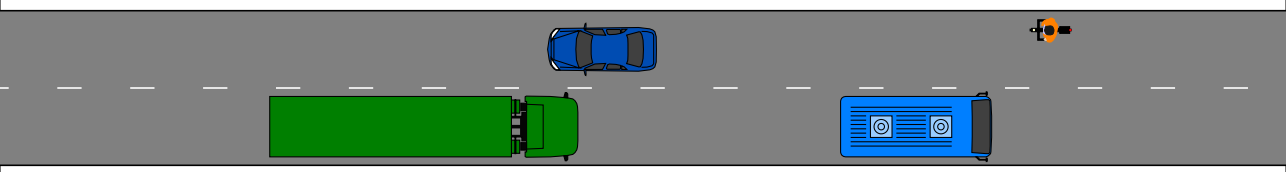
Future work involves determining the statistics of the triggering conditions and the dependencies between triggering conditions and the different scenarios. Furthermore, verification methods for the fidelity of the simulation results is a topic of ongoing research. Finally, more research is needed to determine the relevant scenario categories that are to be considered for a full risk assessment of an ADS.



8

Probabilistic risk measure derivation method

This chapter is based on:
E. de Gelder, K. Adjenughwure, J. Manders, R. Snijders, J.-P. Paardekooper, O. Op den Camp, A. Tejada, and B. De Schutter, *PRISMA: A novel approach for deriving probabilistic surrogate safety measures for risk evaluation,* Under review (2022).



Surrogate Safety Measures (SSMs) are used to express road safety in terms of the safety risk in traffic conflicts. Typically, SSMs rely on assumptions regarding the future evolution of traffic participant trajectories to generate a measure of risk. As a result, they are only applicable in scenarios where those assumptions hold. To address this issue, we present a novel data-driven Probabilistic RISk Measure derivAtion (PRISMA) method. The PRISMA method is used to derive SSMs that can be used to calculate in real time the probability of a specific event (e.g., a crash). Because we adopt a data-driven approach to predict the possible future evolutions of traffic participant trajectories, less assumptions on these trajectories are needed. Since the PRISMA is not bound to specific assumptions, multiple SSMs for different types of scenarios can be derived. To calculate the probability of the specific event, the PRISMA method uses Monte Carlo simulations to estimate the occurrence probability of the specified event. We further introduce a statistical method that requires fewer simulations to estimate this probability. Combined with a regression model, this enables our derived SSMs to make real-time risk estimations.

To illustrate the PRISMA method, an SSM is derived for risk evaluation during longitudinal traffic interactions. Since there is no known method to objectively estimate risk from first principles, i.e., there is no known risk ground truth, it is very difficult, if not impossible, to objectively compare the relative merits of two SSMs. Instead, we provide a method for benchmarking our derived SSM with respect to expected risk trends. The application of the benchmarking illustrates that the SSM matches the expected risk trends.

Whereas the derived SSM shows the potential of the PRISMA method, future work involves applying the approach for other types of traffic conflicts, such as lateral traffic conflicts or interactions with vulnerable road users.

This chapter is based on [77].

8.1. Introduction

Road safety is an important key performance indicator in transportation. In addition to the suffering of people as a consequence of crashes in traffic, these crashes cause enormous societal and economic losses. As a result, road safety research is an important research topic. For example, in 2018[1], there were over 6.7 million crashes in the U.S.A. [210], which is about 1.3 crashes per 1 million vehicle kilometers driven. These crashes in 2018 led to 2.7 million injured people and 37 thousand fatalities [210]. Furthermore, apart from these societal losses, the economic costs of all crashes in the U.S.A. in 2018 was 242 billion dollars [210]. Similarly, the European Commission [96] reported over 22 thousand fatalities in 2019.

Road safety can be expressed in terms of injuries, fatalities, or crashes per kilometer of driving, but "that is a slow, reactive process" [17]. Furthermore, "crashes are rare events and historical crash data does not capture near crashes that are also critical for improving safety" [296]. An alternative for expressing road safety that does not rely on historical crash data is the use of safety indicators that directly measure the safety risk in traffic conflicts [17, 275, 296]. Traffic conflicts are far more frequent than traffic crashes and the frequency of traffic conflicts can be used to predict the frequency of crashes [65, 274]. To define traffic conflicts, thresholds on so-called Surrogate Safety Measures (SSMs) are used, where SSMs characterize the risk of a crash or harm given an initial condition [17]. SSMs vary from measures that estimate the remaining time until a crash, such as the well-known Time to Collision (TTC) [130], to metrics that estimate the probability that a human driver cannot avoid a crash, see, e.g., [295].

SSMs typically rely on assumptions of what drivers or systems controlling the vehicles of interest are capable of doing and how their future trajectories — given an initial condition — will develop. For example, TTC [130], the ratio of the distance toward and the speed difference with an approaching object, is computed by assuming a constant relative velocity. As a result of these assumptions, SSMs are only applicable in certain types of scenarios. For example, TTC is only applicable when approaching an object. More complex SSMs consider, e.g., a human model that can react to a risky situation by braking [295] or the uncertainty over the future ambient traffic state [202]. Regardless of the complexity of these models, however, these SSMs consider neither the specific capabilities of the driver or of the system controlling the vehicle, nor the local context for predicting the future of the vehicle's environment.

This chapter presents the novel Probabilistic RISk Measure derivAtion (PRISMA) method, which is a data-driven approach for deriving SSMs that are not limited to certain types of scenarios. Because the method is not bound to certain predetermined assumptions about driver behavior, the derived SSMs can be adapted to the situations in which they are applied. In addition, to avoid relying on predetermined assumptions on how the ambient traffic evolves over time, the PRISMA method includes a data-driven approach for modeling the variations of the trajectories of

[1] At the time of writing, more recent results were not yet available.

the ambient traffic. Monte Carlo simulations are employed to accurately predict the safety risk given these variations. To enable the real-time evaluation of the derived SSMs, we use the Nadaraya-Watson (NW) kernel estimator [300] for local regression. The PRISMA method has the following characteristics:

- The derived SSMs give a probability that a specified event, e.g., a crash or a near miss will happen in the near future, e.g., within the next 10 seconds, given an initial state and the foreseen evolutions of traffic participant trajectories. Since a traffic conflict can be defined as the probability of an unsuccessful evasion in a traffic interaction, according to Davis *et al.* [65], a probability is easier to interpret than, e.g., a value ranging from zero to infinity such as the TTC.

- Next to deriving new SSMs, i.e., new ways to estimate the probability of an event such as a crash, it is possible to reproduce already existing measures that provide a probability. Therefore, the PRISMA method can be seen as a generalization for deriving such existing SSMs.

- A driver behavior model can be used. It is also possible to use a model of an Automated Driving System (ADS), such that the derived SSM estimates the safety risk if this ADS controls the vehicle.

- Because a data-driven approach is adopted, the derived SSM adapts to the recorded data. In this way, it is possible to adapt the SSM to, e.g., the local traffic behavior provided that this local traffic behavior is captured by the recorded data.

- The PRISMA method is not limited to one type of scenario.

We illustrate the PRISMA method and its benefits by means of a case study. The case study demonstrates that when using the PRISMA method with the assumptions of the SSM of Wang and Stamatiadis [295], both the SSM derived by the PRISMA method and the latter yield the same result. The case study continues with evaluating the crash risk of three longitudinal traffic conflicts which are a priori known to be, respectively, safe (i.e., no crash possible), moderately safe, and unsafe (i.e., a crash occurs), based on vehicle kinematics. We evaluate the risk of each of the scenarios using the SSM by Wang and Stamatiadis [295] and an SSM derived by the PRISMA method, based on data from the Next Generation SIMulation (NGSIM) [7]. Moreover, since a comparison between these measures is not directly possible in general scenarios, a method to benchmark SSMs using expected risk trends is introduced in the case study.

This chapter is organized as follows. Section 8.2 provides an overview of SSMs described in the literature. The proposed PRISMA method is presented in Section 8.3. In Section 8.4, we illustrate the method in a case study. Some implications of this work are discussed in Section 8.5. The chapter is concluded in Section 8.6.

8.2. Literature review

Risk in the context of traffic safety is often defined as the probability of an accident occurring [126]. Most SSMs are derived under specific assumptions of the expected behavior of the driving participants under a specific driving scenario. Several SSMs have been developed under such assumptions with the goal of quantifying the risk involved in driving in traffic on the road [63, 178, 200, 220]. In general, the risk is quantified in terms of the proximity between two traffic agents in time and/or space, the ability to perform evasive actions like braking or swerving, or the magnitude of such actions [260, 324]. In a potential crash situation, the proximity indicator is close to zero while the magnitude of evasive action is close to the limits of the driver and the vehicle [324]. The above clustering of SSMs in terms of time, space, and evasive action is common in the literature; so our literature review follows this pattern of clustering SSMs. We focus on the most commonly used measures in each cluster and their underlying assumptions.

The most common SSMs are time based. A popular time-based proximity indicator is the TTC, which is an estimate of the remaining time until two vehicles collide and is defined as the time remaining until two vehicles collide if they would continue on the same course and speed [130]. The assumption for the TTC is that the relative speed and course will remain the same. In addition, the TTC is only relevant when two objects are approaching each other. These assumptions make it difficult to use it for various driving scenarios. Several other time-based SSMs have been derived from or based on the TTC. Notable among those are:

- the time-exposed TTC, which measures the amount of time the TTC is below a certain threshold [200];
- the Time Integrated TTC (TIT), which calculates the total area in a TTC versus time diagram where the TTC is below a certain threshold [200]; and
- the Modified TTC (MTTC), which is able to calculate the TTC for cases where vehicles do not keep a constant speed and the follower is slower than the leader [220].

For the MTTC, the relative speed is not assumed to be constant, but new assumptions on the acceleration and speed of the objects are introduced.

Other time-based proximity indicators include Post-Encroachment Time (PET), which measures the "time between the moment that a vehicle leaves the area of potential collision, i.e., the area in which the paths of the two vehicles intersect, and the other vehicle arrives in the same area" [192] and Time Headway (THW). The PET can only be calculated when the collision area of the two participants is known. This assumption makes it mostly useful for scenarios with obvious crossing conflicts like intersections.

For distance-based proximity indicators, the Potential Index for Collision with Urgent Deceleration (PICUD) measures the remaining distance between vehicles during an emergency stop [141, 285] and the Proportion of Stopping Distance (PSD) measures the remaining distance to the potential point of collision divided by the minimum acceptable stopping distance [9, 123, 192]. These two measures assume

that the vehicles will apply the maximum deceleration during emergency situations. This makes them suitable for emergency situations for which these assumption will most likely hold. For non-critical situations, however, the deceleration that the drivers will apply, may vary. More recently, a distance-based measure that assumes "correct" driving behavior has been proposed [258]. This measure calculates the minimum safety distance between a follower and its leader, such that no crash occurs if the leader vehicle brakes with a specified deceleration and the follower brakes after a specified reaction time with another specified deceleration. Based on the definition of this measure, it is not suitable for driving situations where the driver does not follow the description of "correct" driving given above.

In terms of indicators relating to performing evasive actions, the Deceleration Rate to Avoid Collision (DRAC) is the most widely used. The DRAC is calculated as the ratio of the difference in speed between a following vehicle and a leading vehicle and their closing time [10, 192]. Another indicator is the Crash Potential Index (CPI), which calculates the probability that a vehicle's DRAC will exceed its Maximum Available Deceleration Rate (MADR) in a given time interval [63]. The DRAC is not a risk measure on its own if it is not compared with the braking capacity. This is a limitation and this is why the CPI measure has been developed. Both DRAC and CPI are mostly suitable for a car-following situation and are not suitable for lateral movements [192].

Wang and Stamatiadis [295] derive a "crash propensity metric" using a combination of the TTC, the vehicle braking capability, and the driver's reaction time. Although this metric is suitable for various car-following situations, lane-change conflicts, and crossing conflicts, it is limited because it uses TTC in its calculations, so this metric is undefined when the TTC is undefined. In addition, it assumes that the driver will keep a constant speed before reacting and, after a reaction time, the driver will apply the maximum deceleration.

Shi *et al.* [260] use indicators like TIT, CPI, and PSD to measure the effectiveness of risk indicators for predicting accidents. The idea is to use a combination of indicators and thresholds on the indicators to predict whether an interaction may become a crash. This results in new indicators, but they inherit the union of the assumptions of the other indicators. Mullakkal-Babu *et al.* [202] propose a probabilistic driving risk field. The method derives the risk a vehicle is exposed to using a kinematic approach with the inclusion of uncertainty in the vehicle's future state. Mullakkal-Babu *et al.* [202] define this for an encounter between the ego vehicle and a road obstacle, such as other vehicles or objects. This research shares similar ideas with our proposed method of risk estimation, but Mullakkal-Babu *et al.* [202] do not use a data-driven approach to derive the SSM. Furthermore, the future state of the vehicle is estimated with a fixed distribution (i.e., a normal distribution). This limits the application in scenarios where the data may have an entirely different distribution.

To estimate crash probabilities based on existing SSMs, the probabilistic approach using the Extreme Value Theory (EVT) has been applied successfully [267, 273, 296]. For example, based on a specific TTC value, EVT can be used to predict the probability of a crash. Using EVT, the crash probability is estimated by assum-

ing the generalized extreme value distribution or the generalized Pareto distribution and fitting the parameters of the distribution using the "block maxima" approach or the "peak over" approach, respectively [296]. It is also possible to combine multiple SSMs using the EVT. The advantage of EVT is that EVT provides probabilities that are directly linked to historical data and that these probabilities have been used successfully to predict the frequency of crashes [18, 267]. Disadvantages of EVT are that it might inherit the assumptions of the SSMs that it uses to estimate the crash probability and that it assumes a fixed distribution of the extreme events, which is only justified if many data are used. Furthermore, as the estimated crash probability is solely based on the fitted distribution, it does not consider potential changes to the driver's behavior (model).

8.3. Probabilistic RISk Measure derivAtion

In this section, we propose the PRISMA method which is a method for deriving a measure that quantifies the risk of a certain event, such as a crash, in a particular situation in which a vehicle - hereafter, the *ego vehicle* - is in and that is applicable for real-time use. The PRISMA method consists of four steps. The first step is the parameterization of the "initial situation" and the possible "future situations". Second, based on the initial situation, we estimate the probability (density) for the possible future situations. The third step includes determining the probability of the specified event based on the initial and the future situations. Finally, local regression is used to speed up the calculations and to make it possible to use the SSM in real time. These four steps are described in the following subsections.

In this chapter, the following notation is used. To denote a probability function, $\mathbb{P}(\cdot)$ is used. A probability density function (pdf) is denoted by $p(\cdot)$. The probability of u given v is denoted by $\mathbb{P}(u|v)$. Similarly, a conditional pdf is denoted by $p(\cdot|\cdot)$. To denote the estimation of $\mathbb{P}(u)$, $\mathbb{P}(u|v)$, $p(u)$, and $p(u|v)$, we use $\hat{P}(u)$, $\hat{P}(u|v)$, $\hat{p}(u)$, and $\hat{p}(u|v)$, respectively.

8.3.1. Parameterize initial and future situations

The first step is to parameterize the initial situation the ego vehicle is in. In other words, the initial situation needs to be described using n_x numbers that are stacked into one vector $x \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$. This vector contains all relevant aspects for determining the risk. As an example, x could contain the speed of the ego vehicle and the distance toward its preceding vehicle. In Section 8.4, we will consider more examples.

Next to describing the initial situation, the future situation is described using n_y numbers stacked into one vector $y \in \mathcal{Y} \subseteq \mathbb{R}^{n_y}$. Together with x, y contains enough information to describe how the relevant future, e.g., the next five seconds, around the ego vehicle develops over time. As an example, y could contain the speed for the next five seconds of the leading vehicle (if any) that is in front of the ego vehicle. In Section 8.4, we will consider more examples.

Let C denote an event, e.g., a crash or a near miss, such that the probability of this event is $\mathbb{P}(\mathrm{C})$. The goal of our SSM is to estimate the probability of the event

C given a particular situation x, i.e., $\mathbb{P}(\mathrm{C}|x)$. We do this by considering all future situations, $\mathcal{Y}$, and calculating the probability of the event C given each possible value of y. Using integration, we obtain $\mathbb{P}(\mathrm{C}|x)$:

$$\mathbb{P}(\mathrm{C}|x) = \int_y \mathbb{P}(\mathrm{C}|x,y)p(y|x)\,\mathrm{d}y. \tag{8.1}$$

8.3.2. Estimate $p(y|x)$

In this section, we propose a method to estimate $p(y|x)$, i.e., the pdf of y given x. Using the product rule for probability, we can write:

$$p(y|x) = \frac{p(x,y)}{p(x)} = \frac{p(x,y)}{\int_y p(x,y)\,\mathrm{d}y}. \tag{8.2}$$

Thus, it suffices to estimate $p(x,y)$.

Our proposal is to estimate $p(x,y)$ in a data-driven manner. A data-driven approach brings several benefits. First, the estimate automatically adapts to local driving styles and behaviors, which can change from region to region, provided that the data are obtained from the same local traffic. Second, assumptions such as a constant speed of other vehicles, are not needed. For our data-driven approach, let us assume that we have obtained N situations from data. For the i-th situation, we denote the initial situation and the future situation by $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, respectively. The remainder of this subsection describes how we estimate $p(x,y)$ using $\{(x_i,y_i)\}_{i=1}^N$.

Kernel density estimation

We first explain how to estimate $p(x,y)$ if we assume that all $n_x + n_y$ parameters depend on each other. If the shape of the pdf is known, a particular functional form can be fitted to the data, e.g., by estimating the parameters of a distribution by maximizing the likelihood. For example, if it is known that the data $\{(x_i,y_i)\}_{i=1}^N$ come from a multivariate normal distribution, it suffices to estimate the mean and the covariance. If, however, the shape is unknown, fitting a particular parametric distribution may lead to very inaccurate results [52]. Furthermore, the shape of the estimated pdf might change as more data are acquired. Assuming a functional form of the pdf and fitting the parameters of the pdf to the data may therefore lead to inaccurate fits unless extensive manual tuning is applied.

In the remainder of this work, we assume that the shape of the pdf $p(x,y)$ is unknown a priori. Therefore, we employ a non-parametric approach using Kernel Density Estimation (KDE) [222, 237] because the shape of the pdf is then automatically computed and KDE is highly flexible regarding the shape of the pdf. Using KDE, the estimated pdf becomes:

$$\hat{p}(x,y) = \frac{1}{N}\sum_{i=1}^{N} K_H\left(\begin{bmatrix}x\\y\end{bmatrix} - \begin{bmatrix}x_i\\y_i\end{bmatrix}\right), \tag{8.3}$$

where $K_H(\cdot)$ is an appropriate kernel function with an $(n_x + n_y)$-by-$(n_x + n_y)$ symmetric positive definite *bandwidth* or *smoothing* matrix H. The choice of the kernel $K_H(\cdot)$ is not as important as the choice of the bandwidth matrix H [283]. We use the often-used Gaussian kernel [87]:

$$K_H(u) = \frac{1}{(2\pi)^{(n_x+n_y)/2}|H|^{1/2}} \exp\left\{-\frac{1}{2}u^\mathsf{T}H^{-1}u\right\}. \tag{8.4}$$

The bandwidth matrix H controls the width of the kernel, or, in other words, the influence of each data point (i.e., $\begin{bmatrix}x_i^\mathsf{T} & y_i^\mathsf{T}\end{bmatrix}^\mathsf{T}$) on nearby regions (see [294] for a more extensive explanation of the bandwidth matrix). There are many different ways of estimating the bandwidth matrix, ranging from simple reference rules like, e.g., Silverman's rule of thumb [262] to more elaborate methods; see [24, 55, 151, 283, 319] for reviews of different bandwidth selection methods.

Drawing samples from the estimated pdf in (8.3) is straightforward: two random numbers are drawn, one to choose a random generator kernel out of the N kernels that are used to construct the KDE, and one random number from that kernel. Sampling from $\hat{p}(y|x)$ works similarly, but instead of using an equal probability for each random generator kernel to be selected, different probabilities are used based on x. For more information on sampling from a conditional pdf obtained using KDE, see Chapter 6 [75] and [134].

Assuming independence

Due to the curse of dimensionality [254], estimating $p(x, y)$ with one KDE according to (8.3) becomes inaccurate if $n_x + n_y$ becomes large. One option to avoid this curse of dimensionality is to assume that one or more parameters are independent of the other parameters. E.g., suppose that $y^\mathsf{T} = \begin{bmatrix}\bar{y}^\mathsf{T} & \tilde{y}^\mathsf{T}\end{bmatrix}$, such that $\tilde{y}$ is independent of x and $\bar{y}$. Then we can write

$$p(x, y) = p(x, \bar{y}, \tilde{y}) = p(x, \bar{y})p(\tilde{y}). \tag{8.5}$$

In this case, we would need to estimate $p(x, \bar{y})$ and $p(\tilde{y})$, which can be done in a similar manner as presented in (8.3). Because these two pdfs have fewer variables than $p(x, y)$, the two estimated pdfs will suffer less from the curse of dimensionality [254].

Another option is to model $p(y|x)$ as a cascade of conditional probabilities. For example, using the partitioning $y^\mathsf{T} = \begin{bmatrix}\bar{y}^\mathsf{T} & \tilde{y}^\mathsf{T}\end{bmatrix}$, $p(x|y)$ can be approximated using two conditional densities:

$$p(y|x) = p(\bar{y}, \tilde{y}|x) = p(\bar{y}|\tilde{y}, x)p(\tilde{y}|x) \approx p(\bar{y}|\tilde{y})p(\tilde{y}|x). \tag{8.6}$$

This approximation is valid if $\bar{y}$ and x are *conditionally independent given* $\tilde{y}$ [208]. The same partitioning can be applied to $p(\bar{y}|\tilde{y})$ and $p(\tilde{y}|x)$ to the point that only two-dimensional pdfs need to be estimated. Although this will lead to larger approximation errors, the lower-dimensional pdfs can be estimated more accurately. For more information on this approach, we refer the reader to [2, 208].

Reduce number of parameters using singular value decomposition

Another way to avoid the curse of dimensionality is to use a Singular Value Decomposition (SVD) [116] to reduce the number of parameters. With an SVD, the parameters x and y are transformed into a lower-dimensional vector of parameters in such a way that the reduced vector of parameters describes as much of the variation as possible. To do this, an SVD is made of the matrix that contains all N observed situations:

$$\begin{bmatrix} x_1 - \mu_x & \cdots & x_N - \mu_x \\ y_1 - \mu_y & \cdots & y_N - \mu_y \end{bmatrix} = U\Sigma V^\mathsf{T}. \tag{8.7}$$

Here, $\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i$ and $\mu_y = \frac{1}{N}\sum_{i=1}^{N} y_i$. The matrices $U \in \mathbb{R}^{(n_x+n_y)\times(n_x+n_y)}$ and $V \in \mathbb{R}^{N\times N}$ are orthonormal, i.e., $U^{-1} = U^\mathsf{T}$ and $V^{-1} = V^\mathsf{T}$. Moreover, $\Sigma \in \mathbb{R}^{(n_x+n_y)\times N}$ has only zeros except at the diagonal: the (j,j)-th element is σ_j, $j \in \{1, \ldots, \bar{N}\}$ with $\bar{N} = \min(n_x + n_y, N)$, such that

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\bar{N}} \geq 0. \tag{8.8}$$

Because these so-called singular values are in decreasing order, we can approximate x and y by setting $\sigma_j = 0$ for $j > d$ with d chosen[2] such that $n_x < d < n_x + n_y$:

$$\begin{bmatrix} x_i - \mu_x \\ y_i - \mu_y \end{bmatrix} = \sum_{j=1}^{\bar{N}} \sigma_j v_{ij} u_j \approx \sum_{j=1}^{d} \sigma_j v_{ij} u_j, = \begin{bmatrix} \bar{U}_1 \\ \bar{U}_2 \end{bmatrix} \bar{\Sigma} \tilde{v}_i, \tag{8.9}$$

where v_{ij} is the (i,j)-th element of V and u_j is the j-th column of U. Moreover, $\bar{U}_1$ is the n_x-by-d upper left submatrix of U, $\bar{U}_2$ is the n_y-by-d lower left submatrix U, $\bar{\Sigma} \in \mathbb{R}^{d\times d}$ is the diagonal matrix with the first d singular values on its diagonal, and $\tilde{v}_i^\mathsf{T} = \begin{bmatrix} v_{i1} & \cdots & v_{id} \end{bmatrix}$. Thus, with μ_x, μ_y, $\bar{U}_1$, $\bar{U}_2$, and $\bar{\Sigma}$, the $(n_x + n_y)$-dimensional vector $\begin{bmatrix} x_i^\mathsf{T} & y_i^\mathsf{T} \end{bmatrix}^\mathsf{T}$ is approximated using the d-dimensional vector $\tilde{v}_i$.

Instead of estimating the pdf of $\begin{bmatrix} x_i^\mathsf{T} & y_i^\mathsf{T} \end{bmatrix}^\mathsf{T}$, we now estimate the pdf of $\tilde{v}_i$ using KDE as described in (8.3). To sample from $\hat{p}(y|x)$, we can sample from the estimated distribution of $\tilde{v}_i$. Because (8.9) is a linear mapping, the sample $\bar{v}$ that is drawn from the estimated distribution of $\tilde{v}_i$ is subject to a linear constraint:

$$\bar{U}_1 \bar{\Sigma} \bar{v} = x - \mu_x. \tag{8.10}$$

In Chapter 6 [75], an algorithm is provided for sampling from a KDE with a Gaussian kernel of (8.4) such that the resulting samples are subject to a linear constraint such as (8.10).

8.3.3. Estimate crash probability using a Monte Carlo simulation

Monte Carlo simulations are used to estimate $\mathbb{P}(\mathrm{C}|x)$, i.e., the probability of an event C given the initial situation described by x. The details of the simulation

[2] We have $d < n_x + n_y$, such that the dimension is reduced (from $n_x + n_y$ to d) and we have $d > n_x$, such that the number of linear constraints in (8.10) (n_x) is smaller than the number of variables (d).

depend on the actual application. For example, if the goal of our SSM is to evaluate the risk that a human-driven vehicle collides, the simulation should involve a human driving behavior model. On the other hand, if the goal is to evaluate the risk of crash when an ADS is controlling the vehicle, the simulation should include the model of this ADS.

A straightforward way to compute $\mathbb{P}(\mathrm{C}|x)$ is to repeat a certain number of simulations with the same x and count the number of simulations that result in the event C. If N_{sim} denotes the number of simulations and N_{C} is the number of events C, then $\mathbb{P}(\mathrm{C}|x)$ could be estimated using

$$\hat{P}(\mathrm{C}|x) = \frac{N_{\mathrm{C}}}{N_{\mathrm{sim}}}. \tag{8.11}$$

An important choice for estimating $\mathbb{P}(\mathrm{C}|x)$ is the number of simulations, N_{sim}. One approach is to keep increasing N_{sim} until there is enough confidence in the estimation of (8.11). For example, the Clopper-Pearson interval [56] or the Wilson score interval [305] can be used to determine the confidence of the estimation of (8.11). A disadvantage of (8.11) is that only the fact whether the event C occurred or not is used, while the simulation provides more information, such as the minimum distance between two objects or the impact speed in case of a crash. Therefore, we provide an alternative approach to estimate $\mathbb{P}(\mathrm{C}|x)$.

For the alternative approach, let us assume that one simulation run provides more information than just the fact that the event C occurred or not. Let $z \in \mathbb{R}^{n_z}$ be a continuous variable representing the result of a simulation run and let $\mathcal{Z}_{\mathrm{C}}$ denote the set of possible simulation results in which the event C occurred. Thus, $z \in \mathcal{Z}_{\mathrm{C}}$ if and only if the simulation results in the event C. We assume $\mathcal{Z}_{\mathrm{C}}$ is known; see, e.g., the example in Section 8.4.1. Therefore, we have

$$\mathbb{P}(\mathrm{C}|x) = \mathbb{P}(z \in \mathcal{Z}_{\mathrm{C}}|x) = \int_{\mathcal{Z}_{\mathrm{C}}} p(z|x)\,\mathrm{d}z. \tag{8.12}$$

Similar as with the estimation of $p(x, y)$ in Section 8.3.2, we employ KDE to estimate $p(z|x)$:

$$\hat{p}(z|x) = \frac{1}{N_{\mathrm{sim}}} \sum_{j=1}^{N_{\mathrm{sim}}} K_{H_z}(z_j - z), \tag{8.13}$$

where z_j denotes the result of the j-th simulation and H_z denotes an appropriate bandwidth matrix. The kernel function $K_{H_z}(\cdot)$ is similarly defined as (8.4). We can now estimate $\mathbb{P}(\mathrm{C}|x)$ by substituting $\hat{p}(z|x)$ of (8.13) for $p(z|x)$:

$$\hat{P}(\mathrm{C}|x) = \hat{P}(z \in \mathcal{Z}_{\mathrm{C}}|x) = \int_{\mathcal{Z}_{\mathrm{C}}} \hat{p}(z|x)\,\mathrm{d}z = \frac{1}{N_{\mathrm{sim}}} \sum_{j=1}^{N_{\mathrm{sim}}} \int_{\mathcal{Z}_{\mathrm{C}}} K_{H_z}(z_j - z)\,\mathrm{d}z. \tag{8.14}$$

Similar as with (8.11), we need to choose the number of simulations N_{sim}. Our proposal is to keep increasing N_{sim} until the variance of $\hat{P}(z \in \mathcal{Z}_{\mathrm{C}}|x)$ is below a

threshold $\epsilon > 0$. The variance follows from [207]:

$$\mathrm{Var}\big[\hat{P}(z \in \mathcal{Z}_\mathrm{C}|x)\big] = \frac{\mathbb{P}(z \in \mathcal{Z}_\mathrm{C}|x)(1 - \mathbb{P}(z \in \mathcal{Z}_\mathrm{C}|x))}{N_\mathrm{sim}}. \tag{8.15}$$

Because $\mathbb{P}(z \in \mathcal{Z}_\mathrm{C}|x)$ is unknown, we use the estimated counterpart of (8.14). Thus, N_sim is increased until the following condition is met:

$$\frac{\hat{P}(z \in \mathcal{Z}_\mathrm{C}|x)\big(1 - \hat{P}(z \in \mathcal{Z}_\mathrm{C}|x)\big)}{N_\mathrm{sim}} < \epsilon. \tag{8.16}$$

8.3.4. Regression for real-time estimation of crash probability

To evaluate the risk measure during real-time operation of the ego vehicle, the expression of (8.14) is problematic because it would require N_sim simulations. Even if the calculation is accelerated using a technique such as importance sampling, it might take too long. Therefore, we propose to evaluate (8.14) only for some fixed $\{x'_k\}_{k=1}^m$. Next, regression is used to estimate (8.14). One option is to choose a parametric model, e.g., a logistic model, and estimate the parameters of the model using $\{(x'_k, \hat{P}(\mathrm{C}|x'_k))\}_{k=1}^m$. Up to our knowledge, however, there is no good reason to assume a particular parametric model, so we use a non-parametric regression technique to estimate (8.14). More specifically, we use the Nadaraya-Watson (NW) kernel estimator [300] because it automatically smooths the data (as is demonstrated in Section 8.4.1) and the approximation is guaranteed to give a number between 0 and 1, also when extrapolating the data. The NW kernel estimator is given by:

$$\hat{P}(\mathrm{C}|x) \approx \frac{\sum_{k=1}^m K_{H_\mathrm{NW}}(x - x'_k)\hat{P}(\mathrm{C}|x'_k)}{\sum_{k=1}^m K_{H_\mathrm{NW}}(x - x'_k)}. \tag{8.17}$$

Here, $\hat{P}(\mathrm{C}|x'_k)$ is based on (8.14) and $K_{H_\mathrm{NW}}(\cdot)$ represents the Gaussian kernel given by (8.4). Two important choices have to be made: The choice of the $\{x'_k\}_{k=1}^m$ for which to evaluate (8.14) and the choice of the bandwidth matrix H_NW. We suggest to base the design points $\{x'_k\}_{k=1}^m$ on the data that is used to estimate $p(y|x)$ in Section 8.3.2, i.e., $\{x_i\}_{i=1}^N$, such that all x_i have at least one design point x'_k nearby. In other words, $\{x'_k\}_{k=1}^m$ are chosen such that

$$\min_k (x_i - x'_k)^\mathsf{T} W (x_i - x'_k) \leq 1, \quad \forall i \in \{1, \ldots, N\}, \tag{8.18}$$

where W denotes a weighting matrix. Note that if W is the identity matrix, then (8.18) calculates the minimum squared Euclidean distance. In general, W is a diagonal matrix. Choosing the diagonal elements of W is a trade-off; if the elements are too large, then too many details are lost in the approximation of (8.17); if the elements are too small, it takes too long to evaluate (8.14) m times, as m increases for lower diagonal elements of W. The bandwidth matrix H_NW might be based on W, e.g., $H_\mathrm{NW} = W^{-1}$. Alternatively, H_NW might be based on the measurement

uncertainty of x if this measurement uncertainty is significant, where a larger H_{NW} applies in case of a larger measurement uncertainty of x. Note that if H_{NW} is a diagonal matrix with positive values on the diagonal that are close to zero, then the NW kernel estimation of (8.17) acts like nearest-neighbor interpolation.

8.4. Case study

In the first part of the case study, we illustrate that the PRISMA method generalizes the SSM proposed by Wang and Stamatiadis [295]. Here, we also demonstrate the effect of ϵ on the accuracy of the SSM derived by the PRISMA method and we show the difference between $\hat{P}(\mathrm{C}|x)$ of (8.14) and the approximation of $\hat{P}(\mathrm{C}|x)$ using the NW kernel estimator of (8.17). In Section 8.4.2, we demonstrate how the PRISMA method can be used to create a new SSM that calculates the risk of a crash in a longitudinal interaction between two vehicles. The SSM derived in Section 8.4.2 is analyzed in Section 8.4.3. The last part of the case study discusses and illustrates a method for benchmarking an SSM.

8.4.1. Comparison with Wang and Stamatiadis' measure

Wang and Stamatiadis [295] provide an SSM, which we denote by WS, that calculates the probability of a crash under certain assumptions. We first explain how WS is calculated. Next, we show how to estimate this SSM using our method. Finally, we illustrate the results of both.

Measure of Wang and Stamatiadis

The SSM WS calculates the probability of a crash of the ego vehicle and the leading vehicle, where the ego vehicle is following an initially slower driving leading vehicle. The SSM WS is based on the following assumptions [295]:

- the leading vehicle keeps a constant speed;
- the (driver of the) ego vehicle starts to brake after its reaction time, denoted by t_{r};
- based on [121], the reaction time t_{r} is distributed according to a log-normal distribution, such that the mean is $0.92\,\mathrm{s}$ and the standard deviation is $0.28\,\mathrm{s}$;
- when the ego vehicle reacts, it brakes with its MADR, denoted by a_{MADR}; and
- a_{MADR} is distributed according to a truncated normal distribution with a mean of $9.7\,\mathrm{m/s^2}$, a standard deviation of $1.3\,\mathrm{m/s^2}$, a lower bound of $L = 4.2\,\mathrm{m/s^2}$ [64], and an upper bound of $U = 12.7\,\mathrm{m/s^2}$ [64].

To calculate WS at a given time t, the speed difference between the ego vehicle and the leading vehicle, $\Delta_v(t)$, and the TTC, $t_{\mathrm{TTC}}(t)$, are used. Note that $t_{\mathrm{TTC}}(t)$ is the ratio of the gap, $g(t)$, between the ego vehicle and the leading vehicle and $\Delta_v(t)$. If $\Delta_v(t) \leq 0$, then the ego vehicle drives slower and there is no risk of a future

crash according to Wang and Stamatiadis [295], so $\mathrm{WS}(t) = 0$. Given a_{MADR}, the driver of the ego vehicle needs to react within

$$t_{\max}(t) = t_{\mathrm{TTC}}(t) - \frac{\Delta_v(t)}{2a_{\mathrm{MADR}}} \tag{8.19}$$

in order to avoid a crash. Using the distributions of a_{MADR} and t_{r}, we can calculate the probability that this is the case, resulting in:

$$\mathrm{WS}(t) = \begin{cases} 0 & \text{if} \quad \Delta_v(t) \leq 0 \\ \int_{\hat{L}}^{U} \int_0^{t_{\max}(t)} p(t_{\mathrm{r}}) p(a_{\mathrm{MADR}}) \, \mathrm{d}t_{\mathrm{r}} \, \mathrm{d}a_{\mathrm{MADR}} & \text{if} \quad \Delta_v(t) > 0 \wedge \frac{\Delta_v(t)}{2t_{\mathrm{TTC}}(t)} < U, \\ 1 & \text{otherwise} \end{cases} \tag{8.20}$$

with $\hat{L} = \max\left(L, \frac{\Delta_v(t)}{2t_{\mathrm{TTC}}(t)}\right)$, $p(t_{\mathrm{r}})$ is the log-normal probability density of t_{r}, and $p(a_{\mathrm{MADR}})$ is the truncated normal probability density of a_{MADR}.

Replicating Wang and Stamatiadis' measure

Because WS is based on $\Delta_v(t)$ and $t_{\mathrm{TTC}}(t)$, these two variables are also used by the PRISMA method to describe the initial situation:

$$x^{\mathsf{T}}(t) = \begin{bmatrix} \Delta_v(t) & t_{\mathrm{TTC}}(t) \end{bmatrix}. \tag{8.21}$$

The leading vehicle is assumed to have a constant speed, so $x(t)$ of (8.21) already describes the future situation of the leading vehicle. Therefore, there is no need to estimate $p(y|x)$. At the start of each simulation run, the driver of the ego vehicle is not braking. After the reaction time t_{r}, the driver starts braking with a_{MADR}. The random parameters t_{r} and a_{MADR} are similarly distributed as described earlier.

Since we are interested in the probability of a crash, the event C denotes a crash. A simulation run ends if either the ego vehicle and the leading vehicle are colliding or if the gap between the ego vehicle and the leading vehicle is not decreasing. Depending on the reason for a simulation run to end, we consider the following result:

- If the ego vehicle and the leading vehicle are colliding, we are interested in the "severity" of the crash. This is expressed using the speed difference: $v_{\mathrm{l}}(t_1) - v_{\mathrm{e}}(t_1)$, with t_1 denotes the final time of the simulation run.
- If there is no crash, we are interested in how close the two vehicles came. Therefore, the minimum gap is used, which is $g(t_1)$.

Thus, we have:

$$z = \begin{cases} v_{\mathrm{l}}(t_1) - v_{\mathrm{e}}(t_1) & \text{if crash} \\ g(t_1) & \text{otherwise} \end{cases}. \tag{8.22}$$

Clearly, $z \leq 0$ indicates a crash, so $\mathcal{Z}_{\mathrm{C}} = (-\infty, 0]$. The minimum number of simulations to estimate $\mathbb{P}(\mathrm{C}|x)$ is set to 10. The number of simulations is further

increased until the condition in (8.16) with $\epsilon = 0.2$ or $\epsilon = 0.02$ is met. For the design points $\{x'_k\}_{k=1}^{m}$, we use a rectangular grid with Δ_v ranging from $0\,\mathrm{m/s}$ till $40\,\mathrm{m/s}$ with steps of $2\,\mathrm{m/s}$ and t_{TTC} ranging from $0.5\,\mathrm{s}$ till $4\,\mathrm{s}$ in steps of $0.1\,\mathrm{s}$. Thus, $m = 21 \cdot 36 = 756$. For H_{NW}, a diagonal matrix is chosen with the diagonal elements corresponding to the square of the step size of the grid, i.e., $4\,\mathrm{m^2/s^2}$ and $0.01\,\mathrm{s^2}$.

Comparison

Figure 8.1 shows the results of the comparison between the measure of Wang and Stamatiadis [295] and the measure derived using the PRISMA method described in Section 8.3. The blue lines in Figure 8.1 denote WS of (8.20). These lines show that for lower values of t_{TTC}, WS increases. Also, for increasing values of Δ_v (solid, dashed, and dotted lines), the risk measure WS increases. Both these observations match the intuition that a lower TTC and a higher speed difference are less safe.

The red lines in Figure 8.1 denote $\hat{P}(\mathrm{C}|x)$ of (8.14). Figure 8.1 illustrates that $\hat{P}(\mathrm{C}|x)$ follows the same trend as WS. Figure 8.1 also illustrates the effect of the choice of the threshold ϵ. In general, for a lower value of ϵ, the number of simulations N_{sim} used in (8.13) is higher. As a result, it can be expected that the estimation $\hat{P}(\mathrm{C}|x)$ is closer to $\mathbb{P}(\mathrm{C}|x)$ (cf. (8.15)). A comparison of Figure 8.1a ($\epsilon = 0.2$) and Figure 8.1b ($\epsilon = 0.02$) demonstrates this effect.

The green lines in Figure 8.1 represent the approximation of $\hat{P}(\mathrm{C}|x)$ using the NW kernel estimator of (8.17). This illustrates the regression using the NW kernel estimator: the green lines in can be seen as smoothed versions of the red lines.

8.4.2. Developing an SSM for longitudinal interactions

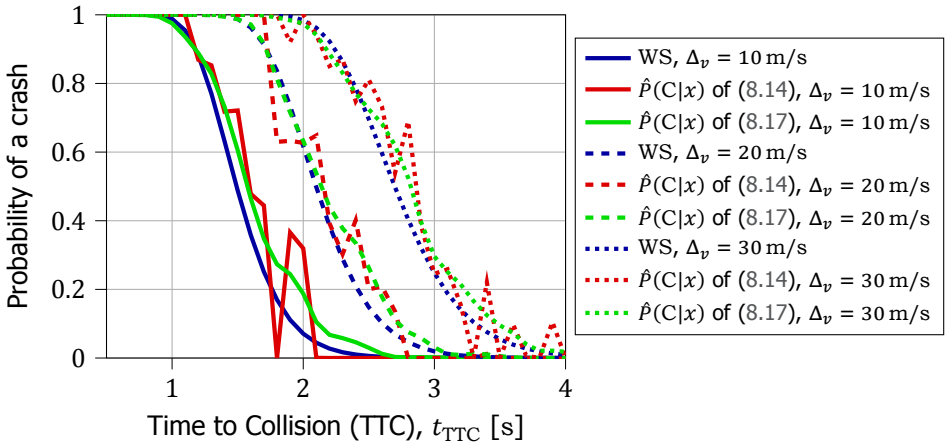
To further illustrate the PRISMA method, we apply it to derive an SSM that calculates the risk of a crash in a longitudinal interaction between two vehicles. The SSM is based on the NGSIM data set [7]. The NGSIM data set contains vehicle trajectories obtained from video footage of cameras that were located at several motorways in the U.S.A. The derived SSM estimates the risk of a crash of the ego vehicle with its leading vehicle. To describe the initial situation at time t, $n_x = 4$ parameters are used:

- the speed of the leading vehicle ($v_{\mathrm{l}}(t)$);
- the acceleration of the leading vehicle ($a_{\mathrm{l}}(t)$);
- the speed of the ego vehicle ($v_{\mathrm{e}}(t)$); and
- the log of the gap between the leading vehicle and the ego vehicle[3] $\log g(t)$.

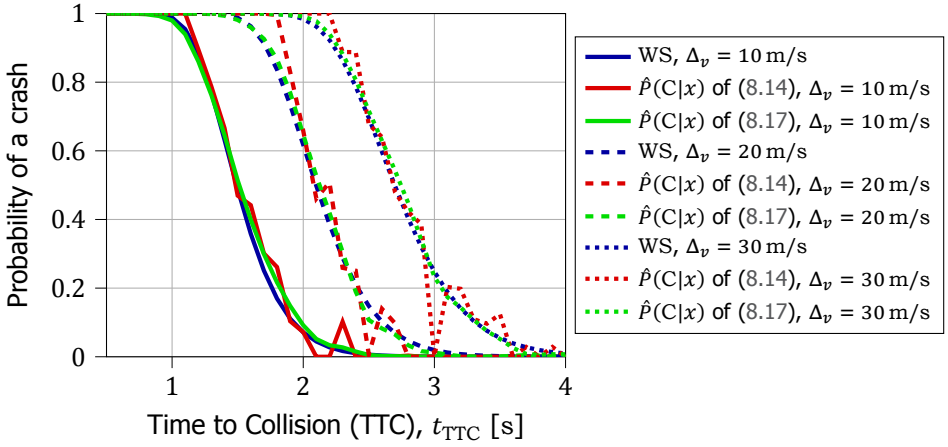
Thus, we have:

$$x^{\mathrm{T}}(t) = \begin{bmatrix} v_{\mathrm{l}}(t) & a_{\mathrm{l}}(t) & v_{\mathrm{e}}(t) & \log g(t) \end{bmatrix}. \tag{8.23}$$

[3] Note that the log is used, such that there are, relatively speaking, more simulations performed with a small initial gap, cf. (8.18).

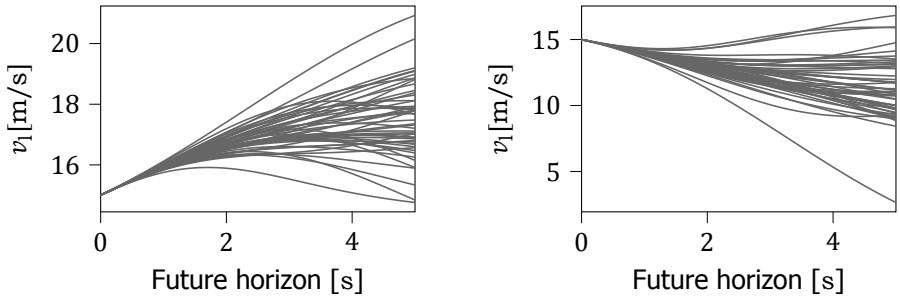


(a) $\epsilon = 0.2$.



(b) $\epsilon = 0.02$.

Figure 8.1: Comparison of WS of (8.20) and $\hat{P}(\mathrm{C}|x)$, where $\hat{P}(\mathrm{C}|x)$ is based on either (8.14) (red lines) or (8.17) (green lines). Here, $\hat{P}(\mathrm{C}|x)$ is based on the same underlying assumptions as WS. The influence of the parameter ϵ, which determines the number of simulations to estimate $\mathbb{P}(\mathrm{C}|x)$, is illustrated by using different values.



(a) Initial situation: $v_l = 15\,\mathrm{m/s}$ and $a_l = 1\,\mathrm{m/s^2}$.

(b) Initial situation: $v_l = 15\,\mathrm{m/s}$ and $a_l = -1\,\mathrm{m/s^2}$.

Figure 8.2: 50 potential future situations samples from the KDE that is constructed using data from the NGSIM data set.

The speed of the leading vehicle at $n_h = 50$ instances, each $\Delta t = 0.1\,\mathrm{s}$ apart, describes the future situation:

$$y^T(t) = \begin{bmatrix} v_l(t+\Delta t) & \cdots & v_l(t+n_h\Delta t) \end{bmatrix}. \tag{8.24}$$

It is assumed that $y(t)$ depends on $v_l(t)$ and $a_l(t)$. To model this dependency with a single kernel density estimator would give us a pdf with $n_h + 2$ dimensions. To reduce the dimensionality, we use an SVD as described in Section 8.3.2 with[4] $d = 4$. In total, 18182 longitudinal interactions between two vehicles have been analyzed. For each second of an interaction, we extract an "initial situation" x_i and a corresponding "future situation" y_i. This leads to $N = 469453$ data points. Based on Silverman's rule of thumb [262], we use a bandwidth matrix $H = h^2 I_4$ for the KDE with $h \approx 0.186$ and I_4 denoting the 4-by-4 identity matrix.

To demonstrate the sampling from the estimated density of the reduced parameter vector subject to a linear constraint such as (8.10), the plots in Figure 8.2 show 50 different future situations in the form of (8.24). Figure 8.2a assumes an initial situation with $v_l = 15\,\mathrm{m/s}$ and $a_l = 1\,\mathrm{m/s^2}$ and Figure 8.2b assumes an initial situation with $v_l = 15\,\mathrm{m/s}$ and $a_l = -1\,\mathrm{m/s^2}$. Note that the same pdf is used to produce the lines in Figure 8.2; the only difference between Figure 8.2a and Figure 8.2b is a different linear constraint (based on v_l and a_l) on the generated samples. In case a simulation run is longer than $5\,\mathrm{s}$, the speed of the leading vehicle is assumed to remain constant after these $5\,\mathrm{s}$. Note that a simulation run is rarely longer than $5\,\mathrm{s}$, so this assumption does not have a significant effect on the results.

To estimate $\mathbb{P}(\mathrm{C}|x)$ (Section 8.3.3), we use the Intelligent Driver Model plus (IDM+) [247] for modeling the ego vehicle driver behavior and response. In addition to IDM+, we assume that the driver has a reaction time that is similarly distributed as t_r in Section 8.4.1 and that the MADR is similarly distributed as a_{MADR} in Section 8.4.1. The simulation result z is defined according to (8.22). The minimum

[4] Note that because we assume that $y(t)$ depends on two parameters of $x(t)$, i.e., $v_l(t)$ and $a_l(t)$, we need to choose d such that $2 < d < n_h + 2$.

number of simulations to estimate $\mathbb{P}(\mathrm{C}|x)$ is set to 10 and this number is further increased until the condition in (8.16) with $\epsilon = 0.1$ is met.

To calculate $\mathbb{P}(\mathrm{C}|x)$ using (8.17), we create a grid of points $\{x'_k\}_{k=1}^{m}$ using the method explained in Section 8.3.4. For W, we use a diagonal matrix with diagonal elements: 0.25, 4, 0.25, and 0.25, which is a trade-off between keeping many points such that the estimation in (8.17) is accurate while also keeping the total number of points for which $\mathbb{P}(\mathrm{C}|x)$ is estimated manageable. With this choice of W, we have $m = 10129$. For the regression of (8.17), we use $H_{\mathrm{NW}} = W^{-1}$.

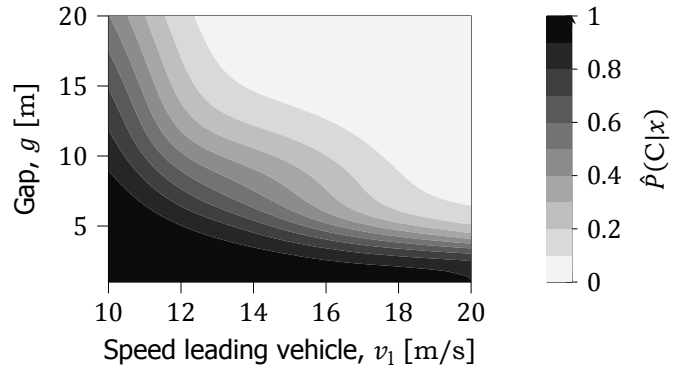
8.4.3. Analyzing the SSMs for longitudinal interactions

The heat maps in Figure 8.3 show how the developed SSM depends on the input variables v_{l} and g. The other two parameters, v_{e} and a_{l}, are fixed for each heat map. The heat maps shows that the estimated crash probability is practically 0 if both v_{l} and g are large. This seems reasonable because in that case the ego vehicle is at a safe distance from its leading vehicle while the approaching speed is small. In addition, for a fixed v_{e}, we see that the crash risk increases as the difference in speed increases, as is expected. The same applies for a decreasing distance between the two vehicles. For small values of v_{l} and g, the estimated crash probability is practically 1. The left and center heat maps of Figure 8.3 show that for a higher speed of the ego vehicle, the crash probability is estimated to be higher. Similarly, the right and center heat maps of Figure 8.3 show that for a lower initial acceleration of the leading vehicle, the crash probability is estimated to be higher.

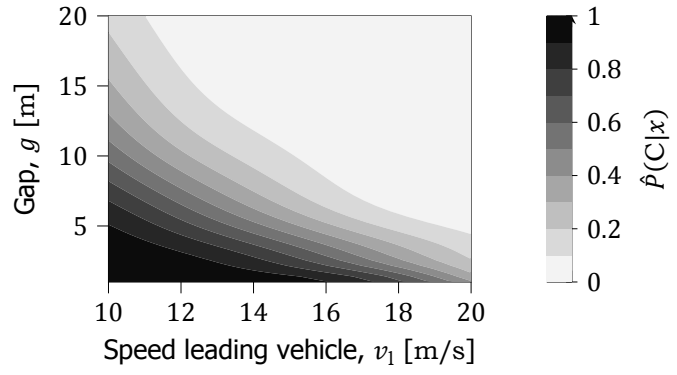
In Figure 8.4, the evaluations of the measure described in Section 8.4.2 are shown for three different scenarios. Each of the three scenarios considers an ego vehicle and a leading vehicle driving in front of the ego vehicle. Both vehicles are driving in the same direction and in the same lane. For comparison, the right plots also include the evaluations of WS of (8.20).

The first scenario in Figure 8.4 (top row) shows a scenario in which the leading vehicle initially drives with a speed of $20\,\mathrm{m/s}$. The leading vehicle starts to decelerate after $3\,\mathrm{s}$ toward a speed of $10\,\mathrm{m/s}$ with an average deceleration of $3\,\mathrm{m/s^2}$. The ego vehicle initially drives with a speed of $24\,\mathrm{m/s}$ at a distance of $40\,\mathrm{m}$ from the leading vehicle. The ego vehicle starts decelerating after $2\,\mathrm{s}$ toward a speed of $8\,\mathrm{m/s}$ within $4\,\mathrm{s}$. It takes $4\,\mathrm{s}$ more to reach the speed of the leading vehicle. Because the ego vehicle always maintains a relatively large distance toward the leading vehicle, both SSMs do not qualify this scenario as risky, considering the estimated crash probability that stays below 0.1.

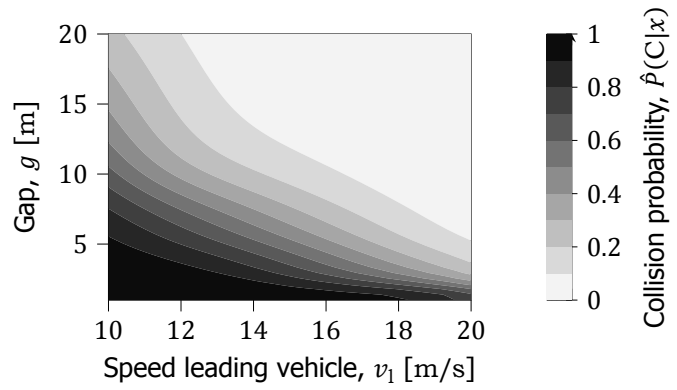
The second scenario in Figure 8.4 (center row) differs from the first scenario in that the ego vehicle starts to decelerate $2\,\mathrm{s}$ later. As a result, the ego vehicle approaches the leading vehicle up to a distance of $5.4\,\mathrm{m}$. According to $\hat{P}(\mathrm{C}|x)$ from Section 8.4.2 (blue line in the right plot of Figure 8.4), the probability of a crash reaches almost 1, indicating that around that time, the risk of a crash is high. The local minimum of $\hat{P}(\mathrm{C}|x)$ at around $6\,\mathrm{s}$ illustrates the effect of the numerical approximation of $\mathbb{P}(\mathrm{C}|x)$. Because we have used $\epsilon = 0.1 > 0$, the resulting estimation may have an error. When lowering the threshold ϵ, the resulting



(a) $v_e = 25\,\mathrm{m/s}$, $a_l = 0\,\mathrm{m/s^2}$.



(b) $v_e = 20\,\mathrm{m/s}$, $a_l = 0\,\mathrm{m/s^2}$.



(c) $v_e = 20\,\mathrm{m/s}$, $a_l = -1\,\mathrm{m/s^2}$.

Figure 8.3: Heat maps of the SSM described in Section 8.4.2 as a function of the speed of the leading vehicle (v_l) and the gap between the ego vehicle and the leading vehicle (g). For each heat map, the other two input parameters are fixed at $v_e = 25\,\mathrm{m/s}$ (a) or $v_e = 20\,\mathrm{m/s}$ (b and c) and $a_l = 0\,\mathrm{m/s^2}$ (a and b) or $a_l = -1\,\mathrm{m/s^2}$ (c). The estimated crash probability ranges from 0 (white) to 1 (black).

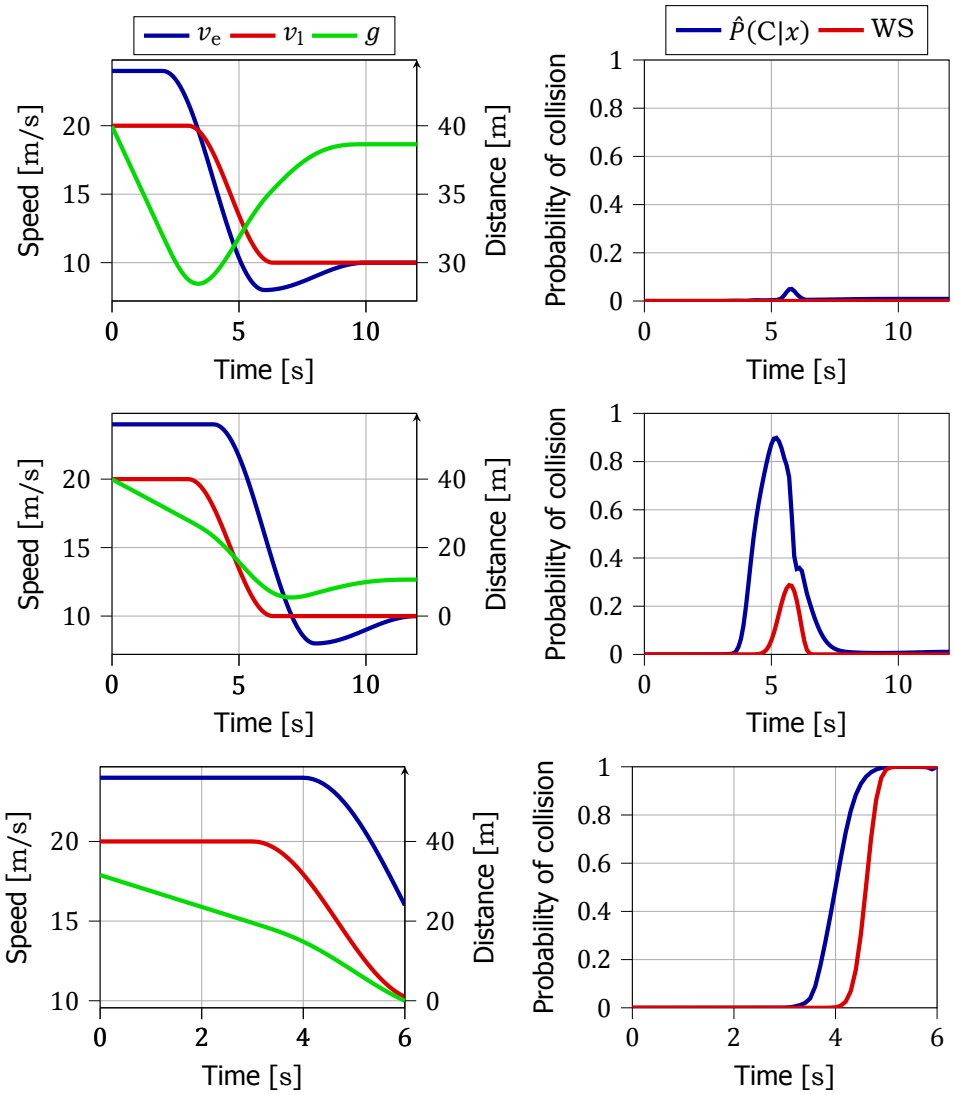


Figure 8.4: Demonstration of SSMs for 3 hypothetical scenarios. The left plots show the speeds of the ego vehicle (blue line) and leading vehicle (red line) and the distance between the ego vehicle and the leading vehicle (green line, scale on the right of the plot). The right plots show the estimated probability of a crash corresponding to the three scenarios according to the SSM explained in Section 8.4.2 (blue lines) and the SSM of Wang and Stamatiadis [295] explained in Section 8.4.1 (red lines).

$\hat{P}(\mathrm{C}|x)$ in the center right plot in Figure 8.4 will be smoother. This goes, however, at the cost of an increased number of simulations[5].

The third scenario in Figure 8.4 (bottom row) differs from the second scenario in that the initial distance between the ego vehicle and the leading vehicle is $31.5\,\mathrm{m}$ instead of $40\,\mathrm{m}$. As a result, the ego vehicle collides with the leading vehicle after $6\,\mathrm{s}$. As expected, the SSMs in Figure 8.4 indicate a crash probability of 1. The difference between $\hat{P}(\mathrm{C}|x)$ and WS is that $\hat{P}(\mathrm{C}|x)$ increases earlier. Note that $\hat{P}(\mathrm{C}|x)$ increasing sooner than WS does not necessarily mean that it is better: because there is no objective truth for an SSM, we cannot argue that one SSM is better than another SSM. Hence, in the next section, we will present a quantitative approach to benchmark an SSM.

8.4.4. Benchmarking an SSM with expected risk trends

In this section, we demonstrate an approach for benchmarking an SSM that is based on expected risk trends discussed in Mullakkal-Babu *et al.* [203], who argue that the risk increases if the approaching speed of the ego vehicle toward the leading vehicle increases. Also, the risk increases with a higher ego vehicle speed [1] or a higher driver reaction time [163]. On the other hand, the risk decreases with a higher road friction [293] or a larger intervehicle spacing [203].

To check whether the developed SSM follows these five expected risk trends[6], we evaluate the partial derivatives of the measure of (8.17). The intuition is as follows: If the expected risk trend for an input X (e.g., the ego vehicle speed) is that the risk increases as X increases, then we expect the partial derivative of our SSM with respect to X to be positive. Furthermore, if we evaluate the partial derivative at many points, we expect that at least the majority of these evaluated partial derivatives is positive. Similarly, if we expect that the risk measure decreases with increasing X, then we expect that at least the majority of the evaluated partial derivatives is negative.

To illustrate the approach for benchmarking an SSM, we use the SSM of Section 8.4.2 with a few different assumptions. Because we have not described an expected trend regarding a_l, we simply use $a_\mathrm{l} = 0$. Also, because the expected risk trend for the relative speed is defined, we use the relative speed, i.e., $\Delta_v = v_\mathrm{e} - v_\mathrm{l}$, instead of v_l. For the same reason, instead of assuming a random reaction time t_r and MADR a_MADR, these are now considered as inputs to our measure. Finally, instead of using the log of the gap between the ego vehicle and the leading vehicle, we use the gap as a direct input. Thus, we have:

$$x^\mathsf{T} = \begin{bmatrix} v_\mathrm{e} - v_\mathrm{l} & v_\mathrm{e} & t_\mathrm{r} & g & a_\mathrm{MADR} \end{bmatrix}. \tag{8.25}$$

We compute $\hat{P}(\mathrm{C}|x)$ using (8.17) where the points $\{x'_k\}_{k=1}^m$ are taken from a grid.

[5] Alternatively, the bandwidth matrix H_NW may be increased. On the one hand, this will lower the variance of the error, but, on the other hand, it will increase the bias of the result. We refer the interested reader to [52] for more details on the effect of H_NW.

[6] In [203], a sixth expected risk trend is mentioned based on [97], namely the vehicle mass. Our interpretation of [97], however, is that the ratio of masses of two colliding vehicles influences the safety risk and that one cannot argue that a higher mass of the ego vehicle necessarily increases the safety risk. Therefore, we exclude the ego vehicle mass from our analysis.

Table 8.1: Percentiles of the partial derivatives of the SSM and the corresponding expected risk trends.

	$v_e - v_l$	v_e	t_r	g	a_{MADR}
Expected trend	Increase	Increase	Increase	Decrease	Decrease
Maximum	0.1629	0.1555	1.3136	0.0010	0.0037
99th percentile	0.1585	0.1162	0.8968	0.0002	0.0002
95th percentile	0.1495	0.0524	0.6765	0.0000	−0.0000
90th percentile	0.1346	0.0151	0.5351	−0.0000	−0.0000
75th percentile	0.0746	0.0012	0.2917	−0.0002	−0.0003
50th percentile	0.0114	0.0001	0.0605	−0.0070	−0.0054
25th percentile	0.0004	0.0000	0.0022	−0.0320	−0.0290
10th percentile	0.0000	−0.0000	0.0000	−0.0545	−0.0654
5th percentile	0.0000	−0.0002	0.0000	−0.0645	−0.0880
1th percentile	0.0000	−0.0007	−0.0020	−0.0781	−0.1337
Minimum	−0.0035	−0.0030	−0.0180	−0.1076	−0.2030

For each input variable, 10 different values at equal distance are used, resulting in $m = 10^5$. Here, $v_e - v_l$ ranges from $0\,\mathrm{m/s}$ to $20\,\mathrm{m/s}$, v_e ranges from $10\,\mathrm{m/s}$ to $30\,\mathrm{m/s}$, t_r ranges from $0.5\,\mathrm{s}$ to $1.5\,\mathrm{s}$, g ranges from $5\,\mathrm{m}$ to $30\,\mathrm{m}$, and a_{MADR} ranges from $4\,\mathrm{m/s^2}$ to $10\,\mathrm{m/s^2}$. A threshold $\epsilon = 0.02$ is used. For the bandwidth matrix H_{NW}, we use a diagonal matrix with the (i,i)-th entry corresponding to the squared difference between two consecutive values of the i-th entry of x. For example, the first value is $(20\,\mathrm{m/s}/(10-1))^2 \approx 4.9\,\mathrm{m^2/s^2}$. The other values on the diagonal are: $4.9\,\mathrm{m^2/s^2}$, $0.012\,\mathrm{s^2}$, $7.7\,\mathrm{m^2}$, and $0.44\,\mathrm{m^2/s^4}$. For each input variable listed in (8.25), we evaluate the partial derivative of (8.17) at each x'_k, $k \in \{1, \ldots, m\}$.

Table 8.1 shows the result of the benchmarking. It shows that the SSM follows the expected risk trends mostly. E.g., in more than $99\,\%$ of the cases, the partial derivative of the relative speed $(v_e - v_l)$ is positive. For the remaining $1\,\%$, the partial derivative is negative, albeit only slightly. One explanation is that this remaining $1\,\%$ is caused by the inaccuracies introduced by the numerical approximation of (8.14).

8.5. Discussion

Typically, SSMs rely on assumptions regarding the behavior of traffic participants. An advantage of the presented PRISMA method for deriving SSMs is that the PRISMA method is not bound to certain predetermined assumptions. We want to stress, however, that when using the PRISMA method for deriving an SSM, a set of assumptions is still needed. In fact, multiple SSMs can be derived by using the PRISMA method with different sets of assumptions. As a result, the PRISMA method can be used to derive multiple SSMs that are applicable in various types of scenarios, e.g., ranging from vehicle-following scenarios to scenarios at intersections. Note that although the PRISMA method is applicable in various types of scenarios, the current case study focuses on longitudinal traffic conflicts. In a future work, we will

present the application of the PRISMA method for deriving SSMs for lateral traffic conflicts.

The PRISMA method uses data to adapt the SSMs to, e.g., the local traffic behavior. More specifically, the data are used to predict the possible future situations (y) given an initial situation (x). This can be an advantage because the data can be used to rely less on assumptions as to how the future develops given an initial situation. To fully benefit from this approach, the data should satisfy a few conditions. First, the recorded data need to represent the actual traffic behavior in which the SSMs are applied. Second, we need enough data to estimate $p(y|x)$. In Chapter 3 [72], a metric is presented that can be used to determine whether enough data have been collected to estimate $p(y|x)$ accurately.

The PRISMA method can still be applied in case no data are available. The first alternative is to use existing knowledge to determine an estimate of $p(y|x)$ instead of estimating $p(y|x)$ on the basis of data. For example, statistics or literature on driving behavior of traffic participants may be used. The second alternative is to use assumptions on how the future develops given an initial situation x. For example, when assuming that the speed of the leading vehicle in Section 8.4.1 remains constant, it is not needed to estimate $p(y|x)$. Note that a combination is also possible. For example, estimate $p(y|x)$ based on data in case x is well represented in the data, but define $p(y|x)$ on the basis of existing knowledge and/or assumptions for the cases where x is underrepresented in the data.

Note that the PRISMA method is used to derive SSMs that predict the probability of a specific event, such as a crash, i.e., the derived SSMs can be used as a measure of proximity of the specified event. However, the PRISMA method is not used to measure the severity of an interaction, i.e., the extent of harm in case the interaction leads to a crash. For measuring the severity of an interaction, typically energy-based SSMs are used [296]. So, if there is a need to also have an indicator of the severity of an interaction, an energy-based SSM, e.g., see [8, 179, 202, 220], may be considered alongside an SSM derived using the PRISMA method.

We have illustrated the PRISMA method through different derived SSMs in the case study. The derived SSMs estimate the probability of a crash with a leading vehicle under different assumptions. Because of the focus on crashes, the resulting SSMs may still be low given an initial situation that is generally considered to be unsafe. For example, the SSM described in Section 8.4.2 gives a crash probability of approximately $14\,\%$ when approaching a leading vehicle that is driving at a constant speed of $v_\mathrm{l} = 12\,\mathrm{m/s}$ ($a_\mathrm{l} = 0\,\mathrm{m/s^2}$) with a speed of $v_\mathrm{e} = 25\,\mathrm{m/s}$ and a gap of $g = 20\,\mathrm{m}$ (see left heat map in Figure 8.3). In this initial situation, the THW is only $g/v_\mathrm{e} = 0.8\,\mathrm{s}$ and the TTC is only $g/(v_\mathrm{e} - v_\mathrm{l}) = 1.5\,\mathrm{s}$, whereas a THW of less than $1\,\mathrm{s}$ or a TTC of less than $1.5\,\mathrm{s}$ is considered unsafe [289]. In order to put more emphasis on such unsafe situations, different events — instead of crashes — can be considered. For example, we can derive an SSM that estimates the probability that the TTC is below $1\,\mathrm{s}$ within the next five seconds. More research is needed to investigate whether such SSMs can be of practical use, e.g., for evaluating whether a driver is actively pursuing large safety margins.

A few choices have to be made when using the PRISMA method for deriving

SSMs. One such a choice is the set of initial situations $\{x_1, \dots, x_m\}$ for which the probability $\mathbb{P}(\mathrm{C}|x)$ is estimated. Generally speaking, for larger m, the approximation of $\mathbb{P}(\mathrm{C}|x)$ in (8.17) improves. One disadvantage, however, is that more simulation runs are required when m is larger, but because these simulation runs are performed offline, this problem might be solved by, e.g., parallel computing resources. Another disadvantage is that the computational cost of the approximation in (8.17) scales linearly with m. Especially when using this approximation for real-time evaluation of the SSM, this can be a bottleneck. One solution to this is to not use all m initial situations for evaluating (8.17). The intuition is as follows: since (8.17) uses local regression, an initial situation x_k can be removed from the set $\{x_1, \dots, x_m\}$ if all neighboring data points give (approximately) the same probability of the event C, i.e., $|\hat{P}(\mathrm{C}|x_i) - \hat{P}(\mathrm{C}|x_k)|$ is below a threshold for all x_i, $i \neq k$ for which $\|x_i - x_k\|_2$ is below another threshold (assuming that $\hat{P}(\mathrm{C}|x)$ is sufficiently smooth). For example, the SSM that is shown in Figure 8.3, only a few initial situations are required in the upper right region of the heat maps, since the estimated probability is always lower than 0.1.

Another choice is the threshold ϵ that controls the number of simulation runs (N_{sim}) that are used to estimate $\mathbb{P}(\mathrm{C}|x)$. According to (8.16), N_{sim} is increased until the variance of the estimation error is below ϵ, i.e., $\mathrm{Var}\left[\mathbb{P}(\mathrm{C}|x) - \hat{P}(\mathrm{C}|x)\right] < \epsilon$. Therefore, a lower ϵ generally results in more accurate estimations of the probability, as illustrated in Figure 8.1. The downside, however, is that for a lower ϵ, more offline simulation runs are required. Although a good choice of ϵ remains a topic of research, based on experience, we advice to use a maximum threshold of $\epsilon = 0.1$ and lower values if the computational resources allow for this.

In the examples presented in Section 8.4, we have considered the leading vehicle as the only traffic participant other than the ego vehicle. The PRISMA method can be applied in scenarios with multiple traffic participants other than the ego vehicle. However, the number of parameters (n_x, i.e., the size of x) then becomes larger. As a result, two problems may arise. First, as m grows exponentially with n_x, so does the number of simulation runs. Second, even if these simulation runs can be performed, the regression using (8.17) becomes slow due to the large m. To overcome these problems, an SSM can be computed for each traffic participant independently. For example, let N_{tp} denote the number of traffic participants other than the ego vehicle. With $i \in \{1, \dots, N_{\mathrm{tp}}\}$, let C_i denote the event of colliding with the i-th traffic participant and let x^i denote the initial situation considering the i-th traffic participant. Under the assumption that $\mathbb{P}(\mathrm{C}_i|x^i)$ is independent of x^j for all $i \neq j$, we can calculate the probability of colliding with one or more traffic participants using

$$1 - \prod_{i=1}^{N_{\mathrm{tp}}} \left(1 - \mathbb{P}(\mathrm{C}_j|x^j)\right). \tag{8.26}$$

For example, consider a scenario with multiple crossing pedestrians. Using the PRISMA method, we can derive an SSM that estimates the probability of colliding with a pedestrian. Then, after evaluating this SSM for each pedestrian, the probability of colliding with one or more pedestrians can be calculated using (8.26) without

the need for an SSM that considers multiple pedestrians.

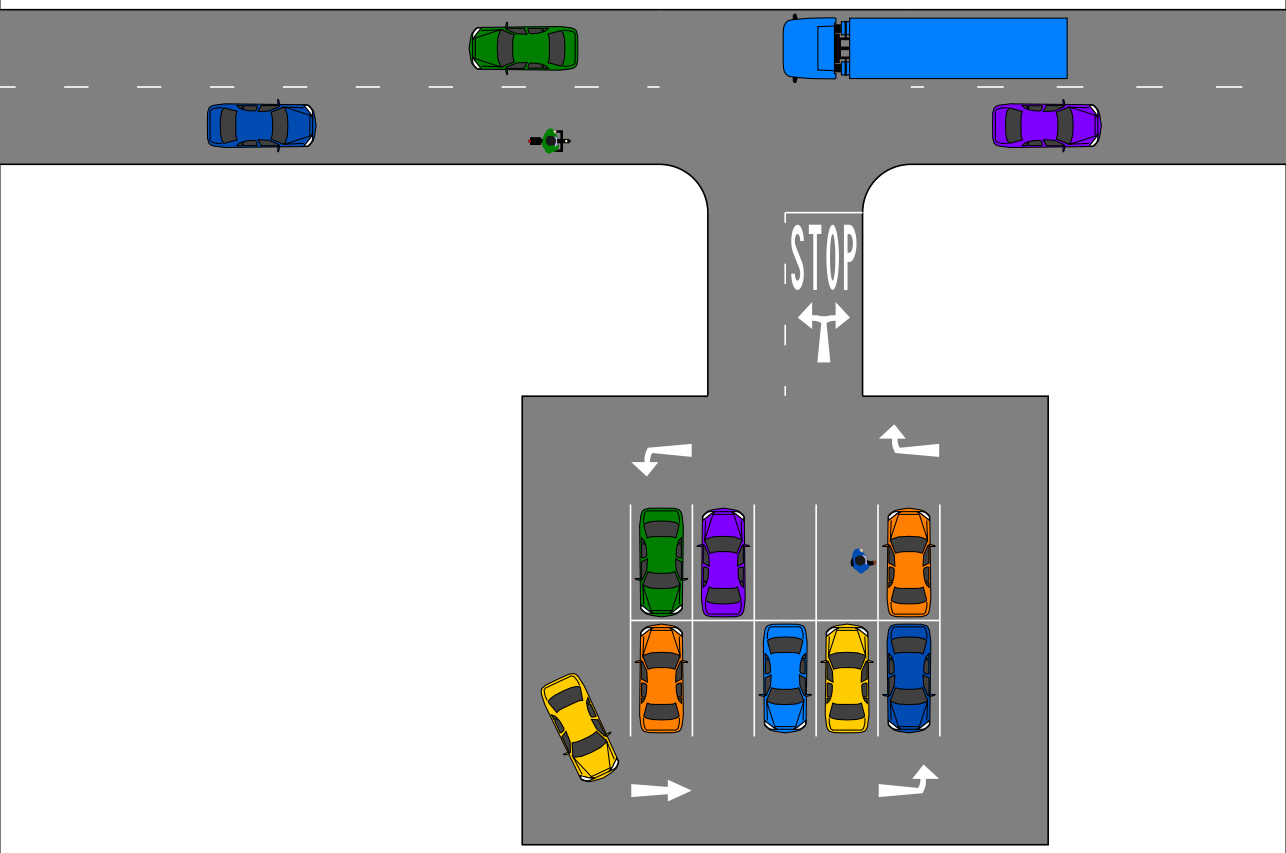
In the case study, we have shown how to analyze an SSM both qualitatively, using heat maps and testing the SSM in different scenarios, and quantitatively by benchmarking the SSM with expected risk trends [203]. Since the SSMs derived using the PRISMA method provide a probability, it is also possible to verify the estimated probability by comparing it with real data. This requires, however, an extensive data set that would allow for estimating the probability of the event C, e.g., a crash, in the near future given a certain situation a vehicle is in. It remains a topic for future work to use such a data set to verify the SSMs derived using the PRISMA method.

8.6. Conclusions

Road safety is an important research topic. To quantify the safety at a vehicle level, Surrogate Safety Measures (SSMs) are often used to characterize the risk of a crash. We have proposed a novel approach called the Probabilistic RISk Measure derivAtion (PRISMA) method for deriving SSMs that calculate the probability that a certain event, e.g., a crash, will happen in the near future given an initial situation. Whereas traditional SSMs are generally only applicable in certain types of scenarios, the PRISMA method can be applied to various types of scenarios. Furthermore, because the PRISMA method is data-driven, the derived SSMs can be adapted to the local traffic behavior that is captured by the data. Also, no assumptions on the driver behavior are made. Therefore, the PRISMA method has the potential for deriving multiple SSMs for quantifying the safety of a — possibly automated — vehicle.

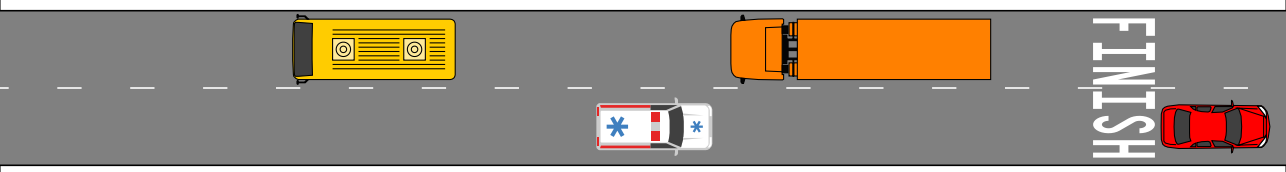
We have illustrated that the PRISMA method can be used to reproduce known probabilistic SSMs. In an example, we have derived a new SSM based on the Next Generation SIMulation (NGSIM) data set that calculates the risk of a crash in a longitudinal interaction between two vehicles. Through several explanatory scenarios, it has been shown that the derived SSM correctly provides a quantification of the crash risk. We have also presented how the evaluation of the partial derivatives of the SSM can be used to benchmark an SSM using expected risk trends.

The SSMs derived using the presented PRISMA method can be used to warn drivers for unsafe situations and ensuring that proper attention is being paid to the road situation. Furthermore, the derived measures can prospectively estimate the impact of newly introduced systems on traffic safety. A limitation of the current study is that the presented approach is only applied to longitudinal traffic interactions. Future work involves applying the PRISMA method for the derivation of SSMs that measure the risk of lateral traffic interactions, interactions with vulnerable road users, and interactions with multiple (different types of) traffic participants. Furthermore, more research is needed to investigate whether the SSMs derived by the PRISMA method can be used to evaluate whether a driver is actively pursuing (large) safety margins.


STOP

9

Conclusions and outlook



This thesis has presented methods for the data-driven scenario-based assessment of Automated Vehicles (AVs). We have proposed a framework for specifying a scenario in the context of the scenario-based assessment of AVs, a method to quantify the degree of completeness of a database with real-world driving data, and a method to extract such scenarios from real-world driving data. Next, this thesis has proposed methods to generate test scenarios for the assessment of AVs based on a set of scenarios extracted from real-world data. Finally, we have introduced novel methods for quantifying the risk of collisions and/or (fatal) injuries, either through a prospective assessment or through Surrogate Safety Measures (SSMs). In this final chapter, we first summarize the main conclusions from the research presented in this dissertation in Section 9.1. Next, in Section 9.2, directions for future research are presented. The full integration of a scenario-based assessment for the type approval of AVs requires — in addition to future research — an open discussion among policy makers, authorities, developers, researchers, and the public, which is why Section 9.3 describes directions for future work that also involves policy decisions.

9.1. Conclusions

This dissertation has aimed to contribute to the development of the data-driven scenario-based assessment of AVs. To report how this dissertation has achieved this, conclusions are drawn for each of the Research questions 1.1 to 1.5:

- **What is a scenario and how to specify a scenario in the context of the scenario-based assessment of AVs?**

  As defined in Definition 2.1 in Chapter 2, a scenario is a quantitative description of the relevant characteristics and activities and/or goals of the ego vehicle(s), the static environment, the dynamic environment, and all events that are relevant to the ego vehicle(s) within the time interval between the first and the last relevant event. In addition, we have defined a scenario category in Definition 2.4 as the qualitative counterpart of a scenario. Furthermore, we have provided definitions for the building blocks of a scenario, such as activities and events. While the provided definitions are consistent with other definitions from the literature, they are more concrete, which has allowed us to formalize the concepts of scenario, event, activity, and scenario category using an Object-Oriented Framework (OOF). The OOF can be directly translated into a class structure for an object-oriented software implementation. This has been demonstrated using a publicly available implementation in the coding language Python[1]. The OOF enables the description of scenarios using software code, such that both domain experts and software programs, such as simulations tools, are able to understand and interpret the content of scenarios.

- **How to quantify whether we have collected enough field data?**

  In Chapter 3, we have presented a metric for quantifying the degree of completeness of a data set, which can be used to quantify whether or not we have

[1] https://github.com/ErwindeGelder/ScenarioDomainModel

collected enough field data. To evaluate the proposed metric, the parameters of activities, such as a vehicle changing lane or decelerating, are extracted from the data. The underlying probability density function (pdf) of these parameters is estimated. The metric for quantifying the degree of completeness is the estimated Asymptotic Mean Integrated Squared Error (AMISE) of the estimated pdf. The smaller the estimated AMISE, the more complete the data set is. By exploiting the logarithmic dependence of the AMISE on the number of samples, the proposed metric can be used to estimate the required amount of data such that the estimated AMISE is below a specified threshold.

- **How to extract a specific type of scenario from a given data set with real-world traffic data and how to easily extend this approach to other types of scenarios?**

  In Chapter 4, we have proposed a two-step approach for mining real-world scenarios from a data set. The first step is to label the data with tags. These tags describe the various aspects of scenarios, e.g., the lateral and longitudinal activities of the different actors, elements of the static environment, and information on the environmental conditions, such as the weather and lighting conditions. The second step mines the scenarios by searching for a particular combination of tags, where this particular combination of tags corresponds to a specific scenario category. Provided that there are no new tags required for the definition of another scenario category, there are no new algorithms required for mining scenarios for other scenario categories, i.e., other types of scenarios. Therefore, the proposed approach is easily extended to other types of scenarios.

- **Based on a set of observed scenarios, how to generate test scenarios for the assessment of AVs without oversimplifying the scenarios?**

  Two different complementary strategies for generating test scenarios are presented in Chapters 5 and 6 of this dissertation. Both these strategies use parameters to describe the observed scenarios and the estimated underlying pdf of these parameters. Using too many parameters leads to inaccurate estimation of the pdf due to the curse of dimensionality, which is why Chapter 5 has proposed the use of a Singular Value Decomposition (SVD) to reduce the number of parameters without loosing essential information. Given d, the SVD automatically determines the d parameters that best describe the scenarios. Next, to not assume a particular shape of the pdf, Kernel Density Estimation (KDE) is used for estimating the pdf of the reduced set of parameters. To quantify to what degree the generated scenarios represent realistic scenarios while covering the same variety that is found in real-world traffic, Chapter 5 has also proposed the novel Scenario Representativeness (SR) metric. The SR metric is based on the Wasserstein metric and compares a set of generated scenarios with a set of observed scenarios. The SR metric can be used to tune and optimize hyperparameters that influence the way test scenarios are generated.

In Chapter 6, a method has been proposed to sample from a KDE such that the generated vectors of parameters satisfy a linear equality constraint. The proposed sampling method can be used to generate test scenarios that satisfy a predetermined condition, e.g., scenarios with a fixed speed of the ego vehicle at the start of a scenario. As demonstrated in Chapter 6, this is particularly useful when using an SVD to reduce the number of parameters, which makes the proposed method for sampling helpful in combination with the proposed method for generated test scenarios in Chapter 5.

- **How to quantify the risk of an AV in real-world scenarios?**

  Chapter 7 has proposed a method to quantify the risk of an AV prospectively, i.e., before the actual deployment of the AV in real-world traffic. Based on recorded data, the likelihood of encountering specific scenarios is estimated. The recorded data are also used to generate test scenarios for virtual simulations of the AV. These virtual simulations are used to estimate the probability and the severity of a collision. Combining this probability with the likelihood of encountering specific scenarios leads to a quantified risk expressed as the expected number of injuries per unit of time. Chapter 7 has also presented how this risk can be decomposed into the terms exposure, severity, and controllability, which are the risk aspects used by the leading standards in automotive safety, ISO 26262 and ISO 21448. The proposed method for quantifying the risk can be used to evaluate whether it is safe to actually deploy an AV in the real-world traffic. Another purpose of the proposed method — especially the decomposition of the risk into the terms exposure, severity, and controllability — is to facilitate the design decisions during the development of an AV.

  The Probabilistic RISk Measure derivAtion (PRISMA), proposed in Chapter 8, is a method for deriving SSMs. The SSMs measure the risk of a — possibly automated — vehicle while operating in real-world traffic. The PRISMA method derives SSMs by pre-calculating the probability of a specified event, e.g., a collision, under various conditions. Based on the pre-calculated probabilities, regression is used to quickly evaluate the SSM for some specific conditions. Whereas traditional SSMs are generally only applicable in certain types of scenarios, the PRISMA method can be applied to various types of scenarios. Therefore, the PRISMA method has the potential for deriving multiple SSMs for quantifying the risk of an AV.

In conclusion, by answering the research questions, this dissertation has presented novel methods for the scenario-based assessment of AVs. Although the research forms a substantial contribution to the full integration of a scenario-based assessment for the type approval of AVs, the research has also led to questions that are to be addressed in future work. The remaining part of this chapter addresses future work that is required in order to fully integrate scenario-based assessment for the type approval of AVs.

9.2. Recommendation for future research

In this section, we discuss directions for future research regarding the scenario-based assessment of Automated Driving Systems (ADSs) and AVs. The directions for future research concern the input data of the scenario-based assessment, the Operational Design Domain (ODD) of the ADS, the explainability of the ADS behavior, over-the-air updated of AVs, and the use of simulators in the scenario-based assessment.

Input data for scenario-based assessment

While we have advertised the use of data, it might be difficult to justify the adequacy of the data. First, whereas we have proposed a metric for measuring the degree of completeness of the data, it remains future work to determine what thresholds for this metric are appropriate. Second, the quality of the data needs to be sufficient. For example, inaccuracies in the data may lead to unrealistic scenarios [57]. To address this, efforts are made to release high-quality data sets (e.g., [33, 171]) or data sets with extensive annotations (e.g., [43]), but it is questionable whether it is feasible to collect enough data in this way.

One solution for the potential shortage of (high-quality) data is the use of accident data (e.g., the German in-depth accident study [99]) or data from event data recorders that only record data of (near-) accidents [108]. The advantage of these sources is that the recorded scenarios typically address situations that are challenging for an AV to deal with. Obviously, however, data from these sources are not identically distributed as naturalistic driving studies. Also, data from different event data recorders may not be identically distributed since different triggers may be used to initiate the data recording. Future work involves the estimation of the underlying statistics of scenarios using non-identically distributed recordings.

Operational design domain

An ADS up to SAE level 4 is designed to operate in a specific ODD. The ODD defines, among others, the spatial areas and the environmental conditions for which the ADS is designed to operate in. One part of the assessment of an ADS should focus on the performance of the ADS in its ODD [125, 166]. For example, if rainy conditions are part of an ADS' ODD, a selection of the test scenarios should consider rainy conditions. One open question is to how to express the ODD. Although initiatives has been started [146, 278], there is no consensus yet on a standardized taxonomy and format for describing an ODD.

Another open question to be addressed in future research is to express to what degree a set of scenarios covers an ODD. Such an expression could then be used to express to what degree a scenario-based assessment covers the ODD of an ADS.

A third open question concerns the assessment of an ADS when exiting its ODD. For higher levels of automation, it is the duty of the ADS to not exit the ODD, to warn the driver in time to take over the control of the vehicle, or to bring the vehicle to a safe, stationary condition. To assess whether the ADS properly fulfills this duty, the ADS needs to be exposed to scenarios that end up at the boundaries of the ADS' ODD with the potential to cross these boundaries. The assessment of an ADS when

it is about to exit its ODD is not covered by this dissertation and requires further research.

Explainability of ADS behavior

We have proposed methods for generating test scenarios and, as we have demonstrated, the generated test scenarios can be used to determine, e.g., if the ADS behaves appropriately. Besides estimating this probability, it would be interesting to be able to *explain* the ADS behavior. For the ADS developer, this explainability provides useful information for improving the ADS performance. Perhaps more importantly, this explainability of the results could also help to ensure that the logic behind sensitive decisions made by AVs are transparent and explainable to the public [34]. One challenge toward the explainability of the ADS behavior that requires more research is the increasing use of machine learning algorithms, see, e.g., [84, 124, 135].

Over-the-air updates

Over-the-air updates are already used by most car manufacturers: new software is sent to the car while the car is parked in order to provide an enhanced experience after successful installation of the software. Typically, the new software concerns small tweaks to, e.g., the infotainment, but with the introduction of AVs, over-the-air updates provide an opportunity to improve the driving performance of the AVs. There is, however, one potential problem: the vehicle with the new software does not correspond to the vehicle that went through the assessment before the type approval of the AV. The scenario-based assessment proposed in this thesis could be used to assess the vehicle with the new software, but it is very cumbersome to do a full assessment after each software update. Future work involves, therefore, the development of (scenario-based) assessment methods that address incremental changes, such as changes after an over-the-air update.

Using virtual simulations for the type approval

Given the complexity of AVs, the required safety level, the required trust that AVs are safe once they are deployed on the public roads, and the costs of physical simulations, it seems that virtual simulations play an increasingly important role in the type approval of AVs [228]. That is one of the reasons for the recent developments regarding virtual simulations of AVs [85, 158, 204, 238, 257]. Although the simulation models and tools get more and more realistic, there are still challenges that are to be addressed in future research [133, 306]. More research is also needed for developing methods for quantifying the fidelity of virtual simulations.

9.3. Additional directions for future work

The road toward the full deployment of AVs with a high automation level does not only require more research. It also requires a debate among policy makers, developers, operators, etc., because new policies need to be drafted to even allow AVs of SAE level 3 and above on public roads. In this section, few aspects that require such a political debate are discussed.

Organizing legal structure for type approval

In addition to the research presented in this thesis, much research is being conducted to provide methods for a reliable evaluation of an AV. When such a reliable evaluation for the type approval of an AV is fully developed, there remain challenges to agree on a procedure to conduct the evaluation. One of the challenges of such a procedure is that proprietary, confidential information regarding the development of the AV must be respected [67]. At the same time, the authority that is responsible for the type approval needs enough confidence in the evaluation. Therefore, the authority needs enough information to support the decision to approve the AV for deployment on public roads.

One of the contradicting interests between the need for information of the authority and the desire to respect the proprietary, confidential information, is the disclosure of detailed test results. Due to the complex ODD, there are many tests required to obtain enough confidence in the evaluation, so virtual simulations become a necessity to limit the expensive physical testing load. The stakeholder responsible for the evaluation might conduct them, which means that the developer has to provide a model of the AV. Even if a black-box model is provided, the detailed test results obtained with the virtual simulations contain proprietary and confidential information. Alternatively, if the developer conducts the virtual simulations, it is conceivable that the developer does not want to disclose the detailed test results because of the proprietary and confidential information contained by the detailed test results. A political debate is needed to determine a procedure that respects the developer's intellectual property and the developer's desire to not disclose too many details regarding the test results while ensuring the authority has enough evidence and confidence that the AV is safe enough. As a starting point for such a debate, the proposed procedure in [67] could be used.

Data sharing

Considering a complex ODD, there is a large set of data needed to conduct the scenario-based assessment as proposed in this dissertation. The question is: how to organize such a data collection? Large data sets are already collected by different organizations in the automotive field, so one solution would be to combine the different efforts made by the different organizations. Instead of sharing the data among different stakeholders, the scenarios that are extracted from the data could be shared in order to prevent the disclosure of propriety information that is contained in the raw data. Technically, this seems feasible, considering the different scenario databases that have been developed [15, 94, 244]. The challenge is to agree with multiple stakeholders, including competitors, on how to organize the scenario collection campaign and on the conditions under which the collected scenarios can be shared with other stakeholders.

In-service monitoring

Once an AV is approved for deployment on public roads, it is expected that the AV is still monitored during its deployment. In this way, the road and/or vehicle authorities can monitor the safety continuously while the AV is in service. To enable the so-called in-service monitoring (or monitored deployment [67]), the operator

of the AV might be required to upload detailed driving data to allow the monitoring of the AV behavior. There are some open questions for which a discussion is needed with different stakeholders, such as AV operators, AV developers, authorities, independent research and development organizations, and independent test institutes:

- What kind of data needs to be uploaded? There is a trade-off between uploading very detailed driving data, which allows a thorough analysis of the operational safety, and uploading only very few measures as to keep as much of the system behavior undisclosed as possible. The former is in interest of the authorities whereas the latter is in interest of the AV developer.
- Who analyzes the data? Both the AV operator and the authorities might have their own motivation to keep the AV in service, so it might be preferred to have an independent organization responsible for the analysis of the data.
- What decisions are made based on the data? For example, if a (near-) accident takes place, does that lead to a temporal withdrawal of the permission to have the AV in service? What are the criteria to revoke the approval for the deployment on public roads?

What is safe enough?

In this dissertation, methods has been proposed to quantify the risk. From safety perspective, it would be best if the quantified risk can be reduced to zero. However, AVs cannot eliminate all accidents. One open question facing authorities, AV developers, and the public, is "how safe is safe enough?" There are several views on this question:

- ISO 26262 [144] and ISO 21448 [143], the leading standards in automotive safety, state that the objective is to keep the system free from unreasonable risk. This is based on the industry practice of keeping the risk "as low as reasonably possible (ALARP)" [26]. This suggests a trade-off between a certain safety benefit and the required effort to reach that certain safety benefit. Although this principle is widely adopted, it cannot directly be used to set a quantified safety target.
- One can argue that as long as AVs cause relatively fewer fatalities than human-driven vehicles, the introduction of AVs itself will lead to safer traffic. This would give a clear safety objective because statistics on the current fatality rate in traffic are available. This is also known as a "positive balance of risks" [189]. Note, however, that for the public acceptance of AVs, the AVs might need to be five times safer than human-driven vehicles [188]. Also note that a positive balance of risks implies that AVs might perform worse than human-driven vehicles in certain scenarios, as long as this is compensated by better performance in other scenarios.
- A proposal for a new UN Regulation concerning the approval of vehicles with an automated lane keeping system, a level 3 ADS, states that an ADS "shall

not cause any collisions that are reasonably foreseeable and preventable" [69, 90]. Here, a collision is assumed to be preventable if a skilled and attentive human driver could prevent the collision. It is not further specified what reasonably foreseeable means.

Bibliography

[1] L. Aarts and I. Van Schagen, *Driving speed and the risk of road crashes: A review,* Accident Analysis & Prevention **38**, 215 (2006).

[2] K. Aas, C. Czado, A. Frigessi, and H. Bakken, *Pair-copula constructions of multiple dependence,* Insurance: Mathematics and Economics **44**, 182 (2009).

[3] H. Abdi and L. J. Williams, *Principal component analysis,* Wiley Interdisciplinary Reviews: Computational Statistics **2**, 433 (2010).

[4] A. Abdulkhaleq, D. Lammering, S. Wagner, J. Röder, N. Balbierer, L. Ramsauer, T. Raste, and H. Boehmert, *A systematic approach based on STPA for developing a dependable architecture for fully automated driving vehicles,* Procedia Engineering **179**, 41 (2017).

[5] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, *A comprehensive survey on vehicular ad hoc network,* Journal of Network and Computer Applications **37**, 380 (2014).

[6] J. Alcamo, *Scenarios as Tools for International Environmental Assessment*, Tech. Rep. Environmental Issue Report No 24 (European Environment Agency, 2001).

[7] V. Alexiadis, J. Colyar, J. Halkias, R. Hranac, and G. McHale, *The next generation simulation program,* Institute of Transportation Engineers. ITE Journal **74**, 22 (2004).

[8] W. K. M. Alhajyaseen, *The integration of conflict probability and severity for the safety assessment of intersections,* Arabian Journal for Science and Engineering **40**, 421 (2015).

[9] B. L. Allen, B. T. Shin, and P. J. Cooper, *Analysis of traffic conflicts and collisions,* Transportation Research Board **667**, 67 (1978).

[10] S. Almqvist, C. Hydén, and R. Risser, *Use of speed limiters in cars for increased safety and a better environment,* Transportation Research Record **1318** (1991).

[11] M. Althoff and S. Lutz, *Automatic generation of safety-critical test scenarios for collision avoidance of road vehicles,* in *IEEE Intelligent Vehicles Symposium (IV)* (2018) pp. 1326–1333.

[12] M. Althoff, M. Koschi, and S. Manzinger, *CommonRoad: Composable benchmarks for motion planning on roads,* in *IEEE Intelligent Vehicles Symposium (IV)* (2017) pp. 719–726.

[13] R. Alur and D. L. Dill, *A theory of timed automata,* Theoretical Computer Science **126**, 183 (1994).

[14] S. Alvarez, Y. Page, U. Sander, F. Fahrenkrog, T. Helmer, O. Jung, T. Hermitte, M. Düering, S. Döering, and O. Op den Camp, *Prospective effectiveness assessment of ADAS and active safety systems via virtual simulation: A review of the current practices,* in *25th International Technical Conference on the Enhanced Safety of Vehicles (ESV)* (2017).

[15] J. Antona-Makoshi, N. Uchida, K. Yamazaki, K. Ozawa, E. Kitahara, and S. Taniguchi, *Development of a safety assurance process for autonomous vehicles in Japan,* in *26th International Technical Conference on the Enhanced Safety of Vehicles (ESV)* (2019) pp. 1–18.

[16] A. Aparicio, S. Baurès, J. Bargalló, C. Rodarius, J. Vissers, O. Bartels, P. Seiniger, P. Lemmen, T. Unselt, M. Ranovona, T. Okawa, and S. Schaub, *Pre-crash performance of collision mitigation and avoidance systems: Results from the ASSESS project,* in *FISITA 2012 World Automotive Congress* (2013) pp. 489–505.

[17] A. Arun, M. M. Haque, A. Bhaskar, S. Washington, and T. Sayed, *A systematic mapping review of surrogate safety assessment using traffic conflict techniques,* Accident Analysis & Prevention **153** (2021), 10.1016/j.aap.2021.106016.

[18] D. Åsljung, J. Nilsson, and J. Fredriksson, *Using extreme value theory for vehicle level safety validation and implications for autonomous vehicles,* IEEE Transactions on Intelligent Vehicles **2**, 288 (2017).

[19] Association for Standardization of Automation and Measuring Systems, *OpenSCENARIO 2.0,* (2020), accessed July 2022.

[20] Association for Standardization of Automation and Measuring Systems, *OpenSCENARIO,* (2021), accessed July, 2022.

[21] J. Augenstein, E. Perdeck, J. Stratton, K. Digges, G. Bahouth, N. Borchers, and P. Baur, *Methodology for the development and validation of injury predicting algorithms,* in *18th ESV Conference*, 467 (2003).

[22] Aurora, *The New Era of Mobility*, Tech. Rep. (Aurora, 2019).

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

[24] D. M. Bashtannyk and R. J. Hyndman, *Bandwidth selection for kernel conditional density estimation,* Computational Statistics & Data Analysis **36**, 279 (2001).

[25] R. Batres, M. West, D. Leal, D. Price, K. Masaki, Y. Shimada, T. Fuchino, and Y. Naka, *An upper ontology based on ISO 15926,* Computers & Chemical Engineering **31**, 519 (2007).

[26] P. Baybutt, *The ALARP principle in process safety,* Process Safety Progress **33**, 36 (2014).

[27] K. Bengler, K. Dietmayer, B. Färber, M. Maurer, C. Stiller, and H. Winner, *Three decades of driver assistance systems: Review and future perspectives,* IEEE Intelligent Transportation Systems Magazine **6**, 6 (2014).

[28] C. Bergenhem, M. Majdandzic, and S. Ursing, *Concepts and risk analysis for a cooperative and automated highway platooning system,* in *European Dependable Computing Conference* (2020) pp. 200–213.

[29] K. Bimbraw, *Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology,* in *12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, Vol. 1 (2015) pp. 191–198.

[30] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, 2006).

[31] P. Bishop, A. Hines, and T. Collins, *The current state of scenario development: An overview of techniques,* Foresight **9**, 5 (2007).

[32] S. N. Blair, M. J. LaMonte, and M. Z. Nichaman, *The evolution of physical activity recommendations: How much is enough?* The American Journal of Clinical Nutrition **79**, 913S (2004).

[33] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, *The inD dataset: A drone dataset of naturalistic road user trajectories at German intersections,* in *IEEE Intelligent Vehicles Symposium (IV)* (2020) pp. 1929–1934.

[34] J.-F. Bonnefon, D. Černy, J. Danaher, N. Devillier, V. Johansson, T. Kovacikova, M. Martens, M. Mladenovic, P. Palade, N. Reed, F. Santoni De Sio, S. Tsinorema, S. Wachter, and K. Zawieska, *Ethics of Connected and Automated Vehicles: Recommendations on Road Safety, Privacy, Fairness, Explainability and Responsibility*, Tech. Rep. (European Commission, 2020).

[35] S. Bonnin, T. H. Weisswange, F. Kummert, and J. Schmuedderich, *General behavior prediction by a combination of scenario-specific models,* IEEE Transactions on Intelligent Transportation Systems **15**, 1478 (2014).

[36] A. Borji, *Pros and cons of GAN evaluation measures,* Computer Vision and Image Understanding **179**, 41 (2019).

[37] M. S. Branicky, V. S. Borkar, and S. K. Mitter, *A unified framework for hybrid control: Model and optimal control theory,* IEEE Transactions on Automatic Control **43**, 31 (1998).

[38] R. Breu, U. Hinkel, C. Hofmann, C. Klein, B. Paech, B. Rumpe, and V. Thurner, *Towards a formalization of the unified modeling language,* in *European Conference on Object-Oriented Programming* (1997) pp. 344–366.

[39] J. C. Brewer, *Effects of angles and offsets in crash simulations of automobiles with light trucks,* in *17th International Technical Conference on the Enhances Safety of Vehicles* (2001).

[40] A. Broggi, M. Buzzoni, S. Debattisti, P. Grisleri, M. C. Laghi, P. Medici, and P. Versari, *Extensive tests of autonomous driving technologies,* IEEE Transactions on Intelligent Transportation Systems **14**, 1403 (2013).

[41] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, *Class-based n-gram models of natural language,* Computational Linguistics **18**, 467 (1992).

[42] J. Bunge and M. Fitzpatrick, *Estimating the number of species: A review,* Journal of the American Statistical Association **88**, 364 (1993).

[43] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, *nuScenes: A multimodal dataset for autonomous driving,* in *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020) pp. 11621–11631.

[44] S. Calonico, M. D. Cattaneo, and M. H. Farrell, *On the effect of bias estimation on coverage accuracy in nonparametric inference,* Journal of the American Statistical Association **113**, 767 (2018).

[45] I. Cara and E. de Gelder, *Classification for safety-critical car-cyclist scenarios using machine learning,* in *IEEE 18th International Conference on Intelligent Transportation Systems* (2015) pp. 1995–2000.

[46] F. Carollo, G. Virzi-Mariotti, and E. Scalici, *Injury evaluation in teenage cyclist-vehicle crash by multibody simulation,* WSEAS Transactions on Biology and Biomedicine **11**, 203 (2014).

[47] S.-H. Cha, *Comprehensive survey on distance/similarity measures between probability density functions,* International Journal of Mathematical Models and Methods in Applied Sciences **1**, 300 (2007).

[48] C.-Y. Chan, *Advancements, prospects, and impacts of automated driving systems,* International Journal of Transportation Science and Technology **6**, 208 (2017).

[49] S. Chan-Edmiston, S. Fischer, S. Sloan, and M. Wong, *Intelligent Transportation Systems (ITS) Joint Program Office: Strategic Plan 2020–2025*, Tech. Rep. FHWA-JPO-18-746 (U.S. Department of Transportation, 2020).

[50] Q. Chao, H. Bi, W. Li, T. Mao, Z. Wang, M. C. Lin, and Z. Deng, *A survey on visual traffic simulation: Models, evaluations, and applications in autonomous driving,* in *Computer Graphics Forum*, Vol. 39 (Wiley Online Library, 2020) pp. 287–308.

[51] W. Chen and L. Kloul, *An ontology-based approach to generate the advanced driver assistance use cases of highway traffic,* in *10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management* (2018).

[52] Y.-C. Chen, *A tutorial on kernel density estimation and recent advances,* Biostatistics & Epidemiology **1**, 161 (2017).

[53] W. M. D. Chia, S. L. Keoh, A. L. Michala, and C. Goh, *Real-time recursive risk assessment framework for autonomous vehicle operations,* in *IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)* (2021) pp. 1–7.

[54] S. Childress, B. Nichols, B. Charlton, and S. Coe, *Using an activity-based model to explore the potential impacts of automated vehicles,* Transportation Research Record **2493**, 99 (2015).

[55] S.-T. Chiu, *A comparative review of bandwidth selection for kernel density estimation,* Statistica Sinica **6**, 129 (1996).

[56] C. J. Clopper and E. S. Pearson, *The use of confidence or fiducial limits illustrated in the case of the binomial,* Biometrika **26**, 404 (1934).

[57] B. Coifman and L. Li, *A critical evaluation of the Next Generation SIMulation (NGSIM) vehicle trajectory dataset,* Transportation Research Part B: Methodological **105**, 362 (2017).

[58] A. Corso and M. J. Kochenderfer, *Interpretable safety validation for autonomous vehicles,* in *IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)* (2020) pp. 1–6.

[59] R. V. Cowlagi, R. C. Debski, and A. M. Wyglinski, *Risk quantification for automated driving using information from V2V basic safety messages,* in *IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)* (2021) pp. 1–5.

[60] D. H. Craft, P. A. Caro, J. Pasqua, and D. C. Brotsky, *Tagging Data Assets*, United States Patent (US 6,704,739 B2, 2004).

[61] D. Craigen, N. Diakun-Thibault, and R. Purse, *Defining cybersecurity,* Technology Innovation Management Review **4**, 13 (2014).

[62] J. Cui, L. S. Liew, G. Sabaliauskaite, and F. Zhou, *A review on safety failures, security attacks, and available countermeasures for autonomous vehicles,* Ad Hoc Networks **90**, 101823 (2019), in press.

[63] F. Cunto and F. F. Saccomanno, *Simulated safety performance of rear-end and angled vehicle interactions at isolated intersections,* Canadian Journal of Civil Engineering **36**, 1794 (2009).

[64] F. J. C. Cunto, *Assessing Safety Performance of Transportation Systems Using Microscopic Simulation*, Ph.D. thesis, University of Waterloo (2008).

[65] G. A. Davis, J. Hourdos, H. Xiong, and I. Chatterjee, *Outline for a causal model of traffic conflicts and crashes,* Accident Analysis & Prevention **43**, 1907 (2011).

[66] C. de Boor, *A Practical Guide to Splines*, Vol. 27 (Springer, 1978).

[67] E. de Gelder and O. Op den Camp, *Procedure for the safety assessment of an autonomous vehicle using real-world scenarios,* in *38th FISITA World Congress*, F2020-VES-014 (2020).

[68] E. de Gelder and O. Op den Camp, *Tagging real-world scenarios for the assessment of autonomous vehicles,* in *38th FISITA World Congress*, F2020-PIF-048 (2020).

[69] E. de Gelder and O. Op den Camp, *A quantitative method to determine what collisions are reasonably foreseeable and preventable,* in *8th Road Safety & Simulation International Conference* (2022).

[70] E. de Gelder and J.-P. Paardekooper, *Assessment of automated driving systems using real-life scenarios,* in *IEEE Intelligent Vehicles Symposium (IV)* (2017) pp. 589–594.

[71] E. de Gelder, A. Khabbaz Saberi, and H. Elrofai, *A method for scenario risk quantification for automated driving systems,* in *26th International Technical Conference on the Enhanced Safety of Vehicles (ESV)* (2019).

[72] E. de Gelder, J.-P. Paardekooper, O. Op den Camp, and B. De Schutter, *Safety assessment of automated vehicles: How to determine whether we have collected enough field data?* Traffic Injury Prevention **20**, 162 (2019).

[73] E. de Gelder, J. Manders, C. Grappiolo, J.-P. Paardekooper, O. Op den Camp, and B. De Schutter, *Real-world scenario mining for the assessment of automated vehicles,* in *IEEE International Transportation Systems Conference (ITSC)* (2020) pp. 1073–1080.

[74] E. de Gelder, O. Op den Camp, and N. de Boer, *Scenario Categories for the Assessment of Automated Vehicles*, Tech. Rep. (CETRAN, 2020) version 1.7.

[75] E. de Gelder, E. Cator, J.-P. Paardekooper, O. Op den Camp, and B. De Schutter, *Constrained sampling from a kernel density estimator to generate scenarios for the assessment of automated vehicles,* in *IEEE Intelligent Vehicles Symposium Workshops (IV Workshop)* (2021) pp. 203–208.

[76] E. de Gelder, H. Elrofai, A. Khabbaz Saberi, O. Op den Camp, J.-P. Paardekooper, and B. De Schutter, *Risk quantification for automated driving systems in real-world driving scenarios,* IEEE Access **9**, 168953 (2021).

[77] E. de Gelder, K. Adjenughwure, J. Manders, R. Snijders, J.-P. Paardekooper, O. Op den Camp, A. Tejada Ruiz, and B. De Schutter, *PRISMA: A novel approach for deriving probabilistic surrogate safety measures for risk evaluation,* Under review (2022).

[78] E. de Gelder, J. Hof, E. Cator, J.-P. Paardekooper, O. Op den Camp, J. Ploeg, and B. De Schutter, *Scenario parameter generation method and scenario representativeness metric for scenario-based assessment of automated vehicles,* IEEE Transactions on Intelligent Transportation Systems **Early access** (2022), 10.1109/TITS.2022.3154774.

[79] E. de Gelder, J.-P. Paardekooper, A. Khabbaz Saberi, H. Elrofai, O. Op den Camp, S. Kraines, J. Ploeg, and B. De Schutter, *Towards an ontology for scenario definition for the assessment of automated vehicles: An object-oriented framework,* IEEE Transactions on Intelligent Vehicles **7**, 300 (2022).

[80] B. De Schutter, *Optimal control of a class of linear hybrid systems with saturation,* SIAM Journal on Control and Optimization **39**, 835 (2000).

[81] T. A. Dingus, F. Guo, S. Lee, J. F. Antin, M. Perez, M. Buchanan-King, and J. Hankey, *Driver crash risk factors and prevalence evaluation using naturalistic driving data,* Proceedings of the National Academy of Sciences **113**, 2636 (2016).

[82] B. Doerr and A. M. Sutton, *When resampling to cope with noise, use median, not mean,* in *Genetic and Evolutionary Computation Conference* (2019) pp. 242–248.

[83] R. C. Dorf and R. H. Bishop, *Modern Control Systems Twelfth Edition* (Prentice Hall, 2011).

[84] F. K. Došilović, M. Brčić, and N. Hlupić, *Explainable artificial intelligence: A survey,* in *41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (2018) pp. 0210–0215.

[85] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, *CARLA: An open urban driving simulator,* in *Conference on Robot Learning* (2017) pp. 1–16.

[86] R. P. W. Duin, *On the choice of smoothing parameters for Parzen estimators of probability density functions,* IEEE Transactions on Computers **C-25**, 1175 (1976).

[87] T. Duong, *ks: Kernel density estimation and kernel discriminant analysis for multivariate data in R,* Journal of Statistical Software **21**, 1 (2007).

[88] M. Dupuis, M. Strobl, and H. Grezlikowski, *Opendrive 2010 and beyond – status and future of the de facto standard for the description of road networks,* in *Driving Simulation Conference Europe* (2010) pp. 231–242.

[89] A. Ebner, T. Helmer, R. R. Samaha, and P. Scullion, *Identifying and analyzing reference scenarios for the development and evaluation of active safety: Application to preventive pedestrian safety,* International Journal of Intelligent Transportation Systems Research **9**, 128 (2011).

[90] ECE/TRANS/WP.29/2020/81, *Proposal for a New UN Regulation on Uniform Provisions Concerning the Approval of Vehicles with Regards to Automated Lane Keeping System*, Standard (World Forum for Harmonization of Vehicle Regulations, 2020).

[91] B. Efron, *Bootstrap methods: Another look at the jackknife,* Annals of Statistics **7**, 1 (1979).

[92] J. Elfring, R. Appeldoorn, S. van den Dries, and M. Kwakkernaat, *Effective world modeling: Multisensor data fusion methodology for automated driving,* Sensors **16**, 1 (2016).

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

[94] H. Elrofai, J.-P. Paardekooper, E. de Gelder, S. Kalisvaart, and O. Op den Camp, *Scenario-Based Safety Validation of Connected and Automated Driving*, Tech. Rep. (Netherlands Organization for Applied Scientific Research, TNO, 2018).

[95] A. Eskandarian, *Introduction to intelligent vehicles,* in *Handbook of Intelligent Vehicles* (Springer London, London, 2012) Chap. 1, pp. 1–13.

[96] European Commission, *Road Safety Targets — Monitoring Report June 2020*, Tech. Rep. (European Road Safety Observatory, 2020).

[97] L. Evans, *Driver injury and fatality risk in two-car crashes versus mass ratio inferred using Newtonian mechanics,* Accident Analysis & Prevention **26**, 609 (1994).

[98] F. Fahrenkrog, L. Wang, C. Roesener, J. Sauerbier, and S. Breunig, *Impact Analysis for Supervised Automated Driving Applications*, Tech. Rep. Deliverable D7.3 (AdaptIVe, 2017).

[99] Federal Highway Research Institute, *Accident research — GIDAS,* (2021), accessed July 2022.

[100] S. Feng, Y. Feng, X. Yan, S. Shen, S. Xu, and H. X. Liu, *Safety assessment of highly automated driving systems in test tracks: A new framework,* Accident Analysis & Prevention **144**, 105664 (2020).

[101] S. Feng, Y. Feng, C. Yu, Y. Zhang, and H. X. Liu, *Testing scenario library generation for connected and automated vehicles, part I: Methodology,* IEEE Transactions on Intelligent Transportation Systems **22**, 1573 (2020).

[102] S. Feng, X. Yan, H. Sun, Y. Feng, and H. X. Liu, *Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment,* Nature Communications **12**, 1 (2021).

[103] V. Fontana, G. Singh, S. Akrigg, M. Di Maio, S. Saha, and F. Cuzzolin, *Action detection from a robot-car perspective,* arXiv preprint arXiv:1807.11332 (2018).

[104] A. B. Frost, *MUSICC: Multi User Scenario Catalogue for CAVs, Glossary of Terms and Definitions*, Tech. Rep. (Catapult Transport Systems, 2018).

[105] D. J. Gabauer and H. C. Gabler, *Acceleration-based occupant injury criteria for predicting injury in real-world crashes,* Biomedical Sciences Instrumentation **44**, 213 (2008).

[106] D. J. Gabauer and H. C. Gabler, *Comparison of delta-v and occupant impact velocity crash severity metrics using event data recorders,* in *Association for the Advancement of Automotive Medicine*, Vol. 50 (2006) pp. 57–71.

[107] D. J. Gabauer, *Predicting Occupant Injury with Vehicle-Based Injury Criteria in Roadside Crashes*, Ph.D. thesis, Virginia Tech (2008).

[108] H. C. Gabler, D. J. Gabauer, H. L. Newell, and M. E. O'Neill, *Use of Event Data Recorder (EDR) Technology for Highway Crash Data Analysis*, Tech. Rep. NCHRP Project 17-24 (National Cooperative Highway Research Program, 2004).

[109] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design patterns: Abstraction and reuse of object-oriented design,* in *European Conference on Object-Oriented Programming* (1993) pp. 406–431.

[110] A. Gandolfi and C. C. A. Sastri, *Nonparametric estimations about species not observed in a random sample,* Milan Journal of Mathematics **72**, 81 (2004).

[111] General Motors, *Self-Driving Safety Report*, Tech. Rep. (General Motors, 2018).

[112] S. Geyer, M. Baltzer, B. Franz, S. Hakuli, M. Kauer, M. Kienle, S. Meier, T. Weißgerber, K. Bengler, R. Bruder, F. Flemisch, and H. Winner, *Concept and development of a unified ontology for generating test and use-case catalogues for assisted and automated vehicle guidance,* IET Intelligent Transport Systems **8**, 183 (2014).

[113] O. Gietelink, *Design and Validation of Advanced Driver Assistance Systems*, Ph.D. thesis, Delft University of Technology (2007).

[114] O. Gietelink, J. Ploeg, B. De Schutter, and M. Verhaegen, *Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations,* Vehicle System Dynamics **44**, 569 (2006).

[115] K. Go and J. M. Carroll, *The blind men and the elephant: Views of scenario-based system design,* Interactions **11**, 44 (2004).

[116] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Vol. 3 (John Hopkins University Press, 2013).

[117] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets,* in *27th International Conference on Neural Information Processing Systems*, Vol. 2 (2014) pp. 2672–2680.

[118] L. A. Goodman, *On the exact variance of products,* Journal of the American Statistical Association **55**, 708 (1960).

[119] A. Gramacki, *Nonparametric Kernel Density Estimation and Its Computational Aspects*, edited by J. Kacprzyk (Springer, 2018).

[120] A. Gramacki and J. Gramacki, *FFT-based fast bandwidth selector for multivariate kernel density estimation,* Computational Statistics & Data Analysis **106**, 27 (2017).

[121] M. Green, *"How long does it take to stop?" Methodological analysis of driver perception-brake times,* Transportation Human Factors **2**, 195 (2000).

[122] G. Guest, A. Bunce, and L. Johnson, *How many interviews are enough? An experiment with data saturation and variability,* Field Methods **18**, 59 (2006).

[123] G. Guido, F. Saccomanno, A. Vitale, V. Astarita, and D. Festa, *Comparing safety performance measures obtained from video capture data,* Journal of transportation engineering **137**, 481 (2011).

[124] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang, *Xai—explainable artificial intelligence,* Science Robotics **4**, eaay7120 (2019).

[125] M. Gyllenhammar, R. Johansson, F. Warg, D. Chen, H.-M. Heyn, M. Sanfridson, J. Söderberg, A. Thorsén, and S. Ursing, *Towards an operational design domain that supports the safety argumentation of an automated driving system,* in *10th European Congress on Embedded Real Time Systems (ERTS)* (2020).

[126] A. S. Hakkert, L. Braimaister, and I. Van Schagen, *The Uses of Exposure and Risk in Road Safety Studies*, Tech. Rep. R-2002-12 (SWOV Institute for Road Safety, 2002).

[127] P. Hall, *On global properties of variable bandwidth density estimators,* The Annals of Statistics **20**, 762 (1992).

[128] P. Hall and B. Presnell, *Density estimation under constraints,* Journal of Computational and Graphical Statistics **8**, 259 (1999).

[129] F. Hauer, T. Schmidt, B. Holzmüller, and A. Pretschner, *Did we test all scenarios for automated and autonomous driving systems?* in *IEEE Intelligent Transportation Systems Conference (ITSC)* (2019) pp. 2950–2955.

[130] J. C. Hayward, *Near Miss Determination Through Use of a Scale of Danger*, Tech. Rep. TTSC-7115 (Pennsylvania State University, 1972).

[131] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, *An introduction to event-triggered and self-triggered control,* in *51st IEEE Conference on Decision and Control* (2012) pp. 3270–3285.

[132] T. Helmer, K. Kompaß, L. Wang, T. Kühbeck, and R. Kates, *Safety performance assessment of assisted and automated driving in traffic: Simulation as knowledge synthesis,* in *Automated Driving: Safer and More Efficient Future Driving*, edited by D. Watzenig and M. Horn (Springer International Publishing, 2017) pp. 473–494.

[133] M. Holder, P. Rosenberger, H. Winner, T. D'hondt, V. P. Makkapati, M. Maier, H. Schreiber, Z. Magosi, Z. Slavik, O. Bringmann, *et al.*, *Measurements revealing challenges in radar sensor modeling for virtual validation of autonomous driving,* in *21st International Conference on Intelligent Transportation Systems (ITSC)* (2018) pp. 2616–2622.

[134] M. P. Holmes, A. G. Gray, and C. L. Isbell, *Fast nonparametric conditional density estimation,* in *23rd Conference on Uncertainty in Artificial Intelligence* (2007) pp. 175–182.

[135] A. Holzinger, *From machine learning to explainable AI,* in *World Symposium on Digital Intelligence for Systems and Machines (DISA)* (2018) pp. 55–66.

[136] C. M. Hruschka, D. Töpfer, and S. Zug, *Risk assessment for integral safety in automated driving,* in *2019 2nd International Conference on Intelligent Autonomous Systems (ICoIAS)* (2019) pp. 102–109.

[137] W. Huang, D. Wen, J. Geng, and N.-N. Zheng, *Task-specific performance evaluation of UGVs: Case studies at the IVFC,* IEEE Transactions on Intelligent Transportation Systems **15**, 1969 (2014).

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

[139] J. J. Hull and S. N. Srihari, *Experiments in text recognition with binary n-gram and viterbi algorithms,* IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-4**, 520 (1982).

[140] W. Hulshof, I. Knight, A. Edwards, M. Avery, and C. Grover, *Autonomous emergency braking test results,* in *23rd International Technical Conference on the Enhanced Safety of Vehicles (ESV)* (2013) pp. 1–13.

[141] Y. Iida, N. Uno, S. Itsubo, and M. Suganuma, *Traffic conflict analysis and modeling of lane-changing behavior at weaving section,* in *Infrastructure Planning* (2001) pp. 305–308.

[142] ISO 21434, *Road Vehicles – Cybersecurity Engineering*, Standard (International Organization for Standardization, 2021).

[143] ISO 21448, *Road Vehicles – Safety of the intended functionality*, Standard (International Organization for Standardization, 2021).

[144] ISO 26262, *Road Vehicles – Functional Safety*, Standard (International Organization for Standardization, 2018).

[145] ISO 8855, *Road Vehicles – Vehicle dynamics and road-holding ability*, Standard (International Organization for Standardization, 2011).

[146] ISO/DIS 34503, *Road Vehicles – Test Scenarios for Automated Driving Systems – Taxonomy for Operational Design Domain for Automated Driving Systems*, Standard (International Organization for Standardization, 2022).

[147] ISO/FDIS 34501, *Road Vehicles – Test Scenarios for Automated Driving Systems – Vocabulary*, Standard (International Organization for Standardization, 2022).

[148] ISO/FDIS 34502, *Road Vehicles – Test Scenarios for Automated Driving Systems – Engineering Framework and Process of Scenario-Based Safety Evaluation*, Standard (International Organization for Standardization, 2022).

[149] S. Jesenski, N. Tiemann, J. E. Stellet, and J. M. Zöllner, *Scalable generation of statistical evidence for the safety of automated vehicles by the use of importance sampling,* in *IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)* (2020) pp. 1–8.

[150] R. E. Johnson and B. Foote, *Designing reusable classes,* Journal of Object-Oriented Programming **1**, 22 (1988).

[151] M. C. Jones, J. S. Marron, and S. J. Sheather, *A brief survey of bandwidth selection for density estimation,* Journal of the American Statistical Association **91**, 401 (1996).

[152] C. R. Jung and C. R. Kelber, *A lane departure warning system based on a linear-parabolic lane model,* in *IEEE Intelligent Vehicles Symposium* (2004) pp. 891–895.

[153] C. Jurewicz, A. Sobhani, J. Woolley, J. Dutschke, and B. Corben, *Exploration of vehicle impact speed — injury severity relationships for application in safer road design,* Transportation Research Procedia **14**, 4247 (2016).

[154] H. Kahn, *Thinking the Unthinkable* (New York: Horizon Press, 1962).

[155] N. Kalra and S. M. Paddock, *Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?* Transportation Research Part A: Policy and Practice **94**, 182 (2016).

[156] J. Kapinski, J. V. Deshmukh, X. Jin, H. Ito, and K. Butts, *Simulation-based approaches for verification of embedded control systems: An overview of traditional and advanced modeling, testing, and verification techniques,* IEEE Control Systems Magazine **36**, 45 (2016).

[157] D. Kasper, G. Weidl, T. Dang, G. Breuel, A. Tamke, A. Wedel, and W. Rosenstiel, *Object-oriented Bayesian networks for detection of lane change maneuvers,* IEEE Intelligent Transportation Systems Magazine **4**, 19 (2012).

[158] P. Kaur, S. Taghavi, Z. Tian, and W. Shi, *A survey on simulators for testing self-driving cars,* arXiv preprint arXiv:2101.05337 (2021).

[159] A. Kesting, M. Treiber, and D. Helbing, *General lane-changing model MOBIL for car-following models,* Transportation Research Record **1999**, 86 (2007).

[160] M. A. Khan and K. Salah, *IoT security: Review, blockchain solutions, and open challenges,* Future Generation Computer Systems **82**, 395 (2018).

[161] S. Khastgir, S. Birrell, G. Dhadyalla, H. Sivencrona, and P. Jennings, *Towards increased reliability by objectification of Hazard Analysis and Risk Assessment (HARA) of automated automotive systems,* Safety Science **99**, 166 (2017).

[162] R. J. Kiefer, D. J. LeBlanc, and C. A. Flannagan, *Developing an inverse time-to-collision crash alert timing approach based on drivers' last-second braking and steering judgments,* Accident Analysis & Prevention **37**, 295 (2005).

[163] S. G. Klauer, T. A. Dingus, V. L. Neale, J. D. Sudweeks, and D. J. Ramsey, *The Impact of Driver Inattention on Near-Crash/Crash Risk: An Analysis Using the 100-Car Naturalistic Driving Study Data*, Tech. Rep. DOT HS 810 594 (Virginia Tech Transportation Institute, 2006).

[164] A. Knapp, M. Neumann, M. Brockmann, R. Walz, and T. Winkle, *Code of Practice for the Design and Evaluation of ADAS*, Tech. Rep. RESPONSE III: a PReVENT Project (The European Automobile Manufacturers' Association (ACEA), 2009).

[165] T. F. Kone, E. Bonjour, E. Levrat, F. Mayer, and S. Géronimi, *Safety demonstration of autonomous vehicles: A review and future research questions,* in *International Conference on Complex Systems Design & Management* (2019) pp. 176–188.

[166] P. Koopman and F. Fratrik, *How many operational design domains, objects, and events?* in *AAAI Workshop on Artificial Intelligence Safety* (2019) pp. 45–48.

[167] P. Koopman and M. Wagner, *Challenges in autonomous vehicle testing and validation,* SAE International Journal of Transportation Safety **4**, 15 (2016).

[168] P. Koopman and M. Wagner, *Autonomous vehicle safety: An interdisciplinary challenge,* IEEE Intelligent Transportation Systems Magazine **9**, 90 (2017).

[169] M. Koren, S. Alsaif, R. Lee, and M. J. Kochenderfer, *Adaptive stress testing for autonomous vehicles,* in *IEEE Intelligent Vehicles Symposium (IV)* (2018) pp. 1898–1904.

[170] V. G. Kovvali, V. Alexiadis, and L. Zhang, *Video-based vehicle trajectory data collection,* in *Transportation Research Board 86th Annual Meeting* (2007).

[171] 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 *IEEE 21st International Conference on Intelligent Transportations Systems (ITSC)* (2018) pp. 2118–2125.

[172] B. Kramer, C. Neurohr, M. Büker, E. Böde, M. Fränzle, and W. Damm, *Identification and quantification of hazardous scenarios for automated driving,* in *International Symposium on Model-Based Safety and Assessment* (2020) pp. 163–178.

[173] S. Kullback and R. A. Leibler, *On information and sufficiency,* The Annals of Mathematical Statistics **22**, 79 (1951).

[174] K. D. Kusano and H. C. Gabler, *Potential occupant injury reduction in pre-crash system equipped vehicles in the striking vehicle of rear-end crashes,* in *Annals of Advances in Automotive Medicine/Annual Scientific Conference*, Vol. 54 (2010).

[175] K. D. Kusano and H. C. Gabler, *Safety benefits of forward collision warning, brake assist, and autonomous braking systems in rear-end collisions,* IEEE Transactions on Intelligent Transportation Systems **13**, 1546 (2012).

[176] U. Lages, M. Spencer, and R. Katz, *Automatic scenario generation based on laserscanner reference data and advanced offline processing,* in *IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)* (2013) pp. 146–148.

[177] A. Lara, J. Skvarce, H. Feifel, M. Wagner, and T. Tengeiji, *Harmonized pre-crash scenarios for reaching global vision zero,* in *26th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, 19-0110 (2019).

[178] A. Laureshyn, Å. Svensson, and C. Hydén, *Evaluation of traffic safety, based on micro-level behavioural data: Theoretical framework and first implementation,* Accident Analysis & Prevention **42**, 1637 (2010).

[179] A. Laureshyn, T. De Ceunynck, C. Karlsson, Å. Svensson, and S. Daniels, *In search of the severity dimension of traffic events: Extended Delta-V as a traffic conflict indicator,* Accident Analysis & Prevention **98**, 46 (2017).

[180] J. Lenard, A. Badea-Romero, and R. Danton, *Typical pedestrian accident scenarios for the development of autonomous emergency braking test protocols,* Accident Analysis & Prevention **73**, 73 (2014).

[181] G. Li, Y. Yang, S. Li, X. Qu, N. Lyu, and S. E. Li, *Decision making of autonomous vehicles in lane change scenarios: Deep reinforcement learning approaches with risk awareness,* Transportation Research Part C: Emerging Technologies , 103452 (2021).

[182] G. Li, Y. Yang, T. Zhang, X. Qu, D. Cao, B. Cheng, and K. Li, *Risk assessment based collision avoidance decision-making for autonomous vehicles in multi-scenarios,* Transportation Research Part C: Emerging Technologies **122**, 102820 (2021).

[183] L. Li, W.-L. Huang, Y. Liu, N.-N. Zheng, and F.-Y. Wang, *Intelligence testing for autonomous vehicles: A new approach,* IEEE Transactions on Intelligent Vehicles **1**, 158 (2016).

[184] L. Li, N. Zheng, and F.-Y. Wang, *A theoretical foundation of intelligence testing and its application for intelligent vehicles,* IEEE Transactions on Intelligent Transportation Systems **22**, 6297 (2020).

[185] Y. Li, J. Tao, and F. Wotawa, *Ontology-based test generation for automated and autonomous driving functions,* Information and Software Technology **117**, 106200 (2020).

[186] G. Liu, M. Zhou, L. Wang, H. Wang, and X. Guo, *A blind spot detection and warning system based on millimeter wave radar for driver assistance,* Optik **135**, 353 (2017).

[187] J. Liu, J. Yu, and S. Shen, *Energy-efficient two-layer cooperative defense scheme to secure sensor-clouds,* IEEE Transactions on Information Forensics and Security **13**, 408 (2017).

[188] P. Liu, R. Yang, and Z. Xu, *How safe is safe enough for self-driving vehicles?* Risk Analysis **39**, 315 (2019).

[189] C. Luetge, *The German ethics code for automated and connected driving,* Philosophy & Technology **30**, 547 (2017).

[190] A. M. Madni, *Autonomous system-of-systems,* in *Transdisciplinary Systems Engineering* (Springer, 2018) pp. 161–186.

[191] I. Mahdinia, R. Arvin, A. J. Khattak, and A. Ghiasi, *Safety, energy, and emissions impacts of adaptive cruise control and cooperative adaptive cruise control,* Transportation Research Record **2674**, 253 (2020).

[192] S. M. S. Mahmud, L. Ferreira, M. S. Hoque, and A. Tavassoli, *Application of proximal surrogate indicators for safety evaluation: A review of recent developments and research needs,* IATSS Research **41**, 153 (2017).

[193] A. Mammeri, G. Lu, and A. Boukerche, *Design of lane keeping assist system for autonomous vehicles,* in *7th International Conference on New Technologies, Mobility and Security (NTMS)* (2015) pp. 1–5.

[194] I. H. Marks, Z. V. Fong, S. M. Stapleton, Y.-C. Hung, Y. J. Bababekov, and D. C. Chang, *How much data are good enough? using simulation to determine the reliability of estimating pomr for resource-constrained settings,* World Journal of Surgery **42**, 2344 (2018).

[195] J. S. Marron and M. P. Wand, *Exact mean integrated squared error,* The Annals of Statistics **20**, 712 (1992).

[196] T. Menzel, G. Bagschik, and M. Maurer, *Scenarios for development, test and validation of automated vehicles,* in *IEEE Intelligent Vehicles Symposium (IV)* (2018) pp. 1821–1827.

[197] B. Meyer, *Reusability: The case for object-oriented design,* IEEE Software **4**, 50 (1987).

[198] D. Milakis, M. Snelder, B. van Arem, B. van Wee, and G. H. de Almeida Correia, *Scenarios about development and implications of automated vehicles in the Netherlands,* in *95th Annual Meeting Transportation Research Board* (2016).

[199] V. Milanés and S. E. Shladover, *Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data,* Transportation Research Part C: Emerging Technologies **48**, 285 (2014).

[200] M. M. Minderhoud and P. H. Bovy, *Extended time-to-collision measures for road traffic safety assessment,* Accident Analysis & Prevention **33**, 89 (2001).

[201] S. Molloy, T. Ramos, and S. Thakar, *Dynamic Hierarchical Tagging System and Method*, United States Patent (US 9,613,099 B2, 2017).

[202] F. A. Mullakkal-Babu, M. Wang, X. He, B. van Arem, and R. Happee, *Probabilistic field approach for motorway driving risk assessment,* Transportation Research Part C: Emerging Technologies **118**, 102716 (2020).

[203] F. A. Mullakkal-Babu, M. Wang, H. Farah, B. van Arem, and R. Happee, *Comparative assessment of safety indicators for vehicle trajectories on highways,* Transportation Research Record **2659**, 127 (2017).

[204] M. Müller, V. Casser, J. Lahoud, N. Smith, and B. Ghanem, *Sim4CV: A photo-realistic simulator for computer vision applications,* International Journal of Computer Vision **126**, 902 (2018).

[205] G. E. Mullins, P. G. Stankiewicz, R. C. Hawthorne, and S. K. Gupta, *Adaptive generation of challenging scenarios for testing and evaluation of autonomous vehicles,* Journal of Systems and Software **137**, 197 (2018).

[206] R. Myers and Z. Saigol, *Pass-fail criteria for scenario-based testing of automated driving systems,* arXiv preprint arXiv:2005.09417 (2020).

[207] E. A. Nadaraya, *Some new estimates for distribution functions,* Theory of Probability & Its Applications **9**, 497 (1964).

[208] T. Nagler and C. Czado, *Evading the curse of dimensionality in nonparametric density estimation with simplified vine copulas,* Journal of Multivariate Analysis **151**, 69 (2016).

[209] W. G. Najm, J. D. Smith, and M. Yanagisawa, *DOT HS*, Tech. Rep. DOT HS 810 767 (U.S. Department of Transportation Research and Innovative Technology Administration, 2007).

[210] National Center for Statistics and Analysis, *Summary of Motor Vehicle Crashes: 2018 data*, Tech. Rep. DOT HS 812 961 (National Highway Traffic Safety Administration, 2020).

[211] A. Nehemiah, P. Fryscak, and M. Sasena, *Building an Autonomous Vehicle (AV) Simulation Toolchain with Simulink, RoadRunner and NVIDIA DRIVE Sim*, Blog post (Mathworks, 2021).

[212] C. Neurohr, L. Westhofen, M. Butz, M. Bollmann, U. Eberle, and R. Galbas, *Criticality analysis for the verification and validation of automated vehicles,* IEEE Access **9**, 18016 (2021).

[213] P. Nitsche, R. H. Welsh, A. Genser, and P. D. Thomas, *A novel, modular validation framework for collision avoidance of automated vehicles at road junctions,* in *21st International Conference on Intelligent Transportation Systems (ITSC)* (2018) pp. 90–97.

[214] S. N. Norman, *Control Systems Engineering* (John Wiley & Sons, 2011).

[215] A. H. Oh and A. I. Rudnicky, *Stochastic natural language generation for spoken dialog systems,* Computer Speech & Language **16**, 387 (2002).

[216] O. Op den Camp, A. Ranjbar, J. Uittenbogaard, E. Rosen, R. Fredriksson, and S. de Hair, *Overview of main accident scenarios in car-to-cyclist accidents for use in AEB-system test protocol,* in *International Cycling Safety Conference* (2014).

[217] O. Op den Camp, J. van de Sluis, E. de Gelder, and I. Yalcinkaya, *Generation of tests for safety assessment of V2V platooning trucks,* in *27th ITS World Congress*, 70 (2021).

[218] OpenStreetMaps, *Key:highway,* (2021), accessed July 2022.

[219] A. B. Owen, *Monte Carlo theory, methods and examples,* (2013).

[220] K. Ozbay, H. Yang, B. Bartin, and S. Mudigonda, *Derivation and validation of new simulation-based surrogate safety measure,* Transportation Research Record **2083**, 105 (2008).

[221] J.-P. Paardekooper, S. Montfort, J. Manders, J. Goos, E. de Gelder, O. Op den Camp, A. Bracquemond, and G. Thiolon, *Automatic identification of critical scenarios in a public dataset of 6000 km of public-road driving,* in *26th International Technical Conference on the Enhanced Safety of Vehicles (ESV)* (2019).

[222] E. Parzen, *On estimation of a probability density function and mode,* The Annals of Mathematical Statistics **33**, 1065 (1962).

[223] C. Patridge, *Business Objects: Re-Engineering for Re-Use* (The BORO Centre, 2005).

[224] Y. Peng, Y. Chen, J. Yang, D. Otte, and R. Willinger, *A study of pedestrian and bicyclist exposure to head injury in passenger car collisions based on accident data and simulations,* Safety Science **50**, 1749 (2012).

[225] E. Petrucelli, J. D. States, and L. N. Hames, *The abbreviated injury scale: Evolution, usage and future adaptability,* Accident Analysis & Prevention **13**, 29 (1981).

[226] P. E. Pfeiffer, *Concepts of Probability Theory* (Courier Corporation, 2013).

[227] J. Piao, M. McDonald, N. Hounsell, M. Graindorge, T. Graindorge, and N. Malhene, *Public views towards implementation of automated vehicles in urban areas,* Transportation Research Procedia **14**, 2168 (2016).

[228] J. Ploeg, E. de Gelder, M. Slavík, E. Querner, T. Webster, and N. de Boer, *Scenario-based safety assessment framework for automated vehicles,* in *16th ITS Asia-Pacific Forum* (2018) pp. 713–726.

[229] A. Pütz, A. Zlocki, J. Bock, and L. Eckstein, *System validation of highly automated vehicles with a database of relevant traffic scenarios,* in *12th ITS European Congress* (2017) pp. 1–8.

[230] I. A. Rafukka, B. B. Sahari, A. Nuraini, and A. Manohar, *Child dummy finite element models development: A review,* ARPN Journal of Engineering and Applied Sciences **11**, 6649 (2006).

[231] F. Remmen, I. Cara, E. de Gelder, and D. Willemsen, *Cut-in scenario prediction for automated vehicles,* in *IEEE International Conference on Vehicular Electronics and Safety (ICVES)* (2018).

[232] S. Riedmaier, J. Nesensohn, C. Gutenkunst, T. Düser, B. Schick, and H. Abdellatif, *Validation of x-in-the-loop approaches for virtual homologation of automated driving functions,* in *11th Graz Symposium Virtual Vehicle* (2018).

[233] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, *Survey on scenario-based safety assessment of automated vehicles,* IEEE Access **8**, 87456 (2020).

[234] C. Roesener, F. Fahrenkrog, A. Uhlig, and L. Eckstein, *A scenario-based assessment approach for automated driving by using time series classification of human-driving behaviour,* in *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)* (2016) pp. 1360–1365.

[235] C. Roesener, J. Sauerbier, A. Zlocki, F. Fahrenkrog, L. Wang, A. Várhelyi, E. de Gelder, J. Dufils, S. Breunig, P. Mejuto, F. Tango, and J. Lanati, *A comprehensive evaluation approach for highly automated driving,* in *25th International Technical Conference on the Enhanced Safety of Vehicles (ESV)* (2017).

[236] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, E. Agafonov, T. H. Kim, E. Sterner, K. Ushiroda, M. Reyes, D. Zelenkovsky, and S. Kim, *LGSVL simulator: A high fidelity simulator for autonomous driving,* in *IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)* (2020).

[237] M. Rosenblatt, *Remarks on some nonparametric estimates of a density function,* The Annals of Mathematical Statistics **27**, 832 (1956).

[238] F. Rosique, P. J. Navarro, C. Fernández, and A. Padilla, *A systematic review of perception system and simulators for autonomous vehicles research,* Sensors **19**, 648 (2019).

[239] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo Method* (John Wiley & Sons, 2016).

[240] Y. Rubner, C. Tomasi, and L. J. Guibas, *The earth mover's distance as a metric for image retrieval,* International Journal of Computer Vision **40**, 99 (2000).

[241] L. Rüschendorf, *The Wasserstein distance and approximation theorems,* Probability Theory and Related Fields **70**, 117 (1985).

[242] D. Sadigh, K. Driggs-Campbell, A. Puggelli, W. Li, V. Shia, R. Bajcsy, A. L. Sangiovanni-Vincentelli, S. S. Sastry, and S. A. Seshia, *Data-driven probabilistic modeling and verification of human driver behavior,* in *AAAI Spring Symposium Series* (2014).

[243] SAE J3016, *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, Tech. Rep. (SAE International, 2021).

[244] Safety Pool, *The global initiative for certifiable AV safety,* (2021), accessed July 2022.

[245] Z. Saigol, A. Peters, M. Barton, and M. Taylor, *Regulating and Accelerating Development of Highly Automated and Autonomous Vehicles Through Simulation and Modelling*, Tech. Rep. (Catapult Transport Systems, 2018).

[246] F. Sakiz and S. Sen, *A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV,* Ad Hoc Networks **61**, 33 (2017).

[247] W. J. Schakel, B. van Arem, and B. D. Netten, *Effects of cooperative adaptive cruise control on traffic flow stability,* in *13th International IEEE Conference on Intelligent Transportation Systems* (2010) pp. 759–764.

[248] J. Schlechtriemen, A. Wedel, J. Hillenbrand, G. Breuel, and K.-D. Kuhnert, *A lane change detection approach using feature ranking with maximized predictive power,* in *IEEE Intelligent Vehicles Symposium (IV)* (2014) pp. 108–114.

[249] B. Schölkopf, A. Smola, and K.-R. Müller, *Kernel principal component analysis,* in *International Conference on Artificial Neural Wetworks* (1997) pp. 583–588.

[250] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, *Estimating the support of a high-dimensional distribution,* Neural Computation **13**, 1443 (2001).

[251] M. Scholtes, L. Westhofen, L. R. Turner, K. Lotto, M. Schuldes, H. Weber, N. Wagener, C. Neurohr, M. H. Bollmann, F. Körtke, J. Hiller, M. Hoss, J. Bock, and L. Eckstein, *6-layer model for a structured description and categorization of urban traffic and environment,* IEEE Access **9**, 59131 (2021).

[252] R. Schram, A. Williams, M. van Ratingen, S. Ryrberg, and R. Sferco, *Euro NCAP's first step to assess Autonomous Emergency Braking (AEB) for vulnerable road users,* in *24th Enhanced Safety of Vehicles (ESV) conference* (2015).

[253] 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 *Automotive Systems Engineering II* (Springer, 2018) pp. 147–175.

[254] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization* (John Wiley & Sons, 2015).

[255] D. W. Scott and S. R. Sain, *Multi-dimensional density estimation,* Handbook of Statistics **24**, 229 (2005).

[256] P. Seiniger, A. Hellmann, O. Bartels, M. Wisch, and J. Gail, *Test procedures and results for pedestrian AEB systems,* in *24th International Technical Conference on the Enhanced Safety of Vehicles (ESV)* (2015).

[257] S. Shah, D. Dey, C. Lovett, and A. Kapoor, *Airsim: High-fidelity visual and physical simulation for autonomous vehicles,* in *Field and Service Robotics* (2018) pp. 621–635.

[258] S. Shalev-Shwartz, S. Shammah, and A. Shashua, *On a formal model of safe and scalable self-driving cars,* arXiv preprint arXiv:1708.06374 (2017).

[259] Y. Shao, M. A. Mohd Zulkefli, Z. Sun, and P. Huang, *Evaluating connected and autonomous vehicles using a hardware-in-the-loop testbed and a living lab,* Transportation Research Part C: Emerging Technologies **102**, 121 (2019).

[260] X. Shi, Y. D. Wong, M. Z. F. Li, and C. Chai, *Key risk indicators for accident assessment conditioned on pre-crash vehicle trajectory,* Accident Analysis & Prevention **117**, 346 (2018).

[261] S. Siegel, *Nonparametric statistics,* The American Statistician **11**, 13 (1957).

[262] B. W. Silverman, *Density Estimation for Statistics and Data Analysis* (CRC press, 1986).

[263] W. V. Siricharoen, *Ontology modeling and object modeling in software engineering,* International Journal of Software Engineering and Its Applications **3**, 43 (2009).

[264] G. Smith, *Tagging: People-Powered Metadata for the Social Web* (New Riders Publishing, 2007).

[265] A. Snyder, *Encapsulation and inheritance in object-oriented programming languages,* in *Object-Oriented Programming Systems, Languages and Applications* (1986) pp. 38–45.

[266] M. Sommerfeld and A. Munk, *Inference for empirical Wasserstein distances on finite spaces,* Journal of the Royal Statistical Society Series B **80**, 219 (2018).

[267] P. Songchitruksa and A. P. Tarko, *The extreme value theory approach to safety estimation,* Accident Analysis & Prevention **38**, 811 (2006).

[268] Special Interest Group (SIG) on SImulation and Modeling (SIM), *Modeling and simulation glossary,* (2021), accessed July 2022.

[269] J. Spooner, V. Palade, M. Cheah, S. Kanarachos, and A. Daneshkhah, *Generation of pedestrian crossing scenarios using ped-cross generative adversarial network,* Applied Sciences **11**, 471 (2021).

[270] J. E. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, F. Niewels, and J. M. Zöllner, *Testing of advanced driver assistance towards automated driving: A survey and taxonomy on existing approaches and open questions,* in *IEEE 18th International Conference on Intelligent Transportation Systems* (2015) pp. 1455–1462.

[271] L. Stepien, S. Thal, R. Henze, H. Nakamura, J. Antona-Makoshi, N. Uchida, and P. Raksincharoensak, *Applying heuristics to generate test cases for automated driving safety evaluation,* Applied Sciences **11**, 10166 (2021).

[272] J. Sun, H. Zhou, H. Xi, H. Zhang, and Y. Tian, *Adaptive design of experiments for safety evaluation of automated vehicles,* IEEE Transactions on Intelligent Transportation Systems (2021), 10.1109/TITS.2021.3130040.

[273] A. P. Tarko, *Use of crash surrogates and exceedance statistics to estimate road safety,* Accident Analysis & Prevention **45**, 230 (2012).

[274] A. P. Tarko, *Estimating the expected number of crashes with traffic conflicts and the Lomax distribution — a theoretical and numerical exploration,* Accident Analysis & Prevention **113**, 63 (2018).

[275] A. P. Tarko, *Surrogate measures of safety,* in *Safe Mobility: Challenges, Methodology and Solutions* (Emerald Publishing Limited, 2018).

[276] S. Teuchert, *ISO 26262 — blessing or curse?* ATZelektronik worldwide **7**, 4 (2012).

[277] S. Thal, H. Znamiec, R. Henze, H. Nakamura, H. Imanaga, J. Antona-Makoshi, N. Uchida, and S. Taniguchi, *Incorporating safety relevance and realistic parameter combinations in test-case generation for automated driving safety assessment,* in *IEEE Intelligent Transportation Systems Conference (ITSC)* (2020) pp. 666–671.

[278] E. Thorn, S. Kimmel, and M. Chaka, *A Framework for Automated Driving System Testable Cases and Scenarios*, Tech. Rep. DOT HS 812623 (National Highway Traffic Safety Administration, 2018).

[279] D. Tokody, I. J. Mezei, and G. Schuster, *An overview of autonomous intelligent vehicle systems,* in *Vehicle and Automotive Engineering*, edited by K. Jármai and B. Bolló (Springer, 2017) pp. 287–307.

[280] Transport Systems Catapult, *Taxonomy of Scenarios for Automated Driving*, Tech. Rep. (Transport Systems Catapult, 2017).

[281] M. Treiber, A. Hennecke, and D. Helbing, *Congested traffic states in empirical observations and microscopic simulations,* Physical review E **62**, 1805 (2000).

[282] W. T. Tsai, A. Saimi, L. Yu, and R. Paul, *Scenario-based object-oriented testing framework,* in *Third International Conference on Quality Software* (2003) pp. 410–417.

[283] B. A. Turlach, *Bandwidth Selection in Kernel Density Estimation: A Review*, Tech. Rep. (Institut für Statistik und Ökonometrie, Humboldt-Universität zu Berlin, 1993).

[284] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer, *Defining and substantiating the terms scene, situation, and scenario for automated driving,* in *IEEE 18th International Conference on Intelligent Transportation Systems* (2015) pp. 982–988.

[285] N. Uno, Y. Iida, S. Yasuhara, and M. Suganuma, *Objective analysis of traffic conflict and modeling of vehicular speed adjustment at weaving section,* Infrastructure Planning Review **20**, 989 (2003).

[286] M. Utting, A. Pretschner, and B. Legeard, *A taxonomy of model-based testing approaches,* Software Testing, Verification and Reliability **22**, 297 (2012).

[287] J. van de Sluis, O. Op den Camp, J. Broos, I. Yalcinkaya, and E. de Gelder, *Describing I2V communication in scenarios for simulation-based safety assessment of truck platooning,* Electronics **10**, 1 (2021).

[288] P. W. F. van Notten, J. Rotmans, M. B. A. Van Asselt, and D. S. Rothman, *An updated scenario typology,* Futures **35**, 423 (2003).

[289] K. Vogel, *A comparison of headway and time to collision as safety indicators,* Accident Analysis & Prevention **35**, 427 (2003).

[290] M. D. Vose, *A linear algorithm for generating random numbers with a given distribution,* IEEE Transactions on Software Engineering **17**, 972 (1991).

[291] W. Wachenfeld and H. Winner, *The release of autonomous vehicles,* in *Autonomous Driving* (Springer, 2016) pp. 425–449.

[292] S. Wagner, K. Groh, T. Kuhbeck, M. Dorfel, and A. Knoll, *Using time-to-react based on naturalistic traffic object behavior for scenario-based risk assessment of automated driving,* in *IEEE Intelligent Vehicles Symposium (IV)* (2018) pp. 1521–1528.

[293] C.-G. Wallman and H. Åström, *Friction Measurement Methods and the Correlation between Road Friction and Traffic Safety: A Literature Review*, Tech. Rep. 911A (Swedish National Road and Transport Research Institute (VTI), 2001).

[294] M. P. Wand and M. C. Jones, *Multivariate plug-in bandwidth selection,* Computational Statistics **9**, 97 (1994).

[295] C. Wang and N. Stamatiadis, *Evaluation of a simulation-based surrogate safety metric,* Accident Analysis & Prevention **71**, 82 (2014).

[296] C. Wang, Y. Xie, H. Huang, and P. Liu, *A review of surrogate safety measures and their applications in connected and automated vehicles safety modeling,* Accident Analysis & Prevention **157**, 106157 (2021).

[297] T. Wang, M. Z. A. Bhuiyan, G. Wang, L. Qi, J. Wu, and T. Hayajneh, *Preserving balance between privacy and data integrity in edge-assisted internet of things,* IEEE Internet of Things Journal **7**, 2679 (2019).

[298] W. Wang, C. Liu, and D. Zhao, *How much data are enough? A statistical approach with case study on longitudinal driving behavior,* IEEE Transactions on Intelligent Vehicles **2**, 85 (2017).

[299] F. Warg, M. Skoglund, A. Thorsén, R. Johansson, M. Brännström, M. Gyllenhammar, and M. Sanfridson, *The quantitative risk norm — a proposed tailoring of HARA for ADS,* in *50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)* (2020) pp. 86–93.

[300] L. Wasserman, *All of Nonparamatric Statistics*, edited by G. Casella, S. Fienberg, and I. Olkin (Springer, 2006).

[301] Waymo, *Waymo Safety Report*, Tech. Rep. (Waymo, 2021).

[302] H. Weber, J. Bock, J. Klimke, C. Roesener, J. Hiller, R. Krajewski, A. Zlocki, and L. Eckstein, *A framework for definition of logical scenarios for safety assurance of automated driving,* Traffic Injury Prevention **20**, S65 (2019).

[303] P. Wegner, *Concepts and paradigms of object-oriented programming,* ACM SIGPLAN OOPS Messenger **1**, 7 (1990).

[304] A. Williamson, D. A. Lombardi, S. Folkard, J. Stutts, T. K. Courtney, and J. L. Connor, *The link between fatigue and safety,* Accident Analysis & Prevention **43**, 498 (2011).

[305] E. B. Wilson, *Probable inference, the law of succession, and statistical inference,* Journal of the American Statistical Association **22**, 209 (1927).

[306] P. Wimmer, M. Düring, H. Chajmowicz, F. Granum, J. King, H. Kolk, O. Op den Camp, P. Scognamiglio, and M. Wagner, *Toward harmonizing prospective effectiveness assessment for road safety: Comparing tools in standard test case simulations,* Traffic Injury Prevention **20**, S139 (2019).

[307] D. Wittmann, M. Lienkamp, and C. Wang, *Method for comprehensive and adaptive risk analysis for the development of automated driving,* in *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)* (2017) pp. 1–7.

[308] M. A. Wolters and W. J. Braun, *A practical implementation of weighted kernel density estimation for handling shape constraints,* Stat **7**, e202 (2018).

[309] Z. Wu, S. Shen, X. Lian, X. Su, and E. Chen, *A dummy-based user privacy protection approach for text information retrieval,* Knowledge-Based Systems **195**, 105679 (2020).

[310] Z. Wu, R. Wang, Q. Li, X. Lian, G. Xu, E. Chen, and X. Liu, *A location privacy-preserving system based on query range cover-up or location-based services,* IEEE Transactions on Vehicular Technology **69**, 5244 (2020).

[311] Z. Wu, G. Li, S. Shen, X. Lian, E. Chen, and G. Xu, *Constructing dummy query sequences to protect location privacy and query privacy in location-based services,* World Wide Web **24**, 25 (2021).

[312] L. Xiao, M. Wang, and B. van Arem, *Realistic car-following models for microscopic simulation of adaptive and cooperative adaptive cruise control vehicles,* Transportation Research Record **2623**, 1 (2017).

[313] J. Xie, A. R. Hilal, and D. Kulić, *Driving maneuver classification: A comparison of feature extraction methods,* IEEE Sensors Journal **18**, 4777 (2018).

[314] Z. Xiong and J. Olstam, *Orchestration of driving simulator scenarios based on dynamic actor preparation and automated action planning,* Transportation Research Part C: Emerging Technologies **56**, 120 (2015).

[315] Y. Xu, Y. Zou, and J. Sun, *Accelerated testing for automated vehicles safety evaluation in cut-in scenarios based on importance sampling, genetic algorithm and simulation applications,* Journal of Intelligent and Connected Vehicles **1**, 28 (2018).

[316] H. Yang, B. F. Van Dongen, A. H. M. Ter Hofstede, M. T. Wynn, and J. Wang, *Estimating Completeness of Event Logs*, Tech. Rep. (BPM center, 2012).

[317] K. H. Yang, J. Hu, N. A. White, A. I. King, C. C. Chou, and P. Prasad, *Development of numerical models for injury biomechanics research: A review of 50 years of publications in the Stapp Car Crash Conference,* Stapp Car Crash Journal **50**, 429 (2006).

[318] S. Yoshida, T. Hasegawa, S. Tominaga, and T. Nishimoto, *Development of injury prediction models for advanced automatic collision motification based on Japanese accident data,* International Journal of Crashworthiness **21**, 112 (2016).

[319] A. Z. Zambom and R. Dias, *A review of kernel density estimation with applications to econometrics,* International Econometric Review (IER) **5**, 20 (2013).

[320] F. Zhang, *The Schur Complement and Its Applications*, Vol. 4 (Springer Science & Business Media, 2006).

[321] P. Zhang, *Nonparametric importance sampling,* Journal of the American Statistical Association **91**, 1245 (1996).

[322] D. Zhao, H. Lam, H. Peng, S. Bao, D. J. LeBlanc, K. Nobukawa, and C. S. Pan, *Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques,* IEEE Transactions on Intelligent Transportation Systems **18**, 595 (2016).

[323] D. Zhao, X. Huang, H. Peng, H. Lam, and D. J. LeBlanc, *Accelerated evaluation of automated vehicles in car-following maneuvers,* IEEE Transactions on Intelligent Transportation Systems **19**, 733 (2018).

[324] L. Zheng, T. Sayed, and F. Mannering, *Modeling traffic conflicts for use in road safety analysis: A review of analytic methods and future directions,* Analytic Methods in Accident Research **29**, 100142 (2020).

[325] J. Zhou and L. del Re, *Safety verification of ADAS by collision-free boundary searching of a parameterized catalog,* in *Annual American Control Conference (ACC)* (2018) pp. 4790–4795.

[326] J. Zhou, Z. Cao, X. Dong, and A. V. Vasilakos, *Security and privacy for cloud-based IoT: Challenges,* IEEE Communications Magazine **55**, 26 (2017).

[327] M. R. Zofka, F. Kuhnt, R. Kohlhaas, C. Rist, T. Schamm, and J. M. Zöllner, *Data-driven simulation and parametrization of traffic scenarios for the development of advanced driver assistance systems,* in *18th International Conference on Information Fusion* (2015) pp. 1422–1428.

[328] M. R. Zofka, S. Klemm, F. Kuhnt, T. Schamm, and J. M. Zöllner, *Testing and validating high level components for automated driving: Simulation framework for traffic scenarios,* in *IEEE Intelligent Vehicles Symposium (IV)* (2016) pp. 144–150.

List of Publications

Journal articles

7. **E. de Gelder** and O. Op den Camp, *A quantitative method to determine what collisions are reasonably foreseeable and preventable,* Under review (2022).

6. **E. de Gelder**, K. Adjenughwure, J. Manders, R. Snijders, J.-P. Paardekooper, O. Op den Camp, A. Tejada, and B. De Schutter, *PRISMA: A novel approach for deriving probabilistic surrogate safety measures for risk evaluation,* Under review (2022).

5. **E. de Gelder**, J. Hof, E. Cator, J.-P. Paardekooper, O. Op den Camp, J. Ploeg, and B. De Schutter, *Scenario parameter generation method and scenario representativeness metric for scenario-based assessment of automated vehicles,* IEEE Transactions on Intelligent Transportation Systems (Early access).

4. **E. de Gelder**, J.-P. Paardekooper, A. Khabbaz Saberi, H. Elrofai, O. Op den Camp, S. Kraines, J. Ploeg, and B. De Schutter, *Towards an ontology for scenario definition for the assessment of automated vehicles: An object-oriented framework,* IEEE Transactions on Intelligent Vehicles **7**, 300 (2022).

3. **E. de Gelder**, H. Elrofai, A. Khabbaz Saberi, O. Op den Camp, J.-P. Paardekooper, and B. De Schutter, *Risk quantification for automated driving systems in real-world driving scenarios,* IEEE Access **9**, 168953 (2021).

2. J. van de Sluis, O. Op den Camp, J. Broos, I. Yalcinkaya, and **E. de Gelder**, *Describing I2V communication in scenarios for simulation-based safety assessment of truck platooning,* Electronics **10**, 1 (2021).

1. **E. de Gelder**, J.-P. Paardekooper, O. Op den Camp, and B. De Schutter, *Safety assessment of automated vehicles: How to determine whether we have collected enough field data?* Traffic Injury Prevention **20**, 162 (2019).

Conference papers

11. **E. de Gelder** and O. Op den Camp, *A quantitative method to determine what collisions are reasonably foreseeable and preventable,* in *8th Road Safety & Simulation International Conference* (2022).

10. **E. de Gelder**, E. Cator, J.-P. Paardekooper, O. Op den Camp, and B. De Schutter, *Constrained sampling from a kernel density estimator to generate scenarios for the assessment of automated vehicles,* in *IEEE Intelligent Vehicles Symposium Workshops (IV Workshop)* (2021) pp. 203–208.

9. O. Op den Camp, J. van de Sluis, **E. de Gelder**, and I. Yalcinkaya, *Generation of tests for safety assessment of V2V platooning trucks,* in *27th ITS World Congress*, 70 (2021).

8. **E. de Gelder** and O. Op den Camp, *Tagging real-world scenarios for the assessment of autonomous vehicles,* in *38th FISITA World Congress*, F2020-PIF-048 (2020).

7. **E. de Gelder** and O. Op den Camp, *Procedure for the safety assessment of an autonomous vehicle using real-world scenarios,* in *38th FISITA World Congress*, F2020-VES-014 (2020).

6. **E. de Gelder**, J. Manders, C. Grappiolo, J.-P. Paardekooper, O. Op den Camp, and B. De Schutter, *Real-world scenario mining for the assessment of automated vehicles,* in *IEEE International Transportation Systems Conference (ITSC)* (2020) pp. 1073–1080.

5. **E. de Gelder**, A. Khabbaz Saberi, and H. Elrofai, *A method for scenario risk quantification for automated driving systems,* in *26th International Technical Conference on the Enhanced Safety of Vehicles (ESV)* (2019).

4. J.-P. Paardekooper, S. Montfort, J. Manders, J. Goos, **E. de Gelder**, O. Op den Camp, A. Bracquemond, and G. Thiolon, *Automatic identification of critical scenarios in a public dataset of 6000 km of public-road driving,* in *26th International Technical Conference on the Enhanced Safety of Vehicles (ESV)* (2019).

3. F. Remmen, I. Cara, **E. de Gelder**, and D. Willemsen, *Cut-in scenario prediction for automated vehicles,* in *IEEE International Conference on Vehicular Electronics and Safety (ICVES)* (2018).

2. J. Ploeg, **E. de Gelder**, M. Slavík, E. Querner, T. Webster, and N. de Boer, *Scenario-based safety assessment framework for automated vehicles,* in *16th ITS Asia-Pacific Forum* (2018) pp. 713-726.

1. **E. de Gelder** and J.-P. Paardekooper, *Assessment of automated driving systems using real-life scenarios,* in *IEEE Intelligent Vehicles Symposium (IV)* (2017) pp. 589–594.

Other publications

2. **E. de Gelder**, O. Op den Camp, and N. de Boer, *Scenario Categories for the Assessment of Automated Vehicles,* Tech. Rep. (CETRAN, 2020) version 1.7.

1. H. Elrofai, J.-P. Paardekooper, **E. de Gelder**, S. Kalisvaart, and O. Op den Camp, *Scenario-Based Safety Validation of Connected and Automated Driving,* Tech. Rep. (Netherlands Organization for Applied Scientific Research, TNO, 2018).

Curriculum Vitæ

Erwin de Gelder was born on October 10, 1989, in Zwijndrecht, The Netherlands. He finished his pre-university education in 2008 at the DevelsteinCollege in Zwijndrecht. After finishing his bachelor of science in mechanical engineering (cum laude) at Delft University of Technology in Delft, The Netherlands, in 2011, he started with a master of science in Systems & Control at Delft University of Technology, which he completed (cum laude) in 2014.

Since 2014, he has been working for the Integrated Vehicles Safety department of the Netherlands Organization for Applied Scientific Research (TNO) in Helmond, The Netherlands. His current research focuses on a quantitative assessment methodology for automated vehicles using scenarios obtained from real-world data. From 2017 to 2019, Erwin worked at the Nanyang Technological University, Singapore, to apply his research for an assessment procedure for automated vehicles in Singapore. In 2017, he started pursuing his PhD degree on this topic at the Delft University of Technology, of which the results are presented in this dissertation.

Dankwoord

Nog voordat ik afstudeerde voor mijn Master of Science Systems and Control aan de Technische Universiteit Delft, besloot ik te gaan werken bij TNO, de Nederlandse organisatie voor Toegepast Natuurwetenschappelijk Onderzoek. Het bleek een geweldige beslissing te zijn, aangezien ik het heel leuk vond (en vind) om zowel onderzoek te doen als het onderzoek toe te passen om zo het leven van de mensheid iets te verbeteren; al is het maar een klein beetje. Al snel kwam ik erachter dat het onderzoek doen naar de veiligheid van (deels) zelfrijdende auto's mijn interesse had. Omdat mij dit zo aansprak, ben ik in oktober 2017 gestart met mijn promotieonderzoek naar de veiligheidsanalyse van geautomatiseerde voertuigen. Net als mijn beslissing om voor TNO te gaan werken, was dit voor mij de juiste keuze, omdat ik heb genoten van mijn tijd als PhD-kandidaat. TNO heeft mij ook de kans gegeven om de eerste twee jaar van mijn promotieonderzoek in Singapore uit te voeren. Ik zal deze ervaring altijd koesteren. Natuurlijk zouden mijn jaren als PhD-kandidaat niet zo leuk noch zo succesvol zijn geweest zonder de hulp van zoveel anderen. Ik wil graag alle mensen die mij hebben geholpen heel hartelijk bedanken.

Bart De Schutter, toen ik je voor het eerst ontmoette om mijn ideeën voor mijn promotieonderzoek te presenteren en om te vragen of je mijn promotor wilde zijn, was ik voorbereid op een discussie waarom je jouw kostbare tijd en moeite zou moeten steken in het begeleiden van mij. Deze discussie heeft nooit plaatsgevonden. In plaats daarvan begon je een discussie over de uitdagingen met betrekking tot mijn onderzoek. Dat was voor mij het bewijs dat je de perfecte promotor zou zijn. En, je hebt me nooit teleurgesteld. Wel moet ik toegeven dat ik een beetje schrok toen ik voor het eerst feedback van je ontving op mijn werk, deels vanwege de hoeveelheid feedback en deels vanwege het handschrift. Maar na al deze jaren raakte ik eraan gewend (ook het handschrift kan ik nu lezen). Bedankt voor al je inspanningen! Ik ben ervan overtuigd dat ik zonder jouw hulp en feedback niet de onderzoeker zou zijn die ik nu ben.

Jan-Pieter Paardekooper, ik hoefde niet lang na te denken wie ik zou vragen om mijn copromotor te zijn. Vanaf het moment dat je bij TNO kwam (net na mij), viel het me op dat je altijd een andere invalshoek had op zaken dan ik. Dat heeft me altijd geprikkeld. Jouw originele en authentieke kijk — en niet alleen op het onderzoek — blijft mij inspireren. Niet alleen heb je een belangrijke bijdrage geleverd aan de totstandkoming van dit proefschrift, ook buiten het werk om bleek je een uitstekende gesprekspartner (en wandelpartner) te zijn. Ontzettend bedankt voor al je hulp! Ik hoop dat we ook in de toekomst kunnen blijven samenwerken en nog vele mooie wandelingen kunnen maken.

Olaf Op den Camp, ook jij hebt een belangrijke bijdrage geleverd aan dit proefschrift. Of je nu druk was of niet, ik kon altijd rekenen op jouw feedback. Maar

nog meer dan jouw hulp bij mijn proefschrift, zal ik onze samenwerking blijven koesteren. Eerst in Singapore, waar je mij gedurende de twee jaar dat ik daar zat vaak hebt opgezocht om een paar weken samen te werken. Een goede herinnering aan deze tijd zijn de uitgebreide gesprekken tijdens onze koffiemomenten, lunches en diners. Je hebt me niet alleen geholpen bij het inhoudelijke werk, maar ook met het omgaan met politieke gevoeligheden en hoe ik bepaalde zaken het beste kon aanpakken. Nog een wijze les, die ik dankzij jou weet, is dat lang niet alles zwart-wit is. Hartstikke bedankt voor alles en ik hoop dat ik nog lang gebruik mag maken van jouw kennis en wijsheid.

Jeroen Ploeg, ik zal mijn eerste drie maanden van mijn promotie — wat ook mijn eerste maanden in Singapore waren — niet snel vergeten. Het was heerlijk om samen met jou te discussiëren over het Singaporese eten en lineaire algebra (dat liep naadloos in elkaar over). Elke zin of vergelijking die ik opschreef, werd vakkundig gecontroleerd door jou en dat heeft mij zeker geholpen. Ik ben dankbaar dat je, ook nadat je TNO hebt verlaten, mij wilde helpen bij het schrijven van twee artikelen die nu ook als hoofdstuk in dit boek zijn opgenomen.

Hala Elrofai, at the beginning of 2017, you were the first person that I told about my idea to start with a PhD. You immediately started arguing why on earth I would do such a thing and the troubles that I would face. But after this discussion, you realized that I already made my decision and from that moment on, you helped me with finding enough support within TNO to (partially) fund my PhD. I will be forever grateful for your help. Next to this, you also contributed to two articles that are now chapters of my dissertation.

Mijn eerste twee jaar van mijn onderzoek heb ik gedaan in Singapore. Peter van Hooft, jij was toen ook mijn nieuwe afdelingshoofd. Je hebt mij geholpen bij mijn uitdagingen wat betreft het combineren van mijn werk in Singapore en het doen van mijn promotieonderzoek. Bedankt daarvoor! Ronnie van Munster, mijn tijd in Singapore was zeker niet zo aangenaam geweest zonder jou. Jij hebt ervoor gezorgd dat ik me zo ver van huis snel thuis voelde. Wat uiteindelijk "één jaar Singapore" had moeten zijn, werd mede door jou "twee jaar Singapore".

Natuurlijk zijn er veel meer collega's van TNO die mij hebben geholpen en die ik wil bedanken. Bastiaan Krosse en Maurice Kwakkernaat, toen ik weer terug was in Nederland hebben jullie mijn promotieonderzoek als afdelingshoofden altijd ondersteund. Door jullie interesse in mijn onderzoek, hebben jullie mij de waardering gegeven die mij motiveerde om door te gaan met het onderzoek. Emilia Silvas, thank you for your sincere interest in my research and your creativity in finding financial resources for my research. Henk Goossens, jij bent dikwijls enthousiaster over mijn onderzoek dan ikzelf. Mede daardoor ben je uitstekend geschikt om ervoor te zorgen dat mijn onderzoek niet verdwijnt in een laatje, maar dat het onderzoek ook daadwerkelijk zijn toepassing vindt. Mijn collega's van het TNO StreetWise team, Jeroen Broos, Sytze Kalisvaart, Jeroen Uittenbogaard en Geert Verhaeg helpen ook bij het toepassen van mijn onderzoek, dus ik ben ook jullie dank verschuldigd.

Veel (oud-) collega's hebben mij geholpen met de publicaties in dit proefschrift. Arash Saberi, it was nice to have you as a TNO colleague and as a fellow PhD candidate. It was great that I could make use of your expertise even after you left

TNO. Jeroen Manders, jouw skills om data te analyseren en te prepareren hebben mij enorm geholpen. Corrado Grappiolo, your implementation of mining scenarios from data was the perfect starting point for our publication on the same topic. Kingsley Adjenughwure, I really benefited from your knowledge about surrogate safety measures. Ron Snijders, dankzij jou kon ik gelijk aan de slag met de data. Arturo Tejada, by questioning almost everything — even the things that I thought to be obvious — you improved the clarity of our work substantially.

Tijdens mijn promotie heb ik Stefano Celin, Jasper Hof en Rens Smeets mogen begeleiden tijdens hun afstuderen. Stefano, it was great to have you as a student. You did a great job to continue the work regarding data completeness. Your enthusiasm was really inspiring. Jasper, ik heb veel geluk gehad met jou als student. Jouw afstudeerwerk is de basis geweest voor een hoofdstuk van dit proefschrift en je hebt als co-auteur ook bijgedragen aan de publicatie van een artikel. Rens, ook jij hebt geweldig werk afgeleverd en een mooie basis neergelegd voor toekomstig onderzoek.

Hierboven heb ik al vele co-auteurs genoemd, maar nog niet allemaal. Eric Cator, het was fijn om feedback te krijgen van iemand met een sterke wiskundige achtergrond. Uiteindelijk heeft dat geleid tot twee hoofdstukken in dit proefschrift, dus bedankt daarvoor. Steven Kraines, it was great that you could teach me all about object-oriented modeling and ontologies. If I would work on anything related to ontologies, you would be the first person I would go to. Thank you for your contribution.

De geleverde arbeid die tot dit proefschrift heeft geleid, had niet mogelijk geweest zonder uitlaatklep en dat is tafeltennisvereniging Stiphout voor mij geweest. Niet alleen omdat ik hier vaak heb getafeltennist, maar ook door de fijne tijd in het bestuur van de vereniging en de fantastische bestuursgenoten. Ik durf te stellen dat de harmonie waarin binnen het bestuur wordt samengewerkt vrij zeldzaam is. En hoewel het zeker niet ideaal is als een bestuurslid in Singapore verblijft, heb ik me door jullie hierin altijd gesteund gevoeld. Via deze weg wil ik de vereniging — en in het bijzonder het bestuur — dan ook bedanken voor de samenwerking, de sportiviteit en de gezelligheid.

Het is ontzettend waardevol als je altijd terug kan vallen op je vrienden en ook zij verdienen het om hier genoemd te worden. In het bijzonder wil ik Chris, Lars en Marco bedanken. We kennen elkaar al zo lang en wat er ook gebeurt, ik weet dat jullie er altijd voor me zijn. En ondanks dat we tegenwoordig iets minder fanatiek zijn, is het nog steeds heerlijk om samen naar Ajax te kijken. Marco, jou wil ik nog speciaal bedanken omdat je mijn paranimf wilt zijn. Omdat we ook tot ver in onze studententijd samen zijn opgetrokken, was het voor mij een makkelijke keuze om jou hiervoor te vragen.

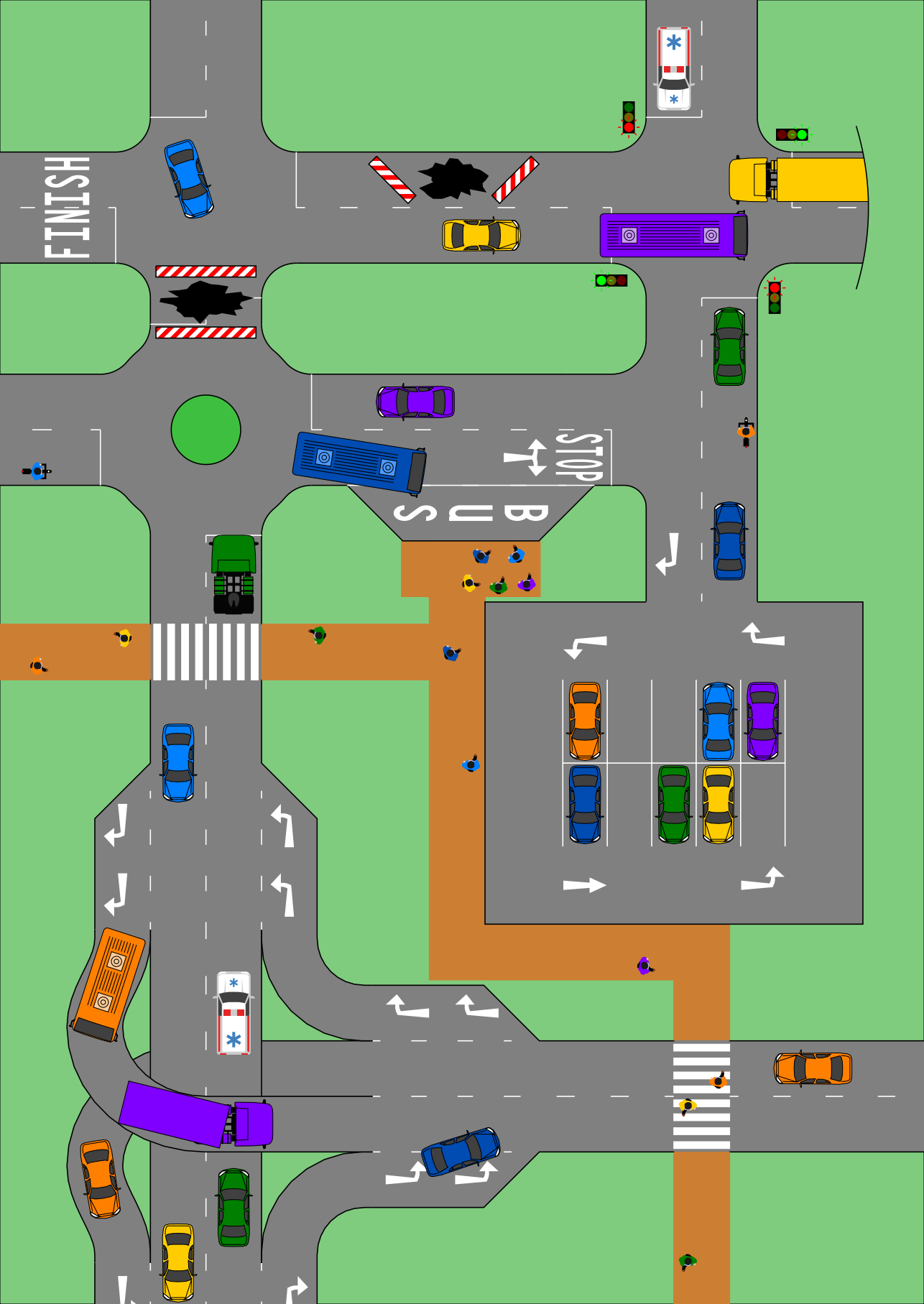
Jan, Jan-Joost & Marieke en Esther & Freek, wat leuk dat ik jullie er als familie bij heb gekregen. Ik heb me vanaf het eerste moment bij jullie thuis gevoeld en ik ben blij dat jullie mijn bijdehandheid tolereren. Het is jammer dat Aartje er niet meer bij kan zijn. Het is bijzonder hoe betrokken ze was en ik ben dankbaar voor het feit dat ik haar nog heb leren kennen.

Natuurlijk wil ik ook nog mijn familie benoemen. Jeannette & Herman, Yolanda &

Steve, Remco & Wieteke, Vincent & Emma en mijn moeder Adrie, heel erg bedankt dat ik altijd op jullie kan terugvallen en bedankt voor jullie interesse. Ma, jij hebt ongelooflijk veel moeten aanhoren van mijn wel en wee. Bedankt voor je geduld en je luisterend oor. Ik weet dat je trots bent net zoals pa dat ongetwijfeld ook was geweest. Vincent, als tweelingbroer weet jij als geen ander hoe ik me voel. Of we nu samen koffiedrinken, eten, klussen of sporten, onze tijd samen is en blijft waardevol. Het was daarom logisch om jou ook te vragen als paranimf en een dankjewel is hiervoor zeker op zijn plaats.

Als laatste wil ik mijn lieve Judith bedanken. Het moet niet altijd makkelijk zijn geweest als ik weer even achter mijn laptop kroop, maar jij hebt mij hierin altijd gesteund. Ik kan je niet genoeg bedanken voor jouw liefde en voor alles wat je hebt gedaan. Ik kijk uit naar onze toekomst samen en ik hoop dat er nog vele jaren volgen waarin we samen het leven kunnen delen.

Erwin de Gelder
Dordrecht, augustus 2022


FINISH
STOP
BUS