

Technische Universität München

TUM School of Engineering and Design

**Safeguarding Complex and Learning-Based
Automated Driving Functions via Online Verification**

Tim Nikolaus Stahl, M.Sc.

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

Doktors der Ingenieurwissenschaften

genehmigten Dissertation.

Vorsitzender:

Prof. Dr.-Ing. Boris Lohmann

Prüfer der Dissertation:

1. Prof. Dr.-Ing. Markus Lienkamp

2. Prof. Dr. rer. nat. Hermann Winner

Die Dissertation wurde am 17.11.2021 bei der Technischen Universität München eingereicht
und durch die TUM School of Engineering and Design am 09.02.2022 angenommen.

*“Safety doesn’t happen
by accident.”*

Author Unknown

Acknowledgments

This dissertation was created during my work as a research assistant at the *Institute of Automotive Technology* at the *Technical University of Munich* in cooperation with the department for *Highly Automated Driving* at *TÜV SÜD Auto Service GmbH* from 2018 to 2021.

First and foremost, I would like to thank my supervisor Prof. Dr.-Ing. Markus Lienkamp, who provided excellent supervision on the basis of trust and the encouragement for a free and self-determined working environment. I would also like to thank my second examiner Prof. Dr. rer. nat. Hermann Winner for taking over the second assessment and for his valuable advice. Furthermore, I would like to thank Prof. Dr.-Ing. Boris Lohmann for taking over the chairmanship of the examination.

I would like to acknowledge *TÜV SÜD Auto Service GmbH* for financing the research project. Special thanks goes to Christian Gndt, Christian Pahlke, Matthis Eicher, Patrick Scharpfenecker, Dr. rer. nat. Simon Rößner, Dieter Ludwig, Alexander Schwab, Benjamin Koller, and Dr.-Ing. Housseem Abdellatif. I would like to thank all of you for your continuous support and enriching discussions. I would also like to express my appreciation to the company *Roborace*. Starting with the provision of the automated race vehicles, the logistics, rental of the race tracks, up to the track engineering support and much more, a productive development and evaluation was possible. Without this support, this work would not have been possible on this scale.

I would also like to thank Dr.-Ing. Frank Diermeyer, who, as group leader at the institute, supported the development of the thesis and provided technical assistance. Moreover, I extend my gratitude to my proofreaders Dr.-Ing. Johannes Betz, Dr.-Ing. Thomas Ponn, Stefan Riedmaier, and Prof. Dr.-Ing. Holger Stahl.

I would like to thank all colleagues and friends at the *Institute of Automotive Technology* for the wonderful time we spent together. I will always have the best memories about the professional discussions, the intensive doctoral colloquia and the many little moments that enriched my everyday work. In this context, my special regards go to the PhD students of the *TUM Autonomous Motorsport* team for the unforgettable time during development sprints and test sessions at race tracks all over Europe. A big thank you goes to all the students I supervised, who supported the research project through commitment and fruitful discussions.

My family, especially my grandparents, parents, brothers and sisters as well as my girlfriend Magdalena deserve my deepest gratitude. You stood by me during difficult times of the PhD project and encouraged me in all my plans. I thank you for your continued support.

Garching, November 2021

Tim Stahl

Contents

List of Abbreviations	V
Formula Symbols	VII
1 Introduction and Scope	1
2 Related Work	5
2.1 Terms and Definitions	5
2.2 Automated Driving Software Architecture	8
2.3 Motion Planning	9
2.3.1 Classic Approaches	10
2.3.2 Machine-Learning-Based Approaches	12
2.4 Automotive Safety	13
2.4.1 Regulations and Approval Procedures	13
2.4.2 Safety Assessment Techniques	16
2.5 Online Monitoring	19
2.5.1 Online Risk Assessment	20
2.5.2 Online Verification	22
2.5.3 Overall Valuation and Classification	24
3 Problem Description and Methodology	27
3.1 Problem Description	27
3.2 Methodology and Outline of the Thesis	28
4 Generic Method for the Development of a Safeguarding Framework	31
4.1 Requirements	32
4.1.1 Criteria for Safe Operation	32
4.1.2 Requirements to Be Met by the Supervisor	32
4.2 Implementation	34
4.2.1 Supervisor Architecture	34
4.2.2 Assessment Metrics for Identified Criteria	34
4.2.3 Classification and Selection	35
4.3 Integration and Test	35

4.3.1	Software Stack Integration	35
4.3.2	Test of Internal and External Interfaces	35
4.3.3	Tests of the Requirements	36
4.4	Validation	36
4.4.1	Scenario-Based Evaluation	36
4.4.2	Random-Based Evaluation	36
5	Online Verification of a Motion Planning Module	39
5.1	Criteria for Safe Trajectories	40
5.2	Online Verification Architecture	42
5.3	Assessment Metrics for Identified Criteria – Preliminaries	43
5.3.1	General Notation	44
5.3.2	Reachable Sets	44
5.3.3	Past Time Linear Temporal Logic	45
5.3.4	Preparation and Analysis of the Criteria for Safe Operation	46
5.4	Assessment Metrics for Identified Criteria – Static Environment	47
5.4.1	Collision with Static Obstacles	47
5.4.2	Safe End State	49
5.4.3	Friction and Tire Forces	49
5.4.4	Rules and Regulations	52
5.4.5	Kinematic and Dynamic Properties	52
5.4.6	Data Correctness	53
5.5	Assessment Metrics for Identified Criteria – Dynamic Environment	54
5.5.1	Rule-Based Reachable Set	55
5.5.2	Guaranteed Occupied Set	58
5.6	Classification and Selection	60
6	Results	63
6.1	Software Stack Integration	63
6.2	Integration Tests	64
6.2.1	Test of Internal and External Interfaces	64
6.2.2	Test of the Requirements	65
6.3	Scenario-Based Evaluation	66
6.3.1	Scenario Architect	67
6.3.2	Evaluation Results	72
6.4	Random-Based Evaluation	79

6.4.1 Testbed DevBot by Roborace	80
6.4.2 Test Grounds	80
6.4.3 Evaluation Results	81
7 Discussion	89
7.1 Review of Posed Safety Standard Limitations	89
7.1.1 Online Adaptation	89
7.1.2 Complex and Non-transparent	89
7.2 Review of Posed Guiding Theses and Research Questions	90
7.2.1 Evaluation of Guiding Thesis 1	90
7.2.2 Evaluation of Guiding Thesis 2	91
7.2.3 Evaluation of Guiding Thesis 3	91
7.2.4 Evaluation of the Research Gap	92
7.3 Findings and Future Work	92
7.3.1 Influence of Emergency Trajectory Planning Performance	93
7.3.2 Added Value During the Development Phase	94
7.3.3 Extension to Road Traffic	95
7.3.4 Extension to Other Software Modules in the Driving Stack	96
7.3.5 Safeguarding of the Overall Software	96
8 Summary and Conclusion	97
List of Figures	i
List of Tables	xiii
Bibliography	xv
Prior Publications	xxxvii
Supervised Student Theses	xxxix
Appendix	xli

List of Abbreviations

ADAS	Advanced Driver Assistance Systems
ADS	Automated Driving System
AI	Artificial Intelligence
ASIL	Automotive Safety Integrity Level
CV	Constant Velocity
DGPS	Differential Global Positioning System
DSS	Difference of Space Distance and Stopping Distance
E/E	Electrical and/or Electronic
FuSa	Functional Safety
HARA	Hazard Analysis and Risk Assessment
HiL	Hardware-in-the-Loop
HW	Hardware
LIDAR	Light Detection and Ranging
LTL	Linear Temporal Logic
ML	Machine Learning
MPC	Model Predictive Control
NN	Neural Network
ODD	Operational Design Domain
OEM	Original Equipment Manufacturer
OV	Online Verification
PDF	Probability Density Function
ptLTL	Past Time Linear Temporal Logic
RL	Reinforcement Learning
RSS	Responsibility-Sensitive Safety
SAE	Society of Automotive Engineers
SiL	Software-in-the-Loop
SOTIF	Safety of the Intended Functionality
SSM	Surrogate Safety Metric
STL	Signal Temporal Logic
SupMod	Supervisor Module
SW	Software
TP	Traffic Participant
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
WHO	World Health Organization

Formula Symbols

Formula Symbol	Unit	Description
A	m^2	Reference area
a_{comb}	m s^{-2}	Combined (lateral and longitudinal) acceleration
a_{max}	m s^{-2}	Maximum combined acceleration
a_x	m s^{-2}	Longitudinal acceleration
a_y	m s^{-2}	Lateral acceleration
$a_{x,\text{lim}}$	m s^{-2}	Maximum longitudinal acceleration generated by the motor
$a_{x,\text{max}}$	m s^{-2}	Maximum feasible longitudinal acceleration transferred by the tires
$a_{y,\text{max}}$	m s^{-2}	Maximum feasible lateral acceleration transferred by the tires
$a_{x,\text{tire}}$	m s^{-2}	Proportion of longitudinal acceleration transferred by the tires
$a_{y,\text{tire}}$	m s^{-2}	Proportion of lateral acceleration transferred by the tires
\mathcal{B}_l	-	Polyline describing the left track boundary
\mathcal{B}_r	-	Polyline describing the right track boundary
$\chi(\cdot)$	-	Function operator describing the solution of a motion equation
c_d	-	Drag coefficient
$c_{d,\text{ext}}$	m	Extended drag coefficient representing a vehicle-specific value
c_k	-	Triggering condition k for a rule-based set reduction
c_{RR}	-	Triggering condition for the <i>Roborace</i> overtaking regulation
c_{RA}	-	Triggering condition for the racing alongside regulation
c_{DO}	-	Triggering condition for the defending off-line regulation
$d_{\text{brake,ego}}$	m	Brake distance of ego-vehicle
$d_{\text{brake,le}}$	m	Brake distance of lead vehicle
d_{corner}	m	Distance relevant for defining the approach to an corner within the defending off-line regulation
d_{defend}	m	Distance relevant for defining the defend state of a vehicle within the defending off-line regulation
d_{infl}	m	Inflation diameter

d_{overlap}	m	Overlap distance for the racing alongside regulation
$d_{\text{react,ego}}$	m	Distance traveled during a reaction time
d_x	m	Distance in longitudinal direction measured from bumper-to-bumper
DSS	m	Difference of space distance and stopping distance
$\mathcal{E}(\cdot)$	-	Occupation operator for the provided state or set
F_a	N	Combined force acting on the tires of a vehicle
$F_{a,\text{max}}$	N	Maximum possible combined force that can act on the tires of a vehicle without losing grip
F_N	N	Normal force
F_x	N	Longitudinal force acting on the tires of a vehicle
F_y	N	Lateral force acting on the tires of a vehicle
$\mathcal{G}^\mathcal{E}$	-	Occupied states by guaranteed occupied set
$\mathcal{I}(\cdot)$	-	Inflation operator
k	-	Formalized rule within the rule-based set reduction
κ	m^{-1}	Curvature
κ_{thr}	m^{-1}	Threshold curvature for differentiation of straight and turn segments
\mathcal{K}	-	Set of formalized rules for the rule-based set reduction
\mathcal{K}_c	-	Set of triggered formalized rules for the rule-based set reduction
l_i	-	Hardware-related limitation to be obeyed by the ego-trajectory
\mathcal{L}	-	Set of hardware-related limitations to be obeyed by the ego-trajectory
\mathcal{M}	-	Environment map
μ	-	Friction coefficient
n	m	n -coordinate in the Frenet frame: measured lateral offset to the reference line (based on the normal vectors of the reference line)
\mathcal{O}	-	Object list
p	-	Position in the Cartesian coordinate frame
p_i	-	Plausibility check to be fulfilled by an ego-vehicle trajectory
\mathcal{P}	-	Set of plausibility checks to be fulfilled by an ego-vehicle trajectory
Ψ	-	Set of Supervisor Modules (SupMods)
Ψ_{em}	-	Set of Supervisor Modules (SupMods) relevant for the emergency trajectory
Ψ_{perf}	-	Set of Supervisor Modules (SupMods) relevant for the performance trajectory
$\psi_i(\cdot)$	-	Individual Supervisor Module (SupMod) i in the set Ψ generating a boolean safety rating s_i based on given inputs

$\psi_{\text{acc}}(\cdot)$	-	Supervisor Module (SupMod) generating boolean safety rating for a given trajectory w.r.t. acceleration limits
$\psi_{\text{dc}}(\cdot)$	-	Supervisor Module (SupMod) generating boolean safety rating for a given trajectory w.r.t. data correctness
$\psi_{\text{kd}}(\cdot)$	-	Supervisor Module (SupMod) generating boolean safety rating for a given trajectory w.r.t. kinematic and dynamic limits
$\psi_{\text{occ}}(\cdot)$	-	Supervisor Module (SupMod) generating boolean safety rating for a given trajectory w.r.t. the guaranteed occupied sets of other vehicles
$\psi_{\text{reach}}(\cdot)$	-	Supervisor Module (SupMod) generating boolean safety rating for a given trajectory w.r.t. the reachable sets of other vehicles
$\psi_{\text{rule}}(\cdot)$	-	Supervisor Module (SupMod) generating boolean safety rating for a given trajectory w.r.t. applicable rules
$\psi_{\text{ses}}(\cdot)$	-	Supervisor Module (SupMod) generating boolean safety rating for a given trajectory w.r.t. a safe end state
$\psi_{\text{stat}}(\cdot)$	-	Supervisor Module (SupMod) generating boolean safety rating for a given trajectory w.r.t. a collision with a static environment
P	-	Power set for the provided argument
ρ	kg m^{-3}	Mass density of the air
Q_k	-	Reduction set k for a rule-based set reduction
Q_{RR}	-	Reduction set for the <i>Roborace</i> overtaking regulation
Q_{RA}	-	Reduction set for the racing alongside regulation
Q_{DO}	-	Reduction set for the defending off-line regulation
\mathcal{R}	-	Reachable set (set of reachable states within the state space \mathcal{X})
\mathcal{R}_0	-	Initial set of states
\mathcal{R}^e	-	Exact reachable set
$\mathcal{R}^\mathcal{E}$	-	Occupied states by reachable set \mathcal{R}
$\mathcal{R}_{\mathcal{K},o}^\mathcal{E}$	-	Occupied states by rule-based reachable set
r_{turn}	m	Turn radius of the ego-vehicle
s	m	s -coordinate in the Frenet frame: measured arc length along a reference line of the track
s_{corner}	m	s -coordinate of a corner
s_i	-	Boolean safety rating for Supervisor Module (SupMod) i
s_{em}	-	Boolean safety rating for an emergency trajectory
s_{perf}	-	Boolean safety rating for a performance trajectory
\hat{s}_{stat}	-	Boolean ground truth for safety rating w.r.t. a static environment
\hat{s}_{dyn}	-	Boolean ground truth for safety rating w.r.t. a dynamic environment
t	s	Time

t_0	s	Initial time
t_h	s	Time horizon of a planned trajectory
t_f	s	Time horizon for determined reachable set
t_{react}	s	Reaction time
t_{trig}	s	Trigger time within the <i>Roborace Season Alpha</i> regulation
\mathcal{U}	-	Set of admissible control inputs
u	-	Control input of arbitrary dimension
u_1	-	Normalized steering
u_2	-	Normalized acceleration
v_x	-	Longitudinal velocity
v_{ego}	-	Velocity of ego-vehicle
v_{le}	-	Velocity of lead vehicle
w_{veh}	m	Vehicle width
x	m	x -coordinate in the Cartesian coordinate system
\mathcal{X}	-	State space as a subset of \mathbb{R}^n
ξ	-	State within the state space $\mathcal{X} \subset \mathbb{R}^n$
ξ_0	-	Initial state
ξ_{em}	-	State of the emergency trajectory
ξ_{perf}	-	State of the performance trajectory
ξ_o	-	State of object o
y	m	y -coordinate in the Cartesian coordinate system
z_i	-	Rule to be obeyed by the ego-vehicle
z_{vmax}	-	Rule restricting the maximum velocity of the ego-vehicle
z_{rev}	-	Rule prohibiting the ego-vehicle from reversing
z_{amin}	-	Rule restricting the maximum deceleration of the ego-vehicle
\mathcal{Z}	-	Set of applicable rules
\mathcal{Z}_1	-	Track segment specified as the track inside in front of a turn
\mathcal{Z}_O	-	Overtaking zone within the <i>Roborace Season Alpha</i> regulation
$\mathcal{Z}_{O,l}$	-	Left half of the overtaking zone within the <i>Roborace Season Alpha</i> regulation
$\mathcal{Z}_{O,r}$	-	Right half of the overtaking zone within the <i>Roborace Season Alpha</i> regulation
\mathcal{Z}_T	-	Trigger zone within the <i>Roborace Season Alpha</i> regulation

1 Introduction and Scope

Mobility is in constant flux in the current era. New drive concepts such as e-mobility as well as increasing automation are one of the central aspects of current development. With further electrification and automation, there is a need for safety criteria and mechanisms to grow along with it.

Every 24 seconds, on average, someone dies on the roads according to the latest road safety report [1, p. 4] published by the World Health Organization (WHO) in 2018. One of the key visions in the domain of automated driving is to obtain a flawless, accident-free traffic. In 1995, the Swedish government launched the so-called Vision Zero for the road transport sector with the aim of achieving zero road deaths and zero serious injuries [2]. This may sound utopian and the originally targeted deadline, the year 2015, already passed by. Nevertheless, it describes the philosophy that has been increasingly pursued since then.

Over the years, the introduction of passive safety mechanisms like seat-belts or improvements of the vehicle body frame led to a decline in fatal accidents [3, p. 600]. However, the human as a driver makes mistakes by nature. Although these do not necessarily lead to an accident, if they do, they should be prevented from doing so by appropriate intervention of a safety system. In the last couple of decades numerous Advanced Driver Assistance Systems (ADAS) have found their way into production vehicles. Several studies [4, pp. 7-20, 5, p. 364, 6, p. 8] have shown that ADAS help to reduce the number of traffic accidents further. A next step in this direction is full automation, with the goal of reducing the traffic mortality rate even more.

Industry and academics dedicate their research and development to the field of automated driving and the attained research interest is constantly growing. However, Automated Driving Systems (ADSs) are far from being capable to handle every situation flawlessly yet. Currently, there are only a few ADSs registered in public road traffic. For most of the systems currently operating in road traffic, the driving task has to be continuously monitored by a human being. Nevertheless, the automated driving function of some level 2 (term definition in Section 2.1) vehicles [7–10] and level 3 vehicles [11, 12] was found as one of the root causes (despite lacking supervision by the human) for fatal accidents that occurred in the past years.

The reason for the safety of ADS being such a challenging task is the result of a complex Software (SW) paired with an infinite amount of possible situations a vehicle can face during operation. The range of methods used in this domain is wide. In the domain of motion planning alone, there exist various search-based, optimization-based and even Artificial Intelligence (AI)-based approaches [13, 14]. Furthermore, it is expected that the share of AI-based methods will grow continuously in the coming years. As soon as the human no longer acts as a monitor and fallback for the system, public acceptance requires evidence that the system is safer than a human driver by a significant factor. However, since human road traffic accidents are extremely rare events—in Germany, for example, in 2019 a fatal accident occurred only every 250×10^6 km [15]—an ADS would have to pass a 2.5×10^9 km test drive, to show with 5 % probability of error that an ADS causes less accidents than an average human driver [16]. Winner [16, pp. 1174-1177] provides

further mathematical background and reasoning for the derivation of the required ratio between manual incident-free kilometers and required autonomous test kilometers.

Opposed to the development towards enhanced efficiency and functionality, the safeguarding and approval of such complex and frequently changing (online learning and/or updates) SW stacks gets progressively more challenging [17]. Relying on standard methods, approval would require unbearably many test cases and systems continuing their learning process at the consumer are not tackled at all. Furthermore, the more complex an ADS function is, the more often SW updates are inevitable, which render previous post-release evaluations and tests invalid and require a new review. This trend indicates the need of new safeguarding and approval methods [18, p. 21, 19]. Consequently, the superordinate guiding thesis to be supported in this work is:

T 1: *Complex and frequently changing SW for driving functions of ADSs can be safeguarded, while taking into account applicable standards.*

Based on the analysis of related work (Chapter 2), Online Verification (OV) emerges as one of the most promising but as yet little holistically studied methods for safeguarding of complex and frequently changing SW. This fact is investigated to support the second guiding thesis:

T 2: *A procedure can be found for the development of an OV method for safeguarding ADS functions, taking into account applicable standards.*

After a generic approach to OV of driving functions has been elaborated, the required steps for the illustrative implementation for a trajectory planner of a real-world automated race vehicle are elaborated. The objective of the implementation is to show a concrete application of the OV concept with all the challenges of a real-world setting and to serve as a proof of concept. Research in this regard serves to support the third guiding thesis:

T 3: *A prototypical implementation of an OV for the trajectory planner of a real-world automated race vehicle demonstrates the viability and serves as proof of concept.*

On the way of supporting the guiding theses stated above, this paragraph outlines some of the challenges faced for a stringent safety approval for trajectory planners. In particular, AI-based and complex methods are not yet covered by current standards. In order to guarantee safety within a certain context given applicable standards, transparent and deterministic approaches have to be applied. Due to the inherently unsafe nature of road traffic, deterministic OV methods that rely on worst-case assumptions for other Traffic Participants (TPs) would not always allow a safety approval paired with fluent traffic flow. In an unconstrained scenario, every TP has the possibility to crash into another vehicle on purpose [20, p. 2]. Therefore, a deterministic and conservative rating would always declare such situations as unsafe. A schematic illustration for this problem is depicted in Figure 1.1, where two race vehicle drive alongside. By nature, this is a common situation in a race that most drivers handle without problems. However, if the behavior of the vehicles is unconstrained (no rules or knowledge applied) and unknown, all physically possible maneuvers have to be considered (Figure 1.1a). Since this situation could potentially result in a collision (the trajectory of the ego-vehicle intersects the determined set), it has to be rated as unsafe. By contrast, if the vehicles know or share their plans via Vehicle-to-Vehicle (V2V)-communication or constrain their motion via rules, the situation can be classified as safe (Figure 1.1b). Within this thesis, the focus is on a deterministic implementation with guarantees, while not relying on knowledge about other vehicles' actually intended maneuvers. This is enabled by relying on a not-at-fault policy given applicable regulations, i.e. accidents caused by the ego-vehicle in accordance with applicable regulations have to be avoided.

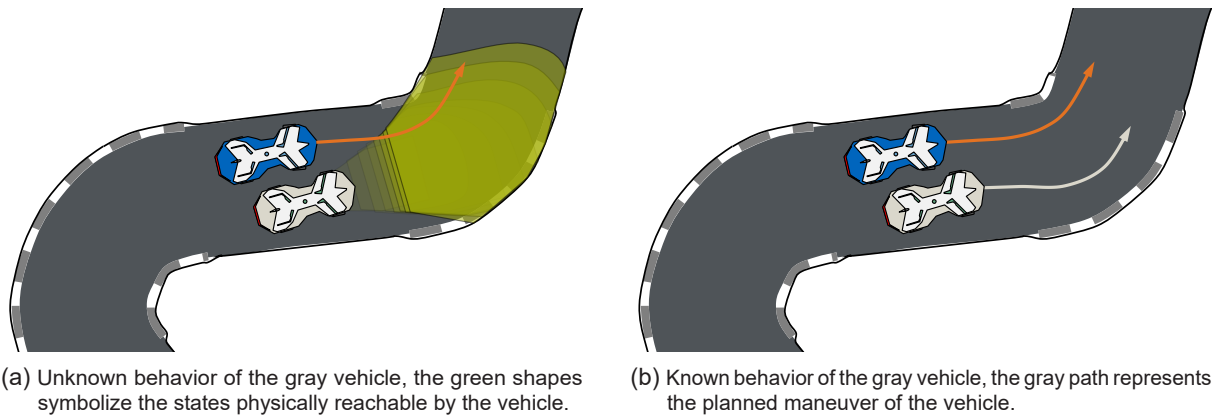


Figure 1.1: Schematic illustration of differences in safety challenges for the ego-vehicle (blue) between unknown and known behavior of a TP (gray).

The developed framework for OV of an ADS component targets on a generic principle, serving as a template for the development of any OV module. One specific and illustrative realization is showcased on the use-case of safeguarding the trajectory planner of a real-world autonomous race vehicle. The next chapter highlights related work from which the research questions and methodology of this thesis are derived in Chapter 3. In this context, the outline of this thesis is worked out and presented there.

2 Related Work

In this section it is deduced why the developed OV framework for trajectory planners poses a meaningful contribution to the research of ADS's safety. First, terms necessary for an equal basis of interpretation are defined (Section 2.1). Providing further fundamentals, a brief introduction to the automated driving SW architecture (Section 2.2) and principles as well as challenges of motion planning concepts (Section 2.3) are given. Subsequently, Section 2.4 provides an overview in the domain of automotive safety. This includes currently applied standards and procedures as well as safety assessment techniques actively developed in the research domain. In Section 2.5, a special focus will be placed on the research work carried out to date in the field of online monitoring approaches for safety assessment.

It should be noted that this related work section is intended to provide an overview in order to stringently point out the research gaps and to derive the guiding theses and questions for this work. Accordingly, this section does not necessarily go into depth and mathematical details. Where mathematical details from related work are required for the purpose of this thesis, they are separately presented in preliminaries in the respective methods section.

2.1 Terms and Definitions

In order to establish a consistent base of understanding, the most relevant terms for the scope of this work are defined in the following. The definitions follow common usage and definitions in related work. It should be noted that in the case of multiple interpretations of a term in usage or literature, only the one specified here is valid in the context of this work. The terms and definitions are presented in a logical order and are therefore not sorted alphabetically.

Artificial Intelligence (AI): Any algorithm that simulates smart behavior is called AI [21, p. 15]. Machine Learning (ML) is a subclass in which algorithms learn and modify itself over time without human input [22, p. 7]. Neural Networks (NNs) have proven to be a particularly efficient method for mapping complex relationships in the field of ML [22, pp. 12-15]. Deep learning is the subclass of multi-layer NNs that are used in particular for image processing. Figure 2.1 depicts an overview of these fields. Popular methods are assigned to the clusters without further explanation.

Complex SW: In the context of this work, a SW or function is interpreted as complex, if it holds properties conforming to the conception in ISO 26262-6:2018, 7.4.3 [23]. Accordingly, the following properties are a selection of indications for a complex SW:

- Highly branched data flow
- High number of parameters
- Comprehensible only to a few experts or project members

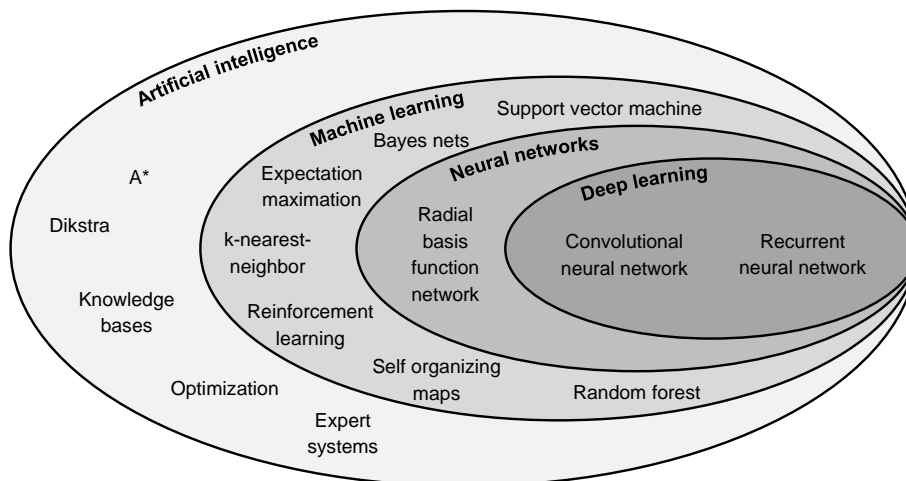


Figure 2.1: Subdivision and classification of terms in the field of AI, based on [22, p. 6]. Popular methods are assigned to the clusters, but are not explained any further.

Automation Levels: In 2014, the Society of Automotive Engineers (SAE) published the first version of a standard [24, p. 17] defining a taxonomy for levels of automation for driving automation systems. In the meantime, the terms have become well established and widely used in the automotive domain. The SAE distinguishes between the following six levels:

- Level 0: No automation
- Level 1: Driver assistance (either longitudinal or lateral automation)
- Level 2: Partial automation (human supervision and human fallback)
- Level 3: Conditional automation (no human supervision, but human fallback)
- Level 4: High automation (limited Operational Design Domain (ODD))
- Level 5: Full automation

Consequently, current driver assistance systems such as lane keeping assist or distance control systems are classified as level 1 and systems that combine these, such as the Tesla Autopilot [25], are classified as level 2. In the context of this work, the notation with an appended “+” sign describes the corresponding level and all levels above it, for example, level 3+ includes levels 3, 4, and 5.

Automated Driving System (ADS): According to the declaration of SAE [24, p. 3] a Hardware (HW) and SW that is collectively capable of level 3+ automation is referred to as ADS. When referring to the ADS operated on a test vehicle, this work refers to this specific vehicle as “ego-vehicle”. The vehicles deployed for the use-case of this work are operated corresponding to SAE level 4, but the described procedure is valid for the generic class of ADS. It should be noted that, in contrast, the term “driving automation system” in the context of this work refers to level 1+ vehicles. Apart from the definitions of the SAE, in this work ADAS are systems conforming to level 1 or 2.

Operational Design Domain (ODD): The ODD limits the permissible operating conditions for an ADS to a set specified by the manufacturer [24, p. 12]. The restrictions can be of different nature and may, for example, include spatial restrictions (e.g. certain roads), but also temporal restrictions (e.g. only during the day) or weather-related restrictions (e.g. not during rain).

Scene: Following the notion of Ulbrich et al. [26], a scene (Figure 2.2a) describes a snapshot of the environment at a specific point in time. In this context, all static entities in the environment (road, obstacles, etc.) as well as all dynamic elements (vehicles, pedestrians, etc.) are specified.

Situation: A situation is derived from a specific scene and describes an element's point of view [26]. A schematic situation for the orange vehicle in Figure 2.2a is depicted in Figure 2.2b.

Scenario: According to the interpretation of Ulbrich et al. [26], a scenario is consequently described as the temporal development of several scenes in a sequence. The scenario starts with an initial scene and is modified by actions, events, goals, and values. A scenario spans a finite time interval. Figure 2.2c shows a schematic scenario based on the initial scene from Figure 2.2a.

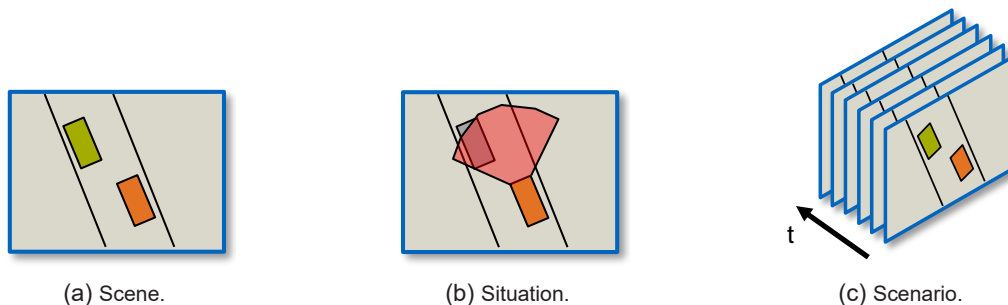


Figure 2.2: Schematic illustration of the terms scene, situation, and scenario.

Verification: Verification describes the process of examining whether a system meets the specification/requirements [27]. An exemplary simplified verification process is shown in Figure 2.3.

Validation: Validation describes the process of examining whether the expectations and the specific intended purposes of the customer/developer are fulfilled [27]. In this process, tests are not performed against individual, concretely derived requirements, but rather an entire or real system is typically evaluated. An example of a simplified validation process is shown in Figure 2.3

Safety Assessment: Safety assessment is a widely used term in related work and is consequently adopted as an umbrella term in this thesis. Accordingly, it is used to describe the process of a safety rating in any form. This measure can be performed both online via a monitor as well as offline via evaluation tests or detailed analyses. It is important to note that methods in this group do not necessarily have to be exhaustive or provide a comprehensive or totaling safety score. Accordingly, the evaluation of a certain aspect (e.g. a collision risk w.r.t. other TPs) is also covered by this term. However, depending on the scope and derivation of the measure, it can provide safety guarantees and thus be qualified as a safeguarding measure.

Safeguarding: In this work, safeguarding describes an extension of a basic “safety assessment” by further measures that provide safety guarantees. This includes methods which, in addition to the safety assessment, guarantee by means of suitable measures or observations in evaluations that a specified part of a system or the entire system operates within a safe state-space at any time (under given constraints).

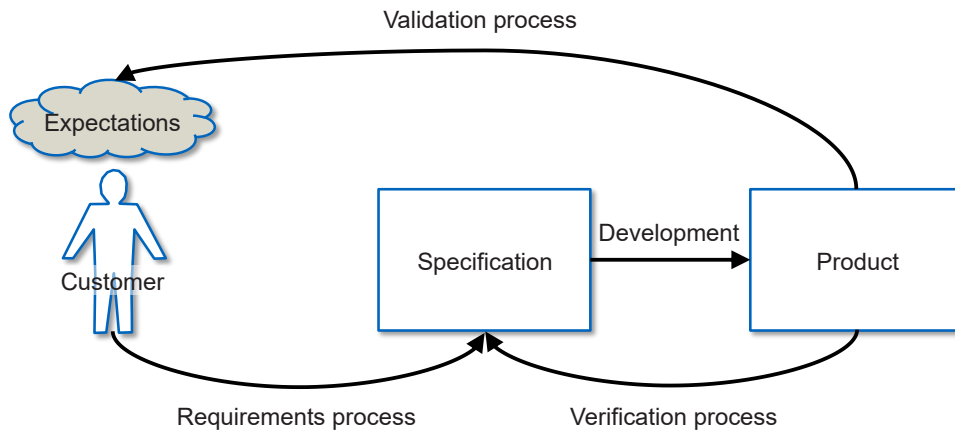


Figure 2.3: Simplified schematic illustration of the difference between verification and validation, based on [27]. Here shown for a single-stage development process. With multi-layer development processes (e.g. in accordance with the V-model) the verification can also take place between different layers against the respective requirement.

Approval: Approval or type approval (also known as homologation) describes the process of measures that are necessary in a final step for a regulatory permission for market introduction. This process is therefore primarily relevant for systems with a valid safeguarding concept.

2.2 Automated Driving Software Architecture

ADSs follow the basic robot control paradigm sense-plan-act, originating in the late 1980's [21, p. 105]. A widely used adoption of this deliberative scheme for ADSs is shown in Figure 2.4. In this variant the “sense”-component is split into a “sensors” and “perception” block and the “act”-component is represented by “control” and “actuators”. In the following, the three (inner) automated driving software blocks will be outlined briefly along with references to methods that have been used to date. In this section, the references are limited to the racing domain, which is also the domain of the use-case in this thesis, without loss of generality (similar SW architectures). Interested readers are referred to a survey by Pendleton et al. [28] for a comprehensive overview of all SW components including examples in regular road traffic.

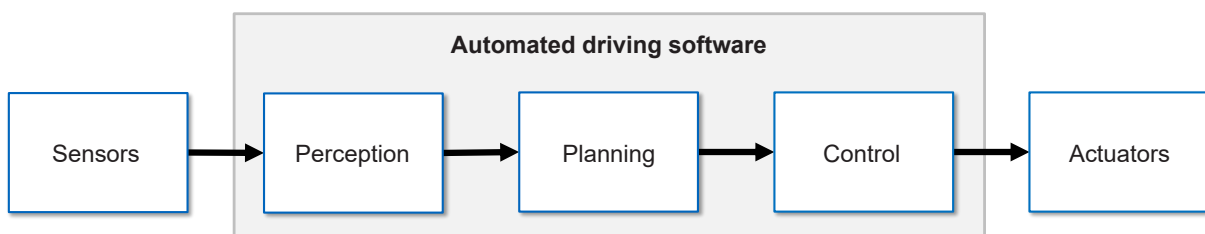


Figure 2.4: Basic automated driving architecture.

Perception: The perception module contains all the functions necessary to perceive relevant aspects of the environment and to localize the ego-vehicle within it.

In this process, the drivable space is to be identified [29] and, depending on the application together with further aspects, stored in a map [30, 31]. During runtime, the vehicle uses this map to plan trajectories in the drivable space and/or to localize itself in it. The localization is done globally via dedicated sensors (e.g. Differential Global Positioning System (DGPS) systems) or relatively via Light Detection and Ranging (LIDAR) [32–36] and/or camera localization [33] by comparing features in the perceived situation with features in the map. A fusion of multiple sources of localization data is done for instance by a particle [33, 37] or Kalman [35, 38] filter. So-called Simultaneous Localization and Mapping (SLAM) methods [39, 40] combine mapping and localization by extending the map at runtime.

A further major aspect is to detect certain static objects [37, 41–43] and/or vehicles [44, 45]. This is commonly accomplished using NNs on camera or LIDAR data. The result is typically a list containing the coordinates and other properties of relevant objects in the scene.

Planning: The planning module, also known as motion planning module, is responsible for generating movement plans for the vehicle to follow. In this context, global planning is often distinguished from local planning.

In global planning, a rough route from start to goal is generated without taking dynamic obstacles into account. In racing, for example, the global planner generates an optimal race line for an entire lap. In this process, details such as the exact velocity or acceleration of the vehicle at individual points are less relevant.

In local planning, dynamic obstacles are taken into account and a detailed trajectory is generated for a specific, limited planning horizon. The local plan may be similar or identical to the global plan in the case of no other TPs nearby, but will usually deviate from it in their vicinity.

Since the use-case in this work is oriented towards the planning module, Section 2.3 details further on motion planning techniques.

Control: The controller has the task of guiding and stabilizing the vehicle along a planned trajectory (spatial path and temporal course of velocity). In this process, it is necessary to generate suitable command signals for the actuators of the vehicle on the basis of the planning data. A number of approaches have been established for this purpose. Among the currently studied approaches are classical path tracking controllers [46–51], model predictive control [52–58], and learning-based methods [59–65].

In addition to controllers that perform the pure driving task, there are also individual dedicated approaches that investigate special applications, for example, stable drifting [66–68].

2.3 Motion Planning

In the domain of motion planning, a vast variety of approaches have been developed in the past decades. Recent surveys [13, 69, 70] distinguished four types of underlying concepts. As presented in previous work [71], a combination of their findings results in the following four clusters of motion-planning approaches:

- Sampling-based approaches / incremental search
- Interpolation methods

- Numerical optimization / variation-based methods
- Graph-based approaches

Besides these classic motion planning approaches, several ML-based approaches are emerging recently [72]. Subsection 2.3.1 details on the four clusters of classic motion planning approaches, while Subsection 2.3.2 outlines the findings in the domain of ML-based motion planning. This section includes references from the field of automated driving in road traffic as well as from racing.

2.3.1 Classic Approaches

Sampling-based methods incrementally explore the surrounding space by placing samples and connecting them with path segments. Once the desired goal is reached by repeating this process, the identified path is further optimized and selected as final solution. Commonly this approach is used for unstructured or unknown environments [73] and is challenging when targeting the generation of smooth or curvature minimal paths. The Rapidly-Exploring Random Tree (RRT) is best known in this domain and often used for global planning tasks [74–76] or between remote waypoints [77]. Nevertheless, there are approaches [78–81] that investigate the use of RRT as basis for local planning in simulation.

Interpolation-based methods generate paths or trajectories for given coordinates while fulfilling given constraints. Therefore, their primary focus is not on the actual planning task, but rather on the establishment of a smooth transition between multiple coordinates. This is why these approaches are also found within graph or sampling-based methods when generating the final path from identified nodes or samples. Among the most known strategies are Béziers [82–84], polynomial splines [85, 86], and clothoids [87].

Numerical optimization approaches describe the motion-planning task as a mathematical minimization or maximization problem. Since these methods tend to detect local minima only, the problem is often formulated in a convex manner, which in turn reduces the solution space.

The spectrum of optimization-based motion planning approaches itself is broad and addressed in a large number of publications. Among these, Model Predictive Control (MPC) is one of the most common and promising. The most prominent application of MPC is for urban or highway motion planning problems [13, 69, 70]. Depending on the planning task, different boundary conditions and requirements have to be met. Optimization-based approaches, including learning MPC, have been also applied to automated racing projects [88–92]. However, dynamical objects are not considered in any of these approaches. In order to cope with dynamic objects at high velocities, far planning horizons are required, which in turn challenge the real-time capability.

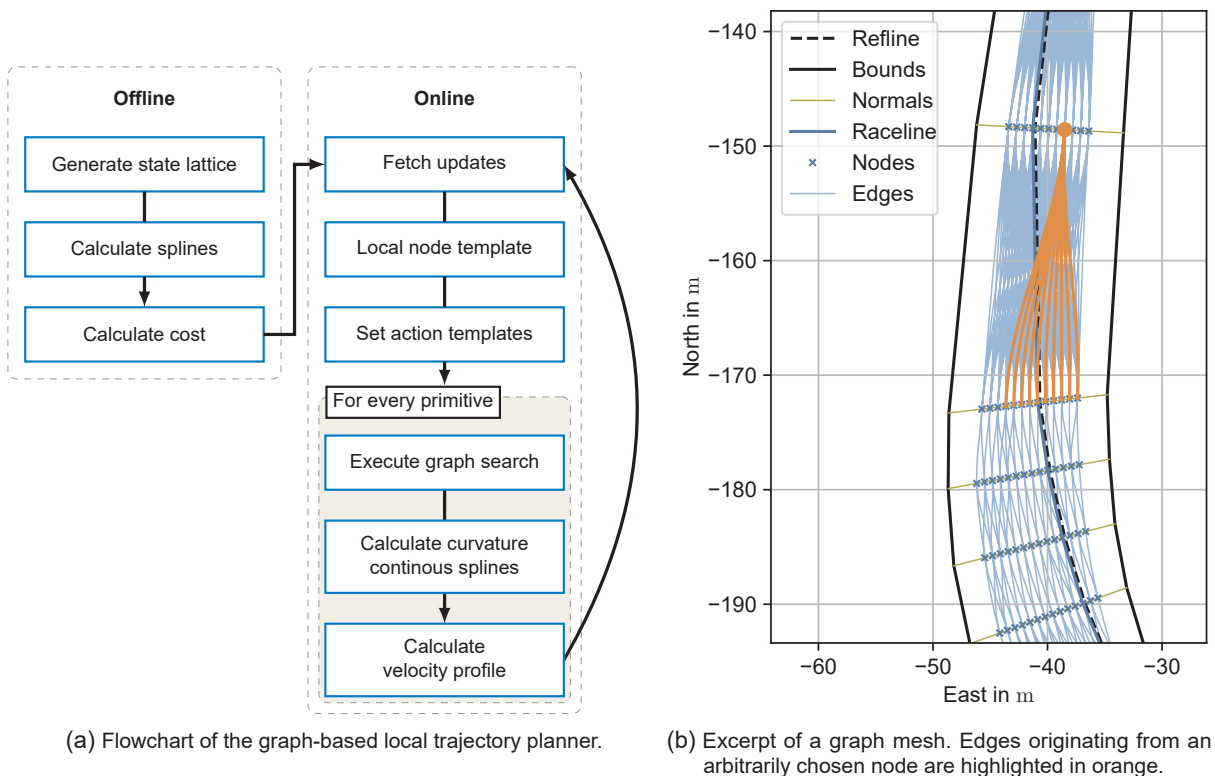
While the optimization approaches described above represent local planners, optimization is furthermore especially popular in the race domain for generating time- [93–98] or curvature-optimal [99–101] global race lines.

Graph-based methods make use of a spacial or spatio-temporal mesh that overlays the drivable area. Usually, the mesh is realized in form of a one-dimensional or hierarchical tree, with each edge assigned to a cost value. Standard search algorithms are used to solve for the path that minimizes the cost between a start and end node. Graph-based approaches have proven to be effective for trajectory planning [102–106].

Spatial, one-layered graphs [102–104] are established by sampling smooth transitions to various laterally and sometimes longitudinally distributed states within the track boundaries. Some of

them allow for interaction with other dynamic objects within a short time horizon. Gu et al. [105] stack multiple spatially arranged nodes and obtain a hierarchical tree. The edges are described by straight line segments, which are used as an input to an adjacent path optimization. McNaughton et al. [106] add the time dimension to the graph and result in a spatio-temporal hierarchical tree. The solution to this graph is a trajectory holding a globally optimal velocity profile. However, it should be noted, that due to the curse of dimensionality, the discretization of time and spatial domain cannot be chosen arbitrarily small.

Within the *Roborace Season Alpha* [107], the first full-scale automated racing series, the author developed a graph-based trajectory planner, that addresses shortcomings of existing approaches by spanning a spatial hierarchical tree with a far planning horizon [71, 108]. The temporal problem is solved in an adjacent step. Since a large portion of the real-world evaluation data (Chapter 6) is generated with this planner, the concept is outlined in the following. A detailed elaboration can be found in [71]. In order to reduce the online computation load, the planner is split into an offline and online phase (Figure 2.5a).



(a) Flowchart of the graph-based local trajectory planner.

(b) Excerpt of a graph mesh. Edges originating from an arbitrarily chosen node are highlighted in orange.

Figure 2.5: Flowchart and lattice visualization of the developed graph-based trajectory planner [71].

The *offline* process starts with generating a state lattice overlaying the whole driving space. First, the imported track, defined by a reference line (Reflines) holding normal vectors (Normals) with a specified extension to the left and right (Bounds) [109], is used to set the graph's nodes. Therefore, equally spaced nodes are distributed on the track's normal vectors. The discretization is chosen around the race line of the global planner in such a way that it is guaranteed that one node per layer lies on the race line (Raceline). Second, edges are generated by calculating $C1$ continuous cubic splines between the nodes of adjoining layers (Figure 2.5b). Finally, each edge in the graph is linked to a cost value, which represents the effort of the vehicle following that segment. The curvature as well as the lateral displacement to the time-optimal race line are one of the most crucial parameters defining the cost of an edge.

In the *online* phase, a set of trajectories is generated by relying on the offline graph. The following steps are executed in an iterative manner: First, a local portion, ranging from the pose of the ego-vehicle to a specified planning horizon, is pulled out of the full offline graph via a local node template. Second, the nodes and edges in the planning range are manipulated (filtered out or altered cost) based on other objects and behavioral intents. For example, in order to overtake another vehicle on the left side, the vehicle to be overtaken (including its prediction) as well as all edges to the right of the vehicle are removed. That way, it is possible to plan various action primitives, like “straight”, “left” and “right” (here called action templates). Third, a cost optimal path along the graph’s edges is found in the remaining set of edges by executing a graph search (e.g., Dijkstra or A*). Finally, the generated paths are refined by calculating a C^2 continuous spline and completed to a trajectory by computing a matching velocity profile. The velocity profile thereby maximizes the feasible acceleration by applying a forward-backward solver [101].

2.3.2 Machine-Learning-Based Approaches

Classic motion planning algorithms may reach their limits in daily traffic soon. According to Nessler [110], rule-based, conventional, path planning is no longer sufficient when aiming for complex planning tasks in an all-encompassing environment with countless constellations and circumstances. According to his findings, ML methods are promising in this regard, but still need time for development and elaboration. In the following, an insight into current approaches for ML-based motion planning is provided and their challenges are pointed out.

In the late 1980s, the first end-to-end approach called ALVINN [111] was presented for the navigation of an ADS. A NN is fed with image data and LIDAR data and generates a steering angle from this, i.e. combining perception, planning, and control in one single module. NVIDIA published a similar approach with extended computing capacities and therefore larger NNs in the modern era [112]. End-to-end approaches have also been tested in a racing environment, for example in a realistic racing computer game [113–115] or with small scale vehicles [116]. While end-to-end approaches deliver remarkable results given their straightforward architecture, the approaches still suffer from lack of robustness [112]. In addition, there is the problem that the system is not interpretable and no foresighted planning is generated. Thus, individual incorrect decisions can lead to abrupt critical maneuvers without any feasible emergency maneuver. Accordingly, the approaches of this category are not in the scope of this work, as is the case with safety experts from well-known Original Equipment Manufacturers (OEMs) [117].

By contrast, ML approaches that only perform individual tasks of the driving stack offer more transparency. In the area of behavior planning, deep Reinforcement Learning (RL) approaches can be found. In a merging scenario, for example, such approaches are used to generate the decisions in the merging process using previously learned experiences [118–120]. When using ML for pure behavior planning, decision making may be not interpretable, but a trajectory can still be generated that describes the future intended behavior of the ego-vehicle.

However, there are also approaches that generate trajectories, similar to classical approaches. Cai et al. [121] use several NNs that generate trajectories from actual camera images. Another concept that Waymo has presented is trajectory planning using NNs [14]. In this process, the environment is represented in a simplified bird’s-eye view and made available to the NN as input. The NN then generates trajectory points based on previously learned weights. A similar approach is taken by Chen et al. [122]. While in these approaches, as with end-to-end approaches, the decision-making itself is not interpretable, there is a trajectory that clearly describes the future

intentions of the vehicle. By planning ahead in this way, there is a chance to identify abrupt or incorrect decisions upfront and to trigger appropriate emergency maneuvers.

Since a trajectory planner behaves differently depending on the situation, the risk that the system might behave critically in certain situations is to be minimized. This applies to both classic and, in particular, AI-based planners. Since there is a wide variety of possible scenarios and situations an ADS could face, safeguarding is a major ongoing challenge. In the following, existing standards in the area of safeguarding and approval are first outlined and then their constraints and limitations are discussed.

2.4 Automotive Safety

Within the engineering domain, “safety” is understood as the absence of unacceptable risks [123, p. 3]. Since unacceptable risks are intolerable in public traffic, safety assessment and approval for ADSs is one of the key factors influencing whether such agents will hit the public roads or not. Industry and several research institutions proposed theorems, methods and approaches addressing this domain [124]. The following sections will cover existing safeguarding standards and procedures applied in industry (Subsection 2.4.1) as well as published safety assessment techniques in the research domain (Subsection 2.4.2).

2.4.1 Regulations and Approval Procedures

During the 20th century, few, if any, electric or SW parts have been embedded in commercial vehicles. Thus, proper mechanical design with high safety factors ensured safety. Up till now, opposed to electrical parts, properly built mechanical devices are considered as safe and do not need any additional safety precautions [23]. Once Electrical and/or Electronic (E/E) systems are added, it becomes clear that these systems continue to be vulnerable to failure or misbehavior.

Therefore, before commercial vehicles are introduced to the market, they have to pass a type-approval (also known as homologation or certification) that guarantees a minimum level of safety. For this approval, the legislator defines a set of requirements that have to be verified within specified tests. Based on the country, the homologation is achieved by self-certification (executed by the OEM) or certification by an independent third party [125]. The latter one is common in Europe and executed by a technical service. The framework for the tests is provided by the *United Nations Economic Commission for Europe* (UNECE). As part of the so-called 1958 Agreement [126], there are 159 regulations that have to be fulfilled before a vehicle can be released for sale (as of September 2021). However, SW approval does not guarantee safety thoroughly, but rather provides a confirmation of a minimum level of safety by testing selective samples. A comprehensive safeguarding has to be carried out by the OEM in the context of product liability. This aspect—the creation of a system that is safe and ready for type approval and certification—is also the focus of this thesis as opposed to the more political venture of implementing type approval and certification.

ISO Standards

In the spirit of uniform safety and security standards, the automotive industry subsequently launched specific standards that provide recommendations for detecting possible hazards and developing suitable steps to ensure a certain degree of safety. In this domain, safety and security

is considered as the interplay of three partial aspects (Figure 2.6), each addressed in a dedicated standard [123, 127]:

- Functional Safety (FuSa) - ISO 26262 [23]
- Safety of the Intended Functionality (SOTIF), also known as “safety in use” and “functional performance” - ISO/WD PAS 21448 [128]
- (Cyber-)Security - ISO/SAE AWI 21434 [129, 130]

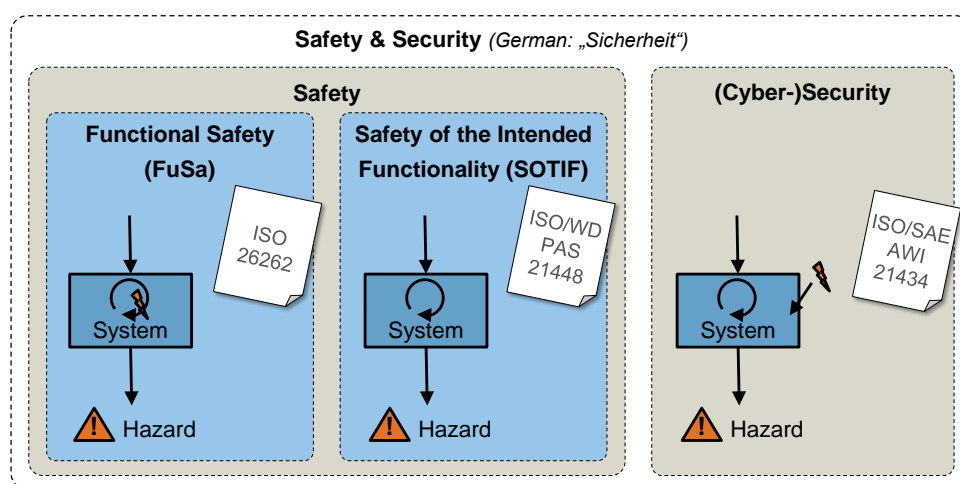


Figure 2.6: Overview of the three types of safety and security with associated standards. The meaning of the individual terms is illustrated by a symbolic system that transitions into a hazard. In this context, a possible cause of error and its location is represented by an orange lightning bolt.

This thesis addresses the “operational safety”: FuSa and SOTIF, excluding the field of security. Nevertheless, it should be noted that cybersecurity (protection against potential hazards caused by external interventions) is an important aspect and hosts a large field of research itself [131, 132]. Furthermore, since safety approval of HW components, including E/E HW tackled in the ISO 26262, is state of the art (as elaborated in the first paragraph), the author narrows the focus on SW development. This branch of safety is called algorithmic safety in the context of this thesis. In the following, the two most relevant standards ISO 26262 and ISO 21448 along with their core topics FuSa and SOTIF are outlined¹.

The goal of the FuSa (Figure 2.6, left) is to protect against potential hazards due to faults in the system. The *ISO 26262* is designed to specify and support FuSa in the entire life-cycle of electronic and electrical safety-related systems in consumer passenger cars. In this regard, the norm is aligned with the V-model [134] and deals with functional safety aspects on every development step in the model (including requirements, design, implementation, integration, verification, validation, and operation). Furthermore, it defines an approach to determine and rate the risk of individual system functions with Automotive Safety Integrity Levels (ASILs). The ASILs—four levels, ranging from A to D—are used to assign the required safety measures for the rated item in order to attain a defined acceptable residual risk. In addition, the norm specifies validation procedures ensuring an acceptable and sufficient level of safety.

The goal of SOTIF (Figure 2.6, middle) is to protect against potential hazards caused by the system without a fault. The *ISO 21448* covers the central aspect of SOTIF, which is the goal of

¹It should be noted, that further standards like the UL 4600 [133] exist, but are less common in the European countries.

finding a way to specify, develop, verify, and validate a target function such that it is considered sufficiently safe. The causes for unacceptable risks are sought in particular in inadequacies of the intended function or through foreseeable misuse by the operator. In order to turn for the better, the standard provides information on the desired features of a product as well as on a holistic testing and development process, reducing the risk of missing out some risks. In doing so, the following four main fields are worked with: known safe scenarios (to be maximized during development), known uncertain scenarios (to be mastered), unknown safe scenarios (no further measures), and unknown uncertain scenarios (to be uncovered and moved to any of the other classes).

In the course of this thesis, the scheme used to reference individual clauses within the standards is adopted from the standards themselves. Accordingly, “ISO 26262-4:2018, 8.4.3.4” refers to clause 8.4.3.4 in part 4 of the 2018 version of ISO 26262.

Limits of the Standards

The further the development of automation advances, the more complex becomes the SW to fulfill the driving task. Some tasks are targeted by AI algorithms, with the underlying decision metric not being explainable by human experts [135]. The current standards—especially the ISO 26262, derived from principles developed in the 1990s—reach their limit at that point [117, 136–143]. The following four paragraphs highlight some exemplary weak spots of the ISO 26262.

New types of hazards. According to the ISO 26262, harms are “caused by malfunctioning behavior of the item” [23]. However, following the findings of Salay et al. [136], ML can cause new types of hazards. For example, faults in the reward function of a RL framework can cause the agent to negatively affect the environment in order to achieve its goal [144, p. 3]. An illustrative example is a hypothetical vacuum cleaner robot, which is equipped with a RL function: If the reward is defined by the amount of dirt collected, the robot will adjust its behavior to primarily clean particularly dirty areas and skip other areas. A suitably selected reward function would therefore have to aim for reduction of uncleaned areas. Since these behavioral changes take effect during online training, after the validation phase, standard validation procedures are undermined and it cannot be guaranteed that the system stays within safe bounds [145, 146].

Training data hinders specification. The ISO 26262 builds upon the V-model, which is based on the premise that a component is fully specified on any level in the process. A key principle in the standard is to verify every refinement against the specification of its parent [147]. Only if this connection is intact, it is possible to trace and link hazards or safety goals to the actual implementation. However, when making use of ML, the algorithms alone do not allow for a detailed specification of the behavior. The behavior of the SW is significantly affected by the training set, which inevitably has gaps and no proof of consistent hazard avoidance can be provided [136, 148, 149].

Transparency of algorithms is lost. In order to have a manageable, contained, and safe SW architecture, the ISO 26262 demands the use of a modular structure. Furthermore, the standard restricts the allowed size of these modules in order to keep the transparency and internal cohesion high. However, when deploying complex, stochastic, or even AI-based approaches, even a single module can become non-transparent and leave the range of what can be explained by human experts [146, 148–151].

Development techniques tailored for manual vehicles. The current version 2018 of the ISO 26262 is designed for manual vehicles with simple algorithms only. Within part 6 of the standard, SW

development techniques are recommended in order to mitigate the onset of errors. However, according to Salay et al. [136], several of those techniques cannot be applied to complex or ML-based approaches. Likewise, the ASIL assessment in the concept phase of the standard is strongly focused on manual vehicles. The controllability is one of the three evaluation pillars for this metric and describes the handling of a hazard situation by the driver. Monkhouse et al. [137] emphasize that in the case of SAE level 3+ vehicles, the concept of controllability is not designed for ADSs and needs to be adapted.

The ISO/WD PAS 21448 (SOTIF) differs a lot from the ISO 26262 in regards of instructions and applicability to SAE level 3+ vehicles. The SOTIF standard allows the application for ADSs, although further measures might be necessary in individual cases (ISO/PAS 21448-1:2019 [128]). Therefore, the standard faces fewer issues compared to the ISO 26262, as elaborated above. However, the first version of the standard is vague in its recommendations or requirements and leaves the user a lot of flexibility in interpretation. Therefore, concrete implementations have not been made so far.

In conclusion, the following two key properties of a driving function hinder the usage/integration according to existing standards:

Online adaptation: Systems that continue learning after the validation phase can develop an unsafe behavior (new types of hazards and no specification possible due to training data).

Complex or non-transparent: The module cannot be formally approved by an expert (non-transparent functions and standards tailored for manual vehicles).

As a consequence, new safeguarding approaches have to be developed. The following section gives an overview of existing safety assessment approaches in the research domain.

2.4.2 Safety Assessment Techniques

In related work, there exist multiple techniques assessing SOTIF- and FuSa-related capabilities of ADSs. Each of the existing approaches copes differently well with the issues elaborated in the previous section. This section gives an overview of existing approaches. Both currently applied methods and active research fields are listed. Not all of the approaches shown have been fully solved and guarantee safety. Consequently, the methods are classified with respect to their challenges and the set of techniques further investigated in this thesis is narrowed down accordingly. In this spirit, the author fused and clustered the approaches (Figure 2.7) summarized in existing survey papers [152–156]. Each of the clusters is further elaborated in the following.

Validation/Tests

One of the most prominent approaches to verify sufficiently safe behavior is based on tests. Initially, these tests took place primarily in the *real world*, but recently concepts for *simulative* testing have also been developed. The following paragraphs cover prominent techniques from both domains.

A straight forward *real world* approach to prove that an ADS is equally safe or better compared to human drivers is to produce a statistically significant result based on kilometers driven. Since human drivers are on average involved in fatal accidents every 614×10^6 km on highways, a statistically significant statement for an ADS outperforming a human can only be made with several billions of test-kilometers driven in autonomous mode [157]. This phenomenon is referred

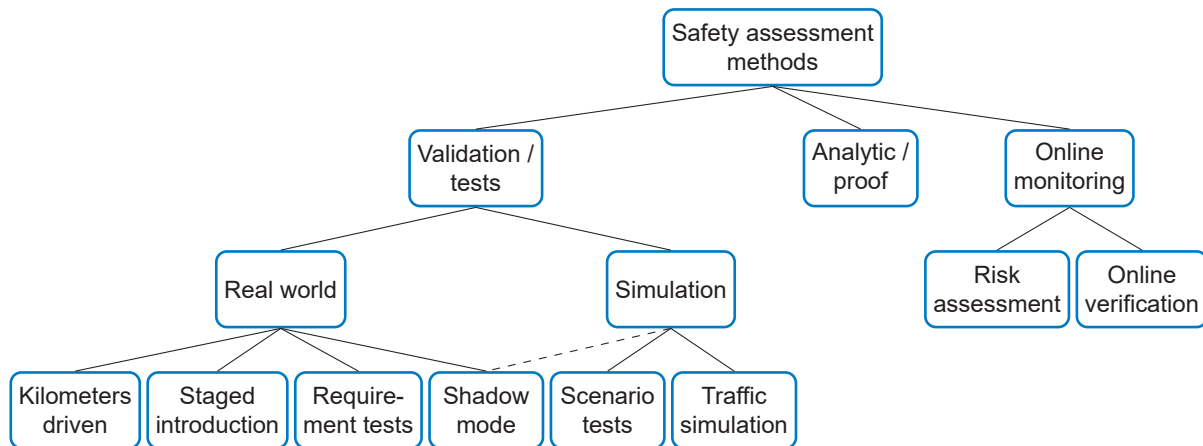


Figure 2.7: Clustered overview of existing safety assessment approaches.

to as the “approval trap” by Winner [19], since in order to generate that many test kilometers market introduction would be required before approval.

Due to this contradiction there are attenuated approaches that pursue this goal. One method is termed “shadow mode” and considers the use of the sensors of a level-5-capable vehicle that runs in manual mode. While a human driver steers the vehicle, the perceived sensor data is used to calculate maneuvers, which are then evaluated for possible collisions. This approach is not only introduced in research [158], but also Tesla claimed at the “Tesla Autonomy Day” [159] that this technique is applied.

A further approach is the staged introduction, where the ODD is deliberately limited (e.g., initially only on the freeway up to 60.0 km h^{-1} or a well mapped specific region), such that considerably less tests are necessary in advance. When the public fleet has accumulated many operating hours in the limited ODD, the ODD can be expanded step by step [160, p. 16, 161].

Even though the two methods presented are already actively applied and are achieving first successes, they bear limitations. With the shadow mode, the prediction of other TPs assumes/simulates a behavior that will not always correspond to reality. Furthermore, the actual behavioral reactions of the TPs to a deviating (calculated) behavior of the ego-vehicle compared to the human driver are missing. For these reasons, such tests primarily validate the virtual version of traffic but less the actual reality. Also, the staged introduction only scales up to a certain extent. Since highly accurate maps and specific tests have to be carried out on all routes for each of these stages, expansion is only possible on a feasible scale.

Besides relying on long runs in the real world, a well-established procedure examines whether the specified requirements of a driving function are met. Special test cases and maneuvers are specified, which are then evaluated on a test site. For ADAS in particular, this principle has been applied and is defined in corresponding UNECE regulations (e.g. UNECE-R 131 for emergency brake assistants). Ponn et al. [162] investigated a method to extend such test procedures towards approval of ADSs.

The challenge of this approach is scalability with the complexity of automated driving. Previous ADAS are of a relatively simple nature and come with clearly specifiable functionalities in a defined ODD. In this context, it is possible to cover almost all boundary conditions with dedicated tests at a manageable cost. However, with complex SW for ADSs, which is supposed to react adequately in a multitude of scenarios, it is no longer possible to specify these requirements. Furthermore, there is the problem that once a fixed set of scenarios would be defined that the

vehicle has to pass, it is likely that the OEM would develop the vehicle in a target-oriented way. Thus, over-fitting to the test scenarios is possible. Accordingly, the vehicle could master the test scenarios, but fail to respond adequately in other scenarios.

By using *simulation*, developers believe that they will be able to master the challenge of approval. The advantages compared to real tests are obviously the comparatively low costs, the possibility of parallel execution, and the abstinence of risks (both injuries and costs). Nevertheless, there are still some difficulties that need to be overcome.

One approach in this area is traffic simulation. The goal is to simulate real traffic in an artificial world and test the ADS in it [163, 164]. Commonly, however, the intention here is not to test a single ADS, but rather to assess potential road safety improvements in mixed or purely automated traffic.

Another approach with many supporters is scenario-based testing. The goal is to generate or automatically extract critical or challenging scenarios for vehicle testing [165–179]. The idea is that the number of necessary driven kilometers or operating hours can be significantly reduced because instead of “boring” driving segments only relevant sequences are tested. Ponn [179, p. 7] depicts these frequency ratios with a pyramid (Figure 2.8a), whereby corner cases (most challenging) and challenging scenarios (difficult for an ADS to master) occur significantly less frequent than the remaining scenarios. Consequently, with the same number of tests, they allow a statement about the safety of the entire system to be made significantly earlier. Another important aspect of scenario coverage is the analysis of yet unknown scenarios, as unsafe situations may arise in these cases. The ISO 21448 [128, p. 7] divides the entire scenario space into known and unknown situations as well as into safe and unsafe situations (Figure 2.8b) and aims to reduce unsafe and unknown areas. These are then to be converted into safe known areas, so that this class represents the largest portion. Most of the approaches are based on simulative scenarios, because in this case the boundary conditions can easily be defined. Nevertheless, there is also the possibility to map the found scenarios on a test field and evaluate them in the real world.

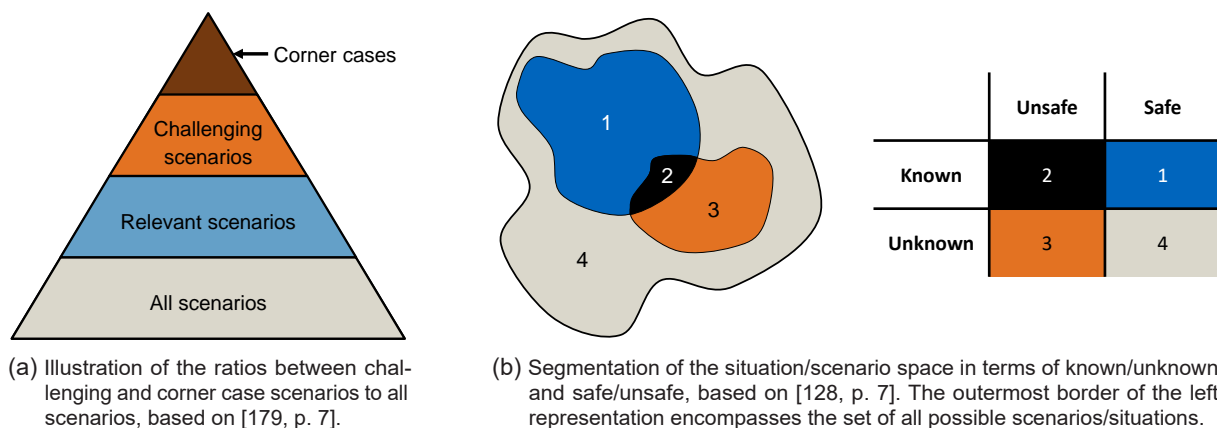


Figure 2.8: Subdivision of scenarios/situations with respect to distinct characteristics.

The challenges of simulative testing [180] are, on the one hand, the need for a sufficiently detailed representation of reality. Only if this gap is small enough it is possible to draw conclusions about the real system once a vehicle has been approved in simulation. Riedmaier et al. [181] propose a method to address this challenge, with sensor simulation in particular remaining difficult/challenging. On the other hand, as with the real tests, the question of the number of scenarios that have to be passed for approval is still an open research topic [167, 169]. In the

process, the selection of the types of scenarios is also still an active research topic and there is a residual risk of omitting individual situations.

In summary, the validation and testing methods offer innovative and promising advancements. However, all methods still face open challenges. In addition, the methods are not suitable to sustainably validate online learning AI, as the properties of the function may change for the worse after passing the tests.

Analytic / Proof

Especially for simple functions, it is possible to formally prove that a system is safe within an ODD. But there exist approaches [182] that aim to analytically prove safety for more complex systems.

Shalev-Shwartz et al. [20] use theorems for worst-case behaviors of all traffic agents in order to formally prove safety for a given planner based on mathematical induction. In a similar manner, theorem provers have been applied to attest safe conduct of ADASs [183, 184]. Besides, Linear Temporal Logic (LTL) is used to specify the automated behavior and formally construct safe functionalities [185, 186]. In some scenarios it is possible to determine barrier certificates, a function that guarantees to stay within safe states [187, 188].

However, the aforementioned approaches scale poorly with increasing function complexity and cannot provide safety guarantees for online-learning-based systems. Nevertheless, there exist some approaches aiming for safe AI. Survey papers [189–191] provide an overview in this domain. Hendricks et al. [192] target explainable and transparent AI, which in turn would benefit analytical approval. A classification algorithm is used to generate a descriptive statement about the reasons the underlying AI uses for generating a certain decision. It should be noted, that currently this method relies on another ML principle and therefore is hard to approve in total. A different approach is to establish a safety certification or verification for AI [193–196]. However, this is currently only possible for selected properties of a NN or limited to simple model structures.

In summary, analytic proofs work well for manageable function blocks but do not scale well to complex functions or AI. Methods targeting such properties are an active research domain.

Online Monitoring

Some authors [148, 18, p. 13] claim that current approaches are not able to cope with online learning methods. The reason for this is that the underlying system changes as it learns and thus invalidates all static validation done beforehand. In order to safeguard online learning or frequently changing code, system monitoring at runtime is suggested.

Since this type of safety assessment method is the only promising technique when facing online learning methods, related work in the field of online monitoring is analyzed thoroughly in the next section.

2.5 Online Monitoring

The findings of Subsection 2.4.2 show that online monitoring is the most promising approach for safeguarding of online learning and frequently changing functions. Accordingly, existing

approaches are reviewed and evaluated in this section. In the literature analysis, primarily methods used in the field of automated driving and motion planning (the use-case elaborated in the course of this thesis) are evaluated. However, isolated work relevant from other domains, for instance robotics and aviation applications, are also incorporated.

The remainder of this section details on related work in the two identified main clusters: risk assessment and OV methods (Figure 2.7). The stochastic online monitoring methods primarily calculate numerical representations of risk and are therefore referred to as online risk assessment (Subsection 2.5.1) in this section. The deterministic online monitoring methods examine against specified requirements and are therefore referred to as OV (Subsection 2.5.2) in this section. Finally, an overall valuation and classification of the presented approaches is given in Subsection 2.5.3.

2.5.1 Online Risk Assessment

In the domain of stochastic online monitoring approaches, a typical intention is the calculation of the collision probability or the collision risk. Various underlying techniques are used; the author identified five groups of approaches. Their basic principles are presented in the following. The first three clusters (maneuver prediction, stochastic occupation, and motion modeling) rely on a projection of other TPs into the future in order to relate them to the ego-vehicle for a risk assessment. The remaining two clusters take a different approach and classify into risk classes instead of a scalar risk value or calculate a scalar risk score based on available data without anticipating the position of the TP.

Maneuver prediction: One group of authors use various maneuver recognition algorithms to predict a TP's motion and to conclude in a collision probability with the ego-vehicle (Figure 2.9). The different underlying principles and nuances are summarized below.

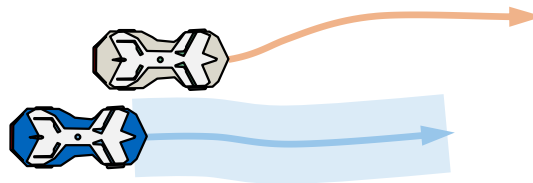


Figure 2.9: Illustrative sketch of underlying principles for the maneuver prediction based risk assessment approach with the ego-vehicle in blue and TP in gray.

Schreier et al. [197] and Schreier [198] use Bayesian inference to select the most probable maneuver among a set of possible maneuvers for all vehicles in the scene. The most probable maneuver is then used to calculate a collision probability by using a Monte Carlo simulation. Anell et al. [199] apply a similar strategy, but use an intention estimation paired with a probability field for movement prediction. Shangguan et al. [200] rely on a long-short-term-memory network to determine the movement of other road users, but then also use Monte Carlo simulation to determine the collision risk. Houenou et al. [201] apply formalized policies to reason from vehicle motion in relation to road geometry to detected maneuvers. The recognized maneuvers are then described by cubic polynomials. A combination with a more accurate physics-based prediction in the short term increases the accuracy. Again, Monte Carlo simulation is then used to infer the collision risk. Zhang et al. [202] recognize maneuvers with a simple temporal-spatial correlation of reference maneuvers and current driving history. Based on the detected maneuver, the motion is

then predicted with an associated polynomial. As with the approach of Houenou et al. [201], in the short time horizon, a more accurate physics-based prediction is incorporated. A proposed risk indicator is calculated by correlating the other vehicles' prediction with the ego-vehicle maneuver.

Stochastic occupation: Another group of authors relies on stochastic prediction of other TPs without identifying a specific maneuver. In comparison to the motion prediction, here the spatial distribution of the most probable areas to occupy is determined by stochastic means, instead of deciding on a maneuver. The underlying principles used are reviewed in the following.

Lambert et al. [203] overlay the ego-motion as well as the position or motion of obstacles with Probability Density Functions (PDFs) (Figure 2.10a). From the proportion of overlap of the own PDF with that of the object vehicles, a collision PDF is calculated. Thus not only the probability of a collision is determined but also its most probable location. Ward et al. [204] perform a Monte Carlo simulation on the covariance of a Kalman filter based motion prediction result, while Eidehall and Petersson [205] and Wang et al. [206] perform the simulation for all feasible primitives (Figure 2.10b).

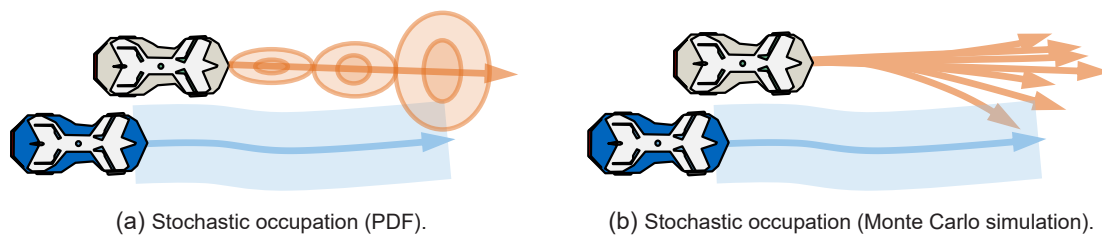


Figure 2.10: Illustrative sketch of underlying principles for two stochastic occupation based online risk assessment approaches with the ego-vehicle in blue and TP in gray.

Motion modelling: Besides the detection of a single maneuver and the stochastic mapping of the total set of most probable states, some approaches pursue the modeling of several maneuvers. In doing so, individual maneuver options are predicted and usually included in the risk assessment according to their likelihood of occurrence. In the following, the underlying methods for some approaches are discussed.

Althoff et al. [207] and Kim et al. [208] use Markov chains (Figure 2.11) to generate a stochastic reachable set, modeling the probability of future occupied states for other TPs. In order to result in a collision probability, ego-trajectory candidates are matched against the determined stochastic reachable sets. Laugier et al. [209] also work according to the same basic principle, but extend the model with hidden states. Here, a hidden Markov model is used to model possible maneuvers of the TP and derive concrete realizations with Gaussian processes. The collision risk is then determined according to the probabilities for all individual actions or based on the most probable maneuver (this would then count towards the first category of pure motion prediction).

Classification: Another group of authors propose algorithms that assign a certain risk class to every situation on the basis of defined characteristics or classifiers. In this process, the movement of the TP does not always necessarily have to be predicted.

Wardzinski [210] use a Constant Velocity (CV) model to predict the other vehicle's motion and then classify the situation based on the calculated minimal distance in dedicated risk classes. Tami et al. [211] and Bao et al. [212], on the other hand, build on dedicated

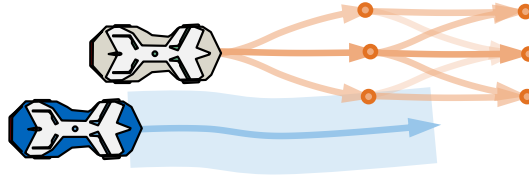


Figure 2.11: Illustrative sketch of underlying principles for the motion modeling (Markov chains) based approach with the ego-vehicle in blue and TP in gray.

classifiers that assign a risk class based on the current situation and trained behavior. Tami et al. use an Adaboost classifier, while Bao et al., Baur use a random forest classifier. Yang et al. [214], as in the previous category, model the behavior of the TP with a hidden Markov model. However, attention is paid to unusual behaviors. If a vehicle does not behave according to the model, an anomaly is assumed and the vehicle is put on alert accordingly. Three different levels of risk are differentiated.

Indicators: A widely used technique is the use of indicators to quantify risk. In particular, the simplicity of use, due to the absence of complex prediction algorithms, and low computation time stand out. In the following, a compact summary of relevant work is given.

One of the best-known indicators is the time-to-collision, which indicates the relative time a collision would occur based on the current velocity difference and bumper-to-bumper distance, given the assumption of maintaining a collision course and constant velocities. St-Aubin et al. [215] investigate this and other related indicators. Mattas et al. [216] and Aramrattana et al. [217] present new indicators that promise improvements over existing approaches. A more pragmatic and empirically driven approach is taken by Reschka et al. [218]. In their approach, threshold values for determined performance indications are used to infer an increased risk and to take appropriate behavioral measures. For example, the maximum velocity is reduced depending on the temperature in order to counteract the increased risk of reduced grip.

A first summary of the online risk assessment indicates that the focus is mainly on the probability of collision between a TP and the ego-vehicle. Therefore there is no focus on safeguarding. In this course a holistic approach, covering all aspects required for safe operation, is not intended either.

2.5.2 Online Verification

Besides the risk assessment approaches introduced in the preceding section, which try to predict the collision risk, there are also formal methods, which aim for a verification against specified requirements at runtime. The identified clusters of techniques and their properties are elaborated in the following.

Monitor concept: There are authors that suggest the use of an OV monitor in accordance with the standards and the objective of type approval. The sources are briefly referenced below.

Hörwick and Siedersberger [219] provide a high level description of safety observers without going into details. However, this concept demands the online monitor to be developed in accordance with the standards. Feth et al. [220] describe a concept for a “safety supervisor” for forward collision mitigation capable of safeguarding AI-based functions. Again, although

the concept is still mainly described in a superficial way, the conformity with the standards and the intention for approval are mentioned.

Modal logic: A larger number of approaches use modal logic to verify time sequences and processes against a given specification.

Linear Temporal Logic (LTL) [221–223] and Signal Temporal Logic (STL) [224] are one of the most prominent approaches to define and monitor temporal flows. Kane et al. [221] monitor with a LTL based monitor the validity of switching operations between automated and manual modes as well as heartbeats to detect dropped packages during transmission. Esterle et al. [222] present a semantic abstraction of traffic and applicable traffic rules with LTL and check online whether the maneuvers comply with the formalized regulations. For example, the approach was used to verify the traffic rule that prohibits overtaking via the right lane on the German Autobahn. Mao and Chen [223] specify desired characteristics of an adaptive cruise control system and formalize them using LTL. At runtime, it is then verified whether the system acts according to the specification. Watanabe et al. [224] describe the general application domain of STL and show an example using the logic to avoid collisions. Maierhofer et al. [225] use a further variant of the temporal modal logic, the so-called metric temporal logic. The authors formalize the traffic rules that apply on the German Autobahn and check their compliance at runtime. Xu et al. [226] use a custom defined spatial logic to verify behavioral actions like entering a roundabout. Verification logic has also been applied in the aviation domain [227].

Formal rules: Another group of authors mathematically formulate necessary boundary conditions, for example, distance to the vehicle in front, which has to be met to ensure safe operation (Figure 2.12a). At runtime, compliance with these boundary conditions is checked. In the following, studies in this category are listed.

Shalev-Shwartz et al. [20] address the approval aspect with a formal mathematical approach based on worst-case assumptions, called Responsibility-Sensitive Safety (RSS). In longitudinal and lateral direction, a required safe distance based on the current velocities, brake potentials, and reaction time is designed. The authors provide a proof based on mathematical induction. Chai et al. [228] investigate the application of RSS to online monitoring of an adaptive cruise control system. Orzechowski et al. [229] also apply the RSS framework and combine it with lane-based reachable sets to achieve safe behavior particularly at intersections. Grieser et al. [230] propose an online monitoring approach that guarantees safe operation for an end-to-end learning approach. For this purpose, boundary conditions to be met, such as stopping distance and track geometry, are correlated with the desired actions and accordingly permitted or denied. A similar method is used by Mirchevska et al. [231] to verify the generated maneuvers of a deep learning based planner.

Reachable sets: Reachable sets are used to determine all states that can be physically reached by a TP, i.e., taking into account all possible combinations and sequences of control variables such as acceleration, braking, and steering. Figure 2.12b sketches the sets of all reachable states of a TP at different successive time intervals with orange patches. Further details drawn from related work on this method are described in Subsection 5.3.2. As long as the driving tube of the planned trajectory of the ego-vehicle does not intersect one of the reachable sets of other TP, a collision with it is guaranteed to be excluded. For this reason, this method has a great importance in the field of safety guarantees and has been investigated and extended by a number of researchers.

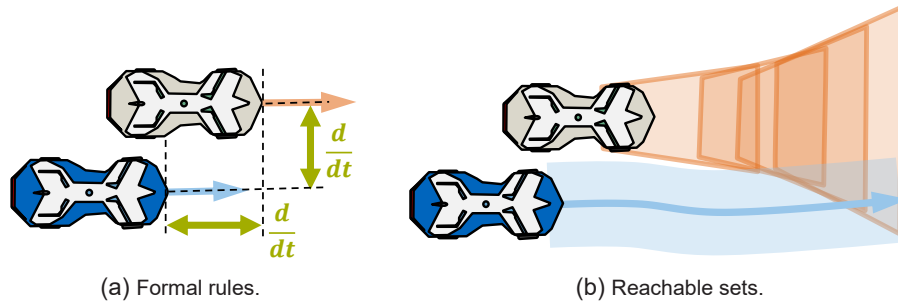


Figure 2.12: Illustrative sketch displaying the underlying principles of two OV approaches with the ego-vehicle in blue and TP in gray.

On the basis of Althoff [187], a number of approaches focus on the application of reachable sets [232–239]. With this method all reachable states of other TPs within a temporal interval are formally determined with an underlying dynamic model. The actions of the ego-vehicle falling in the same temporal domain are then verified against these sets. In addition to the dynamic collision analysis, some of the approaches also consider other aspects such as vehicle dynamics or adherence to track limits. Du et al. [240] investigate the application of reachable sets to pedestrians in order to make safety guarantees in road traffic involving pedestrians. The application thereby demonstrates the overall applicability of the approach. Also for safety guarantees regarding collisions with other TPs when employing RL planners, reachable states were used in the work of Krasowski et al. [241]. Kojchev et al. [242] pursued a slightly different approach but still with the same basic idea. Here, reverse reachable sets are used to guarantee safe behavior. Starting from a safe end state, a set of states is determined in reverse that have to be passed by the ego-vehicle trajectory in order to be considered safe.

Compared to the online risk assessment methods (Subsection 2.5.1), some approaches in the OV domain support type approval or consider applicable standards. However, none of the approaches covers both applicable standards and a detailed implementation to a desired extent. The following section details more on the open challenges and strengths of the presented approaches.

2.5.3 Overall Valuation and Classification

In this section, the findings of Subsection 2.5.1 and Subsection 2.5.2 are summarized, the further focus within these approaches is derived, and the missing aspects in the state of research are highlighted.

To give this objective more clarity, the relevant papers from related work are clustered and assessed in Table 2.1. In this context, four essential aspects are assessed. Two of them evaluate the applicability and preparation for approval. On the one hand, it is assessed whether the applicable standards were referred to and taken into account and, on the other hand, whether the central aspect of holistic coverage (not leaving selected aspects out) is addressed. As previously elaborated (Subsection 2.4.1), in ISO 26262 and ISO PAS 21448 the identification of all aspects that constitute safety or risks using structured methods is fundamental. Two further aspects assess the maturity of the work to illustrate whether it is a first concept or whether extensive testing has already been carried out. In this sense, the maturity of the method as well as the scope of testing is evaluated. In detail, the following four criteria (first column in Table 2.1) are graded:

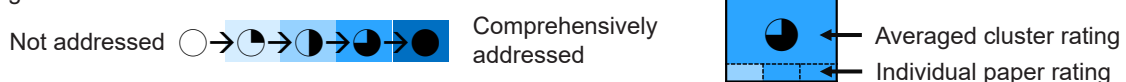
- Approval/standards: Rating on a scale from approval or applicable standards not being addressed in the publication (○) up to a full consideration and implementation (●).
- Holistic approach: Assessment on a scale from only a single safety aspect being considered (○) up to a comprehensive systematic coverage (●).
- Degree of maturity: Evaluation on a scale from rough concept (○) to detailed real-time capable implementation (●).
- Extent of testing: Assessment on a scale from theoretical consideration (○) to validation on real vehicles (●).

A detailed specification of the individual evaluation levels can be found in the Appendix in Table A.1. In the following paragraphs, the previously presented work is reviewed and briefly classified in written form in addition to the detailed evaluation in Table 2.1.

Table 2.1: Overview and evaluation of the addressed aspects in related work regarding online safety assessment of trajectories for ADSs. Related work is clustered into sections (top two rows) and rated against four metrics (first column). The resulting rating is indicated via Harvey Balls (○ – not addressed to ● – comprehensively addressed) along with the brightness of the color in the cell (from light to dark). Details on the rating metric can be found in Table A.1. Each cell with a Harvey Ball holds the rating for the corresponding metric and method cluster. The smaller (sub)cells below each Harvey ball cell hold the ratings of the individual papers via encoded colors only. The subcells are sorted from left to right according to the papers referenced in the table header and are thus also assigned to a single paper along a vertical line. The cluster rating is obtained by taking the rounded mean value of the individual ratings.

	Online risk assessment					Online verification			
	Maneuver prediction [197–202]	Stoch. occup. [203–206]	Motion modelling [207–209]	Classification [210–214]	Indicator [215–218]	Monitor concept [219, 220]	Modal logic [221–227]	Formal rules [228–231]	Reachable sets [232–242]
Approval / standards	○	○	○	○	○	●	○	○	●
Holistic approach	○	●	●	●	●	●	○	●	●
Degree of maturity	●	●	●	●	●	●	●	●	●
Extent of testing	●	●	●	●	●	●	●	●	●

Legend:



The risk assessment methods have their strengths in the area of collision assessment among different TPs. Numerous approaches have been established here. While all of the maneuver-prediction-based approaches are implemented thoroughly and evaluated at least in simulation, all of them only cover few aspects of safety (most of them even only one: a collision with other TPs) and do not tackle the aspect of type approval or applicable standards. As with the maneuver recognition approaches, the publications in the domain of stochastic occupation prediction provide a detailed implementation and evaluation, but solely focus on the dynamic collision aspect without the type approval in scope. Following this scheme, neither applicable

standards nor holistic coverage are in the scope of motion modeling approaches. Some of the classification-based approaches were primarily evaluated in simple simulation environments only. However, in terms of the overall evaluation, the lack of a holistic treatment and approval targeting is more striking. While the indicator-based approaches are for the most part fully implemented and are not prone to computational time problems, holistic coverage is clearly a problem, as only individual aspects are taken into account and correlations are lost. Also, none of the work intends to address applicable standards with this approach.

In summary, risk assessment approaches have not yet been used to support applicable standards. Furthermore, the stochastic nature of most approaches introduces difficulties for the approval of the methods itself (in line with the standards). For these reasons, risk assessment approaches are not suitable for the purpose of safeguarding complex and AI-based methods.

By contrast, both referenced monitor concepts in the cluster of online verification address the type approval issue, however, a detailed procedure description or implementation is missing. While the group of logic-based approaches is strong in their maturity, it mainly focuses on the verification of state transitions or traffic rules but lacks the holistic view. As a consequence, type approval is not in the scope of these approaches. Although the formal-rule-based approaches—especially the RSS approach by Shalev-Shwartz et al.—are structured to guarantee safety under certain aspects, the methods lack a holistic approach. Only individual aspects, selected without further explanation, are monitored. In this context, monitoring the collision risk with other TPs is among the most common. Furthermore, these methods usually do not include crucial aspects such as feasibility of the trajectories or the consideration of the track geometry, so that it is not always guaranteed that a feasible emergency maneuver is possible in every situation. Most of the approaches published so far in the domain of reachable sets are comprehensive and in some cases not only simulated but also examined in real experiments. However, neither a holistic coverage nor a structured approach to result in a holistic coverage is provided in any of the approaches. Furthermore, applicable standards in combination with this method are not in focus.

In summary, concepts in the area of OV have indicated that the suitability of these methods for safeguarding—even of complex and AI-based methods—is promising. However, no approach demonstrated type approval capabilities paired with implementation and evaluation. Furthermore, it is essential that all safety risks are taken into account and mitigated. In this sense, a holistic coverage of all safety-relevant features has to be systematically developed. This aspect has not yet been adequately addressed in related work.

Concluding, the focus in the remainder of this work is on an OV method. The author aims at covering both suitability for type approval and the evaluation of a comprehensive and holistic implementation. According to the overview Table 2.1, there are currently no studies that sufficiently address all aspects of interest at the same time. This is evident, in particular, from the fact that none of the works considered receives a high rating with regard to all four assessment criteria in a vertical line. This applies both to the rating of the averaged classes with the Harvey ball shown, as well as to the individual papers where the rating can be obtained by color coding. The aim of this work is to achieve a solid rating (◐ / ●) in all four aspects considered. A more specific derivation of the guiding theses and research questions based on a critical remark of related work is given in the following section.

3 Problem Description and Methodology

Based on the findings in related work (Chapter 2), open challenges and conclusions for the own method in this thesis are elaborated in Section 3.1 and Section 3.2.

3.1 Problem Description

In relation to the guiding theses (Chapter 1) and based on the findings gained from related work (Chapter 2), the research questions of this thesis are elaborated in the following. In each case, a brief critical remark on the state of science is given, followed by a formulation of a hypothesis with each one or multiple resulting research questions.

Critical remark: Complex and learning-based driving functions exceed the limits of existing safeguarding approaches and standards.

Classical safeguarding approaches rely on a mental comprehension of the used functional logic or SW in order to implement adequate safeguarding mechanisms. In addition, certification is achieved by passing defined tests. For complex or even AI-based algorithms, these methods reach their limits. If online learning is used, previously passed tests lose their validity as soon as a behavioral adaptation takes place. In a similar manner, safety risks arise when updates are frequently deployed [243], causing validation tests that were previously performed to lose their validity.

In order to address the critical remark and to take the first steps towards supporting the first guiding thesis,

T 1: Complex and frequently changing SW for driving functions of ADSs can be safeguarded, while taking into account applicable standards.

the following research question is to be answered:

Q 1: *Which safeguarding method is most promising to address the challenges of approval for complex and learning functions, while taking into account applicable standards?*

An initial suggestion in response to the first research question (Q 1) can already be given by the structured review of related work in the previous chapter. Here, an OV function is proposed in related work to overcome these shortcomings, and the author considers it to be the most effective option. However, a conclusive answer to this research question is only possible after a dedicated study including an actual implementation, which is missing in related work. This leads to the following critical remark and research questions.

Critical remark: A comprehensive and thoroughly tested concept for the online monitoring of a driving function has not yet been investigated.

In related work, OV concepts are referred to as a possible solution to the aforementioned problems, but detailed and holistic concepts are lacking. A part of the papers in related work are concepts only and claim that online monitoring in compliance with the standards is a promising approach, but do not provide any further details or studies. The remaining part of the papers in related work is detailed, but only addresses individual aspects of detail, without considering holistic coverage or compliance with applicable standards. The concrete consideration of the development steps for such an OV (in the area of trajectory planners) including all requirements for a holistic and homologation-capable evaluation is not available. This work aims to take a step in this direction with a comprehensive investigation to assess the feasibility and to identify future directives and challenges.

This critical remark is addressed in two stages, analogous to the guiding theses defined at the beginning. First, in order to support the second guiding thesis,

T 2: A procedure can be found for the development of an OV method for safeguarding ADS functions, taking into account applicable standards.

the following research question is to be answered:

Q 2: *What development steps does an OV method respecting applicable standards have to follow, in order to meet the requirements of a comprehensive safety assessment?*

Second, after a generic approach has been established and in order to support the third guiding thesis,

T 3: A prototypical implementation of an OV for the trajectory planner of a real-world automated race vehicle demonstrates the viability and serves as proof of concept.

the following research questions have to be addressed:

Q 3: *What are the specifics of a prototypical implementation of the derived generic OV method targeting the holistic safety assessment of a trajectory planner?*

Q 4: *What does the validation of a prototypical OV method involve in order to meet the purpose of a proof of concept for the overall method?*

The derived research questions are going to be answered in the course of this work and revisited in Section 7.2.

3.2 Methodology and Outline of the Thesis

The goal of this work is to address the problem formulated in the first research question (Q 1). According to the structured analysis of related work (Chapter 2), OV has been identified as promising. Since this hypothesis has not yet been comprehensively evaluated in related work, an OV will be conceived and implemented in the context of this work. The structure and layout of this thesis is depicted in Figure 3.1 and outlined in the following.

In Chapter 4, a generic development process for an OV framework is elaborated (Q 2). Since the OV module itself has to comply with the applicable standards, the procedure is aligned with the V-model used in ISO 26262. In this regard, the requirements for the OV module itself and

the criteria that describe a safe function are first identified. Then, the implementation phase of an OV with the necessary development steps is compiled. In accordance with the ascending branch of the V-model, integration into the overall system with corresponding integration tests follows. Finally, the correct functioning of the OV is certified in a validation phase. To enable a feasible implementation within this work, particular emphasis is placed on the challenges regarding the compliance with the standards, which have been elaborated in the related work section (Subsection 2.4.1). In this course, a detailed implementation of all measures contained in the standards is not feasible. Accordingly, reference is made to the passages that are expected to be implemented straightforwardly, and general applicability is pursued.

After the generic procedure has been formulated, a detailed implementation for an example system is to follow. Consequently, the rest of this work focuses on the OV of trajectories in a race vehicle in line with the third research question (Q 3).

In Chapter 5, the presented generic development steps are applied to a trajectory planning module (indicated by orange patches between the left and right column in Figure 3.1). In this course, the criteria that must hold for a trajectory to be considered safe are elaborated. These include but are not limited to no physical interactions with static or dynamic objects and consideration of the friction potential. Subsequently, the architecture of the OV module is elaborated. The OV is integrated in the ADS SW stack in a way that allows the system to safeguard the monitored driving function. Once the architecture is defined and thereby all available inputs and outputs are known, evaluation metrics monitoring the criteria for a safe trajectory are implemented. In addition to metrics that track compliance with track or acceleration limits, evaluating safety in the context of other dynamic objects is challenging. In order to evaluate the safety of the ego-trajectory, it must be placed in relation to the maneuvers of other TPs. If the plans of the other TPs are known, this analysis is straightforward. In reality, however, little is known about the intentions of the other agents. Accordingly, prediction measures are used. Since an informative detailed prediction (e.g. single trajectory estimation) excludes possible behaviors and a robust prediction (e.g. reachable sets) would lock a large portion of the driving space, a method incorporating applicable rules is proposed. With rule-based reachable sets, rule-compliant safety is guaranteed by robust estimation and incorporation of informative knowledge. Finally, a classification function for the classification of the trajectories into safe and unsafe is defined.

The implemented OV framework is examined in Chapter 6 (Q 4). First, the integration into the SW stack of an ADS is performed and completed by integration tests. The approach is validated in scenario-based and real-world tests. In scenario-based testing, a wide variety of faults and critical situations are deliberately injected. In this way, dangerous and expensive/destructive real tests are avoided and the behavior of the OV module can still be examined in these situations. Real-world tests, on the other hand, can provide long and realistic data during safe operation. Accordingly, it is investigated in real-world tests, whether the OV module causes false triggering that would limit the availability of the system. It should be noted that this chapter is intended as a result of the specified and implemented OV module. Unlike a classic “results” chapter, the procedures for testing and validation will also be described in this chapter before the evaluation results are presented.

The work concludes with a discussion, summary and future research directions in Chapter 7 and 8.

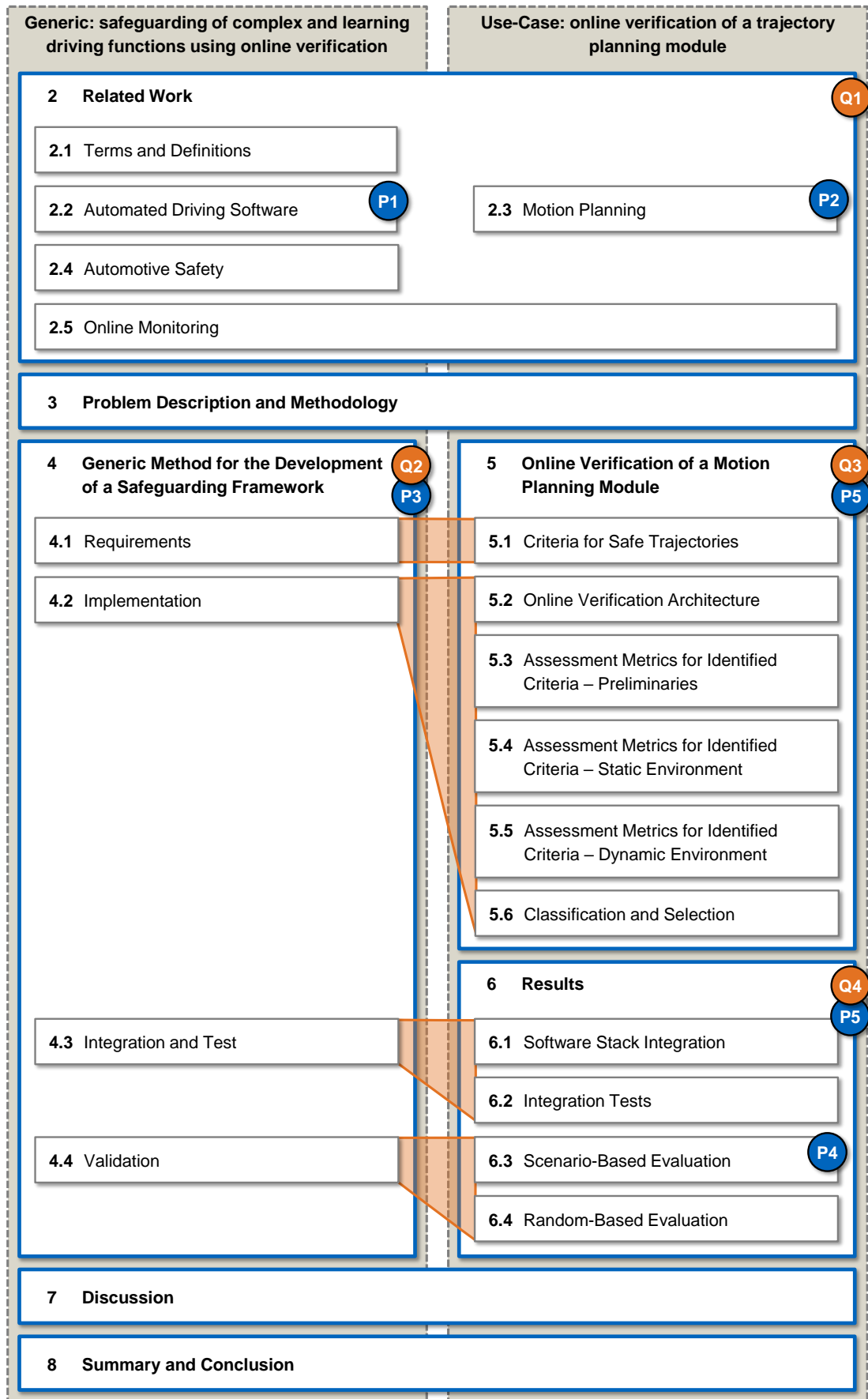


Figure 3.1: Structure of the thesis. Circles with “Q” indicate the coverage of the respective research question in the corresponding chapter, circles with “P” indicate an associated first author publication with the following mapping: P1=[32], P2=[71], P3=[244], P4=[245], P5=[246].

4 Generic Method for the Development of a Safeguarding Framework

In this chapter, the second research question is addressed:

Q 2: What development steps does an OV method respecting applicable standards have to follow, in order to meet the requirements of a comprehensive safety assessment?

The analysis in the related work section has shown that OV is promising for safeguarding complex and learning driving functions. The core idea is to verify the results generated by an algorithm that cannot be approved given applicable standards (e.g. AI) by means of a simple and approvable monitor. In particular, it is always checked whether the system can be transferred to a safe state after carrying out the action suggested by the algorithm. In this section, a generic method for such a scheme is elaborated, followed by a concrete implementation for a use-case in the next section.

In a previous work [244] the author of this thesis presented a concept for OV of driving functions: the “Supervisor”. This approach consists of a four-stage development process (Figure 4.1), which follows the procedure of the V-model and thus also the underlying principles of ISO 26262. Consistent with Figure 4.1 and previous work [244], this chapter describes the elaborated development process for an OV of driving functions. Each of the following sections deals with a separate stage (S-1 to S-4) of the procedure depicted in Figure 4.1. Each of the stages consists of up to three key steps (A, B, C), each tackled in a corresponding subsection in this chapter. In the course of this thesis, step y of stage x in the development process is referred to by the notation S- x - y .

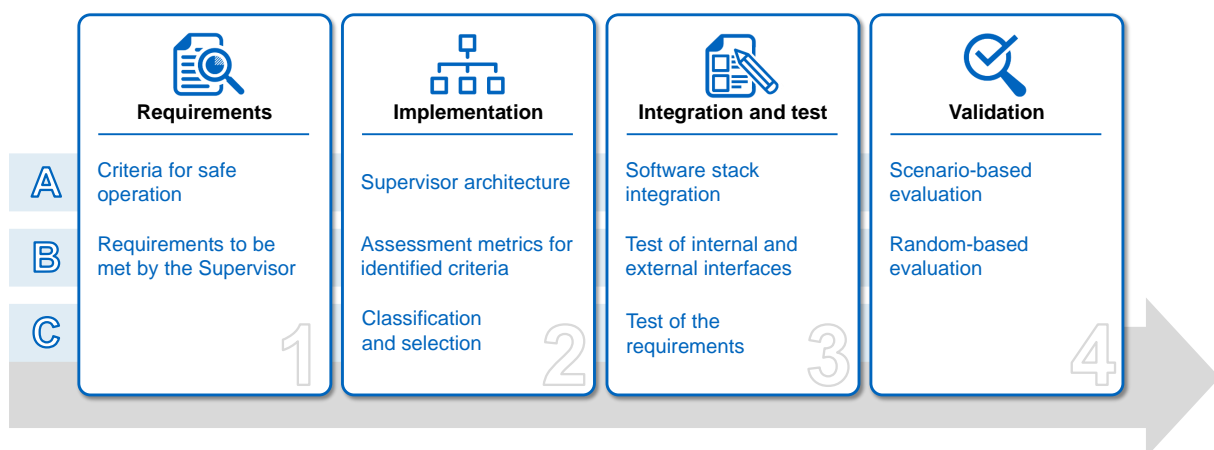


Figure 4.1: Generic method for the development of an OV module, the Supervisor (S), structured into four stages (1-4 – indexed from left to right), each comprising up to three steps (A, B, C – indexed from top to bottom), based on [244]. Each step can be indexed with a short notation, for example S-2-C refers to the “Classification function” step of the “Implementation” stage.

First, in S-1, the requirements are identified according to the first step of the V-model (Section 4.1). Here, both the requirements for a safe target function of the algorithm to be monitored and the requirements for the OV module itself are identified. Second, in S-2, the left branch of the V-model is traced downward (Section 4.2). Here, measures and architecture are derived from the requirements of the first step and implemented. Third, in S-2, the right-hand branch of the V-model is followed upwards again (Section 4.3). The OV module is integrated into the SW stack and tested against the requirements defined at the beginning. Finally, in S-4, the implemented OV module is validated (Section 4.4).

4.1 Requirements

At the beginning of the development of an OV module, in S-1, the requirements are identified and specified in a first step, in line with the V-model. On the one hand, the criteria for a safe operation of the driving function to be monitored (Subsection 4.1.1) and, on the other hand, the requirements for the OV module itself (Subsection 4.1.2) are developed. This procedure is carried out in accordance with the standards, in particular ISO 26262-3:2018 [23] and ISO 26262-6:2018, 6 [23].

4.1.1 Criteria for Safe Operation

The first step is to identify and define the criteria for a safe operation. This is done under the aspects of SOTIF and FuSa. According to SOTIF (ISO/PAS 21448-5:2019 [128]), it is first defined how the proper function of the system is described. In the context of this work, this is in particular the understanding of safety. Since the utopia of guaranteed safety can never be given [117, p. 74], it is important to determine exactly which objectives are being pursued.

Based on this specification, the criteria for a safe operation are derived. The particular challenge here is to identify a holistic list of all potential deficiencies [247]. It can certainly happen that individual features are overseen and later constitute a safety risk. Accordingly, it is advisable to proceed in accordance with a structured and systematic procedure. The procedure is closely related to the determination of safety requirements in ISO 26262-3:2018 [23]. Here, a structured Hazard Analysis and Risk Assessment (HARA) is proposed for the determination of safety objectives, from which one or multiple safety requirements are derived. In the case of an OV module, the criteria to be identified go along with safety requirements of the type “fault detection” in ISO 26262-3:2018, 7.4.2.3 [23].

If a HARA has already been carried out for the function to be monitored, the results can be directly derived and adopted. A HARA itself is a comprehensive and lengthy undertaking that usually requires several experts. For this reason, a reduced structured procedure for identifying the criteria for a trajectory planner are presented in Section 5.1, which does justice to the prototypical character of this proof-of-concept.

4.1.2 Requirements to Be Met by the Supervisor

The goal of the proposed online monitoring approach is to reduce the high ASIL requirements for the system of interest. High ASIL requirements would prohibit the use of non-transparent methods such as AI [136]. With the method of ASIL decomposition described in ISO 26262 [23], it is possible to provide the monitored system with lower requirements, if it is coupled to a

monitor with a high ASIL rating (Figure 4.2) [148]. If this is used to its full potential, a requested ASIL D rating can be split into a ASIL QM (D) and ASIL D (D) rating. Consequently, the AI-based or complex algorithm only has to meet basic constraints, while the comprehensive and hard constraints apply to the deterministic Supervisor.

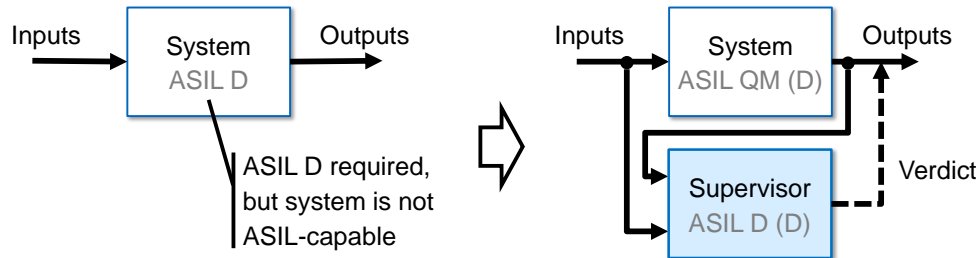


Figure 4.2: ASIL decomposition [23] can be used to lower the requirements for the system if another system with high requirements is monitoring it. This scheme is shown here for a combination of a system with ASIL QM (D) and the Supervisor with ASIL D (D), based on [244].

In the development of the Supervisor itself, the requirements imposed by the respective ASIL level as well as those resulting from the monitoring task are taken into account. ISO 26262-4:2018, 6 and ISO 26262-6:2018, 6 [23] state a number of recommended measures to be taken into account during SW development, graded according to the identified ASIL. These have to be followed in particular when implementing an OV in series production. For the prototypical realization in this work, a comprehensive implementation of all features is not feasible.

Hörwick [248] deals with the concept of a safety monitor for ADSs at a high level. This monitor concept complies with existing regulations, in particular ISO 26262. The work identified seven overall principles, which summarize the core properties that a safety monitor has to satisfy [248, p. 26]. Accordingly, these are considered as requirements in the development of the Supervisor. The principles (S-1-B-1 to S-1-B-7) are listed below:

- S-1-B-1 *Simplicity and predictability*: The lower the complexity, the less error-prone the system. In addition, the system's reactions become predictable and detectable, which enables systematic evaluation of system safety.
- S-1-B-2 *Complete fault detection*: The safety system must be able to detect all faults.
- S-1-B-3 *Guarantee of a safe state*: For each detected fault, an action plan must exist that can return the vehicle to a safe state.
- S-1-B-4 *Prevention*: The earlier faults are detected, the earlier countermeasures can be initiated.
- S-1-B-5 *Real-time capability*: To comply with the principle of prevention, the Supervisor must have real-time capability.
- S-1-B-6 *Separation from the overall system*: A clear separation between the components of the normal function and the components of the safety concept must be ensured. The interfaces between these two areas should be well defined and kept to a minimum (see S-1-B-1).
- S-1-B-7 *Modularity and adaptability*: The safety concept should have a modular character and be adaptable to special function specifics and system architectures.

It should be noted that these requirements apply generically to any OV method and, unlike safe operation (S-1-A), do not have to be derived again for each use-case.

4.2 Implementation

After the requirements for the Supervisor have been specified, the second stage (S-2) deals with the implementation of the Supervisor. First, the architecture is defined (Subsection 4.2.1), then the evaluation metrics are established (Subsection 4.2.2) and finally they are integrated into a classification and selection procedure (Subsection 4.2.3).

4.2.1 Supervisor Architecture

When determining the Supervisor architecture, the interfaces and signal flows for integration into the ADS SW stack are specified. The general architecture is based on the monitor principle shown in Figure 4.2. The definition of the starting point (inputs) and end point (outputs) for the system to be monitored is thereby essential. Based on the previously identified requirements for a Supervisor (Subsection 4.1.2), the following guidelines are drawn for the definition.

End point:

- As close as possible to the output of the system to be monitored: The complexity (i.e. also extent) of the SW being monitored should be kept as low as possible to allow for the simplest feasible verification (S-1-B-1).
- Bottleneck in the system (no parallel paths): For full safety control, it is essential that there are no parallel signal flows that may shortcut the verification decisions (S-1-B-3).

Start point:

- As close as possible to the input of the system being monitored: The span of the modules being monitored should be as small as possible to avoid Supervisor complexity and to maintain the ability to use established verification methods (S-1-B-1).
- Modularization: Approach-specific modules should be avoided as a starting point in order to obtain a generic OV module and thus facilitate the exchange or fundamental update of monitored approaches and/or system.

To illustrate these rules, a fictional SW diagram with possible start and end points is shown in Figure 4.3. The module shown in blue is to be covered by an OV. After applying the rules presented above, the relevant end point would be “E4”, since the nodes before do not represent a straightforward bottleneck and this is the first such node in the sequence. “S2” should be selected as the starting point, since this is closest to the module to be safeguarded, in order to keep the complexity low.

Provided that the start and end points of the Supervisor have been found, the rest of the architecture results from the requirements to be met by the Supervisor (S-1-B). The inputs that feed the Supervisor have to meet the specifications and requirements that have been derived for safe operation (S-1-A).

4.2.2 Assessment Metrics for Identified Criteria

After the architecture with the interfaces has been defined, evaluation metrics are established for the criteria of a safe operation (S-1-A). It is essential that each of the identified characteristics is

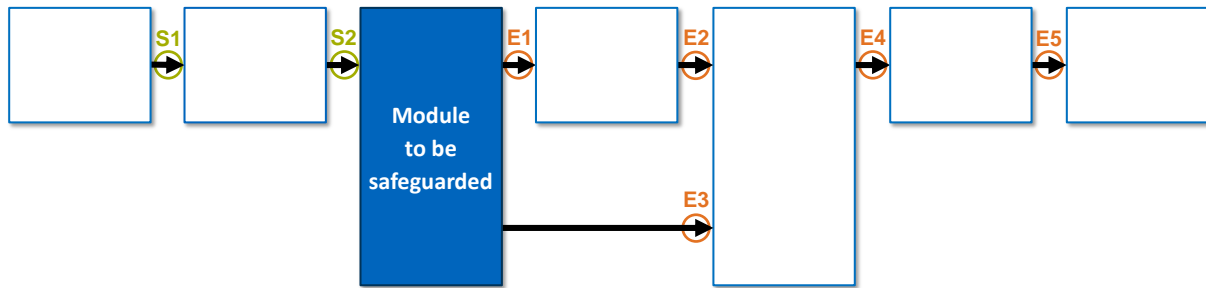


Figure 4.3: Fictional SW diagram with various modules including possible start and end points for an OV deployment.

covered by at least one metric. In addition, the metrics have to follow the other requirements (S-1-B and S-2-A). Expert knowledge and insight into the state of the art may be required to find and develop appropriate metrics. If it is not possible to comply with the elaborated requirements or to find an evaluation metric for individual criteria, the architecture (S-2-A) has to be revised in iterative manner. It is noted that this step is the most comprehensive and demanding in the implementation stage.

4.2.3 Classification and Selection

At the end of the implementation stage, an overarching classification function is developed that aggregates the individual evaluation metric scores. On the one hand, a binary evaluation (“safe” or “unsafe”) is generated for the given inputs, and, on the other hand, a mechanism that guarantees a safe state at all times (S-1-B-3) is developed. Here, especially the behavior in case of an “unsafe” rating is crucial. For this purpose, based on the evaluation and classification, a selection of a suitable measure to guarantee a safe condition is carried out.

4.3 Integration and Test

After the implementation phase is completed, the Supervisor is integrated into the SW stack (Subsection 4.3.1) and the functionality is verified. In the course of this, the interfaces (Subsection 4.3.2) as well as the compliance with the defined requirements (Subsection 4.3.3) are tested.

4.3.1 Software Stack Integration

During integration, the developed Supervisor is assembled according to the steps in the implementation stage (S-2) and integrated into the overall SW. Accordingly, the interfaces for the specified inputs are provided and the outputs with the associated safety mechanisms (guarantee of a safe state) are to be accommodated. A successful integration is then confirmed with adequate tests.

4.3.2 Test of Internal and External Interfaces

Following integration, the interfaces are first checked for consistent and correct implementation. This also involves checking the handling of incorrect data (error injection), missing data, and

timing variations. Further guidelines and recommendations for structured integration and testing can be found in ISO 26262-4:2018, 7 [23].

4.3.3 Tests of the Requirements

The most important integration test is testing against the requirements. The requirements defined in the first stage (S-1-A and S-1-B) are examined for their conformance. The requirements should be evaluated with a requirement-based test as well as with fault injection (ISO 26262-4:2018, 7 and ISO 26262-6:2018, 10 and 11 [23]). Both functional (e.g. detection of a certain behavior) and non-functional (e.g. real-time capability) requirements can be tested. All requirements should be verified with at least one corresponding test. Further details and requirements can be found in the ISO 26262-4:2018, 7 and ISO 26262-6:2018, clause 10 and 11 [23].

4.4 Validation

In order to mitigate design or implementation flaws that could compromise the safety goal, validation of the entire framework is required (in accordance with ISO 26262-4:2018, 8 [23]). A selection of validation approaches can be found in ISO 26262-4:2018, 8.4.3.4 [23]. According to the standard, an “appropriate set” of these methods should be applied. In the context of this work, the author considers scenario-based approaches (Subsection 4.4.1) as well as random-based real-world tests (Subsection 4.4.2) as the most promising in accordance with the state of the art (Subsection 2.4.2).

4.4.1 Scenario-Based Evaluation

The scenario-based approach emerges in related work as one of the most promising for the validation of ADS functions. The underlying idea is to evaluate critical and challenging scenarios instead of frequently occurring uninteresting situations. Furthermore, the real vehicle will rarely or never encounter safety-critical situations. If the scenarios are tested in simulation, it is recommended to evaluate destructive scenarios including collisions. It is primarily in these cases that the Supervisor is forced to become proactive and therefore provides the greatest informative value.

In this context, it is advisable to deliberately evaluate critical and unsafe scenarios by means of error injection. If a safe state exists at the beginning of the scenario, the Supervisor is expected to succeed in keeping the system in a safe state at all times. The tests therefore follow the principle of falsification, i.e., one wants to find a scenario where the Supervisor does not detect an unsafe state (false positive) and therefore fails to transfer the system into a safe state.

One of the more difficult questions is the extent of testing required to certify successful validation. Related work [167] deals with the question of completeness, but this has not yet been conclusively clarified.

4.4.2 Random-Based Evaluation

The random-based tests correspond to the field operation, in which the vehicle is tested in beforehand not further specified situations. This conforms to an evaluation over long periods in

a typical operational environment. Here, no-fire tests can be used in real operations, so that it is evaluated whether the Supervisor triggers, but no real intervention takes place.

The main goal of this approach is to evaluate false negatives (judging a situation as unsafe even though it is safe) during long deployments in a typical real-world environment. Under normal circumstances, the system to be monitored should be mostly in the safe area during real operations and should not require any intervention. This test therefore follows the principle of testing, i.e., one wants to show that the system does not generate any false negatives in regular use and that the availability is therefore high.

As with the scenario-based evaluation, one of the challenging questions is determining the necessary extent of testing. In this respect, ISO 26262 does not specify any requirements. Consequently, this measure is determined according to one's own expertise or the state of the art. For testing-based approaches, there are already publications that quantify the theoretically necessary testing effort [249, 16, pp. 1173-1177, 250, p. 457]. However, it is noted that this effort can quickly become infeasible from an economic point of view, especially for more complex systems.

Based on the generic steps for establishing an OV module elaborated in this chapter, the next chapter goes through the steps specifically for the use-case of a trajectory planner serving an automated race vehicle.

5 Online Verification of a Motion Planning Module

In this chapter, the third research question is addressed:

Q 3: What are the specifics of a prototypical implementation of the derived generic OV method targeting the holistic safety assessment of a trajectory planner?

The goal of this use-case is a prototypical implementation of the generic Supervisor concept (Chapter 4) as a proof of concept for an automated race vehicle. In particular, the motion planner of the race vehicle is assumed to be not ASIL-capable (ISO 26262-1:2018, 3.2 [23]) and considered in the course of this work. Although the planner can be considered complex in terms of the definition presented (Section 2.1), it is not an online learning AI. In order to examine this characteristic as well, additional simulative tests with error injection are employed in the results chapter (6). Furthermore, the Supervisor developed is intended to facilitate safe motion planning at all times, regardless of the planning approach used. The integration of the planner into the overall SW of an ADS as well as the basic functionality are discussed in Section 2.2 and 2.3. The method is applied and evaluated in the context of the *Roborace Season Alpha*. This is an autonomous racing series in which autonomous racing vehicles compete against each other on race tracks under given regulations [107].

Utilizing an automated race vehicle for the proof of concept offers advantages. The biggest benefit is the ability to test vehicles in its regular environment without putting humans at risk. The use of the race car usually takes place on *Fédération Internationale de l'Automobile* (FIA) approved race tracks where no humans are on the track. Furthermore, at high speeds, there are no safety drivers on board, which completely eliminates the risk of human injuries, something that is currently not possible in real road traffic. In addition, the vehicles consistently move at the dynamic limit of the vehicles. This area is particularly interesting from a safety point of view, but rarely occurs in regular road traffic (e.g. only during emergency maneuvers). Lastly, the ODD is limited, allowing for a reasonable prototype implementation. These and other advantages have been highlighted in a publication [251].

Besides the advantages, however, there are a few aspects to consider. The standards presented (Subsection 2.4.1) have been developed for road vehicles and are currently not applied to motorsports. For the proof of concept, the standards are nevertheless applied to race vehicles. The reason for this is that a large number of the aspects in the standard can be applied generically to any automotive system. The findings of the feasibility of an OV method can be achieved in the same way by the proof of concept in this domain and subsequently transferred to other domains.

In the following, the first two stages of the generic development concept presented in the preceding Chapter 4 are applied in detail to the trajectory planner of an automated race vehicle. First, the requirements are identified (S-1). Following S-1-A, the criteria of a safe trajectory are developed (Section 5.1). The requirements for the Supervisor are taken directly from the generic

concept S-1-B. Subsequently, the implementation stage (S-2) is addressed. After the architecture of the Supervisor has been defined (Section 5.2), the development of the assessment metrics is discussed in consecutive sections. Preliminaries are presented in Section 5.3, which provide the necessary expert knowledge from the state of science. Based on these preliminaries, the evaluation metrics for a static and a dynamic environment are presented in Section 5.4 and 5.5, respectively. Finally, the classification and selection of safe trajectories is discussed (Section 5.6).

The procedure goes hand in hand with the basics of a preceding publication [246] by the author of the thesis, yet offers further details in this work.

5.1 Criteria for Safe Trajectories

Before the criteria of a safe trajectory can be identified, the term “safety” is specified in the context of motion planning. According to ISO 26262-1:2018, 3.132 [23], safety is the absence of unreasonable risk. Based on this, the author follows the interpretation of Mobileye published in a paper by Shalev-Shwartz et al. [20, p. 2]. In the paper it is shown that a vehicle can be forced into an accident without any fault of its own and thus there can never be guaranteed absolute safety. A striking example is an ego-vehicle surrounded by four TPs on a multi-lane highway. As soon as one of the vehicles drives into the ego-vehicle, it cannot avoid the collision. Consequently, the aim of this work is to avoid accidents that are caused by the ego-vehicle’s actions given applicable regulations and rules.

Based on this understanding of safety, a holistic list of criteria that a safe trajectory has to fulfill is established. The challenge in the process of generating a holistic list is that features can be overlooked at ease. Therefore, it is crucial to use a structured process that makes missing out individual features less probable. In the standards (ISO 26262 [23] and ISO/PAS 21448 [128]), the HARA is regarded as a proven structured tool for establishing the corresponding requirements. However, the HARA has so far only been applied at great expense to simple systems such as an emergency braking assistant. The application of the HARA for a trajectory planner does not do justice to the prototypical scope and is not feasible within the given means. Accordingly, Stasinski and the author of this thesis developed a more straightforward structured method in previous work [244, 252], which is outlined below.

The concept developed is based on a structured analysis of the interfaces that a system has with the environment. For this purpose, the AV and its environment are modeled using a systems engineering approach [253, p. 265, 254, p. 4]. The subsystems represented in this approach are derived from related work. Bagschik et al. [255] developed an ontology that defines all relevant entities for an ADSs in a scene. The entities are represented in six layers: road (L1), traffic infrastructure (L2), temporal adaptations of L1 and L2 (L3), objects (L4), environment (L5) and digital information (L6). For the systems engineering approach, all levels except L3 are adopted, since this is already represented in the entities of L1 and L2. The objects of L4 are split into static objects and TPs to allow for more differentiation. These entities are put into relation with the ego-vehicle by defining interfaces between any possible combination of subsystems. Details about the internal function of the system are not required to be known. For an ADS, the system with corresponding interfaces displayed in Figure 5.1 is obtained.

The criteria of a safe trajectory are derived by enumerating every connection between the automated vehicle and all linked subsystems. Each interface and the automated vehicle itself is examined for possible safety risks in the event of irregularities. The following guiding question

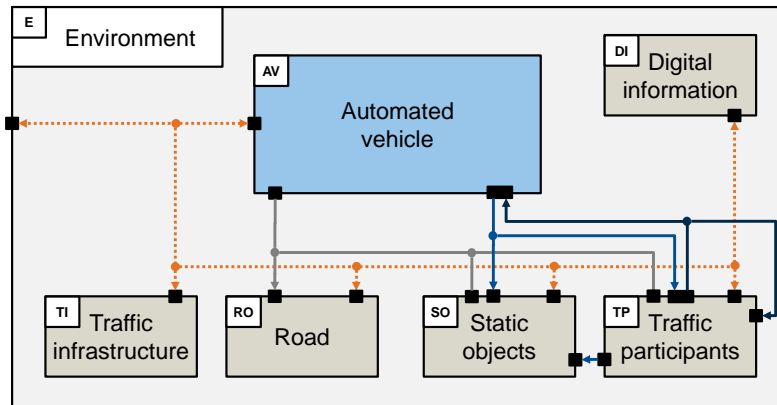


Figure 5.1: System model and its interfaces based on identified subsystems, based on [244]. Solid lines indicate a physical interaction and dashed lines an information flow. Directed arrows imply a physical interaction initiated by the subsystem at the tail.

can be used to help identifying criteria: “What kind of issue with this specific interface could pose a risk to safe operation in the context of the trajectory?”. For example, the interface [AV-RO] (connection between automated vehicle and road) may pose a safety risk if the trajectory does not consider the friction between road and tires. The identifiers used to reference a subsystem are given in Figure 5.1 in the upper left corner of the respective subsystem, e.g. RO for road. If the described process is continued for all interfaces in contact with the automated vehicle, the author obtained the core of the following list of criteria (S-1-A-1 to S-1-A-7) for a safe trajectory in a previous study [244]:

- S-1-A-1 *Accurate object list*: All objects in the scene must be recognized and perceived properly ([AV-SO], [AV-TP], [AV-TI], [AV-RO], [AV-DI], [AV-E]).
- S-1-A-2 *No physical interactions*: Physical interactions with static objects or other TPs initiated by the automated vehicle must not occur at any time ([AV-SO], [AV-TP]).
- S-1-A-3 *Correct origin*: The trajectory must match the actual pose in the real world ([AV-E]).
- S-1-A-4 *Feasible acceleration request*: The trajectory must respect the friction between road and tires at all times ([AV-RO]).
- S-1-A-5 *Obeyed rules*: Applicable rules of conduct (e.g., traffic or race rules) must be obeyed ([AV-TI], [AV-DI]).
- S-1-A-6 *Respected vehicle characteristics*: Kinematic and dynamic properties of the automated vehicle must be respected at all times ([AV]).
- S-1-A-7 *Intrinsic correctness*: The data of the trajectory must be correct (physically reasonable range) and linked (e.g., velocity and acceleration) ([AV]).

It should be noted here that criteria that were missed at this point will likely be exposed in the final validation step (S-4). In such a case, however, the entire development process is required to be conducted again in an iterative manner.

5.2 Online Verification Architecture

Applying the scheme presented in Subsection 4.2.1 for specification of the OV architecture, the endpoint of the module to be monitored is determined first. This should be chosen as close as possible to the output of the module to be monitored as well as to a bottleneck. In this case, these are the generated trajectories, which are sent to and traced by the controller. Then the start point is defined, which should be as close as possible to the input of the system to be monitored while representing a generic (not implementation-specific) data type. The module to be monitored is the motion planning module including the traffic prediction. A generic input to this module is the data provided by the perception module. A schematic visualization of the architecture based on the generic ADS pipeline (introduced in Figure 2.4) is shown in Figure 5.2.

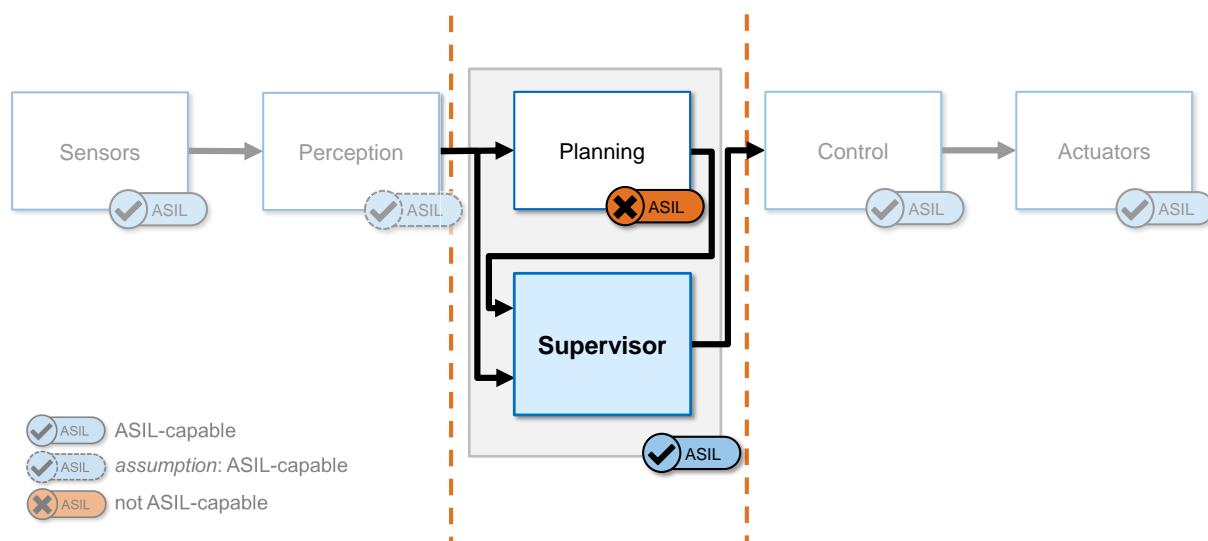


Figure 5.2: Supervisor architecture for the use-case of trajectory OV. The planning module is assumed to be not ASIL-capable (ISO 26262-1:2018, 3.2 [23]), but by integrating the Supervisor, the overall system can be approved. Unsafe behavior in the planning module, for example due to non-transparent or online learning methods, is detected and intercepted by the Supervisor. The orange dashed lines indicate the distinct interfaces with the rest of the SW stack.

The Supervisor verifies each trajectory against the inputs and guarantees provision of a safe trajectory to the controller. Since the Supervisor aims at always assuring safety (even in the case of an invalid trajectory and uncertain turn for the better in subsequent planning steps), the transition to a safe state has to be available at any time. To enable this, the emergency trajectory of the planning module in particular has to be verified. Accordingly, the planning module to be safeguarded has to provide a performance trajectory (regular operation) as well as an emergency trajectory (transfer to a safe state). Two separate planners are also plausible for this task, both feeding the Supervisor. The detailed logic for the switching behavior based on the safety rating of the two trajectories is defined in step S-2-C (Section 5.6).

The inputs and outputs of the planning module and thus also the Supervisor follow a specification. In this case, it is essential to define a specification as generic as possible for the object list as input and the trajectory as output. At this point it is also emphasized again that this work is limited to the safeguarding of the planning module. According to the illustration in Figure 5.2, the other modules are assumed to be approvable and safe. This has to be ensured by separate procedures or measures. For this reason, in this work ideal perception data (realizable for testing via Vehicle-to-Everything (V2X)/V2V communication) and an ideal controller (achievable by specifying a trajectory corridor, i.e. planning for a wider vehicle, and a controller that guarantees

path tracking within this corridor [58]) is assumed. The Supervisor's responsibility is to verify that the planned trajectory is valid, given the input data provided (regardless of its correctness).

In order to allow a valid correlation of the different input data, it is required that all input data provided to the Supervisor in one time step originate from the same planning step. As a consequence, the object list provided to the Supervisor is the one used as the planning basis for the trajectories to be examined. If this is not possible, a synchronization of the trajectories and the object list is required. The time offset is compensated for by physics-based prediction of the objects or by moving forward on the trajectory, depending on which of the two hosts more recent data. The time-stamp used for the synchronization has to originate from the same basis and the time-stamp of the trajectory corresponds to that of the objects used for the planning step.

The framework derived from the directives described above is depicted in Figure 5.3. The Supervisor is supplied with perception data (including environment map \mathcal{M} and object list $\mathcal{O}(t_0)$ for time-stamp t_0), as well as the trajectories (performance trajectory $\xi_{\text{perf}}([t_0, t_h])$ and emergency trajectory $\xi_{\text{em}}([t_0, t_h])$ for a time interval $[t_0, t_h]$, $t_0 < t_h$). If necessary, this data is synchronized. Subsequently, assessment metrics are evaluated for both trajectories according to the criteria defined in S-1-A. The evaluation metrics are implemented in a set of Supervisor Modules (SupMods) Ψ . Depending on the trajectory type, a (sub)set of SupMods is evaluated ($\Psi_{\text{perf}} \subset \Psi$ and $\Psi_{\text{em}} \subset \Psi$). Each SupMod $\psi_i \in \Psi$ returns a boolean safety rating $s_i = \psi_i(\mathcal{M}, \mathcal{O}(t_0), \xi([t_0, t_h]))$ based on the provided data. Subsequently, the ratings s_i are aggregated with a classification and selection method in order to pass a safe trajectory to the controller (Section 5.6). Details of the evaluation metrics employed in the SupMods are derived in the following three sections (5.3, 5.4, 5.5).

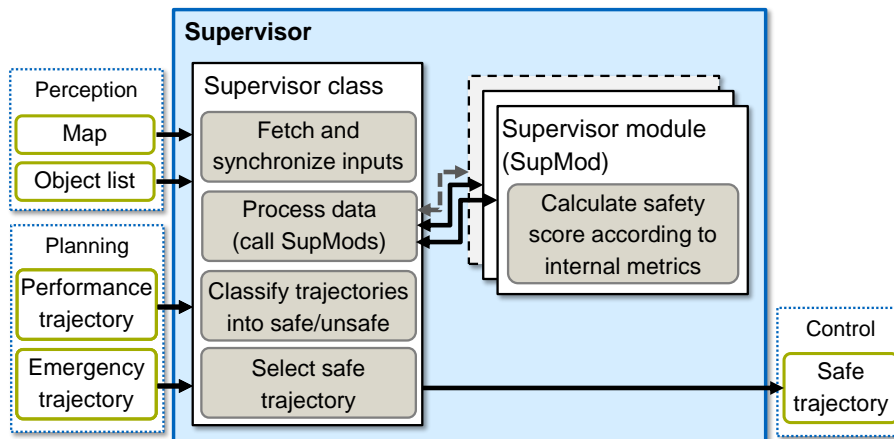


Figure 5.3: Proposed framework of the Supervisor, based on [246]. The input data (map, object list, performance trajectory, and emergency trajectory) is processed in dedicated SupMods. Based on the individual rating, a safe trajectory is selected and sent to the controller.

5.3 Assessment Metrics for Identified Criteria – Preliminaries

As discussed in Subsection 4.2.2, expert knowledge is needed for the implementation of the evaluation metrics for the requirements defined in S-1-A. After outlining the used general notation in Subsection 5.3.1, relevant basics from related work are briefly described. In related work of this thesis, reachable sets have already been considered promising and are introduced in

Subsection 5.3.2. In Subsection 5.3.3 the basics of past time temporal logic are presented. Finally, the identified criteria for safe operation (S-1-A) are analyzed and structured into problems with respect to a static and dynamic environment (Subsection 5.3.4).

5.3.1 General Notation

In line with Pek et al. [256], the state space $\mathcal{X} \subset \mathbb{R}^n$ is defined as the set of possible states ξ . For the ego-vehicle, $\mathcal{U} \subset \mathbb{R}^m$ denotes the set of admissible control inputs u . Given an input trajectory $u([t_0, t_h])$ for the time interval $[t_0, t_h]$, $t_0 < t_h$, the motion of the vehicle along the trajectory is described by the differential equation

$$\dot{\xi}(t) = f(\xi(t), u(t)). \quad (5.1)$$

The state $\xi(t)$ defined by the solution of Equation (5.1) at time $t \in [t_0, t_h]$ with initial state $\xi(t_0) = \xi_0$ and input trajectory $u([t_0, t_h])$ is expressed by $\chi(t, \xi(t_0), u([t_0, t_h]))$. In contrast to the input trajectory, which is usually generated by the controller, $\xi([t_0, t_h])$ describes the state trajectory, which usually represents the input to the controller / output of the planner.

The race tracks used for this use-case are defined by a left \mathcal{B}_l and right boundary \mathcal{B}_r , each represented by a polyline. Any point p on the track can either be referred to by a Cartesian coordinate $\langle x, y \rangle$ or within the Frenet-frame $\langle s, n \rangle$. The Frenet-frame is a lane-based coordinate system with s describing the arc length along a reference line of the track and n describing the lateral offset to the reference line (both measured with use of the normal vectors based on the reference line). The notation $\cdot(\xi)$ expresses the track coordinate for a given state ξ , for example $s(\xi)$ for the longitudinal coordinate. Furthermore, the lateral offset of the left and right boundary to the reference line at a coordinate s is denoted with $n_{\mathcal{B}_l}(s)$ and $n_{\mathcal{B}_r}(s)$, respectively.

The operator $\mathcal{E}(\xi) : \mathcal{X} \rightarrow \mathcal{P}(\mathbb{R}^2)$ defines the set of states \mathcal{X} occupied by a vehicle at a certain state ξ , where $\mathcal{P}(\mathbb{R}^2)$ denotes the power set of \mathbb{R}^2 . The occupation operator also holds for a set \mathcal{X} , following the specification $\mathcal{E}(\mathcal{X}) := \{\mathcal{E}(\xi) \mid \xi \in \mathcal{X}\}$.

Furthermore, it should be noted that in the context of this work, the integer 1 in equations is interpreted as a Boolean True and the integer 0 as a Boolean False.

5.3.2 Reachable Sets

Reachable sets have been applied by several authors in the automotive domain [232–239, 257–259]. Commonly, these sets are computed for consecutive time intervals and contain all reachable states of a considered agent in compliance with an underlying dynamic model.

Following the notation of Althoff et al. [257], within the set of states \mathcal{X} a vehicle can reach at any time, the reachable set \mathcal{R} is defined as a subset of this set, $\mathcal{R} \subseteq \mathcal{X}$. The states a vehicle can reach are deduced by following any possible action

$$u(t) = [u_1(t) \quad u_2(t)]^T, \forall t : u(t) \in \mathcal{U}, \quad (5.2)$$

where for an ADS, u_1 describes normalized steering and u_2 normalized acceleration. Thus, the exact reachable set \mathcal{R}^e is defined as the set of states that can be reached from an initial set of states \mathcal{R}_0 by executing any possible action $u(t)$:

$$\mathcal{R}^e(t_f, \xi_0, \mathcal{U}) := \{\chi(t_f, \xi_0, u(\cdot)) \mid \xi_0 \in \mathcal{R}_0, \forall t \in [0, t_f] : u(t) \in \mathcal{U}\}. \quad (5.3)$$

Since the exact reachable set \mathcal{R}^e can only be determined for simple dynamic models [257, 258], commonly a spatial over-approximation is used:

$$\mathcal{R}(t_f, \xi_0, \mathcal{U}) \supseteq \mathcal{R}^e(t_f, \xi_0, \mathcal{U}). \quad (5.4)$$

The points occupied by a vehicle at all reachable states are accessed with the previously defined occupation operator $\mathcal{E}(\mathcal{R}^e(t_f, \xi_0, \mathcal{U}))$. Althoff and Magdici [260] provide a computation-efficient over-approximation for the vehicle-specific occupation by the reachable set $\mathcal{E}(\mathcal{R}(t_f, \xi_0, \mathcal{U}))$, which is, for convenience, referenced as $\mathcal{R}^{\mathcal{E}}(t_f, \xi_0, \mathcal{U})$ in the course of this work.

The mentioned approach [260] is based on the pure progression of the vehicle pose using a CV model and applying a superimposed feasible uniform acceleration in all directions. This implies the assumption that the tire has the same traction in all directions and the engine can provide the corresponding acceleration. The set of all possible states of the vehicle center applying this principle is shown in Figure 5.4a. The over-approximation of the reachable set based on this principle is shown in Figure 5.4b. It should be noted that this reachable set covers the interval between two time steps.

In order to illustrate that this over-approximation is consistently on the safe side and to what extent it overestimates, over 400 vehicle trajectories were simulated using a simple vehicle dynamics model in Figure 5.4c for each time step in order to obtain an approximation of the real reachable set via sampling. In the process, the actual combined acceleration potential and kinematic constraints were taken into account. The greatest influence, however, is the consideration of the engine power. For the acceleration that can be applied via traction, 13.0 m s^{-2} was assumed in all directions for all models. In the more realistic simulation, the engine power (positive longitudinal acceleration) was limited to 6.0 m s^{-2} in accordance with the test vehicle. For better comparability, a superimposition of the methods explained is shown in Figure 5.4d. The over-approximation occupies significantly larger areas, especially with a longer time horizon, but can be calculated efficiently at runtime. Rack [261] and Radecker [262] investigated methods for slight but computation-efficient reductions of the over-approximated reachable sets. The simulation of the vehicle poses and intersection checks would, especially for several vehicles in the environment, require large computing capacities for a real time execution.

5.3.3 Past Time Linear Temporal Logic

For the formalization of temporal relations Past Time Linear Temporal Logic (ptLTL) introduced by Havelund and Roşu [263] is used in this work. The standard propositional operators like \neg (negation), \vee (disjunction), and \wedge (conjunction) are supplemented by further operators¹: \odot (previously) and \mathcal{S}_s (strong since). The semantics of ptLTL specific operators are given in the following. If $\theta = \sigma_1\sigma_2\dots\sigma_q$ is a finite sequence of abstract states σ_i , then θ_i denotes the trace $\sigma_1\sigma_2\dots\sigma_i$ for each $1 \leq i \leq q$. Given F , being a boolean value, variable, or an arbitrary number of atomic propositions, then the semantics of the introduced operators are:

$$\theta \models \odot F \iff \theta' \models F, \text{ with } \theta' = \begin{cases} \theta_{m-1}, & \text{if } m > 1 \\ \theta, & \text{if } m = 1 \end{cases}, \quad (5.5)$$

$$\theta \models F_1\mathcal{S}_s F_2 \iff (\exists j \in [1..m] \theta_j \models F_2) \wedge (\forall i \in [j..m] \theta_i \models F_1). \quad (5.6)$$

¹List limited to the operators used in this work, further operators in the ptLTL can be found in the referenced paper.

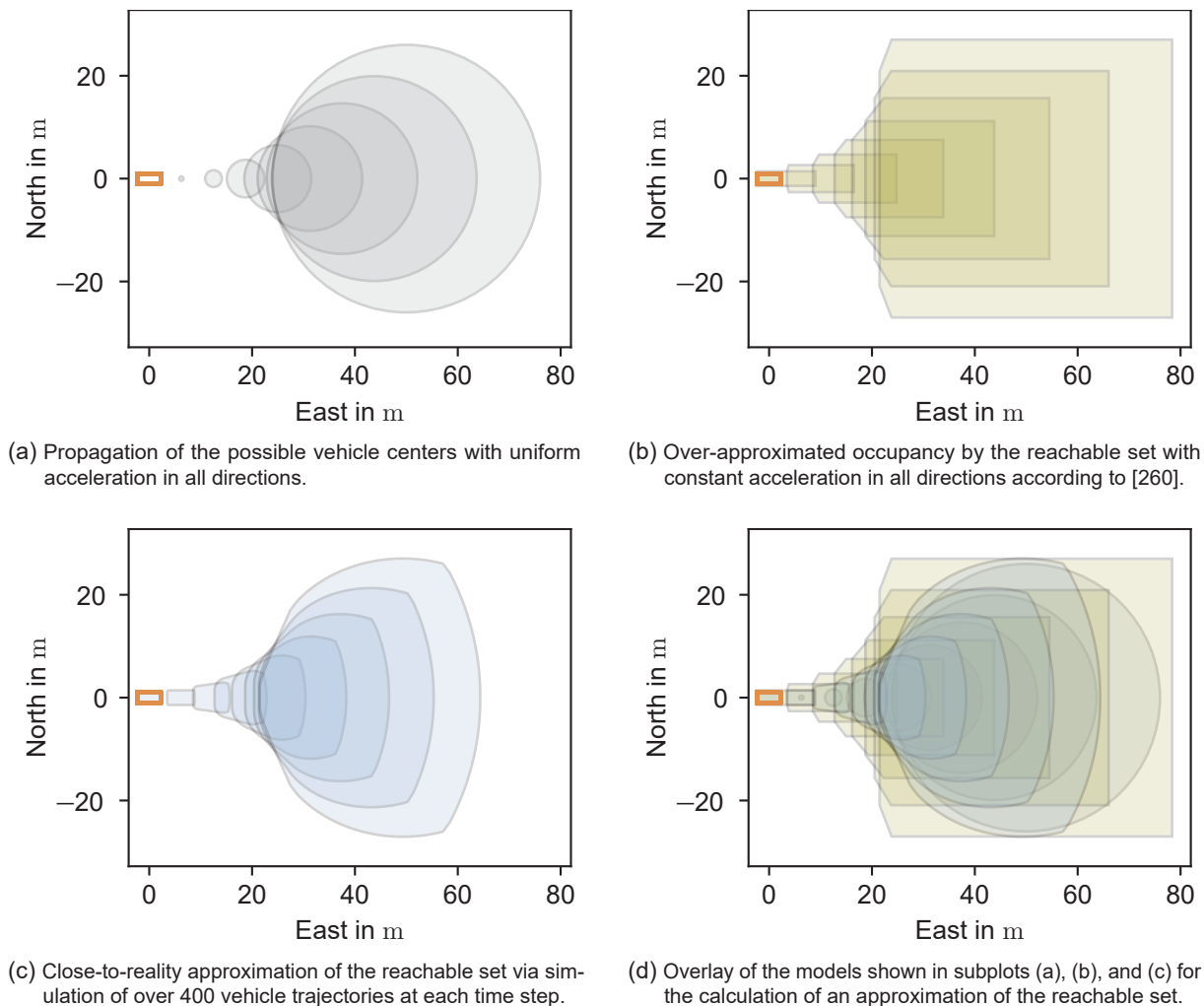


Figure 5.4: Different approaches for an approximation of the reachable set for a vehicle (orange rectangle) with initial velocity $v_0 = 25 \text{ m s}^{-1}$ and time horizon of $t = 2 \text{ s}$ at a separation of 0.25 s .

5.3.4 Preparation and Analysis of the Criteria for Safe Operation

As discussed in Section 5.1, each of the metrics developed there (S-1-A) is addressed. In the following, the previously identified criteria are listed and analyzed in terms of required measures. In the following two sections, these criteria and the respective metrics are addressed in terms of a static (Section 5.4) and dynamic (Section 5.5) environment. An assignment of the criteria to the corresponding sections is given in the following:

- S-1-A-1 (*Accurate object list*): No measures are required. Safeguarding targets the planning module, perception data is assumed to be approved and ideal (Section 5.2), in this case guaranteed by V2X communication.
- S-1-A-2 (*No physical interactions*): This requirement is split into two independent sub-problems. Collision with static, non-movable objects can be examined in a straightforward manner and is addressed in Subsection 5.4.1. The more complex problem of safeguarding with respect to dynamic objects is addressed in Section 5.5. Since in case of failing to find a new trajectory the emergency trajectory is traced terminally, it must be guaranteed that a safe end state is reached (Subsection 5.4.2).

- S-1-A-3 (*Correct origin*): No measures are required. Since ideal perception data (i.e. including localization data) are assumed (S-1-A-1), it is only necessary to check whether the current position relates to the trajectory origin for approaches that do not guarantee correspondence per se.
- S-1-A-4 (*Feasible acceleration request*): The maximum acceleration values based on the friction between track surface and tires must not be exceeded by the trajectory (Subsection 5.4.3).
- S-1-A-5 (*Obeyed rules*): Monitoring for compliance with traffic rules is covered in Subsection 5.4.4.
- S-1-A-6 (*Respected vehicle characteristics*): It must be verified that the trajectory takes into account the kinematic and dynamic limits of the vehicle (Subsection 5.4.5).
- S-1-A-7 (*Proper trajectory data*): The data of the trajectory must be checked for validity and whether individual physical quantities are linked properly (Subsection 5.4.6).

The assessment metrics referenced above are each implemented in a dedicated SupMod (as proposed in Figure 5.3). In the following sections, the individual assessment metrics are derived and specified.

5.4 Assessment Metrics for Identified Criteria – Static Environment

This section explains the evaluation metrics that can be evaluated in a static environment, i.e. the influence of other moving objects must not be taken into account. In the given order, collisions with static objects (Subsection 5.4.1), a safe end state (Subsection 5.4.2), acceleration limits (Subsection 5.4.3), kinematic and dynamic vehicle characteristics (Subsection 5.4.5), and applicable rules (Subsection 5.4.4) are addressed.

5.4.1 Collision with Static Obstacles

When checking against collisions with static, non-movable objects, in the use-case of motorsports, only the ego-vehicle staying within the track boundaries \mathcal{B}_l and \mathcal{B}_r at each point in time has to be verified. Given the assumption that each trajectory $\xi([t_0, t_h])$ hosts at least one point on the track, this property can be formally expressed as

$$\forall t \in [t_0, t_h], \forall \mathcal{B}_i \in \{\mathcal{B}_l, \mathcal{B}_r\} : \mathcal{E}(\xi(t)) \cap \mathcal{B}_i = \emptyset. \quad (5.7)$$

For analytically precise verification of this property, the vehicle footprint at each individual pose along the trajectory would have to be checked for overlap with one of the boundaries. Maierhofer et al. [225] follow a similar approach with the road-compliance checks of their drivability checker. However, this method is primarily designed for offline use and is computationally demanding. For a performant online use, a simplified approach that omits the orientation along the poses of the trajectory is implemented.

For the computationally optimized checks, the spline defined by the sequence of points along the trajectory $\xi([t_0, t_h])$ is inflated by a specified radius d_{infl} . The resulting inflation $\mathcal{I}(\mathcal{X}) : \mathcal{X} \rightarrow \mathbb{P}(\mathbb{R}^2)$

contains all points in the \mathbb{R}^2 domain that fall within the Euclidean distance d_{infl} to any point in the given set \mathcal{X} . A corresponding implementation of this method is provided by the “buffer” method of the Python library “Shapely”. If the radius is set equal to the vehicle width, an underestimation is obtained that only detects definite collisions. If an absolute safety guarantee is required, the radius is set equal to half the vehicle’s footprint diagonal. A comparison of these two approaches is shown in Figure 5.5. It can be seen that the underestimating variant only fails in detection on borderline cases. While the overestimating variant guarantees safety, it can lead to false alarms in the case of competitive driving behavior. In the context of this work, the underestimating variant was chosen for the race application, since the race line is commonly planned in the vicinity of the track boundaries.

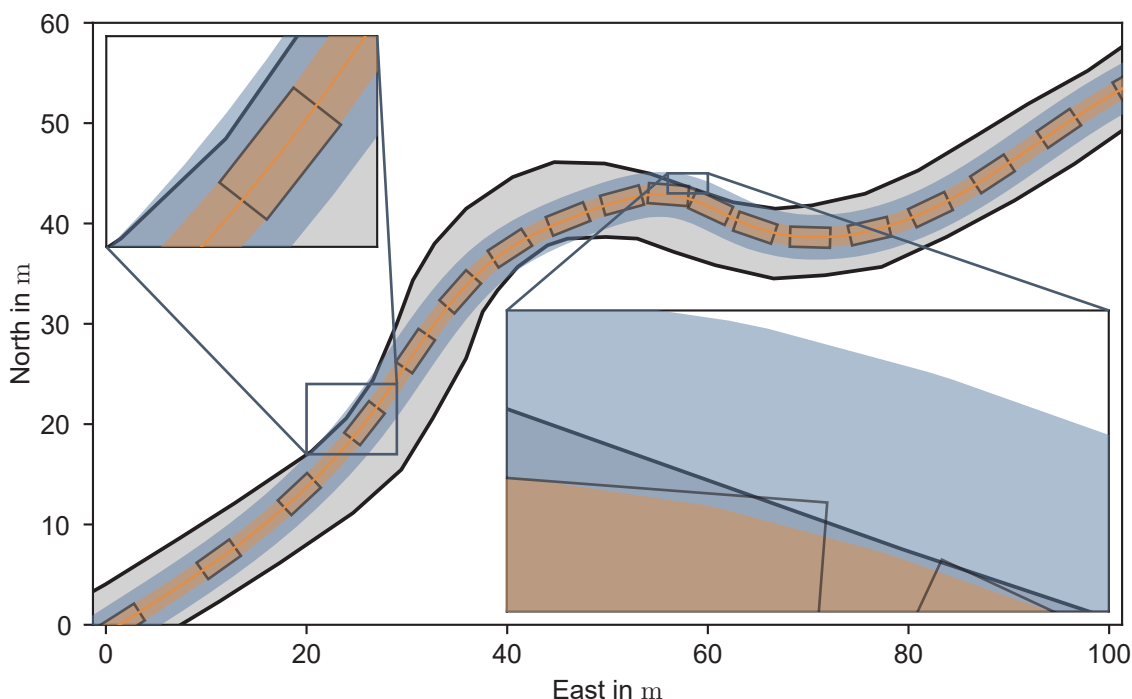


Figure 5.5: Illustration of different approximation methods for collision detection with the track (gray) boundaries (black). The trajectory of the vehicle is indicated by the orange line. Vehicle poses are depicted with a black frame at a spacing of 400 ms. A tube around the trajectory with the width of the vehicle footprint is shown in orange. A tube around the trajectory with the width of the vehicle footprint diagonal is shown in blue. A frame on the upper left shows a magnification of one false positive detection where the tube with the vehicle diagonal (blue) intersects the track boundary, while the real vehicle remains within the track boundary. The lower right frame displays a false negative detection, in which the tube with the vehicle width (orange) does not detect a boundary collision, while the vehicle slightly intersects the boundary.

The Boolean safety rating $\psi_{\text{stat}}(\cdot)$ resulting from the static collision checks is formally evaluated by applying the following principle:

$$\psi_{\text{stat}}(\mathcal{M}, \xi([t_0, t_h])) = \begin{cases} 1, & \text{if } \forall \mathcal{B}_i \in \{\mathcal{B}_l, \mathcal{B}_r\} : \mathcal{I}(\xi([t_0, t_h])) \cap \mathcal{B}_i = \emptyset \\ 0, & \text{otherwise} \end{cases}. \quad (5.8)$$

In order not to miss collisions in states between the mapped support points of a discrete trajectory, the trajectories are represented as continuous lines using the “Shapely” library. Consequently, the resulting inflation forms a tube that is checked for collisions with the boundaries.

At this point it is noted that a wider choice of tube can give more margin for the controller. In the proof-of-concept applied here, an ideal controller is assumed. With a tightly chosen inflation radius, there would be little to no room for control errors. However, if this distance is chosen large enough, there are approaches [58] that guarantee the vehicle to remain within a given trajectory-tube. Furthermore, it is noted that while only the track boundaries are considered here, individual (eventually movable) static objects can be modeled with the approach explained in Section 5.5.

5.4.2 Safe End State

In the event that no valid trajectories are found, the last valid emergency trajectory is executed. This is carried out until a new valid trajectory is generated. In the worst case it has to be assumed that the planner cannot find a valid solution anymore. For this reason, the emergency trajectory is required to be safe for an infinite time horizon. This is attestable by guaranteeing that the trajectory, among the other criteria of a safe trajectory, ends in a safe end state $\xi(t_h)$. In this use-case, standstill on the track is defined as sufficiently safe. An emergency trajectory $\xi_{em}([t_0, t_h])$ that satisfies the requirements regarding a sufficiently safe end state is formally expressed as

$$\forall t \geq t_h : v_x(\xi_{em}(t)) = 0 \iff v_x(\xi_{em}(t_h)) = 0, \quad (5.9)$$

where $v_x(\xi_{em}(t))$ denotes the associated longitudinal velocity of state $\xi_{em}(t)$.

The Boolean safety rating $\psi_{ses}(\cdot)$ w.r.t. to a safe end state generated by this SupMod is formalized as follows:

$$\psi_{ses}(\xi_{em}([t_0, t_h])) = \begin{cases} 1, & \text{if } v_x(\xi_{em}(t_h)) = 0 \\ 0, & \text{otherwise} \end{cases}. \quad (5.10)$$

Depending on the application, more complex requirements can also be imposed here. For example, on a highway it may be more appropriate to navigate to the emergency lane and stop there. Corresponding requirements can be formalized and employed in this SupMod. Related work deals with this topic in more detail [264, 265].

5.4.3 Friction and Tire Forces

The planned trajectory has to comply with the specified friction between track surface and tires at all times. The friction value, or the corresponding acceleration limits, are assumed to be given by the perception module. A convenient way to guarantee safety is to underestimate the actual friction potential. A trajectory $\xi([t_0, t_h])$ that accounts for the friction limits is formally stated as

$$\forall t \in [t_0, t_h] : F_a(\xi(t)) \leq F_{a,max}(\xi(t)), \quad (5.11)$$

where $F_a(\xi(t))$ denotes the combined force acting on the tires of a vehicle, resulting from the request by the trajectory and $F_{a,max}(\xi(t))$ the maximum possible force at state $\xi(t)$.

The combined force $F_a(\xi(t))$ that a vehicle's tires have to carry at a state $\xi(t)$ depends primarily on the lateral $a_{y,tire}(\xi(t))$ and longitudinal acceleration $a_{x,tire}(\xi(t))$ acting on the vehicle (Figure 5.6). The accelerations are defined by the planned velocity $v_x(\xi(t))$ at a given trajectory state $\xi(t)$, curvature $\kappa(\xi(t))$, and the requested longitudinal acceleration $a_x(\xi(t))$, respectively. All the listed quantities result implicitly from the sequence of states in the trajectory. It should be noted

that for longitudinal acceleration, drag resulting from air resistance must also be taken into account. The drag decelerates the vehicle quadratically to the speed and has to be additionally transferred by the tire if a positive acceleration is requested. When braking, correspondingly less force has to be exerted by the tire.

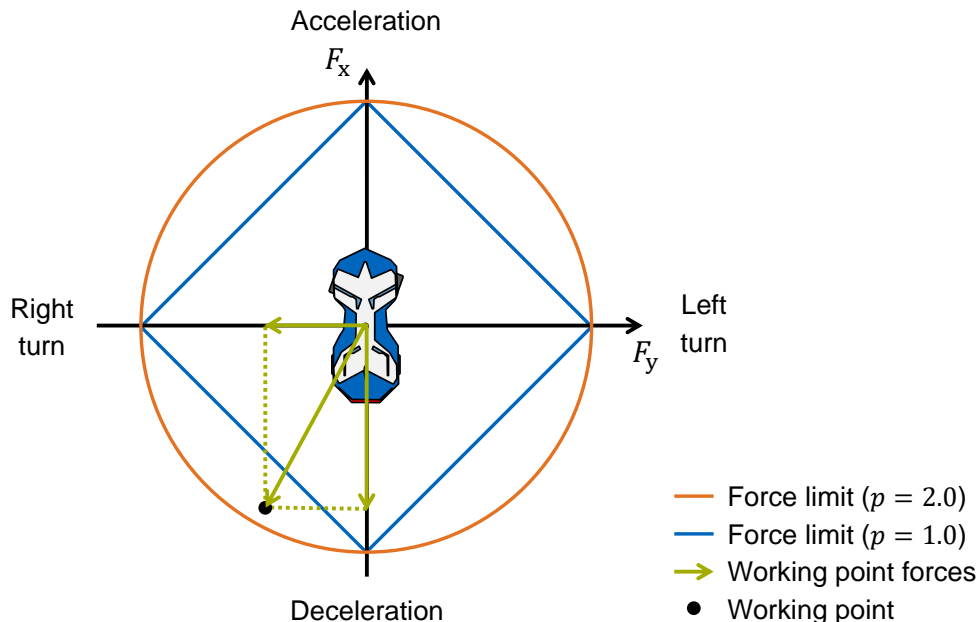


Figure 5.6: Visualization of simple models (orange, blue) that show the limits of the forces that a tire-road combination can withstand. The actual resultant force at an operating point (black dot) results from a lateral (centrifugal force) and a longitudinal (acceleration) component, shown here (green) as an example for deceleration in a right-hand turn.

The accelerations acting on the tire are determined as

$$a_{x,\text{tire}}(\xi(t)) = a_x(\xi(t)) + c_{d,\text{ext}} v_x(\xi(t))^2 \quad \text{and} \quad (5.12)$$

$$a_{y,\text{tire}}(\xi(t)) = v_x(\xi(t))^2 \kappa(\xi(t)), \quad (5.13)$$

with the extended drag coefficient $c_{d,\text{ext}} = \frac{c_d \rho A}{2m}$ representing a vehicle-specific value based on reference area A , mass density of the air ρ , and the pure drag coefficient c_d .

Assuming a planar and level track, the maximum allowed force $F_{a,\text{max}}(\xi(t))$ the tire can physically transmit is defined by the friction coefficient $\mu(\xi(t))$ and the normal force F_N :

$$F_{a,\text{max}}(\xi(t)) = \mu(\xi(t)) F_N(\xi(t)). \quad (5.14)$$

However, it is not trivial to determine the correct friction coefficient, as this depends not only on the interaction between tire and track surface, but also on other influences such as weather or temperature. Furthermore, the normal force is dependent on the speed, especially in the case of race cars with aero equipment. For this reason, the limits are often determined manually with test drives on the respective vehicle for a specific test track and weather. The acceleration limits for the tire-road combination are then be approximated with models spanning the entire working range. One of the most popular and straightforward models is the friction circle [266] (orange, Figure 5.6). However, test drives with the racing vehicle under consideration by a professional race driver have shown that the tire transmitted more force in purely lateral and longitudinal cases than in combined cases [267]. Accordingly, this observation is approximated with a diamond shape (blue, Figure 5.6). The assessment with regard to compliance with the limits is described

with the inequality

$$\left(\frac{|a_{x,\text{tire}}(\xi(t))|}{a_{x,\text{max}}(\xi(t))} \right)^p + \left(\frac{|a_{y,\text{tire}}(\xi(t))|}{a_{y,\text{max}}(\xi(t))} \right)^p \leq 1, \quad (5.15)$$

where the boundary between rhombus ($p = 1$) and ideal circle ($p = 2$) can be continuously adjusted with the parameter $p \in [1, 2]$ [268]. The parameter p and the maximum acceleration values $a_{x,\text{max}}(\xi(t))$ and $a_{y,\text{max}}(\xi(t))$ are determined on site after evaluation of (manual) test run data. It is noted that the maximum acceleration values may depend on the state $\xi(t)$. On the one hand, this is position-dependent due to the corresponding friction value $\mu(\xi(t))$ and, on the other hand, velocity-dependent due to the aero-influenced normal force $F_N(\xi(t))$.

An example application of this procedure is shown in Figure 5.7. The path of a trajectory for approaching a turn is shown in Figure 5.7a. The corresponding acceleration values in lateral and longitudinal direction at discrete time points along the trajectory are plotted with crosses in Figure 5.7b. An acceleration of 15m s^{-2} with $p = 2$ is assumed as the limit (orange circle). The crosses indicate that the vehicle is braking purely longitudinally at the beginning and that there is a gradual load change to purely lateral acceleration towards the apex of the turn. However, the points near the apex of the turn exceed the specified limit. Accordingly, this situation is to be evaluated as unsafe according to the given limits.

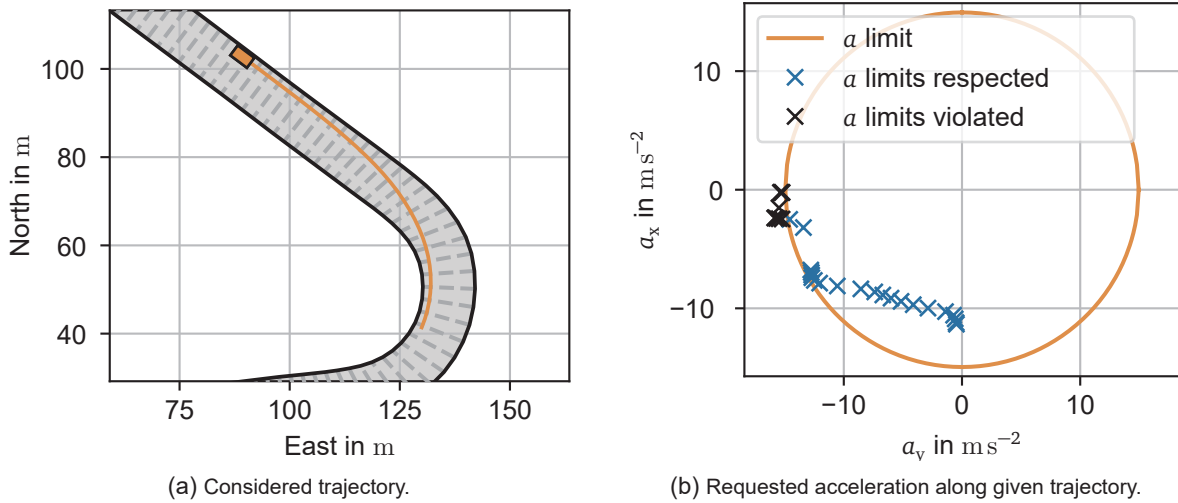


Figure 5.7: Exemplary trajectory with evaluated acceleration request for approaching a turn.

The overall Boolean safety rating $\psi_{\text{acc}}(\cdot)$ w.r.t. to the acceleration limits calculated by this SupMod is formalized as follows:

$$\psi_{\text{acc}}(\xi([t_0, t_h])) = \begin{cases} 1, & \text{if } \forall t \in [t_0, t_h] : \left(\frac{|a_{x,\text{tire}}(\xi(t))|}{a_{x,\text{max}}(\xi(t))} \right)^p + \left(\frac{|a_{y,\text{tire}}(\xi(t))|}{a_{y,\text{max}}(\xi(t))} \right)^p \leq 1 \\ 0, & \text{otherwise.} \end{cases} \quad (5.16)$$

While this approach guarantees that the planner generates a trajectory that satisfies the input constraints (acceleration limits in this case), it should be noted that for an overall safety guarantee, the actual friction value should be underestimated.

5.4.4 Rules and Regulations

In order to meet the specified safety goal of not causing an accident, the vehicle has to comply with the set of rules \mathcal{Z} in place, formally expressed as

$$\forall z_i \in \mathcal{Z} : z_i(\xi([t_0, t_h])), \quad (5.17)$$

where $z_i(\xi(t))$ holds for a rule z_i that is obeyed in state $\xi(t)$ of the trajectory.

Applicable rules are first formalized with temporal logic (e.g. ptLTL). At runtime, the trajectories to be verified are then checked against this specification. Since the formalization and rule verification is a separate field of research, this work focuses on a limited set of exemplary motorsport rules (Table 5.1), which serve the illustrative character. Further research and approaches in the field of rule formalization can be found in related work [222, 225, 269].

Table 5.1: List of exemplary rules $z_i(\xi(t)) \in \mathcal{Z}$ to be obeyed by the ego-vehicle.

Description	Mathematical formulation
<p>Maximum velocity In races, the race control can set a maximum velocity v_{\max} that may not be exceeded, either permanently or at specific times or locations (e.g. pit lane).</p>	$z_{v_{\max}}(\cdot) = \begin{cases} 1, & \text{if } \forall t \in [t_0, t_h] : v(\xi(t)) \leq v_{\max} \\ 0, & \text{otherwise} \end{cases}$
<p>No reversing In most races it is not allowed to reverse ($v(\xi(t)) < 0$). Especially in automated races, this could lead to fatal accidents, since the motion models of other vehicles may not reflect this.</p>	$z_{\text{rev}}(\cdot) = \begin{cases} 1, & \text{if } \forall t \in [t_0, t_h] : v(\xi(t)) \geq 0 \\ 0, & \text{otherwise} \end{cases}$
<p>Maximum deceleration The maximum deceleration $a_{\min} < 0$ can be fixed in advance (independently of the track and tires) to allow other vehicles to make appropriate assumptions.</p>	$z_{a_{\min}}(\cdot) = \begin{cases} 1, & \text{if } \forall t \in [t_0, t_h] : a(\xi(t)) \geq a_{\min} \\ 0, & \text{otherwise} \end{cases}$

It should be noted that only applicable regulations in a static environment (independent of other road users) are implemented here. These rules address risks caused by, for example, unexpected behavior (not represented in behavioral models) or excessive kinetic energies (breaking through safety barriers). Rules in a dynamic environment are not considered here, since the measures in Section 5.5 guarantee freedom from self-inflicted collisions under applicable regulations. This is sufficient for the safety objective to be supported. However, if it is still to be guaranteed that the planner does not violate any rules (rule violation while still behaving safely), then further rules regarding the dynamic environment have to be implemented.

The Boolean safety rating $\psi_{\text{rule}}(\cdot)$ w.r.t. to a set of rules to be monitored by this SupMod is stated as follows:

$$\psi_{\text{rule}}(\xi([t_0, t_h])) = \begin{cases} 1, & \text{if } \forall z \in \mathcal{Z} : z(\xi([t_0, t_h])) \\ 0, & \text{otherwise.} \end{cases} \quad (5.18)$$

5.4.5 Kinematic and Dynamic Properties

In order to enable the vehicle and thus also the controller to guide the vehicle along the trajectory, not only the friction-based acceleration limits have to be obeyed. Especially at slow speeds, kinematic or dynamic properties of the ego-vehicle can also be a limiting factor. The compliance

with a set of HW-related limitations \mathcal{L} along the trajectory $\xi([t_0, t_h])$ is formally stated as

$$\forall t \in [t_0, t_h], \forall l_i \in \mathcal{L} : l_i(\xi(t)), \quad (5.19)$$

where $l_i(\xi(t))$ holds for a limitation l_i that is obeyed in state $\xi(t)$ of the trajectory. In this work, the turn radius of the vehicle as well as the acceleration limits imposed by the power capabilities of the installed engine are considered in the set of limitations \mathcal{L} .

The reciprocal value of the requested turn radius, the curvature κ , is indirectly given in the trajectory. By tracing the discrete points of a trajectory $\xi([t_0, t_h])$, a curvature κ is given, which is required not to violate the turn radius $r_{\text{turn}} > 0$ at any time:

$$\forall t \in [t_0, t_h] : |\kappa(\xi(t))| \leq \frac{1}{r_{\text{turn}}}. \quad (5.20)$$

In addition to the acceleration limits imposed by the tire-road combination (Subsection 5.4.3), it is checked whether the vehicle's HW can provide these accelerations. In the case of negative accelerations, this is never a limiting factor due to safety-related over-dimensioning of the brakes. However, for positive accelerations, especially in the high-speed range, the motor is usually not able to provide an arbitrarily high acceleration. Therefore, it is checked whether the requested acceleration complies with the velocity-dependent acceleration limits $a_{x,\text{lim}}(v)$ at all times:

$$\forall t \in [t_0, t_h] : a(\xi(t)) \leq a_{x,\text{lim}}(v(\xi(t))). \quad (5.21)$$

It should be noted that a violation of this factor is not always safety critical, the vehicle would merely be able to accelerate less than requested by the trajectory. However, since there may be situations in which a collision can only be prevented if the vehicle actually accelerates as specified in the trajectory, compliance with these limits has to be checked.

The Boolean safety rating $\psi_{\text{kd}}(\cdot)$ w.r.t. to stated limits generated by this SupMod is formalized as follows:

$$\psi_{\text{kd}}(\xi([t_0, t_h])) = \begin{cases} 1, & \text{if } \forall t \in [t_0, t_h] : \kappa(\xi(t)) \leq \frac{1}{r_{\text{turn}}} \wedge a(\xi(t)) \leq a_{x,\text{lim}}(v(\xi(t))) \\ 0, & \text{otherwise.} \end{cases} \quad (5.22)$$

5.4.6 Data Correctness

To ensure that the tests performed by the SupMods on the trajectory are based on a reliable basis, the physical correctness of the data in the trajectory is examined. For this purpose, a set of plausibility checks \mathcal{P} are performed on the trajectory. A trajectory $\xi([t_0, t_h])$ that satisfies all requirements at each point in time t is formally stated as:

$$\forall t \in [t_0, t_h], \forall p_i \in \mathcal{P} : p_i(\xi(t)), \quad (5.23)$$

where $p_i(\xi(t))$ holds for a plausibility check p_i that is satisfied in state $\xi(t)$ of the trajectory.

In this work it is assumed that the discrete representation of a trajectory contains the quantities s -coordinate $s(\xi(t))$, x -coordinate $x(\xi(t))$, y -coordinate $y(\xi(t))$, heading $\psi(\xi(t))$, curvature $\kappa(\xi(t))$, velocity $v(\xi(t))$, and acceleration $a(\xi(t))$ for each discrete state $\xi(t)$ of the trajectory $\xi([t_0, t_h])$. If a different realization is provided, the requirements have to be adapted accordingly.

For each trajectory, the following plausibility checks p are performed:

- Check of the data format (e.g. the correct number of entities must be provided).
- Check of all individual physical quantities for plausible value ranges (e.g. since no turns smaller than 1 m are expected, the curvature must remain in the range $[0 \text{ m}^{-1}, 1 \text{ m}^{-1}]$).
- Correlation of s -coordinate to x - and y -coordinate (the distance between individual coordinates must correspond approximately to the distance between the corresponding s -coordinates, while accounting for slight deviation due to radian vs. euclidean distance).
- Correlation of heading to x - and y -coordinate (the orientation imposed by successive states must be close to the specified heading).
- Correlation of heading and curvature (the change of heading over distance must be close to the curvature).
- Correlation of velocity, s -coordinate, and acceleration (the acceleration must be close to the physical correlation of start and end velocity between two states and their distance).

When defining the checks, it is essential that each physical quantity listed in the trajectory is correlated with other quantities by at least one check.

The Boolean safety rating $\psi_{\text{dc}}(\cdot)$ w.r.t. to stated plausibility checks generated by this SupMod is stated as follows:

$$\psi_{\text{dc}}(\xi([t_0, t_h])) = \begin{cases} 1, & \text{if } \forall t \in [t_0, t_h], \forall p_i \in \mathcal{P} : p_i(\xi(t)) \\ 0, & \text{otherwise.} \end{cases} \quad (5.24)$$

5.5 Assessment Metrics for Identified Criteria – Dynamic Environment

The assessment of the dynamic environment is one of the biggest challenges in the OV of trajectories. The reason for this is the interaction with independently acting agents, whose intentions and detailed behavioral plans are not known. For trajectory planning, TPs are often propagated into the future with an informative prediction to lay out their own plan. However, in order to make safety guarantees independent of the actual actions of other TPs, robust prediction methods have to be used. The reachable sets (Subsection 5.3.2) represent such an approach.

An emergency trajectory has to be valid regardless of the action another TP takes. In this context, even worst-case maneuvers have to be assumed, comparable to a kind of rampage driver trying to hit the ego-vehicle. Accordingly, the emergency trajectories of the ego-vehicle are checked against the reachable sets of other TPs. The challenge in this case is that reachable sets tend to expand fast and would make it impossible to overtake or similar. To deal with this, rule-based reachable sets are introduced in Subsection 5.5.1.

For the performance trajectory, these hard requirements do not have to be met. To detect a collision as early as possible, trajectories that are guaranteed to result in a collision are marked as unsafe. This assumption corresponds to a fully cooperative TP. For this purpose, guaranteed occupied sets are presented in Subsection 5.5.2.

5.5.1 Rule-Based Reachable Set

As described before, the emergency trajectory $\xi_{em}([t_0, t_h])$ must not collide with any object o in the object list $\mathcal{O}(t)$. Formally, this is formulated outright with known object states $\xi_o(t)$:

$$\forall t \in [t_0, t_h], \forall o \in \mathcal{O}(t) : \mathcal{E}(\xi_{em}(t)) \cap \mathcal{E}(\xi_o(t)) = \emptyset. \quad (5.25)$$

However, since the future states ($\xi_o(t), t > t_0$) of the other objects o are usually unknown, every feasible maneuver must be assumed. Reachable sets model corresponding properties. The problem, however, is the fast growth of the occupied space, especially in the case of race vehicles with high acceleration capabilities and high speeds (green patches in Figure 5.8). Without further measures, vehicles cannot get close or even overtake each other without the reachable sets intersecting the planned (emergency) trajectory. To address this problem, the author presented rule-based reachable sets in a preceding work [246]. Similar approaches have been outlined for road traffic [237]. The goal is to restrict the large volume of physically feasible states while strengthening the superior safety goal (avoidance of accidents caused by the ego-vehicle).

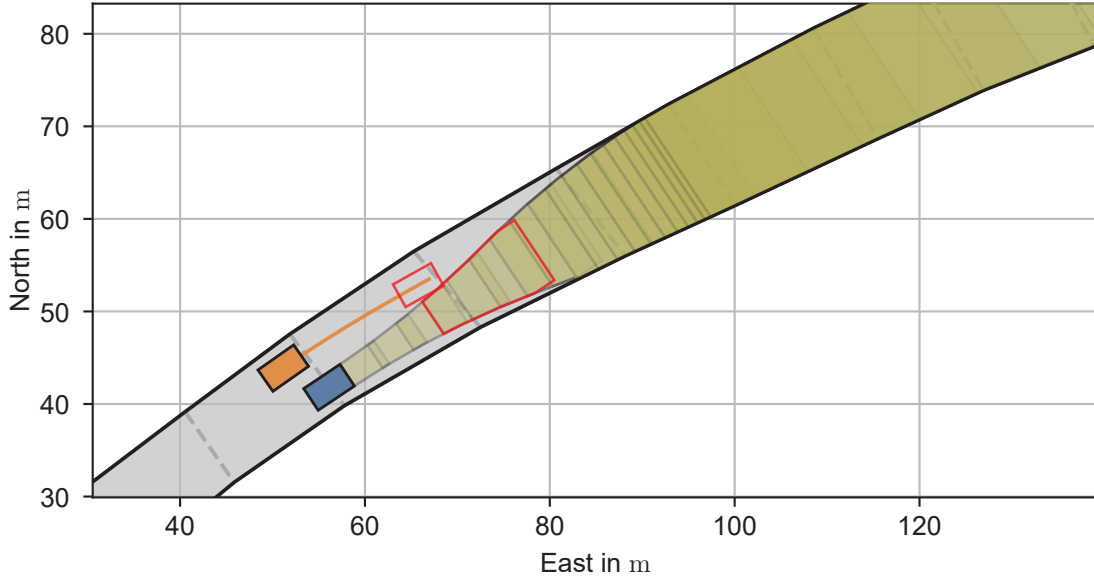


Figure 5.8: Reachable sets without rule restrictions would not allow overtaking each other, as the sets soon occupy the entire width of the track. The reachable set (green) of the TP (blue) intersects the poses along the emergency trajectory (orange line) of the ego-vehicle (orange) at a certain point in time (red).

To implement this behavior, the occupancy of the reachable set $\mathcal{R}^\mathcal{E}$ is further reduced by applicable rules \mathcal{K} (in this case motorsport rules). In particular, rules that influence TPs play a role here. Rules that only influence the ego-vehicle are covered in Subsection 5.4.4. Each formalized rule $k \in \mathcal{K}$ consists of two parts: a ptLTL triggering condition c_k and a mathematical description of the reduction set $Q_k \in P(\mathbb{R}^2)$. The occupancy by the rule-based reachable set $\mathcal{R}_\mathcal{K}^\mathcal{E}$ is formally described as

$$\mathcal{R}_\mathcal{K}^\mathcal{E}(t_h, \xi_0, \mathcal{U}) = \mathcal{R}^\mathcal{E}(t_h, \xi_0, \mathcal{U}) \setminus \bigcup_{k \in \mathcal{K}_c} Q_k, \quad \text{with } \mathcal{K}_c = \{k \in \mathcal{K} \mid c_k\}. \quad (5.26)$$

For clarity, this formula describes the relationship for the entire time horizon $[t_0, t_h]$ of the trajectory and the reachable set. In practice, this set difference is carried out for each time interval, enclosed in $[t_0, t_h]$. For computational reasons, the rules are evaluated for a given

state and propagated in time. A detailed implementation would require the evaluation along the evolution of the reachable sets.

For each rule, a triggering condition c_k is implemented that evaluates whether a rule is active based on the interaction of the ego-vehicle with the surrounding vehicles. To evaluate the temporal correlations in this context, ptLTL is used. Discrete time steps are used, where each entity holding time t as argument $\cdot(t)$ can be interpreted as a discretized series of states. The time series begins with the launch of the vehicle up to the current time. For each active rule the corresponding deletion patch \mathcal{Q}_k is determined according to its mathematical definition. For the sake of convenience, this patch is defined in the lane-based coordinate system.

In this thesis the implementation of rules originating the *Roborace Season Alpha* series as well as the *Formula 1* series [270, 271] are analyzed. The implemented rules are listed in Table 5.2. A verbal description, the trigger condition c_k , and the mathematical description of the reduction set \mathcal{Q}_k are given for each rule. These illustrative examples are used to demonstrate the implementation and evaluation of the rules. While these rules cover most situations, there are other rules that were not implemented in the prototype of this work. However, it should be noted at this point that the omission of rules does not lead to unsafe behavior, since the reachable sets without rule influence tend to cover a larger area, leading to more defensive behavior of the ego-vehicle. Figure 5.9 illustrates an exemplary situation in which the effects of a respective rule are demonstrated by showing an overtake maneuver with and without rule. The top row in Figure 5.9 depicts the situations with pure reachable sets without rule application. In these cases overtaking would not be possible. The bottom row in Figure 5.9 hosts the same scenario but with consideration of applicable rules. The sequence is similar until the ego-vehicle gets close to the other vehicle and the “*Formula 1* – racing alongside another car” regulation takes effect (Figure 5.9f).

The overall Boolean safety rating $\psi_{\text{reach}}(\cdot)$ w.r.t. to a guarantee of not causing a collision generated by this SupMod is stated as:

$$\psi_{\text{reach}}(\mathcal{M}, \mathcal{O}(t_0), \xi([t_0, t_h])) = \begin{cases} 1, & \text{if } \forall t \in [t_0, t_h], \forall o \in \mathcal{O}(t_0) : \\ & \mathcal{E}(\xi(t)) \cap \mathcal{R}_{\mathcal{K},o}^{\mathcal{E}}(t, \xi_0, \mathcal{U}) = \emptyset. \\ 0, & \text{otherwise} \end{cases} \quad (5.27)$$

Here, $\mathcal{R}_{\mathcal{K},o}^{\mathcal{E}}(t, \xi_0, \mathcal{U})$ represents the concise notation for the prospective occupancy of object o by the rule-based reachable sets at a time t . For a reachable set discretized in time into successive intervals within $[t_0, t_h]$, $\mathcal{R}_{\mathcal{K},o}^{\mathcal{E}}(t, \xi_0, \mathcal{U})$ is understood as the selection of the interval of the reachable set accommodated for the given time t . The time horizon of the reachable set can be chosen larger than that of the ego-vehicle’s emergency trajectory, but should not be smaller. The extent of discretization is a trade off between computational cost and false negative rate. The coarser the temporal discretization, the more false negatives (rating a safe situation as unsafe) will occur. Static objects are represented by a single (non-variable) set.

While race vehicles are considered in this work, it should be noted that an extension to road traffic, especially for highways due to the similar nature, is possible. This would require formalizing and integrating the applicable traffic rules. For rules that are difficult to formalize, for example, due to their imprecise or soft specifics, past court decisions can be included [225].

Table 5.2: Implemented illustrative trigger conditions c_k and associated reduction sets \mathcal{Q}_k . To obtain rule-based reachable sets, the reduction set is subtracted from the unconstrained reachable set for all objects o for which the trigger condition holds. For clarity, mathematically Simplified Representations (SR) are marked.

Description	Mathematical formulation
<p>Roborace – overtaking regulation Season Alpha An overtaking maneuver is authorized in an overtake zone $\mathcal{Z}_O \in \mathcal{P}(\mathbb{R}^2)$ if a chase vehicle in a preceding trigger zone $\mathcal{Z}_T \in \mathcal{P}(\mathbb{R}^2)$ brings the headway to the vehicle in front below a threshold t_{trig}. This principle is similar to the drag reduction system (DRS) regulation in the <i>Formula 1</i> series (Article 3.18 of the <i>F1</i> Technical Regulations). Once approved, the vehicle in front has to remain on a specified side ($\mathcal{Z}_{O,1} \subset \mathcal{Z}_O$ or $\mathcal{Z}_{O,r} \subset \mathcal{Z}_O$) of the track.</p>	<p>c_k SR^a</p> $c_{\text{RR}} = (-F_T(t) \vee F_C(t)) \mathcal{S}_s(F_T(t) \wedge F_C(t)), \text{ where}$ $F_T(t) = \mathcal{E}(\xi(t)) \cap \mathcal{Z}_T \text{ and } F_C(t) = \left(\frac{s(\xi_o(t)) - s(\xi(t))}{v(t)} < t_{\text{trig}} \right)$ <p>\mathcal{Q}_k</p> $\mathcal{Q}_{\text{RR}} = \begin{cases} \mathcal{Z}_{O,1}, & \text{if overtake specified left} \\ \mathcal{Z}_{O,r}, & \text{otherwise} \end{cases}$
<p>Formula 1 – racing alongside another car The following regulations are considered:</p> <ul style="list-style-type: none"> “Manoeuvres liable to hinder other drivers, such as deliberate crowding of a car beyond the edge of the track or any other abnormal change of direction, are strictly prohibited” [271]-§2b. “It is not permitted to drive any car unnecessarily slowly, erratically or in a manner deemed potentially dangerous to other drivers at any time” [271]-§2d. “Any driver [...] may use the full width of the track [...], provided no significant portion of the car attempting to pass is alongside his. For the avoidance of doubt, if any part of the front wing of the car attempting to pass is alongside the rear wheel of the car in front this will be deemed to be a ‘significant portion’” [270]-§27.7. <p>Consequently, another vehicle may not enter the driving tube of the ego-vehicle if their footprints overlap more than d_{overlap} along the track coordinate s. If driving close to the edge, at least one vehicle width w_{veh} is required to be left to the edge.</p>	<p>c_k</p> $c_{\text{RA}} = \odot((s(\xi_o(t)) - s(\xi(t))) < (1 - d_{\text{overlap}})) l_{\text{veh}}$ <p>\mathcal{Q}_k SR^b</p> $\mathcal{Q}_{\text{RA}} = \left\{ (s, n) \in \mathcal{P}(\mathbb{R}^2) \mid \forall s \in [s(\xi(t_0)), s(\xi(t_h))]: n \in [l(s), u(s)] \right\},$ <p>with $l(s) = \begin{cases} n_{B_1}(s), & \text{if } n(\xi(t_0)) < n(\xi_o(t_0)) \\ \max(n_{B_1}(s) + c(s), n_{B_1}(s) + w_{\text{veh}}), & \text{otherwise} \end{cases}$, and $u(s) = \begin{cases} \min(n_{B_1}(s) + c(s), n_{B_1}(s) - w_{\text{veh}}), & \text{if } n(\xi(t_0)) < n(\xi_o(t_0)) \\ n_{B_1}(s), & \text{otherwise} \end{cases}$, where $c(s) = (n_{B_1}(s) + n_{B_1}(s)) \frac{ n(\xi(t_0)) + n(\xi_o(t_0)) }{2}$</p>
<p>Formula 1 – defending off-line “Any driver moving back towards the racing line, having earlier defended his position off-line, should leave at least one car width between his own car and the edge of the track on the approach to the corner” [271]-§2b.</p> <p>Hence, a defender must leave a vehicle width w_{veh} to the edge of the track if it is closer than d_{corner} to a corner and thereby defends (distance between two vehicles along the s-coordinate below d_{defend}) itself off the racing line (within the track half preceding the curve’s inside \mathcal{Z}_T) against a trailing vehicle. A track segment where the absolute value of the centerline’s curvature exceeds a threshold κ_{thr} is defined as corner. Starting from $s(\xi_o(t))$, $\mathcal{S}_{\text{corner}}$ describes the s-coordinate of the next corner.</p>	<p>c_k SR^{a,c}</p> $c_{\text{DO}} = ((s_{\text{corner}} - s(\xi_o(t))) \leq d_{\text{corner}}) \mathcal{S}_s(F_O(t) \wedge F_D(t)), \text{ where}$ $F_O(t) = \mathcal{E}(\xi(t)) \cap \mathcal{Z}_1 \text{ and } F_D(t) = ((s(\xi_o(t)) - s(\xi(t))) \leq d_{\text{defend}})$ <p>\mathcal{Q}_k SR^b</p> $\mathcal{Q}_{\text{DO}} = \left\{ (s, n) \in \mathcal{P}(\mathbb{R}^2) \mid \forall s \in [s(\xi(t_0)), s(\xi(t_h))]: n \in [l(s), u(s)] \right\}, \text{ with}$ $l(s) = \begin{cases} n_{B_1}(s) - w_{\text{veh}}, & \text{if left turn} \\ n_{B_1}(s), & \text{otherwise} \end{cases}, \quad u(s) = \begin{cases} n_{B_1}(s) & \text{if left turn} \\ n_{B_1}(s) + w_{\text{veh}}, & \text{otherwise} \end{cases}$

^aThe difference of s -coordinates (e.g. $s_o(t) - s(t)$) is only valid for unclosed tracks. For circuits, crossing of the start-finish-line has to be tracked accordingly.

^bA track wider than $2w_{\text{veh}}$ at any position is assumed. If this is not the case, the leading vehicle has to be prioritized for the shared space.

^cCombined turns (e.g. S-turn) must be handled as single turn combination instead of multiple turns.

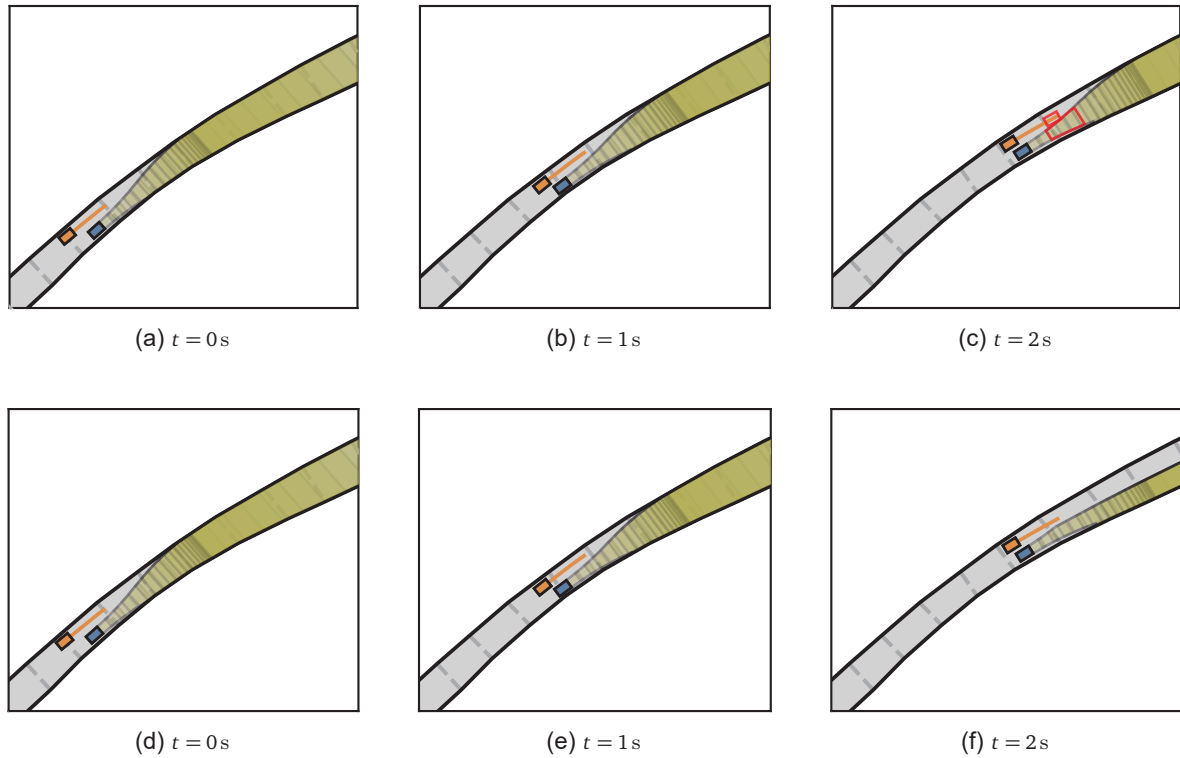


Figure 5.9: Three time instances of an overtaking maneuver of the ego-vehicle (orange) to pass a TP (blue) with its reachable set (green). Guaranteeing safe conditions without any rule-knowledge (top row), the ego-vehicle cannot complete the overtaking maneuver because the emergency trajectory (orange line) intersects the reachable set (red marking). Under applicable regulations (bottom row), the ego-vehicle can safely complete the overtaking maneuver. When comparing the last time step of top and bottom row, it can be seen that the rule “*Formula 1* – racing alongside another car” is in effect, which does not allow the other vehicle to push the ego-vehicle off the track as soon as the vehicles’ footprints overlap along the s -coordinate. The reachable set is reduced accordingly.

5.5.2 Guaranteed Occupied Set

The guaranteed occupied sets have the same objective of detecting and avoiding imminent collisions (Equation (5.25)) as targeted with the reachable sets, but the meaning gained is different. While collisions can be ruled out with reachable sets, only collisions that will inevitably occur if the trajectory is pursued further are detected with the guaranteed occupied sets. The guaranteed occupied set describes the set of states that are guaranteed to be occupied by any part of the vehicle in the future, considering all physically feasible maneuvers of the vehicle. Based on the exact reachable set $\mathcal{R}^e(t_f, \xi_0, \mathcal{U})$, the guaranteed occupied set $\mathcal{G}^\mathcal{E}(t_f, \xi_0, \mathcal{U})$ can be considered as the intersection of the state occupancy $\mathcal{E}(\xi)$ of all the states ξ contained in it [272, pp. 30-32], formally stated as:

$$\mathcal{G}^\mathcal{E}(t_f, \xi_0, \mathcal{U}) = \left\{ \bigcap \mathcal{E}(\xi) \mid \forall \xi \in \mathcal{R}^e(t_f, \xi_0, \mathcal{U}) \right\}. \quad (5.28)$$

This approach is similar to the inevitable collision states studied in robotics [273–275]. An inevitable collision state is a state from which there is eventually a collision with an obstacle independent of the future selected trajectory [273]. In the mentioned concepts in robotics it is to aim at avoiding to enter such a state. The same objective is pursued with the guaranteed sets presented here, but the own trajectory candidate is known and incorporated. Thus, when the occupation along the ego-trajectory intersects the guaranteed occupied set of a TP, the vehicle must not yet be in an inevitable collision state, since there may exist a trajectory other

than the checked one that avoids the collision. Schmidt [276, pp. 60-65] and Schmidt et al. [277] presented a simple underestimation of the guaranteed occupied areas and thus cover one of the few methods of this nature in the automotive domain.

As shown in Figure 5.10, the guaranteed occupancy shrinks and disappears soon as time progresses, while the reachable set expands. This phenomenon is particularly eminent in race vehicles with high accelerations. The underestimation of the guaranteed occupied set presented by Schmidt [276, pp. 60-65] (building on a circle that lies within the vehicle footprint) would exist for an even shorter period of time compared to the guaranteed occupancy set demonstrated here and therefore offers less information value.

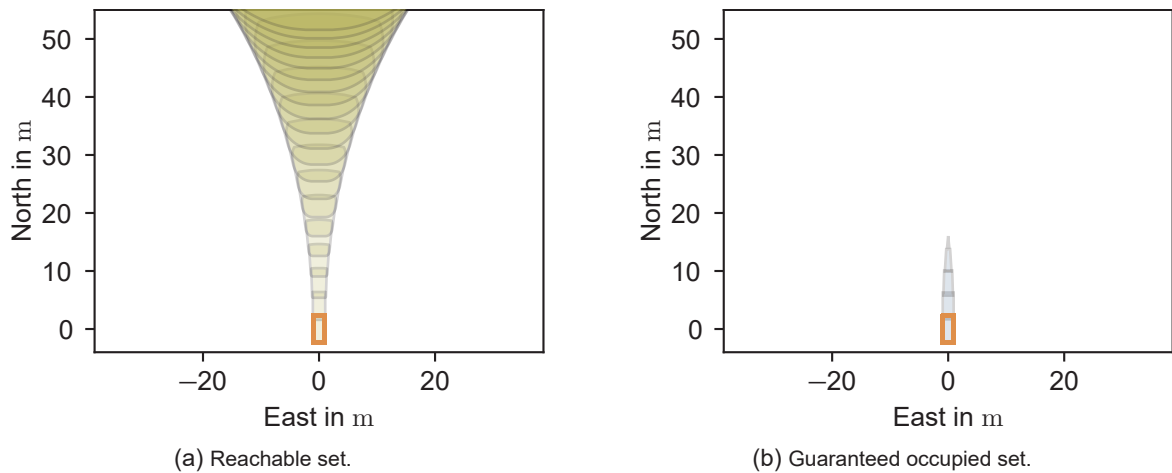


Figure 5.10: Comparison of approximated reachable set (a) and guaranteed occupied set (b). Here shown for a vehicle (orange) with initial velocity 40 m s^{-1} , a time horizon of 2 s , and a time interval of 0.1 s .

Since the underestimation of the guaranteed occupied set offers less information value, a more complex numerical method [272, pp. 30-32] to determine the guaranteed occupied set is used in this work. In this procedure, starting from an initial state ξ_0 , a large number (e.g. 1000) of possible trajectories are numerically simulated up to a given time t_i . Thereby all possible acceleration and steering maneuver combinations are sampled in a fine discretization. This step is necessary, because the previously presented simplified reachable set $\mathcal{R}(t_f, \xi_0, \mathcal{U})$ does not contain any heading information of the vehicle anymore. Based on the determined numerical trajectories, the occupancy $\mathcal{E}(\xi_j)$ by the vehicles at the endpoints of the respective trajectories is determined. The guaranteed occupied set for the time t_i is then the intersection of the individual occupation areas (Figure 5.11). Here, the states that diverge the most are decisive [272, pp. 30-32]. This procedure is carried out in analogous way for a desired number of discrete time points $t_i \in [t_0, t_h]$. Since the calculation is too time-consuming for an online application, this calculation is performed offline for different finely graduated starting velocities. Online, the recalled result for a given velocity $v(\xi_0)$ is translated and rotated to the corresponding state ξ_0 .

As mentioned before, the testing of the emergency trajectories with a robust safety criterion—the rule-based reachable sets—are sufficient for a comprehensive safety guarantee. However, the guaranteed occupied sets serve for early detection of guaranteed unsafe performance trajectories (in accordance with requirement S-1-B-4). For the performance trajectories, the procedure also makes more sense, because the trajectories cover a large time horizon at high speeds. The reachable sets in these time horizons would become unbearably large, and the evaluation of the rules taken at this point in time would also reach its limits when adopted into the future.

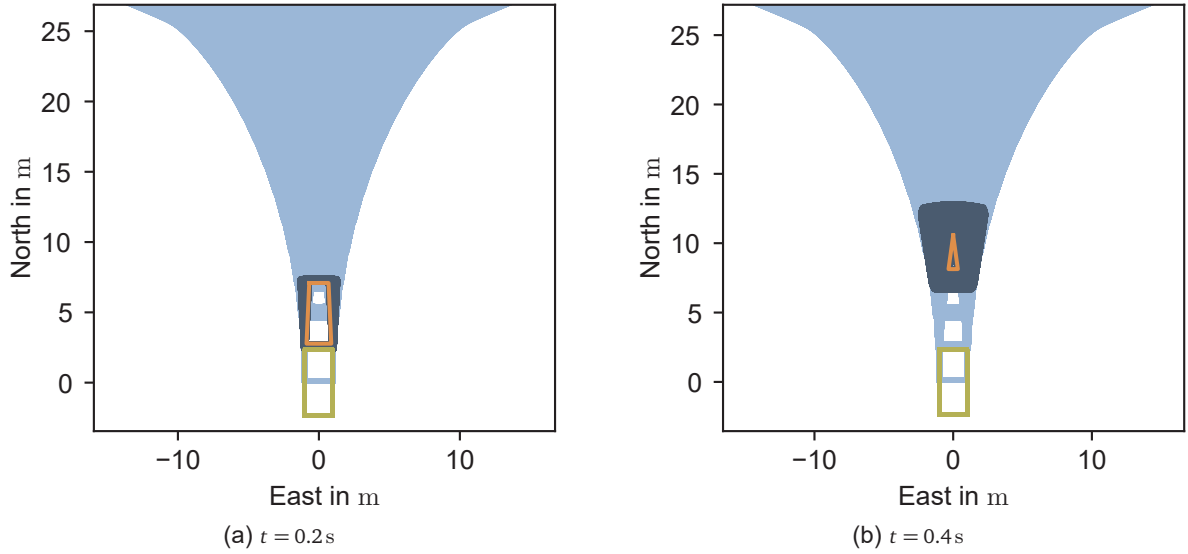


Figure 5.11: Illustration of the procedure for determining the guaranteed occupied set on the basis of two time instances for a vehicle (green) with initial velocity 25 m s^{-1} . The reachable set generated via simulation of more than 400 vehicle trajectories is shown in light blue (plot of individual vehicle footprints constitute the surface). The outline of the vehicle footprint of the respective vehicle poses in the reachable set at the respective time point are displayed in dark blue. The guaranteed occupied set (orange) is the set intersection of the more than 400 vehicle footprints at the respective point in time.

The overall Boolean safety rating $\psi_{\text{occ}}(\cdot)$ w.r.t. to an inevitable collision along the given trajectory by this SupMod is stated as:

$$\psi_{\text{occ}}(\mathcal{M}, \mathcal{O}(t_0), \xi([t_0, t_h])) = \begin{cases} 1, & \text{if } \forall t \in [t_0, t_h], \forall o \in \mathcal{O}(t_0) : \\ & \mathcal{E}(\xi(t)) \cap \mathcal{G}_o^{\mathcal{E}}(t, \xi_0, \mathcal{U}) = \emptyset, \\ 0, & \text{otherwise} \end{cases} \quad (5.29)$$

where $\mathcal{G}_o^{\mathcal{E}}(t, \xi_0, \mathcal{U})$ denotes the concise notation for the guaranteed occupied area by object o at time t . For a guaranteed occupied set discretized in time into successive intervals within $[t_0, t_h]$, $\mathcal{G}_o^{\mathcal{E}}(t, \xi_0, \mathcal{U})$ is understood as the selection of the interval of the guaranteed occupied set accommodated for the given time t .

5.6 Classification and Selection

Based on the ratings of the SupMods Ψ presented earlier, each performance and emergency trajectory is classified as safe or unsafe. This is implemented via a conjunction over the boolean safety scores $s_i = \psi_i(\mathcal{M}, \mathcal{O}(t_0), \xi([t_0, t_h]))$ generated by the set of SupMods $\Psi_{\text{perf}} \subset \Psi$ and $\Psi_{\text{em}} \subset \Psi$ relevant for the respective trajectory:

$$s_{\text{perf}} = \bigwedge_{\psi_i \in \Psi_{\text{perf}}} s_i = s_{\text{perf},1} \wedge s_{\text{perf},2} \wedge \dots \wedge s_{N_{\text{perf}}}, \quad s_{\text{em}} = \bigwedge_{\psi_i \in \Psi_{\text{em}}} s_i. \quad (5.30)$$

The SupMods relevant for each trajectory have already been mentioned in the previous sections and are marked in Figure 5.12. Formally summarized, the respective sets are composed as

follows:

$$\Psi_{\text{perf}} = \{\psi_{\text{stat}}, \psi_{\text{acc}}, \psi_{\text{rule}}, \psi_{\text{dc}}, \psi_{\text{occ}}\}, \quad \Psi_{\text{em}} = \{\psi_{\text{stat}}, \psi_{\text{ses}}, \psi_{\text{acc}}, \psi_{\text{rule}}, \psi_{\text{dc}}, \psi_{\text{reach}}\}. \quad (5.31)$$

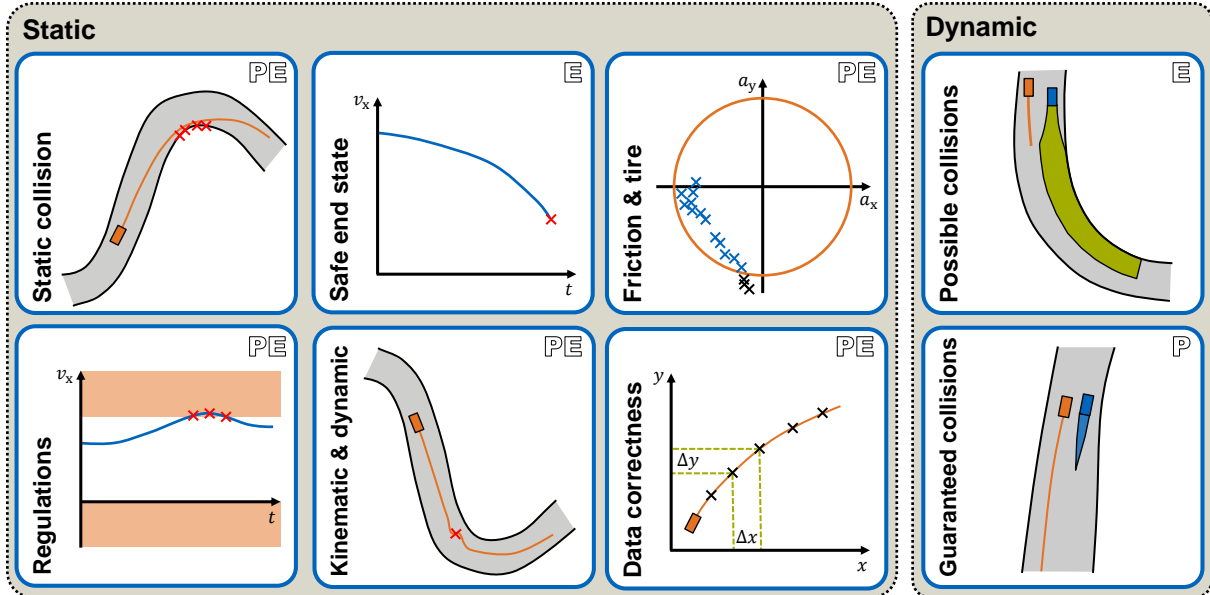


Figure 5.12: Overview of all implemented SupMods in schematic representation. The SupMods are assigned to clusters static and dynamic according to the application in the respective static or dynamic environment. An identifier in the upper right corner indicates whether the module is applied for the performance (P) and/or emergency (E) trajectory.

Based on the aggregated scores s_{perf} and s_{em} for performance and emergency trajectory, a safe candidate is selected based on well established mechanisms [237] and passed to the controller (Figure 5.13). It should be noted that a safe emergency plan has to exist at any time. Following this principle, even if the performance trajectory is safe but the emergency trajectory is unsafe, a stopping maneuver is still carried out on the last valid emergency trajectory. This is necessary, because it cannot be guaranteed that the underlying trajectory planner will find a new valid emergency plan in the near future. To facilitate this fallback to an emergency trajectory from the past time step, it must be guaranteed that the performance and emergency trajectory in each planning step are congruent for a certain amount of time (safety factor times average update frequency of the planner). This principle is analyzed and illuminated in more detail in Figure 5.14. Furthermore, if the planner fails to generate a new pair of trajectories within this congruent time-interval (timeout via watchdog), the Supervisor also has to switch to the latest valid emergency trajectory (this aspect, however, is not investigated any further in this work).

In addition, it should be noted for the sake of completeness that this principle only works, if the first set of trajectories contains a safe emergency trajectory. This condition is straightforward to meet, if the system is launched from standstill and is not switched on while the vehicle is moving.

In summary, in this chapter the individual steps for the development of an OV for a trajectory planner of an automated race vehicle were worked through. In doing so, requirements were first defined and, based on this, the architecture and corresponding metrics were developed. In the next chapter, the developed modules are to be integrated into the overall system, tested and evaluated.

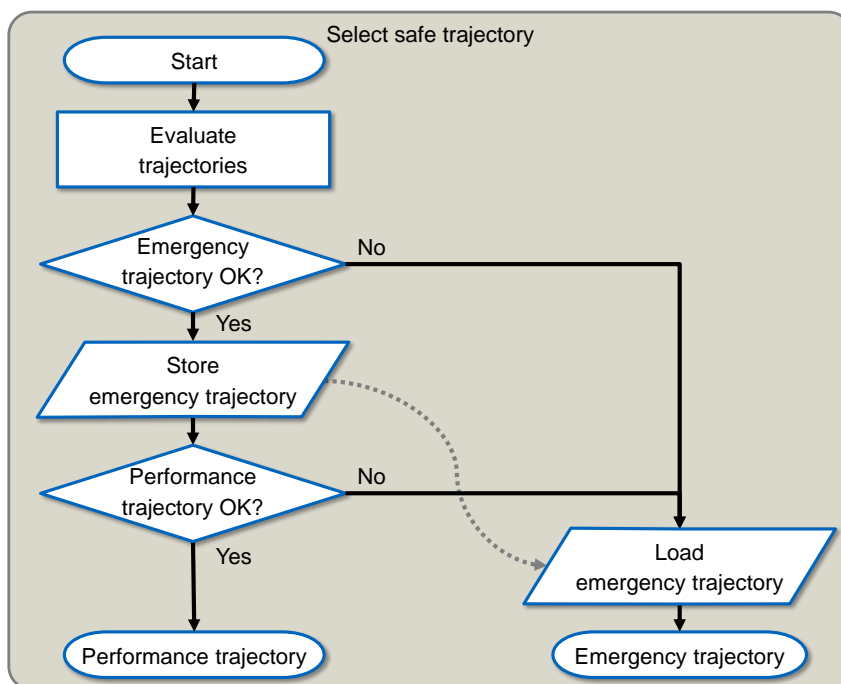


Figure 5.13: Decision procedure for the selection of a safe trajectory, based on [246].

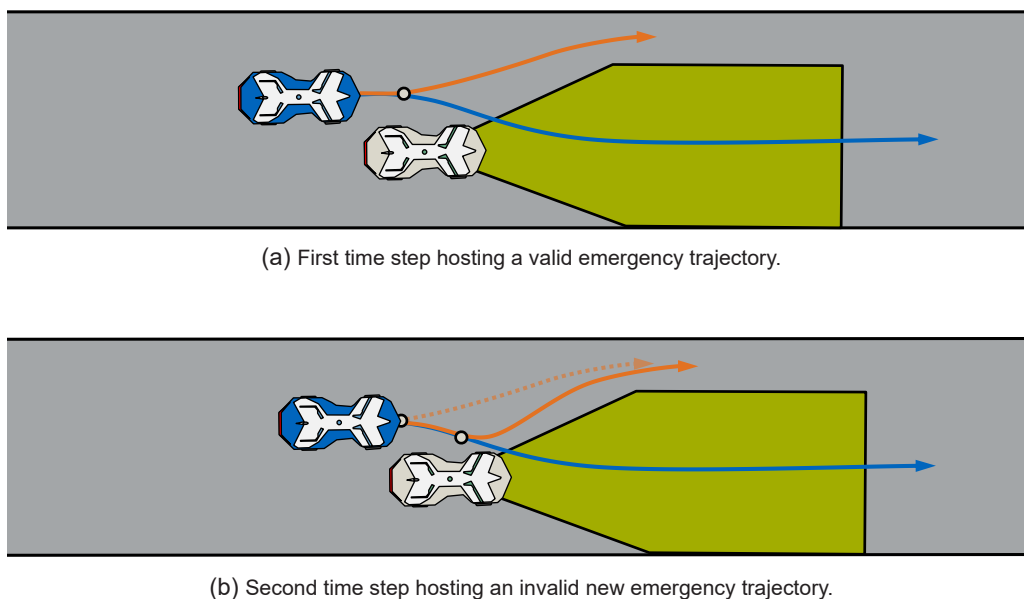


Figure 5.14: Schematic diagram illustrating the congruence of emergency and performance trajectories for a certain time period. The next point up to which the trajectories are congruent is indicated by a white circle with a black border. In this graph, it is assumed that only the orange emergency trajectory of the ego-vehicle (blue) is tested against the abstracted green reachable set of a TP (gray). In the first time step, the emergency trajectory is still valid with respect to this assumption. In the second time step, the vehicles are further ahead and the ego-vehicle has generated a new set of trajectories. The new emergency trajectory would result in the ego-vehicle intersecting the set and is classified as unsafe accordingly. For this reason, it is necessary to switch to the previous emergency trajectory (dashed). Since the vehicle has only moved on the congruent trajectory part in the meantime, a changeover at this point is not problematic.

6 Results

In this chapter, the fourth research question is addressed:

Q 4: *What does the validation of a prototypical OV method involve in order to meet the purpose of a proof of concept for the overall method?*

In this chapter, the approach explained in Chapter 4 and tailored for a trajectory planner in Chapter 5—available open-source [278]—is integrated into the overall SW (Section 6.1), tested (Section 6.2), and evaluated. The evaluation is done with two different approaches. In the scenario-based evaluation (Section 6.3), among other things, deliberate critical or collision scenarios are tested. In the random-based evaluation (Section 6.4), no-fire tests are performed on real vehicle data.

6.1 Software Stack Integration

The implemented Supervisor is evaluated within the scope of the *Roborace* race series [279, 280]. For this purpose, an overall SW [109, 281] was developed at the Institute of Automotive Technology to operate the race vehicle. An overview of the software modules is depicted in Figure 6.1. The Supervisor is integrated according to the architecture definition given in Section 5.2.

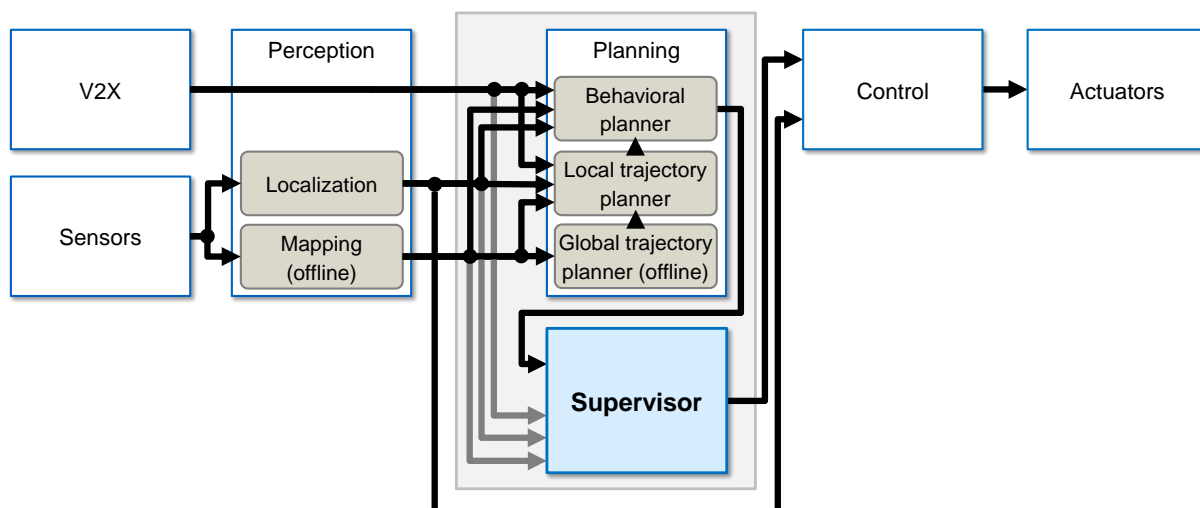


Figure 6.1: Schematic illustration of the SW framework used during the *Roborace Season Alpha*, supplemented by the Supervisor.

This paragraph explains the basic principle of the SW stack and the integration of the Supervisor in accordance with the system architectural design (Section 5.2). The explanation is based on the central planning module. The local trajectory planner [71, 108] of the vehicle is initialized with an offline generated map [30] and global trajectory [101, 282] (optimal racing line). During

runtime, the local trajectory planner generates a set of feasible trajectories (e.g., overtake left, follow, overtake right) based on the global trajectory, current pose, and object list (V2X in this case). The downstream behavioral planner selects based on a global view (e.g. V2X information or global track conditions and regulations) an appropriate performance trajectory paired with a matching emergency trajectory. The trajectories and all perception data used for planning are sent via ZeroMQ [283] interface from the planner to the Supervisor. A trajectory evaluated as safe by the Supervisor is then transmitted via User Datagram Protocol (UDP) communication to the controller [101, 284], which incorporates localization estimates [32, 38] to control the vehicle accordingly.

It should be noted that during the multi-vehicle real tests in the *Roborace Season Alpha* (2019), the Supervisor was not yet included in the SW stack and the trajectories were sent from the planner directly to the controller. However, all data from the planner (outputs and inputs) were logged during the real runs and are thus used as inputs without any negative effects. In this context, only passive Supervisor tests (i.e. no intervention in case of an unsafe rating takes place) are evaluated. This is reasonable in this case, because primarily safe states are expected during the real-world tests. Due to the COVID-19 pandemic and strategic decisions on the part of *Roborace*, no races with multiple vehicles simultaneously on the track took place in the subsequent events until the writing of this thesis.

6.2 Integration Tests

The SW stack integration is followed by tests that confirm a successful integration. First, the interfaces are tested (Subsection 6.2.1) and then the fulfillment of the requirements is assessed (Subsection 6.2.2).

6.2.1 Test of Internal and External Interfaces

Once the integration is completed, tests are performed to prove that the process was consistent and correct. For this purpose, ISO 26262-4:2018, 7 [23] proposes a number of individual tests. In the context of this prototypical implementation, 21 interface tests were performed. A detailed list of the performed tests can be found in the Appendix in Table A.2. The tests belong to the categories normal operation, missing data, incorrect data, and timing variations. For example, while one test checks the normal operation of the vehicle, another checks whether this is not the case when data of individual modules is missing. In this way, the proper interface structure of the system is verified.

For each of these tests, a specific input configuration was chosen and the expected output was defined. After execution of the test, the observed output of the system was compared with the expectation. The test is classified as passed or failed according to match or mismatch of expectation and observation. In 2 of the 21 tests performed, the output expected for an ideal system was not observed and was accordingly classified as failed. These test cases provided an incorrect map and an incorrect object list, respectively. However, since according to stated assumptions (Section 5.2), a validated perception is assumed, the tests are listed for the sake of completeness but are evaluated as passed under given assumptions. These assumptions are also feasible in practice with human review of the offline generated maps, as well as reliable V2X communication for the object list.

The tests carried out here do not satisfy the comprehensive and systematic requirement of compliance with the standard for a series product. However, from the point of view of the author and related work, there are no indications of problems that would arise with a comprehensive implementation. Further guidelines and recommendations can be found in ISO 26262-4:2018, 7 [23].

6.2.2 Test of the Requirements

The integration tests also include reviewing the requirements established in Section 4.1. In the first stage of the development process presented here, the requirements for a safe operation (S-1-A, Section 5.1) and for an OV module itself (S-1-B, Subsection 4.1.2) were discussed. This section examines the conformance with these requirements of the developed and integrated Supervisor.

The requirements for a safe trajectory (S-1-A) are tackled with functional tests, for which ISO 26262-4:2018, 7 and ISO 26262-6:2018, 10 and 11 [23] recommends requirement-based tests and fault injection. According to Table 6.1, for each requirement, one test was performed with a compliant test and one with fault injection. For example, for testing requirement S-1-A-2, no physical interaction, a safe trajectory candidate and a candidate eventually colliding were evaluated. The system successfully detected, in line with the requirements, the hazard and switched to the emergency trajectory. During these tests, a single snapshot of the environment and a single trajectory candidate was injected and evaluated. While these requirement tests only evaluate individual snapshots, the evaluation (in the following Sections) aims to provide a holistic assessment. In total, 14 tests (2 per requirement) were evaluated. Expectation and observation differed for only two candidates. However, this was a situation excluded by the assumptions. Thus, the requirement tests regarding S-1-A are evaluated as passed.

In contrast to the functional tests for the requirements from S-1-A, the requirements S-1-B, which concern the Supervisor itself, are examined with non-functional checks (Table 6.2). In this process, it is evaluated whether the system has specified/expected properties. For example, the property S-1-B-5, real-time capability, is evaluated by examining the actual computing time over a longer period of time. In this case, Figure 6.2 shows a plot of the distribution of computation time required for one iteration (safety evaluation of the performance and emergency trajectory), logged over several laps in a multi-vehicle scenario on a race track in Modena, Italy (Figure A.3). The computing time for the prototype Python 3.7 code deployed on an Intel Xeon E3-1245 3.3 GHz does not exceed the posed real-time limit of 100 ms. This limit was set for the prototype in order to guarantee processing of a pair of trajectories, before a new set is generated by the planner (10 Hz update-rate). Consequently, this test is marked as passed. For this test, it should be noted that the final implementation of the Supervisor should be done in a compiled language, for example C++, where significantly faster performance is expected. All requirements of group S-1-B are evaluated as passed.

As stated in the previous section, the tests performed for this prototype do not meet the extent requested by ISO 26262 [23]. Since no problems are expected in the implementation of these tests in accordance with the derivation in the related work section, this is not in the focus of this thesis. Further details and requirements can be taken from ISO 26262-4:2018, 7 and ISO 26262-6:2018, clause 10 and 11 [23]. Investigations towards tests in agile SW development is part of current research [285].

6 Results

Table 6.1: Evaluation of requirements for a safe trajectory (S-1-A) with associated scores (passed test (PT) or passed test given assumptions (PTGA)) based on expected and observed output.

Re-quire-ment	Descrip-tion	Input details	Expected output	Observed output	PT	PTGA
S-1-A-1	Accu-rate object list	Proper object list	Vehicle following trajectory	Vehicle following trajectory	True	True
		Fault-injection – object list hosting wrong pose	Vehicle decelerating on latest emergency trajectory	Vehicle following trajectory ^a	False	True
S-1-A-2	No physical interactions	Trajectory candidate not intersecting object vehicle path	Vehicle following trajectory	Vehicle following trajectory	True	True
		Fault-injection – trajectory candidate colliding with object vehicle in future	Vehicle decelerating on latest emergency trajectory	Vehicle decelerating on latest emergency trajectory	True	True
S-1-A-3	Correct origin	Proper localization estimate	Vehicle following trajectory	Vehicle following trajectory	True	True
		Fault-injection - localization estimate hosting wrong pose	Vehicle decelerating on latest emergency trajectory	Vehicle following trajectory ^a	False	True
S-1-A-4	Feasible friction request	Trajectory candidate hosting feasible friction request	Vehicle following trajectory	Vehicle following trajectory	True	True
		Fault-injection – trajectory candidate exceeding friction limit	Vehicle decelerating on latest emergency trajectory	Vehicle decelerating on latest emergency trajectory	True	True
S-1-A-5	Obeyed rules	Trajectory candidate respecting applicable rules	Vehicle following trajectory	Vehicle following trajectory	True	True
		Fault-injection – trajectory candidate violating rule (here: maximum velocity)	Vehicle decelerating on latest emergency trajectory	Vehicle decelerating on latest emergency trajectory	True	True
S-1-A-6	Re-spected vehicle characteristics	Trajectory candidate respecting vehicle characteristics	Vehicle following trajectory	Vehicle following trajectory	True	True
		Fault-injection – trajectory candidate violating vehicle characteristics (here: curvature exceeds turn radius)	Vehicle decelerating on latest emergency trajectory	Vehicle decelerating on latest emergency trajectory	True	True
S-1-A-7	Proper trajectory data	Trajectory candidate hosting proper data	Vehicle following trajectory	Vehicle following trajectory	True	True
		Fault-injection – trajectory candidate hosting violating data (here: s -coordinate not matching x - y -coordinates)	Vehicle decelerating on latest emergency trajectory	Vehicle decelerating on latest emergency trajectory	True	True

^aFault not tracked due to assumption of reliable perception data.

6.3 Scenario-Based Evaluation

To evaluate the validity of the implemented framework, it is necessary to conduct extensive testing. As discussed in Section 4.4, the focus—in line with the “appropriate set” of measures called for in ISO 26262-4:2018, 8.4.3.4 [23]—is on scenario-based and random-based tests. In this section, the scenario-based evaluations are presented. These tests involve testing a wide range of “known” scenarios according to SOTIF [128]. Among the scenarios are explicit situations,

Table 6.2: Evaluation of the requirements imposed on a Supervisor (S-1-B) with associated scores (passed test (PT)).

Re-quire-ment	Expectation	Observation	PT
S-1-B-1	Simple and predictable	Straightforward formal methods used (Section 5.4 and 5.5)	True
S-1-B-2	Detection of all faults	Holistic coverage pursued (Section 5.1) to be proved by evaluation (Section 6.3 and 6.4)	True
S-1-B-3	Guarantee of a safe state	All-time feasible emergency trajectory (Section 5.6)	True
S-1-B-4	Prevention	Detection of collisions or issues as soon as they appear in the trajectory (reachable sets by nature, in addition guaranteed occupied sets in large horizon performance trajectories)	True
S-1-B-5	Real-time capability	Prototype real-time limit of 100 ms not exceeded (Figure 6.2), compiled C++ implementation assumed significantly faster	True
S-1-B-6	Separation from overall system	Individual module (Section 6.1), can operate on separate HW	True
S-1-B-7	Modularity and adaptability	Modular architecture (SupMods), which can be adapted to current conditions as required	True

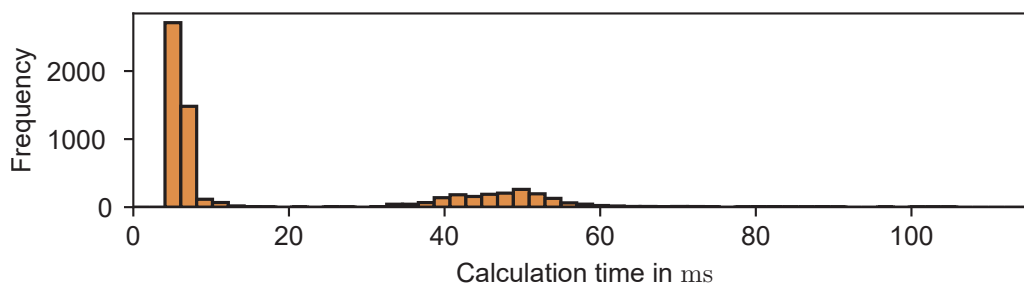


Figure 6.2: Distribution of logged calculation time per iteration (evaluation of both, performance and emergency trajectory) for several multi-vehicle laps on a track in Modena, Italy. Depending on the length of the trajectories provided and vehicles in the vicinity of the vehicle, the calculation time varies in each iteration. The two observed clusters result from purely static tests when no other vehicle is in the vicinity (left cluster) and a full test coverage in the vicinity of other vehicles (right cluster).

which are difficult to reproduce in reality or require expensive consequences (e.g., collisions or critical events). Furthermore, the goal of the scenario generation is to include the injection of faults that an online learning planner could generate. The overall aim of these tests, in line with the philosophy of SOTIF [128], is to know in advance that the vehicle will behave safely in certain situations, so as not to experience the effects for the first time in real operation. Since handcrafted scenario-based tests strongly depend on the view of the creator [127], the author developed a graphical user interface [245, 286] that makes it easy to construct scenarios. This allowed several students to contribute to a scenario database. The graphical user interface—the Scenario Architect—for generation of evaluation scenarios is briefly introduced in Subsection 6.3.1. Then, the evaluation methods used and associated results are discussed in Subsection 6.3.2.

6.3.1 Scenario Architect

The contribution and need for the Scenario Architect created was elaborated in detail in a dedicated paper [245]. The paper explains the underlying mechanisms in detail. The Scenario Architect is available on GitHub [286]. In this section, the core aspects of the Scenario Architect are briefly revisited and new features (that were not covered in the paper yet) are described in more detail.

Basic Functionality

The graphical user interface of the Scenario Architect consists of two windows (Figure 6.3 and 6.4). In the main window (Figure 6.3), it is possible to define the track boundaries \mathcal{B}_l and \mathcal{B}_r as well as the trajectories $\xi([t_0, t_h])$ of multiple vehicles with a pointing device only. The coordinates of the boundaries are connected in pairs and form a track section (gray). The trajectory of the object vehicle (blue) is currently in edit mode. Here the points that have been defined are visible (black crosses). Between the points defined with the pointing device, the Scenario Architect calculates a curvature continuous trajectory using cubic splines. In addition to the paths displayed throughout the entire scenario, a single point in time is highlighted using two vehicle footprints (rectangles) and the trajectory of the ego-vehicle (red). This time instant can be selected in the time window (Figure 6.4) with the pointing device and displayed dynamically/live.

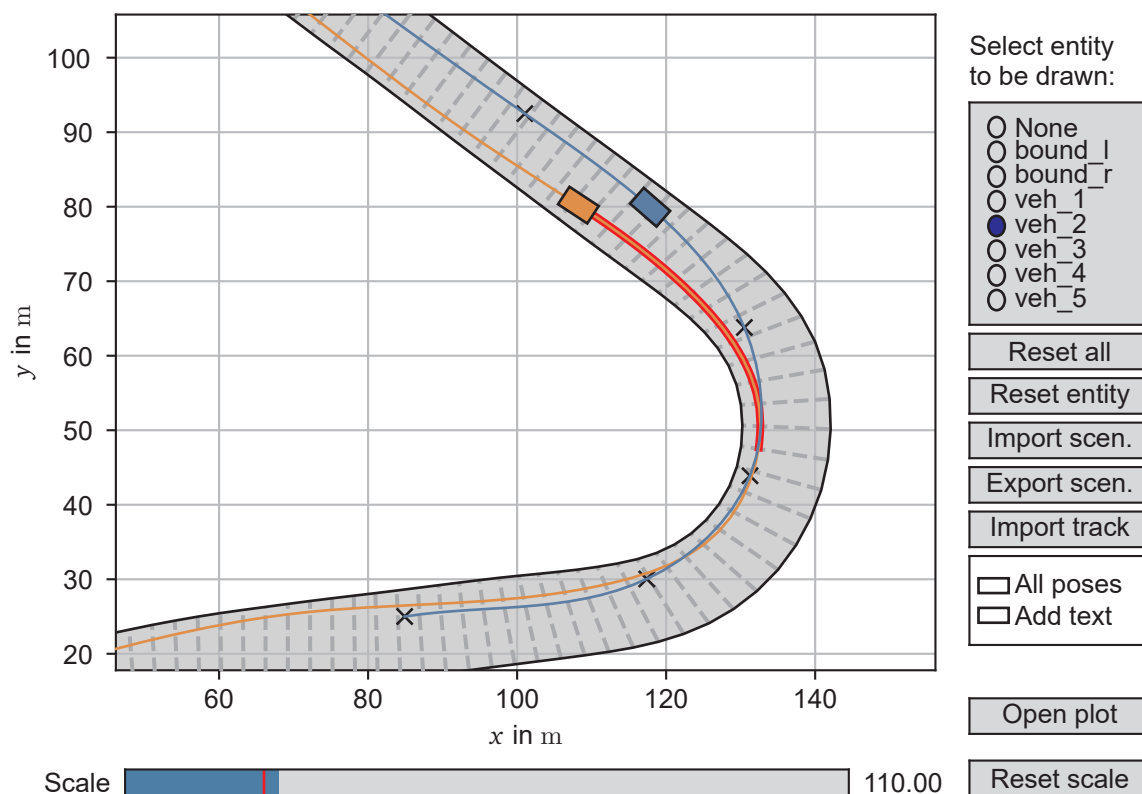


Figure 6.3: Main window of the Scenario Architect. A specific time of the scenario is highlighted by associated vehicle footprint boxes and the red trajectory of the ego-vehicle. The right side contains radio buttons, buttons, and checkboxes to make changes to individual entities in the scenario. Currently the vehicle “veh_2” is selected and in edit mode accordingly.

Figure 6.4 depicts the corresponding temporal information window for the scenario shown in Figure 6.3. In this window the temporal information of the scenario is displayed and edited. The velocity and acceleration of each vehicle in the scene is plotted over time. In addition, the velocity is indicated over the location, since in this plot the speed can be modified by the user using a graphical interface, here shown for the object vehicle (blue), which is in edit mode. The black points in the velocity-location-plot can be moved by the pointing device individually or by dragging in a batch. Initially, the velocity profile is initialized live with the maximum possible velocity, taking into account the acceleration limits, while the corresponding path is drawn in the main window. The red cursor represents the point in time that is currently displayed in the main window (Figure 6.3). The cursor can be adjusted in real-time by moving the pointing device above one of the time plots. The lowest plot deals with the ground truth generation for the safety of

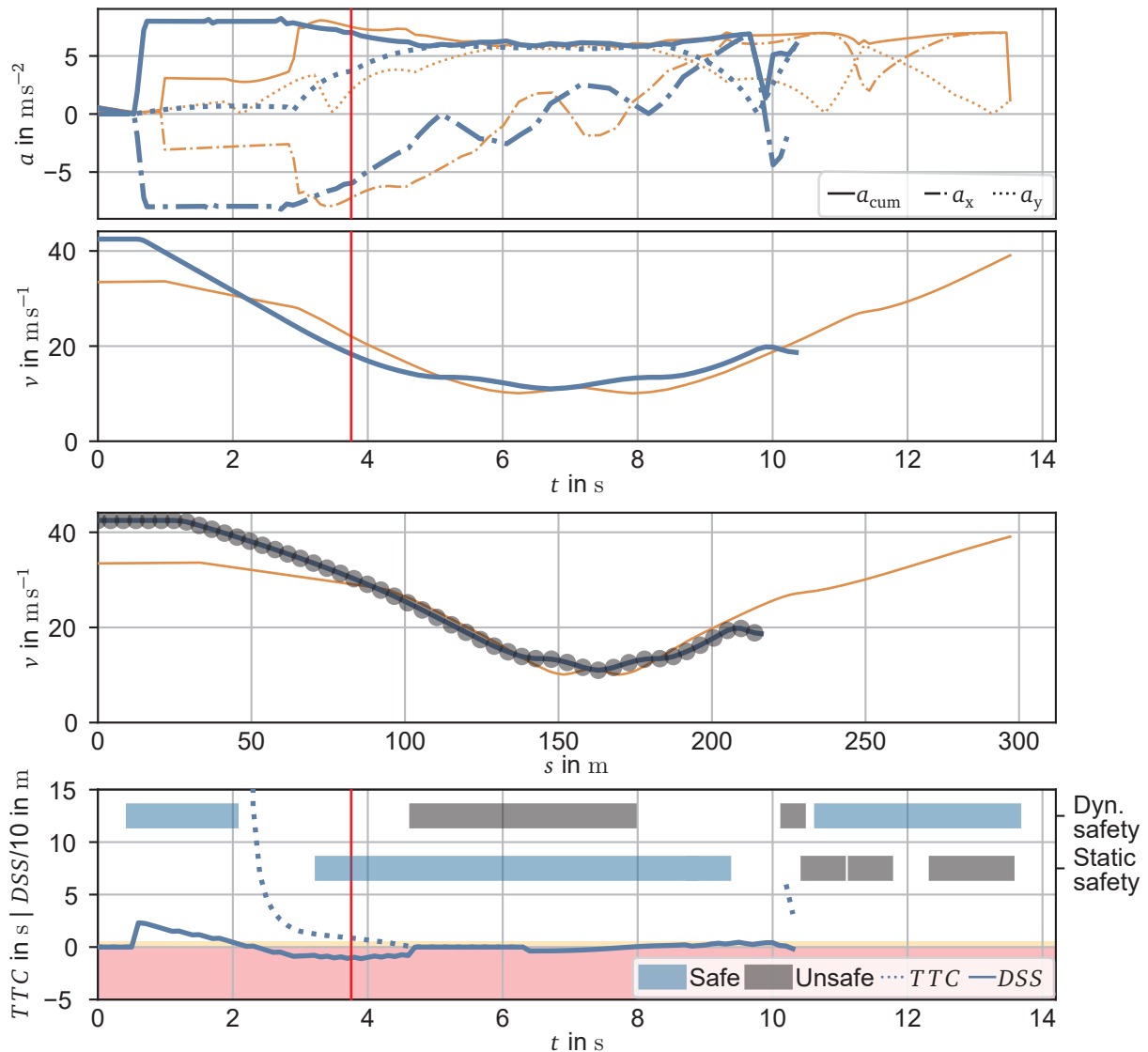


Figure 6.4: Temporal information window of the Scenario Architect that displays the progression of acceleration and velocity over the entire scenario horizon. With the red marker a time-stamp can be selected, which is then displayed in the main window.

each situation in the scenario. This part was not included in the paper [245] and will be explained in more detail in the following.

Ground Truth Generation

A conventional evaluation of the scenarios would take place with an active Supervisor. Here, the system must never enter a self-inflicted unsafe situation - even in scenarios with fault injection. This is similar to conventional field tests. However, the problem is that even a restrictive (e.g. a system that prefers to stop even in safe situations) or accidentally correct (e.g. a system that stops early due to other influences without detecting the actual error) system may pass this test.

To get more details about the performance of the Supervisor, more detailed tests are performed. For this purpose, it is determined in which scenes the system has to achieve a particular rating. For this purpose, areas that have to be definitely evaluated as unsafe and areas that have to be definitely evaluated as safe have to be defined. In order for the defined areas (ground truth) to retain their validity, these tests have to be carried out with a passive Supervisor. In other words,

the Supervisor assesses the situation, but does not switch to another trajectory when facing an unsafe rating, as this would interfere with the control loop and change the future situations. Accordingly, it is checked whether the generated evaluation is consistent with the ground truth.

The assignment of scenes in a scenario to a corresponding ground truth rating can be done by a human, or (ideally) systematically by machines. Such an automated process is accommodated in the Scenario Architect and shown in the bottom plot of Figure 6.4. It can be seen that the time-line is colored in an black tone when an unsafe area is present and in a blue tone when the corresponding scene is considered safe. The evaluation is done separately for a static and a dynamic environment (here called “static safety” and “dyn. safety”) and fused afterwards. Since there are different challenges, these will be described separately in the following paragraphs.

For the ground truth in the static environment $\hat{s}_{\text{stat}}(\xi(t))$ for a state $\xi(t)$, straightforward methods are applied. For this purpose, each state $\xi(t)$ of the ego-vehicle within the scenario is checked whether it complies with all safety-relevant requirements. In addition, an interval can be defined in which the evaluation is allowed to take any ratings. This is required because a hard switch from unsafe to safe can lead to discretization problems in the border area and is non-trivial to specify.

A concrete example for this problem is the definition of the limits regarding a collision with the track boundaries. If the situation were to be judged unsafe as soon as the vehicle touches the barrier and safe in all other cases, the Supervisor would have to assess the situation as safe even with a small distance to the barrier (e.g. 1 mm). Since an evaluation as unsafe in such a situation is not necessarily wrong, and no clear boundary can be defined beyond which a safe evaluation is expected, it helps to define a region without any specifications within the area that is safe on paper. Consequently, in this example, an unsafe evaluation would be expected upon contact with the track boundary. Up to a defined distance (e.g. 30 cm) any evaluation is allowed. From the defined distance on, the assessment is expected to result as safe. From a safety point of view, it is essential that the unspecified interval lies within the range specified as “safe” on paper and that the Supervisor requires an “unsafe” rating in the case of an actual unsafe condition.

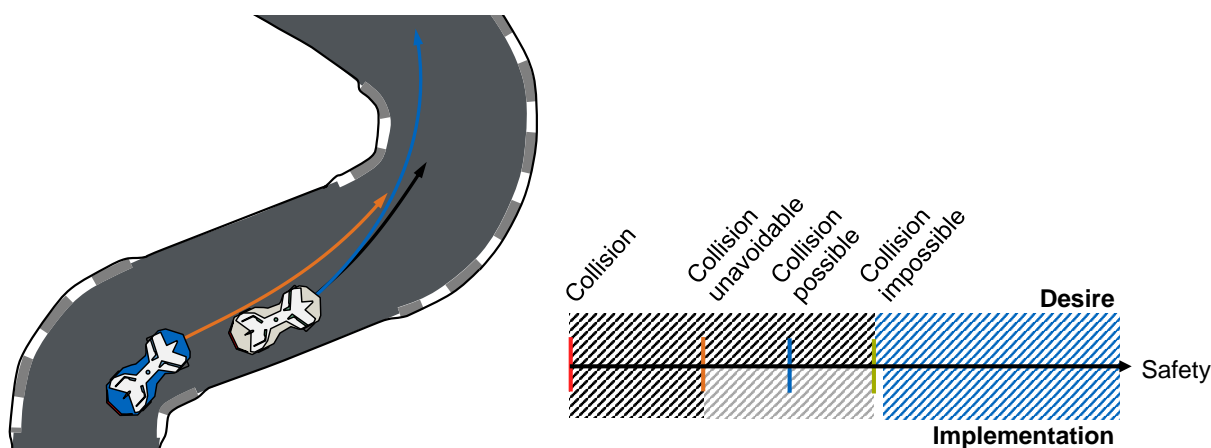
Within the checks of an static environment the following properties are evaluated:

- Collisions with the track boundaries: Examine the considered pose for overlapping with one of the track boundaries.
- Acceleration limits exceeded: Check if a_{comb} exceeds the specified limits in the corresponding state.
- Motor limits exceeded: Check if the desired acceleration can be achieved by the motor in the current state.
- Turning radius maintained: Check whether the desired turning radius is drivable in the current state.

As mentioned before, for each of these checks two thresholds are introduced. For example, for the collision with the track limits $\hat{s}_{\text{stat}}(\xi(t)) = 0$ (unsafe) is set for overlap of the track boundaries with the actual vehicle footprint at state $\xi(t)$ and $\hat{s}_{\text{stat}}(\xi(t)) = \emptyset$ (any rating allowed) is set for overlap with a slightly larger vehicle geometry at state $\xi(t)$. The safe rating $\hat{s}_{\text{stat}}(\xi(t)) = 1$ is expected whenever there is no overlap with any of the shapes. A similar procedure is followed for the other properties. At this point it should be noted that the tests performed for generation of the ground truth should not be identical / based on the same method as those for the safety score

of the Supervisor, otherwise systematic errors can occur in both systems, which can then only be uncovered in the real tests. In concrete terms, this means that for the previously mentioned example with the track limits, the tests in the Supervisor were addressed by overlapping with a tube around the entire trajectory, while in the Scenario Architect each individual pose is evaluated.

When defining ground truth in a dynamic environment, it is no longer trivial to determine when a situation is safe or unsafe. Other road users can choose arbitrary actions and thus lead to critical situations in different ways. In Figure 6.5a a situation is shown, where with fixed trajectory of the ego-vehicle (orange) depending on the choice of maneuver of the preceding vehicle (gray) a collision can occur (black trajectory) or not (blue trajectory).



(a) Example situation where, depending on the maneuver of the vehicle in front, the situation can result in a collision (black trajectory) or not (blue trajectory).

(b) Qualitative listing of the different levels of safety regarding another dynamic vehicle with desired as well as implemented ground truth areas (hatched blue: Supervisor must evaluate situation as safe; hatched black: Supervisor must evaluate situation as unsafe; hatched faint black: identical to hatched black, but only imposed when certain situational conditions are met, for example intersecting paths, not during overtaking).

Figure 6.5: Schematic representation of the challenges and characteristic thresholds in the generation of ground truth for dynamic scenarios.

In Figure 6.5b concrete safety levels are plotted on a beam in a qualitative space. Where the collision has occurred on the far left, the situation just explained is located at “collision possible” (blue line). Two other unambiguous boundaries are defined here: “collision unavoidable”, when a collision is present regardless of the action of the other road user and “collision impossible”, when no collision can occur regardless of the action of the other road user.

The desired ground truth would be the ranges shown in the top of Figure 6.5b. Thus, the Supervisor would always have to evaluate as safe, if a collision is impossible in accordance with applicable regulations and evaluate as unsafe in all other cases. However, since the applicable regulations play a decisive role here and have to be implemented indent as in the Supervisor, this would lead to the same assessment methods and would not provide any added value in the evaluation [148]. For this reason, independent of the rules, the barriers “collision unavoidable” ($\hat{s}_{\text{dyn}}(\xi(t)) = 0$) and “collision impossible” ($\hat{s}_{\text{dyn}}(\xi(t)) = 1$) were chosen as ground truth (Figure 6.5b bottom ranges). In the range in between, only possible collisions in currently overlapping path tubes (the future of the other vehicles is known in the Scenario Architect) are expected to be assessed as unsafe (this also includes the situation in Figure 6.5a). All other cases in this range are undefined and accept any rating of the Supervisor, but are judged separately by a human with respect to applicable rules. This includes, for example, successful overtaking maneuvers. It should be noted that the ground truth range for “safe” is set somewhat further to the right than “collision impossible”, since the online safety assessment often relies on

over-approximated worst-case maneuvers (e.g. reachable sets) for the object vehicles, which therefore detect unsafe behavior due to this approximation slightly more conservative than in reality.

In this work the determination of these intervals is done with the Difference of Space Distance and Stopping Distance (DSS) [287, 288]. The DSS is a Surrogate Safety Metric (SSM) and defined as the distance between two vehicles after applying the maximum braking deceleration a_{\max} (assuming identical for both vehicles), with considering reaction time t_{react} and initial bumper-to-bumper distance d_x :

$$DSS = d_{\text{brake,le}} + d_x - d_{\text{brake,ego}} - d_{\text{react,ego}} = \frac{v_{\text{le}}^2}{2a_{\max}} + d_x - \frac{v_{\text{ego}}^2}{2a_{\max}} - t_{\text{react}}v_{\text{ego}}, \quad (6.1)$$

where v_{le} and v_{ego} denote the initial velocity of the lead vehicle and ego-vehicle, respectively. In addition to the bumper-to-bumper distance d_x , each of the other summands also describes a distance, namely the braking distance of the vehicle in front $d_{\text{brake,le}}$, the braking distance of the ego-vehicle $d_{\text{brake,ego}}$, and the distance traveled during the reaction time $d_{\text{react,ego}}$ (assuming constant velocity v_{ego}).

The Scenario Architect determines a concrete emergency trajectory for the ego-vehicle in each time step, which is passed to the Supervisor as a bundle with the performance trajectory when a scenario is replayed. For this reason, the DSS is modified to use the length of the concrete emergency trajectory of the ego-vehicle to standstill at each time step instead of the calculated braking distance $d_{\text{brake,ego}}$. If the DSS is less than or equal to zero and the vehicles are on a collision course, a collision can occur under worst case assumptions for the lead vehicle. If the DSS is greater than zero, the collision can definitely be avoided with the current emergency trajectory. Since a bumper-to-bumper distance close to zero meters can still be considered critical, a slightly larger value of DSS is recommended as ground truth, above which the assessment is expected to result in a safe rating. Following the principle introduced earlier, two thresholds are defined, separating the unsafe, undefined and safe ground truth safety areas. It should be noted here that collisions shortly before impact, for which the leading vehicle may be responsible, also have to be detected by this means (the DSS does not take into account the applicable rules). However, this is also desirable in the close range and checked via guaranteed occupied sets. In case the DSS is negative, but the vehicles are not on a collision course (simple forward checking of the defined paths), the previously unspecified evaluation is applied, since knowledge about applicable rules is necessary to define the evaluation explicitly. Further details and parameters regarding the ground truth generation are given in the Appendix in Table A.3.

Scenarios exported by the Scenario Architect can be replayed afterwards with ground truth, data of the ego-vehicle, and all environment data.

6.3.2 Evaluation Results

In this section, the evaluation results of the scenario-based tests are presented. In this context, 121 scenarios with up to three vehicles were examined. The scenarios were created with the Scenario Architect [245, 286] from scratch, based on race tracks [289], log data, or critical real situations. To increase the diversity of the data base, the scenarios were created by multiple individuals. The scenarios are of different duration and describe either a safe situation or a situation injected by one or multiple of different faults (e.g. collision with other vehicles). A visualization of each of the scenarios can be found in Appendix A.5.

The scenario-based tests are performed with both passive and active Supervisor, as described previously (Subsubsection 6.3.1). The principles and intentions of the two strategies are as follows:

Passive Supervisor tests: The Supervisor evaluates each situation in the scenario, but does not switch to the emergency trajectory if the evaluation results unsafe. This allows to evaluate against the ground truth determined for each situation. With this method it is possible to evaluate the performance and algorithmic details of the Supervisor, since in given situations an appropriate safe or unsafe rating has to be made.

Active Supervisor tests: The Supervisor evaluates each situation in the scenario and switches to an emergency trajectory as soon as and as long as there is an unsafe rating. Because of the active intervention and the resulting modification of the scenario, the ground truth loses its validity. In these tests it is checked whether the Supervisor manages to keep the system in a state without an incident caused by the ego-vehicle. The focus is not on performance or algorithmic details, but on overall safety.

In the following, first, the passive Supervisor tests in Subsubsection 6.3.2 and then the active Supervisor tests in Subsubsection 6.3.2 are discussed. Both evaluations use the scenario collection described above.

Passive Supervisor Tests

In the passive Supervisor test, ground truth ratings exist for each scenario, indicating regions where the Supervisor is expected to provide a safe rating and regions where the Supervisor is expected to provide an unsafe rating. The ground truth is drawn as a lighter background and the actual ratings of the passive Supervisor as vertical lines.

A schematic illustration of this evaluation scheme based on the boundary collision example given in Subsubsection 6.3.1 is shown in Figure 6.6. For the sake of better comprehensibility of this illustrative example, only the respective state of the vehicle is rated instead of the trajectory associated with each time-step. In addition, the example does not show the interplay of all SupMods, but only the check for collision with the track boundary. The blue vehicle travels along the indicated path (orange), whereby the vehicle intersects the track boundary during a certain time interval (orange tube around the path overlaps the track boundary). Along this path, there exist expected safety evaluation intervals (ground truth). If the vehicle intersects the track boundary, an unsafe rating is expected to be generated by the Supervisor (gray shaded area in the upper plot half). If the vehicle is sufficiently far away from the boundary (outside the green highlighted area on the track), a safe rating is expected to be generated by the Supervisor (light blue shaded area in the upper plot half). If the vehicle is close to the boundary (intersects the green highlighted area), the situation is safe on paper, but driving close to the boundary of the track (e.g. 1 mm) may be unsafe. For this reason, the Supervisor is allowed to generate unsafe ratings in this case as well. Accordingly, there are areas without safe or unsafe ground truth, which allow any rating of the Supervisor. The actual ratings of the Supervisor are plotted over the ground truth for each time step by black and blue bars, so that a colored bar is created in the overall image (upper plot half). As shown for the evaluation of static collisions with the track boundary in this example, the ground truth intervals are generated in a similar manner for all other aspects and then merged in a single ground truth rating.

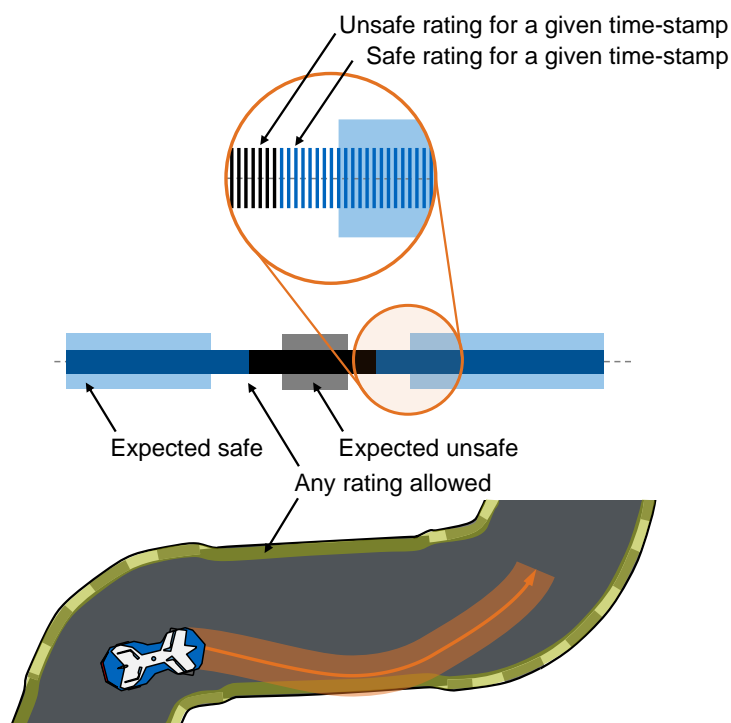


Figure 6.6: Illustrative sketch of the evaluation method based on a point-by-point analysis with regard to track boundary intersections.

An overview of the corresponding passive ratings (i.e., rating but not intervening) and ground truth regions of the implemented 121 scenarios is shown in Figure 6.7.

The generated ratings of the passive Supervisor all fall within the expected ratings for all scenarios. Hence, no false negatives nor false positives occurred in the passive scenario tests. An overview is given in Table 6.3. The number of evaluations without unambiguous ground truth is given in the “any” column of the table. The number of these candidates show a tendency towards more “safe” evaluations, which corresponds to a high availability of the system. However, it has still to be checked in these cases whether critical situations were missed and incorrectly evaluated as safe. This is done by inspecting samples of the ratings that are without ground truth by hand, since a human can take into account the applicable rules. Furthermore, additional test runs with an active Supervisor (i.e. intervening once an unsafe rating occurs), in which no hazards are allowed to occur, demonstrate timely intervention. In the following, an exemplary situation without ground truth for one of the scenarios investigated is presented and analyzed. This is followed by an evaluation of the tests with an active Supervisor in Subsubsection 6.3.2.

Table 6.3: Number of individual ratings received across the 121 evaluated scenarios per binary classification type.

		Expected rating		
		Safe	Unsafe	Any
Observed rating	Safe	12085	0	10382
	Unsafe	0	5449	1096

Figure 6.8 depicts a situation from scenario S43. No ground truth was stored for this specific situation. The reason for this is that the DSS value falls below the defined threshold (Subsubsection 6.3.1), but the vehicles are not on a collision course (no crossing of the trajectories). In order to be able to generate a ground truth here, the formalized rule knowledge would already have

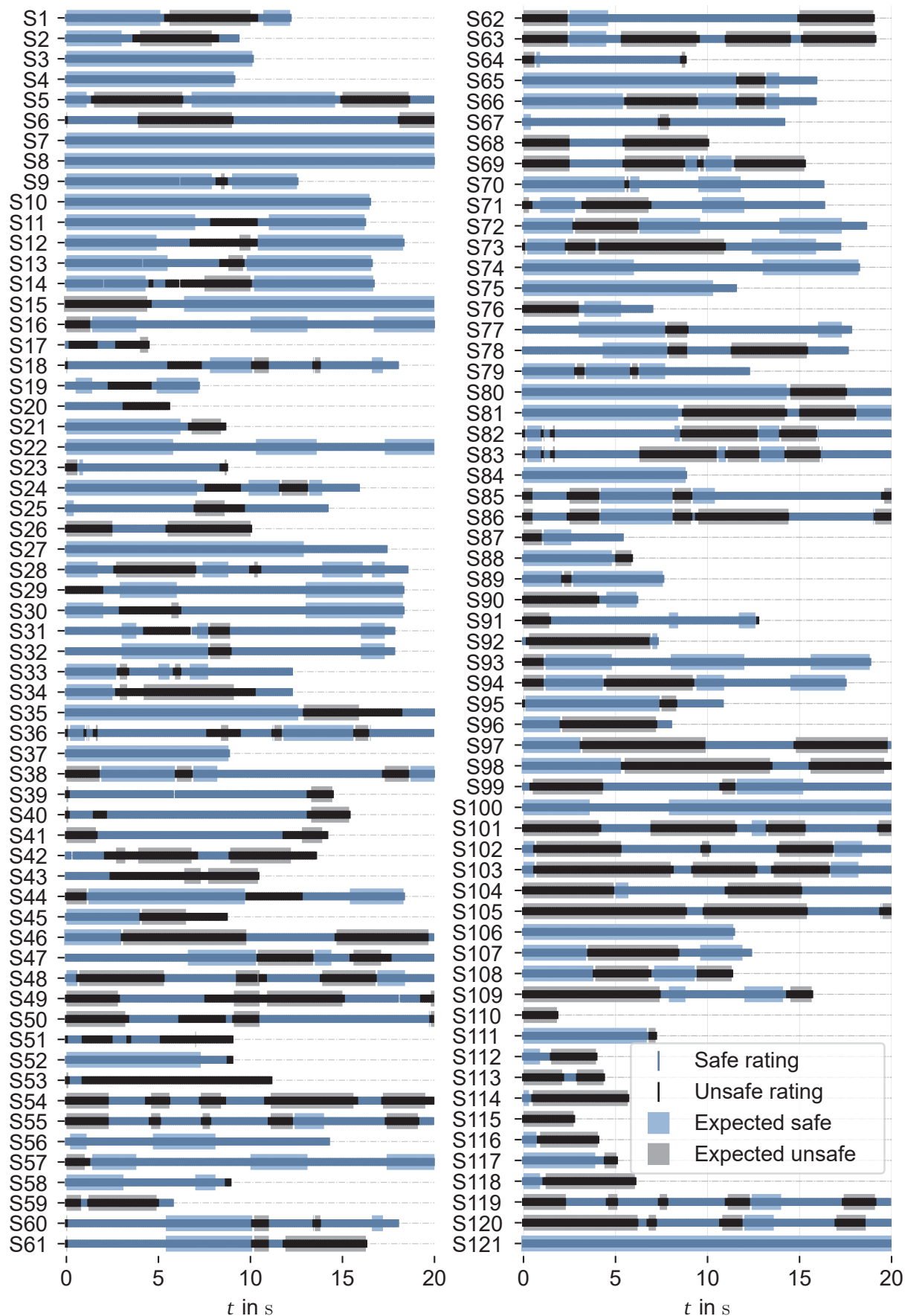


Figure 6.7: Passive Supervisor ratings for the 121 scenarios (S1-S121) examined, each paired with their expected safety rating. Regions where no expected rating is stated can take any Supervisor rating. The plot is cropped to a maximum duration of 20.0 s for better readability.

to be included in the creation and thus contain a duplicate of the SW to be tested, which does not provide any additional validation value [148]. In the situation shown, the driving tube of the emergency trajectory of the ego-vehicle (orange) intersects the reachable set of a TP (green) slightly (red highlights). Thus, after a series of preceding safe evaluations, this situation is among one of the first evaluated as unsafe. The reachable set is not further reduced in this situation, since none of the formalized rules apply. After a manual inspection, the Supervisor's assessment of the situation as unsafe is considered correct, since the ego-vehicle is approaching the inside of the curve too fast to be able to react to possible maneuvers to the inside of the curve by the vehicle in front. In the further course of the scenario, a collision occurs (time step t_2^{veh1} and t_2^{veh2} in Figure 6.8). Similarly, a manual check was performed on samples of other situations without ground truth throughout all scenarios. No irregularities were found.

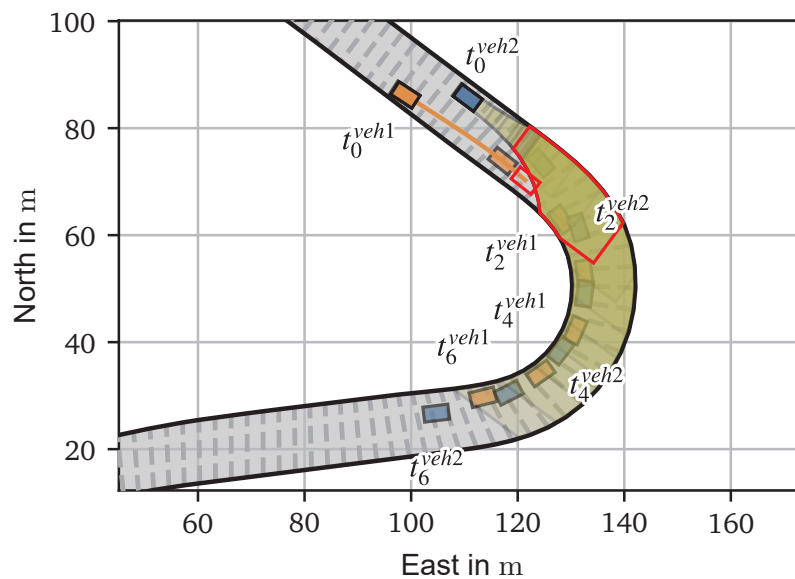


Figure 6.8: Reachable set of a TP (green) and emergency trajectory of the ego-vehicle (orange) at 3.2 s into scenario S43, based on [246]. Future vehicle poses are drawn at 1.0 s increments, while every second instance i is denoted by t_i^{veh1} and t_i^{veh2} for the ego-vehicle and TP, respectively.

Active Supervisor Tests

In active Supervisor tests, it is checked whether the combined system, trajectory planner including the Supervisor, is capable of preventing critical situations. For this purpose, all 121 scenarios examined above are used once more. This time, as soon as the Supervisor detects an unsafe situation, the system switches to the emergency trajectory and follows this trajectory until a safe pair of trajectories is found again. In accordance with the assumptions stated (Section 5.2) and in order to exclude further influences in the evaluation of the module under consideration, an ideal controller is used. The controller follows the trajectory in an ideal manner, which satisfies the objective of validating the planner-Supervisor combination, i.e. to show that this combination always provides sufficiently safe trajectories under given input data. In a real setting, the identified control uncertainties have to be taken into account in the planner and thus also in the Supervisor (as mentioned in Section 5.2, for example, with trajectory tubes by the planner and compliance guarantees by the controller).

A scenario is considered passed if no unsafe situation occurs, i.e. the Supervisor prevents any incident. For repeatability and comparability, the other TPs in the scenario are non-reactive and travel their trajectories in a fixed manner, therefore situations in which the TPs is responsible for a

collision with the ego-vehicle (e.g. running over the stationary ego-vehicle) are excluded from the evaluation. An overview of all scenarios carried out and their evaluation is shown in Figure 6.9. It is important to distinguish here in comparison to Figure 6.7 that in this figure the safety of the situation is depicted instead of the Supervisor ratings. Consequently, the expectation for the scenario whether an incident occurs is drawn as a semitransparent background and the fact whether an incident has occurred for a certain time stamp as vertical lines. Over the entire scenario period, the individual vertical lines in a row form a bar with color-coded evaluations for the corresponding time periods.

The evaluation generated with the active Supervisor is in line with expectations. All observed incidents were not caused by the ego-vehicle following given assumptions and rules. A summary is given in Table 6.4 and causes for occurred incidents are addressed and explained in the following. Since isolated incidents were caused by a TP (e.g. driving through a stationary ego-vehicle) or an unsafe initial state, no relevant false positives nor false negatives were observed. All types of incidents that occurred in Figure 6.9 are listed and briefly explained in the following:

- The TP violates an applicable rule (S13: TP leaves dedicated overtaking zone according to the *Roborace* regulation, S20: TP pushing ego-vehicle off track while driving alongside).
- The TP hits the ego-vehicle from behind (S15, S16, S40 [second incident interval]).
- The initial state of the ego-vehicle is unsafe due to the vehicle footprint intersecting the track boundary. Due to a non-zero initial velocity, the ego-vehicle leaves the boundary on the emergency trajectory and continues safe thereafter (S39, S40 [first incident interval], S53, S109 [first incident interval]).
- Collision at low velocity as the ego-vehicle encounters a simulated crash of two TPs ahead, which do not comply with the assumptions due to backing up/rebounding of the vehicles (S48).
- The provided emergency trajectory holds one time instant slightly exceeding the acceleration limits (S71).
- The initial state of the ego-vehicle is unsafe due to a slight boundary intersection that cannot be avoided due to a high initial velocity (S76, S113, S115, S120).
- The initial state of the ego-vehicle is unsafe due to the vehicle resting outside track. The vehicle remains in this state throughout the scenario due to a low initial velocity and the fact that no safe new trajectory is provided (S91).
- The initial state of the ego-vehicle is unsafe due a trajectory segment violating the turn radius that cannot be avoided due to a high initial velocity (S109 [second incident interval]).

Table 6.4: Number of incidents and safe states observed during active Supervisor simulation across the 121 evaluated scenarios per binary classification type. The false negatives are shown in parentheses, as they were classified as reasonable unsafe states (true negatives) in a subsequent analysis.

		Expected state	
		Safe	Unsafe
Observed state	Safe	27740	0
	Unsafe	(370)	0

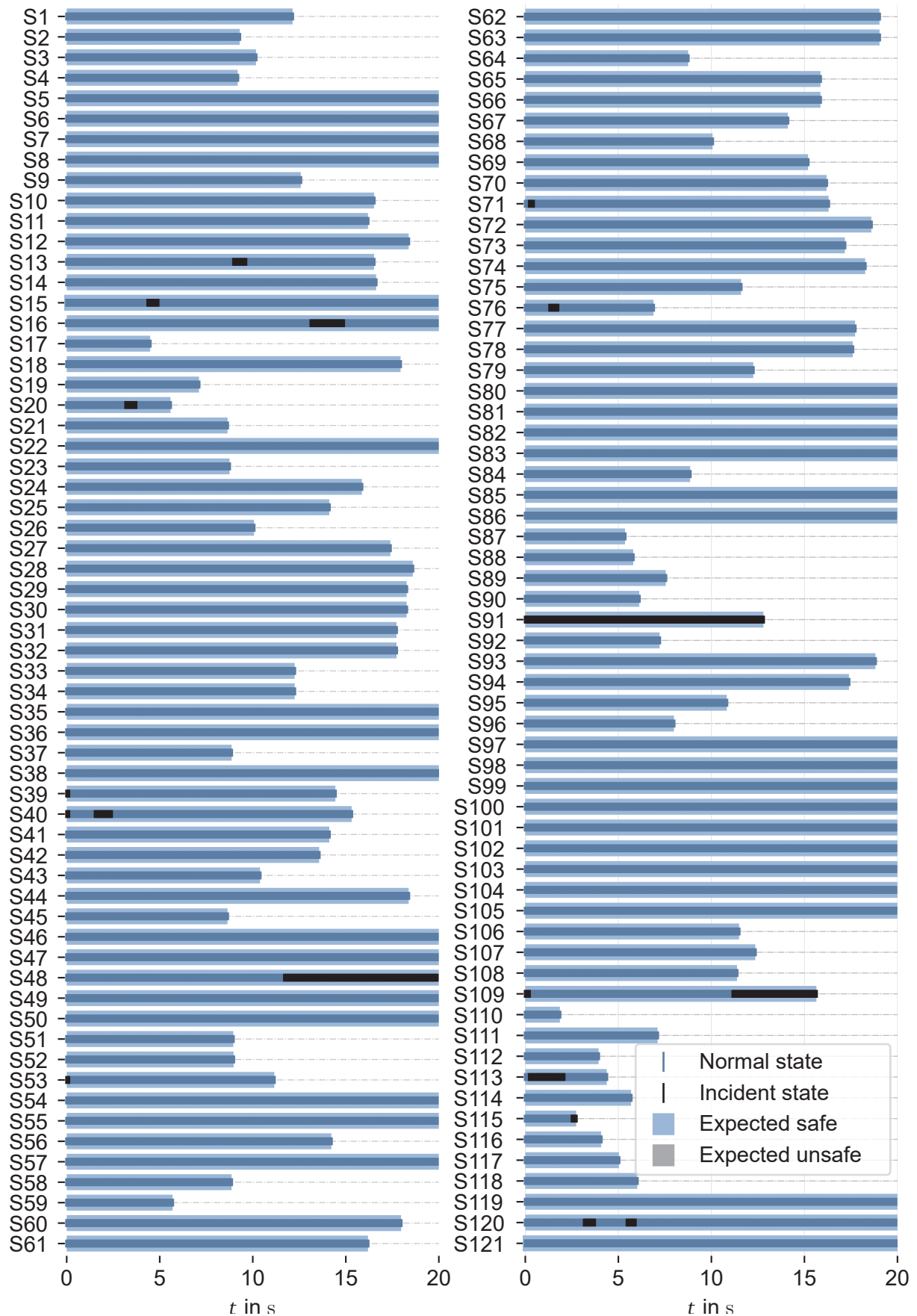


Figure 6.9: Observed safety status of individual time-steps during active Supervisor runs for the 121 scenarios (S1-S121) examined, each paired with their expected safety state (expected throughout safe). The plot is cropped to a maximum duration of 20.0s for better readability.

In the same way as before, one of the scenarios with an active Supervisor is discussed in more detail as an illustrative sample. For direct comparability, the previously discussed scenario S43 is also selected here. The situation at the same point in time as before in the scenario with active Supervisor is shown in Figure 6.10. It can be seen that the TP is still at the same position and has an identical reachable set, but the ego-vehicle is now a bit further back, so that the emergency trajectory does not intersect with its driving tube. Thus, the instant shown is to be considered safe. This is due to the fact that in a previous time step the Supervisor evaluated the emergency trajectory as unsafe due to the intersection with the reachable set of the TP. Consequently, the Supervisor forced a switch to the latest safe emergency trajectory and the ego-vehicle was slowed down until a new safe trajectory was generated. As a result, it can also be seen in the further course of the scenario (plotted vehicle poses) that there is now no collision and the ego-vehicle passes through the corner behind the TP. Accordingly, this scenario with active Supervisor intervention is to be evaluated as passed.

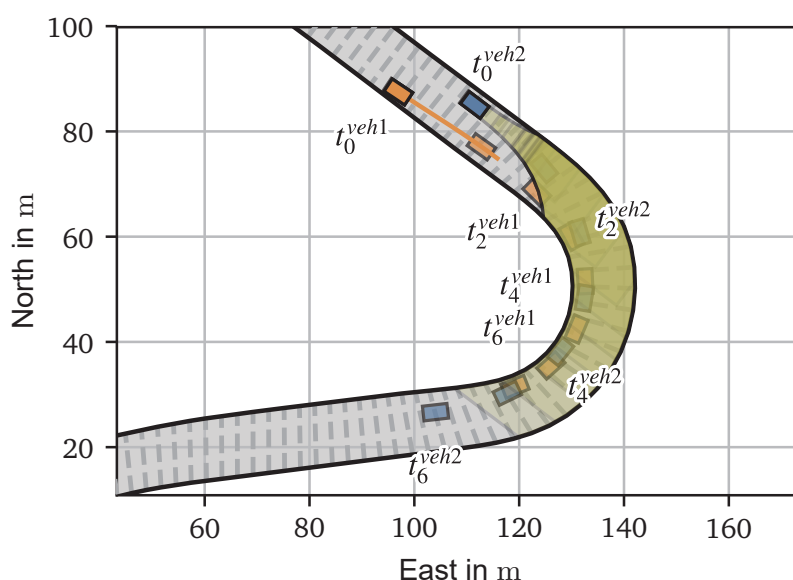


Figure 6.10: Reachable set of a TP (green) and emergency trajectory of the ego-vehicle (orange) at 3.2 s into scenario T43, based on [246]. Future vehicle poses are drawn at 1.0 s increments, while every second instance i is denoted by t_i^{veh1} and t_i^{veh2} for the ego-vehicle and TP, respectively.

6.4 Random-Based Evaluation

Random-based evaluation involves evaluating real test runs in a normal operating environment, i.e. in this case during a race. The objective is to detect false negatives in particular, i.e. to check whether the Supervisor initiates unauthorized interventions in supposedly safe and common situations. For these tests, data from automated race vehicles generated during the *Roborace Season Alpha* [107] are analyzed. In the following, the automated race vehicle (Subsection 6.4.1), the relevant test tracks (Subsection 6.4.2), and finally the evaluation results (Subsection 6.4.3) will be outlined in more detail.

6.4.1 Testbed DevBot by Roborace

Recorded automated race vehicle missions of the trajectory planner presented in Subsection 2.3.1 serve as the basis for the random-based evaluation. These were recorded during automated test drives with the *Roborace DevBot 1.0* (Figure 6.11a) and *DevBot 2.0* (Figure 6.11b). Both vehicles host an electric drive train with 270 kW drive power and are equipped with the same comprehensive sensor technology. This includes six cameras, five LIDAR systems, two radars, two differential global positioning systems, two inertial measure units, and two optical speed sensors.



(a) *DevBot 1.0* during tests on an abandoned airfield in Upper Heyford, United Kingdom.



(b) *DevBot 2.0* during tests on a race track in Monteblanco, Spain.

Figure 6.11: Automated test vehicles used for evaluation: *Roborace DevBot 1.0* and *DevBot 2.0*.

All the SW driving the vehicle was developed by a team of researchers from two institutes at the *Technical University of Munich*. The most relevant aspects, trajectory planner and SW architecture, have already been briefly presented in previous Subsection 2.3.1 and 6.1, respectively. For interested readers, further details (less relevant for the understanding of the following sections) can be found in two overview papers [109, 281].

6.4.2 Test Grounds

The data used for evaluation was obtained on test tracks in the United Kingdom, Spain, Italy, and France. The corresponding track maps are given in Figure 6.12, detailed versions including distance information can be found in Appendix A.4. These track representations, based on two boundaries, serve as a basis for the planner. Accordingly, they are also used in the Supervisor module for OV. The map was created via high-precision DGPS recordings or via simultaneous localization and mapping using LIDAR sensors [245].

The tracks were raced by one vehicle alone (map checkups, performance maximization, time trials) as well as with multiple vehicles (races). As soon as several vehicles were on the track, the *Roborace Season Alpha* rules were applied (Subsection 5.5.1).

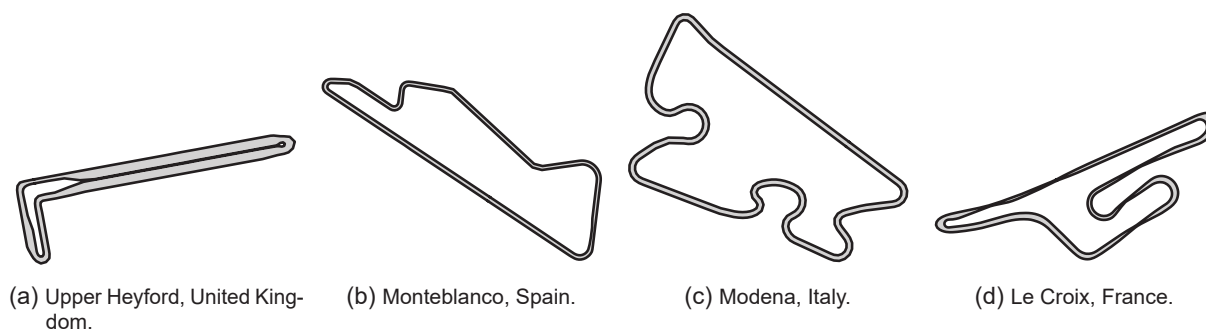


Figure 6.12: Overview of the tracks on which the automated race vehicle was operated for this evaluation. Different scales are applied.

6.4.3 Evaluation Results

Due to the COVID-19 pandemic and strategic decisions to refrain from multi-vehicle operations in the context of *Roborace*, no further actual runs with an active Supervisor took place during this research work. For this reason, an evaluation of real data recorded on the previously described tracks prior to the implementation of the Supervisor is carried out in this section. The actual trajectories and the corresponding environmental data faced on in the real world were logged in their entirety for each time step and are replayed in real time for evaluation. Following the architecture depicted in Section 6.1, this data is fed into the Supervisor. The Supervisor then performs a passive evaluation, i.e., without intervention upon detection of an unsafe situation. The hypothesis for the actual test drives is that the vehicles are always in safe situations and thus primarily safe ratings are to be expected. In this context, a retrospective, passive evaluation is not to be considered as diminishing. Any unsafe ratings that occur are initially classified as false negatives, but are subsequently examined with regard to their justification.

The evaluation takes into account 162.36 km of automated driving data with up to three automated vehicles simultaneously on the track and speeds up to 210 km h^{-1} . The corresponding scores of the Supervisor in the individual runs are displayed in Figure 6.13. The exact numbers of the individual assessments are listed in Table 6.5. It can be seen that, besides to the expectation of primarily “safe” ratings, there were a few isolated “unsafe” readings. These alleged false negatives have been analyzed manually and evaluated with regard to their justifiability (for this reason shown in brackets). Individual illustrative incidents are examined in the following.

Table 6.5: Number of individual ratings received across the 17 evaluated real-world test runs per binary classification type. The false negatives are shown in parentheses, as they were classified as valid ratings (true negatives) in a subsequent analysis.

		Expected rating	
		Safe	Unsafe
Observed rating	Safe	52843	0
	Unsafe	(472)	0

Incident in Montebianco, Spain

In Montebianco, Spain, on the last day of testing in a race with two vehicles (T8), a fault occurred in which the ego-vehicle went off track while following another vehicle shortly before the apex of the turn (Figure 6.14). In the data (Figure 6.14c), it can be observed that the ego-vehicle brakes

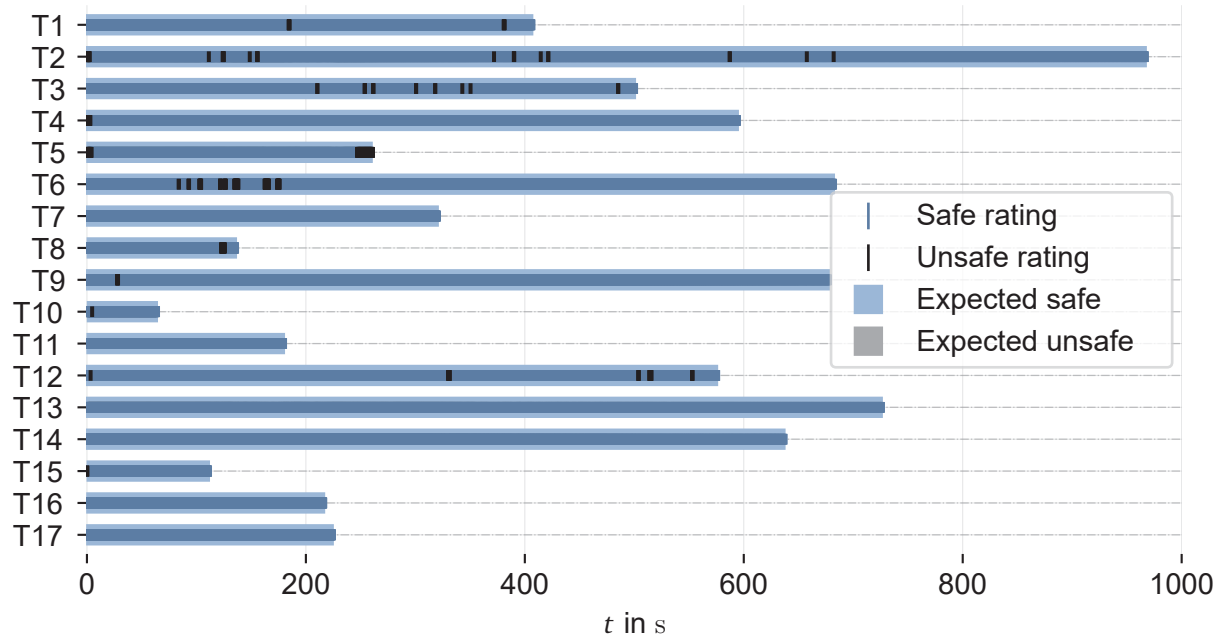


Figure 6.13: Passive Supervisor ratings for 17 examined real-world test runs (T1-T17) each paired with their expected safety rating (expected throughout safe).

for the turn, comes off the brake shortly before the apex, passes through the turn at a constant velocity that is too high, and only when the control error ($t = 5.19\text{ s}$ in Figure 6.14c) is too large an emergency stop is performed with full brake pressure.

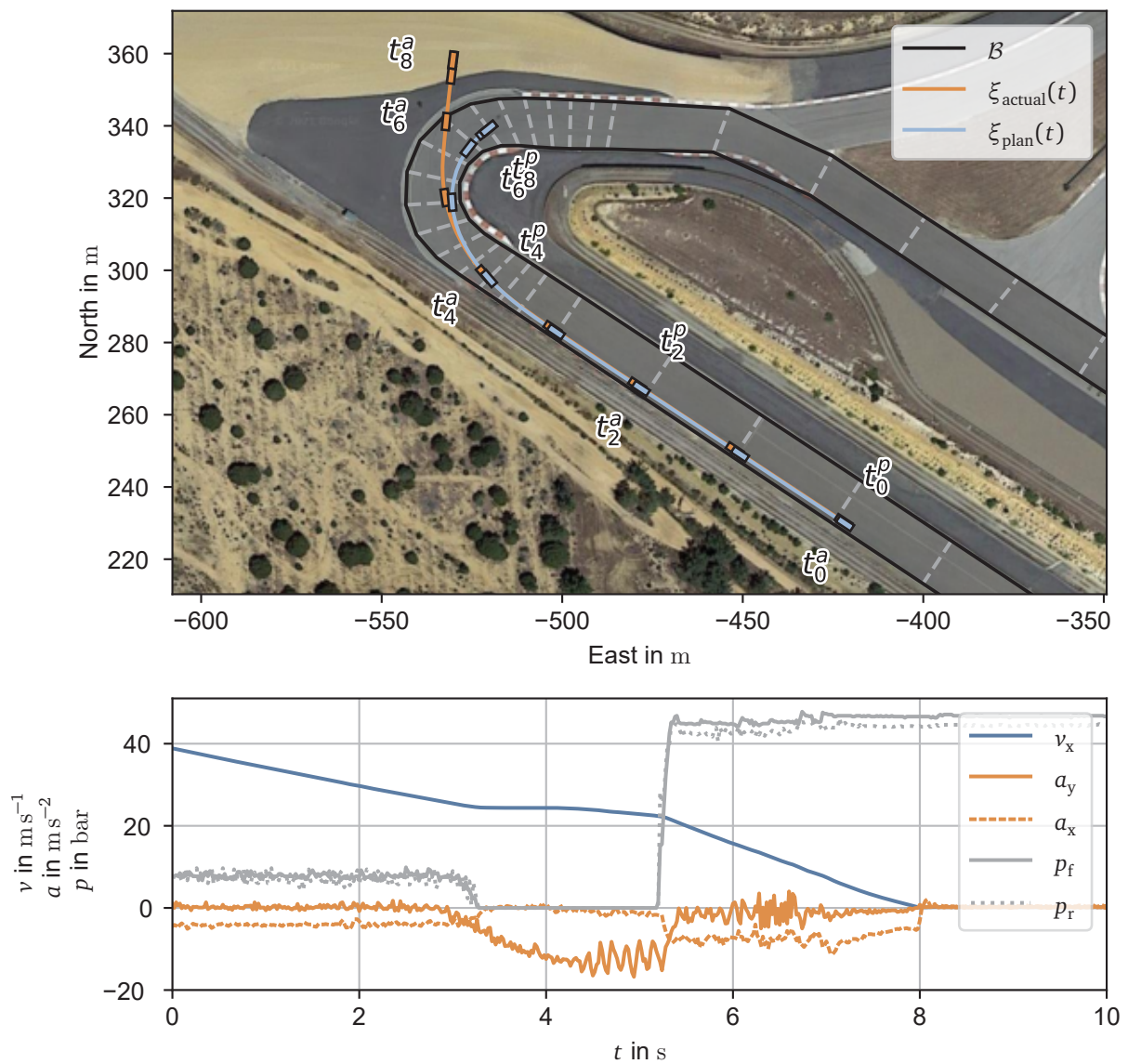
After analyzing the data, it became apparent that the velocity profile was too fast for the upcoming turn and therefore required high lateral forces (Figure 6.15). The cause for this phenomenon was tracking of the vehicle in front. For the applied velocity planner, two optimization problems interacted. One that determines the appropriate feasible cornering velocity profile as well as a tracking controller that regulates the (safety) distance to the vehicle in front. In some situations, when a vehicle was directly ahead in front of a turn, the tracking controller incorrectly gained dominance. This caused the ego-vehicle to maintain the distance instead of properly decelerating for the apex. This occurred mainly when the vehicle in front allowed higher lateral accelerations than the ego-vehicle or the ego-vehicle was in the process of catching up. As a consequence, the velocity of the ego-vehicle at the entry of a turn was in some instances too high to be able to brake sufficiently for the apex of the turn. In these cases, the ego-vehicle decelerated with the maximum remaining longitudinal acceleration potential in front of the turn apex, but as soon as the lateral acceleration reached the total configured acceleration limit the tire-road combination can bear (Figure 6.15b), the ego-vehicle could not apply any further longitudinal deceleration (longitudinal velocity plateau and longitudinal acceleration being zero in the top plot of Figure 6.15a). This systematic error was only noticed when the vehicle went off the track like mentioned in this incident. However, when analyzed with the Supervisor, it is evident that this phenomenon occurred much more frequently when driving through turns behind a lead vehicle (all generated “unsafe” ratings for $t \geq 20.0\text{ s}$ in T1, T2, T3, T6, T9, and T12). In these cases the vehicle did stay on the track, because either the error was still so far in the future of the trajectory that the vehicle re-planned and executed a new valid trajectory in time, or the controller, and the tire-road combination were able to compensate for this error. With the Supervisor in use, the error would have been detected at the first occurrence and could have been addressed adequately. The corresponding “unsafe” evaluations by the Supervisor are therefore justified and is classified as valid.



(a) Superimposed frames at 400 ms intervals taken from a video [290]. The lead vehicle exits the turn safely and the following vehicle leaves the track at the apex of the turn.



(b) Incident race car in the gravel close to the apex of the turn. Track marshals and engineers inspect and prepare for the recovery of the vehicle.



(c) 10 s excerpt of the ego-vehicle from the incident in Montebancho, Spain. The track data (black) as well as the actual position course (orange) recorded via state estimate and the respective intended/planned poses (light blue) are shown above a Google maps satellite image in the upper plot. Vehicle poses are drawn at 1.0 s increments, while every second instance i is denoted by t_i^a and t_i^p for the actual and planned poses, respectively. In the lower plot, the longitudinal speed v_x , lateral (a_y) as well as longitudinal (a_x) acceleration and front (p_f) as well as rear (p_r) brake pressures are given for the same period.

Figure 6.14: Vehicle going off track due to infeasible velocity plan in Montebancho, Spain (test run T8).

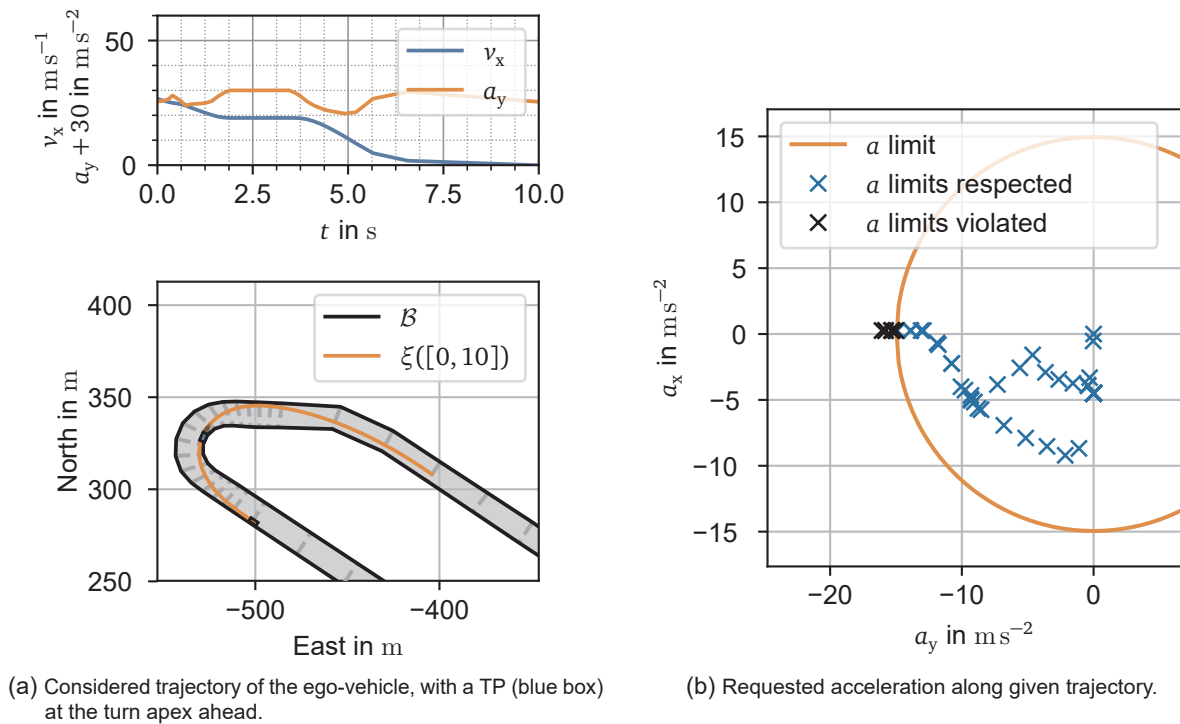


Figure 6.15: Infeasible velocity plan in Monteblanco, Spain (test run T8), shown for time-step $t = 3.0\text{s}$ of Figure 6.14.

Incident in Modena, Italy

The “unsafe” evaluations caused by the Supervisor in test run T5 towards the end (around 250s into the test run) represent a unique situation. At the beginning of a closing cool-down lap following several high-speed laps, the vehicle left the track and collided with a tire pile at a speed of around 100 km h^{-1} (Figure 6.16). The impact ($t = 6.3\text{s}$) corresponded to a load of $8.2g$ (Figure 6.16c). It should be noted, that this figure holds less data with a less frequent sampling compared to the previous evaluation since only data transmitted via telemetry is available, due to corrupted controller logs as a result of the crash. At first glance, the phenomenon—insufficient deceleration before a turn apex—is the same in this case as in the previously described incident. However, the root cause for the incident is different and will be explained below.

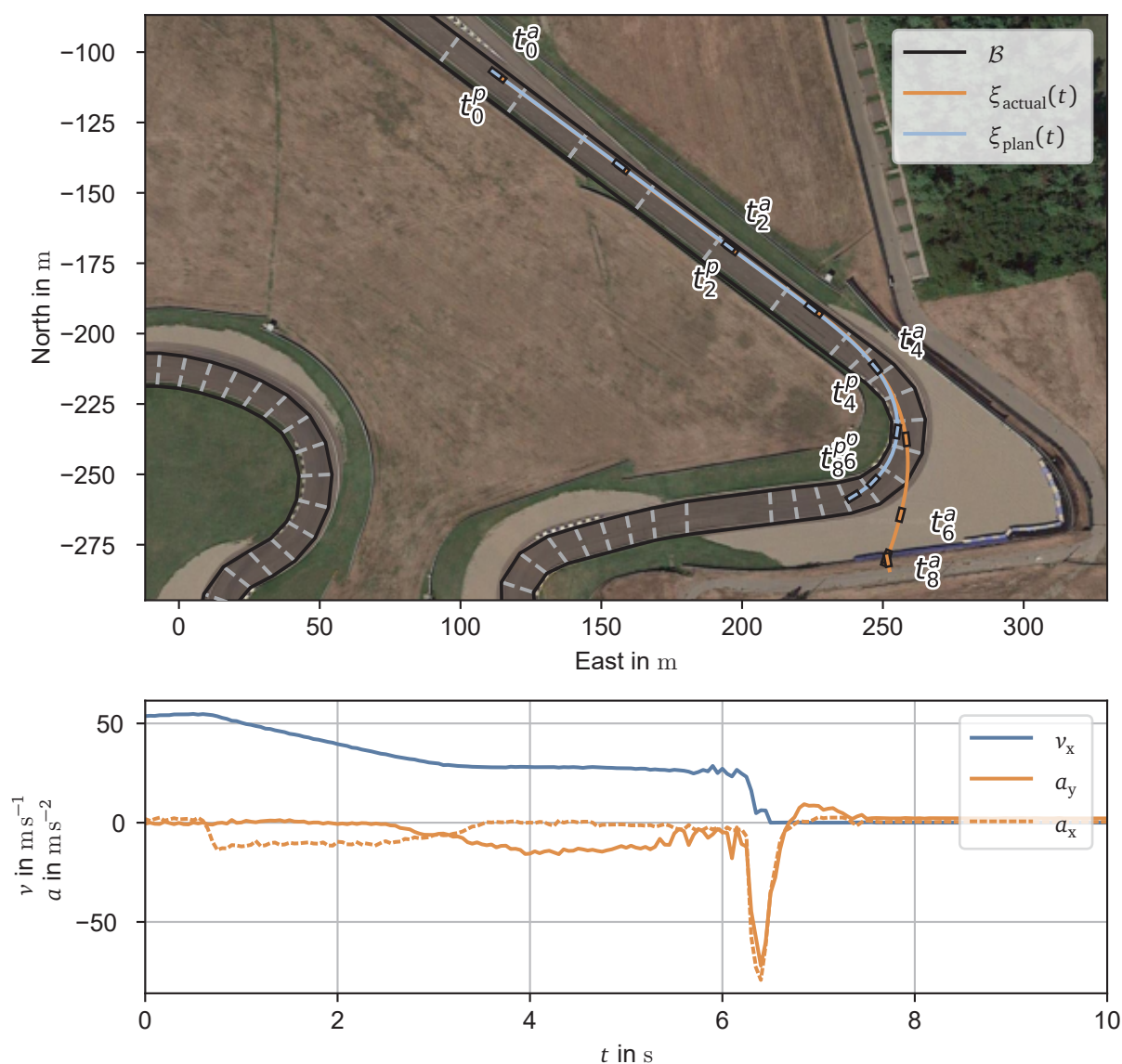
Due to the scheduled cool-down lap aimed at cooling down the braking and actuation system, reduced acceleration limits were set active at the start/finish line just before turn 1 of the race track in Modena, Italy. As a result of this switch and the associated reduced acceleration reserves, the vehicle was no longer able to brake sufficiently before the apex of the upcoming turn. The planner detected this problem and intended to bring the vehicle to a standstill as a consequence. Trajectories that were planned significantly before the incident—in Figure 6.17 exemplary shown for $t = 2.0\text{s}$ of the interval introduced in Figure 6.16c—resulted, as intended, in a standstill. However, the velocity planner also adopted the same reduced acceleration limits, so that even this deceleration trajectory was not able to decelerate the vehicle sufficiently before the apex of the turn. As soon as the lateral forces caused by the turn have exhausted the tire’s entire (reduced) acceleration potential, the planner was no longer allowed to apply any further longitudinal acceleration (velocity plateau in Figure 6.17a). Due to the high lateral forces (Figure 6.17b) resulting from the high velocity, the vehicle left the track and hit the barrier. For interested readers, further details of the incident can be found in a dedicated publication [268]. The Supervisor detects the situation early and would have switched to the last valid emergency trajectory with



(a) Vehicle that has left the track and has come to a stop in the tire pile of the track barrier.



(b) Accident vehicle being recovered by track marshals and engineers using a telescopic forklift.



(c) 10 s excerpt of the ego-vehicle from the incident in Modena, Italy. The track data (black) as well as the actual position course (orange) recorded via state estimate and the respective intended/planned poses (light blue) are shown above a Google maps satellite image in the upper plot. Vehicle poses are drawn at 1.0 s increments, while every second instance i is denoted by t_i^a and t_i^p for the actual and planned poses, respectively. In the lower plot, the longitudinal speed v_x , and lateral (a_y) as well as longitudinal (a_x) acceleration are given for the same period.

Figure 6.16: Vehicle going off track due to infeasible cool-down plan in Modena, Italy (test run T5).

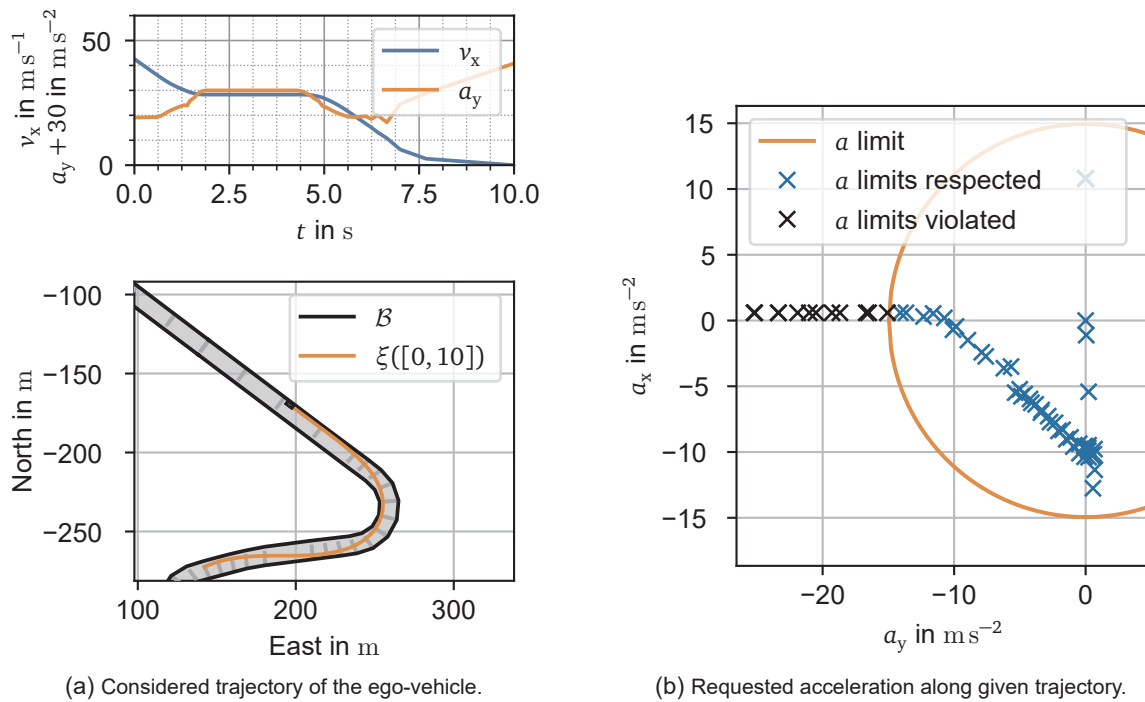


Figure 6.17: Infeasible cool-down plan in Modena, Italy (test run T5), shown for time-step $t = 2.0\text{s}$ of Figure 6.16.

unreduced acceleration limits if it had been deployed during this run. Due to the relevance of this situation, the isolated event of this incident was replicated in scenario S121 (Subsection 6.3.2) and thus also successfully passed the active Supervisor test. In the corresponding scenario, the Supervisor switches to the emergency profile, brakes the vehicle down towards the apex until a new valid trajectory with the reduced acceleration limits is found. Consequently, the “unsafe” ratings by the Supervisor are justified and valid in this case.

Concluding Judgment on the Random-Based Evaluation

The majority of the “unsafe” ratings correspond to the types of incidents explained earlier. In addition, there were isolated events when starting from standstill (all generated “unsafe” ratings for $t < 20.0\text{s}$ in T2, T4, T5, T10, T12, and T15), which were caused by exceeding the motor potential. These were not noticed in the field, because the friction force limit is much higher than the limitation of the motor. Thus, the vehicle accelerated slightly slower than initially intended by the trajectory planner, but no critical situations occurred.

In summary, all ratings of the Supervisor were justified. In addition to the expected safe ratings, the Supervisor also provided occasional unsafe ratings, as suspected. Each of the ratings initially classified as false negatives was examined with regard to its justification and qualified as true negatives. Consequently, the random-based evaluations with real-data tests are to be judged as successful and passed.

It is noted once more that the scope of the selected measures does justice to a proof of concept of the prototype in question for this project, but that significantly more effort is required for an official approval. Related research deals with the question of the required extent of testing that is sufficient for sound approval [167].

In summary, in this chapter, an OV module for the use-case of a trajectory planner of an automated race car was integrated, tested and successfully evaluated. In the next chapter, the results obtained will be reviewed and put in relation to the guiding theses and research questions posed at the beginning of this work. In addition, acquired findings and future work will be derived.

7 Discussion

This section positions the results obtained in Chapter 4, 5, and 6 with respect to their ability to cope with the challenges of approval (Subsection 2.4.1) in Section 7.1 and with respect to the guiding theses and research questions (Chapter 3) in Section 7.2. In addition, an insight on identified findings and an outlook on future extension and application opportunities is given in Section 7.3.

7.1 Review of Posed Safety Standard Limitations

At the beginning of this work, Subsection 2.4.1 highlighted the limits of current automotive standards when faced with complex or learning driving functions. Two core challenges were identified based on the standards itself and related work. In this section, a review is made of whether these two challenges have been adequately addressed by the chosen approach.

7.1.1 Online Adaptation

The first core challenge identified is:

Online adaptation: Systems that continue learning after the validation phase can develop an unsafe behavior (new types of hazards and no specification possible due to training data).

The following explains how this challenge is addressed with the implemented approach. The challenges arising from the characteristics of an AI-based algorithm are intercepted by the encapsulation with the Supervisor. The Supervisor guarantees that, regardless of the underlying system, the generated trajectories comply with all formal criteria for safe behavior (through holistic elaboration). Accordingly, new types of hazards cannot occur under assumptions made. Since the planner used in this use-case does not rely on online learning AI, this aspect is confirmed by the scenario-based validation, where unnatural trajectories and faults are injected and tested.

The lack of specifiability of the underlying system based on training data is also eliminated by the formally well-defined Supervisor. Due to the formal nature and the deterministic behavior of the Supervisor, it can be precisely determined which behavioral characteristics the combined system, black-box planner and Supervisor, can adopt. In conclusion, the challenge discussed here can be addressed by the proposed Supervisor framework.

7.1.2 Complex and Non-transparent

The second core challenge identified is:

Complex or non-transparent: The module cannot be formally approved by an expert (non-transparent functions and standards tailored for manual vehicles).

The following explains how this challenge is addressed with the implemented approach. The main challenge, that the function to be safeguarded is not transparent and too complex for an expert, is significantly mitigated by the explained ASIL decomposition into an ASIL QM (D) for the underlying function and an ASIL D (D) for the Supervisor. For the QM qualification, only basic measures have to be performed and the more complex ASIL D measures are performed on the Supervisor, which is based on transparent and straightforward principles. Accordingly, all of the measures explained in Chapter 5 are based on deterministic and straightforward methods.

Nevertheless, the standards are formally only conditionally or not yet valid for ADSs of SAE level 4+. While the main challenge of the application to such systems is addressed by the use of the Supervisor (as stated above), a formal qualification with all details is neither given in ISO 26262 nor in ISO PAS 21448. In conclusion, the prerequisites for approval in line with the standards are given with the proposed approach, but formal further development of the standards is still pending.

7.2 Review of Posed Guiding Theses and Research Questions

The guiding theses and research questions derived in Section 3.1 are conclusively answered below with regard to the procedure developed and the results shown. In this process, the different guiding theses and related research questions will be reviewed and answered individually. Finally, the research gap derived in Subsection 2.5.3 is revisited in terms of its fulfillment.

7.2.1 Evaluation of Guiding Thesis 1

In the following, the first guiding thesis,

T 1: Complex and frequently changing SW for driving functions of ADSs can be safeguarded, while taking into account applicable standards.

is addressed on the basis of the associated research question:

Q 1: *Which safeguarding method is most promising to address the challenges of approval for complex and learning functions, while taking into account applicable standards?*

The answer to the first research question (Q 1) was initially worked out in Section 3.1 with a structured analysis in the light of related work. An OV approach was found to be the most promising. In order to support the first guiding thesis (T 1), however, it is now necessary to include the findings from the concept and the prototypical implementation.

By conducting scenario-based tests with various error injections, the output of an arbitrary (black-box) SW was emulated. Accordingly, functions that have a complex and frequently changing character as specified in T 1 can thus also be represented within these tests. The results (Chapter 6) confirm that the presented prototype, the Supervisor, is capable of safeguarding trajectories with miscellaneous fault injections (as per the scenario-based testing). Following the preceding findings (Section 7.1), applicable standards are taken into account within this concept. Consequently, guiding thesis T 1 holds and is considered valid.

7.2.2 Evaluation of Guiding Thesis 2

In the following, the second guiding thesis,

T 2: A procedure can be found for the development of an OV method for safeguarding ADS functions, taking into account applicable standards.

is addressed on the basis of the associated research question:

Q 2: *What development steps does an OV method respecting applicable standards have to follow, in order to meet the requirements of a comprehensive safety assessment?*

The second research question (Q 2) is answered by the four-stage method for the development of an OV module presented in Chapter 4. The methodology was established following the principles of applicable standards, in particular ISO 26262 due to the most detailed requirements and specifications at the time of writing. The method presented follows the V-model on which ISO 26262 is based. For a comprehensive safety assessment, the first step of the procedure, the holistic requirements analysis, is essential. To support this goal, a structured method for the development of safety-relevant criteria is presented (Section 5.1).

The prototype implementation of the Supervisor (Chapter 5) demonstrates that it is possible to find a procedure for the development of a method that fulfills the safeguarding of ADS functions in line with the standards. The focus in this guiding thesis (T 2) is primarily on the ability of finding a valid development process, whereas the third guiding thesis (T 3) examines the effectiveness of the implemented function. Consequently, guiding thesis T 2 holds and is considered valid.

7.2.3 Evaluation of Guiding Thesis 3

In the following, the third guiding thesis,

T 3: A prototypical implementation of an OV for the trajectory planner of a real-world automated race vehicle demonstrates the viability and serves as proof of concept.

is addressed on the basis of the associated research questions:

Q 3: *What are the specifics of a prototypical implementation of the derived generic OV method targeting the holistic safety assessment of a trajectory planner?*

Q 4: *What does the validation of a prototypical OV method involve in order to meet the purpose of a proof of concept for the overall method?*

The answer to the third research question (Q 3) is provided by the detailed implementation of an OV module for the trajectory planner of an automated race vehicle in Chapter 5. In the course of the implementation, the requirements for a safe trajectory were elaborated in a dedicated manner using a holistic method. These include, for example, the avoidance of physical contact with static and dynamic entities in the environment as well as the consideration of the friction potential between tires and track surface. Subsequently, after defining the OV architecture, evaluation metrics were developed for all these criteria. For example, temporal logic was utilized to establish reduced reachable sets for the detection of any potential collision with TPs. Starting from a safe initial state (standstill) and based on the individual results obtained with the evaluation metrics, a safe trajectory is guaranteed at any time. The procedure for selecting such a trajectory is explained in Section 5.6.

The fourth research question (Q 4) is answered primarily by means of a comprehensive evaluation of the implemented Supervisor in Chapter 6. According to the recommendations in ISO 26262,

a scenario-based as well as random-based evaluation was carried out. In the course of the prototypical implementation, 121 scenarios were realized and evaluated. The evaluation was carried out in passive, i.e. non-intervening mode against deterministically determined ground truth, as well as with active intervening Supervisor, where no hazards may occur. All scenario-based results were in line with expectations. The random-based evaluation used 162.36 km real-world data of automated driving. Contrary to the initial expectation of exclusively safe evaluations, isolated evaluations resulted in an unsafe rating. However, it could be shown that the unsafe evaluations made by the Supervisor are justified and are traced back to faults in the planning task. This extent reflects an adequate effort for a prototypical implementation, but requires significant increase for official approval. Amersbach [291] investigates a functional decomposition approach aimed at reducing the effort required for safety validation. Corresponding concepts could investigate validation exclusively for the planner and the Supervisor in order to reduce the validation effort.

By combining the findings from the two research questions above, it is concluded that a specific implementation for the OV of trajectories was implemented, which did not show any shortcomings in the course of the validation. Accordingly, the implementation of the Supervisor is considered a successful proof of concept. Consequently, guiding thesis T 3 holds and is considered valid.

7.2.4 Evaluation of the Research Gap

In this subsection, the Table 2.1 developed in Subsection 2.5.3 is revisited for its fulfillment given the research gap identified. In the analysis, it was found that none of related work adequately addresses all four of the assessment criteria identified (described in Table A.1). The goal of this work was set to adequately investigate (☉ / ●) the four assessment criteria.

The scores obtained for this work are shown in Table 7.1. In the following, the individual assessment scores are briefly explained:

Approval/standards: According to the explanations in Section 7.1, core challenges were identified and examined with regard to approval and applicable standards.

Holistic approach: Within the framework of this work, a systematic approach was developed to achieve holistic coverage of all criteria necessary for a safe operation (Section 5.1).

Degree of maturity: In this work, a fully implemented and real-time capable prototype realization was investigated.

Extent of testing: The implemented prototype was evaluated using data from an automated real vehicle.

7.3 Findings and Future Work

In this section, findings and future work are highlighted. In terms of findings, the focus is on the impact of the quality of the emergency trajectories (Subsection 7.3.1) and the benefits resulting from using an OV module during the development phase (Subsection 7.3.2). In the domain of future work, the extension to road traffic (Subsection 7.3.3), the extension to modules other than the planning module (Subsection 7.3.4), and safeguarding of the overall SW (Subsection 7.3.5) are discussed.

Table 7.1: Evaluation of the four specified criteria (first column) with the OV method demonstrated in this work. The resulting rating is indicated via Harvey Balls (○ – not addressed to ● – comprehensively addressed) along with the brightness of the color in the cell (from light to dark). Details on the rating metric can be found in Table A.1.

	Proposed method / prototype
Approval / standards	●
Holistic approach	●
Degree of maturity	●
Extent of testing	●/○

7.3.1 Influence of Emergency Trajectory Planning Performance

During the evaluation of the tests, it was noticed that the choice of the emergency trajectory of the ego-vehicle has an essential influence on the availability and, in a race, on the competitive behavior of the vehicle. This fact is explained and discussed on the basis of an exemplary situation (Figure 7.1) that occurred in Upper Heyford, England.

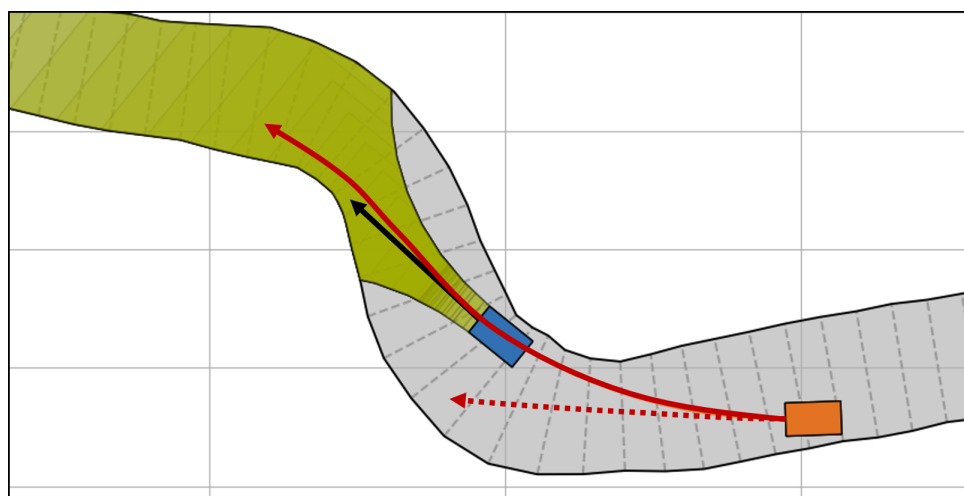


Figure 7.1: Sketch based on a situation taken from a log of a test run in Upper Heyford, England. The current planned emergency trajectory of the following ego-vehicle (orange) is sketched in solid red. A much more efficient one (with earlier standstill) is shown in dashed red.

In the SW stack used here, the emergency trajectory was planned along the path of the performance trajectory. For this purpose, instead of a velocity profile as fast as possible, a deceleration as strong as possible until standstill is planned along the identical support points. The red solid line in Figure 7.1 represents the sketch of such a trajectory for the ego-vehicle. However, the problem with this curved trajectory is that a large portion of the friction potential is used by the lateral force, leaving little acceleration potential in the longitudinal direction for decelerating the vehicle. In addition, a racing vehicle always drives as fast as possible and close to the friction limit, so that in a combination of curves close to the apex, lateral acceleration usually accounts for the entire acceleration share. These conditions result in a long braking distance.

Since the physically shortest possible braking distance is taken into account for the vehicle in front by using the reachable sets, a short braking distance is achieved. A possible variant is sketched with the black arrow. With this combination, the Supervisor detects a possible collision in curves at an early stage (at a greater distance) and decelerates the ego-vehicle accordingly. In this way, the ego-vehicle falls behind in curves by guaranteeing a safe backup plan via the (long) emergency trajectory. In the example shown here, the emergency trajectory (red) already overlaps the reachable set (green) significantly. Here, the Supervisor would have intervened much earlier and let the ego-vehicle fall behind in order to always ensure a safe emergency plan.

If the emergency trajectory was selected as intelligently as possible, i.e. if it tried to come to a standstill as early as possible, the ego-vehicle could follow the vehicle ahead more closely with a safe backup plan. This would result in a more competitive race vehicle that drops back less in curves. A possible trajectory with high longitudinal deceleration due to low lateral acceleration is shown in dashed red. In this case, the emergency trajectory does not intersect the reachable set of the vehicle ahead and the ego vehicle could follow the other vehicle as shown, unlike in the situation described before.

In a similar manner, for overtaking maneuvers in accordance with the regulations implemented here, the emergency trajectory has also to be as effective as possible. In accordance with the Formula 1 driving alongside rule, a TP is not allowed to force the ego-vehicle off the track as soon as the vehicles are side by side. However, in order to approach this area, the ego-vehicle must always be able to come to a stop behind the vehicle in all extreme cases (e.g. sudden brake maneuver of the TP). For this purpose, the difference in speed between the two vehicles must not be too large (e.g. a slow vehicle must not be overtaken at high speed), and maximum deceleration and swerving to the side away from the vehicle are beneficial.

Consequently, regardless of the quality of the emergency trajectory, a safe behavior is guaranteed, but a more competitive behavior and a higher availability can be achieved by an intelligent choice of the emergency trajectory. The problem of finding an adequate emergency trajectory is a separate field of research and investigates dedicated approaches [265, 292, 293].

7.3.2 Added Value During the Development Phase

In addition to the purpose of safeguarding complex and learning functions, OV has also proven helpful in the development phase. The verification tool identifies and localizes possible planning errors on the fly.

The identification of errors is particularly helpful when long tests are executed in which only individual planning steps contain unsafe situations. If errors occur only for a short time, a human easily overlooks these errors. In this case the Supervisor was able to help report and illustrate the situations. Likewise, during Hardware-in-the-Loop (HiL) tests with the own SW deployed on multiple vehicles, an incident can easily be overlooked on one of the vehicles, since the attention is usually focused on one or at most two vehicles. Any potential issues with the remaining vehicles can be highlighted by the Supervisor and prepared for the human to analyze in more detail.

In cases where errors are not visually apparent to humans (e.g. acceleration violations as faced in Subsection 6.4.3), the chance of detecting such problems is only possible with the most thorough and comprehensive human inspection. Here, as well, the Supervisor helps to point out possible problems at an early stage, even if in the specific situation they have not resulted in a hazardous situation. In the case of purely machine-based tests, which, for example, automatically check

the performance (e.g., lap or computing times) of the code at every stage of development via continuous integration, the Supervisor can be used to test safety aspects in addition.

During the SW development phase for the Indy Autonomous Challenge [294, 295]—won by the team of the *Technical University of Munich* [296–300], the Supervisor was used in automated tests as well as during test runs in Software-in-the-Loop (SiL), HiL, and real world. This revealed various planning incidents that were not noticed outright by humans. As an example, Figure 7.2a shows a situation in which a minor side collision occurred during a HiL test involving four vehicles. The Supervisor triggered via the guaranteed occupied sets and marked the affected instants accordingly. In another exemplary case, an off-track trajectory was planned for a few instances during a HiL run at the Indianapolis Motor Speedway (Figure 7.2b). Since this plan occurred only for a short time and was back on the track in time before the vehicle passed through the section, the error was only noticed by the Supervisor.

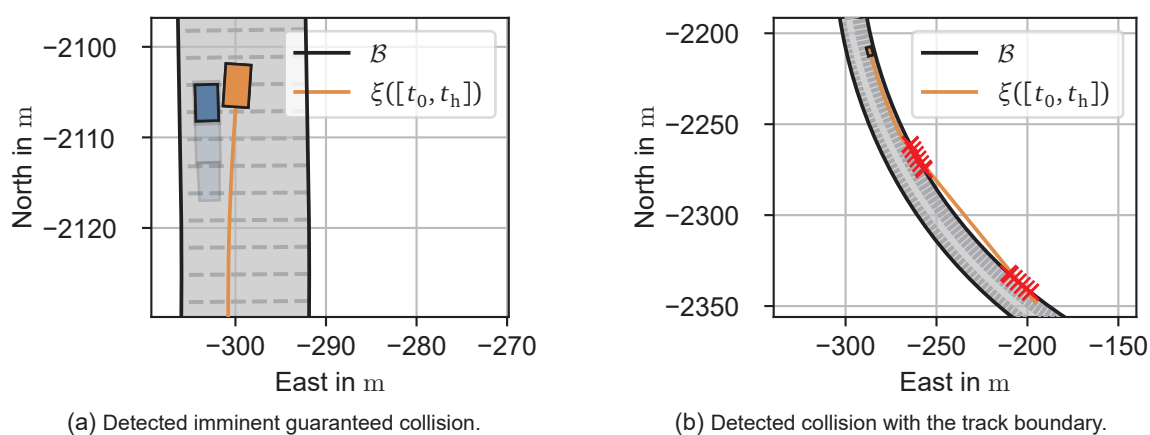


Figure 7.2: Incidents exposed by the Supervisor during simulative testing for the Indy Autonomous Challenge on the digital twin of the Indianapolis Motor Speedway.

In summary, the Supervisor helps in the development phase to identify flaws in the planner development at an early stage. In this way, these errors can be addressed immediately and do not reach the real vehicle undetected. In this way, the valuable test time on the track can be used more efficiently and the likelihood for unsafe or even destructive situations to occur can be reduced.

7.3.3 Extension to Road Traffic

This work deals with the OV of trajectories, which is implemented and evaluated on the use-case of a race vehicle. Future work deals with the adaptation to regular road traffic.

In general, all of the physical boundary conditions discussed in this work required for a safe trajectory are equally valid for a vehicle in road traffic. An adaptation is primarily only necessary for the formalized regulations that are used for the reduced reachable sets. This requires the implementation of applicable road traffic regulations instead of applicable race regulations.

In this context, a challenge is the formalization of all applicable regulations. During the formalization of the illustrative regulations for the race scenario, it has already become apparent that especially vague regulations, which do not contain any fixed or detailed specifications, are not trivial to formalize. Specifications like “from the approach to the corner” [271]-§2b allow different interpretations in the formalization. Here it is necessary to incorporate past race control decisions. Nevertheless, there will not always be congruence between race control decisions and formalized

rules. In the same manner, there are also vague rules in the road traffic regulations. In this case, past court decisions can be used to achieve more specific rules that can be formalized. Such a procedure has already been used by Maierhofer et al. [225] concerning German traffic rules on the Autobahn. After implementation of the rules for road traffic, the Supervisor can be used without further major adaptations according to current findings.

7.3.4 Extension to Other Software Modules in the Driving Stack

The generic method for the development of an OV module (Chapter 4) has been implemented in this work for the planning module of an ADS. Future work is to investigate to what extent this method can be applied to other modules of the automated driving pipeline.

Following the automated driving pipeline towards the actuators, this is possible without further restrictions according to current knowledge, since deterministic and formal safety criteria can be found for the OV of the control module, for instance. In this way, AI-based control approaches can also be safeguarded. Only in terms of guaranteeing a safe state at all times, different options (e.g. validation of a guaranteed controllable driving tube) have to be evaluated.

Following the automated driving pipeline towards the sensors, the application of the OV method is much more complex. Since currently only AI-based methods exist to translate camera data into machine interpretable parameters, verification with deterministic and formal metrics is not necessarily possible. In this case, it must be investigated to what extent safety guarantees can be provided via redundancies and combination of different sensor modalities.

7.3.5 Safeguarding of the Overall Software

In the course of this work, it was investigated how a single module of the autonomous driving stack can be safeguarded. However, it is not necessarily clear to what extent this affects the safety of the overall SW.

In this context, the question arises of how a safeguarded module behaves with faulty/unsafe input data. For instance, in the use-case shown for safeguarding a trajectory planner, faulty perception data (e.g., a missing object in the object list) would be provided. Obviously, the safeguarded module (in this case, trajectory planning module with Supervisor) may generate erroneous data in such a scenario as well. However, as stated in Section 5.2, only safeguarded individual modules in the entire pipeline are assumed. Accordingly, the online verification serves exclusively to check whether the generated result of the module is valid under the given input data. It verifies whether the system correctly processes the given input data according to a simple/stringent specification and whether it generates a valid solution given this input data constellation. In this way, it is now possible to establish clear and guaranteed functional specifications for complex and learning functions.

However, assuming that all modules are successfully safeguarded individually, it cannot be guaranteed that the entire system is necessarily safe. The interaction of the individual (safe) modules can still lead to unsafe situations. The entire pipeline must therefore be validated with extensive tests (Subsection 2.4.2). The scope required to make a final statement here is not yet feasible at the present time and is the subject of ongoing debate in related work [19, 167, 169]. As mentioned before, Amersbach [291] investigates methods to reduce the validation effort in this area. Nevertheless, a profound investigation of the interrelations and influence with other modules is yet to be done and is part of future work.

8 Summary and Conclusion

This thesis presents a methodology for safeguarding complex and learning driving functions by guaranteeing a safe state at any time based on online verification. With this method, it is now possible to safeguard such algorithms in accordance with existing standards. The methodology was developed generically based on related work and implemented for the trajectory planner of an automated race vehicle. The evaluation of the implemented prototype demonstrated its effectiveness.

With increasing deployment rates of advanced driver assistance systems, the number of accidents has decreased, and so have the fatalities. Many people associate increasing automation with Vision Zero, the goal of zero accident fatalities with full automation. However, on the way to full automation, current safety standards are reaching their limits as a result of increasingly complex and online-learning algorithms. An initial literature review identified online monitoring methods as most promising for investigation.

In the related work section, relevant standards were outlined and their shortcomings with respect to modern algorithms were highlighted. Subsequently, relevant work in the field of online monitoring was presented and evaluated with regard to its addressing of safety approval, holistic coverage, degree of maturity, and extent of evaluation. Online verification was found to be the most promising. In a subsequent discussion of related work, research questions were derived in accordance with guiding theses, which led to the presentation of the idea of this thesis and its structure.

Before the implementation of a detailed use-case, a generic method for the development of an online verification module was developed and presented. The method takes into account applicable standards and therefore also proceeds in accordance with the V-model of ISO 26262. Four central development stages are worked through. In the first stage, the requirements for a safe operation as well as for the online verification module itself have to be identified. In the second stage, the architecture and assessment metrics are defined in line with the identified requirements in order to be able to classify the input data as safe or unsafe. In the third stage, these findings are integrated into the overall software and tested against the requirements. In the fourth stage, the implemented Supervisor is evaluated.

Following the elaborated generic development method, an online verification module—the Supervisor—was implemented for the trajectory planner of an automated race vehicle. For this purpose, the elaborated development steps were individually worked through one by one. The first step was to derive a holistic list of criteria that a safe trajectory must meet, using a developed structured approach. Among the derived criteria is, for example, the requirement of avoiding contact with static and dynamic objects in the environment or the proper consideration of the friction between track surface and tires at any time. The Supervisor was integrated following the principle of ASIL decomposition in such a way that it verifies the trajectories generated by the planner (performance and emergency trajectory) against its input data (e.g., object list and environment map). Based on this architecture and for each criterion of the list of requirements

for a safe trajectory, an assessment metric was developed. These metrics are of deterministic and therefore safeguardable nature. A noteworthy example are the rule-based reachable sets developed in this course for the check of guaranteed collision absence in compliance with applicable rules. Based on the assessment of the trajectories with all assessment metrics, a trajectory is always selected in such a way that an emergency plan transferring the vehicle to a safe state exists.

Towards the evaluation of the developed prototype, the Supervisor was first integrated into the software stack developed by the team of the *Technical University of Munich* for the *Roborace* racing series. The trajectory planner developed in the course of this work was briefly introduced in the related work section and represents the central module to be safeguarded. Following the integration, dedicated tests were carried out according to the requirements defined at the beginning. Finally, a two-fold evaluation was carried out. On the one hand, scenario-based tests with fault injection were performed. This involved examining 121 different scenarios with a passive, i.e. non-intervening, Supervisor against a deterministically generated ground truth. No abnormalities were found. However, since the ground truth cannot specify an expected evaluation for every situation, all scenarios were additionally run through with an active Supervisor, in which no hazardous situation caused by the ego-vehicle occurred. On the other hand, data from various real-world automated race drives were evaluated. Here, a safe rating was expected throughout the runs. Unsafe evaluations were checked in detail for their justification. In the process, all ratings were found to be valid. Accordingly, the Supervisor prototype is considered a successful implementation.

The presented method was validated by a prototype and fulfilled the purpose of online verification of trajectories. In retrospective examination of the research questions and guiding theses, it was concluded on the basis of the prototype that the online verification approach is suitable for safeguarding complex and learning algorithms in accordance with applicable standards. Nevertheless, an expansion of safety standards beyond SAE level 0-3 systems towards formal eligibility for automated driving systems (ADS) is still ongoing/pending. The importance of choosing emergency trajectories as wisely as possible has been emphasized for availability and competitive behavior. In addition, online verification is suitable as a tool for efficient improvements during the development phase. Looking into the future, the approach offers promising transferability capabilities to road traffic, with the main challenge being the formalization of applicable traffic rules. The transferability of the approach to other software modules besides the planning module revealed a basic applicability as well as open challenges.

List of Figures

Figure 1.1:	Schematic illustration of differences in safety challenges for the ego-vehicle (blue) between unknown and known behavior of a TP (gray).	3
Figure 2.1:	Subdivision and classification of terms in the field of AI, based on [22, p. 6]. Popular methods are assigned to the clusters, but are not explained any further.	6
Figure 2.2:	Schematic illustration of the terms scene, situation, and scenario.	7
Figure 2.3:	Simplified schematic illustration of the difference between verification and validation, based on [27]. Here shown for a single-stage development process. With multi-layer development processes (e.g. in accordance with the V-model) the verification can also take place between different layers against the respective requirement.	8
Figure 2.4:	Basic automated driving architecture.	8
Figure 2.5:	Flowchart and lattice visualization of the developed graph-based trajectory planner [71].	11
Figure 2.6:	Overview of the three types of safety and security with associated standards. The meaning of the individual terms is illustrated by a symbolic system that transitions into a hazard. In this context, a possible cause of error and its location is represented by an orange lightning bolt.	14
Figure 2.7:	Clustered overview of existing safety assessment approaches.	17
Figure 2.8:	Subdivision of scenarios/situations with respect to distinct characteristics.	18
Figure 2.9:	Illustrative sketch of underlying principles for the maneuver prediction based risk assessment approach with the ego-vehicle in blue and TP in gray.	20
Figure 2.10:	Illustrative sketch of underlying principles for two stochastic occupation based online risk assessment approaches with the ego-vehicle in blue and TP in gray.	21
Figure 2.11:	Illustrative sketch of underlying principles for the motion modeling (Markov chains) based approach with the ego-vehicle in blue and TP in gray.	22
Figure 2.12:	Illustrative sketch displaying the underlying principles of two OV approaches with the ego-vehicle in blue and TP in gray.	24
Figure 3.1:	Structure of the thesis. Circles with “Q” indicate the coverage of the respective research question in the corresponding chapter, circles with “P” indicate an associated first author publication with the following mapping: P1=[32], P2=[71], P3=[244], P4=[245], P5=[246].	30
Figure 4.1:	Generic method for the development of an OV module, the Supervisor (S), structured into four stages (1-4 – indexed from left to right), each comprising up to three steps (A, B, C – indexed from top to bottom), based on [244]. Each step can be indexed with a short notation, for example S-2-C refers to the “Classification function” step of the “Implementation” stage.	31

Figure 4.2:	ASIL decomposition [23] can be used to lower the requirements for the system if another system with high requirements is monitoring it. This scheme is shown here for a combination of a system with ASIL QM (D) and the Supervisor with ASIL D (D), based on [244].	33
Figure 4.3:	Fictional SW diagram with various modules including possible start and end points for an OV deployment.	35
Figure 5.1:	System model and its interfaces based on identified subsystems, based on [244]. Solid lines indicate a physical interaction and dashed lines an information flow. Directed arrows imply a physical interaction initiated by the subsystem at the tail.	41
Figure 5.2:	Supervisor architecture for the use-case of trajectory OV. The planning module is assumed to be not ASIL-capable (ISO 26262-1:2018, 3.2 [23]), but by integrating the Supervisor, the overall system can be approved. Unsafe behavior in the planning module, for example due to non-transparent or online learning methods, is detected and intercepted by the Supervisor. The orange dashed lines indicate the distinct interfaces with the rest of the SW stack.	42
Figure 5.3:	Proposed framework of the Supervisor, based on [246]. The input data (map, object list, performance trajectory, and emergency trajectory) is processed in dedicated SupMods. Based on the individual rating, a safe trajectory is selected and sent to the controller.	43
Figure 5.4:	Different approaches for an approximation of the reachable set for a vehicle (orange rectangle) with initial velocity $v_0 = 25 \text{ m s}^{-1}$ and time horizon of $t = 2 \text{ s}$ at a separation of 0.25 s .	46
Figure 5.5:	Illustration of different approximation methods for collision detection with the track (gray) boundaries (black). The trajectory of the vehicle is indicated by the orange line. Vehicle poses are depicted with a black frame at a spacing of 400 ms . A tube around the trajectory with the width of the vehicle footprint is shown in orange. A tube around the trajectory with the width of the vehicle footprint diagonal is shown in blue. A frame on the upper left shows a magnification of one false positive detection where the tube with the vehicle diagonal (blue) intersects the track boundary, while the real vehicle remains within the track boundary. The lower right frame displays a false negative detection, in which the tube with the vehicle width (orange) does not detect a boundary collision, while the vehicle slightly intersects the boundary.	48
Figure 5.6:	Visualization of simple models (orange, blue) that show the limits of the forces that a tire-road combination can withstand. The actual resultant force at an operating point (black dot) results from a lateral (centrifugal force) and a longitudinal (acceleration) component, shown here (green) as an example for deceleration in a right-hand turn.	50
Figure 5.7:	Exemplary trajectory with evaluated acceleration request for approaching a turn.	51
Figure 5.8:	Reachable sets without rule restrictions would not allow overtaking each other, as the sets soon occupy the entire width of the track. The reachable set (green) of the TP (blue) intersects the poses along the emergency trajectory (orange line) of the ego-vehicle (orange) at a certain point in time (red).	55

Figure 5.9:	Three time instances of an overtaking maneuver of the ego-vehicle (orange) to pass a TP (blue) with its reachable set (green). Guaranteeing safe conditions without any rule-knowledge (top row), the ego-vehicle cannot complete the overtaking maneuver because the emergency trajectory (orange line) intersects the reachable set (red marking). Under applicable regulations (bottom row), the ego-vehicle can safely complete the overtaking maneuver. When comparing the last time step of top and bottom row, it can be seen that the rule “ <i>Formula 1 – racing alongside another car</i> ” is in effect, which does not allow the other vehicle to push the ego-vehicle off the track as soon as the vehicles’ footprints overlap along the s -coordinate. The reachable set is reduced accordingly.	58
Figure 5.10:	Comparison of approximated reachable set (a) and guaranteed occupied set (b). Here shown for a vehicle (orange) with initial velocity 40 m s^{-1} , a time horizon of 2 s , and a time interval of 0.1 s	59
Figure 5.11:	Illustration of the procedure for determining the guaranteed occupied set on the basis of two time instances for a vehicle (green) with initial velocity 25 m s^{-1} . The reachable set generated via simulation of more than 400 vehicle trajectories is shown in light blue (plot of individual vehicle footprints constitute the surface). The outline of the vehicle footprint of the respective vehicle poses in the reachable set at the respective time point are displayed in dark blue. The guaranteed occupied set (orange) is the set intersection of the more than 400 vehicle footprints at the respective point in time.	60
Figure 5.12:	Overview of all implemented SupMods in schematic representation. The SupMods are assigned to clusters static and dynamic according to the application in the respective static or dynamic environment. An identifier in the upper right corner indicates whether the module is applied for the performance (P) and/or emergency (E) trajectory.	61
Figure 5.13:	Decision procedure for the selection of a safe trajectory, based on [246]. .	62
Figure 5.14:	Schematic diagram illustrating the congruence of emergency and performance trajectories for a certain time period. The next point up to which the trajectories are congruent is indicated by a white circle with a black border. In this graph, it is assumed that only the orange emergency trajectory of the ego-vehicle (blue) is tested against the abstracted green reachable set of a TP (gray). In the first time step, the emergency trajectory is still valid with respect to this assumption. In the second time step, the vehicles are further ahead and the ego-vehicle has generated a new set of trajectories. The new emergency trajectory would result in the ego-vehicle vehicle intersecting the set and is classified as unsafe accordingly. For this reason, it is necessary to switch to the previous emergency trajectory (dashed). Since the vehicle has only moved on the congruent trajectory part in the meantime, a changeover at this point is not problematic.	62
Figure 6.1:	Schematic illustration of the SW framework used during the <i>Roborace Season Alpha</i> , supplemented by the Supervisor.	63

Figure 6.2:	Distribution of logged calculation time per iteration (evaluation of both, performance and emergency trajectory) for several multi-vehicle laps on a track in Modena, Italy. Depending on the length of the trajectories provided and vehicles in the vicinity of the vehicle, the calculation time varies in each iteration. The two observed clusters result from purely static tests when no other vehicle is in the vicinity (left cluster) and a full test coverage in the vicinity of other vehicles (right cluster).	67
Figure 6.3:	Main window of the Scenario Architect. A specific time of the scenario is highlighted by associated vehicle footprint boxes and the red trajectory of the ego-vehicle. The right side contains radio buttons, buttons, and checkboxes to make changes to individual entities in the scenario. Currently the vehicle "veh_2" is selected and in edit mode accordingly.	68
Figure 6.4:	Temporal information window of the Scenario Architect that displays the progression of acceleration and velocity over the entire scenario horizon. With the red marker a time-stamp can be selected, which is then displayed in the main window.	69
Figure 6.5:	Schematic representation of the challenges and characteristic thresholds in the generation of ground truth for dynamic scenarios.	71
Figure 6.6:	Illustrative sketch of the evaluation method based on a point-by-point analysis with regard to track boundary intersections.	74
Figure 6.7:	Passive Supervisor ratings for the 121 scenarios (S1-S121) examined, each paired with their expected safety rating. Regions where no expected rating is stated can take any Supervisor rating. The plot is cropped to a maximum duration of 20.0 s for better readability.	75
Figure 6.8:	Reachable set of a TP (green) and emergency trajectory of the ego-vehicle (orange) at 3.2 s into scenario S43, based on [246]. Future vehicle poses are drawn at 1.0 s increments, while every second instance i is denoted by t_i^{veh1} and t_i^{veh2} for the ego-vehicle and TP, respectively.	76
Figure 6.9:	Observed safety status of individual time-steps during active Supervisor runs for the 121 scenarios (S1-S121) examined, each paired with their expected safety state (expected throughout safe). The plot is cropped to a maximum duration of 20.0 s for better readability.	78
Figure 6.10:	Reachable set of a TP (green) and emergency trajectory of the ego-vehicle (orange) at 3.2 s into scenario T43, based on [246]. Future vehicle poses are drawn at 1.0 s increments, while every second instance i is denoted by t_i^{veh1} and t_i^{veh2} for the ego-vehicle and TP, respectively.	79
Figure 6.11:	Automated test vehicles used for evaluation: <i>Roborace DevBot 1.0</i> and <i>DevBot 2.0</i>	80
Figure 6.12:	Overview of the tracks on which the automated race vehicle was operated for this evaluation. Different scales are applied.	81
Figure 6.13:	Passive Supervisor ratings for 17 examined real-world test runs (T1-T17) each paired with their expected safety rating (expected throughout safe).	82
Figure 6.14:	Vehicle going off track due to infeasible velocity plan in Monteblanco, Spain (test run T8).	83
Figure 6.15:	Infeasible velocity plan in Monteblanco, Spain (test run T8), shown for time-step $t = 3.0$ s of Figure 6.14.	84
Figure 6.16:	Vehicle going off track due to infeasible cool-down plan in Modena, Italy (test run T5).	85

Figure 6.17:	Infeasible cool-down plan in Modena, Italy (test run T5), shown for time-step $t = 2.0\text{s}$ of Figure 6.16.	86
Figure 7.1:	Sketch based on a situation taken from a log of a test run in Upper Heyford, England. The current planned emergency trajectory of the following ego-vehicle (orange) is sketched in solid red. A much more efficient one (with earlier standstill) is shown in dashed red.	93
Figure 7.2:	Incidents exposed by the Supervisor during simulative testing for the Indy Autonomous Challenge on the digital twin of the Indianapolis Motor Speedway.	95
Figure A.1:	Test track in Upper Heyford, United Kingdom. The origin was arbitrarily chosen in accordance with the reference pose of the DGPS system.	xlvi
Figure A.2:	Test track in Monteblanco, Spain. The origin was arbitrarily chosen in accordance with the reference pose of the DGPS system.	xlvi
Figure A.3:	Test track in Modena, Italy. The origin was arbitrarily chosen in accordance with the reference pose of the DGPS system.	xlvi
Figure A.4:	Test track in Le Croix, France. The origin was arbitrarily chosen in accordance with the reference pose of the DGPS system. The narrows of the track were artificially generated by cones.	xlvi
Figure A.5:	Bird's eye view of scenario S1. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	xlvi
Figure A.6:	Bird's eye view of scenario S2. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	xlvi
Figure A.7:	Bird's eye view of scenario S3. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	xlvi
Figure A.8:	Bird's eye view of scenario S4. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	xlvi
Figure A.9:	Bird's eye view of scenario S5. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every eighth interval is marked with a text label.	l
Figure A.10:	Bird's eye view of scenario S6. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every eighth interval is marked with a text label.	l
Figure A.11:	Bird's eye view of scenario S7. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every eighth interval is marked with a text label.	li
Figure A.12:	Bird's eye view of scenario S8. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every eighth interval is marked with a text label.	li
Figure A.13:	Bird's eye view of scenario S9. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lii
Figure A.14:	Bird's eye view of scenario S10. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every sixth interval is marked with a text label.	lii

Figure A.15:	Bird's eye view of scenario S11. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every sixth interval is marked with a text label.	lii
Figure A.16:	Bird's eye view of scenario S12. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every sixth interval is marked with a text label.	liii
Figure A.17:	Bird's eye view of scenario S13. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every sixth interval is marked with a text label.	liii
Figure A.18:	Bird's eye view of scenario S14. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every sixth interval is marked with a text label.	liii
Figure A.19:	Bird's eye view of scenario S15. The vehicle movements are shown via trace (line) and vehicle shape samples at 10 s interval. Every fifth interval is marked with a text label.	liv
Figure A.20:	Bird's eye view of scenario S16. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	liv
Figure A.21:	Bird's eye view of scenario S17. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	lv
Figure A.22:	Bird's eye view of scenario S18. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lv
Figure A.23:	Bird's eye view of scenario S19. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lvi
Figure A.24:	Bird's eye view of scenario S20. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lvi
Figure A.25:	Bird's eye view of scenario S21. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lvii
Figure A.26:	Bird's eye view of scenario S22. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lvii
Figure A.27:	Bird's eye view of scenario S23. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lviii
Figure A.28:	Bird's eye view of scenario S24. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lviii
Figure A.29:	Bird's eye view of scenario S25. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lix
Figure A.30:	Bird's eye view of scenario S26. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lix

Figure A.31:	Bird's eye view of scenario S27. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	Ix
Figure A.32:	Bird's eye view of scenario S28. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	Ix
Figure A.33:	Bird's eye view of scenario S29. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	Ixi
Figure A.34:	Bird's eye view of scenario S30. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	Ixi
Figure A.35:	Bird's eye view of scenario S31. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.	Ixii
Figure A.36:	Bird's eye view of scenario S32. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.	Ixii
Figure A.37:	Bird's eye view of scenario S33. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	Ixiii
Figure A.38:	Bird's eye view of scenario S34. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	Ixiii
Figure A.39:	Bird's eye view of scenario S35. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fifth interval is marked with a text label.	Ixiv
Figure A.40:	Bird's eye view of scenario S36. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.	Ixiv
Figure A.41:	Bird's eye view of scenario S37. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	Ixv
Figure A.42:	Bird's eye view of scenario S38. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	Ixv
Figure A.43:	Bird's eye view of scenario S39. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	Ixvi
Figure A.44:	Bird's eye view of scenario S40. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	Ixvi
Figure A.45:	Bird's eye view of scenario S41. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	Ixvii
Figure A.46:	Bird's eye view of scenario S42. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	Ixvii

Figure A.47:	Bird's eye view of scenario S43. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	lxviii
Figure A.48:	Bird's eye view of scenario S44. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	lxviii
Figure A.49:	Bird's eye view of scenario S45. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxix
Figure A.50:	Bird's eye view of scenario S46. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every ninth interval is marked with a text label.	lxix
Figure A.51:	Bird's eye view of scenario S47. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.	lxx
Figure A.52:	Bird's eye view of scenario S48. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.	lxx
Figure A.53:	Bird's eye view of scenario S49. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxxi
Figure A.54:	Bird's eye view of scenario S50. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxxi
Figure A.55:	Bird's eye view of scenario S51. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	lxxii
Figure A.56:	Bird's eye view of scenario S52. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	lxxii
Figure A.57:	Bird's eye view of scenario S53. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxxiii
Figure A.58:	Bird's eye view of scenario S54. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fifth interval is marked with a text label.	lxxiii
Figure A.59:	Bird's eye view of scenario S55. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fifth interval is marked with a text label.	lxxiv
Figure A.60:	Bird's eye view of scenario S56. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxxiv
Figure A.61:	Bird's eye view of scenario S57. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxxv
Figure A.62:	Bird's eye view of scenario S58. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	lxxv

Figure A.63:	Bird's eye view of scenario S59. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	lxxvi
Figure A.64:	Bird's eye view of scenario S60. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxxvi
Figure A.65:	Bird's eye view of scenario S61. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxxvii
Figure A.66:	Bird's eye view of scenario S62. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxxvii
Figure A.67:	Bird's eye view of scenario S63. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxxviii
Figure A.68:	Bird's eye view of scenario S64. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxxviii
Figure A.69:	Bird's eye view of scenario S65. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxxix
Figure A.70:	Bird's eye view of scenario S66. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxxix
Figure A.71:	Bird's eye view of scenario S67. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxxx
Figure A.72:	Bird's eye view of scenario S68. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxxx
Figure A.73:	Bird's eye view of scenario S69. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxxx
Figure A.74:	Bird's eye view of scenario S70. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	lxxxii
Figure A.75:	Bird's eye view of scenario S71. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	lxxxii
Figure A.76:	Bird's eye view of scenario S72. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxxxii
Figure A.77:	Bird's eye view of scenario S73. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxxxiii
Figure A.78:	Bird's eye view of scenario S74. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxxxiii

Figure A.79:	Bird's eye view of scenario S75. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	lxxxiv
Figure A.80:	Bird's eye view of scenario S76. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	lxxxiv
Figure A.81:	Bird's eye view of scenario S77. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.	lxxxv
Figure A.82:	Bird's eye view of scenario S78. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.	lxxxv
Figure A.83:	Bird's eye view of scenario S79. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	lxxxvi
Figure A.84:	Bird's eye view of scenario S80. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fifth interval is marked with a text label.	lxxxvi
Figure A.85:	Bird's eye view of scenario S81. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fifth interval is marked with a text label.	lxxxvii
Figure A.86:	Bird's eye view of scenario S82. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.	lxxxvii
Figure A.87:	Bird's eye view of scenario S83. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.	lxxxviii
Figure A.88:	Bird's eye view of scenario S84. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxxxviii
Figure A.89:	Bird's eye view of scenario S85. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxxxix
Figure A.90:	Bird's eye view of scenario S86. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	lxxxix
Figure A.91:	Bird's eye view of scenario S87. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	xc
Figure A.92:	Bird's eye view of scenario S88. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	xc
Figure A.93:	Bird's eye view of scenario S89. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	xc
Figure A.94:	Bird's eye view of scenario S90. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	xc

Figure A.95:	Bird's eye view of scenario S91. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	xcii
Figure A.96:	Bird's eye view of scenario S92. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	xcii
Figure A.97:	Bird's eye view of scenario S93. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	xciii
Figure A.98:	Bird's eye view of scenario S94. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	xciii
Figure A.99:	Bird's eye view of scenario S95. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	xciv
Figure A.100:	Bird's eye view of scenario S96. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	xciv
Figure A.101:	Bird's eye view of scenario S97. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every ninth interval is marked with a text label.	xcv
Figure A.102:	Bird's eye view of scenario S98. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every ninth interval is marked with a text label.	xcv
Figure A.103:	Bird's eye view of scenario S99. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.	xcvi
Figure A.104:	Bird's eye view of scenario S100. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.	xcvi
Figure A.105:	Bird's eye view of scenario S101. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.	xcvii
Figure A.106:	Bird's eye view of scenario S102. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.	xcvii
Figure A.107:	Bird's eye view of scenario S103. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.	xcviii
Figure A.108:	Bird's eye view of scenario S104. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	xcviii
Figure A.109:	Bird's eye view of scenario S105. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	xcix
Figure A.110:	Bird's eye view of scenario S106. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	xcix

Figure A.111:	Bird's eye view of scenario S107. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	c
Figure A.112:	Bird's eye view of scenario S108. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	c
Figure A.113:	Bird's eye view of scenario S109. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	ci
Figure A.114:	Bird's eye view of scenario S110. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	ci
Figure A.115:	Bird's eye view of scenario S111. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	cii
Figure A.116:	Bird's eye view of scenario S112. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	cii
Figure A.117:	Bird's eye view of scenario S113. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	ciii
Figure A.118:	Bird's eye view of scenario S114. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	ciii
Figure A.119:	Bird's eye view of scenario S115. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	civ
Figure A.120:	Bird's eye view of scenario S116. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	civ
Figure A.121:	Bird's eye view of scenario S117. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.	civ
Figure A.122:	Bird's eye view of scenario S118. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.	cv
Figure A.123:	Bird's eye view of scenario S119. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fifth interval is marked with a text label.	cv
Figure A.124:	Bird's eye view of scenario S120. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fifth interval is marked with a text label.	cvi
Figure A.125:	Bird's eye view of scenario S121. The vehicle movements are shown via trace (line) and vehicle shape samples at 10 s interval. Every fifth interval is marked with a text label.	cvi

List of Tables

Table 2.1:	Overview and evaluation of the addressed aspects in related work regarding online safety assessment of trajectories for ADSs. Related work is clustered into sections (top two rows) and rated against four metrics (first column). The resulting rating is indicated via Harvey Balls (○ – not addressed to ● – comprehensively addressed) along with the brightness of the color in the cell (from light to dark). Details on the rating metric can be found in Table A.1. Each cell with a Harvey Ball holds the rating for the corresponding metric and method cluster. The smaller (sub)cells below each Harvey ball cell hold the ratings of the individual papers via encoded colors only. The subcells are sorted from left to right according to the papers referenced in the table header and are thus also assigned to a single paper along a vertical line. The cluster rating is obtained by taking the rounded mean value of the individual ratings.....	25
Table 5.1:	List of exemplary rules $z_i(\xi(t)) \in \mathcal{Z}$ to be obeyed by the ego-vehicle.	52
Table 5.2:	Implemented illustrative trigger conditions c_k and associated reduction sets \mathcal{Q}_k . To obtain rule-based reachable sets, the reduction set is subtracted from the unconstrained reachable set for all objects o for which the trigger condition holds. For clarity, mathematically Simplified Representations (SR) are marked.....	57
Table 6.1:	Evaluation of requirements for a safe trajectory (S-1-A) with associated scores (passed test (PT) or passed test given assumptions (PTGA)) based on expected and observed output.....	66
Table 6.2:	Evaluation of the requirements imposed on a Supervisor (S-1-B) with associated scores (passed test (PT)).	67
Table 6.3:	Number of individual ratings received across the 121 evaluated scenarios per binary classification type.....	74
Table 6.4:	Number of incidents and safe states observed during active Supervisor simulation across the 121 evaluated scenarios per binary classification type. The false negatives are shown in parentheses, as they were classified as reasonable unsafe states (true negatives) in a subsequent analysis.....	77
Table 6.5:	Number of individual ratings received across the 17 evaluated real-world test runs per binary classification type. The false negatives are shown in parentheses, as they were classified as valid ratings (true negatives) in a subsequent analysis.....	81
Table 7.1:	Evaluation of the four specified criteria (first column) with the OV method demonstrated in this work. The resulting rating is indicated via Harvey Balls (○ – not addressed to ● – comprehensively addressed) along with the brightness of the color in the cell (from light to dark). Details on the rating metric can be found in Table A.1.	93
Table A.1:	Evaluation criteria and their scale for the analysis of the relevant literature. .	xliii

List of Tables

Table A.2: Evaluated interface tests with associated scores (passed test (PT) or passed test given assumptions (TPGA)) based on expected and observed output. xliv

Table A.3: Crucial parameters (with explanation) for the generation of the ground truth in multi-vehicle scenarios. For most safety parameters, an upper and lower bound is specified to define the three ranges: safe, unspecified rating, unsafe. xlv

Bibliography

- [1] World Health Organization, *Global Status Report on Road Safety 2018*, (Nonserial Publication), World Health Organization, 2019, ISBN: 978-92-4-156568-4.
- [2] Vägverket, *Vision Zero on the Move*, Third, Swedish Transport Administration, 2013, ISBN: 978-91-7467-234-3.
- [3] R. Elvik, T. Vaa, A. Høy and M. Sørensen, *The Handbook of Road Safety Measures: Second Edition*, Emerald Group Publishing, 2009, ISBN: 978-1-84855-251-7.
- [4] European Commission, „Advanced Driver Assistance Systems,“ European Commission, Directorate General for Transport, 2016.
- [5] T. Winkle, „Sicherheitspotenzial automatisierter Fahrzeuge: Erkenntnisse aus der Unfallforschung,“ in *Autonomes Fahren* Springer Vieweg, Berlin, Heidelberg, 2015, pp. 351–376.
- [6] GDV - Unfallforschung der Versicherer, „Erhöht automatisiertes Fahren die Sicherheit?: Unfallforschung kompakt,“ 2018.
- [7] N. E. Boudette, „Autopilot Cited in Death of Chinese Tesla Driver,“ *The New York Times*, 2016.
- [8] H. Herrera, „Investigation of a Fatal Crash Involving a Vehicle with Level 2 Automation,“ *NHTSA*, p. 20, 2018.
- [9] National Transportation Safety Board (NTSB), „Collision between a Sport Utility Vehicle Operating with Partial Driving Automation and a Crash Attenuator, Mountain View, California, March 23, 2018,“ p. 96, 2020.
- [10] National Transportation Safety Board (NTSB), „Collision between Car Operating with Partial Driving Automation and Truck-Tractor Semitrailer, Delray Beach, Florida, March 1, 2019,“ p. 15, 2019.
- [11] National Transportation Safety Board (NTSB), „Collision between Vehicle Controlled by Developmental Automated Driving System and Pedestrian, Tempe, Arizona, March 18, 2018,“ p. 78, 2019.
- [12] S. H. Leilabadi and S. Schmidt, „In-Depth Analysis of Autonomous Vehicle Collisions in California,“ in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 889–893, doi: 10/ggdvtg.
- [13] B. Paden, M. Cap, S. Z. Yong, D. Yershov and E. Frazzoli, „A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles,“ *IEEE Transactions on Intelligent Vehicles*, pp. 33–55, 2016, doi: 10.1109/TIV.2016.2578706.
- [14] M. Bansal, A. Krizhevsky and A. Ogale, „ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst,“ in *arXiv Preprint*, 2018.
- [15] bast - Federal Highway Research Institute, „Traffic and Accident Data - Summary Statistics - Germany,“ Bergisch Gladbach, 2020.

- [16] H. Winner, „Quo vadis, FAS?“, in *Handbuch Fahrerassistenzsysteme: Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort* (ATZ/MTZ-Fachbuch), H. Winner, S. Hakuli, F. Lotz and C. Singer, ed. Wiesbaden: Springer Fachmedien, 2015, pp. 1167–1186, ISBN: 978-3-658-05734-3. DOI: 10.1007/978-3-658-05734-3_62.
- [17] N. Rajabli, F. Flammini, R. Nardone and V. Vittorini, „Software Verification and Validation of Safe Autonomous Cars: A Systematic Literature Review“, *IEEE Access*, vol. 9, pp. 4797–4819, 2021, DOI: 10/ghs2pz.
- [18] P. van Wesel and A. Goodloe, „Challenges in the Verification of Reinforcement Learning Algorithms“, *NASA/TM–2017–219628*, 2017.
- [19] H. Winner, „Introducing Autonomous Driving: An Overview of Safety Challenges and Market Introduction Strategies“, *at - Automatisierungstechnik*, vol. 66, no. 2, pp. 100–106, 2018, DOI: 10.1515/auto-2017-0106.
- [20] S. Shalev-Shwartz, S. Shammah and A. Shashua, „On a Formal Model of Safe and Scalable Self-Driving Cars“, in *arXiv Preprint*, 2017.
- [21] R. Murphy, R. R. Murphy and R. C. Arkin, *Introduction to AI Robotics*, MIT Press, 2000, ISBN: 978-0-262-13383-8.
- [22] S. Cai, S. Bileschi, E. D. Nielsen, N. Thorat and F. Chollet, *Cai, S: Deep Learning with JavaScript: Neural Networks in Tensorflow.js*, Shelter Island, NY, Manning Publications, 2020, ISBN: 978-1-61729-617-8.
- [23] ISO, „ISO 26262: Road Vehicles - Functional Safety“, 2018.
- [24] SAE International. „J3016B: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles“, 2018.
- [25] Tesla. „Autopilot“, <https://www.tesla.com/autopilot>. [Accessed: 2021-11-06].
- [26] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldts and M. Maurer, „Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving“, in *2015 IEEE International Conference on Intelligent Transportation Systems*, 2015, pp. 982–988, DOI: 10/gftjzj.
- [27] J. B. Michael, D. Drusinsky, T. W. Otani and M.-T. Shing, „Verification and Validation for Trustworthy Software Systems“, *IEEE Software*, vol. 28, no. 6, pp. 86–92, 2011, DOI: 10/d7sgsj.
- [28] S. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. Eng, D. Rus and M. Ang, „Perception, Planning, Control, and Coordination for Autonomous Vehicles“, *Machines*, vol. 5, no. 1, p. 6, 2017, DOI: 10.3390/machines5010006.
- [29] P. R. Palafox, J. Betz, F. Nobis, K. Riedl and M. Lienkamp, „SemanticDepth: Fusing Semantic Segmentation and Monocular Depth Estimation for Enabling Autonomous Driving in Roads without Lane Lines“, *Sensors*, vol. 19, no. 14, p. 3224, 2019, DOI: 10/ggjvq7.
- [30] F. Nobis, J. Betz, L. Hermansdorfer and M. Lienkamp, „Autonomous Racing: A Comparison of SLAM Algorithms for Large Scale Outdoor Environments“, in *Proceedings of the 2019 3rd International Conference on Virtual and Augmented Reality Simulations*, 2019, pp. 82–89, ISBN: 978-1-4503-6592-5. DOI: 10/ggjvq6.

- [31] L. Andresen, A. Brandemuehl, A. Honger, B. Kuan, N. Vödisch, H. Blum, V. Reijgwart, L. Bernreiter, L. Schaupp, J. J. Chung, M. Burki, M. R. Oswald, R. Siegwart and A. Gawel, „Accurate Mapping and Planning for Autonomous Racing,“ in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4743–4749, doi: 10/gmmf6n.
- [32] T. Stahl, A. Wischnewski, J. Betz and M. Lienkamp, „ROS-Based Localization of a Race Vehicle at High-Speed Using LIDAR,“ *E3S Web of Conferences*, vol. 95, 2019, doi: 10/ggfhgj.
- [33] N. Gosala, A. Bühler, M. Prajapat, C. Ehmke, M. Gupta, R. Sivanesan, A. Gawel, M. Pfeiffer, M. Bürki, I. Sa, R. Dubé and R. Siegwart, „Redundant Perception and State Estimation for Reliable Autonomous Racing,“ in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6561–6567, doi: 10/ggqcg3.
- [34] T. Renzler, M. Stolz, M. Schratte and D. Watzenig, „Increased Accuracy For Fast Moving LiDARS: Correction of Distorted Point Clouds,“ in *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2020, pp. 1–6, doi: 10/gmmgdf.
- [35] F. Massa, L. Bonamini, A. Settimi, L. Pallottino and D. Caporale, „LiDAR-Based GNSS Denied Localization for Autonomous Racing Cars,“ *Sensors*, vol. 20, no. 14, p. 3992, 2020, doi: 10/gjn7mn.
- [36] A. Gotlib, K. Łukojć and M. Szczygielski, „Localization-Based Software Architecture for 1:10 Scale Autonomous Car,“ in *2019 International Interdisciplinary PhD Workshop (IIPhDW)*, 2019, pp. 7–11, doi: 10/gmmgdh.
- [37] A. Brunnbauer and M. Bader, „Traffic Cone Based Self-Localization on a 1 : 10 Race Car,“ in *Traffic Cone Based Self-Localization on a 1:10 Race Car*, 2019, doi: 10/gmmttk.
- [38] A. Wischnewski, T. Stahl, J. Betz and B. Lohmann, „Vehicle Dynamics State Estimation and Localization for High Performance Race Cars,“ *IFAC-PapersOnLine*, vol. 52, no. 8, pp. 154–161, 2019, doi: 10/ggjquz.
- [39] N. L. Large, F. Bieder and M. Lauer, „Comparison of Different SLAM Approaches for a Driverless Race Car,“ *tm - Technisches Messen*, vol. 88, no. 4, pp. 227–236, 2021, doi: 10/gmmgdv.
- [40] W.-z. Peng, Y.-h. Ao, J.-h. He and P.-f. Wang, „Vehicle Odometry with Camera-Lidar-IMU Information Fusion and Factor-Graph Optimization,“ *Journal of Intelligent & Robotic Systems*, vol. 101, no. 4, p. 81, 2021, doi: 10/gmmgdz.
- [41] N. De Rita, A. Aimar and T. Delbruck, „CNN-Based Object Detection on Low Precision Hardware: Racing Car Case Study,“ in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 647–652, doi: 10/gmmgfx.
- [42] A. Dhall, D. Dai and L. Van Gool, „Real-Time 3D Traffic Cone Detection for Autonomous Driving,“ in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 494–501, doi: 10/gmmgdd.
- [43] K. Strobel, S. Zhu, R. Chang and S. Koppula, „Accurate, Low-Latency Visual Perception for Autonomous Racing: Challenges, Mechanisms, and Practical Solutions,“ in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 1969–1975, doi: 10/gmmggk.
- [44] F. Nobis, M. Geisslinger, M. Weber, J. Betz and M. Lienkamp, „A Deep Learning-Based Radar and Camera Sensor Fusion Architecture for Object Detection,“ in *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, 2019, pp. 1–7, doi: 10/ggjqu9.

- [45] F. Nobis, F. Brunhuber, S. Janssen, J. Betz and M. Lienkamp, „Exploring the Capabilities and Limits of 3D Monocular Object Detection - A Study on Simulation and Real World Data,“ in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–8, doi: 10/gmmgjd.
- [46] N. R. Kapania and J. C. Gerdes, „Design of a Feedback-Feedforward Steering Controller for Accurate Path Tracking and Stability at the Limits of Handling,“ *Vehicle System Dynamics*, vol. 53, no. 12, pp. 1687–1704, 2015, doi: 10/gdgxxf.
- [47] J. Ni and J. Hu, „Path Following Control for Autonomous Formula Racecar: Autonomous Formula Student Competition,“ in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1835–1840, doi: 10/gh4vms.
- [48] V. A. Laurence, J. Y. Goh and J. C. Gerdes, „Path-Tracking for Autonomous Vehicles at the Limit of Friction,“ in *2017 American Control Conference (ACC)*, 2017, pp. 5586–5591, doi: 10/gmmp4v.
- [49] M. Fu, J. Ni, X. Li and J. Hu, „Path Tracking for Autonomous Race Car Based on G-G Diagram,“ *International Journal of Automotive Technology*, vol. 19, no. 4, pp. 659–668, 2018, doi: 10/gds5gg.
- [50] J. Ni, J. Hu and C. Xiang, „Robust Path Following Control at Driving/Handling Limits of an Autonomous Electric Racecar,“ *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5518–5526, 2019, doi: 10/gmmp5d.
- [51] E. Wachter, A. Schmeitz, F. Bruzelius and M. Alirezaei, „Path Control in Limits of Vehicle Handling: A Sensitivity Analysis,“ in *Advances in Dynamics of Vehicles on Roads and Tracks*, 2020, pp. 1089–1095, ISBN: 978-3-030-38077-9. doi: 10/gh9tmg.
- [52] C. E. Beal and J. C. Gerdes, „Model Predictive Control for Vehicle Stabilization at the Limits of Handling,“ *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1258–1269, 2013, doi: 10/f43q9f.
- [53] G. Williams, B. Goldfain, P. Drews, J. M. Rehg and E. A. Theodorou, „Best Response Model Predictive Control for Agile Interactions Between Autonomous Ground Vehicles,“ in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2403–2410, doi: 10/gmmp5h.
- [54] G. Williams, P. Drews, B. Goldfain, J. M. Rehg and E. A. Theodorou, „Information-Theoretic Model Predictive Control: Theory and Applications to Autonomous Driving,“ *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1603–1622, 2018, doi: 10/gg3jhh.
- [55] T. Novi, A. Liniger, R. Capitani and C. Annicchiarico, „Real-Time Control for at-Limit Handling Driving on a Predefined Path,“ *Vehicle System Dynamics*, vol. 58, no. 7, pp. 1007–1036, 2020, doi: 10/ghdcdw.
- [56] Y. Liu, L. Shi, W. Xu, X. Xiong, W. Sun and L. Qu, „Design of Driverless Racing Chassis Based on MPC,“ in *2020 Chinese Automation Congress (CAC)*, 2020, pp. 6061–6066, doi: 10/gmmp5t.
- [57] M. S. Gandhi, B. Vlahov, J. Gibson, G. Williams and E. A. Theodorou, „Robust Model Predictive Path Integral Control: Analysis and Performance Guarantees,“ *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1423–1430, 2021, doi: 10/gmmp52.
- [58] A. Wischnewski, M. Euler, S. Gümüs and B. Lohmann, „Tube Model Predictive Control for an Autonomous Race Car,“ *Vehicle System Dynamics*, vol. 0, no. 0, pp. 1–23, 2021, doi: 10/gmmp53.

- [59] U. Rosolia, X. Zhang and F. Borrelli, „Robust Learning Model Predictive Control for Iterative Tasks: Learning from Experience,“ in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 1157–1162, doi: 10/gmmq2s.
- [60] A. Wischnewski, J. Betz and B. Lohmann, „Real-Time Learning of Non-Gaussian Uncertainty Models for Autonomous Racing,“ in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 609–615, doi: 10/gmmq2x.
- [61] U. Rosolia and F. Borrelli, „Learning How to Autonomously Race a Car: A Predictive Control Approach,“ *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2713–2719, 2020, doi: 10/gg94cs.
- [62] N. R. Kapania and J. C. Gerdes, „Learning at the Racetrack: Data-Driven Methods to Improve Racing Performance Over Multiple Laps,“ *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8232–8242, 2020, doi: 10/gmmq2w.
- [63] A. Jain, M. O’Kelly, P. Chaudhari and M. Morari, „BayesRace: Learning to Race Autonomously Using Prior Experience,“ in *Proceedings of the 2020 Conference on Robot Learning*, 2021, pp. 1918–1929.
- [64] X. Xiao, J. Biswas and P. Stone, „Learning Inverse Kinodynamics for Accurate High-Speed Off-Road Navigation on Unstructured Terrain,“ *IEEE Robotics and Automation Letters*, 2021, doi: 10/gmmq2z.
- [65] S. Kuutti, S. Fallah, R. Bowden and P. Barber, „Deep Learning for Autonomous Vehicle Control: Algorithms, State-of-the-Art, and Future Prospects,“ *Synthesis Lectures on Advances in Automotive Technology*, vol. 3, no. 4, pp. 1–80, 2019, doi: 10/ghhxs9.
- [66] R. Y. Hindiyeh and J. Christian Gerdes, „A Controller Framework for Autonomous Drifting: Design, Stability, and Experimental Validation,“ *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, no. 5, 2014, doi: 10/ggdj6q.
- [67] J. Y. Goh, T. Goel and J. Christian Gerdes, „Toward Automated Vehicle Control Beyond the Stability Limits: Drifting Along a General Path,“ *Journal of Dynamic Systems, Measurement, and Control*, vol. 142, no. 2, 2019, doi: 10/gmmq32.
- [68] I. Zubov, I. Afanasyev, A. Gabdullin, R. Mustafin and I. Shimchik, „Autonomous Drifting Control in 3D Car Racing Simulator,“ in *2018 International Conference on Intelligent Systems (IS)*, 2018, pp. 235–241, doi: 10/gmmq33.
- [69] D. Gonzalez, J. Perez, V. Milanés and F. Nashashibi, „A Review of Motion Planning Techniques for Automated Vehicles,“ *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2016, doi: 10.1109/TITS.2015.2498841.
- [70] C. Katrakazas, M. Quddus, W.-H. Chen and L. Deka, „Real-Time Motion Planning Methods for Autonomous on-Road Driving: State-of-the-Art and Future Research Directions,“ *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416–442, 2015, doi: 10.1016/j.trc.2015.09.011.
- [71] T. Stahl, A. Wischnewski, J. Betz and M. Lienkamp, „Multilayer Graph-Based Trajectory Planning for Race Vehicles in Dynamic Scenarios,“ in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3149–3154, doi: 10/ggdtkc.
- [72] T. T. Mac, C. Copot, D. T. Tran and R. Keyser, „Heuristic Approaches in Robot Path Planning: A Survey,“ *Robotics and Autonomous Systems*, vol. 86, pp. 13–28, 2016, doi: 10.1016/j.robot.2016.08.001.

- [73] S. Karaman and E. Frazzoli, „Optimal Kinodynamic Motion Planning Using Incremental Sampling-Based Methods,“ in *IEEE Conference on Decision and Control*, 2010, pp. 7681–7687, ISBN: 978-1-4244-7745-6. doi: 10.1109/CDC.2010.5717430.
- [74] J. Kuffner and S. LaValle, „RRT-Connect: An Efficient Approach to Single-Query Path Planning,“ in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, 2000, 995–1001 vol.2, doi: 10/d7hmkm.
- [75] L. Jaillet, J. Cortes and T. Simeon, „Transition-Based RRT for Path Planning in Continuous Cost Spaces,“ in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 2145–2150, doi: 10/dqksgs.
- [76] J. D. Gammell, S. S. Srinivasa and T. D. Barfoot, „Informed RRT*: Optimal Sampling-Based Path Planning Focused via Direct Sampling of an Admissible Ellipsoidal Heuristic,“ in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 2997–3004, doi: 10/gctq9p.
- [77] A. Joglekar, B. Deshpande, M. Basuthakur and V. N. Krovi, „Fusing Offline and Online Trajectory Optimization Techniques for Goal-to-Goal Navigation of a Scaled Autonomous Vehicle,“ SAE International, Warrendale, PA rep. 2021-01-0097, 04/2021, doi: 10.4271/2021-01-0097.
- [78] J. hwan Jeon, S. Karaman and E. Frazzoli, „Anytime Computation of Time-Optimal off-Road Vehicle Maneuvers Using the RRT*,“ in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 3276–3282, doi: 10/gg55gn.
- [79] J. hwan Jeon, R. V. Cowlagi, S. C. Peters, S. Karaman, E. Frazzoli, P. Tsiotras and K. Iagnemma, „Optimal Motion Planning with the Half-Car Dynamical Model for Autonomous High-Speed Driving,“ in *2013 American Control Conference*, 2013, pp. 188–193, doi: 10/gh22n4.
- [80] O. Arslan, K. Berntorp and P. Tsiotras, „Sampling-Based Algorithms for Optimal Motion Planning Using Closed-Loop Prediction,“ in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4991–4996, doi: 10/gmmq6g.
- [81] S. Feraco, S. Luciani, A. Bonfitto, N. Amati and A. Tonoli, „A Local Trajectory Planning and Control Method for Autonomous Vehicles Based on the RRT Algorithm,“ in *2020 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE)*, 2020, pp. 1–6, doi: 10/gmmq6k.
- [82] J.-w. Choi, R. Curry and G. Elkaim, „Path Planning Based on Bézier Curve for Autonomous Ground Vehicles,“ in *Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science 2008*, 2008, pp. 158–166, doi: 10/cwvzx5.
- [83] K. G. Jolly, R. Sreerama Kumar and R. Vijayakumar, „A Bezier Curve Based Path Planning in a Multi-Agent Robot Soccer System without Violating the Acceleration Limits,“ *Robotics and Autonomous Systems*, vol. 57, no. 1, pp. 23–33, 2009, doi: 10/bbt7xh.
- [84] L. Han, H. Yashiro, H. Tehrani Nik Nejad, Q. H. Do and S. Mita, „Bézier Curve Based Path Planning for Autonomous Vehicle in Urban Environment,“ in *2010 IEEE Intelligent Vehicles Symposium*, 2010, pp. 1036–1042, doi: 10/dzdm2r.

- [85] T. Berglund, A. Brodник, H. Jonsson, M. Staffanson and I. Soderkvist, „Planning Smooth and Obstacle-Avoiding B-Spline Paths for Autonomous Mining Vehicles,“ *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 1, pp. 167–172, 2010, doi: 10/bmwszw.
- [86] F. Ahmed and K. Deb, „Multi-Objective Path Planning Using Spline Representation,“ in *2011 IEEE International Conference on Robotics and Biomimetics*, 2011, pp. 1047–1052, doi: 10/gg58hs.
- [87] D. H. Shin, S. Singh and W. Whittaker, „Path Generation for a Robot Vehicle Using Composite Clothoid Segments,“ *IFAC Proceedings Volumes*, vol. 25, no. 6, pp. 443–448, 1992, doi: 10/gg58hz.
- [88] G. Williams, P. Drews, B. Goldfain, J. M. Rehg and E. A. Theodorou, „Aggressive Driving with Model Predictive Path Integral Control,“ in *IEEE International Conference on Robotics and Automation*, 2016, pp. 1433–1440, ISBN: 978-1-4673-8026-3. doi: 10.1109/ICRA.2016.7487277.
- [89] T. Rizano, D. Fontanelli, L. Palopoli, L. Pallottino and P. Salaris, „Local Motion Planning for Robotic Race Cars,“ in *IEEE Conference on Decision and Control*, 2013, ISBN: 978-1-4673-5717-3.
- [90] M. Brunner, U. Rosolia, J. Gonzales and F. Borrelli, „Repetitive Learning Model Predictive Control: An Autonomous Racing Example,“ in *IEEE Conference on Decision and Control*, 2017, pp. 2545–2550, ISBN: 978-1-5090-2873-3. doi: 10.1109/CDC.2017.8264027.
- [91] T. Brüdigam, A. Capone, S. Hirche, D. Wollherr and M. Leibold, „Gaussian Process-Based Stochastic Model Predictive Control for Overtaking in Autonomous Racing,“ *ArXiv*, 2021.
- [92] J. R. Anderson, B. Ayalew and T. Weiskircher, „Modeling a Professional Driver in Ultra-High Performance Maneuvers with a Hybrid Cost MPC,“ in *2016 American Control Conference (ACC)*, 2016, pp. 1981–1986, doi: 10/gmmq4g.
- [93] T. Herrmann, F. Passigato, J. Betz and M. Lienkamp, „Minimum Race-Time Planning-Strategy for an Autonomous Electric Racecar,“ in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6, doi: 10/gmmq4h.
- [94] A. Rucco, G. Notarstefano and J. Hauser, „An Efficient Minimum-Time Trajectory Generation Strategy for Two-Track Car Vehicles,“ *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, pp. 1505–1519, 2015, doi: 10/f7g96v.
- [95] F. Christ, A. Wischnewski, A. Heilmeier and B. Lohmann, „Time-Optimal Trajectory Planning for a Race Car Considering Variable Tyre-Road Friction Coefficients,“ *Vehicle System Dynamics*, vol. 0, no. 0, pp. 1–25, 2019, doi: 10/ggjqwg.
- [96] I. Gundlach and U. Konigorski, „Modellbasierte Online-Trajektorienplanung für zeitoptimale Rennlinien,“ *at - Automatisierungstechnik*, vol. 67, no. 9, pp. 799–813, 2019, doi: 10/gmmq4j.
- [97] F. Gottmann, H. Wind and O. Sawodny, „On the Influence of Rear Axle Steering and Modeling Depth on a Model Based Racing Line Generation for Autonomous Racing,“ in *2018 IEEE Conference on Control Technology and Applications (CCTA)*, 2018, pp. 846–852, doi: 10/gmmq4k.
- [98] J. K. Subosits and J. C. Gerdes, „Impacts of Model Fidelity on Trajectory Optimization for Autonomous Vehicles in Extreme Maneuvers,“ *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 3, pp. 546–558, 2021, doi: 10/gmmq4m.

- [99] F. Braghin, F. Cheli, S. Melzi and E. Sabbioni, „Race Driver Model,“ *Computers & Structures*, vol. 86, no. 13-14, pp. 1503–1516, 2008, doi: 10.1016/j.compstruc.2007.04.028.
- [100] L. Cardamone, D. Loiacono, P. L. Lanzi and A. P. Bardelli, „Searching for the Optimal Racing Line Using Genetic Algorithms,“ in *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*, 2010, pp. 388–394, doi: 10/fw8cmj.
- [101] A. Heilmeyer, A. Wischnewski, L. Hermansdorfer, J. Betz, M. Lienkamp and B. Lohmann, „Minimum Curvature Trajectory Planning and Control for an Autonomous Race Car,“ *Vehicle System Dynamics*, pp. 1–31, 2019, doi: 10/ggjqvx.
- [102] X. Li, Z. Sun, D. Cao, D. Liu and H. He, „Development of a New Integrated Local Trajectory Planning and Tracking Control Framework for Autonomous Ground Vehicles,“ *Mechanical Systems and Signal Processing*, vol. 87, pp. 118–137, 2017, doi: 10.1016/j.ymsp.2015.10.021.
- [103] X. Hu, L. Chen, B. Tang, D. Cao and H. He, „Dynamic Path Planning for Autonomous Driving on Various Roads with Avoidance of Static and Moving Obstacles,“ *Mechanical Systems and Signal Processing*, vol. 100, pp. 482–500, 2018, doi: 10.1016/j.ymsp.2017.07.019.
- [104] M. Werling, J. Ziegler, S. Kammel and S. Thrun, „Optimal Trajectory Generation for Dynamic Street Scenarios in a Frenét Frame,“ in *IEEE International Conference on Robotics and Automation*, 2010, pp. 987–993, ISBN: 978-1-4244-5038-1. doi: 10.1109/ROBOT.2010.5509799.
- [105] T. Gu, J. Snider, J. M. Dolan and J.-w. Lee, „Focused Trajectory Planning for Autonomous On-Road Driving,“ in *IEEE Intelligent Vehicles Symposium*, 2013, pp. 547–552, ISBN: 978-1-4673-2755-8. doi: 10.1109/IVS.2013.6629524.
- [106] M. McNaughton, C. Urmson, J. M. Dolan and J.-w. Lee, „Motion Planning for Autonomous Driving with a Conformal Spatiotemporal Lattice,“ in *IEEE International Conference on Robotics and Automation*, 2011, pp. 4889–4895, ISBN: 978-1-61284-386-5. doi: 10.1109/ICRA.2011.5980223.
- [107] D. Tobin. „*First Roborace Driverless Car Race Held in Spain*,“ <https://www.motorsportmagazine.com/articles/sports-cars/first-robora-ce-driverless-car-race-held-spain>. [Accessed: 2021-10-01]. 2019.
- [108] T. Stahl. „*GitHub | Graph-Based Local Trajectory Planner*,“ <https://github.com/TUMFTM/GraphBasedLocalTrajectoryPlanner>. [Accessed: 2021-06-02]. 2020.
- [109] J. Betz, A. Wischnewski, A. Heilmeyer, F. Nobis, L. Hermansdorfer, T. Stahl, T. Herrmann and M. Lienkamp, „A Software Architecture for the Dynamic Path Planning of an Autonomous Racecar at the Limits of Handling,“ in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, 2019, doi: 10/gg6x8c.
- [110] B. Nessler. „*Künstliche Intelligenz für automatisierte Fahrzeuge: Möglichkeiten und Stand der AI aus dem Blick winkel Computer und Neurowissenschaft*,“ Tagung Automatisiertes Fahren, Munich, 2019.
- [111] D. A. Pomerleau, „ALVINN: An Autonomous Land Vehicle in a Neural Network,“ in *Advances in Neural Information Processing Systems*, 1989.
- [112] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao and K. Zieba, „End to End Learning for Self-Driving Cars,“ 2016.

- [113] T. Weiss and M. Behl, „DeepRacing: A Framework for Autonomous Racing,“ in *2020 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2020, pp. 1163–1168, doi: 10/gjnk87.
- [114] E. Perot, M. Jaritz, M. Toromanoff and R. De Charette, „End-to-End Driving in a Realistic Racing Game with Deep Reinforcement Learning,“ in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 474–475, doi: 10/ghgdp5.
- [115] M. Jaritz, R. de Charette, M. Toromanoff, E. Perot and F. Nashashibi, „End-to-End Race Driving with Deep Reinforcement Learning,“ in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2070–2075, doi: 10/ggb45w.
- [116] Y. Pan, C.-A. Cheng, K. Saigol, K. Lee, X. Yan, E. Theodorou and B. Boots, „Agile Autonomous Driving Using End-to-End Deep Imitation Learning,“ in *Robotics: Science and Systems XIV*, 2018, ISBN: 978-0-9923747-4-7.
- [117] M. Wood, P. Robbel, M. Maass, R. D. Tebbens, M. Meijs, M. Harb, J. Reach, K. Robinson, D. Wittmann, T. Srivastava, et al., „Safety First for Automated Driving,“ *Aptiv, Audi, BMW, Baidu, Continental Teves, Daimler, FCA, HERE, Infineon Technologies, Intel, Volkswagen*, 2019.
- [118] M. Bouton, A. Nakhaei, K. Fujimura and M. J. Kochenderfer, „Cooperation-Aware Reinforcement Learning for Merging in Dense Traffic,“ in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3441–3447, doi: 10/gg3q85.
- [119] Y. Hu, A. Nakhaei, M. Tomizuka and K. Fujimura, „Interaction-Aware Decision Making with Adaptive Strategies under Merging Scenarios,“ in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 151–158, doi: 10/gjnk62.
- [120] C.-J. Hoel, K. Wolff and L. Laine, „Tactical Decision-Making in Autonomous Driving by Reinforcement Learning with Uncertainty Estimation,“ in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1563–1569, doi: 10/gh9t3t.
- [121] P. Cai, Y. Sun, Y. Chen and M. Liu, „Vision-Based Trajectory Planning via Imitation Learning for Autonomous Vehicles,“ in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 2736–2742, doi: 10/ggdtwk.
- [122] J. Chen, B. Yuan and M. Tomizuka, „Deep Imitation Learning for Autonomous Driving in Generic Urban Scenarios with Enhanced Safety,“ in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 2884–2890, doi: 10/gjnz6r.
- [123] L. Schnieder and R. S. Hosse, *Leitfaden Safety of the Intended Functionality: Verfeinerung der Sicherheit der Sollfunktion auf dem Weg zum autonomen Fahren*, Springer Vieweg, 2019, ISBN: 978-3-658-25022-5.
- [124] Á. Takács, D. A. Drexler, P. Galambos, I. J. Rudas and T. Haidegger, „Assessment and Standardization of Autonomous Vehicles,“ in *2018 IEEE 22nd International Conference on Intelligent Engineering Systems (INES)*, 2018, pp. 000185–000192, doi: 10/ghd2jz.
- [125] S. E. Shladover and C. Nowakowski, „Regulatory Challenges for Road Vehicle Automation: Lessons from the California Experience,“ *Transportation Research Part A: Policy and Practice*, vol. 122, pp. 125–133, 2019, doi: 10/gc4n5t.
- [126] UNECE. „Text of the 1958 Agreement,“ <https://unece.org/trans/main/wp29/wp29regs>. [Accessed: 2021-09-14].

- [127] W. von Wendorff, „Quantitative SOTIF Analysis: For Automated Driving Systems,“ 2019.
- [128] ISO, „ISO/PAS 21448: Road Vehicles - Safety of the Intended Functionality,“ 2019.
- [129] ISO, „ISO/SAE 21434: Road Vehicles - Cybersecurity Engineering,“ 2019.
- [130] G. Macher, C. Schmittner, O. Veledar and E. Brenner, „ISO/SAE DIS 21434 Automotive Cybersecurity Standard - In a Nutshell,“ in *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops*, 2020, pp. 123–135, ISBN: 978-3-030-55583-2. DOI: 10/gcxtx.
- [131] F. Sommer, J. Dürrwang and R. Kriesten, „Survey and Classification of Automotive Security Attacks,“ *Information*, vol. 10, no. 4, p. 148, 2019, doi: 10/gg7sq7.
- [132] M. Ring, D. Frkat and M. Schmiedecker, „Cybersecurity Evaluation of Automotive e/e Architectures,“ in *ACM Computer Science In Cars Symposium (CSCS 2018)*, 2018.
- [133] UL, „UL 4600: Standard for Safety for the Evaluation of Autonomous Products,“ vol. 1, 2020.
- [134] IT-Beauftragter der Bundesregierung. „V-Modell XT,“ https://www.cio.bund.de/Web/DE/Architekturen-und-Standards/V-Modell-XT/vmodell_xt_node.html. [Accessed: 2020-08-19].
- [135] D. Gunning, „Explainable Artificial Intelligence (Xai),“ *Defense Advanced Research Projects Agency (DARPA), nd Web*, vol. 2, 2017.
- [136] R. Salay, R. Queiroz and K. Czarnecki, „An Analysis of ISO 26262: Machine Learning and Safety in Automotive Software,“ in *SAE Technical Paper*, 2018.
- [137] H. Monkhouse, I. Habli, J. McDermid, S. Khastgir and G. Dhadyalla, „Why Functional Safety Experts Worry about Automotive Systems Having Increasing Autonomy,“ in *International Workshop on Driver and Driverless Cars: Competition or Coexistence*, 2017, pp. 1–6, doi: 10/ggcfng.
- [138] A. Koenig, K. Witzlsperger, F. Leutwiler and S. Hohmann, „Overview of HAD Validation and Passive HAD as a Concept for Validating Highly Automated Cars,“ *at - Automatisierungstechnik*, vol. 66, no. 2, pp. 132–145, 2018, doi: 10.1515/auto-2017-0113.
- [139] K. Heckemann, M. Gesell, T. Pfister, K. Berns, K. Schneider and M. Trapp, „Safe Automotive Software,“ in *Knowledge-Based and Intelligent Information and Engineering Systems*, 2011, pp. 167–176, ISBN: 978-3-642-23866-6. doi: 10/cxc5rn.
- [140] F. Falcini and G. Lami, „Deep Learning in Automotive: Challenges and Opportunities,“ in *Software Process Improvement and Capability Determination*, 2017, pp. 279–288, ISBN: 978-3-319-67383-7. doi: 10/ghd79d.
- [141] K.-W. Wu, C.-C. Liao and W.-F. Wu, „Reliability and Safety Assessment of Automated Driving Systems: Review and Preview,“ in *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2020, pp. 390–394, doi: 10/ghs2p4.
- [142] A. K. Saberi, J. Hegge, T. Fruehling and J. F. Groote, „Beyond SOTIF: Black Swans and Formal Methods,“ in *2020 IEEE International Systems Conference (SysCon)*, 2020, pp. 1–5, doi: 10/ghn5c2.
- [143] 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 *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2020, pp. 86–93, doi: 10/gg6rbm.

- [144] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman and D. Mané, „Concrete Problems in AI Safety,“ *arXiv:1606.06565*, 2016. arXiv: 1606.06565.
- [145] S. Burton, L. Gauerhof and C. Heinzemann, „Making the Case for Safety of Machine Learning in Highly Automated Driving,“ in *Computer Safety, Reliability, and Security*, 2017, pp. 5–16, ISBN: 978-3-319-66284-8. DOI: 10/ggcfnn.
- [146] M. Henzel, W. Hermann and B. Lattke, „Herausforderungen in der Absicherung von Fahrerassistenzsystemen bei der Benutzung maschinell gelernter und lernender Algorithmen,“ *11. Workshop Fahrerassistenzsysteme, Walting*, 2017.
- [147] B. Spanfelner, D. Richter, S. Ebel, U. Wilhelm, W. Branz and C. Patz, „Challenges in Applying the ISO 26262 for Driver Assistance Systems,“ *Tagung Fahrerassistenz, München*, vol. 15, no. 16, 2012.
- [148] P. Koopman and M. Wagner, „Challenges in Autonomous Vehicle Testing and Validation,“ *SAE International Journal of Transportation Safety*, vol. 4, no. 1, pp. 15–24, 2016, DOI: 10/ggcfnp.
- [149] J. M. Faria, „Machine Learning Safety: An Overview,“ in *Proceedings of the 26th Safety-Critical Systems Symposium, York, UK*, 2018.
- [150] S. Wachter, B. Mittelstadt and L. Floridi, „Transparent, Explainable, and Accountable AI for Robotics,“ *Science Robotics*, vol. 2, no. 6, p. 6080, 2017, DOI: 10.1126/scirobotics.aan6080.
- [151] S. Kuutti, R. Bowden, H. Joshi, R. de Temple and S. Fallah, „Safe Deep Neural Network-Driven Autonomous Vehicles Using Software Safety Cages,“ in *Intelligent Data Engineering and Automated Learning – IDEAL 2019*, 2019, pp. 150–160, ISBN: 978-3-030-33617-2. DOI: 10/gghbk7.
- [152] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick and F. Diermeyer, „Survey on Scenario-Based Safety Assessment of Automated Vehicles,“ *IEEE Access*, pp. 1–1, 2020, DOI: 10/ggv7ng.
- [153] J. E. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, F. Niewels and J. M. Zöllner, „Testing of Advanced Driver Assistance towards Automated Driving: A Survey and Taxonomy on Existing Approaches and Open Questions,“ in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 1455–1462, DOI: 10/gghpvs.
- [154] W. Huang, K. Wang, Y. Lv and F. Zhu, „Autonomous Vehicles Testing Methods Review,“ in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 163–168, DOI: 10/ggh2p7.
- [155] P. Junietz, W. Wachenfeld, K. Klonecki and H. Winner, „Evaluation of Different Approaches to Address Safety Validation of Automated Driving,“ in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 491–496, DOI: 10/ggntm3.
- [156] H. Winner, W. Wachenfeld and P. Junietz, „Validation and Introduction of Automated Driving,“ in *Automotive Systems Engineering II*, H. Winner, G. Prokop and M. Maurer, ed. Cham: Springer International Publishing, 2018, pp. 177–196, ISBN: 978-3-319-61607-0. DOI: 10.1007/978-3-319-61607-0_8.
- [157] U. Steininger. „How Safe Is Safe Enough? PEGASUS Delivers the Standards for Highly Automated Driving,“ Munich, 2019.
- [158] C. Wang and H. Winner, „Overcoming Challenges of Validation Automated Driving and Identification of Critical Scenarios,“ in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 2639–2644, DOI: 10/ghfmhc.

- [159] Tesla. „*Tesla Autonomy Day*,“ <https://www.youtube.com/watch?v=Ucp0TTmvqOE>. [Accessed: 2020-10-14]. 2019.
- [160] Waymo, „Waymo Safety Report,“ 2021.
- [161] Daimler. „*Daimler and Bosch: Start of the San José Pilot Project for Automated Ride-Hailing Service*,“ <https://www.daimler.com/innovation/case/autonomous/pilot-city-san-jose.html>. [Accessed: 2021-09-14]. 2019.
- [162] T. Ponn, D. Fratzke, C. Gnanndt and M. Lienkamp, „Towards Certification of Autonomous Driving: Systematic Test Case Generation for a Comprehensive but Economically-Feasible Assessment of Lane Keeping Assist Algorithms,“ in *VEHITS 2019 - Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems*, 2019, pp. 333–342.
- [163] S. Kitajima, K. Shimono, J. Tajima, J. Antona-Makoshi and N. Uchida, „Multi-Agent Traffic Simulations to Estimate the Impact of Automated Technologies on Safety,“ *Traffic Injury Prevention*, vol. 20, no. sup1, S58–S64, 2019, doi: 10/ggkwr4.
- [164] M. Saraoglu, A. Morozov and K. Janschek, „MOBATSim: MOdel-Based Autonomous Traffic Simulation Framework for Fault-Error-Failure Chain Analysis,“ *IFAC-PapersOnLine*, vol. 52, no. 8, pp. 239–244, 2019, doi: 10/ghfnp4.
- [165] M. Althoff and S. Lutz, „Automatic Generation of Safety-Critical Test Scenarios for Collision Avoidance of Road Vehicles,“ in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1326–1333, doi: 10/ggjbbx.
- [166] P. Junietz, F. Bonakdar, B. Klamann and H. Winner, „Criticality Metric for the Safety Validation of Automated Driving Using Model Predictive Trajectory Optimization,“ in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 60–65, doi: 10/ggj782.
- [167] C. Amersbach and H. Winner, „Defining Required and Feasible Test Coverage for Scenario-Based Validation of Highly Automated Vehicles,“ in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 425–430, doi: 10/ggdt8.
- [168] Y. Akagi, R. Kato, S. Kitajima, J. Antona-Makoshi and N. Uchida, „A Risk-Index Based Sampling Method to Generate Scenarios for the Evaluation of Automated Driving Vehicle Safety,“ in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 667–672, doi: 10/ggdtvd.
- [169] F. Hauer, T. Schmidt, B. Holzmüller and A. Pretschner, „Did We Test All Scenarios for Automated and Autonomous Driving Systems?,“ in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 2950–2955, doi: 10/ggdtwh.
- [170] J. Kerber, S. Wagner, K. Groh, D. Notz, T. Kühbeck, D. Watzenig and A. Knoll, „Clustering of the Scenario Space for the Assessment of Automated Driving,“ in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 578–583, doi: 10/ghs2p8.
- [171] Z. Xinxin, L. Fei and W. Xiangbin, „CSG: Critical Scenario Generation from Real Traffic Accidents,“ in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1330–1336, doi: 10/ghs2p6.
- [172] R. Yu, H. Ai and Z. Gao, „Identifying High Risk Driving Scenarios Utilizing a CNN-LSTM Analysis Approach,“ in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6, doi: 10/gm2gzv.

- [173] M. Hekmatnejad, B. Hoxha and G. Fainekos, „Search-Based Test-CASe Generation by Monitoring Responsibility Safety Rules,” in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–8, doi: 10/gjrbsbx.
- [174] J. Sun, H. Zhou, H. Zhang, Y. Tian and Q. Ji, „Adaptive Design of Experiments for Accelerated Safety Evaluation of Automated Vehicles,” in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–7, doi: 10/gm2gz2.
- [175] D. Karunakaran, S. Worrall and E. Nebot, „Efficient Statistical Validation with Edge Cases to Evaluate Highly Automated Vehicles,” in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–8, doi: 10/gm2gz4.
- [176] 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 *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6, doi: 10/gj3wzc.
- [177] M. Moradi, B. J. Oakes, M. Saraoglu, A. Morozov, K. Janschek and J. Denil, „Exploring Fault Parameter Space Using Reinforcement Learning-Based Fault Injection,” in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2020, pp. 102–109, doi: 10/gg6rbn.
- [178] T. Ponn, T. Lanz and F. Diermeyer, „Automatic Generation of Road Geometries to Create Challenging Scenarios for Automated Vehicles Based on the Sensor Setup,” in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 694–700, doi: 10/ghs2p9.
- [179] T. Ponn, „How to Define System-Specific Corner Cases for the Type-Approval of Automated Vehicles,” Dissertation, Technical University of Munich, Munich, 2021.
- [180] C. Neurohr, L. Westhofen, T. Henning, T. de Graaff, E. Möhlmann and E. Böde, „Fundamental Considerations around Scenario-Based Testing for Automated Driving,” in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 121–127, doi: 10/ghs2p2.
- [181] S. Riedmaier, J. Schneider, B. Danquah, B. Schick and F. Diermeyer, „Non-Deterministic Model Validation Methodology for Simulation-Based Safety Assessment of Automated Vehicles,” *Simulation Modelling Practice and Theory*, vol. 109, p. 102274, 2021, doi: 10/gmt5zw.
- [182] M. Luckcuck, M. Farrell, L. A. Dennis, C. Dixon and M. Fisher, „Formal Specification and Verification of Autonomous Robotic Systems: A Survey,” *ACM Computing Surveys*, vol. 52, no. 5, pp. 1–100:41, 2019, doi: 10/gg336f.
- [183] N. Aréchiga, S. M. Loos, A. Platzer and B. H. Krogh, „Using Theorem Provers to Guarantee Closed-Loop System Properties,” in *2012 American Control Conference (ACC)*, 2012, pp. 3573–3580, doi: 10/ghfnvk.
- [184] H. Täubig, U. Frese, C. Hertzberg, C. Lüth, S. Mohr, E. Vorobev and D. Walter, „Guaranteeing Functional Safety: Design for Provability and Computer-Aided Verification,” *Autonomous Robots*, vol. 32, no. 3, pp. 303–331, 2012, doi: 10/fx283v.
- [185] B. Johnson, F. Havlak, H. Kress-Gazit and M. Campbell, „Experimental Evaluation and Formal Analysis of High-Level Tasks with Dynamic Obstacle Anticipation on a Full-Sized Autonomous Vehicle,” *Journal of Field Robotics*, vol. 34, no. 5, pp. 897–911, 2017, doi: 10/gbn97s.

- [186] T. Wongpiromsarn, U. Topcu and R. M. Murray, „Receding Horizon Temporal Logic Planning,“ *IEEE Transactions on Automatic Control*, vol. 57, no. 11, pp. 2817–2830, 2012, doi: 10/ghf5q2.
- [187] M. Althoff, „Reachability Analysis and Its Application to the Safety Assessment of Autonomous Cars,“ Dissertation, Technical University of Munich, Munich, 2010.
- [188] C. E. Tuncali, J. Kapinski, H. Ito and J. V. Deshmukh, „Reasoning about Safety of Learning-Enabled Components in Autonomous Cyber-Physical Systems,“ in *Proceedings of the 55th Annual Design Automation Conference*, 2018, pp. 1–6, ISBN: 978-1-4503-5700-5. doi: 10/ghfnwt.
- [189] C. Liu, T. Arnon, C. Lazarus, C. Strong, C. Barrett and M. J. Kochenderfer, „Algorithms for Verifying Deep Neural Networks,“ *Foundations and Trends® in Optimization*, vol. 4, no. 3-4, pp. 244–404, 2021, doi: 10/gm2ht8.
- [190] G. Hains, A. Jakobsson and Y. Khmelevsky, „Towards Formal Methods and Software Engineering for Deep Learning: Security, Safety and Productivity for DL Systems Development,“ in *The 12th Annual IEEE International Systems Conference: SysCon 2018, Vancouver, Canada : 2018 Conference Proceedings*, 2018, ISBN: 978-1-5386-3664-0. doi: 10.1109/SYSCON.2018.8369576.
- [191] G. J. D. R. Hains, A. Jakobsson and Y. Khmelevsky, „Formal Methods and Software Engineering for DL. Security, Safety and Productivity for DL Systems Development,“ *arXiv:1901.11334*, 2019. arXiv: 1901.11334.
- [192] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele and T. Darrell, „Generating Visual Explanations,“ in *Conference on Computer Vision*, 2016, pp. 3–19, ISBN: 978-3-319-46492-3. doi: 10.1007/978-3-319-46493-0_1.
- [193] T. Gehr, M. Mirman, D. Drachler-Cohen, P. Tsankov, S. Chaudhuri and M. Vechev, „AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation,“ in *2018 IEEE Symposium on Security and Privacy (SP)*, 2018, pp. 3–18, doi: 10/gg78wc.
- [194] G. Katz, C. Barrett, D. L. Dill, K. Julian and M. J. Kochenderfer, „Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks,“ in *Computer Aided Verification*, 2017, pp. 97–117, ISBN: 978-3-319-63387-9. doi: 10/ggcfm8.
- [195] D. Selsam, P. Liang and D. L. Dill, „Developing Bug-Free Machine Learning Systems with Formal Mathematics,“ in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017, pp. 3047–3056.
- [196] E. Gracic, F. Svensson, J. Ehrich, O. Beck and M. Jansen, „Concept for Safety-Related Development of Deep Neural Networks in the Automotive,“ in *2020 Fourth International Conference on Multimedia Computing, Networking and Applications (MCNA)*, 2020, pp. 10–15, doi: 10/ghm9jn.
- [197] M. Schreier, V. Willert and J. Adamy, „An Integrated Approach to Maneuver-Based Trajectory Prediction and Criticality Assessment in Arbitrary Road Environments,“ *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2751–2766, 2016, doi: 10.1109/TITS.2016.2522507.
- [198] M. Schreier, „Bayesian Environment Representation, Prediction, and Criticality Assessment for Driver Assistance Systems,“ *at - Automatisierungstechnik*, vol. 65, no. 2, 2017, doi: 10.1515/auto-2016-0129.

- [199] S. Annell, A. Gratner and L. Svensson, „Probabilistic Collision Estimation System for Autonomous Vehicles,“ in *IEEE Conference on Intelligent Transportation Systems*, 2016, pp. 473–478, ISBN: 978-1-5090-1889-5. doi: 10.1109/ITSC.2016.7795597.
- [200] A. Shangguan, G. Xie, D. Wang, R. Fei, X. Hei and W. Ji, „Analyzing the Collision Probability of Autonomous Vehicle at Crossroad,“ in *2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS)*, 2020, pp. 30–35, doi: 10/ghn5cx.
- [201] A. Houenou, P. Bonnifait and V. Cherfaoui, „Risk Assessment for Collision Avoidance Systems,“ in *IEEE Conference on Intelligent Transportation Systems*, 2014, pp. 386–391, ISBN: 978-1-4799-6078-1. doi: 10.1109/ITSC.2014.6957721.
- [202] L. Zhang, W. Xiao, Z. Zhang and D. Meng, „Surrounding Vehicles Motion Prediction for Risk Assessment and Motion Planning of Autonomous Vehicle in Highway Scenarios,“ *IEEE Access*, vol. 8, pp. 209356–209376, 2020, doi: 10/ghm9js.
- [203] A. Lambert, D. Gruyer, G. S. Pierre and A. N. Ndjeng, „Collision Probability Assessment for Speed Control,“ in *IEEE Conference on Intelligent Transportation Systems*, 2008, pp. 1043–1048, ISBN: 978-1-4244-2111-4. doi: 10.1109/ITSC.2008.4732692.
- [204] J. Ward, G. Agamennoni, S. Worrall and E. Nebot, „Vehicle Collision Probability Calculation for General Traffic Scenarios under Uncertainty,“ in *IEEE Intelligent Vehicles Symposium*, 2014, pp. 986–992, ISBN: 978-1-4799-3638-0. doi: 10.1109/IVS.2014.6856430.
- [205] A. Eidehall and L. Petersson, „Statistical Threat Assessment for General Road Scenes Using Monte Carlo Sampling,“ *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 137–147, 2008, doi: 10.1109/TITS.2007.909241.
- [206] Y. Wang, Z. Liu, Z. Zuo, Z. Li, L. Wang and X. Luo, „Trajectory Planning and Safety Assessment of Autonomous Vehicles Based on Motion Prediction and Model Predictive Control,“ *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 8546–8556, 2019, doi: 10/gf9qzb.
- [207] M. Althoff, O. Stursberg and M. Buss, „Safety Assessment of Driving Behavior in Multi-Lane Traffic for Autonomous Vehicles,“ in *IEEE Intelligent Vehicles Symposium*, 2009, pp. 893–900, ISBN: 978-1-4244-3503-6. doi: 10.1109/IVS.2009.5164398.
- [208] B. Kim, K. Park and K. Yi, „Probabilistic Threat Assessment with Environment Description and Rule-Based Multi-Traffic Prediction for Integrated Risk Management System,“ *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 3, pp. 8–22, 2017, doi: 10.1109/MITS.2017.2709807.
- [209] C. Laugier, I. E. Paromtchik, M. Perrollaz, M. Yong, J.-D. Yoder, C. Tay, K. Mekhnacha and A. Nègre, „Probabilistic Analysis of Dynamic Scenes and Collision Risks Assessment to Improve Driving Safety,“ *IEEE Intelligent Transportation Systems Magazine*, vol. 3, no. 4, pp. 4–19, 2011, doi: 10/bn9862.
- [210] A. Wardzinski, „Dynamic Risk Assessment in Autonomous Vehicles Motion Planning,“ in *2008 1st International Conference on Information Technology*, 2008, pp. 1–4, doi: 10/d6ktnk.
- [211] R. Tami, B. Soualmi, A. Doufene, J. Ibanez and J. Dauwels, „Machine Learning Method to Ensure Robust Decision-Making of AVs,“ in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 1217–1222, doi: 10/ggdzs8.
- [212] N. Bao, D. Yang, A. Carballo, Ü. Özgüner and K. Takeda, „Personalized Safety-Focused Control by Minimizing Subjective Risk,“ in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3853–3858, doi: 10/ggdztb.

- [213] M. Baur, „Fault Detection for Robotic Systems Based on Introspective Classification,“ Bachelor’s Thesis, Technical University of Munich, 2017.
- [214] C. Yang, A. Renzaglia, A. Paigwar, C. Laugier and D. Wang, „Driving Behavior Assessment and Anomaly Detection for Intelligent Vehicles,“ in *2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, 2019, pp. 524–529, doi: 10/ggw4pq.
- [215] P. St-Aubin, N. Saunier and L. Miranda-Moreno, „Comparison of Various Time-to-Collision Prediction and Aggregation Methods for Surrogate Safety Analysis,“ *Transportation Research Board 94th Annual Meeting*, 2014.
- [216] K. Mattas, M. Makridis, G. Botzoris, B. Ciuffo and B. Papadopoulos, „Fuzzy Surrogate Safety Metrics,“ in *2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, 2019, pp. 1–11, doi: 10/ggcdrs.
- [217] M. Aramrattana, T. Larsson, C. Englund, J. Jansson and A. Nåbo, „A Novel Risk Indicator for Cut-In Situations,“ in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6, doi: 10/gh8gcz.
- [218] A. Reschka, J. R. Bohmer, T. Nothdurft, P. Hecker, B. Lichte and M. Maurer, „A Surveillance and Safety System Based on Performance Criteria and Functional Degradation for an Autonomous Vehicle,“ in *IEEE Conference on Intelligent Transportation Systems*, 2012, pp. 237–242, ISBN: 978-1-4673-3063-3. doi: 10.1109/ITSC.2012.6338682.
- [219] M. Hörwick and K.-H. Siedersberger, „Strategy and Architecture of a Safety Concept for Fully Automatic and Autonomous Driving Assistance Systems,“ in *IEEE Intelligent Vehicles Symposium*, 2010, pp. 955–960, ISBN: 978-1-4244-7866-8. doi: 10.1109/IVS.2010.5548115.
- [220] P. Feth, D. Schneider and R. Adler, „A Conceptual Safety Supervisor Definition and Evaluation Framework for Autonomous Systems,“ in *Computer Safety, Reliability, and Security*, 2017, pp. 135–148, ISBN: 978-3-319-66266-4. doi: 10/ggjtwb.
- [221] A. Kane, O. Chowdhury, A. Datta and P. Koopman, „A Case Study on Runtime Monitoring of an Autonomous Research Vehicle (ARV) System,“ in *Runtime Verification*, 2015, pp. 102–117, ISBN: 978-3-319-23820-3. doi: 10/ggjtqm.
- [222] K. Esterle, V. Aravantinos and A. Knoll, „From Specifications to Behavior: Maneuver Verification in a Semantic State Space,“ in *IEEE Intelligent Vehicles Symposium*, 2019, pp. 2140–2147, doi: 10.1109/IVS.2019.8814241.
- [223] J. Mao and L. Chen, „Runtime Monitoring for Cyber-Physical Systems: A Case Study of Cooperative Adaptive Cruise Control,“ in *2012 Second International Conference on Intelligent System Design and Engineering Application*, 2012, pp. 509–515, doi: 10/gf9qbr.
- [224] K. Watanabe, E. Kang, C.-W. Lin and S. Shiraishi, „Runtime Monitoring for Safety of Intelligent Vehicles,“ in *IEEE Design Automation Conference (DAC)*, 2018, pp. 1–6, ISBN: 978-1-5386-4114-9. doi: 10.1109/DAC.2018.8465912.
- [225] S. Maierhofer, A.-K. Rettinger, E. C. Mayer and M. Althoff, „Formalization of Interstate Traffic Rules in Temporal Logic,“ in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020.
- [226] B. Xu, Q. Li, T. Guo, Y. Ao and D. Du, „A Quantitative Safety Verification Approach for the Decision-Making Process of Autonomous Driving,“ in *2019 International Symposium on Theoretical Aspects of Software Engineering (TASE)*, 2019, pp. 128–135, doi: 10/ggdztd.
- [227] M. Lowry and D. Dvorak, „Analytic Verification of Flight Software,“ *IEEE Intelligent Systems and their Applications*, vol. 13, no. 5, pp. 45–49, 1998, doi: 10/ddtqrj.

- [228] C. Chai, X. Zeng, X. Wu and X. Wang, „Safety Evaluation of Responsibility-Sensitive Safety (RSS) on Autonomous Car-Following Maneuvers Based on Surrogate Safety Measurements,“ in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 175–180, doi: 10/ggdtks.
- [229] P. F. Orzechowski, K. Li and M. Lauer, „Towards Responsibility-Sensitive Safety of Automated Vehicles with Reachable Set Analysis,“ in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, 2019, pp. 1–6, doi: 10/ggjqt.
- [230] J. Grieser, M. Zhang, T. Warnecke and A. Rausch, „Assuring the Safety of End-to-End Learning-Based Autonomous Driving through Runtime Monitoring,“ in *2020 23rd Euromicro Conference on Digital System Design (DSD)*, 2020, pp. 476–483, doi: 10/gjnx8p.
- [231] B. Mirchevska, C. Pek, M. Werling, M. Althoff and J. Boedecker, „High-Level Decision Making for Safe and Reasonable Autonomous Lane Changing Using Reinforcement Learning,“ in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2156–2162, doi: 10/ggrdh8.
- [232] F. Gruber and M. Althoff, „Anytime Safety Verification of Autonomous Vehicles,“ in *IEEE Conference on Intelligent Transportation Systems*, 2018, pp. 1708–1714, doi: 10.1109/ITSC.2018.8569950.
- [233] B. Schürmann, D. Heß, J. Eilbrecht, O. Stursberg, F. Koster and M. Althoff, „Ensuring Drivability of Planned Motions Using Formal Methods,“ in *IEEE Conference on Intelligent Transportation Systems*, 2017, pp. 1–8, ISBN: 978-1-5386-1526-3. doi: 10.1109/ITSC.2017.8317647.
- [234] M. Althoff and J. M. Dolan, „Online Verification of Automated Road Vehicles Using Reachability Analysis,“ *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 903–918, 2014, doi: 10.1109/TRO.2014.2312453.
- [235] C. Pek, M. Koschi and M. Althoff, „An Online Verification Framework for Motion Planning of Self-Driving Vehicles with Safety Guarantees,“ in *AAET - Automatisiertes Und Vernetztes Fahren*, 2019, pp. 260–274.
- [236] C. Pek and M. Althoff, „Computationally Efficient Fail-Safe Trajectory Planning for Self-Driving Vehicles Using Convex Optimization,“ in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 1447–1454, doi: 10/gm2h5g.
- [237] C. Pek, S. Manzinger, M. Koschi and M. Althoff, „Using Online Verification to Prevent Autonomous Vehicles from Causing Accidents,“ *Nature Machine Intelligence*, vol. 2, no. 9, pp. 518–528, 2020, doi: 10/ghcb6v.
- [238] A. Rizaldi, F. Immler, B. Schürmann and M. Althoff, „A Formally Verified Motion Planner for Autonomous Vehicles,“ in *Automated Technology for Verification and Analysis*, 2018, pp. 75–90, ISBN: 978-3-030-01090-4. doi: 10/ggfjh9.
- [239] D. Heß, M. Althoff and T. Sattel, „Formal Verification of Maneuver Automata for Parameterized Motion Primitives,“ in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 1474–1481, doi: 10/ggbtt6.
- [240] P. Du, Z. Huang, T. Liu, T. Ji, K. Xu, Q. Gao, H. Sibai, K. Driggs-Campbell and S. Mitra, „Online Monitoring for Safe Pedestrian-Vehicle Interactions,“ in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–8, doi: 10/gm2h5j.

- [241] H. Krasowski, X. Wang and M. Althoff, „Safe Reinforcement Learning for Autonomous Lane Changing Using Set-Based Prediction,“ in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–7, doi: 10/gk5fs7.
- [242] S. Kojchev, E. Klintberg and J. Fredriksson, „A Safety Monitoring Concept for Fully Automated Driving,“ in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–7, doi: 10/gkf9wp.
- [243] M. Khatun, M. Glaß and R. Jung, „An Approach of Scenario-Based Threat Analysis and Risk Assessment Over-the-Air Updates for an Autonomous Vehicle,“ in *2021 7th International Conference on Automation, Robotics and Applications (ICARA)*, 2021, pp. 122–127, doi: 10/gjjqgd.
- [244] T. Stahl, M. Eicher, J. Betz and F. Diermeyer, „Online Verification Concept for Autonomous Vehicles – Illustrative Study for a Trajectory Planning Module,“ in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, doi: 10/gm2h5t.
- [245] T. Stahl and J. Betz, „An Open-Source Scenario Architect for Autonomous Vehicles,“ in *2020 Fifteenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*, 2020, doi: 10.1109/EVER48776.2020.9243029.
- [246] T. Stahl and F. Diermeyer, „Online Verification Enabling Approval of Driving Functions—Implementation for a Planner of an Autonomous Race Vehicle,“ *IEEE Open Journal of Intelligent Transportation Systems*, vol. 2, pp. 97–110, 2021, doi: 10/gj3535.
- [247] T. Fruehling, A. Hailemichael, C. Graves, J. Riehl, E. Nutt, R. Fischer and A. K. Saberi, „Architectural Safety Perspectives Considerations Regarding the AI-Based AV Domain Controller,“ in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, 2019, pp. 1–10, doi: 10/ggjq4.
- [248] M. Hörwick, „Sicherheitskonzept für hochautomatisierte Fahrerassistenzsysteme,“ Dissertation, Technische Universität München, München, 2011.
- [249] N. Kalra and S. M. Paddock, „Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?,“ *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182–193, 2016, doi: 10/f9drfr.
- [250] W. Wachenfeld and H. Winner, „Die Freigabe des autonomen Fahrens,“ in *Autonomes Fahren*, M. Maurer, J. C. Gerdes, B. Lenz and H. Winner, ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 439–464, isbn: 978-3-662-45853-2. doi: 10.1007/978-3-662-45854-9_21.
- [251] J. Betz, A. Wischnewski, A. Heilmeyer, F. Nobis, T. Stahl, L. Hermansdorfer, B. Lohmann and M. Lienkamp, „What Can We Learn from Autonomous Level-5 Motorsport?,“ in *International Munich Chassis Symposium 2018: Chassis.Tech plus : Proceedings* Springer Vieweg, 2018, pp. 123–146, isbn: 978-3-658-22049-5. doi: 10.1007/978-3-658-22050-1_12.
- [252] N. Stasinski, „Identification and Characterization of Safety Relevant Criteria of Trajectories,“ Bachelor’s Thesis, Technical University of Munich, 2019.
- [253] D. D. Walden, G. J. Roedler, K. Forsberg, R. D. Hamelin and T. M. Shortell, *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, John Wiley & Sons, 2015.
- [254] V. D. Bhise, *Automotive Product Development: A Systems Engineering Implementation*, First, CRC Press, 2017.

- [255] G. Bagschik, T. Menzel and M. Maurer, „Ontology Based Scene Creation for the Development of Automated Vehicles,“ in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1813–1820, doi: 10/ggh5wt.
- [256] C. Pek, V. Rusinov, S. Manzinger, M. C. Üste and M. Althoff, „CommonRoad Drivability Checker: Simplifying the Development and Validation of Motion Planning Algorithms,“ in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1013–1020, doi: 10/gm2jdd.
- [257] M. Althoff, D. Heß and F. Gamber, „Road Occupancy Prediction of Traffic Participants,“ in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 2013, pp. 99–105, doi: 10/ggbtt7.
- [258] G. Lafferriere, G. J. Pappas and S. Yovine, „Symbolic Reachability Computation for Families of Linear Vector Fields,“ *Journal of Symbolic Computation*, vol. 32, no. 3, pp. 231–253, 2001, doi: 10/ddr3b5.
- [259] D. Beckert, A. Pereira and M. Althoff, „Online Verification of Multiple Safety Criteria for a Robot Trajectory,“ in *IEEE Conference on Decision and Control*, 2017, ISBN: 978-1-5090-2873-3.
- [260] M. Althoff and S. Magdici, „Set-Based Prediction of Traffic Participants on Arbitrary Road Networks,“ *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 2, pp. 187–202, 2016, doi: 10/ghfs7z.
- [261] N. Rack, „Entwicklung einer Methode zur prädiktionsbasierten Bestimmung definitiv belegter Zustände dynamischer Objekte,“ Semester Thesis, Technical University of Munich, 2020.
- [262] P. Radecker, „Entwicklung einer Methode zur prädiktionsbasierten Bestimmung definitiv belegter Zustände,“ Semester Thesis, Technical University of Munich, 2020.
- [263] K. Havelund and G. Roşu, „Synthesizing Monitors for Safety Properties,“ in *Tools and Algorithms for the Construction and Analysis of Systems*, 2002, pp. 342–356, ISBN: 978-3-540-46002-2. doi: 10/fs76c3.
- [264] R. Homma, T. Wakasugi and K. Kodaka, „Influence of Difference of Minimum Risk Maneuver of an Automated Vehicle on the Following Vehicle,“ *Transactions of Society of Automotive Engineers of Japan*, vol. 51, no. 1, 2020.
- [265] L. Wang, Z. Wu, J. Li and C. Stiller, „Real-Time Safe Stop Trajectory Planning via Multidimensional Hybrid A*-Algorithm,“ in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–7, doi: 10/gmszg5.
- [266] H. B. Pacejka, *Tyre and Vehicle Dynamics*, 2. ed., Amsterdam, Elsevier, 2006, ISBN: 0-7680-1702-5.
- [267] L. Hermansdorfer, J. Betz and M. Lienkamp, „Benchmarking of a Software Stack for Autonomous Racing against a Professional Human Race Driver,“ in *2020 Fifteenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*, 2020, pp. 1–8, doi: 10/gjzhrk.
- [268] J. Betz, A. Heilmeyer, A. Wischnewski, T. Stahl and M. Lienkamp, „Autonomous Driving — A Crash Explained in Detail,“ *Applied Sciences*, vol. 9, no. 23, 2019, doi: 10/ggjqqm.
- [269] A. Censi, K. Slutsky, T. Wongpiromsarn, D. Yershov, S. Pendleton, J. Fu and E. Frazzoli, „Liability, Ethics, and Culture-Aware Behavior Specification Using Rulebooks,“ in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8536–8542, ISBN: 978-1-5386-6027-0. doi: 10.1109/ICRA.2019.8794364.

- [270] Federation Internationale de L'Automobile, „Formula 1 - Sporting Regulations,“ 2016.
- [271] Federation Internationale de L'Automobile, „Appendix L to the International Sporting Code,“ 2020.
- [272] M. Bayerlein, „Kollisionserkennung für Trajektorien eines autonomen Fahrzeugs im dynamischen Umfeld,“ Semester Thesis, Technical University of Munich, 2020.
- [273] T. Fraichard and H. Asama, „Inevitable Collision States — a Step towards Safer Robots?,“ *Advanced Robotics*, vol. 18, no. 10, pp. 1001–1024, 2004, doi: 10/cqnvs2.
- [274] S. Petti and T. Fraichard, „Safe Motion Planning in Dynamic Environments,“ in *IEEE Conference on Intelligent Robots and Systems*, 2005, pp. 2210–2215, ISBN: 0-7803-8912-3. doi: 10.1109/IROS.2005.1545549.
- [275] A. Bautin, L. Martinez-Gomez and T. Fraichard, „Inevitable Collision States: A Probabilistic Perspective,“ in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 4022–4027, doi: 10/c2h2gg.
- [276] C. Schmidt, „Fahrstrategien zur Unfallvermeidung im Straßenverkehr für Einzel- und Mehrobjektszenarien,“ Dissertation, Karlsruher Institut für Technologie, Karlsruhe, 2013, ISBN: 9783731501985. doi: 10.5445/KSP/1000039757.
- [277] C. Schmidt, F. Oechsle and W. Branz, „Research on Trajectory Planning in Emergency Situations with Multiple Objects,“ in *2006 IEEE Intelligent Transportation Systems Conference*, 2006, pp. 988–992, doi: 10/dgb4s4.
- [278] T. Stahl. „*GitHub | Trajectory Supervisor*,“ <https://github.com/TUMFTM/TrajectorySupervisor>. [Accessed: 2021-06-02]. 2021.
- [279] Roborace. „*Roborace | Global Championship of Driverless Cars*,“ <https://www.roborace.com/>. [Accessed: 2018-11-08].
- [280] L. Goudkamp and Bayerischer Rundfunk. „*Roboter-Rennauto vs Mensch: Mit einem Algorithmus beim Roborace 2018 zum Sieg?*,“ <https://www.br.de/fernsehen/ard-alpha/sendungen/campusmagazin/robo-race-autonome-auto-roboter-rennen-berlin-challenge-100.html>. [Accessed: 2021-10-06]. 2018.
- [281] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, T. Stahl, L. Hermansdorfer and M. Lienkamp, „A Software Architecture for an Autonomous Racecar,“ in *IEEE Vehicular Technology Conference (VTC2019-Spring)*, 2019, ISBN: 978-1-72811-217-6. doi: 10.1109/VTCSpring.2019.8746367.
- [282] A. Heilmeier, F. Christ, L. Hermansdorfer, T. Herrmann and T. Stahl. „*GitHub | Global Racetrajectory Optimization*,“ https://github.com/TUMFTM/global_racetrajectory_optimization. [Accessed: 2021-06-02]. 2019.
- [283] ZeroMQ authors. „*ZeroMQ*,“ <https://zeromq.org/>. [Accessed: 2021-06-02].
- [284] A. Wischnewski. „*GitHub | Autonomous Driving Control Software of TUM Roborace Team*,“ https://github.com/TUMFTM/mod_vehicle_dynamics_control. [Accessed: 2021-06-02]. 2019.
- [285] V. Antinyan and H. Sandgren, „A Method for Software Safety Analysis to Support ISO 26262 Compliance in Agile Development,“ *IEEE Software*, pp. 52–60, 2020, doi: 10/gcppw.
- [286] T. Stahl. „*GitHub | Scenario Architect*,“ <https://github.com/TUMFTM/ScenarioArchitect>. [Accessed: 2021-06-02]. 2020.

- [287] K. Mattas, M. Makridis, G. Botzoris, B. Ciuffo and B. Papadopoulos, „Fuzzy Surrogate Safety Metrics,“ in *2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, 2019, pp. 1–11, doi: 10/ggcdrs.
- [288] S. M. S. Mahmud, L. Ferreira, M. S. Hoque and A. Tavassoli, „Application of Proximal Surrogate Indicators for Safety Evaluation: A Review of Recent Developments and Research Needs,“ *IATSS Research*, vol. 41, no. 4, pp. 153–163, 2017, doi: 10/gf9kms.
- [289] A. Heilmeier. „*GitHub | Racetrack-Database*,“ <https://github.com/TUMFTM/racetrack-database>. [Accessed: 2021-09-15]. 2021.
- [290] Roborace. „*FIRST-EVER Autonomous Race | Montebianco Episode 03 | Roborace*,“ <https://www.youtube.com/watch?v=yoTQlaoVuHg>. [Accessed: 2021-08-01].
- [291] C. T. Amersbach, „Functional Decomposition Approach - Reducing the Safety Validation Effort for Highly Automated Driving,“ Dissertation, Technical University Darmstadt, Darmstadt, 2020, doi: 10.25534/tuprints-00011520.
- [292] J. Yu and F. Luo, „Fallback Strategy for Level 4+ Automated Driving System,“ in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 156–162, doi: 10/ggdtsg.
- [293] F. Duerr, J. Ziehn, R. Kohlhaas, M. Roschani, M. Ruf and J. Beyerer, „Realtime Global Optimization of a Fail-Safe Emergency Stop Maneuver for Arbitrary Electrical/ Electronical Failures in Automated Driving,“ in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–8, doi: 10/gm2jh5.
- [294] Energy Systems Network. „*Indy Autonomous Challenge*,“ <https://www.indyautonomouchallenge.com>. [Accessed: 2021-08-20].
- [295] A. Wischnewski, M. Geisslinger, J. Betz, T. Betz, F. Fent, A. Heilmeier, L. Hermansdorfer, T. Herrmann, S. Huch, P. Karle, F. Nobis, L. Ögretmen, M. Rowold, F. Sauerbeck, T. Stahl, R. Trauth and M. Lienkamp, „Indy Autonomous Challenge - Autonomous Race Cars at the Handling Limits,“ in *International Munich Chassis Symposium 2021: Chassis.Tech plus : Proceedings* Springer Vieweg, 2021.
- [296] H. Schrieber, „Indy Autonomous Challenge: Ohne Fahrer aufs Treppchen,“ *Auto Bild*, pp. 44–49, 2021.
- [297] Deutschlandfunk. „*Autonome Rennwagen - Mehr als 200 km/h ohne Mensch im Cockpit*,“ https://www.deutschlandfunk.de/autonome-rennwagen-mehr-als-200-km-h-ohne-mensch-im-cockpit.676.de.html?dram:article_id=504729. [Accessed: 2021-11-10]. 2021.
- [298] Stern. „*TU München siegt bei Autonomous Challenge: Der autonome Indy-Sieg*,“ <https://www.stern.de/auto/news/tu-muenchen-siegt-bei-autonomous-challenge-der-autonome-indy-sieg-30863192.html>. [Accessed: 2021-11-10]. 2021.
- [299] M. Oitzman. „*TUM Autonomous Motorsport Victorious at the Indy Autonomous Challenge*,“ <https://www.roboticsbusinessreview.com/rbr-news/tum-autonomous-motorsport-victorious-at-the-indy-autonomous-challenge/>. [Accessed: 2021-11-10]. 2021.
- [300] Z. Horrall. „*Boundaries Pushed: German Team Wins Indy Autonomous Challenge*,“ <https://www.indianapolismotorspeedway.com/news-multimedia/news/2021/10/25/Technische-Universitat-Munchen-indy-autonomous-challenge-ims-innovation>. [Accessed: 2021-11-10]. 2021.

Prior Publications

During the development of this dissertation, publications and student theses were written in which partial aspects of this work were presented.

Journals; Scopus/Web of Science listed (peer-reviewed)

- [32] T. Stahl, A. Wischnewski, J. Betz and M. Lienkamp, „ROS-Based Localization of a Race Vehicle at High-Speed Using LIDAR,“ *E3S Web of Conferences*, vol. 95, 2019, doi: 10/ggfhgj.
- [38] A. Wischnewski, T. Stahl, J. Betz and B. Lohmann, „Vehicle Dynamics State Estimation and Localization for High Performance Race Cars,“ *IFAC-PapersOnLine*, vol. 52, no. 8, pp. 154–161, 2019, doi: 10/ggjqvz.
- [246] T. Stahl and F. Diermeyer, „Online Verification Enabling Approval of Driving Functions—Implementation for a Planner of an Autonomous Race Vehicle,“ *IEEE Open Journal of Intelligent Transportation Systems*, vol. 2, pp. 97–110, 2021, doi: 10/gj3535.
- [268] J. Betz, A. Heilmeier, A. Wischnewski, T. Stahl and M. Lienkamp, „Autonomous Driving — A Crash Explained in Detail,“ *Applied Sciences*, vol. 9, no. 23, 2019, doi: 10/ggjqwm.

Conferences, Periodicals; Scopus/Web of Science listed (peer-reviewed)

- [71] T. Stahl, A. Wischnewski, J. Betz and M. Lienkamp, „Multilayer Graph-Based Trajectory Planning for Race Vehicles in Dynamic Scenarios,“ in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3149–3154, doi: 10/ggdtkc.
- [109] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, L. Hermansdorfer, T. Stahl, T. Herrmann and M. Lienkamp, „A Software Architecture for the Dynamic Path Planning of an Autonomous Racecar at the Limits of Handling,“ in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, 2019, doi: 10/gg6x8c.
- [244] T. Stahl, M. Eicher, J. Betz and F. Diermeyer, „Online Verification Concept for Autonomous Vehicles – Illustrative Study for a Trajectory Planning Module,“ in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, doi: 10/gm2h5t.
- [245] T. Stahl and J. Betz, „An Open-Source Scenario Architect for Autonomous Vehicles,“ in *2020 Fifteenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*, 2020, doi: 10.1109/EVER48776.2020.9243029.
- [281] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, T. Stahl, L. Hermansdorfer and M. Lienkamp, „A Software Architecture for an Autonomous Racecar,“ in *IEEE Vehicular Technology Conference (VTC2019-Spring)*, 2019, ISBN: 978-1-72811-217-6. doi: 10.1109/VTCSpring.2019.8746367.

Journals, Conferences, Periodicals, Reports, Conference Proceedings and Poster, etc.; not Scopus/Web of Science listed

- [251] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, T. Stahl, L. Hermansdorfer, B. Lohmann and M. Lienkamp, „What Can We Learn from Autonomous Level-5 Motorsport?“, in *International Munich Chassis Symposium 2018: Chassis.Tech plus : Proceedings* Springer Vieweg, 2018, pp. 123–146, ISBN: 978-3-658-22049-5. DOI: 10.1007/978-3-658-22050-1_12.
- [295] A. Wischnewski, M. Geisslinger, J. Betz, T. Betz, F. Fent, A. Heilmeier, L. Hermansdorfer, T. Herrmann, S. Huch, P. Karle, F. Nobis, L. Ögretmen, M. Rowold, F. Sauerbeck, T. Stahl, R. Trauth and M. Lienkamp, „Indy Autonomous Challenge - Autonomous Race Cars at the Handling Limits“, in *International Munich Chassis Symposium 2021: Chassis.Tech plus : Proceedings* Springer Vieweg, 2021.

Non-thesis-relevant publications; Scopus/Web of Science listed (peer-reviewed)

V. Gabler, T. Stahl, G. Huber, O. Oguz and D. Wollherr, „A Game-Theoretic Approach for Adaptive Action Selection in Close Proximity Human-Robot-Collaboration“, in *IEEE International Conference on Robotics and Automation*, 2017, pp. 2897–2903, ISBN: 978-1-5090-4633-1. DOI: 10.1109/ICRA.2017.7989336.

Thesis-relevant open-source software

- [108] T. Stahl. „*GitHub* | *Graph-Based Local Trajectory Planner*“, <https://github.com/TUMFTM/GraphBasedLocalTrajectoryPlanner>. [Accessed: 2021-06-02]. 2020.
- [278] T. Stahl. „*GitHub* | *Trajectory Supervisor*“, <https://github.com/TUMFTM/TrajectorySupervisor>. [Accessed: 2021-06-02]. 2021.
- [282] A. Heilmeier, F. Christ, L. Hermansdorfer, T. Herrmann and T. Stahl. „*GitHub* | *Global Racetrajectory Optimization*“, https://github.com/TUMFTM/global_racetrajectory_optimization. [Accessed: 2021-06-02]. 2019.
- [286] T. Stahl. „*GitHub* | *Scenario Architect*“, <https://github.com/TUMFTM/ScenarioArchitect>. [Accessed: 2021-06-02]. 2020.

Supervised Student Theses

The following student theses were written within the framework of the dissertation under the supervision of the author in terms of content, technical and scientific support as well as under relevant guidance of the author. In the following, the bachelor, semester and master theses relevant and related to this dissertation are listed. Many thanks to the authors of these theses for their extensive support within the framework of this research project.

- [213] M. Baur, „Fault Detection for Robotic Systems Based on Introspective Classification,“ Bachelor’s Thesis, Technical University of Munich, 2017.
- [252] N. Stasinski, „Identification and Characterization of Safety Relevant Criteria of Trajectories,“ Bachelor’s Thesis, Technical University of Munich, 2019.
- [261] N. Rack, „Entwicklung einer Methode zur prädiktionsbasierten Bestimmung definitiv belegter Zustände dynamischer Objekte,“ Semester Thesis, Technical University of Munich, 2020.
- [262] P. Radecker, „Entwicklung einer Methode zur prädiktionsbasierten Bestimmung definitiv belegter Zustände,“ Semester Thesis, Technical University of Munich, 2020.
- [272] M. Bayerlein, „Kollisionserkennung für Trajektorien eines autonomen Fahrzeugs im dynamischen Umfeld,“ Semester Thesis, Technical University of Munich, 2020.
M. Abdelmeguid, „Risk Assessment of Trajectories of an Autonomous Vehicle Regarding Static Factors,“ Semester Thesis, Technical University of Munich, 2019.
A. Candussio, „Entwicklung einer Methode zur Bewegungsprädiktion autonomer Rennfahrzeuge,“ Master’s Thesis, Technical University of Munich, 2018.
P. Enzinger, „Development and Evaluation of Deep Learning-Based Planners for Autonomous Racing,“ Master’s Thesis, Technical University of Munich, 2021.
M. Euler, „Optimierung von Überholmanövern eines Trajektorienplaners für autonome Rennfahrzeuge,“ Semester Thesis, Technical University of Munich, 2019.
P. George, „Gefahrenanalyse und Risikobewertung modularer Funktionsgruppen autonomer Fahrzeuge,“ Semester Thesis, Technical University of Munich, 2019.
T. Kosmata, „Entwicklung der Pfadplanung eines autonomen Rennfahrzeugs,“ Semester Thesis, Technical University of Munich, 2019.
V. A. Le Cong, „Concept Analysis of an Artificial-Intelligence-Based Supervisor for Safety Assessment of Autonomous Vehicles,“ Master’s Thesis, Technical University of Munich, 2020.
Y. Lian, „Risk Assessment of Trajectories of an Autonomous Vehicle in a Dynamic Environment,“ Semester Thesis, Technical University of Munich, 2019.






Appendix

A	Appendix	xliii
A.1	Literature Evaluation Metrics	xliii
A.2	Interface Testing	xliv
A.3	Ground Truth Generation Parameters	xliv
A.4	Test Tracks	xlvi
A.5	Scenarios	xlviii

A Appendix

A.1 Literature Evaluation Metrics

Table A.1: Evaluation criteria and their scale for the analysis of the relevant literature.

					
Approval/standards	Approval not mentioned	Approval / formal correctness mentioned, but not further tackled	Approval addressed - little insights/references	Approval addressed - key points elaborated/implemented	Approval addressed - fully elaborated/implemented
Holistic approach	Only one aspect addressed	Multiple aspects addressed	Extensive coverage, holistic coverage not advertised	Holistic coverage advertised, but no systematic approach	Systematic approach for holistic coverage
Degree of maturity	High-level concept only	Partially detailed/implemented concept (selected sub-modules)	Detailed theoretical concept (top to bottom)	Fully implemented approach, not real-time capable	Fully implemented approach, real-time capable
Extent of testing	Theoretical elaboration only	Prototype tests / partial tests	Evaluation in simple simulation	Evaluation in detailed simulation / scale vehicle	Evaluation on real vehicle

A.2 Interface Testing

Table A.2: Evaluated interface tests with associated scores (passed test (PT) or passed test given assumptions (TPGA)) based on expected and observed output.

Test class	Test description	Input specifics	Expected output	Observed output	PT	PTGA
Normal operation	Placed on straight, no objects	-	Vehicle following trajectory	Vehicle following trajectory	True	True
	Placed on straight, TP ahead	-	Vehicle following trajectory	Vehicle following trajectory	True	True
	Placed on turn, no objects	-	Vehicle following trajectory	Vehicle following trajectory	True	True
	Placed on turn, object ahead	-	Vehicle following trajectory	Vehicle following trajectory	True	True
Missing data	Missing map	Map server not launched	Vehicle not launching	Vehicle not launching	True	True
	Missing localization	Localization module terminated during execution	Vehicle decelerating on emergency trajectory	Vehicle decelerating on emergency trajectory	True	True
	Missing object data	Object server terminated during execution	Vehicle decelerating on emergency trajectory	Vehicle decelerating on emergency trajectory	True	True
	Trajectory planner stalled	Trajectory planner terminated during execution	Vehicle decelerating on emergency trajectory	Vehicle decelerating on emergency trajectory	True	True
Supervisor stalled	Supervisor stalled	Supervisor terminated during execution	Vehicle decelerating on emergency trajectory	Vehicle decelerating on emergency trajectory	True	True
	Map in wrong format	Fault-injection - sending map with altered format	Vehicle not launching	Vehicle not launching	True	True
	Map hosting wrong information	Fault-injection - sending map for different location	Vehicle not launching	Fault not detected due to assumption of reliable perception data	False	True
	Localization in wrong format	Fault-injection - sending localization with altered format	Module(s) exiting, vehicle decelerating on emergency trajectory	Planner exiting, vehicle decelerating on emergency trajectory	True	True
Localization hosting wrong pose	Localization hosting wrong pose	Fault-injection - sending localization for arbitrary pose	Vehicle decelerating on emergency trajectory	Uncertainty in pose estimate, decelerating on emergency trajectory	True	True
	Object data in wrong format	Fault-injection - sending object data with altered format	Module(s) exiting, vehicle decelerating on emergency trajectory	Planner exiting, vehicle decelerating on emergency trajectory	True	True
	Object data hosting wrong information	Fault-injection - sending arbitrary pose for one object	Vehicle decelerating on emergency trajectory	Fault not detected due to assumption of reliable perception data	False	True
	Trajectory in wrong format	Fault-injection - sending trajectory with altered format	Module(s) exiting, vehicle decelerating on emergency trajectory	Supervisor rating unsafe, vehicle decelerating on emergency trajectory	True	True
Trajectory hosting wrong information	Trajectory hosting wrong information	Fault-injection - sending random trajectory	Vehicle decelerating on emergency trajectory	Supervisor rating unsafe, vehicle decelerating on emergency trajectory	True	True
	Delayed map transmission	Delay map for 500 ms	Vehicle not launching	Vehicle not launching	True	True
	Delayed localization update	Delay localization update for 500 ms	Vehicle decelerating on latest emergency trajectory	Watchdog timeout, vehicle decelerating on latest emergency trajectory	True	True
	Delayed object-data update	Delay object-data update for 500 ms	Vehicle decelerating on latest emergency trajectory	Watchdog timeout, vehicle decelerating on latest emergency trajectory	True	True
Timing variations	Delayed trajectory update	Delay trajectory update for 500 ms	Vehicle decelerating on latest emergency trajectory	Watchdog timeout, vehicle decelerating on latest emergency trajectory	True	True

A.3 Ground Truth Generation Parameters

Table A.3: Crucial parameters (with explanation) for the generation of the ground truth in multi-vehicle scenarios. For most safety parameters, an upper and lower bound is specified to define the three ranges: safe, unspecified rating, unsafe.

Parameter	Description	Value
SSM temporal increment	Temporal discretization for the ground truth evaluation along the trajectory.	0.1 s
DSS t_{react}	Maximum reaction time assumed for the ego-vehicle for the DSS calculation, here representing the worst-case calculation time of the SW stack.	0.1 s
DSS a_{max}	Maximum acceleration assumed for the lead vehicle for the DSS calculation, here maximum acceleration possible for a given track is used.	track specific maximum
DSS unsafe	Unsafe threshold for the DSS-based ground truth generation. Any value below this threshold is expected to be rated as unsafe by the Supervisor.	0.0 m
DSS safe	Safe threshold for the DSS-based ground truth generation. Any value above this threshold is expected to be rated as safe by the Supervisor.	5.0 m
Vehicle inflation factor	Inflation of vehicle shape for track collision checks. If an intersection with the track boundaries occurs with the native vehicle footprint the Supervisor is expected to rate the trajectory as unsafe, if an intersection with the track boundaries does not occur with the inflated footprint the Supervisor is expected to rate the trajectory as safe, in between the Supervisor may post any rating.	1.5
Combined acceleration factor	Factor for the combined acceleration acting on the tires. If the track specific limits are taken into account, the ground truth is set as safe, otherwise, if the track specific limits scaled by the specified factor is violated, the Supervisor is expected to rate the trajectory as unsafe.	1.2
Machine acceleration limit factor	Factor for the machine acceleration limit. If the specified machine acceleration limits are respected, the Supervisor is expected to rate the trajectory as safe, otherwise, if the machine acceleration limit scaled by the specified factor is violated, the Supervisor is expected to rate the trajectory as unsafe.	1.2
Turn radius unsafe	Minimum turn radius. If the trajectory hosts a curvature resulting in a turn radius below this threshold, the Supervisor is expected to rate the trajectory as unsafe.	11.0 m
Turn radius safe	Unproblematic turn radius. If the trajectory does not host a curvature resulting in a turn radius below this threshold, the Supervisor is expected to rate the trajectory as safe.	13.0 m

A.4 Test Tracks

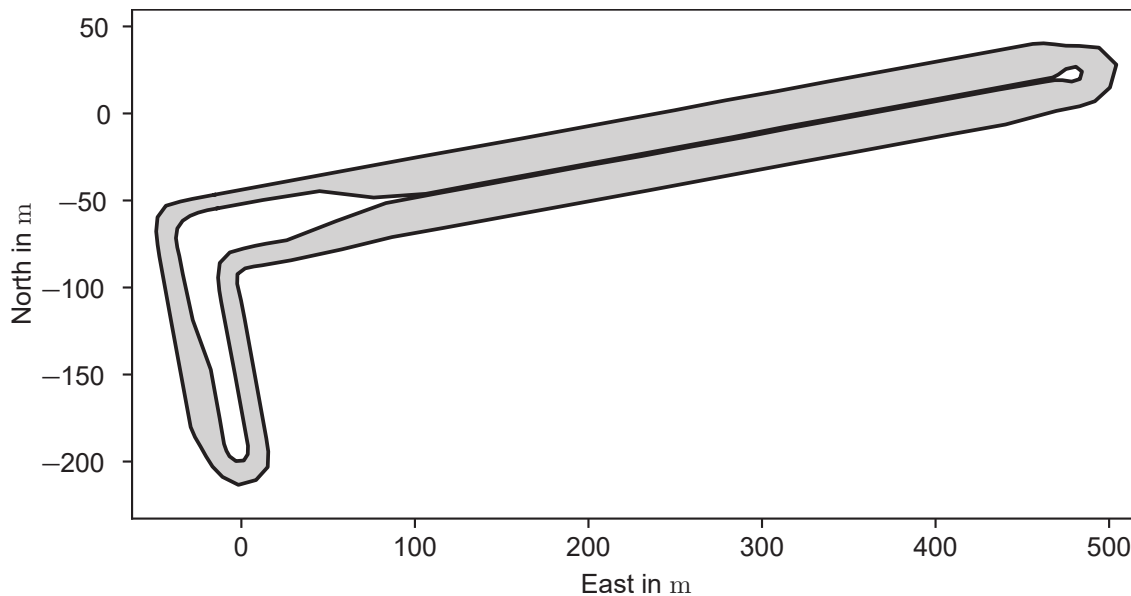


Figure A.1: Test track in Upper Heyford, United Kingdom. The origin was arbitrarily chosen in accordance with the reference pose of the DGPS system.

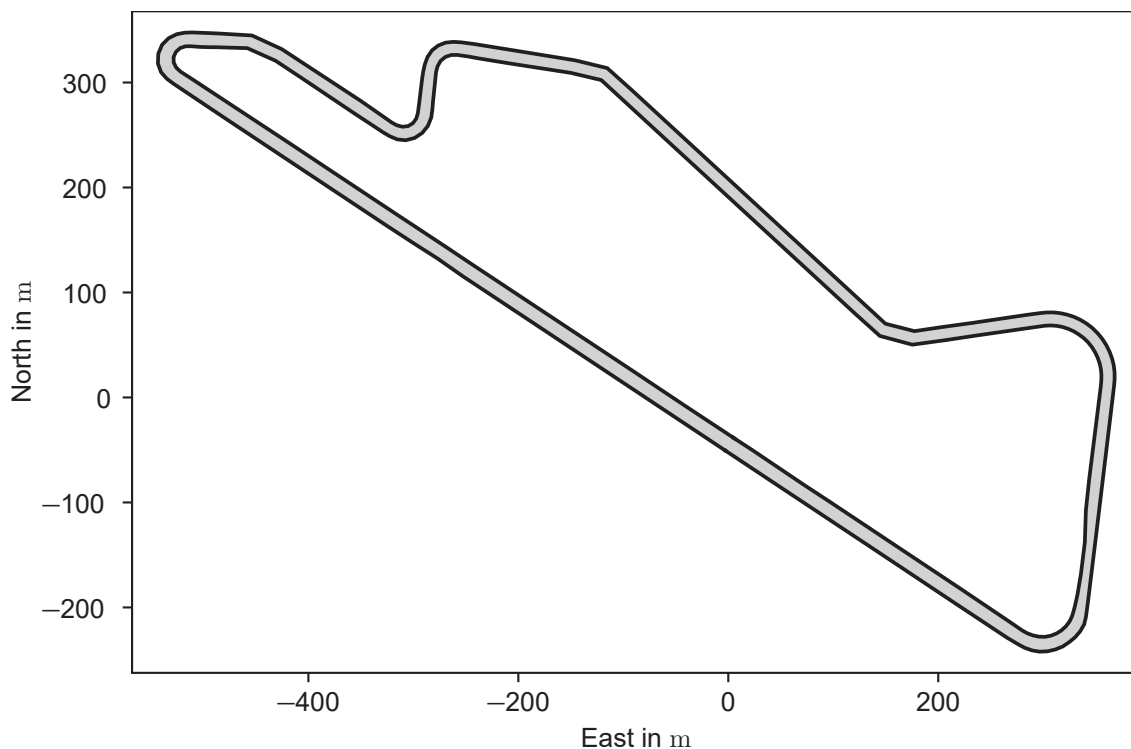


Figure A.2: Test track in Monteblanco, Spain. The origin was arbitrarily chosen in accordance with the reference pose of the DGPS system.

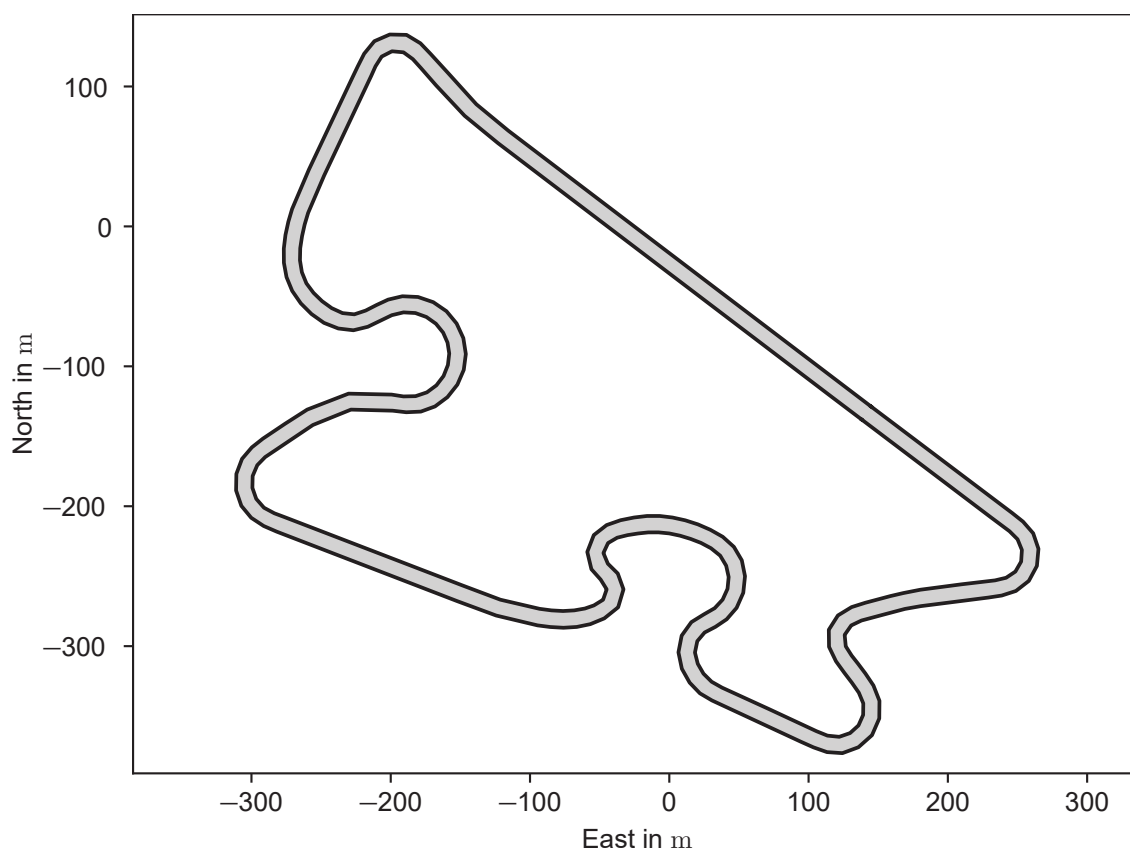


Figure A.3: Test track in Modena, Italy. The origin was arbitrarily chosen in accordance with the reference pose of the DGPS system.

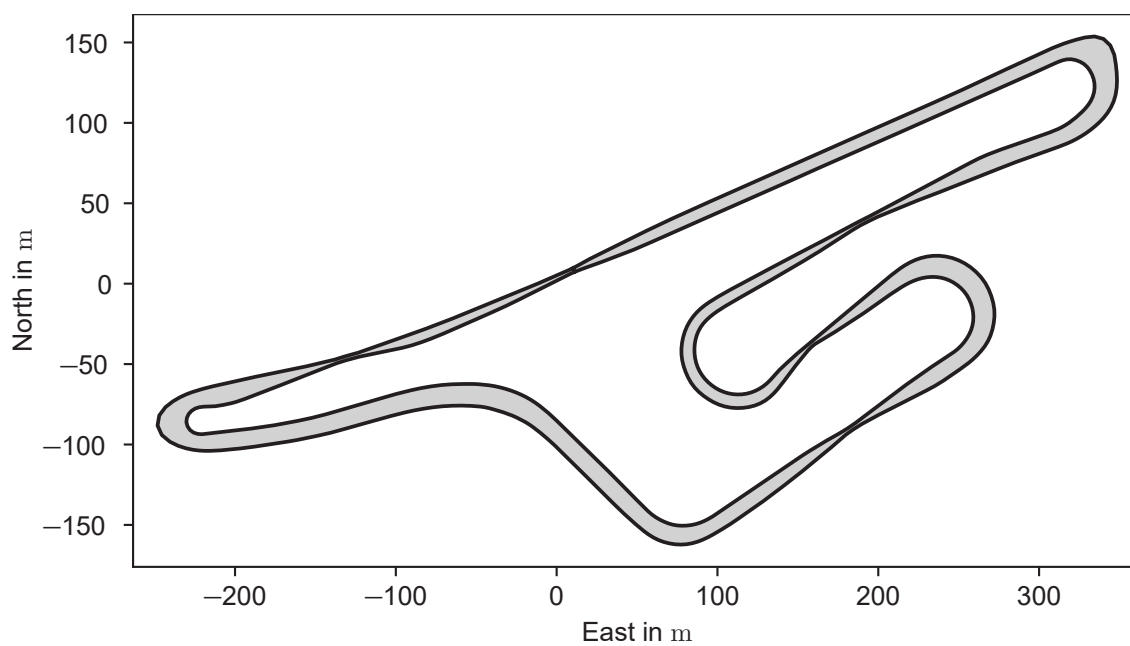


Figure A.4: Test track in Le Croix, France. The origin was arbitrarily chosen in accordance with the reference pose of the DGPS system. The narrows of the track were artificially generated by cones.

A.5 Scenarios

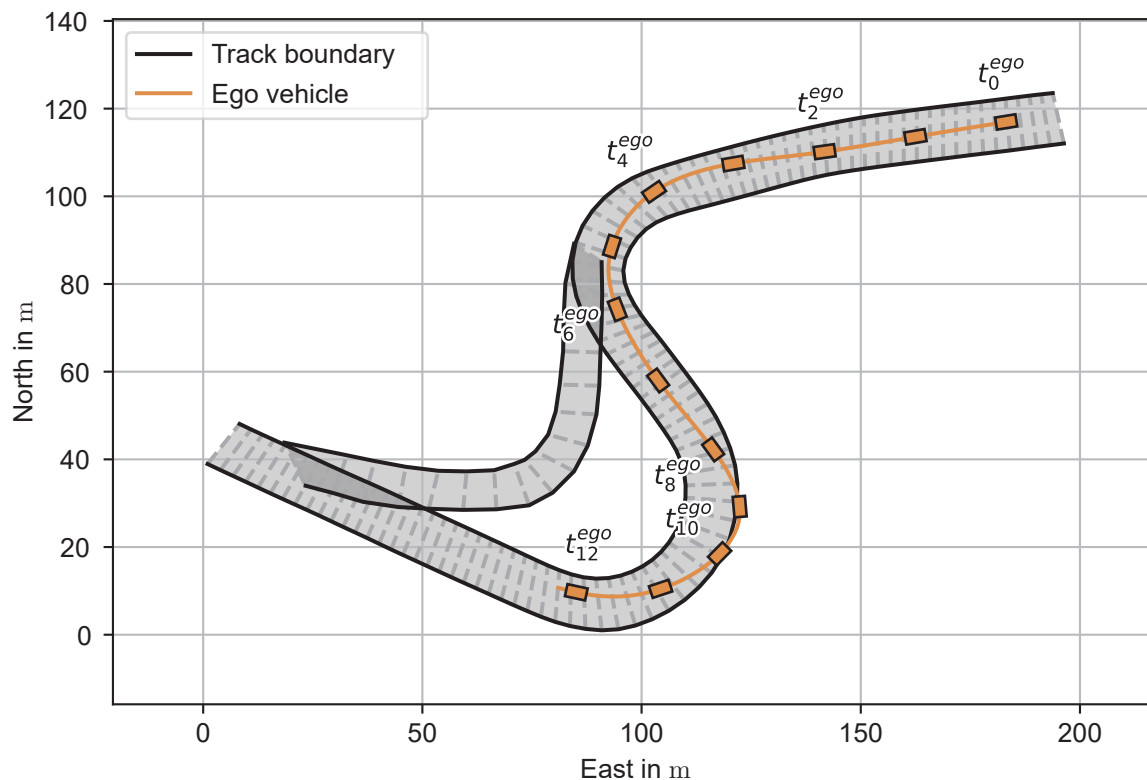


Figure A.5: Bird's eye view of scenario S1. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

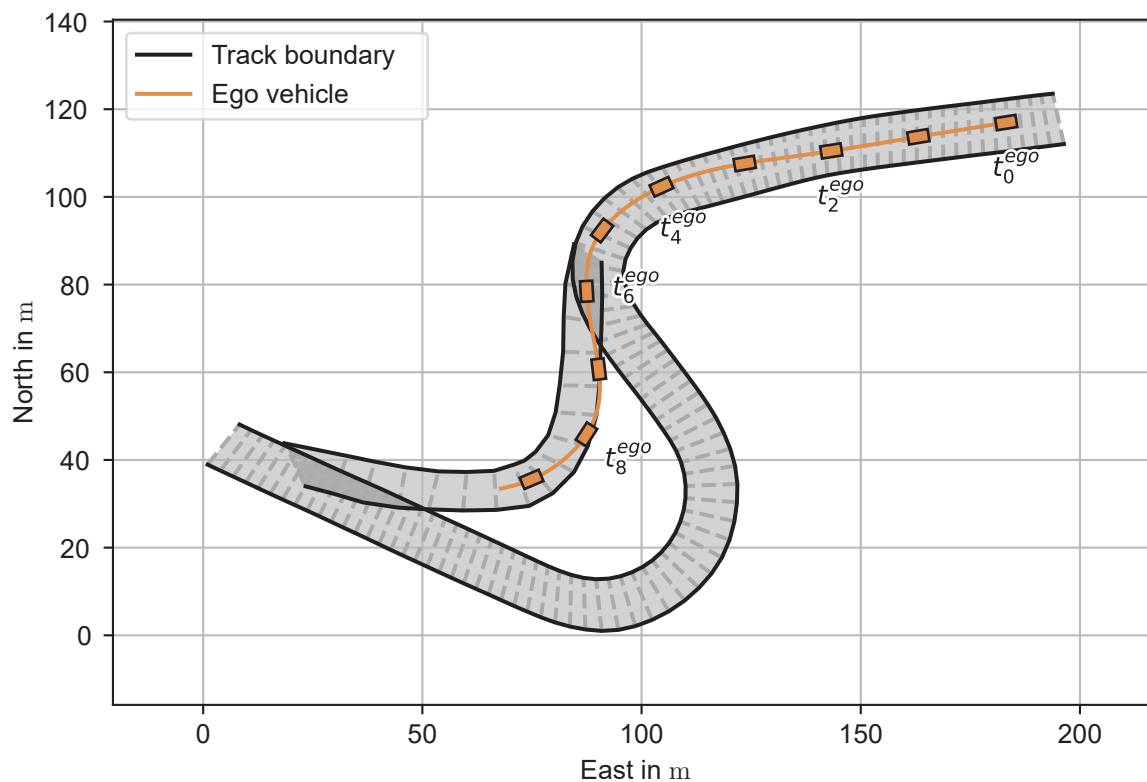


Figure A.6: Bird's eye view of scenario S2. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

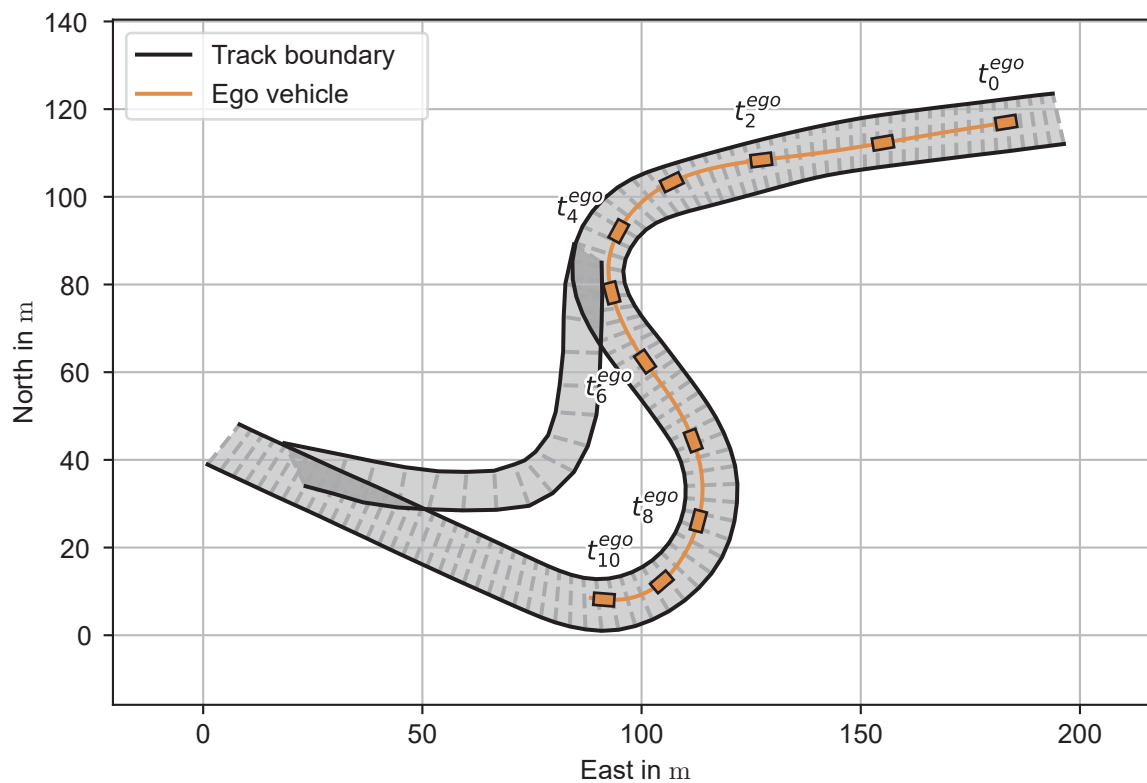


Figure A.7: Bird's eye view of scenario S3. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

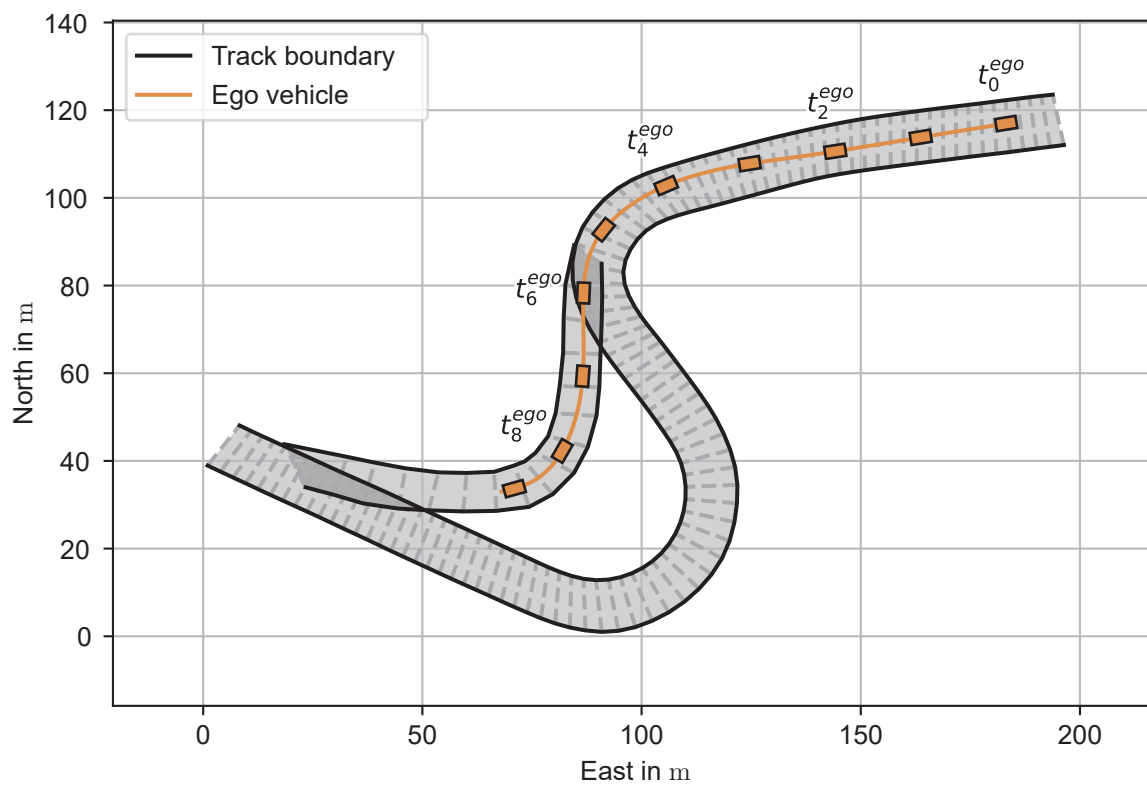


Figure A.8: Bird's eye view of scenario S4. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

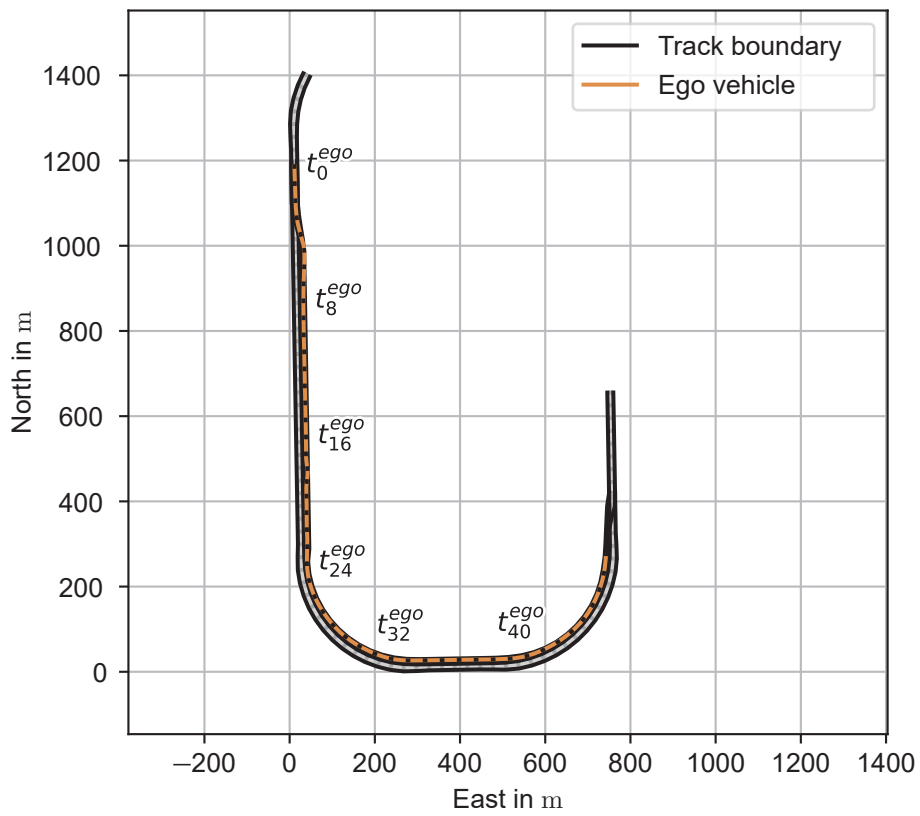


Figure A.9: Bird's eye view of scenario S5. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every eighth interval is marked with a text label.

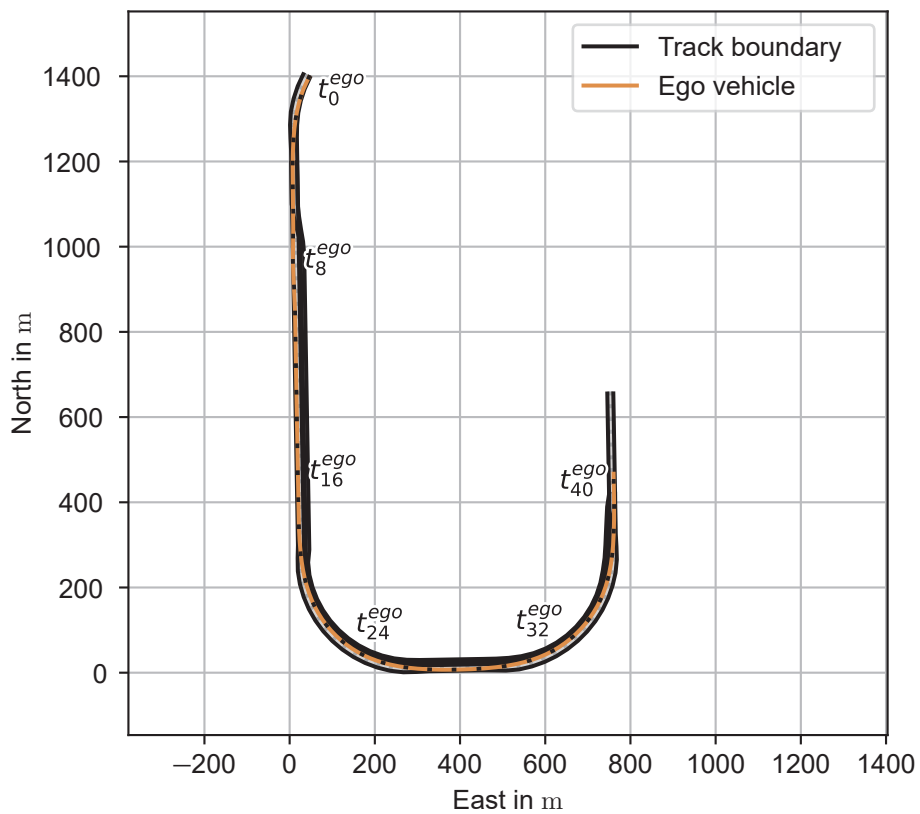


Figure A.10: Bird's eye view of scenario S6. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every eighth interval is marked with a text label.

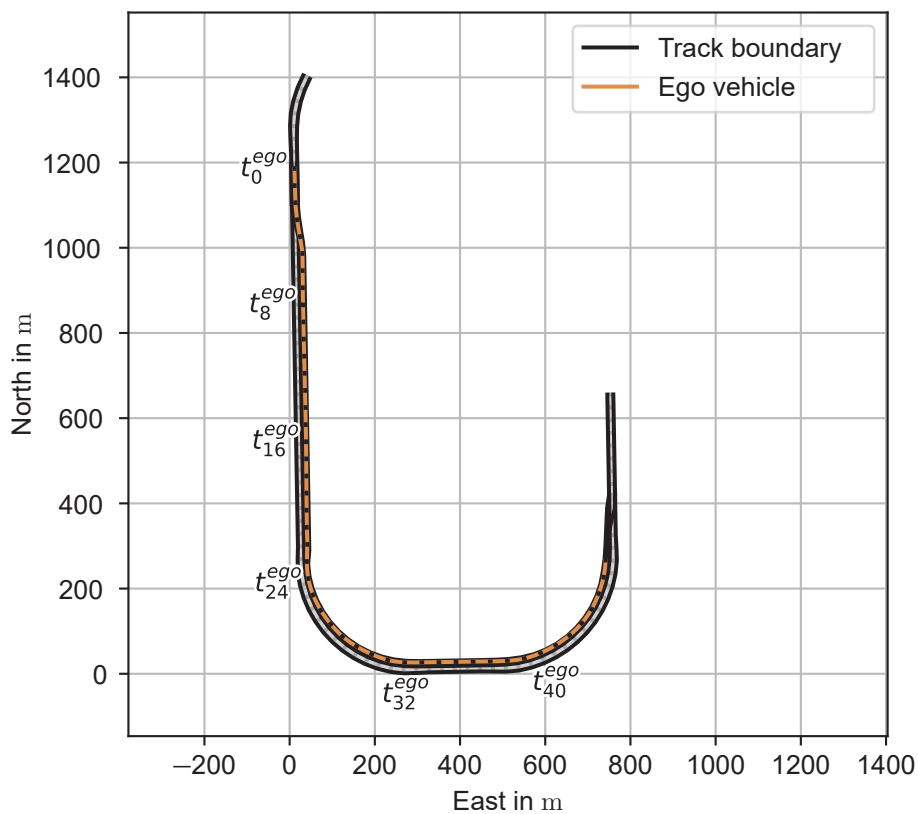


Figure A.11: Bird's eye view of scenario S7. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every eighth interval is marked with a text label.

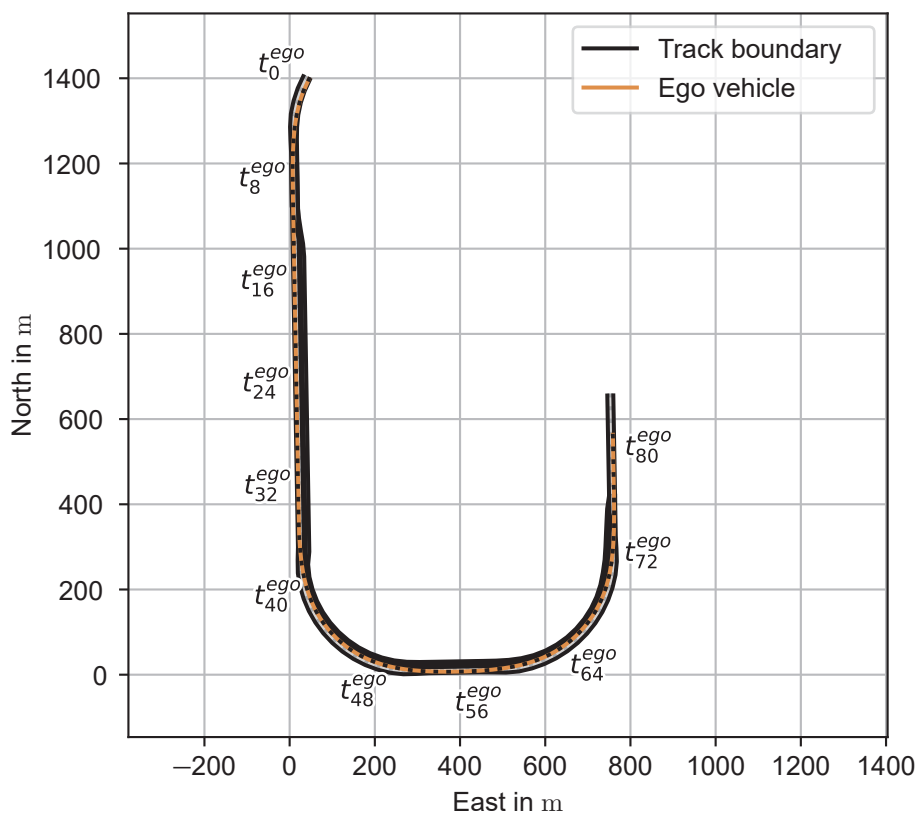


Figure A.12: Bird's eye view of scenario S8. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every eighth interval is marked with a text label.

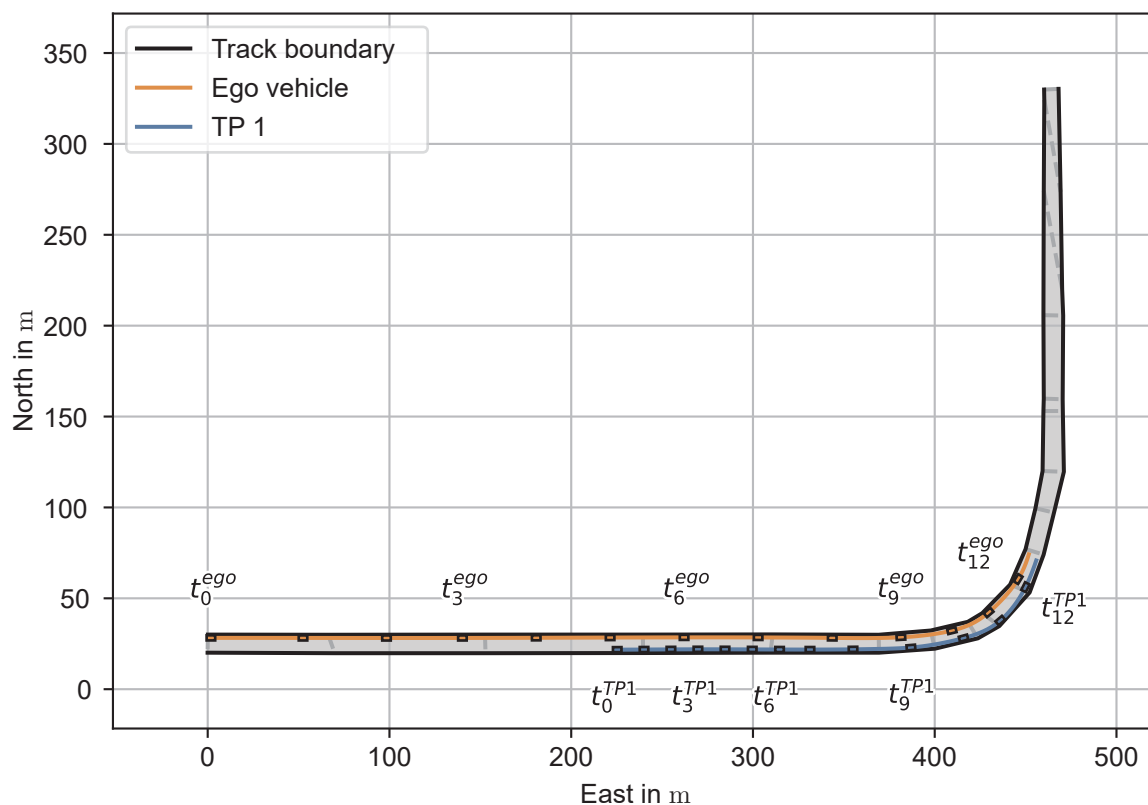


Figure A.13: Bird's eye view of scenario S9. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

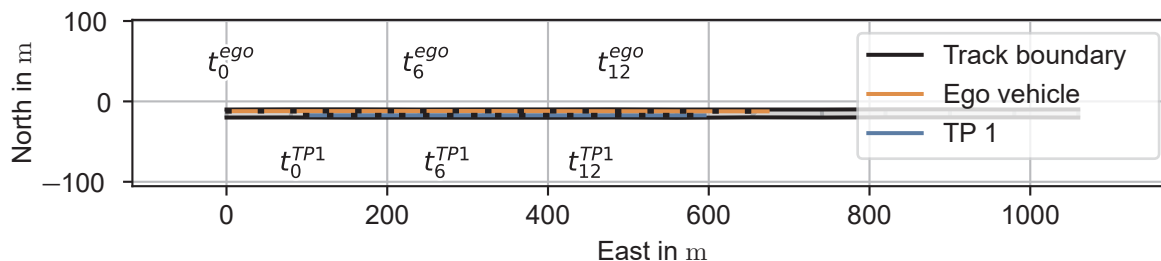


Figure A.14: Bird's eye view of scenario S10. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every sixth interval is marked with a text label.

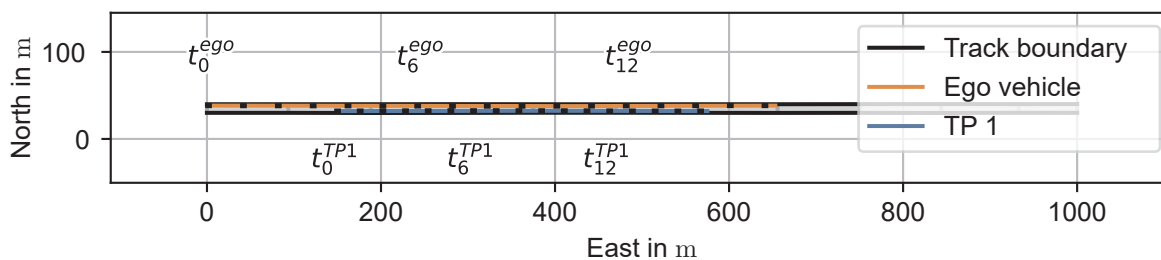


Figure A.15: Bird's eye view of scenario S11. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every sixth interval is marked with a text label.

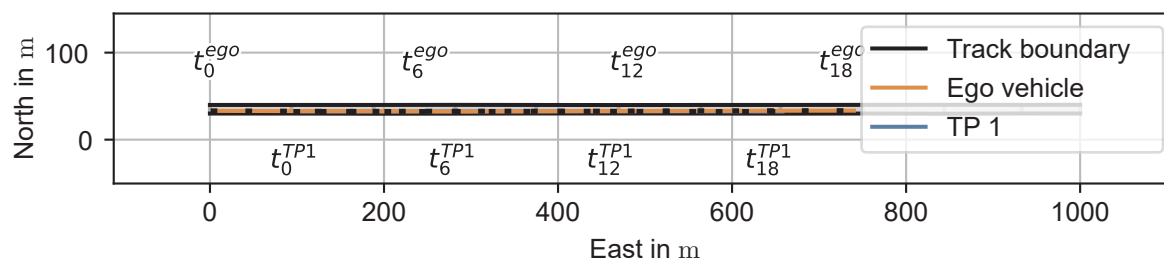


Figure A.16: Bird's eye view of scenario S12. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every sixth interval is marked with a text label.

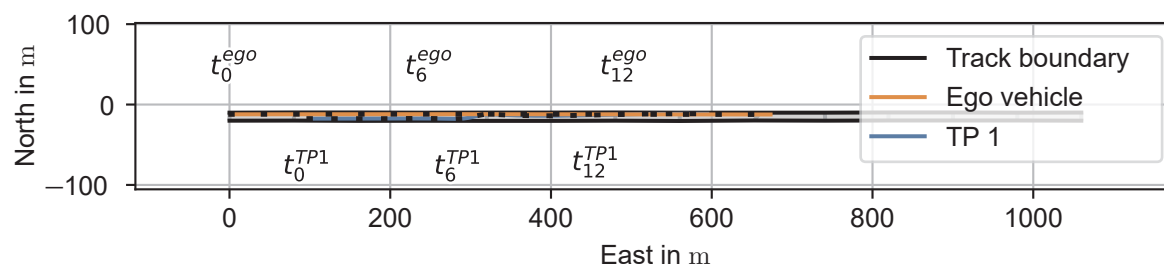


Figure A.17: Bird's eye view of scenario S13. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every sixth interval is marked with a text label.

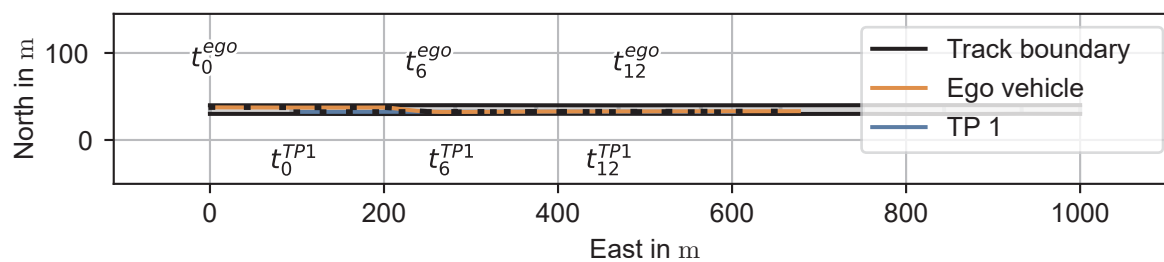


Figure A.18: Bird's eye view of scenario S14. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every sixth interval is marked with a text label.

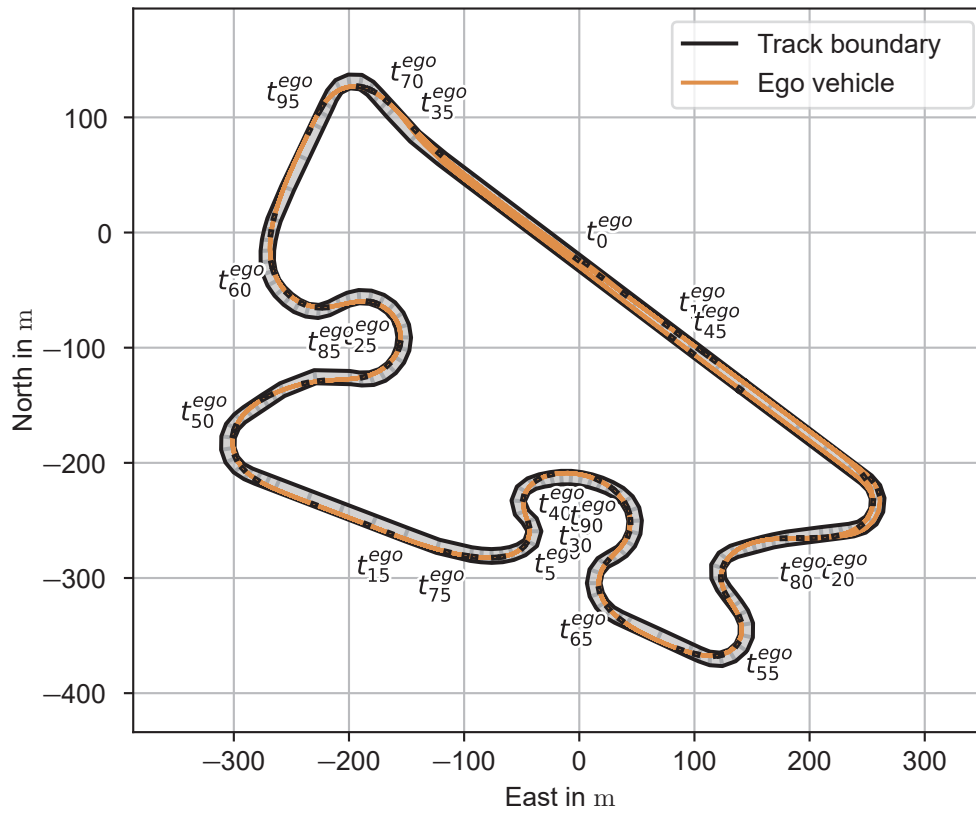


Figure A.19: Bird's eye view of scenario S15. The vehicle movements are shown via trace (line) and vehicle shape samples at 10 s interval. Every fifth interval is marked with a text label.

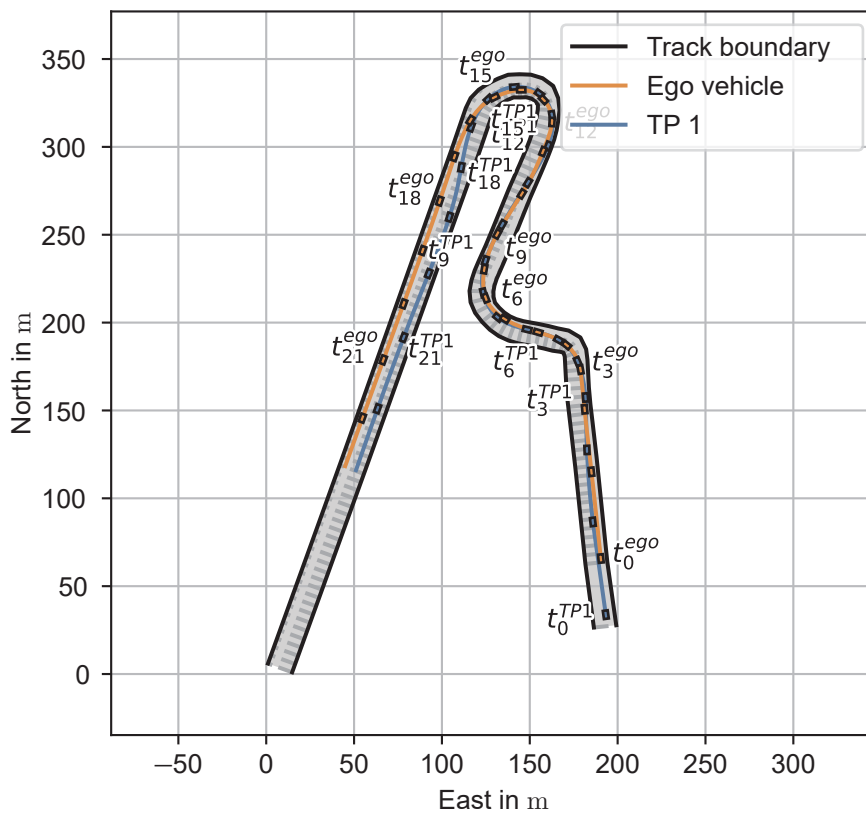


Figure A.20: Bird's eye view of scenario S16. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

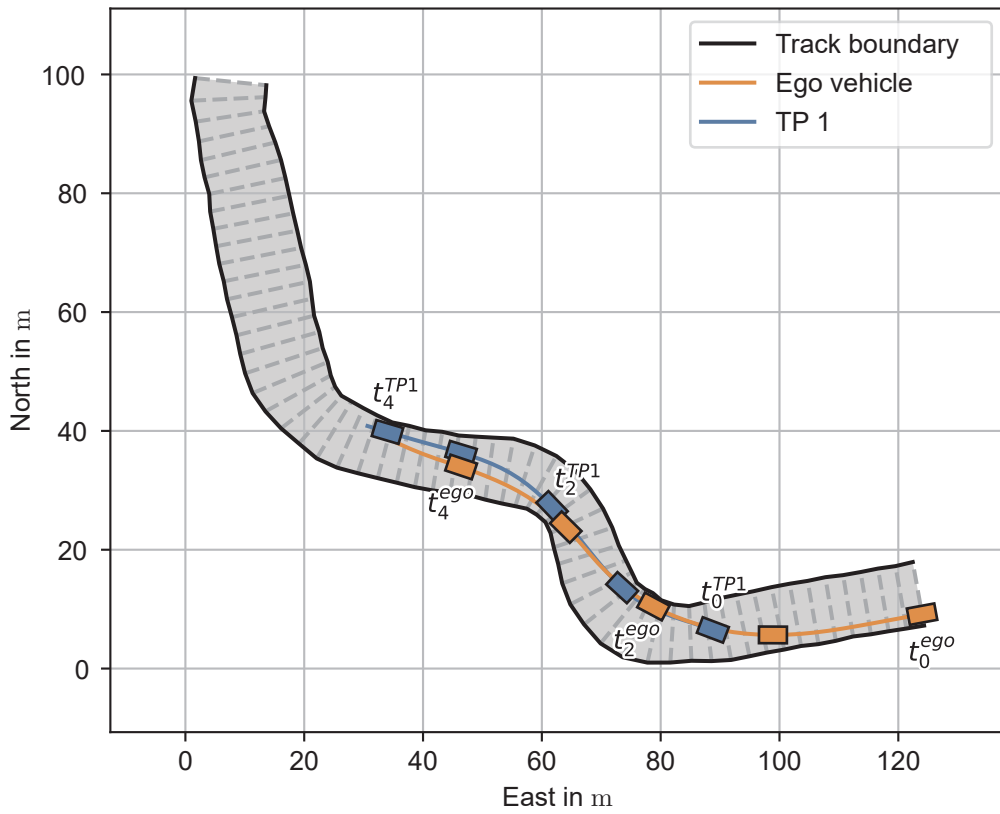


Figure A.21: Bird's eye view of scenario S17. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

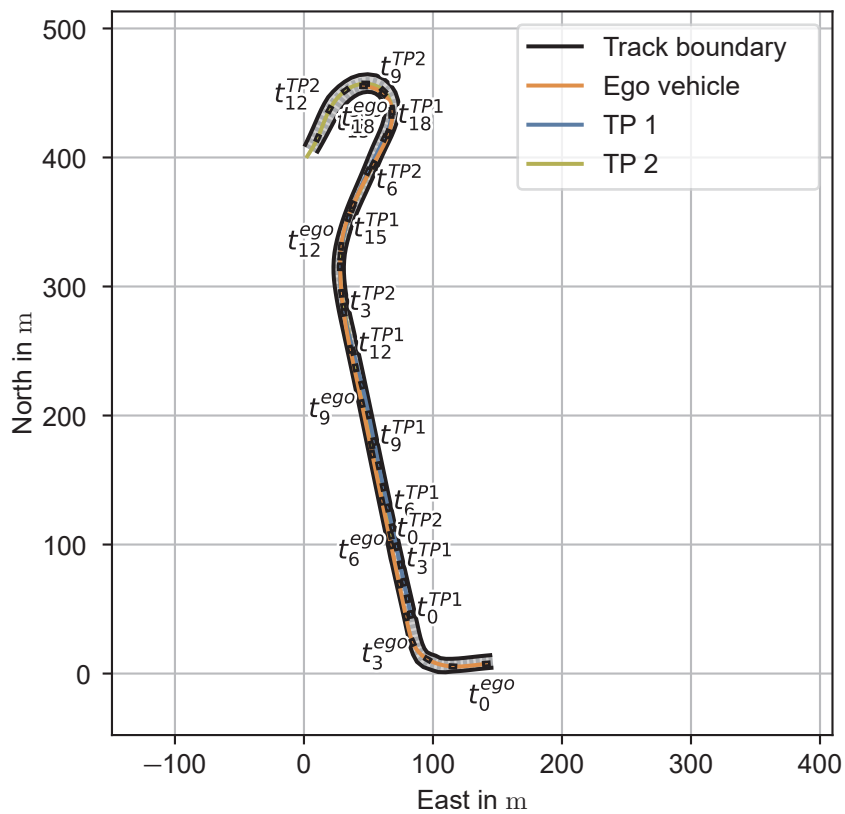


Figure A.22: Bird's eye view of scenario S18. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

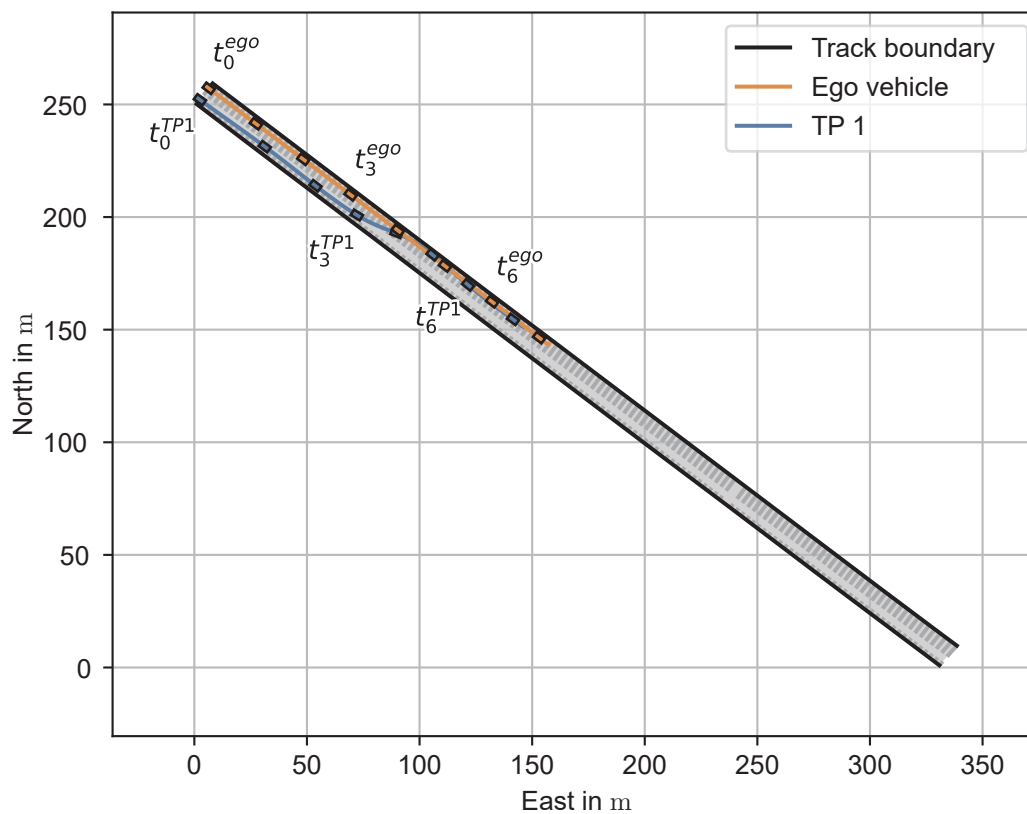


Figure A.23: Bird's eye view of scenario S19. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

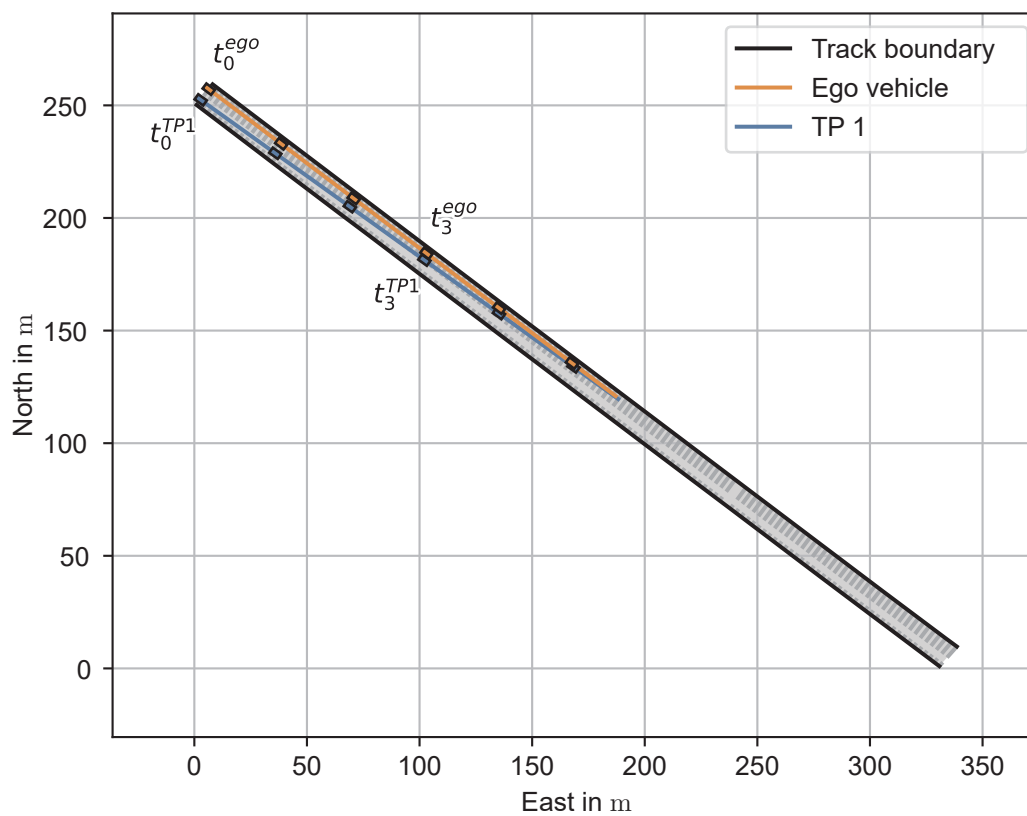


Figure A.24: Bird's eye view of scenario S20. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

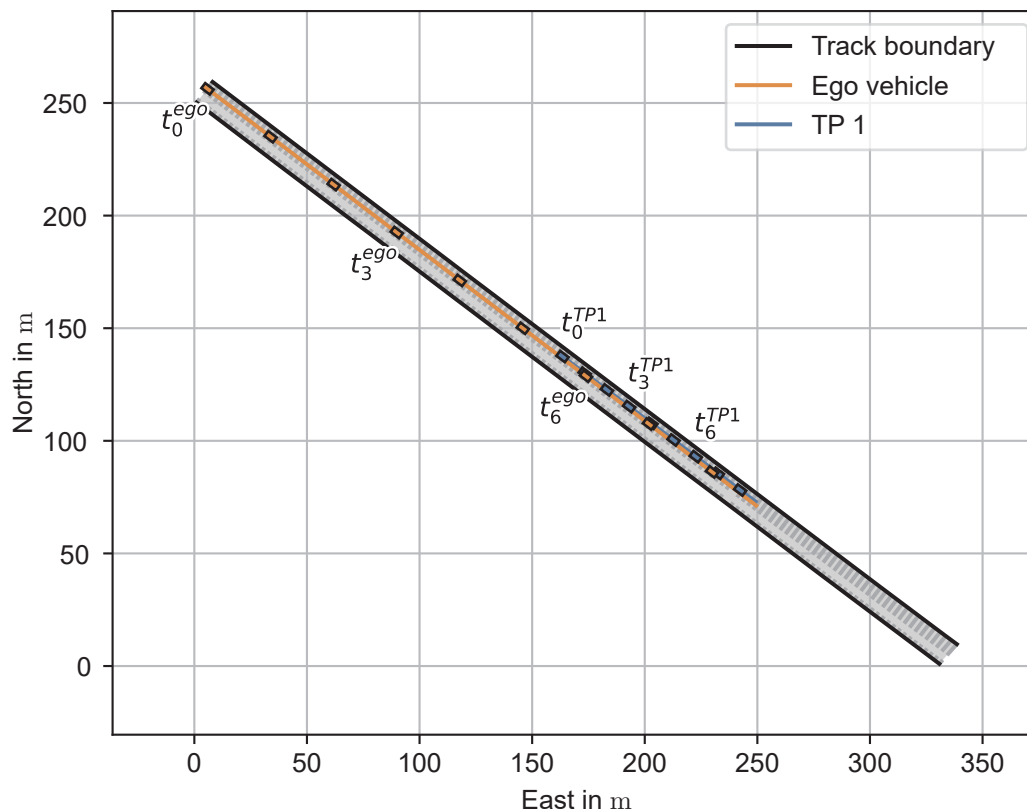


Figure A.25: Bird's eye view of scenario S21. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

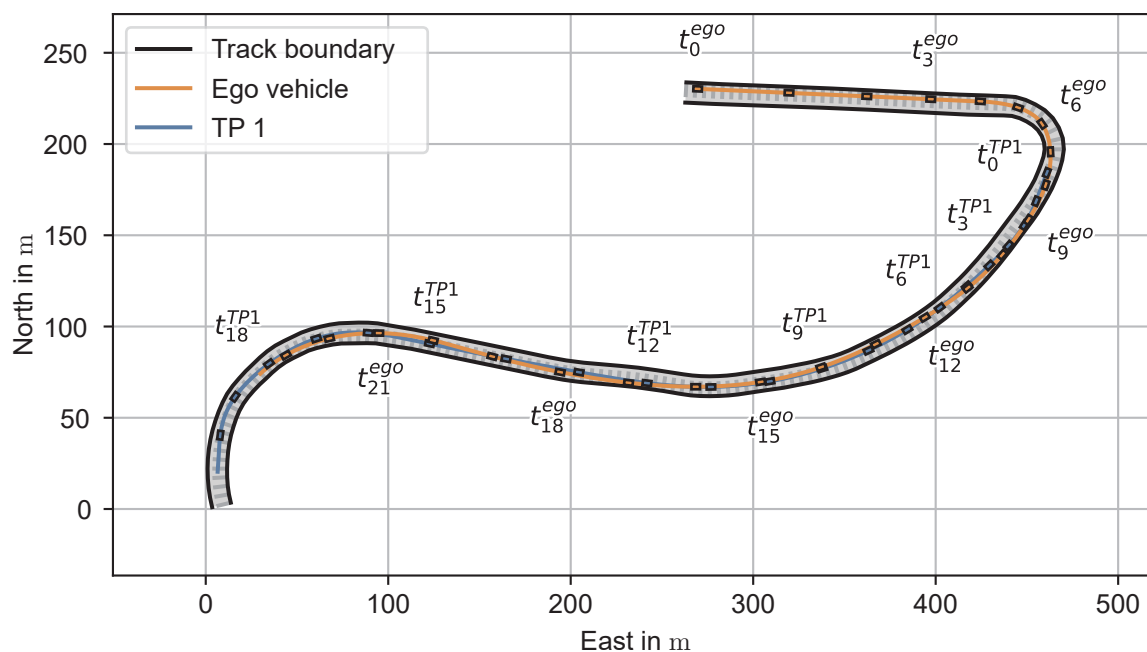


Figure A.26: Bird's eye view of scenario S22. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

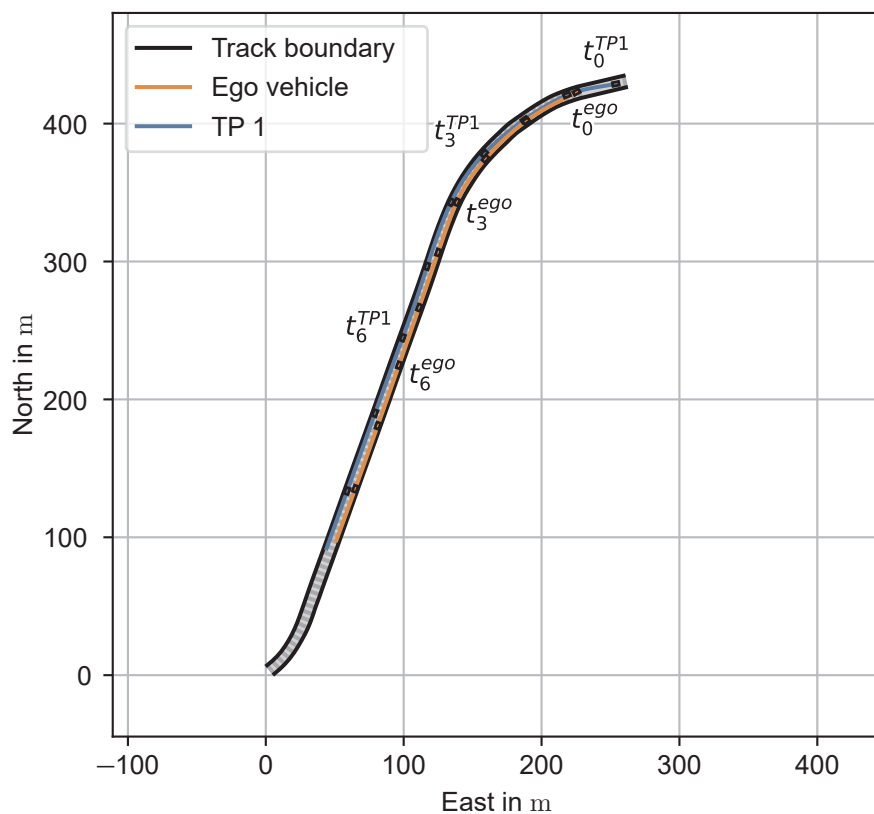


Figure A.27: Bird's eye view of scenario S23. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

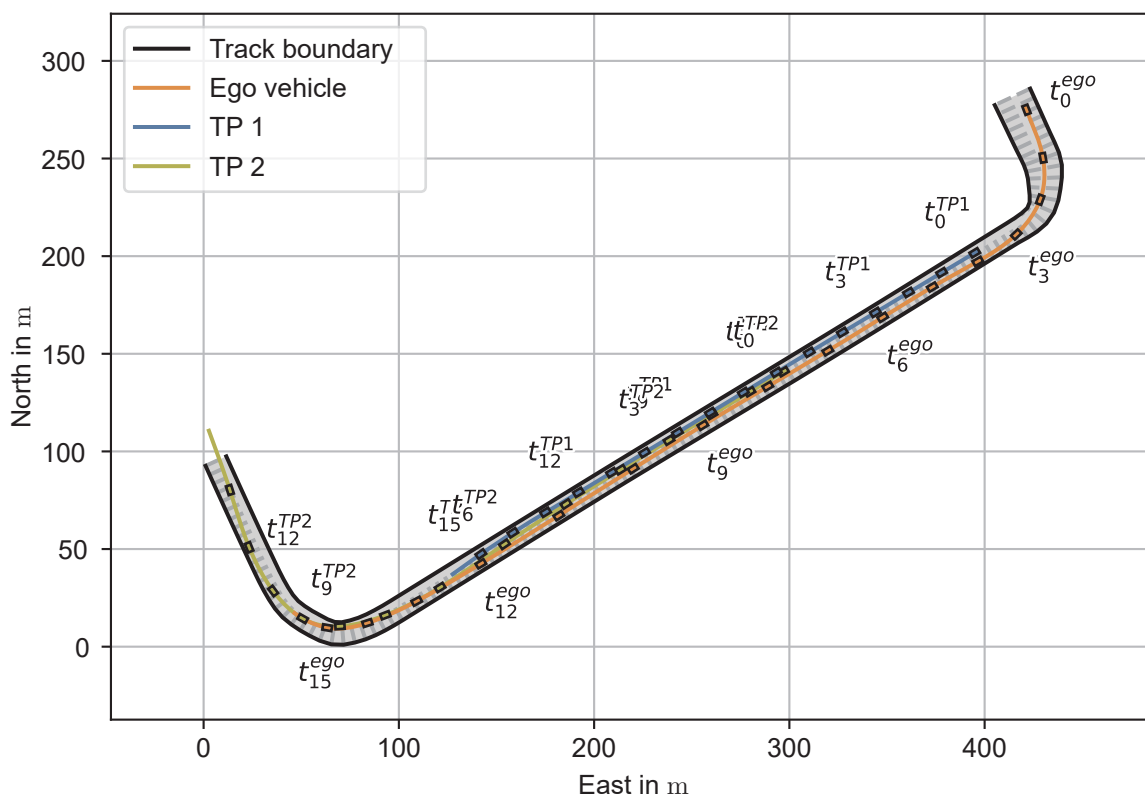


Figure A.28: Bird's eye view of scenario S24. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

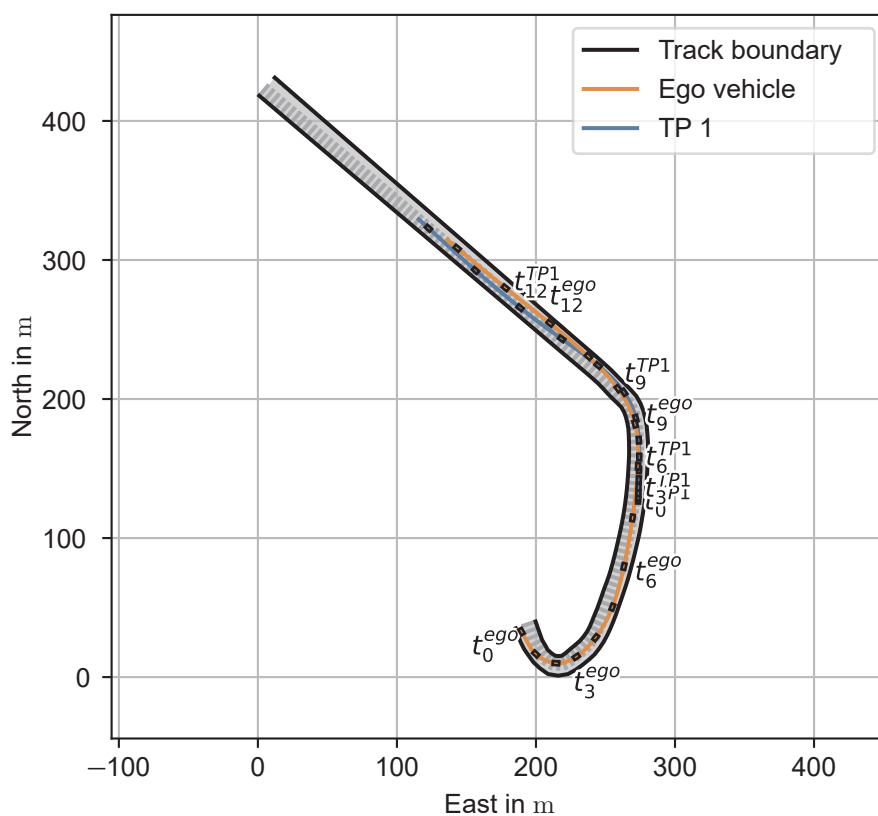


Figure A.29: Bird's eye view of scenario S25. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

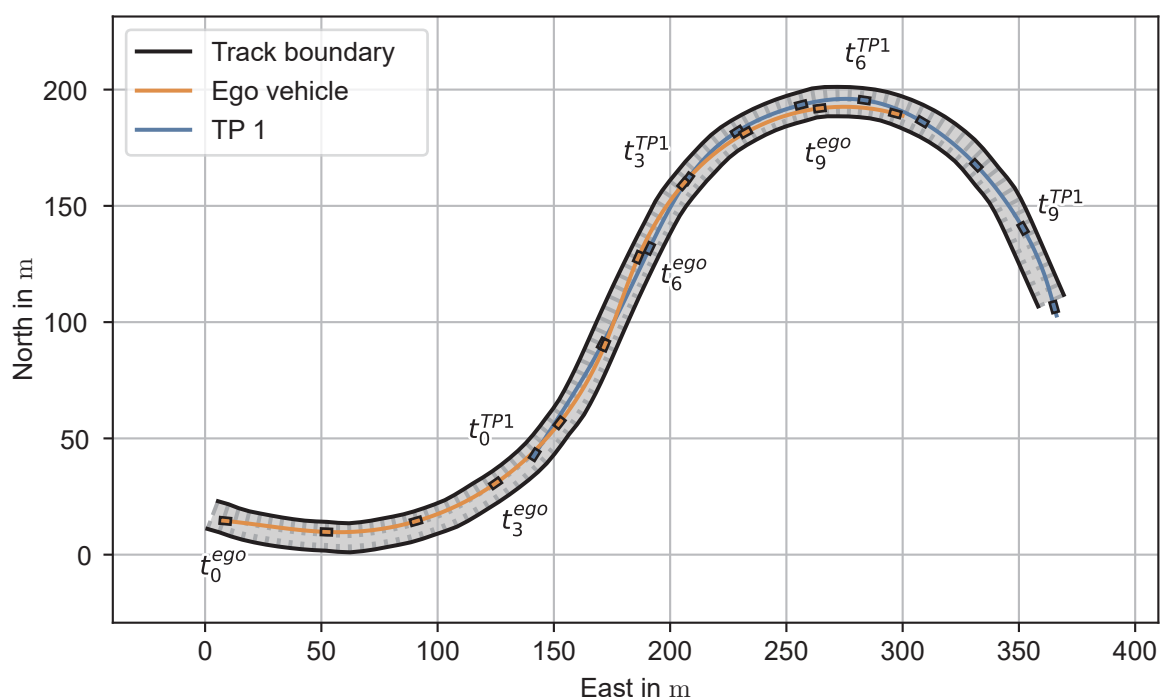


Figure A.30: Bird's eye view of scenario S26. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

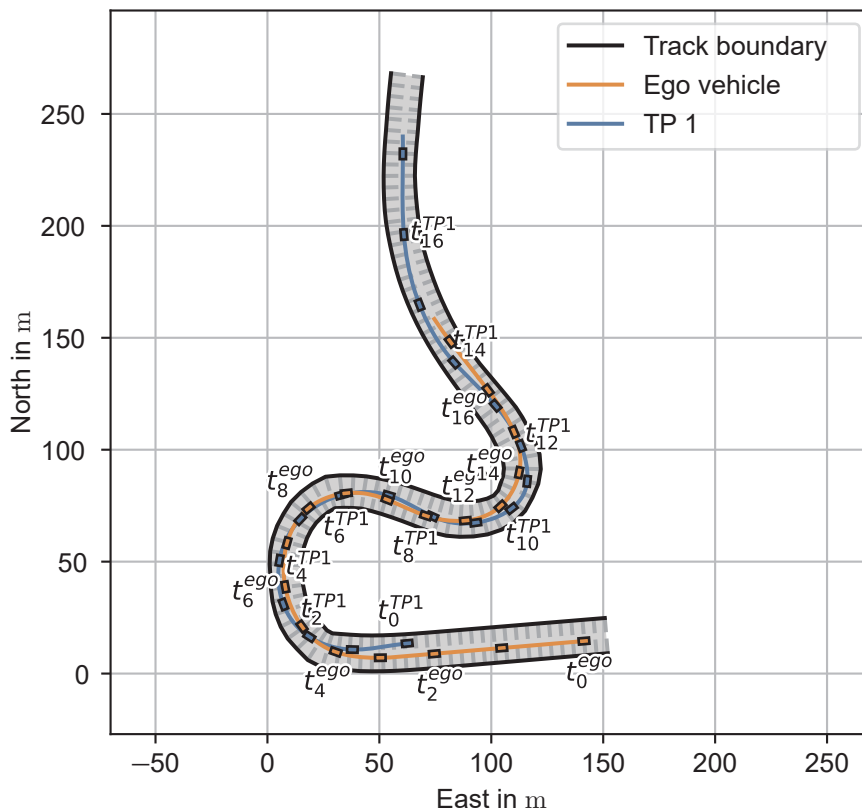


Figure A.31: Bird's eye view of scenario S27. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

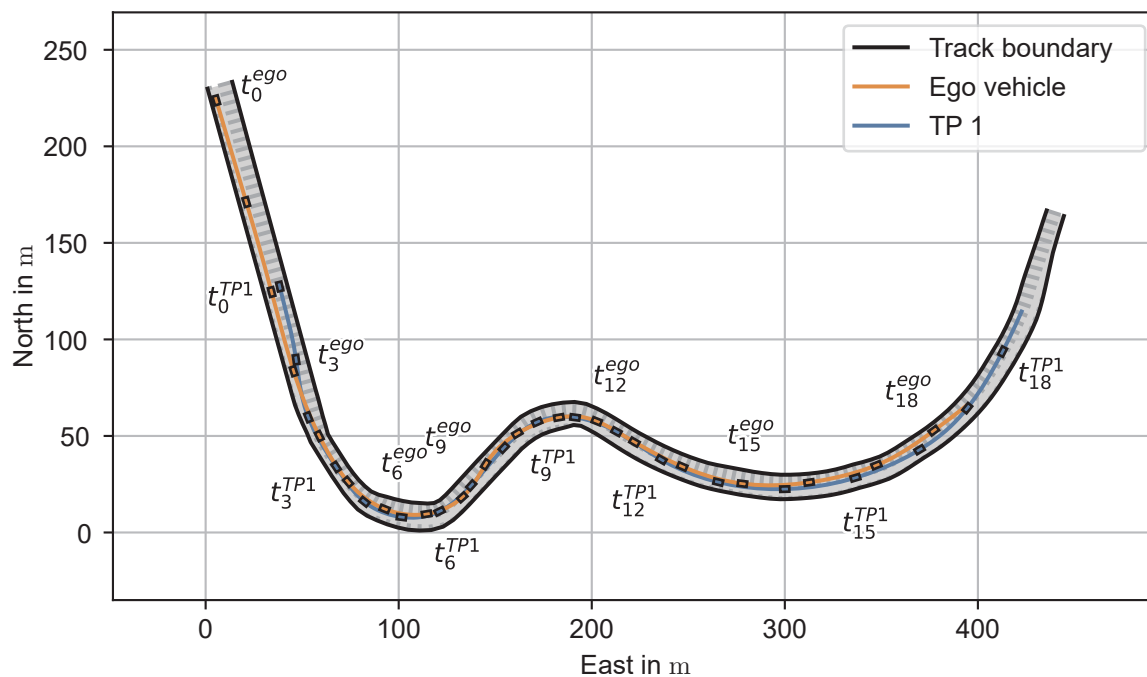


Figure A.32: Bird's eye view of scenario S28. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

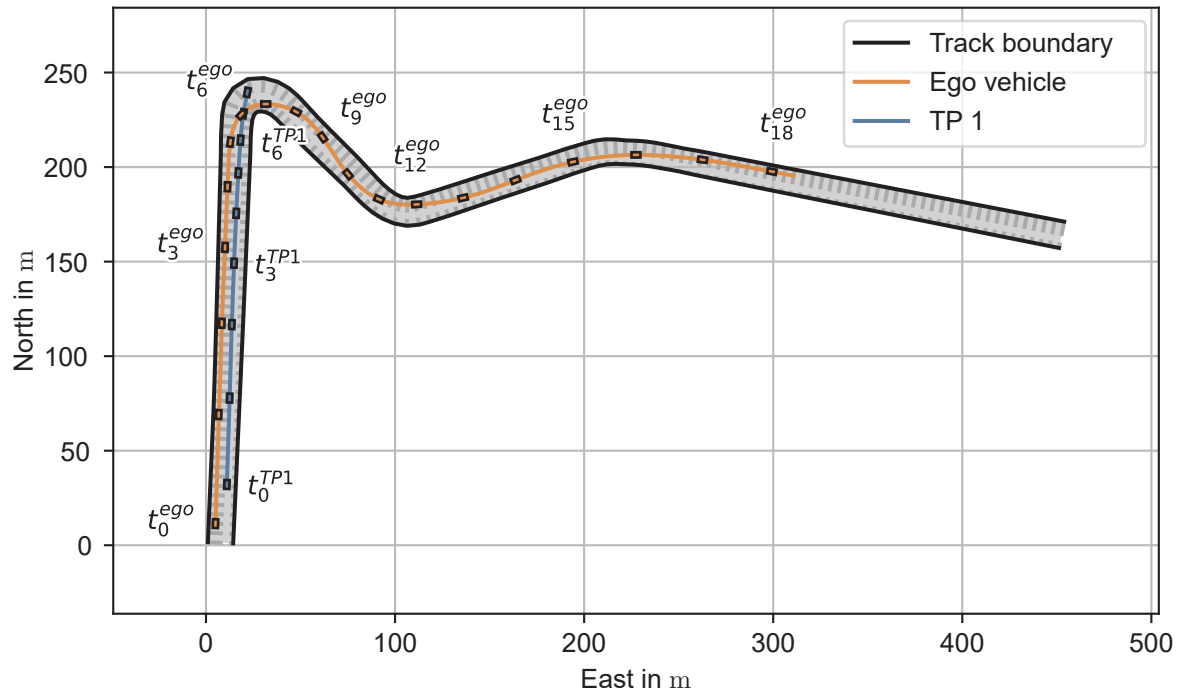


Figure A.33: Bird's eye view of scenario S29. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

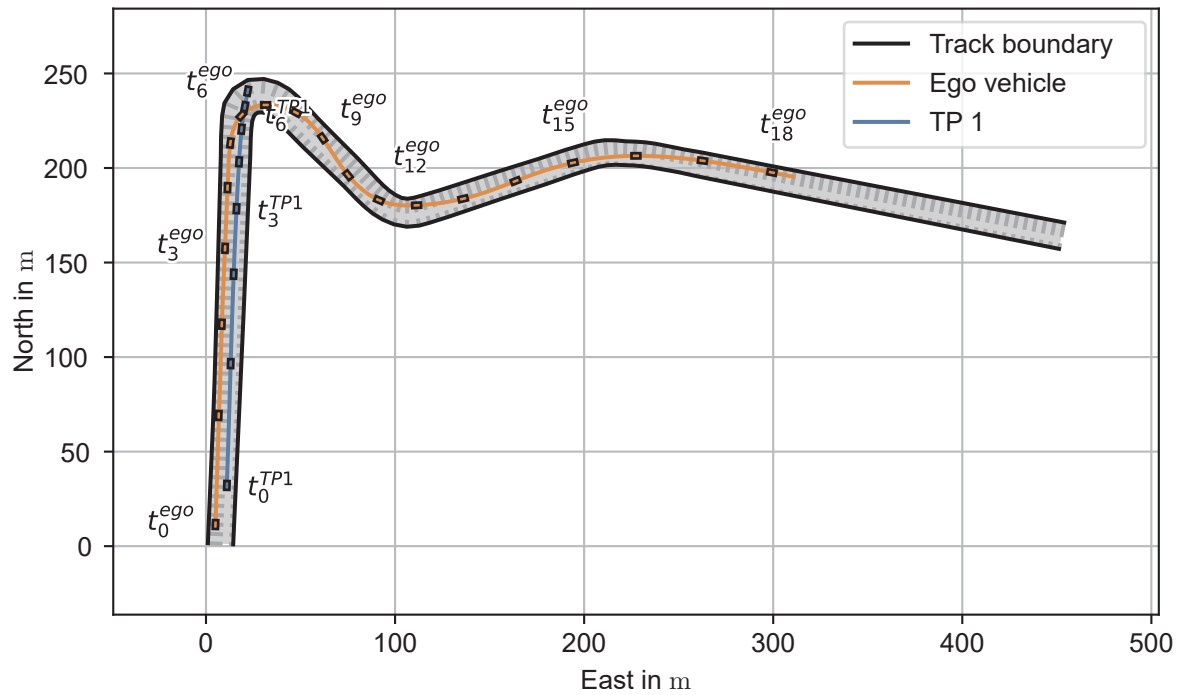


Figure A.34: Bird's eye view of scenario S30. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

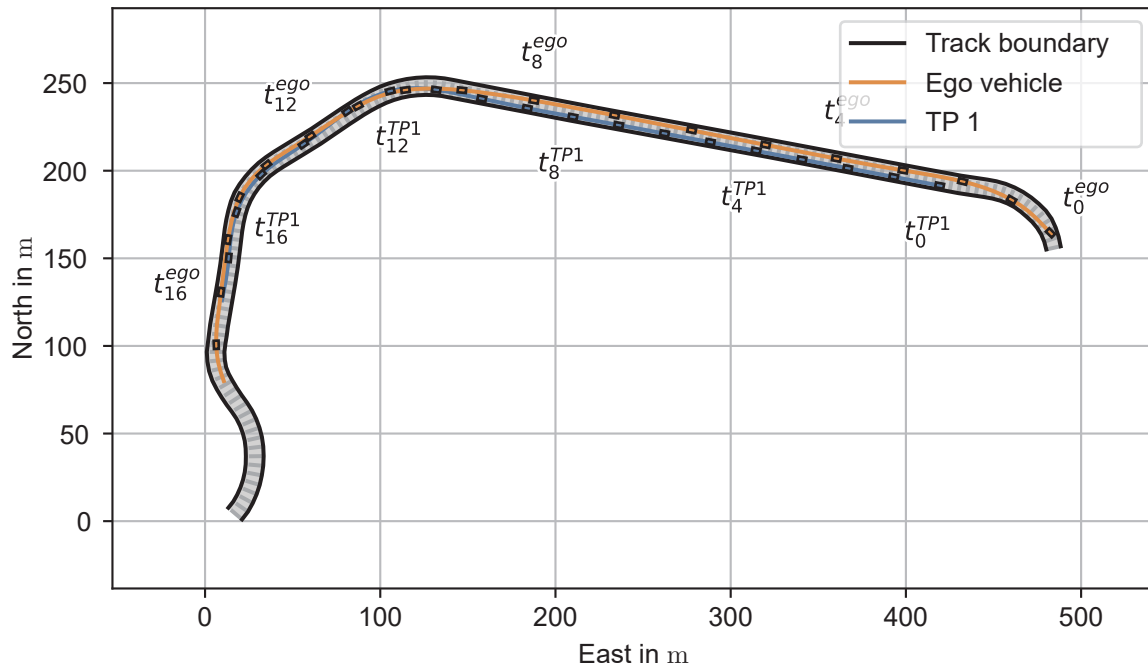


Figure A.35: Bird's eye view of scenario S31. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.

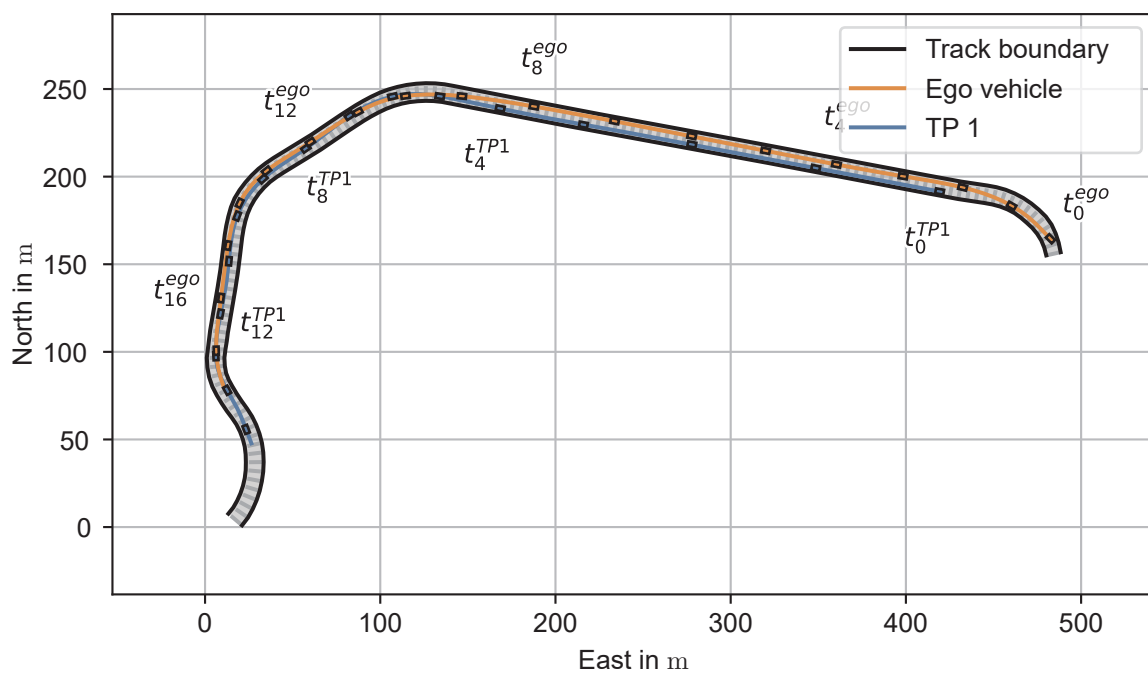


Figure A.36: Bird's eye view of scenario S32. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.

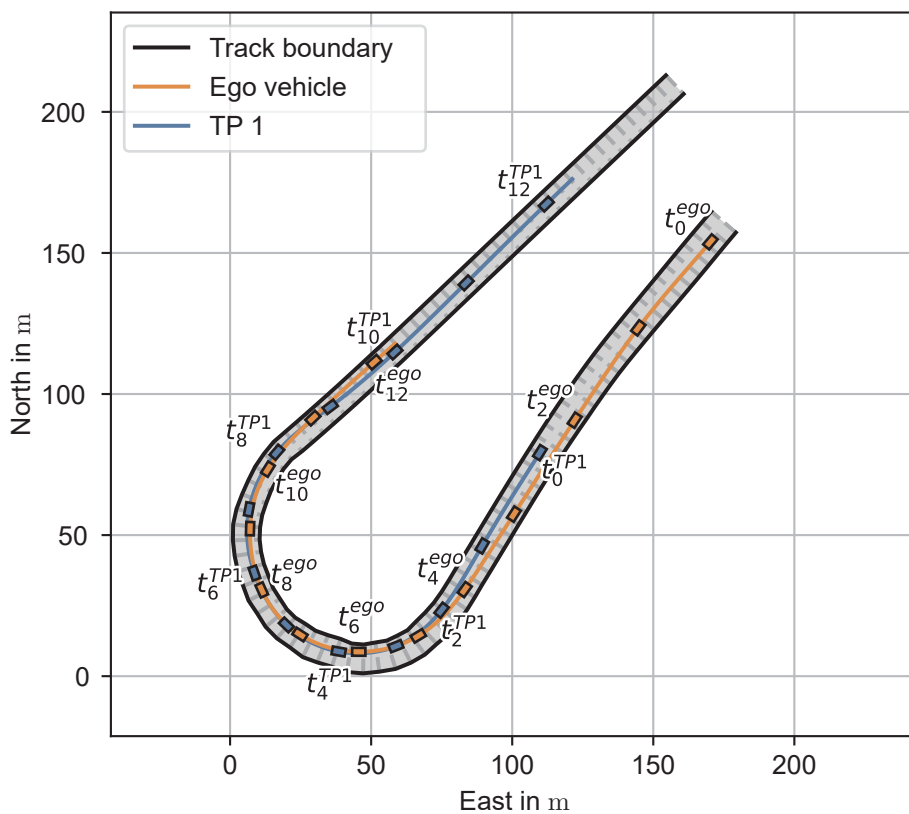


Figure A.37: Bird's eye view of scenario S33. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

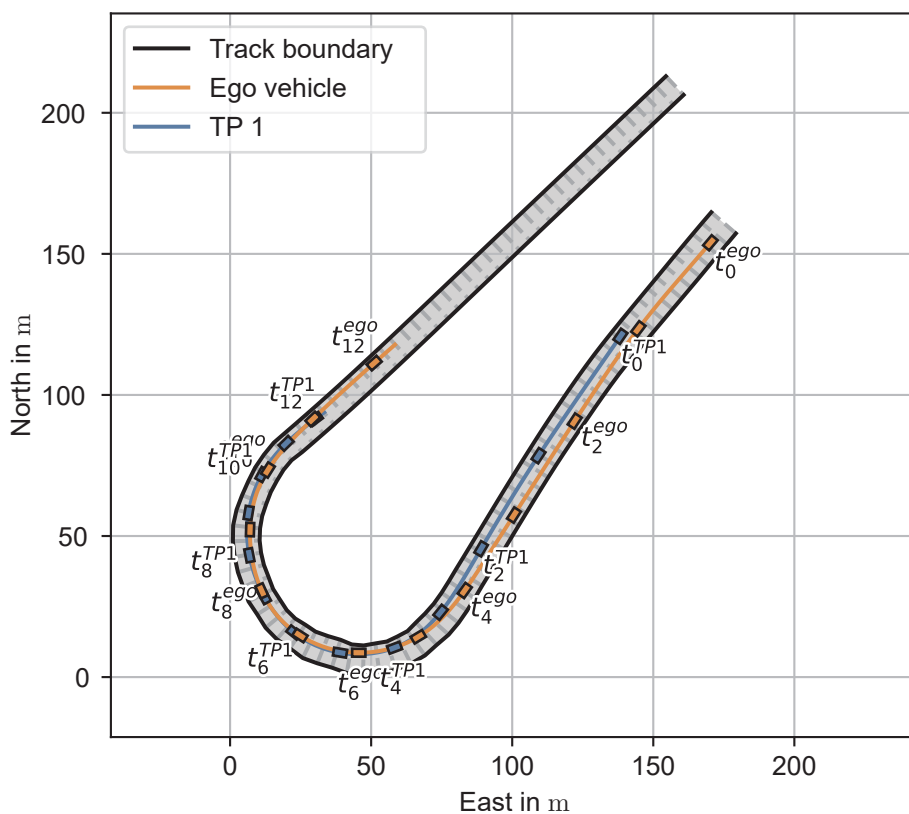


Figure A.38: Bird's eye view of scenario S34. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

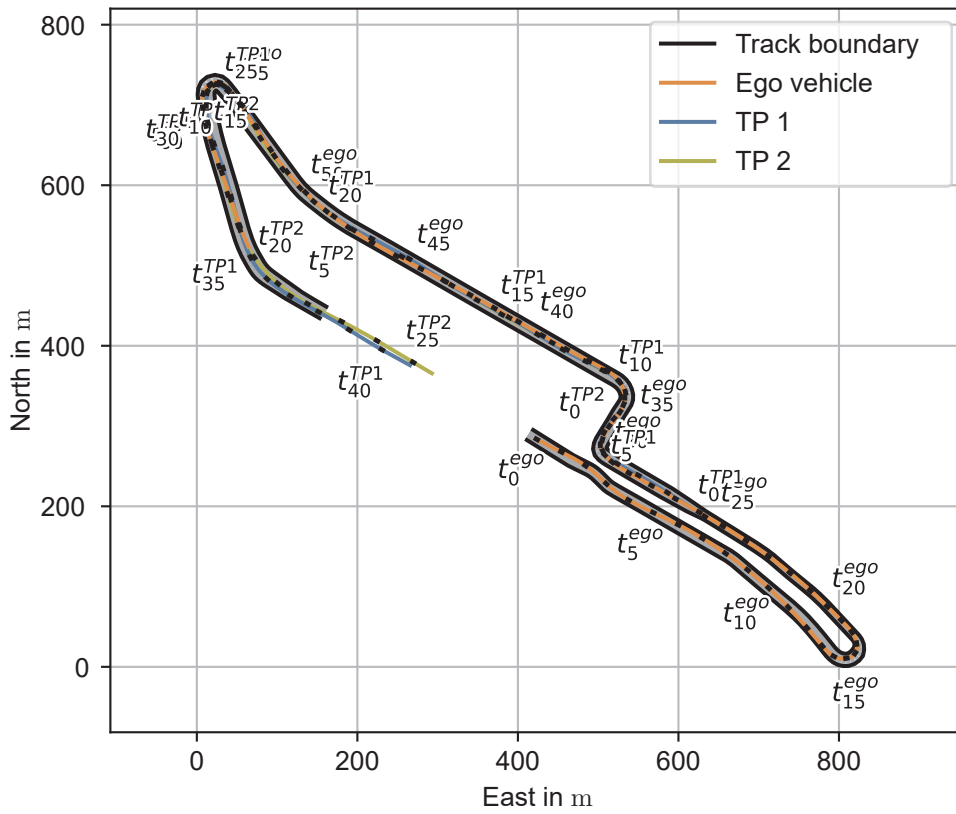


Figure A.39: Bird's eye view of scenario S35. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fifth interval is marked with a text label.

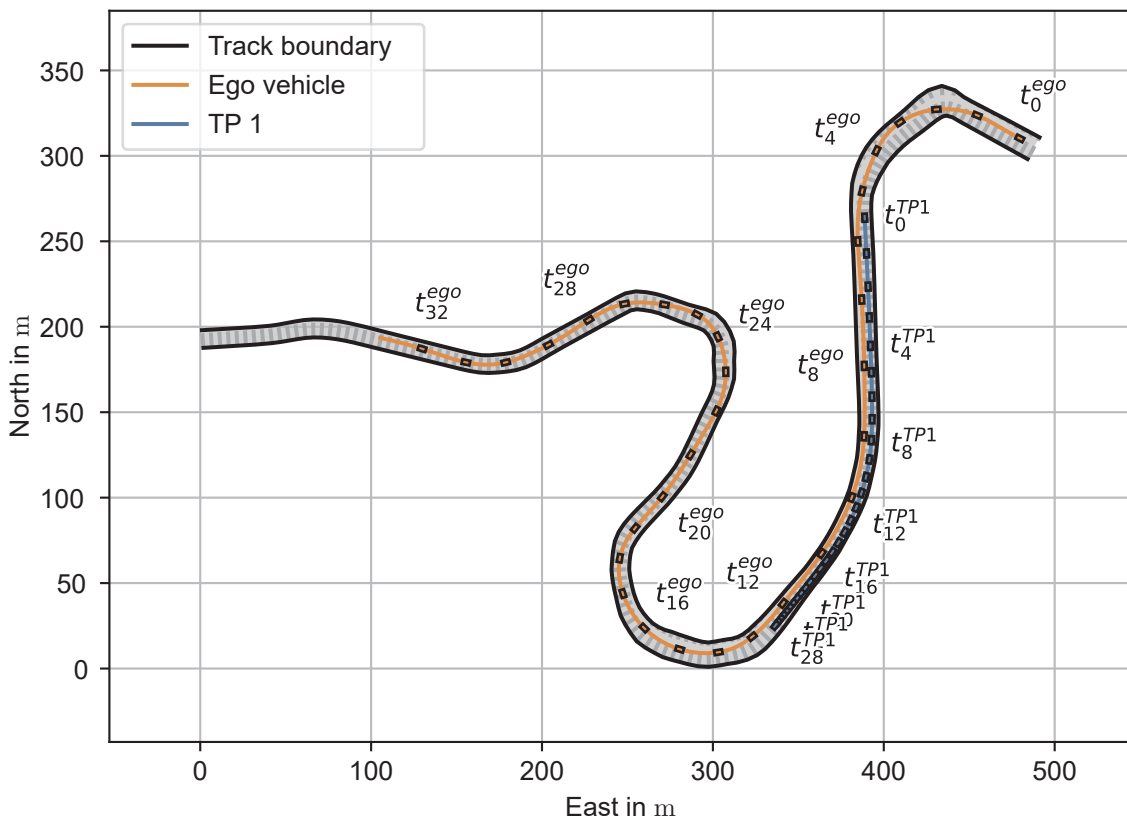


Figure A.40: Bird's eye view of scenario S36. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.

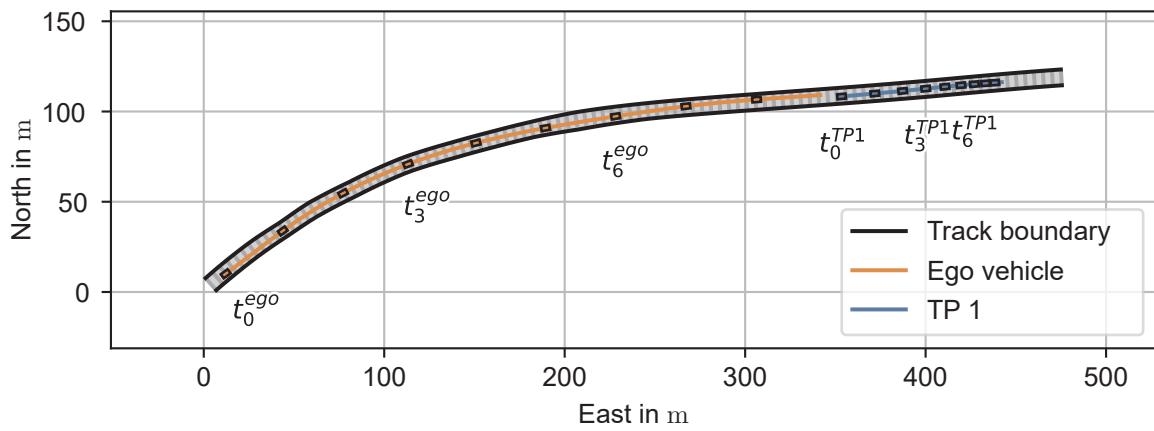


Figure A.41: Bird's eye view of scenario S37. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

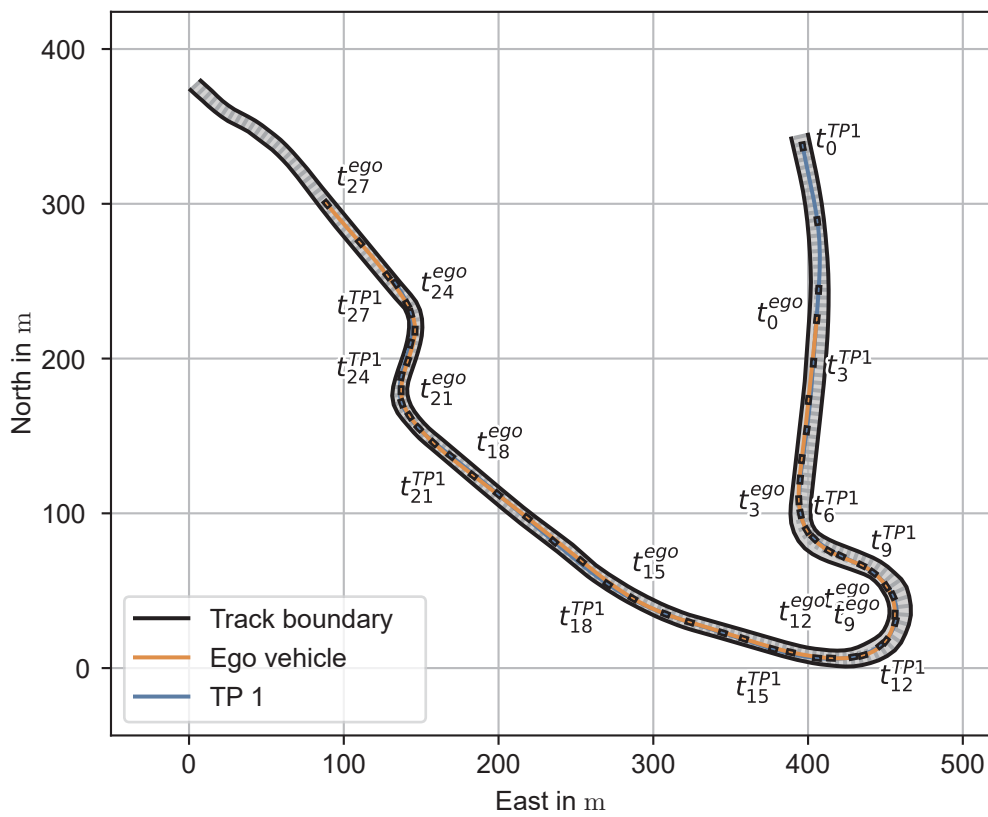


Figure A.42: Bird's eye view of scenario S38. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

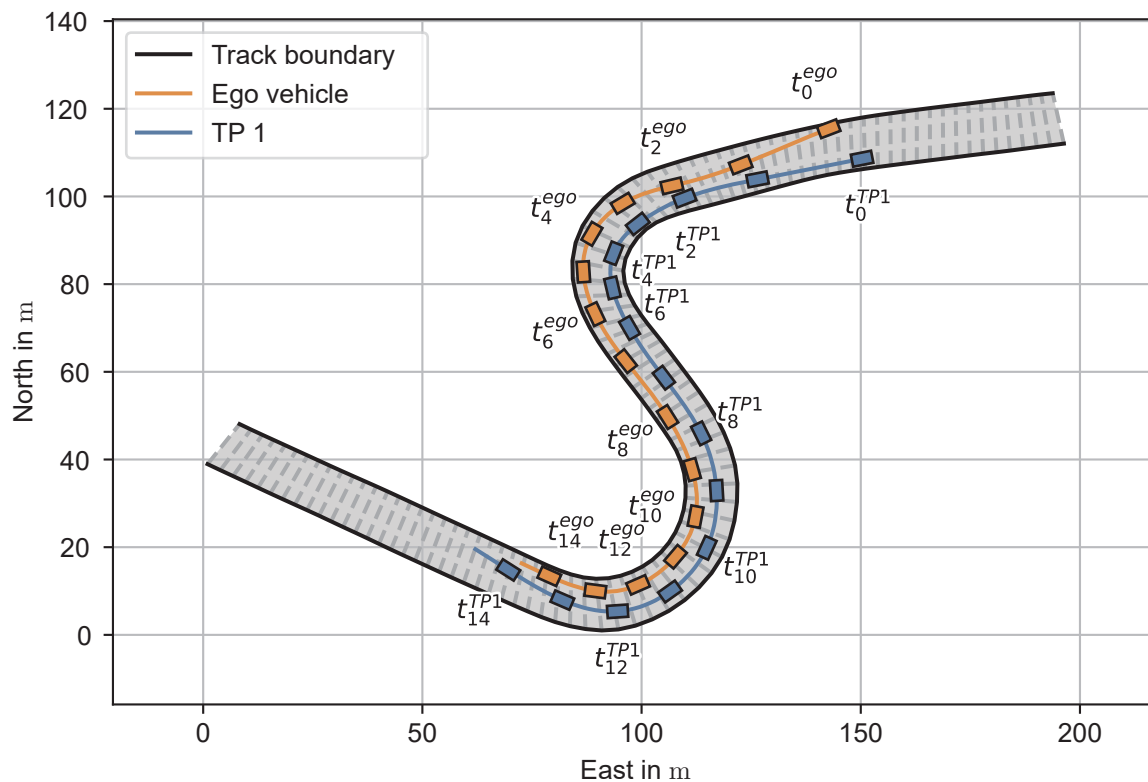


Figure A.43: Bird's eye view of scenario S39. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

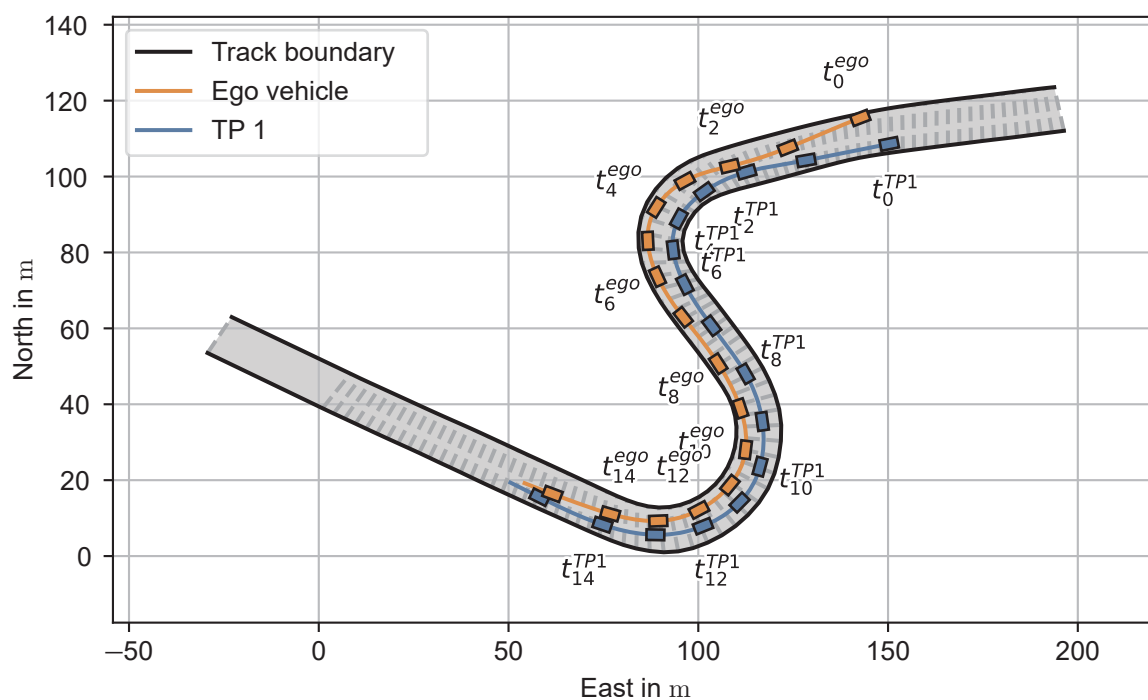


Figure A.44: Bird's eye view of scenario S40. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

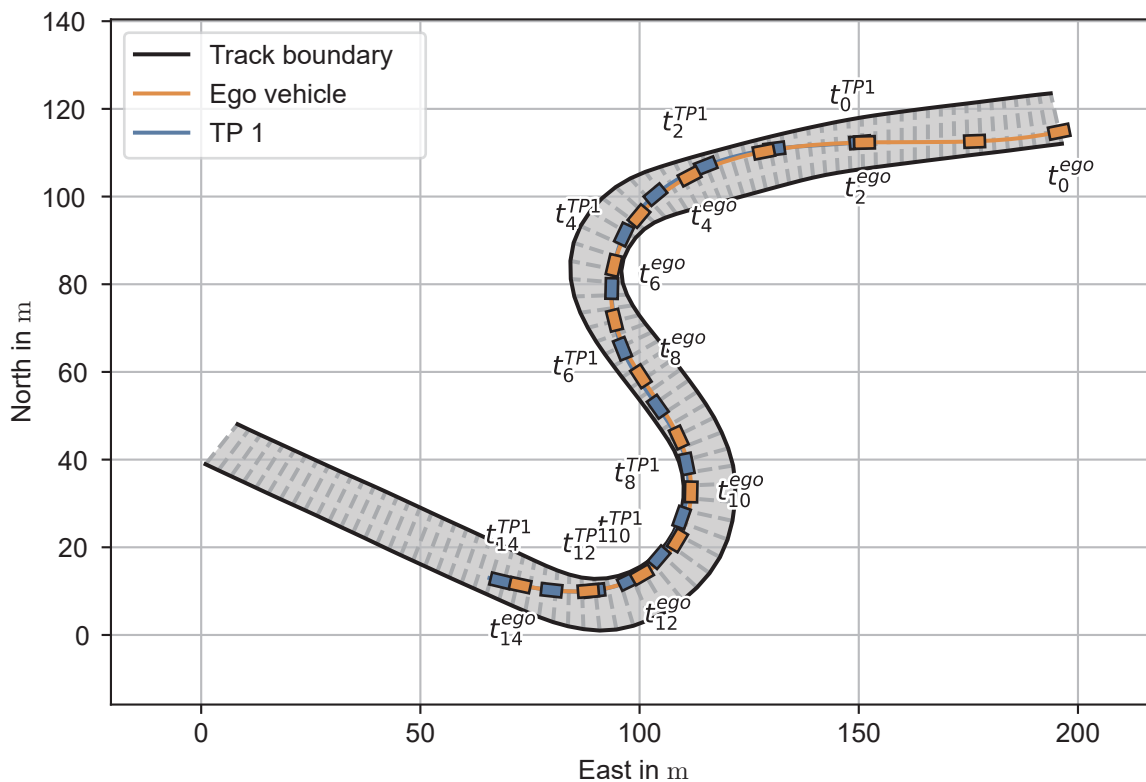


Figure A.45: Bird's eye view of scenario S41. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

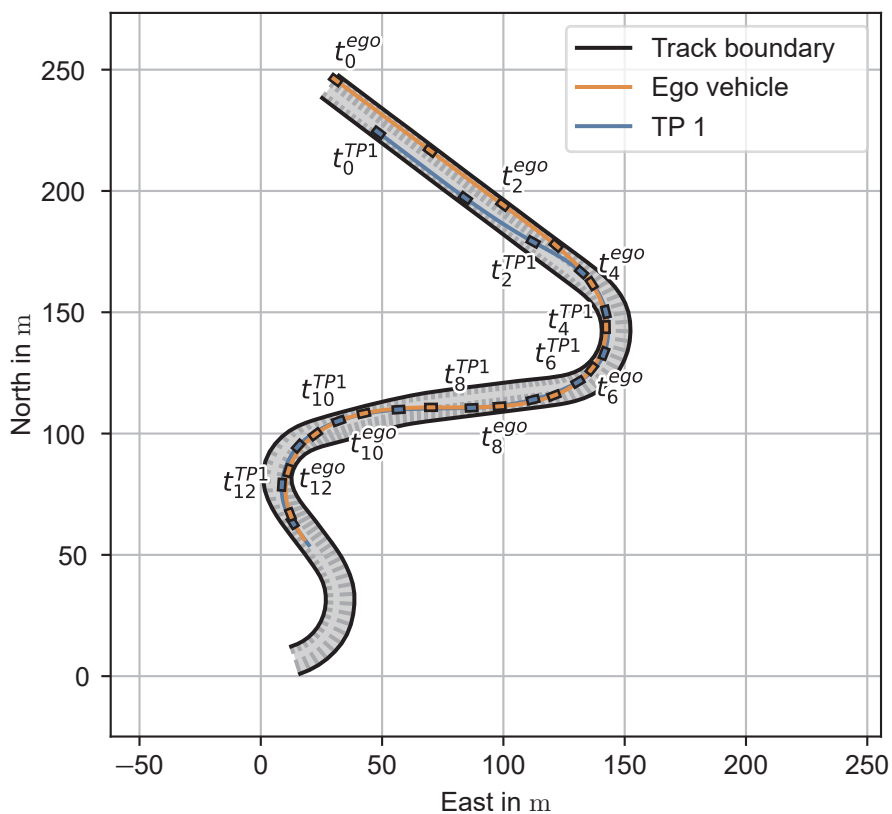


Figure A.46: Bird's eye view of scenario S42. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

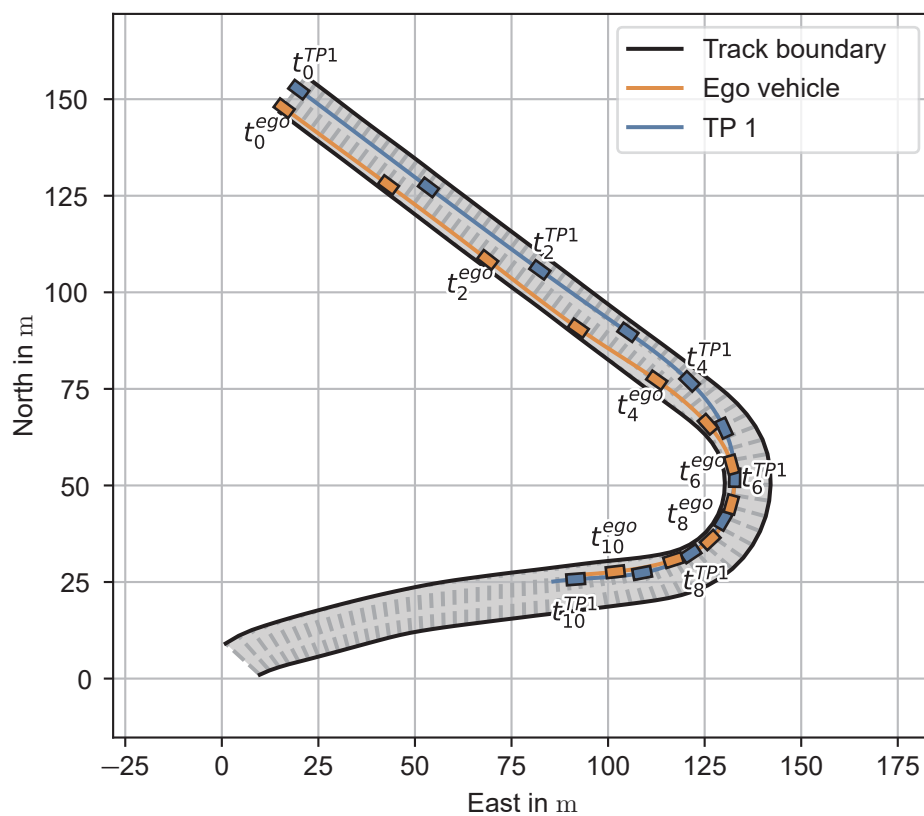


Figure A.47: Bird's eye view of scenario S43. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

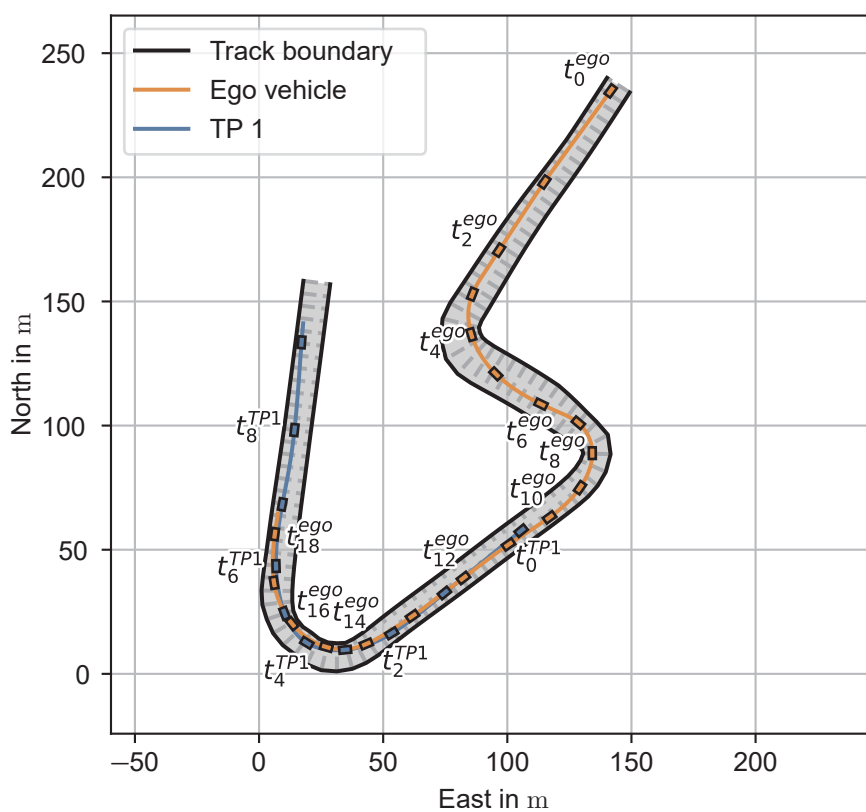


Figure A.48: Bird's eye view of scenario S44. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

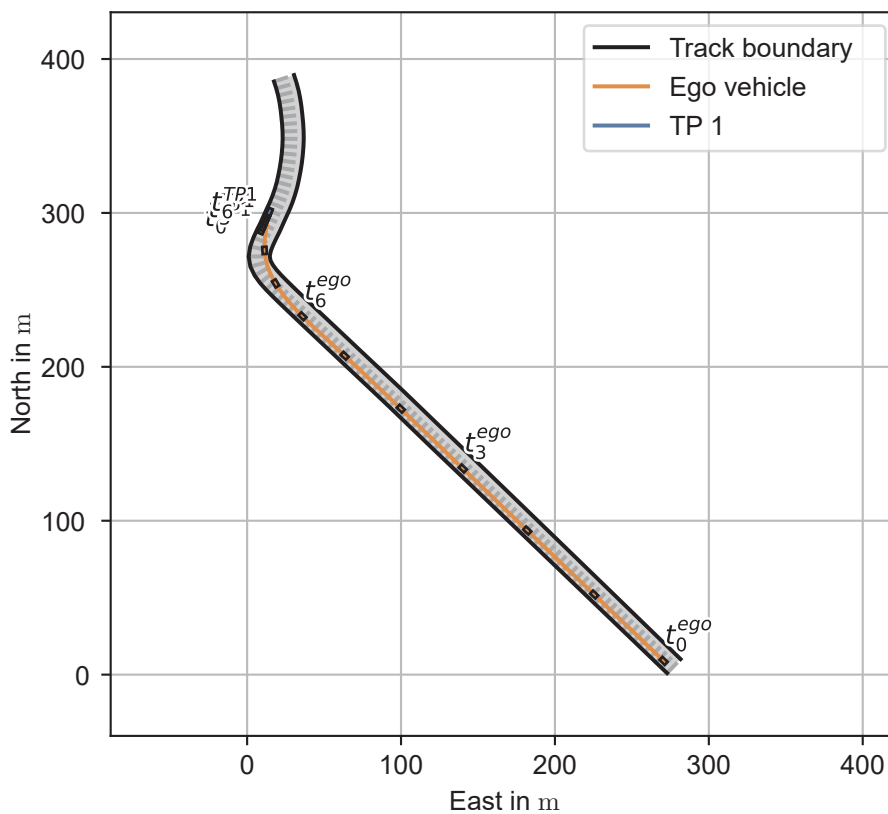


Figure A.49: Bird's eye view of scenario S45. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

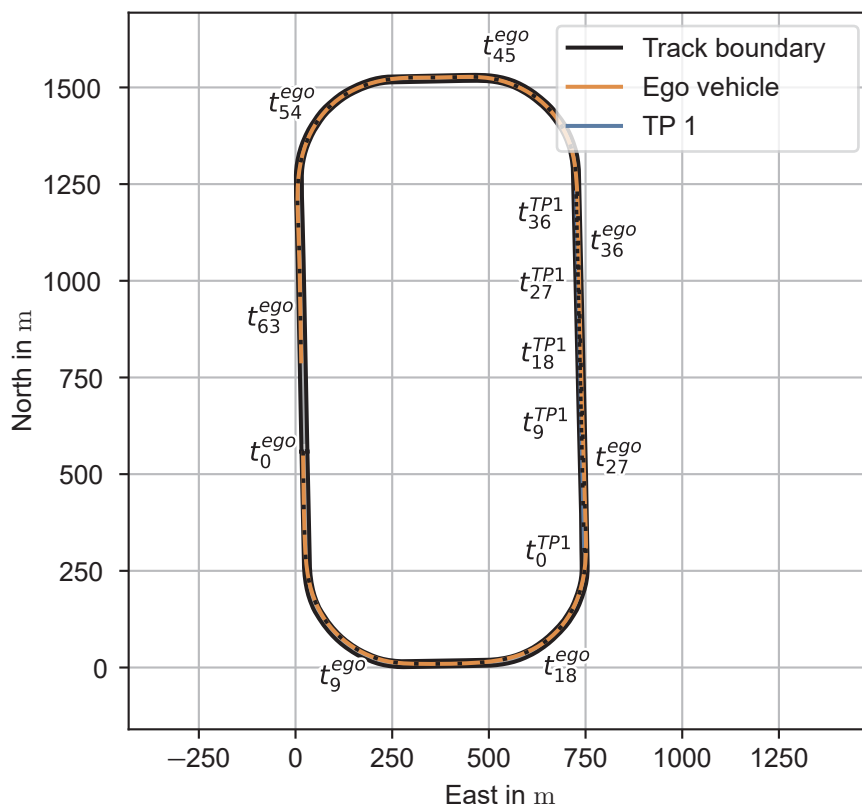


Figure A.50: Bird's eye view of scenario S46. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every ninth interval is marked with a text label.

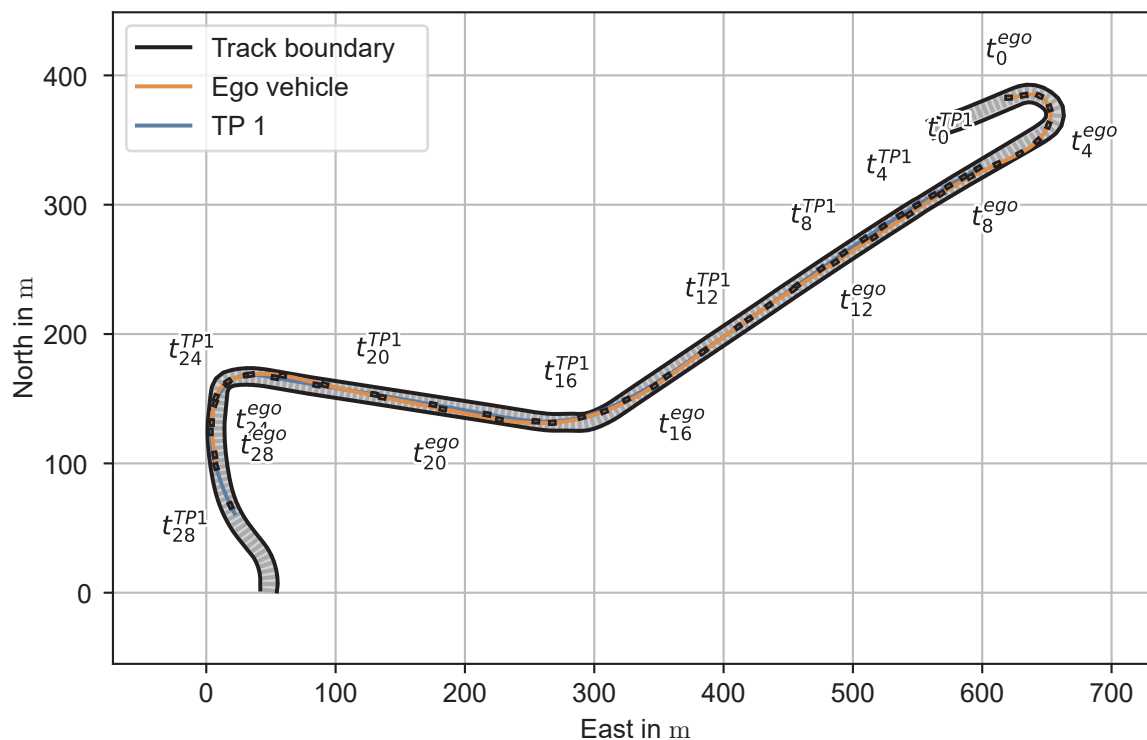


Figure A.51: Bird's eye view of scenario S47. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.

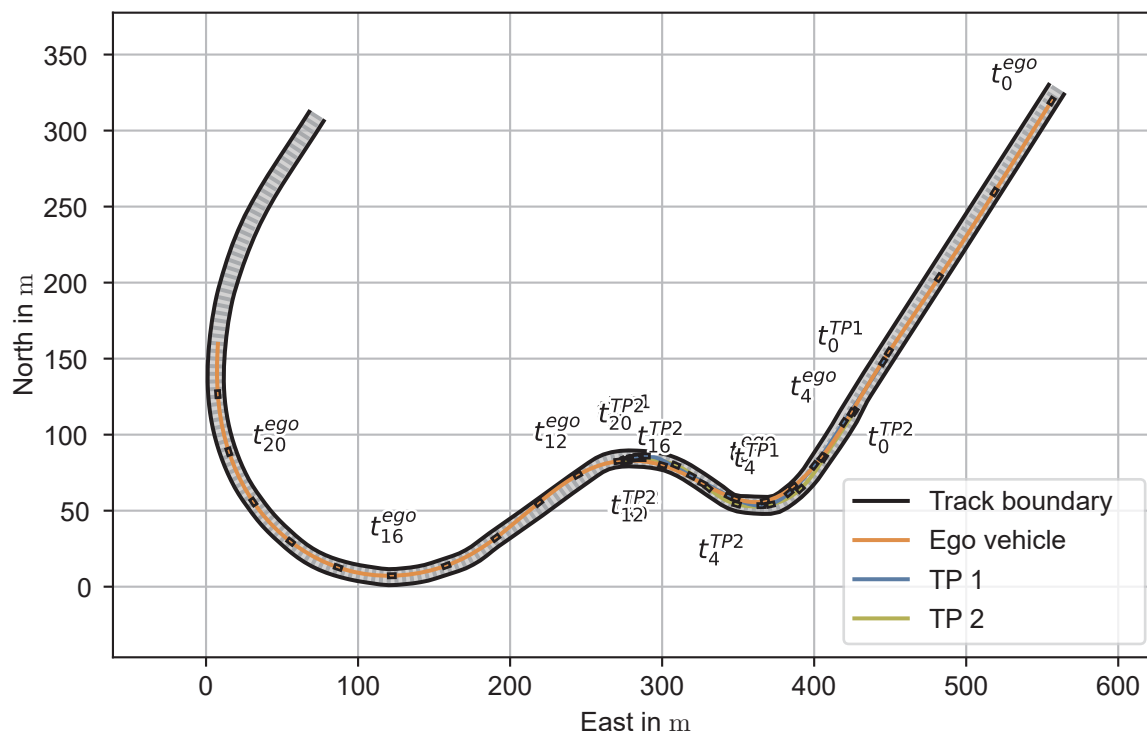


Figure A.52: Bird's eye view of scenario S48. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.

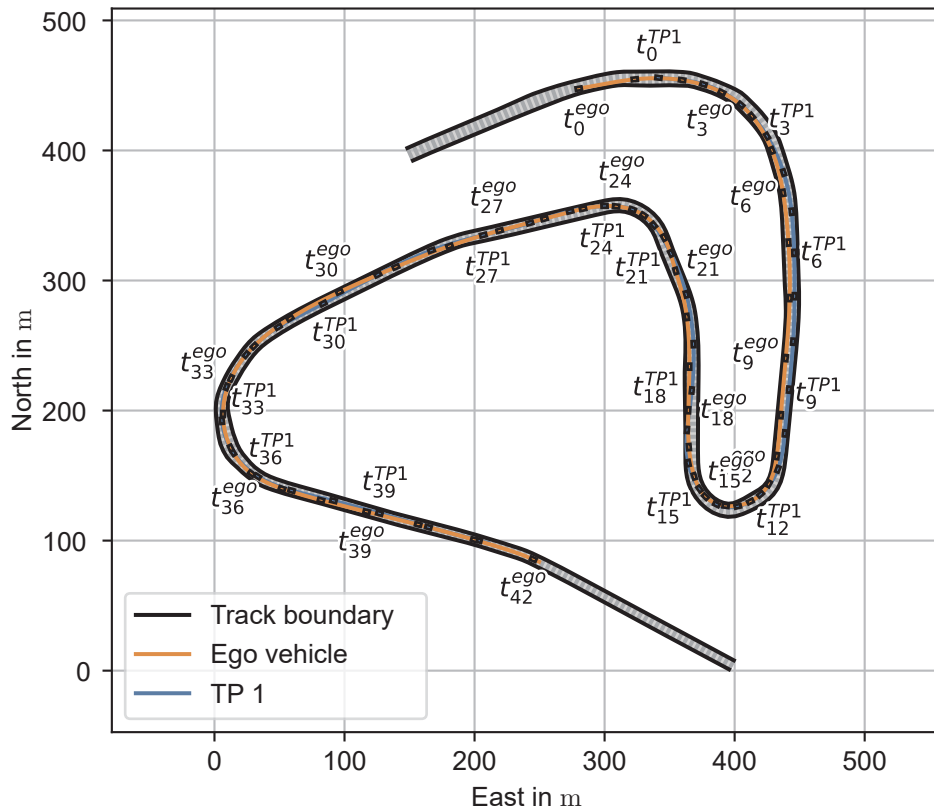


Figure A.53: Bird's eye view of scenario S49. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

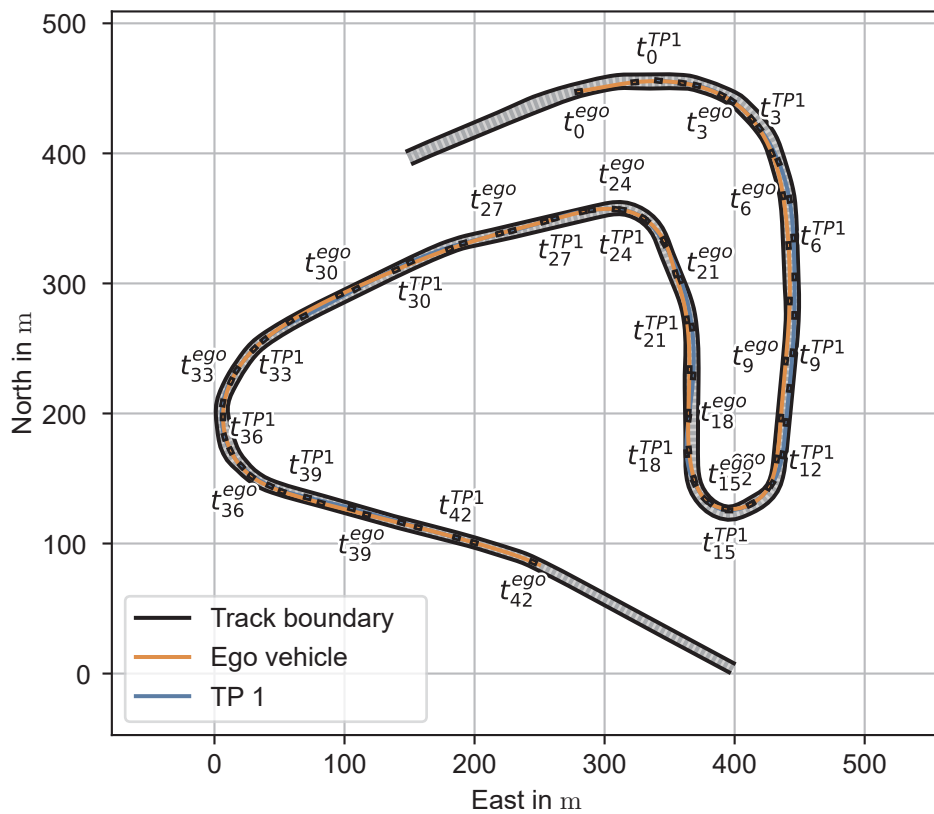


Figure A.54: Bird's eye view of scenario S50. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

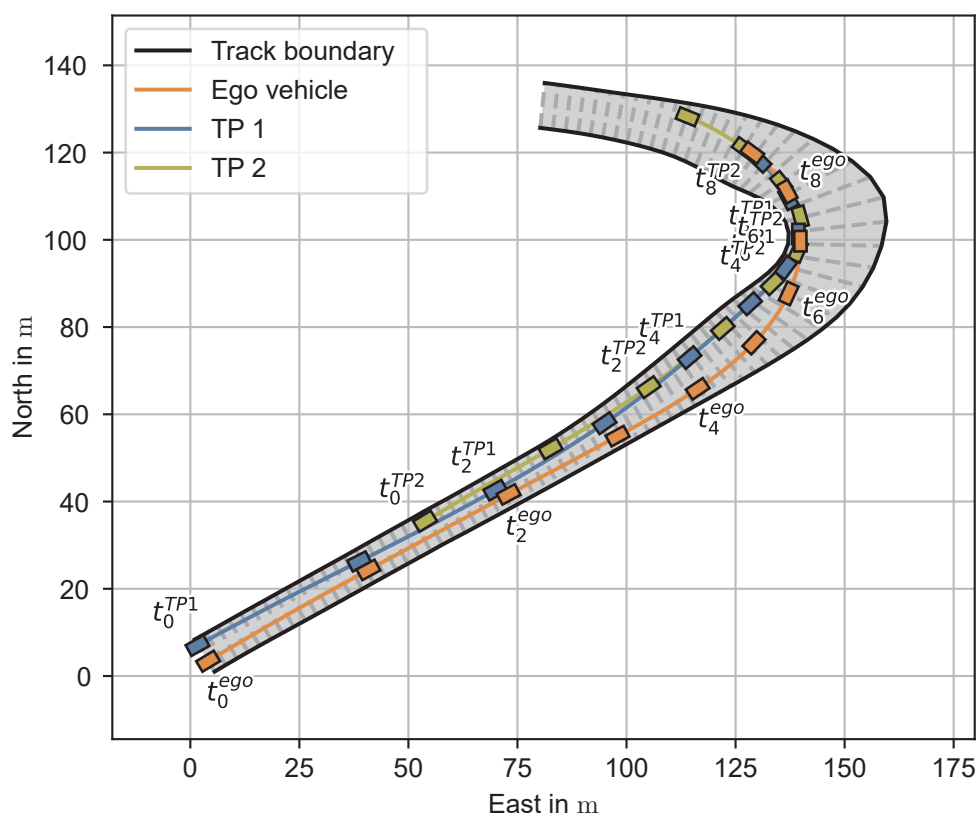


Figure A.55: Bird's eye view of scenario S51. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

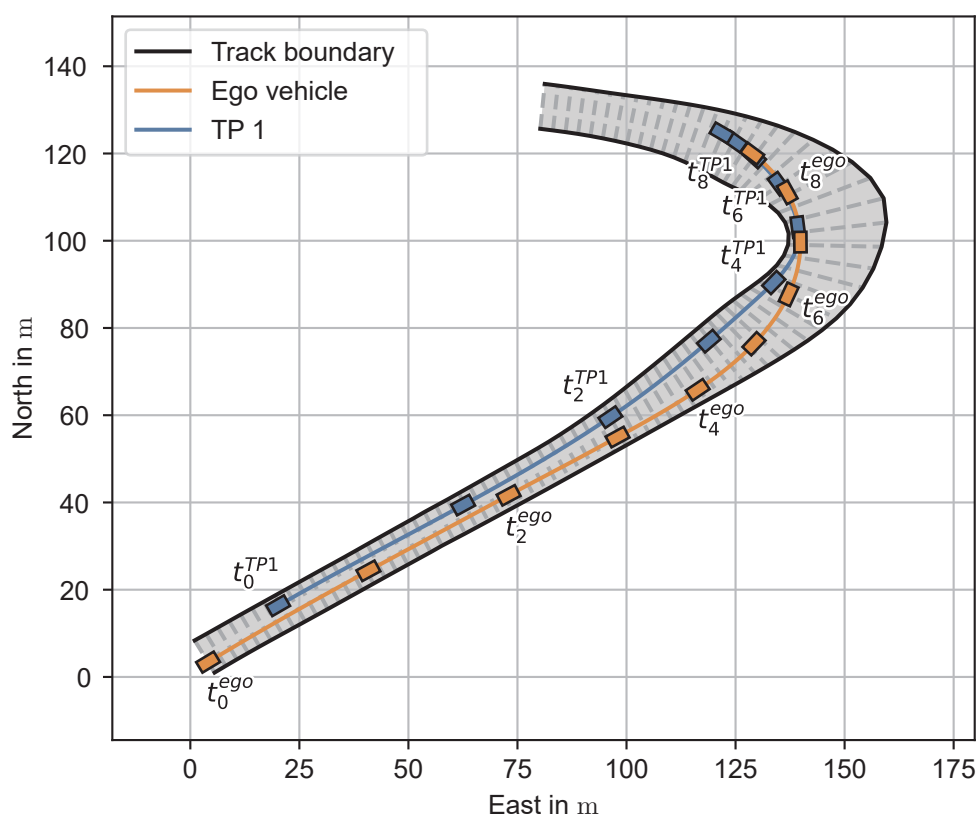


Figure A.56: Bird's eye view of scenario S52. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

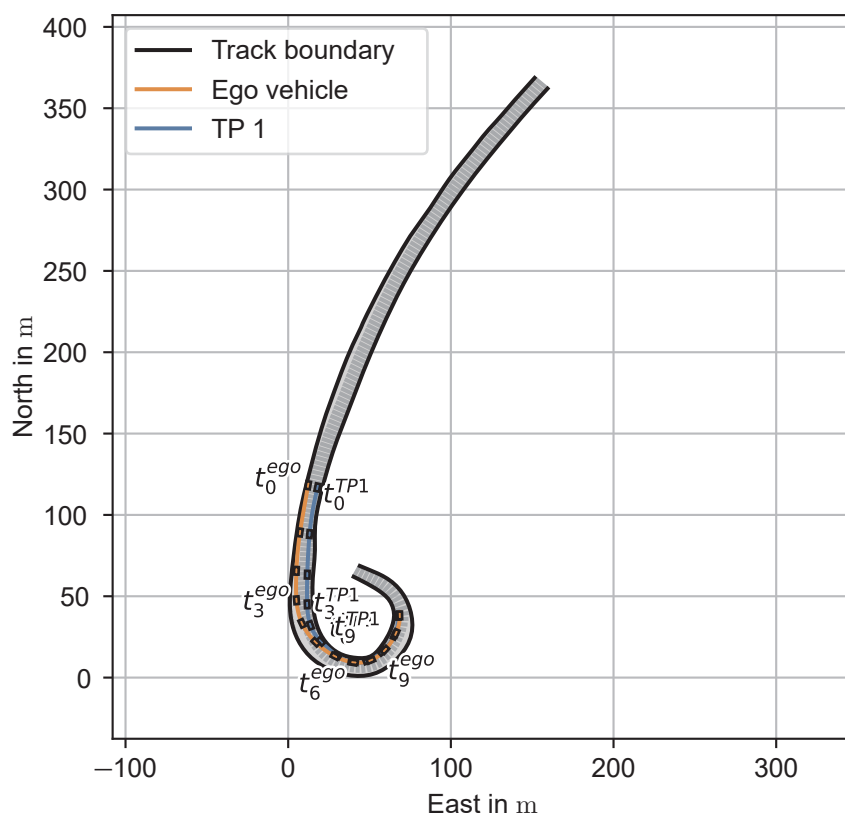


Figure A.57: Bird's eye view of scenario S53. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

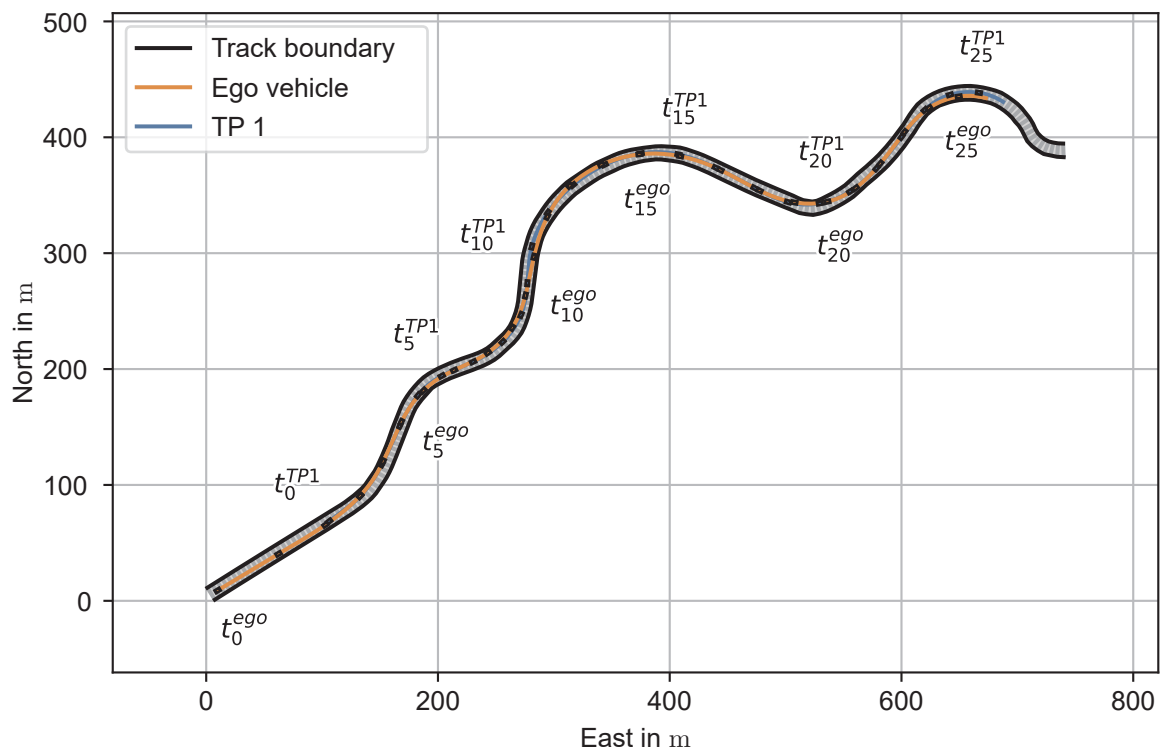


Figure A.58: Bird's eye view of scenario S54. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fifth interval is marked with a text label.

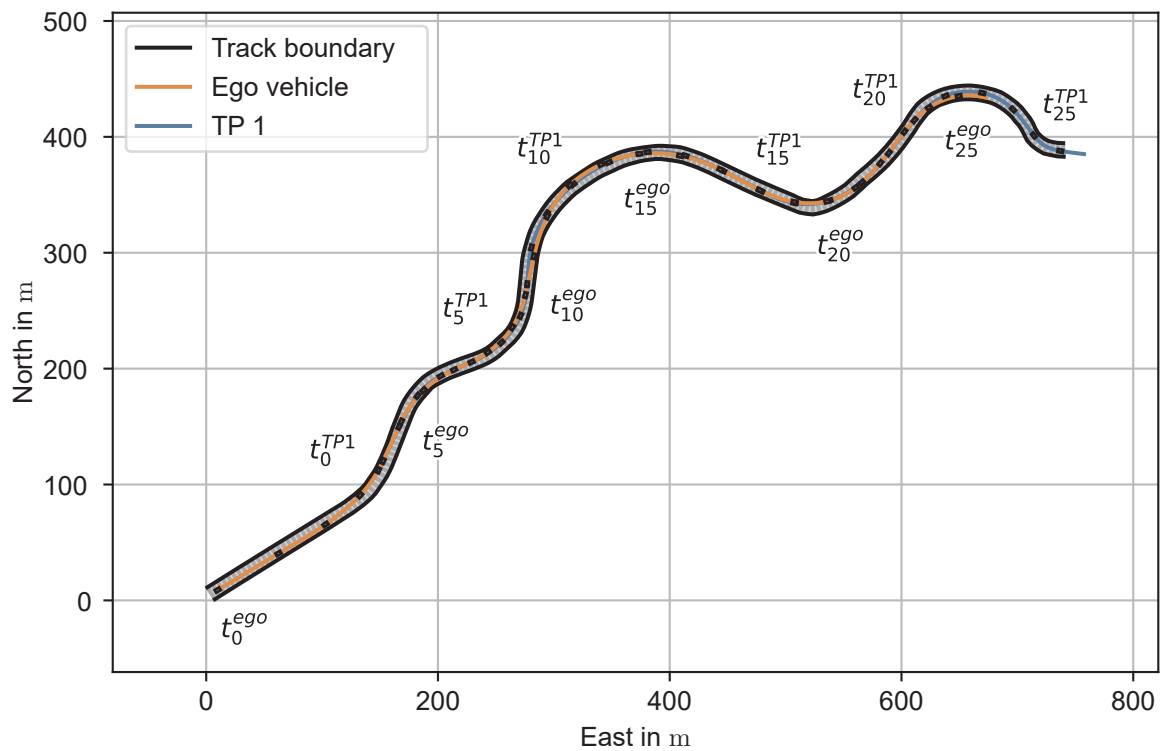


Figure A.59: Bird's eye view of scenario S55. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fifth interval is marked with a text label.

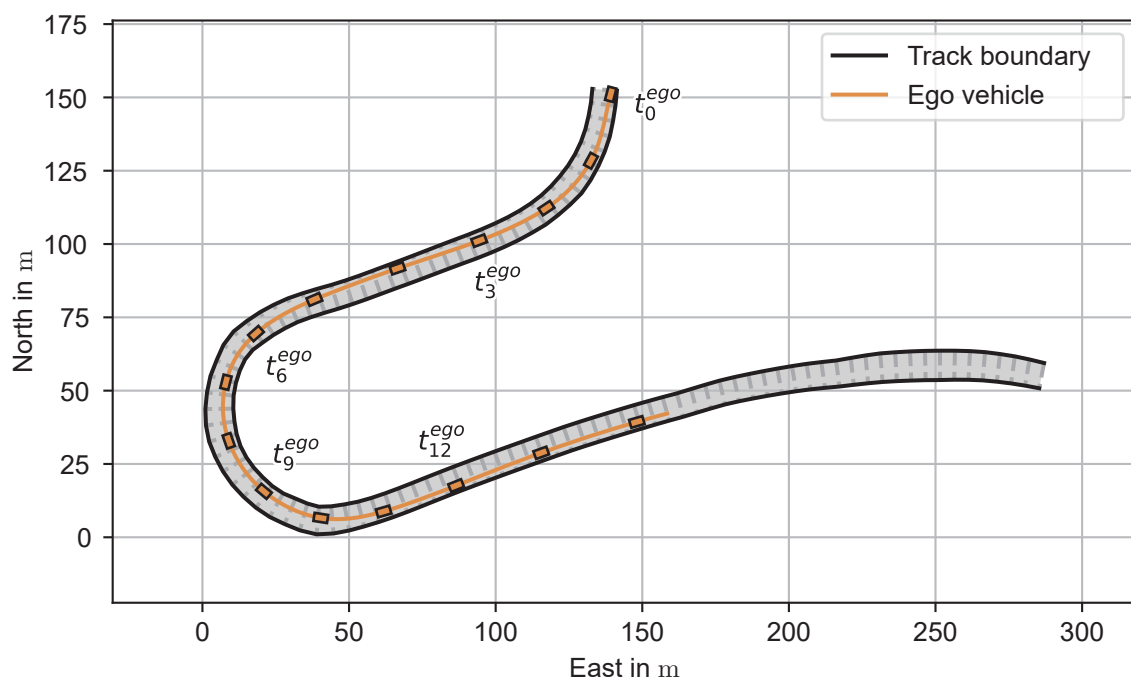


Figure A.60: Bird's eye view of scenario S56. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

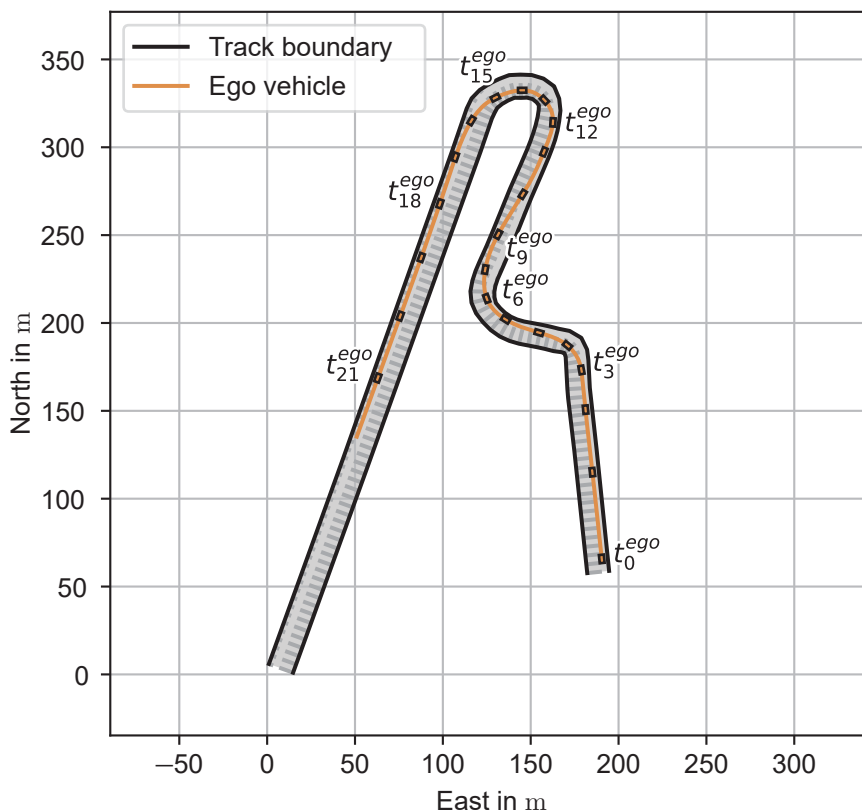


Figure A.61: Bird's eye view of scenario S57. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

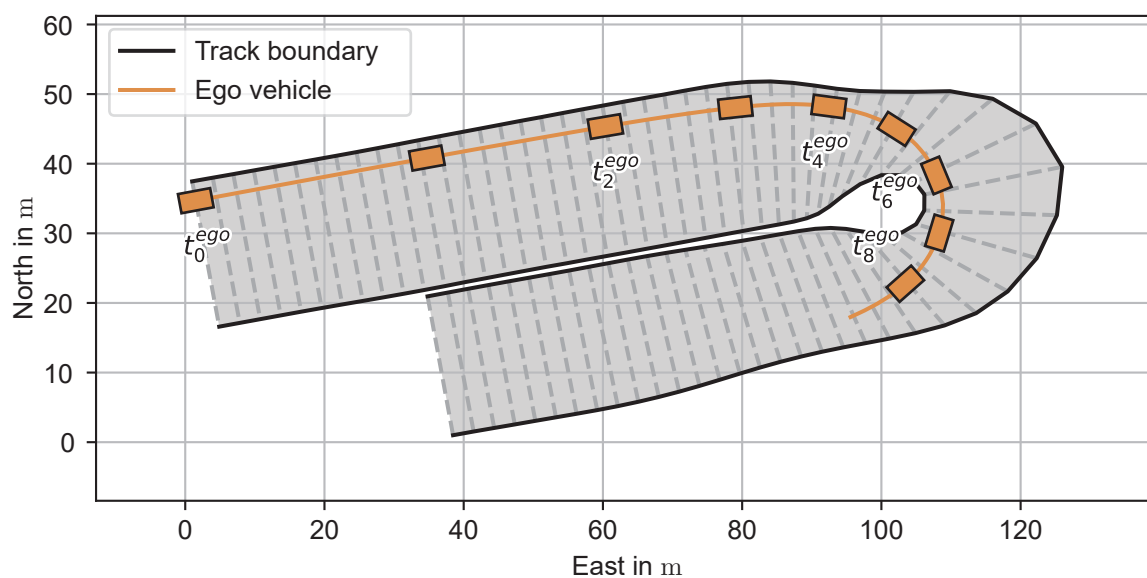


Figure A.62: Bird's eye view of scenario S58. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

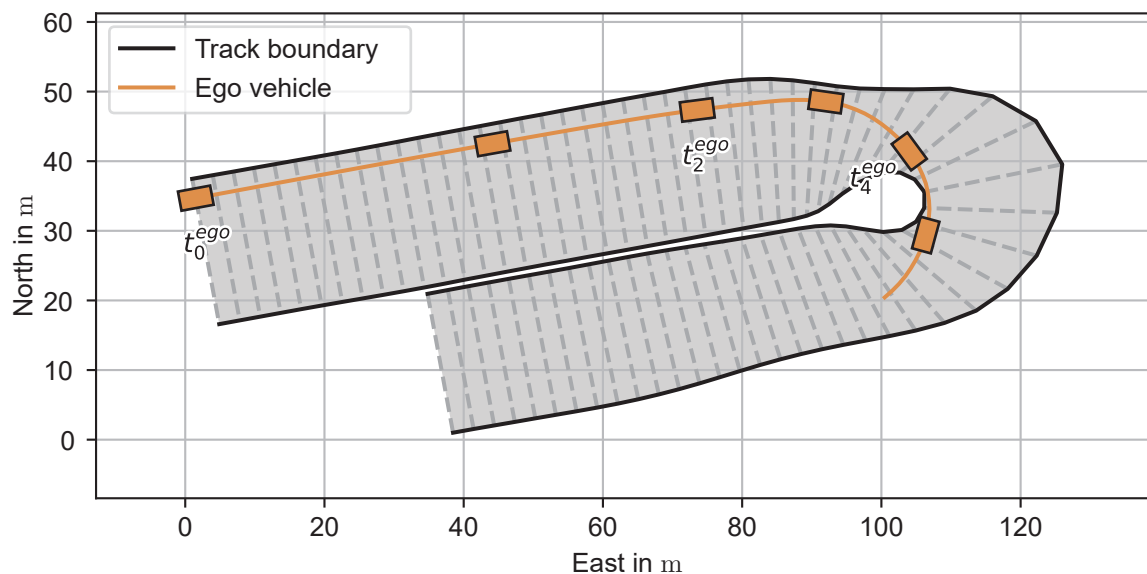


Figure A.63: Bird's eye view of scenario S59. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

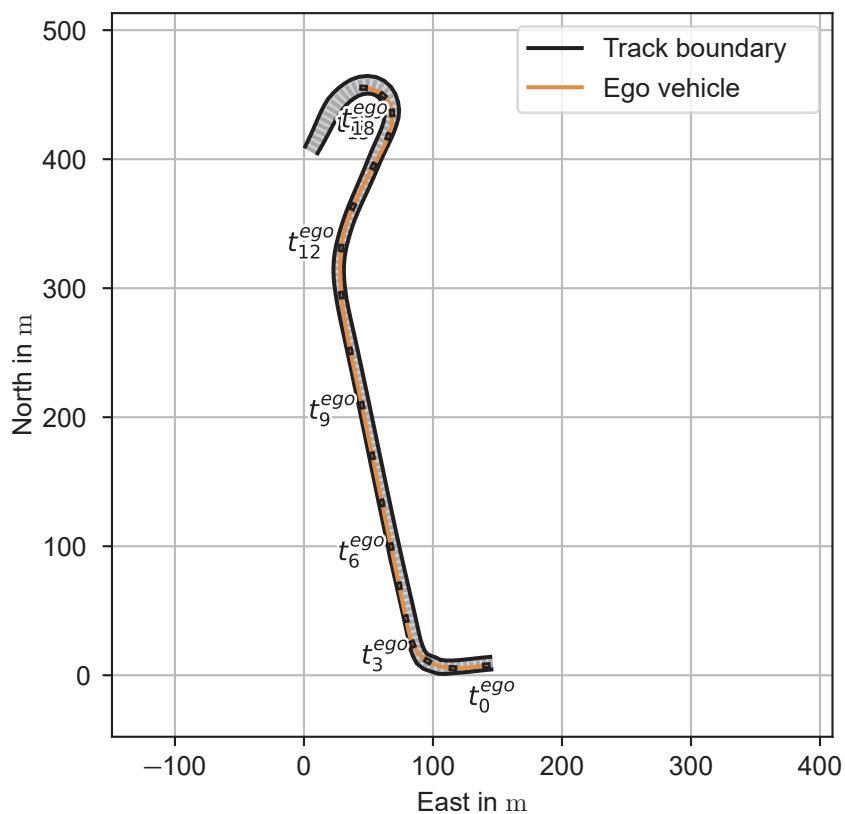


Figure A.64: Bird's eye view of scenario S60. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

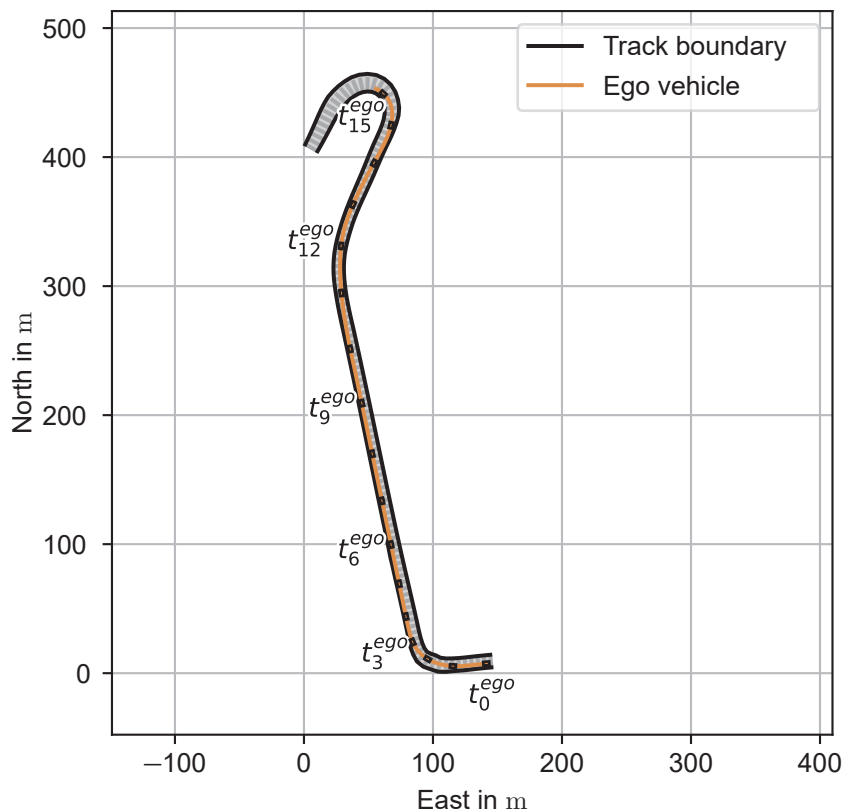


Figure A.65: Bird's eye view of scenario S61. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

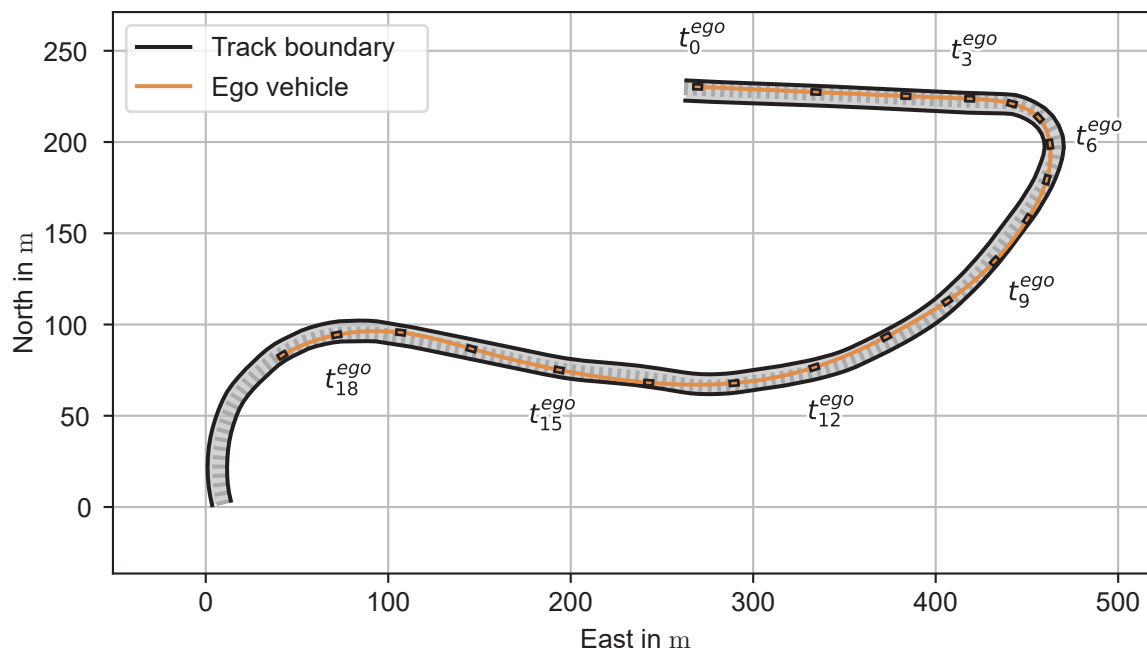


Figure A.66: Bird's eye view of scenario S62. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

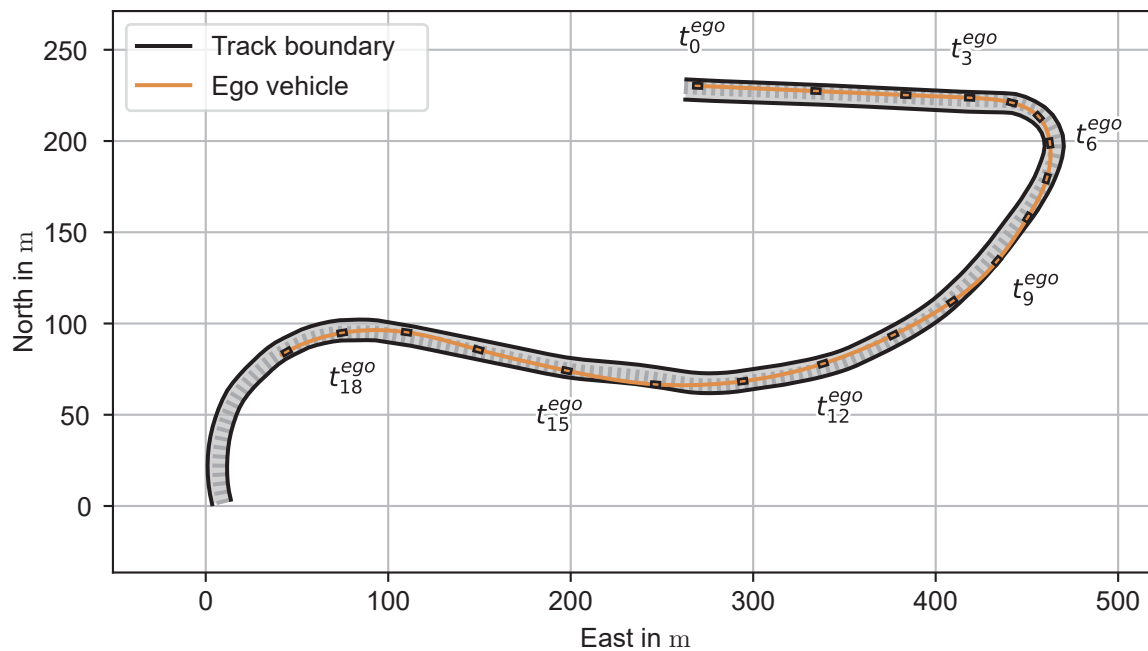


Figure A.67: Bird's eye view of scenario S63. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

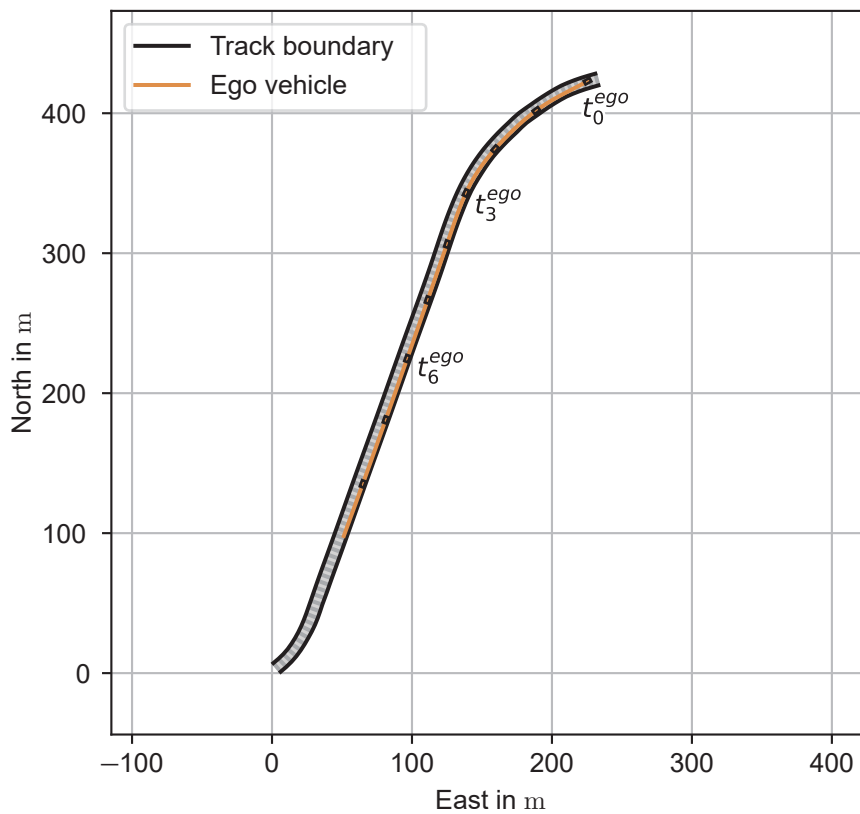


Figure A.68: Bird's eye view of scenario S64. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

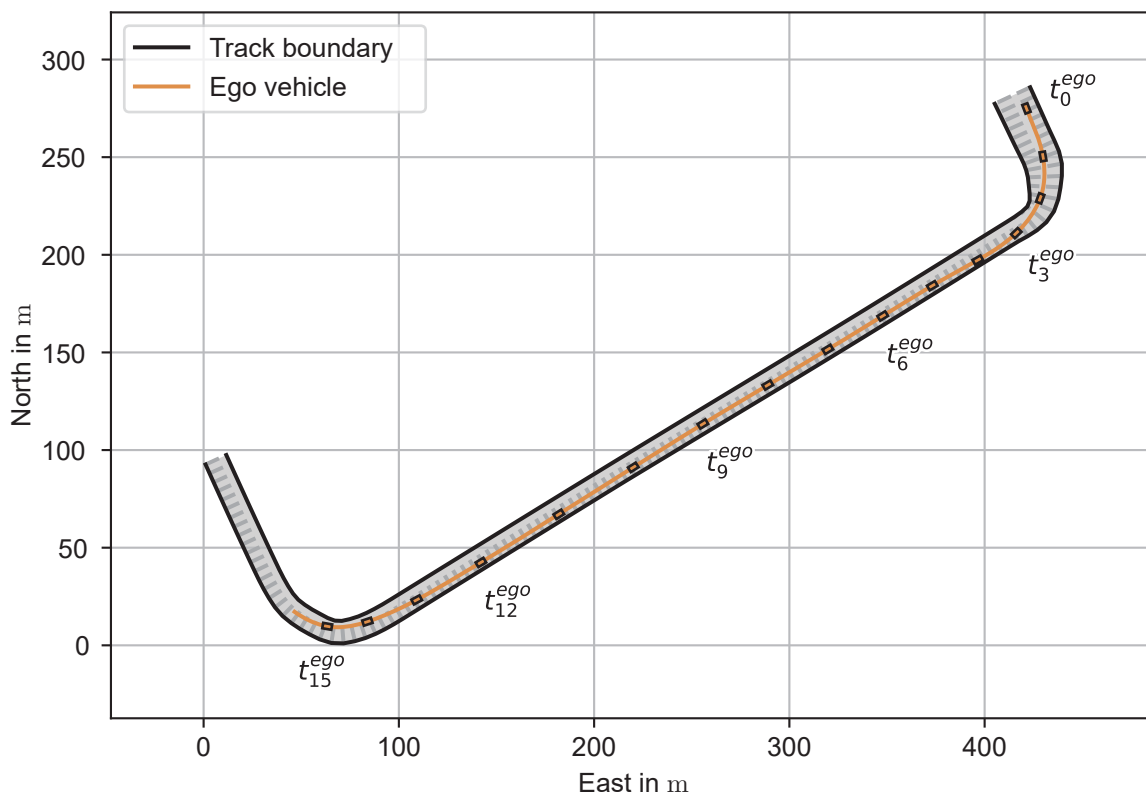


Figure A.69: Bird's eye view of scenario S65. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

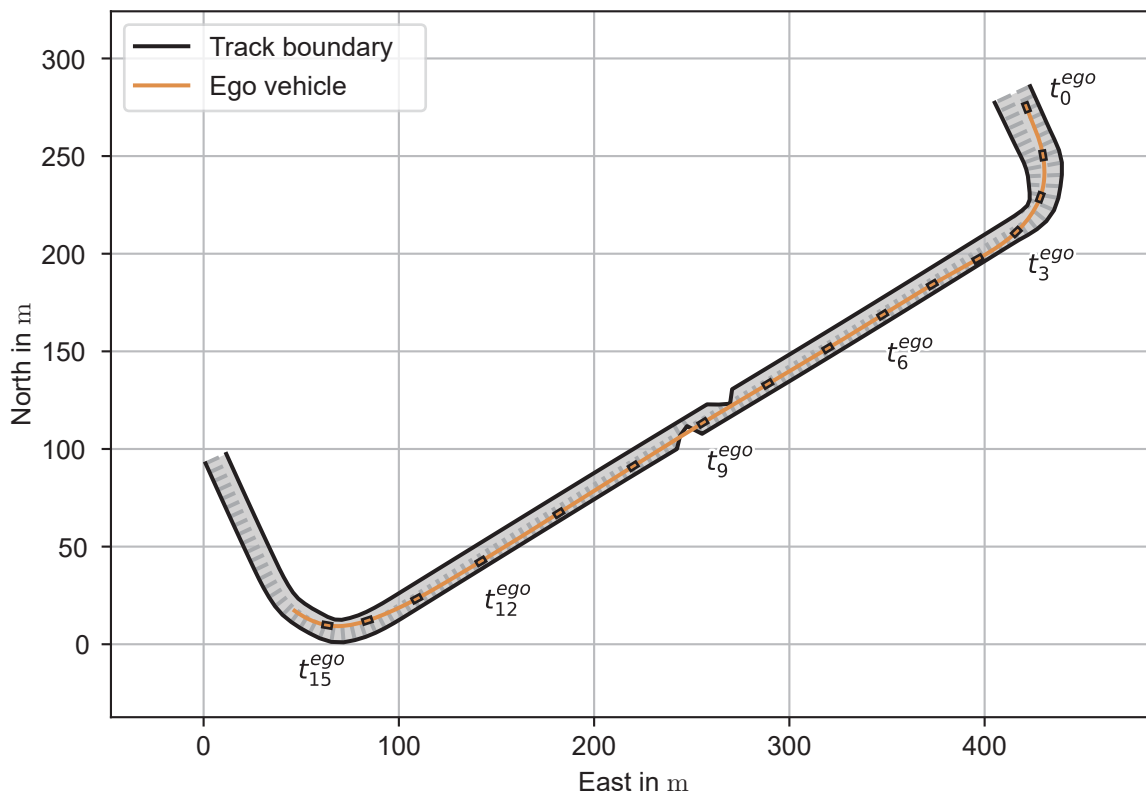


Figure A.70: Bird's eye view of scenario S66. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

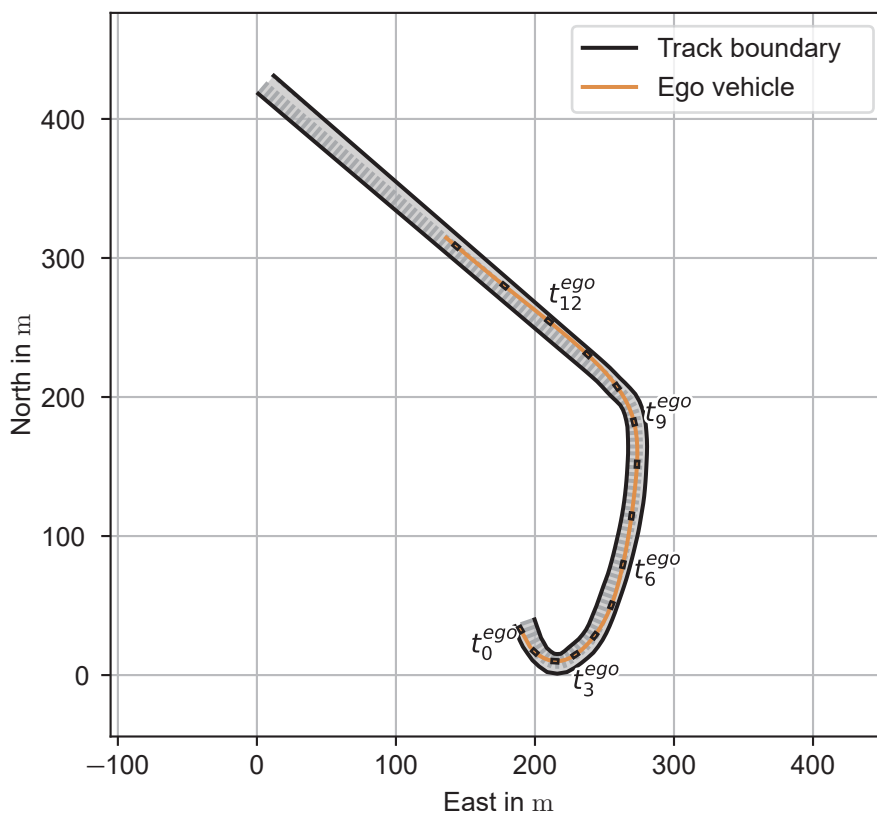


Figure A.71: Bird's eye view of scenario S67. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

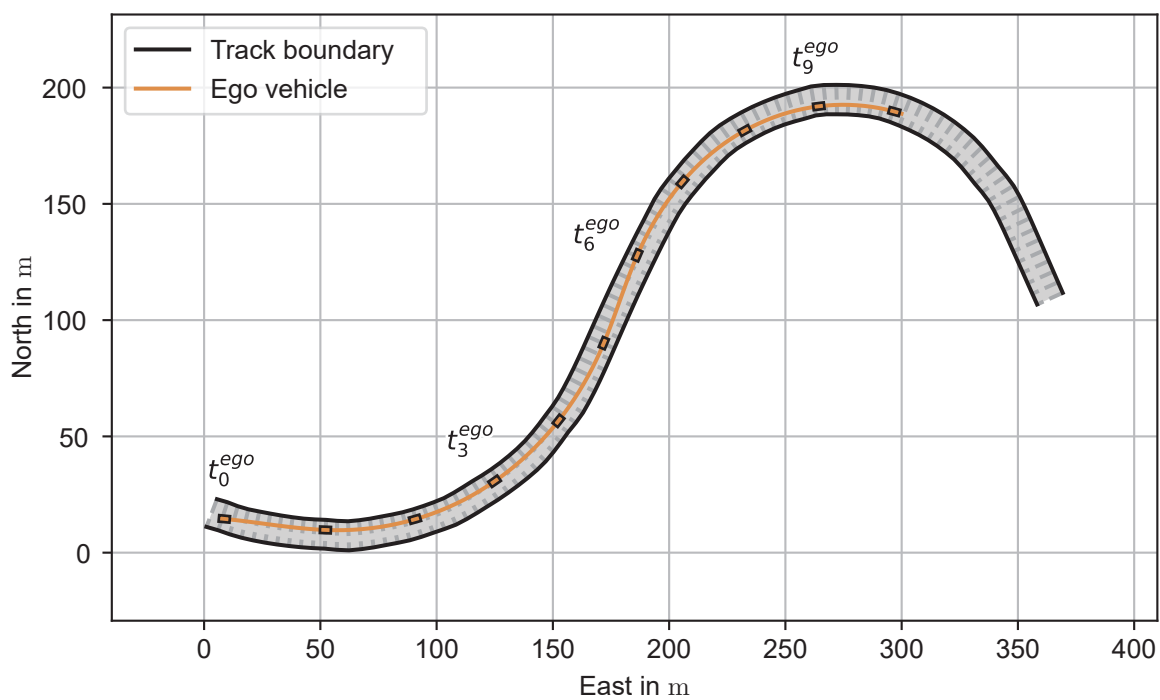


Figure A.72: Bird's eye view of scenario S68. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

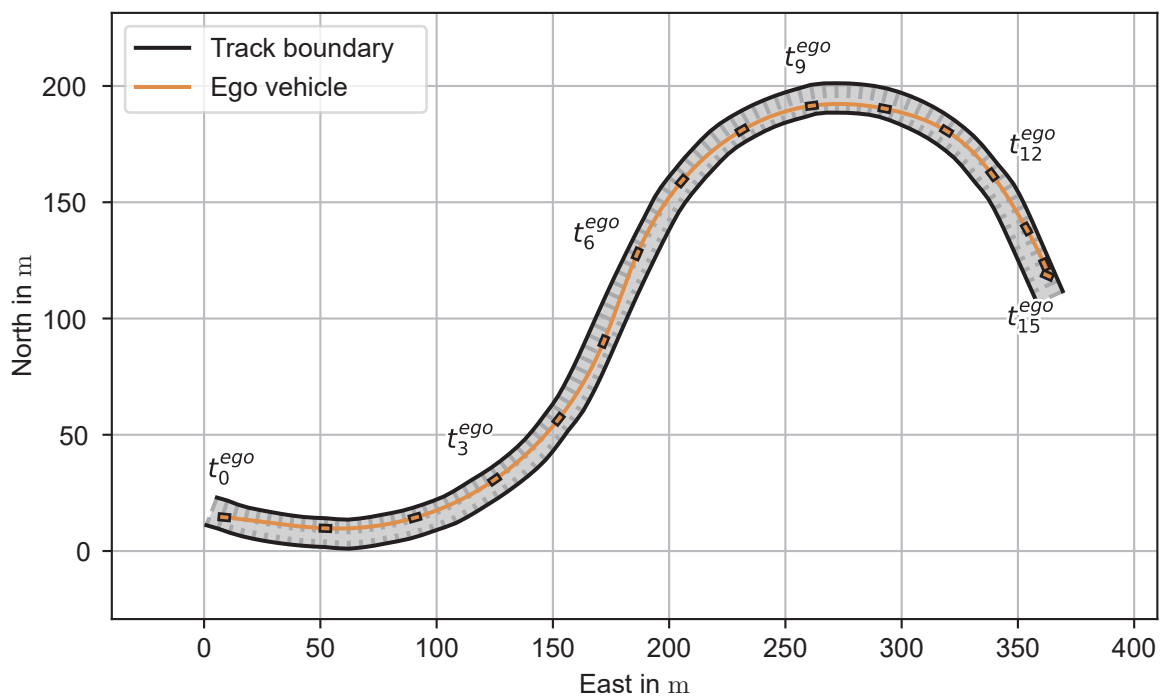


Figure A.73: Bird's eye view of scenario S69. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

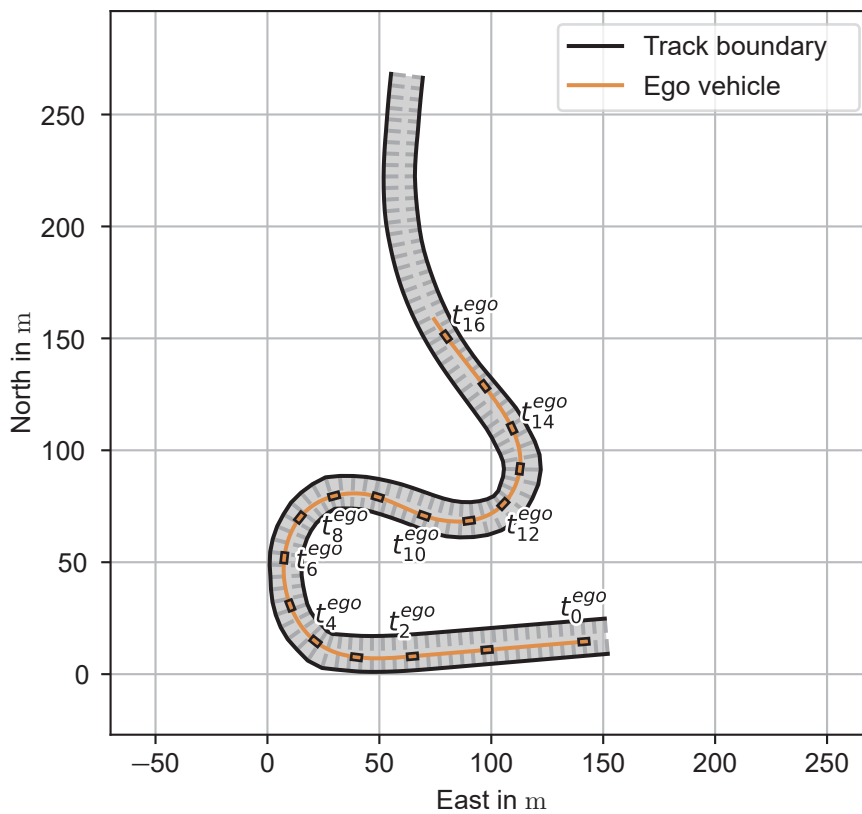


Figure A.74: Bird's eye view of scenario S70. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

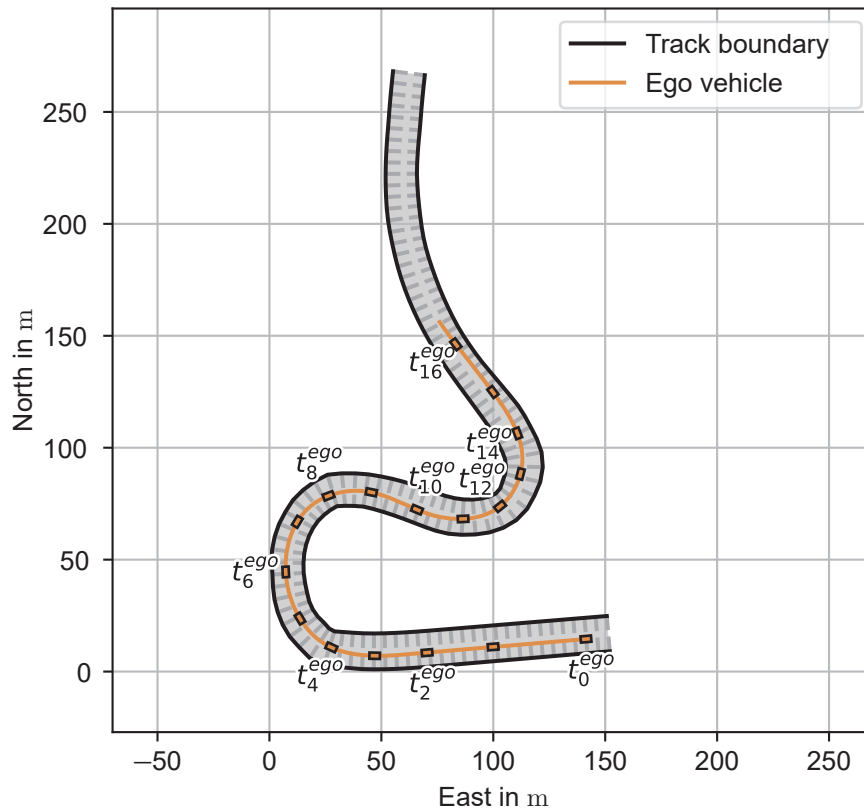


Figure A.75: Bird's eye view of scenario S71. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

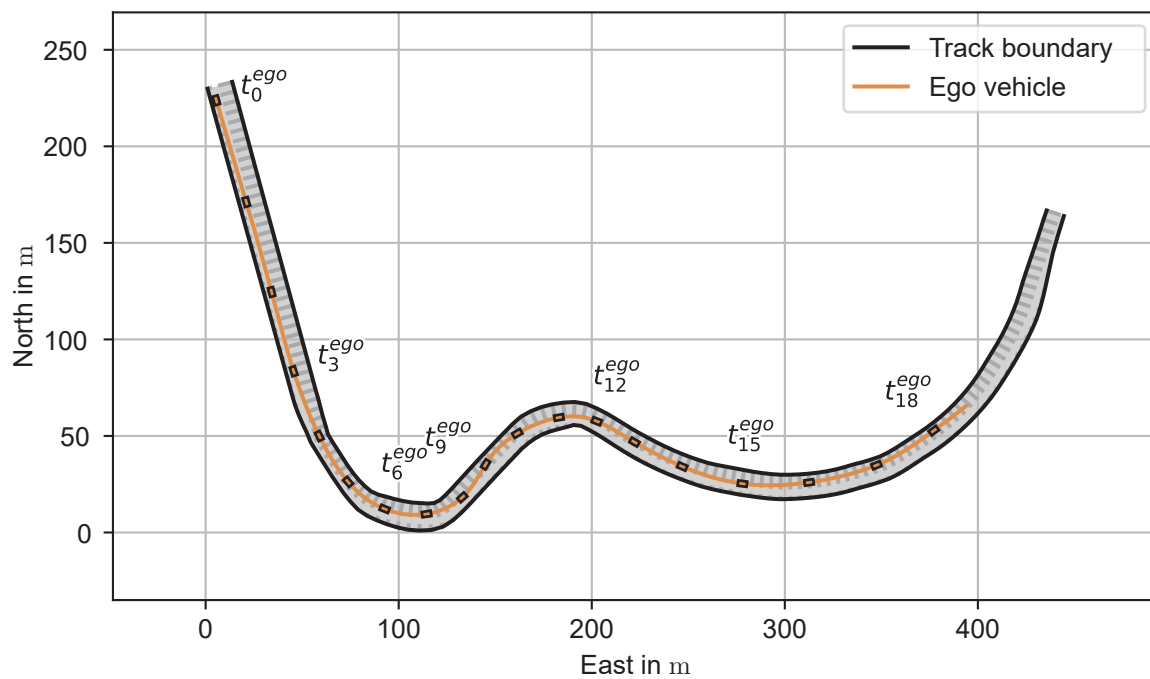


Figure A.76: Bird's eye view of scenario S72. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

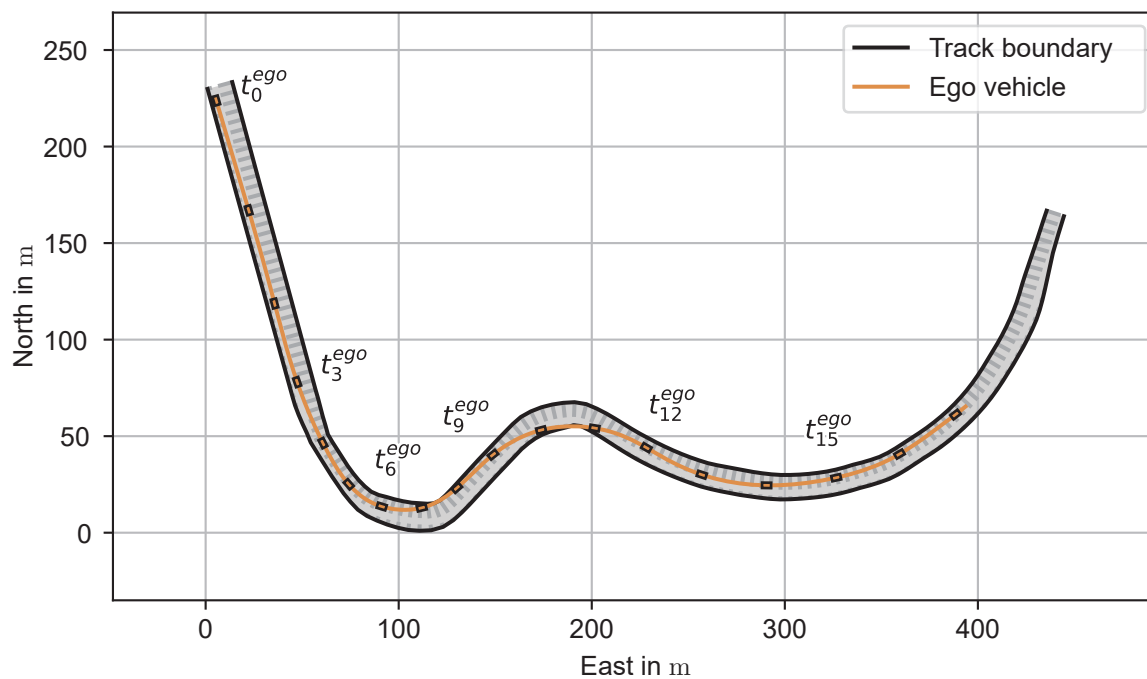


Figure A.77: Bird's eye view of scenario S73. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

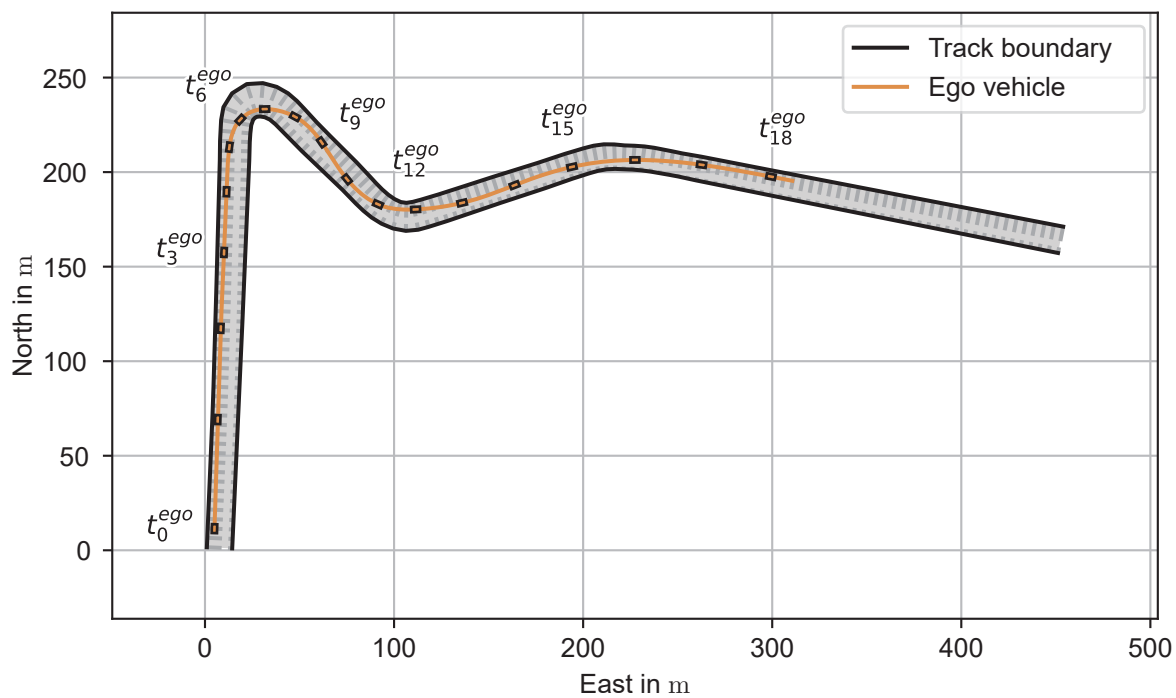


Figure A.78: Bird's eye view of scenario S74. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

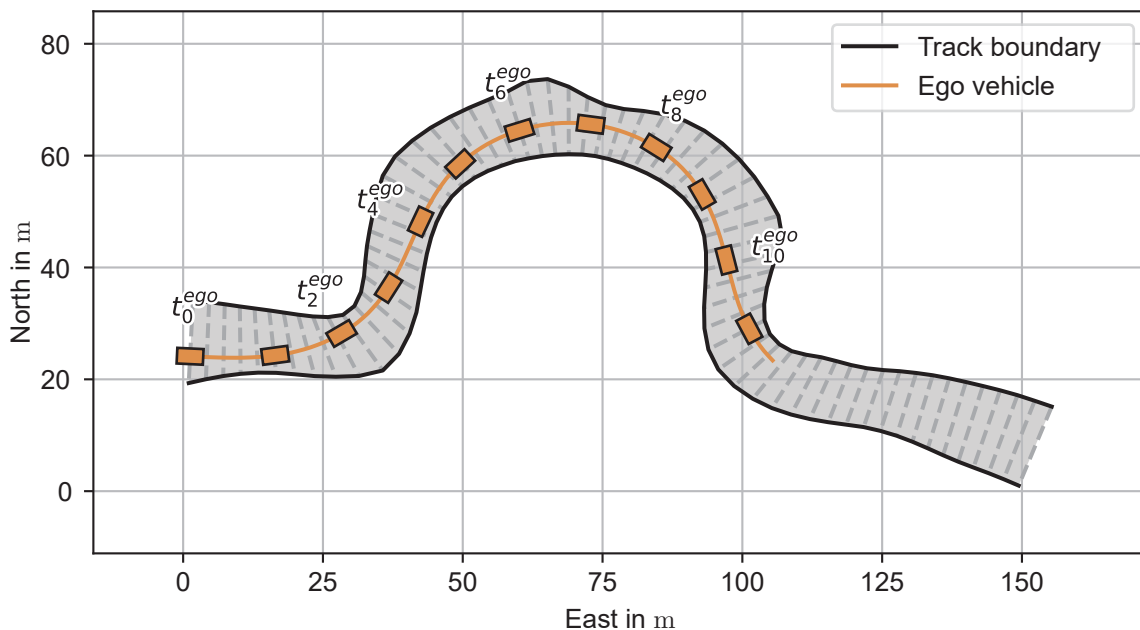


Figure A.79: Bird's eye view of scenario S75. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

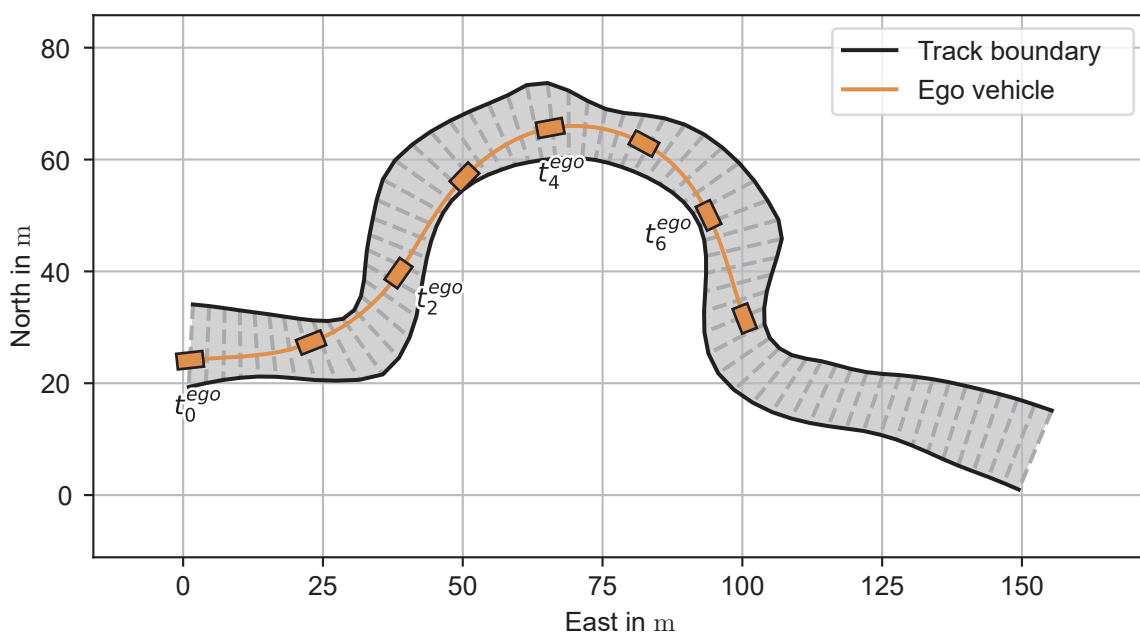


Figure A.80: Bird's eye view of scenario S76. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

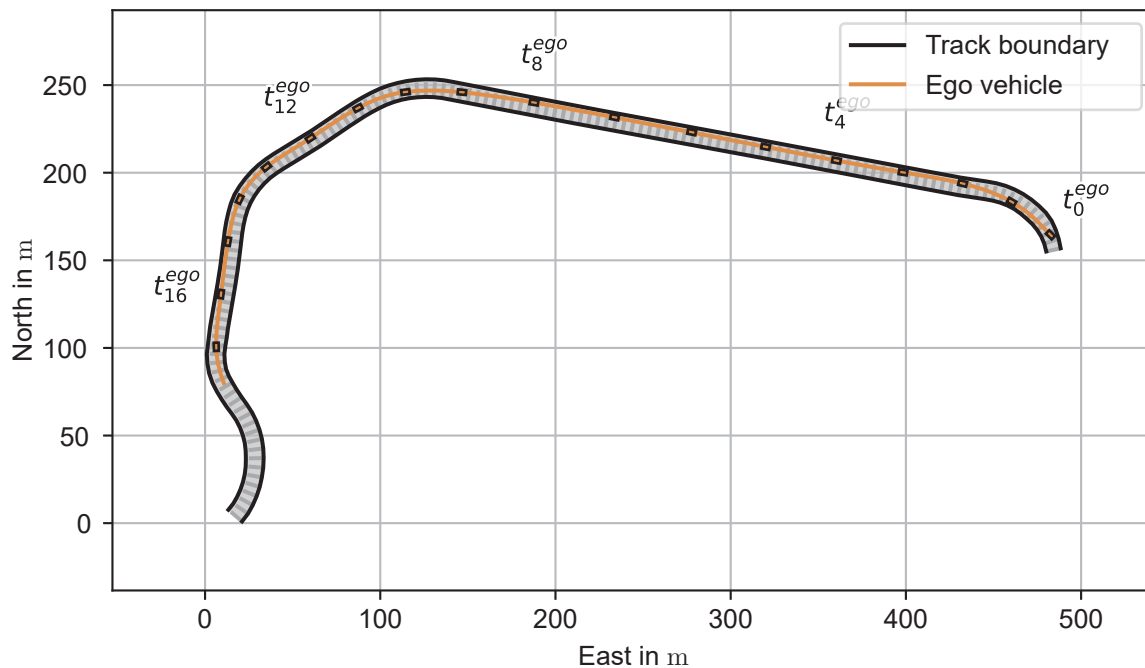


Figure A.81: Bird's eye view of scenario S77. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.

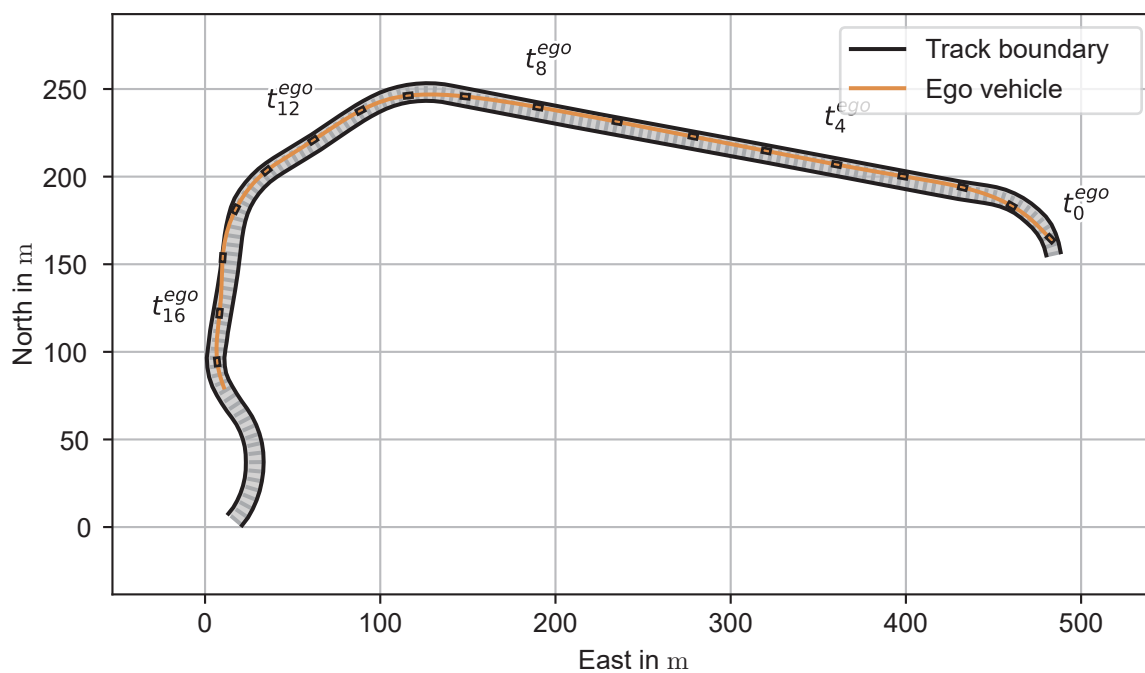


Figure A.82: Bird's eye view of scenario S78. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.

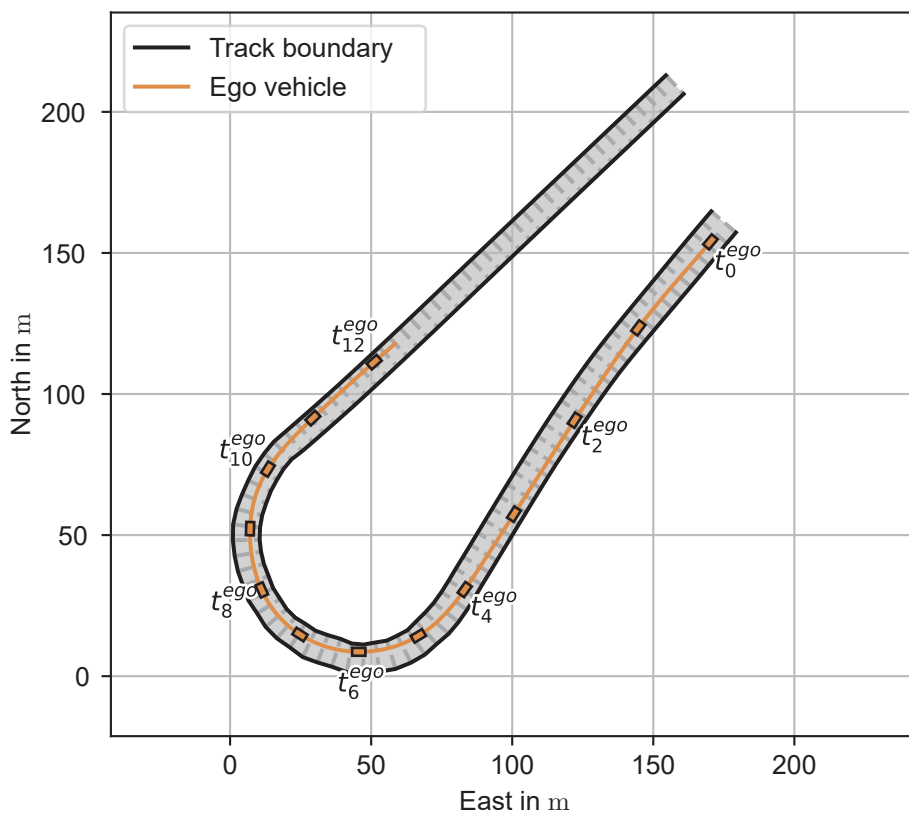


Figure A.83: Bird's eye view of scenario S79. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

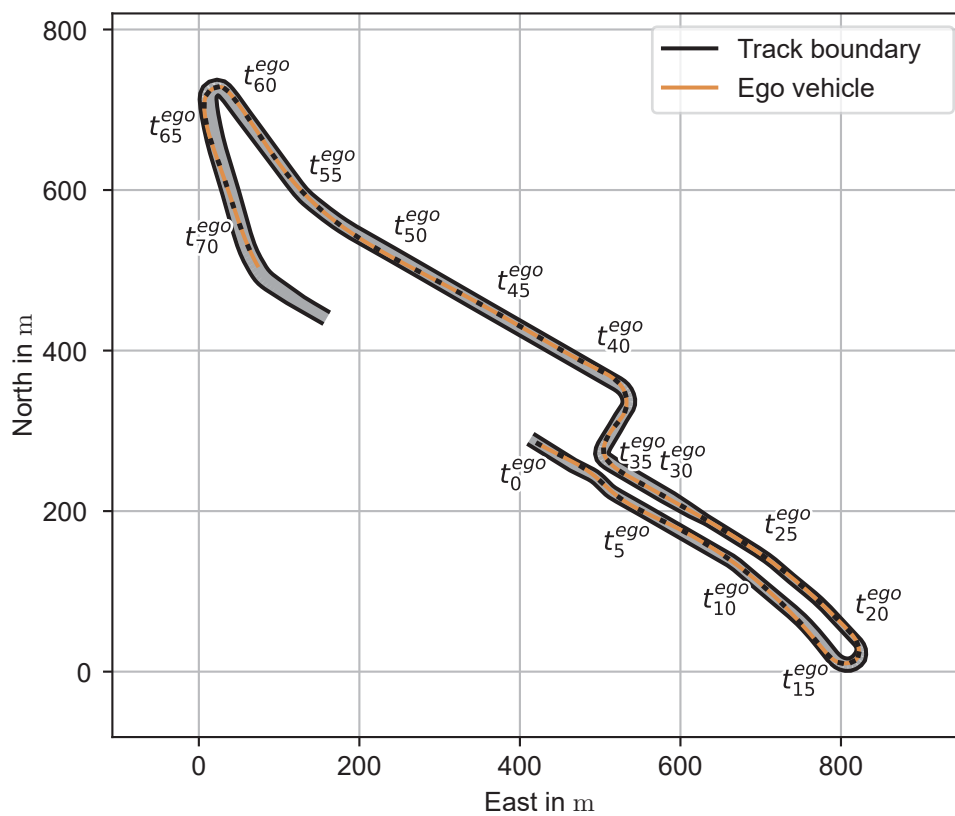


Figure A.84: Bird's eye view of scenario S80. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fifth interval is marked with a text label.

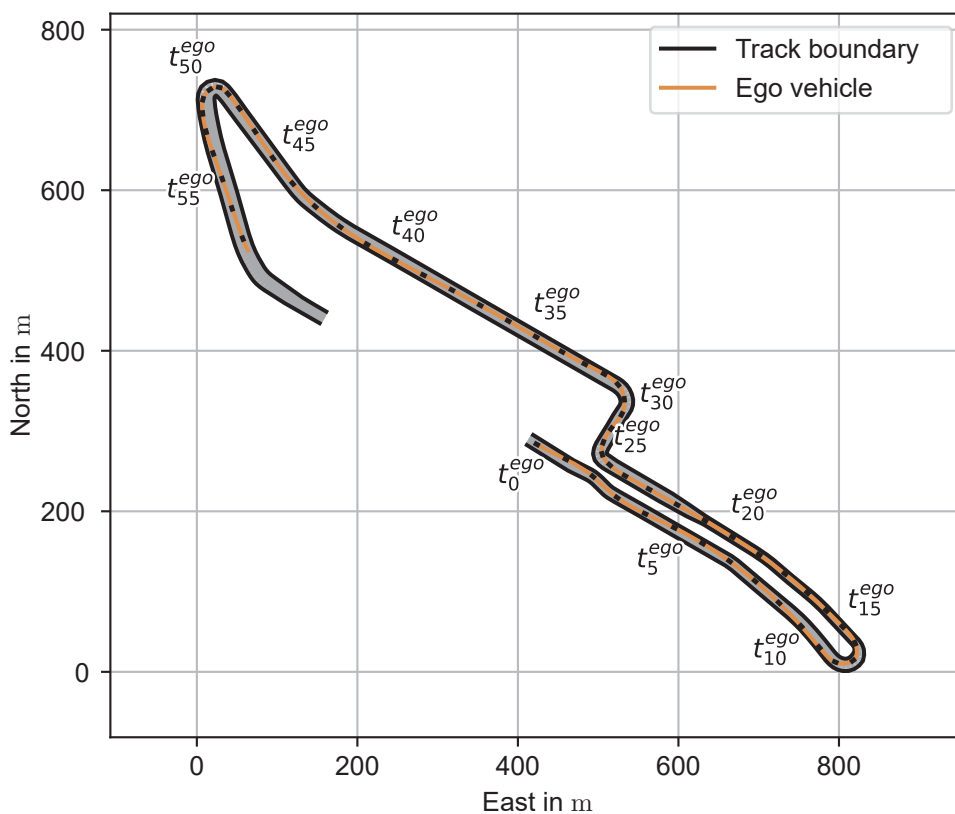


Figure A.85: Bird's eye view of scenario S81. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fifth interval is marked with a text label.

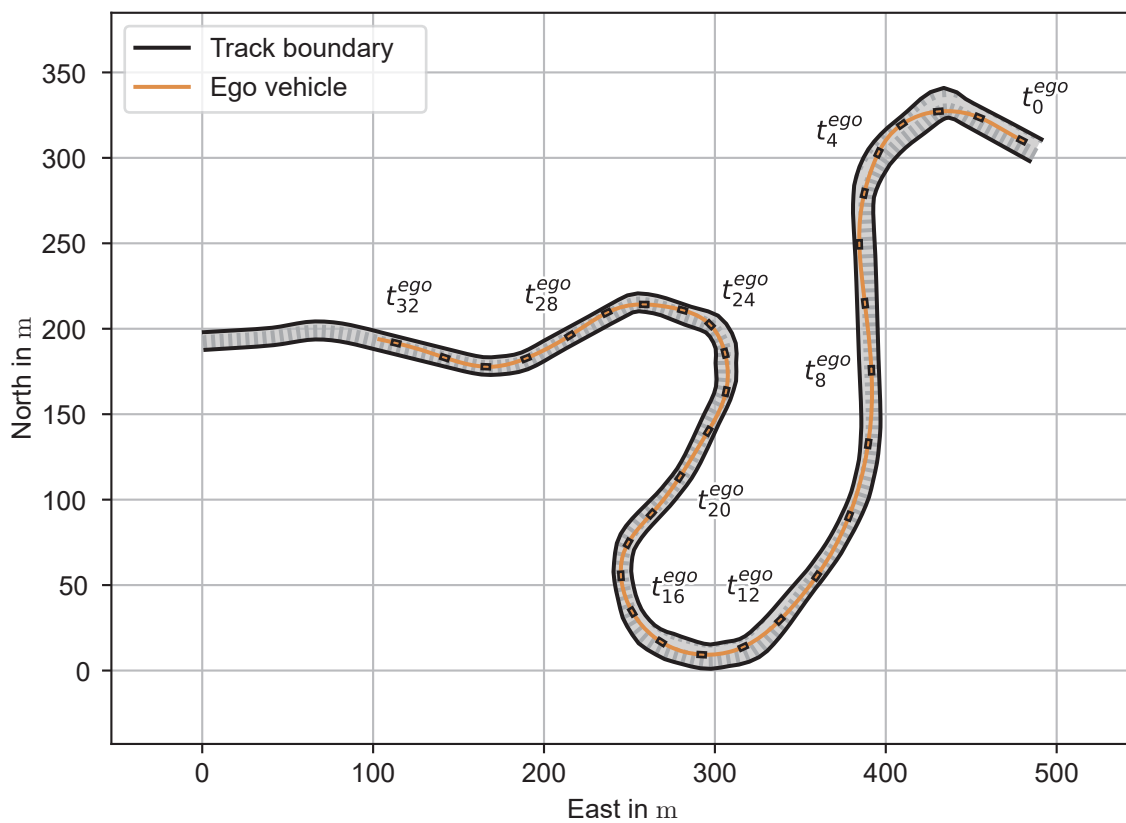


Figure A.86: Bird's eye view of scenario S82. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.

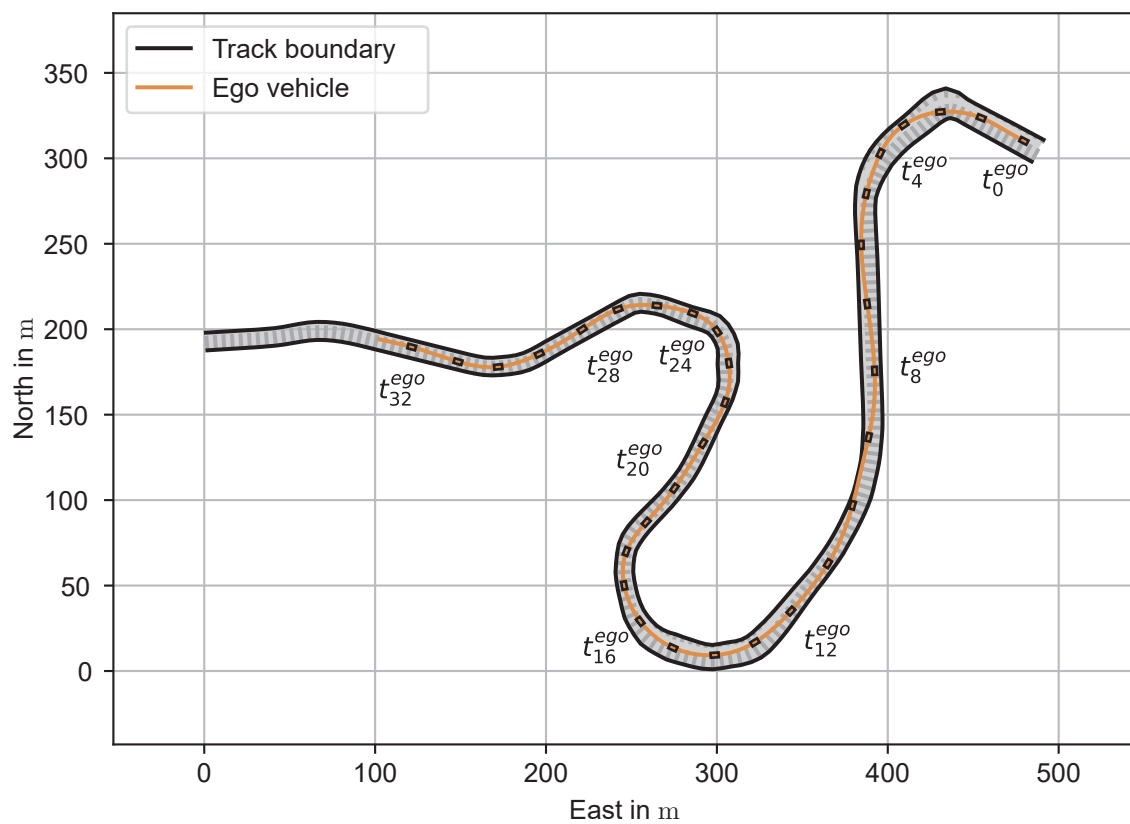


Figure A.87: Bird's eye view of scenario S83. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.

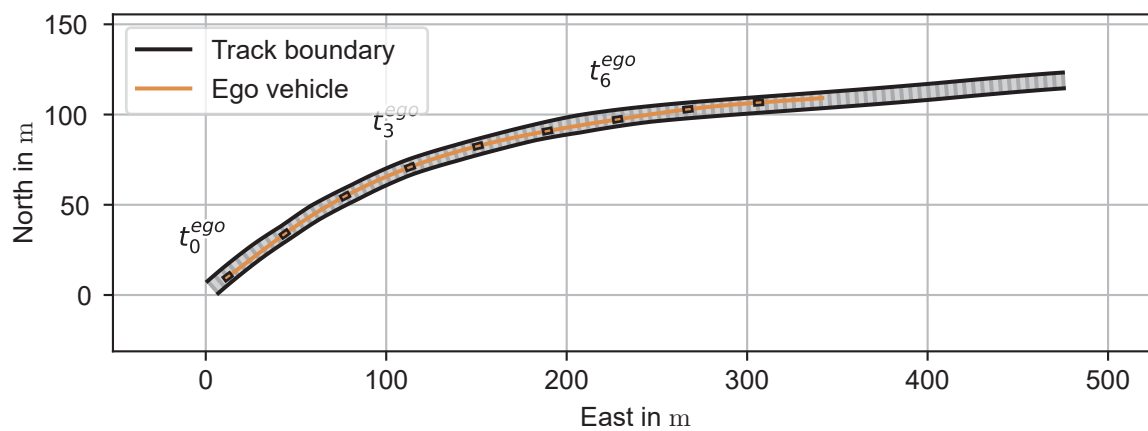


Figure A.88: Bird's eye view of scenario S84. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

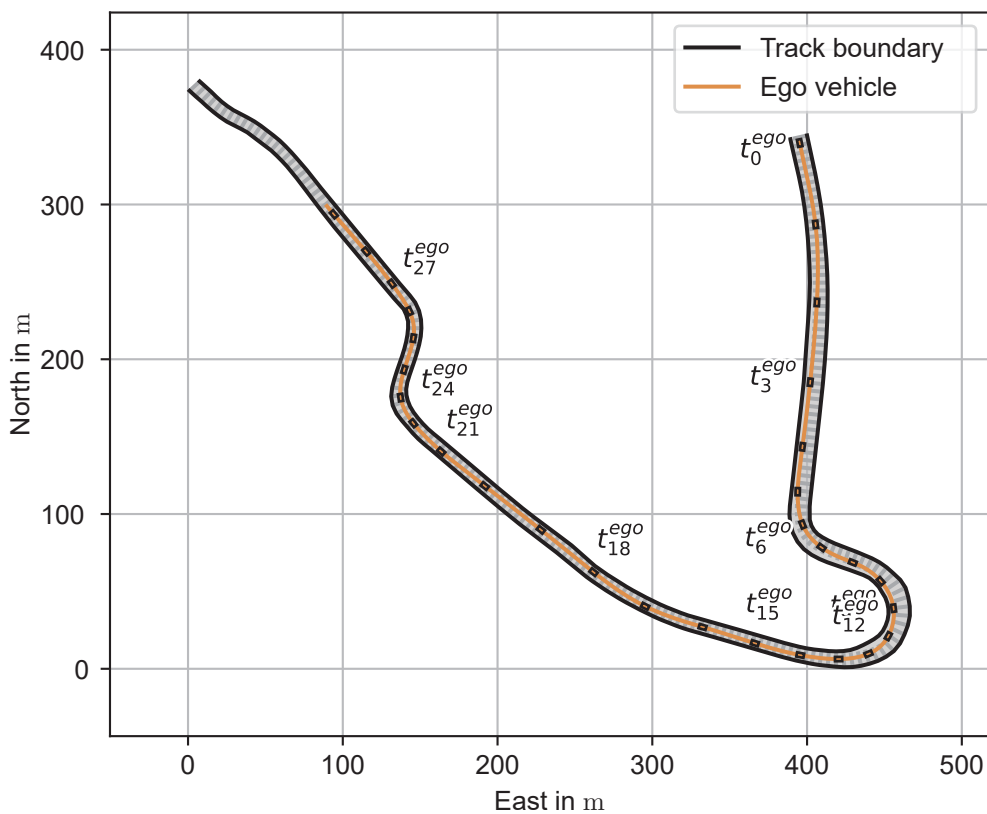


Figure A.89: Bird's eye view of scenario S85. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

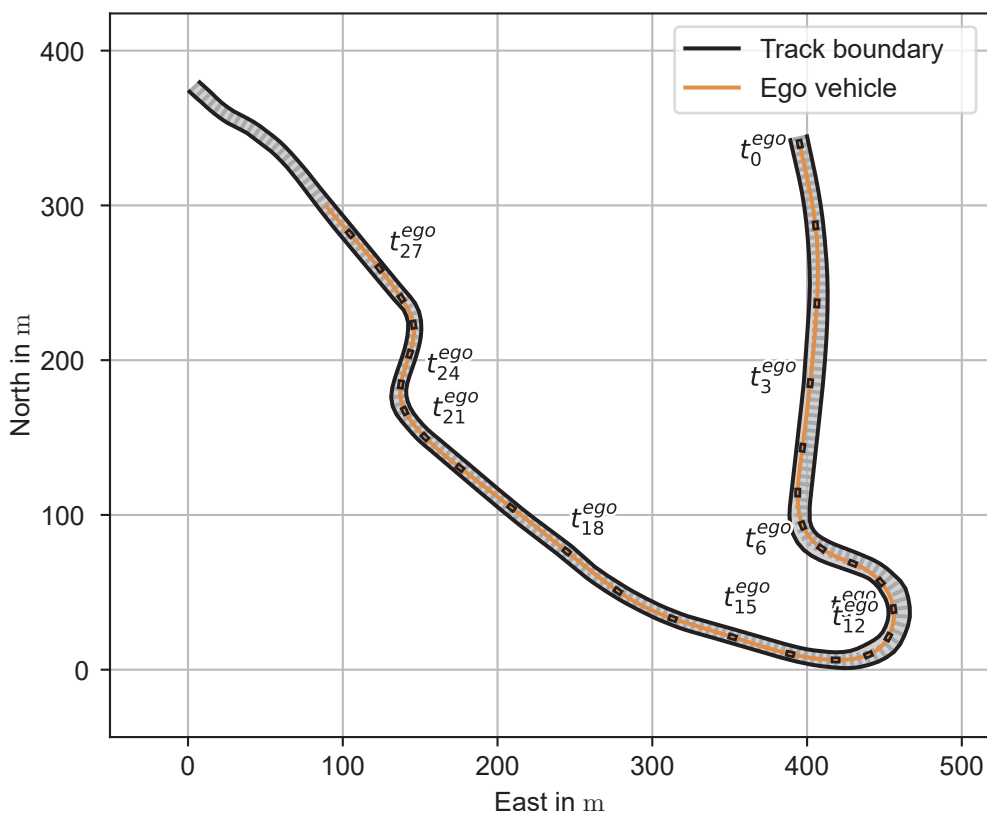


Figure A.90: Bird's eye view of scenario S86. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

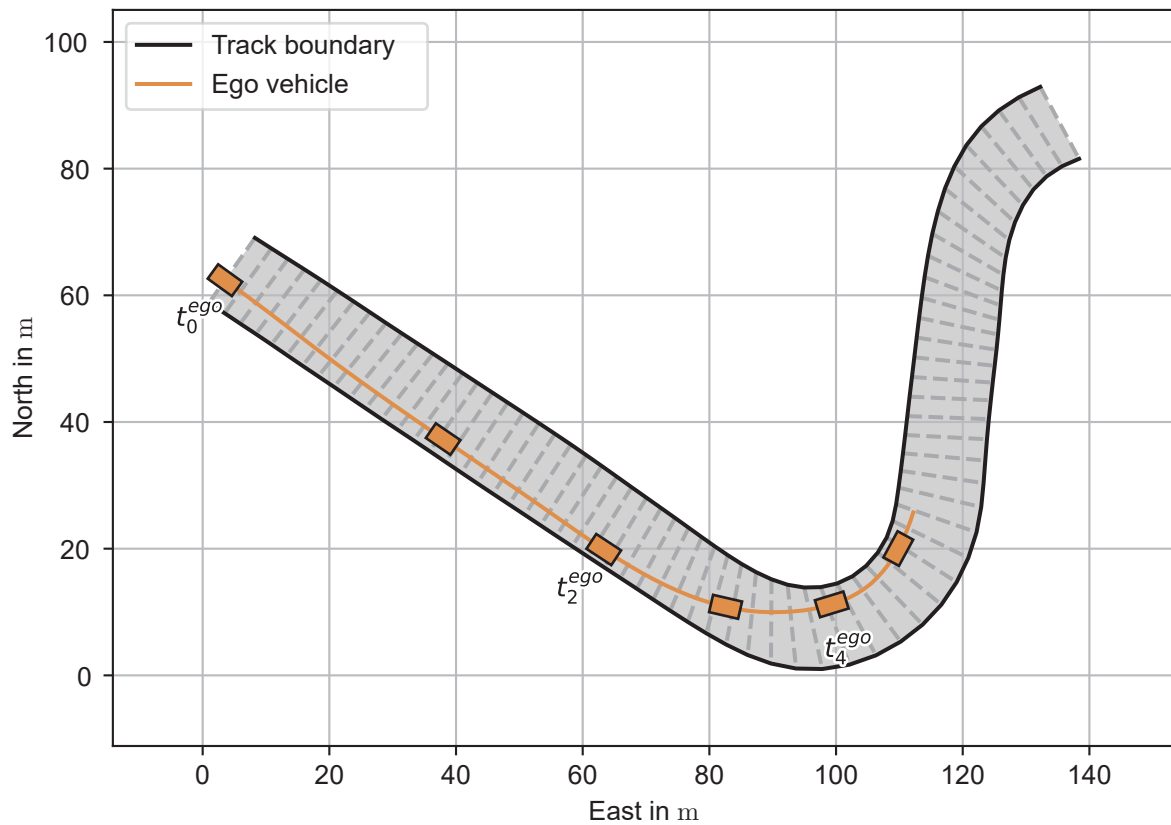


Figure A.91: Bird's eye view of scenario S87. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

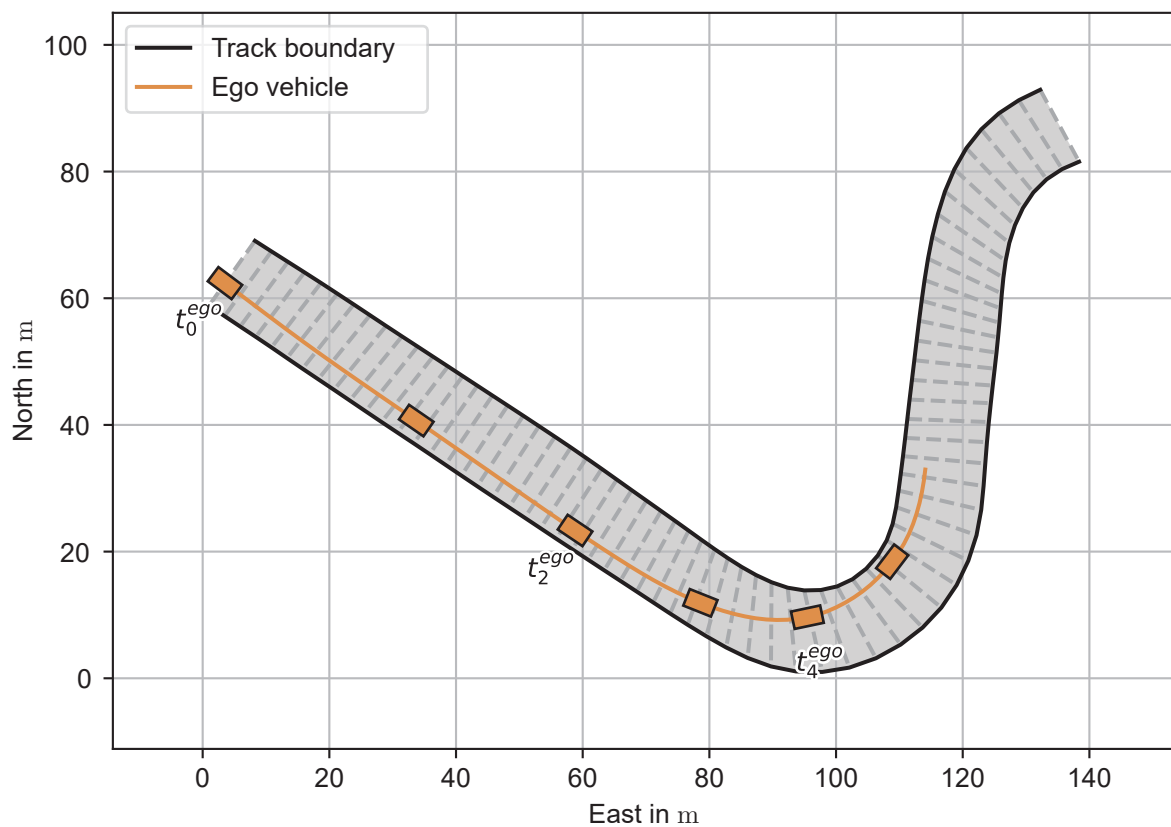


Figure A.92: Bird's eye view of scenario S88. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

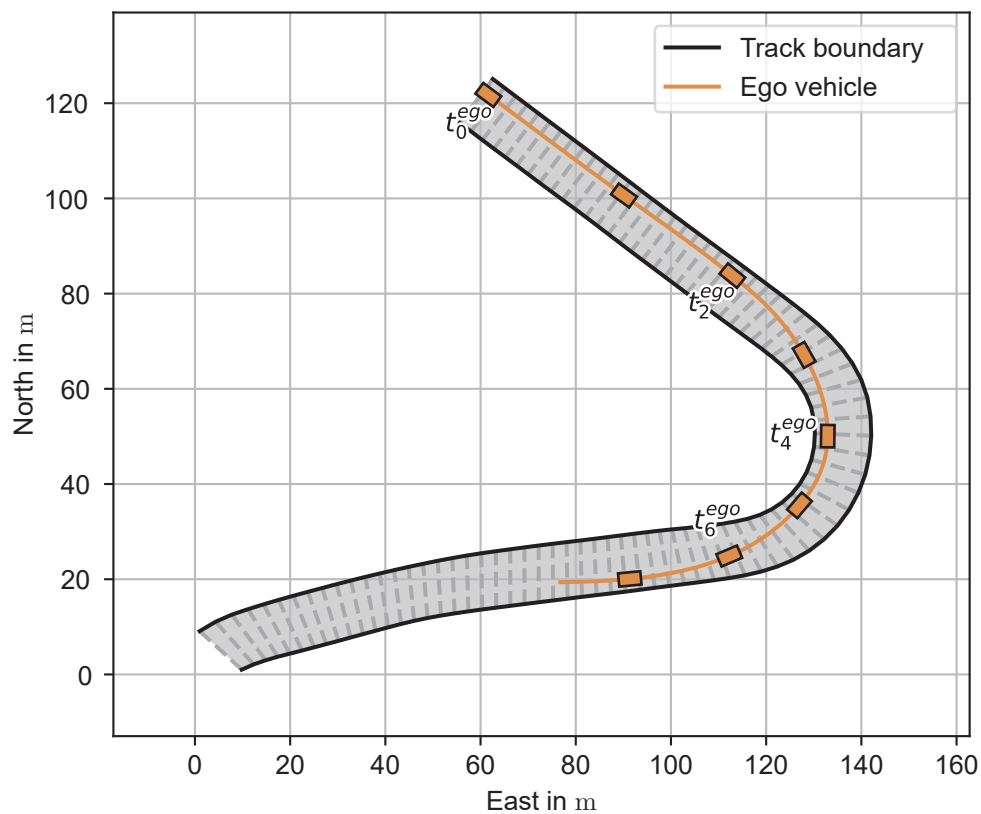


Figure A.93: Bird's eye view of scenario S89. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

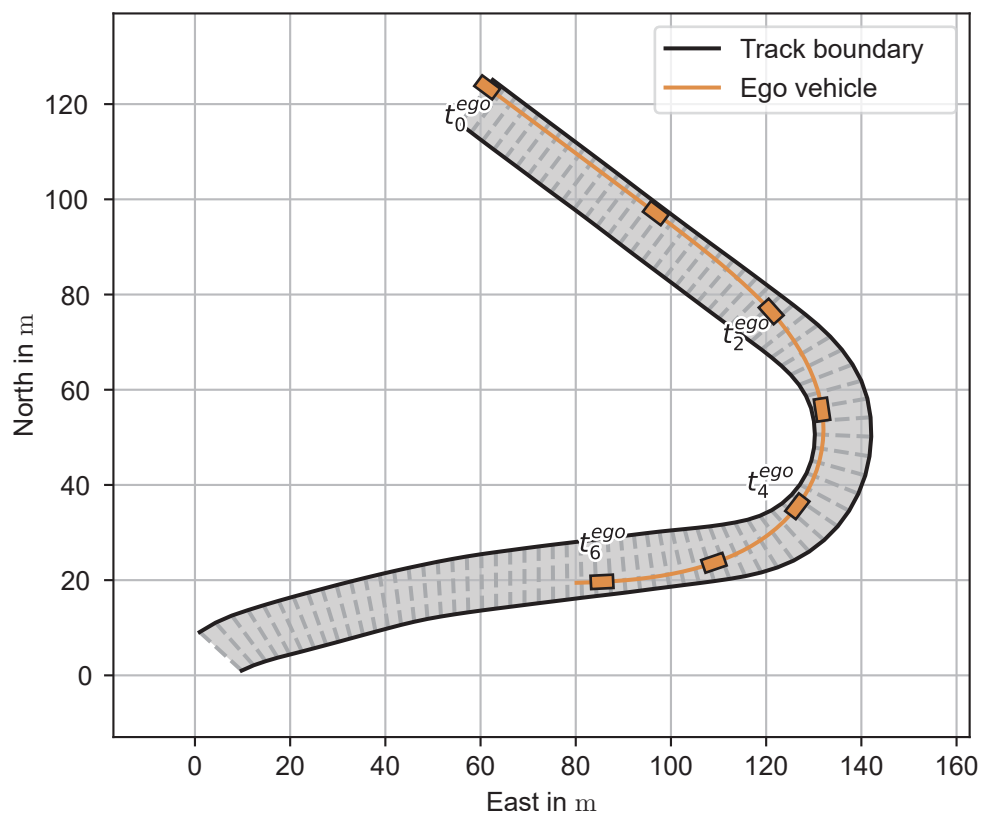


Figure A.94: Bird's eye view of scenario S90. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

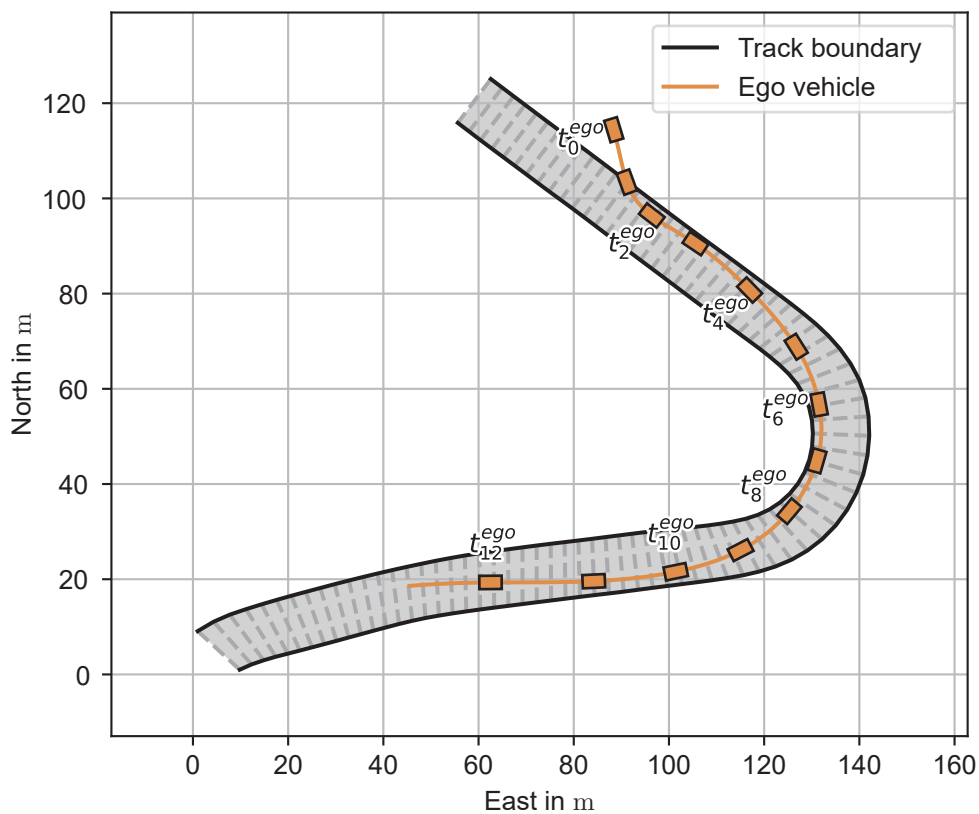


Figure A.95: Bird's eye view of scenario S91. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

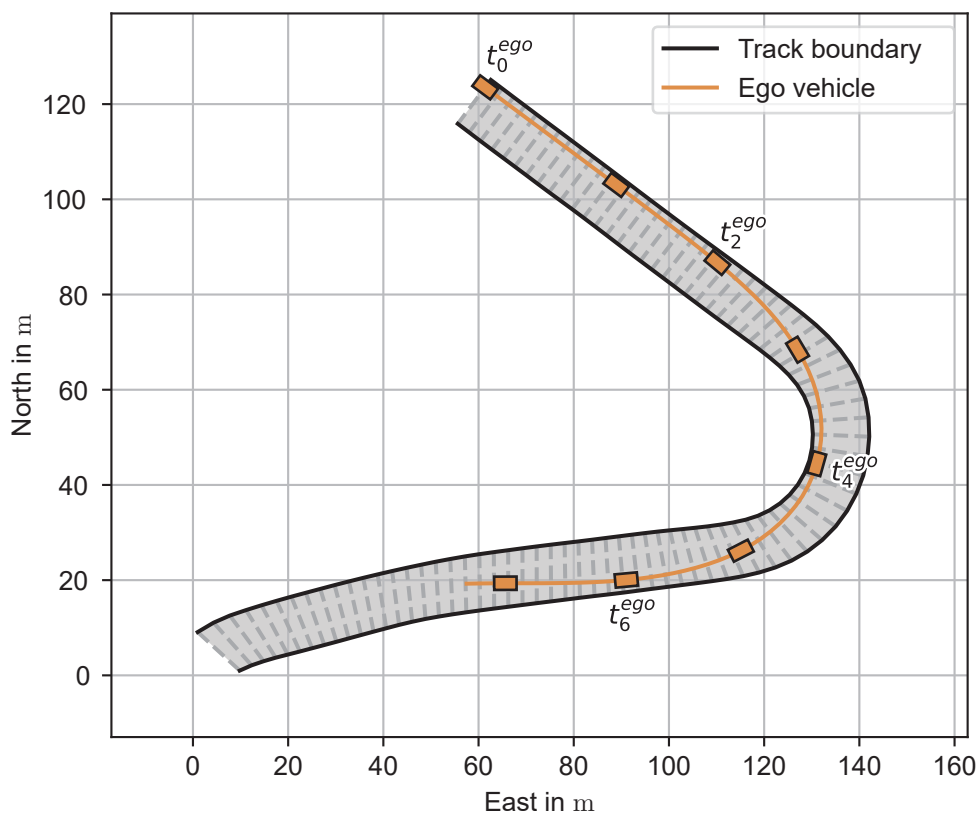


Figure A.96: Bird's eye view of scenario S92. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

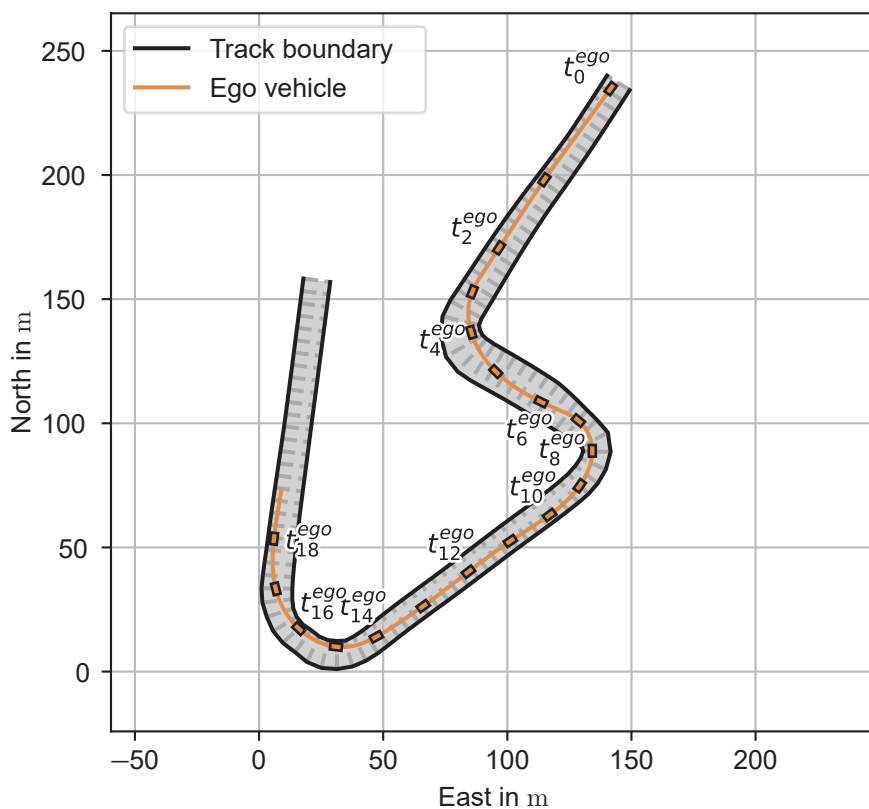


Figure A.97: Bird's eye view of scenario S93. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

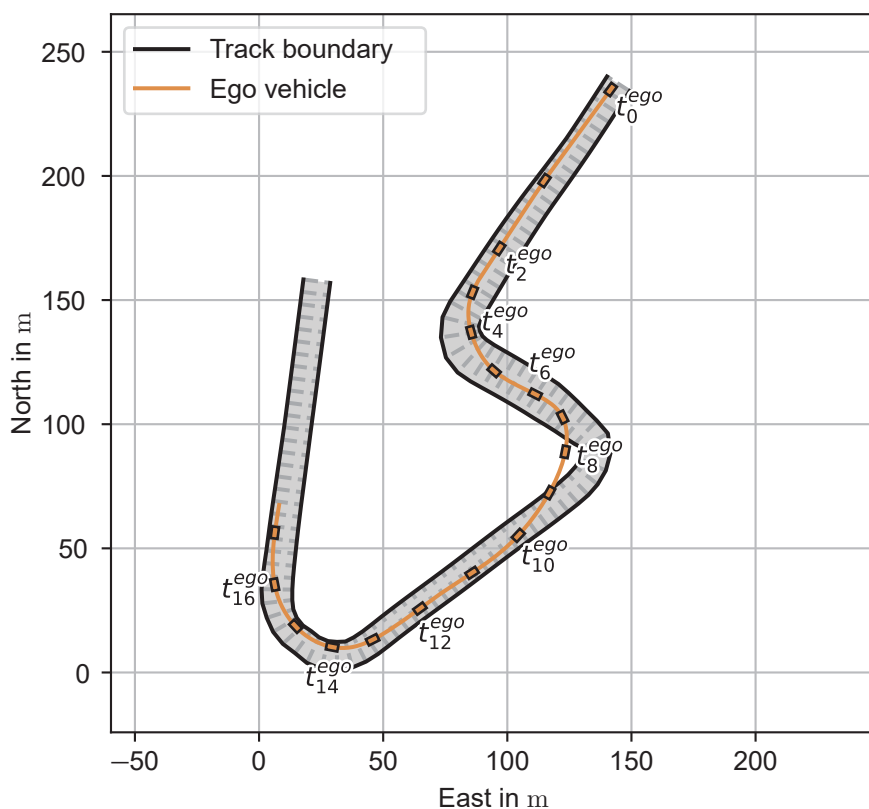


Figure A.98: Bird's eye view of scenario S94. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

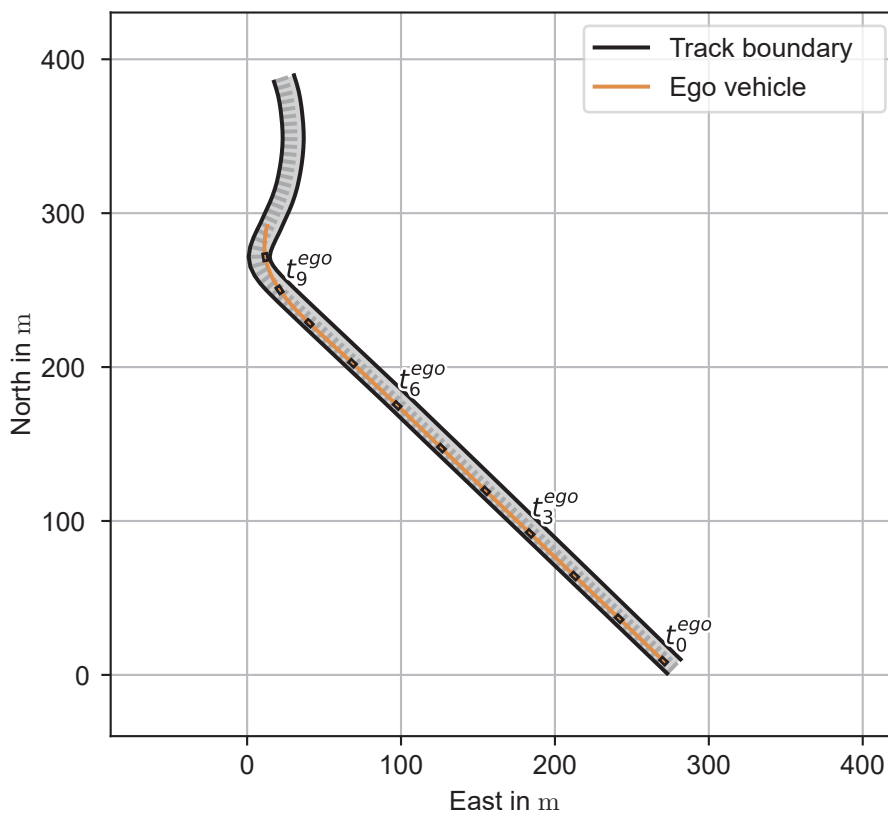


Figure A.99: Bird's eye view of scenario S95. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

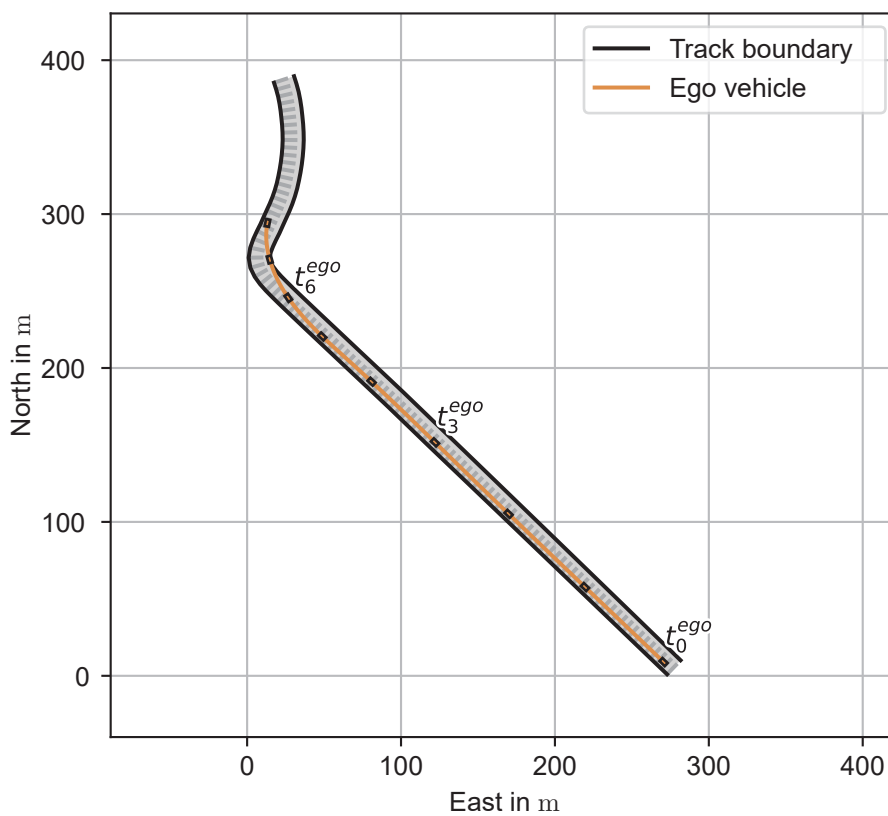


Figure A.100: Bird's eye view of scenario S96. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

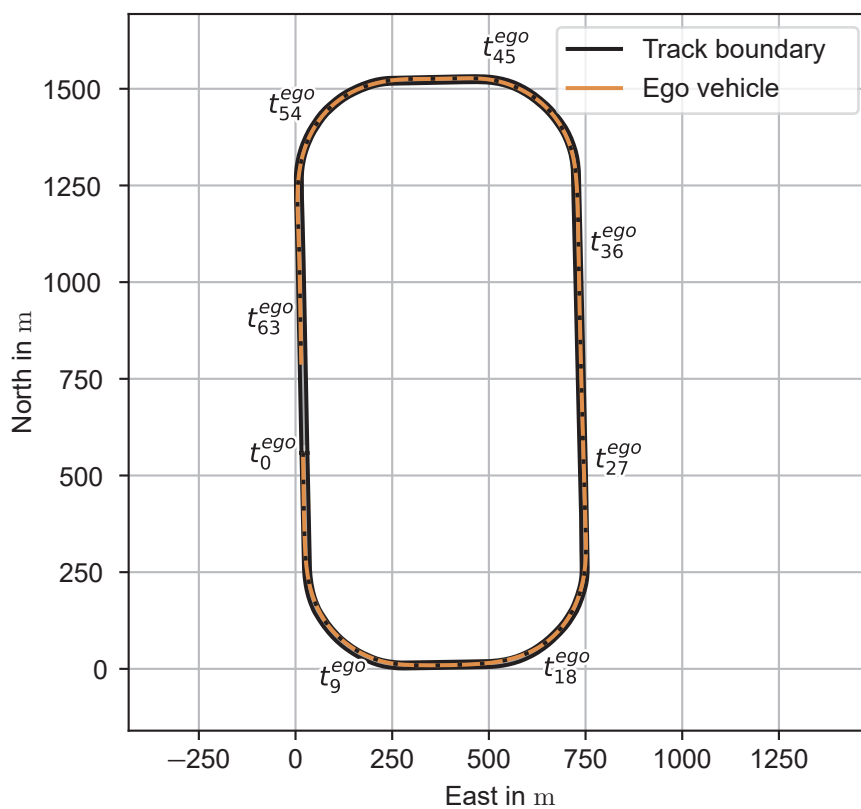


Figure A.101: Bird's eye view of scenario S97. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every ninth interval is marked with a text label.

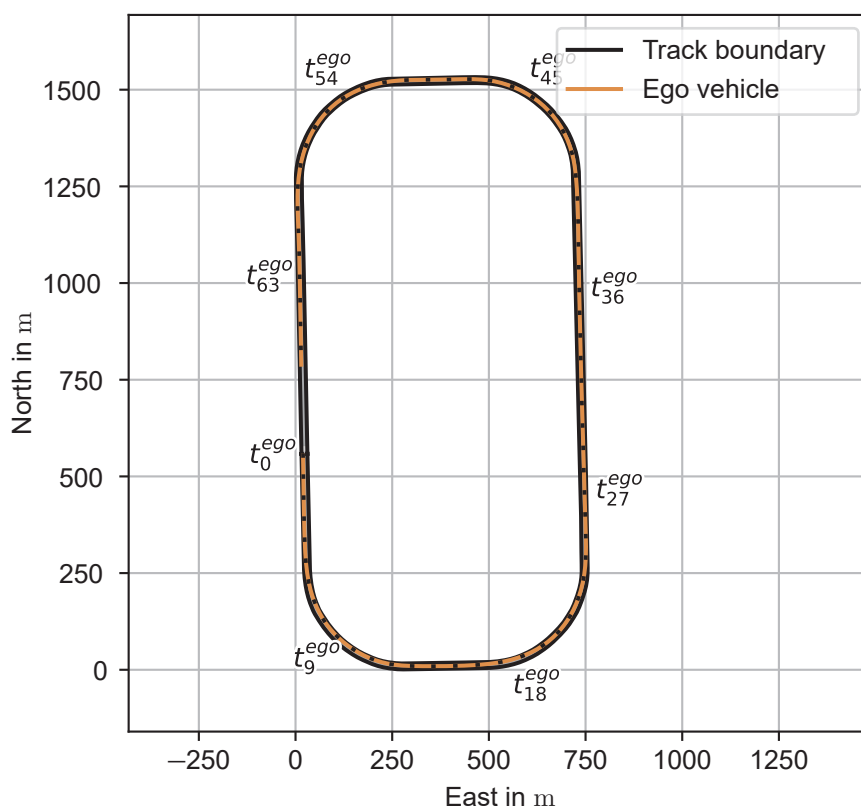


Figure A.102: Bird's eye view of scenario S98. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every ninth interval is marked with a text label.

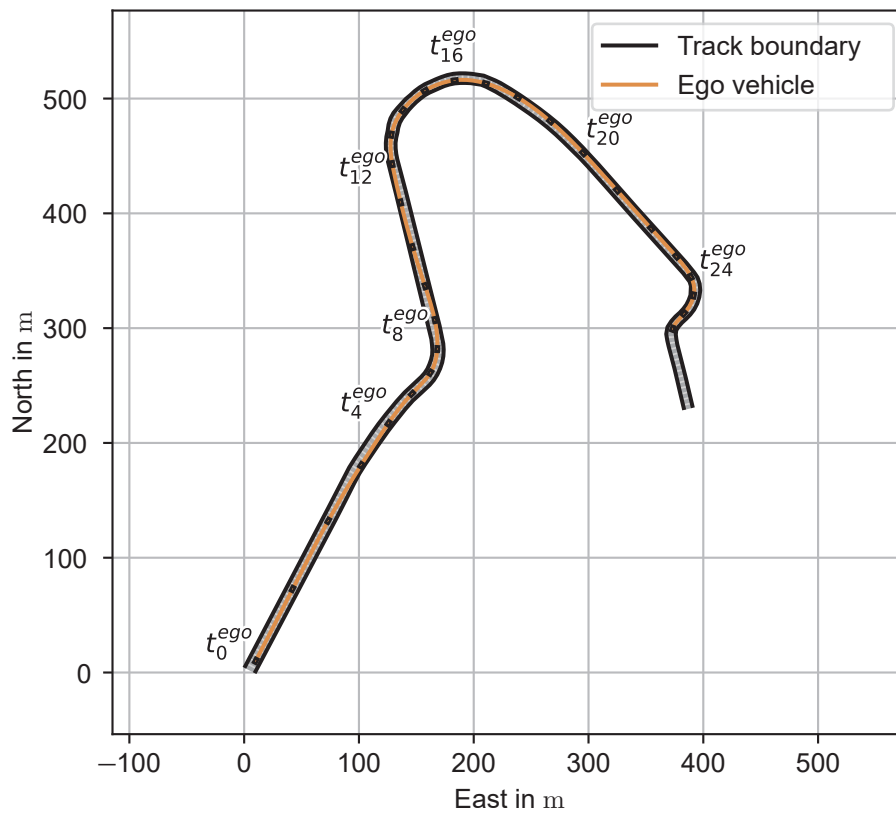


Figure A.103: Bird's eye view of scenario S99. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.

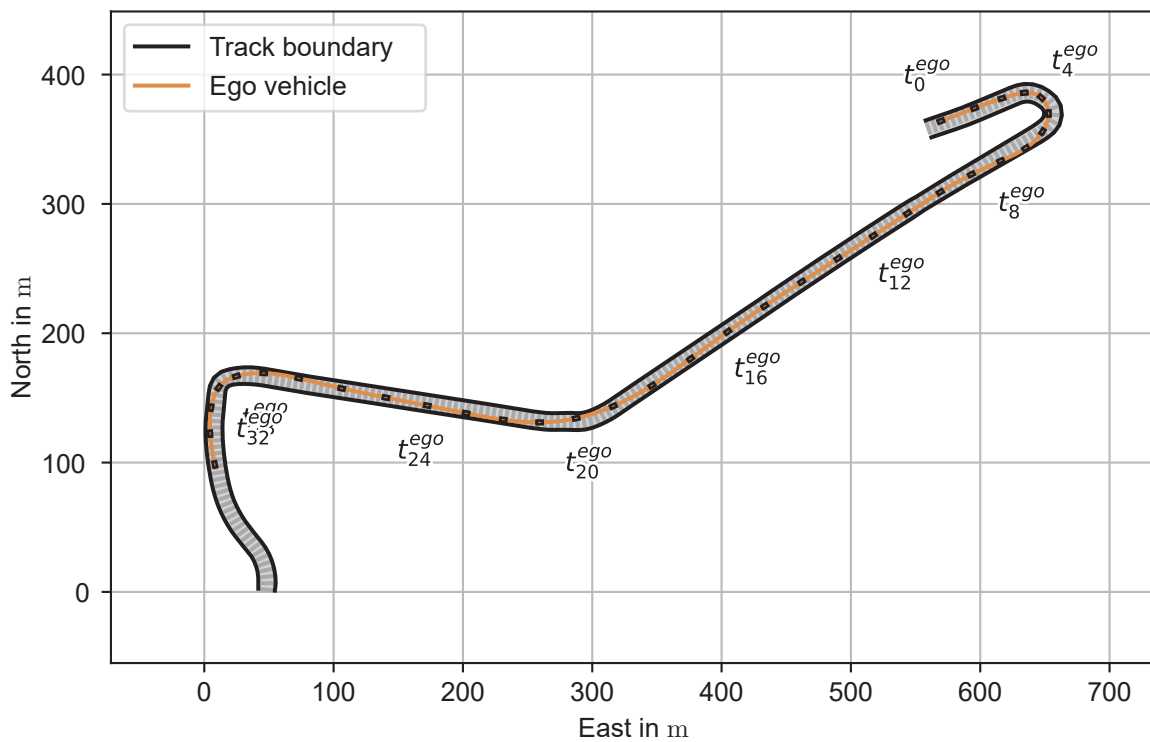


Figure A.104: Bird's eye view of scenario S100. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.

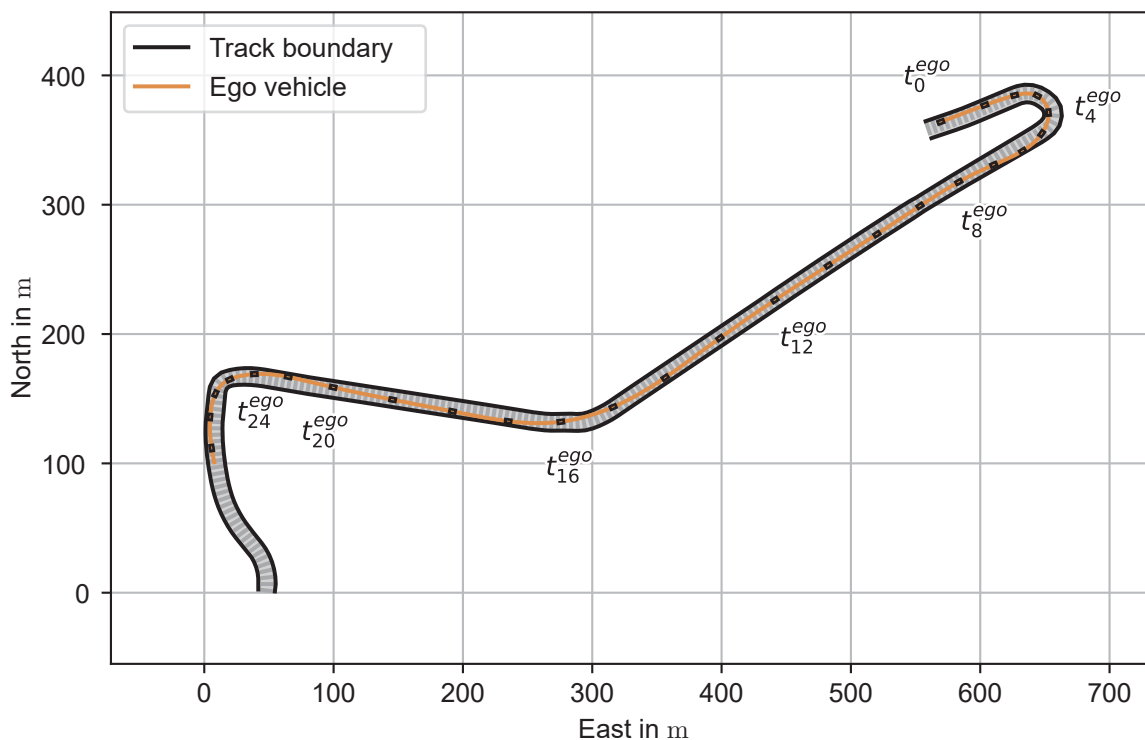


Figure A.105: Bird's eye view of scenario S101. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.

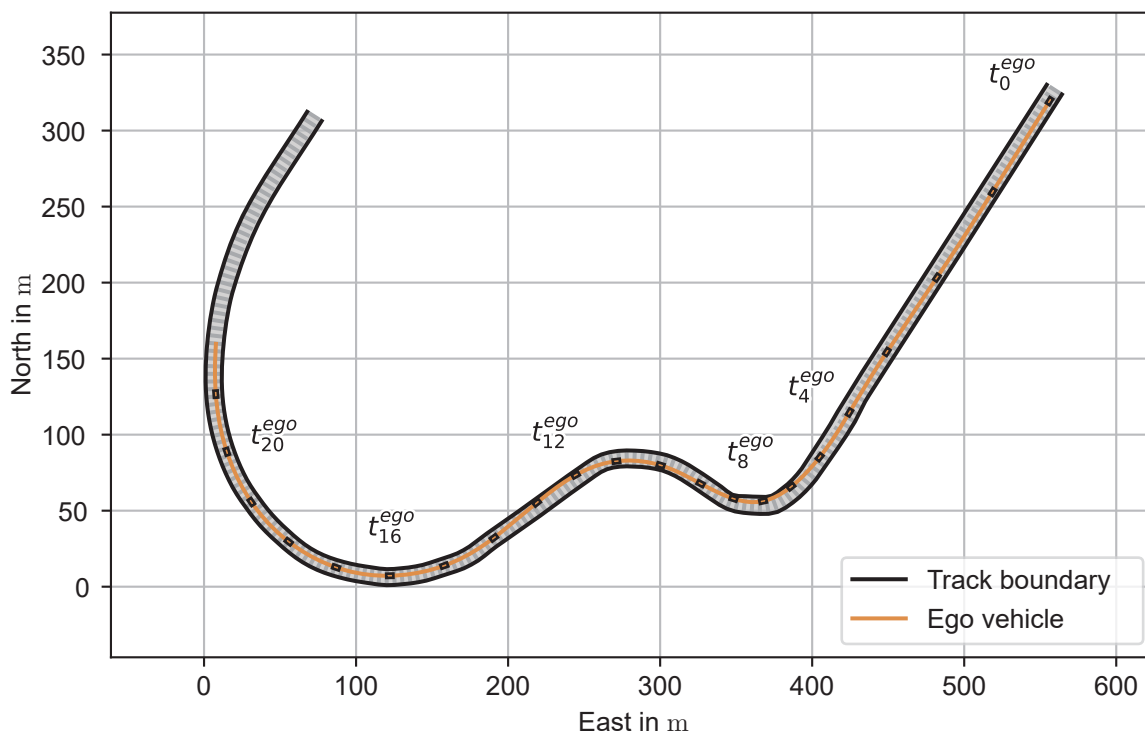


Figure A.106: Bird's eye view of scenario S102. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.

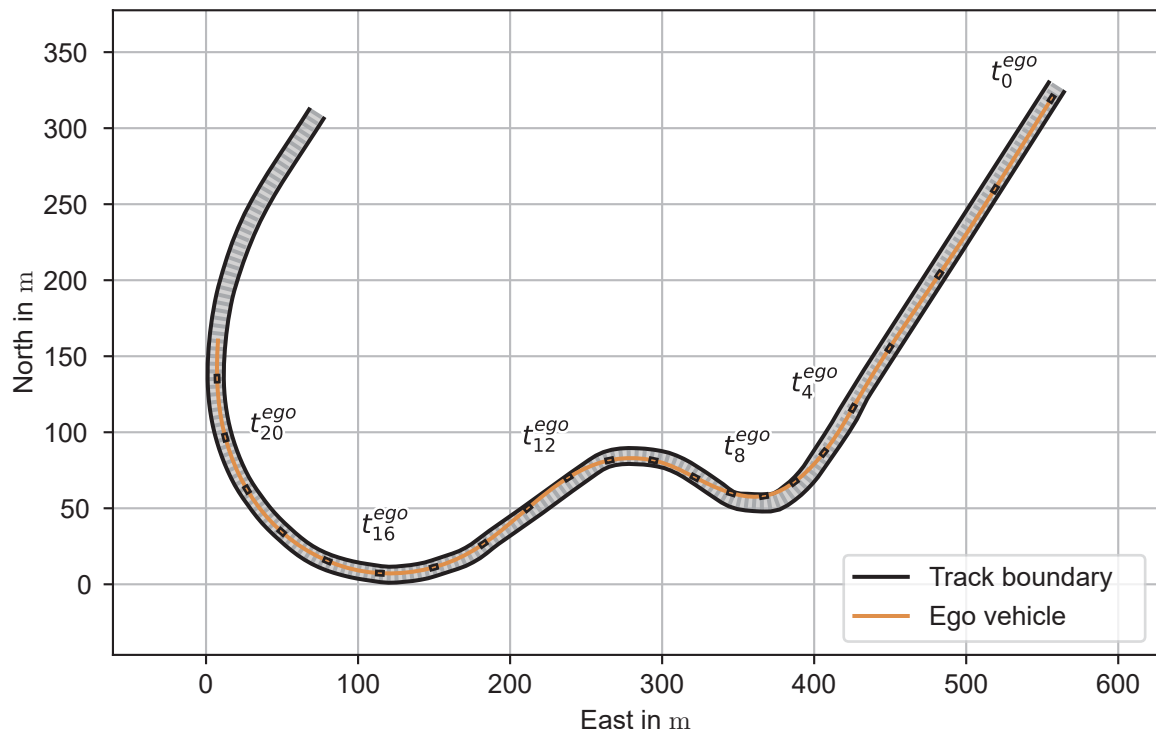


Figure A.107: Bird's eye view of scenario S103. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fourth interval is marked with a text label.

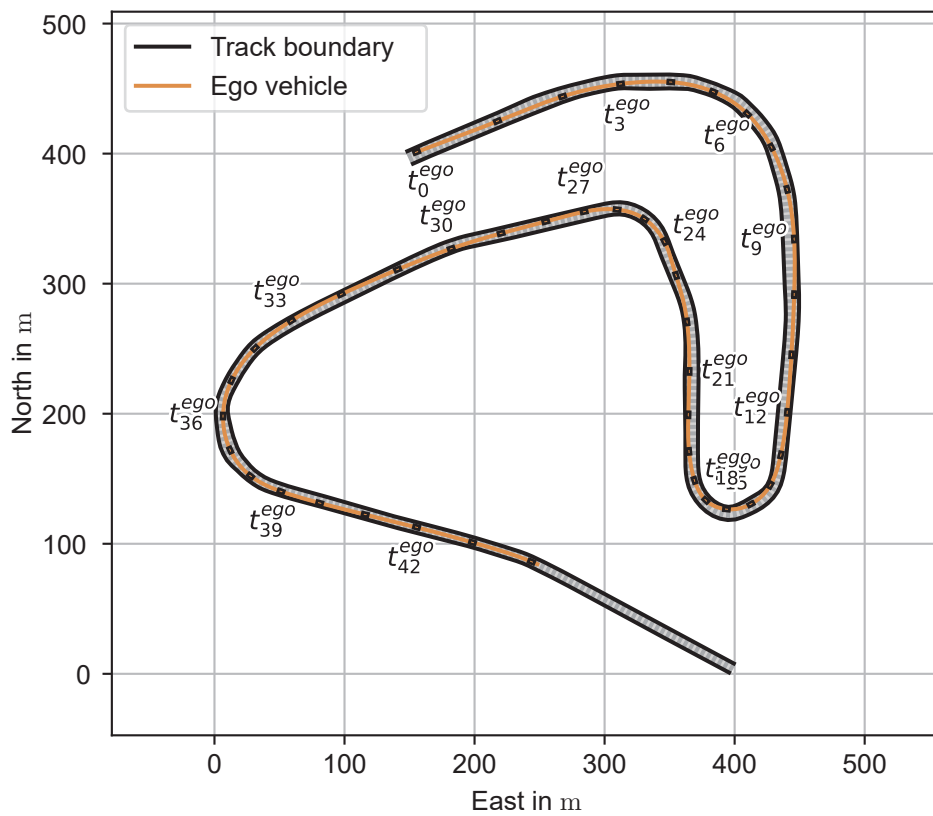


Figure A.108: Bird's eye view of scenario S104. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

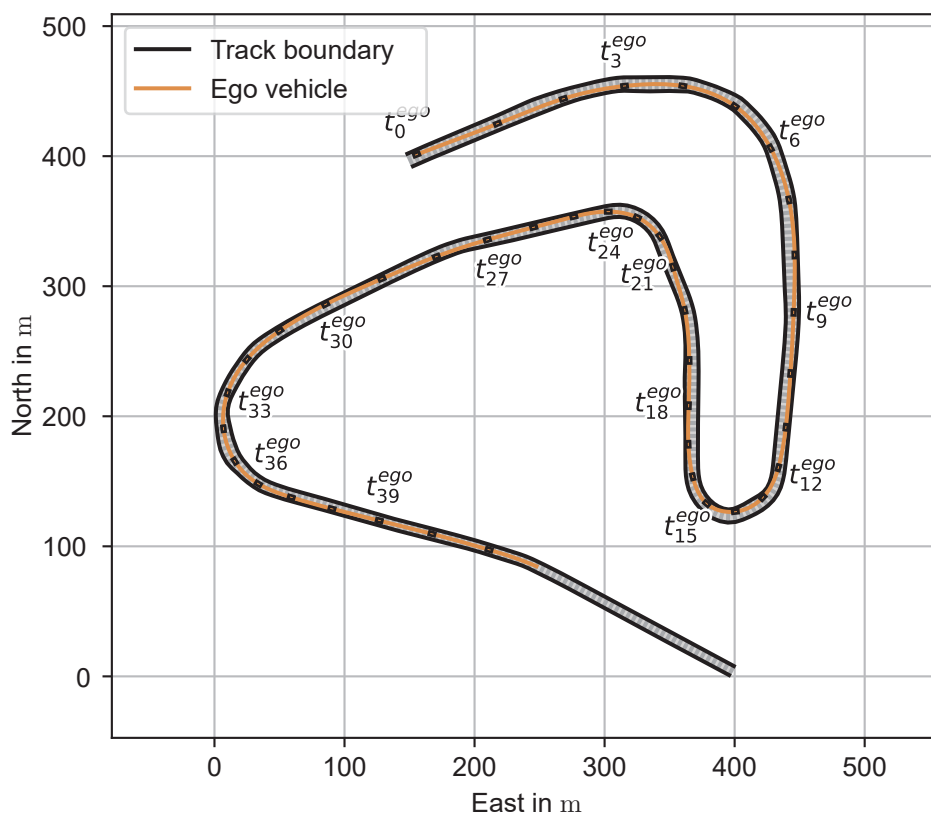


Figure A.109: Bird's eye view of scenario S105. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

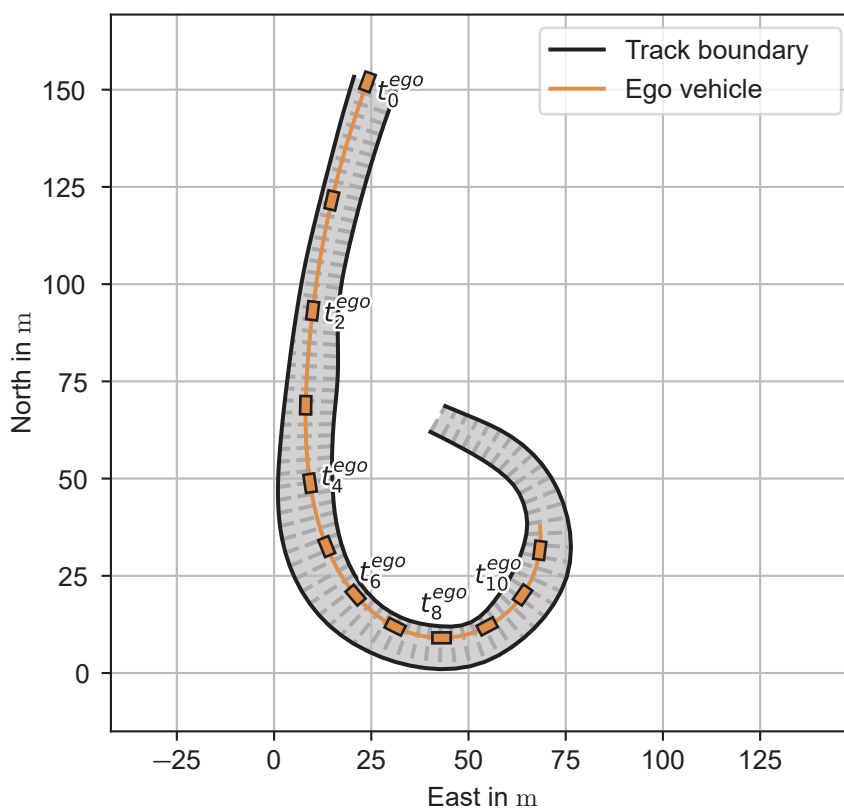


Figure A.110: Bird's eye view of scenario S106. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

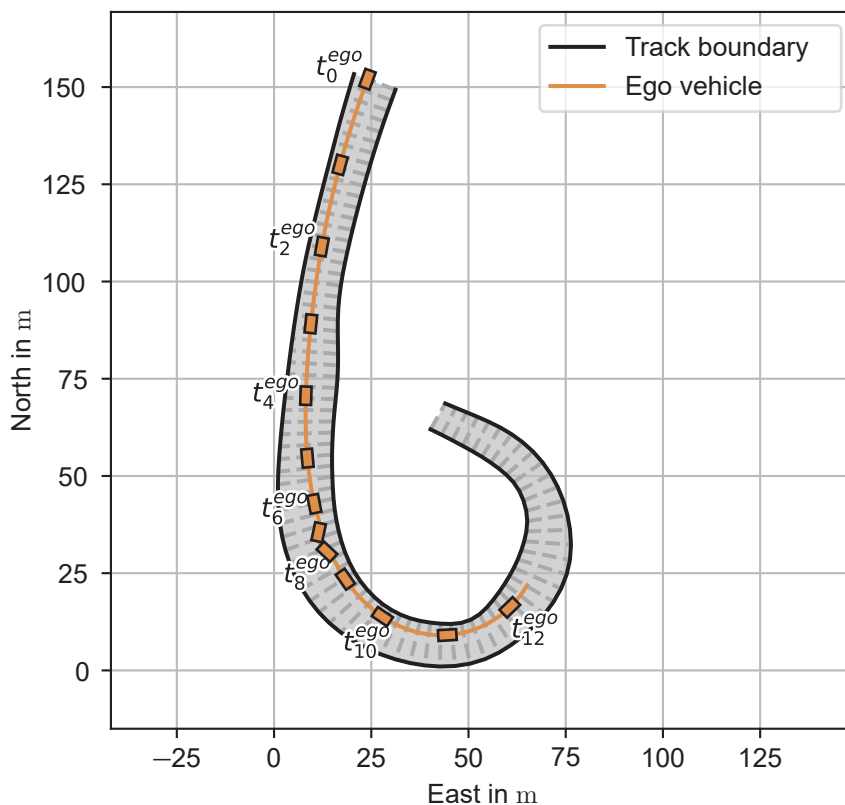


Figure A.111: Bird's eye view of scenario S107. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

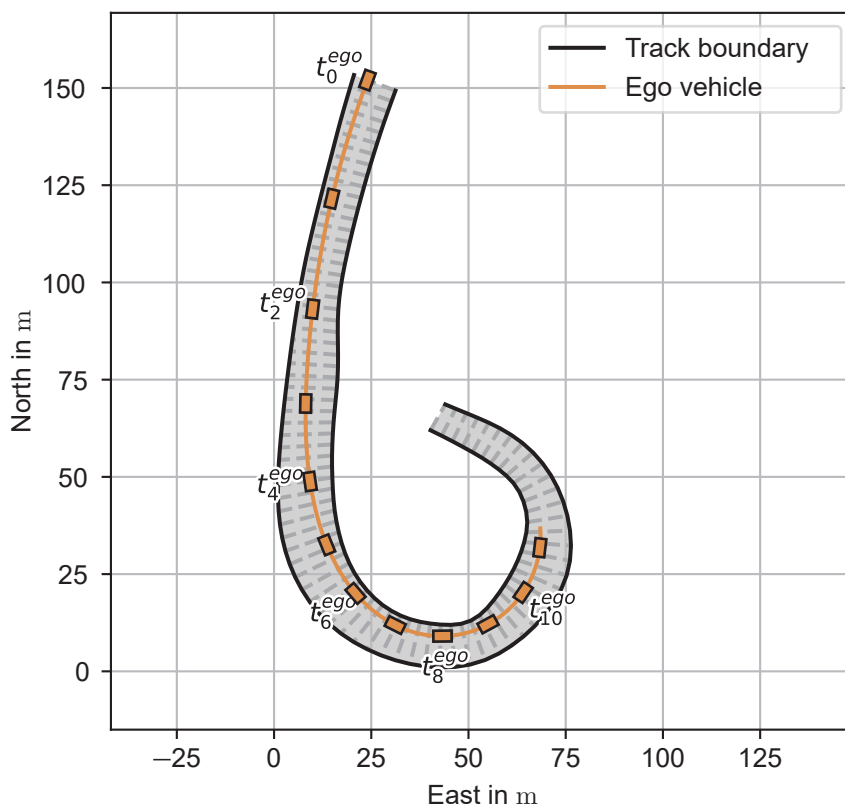


Figure A.112: Bird's eye view of scenario S108. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

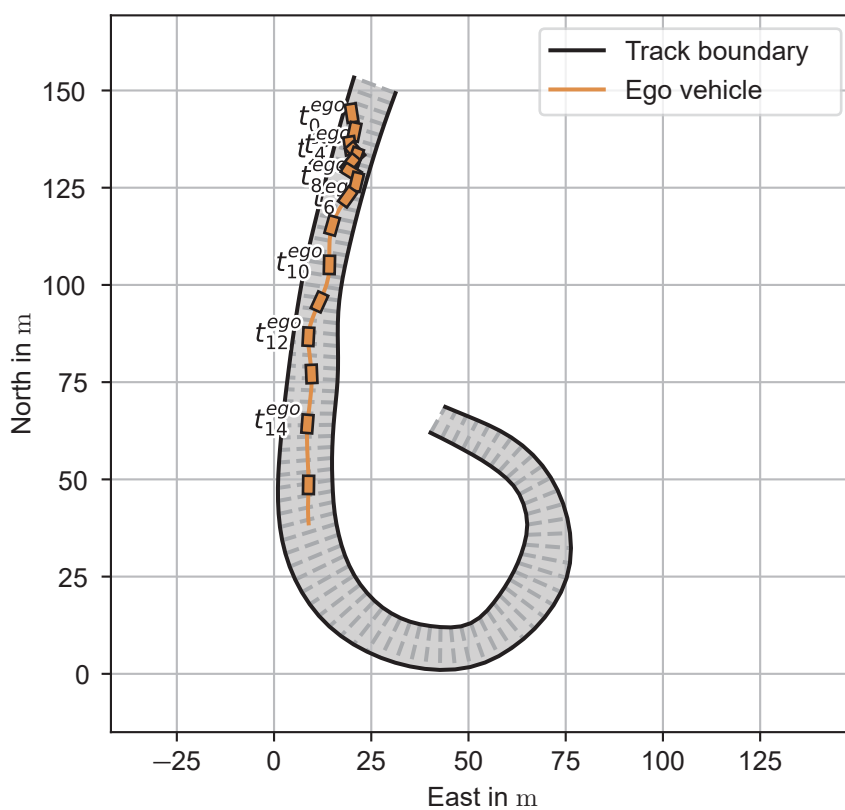


Figure A.113: Bird's eye view of scenario S109. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

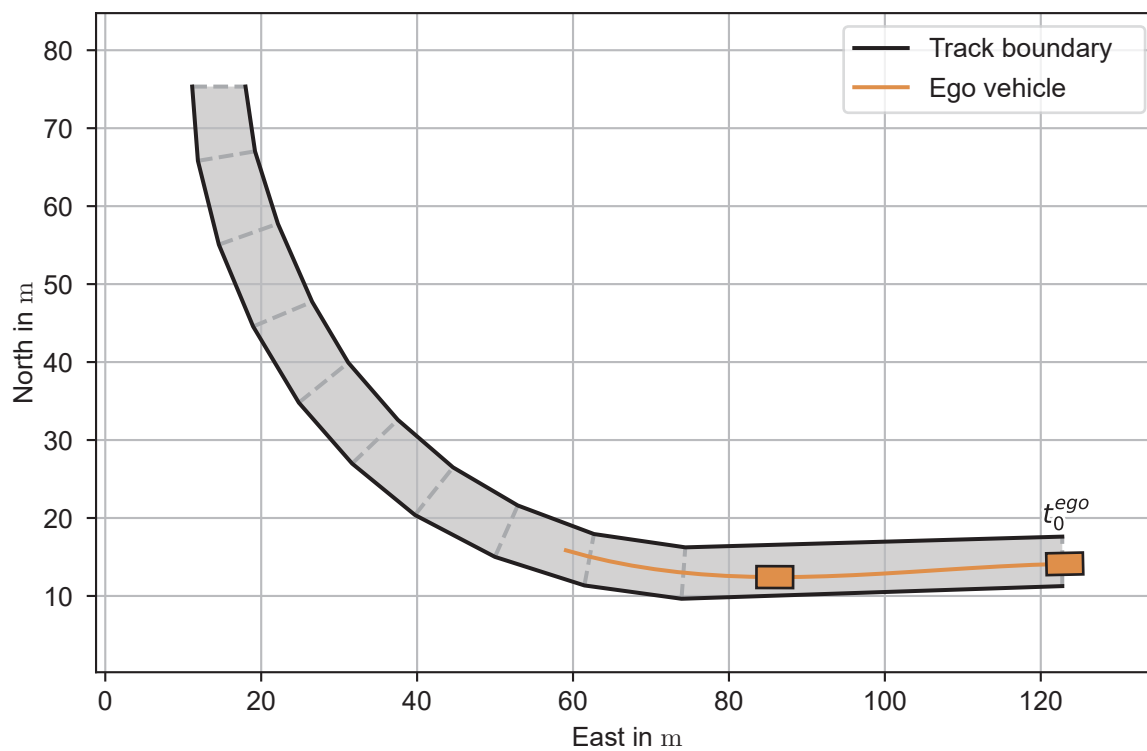


Figure A.114: Bird's eye view of scenario S110. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

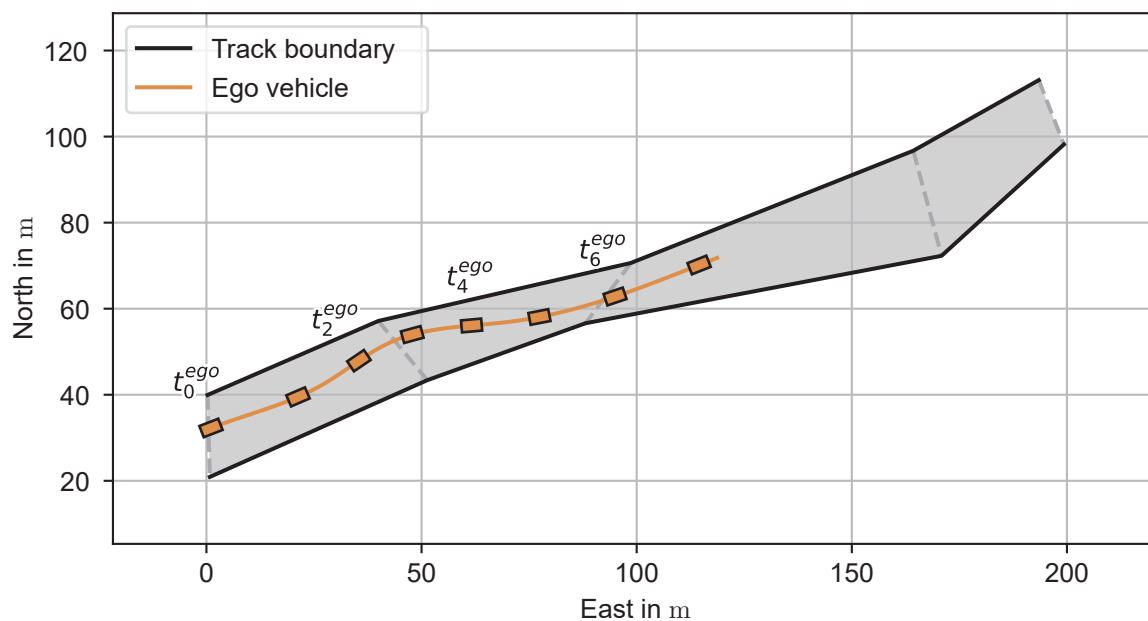


Figure A.115: Bird's eye view of scenario S111. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

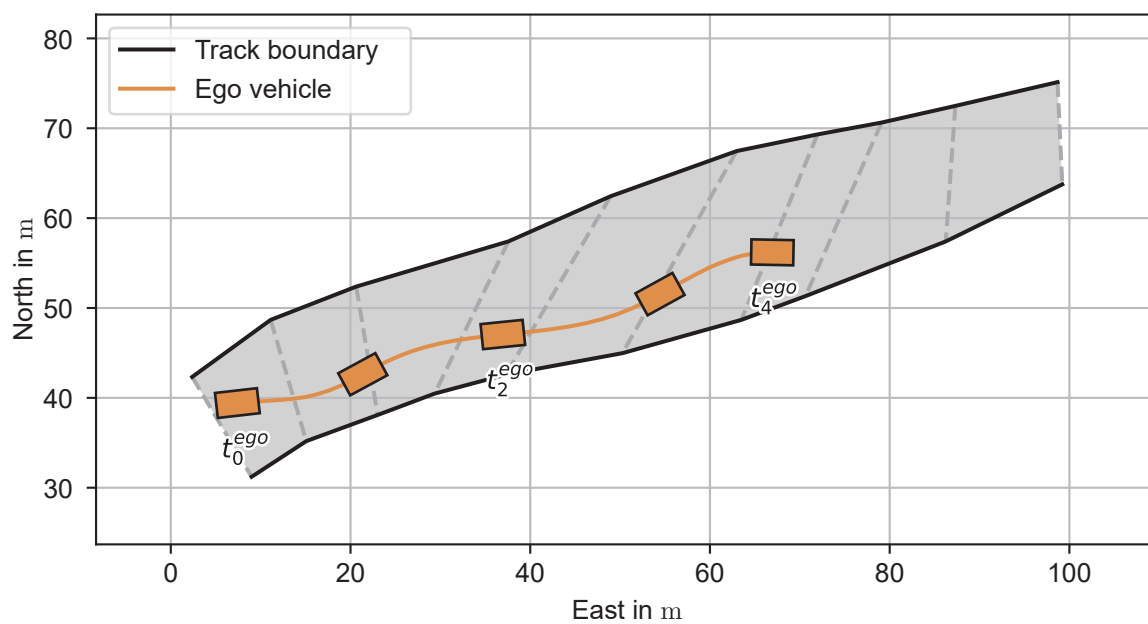


Figure A.116: Bird's eye view of scenario S112. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

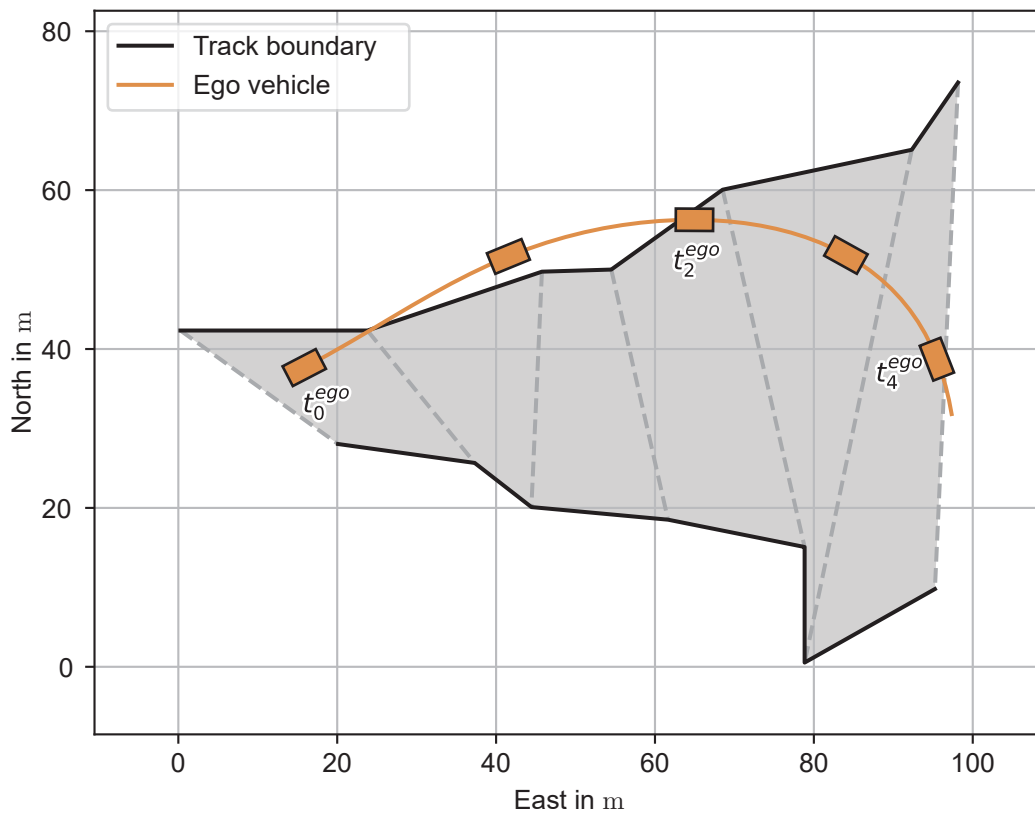


Figure A.117: Bird's eye view of scenario S113. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

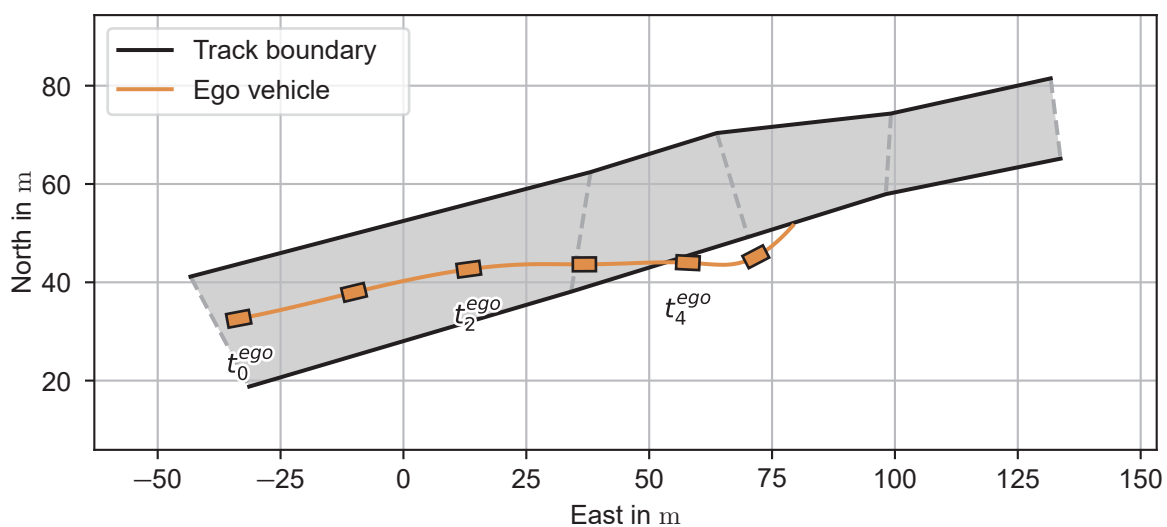


Figure A.118: Bird's eye view of scenario S114. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

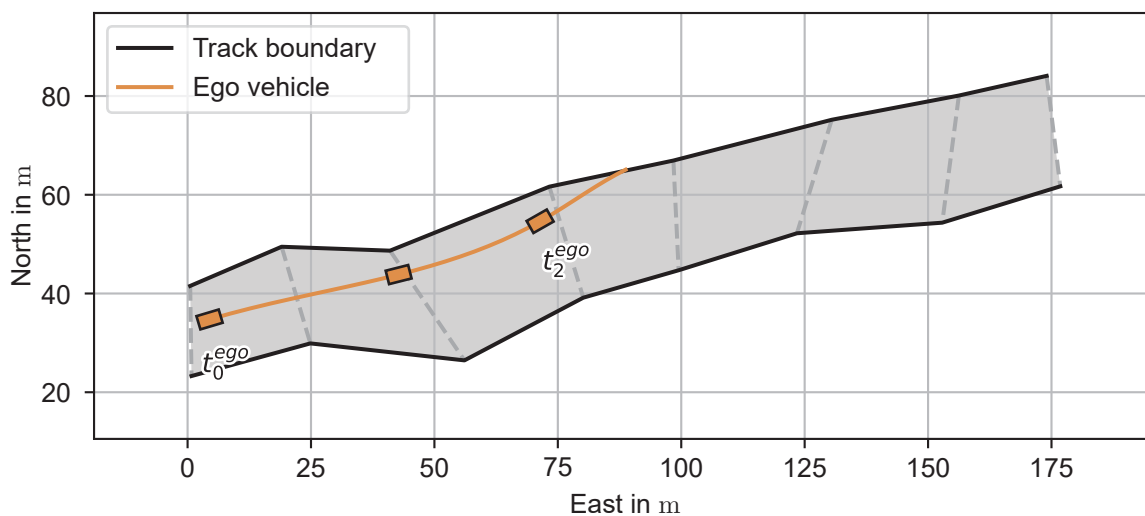


Figure A.119: Bird's eye view of scenario S115. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

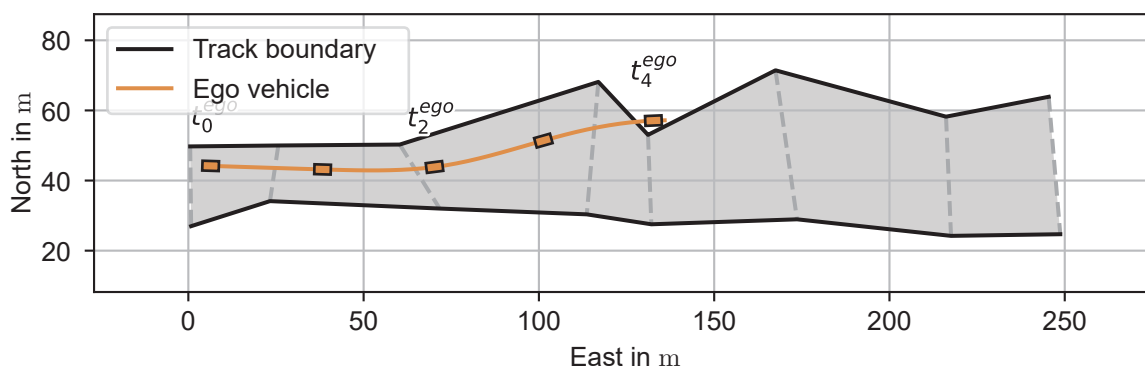


Figure A.120: Bird's eye view of scenario S116. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

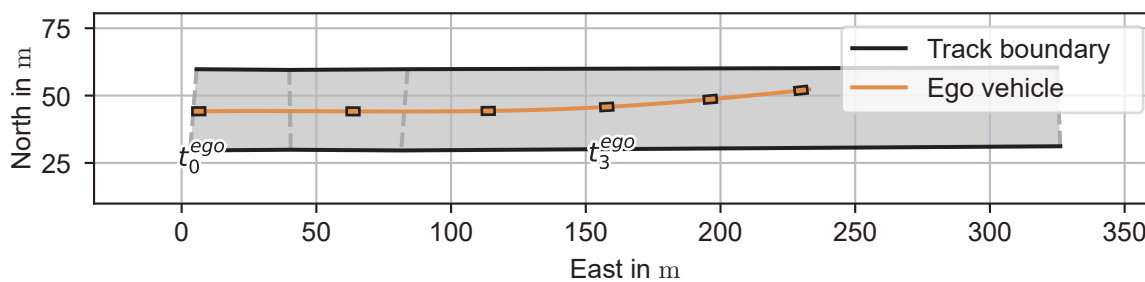


Figure A.121: Bird's eye view of scenario S117. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every third interval is marked with a text label.

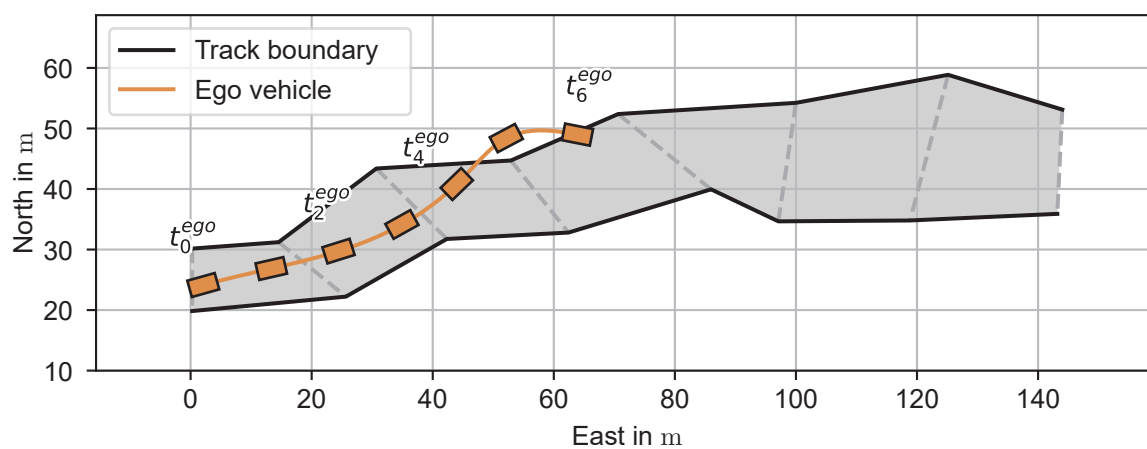


Figure A.122: Bird's eye view of scenario S118. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every second interval is marked with a text label.

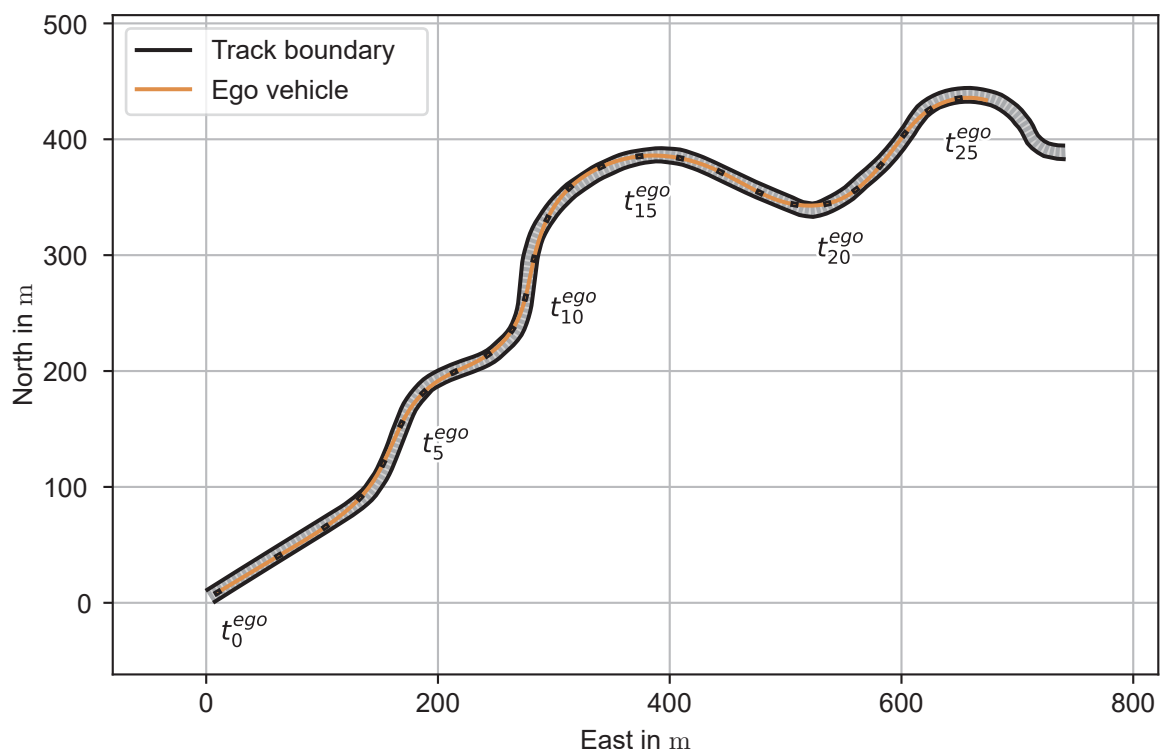


Figure A.123: Bird's eye view of scenario S119. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fifth interval is marked with a text label.

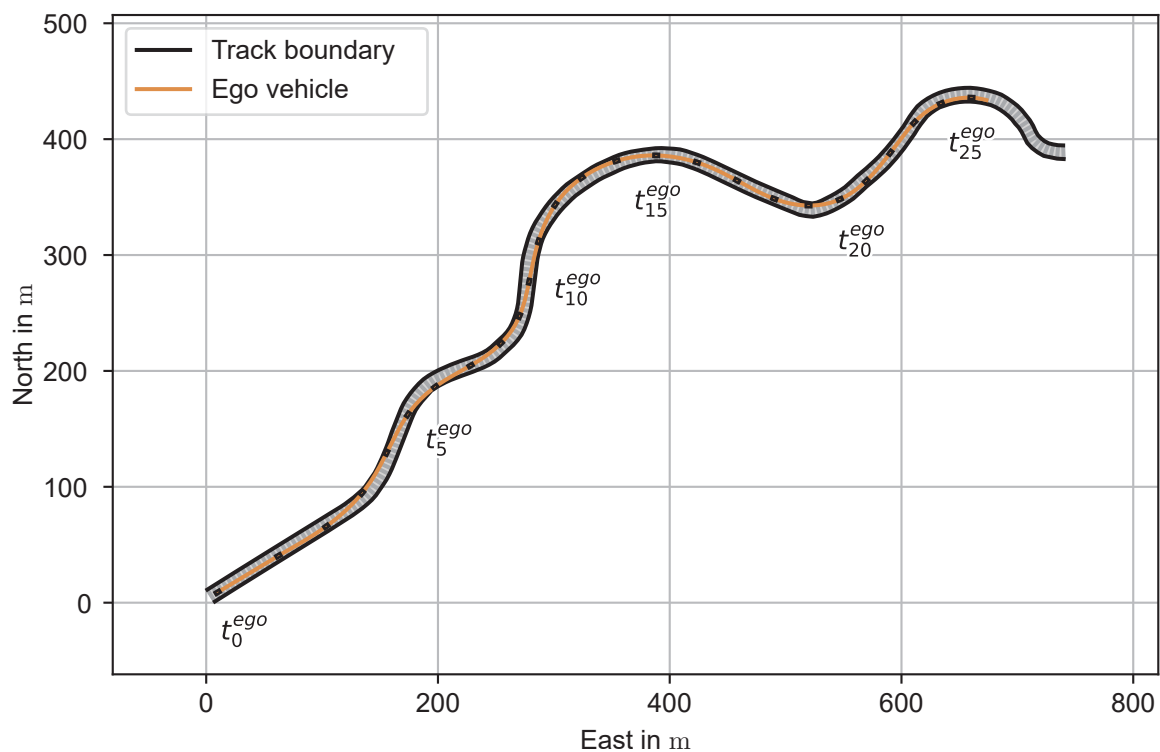


Figure A.124: Bird's eye view of scenario S120. The vehicle movements are shown via trace (line) and vehicle shape samples at 1 s interval. Every fifth interval is marked with a text label.

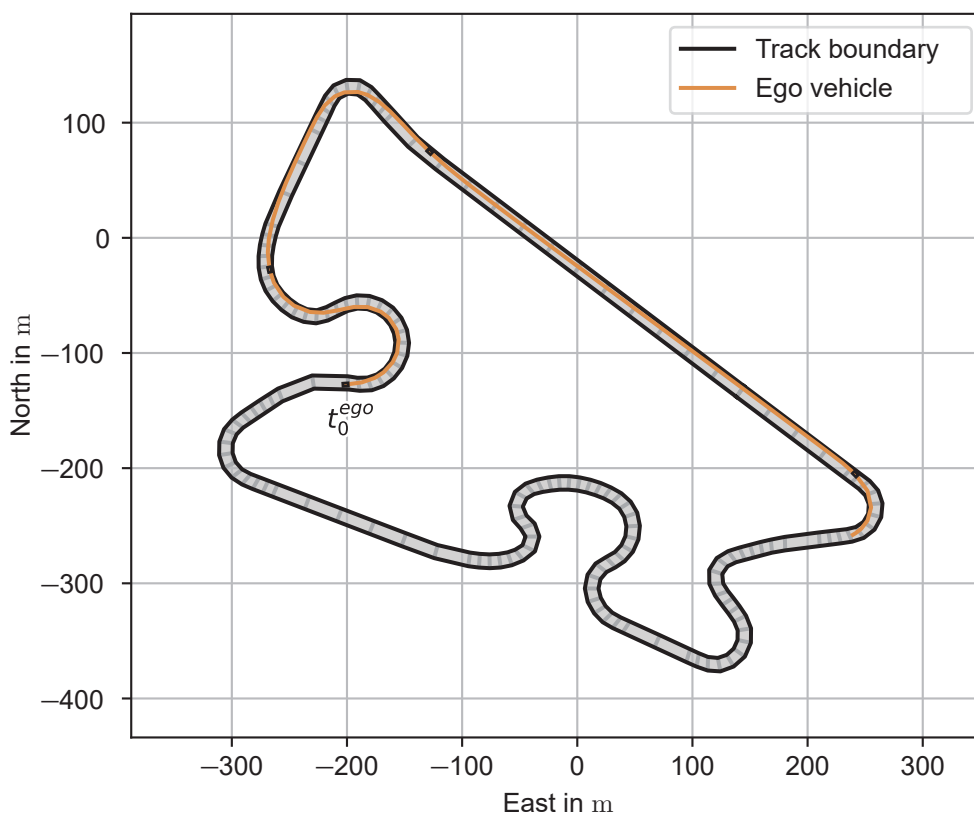


Figure A.125: Bird's eye view of scenario S121. The vehicle movements are shown via trace (line) and vehicle shape samples at 10 s interval. Every fifth interval is marked with a text label.