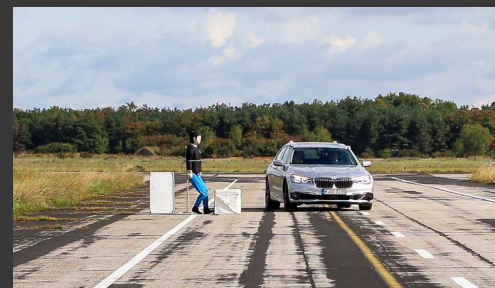
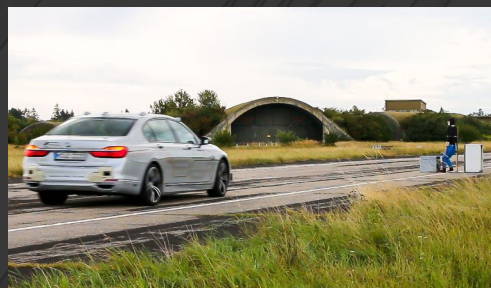
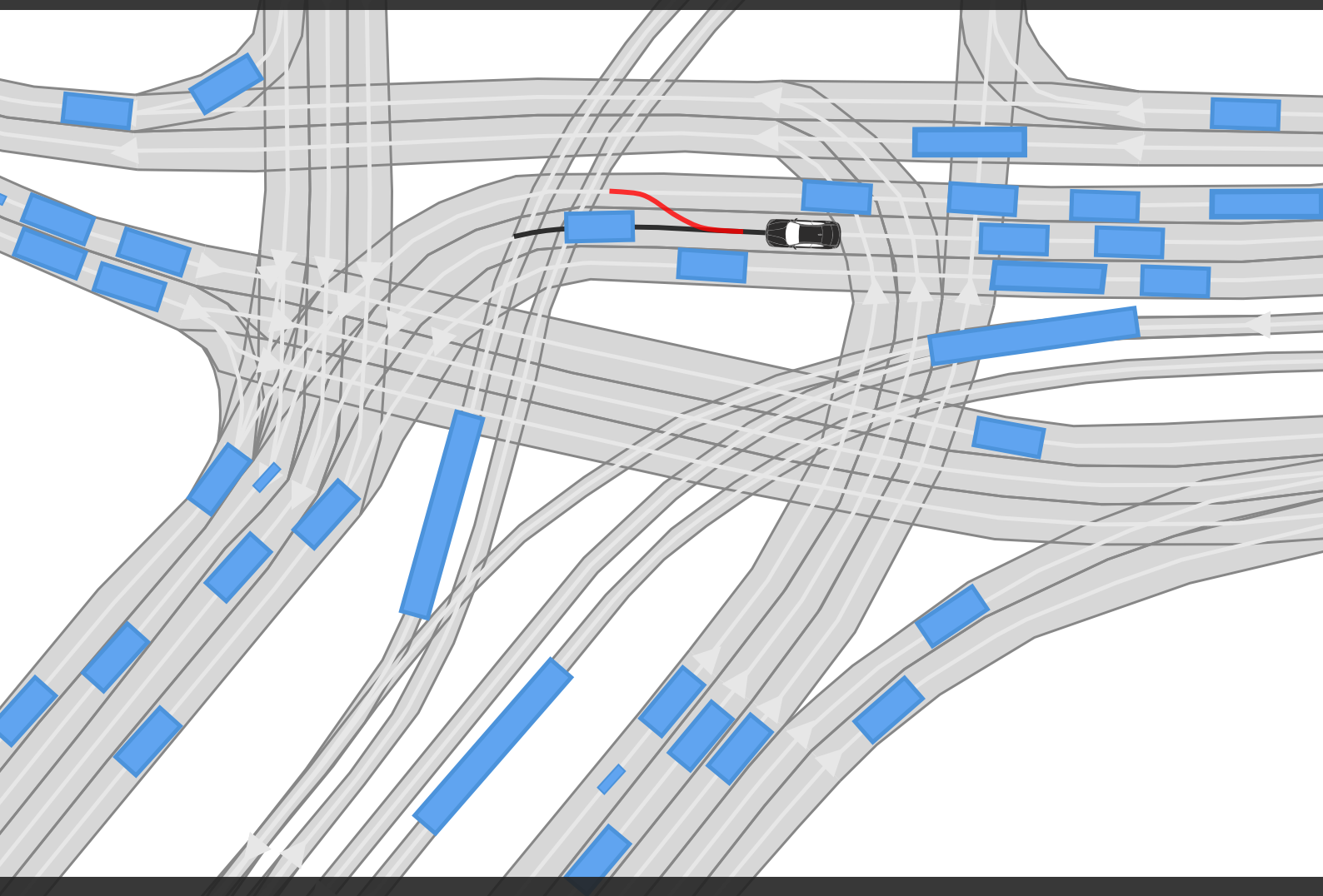


PhD Thesis

Provably Safe Motion Planning for Autonomous Vehicles Through Online Verification

Christian Pek





Provably Safe Motion Planning for Autonomous Vehicles Through Online Verification

Christian Friedrich Pek

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

Doktor der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:

Prof. Dr. Jan Křetínský

Prüfende der Dissertation:

1. Prof. Dr.-Ing. Matthias Althoff
2. Prof. Dr.-Ing. Christoph Stiller,
Karlsruher Institut für Technologie

Die Dissertation wurde am 07.01.2020 bei der Technischen Universität München
eingereicht und durch die Fakultät für Informatik am 23.07.2020 angenommen.

Foreword

*So throw off the bowlines. Sail away from the safe
harbour. Catch the trade winds in your sails.
Explore. Dream. Discover.*
—Mark Twain

This quote perfectly summarizes my motivation when I started the adventure of being a PhD student at TUM and BMW back in 2015. Now, after having spent nearly four years conducting research in safe motion planning, I know what my journey has offered to me: I had the chance to investigate challenging problems, to present and discuss my results at various international venues, and most importantly to broaden my horizon in many ways. Along my journey, I was lucky enough to never be alone. I met and worked with incredible people that challenged and supported me to finally arrive at this point.

First of all, I would like to thank my advisor Prof. Matthias Althoff who introduced me to the world of science. Without him, this thesis would not have been possible. Moreover, I am grateful for the fruitful discussions with Prof. Christoph Stiller from KIT over the years who agreed to be the second examiner of my thesis. I also like to thank Prof. Jan Křetínský for chairing my defense in the difficult time of the Corona pandemic. Many thanks also go to my supervisors at BMW, Dr. Peter Zahn and PD. Dr. Moritz Werling, who offered me a PhD position and showed great trust in my abilities. Furthermore, I would like to thank my executives at BMW, Oliver Poguntke, Simon Fürst, Peter Schiele and René Grosspietsch for their support.

This thesis has evolved from the countless discussions, funny moments, and collaborations with my dear colleagues. In particular, I am deeply grateful to Stefanie Manzinger and Markus Koschi for their friendship and giving me help whenever I needed it. Moreover, I am honored to have worked with my friends Dr. Constantin Hubmann, Jens Schulz, Sascha Steyer, Kai Stiens, and Branka Mirchevska. A big thank-you also goes to Dr. Daniel Althoff for his continuous support at all times. Furthermore, this thesis has been supported by the work of my very talented students Mona Beikirch, Christina Miller, Leo Tappe, Julia Kabalar, Anna-Katharina Rettinger, Sebastian Maierhofer, Sebastian Kaster, Maria Althaus, and Marco Both.

In addition, I would like to thank my other BMW colleagues Tobias Rehder, Dr. Sebastian Gnatzig, Dr. Benjamin Gutjahr, Julian Thomas, Julian Tatsch, Thomas Barowski, Andreas Lawitzky, Egon Ye, Dr. Dominik Sojer, Thomas Bleher, Udo Rietschel, Alexander Terres and Florian Roeder. I also like to say thank-you to my

great TUM colleagues Felix Gruber, Dr. Aaron Pereira, Dr. Alexander Lenz, Xiao Wang, Dr. Morteza Hashemi Farzaneh, Amy Bücherl, Sina Shafaei, Dr. Johannes Betz, Emec Ercelik, Bastian Schürmann, Edmond Irani Liu, Amr Alanwar, Moritz Klischat, Niklas Kochdumper, Viktor Gaßmann, Stefan Liu, Martin Riedel, Constantin Dresel, Roman Hölzl, and Paul Maroldt. I also thank Carmella Schürmann for being a friend and for the voice-overs in the supplementary videos. I am grateful for the support of Manuela Fischer, Claudia Link, and Prof. Hans-Joachim Bungartz. Needless to say, I would have been lost in bureaucracy without the heart of our chair Ute Lomp.

Besides technical issues, a large project such as a PhD thesis also comes with emotionally difficult times and countless hours in the office. I am very grateful to my parents, Bettina and Dr. Friedrich Pek, and my brother Stefan Pek for their enduring support which brought me into the position of writing this thesis, and their sympathy and support when I required it. I am deeply indebted to my partner Karin Landgraf. Without her unconditional love in good times as in bad, this thesis would not have been possible. Moreover, I would like to thank my family and friends Jutta and Klaus Holstein, Elena and Andreas Landgraf, Carina Werner, Anastasia and Sven Fork, Andrea and Roman Landgraf, Yevgen Pikus, Jannick Hilscher, Tanja Lampe, Greta Nachbar, Lukas Munteanu, Katrin Schenberger, Patrick Tieben, and Angelika Ossowicki for cheering me up when I needed it. A big thank-you also goes to Lars Luthmann for encouraging me to continue my journey in difficult times.

Last but not least, I gratefully acknowledge the financial support of this work by the BMW Group, the German Academic Exchange Service through the congress travel program, the TUM Graduate School through the internationalization grant, the project interACT within the EU Horizon 2020 programme under grant agreement No 723395, and the German Federal Ministry of Economics and Technology through the research initiative Ko-HAF. Moreover, I appreciate Google and GeoBasis-DE/BKG for allowing researchers to use their satellite images in scientific publications free of charge.

Abstract

Safe motion planning remains an unsolved challenge in the development of autonomous vehicles. This thesis introduces fail-safe motion planning as the first approach to guarantee legal safety in arbitrary traffic situations. By employing fail-safe motion planning, autonomous vehicles never cause accidents even if other traffic participants are allowed to perform any legal behavior. The proposed safety layer verifies whether intended trajectories comply with legal safety and provides fail-safe trajectories when intended trajectories result in safety-critical situations. The proposed fail-safe motion planning technique can be easily integrated into existing motion planning frameworks and can be used with arbitrary trajectory planners.

Fail-safe motion planning employs set-based predictions to handle measurement uncertainties and to predict all possible legal behaviors of other traffic participants online. Based on the computed prediction, fail-safe trajectories ensure that autonomous vehicles never enter possibly occupied spaces in the environment. In addition, fail-safe trajectories guide the vehicle to invariably safe sets that allow autonomous vehicles to remain safe at all times. The correct-by-construction safety layer is real-time capable and thus allows the fail-safe operation of autonomous vehicles.

The safety benefits are validated in over a hundred tests with a BMW 7-series test vehicle and in simulation with real-world data. Even in the most dangerous accident hotspots in urban environments, fail-safe motion planning ensures the safety of autonomous vehicles at all times. In all scenarios, the autonomous vehicle executes only safe trajectories even when using intended motion planners that actively ignore other traffic participants or machine learning to plan intended trajectories. User studies with an adaptive cruise control system suggest that the proposed safety layer provides a significantly greater feeling of safety and comfort for passengers. In addition, tests with recorded real traffic show that fail-safe motion planning does not result in overly conservative behaviors of autonomous vehicles.

Summary: Fail-safe motion planning ensures the provably safe operation of autonomous vehicles for arbitrary intended trajectories. The presented results indicate that the use of fail-safe motion planning can drastically reduce the number of traffic accidents.

Zusammenfassung

Die sichere Bewegungsplanung ist weiterhin ein ungelöstes Problem in der Entwicklung von autonomen Fahrzeugen. Die vorliegende Arbeit führt ein neuartiges und ausfallsicheres Verifikationsverfahren ein, mit deren Hilfe zum ersten Mal die verkehrsregelkonforme Sicherheit von autonomen Fahrzeugen in beliebigen Verkehrssituationen gewährleistet werden kann. Insbesondere garantiert das vorgestellte Verfahren, dass autonome Fahrzeuge niemals einen Unfall verursachen, auch wenn andere Verkehrsteilnehmer jede mögliche und legale Bewegung ausführen dürfen. Das Verifikationsverfahren überprüft, ob geplante Trajektorien des Fahrzeuges sicher sind und generiert Rückfalltrajektorien falls diese zu einer unsicheren Situation führen. Das Verfahren kann leicht in bestehende Bewegungsplanungskomponenten integriert werden und sichert beliebig geplante Trajektorien ab.

Die vorliegende Arbeit verwendet mengenbasierte Prädiktionen, um Messunsicherheiten sowie alle legalen Bewegungen anderer Verkehrsteilnehmer zu berechnen. Rückfalltrajektorien garantieren, basierend auf den Prädiktionen, dass das autonome Fahrzeug niemals mit anderen Verkehrsteilnehmern kollidiert. Weiterhin enden die Rückfallbewegungen in invariabel sicheren Zustandsmengen, sodass die Sicherheit des Fahrzeugs auch über einen unendlichen langen Zeithorizont garantiert werden kann. Das mathematisch korrekte Verfahren ist echtzeitfähig und erlaubt den ausfallsicheren Betrieb von autonomen Fahrzeugen.

Die Sicherheitsvorteile wurden in über hundert Versuchen mit einem BMW 7er Versuchsfahrzeug validiert. Das vorgestellte Verfahren garantiert die Sicherheit auch in den kritischsten Situationen im städtischen Verkehr. In allen Szenarien hat das autonome Fahrzeug nur beweisbar sichere Trajektorien ausgeführt auch wenn die geplanten Trajektorien keine anderen Verkehrsteilnehmer berücksichtigen oder von maschinellem Lernen geplant wurden. Benutzerstudien deuten zudem darauf hin, dass die Verwendung des vorgestellten Verfahrens in einem höheren Sicherheitsgefühl und Komfort für Passagiere resultiert. Weitere Versuche in dichtem Stadtverkehr haben gezeigt, dass das Verifikationsverfahren nicht zu einem konservativen Verhalten des Fahrzeugs führt.

Kurzdarstellung: Das entwickelte Verifikationsverfahren garantiert den sicheren Betrieb von autonomen Fahrzeugen für beliebig geplante Trajektorien. Die vorliegenden Ergebnisse zeigen, dass die Verwendung des Verfahrens zu einer deutlichen Reduktion von Verkehrsunfällen führt.

Contents

Abstract	v
Zusammenfassung	vii
List of Figures	xiii
List of Tables	xvii
List of Algorithms	xix
List of Symbols	xxi
1 Introduction	1
1.1 Safety Assessment of Autonomous Vehicles	3
1.1.1 Non-formal methods	4
1.1.2 Formal verification methods	5
1.2 Contributions to Provably Safe Motion Planning	7
1.3 Outline of the Thesis	10
2 Notation and Preliminaries	13
2.1 Mathematical Notation	13
2.2 CommonRoad Benchmark Suite	15
2.3 Reachability Analysis of Dynamical Systems	16
2.4 Set-Based Prediction of Other Traffic Participants	18
2.5 Convex Optimization	21
3 Computationally Efficient Fail-Safe Trajectory Planning	25
3.1 Introduction and State of the Art	25
3.1.1 Discrete trajectory planning techniques	27
3.1.2 Continuous trajectory planning techniques	28
3.2 Real-Time Trajectory Planning Using Convex Optimization	30
3.2.1 Planning longitudinal motions	31
3.2.2 Planning lateral motions	32
3.2.3 Enhancing passenger comfort through slack variables	34
3.3 Fail-Safe Trajectory Planning in Arbitrary Traffic Scenarios	35
3.4 Exploration of Non-convex Search Spaces for Fail-Safe Solutions	40
3.4.1 Enumerating possible driving corridors	41
3.4.2 Computing the drivable area of autonomous vehicles	43

3.4.3	Determining driving corridors using the drivable area	44
3.5	Numerical Experiments	50
3.5.1	Cut-in vehicles on highways	50
3.5.2	Urban T-junction	52
3.5.3	Avoiding collisions with crossing pedestrians	55
3.5.4	Comparison with discrete planning approaches	57
3.5.5	Fail-safe planning with driving corridors	58
3.5.6	Managing complex scenarios with small solution spaces	58
3.6	Summary	60
4	Invariably Safe Sets for Infinite Time Horizon Planning	63
4.1	Introduction and State of the Art	63
4.1.1	Inevitable collision states	65
4.1.2	Control invariant sets	66
4.2	Invariably Safe States	67
4.3	Under-Approximation of Invariably Safe Sets	69
4.3.1	Environment representation	72
4.3.2	Algorithmic steps	73
4.3.3	Computational complexity	76
4.4	Exploiting Invariably Safe Sets for Motion Planning	77
4.5	Integration of Invariably Safe Sets into Linear-Quadratic Programs	78
4.5.1	Linear safe distance constraints	78
4.5.2	Linear evasive distance constraints	80
4.6	Numerical Experiments	83
4.6.1	Verifying intended trajectories for infinite horizons	83
4.6.2	Evaluating the tightness of the under-approximation	85
4.6.3	Urban T-junction	85
4.6.4	Determining the existence of fail-safe trajectories	87
4.6.5	Safety assessment of machine learning approaches	88
4.7	Summary	93
5	Online Safety Verification of Arbitrary Motions	95
5.1	Introduction to Motion Planning Frameworks	95
5.2	Integration in Motion Planning Frameworks	96
5.3	Details of the Verification Technique	97
5.4	Computation Steps of the Verification Procedure	101
5.5	Summary	104
6	Experiments with Test Vehicles and Driving Simulators	105
6.1	Introduction to the Vehicle Setup	106
6.2	Driving Experiments	107
6.2.1	Verifying randomly generated trajectories	108
6.2.2	Verifying planned motions in dynamic environments	113
6.2.3	Avoiding collisions with vulnerable road users	117

6.2.4	Summary of driving experiments	121
6.3	Fail-Safe Trajectories in Complex Urban Traffic Scenarios	122
6.3.1	Left turn at an urban intersection	122
6.3.2	Lane changes in dense urban traffic	124
6.3.3	Jaywalking pedestrians	126
6.3.4	Verification of arbitrary intended motions	128
6.3.5	Summary of experiments with urban traffic scenarios	130
6.4	Assessment of Intervention Rates and Passenger Comfort	130
6.4.1	Adaptive cruise control user study	131
6.4.2	Intervention assessment in dense urban traffic	134
6.4.3	Summary of conducted studies	140
6.5	Summary	140
7	Conclusions and Perspectives	143
7.1	Summary of Contributions	143
7.2	Impacts of Fail-Safe Motion Planning	146
7.2.1	Certification	146
7.2.2	Merits of self-verifying robots	147
7.2.3	Toward safe human-robot coexistence	147
7.3	Perspectives	148
7.3.1	ASIL-D compliant safety layer	148
7.3.2	Ensuring drivability despite disturbances	149
7.3.3	Further improving the verification performance	151
7.4	Closing Remarks	152
8	Publications	153
	Bibliography	157
A	Appendix	181
A.1	Vehicle Shape Approximation	181
A.2	Random Planner	181
A.3	Parameters of the Fail-Safe Planning Experiments	182
A.3.1	Cut-in vehicles on highway	182
A.3.2	Urban T-junction	182
A.3.3	Intersection with crossing pedestrian	183
A.3.4	Distinct driving corridors	183
A.4	Parameters of the Invariably Safe Set Experiments	184
A.4.1	Verification of trajectories for infinite time horizons	184
A.4.2	Invariably safe set for urban T-junction	184
A.4.3	Existence of fail-safe trajectories	184
A.5	Parameters of the Driving Experiments	185
A.5.1	Experiments with static obstacle	185
A.5.2	Experiments with simulated vehicles	186

Contents

A.5.3	Experiments with simulated pedestrians	187
A.6	Post-processing Urban Traffic Situations	188
A.6.1	Post-processing steps	188
A.6.2	Parameters for scenarios	189
A.6.3	Parameterization of planners	191
A.6.4	Detailed planning cycle of intersection scenario	192
A.6.5	Detailed planning cycle of the lane change scenario	193
A.6.6	Detailed planning cycle of the jaywalking pedestrian scenario	194
A.7	User Study in Driving Simulator	195
A.7.1	Overview of the safety-critical scenarios	195
A.7.2	Additional results of the simulations	197
A.8	Intervention Assessment Experiments	199
A.8.1	Used parameters	199
A.8.2	Additional results	200
A.9	Summary of Supplementary Material	201
A.9.1	Video files	201
A.9.2	CommonRoad scenarios	203

List of Figures

1.1	Driving situations.	1
1.2	A variety of real-world scenarios	2
1.3	Motivation for safety assessment	3
1.4	Overview of safety assessment approaches	4
1.5	Safety assessment using simulation	5
1.6	Formal verification using reachable sets	7
1.7	Proposed online verification approach	8
2.1	Curvilinear coordinate system	14
2.2	Overview of CommonRoad	15
2.3	Comparison between exact and over-approximative reachable sets	17
2.4	Example of set-based predictions	20
2.5	Collision-free input trajectory	21
2.6	Examples of convex and non-convex sets	22
2.7	Examples of convex and non-convex functions	23
3.1	Fail-safe trajectory concept	26
3.2	Examples of discrete and continuous planning techniques	29
3.3	Linearized kinematic model for planning	32
3.4	Slack variables for comfortable braking profiles	35
3.5	Two-stage cost increase for slack variables	35
3.6	General procedure to compute fail-safe trajectories	36
3.7	Computation of longitudinal collision constraints	37
3.8	Illustration of the GTTC	38
3.9	Illustration of lateral collision constraints	40
3.10	Non-convex of search spaces	41
3.11	Lateral constraints and passing sides	42
3.12	Visualization of the drivable area	44
3.13	Reachability graph	44
3.14	Fail-safe planning with driving corridors	45
3.15	Identification of driving corridors	47
3.16	Lateral constraints from driving corridors	49
3.17	Highway scenario with cut-in vehicle	50
3.18	Planned fail-safe trajectory of the highway scenario	51
3.19	Urban T-junction scenario	53
3.20	Planned fail-safe trajectory of the urban T-junction scenario	54
3.21	Urban scenario with crossing pedestrian	55

List of Figures

3.22	Planned fail-safe trajectory of the pedestrian scenario	56
3.23	Scenario with distinct driving corridors	59
3.24	Scenario with a small solution space	60
4.1	Safety problem of finite planning horizons	64
4.2	Illustration of ICS and CIS	66
4.3	Subdivision of the configuration space	68
4.4	Illustration of backward computation of invariably safe sets	70
4.5	Under-approximation of invariably safe sets	71
4.6	Illustration of sections	73
4.7	Illustration of safe and evasive distances	75
4.8	Safety properties of trajectories	78
4.9	Piecewise linear approximation of safe distances	79
4.10	Lateral constraint for evasive distances	82
4.11	Urban scenario for verification	84
4.12	Feasible velocity profile	84
4.13	Computed invariably safe sets	86
4.14	Invariably safe set for T-junction	87
4.15	Invariably safe sets in emergency situations	89
4.16	Overview of the safe RL approach	90
4.17	Verification of lane change trajectories	90
4.18	Simulation results without and with the safety layer	92
5.1	Sense-plan-act architecture	95
5.2	Proposed online safety framework	96
5.3	Visualization of the verification approach	98
5.4	Computation steps during the verification	102
6.1	BMW 7-series test vehicle	105
6.2	Illustration of the environment model generation	106
6.3	Foam obstacles for experiments	107
6.4	Braking maneuver to avoid collisions with a static obstacle	110
6.5	Evasive maneuver to avoid collisions with a static obstacle	111
6.6	Invariably safe set of the scenario in Fig. 6.4	112
6.7	Invariably safe set of the scenario in Fig. 6.5	112
6.8	Avoiding collisions with a cut-in vehicle by braking	114
6.9	Avoiding collisions with a cut-in vehicle by swerving	115
6.10	Invariably safe set of the scenario in Fig. 6.8	116
6.11	Invariably safe set of the scenario in Fig. 6.9	116
6.12	Evading stopped pedestrians	118
6.13	Invariably safe set of the scenario in Fig. 6.12	119
6.14	Evading pedestrians by swerving into an adjacent lane	120
6.15	Invariably safe set of the scenario in Fig. 6.14	121
6.16	Left turn at urban intersection	123

6.17	Lane change in dense urban traffic	125
6.18	Jaywalking pedestrian scenario	127
6.19	Verification of three different intended trajectory planners	129
6.20	BMW driving simulator	131
6.21	Front view of a user sitting in the driving simulator	132
6.22	Route of the intervention assessment study	135
6.23	Examples of true negatives	138
6.24	Examples of false positives	139
7.1	Code verification for ASIL-D compliance	149
7.2	Fitting fail-safe trajectories with motion primitives	150
A.1	Computation times for the presented scenarios	188
A.2	Detailed verification results of urban intersection scenario	192
A.3	Detailed verification results of lane change scenario	193
A.4	Detailed verification results of jaywalking pedestrian scenario	194
A.5	Test scenarios for the comfort evaluation	196
A.6	Additional simulation results I	197
A.7	Additional simulation results II	198
A.8	Additional examples of true negatives	200

List of Tables

2.1	Motion assumptions for cars, trucks, motorbikes, and bicyclists . . .	19
2.2	Motion assumptions for pedestrians	19
4.1	Performance of the RL agent	92
6.1	General parameters of the driving experiments.	108
6.2	Confusion matrix of fail-safe motion planning	131
6.3	Results of the Wilcoxon signed-rank t-test	133
6.4	Analysis results of alleged fail-safe trajectory executions	137
A.1	Parameters of the highway scenario in Ch. 3	182
A.2	Parameters of the urban T-junction scenario in Ch. 3	182
A.3	Parameters of the pedestrian scenario in Ch. 3	183
A.4	Parameters of the driving corridor scenario in Ch. 3	183
A.5	Parameters of the urban scenario in Ch. 4	184
A.6	Parameters of the T-junction scenario in Ch. 4	184
A.7	Parameters of the cut-in scenario in Ch. 4	184
A.8	Parameters of the scenario in Fig. 6.4	185
A.9	Parameters of the scenario in Fig. 6.5	185
A.10	Parameters of the scenario in Fig. 6.8	186
A.11	Parameters of the scenario in Fig. 6.9	186
A.12	Parameters of the scenario in Fig. 6.12	187
A.13	Parameters of the scenario in Fig. 6.14	187
A.14	Parameters of the verification cycle	189
A.15	Parameters of the set-based prediction	189
A.16	Parameters of the fail-safe planner	190
A.17	Parameters of the most likely prediction	190
A.18	Parameters of the intended planners 1 & 2	191
A.19	Parameters of the intended planner 3	191
A.20	Parameters of SPOT in the intervention study	199
A.21	Parameters of the fail-safe planner in the intervention study	199

List of Algorithms

1	Identification of driving corridors	46
2	Under-approximation of invariably safe sets	74
3	Computation of safe distance constraints	81
4	Computation of evasive distance constraints	82
5	Online verification during the operation of the autonomous vehicle .	99
6	Computation steps to verify arbitrary trajectories	103

List of Symbols

Variables	
Notation	Description
a	Acceleration
Λ	Assumption on legal motion of obstacles
A	State matrix
B	Input matrix
C	Output matrix
d	Lateral position w.r.t. a curvilinear coordinate system
δ	Reaction time
ϵ	Arbitrary small step size
E	Disturbance matrix
\mathfrak{F}	Fail-safe trajectory
Γ	Reference path
γ	Passing side for fail-safe planning
t_{GTTC}	Guaranteed time-to-collision
\mathcal{G}	Reachability graph
\bar{h}	Node in reachability tree
\mathcal{J}	Intended trajectory
$\mathcal{J}^{\text{safe}}$	Safe part of an intended trajectory
j	Jerk (time derivation of acceleration)
κ	Curvature
ℓ	Distance between rear and front axle of ego vehicle
p	Position in Euclidean space
Φ	Feedback control law
ϕ_{ref}	Reference for feedback control law
r	Radius of circle in vehicle shape approximation
R_{lon}	Smallest circle covering the shape of the ego vehicle
s	Lon. position w.r.t. a curvilinear coordinate system
ς	Slack variable
t	Time
t_0	Initial time
τ	Point in time of known invariably safe set
$T(k)$	k -th time interval for invariably safe set computation
ϱ	Tangential vector
θ	Orientation

List of Symbols

t_{TTR}	Time-to-react
\mathcal{T}_{lon}	Longitudinal reachability tree
\mathcal{T}_{lat}	Lateral reachability tree
u	Input
$u([t_0, t_h])$	Input trajectory for time interval $[t_0, t_h]$
v	Velocity
w	Weight in cost function
x	State
x_0	Initial state
$x([t_0, t_h])$	State trajectory for time interval $[t_0, t_h]$
$x^{(i)}$	i -th component of state x
Ξ	Driving corridor
z	Disturbance

Sets

Notation	Description
\mathcal{B}	Set of indices describing safety-relevant obstacles
\mathfrak{B}_k^i	i -th base set at time step t_k
$\mathfrak{B}_k^{\text{Pi}}$	i -th propagated base at time step t_k
$\mathfrak{B}_{k,n}^{\text{CR}}$	Set of connected base sets at time step t_k
\mathcal{C}	Constraint set
\mathcal{D}	Drivable area
\mathcal{E}	Lane-based environment of the autonomous vehicle
\mathcal{E}_{al}	Allowed lanes of the ego vehicle
\mathcal{E}_{b_i, b_j}	Safe set section between obstacle b_i and b_j
\mathcal{F}	Maximal set of collision-free states
$\mathcal{I}^{(i)}$	Set of lateral deviation intervals
$\mathcal{I}_q^{(i)}$	Lateral deviation interval for circle i
\mathbb{N}	Set of natural numbers (incouding zero)
\mathbb{N}_+	Set of positive natural numbers
\mathcal{O}	Occupancy set
Ω	Allowed area for standstill behind obstacle
\mathcal{P}	Convex polytope set
\mathbb{R}	Set of real numbers
\mathbb{R}_+	Set of real positive numbers
\mathcal{R}	Forward reachable set
$\bar{\mathcal{R}}$	Over-approximative forward reachable set
$\tilde{\mathcal{R}}$	Collision-free backward reachable set
\mathcal{S}	Invariably safe set
\mathcal{S}^1	Safe distance under-approximation of \mathcal{S}
\mathcal{S}^2	Evasive distance under-approximation of \mathcal{S}

\mathcal{U}	Set of admissible inputs
\mathcal{X}	Set of possible states
\mathcal{X}_0	Initial set of states
\mathcal{Z}	Disturbance set

Functions and operators

Notation	Description
dom	Domain of a function
f	Differential equation of system dynamics
g	Linear function
occ	Relates state vector to set of points in environment
J	Cost function
\oplus	Minkowski sum of sets
\mathcal{P}	Power set
\cap	Intersection of sets
\cup	Union of sets
Υ	Transf. from global to curvilinear coordinate system
\setminus	Set difference
χ	Solution of system dynamics

1 Introduction

Safe motion planning is still a major challenge regarding autonomous driving. Autonomous vehicles will undoubtedly become essential mass-deployed robotic systems in our everyday lives, and the safety they provide will be an important factor for their success. These systems have to perform various complex driving tasks in highly uncertain environments without human intervention. For instance, they must be able to safely drive on highways, accomplish valet parking, or maneuver in dense urban traffic [30–34] (cf. Fig. 1.1). However, their full potential will never be exploited if the safety of passengers and other traffic participants cannot be ensured at all times.

Unsafe decisions of autonomous vehicles can cause severe personal injuries and tremendous economic loss in terms of physical damage and product liability. Recent accidents of autonomous driving systems on public roads have raised major concerns among various institutions [35–37], and policy makers continue to debate the adequate safety levels of autonomous vehicles needed to allow them to transport passengers on public roads [35]. To achieve widespread societal acceptance, safety concerns must be resolved to the full satisfaction of all road users.

In this thesis, we develop a verification technique to ensure that autonomous vehicles do not cause accidents. It cannot be excluded that autonomous vehicles may be involved in accidents, for instance, when a following car deliberately provokes a rear-end collision, but self-inflicted accidents can and should be eliminated. Thus, the proposed techniques encourage a paradigm-shift from accepting residual collision risks to ensuring legal safety. In particular, planned motions (also called

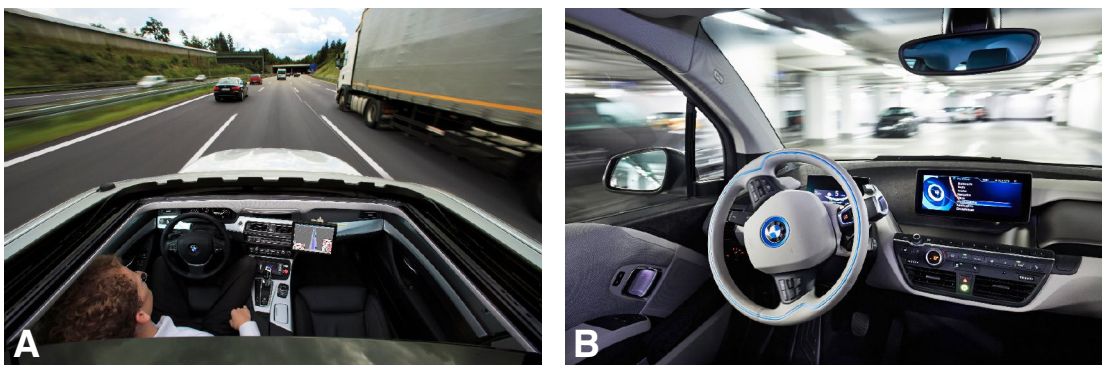


Figure 1.1: Driving situations. (A) Top view of a drive in an autonomous test vehicle on a German highway and (B) valet parking of an autonomous vehicle inside a parking lot. ©BMW AG.

1 Introduction



Figure 1.2: A variety of real-world scenarios. Autonomous vehicles have to properly react to numerous scenarios with varying complexity in the real world. The figure shows excerpts from the recorded data of a single drive with our test vehicle, denoted as AV. (A) The AV has to merge onto a highway with dense traffic. (B) The truck on the right suddenly changes to the AV’s lane. (C) The AV is driving on a highway with dense traffic. (D) A bicycle unexpectedly crosses the path of the AV. (E) Multiple bicycles occupy the AV’s lane. (F) A pedestrian is jaywalking. (G) The AV is driving in dense urban traffic with trams and motorcycles. (H) The field of view of the AV is occluded by a truck on the right side. (I) The AV has to navigate through dense urban traffic.

trajectories in this thesis) of autonomous vehicles must be provably collision-free under the premise that other traffic participants in the environment are allowed to perform any legal behavior in accordance with traffic rules [38, 39].

So far, verification in the automotive industry has mainly relied on testing the vehicle in a multitude of scenarios, aiming at estimating the residual risks. However, testing alone cannot ensure strict levels of safety due to the infinite number of unique real-world scenarios that autonomous vehicles may encounter [40–43]. Fig. 1.2 shows very different scenarios recorded in the area of Munich during one single afternoon; all of them pose distinct difficulties for autonomous vehicles. Even if autonomous vehicles operate with a residual collision risk of 0.01% per kilometer, this can imply one collision per 10.000 kilometers. In addition, just proving that autonomous vehicles are as reliable as human drivers with respect to caused fatalities (with 95% confidence) requires 275 million test kilometers without collisions in real traffic [42]. To put this number into perspective, a fleet of 100 autonomous

vehicles would need to drive for 12.5 years, 24 hours per day, without any failure. This evaluation has to be re-performed every time the vehicle's software is changed.

1.1 Safety Assessment of Autonomous Vehicles

The goal of safety assessment for motion planning of autonomous vehicles is to determine whether an arbitrary motion is safe or potentially unsafe. Since we aim to exclude that autonomous vehicles cause accidents, this thesis focuses on ensuring legal safety. Thus, we define the safety of a motion plan as:

Definition 1 (Safety of Motion Plans) *A motion plan of the autonomous vehicle is called safe if it is provably collision-free with any legal behavior of other traffic participants in the environment.*

For collision checking, we consider the occupancy of the autonomous vehicle (its occupied space in the environment) throughout the motion plan.

Fig. 1.3 shows a typical traffic situation in which the controlled autonomous vehicle, denoted as ego vehicle in the following, plans a change to the left adjacent lane. For this lane change maneuver, the motion planner of the ego vehicle needs to consider the future motion of other traffic participants to plan collision-free motions. However, surrounding traffic participants may perform any legal behavior, making it difficult to decide whether planned motions are safe or not. In the following, we briefly review common techniques to assess the safety of planned motions for robotic systems [44]. The presented approaches can be clustered into non-formal methods, which cannot exclude the possibility of collisions, and formal verification methods, which are able to guarantee the safety of planned motions (cf. overview in Fig. 1.4).

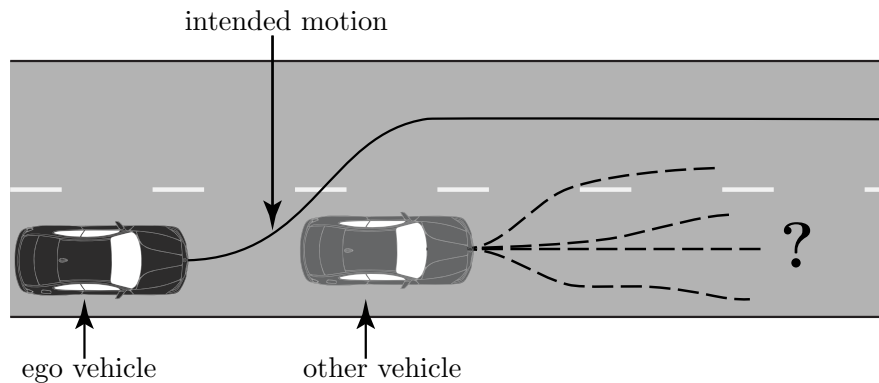


Figure 1.3: Motivation for safety assessment. The ego vehicle needs to ensure that its intended motion is safe with regards to the future motion of the other vehicle. However, this task is difficult since the future behavior of other traffic participants is not usually known.

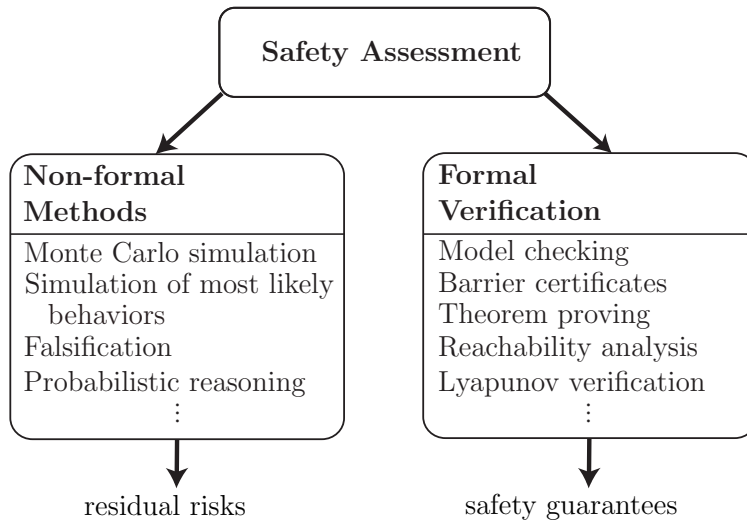


Figure 1.4: Overview of safety assessment approaches. Non-formal methods cannot exclude the possibility of the ego vehicle entering unsafe states along its intended motion. In contrast, formal verification provides guarantees that the intended motion is safe.

1.1.1 Non-formal methods

Simulation-based approaches simulate feasible future evolutions of a traffic scenario to determine possible collisions along the planned motion. For instance, planned motions of the ego vehicle are checked for collisions with the predicted most likely motions of other traffic participants [45,46] (cf. Fig. 1.5). The majority of prediction approaches can only compute a limited set of behaviors online for computational efficiency. For instance, obstacles’ most likely behaviors are computed by applying probabilistic methods [47–53] or machine learning methods [54–59]. However, these simulation techniques can only ensure the safety of planned motions if other traffic participants do not deviate from the predicted behavior [60–65]; yet, such deviations will often occur in real traffic.

Alternatively, probabilistic reasoning approaches estimate the probability of collisions for given planned motions. These approaches consider stochastic motion models of other traffic participants [66–70]. The computed probabilities are used to select the motion plan with the lowest probability of collisions. Monte Carlo simulation is a particularly popular approach for highly complex scenarios [71–73]. Monte Carlo approaches randomly create motion predictions of other traffic participants according to some probability distribution, and they subsequently simulate the generated scenarios to assess the probability of collisions. However, even a small residual risk may result in a collision, harming passengers or other traffic participants. In addition, simulations have the significant disadvantage that they may miss the testing of certain scenarios that would inevitably lead to unsafe situations.

On the other hand, falsification approaches try to disprove safety by determining counter-examples. For instance, these approaches provide safety-critical scenarios,

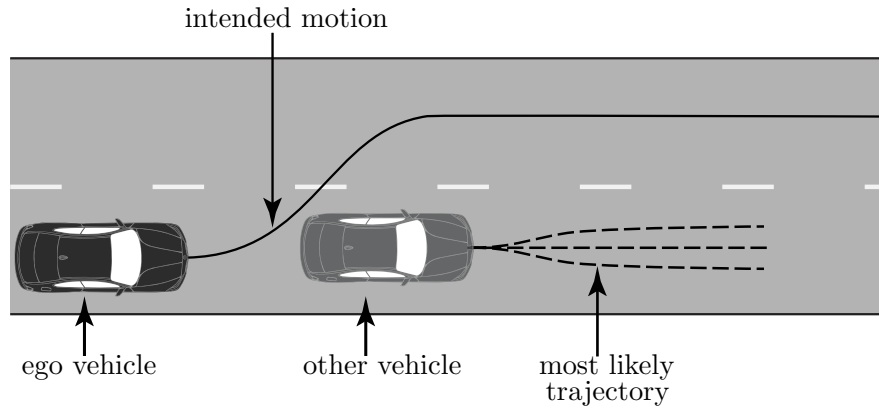


Figure 1.5: Safety assessment using simulation. The safety of intended motions is assessed by a forward simulation of the ego vehicle and other traffic participants along their trajectories. The motion of other traffic participants may correspond to their most likely trajectories, obtained using prediction approaches. The intended motion is collision-free if the ego vehicle’s occupancy along its trajectory does not intersect with any other traffic participant’s occupancy.

which demonstrate that the planned motion is unsafe [74–78]. The authors of [79] propose a systematic approach to test collision avoidance systems by primarily simulating scenarios in which leading vehicles suddenly perform emergency braking maneuvers. More sophisticated methods to automatically generate safety-critical scenarios are presented in [80–84]. These approaches use reachability analysis, neural networks, or performance metrics to synthesize scenarios. However, even if falsification approaches cannot compute a counter-example for a given motion plan, this plan is not necessarily safe since a counter-example may not have been found yet.

1.1.2 Formal verification methods

In contrast to non-formal approaches, formal verification approaches are able to provide safety guarantees. Formal verification describes the process of proving the correctness of a system with respect to a given formal specification or property in a mathematically sound way [85, 86]. If the system has been formally verified, it is guaranteed to meet the given specification. However, the process of determining (and formalizing) a desired specification is not trivial and may take a considerable amount of time, for instance through validation experiments of the system in real-world environments [87, 88].

Model checking is one way to formally verify the properties of systems with discrete state spaces in an automatic fashion [89]. The model of the system and the given specification are formulated within a mathematical framework. Afterwards,

dedicated model checking algorithms [90] prove whether the model satisfies the specification by traversing the state space of the model (e.g., represented in the form of Kripke structures). Model checking has been applied to the safety verification of platoons of autonomous vehicles in [91] and to traversing a crossing in [92]. However, the complexity of autonomous driving applications generally renders model checking infeasible due to the curse of dimensionality - namely, the computational burden of traversing discretized high-dimensional state spaces [93, 94].

On the other hand, theorem proving is usually better suited for high-dimensional systems. The system and the desired properties are formulated using logical equations, often with application-specific logics. The verification is then performed by checking the satisfiability of the logical equations or by formal deduction using a database of base axioms [95]. For the domain of autonomous vehicles, theorem proving has been applied to highway entry systems [96], to lane change controllers [97], and to adaptive cruise control systems [98, 99]. Overtaking maneuvers have also been formally verified [100]. However, although theorem proving is powerful and effective, it usually requires manual intervention to generate desired system behaviors, and logical equations must be adapted to new scenarios often. Moreover, if designers fail to implement certain rules, the system's behavior will no longer fulfill the specification.

To ensure that the formal specification is met at all times, the control community has developed correct-by-design control approaches. Correct-by-design controllers are synthesized directly from the specification and never produce system trajectories that reach a set of undesired states (i.e., states violating the specification). For instance, safe controllers are synthesized from linear temporal logic specifications in [101–106] and signal temporal logic in [107, 108]. Another way to construct correct-by-design control is to use barrier certificates [109, 110]. Barrier certificate techniques serve to find barriers in the state space that separate safe and unsafe states. If no trajectory of the autonomous vehicle is able to cross this barrier, the system is guaranteed to be safe. However, autonomous vehicles are only safe if they solely rely on one of the aforementioned synthesized controllers; yet, autonomous vehicles usually make use of different motion planning and control approaches to achieve high comfort. In addition, the different uncertainties in the environment of autonomous vehicles are often not considered or modeled in these approaches.

Set-based reachability analysis can be used to cope with various uncertainties due to its set-based nature. In brief, the reachable set of a dynamical system corresponds to the set of states the system is able to reach over time considering an initial set of states and all admissible system trajectories [111]. For instance, reachability analysis has been used to determine future constraint violations in [112–115]. If the computed reachable set does not intersect with any unsafe set, the system is verified as safe. Moreover, in [116–119], reachability analysis has been used to predict all possible future motions of dynamic obstacles while accounting for possible measurement uncertainties. Based on the obtained set-based prediction, autonomous vehicles are able to check whether planned motions collide with possible trajectories of obstacles [120] (cf. Fig. 1.6). However, these unsafe regions may grow

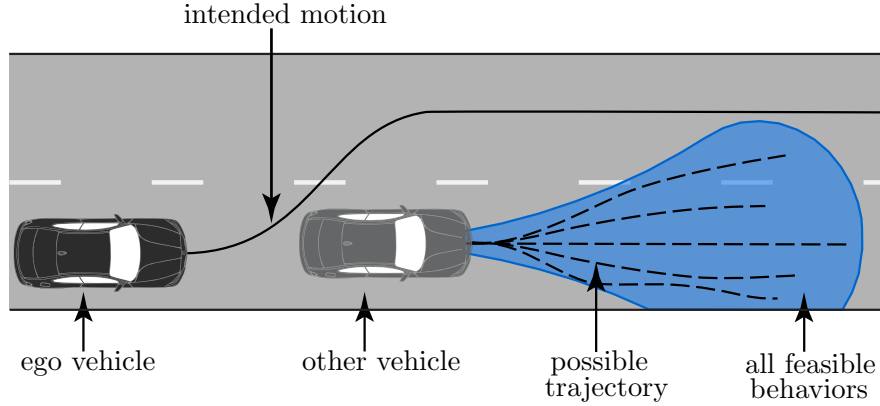


Figure 1.6: Formal verification using reachable sets. The set of all legal behaviors is computed using reachability analysis and includes all possible trajectories of other traffic participants. The ego vehicle’s intended motion is guaranteed to be safe if it never intersects with any computed reachable set (blue area). We note that the reachable set is shown as a projection onto the position domain.

rapidly for long planning horizons (typically used to obtain anticipatory motion plans), eventually blocking all available free space of the autonomous vehicle. As a result, planned motions may often be rejected as being potentially unsafe, leaving the autonomous vehicle without a safe trajectory.

1.2 Contributions to Provably Safe Motion Planning

Existing safety verification techniques are unable to meet the high requirements of legal safety for autonomous vehicles. The majority of existing approaches perform the safety assessment offline before the autonomous vehicle is deployed. However, offline verification cannot provide strict safety guarantees, since autonomous vehicles operate in highly uncertain complex environments. In contrast, existing online verification approaches verify systems during their operation, but still have limitations that restrict their usage in autonomous vehicles. For instance, they require the vehicle to use dedicated controllers [102], leave autonomous vehicles without a safe plan if the intended motion is rejected as unsafe [120], or lose safety guarantees if certain rules have not been implemented [97]. As a result of unsatisfactory verification approaches, new online verification techniques are needed to guarantee legal safety in any traffic situation and for arbitrarily planned motions during the operation of the autonomous vehicle.

This thesis proposes fail-safe motion planning as a novel online verification technique to guarantee the legal safety of autonomous vehicles in arbitrary traffic scenarios during operation. The proposed safety policy ensures that autonomous vehicles only execute provably safe trajectories. Thus, we verify the safety of planned in-

1 Introduction

tended motions in every planning step before execution assuming that other traffic participants obey traffic rules with reasonable care.

In the following, we use Fig. 1.7 to briefly explain our verification technique. Planned intended motions (cf. black lines in Fig. 1.7) of the ego vehicle must be safe considering that other traffic participants may execute any legal behavior (e.g., turning left or right). Using reachability analysis, we first compute all possibly occupied regions in the environment by considering all legal behaviors of surrounding traffic participants (cf. blue areas in Fig. 1.7). For instance, we assume that traffic participants respect the speed limit and do not change to lanes with a different driving direction. The obtained sets are over-approximative and thus always contain the occupancy of other traffic participants, independent of the executed legal behavior. In a second step, we compute fail-safe trajectories. These trajectories branch off at the intended motion of the ego vehicle and do not intersect with any of the possibly occupied regions (cf. red lines in Fig. 3.1). Moreover, fail-safe trajectories end in a set of safe states to ensure that the ego vehicle remains safe for an infinite time horizon. For instance, this set may contain states that correspond to a safe standstill in dedicated areas.

The combination of intended motion plans with fail-safe trajectories considers all legal behaviors of other traffic participants. Even if other traffic participant suddenly change their legal behavior, the ego vehicle remains safe, since it can execute the existing fail-safe trajectory which is provably collision-free. While the ego vehicle moves along the intended motion, our verification technique computes new fail-safe trajectories to ensure safety at all times. The ego vehicle is only

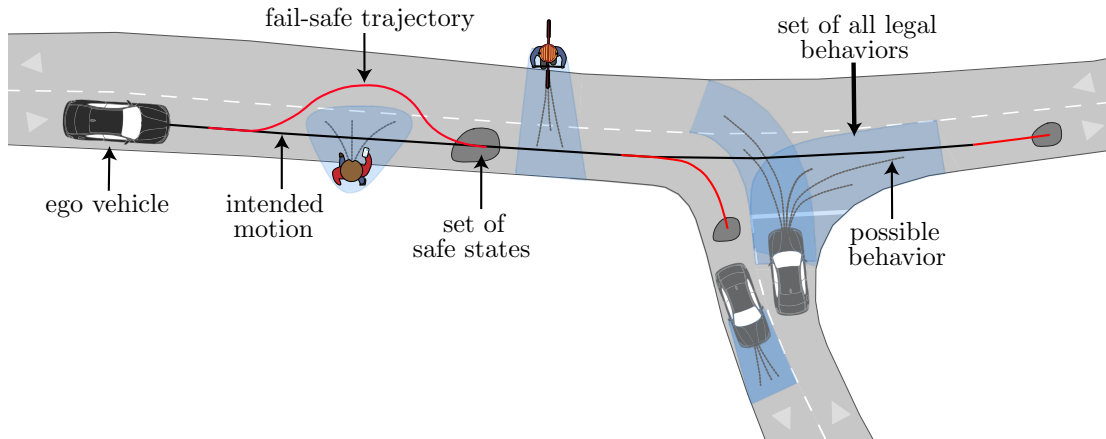


Figure 1.7: Proposed online verification approach [2]. The provably correct verification approach ensures that the ego vehicle maintains fail-safe trajectories at all times (red lines). These trajectories are strictly collision-free against all possible legal behaviors of traffic participants (blue areas) and safeguard the ego vehicle along its intended trajectory (black line) to sets of safe states (dark gray areas). We note that all sets are shown as projections onto the position domain.

allowed to execute the part of the intended motion until it arrives at the fail-safe trajectory if a new fail-safe trajectory has been computed. Consequently, fail-safe motion planning allows us to ensure the legal safety of the ego vehicle in arbitrary traffic situations. This thesis presents the following five major contributions:

1. **Online situation assessment:** Our approach assesses the safety of each traffic situation online during the operation of the autonomous vehicle. Thus, we make use of reachability analysis for other traffic participants to rigorously predict all future evolutions of a scenario (cf. blue areas in Fig. 1.7) while accounting for uncertain measurements. Consequently, the proposed verification approach is able to determine the regions in the environment that enclose the positions (and dimensions) of other traffic participants independent of which future legal motion they execute. This information about unsafe regions is used to evaluate the safety of the vehicle’s intended motion and to compute feasible fail-safe trajectories. In addition, the autonomous vehicle can handle even previously untested scenarios can be handled by the autonomous vehicle on the fly, since the reachable set computation is based on the measured initial states of other traffic participants and a given map.
2. **Fail-safe operation:** The proposed verification approach ensures that autonomous vehicles always maintain a fail-safe trajectory available for execution. These fail-safe trajectories are planned along the intended motion of the vehicle and never enter possibly occupied regions (cf. fail-safe trajectories in Fig. 1.7). Thus, autonomous vehicles remain safe even if the intended motion might lead to a safety-critical situation, that is, if other traffic participants suddenly change their behavior. Moreover, fail-safe trajectories guide vehicles to sets of invariably safe states in the environment. These sets guarantee that autonomous vehicles never enter unsafe states during their operation.
3. **Correct-by-construction:** The proposed approach is based on formal verification to guarantee the safety of the autonomous vehicle using over-approximative motion models for other traffic participants. This approach allows one to reason that collisions are impossible when other traffic participants abide by traffic rules. Conversely, if a collision occurs, another traffic participant must have violated traffic rules. This misbehavior is detected in the proposed approach. The over-approximative design of the approach retains safety even if certain traffic rules are not modeled, since the reachable set computation still considers these behaviors. In these cases, the vehicle only behaves more cautiously. As a result, the proposed approach ensures that autonomous vehicles operate in compliance with legal safety at all times.
4. **Universal design:** The proposed verification approach verifies arbitrarily planned intended motions. When integrated in a vehicle, it is situated between the motion planning and control layer of the autonomous vehicle. The safety of planned motions is evaluated on the fly, and only verified parts are

executed by the controller (i.e., up until the fail-safe trajectory). As soon as a new intended motion is planned, the verification approach tries to verify this motion by computing a new fail-safe trajectory. If the intended motion is rejected as being potentially unsafe, the previously computed fail-safe trajectory is executed, which remains valid by design. The only requirement for the intended motions is that they must be kinematically feasible. As a result, the proposed verification approach allows components above the safety layer to be changed at any time without compromising safety.

5. **Real-world validation:** The proposed verification technique has been extensively validated in simulation and with real test vehicles. This thesis describes one of the most sophisticated evaluations of formal verification for the domain of autonomous vehicles. Based on hand-crafted, safety-critical and recorded traffic scenarios, we demonstrate that autonomous vehicles remain strictly safe at all times. This holds true even when verifying the safety of intended motions that have been planned using machine learning methods. Closed-loop vehicle tests confirm the drivability of fail-safe trajectories and prove that planned motions can be verified during the operation of vehicles. Postprocessing recorded urban traffic indicates that formal verification does not result in conservative behaviors or decreased performance of autonomous vehicles. In a detailed user study, we show that the execution of fail-safe trajectories does not compromise comfort for passengers. In conclusion, the promising results demonstrate the robustness and safety properties of the proposed verification technique for realization in autonomous series vehicles.

1.3 Outline of the Thesis

This thesis is structured as follows. In Ch. 2, we introduce the necessary mathematical foundation for the proposed approaches. The chapter also presents the CommonRoad benchmark suite, which is used throughout the thesis to model scenarios and to reproduce results. Subsequently, we briefly explain reachability analysis of dynamical systems and how this technique is used to predict the legal future behaviors of other traffic participants. Furthermore, we introduce the foundations of convex optimization, which is used to efficiently compute fail-safe trajectories in this thesis.

Next, Ch. 3 introduces fail-safe trajectory planning. After reviewing existing motion planning techniques for autonomous vehicles, we propose the use of convex optimization to efficiently compute trajectories by separating motions into longitudinal and lateral components. Afterwards, the proposed trajectory planning method is adapted for the generation of fail-safe trajectories in arbitrary traffic situations. Since motion planning of autonomous vehicles is usually of a non-convex nature, we show how the non-convex search space can be explored for fail-safe solutions by computing the drivable area (and driving corridors) of the autonomous

vehicle. We conclude this chapter by validating the theoretical contributions to fail-safe trajectory planning in various numerical experiments.

In Ch. 4, we first present common techniques to compute safe states of autonomous vehicles, and we illustrate their drawbacks. Subsequently, we introduce invariably safe sets as a way to compute sets of states that keep autonomous vehicles safe for an infinite time horizon. We propose a recursive definition of invariably safe states and demonstrate how these sets can be determined in a computationally efficient way. We then exploit invariably safe sets for motion planning of autonomous vehicles and describe how they can be used to verify trajectories for infinite time horizons or to determine the time-to-react. Lastly, we validate the proposed safety benefits of invariably safe sets in various numerical experiments.

Subsequently, Ch. 5 presents fail-safe motion planning as a technique to ensure the legal safety of autonomous vehicles by combining fail-safe trajectory planning and invariably safe sets. After briefly introducing common structures of planning frameworks for autonomous vehicles, we show how the proposed verification technique can be integrated into such planning frameworks. We demonstrate the basic steps of the verification during the operation of the autonomous vehicle and formally prove its correctness according to the legal safety specification. Afterwards, we propose the necessary computation steps to verify arbitrary trajectories in detail. We also show how to integrate invariably safe sets in fail-safe planning as linear constraints.

We extensively evaluate the proposed fail-safe motion planning technique in Ch. 6. First, we briefly introduce the utilized vehicle setup. We then present the results of our closed-loop driving experiments, conducted at a fenced BMW test site. Based on recorded traffic situations in the area of Munich, we show how the proposed verification technique ensures safety in typical urban accident hotspots, such as left turns at intersections and jaywalking pedestrians. Afterwards, we assess the intervention rate of our verification technique and the provided passenger comfort in case the vehicle needs to execute fail-safe trajectories.

Ch. 7 summarizes the theoretical and practical contributions of this thesis in the area of provably safe motion planning for autonomous vehicles. We discuss the missing steps toward realizing the approach in series vehicles and the impacts of the proposed online verification technique on the economy and society. Moreover, we outline future work to further improve the performance of the verification technique. Finally, Ch. 8 outlines the scientific publications, patents, and supervised theses that resulted from the research project of this thesis.

2 Notation and Preliminaries

In this chapter, we introduce the necessary mathematical notation and concepts used throughout the thesis. First, we establish the model and environment of the ego vehicle in Sec. 2.1. The CommonRoad benchmark suite for motion planning, presented in Sec. 2.2, is used to implement the developed approaches. A brief overview of reachability analysis is given in Sec. 2.3 and its application to the set-based prediction of traffic participants is presented in Sec. 2.4. Lastly, Sec. 2.5 summarizes the theoretical foundations behind convex optimization.

2.1 Mathematical Notation

We introduce the configuration space $\mathcal{X} \subset \mathbb{R}^n$ as the possible set of states x and $\mathcal{U} \subset \mathbb{R}^m$ as the set of admissible control inputs u of the ego vehicle whose motion is governed by the differential equation

$$\dot{x}(t) = f(x(t), u(t), z(t)), \quad (2.1)$$

where $z(t) \in \mathcal{Z}$ describes disturbances acting on the vehicle's dynamics. We use the notation $x^{(i)}, i \in \mathbb{N}$, to describe the i -th component of the state variable x . Without loss of generality, we assume that the initial time is t_0 . We adhere to the notation $x([t_0, t_1])$ to describe a state trajectory for the time interval $[t_0, t_1], t_0 \leq t_1$. Similarly, we use $u([t_0, t_1])$ to denote an input trajectory for the time interval $[t_0, t_1], t_0 \leq t_1$. By an abuse of notation, we use $u([t_0, t_1]) = \Phi(x([t_0, t_1]), \phi_{\text{ref}}), t_0 \leq t_1$, to emphasize that an input trajectory is generated by a state feedback control law Φ for a given reference trajectory ϕ_{ref} . Furthermore, $\chi(t_1, x(t_0), u([t_0, t_1]), z([t_0, t_1])) \in \mathcal{X}$ denotes the solution of (2.1) at time t_1 subject to the initial state $x(t_0) = x_0$, the input trajectory $u([t_0, t_1])$ and the disturbance $z([t_0, t_1])$. If $z(\cdot) = 0$, we omit the disturbance in the solution χ .

In this thesis, we consider a lane-based environment $\mathcal{E} \subset \mathbb{R}^2$, which is modeled as a subset of the Euclidean space [121]. The set \mathcal{E} is usually extracted from a map of the environment, considering drivable and non-drivable areas. Positions $p_{\text{cart}} = (p_x, p_y)^T \in \mathcal{E}$ are described in a world coordinate frame, such as WGS84 [121]. Below, we introduce the relation occ from the configuration space \mathcal{X} to the lane-based environment \mathcal{E} in world coordinates:

Definition 2 (Occupancy of States) *The operator $\text{occ}(x)$ relates the state vector x to the set of points in the environment \mathcal{E} occupied by the system as $\text{occ}(x) : \mathcal{X} \rightarrow \mathcal{P}(\mathcal{E})$, where $\mathcal{P}(\mathcal{E})$ describes the power set of \mathcal{E} . Given a set $\mathcal{X}' \subset \mathcal{X}$, we define $\text{occ}(\mathcal{X}') := \{\text{occ}(x') \mid x' \in \mathcal{X}'\}$.*

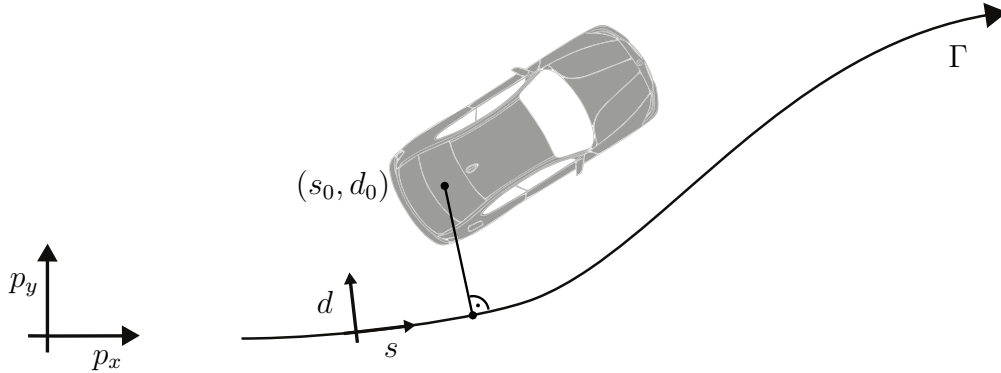


Figure 2.1: Curvilinear coordinate system. The pose (i.e., position and orientation) of the ego vehicle (with respect to the center of the rear axle) is described in a curvilinear coordinate system that is aligned with a given reference path Γ . As a result, positions $p = (p_x, p_y)$ are described by the arc length s_0 and the lateral deviation d_0 .

Besides a world coordinate system, we use a curvilinear coordinate system [122, 123] for motion planning that is aligned with a given reference path Γ . For instance, Γ may correspond to the centerline of a lane and be represented as a polyline $(p_0, p_1, \dots, p_k), p_k \in \mathcal{E}, k \in \mathbb{N}$. As a result, positions in the world coordinate system will be described in terms of the arc length s along Γ and the orthogonal deviation d to Γ (cf. Fig. 2.1). The operator $\Upsilon(p_{\text{cart}})$ transforms a position p_{cart} from the world to the curvilinear coordinate system. The inverse transformation is denoted by Υ^{-1} . It should be noted that depending on Γ , the operator Υ is not necessarily bijective (cf. projection domain) [122].

In this work, we use different operations on sets. For instance, $\mathcal{X}_1 \cup \mathcal{X}_2$ denotes the union and $\mathcal{X}_1 \cap \mathcal{X}_2$ the intersection of two sets \mathcal{X}_1 and \mathcal{X}_2 . Furthermore, the set difference is defined as $\mathcal{X}_1 \setminus \mathcal{X}_2 := \{x_1 \in \mathcal{X}_1 \mid x_1 \notin \mathcal{X}_2\}$ and the Minkowski sum as $\mathcal{X}_1 \oplus \mathcal{X}_2 := \{x_1 + x_2 \mid x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2\}$.

The set $\mathcal{B} \subset \mathbb{N}_+$ contains indices that refer to all safety-relevant dynamic and static obstacles within the environment \mathcal{E} . Information about obstacles in the environment, such as the state and uncertainties, is usually obtained from on-board sensors of the vehicle [124]. The set of possibly occupied points in the environment by an obstacle b at a given time t is represented as an occupancy set:

Definition 3 (Occupancy Set \mathcal{O}) *The occupancy set $\mathcal{O}_b(t) \subseteq \mathcal{E}$ describes the set of points in the environment possibly occupied by an obstacle $b \in \mathcal{B}$ at time t . For the time interval $[t_1, t_2], t_1 < t_2$, we define $\mathcal{O}_b([t_1, t_2]) = \bigcup_{t \in [t_1, t_2]} \mathcal{O}_b(t)$.*

Considering the set of possibly occupied points in the environment, we are able to define the maximal set of collision-free states at a point in time t :

Definition 4 (Collision-Free States \mathcal{F}) *The set $\mathcal{F}(t) \subseteq \mathcal{X}$ is the maximal set of states that are collision-free at time t , that is, $\mathcal{F}(t) := \{x \in \mathcal{X} \mid \text{occ}(x) \cap \mathcal{O}_{\mathcal{B}}(t) = \emptyset\}$ with $\mathcal{O}_{\mathcal{B}}(t) := \bigcup_{b \in \mathcal{B}} \mathcal{O}_b(t)$.*

2.2 CommonRoad Benchmark Suite

CommonRoad is a collection of composable benchmarks for motion planning of autonomous vehicles on roads [125]. It provides researchers with a database of scenarios that can be used for numerical experiments. Each CommonRoad scenario specifies road networks, obstacles and their motion over time, goals, and other constraints. The scenarios are either recorded from real traffic data or handcrafted to create particularly safety-critical situations. Each benchmark is composed of a certain vehicle model and parameter set, a cost function for the evaluation, and a scenario (cf. Fig. 2.2). Solutions to each benchmark can be uploaded to the CommonRoad website and ranked among solutions from other researchers.

Since reproducibility is one of the key aspects of CommonRoad, each numerical experiment can be fully described by a unique ID. The ID defines all required information to reproduce the experiment based on the database that can be downloaded from the CommonRoad website, *commonroad.in.tum.de*. All scenarios used in this thesis are modeled using the CommonRoad specification and are freely available as part of the CommonRoad benchmark suite. We use the CommonRoad release 2018b. The unique ID of each scenario in this thesis is stated in the caption of the corresponding figure.

In addition, we use the Python tools of CommonRoad to plot planning results and scenarios. Advancing the Python tools of CommonRoad for motion planning has been a part of this research project. The tools are open source and available via the Python package-management system pip or public Git repositories. For plotting, trajectories are projected onto the position domain with respect to the center of each obstacle’s shape (rectangles for wheeled traffic and circles for pedestrians) in the world coordinate system. Detailed documentation of the tools can be found on the CommonRoad website.

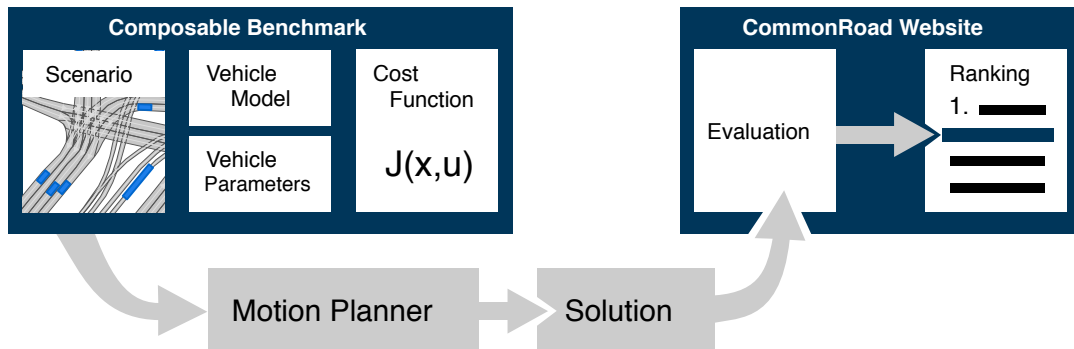


Figure 2.2: Overview of CommonRoad. The CommonRoad suite is a collection of composable benchmarks for motion planning on roads and provides researchers with scenarios, vehicle models, and cost functions. Benchmark solutions can be uploaded to the CommonRoad website and are ranked. ©CommonRoad.

2.3 Reachability Analysis of Dynamical Systems

Set-based reachability analysis computes the set of states that a system can reach at a certain point in time [111, 126–128]. The computation is done by considering all possible input trajectories of a system given an initial set of states. The initial set of states is used to model uncertain initial states. The forward reachable set of a system is formally defined as:

Definition 5 (Forward Reachable Set \mathcal{R}) *The forward reachable set $\mathcal{R} \subseteq \mathcal{X}$ of a system is the set of states that are reachable at time t from an initial set $\mathcal{X}_0 \subset \mathcal{X}$ at time $t_0 = 0$ and subject to the set of inputs \mathcal{U} :*

$$\mathcal{R}(t) := \left\{ \chi(t, x(t_0), u(\cdot)) \mid x(t_0) \in \mathcal{X}_0, \right. \\ \left. \forall t^* \in [t_0, t] : \chi(t^*, x(t_0), u(\cdot)) \in \mathcal{X}, u(t^*) \in \mathcal{U} \right\}.$$

For a given time interval, the forward reachable set is defined as:

Definition 6 (Forward Reachable Set of a Time Interval) *The forward reachable set of a time interval $[t_0, t_1]$ corresponds to the union of the reachable sets at each point $t \in [t_0, t_1]$:*

$$\mathcal{R}([t_0, t_1]) := \bigcup_{t \in [t_0, t_1]} \mathcal{R}(t).$$

The computation of reachable sets for complex systems, such as vehicles, is intractable for most applications with hard time constraints [128, Ch. 3]. Furthermore, the exact reachable set can only be obtained for certain classes of systems [128, Ch. 3].

However, since checking whether a system can reach a set of unsafe states is one of the main applications of reachability analysis, we can also compute over-approximative reachable sets $\bar{\mathcal{R}}$ [129]. A reachable set $\bar{\mathcal{R}}$ is said to be over-approximative for a system if $\forall t \geq t_0 : \bar{\mathcal{R}}(t) \supseteq \mathcal{R}(t)$. Over-approximations $\bar{\mathcal{R}}$ can be achieved by over-approximating computation results in the reachability analysis or by using less restrictive models for which the reachable set can be exactly computed. Due to the over-approximation, a system is provably safe if its over-approximative reachable set does not intersect with any unsafe set. Fig. 2.3 illustrates the difference between exact and over-approximative forward reachable sets. It should be noted that over-approximations might result in overly large reachable sets. For this reason, tight over-approximations are preferred.

Besides the computation of the forward reachable set, reachability analysis can also be used to determine the backward reachable set of a system. In contrast to forward reachable sets, which consider all future evolutions of the system, backward reachability computes the set of states for which at least one trajectory evolves into a certain goal set in a certain time. Specifically, backward reachability analysis

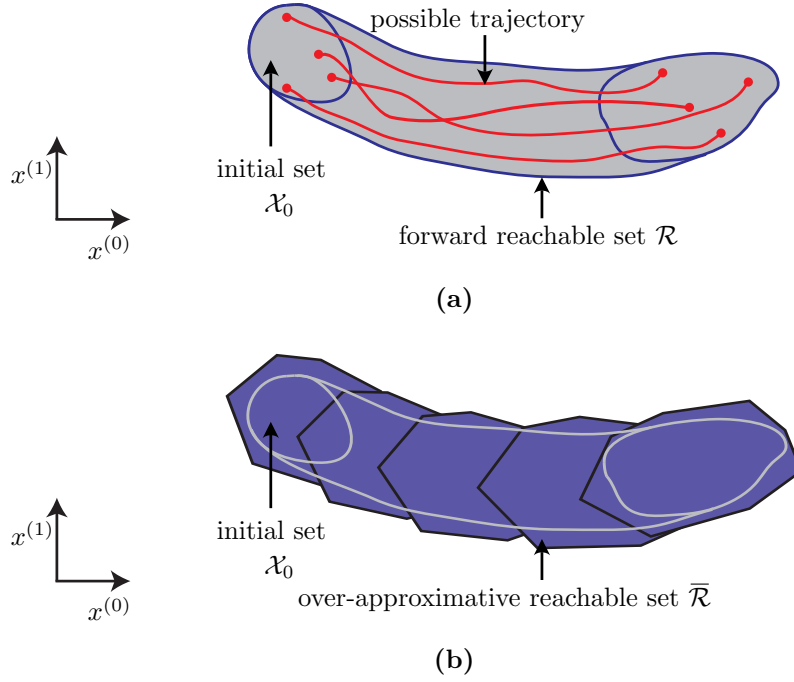


Figure 2.3: Comparison between exact and over-approximative reachable sets. (a) The exact forward reachable set \mathcal{R} contains all states that are reachable by the system. (b) The over-approximative reachable set $\bar{\mathcal{R}}$ encloses the exact reachable set \mathcal{R} . In this example, the over-approximation is achieved by over-approximating computation results and by determining the reachable set for time intervals. Both sets are shown as a projection onto the $x^{(0)}$ - $x^{(1)}$ plane. Figure adapted from M. Althoff.

follows trajectories of a system backward in time while starting in the goal set. In Ch. 4, we use backward reachability analysis to compute the set of states from which our system (2.1) is able to reach a safe goal set collision-free within a certain time [130]. In contrast to Def. 5, we only consider collision-free trajectories of the system in the backward reachability analysis. The collision-free backward reachable set is defined as:

Definition 7 (Collision-Free Backward Reachable Set $\tilde{\mathcal{R}}$) *The collision-free backward reachable set $\tilde{\mathcal{R}} \subseteq \mathcal{X}$ is the set of states from which a system is able to reach a goal set $\mathcal{X}_f \subset \mathcal{X}$ collision-free within a certain finite time $t \geq 0$ considering the set of inputs \mathcal{U} :*

$$\tilde{\mathcal{R}}(t, \mathcal{O}([t_f - t, t_f]), \mathcal{X}_f) := \left\{ x \mid \exists r \in [0, t]: \forall \xi \in [t_f - r, t_f]: \right. \\ \left. \text{occ}(\chi(\xi, x, u([t_f - r, t_f]))) \cap \mathcal{O}(\xi) = \emptyset, u(\xi) \in \mathcal{U}, \right. \\ \left. \chi(t_f, x, u([t_f - r, t_f])) \in \mathcal{X}_f \right\}.$$

2.4 Set-Based Prediction of Other Traffic Participants

Computing occupancy sets \mathcal{O} for a certain point in time is difficult, since the exact future behavior of surrounding traffic participants is usually unknown. We use reachability analysis to account for the uncertain future motions in a set-based fashion [116]. Instead of considering single behaviors, reachability analysis allows us to rigorously predict all feasible future motions of surrounding traffic participants. As a result, we obtain regions in the environment that surrounding traffic participants may occupy over time considering that they are allowed to execute any possible legal behavior. For computational efficiency, we use over-approximative reachable sets $\overline{\mathcal{R}}$ to compute over-approximative occupancy sets based on the current state of obstacles (including measurement uncertainties):

Definition 8 (Over-Approximative Occupancy Sets) *The over-approximative occupancy set $\mathcal{O}_b(t)$ over-approximates the set of occupied points that are reachable by a traffic participant $b \in \mathcal{B}$ at a point in time t : $\mathcal{O}_b(t) := \text{occ}(\overline{\mathcal{R}}_b(t)) \supseteq \text{occ}(\mathcal{R}_b(t))$.*

To obtain the over-approximations, we use less restrictive motion models of other traffic participants for which we can analytically compute the reachable set. More specifically, we utilize a double integrator model (in the form of (2.1)) to predict all feasible future behaviors using over-approximative reachable sets $\overline{\mathcal{R}}$. The model is parameterized according to the type of traffic participant (e.g., cars, trucks, motorbikes, bicyclists, and pedestrians) from a database of parameters. To account for all variations within a class, differential inclusions are used to capture uncertainties, such as varying maximum accelerations and velocities. This set-based prediction approach is implemented in the tool *SPOT* (set-based prediction of other traffic participants) [12, 117], which is available as part of CommonRoad.

In adversarial environments (i.e., obstacles are allowed to perform any trajectory), it is usually impossible to guarantee safety. Instead, we restrict the possible behaviors of other traffic participants according to a legal safety specification inspired by the traffic rules of the Vienna Convention on Road Traffic, which serves as a foundation for safe driving around the world. Thus, we constrain the reachable set computation by defining legal constraints for other traffic participants. Since a specification for legal safety does not yet exist, we propose such a specification by formalizing selected articles of the Vienna Convention on Road Traffic [39] (adopted by 78 countries), ISO norms [131], and physical laws. In general, v and a denote velocity and acceleration, while orientation and curvature are described by θ and κ , respectively. Tab. 2.1 and Tab. 2.2 summarize the legal specifications for wheeled traffic participants and pedestrians, respectively.

We use the legal specification to remove illegal behaviors (according to the specification) from the reachable set. As a result of the removal, the obtained occupancy sets become smaller and only consider legal behaviors. It should be noted that

Table 2.1: List of motion assumptions for cars, trucks, motorbikes, and bicyclists.

Assumption	Description	Source
Λ_{amax}	Absolute acceleration is limited by $ a_{\text{max}} $.	Physical law
Λ_{back}	Driving backwards in a lane is not allowed, i.e., velocities $v \geq 0$.	[39, 14§2]
Λ_{vmax}	Positive longitudinal acceleration is stopped when a parameterized speed $v_{\text{max}} = v_{\text{lim}} f_s$ is reached, where v_{lim} and f_s are the legal speed limit and a speeding factor, respectively.	[39, 13§2]
Λ_{lane}	Leaving the road is forbidden and changing the lane is only allowed if the new lane has the same driving direction as the previous one.	[39, 14§1, 11§2(c)]
Λ_{safe}	Safe distances to the ego vehicle have to be respected to comply with traffic rules. This holds true when driving behind the ego vehicle or merging in front of it.	[39, 13§5, 11§2]
Λ_{over}	When being overtaken by the ego vehicle, a traffic participant is not allowed to endanger the ego vehicle (e.g., by accelerating).	[39, 11§10]

Table 2.2: List of motion assumptions for pedestrians.

Assumption	Description	Source
$\Lambda_{\text{amax,ped}}$	Absolute acceleration is limited by $ a_{\text{max,ped}} $.	Physical law
$\Lambda_{\text{vmax,ped}}$	Maximum absolute velocity is constrained by $v_{\text{max,ped}}$.	[131, ISO norm]
Λ_{slack}	When walking parallel to the road, occupying the strip of the road edge with a width of d_{slack} is allowed (e.g., to avoid obstacles on the sidewalk).	[39, 20§2(a), 20§3, 20§4]
Λ_{perp}	When walking towards the road, crossing the road is only allowed perpendicular, but we allow a deviation of the angle δ_{dev} based on the current heading of the pedestrian.	[39, 20§6(c,d)]

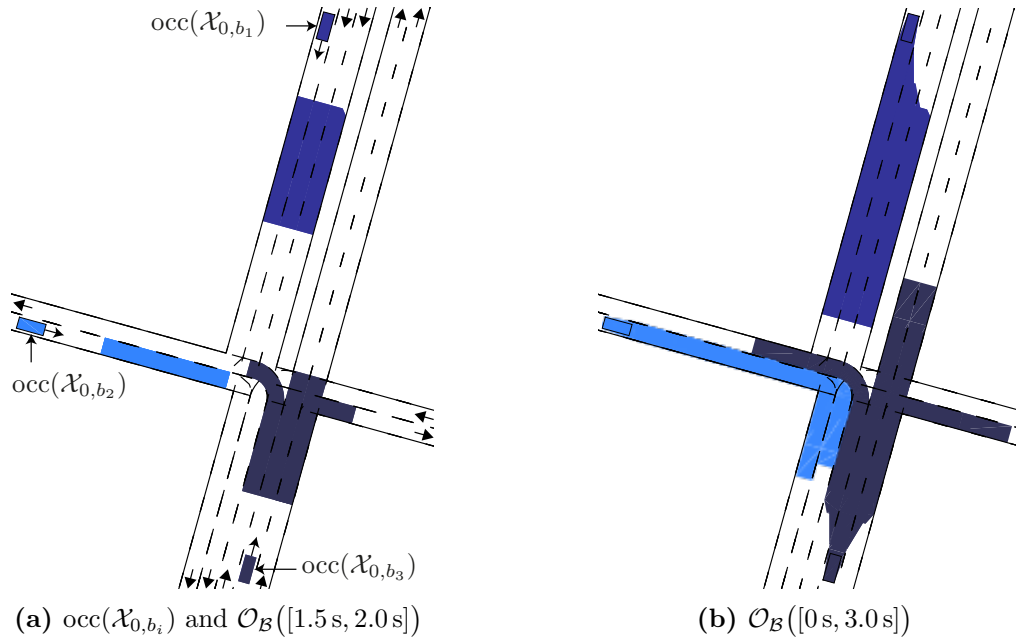


Figure 2.4: Example of set-based predictions. Using reachability analysis, we formally predict all legal behaviors of other traffic participants $b_i \in \mathcal{B}, i \in \{1, 2, 3\}$, considering their initial set of states \mathcal{X}_{0,b_i} to obtain possibly occupied regions $\mathcal{O}_{\mathcal{B}}(t)$ over time. ©2017 IEEE.

even if certain rules are not included in our specification, our result remains over-approximative, since the corresponding illegal behaviors are still included in the reachable set computation. In case a traffic participant violates certain rules, a less restrictive behavior is assumed individually by considering the illegal behavior in the computation. As a result, the prediction directly reacts to possible misbehaviors of traffic participants. Nevertheless, to guarantee legal safety of the ego vehicle, we initially assume that other traffic participants respect traffic rules. In case a collision occurs, we can verifiably argue that another traffic participant has violated the legal specification and thus caused the collision, since our approach ensures that motions of the ego vehicle are provably safe with respect to all legal behaviors of other traffic participants. In these situations, we try to mitigate collisions if possible. Fig. 2.4 illustrates the over-approximative occupancy prediction of SPOT for an uncontrolled intersection.

The set-based prediction makes it possible to prove whether planned trajectories can possibly collide with any legal behavior of other traffic participants (cf. Fig. 2.5). Since the obtained set of collision-free states $\mathcal{F}(t)$ is guaranteed to be collision-free through the computed over-approximative occupancy sets, we can define collision-free input trajectories as follows:

Definition 9 (Collision-Free Input Trajectory) *An input trajectory $u([t_0, t_h])$, $t_0 < t_h$, is called a collision-free input trajectory for the time horizon t_h if $\forall t \in [t_0, t_h] : \chi(t, x(t_0), u([t_0, t])) \in \mathcal{F}(t)$.*

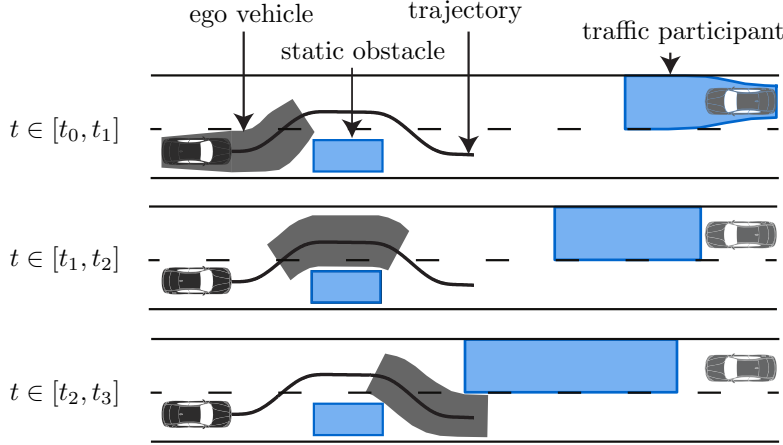


Figure 2.5: Collision-free input trajectory. The trajectory of the ego vehicle is collision-free if the occupancy of the ego vehicle along the trajectory does not intersect with the predicted occupancy sets of other obstacles.

2.5 Convex Optimization

In Ch. 3, we use methods from the field of continuous optimization to generate trajectories for autonomous vehicles in a computationally efficient way. In this regard, an optimal trajectory is obtained by minimizing a certain cost function $J : \mathcal{X} \times \mathcal{U} \times \mathbb{R} \rightarrow \mathbb{R}$ subject to a set of constraints. We formally define a constrained trajectory optimization problem for the time horizon t_h as:

$$\operatorname{argmin}_{x,u} \int_{t_0}^{t_h} J(x(t), u(t)) dt, \quad (2.2)$$

$$\begin{aligned} \text{subject to } & x(t) \in \mathcal{C}_X(t), t \in [t_0, t_h], \\ & u(t) \in \mathcal{C}_U(t), t \in [t_0, t_h], \end{aligned} \quad (2.3)$$

where $\mathcal{C}_X(t)$ and $\mathcal{C}_U(t)$ are constraint sets that describe admissible states $x(t)$ and inputs $u(t)$ of the optimization problem at a point in time t .

In general, optimization problems are classified into convex and non-convex problems. Both categories usually correspond to the complexity of solving the optimization problem. In the following, we first introduce the property of convexity for sets and functions. A set is convex if the line segment between any pair of points within the set lies in the set, which can be formally defined as:

Definition 10 (Convex Set) *A set \mathcal{C} is called a convex set if $\forall x_1, x_2 \in \mathcal{C}, \forall \alpha \in [0, 1] : \alpha x_1 + (1 - \alpha)x_2 \in \mathcal{C}$.*

Convex sets are closed under intersections, Minkowski additions, Cartesian products, and affine functions. Similarly to the convexity of sets, we define convex functions as:

Definition 11 (Convex Function) A function $g : \mathbb{R}^k \rightarrow \mathbb{R}, k \in \mathbb{N}_+$, is a convex function if its domain $\text{dom}(g)$ is a convex set and $\forall o_1, o_2 \in \text{dom}(g), \forall \alpha \in [0, 1] : g(\alpha o_1 + (1 - \alpha)o_2) \leq \alpha g(o_1) + (1 - \alpha)g(o_2)$.

More specifically, a function g is convex if its epigraph $\text{epi}(g) := \{(o, e) \mid x \in \text{dom}(g) \wedge g(o) \leq e\}$ is a convex set. Examples of convex functions are linear and quadratic functions as well as the maximum and the absolute value function. Convex functions are closed under composition.

Fig. 2.6 and 2.7 illustrate the differences between convex and non-convex sets and functions, respectively. Convex functions come with the advantage that any local minimum is also a global minimum [132]. This useful property is a direct result of the convexity (i.e., every point on the line segment between any two points lies either on or above the graph of the function) [132]. Convex optimization problems are defined using convex functions and sets:

Definition 12 (Convex Trajectory Optimization Problem) A trajectory optimization problem is convex if the cost function J in (2.2) is a convex function and the constraint sets $\mathcal{C}_X(t)$ and $\mathcal{C}_U(t)$ in (2.3) are convex sets.

To solve convex optimization problems, solvers exploit the useful property of convexity - namely, every obtained local minimum is also a global minimum. This property allows the solver to follow, for example, the gradient of the cost function until the optimal solution has been found. Conversely, solvers for non-convex problems may be stuck in a local minimum (or maximum). In contrast to non-convex optimization problems, there are many computationally efficient algorithms that solve convex optimization problems, such as the interior-point, cutting-plane, and subgradient methods [133].

In Ch. 3, we use linear-quadratic programs (a special form of convex optimization problems) to compute trajectories for autonomous vehicles [132, Sec. 4.4]. Linear-quadratic programs are optimization problems with a quadratic cost function J and

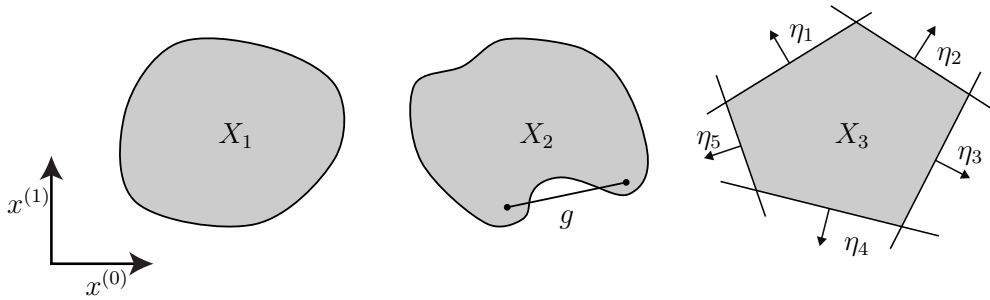


Figure 2.6: Examples of convex and non-convex sets. The set X_1 is a convex set since the line segment between any two points lies within X_1 . On the other hand, the set X_2 is a non-convex set since the line segment g (black line) contains points that are not part of X_2 . The set X_3 is a convex polytope that is formed through the intersection of 5 halfspaces with outward facing normal vectors $\eta_i, i \leq 5$. The sets are shown as a projection onto the $x^{(0)}$ - $x^{(1)}$ plane.

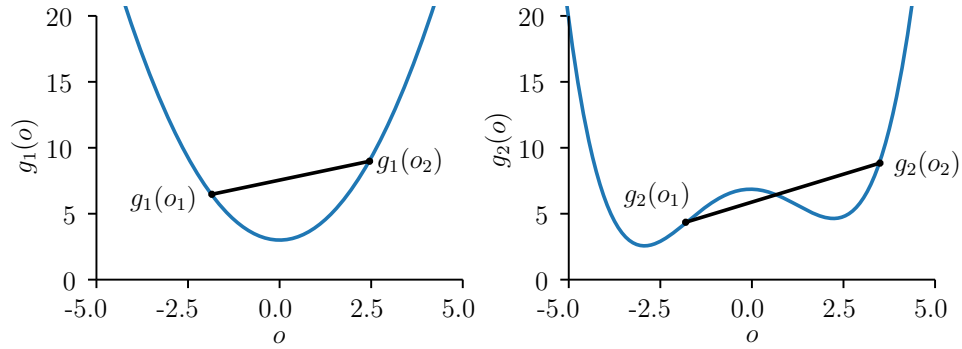


Figure 2.7: Examples of convex and non-convex functions. $g_1(o)$ is a convex function since the line segment between any two points, o_1 and o_2 , lies above the graph. On the other hand, $g_2(o)$ is a non-convex function since there are line segments that intersect the graph. Black lines denote line segments between two example points, o_1 and o_2 .

linear constraint sets. This type of convex optimization problem allows a variety of problems to be modeled and efficiently solved. We model the linear constraint sets of quadratic programs as convex polytopes, which are sets defined through halfspaces:

Definition 13 (Convex Polytope Set Representation) *A convex polytope \mathcal{P} defined by q halfspaces is the set $\mathcal{P} = \{x \in \mathbb{R}^n \mid \mathcal{H}x \leq o, \mathcal{H} \in \mathbb{R}^{q \times n}, o \in \mathbb{R}^q\}$.*

The set X_3 in Fig. 2.6 is a convex polytope, defined by 5 halfspaces. It should be noted that the halfspace representation can also be used to model equality constraints by expressing them through two inequality constraints. For instance, the equality constraint $x = o$ is equivalent to the inequality constraint $(x \leq o \wedge -x \leq -o)$.

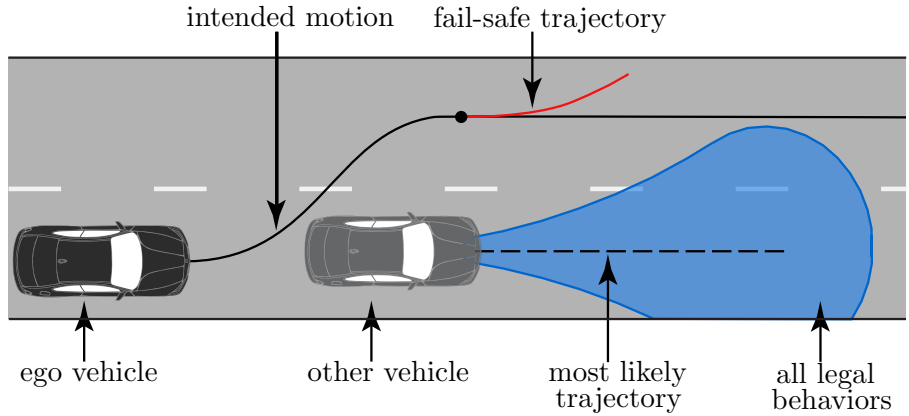
3 Computationally Efficient Fail-Safe Trajectory Planning

In this chapter, we introduce fail-safe trajectory planning as a technique to compute fallback maneuvers for the ego vehicle in safety-critical traffic situations. Sec. 3.1 explains the idea behind fail-safe trajectories and briefly reviews existing motion planning techniques for autonomous vehicles. In Sec. 3.2, we present a novel trajectory planning method that makes use of convex optimization to determine comfortable trajectories. Subsequently, we extend the developed trajectory planner for the computation of fail-safe trajectories in arbitrary traffic scenarios in Sec. 3.3. To obtain fail-safe trajectories even in complex traffic scenarios with small solutions spaces, we propose an approach to efficiently explore the search space of the vehicle in Sec. 3.4. Afterwards, Sec. 3.5 presents numerical experiments highlighting the developed fail-safe trajectory planning approach in different traffic scenarios. This chapter concludes with a summary in Sec. 3.6. The content of this chapter is mainly based on the publications [1, 4, 9, 11, 13, 16].

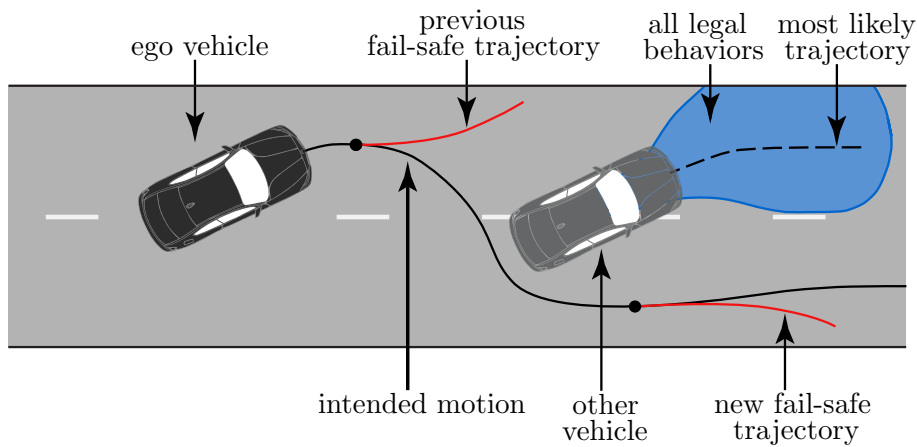
3.1 Introduction and State of the Art

Intended motions of the ego vehicle might be potentially unsafe over finite time horizons when considering all possible legal behaviors of other traffic participants. Usually, intended motions are generated by an intended motion planner and optimized for a comfortable and anticipatory behavior of the ego vehicle for typically long finite time horizons (around 10 s). Hence, intended motion planners usually consider the most likely trajectories (cf. most likely trajectory in Fig. 3.1) of other traffic participants as described in Sec. 1.1.1. Due to the long time horizon of intended motions, the predicted occupancy sets of other traffic participants grow enormously over time, eventually intersecting with the occupancy of the ego vehicle along the intended motion (cf. Sec. 1.1.2). Thus, many intended motions are potentially unsafe for the entire time horizon. However, a short part of each intended motion usually does not result in intersections with predicted occupancy sets, as shown in Fig. 3.1a.

To ensure the safety of intended motions, we apply our fallback maneuver concept only to the first part of the intended long-term motion of the ego vehicle. We consider two time horizons in parallel, as illustrated in Fig. 3.1. We generate provably safe trajectories by appending fail-safe trajectories (depicted by the red path in Fig. 3.1a) to the first part of the intended long-term motion [11, 134].



(a) Initial Scenario



(b) Future Scenario

Figure 3.1: Fail-safe trajectory concept. (a) We combine the first part of the intended motion with our fail-safe trajectory to obtain a safe trajectory that is collision-free with respect to all legal behaviors of obstacles. (b) While the ego vehicle moves along its intended motion, new fail-safe trajectories are computed. If no new valid fail-safe trajectory can be determined, the ego vehicle must execute the previously computed fail-safe trajectory which remains safe by design. ©2020 IEEE.

The time horizon of this provably safe trajectory is significantly shorter than the finite time horizon of the intended motion, such that over-approximative set-based prediction techniques (cf. Sec. 2.4) do not block overly large regions.

Fail-safe trajectories ensure that the ego vehicle remains collision-free in case a safety-critical situation occurs: even if other vehicles deviate from the most likely trajectory by executing another legal behavior, as illustrated in Fig. 3.1b, the ego vehicle remains safe. It should be noted that since the previously computed safe trajectory already anticipates all future legal behaviors of other traffic participants,

it remains safe, even though the traffic situation has changed. In most cases, the motion planner of the ego vehicle obtains a new intended long-term motion and we are able to generate a new valid provably safe trajectory so that the previous fail-safe trajectory does not need to be executed. However, if we cannot compute a fail-safe trajectory for the new intended motion (e.g., if it eventually leads to unsafe situations), the ego vehicle needs to execute the previous fail-safe trajectory if it is located at the branch point (cf. black circle in Fig. 3.1) of this fail-safe trajectory along the intended motion. Even though the ego vehicle has to start executing a fail-safe trajectory, it can recover and return to its nominal planner by computing a fail-safe trajectory for a new intended motion if the safety-critical situation is resolved.

The computation of fail-safe trajectories is done in every planning cycle of the ego vehicle. Therefore, the fail-safe trajectory planning algorithm needs to be real-time capable, meaning faster than the replanning cycle time. Moreover, a fail-safe planner must be able to obtain drivable fail-safe solutions even in small, convoluted solution spaces. Various trajectory planning techniques have been proposed over the years to achieve these goals [135–139]. Most existing motion planning techniques focus on generating of comfortable trajectories, while only a few approaches have been proposed for planning evasive trajectories [114, 140–147]. We first review discrete planning techniques, that is, trajectory planners that obtain trajectories in discretized search spaces, followed by continuous planning methods. Machine learning approaches have also been successfully applied to motion planning, such as [13, 148–158]. However, these techniques are not yet suitable for use in formal verification, since they lack auditability and are difficult to verify [159], so they are not considered for fail-safe trajectory planning in this thesis.

3.1.1 Discrete trajectory planning techniques

Discrete planning approaches are popular planning techniques for autonomous vehicles. These approaches discretize the search space (state or input space) to obtain feasible trajectories. For instance, motion primitives are precomputed trajectory pieces that are concatenated online [115, 160–164]. Since these motion primitives are precomputed offline, the primitive computation can use complex kinematic vehicle models, such as the multi-body model [125]. The online concatenation is often done using classical search algorithms, such as A* search [165, p. 37]. Fig. 3.2a shows an example scenario, in which motion primitives are used to evade a static obstacle that is blocking the ego vehicle’s path. The disadvantage of motion primitives is that a large number of them are often required to solve complex motion planning problems. Moreover, the online search may not be real-time capable when considering a large number of primitives.

Conversely, sampling-based trajectory planners sample states in the search space to obtain feasible trajectories. For instance, *Rapidly-Exploring Random Trees* (RRTs) [166, 167] randomly sample states and connect them to a goal region to obtain drivable trajectories online. Through the random sampling strategy, RRTs

are perfectly suited to traverse high-dimensional search spaces. Their probabilistic completeness ensures that they approach a solution (if it exists) as more time is spent traversing the search space. Their extension RRT* [45, 168] additionally obtains asymptotically optimal trajectories. However, both algorithms, RRT and RRT*, might not obtain motions in time due to the randomized sampling strategy [136].

In contrast, classical graph-search approaches, such as *state lattices*, work on fixed graph structures [169–174]. They obtain sets of trajectories whose goal states are vertices in a fixed predefined grid, resulting in a lattice structure. State lattices are combined with optimal control techniques in [46] to compute jerk-optimal trajectories. The trajectory generation is done by making use of quintic polynomials with fixed initial state and sampled goal states in longitudinal and lateral direction. Fig. 3.2b shows the previous example scenario, but this time we use state lattices to compute an evasive trajectory to avoid a collision with a static obstacle in the ego vehicle’s path. In general, state lattices create drivable trajectories, but they lack optimality due to the fixed grid. In addition, they may require multiple planning cycles to plan complex maneuvers, such as double lane changes [11], resulting in higher computation times in safety-critical situations.

Although discrete planning approaches are often easy to implement and they solve motion problems effectively, they have major disadvantages. Due to the discretization strategy, they may fail to obtain solutions in safety-critical scenarios with small and convoluted solution spaces. For the same reason, they may also fail to determine trajectories ending in small safe terminal sets. However, both these requirements are crucial to meet the high demands of fail-safe trajectory planning.

3.1.2 Continuous trajectory planning techniques

To overcome the limitations of discretization, continuous optimization is increasingly popular in robot motion planning [175–178]. The problem of determining a feasible and collision-free trajectory is solved by minimizing a cost function with respect to a set of state and input constraints (and possibly a set of disturbances). For autonomous mobile robots, the motion planning problem is formulated as a mixed-integer program in [179–184] and as a non-linear optimization problem solved by sequential quadratic programming (SQP) in [122, 185–187]. The resulting optimization problems are non-convex and thus usually not real-time capable, for example since solvers can become stuck in local minima [188].

The generally non-convex motion planning problem can be approximated as a convex problem. For instance, the approximation is done by linearizing the non-linear, non-holonomic vehicle dynamics and separating the motion into a longitudinal and a lateral component [189]. The resulting convex optimization problems can be efficiently solved with global convergence [132, 190, 191]. Convex collision avoidance approaches for autonomous vehicles are proposed in [63, 192]. Optimal longitudinal and lateral trajectories are obtained in [1] using linear-quadratic programs (QP). In [51], a convex formulation is exploited to predict trajectories of

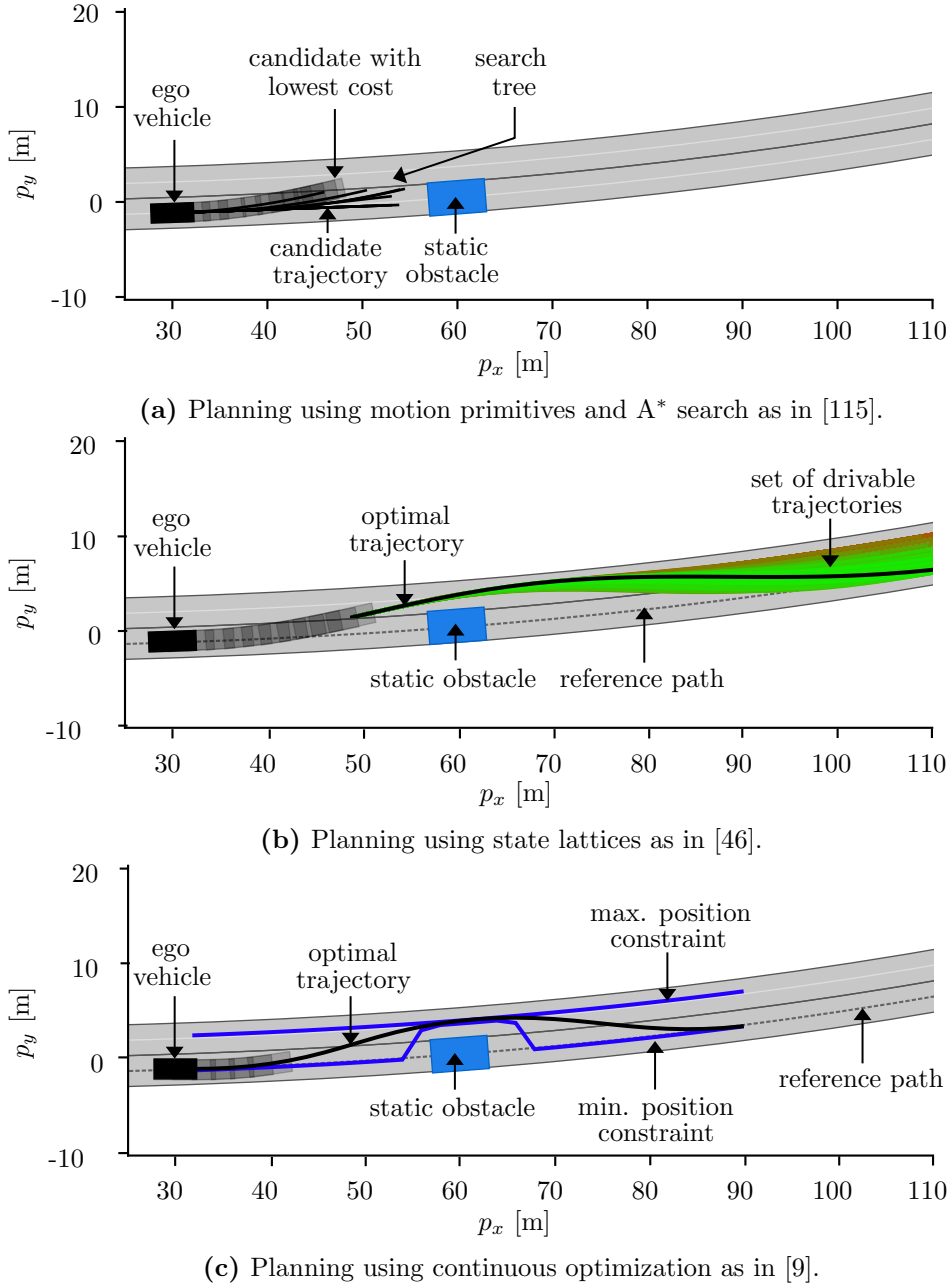


Figure 3.2: Examples of discrete and continuous planning techniques (ZAM_Over-1.1). (a) When planning with motion primitives, the planner constructs a search tree and tries to find a drivable candidate trajectory with the lowest cost. (b) State lattices obtain a set of drivable trajectories within a fixed grid and return the optimal trajectory with the lowest cost. The costs of trajectories are color-coded: red corresponds to high costs and green to low costs. (c) Continuous optimization planners optimize a trajectory in continuous space by minimizing a certain cost function, such as deviations to a reference path, while respecting constraints, such as minimum and maximum position constraints.

traffic participants in multi-vehicle planning. Unfortunately, the longitudinal and lateral separation often results in infeasible trajectories, since both components are heavily linked in complex scenarios, such as evading [9]. Fig. 3.2c illustrates the results of applying convex optimization to plan an evasive trajectory that avoids collisions with a static obstacle in the ego vehicle’s path.

Recent approaches try to eliminate the problem of obtaining infeasible solutions when recombining lateral and longitudinal motions using pre-planning and dexterous constraint formulations. For instance, in [193], a rough longitudinal motion is pre-planned and used to determine a short-term lateral motion afterwards. Pre-planning a rough motion works well in simple scenarios, but is limited when the feasibility of the lateral motion is highly linked to the planned longitudinal motion - for example, when swerving is required to avoid a collision with obstacles. In these scenarios, one requires convex safety regions (e.g., as proposed in [183, 194]) to compute the position constraints for collision-avoidance. Each of the regions corresponds to different valid constraints imposed by safety-relevant obstacles. However, efficient approaches to determine these regions in arbitrary traffic scenarios is not yet available.

Continuous optimization techniques, in particular convex formulations, yield promising results for real-time planning in complex traffic situations. Nevertheless, elaborate problem formulations are required to plan feasible evasive maneuvers in arbitrary traffic situations. In the following section, we present a novel formulation to plan trajectories in real-time by making use of convex optimization. The developed planner is used to plan fail-safe trajectories for the ego vehicle.

3.2 Real-Time Trajectory Planning Using Convex Optimization

Convex optimization offers various benefits for the generation of trajectories. First of all, trajectories are planned in continuous space (cf. Sec. 3.1). Moreover, efficient and mature solving techniques for convex optimization problems exist, allowing trajectories to be obtained in real-time [132]. We use a convex approximation of the motion planning problem by separating motions into a longitudinal (cf. Sec. 3.2.1) and a lateral component (cf. Sec. 3.2.2). In Sec. 3.3, we show how both components can be combined to obtain feasible fail-safe trajectories in many scenarios. The motion of the ego vehicle is described using a curvilinear coordinate system that is aligned to a given reference path Γ (cf. Sec. 2.1), such as the centerline of the current lane. The convex trajectory optimization problem of each component is formulated as a quadratic program (cf. Sec. 2.5). The presented cost functions J in this section are examples and can be modified to include other terms (e.g., separate costs for the final state of a trajectory or punishing high inputs).

3.2.1 Planning longitudinal motions

We describe the state of the ego vehicle's longitudinal motion as $x_{\text{lon}} = (s, v, a, j)^T$, where s is the longitudinal position, v is the velocity, a is the acceleration, and j is the jerk of the vehicle's center point of the rear axle along a given reference path Γ (cf. Fig. 3.3). We choose the rear axle as the reference point to disregard the slip angle, as shown in [195, 196]. Using the jounce as the input, $u_{\text{lon}}(t) = \ddot{a}(t)$, the longitudinal motion of the vehicle is modeled by the linear time-invariant system

$$\frac{d^4}{dt^4}s(t) = u_{\text{lon}}(t). \quad (3.1)$$

In order to express the linear longitudinal model as a set of linear constraints for the convex optimization problem, we use the state space representation of (3.1) to add the equality constraint

$$\dot{x}_{\text{lon}} = \underbrace{\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}}_{A_{\text{lon}} \in \mathbb{R}^{4 \times 4}} x_{\text{lon}}(t) + \underbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}}_{B_{\text{lon}} \in \mathbb{R}^{4 \times 1}} u_{\text{lon}}(t). \quad (3.2)$$

Moreover, we apply the following time-invariant state constraints to ensure that each obtained trajectory is kinematically feasible:

$$\begin{aligned} v_{\min} &\leq x_{\text{lon}}^{(1)}(t) \leq v_{\max}, \\ a_{\min} &\leq x_{\text{lon}}^{(2)}(t) \leq a_{\max}, \\ j_{\min} &\leq x_{\text{lon}}^{(3)}(t) \leq j_{\max}. \end{aligned} \quad (3.3)$$

In order to incorporate collision avoidance, we restrict the set of feasible positions based on obstacles blocking the reference path Γ :

$$s_{\min}(t) \leq x_{\text{lon}}^{(0)}(t) \leq s_{\max}(t). \quad (3.4)$$

The computation of $s_{\min}(t)$ and $s_{\max}(t)$ is described in Sec. 3.3.

The quadratic cost function J_{lon} of the longitudinal trajectory optimization problem favors comfortable trajectories by punishing high accelerations and jerk with weights $w_a \in \mathbb{R}_+$ and $w_j \in \mathbb{R}_+$, respectively, and is defined as:

$$J_{\text{lon}}(x_{\text{lon}}(t)) = w_a x_{\text{lon}}^{(2)}(t)^2 + w_j x_{\text{lon}}^{(3)}(t)^2 dt. \quad (3.5)$$

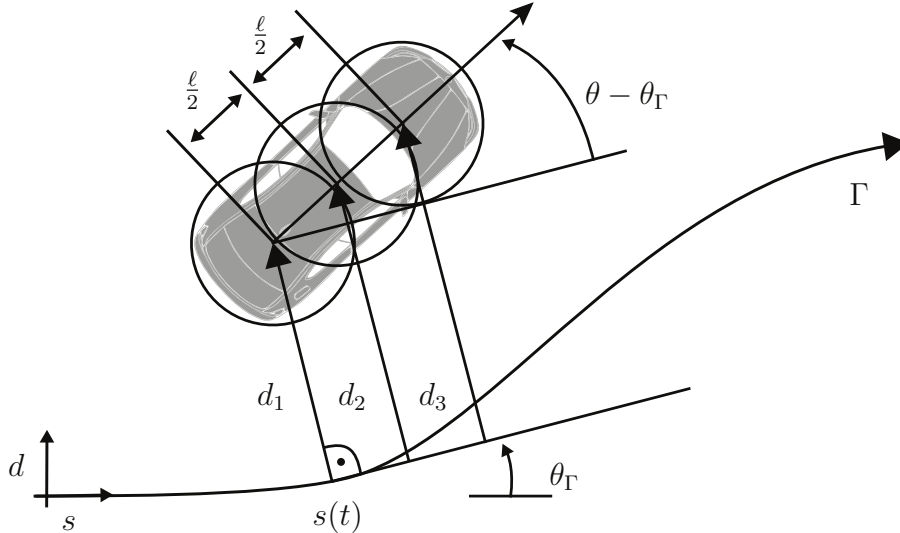


Figure 3.3: Linearized kinematic model for planning. The kinematic model is described with respect to a curvilinear coordinate system aligned to the reference path Γ with orientation θ_Γ . The vehicle's pose is described by the longitudinal position s , the lateral deviation d , and the orientation θ . The vehicle's shape is approximated using three circles with radius r . ©2018 IEEE.

3.2.2 Planning lateral motions

The lateral motion of the vehicle is modeled by the state $x_{\text{lat}} = (d, \theta, \kappa, \dot{\kappa})^T$, where d is the lateral distance to the reference path Γ , θ is the orientation, κ is the curvature, and $\dot{\kappa}$ is the change of curvature of the ego vehicle. We choose the second derivative of the curvature as the input, $u_{\text{lat}}(t) = \ddot{\kappa}(t)$, to obtain smooth lateral trajectories. Since the ego vehicle is supposed to move along the predefined reference path Γ , we can assume that the orientation difference $\Delta = \theta - \theta_\Gamma$ between the current orientation and the reference path orientation θ_Γ is negligibly small; larger deviations usually to more conservative behavior [195]. Thus, we are able to approximate the trigonometric functions as $\sin(\Delta) \approx \Delta$ and $\cos(\Delta) \approx 1$. We use a modeling trick to efficiently compute lateral positions and integrate collision avoidance into the lateral optimization problem: instead of introducing the reference path's orientation θ_Γ as a new state variable, we model θ_Γ as a disturbance $z(t) = \theta_\Gamma(s(t))$ on the lateral motion. This disturbance model allows us to compute lateral position constraints with respect to the desired orientation $\theta \approx \theta_\Gamma$ of the ego vehicle along Γ as shown later. Referring to the kinematic single-track vehicle model [125, 196], the lateral motion of the vehicle is given by the linear system

$$\dot{x}_{\text{lat}} = \underbrace{\begin{pmatrix} 0 & v(t) & 0 & 0 \\ 0 & 0 & v(t) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}}_{A_{\text{lat}} \in \mathbb{R}^{4 \times 4}} x_{\text{lat}}(t) + \underbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}}_{B_{\text{lat}} \in \mathbb{R}^{4 \times 1}} u_{\text{lat}}(t) + \underbrace{\begin{pmatrix} -v(t) \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{E_{\text{lat},1} \in \mathbb{R}^{4 \times 1}} z(t). \quad (3.6)$$

3.2 Real-Time Trajectory Planning Using Convex Optimization

We note that (3.6) qualifies as a linear system because $v(t)$ is not a state variable for the lateral dynamics, but a time-variant parameter provided by the planned longitudinal motion.

For collision avoidance, we over-approximate the shape of the ego vehicle using three circles with equal radius r (cf. Fig. 3.3) [197]. Without loss of generality, we choose the centers of the first and third circle to coincide with the rear and front axle of the ego vehicle, respectively. The distance between the center points of the first and third circle corresponds to ℓ (cf. App. A.1). The center of the second circle is positioned such that the distance to the other circle's center is $\frac{1}{2}\ell$. As a result of this positioning, the lateral distance d_i from the i -th circle's center, $i \in \{1, 2, 3\}$, to the reference path Γ can be computed as (over-approximation):

$$d_i = d + \frac{i-1}{2}\ell \sin(\theta - \theta_\Gamma) \approx d + \frac{i-1}{2}\ell(\theta - \theta_\Gamma). \quad (3.7)$$

We define the constrained values of the system as $x_{\text{constr}} = (d_1, d_2, d_3, \kappa, \dot{\kappa})^T$:

$$x_{\text{constr}}(t) = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & \frac{1}{2}\ell & 0 & 0 \\ 1 & \ell & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{C_{\text{lat}} \in \mathbb{R}^{5 \times 4}} x_{\text{lat}}(t) + \underbrace{\begin{pmatrix} 0 \\ -\frac{1}{2}\ell \\ -\ell \\ 0 \\ 0 \end{pmatrix}}_{E_{\text{lat},2} \in \mathbb{R}^{5 \times 1}} z(t). \quad (3.8)$$

Collision avoidance constraints are incorporated into the lateral motion model by computing the allowed lateral displacement of the ego vehicle along the reference path Γ . Therefore, we compute the minimum and maximum lateral displacement for each circle $i \in \{1, 2, 3\}$ such that the circle is not colliding with any obstacle. Furthermore, the physical constraints of the steering actuators are included:

$$\underbrace{\begin{pmatrix} d_{1,\min}(t) \\ d_{2,\min}(t) \\ d_{3,\min}(t) \\ \kappa_{\text{lim},\min}(t) \\ \dot{\kappa}_{\min}(t) \end{pmatrix}}_{x_{\min}(t)} \leq x_{\text{constr}}(t) \leq \underbrace{\begin{pmatrix} d_{1,\max}(t) \\ d_{2,\max}(t) \\ d_{3,\max}(t) \\ \kappa_{\text{lim},\max}(t) \\ \dot{\kappa}_{\max}(t) \end{pmatrix}}_{x_{\max}(t)}. \quad (3.9)$$

To incorporate the maximum feasible lateral acceleration for higher velocities (with respect to the circle of forces [3]), we set

$$\begin{aligned} |\kappa_{\text{lim},\min}(t)| &= -\max\left(\frac{\sqrt{a_{\text{max}}^2 - a(t)^2}}{v^2(t)}, |\kappa_{\min}|\right), \\ |\kappa_{\text{lim},\max}(t)| &= \min\left(\frac{\sqrt{a_{\text{max}}^2 - a(t)^2}}{v^2(t)}, \kappa_{\max}\right). \end{aligned} \quad (3.10)$$

It should be noted that (3.10) has a singularity at $v(t) = 0$; when implemented, one may use the denominator $\min(v(t), \epsilon)^2$ with an arbitrary small value ϵ for numerical tractability. This change does not influence the constraints, since the curvature is still limited to κ_{\min} (or κ_{\max}). Another possibility is to switch to a different planner for low velocities.

The quadratic cost function J_{lat} of the lateral trajectory optimization problem with weights $w_d \in \mathbb{R}_+$, $w_\theta \in \mathbb{R}_+$, $w_\kappa \in \mathbb{R}_+$, and $w_{\dot{\kappa}} \in \mathbb{R}_+$ minimizes the lateral distance to Γ and orientation deviation from θ_Γ and punishes high curvature rates to achieve smooth trajectories:

$$\begin{aligned} J_{\text{lat}}(x_{\text{lat}}(t)) = & w_d x_{\text{lat}}^{(0)}(t)^2 + w_\theta (x_{\text{lat}}^{(1)}(t) - \theta_\Gamma(t))^2 \\ & + w_\kappa x_{\text{lat}}^{(2)}(t)^2 + w_{\dot{\kappa}} x_{\text{lat}}^{(3)}(t)^2 dt. \end{aligned} \quad (3.11)$$

3.2.3 Enhancing passenger comfort through slack variables

Acceleration profiles with partly constant acceleration phases enhance driving comfort for passengers by reducing maximum accelerations [198]. We model these constant acceleration phases by integrating slack variables $\varsigma \in \mathbb{R}$ and a two-stage cost increase into the longitudinal motion planning problem (cf. Sec. 3.2.1). Slack variables have been used in optimization to convert inequality constraints into equality constraints [132, Ch. 4]. For the sake of clarity, we demonstrate the approach for the case of braking; however, the approach works analogous for positive accelerations. We introduce two additional deceleration limits to model a two-stage cost increase, $a_{\text{lim},1}$ and $a_{\text{lim},2}$, with $-|a_{\text{max}}| < a_{\text{lim},2} < a_{\text{lim},1} < 0$. Furthermore, we define slack variables $\varsigma_{\text{lon},1} \geq 0$ and $\varsigma_{\text{lon},2} \geq 0$ and add the following time-invariant constraints to the longitudinal trajectory optimization problem:

$$x_{\text{lon}}^{(2)}(t) \geq a_{\text{lim},1} - \varsigma_{\text{lon},1}, \quad (3.12a)$$

$$x_{\text{lon}}^{(2)}(t) \geq a_{\text{lim},2} - \varsigma_{\text{lon},2}. \quad (3.12b)$$

By inducing linear costs J_1 for $\varsigma_{\text{lon},1}$ and quadratic costs J_2 for $\varsigma_{\text{lon},2}$, we can model constant acceleration phases, since the solver of the optimization problem aims at minimizing costs. Fig. 3.4 illustrates the resulting acceleration profiles and Fig. 3.5 visualizes the changing costs. Acceleration profiles with accelerations $a \leq a_{\text{lim},1}$ are smoothed during the optimization, since costs for $\varsigma_{\text{lon},1}$ are minimized. In the second stage, profiles with accelerations $a \leq a_{\text{lim},2}$ are optimized as partly constant due to the quadratically increasing costs for the use of $\varsigma_{\text{lon},2}$. It should be noted that the weights of the cost functions J_1 and J_2 for the slack variables must be chosen carefully in order to not distort the optimal solution of the unaltered optimization problem. For instance, if the new cost function of the optimization problem calculates fewer costs when $\varsigma_{\text{lon},1} > 0$, $\varsigma_{\text{lon},2} > 0$, then the optimal solution makes use of the slack variables without getting the desired shape in the acceleration profile.

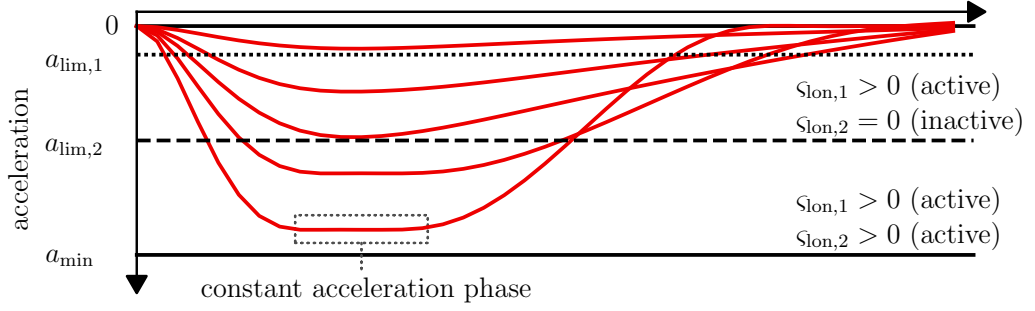


Figure 3.4: Slack variables for comfortable braking profiles. Slack variables are used to influence the shape of deceleration profiles with the aim of enhancing comfort. Planned accelerations are punished with costs in a two-stage approach: accelerations $a \geq a_{\text{lim},1}$ induce linear costs when $\varsigma_{\text{lon},1} > 0$ (active) and $a \geq a_{\text{lim},2}$ induce quadratic costs, resulting in tub-shaped profiles. ©2020 IEEE.

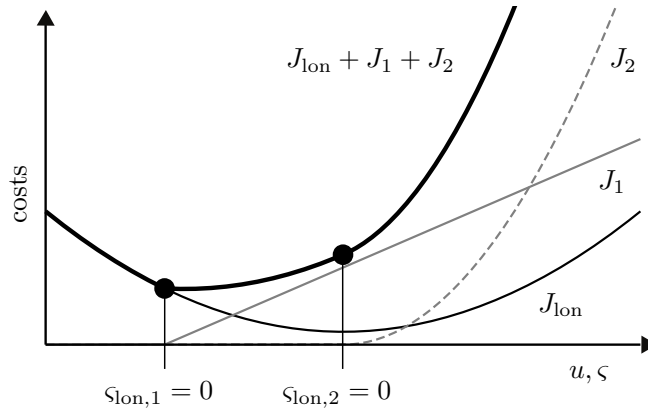


Figure 3.5: Two-stage cost increase for slack variables. Slack variables are used to influence the shape of deceleration profiles with the aim of enhancing comfort. Planned accelerations are punished with costs in a two-stage approach: accelerations $a \geq a_{\text{lim},1}$ induce linear costs when $\varsigma_{\text{lon},1} > 0$ (active) and $a \geq a_{\text{lim},2}$ induce quadratic costs when $\varsigma_{\text{lon},2} > 0$ (active), resulting in tub-shaped profiles.

3.3 Fail-Safe Trajectory Planning in Arbitrary Traffic Scenarios

In order to obtain provably safe trajectories (cf. Sec. 3.1), we have to determine 1) a state along the intended trajectory at which the fail-safe trajectory should start and 2) the optimized fail-safe trajectory itself. The computation of the latest possible position to branch off the fail-safe trajectory is described in Ch. 4 (including desired goal sets of fail-safe trajectories). To find an optimal fail-safe trajectory, we use the proposed convex optimization problems in Sec. 3.2 which separately consider the lateral and longitudinal dynamics of the vehicle [1]. Although this separation into two components may result in infeasible trajectories [9], our developed fail-safe

trajectory planning approach guarantees the drivability of the resulting motion plan in many scenarios.

Fig. 3.6 illustrates the general procedure for computing fail-safe trajectories using the decoupled motion problems described in Sec. 3.2. We assume that the initial state x_0 of the fail-safe trajectory and the reference path Γ are known a priori. This information is typically provided by the ego vehicle's odometry system and a given map. Moreover, the predicted occupancy sets $\mathcal{O}_{\mathcal{B}}$ that capture the feasible future behaviors of other traffic participants are given (cf. Sec. 2.4).

In Step 1 of Fig. 3.6, we compute the longitudinal collision constraints based on the predicted occupancy sets. Inspired by [197], we enlarge $\mathcal{O}_b(t)$ with R_{lon} , which describes the smallest circumscribing circle covering the ego vehicle's dimensions. The enlarged occupancy set is then given by $\mathcal{O}_{b,\text{enl}}(t) := \mathcal{O}_b(t) \oplus R_{\text{lon}}$. Subsequently, the enlarged occupancy $\mathcal{O}_{b,\text{enl}}(t)$ (cf. Def. 3) of each safety-relevant obstacle $b \in \mathcal{B}$ is transformed into the curvilinear coordinate system that is aligned with Γ , resulting in $\mathcal{O}_{b,\text{cls}}(t) := \{\Upsilon(p) \mid p \in \mathcal{O}_{b,\text{enl}}(t)\}$. We use the correction term Δ_{cor} to transform the ego vehicle's reference point on the rear axle to the center of its shape. Based on the longitudinal position of the vehicle s_0 at the initial planning time t_0 , the

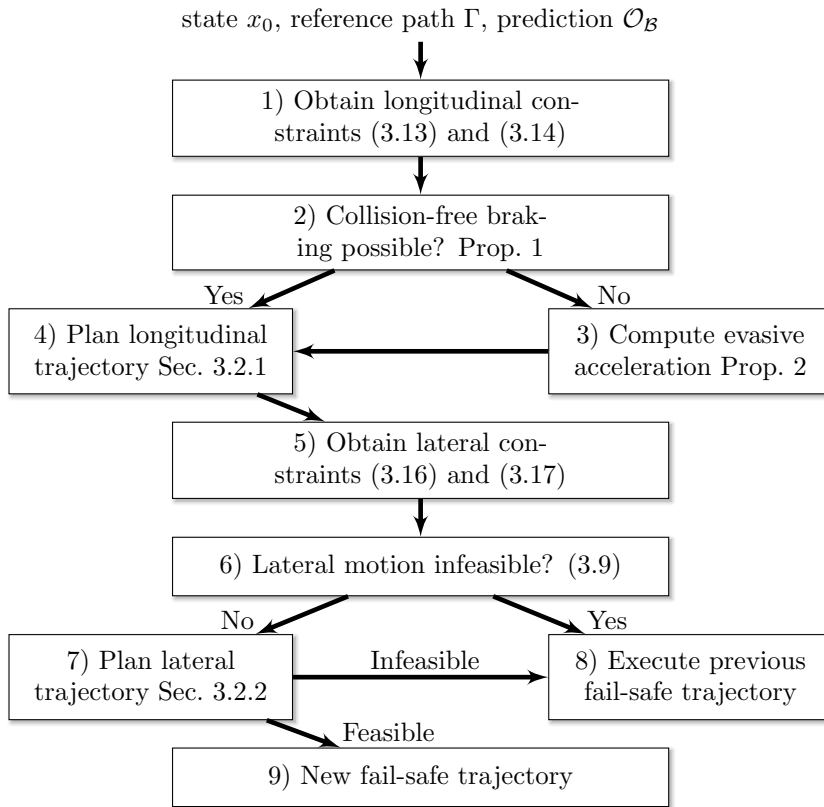


Figure 3.6: General procedure to compute fail-safe trajectories. Based on a given initial state x_0 , reference path Γ , and occupancy prediction $\mathcal{O}_{\mathcal{B}}$, we compute collision-free fail-safe trajectories using separated longitudinal and lateral trajectory optimization problems. ©2018 IEEE.

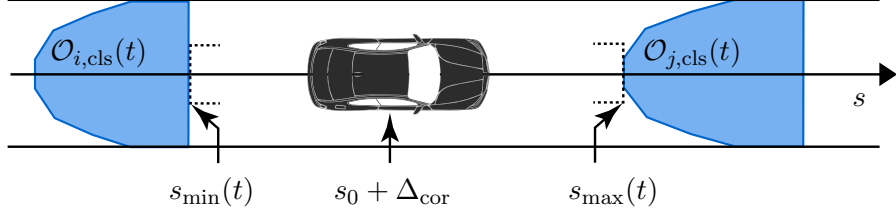


Figure 3.7: Computation of longitudinal collision constraints. The minimum and maximum position constraints, s_{\min} and s_{\max} , in a lane are computed based on the given occupancy sets $\mathcal{O}_{i,\text{cls}}, i \in \mathcal{B}$, and $\mathcal{O}_{j,\text{cls}}, j \in \mathcal{B}$. ©2018 IEEE.

maximum position $s_{\max}(t)$ in (3.4) is computed as (cf. Fig. 3.7):

$$s_{\max}(t) = \inf \{s - \Delta_{\text{cor}} \mid s - \Delta_{\text{cor}} > s_0 \wedge (s, d)^T \in \mathcal{O}_{b,\text{cls}}(t), b \in \mathcal{B}\}. \quad (3.13)$$

The minimum position constraint $s(t) \geq s_{\min}(t)$ is obtained similarly as:

$$s_{\min}(t) = \sup \{s - \Delta_{\text{cor}} \mid s - \Delta_{\text{cor}} < s_0 \wedge (s, d)^T \in \mathcal{O}_{b,\text{cls}}(t), b \in \mathcal{B}\}. \quad (3.14)$$

It should be noted that $s_{\min}(t)$ is only used if the ego vehicle changes to another lane as described in [9]. For the current lane of the ego vehicle, $s_{\min}(t)$ is omitted since following vehicles need to keep a safe distance to the ego vehicle as described in our legal safety specification (cf. Sec. 2.4).

In Step 2 of Fig. 3.6, we check if a braking maneuver alone is sufficient to avoid a collision as this is often considered to be the preferred maneuver for passengers in emergency situations [199]. Since the occupancy sets include information about the dynamics of the obstacles over time, including positions during emergency braking (a legal behavior that is always included in the prediction), we can use (3.13) for this check. We consider rather straight lanes in the following; in Ch. 4, we show the extension to lanes with arbitrary curvatures.

Proposition 1 (Collision Avoidance Through Braking) *A collision with obstacles, represented as a collision constraint $s(t) \leq s_{\max}(t), t \in [t_0, t_h]$, can be avoided for the initial position s_0 , velocity v_0 , and reaction time δ_{brake} of the ego vehicle using emergency braking with $-|a_{\max}|$ if*

$$\forall t \in [t_0, t_h] : s_0 + v_0(\tau) - \frac{1}{2}|a_{\max}|\max(\tau - \delta_{\text{brake}}, 0)^2 \leq s_{\max}(t),$$

$$\tau := \min(t, v_0/|a_{\max}| + \delta_{\text{brake}}).$$

Proof *Using the maximum feasible deceleration a_{\max} , collision-avoidance using braking directly follows from the definition of $s_{\max}(t)$ in (3.13). ■*

In case the ego vehicle is able to avoid a potential collision using a braking maneuver, we compute the longitudinal braking trajectory using the longitudinal planner described in Sec. 3.2.1. It should be noted that this approach also works with

crossing traffic. In this situation, the driving corridor for the longitudinal motion gets blocked by the crossing obstacle at some point. An example, in which the ego vehicle avoids a collision with crossing traffic by initiating a braking maneuver, is illustrated in Sec. 3.5.

If a braking maneuver is not sufficient to remain collision-free, collisions may be avoided by swerving to another lane. For these situations, we must ensure that the required maximum lateral acceleration a_{eva} for evading is feasible throughout the planned maneuver, since the longitudinal and lateral dynamics in the kinematic motion model are decoupled (cf. Sec. 3.2). In the worst case, the evasive maneuver does not allow braking anymore, that is $|a_{\text{eva}}| = |a_{\text{max}}|$. Let us first introduce the *guaranteed time-to-collision* (cf. Fig. 3.8), which is the time until the ego vehicle intersects with occupancy sets when driving with constant velocity.

Definition 14 (Guaranteed Time-To-Collision) *Assuming a collision is possible, the guaranteed time-to-collision (GTTC) with respect to the initial longitudinal position s_0 and velocity v_0 of the ego vehicle and the maximum allowed position $s_{\text{max}}(t), t \in [t_0, t_h]$, is defined as*

$$t_{\text{GTTC}} := \operatorname{argmin}_{t \in [t_0, t_h]} |(s_0 + v_0 t) - s_{\text{max}}(t)|.$$

Several definitions of the time-to-collision exist in the literature [200]; our definition corresponds to the point in time when the occupancy of the ego vehicle definitely intersects with the occupancy of preceding obstacles when assuming constant velocity over the finite planning horizon.

Finally, we introduce the duration of the evasive phase of the maneuver as t_{GTTC} , assuming no deceleration, and the lateral distance to fully reach an adjacent lane as $d_{\text{eva}} > 0$.

Proposition 2 (Evasive Acceleration) *The minimum required lateral acceleration a_{eva} of an evasive maneuver with initial lateral velocity $|v_{\text{lat}}| \geq 0$ over the lateral*

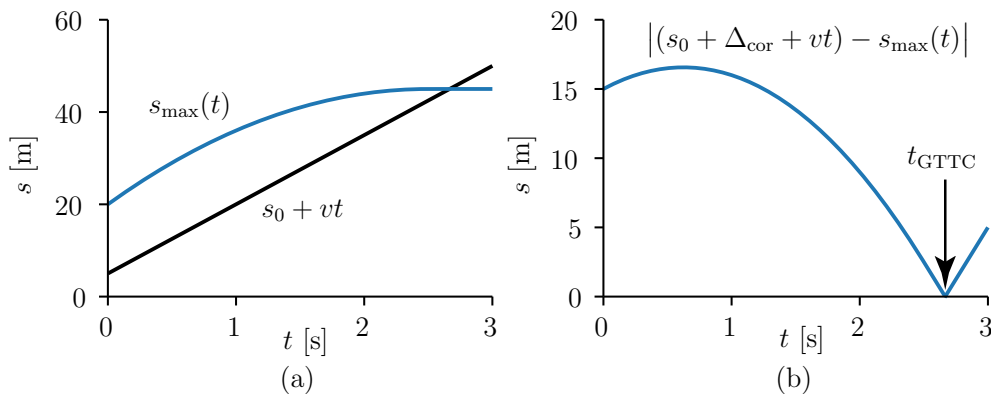


Figure 3.8: Illustration of the GTTC. (a) Constant velocity prediction, $(s_0 + \Delta_{\text{cor}} + vt), t \geq 0$, of the ego vehicle and the maximum position constraints, $s_{\text{max}}(t)$, over time t . (b) Absolute relative distance (convex) and the t_{GTTC} .

distance d_{eva} with duration t_{GTTC} and reaction time for steering $\delta_{\text{steer}} < t_{\text{GTTC}}$ is obtained as

$$a_{\text{eva}} = \frac{2(d_{\text{eva}} - |v_{\text{lat}}|t_{\text{GTTC}})}{(t_{\text{GTTC}} - \delta_{\text{steer}})^2}.$$

Proof The lateral motion of the ego vehicle can be described using the dynamics of a double integrator system [7, III-A]: $d(t) = d_0 + v_{\text{lat}}t + \frac{1}{2}a_{\text{eva}}(t - \delta_{\text{steer}})^2$, $t > \delta_{\text{steer}}$. Setting the desired travelled distance to $d(t) = d_{\text{eva}}$ at time $t = t_{\text{GTTC}}$ (assuming initial time $t_0 = 0$ and distance $d_0 = 0$) results in $d_{\text{eva}} = v_{\text{lat}}t_{\text{GTTC}} + \frac{1}{2}a_{\text{eva}}(t_{\text{GTTC}} - \delta_{\text{steer}})^2$. Solving for a_{eva} results in the required lateral acceleration of the maneuver. ■

Considering the maximum feasible absolute acceleration $|a_{\text{max}}|$ of the ego vehicle, the maximum allowed longitudinal acceleration is computed as

$$a_{\text{lon}} = \sqrt{a_{\text{max}}^2 - a_{\text{eva}}^2}. \quad (3.15)$$

This maximum longitudinal acceleration is added as a constraint $-a_{\text{lon}} \leq a(t) \leq a_{\text{lon}}$ to the longitudinal optimization problem (cf. Sec. 3.2.1). As a result, we are able to plan a longitudinal braking maneuver which ensures that the remaining lateral acceleration capabilities allow swerving. We note that a_{lon} is time-invariant, since we assume that the duration of the evasive maneuver corresponds to the planning horizon of the fail-safe trajectory.

In Step 5 of Fig. 3.6, the lateral collision constraints are computed. Therefore, we predict the poses of the ego vehicle along Γ with respect to the previously planned longitudinal motion while assuming $\theta(s(t)) = \theta_{\Gamma}(s(t))$. This assumption is justified by the small angle approximation in our lateral planner (cf. (3.7) in Sec. 3.2.2). The maximum allowed lateral offsets d_i of each circle i are computed, under the constraint that no collisions with obstacle occupancies occur. Let $\text{circ}_i(d, t)$ denote the occupancy of circle $i \in \{1, 2, 3\}$ (cf. Sec. 3.2.2), which is shifted by d along the normal direction (we note the sign of d) from the ego pose at time t . The maximum lateral offset constraints are computed as

$$d_{i,\text{max}}(t) = \sup \left\{ d \geq 0 \mid \text{circ}_i(d, t) \cap \mathcal{O}_{\mathcal{B}}(t) = \emptyset \right\}. \quad (3.16)$$

The minimum lateral offset constraints $d_{i,\text{min}}(t)$ are obtained analogously for negative values of d as

$$d_{i,\text{min}}(t) = \inf \left\{ d \leq 0 \mid \text{circ}_i(d, t) \cap \mathcal{O}_{\mathcal{B}}(t) = \emptyset \right\}. \quad (3.17)$$

Fig. 3.9 illustrates the computation of the lateral constraints for the consecutive time steps t_1 and t_2 . If a circle initially intersects with an occupancy set for $d = 0$, the circle must be shifted to determine whether the ego vehicle should pass on the left or right. For instance, the circles for the minimum position constraints at t_2 in Fig. 3.9 are shifted in positive d -direction to pass occupancy $\mathcal{O}_j(t_2)$ on the left. In Sec. 3.4, we present two approaches to determine the passing sides.

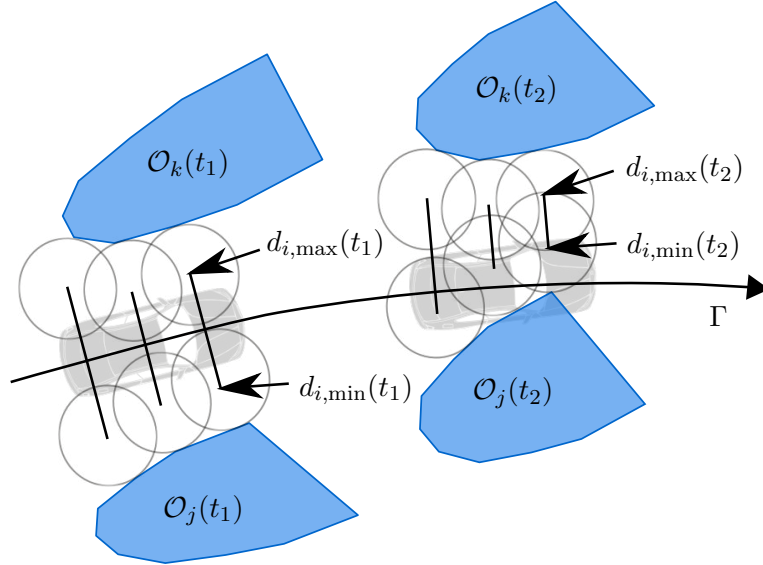


Figure 3.9: Illustration of lateral collision constraints. The minimum and maximum position constraints, $d_{i,\min}$ and $d_{i,\max}$, for each circle i are computed based on the given occupancy sets \mathcal{O}_k and \mathcal{O}_j . The longitudinal position of the ego vehicle is given by the planned longitudinal trajectory. ©2020 IEEE.

In Step 6, we perform a pre-solve check of the lateral optimization problem by evaluating whether the condition $\exists t \in [t_0, t_h] : d_{\min}(t) > d_{\max}(t)$ holds. If this condition holds for a certain t , then there is no feasible solution of the lateral problem, since the lateral position constraints (3.9) have been violated. If the lateral planning problem becomes infeasible, the ego vehicle must use the previously computed fail-safe trajectory which remains safe by design (cf. Fig. 3.1). However, if the evasive maneuver option is feasible, we plan the lateral motion of the ego vehicle as described in Sec. 3.2.2. After combining the longitudinal and lateral motions, we check the feasibility of the combined motion and obtain the new valid fail-safe trajectory if it is feasible.

3.4 Exploration of Non-convex Search Spaces for Fail-Safe Solutions

In the previous section, we demonstrated how convex optimization can be used to plan comfortable fail-safe trajectories. In order to ensure collision freedom, the position constraints need to be exactly computed. However, the difficulty of the computation increases in more complex scenarios due to the non-convexity of the search space. The non-convexity of motion planning problems is mainly caused by obstacles in the environment, which partition the search space into different homotopy classes. Homotopy classes describe “sets of trajectories that can be transformed into each other by gradual bending and stretching without colliding

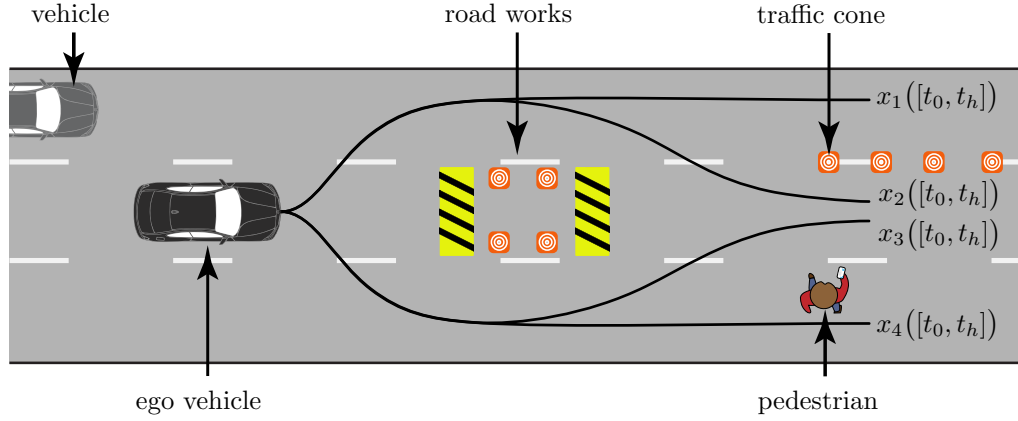


Figure 3.10: Non-convex search spaces. Obstacles in the environment partition the search space of the motion planning problem into different homotopy classes. Each illustrated trajectory $x_i([t_0, t_h])$, $i \in \{1, 2, 3, 4\}$, belongs to a distinctive homotopy class.

with obstacles” [201]. In essence, homotopy classes correspond to series of decisions on when and how to pass obstacles, such as on the left or right side [202–205] and are a crucial part to compute collision-free motions.

Fig. 3.10 illustrates the difficulty of these tactical decisions in a complex situation. If the ego vehicle passes the road works ahead on the left, it must also decide if the vehicle should pass first or not. However, when passing the road works on the right, the ego vehicle must also account for the crossing pedestrian. In this thesis, we denote the temporal orders of such tactical decisions as driving corridors. Driving corridors heavily influence the feasibility of the motion planning problem, which is particularly problematic in safety-critical situations, in which the vehicle must react in a timely manner to avoid a collision.

In the following sections, we present two different ways to obtain driving corridors for fail-safe solutions.

3.4.1 Enumerating possible driving corridors

In simple traffic scenarios, the passing side can be decided by trying different combinatorial sequences of decisions, since the fail-safe trajectory optimization is real-time capable. Therefore, for each obstacle $b \in \mathcal{B}$, we define a passing side $\gamma_b \in \{\triangleleft, \triangleright\}$ where \triangleleft and \triangleright denote passing the obstacle on the left or right side, respectively. Afterwards, we compute the lateral position constraints for each obstacle $b \in \mathcal{B}$. Let us first introduce $D_{i,b}(t) := \{d \mid \text{circ}_i(d, t) \cap \mathcal{O}_b(t) \neq \emptyset\}$ as the set of lateral positions of circle i that collide with obstacle b . Furthermore, $d_{i,b,\min}$ and $d_{i,b,\max}$ denote the minimum and maximum lateral position of the ego vehicle considering

3 Computationally Efficient Fail-Safe Trajectory Planning

obstacle $b \in \mathcal{B}$. If $\gamma_b = \triangleleft$, we choose

$$\begin{aligned} d_{i,b,\min}(t) &:= \sup D_{i,b}(t), \\ d_{i,b,\max}(t) &:= \infty, \end{aligned} \quad (3.18)$$

otherwise ($\gamma_b = \triangleright$), we choose

$$\begin{aligned} d_{i,b,\min}(t) &:= -\infty, \\ d_{i,b,\max}(t) &:= \inf D_{i,b}(t). \end{aligned} \quad (3.19)$$

Fig. 3.11 illustrates the collision-free lateral positions. Finally, we compute (3.16) as $d_{i,\max}(t) := \min\{d_{i,b,\max}(t) \mid b \in \mathcal{B}\}$ and (3.17) as $d_{i,\min}(t) := \max\{d_{i,b,\min}(t) \mid b \in \mathcal{B}\}$.

Naive approaches apply sampling [203, 206–208] or combinatorial enumerations [204, 209–211] to determine passing sequences for possible driving corridors. However, for a number of obstacles $n_{\mathcal{B}}$, there are already $2^{n_{\mathcal{B}}}$ combinatorial sequences and usually only a few sequences allow one to plan drivable trajectories. Thus, these approaches are usually applied in simple scenarios with a small number of traffic participants. In [9], we showed that certain sequences can be disregarded with prior knowledge of the traffic scenario and the optimal temporal sequence can be obtained by mixed-integer programming. Nevertheless, in convoluted solution spaces, the presented approaches often become intractable. In these situations, new fail-safe trajectories cannot be obtained and the ego vehicle has to execute the previous fail-safe trajectory.

In contrast, set-based reachability methods [212–217] are able to cope with arbitrarily complex solution spaces since their speed increases if the solution space becomes smaller. We develop a novel method that applies reachability analysis to efficiently compute suitable driving corridors for fail-safe trajectory planning. Therefore, we first introduce the drivable area as the set of all collision-free trajectories of the ego vehicle projected onto the position domain.

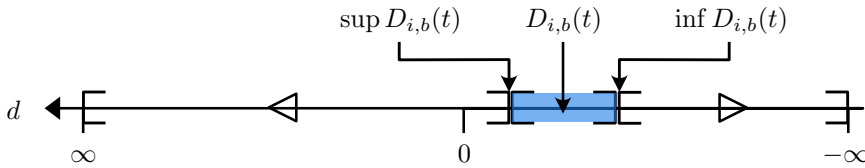


Figure 3.11: Lateral constraints and passing sides. The longitudinal position of the ego vehicle is given by the longitudinal trajectory and the reference path Γ . Thus, we only need to determine the minimum and maximum collision-free lateral offset d . For passing side $\gamma_b = \triangleleft$ (left), we obtain the collision-free interval $[\sup D_{i,b}(t), \infty]$ and for $\gamma_b = \triangleright$ (right), we obtain interval $[-\infty, \inf D_{i,b}(t)]$. $D_{i,b}(t)$ denotes the interval of lateral positions d which result in a collision with obstacle b .

3.4.2 Computing the drivable area of autonomous vehicles

We use the approach in [212, 213] to compute the collision-free forward reachable set (cf. Def. 5) at discrete points in time $t_k = k\Delta t, k \in \mathbb{N}_+$, with discrete time step size $\Delta t \in \mathbb{R}_+$. The dynamics of the ego vehicle are modeled as a double integrator system in a curvilinear coordinate system with bounded velocities and accelerations. States and inputs are modeled as $x_{\text{reach}} = (s, \dot{s}, d, \dot{d})^T$ and $u_{\text{reach}} = (a_{\text{lon}}, a_{\text{lat}})^T$, respectively. The dynamics are given by:

$$\frac{d^2}{dt^2}s(t) = a_{\text{lon}}(t), \quad \frac{d^2}{dt^2}d(t) = a_{\text{lat}}(t), \quad (3.20a)$$

$$\dot{s}_{\min} \leq \dot{s}(t) \leq \dot{s}_{\max}, \quad (3.20b)$$

$$\dot{d}_{\min} \leq \dot{d}(t) \leq \dot{d}_{\max}, \quad (3.20c)$$

$$a_{\text{lon},\min} \leq a_{\text{lon}}(t) \leq a_{\text{lon},\max}, \quad a_{\text{lat},\min} \leq a_{\text{lat}}(t) \leq a_{\text{lat},\max}. \quad (3.20d)$$

It should be noted that this dynamical model deviates from a real vehicle. It allows the ego vehicle to make turns with arbitrarily high velocities, since the curvature of the road is not incorporated. We compensate for this simplification by using conservative parameterizations (e.g., constraining the lateral velocities). Nevertheless, obtained trajectories are drivable, since we only use the drivable area to compute the position constraints; the trajectory is still optimized by the models introduced in Sec. 3.2.

We approximate the reachable set as the union of base sets $\mathfrak{B}_k^i = \mathcal{P}_{k,s}^i \times \mathcal{P}_{k,d}^i$, composed by the Cartesian product of two convex polytopes in the s - \dot{s} - and d - \dot{d} -plane [213]. Without loss of generality, the reachable set is computed with reference to the first circle (rear axle) of the vehicle shape approximation (cf. Sec. 3.2.2) and the initial set of states \mathcal{X}_0 . It should be noted that the reachable set can be computed for any reference point. The reachable set $\mathcal{R}_k = \bigcup_i \mathfrak{B}_k^i$ at time step $t_k, k > 0$, is obtained with the following steps [213]: first, base sets \mathfrak{B}_{k-1}^i (we note that $\mathcal{X}_0 \subseteq \mathfrak{B}_0^0$) of the previous time step $k - 1$ are propagated according to the system model (3.20). The propagated sets are denoted as $\mathfrak{B}_k^{\text{Pi}}$. The union $\bigcup_i \mathfrak{B}_k^{\text{Pi}}$ over-approximates the exact propagated reachable set at time step t_k . Second, unsafe states $\mathcal{X}_{\text{unsafe}}(t_k) := \{x_{\text{reach}} \mid \text{occ}(x_{\text{reach}}) \cap \mathcal{O}(t_k) \neq \emptyset\}$ are removed from the propagated base sets $\bigcup_i \mathfrak{B}_k^{\text{Pi}}$. For the collision check, we assume that the heading of the ego vehicle is given by the reference orientation $\theta_{\Gamma}(s(t))$ (cf. small angle approximation in Sec. 3.2.2) and use the vehicle shape approximation with three circles (cf. Sec. 3.2.2). Since the set $\bigcup_i \mathfrak{B}_k^{\text{Pi}} \setminus \mathcal{X}_{\text{unsafe}}(t_k)$ is usually non-convex, we under-approximate the result by a set of new base sets $\bigcup_i \mathfrak{B}_k^i$. The obtained base sets at each time step t_k are used to compute the drivable area of the ego vehicle as:

Definition 15 (Drivable Area) *The drivable area \mathcal{D}_k at time step t_k is defined as the projection of base sets \mathfrak{B}_k^i onto the position domain. The drivable area is represented as a set of axis-aligned rectangles: $\mathcal{D}_k = \bigcup_i \mathcal{D}_k^i$.*

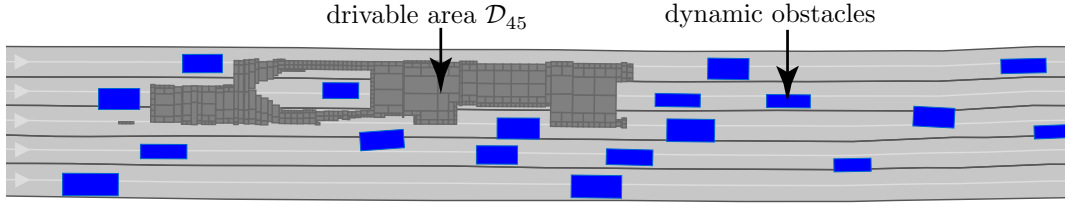


Figure 3.12: Visualization of the drivable area (USA_US101-6.1_T-1). The drivable area and the occupancies of obstacles are shown for $t_{45} = 4.5$ s.

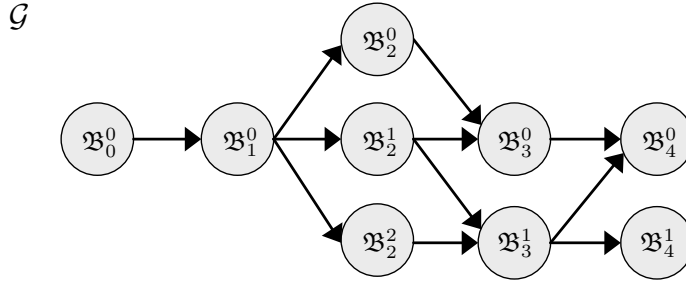


Figure 3.13: Reachability graph. The graph \mathcal{G} stores the reachability between base sets \mathfrak{B}_{k-1}^j and \mathfrak{B}_k^i for consecutive time steps t_{k-1} and t_k .

Fig. 3.12 visualizes the drivable area for a highway scenario of the CommonRoad benchmark suite and the time step $t_{45} = 4.5$ s.

During the computation of the reachable set, we create a graph \mathcal{G} to store information about the reachability between base sets (cf. Fig. 3.13). For instance, several sets \mathfrak{B}_k^j may be reachable in the next time step t_k from a set \mathfrak{B}_{k-1}^i at time step t_{k-1} . In \mathcal{G} , each node represents a base set \mathfrak{B}_k^j and edges represent the reachability between sets \mathfrak{B}_{k-1}^i and \mathfrak{B}_k^j of consecutive time steps.

3.4.3 Determining driving corridors using the drivable area

Our approach determines driving corridors for fail-safe trajectory planning by computing the drivable area of the ego vehicle. From the drivable area, we extract the position constraints for our longitudinal and lateral trajectory planners. To investigate the drivable area in figures, we plot trajectories with respect to center of the ego vehicle's rear axle. Fig. 3.14 illustrates the general procedure: in Step 1 (cf. Fig. 3.14b), we compute the drivable area of the ego vehicle for consecutive time steps. Based on the obtained sets and the reachability graph \mathcal{G} , we identify different driving corridors for the longitudinal motion in Step 2. Within the obtained driving corridor for the longitudinal motion, we optimize the longitudinal motion of the ego vehicle using the model in Sec. 3.2.1. Afterwards, we extract a driving corridor for the lateral motion in Step 3, followed by solving the lateral optimization problem using the model in Sec. 3.2.2 to obtain the final fail-safe trajectory.

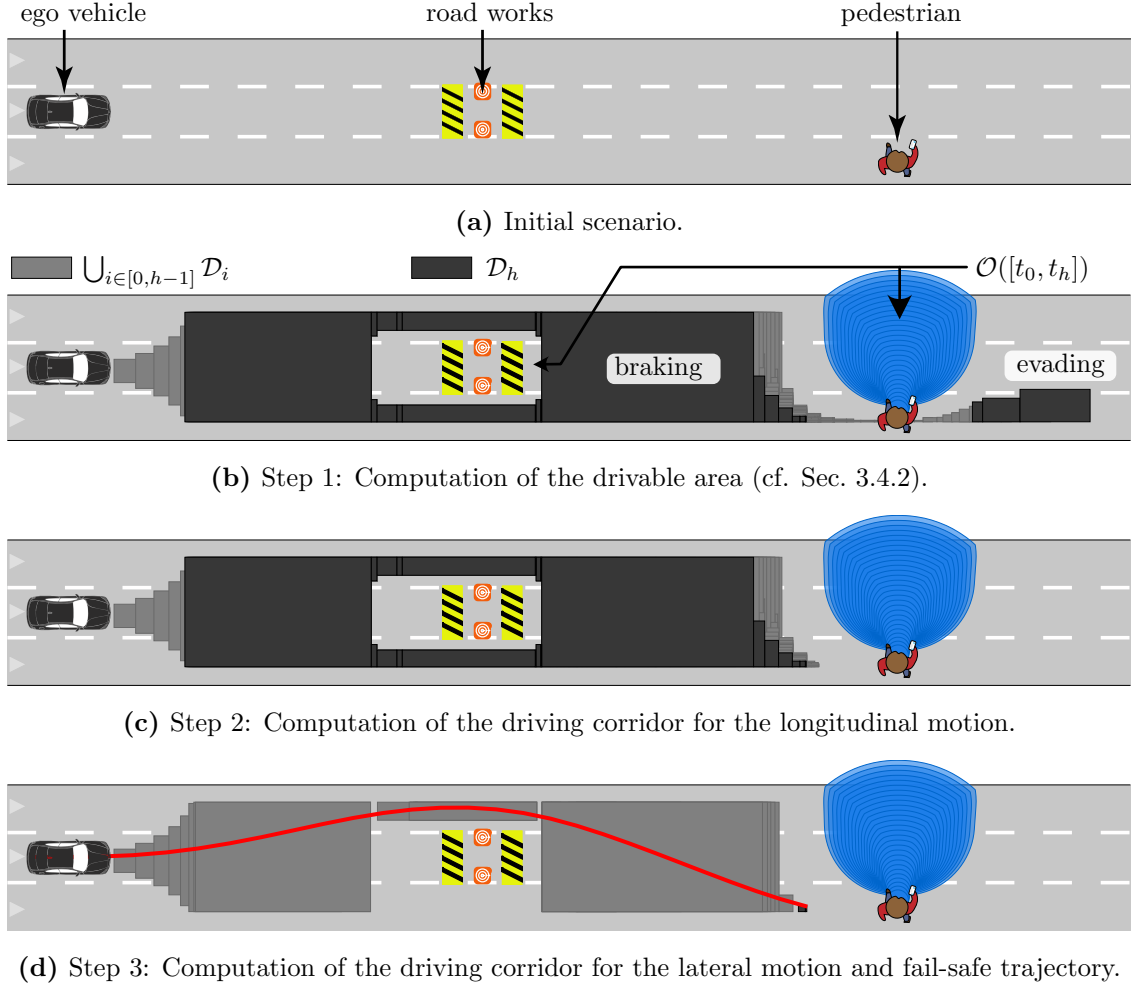


Figure 3.14: Fail-safe planning with driving corridors. (a) The considered traffic scenario. (b) The drivable area of the ego vehicle for the scenario. The ego vehicle can decide between a braking or an evading fail-safe maneuver. (c) The driving corridor for the longitudinal motion considering a braking maneuver. (d) The chosen driving corridor for the lateral motion and the optimized fail-safe trajectory.

Let us elaborate on the procedure to determine driving corridors in the drivable area in the following. Based on the created reachability graph \mathcal{G} , we identify driving corridors for the longitudinal motion by grouping base sets \mathfrak{B}_k^i according to their connectedness in the position domain at each time step t_k . Two sets are connected in the position domain if the intersection of their projection onto the position domain is not empty. For instance, the drivable area D_h at time step t_h in Fig. 3.14b contains two connected sets, which we have labeled as *braking* and *evading*. We require connected sets, since for non-connected sets we obtain position constraints which are not collision-free (cf. pedestrian in Fig. 3.14b). The set $\mathfrak{B}_{k,n}^{\text{CR}} := \{\mathfrak{B}_k^i, \mathfrak{B}_k^j, \dots\}$, $n \in \mathbb{N}$, denotes the n -th group of such connected base sets at

time step t_k (cf. Fig. 3.15a). In order to efficiently detect all pairs of connected base sets, we apply a sweep line algorithm [218]. The reachability between all connected sets $\mathfrak{B}_{k,n}^{\text{CR}}$ and $\mathfrak{B}_{k+1,l}^{\text{CR}}$ (cf. Fig. 3.15a) is stored in separate tree structures \mathcal{T}_{lon} and \mathcal{T}_{lat} for planning the longitudinal and lateral motion, respectively.

Our proposed procedure to identify driving corridors for the longitudinal and lateral motion is shown in Alg. 1 and visualized in Fig. 3.15a. Let us first elaborate on how to compute possible driving corridors for the longitudinal motion. Therefore, we introduce \bar{h} to denote a node within a tree structure, such as \mathcal{T}_{lon} for the corridor of the longitudinal motion. We create and add a new node $\bar{h}(\mathfrak{B}_{k,n}^{\text{CR}})$ for the connected sets $\mathfrak{B}_{k,n}^{\text{CR}}$ at time step t_k to \mathcal{T}_{lon} (cf. Alg. 1, line 1). Next, we insert an edge from the parent node $\bar{h}(\mathfrak{B}_{k-1,m}^{\text{CR}})$, $m \in \mathbb{N}$, to node $\bar{h}(\mathfrak{B}_{k,n}^{\text{CR}})$ into \mathcal{T}_{lon} (cf. Alg. 1, line 2). In order to detect the candidate connected components $\mathfrak{B}_{k+1,j}^{\text{CR}}$, $j \in \mathbb{N}$, of the next time step $k+1$ (cf. Alg. 1, line 7), we identify all reachable base sets from $\bar{h}(\mathfrak{B}_{k,n}^{\text{CR}})$ in the reachability graph \mathcal{G} (cf. Sec. 3.4.2, cf. Alg. 1, line 3). Subsequently, we determine all possible connected components for the reachable base sets at the next time step. Alg. 1 is recursively called and terminates as soon as all possible traces of connected components within the reachable set $\bigcup_k \bigcup_i \mathfrak{B}_k^i$ are found. Finally, we obtain the tree \mathcal{T}_{lon} which stores all information of (reachable) connected components that start from the initial time step and reach the final time step, visualized in Fig. 3.15a. Using \mathcal{T}_{lon} , we define the candidate driving corridors for the longitudinal motion as:

Definition 16 (Driving Corridor for the Longitudinal Motion) *A trace of connected components in \mathcal{T}_{lon} from the initial time step $k=0$ to the final time step $k=h$ constitutes a candidate driving corridor for the longitudinal motion $\Xi_{\text{lon}} := (\mathfrak{B}_{0,1}^{\text{CR}}, \mathfrak{B}_{1,i}^{\text{CR}}, \dots, \mathfrak{B}_{h,j}^{\text{CR}})$.*

The procedure for determining driving corridors for the lateral motion is similar. However, we only consider sets base sets \mathfrak{B}_k^i within the selected driving corridor

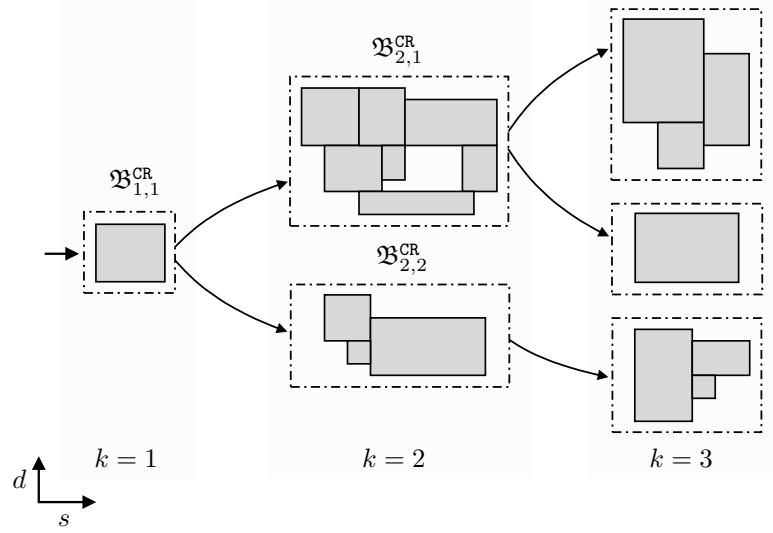
Algorithm 1 identifyCorridors

Input: Tree \mathcal{T} , connected base sets $\mathfrak{B}_{k,n}^{\text{CR}}$, parent node $\bar{h}(\mathfrak{B}_{k-1,l}^{\text{CR}})$,
 driving corridor Ξ_{lon} , longitudinal trajectory $x_{\text{lon}}([t_0, t_{\text{fs}}])$

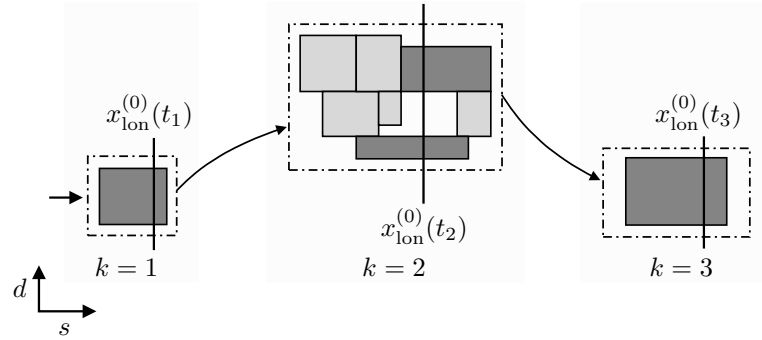
Output: Updated tree \mathcal{T}

- 1: $\mathcal{T}.\text{addNode}(\bar{h}(\mathfrak{B}_{k,n}^{\text{CR}}))$
- 2: $\mathcal{T}.\text{addEdge}(\bar{h}(\mathfrak{B}_{k-1,l}^{\text{CR}}), \bar{h}(\mathfrak{B}_{k,n}^{\text{CR}}))$
- 3: $\mathfrak{B}_{k+1} \leftarrow \mathcal{G}.\text{getChildren}(\mathfrak{B}_{k,n}^{\text{CR}})$
 // If Ξ_{lon} is provided, we determine a corridor for the lateral motion
- 4: **if** Ξ_{lon} is not empty **then**
- 5: $\mathfrak{B}_{k+1} \leftarrow \text{filterChildren}(\mathfrak{B}_{k+1}, x_{\text{lon}}([t_0, t_{\text{fs}}]))$
- 6: **end if**
- 7: **for** $\mathfrak{B}_{k+1,s}^{\text{CR}}$ in $\text{connectedSets}(\mathfrak{B}_{k+1})$ **do**
- 8: IdentifyCorridors(\mathcal{T} , $\mathfrak{B}_{k+1,s}^{\text{CR}}$, $\bar{h}(\mathfrak{B}_{k,n}^{\text{CR}})$, Ξ_{lon} , $x_{\text{lon}}([t_0, t_{\text{fs}}])$)
- 9: **end for**

for the longitudinal motion Ξ_{lon} during the computation of the driving corridor for the lateral motion, stored in \mathcal{T}_{lat} (cf. Alg. 1, line 5). Moreover, we filter all children sets \mathfrak{B}_k^i based on the longitudinal positions of the optimized trajectory $x_{\text{lon}}([t_0, t_{\mathfrak{F}}])$ with horizon $t_{\mathfrak{F}}$ as illustrated in Fig. 3.15b (cf. Alg. 1, line 5). Both selections are done since the longitudinal position of the ego vehicle is fixed by the longitudinal trajectory and thus, the ego vehicle is located in the selected driving corridor for the longitudinal motion; base sets outside of the corridor will lead to infeasible trajectories.



(a) Visualization of the tree \mathcal{T}_{lon} storing possible driving corridors for the longitudinal motion Ξ_{lon} and connected regions $\mathfrak{B}_{k,n}^{\text{CR}}$ at time steps k .



(b) Visualization of the selected driving corridor for the longitudinal motion Ξ_{lon} and the filtered base sets \mathfrak{B}_k^i (dark gray).

Figure 3.15: Identification of driving corridors. The driving corridor for the longitudinal and lateral motion Ξ_{lon} and Ξ_{lat} are obtained within the reachable set $\bigcup_k \bigcup_i \mathfrak{B}_k^i$. Sets \mathfrak{B}_k^i are shown as a projection onto the position domain.

Using the obtained tree structure \mathcal{T}_{lat} , we define the driving corridor for the lateral motion as:

Definition 17 (Driving Corridor for the Lateral Motion) *A trace of connected components in \mathcal{T}_{lat} , coinciding with the planned longitudinal motion, from the initial time step $k = 0$ to the final time step $k = h$ represents a possible driving corridor for the lateral motion $\Xi_{\text{lat}} := (\mathfrak{B}_{0,1}^{\text{CR}}, \mathfrak{B}_{1,q}^{\text{CR}}, \dots, \mathfrak{B}_{h,p}^{\text{CR}})$.*

After showing how driving corridors are obtained from the drivable area of the ego vehicle, we now demonstrate how we compute the position constraints from the corridors. Let us first introduce $\mathcal{D}_{k,p}^{\text{CR}}$ as the projection of the connected region $\mathfrak{B}_{k,p}^{\text{CR}}$ onto the position domain, yielding a subset of the drivable area \mathcal{D}_k at time step t_k . Considering the position constraints of the longitudinal motion (cf. (3.4)), we simply compute the minimum and maximum collision-free longitudinal position within the connected set at each time step t_k from the chosen driving corridor for the longitudinal motion Ξ_{lon} :

$$\begin{aligned} s_{\min}(t_k) &:= \inf \{s \mid (s, d)^T \in \mathcal{D}_{k,n}^{\text{CR}}, d \in \mathbb{R}\}, \\ s_{\max}(t_k) &:= \sup \{s \mid (s, d)^T \in \mathcal{D}_{k,n}^{\text{CR}}, d \in \mathbb{R}\}, \end{aligned} \quad (3.21)$$

where $\mathcal{D}_{k,n}^{\text{CR}}$ is the projection of the k -th component $\mathfrak{B}_{k,n}^{\text{CR}}$ in Ξ_{lon} .

The minimum and maximum admissible lateral deviation (3.9) of the ego vehicle from the reference path Γ are obtained from both Ξ_{lon} and Ξ_{lat} . Since the reachable set is computed with respect to the center of circle $i = 1$, we directly obtain values $d_{1,\min}(t_k)$ and $d_{1,\max}(t_k)$ from Ξ_{lat} (cf. Fig. 3.16):

$$\begin{aligned} d_{1,\min}(t_k) &:= \inf \{d \mid (s, d)^T \in \mathcal{D}_{k,p}^{\text{CR}}, s \in \mathbb{R}\}, \\ d_{1,\max}(t_k) &:= \sup \{d \mid (s, d)^T \in \mathcal{D}_{k,p}^{\text{CR}}, s \in \mathbb{R}\}, \end{aligned} \quad (3.22)$$

where $\mathcal{D}_{k,p}^{\text{CR}}$ is the projection of the k -th component $\mathfrak{B}_{k,p}^{\text{CR}}$ in Ξ_{lat} .

The lateral position constraints for circles $i \in \{2, 3\}$ are extracted from Ξ_{lon} (cf. Fig. 3.16), since the outer circles of the ego vehicle's shape have only been considered during the collision check but not in the base set computation as a reference point. To explain our approach, we remind that if we place the center of the ego vehicle's shape with orientation θ_Γ at any position within the drivable area, the ego vehicle is collision-free. In extreme cases, only the center of circle $i = 2$ is included in the drivable area and the two outer circles lie without the drivable area (cf. Fig. 3.16). First, let us introduce the tangential vector $\varrho(x_{\text{lon}}^{(0)}(t_k))$ of the reference path Γ at the longitudinal position $x_{\text{lon}}^{(0)}(t_k)$ and the center position $c^{(i)}$ of circle i in the world coordinate system. For each circle, we define the straight line $g_i((s, d)^T) := \varrho(x_{\text{lon}}^{(0)}(t_k))((s, d)^T - c^{(i)})$ for which $g_i((s, d)^T) = 0$ holds (cf. Fig. 3.16). We use $g_i((s, d)^T)$ to determine states in d -direction which intersect with parts of the drivable area in Ξ_{lon} (cf. Fig. 3.16). Therefore, we define the set of positions $(s, d)^T$ in $\mathcal{D}_{k,n}^{\text{CR}}$ (projection of $\mathfrak{B}_{k,n}^{\text{CR}}$ in Ξ_{lon}) that intersect with $g_i((s, d)^T) = 0$ (cf. Fig. 3.16) as:

$$\mathcal{Y}_i := \{(s, d)^T \in \mathcal{D}_{k,n}^{\text{CR}} \mid g_i((s, d)^T) = 0\}, \quad (3.23)$$

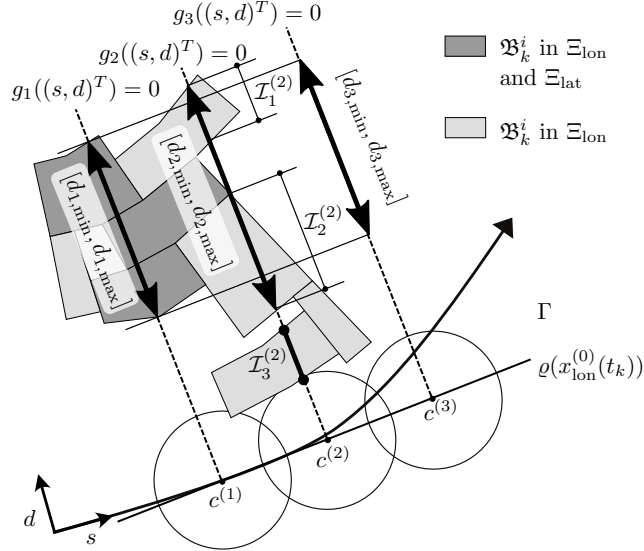


Figure 3.16: Lateral constraints from driving corridors. Projection of base sets \mathfrak{B}_k^i of driving corridors Ξ_{lon} and Ξ_{lat} onto the position domain. The minimum and maximum constraints $d_{i,\text{min}}$, $d_{i,\text{max}}$ are obtained through Ξ_{lon} and Ξ_{lat} .

where $\mathcal{D}_{k,n}^{\text{CR}}$ is the projection of the k -th component $\mathfrak{B}_{k,n}^{\text{CR}}$ in Ξ_{lon} .

If $\mathcal{Y}_i = \emptyset, i \in \{2, 3\}$ (e.g., circle $i = 3$ in Fig. 3.16), we set the constraints $d_{i,\text{min}}(t_k) := d_{1,\text{min}}(t_k)$ and $d_{i,\text{max}}(t_k) := d_{1,\text{max}}(t_k)$. This is possible, since we assume that the ego vehicle's heading is compliant with θ_Γ of the reference path Γ (cf. Sec. 3.4.2) during the reachable set computation. This assumption is also used during the collision check for removing colliding states from the reachable set. In essence, if we move the shape of the ego vehicle along $g_1((s, d)^T) = 0$ in Fig. 3.16 (since the longitudinal position is fixed), the front circle $i = 3$ is always collision-free.

However, if $\mathcal{Y}_i \neq \emptyset, i \in \{2, 3\}$ (e.g., circle $i = 2$ in Fig. 3.16), we can even enlarge the lateral constraints. This enlargement is possible, since the proposed reachable set computation ensures that a circle with radius r is collision-free for all states in $\mathfrak{B}_{k,n}^{\text{CR}}$. We compose the intersection points of $\mathcal{D}_{k,n}^{\text{CR}}$ and $g_i((s, d)^T) = 0$ with intervals $\mathcal{I}_q^{(i)}$ as illustrated in Fig. 3.16. Therefore, we introduce $\mathcal{I}^{(i)}$ as the set of valid intervals $\mathcal{I}_q^{(i)}, q \in \mathbb{N}$ for which $[d_{1,\text{min}}(t_k), d_{1,\text{max}}(t_k)] \cap \mathcal{I}_q^{(i)} \neq \emptyset$ holds. As depicted in Fig. 3.16, intervals $\mathcal{I}_1^{(2)}$ and $\mathcal{I}_2^{(2)}$ are considered as valid intervals, whereas $\mathcal{I}_3^{(2)}$ is not. The lateral deviation limits for the two circles $i \in \{1, 3\}$ are computed as:

$$[d_{i,\text{min}}(t_k), d_{i,\text{max}}(t_k)] := [d_{1,\text{min}}(t_k), d_{1,\text{max}}(t_k)] \cup \bigcup_{\mathcal{I}_q^{(i)} \in \mathcal{I}^{(i)}} \mathcal{I}_q^{(i)}. \quad (3.24)$$

3.5 Numerical Experiments

In the following numerical experiments, we evaluate the approaches presented in this chapter. We implement the longitudinal and lateral planners as well as the drivable area computation partly in *Python* and *C++* (for computational efficiency). For the experiments in Sec. 3.5.1 to 3.5.3, we use a computer with an Intel i5 1.4GHz processor and 8GB of DDR3 1600 MHz memory. In the experiments described in Sec. 3.5.6 and 3.5.5, we employ a computer with an Intel i7 2.6GHz processor and 16GB of DDR3 1866 MHz memory. We use \mathfrak{F} to denote a fail-safe trajectory. The vehicle models of the lateral and longitudinal planners are discretized with step size Δt to construct the optimization by assuming a constant input for each discrete time step $k \in [1, N_{\mathfrak{F}}]$ over the time horizon $t_{\mathfrak{F}}$. We utilize the convex programming packages *CVXPY* [219] and *CVXGEN* [220], as well as the solver *ECOS* [191]. In the experiments in Sec. 3.5.1 to 3.5.3, we use the combinatorial approach to determine possible driving corridors (cf. Sec. 3.4.1). The parameters of each scenario are given in App. A.3. A video of the presented simulations can be found attached to this thesis (cf. App. A.9).

3.5.1 Cut-in vehicles on highways

In our first scenario, we demonstrate how the proposed fail-safe planning approach computes a fail-safe trajectory that lets the ego vehicle swerve into another lane to avoid a collision. Hence, we consider a two-lane highway scenario, as illustrated in Fig. 3.17a, in which the ego vehicle plans an intended motion considering the most likely trajectories of the five surrounding vehicles. However, the ego vehicle might be endangered by a cut-in by the slower driving vehicle b_1 (parameters listed in

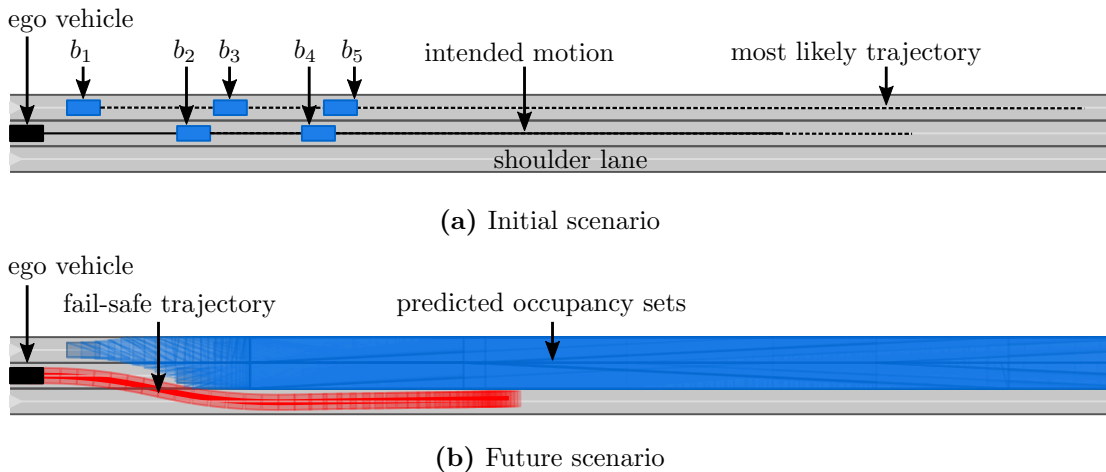
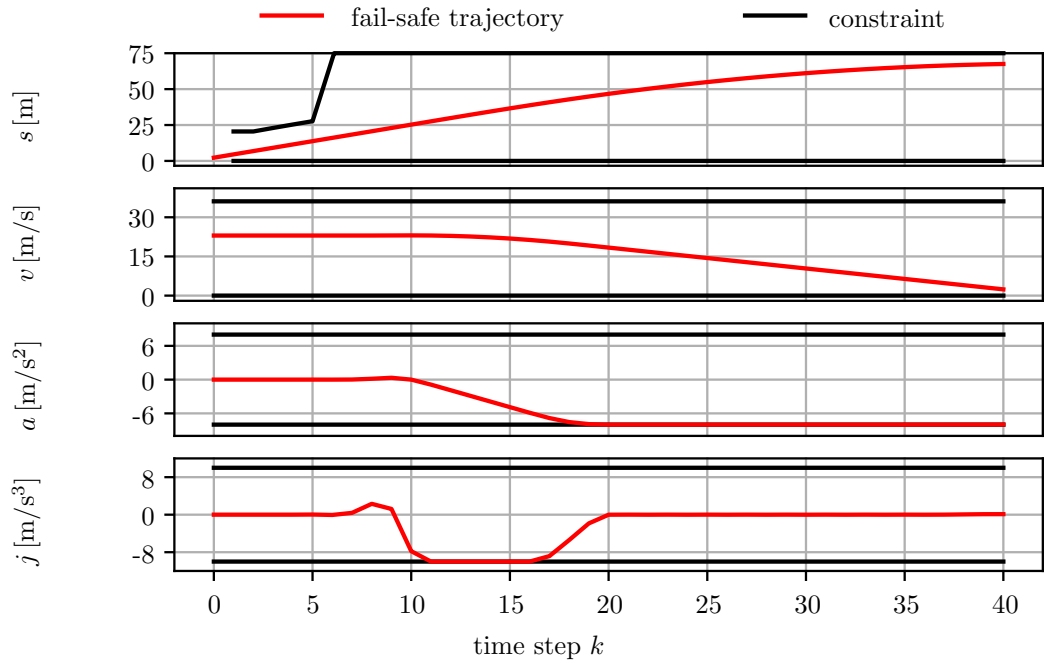
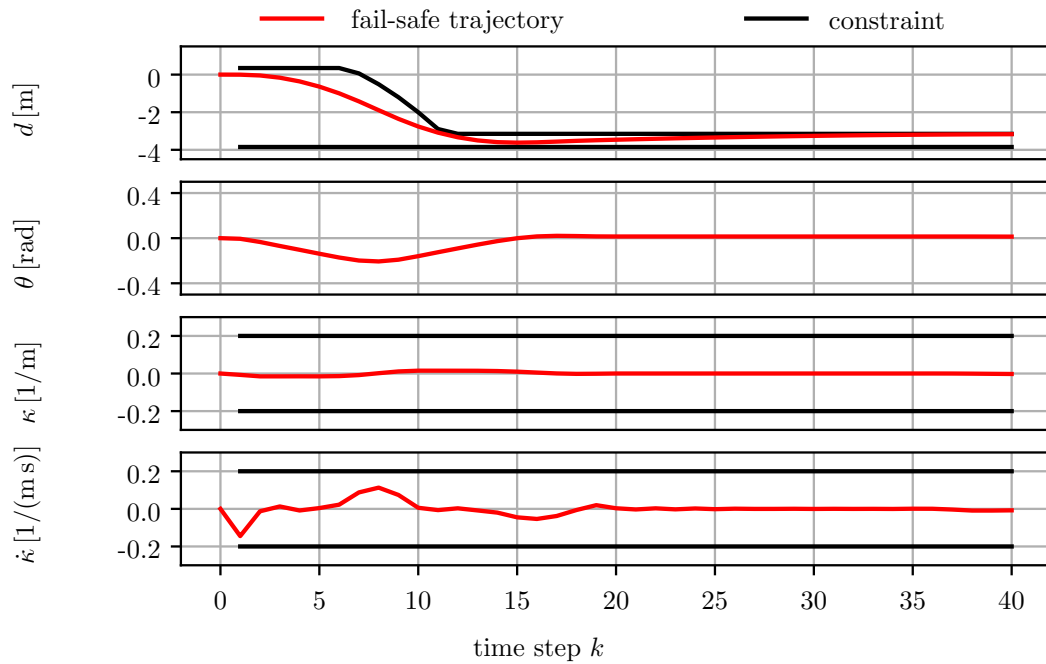


Figure 3.17: Highway scenario with cut-in vehicle (ZAM_HW-1.1.S-1). (a) Vehicle b_1 cuts into the ego vehicle’s lane. (b) The ego vehicle avoids a collision by swerving into the adjacent shoulder lane.



(a) Longitudinal motion



(b) Lateral motion

Figure 3.18: Planned fail-safe trajectory of the highway scenario. (a) The planned longitudinal motion and the considered constraints are shown for each state variable. (b) The lateral motion and constraints are shown for each state variable.

Tab. A.1). If vehicle b_1 changes to the ego vehicle’s lane, the ego vehicle cannot avoid a collision solely by braking, considering its initial velocity of $v_0 = 23$ m/s. Instead, the ego vehicle must perform a swerving maneuver into the adjacent shoulder lane.

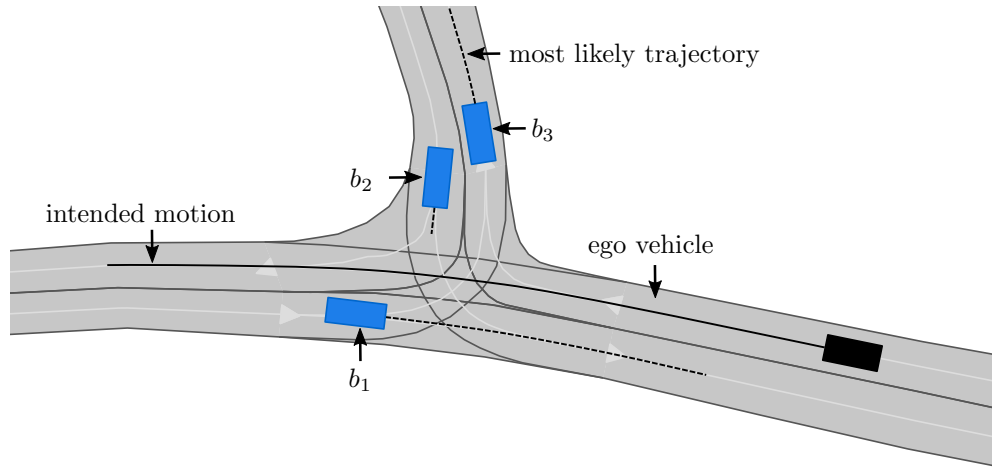
Fig. 3.18 visualizes the state variables of the planned fail-safe trajectory (represented as a red line) and the constraints (represented as a black line) over the optimization horizon of $N_{\mathfrak{F}} = 40$. Our approach automatically computes the position constraints of the longitudinal and lateral optimization problems. The obtained lateral collision constraints consider other obstacles, the left bound of the leftmost lane, as well as the right bound of the shoulder lane. It should be noted that the orientation is not constrained, since the ego vehicle may achieve any orientation depending on the maneuver. Considering these lateral position constraints, the solver is able to determine a feasible and collision-free fail-safe trajectory to the shoulder lane (cf. Fig. 3.17b). Even though this fail-safe trajectory involves swerving, our approach ensures that the obtained input trajectories are smooth and continuous. Thus, they are particularly well suited for tracking, since acceleration and curvature often serve as the control inputs within vehicle frameworks.

3.5.2 Urban T-junction

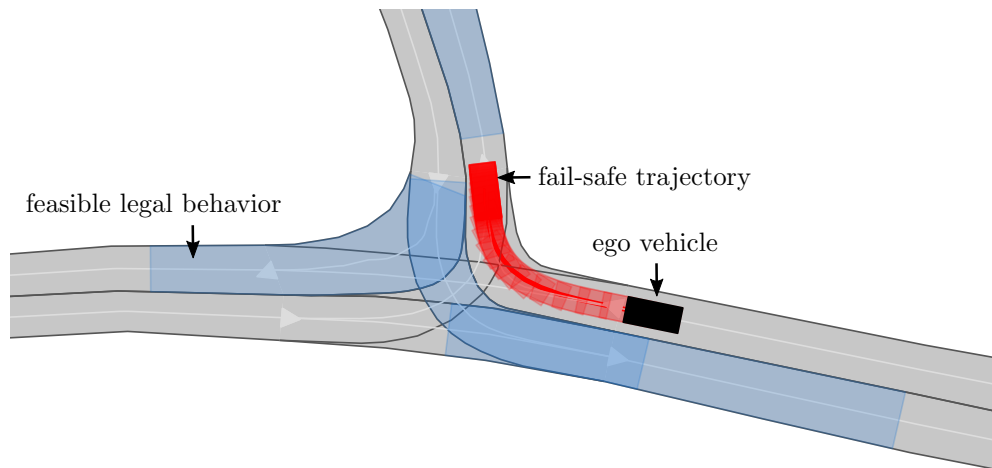
We now consider an urban environment (cf. Fig. 3.19a) in which the ego vehicle is approaching a T-junction along with three other vehicles $b_i, i \in \{1, 2, 3\}$ (parameters listed in Tab. A.2). The ego vehicle is traveling at a velocity of $v_0 = 8.3$ m/s. Since the ego vehicle is driving on a priority road, it plans a collision-free intended motion considering the most likely trajectories of all obstacles b_i in the scenario (cf. Fig. 3.19a).

However, if obstacle b_2 overlooks the ego vehicle (this kind of situation occurs regularly in real traffic) and as a result disrespects its right of way, the intended motion might end in a collision. Right of way rules are not yet considered in our legal safety specification. Based on the available free space, our approach ensures safety by computing a fail-safe trajectory (horizon of $N_{\mathfrak{F}} = 30$) that lets the ego vehicle turn right and come to a stop behind the occupancy set of b_3 (cf. Fig. 3.19b). This fail-safe trajectory starts at the last possible point in time along the intended motion (it should be noted that a braking maneuver without turning right needs to be executed earlier [64]).

Fig. 3.20 visualizes the planned longitudinal and lateral motion of the fail-safe trajectory (represented as a red graph) for each state variable. Our cost function allows the lateral planner to deviate from the reference Γ (centerline of the lane) to provide higher comfort while turning right (cf. Fig. 3.20b). Since we consider the maximal applicable braking acceleration in curves by computing the maximum longitudinal acceleration based on the curvature of the road (cf. (3.10)), we are able to guarantee the feasibility of the lateral motion.



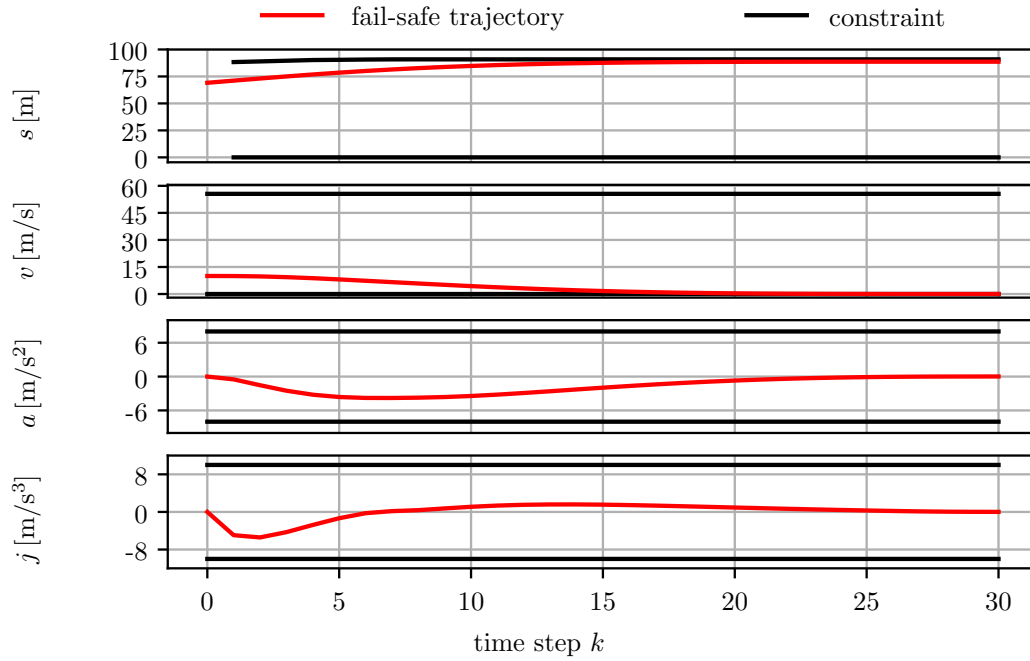
(a) Initial scenario



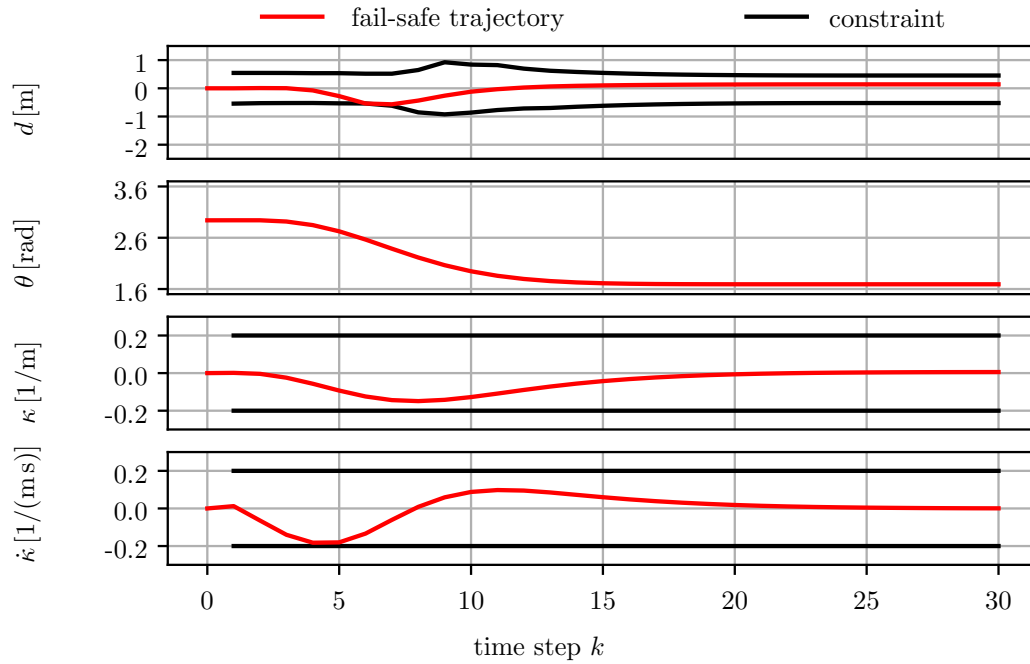
(b) Future scenario

Figure 3.19: Urban T-junction scenario (DEU_Ffb-2.2_S-1). (a) The ego vehicle is endangered by vehicle b_2 which violates the right of way of the ego vehicle. (b) The ego vehicle can avoid a collision by executing a combined braking and evasive maneuver to turn right. The predicted occupancies are shown for $t_{\mathfrak{F}} = 6$ s for clarity.

3 Computationally Efficient Fail-Safe Trajectory Planning



(a) Longitudinal motion



(b) Lateral motion

Figure 3.20: Planned fail-safe trajectory of the urban T-junction scenario. (a) The planned longitudinal motion and the considered constraints are shown for each state variable. (b) The lateral motion and constraints are shown for each state variable.

3.5.3 Avoiding collisions with crossing pedestrians

The following scenario demonstrates how the proposed fail-safe trajectory planning approach ensures safety in environments with vulnerable road users, such as pedestrians. In this scenario, the ego vehicle moves toward an intersection at a velocity of 13.8 m/s (cf. Fig. 3.21a). The parameters are given in Tab. A.3. A pedestrian approaches the lane of the ego vehicle at a velocity of $v_{0,\text{ped}} = 1.35$ m/s and the intention to cross it. The assumption management of the set-based prediction automatically detects that the pedestrian can no longer stop without entering the lane. The prediction considers maneuvers of the pedestrian such as stopping on the road, crossing the road perpendicularly or moving back to the sidewalk (detailed explanation in [12]). In contrast to the previous scenarios, our approach computes

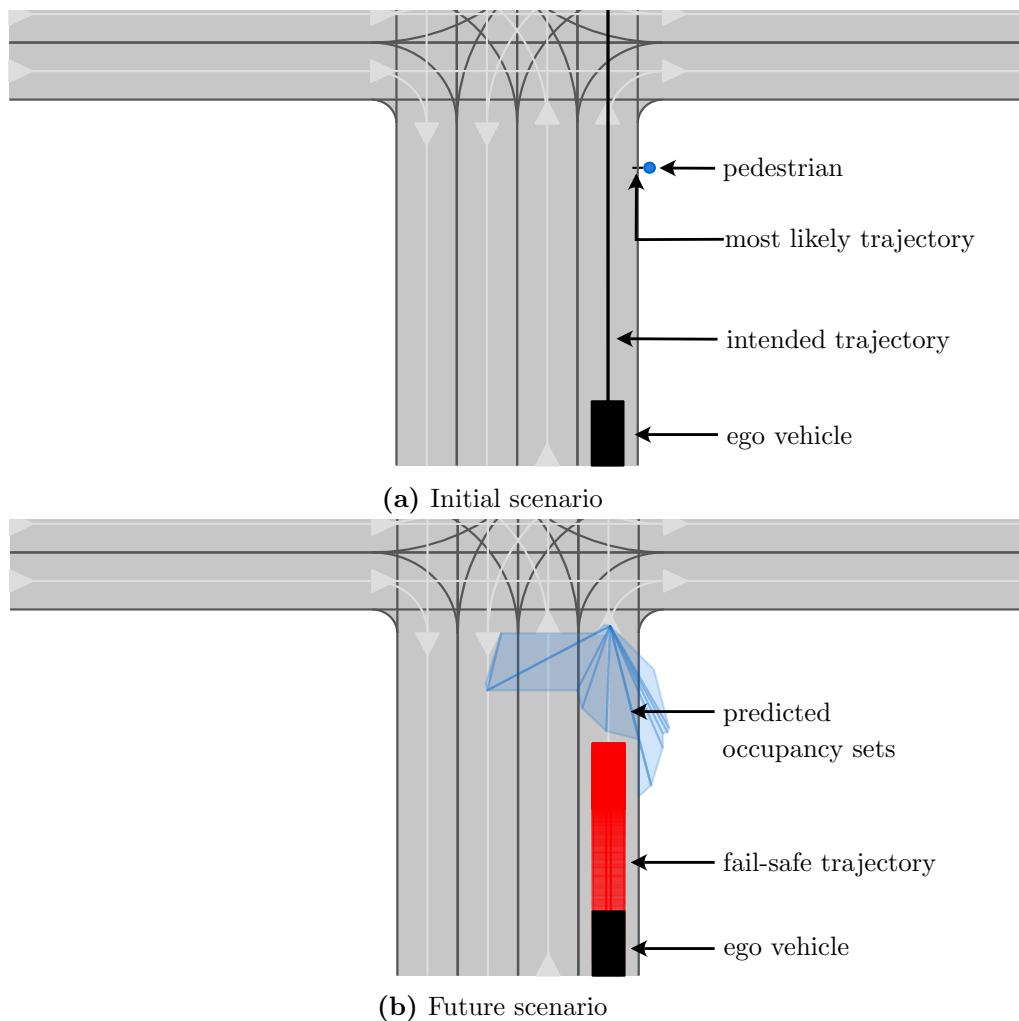
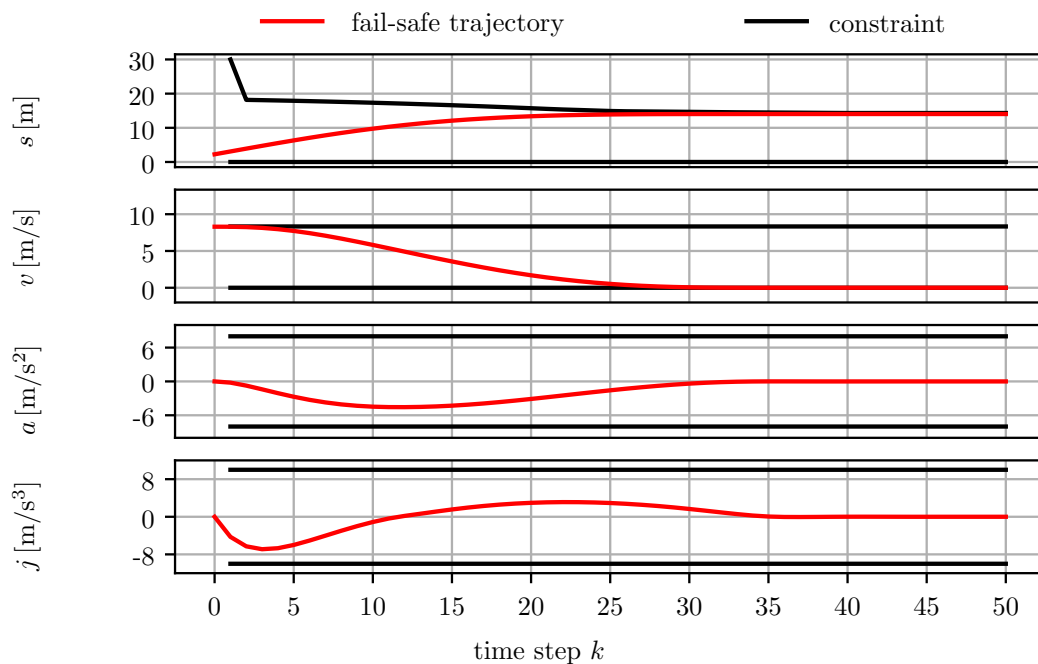
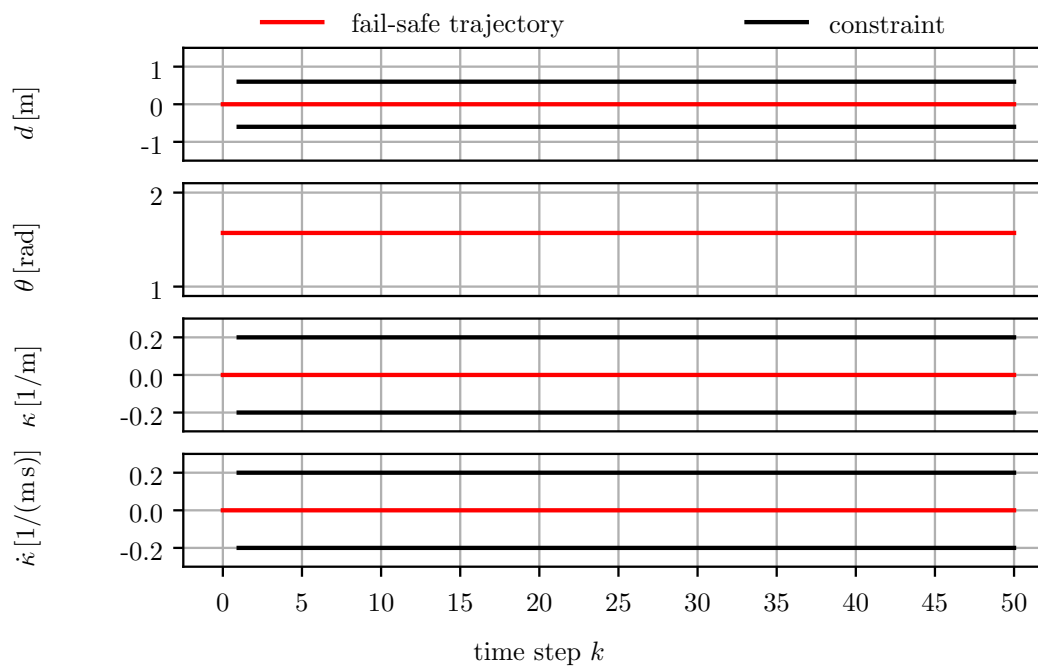


Figure 3.21: Urban scenario with crossing pedestrian (ZAM_Intersect-1.2.S-1-2). (a) The pedestrian crosses the path of the ego vehicle. (b) The ego vehicle can avoid a collision by performing an emergency braking maneuver.

3 Computationally Efficient Fail-Safe Trajectory Planning



(a) Initial scenario



(b) Future scenario

Figure 3.22: Planned fail-safe trajectory of the pedestrian scenario. (a) The planned longitudinal motion and the considered constraints are shown for each state variable. (b) The lateral motion and constraints are shown for each state variable.

a fail-safe trajectory that only involves emergency braking to avoid a collision (cf. Fig. 3.21b).

Fig. 3.22 visualizes the longitudinal and lateral motion of the planned braking maneuver for each state variable. The longitudinal planner computes a comfortable braking maneuver with a smooth deceleration profile (cf. Fig. 3.22a). Subsequently, the lateral planner computes a lateral motion that keeps the ego vehicle close to the reference path Γ .

3.5.4 Comparison with discrete planning approaches

To compare our approach to others proposed in the literature, we implement the popular sampling-based trajectory planner that uses quintic polynomials to minimize jerk [46] and a motion planner that uses motion primitives to plan trajectories [115]. Fig. 3.2 shows the planning results for a simple scenario in which a static obstacle is blocking the ego vehicle’s path.

Compared to our approach (cf. Fig. 3.2c), both discrete planning approaches have several disadvantages that limit their usage for fail-safe motion planning. For instance, a large number of motion primitives is usually required to obtain solutions in narrow solution spaces (cf. Fig. 3.2a). In our experiments, we use a database of 6.000 motion primitives, and larger databases increase the computation time of the search. In contrast, quintic polynomials can only produce trajectories with a sigmoidal shape (cf. Fig. 3.2b). Depending on the complexity of the fail-safe maneuver, multiple consecutive replanning phases may be required to obtain a fail-safe trajectory. For instance, to return to the initial lane in Fig. 3.2b, two consecutive fail-safe trajectories need to be sampled in our scenario. This process increases the required computation time, which may endanger the ego vehicle and its passenger in time-critical situations. Moreover, the feasibility check of each trajectory in the sampling-based planner requires a high smoothness of the reference path Γ . Since the states along a sampled trajectory are transformed from the curvilinear into the Cartesian world coordinate system using Γ (cf. closed-form transformations in [46]), inaccuracies and discontinuities (e.g., in the curvature of Γ) are directly transferred to the sampled trajectory, rendering them infeasible in the worst case. Due to the discretization, it is further not guaranteed that either discrete algorithms will obtain a solution, even though it may exist (cf. definition of completeness in [165, Ch. 5]).

Conversely, our approach directly obtains the optimal solution with global convergence. The feasibility of each trajectory is ensured through the incorporation of the kinematic model as a constraint in the optimization problem. As a result, we do not require reference paths with high smoothness. Moreover, we do not need to evaluate several trajectory candidates, but directly obtain the optimal solution instead. Since our fail-safe planner optimizes in a continuous search space, we do not suffer from discretization effects either. This property is particularly useful for fail-safe planning, since fail-safe trajectories are usually planned in narrow solution spaces (due to the growing occupancy sets over time).

3.5.5 Fail-safe planning with driving corridors

In our next experiment (cf. Fig. 3.23a), we exploit the proposed driving corridor computation to plan distinct fail-safe maneuvers. We create an artificial scenario to better visualize different maneuver classes. The parameters of this scenario are given in Tab. A.4. Initially, the ego vehicle is stopped (to increase the difficulty of finding a swerving maneuver) and a pedestrian is going to cross the ego vehicle’s path. The adjacent left lane is blocked by a static obstacle, increasing the difficulty of evading the pedestrian. By making use of our proposed approach, the ego vehicle can determine its maneuver options to avoid a collision with the two obstacles. Fig. 3.23b shows the two determined maneuver classes: stopping in front of the pedestrian or swerving. For each maneuver, we obtain the driving corridor and constraints for the convex optimization problems. The planned braking trajectory and the driving corridor for the lateral motion are shown in Fig. 3.23c. To compare the planning results with the computed drivable area, we plot trajectories with respect to center of the ego vehicle’s rear axle.

The fail-safe trajectory of the evasive maneuver is illustrated in Fig. 3.23d. It should be noted that the drivable area is shown at the discrete time steps k . Here, the ego vehicle first passes the pedestrian to the left and then evades the static obstacle by changing back to the initial lane. The solution space for evading in this scenario is small due to the crossing pedestrian. Fig. 3.23e shows the computed drivable area of the scenario at $t_{30} = 6.0$ s. Even though the solution space is small, the ego vehicle is able to pass the pedestrian. Thus, by considering the position constraints of the driving corridor for the longitudinal motion during the trajectory optimization, we are able to obtain a drivable evasive trajectory (cf. Fig. 3.23d).

3.5.6 Managing complex scenarios with small solution spaces

In our last scenario, we demonstrate that the computation of driving corridors is suitable to plan fail-safe trajectories in critical traffic situations where fast reactions are crucial to avoid collisions. Fig. 3.24a shows the initial position of the ego vehicle and other traffic participants in a five-lane highway scenario. The initial velocity of the ego vehicle is $v(t_0) = 16.8$ m/s and the average velocity of surrounding traffic participants is $v_{\mathcal{B}}(t_0) = 17.3$ m/s. In this experiment, we use the most likely prediction of other traffic participants as specified in the CommonRoad scenario. To increase the criticality of the scenario, we gradually raise the initial velocity $v(t_0)$ of the ego vehicle in steps of 1.4 m/s. Afterwards, we compute the reachable set for a time horizon of $t_h = 8$ s and compare the runtime for the reachable set computation. Fig. 3.24b illustrates the average computation time of 20 runs per scenario. The computation time of the approach decreases as the criticality of the traffic situation increases, since fewer sets \mathcal{B}_k^i have to be propagated and fewer collision checks have to be performed for these sets. For instance, when the initial velocity is almost doubled, the computation becomes two times faster. Thus, our approach is particularly suited for complex situations with small solution spaces.

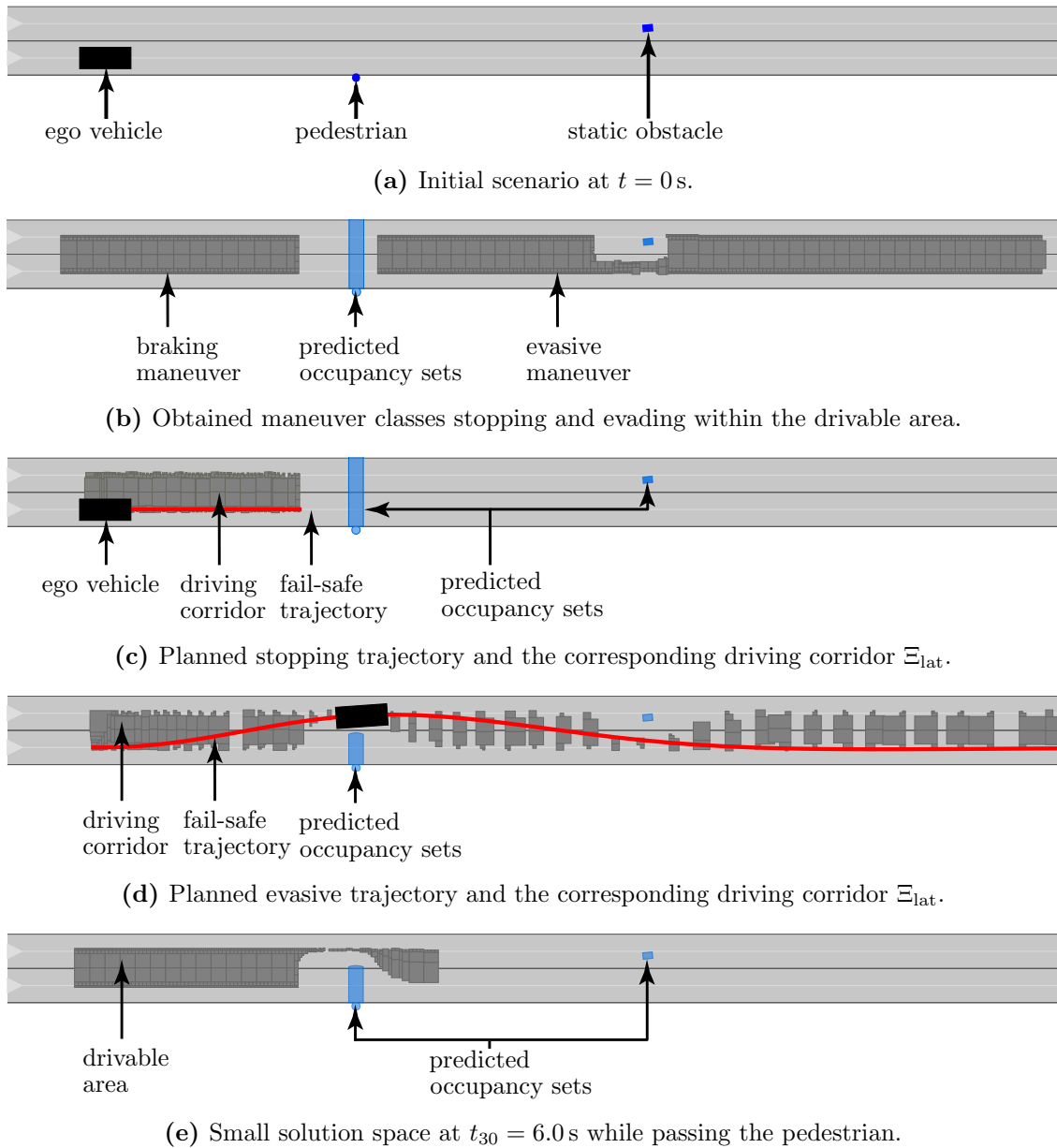


Figure 3.23: Scenario with distinct driving corridors. (a) A pedestrian is suddenly crossing the road. (b) The obtained fail-safe maneuvers are stopping in front of the pedestrian or passing the pedestrian on the left. (c) The braking fail-safe trajectory and the corresponding driving corridor for the lateral motion. (d) The planned evasive fail-safe trajectory and the driving corridor for the lateral motion. (e) The small solution space of the evasive maneuver. ©2020 IEEE.

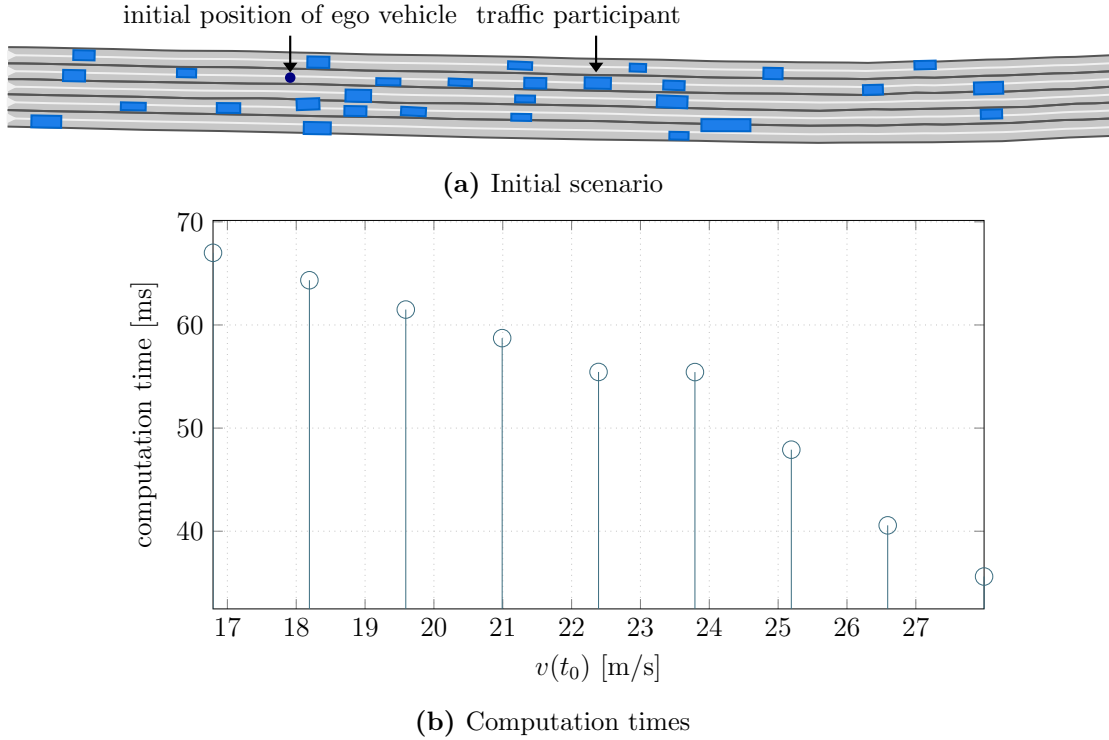


Figure 3.24: Scenario with a small solution space (USA_US101-6.1-T-1). (a) Initial highway scenario. (b) Decrease in the reachable set computation time as the initial velocity of the ego vehicle increases.

3.6 Summary

This chapter introduced fail-safe trajectory planning as a technique to ensure the safety of planned motions. After a brief overview of existing motion planning approaches, we presented a novel trajectory planning approach based on convex optimization. This convex formulation allows us to compute trajectories in real-time with global convergence, but it generally requires the separation of planned motions into a longitudinal and a lateral component. We use a fourth-order integrator and kinematic single-track model to ensure comfortable and collision-free longitudinal and lateral motions, respectively. Moreover, we demonstrated how slack variables can be used to plan trajectories with partly constant acceleration phases. This modification enhances comfort for passengers in various driving situations.

Based on the proposed trajectory planning formulation, we developed an approach to obtain fail-safe trajectories in arbitrary traffic scenarios. By considering the required lateral acceleration in the longitudinal planning problem, we are able to guarantee the feasibility of the resulting fail-safe trajectory despite the separation into longitudinal and lateral components. We demonstrated how the position constraints can be computed with respect to the predicted occupancy sets. To guide our convex fail-safe trajectory planner to solutions in complex search spaces,

we first presented a naive approach that uses combinatorial enumerations to determine driving corridors. Subsequently, we developed an approach to compute driving corridors based on the drivable area of the ego vehicle. We identify driving corridors by grouping subsets of the drivable area according to their connectedness in the position domain. The obtained driving corridors are used to constrain the trajectory optimization to plan collision-free motions in non-convex search spaces.

Lastly, we highlighted the benefits and capabilities of the proposed approaches in multiple numerical experiments. In scenarios with static and dynamic vehicles as well as vulnerable road users, such as pedestrians, we demonstrated that fail-safe trajectories ensure the safety of the ego vehicle with respect to predicted occupancy sets. In one of these scenarios, we showed that the computation time of the drivable area decreases when the criticality of the scenario increases. Furthermore, we exploited driving corridors to plan different fail-safe maneuvers in a complex traffic situation.

The proposed approaches pave the way for a novel verification technique that can be used during the operation of the vehicle. By computing fail-safe trajectories, the ego vehicle is empowered to ensure the safety of its intended motions before execution in just a few milliseconds. Since fail-safe trajectories are collision-free with respect to any feasible, legal behavior of obstacles, the ego vehicle always maintains a safe plan if other traffic participants behave differently than predicted. If the ego vehicle cannot compute a new fail-safe trajectory for a new planned intended trajectory, the previously computed fail-safe trajectory remains safe by design. Even if the ego vehicle has to execute a fail-safe trajectory, passengers do not have to compromise their comfort. Fail-safe trajectories are jerk-optimal when switching from the intended trajectory and are optimized for comfort over the whole time horizon. The developed driving corridor approach allows the ego vehicle to compute fail-safe trajectories even in highly complex scenarios with small solution spaces in a reasonable time.

Although fail-safe trajectories are collision-free against the predicted occupancy sets, the safety of the ego vehicle is only guaranteed over the planning horizon of the fail-safe trajectory. After that, the ego vehicle might potentially collide with another traffic participant. To ensure that the ego vehicle remains safe even after the planning horizon, fail-safe trajectories need to end in a set of safe states. In the following chapter, we address the problem of guaranteeing safety for infinite time horizons by introducing invariably safe sets.

4 Invariably Safe Sets for Infinite Time Horizon Planning

In this chapter, we propose invariably safe sets as a technique to compute safe states for autonomous vehicles and to verify the safety of fail-safe trajectories for infinite planning horizons. Sec. 4.1 introduces infinite time horizon planning and briefly reviews existing definitions of safe states for autonomous systems. Then, after defining invariably safe states in Sec. 4.2, we present how an under-approximation of invariably safe sets can efficiently be computed in Sec. 4.3. Subsequently, Sec. 4.4 demonstrates how invariably safe sets can be used for the safety assessment of trajectories, for example to determine whether a given trajectory is safe for an infinite time horizon. The benefits of the proposed approaches are demonstrated in different numerical experiments in Sec. 4.6. The chapter finishes with conclusions in Sec. 4.7. The content of this chapter is mainly based on the publications [7, 8, 10–12, 15].

4.1 Introduction and State of the Art

Motion planners typically plan trajectories for finite planning horizons, such as partial motion planning [221] or receding horizon control [222]. This simplification is often motivated by computational efficiency or the uncertain evolution of traffic scenarios for large time horizons. However, the length of the considered planning horizon plays a crucial role in achieving safe motions for autonomous vehicles. For instance, trajectory planning with short planning horizons may cause stability problems and not maintain persistent feasibility. In this chapter, we focus on persistent feasibility - that is, ensuring that the trajectory planning problem is recursively feasible without violating collision constraints.

To illustrate the importance of the planning horizon for persistent feasibility, we consider the traffic situation shown in Fig. 4.1. The ego vehicle plans two trajectories, $x_1([t_0, t_h])$ and $x_2([t_0, t_h])$, which are both collision-free for the considered planning horizon $t_{\text{horizon}} = t_h - t_0$ and end in states with velocities $v \gg 0$. However, trajectories with finite horizons may result in collisions directly after the horizon. For instance, planning a new collision-free motion starting at the final state $x_1(t_h)$ of trajectory $x_1([t_0, t_h])$ is infeasible, since the velocity $v \gg 0$ is too large to avoid a collision with the road works ahead. On the other hand, persistent feasibility is ensured for trajectory $x_2([t_0, t_h])$ from its final state $x_2(t_h)$. Simply increasing

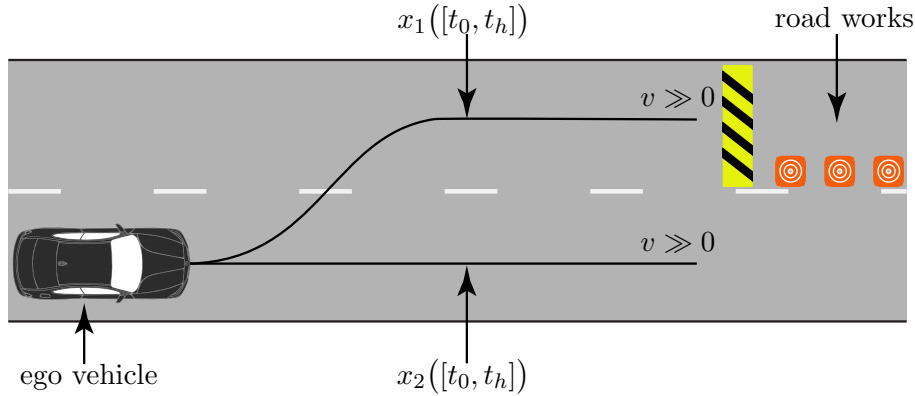


Figure 4.1: Safety problem of finite planning horizons. The ego vehicle plans two trajectories, $x_1([t_0, t_h])$ and $x_2([t_0, t_h])$, which end in states with high velocities $v \gg 0$. Both trajectories are collision-free in the considered finite planning horizon $t \in [t_0, t_h]$. However, only trajectory $x_2([t_0, t_h])$ remains safe beyond the finite horizon, since the vehicle can plan a feasible trajectory starting at $x_2(t_h)$. In contrast, trajectory $x_1([t_0, t_h])$ inevitably leads to a collision with the road works.

the planning horizon - choosing $t'_{\text{horizon}} > t_{\text{horizon}}$ - is also no remedy, since finite horizons in general may lead to inevitable collisions.

Many approaches circumvent the problem of finite planning horizons by ensuring that planned trajectories are collision-free within the finite planning horizon and end in a given set of safe states that allow persistent feasibility [31, 46, 62]. However, we cannot assume that such safe sets are provided, which raises the question of how to define and efficiently compute safe states for autonomous vehicles. Unfortunately, this question has not yet been adequately answered. To compute safe states, the autonomous vehicle has to 1) consider its own dynamics, 2) account for the future behavior of obstacles in the environment, and 3) reason over an infinite time horizon [223]. Applying these three requirements to the motion planning of autonomous vehicles is challenging.

Various governmental institutions around the world have also identified the issue of unsatisfactory definitions of safe states for the domain of autonomous vehicles [35]. Legislative powers have already tried to specify safety requirements for developing and testing autonomous vehicles, but they have clarified that defining safe states for motion planning is still an open problem that urgently needs to be solved [224, p. 13]. Particularly in emergency situations, the autonomous vehicle must be able to determine safe states in a timely manner to avoid endangering human lives.

Over the years, many different approaches have been proposed to deal with infinite horizons, such as linear-quadratic regulators [186, 225–227], Lyapunov stability [228–230], receding horizon control [231], Markov decision processes with

infinite horizon objectives [232–234], linear temporal logic [235, 236], and machine learning [237–240]. In the following sections, we extensively review the two infinite horizon techniques most relevant to our developed invariably safe sets: inevitable collision states and control invariant sets.

4.1.1 Inevitable collision states

Trajectories that do not end in an inevitable collision state (ICS) allow persistent feasibility for an infinite time horizon. ICSs are states in which, regardless of the followed trajectory, the ego vehicle eventually collides with an obstacle [241]. We formally define an ICS as:

Definition 18 (Inevitable Collision State) *A given state x at time t_0 is called an inevitable collision state (ICS) if $\forall u([t_0, \infty)) : \exists t \geq t_0 : \text{occ}(\chi(t, x, u([t_0, t]))) \cap \mathcal{O}(t) \neq \emptyset$.*

Sets of ICSs are denoted as regions of inevitable collision (RICs) $\mathcal{X}_{\text{RIC}} \subset \mathcal{X}$. In [242], the complementary concepts of regions of potential collision (RPCs) and regions of near collision (RNCs) are proposed. RPCs are sets $\mathcal{X}_{\text{RPC}} \subset \mathcal{X}$ that contain states for which only a small set of trajectories do not lead into an RIC [243]. Therefore, planners must be precise enough to obtain trajectories in small solution spaces. On the other hand, RNCs are sets $\mathcal{X}_{\text{RNC}} \subset \mathcal{X}$ that contain states that lead to an RIC if the robot does not change its current motion in a certain time frame [242, 243].

The computation of ICSs (or RICs) is often intractable for robots in uncertain environments. The approaches in [244–247] provide algorithms to check whether a given state is an ICS. In [248, 249], reachability analysis is used to determine RICs. The probability of states being part of RICs is assessed in [250, 251]. To reduce computational effort, one can also check whether a state allows the robot to remain collision-free by considering a subset of all possible trajectories [241, Prop. 4], such as braking maneuvers. As a result, one obtains a superset of RICs, since states within the superset may be ICSs, or the collision-free trajectory is not enclosed in the considered subset of trajectories. The approaches in [143, 144, 221, 252] obtain these supersets for a set of selected evasive maneuvers. However, determining ICSs is computationally intense, and most works can only handle a single trajectory prediction of traffic participants for online computation. Consequently, ICSs also suffer from the uncertain future motion of obstacles.

Fig. 4.2a illustrates the differences between RICs, RPCs, and RNCs in an example scenario. In this scenario, the ego vehicle approaches a traffic jam (distance 40 m) at a velocity of 10 m/s (other vehicles are at a standstill). We use the set of possible braking trajectories with constant control input to compute RICs efficiently in this scenario, similar to [221]; computing RICs for the set of all trajectories is intractable. In our example, we parameterized RPC to contain states for which less than 10% of feasible braking trajectories are able to avoid a collision - in other words, only trajectories with accelerations $a \in [a_{\min}, 0.9a_{\min}]$ are collision-free, where $a_{\min} < 0$ is the minimum feasible acceleration of the ego vehicle. On the

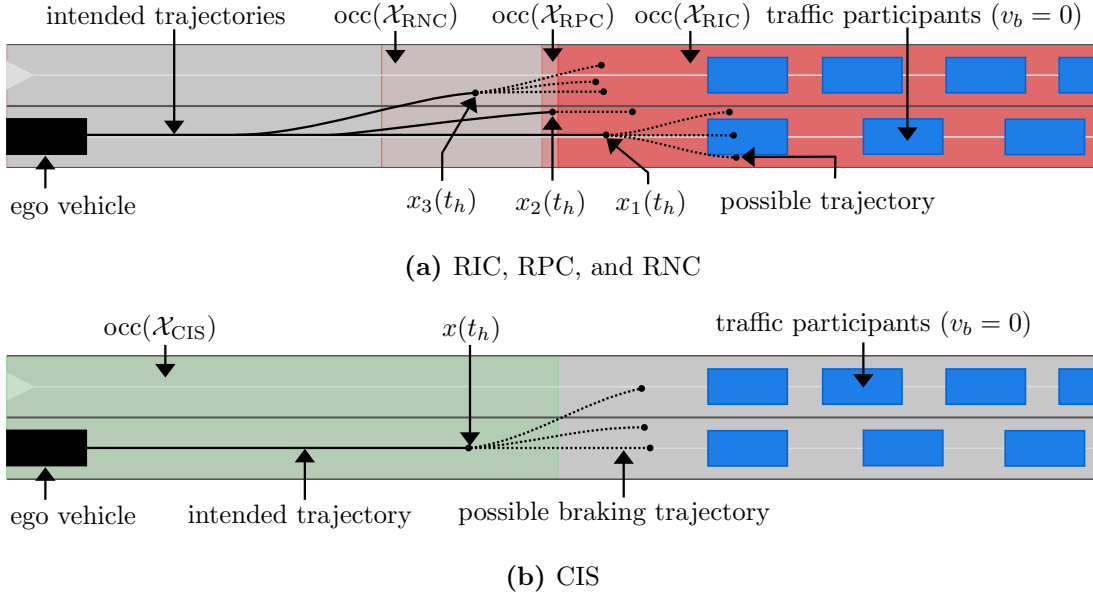


Figure 4.2: Illustration of ICS and CIS in a traffic jam scenario, in which the ego vehicle moves at a constant velocity. (a) In the ICS concept, intended trajectories that end in RIC eventually lead to a collision after the horizon (e.g., $x_1(t_h)$). Trajectories ending in RPC can only be continued with a small set of collision-free solutions (here, less than 10% of braking trajectories) after t_h (e.g., $x_2(t_h)$). Conversely, trajectories ending in RNC require a fast reaction to avoid causing a collision (here, less than 1 s) (e.g., $x_3(t_h)$). (b) Trajectories that end in a CIS can be continued collision-free for an infinite time horizon (here, by coming to a stop with braking trajectories). All sets are shown for time t_h as a projection onto the position domain.

other hand, RNCs are computed such that the vehicle needs to execute a braking maneuver in less than 1 s to avoid collisions.

4.1.2 Control invariant sets

In contrast to ICSs, controlled invariant sets (CISs) guarantee persistent feasibility. For every state within a CIS, there exists at least one feasible trajectory that keeps the autonomous vehicle within the CIS for an indefinite amount of time [253, 254]. As a result, the ego vehicle is able to determine feasible trajectories at all times and thus to remain safe for an infinite time horizon. Classical definitions of a CIS, such as that presented in [253], usually do not consider dynamic environments. To ensure safety, we adapt the CIS definition in a similar way to [255] to consider dynamic obstacles:

Definition 19 (Control Invariant Set) *A set \mathcal{X}_{CIS} is called a control invariant set (CIS) if $\forall x \in \mathcal{X}_{\text{CIS}} : \exists u([t_0, t]) : \forall t \geq t_0 : \text{occ}(\chi(t, x, u([t_0, t]))) \cap \mathcal{O}(t) = \emptyset$.*

In terms of persistent feasibility, RIC and CIS are related as $\mathcal{X}_{\text{CIS}} = \mathcal{X} \setminus \mathcal{X}_{\text{RIC}}$, where \mathcal{X} is the set of feasible states.

In [255–260], CISs are applied to motion planning of various autonomous systems. They are also well suited for safety verification. For instance, CISs are used to verify the safety of unmanned aerial vehicles (UAVs) [261, 262]. The CIS of UAVs consists of special steady state maneuvers, called Loiter circles. The safety of the UAV is guaranteed if it can execute this steady state maneuver at any time in the motion plan. In combination with reachability analysis, CISs for autonomous vehicles are used to verify the safety of adaptive cruise control systems in [86, 263] or for predictive threat assessment in [112]. CISs have also been applied to safe controller design [264, 265].

Fig. 4.2b illustrates a CIS in an example scenario, in which the ego vehicle approaches a traffic jam (vehicles are at a standstill). If a trajectory ends in the CIS, the ego vehicle is definitely able to obtain a feasible braking maneuver to avoid collisions. The CIS is computed by forward simulation of braking maneuvers to avoid collisions with the traffic jam [143]. However, determining invariant sets is computationally costly, and existing CIS approaches mainly work in static environments [262]. In addition, computing approximations of a CIS is usually difficult in dynamic environments [255], since $\mathcal{O}(t)$ is unknown for an infinite time horizon (cf. Def. 19). Nevertheless, applying invariant sets to ensure feasibility is promising, as they can guarantee safety for an infinite time horizon by definition. To overcome the limitations of CISs, we propose invariably safe sets. With these sets, we are able to efficiently guarantee the persistent feasibility of trajectories in uncertain, dynamic environments.

4.2 Invariably Safe States

In terms of motion planning, we are particularly interested in finding (collision-free) states that allow the autonomous vehicle to remain collision-free for an infinite time horizon. We define such safe states by making use of recursion: we denote a state as safe if a collision-free trajectory to another safe state exists. This recursive definition allows us to derive subsets of the set of collision-free states $\mathcal{F}(t)$ (cf. Fig. 4.3). By definition, these subsets of $\mathcal{F}(t)$ only contain states that guarantee a safe transition to another safe state for an infinite time horizon $t_{\text{horizon}} \rightarrow \infty$. As a result, these subsets do not include ICSs (cf. Sec. 4.1.1) and thus are invariably safe. We formally define the set of invariably safe states as:

Definition 20 (Invariably Safe Set \mathcal{S}) *The invariably safe set $\mathcal{S}(t)$ for a point in time t contains all collision-free states $x \in \mathcal{F}(t)$ that allow the ego vehicle to be safe for an infinite time horizon and is defined as*

$$\mathcal{S}(t) := \{x \in \mathcal{F}(t) \mid \forall t' > t : \chi(t', x, \Phi(x([t, t'], \phi_{\text{ref}}))) \in \mathcal{F}(t')\}.$$

In contrast to CISs, we determine invariably safe sets using correct-by-construction control laws Φ that keep the ego vehicle safe (more details are given in Sec. 4.3).

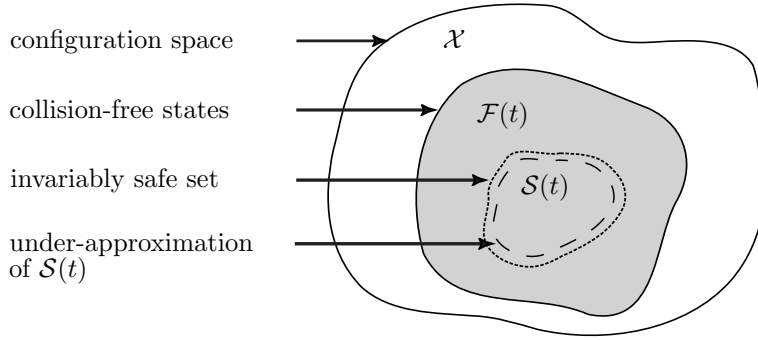


Figure 4.3: Subdivision of the configuration space. Relation between the configuration space \mathcal{X} , collision-free states $\mathcal{F}(t)$, and invariably safe sets $\mathcal{S}(t)$. ©2018 IEEE.

Unfortunately, determining the maximal invariably safe set is a computationally intractable task in most traffic scenarios, since we need to reason over an infinite time horizon for all states. However, we show that an under-approximation of the maximal invariably safe set can be computed from a known invariably safe set at a future point in time τ , allowing us to stop the recursive computation.

We first focus on determining an invariably safe set, which allows us to inductively derive other such sets. To this end, we consider a frequent traffic situation in which the ego vehicle is following an arbitrary obstacle in its lane. Based on the traffic rules of the Vienna Convention [39, Art. 13 and Art. 31], we can state that if this preceding obstacle comes to a stop, the ego vehicle is allowed to stop behind it within a certain area. This state is safe since the following vehicles are not allowed to cause a rear-end collision [39]. Moreover, the ego vehicle can remain in this safe state for an infinite time horizon since it is at a standstill.

We model this safe vehicle following scenario by introducing $\Omega(b, \beta) \subset \mathcal{E}$ as the area in a lane where it is admissible to come to a standstill behind a stopped obstacle $b \in \mathcal{B}$ within a certain distance β (to disregard stopping far away from the preceding obstacle). Without loss of generality, the distance β is defined as ranging from the rear bumper of the ego vehicle to the occupancy of b along Γ in the curvilinear coordinate system of the lane. Usually, β is at least as long as the length of the ego vehicle so that the ego vehicle remains collision-free when occupying $\Omega(b, \beta)$. We now show that the set of collision-free states behind a stopped preceding obstacle within Ω is an invariably safe set according to Def. 20.

Lemma 1 (Invariably Safe Set $\mathcal{S}(\tau)$ at a Standstill) *Assuming that the preceding obstacle b stops at any future time $\tau > t$, the set $\mathcal{S}(\tau) := \{x \mid v_{[x]} = 0 \wedge \text{occ}(x) \subseteq \Omega(b, \beta)\}$ is an invariably safe set according to Def. 20, where $v_{[x]}$ describes the velocity in state x .*

Proof *By definition, states $x \in \mathcal{S}(\tau)$ are collision-free, and thus, $\mathcal{S}(\tau) \subseteq \mathcal{F}(\tau)$. All $x \in \mathcal{S}(\tau)$ remain safe for all times $t' > \tau$ by choosing a control law $\Phi(x([t, t'], \phi_{\text{ref}}) = u([t, t']) = 0$. ■*

We use collision-free backward reachable sets (cf. Def. 7) to derive additional invariably safe sets for times prior to τ . To use induction, we determine invariably safe sets for time intervals prior to τ . The set $\mathcal{S}_k := \mathcal{S}(\mathbb{T}_k)$, $k \in \mathbb{N}_+$ denotes the invariably safe set for the time interval $\mathbb{T}_k := [\tau - k\epsilon, \tau - (k-1)\epsilon]$, prior to τ , where $\epsilon \in \mathbb{R}_+$ is an arbitrarily small step size.

Theorem 1 (Determining Invariably Safe Sets) *The set $\mathcal{S}_k := \tilde{\mathcal{R}}(\epsilon, \mathcal{O}_{\mathcal{B}}(\mathbb{T}_k), \mathcal{S}_{k-1})$ for the time interval \mathbb{T}_k and $\mathcal{S}_0 = \mathcal{S}(\tau)$ is an invariably safe set according to Def. 20.*

Proof *We prove the theorem inductively.*

Base case ($k = 1$): $\mathcal{S}_1 = \mathcal{S}([\tau - \epsilon, \tau]) = \tilde{\mathcal{R}}(\epsilon, \mathcal{O}_{\mathcal{B}}([\tau - \epsilon, \tau]), \mathcal{S}(\tau))$. Based on the collision-free backward reachable set, for every state $x \in \mathcal{S}_1$, there exists a collision-free trajectory to the invariably safe set $\mathcal{S}(\tau)$ (cf. Lem. 1) - that is, $\forall x \in \mathcal{S}_1 : \exists r \leq \epsilon : \exists u([\tau - r, \tau]) : \chi(\tau, x, u([\tau - r, \tau])) \in \mathcal{S}(\tau)$. As a result, persistent feasibility (cf. Sec. 4.1) is guaranteed for times $t' > \tau$.

Inductive step: We show that $\mathcal{S}_{k+1} = \tilde{\mathcal{R}}(\epsilon, \mathcal{O}_{\mathcal{B}}(\mathbb{T}_{k+1}), \mathcal{S}_k)$ is an invariably safe set, which allows us to determine a collision-free trajectory to \mathcal{S}_k for every state $x \in \mathcal{S}_{k+1}$ (analogous to base case). Since \mathcal{S}_k is an invariably safe set, every invariably safe set \mathcal{S}_j , $0 \leq j \leq k$, is reachable from \mathcal{S}_{k+1} collision-free (cf. assumption of inductive step). ■

Fig. 4.4 illustrates this iterative computation of invariably safe sets using backward reachability analysis [266]. The computation terminates when the initial time t_0 of the current motion planning problem has been reached at a certain step k_n (i.e., $t_0 \in \mathbb{T}_{k_n}$).

4.3 Under-Approximation of Invariably Safe Sets

The backward reachability approach proposed in Sec. 4.2 makes the problem of determining invariably safe sets computationally tractable. However, it is still not on-line capable, and it is thus not applicable to planning problems with hard real-time constraints. In this section, we show how we can compute an under-approximation of invariably safe sets that can be obtained with linear computational complexity.

Similar to CISs, the computation of invariably safe sets requires us to reason over infinite time horizons (cf. Sec. 4.1.2 and Def. 20). However, the occupancy of other traffic participants is usually unknown for infinite time horizons. This makes it difficult to compute the known invariably safe set $\mathcal{S}(\tau)$ at a standstill (cf. Lem. 1), since preceding obstacles may stop at any future time τ . In real-world applications, it is sufficient to consider the worst case to ensure safety: we assume that the preceding obstacle $b \in \mathcal{B}$ immediately starts braking at the current time t (as predicted by the set-based prediction) and consider the time horizon $\tau \approx t + t_{\text{stop},b}$, where $t_{\text{stop},b} = v_b/|a_{\text{max},b}|$ corresponds to the minimum time required for the preceding obstacle b to come to a stop when fully decelerating with $-|a_{\text{max},b}|$ and velocity v_b .

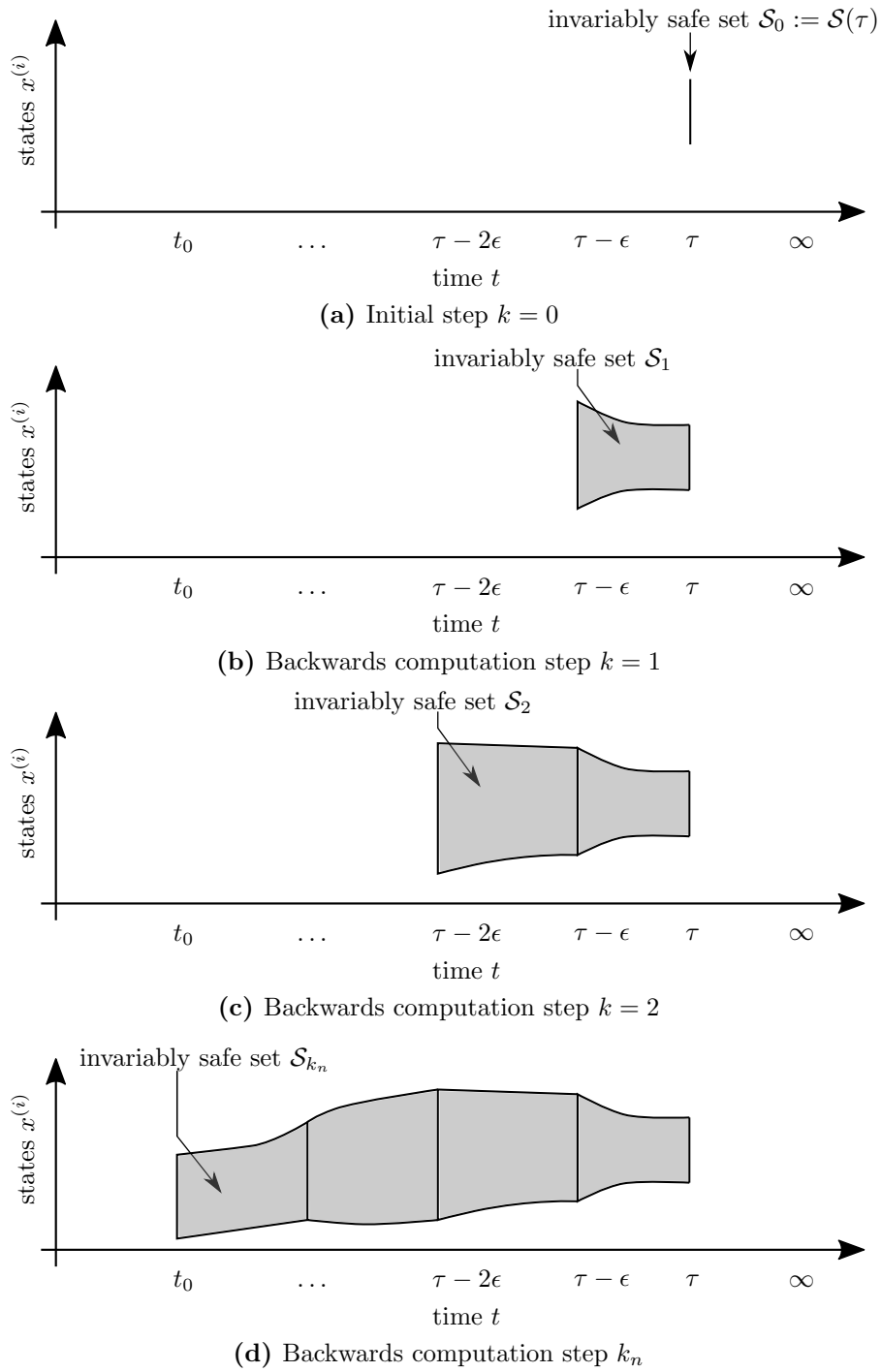


Figure 4.4: Illustration of backward computation of invariably safe sets. We iteratively compute invariably safe sets using backward reachability analysis and starting from a known invariably safe set \mathcal{S}_0 . The computation stops at step k_n when the initial time t_0 has been reached.

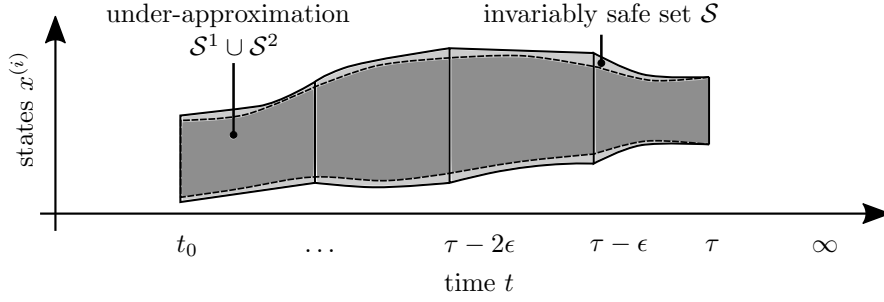


Figure 4.5: Under-approximation of invariably safe sets. The union of \mathcal{S}^1 and \mathcal{S}^2 is a tight under-approximation of the invariably safe set \mathcal{S} .

This simplification allows us to efficiently derive a tight under-approximation of \mathcal{S} (cf. Fig. 4.3) using the following sets

- \mathcal{S}^1 of states in which the ego vehicle respects formal safe distances to preceding vehicles according to [267], and
- \mathcal{S}^2 of states in which the ego vehicle respects evasive distances to preceding vehicles according to [7].

Safe distances allow the ego vehicle to safely stop if preceding vehicles suddenly perform emergency braking. On the other hand, evasive distances allow the ego vehicle to safely change to an adjacent lane (with constant velocity) while respecting formal safe distances to obstacles in the target lane. For higher velocities, evasive distances are usually shorter than safe distances [64]. More details on both distances are given later in Sec. 4.3.2. We formally define the under-approximation of \mathcal{S} as:

Proposition 3 (Under-Approximation of \mathcal{S}) *The union of the set \mathcal{S}^1 of states respecting safe distances and of the set \mathcal{S}^2 of states respecting evasive distances to a preceding obstacle at time t is an under-approximation of the invariably safe set \mathcal{S} , that is, $\mathcal{S}^1 \cup \mathcal{S}^2 \subset \mathcal{S}$.*

Proof *The soundness of safe and evasive distances is shown in [7, 267, 268]. We provide a counter-example to demonstrate that the resulting set is an under-approximation: based on [64], the last possible evasive maneuver to avoid a collision must be a combination of braking and steering, in contrast to safe (solely braking) or evasive distances (solely evading) alone. ■*

Fig. 4.5 illustrates the relation of the under-approximation to the invariably safe set. In fact, in Sec. 4.6.2, we show that the obtained set is a tight under-approximation of the invariably safe set \mathcal{S} . First, in the following sections we demonstrate how the under-approximation of \mathcal{S} can be computed efficiently.

4.3.1 Environment representation

Similar to in Ch. 3, we describe predicted occupancy sets $\mathcal{O}_{\mathcal{B}}(t)$ of obstacles and the environment \mathcal{E} by using a curvilinear coordinate system aligned with a reference path Γ , such as the center of the lane (cf. Fig. 2.1). In addition, we enlarge $\mathcal{O}_{\mathcal{B}}(t)$ for collision checking by adding the dimensions of the ego vehicle [197] (cf. Sec. 3.3). It should be noted that the occupancy sets are an input of the presented algorithm; thus, the computation automatically adapts to new legal safety specifications or violations by other traffic participants in the set-based prediction (cf. Sec. 2.4). For instance, if obstacles violate certain assumptions, the occupancies become larger and our obtained safe sets smaller.

Without loss of generality, we model the state of the ego vehicle as $x = (s, d, v)^T \in \mathbb{R}^3$, where s is the longitudinal position, d is the lateral position, and v is the velocity. Positions $(s, d)^T$ describe the geometric center of the ego vehicle. Of note is that other state models can be incorporated as well, for example by converting them to the used model in this section. To account for the limited field of view of the ego vehicle, we place static obstacles at the field of view's border to guarantee that the ego vehicle is able to stop within its sensor view [143]. Road boundaries and varying lane widths are integrated by limiting the allowed lateral deviations d . Therefore, we remove states from \mathcal{S} that lead to a violation of the road boundary constraints.

We divide the area of a lane into sections $\mathcal{E}_{b_1, b_2}(t_0) \subset \mathcal{E}$, $b_1, b_2 \in \mathcal{B}$, (cf. Fig. 4.6) delimited by the occupancies of a pair of obstacles b_1 (following) and b_2 (preceding) within the considered lane (ordered along the lane with ascending s position in the curvilinear coordinate system). For instance, for obstacles b_1 and b_2 and occupancies $\mathcal{O}_1(t_0)$ and $\mathcal{O}_2(t_0)$, we compute

$$\mathcal{E}_{b_1, b_2}(t_0) := \left\{ (s, d)^T \in \mathbb{R}^2 \mid \forall (s_1, d_1)^T \in \mathcal{O}_1(t_0), \forall (s_2, d_2)^T \in \mathcal{O}_2(t_0) : \right. \\ \left. s_1 < s < s_2 + \Delta s_{2, \text{stop}} \right\} \subset \mathcal{E}, \quad (4.1)$$

where $\Delta s_{2, \text{stop}} := v_2^2 / 2|a_{\min, b_2}|$ is the distance required for obstacle b_2 to come to a stop when performing emergency braking with deceleration a_{\min, b_2} and velocity v_2 . We include $\Delta s_{2, \text{stop}}$ in sections to obtain the maximal area in a lane that the ego vehicle can use to perform evasive maneuvers, as described in Sec. 4.3.2. As a result, we can determine the curvature in the section to incorporate the maximum feasible braking (or evasive) acceleration of the ego vehicle along the curved road in the invariably safe set computation.

Fig. 4.6 displays the concept of sections in an example scenario. The illustrated section \mathcal{E}_{b_1, b_2} includes the stopping distance of the preceding obstacle b_2 to account for the road geometry until obstacle b_2 has stopped (the gray rectangle representing \mathcal{E}_{b_1, b_2} ends at the beginning of $\mathcal{O}_{b_2}(t_2)$).

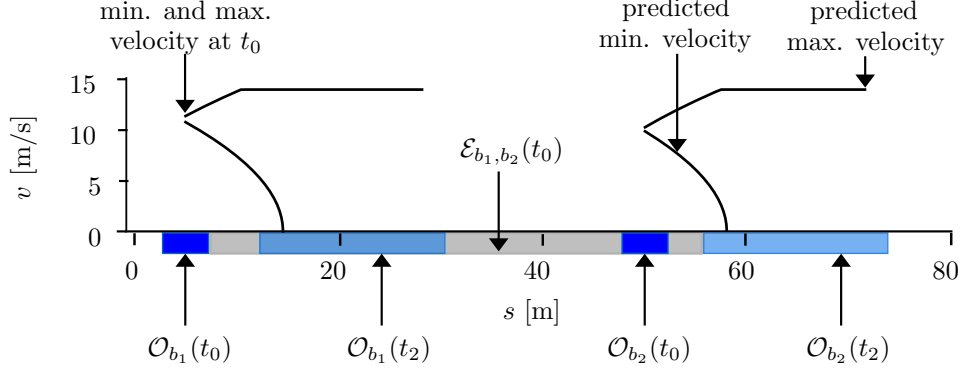


Figure 4.6: Illustration of sections. Section $\mathcal{E}_{b_1, b_2}(t_0)$ includes all positions between obstacles b_1 and b_2 at time t_0 . We include the stopping distance of b_2 to obtain the road geometry until obstacle b_2 has fully stopped. The predicted occupancy sets are shown at a future point in time t_2 . The predicted velocities over time $t \in [t_0, t_2]$ are shown as a projection onto the s - v plane. The velocities illustrate that both obstacles are either at a standstill at time t_2 or moving (initial velocities are uncertain and are thus intervals).

4.3.2 Algorithmic steps

Without loss of generality, we assume that the desired lanes along the reference path Γ are given to obtain all relevant sections $\mathcal{E}_{b_i, b_j}, b_i, b_j \in \mathcal{B}$. Alg. 2 computes the under-approximation $\mathcal{S}^1 \cup \mathcal{S}^2 \subset \mathcal{S}$ for a time t and a section \mathcal{E}_{b_i, b_j} along arbitrary road networks. The algorithm is recursively applied to every section (computed for each neighboring pair of obstacles), which can be done in a parallelized fashion.

a) Velocity and acceleration constraint subroutines Without loss of generality, we assume that the ego vehicle closely follows the reference path Γ . The curvature $\kappa_\Gamma(s)$ of Γ at position s limits the maximum feasible velocities and accelerations of the ego vehicle. To ensure controllability, we account for these constraints during the computation of the under-approximation. Based on [269, Eq. 1], the maximum feasible velocity $v_{\text{crit}}(s)$ at a certain longitudinal position s is given by:

$$v_{\text{crit}}(s) := \sqrt{a_{d, \max} / \kappa_\Gamma(s)}. \quad (4.2)$$

The maximum critical velocity within a section \mathcal{E}_{b_i, b_j} is obtained by considering the maximum curvature of this section $\kappa_{\Gamma, \max} = \sup\{\kappa_\Gamma(s) \mid s \in \mathcal{E}_{b_i, b_j}\}$ in (4.2), denoted as v_{crit}^* in the following (cf. lines 1-2 of Alg. 2). The under-approximation further incorporates any given legal speed limit $v_{\text{limit}}(s)$. The allowed maximum velocity of the ego vehicle is determined by $v_{\max}(s) = \min(v_{\text{crit}}(s), v_{\text{limit}}(s))$.

In lines 4-5 of Alg. 2, we compute the maximum feasible lateral and longitudinal accelerations, $a_d(v)$ and $a_s(v)$, for all possible velocities v and the lane's curvature $\kappa_\Gamma(s), s \in \mathcal{E}_{b_i, b_j}$. Given the maximum curvature $\kappa_{\Gamma, \max}$ within a section \mathcal{E}_{b_i, b_j} , we

Algorithm 2 invariablySafeSets

Input: $t, \mathcal{E}_{b_i, b_j}, \kappa, v_{\text{limit}}, \mathcal{O}_j(t), v_j(t), \delta_{\text{brake}}, \delta_{\text{steer}}$
Output: Under-approximation of \mathcal{S}^t

- a) *Velocity and acceleration constraint subroutines* [269, Eq. 2-4]:
 - 1: $\kappa_{\Gamma, \max} \leftarrow \sup\{\kappa_{\Gamma}(s) \mid s \in \mathcal{E}_{b_i, b_j}\}$
 - 2: $v_{\text{crit}}^* \leftarrow \sqrt{a_{d, \max}/\kappa_{\Gamma, \max}}$
 - 3: $v_{\max}(s) := \min(v_{\text{crit}}^*, v_{\text{limit}}(s))$
 - 4: $a_d(v) := a_{d, \max}(v/v_{\text{crit}}^*)^2$
 - 5: $a_s(v) := a_{s, \max} \sqrt{1 - (v^2/(v_{\text{crit}}^*)^2)^2}$
 - b) *Safe distance subroutine* [267, Eq. 17]:
 - 6: $\Delta_{\text{safe}, 2}(v, b_j) := \max((v_j^2/-2|a_{s, \max, j}|) - (v^2/-2|a_s(v)|) + v\delta_{\text{brake}}, 0)$
 - c) *Evasive distance subroutines* [7, Eq. 11-13]:
 - 7: $t_{\text{eva}}(v) := \sqrt{(2d_{\text{eva}}/(a_{d, \max} - a_d(v)))} + \delta_{\text{steer}}$
 - 8: $\Delta_{\text{eva}}(v, b_j) := vt_{\text{eva}}(v) - (v_j(t)t_b - \frac{1}{2}a_{s, \max, j}t_b^2),$
 $t_b = \min(v_j(t)/a_{s, \max, j}, t_{\text{eva}}(v))$
 - d) *Invariably safe sets* \mathcal{S}^1 and \mathcal{S}^2 :
 - 9: $\mathcal{S}^1 \leftarrow \{(s, d, v)^T \in \mathcal{X} \mid \forall (s_j, \cdot)^T \in \mathcal{O}_j(t) : s \leq s_j - \Delta_{\text{safe}, 2}^t(v, b_j) \wedge v \leq v_{\max}(s) \wedge s \in \mathcal{E}_{b_i, b_j}\}$
 - 10: $\mathcal{S}^2 \leftarrow \{(s, d, v)^T \in \mathcal{X} \mid \forall (s_j, \cdot)^T \in \mathcal{O}_j(t) : s \leq s_j - \Delta_{\text{eva}}^t(v, b_j) \wedge v \leq v_{\max}(s) \wedge s \in \mathcal{E}_{b_i, b_j} \wedge (\forall r \in [0, t_{\text{eva}}(v)] : (s + vr, d', v)^T \in \mathcal{S}^1(t + r))\}$
 - 11: **return** $\mathcal{S}^1 \cup \mathcal{S}^2$
-

relate the maximum feasible lateral acceleration $a_{d, \max} = (v_{\text{crit}}^*)^2 \kappa_{\Gamma, \max}$ to the lateral acceleration $a_d(v) = v^2 \kappa_{\Gamma, \max}$ for velocities $v \leq v_{\text{crit}}^*$. Solving for $a_d(v)$ results in:

$$a_d(v) = a_{d, \max} (v/v_{\text{crit}}^*)^2. \quad (4.3)$$

Furthermore, we use the friction ellipse [269] to compute the feasible longitudinal acceleration in the section as:

$$a_s(v) = a_{s, \max} \sqrt{1 - (v^2/(v_{\text{crit}}^*)^2)^2}. \quad (4.4)$$

b) Safe distance subroutine The safe distance to a preceding obstacle is defined as a “sufficient distance [...] to avoid [a] collision if the vehicle in front should suddenly slow down or stop” [39, §13]. This definition has been formalized using higher-order logics and the Isabelle theorem prover [95] in [267]. We briefly recall the results from [267] here. The computation of the minimum required safe distance between the ego vehicle with velocity v_{ego} and absolute deceleration $-|a_{s, \max}|$ and a preceding obstacle $b \in \mathcal{B}$ with velocity v_b and maximum absolute deceleration $-|a_{s, \max, b}|$ depends on the following condition [270]:

$$(|a_{s, \max, b}| < |a_{s, \max}|) \wedge (v_b^* < v_{\text{ego}}) \wedge (v_{\text{ego}}/|a_{s, \max}| < v_b^*/|a_{s, \max, b}|), \quad (4.5)$$

where δ_{brake} denotes the reaction time of the ego vehicle to perform braking, and v_b^* represents the remaining velocity of obstacle b after an emergency brake maneuver of obstacle b with duration δ_{brake} , defined as:

$$v_b^* := \begin{cases} v_b - |a_{s,\text{max},b}|\delta_{\text{brake}} & \delta_{\text{brake}} \leq v_b/|a_{\text{max},b}|, \\ 0 & \text{otherwise.} \end{cases} \quad (4.6)$$

If condition (4.5) evaluates to true, the ego vehicle has to maintain the safe distance $\Delta_{\text{safe},1}$ to obstacle b . Otherwise, $\Delta_{\text{safe},2}$:

$$\begin{aligned} \Delta_{\text{safe},1}(v_{\text{ego}}, b) &:= \frac{(v_b - |a_{s,\text{max},b}|\delta_{\text{brake}} - v_{\text{ego}})^2}{-2(|a_{s,\text{max},b}| - |a_{s,\text{max}}|)} - v_b\delta_{\text{brake}} + \frac{1}{2}|a_{s,\text{max},b}|\delta_{\text{brake}}^2 \\ &\quad + v_{\text{ego}}\delta_{\text{brake}}, \\ \Delta_{\text{safe},2}(v_{\text{ego}}, b) &:= \frac{v_b^2}{-2|a_{s,\text{max},b}|} - \frac{v_{\text{ego}}^2}{-2|a_{s,\text{max}}|} + v_{\text{ego}}\delta_{\text{brake}}. \end{aligned} \quad (4.7)$$

It should be noted that the initial state of the ego vehicle and the prediction of other traffic participants are given.

To provide a conservative estimation of the deceleration capabilities of obstacles, we assume that obstacles have equal or greater deceleration capabilities than the ego vehicle. As a result, (4.5) always evaluates to false and the ego vehicle has to respect the safe distance $\Delta_{\text{safe},2}$ to other obstacles. Fig. 4.7a illustrates $\Delta_{\text{safe},2}$ for different velocities of the ego vehicle and of the preceding obstacle. Line 6 of Alg. 2 computes this safe distance to a preceding obstacle b_j with velocity v_j for a provided ego vehicle velocity v_{ego} and available longitudinal deceleration $-|a_s(v_{\text{ego}})|$.

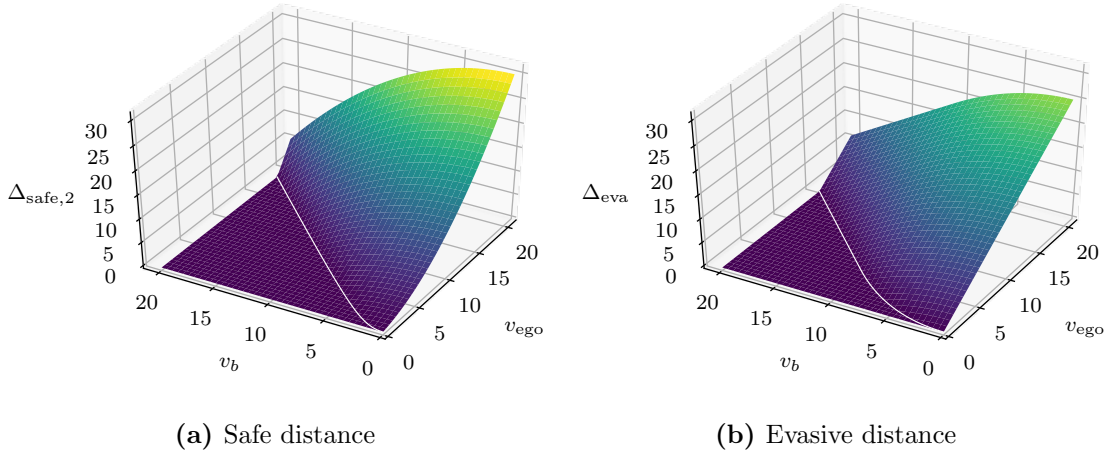


Figure 4.7: Illustration of safe and evasive distances. (a) Safe distances for different velocities of the ego vehicle v_{ego} and a preceding obstacle v_b . (b) Evasive distances for the same velocities as in (a). Both figures share the same colormap for comparison.

c) Evasive distance subroutines In contrast to safe distances, evasive distances provide the ego vehicle with a sufficient distance to avoid a collision by evading if the preceding obstacle b performs emergency braking. We introduce $d_{\text{eva}}(d)$ as the lateral distance necessary to fully enter an adjacent lane from a given lateral position d (whole shape of the ego vehicle). For clarity, we omit the dependence on d in Alg. 2. Based on the maximum lateral acceleration $a_{d,\text{max}}$ in the current section \mathcal{E}_{b_i,b_j} , the time t_{eva} required to perform the evading maneuver can be computed as:

$$t_{\text{eva}} := \sqrt{2d_{\text{eva}}/a_{d,\text{max}}} + \delta_{\text{steer}}, \quad (4.8)$$

where δ_{steer} denotes the reaction time of the ego vehicle's steering system (cf. line 7 in Alg. 2). Using the dynamics of a double integrator system, we compute the traveled distance Δs_b of obstacle b during emergency braking with a deceleration of $-|a_{s,\text{max},b}|$:

$$\begin{aligned} \Delta s_b &:= v_b t_b - \frac{1}{2} |a_{s,\text{max},b}| t_b^2, \\ t_b &:= \min(t_{\text{eva}}, v_b/|a_{s,\text{max},b}|). \end{aligned} \quad (4.9)$$

The evasive distance Δ_{eva} to the preceding obstacle b is obtained by [7, Eq. 12-13]:

$$\Delta_{\text{eva}}(v_{\text{ego}}, b) := v_{\text{ego}} t_{\text{eva}} - \Delta s_b. \quad (4.10)$$

Fig. 4.7b illustrates Δ_{eva} for different velocities of the ego vehicle and the preceding obstacle. Line 8 of Alg. 2 computes the evasive distance to a preceding obstacle according to (4.10), and it also accounts for the remaining lateral acceleration $a_d(v)$.

Invariably safe sets \mathcal{S}^1 and \mathcal{S}^2 We compute set \mathcal{S}^1 , which contains states that respect a safe distance to preceding obstacles at a point in time t based on (4.7) in line 9 of Alg. 2. It should be noted that the proposed approach can also consider safe distances to following obstacles to prohibit the ego vehicle from directly merging in front of another obstacle during lane changes; for clarity, this part is omitted in Alg. 2, but it can be obtained analogously to safe distances to preceding obstacles by adding a position constraint considering following vehicles. Set \mathcal{S}^2 denotes the set of states that respect evasive distances to preceding obstacles based on (4.10) and a safe distance to preceding obstacles in the adjacent lane. We check the latter requirement by ensuring that states in \mathcal{S}^2 are also enclosed in \mathcal{S}^1 for the adjacent lane. Since the ego vehicle moves with constant velocity during the evasive maneuver, we can simply implement this check with the constraint $\forall r \in [0, t_{\text{eva}}(v)] : (s+vr, d', v)^T \in \mathcal{S}^1(t+r)$.

4.3.3 Computational complexity

According to [271], the complexity of the occupancy prediction is linear with respect to the number of traffic participants. The computational complexity of Alg. 2

for all sections is also linear, since one has to perform a constant number of calculations (number of discrete velocities) per section. Thus, the overall complexity of computing the under-approximation of \mathcal{S} corresponds to $O(n_{\mathcal{B}})$ with $n_{\mathcal{B}} = |\mathcal{B}|$. In Sec. 4.6, we demonstrate that the linear complexity translates to a computation of invariably safe sets in real-time. As a result, the obtained under-approximation is well suited to ensure the safety of autonomous vehicles during the operation.

4.4 Exploiting Invariably Safe Sets for Motion Planning

Invariably safe sets offer many advantages for safe motion planning of autonomous vehicles. In general, planned trajectories $u([t_0, t_h])$ are verified as collision-free within the time interval $[t_0, t_h]$ prior to their execution (cf. Def. 9). Invariably safe sets can be used to guarantee that the ego vehicle is able to remain safe for times $t' > t_h$. This property is called persistent feasibility and is particularly important for cyclic replanning approaches, such as model predictive control as they rely on finite planning horizons. We verify planned trajectories for an infinite time horizon by checking whether their final state is an invariably safe state:

Definition 21 (Invariably Safe Input Trajectory) *The trajectory $u([t_0, t_h])$, $t_0 < t_h$, is called an invariably safe input trajectory if $u([t_0, t_h])$ is a collision-free input trajectory (cf. Def 9) and $\chi(t_h, x(t_0), u([t_0, t_h])) \in \mathcal{S}(t_h)$ (cf. Def. 20).*

The size of $\mathcal{S}(t)$ depends on the predicted occupancies of obstacles. If the occupancy $\mathcal{O}_{\mathcal{B}}$ of obstacles becomes larger, $\mathcal{S}(t)$ becomes smaller and trajectories $u([t_0, t_h])$ may not be enclosed in $\mathcal{S}(t)$ from some point in time $t \in]t_0, t_h]$ (assuming that $x(t_0) \in \mathcal{S}(t_0)$). Thus, another use of invariably safe sets is to obtain the *time-to-react* (TTR) [272, Sec. II].

Definition 22 (Time-To-React) *Assuming that $x(t_0) \in \mathcal{S}(t_0)$, the time-to-react (TTR) is the maximum time the ego vehicle can continue the trajectory $u([t_0, t_h])$ for which the existence of an evasive trajectory is guaranteed: $t_{\text{TTR}} := \sup \{t \mid t \in [t_0, t_h] \wedge \chi(t, x(t_0), u([t_0, t])) \in \mathcal{S}(t)\}$.*

The TTR is an often-used metric for criticality assessment [272, 273]. If the obtained TTR is small, a safety-critical situation might approach soon. Larger values indicate that the ego vehicle does not need to intervene soon and it has more time to observe the scenario [213]. In contrast, the popular time-to-collision (TTC) metric corresponds to the time until a collision occurs based on constant-acceleration predictions of obstacles and the intended trajectory of the ego vehicle [200, 274]. The time-to-react (TTR) [45, 272, 273] is considered a more informative metric, since it is the remaining time along the intended trajectory until one can avoid a collision. We use the obtained TTR as the optimal point in time at which the ego vehicle should intervene to avoid a potential collision. This choice reflects that

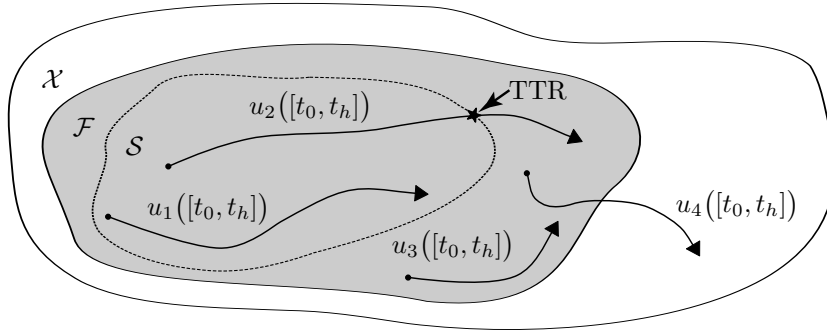


Figure 4.8: Safety properties of trajectories. Only trajectory u_1 is an invariably safe trajectory which remains safe for $t > t_h$, since it remains in the invariably safe set \mathcal{S} . Trajectory u_2 leaves \mathcal{S} at the time-to-react TTR. Trajectory u_3 is a collision-free trajectory, since it remains in the set of collision-free states \mathcal{F} . Trajectory u_4 , on the other hand, is not collision-free. ©2020 IEEE.

system designers usually want safety systems to intervene at the latest possible point in time. Consequently, our computed TTR corresponds to the optimal point in time to execute a fail-safe trajectory along a given trajectory. We will use this information in Ch. 5 to develop a new verification technique that combines fail-safe trajectories and invariably safe sets. Fig. 4.8 illustrates the different safety properties of trajectories and the TTR.

4.5 Integration of Invariably Safe Sets into Linear-Quadratic Programs

To ensure the safety of the ego vehicle at all times, we are interested in planning fail-safe trajectories (cf. Ch. 3) which are invariably safe input trajectories (cf. Def. 21). Therefore, we need to integrate invariably safe sets as a terminal constraint into our fail-safe trajectory planner. However, our fail-safe trajectory planner only accepts linear constraints of the general form $\mathcal{H}x \leq o$. For this reason, we present how the under-approximation of invariably safe sets can be transformed to sets of linear constraints for the convex optimization problems (cf. Sec. 3.2 and Sec. 3.3). For the sake of brevity, we focus on the linearization aspects in the following sections and neglect the additional technicalities of invariably safe sets, such as the computation of accelerations with respect to the road geometry.

4.5.1 Linear safe distance constraints

Safe distances cannot be directly included in linear-quadratic programs (cf. Sec. 2.5), since they are quadratic in the velocity of the ego vehicle. To circumvent this prob-

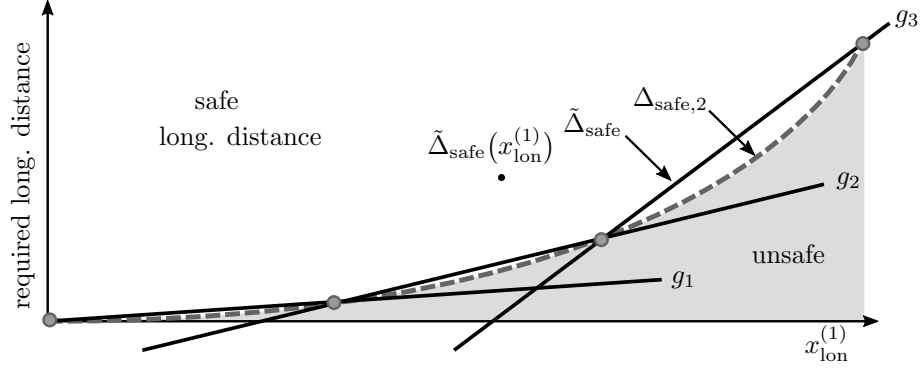


Figure 4.9: Piecewise linear approximation of safe distances. The ego vehicle must maintain the longitudinal safe distance $\Delta_{\text{safe},2}$ to preceding obstacles to remain safe (white area). This convex safe distance $\Delta_{\text{safe},2}$ can be approximated by a linear piecewise function $\tilde{\Delta}_{\text{safe}}$, composed by p linear functions $g_i, i \in \{1, \dots, p\}$. A safe point $\tilde{\Delta}_{\text{safe}}(x_{\text{lon}}^{(1)})$ which respects the linear safe distance also fulfills $\forall i \in \{1, \dots, p\} : \tilde{\Delta}_{\text{safe}}(x_{\text{lon}}^{(1)}) \geq g_i(x_{\text{lon}}^{(1)})$. ©2020 IEEE.

lem, we exploit the convexity of the safe distance functions (they are a sum of convex quadratic and linear functions) and use a piecewise linear approximation of the safe distance instead. The resulting linear approximation of the safe distance is over-approximative and therefore still ensures safety. The presented constraints are added to the longitudinal optimization problem (cf. Sec. 3.2.1).

We use p linear functions $g_1, g_2, \dots, g_p : \mathbb{R} \rightarrow \mathbb{R}$ to approximate the safe distance $\Delta_{\text{safe}} \in \{\Delta_{\text{safe},1}, \Delta_{\text{safe},2}\}$. To achieve this, we divide the velocity range $[v_{\min}, v_{\max}]$, $0 < v_{\min} < v_{\max}$, of the ego vehicle in p equally large intervals $[v_i, v_{i+1}]$, $i \in \{0, \dots, p-1\}$, by setting

$$v_i = (v_{\max} - v_{\min}) \frac{i}{p} + v_{\min}. \quad (4.11)$$

For the sake of clarity, we demonstrate the linearization with $\Delta_{\text{safe},2}$ in the following paragraphs. Note that the prediction of the future motion of preceding vehicles is provided as a parameter in the computations. For each interval, we approximate the safe distance $\Delta_{\text{safe},2}$ using linear functions g_i , resulting in the linear safe distance formulation:

$$\tilde{\Delta}_{\text{safe}}(x_{\text{lon}}^{(1)}) = \begin{cases} g_1(x_{\text{lon}}^{(1)}), & v_0 \leq x_{\text{lon}}^{(1)} < v_1, \\ g_2(x_{\text{lon}}^{(1)}), & v_1 \leq x_{\text{lon}}^{(1)} < v_2, \\ \vdots & \\ g_p(x_{\text{lon}}^{(1)}), & x_{\text{lon}}^{(1)} \geq v_{p-1}. \end{cases} \quad (4.12)$$

Fig. 4.9 illustrates the piecewise linear approximation of the safe distance. The ego vehicle is not allowed to enter the shaded region in order to guarantee safety.

In order to integrate the p linear functions into the optimization problem, we make use of the fact that each convex, piecewise linear function can be represented

as a maximum function [275]. Thus, the safe distance can be reformulated as

$$\tilde{\Delta}_{\text{safe}}(x_{\text{lon}}^{(1)}) = \max \left\{ g_1(x_{\text{lon}}^{(1)}), g_2(x_{\text{lon}}^{(1)}), \dots, g_p(x_{\text{lon}}^{(1)}) \right\}.$$

Respecting the maximum of these p linear functions is equivalent to satisfying every single one of them due to convexity (cf. example point $\tilde{\Delta}_{\text{safe}}(x_{\text{lon}}^{(1)})$ in Fig. 4.9).

According to the unique general form of linear equations, each linear function $g_i(x_{\text{lon}}^{(1)})$ can be expressed as $g_i(x_{\text{lon}}^{(1)}) = m_i(x_{\text{lon}}^{(1)} - v_i) + \Delta_i$, where $m_i \in \mathbb{R}$ is the slope and $\Delta_i \in \mathbb{R}$ is the offset of the linear function. We integrate the safe distance equation $\Delta_{\text{safe},2}$ (cf. (4.7)) into the longitudinal position constraint (3.4) to obtain:

$$x_{\text{lon}}^{(0)}(t) + \Delta_{\text{safe},2}(x_{\text{lon}}^{(1)}) \leq s_{\text{max}}(t). \quad (4.13)$$

With the linear function $g_i(x_{\text{lon}}^{(1)})$, we can further rearrange the constraint to:

$$x_{\text{lon}}^{(0)}(t) + (g_i(x_{\text{lon}}^{(1)}) + \delta_{\text{brake}}x_{\text{lon}}^{(1)}) \leq s_{\text{max}}(t), \quad (4.14)$$

which is equal to:

$$x_{\text{lon}}^{(0)}(t) + m_i(x_{\text{lon}}^{(1)}(t) - v_i) + \Delta_i + \delta_{\text{brake}}x_{\text{lon}}^{(1)}(t) \leq s_{\text{max}}(t). \quad (4.15)$$

As a result, we obtain p linear position constraints (4.15). These constraints are added to the longitudinal optimization problem to constrain the terminal state of the trajectory (since the terminal state needs to be invariably safe as shown in Def. 21). Larger numbers of linear functions p decrease the approximation error, but increase the computational time of solving the optimization problem.

The algorithmic realization of the linearization is shown in Alg. 3. This algorithm creates a set of p linear safe distance constraints for a point in time t and a preceding obstacle $b \in \mathcal{B}$. The prediction of the preceding obstacle b is given by s_{max} and $v_{\text{min},b}$ for the position and velocity. The parameters $a_{\text{s,max}}$ and $a_{\text{s,max},b}$ denote the maximum absolute acceleration of the ego vehicle and the preceding obstacle b , respectively. In lines 5-8, we compute the slope m_i and the offset Δ_i of the linear function for step i (required in (4.15)). The obtained set of constraints $\mathcal{C}_{\Delta_{\text{safe}}}(t)$ is added to the longitudinal optimization problem.

4.5.2 Linear evasive distance constraints

The integration of evasive distance constraints is simpler than safe distances for motion planning problems which are separated into longitudinal and lateral components. However, the integration may result in a strong over-approximation of the evasive distance, as shown later. The evasive distance (4.10) for an obstacle $b \in \mathcal{B}$ is already in a linear form. It is added to the longitudinal optimization problem with the constraint:

$$x_{\text{lon}}^{(0)}(t) + \Delta_{\text{eva}}(x_{\text{lon}}^{(1)}(t), b) \leq s_{\text{max},b}(t), \quad (4.16)$$

Algorithm 3 linearizeSafeDistance

Input: $p, v_{\min}, v_{\max}, a_{s,\max}, a_{s,\max,b}, \delta_{\text{brake}}, t, s_{\max}, v_{\min,b}$
Output: Set of linear safe distance constraints $\mathcal{C}_{\Delta_{\text{safe}}}(t)$

```

1:  $\Delta v := (v_{\max} - v_{\min})/p$ 
2:  $i \leftarrow 0$ 
3:  $\mathcal{C}_{\Delta_{\text{safe}}}(t) := \emptyset$ 
4: for  $i < p$  do
5:    $v_i := i\Delta v + v_{\min}$ 
6:    $v_{i+1} := (i+1)\Delta v + v_{\min}$ 
7:    $m_i := (v_{i+1}^2 - v_i^2)/(2|a_{s,\max}|\Delta v)$ 
8:    $\Delta_i := -v_i^2/(-2|a_{s,\max}|) + v_{\min,b}^2(t)/(-2|a_{s,\max,b}|)$ 
9:    $\mathcal{C}_{\Delta_{\text{safe}}}(t).append([x_{\text{lon}}^{(0)}(t) + m_i(x_{\text{lon}}^{(1)}(t) - v_i) + \Delta_i + \delta_{\text{brake}}x_{\text{lon}}^{(1)}(t) \leq s_{\max}(t)])$ 
10: end for
11: return  $\mathcal{C}_{\Delta_{\text{safe}}}(t)$ 
    
```

where $s_{\max,b}(t)$ is the maximum position constraint with respect to the obstacle b . We rearrange (4.16) by using (4.10) to obtain:

$$x_{\text{lon}}^{(0)}(t) + x_{\text{lon}}^{(1)}(t)t_{\text{eva}} \leq s_{\max,b}(t) + \Delta s_b. \quad (4.17)$$

For the preceding obstacle $b \in \mathcal{B}$ in the ego vehicle's lane (or target lane), we add (4.17) to the longitudinal optimization problem, resulting in one additional constraint in the longitudinal optimization problem.

Besides the longitudinal optimization problem, we also have to modify the lateral optimization problem by adding an additional constraint. Since the evasive distance Δ_{eva} (cf. (4.10)) depends on the lateral evasive distance d_{eva} (required distance to fully enter an adjacent lane) for a chosen lateral position, we have to constrain the admissible lateral positions d of the ego vehicle. Without loss of generality, let us assume that the ego vehicle intends to swerve to the adjacent left lane, as illustrated in Fig. 4.10. If Δ_{eva} in the longitudinal motion has been computed with respect to the lateral evasive distance $d_{\text{eva},1}$, the ego vehicle is not allowed to occupy lateral positions $d < 0$. The positions $d < 0$ correspond to driving right of the reference path Γ and increase the required lateral evasive distance for the ego vehicle (e.g., $d_{\text{eva},0}$ in Fig. 4.10).

Consequently, the ego vehicle needs to keep a larger distance to the preceding obstacle b if the ego vehicle occupies lateral positions $d < 0$. Thus, we need to restrict the lateral motion of the ego vehicle depending on the chosen evasive distance d_{eva} . We use $d_{\text{eva},\Gamma}$ to denote the minimum allowed lateral position of the ego vehicle with respect to the lateral evasive distance d_{eva} and add the constraint

$$x_{\text{lat}}^{(0)}(t) \geq d_{\text{eva},\Gamma} \quad (4.18)$$

to the lateral optimization problem. Note that the inequality needs to be changed if the ego vehicle intends to swerve to the adjacent right lane.

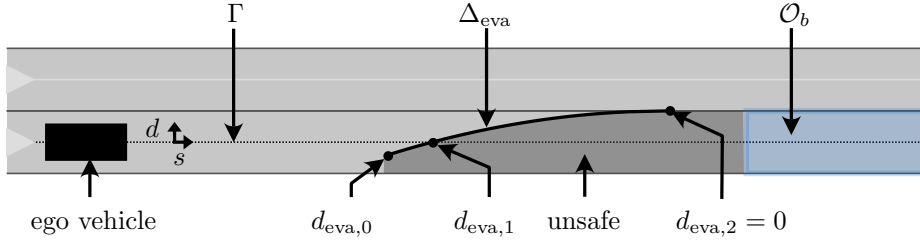


Figure 4.10: Lateral constraint for evasive distances. The ego vehicle must respect the evasive distance Δ_{eva} to be able to swerve into the adjacent left lane (unsafe area denoted in dark gray). If the longitudinal motion considers evasive distances for the lateral distance $d_{\text{eva},1}$, the ego vehicle must maintain lateral position $d \geq 0$ to ensure safety. Otherwise, the required lateral evasive distance increases (e.g., illustrated for $d_{\text{eva},0}$).

Algorithm 4 evasiveDistanceConstraints

Input: $t_{\text{eva}}, v_b, a_{s,\text{max},b}$

Output: Set of linear evasive distance constraints $\mathcal{C}_{\Delta_{\text{eva}}}(t)$

- 1: $\mathcal{C}_{\Delta_{\text{eva}}}(t) := \emptyset$
 - 2: $t_b := \min(t_{\text{eva}}, v_b/|a_{s,\text{max},b}|)$
 - 3: $\Delta s_b := v_b t_b - \frac{1}{2}|a_{s,\text{max},b}|t_b^2$,
 - 4: $\mathcal{C}_{\Delta_{\text{eva}}}(t).\text{append}([x_{\text{lon}}^{(0)}(t) + x_{\text{lon}}^{(1)}(t)t_{\text{eva}} \leq s_{\text{max}}(t) + \Delta s_b])$
 - 5: **return** $\mathcal{C}_{\Delta_{\text{eva}}}(t)$
-

The added constraint (4.18) restricts the feasible state space of the lateral optimization problem. In order to reduce this effect on the feasible states, one can consider the largest lateral evasive distance d_{eva} ($d_{\text{eva},0}$ in Fig. 4.10). In this case, constraint (4.18) simply restricts the ego vehicle to not swerve to the incorrect adjacent right (or left) lane. Since the maneuver option (i.e., passing sides of obstacles) is pre-determined in convex motion planners, the constraint usually does not influence the lateral solution anymore. However, choosing larger values of d_{eva} result in larger evasive distances in the longitudinal optimization problem. This is a tradeoff, since the lateral motion is not yet determined.

The algorithmic realization of the linearization is shown in Alg. 4. This algorithm computes the set of linear evasive distance constraints for the longitudinal optimization problem and a given preceding obstacle $b \in \mathcal{B}$. The velocity and the maximum acceleration of b are given by v_b and $a_{s,\text{max},b}$, respectively. The obtained set of constraints $\mathcal{C}_{\Delta_{\text{eva}}}(t)$ needs to be added to the longitudinal optimization problem.

4.6 Numerical Experiments

In this section, we compute the under-approximation of invariably safe sets for different scenarios and demonstrate its usage for motion planning. We implement Alg. 2 in MATLAB R2015b as well as in Python 3.7, using a computer with an Intel i5-4260U 1.4GHz processor and 8GB of DDR3 1600MHz memory. For the MATLAB implementation, we use the MPT toolbox V3.0 [276] to visualize \mathcal{S} by approximating it with halfspaces. For brevity, we omit the time dependency of \mathcal{S} if it is not necessary in the given context. The parameters of the presented scenarios are summarized in App. A.4. Videos of the simulation results can be found in the supplementary materials of this thesis (cf. App. A.9).

4.6.1 Verifying intended trajectories for infinite horizons

To demonstrate the verification of trajectories for infinite time horizons (cf. Def. 21), we investigate an urban traffic scenario (cf. Fig. 4.11). The scenario consists of two lanes with opposite driving directions (direction of travel is indicated by arrows). The lane of the ego vehicle is occupied by four other traffic participants $b_i, i \leq 4$ (parameters given in Tab. A.5). In this scenario, the ego vehicle has the task of overtaking the preceding vehicle b_1 .

Fig. 4.12 illustrates the feasible velocity profile and the speed limit along the ego vehicle's lane. The maximum feasible velocity is used to compute the under-approximation of \mathcal{S} . The ego vehicle plans two overtaking trajectories $u_1([t_0, t_h])$ and $u_2([t_0, t_h])$ with equal time horizons $t_h = 3.5$ s but differing goal states and velocities, 10.3 m/s and 11.1 m/s, respectively. The final positions of both trajectories are shown as red crosses in Fig. 4.11.

Considering Def. 21, we compute the under-approximation of \mathcal{S} for the initial scenario at $t_0 = 0$ s and for the end of the planning horizon at $t_h = 3.5$ s by utilizing the predicted occupancy sets and Alg. 2. The obtained under-approximations are visualized in Fig. 4.13 as projections onto the s - v plane. Less than 0.3 ms is required for the computation of the under-approximation in this scenario and to check whether $x(t_h) \in \mathcal{S}(t_h)$. It should be noted that the predicted occupancy of vehicle b_1 is shorter due to assumption Λ_{over} (cf. Tab. 2.1).

The required safe distance for a given velocity v can be directly extracted from \mathcal{S} by determining the distance from the boundary point of \mathcal{S} at v to the occupancy of an obstacle b . Our proposed approach is also able to consider safe distances to following vehicles (e.g., for overtaking). This is illustrated for vehicle b_2 in Fig. 4.13 (cf. label of safe distance to b_2): we remove states x from \mathcal{S} that have lower velocities than the velocity of obstacle b_2 , and corresponding relative distances that are smaller than the required safe distance to obstacle b_2 .

In Fig. 4.13b, the final states $x_1(t_h) = (37.2 \text{ m}, 0 \text{ m}, 10.3 \text{ m/s})^T$ and $x_2(t_h) = (39.9 \text{ m}, 0 \text{ m}, 11.1 \text{ m/s})^T$ of $u_1([t_0, t_h])$ and $u_2([t_0, t_h])$, respectively, are indicated with red crosses. Both trajectories are collision-free within the time interval $[0, t_h]$ (cf. Def. 9). However, we note that $x_1(t_h) \in \mathcal{S}(t_h)$, but $x_2(t_h) \notin \mathcal{S}(t_h)$. The ego

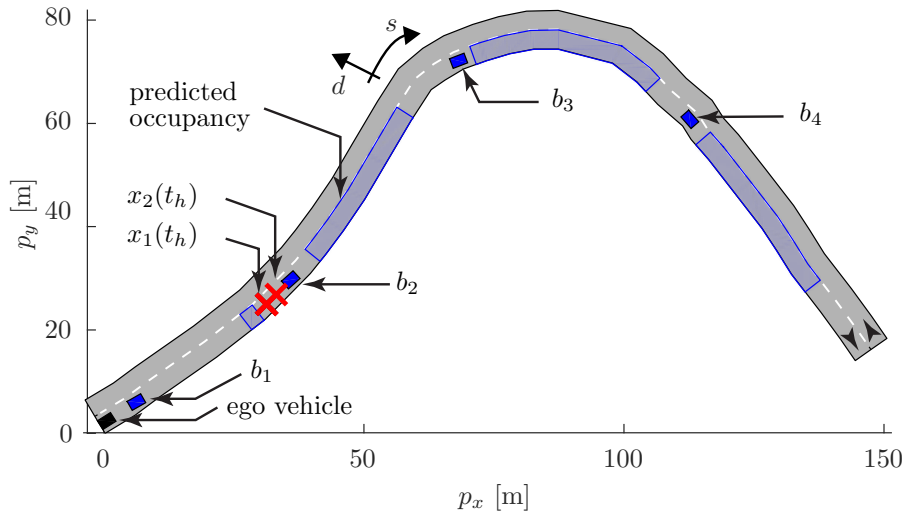


Figure 4.11: Urban scenario for verification (ZAM_Urban-1.1.S-1). The figure shows the initial occupancy of dynamic obstacles $b_i, i \leq 4$ and their predicted occupancies at $t_h = 3.5$ s (light blue). Positions corresponding to the final states $x_1(t_h)$ and $x_2(t_h)$ of the two overtaking trajectories of the ego vehicle are shown in red. ©2018 IEEE.

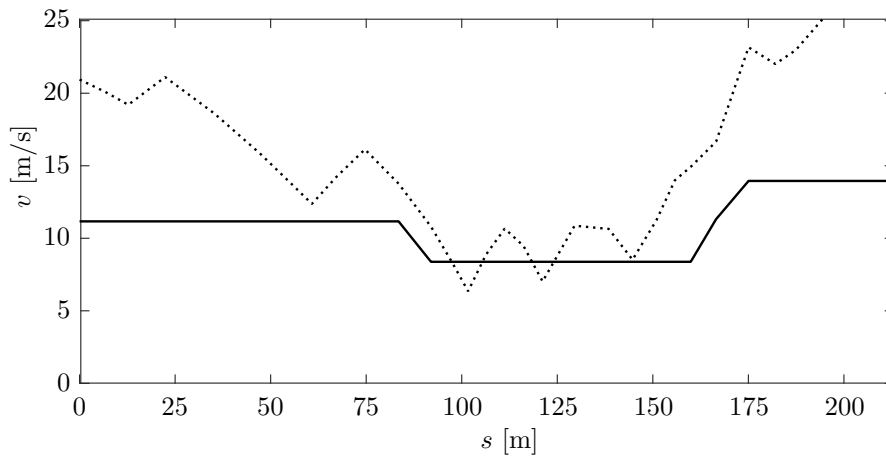


Figure 4.12: Feasible velocity profile. The dashed line illustrates the maximum velocity considering the curvature and the solid line denotes the speed limit along the ego vehicle's lane. The ego vehicle has to respect the minimum of both velocities. ©2018 IEEE.

vehicle can only come to a stop without colliding with vehicle b_2 if it executes the invariably safe input trajectory $u_1([t_0, t_h])$, which is enclosed in $\mathcal{S}(t_h)$. Otherwise, by executing trajectory $u_2([0, t_h])$, the ego vehicle inevitably collides with obstacle b_2 after the finite planning horizon. We validate our findings by simulating the scenario for times $t > t_h$.

4.6.2 Evaluating the tightness of the under-approximation

We evaluate the tightness of our under-approximation in Sec. 4.6.1 by computing an over-approximation using reachability analysis as shown in [249]. To this end, we calculate the over-approximative reachable set of the ego vehicle along its lane and determine at which velocity the reachable set becomes empty (i.e., the ego vehicle eventually collides in all possible trajectories). Since the approach in [249] does not incorporate the feasible velocity profile (cf. Fig. 4.12), the obtained over-approximation also considers velocities above the speed limit. To better compare the over- and under-approximations, we normalized the over-approximation by incorporating the speed limit - in other words, if the velocity within the over-approximation is larger than the velocity constraint (cf. Fig. 4.12), then the velocity constraint is shown in Fig. 4.13. As a result, the normalized over-approximation provides us with the set of states for which it may still be possible to find a collision-free trajectory under the given velocity constraints.

The boundary of the over-approximation is illustrated as a dashed line in Fig. 4.13. The computation of the set's boundary takes about 1 s per sampled longitudinal position s . States that are not enclosed in the over-approximation indicate the non-existence of a collision-free evasive maneuver under the given velocity constraints. The boundary of the exact maximal invariably safe set \mathcal{S} must be located between the boundary of our proposed under-approximation and the computed over-approximation. We can investigate the tightness of our approximation by computing the maximum gap between both sets. The largest deviation between the under-approximation and over-approximation is $\Delta s = 3.1$ m for $v = 13.9$ m/s. This deviation is less than a typical vehicle length, and our under-approximation can thus be considered as tight.

4.6.3 Urban T-junction

Invariably safe sets can also be applied to more complex scenarios as shown in Fig. 4.14a, where the ego vehicle approaches a T-junction with three other vehicles $b_i, i \leq 3$ (parameters given in Tab. A.6). Even if the intended lane of the ego vehicle (i.e., driving straight or turning right) is not yet known in the driving strategy, we are able to consider both lane options during the computation of our invariably safe sets. Without loss of generality, we assume that the driving strategy decides the lane at $t = 2$ s. With our approach, we are able to ensure that the ego vehicle remains safe for both lane options. We compute the under-approximation $\mathcal{S}^{\text{straight}}(t)$ and

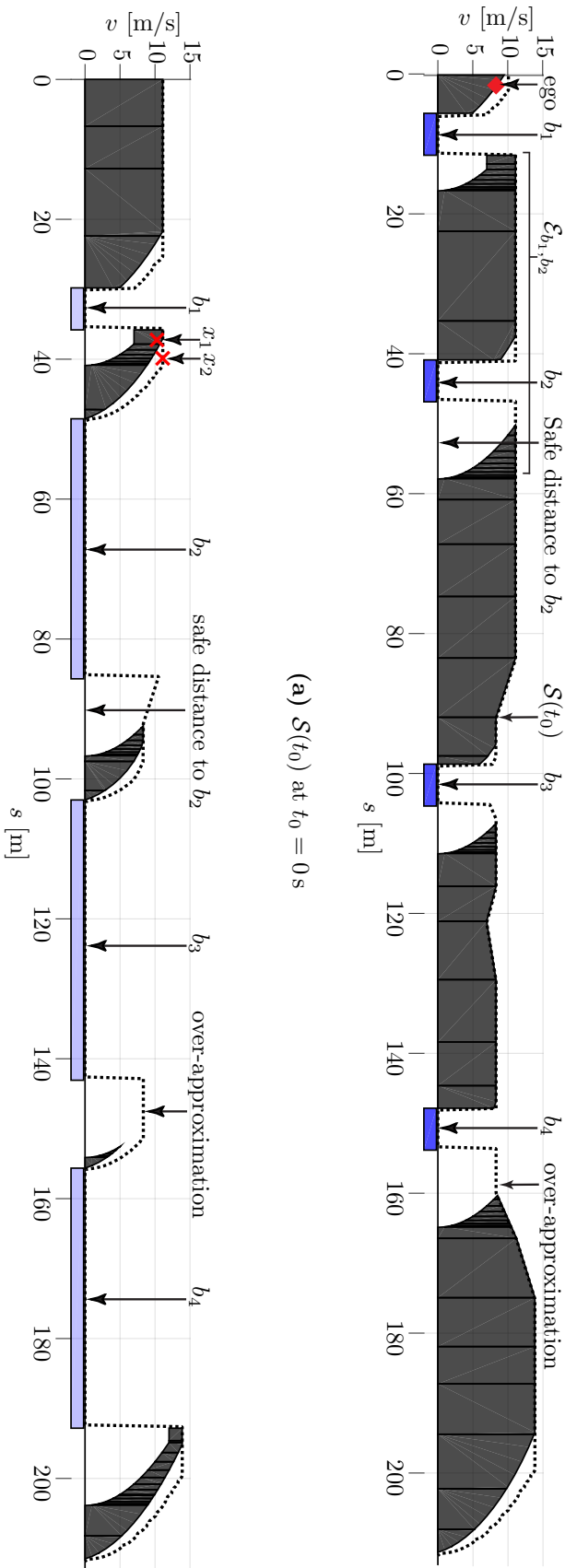


Figure 4.13: Computed invariably safe sets. (a) The under-approximation of $S(t_0)$ (gray) for the scenario in Fig. 4.11 at $t_0 = 0$ s. (b) The under-approximation at $t_h = 3.5$ s. The sets are shown as a projection onto the s - v -plane. The over-approximation (computation described in Sec. 4.6.2) is shown as a dashed line, occupancies of obstacles in blue, and initial and final ego positions in red. ©2018 IEEE.

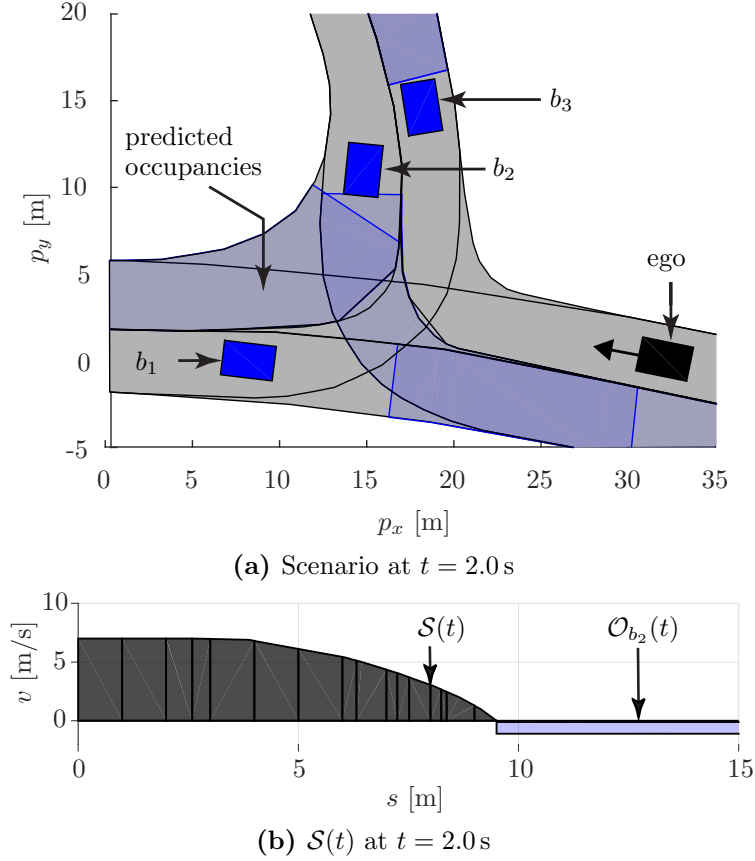


Figure 4.14: Invariably safe set for T-junction (DEU_Ffb-2.2.S-1). (a) The initial scenario with dynamic obstacles $b_i, i \leq 3$, and their predicted occupancies at $t = 2.0$ s (light blue). (b) The invariably safe set $\mathcal{S}(t)$, which ensures safety for both lane options: driving straight and turning right. ©2018 IEEE.

$\mathcal{S}^{\text{right}}(t)$ for the lane options straight and right at $t = 2$ s, respectively. We apply Alg. 2 to each lane option.

Finally, we determine the invariably safe set $\mathcal{S}(t) = \mathcal{S}^{\text{straight}}(t) \cap \mathcal{S}^{\text{right}}(t)$ by computing the intersection, since the ego vehicle should be safe for both lane options. The set $\mathcal{S}(t)$ is visualized in Fig. 4.14 as a projection onto the s - v plane. The occupancy of b_2 limits the size of $\mathcal{S}(t)$, since obstacle b_2 is closer to the ego vehicle than obstacle b_3 is. The obtained under-approximation ensures safety for both possible lane options. Thus, if a trajectory of the ego vehicle ends in $\mathcal{S}(t)$, it is still safely able to continue going straight or to turn right.

4.6.4 Determining the existence of fail-safe trajectories

We exploit the property of persistent feasibility in a safety-critical scenario in which the ego vehicle is endangered by a cut-in vehicle that suddenly performs emergency braking (parameters given in Tab. A.7). The goal is to determine whether a fail-safe trajectory to avoid a collision still exists. The ego vehicle is driving in the right

lane of a two-lane highway with $v_{\text{ego}} = 20$ m/s. Vehicle b_1 is driving in the left-adjacent lane at a velocity of $v_1 = 13.5$ m/s. Fig. 4.15a shows the initial positions of both vehicles after the cut-in maneuver of vehicle b_1 for time t_0 . The relative longitudinal distance of b_1 to the ego vehicle is $\Delta s = 15.0$ m. We simulate that vehicle b_1 suddenly performs emergency braking (cf. Fig. 4.15a), exposing the ego vehicle to a potential collision.

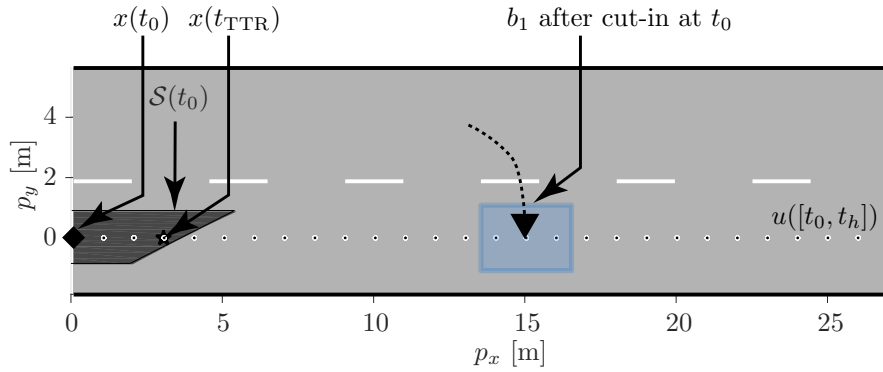
Fig. 4.15a illustrates the computed invariably safe set $\mathcal{S}(t_0)$ at $t_0 = 0$ s. We assess whether the ego vehicle remains safe by checking whether $x_{\text{ego}}(t_0) \in \mathcal{S}(t_0)$ (cf. black diamond). Since the current state of the ego vehicle is invariably safe, a collision-free fail-safe trajectory exists. In our scenario, this fail-safe trajectory corresponds to swerving to the left lane if vehicle b_1 suddenly performs emergency braking.

We evaluate the criticality of the scenario by computing the time-to-react. The intended trajectory $u([t_0, t_h])$ lets the ego vehicle travel with constant velocity along its current lane; it is illustrated in discrete time steps of 50 ms in Fig. 4.15a. We obtain $t_{\text{TTR}} = 0.15$ s (computed using Def. 22), which corresponds to a high criticality, as the ego vehicle needs to react immediately to avoid a collision [277]. Hence, a fail-safe maneuver must be executed as soon as vehicle b_1 starts braking. Fig. 4.15b shows the corresponding fail-safe trajectory, obtained using a sampling-based planner [46]. The fail-safe trajectory starts at $x(t_{\text{TTR}})$ of $u([t_0, t_h])$. Fig. 4.15b shows the positions of both vehicles at $t = t_{\text{TTR}} + t_{\text{eva}} = 0.99$ s, where t_{eva} (cf. (4.8)) is the time required for the ego vehicle to reach the adjacent lane.

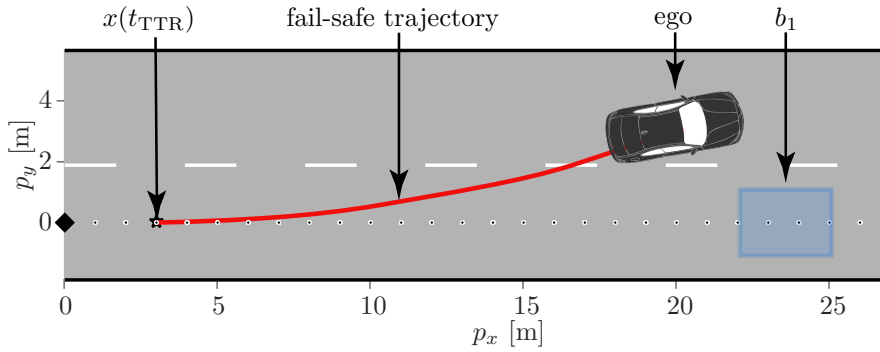
In a next step, we increase the complexity of the scenario by blocking the left lane with a static obstacle, illustrated in Fig. 4.15c. In this case, the ego vehicle can no longer swerve into the adjacent left lane, and we need to compute a new fail-safe trajectory. However, in this situation, a fail-safe trajectory to avoid a collision with b_1 only exists if the ego vehicle is allowed to use the shoulder lane. This fail-safe trajectory is shown in Fig. 4.15c. In these safety-critical situations, we recommend that the ego vehicle is allowed to occupy regions such as the shoulder lane. Thus, this particular problem requires the legislative power to refine the traffic rules for autonomous vehicles.

4.6.5 Safety assessment of machine learning approaches

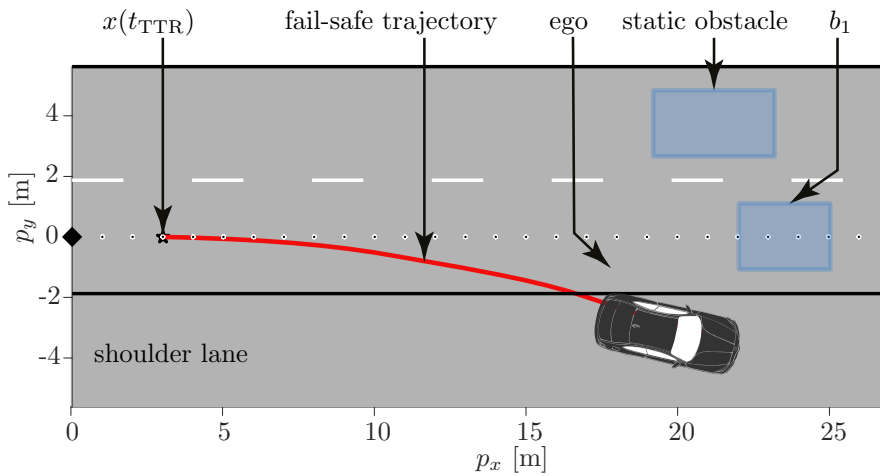
The safety properties of invariably safe sets can also be exploited to verify the safety of machine learning approaches. To this end, we consider a reinforcement learning (RL) agent that learns to perform lane changes in a three-lane highway scenario. In every discrete time step t_k , the RL agent receives the current state x_k of the environment (own state and states of other traffic participants) and the reward φ_k (cf. Fig. 4.16). Given this information, the agent chooses an action $\alpha_k \in \mathcal{U}_{\text{RL}}$ that is executed in the environment. This cycle repeats every time step and over time, the RL agent optimizes its action based on the expected reward of the environment [278]. The agent can choose between three different actions $\alpha_k^i, i \leq 3$: lane keeping ($i = 1$), lane change to the left ($i = 2$), and lane change



(a) Initial scenario at $t_0 = 0$ s



(b) Scenario at $t = t_{TTR} + t_{eva} = 0.99$ s



(c) Blocked lane scenario at $t = t_{TTR} + t_{eva} = 0.99$ s

Figure 4.15: Invariably safe sets in emergency situations. (a) Vehicle b_1 changes to the ego vehicle’s lane, but the ego vehicle (diamond indicates initial state $x(t_0) \in \mathcal{S}(t_0)$ of the ego vehicle) remains within an invariably safe set (gray). (b) A collision with vehicle b_1 can be avoided. (c) If the left lane is occupied by a static obstacle, the ego vehicle can only swerve into the shoulder lane. ©2018 IEEE.

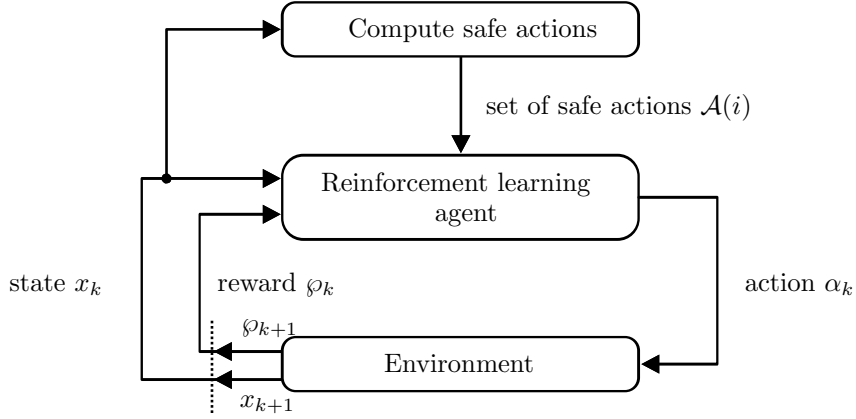


Figure 4.16: Overview of the safe RL approach. The RL agent receives the current state x_k of the environment and performs an action α_k . Based on α_k , the environment generates a new state x_{k+1} and a reward φ_{k+1} for the agent (dashed line). Invariably safe sets are used to compute safe actions from which the RL agent is allowed to choose.

to the right ($i = 3$). For the chosen action α_k^i , the simulator plans and executes a trajectory with a time horizon of 3.5 s. To reward behavior that maximizes the velocity of the agent, we choose the following reward function:

$$\varphi_{k+1} = -|v_{RL,k} - v_{des,k}|,$$

where φ_k is the immediate reward for the RL agent in time step t_k , and $v_{RL,k}$ and $v_{des,k}$ are the absolute velocity of the RL agent and the desired velocity of the RL agent in time step t_k , respectively.

The RL agent learns to choose the optimal action by observing the reward of executing various actions in different situations. In this learning process, the RL agent may also cause collisions with other traffic participants when choosing an un-

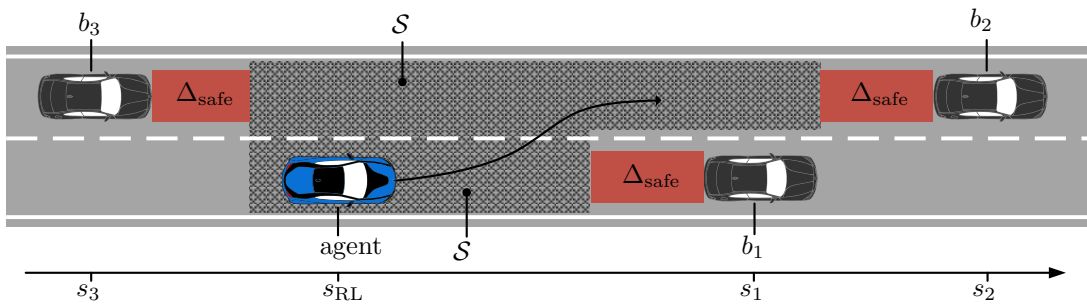


Figure 4.17: Verification of lane change trajectories. The RL agent is safe if it is enclosed in an invariably safe set at all times. The safe set \mathcal{S} is computed with the predicted occupancies of surrounding traffic participants $b_i, i \in \{1, 2, 3\}$. We determine the set of safe actions at any given point in time by checking whether their trajectories are invariably safe. ©2018 IEEE.

safe action. This circumstance requires the simulation environment to be restarted and eliminates the possibility to learn in a real vehicle. Our goal is to eliminate these situations. Therefore, we provide the RL agent with a set of safe actions in every time step t_k . To determine safe actions, we compute invariably safe sets (cf. Fig. 4.17). By checking that the agent is only driving within \mathcal{S} when executing a given action α , we are able to verify whether α is safe prior to execution. The future motion of obstacles is predicted with SPOT. We use $x_\alpha(t)$ to describe the state of the agent while executing action $\alpha \in \mathcal{U}_{\text{RL}}$ over time interval $t \in [t_k, t_{k+1}]$, where $t_{k+1} - t_k$ is fixed by the simulation environment. The set of safe actions $\mathcal{U}_{\text{RL, safe}}(t_k) \subseteq \mathcal{U}_{\text{RL}}$ for a time step t_k is defined as

$$\mathcal{U}_{\text{RL, safe}}(t_k) := \{a \in \mathcal{U}_{\text{RL}} \mid \forall t \in [t_k, t_{k+1}] : x_\alpha(t) \in \mathcal{S}(t)\}.$$

We assume that the agent starts in an invariably safe state. Then, we constrain the agent to always choose safe actions $\alpha \in \mathcal{U}_{\text{RL, safe}}(t_k)$ at each time step t_k .

The performance of the RL agent is evaluated in 10 selected simulated highway traffic scenarios [13]. Each scenario lasts 500.5 s, which means that the RL agent needs to make 143 decisions (the duration of any decision is 3.5 s, which is a fixed parameter in the simulation environment). In each scenario, around 50 other vehicles occupy the highway, which is modeled as a ring such that vehicles do not leave it. We create realistic traffic situations with high traffic densities by computing the number of other vehicles considering the length of 1255 m and the width 11.25 m of the highway. The desired velocity of the RL agent is set to $v_{\text{des}} = 19.5$ m/s (the curvature of the highway constrains the maximal velocity to $v_{\text{max}} = 24$ m/s). We randomly position each surrounding traffic participant on the highway with random initial velocities, and we assign arbitrary desired velocities $10 \text{ m/s} < v < v_{\text{max}}$ to each participant. In each scenario, other traffic participants start from a random position in one of the three highway lanes with a random starting velocity, and they aim to maintain or reach their desired velocity. The RL agent does not have any information about the intentions of the other traffic participants. They change lanes and velocities according to a rule-based system that is part of the simulation environment.

During the simulations, we observe that the RL agent achieves an average velocity of 17.3 m/s over all 10 scenarios without causing collisions. Fig. 4.18a shows excerpts of the simulations when the RL agent is constrained to only execute safe actions. To demonstrate the effectiveness of the safety layer, we let the trained RL agent drive in the same 10 test scenarios with the safety layer turned off. Tab. 4.1 summarizes the results when the safety verification is turned off. Here, the agent collides in 9 out of 10 scenarios, on average after one third of the simulation time has passed. Fig. 4.18b illustrates excerpts from scenarios with a collision.

In contrast, if only safe actions are executed, the RL agent never produces a collision. Thus, by utilizing our invariably safe sets, we can ensure that RL agents never cause collisions, which enables one to use reinforcement learning in real vehicles. In addition, it helps the RL agent’s strategy to reach convergence faster, since the agent does not need to learn collision avoidance at the same time.

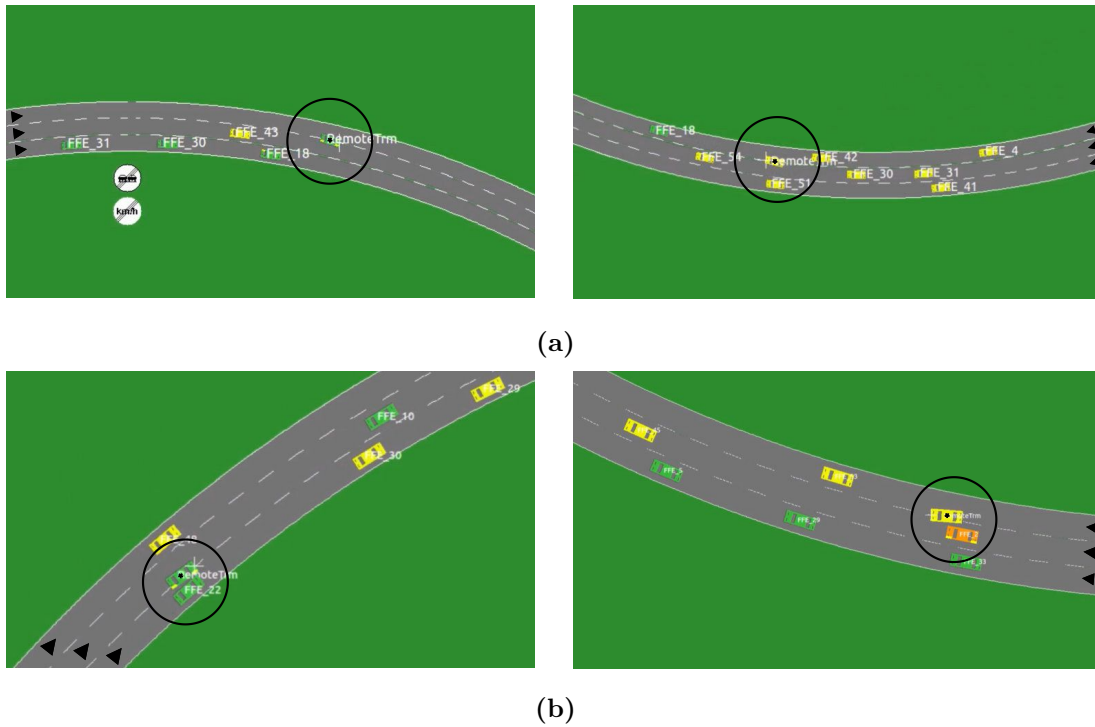


Figure 4.18: Simulation results without and with the safety layer. (a) When the RL agent (circled and highlighted with a black dot) can only choose provably safe actions, it never causes collisions during driving. (b) The RL agent (circled) causes collisions during lane changes if the safety layer is turned off. ©2018 IEEE.

Table 4.1: Performance of the RL agent. The RL agent produces collisions when its allowed to choose any action. ©2018 IEEE.

Test simulation	Seconds until collision
1	249
2	96
3	no crash
4	159
5	30
6	208
7	412
8	219
9	40
10	327

4.7 Summary

This chapter introduced invariably safe sets as a technique to ensure persistent feasibility of planned motions. After briefly presenting the problem of infinite time horizon planning, we gave an overview of the major concepts: inevitable collision states and controlled invariant sets. By using recursion, we defined states as safe if collision-free trajectories to another safe state exist. We showed how one can compute these invariably safe states with backward reachability analysis. The proposed approach makes determining invariably safe sets computationally tractable. Finally, we demonstrated how invariably safe sets can be transformed into linear constraints for use in the convex optimization problems of fail-safe planning.

For online usage, we demonstrated how one can compute an under-approximation of invariably safe sets. To this end, we exploited safe and evasive distances. The former ensures that the ego vehicle is able to compute a collision-free braking maneuver, while the latter ensures that the ego vehicle can compute an evasive maneuver to an adjacent lane. We showed that both distances fulfill the property of persistent feasibility. The proposed under-approximation can be computed with linear time complexity (a few milliseconds on a standard computer). Thus, the proposed approach can be used during the operation of autonomous vehicles.

Invariably safe sets can be used to provide strong safety guarantees for intended motion plans. By ensuring that the final state of a given collision-free trajectory is enclosed in an invariably safe set (necessary condition), the ego vehicle is able to remain collision-free for an infinite time horizon. We categorize the safety of trajectories according to the safety of their states. For instance, trajectories may start in an invariably safe set but exit the set at some point in time, denoted as the time-to-react. In essence, when the ego vehicle follows the trajectory, the time-to-react denotes the time until the existence of a collision-free maneuver is guaranteed.

Furthermore, we highlighted the benefits of invariably safe sets for motion planning in different numerical experiments. For instance, we demonstrated how the ego vehicle can check whether given trajectories are invariably safe. Trajectories that are not invariably safe may result in collisions with other traffic participants, as depicted in an example scenario. Based on a computed over-approximation for the same scenario, we were able to show that the obtained under-approximation is tight. Invariably safe sets can also be used if the intended lane of the ego vehicle is not yet decided. In these cases, invariably safe sets are computed for each lane option and the intersection of these sets ensures safety for all possible lane options of the ego vehicle. To demonstrate the power of invariably safe sets in complex systems, we used invariably safe sets to guarantee the safety of a reinforcement learning agent. By employing the proposed approach, the agent never caused a collision in all conducted simulations, whereas it caused collisions in 90% of simulations when actions were not invariably safe.

Invariably safe sets are a powerful technique to guarantee persistent feasibility of motion plans, which drastically enhances the safety of autonomous vehicles. In

contrast to existing approaches, invariably safe sets represent the first technique of its kind that can be computed in real-time and, most importantly, in dynamic environments as well. This real-time capability makes it possible to verify given trajectories for an infinite horizon during operation of the vehicle in arbitrary traffic situations. If a state is enclosed within an invariably safe set, it is guaranteed that a safe solution can be found (e.g., a braking maneuver). Moreover, the proposed sets also consider evasive maneuvers and safe vehicle following. The sets incorporate the dynamics of other traffic participants and the road geometry. These properties allow the application of invariably safe sets in urban as well as highway environments. In safety-critical situations, the under-approximation of invariably safe sets can be used to quickly check whether a collision-free maneuver still exists. In addition, the sets can be used to assess the criticality of traffic situations. The criticality assessment allows vehicles to determine the remaining time until a fail-safe trajectory needs to be executed. This time can be used to further increase the tightness of the under-approximation over time.

So far in this thesis, we have shown how the ego vehicle can compute fail-safe trajectories that ensure safety for a finite time horizon and how trajectories can be verified over infinite time horizons. In the next chapter, we present an online verification framework that combines both techniques to guarantee the safety of the ego vehicle at all times. With this framework, we are able to ensure legal safety during the operation of the vehicle.

5 Online Safety Verification of Arbitrary Motions

In this chapter, we demonstrate how fail-safe trajectories (cf. Ch. 3) and invariably safe sets (cf. Ch. 4) can be used to verify the safety of planned motions during the operation of autonomous vehicles. First, Sec. 5.1 introduces general structures of motion planning frameworks in autonomous vehicles. In Sec. 5.2, we demonstrate the procedure of our proposed fail-safe motion planning layer and its integration into existing motion planning frameworks. Subsequently in Sec. 5.3 and 5.4, we present the process of the trajectory verification in detail. We also show how invariably safe sets are formulated as linear constraints and are integrated in the linear-quadratic optimization problems of fail-safe trajectories. Conclusions are provided in Sec. 5.5. The content of this chapter is mainly based on the author’s publications [2, 3, 14].

5.1 Introduction to Motion Planning Frameworks

Various planning frameworks have been introduced over the years. However, the vast majority use a similar framework structure for motion planning [135–137]. This framework structure is inspired by the divide-and-conquer principle, namely, dividing difficult-to-solve problems into easier-to-solve subproblems. Applied to the motion planning problem, multiple layers (composed of multiple modules) are created to solve dedicated subtasks of steering an autonomous vehicle from a start to a goal pose. Usually, the general architecture is split into three layers: 1) perception (incl. localization) modules, 2) planning modules, and 3) control modules. This popular *sense-plan-act* architecture is shown in Fig. 5.1.

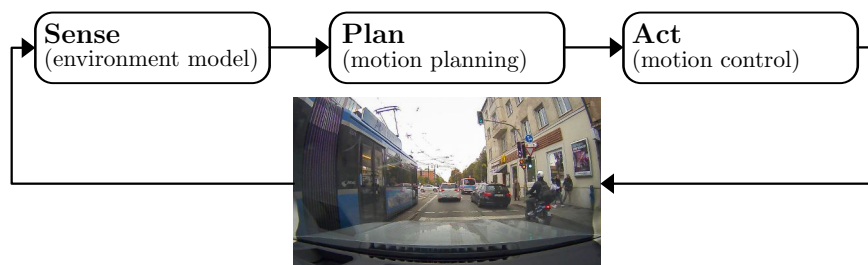


Figure 5.1: Sense-plan-act architecture. Robots first sense their environment through sensors. Subsequently, they plan and execute a motion in the environment.

An advantage of this architecture is that each layer is only responsible for a specific task. Furthermore, the information flow between layers is usually observable, directed, and modifiable; for example, trajectories can be exchanged before sent to the controller.

Conversely, recent efforts have tried to model the motion planning task with just one module. These efforts are mainly driven by the field of end-to-end learning [148, 158, 279, 280]. Here, sensor data is directly converted into actuator commands for the vehicle by a learned policy. This policy can initially be learned with millions of examples offline and subsequently improved online. However, the disadvantage is that decisions are not comprehensible [40, 41] - that is, one cannot determine why a control command has been chosen by the policy. Hence, intermediate results, such as an environment model or intended trajectories, cannot be extracted within this type of framework.

5.2 Integration in Motion Planning Frameworks

The proposed online safety verification framework works with any motion planning framework that provides information about obstacles in the environment and the intended trajectories [120]. Fig. 5.2 illustrates its integration between the planning and the control layers of the vehicle framework. Our safety framework is composed of modules for set-based prediction (cf. Sec. 2.4), invariably safe set computation (cf. Ch. 4), and fail-safe motion planning (cf. Sec. 5.3). The framework only requires the planned trajectories of the vehicle and the environment model with

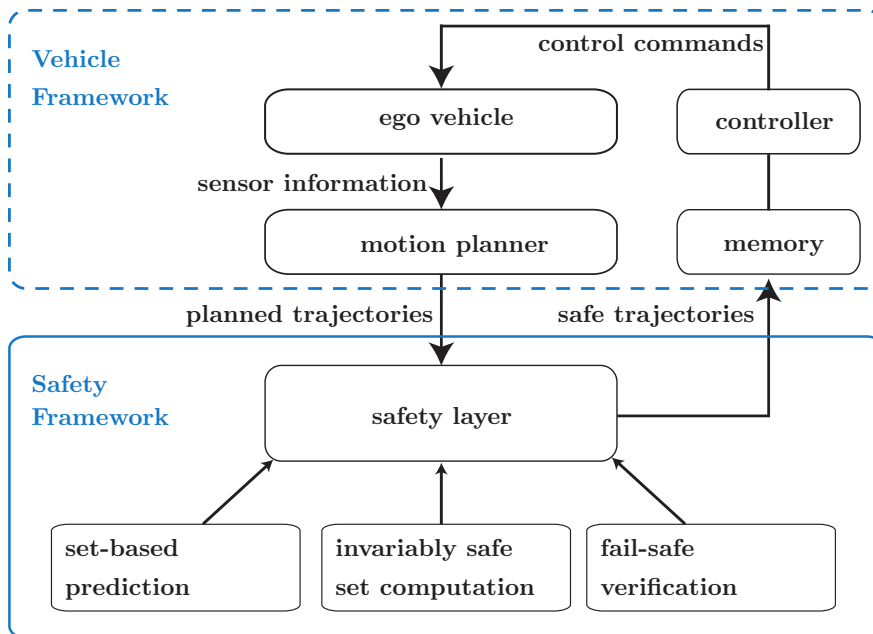


Figure 5.2: Proposed online safety framework. The figure shows how the proposed safety framework can be integrated into an existing vehicle framework.

detected surrounding obstacles (including uncertainties) as inputs. This design allows us to exchange the vehicle framework at any time without sacrificing safety, since our safety framework will verify on the fly whether the proposed motions are safe before execution. Another advantage of this design is that only the safety layer needs to be certified (cf. ISO 26262 [281]), not the techniques developed for planning intended trajectories. Hence, developers can change and update the motion planner of the vehicle framework at any time. Furthermore, our safety layer works with arbitrarily planned trajectories and can even ensure safety for machine learning approaches, which are usually difficult to certify [137].

Another core idea of our safety framework is the redundant memory module (cf. memory module in Fig. 5.2), which stores verified trajectories for execution. In case of any malfunction, where new trajectories cannot be obtained during runtime (e.g., due to an error in the computing hardware, including our safety layer), the autonomous vehicle remains safe, since it can execute the previously verified trajectory that is still stored in redundant memory. Moreover, by using a redundant memory module, a highly redundant computing hardware for the safety layer becomes obsolete, reducing the number of hardware components that require certification.

5.3 Details of the Verification Technique

Here, we briefly review the basic idea of our proposed online verification technique: in each planning cycle of the vehicle, we predict the feasible, legal future motion of obstacles in the environment. Subsequently, we compute fail-safe trajectories that are collision-free with the predicted and possibly occupied spaces in the environment. The fail-safe trajectories are constrained to end in an invariably safe state to ensure safety beyond the planning horizon. As a result, with fail-safe trajectories, we are able to ensure legal safety (cf. Sec. 2.4) during the operation of the vehicle.

In the following, we describe the technical details of computing fail-safe trajectories for consecutive planning cycles. We denote intended trajectories with \mathfrak{I} and fail-safe trajectories with \mathfrak{F} . Fig. 5.3 illustrates how we guarantee legal safety with fail-safe trajectories: in each planning cycle $c \in \mathbb{N}_+$, lasting from t_{c-1} to t_c , the ego vehicle computes an intended trajectory \mathfrak{I}_c as $x_{\mathfrak{I}}([t_c, t_c + t_{\mathfrak{I}}])$ starting at time t_c with planning horizon $t_{\mathfrak{I}}$ using an arbitrary motion planner. As discussed in Sec. 3.1, the predicted occupancy sets of other traffic participants become increasingly larger for longer time horizons due to growing uncertainties. Thus, \mathfrak{I}_c is often not safe for the whole time horizon $t_{\mathfrak{I}}$. Therefore, we only verify a short part of the intended trajectory \mathfrak{I}_c by exploiting the time-to-react:

Proposition 4 (Safe Part of Intended Trajectory) *Based on the computed time-to-react t_{TTR} (cf. Def. 22) of the intended trajectory \mathfrak{I}_c , the safe part of \mathfrak{I}_c is given by $x_{\mathfrak{I}_c}([t_c, t_{c+1}])$ with $t_{c+1} = t_{\text{TTR}}$, denoted as $\mathfrak{I}_c^{\text{safe}}$.*

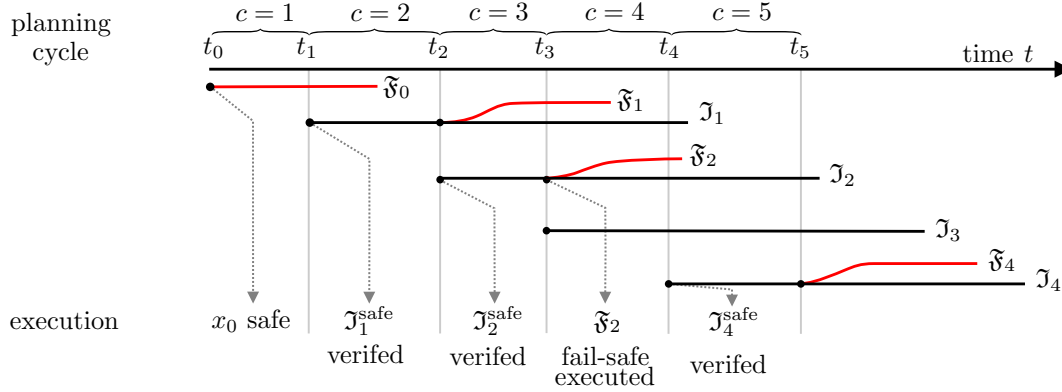


Figure 5.3: Visualization of the verification approach. In planning cycle c , an intended trajectory \mathcal{J}_c is verified as safe if we can compute a fail-safe trajectory \mathfrak{F}_c for it. The ego vehicle is then allowed to execute the safe part $\mathcal{J}_c^{\text{safe}}$ of \mathcal{J}_c at t_c . If a trajectory cannot be verified (e.g., \mathcal{J}_3 in $c = 3$), the previous fail-safe trajectory \mathfrak{F}_{c-1} needs to be executed. As soon as a new trajectory is verified, the safe part of the intended trajectory can be executed again. This verification approach ensures legal safety of the ego vehicle during its operation.

Proof *By definition, the trajectory $x_{\mathcal{J}_c}([t_c, t_{c+1}])$ is an invariably safe trajectory that is collision-free considering any feasible, legal future behavior of obstacles, and that allows the vehicle to remain safe.* ■

The safe part $x_{\mathcal{J}_c}([t_c, t_{c+1}])$ of \mathcal{J}_c allows the ego vehicle to remain safe by executing a consecutive fail-safe trajectory (computation described in Sec. 5.4) starting at t_{c+1} , that is, when the next planning cycle $c + 1$ ends. The fail-safe trajectory \mathfrak{F}_c with time horizon $t_{\mathfrak{F}}$ continues $\mathcal{J}_c^{\text{safe}}$ at time t_{c+1} as $x_{\mathfrak{F}}([t_{c+1}, t_{c+1} + t_{\mathfrak{F}}])$ (cf. Fig. 5.3). While both $\mathcal{J}_c^{\text{safe}}$ and \mathfrak{F}_c are provably collision-free with respect to any legal motion of other traffic participants, only \mathfrak{F}_c guides the ego vehicle to a safe terminal set. Fail-safe trajectories transition the vehicle to a standstill in safe areas or allow the vehicle to switch into safe vehicle following (i.e., keeping a safe distance to another vehicle in this lane by using a verified adaptive cruise control system [268]); both possibilities are ensured by constraining fail-safe trajectories to end in an invariably safe set. We now define verified intended trajectories \mathcal{J}_c :

Definition 23 (Verified Intended Trajectory) *A given intended trajectory \mathcal{J}_c is verified as safe if $\mathcal{J}_c^{\text{safe}}$ and \mathfrak{F}_c have been correctly computed.*

Without loss of generality, we assume that the ego vehicle is initially in an invariably safe state in which it can remain (cf. x_0 in Fig. 5.3), such as a standstill in a safe area. This assumption allows us to define the initial fail-safe trajectory \mathfrak{F}_0 (cf. Fig. 5.3):

Definition 24 (Initial Fail-Safe Trajectory \mathfrak{F}_0) *When starting in an invariably safe state, the initial fail-safe trajectory \mathfrak{F}_0 is defined as an input trajectory such that the ego vehicle remains in this safe state.*

The initial fail-safe trajectory is usually implicitly given; for example, if the ego vehicle starts at a standstill, we choose an input trajectory to remain at a standstill.

Immediately after a given intended motion \mathcal{I}_1 is verified as safe, the ego vehicle is allowed to execute the safe part of the intended trajectory $\mathcal{I}_1^{\text{safe}}$ at t_1 (cf. Fig. 5.3, planning cycle $c = 1$). In every planning cycle c , the proposed verification technique tries to verify the next intended trajectory \mathcal{I}_c . If the fail-safe trajectory \mathfrak{F}_c is obtained prior to t_c , $\mathcal{I}_c^{\text{safe}}$ is released for execution (cf. Fig. 5.3, planning cycle $c \in \{1, 2, 4\}$). Yet, if the verification fails, for instance when the intended trajectory \mathcal{I}_c leads to an unsafe situation, the ego vehicle executes the previously computed fail-safe trajectory \mathfrak{F}_{c-1} (cf. Fig. 5.3, planning cycle $c = 3$). It should be noted that previously computed fail-safe trajectories \mathfrak{F} remain valid by design, since the set-based prediction already anticipates all feasible, legal motions of other traffic participants. Even in the case in which no intended motion \mathcal{I}_c is provided prior to t_c , for example due to a time-out or hardware fault, the ego vehicle still has a fail-safe trajectory \mathfrak{F}_{c-1} . In the event that the vehicle has to execute a fail-safe trajectory \mathfrak{F}_c , the intended trajectory is continuously replanned. As soon as a new intended trajectory \mathcal{I}_{c+i} , $i \in \mathbb{N}_+$, is verified, the ego vehicle can recover from \mathfrak{F}_c to the new intended trajectory $\mathcal{I}_{c+i}^{\text{safe}}$ (cf. Fig. 5.3, planning cycle $c = 4$).

Alg. 5 summarizes the main computation steps of our safety layer. In line 1, we empty the trajectory memory of the ego vehicle, since each piece of stored trajectory information on the memory will be executed by the vehicle (cf. Sec. 5.2). Subsequently, we push the initial fail-safe trajectory \mathfrak{F}_0 (cf. Def. 24) into the memory in lines 2. Thus, the ego vehicle remains safe until we verify an intended

Algorithm 5 safetyLayer

Input: memory, \mathfrak{F}_0

```

1: memory  $\leftarrow \emptyset$ 
2: memory.push( $\mathfrak{F}_0$ )
3:  $c \leftarrow 1$ 
   // Verify newly planned trajectories during operation
4: while  $c \geq 1$  do
5:    $\mathcal{I}_c \leftarrow \text{obtainIntendedTrajectory}(c, \dots)$ 
   // Apply verification procedure as described in Alg. 6
6:    $\mathcal{I}_c^{\text{safe}}, \mathfrak{F}_c \leftarrow \text{verifyTrajectory}(\dots)$ 
7:   if  $\mathfrak{F}_c \neq \emptyset \wedge t < t_c$  then
8:     memory.push( $\mathcal{I}_c^{\text{safe}}, \mathfrak{F}_c$ )
9:   end if
10:   $c \leftarrow c + 1$ 
11: end while

```

trajectory. In lines 5-6, we obtain a new intended trajectory from the vehicle framework for planning cycle c and try to verify it. If a trajectory \mathcal{I}_c has been verified and it starts at a future point in time $t_c > t$ (where t is the current time), the safe part of the intended and the fail-safe trajectory are pushed into the memory for execution (old information prior to t is removed). Lines 5 to 11 are repeated as long as the vehicle is operating or an external signal resets the autonomous vehicle (e.g., to manual driving).

It should be noted that intended trajectories can also be planned more frequently, but only verified trajectories can be executed prior to t_c (since a new fail-safe trajectory exists). Moreover, when implementing Alg. 5, computation times and delays have to be explicitly considered, for instance when computing the set-based prediction.

Below, we prove the correctness of our verification scheme with respect to the proposed legal safety specification (cf. Def. 1 and Sec. 2.4).

Theorem 2 (Online Verification) *If the ego vehicle starts in a safe state with the initial fail-safe trajectory \mathfrak{F}_0 , Alg. 5 ensures legal safety in each planning cycle $c \in \mathbb{N}_+$.*

Proof *We prove the theorem inductively.*

Base case ($c = 1$): If we cannot verify \mathcal{I}_1 , the ego vehicle remains safe by executing \mathfrak{F}_0 . Otherwise, both trajectories, $\mathcal{I}_1^{\text{safe}}$ and \mathfrak{F}_1 , are collision-free with respect to all feasible legal behaviors of other traffic participants (cf. Def. 9). Since \mathfrak{F}_1 is an invariably safe trajectory (cf. Def. 21), it keeps the ego vehicle within a safe state according to the legal safety specification at all times, for instance by guiding the vehicle to standstill or safe vehicle following [268]. By executing $\mathcal{I}_1^{\text{safe}}$ (and \mathfrak{F}_1), the ego vehicle is compliant with legal safety.

Inductive step ($c=k$): Assuming that legal safety is guaranteed for an arbitrary planning cycle $c = k$ with random integer $k \in \mathbb{N}_+$, we show that planning cycle $c + 1$ ensures legal safety. We distinguish two cases: 1) \mathcal{I}_{c+1} cannot be verified, and 2) \mathcal{I}_{c+1} can be verified. In the first case, the ego vehicle can simply execute the fail-safe trajectory \mathfrak{F}_c from the previous planning cycle c . This fail-safe trajectory exists, since the verification was successful in the previous planning cycle c (cf. inductive step) and therefore ensures legal safety. For the second case, we can determine a pair $\mathcal{I}_{c+1}^{\text{safe}}$ and \mathfrak{F}_{c+1} that ensures legal safety analogous to the base case. ■

Remark 1 (Safety Properties of Alg. 5) *Alg. 5 ensures that the state of the ego vehicle is invariably safe at all times: $\forall t \geq t_0 : x(t) \in \mathcal{S}(t)$.*

Remark 2 (Length of the Safe Part of \mathcal{I}_c) *Since each \mathcal{I}_c starts along a verified trajectory (either a previous safe part or a fail-safe trajectory), the lower bound of the TTR is $t_{\text{TTR}} = 0$ and the size of each $\mathcal{I}_c^{\text{safe}}$ is never zero. For this reason, one may prefer to continue the computation of \mathfrak{F}_c only if $t_{\text{TTR}} > 0$.*

Remark 3 (Anytime Verification) *Previous verification results or set-based predictions may be reused to verify intended trajectories in an anytime fashion [282].*

5.4 Computation Steps of the Verification Procedure

Fig. 5.4 visualizes the necessary computation steps to verify a given intended trajectory \mathfrak{J} . In the following paragraphs, we briefly describe each step of the verification. We focus on the fail-safe trajectory computation using driving corridors (cf. Sec. 3.4.3). The simpler version using combinatorial enumerations of fail-safe maneuvers (cf. Sec. 3.4.1) can be similarly integrated.

Occupancy prediction In the first step ① (cf. Fig. 5.4) of the trajectory verification, we predict the legal, future motions of each traffic participant $b \in \mathcal{B}$ over time (cf. Sec. 2.4). To this end, we initially retrieve parameters (e.g., maximum acceleration) specific to each type of traffic participant in the environment from a database. Subsequently, we determine the initial occupancy of traffic participants while including measurement uncertainties. Based on the over-approximative reachable set $\bar{\mathcal{R}}$, we compute the dynamically feasible behaviors of each traffic participant (cf. Def. 8). We remove illegal behaviors according to our legal specification to obtain the over-approximative occupancy set $\mathcal{O}(t)$.

Invariably safe set computation Afterwards, we use the obtained occupancy sets to compute invariably safe sets in step ② (cf. Sec. 4.3). Hence, we further incorporate the allowed lanes of the ego vehicle as well as the predicted velocities of each traffic participant $b \in \mathcal{B}$. We use the proposed Alg. 2 (cf. Sec. 4.3) to first determine the invariably safe set \mathcal{S}^1 (of states respecting safe distances) and, subsequently, safe set \mathcal{S}^2 (of states respecting evasive distances). We obtain the under-approximation of the invariably safe set as $\mathcal{S}(t) := \mathcal{S}^1(t) \cup \mathcal{S}^2(t)$.

Drivable area computation In step ③ (cf. Fig. 5.4), we compute the drivable area of the ego vehicle to determine suitable driving corridors for fail-safe trajectory computation (cf. Sec. 3.4). The initial set \mathcal{X}_0 of the computation encloses the last state along the intended trajectory \mathfrak{J} , which is still invariably safe - that is, $\{x_{\mathfrak{J}}(t_{\text{TR}})\} \subseteq \mathcal{X}_0$ (cf. Prop. 4). This initial set is propagated in time as described in Sec. 3.4.2. The propagated set is constrained to only include collision-free states and to occupy allowed lanes (checked through projection onto the position domain). After each propagation, the reachability graph \mathcal{G} is updated. In the last propagation step (when reaching the final time step), the propagated set is further constrained to end in the computed invariably safe sets. The projection of the computed sets onto the position domain yields the drivable area $\mathcal{D}(t)$ of the ego vehicle.

Driving corridor and trajectory optimization Finally, we determine the fail-safe trajectory \mathfrak{F} in step ④. First, we determine a longitudinal driving corridor Ξ_{lon} from the reachability graph \mathcal{G} (cf. Sec. 3.4.3). From this corridor, we obtain the constraints for the longitudinal trajectory optimization, and we conduct this optimization (cf. Sec. 3.2.1 and Sec. 3.3). Afterwards, we select a lateral driving

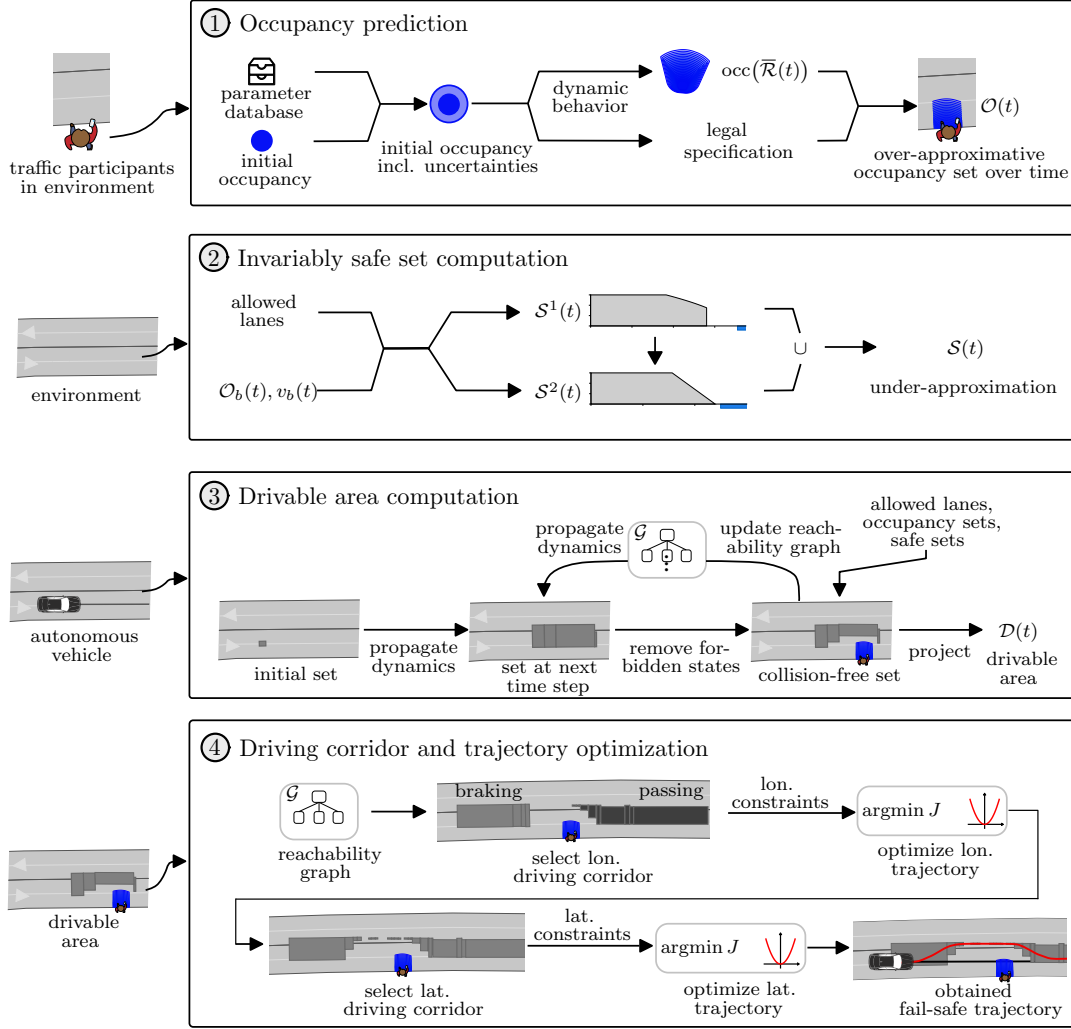


Figure 5.4: Computation steps during the verification. (1) The feasible legal, future behaviors are predicted for each traffic participant in the environment. (2) The under-approximations, $\mathcal{S}^1(t)$ and $\mathcal{S}^2(t)$, of invariably safe sets are computed. (3) The drivable area of the ego vehicle is computed and restricted to end in invariably safe sets. (4) The driving corridors are selected and the fail-safe trajectory is computed.

corridor based on the obtained longitudinal trajectory, and we optimize the lateral motion (cf. Sec. 3.2.2 and Sec. 3.3). Ultimately, we combine both motions to obtain the final fail-safe trajectory \mathfrak{F} .

Algorithmic realization Alg. 6 shows the algorithmic realization of the above steps to verify a given trajectory. The safe part of \mathcal{I}_c and the invariably safe set are computed in lines 2-4. Subsequently, the longitudinal and lateral motions, $x_{\text{lon}}([t_{\text{TTR}}, t_{\text{TTR}} + t_{\mathfrak{F}}])$ and $x_{\text{lat}}([t_{\text{TTR}}, t_{\text{TTR}} + t_{\mathfrak{F}}])$, of the fail-safe trajectory are computed in lines 5-11. Lines 12-14 check whether recombining the longitudinal and lateral motions results in a feasible trajectory (e.g., the lateral optimization problem may be infeasible). The fail-safe trajectory is obtained these lines. The algorithm can also be applied to the verification of a set of intended trajectories. Here, the algorithm is recursively called to obtain the verified trajectory pairs $(\mathcal{I}_c^{\text{safe}}, \mathfrak{F}_c)$ for each trajectory \mathcal{I}_c . The results of the occupancy and invariably safe set computation can be reused in each run. The ego vehicle can then choose an execution candidate from the set of verified trajectories (e.g., according to a certain cost function).

If Alg. 6 is modified to use the combinatorial fail-safe planning approach, the invariably safe sets can no longer be integrated as terminal constraints, since this is not supported in the combinatorial approach. Instead, we use the linear approximations presented in Sec. 4.5 as terminal constraints in the fail-safe trajectory planner.

Algorithm 6 verifyTrajectory

Input: $x_0, \mathcal{O}(t)$, planned trajectory \mathcal{I}_c , allowed lanes $\mathcal{E}_{\text{al}} \subseteq \mathcal{E}$

Output: safe part $\mathcal{I}_c^{\text{safe}}$, fail-safe trajectory \mathfrak{F}_c

- 1: $\mathfrak{F}_c \leftarrow \emptyset$
 - 2: $\mathcal{S} \leftarrow \text{invariablySafeSet}(\dots)$
 - 3: $t_{\text{TTR}} \leftarrow \mathcal{S}.\text{computeTTR}(\mathcal{I})$
 - 4: $\mathcal{I}_c^{\text{safe}} \leftarrow x([t_c, t_{\text{TTR}}])$
 - 5: $\mathcal{D} \leftarrow \text{computeDrivableArea}(x(t_{\text{TTR}}), \mathcal{O}(t), \mathcal{E}_{\text{al}})$
 - 6: $\Xi_{\text{lon}} \leftarrow \mathcal{D}.\text{selectLonCorridor}()$
 - 7: $\mathcal{C}_{\text{lon}}(t) \leftarrow \text{computeLonConstraints}(\Xi_{\text{lon}}, \mathcal{S})$
 - 8: $x_{\text{lon}}([t_{\text{TTR}}, t_{\text{TTR}} + t_{\mathfrak{F}}]) \leftarrow \text{solveLonTrajProblem}(x(t_{\text{TTR}}), \mathcal{C}_{\text{lon}}(t))$
 - 9: $\Xi_{\text{lat}} \leftarrow \mathcal{D}.\text{selectLatCorridor}(x_{\text{lon}}([t_{\text{TTR}}, t_{\text{TTR}} + t_{\mathfrak{F}}]))$
 - 10: $\mathcal{C}_{\text{lat}}(t) \leftarrow \text{computeLatConstraints}(\Xi_{\text{lat}}, \mathcal{S})$
 - 11: $x_{\text{lat}}([t_{\text{TTR}}, t_{\text{TTR}} + t_{\mathfrak{F}}]) \leftarrow \text{solveLatTrajProblem}(x(t_{\text{TTR}}), \mathcal{C}_{\text{lat}}(t))$
 - 12: **if** $\text{valid}(x_{\text{lon}}([t_{\text{TTR}}, t_{\text{TTR}} + t_{\mathfrak{F}}]), x_{\text{lat}}([t_{\text{TTR}}, t_{\text{TTR}} + t_{\mathfrak{F}}]))$ **then**
 - 13: $x_{\mathfrak{F}}([t_{\text{TTR}}, t_{\text{TTR}} + t_{\mathfrak{F}}]) \leftarrow \text{combineMotions}(x_{\text{lon}}, x_{\text{lat}})$
 - 14: $\mathfrak{F}_c = x_{\mathfrak{F}}([t_{\text{TTR}}, t_{\text{TTR}} + t_{\mathfrak{F}}])$
 - 15: **end if**
 - 16: **return** $\mathcal{I}_c^{\text{safe}}, \mathfrak{F}_c$
-

5.5 Summary

This chapter presented fail-safe motion planning as a technique to ensure legal safety during the operation of autonomous vehicles. After briefly reviewing common structures of motion planning frameworks, we introduced the structure of our safety layer and its integration into motion planning frameworks of autonomous vehicles. We summarized the steps of the verification cycle in detail and provided an algorithm for verification during the operation of the vehicle. Moreover, we proved that the presented verification technique ensures the legal safety of autonomous vehicles at all times and we introduced the computation steps of verifying a given intended trajectory in detail.

The proposed fail-safe motion planning technique is the first online verification technique for autonomous vehicles that enables fail-safe operation. As discussed in Sec. 1.1.2, existing verification techniques may reject motion plans as unsafe, leaving the autonomous vehicle without a safe plan. In contrast, our fail-safe motion planning technique ensures that the vehicle still executes provably safe trajectories even if the verification of intended trajectories fails. Our technique resolves many of the drawbacks of current verification techniques (cf. Sec. 1.1): it is real-time capable, provides fallback solutions, and remains safe even if certain traffic rules have not yet been implemented. We believe that fail-safe motion planning is the next step for verification algorithms that are perfectly suited for complex systems. In the next chapter, we demonstrate the safety benefits of the proposed verification technique in various experiments and assess its applicability to real-world traffic situations.

6 Experiments with Test Vehicles and Driving Simulators

In this chapter, we show how the proposed online safety layer (cf. Ch. 5) performs in real-world traffic situations. Using various experiments, we investigate the following research questions:

1. Do the proposed safety benefits hold in reality?
2. How does fail-safe motion planning perform in complex traffic situations?
3. How often are fail-safe trajectories executed in real traffic situations?
4. Does the execution of fail-safe trajectories compromise passenger comfort?

Sec. 6.1 introduces the technicalities of the test vehicle (cf. Fig. 6.1) used for most of the experiments in this chapter. In Sec. 6.2, we demonstrate the benefits of fail-safe motion planning in test drives with a real vehicle, conducted on a fenced test track. Afterwards, Sec. 6.3 presents the performance of fail-safe motion planning in complex real-world situations by postprocessing datasets recorded in the area of Munich, Germany. Sec. 6.4 investigates the intervention rate and comfort of the proposed safety layer. Conclusions are provided in Sec. 6.5. The content of this chapter is mainly based on the author's publications [2, 3, 5].



Figure 6.1: BMW 7-series test vehicle. The vehicle is equipped with various sensors and computers and is used to test the proposed safety framework (copyright BMW AG).

6.1 Introduction to the Vehicle Setup

For most experiments in this chapter, we use a BMW 7-series test vehicle to either execute fail-safe trajectories or record scenarios in real traffic. The vehicle is equipped with multiple camera, radar, and LiDAR sensors to create a 360° model of the environment (cf. Fig. 6.2A). The obtained sensor data is fused and mapped onto an occupancy grid [283]. Subsequently, the mapped data is processed to detect and track both static and dynamic obstacles based on the approaches in [124, 284] (cf. Fig. 6.2B–D).

The obtained environment model contains estimates of the state, type, and shape of obstacles. Uncertainties are extracted from the covariance matrix of the unscented Kalman filter used during processing. This matrix contains the variance for each measured state variable. Using the variances, we compute the standard deviations and consider them as the uncertainties of the measured state variables. In CommonRoad, these uncertainties are then modeled as intervals. The map data

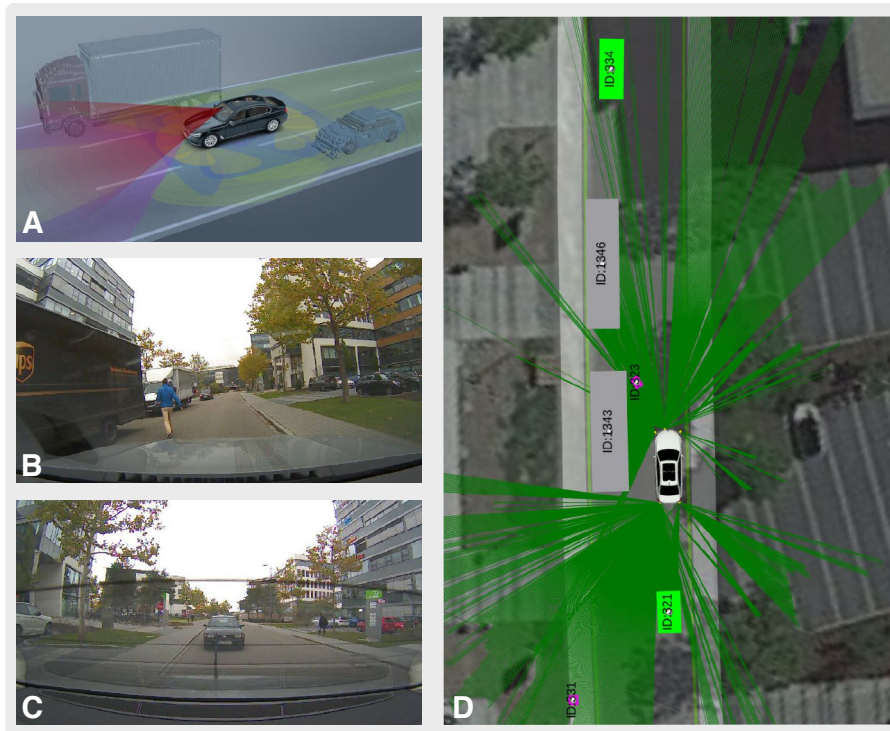


Figure 6.2: Illustration of the environment model generation. (A) We use a BMW 7-series test vehicle equipped with camera, radar, and LiDAR sensors to create a 360° environment model. (B) The front camera image during recording. (C) The rear camera image during recording. (D) The environment model of the test vehicle with detected obstacles and excerpts from the LiDAR data (green rays). Static obstacles are denoted as gray, dynamic obstacles as green, and pedestrians as magenta boxes.

is provided by BMW in the OpenDrive format [285]. The vehicle estimates its own dynamics using built-in IMU sensors, odometry ECUs, and GPS data.

The onboard computer runs the Melodic Morenia release of the robot operating system (ROS) [286] on top of Ubuntu 18.04 Bionic. In each planning cycle, we convert the provided environment model, map data, and information about the ego vehicle (e.g., dynamics) in a dedicated ROS node to the CommonRoad format for use in our safety layer. The verified safe trajectories (cf. Sec. 5.3 and Sec. 5.4) are sent to a safety electronic control unit (ECU), which checks whether provided trajectories exceed certain predefined dynamical limits (parameters specific to the test vehicle). The safety ECU passes validated trajectories to a control layer on a dSPACE AutoBox which subsequently executes control commands on the vehicle's actuators. The system can also be switched to an open-loop mode, in which planned trajectories are not executed by the vehicle. The data on each test drive is recorded using ROS Bags and the CommonRoad format, and it can be postprocessed without loss of information.

6.2 Driving Experiments

The following driving experiments were conducted on the 13th and 14th of August, 2018, at a fenced BMW test site near Maisach, Germany. For safety reasons, no other traffic participants were allowed to be present during the tests. However, we used different static obstacles, such as vehicle and pedestrian dummies, made out of foam (cf. Fig. 6.3). These foam obstacles were randomly placed in the two-lane road (with an additional shoulder lane) environment and were detected using the vehicle's onboard sensors. Dynamic vehicles and pedestrians were solely simulated and added to the environment model. The maximum allowed speed on the test site was $v_{\max} = 15$ m/s. During the two days, the weather varied from strong rain to sunshine; thus, we were also able to validate the tracking performance of fail-safe trajectories under harsh weather conditions.

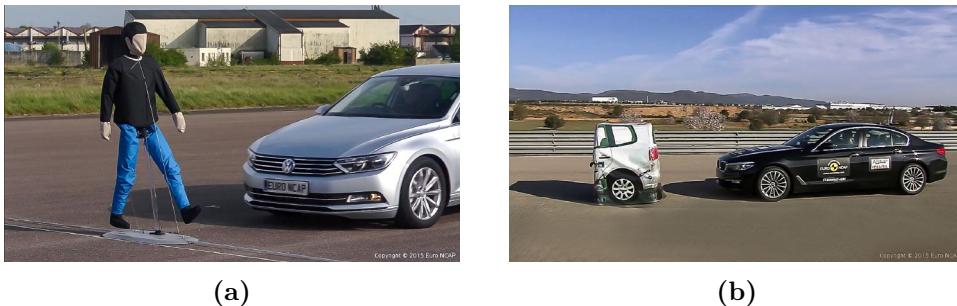


Figure 6.3: Foam obstacles for experiments. (a) The used pedestrian dummy. (b) The used vehicle target. Both dummies are NCAP conform and can be used for vehicle tests. Images taken from Euro NCAP [287, 288].

Table 6.1: General parameters of the driving experiments.

Description	Parameter with value
Velocity range	$v_{\text{ego}} \in [0 \text{ m/s}, 15 \text{ m/s}]$
Desired velocity	$v_{\text{des}} = 13.9 \text{ m/s}$
Lon. acceleration range	$a_{\text{ego,lon}} \in [-4.0 \text{ m/s}^2, 2.0 \text{ m/s}^2]$
Lat. acceleration range	$a_{\text{ego,lat}} \in [-8.0 \text{ m/s}^2, 8.0 \text{ m/s}^2]$
Jerk range	$j_{\text{ego}} \in [-10 \text{ m/s}^3, 10 \text{ m/s}^3]$
Curvature range	$\kappa_{\text{ego}} \in [-0.2/\text{m}, 0.2/\text{m}]$
Curvature change range	$\dot{\kappa}_{\text{ego}} \in [-0.2/\text{m}, 0.2/\text{m}]$
Length and width of ego vehicle	length= 5.238 m, width= 2.169 m
Circle approximation of ego vehicle	$\ell = 3.5 \text{ m}, r = 1.4 \text{ m}$
Reaction time braking	$\delta_{\text{brake}} = 0.3 \text{ s}$
Reaction time steering	$\delta_{\text{steer}} = 0.3 \text{ s}$
Time step size	$\Delta t = 0.25 \text{ s}$
Lane width	width= 3.5 m
Evasive distance	$d_{\text{eva}} = 3.5 \text{ m}$

In total, 127 experiments were performed during the two days of testing. In the following sections, we present detailed excerpts from our tests. The general parameters are valid for all scenarios and summarized in Tab. 6.1. These parameters cover the constraints on the ego vehicle’s dynamics (variables with index ego), the shape of the ego vehicle, and reaction times. Parameters specific to each scenario are listed in the appendix (cf. A.5). We used the modified version of Alg. 6 that does not compute driving corridors, but instead utilizes the combinatorial maneuver selection (cf. Sec. 3.4.1 and 5.4).

For each experiment, we provide an overview figure that shows camera images of the test drive, the planned trajectories within the CommonRoad scenario (top view), and the nominal and measured trajectories of the ego vehicle. Furthermore, we present figures that illustrate a selection of the computed invariably safe sets (a selection is made to highlight parts of the verification process). Videos of the different experiments and simulations in this section are provided in the supplementary materials of this thesis (cf. A.9).

6.2.1 Verifying randomly generated trajectories

In our first two experiments, we show that by design, the proposed online verification framework ensures safe behaviors of the ego vehicle for any given intended trajectory. This property is especially important when the intended motion planner of the ego vehicle needs to be changed or machine learning is employed. We created a malicious intended trajectory planner to demonstrate this property under extreme conditions. The planner is based on the techniques presented in Sec. 3.2, but it adds random offsets to the longitudinal position constraints, tries to reach

random desired velocities, and performs oscillating lateral motions with random frequency and amplitude (more details are given in A.2). Nevertheless, every obtained trajectory fulfills the kinematic constraints of the ego vehicle. In the driving experiments, we place the foam vehicle dummy randomly in the ego vehicle’s path. The parameters of the presented scenarios are given in A.5.1.

Braking maneuver Fig. 6.4 shows an intended trajectory and a fail-safe trajectory, which avoids a collision through braking. The parameters of this scenario are given in Tab. A.8. The intended trajectory lets the ego vehicle accelerate to a velocity of about 10 m/s without reacting to the static obstacle. The proposed verification technique automatically verifies the malicious intended trajectory by computing the verified part and a subsequent fail-safe trajectory. This fail-safe trajectory with a horizon of $t_{\mathfrak{S}} = 5$ s starts at the time-to-react of $t_{\text{TTR}} = 7$ s. By automatically executing the computed fail-safe trajectory, the ego vehicle avoids a collision and comes to a standstill directly in front of the static obstacle.

To more closely examine the verification, Fig. 6.6 illustrates the invariably safe set of the scenario as a projection onto the s - v plane. Since this scenario is static, the resulting invariably safe sets are also time-invariant. The fail-safe trajectory starts at the last state of the intended trajectory that is still enclosed in \mathcal{S}^1 (light gray set in Fig. 6.6). The set \mathcal{S}^1 is computed with a reaction time of $\delta_{\text{brake}} = 0.3$ s to indicate when fail-safe trajectories need to start. As a result, executed fail-safe trajectories (states after t_{TTR}) may not be fully enclosed in \mathcal{S}^1 , since the reaction time to start braking no longer needs to be considered. The set $\mathcal{S}^{1,\text{rel}}$ in Fig. 6.6 corresponds to the relaxed invariably safe sets with reaction time $\delta_{\text{brake}} = 0$ s (dark gray set in Fig. 6.6). For the relaxed set $\mathcal{S}^{1,\text{rel}}$, the executed fail-safe trajectory is fully enclosed and thus, invariably safe.

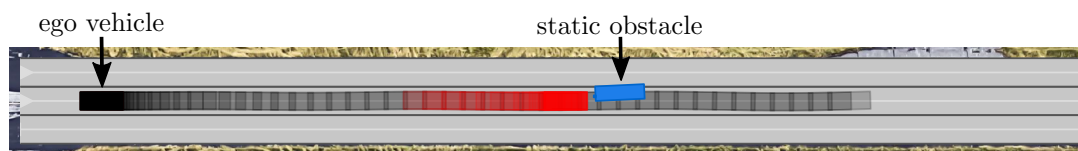
Evasive maneuver In this experiment, we demonstrate evasive maneuvers. Fig. 6.5 shows the result of verifying a malicious intended trajectory with a fail-safe trajectory, which avoids a collision with the static obstacle by swerving to the left adjacent lane. In contrast to the previous scenario, the ego vehicle has a higher velocity, and an evasive maneuver can be performed at a later point in time than a braking maneuver; this is automatically detected by our safety framework. The parameters of this scenario are given in Tab. A.9. The intended trajectory accelerates the ego vehicle in two phases to randomly chosen desired velocities of 4 m/s and 15 m/s (cf. acceleration and velocity plots in Fig. 6.5C). The computed fail-safe trajectory starts at the time-to-react of $t_{\text{TTR}} = 11$ s and lets the ego vehicle swerve into the left adjacent lane. The maximum lateral acceleration of the evasive maneuver corresponds to 4.4 m/s^2 . After arriving in the desired left lane, the ego vehicle performs a comfortable braking maneuver until it reaches a standstill.

The computed invariably safe set is shown in Fig. 6.7 in two different projections. Fig. 6.7a demonstrates that the fail-safe trajectory starts at the last state of the intended trajectory that is still enclosed in the invariably safe set \mathcal{S}^2 , depicted as a

A Camera images



B Scenario



C Measurement data

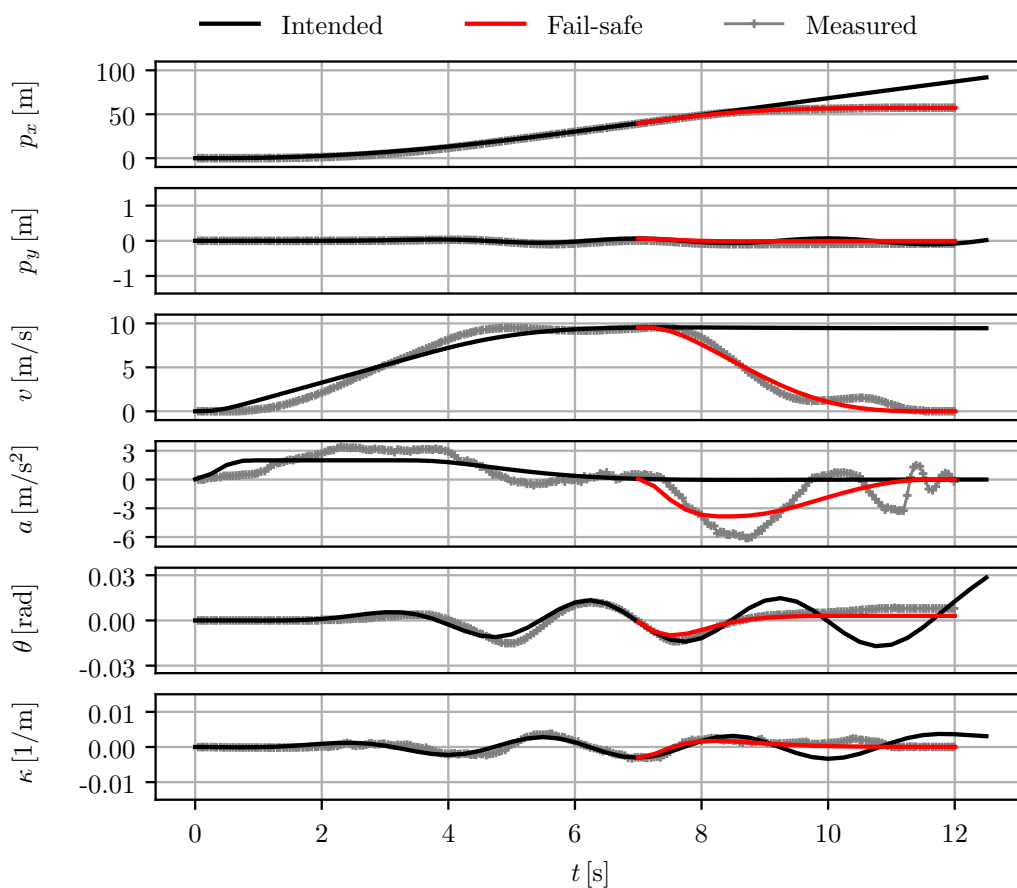
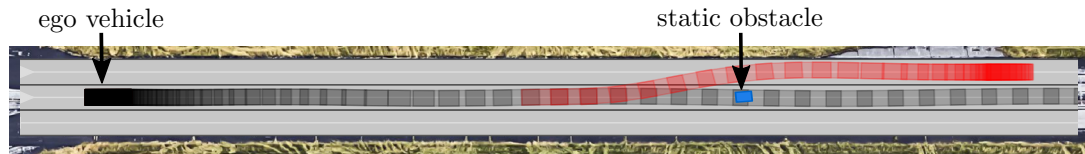


Figure 6.4: Braking maneuver to avoid collisions with a static obstacle (ZAM_Urban-2_1). (A) Camera images of the experiment. (B) The planned trajectories and the occupancy set of the static obstacle. (C) The measured data of the experiment. ©2020 IEEE.

A Camera images



B Scenario



C Measurement data

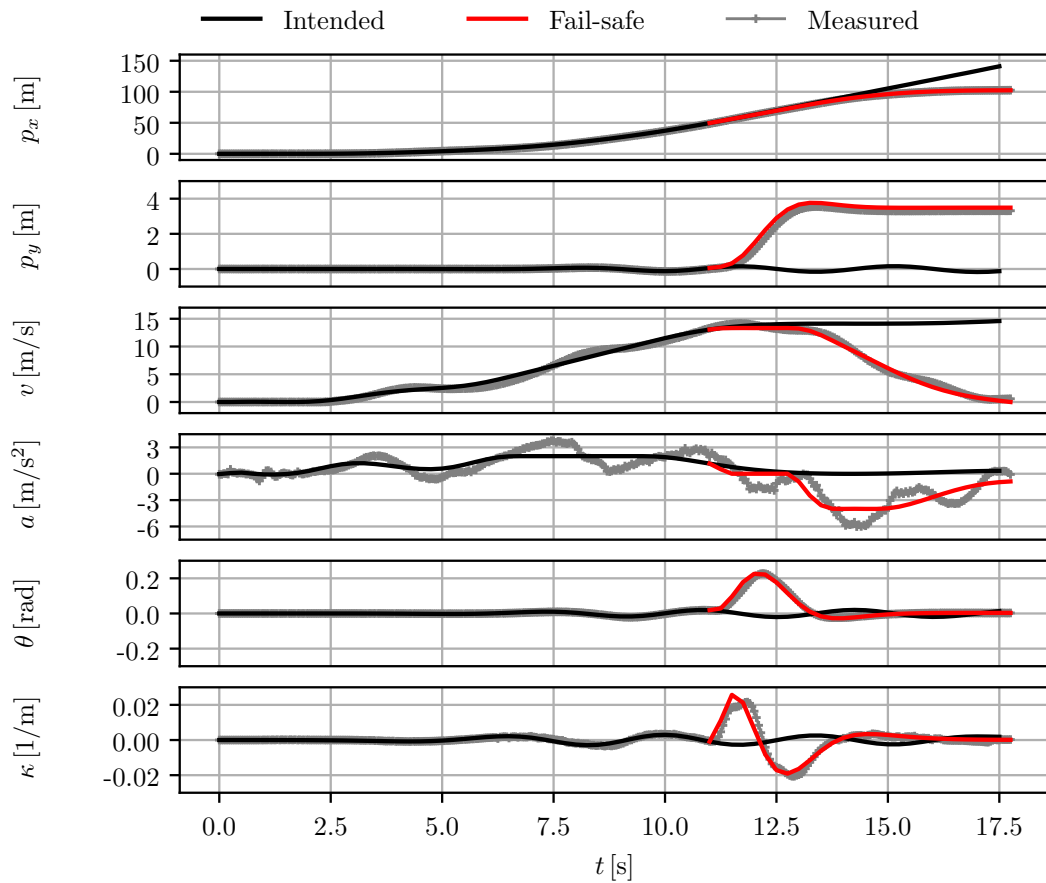


Figure 6.5: Evasive maneuver to avoid collisions with a static obstacle (ZAM_Urban-3_1). (A) Camera images of the experiment. (B) The planned trajectories and the occupancy set of the static obstacle. (C) The measured data of the experiment.

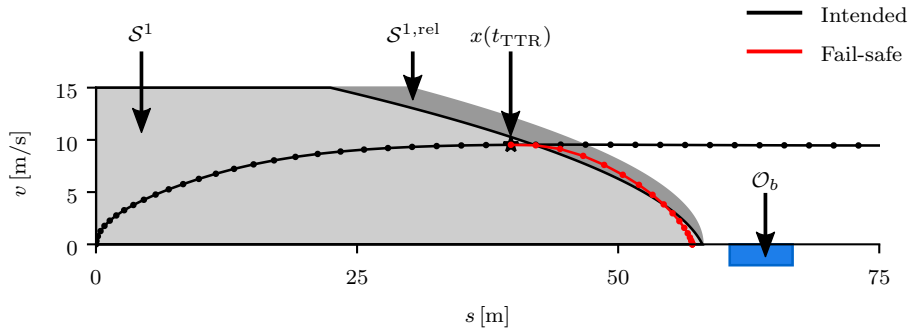
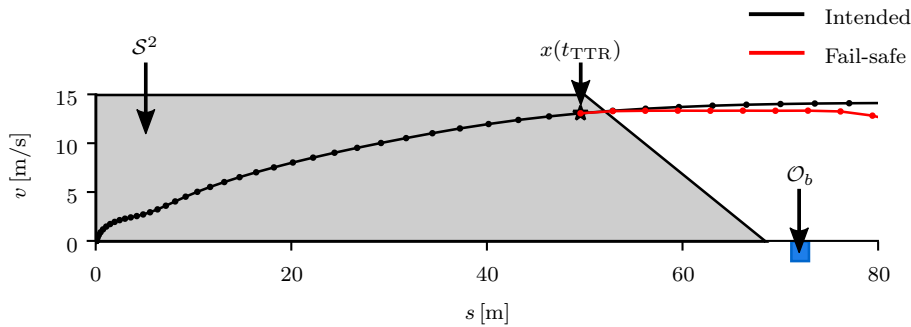
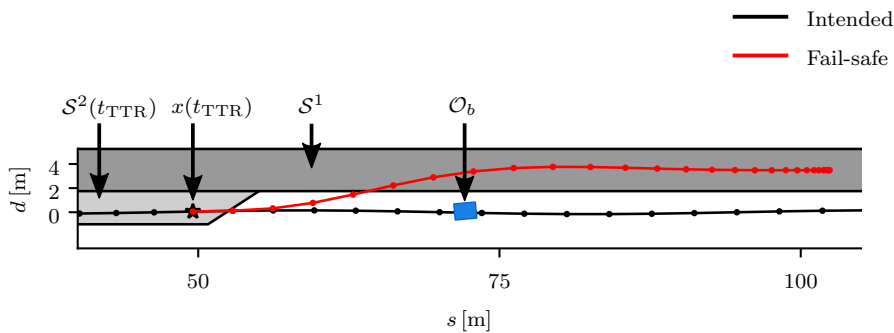


Figure 6.6: Invariably safe set of the scenario in Fig. 6.4. The fail-safe trajectory starts at the state $x(t_{\text{TTR}})$, which is the last state along the intended trajectory that is still enclosed in the invariably safe set \mathcal{S}^1 . The relaxed set $\mathcal{S}^{1,\text{rel}}$ is computed with $\delta_{\text{brake}} = 0$ s and fully encloses the fail-safe trajectory. The sets are shown as projections onto the s - v plane. ©2020 IEEE.



(a) Projection of \mathcal{S}^2 onto the s - v plane.



(b) Projection of \mathcal{S}^1 and \mathcal{S}^2 onto the s - d plane.

Figure 6.7: Invariably safe set of the scenario in Fig. 6.5. (a) The computed safe set \mathcal{S}^2 is shown as a projection onto the s - v plane. (b) The fail-safe trajectory starts in \mathcal{S}^2 and ends in \mathcal{S}^1 , shown as projections onto the s - d plane.

projection onto the s - v plane. Fig. 6.7b illustrates the projection of \mathcal{S}^2 onto the s - d plane by considering the velocity slice at $v(t_{\text{TTR}}) = 13.06$ m/s. Moreover, this figure demonstrates the computed set \mathcal{S}^1 for the left adjacent lane. This set covers the whole lane, since no obstacles occupy this lane. Again, we see that the computed fail-safe trajectory is an invariably safe trajectory: it starts in \mathcal{S}_2 (light gray set) and ends in \mathcal{S}_1 for the left adjacent lane (dark gray set).

6.2.2 Verifying planned motions in dynamic environments

The following two experiments demonstrate how our framework ensures safety in dynamic environments. Here, we examine a situation in which a vehicle in an adjacent lane cuts into the ego vehicle’s lane and then performs emergency braking. The dynamic vehicle is randomly placed in the adjacent left lane in the environment model with an initial velocity of $v = 13.89$ m/s. We vary the set of allowed lanes for the fail-safe trajectory computation in this scenario to demonstrate how our framework automatically computes different fail-safe maneuvers. The parameters of the presented scenarios are given in A.5.2.

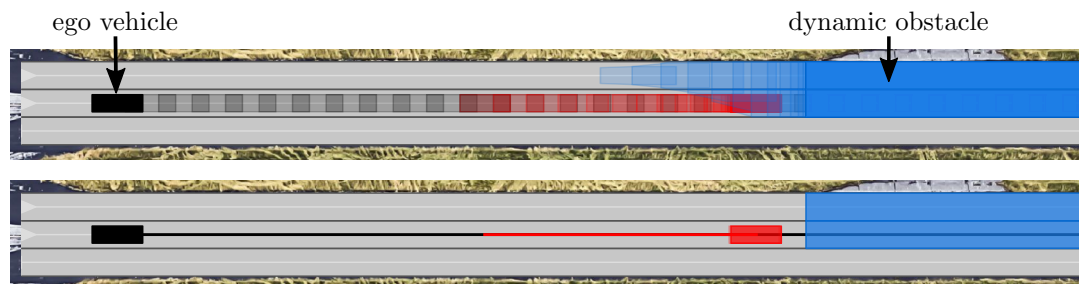
Braking maneuver Fig. 6.8 shows the results when the ego vehicle is only allowed to drive in the two leftmost lanes. The parameters of this scenario are given in Tab. A.10. The initial distance between the ego vehicle and the other vehicle is approximately 54 m. The ego vehicle is travelling at a constant velocity of $v = 13.78$ m/s. The computed fail-safe trajectory starts at the time-to-react of $t_{\text{TTR}} = 2.75$ s, and as a result, the ego vehicle performs an emergency brake maneuver to avoid a collision with the cut-in vehicle. Our framework automatically determines the most comfortable fail-safe maneuver for the ego vehicle: since the adjacent left lane is blocked by the occupancy set of the dynamic vehicle, evading does not provide additional benefits. Fig. 6.8B shows the top view of the scenario for the whole time horizon and for the final time step. The computed invariably safe set $\mathcal{S}^1(t_{\text{TTR}})$ for the time step t_{TTR} is visualized in Fig. 6.10 as a projection onto the s - v plane.

Evasive maneuver In our second experiment with dynamic obstacles, we also allow the ego vehicle to use the adjacent shoulder lane for fail-safe maneuvers (rightmost lane in the scenario). The parameters of this scenario are given in Tab. A.11. Fig. 6.9 illustrates the results of the experiment. This time, the initial distance between the ego vehicle and the other vehicle is shorter, at approximately 45 m. The ego vehicle is traveling at a constant velocity of $v = 12.56$ m/s. The computed fail-safe trajectory starts at the time-to-react of $t_{\text{TTR}} = 3.25$ s. In contrast to the previous experiment, our framework computes a fail-safe trajectory that involves the ego vehicle swerving into the adjacent shoulder lane to avoid colliding with the cut-in vehicle. The maximum lateral acceleration during this evasion

A Camera images



B Scenario



C Measurement data

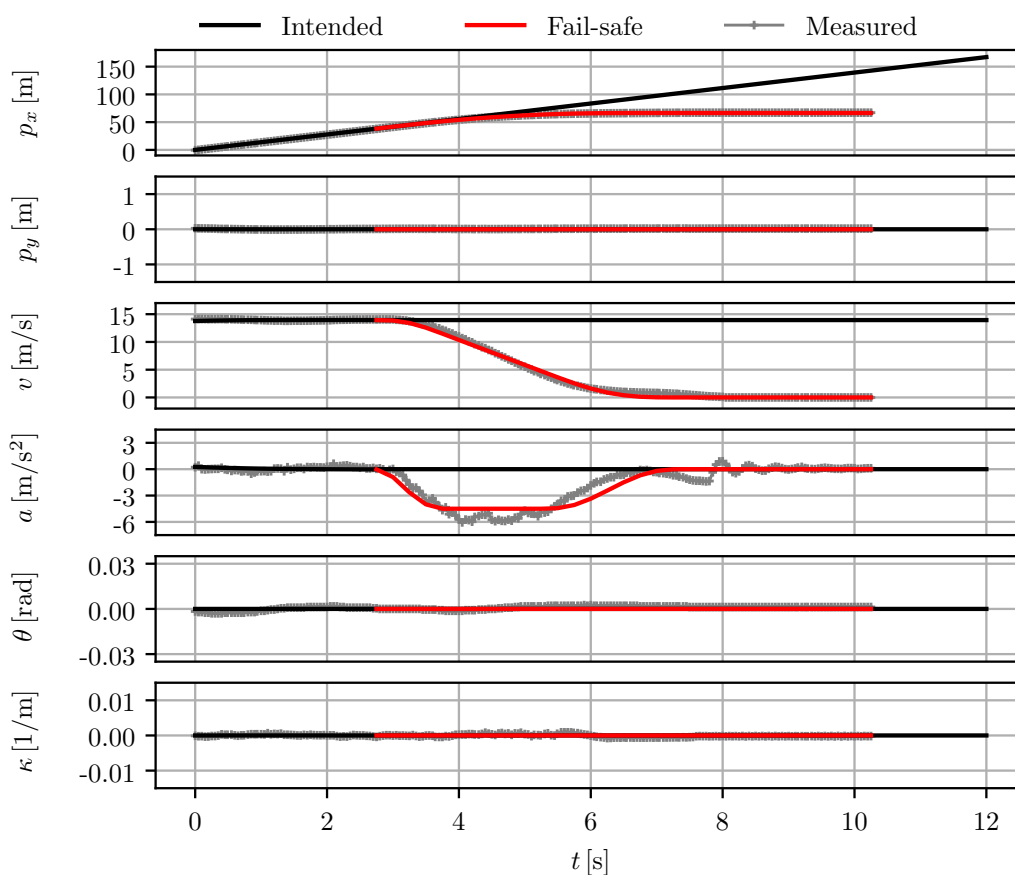
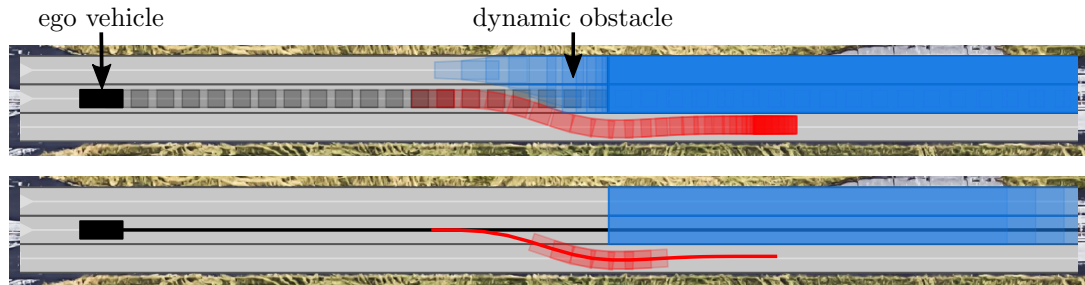


Figure 6.8: Avoiding collisions with a cut-in vehicle by emergency braking (ZAM_Urban-6.1.S-1). (A) Camera images of the experiment. (B) The planned trajectories and the predicted occupancy set of the dynamic obstacle over the whole time horizon and for the final time step. (C) The measured data of the experiment.

A Camera images



B Scenario



C Measurement data

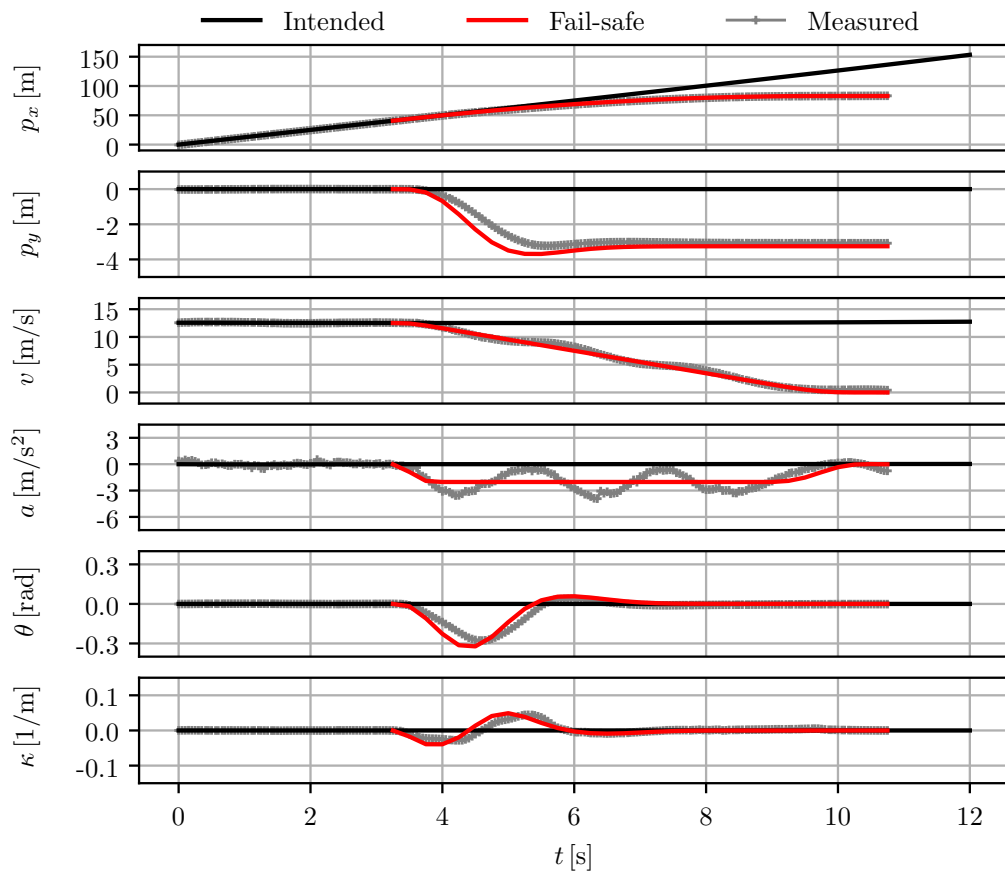


Figure 6.9: Avoiding collisions with a cut-in vehicle by swerving to the adjacent shoulder (ZAM_Urban-7.1.S-1). (A) Camera images of the experiment. (B) The planned trajectories and the predicted occupancy set of the dynamic obstacle over the whole time horizon and a selected interval. (C) The measured data of the experiment. ©2020 IEEE.

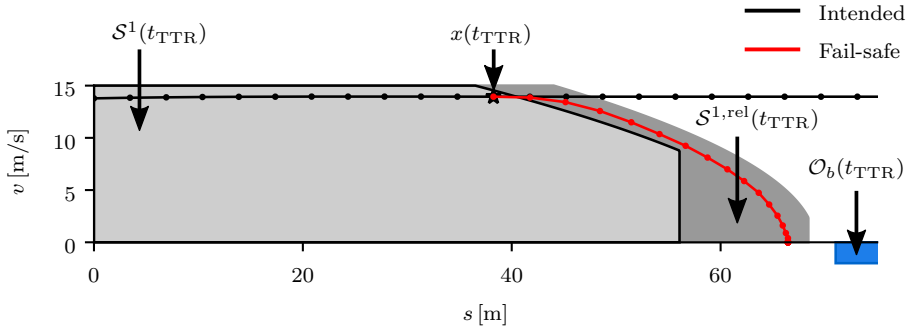
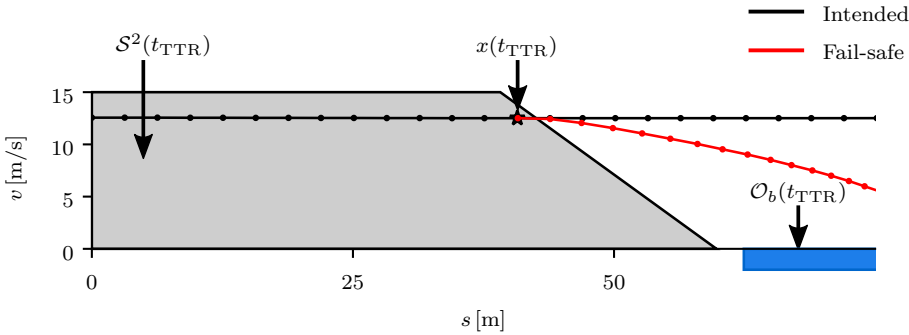
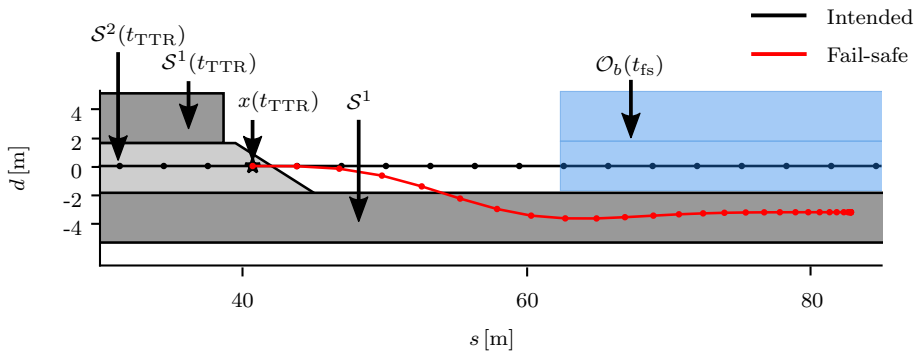


Figure 6.10: Invariably safe set of the scenario in Fig. 6.8. The fail-safe trajectory starts at the state $x(t_{\text{TTR}})$, which is the last state along the intended trajectory that is still enclosed in the invariably safe set \mathcal{S}^1 . The relaxed set $\mathcal{S}^{1,\text{rel}}$ is computed with $\delta_{\text{brake}} = 0\text{ s}$ and fully encloses the fail-safe trajectory. The sets are shown as projections onto the s - v plane.



(a) Projection of \mathcal{S}^2 onto the s - v plane.



(b) Projection of \mathcal{S}^1 and \mathcal{S}^2 onto the s - d plane.

Figure 6.11: Invariably safe set of the scenario in Fig. 6.9. (a) The computed safe set \mathcal{S}^2 is shown as a projection onto the s - v plane. (b) The fail-safe trajectory starts in \mathcal{S}^2 and ends in \mathcal{S}^1 , shown as projections onto the s - d plane. ©2020 IEEE.

maneuver is measured at 4.1 m/s^2 . Fig. 6.9B shows the top view of the scenario for the whole time horizon and selected time steps $t \in [4.5, 5.75]$.

The computed invariably safe sets are shown in Fig. 6.11 in two different projections. Fig. 6.11a visualizes \mathcal{S}^2 for the time step t_{TTR} in the s - v plane together with the intended and fail-safe trajectory, while Fig. 6.11b presents the s - d plane projections of the invariably safe sets \mathcal{S}^1 and \mathcal{S}^2 for the time step t_{TTR} and velocity slice $v(t_{\text{TTR}}) = 12.51 \text{ m/s}$. The computed fail-safe trajectory starts in \mathcal{S}^2 and ends in \mathcal{S}^1 of the shoulder lane.

In this scenario, an evasive maneuver can be executed later than a braking maneuver. We illustrate this fact using the computed invariably safe sets. As a reference, Fig. 6.11b also shows \mathcal{S}^1 for the adjacent left lane at t_{TTR} . Here, the set \mathcal{S}^1 has the same size as for the ego vehicle’s lane, because the minimum longitudinal positions of vehicle b in the occupancy set $\mathcal{O}_b(t_{\text{TTR}})$ are equal. Since \mathcal{S}^2 is larger than \mathcal{S}^1 in this lane and it encloses a state at a later point in time, evading can be performed one step later than braking.

6.2.3 Avoiding collisions with vulnerable road users

The last two driving experiments highlight how our safety framework handles pedestrians who suddenly enter the ego vehicle’s path. We place the foam pedestrian close to the right border of the ego vehicle’s lane. The onboard sensors of the ego vehicle detect the pedestrian and the set-based prediction computes the set of future behaviors based on simulated initial dynamics of the pedestrian that we choose to create critical situations. The parameters of the presented scenarios are given in A.5.3.

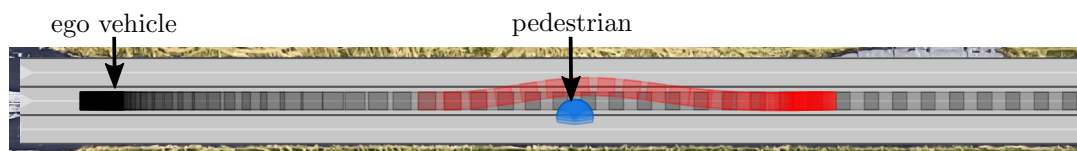
Pedestrian stops when entering lane Our first scenario, illustrated in Fig. 6.12, considers that the pedestrian slows down as soon as he has entered the path. This behavior corresponds to situations in which a pedestrian inattentively enters a lane but immediately reacts to the approaching vehicle by stopping. The parameters of this scenario are given in Tab. A.12. The intended trajectory lets the ego vehicle accelerate to the desired velocity of 13.9 m/s . For the prediction, the pedestrian enters the ego vehicle’s path at a velocity of $v_{\text{ped}} = 1.5 \text{ m/s}$. Since we simulate the dynamics of the pedestrian, we set the time when the pedestrian enters the lane in our scenario to the time-to-react of $t_{\text{TTR}} = 6 \text{ s}$. With this choice, we enforce an evasive instead of a braking maneuver in our safety framework; otherwise, the pedestrian is already blocking the road when the ego vehicle approaches. The computed fail-safe trajectory allows the ego vehicle to evade the pedestrian at a velocity of $v(t_{\text{TTR}}) = 12.24 \text{ m/s}$. After passing the pedestrian and successfully returning to the initial lane, the ego vehicle performs an emergency braking maneuver to come to a standstill.

Fig. 6.13 illustrates the computed invariably safe sets in two different projections. The invariably safe sets \mathcal{S}^1 and \mathcal{S}^2 are visualized in Fig. 6.13a as a projection onto the s - v plane. The fail-safe trajectory starts in \mathcal{S}^2 and lets the ego vehicle swerve

A Camera images



B Scenario



C Measurement data

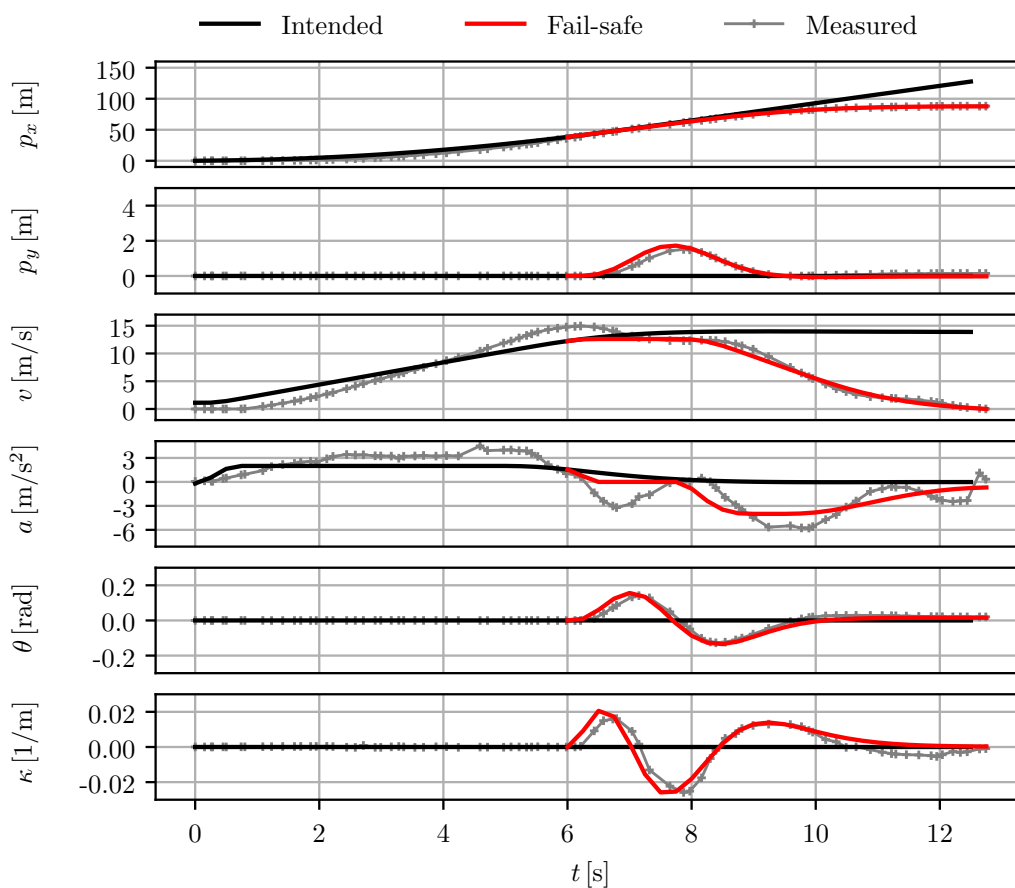


Figure 6.12: Evading stopped pedestrians (ZAM_Urban-4.1_S-1). (A) Camera images of the experiment. (B) The planned trajectories and the predicted occupancy set of the pedestrian over the whole time horizon. (C) The measured data of the experiment.

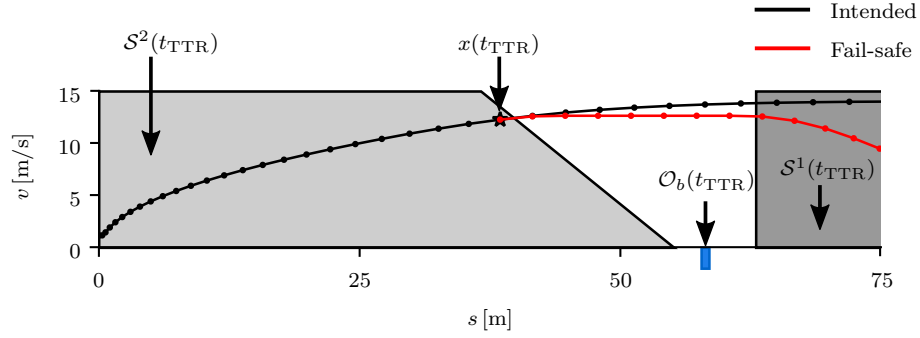
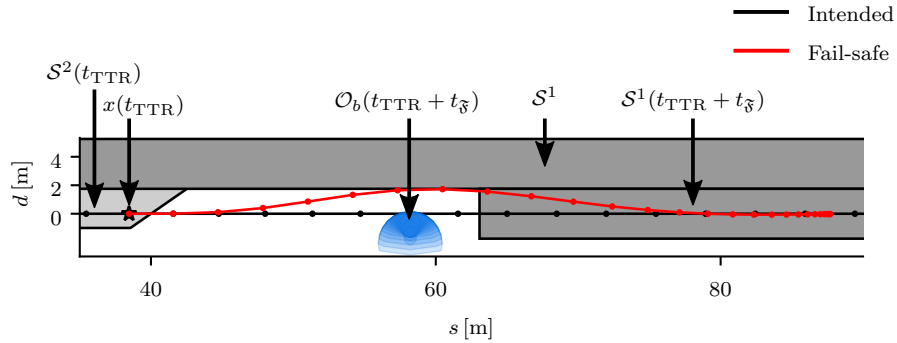
(a) Projection of \mathcal{S}^2 onto the s - v plane.(b) Projection of \mathcal{S}^1 and \mathcal{S}^2 onto the s - d plane.

Figure 6.13: Invariably safe set of the scenario in Fig. 6.12. (a) The computed safe set \mathcal{S}^2 is shown as a projection onto the s - v plane. (b) The fail-safe trajectory starts in \mathcal{S}^2 and ends in \mathcal{S}^1 , shown as projections onto the s - d plane.

around the pedestrian. As soon as the fail-safe trajectory enters \mathcal{S}^1 of the initial lane, the ego vehicle initiates a braking maneuver to safely stop. Both sets, \mathcal{S}^1 and \mathcal{S}^2 , are visualized in Fig. 6.13b as a projection onto the s - d plane.

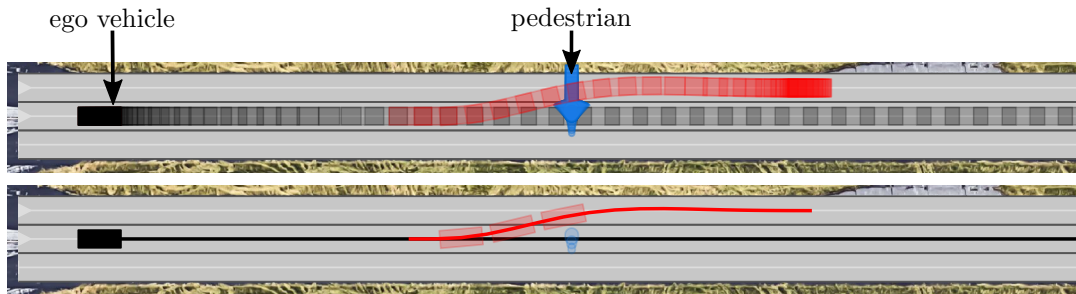
Pedestrian crosses the lane In our second scenario (cf. Fig. 6.14), we predict that the pedestrian does not react to the oncoming ego vehicle and thus continues crossing the lane. The set-based prediction considers this crossing behavior when computing the occupancy prediction. The parameters of this scenario are given in Tab. A.13. Similar to the previous scenario, the intended trajectory accelerates the ego vehicle to the desired velocity of 13.9 m/s.

The pedestrian enters the ego vehicle's path at a velocity of $v_{\text{ped}} = 1.5$ m/s. In our simulation, we set the time when the pedestrian enters the lane to the time-to-react of $t_{\text{TTR}} = 6.5$ s. This choice enables us to enforce an evasive instead of a braking maneuver for demonstration; otherwise, the pedestrian is already blocking the path when ego vehicle approaches. The computed fail-safe trajectory allows the ego vehicle to swerve into the adjacent left lane at a velocity of $v(t_{\text{TTR}}) = 12.22$ m/s.

A Camera images



B Scenario



C Measurement data

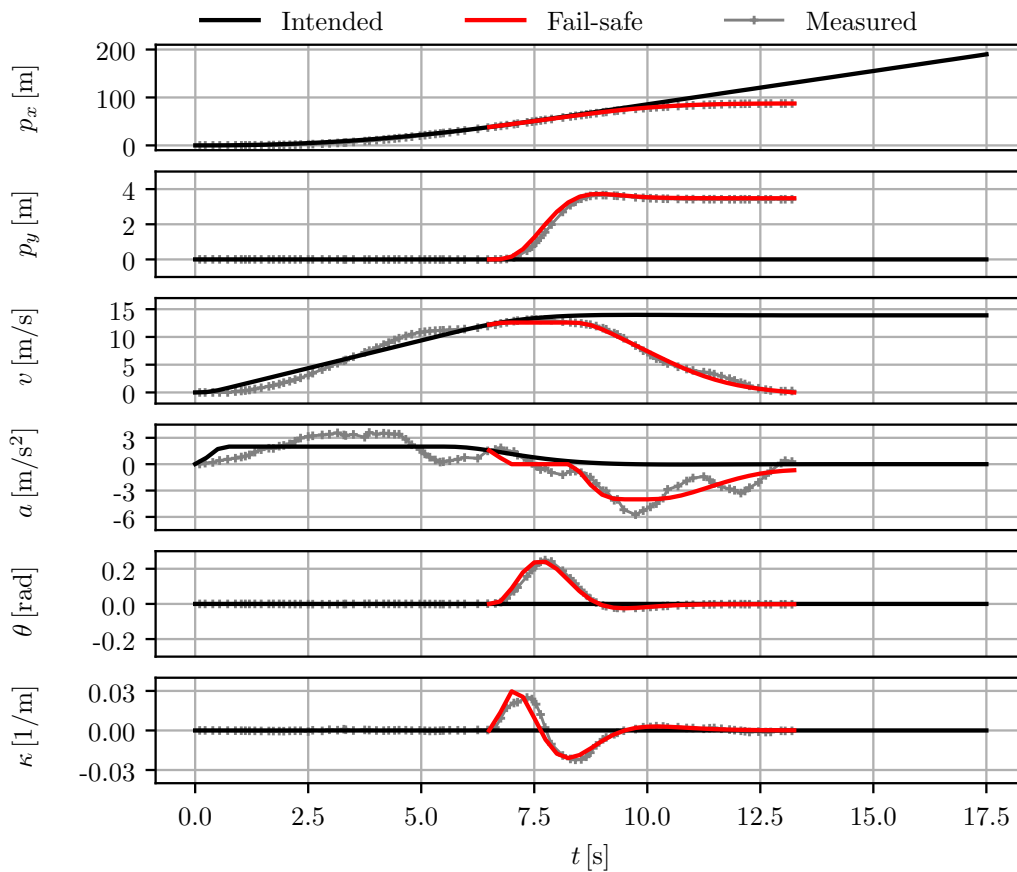


Figure 6.14: Evading pedestrians by swerving into an adjacent lane (ZAM_Urban-5.1_S-1). (A) Camera images of the experiment. (B) The planned trajectories and the predicted occupancy set of the pedestrian over the whole time horizon and a selected time steps $t \in \{0.5 \text{ s}, 1.0 \text{ s}, 1.5 \text{ s}\}$. (C) The measured data of the experiment. ©2020 IEEE.

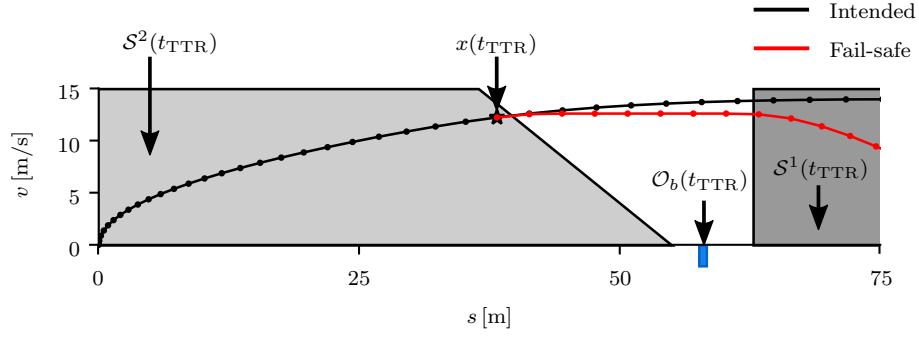
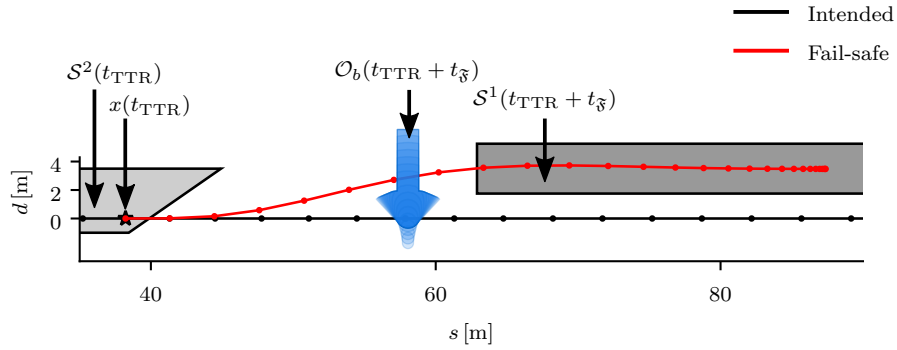
(a) Projection of \mathcal{S}^2 onto the s - v plane.(b) Projection of \mathcal{S}^1 and \mathcal{S}^2 onto the s - d plane.

Figure 6.15: Invariably safe set of the scenario in Fig. 6.14. (a) The safe set \mathcal{S}^2 is shown as a projection onto the s - v plane. (b) The fail-safe trajectory starts in \mathcal{S}^2 and ends in \mathcal{S}^1 , shown as projections onto the s - d plane. ©2020 IEEE.

After fully entering the adjacent lane and passing the pedestrian, the ego vehicle performs a braking maneuver to come to a standstill. During this experiment, we measure a maximum lateral acceleration of 4.8 m/s^2 , which is the highest among all of our experiments.

Fig. 6.15 illustrates the computed invariably safe sets in two different projections. The invariably safe sets \mathcal{S}^1 and \mathcal{S}^2 are visualized in Fig. 6.15a as a projection onto the s - v plane. Similar to the previous pedestrian scenario, the fail-safe trajectory starts in \mathcal{S}^2 and lets the ego vehicle swerve into the left adjacent lane. As soon as the fail-safe trajectory enters \mathcal{S}^1 of the adjacent lane, the ego vehicle initiates a braking maneuver to safely stop. Both sets, \mathcal{S}^1 and \mathcal{S}^2 , are visualized in Fig. 6.15b as a projection onto the s - d plane.

6.2.4 Summary of driving experiments

In 127 driving experiments, we validated the safety benefits of fail-safe motion planning in various situations with static and dynamic obstacles as well as vulnerable

road users. Furthermore, we demonstrated that fail-safe motion planning ensures safety for arbitrary intended trajectories by using an intended trajectory planner that plans random trajectories. Our fail-safe planner generates drivable fail-safe trajectories that can be tracked by a controller even when the maneuver is highly dynamic. Moreover, we used invariably safe sets to determine the time-to-react and to ensure the safety of the ego vehicle for an infinite time horizon, for instance by stopping the ego vehicle in safe areas.

6.3 Fail-Safe Trajectories in Complex Urban Traffic Scenarios

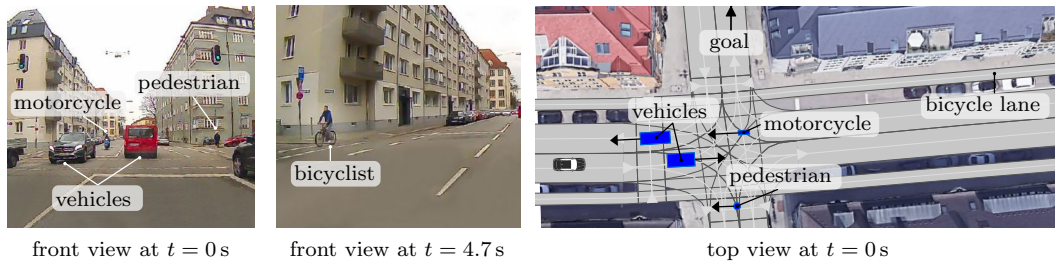
In the previous section, we validated that the execution of fail-safe trajectories ensures the safety of the ego vehicle in real scenarios. For safety reasons, we are unable to test our fail-safe motion planning technique with multiple other traffic participants. However, to demonstrate that the proposed fail-safe motion planning also ensures safety in complex traffic situations, we create three critical scenarios by recording real traffic in the city of Munich. These scenarios correspond to situations in urban environments in which most accidents occur: at intersections, with pedestrians, and when changing lanes [289]. We postprocess the recordings, as described in A.6, and apply the proposed safety layer to ensure safety in the presented scenarios. In these experiments, we use the fail-safe trajectory generation by utilizing driving corridors, as described in Sec. 3.4.3. The parameters of the scenarios are summarized in A.6.2 and A.6.3.

For each scenario, we illustrate the results in a figure containing an overview of the traffic situation and the verification results (cf. Fig. 6.16–6.18). To understand the current traffic situation, we present camera images of the test vehicle and a top view of the CommonRoad scenario. Below the images, the verification results are shown for selected planning cycles to highlight interesting situations. On the left side, the intended trajectories are depicted with the initial states of other traffic participants without uncertainties, and on the right side, the fail-safe trajectories are illustrated with the occupancy sets at the final time of the fail-safe trajectories. The computed lateral driving corridor is shown for all time steps. We plot trajectories with respect to the center of the rear axle as the reference point. All used parameters are summarized in A.6. Additional figures are provided in A.6.4 to A.6.6, and videos are available in A.9.

6.3.1 Left turn at an urban intersection

Left turns are regarded as the most critical maneuvers at urban intersections [289]. In our scenario, the ego vehicle needs to consider the right of way of the oncoming vehicles and yield to potential bicyclists in their dedicated lane (cf. Fig. 6.16 A, top view). However, the motions of oncoming vehicles or passing bicyclists may change rapidly over time. For instance, vehicles may accelerate to approach the

A Scenario overview from recordings



B Planning results

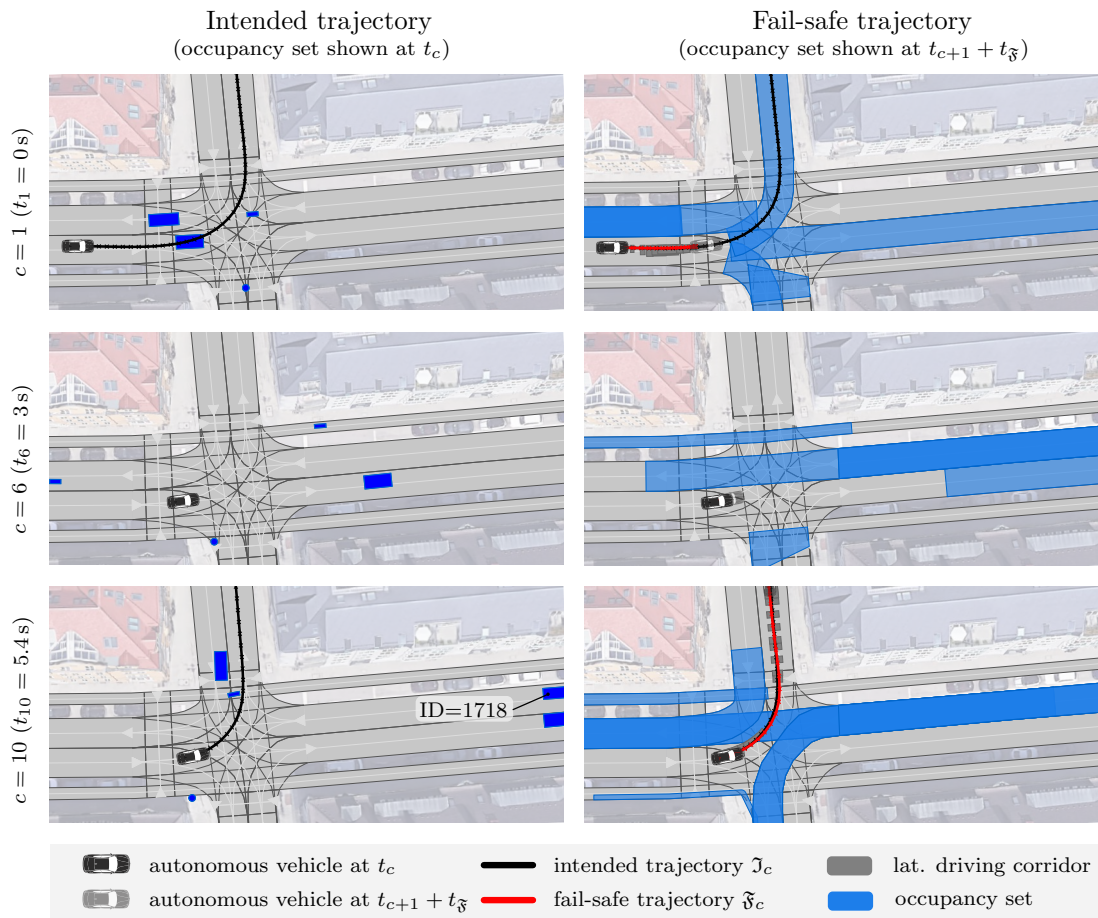


Figure 6.16: Left turn at urban intersection (DEU_Muc-5_1_T-1). (A) Scenario overview from recordings with images of the vehicle’s front camera and a top view of the traffic situation. (B) The planning and verification results for three selected planning cycles $c \in \{1, 6, 10\}$. Satellite Images ©Google, GeoBasis-DE/BKG.

intersection faster and bicyclists may even stop and dismount, which increases the uncertainty about the future evolution of the traffic scene. In any circumstance, the ego vehicle must yield to oncoming traffic while not disrupting the traffic flow with overly conservative behavior.

Fail-safe motion planning addresses this challenge by safeguarding the opportunistic intended motion plan with fail-safe trajectories. The computed fail-safe trajectories ensure compliance with the right of way traffic rule, and they guarantee that the ego vehicle will never come to a standstill within the intersection area. The latter is achieved through invariably safe sets that restrict the fail-safe trajectory to stop the vehicle either before or after the intersection. The former is accomplished through the over-approximative set-based prediction. Since SPOT predicts all admissible legal actions of other traffic participants, the safety layer can determine whether a left turn maneuver can be completed before the oncoming traffic is able to enter the intersection. Thus, if a fail-safe trajectory that crosses the intersection area is found, the ego vehicle automatically respects the right of way.

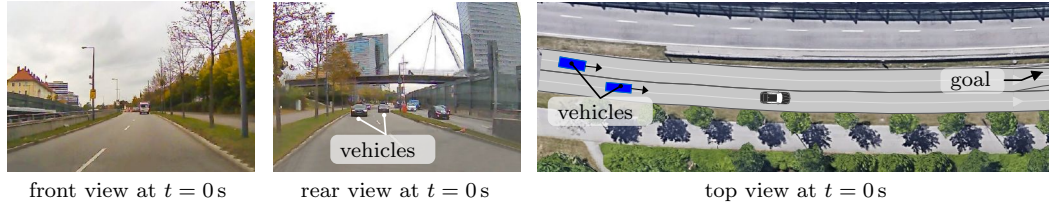
As shown in Fig. 6.16, the ego vehicle first approaches the intersection along its intended motion plan (cf. Fig. 6.16 B, intended trajectory at $t = 0$ s). From $t = 2.4$ s until $t = 5.4$ s, our safety layer automatically detects that the intended trajectory leads to an unsafe situation, where we cannot exclude a collision with the oncoming vehicle within the intersection area before the bicyclist has definitely passed. The computed fail-safe trajectory stops the vehicle at the intersection (cf. Fig. 6.16 B, fail-safe trajectory at $t = 3$ s). Immediately after the bicyclist has passed, the intended trajectory is verified as safe and the ego vehicle continues its left turn before the oncoming traffic (cf. Fig. 6.16 B, intended trajectory at $t = 5.4$ s). As an example of how the legal safety specification excludes certain behaviors of other traffic participants, we consider the prediction of the oncoming vehicle ID = 1718: since the legal safe distance forbids vehicles to traverse the intersection behind the ego vehicle in a way that would violate the safety distance to the ego vehicle, vehicle ID = 1718 is allowed to continue straight or to turn left, but it may not turn right.

The utilized parameters for this scenario can be found in A.6.2 and A.6.3. Additional figures that highlight the results of planning cycle $c = 10$ are presented in A.6.4.

6.3.2 Lane changes in dense urban traffic

The density of urban traffic is expected to increase with the rise of autonomous vehicles [290]. This circumstance will require these vehicles to be able to maneuver in tight spaces. While autonomous vehicles can simply brake when following the current lane (e.g., if a preceding vehicle performs emergency braking), lane changes are more challenging. Furthermore, if autonomous vehicles drive too conservatively, they will probably impede other traffic and only merge into large gaps. Existing approaches (e.g., [291, 292]) make lane changes in dense traffic by starting to slowly merge in between other vehicles. If a gap opens, the autonomous vehicle completes

A Scenario overview from recordings



B Planning results

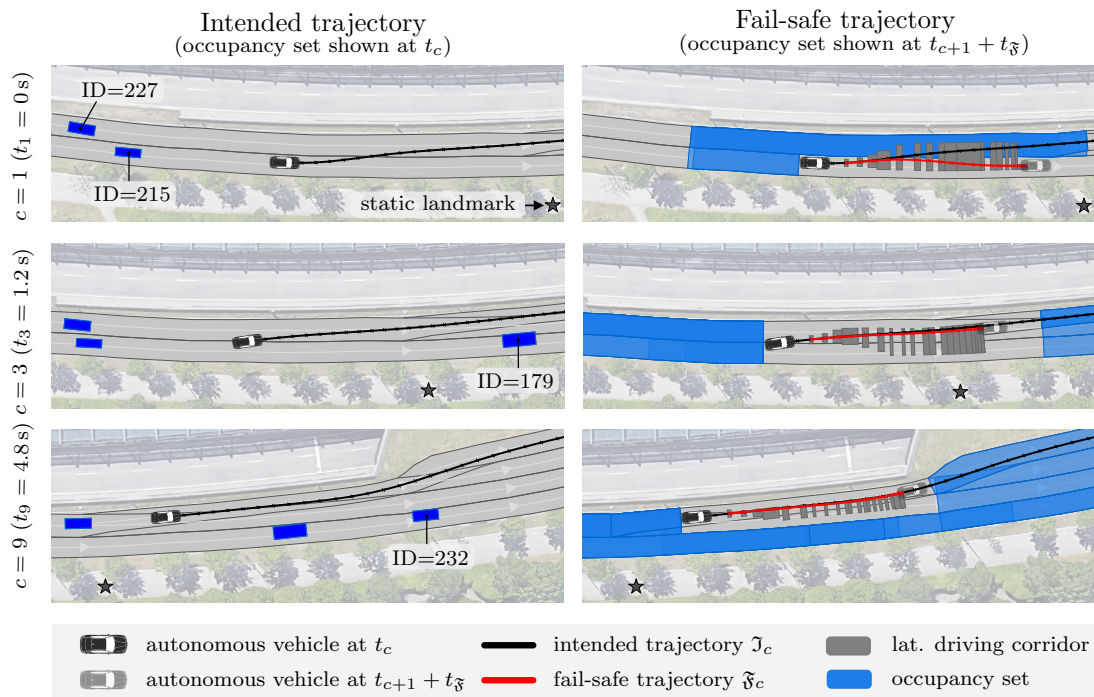


Figure 6.17: Lane change in dense urban traffic (DEU_Muc-6.1-T-1). (A) Scenario overview from recordings with images of the vehicle’s front camera and a top view of the traffic situation. (B) The planning and verification results for three selected planning cycles $c \in \{1, 3, 9\}$. The static landmark is placed in the same location in each figure. Satellite Images ©Google, GeoBasis-DE/BKG.

the lane change; otherwise, it terminates the merging maneuver. Our proposed safety layer ensures safety at all times for arbitrary lane changes by maintaining fail-safe trajectories. Since fail-safe trajectories do not necessarily have to be executed, the ego vehicle can perform lane changes without being overly conservative.

At the beginning of our scenario (cf. Fig. 6.17 A), the ego vehicle intends to change lanes and merge in front of a vehicle approaching from behind (cf. Fig. 6.17 B, $t = 0$ s). However, since the latter vehicle must not maintain a safe distance to the autonomous vehicle, it may accelerate until its velocity reaches the speed limit. If it does, the intended lane change of the ego vehicle would cause a collision. Thus, the fail-safe trajectory swerves back into the initial, right lane. This trajectory is safe, since the vehicle that is currently driving in the same lane behind the ego vehicle must maintain a safe distance (cf. the occupancy set that ends just behind the ego vehicle in Fig. 6.17 B, fail-safe trajectory column).

In the next planning cycle ($t = 0.6$ s), the distance to the vehicle in the left lane is still large enough, and the ego vehicle can thus safely complete the lane change by executing the intended trajectory. Afterwards, the ego vehicle continues in the left lane, while computed fail-safe trajectories always anticipate possible lane changes of leading vehicles in the right lane (cf. the occupancies in front of the autonomous vehicle in Fig. 6.17 B, fail-safe trajectory). It should be noted that throughout this scenario, the ego vehicle always maintains fail-safe trajectories, but never has to execute one, since the opportunistic intended trajectories do not lead to unsafe situations.

The utilized parameters for this scenario can be found in A.6.2 and A.6.3. Additional figures that highlight the results of planning cycle $c = 1$ are presented in A.6.5.

6.3.3 Jaywalking pedestrians

Vulnerable road users pose a special challenge to autonomous vehicles, since they may unexpectedly change their behavior. In particular, pedestrians are able to quickly alter their walking direction, which makes it difficult for autonomous vehicles to react in a timely manner. According to our legal safety definition, it is illegal for pedestrians to cross the road in the presence of passing vehicles. However, pedestrians are sometimes inattentive and cross nevertheless. This requires careful decisions by autonomous vehicles. In the following scenario, the pedestrian with the blue jacket walks on the sidewalk just in front of the ego vehicle while only looking at his cell phone (cf. Fig. 6.18 A). Later, this pedestrian suddenly crosses the road. If the prediction of the autonomous vehicle does not include this behavior, a fatal accident can occur.

At the beginning of the scenario presented in Fig. 6.18, the inattentive pedestrian ID = 323 in front of the ego vehicle is still walking on the sidewalk, but we want to anticipate that he might cross the road. Thus, we proactively remove the constraint forbidding the pedestrian to cross. This decision to remove certain legal constraints can be made by an additional, predictive module. As a result, SPOT computes

occupancies for both crossing the road and walking on the road parallel to the sidewalk (cf. Fig. 6.18 B, fail-safe trajectory at $t = 0$ s). The resulting fail-safe trajectory ensures that the ego vehicle does not pass the pedestrian. In the next planning step, the ego vehicle cannot verify the new intended motion; in fact, by following this motion, the ego vehicle would hit the crossing pedestrian. Thus, the fail-safe trajectory is automatically executed to slow down the ego vehicle and to avoid a collision with the pedestrian. After the pedestrian has crossed, the ego vehicle accelerates to the desired velocity again (cf. Fig. 6.18 B, $t = 7.8$ s).

The utilized parameters for this scenario can be found in A.6.2 and A.6.3. Additional figures highlighting the results of planning cycle $c = 5$ are presented in A.6.6.

A Scenario overview from recordings



B Planning results

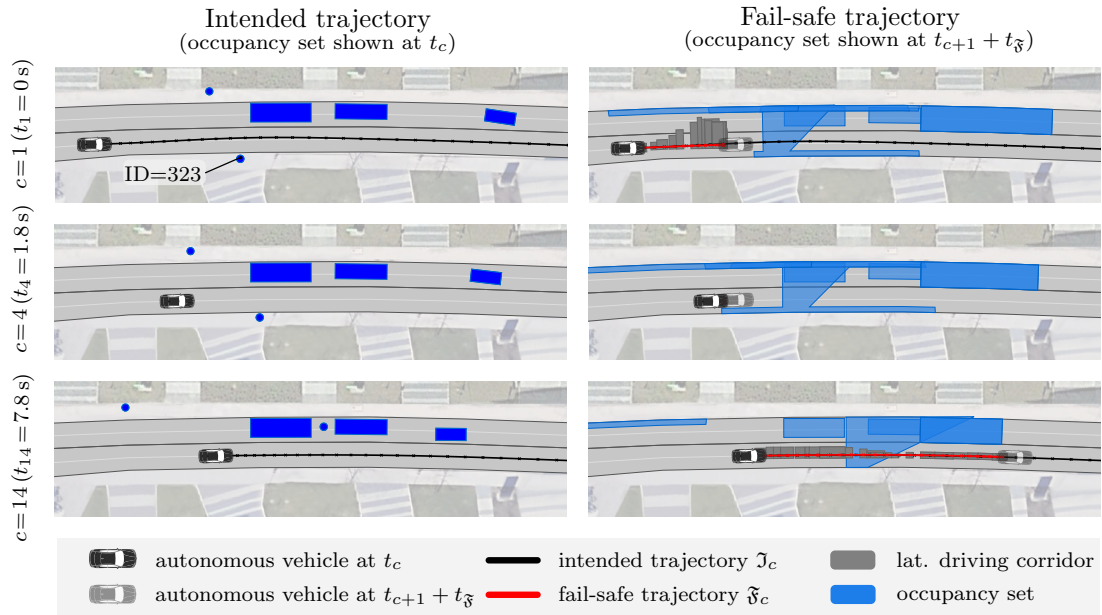


Figure 6.18: Jaywalking pedestrian scenario (DEU_Gar-2.1.T-1). (A) Scenario overview from recordings with images of the vehicle’s front camera and a top view of the traffic situation. (B) The planning and verification results for three selected planning cycles $c \in \{1, 4, 14\}$. Satellite Images ©Google, GeoBasis-DE/BKG.

6.3.4 Verification of arbitrary intended motions

To demonstrate that our verification technique ensures legal safety for arbitrary intended trajectories, we apply our fail-safe motion planning technique to three different intended trajectory planners:

- *Planner 1* uses continuous optimization to plan collision-free intended trajectories with respect to the most likely motion of other traffic participants. This planner is used for the previous results (cf. Sec. 6.3) and is described in A.6.
- *Planner 2* is based on *Planner 1* but ignores other traffic participants in the environment. Thus, obtained trajectories are potentially dangerous.
- *Planner 3* samples intended trajectories in a discrete state space as described in [46]. Obtained intended trajectories are collision-free with respect to the most likely motion of other traffic participants.

Fig. 6.19 illustrates the velocity profile of the ego vehicle in the urban intersection (cf. Sec. 6.3.1) and the jaywalking pedestrian (cf. Sec. 6.3.3) scenario for each intended motion planner. The type of executed trajectory is color-coded to illustrate how often the fail-safe trajectory is executed.

In the urban intersection scenario, our verification technique intervenes independently of the applied intended motion planner such that the ego vehicle stops without entering the intersection. Although *Planner 2* ignores other traffic participants, our proposed fail-safe motion planning technique enables the ego vehicle to safely turn left by triggering fail-safe trajectories more frequently. Since *Planner 2* tries to reach the desired velocity (8 m/s) more aggressively compared to *Planners 1* and *3* in planning cycles 1 to 2 (see Fig. 6.19A), the executed fail-safe trajectories cause a rapid deceleration of the ego vehicle (peak of -6 m/s^2). However, the execution of fail-safe trajectories for *Planner 2* causes only a short delay, as the stopping time at the intersection is less than 2 s.

In the pedestrian scenario, the intended planners are initially not aware of the pedestrian's intention to jaywalk. Therefore, fail-safe trajectories slow down the ego vehicle in planning cycles 2 to 4 for all planners. *Planners 1* and *3* react to the pedestrian as soon as the most likely prediction anticipates that the pedestrian will cross the road, starting at planning cycle 5 (see Fig. 6.19B). In contrast, *Planner 2* requires permanent guidance through the execution of fail-safe trajectories to avoid a collision with the crossing pedestrian (see Fig. 6.19B). Although the type of executed trajectory continuously alternates, the average velocity of the ego vehicle with *Planner 2* is 5% higher than that with *Planner 1* (6.36 m/s and 6.09 m/s, respectively).

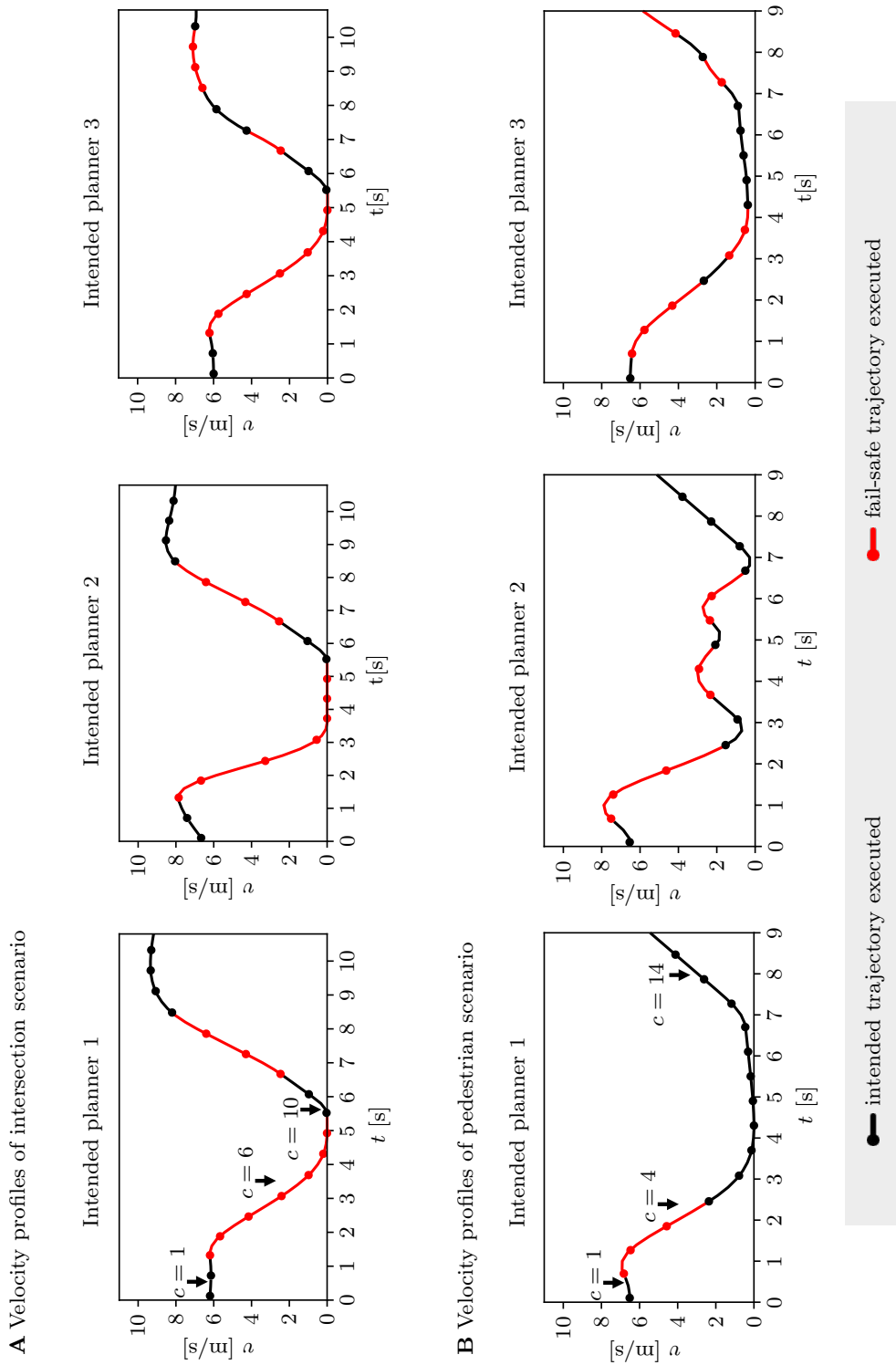


Figure 6.19: Verification of three different intended trajectory planners. (A) Executed velocity profiles in the urban intersection scenario (cf. Sec. 6.3.1). (B) Executed velocity profiles in the pedestrian scenario (cf. Sec. 6.3.3). The labels in the plots for planner 1 correspond to the illustrated planning cycles of the scenarios in Sec. 6.3.1 and 6.3.3.

6.3.5 Summary of experiments with urban traffic scenarios

We demonstrated that fail-safe motion planning ensures safety in the most complex scenarios in urban traffic using three different experiments. For instance, fail-safe trajectories ensure that the ego vehicle safely enters intersections or changes lanes. Moreover, we showed how fail-safe motion planning performs when used over consecutive planning cycles. Even if the ego vehicle has to execute a fail-safe trajectory, it can recover and return to its intended trajectory once the new situation has been verified. Fail-safe motion planning is able to guarantee the safety of the ego vehicle for different planners even when using a planner that ignores other traffic participants. For all considered planners, the resulting velocity profiles are smooth and continuous, since fail-safe trajectories are planned with full consideration of the vehicle’s dynamics and the usage of slack variables.

6.4 Assessment of Intervention Rates and Passenger Comfort

The previous experiments focused on the safety benefits of executing fail-safe trajectories. However, system designers are also interested in the conservativeness of formal verification approaches - that is, the number of interventions and the necessity of each intervention. If the specification holds, (over-approximative) formal methods never produce false negatives (FNs) by definition (i.e., when a situation classified as safe is in fact unsafe), instead generating false positives (FPs) (i.e., when a situation classified as unsafe is in fact safe). Tab. 6.2 illustrates the confusion matrix for the proposed fail-safe motion planning technique. The conservativeness of the formal verification approach applied to a certain system roughly relates to the ratio of FPs to the sum of FPs and true positives (TPs), denoted as the false discovery rate (FDR) [293]:

$$\text{FDR} := \frac{\text{FP}}{\text{FP} + \text{TP}}. \quad (6.1)$$

High values of the FDR may amount to less passenger comfort, since the safety layer intervenes more frequently without justification. Moreover, since fail-safe trajectories start at the last possible point in time, only the transition from the intended trajectory to the fail-safe trajectory is jerk-optimal, but the fail-safe trajectory itself is not the global jerk-optimal solution (the most comfortable maneuver does not start at the time-to-react). As a result, longer executions of fail-safe trajectories may be perceived by passengers and influence their comfort. In the following two sections, we investigate both the comfort of our proposed fail-safe motion planning technique and its intervention rate in typical driving situations.

Table 6.2: Confusion matrix of fail-safe motion planning.

true positives (TPs) fail-safe trajectory executed and situation is unsafe	false positives (FPs) fail-safe trajectory executed, but situation is safe
false negatives (FNs) fail-safe trajectory not executed, but situation is unsafe	true negatives (TNs) fail-safe trajectory not executed and situation is safe

6.4.1 Adaptive cruise control user study

The passenger comfort provided by our fail-safe motion planning technique is assessed within a driving simulator. Since no mature autonomous driving systems are available yet, we compare the comfort to an adaptive cruise control system (ACC) as a baseline instead. ACC systems automate the longitudinal motion of the vehicle while the driver is still controlling the lateral motion through the steering wheel. For this user study, we focus on highway scenarios.

The user study is conducted in a static driving simulator at the BMW Autonomous Driving Campus in Unterschleißheim (cf. Fig. 6.20) [5]. During the study, participants are able to monitor the current velocity and the surrounding environment of the vehicle (including rear mirrors) using three displays. In each



Figure 6.20: BMW driving simulator. A study participant steers the simulated vehicle in a safety-critical scenario of the user study [5]. The longitudinal motion is automated by an adaptive cruise control system (ACC), which is supervised by the proposed fail-safe motion planning technique. We compare the passengers' comfort and feeling of safety by enabling and disabling the verification layer for different runs of a scenario.

scenario, the longitudinal motion of the vehicle is automated by an ACC system that aims to maximize the passenger comfort. To assess passengers' comfort and feeling of safety, the study participants have to steer the simulated vehicle in different safety-critical highway situations with and without our fail-safe motion planning technique enabled. After each scenario, the participants have to rate the performance of the ACC system in a questionnaire.

In total, each participant faces five different scenarios (each with and without the safety layer) that model traffic jams or cut-in vehicles (each scenario is detailed in A.7.1). To make the scenarios more realistic, we add extra traffic participants (vehicles) to each scenario; at most, this has a very limited influence on the scenario's criticality. Moreover, we slightly change the visual appearance of each scenario when the safety layer is enabled. This choice is made so that participants are not able to immediately recognize that a scenario is being shown a second time. During the user study, each participant experiences the scenarios in a different order to prevent bias in the participants' evaluations. Fig. 6.21 shows the simulation view of one of the safety-critical scenarios (additional figures are given in A.7.2).

The default ACC system (the ACC system without the fail-safe motion planning enabled) tries to maintain the minimum inter-vehicle distance according to German law (i.e., approximately a 2 s time gap based on the current velocity). It considers only a single leading vehicle and no cut-in vehicles. If the default ACC is unable to determine a feasible control input that satisfies the vehicle's constraints, linear (jerk-compliant) deceleration is performed until the maximum deceleration is achieved. On the other hand, the supervised ACC is based on the default ACC, but each planned control input is safeguarded by the proposed fail-safe motion plan-



Figure 6.21: Front view of a user sitting in the driving simulator. Study participants are able to monitor the velocity and the environment of the vehicle. If the simulated ego vehicle brakes or accelerates, the view tilts accordingly to give participants visual feedback.

ning technique. The computed fail-safe trajectories only influence the longitudinal motion of the vehicle. Moreover, the supervised ACC receives information about cut-in vehicles and other vehicles in the current lane. In the experiments, we use a simplified version of SPOT. This version only predicts the feasible longitudinal motion of other traffic participants. Cut-in maneuvers of vehicles in adjacent lanes that intend to change to the ego vehicle’s lane (signaled by an external predictive module) are immediately projected to the ego vehicle’s lane and predicted by SPOT.

In the user study, we aim to assess the following hypotheses:

- H1) The feeling of safety with the supervised ACC is at least as high as with the default ACC.
- H2) The passenger comfort provided by the supervised ACC is at least as high as that provided by the default ACC.

During the course of the study, we ask each of the 31 participants to rate the performance of the tested ACC system with the following two questions (Q) and possible answers (A) after each scenario:

- Q1) How do you rate the feeling of safety provided by the algorithm?
A: very low, low, medium, high, very high
- Q2) How do you rate the comfort of this algorithm?
A: very low, low, appropriate, high, very high

Each answer is coded with a numerical value in the range from 1 to 5 (ordinal scale) for the evaluation. The statistical analysis of the user study is performed with the tool JASP [294]. The two hypotheses, H1 and H2, are evaluated using the Wilcoxon

Table 6.3: Results of the Wilcoxon signed-rank t-test. Bold p -values indicate statistically significant results.

Scenario	Hypothesis	Z-value	p-value
Scenario 1	H1	110.00	0.431
	H2	170.50	0.159
Scenario 2	H1	205.00	0.019
	H2	383.00	0.001
Scenario 3	H1	225.50	0.014
	H2	201.00	0.026
Scenario 4	H1	136.00	0.001
	H2	403.50	0.001
Scenario 5	H1	165.00	0.034
	H2	124.50	0.107

signed-rank t-test [295]. Tab. 6.3 lists the resulting Z - and p -values for each of the five scenarios in the user study. In our analysis, $p \leq 0.05$ indicates statistically significant results - in other words, one ACC system obtains a significantly higher score than the other one for the considered question. The statistical analysis reveals significant results for hypothesis H1 in scenarios 2, 3, 4, and 5 and for the hypothesis H2 in scenarios 2, 3, and 4. In scenarios 1 and 5, no significant results are achieved. This is because the behavior of the safe ACC can hardly be distinguished from the default ACC, since both scenarios involve similar emergency brake maneuvers by the ego vehicle. However, the average score (given by the study participants) of the supervised ACC is at least as good as that of the default ACC in all scenarios.

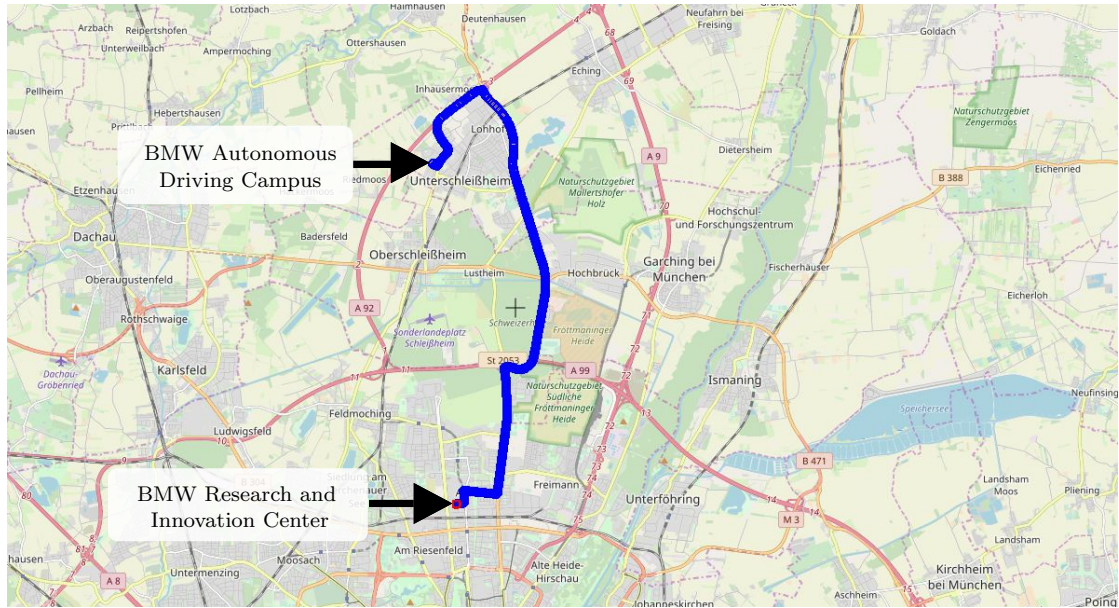
Based on the conducted statistical analysis, we can conclude that the execution of fail-safe trajectories (for the longitudinal motion of an autonomous vehicle) increases passengers' feeling of safety without compromising comfort. One reason for this may be that fail-safe trajectories already anticipate all feasible, legal behaviors of other traffic participants and thus safely handle each situation. Even though the vehicle has to execute a fail-safe maneuver, passengers benefit from the jerk-optimal entry into fail-safe trajectories. We note that the results cannot directly be mapped onto the proposed fail-safe motion planning for combined longitudinal and lateral control; however, the first results are already promising. After analyzing the comfort of fail-safe trajectory executions, we are interested in investigating how often fail-safe trajectories have to be executed in typical traffic scenarios.

6.4.2 Intervention assessment in dense urban traffic

In this section, we assess the intervention rate of the proposed fail-safe motion planning technique in typical urban traffic situations. To this end, we use a BMW 7-series test vehicle to record urban scenarios with dense traffic. All parameters are given in A.8. Since no intended trajectory planner is available, we use the fail-safe motion planning technique to verify the safety of the current control input of the human driver instead. As a result, the intended trajectory of the vehicle corresponds to the human driver's currently chosen input, which is kept constant for a time horizon of 6 s. For safety reasons, we postprocess the data after the test drives with our fail-safe motion planning technique. The postprocessing also allows a detailed analysis of each fail-safe trajectory execution.

Fig. 6.22a shows the 17 km-long route of the driving experiment, carried out between the BMW Research and Innovation Center (FIZ) in Munich and the BMW Autonomous Driving Campus (ADC) in Unterschleißheim. This route covers different urban (cf. crossing pedestrian in Fig. 6.22b) and country road situations (cf. four-lane road in Fig. 6.22c). For most of the roads along this route, the human driver has the right of way. Intersections are controlled by traffic lights. The speed limit ranges from 8.3 m/s in urban areas to 27.8 m/s on country roads.

We conduct four test drives (two in each direction) with a BMW 7-series test vehicle on Wednesday, 13th of March 2019, from 1:30PM until 5PM (usual afternoon commuter traffic). Each drive takes 23 min on average, which implies a mean



(a)



(b)



(c)

Figure 6.22: Route of the intervention assessment study. (a) The study is conducted on the 17 km-long route (©OpenStreetMap contributors) between the BMW Autonomous Driving Campus in Unterschleißheim and the BMW Research and Innovation Center in Munich. (b) The route covers urban areas with vulnerable road users and high traffic densities. (c) The route also covers country roads with velocities of up to 27.8 m/s.

velocity of approximately 12.32 m/s. We sample different traffic densities between recordings. The environment model of the test vehicles has an update frequency of about 5 Hz. The planning horizon of our fail-safe planner is set to $t_{\text{f}} = 5$ s with a step size of $\Delta t = 0.25$ s. The exact parameters of the fail-safe planner and the set-based prediction are listed in A.8.1.

During the test drives, the vehicle has to react to different types of traffic participants: bicycles, vehicles, trucks, buses, motorcycles, and pedestrians. To account for sensor limitations, the set-based prediction adds phantom obstacles to the borders of the ego vehicle's field of view (lateral sight of 100 m and longitudinal sight of

150 m). The average computation times of the prediction and the fail-safe planner are 20.1 ms and 16.1 ms per call, respectively. It should be noted that the human driver is aware of the safety layer in the vehicle, but has the task of driving as normally as possible. He receives no feedback concerning whether or not his driving style is safe.

Since we are interested in the intervention rate of the safety layer, we present the results of the test drive with the most executions of fail-safe trajectories. This test drive is recorded from 2:48PM until 3:09PM (duration of 21 min) on the route from FIZ to ADC. In total, $N_{\text{attempt}} = 6,157$ verification attempts are performed during this test drive, corresponding to a rate of 4.7 Hz. Among these attempts, $N_N = 6056$ situations (98.55 %) are verified as safe by successfully computing a fail-safe trajectory. Fig. 6.23 shows two situations in which the verification is successful (additional figures are presented in A.8.2). All $N_N = 6056$ safe situations are safe according to the proposed legal safety specification; thus, no false negatives (FNs) are generated (cf. Tab. 6.2). Only in $N_P = 101$ cases (1.64 %), the current traffic scenario cannot be verified and the ego vehicle has to execute the previously generated fail-safe trajectory. We manually investigate each verification attempt in detail.

Regarding the number of failed verification attempts, the true positives and false positives amount to $TP = 47$ and $FP = 54$, respectively. Tab. 6.4 summarizes the analysis results of the alleged fail-safe executions during the test drive. For true positives, most fail-safe trajectory executions are caused by the driver violating the safe distance to preceding vehicles (55.3 %). The second major reason for unsafe situations is high uncertainty in the environment model (38.3 %). Even in uncertain scenarios, the safety layer needs to account for these uncertainties to prevent collisions. The last reason for justified fail-safe executions in our experiment is observed in one situation in which a pedestrian suddenly enters the road (6.4 %). In this scenario, the safety layer is unable to compute a new fail-safe trajectory that allows the ego vehicle to come to a standstill in its current lane. Here, the human driver intervenes by slightly occupying the adjacent lane with opposite driving direction while passing the pedestrian.

Considering the 1% of unjustified fail-safe trajectory executions (cf. Tab. 6.4), the majority of false positives amount to unmodeled traffic rules in our legal specification (61.1 %). More specifically, right of way rules are not yet included in our specification (and implemented in the prediction), since they require the future intentions of traffic participants, including the ego vehicle. For instance, Fig. 6.24a illustrates a situation from the test drive in which the ego vehicle intends to turn right. However, an oncoming vehicle is also allowed to turn left in this situation, which is in conflict with the decision of the autonomous vehicle. Another major cause for fail-safe executions lies in the utilized solver (25.9 %), which sometimes fails to obtain fail-safe trajectories. Analyses of the solver failures do not reveal the source of error; the constraints do not indicate infeasibility of the optimization problems. We assume that the errors may be caused by the Python interface, which processes the data of the C++ implementation of the solver. Lastly, 13 % of false

Table 6.4: Analysis results of alleged fail-safe trajectory executions.

Type	Reason	Number	Comment
TP	Safe distance	26	The driver violates the safe distance to preceding vehicles.
	Pedestrian	3	A pedestrian suddenly enters the ego vehicle’s lane.
	Uncertainties	18	High uncertainties in the environment model lead to the rejection of intended trajectories.
FP	Solver error	14	The solver of the fail-safe trajectory planner fails to solve the optimization problem (reasons not comprehensible).
	Vehicle leaves road	7	A vehicle enters a parking area and leaves the current map area. In these situations, the set-based prediction can only predict the legal dynamic behavior without considering lanes and driving directions.
	Traffic rules	33	The right of way is not yet implemented in the set-based prediction and thus, vehicles are predicted to turn in front of the ego vehicle.

positives are caused in a single traffic situation in which a preceding vehicle enters a parking area, leaving the map area. If a vehicle leaves the map, the set-based prediction can only predict the dynamical occupancy of the traffic participant, since we cannot be sure of where the traffic participant is driving. This dynamic occupancy grows enormously over time and intersects with the initial position of the ego vehicle (cf. Fig. 6.24b). Thus, the map data provided to our fail-safe motion planning technique must be complete.

In our intervention rate study, we can conclude that the ego vehicle only has to execute fail-safe trajectories in less than 2% of the verification attempts. Based on the presented numbers, the false discovery rate (FDR) of the proposed fail-safe motion planning approach corresponds to $\text{FDR} = 53.47\%$. Thus, if the ego vehicle has to execute a fail-safe trajectory, approximately half of the executions are justified. The longest execution of a fail-safe trajectory before recovery takes place in the false positive scenario in which a vehicle leaves the map. Here, the last known fail-safe trajectory lets the ego vehicle reduce its velocity from 13.61 m/s to 8.89 m/s in 1.75 s. All results are obtained in dense urban traffic in the area of Munich. To reduce the FDR, we identify the following three challenges:

1. *Numerically stable solver:* Solvers for computing fail-safe trajectories must converge to the optimal constrained solution if the problem is feasible. Moreover, solvers need to return the same solution within given numerical bounds when solving a certain problem multiple times.
2. *Traffic rules and interaction:* The implementation of additional traffic rules in the set-based prediction further tightens the obtained over-approximation. In addition, new approaches are needed to consider the interaction with the ego vehicle, for instance in situations with right of way regulations.

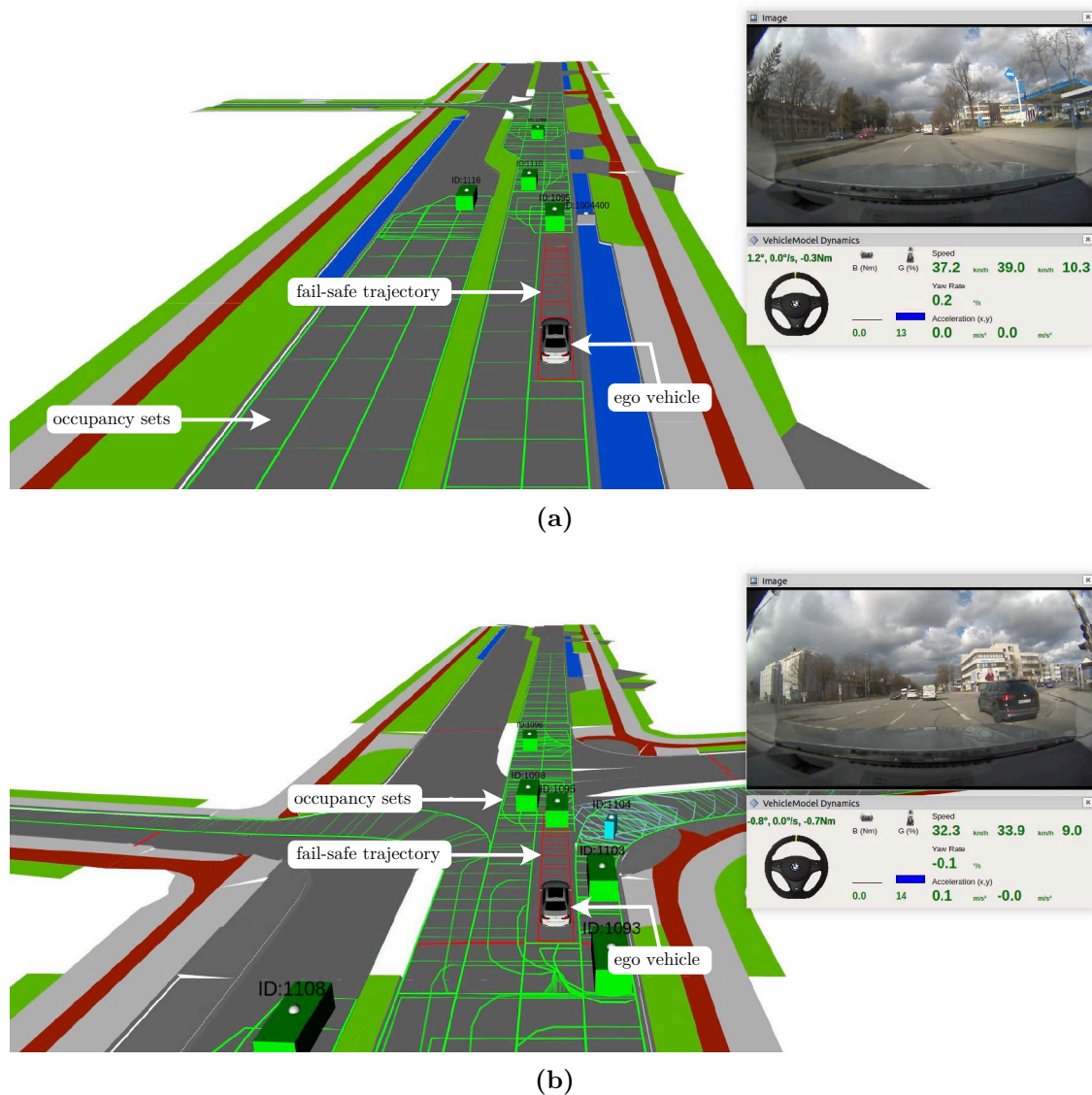


Figure 6.23: Examples of true negatives. (a) The verification successfully computes a fail-safe trajectory (red regions) in a two-lane scenario. (b) Successfully computed fail-safe trajectory at an intersection.

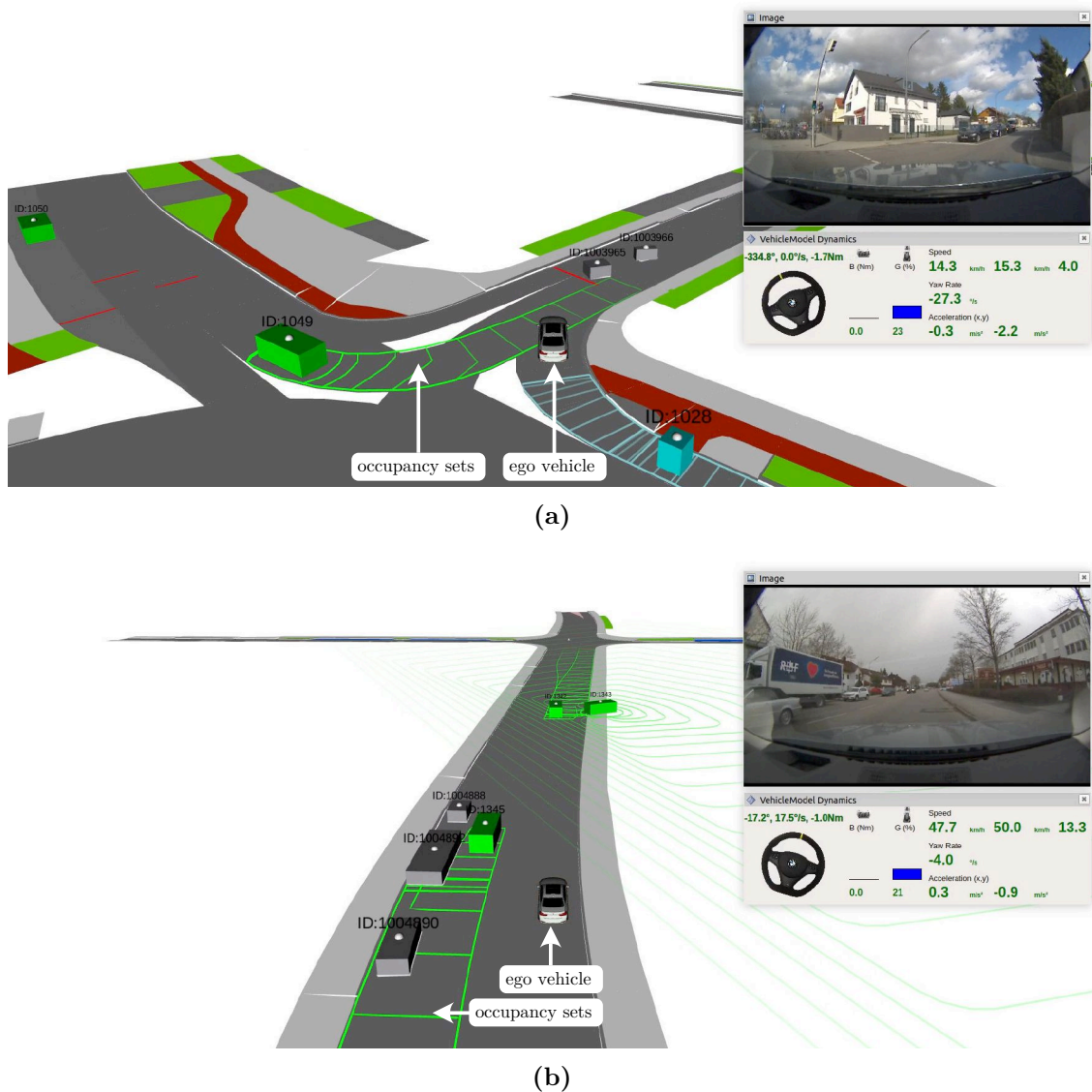


Figure 6.24: Examples of false positives. (a) Priority traffic rules are not yet implemented in the set-based prediction. The ego vehicle cannot compute a fail-safe trajectory, since the green vehicle is predicted to turn left, merging into the ego vehicle’s lane. (b) A vehicle is leaving the road to enter a parking area. In these cases, the set-based prediction can only predict the dynamic behavior, which results in overly large occupancy sets (green areas).

3. *Environment model*: Although the proposed safety framework is able to compensate for environment models with poor data quality (garbage in, safety out), environment models need to be as accurate as possible to reduce false positives. The provided map data and types of traffic participants are important inputs for the set-based prediction and heavily influence the resulting occupancy sets.

6.4.3 Summary of conducted studies

In a user study with 31 participants in a driving simulator, we evaluated the comfort of fail-safe trajectories and passengers' feeling of safety. We equipped an adaptive cruise control system with our fail-safe motion planning technique. Study participants had to steer the vehicle in different traffic scenarios with and without our verification enabled, and they subsequently rated their comfort and feeling of safety. Our results indicate that the use of fail-safe motion planning improves the overall feeling of safety for passengers while not compromising comfort. In a second study, we investigated the intervention rate of our verification technique in typical traffic scenarios. We postprocessed recorded scenarios with dense traffic and analyzed the number and the cause of fail-safe trajectory executions. Our results indicate that fail-safe motion planning has low intervention rates of less than 1.7%. This number can be further reduced, since half of the executions were caused by the missing implementation of right of way rules in our legal specification and the set-based prediction.

6.5 Summary

In this chapter, we evaluated the performance of our fail-safe motion planning technique in various experiments and situations. We implemented the approach in a BMW 7-series test vehicle to demonstrate its safety benefits for autonomous vehicles. In 127 experiments on a fenced test track, we demonstrated that the execution of fail-safe trajectories ensures that autonomous vehicles remain safe at all times. This holds true even if the intended trajectory is planned by a malicious planner that randomly ignores obstacles in the environment. Moreover, invariably safe sets were successfully used to determine the point in time to execute fail-safe trajectories. Our vehicle experiments further validated our developed trajectory planning approach by testing trajectories near the physical limits. For instance, we performed evasive maneuvers with lateral accelerations of 4.8 m/s^2 .

Since we were unable to model complex traffic scenarios on the test track, we also successfully applied our fail-safe motion planning technique to recorded traffic scenarios. These scenarios were recorded in the area of Munich. Among others, our experiments confirm that fail-safe trajectories ensure that autonomous vehicles enter intersections without endangering other traffic participants, react to jaywalking pedestrians, and perform safe lane changes. In addition, we validated that

the proposed safety layer works with any provided intended trajectory. To this end, we used three different planning approaches including a planner that ignores other traffic participants. In all scenarios, the proposed fail-safe motion planning technique guaranteed the safety of the computed trajectories.

Besides validating the safety benefits of fail-safe motion planning, we also investigated the passenger comfort provided by fail-safe trajectories. We conducted a study with 31 participants in a driving simulator to test whether an adaptive cruise control system provides higher comfort with or without our safety layer enabled. Our statistical analysis revealed that fail-safe trajectories yielded significantly higher passenger comfort in most of the test scenarios. Furthermore, study participants' reported feeling of safety significantly increased when our safety layer supervised the adaptive cruise control system.

Another major part of our evaluation was assessing the intervention rate of the proposed fail-safe motion planning layer. We analyzed the number of fail-safe trajectory executions and their cause by postprocessing recorded traffic situations. To stress test our safety layer, these scenarios were recorded in dense commuter traffic in the area of Munich. We manually examined the 6,157 verification attempts. In less than 2% of attempts only, the autonomous vehicles executed the previously computed fail-safe trajectory. The false discovery rate was estimated at about 53%. The major causes of false positives were the missing implementation of additional traffic rules, interaction between traffic participants, and the solver stability when solving the fail-safe trajectory optimization problems.

The conducted experiments represent one of the most sophisticated studies of formal verification for autonomous vehicles to date. Our results indicate that the safety benefits of the proposed fail-safe motion planning technique for autonomous vehicles hold in reality. Even in complex traffic situations, the safety of autonomous vehicles is ensured at all times. Furthermore, our studies reveal that the application of fail-safe motion planning has positive effects on passengers' comfort and feeling of safety. In addition, fail-safe motion planning is not expected to result in overly conservative behaviors of the autonomous vehicle.

7 Conclusions and Perspectives

In this thesis, we developed a novel online verification technique that is able to ensure that autonomous vehicles operate according to the powerful concept of legal safety. We demonstrated our fail-safe motion planning approach in numerous experiments on a real test vehicle and recorded datasets. In this concluding chapter, we first summarize our theoretical and practical contributions toward the goal of provably safe motion planning in Sec. 7.1. Afterwards, in Sec. 7.2, we discuss the remaining steps to realize our verification technique in autonomous series vehicles and the impact of online verification on the development of future autonomous systems. Finally, we conclude this chapter by examining future research directions in Sec. 7.3, and providing closing remarks in Sec. 7.4.

7.1 Summary of Contributions

After introducing the concept of legal safety in Ch. 1 and 2, we developed a real-time capable approach to compute fail-safe trajectories in Ch. 3. Fail-safe trajectories serve as collision-free fallback routines along intended motions of the autonomous vehicle. In case other traffic participants deviate from the predicted most likely motion (used to compute the intended motion of the autonomous vehicle), the autonomous vehicle can execute the fail-safe trajectory to remain collision-free. To compute these fail-safe trajectories, this thesis proposed a novel trajectory planning method that exploits optimization theory. By separating planned motions into longitudinal and lateral components and linearizing vehicle models, we are able to formulate the trajectory generation as convex optimization problems. Moreover, we demonstrated how collision avoidance can be efficiently integrated as linear constraints into the optimization problems. As a result, our technique is able to obtain collision-free fail-safe trajectories in real-time with global convergence. In addition, we presented a novel approach to determine drivable fail-safe trajectories in complex, narrow search spaces by combinatorial enumerations or the computation of the drivable area. The latter allows us to efficiently determine driving corridors to plan feasible fail-safe trajectories. We demonstrated the novelties and safety benefits of the proposed fail-safe trajectory planning approach in various numerical experiments.

Then, in Ch. 4, we introduced invariably safe sets as a technique to compute safe states for autonomous vehicles. If the state of the autonomous vehicle is invariably safe, it is guaranteed that a trajectory exists that remains safe for an infinite time horizon. In contrast to existing safe states concepts in robotics, our

novel algorithm is able to efficiently compute safe states in dynamic environments with linear time complexity. Thus, invariably safe states are the first technique to ensure safety over infinite planning horizons; this can be used during the operation of autonomous vehicles. Furthermore, we demonstrated that invariably safe states have significant safety benefits for motion planning. For instance, they can be used to verify trajectories for infinite time horizons or to determine the time-to-react until which the existence of a safe trajectory is guaranteed. The proposed safety benefits were illustrated in various numerical experiments, such as verifying the safety of machine learning approaches and ensuring the existence of fail-safe trajectories.

We proposed our novel fail-safe motion planning technique in Ch. 5. It combines the previously introduced concepts, namely fail-safe trajectory planning and invariably safe states, to ensure legal safety during the operation of autonomous vehicles. We apply 1) set-based prediction to handle measurement uncertainties and to compute all possible legal behaviors of other traffic participants online, and 2) fail-safe trajectory planning to ensure that autonomous vehicles only execute provably safe motions that keep the vehicle in invariably safe states. While the autonomous vehicle is moving along its intended trajectory, our verification technique continuously maintains fail-safe trajectories at all times. We demonstrated how the proposed online verification technique can be integrated in most state-of-the-art motion planning frameworks. The technique only requires the current environment model and arbitrarily planned trajectories as input. It returns provably safe trajectories, which are stored in redundant memory to ensure fail-safe operation. Furthermore, we formally proved that the proposed fail-safe motion planning approach is correct-by-construction according to our legal specification. We subsequently demonstrated the computation steps to verify arbitrarily planned trajectories. In addition, we derived how the under-approximation of invariably safe sets can be linearized for usage in the presented fail-safe trajectory planning approach.

Finally, Ch. 6 presented the results of our fail-safe motion planning technique in experiments with real test vehicles. We implemented the proposed verification technique as a prototype in C++ and Python for use in a BMW 7-series test vehicle. In various experiments on a fenced test track, we demonstrated that fail-safe motion planning ensured the safety of the vehicle. Moreover, we validated that the proposed verification technique can be successfully applied to complex urban traffic situations. We recorded these real traffic scenarios in the area of Munich; they correspond to the most dangerous situations in urban environments. By applying our verification technique to different intended trajectory planners (one of them ignoring other traffic participants), we demonstrated that our online verification ensures legal safety for arbitrary intended trajectories. Two conducted studies (postprocessing test drives with dense urban traffic and a simulator study with human participants) indicated that fail-safe motion planning achieved low intervention rates and provided high passenger comfort.

In summary, this thesis presents the following significant novel theoretical contributions:

- T1** The development of the first correct-by-construction verification technique to ensure the legal safety of autonomous vehicles. The verification technique works with arbitrary motion plans and during the operation of the vehicle.
- T2** The exploitation of convex optimization theory to develop the first real-time capable fail-safe trajectory planning approach, which enables autonomous vehicles to recover from potentially unsafe situations. The generated fail-safe trajectories are drivable and produce smooth braking and curvature profiles.
- T3** The combination of drivable area computation and variational trajectory planning to create a new dynamics-aware method to explore non-convex search spaces. The obtained driving corridors can be used to plan drivable fail-safe solutions in arbitrarily complex traffic situations.
- T4** The introduction of invariably safe states as a powerful, universal technique to compute safe states for motion planning of autonomous vehicles. Contrary to existing approaches, an under-approximation of invariably safe states can be obtained in real-time and tightened if computation time remains.
- T5** Advancements to infinite time horizon planning during the operation of autonomous vehicles by exploiting invariably safe sets. Among others, this novel approach allows vehicles to verify trajectories over infinite time horizons and to compute the time-to-react.
- T6** The extension of set-based prediction to predict all legal future behaviors of pedestrians. This can predict the legal motion of almost all types of traffic participants.

Furthermore, the following significant novel practical contributions are demonstrated in this thesis:

- P1** The prototypical implementation of the presented fail-safe motion planning technique in C++, Python, and ROS. The implemented modules were used in a BMW 7-series test vehicle.
- P2** The demonstration of the power of fail-safe motion planning in various experiments on a fenced test track with a test vehicle. In total, 127 tests were conducted, resulting in 12 h of recorded data.
- P3** The demonstration of the safety benefits in recorded urban traffic scenarios in the area of Munich. Even in the most critical situations in urban

environments, fail-safe motion planning ensures safety at all times with negligible performance losses.

- P4** The presentation of the first in-depth performance analysis of online verification for autonomous vehicles. The obtained results suggest that fail-safe motion planning has low intervention rates of up to 1.64 % and provides significantly higher passenger comfort compared to driver assistance systems without dedicated verification techniques.
- P5** The demonstration that high levels of safety can be achieved without compromising usability. The proposed fail-safe motion planning technique can be integrated in almost any motion planning framework, even when using machine learning components for the intended motion planner.
- P6** The distribution of data for usage by other researchers and interested parties. All scenarios are publicly available as part of the CommonRoad benchmark suite for motion planning. In addition, the Python tools of CommonRoad were advanced during the course of this research project.

7.2 Impacts of Fail-Safe Motion Planning

In the following sections, we discuss the remaining steps to realize fail-safe motion planning and the impact of online verification for robotic systems [2].

7.2.1 Certification

Certification is the main challenge in introducing the proposed verification technique onto the market and in series vehicles. Regulatory guidelines have already been passed for various domains, such as railway systems, industrial robots, and aviation systems, but only limited regulations exist for the motion planning of autonomous vehicles. For instance, the maximum speed of pedestrians in our legal specification is based on the ISO norm 13855, but this norm is designed for workers in production plants to ensure safe human-robot collaboration [131]. With missing regulations in mind, the fail-safe motion planning technique has been designed to easily adapt to new specifications (since the set-based prediction serves as an input of the verification technique). In the future, policy makers, manufacturers, and mobility providers must agree on applicable legal safety specifications and safe states for autonomous vehicles. This thesis demonstrates how selected formalized traffic rules can be used to ensure legal safety during the operation of autonomous vehicles; however, additional progress is required. When legal safety becomes a recognized standard for autonomous vehicles (e.g., as part of the ISO norms), our proposed fail-safe motion planning technique can be certified for usage in autonomous vehicles.

7.2.2 Merits of self-verifying robots

Self-verification will boost the efficiency of development processes and increase the societal trust in autonomous vehicles. Autonomous vehicles that incorporate fail-safe motion planning execute only provably safe actions at all times, even in untested scenarios and regardless of the intended trajectories' safety. The proposed safety verification technique automatically adapts to the current road network and the states of surrounding traffic participants (including measurement uncertainties). As a result, autonomous vehicles must be tested significantly less during the design phase. The reduced development effort without sacrificing safety can allow mobility providers to easily change and improve components to generate intended trajectories. This is particularly beneficial when using machine learning components, which can provide increased comfort due to their automatic adaptation to new environments. However, machine learning components are difficult to verify [159], as their output may change unexpectedly over learning episodes and may lead to unsafe situations in previously tested scenarios. In addition, our provable and comprehensive verification technique is expected to reduce liability claims for autonomous vehicles. Since our technique ensures that autonomous vehicles are safe with respect to all possible legal behaviors, collisions are only possible if other traffic participants have violated traffic rules.

7.2.3 Toward safe human-robot coexistence

Strict safety will enable new and exciting robotic applications. Although safety verification has been studied extensively from a theoretical perspective, only few robotic systems actually use it. This low adoption rate of verification techniques is usually connected to the complexity of ensuring safety in arbitrary scenarios while accounting for uncertainties or the belief that strong guarantees are only provided when full adversarial behaviors of the environment are assumed. In contrast to existing verification techniques (cf. Sec. 1.1), the proposed fail-safe motion planning technique addresses safety in a holistic way by considering arbitrary traffic scenarios while only incorporating behaviors of other traffic participants that are necessary to ensure safety in traffic - that is, those that comply with a legal specification. In addition, our verification technique puts particular emphasis on robust performance by always maintaining provably safe trajectories. With our extensive evaluation in driving experiments, recorded traffic scenarios, and conducted studies, this work is the first to demonstrate that autonomous vehicles can provide a high level of legal safety despite operating in uncertain environments.

Online verification is often believed to cause performance drops and conservative behavior in robotic systems [292, 296–298]. Although opportunistic autonomous vehicles may have a higher performance, these performance gains come at the cost of endangering the lives of other traffic participants. In contrast, we propose to guarantee legal safety by safeguarding autonomous vehicles if necessary without endangering the lives of other traffic participants. The initial results of this thesis

demonstrate that performance does not significantly suffer from the execution of fail-safe trajectories. In fact, we showed that autonomous vehicles can accomplish complex tasks with negligible performance losses—even if intended trajectories are not aware of obstacles. However, if higher performance is desired, the verification technique can be further improved (cf. Sec. 7.3).

7.3 Perspectives

In the following sections, we briefly discuss future work to obtain a code that complies with the highest automotive safety integrity level (ASIL) [281], to ensure the drivability of fail-safe trajectories despite disturbances acting on the controller, and to further improve the performance of fail-safe motion planning.

7.3.1 ASIL-D compliant safety layer

The presented architecture of our fail-safe motion planning technique (cf. Sec. 5.1) has the important advantage that techniques developed for planning intended motions do not have to be certified (cf. ISO 26262 [281]). This design allows the code of the intended motion planner to be changed or updated at any time. However, the code of our safety layer needs to be certified. In this thesis, we developed a mathematical model that is provably safe according to the definition of legal safety (cf. Fig. 7.1). Nevertheless, our prototypical implementation may result in unsafe behaviors of the vehicle, with the reasons for this discrepancy lying in the implementation and used system (hardware, operating system, other components): neither are verified against our formal specification. For instance, buffer overflows may cause incorrect computations, and failures in the braking system may result in the inability to stop the vehicle.

Fig. 7.1 illustrates the missing steps toward creating a safety layer that fulfills the ASIL-D standard [281] for automotive components, which is the highest ASIL. ASIL-D certified components are required to have a failure rate of less than $1 \cdot 10^{-8}/\text{h}$, since any failure in these components may lead to fatal injuries. In the following, we focus on safe motion planning. To ensure that our proposed safety layer conforms to ASIL-D, all safety-relevant components must comply with the ASIL-D standard. For instance, the autonomous vehicles need to be equipped with redundant hardware to neglect hardware failures [299, 300]. In addition, the implementation of our fail-safe motion planning technique needs to be verified against our specification using formal methods. The modules of our verification technique consist of only a few thousand lines of code. In a first step, we could represent our mathematical model of fail-safe motion planning in a formal, machine-parsable language, such as Isabelle [95] or SPARK [301]. Subsequently, model checking tools could be used to automatically verify whether the implementation of our safety layer conforms to our mathematical model [89]. If the verification is successful, the implementation produces only safe behaviors according to the legal specifica-

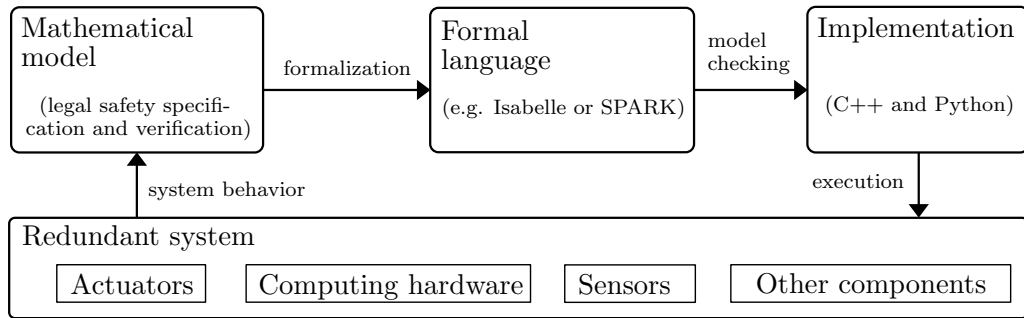


Figure 7.1: Code verification for ASIL-D compliance. Our mathematical model of the legal safety specification and fail-safe motion planning technique needs to be formalized in a formal, machine-parsable language, such as Isabelle or SPARK. Subsequently, model checking is used to verify the correctness of our implementation in C++ and Python.

tion. With redundant hardware (ASIL-D compliant) and a verified implementation of fail-safe motion planning, the autonomous vehicles operate according to legal safety and the ASIL-D safety standard [302].

It should be noted that since the physical vehicle cannot be exactly modeled, the conformance of the system behavior to the mathematical model needs to be shown afterwards [303–306]. For instance, the ego vehicle may not be able to perfectly track a given trajectory due to disturbances. Therefore, the system behaviors need to be recorded and analyzed.

7.3.2 Ensuring drivability despite disturbances

In all of our driving experiments, the execution of fail-safe trajectories ensured the safety of the autonomous vehicles. However, even though our fail-safe motion planning technique may find a collision-free fail-safe trajectory, the autonomous vehicle might deviate from this trajectory when executing it. In fact, we recorded a maximum position error of 2.25 m in our experiments. These deviations may occur due to sensor noise and disturbances acting on the controller of the vehicle and inaccurate models of the vehicle’s dynamics in the controller (since it is difficult to obtain correct models). As a result, the autonomous vehicle is not able to perfectly track the fail-safe trajectory, resulting in collisions in the worst case.

Optimal control techniques can be used to ensure the drivability of fail-safe trajectories despite controller inaccuracies. For instance, the approach in [115] ensures the drivability of trajectories by fitting the trajectories with motion primitives. For these motion primitives, we can pre-compute controllers and reachable sets to obtain set-based optimal controllers. These controllers allow the autonomous vehicle to 1) ensure the drivability of trajectories and 2) determine the maximum deviation from the trajectory with given bounded sets of uncertainties. Estimates of the

7 Conclusions and Perspectives

bounded sets can, for example, be obtained from previous test drives of the vehicle through the process of conformance checking [303–306].

Fig. 7.2 illustrates preliminary results to ensure the drivability of fail-safe trajectories using motion primitives [115] that were obtained in the supervised thesis [25]. We used the data of the conducted test drives to determine the bounded disturbance and sensor noise sets. Subsequently, we created a database of 16,000 motion primitives, each with a time horizon of 1 s, using the CORA [266] and CVX [307] toolboxes. Fig. 7.2a shows the matched trajectory of an executed braking maneu-

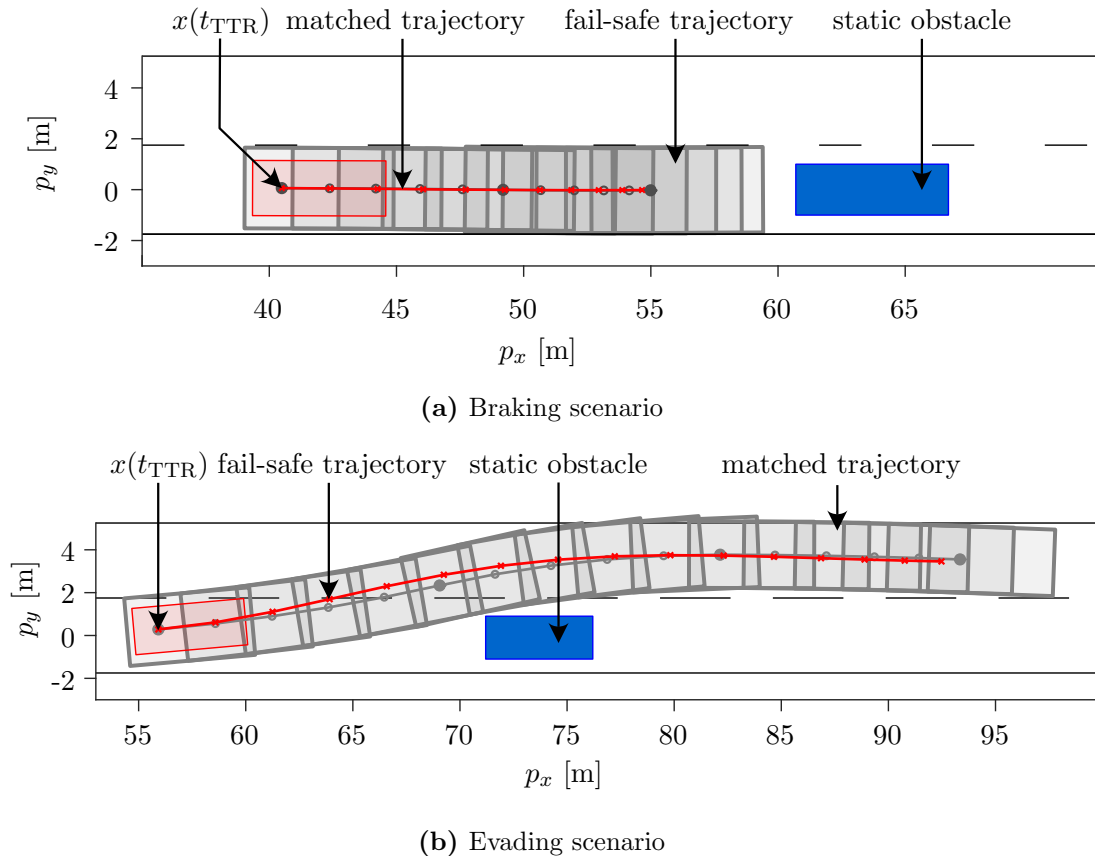


Figure 7.2: Fitting fail-safe trajectories with motion primitives. Motion primitives can be used to ensure the drivability of fail-safe trajectories (red line) despite controller inaccuracies. The fitted trajectory (gray line) and the corresponding occupancies (gray areas) considering possible uncertainties allow one to detect possible collisions if the vehicle cannot perfectly track the fail-safe trajectory. a) Results of fitting a fail-safe trajectory involving braking with two motion primitives. b) Results of fitting a fail-safe trajectory involving evading with three motion primitives. All occupancies of the ego vehicle are shown with respect to the rear axle. The red rectangle shows the occupancy of the ego vehicle at $x(t_{\text{TTR}})$. Each motion primitive is delimited by two filled gray circles.

ver in a scenario with a static obstacle. The resulting drivable trajectory remains collision-free. Fig. 7.2b illustrates the results for an executed evasive maneuver. Although the first results are promising, a large number of motion primitives are usually required to match a given fail-safe trajectory with minor deviations. Moreover, the resulting occupancy of the autonomous vehicle may violate certain constraints, such as leaving the road boundary, as shown in Fig. 7.2b.

7.3.3 Further improving the verification performance

Following our successful experiments on a fenced test track and real-world data, the next steps should include closed-loop driving in urban traffic. To further reduce the false positive rate of our safety layer, additional traffic rules can be formalized to obtain tighter over-approximative occupancy sets. Furthermore, suitable models for interactions between traffic participants can improve the performance in certain traffic situations (e.g., when a vehicle in a neighboring lane creates a gap for the autonomous vehicle to merge into). In addition, the fail-safe motion planning technique can be extended to communicating autonomous vehicles to plan cooperative fail-safe maneuvers. Our fail-safe motion planning approach can also be modified to work with other types of robotic systems, such as mobile robots in production plants or delivery robots.

Invariably safe sets have proven to provide significant benefits for safe motion planning of autonomous vehicles. Our proposed under-approximation is already tight, but additional computation time can be used to further tighten it. Moreover, additional traffic rules and objects can be integrated into the invariably safe set computation. In our current prototype, we are already able to integrate traffic lights and zebra crossings. We model these elements as time-variant obstacles with occupancy predictions. This model allows us to incorporate such elements without changing the interface, since the elements are part of the occupancy set input. For instance, a red traffic light corresponds to a static obstacle whereas a green traffic light is modeled by an empty occupancy set. By integrating additional traffic rules for motion planning, invariably safe sets can be extended to contain only legal safe states (i.e., states that are safe and comply with traffic rules). As a result, invariably safe sets can be used to detect traffic rule violations by the autonomous vehicle.

Although the proposed verification technique is built around legal safety, we also aim to guarantee safety in light of traffic participants disobeying certain traffic rules. Due to the unlawful behavior of other traffic participants, the previously computed fail-safe trajectory can be rendered unsafe. The set-based prediction already provides a mechanism to account for violations of traffic rules by continuously monitoring whether obstacles abide by the traffic rules. If it is detected that an obstacle violates a certain traffic rule, this rule is automatically overridden for that traffic participant. In future work, our fail-safe motion planning approach can be optimized to determine maneuvers that mitigate potential collisions. To boost the performance of our approach in such situations, the reactive constraint man-

agement could be extended to a predictive constraint management that foresees whether a traffic participant is likely to disobey a traffic rule in the future.

7.4 Closing Remarks

Autonomous robots will inevitably become an important part of our everyday lives: drones will deliver parcels to our front door, autonomous vehicles will drive our children to school, and humanoid robots will do our household duties or support elderly care. However, with the great power of these systems comes great responsibility for developers and researchers. These new types of safety-critical, autonomous, and learning systems must provide the highest possible levels of safety before they are deployed in human-centered environments. Ensuring the safety of these systems must be an integral part of each development stage, from design until deployment.

In this thesis, we established the first online verification technique that is able to provide strong safety guarantees for a complex robotic system operating in highly uncertain environments. Our preliminary results indicate that the usage of fail-safe motion planning can drastically reduce the number of traffic accidents, supporting the goal of achieving a future with zero traffic accidents. The author encourages developers and researchers to adopt and advance verification techniques in their robotic systems. The lives of humans should never be at risk in favor of higher performance, faster deployment, or stronger market penetration.

8 Publications

Journal papers

- [1] B. Gutjahr, C. Pek, L. Gröll, and M. Werling. Efficient trajectory optimization for vehicles using quadratic programming. *Automatisierungstechnik*, 64(10):786–794, 2016.
- [2] C. Pek, S. Manzinger, M. Koschi, and M. Althoff. Using online verification to prevent autonomous vehicles from causing accidents. *Nature Machine Intelligence*, 2020. In press.
- [3] C. Pek and M. Althoff. Fail-safe motion planning for online verification of autonomous vehicles using convex optimization. *IEEE Transactions on Robotics*, 2020. In review.
- [4] S. Manzinger, C. Pek, and M. Althoff. Using reachable sets for trajectory planning of automated vehicles. *IEEE Transactions on Intelligent Vehicles*, 2020. In press.
- [5] M. Althoff, S. Maierhofer, and C. Pek. Provably-correct and comfortable adaptive cruise control. *IEEE Transactions on Intelligent Vehicles*, 2020. In press.

Conference papers

- [6] C. Pek, A. Muxfeldt, and D. Kubus. Simplifying synchronization in cooperative robot tasks—an enhancement of the manipulation primitive paradigm. In *Proc. of the IEEE Int. Conf. on Emerging Technologies and Factory Automation*, pages 1–8, 2016.
- [7] C. Pek, P. Zahn, and M. Althoff. Verifying the safety of lane change maneuvers of self-driving vehicles based on formalized traffic rules. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 1477–1483, 2017.
- [8] C. Pek, M. Koschi, M. Werling, and M. Althoff. Enhancing motion safety by identifying safety-critical passageways. In *Proc. of the IEEE Conf. on Decision and Control*, pages 320–326, 2017.
- [9] C. Miller, C. Pek, and M. Althoff. Efficient mixed-integer programming for longitudinal and lateral motion planning of autonomous vehicles. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 1954–1961, 2018.

- [10] C. Pek and M. Althoff. Efficient computation of invariably safe states for motion planning of self-driving vehicles. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, pages 3523 – 3530, 2018.
- [11] C. Pek and M. Althoff. Computationally efficient fail-safe trajectory planning for self-driving vehicles using convex optimization. In *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, pages 1447–1454, 2018.
- [12] M. Koschi, C. Pek, M. Beikirch, and M. Althoff. Set-based prediction of pedestrians in urban environments considering formalized traffic rules. In *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, pages 2704–2711, 2018.
- [13] 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 *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, pages 2156–2162, 2018.
- [14] 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*, pages 260–274, 2019.
- [15] M. Koschi, C. Pek, S. Maierhofer, and M. Althoff. Computationally efficient safety falsification of adaptive cruise control systems. In *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, pages 2879–2886, 2019.
- [16] C. Pek and M. Althoff. Ensuring motion safety of autonomous vehicles through online fail-safe verification. In *Robotics: Science and Systems – Pioneers Workshop*, pages 1–3, 2019.

Patents

- [17] C. Pek and M. Althoff. Determining the safety of lane change maneuvers, 2016.
- [18] C. Pek. Dynamic adjustment of safety-critical assistance systems of self-driving vehicles based on statistical wear, 2017.
- [19] C. Pek and M. Althoff. Verifying the safety of lane change maneuvers of self-driving vehicles based on formalized traffic rules, 2017.
- [20] C. Pek, M. Koschi, and M. Althoff. Enhancing motion safety by identifying passageways using safe invariant sets, 2017.

Supervised theses

- [21] M. Beikirch. *Collision avoidance in urban environments using set-based prediction*. Master's thesis, Technische Universität München, München, 2017.
- [22] C. Miller. *Efficient mixed-integer planning for longitudinal and lateral control of autonomous vehicles*. Master's thesis, Technische Universität München, München, 2017.
- [23] L. Tappe. *Computation of invariably safe sets for autonomous vehicles along arbitrary road networks*. Bachelor's thesis, Technische Universität München, München, 2018.
- [24] J. Kabalar. *Evaluation of variational fail-safe trajectory planning for self-driving vehicles using the CommonRoad benchmark suite*. Bachelor's thesis, Technische Universität München, München, 2018.
- [25] A.-K. Rettinger. *Ensuring drivability of fail-safe trajectories for autonomous vehicles using set-based control techniques*. Master's thesis, Technische Universität München, München, 2018.
- [26] S. Maierhofer. *Enhancement and evaluation of a novel adaptive cruise control system with formal guarantees in a real driving simulator*. Master's thesis, Technische Universität München, München, 2018.
- [27] S. Kaster. *Online prediction of vehicles and pedestrians for guaranteed motion safety of autonomous vehicles*. Master's thesis, Technische Universität München, München, 2019.
- [28] M. Althaus. *Consideration of safe distances in online verification for motion planning of autonomous vehicles*. Master's thesis, Technische Universität München, München, 2019.
- [29] M. Both. *Cross-modal learning for gripping area detection*. Master's thesis, Technische Universität München, München, 2019.

Bibliography

- [30] C. Urmson, C. Baker, J. Dolan, P. Rybski, B. Salesky, W. Whittaker, D. Ferguson, and M. Darms. Autonomous driving in traffic: Boss and the Urban challenge. *AI Magazine*, 30(2):17–28, 2009.
- [31] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, and C. G. Keller. Making Bertha drive - An autonomous journey on a historic route. *IEEE Intelligent Transportation Systems Magazine*, 6(2):8–20, 2014.
- [32] M. Aeberhard, S. Rauch, M. Bahram, G. Tanzmeister, J. Thomas, Y. Pilat, F. Homm, W. Huber, and N. Kaempchen. Experience, results and lessons learned from automated driving on Germany’s highways. *IEEE Intelligent Transportation Systems Magazine*, 7(1):42–57, 2015.
- [33] P. Furgale, U. Schwesinger, M. Ruffi, W. Derendarz, H. Grimmer, P. Mühlfellner, S. Wonneberger, J. Timpner, S. Rottmann, B. Li, B. Schmidt, T. N. Nguyen, E. Cardarelli, S. Cattani, S. Brüning, S. Horstmann, M. Stellmacher, H. Mielenz, K. Köser, M. Beermann, C. Häne, L. Heng, G. H. Lee, F. Fraundorfer, R. Iser, R. Triebel, I. Posner, P. Newman, L. Wolf, M. Pollefeys, S. Brosig, J. Effertz, C. Pradalier, and R. Siegwart. Toward automated driving in cities using close-to-market sensors: An overview of the V-Charge project. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 809–816, 2013.
- [34] D. C. Conner, H. Kress-Gazit, H. Choset, A. A. Rizzi, and G. J. Pappas. Valet parking without a valet. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, pages 572–577, 2007.
- [35] J. M. Anderson, K. Nidhi, K. D. Stanley, P. Sorensen, C. Samaras, and O. A. Oluwatola. *Autonomous vehicle technology: A guide for policymakers*. Rand Corporation, 2014.
- [36] F. Favaró, S. Eurich, and N. Nader. Autonomous vehicles’ disengagements: Trends, triggers, and regulatory limitations. *Accident analysis and prevention*, 110:136–148, 2017.
- [37] S. Wang and Z. Li. Exploring the mechanism of crashes with automated vehicles using statistical modeling approaches. *PLOS ONE*, 14(3):1–16, 2019.

Bibliography

- [38] S. Shalev-Shwartz, S. Shammah, and A. Shashua. On a formal model of safe and scalable self-driving cars. *arXiv:1708.06374 [cs.RO]*, pages 1–37, 2017.
- [39] Economic Commission for Europe: Inland Transport Committee. Vienna Convention on Road Traffic, November 1968. URL: <http://www.unece.org/fileadmin/DAM/trans/conventn/crt1968e.pdf>.
- [40] P. Koopman and M. Wagner. Challenges in autonomous vehicle testing and validation. *SAE Int. Journal of Transportation Safety*, 4(1):15–24, 2016.
- [41] P. Koopman and M. Wagner. Autonomous vehicle safety: An interdisciplinary challenge. *IEEE Intelligent Transportation Systems Magazine*, 9(1):90–96, 2017.
- [42] N. Kalra and S. M. Paddock. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94:182–193, 2016.
- [43] M. Maurer, J. C. Gerdes, B. Lenz, and H. Winner. *Autonomous driving—Technical, Legal and Social Aspects*. Springer Berlin Heidelberg, 2016.
- [44] J. Dahl, G. R. de Campos, C. Olsson, and J. Fredriksson. Collision avoidance: A literature review on threat-assessment techniques. *IEEE Transactions on Intelligent Vehicles*, 4(1):101–113, 2019.
- [45] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How. Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology*, 17(5):1105–1118, 2009.
- [46] M. Werling, S. Kammel, J. Ziegler, and L. Groll. Optimal trajectories for time-critical street scenarios using discretized terminal manifolds. *The Int. Journal of Robotics Research*, 31(3):346–359, 2012.
- [47] S. Lefèvre, D. Vasquez, and C. Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH Journal*, 1(1):1–14, 2014.
- [48] T. Gindele, S. Brechtel, and R. Dillmann. A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments. In *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, pages 1625–1631, 2010.
- [49] A. Lawitzky, D. Althoff, C. F. Passenberg, G. Tanzmeister, D. Wollherr, and M. Buss. Interactive scene prediction for automotive applications. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 1028–1033, 2013.

- [50] S. Klingelschmitt, M. Platho, H. Groß, V. Willert, and J. Eggert. Combining behavior and situation information for reliably estimating multiple intentions. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 388–393, 2014.
- [51] J. Schulz, K. Hirsenkorn, J. Löchner, M. Werling, and D. Burschka. Estimation of collective maneuvers through cooperative multi-agent planning. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 624–631, 2017.
- [52] M. Bahram, C. Hubmann, A. Lawitzky, M. Aeberhard, and D. Wollherr. A combined model- and learning-based framework for interaction-aware maneuver prediction. *IEEE Transactions on Intelligent Transportation Systems*, 17(6):1538–1550, 2016.
- [53] T. Rehder, A. Koenig, M. Goehl, L. Louis, and D. Schramm. Lane change intention awareness for assisted and automated driving on highways. *IEEE Transactions on Intelligent Vehicles*, 4(2):265–276, 2019.
- [54] B. T. Morris and M. M. Trivedi. Learning, modeling, and classification of vehicle track patterns from live video. *IEEE Transactions on Intelligent Transportation Systems*, 9(3):425–437, 2008.
- [55] P. Kumar, M. Perrollaz, S. Lefèvre, and C. Laugier. Learning-based approach for online lane change intention prediction. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 797–802, 2013.
- [56] M. I. Jordan and T. M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.
- [57] S. Sivaraman and M. M. Trivedi. A general active-learning framework for on-road vehicle recognition and tracking. *IEEE Transactions on Intelligent Transportation Systems*, 11(2):267–276, 2010.
- [58] U. Dogan, J. Edelbrunner, and I. Iossifidis. Autonomous driving: A comparison of machine learning techniques by means of the prediction of lane change behavior. In *Proc. of the IEEE Int. Conf. on Robotics and Biomimetics*, pages 1837–1843, 2011.
- [59] Z. Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452, 2015.
- [60] S. Kammel, J. Ziegler, B. Pitzer, M. Werling, T. Gindele, D. Jagzent, J. Schröder, M. Thuy, M. Goebel, F. v. Hundelshausen, O. Pink, C. Frese, and C. Stiller. Team AnnieWAY’s autonomous system for the 2007 DARPA Urban Challenge. *Journal of Field Robotics*, 25(9):615–639, 2008.
- [61] X. Li, Z. Sun, D. Cao, Z. He, and Q. Zhu. Real-time trajectory planning for autonomous urban driving: Framework, algorithms, and verifications. *IEEE Transactions on Mechatronics*, 21(2):740–753, 2016.

- [62] W. Xu, J. Pan, J. Wei, and J. M. Dolan. Motion planning under uncertainty for on-road autonomous driving. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 2507–2512, 2014.
- [63] J. Nilsson, M. Ali, P. Falcone, and J. Sjöberg. Predictive manoeuvre generation for automated driving. In *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, pages 418–423, 2013.
- [64] C. Schmidt, F. Oechsle, and W. Branz. Research on trajectory planning in emergency situations with multiple objects. In *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, pages 988–992, 2006.
- [65] M. Werling and D. Liccardo. Automatic collision avoidance using model-predictive online optimization. In *Proc. of the IEEE Int. Conf. on Decision and Control*, pages 6309–6314, 2012.
- [66] J. Hu, J. Lygeros, M. Prandini, and S. Sastry. A probabilistic framework for highway safety analysis. In *Proc. of the IEEE Int. Conf. on Decision and Control*, pages 3734–3739, 1999.
- [67] M. Althoff, O. Stursberg, and M. Buss. Model-based probabilistic collision detection in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 10(2):299–310, 2009.
- [68] C. Laugier, I. E. Paromtchik, M. Perrollaz, M. Yong, J. 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*, 3(4):4–19, 2011.
- [69] S. Patil, J. van den Berg, and R. Alterovitz. Estimating probability of collision for safe motion planning under gaussian motion and sensing uncertainty. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 3238–3244, 2012.
- [70] G. S. Aoude, B. D. Luders, J. M. Joseph, N. Roy, and J. P. How. Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns. *Autonomous Robots*, 35(1):51–76, 2013.
- [71] A. Broadhurst, S. Baker, and T. Kanade. Monte carlo road safety reasoning. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 319–324, 2005.
- [72] A. Eidehall and L. Petersson. Statistical threat assessment for general road scenes using monte carlo sampling. *IEEE Transactions on Intelligent Transportation Systems*, 9(1):137–147, 2008.
- [73] M. Althoff and A. Mergel. Comparison of Markov chain abstraction and monte carlo simulation for the safety assessment of autonomous cars. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1237–1247, 2011.

- [74] E. Plaku, L. E. Kavradi, and M. Y. Vardi. Falsification of LTL safety properties in hybrid systems. *Int. Journal on Software Tools for Technology Transfer*, 15(4):305–320, 2013.
- [75] T. Dreossi, A. Donzé, and S. A. Seshia. Compositional falsification of cyber-physical systems with machine learning components. In *NASA Formal Methods Symposium*, pages 357–372, 2017.
- [76] H. Abbas, G. Fainekos, S. Sankaranarayanan, F. Ivačić, and A. Gupta. Probabilistic temporal logic falsification of cyber-physical systems. *ACM Transactions on Embedded Computing Systems*, 12(2):1–30, 2013.
- [77] A. Bhatia and E. Frazzoli. Incremental search methods for reachability analysis of continuous and hybrid systems. In *Proc. of the Int. Conf. on Hybrid Systems: Computation and Control*, LNCS 2993, pages 142–156. Springer, 2004.
- [78] T. Dreossi, T. Dang, A. Donzé, J. Kapinski, X. Jin, and J. V. Deshmukh. Efficient guiding strategies for testing of temporal properties of hybrid systems. In *Proc. of the NASA Formal Methods Symposium*, pages 127–142, 2015.
- [79] Z. Yang, X. Wang, X. Pei, S. Feng, D. Wang, J. Wang, and S. C. Wong. Longitudinal safety analysis for heterogeneous platoon of automated and human vehicles. In *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, pages 3300–3305, 2018.
- [80] M. Althoff and S. Lutz. Automatic generation of safety-critical test scenarios for collision avoidance of road vehicles. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 1326–1333, 2018.
- [81] I. R. Jenkins, L. O. Gee, A. Knauss, H. Yin, and J. Schroeder. Accident scenario generation with recurrent neural networks. In *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, pages 3340–3345, 2018.
- [82] C. E. Tuncali, G. Fainekos, H. Ito, and J. Kapinski. Simulation-based adversarial test generation for autonomous vehicles with machine learning components. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 1555–1562, 2018.
- [83] G. E. Mullins, P. G. Stankiewicz, and S. K. Gupta. Automated generation of diverse and challenging scenarios for test and evaluation of autonomous vehicles. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1443–1450, 2017.
- [84] M. O’Kelly, A. Sinha, H. Namkoong, R. Tedrake, and J. C. Duchi. Scalable end-to-end autonomous vehicle testing via rare-event simulation. In *Advances in Neural Information Processing Systems*, pages 9827–9838, 2018.

Bibliography

- [85] A. Platzer. Verification of cyberphysical transportation systems. *IEEE Intelligent Systems*, 24(4):10–13, 2009.
- [86] R. Kianfar, P. Falcone, and J. Fredriksson. Safety verification of automated driving systems. *IEEE Intelligent Transportation Systems Magazine*, 5(4):73–86, 2013.
- [87] H. Roehm, J. Oehlerking, M. Woehrle, and M. Althoff. Model conformance for cyber-physical systems: A survey. *ACM Trans. Cyber-Phys. Syst.*, 3(3):1–26, 2019.
- [88] A. Platzer. *Logical foundations of cyber-physical systems*. Springer.
- [89] E. M. Clarke, O. Grumberg, and D. Peled. *Model checking*. MIT press, 1999.
- [90] S. M. Veres, L. Molnar, N. K. Lincoln, and C. P. Morice. Autonomous vehicle control systems—A review of decision making. *Proc. of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 225(2):155–195, 2011.
- [91] M. Kamali, L. A. Dennis, O. McAree, M. Fisher, and S. M. Veres. Formal verification of autonomous vehicle platooning. *Science of Computer Programming*, 148:88–106, 2017.
- [92] H. Roehm, J. Oehlerking, T. Heinz, and M. Althoff. STL model checking of continuous and hybrid systems. In *Automated Technology for Verification and Analysis*, pages 412–427, 2016.
- [93] S. Berchtold, C. Böhm, and H.-P. Kriegel. The pyramid-technique: Towards breaking the curse of dimensionality. In *ACM SIGMOD Record*, volume 27, pages 142–153, 1998.
- [94] F. Bach. Breaking the curse of dimensionality with convex neural networks. *The Journal of Machine Learning Research*, 18(1):629–681, 2017.
- [95] T. Nipkow, L. C. Paulson, and M. Wenzel. *Isabelle/HOL: A proof assistant for higher-order logic*, volume 2283. Springer Science & Business Media, 2002.
- [96] W. Damm, H.-J. Peter, J. Rakow, and B. Westphal. Can we build it: formal synthesis of control strategies for cooperative driver assistance systems. *Mathematical Structures in Computer Science*, 23(04):676–725, 2013.
- [97] M. Hilscher, S. Linker, and E.-R. Olderog. Proving safety of traffic manoeuvres on country roads. In *Theories of Programming and Formal Methods*, pages 196–212. Springer, 2013.
- [98] S. M. Loos, A. Platzer, and L. Nistor. Adaptive cruise control: Hybrid, distributed, and now formally verified. In *Proc. of the Int. Symposium on Formal Methods*, pages 42–56, 2011.

- [99] S. Mitsch, S. M. Loos, and A. Platzer. Towards formal verification of freeway traffic control. In *Proc. of the IEEE Int. Conf. on Cyber-Physical Systems*, pages 171–180, 2012.
- [100] A. Rizaldi, F. Immler, B. Schürmann, and M. Althoff. A formally verified motion planner for autonomous vehicles. In *Automated Technology for Verification and Analysis*, pages 75–90, 2018.
- [101] S. L. Smith, J. Tůmová, C. Belta, and D. Rus. Optimal path planning under temporal logic constraints. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, pages 3288–3293, 2010.
- [102] T. Wongpiromsarn, S. Karaman, and E. Frazzoli. Synthesis of provably correct controllers for autonomous vehicles in urban environments. In *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, pages 1168–1173, 2011.
- [103] T. Wongpiromsarn, U. Topcu, and R. M. Murray. Receding horizon temporal logic planning. *IEEE Transactions on Automatic Control*, 57(11):2817–2830, 2012.
- [104] J. Tůmová and D. V. Dimarogonas. A receding horizon approach to multi-agent planning from local LTL specifications. In *Proc. of the American Control Conference*, pages 1775–1780, 2014.
- [105] I. Saha, R. Ramaiithima, V. Kumar, G. J. Pappas, and S. A. Seshia. Automated composition of motion primitives for multi-robot systems from safe LTL specifications. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, pages 1525–1532, 2014.
- [106] J. Tůmová, G. C. Hall, S. Karaman, E. Frazzoli, and D. Rus. Least-violating control strategy synthesis with safety rules. In *Proc. of the Int. Conf. on Hybrid Systems: Computation and Control*, pages 1–10, 2013.
- [107] D. Sadigh and A. Kapoor. Safe control under uncertainty with probabilistic signal temporal logic. In *Proc. of Robotics: Science and Systems*, pages 1–10, 2016.
- [108] S. Jha, V. Raman, D. Sadigh, and S. A. Seshia. Safe autonomy under perception uncertainty using chance-constrained temporal logic. *Journal of Automated Reasoning*, 60(1):43–62, 2018.
- [109] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames. Robustness of control barrier functions for safety critical control. *IFAC-PapersOnLine*, 48(27):54 – 61, 2015.

- [110] Y. Chen, H. Peng, and J. Grizzle. Obstacle avoidance for low-speed autonomous vehicles with barrier function. *IEEE Transactions on Control Systems Technology*, 26(1):194–206, 2018.
- [111] O. Maler. Computing reachable sets: An introduction, 2008.
- [112] P. Falcone, M. Ali, and J. Sjöberg. Predictive threat assessment via reachability analysis and set invariance theory. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1352–1361, 2011.
- [113] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin. FaS-Track: a modular framework for fast and guaranteed safe motion planning. In *Proc. of the IEEE Int. Conf. on Decision and Control*, pages 1517–1522, 2017.
- [114] S. Vaskov, S. Kousik, H. Larson, F. Bu, J. Ward, S. Worrall, M. Johnson-Roberson, and R. Vasudevan. Towards provably not-at-fault control of autonomous robots in arbitrary dynamic environments. In *Proc. of Robotics: Science and Systems*, pages 1–10, 2019.
- [115] B. Schürmann, D. Heß, J. Eilbrecht, O. Stursberg, F. Köster, and M. Althoff. Ensuring drivability of planned motions using formal methods. In *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, pages 1661–1668, 2017.
- [116] M. Althoff and S. Magdici. Set-based prediction of traffic participants on arbitrary road networks. *IEEE Transactions on Intelligent Vehicles*, 1(2):187–202, 2016.
- [117] M. Koschi and M. Althoff. SPOT: A tool for set-based prediction of traffic participants. In *Proc. of the IEEE Intelligent Vehicles Symposium*, page 1686–1693, 2017.
- [118] M. Koschi, C. Pek, M. Beikirch, and M. Althoff. Set-based prediction of pedestrians in urban environments considering formalized traffic rules. In *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, pages 2704–2711, 2018.
- [119] P. F. Orzechowski, A. Meyer, and M. Lauer. Tackling occlusions limited sensor range with set-based safety verification. In *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, pages 1729–1736, 2018.
- [120] M. Althoff and J. M. Dolan. Online verification of automated road vehicles using reachability analysis. *IEEE Transactions on Robotics*, 30(4):903–918, 2014.

- [121] P. Bender, J. Ziegler, and C. Stiller. Lanelets: Efficient map representation for autonomous driving. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 420–425, 2014.
- [122] J. Ziegler, P. Bender, T. Dang, and C. Stiller. Trajectory planning for Bertha - A local, continuous method. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 450–457, 2014.
- [123] E. Héry, S. Masi, P. Xu, and P. Bonnifait. Map-based curvilinear coordinates for autonomous vehicles. In *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, pages 1–7, 2017.
- [124] S. Steyer, G. Tanzmeister, and D. Wollherr. Grid-based environment estimation using evidential mapping and particle tracking. *IEEE Transactions on Intelligent Vehicles*, 3(3):384–396, 2018.
- [125] M. Althoff, M. Koschi, and S. Manzinger. CommonRoad: Composable benchmarks for motion planning on roads. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 719–726, 2017.
- [126] I. M. Mitchell and C. J. Tomlin. Overapproximating reachable sets by Hamilton-Jacobi projections. *Journal of Scientific Computing*, 19(1):323–346, 2003.
- [127] E. Asarin, T. Dang, G. Frehse, A. Girard, C. Le Guernic, and O. Maler. Recent progress in continuous and hybrid reachability analysis. In *Proc. of the IEEE Int. Conf. on Computer Aided Control System Design*, pages 1582–1587, 2006.
- [128] M. Althoff. *Reachability analysis and its application to the safety assessment of autonomous cars*. Dissertation, Technische Universität München, München, 2010.
- [129] M. Althoff and J. M. Dolan. Reachability computation of low-order models for the safety verification of high-order road vehicle models. In *Proc. of the American Control Conference*, pages 3559–3566, 2012.
- [130] I. M. Mitchell. Comparing forward and backward reachability as tools for safety analysis. In *Proc. of the Int. Conf. on Hybrid Systems: Computation and Control*, pages 428–443. Springer, 2007.
- [131] Safety of machinery – Positioning of safeguards with respect to the approach speeds of parts of the human body (ISO 13855:2010), 2010.
- [132] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, UK; New York, 2004.

- [133] D. P. Bertsekas. *Convex optimization algorithms*. Athena Scientific Belmont, 2015.
- [134] S. Magdici and M. Althoff. Fail-safe motion planning of autonomous vehicles. In *Proc. of the IEEE. Int. Conf. on Intelligent Transportation Systems*, pages 452–458, 2016.
- [135] B. Paden, M. Čáp, 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*, 1(1):33–55, 2016.
- [136] D. Gonzalez, J. Perez, V. Milanés, and F. Nashashibi. A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1135–1145, 2016.
- [137] W. Schwarting, J. Alonso-Mora, and D. Rus. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):187–210, 2018.
- [138] 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*, 60:416 – 442, 2015.
- [139] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser. A review of motion planning for highway autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–23, 2019. early access.
- [140] T.-C. Au, C.-L. Fok, S. Vishwanath, C. Julien, and P. Stone. Evasion planning for autonomous vehicles at intersections. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, pages 1541–1546, 2012.
- [141] K. Hirsch, J. Hilgert, W. Lalo, D. Schramm, and M. Hiller. Optimization of emergency trajectories for autonomous vehicles with respect to linear vehicle dynamics. In *Proc. of the IEEE Int. Conf. on Advanced Intelligent Mechatronics*, pages 528–533, 2005.
- [142] C. Ackermann, J. Bechtloff, and R. Isermann. Collision avoidance with combined braking and steering. *6th Int. Munich Chassis Symposium*, pages 199–213, 2015.
- [143] S. Bouraine, T. Fraichard, and H. Salhi. Provably safe navigation for mobile robots with limited field-of-views in dynamic environments. In *Proc. of the IEEE Int. Conference on Robotics and Automation*, pages 174–179, 2012.
- [144] K. Macek, D. Vasquez, T. Fraichard, and R. Siegwart. Towards safe vehicle navigation in dynamic urban scenarios. *Automatika*, 50(3-4):184–194, 2009.

- [145] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock. Randomized kinodynamic motion planning with moving obstacles. *Int. Journal of Robotics Research*, 21(3):233–255, 2002.
- [146] M. Seder and I. Petrovic. Dynamic window based approach to mobile robot motion control in the presence of moving obstacles. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1986–1991, 2007.
- [147] S. Kousik, S. Vaskov, M. Johnson-Roberson, and R. Vasudevan. Safe trajectory synthesis for autonomous driving in unforeseen environments. *CoRR*, abs/1705.00091:1–9, 2017.
- [148] H. Xu, Y. Gao, F. Yu, and T. Darrell. End-to-end learning of driving models from large-scale video datasets. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3530–3538, 2017.
- [149] S. Shalev-Shwartz, S. Shammah, and A. Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, pages 1–13, 2016.
- [150] M. Kuderer, S. Gulati, and W. Burgard. Learning driving styles for autonomous vehicles from demonstration. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 2641–2646, 2015.
- [151] T. Kollar and N. Roy. Trajectory optimization using reinforcement learning for map exploration. *The Int. Journal of Robotics Research*, 27(2):175–196, 2008.
- [152] M. Riedmiller, M. Montemerlo, and H. Dahlkamp. Learning to drive a real car in 20 minutes. In *Proc. of the IEEE Int. Conf. on Frontiers in the Convergence of Bioscience and Information Technologies*, pages 645–650, 2007.
- [153] C. Desjardins and B. Chaib-draa. Cooperative adaptive cruise control: A reinforcement learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1248–1260, 2011.
- [154] X. Ma, K. Driggs-Campbell, and M. J. Kochenderfer. Improved robustness and safety for autonomous vehicle control with adversarial reinforcement learning. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 1665–1671, 2018.
- [155] M. Mukadam, A. Cosgun, A. Nakhaei, and K. Fujimura. Tactical decision making for lane changing with deep reinforcement learning. In *NIPS Workshop on Machine Learning for Intelligent Transportation Systems*, 2017.

- [156] P. Wolf, K. Kurzer, T. Wingert, F. Kuhnt, and J. M. Zöllner. Adaptive behavior generation for autonomous driving using deep reinforcement learning with compact semantic states. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 993–1000, 2018.
- [157] D. A. Pomerleau. Alvin: An autonomous land vehicle in a neural network. In *Advances in neural information processing systems*, pages 305–313, 1989.
- [158] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, pages 1–9, 2016.
- [159] S. A. Seshia, D. Sadigh, and S. S. Sastry. Towards verified artificial intelligence. *CoRR*, abs/1606.08514:1–11, 2016.
- [160] D. Heß, M. Althoff, and T. Sattel. Formal verification of maneuver automata for parameterized motion primitives. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, pages 1474–1481, 2014.
- [161] D. J. Grymin, C. B. Neas, and M. Farhood. A hierarchical approach for primitive-based motion planning and control of autonomous vehicles. *Robotics and Autonomous Systems*, 62(2):214–228, 2014.
- [162] J. H. Gillula, H. Huang, M. P. Vitus, and C. J. Tomlin. Design of guaranteed safe maneuvers using reachable sets: Autonomous quadrotor aerobatics in theory and practice. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1649–1654, 2010.
- [163] A. Majumdar and R. Tedrake. Robust online motion planning with regions of finite time invariance. In *Algorithmic Foundations of Robotics X*, pages 543–558. 2013.
- [164] S. Singh, A. Majumdar, J.-J. Slotine, and M. Pavone. Robust online motion planning via contraction theory and convex optimization. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 5883–5890, 2017.
- [165] S. M. LaValle. *Planning algorithms*. Cambridge university press, 2006.
- [166] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. *Int. Journal of Robotics Research*, 20(5):378–400, 2001.
- [167] E. Frazzoli, M. A. Dahleh, and E. Feron. Real-time motion planning for agile autonomous vehicles. In *Proc. of the American Control Conference*, pages 43–49, 2001.
- [168] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *Int. Journal of Robotics Research*, 30(7):846–894, 2011.

- [169] F. von Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller, and H.-J. Wuensche. Driving with tentacles: Integral structures for sensing and motion. *Int. Journal of Field Robotics*, 25(9):640–673, September 2008.
- [170] J. Ziegler and C. Stiller. Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios. In *Proc. of the IEEE Int. Conf. on Intelligent Systems and Robots*, pages 1879–1884, 2009.
- [171] M. McNaughton, C. Urmson, J. M. Dolan, and J.-W. Lee. Motion planning for autonomous driving with a conformal spatiotemporal lattice. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 4889–4895, 2011.
- [172] M. Pivtoraiko and A. Kelly. Kinodynamic motion planning with state lattice motion primitives. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, pages 2172–2179, 2011.
- [173] M. Likhachev and D. Ferguson. Planning long dynamically feasible maneuvers for autonomous vehicles. *The Int. Journal of Robotics Research*, 28(8):933–945, 2009.
- [174] Z. Ajanovic, B. Lacevic, B. Shyrokau, M. Stolz, and M. Horn. Search-based optimal motion planning for automated driving. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, pages 4523–4530, 2018.
- [175] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa. CHOMP: Gradient optimization techniques for efficient motion planning. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 489–494, 2009.
- [176] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun. Towards fully autonomous driving: Systems and algorithms. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 163–168, 2011.
- [177] D. Berenson, J. Kuffner, and H. Choset. An optimization approach to planning for mobile manipulation. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1187–1192, 2008.
- [178] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.
- [179] T. Schouwenaars, B. De Moor, E. Feron, and J. How. Mixed integer programming for multi-vehicle path planning. In *Proc. of the IEEE European Control Conference*, pages 2603–2608, 2001.
- [180] Y. Du, Y. Wang, and C. Chan. Autonomous lane-change controller via mixed logical dynamical. In *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, pages 1154–1159, 2014.

Bibliography

- [181] F. Molinari, Nguyen Ngoc Anh, and L. Del Re. Efficient mixed integer programming for autonomous overtaking. In *Proc. of the American Control Conference*, pages 2303–2308, 2017.
- [182] A. Richards and J. P. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *Proc. of the American Control Conference*, pages 1936–1941, 2002.
- [183] R. Deits and R. Tedrake. Efficient mixed-integer planning for uavs in cluttered environments. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 42–49, 2015.
- [184] B. Landry, R. Deits, P. R. Florence, and R. Tedrake. Aggressive quadrotor flight through cluttered environments using mixed integer programming. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1469–1475, 2016.
- [185] B. Yi, S. Gottschling, J. Ferdinand, N. Simm, F. Bonarens, and C. Stiller. Real-time integrated vehicle dynamics control and trajectory planning with MPC for critical maneuvers. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 584–589, 2016.
- [186] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts. LQR-trees: Feedback motion planning via sums-of-squares verification. *The Int. Journal of Robotics Research*, 29(8):1038–1052, 2010.
- [187] L. Hewing, A. Liniger, and M. N. Zeilinger. Cautious NMPC with gaussian process dynamics for autonomous miniature race cars. In *Proc. of the European Control Conference*, pages 1341–1348, 2018.
- [188] D. Bertsekas. *Nonlinear Programming*. Athena scientific optimization and computation series. Athena Scientific, 2016.
- [189] P. Falcone, M. Tufo, F. Borrelli, J. Asgari, and H. E. Tseng. A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems. In *Proc. of the IEEE Int. Conf. on Decision and Control*, pages 2980–2985, 2007.
- [190] L. Gurobi Optimization. Gurobi optimizer reference manual, 2019. URL: <http://www.gurobi.com>.
- [191] A. Domahidi, E. Chu, and S. Boyd. ECOS: An SOCP solver for embedded systems. In *Proc. of the IEEE European Control Conference*, pages 3071–3076, 2013.

- [192] S. J. Anderson and S. C. Peters. An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios. *Int. Journal of Vehicle Autonomous Systems*, 8:190–216, 2010.
- [193] W. Zhan, J. Chen, C.-Y. Chan, C. Liu, and M. Tomizuka. Spatially-partitioned environmental representation and planning architecture for on-road autonomous driving. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 632–639, 2017.
- [194] J. Nilsson, M. Brännström, J. Fredriksson, and E. Coelingh. Longitudinal and lateral control for automated yielding maneuvers. *IEEE Transactions on Intelligent Transportation Systems*, 17(5):1404–1414, 2016.
- [195] B. Gutjahr. *Recheneffiziente Trajektorienoptimierung für automatisierte Fahreingriffe*. PhD thesis, Karlsruher Institut für Technologie (KIT), 2019.
- [196] B. Gutjahr, L. Gröll, and M. Werling. Lateral vehicle trajectory optimization using constrained linear time-varying MPC. *IEEE Transactions on Intelligent Transportation Systems*, 18(6):1586–1595, 2017.
- [197] J. Ziegler and C. Stiller. Fast collision checking for intelligent vehicle motion planning. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 518–522, 2010.
- [198] M. Elbanhawi, M. Simic, and R. Jazar. In the passenger seat: investigating ride comfort measures in autonomous cars. *IEEE Intelligent Transportation Systems Magazine*, 7(3):4–17, 2015.
- [199] A. Eckert, B. Hartmann, M. Sevenich, and P. Rieth. Emergency steer & brake assist: A systematic approach for system integration of two complementary driver assistance systems. In *Proc. of the Int. Technical Conf. on Enhanced Safety of Vehicles*, pages 1–9, 2011.
- [200] W. Wachenfeld, P. Junietz, R. Wenzel, and H. Winner. The worst-time-to-collision metric for situation identification. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 729–734, 2016.
- [201] S. Bhattacharya, M. Likhachev, and V. Kumar. Search-based path planning with homotopy class constraints in 3D, 2012.
- [202] S. Bhattacharya and R. Ghrist. Path homotopy invariants and their application to optimal trajectory planning. *Annals of Mathematics and Artificial Intelligence*, pages 1–17.
- [203] T. Gu, J. M. Dolan, and J. Lee. Automated tactical maneuver discovery, reasoning and trajectory planning for autonomous driving. In *Proc. of the IEEE Int. Conf. on Intelligent Systems and Robots*, pages 5474–5480, 2016.

- [204] P. Bender, Ö. S. Tas, J. Ziegler, and C. Stiller. The combinatorial aspect of motion planning: Maneuver variants in structured environments. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 1386–1392, 2015.
- [205] J. Park, S. Karumanchi, and K. Iagnemma. Homotopy-based divide-and-conquer strategy for optimal trajectory planning via mixed-integer programming. *IEEE Transactions on Robotics*, 31(5):1101–1115, 2015.
- [206] J. Schlechtriemen, K. P. Wabersich, and K.-D. Kuhnert. Wiggling through complex traffic: Planning trajectories constrained by predictions. In *IEEE Intelligent Vehicles Symposium*, pages 1293–1300, 2016.
- [207] Z. Sun, D. Hsu, T. Jiang, H. Kurniawati, and J. H. Reif. Narrow passage sampling for probabilistic roadmap planning. *IEEE Transactions on Robotics*, 21(6):1105–1115, 2005.
- [208] H. Mohy-ud Din and A. Muhammad. Detecting narrow passages in configuration spaces via spectra of probabilistic roadmaps. In *Proceedings of the ACM Symposium on Applied Computing*, pages 1294–1298. ACM, 2010.
- [209] Q. H. Do, S. Mita, and K. Yoneda. Narrow passage path planning using fast marching method and support vector machine. In *Proc. of the IEEE Intelligent Vehicles Symposium Proceedings*, pages 630–635, 2014.
- [210] H. Liu, F. Xiao, and C. Wang. A predictive model for narrow passage path planner by using support vector machine in changing environments. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 2991–2996, 2015.
- [211] K. Esterle, P. Hart, J. Bernhard, and A. Knoll. Spatiotemporal motion planning with combinatorial reasoning for autonomous driving. In *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, pages 1053–1060, 2018.
- [212] S. Söntges and M. Althoff. Computing possible driving corridors for automated vehicles. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 160–166, 2017.
- [213] S. Söntges and M. Althoff. Computing the drivable area of autonomous road vehicles in dynamic road scenes. *IEEE Transactions on Intelligent Transportation Systems*, 19(6):1855–1866, 2018.
- [214] H. Ahn, K. Berntorp, and S. D. Cairano. Reachability-based decision making for city driving. In *Proc. of the American Control Conference*, pages 3203–3208, 2018.
- [215] S. Kousik, S. Vaskov, F. Bu, M. Johnson-Roberson, and R. Vasudevan. Bridging the gap between safety and real-time performance in receding-horizon trajectory design for mobile robots. *CoRR*, abs/1809.06746:1–58, 2018.

- [216] M. Gerdtts and I. Xausa. Avoidance trajectories using reachable sets and parametric sensitivity analysis. In *IFIP Conf. on System Modeling and Optimization*, pages 491–500. Springer, 2011.
- [217] A. Shkolnik, M. Walter, and R. Tedrake. Reachability-guided sampling for planning under differential constraints. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 2859–2865, 2009.
- [218] H. Edelsbrunner, J. Van Leeuwen, T. Ottmann, and D. Wood. Computing the connected components of simple rectilinear geometrical objects in d -space. *RAIRO, Informatique théorique*, 18(2):171–183, 1984.
- [219] S. Diamond and S. Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *The Journal of Machine Learning Research*, 17(1):2909–2913, 2016.
- [220] J. Mattingley and S. Boyd. CVXGEN: a code generator for embedded convex optimization. *Optimization and Engineering*, 13(1):1–27, 2012.
- [221] S. Petti and T. Fraichard. Safe motion planning in dynamic environments. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, pages 2210–2215, 2005.
- [222] C. E. Garcia, D. M. Preth, and M. Morari. Model predictive control: Theory and practice. *Automatica*, 25(3):335 – 348, 1989.
- [223] T. Fraichard. A short paper about motion safety. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1140–1145, 2007.
- [224] German Federal Ministry of Transport and Digital Infrastructures. Ethics commission — Automated and connected driving, 2017. URL: <https://www.bmvi.de/SharedDocs/DE/Anlage/Presse/084-dobrindt-bericht-der-ethik-kommission.pdf>.
- [225] P. O. Scokaert and J. B. Rawlings. Infinite horizon linear quadratic control with constraints. *IFAC Proceedings Volumes*, 29(1):5905 – 5910, 1996.
- [226] W. Zhang, J. Hu, and A. Abate. Infinite-horizon switched LQR problems in discrete time: A suboptimal algorithm with performance analysis. *IEEE Transactions on Automatic Control*, 57(7):1815–1821, 2012.
- [227] P. Grieder, F. Borrelli, F. Torrisi, and M. Morari. Computation of the constrained infinite time linear quadratic regulator. *Automatica*, 40(4):701 – 708, 2004.
- [228] H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205 – 1217, 1998.

Bibliography

- [229] E. S. Meadows and J. B. Rawlings. Receding horizon control with an infinite horizon. In *Proc. of the American Control Conference*, pages 2926–2930, 1993.
- [230] D. Carlson, A. Haurie, and A. Leizarowitz. *Infinite Horizon Optimal Control: Deterministic and Stochastic Systems*. Springer Berlin Heidelberg, 2012.
- [231] T. Erez, Y. Tassa, and E. Todorov. Infinite-horizon model predictive control for periodic tasks with contacts. *Robotics: Science and Systems*, pages 1–8, 2012.
- [232] D. J. White. A survey of applications of Markov decision processes. *Journal of the Operational Research Society*, 44(11):1073–1096, 1993.
- [233] L. Mihaylova, T. Lefebvre, H. Bruyninckx, K. Gadeyne, and J. De Schutter. A comparison of decision making criteria and optimization methods for active robotic sensing. In *Proc. of the Int. Conf. on Numerical Methods and Applications*, pages 316–324. Springer, 2002.
- [234] D. Szer and F. Charpillet. An optimal best-first search algorithm for solving infinite horizon DEC-POMDPs. In J. Gama, R. Camacho, P. B. Brazdil, A. M. Jorge, and L. Torgo, editors, *Proc. of the European Conf. on Machine Learning*, pages 389–399, 2005.
- [235] B. Lacerda, D. Parker, and N. Hawes. Optimal and dynamic planning for Markov decision processes with co-safe LTL specifications. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, pages 1511–1516, 2014.
- [236] J. Tůmová and D. V. Dimarogonas. Multi-agent planning under local LTL specifications and event-based synchronization. *Automatica*, 70:239–248, 2016.
- [237] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.
- [238] N. Vlassis and M. Toussaint. Model-free reinforcement learning as mixture learning. In *Proc. of the Int. Conf. on Machine Learning*, pages 1081–1088, 2009.
- [239] Q. Liu, L. Li, Z. Tang, and D. Zhou. Breaking the curse of horizon: Infinite-horizon off-policy estimation. In *Advances in Neural Information Processing Systems*, pages 5356–5366, 2018.
- [240] K. G. Vamvoudakis. Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach. *Systems & Control Letters*, 100:14 – 20, 2017.

- [241] T. Fraichard and H. Asama. Inevitable collision states. A step towards safer robots? In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, pages 388–393, 2003.
- [242] N. Chan, J. Kuffner, and M. Zucker. Improved motion planning speed and safety using regions of inevitable collision. In *CISM-IFTOMM Symposium on robot design, dynamics, and control*, pages 103–114, 2008.
- [243] D. Althoff. *Safety Assessment for Motion Planning in Uncertain and Dynamic Environments*. Dissertation, Technische Universität München, München, 2014.
- [244] L. Martinez-Gomez and T. Fraichard. An efficient and generic 2D inevitable collision state-checker. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, pages 234–241, 2008.
- [245] L. Martinez-Gomez and T. Fraichard. Collision avoidance in dynamic environments: An ICS-based solution and its comparative evaluation. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 100–105, 2009.
- [246] D. Althoff, M. Buss, A. Lawitzky, M. Werling, and D. Wollherr. On-line trajectory generation for safe and optimal vehicle motion planning. In *Autonomous Mobile Systems*, pages 99–107. 2012.
- [247] R. Parthasarathi and T. Fraichard. An inevitable collision state-checker for a car-like vehicle. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 3068–3073, 2007.
- [248] A. Lawitzky, A. Nicklas, D. Wollherr, and M. Buss. Determining states of inevitable collision using reachability analysis. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, pages 4142–4147, 2014.
- [249] S. Söntges and M. Althoff. Determining the nonexistence of evasive trajectories for collision avoidance systems. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, pages 956–961, 2015.
- [250] A. Bautin, L. Martinez-Gomez, and T. Fraichard. Inevitable collision states: A probabilistic perspective. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 4022–4027, 2010.
- [251] D. Althoff, M. Althoff, D. Wollherr, and M. Buss. Probabilistic collision state checker for crowded environments. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1492–1498, 2010.
- [252] K. E. Bekris and L. E. Kavraki. Greedy but safe replanning under kinodynamic constraints. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 704–710, 2007.

- [253] F. Blanchini. Set invariance in control. *Automatica*, 35(11):1747 – 1767, 1999.
- [254] H. Michalska and D. Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 38(11):1623–1633, 1993.
- [255] M. Jalalmaab, B. Fidan, S. Jeon, and P. Falcone. Guaranteeing persistent feasibility of model predictive motion planning for autonomous vehicles. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 843–848, 2017.
- [256] C. Danielson, A. Weiss, K. Berntorp, and S. Di Cairano. Path planning using positive invariant sets. In *Proc. of the IEEE Int. Conf. on Decision and Control*, pages 5986–5991, 2016.
- [257] F. Blanchini, F. A. Pellegrino, and L. Visentini. Control of manipulators in a constrained workspace by means of linked invariant sets. *Int. Journal of Robust and Nonlinear Control*, 14(1314):1185–1205, 2004.
- [258] X. Qi, D. Theilliol, D. Song, and J. Han. Invariant-Set-Based Planning approach for obstacle avoidance under vehicle dynamic constraints. In *Proc. of the IEEE Int. Conf. on Robotics and Biomimetics*, pages 1692–1697, 2015.
- [259] G. Franzé and W. Lucia. A receding horizon control strategy for autonomous vehicles in dynamic environments. *IEEE Transactions on Control Systems Technology*, 24(2):695–702, 2016.
- [260] K. Berntorp, A. Weiss, C. Danielson, and S. Di Cairano. Automated driving: safe motion planning using positively invariant sets. In *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, pages 1–6, 2017.
- [261] T. Schouwenaars. *Safe trajectory planning of autonomous vehicles*. Dissertation, Massachusetts Institute of Technology, 2006.
- [262] D. Althoff, M. Althoff, and S. Scherer. Online safety verification of trajectories for unmanned flight with offline computed robust invariant sets. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, pages 3470–3477, 2015.
- [263] S. Mammar, N. A. Oufroukh, Z. Yacine, D. Ichalal, and L. Nouveliere. Invariant set based variable headway time vehicle longitudinal control assistance. In *Proc. of the American Control Conference*, pages 2922–2927, 2012.
- [264] N. Aréchiga and B. Krogh. Using verified control envelopes for safe controller design. In *Proc. of the IEEE American Control Conference*, pages 2918–2923, 2014.

- [265] E. C. Kerrigan and J. M. Maciejowski. Invariant sets for constrained nonlinear discrete-time systems with application to feasibility in model predictive control. In *Proc. of the IEEE Int. Conf. on Decision and Control*, volume 5, pages 4951–4956, 2000.
- [266] M. Althoff. An introduction to CORA 2015. In *Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems*, pages 1–28, 2015.
- [267] A. Rizaldi, F. Immler, and M. Althoff. A formally verified checker of the safe distance traffic rules for autonomous vehicles. In *NASA Formal Methods Symposium*, pages 175–190, 2016.
- [268] S. Magdici and M. Althoff. Adaptive cruise control with safety guarantees for autonomous vehicles. In *Proc. of the 20th World Congress of the Int. Federation of Automatic Control*, pages 5939–5946, 2017.
- [269] E. Velenis and P. Tsotras. Optimal velocity profile generation for given acceleration limits; the half-car model case. In *Proc. of the IEEE Int. Symposium on Industrial Electronics*, pages 361–366, 2005.
- [270] A. Rizaldi, J. Keinholz, M. Huber, J. Feldle, F. Immler, M. Althoff, E. Hilgendorf, and T. Nipkow. Formalising and monitoring traffic rules for autonomous vehicles in isabelle/hol. In *Integrated Formal Methods*, pages 50–66, 2017.
- [271] M. Koschi and M. Althoff. Interaction-aware occupancy prediction of road vehicles. In *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, pages 1885–1892, 2017.
- [272] A. Tamke, T. Dang, and G. Breuel. A flexible method for criticality assessment in driver assistance systems. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 697–702, 2011.
- [273] J. Hillenbrand, A. M. Spieker, and K. Kroschel. A multilevel collision mitigation approach - its situation assessment, decision making, and performance tradeoffs. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):528–540, 2006.
- [274] A. Berthelot, A. Tamke, T. Dang, and G. Breuel. A novel approach for the probabilistic computation of time-to-collision. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 1173–1178, 2012.
- [275] S. Ovchinnikov. Max-min representation of piecewise linear functions. *Contributions to Algebra and Geometry*, 43(1):297–302, 2002.
- [276] M. Herceg, M. Kvasnica, C. Jones, and M. Morari. Multi-Parametric Toolbox 3.0. In *Proc. of the European Control Conference*, pages 502–510, Zürich, Switzerland, 2013. <http://control.ee.ethz.ch/~mpt>.

Bibliography

- [277] S. Söntges, M. Koschi, and M. Althoff. Worst-case analysis of the time-to-react using reachable sets. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 1891–1897, 2018.
- [278] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. 2017.
- [279] V. Rausch, A. Hansen, E. Solowjow, C. Liu, E. Kreuzer, and J. K. Hedrick. Learning a deep neural net policy for end-to-end control of autonomous vehicles. In *Proc. of the American Control Conference*, pages 4914–4919, 2017.
- [280] H. M. Eraqi, M. N. Moustafa, and J. Honer. End-to-end deep learning for steering autonomous vehicles considering temporal dependencies. *arXiv preprint arXiv:1710.03804*, pages 1–8, 2017.
- [281] International Organization for Standardization (ISO). ISO 26262-10:2012 Road vehicles – functional safety, 2012.
- [282] F. Gruber and M. Althoff. Anytime safety verification of autonomous vehicles. In *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, pages 1708–1714, 2018.
- [283] S. Thrun. Learning occupancy grid maps with forward sensor models. *Autonomous robots*, 15(2):111–127, 2003.
- [284] S. Steyer, G. Tanzmeister, and D. Wollherr. Object tracking based on evidential dynamic occupancy grids in urban environments. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 1064–1070, 2017.
- [285] M. Dupuis, M. Strobl, and H. Grezlikowski. OpenDRIVE 2010 and beyond—status and future of the de facto standard for the description of road networks. In *Proc. of the Driving Simulation Conference Europe*, pages 231–242, 2010.
- [286] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. ROS: an open-source robot operating system. In *Proc. of the IEEE Int. Conf. on Robotics and Automation – Workshop on open source software*, pages 1–6, 2009.
- [287] European New Car Assessment Programme. AEB Pedestrian, 2019. URL: <https://www.euroncap.com/en/vehicle-safety/the-ratings-explained/vulnerable-road-user-vru-protection/aeb-pedestrian/>.
- [288] European New Car Assessment Programme. BMW 5-series AEB tests, 2019. URL: <https://www.euroncap.com/en/results/bmw/5-series/26656>.
- [289] J. Archer and K. Vogel. *The Traffic Safety Problems in Urban Areas*. 2000.

- [290] S. Hörl, F. Becker, T. J. P. Dubernet, and K. W. Axhausen. Induced demand by autonomous vehicles: An assessment. Technical report, ETH Zurich, 2019.
- [291] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller. Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction. *IEEE Transactions on Intelligent Vehicles*, 3(1):5–17, 2018.
- [292] M. Naumann, H. Königshof, and C. Stiller. Provably safe and smooth lane changes in mixed traffic. In *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, pages 1832–1837, 2019.
- [293] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300, 1995.
- [294] JASP Team. JASP (Version 0.10.2)[Computer software], 2019. URL: <https://jasp-stats.org/>.
- [295] M. Eid, M. Gollwitzer, and M. Schmitt. *Statistik und Forschungsmethoden: Lehrbuch. Mit Online-Material*. Beltz, 2017.
- [296] P. Trautman and A. Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, pages 797–803, 2010.
- [297] C. Menéndez-Romero, F. Winkler, C. Dornhege, and W. Burgard. Maneuver planning for highly automated vehicles. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 1458–1464, 2017.
- [298] M. Nolte, S. Ernst, J. Richelmann, and M. Maurer. Representing the unknown — Impact of uncertainty on the interaction between decision making and trajectory generation. In *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, pages 2412–2418, 2018.
- [299] J. Wei, J. M. Snider, J. Kim, J. M. Dolan, R. Rajkumar, and B. Litkouhi. Towards a viable autonomous driving research platform. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 763–770, 2013.
- [300] M. Jonasson and M. Thor. Steering redundancy for self-driving vehicles using differential braking. *Vehicle System Dynamics*, 56(5):791–809, 2018.
- [301] J. Barnes. *SPARK: The Proven Approach to High Integrity Software*. Altran Praxis, 2012.
- [302] G. Schildbach. On the application of ISO 26262 in control design for automated vehicles. In *2nd Int. Workshop on Safe Control of Autonomous Vehicles*, pages 74–82, 2018.

Bibliography

- [303] J. Tretmans. A formal approach to conformance testing. In *Proc. of the Int. Workshop on Protocol Test systems*, pages 257–276, 1993.
- [304] M. Van Osch. Hybrid input-output conformance and test generation. In *Formal Approaches to Software Testing and Runtime Verification*, pages 70–84. Springer, 2006.
- [305] H. Roehm, J. Oehlerking, M. Woehrle, and M. Althoff. Reachset conformance testing of hybrid automata. In *Proc. of the Int. Conf. on Hybrid Systems: Computation and Control*, pages 277–286, 2016.
- [306] H. Araujo, G. Carvalho, M. Mohaqeqi, M. R. Mousavi, and A. Sampaio. Sound conformance testing for cyber-physical systems: Theory and implementation. *Science of Computer Programming*, 162:35 – 54, 2018.
- [307] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. <http://cvxr.com/cvx>, March 2014.

A Appendix

A.1 Vehicle Shape Approximation

Given a rectangular shape of the autonomous vehicle with length $p_{\text{length}} \in \mathbb{R}_+$ and width $p_{\text{width}} \in \mathbb{R}_+$, we compute an approximation of the shape with $n \in \mathbb{N}_+$ circles with equal radius r . Therefore, we first divide the original shape in smaller rectangles with equal length

$$p'_{\text{length}} := p_{\text{length}}/n \quad (\text{A.1})$$

and width

$$p'_{\text{width}} := p_{\text{width}}. \quad (\text{A.2})$$

For the obtained smaller rectangles, we compute a circumscribing circle with radius r as:

$$r := \sqrt{(p'_{\text{length}}/2)^2 + (p'_{\text{width}}/2)^2}. \quad (\text{A.3})$$

Finally, the distance between the first and last circle is computed as:

$$\ell := p'_{\text{length}}(n - 1). \quad (\text{A.4})$$

In order to avoid imprecise shape approximations due to numerical issues, the values for r and ℓ should be enlarged by a safety margin for usage in real vehicles.

A.2 Random Planner

The random planner is based on the presented optimization approach in Sec. 3.2. However, the planner modifies the given constraints and goals before optimizing the trajectory. The new longitudinal position constraint $s'_{\text{max}}(t)$ are given by $s'_{\text{max}}(t) = s_{\text{max}}(t) + s_{\text{off}}$, where $s_{\text{off}} \in \mathbb{R}_+$ is a randomly drawn position offset. Similarly, the desired velocity is set to $v'_{\text{des}} = v_{\text{des}} + v_{\text{off}}$, where $v_{\text{off}} \in [0, 100]$ is a random velocity offset. The lateral motion is modified to exhibit oscillating motions. The oscillation is achieved by punishing position deviations from a given reference $\phi_{\text{ref},d}$ in the cost function. This reference is modelled as a sinusoidal curve $\phi_{\text{ref},d}(t) = \rho_{\text{scale}} \sin(\rho_{\text{freq}} t)$, where $\rho_{\text{scale}} \in [0, 1]$ and $\rho_{\text{freq}} \in [0, 1]$ are randomly chosen.

A.3 Parameters of the Fail-Safe Planning Experiments

A.3.1 Cut-in vehicles on highway

Table A.1: Parameters of the highway scenario (cf. Sec. 3.5.1).

Description	Value
Ego vehicle	$(x, y, \theta, v)_{\text{ego}}^T = (2.25 \text{ m}, 3.5 \text{ m}, 0 \text{ rad}, 23 \text{ m/s})^T$
Vehicle b_1	$(x, y, \theta, v)_{b_1}^T = (10 \text{ m}, 7 \text{ m}, 0 \text{ rad}, 20 \text{ m/s})^T$
Vehicle b_2	$(x, y, \theta, v)_{b_2}^T = (25 \text{ m}, 3.5 \text{ m}, 0 \text{ rad}, 25 \text{ m/s})^T$
Vehicle b_3	$(x, y, \theta, v)_{b_3}^T = (30 \text{ m}, 7 \text{ m}, 0 \text{ rad}, 30 \text{ m/s})^T$
Vehicle b_4	$(x, y, \theta, v)_{b_4}^T = (42 \text{ m}, 3.5 \text{ m}, 0 \text{ rad}, 20 \text{ m/s})^T$
Vehicle b_5	$(x, y, \theta, v)_{b_5}^T = (45 \text{ m}, 7 \text{ m}, 0 \text{ rad}, 35 \text{ m/s})^T$
Planning horizon	$t_{\mathfrak{F}} = 4.0 \text{ s}, N_{\mathfrak{F}} = 40, \Delta t = 0.1 \text{ s}$
Vehicle shape approximation	$r = 1.3 \text{ m}, \ell = 3 \text{ m}$
Feasible lon. and lat. acceleration	$a \in [-8 \text{ m/s}^2, 8 \text{ m/s}^2]$
Feasible jerk	$j \in [-10 \text{ m/s}^3, 10 \text{ m/s}^3]$
Feasible curvature	$\kappa \in [-0.2/\text{m}, 0.2/\text{m}]$
Feasible change of curvature	$\dot{\kappa} \in [-0.2/(\text{m s}), 0.2/(\text{m s})]$

A.3.2 Urban T-junction

Table A.2: Parameters of the urban T-junction scenario (cf. Sec. 3.5.2).

Description	Value
Ego vehicle in (a)	$(x, y, \theta, v)_{\text{ego}}^T = (45.8 \text{ m}, -2.7 \text{ m}, 2.9 \text{ rad}, 8.3 \text{ m/s})^T$
Ego vehicle in (b)	$(x, y, \theta, v)_{\text{ego}}^T = (27.2 \text{ m}, 1 \text{ m}, 3 \text{ rad}, 8.3 \text{ m/s})^T$
Vehicle b_1	$(x, y, \theta, v)_{b_1}^T = (14.6 \text{ m}, 11 \text{ m}, -1.67 \text{ rad}, 7 \text{ m/s})^T$
Vehicle b_2	$(x, y, \theta, v)_{b_2}^T = (8 \text{ m}, 0 \text{ m}, -0.1 \text{ rad}, 14 \text{ m/s})^T$
Vehicle b_3	$(x, y, \theta, v)_{b_3}^T = (18 \text{ m}, 14.6 \text{ m}, 1.73 \text{ rad}, 7 \text{ m/s})^T$
Planning horizon	$t_{\mathfrak{F}} = 6.0 \text{ s}, N_{\mathfrak{F}} = 30, \Delta t = 0.2 \text{ s}$
Vehicle shape approximation	$r = 1.3 \text{ m}, \ell = 3 \text{ m}$
Feasible lon. and lat. acceleration	$a \in [-8 \text{ m/s}^2, 8 \text{ m/s}^2]$
Feasible jerk	$j \in [-10 \text{ m/s}^3, 10 \text{ m/s}^3]$
Feasible curvature	$\kappa \in [-0.2/\text{m}, 0.2/\text{m}]$
Feasible change of curvature	$\dot{\kappa} \in [-0.2/(\text{m s}), 0.2/(\text{m s})]$

A.3.3 Intersection with crossing pedestrian

Table A.3: Parameters of the pedestrian scenario (cf. Sec. 3.5.3).

Description	Value
Ego vehicle	$(x, y, \theta, v)_{\text{ego}}^T = (6 \text{ m}, -31.42 \text{ m}, 1.57 \text{ rad}, 12.5 \text{ m/s})^T$
Pedestrian	$(x, y, \theta, v)^T = (8 \text{ m}, -12.7 \text{ m}, 3.64 \text{ rad}, 1.4 \text{ m/s})^T$
Planning horizon	$t_{\mathfrak{F}} = 6.0 \text{ s}, N_{\mathfrak{F}} = 30, \Delta t = 0.2 \text{ s}$
Vehicle shape approximation	$r = 1.3 \text{ m}, \ell = 3 \text{ m}$
Feasible lon. and lat. acceleration	$a \in [-8 \text{ m/s}^2, 8 \text{ m/s}^2]$
Feasible jerk	$j \in [-10 \text{ m/s}^3, 10 \text{ m/s}^3]$
Feasible curvature	$\kappa \in [-0.2/\text{m}, 0.2/\text{m}]$
Feasible change of curvature	$\dot{\kappa} \in [-0.2/(\text{m s}), 0.2/(\text{m s})]$

A.3.4 Distinct driving corridors

Table A.4: Parameters of the driving corridor scenario (cf. Sec. 3.5.5).

Description	Value
Ego vehicle	$(x, y, \theta, v)_{\text{ego}}^T = (45.8 \text{ m}, -2.7 \text{ m}, 2.9 \text{ rad}, 8.3 \text{ m/s})^T$
Obstacle b_1	$(x, y, \theta, v)_{b_1}^T = (14.6 \text{ m}, 11 \text{ m}, -1.67 \text{ rad}, 7 \text{ m/s})^T$
Vehicle b_2	$(x, y, \theta, v)_{b_2}^T = (8 \text{ m}, 0 \text{ m}, -0.1 \text{ rad}, 14 \text{ m/s})^T$
Vehicle b_3	$(x, y, \theta, v)_{b_3}^T = (18 \text{ m}, 14.6 \text{ m}, 1.73 \text{ rad}, 7 \text{ m/s})^T$
Planning horizon	$t_{\mathfrak{F}} = 6.0 \text{ s}, N_{\mathfrak{F}} = 30, \Delta t = 0.2 \text{ s}$
Vehicle shape approximation	$r = 1.3 \text{ m}, \ell = 3 \text{ m}$
Feasible lon. and lat. acceleration	$a \in [-8 \text{ m/s}^2, 8 \text{ m/s}^2]$
Feasible jerk	$j \in [-10 \text{ m/s}^3, 10 \text{ m/s}^3]$
Feasible curvature	$\kappa \in [-0.2/\text{m}, 0.2/\text{m}]$
Feasible change of curvature	$\dot{\kappa} \in [-0.2/(\text{m s}), 0.2/(\text{m s})]$

A.4 Parameters of the Invariably Safe Set Experiments

A.4.1 Verification of trajectories for infinite time horizons

Table A.5: Parameters of the urban scenario in Sec. 4.6.1.

Description	Value
Ego vehicle	$(s, d, v)_{\text{ego}}^T = (1.5 \text{ m}, 0 \text{ m}, 8.3 \text{ m/s})^T$
Vehicle b_1	$(s, d, v)_{b_1}^T = (8.5 \text{ m}, 0 \text{ m}, 6.9 \text{ m/s})^T$
Vehicle b_2	$(s, d, v)_{b_2}^T = (43.8 \text{ m}, 0 \text{ m}, 11.1 \text{ m/s})^T$
Vehicle b_3	$(s, d, v)_{b_3}^T = (101.7 \text{ m}, 0 \text{ m}, 8.3 \text{ m/s})^T$
Vehicle b_4	$(s, d, v)_{b_4}^T = (150.9 \text{ m}, 0 \text{ m}, 11.1 \text{ m/s})^T$
Lengths of vehicles	length = 3.0 m
Speed limit v_{limit}	$v_1 = 11.1 \text{ m/s}, v_2 = 8.3 \text{ m/s}, v_3 = 13.9 \text{ m/s}$
Maximum acceleration	$ a_{s,\text{max}} = 8.0 \text{ m/s}^2, a_{d,\text{max}} = 3.0 \text{ m/s}^2$
Reaction times	$\delta_{\text{brake}} = 0.3 \text{ s}, \delta_{\text{steer}} = 0.1 \text{ s}$

A.4.2 Invariably safe set for urban T-junction

Table A.6: Parameters of the T-junction scenario in Fig. 4.14.

Description	Value
Start of occupancy \mathcal{O}_2	$s_{\text{min}} = 11 \text{ m}$
Start of occupancy \mathcal{O}_3	$s_{\text{min}} = 20 \text{ m}$
Vehicle lengths	length = 3.0 m
Maximum acceleration	$ a_{s,\text{max}} = 10.0 \text{ m/s}^2, a_{d,\text{max}} = 10.0 \text{ m/s}^2$
Reaction times	$\delta_{\text{brake}} = 0.3 \text{ s}, \delta_{\text{steer}} = 0.1 \text{ s}$
Lengths of vehicles	length = 3.0 m

A.4.3 Existence of fail-safe trajectories

Table A.7: Parameters of the cut-in scenario in Fig. 4.15.

Description	Value
Ego vehicle	$(s, d, v)_{\text{ego}}^T = (0 \text{ m}, 0 \text{ m}, 20.0 \text{ m/s})^T$
Vehicle b_1	$(s, d, v)_{b_1}^T = (15.0 \text{ m}, 3.75 \text{ m}, 13.5 \text{ m/s})^T$
Lengths of vehicles	length = 3.0 m
Evasive distance	$d_{\text{eva}} = 3.75 \text{ m}$
Maximum acceleration	$ a_{s,\text{max}} = 8.0 \text{ m/s}^2, a_{d,\text{max}} = 8.0 \text{ m/s}^2$
Reaction times	$\delta_{\text{brake}} = 0.3 \text{ s}, \delta_{\text{steer}} = 0.1 \text{ s}$

A.5 Parameters of the Driving Experiments

A.5.1 Experiments with static obstacle

Table A.8: Parameters of the scenario in Fig. 6.4.

Description	Value
Initial state of ego vehicle	$(x, y, \theta, v, a)_{\text{ego}}^T =$ $(0.0 \text{ m}, 0.0 \text{ m}, 0.0 \text{ rad}, 0.0 \text{ m/s}, 0.02 \text{ m/s}^2)^T$
Position of static obstacle	$(x, y, \theta)_{\text{static}}^T =$ $(63.73 \text{ m}, 1.03 \text{ m}, 0.05 \text{ rad})^T$
Length and width of static obstacle	length = 6.02 m, width = 1.79 m
Time-To-React	$t_{\text{TTR}} = 7.0 \text{ s}$
State of ego vehicle at TTR	$(x, y, \theta, v, a)_{\text{ego}}^T =$ $(39.73 \text{ m}, 0.06 \text{ m}, 0.0 \text{ rad}, 9.53 \text{ m/s}, 0.07 \text{ m/s}^2)^T$
Intended trajectory horizon	$t_{\text{in}} = 12.5 \text{ s}, N_{\text{in}} = 50$
Random planner parameters	$s_{\text{off}} = 52.56 \text{ m}, v'_{\text{des}} = 9.46 \text{ m/s}, \rho_{\text{scale}} = 0.22,$ $\rho_{\text{freq}} = 0.53$
Fail-safe trajectory horizon	$t_{\text{fs}} = 5.0 \text{ s}, N_{\text{fs}} = 20$
Total verification time	$t_{\text{comp}} = 13 \text{ ms}$
Max. position error fail-safe tracking	$e_{p_x} = 0.55 \text{ m}, e_{p_y} = 0.09 \text{ m}$

Table A.9: Parameters of the scenario in Fig. 6.5.

Description	Value
Initial state of ego vehicle	$(x, y, \theta, v, a)_{\text{ego}}^T =$ $(0.0 \text{ m}, 0.0 \text{ m}, 0.0 \text{ rad}, 0.0 \text{ m/s}, -0.04 \text{ m/s}^2)^T$
Position of static obstacle	$(x, y, \theta)_{\text{static}}^T =$ $(72.16 \text{ m}, 0.1 \text{ m}, 0.09 \text{ rad})^T$
Length and width of static obstacle	length = 1.8 m, width = 1.35 m
Time-To-React	$t_{\text{TTR}} = 11.0 \text{ s}$
State of ego vehicle at TTR	$(x, y, \theta, v, a)_{\text{ego}}^T =$ $(49.56 \text{ m}, 0.06 \text{ m}, 0.02 \text{ rad}, 13.06 \text{ m/s}, 1.153 \text{ m/s}^2)^T$
Intended trajectory horizon	$t_{\text{in}} = 17.5 \text{ s}, N_{\text{in}} = 70$
Random planner parameters	$s_{\text{off}} = 73.18 \text{ m}, v'_{\text{des}} = 18.78 \text{ m/s}, \rho_{\text{scale}} = 0.19,$ $\rho_{\text{freq}} = 0.45$
Fail-safe trajectory horizon	$t_{\text{fs}} = 6.75 \text{ s}, N_{\text{fs}} = 27$
Total verification time	$t_{\text{comp}} = 13 \text{ ms}$
Max. position error fail-safe tracking	$e_{p_x} = 0.9 \text{ m}, e_{p_y} = 0.48 \text{ m}$
Max. abs. measured lat. acceleration	$a_{\text{max,lat}} = 4.4 \text{ m/s}^2$

A.5.2 Experiments with simulated vehicles

Table A.10: Parameters of the scenario in Fig. 6.8.

Description	Value
Initial state of ego vehicle	$(x, y, \theta, v, a)_{\text{ego}}^T =$ $(0.0 \text{ m}, 0.0 \text{ m}, 0.0 \text{ rad}, 13.78 \text{ m/s}, 0.26 \text{ m/s}^2)^T$
Initial state of other vehicle	$(x, y, \theta, v)_{\text{veh}}^T =$ $(54.3 \text{ m}, 3.5 \text{ m}, 0.0 \text{ rad}, 13.89 \text{ m/s})^T$
Length and width of other vehicle	length = 4.5 m, width = 2.0 m
SPOT parameters	$a_{\text{max,veh}} = 5 \text{ m/s}^2$, $v_{\text{max,veh}} = 13.9 \text{ m/s}$, $f_S = 1.2$
Time-To-React	$t_{\text{TTR}} = 2.75 \text{ s}$
State of ego vehicle at TTR	$(x, y, \theta, v, a)_{\text{ego}}^T =$ $(38.25 \text{ m}, 0.0 \text{ m}, 0.0 \text{ rad}, 13.94 \text{ m/s}, -0.02 \text{ m/s}^2)^T$
Intended trajectory horizon	$t_{\text{in}} = 12.0 \text{ s}$, $N_{\text{in}} = 48$
Fail-safe trajectory horizon	$t_{\text{fs}} = 7.5 \text{ s}$, $N_{\text{fs}} = 30$
Total verification time	$t_{\text{comp}} = 27 \text{ ms}$
Max. position error fail-safe tracking	$e_{p_x} = 0.61 \text{ m}$, $e_{p_y} = 0.02 \text{ m}$

Table A.11: Parameters of the scenario in Fig. 6.9.

Description	Value
Initial state of ego vehicle	$(x, y, \theta, v, a)_{\text{ego}}^T =$ $(0.0 \text{ m}, 0.0 \text{ m}, 0.0 \text{ rad}, 12.56 \text{ m/s}, 0.0 \text{ m/s}^2)^T$
Initial state of other vehicle	$(x, y, \theta, v)_{\text{veh}}^T =$ $(45.0 \text{ m}, 3.5 \text{ m}, 0.0 \text{ rad}, 13.89 \text{ m/s})^T$
Length and width of other vehicle	length = 4.5 m, width = 2.0 m
SPOT parameters	$a_{\text{max,veh}} = 5 \text{ m/s}^2$, $v_{\text{max,veh}} = 13.9 \text{ m/s}$, $f_S = 1.2$
Time-To-React	$t_{\text{TTR}} = 3.25 \text{ s}$
State of ego vehicle at TTR	$(x, y, \theta, v, a)_{\text{ego}}^T =$ $(40.73 \text{ m}, 0.0 \text{ m}, 0.0 \text{ rad}, 12.51 \text{ m/s}, 0.0 \text{ m/s}^2)^T$
Intended trajectory horizon	$t_{\text{in}} = 12.0 \text{ s}$, $N_{\text{in}} = 48$
Fail-safe trajectory horizon	$t_{\text{fs}} = 7.5 \text{ s}$, $N_{\text{fs}} = 30$
Total verification time	$t_{\text{comp}} = 26 \text{ ms}$
Max. position error fail-safe tracking	$e_{p_x} = 0.83 \text{ m}$, $e_{p_y} = 0.96 \text{ m}$
Max. abs. measured lat. acceleration	$a_{\text{max,lat}} = 4.1 \text{ m/s}^2$

A.5.3 Experiments with simulated pedestrians

Table A.12: Parameters of the scenario in Fig. 6.12.

Description	Value
Initial state of ego vehicle	$(x, y, \theta, v, a)_{\text{ego}}^T = (0.0 \text{ m}, 0.0 \text{ m}, 0.0 \text{ rad}, 1.12 \text{ m/s}, -0.2 \text{ m/s}^2)^T$
Initial state of pedestrian	$(x, y, \theta, v)_{\text{ped}}^T = (58.2 \text{ m}, -1.91 \text{ m}, 1.57 \text{ rad}, 1.5 \text{ m/s})^T$
Radius of pedestrian	radius = 0.35 m
SPOT parameters	$a_{\text{max,ped}} = 0.6 \text{ m/s}^2$, $a_{\text{max,ped,stop}} = 0.6 \text{ m/s}^2$, $v_{\text{max,ped}} = 2 \text{ m/s}$, $b_{\text{cross}} = \text{True}$, $b_{\text{stop}} = \text{False}$, $d_{\text{perp}} = 1.5 \text{ m}$
Time-To-React	$t_{\text{TTR}} = 6.0 \text{ s}$
State of ego vehicle at TTR	$(x, y, \theta, v, a)_{\text{ego}}^T = (38.46 \text{ m}, 0.0 \text{ m}, 0.0 \text{ rad}, 12.24 \text{ m/s}, 1.56 \text{ m/s}^2)^T$
Intended trajectory horizon	$t_{\text{in}} = 12.5 \text{ s}$, $N_{\text{in}} = 50$
Fail-safe trajectory horizon	$t_{\text{fs}} = 6.75 \text{ s}$, $N_{\text{fs}} = 27$
Total verification time	$t_{\text{comp}} = 21 \text{ ms}$
Max. position error fail-safe tracking	$e_{p_x} = 2.25 \text{ m}$, $e_{p_y} = 0.43 \text{ m}$
Max. abs. measured lat. acceleration	$a_{\text{max,lat}} = 4.2 \text{ m/s}^2$

Table A.13: Parameters of the scenario in Fig. 6.14.

Description	Value
Initial state of ego vehicle	$(x, y, \theta, v, a)_{\text{ego}}^T = (0.0 \text{ m}, 0.0 \text{ m}, 0.0 \text{ rad}, 0.0 \text{ m/s}, 0.0 \text{ m/s}^2)^T$
Initial state of pedestrian	$(x, y, \theta, v)_{\text{ped}}^T = (58.05 \text{ m}, -1.91 \text{ m}, 1.57 \text{ rad}, 1.5 \text{ m/s})^T$
Radius of pedestrian	radius = 0.35 m
SPOT parameters	$a_{\text{max,ped}} = 0.3 \text{ m/s}^2$, $a_{\text{max,ped,stop}} = 0.3 \text{ m/s}^2$, $v_{\text{max,ped}} = 2 \text{ m/s}$, $b_{\text{cross}} = b_{\text{stop}} = \text{False}$, $d_{\text{perp}} = 1.5 \text{ m}$
Time-To-React	$t_{\text{TTR}} = 6.5 \text{ s}$
State of ego vehicle at TTR	$(x, y, \theta, v, a)_{\text{ego}}^T = (38.22 \text{ m}, 0.0 \text{ m}, 0.0 \text{ rad}, 12.22 \text{ m/s}, 1.57 \text{ m/s}^2)^T$
Intended trajectory horizon	$t_{\text{in}} = 17.5 \text{ s}$, $N_{\text{in}} = 70$
Fail-safe trajectory horizon	$t_{\text{fs}} = 6.75 \text{ s}$, $N_{\text{fs}} = 27$
Total verification time	$t_{\text{comp}} = 23 \text{ ms}$
Max. position error fail-safe tracking	$e_{p_x} = 0.83 \text{ m}$, $e_{p_y} = 0.44 \text{ m}$
Max. abs. measured lat. acceleration	$a_{\text{max,lat}} = 4.8 \text{ m/s}^2$

A.6 Post-processing Urban Traffic Situations

A.6.1 Post-processing steps

The intended trajectories for our results were obtained using the same convex optimization planning techniques as for our fail-safe planner (cf. *Planner 1*), but with different parametrization. Since the intended trajectories aim to provide comfortable and anticipatory behaviors, the intended planner only considers the most likely behaviors of other traffic participants instead of accounting for all of their legal actions. For the most likely prediction, we assume that traffic participants follow their current lane (or sidewalk) and only slightly accelerate or decelerate. For both intended and fail-safe trajectory optimization, we specified the same desired velocity. Thus, the fail-safe trajectory optimization aims at achieving or maintaining the desired velocity for as long as possible before coming to a standstill. This strategy contributes to improved comfort, as the temporary execution of fail-safe trajectories is not immediately perceived by passengers. Without loss of generality, we set the branch-off point t_{c+1} of the fail-safe trajectory from the intended trajectory to a constant replanning time. All parameters for the prediction of other traffic participants and for the motion planning of the autonomous vehicle, as well as further details, are provided in App. A.6.2.

Numerical experiments were conducted on a machine with a 2.60 GHz Intel Core i7-6700HQ processor and 16 GB of DDR3 memory. On average for all three presented scenarios, the required computation time was 177 ms, which can be split into 29 ms for prediction, 92 ms for drivable area computation, and 56 ms for driving corridor computation and trajectory optimization. Note that we did not include pre-processing steps for the input data. Fig. A.1 summarizes the computation times for the steps of the verification technique in more detail.

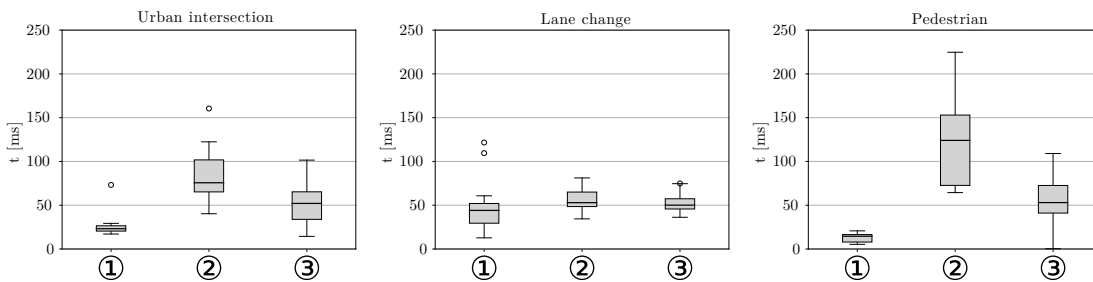


Figure A.1: Computation times for the presented three scenarios. (1) denotes the set-based prediction. (2) denotes the drivable area and safe set computation. (3) denotes the driving corridor and trajectory optimization.

A.6.2 Parameters for scenarios

All parameters of the safety layer and for the prediction of other traffic participants are given in Tab. A.14–A.17.

Table A.14: Parameters of the verification cycle.

Description	Value
Replanning cycle	$t_{c+1} - t_c = 600$ ms
Time discretization	$\Delta t = 200$ ms
Time horizon of \mathcal{J}_c in Scenario I & III	$t_{\mathcal{J}} = 10$ s
Time horizon of \mathcal{J}_c in Scenario II	$t_{\mathcal{J}} = 6$ s
Time horizon of \mathcal{F}_c in Scenario I & III	$t_{\mathcal{F}} = 6$ s
Time horizon of \mathcal{F}_c in Scenario II	$t_{\mathcal{F}} = 4$ s

Table A.15: Parameters of the set-based prediction.

Parameters for vehicles and motorcycles	Value
Maximum absolute acceleration	$ a_{\max, \text{veh}} = 8.0$ m/s ²
Maximum longitudinal acceleration	$ a_{\max, \text{lon}, \text{veh}} = 4.0$ m/s ²
Maximum velocity	$v_{\max, \text{veh}} = 14.0$ m/s
Speed limit in Scenario I	$v_{\text{limit}} = 13.89$ m/s
Speed limit in Scenario II	$v_{\text{limit}} = \infty$
Speed limit in Scenario III	$v_{\text{limit}} = 8.33$ m/s
Speeding factor	$f_S = 1.2$
Switching velocity	$v_{S, \text{veh}} = 7.0$ m/s
Parameters for bicycles	Value
Maximum absolute acceleration	$ a_{\max, \text{cyc}} = 3.5$ m/s ²
Maximum longitudinal acceleration	$ a_{\max, \text{lon}, \text{cyc}} = 3.5$ m/s ²
Maximum velocity	$v_{\max, \text{cyc}} = 7.0$ m/s
Switching velocity	$v_{S, \text{cyc}} = 3.0$ m/s
Parameters for pedestrians	Value
Maximum absolute acceleration	$ a_{\max, \text{ped}} = 0.6$ m/s ²
Maximum absolute velocity ¹	$ v_{\max, \text{ped}} = 3.0$ m/s
Maximum width of road strip	$d_{\text{slack}} = 0.75$ m
Maximum width when crossing	$d_{\text{perp}} = 3.0$ m
Deviation based on orientation $\theta^{(p)}$	$\alpha(\theta^{(p)}) = \max(\theta^{(p)}(t'_0)) + 0.1$ rad

¹The ISO 13855 [131] suggests a lower value of 2.0 m/s as transition speed between walking and running.

Table A.16: Parameters of the fail-safe planner. The parameters include the drivable area computation and the driving corridor and trajectory optimization.

Description	Value
Vehicle length	5.098 m
Vehicle width	1.902 m
Longitudinal accelerations	$a_{\text{lon}} \in [-8 \text{ m/s}^2, 3.5 \text{ m/s}^2]$
Lateral accelerations	$a_{\text{lat}} \in [-5.5 \text{ m/s}^2, 5.5 \text{ m/s}^2]$
Maximum acceleration	$a_{\text{max}} = 9.81 \text{ m/s}^2$
Longitudinal jerk	$j \in [-10 \text{ m/s}^3, 10 \text{ m/s}^3]$
Feasible curvature	$\kappa \in [-0.2/\text{m}, 0.2/\text{m}]$
Feasible curvature change	$\dot{\kappa} \in [-0.2/(\text{m s}), 0.2/(\text{m s})]$
Longitudinal velocities	$v_{\text{lon}} \in [0 \text{ m/s}, 14 \text{ m/s}]$
Lateral velocities	$v_{\text{lat}} \in [-7 \text{ m/s}, 7 \text{ m/s}]$
Desired velocity in Scenario I	$v_{\text{des}} = 8 \text{ m/s}$
Desired velocity in Scenario II & III	$v_{\text{des}} = 13.89 \text{ m/s}$

Table A.17: Parameters of the most likely prediction.

Parameters for vehicles and motorcycles	Value
Maximum absolute acceleration	$ a_{\text{max,veh}} = 2.0 \text{ m/s}^2$
Maximum longitudinal acceleration	$ a_{\text{max,lon,veh}} = 0.5 \text{ m/s}^2$
Minimum longitudinal acceleration	$ a_{\text{min,lon,veh}} = -0.5 \text{ m/s}^2$
Maximum velocity	$v_{\text{max,veh}} = 14.0 \text{ m/s}$
Speeding factor	$f_S = 1.0$
Switching velocity	$v_{S,\text{veh}} = 7.0 \text{ m/s}$
Parameters for bicycles	Value
Maximum absolute acceleration	$ a_{\text{max,cyc}} = 0.5 \text{ m/s}^2$
Maximum longitudinal acceleration	$ a_{\text{max,lon,cyc}} = 0.1 \text{ m/s}^2$
Minimum longitudinal acceleration	$ a_{\text{min,lon,cyc}} = -0.1 \text{ m/s}^2$
Maximum velocity	$v_{\text{max,cyc}} = 5.0 \text{ m/s}$
Switching velocity	$v_{S,\text{cyc}} = 3.0 \text{ m/s}$
Parameters for pedestrians	Value
Maximum absolute acceleration	$ a_{\text{max,ped}} = 0.2 \text{ m/s}^2$
Maximum absolute velocity ²	$ v_{\text{max,ped}} = 2.0 \text{ m/s}$
Maximum width of road strip	$d_{\text{slack}} = 0.75 \text{ m}$
Maximum width when crossing	$d_{\text{perp}} = 3.0 \text{ m}$
Deviation based on orientation $\theta^{(p)}$	$\alpha(\theta^{(p)}) = \max(\theta^{(p)}(t'_0)) + 0.1 \text{ rad}$

A.6.3 Parameterization of planners

All parameters for the prediction of other traffic participants as well as for the motion planning of the autonomous vehicle are given in Tab. A.18 and A.19.

Table A.18: Parameters of the intended planners 1 & 2. The parameters include the drivable area computation and the driving corridor and trajectory optimization.

Description	Value
Length	5.098 m
Width	1.902 m
Longitudinal accelerations	$a_{\text{lon}} \in [-5 \text{ m/s}^2, 2.5 \text{ m/s}^2]$
Lateral accelerations	$a_{\text{lat}} \in [-3 \text{ m/s}^2, 3 \text{ m/s}^2]$
Maximum acceleration	$a_{\text{lat}} = 9.81 \text{ m/s}^2$
Longitudinal jerk	$j \in [-10 \text{ m/s}^3, 10 \text{ m/s}^3]$
Feasible curvature	$\kappa \in [-0.2/\text{m}, 0.2/\text{m}]$
Feasible curvature change	$\dot{\kappa} \in [-0.2/(\text{m s}), 0.2/(\text{m s})]$
Longitudinal velocities	$v_{\text{lon}} \in [0 \text{ m/s}, 14 \text{ m/s}]$
Lateral velocities	$v_{\text{lat}} \in [-7 \text{ m/s}, 7 \text{ m/s}]$
Desired velocity in Scenario I	$v_{\text{des}} = 8 \text{ m/s}$
Desired velocity in Scenario II & III	$v_{\text{des}} = 13.89 \text{ m/s}$

Table A.19: Parameters of the intended planner 3. This planner is based on the sampling-based trajectory planner in [46].

Description	Value
Length	5.098 m
Width	1.902 m
Longitudinal accelerations	$a_{\text{lon}} \in [-9.81 \text{ m/s}^2, 9.81 \text{ m/s}^2]$
Lateral accelerations	$a_{\text{lat}} \in [-9.81 \text{ m/s}^2, 9.81 \text{ m/s}^2]$
Maximum acceleration	$a_{\text{max}} = 9.81 \text{ m/s}^2$
Feasible curvature	$\kappa \in [-0.2/\text{m}, 0.2/\text{m}]$
Feasible curvature change	$\dot{\kappa} \in [-0.2/(\text{m s}), 0.2/(\text{m s})]$
Feasible orientation change	$\dot{\theta} \in [-0.2 \text{ rad/s}, 0.2 \text{ rad/s}]$
Sampling step size for time	$\Delta t_{\text{samp}} = 200 \text{ ms}$
Sampling step size for velocity	$\Delta v_{\text{samp}} = 0.4 \text{ m/s}$
Sampling step size for lateral distance	$\Delta d_{\text{samp}} = 0.2 \text{ m}$
Longitudinal velocities	$v_{\text{lon}} \in [0 \text{ m/s}, 14 \text{ m/s}]$
Desired velocity in Scenario I	$v_{\text{des}} = 8 \text{ m/s}$
Desired velocity in Scenario II & III	$v_{\text{des}} = 13.89 \text{ m/s}$

A.6.4 Detailed planning cycle of intersection scenario

Fig. A.2 illustrates the verification results obtained during the selected planning cycle $c = 10$ of the urban intersection scenario (cf. Sec. 6.3.1). The subfigures show the predicted occupancy sets of obstacles at different times $t = t_c + t'_k$, where $t'_k := k\Delta t, k \leq N_{\mathfrak{F}}$, corresponds to the discrete time step of the fail-safe trajectory with discretization Δt and length $N_{\mathfrak{F}}$. The ego vehicle is depicted along $\mathfrak{J}_c^{\text{safe}}$ and the consecutive \mathfrak{F}_c .

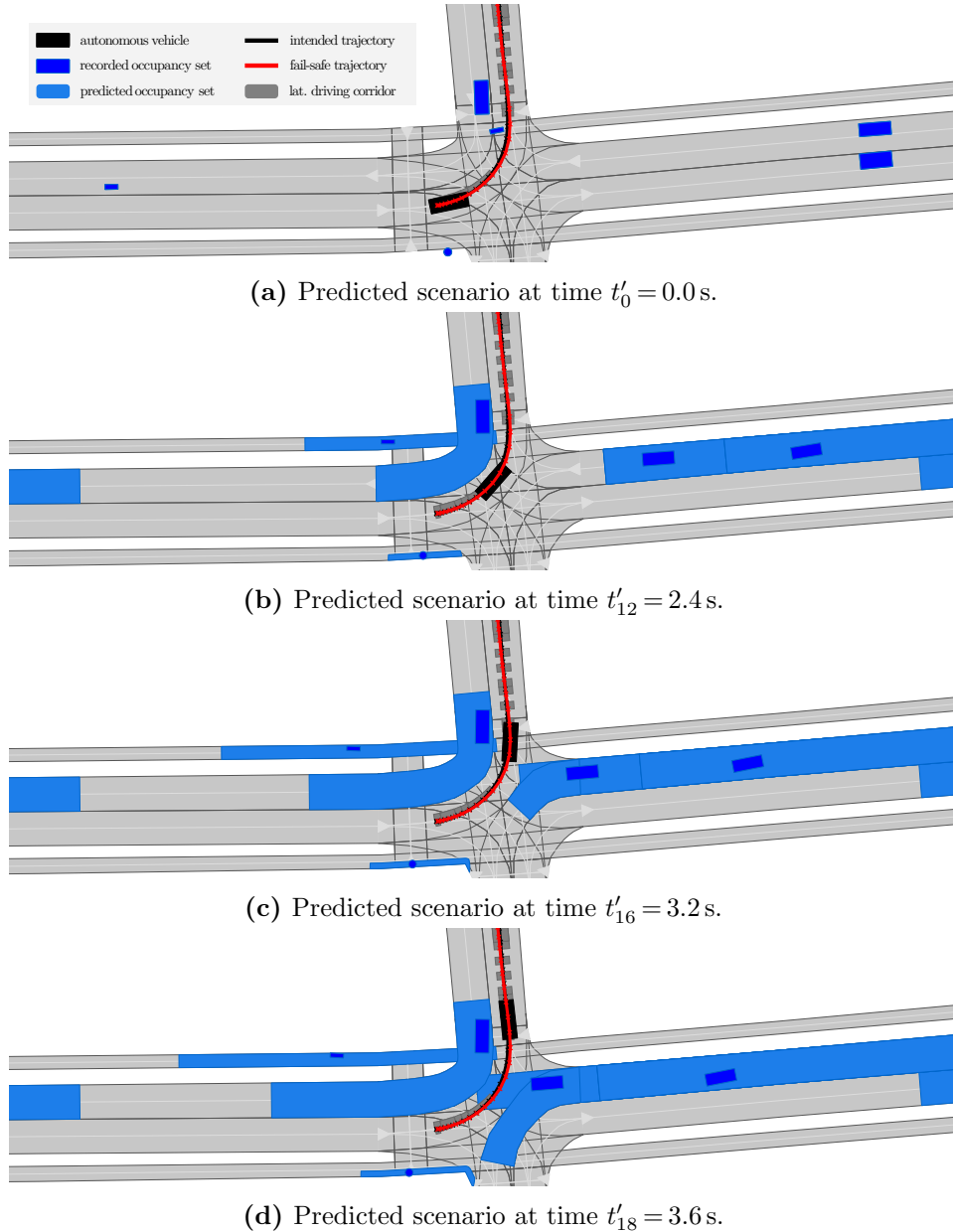


Figure A.2: Detailed verification results of urban intersection scenario. Visualized solution is obtained during planning cycle $c = 10$.

A.6.5 Detailed planning cycle of the lane change scenario

Fig. A.3 illustrates the verification results obtained during the selected planning cycle $c = 1$ of the lane change scenario (cf. Sec. 6.3.1). The subfigures show the predicted occupancy sets of obstacles at different times $t = t_{c+1} + t'_k$, where $t'_k := k\Delta t, k \leq N_{\mathfrak{F}}$, corresponds to the discrete time step of the fail-safe trajectory with discretization Δt and length $N_{\mathfrak{F}}$. The ego vehicle is depicted along $\mathfrak{J}_c^{\text{safe}}$ and the consecutive \mathfrak{F}_c .

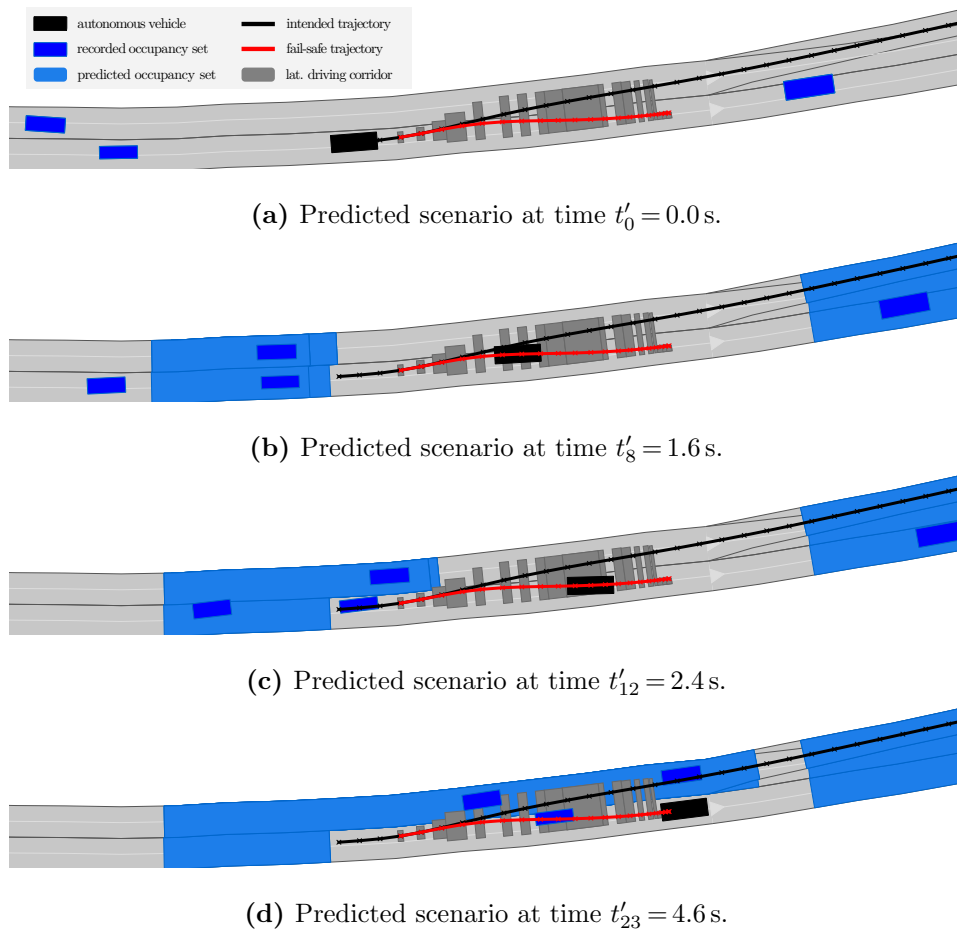


Figure A.3: Detailed verification results of lane change scenario. Visualized solution is obtained during planning cycle $c = 1$

A.6.6 Detailed planning cycle of the jaywalking pedestrian scenario

Fig. A.4 illustrates the verification results obtained during the selected planning cycle $c = 5$ of the pedestrian scenario (cf. Sec. 6.3.1). The subfigures show the predicted occupancy sets of obstacles at different times $t = t_{c+1} + t'_k$, where $t'_k := k\Delta t, k \leq N_{\mathfrak{F}}$, corresponds to the discrete time step of the fail-safe trajectory with discretization Δt and length $N_{\mathfrak{F}}$. The ego vehicle is depicted along $\mathfrak{J}_c^{\text{safe}}$ and the consecutive \mathfrak{F}_c .

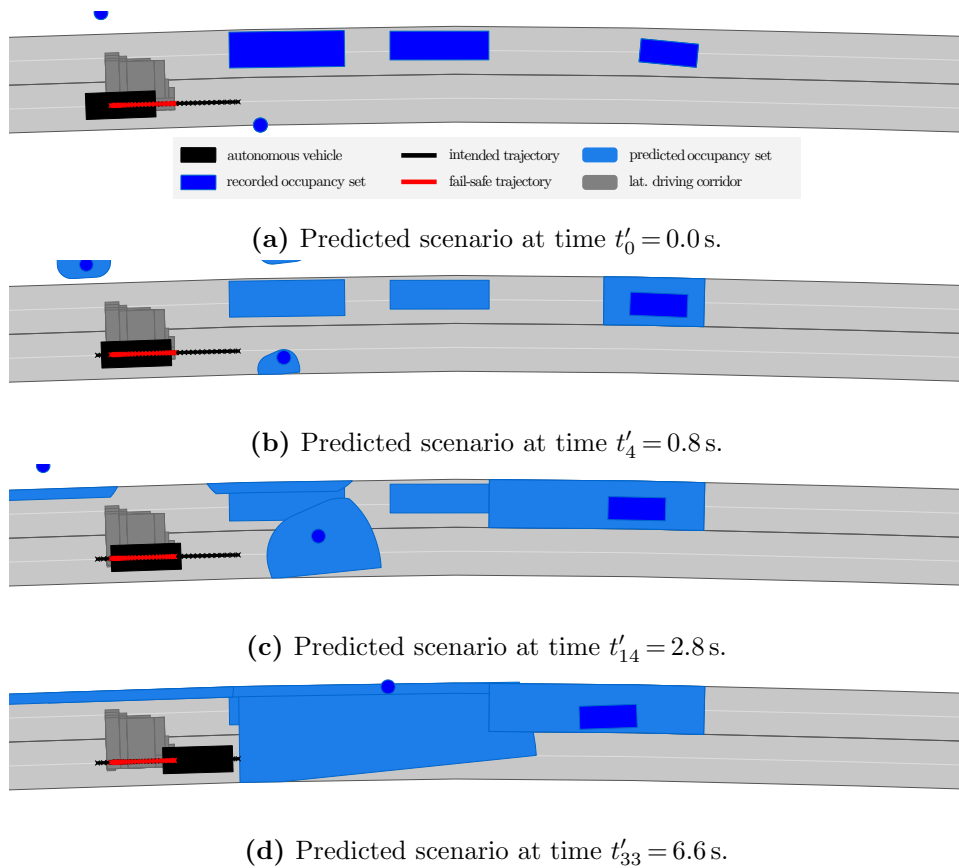


Figure A.4: Detailed verification results of lane change scenario. Visualized solution is obtained during planning cycle $c = 5$.

A.7 User Study in Driving Simulator

A.7.1 Overview of the safety-critical scenarios

Five safety-critical scenarios have been created for the user study. In all scenarios, the ego vehicle starts with a velocity of zero and in an invariably safe state. Fig. A.5 illustrates the different scenarios in a top view. The scenarios model the following traffic situations (description based on [5]):

1. **Emergency brake of the first preceding vehicle:** The ego vehicle follows a preceding vehicle, which suddenly performs an emergency brake maneuver until it has reached standstill. The goal of this scenario is, on the one hand, to evaluate the vehicle following behavior, and, on the other hand, the ability of the ego vehicle to react to a emergency brake maneuvers of other traffic participants.
2. **Emergency brake of the second preceding vehicle:** The ego vehicle follows a small transporter, and, after some time, a static vehicle appears in front of the transporter. The study participant cannot see the static vehicle because the transporter is blocking the view. The transporter performs an unexpected lane change to prevent a collision with the static vehicle. As a result, the static vehicle is now in front of the ego vehicle. This scenario tests the vehicle following behavior, the ability of the ACC to consider more than one leading vehicle, and the reaction when a static vehicle suddenly enters the field of view.
3. **Aggressive cut-in:** A vehicle from an adjacent lane performs an aggressive cut-in in front of the ego vehicle. The merging vehicle has a lower velocity than the ego vehicle and performs braking during its cut-in maneuver. The goal of the scenario is to demonstrate the ACC's ability to react to cut-in vehicles and to evaluate if the ego vehicle's behavior in is an appropriate reaction in this scenario.
4. **Smooth cut-in:** A vehicle from an adjacent lane performs a smooth cut-in into the ego vehicle's lane. The cut-in vehicle has a higher velocity than the ego vehicle and accelerates during its cut-in maneuver. This scenario has the same goals as those of the aggressive cut-in.
5. **Traffic jam end:** The ego vehicle approaches the end of a traffic jam (with zero velocity) and travels with $v_{\max, \text{ego}}$. This scenario demonstrates the ACC's ability to come to a standstill when the ego vehicle travels at high velocities and static vehicles enter the field of view.

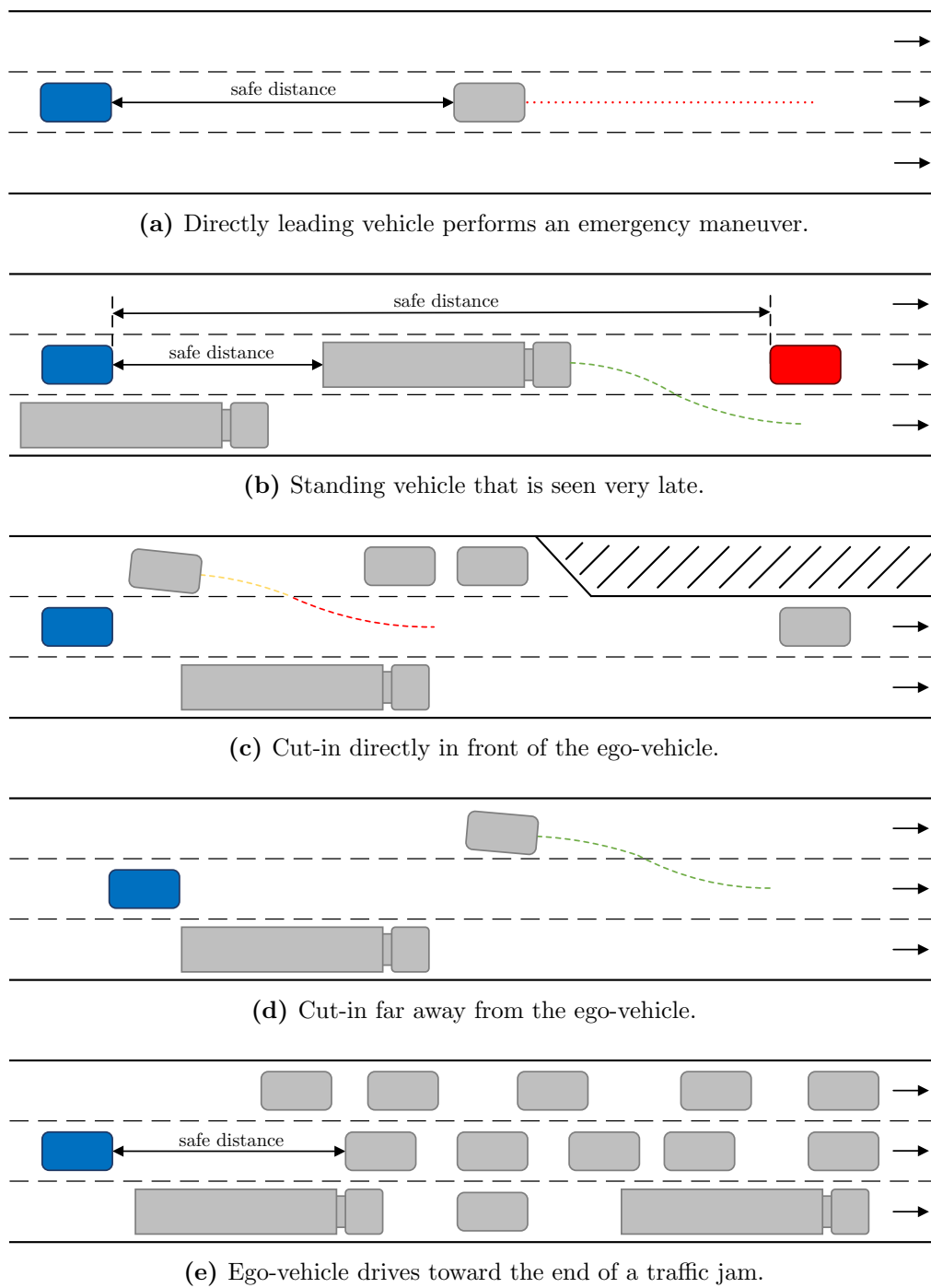


Figure A.5: Test scenarios for the comfort evaluation. The ego vehicle is shown in blue, surrounding vehicles in grey and standing vehicles in red. Figure adapted from [5].

A.7.2 Additional results of the simulations



(a)



(b)

Figure A.6: Additional simulation results I. (a) A vehicle cuts into the lane of the ego vehicle (cf. scenario in Fig. A.5c). (b) A preceding vehicle performs an emergency braking maneuver (cf. scenario in Fig. A.5a).



(a)



(b)

Figure A.7: Additional simulation results II. (a) The preceding vehicle (truck) avoids a collision with a standing vehicle (cf. scenario in Fig. A.5b). The ego vehicle is now heading towards the standing vehicle and has to react appropriately. (b) The ego vehicle avoids collisions with a traffic jam (cf. scenario in Fig. A.5e).

A.8 Intervention Assessment Experiments

A.8.1 Used parameters

Table A.20: Parameters of SPOT in the intervention study.

Parameters for vehicles and motorcycles	Value
Maximum absolute acceleration	$ a_{\max, \text{veh}} = 8.0 \text{ m/s}^2$
Maximum longitudinal acceleration	$ a_{\max, \text{lon}, \text{veh}} = 8.0 \text{ m/s}^2$
Maximum velocity	$v_{\max, \text{veh}} = 83.3 \text{ m/s}$
Speed limit in Scenario I	$v_{\text{limit}} = 13.89 \text{ m/s}$
Speeding factor	$f_S = 1.2$
Switching velocity	$v_{S, \text{veh}} = 5.0 \text{ m/s}$
Parameters for bicycles	Value
Maximum absolute acceleration	$ a_{\max, \text{cyc}} = 0.8 \text{ m/s}^2$
Maximum longitudinal acceleration	$ a_{\max, \text{lon}, \text{cyc}} = 0.8 \text{ m/s}^2$
Maximum velocity	$v_{\max, \text{cyc}} = 12.0 \text{ m/s}$
Switching velocity	$v_{S, \text{cyc}} = 5.0 \text{ m/s}$
Parameters for pedestrians	Value
Maximum absolute acceleration	$ a_{\max, \text{ped}} = 0.6 \text{ m/s}^2$
Maximum absolute velocity ³	$ v_{\max, \text{ped}} = 2.0 \text{ m/s}$
Maximum width of road strip	$d_{\text{slack}} = 1 \text{ m}$
Maximum width when crossing	$d_{\text{perp}} = 3.0 \text{ m}$
Deviation based on orientation $\theta^{(p)}$	$\alpha(\theta^{(p)}) = \max(\theta^{(p)}(t'_0)) + 0.1 \text{ rad}$

Table A.21: Parameters of the fail-safe planner in the intervention study.

Parameter	Value
Vehicle dimensions	length=5.098 m, width=1.902 m
Planning horizon	$N_{\mathfrak{F}} = 20, t_{\mathfrak{F}} = 5 \text{ s}$
Longitudinal accelerations	$a_{\text{lon}} \in [-8 \text{ m/s}^2, 8 \text{ m/s}^2]$
Lateral accelerations	$a_{\text{lat}} \in [-8 \text{ m/s}^2, 8 \text{ m/s}^2]$
Maximum acceleration	$a_{\max} = 8 \text{ m/s}^2$
Jerk	$j \in [-10 \text{ m/s}^3, 10 \text{ m/s}^3]$
Feasible curvature	$\kappa \in [-0.2/\text{m}, 0.2/\text{m}]$
Feasible curvature change	$\dot{\kappa} \in [-0.2/(\text{m s}), 0.2/(\text{m s})]$
Longitudinal velocities	$v_{\text{lon}} \in [0 \text{ m/s}, 55 \text{ m/s}]$
Reaction times	$\delta_{\text{brake}} = 0.3 \text{ s}, \delta_{\text{steer}} = 0.3 \text{ s}$

³The ISO 13855 [131] suggests a lower value of 2.0 m/s as transition speed between walking and running.

A.8.2 Additional results

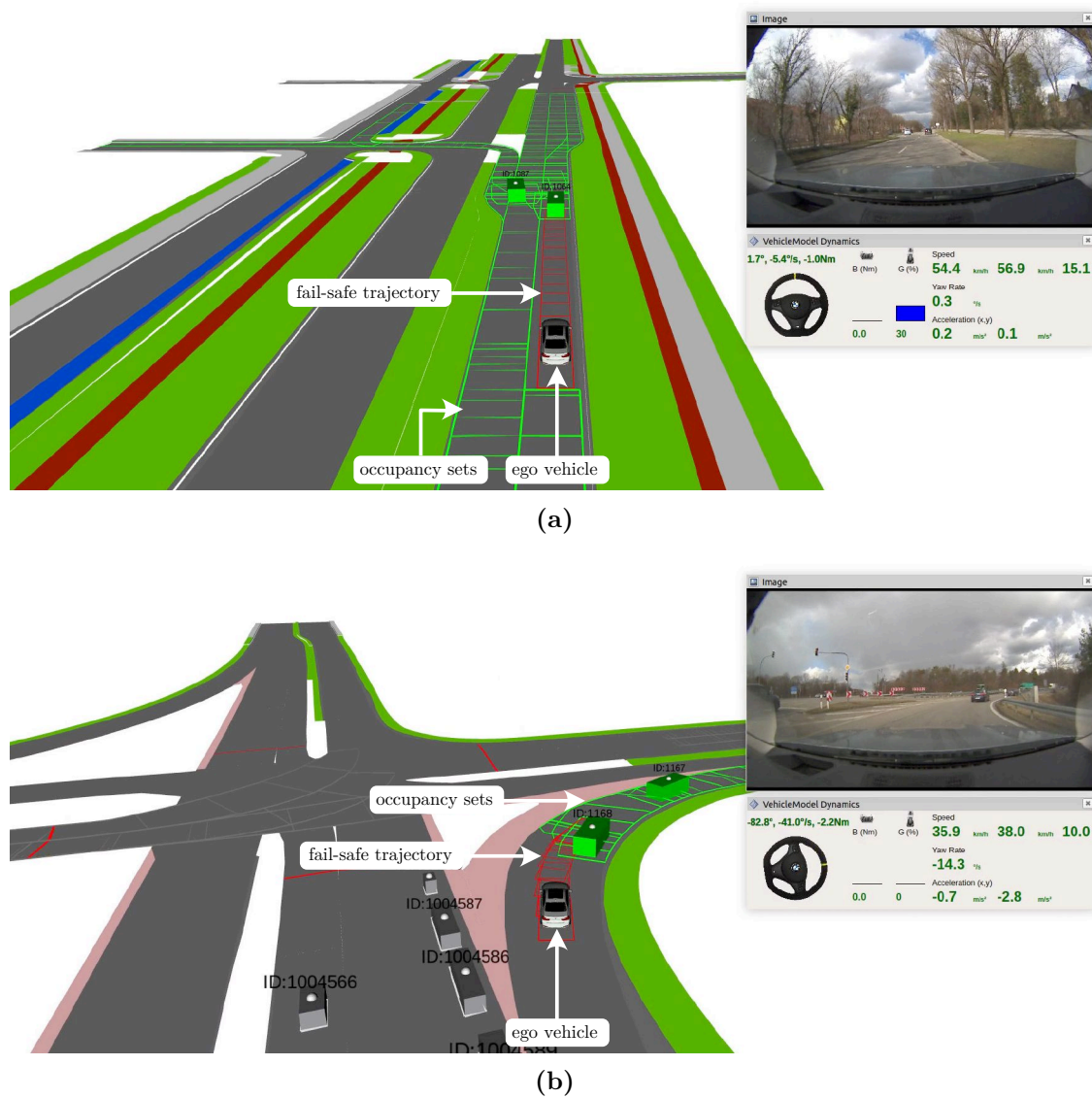


Figure A.8: Additional examples of true negatives. (a) Successfully computed fail-safe trajectory (red regions) in a two lane scenario. (b) Successfully computed fail-safe trajectory (red regions) while turning right.

A.9 Summary of Supplementary Material

A.9.1 Video files

Video V1: Fail-safe trajectory planning This video shows the simulation results of the proposed fail-safe trajectory planning approach (cf. Ch. 3). Our obtained fail-safe trajectories guarantee the safety of the ego vehicle with respect to any legal future motion of obstacles in real-time. We demonstrate the benefits of our comprehensive fail-safe planning approach in different highway and urban scenarios using the CommonRoad benchmark suite.

Link: <https://ieeexplore.ieee.org/document/8569425/media>

Video V2: Fail-safe trajectories to avoid collisions with pedestrians Based on the extended set-based prediction, we ensure the safety of planned motions in environments with pedestrians (cf. Ch. 3). The prediction provides a bounded region which includes all possible future states of the nondeterministic pedestrian model. This video demonstrates the use of our prediction method for fail-safe trajectory planning of autonomous vehicles.

Link: <https://ieeexplore.ieee.org/document/8569434/media>

Video V3: Drivable area computation This video shows the computation of the drivable area for different example scenarios. Moreover, we present the driving corridor selection and the trajectory optimization with the chosen corridors (cf. Ch. 3).

Link: see media attachment of [4] on *IEEE Xplore*.

Video V4: Invariably safe sets This video shows the simulation results of the proposed invariably safe sets approach (cf. Ch. 4). We demonstrate how one can verify planned trajectories of autonomous vehicles for an infinite time horizon in an example scenario. Only if the ego vehicle executes the invariably safe input trajectory, reaching an invariably safe set, it can come to a stop without colliding with other vehicles.

Link: <https://ieeexplore.ieee.org/document/8593597/media>

Video V5: Ensuring safety of reinforcement learning approaches The video shows the results of verifying the actions of an RL agent (cf. Ch. 4). The RL agent is shown in the learning and in the application phase in the simulated environment which we used for training and testing. We show examples of the RL agent when allowed to execute any action or only safe actions.

Link: <https://ieeexplore.ieee.org/document/8569448/media>

Video V6: Experiments with a test vehicle This video shows the results of our conducted vehicle experiments with a BMW 7-series test vehicle (cf. Ch. 5).

We highlight the planning and verification results for each scenario. Moreover, we present videos from each test in different perspectives. In each experiment, our fail-safe motion planning technique ensured the safety of the vehicle. Our results demonstrate the safety benefits and the drivability of fail-safe trajectories.

Link: see media attachment of [3] on *IEEEExplore*.

Video V7: Verification results of presented scenarios Starting with a short introduction to our provably correct verification technique, this video mainly shows the verification results of the urban intersection, lane change and pedestrian scenario (cf. Ch. 5). For each scenario, a video clip of the camera view during recording is shown and a short description of the traffic scene is given. Furthermore, video clips visualize the executed trajectory of the autonomous vehicle and the recorded occupancies of other traffic participants.

Link: see media attachment of [2].

Video V8: Illustration of computation steps during a single planning cycle

This video introduces the necessary computation steps for the verification of an arbitrary intended trajectory (cf. Ch. 5). The cycles $c = 1$ and $c = 10$ of the urban intersection scenario are selected to visualize the obtained drivable area of the autonomous vehicle for times t'_k and the predicted occupancy sets of other traffic participants for time intervals $[t'_k, t'_{k+1}]$. Furthermore, simulations of the obtained trajectories \mathcal{T}_c and \mathcal{F}_c of the autonomous vehicle are shown.

Link: see media attachment of [2].

Video V9: Comparing the results of different intended planners This video demonstrates the verification results of the urban intersection and pedestrian scenario using different intended trajectory planners (cf. Ch. 5). In the first part of the video, the applied intended trajectory planners are briefly introduced. For the urban intersection and pedestrian scenario, a short video clip of the camera view during recording is given as well as a short description of the traffic situation. Then, for each intended planner, the recorded occupancies of other traffic participants and the executed trajectories of the autonomous vehicle are shown for all cycles c .

Link: see media attachment of [2].

Video V10: User study in driving simulator This video shows results from the user study in which we assess the comfort of the presented verification technique (cf. Ch. 5). We present selected scenarios and demonstrate the results when using the default and the supervised ACC.

Link: <https://ieeexplore.ieee.org/document/9091937/media>

Video V11: Intervention assessment study This video shows excerpts from the intervention assessment study (cf. Ch. 5). We present different traffic situations

from the conducted test drive and the planned fail-safe trajectories.

Link: see media attachment of [3] on *IEEE Xplore*.

A.9.2 CommonRoad scenarios

The following CommonRoad scenarios have been created within this research project. All scenarios are included in the CommonRoad benchmark suite and can be downloaded from the CommonRoad website *commonroad.in.tum.de*.

1. ZAM_Over-1_1
2. USA_US101-6_1_T-1
3. ZAM_HW-1_1_S-1
4. DEU_Ffb-2_2_S-1
5. ZAM_Intersect-1_2_S-1-2
6. ZAM_Urban-1_1_S-1
7. ZAM_Urban-2_1
8. ZAM_Urban-3_1
9. ZAM_Urban-6_1_S-1
10. ZAM_Urban-7_1_S-1
11. ZAM_Urban-4_1_S-1
12. ZAM_Urban-5_1_S-1
13. DEU_Muc-5_1_T-1
14. DEU_Muc-6_1_T-1
15. DEU_Gar-2_1_T-1