# Behavior Planning for Automated Highway Driving

## From Situation Analysis and Prediction Towards Resource-Constrained Interaction-Aware Lane Change Planning

DISSERTATION

submitted in partial fulfillment
of the requirements for the degree

Doktor-Ingenieur
(Doctor of Engineering)

in the

Faculty of Electrical Engineering and Information Technology
at Technische Universität Dortmund

by

Manuel Schmidt, M.Sc.

Heilbronn, Germany

Date of submission: March 15, 2022

First examiner: Univ.-Prof. Dr.-Ing. Prof. h.c. Dr. h.c. Torsten Bertram
Second examiner: Univ.-Prof. Dr.-Ing. Ferdinand Svaricek

Date of approval: November 9, 2022

# Preface

This thesis was written during my work as a research assistant at the Institute of Control Theory and Systems Engineering of the Faculty of Electrical Engineering and Information Technology at the TU Dortmund University. It is a result of a research project with ZF Automotive GmbH.
I want to thank Univ.-Prof. Dr.-Ing. Prof. h.c. Dr. h.c. Torsten Bertram provided me guidance along my journey. He took his time to advise me and encouraged me to use my potential and pursue scientific research in an application-oriented project. He provided me with opportunities to present my work at national and international conferences. His feedback was very valuable and crucial for the success of my work. The freedom he gave me allowed me to develop my own ideas and concepts.
Furthermore, I would like to thank Univ.-Prof. Dr.-Ing. Ferdinand Svaricek for agreeing to review my thesis as a second examiner, and I appreciate his valuable feedback and time.
I consider myself very fortunate to work with such a great team at the Institute of Control Theory and Systems Engineering. There is a unique feedback culture at the chair that I benefited enormously from. Especially apl. Prof. Dr. rer. nat. Frank Hoffmann helped me to shape my work and prevented me from running in the wrong direction. I thank Christian Wissing, Niklas Stannartz, Christian Lienke and Andreas Homann for the great teamwork during our work with ZF Automotive GmbH. Many great discussions left their mark on this thesis. Special thanks go to Martin Krüger and Christopher Diehl for the numerous lengthy discussions that greatly impacted my scientific work and understanding of automated driving software systems as a whole. Jan Braun provided invaluable feedback that increased the quality of my work, and I am more than grateful for that. Artemi Makarow always provided me a helping hand, and I am very grateful for that.
This work wouldn't be possible without the support of ZF Automotive GmbH. I thank Carlo Manna and Christian Wissing, who served as my supervisors, and Dr. rer. nat. Till Nattermann for providing me this great opportunity.
Finally, I want to thank my family for their unconditional support during all of the past years. Their encouragement motivated me every single day.

Heilbronn, March 2022 Manuel Schmidt

# Abstract

Automated driving becomes reality and promises enhanced comfort and safety for passengers. One of the first commercial uses will be highway situations since it is a more structured environment with reduced complexity. However, due to the high velocities, safety plays a crucial role.
This work covers certain components of an automated highway driving system with a strong focus on lane change behavior planning.
It starts with a description of a developed module for the generation of discretionary lane change proposals. This algorithm aims at increasing comfort for the passengers by proposing lane changes to the driver when, for example, a slower vehicle is spotted in the front to reduce breaking and travel time. The module's parameters are optimized using a driving simulator study. Finally, the results are evaluated on real data from a ZF Group test vehicle.
Trajectory predictions are necessary for effective lane change behavior planning. An approach from literature is adapted to allow the prediction of complete traffic situations. It is scalable and provides uncertainties that can be used in the subsequent lane change behavior planning step. The evaluation is done in simulation and on the publically available highD dataset.
One major contribution of this thesis is a method for analyzing traffic scenes in a spatiotemporal curvilinear coordinate frame. Polygon Clipping reduces the analysis to a set of geometric operations. The resulting polygons and their adjacency relations are represented using a graph.
Finally, three concepts are developed for lane change planning with increasing complexity and capability. The first one uses spatiotemporal geometry-informed sampling of cost-optimized splines. A second concept uses convex optimization in the form of quadratic programming. This approach results in very low runtimes and can be easily deployed on embedded platforms. It incorporates a novel formulation of hard safety constraints on Time-to-Collisions ($TTC$) and Time Headways ($THW$).
Both former approaches do not consider the multi-modal and uncertain nature of the predicted traffic situations and trajectories. Finally, the last approach is developed that uses a cooperative linear-quadratic game formulation. It accounts for those factors and provides a fallback strategy to ensure safe operation in critical traffic situations.
All behavior planning approaches are evaluated in simulation. Both optimization-based variants are furthermore evaluated on the highD dataset. A runtime estimation is based on real data from a ZF Group test vehicle and done within their software framework.

# Contents

# Nomenclature

**General Notation**

| | |
|---|---|
| $(\cdot)(k)$ | Value of $(\cdot)$ at discrete time step $k$ |
| $(\cdot)(t)$ | Value of $(\cdot)$ at continous time $t$ |
| $(\cdot)_{\text{lb}}$ | Lower boundary of quantity $(\cdot)$ |
| $(\cdot)_{\text{ub}}$ | Upper boundary of quantity $(\cdot)$ |
| $\ddot{X}$ | Second derivative of $X(t)$ |
| $\dot{X}$ | First derivative of $X(t)$ |
| $\hat{(\cdot)}$ | Estimate/Prediction of true value of $(\cdot)$ |
| $\lceil(\cdot)\rceil$ | Upper Gaussian brackets realizing a ceiling operation |
| $\mathbf{A}$ | Matrix |
| $\mathbf{a}$ | Vector |
| $\mathbf{A}^\intercal$ | Transpose of matrix $\mathbf{A}$ |
| $\mathbf{A}^{-1}$ | Inverse of matrix $\mathbf{A}$ |
| $\mathcal{A}$ | Set, polygonal chain, polygon, trigger, utility function or probability distribution |
| $\mathcal{A} \cap \mathcal{B}$ | Set intersection or polygon intersection of $\mathcal{A}$ and $\mathcal{B}$ |
| $\mathcal{A} \cup \mathcal{B}$ | Set union or polygon union of $\mathcal{A}$ and $\mathcal{B}$ |
| $\mathcal{A} \setminus \mathcal{B}$ | Set difference or polygon difference of $\mathcal{A}$ and $\mathcal{B}$ |
| $\mathcal{N}(\mu, \sigma^2)$ | Gaussian probability distribution with mean $\mu$ and variance $\sigma^2$ |
| $\mathcal{U}(a, b)$ | Uniform probability distribution on the interval $[a, b]$ |
| $\mathbb{1}(\cdot)$ | Indicator function |
| $\text{erfc}(a)$ | Complementary error function at point $a$ |
| $\text{erf}(a)$ | Error function at point $a$ |
| $\text{median}((\cdot))$ | Median of quantity $(\cdot)$ |
| $\text{trunc}(p(A), b, c)$ | Truncation of the continous probability density function $p(A)$ onto the interval $[b, c]$ |
| $\text{Var}((\cdot))$ | Variance of quantity $(\cdot)$ |
| $\overline{(\cdot)}$ | Mean of quantity $(\cdot)$ |
| $\Phi_{\mu,\sigma}(a)$ | Cumulative distribution function at point $a$ |
| $\tilde{(\cdot)}$ | Intermediate result or helper variable of $(\cdot)$ |
| $\{(\cdot)\}_a^b$ | Set of quantity $(\cdot)$ from $a$ up to $b$ |
| $a$ | Scalar |
| $P(a)$ | Probability of event $a$ |
| $p(A)$ | Continuous probability density function of the random variable $A$ |
| $P(A > a)$ | Probability of random variable $A$ being greater than $a$ |
| $X^{(n)}$ | n'th derivative of $X(t)$ |
| $\textcircled{i}$ | Polygon or node $i$ |

**Latin Letters**

| | |
|---|---|
| $(L, N)$ | Curvilinear coordinate frame |
| $(L, N, t)$ | Spatiotemporal curvilinear coordinate frame |
| $(X, Y)$ | Ego-vehicle coordinate frame |
| $(X, Y, t)$ | Spatiotemporal ego-vehicle coordinate frame |
| $(x, y, z)$ | Global coordinate frame |
| $(x, y, z, t)$ | Spatiotemporal global coordinate frame |
| $\mathbf{f}$ | Feature vector |
| $\mathbf{g}$ | Gradient vector |
| $\mathbf{H}$ | Hessian matrix |
| $\mathbf{M}$ | Random vector representing the multinoulli distributions used in the lane change prediction |
| $\mathbf{n}_\varrho$ | Normal vector used in pseudo distance calculation |
| $\mathbf{p}$ | Point or vector |
| $\mathbf{t}_i$ | Tangent vector $i$ used in pseudo distance calculation |
| $\mathbf{Z}$ | Random vector representing a latent variable |
| $\mathcal{P}$ | Polygon or polygonal chain |
| $\mathcal{T}$ | Trigger used in the discretionary lane change proposal module |
| $\mathcal{U}$ | Utility function used in the discretionary lane change proposal module |
| $\mathrm{O}_{(\cdot)}$ | Obstacle vehicle, further specified by the subscript $(\cdot)$ |
| $\mathrm{S}_{\mathrm{A,R}}$ | Left lane change scenario for the optimization of the accumulator trigger mechanism |
| $\mathrm{S}_{i,\mathrm{L}}$ | Left lane change scenario $i$ |
| $\mathrm{S}_{i,\mathrm{R}}$ | Right lane change scenario $i$ |
| $A$ | Spatiotemporal area |
| $C_L$ | $L$ coordinate of centroid of a spatiotemporal area |
| $C_t$ | $t$ coordinate of centroid of a spatiotemporal area |
| $d$ | Euclidean distance |
| $E$ | Kinetic energy |
| $f_{\boldsymbol{\gamma}}$ | Function parametrized by the parameter vector $\boldsymbol{\gamma}$ |
| $h$ | Step size for discretizations |
| $I$ | Interval |
| $J$ | Cost function used in optimizations |
| $n$ | Upper boundary of a counting variable |
| $T_{(\cdot)}$ | Lane change intention and proposal times, further specified by the subscript $(\cdot)$ |
| $t_{(\cdot)}$ | Time, further specified by the subscript $(\cdot)$ |
| $THW(T_\mathrm{q})$ | Time Headway at query time $T_\mathrm{q}$ |
| $TTC(T_\mathrm{q})$ | Time-to-Collision at query time $T_\mathrm{q}$ |
| $w_i$ | Weighting factor $i$ of utility $\mathcal{U}_i$ |

**Greek Letters**

| | |
|---|---|
| $\alpha_i$ | Longitudinal cost function parameter $i$ of the optimization-based lane change planning approach |
| $\beta_i$ | Lateral cost function parameter $i$ of the optimization-based lane change planning approach |
| $\boldsymbol{\chi}$ | Feature vector used in the discretionary lane change proposal module |
| $\boldsymbol{\gamma}$ | Parameter vector of a regression function |
| $\boldsymbol{\theta}$ | Parameter vector used in the Monte Carlo situation prediction |
| $\boldsymbol{\xi}$ | Parameter vector of the discretionary lane change proposal module |
| $\epsilon$ | Error of a regression |
| $\eta$ | Probability that a chance constraint should fulfil |
| $\gamma_i$ | Right lane change utility function parameter $i$ |
| $\kappa$ | Curvature of the road |
| $\lambda$ | Politeness or cooperation factor |
| $\mu_A$ | Mean of random variable $A$ |
| $\Psi$ | Heading or orientation angle |
| $\sigma_A^2$ | Variance of random variable $A$ |
| $\tau$ | Trajectory |
| $\theta_i$ | Longitudinal cost function parameter $i$ of the game theoretic lane change planning approach |
| $\varrho$ | Interpolation factor in the pseudo distance calculation and rational approximation of the error function |
| $\zeta$ | Accumulator leakage factor |

**Classification and Regression Metrics**

| | |
|---|---|
| $\mathrm{AC}(cl, \hat{cl})$ | Accuracy of class $cl$ given its prediction $\hat{cl}$ over a dataset |
| BAC | Balanced Accuracy |
| Exp. $\mathrm{Var}(a, \hat{a})$ | Explained Variance of quantity $a$ given its prediction $\hat{a}$ over a datasets |
| $\mathrm{MAE}(a, \hat{a})$ | Mean Absolute Error of quantity $a$ given its prediction $\hat{a}$ over a dataset |
| $\mathrm{MedAE}(a, \hat{a})$ | Median Absolute Error of quantity $a$ given its prediction $\hat{a}$ over a dataset |
| Prec. | Precision |
| Rec. | Recall |
| $\mathrm{RMSE}(a, \hat{a})$ | Root Mean Squared Error of quantity $a$ given its prediction $\hat{a}$ over a dataset |
| $\mathrm{R}^2(a, \hat{a})$ | Coefficient of determination of quantity $a$ given its prediction $\hat{a}$ over a dataset |

**Abbreviations and acronyms**

| | |
|---|---|
| ABS | **A**nti-lock **B**raking **S**ystem |
| ACC | **A**daptive **C**ruise **C**ontrol |

| | |
|---|---|
| ALGAMES | **A**ugmented **L**agrangian **GAME**-theoretic **S**olver |
| CARLA | **CAR** **L**earning to **A**ct |
| CB | **C**urrent ego-vehicle lane **B**ack |
| CF | **C**urrent ego-vehicle lane **F**ront |
| CNN | **C**onvolutional **N**eural **N**etwork |
| CommonRoad | **Com**posable benchmarks for **m**otion planning **on Road**s |
| CPT | **C**umulative **P**rospect **T**heory |
| CS-LSTM | **C**onvolutional **S**ocial Pooling **L**ong **S**hort-**T**erm **M**emory |
| CV | **C**onstant **V**elocity |
| CV-LK | **C**onstant **V**elocity with **L**ane-**K**eeping |
| DESIM | **D**riving **E**nvironment **SIM**ulation |
| DTW | **D**ynamic **T**ime **W**arping |
| ECU | **E**lectronic **C**ontrol **U**nit |
| ESP | **E**lectronic **S**tability **P**rogram |
| ET | **E**xtra **T**rees |
| FCN | **F**ully **C**onvolutional **N**etwork |
| GBT | **G**radient **B**oosted **T**rees |
| highD | **high**way **D**rone dataset |
| IDM | **I**ntelligent **D**river **M**odel |
| IPOPT | **I**nterior **P**oint **OPT**imizer |
| IRL | **I**nverse **R**einforcement **L**earning |
| KDens | **K**ernel **Dens**ity |
| LB | **L**eft adjacent lane **B**ack |
| LCL | **L**ane **C**hange **L**eft |
| LCP | **L**ane **C**hange **P**rediction |
| LCR | **L**ane **C**hange **R**ight |
| LF | **L**eft adjacent lane **F**ront |
| LK | **L**ane-**K**eeping |
| LQ | **L**inear-**Q**uadratic |
| LSTM | **L**ong **S**hort-**T**erm **M**emory |
| LUCIDGames | on**L**ine **UnsC**ented **I**nverse **D**ynamic **Games** |
| MATS | **M**ixtures of **A**ffine **T**ime-varying **S**ystems |
| MIQP | **M**ixed-**I**nteger **Q**uadratic **P**rogramming |
| MITSIM | **MI**croscopic **T**raffic **SIM**ulator |
| MOBIL | **M**inimizing **O**verall **B**raking Deceleration induced by **L**ane **C**hanges |
| OSQP | **O**perator **S**plitting Solver for **Q**uadratic **P**rograms |
| OTP | **O**ther **T**raffic **P**articipants |
| POMDP | **P**artially **O**bservable **M**arkov **D**ecision **P**rocess |
| RB | **R**ight adjacent lane **B**ack |
| RF | **R**ight adjacent lane **F**ront or **R**andom **F**orest |
| ROI | **R**egion **O**f **I**nterest |
| RSS | **R**esponsibility-**S**ensitive **S**afety |

| | |
|---|---|
| RVM | **R**elevance **V**ector **M**achine |
| SAE | **S**ociety of **A**utomotive **E**ngineers |
| SB | Lane change **S**tart lane **B**ack region of interest |
| SF | Lane change **S**tart lane **F**ront region of interest |
| STL | **S**ignal **T**emporal **L**ogic |
| SVM | **S**upport **V**ector **M**achine |
| SVO | **S**ocial **V**alue **O**rientation |
| TB | Lane change **T**arget lane **B**ack region of interest |
| TF | Lane change **T**arget lane **F**ront region of interest |

# 1 Introduction

This Chapter starts with a motivation and an outline of the thesis at hand. Afterward, the related work is discussed. Finally, the last Section gives an overview of the contributions.

## 1.1. Motivation

Safety is of utmost importance and a key driver of the recent developments in automated driving. Figure 1.1 shows the normalized value of road traffic fatalities over registered vehicles in Germany for the years 1960 to 2019. The year 1960 is chosen as the reference. Several important regulatory events are also included in the figure. There is a clear trend towards fewer road traffic fatalities per registered vehicle. The initial strong downward trend is potentially due to imposing a 50 km/h speed limit when driving within towns and cities in Germany. Those imposed rules and laws made driving safer. The mandatory introduction of driver assistance systems such as ABS (**A**nti-lock **B**raking **S**ystem) in 2004 and ESP (**E**lectronic **S**tability **P**rogram) in 2014 aim to reduce the remaining fatalities further. Similarly, the next evolution of those systems are advanced driver assistance systems such as lane-keeping and ACC (**A**daptive **C**ruise **C**ontrol) and even more advanced automated driving systems. The SAE (**S**ociety of **A**utomotive **E**ngineers) classifies driver assistance and automated driving systems in a total of six levels, refer to Figure A.1 in the appendix A.1. Recently, the term SAE Level 2+ gained some usage, in addition to the six official SAE levels. The plus usually reflects the addition of automated lane changes compared to Level 2 systems that generally provide in-lane driving functionality, refer to [ZF20].
The thesis at hand provides modules concerning situation analysis and automated lane change functionality for such Level 2+ systems. For commercial useability, a set of constraints are imposed. The following gives an overview of the requirements that should be met.

**Functional and Safety Relevant Requirements:**

- Situation- and Interaction-awareness for enhanced comfort and safety
- High interpretability of the algorithms

**Industrialization Requirements:**

- Low runtime (typically below 40 ms per algorithm cycle)

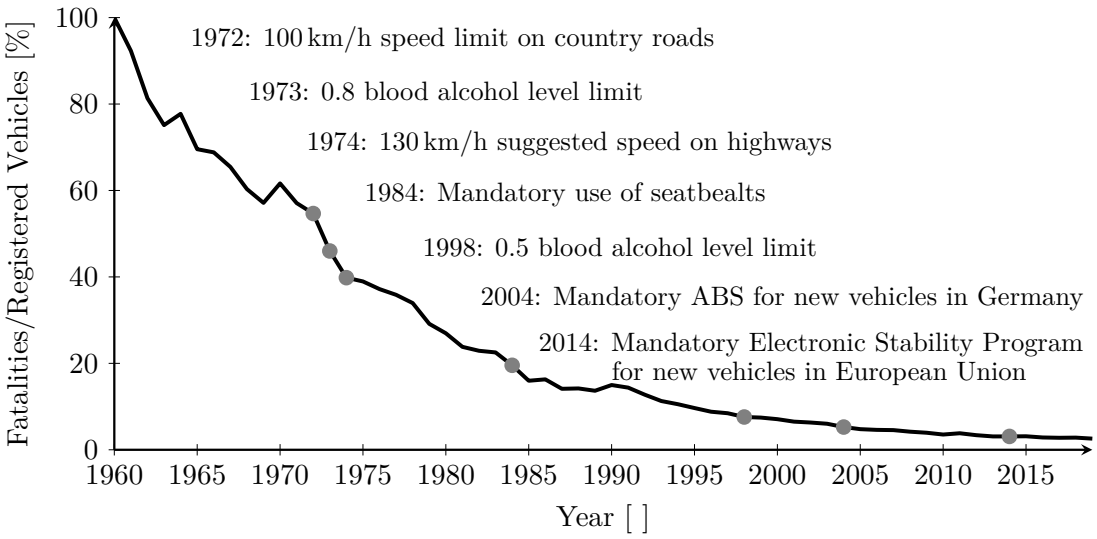



Figure 1.1.: Road traffic fatalities in Germany normalized by the number of registered vehicles for the time between 1960 and 2019. Important regulatory events are highlighted. Data taken from [Fed20a] and [Fed20b].

- Low memory consumption, suitable for operation on current generation embedded automotive hardware platforms

These requirements motivate the design choices and simplifications employed and discussed in the remainder of the thesis.

## 1.2. Outline

The thesis at hand deals with automated lane changes on highways. An overview of the content is shown in Figure 1.2. An environment model and the ego-vehicle state are the inputs. The thesis first provides a situation analysis module that decides if a lane change is beneficial in increasing comfort and decreasing travel time. The developed module takes uncertainties into account and is a fully interpretable model. Such lane changes are called discretionary lane changes in contrast to mandatory ones used for route-following purposes. Trajectory predictions are of high importance to further increase the passenger's comfort. Having a good knowledge of the future development of the traffic scene is especially crucial for comfortable lane change planning of the ego-vehicle. To this end, the thesis at hand describes adaptions of an interpretable and scalable trajectory prediction module from literature. It allows to predict traffic situations and a planned ego-vehicle trajectory can be incorporated into it to allow for counterfactual reasoning. Such functionality is crucial, as is explained in subsequent Chapters of the thesis at hand. After obtaining situation predictions, the traffic situation can be analyzed, and possible lane change maneuvers of the ego-vehicle can be identified. A novel approach for maneuver identification using Computational Geometry in the form of Polygon Clipping is proposed. It is shown that using a curvilinear coordinate frame enables such analysis resulting finally in a graph structure that encodes all relevant information for the actual lane change trajectory planning. Finally, a total of three approaches are described for the synthesis of

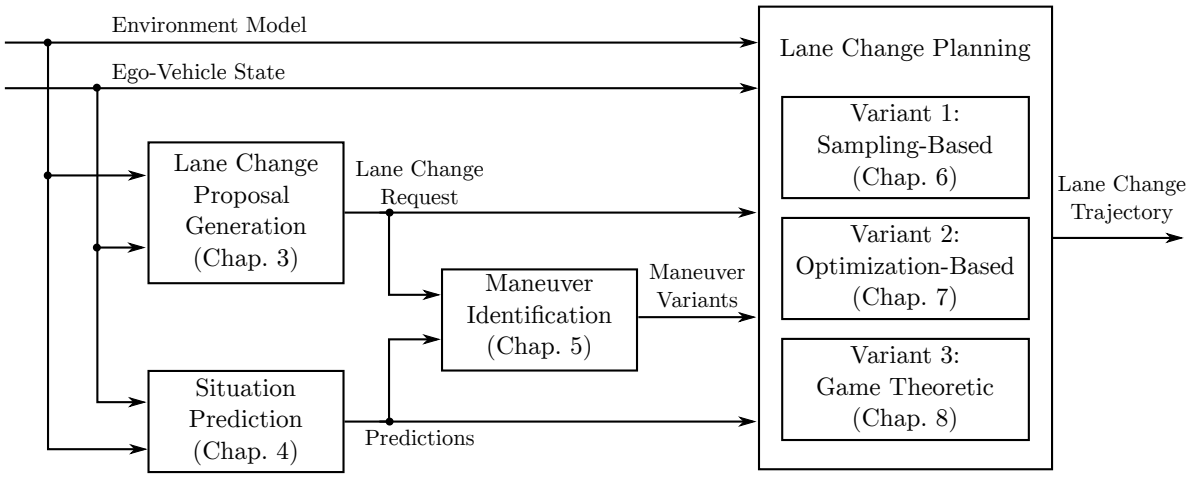



Figure 1.2.: Overview of the modules developed or adapted within the thesis at hand and the respective Chapters.

optimized lane change trajectories. The first is based on the geometry-informed sampling of spline knots. The second approach frames the problem as a Quadratic Program that is solved efficiently, allowing for easy deployment onto embedded platforms and current generation vehicle ECUs (**E**lectronic **C**ontrol **U**nit). One significant contribution is the integration of affine Time-to-Collison and Time Headway safety constraints into the optimization. Former approaches are to a certain degree interaction-aware but use a prior situation prediction to plan the ego-vehicle lane change trajectory. Thus, situation prediction and lane change planning are intertwined. Therefore, the last approach presented in the thesis at hand frames the problem as a cooperative differential linear-quadratic game. This formulation again results in a Quadratic Program that can be solved efficiently. In addition, integration of chance constraints allows handling uncertainties in the situation predictions and multimodality. Finally, a fallback strategy is integrated to ensure that a feasible trajectory always exists. The following list provides an overview of all subsequent Chapters of the thesis at hand and their respective contents.

**Chapter 2:** The functional architecture of automated vehicles is first introduced. After that, the problem of lane change behavior planning is described, and the taxonomy is introduced as used throughout the thesis at hand. The curvilinear coordinate system and Polygon Clipping are introduced due to their importance for the remainder of the thesis. Finally, the development tools and simulation environment are described.

**Chapter 3:** A novel approach for the generation of discretionary lane change proposals is described. Its architecture is discussed. The module includes certain parameters that are optimized using a driving simulator study. The experimental evaluation is conducted using real data from a test vehicle.

**Chapter 4:** This Chapter introduces the situation prediction that serves as a basis for the subsequent lane change planning. Instead of predicting trajectories on an individual traffic participant basis, the whole situation is predicted. The incorporation of a

desired velocity regression is discussed. Finally, the approach is evaluated using simulation data and the publically available highD (**high**way **D**rone) dataset, refer to [Kra+18].

**Chapter 5:** After transforming the traffic situation into a curvilinear coordinate frame, it is possible to analyze it spatiotemporally. This Chapter shows how the ego-vehicle's potential lane change maneuver options can be identified in a principled way.

**Chapter 6:** The first of a total of three lane change planning approaches are introduced in this Chapter. It is shown how the maneuver identification results of Chapter 5 can be used for informed sampling of cost-optimized spline knots. The approach is made interaction-aware by incorporating optimized ego-vehicle lane change trajectories into the situation prediction. The influence of the lane change on the surrounding traffic participants can therefore be analyzed.

**Chapter 7:** While sampling-based approaches have several benefits, the major drawbacks are their suboptimality, runtime, and memory consumption when using current generation vehicle ECUs. For the application of behavior planning on embedded hardware, a Quadratic Programming baseline approach is presented. It is also shown how Time-to-Collison and Time Headway safety constraints can be incorporated into the optimization problem.

**Chapter 8:** The subdivision of situation prediction and lane change planning is disadvantageous. Traffic participants might show courtesy in various traffic situations. Therefore, this Chapter proposes formulating the lane change planning problem in terms of a cooperative linear-quadratic game. The uncertainty of situation predictions is considered using chance constraints. A simple fallback mechanism is furthermore integrated. The quantitative evaluation is done in simulation and on the highD dataset. A runtime estimation is provided based on real data from a ZF Group test vehicle.

**Chapter 9:** This Chapter concludes the thesis. The developments are reflected and an outlook towards promising future research directions is given.

## 1.3. Related Work

The following discussion of the related work is divided into the areas of discretionary lane change proposals, trajectory, situation prediction, and lane change behavior planning.

#### Discretionary Lane Change Proposal Generation

Chapter 3 aims at discretionary lane changes. Discretionary lane changes are mainly done to increase passengers' comfort or gain speed to travel with a specified desired velocity. In contrast, mandatory lane changes are done for following a predefined route. Much research in this area is conducted in the context of microscopic traffic simulations.
[Gip86] is the extension of the car-following model presented in [Gip81]. It models the lane change decision using certain rules and also considers, for example, safety aspects

and the urgency of a lane change. [YK96] presents the lane change model that is used in MITSIM (**MI**croscopic **T**raffic **SIM**ulator) and extends [Gip86]. [McD+97] models the lane change decision process using fuzzy logic. [Bra+98] presents a calibration of the model described in [McD+97] and shows a strong correlation between a collected dataset and their proposed lane change model. Furthermore, the work [Wu+03] validates the model in greater detail. [El +01] focuses on conflict resolution and describes a lane change model based on defined ROIs (**R**egions **O**f **I**nterests) around the ego-vehicle. [Hid02] uses a rule-based lane change model and distinguishes forced and cooperative lane changes. [Tol+03] aims at integrating mandatory and discretionary lane change decisions and introduces lane-specific utility functions that are based on the traffic situation around the ego-vehicle. The extension and calibration of the model are presented in [Tol+09].

[Kes+07] presents the lane change model MOBIL (**M**inimizing **O**verall **B**raking deceleration **I**nduced by **L**ane Changes) which aims at minimizing the overall braking of traffic participants induced by the ego-vehicle lane change. The model also accounts for courtesy by introducing a politeness factor to weigh the ego-vehicle utility of a lane change compared to the utility of all remaining surrounding vehicles in the traffic situation. It extends the **I**ntelligent **D**river **M**odel (IDM) [Tre+00]. [Sch+12] proposes a model that includes the relaxation and synchronization phenomena during lane changes. Relaxation refers to a phase after a lane change where the traffic participants accept temporarily lower safety margins. In contrast, synchronization refers to the longitudinal adaption of a vehicle towards a target gap. A comprehensive survey of lane change models used in microscopic traffic simulations is presented in [Mor+10]. The authors also introduce a taxonomy for an easier distinction of the various models. Another more recent survey is [Zhe14]. The authors of [Ard+12] present a probabilistic approach for highly automated driving on highways with a special focus on lane change functionality. Their work explicitly considers sensor noise and derives a lane change decision using a utility function.

### Trajectory and Situation Prediction

[Sch+14] proposes a probabilistic method for vehicle-individual trajectory predictions based on Gaussian Mixtures. In [KA17], an interactive prediction of road user free space occupancy is presented. The approach is based on reachability analysis. [Rös+17] describes an approach for integrated trajectory planning and prediction in the context of proxemics. The approach uses Learning from Demonstrations to determine weighting factors of the global cost function and explicitly considers topological information. The work [Wis+18] takes a step towards interactive trajectory prediction and proposes the use of Monte Carlo simulations based on driver models from microscopic traffic simulations. However, the trajectory prediction is made on a vehicle-individual basis.

Using a grid-based discretization and an LSTM (**L**ong **S**hort-**T**erm **M**emory) encoder-decoder Neural Network, a multimodal trajectory prediction approach is proposed in [DT18]. The multimodality is realized by maneuver classification within the network. For situation prediction, a grid-based discretization is applied in [Sch+19h]. This allows the use of semantic information. Finally, a **F**ully **C**onvolutional **N**etwork (FCN) is used to predict the overall situation.

[Hu+19b] presents a combined learning- and planning-based approach for trajectory pre-

diction. The approach can account for irrational human behavior. [Lee+19] models pairwise interactions between road users to finally execute trajectory predictions with a horizon of 5 s using a Graph Convolutional Neural Network. [TS19] uses ideas from Probabilistic Graphical Models in Deep Learning for interactive situation prediction. Here, a latent variable is introduced to account for different driving styles.

In [Sch+19i], a Neural Network is used to predict Dynamic Occupancy Grids. The work [Sri+19] proposes a Deep Learning based approach for trajectory prediction. By explicitly using semantic information, the approach shows good generalization ability. Trajectory predictions are conditioned on high-level goals in [Rhi+19]. [Gao+20] proposes the VectorNet architecture. It is a hierarchical graph-based Neural Network for interactive situation prediction. The network uses a vectorized high-accuracy map and the trajectory history of road users as input data. [Zha+20] builds upon VectorNet and proposes a neural network for destination-based trajectory prediction.

[Zen+20] is a continuation of [Zen+19] and describes an approach for multimodal trajectory prediction as well as trajectory planning for the ego-vehicle towards risk minimization. However, in the context of trajectory prediction, the ego-vehicle is not treated equally here, and the interaction of ego-planning and situation prediction is thus neglected. The contribution [Krü+19a] deals with probabilistic trajectory prediction and uses a combination of neural network and a Gaussian Process Regression. Through this combination, it is possible to predict the uncertainty of the prediction and the driving maneuver.

A concept based on 3D Convolutional Neural Networks is proposed in [Krü+20]. Here, the environment of the ego-vehicle is represented by a sequence of potential fields in the form of grayscale images. The resulting three-dimensional tensor represents the input of a Convolutional Neural Network for vehicle-individual trajectory prediction. There is currently a rise in the use of Graph Neural Networks for forecasting applications since this model type represents a way to include inductive bias into predictions. To this end, [Sal+20] proposes an architecture that uses a graph-structured recurrent model.

### Lane Change Planning

In [Ben+14], the Lanelet format for high-accuracy maps and an associated distance transformation is introduced. This allows intuitive and easy generation of maps using publically available software. The Lanelet distance transformation has been widely adopted in the areas of trajectory planning and prediction. The trajectory planner used for the Bertha-Benz drive is described in [Zie+14]. It is a local trajectory planner based on non–linear optimization. Particular emphasis is placed on avoiding collisions. In [SP14], a sampling-based concept for cooperative ego-vehicle planning is proposed. This includes a mechanism to resolve conflicts recursively. The paper [Wei+14] proposes a sampling-based lane change planner. It also incorporates interactions with other road users by modelling them using ACC (**A**daptive **C**ruise **C**ontrol). [Che+15] presents an incremental planning approach in spatiotemporal representation. The focus is on the development of a suitable search heuristic. [Ben+15] also uses a spatiotemporal representation in a curvilinear coordinate system and shows the existence of homotopy classes in the form of different maneuver options.

[UM15] proposes a real-time capable method based on policy iteration and thus POMDPs

(**P**artially **O**bservable **M**arkov **D**ecision **P**rocess). Adequate simplifications can ensure Real-time capability. Cooperations are considered in the approach of [Len+16] in the context of a Monte Carlo Tree Search. The cost function is chosen to relate the cost of the ego-vehicle to the cost of the surrounding road users. [Sch+16] uses the spatiotemporal representation from [Ben+15] and develops a sampling-based trajectory planning approach that exploits the differential flatness of the single-track kinematic vehicle model. In [Nil+16b], lane change processes are studied in detail. The concept explains how the need for a lane change arises from the initial situation and how it should ultimately be executed. A model predictive formulation is proposed. The underlying optimal control problem represents a Quadratic Program. [Qia+16] accounts for the aspect of different maneuver options and proposes a **M**ixed-**I**nteger **Q**uadratic **P**rogramming (MIQP) problem formulation for it. In [Eve+16], an interactive maneuver planning is performed similar to [Wei+14]. Other road users are modeled using the Intelligent Driver Model (IDM) [Tre+00]. Simultaneous consideration of two maneuver options in interSection situations is described in [Zha+16]. Here, a joint optimization problem for both maneuver options is formulated. [Imb+17] investigates human decision behavior at a narrowing road. The key question here is whether humans can resolve this conflict when an automated vehicle is opposite them.

The paper [Ad17] deals with situation analysis in a spatiotemporal curvilinear coordinate system and proposes a graph structure to represent the results. In [MA17], a cooperative conflict resolution strategy is proposed using the tool of reachability analysis. Conflicting domains are divided among agents to resolve the conflict. [Bur+17] introduces a taxonomy for the cooperative behavior of road users. [Mil+18] develops a MIQP for longitudinal and lateral trajectory planning using the Big-M concept, refer to [GH01]. [Men+18] introduces the concept of cooperative planning for highway on-ramps.

[MA18] describes a concept for the coordination of a plurality of road users via allocation of drivable space. A similar concept is presented in [SA18]. There, efficient computation of the drivable space using reachability analysis for dynamic traffic scenes is proposed. The paper [Tas+18] has a similar focus as [Zha+16] and also formulates an optimization problem for simultaneous optimization of different maneuver options. This allows delaying the decision for a maneuver option. In [Kur+18], similar to [Len+16], a Monte Carlo Tree Search is used for decentralized cooperative planning. [Sun+18] integrates a politeness term into the cost function for trajectory planning of automated vehicles. This is to ensure that the planned trajectories are neither too conservative nor too aggressive.

[PA18] presents a concept for real-time generation of trajectories with safety guarantees. The approach uses convex optimization and is evaluated using the CommonRoad benchmark (**Com**posable benchmarks for **m**otion planning **on Road**s, refer to [Alt+17]). In [Sto+19], the importance of the complexity of the traffic situation for the willingness of road users to cooperate is investigated. A model predictive trajectory planning approach is presented in [Yi+19]. The approach partitions the spatiotemporal free space and uses successive linearization of the nonlinear single-track model to ensure real-time capability. [Fis+19] presents a game-theoretic approach to trajectory planning. Specifically, a hierarchical formulation is made as a dynamic game. Interactions between road users can thus be efficiently taken into account. In [Klo+19], a decentralized model predictive control for the coordination of multiple vehicles is proposed. Moreover, the paper discusses a

concept for determining the priority of road users. A POMDP-based concept that can also handle occlusions is presented in [Hub+19]. Here, the problem is solved by Monte Carlo sampling for action strategy optimization.
In [Est+19] an approach for the identification of maneuver options as well as their verification is presented. The representation is done using a graph structure, and formalized rules are used for verification. The approach described in [Ge+19] presents risk-sensitive trajectory planning for automated vehicles. The effect of the automated vehicle's intention on surrounding road users is modeled probabilistically and uses the value-at-risk concept. [Hu+19a] considers merge-in maneuvers in different traffic scenarios. It is a learning-based and interactive approach using the concept of Curriculum Learning and thus Reinforcement Learning. [Sun+19] uses CPT (**C**umulative **P**rospect **T**heory) to model irrational human behavior. The paper proposes a hierarchical learning procedure and evaluates it in merge-in scenes.
[Sch+19j] formulates automated driving as a non-cooperative dynamic game. A Nash equilibrium is computed simultaneously for all agents using nonlinear optimization. The cooperative behavior of the surrounding road users is estimated online using a concept called SVO (**S**ocial **V**alue **O**rientation) for weighting of the joint cost function. [Nau+19] uses the RSS (**R**esponsibility-**S**ensitive **S**afety) framework from [Sha+17] to compute lane change trajectories with safety guarantees.
The paper [Tan19] uses Deep Reinforcement Learning with Self-Play to learn robust sequential decision making. The evaluation is performed in the context of a merge-in scene with a plurality of road users. Here, negotiation among the participants becomes necessary to resolve the conflicting situation. [Zen+19] proposes an end-to-end trajectory planner. Lidar raw data and a highly accurate map serve as input data, and the approach considers different maneuver options of the ego-vehicle.
[Sto+19] analyzes factors influencing the cooperation of road users during lane change operations. The results show that, for example, the criticality of the situation is a decisive factor for cooperation. [Nau+20] investigates the usability of different cost functions for modeling human driving behavior. ALGAMES (**A**ugmented **L**agrangian **GAME**-theoretic **S**olver) is proposed in [Cle+20a] that identifies Generalized Nash Equilibrium strategies and can be applied with a receding horizon. Their evaluation shows the efficacy in on-ramp merging scenarios in automated driving. [Cle+20b] proposes LUCIDGames (on**L**ine **U**ns**C**ented **I**nverse **D**ynamic **Games**), an inverse optimal control algorithm that estimates other agent's objective functions online. They combine LUCIDGames with ALGAMES to solve the resulting optimization problem and evaluate it.
The work [Sch+21a] develops a trajectory optimization algorithm that utilizes gradients from Deep Generative human trajectory prediction models. Their approach, therefore, explicitly accounts for interactions and yields proactive behaviors. Finally, [Sch+21b] combines set-based reachability analysis with optimal motion planning in arbitrary traffic situations and allows the use of arbitrary vehicle dynamics models.
Most of the above state-of-the-art approaches do not fulfill all requirements formulated in Section 1.1 that motivate the thesis at hand. Therefore, the goal is to propose Pareto-optimal concepts fulfilling all of the requirements to maximize the utility of the algorithms on current generation vehicle ECUs.

Besides above mentioned approaches from literature, Tesla, Inc. already offers its customers a lane change functionality within their Full Self-Driving package [Tes22]. During their live-streamed event Tesla AI Day 2021 (refer to [Tes21]), a high-level description of their lane change planning approach was provided. A two-stage approach is used. First, a discrete search is used via sampling of trajectory candidates. This stage is similar to the approach presented in Chapter 6 of the thesis at hand. After this, a convex approximation of the driving corridor is dervied and a continous optimization conducted to obtain the optimal trajectory. In comparison, Chapter 5 of the thesis at hand also describes an approach to obtain convex driving corridors that are subsequently used in Chapter 7 within a continous trajectory optimization. During their presentation, it is also mentioned that in certain situations a joint optimization of two agents needs to be conducted in order for the vehicle to decide for a specific option. This aspect is similar to the described game theoretic approach presented in Chapter 8 of the thesis at hand. The specific details of their software architecture and algorithms are however to this date not known to the public.

## 1.4. Contribution

The contributions of the thesis at hand are structured according to the related work Section. Hence, the contributions in the areas of discretionary lane change proposals, trajectory and situation prediction, and lane change planning are discussed.

**Discretionary Lane Change Proposal Generation**

The contributions of the module, described in Chapter 3, are the following and build upon the concept [Ard+12]. First, a politeness factor is introduced in the utility calculation motivated by [Kes+07]. Second, the desired velocity of the ego-vehicle is modeled using a Gaussian probability distribution, taking into account the passenger's acceptance of certain deviations from the desired velocity. Third, an accumulation mechanism is developed such that long-lasting slight dissatisfaction with the current driving lane of the ego-vehicle is accumulated, eventually leading to a lane change decision. Finally, compared to [Ard+12] an in-depth parameter optimization and analysis of the results using a driving simulator study is provided. This contribution first appeared in the author's publication [Sch+21c] and led to the patent [Nat+20c].

**Trajectory and Situation Prediction**

The developed situation prediction, described in Chapter 4, builds upon the existing concept [Wis+18]. It is an approach based on Monte Carlo simulations for highway scenes based on models from microscopic traffic simulations. It is a vehicle-individual trajectory prediction that leads to the prediction of collisions in certain traffic situations. Furthermore, handling multimodality is difficult due to a potential combinatoric explosion. The thesis at hand modifies the approach to become a situation prediction. This is done by topological clustering of the overall situation. Multimodality is reflected in topologically different situation developments and represented using a single latent random variable.

### Lane Change Planning

There are numerous scientific contributions concerning lane change planning. Firstly, Polygon Clipping is introduced for the identification of distinct lane change maneuvers on highways in Chapter 5. The use of Polygon Clipping allows treating the problem in terms of geometric operations. The traffic scene topology is based on relations between the obtained polygons and represented using a graph. It is shown how this approach can handle lane changes of surrounding traffic participants while being runtime efficient. The works [Sch+16], [Ad17] and [Est+19] are closely related but either handle the problem differently or don't discuss lane changes of surrounding traffic. The contribution described in Chapter 5 first appeared in the author's publications [Sch+19e], [Sch+19f], [Lie+19a] and led to the patents [Wis+20a], [Man+20], [Nat+20b] and [Nat+20a].
Chapter 6 proposes a sampling-based approach for lane change planning and shows how the reachable set of the ego-vehicle and geometric properties can be exploited for the generation of sample points that represent knots of cost-optimized splines. Those splines are described in [Ric+16] for use in quadrotor flight. Their formulation is very general, and the thesis at hand describes the application of the splines for the trajectory optimization of automated vehicles. It is also shown how interaction-awareness is realized by incorporating the optimized ego-vehicle trajectory into the situation prediction. Finally, a novel interaction cost based on generalized kinetic energies that is easy to interpret and parametrize is developed to quantify the effects. The contribution described in Chapter 6 first appeared in the author's publications [Sch+19e] and [Lie+19a]. It furthermore led to the patent [Sch+20c] and additionally inspired the patent [Wis+20a].
Chapter 7 frames the lane change planning problem as a Quadratic Program to circumvent some of the limitations of the sampling-based approach. It serves as a baseline and uses mainly ideas and terminology from [Nil+17]. The core contribution is integrating Time-to-Collision and Time Headway requirements into the optimization using affine constraints based on spatiotemporal geometric relations. It first appeared in the author's publication [Sch+19f].
Chapter 8 presents an extension of the optimization-based behavior planning approach described in Chapter 7. It is strongly inspired by the game theoretic approach presented in [Sch+19j] but tailored to the requirements formulated in the motivation of the thesis at hand. The situation prediction and behavior planning problem are intertwined. While [Sch+19j] formulates the problem in terms of a non-cooperative differential game, the thesis at hand uses a cooperative linear-quadratic game formulation resulting in a Quadratic Program. This results in less conservative behavior of the ego-vehicle compared to the baseline approach of Chapter 7 as it can anticipate cooperation of surrounding traffic participants. A special research question arises for the consideration of uncertainties. In particular, the uncertainty due to multimodality. Chance constraints are integrated into the optimization problem to tackle this. The timing of a lane change is highly important, and this Chapter proposes a geometry-inspired outer optimization for its specification. A fallback system is constructed to enhance safety and robustness further. Finally, Inverse Reinforcement Learning is used to specify the cost function's parameters using the highD dataset, and the results are further analyzed.

# 2
# Fundamental Background

This Chapter first introduces a taxonomy to clarify some frequently used terms and provides the notational convention. It then aims to convey a fundamental understanding of automated driving software systems and introduces the specific problems the thesis at hand deals with. Next, some theoretical background on curvilinear coordinate frames and Polygon Clipping is given. Finally, the chosen simulation environment and development tools are described.

## 2.1. Taxonomy and Notation

Within the thesis at hand, the definitions of the term scene, situation, and scenario given in [Ulb+15] are adopted and restated here for easy reference. [Ulb+15] defines the terms as follows:

**Scene:** *A scene describes a snapshot of the environment including the scenery and dynamic elements, as well as all actors' and observers' self-representations, and the relationships among those entities. Only a scene representation in a simulated world can be all-encompassing (objective scene, ground truth). In the real world it is incomplete, incorrect, uncertain, and from one or several observers' points of view (subjective scene).*

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

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

Figure 2.1 shows an overview of an automated driving software system. On a high level, it is divided into several layers. These are perception and prediction, the strategic, tactical,

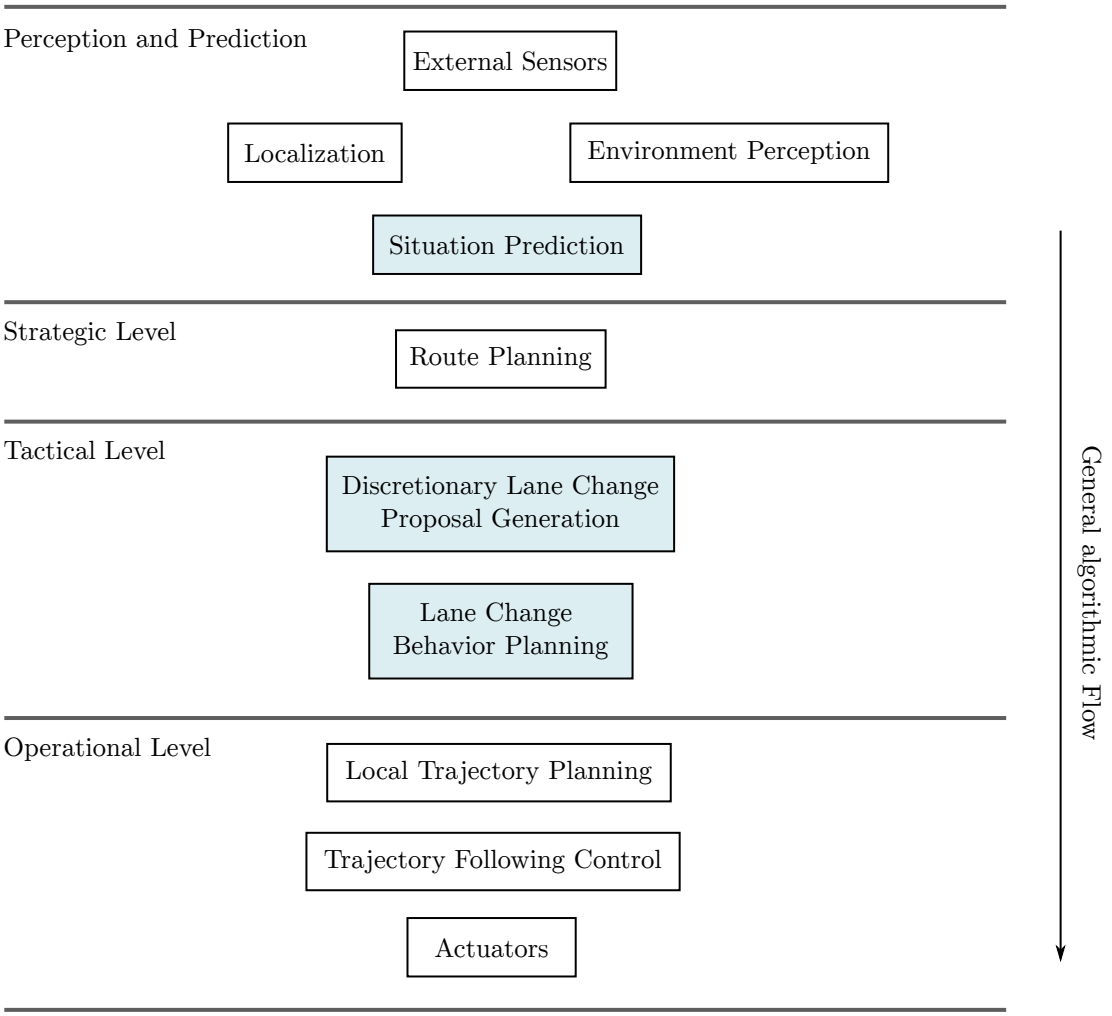



Figure 2.1.: Overview of the modules in a typical automated driving software system. The light-blue boxes are discussed within the thesis at hand.

and finally operational level. The general algorithmic flow is from top to bottom. Cross connections between the various modules are hidden to reduce complexity. The Figure only provides a general overview. Within the perception and prediction layer, external sensors provide raw data used in an environment perception system to create a scene description, often called the environment model. It typically encompasses at least lane marking information, static and dynamic objects. A localization provides information about the location of the ego-vehicle in a global or local coordinate frame.
Chapter 4 describes a situation prediction approach. Its goal is to enhance the scene description by predicting all dynamic objects' future states. According to the above definition of the terms, such prediction incorporates behaviors and intentions of traffic participants and is hence called situation prediction. The strategic level consists of a route planning module that imposes a route-following behavior on the ego-vehicle by providing mandatory lane change requests. In contrast to mandatory lane changes for route-following, the tactical level deals first with discretionary lane changes. Such lane changes aim to increase driver comfort and reduce travel time. This is discussed in Chapter 3 in more detail. The second module on the tactical level is the central focus

of the thesis at hand, the lane change behavior planning. It is important to distinguish behavior planning from local trajectory planning clearly. Hence the following definition is adopted within the thesis at hand:

**Behavior Planning:** *Behavior planning belongs to the tactical level of an automated driving software system. It provides high-level behaviors in the form of a temporal specification. Typically such specification will decide between various options that the ego-vehicle has. Sometimes those options are called homotopy classes or maneuver options. Its output is a trajectory that can initialize a local trajectory planning approach defined next. Hence behavior planning acts as a globalization strategy in contrast to local optimal trajectory planning approaches. Sometimes the terms maneuver planning and decision-making are used in science and industry instead of behavior planning. Maneuvers considered are lane-keeping or lane changes to either the left or right adjacent lane. There can be various maneuver options for lane changes, depending on how it is conducted.*

**Local Trajectory Planning:** *Local trajectory planning refers to locally optimal approaches within a given homotopy class or for a given maneuver option. The initial trajectory provided by a behavior planning module might be altered due to accurate consideration of vehicle dynamics and nonholonomic constraints.*

Note that the coupling of scene prediction, lane change behavior, and local trajectory planning is discussed in more detail in [Lie+19b]. Finally, the locally optimal trajectory is provided to a trajectory following controller that creates control actions using the vehicle's actuators. The above description reflects a typical modular and hierarchical automated driving software system. Other design choices are undoubtedly possible. The thesis at hand, however, contributes to such a modular system. The light-blue modules in Figure 2.1 are the particular focus here.

Regarding the mathematical notation, the thesis at hand aims for readability and comprehensibility by reducing the use of super- and subscripts to a minimum without sacrificing mathematical accuracy. In the following, some notational conventions are described and used in the remainder of the thesis.
A total of four coordinate frames are used, with tuples referencing them. The global coordinate frame is denoted using lowercase letters as the tuple $(x, y)$, and its origin is typically defined within a simulation environment or dataset. The ego-vehicle coordinate frame is denoted using uppercase letters as $(X, Y)$. Similarly, a curvilinear coordinate frame is denoted as $(L, N)$. All mentioned coordinate systems are assumed stationary at $t = 0\,\mathrm{s}$. They can be extended by the time dimension, leading to spatiotemporal variants. Throughout the thesis, the frames $(x, y, t)$, $(X, Y, t)$ and $(L, N, t)$ are used.
A lowercase letter $t$ is used for time as an independent variable and specific timings, whereas an uppercase $T$ lane change intention and proposal times. Discrete timesteps are denoted using the variable $t_k$ or occasionally simply $k$. Derivatives are denoted using dots, for example, $\dot{X}(t)$ and $\ddot{X}(t)$ in case of the first and second derivative of the time-dependent quantity $X(t)$. Higher-order derivatives are denoted using a number in brackets as superscripts, for example, $X^{(3)}(t)$ for a third derivative.

Column vectors are denoted using bold lowercase letters, for example, $\mathbf{a}$, whereas matrices use bold uppercase letters $\mathbf{A}$.
Points are represented as column vectors and denoted in bold, for example $\mathbf{p} = \begin{bmatrix} p_1 & p_2 \end{bmatrix}^\intercal$.
Tuples are reserved to reflect coordinate frames as discussed above.
Sets are generally denoted using a calligraphic font, for example, $\mathcal{S}$. Polygonal chains and polygons are also sets of points and use the same notation, for example:

$$\mathcal{P} = \{(\mathbf{p}_j)_{j=0}^{n_\mathcal{P}}\}, \tag{2.1.1}$$

with a total of $n_\mathcal{P}$ points $\mathbf{p}_j$. Trajectories in discrete time are also polygonal chains. Due to their importance in the thesis at hand, they are denoted using their own symbol $\tau$. Furthermore, curly brackets $\{\cdot\}$ are also used to reflect sets. The lowercase letter $n$ is used for the upper boundaries of counting variables and denote the number of objects or points with respective subscripts. Optimized variables are indicated using the asterisk symbol, for example, $\tau^*$. The mean of a quantity $\xi$ over a dataset consisting of $n_\mathcal{D}$ examples is denoted as $\overline{|\xi|}$. This notation is frequently used throughout the thesis at hand in experimental evaluations.
The cost functions used in the three different lane change planning approaches, refer to Figure 1.2, are distinguished using the superscript S (Sampling-based), O (Optimization-based) and G (Game Theoretic). For example, $J_L^\mathrm{G}$ denotes the longitudinal cost (subscript $L$) of the game theoretic planning approach.
Random variables are generally denoted using uppercase letters. Probabilities are denoted using the letter $P$. Events are denoted using a lowercase letter. The probability of event $v$ is hence $P(V = v)$. The probability density function of a random variable $V$ is denoted as $p(V)$ or also $p(v)$. Probability density functions and probability mass functions are not notationally distinguished within the thesis at hand. The cumulative distribution function of a random variable $V$ is denoted as $\Phi(V)$. Scalar latent random variables are denoted as $Z$ and multidimensional ones as $\mathbf{Z}$. The context always allows distinguishing multidimensional latent variables from matrices.

## 2.2. Curvilinear Coordinate Frame and Polygon Clipping

This Section introduces the curvilinear coordinate frame and a pseudo distance transformation that allows the description of a traffic scene in such frame without discontinuities. Furthermore, Polygon Clipping is briefly introduced due to its importance in this thesis's maneuver option identification.

### Curvilinear Coordinate Frame

The spatiotemporal maneuver identification described in Chapter 5 as well as the lane change planning approaches (Chapters 6 - 8) use formulations in a curvilinear coordinate frame. Its construction is introduced due to its importance for the remainder of the thesis at hand.

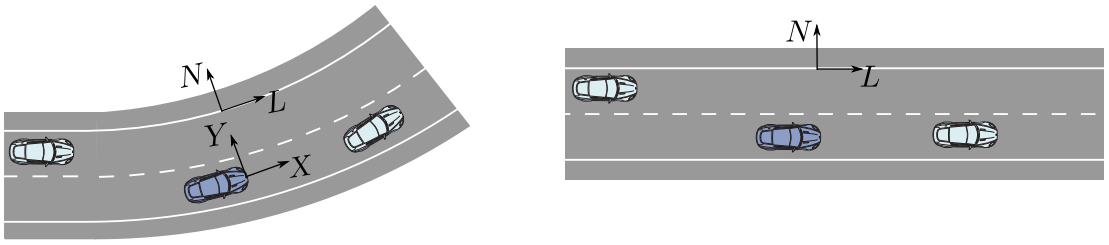



Figure 2.2.: Result of transforming a traffic situation into a curvilinear coordinate frame.

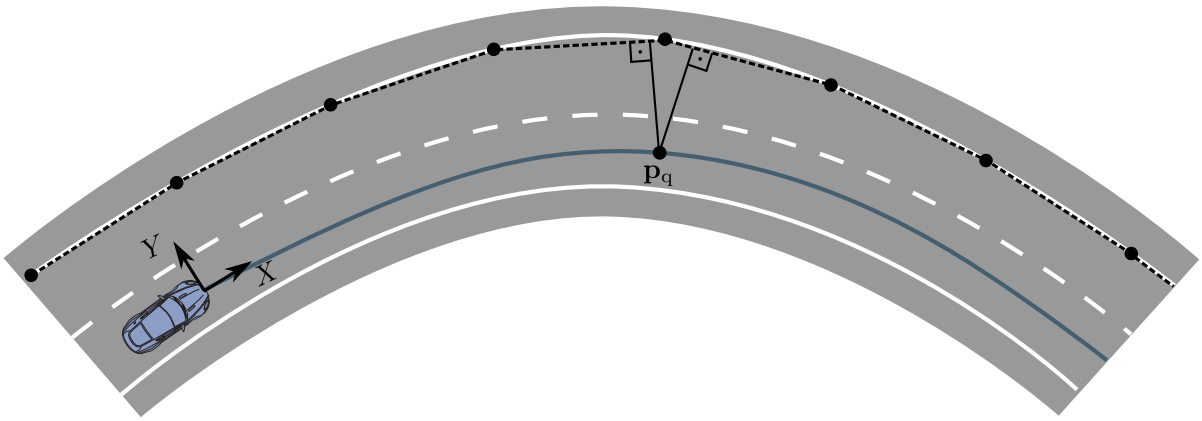



Figure 2.3.: A particular problem arises due to discontinous jumps in the lateral distance to a polygonal chain.

Generally, the static and dynamic environment of the ego-vehicle is described in an ego-vehicle coordinate frame $(X, Y)$. Its origin is located at the front bumper on the height of the road at that point.
In contrast, a curvilinear coordinate frame $(L, N)$ is attached to a choosen reference lane marking, which is represented as a polygonal chain in ego-vehicle coordinates:

$$\mathcal{P}_{\text{ref}} = \{(\mathbf{p}_{\text{ref},j})_{j=0}^{n_{\mathcal{P}_{\text{ref}}}}\} = \{\Big(\begin{bmatrix} X_{\text{ref},j} & Y_{\text{ref},j} \end{bmatrix}^\intercal\Big)_{j=0}^{n_{\mathcal{P}_{\text{ref}}}}\}, \tag{2.2.1}$$

with a total of $n_{\mathcal{P}_{\text{ref}}} + 1$ points. The choice of the coordinate origin is in general arbitary. All static and dynamic elements of the traffic situation need to be described in the curvilinear frame by calculation of the respective $L$ and $N$ coordinates using a distance transformation. After successfull transformation of all elements, even traffic situation with curvy roads appear straight as illustrated in Figure 2.2. This allows for easier maneuver options identification and subsequent lane change planning as will be described later in the thesis at hand.
The choice of a suitable distance transformation is crucial to avoid errors in the necessary transformation. Refer to Figure 2.3 for the illustration of discontinuous jumps that occur between two segments of a reference lane marking in the form of a polygonal chain $\mathcal{P}_{\text{ref}}$. The naive approach to calculate the orthogonal projection of a query point $\mathbf{p}_q$ belonging to a lane-keeping trajectory in ego-vehicle coordinates to the polygonal chain is not unambiguously defined in this particular case.
To circumvent this problem, [Ben+14] introduces a pseudo distance transformation. It

uses an idea from Phong-Shading [Pho75] and hence the field of Computer Graphics that results in a continuous distance metric. The approach is described using the illustration in Figure 2.4. Assume the polygonal chain consists of four points $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ and $\mathbf{p}_4$ initially referenced in an arbitary global coordinate system $(x, y)$. The goal is to calculate the distance between a query point $\mathbf{p}_q$ to the polygonal chain. First, a tangent vector $\mathbf{t}$ is defined at each point of the polygonal chain. For outer points, the definition for a polygonal chain consisting of $n_\mathcal{P}$ points is as follows:

$$\mathbf{t}_1 \coloneqq \mathbf{p}_2 - \mathbf{p}_1, \tag{2.2.2}$$

$$\mathbf{t}_{n_\mathcal{P}} \coloneqq \mathbf{p}_{n_\mathcal{P}} - \mathbf{p}_{n_\mathcal{P}-1}. \tag{2.2.3}$$

On constrast, the following definition holds for inner points:

$$\mathbf{t}_i \coloneqq \mathbf{p}_{i+1} - \mathbf{p}_{i-1}. \tag{2.2.4}$$

Assume for the moment, that the specific segment with a minimum distance to the query point is known. In the example this is the segment between points $\mathbf{p}_2$ and $\mathbf{p}_3$ with respective tangent vectors $\mathbf{t}_2$ and $\mathbf{t}_3$. The next step consists in a linear interpolation between the segment defining points and tangent vectors:

$$\mathbf{t}_\varrho = \varrho\mathbf{t}_3 + (1 - \varrho)\mathbf{t}_2, \tag{2.2.5}$$

$$\mathbf{p}_\varrho = \varrho\mathbf{p}_3 + (1 - \varrho)\mathbf{p}_2. \tag{2.2.6}$$

Once $\varrho$ is known, so are $\mathbf{p}_\varrho$ and $\mathbf{t}_\varrho$ and the pseudo distance is defined as:

$$\|\mathbf{n}_\varrho\|_2 = \|\mathbf{p}_\mathrm{q} - \mathbf{p}_\varrho\|_2, \tag{2.2.7}$$

with $\mathbf{n}_\varrho$ being orthogonal to $\mathbf{t}_\varrho$ in point $\mathbf{p}_\varrho$. Hence the constraint:

$$\mathbf{n}_\varrho^\intercal \mathbf{t}_\varrho = 0 \tag{2.2.8}$$

needs to hold. Refer again to Figure 2.4 for an illustration. At this point however, the value of $\varrho$ is yet unkown and needs to be derived. To this end, a segment local coordinate frame $(u, v)$ is introduced such that the origin coincides with $\mathbf{p}_2$ and the $u$ axis is aligned with the segment. In this coordinate system $\tilde{\mathbf{p}}_2 = \begin{bmatrix} 0 & 0 \end{bmatrix}^\intercal$ and $\tilde{\mathbf{p}}_3 = \begin{bmatrix} l & 0 \end{bmatrix}^\intercal$ hold. This choice allows the tangent vectors to be represented only by their slopes $\mathbf{t}_2 = \begin{bmatrix} 1 & m_2 \end{bmatrix}^\intercal$ and $\mathbf{t}_3 = \begin{bmatrix} 1 & m_3 \end{bmatrix}^\intercal$. Solving the orthogonality constraint Equation (2.2.8) for $\varrho$ leads to the solution:

$$\varrho = \frac{u_\mathrm{q} + v_\mathrm{q} m_2}{l - v_\mathrm{q}(m_3 - m_2)} \tag{2.2.9}$$

with $l$ the length of the segment and $\tilde{\mathbf{p}}_\mathrm{q} = \begin{bmatrix} u_\mathrm{q} & v_\mathrm{q} \end{bmatrix}^\intercal$ in the $(u, v)$ frame. Above mechanism allows the calculation of a pseudo distance to a given segment of a polygonal chain. To complete the algorithm, a search for the segment with minimum pseudo distance has to be conducted. This approach is used within the thesis at hand to transform traffic scenes into a curvilinear coordinate frame.

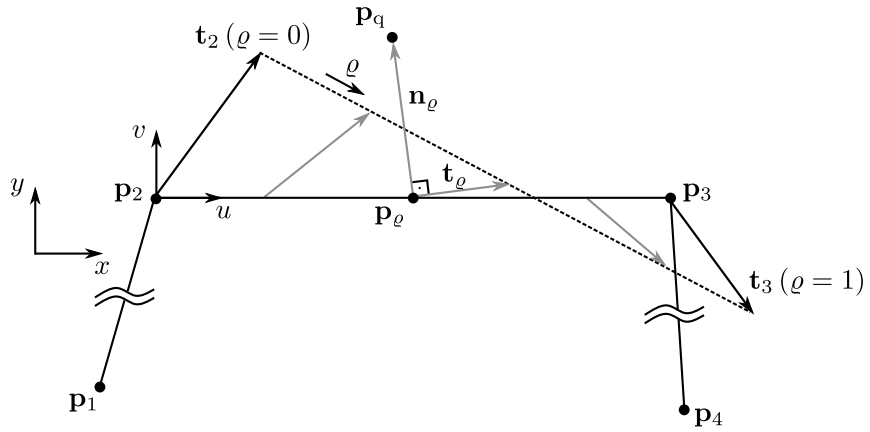



Figure 2.4.: Pseudo distance calculation of a query point $\mathbf{p}_\mathrm{q}$ to a segment between points $\mathbf{p}_2$ and $\mathbf{p}_3$ of a polygonal chain.

### Polygon Clipping

Polygon Clipping realizes certain geometric operations between a subject and clip polygon. For more information on specific algorithms, refer for example to [SH74], [Vat92], [GH98] and [Mar+09]. A polygon is defined as a geometric object consisting of a number of points called vertices and a number of line segments between them called edges. It can be compactly represented as a closed polygonal chain using set notation:

$$\mathcal{P}_i = \{(\mathbf{p}_j)_{j=0}^{n_{\mathcal{P}_i}}\} = \{\Big(\begin{bmatrix} L_j & t_j \end{bmatrix}^\mathsf{T}\Big)_{j=0}^{n_{\mathcal{P}_i}}\}, \tag{2.2.10}$$

in an arbitrary coordinate frame $(L, t)$ consisting of a total of $n_{\mathcal{P}_i} + 1$ points.
Assume that two polygons $\mathcal{P}_1$ and $\mathcal{P}_2$ are given. The most common operations are the following:

$$\begin{aligned} \text{Union} &: \mathcal{P}_1 \cup \mathcal{P}_2, \\ \text{InterSection} &: \mathcal{P}_1 \cap \mathcal{P}_2, \\ \text{Difference} &: \mathcal{P}_1 \setminus \mathcal{P}_2. \end{aligned}$$

Above operations are used extensively in Chapter 5 to derive the ego-vehicles maneuver options. Refer to Figure 2.5 for an illustration of them.

## 2.3. Simulation Environment and Development Tools

The thesis at hand is a result of an research cooperation project with the ZF Automotive GmbH. It aims to provide scientific and technical innovations in the field of lane change behavior planning. Due to the nature and goals of the project, the choice of development tools and simulation environments is limited.
One particular requirement is that the developed algorithms should be embeddable into the ZF Automotive GmbH software framework. Therefore, also the data interfaces need to match. Aforementioned requirement resulted in the development of DESIM (**D**riving

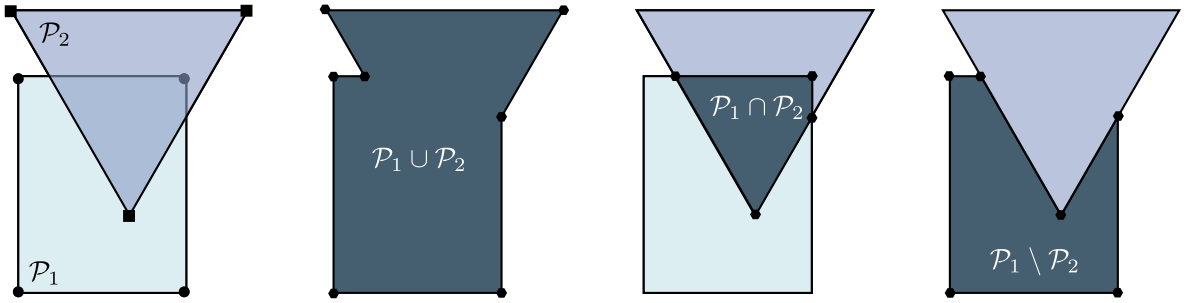
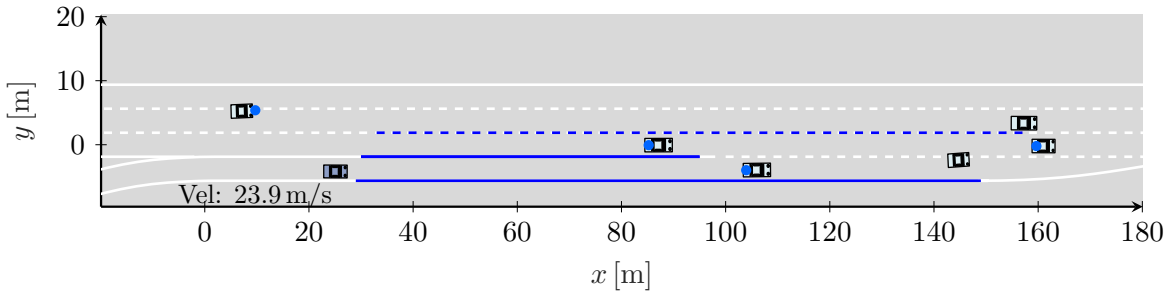
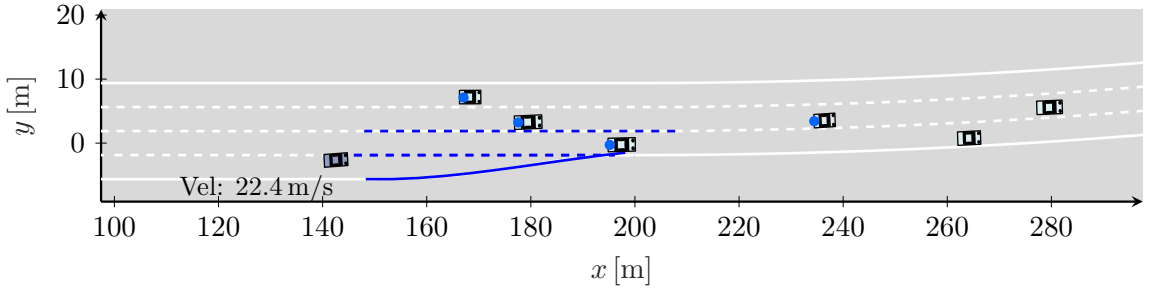



Figure 2.5.: Illustration of typical Polygon Clipping operations. $\mathcal{P}_1$ represents the subject and $\mathcal{P}_2$ the clip polygon. The dark-blue area is the outcome of the respective clipping operation.

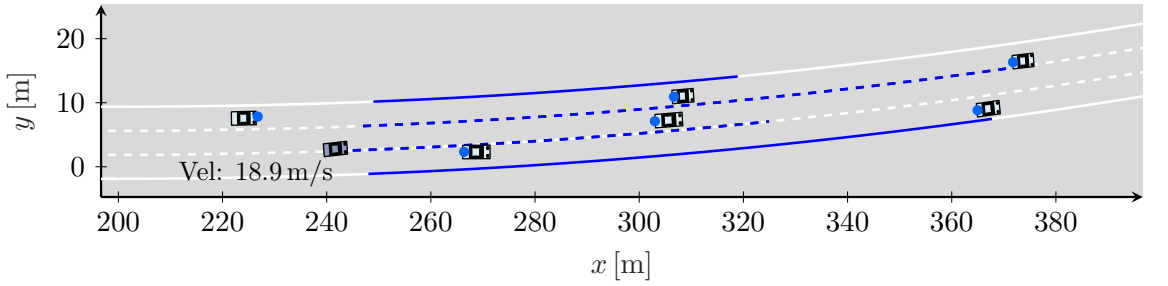
**E**nvironment **SIM**ulation) further described in [Wis+16]. It was developed using C++ and offers the possibility to simulate a wide range of highway traffic scenarios. The Intelligent Driver Model is used as the car-following model for all vehicles. A custom lane-changing model allows surrounding traffic participants to change lanes. Most simulation and driver model parameters can be varied probabilistically to allow numerous evaluations. A sensor system can be specified and is used for an object-level sensor system simulation, yielding measurement uncertainties similar to the data obtained by the real vehicle sensors. More information can be found in [Wis+16].
Figure 2.6 shows a highway entrance scenario. The ego-vehicle is shown in dark-blue with its current velocity. The surrounding vehicles are represented in light-blue. A dot represents simulated sensor detection in the form of radar reflection points. The white lines represent the ground-truth lane markings, whereas detected ones are shown in blue. As for development tools, Python and the scikit-learn library [Ped+11] are used for lane change predictions and the desired velocity regression task in Chapter 4. Apart from that, MATLAB is used for prototyping and C for deployment. There is an interface between the DESIM simulation environment and MATLAB. This combination allows for efficient prototyping.
The above choice is not the only possible choice, but the most comfortable in the described circumstances. However, a critical reflection as of the time of writing the thesis at hand clearly shows a rise in the use of the CARLA (**CAR L**earning to **A**ct) simulation environment, refer to [Dos+17] and the CommonRoad benchmark environment [Alt+17]. These are more standardized choices that allow scientific comparisons. It is noted that those tools should be the choice in the future. Time should be invested to adapt the data interfaces so that prototyped code could be easily deployed into the project partner's software system.



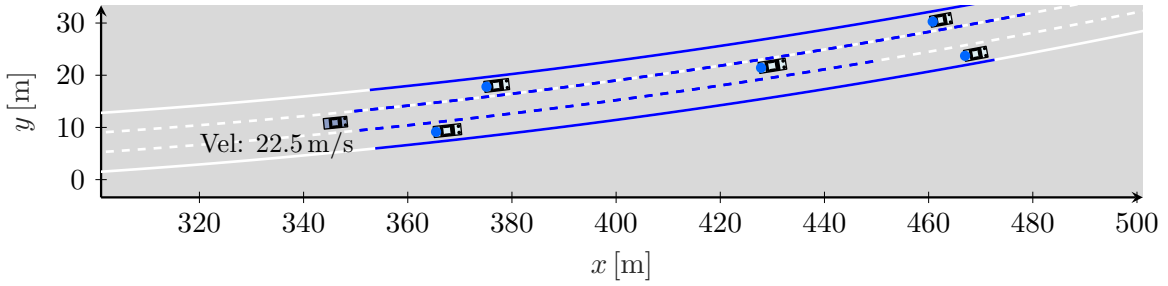

(a) Traffic scene at $t = 8\,\mathrm{s}$.




(b) Traffic scene at $t = 13\,\mathrm{s}$.




(c) Traffic scene at $t = 18\,\mathrm{s}$.

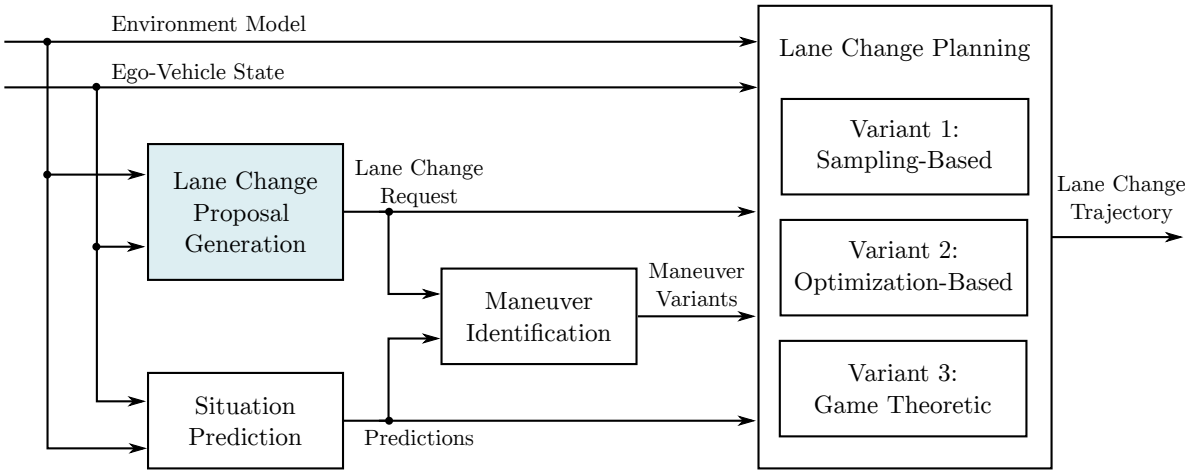



(d) Traffic scene at $t = 23\,\mathrm{s}$.

Figure 2.6.: Exemplary traffic scenario realized using the DESIM simulation environment. The ego-vehicle is represented in dark-blue whereas the surrounding traffic participents are shown in light-blue. The blue dots represent the simulated sensor system detections. Detected lanes are also depicted in blue color. The current ego-vehicle velocity is furthermore shown.

# 3

# Uncertainty-Aware Discretionary Lane Change Proposal Generation




This Chapter focuses on the discretionary lane change proposal generation. It is largely based on the publication [Sch+21d]. The above figure gives an overview of its location among all developed modules within the thesis at hand.
Discretionary lane changes aim to increase passenger comfort and reduce travel time. The questions that the developed module answers are the following. How much velocity deviation is accepted by the driver of the automated vehicle? When will the driver feel the need to prepare for a lane change? The discretionary lane change proposal module analyses the traffic scene, answers both questions, and might recommend a lane change to the driver of the automated vehicle or directly issues the request to further downstream modules. If the proposal is accepted, the lane change behavior planning finally tries to provide a safe and comfortable lane change trajectory.
The subsequent Sections place a particular focus on modeling the utility of driving lanes using a probabilistic framework. This way, noise in the perception system is naturally accounted for. Furthermore, the driver will often accept low deviations of the desired velocity for a certain period of time. This factor is modeled using a probability distribution around the desired velocity. The Chapter is structured as follows. Section 3.1 presents the

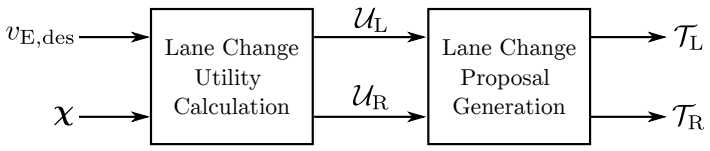



Figure 3.1.: High-level overview of the proposed module. First, the lane change utility to the adjacent lanes are calculated and finally, in case that certain conditions are fulfilled, a lane change is proposed to the driver of the automated vehicle.

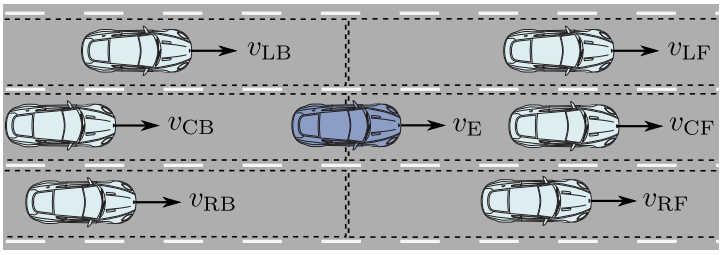



Figure 3.2.: Nomenclature for the traffic scene features $\boldsymbol{\chi}$. Both adjacent lanes are assessed for the calulcation of the corresponding lane change utilities. The traffic participant with velocity $v_{\mathrm{RB}}$ is introduced for completeness sake but not used in the calculations.

architecture of the module and all relevant parameters. These parameters are optimized and the results are discussed in Section 3.2. After that, the optimized module is evaluated on data from a ZF Group test vehicle in Section 3.3.

## 3.1. Architecture of the Discretionary Lane Change Proposal Module

A high-level overview of the proposed module is given in Figure 3.1. It consists of two stages. First the lane change utilities ($\mathcal{U}_{\mathrm{L}}$ and $\mathcal{U}_{\mathrm{R}}$) are caculated and after that proposals ($\mathcal{T}_{\mathrm{L}}$ and $\mathcal{T}_{\mathrm{R}}$) are generated if certain conditions are fulfilled. In the following discussion, all velocities are measured in the stationary ego-vehicle coordinate frame as $v = \sqrt{\dot{X}^2 + \dot{Y}^2}$. The desired velocity $v_{\mathrm{E,des}}$ and the traffic situations features:

$$\boldsymbol{\chi} = \begin{bmatrix} v_{\mathrm{E}} & v_{\mathrm{LF}} & v_{\mathrm{CF}} & v_{\mathrm{RF}} & v_{\mathrm{RB}} & v_{\mathrm{CB}} & v_{\mathrm{LB}} \end{bmatrix}^{\intercal} \in \mathbb{R}_{+}^{7}, \tag{3.1.1}$$

refer also to Figure 3.2, are the two inputs of the module. The left and right lane change utilities $\mathcal{U}_{\mathrm{L}} \in [0, 1]$ and $\mathcal{U}_{\mathrm{R}} \in [0, 1 + \gamma_2 + \gamma_3]$, with $\gamma_2$ and $\gamma_3$ being model parameters, are further analyzed to finally derive binary proposals $\mathcal{T}_{\mathrm{L}} \in \{0, 1\}$ and $\mathcal{T}_{\mathrm{R}} \in \{0, 1\}$.
The nomenclature for the traffic scene features $\boldsymbol{\chi}$ is shown in Figure 3.2. The dark-blue vehicle is the ego-vehicle, and light-blue ones represent surrounding traffic participants. Two subscripts are used for these vehicles, the first denoting the lane to distinguish **C**urrent ego-vehicle lane (C), **L**eft lane (L) and **R**ight lane (R). The second subscript distinguishes **F**ront (F) and **B**ack (B) with respect to the ego-vehicle.

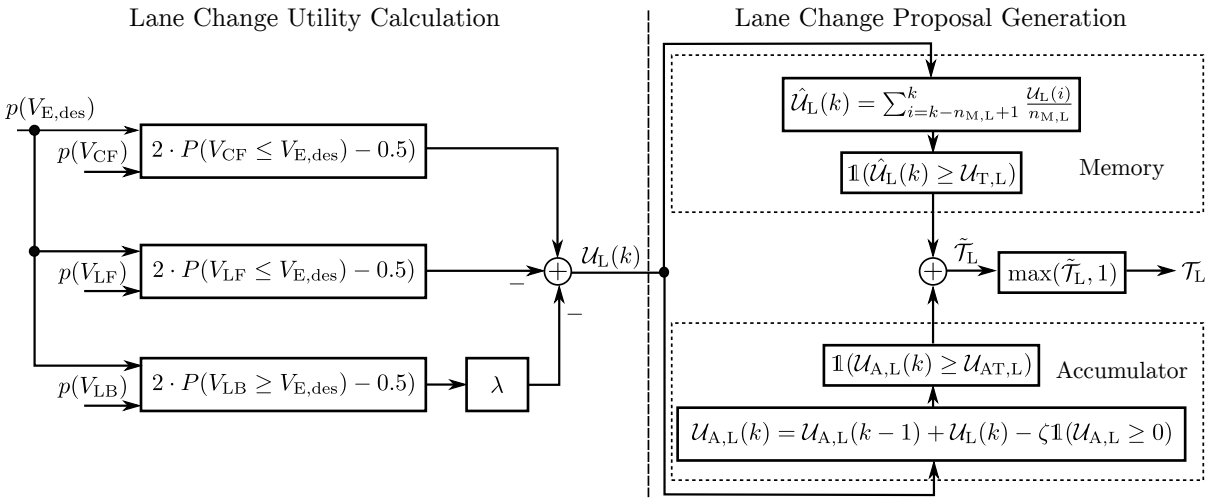



Figure 3.3.: Detailed overview of the module components for left lane changes. The left part of the Figure corresponds to the lane change utility calculation whereas the right part depicts the memory and accumulator trigger modules for the proposal generation. Herein, $\mathbb{1}(\cdot)$ represents the indicator function, refer to Equation (3.1.16) and (3.1.18).

### Module Components

Figure 3.3 gives a detailed overview of all components of the left lane change module. The left part of the Figure represents the calculation of the resulting utility $\mathcal{U}_{\mathrm{L}}(k)$ per discrete algorithm timestep $k$. The utility is calculated based on probabilities of velocity comparisons. A politeness factor $\lambda$ is integrated to include courtesy in the proposal generation process. There are certainly other important features for the lane change utility calculation. However, velocities play a dominant role. An overtaking maneuver is usually triggered to avoid deceleration or reach one's desired velocity. Also note, that the proposed module has a different aim compared to lane change prediction modules. Latter, try to predict lane changes of surrounding vehicles. Such approaches have to consider more features such as distances between vehicles and lateral velocities, refer for example to [Wis+17], [Krü+19b] and Chapter 4 of the thesis at hand.

The right part of Figure 3.3 refers to the functionality of the module that generates the binary trigger signal $\mathcal{T}_{\mathrm{L}}$. A similar Figure can be drawn for the module that generates discretionary right lane change proposals. Again, there are slight adaptions since more parameters are used, and a bias to the right lane is included in that case.

### Calculation of Lane Change Utilities

The calculation of lane change utilities is based on probability distributions of certain velocities. All velocities in Figure 3.2 are assumed to follow Gaussian probability distributions, see Equation (3.1.5) for an example. Therein $\mu$ refers to the mean value, and $\sigma^2$ represents the variance. Measuring these velocities using the vehicle's sensor system and employing a tracking algorithm introduces uncertainties. In the remainder of this Chapter, lowercase letters for velocities always correspond to deterministic quantities, whereas uppercase letters are used for random variables.

The utility for a left lane change is calculated as follows:

$$\begin{aligned}\mathcal{U}_{\mathrm{L}} = \max(2\left(P\left(V_{\mathrm{CF}} \leq V_{\mathrm{E,des}}\right) - 0.5\right) + \\ -2\left(P\left(V_{\mathrm{LF}} \leq V_{\mathrm{E,des}}\right) - 0.5\right) + \\ -2\lambda\left(P\left(V_{\mathrm{LB}} \geq V_{\mathrm{E,des}}\right) - 0.5\right), 0).\end{aligned} \tag{3.1.2}$$

Herein, the first term $P\left(V_{\mathrm{CF}} \leq V_{\mathrm{E,des}}\right)$ gives the probability of the current driving lane being slower than the desired velocity and hence high values favor a lane change. The second term $P\left(V_{\mathrm{LF}} \leq V_{\mathrm{E,des}}\right)$ balances the first term and gives the probability of the left lane being slower than the desired velocity. From the ego-vehicles perspective, it only makes sense to change lanes to the left if speed can be gained through the change. Finally, the last term $P\left(V_{\mathrm{LB}} \geq V_{\mathrm{E,des}}\right)$ considers potentially faster vehicles from the back that could influence the lane change intention. The offset $-0.5$ and scaling factor 2 ensure that the inidivual utilities are always in the range $[0, 1]$. Furthermore, a distance-dependent lower boundary is imposed on the surrounding vehicle's velocity variances, refer to [Sch+21d] for more details. The utility for a right lane change is calculated differently to account for european passing rules and faster vehicles behind the ego-vehicle:

$$\begin{aligned}\mathcal{U}_{\mathrm{R}} = \max(1 - 2\gamma_1\left(P\left(V_{\mathrm{RF}} \leq V_{\mathrm{E,des}}\right) - 0.5\right) + \\ +2\gamma_2\left(P\left(V_{\mathrm{CF}} \leq V_{\mathrm{E,des}}\right) - 0.5\right) + \\ +2\gamma_3\left(P\left(V_{\mathrm{CB,max}} \geq V_{\mathrm{E}}\right) - 0.5\right), 0).\end{aligned} \tag{3.1.3}$$

The constant 1 introduces the right lane bias. Hence, if no vehicles are around the ego-vehicle, the utility is $\mathcal{U}_{\mathrm{R}} = 1$. The second term $P\left(V_{\mathrm{RF}} \leq V_{\mathrm{E,des}}\right)$ represents the probability that the vehicle in front on the right lane is slower than the desired velocity. Such a situation decreases the lane change's utility to the right since it is often better to pass the slower vehicle first. In contrast, the probability $P\left(V_{\mathrm{CF}} \leq V_{\mathrm{E,des}}\right)$ favors a lane change, since a high value indicates that the current driving lane is unsatisfactory. However, special care is taken to ensure the traffic rule that vehicles should not be overtaken on the right. An additional constraint therefore ensures that $\mu_{V_{\mathrm{RF}}} \leq \mu_{V_{\mathrm{CF}}}$. Finally, the last term $P\left(V_{\mathrm{CB,max}} \geq V_{\mathrm{E}}\right)$ represents the probability that a vehicle with higher velocity than the current ego-vehicle velocity drives behind it. Note that this one uses the current velocity and not the desired velocity in contrast to the remaining terms. The ego-vehicle should show courtesy after overtaking a slower vehicle and quickly give way if there is a faster vehicle behind it. The subscript max reflects the fact that the maximum observed velocity of the trailing vehicle is used since it potentially best describes its intended velocity.
Other factors could be included into the utility functions. Assume that there are $n_{\mathcal{U}}$ distinct utilities. One could be the above stated utility based on probabilities of velocity comparisons. Others could be based on instantaneous or predicted accelerations and the traffic density on the respective driving lanes. The resulting utility can be calculated using a convex combination in form of a sum of all individual utilities:

$$\mathcal{U} = \sum_{i=1}^{n_{\mathcal{U}}} w_i \mathcal{U}_i \tag{3.1.4}$$

with $w_i \in \mathbb{R}_+$ and $\sum_{i=1}^{n_{\mathcal{U}}} w_i = 1$.

All uppercase velocity variables are modelled using Gaussian probability distributions, for example:

$$p(V_{\mathrm{E,des}}) = \frac{1}{\sqrt{2\pi\sigma^2_{V_{\mathrm{E,des}}}}} \exp\left(-\frac{(V_{\mathrm{E,des}} - \mu_{V_{\mathrm{E,des}}})^2}{2\sigma^2_{V_{\mathrm{E,des}}}}\right). \tag{3.1.5}$$

In order to calculate the probabilites in both utility functions Equation (3.1.2) and (3.1.3), differences of random variables need to be formed. For example:

$$P\left(V_{\mathrm{CF}} \leq V_{\mathrm{E,des}}\right) = P\left(V_{\mathrm{CF}} - V_{\mathrm{E,des}} \leq 0\right). \tag{3.1.6}$$

The probability distribution $p(\tilde{V}) = p(V_{\mathrm{CF}} - V_{\mathrm{E,des}})$ is found by convolution:

$$p(\tilde{V}) = \frac{1}{\sqrt{2\pi\sigma^2_{\tilde{V}}}} \exp\left(-\frac{(\tilde{V} - \mu_{\tilde{V}})^2}{2\sigma^2_{\tilde{V}}}\right), \tag{3.1.7}$$

with the mean and variance:

$$\mu_{\tilde{V}} = \mu_{V_{\mathrm{CF}}} - \mu_{V_{\mathrm{E,des}}}, \tag{3.1.8}$$

$$\sigma^2_{\tilde{V}} = \sigma^2_{V_{\mathrm{CF}}} + \sigma^2_{V_{\mathrm{E,des}}}. \tag{3.1.9}$$

Prior to the calculation of the probabilities, several mathematical expressions are introduced. The integral of a Gaussian distribution from minus infinity to a certain value $x$ is called cumulative distribution function and denoted as:

$$\Phi_{\mu,\sigma}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{x} \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right) \mathrm{d}t. \tag{3.1.10}$$

The error function $\mathrm{erf}(x)$ and complementary error function $\mathrm{erfc}(x)$ are defined as:

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp\left(-t^2\right) \mathrm{d}t = 1 - \mathrm{erfc}(x). \tag{3.1.11}$$

Equation (3.1.10) can be expressed using the error-function or complementary error function as follows:

$$\begin{aligned} \Phi_{\mu,\sigma}(x) &= \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{x-\mu}{\sqrt{2\sigma^2}}\right)\right) \\ &= 1 - \frac{1}{2}\mathrm{erfc}\left(\frac{x-\mu}{\sqrt{2\sigma^2}}\right). \end{aligned} \tag{3.1.12}$$

The occuring probabilities in the utility functions Equation (3.1.2) and (3.1.3) can therefore be expressed as follows:

$$\begin{aligned} P(X \geq a) =& \int_a^\infty p(x)\mathrm{d}x \\ =& \int_{-\infty}^a p(x)\mathrm{d}x - \int_{-\infty}^a p(x)\mathrm{d}x + \int_a^\infty p(x)\mathrm{d}x \\ =& 1 - \int_{-\infty}^a p(x)\mathrm{d}x \\ =& 1 - \Phi_{\mu,\sigma}(a) \\ =& \frac{1}{2}\mathrm{erfc}\left(\frac{a-\mu}{\sqrt{2\sigma^2}}\right) \end{aligned} \tag{3.1.13}$$

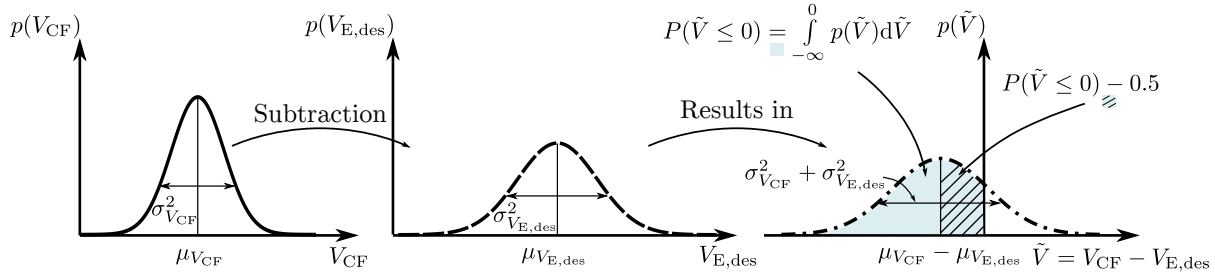



Figure 3.4.: Utility calulcation based on Gaussian probability distributions. Generally, two random variables are subtracted by convolving their corresponding probability distributions. The integral (light-blue area in the figure) is analytically intractable and hence an approximation of the $\mathrm{erf}(x)$ is used to evaluate it efficiently.

and

$$P(X \leq a) = \Phi_{\mu,\sigma}(a) = 1 - \frac{1}{2}\mathrm{erfc}\left(\frac{a-\mu}{\sqrt{2\sigma^2}}\right). \tag{3.1.14}$$

The calculation of all relevant probabilities is illustrated in Figure 3.4. It also illustrates the rational behind the factors 2 and $-0.5$ in both utility functions Equation (3.1.2) and Equation (3.1.3). Take for example $P\left(V_{\mathrm{CF}} \leq V_{\mathrm{E,des}}\right)$ as shown in Figure 3.4. The quantity $P\left(V_{\mathrm{CF}} \leq V_{\mathrm{E,des}}\right) - 0.5$ corresponds to the hatched area under the gaussian density function. In the case that $\mu_{V_{\mathrm{CF}}} = \mu_{V_{\mathrm{E,des}}}$, this area is zero and hence $P\left(V_{\mathrm{CF}} \leq V_{\mathrm{E,des}}\right) - 0.5 = 0$. This realizes the desired behavior of the proposed module that this part of the utility function Equation (3.1.2) is zero in that case. A lane change is of no utility if the vehicle in front of the ego-vehicle is faster than its desired velocity ($\mu_{V_{\mathrm{CF}}} \geq \mu_{V_{\mathrm{E,des}}}$). The hatched area has a maximum value of 0.5 so that the factor 2 in Equation (3.1.2) and Equation (3.1.3) ensure that the respective utility is within the range of $[0, 1]$. Note that $\mu_{V_{\mathrm{CF}}}$ is bounded by $\mu_{V_{\mathrm{E,des}}}$ to ensure that $P\left(V_{\mathrm{CF}} \leq V_{\mathrm{E,des}}\right) - 0.5$ is always a positive quantity. All other velocities are handled similarly.
Unfortunately, the calculation of probabilities is analytically intractable. Instead, a rational approximation of the error function $\mathrm{erf}(x)$ is used. Refer to the Section A.2 of the appendix for further details.

### Generation of Lane Change Proposals

The utilities for a left and right lane change are calculated using Equation (3.1.2) and (3.1.3) respectively. Next, triggering criteria are defined using these quantities on the example of the left lane change proposal module. Two different mechanisms are discussed subsequently. As shown in Figure 3.3, these are a memory and an accumulator mechanism. The rationale behind this choice is as follows. Typically, lane change decisions occur very fast if the driver spots a slow truck on his current driving lane. In such situations, the utility for a lane change is rather high, and the memory mechanism triggers the lane change quickly after the slow vehicle or truck is detected. On the other hand, drivers tend to accept slight deviations from their desired velocity for longer times but eventually change lanes to minimize travel time. The proposed accumulator mechanism is introduced for this purpose, and its parameters are optimized such that it generally triggers after the

memory. All aspects of the optimization are described in the next Section 3.2.
Every discrete algorithm timestep $k$, the corresponding utilities over the last $n_{\mathrm{M,L}}$ steps are averaged and result in $\hat{\mathcal{U}}_{\mathrm{L}}(k) \in [0,1]$:

$$\hat{\mathcal{U}}_{\mathrm{L}}(k) = \sum_{i=k-n_{\mathrm{M,L}}+1}^{k} \frac{\mathcal{U}_{\mathrm{L}}(i)}{n_{\mathrm{M,L}}}. \tag{3.1.15}$$

This way, noise in the detections and hence fluctuations in the utility have less strong implications and still lead eventually to a lane change proposal. The trigger itself is defined using the indicator function:

$$\mathcal{T}_{\mathrm{M,L}} = \mathbb{1}(\hat{\mathcal{U}}_{\mathrm{L}}(k) \geq \mathcal{U}_{\mathrm{T,L}}) \coloneqq \begin{cases} 0, & \hat{\mathcal{U}}_{\mathrm{L}}(k) < \mathcal{U}_{\mathrm{T,L}} \\ 1, & \hat{\mathcal{U}}_{\mathrm{L}}(k) \geq \mathcal{U}_{\mathrm{T,L}} \end{cases}, \tag{3.1.16}$$

which means that the averaged utility $\hat{\mathcal{U}}_{\mathrm{L}}(k)$ needs to be over a defined treshold $\mathcal{U}_{\mathrm{T,L}} \in [0,1]$.
The discrete accumulator is governed by the following difference Equation:

$$\mathcal{U}_{\mathrm{A,L}}(k) = \mathcal{U}_{\mathrm{A,L}}(k-1) + \mathcal{U}_{\mathrm{L}}(k) - \zeta_{\mathrm{L}} \mathbb{1}(\mathcal{U}_{\mathrm{A,L}}(k) \geq 0). \tag{3.1.17}$$

Herein, $\mathcal{U}_{\mathrm{L}}(k)$ represents the utility for the current discrete algorithm timestep $k$ and $\mathcal{U}_{\mathrm{A,L}}(k) \in \mathbb{R}_{+}$ is the accumulated utility. The rightmost term $\zeta_{\mathrm{L}} \mathbb{1}(\mathcal{U}_{\mathrm{A,L}}(k) \geq 0)$ models a leakage with factor $\zeta_{\mathrm{L}} \in [0,1]$ that is active as long as the accumulator state is not empty. The rationale behind this modeling choice is the following. There are situations on highways in which the ego-vehicle has to deviate from its desired velocity temporarily. A typical situation is a cut-in maneuver of a surrounding traffic participant in front of the ego-vehicle. In this case, slowing down is crucial to maintain a safe distance to the new leader vehicle. The utility for a lane change rises in that case; however, if the deviation is only temporary, the leakage factor correctly models "forgetting" this event after a certain time. It furthermore introduces another degree of freedom that is exploited in the parameter optimization.
The trigger is again defined using the indicator function:

$$\mathcal{T}_{\mathrm{A,L}} = \mathbb{1}(\mathcal{U}_{\mathrm{A}}(k) \geq \mathcal{U}_{\mathrm{AT,L}}) \coloneqq \begin{cases} 0, & \mathcal{U}_{\mathrm{A}}(k) < \mathcal{U}_{\mathrm{AT,L}} \\ 1, & \mathcal{U}_{\mathrm{A}}(k) \geq \mathcal{U}_{\mathrm{AT,L}} \end{cases}, \tag{3.1.18}$$

and hence it is checked that the accumulator state $\mathcal{U}_{\mathrm{A}}(k)$ is above a defined and optimized threshold $\mathcal{U}_{\mathrm{AT,L}}$.
Finally, the resulting trigger for proposal generation is realized using a logic OR operation. In case of binary variables, this is simply the addition of both distinct trigger signals:

$$\mathcal{T}_{\mathrm{L}} = \max\left(\mathcal{T}_{\mathrm{M,L}} + \mathcal{T}_{\mathrm{A,L}}, 1\right) = \max(\tilde{\mathcal{T}}_{\mathrm{L}}, 1) \in \{0,1\}. \tag{3.1.19}$$

The above exposition also applies analogously for right lane changes. Both the left and right lane change proposals have equal priority and could be proposed simultaneously to the automated vehicle driver.

## 3.2. Optimization of the Module's Parameters

This Section introduces the optimization of all relevant parameters for the left and right lane change proposal module described in the contribution at hand. The goal of the developed module is to propose the driver of an automated vehicle a lane change when the traffic situation suggests that an advantage results from it. Hence, to optimize its parameters, a dataset is needed. The dataset should include all sensor measurements and the trigger signal. It was collected using a driving simulator at the Institute of Control Theory and Systems Engineering at TU Dortmund University. A total of 11 scenarios were chosen. The study subjects are placed in varying traffic situations and should indicate their lane change intention. All scenarios are designed to ensure that a lane change due to dissatisfaction with the current driving lane will certainly be issued at some point. Refer to Section A.3 of the appendix for further information about the study and scenarios.
Figure 3.5, Figure 3.6 and Figure 3.7 show the driving simulator study participant's timing distributions represented as histograms. Additionally the figures also show probability distributions that are fitted to the data using **K**ernel **Dens**ity (KDens) estimation (see [Par62] and [Hol+15] for a more comprehensive treatment). The histograms suggest rather large variances, depicting that the driving styles of the participants vary to some degree. Moreover, the figures also include the optimized proposals of the developed module. The results are further discussed subsequently.

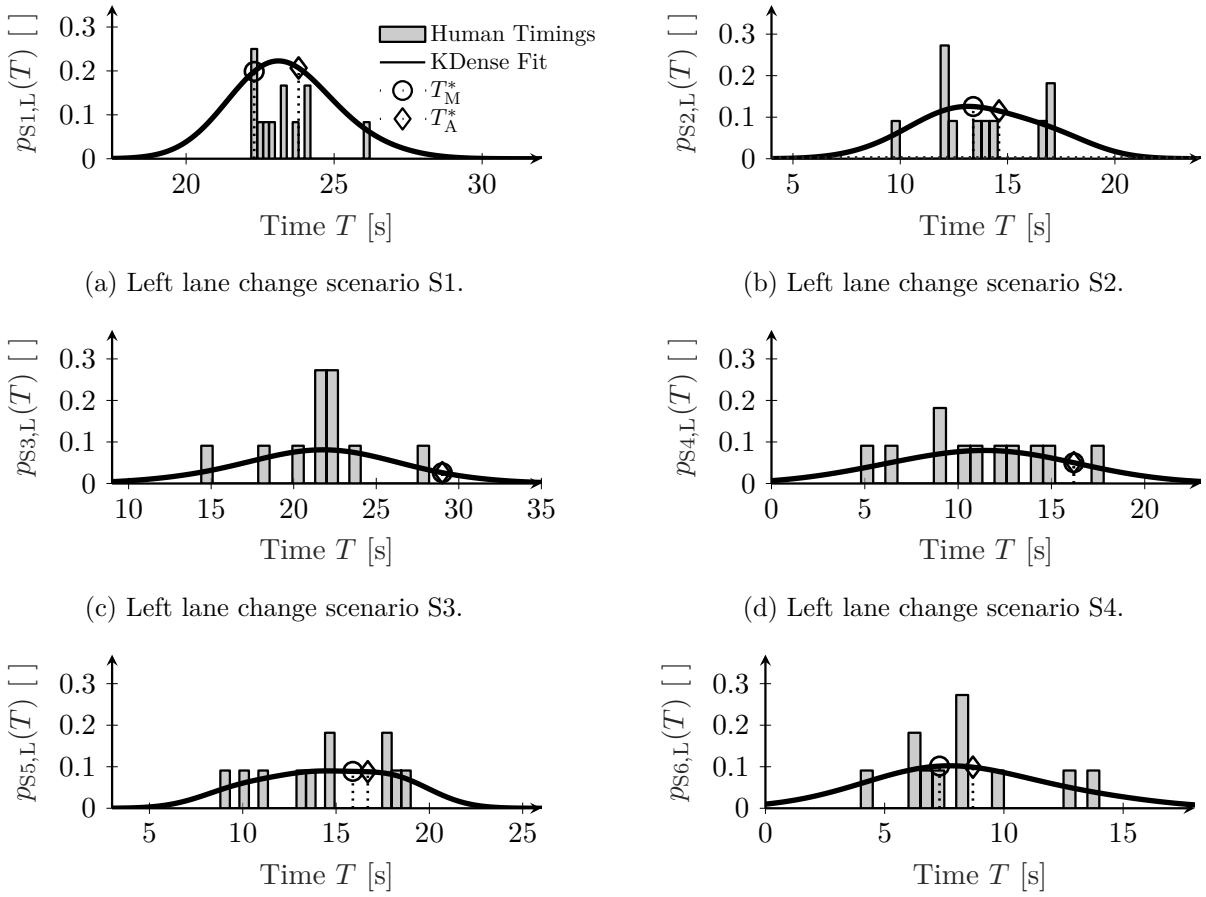
### Optimization Procedure and Results

The proposed module has several design parameters. Using the scenarios that are described in Section A.3 of the appendix and the corresponding data of the driving-simulator study, the free parameters are optimized in a brute-force fashion.
Note that the driving simulator study is limited in both the number of participants and scenarios. Hence, it cannot be claimed that the optimized parameters work well for other drivers that potentially have other driving styles. A larger dataset is needed to obtain results that generalize well.
In the first step, a coarse search grid is used. Afterward, a finer grid is used around the optimal parameter values that the first coarse optimization yielded. The histograms of the lane change intention timing distributions and corresponding fitted probability distributions are shown in Figure 3.5, Figure 3.6 and Figure 3.7 for the left and right lane change scenarios respectively.
Next, the parameter vectors to be optimized are introduced. The first step of the optimization aims at the following parameter vector for the left lane change module:

$$\boldsymbol{\xi}_{\mathrm{M,L}} = \begin{bmatrix} \sigma_{v_{\mathrm{E,des,L}}} & n_{\mathrm{M,L}} & \mathcal{U}_{\mathrm{T,L}} & \lambda \end{bmatrix}^{\mathsf{T}}. \tag{3.2.1}$$

Therein $\sigma_{v_{\mathrm{E,des,L}}}$ represents the desired velocity standard deviation of the ego-vehicle, $n_{\mathrm{M,L}}$ the discrete timesteps $k$ for averaging according to Equation (3.1.15), $\mathcal{U}_{\mathrm{T,L}}$ the treshold of the memory trigger mechanism and finally the politeness factor $\lambda \in [0, 1]$ in Equation (3.1.2). The second optimization aims only at the accumulator trigger module parts:

$$\boldsymbol{\xi}_{\mathrm{T,L}} = \begin{bmatrix} \zeta_{\mathrm{L}} & \mathcal{U}_{\mathrm{AT,L}} \end{bmatrix}^{\mathsf{T}}, \tag{3.2.2}$$




(a) Left lane change scenario S1.

(b) Left lane change scenario S2.

(c) Left lane change scenario S3.

(d) Left lane change scenario S4.

(e) Left lane change scenario S5.

(f) Left lane change scenario S6.

Figure 3.5.: Timing distribution plots with corresponding fitted probability distributions and proposal timings of the optimized module for left lane change scenarios S1-S6, refer to Table A.1 in the appendix. Herein, $T^*_M$ represents the proposal timing of the memory trigger and $T^*_A$ of the accumulator trigger module.

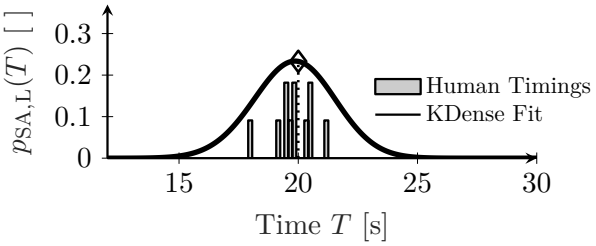
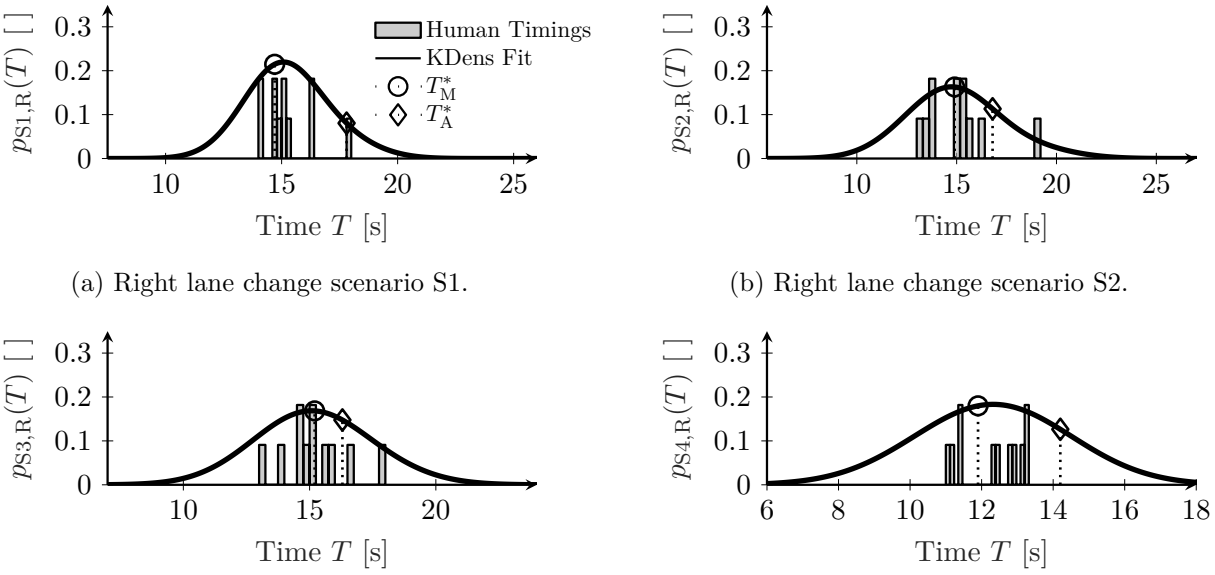



Figure 3.6.: Timing distribution plots with corresponding fitted probability distributions and proposal timings of the optimized module for the left lane change accumulator scenario SA, refer to Table A.1 in the appendix. Herein, $T^*_A$ represents the proposal timing of the accumulator trigger module.




(a) Right lane change scenario S1.

(b) Right lane change scenario S2.

(c) Right lane change scenario S3.

(d) Right lane change scenario S4.

Figure 3.7.: Timing distribution plots with corresponding fitted probability distributions and proposal timings of the optimized module for right lane change scenario S1-S4, refer to Table A.2 in the appendix. Herein, $T_{\mathrm{M}}^*$ represents the proposal timing of the memory trigger and $T_{\mathrm{A}}^*$ of the accumulator trigger module.

with the leakage factor $\zeta_{\mathrm{L}} \in \mathbb{R}_+$ and the trigger treshold $\mathcal{U}_{\mathrm{AT,L}}$, refer also to Figure 3.3. The complete parameter vector is hence:

$$\boldsymbol{\xi}_{\mathrm{L}} = \begin{bmatrix} \boldsymbol{\xi}_{\mathrm{M,L}}^{\intercal} & \boldsymbol{\xi}_{\mathrm{T,L}}^{\intercal} \end{bmatrix}^{\intercal}. \tag{3.2.3}$$

Similar definitions for the right lane change model hold:

$$\boldsymbol{\xi}_{\mathrm{M,R}} = \begin{bmatrix} \sigma_{v_{\mathrm{E,des,R}}}^2 & n_{\mathrm{M,R}} & \mathcal{U}_{\mathrm{T,R}} & \gamma_1 & \gamma_2 & \gamma_3 \end{bmatrix}^{\intercal}, \tag{3.2.4}$$

$$\boldsymbol{\xi}_{\mathrm{T,R}} = \begin{bmatrix} \zeta_{\mathrm{R}} & \mathcal{U}_{\mathrm{AT,R}} \end{bmatrix}^{\intercal}, \tag{3.2.5}$$

$$\boldsymbol{\xi}_{\mathrm{R}} = \begin{bmatrix} \boldsymbol{\xi}_{\mathrm{M,R}}^{\intercal} & \boldsymbol{\xi}_{\mathrm{T,R}}^{\intercal} \end{bmatrix}^{\intercal}. \tag{3.2.6}$$

The optimization for the left and right lane change model is mathematically stated as maximizing the log-Likelihood ($J_{\mathrm{TM,L}}$ and $J_{\mathrm{TM,R}}$) of all respective scenarios:

$$J_{\mathrm{TM,L}}(\boldsymbol{\xi}_{\mathrm{M,L}}) = \sum_{\mathrm{i}=1}^{6} \ln(p_{\mathrm{Si,L}}(T_{\mathrm{Si,L}}(\boldsymbol{\xi}_{\mathrm{M,L}}))), \tag{3.2.7}$$

$$\boldsymbol{\xi}_{\mathrm{M,L}}^* = \underset{\boldsymbol{\xi}_{\mathrm{M,L}}}{\operatorname{argmax}}\, J_{\mathrm{TM,L}}(\boldsymbol{\xi}_{\mathrm{M,L}}), \tag{3.2.8}$$

$$J_{\mathrm{TM,R}}(\boldsymbol{\xi}_{\mathrm{M,R}}) = \sum_{\mathrm{i}=1}^{4} \ln(p_{\mathrm{Si,R}}(T_{\mathrm{Si,R}}(\boldsymbol{\xi}_{\mathrm{M,R}}))), \tag{3.2.9}$$

$$\boldsymbol{\xi}_{\mathrm{M,R}}^* = \underset{\boldsymbol{\xi}_{\mathrm{M,R}}}{\operatorname{argmax}}\, J_{\mathrm{TM,R}}(\boldsymbol{\xi}_{\mathrm{M,R}}). \tag{3.2.10}$$

Therein, $p_{\mathrm{Si,L}}$ and $p_{\mathrm{Si,R}}$ represent the respective timing probabilities in scenarios $\mathrm{Si,L}$ and $\mathrm{Si,R}$.

Table 3.1.: Parameter optimization results regarding the memory trigger module.

| | | | | | | | | | Scenario-Likelihoods | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\sigma_{v_{\mathrm{E,des}}}$ | $N^*_{\mathrm{M}}$ | $\mathcal{U}^*_{\mathrm{T}}$ | $\lambda^*$ | $\gamma^*_1$ | $\gamma^*_2$ | $\gamma^*_3$ | $\overline{\lvert\Delta T^*\rvert}$ | $p^*_{\mathrm{S1}}$ | $p^*_{\mathrm{S2}}$ | $p^*_{\mathrm{S3}}$ | $p^*_{\mathrm{S4}}$ | $p^*_{\mathrm{S5}}$ | $p^*_{\mathrm{S6}}$ | $J^*_{\mathrm{TM}}$ |
| | $[\frac{\mathrm{m}}{\mathrm{s}}]$ | [ ] | [ ] | [ ] | [ ] | [ ] | [ ] | [s] | [ ] | [ ] | [ ] | [ ] | [ ] | [ ] | [ ] |
| Left | 10.0 | 36 | 0.30 | 0.11 | - | - | - | 16.47 | 0.20 | 0.13 | 0.03 | 0.05 | 0.09 | 0.10 | -15.04 |
| Right | 5.5 | 46 | 0.975 | - | 0.95 | 0.825 | 0.25 | 1.18 | 0.22 | 0.16 | 0.17 | 0.18 | - | - | -6.85 |

The above formulation aims at maximizing the joint probability of the timings in all scenarios. First, $\boldsymbol{\xi}_{\mathrm{M,L}}$ and $\boldsymbol{\xi}_{\mathrm{M,R}}$ are optimized using the respective scenarios and therefore datasets from the driving simulator study. Specifically, the optimization is carried out using the kernel density fitted probability distributions. Quantities with a star $()^*$ subsequently denote optimized values. After that, the remaining parts for the accumulator trigger are optimized, again using a log-Likelihood formulation:

$$J_{\mathrm{SA,L}}(\boldsymbol{\xi}_{\mathrm{A,L}}) = \ln(p_{\mathrm{SA,L}}(T_{\mathrm{SA,L}}(\boldsymbol{\xi}_{\mathrm{A,L}}))), \tag{3.2.11}$$

$$J_{\mathrm{TA,L}}(\boldsymbol{\xi}_{\mathrm{A,L}}) = J_{\mathrm{SA,L}}(\boldsymbol{\xi}_{\mathrm{A,L}}) + \sum_{\mathrm{i}=1}^{6} \ln(p_{\mathrm{Si,L}}(T_{\mathrm{Si,L}}(\boldsymbol{\xi}_{\mathrm{A,L}}))), \tag{3.2.12}$$

$$\begin{aligned} \boldsymbol{\xi}^*_{\mathrm{A,L}} = &\underset{\boldsymbol{\xi}_{\mathrm{A,L}}}{\operatorname{argmax}}\, J_{\mathrm{TA,L}}(\boldsymbol{\xi}_{\mathrm{A,L}}) \\ &\text{subject to } T^*_{\mathrm{Si,L,A}} \geq T^*_{\mathrm{Si,L,M}},\ \mathrm{i} = 1, 2, \ldots, 6. \end{aligned} \tag{3.2.13}$$

Note here that a constraint is imposed such that the optimized proposal timings using the accumulator trigger $T^*_{\mathrm{Si,L,A}}$ are greater or equal than the ones of the memory trigger mechanism $T^*_{\mathrm{Si,L,M}}$. The specially designed accumulator scenario is used in the case of the left lane change model. Similarly, for right lane changes:

$$J_{\mathrm{TA,R}}(\boldsymbol{\xi}_{\mathrm{A,R}}) = \sum_{\mathrm{i}=1}^{4} \ln(p_{\mathrm{Si,R}}(T_{\mathrm{Si,R}}(\boldsymbol{\xi}_{\mathrm{A,R}}))) \tag{3.2.14}$$

$$\begin{aligned} \boldsymbol{\xi}^*_{\mathrm{A,R}} = &\underset{\boldsymbol{\xi}_{\mathrm{A,R}}}{\operatorname{argmax}}\, J_{\mathrm{TA,R}}(\boldsymbol{\xi}_{\mathrm{A,R}}) \\ &\text{subject to } T^*_{\mathrm{Si,R,A}} \geq T^*_{\mathrm{Si,R,M}},\ \mathrm{i} = 1, 2, 3, 4. \end{aligned} \tag{3.2.15}$$

Table 3.1 shows the resulting optimized parameter values. The politeness factor $\lambda$ indicates that the lane change intention is mainly driven by dissatisfaction with the current driving lane compared to the potential of the left adjacent lane. This was expected and reasonable since the driving simulator study participants were informed that a subsequent lane change behavior planning module ensures safety. This potentially led to the fact that oncoming vehicles behind the ego-vehicle on the target lane were not that important compared to the other factors in the Equation (3.1.2). However, since $\lambda$ did not vanish, it still helps model the lane change intention more accurately. Further, the Table indicates that left lane change intentions are typically harder to model than those to the right lane. This was already observed during the driving simulator study since all participants eventually passed slow vehicles on the right before signaling their lane change intention. Thinking about overtaking maneuvers, this behavior is reasonable and also observed in real traffic.

Table 3.2.: Parameter optimization results regarding the accumulator trigger module.

| | $\sigma_{v_{\mathrm{E,des}}}$ | $\zeta^*$ | $\mathcal{U}^*_{\mathrm{AT}}$ | $\lambda^*$ | $\gamma^*_1$ | $\gamma^*_2$ | $\gamma^*_3$ | $\overline{\lvert\Delta T^*\rvert}$ | $J^*_{\mathrm{SA}}$ | $J^*_{\mathrm{TA}}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| | $[\frac{\mathrm{m}}{\mathrm{s}}]$ | [ ] | [ ] | [ ] | [ ] | [ ] | [ ] | [s] | [ ] | [ ] |
| Left | 10.0 | 0.03 | 17.37 | 0.11 | - | - | - | 16.48 | 0.23 | -16.58 |
| Right | 5.5 | 0.2395 | 75.26 | - | 0.95 | 0.825 | 0.25 | 7.34 | - | -14.02 |

Using this strategy increases the probability of a successful and comfortable overtake. Signaling the lane change intention too early might confuse and/or influence the vehicle's driver to be overtaken and result in accelerations that prevent being overtaken. Another indication that left lane change intentions are harder to model than the ones to the right is reflected in the higher desired velocity standard deviation $\sigma_{v_{\mathrm{E,des}}}$ in Table 3.1. A higher value means that the model is more uncertain about the lane change decision among all scenarios. The results of the time difference concerning the mean timings of all study participants and all respective scenarios:

$$\overline{|\Delta T^*_L|} = |T^*_{\mathrm{SA}} - \overline{T}_{\mathrm{SA}}| + \sum_{\mathrm{i}=1}^{6} |T^*_{\mathrm{Si,L}} - \overline{T}_{\mathrm{Si,L}}| = 16.47\,\mathrm{s} \tag{3.2.16}$$

and

$$\overline{|\Delta T^*_R|} = \sum_{\mathrm{i}=1}^{4} |T^*_{\mathrm{Si,R}} - \overline{T}_{\mathrm{Si,R}}| = 1.18\,\mathrm{s} \tag{3.2.17}$$

underscore the good performance of the proposed module and reflect the fact that lane change intention to the right is easier to model. The mean deviation per scenario for the left lane changes is $\frac{\overline{|\Delta T^*_L|}}{6} = 2.75\,\mathrm{s}$.

Table 3.2 reports the results of the accumulator parameters. The optimization is done using the specially designed accumulator scenario SA, refer to Table A.1 in the appendix. Note that the memory module does not generate a trigger in this scenario because of comparably low utilities, which is also reflected in Figure 3.6 through the absence of the circle denoting the memory trigger timing $T^*_{\mathrm{M}}$.

Figures 3.5 and 3.7 show the resulting timings when the proposed model is run on the scenarios that are used in the driving simulator study. The results are convincing and reflect the study participant's timing distributions well. Figure 3.6 shows the result of the special test scenario for the optimization of the accumulator. The result is satisfactory since the trigger timing $T^*_{\mathrm{A}}$ is only roughly 0.2 s later than the mean timing of all study participants.

The driving style mainly determines the frequency of lane changes, especially for overtaking slower vehicles. It could be beneficial to group drivers into at least the three classes *defensive*, *neutral* and *aggressive* and obtain optimized model parameters for each class. Even then, the application of the module in the vehicle should allow for easy adjustment of the module to the driver's needs. The memory trigger and accumulator thresholds could be used for this adaption since they directly influence the frequency of lane change proposals.

## 3.3. Evaluation of the Discretionary Lane Change Proposal Module

This Section discusses the application of the optimized model on real data from a ZF Group test vehicle. The vehicle is equipped with four short-range radar sensors at the corners, one long-range radar sensor at the front, and one front-facing camera system. The tracking and fusion system estimates the state vectors of the surrounding traffic participants and their corresponding uncertainties. This Section serves as an additional application illustration of the proposed module but does not imply validity in all traffic situations and for all drivers with various driving styles. All data were post-processed for outlier removal and smoothing to enhance illustration. The data stream consists of recorded data of an overtaking maneuver. First, the vehicle merges onto the highway, and the driver switches on ACC with the current ego-vehicle velocity as set speed. The set speed is increased over time. A slower truck ($v_{\mathrm{CF}} \approx 24\,\mathrm{m/s}$) is in front of the ego-vehicle and the driver eventually decides to overtake it. After the successful overtake, the ego-vehicle drives on the left lane a certain time and eventually changes back to the right lane.

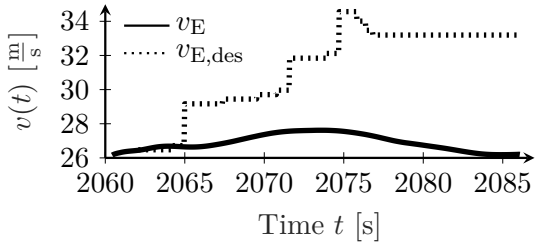
The results for a lane change to the left are shown in the left column of Figure 3.8, specifically Figures (a), (c) and (e). The ego velocity $v_{\mathrm{E}}$ and its desired velocity $v_{\mathrm{E,des}}$ is shown in Figure 3.8 (a). Finally, the utility of a lane change to the left $\mathcal{U}_{\mathrm{L}}$ and accumulator state $\mathcal{U}_{\mathrm{A,L}}$ are shown over the global algorithm time. In the beginning, the ACC is switched off. It is switched on at $t = 2062\,\mathrm{s}$ global algorithm-time and $v_{\mathrm{E,des}} = 26.48$ m/s is set. Note that it is assumed that the desired velocity corresponds exactly to the ACC set speed that the driver chooses using the controls on the steering wheel. Afterward, the desired velocity is increased in certain steps. Finally, at $t = 2077\,\mathrm{s}$, it is set to $v_{\mathrm{E,des}} = 33.20\,\mathrm{m/s}$. The utility $\mathcal{U}_{\mathrm{L}}$ is shown in Figure 3.8 (c). Comparing it to Figure 3.8 (a), the rise starting at $t = 2065\,\mathrm{s}$ clearly corresponds to the jumps in the desired velocity. Figure 3.8 (e) shows the accumulated utility $\mathcal{U}_{\mathrm{A,L}}$. Inspection of Figure 3.8 (c) and (e) reveals that both the triggers based on the memory and accumulator are set roughly 7 s earlier than the turn indicator. Still, the earlier proposal seems appropriate. That is because the truck in front of the ego-vehicle drives with a velocity of roughly $v_{\mathrm{CF}} = 24\,\mathrm{m/s}$. Comparing this to the ACC set speed, refer to Figure 3.8 (a), that is set at $t = 2065\,\mathrm{s}$ to roughly $v_{\mathrm{E,des}} = 30\,\mathrm{m/s}$ and at $t = 2072\,\mathrm{s}$ then to $v_{\mathrm{E,des}} = 32\,\mathrm{m/s}$, there is obviously an utility to change to the left lane in order to drive with the desired velocity, also clearly reflected in Figure 3.8 (c). It can also be argued that the intention to change lanes was probably determined earlier, and the turn indicator was only set after inspection of the traffic situation. Finally, the lane change intention is a hidden variable that cannot be measured. There is always a certain mismatch between the observed variable, the turn indicator state, and the actual intention. Especially in a modular automated driving system, the safety inspection is part of the subsequent lane change behavior planning module such that an earlier proposal can be appropriate. There is a mismatch even when the vehicle is manually driven because the turn indicator state will be switched on after the driver conducts the manual safety inspection.

The left lane change discussed in Figure 3.8 is part of an overtaking maneuver. The

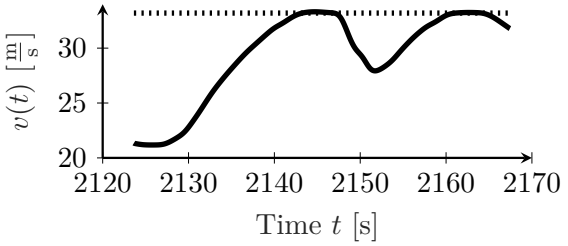
corresponding right lane change and relevant signals are shown in the right column of Figure 3.8, specifically Figures (b), (d), and (f). The desired velocity is still set to $v_{\mathrm{E,des}} = 33.20\,\mathrm{m/s}$ as can be seen in Figure 3.8 (b). There are slower vehicles on the right lane in front of the ego-vehicle. In light of this fact, it is reasonable that the utility $\mathcal{U}_{\mathrm{R}}$ does not reach a condition to trigger the memory module. Part of the reason is that during the driving-simulator study, no lane changes to the right were observed when slower vehicles were driving on the right lane resulting in a rather high threshold for the memory trigger. It can be seen in Figure 3.8 (d) and (f) that the turn indicator is switched to the right at roughly $t = 2147\,\mathrm{s}$ global algorithm-time. At this point, there is still a substantially slower vehicle concerning the desired ego-velocity on the right lane. This vehicle is overtaken first, and the lane change starts later at $t = 2164\,\mathrm{s}$. The accumulator reaches the trigger condition at $t = 2160\,\mathrm{s}$ again, resulting in a satisfactory result for the lane change to the right. In this case, it is questionable if switching on the turn indicator state this early when the intention to overtake a vehicle first is a sensible decision of the driver. In contrast, the postponed proposal seems sensible to not confuse surrounding traffic participants. Algorithm runtimes are of great importance for commercial applications in automated vehicles. The proposed module achieves a runtime of

$$t_{\mathrm{run}} = \mu_{\mathrm{run}} \pm \sigma_{\mathrm{run}} = 3.48\,\mu\mathrm{s} \pm 1.77\,\mu\mathrm{s} \tag{3.3.1}$$

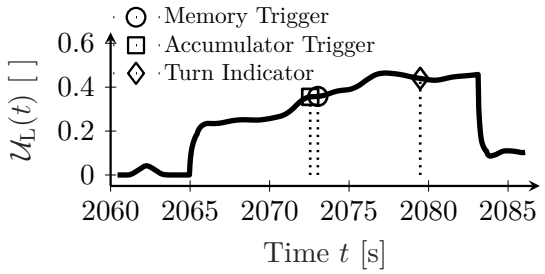
measured on a standard desktop PC (i5-6500, 16GB RAM). This indicates the low computational complexity and allows for easy integration into automated vehicle software systems.
It is emphasized that the above illustration does not substitute a complete acceptance test of the proposed module. A much larger dataset is needed for the optimization and validation. Finally, comparing timings of the turn indicator to the proposals of the module is also not the ideal metric to assess the performance of the proposed module, as was argued in the work at hand. This is because the turn indicator is switched on after a safety evaluation by the driver. Comparing this to the task description that was given to the driving simulator participants, refer to Section A.3, the mismatch becomes obvious. Instead, the acceptance of the module by the driver of the automated vehicle is of much higher importance. This needs to be analyzed in the future.



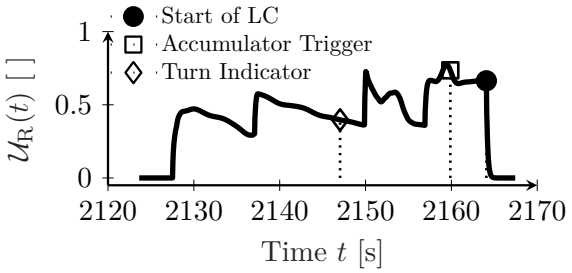

(a) Current and desired ego velocity regarding the left lane change.



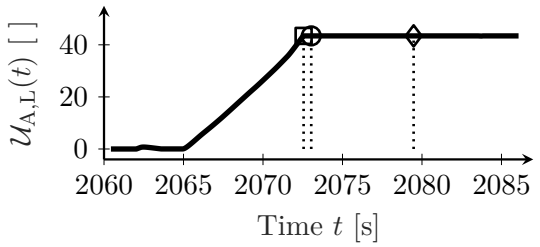

(b) Current and desired ego velocity regarding the right lane change.



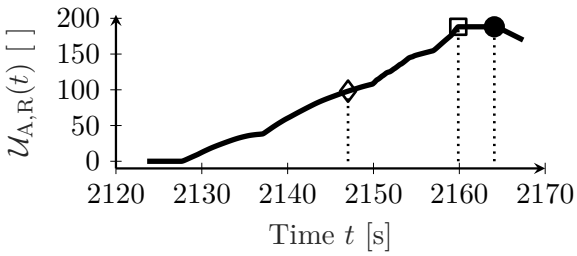

(c) Utility for a lane change to the left lane.




(d) Utility for a lane change to the right lane.




(e) Accumulator state for a lane change to the left lane.

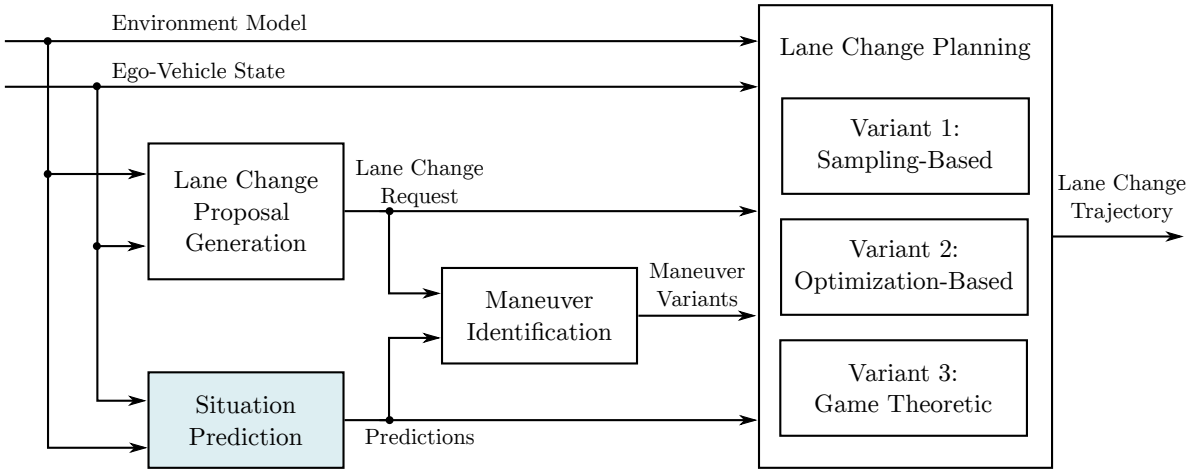



(f) Accumulator state for a lane change to the right lane.

Figure 3.8.: Results of the optimized lane change module applied on real data from a ZF Group test vehicle. The left and right column represent a lane change to left and right respectively. They are part of an overtaking maneuver.

# 4

# Uncertainty-Aware Situation Prediction




Accurate trajectory predictions are crucial for automated driving software systems. Using this information can increase passenger's comfort due to timely reaction to possible scene evolutions. Most approaches in the literature predict on an individual traffic participant basis. There are two problems with that. The first is the loss of causality leading to the prediction of collisions, as will be explained in this Chapter. Furthermore, an unnecessary combinatoric explosion happens that makes behavior planning more difficult. This is because it is not obvious how to handle the individual predictions and multimodality. Framing the problem instead as a situation prediction means that trajectory ensembles are predicted instead. For such interactive situation prediction, the existing concept [Wis+18] is further developed to remove some of its disadvantages. It is an approach based on Monte Carlo simulations for highway situations. All surrounding vehicles are modeled using longitudinal car-following and lane change models from the field of microscopic traffic simulation. The model parameters are represented using probability distributions. Finally, the parameters for the simulations are probabilistically assigned values according to the distributions. [Wis+18] is a vehicle-specific trajectory prediction. Scientifically, the next step thus aims at transforming it into a situation prediction.

This can be done by topological clustering of the overall situation. Multimodality is reflected in topologically different situation developments. The location of the module is shown in the above figure.

## 4.1. Prediction Architecture Adaptions

This Section first starts with a general high-level discussion of the situation prediction algorithm. More details can be found in [Wis+18]. Afterward, the most relevant adaptions of the original concept are discussed. Those include a reworked clustering mechanism and a different output representation.

**Monte Carlo Simulations**

In the following, a trajectory is defined as the following set $\tau$:

$$\tau = \{\begin{bmatrix} L(t_0) & N(t_0)) \end{bmatrix}^\intercal, \begin{bmatrix} L(t_1) & N(t_1)) \end{bmatrix}^\intercal, \ldots, \begin{bmatrix} L(t_{n_\tau}), N(t_{n_\tau}) \end{bmatrix}^\intercal\}. \tag{4.1.1}$$

Here $t_k$ corresponds to a discrete-time variable on a uniform grid, $L$ represents the longitudinal and $N$ the lateral coordinate in a curvilinear coordinate system. A total of $n_\tau + 1$ points are used. Let the number of surrounding road users be $n_\mathrm{O}$ and thus the total number of road users including the ego-vehicle $n_\mathrm{V} = 1 + n_\mathrm{O}$. Next, behavioral models for the vehicles are specified. They typically are taken from the field of microscopic traffic simulation. For example, a well-tested combination is the IDM (Intelligent Driver Model) [Tre+00] vehicle following model together with the MOBIL (Minimizing Overall Braking decelerations Induced by Lane changes) lane change model [TK09]. These are used in [Wis+18]. Now let $\boldsymbol{\theta}_i$ be the set of all model parameters of the selected behavioral models of vehicle $i$. These parameters usually have interpretable meanings and are thus readily parameterizable for a large class of models. In particular, it is possible to use them to represent defensive and offensive behaviors. In [Wis+18], vehicle-individual trajectory prediction was achieved by randomizing the parameters $\boldsymbol{\theta}_i$ of all agents in terms of Monte Carlo simulations using a total of $n_\mathrm{M}$ simulations. In the following, the main steps as used in the thesis at hand are formulated mathematically.
First, a **L**ane **C**hange **P**rediction (LCP) is performed for all surrounding road users. The result is a multinoulli distribution represented by the random vector:

$$\mathbf{M} = \begin{bmatrix} P(LK) & P(LCL) & P(LCR) \end{bmatrix}^\intercal \tag{4.1.2}$$

with the probabilities $P(LK)$ of lane-keeping, $P(LCL)$ of lane change left, and $P(LCR)$ of lane change right. The resulting probabilities are considered in the Monte Carlo simulation so that a very likely lane change is actually executed in the simulation. Subsequently, a randomization of the model parameters of all agents is performed. For this purpose, probability density functions are defined

$$p(\boldsymbol{\theta}_i) = p(\begin{bmatrix} v_{\mathrm{des},i} & \lambda_i \end{bmatrix}^\intercal) \quad \forall \quad i = 1, 2, \ldots, n_\mathrm{V}, \tag{4.1.3}$$

with the IDM desired velocity parameter $v_{\text{des},i}$ and MOBIL politeness $\lambda_i$ for each vehicle $i$. Both parameters are sampled using truncated gaussian distributions with empirically set boundaries:

$$\begin{aligned} p(\boldsymbol{\theta}_i) &= p(v_{\text{des},i}) \cdot p(\lambda_i) \\ &= \text{trunc}(\mathcal{N}(\mu_{v_{\text{des}}}, 0.1\mu_{v_{\text{des}}}), 0.85\mu_{v_{\text{des}}}, 1.15\mu_{v_{\text{des}}}) \cdot \text{trunc}(\mathcal{N}(0.35, 0.15), 0.0, 0.7). \end{aligned} \tag{4.1.4}$$

The truncation operator:

$$\tilde{p}(\theta) = \text{trunc}(p(\theta), \min(\theta), \max(\theta)) \tag{4.1.5}$$

allows to impose boundaries on a randomized parameter, here $\theta$. $\tilde{p}(\theta)$ is zero below $\min(\theta)$ and above $\max(\theta)$ and it is rescaled within the interval $[\min(\theta), \max(\theta)]$ to ensure that the integral of $\tilde{p}(\theta)$ is 1. Accordingly, $n_{\text{M}}$ times model parameters are drawn for each agent. Thus, the result of the Monte Carlo simulations is a total of $n_{\text{M}}$ trajectories $\tau_i$ for each vehicle $i$. Furthermore, each Monte Carlo run is collision-free since the IDM model ensures this.
Lastly, it is possible to force the ego-vehicle to do a lane change that fulfills certain constraints on Time Headway and Time-to-Collisions. It starts the lane change at the earliest possible moment when the constraints are fulfilled. That realizes counterfactual reasoning and answers the question: *what if?* from the perspective of the ego-vehicle. Moreover, as will be shown in Chapter 8, it enhances robustness since breaking maneuvers of following vehicles on the target lane are anticipated.

**Situation Clustering and Uncertainty-Awareness**

First, the orginal clustering mechanism of [Wis+18] based on the DBSCAN algorithm, refer to [Est+96], is discussed. There, a feature matrix $\mathbf{F}_{\tau,i}$ is created for each vehicle $i$. The matrix has $n_{\text{M}}$ rows, one for each Monte Carlo simulation. It has two columns. The first column holds the final longitudinal coordinate $L_{i,j}(t_{n_{\text{k}}})$ of vehicle $i$ and Monte Carlo simulation $j$. The second column similarly holds the respective lateral coordinate $N_{i,j}(t_{n_{\text{k}}})$. DBSCAN is then applied to the feature matrices $\mathbf{F}_{\tau,i}$ for all vehicles $i$. Each Monte Carlo simulation $j$ is assigned to one cluster. The clusters $l$ are represented using a latent variable $\mathbf{Z}_i = \begin{bmatrix} Z_{i,1} & Z_{i,2} & \dots & Z_{i,n_{\text{c},i}} \end{bmatrix}^\intercal$ for each vehicle $i$. The empirical probability of a cluster $l$ is then given by:

$$P(Z_{i,l}) = \frac{n_{i,l}}{n_{\text{M}}}, \tag{4.1.6}$$

with $n_{i,l}$ the total number of Monte Carlo simulations belonging to cluster $l$. Hence a multinoulli probability distribution results for the latent variable $\mathbf{Z}_i$. The mean and variance of the longitudinal and lateral positions as well as velocities are estimated based on all trajectories belonging to the cluster per timestep $k$. Above clustering mechanism results in vehicle-individual trajectory predictions $\hat{\tau}_{i,l}$.
The following example illustrates the problems of such vehicle individual trajectory predictions. Consider the traffic scene in Figure 4.1. The semi-transparent configuration corresponds to the initial scene. Vehicle $\text{O}_1$ on the rightmost lane is assumed to drive with a lower velocity compared to all other vehicles in the scene. This might motivate

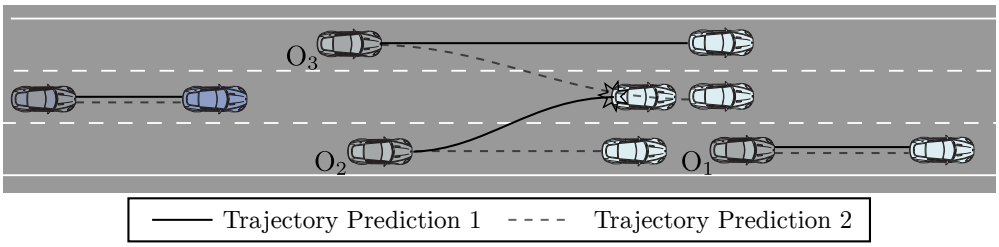



Figure 4.1.: Initial traffic scene (semitransparent) and its evolution with a total of four vehicles. The ego-vehicle is depicted in dark-blue and surrounding vehicles in light-blue. Two possible trajectory predictons are shown for vehicles $O_2$ and $O_3$.

vehicle $O_2$ to do a lane change to the middle lane. Vehicle $O_3$ wants to prevent a collision, and causality ensures that if $O_2$ changes lane, $O_3$ will remain on its lane. On the other hand, if $O_2$ decided to rather slow down instead of changing lane, $O_3$ might adhere to the german right driving rule and change to the middle lane.
The above description was based on causality and trajectory ensembles, and both variants seem likely. Prediction on a vehicle-individual basis removes the above causal reasoning. In this case, one is left with two trajectory predictions for both vehicles $O_2$ and $O_3$ represented with different line styles in Figure 4.1. It is obvious that in this case, trajectory prediction 1 of vehicle $O_2$ collides with the prediction 2 of vehicle $O_3$. This effect was observed and analyzed in [Wis20].
Above problems with predicted collisions do not arise with the situation clustering approach proposed in the thesis at hand. For example, Figure 4.2 illustrates an on-ramp merging traffic scene consisting of four vehicles. In both situation prediction 1 and 2, vehicle $O_1$ changes lanes to enter the highway. However, the reaction of vehicle $O_2$ differs among both situation predictions. In one case, $O_2$ brakes to adapt its velocity to the merging vehicle $O_1$. However, in situation prediction 2, $O_2$ decides to change lane to the left to avoid breaking.
The developed situation clustering mechanism can distinguish such situation evolutions correctly. It uses a very simple mechanism as follows. Assume that the trajectories for all vehicles $i$ and all Monte Carlo simulations $j$ are available. The first step consists of the construction of a feature matrix $\mathbf{F}_\mathrm{S}$. Its rows correspond to the Monte Carlo simulations $j$. For each simulation $j$, it consists of a total of two columns per vehicle, hence $2 \cdot n_\mathrm{V}$ in total. These are the vehicles final lane ID $ID_{\mathrm{Lane},i,j}(t_{n_\mathrm{k}})$ and the ID of its leader vehicle (0 if none) $ID_{\mathrm{Leader},i,j}(t_{n_\mathrm{k}})$ with the final time $t_{n_\mathrm{k}}$, refer to Figure 4.2. Instead of applying a clustering algorithm on the feature matrix $\mathbf{F}_\mathrm{S}$, its unique rows and their frequency are determined. A user-defined number $n_\mathrm{C}$ of situation clusters with the highest frequency of occurrence are chosen from all unique ones.
Hence there is exactly one latent variable $\mathbf{Z}_\mathrm{S} = \begin{bmatrix} Z_{\mathrm{S},1} & Z_{\mathrm{S},2} & \dots & Z_{\mathrm{S},n_\mathrm{c}} \end{bmatrix}^\mathsf{T}$ for the whole situation that follows a multinoulli probability distribution $P(\mathbf{Z}_\mathrm{S})$. The probability mass of cluster $l$ is calculated as:

$$P(Z_{\mathrm{S},l}) = \frac{n_l}{n_\mathrm{M}}. \tag{4.1.7}$$

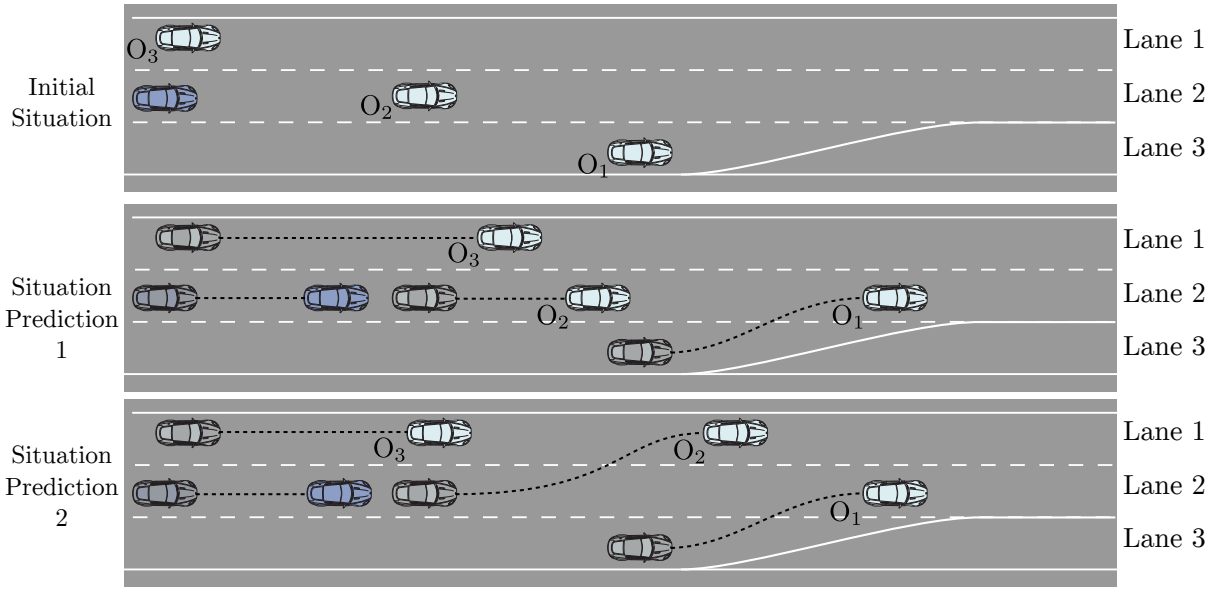



Figure 4.2.: Illustration of an initial highway on-ramp traffic situation. Vehicle $O_1$ wants to merge onto the highway. There are two typical evolutions of this traffic scene, one where $O_2$ assists merging by changing lane and the other where it breaks to adapt its velocity and create a gap. This results in two distinct situation predictions that are distinguished by the final vehicle configuration on all lanes.

Likewise, a joint probability density function for the trajectories of all agents results for each mode $l$ of the multinoulli distribution:

$$p(\hat{\tau}_{0,l}, \hat{\tau}_{1,l}, \ldots, \hat{\tau}_{n_\mathrm{V},l} | Z_{\mathrm{S},l}). \tag{4.1.8}$$

In it, $\hat{\tau}_{i,l}$ represents the predicted trajectory of vehicle $i$. The index 0 represents the ego-vehicle. The latent variable $Z_{\mathrm{S},l}$ reflects the situation mode $l$. The outcome of the Monte Carlo simulation is used to calculate the mean and variance of the predicted positions independently for the longitudinal and lateral coordinate per situation mode $l$.
Furthermore, a multinoulli distribution results for the future lane ID ($ID_{FS,i}$) of the road users:

$$p(ID_{FS,i} | Z_{\mathrm{S},l}). \tag{4.1.9}$$

This information is valuable for uncertainty-aware behavior planning to derive safety constraints. It could also be used in a lane change timing heuristic to aid the behavior planning of the ego-vehicle.

## 4.2. Desired Velocity Regression

This Section aims to enhance the original work [Wis+18] by incorporation of a desired velocity regression. It is developed and evaluated using the highD dataset. Originally, the situation prediction concept makes a rather strong assumption that all traffic participants want to keep their currently observed velocity. One situation where the adaption of the

desired velocity might be beneficial is when predicting overtaking maneuvers. This is because the desired velocity is also indirectly used in the MOBIL lane change model to trigger a lane change, refer to [TK09] for more details. MOBIL will likely issue a lane change to overtake if a vehicle follows a slower vehicle that drives with a substantially lower velocity than the desired velocity.
The above ideas motivate the development of a mechanism to adapt the probability distribution $p(v_{\text{des},i})$ of each vehicle $i$ used in the Monte Carlo simulations to increase the situation prediction accuracy. The approach is developed using the highD dataset, refer to [Kra+18]. Its influence on the prediction accuracy will be described in Section 4.3 whereas, in this Section, the regression problem and results are the focus.
The regression task is defined as follows:

$$v_{\text{des},j} = f_{\boldsymbol{\gamma}}(\mathbf{f}_{v,j}) + \epsilon_j = \hat{v}_{\text{des},j} + \epsilon_j. \tag{4.2.1}$$

Herein $j$ refers to the index of the data, $\mathbf{f}_{v,j}$ corresponds to the input features and $f_{\boldsymbol{\gamma}}(\mathbf{f}_{v,j})$ is the regression function depending on a parameter vector $\boldsymbol{\gamma}$. Finally $\epsilon_j$ accounts for modelling errors due to the particular choice of $f_{\boldsymbol{\gamma}}$ and statistical noise in the data. The function $f$ is determined by the choice of method. In the thesis at hand, a **S**upport **V**ector **M**achine (SVM) [CV95], its Bayesian counterpart **R**elevance **V**ector **M**achine (RVM) [Tip01] and three ensemble methods are used. The chosen ensemble methods are **G**radient **B**oosted **T**rees (GBT) [Fri00], a **R**andom **F**orest (RF) [Bre01] and **E**xtra **Trees** (ET) [Geu+06]. Their implementation from the scikit-learn [Ped+11] library are used. Each of above methods have different parameter vectors $\boldsymbol{\gamma}$. Those are optimized using five-fold crossvalidation and a grid search. The optimization problem is the following:

$$\boldsymbol{\gamma}^* = \underset{\boldsymbol{\gamma}}{\operatorname{argmin}}\, f_L(\{v_{\text{des},j}\}_{j=1}^{n_{\text{D}}}, \{f_{\boldsymbol{\gamma}}(\mathbf{f}_{v,j})\}_{j=1}^{n_{\text{D}}}) \tag{4.2.2}$$

$$= \underset{\boldsymbol{\gamma}}{\operatorname{argmin}}\, f_L(\{v_{\text{des},j}\}_{j=1}^{n_{\text{D}}}, \{\hat{v}_{\text{des},j}\}_{j=1}^{n_{\text{D}}}) \tag{4.2.3}$$

with a chosen loss function $f_L$. Most commonly, a least squares loss is chosen:

$$f_L(\{v_{\text{des},j}\}_{j=1}^{n_{\text{D}}}, \{\hat{v}_{\text{des},j}\}_{j=1}^{n_{\text{D}}}) = \sum_{j=1}^{n_{\text{D}}} (v_{\text{des},j} - \hat{v}_{\text{des},j})^2. \tag{4.2.4}$$

Another very common one is the Huber loss [Hub64].
Refer next to Figure 4.3 for an illustration of the desired velocity regression task and dataset creation based on the highD data. It contains traffic scenes of a 400 m two or three-lane straight highway stretch. A dataset consisting of 6557 training and 1640 test samples is created. The vehicle that should be predicted is referred to as the subject vehicle. The vehicle needs to be observed for at least 2 s to build up the features. There are the same amount of lane changes and lane-keeping maneuvers in the dataset for dataset balancing reasons. The target desired velocity $v_{\text{des}}$ is taken to be the last observed velocity of a vehicle. Figure 4.3 shows one typical example where the subject vehicle overtakes the slower vehicle $\text{O}_2$ and speeds up to match the speed of vehicle $\text{O}_1$. On the other hand, $\text{O}_1$ breaks slightly to show courtesy.
The goal of the regression is to not only predict a point estimate $\hat{v}_{\text{des}} = \hat{\mu}_{v_{\text{des}}}$ of the desired velocity but instead also the standard deviation $\hat{\sigma}_{v_{\text{des}}}$. Together, both estimates comprise

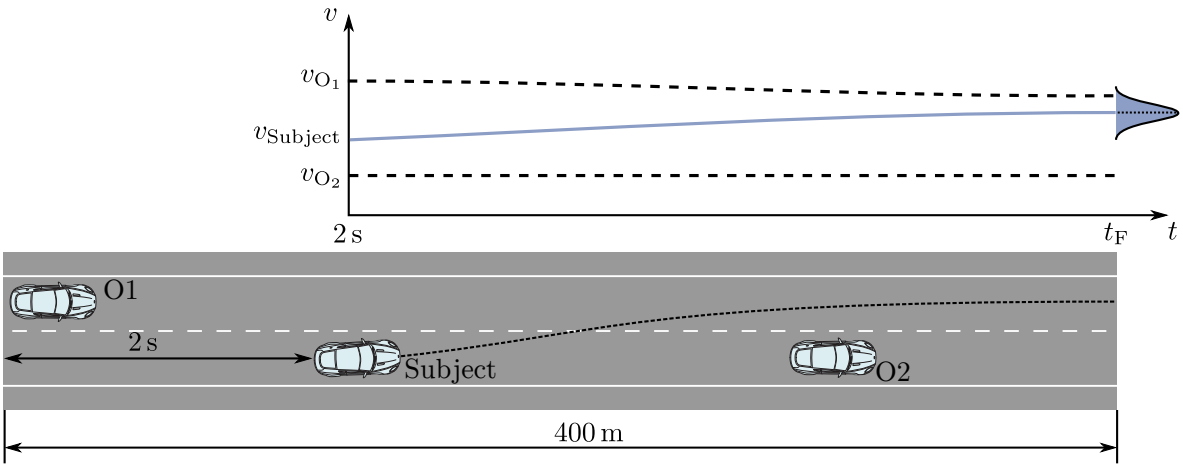



Figure 4.3.: Illustration of the desired velocity regression task using a typical highway driving situation. The subject vehicle overtakes a slower vehicle $O_2$ and adapts its speed accordingly. It is the goal to predict a gaussian probability distribution that can be used in the Monte Carlo situation prediction framework.

the sufficient statistics of a Gaussian probability distribution $p(\hat{v}_{\text{des}})$ that in turn can be used for sampling within the Monte Carlo situation prediction framework described above. Not only the choice of method for the regression is important, but also the chosen features $\mathbf{f}_{\text{v,j}}$ play a crucial role. Three different types of features are used. First, the dynamic features at time $t_k$ are represented by the vector:

$$\mathbf{f}_{\text{dyn},t_k} = \begin{bmatrix} y_{t_k} - y_{\text{CLC},t_k} & \dot{x}_{t_k} & \dot{y}_{t_k} & \ddot{x}_{t_k} & \ddot{y}_{t_k} \end{bmatrix}^\intercal. \tag{4.2.5}$$

All quantities are referenced in the global highD coordinate frame, and $y_{\text{CLC},t_k}$ represents the $y$ coordinate of the center of the current lane at time $t_k$. A compact representation of the relations to other surrounding vehicles are obtained using Time-to-Collisons ($TTC$):

$$\mathbf{f}_{\text{TTC},t_k} = \begin{bmatrix} TTC_{\text{F},t_k} & TTC_{\text{B},t_k} & TTC_{\text{LF},t_k} & TTC_{\text{LA},t_k} & TTC_{\text{LB},t_k} & TTC_{\text{RF},t_k} & TTC_{\text{RA},t_k} & TTC_{\text{RB},t_k} \end{bmatrix}^\intercal \tag{4.2.6}$$

where F, B and A refer to front, back and alongside whereas L and R refer to left and right. It uses the following definition in global coordinates at a query time $t_{\text{q}}$:

$$TTC(t_{\text{q}}) = \frac{x_{\text{L}}(t_{\text{q}}) - x_{\text{F}}(t_{\text{q}})}{\dot{x}_{\text{F}}(t_{\text{q}}) - \dot{x}_{\text{x}}(t_{\text{q}})}, \tag{4.2.7}$$

with F and L denoting the follower and leader vehicle respectively. A default value of 1000 is used in case of missing ROI vehicles or infinite Time-to-Collision values.
Finally, the third feature is the vehicle type $VT$ which is set to 0 for cars and to $-1$ for trucks. Above features are evaluated at certain timesteps with a 0.2 s stepsize as follows:

$$t_k = -k \cdot 0.2\,\text{s}, \quad k = 0, 1, \ldots, n_{\text{h}} = 10. \tag{4.2.8}$$

Table 4.1.: Results of the desired velocity prediction. Abbrevitations: SVM: Support Vector Machine, RVM: Relevance Vector Machine, RF: Random Forest, ET: Extra Trees, GBT: Gradient Boosted Trees.

| Regressor | Estimators | Exp. Var $[\frac{m^2}{s^2}]$ | MAE $[\frac{m}{s}]$ | RMSE $[\frac{m}{s}]$ | MedAE $[\frac{m}{s}]$ | $R^2$ $[\,]$ | $J_{v_{\text{des}}}$ $[\,]$ |
|---|---|---|---|---|---|---|---|
| $\hat{v}_{\text{des},j} = v_{0,j}$ | - | 0.86 | 1.91 | 2.77 | 1.37 | 0.84 | - |
| SVM | - | 0.89 | 1.39 | 2.26 | 0.82 | 0.89 | - |
| RVM | - | 0.88 | 1.51 | 2.38 | 0.95 | 0.88 | -2.43 |
| RF | 250 | 0.91 | 1.30 | 2.03 | 0.79 | **0.91** | -1.98 |
| ET | 300 | 0.91 | 1.31 | 2.07 | 0.83 | **0.91** | -1.97 |
| GBT | 250 | **0.92** | **1.23** | **2.01** | **0.75** | **0.91** | **-2.60** |

The complete feature vector for example $j$ is hence:

$$\mathbf{f}_{\text{v,j}} = \begin{bmatrix} VT & \mathbf{f}_{\text{dyn},t_1}^\intercal & \mathbf{f}_{\text{TTC},t_1}^\intercal & \mathbf{f}_{\text{dyn},t_2}^\intercal & \mathbf{f}_{\text{TTC},t_2}^\intercal & \dots & \mathbf{f}_{\text{dyn},t_{10}}^\intercal & \mathbf{f}_{\text{TTC},t_{10}}^\intercal \end{bmatrix}^\intercal, \tag{4.2.9}$$

and uses a total history of 2 s. The implementation of the feature calculation is built upon [Kra20] and [Chi19].
The results of the desired velocity regression task are shown in Table 4.1. All methods are compared to the baseline when always predicting the initial velocity $\hat{v}_{\text{des},j} = v_{0,j}$. The Table shows that all learning-based methods outperform the baseline in all metrics. For mathematical definitions of these standard regression metrics, refer to appendix A.5. Gradient Boosted Trees perform best on all metrics. The Relevance Vector Machine performs only slightly worse on the probabilistic $J_{v_{\text{des}}}$ metric defined as follows for Gaussian probability distributions:

$$J_{v_{\text{des}}} = \sum_{i=1}^{n_{\mathcal{D}}} -\frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln(\sigma_{\hat{v}_{\text{des}}}^2) - \frac{1}{2\sigma_{\hat{v}_{\text{des}}}}(v_{\text{des},i} - \hat{v}_{\text{des},i})^2 \tag{4.2.10}$$

compared to Gradient Boosted Trees. Both Random Forest and Extra Trees perform worse in $J_{v_{\text{des}}}$ sense. Closer inspection revealed that this is because both approaches are more uncertain in their predictions and provide a comparably high predicted variance. Due to the results, the influence of using Gradient Boosted Trees for generating the desired velocity probability distribution for use in Monte Carlo simulations will be evaluated in the next Section 4.3 on the highD dataset.

## 4.3. Evaluation of the Situation Prediction

This Section starts with an evaluation using simulation data to compare the modified Monte Carlo prediction model with the original one described in [Wis+18]. It will be shown that both approaches have similar prediction accuracies on simulation data. At the same time, the adapted variant removes the disadvantage of predicting collisions and avoiding combinatoric explosions due to using only one latent variable $\mathbf{Z}_{\text{S}}$.

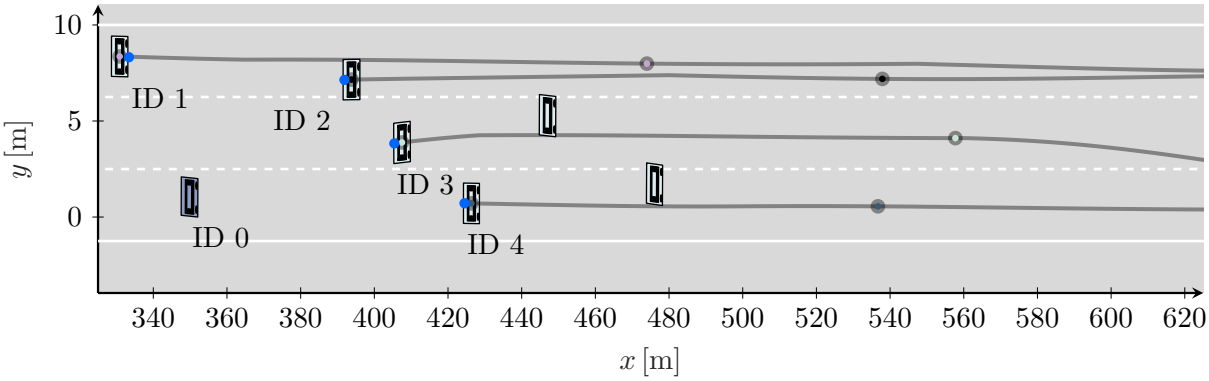



Figure 4.4.: Snapshot of one examplary three lane highway traffic scenario with a total of seven vehicles. The dark-blue vehicle with ID 0 represents the ego-vehicle that observes the traffic scene. Only detected vehicles marked with a blue dot are predicted. Their driven trajectories are shown and colored dots are placed with 5 s spacing.

Table 4.2.: Configurations of the Monte Carlo prediction approach used for the evaluation on the simulation dataset.

| | Situation Prediction | | | | Trajectory Prediction | | | |
|---|---|---|---|---|---|---|---|---|
| Configuration | C1-S | C2-S | C3-S | C4-S | C1-S | C2-S | C3-S | C4-S |
| Monte Carlo simulations | 50 | 50 | 25 | 25 | 50 | 50 | 25 | 25 |
| LCP Treshold | None | 0.65 | None | 0.65 | None | 0.65 | None | 0.65 |

### Evaluation using Simulation Data

The evaluation is based on data that is created using the DESIM simulation environment, refer to [Wis+16]. Variability is introduced by using probability distributions to alter the initial placement of all vehicles in the scene and their driver model parameters. A total of 3529 trajectories are created this way. One example traffic scene is shown in Figure 4.4. Therein also the driven trajectories of the predicted vehicles are shown.

The original work is called trajectory prediction due to its vehicle-individual predictions, whereas the adapted model is called situation prediction. A total of four different configurations, C1-S to C4-S, are evaluated as shown in Table 4.2. The letter S reflects that evaluation is conducted using simulation data compared to the results presented later using the highD dataset. Two parameters are varied. One is the number of Monte Carlo simulations, using 25 and 50. The other is an imposed threshold on the lane change prediction results. If no threshold is imposed, then the original mechanism described in [Wis+18] is used that samples initial lane change trajectories for the vehicles based on the probabilities of the classes **L**ane-**K**eeping $P(LK)$, **L**ane **C**hange **L**eft $P(LCL)$ and **L**ane **C**hange **R**ight $P(LCR)$. If a threshold is imposed, that means that if the probability for a specific class exceeds it, that specific maneuver is always imposed initially.

Two baseline methods are also included in the evaluation. One is a **C**onstant **V**elocity (CV) prediction that takes into account the heading of the vehicles. Due to the sensor simulation, there is noise in the estimated heading angle, increasing lateral errors over the prediction horizon. To circumvent this error and allow for a fair comparison,

the second baseline approach neglects the heading and predicts **C**onstant **V**elocity with **L**ane-**K**eeping (CV-LK). Predictions are evaluated up to 10 s. The evaluation results of configurations C3-S and C4-S with reduced number of Monte Carlo simulations are provided in the appendix A.3, Table A.3. A probabilistic metric abbreviated $P(\tau_i|\{\hat{\tau}_i\})$ is defined as follows for the trajectory prediction of vehicle $i$:

$$P(\tau_i|\{\hat{\tau}_i\}) = P(\tau_i|\{\hat{\tau}_i\}_{l=1}^{n_{\mathrm{c},i}}) = \sum_{l=1}^{n_{\mathrm{c},i}} P(Z_{i,l})P(\tau_i|\hat{\tau}_{i,l}) \tag{4.3.1}$$

and for the situation prediction:

$$P(\tau_i|\{\hat{\tau}_i\}) = P(\tau_i|\{\hat{\tau}_i\}_{l=1}^{n_{\mathrm{c}}}) = \sum_{l=1}^{n_{\mathrm{c}}} P(Z_{S,l})P(\tau_i|\hat{\tau}_{i,l}) \tag{4.3.2}$$

Its interpretation is the probability of the driven trajectory $\tau_i$ given the predictions $\hat{\tau}_i$. On the other hand, the notation $P(\tau|\{\hat{\tau}\})$ is used when the set of trajectories of all vehicles is considered and is used in subsequent results tables 4.3 and 4.7. Note also that the $RMSE$ and $MAE$ metrics provided in the tables use the probabilities of the situation modes $P(Z_{S,l})$ for weighting and hence correctly consider the multimodality.
Table 4.3 shows that both the situation prediction and trajectory prediction provide comparable results for their respective configurations. Imposing a lane change prediction threshold leads to better results on simulation data. Notably, the lateral errors of the constant velocity baseline exceed the errors of all other approaches. While the situation and trajectory prediction approaches using Monte Carlo simulations perform better in the probability metric, they lack behind the constant velocity lane-keeping baseline in almost all other metrics. It is concluded that this is a particularly strong baseline and the reason why this approach is often applied in practice. However, it is interaction-free and cannot be used for counterfactual reasoning, as it is used later in Chapter 6 to derive an interaction cost. For interaction-aware lane change planning, the situation prediction approach is hence the most suitable.

### Evaluation using highD Data

The highD dataset, refer to [Kra+18], is a dataset of naturalisitic vehicle trajectories recorded on german highways using a dronge. A straight stretch of a german highway is recorded spanning a total of 420 m distance. Vehicles are tracked and using image processing and the vehicle states including positions and velocities are provided for each frame in the dataset. The usage of this data allows the comparison of the situation prediction performance of the approach described in this Section to other variants from literature.
For the evaluation of the prediction methods on the highD dataset, a lane change classifier needs to be developed first, refer to the original work [Wis+18]. Former situation prediction results use the classifier from [Wis+18] that is trained on simulation data. The dataset used for this task is shown in Table 4.4. It is approximately balanced using undersampling of the majority lane-keeping class.
The metrics are described in the appendix A.5 and consist of the standard ones used for the evaluation of classifiers. As in the case of the desired velocity prediction described

Table 4.3.: Results of the evaluation on a total of 3529 trajectories using the DESIM simulation environment. The mean and standard deviation is provided for all metrics.

| | | | | Situation Prediction | | Trajectory Prediction | |
|---|---|---|---|---|---|---|---|
| Metric | Time [s] | CV | CV-LK | C1-S | C2-S | C1-S | C2-S |
| RMSE [m] | 1 | **0.32±0.33** | 0.49±0.41 | 0.52±0.45 | 0.51±0.45 | 0.50±0.46 | 0.48±0.45 |
| | 2.5 | 1.81±1.66 | **1.74±1.66** | 1.92±1.74 | 1.84±1.70 | 1.88±1.73 | 1.77±1.70 |
| | 5 | 6.05±5.52 | **5.29±5.26** | 5.83±5.20 | 5.58±5.10 | 5.80±5.22 | 5.48±5.15 |
| | 7.5 | 11.47±10.38 | **9.99±9.80** | 11.04±9.71 | 10.70±9.60 | 10.96±9.75 | 10.50±9.68 |
| | 10 | 17.7±15.85 | **15.56±14.90** | 17.32±14.92 | 16.87±14.80 | 17.12±15.01 | 16.53±14.90 |
| $MAE_x$ [m] | 1 | **0.22±0.30** | **0.22±0.30** | 0.26±0.36 | 0.25±0.35 | 0.26±0.36 | 0.25±0.35 |
| | 2.5 | **1.25±1.51** | **1.25±1.51** | 1.42±1.60 | 1.36±1.54 | 1.41±1.60 | 1.33±1.54 |
| | 5 | 4.49±5.02 | **4.48±5.02** | 4.94±4.93 | 4.78±4.84 | 4.88±4.93 | 4.67±4.86 |
| | 7.5 | 8.95±9.53 | **8.94±9.53** | 9.89±9.39 | 9.62±9.29 | 9.75±9.41 | 9.40±9.33 |
| | 10 | 14.24±14.66 | **14.22±14.66** | 15.91±14.61 | 15.51±14.51 | 15.66±14.68 | 15.16±14.58 |
| $MAE_y$ [m] | 1 | **0.10±0.1** | 0.27±0.26 | 0.25±0.24 | 0.26±0.26 | 0.23±0.26 | 0.23±0.26 |
| | 2.5 | 0.56±0.57 | 0.50±0.57 | 0.50±0.53 | 0.48±0.54 | 0.47±0.51 | **0.44±0.55** |
| | 5 | 1.56±1.53 | **0.81±1.01** | 0.89±1.04 | **0.81±1.05** | 0.92±1.03 | **0.81±1.09** |
| | 7.5 | 2.52±2.36 | **1.05±1.26** | 1.15±1.27 | 1.08±1.30 | 1.21±1.22 | 1.10±1.30 |
| | 10 | 3.46±3.17 | **1.34±1.53** | 1.42±1.49 | 1.36±1.52 | 1.46±1.42 | 1.37±1.50 |
| $\overline{\lvert P(\tau\lvert\hat{\tau})\rvert}$ [ ] | 1 | 0.60±0.49 | 0.55±0.48 | 1.78±1.78 | **1.80±1.82** | 1.75±1.70 | 1.79±1.75 |
| | 2.5 | 0.10±0.10 | 0.10±0.10 | 0.23±0.28 | **0.24±0.32** | 0.22±0.24 | **0.24±0.28** |
| | 5 | 0.02±0.03 | 0.02±0.03 | **0.05±0.03** | **0.05±0.03** | **0.05±0.03** | **0.05±0.03** |
| | 7.5 | 0.01±0.02 | 0.01±0.02 | **0.02±0.02** | **0.02±0.02** | **0.02±0.02** | **0.02±0.02** |
| | 10 | 0.00±0.01 | 0.00±0.01 | **0.01±0.01** | **0.01±0.01** | **0.01±0.01** | **0.01±0.01** |

Table 4.4.: Dataset based on highD data used to train lane change prediction classifiers.

| | LK | LCL | LCR |
|---|---|---|---|
| Train | 21552 | 16905 | 2401 |
| Test | 5434 | 4061 | 720 |

Table 4.5.: Results of various classifiers based on the test data.

| | | LK | | | LCL | | | LCR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Classifier | Estimators | Prec. | Rec. | F1 | Prec. | Rec. | F1 | Prec. | Rec. | F1 | ACC | BAC |
| SVM | - | 0.94 | 0.89 | 0.91 | 0.92 | 0.93 | 0.93 | 0.70 | **0.94** | 0.81 | 0.91 | 0.92 |
| RF | 300 | 0.94 | **0.97** | 0.95 | **0.96** | 0.94 | 0.95 | 0.95 | 0.88 | **0.92** | 0.95 | 0.93 |
| ET | 350 | 0.93 | 0.96 | 0.94 | **0.96** | 0.92 | 0.94 | 0.94 | 0.89 | 0.91 | 0.94 | 0.92 |
| GBT | 350 | **0.96** | 0.96 | **0.96** | **0.96** | **0.96** | **0.96** | **0.96** | 0.89 | **0.92** | **0.96** | **0.94** |

above, a Support Vector Machine (SVM), Random Forest (RF), Extra Trees (ET), and Gradient Boosted Trees (GBT) are trained using five-fold cross-validation and a grid search for hyperparameter tuning. The same features are used as in the regression task, refer to Equation (4.2.9). The classification results are shown in Table 4.5. Besides the notable worse performance of the SVM in the precision and $F1$ metric in the lane change right class, all results are comparable. There are slight advantages for Gradient Boosted Trees that were also the case in the desired velocity regression task. Hence, it was used for further analysis within the Monte Carlo situation prediction approach on the highD data. The confusion matrix in Figure 4.5 shows a strong main diagonal. Lane changes left and right are never confused by the algorithm. However, a certain amount of lane changes are assigned to the lane-keeping class and vice versa.

Table 4.6 shows the used configurations of the Monte Carlo situation prediction approach. The letters hD distinguish the configurations from the ones used in the simulation. Note that the desired velocity regression using Gradient Boosted Trees (GBT Reg.) as described in Section 4.2 is also evaluated against the baseline of simply using a probability

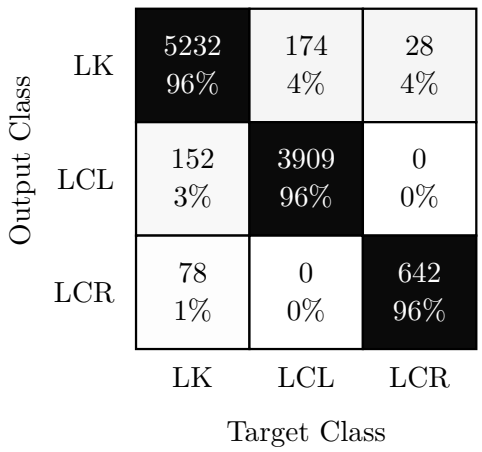



Figure 4.5.: Confusion matrix of the Gradient Boosted Trees classifier on the test data.

Table 4.6.: Configurations of the Monte Carlo prediction approach used for the evaluation on the highD dataset. Abbreviations: LCP: Lane Change Prediction, GBT: Gradient Boosted Trees

| | Situation Prediction | | | |
|---|---|---|---|---|
| Configuration | C1-hD | C2-hD | C3-hD | C4-hD |
| Monte Carlo simulations | 50 | 50 | 50 | 50 |
| LCP Treshold | None | 0.65 | None | 0.65 |
| $\hat{v}_{\text{des,i}}$ | $v_{0,i}$ | $v_{0,i}$ | GBT Reg. | GBT Reg. |

distribution with the mean set to the initial velocity $v_0$.
Some results for state-of-the-art Deep Learning-based trajectory prediction approaches are taken from [Krü+20]. However, those approaches likely use a different set of data compared to the one used here. Still, the results will most likely generalize, and a comparison is therefore possible. All results are shown in Table 4.7. Note further that [Krü+20] does not provide results for the probability metric, and the prediction horizon ends at 5 s.
First of all, the results clearly indicate that, while the desired velocity regression provides improved results on the regression task itself compared to the baseline $\hat{v}_{\text{des},i} = v_{0,i}$, it leads to worse results when used in the Monte Carlo situation prediction. An evaluation of the RMSE and MAE for the regression on this particular dataset also showed a clear advantage of the regression approach compared to using the initial velocity:

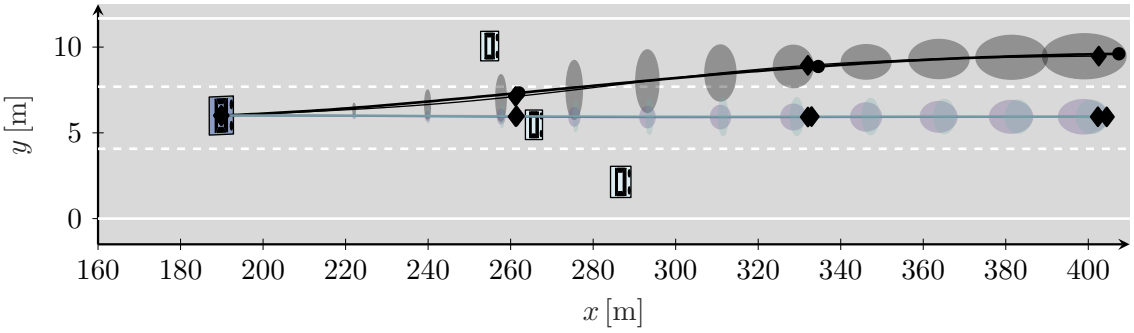
$$\begin{aligned} \text{RMSE}_{v_0} &= 4.44\,\text{m/s}, \quad \text{MAE}_{v_0} = 2.95\,\text{m/s}, \\ \text{RMSE}_{\hat{v}_{\text{des}}} &= 3.20\,\text{m/s}, \quad \text{MAE}_{\hat{v}_{\text{des}}} = 2.56\,\text{m/s}. \end{aligned} \tag{4.3.3}$$

The hypothetical reason for worse situation prediction results is that the vehicles' predicted movements are more strongly influenced by the scene configuration than the desired velocity. Even a comparably high desired velocity will not result in speeding up if a vehicle follows a slower vehicle. The conclusion is that such simple desired velocity regression does not lead to better results. Future work should analyze a different way of setting the desired velocity regression targets by optimizing them using the Intelligent Driver Model. Using an optimization mechanism to choose the desired velocity minimizes the error of a trajectory generated using the intelligent driver model compared to the driven trajectory. This is left for future work. The results here indicate that the desired velocity is an important parameter and must be chosen very carefully. It influences the prediction performance strongly.
Another important observation from Table 4.7 is that state-of-the-art Deep Learning-based approaches outperform the Monte Carlo situation prediction by a comparably big margin. The most likely reason for this is that the learning objectives better reflect the prediction task. Those approaches directly optimize the negative log-likelihood NLL and hence learn to predict the right uncertainty that reflects the data best, refer for example to [DT18]. On the other hand, the Monte Carlo approach relies on user-defined sampling distributions that are not optimized and reflect a prior belief. In turn, this leads to inferior performance.
Yet, the approach has other advantages that make it a valid choice for the application in current generation automated driving software systems. Those are its interpretability and scalability. In addition, it can be easily tailored to the target hardware platform by

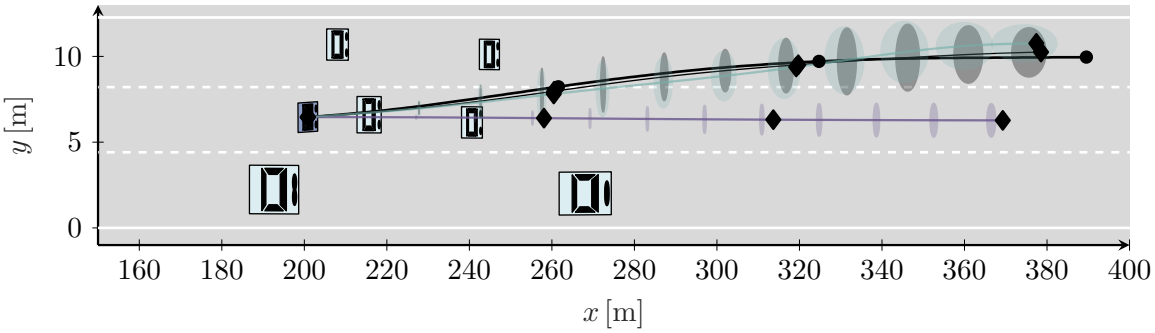
reducing the number of Monte Carlo simulations. But most importantly, it allows counterfactual reasoning as used in Section 6 to enable interaction-awareness by conditioning the situation prediction on the ego-vehicle trajectory. The Monte Carlo approach allows this easily, whereas such functionality is currently the focus of many research works for Deep Learning-based approaches.
Some results on the highD dataset using the situation prediction are shown in Figure 4.6 focusing only on the prediction of the dark-blue vehicle. The uncertainties are shown as ellipses. The black ellipses correspond to the most likely cluster, the purple to the second most likely, and turquoise to the least likely one. Driven trajectories are shown as solid black lines. A black circle indicates the positions in 2 s time spacing. Diamonds are used in the case of predictions. Both examples 4.6a and 4.6b represent good examples where the most likely cluster corresponds to the driven trajectory. In contrast 4.6c and 4.6d show examples where this is not the case. However, the driven trajectories are correctly predicted as the second and third most likely, respectively. This is a strength of predicting several situation evolutions jointly. Notice that in some cases, the clusters nearly overlap. That is typical when working with situation predictions and reflects that any predicted vehicles can lead to a different final traffic configuration and, hence, create another cluster. Refer to Figure A.5 in appendix A.4 for a threedimensional illustration of all clusters of example 4.6a.

Table 4.7.: Results of the evaluation of the situation prediction on data from the highD dataset compared to several Deep Learning-based state of the art methods. The number of samples used for the situation prediction approaches are shown in the last column. The mean and standard deviation is provided for all metrics of the situation prediction evaluation. Abbreviations: CS-LSTM: **C**onvolutional **S**ocial Pooling **Long** **S**hort-**T**erm **M**emory, CNN: **C**onvolutional **N**eural **N**etwork

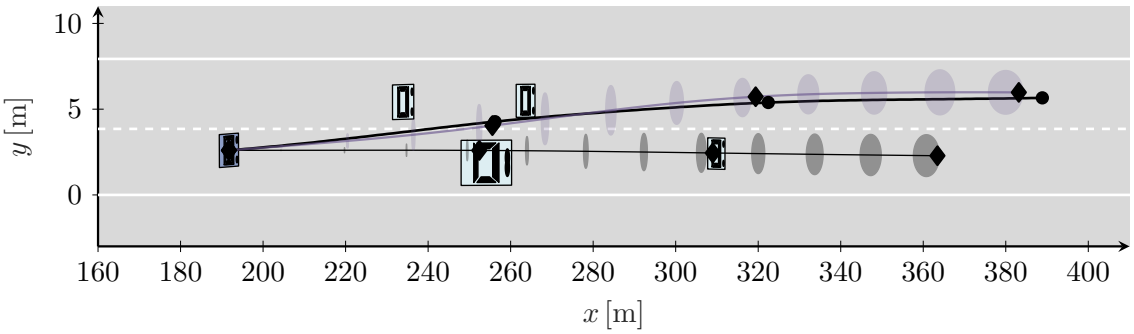
| Metric | Time [s] | LSTM | CS-LSTM | 2D-CNN | 3D-CNN | C1-hD | C2-hD | C3-hD | C4-hD | Samples |
|---|---|---|---|---|---|---|---|---|---|---|
| RMSE [m] | 1 | 0.57 | 0.33 | 0.34 | **0.26** | 0.31±0.35 | 0.31±0.35 | 0.34±0.32 | 0.35±0.33 | 521 |
| | 2 | 1.17 | 0.81 | 0.79 | **0.65** | 0.97±1.05 | 0.98±1.05 | 1.08±1.00 | 1.10±1.01 | 521 |
| | 3 | 1.80 | 1.44 | 1.38 | **1.20** | 1.91±1.92 | 1.93±1.94 | 2.13±1.87 | 2.17±1.90 | 521 |
| | 4 | 2.51 | 2.22 | 2.12 | **1.91** | 3.10±2.91 | 3.12±2.95 | 3.46±2.85 | 3.51±1.90 | 521 |
| | 5 | 3.49 | 3.13 | 3.03 | **2.74** | 4.50±3.97 | 4.52±4.03 | 5.02±3.89 | 5.09±3.96 | 521 |
| | 7.5 | - | - | - | - | 8.66±6.87 | 8.71±7.00 | 9.81±6.78 | 9.92±6.91 | 462 |
| | 10 | - | - | - | - | 12.64±8.37 | 12.61±8.41 | 15.35±9.03 | 15.44±9.17 | 335 |
| $MAE_x$ [m] | 1 | 0.56 | 0.32 | 0.30 | **0.23** | **0.23±0.32** | **0.23±0.32** | 0.26±0.30 | 0.27±0.30 | 521 |
| | 2 | 1.15 | 0.77 | 0.71 | **0.59** | 0.81±0.98 | 0.82±1.00 | 0.92±0.97 | 0.94±0.96 | 521 |
| | 3 | 1.75 | 1.38 | 1.27 | **1.12** | 1.69±1.82 | 1.70±2.82 | 1.91±1.79 | 1.94±1.82 | 521 |
| | 4 | 2.44 | 2.14 | 1.99 | **1.81** | 2.82±2.78 | 2.84±2.82 | 3.19±2.74 | 3.24±2.78 | 521 |
| | 5 | 3.41 | 3.04 | 2.91 | **2.63** | 4.18±3.81 | 4.20±3.88 | 4.70±3.76 | 4.77±3.83 | 521 |
| | 7.5 | - | - | - | - | 8.28±6.73 | 8.32±6.87 | 9.42±6.69 | 9.54±6.82 | 462 |
| | 10 | - | - | - | - | 12.30±8.35 | 12.27±8.38 | 15.00±9.03 | 15.11±9.17 | 335 |
| $MAE_y$ [m] | 1 | **0.05** | **0.05** | 0.11 | 0.07 | 0.08±0.10 | 0.08±0.10 | 0.08±0.10 | 0.08±0.10 | 521 |
| | 2 | **0.13** | 0.14 | 0.21 | 0.16 | 0.16±0.22 | 0.16±0.22 | 0.16±0.22 | 0.16±0.22 | 521 |
| | 3 | **0.22** | **0.22** | 0.31 | 0.25 | **0.22±0.33** | **0.22±0.33** | **0.22±0.33** | **0.22±0.33** | 521 |
| | 4 | 0.30 | 0.30 | 0.39 | 0.32 | **0.27±0.42** | **0.27±0.42** | **0.27±0.42** | 0.27±0.44 | 521 |
| | 5 | 0.37 | 0.37 | 0.45 | 0.39 | **0.32±0.51** | **0.32±0.52** | **0.32±0.51** | **0.32±0.53** | 521 |
| | 7.5 | - | - | - | - | 0.39±0.62 | **0.38±0.62** | 0.39±0.61 | 0.39±0.62 | 462 |
| | 10 | - | - | - | - | **0.34±0.56** | **0.34±0.56** | 0.35±0.56 | **0.34±0.54** | 335 |
| $\overline{\lvert P(\tau\lvert\hat{\tau})\rvert}$ [ ] | 1 | - | - | - | - | **2.94±1.57** | **2.94±1.59** | 2.69±1.57 | 2.67±1.59 | 521 |
| | 2 | - | - | - | - | **0.76±0.79** | 0.74±0.79 | 0.63±0.86 | 0.65±0.89 | 521 |
| | 3 | - | - | - | - | **0.26±0.32** | 0.25±0.30 | 0.20±0.36 | 0.21±0.35 | 521 |
| | 4 | - | - | - | - | **0.13±0.17** | 0.12±0.14 | 0.10±0.20 | 0.10±0.20 | 521 |
| | 5 | - | - | - | - | **0.07±0.10** | **0.07±0.10** | 0.06±0.10 | 0.06±0.10 | 521 |
| | 7.5 | - | - | - | - | **0.03±0.02** | **0.03±0.02** | 0.02±0.02 | 0.02±0.02 | 462 |
| | 10 | - | - | - | - | 0.00±0.01 | 0.00±0.01 | 0.00±0.01 | 0.00±0.01 | 335 |



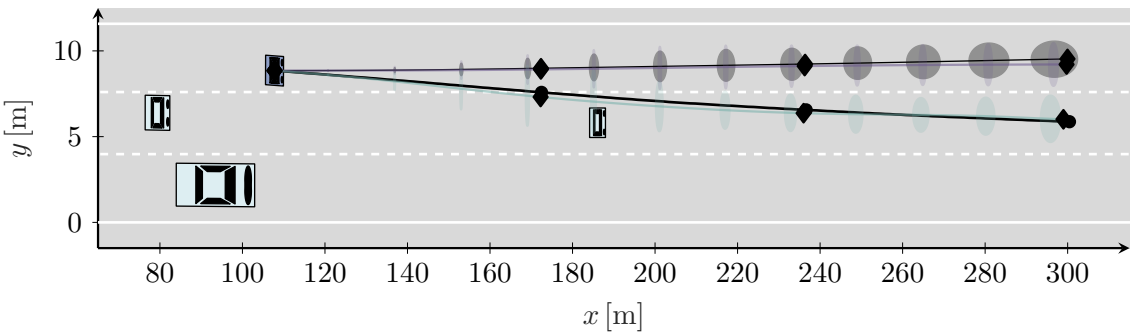

(a) Exemplary prediction result 1.




(b) Exemplary prediction result 2.




(c) Exemplary prediction result 3.

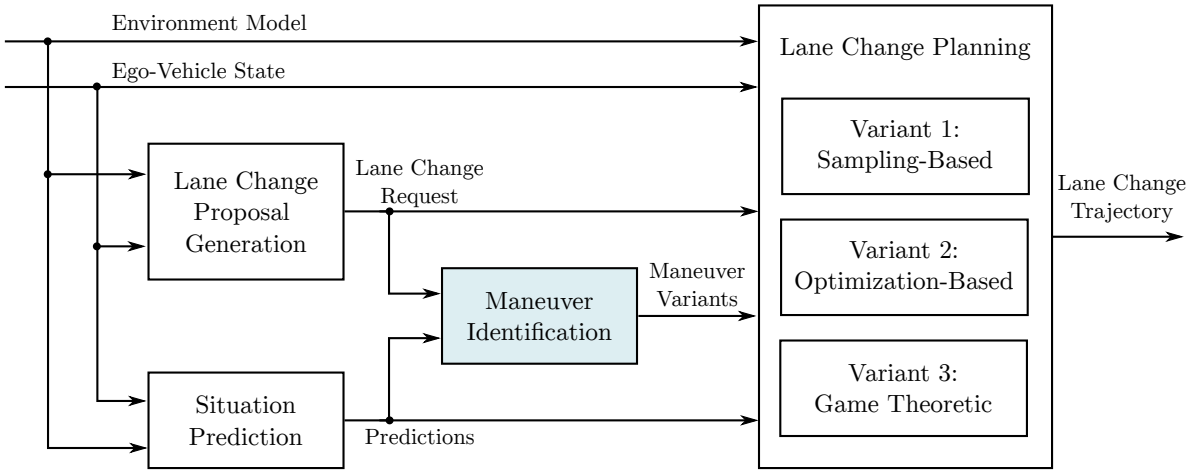



(d) Exemplary prediction result 4.

Figure 4.6.: Four examplary traffic scenes of the highD dataset for the illustration of the Monte Carlo situation prediction results. Driven trajectories are shown as solid black lines. A black circle indicates the positions in 2 s time spacing. Diamonds are used in the case of predictions with the same temporal spacing.

# 5

# Spatiotemporal Lane Change Maneuver Identification




This Chapter describes the spatiotemporal maneuver identification, refer to the above figure for its location within the architecture of the thesis at hand. It uses Polygon Clipping to derive spatiotemporal free space for adjacent lanes. A total of three variants are described in this Chapter. The Chapter is largely based on the publications [Sch+19a] and [Sch+19b].

## 5.1. Fundamental Maneuver Identification Concept

This Section introduces the spatiotemporal maneuver identification. It consists of the transformation of the traffic scene to a curvilinear coordinate frame, refer to Section 2.2 of Chapter 2, and a scene analysis based on Polygon Clipping

The motivation to use curvilinear coordinates for automated driving is to exploit road structure. In this new frame, tube-like traffic scenes always appear as straight scenes to achieve unified handling of many different scenarios. The coordinate frame origin, subsequently denoted as $(L, N)$, is attached at $X = 0\,\mathrm{m}$ on the chosen reference lane marking.

The Lanelet pseudo distance from [Ben+14] is used for the transformation of lane markings and predicted trajectories of surrounding traffic participants since transformed paths remain continuous, and it fulfills runtime requirements.

### Lane-discrete Spatiotemporal Free Space Description

One central contribution of the approaches presented here is the Polygon Clipping in the spatiotemporal domain to obtain lane change variants efficiently. An illustration of the interSection of two polygons is shown in Figure 5.1a. The operations interSection ($\cap$), union ($\cup$) and difference ($\setminus$) are applied using an adaption of Vattis Polygon Clipping algorithm, refer to [Vat92], [Mur04] and [Höl06].
Consider the right part of Figure 5.1. The light-blue area is the ego lane space ahead and a rectangle in $(L, t)$ domain, up to some limits $L_{\max}$ and $t_{\max}$. This is represented by the polygon:

$$\mathcal{P}_{\mathrm{SL}} = \{\left(\begin{bmatrix} L_{\mathrm{SL},j} & t_{\mathrm{SL},j} \end{bmatrix}^{\intercal}\right)_{j=0}^{n_{\mathrm{SL}}}\}, \tag{5.1.1}$$

where the subscript SL is short for start lane. A polygon is a closed polygonal chain of a set of points $\begin{bmatrix} L_j & t_j \end{bmatrix}^{\intercal}$ that are traversed in mathematical positive direction. On the current ego-vehicle lane drives one traffic participant in front of it. If, for example, this vehicle drives with constant velocity $\dot{L}$, it occupies the dashed area in the $(L, t)$ spatiotemporal domain on the ego lane in the lower right part of Figure 5.1. This is mathematically represented by the polygon:

$$\mathcal{P}_{\mathrm{OTP}} = \{\left(\begin{bmatrix} L_{\mathrm{OTP,j}} & t_{\mathrm{OTP,j}} \end{bmatrix}^{\intercal}\right)_{j=0}^{n_{\mathrm{OTP}}}\}. \tag{5.1.2}$$

The subscript OTP is short for all **O**ther **T**raffic **P**articipants of a scene, except the ego-vehicle. Note that the slope in Figure 5.1 (right) corresponds to the vehicle's velocity. Safety margins can be directly accounted for by extending the vehicle's occupied area.
To get the free space on the ego lane, the difference operator ($\setminus$) is applied between the ego lane free space polygon and the traffic participant occupancy polygon:

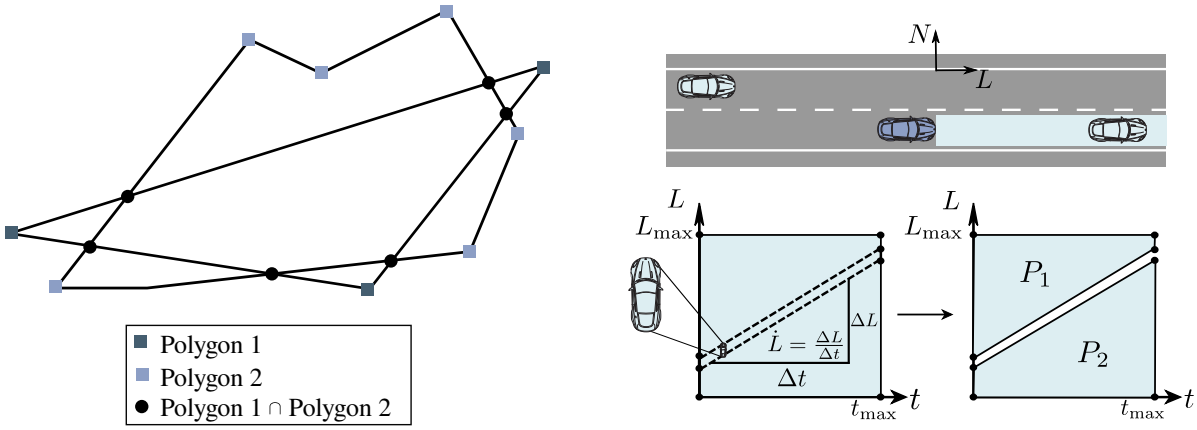
$$\{\mathcal{P}_i\} = \mathcal{P}_{\mathrm{SL}} \setminus \mathcal{P}_{\mathrm{OTP}}. \tag{5.1.3}$$

In the example, this results in two ($i \in \{1, 2\}$) distinct polygons $\mathcal{P}_1$ and $\mathcal{P}_2$ shown in the lower right part of the Figure 5.1. The meaning of the upper area is everything that is in front of the traffic participant in $L$ and $t$. Similarly, the lower part is the spatiotemporal free space behind the traffic participant, and the ego-vehicle resides at $\begin{bmatrix} L = 0\,\mathrm{m} & t = 0\,\mathrm{s} \end{bmatrix}^{\intercal}$ within this second area at the start of a planning cycle.

### Spatiotemporal Scene Analysis using Polygon Clipping

Consider a straight traffic situation similar to the one of Figure 5.1b and that the ego-vehicle wants to change to the left lane. Applying the previously described method based on Polygon Clipping on the start lane and the target lane gives the enumerated lane free space areas shown in the left and right $L$-$t$ diagrams in Figure 5.2. Every number corresponds to a polygon in Figure 5.2, for example:

$$① \mathrel{\hat{=}} \mathcal{P}_1 = \{\left(\begin{bmatrix} L_{1,j} & t_{1,j} \end{bmatrix}^{\intercal}\right)_{j=0}^{n_1}\}. \tag{5.1.4}$$




(a) Lane free space derivation using Polygon Clipping.

(b) Lane free space derivation using Polygon Clipping.

Figure 5.1.: Lane free space derivation using Polygon Clipping.

To reflect the fact, that there are three different types of areas, three index sets are introduced. In the example case:

$$\tilde{\mathcal{SL}} = \{1, 2\}, \quad \tilde{\mathcal{LC}} = \{3, 4, 5\}, \quad \tilde{\mathcal{TL}} = \{6, 7\}. \tag{5.1.5}$$

The set $\tilde{\mathcal{SL}}$ indexes spatiotemporal free space areas on the start lane of the ego-vehicle, $\tilde{\mathcal{LC}}$ represents the lane change areas and $\tilde{\mathcal{TL}}$ is the index set of the areas on the target lane.

In order to identify maneuver variants, Polygon Clipping is utilized, this time applying the interSection ($\cap$) operator. The interSection of all areas of the start lane free space (① and ② in Figure 5.2) with the free space areas in the target lane (⑥ and ⑦) are calculated. So for example $\mathcal{P}_2 \cap \mathcal{P}_6 = \mathcal{P}_3$. This results in three distinct areas that are subsequently called lane change areas, shown in in the middle of Figure 5.2, and indexed using the set $\mathcal{LC}$. Finally, the directed acyclic graph in Figure 5.3 is created that represents the connectivities of all obtained free space areas and therefore reflects the scene's topology. Since the ego-vehicle starts in area ②, only two paths are actually traversable to get to the areas ③ and ⑤ on the target lane. Therefore the index sets can be further pruned:

$$\mathcal{SL} = \{2\}, \quad \mathcal{LC} = \{3, 5\}, \quad \mathcal{TL} = \tilde{\mathcal{TL}}. \tag{5.1.6}$$

The lane change areas denote spatiotemporal regions in $(L, t)$ domain in which both current and target lane are free, so a lane change is possible. To summarize, the ego-vehicle starts in area ②. From there, it has the option to do an immediate lane change in area ⑤ and to end up in front of the other traffic participant on the target lane in area ⑦. The other option is to first stay in area ② on its lane, wait until the other traffic participant on the target lane passes the ego-vehicle, conduct the lane change in area ③ and finally end up in area ⑥. That means that the ego-vehicle is behind the traffic participant on the target lane at the end of the maneuver.

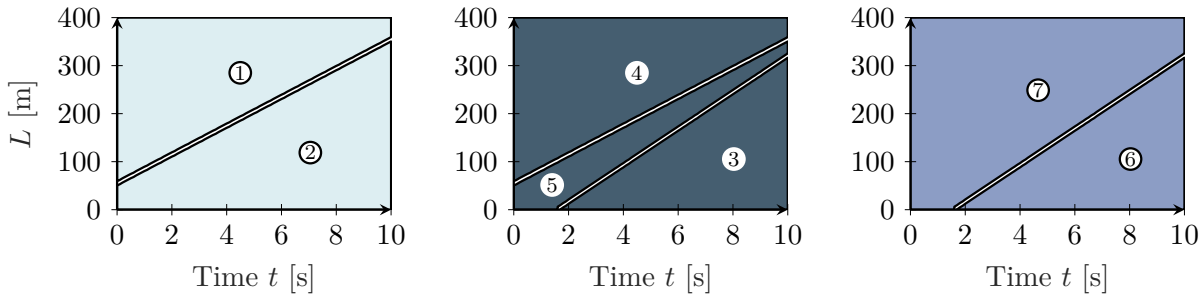



Figure 5.2.: Spatiotemporal lane's free space areas for the described example. Each area ⓘ is represented using a polygon $\mathcal{P}_i$. Left: Free space on current ego-vehicle lane. Middle: Lane change areas, that correspond to free space simultaneously on current and target lane. Right: Free space on left target lane.

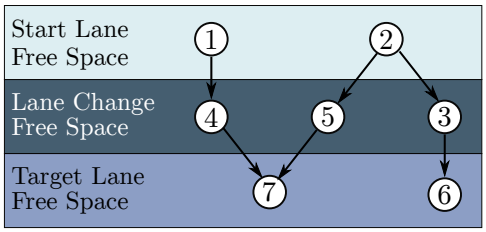



Figure 5.3.: Resulting graph encoding the ego-vehicles maneuver options.

## 5.2. Consideration of Lane Changes of Surrounding Traffic

This Section describes the functionality to handle lane changes of surrounding vehicles using Polygon Clipping in spatiotemporal $(L, t)$ domain and construct a maneuver options graph that can encode the topology of such complex traffic scenes.

### Lane-Discrete Spatiotemporal Free Space Description with Consideration of surrounding Vehicle Lane Changes

Consider the synthetic traffic scene in Figure 5.4 with the dark-blue ego-vehicle E and the two light-blue surrounding vehicles $\mathrm{O}_1$ and $\mathrm{O}_2$. The dashed, black lines depict the lane change maneuvers of both surrounding vehicles. The trajectory prediction module described in Chapter 4 is used. The obtained information also includes a lane association of each vehicle to driving lanes perceived by the ego-vehicle's sensor system. A driving lane occupancy of an surrounding vehicle $\mathrm{O}_i$ is represented using a spatiotemporal polygon consisting of $n_{\mathrm{O}_i}$ points:

$$\mathcal{P}_{\mathrm{O}_i} = \left\{ \left( \left[ \mathrm{O}_i, k, t_{\mathrm{O}_i,k} \right]^{\mathsf{T}} \right)_{k=0}^{n_{\mathrm{O}_i}} \right\}. \tag{5.2.1}$$

Let $t_{\mathrm{O,LC}}$ be the time instant when an vehicle is assigned for the first time to the target lane of its lane change. A fixed transition period $t_{\mathrm{O,LCT}} = 1.3\,\mathrm{s}$ is defined empirically in order to mark the start lane (subscript $(\cdot)_{\mathrm{SL}}$) of the lane change as occupied for the time interval $[0\,\mathrm{s}, t_{\mathrm{O,LC}} + t_{\mathrm{O,LCT}}]$ and the target lane (subscript $(\cdot)_{\mathrm{TL}}$) for $[t_{\mathrm{O,LC}} - t_{\mathrm{O,LCT}}, t_{\max}]$ with the maximum planning time horizon $t_{\max} = 10\,\mathrm{s}$. The algorithm uses axis-parallel

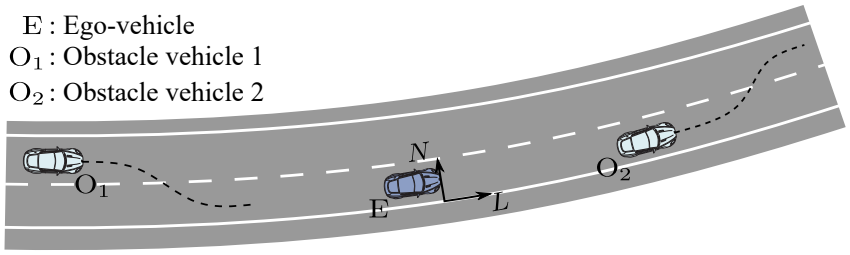



Figure 5.4.: Exemplary traffic scene with two lane changing surrounding vehicles.

rectangles ranging over the whole longitudinal planning horizon and starting at the times $t_\mathrm{a}$ and ending at $t_\mathrm{b}$ abbreviated as :

$$\mathcal{P}_\mathrm{rect}(t_\mathrm{a}, t_\mathrm{b}) \coloneqq \{\begin{bmatrix} L_\mathrm{min} & t_\mathrm{a} \end{bmatrix}^\intercal, \begin{bmatrix} L_\mathrm{min} & t_\mathrm{b} \end{bmatrix}^\intercal, \begin{bmatrix} L_\mathrm{max} & t_\mathrm{b} \end{bmatrix}^\intercal, \begin{bmatrix} L_\mathrm{max} & t_\mathrm{a} \end{bmatrix}^\intercal\}. \tag{5.2.2}$$

Using the difference operation ($\setminus$) on the start lane of the ego-vehicle and considering all obstacle occupancies, one obtains the spatiotemporal free space as a set of distinct polygons $\mathcal{P}_i$:

$$\{\mathcal{P}_i\} = \mathcal{P}_\mathrm{SL} \setminus \{\mathcal{P}_{\mathrm{O}_i}\}, \tag{5.2.3}$$

with the rectangular polygon $\mathcal{P}_\mathrm{SL} = \mathcal{P}_\mathrm{rect}(0\,\mathrm{s}, t_\mathrm{max})$ that represents the spatiotemporal free space on the start lane for the case of complete absence of surrounding vehicles on that lane. Herein $L_\mathrm{min}$ and $L_\mathrm{max}$ denote the minimum and maximum longitudinal planning horizon. The same procedure is applied to obtain the set of spatiotemporal free space areas on the target lane. Let $t_{\mathrm{C},k}$ denote time instants when a lane occupancy begins or ends. There are two such time instants in our example on the starting lane of the ego-vehicle, refer to the top-left graph in Figure 5.5, $t_{\mathrm{C},1} = 2.1\,\mathrm{s}$ and $t_{\mathrm{C},2} = 4\,\mathrm{s}$. The following three polygons are formed in the example case: $\mathcal{P}_{\mathrm{SL},1} = \mathcal{P}_\mathrm{rect}(0\,\mathrm{s}, t_{\mathrm{C},1})$, $\mathcal{P}_{\mathrm{SL},2} = \mathcal{P}_\mathrm{rect}(t_{\mathrm{C},1}, t_{\mathrm{C},2})$ and $\mathcal{P}_{\mathrm{SL},3} = \mathcal{P}_\mathrm{rect}(t_{\mathrm{C},2}, t_\mathrm{max})$.
Using Polygon Clipping with the interSection operation ($\cap$) of both sets of polygons $\{\mathcal{P}_i\}$ and $\{\mathcal{P}_{\mathrm{SL,j}}\}$ one obtains in the exemplary traffic scene a total of seven spatiotemporal free space areas, refer to the top-left graph in Figure 5.5.
The notation $\odot$ represents the respective polygon $\mathcal{P}_{(\cdot)}$. In the example case the areas ④ and ⑥ as well as ⑤ and ⑦ can be combined using a union operation ($\cup$) on the respective polygons. This reduction of the number of distinct areas reduces complexity. The criterion for joining two polygons is a joint edge at a temporal division line since no information, but only complexity is added in such cases. Next, the adjacency of all nodes of the start lane is checked for connectivity with the start node ① of the ego-vehicle. Unreachable nodes, as for example the resulting fused node of ④ and ⑥, are removed. Finally, only four distinct spatiotemporal areas remain on the start lane of the ego-vehicle, as shown in the top-right graph in Figure 5.5. Using the same procedure on the target lane, five areas arise, as shown in the bottom-right graph of Figure 5.5.
The interSection operation of all areas of the start lane with the areas of the target lane and application of the aforementioned fusion mechanism results in three distinct lane change areas ⑤, ⑥ and ⑦ in the bottom-left graph in Figure 5.5. They describe spatiotemporal free space simultaneously on both the start and target lane and represent areas in which lateral movement of the ego-vehicle is permitted.

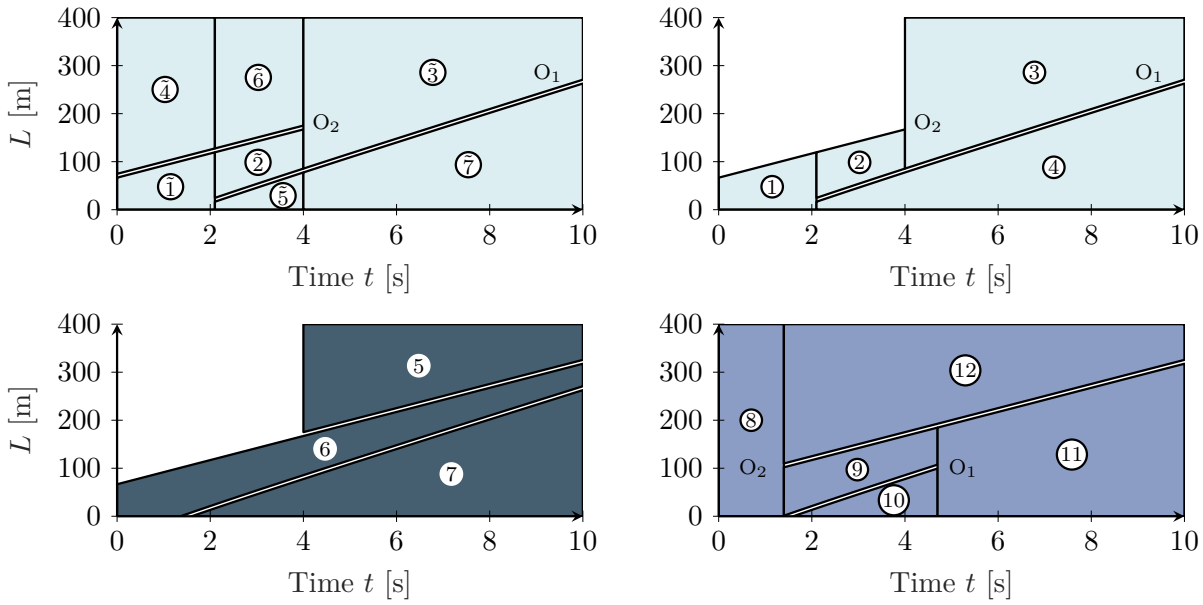



Figure 5.5.: Spatiotemporal free space areas for the traffic scene shown in Figure 5.4 with surrounding vehicle lane changes. From top-left to bottom-right, the plots depict: The detailed polygons of the start lane of the ego-vehicle; unified polygons of the start lane of the ego-vehicle; lane change area polygons; unified target lane area polygons.

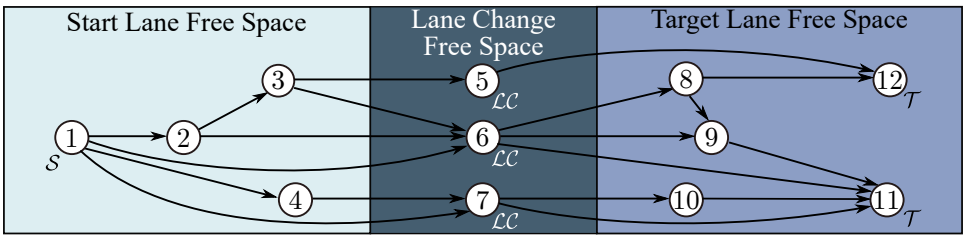
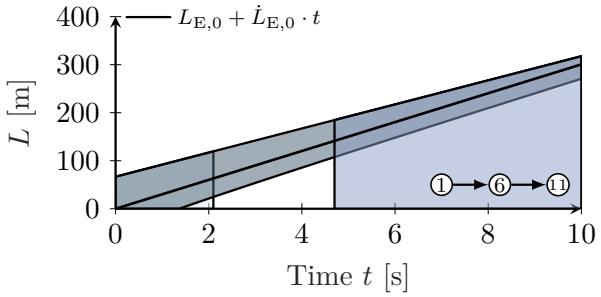


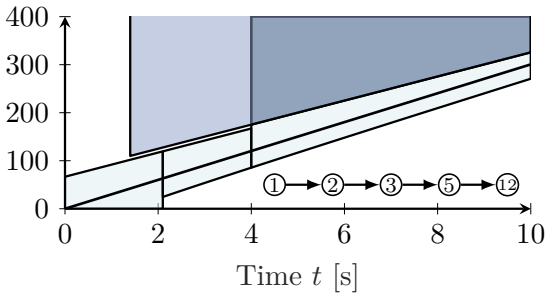

Figure 5.6.: Graph encoding the traffic scene topology corresponding to the spatiotemporal free space areas and adjacencies shown in Figure 5.5.

### Complete Traffic Scene Topology Graph

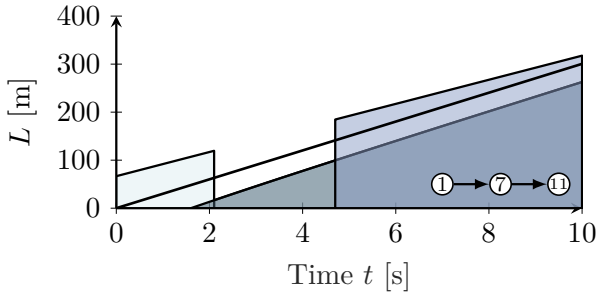
Figure 5.6 shows the resulting graph that encodes the traffic scene topology. Specific paths in this graph represent lane change execution variants of the ego-vehicle. The ego-vehicle starts at $\begin{bmatrix} L_{\mathrm{E},0} & t_{\mathrm{E},0} \end{bmatrix}^\intercal = \begin{bmatrix} 0\,\mathrm{m} & 0\,\mathrm{s} \end{bmatrix}^\intercal$ so that area ① represents the start node in the graph and is marked with the symbol $\mathcal{S}$. The nodes that correspond to lane change areas are marked with the symbol $\mathcal{LC}$. All target areas are marked with the symbol $\mathcal{T}$ (⑪ and ⑫). Both areas are characterized by the fact, that their right boundary includes the maximum planning time horizon $t_{\max} = 10\,\mathrm{s}$. Maneuver execution variants correspond to paths starting in node ①, passing through exactly one lane change node (⑤, ⑥ or ⑦) and ending in one of the final nodes, either ⑪ or ⑫. Three exemplary paths in the graph are depicted in Figure 5.7. Figure 5.7a corresponds to the maneuver ①→⑥→⑪ where the ego-vehicle conducts its lane change between both surrounding vehicles.
Figure 5.7b represents the maneuver ①→②→③→⑤→⑫. Here the ego-vehicle waits for the leading obstacle vehicle $\mathrm{O}_2$ to change to the left lane then overtakes it on the right side and finally changes lane in front of it. Analysis of the reachable set shows that this




(a) Immediate lane change manuever.




(b) Delayed lane change maneuver option 1.



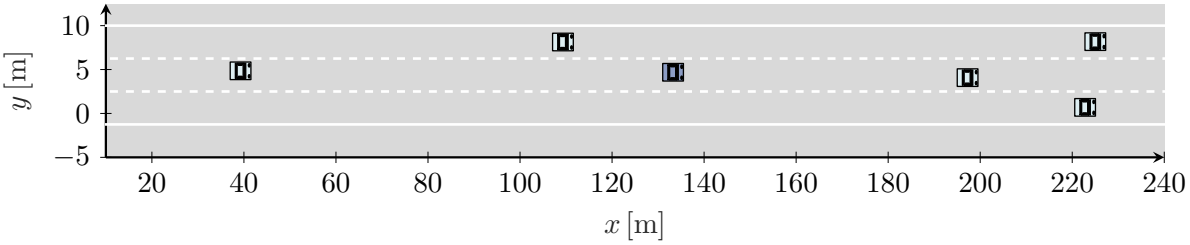

(c) Delayed lane change maneuver option 2.

Figure 5.7.: Three exemplary lane change maneuver paths that the graph in Figure 5.6 encodes.

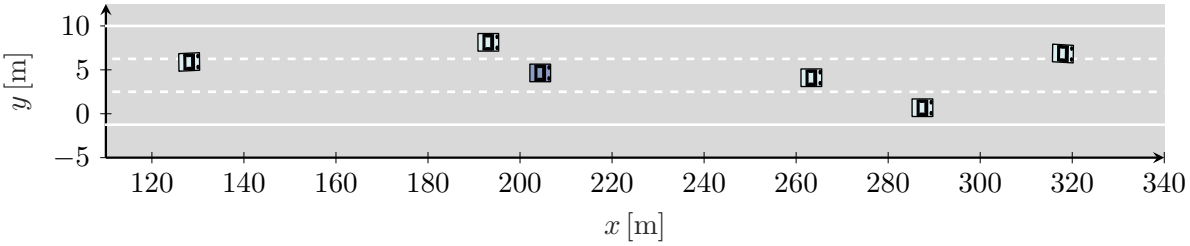
maneuver can be achieved within the maximum planning horizon when driving almost at the acceleration limits.
Finally, Figure 5.7c corresponds to the maneuver ①→⑦→⑪. Here, the ego-vehicle would break to fall behind the trailing vehicle $O_1$ and then execute the lane change to the left lane. Another important maneuver is represented by the path ①→⑥→⑧→⑫ that corresponds to overtaking the vehicle $O_2$ on the start lane of the ego-vehicle before the obstacle vehicle $O_2$ conducts its own lane change.

## 5.3. Complexity Reduction

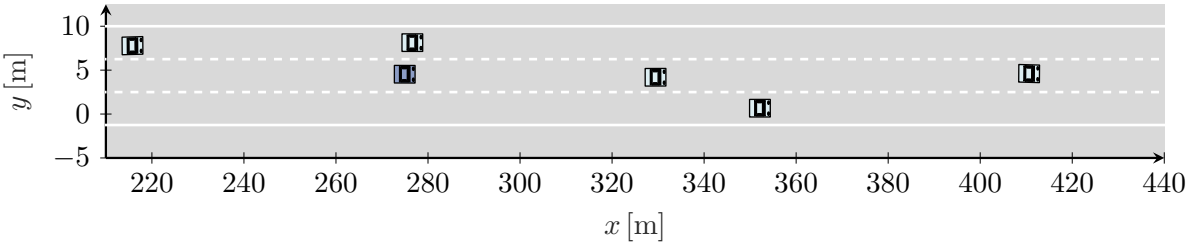
The complete traffic scene topology graph described in the previous Section 5.2 represents a traffic scene, its evolution, and all maneuver options in a compact way. However, its main disadvantage is that ego-vehicle lane change maneuvers are ambiguous, and the number of nodes in a path varies. Therefore, the description in Section 5.1 when disregarding lane changes of surrounding vehicles, is easier to work with in the subsequent lane change planning step. Therefore, this Section aims to reduce the complexity of the complete traffic scene topology graph such that a lane change path consists of exactly three nodes.
The initial traffic scene and its evolution are shown in Figure 5.8. Two of the vehicles change lanes during the scene evolution. Applying the algorithm of the previous Section leads to the spatiotemporal areas shown in Figure 5.9. Unreachable areas are already removed.
The complete traffic scene topology graph is shown in Figure 5.10. Paths in this graph again correspond to lane change maneuvers of the ego-vehicle. A full specification of a



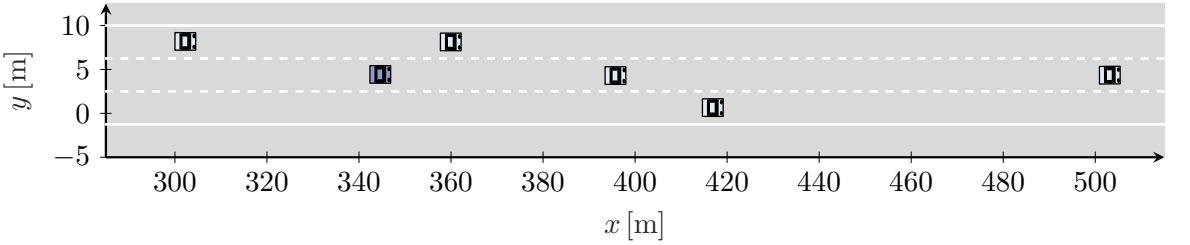

(a) Traffic situation at $t = 0\,\mathrm{s}$.



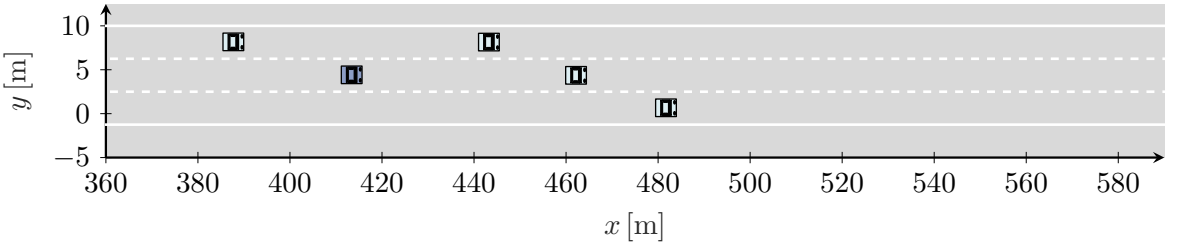

(b) Traffic situation at $t = 2.5\,\mathrm{s}$.




(c) Traffic situation at $t = 5\,\mathrm{s}$.

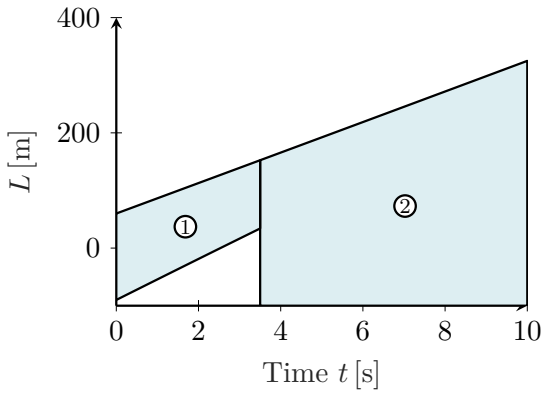



(d) Traffic situation at $t = 7.5\,\mathrm{s}$.

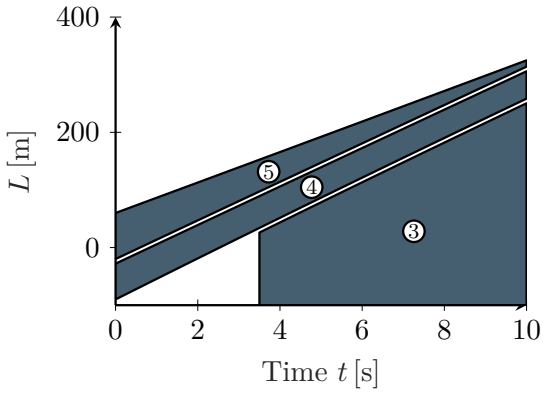



(e) Traffic situation at $t = 10\,\mathrm{s}$.

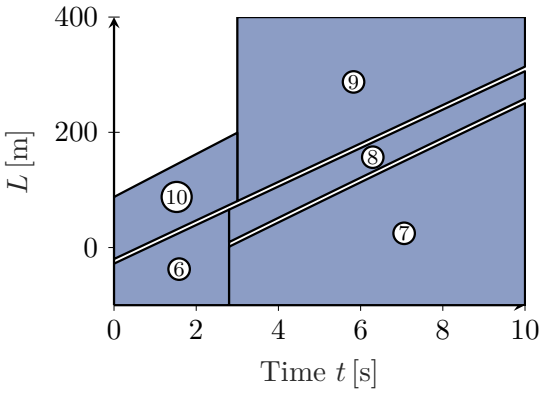
Figure 5.8.: Initital traffic scene and its evolution used as an example for the illustration of the graph complexity reduction mechanism.




(a) Start lane free space areas.




(b) Lane change free space areas.




(c) Target lane free space areas.

Figure 5.9.: Resulting free space areas for the example traffic scene prior to the complexity reduction. Each area ⓘ is represented using a polygon $\mathcal{P}_i$.

lane change requires setting the start and end times of the lane change to derive the longitudinal and lateral constraints for trajectory optimization.

The complexity reduction is explained using the example graph in Figure 5.10. In a first step, all start lane free space polygons are analyzed and grouped into two distinct sets $\mathcal{S}_0$ and $\mathcal{S}_F$. A polygon $\mathcal{P}_i \in \mathcal{S}_0$ if the initial time $t_0 = 0\,\mathrm{s}$ is part of the respective polygon. On the other hand, $\mathcal{P}_i \in \mathcal{S}_F$ if the final time $t_{\max} = 10\,\mathrm{s}$ is part of the polygon. A third set defined as $\mathcal{S}_K = \mathcal{S}_0 \cap \mathcal{S}_F$ is introduced. Polygons in this set extend from $t_0 = 0\,\mathrm{s}$ up to $t_{\max} = 10\,\mathrm{s}$ and remain unmodified in the complexity reduction. Similar sets are created holding the respective target lane free space polygons. Those are represented as $\mathcal{T}_0$, $\mathcal{T}_F$ and $\mathcal{T}_K$.

Next, two further sets are formed $\tilde{\mathcal{S}}_0 = \mathcal{S}_0 \setminus \mathcal{S}_K$ and $\tilde{\mathcal{T}}_F = \mathcal{T}_F \setminus \mathcal{T}_K$. All paths between nodes of the set $\tilde{\mathcal{S}}_0$ and $\mathcal{S}_F$ are derived. For each path, a new polygon is created by the union of all corresponding polygons. In the example in Figure 5.10, the nodes ① and ② are combined. Similarly, for the target lane areas, all paths between nodes of the set $\mathcal{T}_0$ and $\tilde{\mathcal{T}}_F$ are derived and combined accordingly. This results in three union operations shown in Figure 5.10.

This concludes the complexity reduction of the complete traffic scene graph. After it, the resulting graph is shown in Figure 5.11. Notably, all paths in this graph consist of

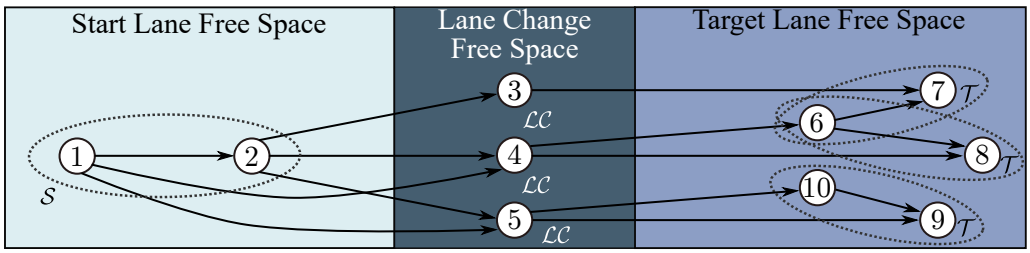



Figure 5.10.: Resulting graph encoding the ego-vehicles maneuver options.

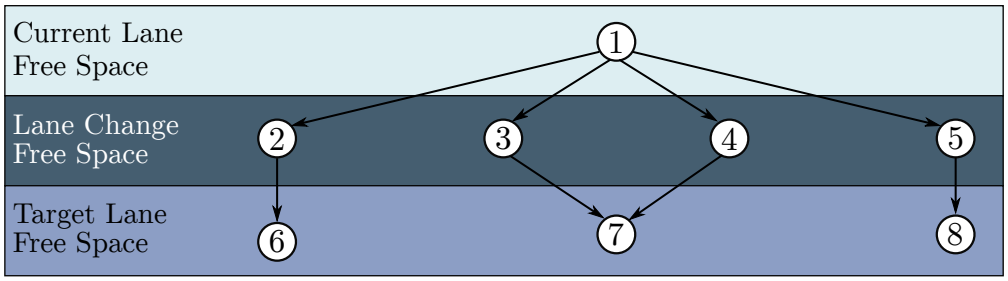
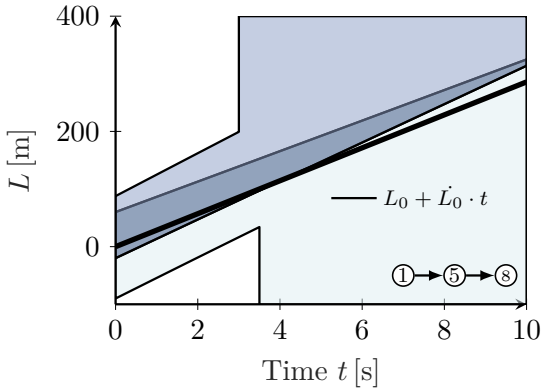
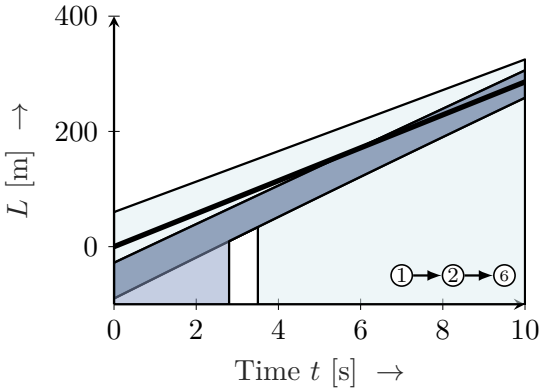



Figure 5.11.: Complexity reduced graph encoding the traffic scene spatiotemporally.

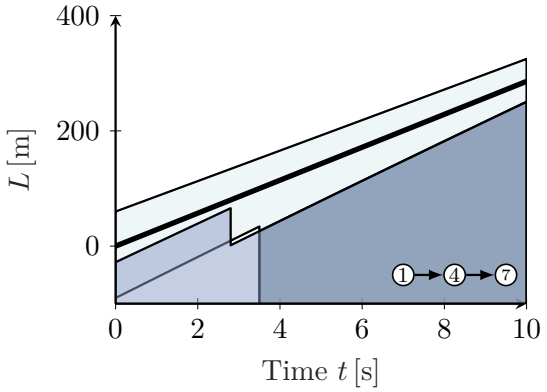
exactly three traversed nodes. All resulting polygons are shown in Figure A.7 in appendix A.6. The complexity reduction introduces redundancy, but it is easier to work with in the lane change behavior planning since no more graph search is necessary. Furthermore, it is straightforward to obtain the boundaries of each maneuver used in the trajectory optimization once the lane change start and end times are specified.
Four different maneuver options encoded in the graph are shown in Figure 5.12. Figure 5.12a represents the immediate lane change. The analysis shows that it might be possible that the ego-vehicle increases its velocity during the maneuver. Similarly, Figure 5.12b shows the delayed lane change maneuver that might be possible by deceleration. On the other hand, delayed lane changes in Figures 5.12c and 5.12d are not possible under the assumed constraints since the respective lane change areas are not reachable.
To summarize, this Chapter introduces algorithms to obtain lane change maneuver options of the ego-vehicle by reducing the problem to certain geometric operations realized with Polygon Clipping. Furthermore, it leads to efficient representations used for the generation of lane change trajectories in the remaining Chapters of the thesis at hand. Finally, apart from the uses within the thesis at hand, there is still an unused potential for the shape of the resulting lane change areas, which could be exploited to increase downstream modules' performance further.



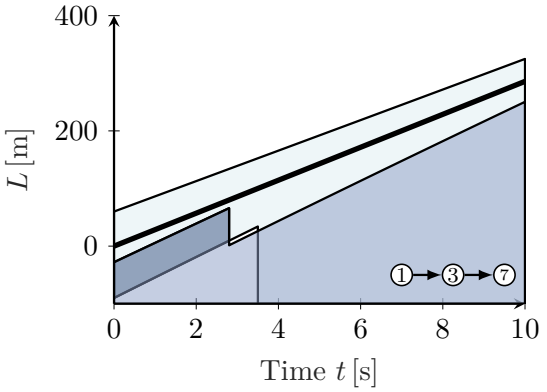

(a) Immediate lane change maneuver.




(b) Delayed lane change maneuver option 1.




(c) Delayed lane change maneuver option 2.

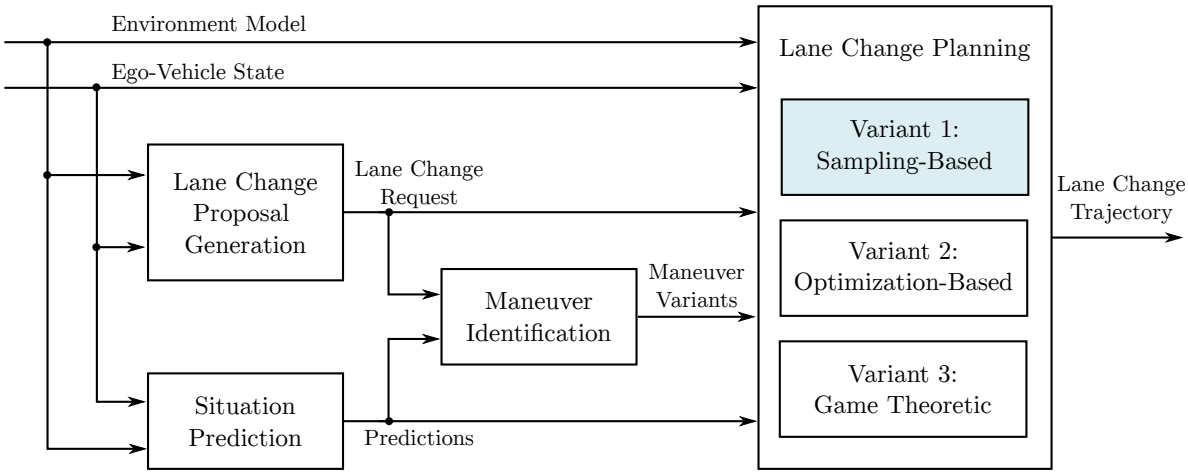



(d) Delayed lane change maneuver option 3.

Figure 5.12.: Examplary paths in the complexity reduced graph.

# 6

# Sampling-based Lane Change Planning




This Chapter discusses the first of a total of three approaches for lane change planning. It is largely based on the publication [Sch+19a]. After identification of maneuver options, the polygons are used to sample points for efficient calculation of trajectories. Next, several constraint checks are conducted for each trajectory to ensure safety and driveability. Finally, an interaction-aware trajectory ranking is done by estimating the influence of each maneuver option on the surrounding traffic.
The approach assumes that other surrounding traffic participants stick to their respective lanes and don't change lanes. This assumption will be relaxed by the remaining lane change planning approaches discussed in Chapters 7 and 8.

## 6.1. Geometry-Informed Sampling of Lane Change Trajectories

To make sampling of trajectories efficient, the geometric features of the identified lane change areas resulting from the maneuver identification (Chapter 5) are analyzed.

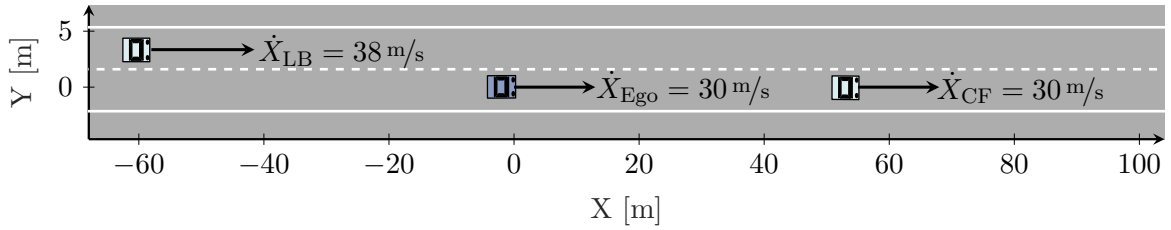



Figure 6.1.: Exemplary traffic scene in simulation used as the running example in this Chapter.

The traffic situation shown in Figure 6.1 is used to illustrate the algorithm. The ego-vehicle has the intention to do a lane change in order to drive with a higher velocity of $\dot{X}_{\mathrm{des}} = 33.33\,\frac{\mathrm{m}}{\mathrm{s}}$. Refer also to figures 5.2 and 5.3 of the previous Chapter for the corresponding polygons of the free space areas and maneuver options graph for this exact example.
The lane change areas are represented by the polygons $\mathcal{P}_3$ and $\mathcal{P}_5$ illustrated in Figure 5.2. The starting point is the calculation of the reachable set $\mathcal{L}$ in $(L,t)$ domain considering maximum deceleration $\ddot{L}_{\min}$ and acceleration $\ddot{L}_{\max}$ starting from the current initial longitudinal velocity $\dot{L}_0$ of the ego-vehicle:

$$L_{\mathcal{L},ub}(t) = \int_{t=0}^{t_{\max}} \min\left(\dot{L}_o + \ddot{L}_{\max}\, t, \dot{L}_{\max}\right) \mathrm{d}t, \tag{6.1.1}$$

$$L_{\mathcal{L},lb}(t) = \int_{t=0}^{t_{\max}} \max\left(\dot{L}_o + \ddot{L}_{\min}\, t, \dot{L}_{\min}\right) \mathrm{d}t, \tag{6.1.2}$$

with some situation dependent lower $\dot{L}_{\min}$ and upper bounds $\dot{L}_{\max}$ typically reflecting speed limits or other restrictions. The upper and lower bound $L_{\mathcal{L},ub}$ and $L_{\mathcal{L},lb}$ are represented with polygonal chains. Closing them results in the dashed polygon $\mathcal{P}_{\mathcal{L}}$ in the first row in Figure 6.2. Applying the interSection operator ($\cap$) using Polygon Clipping results in modified, often smaller lane change areas $\hat{P}_3 = \mathcal{P}_3 \cap \mathcal{P}_{\mathcal{L}}$ and $\hat{\mathcal{P}}_5 = \mathcal{P}_5 \cap \mathcal{P}_{\mathcal{L}}$. This way, sample-complexity is reduced. The reachable set can make use of prior knowledge. If it is known that the driver does not accept loosing a certain amount of velocity for a successful lane change, then this can be directly imposed as a lower boundary condition and guide the sampling accordingly.
Next is the calculation of the spatiotemporal area of the modified lane change areas $\hat{P}_k$:

$$A_{\hat{\mathcal{P}}_k} = \frac{1}{2}\left(\sum_{i=0}^{n_i-1}\left(L_{\hat{\mathcal{P}}_k,i+1}\, t_{\hat{\mathcal{P}}_k,i} - L_{\hat{\mathcal{P}}_k,i}\, t_{\hat{\mathcal{P}}_k,i+1}\right)\right), k \in \mathcal{LC} \tag{6.1.3}$$

and it is checked for exceeding some user-defined threshold $A_{\min}$. Figuratively speaking, this means that the space and time that the ego-vehicle has to conduct its lane change is probably sufficient. This feature is used as a pruning heuristic to reduce the computational cost by ruling out maneuver options beforehand. Next is the calculation of the equivalent height $L_{\mathrm{R},k}$ of a spatiotemporal rectangle with width $t_{\mathrm{LC},k}$ for the lane change areas of all maneuver options:

$$L_{\mathrm{R,k}} = \frac{A_{\hat{\mathcal{P}}_k}}{t_{\mathrm{LC},k}}, k \in \mathcal{LC}. \tag{6.1.4}$$

Herein, $t_{\mathrm{LC},k}$ is the temporal extent of area $\hat{\mathcal{P}}_k$. This feature is important when comparing two maneuver options because the geometric shape of the lane change areas is a key aspect. Assuming two lane change areas with the same area $A$, it is often better to choose the one with a longer duration allowing more time for maneuver completion.
The final step consists of the generation of sample point sets within the lane change areas:

$$\mathcal{S}_k = \{S_j\}_{j=1}^{n_\mathcal{S}},\ k \in \mathcal{LC} \tag{6.1.5}$$

with samples $S_j = \{\begin{bmatrix} L_{j,\mathcal{SL}} & t_{j,\mathcal{SL}} \end{bmatrix}^\intercal, \begin{bmatrix} L_{j,\mathcal{TL}} & t_{j,\mathcal{TL}} \end{bmatrix}^\intercal\}$ that the trajectories have to pass through. Herein $n_\mathcal{S}$ denotes the maximum number of samples. The times $t_{j,\mathcal{SL}}$ and $t_{j,\mathcal{TL}}$ correspond to the start and end of the ego-vehicle's lateral movement with the lane change duration $t_\mathrm{LC} = t_{j,\mathcal{TL}} - t_{j,\mathcal{SL}}$. Both $L_{j,\mathcal{SL}}$ and $L_{j,\mathcal{TL}}$ characterize the associated longitudinal coordinates of the start and end of the lateral movement. Then the centroids of the polygons $\hat{\mathcal{P}}_k,\ k \in \mathcal{LC}$ are calculated. The calculation of the centroids $\begin{bmatrix} C_{L,\hat{\mathcal{P}}_k} & C_{t,\hat{\mathcal{P}}_k} \end{bmatrix}^\intercal, k \in \mathcal{LC}$ is as follows:

$$A_{\hat{\mathcal{P}}_k} = \frac{1}{2}\left(\sum_{i=0}^{n_{\hat{\mathcal{P}}_k}-1} \left(L_{\hat{\mathcal{P}}_k,i+1} t_{\hat{\mathcal{P}}_k,i} - L_{\hat{\mathcal{P}}_k,i}\, t_{\hat{\mathcal{P}}_k,i+1}\right)\right), \tag{6.1.6}$$

$$C_{L,\hat{\mathcal{P}}_k} = \frac{1}{6A_{\hat{\mathcal{P}}_k}} \sum_{i=0}^{n_{\hat{\mathcal{P}}_k}-1} \left(L_{\hat{\mathcal{P}}_k,i} + L_{\hat{\mathcal{P}}_k,i+1}\right)\left(L_{\hat{\mathcal{P}}_k,i+1}\, t_{\hat{\mathcal{P}}_k,i} - L_{\hat{\mathcal{P}}_k,i}\, t_{\hat{\mathcal{P}}_k,i+1}\right), \tag{6.1.7}$$

$$C_{t,\hat{\mathcal{P}}_k} = \frac{1}{6A_{\hat{\mathcal{P}}_k}} \sum_{i=0}^{n_{\hat{\mathcal{P}}_k}-1} \left(t_{\hat{\mathcal{P}}_k,i} + t_{\hat{\mathcal{P}}_k,i+1}\right)\left(L_{\hat{\mathcal{P}}_k,i+1}\, t_{\hat{\mathcal{P}}_k,i} - L_{\hat{\mathcal{P}}_k,i}\, t_{\hat{\mathcal{P}}_k,i+1}\right). \tag{6.1.8}$$

The centroids are represented in the first row of Figure 6.2 using white dots. The time $C_{t,\hat{\mathcal{P}}_k}$ of the centroid divides a lane change area naturally into two intervals:

$$I_{\hat{\mathcal{P}}_k,1} = \left[t_{\hat{\mathcal{P}}_k,\min}, C_{t,\hat{\mathcal{P}}_k}\right), \quad I_{\hat{\mathcal{P}}_k,2} = \left[C_{t,\hat{\mathcal{P}}_k}, t_{\hat{\mathcal{P}}_k,\max}\right], \tag{6.1.9}$$

where $t_{\hat{\mathcal{P}}_k,\min}$ denotes the temporal beginning of the reachable lane change area and $t_{\hat{\mathcal{P}}_k,\max}$ the ending. In case of immediate lane change areas $t_{\hat{\mathcal{P}}_k,\min} = 0\,\mathrm{s}$. For example, for area ③ the values are roughly $t_{\hat{\mathcal{P}}_k,\min} = 4\,\mathrm{s}$ and $t_{\hat{\mathcal{P}}_k,\max} = 10\,\mathrm{s}$ which is also the upper bound of the planning horizon. Then the intervals $I_{\hat{\mathcal{P}}_k,1}$ and $I_{\hat{\mathcal{P}}_k,2}$ are subdivided by a user-defined number of equidistant steps along the time axis, while omitting both boundaries. Finally, for every discrete time in both sets, a user-defined number of points between the upper and lower boundary of the interSection of the lane change area and the reachable set are sampled. The first row of Figure 6.2 also represents both sample sets $\mathcal{S}_3$ and $\mathcal{S}_5$.
The above mechanism generates a certain number of geometry-informed samples. The absolute amount is upper-bounded and the boundary depends on user-defined choices of interval subdivisions and number of points per time slice $t_{j,\mathcal{SL}}$ and $t_{j,\mathcal{TL}}$. Then several sanity checks for reducing the number of samples are conducted. For this, the mean longitudinal velocity $\overline{\dot{L}}_{\hat{\mathcal{P}}_k,j}$ for every sample $j$ is estimated by using a linear connection of the two associated points $\begin{bmatrix} L_{j,\mathcal{SL}} & t_{j,\mathcal{SL}} \end{bmatrix}^\intercal$ and $\begin{bmatrix} L_{j,\mathcal{TL}} & t_{j,\mathcal{TL}} \end{bmatrix}^\intercal$, that are shown in red and beige in the first row of Figure 6.2. It is checked if this velocity is below the speed limit and also if the corresponding lane change has a duration of at least $t_{\mathrm{LC},\min} = 2.5\,\mathrm{s}$. A lane

change with a shorter duration is likely to be perceived uncomfortable by passengers and hence rarely seen in datasets such as in the highD data.
Samples with a negative or zero mean velocity $\overline{\dot{L}}_{\hat{\mathcal{P}}_k,j} \leq 0$ are removed, too. To ensure a deterministic upper boundary of the runtime for the trajectory calculation, it is necessary to have a fixed number of samples. It is intuitively clear, that an even distribution of samples within each maneuver option's lane change area with regards to the mean velocities $\overline{\dot{L}}_{\hat{\mathcal{P}}_k,j}$, end time of the lane change $t_{j,\mathcal{TL}}$ and final longitudinal coordinate $L_{j,\mathcal{TL}}$ for sets $\mathcal{S}_k,\ k \in \mathcal{LC}$ is desirable. K-Means [Mac67] clustering is used with a fixed number of iterations to cluster samples based on the above mentioned three attributes into a user-defined number of clusters. Then a user-defined number of samples is collected from each cluster. Consider one lane change area, represented by the polygon $\hat{\mathcal{P}}_k$. Let $\dot{L}_{m_i}$, $L_{m_i}$ and $t_{m_i}$ denote and the mean longitudinal velocity, coordinate and end time of cluster $i$ for this area. Let furthermore $L_{m_i,0} = L_{m_i} - \dot{L}_{m_i} t_{m_i}$ denote the $L$-axis intercept of cluster $i$. All clusters are sorted based on the euclidean distance:

$$d_i = \left\| \begin{bmatrix} L_{m_i,0} & \dot{L}_{m_i} \end{bmatrix}^\intercal - \begin{bmatrix} L_0 & \dot{L}_0 \end{bmatrix}^\intercal \right\|_2, \tag{6.1.10}$$

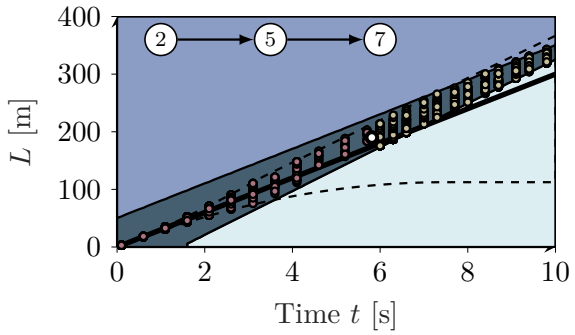
with the initial $L$ coordinate $L_0$ and velocity $\dot{L}_0$. Most samples are chosen from clusters with the minimal distance $d_i$. This has the effect to guide the sampling such that the ego-vehicle has the least need to change it's kinetic energy by decelerating or accelerating.

**Synthesis of Longitudinal and Lateral Trajectories**

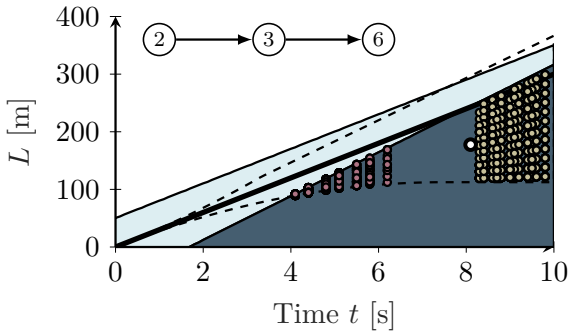
The first step consists of calculating a set of candidate trajectories for all identified maneuver options, using the set of sample points. Next, methods from optimal control for trajectory generation are chosen to ensure comfort for the passengers. The lateral trajectory synthesis uses fifth-order polynomials. It is shown in [Wer+10] that they result in jerk-optimal trajectories.
As already mentioned in Chapter 5, lane change maneuver options can be classified into immediate and delayed lane changes. Intuitively, an immediate lane change is the case if the ego-vehicle can directly start to move laterally towards the target lane because the point $\begin{bmatrix} L = 0\,\mathrm{m} & t = 0\,\mathrm{s} \end{bmatrix}^\intercal$ is in the lane change area. Consider again Figure 6.2. Here, area ⑤ is an immediates lane change area and area ③ is a delayed lane change area. Delayed describes the fact, that the ego-vehicle has to stick on its current lane for some time $t$ before starting to move laterally towards the target lane.
The distinction is important for the trajectory calculation. In the case of immediate lane changes, only one segment $N_1(t)$ is used that takes the ego-vehicle directly to the target lane with correct orientation with the lane's centerline:

$$\Psi(t_{j,\mathcal{TL}}) = \Psi_{\mathcal{TL}}(L_{j,\mathcal{TL}}), \tag{6.1.11}$$

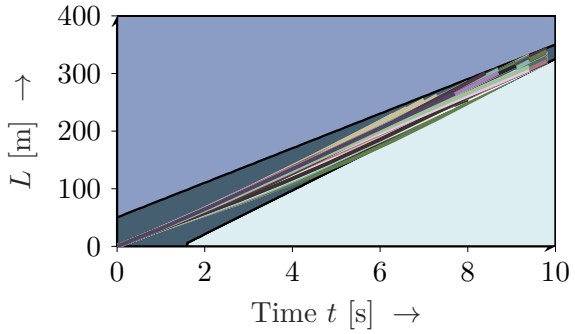
with the orientation of the ego-vehicle $\Psi$ and the orientation of the target lane centerline (TL) $\Psi_{\mathcal{TL}}$, both in the cuvilinear coordinate frame. Additionally, zero lateral acceleration and velocity is imposed to allow for a complete specification of the lateral polynomial trajectory. On the other hand, two segments are used for a delayed lane change. First, a trajectory segment $N_1(t)$ that orients the ego-vehicle correctly with the current lane's centerline at the first sample point (red in first row of Figure 6.2) is calculated.



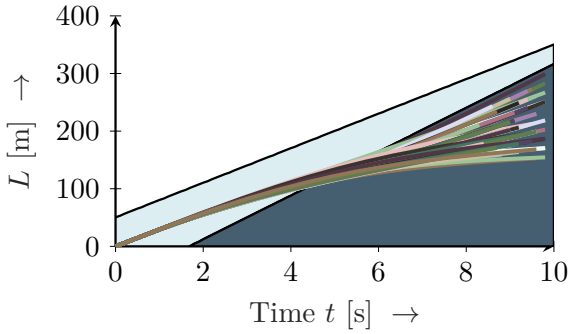

(a) Distribution of sample points for the immediate lane change maneuver.



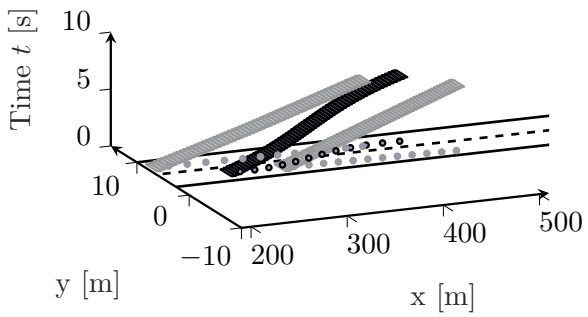

(b) Distribution of sample points for the delayed lane change maneuver.



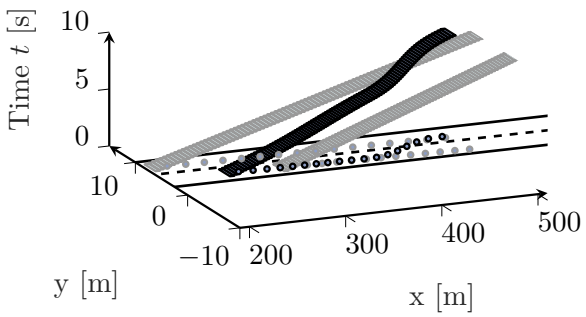

(c) Optimized trajectory set for the immediate lane change maneuver.




(d) Optimized trajectory set for the delayed lane change maneuver.




(e) Minimum total cost trajectory for the immediate lane change maneuver.




(f) Minimum total cost trajectory for the delayed lane change maneuver.

Figure 6.2.: Procedure for immediate (left column, ②→⑤→⑦) and delayed (right column, ②→③→⑥) lane change maneuver. Top row: Sample points and reachable set. Middle row: Calculated longitudinal trajectories. Bottom row: Trajectory with minimum total cost $J_{\mathrm{T}}^{\mathrm{S}}$ using $\lambda = 0.6$.

Afterwards a segment $N_2(t)$ that takes the ego-vehicle to the target lane with correct orientation at the end of the maneuver is generated. Therefore, two constraints arise:

$$\begin{aligned} \Psi(t_{j,\mathcal{CL}}) &= \Psi_{\mathcal{CL}}(L_{j,\mathcal{CL}}), \\ \Psi(t_{j,\mathcal{TL}}) &= \Psi_{\mathcal{TL}}(L_{j,\mathcal{TL}}), \end{aligned} \tag{6.1.12}$$

now also taking into account the orientation of the current lane centerline (CL) $\Psi_{\mathcal{CL}}(L_{j,\mathcal{CL}})$ in the curviliear frame. This happens between the red and beige sample points in the first row of Figure 6.2. Again, additionally zero lateral acceleration and velocity is imposed to completely specify the polynomials.
The calculation of each polynomial segment:

$$\begin{aligned} N_j(t) &= a_{j,o} + a_{j,1}t + a_{j,2}t^2 + a_{j,3}t^3 + a_{j,4}t^4 + a_{j,5}t^5 \\ &= \mathbf{p}_{N_j}^\intercal \mathbf{t} = \begin{bmatrix} a_{j,0} & a_{j,1} & a_{j,2} & a_{j,3} & a_{j,4} & a_{j,5} \end{bmatrix}^\intercal \begin{bmatrix} 1 & t & t^2 & t^3 & t^4 & t^5 \end{bmatrix} \end{aligned} \tag{6.1.13}$$

is straightforward given the boundary conditions that the polynomial should fulfil.
The proposed novel approach for the calculation of longitudinal trajectories is motivated by quadrotor flight. For this, also two fifth-order polynomial segments are used for the calculation of the longitudinal trajectory $L(t)$ and builds upon the work of [Bry12]. The main idea is to transform a constrained optimization problem into an unconstrained one to reduce computational cost and enhance numerical robustness. It is shown in [Bry12] that a cost $J_L^{\mathrm{S}}(t_{\mathrm{d}})$, penalizing the squares of the derivatives of the polynomial $L(t)$, for one single segment with duration $t_{\mathrm{d}}$ can be expressed in terms of the coefficients $\mathbf{p}_L \in \mathbb{R}^{n+1}$ of a $n$'th order polynomial of the segment and a final time dependent weight matrix:

$$J_L^{\mathrm{S}}(t_{\mathrm{d}}) = \mathbf{p}_L^\intercal \mathbf{Q}_L(t_{\mathrm{d}}) \mathbf{p}_L, \tag{6.1.14}$$

$\mathbf{Q}_L \in \mathbb{R}^{(n+1)\times(n+1)}$. The order $n = 5$ is used. Refer to [Bry12] for the construction of $\mathbf{Q}_L$. For the application at hand, a matrix $\mathbf{Q}_L = 0.7\mathbf{Q}_{L,\mathrm{acc}} + 0.3\mathbf{Q}_{L,\mathrm{jerk}}$ that punishes acceleration ($\mathbf{Q}_{L,\mathrm{acc}}$) and jerk ($\mathbf{Q}_{L,\mathrm{jerk}}$) is used. When using two segments, the calculation of a total cost can be conducted:

$$J_{L,\mathrm{total}}^{\mathrm{S}} = \begin{bmatrix} \mathbf{p}_{L,1} \\ \mathbf{p}_{L,2} \end{bmatrix}^\intercal \begin{bmatrix} \mathbf{Q}_{L,1}(t_1) & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{L,2}(t_2) \end{bmatrix} \begin{bmatrix} \mathbf{p}_{L,1} \\ \mathbf{p}_{L,2} \end{bmatrix}. \tag{6.1.15}$$

Next is the formulation of the constraint Equation:

$$\mathbf{A}_{L,\mathrm{total}} \begin{bmatrix} \mathbf{p}_{L,1} \\ \mathbf{p}_{L,2} \end{bmatrix} = \begin{bmatrix} \mathbf{d}_{L,1} \\ \mathbf{d}_{L,2} \end{bmatrix}, \tag{6.1.16}$$

with $\mathbf{A}_{L,\mathrm{total}} \in \mathbb{R}^{2(n+1)\times 2(n+1)}$, $\mathbf{d}_{L,i} \in \mathbb{R}^{n+1}, i \in \{1,2\}$. The constraints are placed on position, velocity and acceleration at the segment boundaries. If some are not known in advance, only continuity constraints are imposed. Then the total cost is reformulated:

$$J_{\mathrm{L,total}}^{\mathrm{S}} = \begin{bmatrix} \mathbf{d}_{\mathrm{L},1} \\ \mathbf{d}_{\mathrm{L},2} \end{bmatrix}^\intercal \mathbf{A}_{\mathrm{L,total}}^{-\intercal} \mathbf{Q}_{\mathrm{L,total}} \mathbf{A}_{\mathrm{L,total}}^{-1} \begin{bmatrix} \mathbf{d}_{\mathrm{L},1} \\ \mathbf{d}_{\mathrm{L},2} \end{bmatrix} \tag{6.1.17}$$

and reordered for grouping fixed and free derivatives:

$$J_{\mathrm{L,total}}^{\mathrm{S}} = \begin{bmatrix} \mathbf{d}_{\mathrm{F}} \\ \mathbf{d}_{\mathrm{O}} \end{bmatrix}^\intercal \begin{bmatrix} \mathbf{R}_{\mathrm{FF}} & \mathbf{R}_{\mathrm{FO}} \\ \mathbf{R}_{\mathrm{OF}} & \mathbf{R}_{\mathrm{OO}} \end{bmatrix} \begin{bmatrix} \mathbf{d}_{\mathrm{F}} \\ \mathbf{d}_{\mathrm{O}} \end{bmatrix}. \tag{6.1.18}$$

Herein $\mathbf{d}_\mathrm{F}$ are the imposed, fixed derivatives on segment boundaries and $\mathbf{d}_\mathrm{O}$ are free and can float. The dimensions of matrices $\mathbf{R}_\mathrm{FF}$, $\mathbf{R}_\mathrm{FO}$, $\mathbf{R}_\mathrm{OF}$, $\mathbf{R}_\mathrm{OO}$ depend on the number of fixed constraints. All free derivatives for achieving optimal cost $\mathbf{d}_\mathrm{O}^*$ according to $J_\mathrm{L,total}^\mathrm{S}$ can be analytically obtained:

$$\frac{\partial J_\mathrm{L,total}^\mathrm{S}}{\partial \mathbf{d}_\mathrm{O}} \stackrel{!}{=} 0, \quad \mathbf{d}_\mathrm{O}^* = -\mathbf{R}_\mathrm{OO}^{-1}\mathbf{R}_\mathrm{FO}^\intercal \mathbf{d}_\mathrm{F}. \tag{6.1.19}$$

The coefficients of the polynomials are obtained by substitution into the constraint Equation (6.1.16). Only the positions according to the sampling points and initial kinematic quantities are fixed, whereas all remaining derivatives are free and result from the above optimization. Notice that both the original formulation in [Ric+16] and also above treatment describes the approach in a general setting. It is up to the user which derivatives are chosen to be fixed and free. Other choices as the one used here are possible, and the generalization is straightforward. [Ric+16] also shows that instead of using the transformation to a unconstrainted problem, it is possible to simply solve the resulting Quadratic Program at the cost of increased runtimes.
Using this method, the longitudinal trajectories $L_j(t)$ consisting of two polynomial segments for each sample $\mathcal{S}_j$ are obtained. Note that an unfixed planning horizon is used that corresponds to the time $t_{j,\mathcal{TL}}$ of a sample. This gives the approach flexibility. Several generated longitudinal trajectories are shown in the second row of Figure 6.2 for both maneuver options.

## 6.2. Interaction-Aware Trajectory Assessment

This Section introduces the assessment of the calculated trajectories. Each trajectory is checked for feasibility and assigned an interaction-aware cost. It is a three-step process involving three different cost functions that will be described in this Section.

#### Safety Check and Interaction-Aware Assessment of Sample Trajectories

The goal is to find a fixed number of best trajectories for each maneuver options among the sampled ones. A multi-stage optimization is used for this. Using the aforementioned scheme for the construction of the sample set $\mathcal{S}_i$, $i \in \mathcal{LC}$, calculation of trajectories $\begin{bmatrix} L_j(t) & N_j(t) \end{bmatrix}^\intercal$ and transformation to the ego frame, the following set of trajectories is obtained:

$$\mathcal{T}_k = \{\tau_j\}_{j=1}^{n_\mathcal{T}}, k \in \mathcal{LC}, \tag{6.2.1}$$

with a total of $n_\mathcal{T}$ trajectories $\tau_j = \begin{bmatrix} X_j(t) & Y_j(t) \end{bmatrix}^\intercal$. The inner optimization problem is as follows:

$$\begin{aligned} &\min J_\mathrm{V}^\mathrm{S}(\boldsymbol{\tau}_j), \text{ subject to:} \\ &\boldsymbol{\tau}_j \in (\mathcal{T}_\mathrm{CF} \cap \mathcal{T}_\mathrm{VC} \cap \mathcal{T}_\mathrm{OTPC} \cap \mathcal{T}_\mathrm{TR}), \end{aligned} \tag{6.2.2}$$

with the set $\mathcal{T}_\mathrm{CF}$ of collision free trajectories, the set $\mathcal{T}_\mathrm{VC}$ of trajectories that fulfill imposed vehicle dynamics and nonholonomic constraints. The differential flatness property of a

kinematic nonlinear double-track model is used to check if steering angles and rates fulfill box constraints. Refer to the Appendix A.7 for a mathematical definition of differential flatness and Appendix A.8 for more details on the used vehicle model. The set $\mathcal{T}_{\mathrm{OTPC}}$ consists of the trajectories that fulfill constraints on Time Headways and Time-to-Collisions to surrounding traffic and with $\mathcal{T}_{\mathrm{TR}}$, the set satisfying traffic rules like for example speed limits. Box constraints for $\mathcal{T}_{\mathrm{OTPC}}$ and for $\mathcal{T}_{\mathrm{VC}}$ are used and can be scaled with the urgency of the lane change. The cost function is:

$$\begin{aligned} J_{\mathrm{V}}^{\mathrm{S}} =& \omega_1 |\ddot{X}_{\max}| + \omega_2 |\ddot{Y}_{\max}| + \omega_3 |X^{(3)}_{\max}| + \omega_4 |Y^{(3)}_{\max}| + \\ &+ \omega_5 \overline{|\ddot{X}|} + \omega_6 \overline{|\ddot{Y}|} + \omega_7 \overline{|X^{(3)}|} + \omega_8 \overline{|Y^{(3)}|} + \\ &+ \omega_9 \left( |\min\left(\dot{X}_{\mathrm{TL}}, \dot{X}_{\mathrm{des}}\right) - \dot{X}_{\mathrm{T}}| \right), \end{aligned} \tag{6.2.3}$$

with weighting factors $\omega_i \in \mathbb{R}$, maximum and mean absolute values $\overline{|\cdot|}$ of acceleration and jerk in $X$ and $Y$ direction, the estimated speed on the target lane $\dot{X}_{\mathrm{TL}}$, the desired velocity given by the mission planning $\dot{X}_{\mathrm{des}}$ and final ego-vehicle velocity $\dot{X}_{\mathrm{T}}$. All weighting factors $\omega_i \in \mathbb{R}$ were empirically determined using simulations. Section 8.2 in Chapter 8 presents an alernative based on IRL (**I**nverse **R**einforcement **L**earning) and it shown how weighting factors can be optimized given a set of demonstration trajectories. The optimization results in an ordered set of $n_{\tilde{T}}$ feasible trajectories $\tilde{\mathcal{T}}_k,\ k \in \mathcal{LC}$.
To account for interaction-awareness, the feasible trajectories are incorporated into the scene prediction engine [Wis+18] that uses the Intelligent Driver Model [Tre+00] and evaluate the reactions of all surrounding traffic participants. A new method to measure the interaction quantitatively is proposed. The idea is to use generalized kinetic energies. The ego-vehicle and all other traffic participants have certain kinetic energies calculated based on initial predictions and that they ideally would maintain. It is, therefore, very intuitive to compare kinetic energies. However, one challenge that arises is the correct handling of big and heavy trucks. These would unbalance the calculation of the energies strongly. This is handled by using a normalized mass $\hat{m}$ that also accounts for the fact that big trucks have stronger braking systems and therefore can decrease kinetic energy stronger than passenger vehicles. The calculated trajectories in curvilinear coordinates are used to obtain the expression:

$$\hat{E}(t) = \frac{1}{2}\hat{m}\left(\dot{L}^2(t) + \dot{N}^2(t)\right) \tag{6.2.4}$$

It is propose to use $\hat{m}_P = 1\,\mathrm{kg}$ for passenger vehicles, $\hat{m}_M = 0.8\,\mathrm{kg}$ for motorbikes and $\hat{m}_T = 3\,\mathrm{kg}$ for trucks. In the outer optimization loop, the aim is to minimize an interaction-aware cost function $J_{\mathrm{I}}^{\mathrm{S}}$:

$$J_{\mathrm{I}}^{\mathrm{S}} = \lambda \tilde{E}_{\mathrm{Ego}} + (1-\lambda)\tilde{E}_{\mathrm{OTP}}. \tag{6.2.5}$$

The tilde ($\tilde{E}$) in Equation (6.2.5) reflects the fact, that the values of the energies are normalized to always make them comparable to each other. The approach benefits from good interpretability, and there is only one parameter $\lambda \in \mathbb{R}$ that realizes the transition between cooperative and non-cooperative behavior. Subsequently, all steps needed to evaluate Equation (6.2.5) are described. First is the formulation of several generalized

kinetic energies for all $i \in \{0 \cup \mathcal{O}\}$ vehicles. The index $i = 0$ reflects the ego-vehicle and the index set $\mathcal{O}$ all $n_{\mathrm{OTP}}$ surrounding traffic participants:

$$E_{i,\mathrm{prior}}(t) = \frac{1}{2}\hat{m}_{(\cdot)}\left(\dot{L}^2_{i,\mathrm{prior}}(t) + \dot{N}^2_{i,\mathrm{prior}}(t)\right), \tag{6.2.6}$$

$$E_{i,\mathrm{post}}(t) = \frac{1}{2}\hat{m}_{(\cdot)}\left(\dot{L}^2_{i,\mathrm{post}}(t) + \dot{N}^2_{i,\mathrm{post}}(t)\right), \tag{6.2.7}$$

$$\overline{|E_i|} = \frac{1}{n_i}\sum_{j=1}^{n_i} \max\left(0, E_{i,\mathrm{prior}}\left(t_j\right) - E_{i,\mathrm{post}}\left(t_j\right)\right) \tag{6.2.8}$$

$$\overline{|E_0|} = \frac{1}{n_0}\sum_{j=1}^{n_0} \left|E_{0,\mathrm{prior}}\left(t_j\right) - E_{0,\mathrm{post}}\left(t_j\right)\right| \tag{6.2.9}$$

$$E_{\mathrm{OTP}} \coloneqq \frac{1}{n_{\mathrm{OTP}}}\sum_{i=1}^{n_{\mathrm{OTP}}} |E_i|, \quad E_{\mathrm{Ego}} \coloneqq \overline{|E_0|}, \tag{6.2.10}$$

with the number of timesteps $n_i$. The subscript prior reflects the situation if the ego-vehicle would continue with it's initial velocity on the start lane. Special attention needs to be paid at the differences of the mean generalized energy calculation for other traffic participants and the ego-vehicle in Equations (6.2.8) and (6.2.9). Consider the case in which the ego-vehicle conducts a lane change with the result that the vehicle on the start lane behind it can now drive faster and gain velocity. This results in $E_{i,\mathrm{prior}}(t_j) - E_{i,\mathrm{post}}(t_j) < 0$ such that the generalized energy is raised. The approach should not punish such behavior and the max operator in Equation (6.2.8) realizes that. The ego-vehicle is handled differently and it is assumed, that it wants to keep it's initial generalized energy. This is accounted for by using the absolute value in Equation (6.2.9). Notice, that this does not prohibit speeding up, since the cost functional in Equation (6.2.3) of the previous optimization step puts weight on achieving the desired velocity.
A higher value of $\lambda$ in Equation (6.2.5) punishes changes of the normalized generalized kinetic energy of the ego-vehicle more compared to the influence onto the surrounding traffic. This, in turn, leads to a more comfortable but non-cooperative ride. As a result, it is expected to see lower values of jerk and accelerations but probably at the cost of lower Time-to-Collisions within the range of feasible $TTC$s. At the end, both cost functions are normalized such that $\tilde{J}^{\mathrm{S}}_{\mathrm{V}}, \tilde{J}^{\mathrm{S}}_{\mathrm{I}} \in [0, 1]$ and the total cost formulated:

$$J^{\mathrm{S}}_{\mathrm{T}} = \nu_1 \tilde{J}^{\mathrm{S}}_{\mathrm{V}} + \nu_2 \tilde{J}^{\mathrm{S}}_{\mathrm{I}}, \tag{6.2.11}$$

with weight factors $\nu_1$ and $\nu_2$.

## 6.3. Evaluation of the Sampling-based Lane Change Planning

The scenario under consideration is the running example of this Chapter, refer to Figure 6.1. The simulation environment described in [Wis+16] is used that takes into account uncertain measurements of the vehicle's sensors and provides the correct data structures that allow the quick deployment of developed algorithms. Let $X_{\mathrm{LB}}(t)$ and $\dot{X}_{\mathrm{LB}}(t)$ denote the longitudinal coordinate and velocity of the traffic participant on the target lane behind

Table 6.1.: Vehicle dynamics of the ego-vehicle corresponding to both minimum $J_{\mathrm{T}}^{\mathrm{S}}$ results in Figure 6.2.

| | $\max(\lvert\cdot\rvert)$ | | | | $\overline{\lvert\cdot\rvert}$ | | | | $\Delta\dot{X} = \dot{X}_{(\cdot)} - \dot{X}_{(\cdot)}$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\ddot{X}$ | $\ddot{Y}$ | $X^{(3)}$ | $Y^{(3)}$ | $\ddot{X}$ | $\ddot{Y}$ | $X^{(3)}$ | $Y^{(3)}$ | T-0 | T-des |
| Maneuver | $[\frac{\mathrm{m}}{\mathrm{s}^2}]$ | $[\frac{\mathrm{m}}{\mathrm{s}^2}]$ | $[\frac{\mathrm{m}}{\mathrm{s}^3}]$ | $[\frac{\mathrm{m}}{\mathrm{s}^3}]$ | $[\frac{\mathrm{m}}{\mathrm{s}^2}]$ | $[\frac{\mathrm{m}}{\mathrm{s}^2}]$ | $[\frac{\mathrm{m}}{\mathrm{s}^3}]$ | $[\frac{\mathrm{m}}{\mathrm{s}^3}]$ | $[\frac{\mathrm{m}}{\mathrm{s}}]$ | $[\frac{\mathrm{m}}{\mathrm{s}}]$ |
| ②→⑤→⑦ | 1.49 | 0.68 | 2.47 | 1.13 | 0.87 | 0.42 | 0.44 | 0.43 | 5.24 | 1.91 |
| ②→③→⑥ | 2.64 | 1.59 | 4.6 | 4.20 | 1.38 | 0.44 | 0.93 | 0.72 | 0.42 | -2.93 |

Abbreviations: T-0: Longitudinal velocity difference between ego velocity at beginning and end of lane change, T-des: Longitudinal velocity difference between ego velocity at end of lane change and desired velocity (here: speed limit on target lane)

Table 6.2.: Situation features corresponding to both minimum $J_{\mathrm{T}}^{\mathrm{S}}$ results in Figure 6.2. CB (**C**urrent lane **B**ack), CF (**C**urrent lane **F**ront), LB (**L**eft lane **B**ack), and LF (**L**eft lane **F**ront) correspond to regions of interest around the ego-vehicle. Notice that a vehicle on the left lane behind the ego-vehicle acts as the target lane follower in case of an immediate lane change but as the target lane preceding vehicle in case of a delayed one.

| | $\min(TTC_{(\cdot)})$ | | | | Energies $E$ | |
|---|---|---|---|---|---|---|
| | CB | CF | LB | LF | OTP | Ego |
| Maneuver | [s] | [s] | [s] | [s] | [J] | [J] |
| ②→⑤→⑦ | - | 14.46 | 19.57 | - | 22.39 | 100.28 |
| ②→③→⑥ | - | $\infty$ | - | $\infty$ | 0 | 80.36 |

and $X_{\mathrm{CF}}(t)$ and $\dot{X}_{\mathrm{CF}}(t)$ on the start lane in front of the ego-vehicle. Now the initial values of the longitudinal coordinate and velocity of the vehicle on the target lane are randomized by sampling from uniform probability distributions, $X_{\mathrm{LB}} \sim U(-85\,\mathrm{m}, -25\,\mathrm{m})$ and $\dot{X}_{\mathrm{LB}} \sim U(27.5\,\mathrm{m/s}, 46.5\,\mathrm{m/s})$. The longitudinal velocities of the ego-vehicle and the traffic participant in front of it are always fixed to $\dot{X}_{\mathrm{Ego}} = \dot{X}_{\mathrm{CF}} = 30\,\mathrm{m/s}$. For the ego-vehicle's desired velocity, the value $\dot{X}_{\mathrm{des}} = 33.3\,\mathrm{m/s}$ is used. The evaluation of the approach is done focusing on the introduced interaction-aware quality measure introduced in the last Section. A total of 500 simulation runs are used for the evaluation. The histograms in Figure 6.3 show two critical features, the minimum Time-to-Collision $\min(TTC)$ and the mean longitudinal acceleration $\overline{\ddot{X}}_{\mathrm{Ego}}$ of the ego-vehicle. They show that a high $\lambda$ tends to result in lower mean longitudinal acceleration since changes of kinetic energy that correspond to ego-velocity changes are stronger punished. A side effect of this is that the $TTC$s are also lowered, resulting in a more egoistic behavior.

The bottom row of Figure 6.2 shows the best trajectories in $(x, y, t)$ domain for both maneuver options for one simulation run corresponding to the configuration shown in Figure 6.1. Table 6.1 shows a subset of the calculated features. The factor $\lambda = 0.6$ was used for generating these results. The rows in Table 6.1 correspond to both identified maneuver variants. It might be somewhat surprising that the maximum and mean values

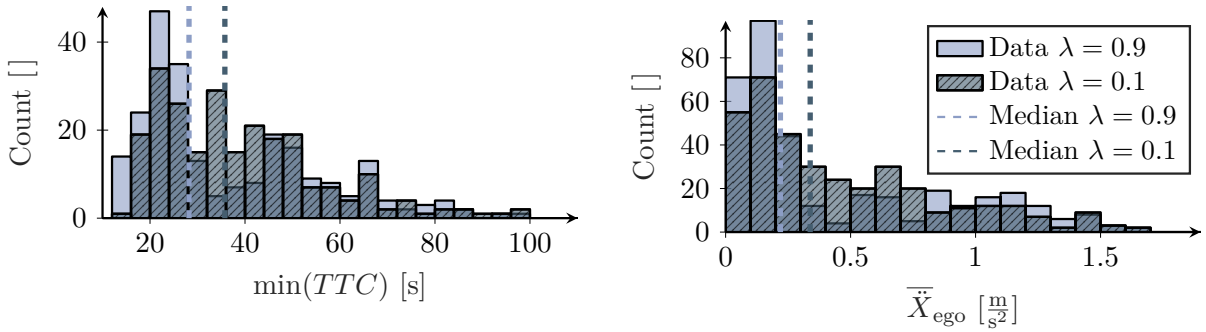
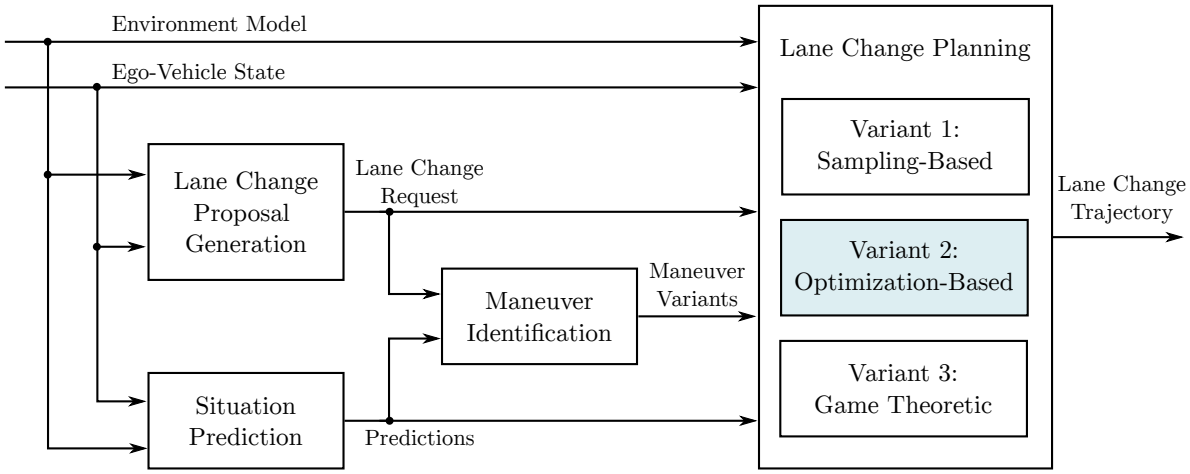



Figure 6.3.: Histogram and median values of $\min(TTC)$ (a) and $\overline{\ddot{X}}$ (b) for $\lambda = 0.1$ and $\lambda = 0.9$.

of acceleration and jerk in both directions $X$ and $Y$ are higher in the second maneuver. The ego-vehicle waits to merge behind the faster vehicle from the back on the target lane. The explanation for this lies in the fact that the results correspond to the first planning cycle at $t = 0\,\text{s}$ with a fixed maximum time horizon of $t_{\max} = 10\,\text{s}$ in which the lane change has to be conducted. Looking at the traffic scene, refer to Figure 6.1, it is obvious that the ego-vehicle has several options that are traded-off by the weighting factors in Equations (6.2.3), (6.2.5) and (6.2.11). Since the weight $\omega_9$ in Equation (6.2.3) is comparably high to the other weights, the ego-vehicle prefers a lane change where it first breaks slightly to reduce its velocity to change lane afterward and during the lane change speed up to achieve the target velocity. The lane change duration for this is short, and therefore the lateral dynamics are also higher compared to the immediate lane change in row 1 of Table 6.1. Considering the generalized, unnormalized energies $E_{\text{OTP}}$ and $E_{\text{Ego}}$, it can be seen in Table 6.2 that the influence of maneuver ②→⑤→⑦ onto the surrounding traffic is stronger than it is for maneuver ②→③→⑥. Note, that the accompanying video to the original publication [Sch+19a] illustrates the results of the proposed framework in more complex highway traffic scenarios with more surrounding traffic participants and curves. This Chapter also illustrates the profound influence of the choice of cost function weights on the optimization results. The weights here are chosen manually in an empirical fashion. As the process is cumbersome, Inverse Reinforcement Learning is used in Section 8.2 in Chapter 8 to circumvent this problem.

# 7

# Optimization-Based Lane Change Planning




The sampling-based approach discussed in the previous Chapter has several limitations. First, the memory consumption is high, as is the case for most sampling strategies. Second and more importantly, the distribution of sampling points does not generalize to arbitrary traffic situations when there are lane changes of surrounding vehicles. This Chapter, therefore, discusses an optimization-based lane change planning approach that does not suffer from these shortcomings. The main contribution here is the formulation of affine time-variant safety constraints for Time Headways and Time-to-Collisions. The subsequent discussion is largely based on the publication [Sch+19b] which in turn is based on [Nil+17]. The approach can work with the complete traffic scene topology graph or its complexity reduced variant discussed in Section 5.2 and can deal with lane changes of surrounding traffic participants. This Chapter mainly aims to provide a baseline for the discussion in Chapter 8 where the optimization is formulated, including additionally the target lane follower of the ego-vehicle.

## 7.1. Derivation of Safety Constraints Based on Spatiotemporal Boundaries

The optimization-based lane change planning algorithm also distinguishes immediate and delayed lane changes. Both variants are handled differently. First, some relevant notions are defined that will be used throughout the remainder of the thesis at hand. The pre-region is defined as the time interval $[0, t_{\text{pre}}]$ on the start lane. A peri-region represents the time interval $(t_{\text{pre}}, t_{\text{peri}}]$ in which a lateral movement to the target lane is realized. Finally, the post-region is the interval $(t_{\text{peri}}, t_{\max}]$ on the target lane of the respective lane change maneuver. Refer to [Nil+16a] for a detailed description of these regions. A lane change maneuver is unambiguously defined by a path in the graph and the determination of the start time $t_{\text{pre}}$ and end time $t_{\text{peri}}$ of the lateral movement. A method for obtaining these times is described next.
The polygons of lane change maneuver paths in an options graph include obstacle vehicle positions and velocities. This allows the formulation of linear time-variant $TTC$ and $THW$ safety constraints. The definition of a $THW$ and $TTC$ in the curvilinear coordinate frame at a query time $t_{\text{q}}$ is as follows:

$$THW(t_{\text{q}}) = \frac{L_{\text{L}}(t_{\text{q}}) - L_{\text{F}}(t_{\text{q}})}{\dot{L}_{\text{F}}(t_{\text{q}})}, \tag{7.1.1}$$

$$TTC(t_{\text{q}}) = \frac{L_{\text{L}}(t_{\text{q}}) - L_{\text{F}}(t_{\text{q}})}{\dot{L}_{\text{F}}(t_{\text{q}}) - \dot{L}_{\text{L}}(t_{\text{q}})}. \tag{7.1.2}$$

Herein the subscripts $(\cdot)_{\text{F}}$ and $(\cdot)_{\text{L}}$ denote the follower and leader vehicle. Both safety constraints play a central role in the Section on the trajectory optimization. It is necessary to derive the time durations of the pre-, peri- and post-regions of the lane change maneuver. The times $t_{\text{pre}}$ and $t_{\text{peri}}$ are the two degrees of freedom. They are chosen using a prior safety evaluation of the spatiotemporal lane change area. The time interval $[t_1, t_2]$ is identified by analyzing the Time Headway Equation (7.1.1) using the initial ego-vehicle velocity $\dot{L}_{\text{E},0}$ and predicted follower velocity $\dot{L}_{\text{F}}(t)$. A minimum spatial gap is defined as: $L_{\text{minGap}}(t) = (\dot{L}_{\text{E},0} + \dot{L}_{\text{F}}(t))THW_{\min}$, with the minimum prior Time Headway $THW_{\min}$. Next the lower and upper bounds of the lane change area $L_{\text{LC,ub}}(t)$ and $L_{\text{LC,lb}}(t)$ are used to find all intervals that fulfill the constraint: $L_{\text{LC,ub}}(t) - L_{\text{LC,lb}}(t) \geq L_{\text{minGap}}(t)$.
The next step is marking intervals as infeasible in case they are shorter than a user-defined minimum lane change time $t_{\text{LC,min}}$. The distinction between immediate and delayed lane changes is taken into account. In case of an immediate lane change, the longest interval is found and checked if it is longer than the maximum lane change time $t_{\text{LC,max}}$. If that is the case, then the peri-region is then chosen to be $(t_{\text{pre}}, t_{\text{peri}}] = (t_1, t_1 + t_{\text{LC,max}}]$, else $(t_{\text{pre}}, t_{\text{peri}}] = (t_1, t_2]$. Therefore taking into account that immediate lane changes should start as early as possible. In the case of delayed lane changes, the same check against the maximum lane change time is done. If the interval is longer, then the peri-region is chosen to be $(t_{\text{pre}}, t_{\text{peri}}] = (t_2 - t_{\text{LC,max}}, t_2]$, else $(t_{\text{pre}}, t_{\text{peri}}] = (t_1, t_2]$. The idea is to conduct a delayed lane change as late as possible to reduce the need for decelerations. The duration of the pre- and post-regions are then obtained as the remaining times before and after

the peri-region in the planning horizon. In the case of a receding horizon application, the durations are adapted according to the lateral progress towards the target lane.

At this point, there is still ambiguity left since a path consists of possibly multiple traversed areas in the start lane, exactly one lane change area, and possibly multiple traversed target lane areas. Two Dijkstra shortest path searches with unit cost on the edges in the graph are used to obtain the complete maneuver path. For each lane change area, the start lane free space corresponding to $t_{\text{pre}}$ is found, and a search from that area in the backward direction to the start areas of the ego-vehicle is conducted. Another search in the forward direction from the target area node corresponding to $t_{\text{peri}}$ to all areas in the set $\mathcal{T}$ completes the path. Only the shortest path is picked for the forward direction search in case that several paths are found. This procedure integrates the preference to traverse as few areas as possible.

## 7.2. Lane Change Planning using Quadratic Programming

The system model used for the motion planning within the contribution at hand is a triple integrator for both the lateral and longitudinal movement of the vehicle. The time-discrete description of the system dynamics is obtained using the theory of sampled data systems. It is assumed that the control input, here the longitudinal and lateral jerk, is piecewise constant during each sampling interval. Assume further that for each lane change maneuver $i$, there are lower and upper bounds for the longitudinal and lateral movement, determined by the maneuver's pre-, peri- and post-region. These are $L_{i,\text{ub}}(k)$, $L_{i,\text{lb}}(k)$, $N_{i,\text{ub}}(k)$ and $N_{i,\text{lb}}(k)$. Choosing how to impose safety constraints is up to the user and will determine how conservative the vehicle behaves. The trajectory optimization problem is formulated similarly to [Nil+16a]. It is however conducted in the curvilinear coordinate frame to deal with curvy highway scenarios using the following small-angle approximations $\ddot{L} \approx \ddot{X}$ and $\ddot{N} \approx \ddot{Y} - \kappa(L)\dot{L}^2$, with the curvature of the reference lane at the coordinate $L$ of the curvilinear frame $\kappa(L)$, refer to [Rat16]. Furthermore, compared to [Nil+16a], a triple integrator system dynamics is used and the jerk introduced into both the longitudinal and lateral cost functions. Finally, time-variant Time Headway and Time-to-Collision safety constraints are introduced into the longitudinal optimization making direct use of the maneuver representation using polygons as described in Chapter 5.

The longitudinal cost function is defined as:

$$J_L^{\text{O}} = \sum_{k=1}^{N_p} \left( \alpha_0 \left( \dot{L}_{\text{E}}(k) - \dot{L}_{\text{des}}(k) \right)^2 + \alpha_1 \ddot{L}_{\text{E}}(k)^2 + \alpha_2 L_{\text{E}}^{(3)}(k)^2 \right), \tag{7.2.1}$$

with the desired longitudinal velocity $\dot{L}_{\text{des}}(k)$, weighting factors $\alpha_j, j \in \{0, 1, 2\}$ and the

last sample $n_\mathrm{p}$. The longitudinal optimization is subject to the constraints:

$$\begin{aligned} L_\mathrm{lb}(k) \leq L_\mathrm{E}(k) \leq L_\mathrm{ub}(k), \\ \dot{L}_\mathrm{min} \leq \dot{L}_\mathrm{E}(k) \leq \dot{L}_\mathrm{max}, \\ \ddot{L}_\mathrm{min} \leq \ddot{L}_\mathrm{E}(k) \leq \ddot{L}_\mathrm{max}, \\ L^{(3)}_\mathrm{min} \leq L^{(3)}_\mathrm{E} \leq L^{(3)}_\mathrm{max}, \end{aligned} \tag{7.2.2}$$

$$\begin{aligned} L_\mathrm{E}(k+1) &= L_\mathrm{E}(k) + \dot{L}_\mathrm{E}(k)h + \ddot{L}_\mathrm{E}(k)\frac{h^2}{2} + L^{(3)}_\mathrm{E}(k)\frac{h^3}{6}, \\ \dot{L}_\mathrm{E}(k+1) &= \dot{L}_\mathrm{E}(k) + \ddot{L}_\mathrm{E}(k)h + L^{(3)}_\mathrm{E}(k)\frac{h^2}{2}, \\ \ddot{L}_\mathrm{E}(k+1) &= \ddot{L}_\mathrm{E}(k) + L^{(3)}_\mathrm{E}(k)h, \\ L_\mathrm{E}(0) &= L_{\mathrm{E},0}, \quad \dot{L}_\mathrm{E}(0) = \dot{L}_{\mathrm{E},0}, \quad \ddot{L}_\mathrm{E}(0) = \ddot{L}_{\mathrm{E},0}, \end{aligned} \tag{7.2.3}$$

with the stepsize $h$. The jerk $L^{(3)}_\mathrm{E}(k)$ acts as the control input. Further constraints are added to account for minimum $TTC$ and $THW$. Using the upper and lower boundaries $L_{i,\mathrm{ub}}(k)$ and $L_{i,\mathrm{lb}}(k)$, the constraints are:

$$\begin{aligned} L_\mathrm{lb}(k + n_\mathrm{TTC}) \leq L_\mathrm{E}(k) + \dot{L}_\mathrm{E}(k)TTC_\mathrm{min} \\ L_\mathrm{E}(k) + \dot{L}_\mathrm{E}(k)TTC_\mathrm{min} \leq L_\mathrm{ub}(k + n_{TTC_\mathrm{min}}), \\ THW_\mathrm{min}\dot{L}_\mathrm{E}(k) + L_\mathrm{E}(k) \leq L_\mathrm{ub}(k), \\ L_\mathrm{lb}(k + n_{THW_\mathrm{min}}) \leq L_\mathrm{E}(k), \end{aligned} \tag{7.2.4}$$

with the number of samples $n_{THW_\mathrm{min}} = \lceil \frac{THW_\mathrm{min}}{h} \rceil$ and $n_{TTC_\mathrm{min}} = \lceil \frac{TTC_\mathrm{min}}{h} \rceil$ corresponding to the minimum $TTC$ and $THW$ times. The $TTC$ and $THW$ constraints are derived from their geometric meaning in spatiotemporal $(L, t)$ domain. This formulation results in a convex Quadratic Program that is solved using the OSQP (**O**perator **S**plitting Solver for **Q**uadratic **P**rograms) solver [Ste+20]. The lateral trajectory optimization is conducted after obtaining the optimal longitudinal trajectory $L^*_\mathrm{E}(k)$. The following cost function is optimized:

$$J^\mathrm{O}_N = \sum_{k=1}^{N_p} \left( \beta_0 \left( N_\mathrm{E}(k) - N_\mathrm{des}(k) \right)^2 + \beta_1 \dot{N}_\mathrm{E}(k)^2 + \beta_2 \ddot{N}_\mathrm{E}(k)^2 + \beta_3 N^{(3)}_\mathrm{E}(k)^2 \right), \tag{7.2.5}$$

with weighting factors $\beta_j, j \in \{0, 1, 2, 3\}$. $N_\mathrm{des}(k)$ represents a lateral tracking reference. It helps to push the trajectory further away from the upper and lower boundaries $N_\mathrm{lb}(k)$ and $N_\mathrm{ub}(k)$ and is further examined in Section 7.3. The lateral optimization is subject to the analogous system dynamics and initial value constraints as in Equation (7.2.3). Furthermore, the following set of constraints is applied:

$$\begin{aligned} N_\mathrm{lb}(k) \leq N_\mathrm{E}(k) \leq N_\mathrm{ub}(k), \\ \dot{L}^*_\mathrm{E}(k)\tan(-\theta_\mathrm{max}) \leq \dot{N}_\mathrm{E}(k) \leq \dot{L}^*_\mathrm{E}(k)\tan(\theta_\mathrm{max}), \\ \ddot{Y}_\mathrm{min} - \kappa(L^*_\mathrm{E}(k))(\dot{L}^*_\mathrm{E}(k))^2 \leq \ddot{N}_\mathrm{E}(k), \\ \ddot{Y}_\mathrm{max} - \kappa(L^*_\mathrm{E}(k))(\dot{L}^*_\mathrm{E}(k))^2 \geq \ddot{N}_\mathrm{E}(k), \end{aligned} \tag{7.2.6}$$

Table 7.1.: Parameters used for the experimental evaluation of the optimization-based approach.

| Longitudinal | Lateral | Remaining Parameters |
|---|---|---|
| $\alpha_0 = 1$ | $\beta_0 = 2$ | $h = 0.5\,\mathrm{s}$ |
| $\alpha_1 = 2$ | $\beta_1 = 2$ | $TTC_{\min} = 5\,\mathrm{s}$ |
| $\alpha_2 = 2.5$ | $\beta_2 = 2.5$ | $THW_{\min} = 1\,\mathrm{s}$ |
| | $\beta_3 = 5$ | |

The trajectory optimization problem is solved sequentially, starting with the longitudinal optimization and finally solving the lateral problem. This allows to consider the coupling between $L_\mathrm{E}$ and $N_\mathrm{E}$ and to account for a maximum heading angle $\theta_{\max}$. Imposing box constraints on accelerations implies an underapproximation of Kamm's circle. The approach plans comfortable lane change maneuvers with tight and conservative bounds using simplified motion models in the form of two triple integrator systems. This choice is justified since it is used prior to a local trajectory planner in an automated driving system with less conservative bounds and uses high fidelity vehicle models to cater to potentially nonlinear effects. Furthermore, using this simplified motion model, it is ensured that tight runtime constraints can be fulfilled so that the algorithms can be implemented on current generation ECUs.

## 7.3. Evaluation of the Optimization-Based Lane Change Planning

This Section presents the results of applying the approach in simulation with a focus on trajectory optimization. The terminology used is shown in Figure 7.1. A highway entrance scenario is chosen with a total of four obstacle vehicles. The simulation environment described in [Wis+16] is used. The trigger is a lane change request to the left and maneuver variants are identified for every time step of the simulation. All results are obtained using the set of parameters shown in Table 7.1. The initial traffic scene configurations are represented in Table 7.2. Each row in Figure 7.2 corresponds to a certain time step of the simulation and shows the results.
The optimized trajectories are then injected into the forward simulation of the traffic scene, refer to Chapter 4, for an assessment of the traffic interaction based on generalized kinetic energies as introduced in [Sch+19a]. Finally, ego-vehicle dynamics and situation features for all three time steps are shown in Table 7.3. The desired longitudinal velocity is set to $\dot{L}_\mathrm{des}(k) = 38.9\,\mathrm{m/s}$. The graphs corresponding to each situation encodes at least two lane change variants and each row in Figure 7.2 always shows the feasible, least total cost $J = J_L^\mathrm{O} + J_N^\mathrm{O}$ solution.
The first row shows an immediate lane change maneuver. The ego-vehicle has to accelerate since the trailing vehicle TB on the target lane is significantly faster than the ego-vehicle. Here, the ego-vehicle comes close to its desired velocity (refer to Table 7.3). The second row in Figure 7.2 shows a delayed lane change in which the ego-vehicle has to break significantly to let vehicle LB pass and then execute the lane change within the planning

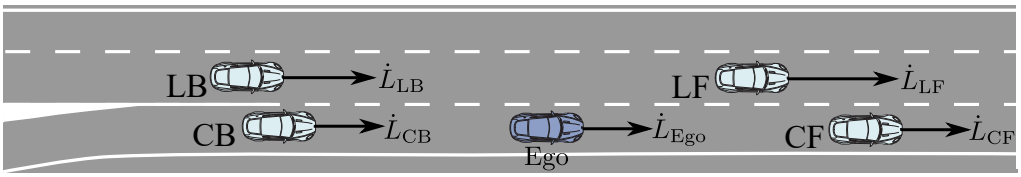



Figure 7.1.: Illustation of the traffic scene used for the evaluation of the optimization-based lane change planning approach.

Table 7.2.: Initial traffic scene cofigurations corresponding to Figure 7.2. CB (**C**urrent Lane **B**ack), CF (**C**urrent Lane **F**ront), LB (**L**eft Lane **B**ack) and LF (**L**eft Lane **F**ront) refrence the vehicles at $t = 0\,\mathrm{s}$.

| | $L_{0,(\cdot)}$ | | | | | $\dot{L}_{0,(\cdot)}$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Maneuver | Ego | CB | CF | LB | LF | Ego | CB | CF | LB | LF |
| in Figure 7.2 | m | m | m | [m] | [m] | $[\frac{\mathrm{m}}{\mathrm{s}}]$ | $[\frac{\mathrm{m}}{\mathrm{s}}]$ | $[\frac{\mathrm{m}}{\mathrm{s}}]$ | $[\frac{\mathrm{m}}{\mathrm{s}}]$ | $[\frac{\mathrm{m}}{\mathrm{s}}]$ |
| First row | 0 | -38.41 | 53.99 | -81.88 | 101.72 | 17.24 | 14.18 | 17.09 | 26.73 | 23.93 |
| Second row | 0 | -50.64 | 53.97 | -37.53 | / | 19.21 | 16.94 | 19.33 | 27.33 | / |
| Third row | 0 | -60.95 | 53.98 | -2.36 | / | 19.79 | 17.84 | 20.15 | 24.99 | / |

horizon. Note that the simulation environment models fluctuations of traffic participants within their respective lanes, and here the ego-vehicle is initially not centered within the start lane. The framework copes with this deviation. Finally, the third row of Figure 7.2 shows a time step 5 s and hence later than the one in the second row. This time, the ego-vehicle does not have to break severely for this delayed lane change and can conduct it more comfortably compared to both other results.

The introduction of linear time-variant constraints on the $TTC$ and $THW$ results in corresponding high values in Table 7.3. Therefore, the planned lane changes have desired safety margins. Inspection of the generalized OTP (Obstacle Traffic Participants) energies, refer to 6.2 for the mathematical definition, also shows that these constraints keep the influence on the obstacle vehicles low. Loosening these constraints results in stronger influences.

The values of the generalized kinetic energies of the ego-vehicle are proportional to the area between the constant velocity prediction $L_{\mathrm{E},0} + \dot{L}_{\mathrm{E},0}t$ and $L^*_{\mathrm{E}}(t)$, refer to the second column in Figure 7.2. This clarifies that the immediate lane change is the least comfortable while keeping with safety margins.

Finally, the sampling-based approach is compared to the optimization-based approach of this Chapter. As the presented approach is a modified variant of [Nil+17], the constraints for longitudinal and lateral acceleration are hence taken from [Nil+17]. Furthermore, the aim is to minimize the interaction cost $J_{\mathrm{I}}^{\mathrm{O}}$ in Equation (6.2.5) via using the parameter $\lambda = 0$ in Equation (6.2.5) to show performance in this interesting corner case. The example scene in Figure 6.1 of the previous Chapter 6 with parametrization $\{X_{\mathrm{E}} = 0\,\mathrm{m}, \dot{X}_{\mathrm{E}} = 33.3\,\mathrm{m/s}, X_{\mathrm{CF}} = 150\,\mathrm{m}, \dot{X}_{\mathrm{CF}} = 33.3\,\mathrm{m/s}, X_{\mathrm{LB}} = -70\,\mathrm{m}, \dot{X}_{\mathrm{LB}} = 37\,\mathrm{m/s}\}$ is used. Since the first optimization stage aims to maximize ride comfort, the above parameters are a

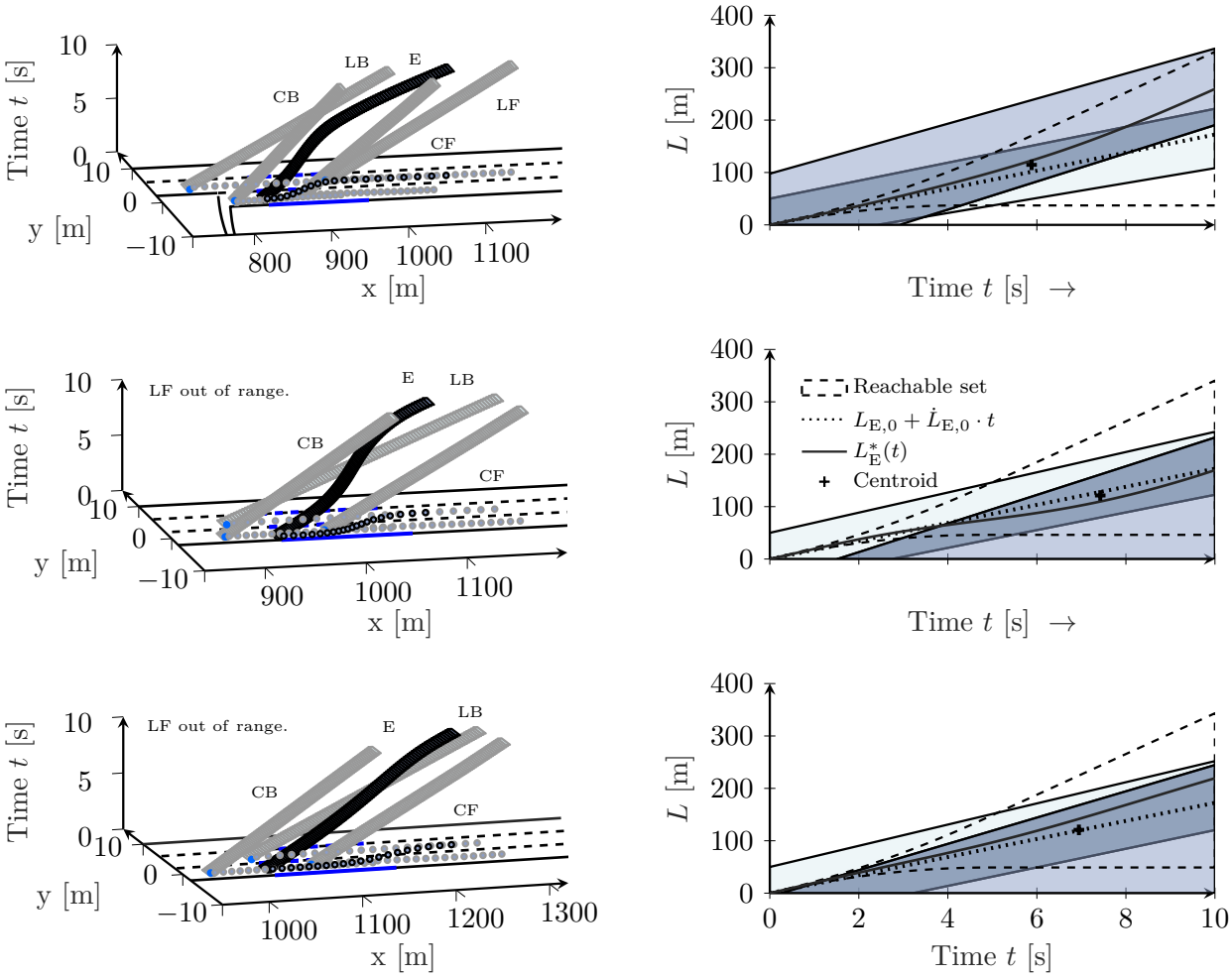



Figure 7.2.: Results of the optimization-based lane change planning approach. The rows correspond to ones in Table 7.2. A three-dimensional representation is shown in the left column whereas the polygons resulting from the maneuver identification are shown on the right.

Table 7.3.: Ego-vehicle dynamics and situation features corresponding to the optimized ego-vehicle trajectories in Figure 7.2. Again, CB (**C**urrent lane **B**ack), CF (**C**urrent lane **F**ront), LB (**L**eft lane **B**ack), and LF (**L**eft lane **F**ront) correspond to regions of interest around the ego-vehicle at $t = 0\,\mathrm{s}$.

| | $\Delta\dot{L}$ | | $\min(TTC, (\cdot))$ | | | | $\min(THW, (\cdot))$ | | | | Energies $E$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Maneuver in Figure 7.2 | T-0 $[\frac{\mathrm{m}}{\mathrm{s}}]$ | T-des $[\frac{\mathrm{m}}{\mathrm{s}}]$ | CB [s] | CF [s] | LB [s] | LF [s] | CB [s] | CF [s] | LB [s] | LF [s] | OTP [J] | Ego [J] |
| First row | 20.8 | -0.8 | $\infty$ | 12.7 | 9.5 | 5.9 | 2.7 | 2.3 | 1.8 | 2.2 | 3.8 | 213.4 |
| Second row | 9.0 | -10.7 | 7.6 | $\infty$ | - | 74.0 | 2.1 | 2.8 | - | 2.3 | 1.9 | 77.3 |
| Third row | 5.4 | -13.7 | $\infty$ | 9.9 | - | 44.2 | 3.4 | 2.0 | - | 2.0 | 1.1 | 59.3 |

Abbreviations: T-0: Longitudinal velocity difference between ego velocity at beginning and end of lane change, T-des: Longitudinal velocity difference between ego velocity at end of lane change and desired velocity (here: speed limit on target lane), OTP: **O**ther **T**raffic **P**articipants.

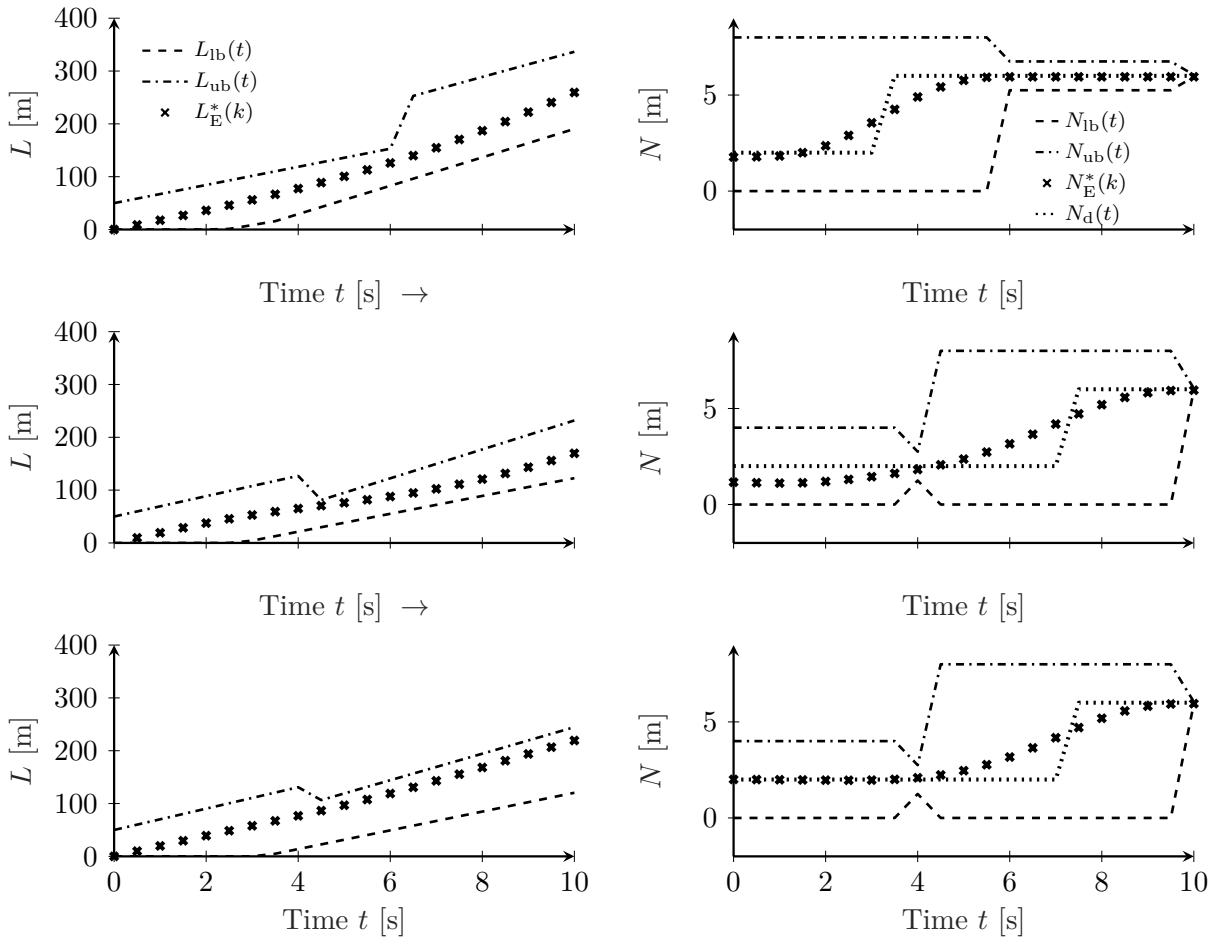



Figure 7.3.: Illustation of the optimized longitudinal and lateral trajectories for all three examples scenes corresponding to Table 7.2 and Figure 7.2.

sensible choice. Figure 7.4 shows the obtained positions and velocity of both approaches. We used a trajectory to represent our approach that corresponds well to the mean values in Table 7.4. It can be seen that the results are comparable concerning ride comfort, especially for the lateral movement.

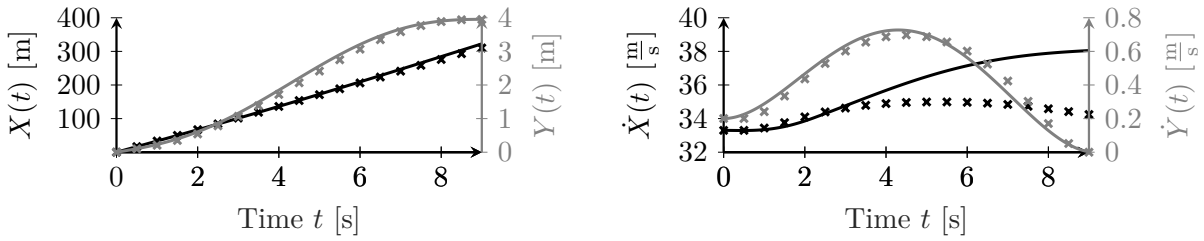
Since k-Means clustering is used, the mean $\mu$ and standard deviation $\sigma$ of the quantities are reported. The key difference is that the sampling-based approach prefers to speed up to minimize interaction cost such that the traffic participant on the target lane has to break less compared to the result of the optimization-based approach. This can be seen by inspection of $E_{\mathrm{OTP}}$ and $\min(TTC_{\mathrm{LB}})$ in Table 7.4. However, the optimization-based approach incorporates hard constraints on Time Headways to surrounding traffic. Trajectories, therefore, maintain safe distances concerning the fixed initial trajectory predictions of the surrounding traffic.

Yet, there is no feedback to measure the influence of a planned trajectory on traffic. In comparison, the sampling-based approach has an advantage since it closes the feedback loop by injecting planned trajectories into the scene prediction to account directly for interactions before choosing the minimum total cost trajectory.

Furthermore, using a sampling approach is advantageous in safety-critical situations when the optimizer cannot provide a solution because of violated hard constraints. In such

Table 7.4.: Comparision of the optimization-based and sampling-based lane change planning approaches to evaluate the non-deterministic k-Means clustering of the sampling approach.

| Approach | $\overline{\lvert\ddot{X}\rvert}$ $[\frac{\mathrm{m}}{\mathrm{s}^2}]$ | $\overline{\lvert\ddot{Y}\rvert}$ $[\frac{\mathrm{m}}{\mathrm{s}^2}]$ | $\overline{\lvert X^{(3)}\rvert}$ $[\frac{\mathrm{m}}{\mathrm{s}^3}]$ | $\overline{\lvert Y^{(3)}\rvert}$ $[\frac{\mathrm{m}}{\mathrm{s}^3}]$ | $\min(TTC_{\mathrm{TB}})$ [s] | $E_{\mathrm{OTP}}$ [J] |
|---|---|---|---|---|---|---|
| Sampling-Based | | | | | | |
| ($\mu$) | 0.44 | 0.14 | 0.18 | 0.09 | 26.76 | 2.06 |
| ($\sigma$) | 0.10 | 0.03 | 0.05 | 0.03 | 4.27 | 1.71 |
| Optimization-Based | 0.26 | 0.12 | 0.18 | 0.10 | 18.02 | 13.80 |




(a) Positions $X$ and $Y$.

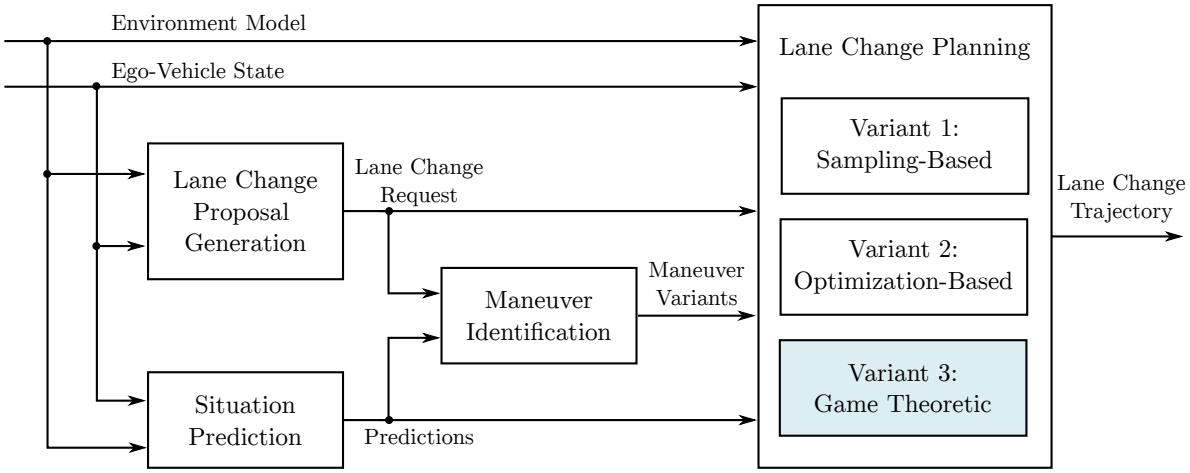
(b) Velocities $\dot{X}$ and $\dot{Y}$.

Figure 7.4.: Comparison of the optimization-based lane change planing approach to the sampling-based variant described in Chapter 6 using $\lambda = 0$. The crosses correspond to the optimization result and the sampling result is shown using solid lines. Black color is used to represent positions and grey for velocities.

situations, using the sampling-based algorithm, one can still choose the trajectory that corresponds to the minimum global cost. This is also why sampling approaches are often used as a redundancy layer in complex motion planning systems.

# 8

# Game Theoretic Lane Change Planning




This Chapter discusses the final lane change planning concept proposed in the thesis at hand. It is an uncertainty-aware approach using a cooperative linear-quadratic (LQ) game formulation. [Sch+19j] inspired the formulation of the lane change planning problem using game theory. There, a non-cooperative game is formulated for interactive traffic scenes with several vehicles involved. A nonlinear kinematic double-track vehicle model is used leading to a nonlinear optimization problem. In the thesis at hand in constrast, a cooperative linear-quadratic game is formulated. This is achieved by using a triple integrator system for the longitudinal and lateral dynamics and focusing on just two agents, the ego-vehicle and the target lane follower vehicle of the lane change. These simplifications lead to a quadratic program that can be solved efficiently and within given runtime constraints. As described in Chapter 4, the situation prediction approach provides the uncertainties of the future positions of all surrounding vehicles. These uncertainties are accounted for using chance-constraints resulting in affine inequalities. An outer optimization is introduced for deriving the first feasible time to start the lane change. The optimization is described in Section 8.1. It also provides an analysis of the influence of the parameters involved in the optimization.

Optimization-based planning algorithms are always at risk of not finding a feasible solution due to the hard safety constraints. Careless drivers surrounding the ego-vehicle or emergency situations can lead to such situations. Assume, for example, a traffic scene where an accident happens directly in front of the ego-vehicle, and its best option is to brake strongly.

Behavior planning algorithms impose hard constraints on the accelerations and will not provide a feasible solution since the constraints aim for comfortable driving. The Section also discusses a fallback solution based on the Intelligent Driver Model (IDM) that always provides a trajectory since it is a collision-free car-following model. An extensive discussion of the architectural implications of integrating a fallback strategy into an automated driving software system is beyond this thesis's scope.

Finally, this Chapter provides an experimental evaluation of the algorithm using the highD dataset and the DESIM simulation environment in Section 8.3. To this end, first, a dataset is extracted from the highD data used to obtain the cost function weights using a Maximum Entropy **I**nverse **R**einforcement **L**earning (IRL) formulation. Furthermore, it is analyzed in terms of observed Time Headways and Time-to-Collisions to set those constraints suitably in the optimization. Next, the approach described in this Chapter is compared to the one described in the previous Chapter 7 in terms of several metrics. Finally, a runtime comparison is provided, showing that the approach can be applied receding horizon in a closed-loop. It should be emphasized that the closed-loop application is not how the lane change behavior planning approach should ideally be used in an automated driving software system. Refer to [Lie+19b] for a more in-depth architectural discussion.

## 8.1. Lane Change Planning using Chance-Constrained Cooperative LQ Games

The theory of cooperative differential Linear-Quadratic (LQ) games is described in [Eng05] for the case of continuous-time. The thesis at hand formulates all optimization problems in discrete time. The modifications of the problem formulation described in Chapter 7 are as follows. Again, the traffic situation shown in Figure 5.4 is used for the following discussion. Compared to optimizing the ego-vehicle trajectory, now the cost function also includes a part for the interacting vehicle. There can be more than one interacting vehicle, but the thesis at hand only focuses on the following vehicle on the target lane of a lane change which is abbreviated with LB (**L**eft lane **B**ack). The target lane following vehicles regularly show courtesy during merging maneuvers, and it is the most suitable choice. Incorporating more interacting vehicles is possible but leads to increased runtime due to the larger resulting quadratic programs. For the following formulation of the optimization problem, it is assumed that for each lane change maneuver $i$, there are lower and upper bounds derived from the spatiotemporal polygons for the longitudinal and lateral movement, determined by the maneuver's pre-, peri- and post-region. For a definition of these terms, please refer to Chapters 5 and 7.

The scalar interaction-aware cost function for the longitudinal trajectory optimization is:

$$J_{\mathrm{I}}^{\mathrm{G}} = \lambda[\theta_1 \sum_{i=1}^{n_p}(\dot{L}_{\mathrm{E},i} - \dot{L}_{\mathrm{E,des}})^2 + \theta_2 \sum_{i=1}^{n_p} \ddot{L}_{\mathrm{E},i}^2 + \theta_3 \sum_{i=1}^{n_p}(L_{\mathrm{E},i}^{(3)})^2] + \tag{8.1.1}$$
$$+ (1-\lambda)[\theta_4 \sum_{i=1}^{n_p}(\dot{L}_{\mathrm{LB},i} - \dot{L}_{\mathrm{LB,des}})^2 + \theta_5 \sum_{i=1}^{n_p} \ddot{L}_{\mathrm{LB},i}^2 + \theta_6 \sum_{i=1}^{n_p}(L_{\mathrm{LB},i}^{(3)})^2].$$

Herein, the ego-vehicle states are subject to the triple integrator system dynamics constraints formulated as forward differences in discrete time with step size $h$:

$$\begin{aligned}
L_{\mathrm{E}}(k+1) &= L_{\mathrm{E}}(k) + \dot{L}_{\mathrm{E}}(k)h + \ddot{L}_{\mathrm{E}}(k)\frac{h^2}{2} + L_{\mathrm{E}}^{(3)}(k)\frac{h^3}{6},\\
\dot{L}_{\mathrm{E}}(k+1) &= \dot{L}_{\mathrm{E}}(k) + \ddot{L}_{\mathrm{E}}(k)h + L_{\mathrm{E}}^{(3)}(k)\frac{h^2}{2},\\
\ddot{L}_{\mathrm{E}}(k+1) &= \ddot{L}_{\mathrm{E}}(k) + L_{\mathrm{E}}^{(3)}(k)h,\\
L_{\mathrm{E}}(0) &= L_{\mathrm{E},0}, \quad \dot{L}_{\mathrm{E}}(0) = \dot{L}_{\mathrm{E},0}, \quad \ddot{L}_{\mathrm{E}}(0) = \ddot{L}_{\mathrm{E},0}.
\end{aligned} \tag{8.1.2}$$

Similarly, the same set of constraints are formulated for the target lane follower vehicle:

$$\begin{aligned}
L_{\mathrm{LB}}(k+1) &= L_{\mathrm{LB}}(k) + \dot{L}_{\mathrm{LB}}(k)h + \ddot{L}_{\mathrm{LB}}(k)\frac{h^2}{2} + L_{\mathrm{LB}}^{(3)}(k)\frac{h^3}{6},\\
\dot{L}_{\mathrm{LB}}(k+1) &= \dot{L}_{\mathrm{LB}}(k) + \ddot{L}_{\mathrm{LB}}(k)h + L_{\mathrm{LB}}^{(3)}(k)\frac{h^2}{2},\\
\ddot{L}_{\mathrm{LB}}(k+1) &= \ddot{L}_{\mathrm{LB}}(k) + L_{\mathrm{LB}}^{(3)}(k)h,\\
L_{\mathrm{LB}}(0) &= L_{\mathrm{LB},0}, \quad \dot{L}_{\mathrm{LB}}(0) = \dot{L}_{\mathrm{LB},0}, \quad \ddot{L}_{\mathrm{LB}}(0) = \ddot{L}_{\mathrm{LB},0}.
\end{aligned} \tag{8.1.3}$$

Exactly as it is done in Chapter 7 for the baseline optimization, further constraints are added to account for minimum $TTC$ and $THW$. Using the upper and lower boundaries $L_{i,\mathrm{ub}}(k)$ and $L_{i,\mathrm{lb}}(k)$, the constraints are:

$$\begin{aligned}
L_{\mathrm{LB}}(k) + \dot{L}_{\mathrm{LB}}(k)TTC_{\min} &\leq L_{\mathrm{E}}(k) + \dot{L}_{\mathrm{E}}(k)TTC_{\min}\\
L_{\mathrm{E}}(k) + \dot{L}_{\mathrm{E}}(k)TTC_{\min} &\leq L_{\mathrm{ub}}(k + n_{TTC_{\min}}),\\
THW_{\min}\dot{L}_{\mathrm{E}}(k) + L_{\mathrm{E}}(k) &\leq L_{\mathrm{ub}}(k),\\
THW_{\min}\dot{L}_{\mathrm{LB}}(k) + L_{\mathrm{LB}}(k) &\leq L_{\mathrm{E}}(k),
\end{aligned} \tag{8.1.4}$$

with the number of samples $n_{THW_{\min}} = \lceil \frac{THW_{\min}}{h} \rceil$ and $n_{TTC_{\min}} = \lceil \frac{TTC_{\min}}{h} \rceil$ corresponding to the minimum $TTC$ and $THW$ times. The $TTC$ and $THW$ constraints are derived from their geometric meaning in spatiotemporal $(L,t)$ domain. Notice, that compared to the constraints in Equation 7.2.4 used for the baseline optimization in Chapter 7, here $L_{\mathrm{LB}}(k)$ and $\dot{L}_{\mathrm{LB}}(k)$ are now part of the decision vector in the optimization and hence the safety constraints are not directly taken from the spatiotemporal polygons but rather formulated explicitly using the quantities of the target lane follower vehicle LB.

**Uncertainty-Awareness using Chance-Constraints**

In the following, it is shown that safety constraints can be formulated as chance-constraints considering uncertainties. [BV04] provides the mathematical foundations for below derivation.Therein, the definition of a chance-constraint starts with an affine constraint

$$\mathbf{a}^{\intercal}\mathbf{x} \leq b, \tag{8.1.5}$$

with vector $\mathbf{a}$, decision vector $\mathbf{x}$ and scalar $b$. Such formulation assumes that all variables are deterministic quantities. In case of chance-constraints, it is assumed that the vector $\mathbf{a}$ is a random variable. This assumptions allows to formulate the constraint satisfaction in terms of a probability (or confidence) $\eta$:

$$\mathrm{P}\left(\mathbf{a}^\intercal \mathbf{x} \leq b\right) \geq \eta. \tag{8.1.6}$$

Next, it will be shown how such constraints can be formulated for the Time-to-Collision safety constraint as an example.
Consider the two possible formulations of the Time-to-Collision constraint of the ego-vehicle (subscript E) to a leading vehicle (subscript L) using positions and velocities:

$$L_{\mathrm{E}}(t_{\mathrm{c}}) + \dot{L}_{\mathrm{E}}(t_{\mathrm{c}})\mathrm{TTC} \leq L_{\mathrm{L}}(t_{\mathrm{c}} + TTC) \tag{8.1.7}$$

which was proposed in Chapter 7 or the slighly more conservative formulation based on the $TTC$ Equation:

$$L_{\mathrm{E}}(t_{\mathrm{c}}) + \dot{L}_{\mathrm{E}}(t_{\mathrm{c}})TTC \leq L_{\mathrm{L}}(t_{\mathrm{c}}) + \dot{L}_{\mathrm{L}}(t_{\mathrm{c}})TTC. \tag{8.1.8}$$

The situation prediction provides gaussian probability distributions specifying the position and velocity of the leading vehicles for all discrete times $t_k$ from $t_0$ up to the planning horizon. Assume that the second variant of the constraint is formulated at timestep $t_c$ and the positions and velocities follow Gaussian distributions:

$$L_{\mathrm{L}}(t_{\mathrm{c}}) \sim \mathcal{N}(\mu_{L_{\mathrm{L}}}(t_{\mathrm{c}}), \sigma^2_{L_{\mathrm{L}}}(t_{\mathrm{c}})) \tag{8.1.9}$$
$$\dot{L}_{\mathrm{L}}(t_{\mathrm{c}}) \sim \mathcal{N}(\mu_{\dot{L}_{\mathrm{L}}}(t_{\mathrm{c}}), \sigma^2_{\dot{L}_{\mathrm{L}}}(t_{\mathrm{c}})) \tag{8.1.10}$$

Instead of the constraint Equation (8.1.8), the corresponding chance-constraints reads:

$$\mathrm{P}(L_{\mathrm{E}}(t_{\mathrm{c}}) + \dot{L}_{\mathrm{E}}(t_{\mathrm{c}})TTC - L_{\mathrm{L}}(t_{\mathrm{c}}) - \dot{L}_{\mathrm{L}}(t_{\mathrm{c}})TTC \leq 0) \geq \eta. \tag{8.1.11}$$

Hence the probability of constraint satisfaction is required to be greater or equal to $\eta$.
In order to formulate it, several substitutions are introduced in the following:

$$a := L_{\mathrm{E}}(t_{\mathrm{c}}) + \dot{L}_{\mathrm{E}}(t_{\mathrm{c}})TTC \tag{8.1.12}$$
$$B := L_{\mathrm{L}}(t_{\mathrm{c}}), \quad B \sim \mathcal{N}(\mu_B, \sigma^2_B) \tag{8.1.13}$$
$$C := \dot{L}_{\mathrm{L}}(t_{\mathrm{c}}), \quad C \sim \mathcal{N}(\mu_C, \sigma^2_C) \tag{8.1.14}$$
$$D := \dot{L}_{\mathrm{L}}(t_{\mathrm{c}})TTC, \quad D \sim \mathcal{N}(TTC\mu_C, TTC^2\sigma^2_C) \tag{8.1.15}$$

Using these substitutions, the chance-constraint is:

$$\mathrm{P}(a - B - D \leq 0) \geq \eta \tag{8.1.16}$$

The goal is to arrive at an affine constraint that can be used in Quadratic Programming and includes only fixed parameters besides the ego-vehicle decision variables ($L_E(t_c)$ and $\dot{L}_E(t_c)$). To this end, some more substitutions are necessary:

$$E := B + D, \quad E \sim \mathcal{N}(\mu_B + TTC\mu_C, \sigma^2_B + TTC^2\sigma^2_B) \tag{8.1.17}$$

$$\mathrm{P}(a - E \leq 0) \geq \eta \tag{8.1.18}$$

$$F := a - E, \quad F \sim \mathcal{N}(a - \mu_B - TTC\mu_C, \sigma_B^2 + TTC^2\sigma_C^2) \tag{8.1.19}$$

$$\mathrm{P}(F \leq 0) = \phi_{\mu_F,\sigma_F}(0) \geq \eta \tag{8.1.20}$$

The last substitution introduces the random variable $G$ that has zero mean and unit variance:

$$G := \frac{F - (a - \mu_B - TTC\mu_C)}{\sqrt{\sigma_B^2 + TTC^2\sigma_C^2}} = \frac{F - \mu_F}{\sigma_F} \tag{8.1.21}$$

$$G \sim \mathcal{N}(0, 1) \tag{8.1.22}$$

This allows to use the tabulated cummulative distribution function or any standard approximation to it:

$$\phi_{\mu_G,\sigma_G}(\xi) = \phi_{0,1}(\xi) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\xi} \exp\left(-\frac{\tilde{\xi}^2}{2}\right) \mathrm{d}\tilde{\xi} \tag{8.1.23}$$

The chance-constraints now reads:

$$\mathrm{P}(F \leq 0) \Leftrightarrow \mathrm{P}\left(\frac{F - \mu_F}{\sigma_F} \leq -\frac{\mu F}{\sigma_F}\right) = \phi_{\mu_G,\sigma_G}\left(-\frac{\mu_F}{\sigma_F}\right) \geq \eta \tag{8.1.24}$$

leading to:

$$\begin{aligned} \phi_{\mu_G,\sigma_G}\left(-\frac{\mu_F}{\sigma_F}\right) \geq \eta &\Leftrightarrow -\frac{\mu_F}{\sigma_F} \geq \phi_{\mu_G,\sigma_G}^{-1}(\eta) \\ &\Leftrightarrow -\frac{1}{\sqrt{\sigma_B^2 + TTC^2\sigma_C^2}} \cdot (a - \mu_B - TTC\mu_C) \geq \phi_{\mu_G,\sigma_G}^{-1}(\eta) \end{aligned} \tag{8.1.25}$$

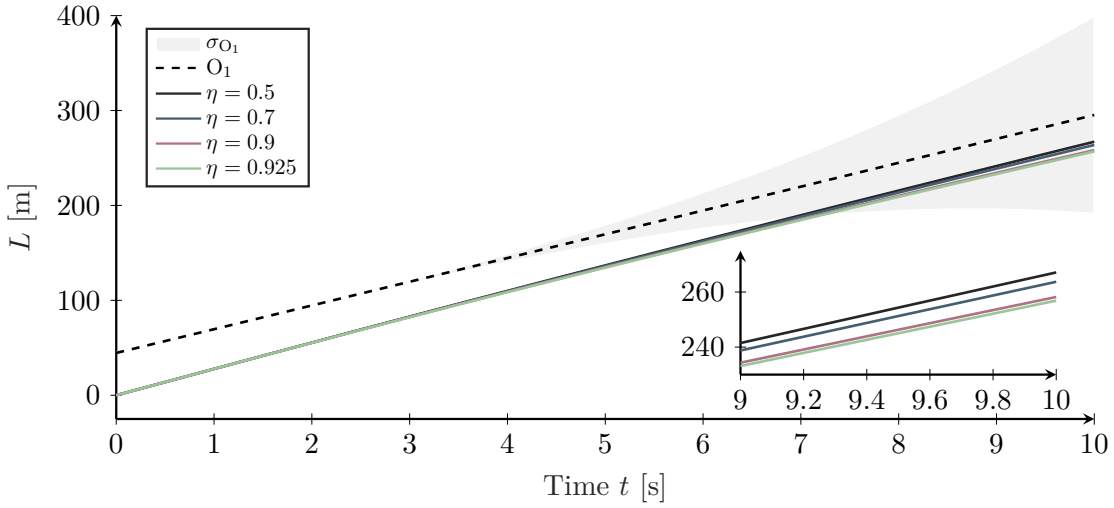
Resubtitution of all variables gives the following constraint that is affine in the decision variables and can hence be used in Quadratic Programming:

$$-\frac{1}{\sqrt{\sigma_{L_L}^2(t_c) + TTC^2\sigma_{\dot{L}_L}^2(t_c)}} \cdot (L_{\mathrm{E}}(t_c) + \dot{L}_{\mathrm{E}}(t_c)TTC - \mu_{L_L}(t_c) - TTC\mu_{\dot{L}_L}(t_c)) \geq \phi_{0,1}^{-1}(\eta) \tag{8.1.26}$$

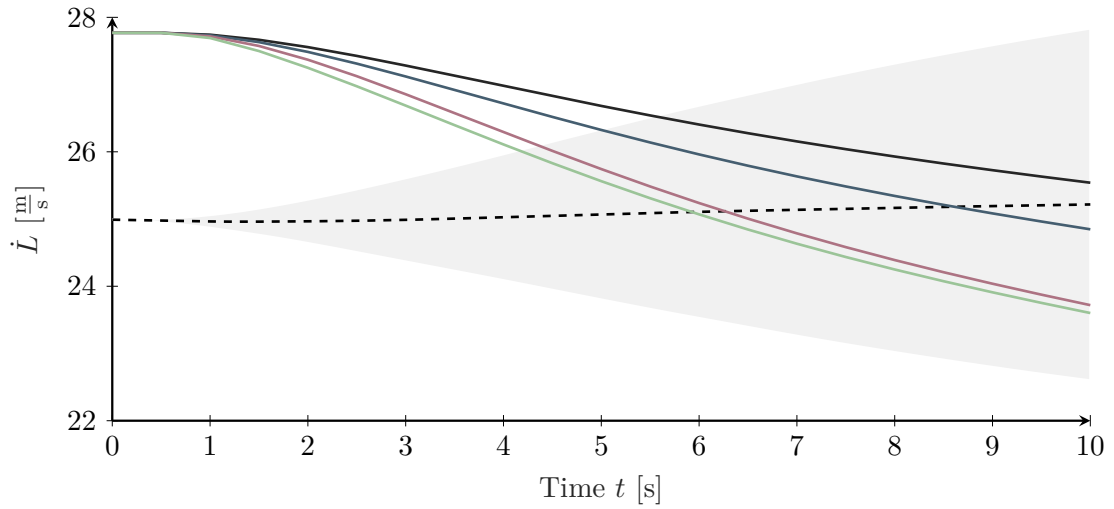
A similar approach can be used to derive the following Time Headway safety constraint:

$$-\frac{1}{\sqrt{\sigma_{L_L}^2(t_c)}} \cdot (L_E(t_c) + \dot{L}_E(t_c)THW - \mu_{L_L}(t_c)) \geq \phi_{0,1}^{-1}(\eta) \tag{8.1.27}$$

The generalization to formulate chance-constraints concerning following vehicles in the back of the ego-vehicle works the same way.
Figure 8.1 shows the trajectory optimization results for a simple lane-keeping maneuver of the ego-vehicle. It approaches a slower vehicle $\mathrm{O}_1$. The increasing standard deviation of the position and velocity are shown in light-grey. The optimization results for a set




(a) Influence of $\eta$ on the optimized longitudinal ego-vehicle trajectory.



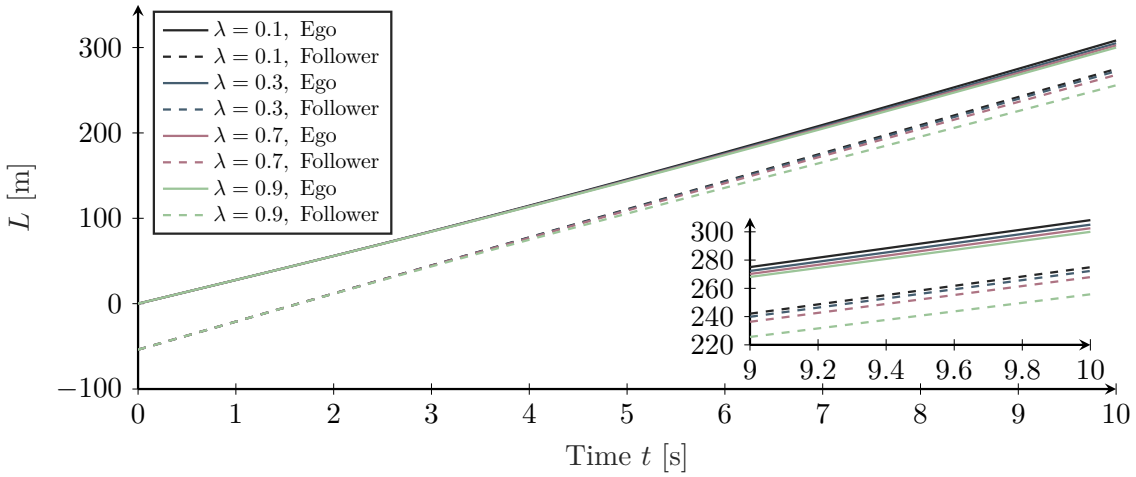

(b) Influence of $\eta$ on the optimized longitudinal velocity.

Figure 8.1.: Influence of $\eta$ on the optimized trajectory of a lane-keeping maneuver with a slower vehicle in front of the ego-vehicle.

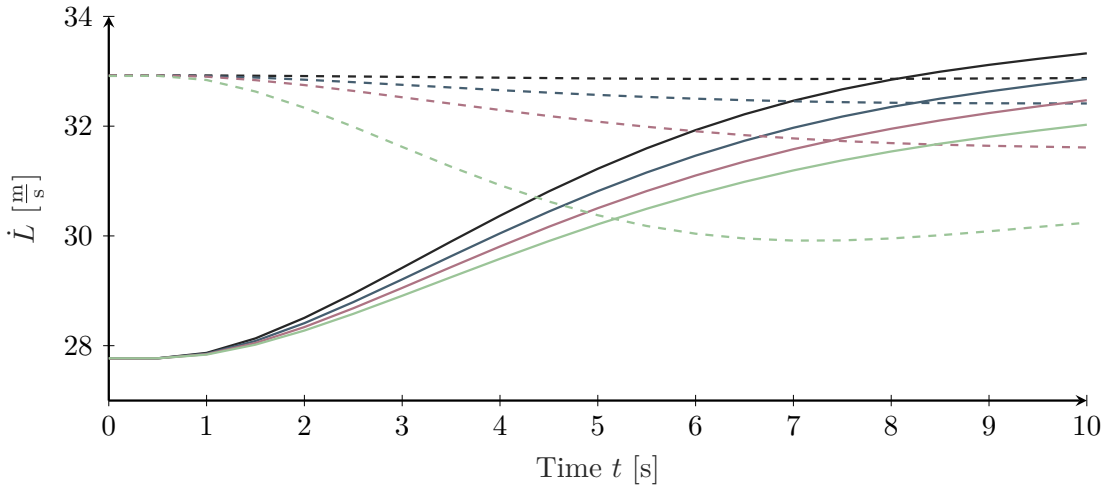
of different values of $\eta$ are shown. Clearly, with higher $\eta$, the chance-constraint becomes more strict and leads to more conservative ego-vehicle behavior as expected. Thus, a careful choice of $\eta$ is necessary for avoiding over-conservative ego-vehicle behavior. Another option is to decrease the variance of the sampling distributions in the Monte Carlo situation prediction approach, leading to decreased uncertainty.

### Influence of the Cooperation Factor $\lambda$ on the Optimization

The influence of $\lambda$ on the trajectory optimization results is next analyzed for a simple two-vehicle traffic scene. The follower vehicle drives on the left lane 50 m behind the ego-vehicle with a velocity of 33.33 m/s. Due to the slower ego-vehicle velocity of 27.7 m/s the amount of velocity increase of the ego-vehicle will depend on the cooperativeness of the follower.




(a) Influence of $\lambda$ on the optimized longitudinal ego-vehicle trajectory.




(b) Influence of $\lambda$ on the optimized longitudinal velocity.

Figure 8.2.: Influence of $\lambda$ on the optimized trajectory of a lane change to the left with a faster follower vehicle on the target lane.

Figure 8.2 shows the results of the position and velocity over the optimization horizon. The dashed lines correspond to the follower quantities, whereas the solid ones reflect the ego-vehicle motion. Notice that a small $\lambda$ such as $\lambda = 0.1$ means that the ego-vehicle quantities are punished much less than the follower with $(1 - \lambda) = 0.9$. Hence to fulfill safety constraints to ensure a safe Time Headway and Time-to-Collision between both vehicles, the ego-vehicle will speed up and accelerate strongly. On the other hand, the follower vehicle only slightly reduces its velocity to show courtesy. This is reflected in the black lines in Figure 8.2. The other shown extreme case is when $\lambda = 0.9$ is chosen and shown in green. Here, the follower vehicle strongly reduces its velocity to give way to the ego-vehicle, which has to accelerate less to ensure safety. Notably, the ego-vehicle reaches a velocity of approximately 32 m/s, but not necessarily to ensure safety. The reason for this behavior is that the desired velocity used in the optimization is set to 33.33 m/s. Notice that it is necessary to scale $\eta$ by the situation probabilities to ensure that unlikely predictions result in less strict constraints.
Next, the value $\lambda = 0.5$ is chosen for the illustration of the influence of varying distances

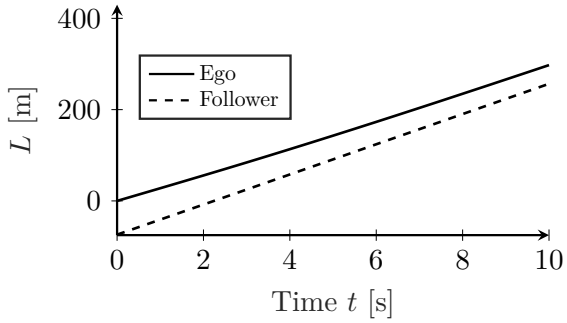
of the follower vehicle to the ego-vehicle. Figures 8.3 (a) and (b) correspond to an initial distance of 70 m between both vehicles. Figures 8.3 (c) and (d) correspond to 50 m and (e) and (f) to 30 m.
8.3 (b) shows that in the first case, the follower vehicle does not need to show courtesy to ensure safety. The ego-vehicle accelerates to approach its desired velocity. On the other hand, 8.3 (d) shows in case of 50 m distance, a certain amount of courtesy is necessary. The most interesting case is when the distance is 30 m. As shown in 8.3 (f), the follower needs to decelerate comparably strong, whereas the ego-vehicle needs to accelerate. Only this way, the safety constraints can be fulfilled. Notice that in this case, optimizing only the ego-vehicle trajectory would lead to an infeasible solver state. It is especially those situations where the cooperative LQ game formulation adds the most value in robustness.

### Fallback Lane-Keeping Solution

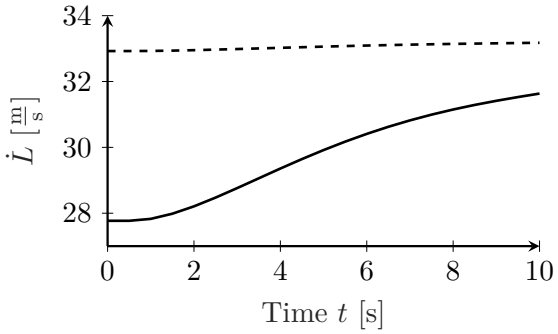
As mentioned in the introduction of this Chapter, the above optimization can lead to situations where no feasible trajectory exists due to constraints. Typically, this happens in emergency situations where the limits of comfortable acceleration or yerks are violated. Chapter 4 introduced the use of the Intelligent Driver Model as the chosen car-following model used in Monte Carlo simulations for situation predictions. The fact that the model is collision-free makes it an appropriate choice for obtaining a fallback lane-keeping solution. To this end, a worst-case situation evolution is derived for the lane-keeping maneuver. This imposes both a upper bound on the position $L_{\text{ub}}(t_k)$ and velocity $\dot{L}_{\text{ub}}(t_k)$ for all timesteps of the optimization horizon $t_{\max}$. With this information, calculating the Intelligent Driver Model (IDM) lane-keeping trajectory can be done assuming double integrator system dynamics. The IDM acceleration $\ddot{L}_{E,IDM}$ of the ego-vehicle acts as the control input of the system.
Figure 8.4 shows the comparison of the chance-constrained optimization approach and the fallback solution for both the position $L$ and velocity $\dot{L}$ in an exemplary traffic scene. The deceleration of the IDM trajectory is more immediate compared to the optimized one. This can be explained since the jerk is punished in the optimization. The IDM trajectory matches the velocity of the preceding vehicle at the end of the horizon. In contrast, the optimization approach leads to a slightly lower velocity, potentially dictated by the change constraint and the uncertainty of the situation prediction.
The above fallback solution is very simple and effective. Another more complex alternative would be introducing slack variables to relax the constraints until a feasible solution can be found.

### Lane Change Initiation Time Determination

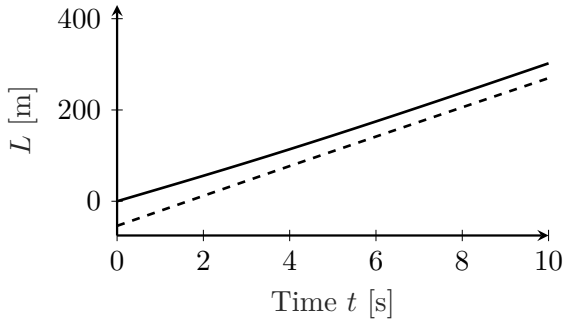
The baseline optimization approach described in Chapter 7 uses a safety-based heuristic to derive the time $T_I$ to initiate the lateral movement. However, such a heuristic might lead to suboptimal results, and this Subsection proposes an alternative. The geometric intuition behind the idea is illustrated in Figure 8.5.
In a first step, the time $t_{\text{I,min}}$ is found. It represents the moment when the reachable set first crosses the boundary of a lane change area shown in dark-blue. By definition, this time is $t_{\text{I,min}} = 0\,\text{s}$ in case of immediate lane changes. Next $t_{\text{I,max}} = 4\,\text{s}$ is defined



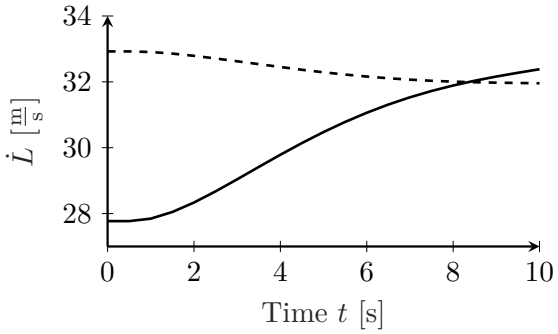

(a) Optimized positions for the case $x_{\mathrm{E}} - x_{\mathrm{LB}} = 70\,\mathrm{m}$.



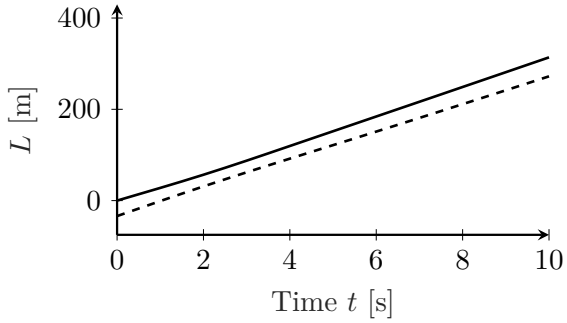

(b) Optimized velocities for the case $x_{\mathrm{E}} - x_{\mathrm{LB}} = 70\,\mathrm{m}$.



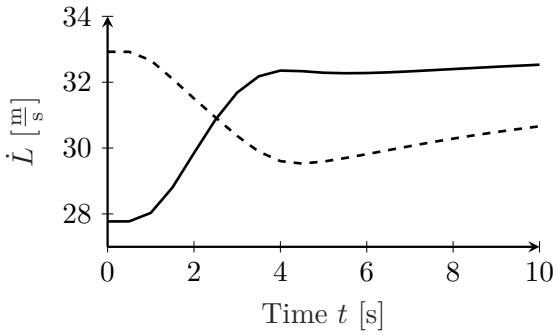

(c) Optimized positions for the case $x_{\mathrm{E}} - x_{\mathrm{LB}} = 50\,\mathrm{m}$.




(d) Optimized velocities for the case $x_{\mathrm{E}} - x_{\mathrm{LB}} = 50\,\mathrm{m}$.

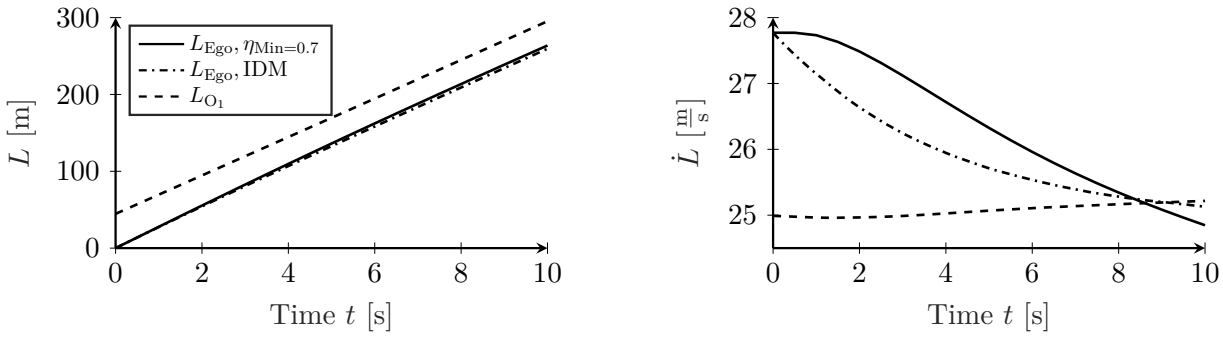



(e) Optimized positions for the case $x_{\mathrm{E}} - x_{\mathrm{LB}} = 30\,\mathrm{m}$.




(f) Optimized velocity for the case $x_{\mathrm{E}} - x_{\mathrm{LB}} = 30\,\mathrm{m}$.

Figure 8.3.: Illustration of the optimization results in a simple lane change left scenario with a faster following vehicle using $\lambda = 0.5$.




(a) Position $L$ of the ego-vehicle and its leader.

(b) Velocity $\dot{L}$ of the ego-vehicle and its leader.

Figure 8.4.: Illustration of the fallback lane-keeping solution using the IDM model and compared to the result obtained using chance-constrained optimization.

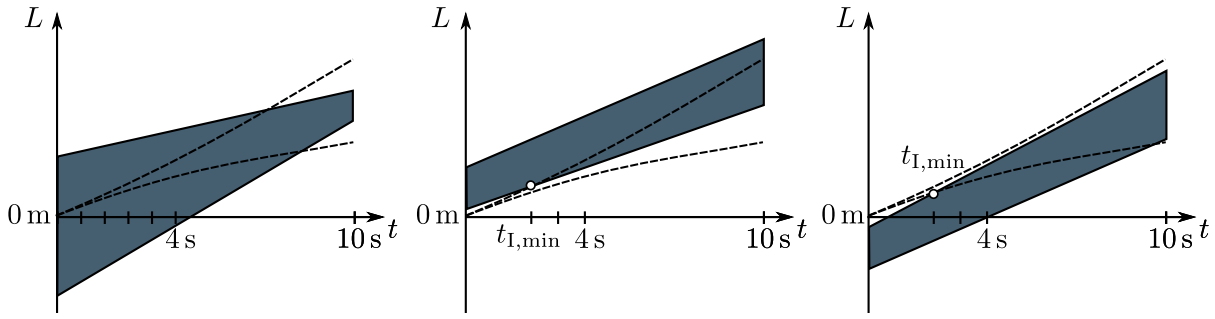



Figure 8.5.: Illustration of the geometric ideas for determining the lane change initiation time. The left Figure shows an immediate lane change. On the other hand, the middle Figure shows a delayed one in a gap in front of the ego-vehicle. The right Figure corresponds to a delayed lane change in a gap behind the ego-vehicle.

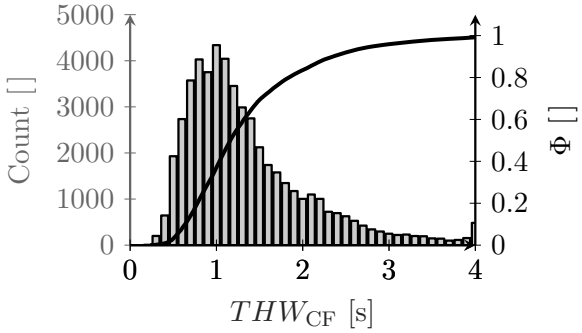
to be the last point in time of valid initiation of the lane change in the current planning cycle. The interval $[t_{\text{I,min}}, t_{\text{I,max}}]$ is next subdivided uniformly using $n_{\text{I}} = 5$ points. This number is motivated since $h = 0.5\,\text{s}$ is used as the step size in any optimization within the thesis at hand. In case of immediate lane changes, this means that the following candidate times $\{0\,\text{s}, 1\,\text{s}, 2\,\text{s}, 3\,\text{s}, 4\,\text{s}\}$ are analyzed. For each of these candidate times, the optimization is done until a feasible solution is found. Note that this choice saves runtime. An alternative would be to conduct all optimizations and compare their resulting cost values to finally choose the solution with the lowest cost. However, the user should rather consider imposing comfortable boundaries to calculate the reachable set such that all feasible solutions are by definition comfortable. Similarly, in case of a delayed lane change (middle and right graphs in Figure 8.5), the interval $[t_{\text{min}}, t_{\text{I,max}}]$ is subdivided uniformly using $n_{\text{I}} = 5$ points and the resulting candidates times are rounded towards the uniform grid with $h = 0.5\,\text{s}$ stepsize and filtered for uniqueness. The above algorithm can be modified to, for example, use a more simple optimization for the determination of the lane change initiation time if runtime needs to be saved. Due to the low runtime of the OSQP solver, such modifications are not made here.

## 8.2. Cost Specification using Inverse Reinforcement Learning

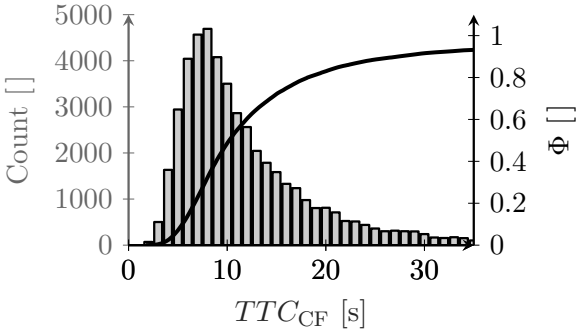
Inverse Reinforcement Learning allows obtaining cost function parameters such that the demonstrated behavior is best explained. It hence removes the need for cumbersome manual tuning of those parameters and therefore saves time. The Maximum Entropy Inverse Reinforcement Learning approach is used in the thesis at hand, refer to [Zie+08]. It models the demonstrated behavior or sequence of actions using a exponential probability distribution and optimizes parameters of a given cost function such that the likelihood of demonstrations or actions is maximized. It does not require an online application in a simulation and is well suited for the optimization of cost function parameters of trajectory and behavior planning approaches in automated driving, refer to [Nau+20].
Later in this Section, the approach is mathematically formulated and the results described.

**Dataset Creation for IRL**

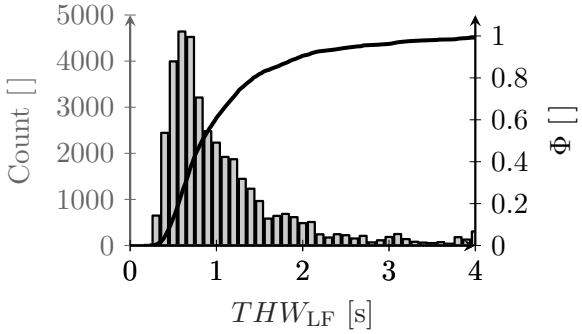
This Subsection describes the creation of a trajectory dataset using the highD data [Kra+18]. In a first step, all traffic scenes are found where a vehicle is changing lane to the left, is visible for at least 6 s and has a follower vehicle during the past 2 s. A total of 906 trajectories fulfill these requirements. Next, only those traffic scenes are chosen where the ego-vehicle has a leading vehicle both on the start and target lane of a lane change such that safety constraints with respect to them need to be fulfilled in the optimization. Finally, a simplification is made to filter out scenes in which vehicles other than the ego-vehicle are changing lanes. This way, a total of 713 trajectories remain.
Those 713 traffic scenes are further analyzed regarding Time Headways and Time-to-Collisions. The histograms and empirical cumulative distribution functions $\Phi$ are shown in Figure 8.6. The subscript indicates the respective vehicle pair. German law treats Time Headways below 1.5 s as a misdemeanor, and below 0.9 s fines are imposed on the driver. It is striking that a huge part of the drivers does not drive rule-consistent. The Time Headway to the leading vehicle at the end of a lane change is frequently very low, as reflected in Figure 8.6 (c). One possible explanation for this behavior is a phenomenon called relaxation, refer to [Sch+12] for more information. Drivers tend to accept smaller headways during merging to apply smaller decelerations and drive more comfortably. Hence a trade-off between comfort and safety is observed in real-world driving on highways.
The analysis results can be used to set the Time Headway and Time-to-Collisions constraints used in lane change behavior planning. Another option is to impose constraints that ensure rule-conformity. Since this Section aims to analyze the performance of lane change behavior planning algorithms on the created dataset, a compromise needs to be found. Hence a Time Headway constraint of 1.0 s and Time-to-Collisions constraint of 6.0 s is used.
Applying this constraint to the dataset leads to just 97 trajectories. Those will be used for the Maximum Entropy Inverse Reinforcement Learning to obtain the weights of all cost functions involved in the various optimizations. Relaxing the constraints to include scenes with minimum Time-to-Collisions of 4.5 s and Time Headways of 0.5 s leads to a total of 423 scenes. Those will be used later to evaluate the robustness gain of formulating



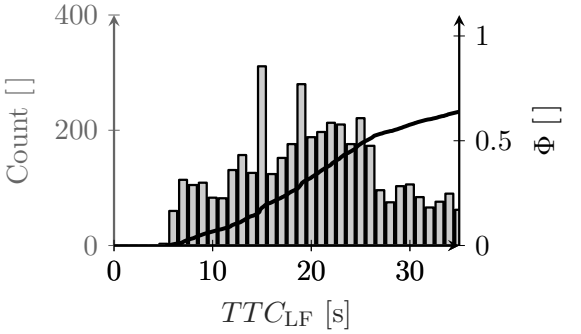

(a) Time Headway distribution of ego-vehicle to its current lane leader vehicle.



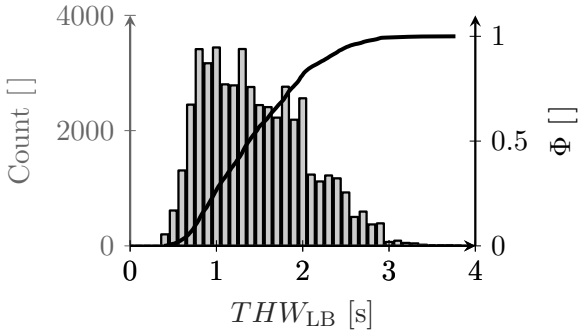

(b) Time-to-Collision distribution of ego-vehicle to its current lane leader vehicle.



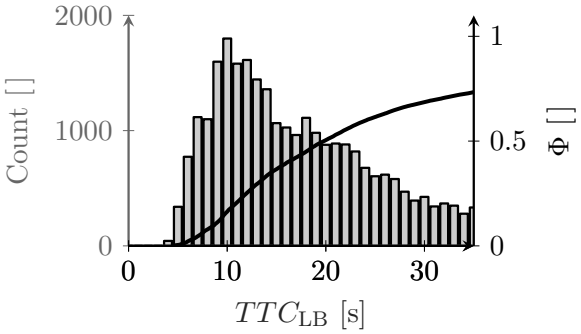

(c) Time Headway distribution of ego-vehicle to its left lane leader vehicle.




(d) Time-to-Collision distribution of ego-vehicle to its left lane leader vehicle.




(e) Time Headway distribution of ego-vehicle to its left lane follower vehicle.




(f) Time-to-Collision distribution of ego-vehicle to its left lane follower vehicle.

Figure 8.6.: Time Headway and Time-to-Collision statistics of the created example trajectory subset based on the highD dataset. The distribution of data is shown using histograms and the corresponding cummulative distributions functions.

Table 8.1.: Datasets created based on the highD data used for Inverse Reinforcement Learning and subsequent evaluations of its results.

| | Number of trajectories |
|---|---|
| Initial filtered dataset | 906 |
| All ROI vehicles without doing lane changes | 713 |
| $TTC \leq 4.5\,\mathrm{s} \wedge THW \leq 0.5\,\mathrm{s}$ | 423 |
| $TTC \leq 6.0\,\mathrm{s} \wedge THW \leq 1.0\,\mathrm{s}$ | 97 |

the optimization as a cooperative LQ game. All information is summarized in Table 8.1.

**Inverse Reinforcement Learning Results and Evaluation on highD dataset**

This Subsection describes the Maximum Entropy Inverse Reinforcement Learning approach applied to the following three cost functions:

$$\begin{aligned} J_{\mathrm{I}}^{\mathrm{G}} =& \lambda[\theta_1 \sum_{i=1}^{n_p}(\dot{L}_{\mathrm{E},i} - \dot{L}_{\mathrm{E,des}})^2 + \theta_2 \sum_{i=1}^{n_p} \ddot{L}_{\mathrm{E},i}^2 + \theta_3 \sum_{i=1}^{n_p}(L_{\mathrm{E},i}^{(3)})^2] + \\ &+ (1-\lambda)[\theta_4 \sum_{i=1}^{n_p}(\dot{L}_{\mathrm{LB},i} - \dot{L}_{\mathrm{LB,des}})^2 + \theta_5 \sum_{i=1}^{n_p} \ddot{L}_{\mathrm{LB},i}^2 + \theta_6 \sum_{i=1}^{n_p}(L_{\mathrm{LB},i}^{(3)})^2], \end{aligned} \tag{8.2.1}$$

$$J_{\mathrm{L}}^{\mathrm{O}} = \alpha_1 \sum_{i=1}^{n_p}(\dot{L}_{\mathrm{E},i} - \dot{L}_{\mathrm{E,des}})^2 + \alpha_2 \sum_{i=1}^{n_p} \ddot{L}_{\mathrm{E},i}^2 + \alpha_3 \sum_{i=1}^{n_p}(L_{\mathrm{E},i}^{(3)})^2, \tag{8.2.2}$$

$$J_{\mathrm{N}}^{\mathrm{O}} = \beta_1 \sum_{i=1}^{n_p}(N_{\mathrm{E},i} - N_{\mathrm{E,des}})^2 + \beta_2 \sum_{i=1}^{n_p} \ddot{N}_{\mathrm{E},i}^2 + \beta_3 \sum_{i=1}^{n_p}(N_{\mathrm{E},i}^{(3)})^2. \tag{8.2.3}$$

Cost function $J_{\mathrm{I}}^{\mathrm{G}}$ reflects the one used in the cooperative LQ game formulation, $J_{\mathrm{L}}^{\mathrm{O}}$ is used for optimizing just the ego-vehicle longitudinal trajectory, and $J_{\mathrm{N}}^{\mathrm{O}}$ reflects the lateral optimization for both former cases. Notice, that in above lateral cost formulation the lateral velocity is omitted compared to Equation 7.2.5.
The Maximum Entropy formulation of [LK12] or more specifically [Nau+20] is used. Assuming a set of demonstration $\mathcal{D} = \{\tau_{D,j}\}_{j=1}^{n_D}$, the Maximum Entropy formulations aims to maximize the feature likelihood. Practically making trajectories with lower cost more likely outcomes of the optimization. For some general parameter vector $\Theta$ and cost function depending on those parameters $J\tau, \Theta$, the following problem is solved:

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} -P(\mathcal{D}|\Theta) \tag{8.2.4}$$

$$= \underset{\Theta}{\operatorname{argmin}} -\sum_{j=1}^{n_D} \exp(-J(\tau_{D,j}, \Theta)) \left[\int \exp(J(\tilde{\tau}_{D,j}, \Theta)) \mathrm{d}\tilde{\tau}_{D,j}\right]^{-1}. \tag{8.2.5}$$

Assuming a Maximum Entropy probability distribution leads to:

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \sum_{j=1}^{n_D} \frac{1}{2} \mathbf{g}_\Theta^*(\tau_{D,j}) \mathbf{H}_\Theta^{-1}(\tau_{D,j}) \mathbf{g}_\Theta(\tau_{D,j}) - \frac{1}{2} \log(\det(\mathbf{H}_\Theta(\tau_{D,j}))) + \frac{d_\tau}{2} \log(2\pi) \tag{8.2.6}$$

Table 8.2.: Resulting optimized weights for each respective cost function.

| Interactive ($J_{\mathrm{I}}^{\mathrm{G}}$) | Ego-Only ($J_{\mathrm{L}}^{\mathrm{O}}$) | Lateral ($J_{\mathrm{N}}^{\mathrm{O}}$) |
|---|---|---|
| $\theta_1 = 0.21$ | $\alpha_1 = 0.10$ | $\beta_1 = 0.13$ |
| $\theta_2 = 6.00$ | $\alpha_2 = 2.82$ | $\beta_2 = 13.47$ |
| $\theta_3 = 38.34$ | $\alpha_3 = 17.99$ | $\beta_3 = 27.31$ |
| $\theta_4 = 0.12$ | | |
| $\theta_5 = 5.80$ | | |
| $\theta_6 = 40.21$ | | |
| $\lambda = 0.47$ | | |

Therein $\mathbf{g}_\Theta$ and $\mathbf{H}_\Theta$ are the gradient and Hessian of the cost function with respect to its parameters $\Theta$. $d_\tau$ is the dimension of a demonstrated trajectory $\tau$. A detailed derivation is found in [Zie+08] and [LK12]. Problem (8.2.6) is formulated using CasADi [And+19] and its automatic differentiation functionality. The resulting problem is then solved with the IPOPT (**I**nterior **P**oint **OPT**imizer) solver [WB06].
The optimized parameters of the three cost functions are shown in Table 8.2. Notice that $\lambda = 0.47$ was found, indicating that the demonstrations are best explained using almost the most cooperative cost-sharing between the ego-vehicle and left following vehicle. The parameters of $J_{\mathrm{E}}^{\mathrm{G}}$ are approximately half that of the ones of $J_{\mathrm{I}}^{\mathrm{G}}$ which is reasonable due to $\lambda = 0.47$. A comparably low cost is assigned to reaching the desired velocity. However, the cost is quadratic, and there are significant deviations from the desired velocity in the demonstrations. A comparably high parameter is assigned to the jerk part of the cost function. These results match the discussion of [Wer+10] well, that indicates that human drivers tend to drive jerk optimal. Similarly, the parameter regarding the lateral deviation is comparably low, again due to the high lateral deviation values observed during a lane change. In the following Section 8.3, the resulting cost functions with parameters shown in 8.2 are evaluated using a set of demonstration trajectories of the highD dataset. This way, a comparison of both the interactive and baseline optimization are assessed in terms of how good the optimized trajectories match the observed ones of human drivers.

## 8.3. Evaluation of the Game Theoretic Lane Change Planning

Those optimized cost function parameters are used in the following. The first evaluation analyzes the success rate of using the cooperative LQ game formulation compared to optimizing only the ego-vehicle longitudinal trajectory with identical safety constraints. The data with minimum Time-to-Collisions of 4.5 s and Time Headways of 0.5 s consisting of 423 demonstrations is used, refer to Table 8.3. A total of four configurations, C1 to C4, are analyzed. The distinguishing factors are whether chance-constraints are imposed and the ego-vehicle is forced to change lanes in the situation prediction. Therefore the column *Force Ego-LC* means that in the Monte Carlo Situation Prediction approach described in Chapter 4, in each simulation run, the ego-vehicle is forced to change lane at

Table 8.3.: Success rates on the highD trajectory dataset consisting of a total of 423 examples.

| Configuration | Force Ego-LC | Chance-Const. ($\eta = 0.85$) | Success Rates | |
|---|---|---|---|---|
| | | | Interactive | Ego-Only |
| C1 | No | No | 300/423(70.9%) | 206/423(48.7%) |
| C2 | No | Yes | 274/423(64.8%) | 169/423(40.0%) |
| C3 | Yes | No | 301/423(71.2%) | 212/423(50.1%) |
| C4 | Yes | Yes | 276/423(65.2%) | 188/423(44.4%) |

the earliest possible and safe time. Forcing a lane change leads potentially to a breaking maneuver of the left following vehicle and could increase the success rate. The column Chance-Const. ($\eta = 0.85$) distinguishes if in the respective configuration the Time Headway and Time-to-Collision safety constraints are formulated in a probabilistic manner or assuming the predicted trajectories of surrounding vehicles are deterministic. Imposing chance-constraints leads to more conservative safety constraints.
The results are shown in Table 8.3. Imposing the chance-constraints during the default lane change duration of $t_{\mathrm{LC}} = 6\,\mathrm{s}$ does alter the success rate when using $\eta = 0.85$ due to accounting for uncertainties. Forcing an ego-vehicle lane change during situation prediction has only a slight effect on the results. Still, it depends heavily on the parametrization of the driver models during the Monte Carlo simulations. This aspect is not analyzed in-depth within the thesis at hand. The results indicate that the cooperative LQ game formulation leads to a significantly increased success rate, meaning that feasible solutions are successfully calculated.
Another important evaluation considers how well each approach reconstructs the driven trajectories of the ego-vehicle. The hypothesis is that the target lane following vehicle will show courtesy which is reflected in the parameter $\lambda$ leading to better results for the cooperative LQ game formulation referred to as interactive in Table 8.4. It shows a total of eight metrics. To ensure a fair comparison, only examples are used that lead to feasible solutions for both optimization variants. The first three consider the absolute deviation of the optimized longitudinal position, velocity, acceleration, and jerk to the driven trajectories in the ego-vehicle coordinate frame for several timesteps. The last three consider the trajectory as a whole and use the DTW (**D**ynamic **T**ime **W**arping) metric, refer for example to [BC94]. It is a typical metric used for the comparison of two time-series with frequent uses in automated driving, see [Hu+19c] for an example of its use in trajectory prediction. Assuming two time-series $\mathbf{a} = [a_1 \quad a_2 \quad \dots a_{\mathrm{n}}]^\intercal$ and $\mathbf{b} = [b_1 \quad b_2 \quad \dots b_{\mathrm{m}}]^\intercal$ represented as vectors. The series are first arranged as a n-by-m grid. Each point $(i, j)$ in the grid represents a alignment of points $a_i$ and $b_j$. A path in the grid represents a warping between both series such that the distance between both is minimized. The problem can be formulated as a recursive optimization problem under a certain set of constraints. This way, similar time-series with for example just slight phase shifts lead to small DTW distances whereas a simple euclidean distance calculation might lead to significantly higher values.
The results indicate that forcing ego-vehicle lane changes within the situation prediction

leads to worse results. Similarly, imposing chance-constraints leads to safer driving but worse results for imitating human driving behavior, at least on the dataset considered here. Most importantly, the cooperative LQ formulation outperforms the ego-only optimization in all metrics when comparing identical configurations. Therefore, the novel formulation not only increases the success rates and robustness but also leads to optimized trajectories closer to the driven ones. The runtime analysis shows that this comes at the cost of an increased runtime due to larger matrices involved in the Quadratic Program.

Some example reference trajectories (denoted as Ref) and optimization results (denoted as Opt) are shown in Figure 8.7 (a) - (d) with their respective DTW metrics. Solid lines represent ego-vehicle velocity trajectories (denoted as Ego), whereas dashed lines represent the left lane following vehicle (denoted as LB). Its predicted trajectory is referred to as LB-Pred. Example (a) shows an example where the cooperative LQ game formulation (denoted as Int-Opt) leads to a velocity trajectory almost identical to the driven one. It is also shown that the left following vehicle shows courtesy, which also results from the optimization. In contrast, if only the ego-vehicle trajectory is optimized (denoted as Ego-Only-Opt) based on the initial trajectory prediction, stronger acceleration is involved to ensure safety. In example (b), both approaches lead to similar results mainly because the velocities of ego-vehicle and left following vehicle are similar and safety is hence easier to ensure without requiring courtesy. In example (c), the ego-only optimization leads to better results. Notice here how the initial situation prediction is closer to the driven trajectory of the left following vehicle compared to its optimized trajectory. The final example (d) again shows an example where the optimization of both vehicle's trajectories leads to superior results. Again, the initial situation prediction does not reflect the driven trajectory of the left following vehicle well, and the optimization accounts for this fact.

**Runtime Evaluation in the ZF Software Framework**

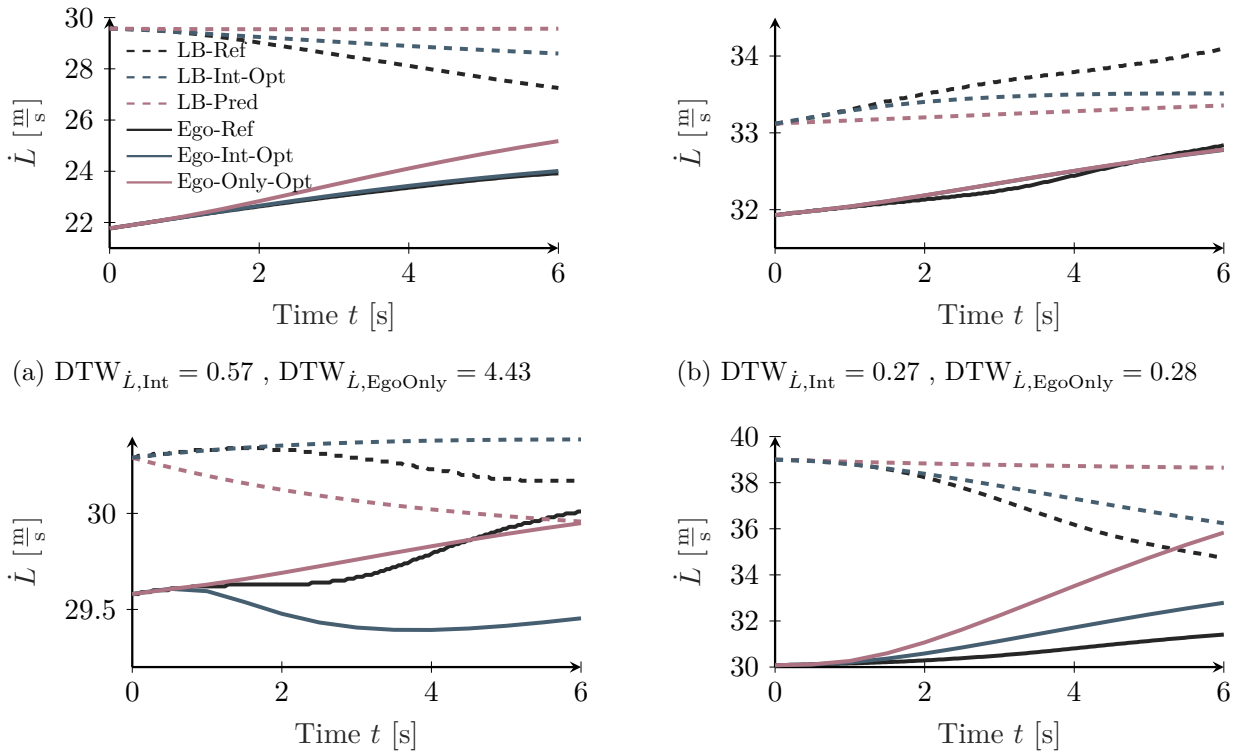
Both the approach of the previous Chapter 7 (called Ego-Only) and the cooperative LQ game variant are integrated into the ZF software framework using the C programming language. The algorithms can be configured to use Polygon Clipping or avoid it by assuming lane-keeping surrounding vehicles. Table 8.5 provides an overview of the achieved runtimes averaged over a total of 1538 cycles of the algorithm. Both the mean and standard deviation of the runtimes are provided.

Inspection of the Table shows that the Polygon Clipping application takes more time compared to all involved optimizations. [Mar+09] is a much more efficient algorithm compared to [Vat92] that was used here. [Mar+09] should therefore be the algorithm of choice to reduce the runtime further. The maximum number of lane change initiation time optimizations for immediate and delayed lane changes are set to 5 and reflected in the $n_{\mathrm{I}}$.

Comparing the last two rows of the Table shows that the runtime of the cooperative LQ game formulation is triple that of the ego-only formulation. Part of the reason is that no simplified dynamics are used for the lane change initiation time optimization. But the higher runtime is mainly due to the larger matrices involved since the trajectories of two vehicles are optimized simultaneously. Hence runtime is traded-off against robustness, and the system designer decides which approach is best suited for their application.

Table 8.4.: Results reflecting the reconstruction capability of both the interactive and ego-only optimization variant using the highD dataset.

| | | Interactive | | | | Ego-Only | | | |
|---|---|---|---|---|---|---|---|---|---|
| Metric | Time [s] | C1 | C2 | C3 | C4 | C1 | C2 | C3 | C4 |
| $\overline{\lvert L - L_{\text{ref}} \rvert}$ [m] | 1 | **0.03 ± 0.03** | **0.03 ± 0.03** | 0.04 ± 0.03 | 0.04 ± 0.03 | **0.03 ± 0.03** | 0.04 ± 0.03 | 0.04 ± 0.03 | 0.04 ± 0.03 |
| | 2 | **0.18 ± 0.21** | **0.18 ± 0.22** | 0.19 ± 0.22 | 0.20 ± 0.23 | 0.20 ± 0.21 | 0.26 ± 0.25 | 0.20 ± 0.21 | 0.26 ± 0.26 |
| | 4 | **1.35 ± 1.71** | 1.42 ± 1.71 | 1.50 ± 1.81 | 1.63 ± 1.88 | 1.63 ± 1.78 | 1.99 ± 2.00 | 1.68 ± 1.86 | 2.15 ± 2.14 |
| | 6 | **3.31 ± 4.01** | 3.46 ± 4.04 | 3.65 ± 4.23 | 3.94 ± 4.42 | 4.18 ± 4.56 | 4.92 ± 4.91 | 4.31 ± 4.81 | 5.32 ± 5.29 |
| $\overline{\lvert \dot{L} - \dot{L}_{\text{ref}} \rvert}$ [$\frac{\text{m}}{\text{s}}$] | 1 | **0.05 ± 0.06** | **0.05 ± 0.06** | **0.05 ± 0.06** | **0.05 ± 0.06** | **0.05 ± 0.05** | 0.07 ± 0.06 | **0.05 ± 0.05** | 0.07 ± 0.07 |
| | 2 | **0.30 ± 0.39** | 0.32 ± 0.39 | 0.33 ± 0.41 | 0.36 ± 0.43 | 0.35 ± 0.38 | 0.44 ± 0.45 | 0.36 ± 0.40 | 0.47 ± 0.47 |
| | 4 | **0.85 ± 1.02** | 0.89 ± 1.03 | 0.94 ± 1.09 | 1.02 ± 1.14 | 1.07 ± 1.16 | 1.27 ± 1.26 | 1.11 ± 1.23 | 1.38 ± 1.37 |
| | 6 | **1.11 ± 1.25** | 1.16 ± 1.27 | 1.19 ± 1.30 | 1.27 ± 1.36 | 1.48 ± 1.66 | 1.68 ± 1.75 | 1.50 ± 1.71 | 1.77 ± 1.80 |
| $\overline{\lvert \ddot{L} - \ddot{L}_{\text{ref}} \rvert}$ [$\frac{\text{m}}{\text{s}^2}$] | 1 | **0.17 ± 0.23** | 0.18 ± 0.23 | 0.19 ± 0.24 | 0.21 ± 0.25 | 0.20 ± 0.21 | 0.25 ± 0.26 | 0.20 ± 0.22 | 0.26 ± 0.26 |
| | 2 | **0.32 ± 0.40** | 0.33 ± 0.40 | 0.35 ± 0.42 | 0.38 ± 0.44 | 0.38 ± 0.41 | 0.46 ± 0.47 | 0.39 ± 0.43 | 0.50 ± 0.50 |
| | 4 | **0.22 ± 0.22** | 0.23 ± 0.23 | 0.24 ± 0.24 | 0.26 ± 0.26 | 0.32 ± 0.38 | 0.36 ± 0.42 | 0.33 ± 0.39 | 0.38 ± 0.42 |
| | 6 | **0.18 ± 0.16** | 0.19 ± 0.17 | **0.18 ± 0.16** | 0.20 ± 0.17 | 0.26 ± 0.26 | 0.30 ± 0.32 | 0.24 ± 0.25 | 0.27 ± 0.30 |
| $\overline{\lvert L^{(3)} - L^{(3)}_{\text{ref}} \rvert}$ [$\frac{\text{m}}{\text{s}^3}$] | 1 | **0.23 ± 0.28** | 0.24 ± 0.28 | 0.25 ± 0.30 | 0.27 ± 0.32 | 0.27 ± 0.29 | 0.33 ± 0.35 | 0.27 ± 0.30 | 0.35 ± 0.36 |
| | 2 | **0.10 ± 0.10** | **0.10 ± 0.09** | **0.10 ± 0.10** | 0.11 ± 0.11 | 0.13 ± 0.18 | 0.17 ± 0.23 | 0.13 ± 0.15 | 0.15 ± 0.21 |
| | 4 | **0.14 ± 0.15** | **0.14 ± 0.15** | 0.16 ± 0.17 | 0.16 ± 0.19 | 0.16 ± 0.18 | 0.19 ± 0.23 | 0.16 ± 0.19 | 0.21 ± 0.25 |
| | 6 | **0.09 ± 0.09** | **0.09 ± 0.09** | 0.10 ± 0.11 | 0.11 ± 0.11 | 0.11 ± 0.11 | 0.12 ± 0.14 | 0.11 ± 0.12 | 0.12 ± 0.13 |
| $\text{DTW}_L$ [m] | All | **12.32 ± 13.05** | 13.01 ± 13.60 | 13.65 ± 14.10 | 14.75 ± 14.68 | 15.83 ± 15.31 | 18.45 ± 16.46 | 16.23 ± 16.20 | 19.56 ± 17.03 |
| $\text{DTW}_{\dot{L}}$ [$\frac{\text{m}}{\text{s}}$] | All | **6.32 ± 8.61** | 6.68 ± 8.75 | 6.93 ± 9.10 | 7.58 ± 9.53 | 7.89 ± 9.61 | 9.52 ± 10.52 | 8.27 ± 10.20 | 10.40 ± 11.35 |
| $\text{DTW}_{\ddot{L}}$ [$\frac{\text{m}}{\text{s}^2}$] | All | **2.08 ± 2.02** | 2.17 ± 2.07 | 2.21 ± 2.07 | 2.37 ± 2.22 | 2.93 ± 3.17 | 3.48 ± 3.49 | 2.85 ± 3.13 | 3.46 ± 3.46 |
| $\text{DTW}_{L^{(3)}}$ [$\frac{\text{m}}{\text{s}^3}$] | All | **1.28 ± 1.34** | 1.39 ± 1.30 | 1.47 ± 1.41 | 1.56± 1.52 | 1.58 ± 1.55 | 2.03 ± 2.03 | 1.60 ± 1.52 | 2.06 ± 2.03 |




(a) $\mathrm{DTW}_{\dot{L},\mathrm{Int}} = 0.57$ , $\mathrm{DTW}_{\dot{L},\mathrm{EgoOnly}} = 4.43$

(b) $\mathrm{DTW}_{\dot{L},\mathrm{Int}} = 0.27$ , $\mathrm{DTW}_{\dot{L},\mathrm{EgoOnly}} = 0.28$

(c) $\mathrm{DTW}_{\dot{L},\mathrm{Int}} = 3.52$ , $\mathrm{DTW}_{\dot{L},\mathrm{EgoOnly}} = 0.23$

(d) $\mathrm{DTW}_{\dot{L},\mathrm{Int}} = 4.90$ , $\mathrm{DTW}_{\dot{L},\mathrm{EgoOnly}} = 19.88$

Figure 8.7.: Several example reference trajectories of the ego-vehicle and its following vehicle on the left lane. The optimization results are shown and the captions of each Figure shows the DTW results for the interactive (Int) and baseline (Ego-Only) optimizations.

Note that all approaches result in low runtimes, allowing for applying the algorithms on embedded hardware platforms. This is in stark contrast to many literature approaches that require more accurate satisfaction of non-holonomic and system dynamics constraints leading to nonlinear programs that are harder to solve and hence take more time. It is argued within the thesis at hand that a Quadratic Programming approach using simplified dynamics is a good choice for lane change behavior planning for application in today's software systems without the need for parallel computing hardware.

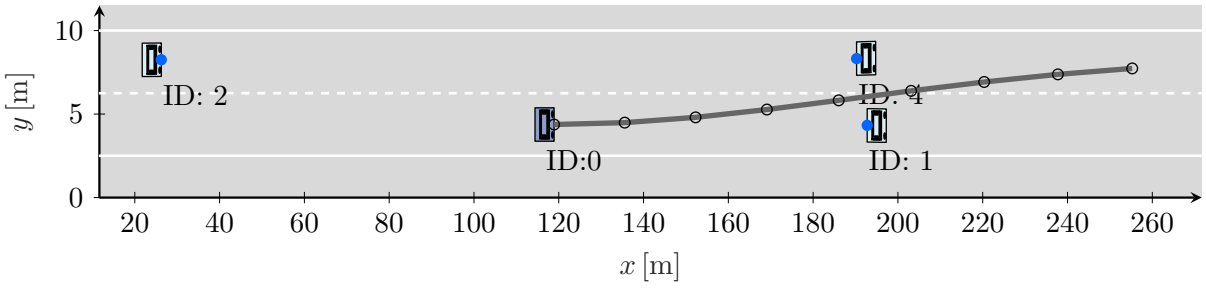
### Qualitative Results of the Closed-Loop Application in Simulation

Note that the term lane change behavior planning was defined in Chapter 2, more specif-

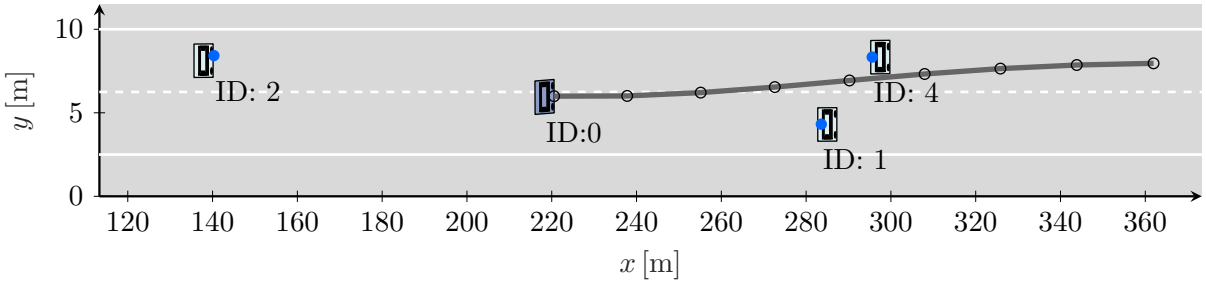
Table 8.5.: Runtimes of several configurations of the optimization-based lane change planning algorithms described in the thesis at hand. They are measured on a standard PC (i5-6500, 16GB RAM) and realized using the C programming language.

| Type | Polygon Clipping | $n_\mathrm{I}$ | Avg. runtime [ms] | Cycles |
|---|---|---|---|---|
| None | Yes | - | 10.60± 6.70 | 1538 |
| Ego-Only | Yes | 5 | 18.80± 7.00 | 1538 |
| Ego-Only | No | 5 | 5.80± 7.20 | 1538 |
| Interactive | No | 5 | 15.30±13.70 | 1538 |

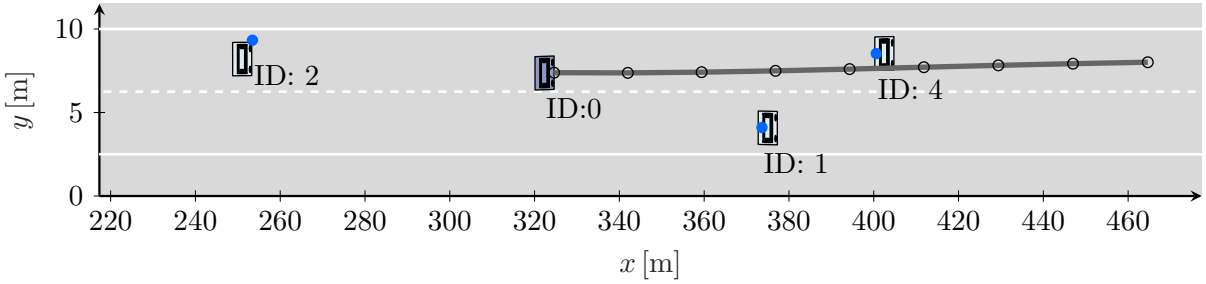
ically in Section 2.1. From that discussion, it is obvious that the goal of a lane change behavior planning algorithm is not its closed-loop application. It rather just serves as a globalization strategy for any suitable local trajectory planning approach. Ideally, the local planner should make fewer vehicle dynamics simplifications and consider the non-holonomic constraints accurately. The above box constraint formulation is just an underapproximation of Kamm's Circle. Nevertheless, the closed-loop application of the cooperative LQ approach with chance-constraints is discussed. The DESIM simulation environment is used for the evaluation. A simple, pure pursuit trajectory tracking approach with 1 s look-ahead time is used. Note that the following only presents a qualitative evaluation. A quantitative evaluation needs to analyze the interplay of all algorithms in more detail. The work [Lie+19b] was the first step in that direction.
Figure 8.8 shows the initial traffic scene and its temporal evolution. The optimized ego-vehicle trajectory is shown in grey, and nine dots are placed uniformly in time along the trajectory. A planning horizon of $t_{\max} = 10\,\mathrm{s}$ is used. The ego-vehicle merges into the gap between the vehicles with ID2 and ID4 to overtake the slower vehicle with ID1.
A more detailed look at the driven trajectory of the ego-vehicle is shown in Figure 8.9. First, Figure 8.9 (a) shows the driven trajectories of all vehicles in the initial ego-vehicle coordinate at time $t = 0\,\mathrm{s}$. Figure 8.9 (c) and (e) show the velocities and acceleration. The ego-vehicle finally drives with the same velocity as its new leading vehicle on the target lane and keeps a safe distance to it as specified by the Time Headway and Time-to-Collision constraints. The lateral position trajectory is shown in Figure 8.9 (b), whereas 8.9 (d) and (f) show the lateral velocity and acceleration respectively. Note that the ego-vehicle reaches the target lane and aligns itself correctly to it.
One adaption necessary for the lane change behavior planning approach to work in a closed-loop is adapting the optimization constraints according to the lateral progress towards the target lane center. For example, at the start of the lane change, the safety constraints regarding the start lane leading vehicle need to be imposed at least for three seconds, assuming a default lane change duration of $t_{\mathrm{LC}} = 6\,\mathrm{s}$. However, if the ego-vehicle already approached the target lane, a constraint adaption is necessary. Therefore, the constraints shift in time backward with further lateral progress towards the target lane, similar to a shrinking horizon mechanism.



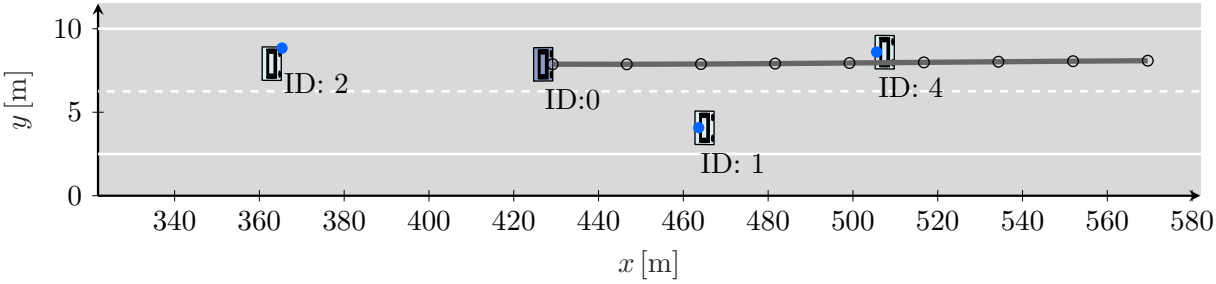

(a) Traffic scene and optimized trajectory at $t = 0\,\mathrm{s}$.




(b) Traffic scene and optimized trajectory at $t = 3\,\mathrm{s}$.

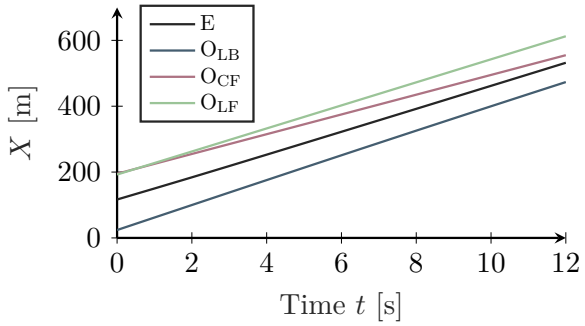



(c) Traffic scene and optimized trajectory at $t = 6\,\mathrm{s}$.

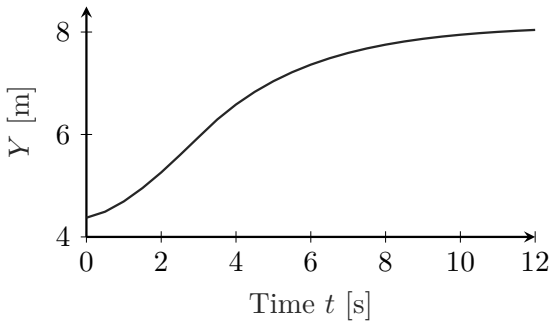



(d) Traffic scene and optimized trajectory at $t = 9\,\mathrm{s}$.

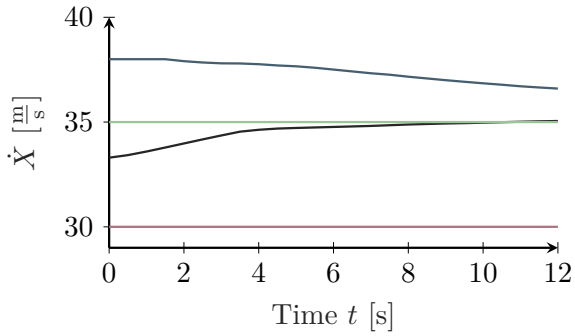
Figure 8.8.: Illustration of the closed-loop application of the cooperative LQ game formulation with imposed chance-constraints in simulation. A simple left lane change scenario was chosen.



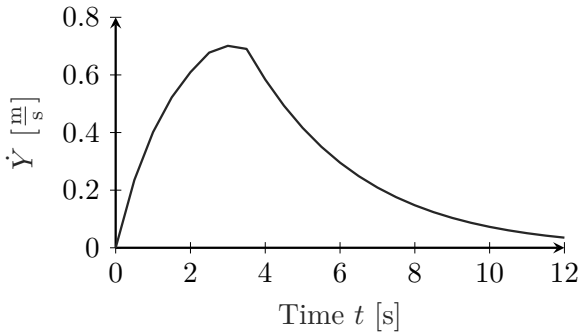

(a) Position $X$ of all vehicles in the scene.



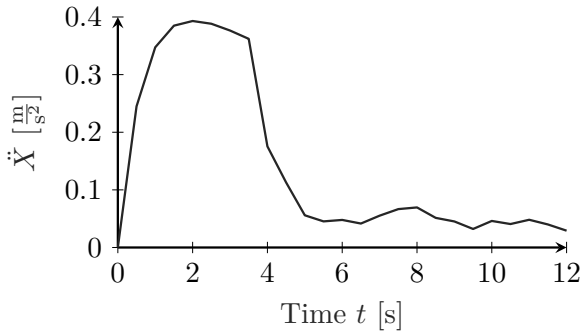

(b) Position $Y$ of the ego-vehicle.



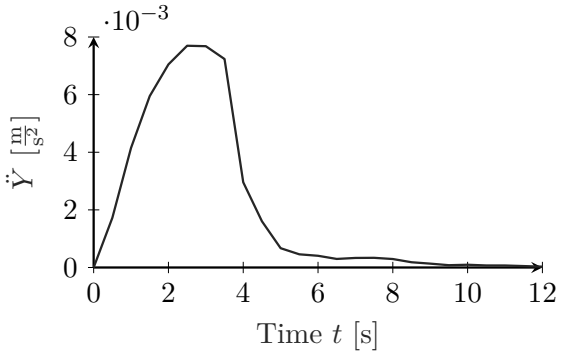

(c) Velocities $\dot{X}$ of all vehicles in the scene.




(d) Velocity $\dot{Y}$ of the ego-vehicle.




(e) Acceleration $\ddot{X}$ of the ego-vehicle.




(f) Acceleration $\ddot{Y}$ of the ego-vehicle.

Figure 8.9.: Results using the cooperative LQ game approach closed-loop in a simple lane change left scenario. Left left column represents the longitudinal quantities wheres the right on shows the lateral quantities.

# 9
# Conclusion and Outlook

Automated highway driving promises increased comfort for passengers, enhanced safety and offers many commercial advantages by automating various transportation tasks. Compared to urban scenarios, there is more structure on highways, such as lane markings reducing the complexity significantly. Yet, the problem is not solved, and the thesis at hand contributes to the behavior planning aspect, focusing on automated lane changes. A probabilistic model for discretionary lane change proposals in highway driving situations is presented. The parameters of the module are optimized using data from a driving simulator study. The results show that the model can accurately mirror the driver's lane change intentions and propose suitable discretionary lane changes. Evaluation of data from an actual test vehicle also confirms the effectiveness and suitability of the model.
The consideration of trajectory predictions promises several benefits for lane change behavior planning. It avoids the combinatoric explosion when traffic participants are considered individually and allows seamless use within lane change behavior planning. To this end, an approach from literature is enhanced to provide situation predictions for the whole traffic scene. Next, it is shown how maneuver options for lane changes of the ego-vehicle can be identified using Polygon Clipping. A graph structure is proposed that encodes these options. The situation prediction predicts lane changes of surrounding traffic participants, and it is beneficial to use this information in lane change behavior planning. Hence, the maneuver identification is modified to handle lane changes of surrounding traffic participants in a principled way. Finally, a complexity reduction step results in reduced graphs that are easier to work within the subsequent lane change planning step.
A total of three different approaches for lane change behavior planning are presented in the thesis at hand. The first assumes that surrounding traffic participants stick to their respective lanes during the whole planning horizon. The spatiotemporal areas resulting from maneuver identification are exploited for sampling knots of splines for the longitudinal and lateral lane change trajectories. Interactions are considered by injecting a set of ego-trajectories back into the situation prediction to calculate an interaction cost. Finally, a combined cost function is used to choose a trajectory.
The sampling of trajectories is, in general, computationally expensive when current generation ECUs without parallelization are used. Therefore, an alternative approach for the optimization of lane change trajectories is developed that uses Quadratic Programming. Safety plays a crucial role in automated driving. It is shown how Time-to-Collision and Time Headway constraints can be integrated into the Quadratic Program for longitudinal trajectory optimization.

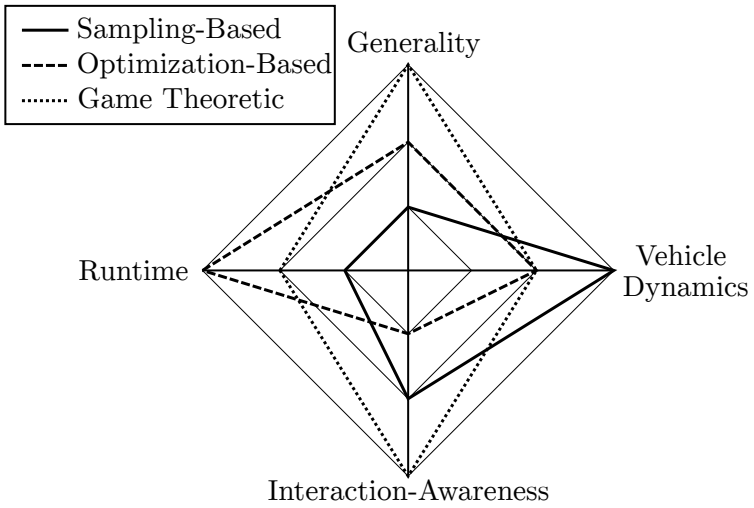



Figure 9.1.: Spider diagramm giving an overview of the attributes of all three lane change behavior planning approaches described within the thesis at hand. The sampling-based approach is the slowest when no parallelization is used and hence scores worst in the runtime dimension.

Both approaches mentioned above inject the optimized ego-vehicle trajectories into the situation prediction engine for interaction awareness. However, the sampling of knots and the constraints of the trajectory optimization result from the initial trajectory predictions. Prediction and planning are therefore decoupled. This results in conservative behavior in traffic situations where cooperation is needed. The third approach for lane change behavior planning is presented to circumvent this problem, where the trajectory optimization problem is formulated as a cooperative linear-quadratic (LQ) game. Uncertainty due to the uncertain sensor reading and multimodality are incorporated using chance constraints. A fallback strategy is developed to handle the situation in which an already started lane change needs to be aborted for safety reasons.
The spider diagram shown in Figure 9.1 ranks the three lane change behavior planning approaches in the dimensions runtime, generality, fulfillment of vehicle dynamics constraints, and interaction awareness. All approaches have different strengths and weaknesses. The sampling-based approach allows for accurate vehicle dynamics checks, but the developed heuristic does not generalize to traffic scenes where surrounding vehicles change lanes. On the other hand, the optimization-based approach is the fastest but does not anticipate other vehicles' cooperative behavior and acts more conservatively. Finally, the LQ game approach corrects this weakness at the cost of higher runtime. Of all approaches, it seems the most suitable for practical application on current generation ECUs in a modular automated driving software system. It fulfills all requirements formulated in the motivation of the thesis at hand.
Note that all behavior planning approaches described in the thesis at hand have several limitations. A Quadratic Programming formulation allows the use of linear system dynamics and achieves the runtime requirements. Such simplified dynamics will not perform equally well in all highway driving velocity regions, leading to unsuitable trajectories provided to a local trajectory planner. There are two general strategies to cope with this. One possibility is to eradicate the behavior planning stage and work with a local trajectory planning approach with responsibilities including discretionary lane changes. The

second approach is to use more accurate system dynamics models within the behavior planning stage and stick to a modular software system. In both cases, the requirement for a low runtime needs to be relaxed.
There are still numerous challenges in the field of trajectory prediction and behavior planning for automated driving. One exciting direction focuses on the integration of driving rules into trajectory predictions, refer to [Li+21]. The authors analyze various approaches for integration such rules as inductive biases into Deep Learning-based trajectory predictions. They use a generative adversarial network and analyze the integration of **S**ignal **T**emporal **L**ogic (STL) and syntax trees into the learning problem. [Iva+20]] proposes MATS (**M**ixtures of **A**ffine **T**ime-varying **S**ystems), where they learn dynamical system representations for trajectory forecasting and integrate those predictions with a multimodal planning methodology. Hence, it aims to solve the prediction and planning problem at once. The approach [GS21] also focuses on the integration of both problems. They develop an end-to-end trainable architecture to solve prediction and planing together by integrating game-theoretic reasoning into neural networks. An implicit differentiable layer maps preferences based on the agent's past trajectories to local Nash Equilibria representing the potential situation modes. Finally, [Liu+21] proposes an energy-based Deep Learning model to solve the combined prediction and planning problem for automated driving.
While learning-based approaches seem omnipresent in current research, it should be noted that human driving behavior is frequently not rule-consistent and doesn't always serve as the proper behavior to imitate. So, to summarize, future research should focus on a close integration of the situation prediction and behavior planning task. The critical research questions will be the algorithms' safety, robustness, high interpretability, and predictable behavior.

# A
# Appendix

## A.1. SAE Levels of Driving Automation

The SAE (**S**ociety of **A**utomotive **E**ngineers) J3016 norm, refer to [Soc21], describes the taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. It defines a total of six levels counted from zero to five. The definitions of these levels are used in the thesis at hand. Figure A.1 provides an overview of them.

## A.2. Rational Approximation of the Error Function

There exists no closed-form solution of the cumulative distribution function $\Phi$ of a Gaussian probability distribution. An approximation is therefore used in Chapter 3. Specifically, the rational approximation using economized Chebyshev polynomials from [Has55] is utilized:

$$\operatorname{erf}(x) = 1 - [\varrho(a_1 + \rho\,(a_2 + \varrho\,(a_3 + \varrho\,(a_4 + a_5\varrho))))] \cdot \exp\left(-x^2\right) + \epsilon(x), \tag{A.2.1}$$

with

$$\varrho = \frac{1}{1 + px}. \tag{A.2.2}$$

The coefficients are the following:

$$p = 0.3275911, \tag{A.2.3}$$
$$a_1 = 0.254829592, \tag{A.2.4}$$
$$a_2 = -0.284496736, \tag{A.2.5}$$
$$a_3 = 1.421413741, \tag{A.2.6}$$
$$a_4 = -1.453152027, \tag{A.2.7}$$
$$a_5 = 1.061405429, \tag{A.2.8}$$

and it achieves a maximum absolute error of:

$$|\epsilon(x)| < 1.5 \cdot 10^{-7}. \tag{A.2.9}$$

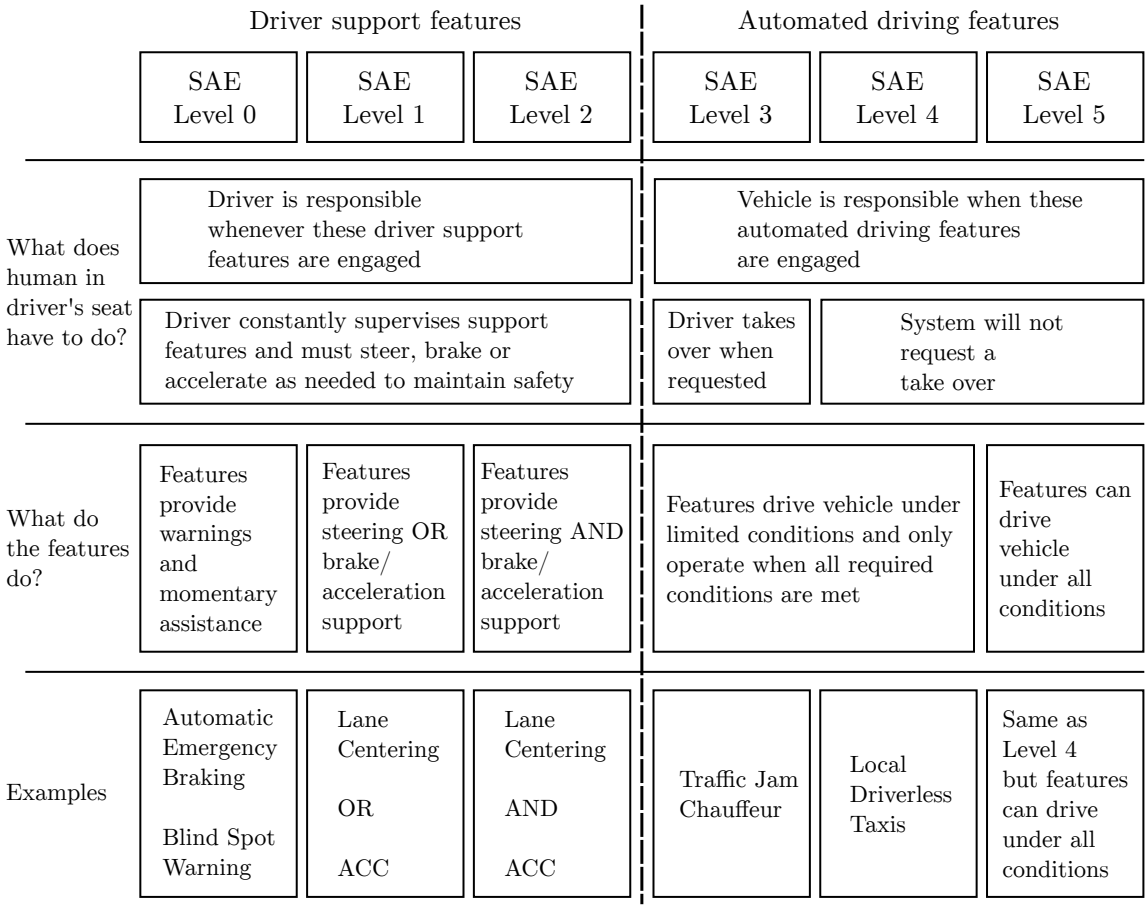



Figure A.1.: Adaption of the chart regarding SAE J3016, refer to [Soc21].

## A.3. Driving Simulator Study and Scenarios

The driving simulator mock-up is shown in Figure A.2. A total of eleven male drivers participated in the study. Nine participants were final-year undergraduate students and have limited driving experience. One participant was a final-year graduate student, and the last one was a second-year Ph.D. student. Before the study, the task was clearly explained to the participants. The trigger signal was recorded using a specific button of the driving simulator mock-up. All participants were placed in several scenarios, refer to Table A.1 and A.2, with varying traffic situations Si,L/R and SA,L. Their task was to indicate their desire to do a lane change, independently of the safety of a lane change. This is very important since the developed model triggers the maneuver planning module in an modular automated driving software system. The maneuver planning module's task is to prepare for a safe lane change finally and position the ego-vehicle correctly next to a target traffic gap. Therefore, the focus here is solely on lane change intention based on dissatisfaction with the current driving lane.

Figure A.2.: Driving simulator of the Institute of Control Theory and Systems Engineering at TU Dortmund University. It was used for conducting the driving simulator study.

The following advice was communicated to the study participants concerning their task:

> *You will be placed in a total of 11 scenarios. Your vehicle is doing lane-keeping and Adaptive Cruise Control (ACC). Assume that your vehicle also can perform automated lane changes. Your initial speed is* 30 m/s *which is also your desired velocity. By pushing the respective button on the steering wheel of the driving simulator mock-up, you can issue a lane change request to your automated vehicle. Assume that your vehicle will subsequently attempt to perform a lane change as soon as it is safe to change lanes based on the traffic situation. Hence, you do not have to worry about lane change safety since the vehicle takes care of it. Therefore, please push the button on the steering wheel as soon as you feel that your current lane is less beneficial than the target lane (left or right, depending on the scenario).*

The scenarios were designed in a way to ensure that the model parameters can be optimized. This in turn means, that all influences in the utility functions Equation (3.1.2) and Equation (3.1.3) needed to undergo variations. Tables A.1 and A.2 give an overview of the scenarios, the initial ROI velocities, and their corresponding transitions for the left and right lane change cases, respectively. A top view of the scenarios S2,L and S2,R is given in Figure A.3 and Figure A.4 respectively. The dark vehicle in the middle always corresponds to the ego-vehicle. Note how the traffic situation changes according to the velocity transitions. There is a special scenario SA,L, for optimizing the accumulator

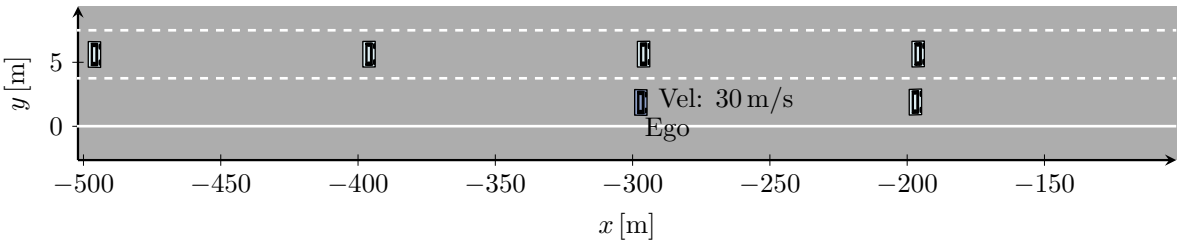
Table A.1.: Description of left lane change scenarios, the ROI vehicle velocities and their transitions.

| Scenario | $v_{\mathrm{CF}}$ $\left[\frac{\mathrm{m}}{\mathrm{s}}\right]$ | $v_{\mathrm{LB}}$ $\left[\frac{\mathrm{m}}{\mathrm{s}}\right]$ | $v_{\mathrm{LF}}$ $\left[\frac{\mathrm{m}}{\mathrm{s}}\right]$ |
|---|---|---|---|
| S1,L | $30 \rightarrow 10$ | – | – |
| S2,L | $30 \rightarrow 10$ | 40 | 40 |
| S3,L | $30 \rightarrow 10$ | 20 | 20 |
| S4,L | $30 \rightarrow 10$ | $50 \rightarrow 20$ | 20 |
| S5,L | 20 | $10 \rightarrow 30$ | $10 \rightarrow 30$ |
| S6,L | 20 | $50 \rightarrow 30$ | $20 \rightarrow 30$ |
| SA,L | $27.\overline{7}$ | – | – |

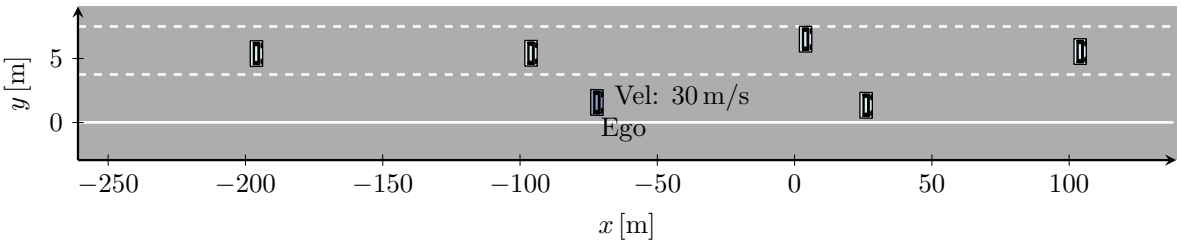
Table A.2.: Description of right lane change scenarios, the ROI vehicle velocities and their transitions.

| Scenario | $v_{\mathrm{CF}}$ $\left[\frac{\mathrm{m}}{\mathrm{s}}\right]$ | $v_{\mathrm{CB}}$ $\left[\frac{\mathrm{m}}{\mathrm{s}}\right]$ | $v_{\mathrm{RF}}$ $\left[\frac{\mathrm{m}}{\mathrm{s}}\right]$ |
|---|---|---|---|
| S1,R | 30 | – | $10 \rightarrow 40$ |
| S2,R | 30 | $50 \rightarrow 30$ | $10 \rightarrow 40$ |
| S3,R | $30 \rightarrow 10$ | – | 20 |
| S4,R | $30 \rightarrow 10$ | $50 \rightarrow 10$ | 20 |

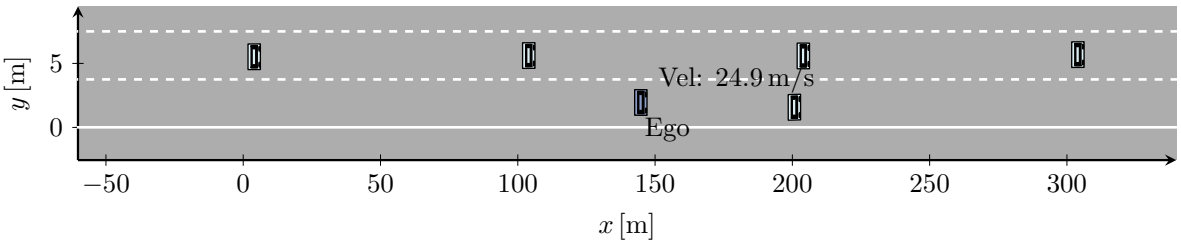
trigger module for the left lane change. In this scenario, the desired ego velocity is undershot only slightly, leading to the temporal accumulation of dissatisfaction with the current driving lane. No such scenario is designed for the right lane change since the german obligation to drive on the right-hand side of the road ensures a lane change if the traffic situation permits it. The desired velocity of the ego-vehicle is in all scenarios set fixed to $v_{\mathrm{E,des}} = 30\,\mathrm{m/s}$. All surrounding traffic participants use an Intelligent Driver Model, refer to [Tre+00], with the standard parameters and are configured to stick on their corresponding initial lanes. The velocity transitions, therefore, happen because of the safe car-following behavior.




(a) Traffic situation at $t = 0\,\mathrm{s}$.

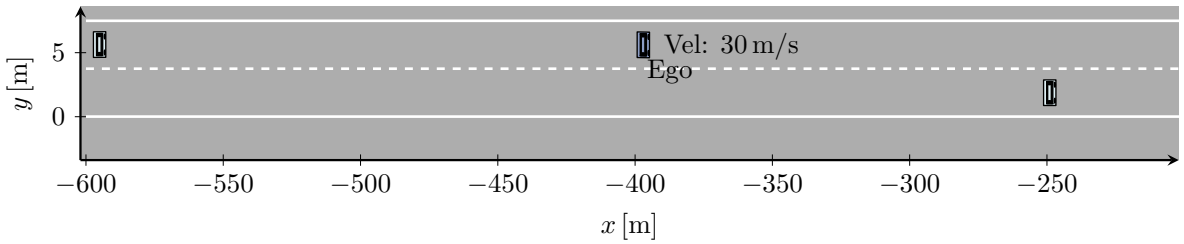



(b) Traffic situation at $t = 7.5\,\mathrm{s}$.

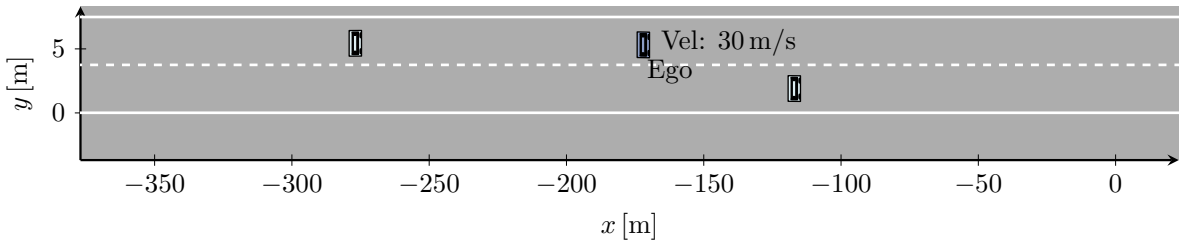



(c) Traffic situation at $t = 15\,\mathrm{s}$.

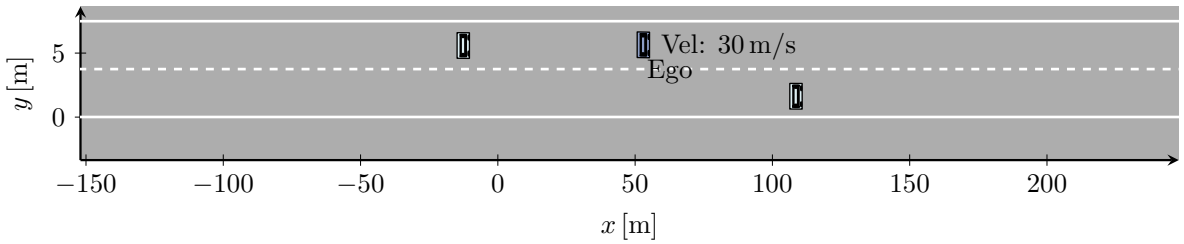
Figure A.3.: Scenario S2,L (refer to Table A.1) at three different times $t = 0\,\mathrm{s}$, $t = 7.5\,\mathrm{s}$ and $t = 15\,\mathrm{s}$.




(a) Traffic situation at $t = 0\,\mathrm{s}$.




(b) Traffic situation at $t = 7.5\,\mathrm{s}$.




(c) Traffic situation at $t = 15\,\mathrm{s}$.

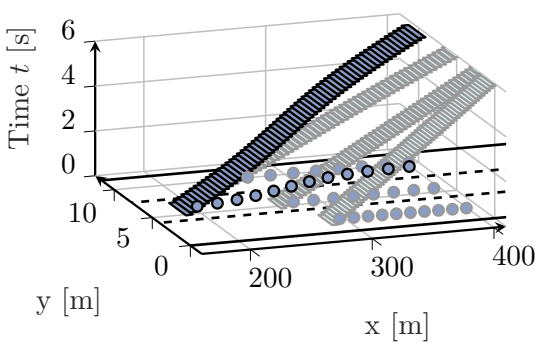
Figure A.4.: Scenario S2,R (refer to Table A.2) at three different times $t = 0\,\mathrm{s}$,$t = 7.5\,\mathrm{s}$ and $t = 15\,\mathrm{s}$.

## A.4. Additional Situation Prediction Results

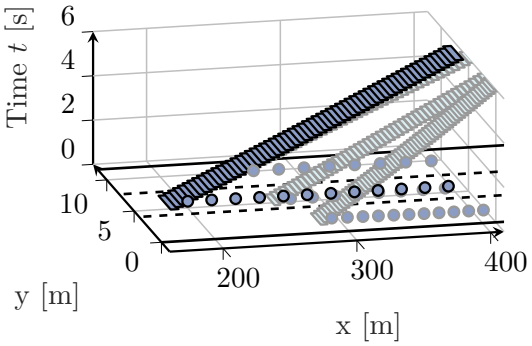
This Section provides several additional illustrations and results of the situation prediction approach described in Chapter 4 of the thesis at hand.
Table A.3 provides further situation prediction results on the simulation data for the configurations C3-S and C4-S, refer to Table 4.2 in Chapter 4.
Figure A.5 shows an threedimensional illustration of all clusters of example 4.6a.
Figure A.6 shows four additional prediction results of the application of the situation prediction approach on the highD dataset as described in Chapter 4.

Table A.3.: Results of the evaluation on a total of 3529 trajectories using the DESIM simulation environment. The mean and standard deviation is provided for all metrics.

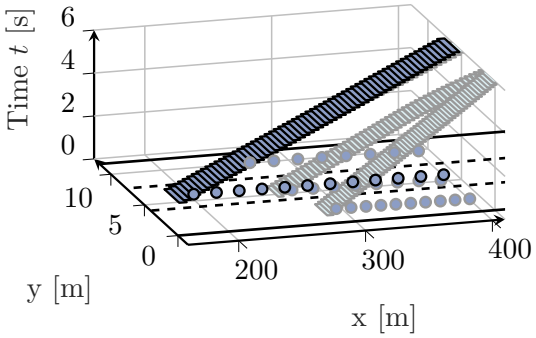
| | | | | Situation Prediction | | Trajectory Prediction | |
|---|---|---|---|---|---|---|---|
| Metric | Time [s] | CV | CV-LK | C3-S | C4-S | C3-S | C4-S |
| RMSE [m] | 1 | **0.32±0.33** | 0.49±0.41 | 0.52±0.45 | 0.51±0.45 | 0.49±0.46 | 0.48±0.46 |
| | 2.5 | 1.81±1.66 | **1.74±1.66** | 1.95±1.76 | 1.86±1.71 | 1.86±1.73 | 1.78±1.70 |
| | 5 | 6.05±5.52 | **5.29±5.26** | 5.92±5.21 | 5.63±5.12 | 5.74±5.21 | 5.50±5.15 |
| | 7.5 | 11.47±10.38 | **9.99±9.80** | 11.20±9.72 | 10.76±9.63 | 10.87±9.73 | 10.52±9.67 |
| | 10 | 17.7±15.85 | **15.56±14.90** | 17.53±14.91 | 16.96±14.83 | 17.00±14.97 | 16.57±14.88 |
| $MAE_x$ [m] | 1 | **0.22±0.30** | **0.22±0.30** | 0.27±0.37 | 0.25±0.35 | 0.26±0.36 | 0.25±0.35 |
| | 2.5 | **1.25±1.51** | **1.25±1.51** | 1.44±1.62 | 1.37±1.55 | 1.39±1.60 | 1.33±1.54 |
| | 5 | 4.49±5.02 | **4.48±5.02** | 5.00±4.95 | 4.81±4.85 | 4.84±4.92 | 4.69±4.86 |
| | 7.5 | 8.95±9.53 | **8.94±9.53** | 10.02±9.41 | 9.68±9.31 | 9.69±9.39 | 9.43±9.33 |
| | 10 | 14.24±14.66 | **14.22±14.66** | 16.09±14.62 | 15.60±14.53 | 15.57±14.65 | 15.21±14.57 |
| $MAE_y$ [m] | 1 | **0.10±0.10** | 0.27±0.26 | 0.25±0.24 | 0.26±0.26 | 0.23±0.26 | 0.23±0.26 |
| | 2.5 | 0.56±0.57 | **0.50±0.57** | 0.51±0.52 | 0.49±0.54 | 0.46±0.52 | 0.44±0.55 |
| | 5 | 1.56±1.53 | **0.81±1.01** | 0.92±1.04 | 0.82±1.06 | 0.90±1.28 | 0.81±1.10 |
| | 7.5 | 2.52±2.36 | **1.05±1.26** | 1.18±1.27 | 1.09±1.30 | 1.18±1.28 | 1.09±1.32 |
| | 10 | 3.46±3.17 | **1.34±1.53** | 1.44±1.50 | 1.36±1.52 | 1.43±1.49 | 1.36±1.53 |
| $\overline{\lvert P(\tau\lvert\hat{\tau})\rvert}$ [ ] | 1 | 0.60±0.49 | 0.55±0.48 | 1.76±1.75 | 1.79±1.81 | 1.76±1.71 | **1.80±1.75** |
| | 2.5 | 0.10±0.10 | 0.10±0.10 | 0.23±0.32 | **0.24±0.32** | 0.22±0.26 | **0.24±0.30** |
| | 5 | 0.02±0.02 | 0.02±0.02 | **0.05±0.03** | **0.05±0.03** | **0.05±0.03** | **0.05±0.03** |
| | 7.5 | 0.01±0.01 | 0.01±0.01 | **0.02±0.01** | **0.02±0.01** | **0.02±0.00** | **0.02±0.00** |
| | 10 | 0.00±0.00 | 0.00±0.00 | **0.01±0.00** | **0.01±0.00** | **0.01±0.00** | **0.01±0.00** |




(a) Cluster 1 with probability 0.37.

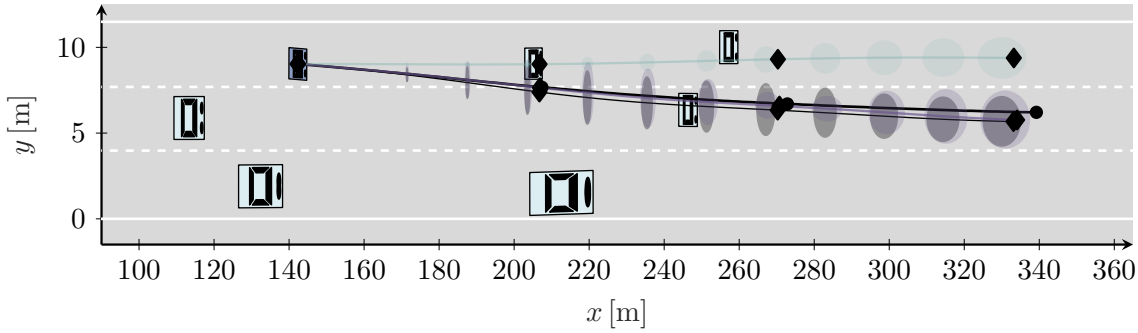



(b) Cluster 2 with probability 0.35.

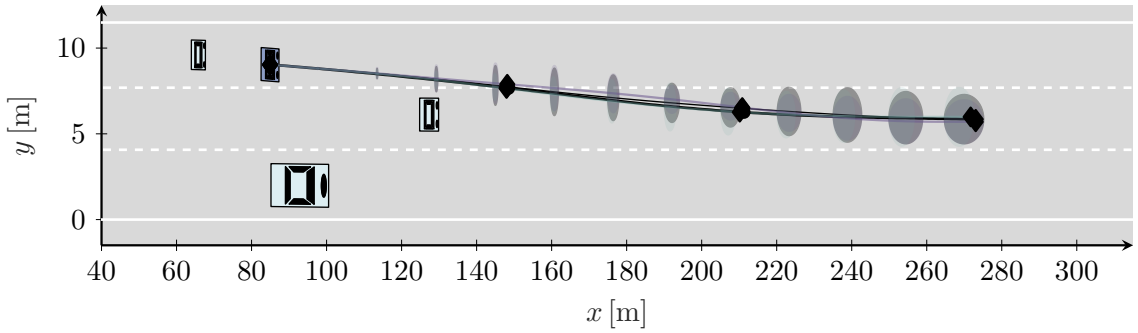



(c) Cluster 3 with probability 0.28.

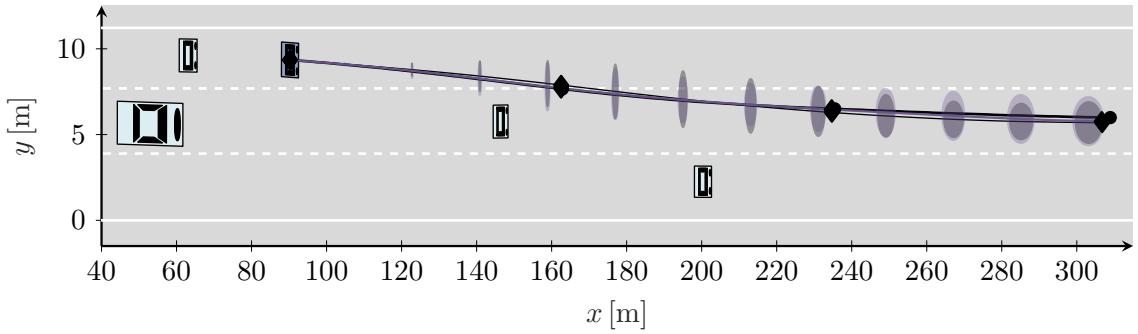
Figure A.5.: Threedimensional illustration of all clusters of example 4.6a.



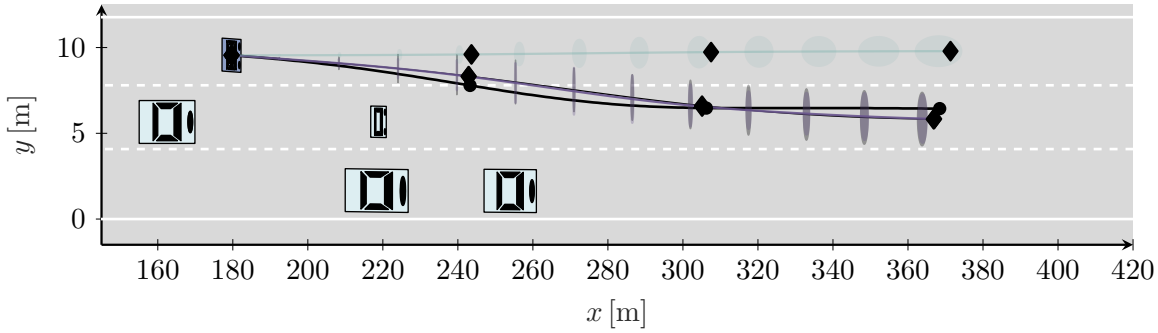

(a) Exemplary prediction result 5.




(b) Exemplary prediction result 6.




(c) Exemplary prediction result 7.




(d) Exemplary prediction result 8.

Figure A.6.: Four examplary traffic scenes of the highD dataset for the illustration of the Monte Carlo situation prediction results in additions to the ones shown in Figure 4.6.

## A.5. Classification and Regression Metrics

This Section introduces all classification and regression metrics as used in Chapter 4 of the thesis at hand.

**Explained Variance:**

$$\text{Exp. Var} = \text{Exp. Var}(v_{\text{des}}, \hat{v}_{\text{des}}) = 1 - \frac{\text{Var}(v_{\text{des}} - \hat{v}_{\text{des}})}{\text{Var}(v_{\text{des}})} \tag{A.5.1}$$

**Mean Absolute Error:**

$$\text{MAE} = \text{MAE}(v_{\text{des}}, \hat{v}_{\text{des}}) = \frac{1}{n_{\mathcal{D}}} \sum_{i=1}^{n_{\mathcal{D}}} |v_{\text{des},i} - \hat{v}_{\text{des},i}| \tag{A.5.2}$$

**Root Mean Squared Error:**

$$\text{RMSE} = \text{RMSE}(v_{\text{des}}, \hat{v}_{\text{des}}) = \sqrt{\frac{1}{n_{\mathcal{D}}} \sum_{i=1}^{n_{\mathcal{D}}} (v_{\text{des},i} - \hat{v}_{\text{des},i})^2} \tag{A.5.3}$$

**Median Absolute Error:**

$$\begin{aligned} \text{MedAE} &= \text{MedAE}(v_{\text{des}}, \hat{v}_{\text{des}}) \\ &= \text{median}(|v_{\text{des},1} - \hat{v}_{\text{des},1}|, |v_{\text{des},2} - \hat{v}_{\text{des},2}|, \ldots, |v_{\text{des},n_{\mathcal{D}}} - \hat{v}_{\text{des},n_{\mathcal{D}}}|) \end{aligned} \tag{A.5.4}$$

**Coefficient of Determination $\text{R}^2$:**

$$\text{R}^2 = \text{R}^2(v_{\text{des}}, \hat{v}_{\text{des}}) = 1 - \frac{\sum_{i=1}^{n_{\mathcal{D}}} (v_{\text{des},i} - \hat{v}_{\text{des},i})^2}{\sum_{i=1}^{n_{\mathcal{D}}} (v_{\text{des},i} - \frac{1}{n_{\mathcal{D}}} \sum_{i=1}^{n_{\mathcal{D}}} v_{\text{des},i})^2} \tag{A.5.5}$$

**Accuracy:**

$$\text{AC} = \text{AC}(cl, \hat{cl}) = \frac{1}{n_{\mathcal{D}}} \sum_{i=1}^{n_{\mathcal{D}}} \mathbb{1}(\hat{cl} = cl) \tag{A.5.6}$$

**Precision:**

$$\text{Precision} = \text{Prec.} = \frac{TP}{TP + FP} \tag{A.5.7}$$

**Recall:**

$$\text{Recall} = \text{Rec.} = \frac{TP}{TP + FN} \tag{A.5.8}$$

**$F1$-Score:**

$$\text{F1} = 2\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{A.5.9}$$

**Balanced Accuracy:**

$$\text{BAC} = \frac{1}{n_{cl}} \sum_{m=1}^{n_{cl}} \text{Recall}_m \tag{A.5.10}$$

## A.6. Additional Lane Change Maneuver Identification Illustrations

This Section provides further illustrations of all polygons involved in the complexity reduction mechanism described in Section 5.3 of Chapter 5. The areas shown in Figure A.7 correspond to the graph shown in Figure 5.11.

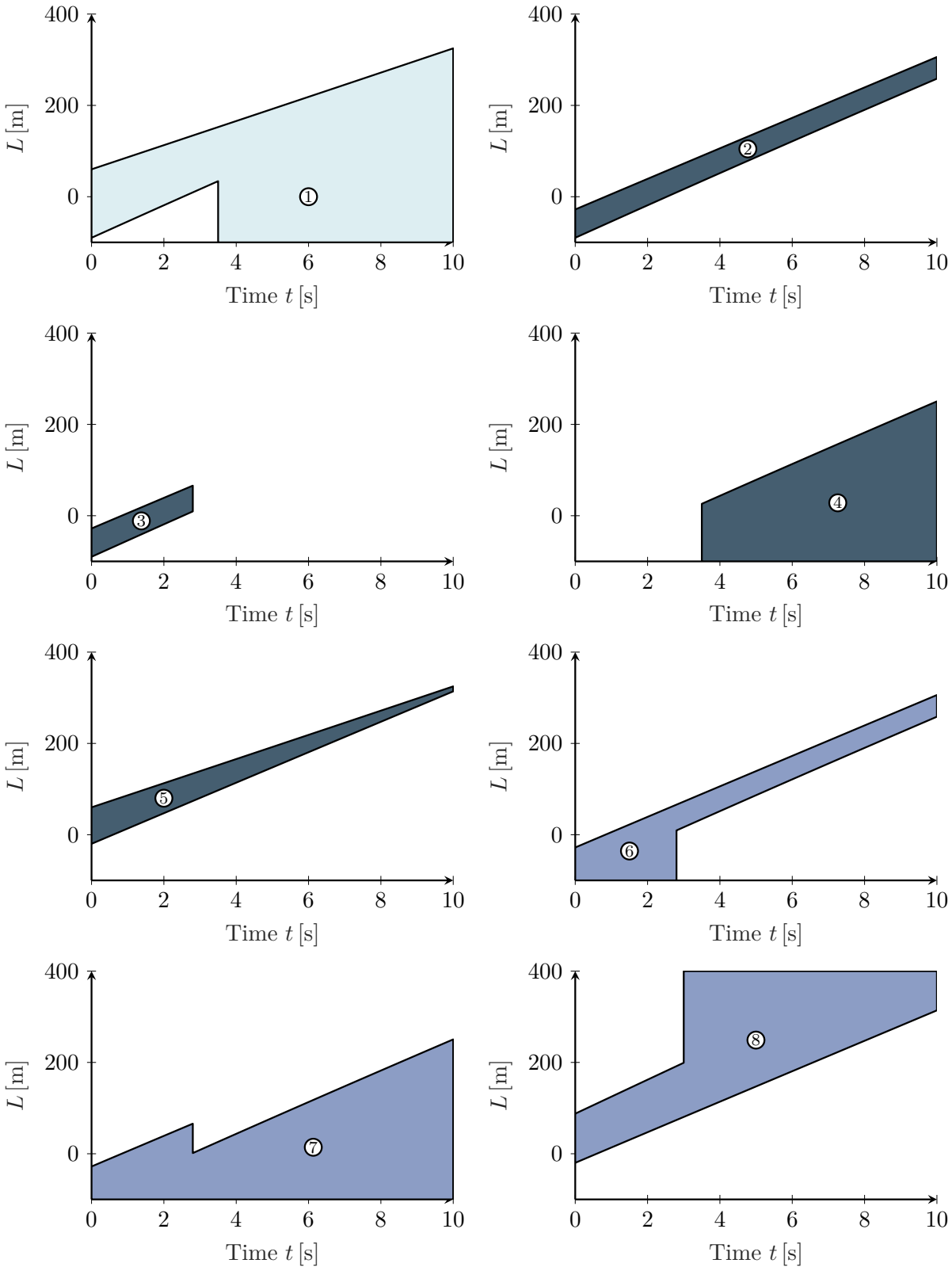



Figure A.7.: Overview of all resulting spatiotemporal areas after applying the complexity reduction mechanism.

## A.7. Definition of Differential Flatness

The following definition is taken from [Rat+95]. A (possibly) nonlinear system

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \in \mathbb{R}^n, \quad \mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{u}(t) \in \mathbb{R}^m, \tag{A.7.1}$$

is flat, if there exists an (possibly fictious) output

$$\mathbf{z}(t) = \mathbf{z}\left(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}, \ldots, \mathbf{u}^{(p)}\right) \in \mathbb{R}^m \tag{A.7.2}$$

such that the state and input can be expressed as follows:

$$\mathbf{x} = \mathbf{x}\left(\mathbf{z}, \dot{\mathbf{z}}, \ldots, \mathbf{z}^{(q)}\right) \tag{A.7.3}$$

$$\mathbf{u} = \mathbf{u}\left(\mathbf{z}, \dot{\mathbf{z}}, \ldots, \mathbf{z}^{(q)}\right) \tag{A.7.4}$$

The property is especially helpful in the field of trajectory planning. Given a flat model, planning can be done in the space of the flat output and mapped back to the input trajectory $\mathbf{u}(t)$.

## A.8. Kinematic Double-Track Vehicle Model

In Chapter 6, a nonlinear kinematic double-track vehicle model is used to check the sampled trajectories for feasibility in term of the front-wheel steering angle and steering angle rate. The model, refer to [Sch+19j], is mathematically stated as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\Psi} \\ \dot{\delta} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v\cos(\Psi) \\ v\sin(\Psi) \\ \frac{v}{l}\tan(\delta) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{\text{steer}} \\ u_{\text{acc}} \end{bmatrix}. \tag{A.8.1}$$

Herein, the state $\mathbf{x} = [x \quad y \quad \Psi \quad \delta \quad v]^\intercal$ consists of the position $x$ and $y$, orientation in the inertial system $\Psi$, front-wheel steering angle $\delta$ and speed $v$. The control input $\mathbf{u} = [u_{\text{steer}} \quad u_{\text{acc}}]^\intercal$ consists of the front-wheel steering angle rate $u_{\text{steer}}$ and acceleration measured in the inertial system $u_{\text{acc}}$. The length of the vehicle is represented by $l$.
It is shown in [Vu+21] that the nonlinear kinematic double-track vehicle model is flat with the output $\mathbf{z} = [x \quad y]^\intercal$. Chapter 6 makes use of the following two expressions for the front-wheel steering angle and rate expressed in terms of the flat output:

$$\delta = \arctan\left(l\frac{\ddot{y}\dot{x} - \ddot{x}\dot{y}}{\dot{x}^2 + \dot{y}^2}\right) \tag{A.8.2}$$

$$\dot{\delta} = \frac{\mathrm{d}}{\mathrm{d}t}\left(l\frac{\ddot{y}\dot{x} - \ddot{x}\dot{y}}{\dot{x}^2 + \dot{y}^2}\right)\left(\frac{1}{1 + l\frac{\ddot{y}\dot{x} - \ddot{x}\dot{y}}{\dot{x}^2 + \dot{y}^2}}\right). \tag{A.8.3}$$

## A.9. Additional Game Theoretic Lane Change Planning Results

The disussion of Section 8.3 of Chapter 8 used the dataset consisting of 423 trajectories, refer to Table 8.1. Here, additional results are provided using the dataset consisting of 97 examples.
The success rates are shown in A.4 and the results in terms of the remaining metrics are shown in Table A.5.

Table A.4.: Success rates on the highD trajectory dataset consisting of a total of 97 examples, refer to Table 8.1.

| Configuration | Force Ego-LC | Chance-Const. ($\eta = 0.85$) | Success Rates | |
|---|---|---|---|---|
| | | | Interactive | Ego-Only |
| C5 | No | No | 97/97(100%) | 95/97(97.9%) |
| C6 | No | Yes | 97/97(100%) | 95/97(97.9%) |
| C7 | Yes | No | 97/97(100%) | 95/97(97.9%) |
| C8 | Yes | Yes | 97/97(100%) | 95/97(97.9%) |

Table A.5.: Additional results reflecting the reconstruction capability of both the interactive and ego-only optimization variant using the highD dataset using the set of 97 trajectories, refer to Table 8.1.

| | | Interactive | | | | Ego-Only | | | |
|---|---|---|---|---|---|---|---|---|---|
| Metric | Time [s] | C5 | C6 | C7 | C8 | C5 | C6 | C7 | C8 |
| $\overline{\lvert L - L_{\text{ref}} \rvert}$ [m] | 1 | **0.04 ± 0.04** | **0.04 ± 0.04** | **0.04 ± 0.03** | **0.04 ± 0.04** | **0.04 ± 0.04** | **0.04 ± 0.04** | **0.04 ± 0.04** | **0.04 ± 0.04** |
| | 2 | 0.21 ± 0.24 | 0.21 ± 0.24 | 0.21 ± 0.24 | 0.21 ± 0.24 | 0.21 ± 0.22 | 0.21 ± 0.22 | **0.20 ± 0.22** | **0.20 ± 0.22** |
| | 4 | 1.57 ± 1.83 | 1.57 ± 1.82 | 1.59 ± 1.85 | 1.59 ± 1.84 | 1.56 ± 1.77 | 1.56 ± 1.77 | **1.51 ± 1.70** | 1.52 ± 1.71 |
| | 6 | 3.63 ± 4.11 | 3.62 ± 4.08 | 3.70 ± 4.14 | 3.69 ± 4.12 | 3.56 ± 4.02 | 3.56 ± 4.03 | **3.54 ± 3.93** | 3.55 ± 3.94 |
| $\overline{\lvert \dot{L} - \dot{L}_{\text{ref}} \rvert}$ [$\frac{\text{m}}{\text{s}}$] | 1 | 0.06 ± 0.06 | 0.06 ± 0.06 | 0.06 ± 0.06 | 0.06 ± 0.06 | 0.06 ± 0.06 | 0.06 ± 0.06 | **0.05 ± 0.06** | **0.05 ± 0.06** |
| | 2 | 0.37 ± 0.43 | 0.37 ± 0.43 | 0.37 ± 0.43 | 0.37 ± 0.43 | 0.36 ± 0.41 | 0.36 ± 0.41 | **0.35 ± 0.40** | **0.35 ± 0.40** |
| | 4 | 0.94 ± 1.07 | 0.94 ± 1.06 | 0.96 ± 1.08 | 0.96 ± 1.07 | 0.93 ± 1.05 | 0.93 ± 1.05 | **0.92 ± 1.02** | **0.92 ± 1.02** |
| | 6 | 1.12 ± 1.23 | 1.12 ± 1.22 | 1.14 ± 1.23 | 1.14 ± 1.23 | **1.10 ± 1.25** | **1.10 ± 1.25** | 1.12 ± 1.22 | 1.12 ± 1.22 |
| $\overline{\lvert \ddot{L} - \ddot{L}_{\text{ref}} \rvert}$ [$\frac{\text{m}}{\text{s}^2}$] | 1 | 0.21 ± 0.25 | 0.22 ± 0.26 | 0.22 ± 0.26 | 0.22 ± 0.26 | 0.21 ± 0.24 | 0.21 ± 0.24 | **0.20 ± 0.23** | **0.20 ± 0.23** |
| | 2 | 0.37 ± 0.43 | 0.37 ± 0.43 | 0.37 ± 0.44 | 0.37 ± 0.43 | 0.37 ± 0.43 | 0.37 ± 0.43 | **0.35 ± 0.40** | 0.36 ± 0.41 |
| | 4 | **0.24 ± 0.23** | **0.24 ± 0.23** | **0.24 ± 0.22** | **0.24 ± 0.22** | **0.24 ± 0.27** | **0.24 ± 0.28** | **0.24 ± 0.25** | **0.24 ± 0.25** |
| | 6 | **0.20 ± 0.22** | **0.20 ± 0.22** | **0.20 ± 0.20** | **0.20 ± 0.20** | 0.24 ± 0.26 | 0.24 ± 0.26 | 0.21 ± 0.23 | 0.21 ± 0.23 |
| $\overline{\lvert L^{(3)} - L^{(3)}_{\text{ref}} \rvert}$ [$\frac{\text{m}}{\text{s}^3}$] | 1 | 0.25 ± 0.32 | **0.25 ± 0.32** | **0.25 ± 0.32** | 0.26 ± 0.32 | 0.26 ± 0.31 | 0.26 ± 0.31 | **0.25 ± 0.29** | **0.25 ± 0.29** |
| | 2 | **0.12 ± 0.14** | **0.12 ± 0.14** | **0.12 ± 0.13** | **0.12 ± 0.32** | 0.14 ± 0.18 | 0.14 ± 0.19 | **0.12 ± 0.12** | **0.12 ± 0.12** |
| | 4 | **0.17 ± 0.20** | **0.17 ± 0.19** | **0.17 ± 0.20** | **0.17 ± 0.20** | 0.19 ± 0.22 | 0.18 ± 0.21 | **0.17 ± 0.19** | **0.17 ± 0.19** |
| | 6 | **0.11 ± 0.12** | **0.11 ± 0.12** | **0.11 ± 0.12** | **0.11 ± 0.12** | 0.12 ± 0.14 | 0.12 ± 0.14 | **0.11 ± 0.13** | **0.11 ± 0.13** |
| $\text{DTW}_L$ [m] | All | 14.19 ± 14.56 | 14.22 ± 14.58 | 14.35 ± 14.60 | 14.36 ± 14.63 | 14.38 ± 14.67 | 14.38 ± 14.67 | **14.17 ± 14.45** | 14.19 ± 14.45 |
| $\text{DTW}_{\dot{L}}$ [$\frac{\text{m}}{\text{s}}$] | All | 6.77 ± 8.67 | 6.75 ± 8.61 | 6.88 ± 8.76 | 6.86 ± 8.72 | **6.54 ± 8.36** | **6.54 ± 8.38** | 6.57 ± 8.14 | 6.58 ± 8.17 |
| $\text{DTW}_{\ddot{L}}$ [$\frac{\text{m}}{\text{s}^2}$] | All | 2.31 ± 2.15 | **2.30 ± 2.14** | 2.31 ± 2.13 | 2.30 ± 2.12 | 2.45 ± 2.50 | 2.45 ± 2.51 | 2.33 ± 2.23 | 2.33 ± 2.24 |
| $\text{DTW}_{L^{(3)}}$ [$\frac{\text{m}}{\text{s}^3}$] | All | 1.64 ± 1.57 | 1.65 ± 1.58 | 1.64 ± 1.57 | 1.64 ± 1.58 | 1.75 ± 1.86 | 1.76 ± 1.85 | **1.59 ± 1.51** | **1.59 ± 1.53** |

# Bibliography


[Ad17] F. Altché and A. de La Fortelle. "Partitioning of the free space-time for on-road navigation of autonomous ground vehicles". In: *IEEE Conference on Decision and Control (CDC)*. 2017, pp. 2126–2133.

[Alt+17] M. Althoff, M. Koschi, and S. Manzinger. "CommonRoad: Composable benchmarks for motion planning on roads". In: *IEEE Intelligent Vehicles Symposium (IV)*. 2017, pp. 719–726.

[And+19] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. "CasADi – A software framework for nonlinear optimization and optimal control". In: *Mathematical Programming Computation* 11 (2019), pp. 1–36.

[Ard+12] M. Ardelt, C. Cöster, and N. Kämpchen. "Highly Automated Driving on Freeways in Real Traffic Using a Probabilistic Framework". In: *IEEE Transactions on Intelligent Transportation Systems* 13.4 (Dec. 2012), pp. 1576–1585.

[BC94] D. J. Berndt and J. Clifford. "Using Dynamic Time Warping to Find Patterns in Time Series". In: *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*. AAAIWS'94. Seattle, WA: AAAI Press, 1994, pp. 359–370.

[Ben+14] P. Bender, J. Ziegler, and C. Stiller. "Lanelets: Efficient map representation for autonomous driving". In: *IEEE Intelligent Vehicles Symposium (IV)*. 2014, pp. 420–425.

[Ben+15] P. Bender, Ö. Ş. Taş, J. Ziegler, and C. Stiller. "The combinatorial aspect of motion planning: Maneuver variants in structured environments". In: *IEEE Intelligent Vehicles Symposium (IV)*. 2015, pp. 1386–1392.

[Bra+98] M. Brackstone, M. McDonald, and J. Wu. "Lane changing on the motorway: factors affecting its occurrence, and their implications". In: *9th International Conference on Road Transport Information and Control, 1998. (Conf. Publ. No. 454)*. Apr. 1998, pp. 160–164.

[Bre01] L. Breiman. "Random Forests". English. In: *Machine Learning* 45.1 (2001), pp. 5–32.

[Bry12] A. Bry. "Control, Estimation, and Planning Algorithms for Aggressive Flight using Onboard Sensing". PhD Thesis. Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics, July 2012.

[Bur+17] C. Burger, P. F. Orzechowski, Ö. Ş. Taş, and C. Stiller. "Rating cooperative driving: A scheme for behavior assessment". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2017, pp. 1–6.

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[Che+15] C. Chen, M. Rickert, and A. Knoll. "Kinodynamic motion planning with Space-Time Exploration Guided Heuristic Search for car-like robots in dynamic environments". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, pp. 2666–2671.

[Chi19] T. Chi. *ECE 598 SG Final project: Lane-change Prediction Based on LSTM*. Dec. 2019. URL: `https://github.com/chitianhao/lane-change-prediction-lstm`.

[Cle+20a] S. L. Cleac'h, M. Schwager, and Z. Manchester. "ALGAMES: A Fast Solver for Constrained Dynamic Games". In: *Proceedings of Robotics: Science and Systems*. Corvalis, Oregon, USA, July 2020.

[Cle+20b] S. L. Cleac'h, M. Schwager, and Z. Manchester. *LUCIDGames: Online Unscented Inverse Dynamic Games for Adaptive Trajectory Prediction and Planning*. 2020. arXiv: `2011.08152`.

[CV95] C. Cortes and V. Vapnik. "Support-vector networks". In: *Machine learning* 20.3 (1995), pp. 273–297.

[Dos+17] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. "CARLA: An Open Urban Driving Simulator". In: *Proceedings of the 1st Annual Conference on Robot Learning*. 2017, pp. 1–16.

[DT18] N. Deo and M. M. Trivedi. "Convolutional Social Pooling for Vehicle Trajectory Prediction". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2018, pp. 1549–15498.

[El +01] S. El Hadouaj, A. Drogoul, and S. Espié. "How to Combine Reactivity and Anticipation: The Case of Conflicts Resolution in a Simulated Road Traffic". In: *Multi-Agent-Based Simulation*. Ed. by S. Moss and P. Davidsson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 82–96.

[Eng05] J. Engwerda. *LQ Dynamic Optimization and Differential Games*. John Wiley & Sons, 2005.

[Est+19] K. Esterle, V. Aravantinos, and A. Knoll. "From Specifications to Behavior: Maneuver Verification in a Semantic State Space". In: *IEEE Intelligent Vehicles Symposium (IV)*. 2019, pp. 2140–2147.

[Est+96] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. "A density-based algorithm for discovering clusters in large spatial databases with noise". In: AAAI Press, 1996, pp. 226–231.

[Eve+16] N. Evestedt, E. Ward, J. Folkesson, and D. Axehill. "Interaction aware trajectory planning for merge scenarios in congested traffic situations". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2016, pp. 465–472.

[Fed20a] Federal Motor Transport Authority of Germany. *Press release No. 6/2020.* 2020. URL: https://de.statista.com/statistik/daten/studie/12131/umfrage/pkw-bestand-in-deutschland/.

[Fed20b] Federal Statistical Office of Germany. *Press release No. 061.* 2020. URL: https://www.destatis.de/EN/Press/2020/02/PE20_061_46241.html.

[Fis+19] J. F. Fisac, E. Bronstein, E. Stefansson, D. Sadigh, S. S. Sastry, and A. D. Dragan. "Hierarchical Game-Theoretic Planning for Autonomous Vehicles". In: *IEEE International Conference on Robotics and Automation (ICRA).* 2019, pp. 9590–9596.

[Fri00] J. H. Friedman. "Greedy Function Approximation: A Gradient Boosting Machine". In: *Annals of Statistics* 29 (2000), pp. 1189–1232.

[Gao+20] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid. "VectorNet: Encoding HD Maps and Agent Dynamics From Vectorized Representation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 2020, pp. 11522–11530.

[Ge+19] J. I. Ge, B. Schürmann, R. M. Murray, and M. Althoff. "Risk-aware motion planning for automated vehicle among human-driven cars". In: *American Control Conference (ACC).* 2019, pp. 3987–3993.

[Geu+06] P. Geurts, D. Ernst, and L. Wehenkel. "Extremely Randomized Trees". In: *Mach. Learn.* 63.1 (Apr. 2006), pp. 3–42.

[GH01] S. I. Gass and C. M. Harris. "Big-M method". In: *Encyclopedia of Operations Research and Management Science.* Ed. by S. I. Gass and C. M. Harris. New York, NY: Springer US, 2001, pp. 62–62.

[GH98] G. Greiner and K. Hormann. "Efficient Clipping of Arbitrary Polygons". In: *ACM Trans. Graph.* 17.2 (Apr. 1998), pp. 71–83.

[Gip81] P. Gipps. "A behavioural car-following model for computer simulation". In: *Transportation Research Part B: Methodological* 15.2 (1981), pp. 105–111.

[Gip86] P. Gipps. "A model for the structure of lane-changing decisions". In: *Transportation Research Part B: Methodological* 20.5 (1986), pp. 403–414.

[GS21] P. Geiger and C.-N. Straehle. "Learning game-theoretic models of multiagent trajectories using implicit layers". In: *Proceedings of the AAAI Conference on Artificial Intelligence.* 2021.

[Has55] C. Hastings. *Approximations for Digital Computers.* USA: Princeton University Press, 1955.

[Hid02] P. Hidas. "Modelling lane changing and merging in microscopic traffic simulation". In: *Transportation Research Part C: Emerging Technologies* 10.5 (2002), pp. 351–371.

[Hol+15] M. Hollander, D. A. Wolfe, and E. Chicken. *Nonparametric Statistical Methods*. Second Edition. Wiley, 2015.

[Höl06] S. Hölz. *Polygon Clipper, MATLAB Central File Exchange*. 2006. URL: https://www.mathworks.com/matlabcentral/fileexchange/8818-polygon-clipper.

[Hu+19a] Y. Hu, A. Nakhaei, M. Tomizuka, and K. Fujimura. "Interaction-aware Decision Making with Adaptive Strategies under Merging Scenarios". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019, pp. 151–158.

[Hu+19b] Y. Hu, L. Sun, and M. Tomizuka. "Generic Prediction Architecture Considering both Rational and Irrational Driving Behaviors". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019, pp. 3539–3546.

[Hu+19c] Y. Hu, W. Zhan, L. Sun, and M. Tomizuka. "Multi-modal Probabilistic Prediction of Interactive Behavior via an Interpretable Model". In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. 2019, pp. 557–563.

[Hub+19] C. Hubmann, N. Quetschlich, J. Schulz, J. Bernhard, D. Althoff, and C. Stiller. "A POMDP Maneuver Planner For Occlusions in Urban Scenarios". In: *IEEE Intelligent Vehicles Symposium (IV)*. 2019, pp. 2172–2179.

[Hub64] P. J. Huber. "Robust Estimation of a Location Parameter". In: *The Annals of Mathematical Statistics* 35.1 (1964), pp. 73–101.

[Imb+17] J. Imbsweiler, R. Palyafári, F. P. León, and B. Deml. "Untersuchung des Entscheidungsverhaltens in kooperativen Verkehrssituationen am Beispiel einer Engstelle". In: *at - Automatisierungstechnik* 65.7 (2017), pp. 477–488.

[Iva+20] B. Ivanovic, A. Elhafsi, G. Rosman, A. Gaidon, and M. Pavone. "MATS: An Interpretable Trajectory Forecasting Representation for Planning and Control". In: *Conference on Robot Learning (CoRL*. 2020.

[KA17] M. Koschi and M. Althoff. "Interaction-aware occupancy prediction of road vehicles". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2017, pp. 1–8.

[Kes+07] A. Kesting, M. Treiber, and D. Helbing. "General Lane-Changing Model MOBIL for Car-Following Models". In: *Transportation Research Record* 1999.1 (2007), pp. 86–94.

[Klo+19] M. Kloock, L. Kragl, J. Maczijewski, B. Alrifaee, and S. Kowalewski. "Distributed Model Predictive Pose Control of Multiple Nonholonomic Vehicles". In: *IEEE Intelligent Vehicles Symposium (IV)*. 2019, pp. 1620–1625.

[Kra+18] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein. "The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems". In: *2018 21st Inter-*

*national Conference on Intelligent Transportation Systems (ITSC)*. 2018, pp. 2118–2125.

[Kra20] R. Krajewski. *highD Tools*. Jan. 2020. URL: `https://github.com/RobertKrajewski/highD-dataset`.

[Krü+19a] M. Krüger, A. S. Novo, T. Nattermann, and T. Bertram. "Probabilistic Lane Change Prediction using Gaussian Process Neural Networks". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019, pp. 3651–3656.

[Krü+19b] M. Krüger, A. S. Novo, T. Nattermann, and T. Bertram. "Probabilistic Lane Change Prediction using Gaussian Process Neural Networks". In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019, pp. 3651–3656.

[Krü+20] M. Krüger, A. S. Novo, T. Nattermann, and T. Bertram. "Interaction-Aware Trajectory Prediction based on a 3D Spatio-TemporalTensor Representation using Convolutional-Recurrent Neural Networks". In: *IEEE Intelligent Vehicles Symposium (IV)*. 2020.

[Kur+18] K. Kurzer, F. Engelhorn, and J. M. Zöllner. "Decentralized Cooperative Planning for Automated Vehicles with Continuous Monte Carlo Tree Search". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2018, pp. 452–459.

[Lee+19] D. Lee, Y. Gu, J. Hoang, and M. Marchetti-Bowick. "Joint Interaction and Trajectory Prediction for Autonomous Driving using Graph Neural Networks". In: *Machine Learning for Autonomous Driving NeurIPS Workshop*. 2019.

[Len+16] D. Lenz, T. Kessler, and A. Knoll. "Tactical cooperative planning for autonomous highway driving using Monte-Carlo Tree Search". In: *IEEE Intelligent Vehicles Symposium (IV)*. 2016, pp. 447–453.

[Li+21] X. Li, G. Rosman, I. Gilitschenski, C. .-. Vasile, J. A. DeCastro, S. Karaman, and D. Rus. "Vehicle Trajectory Prediction Using Generative Adversarial Network With Temporal Logic Syntax Tree Features". In: *IEEE Robotics and Automation Letters (RA-L)* 6.2 (2021), pp. 3459–3466.

[Lie+19b] C. Lienke, M. Schmidt, C. Wissing, M. Keller, C. Manna, T. Nattermann, and T. Bertram. "Core components of automated driving – algorithms for situation analysis, decision-making, and trajectory planning". In: *ATZ live - Automatisiertes Fahren*. 2019, pp. 195–215.

[Liu+21] J. Liu, W. Zeng, R. Urtasun, and E. Yumer. "Deep Structured Reactive Planning". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2021.

[LK12] S. Levine and V. Koltun. "Continuous Inverse Optimal Control with Locally Optimal Examples". In: *ICML '12: Proceedings of the 29th International Conference on Machine Learning*. 2012.

[MA17] S. Manzinger and M. Althoff. "Negotiation of drivable areas of cooperative vehicles for conflict resolution". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2017, pp. 1–8.

[MA18] S. Manzinger and M. Althoff. "Tactical Decision Making for Cooperative Vehicles Using Reachable Sets". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2018, pp. 444–451.

[Mac67] J. Macqueen. "Some methods for classification and analysis of multivariate observations". In: *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*. 1967, pp. 281–297.

[Mar+09] F. Martinez, A. J. Rueda, and F. R. Feito. "A new algorithm for computing Boolean operations on polygons". In: *Computers and Geosciences* 35.6 (2009), pp. 1177–1185.

[McD+97] M. McDonald, J. Wu, and M. Brackstone. "Development of a fuzzy logic based microscopic motorway simulation model". In: *Proceedings of Conference on Intelligent Transportation Systems*. Nov. 1997, pp. 82–87.

[Men+18] C. Menéndez-Romero, M. Sezer, F. Winkler, C. Dornhege, and W. Burgard. "Courtesy Behavior for Highly Automated Vehicles on Highway Interchanges". In: *IEEE Intelligent Vehicles Symposium (IV)*. 2018, pp. 943–948.

[Mil+18] C. Miller, C. Pek, and M. Althoff. "Efficient Mixed-Integer Programming for Longitudinal and Lateral Motion Planning of Autonomous Vehicles". In: *IEEE Intelligent Vehicles Symposium (IV)*. 2018, pp. 1954–1961.

[Mor+10] S. Moridpour, M. Sarvi, and G. Rose. "Lane changing models: a critical review". In: *Transportation Letters* 2.3 (2010), pp. 157–173.

[Mur04] A. Murta. *GPC: General Polygon Clipper library*. Dec. 2004. URL: `https://github.com/rickbrew/GeneralPolygonClipper`.

[Nau+19] M. Naumann, H. Königshof, and C. Stiller. "Provably Safe and Smooth Lane Changes in Mixed Trafic". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019, pp. 1832–1837.

[Nau+20] M. Naumann, L. Sun, W. Zhan, and M. Tomizuka. "Analyzing the Suitability of Cost Functions for Explaining and Imitating Human Driving Behavior based on Inverse Reinforcement Learning". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 5482–5487.

[Nil+16a] J. Nilsson, M. Brännström, J. Fredriksson, and E. Coelingh. "Longitudinal and Lateral Control for Automated Yielding Maneuvers". In: *IEEE Transactions on Intelligent Transportation Systems* (May 2016).

[Nil+16b] J. Nilsson, J. Silvlin, M. Brannstrom, E. Coelingh, and J. Fredriksson. "If, When, and How to Perform Lane Change Maneuvers on Highways". In: *IEEE Intelligent Transportation Systems Magazine* 8.4 (2016), pp. 68–78.

[Nil+17] J. Nilsson, M. Brännström, E. Coelingh, and J. Fredriksson. "Lane Change Maneuvers for Automated Vehicles". In: *IEEE Transactions on Intelligent Transportation Systems* 18.5 (2017), pp. 1087–1096.

[PA18] C. Pek and M. Althoff. "Computationally Efficient Fail-safe Trajectory Planning for Self-driving Vehicles Using Convex Optimization". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2018, pp. 1447–1454.

[Par62] E. Parzen. "On Estimation of a Probability Density Function and Mode". In: *Ann. Math. Statist.* 33.3 (Sept. 1962), pp. 1065–1076.

[Ped+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[Pho75] B. T. Phong. "Illumination for computer generated pictures". In: *Commun. ACM* 18 (1975), pp. 311–317.

[Qia+16] X. Qian, F. Altché, P. Bender, C. Stiller, and A. de La Fortelle. "Optimal trajectory planning for autonomous driving integrating logical constraints: An MIQP perspective". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2016, pp. 205–210.

[Rat+95] M. Rathinam, R. M. Murray, and W. M. Sluis. *Differential Flatness of Mechanical Control Systems: A Catalog of Prototype Systems*. 1995.

[Rat16] C. Rathgeber. "Trajektorienplanung und -folgeregelungfür assistiertes bis hochautomatisiertes Fahren". Doctoral Thesis, TU Berlin. Berlin, 2016.

[Rhi+19] N. Rhinehart, R. Mcallister, K. Kitani, and S. Levine. "PRECOG: PREdiction Conditioned on Goals in Visual Multi-Agent Settings". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 2821–2830.

[Ric+16] C. Richter, A. Bry, and N. Roy. "Polynomial Trajectory Planning for Aggressive Quadrotor Flight in Dense Indoor Environments". In: *Robotics Research: The 16th International Symposium ISRR*. Ed. by M. Inaba and P. Corke. Intl. Symposium ISRR, 2016, pp. 649–666.

[Rös+17] C. Rösmann, M. Oeljeklaus, F. Hoffmann, and T. Bertram. "Online trajectory prediction and planning for social robot navigation". In: *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. Piscataway, NJ: IEEE, 2017, pp. 1255–1260.

[SA18] S. Söntges and M. Althoff. "Computing the Drivable Area of Autonomous Road Vehicles in Dynamic Road Scenes". In: *IEEE Transactions on Intelligent Transportation Systems* 19.6 (2018), pp. 1855–1866.

[Sal+20] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone. "Trajectron++: Dynamically-Feasible Trajectory Forecasting With Heterogeneous Data". In: *European Conference on Computer Vision (ECCV)*. 2020.

[Sch+12] W. J. Schakel, V. L. Knoop, and B. van Arem. "Integrated Lane Change Model with Relaxation and Synchronization". In: *Transportation Research Record* 2316.1 (2012), pp. 47–57.

[Sch+14] J. Schlechtriemen, A. Wedel, G. Breuel, and K. Kuhnert. "A probabilistic long term prediction approach for highway scenarios". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2014, pp. 732–738.

[Sch+16] J. Schlechtriemen, K. P. Wabersich, and K. Kuhnert. "Wiggling through complex traffic: Planning trajectories constrained by predictions". In: *IEEE Intelligent Vehicles Symposium (IV)*. 2016, pp. 1293–1300.

[Sch+19a] M. Schmidt, C. Manna, J. Braun, C. Wissing, M. Mohamed, and T. Bertram. "An Interaction-Aware Lane Change Behavior Planner for Automated Vehicles on Highways Based on Polygon Clipping". In: *IEEE Robotics and Automation Letters (RA-L)* 4.2 (2019), pp. 1876–1883.

[Sch+19b] M. Schmidt, C. Wissing, J. Braun, T. Nattermann, and T. Bertram. "Maneuver Identification for Interaction-Aware Highway Lane Change Behavior Planning based on Polygon Clipping and Convex Optimization". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019, pp. 3948–3953.

[Sch+19h] P. Schörner, C. Hubschneider, J. Härtl, R. Polley, and J. M. Zöllner. "Grid-Based Micro Traffic Prediction using Fully Convolutional Networks". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019, pp. 4540–4547.

[Sch+19i] M. Schreiber, S. Hoermann, and K. Dietmayer. "Long-Term Occupancy Grid Prediction Using Recurrent Neural Networks". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2019, pp. 9299–9305.

[Sch+19j] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus. "Social behavior for autonomous vehicles". In: *Proceedings of the National Academy of Sciences (PNAS)* 116.50 (2019), pp. 24972–24978.

[Sch+21a] S. Schaefer, K. Leung, B. Ivanovic, and M. Pavone. "Leveraging Neural Network Gradients within Trajectory Optimization for Proactive Human-Robot Interactions". In: *IEEE International Conference on Robotics and Automation (ICRA)*. Xi'an, China, May 2021.

[Sch+21b] L. Schafer, S. Manzinger, and M. Althoff. "Computation of Solution Spaces for Optimization-based Trajectory Planning". In: *IEEE Transactions on Intelligent Vehicles* (2021).

[Sch+21d] M. Schmidt, C. Wissing, T. Nattermann, and T. Bertram. "A probabilistic model for discretionary lane change proposals in highway driving situations". In: *Forschung im Ingenieurwesen, Open Access* (2021).

[SH74] I. E. Sutherland and G. W. Hodgman. "Reentrant Polygon Clipping". In: *Commun. ACM* 17.1 (Jan. 1974), pp. 32–42.

[Sha+17] S. Shalev-Shwartz, S. Shammah, and A. Shashua. "On a Formal Model of Safe and Scalable Self-driving Cars". In: *CoRR* abs/1708.06374 (2017).

[Soc21] Society of Automotive Engineers (SAE). *Standard J3016*. 2021. URL: `https://www.sae.org/standards/content/j3016_202104/`.

[SP14] W. Schwarting and P. Pascheka. "Recursive conflict resolution for cooperative motion planning in dynamic highway traffic". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2014, pp. 1039–1044.

[Sri+19] S. Srikanth, J. A. Ansari, R. K. Ram, S. Sharma, J. K. Murthy, and K. M. Krishna. "INFER: INtermediate representations for FuturE pRediction". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019, pp. 942–949.

[Ste+20] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. "OSQP: An Operator Splitting Solver for Quadratic Programs". In: *Mathematical Programming Computation* (2020).

[Sto+19] T. Stoll, J. Imbsweiler, B. Deml, and M. Baumann. "Three Years CoInCar: What Cooperatively Interacting Cars Might Learn from Human Drivers". In: *IFAC Symposium on Intelligent Autonomous Vehicles (IAV)* 52.8 (2019), pp. 105–110.

[Sun+18] L. Sun, W. Zhan, M. Tomizuka, and A. D. Dragan. "Courteous Autonomous Cars". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 663–670.

[Sun+19] L. Sun, W. Zhan, Y. Hu, and M. Tomizuka. "Interpretable Modelling of Driving Behaviors in Interactive Driving Scenarios based on Cumulative Prospect Theory". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019, pp. 4329–4335.

[Tan19] Y. Tang. "Towards Learning Multi-Agent Negotiations via Self-Play". In: *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. 2019, pp. 2427–2435.

[Tas+18] Ö. S. Tas, F. Hauser, and C. Stiller. "Decision-Time Postponing Motion Planning for Combinatorial Uncertain Maneuvering". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2018, pp. 2419–2425.

[Tes21] I. Tesla. *Tesla AI Day 2021 - Youtube Livestream*. Aug. 2021. URL: `https://www.youtube.com/watch?v=j0z4FweCy4M`.

[Tes22] I. Tesla. *Description of Tesla Autopilot and Full Self-Driving Functionalities*. Aug. 2022. URL: `https://www.tesla.com/support/autopilot`.

[Tip01] M. E. Tipping. "Sparse Bayesian Learning and the Relevance Vector Machine". In: *J. Mach. Learn. Res.* 1 (Sept. 2001), pp. 211–244.

[TK09] M. Treiber and A. Kesting. “Modeling Lane-Changing Decisions with MOBIL”. In: *Traffic and Granular Flow ’07*. Ed. by C. Appert-Rolland, F. Chevoir, P. Gondret, S. Lassarre, J.-P. Lebacque, and M. Schreckenberg. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 211–221.

[Tol+03] T. Toledo, H. N. Koutsopoulos, and M. E. Ben-Akiva. “Modeling Integrated Lane-Changing Behavior”. In: *Transportation Research Record* 1857.1 (2003), pp. 30–38.

[Tol+09] T. Toledo, H. N. Koutsopoulos, and M. Ben-Akiva. “Estimation of an integrated driving behavior model”. In: *Transportation Research Part C: Emerging Technologies* 17.4 (2009), pp. 365–380.

[Tre+00] M. Treiber, A. Hennecke, and D. Helbing. “Congested traffic states in empirical observations and microscopic simulations”. In: *Phys. Rev. E* 62 (2 2000), pp. 1805–1824.

[TS19] Y. C. Tang and R. Salakhutdinov. “Multiple Futures Prediction”. In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2019.

[Ulb+15] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer. “Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving”. In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. 2015, pp. 982–988.

[UM15] S. Ulbrich and M. Maurer. “Towards Tactical Lane Change Behavior Planning for Automated Vehicles”. In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2015, pp. 989–995.

[Vat92] B. R. Vatti. “A Generic Solution to Polygon Clipping”. In: *Commun. ACM* 35.7 (July 1992), pp. 56–63.

[Vu+21] T. M. Vu, R. Moezzi, J. Cyrus, J. Hlava, and M. Petru. “Feasible Trajectories Generation for Autonomous Driving Vehicles”. In: *Applied Sciences* 11.23 (2021).

[WB06] A. Wächter and L. T. Biegler. “On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming”. In: *Math. Program.* 106.1 (Mar. 2006), pp. 25–57.

[Wei+14] J. Wei, J. M. Snider, T. Gu, J. M. Dolan, and B. Litkouhi. “A behavioral planning framework for autonomous driving”. In: *IEEE Intelligent Vehicles Symposium (IV)*. 2014, pp. 458–464.

[Wer+10] M. Werling, J. Ziegler, S. Kammel, and S. Thrun. “Optimal trajectory generation for dynamic street scenarios in a Frenet Frame”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. May 2010.

[Wis+16] C. Wissing, T. Nattermann, K. Glander, A. Seewald, and T. Bertram. “Environment Simulation for the Development, Evaluation and Verification of

Underlying Algorithms for Automated Driving". In: *AmE 2016 - Automotive meets Electronics; 7th GMM-Symposium*. Mar. 2016.

[Wis+17] C. Wissing, T. Nattermann, K. Glander, and T. Bertram. "Probabilistic time-to-lane-change prediction on highways". In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. 2017, pp. 1452–1457.

[Wis+18] C. Wissing, T. Nattermann, K. Glander, and T. Bertram. "Interaction-Aware Long-term Driving Situation Prediction". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2018, pp. 137–143.

[Wis20] C. Wissing. "Trajektorienprädiktion für das automatisierte Fahren". Doctoral Thesis, TU Dortmund University. Dortmund, 2020.

[Wu+03] J. Wu, M. Brackstone, and M. McDonald. "The validation of a microscopic simulation model: a methodological case study". In: *Transportation Research Part C: Emerging Technologies* 11.6 (2003), pp. 463–479.

[Yi+19] B. Yi, P. Bender, F. Bonarens, and C. Stiller. "Model Predictive Trajectory Planning for Automated Driving". In: *IEEE Transactions on Intelligent Vehicles* 4.1 (2019), pp. 24–38.

[YK96] Q. Yang and H. N. Koutsopoulos. "A Microscopic Traffic Simulator for evaluation of dynamic traffic management systems". In: *Transportation Research Part C: Emerging Technologies* 4.3 (1996), pp. 113–129.

[Zen+19] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, and R. Urtasun. "End-To-End Interpretable Neural Motion Planner". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 8652–8661.

[Zen+20] W. Zeng, S. Wang, R. Liao, Y. Chen, B. Yang, and R. Urtasun. "DSDNet: Deep Structured self-Driving Network". In: *CoRR* abs/2008.06041 (2020).

[ZF20] ZF. *Press release No. 14023*. 2020. URL: `https://press.zf.com/press/de/releases/release_14023.html`.

[Zha+16] W. Zhan, C. Liu, C. Chan, and M. Tomizuka. "A non-conservatively defensive strategy for urban autonomous driving". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2016, pp. 459–464.

[Zha+20] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, C. Li, and D. Anguelov. "TNT: Target-driveN Trajectory Prediction". In: *CoRR* abs/2008.08294 (2020).

[Zhe14] Z. Zheng. "Recent developments and research needs in modeling lane changing". In: *Transportation Research Part B: Methodological* 60 (2014), pp. 16–32.

[Zie+08] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey. "Maximum Entropy Inverse Reinforcement Learning". In: *Proc. AAAI*. 2008, pp. 1433–1438.

[Zie+14] J. Ziegler, P. Bender, T. Dang, and C. Stiller. "Trajectory planning for Bertha — A local, continuous method". In: *IEEE Intelligent Vehicles Symposium (IV)*. 2014, pp. 450–457.

# Own Publications

M. Schmidt, C. Wissing, T. Nattermann, and T. Bertram. “A probabilistic model for discretionary lane change proposals in highway driving situations”. In: *Forschung im Ingenieurwesen, Open Access* (2021).

K. Bartsch, N. Stannartz, M. Schmidt, and T. Bertram. “Functional Simulation of automotive Lidar and Camera Sensors”. In: *AmE 2020 - Automotive meets Electronics; 11th GMM-Symposium*. 2020, pp. 1–6.

M. Schmidt, N. Stannartz, and T. Bertram. “Analysis of Depth Estimation and Semantic Segmentation Algorithms for the Environment Perception of Automated Vehicles”. In: *ATZ live - Automatisiertes Fahren*. 2020.

C. Lienke, M. Schmidt, C. Wissing, M. Keller, C. Manna, T. Nattermann, and T. Bertram. “Core components of automated driving – algorithms for situation analysis, decision-making, and trajectory planning”. In: *ATZ live - Automatisiertes Fahren*. 2019, pp. 195–215.

M. Oeljeklaus, N. Stannartz, M. Schmidt, F. Hoffmann, and T. Bertram. “Fahrzeugdetektion mit stationären Kameras zur automatischen Verkehrsüberwachung”. In: *Forschung im Ingenieurwesen* 83.2 (2019), pp. 163–171.

M. Schmidt, M. Krüger, C. Lienke, M. Oeljeklaus, T. Nattermann, M. Mohamed, F. Hoffmann, and T. Bertram. “Fahrstreifenerkennung mit Deep Learning für automatisierte Fahrfunktionen”. In: *at - Automatisierungstechnik* 67.10 (2019), pp. 866–878.

M. Schmidt, C. Manna, J. Braun, C. Wissing, M. Mohamd, and T. Bertram. “An Interaction-Aware Lane Change Behavior Planner for Automated Vehicles on Highways Based on Polygon Clipping”. In: *IEEE Robotics and Automation Letters (RA-L)* 4.2 (2019), pp. 1876–1883.

M. Schmidt, C. Wissing, J. Braun, T. Nattermann, and T. Bertram. “Maneuver Identification for Interaction-Aware Highway Lane Change Behavior Planning based on Polygon Clipping and Convex Optimization”. In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019, pp. 3948–3953.

M. Schmidt, C. Lienke, M. Oeljeklaus, M. Krüger, T. Nattermann, M. Mohamed, F. Hoffmann, and T. Bertram. “Fahrspurerkennung mit Deep Learning für automatisierte Fahrfunktionen”. In: *28. Workshop Computational Intelligence*. 2018, pp. 147–173.

M. Schmidt, C. Manna, T. Nattermann, K.-H. Glander, and T. Bertram. “Incorporating Human Driving Data into Simulations and Trajectory Predictions”. In: *4. Internationale ATZ-Fachtagung Fahrerassistenzsysteme*. 2018, pp. 212–232.

M. Schmidt, T. Nattermann, K.-H. Glander, and T. Bertram. “Online Calibration of a Camera System in Road Traffic Scenarios for the Validation of Automated Driving Functionalities”. In: *AmE 2018 - Automotive meets Electronics; 9th GMM-Symposium*. 2018, pp. 1–6.

## Additional Presentations and Poster Contributions

M. Schmidt, C. Wissing, J. Braun, T. Nattermann, and T. Bertram. "Interaktions- und optimierungsbasierte Verhaltensplanung für Spurwechselvorgänge automatisierter Fahrzeuge". In: *15. Dortmunder AutoTag.* 2020.

M. Schmidt, C. Wissing, J. Braun, T. Nattermann, and T. Bertram. "Prädiktive Manöverplanung automatisierter Fahrzeuge". In: *14. Dortmunder AutoTag.* 2019.

M. Schmidt and T. Bertram. "Anwendungen maschineller Lernverfahren im Automobil - am Beispiel der Bildverarbeitung". In: *GMA-FA 4.15 Mechatronik Sitzung.* 2018.

M. Schmidt and T. Bertram. "Anwendungen maschineller Lernverfahren im Automobil - am Beispiel der Bildverarbeitung". In: *12. BusinessForum21 - Schadenkongress.* 2018.

M. Schmidt, C. Manna, T. Nattermann, M. Mohamed, and T. Bertram. "Entwicklung eines Verfahrens zur Einbeziehung menschlicher Fahrdaten in Simulationen und Trajektorienprädiktionen". In: *13. Dortmunder AutoTag.* 2018.

M. Schmidt, C. Manna, K.-H. Glander, C. Haß, and T. Bertram. "Entwicklung einer hybriden, lernfähigen Entscheidungsstrategie für das hochautomatisierte Fahren". In: *12. Dortmunder AutoTag.* 2017.

## Patents Pending

M. Buss, M. Keller, T. Bertram, C. Lienke, C. Wissing, A. Homann, M. Schmidt, and N. Stannartz. "Method for the automated control of a motor vehicle". DE102019110217A1, US20200333792A1, CN111845766A. 2020.

C. Manna, M. Schmidt, C. Wissing, A. Homann, C. Lienke, N. Stannartz, and T. Bertram. "Method and control unit for a system for controlling a motor vehicle". DE102019107414A1, US20200301434A1, CN111731305A. 2020.

T. Nattermann, C. Wissing, M. Schmidt, A. Homann, T. Bertram, N. Stannartz, and C. Lienke. "Method and control device for controlling a motor vehicle". DE102019129879.2A, US20210129863A1, US20210129863A1. 2020.

T. Nattermann, C. Wissing, M. Schmidt, A. Homann, C. Lienke, N. Stannartz, and T. Bertram. "Method and control device for a system for the control of a motor vehicle". DE102019118366A1, US20210009119A1. 2020.

T. Nattermann, C. Wissing, M. Schmidt, A. Homann, C. Lienke, N. Stannartz, and T. Bertram. "Method for determining a lane change, driver assistance system and vehicle". DE102019122249A1, US20210053571A1. 2020.

M. Schmidt, C. Wissing, A. Homann, C. Lienke, T. Bertram, C. Manna, and K.-H. Glander. "Method and system for controlling a motor vehicle". DE102018132520A1, US20200189580A1, CN111332283A. 2020.

M. Schmidt, C. Wissing, A. Homann, C. Lienke, T. Bertram, C. Manna, and K.-H. Glander. “Method and system for determining a driving maneuver”. DE102019104974A1, US20200269871A1, CN111626538A. 2020.

C. Wissing, M. Schmidt, A. Homann, C. Lienke, T. Bertram, C. Manna, and K.-H. Glander. “Method and control unit for a system for controlling a motor vehicle”. US20200269843A1. 2020.

C. Wissing, M. Schmidt, A. Homann, C. Lienke, T. Bertram, A. Stockem Novo, M. Krüger, T. Nattermann, and K.-H. Glander. “Control system and control method for interaction-based long-term determination of trajectories for motor vehicles”. DE102019103106A1, US20200258380A1, CN111547051A. 2020.

# Awards

M. Schmidt, C. Lienke, C. Wissing, N. Stannartz, M. Krüger, and A. Homann. “Maschinelles Lernen zur Erfassung relevanter Straßeninformationen”. Award "Forum junge Spitzenforscher" 2019, Centrum für Entrepreneurship & Transfer, TU Dortmund, 2019.

M. Schmidt, C. Lienke, M. Oeljeklaus, M. Krüger, T. Nattermann, M. Mohamed, F. Hoffmann, and T. Bertram. “Fahrspurerkennung mit Deep Learning für automatisierte Fahrfunktionen”. Young-Author Award, 28. Computational Intelligence Workshop, VDI/VDE-GMA FA 5.14 Computational Intelligence, 2018.

# Supervised Theses

J. Adamek. “Entwicklung eines Modells zur Ableitung von Spurwechselwünschen automatisierter Fahrzeuge”. TU Dortmund University, 2019.

K. Bartsch. “Echtzeitfähige Objektdetektion von exterozeptiven Fahrzeugsensoren zur Simulation und Optimierung verschiedener Sensortopologien”. TU Dortmund University, 2019.

L. Meier-Ebert. “Analyse von Tiefenschätzalgorithmen für die Umfeldwahrnehmung automatisierter Fahrzeuge”. TU Dortmund University, 2019.

M. Nikolov. “Comparison of 3D-Lidar-SLAM Methods for high-precision Mapping and Localization”. TU Dortmund University, 2019.

K. N. Pham. “Robuste Vanishing Point Schätzung zur Bestimmung extrinsischer Kameraparameter”. TU Dortmund University, 2019.

R. Beckmann. “Entwicklung von Entscheidungsalgorithmen für das automatisierte Fahren”. TU Dortmund University, 2018.

A. Dikarew. “Entwicklung einer Multi-Sensor-Datenfusion zur Umfelderfassung automatisierter Fahrzeuge”. TU Dortmund University, 2018.

M. Hamza. “Causal-Dependency-Aware Lane Change Prediction using Bayesian Networks”. TU Dortmund University, 2018.

D. Sarmiento. “Development of an Incremental Search Trajectory Planner for Automated Driving in Structured Environments”. TU Dortmund University, 2018.

C. Schüler. “Entwicklung einer Multi-Sensor-Datenfusion zur Umfelderfassung automatisierter Fahrzeuge”. TU Dortmund University, 2018.

# Supervised Student Project Groups

M. Deharde, K. Jahan, T. Osterburg, T. Schmidt, S. Schütte, and S. Siddiqui. “Automotive Environment Perception”. TU Dortmund University, 2019.

K. Shinde, M. Peredo, M. Elsheikh, T.-J. Chang, M. Popescu, P. Lenzen, S. H. Raza, and R. Hana. “Environment Perception for Automated Driving”. TU Dortmund University, 2019.

P. Dorpmüller, H. Qi, F. Geiger, N. Jannasch, M. König, L. Nguyen, R. Nocon, J. Schmitz, R. Sicking, J. Stockermann, U. Ahasan, and W. Qingyang. “360° Surround View Assistant”. TU Dortmund University, 2018.