

Trajectory Planning for Automated Driving in Dynamic Environments

An Integrated Approach to Spline-based Motion Planning using
Hierarchical Trajectory Optimization

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

M.Sc. Christian Lienke, born Götte
Gelsenkirchen, Germany

Date of submission: November 23, 2021

First examiner: Univ.-Prof. Dr.-Ing. Prof. h.c. Dr. h.c. Torsten Bertram
Second examiner: Univ.-Prof. Dr.-Ing. Ulrich Konigorski

Date of approval: June 1, 2022

To my family

Acknowledgement

This thesis was written during my work at the Institute of Control Theory and Systems Engineering of the Faculty of Electrical Engineering and Information Technology at the Technical University of Dortmund.

I would like to thank Univ.-Prof. Dr.-Ing. Prof. h.c. Dr. h.c. Torsten Bertram for the unconditional support, the personal and scientific feedback and for providing me the necessary resources to successfully accomplish my doctoral studies. I am deeply grateful for the given possibility, which helped to advance my skills in the field of science, but also enabled me to grow as a person.

I would also like to thank Univ.-Prof. Dr.-Ing. Ulrich Konigorski for the interest in my work and for reviewing this thesis as a second examiner. Additionally, I would like to thank Univ.-Prof. Dr.-Ing. Timm Faulwasser for his involvement as the third examiner and Univ.-Prof. Dr.-Ing. Martin Pfost for chairing the examination committee.

I sincerely thank all employees and former employees of the institute for their valuable input and the inspiring atmosphere during my time at the department. I would like to thank apl. Prof. Dr. rer. nat. Frank Hoffmann for his ingenious suggestions and Dr.-Ing. Daniel Schauten for the instructive discussions on general control systems theory topics and for sharing his experience with regard to administrative affairs. Special thanks go to Dr.-Ing. Christian Wissing and Andreas Homann, not only for the trusting collaboration dedicated to advance automated driving functionality, but also for taking the time to provide helpful feedback and their valuable fellowship. For the close cooperation and the excellent scientific exchange I thank Manuel Schmidt, Niklas Stannartz and Martin Krüger. I also thank Christopher Diehl and Philip Dorpmüller who have supported my work in many different aspects. In addition, I would like to thank Dr.-Ing. Malte Oeljeklaus, Dr.-Ing. Christoph Termühlen, Dr.-Ing. Javier Oliva, Jan Braun, Myrel Tiemann, Franz Albers and many others whether for the cooperative work in a wealth of given tutorials, or with regard to the daily business and organizational tasks. Likewise, I thank Dr.-Ing. Christoph Rösmann for sharing his immense expertise across many different topics. For the technical support I sincerely thank Jürgen Limhoff, Rainer Müller-Burtscheid, Sascha Kersting and Halit Cicek. On the same lines I thank Gabriele Rebbe and Nicole Czerwinski for their assistance in all administrative and organizational matters.

Furthermore, I would like to express my gratitude to Prof. Dr.-Ing. Martin Keller for his encouragement and companionship throughout my career. Without his belief and persuasiveness I would not have taken this path in the first place and I am very fortunate for his personal support and thankful for the comprehensive guidance with respect to my research. The results and experiences gained throughout my research project would not have been possible without the great and universal support of the colleagues of the ZF Group. I sincerely thank Dr. rer. nat. Karl-Heinz Glander, Dr.-Ing. Carsten Haß, Dr. rer. nat. Till Nattermann and all other ZF employees involved in the

work to get the developed ideas on the road.

I also thank my parents for their relentless support which brought me into the position of pursuing my ambitious goals. Moreover, it is important to mention that this thesis could not have been finished without the substantial support of my wife Magalie. She has made many sacrifices during this time and deserves more than I could ever hope to repay. Finally, I want to thank my children for their love and distraction they provided. Your lightheartedness kept reminding me what is most important in life.

Gütersloh, June 2022

Christian Lienke

Abstract

Considering the last decades, the trend in the automotive industry to continuously increase the level of automation of vehicles is evident. A lot of research and development effort has been invested to improve upon driving safety and comfort in traffic. Nowadays, advanced driver assistance systems, and the development of automated driving functions in particular, represent one of the main areas of innovation in automotive engineering. In order to cope with challenges arising from complex dynamic environments the automated vehicle needs to perform comprehensive cognitive tasks that come along with the presence of other traffic participants and the necessity to adhere to prevailing traffic regulations. As a consequence, the automated driving task is decomposed into several sub problems. In the functional architecture of automated vehicles, motion planning that addresses the generation of a comfortable and safe trajectory is a key component that directly affects the overall driving performance. This thesis is about the development of a trajectory planning approach suitable to deal with dynamic environments. A two level hierarchical trajectory planning framework is proposed that unites the capability of optimality and spline interpolation and explicitly considers the aspect of contradicting planning objectives. The framework is designed to work in receding horizon fashion by performing cyclic replanning and hence accounts for the dynamic character of the environment. The hierarchization into two separate levels of optimization leads to an approach that covers basic driving functionality on low level, while required high level behavior is still prioritized. The presented framework relies on a spline-based trajectory representation with an underlying optimal interpolation strategy. The optimal trajectory with respect to a certain situation is found by joint optimization on high and low level. A continuous and a discrete trajectory optimization variant to generate an optimal trajectory with respect to high level objectives are presented that basically differ in the definition of possible solutions in terms of the optimal decision variables. Constraints like drivability incorporated by exploiting the flatness property of the applied vehicle model and accurate collision avoidance checking are considered explicitly to comply to essential requirements for automated driving. To evaluate the quality of the trajectory in terms of the associated driving behavior, several objectives are defined. For dedicated objectives a curvilinear frame is used, which enables a precise formulation of the desired vehicle behavior with respect to driving applications in structured environments. Hence, this measure permits to formulate objectives independent of road curvature, extending the scope of the applied trajectory planning approach to a wide range of scenarios. Evaluation works out the distinct characteristic features of the two presented high level optimization approaches, showing the achieved performance at the example of typical (highway) traffic scenarios. It is shown that both, the continuous as well as the discrete approach, are suitable to solve the trajectory generation problem supporting the idea of creating a generic trajectory planning framework for automated driving.

Contents

Nomenclature	iii
1. Introduction to Trajectory Planning in the Context of Automated Driving	1
1.1. Motivation	2
1.2. Motion Planning as Constrained Optimization	3
1.3. Coordinate Systems	5
1.4. Scope and Organization	7
2. Survey on Trajectory Planning Approaches to Automated Driving	10
2.1. Taxonomy of Trajectory Planning Approaches	11
2.2. Related Work for Automated Driving Applications	14
3. A Concept for Trajectory Planning in Dynamic Environments	20
3.1. Incorporation in the Functional Architecture of Automated Vehicles	20
3.1.1. Prerequisites and Realization	22
3.1.2. Interfaces for Trajectory Planning	23
3.2. Requirements	24
3.3. Derivation of a Trajectory Planning Concept	25
3.4. Comparison to Other Trajectory Planning Approaches	28
4. Environment-aware Maneuver Planning	30
4.1. Extraction of Maneuver Relevant Information	31
4.2. Longitudinal Maneuver Planning	35
4.3. Lateral Maneuver Planning	39
4.4. Maneuver Trajectory Generation	40
5. Modeling and Objectives for Trajectory Planning	42
5.1. Modeling of Vehicle Motion	42
5.1.1. System Modeling	42
5.1.2. Control Input Calculation	47
5.2. Modeling of the Environment	48
5.3. Trajectory Planning Objectives	52
6. Spline-based Trajectory Representation	54
6.1. Variational Formulation	55
6.2. Distinct Analytic Spline Interpolation	57
6.3. Minimum Kinematics Trajectory Generation	58
6.3.1. Formulation of the Optimal Interpolation Problem	58
6.3.2. Solution of the Optimal Interpolation Problem	60

7. Trajectory Planning for Automated Driving	64
7.1. Bilevel Programming	64
7.2. Formulation of the Trajectory Optimization Problem	65
7.3. Discrete Trajectory Optimization	71
7.3.1. Previous Work	71
7.3.2. Sampling Strategy and Trajectory Evaluation	72
7.4. Continuous Trajectory Optimization	75
7.4.1. Previous Work	76
7.4.2. Nonlinear Constrained Optimization	77
8. Evaluation in a Dynamic Environment	80
8.1. Setup of the Trajectory Planning Approaches	80
8.2. Analysis and Evaluation	83
8.3. Discussion of the Results	94
9. Conclusion and Outlook	97
9.1. Summary of the Work	97
9.2. Outlook	99
Bibliography	100
A. Appendix	114
A.1. Terminology and Definitions	114
A.2. Components of the Functional Architecture	115
A.3. Supplements to Environment-aware Maneuver Planning	119
A.4. Linear Single Track Model	125
A.5. Kinematic Vehicle Model with Ackermann Steering	125
A.6. Calculation of the Minimum Curve Radius according to RAS-L	125
A.7. Inverse Flat Transform	126
A.8. Adaptive Sampling Strategy	127
A.9. Sequential Quadratic Programming Algorithm	128
A.10. Convergence Test and Runtime Assessment for Continuous Approach	129
A.11. Further Results	131
A.12. Settings and Configuration	145

Nomenclature

The following list explains abbreviations and symbols used throughout this work. In general all mathematical symbols are introduced in their respective context. On that note the definition of symbols conforms to the following pattern:

scalars	normal font letters	a, b, \dots
vectors	bold lower case letters	$\mathbf{a}, \mathbf{b}, \dots$
matrices	bold upper case letters	$\mathbf{A}, \mathbf{B}, \dots$

Sets are denoted by uppercase blackboard bold letters ($\mathbb{A}, \mathbb{B}, \dots$) and spaces are denoted by uppercase Zapf Chancery letters ($\mathcal{A}, \mathcal{B}, \dots$). Coordinate frames are given by a leading superscript to the symbol and if not stated otherwise running indices are uniformly incremented by one. Newton's notation, also referred to as dot notation $\dot{\square}$ is used to denote the time derivative of a function. Throughout this work vehicles are generally illustrated as follows:



When referring to an obstacle it is denoted by its ID: q to label them without ambiguity.

Abbreviations and Acronyms

ABS	Antilock Braking System
ACC	Adaptive Cruise Control
AD*	Anytime Dynamic A*
ADMM	Alternating Direction Method of Multipliers
ARA*	Anytime Repairing A*
BFGS	Broyden-Fletcher-Goldfarb-Shanno
CL-RRT	Closed-loop Rapidly-exploring Random Tree
CoG	Center of Gravity
DAG	Directed Acyclic Graph
DARPA	Defense Advanced Research Projects Agency
DSC	Dynamics Stability Control
ESP	Electronic Stability Program
MATLAB	MATrix LABoratory
MPC	Model Predictive Control
OSQP	Operator Splitting Quadratic Program
PID	Proportional-integral-derivative
PROMETHEUS	PROgramme for a European Traffic of Highest Efficiency and Unprecedented Safety
QP	Quadratic Programming
RRT	Rapidly-exploring Random Tree

SQP	Sequential Quadratic Programming
cmp.	compare
e.g.	exempli gratia, for example
etc.	et cetera, and so forth
i.e.	id est, that is

Coordinate Systems

C_{\square}	Curvilinear Coordinate System
E_{\square}	World Coordinate System
F_{\square}	Vehicle Coordinate System
N_{\square}	Natural Coordinate System
R_{\square}	Tire Coordinate System

Running Indices

ℓ	Lane marker index $\ell = 0, 1, 2, 3$
ε	Minimum kinematics derivative index
ι	General running index
i	(High level) equality constraints index $i = 1, 2, \dots, \Lambda$ (Low level) equality constraints index $i = 1, 2, \dots, \Omega$
ι	Coefficient index $\iota = 0, 1, \dots, \mathcal{V}$
j	General running index
j	(High level) inequality constraints index $j = 1, 2, \dots, \Gamma$ (Low level) inequality constraints index $j = 1, 2, \dots, \Theta$
k	Ego trajectory index $k = 0, 1, \dots, K$
κ	Spline segment index $\kappa = 0, 1, \dots, \mathcal{K}$
\varkappa	Collision circles ego index $\varkappa = 1, 2, \dots, \dot{\eta}$
\dagger	SQP subproblem constraints index $\dagger = 1, 2, \dots, \Xi$
ν	Collision circles obstacle index $\nu = 1, 2, \dots, \dot{\eta}$
ω	Ego bounding box cornerpoint index $\omega = 1, 2, 3, 4$
ϱ	Obstacle index $\varrho = 1, 2, \dots, F$

Symbols and Functions

a	Acceleration
a_{\max}	Maximum vehicle acceleration
a_g	Acceleration of gravity
a_{comf}	Comfort acceleration
\check{a}_x, \check{a}_y	Comfort longitudinal and lateral acceleration limits
$\underline{\mathbf{A}}$	Minimum kinematics augmented constraint matrix
\mathbf{A}	Minimum kinematics constraint matrix
\mathbf{A}_s	Constraint matrix with specified constraint values
\mathbf{A}_u	Constraint matrix with unspecified constraint values
$\mathbf{A}_{\dagger}^{\text{con}}, \mathbf{A}_{\dagger}^{\text{con}}$	Continuity constraint matrix at the start of a spline segment
$\mathbf{A}_{\dagger}^{\text{val}}, \mathbf{A}_{\dagger}^{\text{val}}$	Value constraint matrix at the start of a spline segment
$\hat{\mathbf{A}}$	SQP jacobian of constraints
$\hat{\mathbf{a}}, \hat{\mathbf{a}}$	SQP subproblem constraints
\bar{a}_x, \bar{a}_y	Comfort longitudinal and lateral acceleration costs

$\mathbf{A}_{\hat{g}}, \mathbf{A}_{\hat{h}}$	QP linear equality and inequality constraints matrix
$\mathbf{a}_{\pm}^{\text{con}}, \mathbf{a}_{\pm}^{\text{con}}$	Mapping of spline coefficients to build a continuity constraint
$\mathbf{a}_{\pm}^{\text{val}}, \mathbf{a}_{\pm}^{\text{val}}$	Mapping of spline coefficients to build a value constraint
$\hat{\alpha}$	SQP line search step size parameter
α	Tire slip angle
\hat{b}	Collision circles distance
$\hat{\mathbf{B}}$	Damped BFGS hessian
$\underline{\mathbf{b}}$	Minimum kinematics augmented specified derivatives
\mathbf{b}	Minimum kinematics specified derivatives
\mathbf{b}_s	Constraint value vector with specified constraint values
\mathbf{b}_u	Constraint value vector with unspecified constraint values
$\mathbf{b}_{\rightarrow}, \mathbf{b}_{\rightarrow}$	Value constraints at the end of a spline segment
$\mathbf{b}_{\leftarrow}, \mathbf{b}_{\leftarrow}$	Value constraints at the start of a spline segment
$\mathbf{b}_{\hat{g}}, \mathbf{b}_{\hat{h}}$	QP linear equality and inequality constraints vector
b_{\rightarrow}	Continuity constraint value
β	Side slip angle
\mathcal{C}	Configuration space
c_v, c_h	Cornering stiffness of the front and rear tire
c	Spline coefficient
d_x, d_y	Longitudinal and lateral distance
\hat{D}	SQP merit function directional derivative
$^C d_l, ^C d_t$	Actual lead and tail vehicle distance
\bar{d}_l, \bar{d}_t	Lead and tail vehicle distance costs
d_{\min}	Minimum distance to lead and tail vehicle
$^C \bar{d}_l, ^C \bar{d}_t$	Reference lead and tail vehicle distance
δ_r	Steering angle
δ_A	Ackermann steering angle
δ_{\max}	Maximum steering angle
e	Cumulative product
$\hat{\eta}$	Collision circles number of circles
$\hat{\eta}$	SQP line search acceptance parameter
$\bar{\delta}$	Adaptive discretization variable
F	Force
$F_{a,x}, F_{a,y}$	Aerodynamic drag forces
\mathcal{F}	Functional
\hat{F}	(High level) objective function
\hat{f}	(Low level) objective function
F_z	Normal force
$F_{y,\max}$	Maximum lateral force
\hat{F}_c	High level objective for comfort driving
\hat{F}_d	High level objective for distance keeping
\hat{F}_p	High level objective for reference lateral position
\hat{F}_v	High level objective for reference velocity
\hat{G}	(High level) equality constraints
\hat{g}	(Low level) equality constraints

$\hat{\gamma}$	Lagrange multipliers for equality constraints
\hat{H}	(High level) inequality constraints
\hat{h}	Number of nodes on a level of respective sampling structure
\hat{h}	(Low level) inequality constraints
${}_a\hat{H}_4, {}_\delta\hat{H}_4, {}_g\hat{H}_4$	High level inequality constraint for system dynamics
${}_x\hat{H}_1, {}_y\hat{H}_1$	High level inequality constraint for time
${}_e\hat{H}_2$	High level inequality constraint for collision avoidance
${}_l\hat{H}_3, {}_r\hat{H}_3$	High level inequality constraint for road boundaries
\hat{i}	SQP iteration
J_z	Moment of inertia
l	Vehicle length
l_{ego}	Ego vehicle length
l_a	Inter axle distance
l_v, l_h	Distances from the center of gravity to the front and rear axle
l_c	Distance between center of gravity and center of pressure
\mathcal{L}	Lagrangian
λ	Course angle
$\hat{\lambda}$	Lagrange multipliers
m	Vehicle mass
$\mathcal{U}, \mathcal{U}_1, \mathcal{U}_2$	Adaptive discretization functions
$\hat{\mu}$	SQP merit function parameter
μ_r	Maximum radial adhesion coefficient
μ_t	Maximum tangential adhesion coefficient
$n_{\hat{B}}$	Dimension of damped BFGS Hessian
$n_{\text{px}}, n_{\text{py}}$	Number of samples for discrete optimization of the position
$n_{\text{tx}}, n_{\text{ty}}$	Number of samples for discrete optimization of time
$n_{\hat{p}}$	QP decision variable dimension
o	Configuration
$\omega_d, \omega_v, \omega_p, \omega_c$	High level objective weights
ω_r	Minimum kinematics weight
\mathbf{p}_{cc}	Center of curvature
\mathbf{p}_{cr}	Center of rotation
$\dot{\mathbf{p}}$	Ego and obstacle collision circles center
$\ddot{\mathbf{p}}$	Ego bounding box corner point
\mathbf{p}_s	Stop position
\tilde{p}_y	Lateral reference position
P	Polynomial
p	Position
p_x, p_y	Longitudinal and lateral position
\mathbf{P}_{ego}	Ego trajectory
$\hat{\mathbf{p}}$	QP decision variables
Φ	Flat transform
$\hat{\phi}$	SQP merit function
Ψ	Inverse flat transform
ψ	Yaw angle

\mathbf{Q}	QP cost matrix
\mathbf{q}	QP cost vector
r_u	Control input derivative
$\hat{\mathbf{r}}$	Damped BFGS intermediate result
r_z	Flat output derivative
\mathbb{R}	Set of real numbers
$\mathbf{R}_{ss}, \mathbf{R}_{su}, \mathbf{R}_{us}, \mathbf{R}_{uu}$	Matrices for unconstrained solution to the optimal interpolation
r_c	Minimum kinematics demanded continuity derivative order
r_f	Minimum kinematics demanded cost derivative order
$r_{\mathcal{V}}$	Distinct spline interpolation derivative
r	Derivative order
$\hat{\rho}$	Collision circles radius of obstacle vehicle
$\hat{\rho}_{ego}$	Collision circles radius of ego vehicle
ρ_{cc}	Radius of curvature
ρ_{min}	Minimum curve radius
ρ_{cr}	Radius of rotation
$\hat{\mathbf{s}}$	Damped BFGS intermediate result
\mathbf{S}	Low level solution set
S	Spline
ζ	Exploitation of the radial adhesion coefficient
$\hat{\sigma}$	Lagrange multipliers for inequality constraints
T_m	Minimum knot time difference
T_l	Constant time gap for lead vehicle
T_t	Constant time gap for tail vehicle
T_p	Ego planning horizon
T_r	Target lane reaching time
T_s	Ego trajectory sample time
t	Time
t_{sim}	Simulation time
$\hat{\tau}$	SQP line search step size parameter factor
$\hat{\theta}$	Damped BFGS parameter
\mathcal{U}	Input space
u	Control input trajectory
v_{ch}	Characteristic velocity
\mathcal{V}	Interpolation order
\tilde{v}	Reference velocity
v	Velocity
w	Vehicle width
w_{ego}	Ego vehicle width
\mathcal{X}	State space
\mathcal{X}_{free}	Free region
\mathcal{X}_{goal}	Goal region
\mathcal{X}_{dyn}	Dynamic environment constraints region
\mathcal{X}_{env}	Environment constraints region
\mathcal{X}_{stat}	Static environment constraints region

x, x_+	State trajectory, start state
ξ	Cross slope
\hat{y}	Damped BFGS intermediate result
\mathcal{Y}	Output space
y	System output trajectory
z	Flat output
\mathcal{Z}	Flat output space
\hat{z}_h	High level decision variables
\hat{z}_l	Low level decision variables
\underline{z}	Spline breakpoint, also referred to as spline knot
$\mathbf{z}_x, \mathbf{z}_y$	Flat output for longitudinal and lateral motion of the vehicle
\hat{z}	SQP decision variables

1

Introduction to Trajectory Planning in the Context of Automated Driving

In recent years a lot of effort has been investigated towards the idea of self-driving vehicles. From the viewpoint of many manufacturers the development of automated driving functions will have disruptive consequences for society. The concept of mobility for individuals and also in the context of logistics will change fundamentally. The targeted advantages comprise the aspects of improved safety in traffic, expanded access to affordable and efficient mobility as well as the ecological aspect by offering sustainable transportation solutions.

The development of advanced driver assistance functions in the past years highly contributed to improvements in road safety. The continuing development towards automated vehicles is not only leading to the improvement of active safety functions, but also to an advance in driving comfort, which relates to the effort of relieving the driver from the complex and exhausting task of driving, while enabling entertainment or an increase of productivity for the passengers.

The consequences of climate change highlight the necessity of considering ecological aspects in future mobility concepts. In the automotive industry this leads to the goal of fuel consumption reduction coming along with reduced pollution. For sure this mainly addresses alternative drivetrains, but also covers predictive driving strategies that can rigorously be achieved by means of automated vehicles.

With respect to the demographic change accompanied with an increasing average age of the population, automated driving has the potential to offer unconditional access to mobility for the elderly. The same holds true for physically handicapped people, for whom automated driving functions promise a more inclusive society, and for the rural population that can seamlessly be connected to the urban infrastructure.

The increasing traffic density in the metropolises of this world represents a major challenge for future mobility concepts. Automated driving functions could remedy traffic jams by optimizing the traffic flow and by efficient exploitation of the available traffic area.

Furthermore, the demand on individual and flexible mobility solutions gives raise to agile transportation services, which should help to provide flexible, fast and efficient mobility. Shared mobility concepts intend to decrease the number of vehicles in the urban environment to cover the problems of limited traffic area and parking space.

The development of automated driving functions will highly impact the progress and shape of services in this area.

The application of automated driving functions also allows to economize logistics in a globalized economy. It is expected that the transition of automated driving functions from research to series production will be in the area of commercial vehicles. This is because the transportation of goods over a long distance offers attractive terms. Furthermore, it provides high economic opportunities for logistics companies.

The market players cannot only be found in the same line of business. Tech companies and mobility service providers are expecting growth opportunities, since it is hardly surprising that in the presence of automated driving functions the importance of valuable software development increases rapidly.

With all these aspects in mind innovative concepts are required to efficiently shape the future mobility. The technological ingenuity and creativity of engineers is not limited with respect to highly automated driving and the vision of a world with autonomous vehicles can be advanced immeasurably. Still, today's vehicles are not entirely capable to drive autonomously. The levels of driving automation are defined by SAE (2018), reaching from SAE Level 0 (no automation) to SAE Level 5 (full vehicle autonomy). The levels apply to the driving automation feature(s) that are engaged during on-road driving of an automated vehicle.

1.1. Motivation

On the way to automated driving the vehicles are equipped with an increasing number of driver assistance systems with the ability to accurately sense their environment.

The beginning of safety-oriented driver assistance systems is marked by the antilock braking system (ABS) as well as the yaw dynamics stability control (DSC) that is also referred to as electronic stability program (ESP). Statistics attest the decrease of the number of accidents since safety-oriented driver assistance systems have been launched, but also reveal the necessity of an active driver support, as 90% of the accidents can be traced back to human errors. To reach the goal of reducing the accidents caused by human misbehavior, assistance systems could take over vehicle control.

The technical challenges regarding the application of automated driving functions are manifold. One challenge is that the approaches have to cope with highly complex urban environments and especially have to take care of collision avoidance. Hence, constraints arise from other traffic participants, but as well from vehicle dynamics. Another problem is the perception, which goes along with uncertainty, effects like occlusion of traffic participants and possibly inaccurate detection of objects. For the application in dynamic traffic it is furthermore fundamental to achieve real time operation of the developed algorithms, since the automated vehicle should account for sudden changes in the situational awareness. A vital capability of an automated vehicle is the planning of an appropriate motion with respect to various time frames. Hence, the planning algorithm should find a collision-free and comfortable trajectory on short term, but also contribute to mid and long term requirements given by high

level modules like mission planning and decision making. For reliable and safe driving in the dynamic environment, trajectory planning is fundamentally important, as new types of trajectory planning and control approaches are exactly addressing the aforementioned challenges. With the aim to achieve a further reduction in road-traffic accidents despite continued growth in traffic volumes, the vehicle has to be capable of detecting imminent collisions and to perform an adequate reaction. The vehicle environment is dynamic and highly uncertain, which necessitates the online generation of the trajectory.

1.2. Motion Planning as Constrained Optimization

A core problem of automated driving is the task of motion planning. LaValle (2006) considers motion planning under differential constraints as a variant of the classical two-point boundary value problem. The difficulty for motion planning lies in the fact that obstacle avoidance is performed additionally to the task of finding a solution through a state space that connects initial and goal states while satisfying differential constraints at the same time. The system dynamics are commonly described by:

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_+, \quad (1.2.1)$$

$$y(t) = h(x(t), u(t)), \quad (1.2.2)$$

with state $x(t) \in \mathcal{X}$, output $y(t) \in \mathcal{Y}$ and control input $u(t) \in \mathcal{U}$, for all $t \geq 0$, as well as start state $x_+ \in \mathcal{X}$. The general motion planning problem for dynamical systems is composed of:

1. A state transition equation $\dot{x}(t) = f(x(t), u(t))$ (cmp. equation 1.2.1), defined for every $x(t) \in \mathcal{X}$ and $u(t) \in \mathcal{U}$ that accounts for system dynamics, which could arise from any differential model.
2. A set \mathcal{X}_{env} of constraints on the states arising from the environment, accounting for dynamic \mathcal{X}_{dyn} and static restrictions $\mathcal{X}_{\text{stat}}$ such as lane boundaries and obstacle avoidance given by $\mathcal{X}_{\text{env}} = \mathcal{X}_{\text{stat}} \cup \mathcal{X}_{\text{dyn}}$. The free space then is $\mathcal{X}_{\text{free}} = \mathcal{X} \setminus \mathcal{X}_{\text{env}}$.
3. An initial state $x_+ \in \mathcal{X}_{\text{free}}$.
4. A set $\mathcal{X}_{\text{goal}} \subseteq \mathcal{X}_{\text{free}}$, which represents the desired goal region for the vehicle.

Then a complete algorithm must find a trajectory $x(t)$, $t \in [t_0, T_p]$ and associated control inputs $u(t)$, $t \in [t_0, T_p]$ such that the motion satisfies $x(0) = x_+$ and $x(T_p) \in \mathcal{X}_{\text{goal}}$, $x(t) \in \mathcal{X}_{\text{free}}$ and $u(t) \in \mathcal{U}$ for all $t \in [0, T_p]$. In other words the trajectory planning approach should find a solution that moves the vehicle from start to a desired goal without colliding with other obstacles.

To solve the motion planning problem the planning approach has to conduct a search in the state and/or control input space. In general there exists more than one solution to the motion planning problem. Then, usually additional objectives to be optimized are specified to rank found solutions among each other. These objectives account e.g. for minimum fuel consumption, maximum comfort or minimum travel time. This way

relevant information about the surrounding environment, vehicle dynamics as well as the desired driving behavior can seamlessly be merged in a framework that builds upon performance criteria.

The process of generating a trajectory in a way that it minimizes some measure of performance, while simultaneously satisfying a set of constraints is referred to as optimal control. Typically, methods in this field deal with generating the best trajectory by finding the control inputs of the system as functions of time.

In general an optimal control problem is concerned with minimizing a functional $\mathcal{F}(x(t), u(t))$ and a function $u(t)$ as decision variable:

$$\min_{u(t)} \mathcal{F}(x(t), u(t)) \quad (1.2.3a)$$

$$\text{s.t.} \quad \dot{x}(t) = f(x(t), u(t)) \quad \forall t \in [0, T_p], \quad (1.2.3b)$$

$$u(t) \in \mathcal{U} \quad \forall t \in [0, T_p], \quad (1.2.3c)$$

$$x(t) \in \mathcal{X}_{\text{free}} \quad \forall t \in [0, T_p], \quad (1.2.3d)$$

$$x(0) = x_{\vdash}, \quad (1.2.3e)$$

$$x(T_p) \in \mathcal{X}_{\text{goal}}. \quad (1.2.3f)$$

Obviously, the constraints 1.2.3b-1.2.3f of the optimal control problem align with the definitions of the motion planning problem stated previously.

For differentially flat systems there exists a unique relation between trajectories in the output space \mathcal{Y} and the state space \mathcal{X} and control input space \mathcal{U} . In this context the flat output $z(t) \in \mathcal{Z}$ is introduced, as the output $y(t)$ and the flat output $z(t)$ are not necessarily the same. For more information regarding differential flatness it is referred to section 5.1.2. The advantage of differential flatness in the context of trajectory planning is that the optimization problem can directly be defined with respect to the flat output and control inputs can analytically be derived as a function of the output trajectory and its derivatives. Furthermore, the representation in the flat output space might simplify the specification and optimization of the trajectory in comparison to a search over the control input space. The goal of motion planning in the flat output space is to find a flat output trajectory $z(t) \in \mathcal{Z}$, such that the corresponding unique state trajectory $x(t) \in \mathcal{X}$ and control inputs $u(t) \in \mathcal{U}$ obey the differential constraints of the vehicle for all $t \in [0, T_p]$, consider constraints imposed by the environment $x(t) \in \mathcal{X}_{\text{free}}$ for all $t \in [0, T_p]$ and reach the goal region $x(T_p) \in \mathcal{X}_{\text{goal}}$. Hence for nonlinear systems that are differentially flat the optimal control problem (1.2.3) can be mapped to a problem with the flat output $z(t)$ as decision variable.

One of the major challenges with regard to the solution of the planning problem is the handling of the complex restrictions arising from the environment and the vehicle dynamics. According to Werling et al. (2010) this prohibits to limit the optimal solution of the planning problem to a specific function class.

In this context the optimal control problem can be solved by writing it as a nonlinear program¹. This includes the transition from a functional to a function optimization

¹A nonlinear program is a constrained parameter optimization problem where either the objective function or the constraints are nonlinear. For further reading it is referred to e.g. Betts (2010) or Kelly (2017).

problem, in which the decision variables are considered to be parameters instead of functions. That way trajectory optimization deals with finding a local solution (a sequence of control inputs or e.g. flat output variables) to the optimal control problem in order to obtain an optimal trajectory².

A general solution to the constrained trajectory optimization problem can ideally be found by directly accounting for the constraints, whereas another possible way is to apply an optimization heuristic. In the latter case a search within the set of optimal solutions to the unconstrained problem is performed and the best solution, which fulfills the restrictions is chosen in order to derive a closed-form solution that approximates the constrained problem reasonably well (Werling et al. 2010).

This thesis investigates on trajectory planning approaches that solve the trajectory optimization problem in the context of automated driving. Instead of solving one comprehensive trajectory optimization problem, objectives are prioritized and grouped into two hierarchical levels. However, due to the exploitation of particular properties it is still possible to reformulate the planning problem, such that the final solution is once more obtained by solving a nonlinear program. This certainly leads to a trajectory planning approach that is in general based on constrained optimization. Solutions can be found within the designated classes of trajectory planning approaches. The class of sampling-based planning approaches is in this thesis referred to as discrete trajectory optimization, which relates to the process of constrained numerical optimization over a discretized parameter space. Optimization-based planning approaches on the other hand are also referred to as continuous trajectory optimization to account for the continuous character of the decision variables. Chapter 7 elaborates on a discrete and continuous trajectory optimization approach to automated driving in dynamic environments and discusses the characteristics and challenges with respect to the respective approach in detail.

1.3. Coordinate Systems

Throughout this thesis several coordinate systems are used, which are defined within this section. Figure 1.1 illustrates the world coordinate system, the vehicle coordinate system as well as the curvilinear coordinate system, which represent the most relevant frames within the scope of this thesis.

World Coordinate System

The world coordinate system is a global coordinate system with a fixed origin and is denoted by a leading superscript E . The world frame is a right-handed coordinate system. Whereas p_x and p_y describe the two-dimensional plane, elevation is considered to play a neglectable role in the context of trajectory planning for automated driving and will thus be omitted within this thesis.

²In contrast to an optimal policy (or global solution to the optimal control problem), which provides the optimal control for every point in the state space.

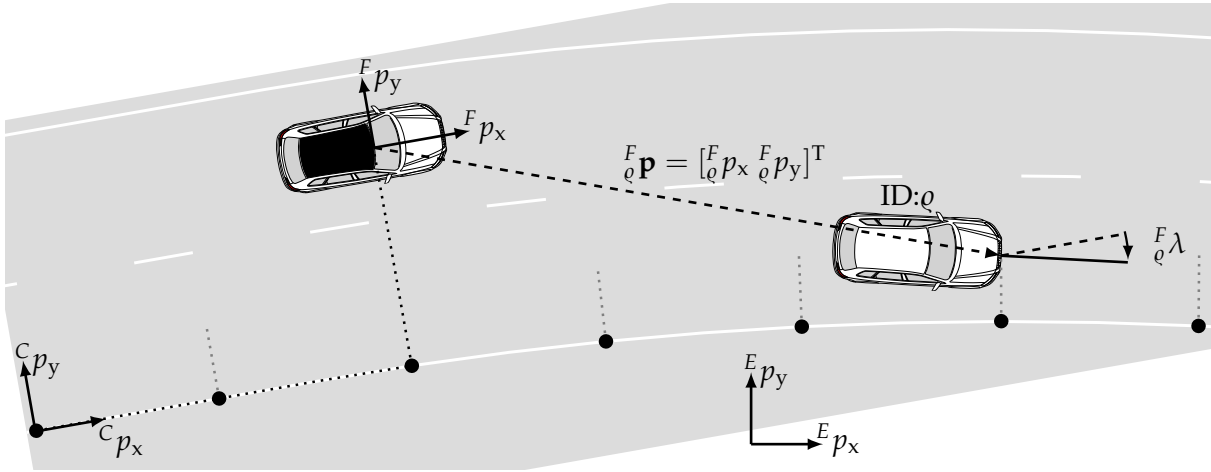


Figure 1.1.: Major coordinate frames applied throughout this thesis. The world coordinate system is a cartesian frame with fixed origin, whereas the curvilinear coordinate frame corresponds to a Frenet-Serret coordinate system that enables a description with respect to the course of the road. Especially sensor measurements, such as for example position and course angle relative to the ego vehicle, are naturally given in vehicle coordinates.

Vehicle Coordinate System

The ego coordinate system is aligned with the longitudinal axis of the vehicle pointing forward. The p_y -axis is orthogonal to the p_x -axis and points to the left. Popular choices for the origin position are the center of the front bumper or the center of gravity of the vehicle. Most of the data that is processed within the automated vehicle framework is naturally defined in this ego-centered coordinate system. This particularly encompasses sensor measurements like the position $F_q \mathbf{p}$ and course angle $F_q \lambda$ of obstacle vehicle $ID:q$ (see Figure 1.1), as well as prediction and planning algorithms. In virtue of the applied vehicle dynamics model and to the benefit of a more precise approximation of the vehicle shape in terms of collision avoidance, in this thesis the center of gravity is chosen as the origin of the vehicle frame. The vehicle coordinate system is indexed by a leading superscript F .

Curvilinear Coordinate System

The curvilinear coordinate system is defined along an arbitrary reference curve and does hence not belong to the class of cartesian coordinate systems like the world and vehicle coordinate system. From mathematical point of view the curvilinear frame corresponds to the *Frenet-Serret* coordinate system. The description in curvilinear frame enables a very intuitive way of coping with various, complex traffic scenarios. Finally, in on-road driving traffic scenarios the lateral and longitudinal driving tasks are performed with respect to the lane markers. This accounts for following the course of the road as well as for distance keeping functionality concerning other objects.

A curvilinear coordinate system with a respective lane marker as reference naturally handles curved road topologies. All quantities in curvilinear coordinates are described with respect to the reference curve by a longitudinal coordinate along the reference

curve (i.e. the arc length) and a coordinate which is orthogonal to this. To get from curvilinear to cartesian frame and vice versa a nonlinear transform has to be applied, which has to find the minimum distance between the query point and the reference curve. This certainly represents the major drawback of this coordinate system as sensor data is given in vehicle coordinates and perception results have to be transformed at least at one stage in the functional architecture of the vehicle to make them accessible in curvilinear coordinates. As it is aligned with the road course all techniques can be applied likewise for a curved and straight road geometry. Within this thesis all curvilinear coordinates are denoted with a leading superscript C .

Natural Coordinate System

The natural coordinate system basically corresponds to the vehicle coordinate frame that is shifted along the planned trajectory. It hence accounts for the future motion of the ego vehicle and can be used to express kinematic quantities for planned states from vehicle perspective. The transform thus corresponds to a two-dimensional rotation by the heading angle at the respective planned trajectory position. Within this thesis natural coordinates are denoted with a leading superscript N .

Tire Coordinate System

For the sake of completeness the tire coordinate system is mentioned here as well. This coordinate system is in particular used to define important quantities with respect to the lateral vehicle dynamics (e.g. tire forces) in section 5.1. The tire coordinate system is denoted by a leading superscript R .

1.4. Scope and Organization

In this thesis a suitable concept for trajectory planning for on-road driving in structured environments should be developed. Derived approaches should be capable to master dynamic traffic and should seamlessly integrate into the functional architecture of automated vehicles. The scope of this work should cover a set of motion primitives which represent the basic functionality that should be provided with respect to on-road driving. Namely this is lane keeping (and likewise adaptive cruise control (ACC) capability), lane changes and a target brake maneuver.

The developed approaches include aspects of several state-of-the-art trajectory planners, which hence build the fundamentals of the work presented in this thesis. The idea of Werling et al. (2010) for trajectory planning is extended from polynomials to polynomial splines via an optimal interpolation strategy developed by Bry et al. (2015). This improves the flexibility of the trajectory and enables the presented two level hierarchical trajectory optimization approach. The longitudinal and lateral trajectory are coupled via the defined objectives and constraints, intended to generate a human-like high level driving behavior and to produce dynamically feasible results for different kinds of maneuvers. The distinction between constraints and objectives enables to distinguish between basic driving goals that have to be satisfied (constraints) and

a desired non-mandatory behavior like reaching set speed, or keeping an extended safety distance (objectives). To include vehicle dynamics in the trajectory planning approach this thesis advances the concept introduced by Schlechtriemen et al. (2016). Instead of using Ackermann steering, the vehicle model utilized to derive the control inputs via flat transform is enhanced to provide a higher range of validity. The usage of a curvilinear frame is applied gingerly at different stages and beneficially supports the precise formulation of objectives and constraints. Still, trajectories are evaluated in cartesian frame. This guarantees drivability and jerk optimality also in such cases when the reference for curvilinear transform contains discontinuities.

This thesis is dedicated to develop a generic trajectory planning approach to automated driving and addresses the problem of conflicting planning objectives³ by proposing an hierarchical two level trajectory planning approach. The approach separates basic low level optimality and desired high level behavior and inherently accounts for the fact that in automated driving the desired high level behavior and hence the associated low level behavior change with respect to the current situation. The procedure can be compared to an optimization heuristic of determining the optimal solution of an unconstrained problem first and to choose the best high level solution that fulfills the constraints in a second step. The applied interpolation strategy improves on the efficiency and covers the optimal low level behavior in terms of minimum kinematics for comfort related aspects. By means of an objective function a suitable trajectory is chosen in dependence on the desired maneuver as well as on the current traffic situation. This enables the application in various, different traffic scenarios. This thesis makes the following core contributions:

- Development of a hierarchical trajectory optimization framework for trajectory planning in the context of automated driving.
- Time-parameterized spline-based trajectory representation with optimal interpolation strategy suitable for dynamic traffic.
- Seamless integration of ego vehicle dynamics via the property of differential flatness.

This work is intended to provide insight in the complex world of trajectory planning for automated vehicles. The development and analysis of trajectory planning approaches suitable for dynamic environments is reflected by the following outline:

Chapter 2 This chapter gives a comprehensive overview on planning approaches to automated driving. For reasons of clarity and comprehensibility it is necessary to group the vast amount of approaches in terms of common features. In the first section a taxonomy is introduced in order to classify planning approaches with respect to their characteristics. The second section then reviews relevant approaches in each respective class, focusing on but not limited to trajectory planning approaches to on-road automated driving.

³A typical example is the minimization of lateral acceleration for comfort purpose. This objective is in conflict with each maneuver that demands lateral acceleration such as e.g. lane changes.

- Chapter 3** Automated driving is a highly complex task. The functional architecture of automated vehicles hence decomposes the task into several sub problems and defines the interrelation between modules as well as their collaboration. To derive a concept for trajectory planning the incorporation into the functional architecture has to be considered, providing the interfaces and requirements for the developed trajectory planning approach. For this reason, this chapter introduces a functional architecture and therein defines the specific task for the trajectory planner. A holistic concept is derived, leading to the proposed two level hierarchical trajectory planning framework.
- Chapter 4** For a generic design of the trajectory planner it is vital to process each kind of maneuver decision. As a trajectory planning approach clearly benefits from a rough future motion plan that results from the maneuver decision, it is assumed that if not already considered within the decision making process, a maneuver planning approach is obligatory. Hence, for seamless cooperation the maneuver decision is compiled to a maneuver trajectory and target region. In this chapter a maneuver planning approach is presented that generates maneuver related features, which serve as an input to the trajectory planner.
- Chapter 5** For the purpose of trajectory planning in the field of automated driving particular attention has to be directed to the topic of vehicle dynamics modeling as well as to the modeling of the environment. Furthermore, the goals of automated driving have to be mapped to respective objectives in the trajectory planning approach. This chapter deals with the formulation of trajectory planning objectives, introduces approaches to static and dynamic environment modeling and gives a detailed description of the applied vehicle model.
- Chapter 6** This chapter proposes a spline-based trajectory representation for trajectory planning to improve on the overall performance of the developed trajectory planner. An optimal solution for an unconstrained planning problem is derived inferring a beneficial interpolation strategy. This approach supports the idea of the two level hierarchical trajectory planning framework and hence serves as the basis for the developed trajectory planners.
- Chapter 7** This chapter presents the mathematical background for the proposed two level hierarchical trajectory optimization framework. The trajectory optimization problem is formulated and a discrete as well as a continuous trajectory optimization approach are introduced.
- Chapter 8** In this chapter a thorough analysis with respect to the advantages and disadvantages of the developed approaches is given. The characteristics of discrete and continuous trajectory planning approaches are worked out and discussed in detail. The evaluation should furthermore give an impression of the general applicability for automated driving in dynamic environments.
- Chapter 9** This chapter briefly concludes the thesis including a discussion about the accomplished results and potential directions for future research.

2

Survey on Trajectory Planning Approaches to Automated Driving

Approaches to trajectory planning arise from many different fields. But even though each field originally considered different problems the approaches have been broadened in scope to consolidate in the topic of motion planning. Recent developments in robotics and control theory contribute to more advanced solutions with respect to the motion planning problem, such that there exists a vast amount of diverse trajectory planning approaches. The reason for this is that algorithms in this field always come along with weaknesses that lack general suitability for each planning problem since requirements largely differ with respect to the application.

Mobile robotics investigated the motion planning problem striving for advances in service robotics and unmanned vehicles to perform e.g. difficult or even dangerous tasks in areas which might not be accessible to humans. Hence, a transfer of technology from mobile robotics to the field of automotive can be identified.

However, with respect to automated driving a lot of innovative research has been done in the last decades. The team of Ernst Dickmanns at Bundeswehr University Munich started their pioneering work in the 1980's, showing the capability of on-road driving on highways. The concept was basically relying on visual information and interpretation (Dickmanns and Zapp 1987). This work was followed by the PROgramme for a European Traffic of Highest Efficiency and Unprecedented Safety (PROMETHEUS), which intended to enhance efficiency, sustainability and safety in road traffic and significantly contributed to the development of fundamentals for automated driving. In 2007 the Defense Advanced Research Projects Agency (DARPA) Urban Challenge attracted attention, in which each team was assigned individual missions. The research effort of all participants generated many innovations in the field of automated driving. An excellent overview about the achievements from technical perspective is given in Buehler et al. (2009) and Campbell et al. (2010).

In general typical fields of application concern driving in highway and urban traffic scenarios or parking. It is obvious that parking more or less concerns static scenarios with lower run time restrictions (algorithms are allowed to take more time to find a solution, as this will certainly not end up in a critical situation), whereas driving in traffic has to deal with dynamic objects, which complicates the planning problem with respect to collision avoidance and run time. The summary of related work in

this section should provide an overview of trajectory planning approaches focusing on on-road driving in dynamic environments, complemented by some considerable methods for other specific applications. In this regard the term on-road¹ refers to navigating in structured (driving applications) and semi-structured (parking applications) environments.

2.1. Taxonomy of Trajectory Planning Approaches

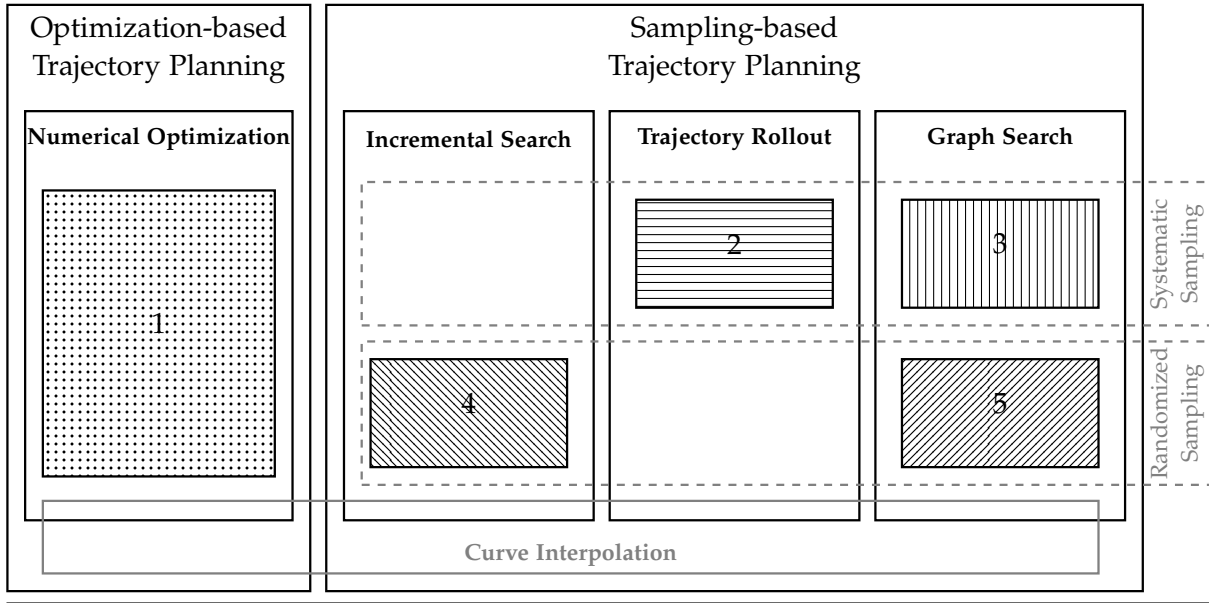
The taxonomy of trajectory planning approaches is not standardized and many contributions in the past have been designed for path instead of trajectory planning. Nevertheless, without regard to each individual case methods of path planning are included in the overview as the focus should be on the basic working principle and it is expected that path planning approaches can in general be adjusted to work for trajectory planning as well.

Katrakazas et al. (2015) tailor their survey about real-time motion planning methods to the functional architecture of an automated vehicle, suggesting three levels of planning. The proposed classification tree separates planning approaches with respect to their functionality in the hierarchical functional architecture and does not group the approaches by means of their key properties. Gonzalez et al. (2016) and Nilsson (2016) divide planning approaches into graph search, sampling based planners, interpolating curve planners and numerical optimization approaches, whereas Paden et al. (2016) for example categorizes into geometric methods, variational methods, graph search and incremental search. Beside different naming at least an evident overlap in the classification of planning approaches can be noted. In this thesis the terminology is chosen to reflect and emphasize the core differences between the approaches.

Figure 2.1 shows the consolidated taxonomy that should highlight the differences between the planning approaches with respect to their distinctive characteristics. Generally, trajectory planning approaches to automated driving can be divided into two basic categories: optimization and sampling based approaches. This distinction is emphasized by the fact that all sampling based methods can be represented as a graph generated from the discretized state space, whereas optimization based algorithms focus on (quasi-) continuous solutions. Optimization based trajectory planning mainly deals with approaches performing numerical optimization. Sampling based methods can be subdivided into trajectory rollout, graph search and incremental search. The class of trajectory rollout is introduced as a special case of graph search and accounts for the fact that trajectory samples are compared directly to each other, which renders graph search unnecessary. Hence, trajectory rollout is listed separately as graph search would be a misleading description for this concept.

Numerical optimization (also referred to as variational methods) is concerned with formulating the motion planning problem as an optimal control problem. Solutions in this context are given by optimal control theory. Numerical methods for optimal control are dynamic programming, indirect methods and direct methods (Diehl et al. 2005). Dynamic programming intends to find an optimal policy, which provides the

¹In contrast to off-road, which applies to navigation in unstructured environments.



representative contributions for respective field:






-  Werling and Liscardo (2012), Ziegler et al. (2014)
-  Montemerlo et al. (2008), Werling et al. (2012), Schwesinger et al. (2013)
-  Ziegler and Stiller (2009), McNaughton et al. (2011)
-  LaValle (1998), Karaman and Frazzoli (2011)
-  Kavraki et al. (1996), Karaman and Frazzoli (2011)

Figure 2.1.: Taxonomy of trajectory planning approaches to automated driving.

optimal control for every point in the state space. Dynamic programming is closely connected to *Bellman's Principle of Optimality* (Bellman 1957), which states that an optimal solution is composed of optimal partial solutions. A major drawback of dynamic programming is the computational burden, which rapidly increases with the order of the system, also referred to as *Bellman's Curse of Dimensionality*. Indirect methods use the necessary conditions of optimality of the infinite problem to derive a boundary value problem, which then has to be solved numerically. The class of indirect methods encompasses the calculus of variations as well as the *Pontryagin Maximum Principle*. Direct methods transform the optimal control problem into a finite dimensional nonlinear programming problem, which can then be solved by methods of nonlinear optimization. The resulting parameter optimization problem can be solved by convenient numerical solvers. The transformation from the original optimal control problem into a nonlinear program is also denoted as transcription. For the discretization of the optimal control problem with direct methods three methods are distinguished: direct single shooting, direct multiple shooting and direct collocation. Direct methods are frequently used for *Model Predictive Control* (MPC) that cyclically solve the optimal control problem. Starting with the current state observation as initial value the first part of the calculated optimal plan is applied to the real world system. As this procedure is repeated, feedback is generated that accounts for disturbances as well as for the

actual model-plant-mismatch. Further advantages are that MPC algorithms consider the system dynamics and constraints directly. As disadvantages the computational burden and non trivial stability proof can be listed.

Sampling-based approaches often make use of simplifications of the motion planning problem to meet requirements like for example the limited computation time as well as to facilitate the handling with constraints arising from the environment. The overall goal is to approximate a continuous space with a finite number of samples. Hence, sampling based methods tackle the motion planning problem by discretizing the state space in form of a graph. The graph represents a set of possible trajectory candidates and is generated by either sampling in action or state space. The best trajectory is then found by evaluation of each trajectory sample. Typically the trajectory optimization problem is relaxed to a search of possible (open-loop) solutions, which are checked for compliance with constraints afterwards. This certainly simplifies the motion planning problem, which is why sampling based methods are popular in automated driving applications.

The basic difference between the approaches within the class of sampling based planning methods lies in the way trajectories are sampled and in the way the best solution is chosen within the set of generated trajectories. First of all the trajectory set can be sampled in batch fashion or incrementally, which might also impact the choice of an appropriate graph type. In addition to this, for the sake of efficiency, the evaluation is mainly dependent on the graph representation. The term graph search refers to the fact, that for the purpose of exploring the environment the state space is divided into graph nodes or grid cells leading to a graph or lattice, which can subsequently be searched by graph search algorithms to find the minimum cost path through the graph. In this thesis trajectory rollout is defined as a special case of graph search algorithms. The main feature concentrates on the fact that a trajectory rollout approach is represented by a rooted tree. This obviates the use of graph search algorithms and leads to a pure comparison between complete trajectory samples. The distinctive property between graph search and incremental search algorithms is that for graph search the graph is built one step in advance and then performing a search in the full graph, whereas incremental algorithms explore the state space for the best solution while simultaneously expanding the graph by adding new states. Karaman and Frazzoli (2011) uses the terms batch and incremental to distinguish between graph and incremental search. Incremental search algorithms usually add new nodes if there exists a valid trajectory that connects the randomly sampled query node to the graph. Advantageously incremental planners are not required to connect two nodes exactly. Additionally, it is not necessary to define the number of samples a priori, as because of the incremental property the algorithm returns a solution as soon as one is found. The remaining time might be used to refine the solution by adding more samples to the graph.

The major challenge of sampling based methods in general is the task of adequately approximating the state space. This comprises the generation of expressive trajectory candidates to avoid analyzing samples with a very low probability of contributing to the optimal solution, not to mention the effort of approximating the optimal solution in the best possible way. Obviously, knowledge about the surrounding environment such as dynamic and static objects, as well as information about the course of the road

can contribute to a better sampling strategy. Moreover, it is easy to ascertain that a compromise between resolution of the discretized state space and the overall number of processable candidates has to be found. As the first aspect impacts the quality of the trajectory the latter is connected to the run time of the algorithm.

In contrast to several references curve interpolation is not listed as a method for trajectory planning. The reason why curve interpolation is not considered to be an unmitigated trajectory planning approach is the fact that it does not cover the generation of reference points, which are elementary in a planning concept. Curve planning can be utilized as an underlying layer in each of the mentioned planning categories. Clearly the sampling and the optimization based approach (cf. section 7.3 and section 7.4) presented in this thesis make use of an interpolation strategy leading to a spline based trajectory representation, which is explained in more detail in chapter 6.

2.2. Related Work for Automated Driving Applications

Perhaps the most difficult issue with respect to a survey of a field as diverse as trajectory planning is to restrict the scope of the survey to permit a meaningful discussion. This thesis will focus on approaches with sufficiently low run time suitable to be applied in a dynamic environment, ranging from urban to highway driving. Nevertheless, to account for the advantages of each class inferring suitability for a particular application and in order to sketch a comprehensive view of planning approaches, some considerable methods for different (parking) applications will also be mentioned in the corresponding sections. It is assumed that the planning algorithm is only relying on ego vehicle sensor data and on-board computational power excluding the possibility of data acquisition via V2X communication and distributed calculation of trajectories.

Numerical Optimization

Motion planning problems can be transformed to nonlinear optimization problems to make them accessible from the solver's point of view. The requirements on the desired trajectory are formulated in terms of an objective function and constraints. The resulting problem is to find a finite number of decision variables that minimize a cost function while satisfying equality as well as inequality constraints. The solution is typically dependent on an initial guess and might not converge to the optimal solution as the objective function and the set of constraints are generally not convex (for minimization problems). Especially for dynamic environments the concept of optimization leads to a more general solution of the planning and control problem. The works of Hilgert et al. (2003) and Sattel and Brandt (2005) for example show the capabilities of path optimization for automotive applications based on elastic band theory. Advancing further from this point Ziegler et al. (2015) describe how in 2013 a Mercedes-Benz S-Class prototype completed the Bertha-Benz-Memorial-Route, referred to as the Bertha Benz drive. The automated vehicle had to handle different traffic scenarios reaching from low to high difficulty and variability. Ziegler et al. (2014) explain the applied trajectory planning approach in detail. The optimization

problem is designed with a quadratic objective function constrained by nonlinear inequalities. The latter are composed of external (collision avoidance and driving corridor) and internal constraints (limits of vehicle kinematics and dynamics). To solve for the optimal trajectory a gradient based method namely *Sequential Quadratic Programming* (SQP) is used. Therefore, the constraints have been designed such that the solution converges to a single optimum, which then is supposed to be equal to the global optimal solution. A similar approach in the context of emergency situations is introduced by Keller et al. (2014). The approach differs to the previous mentioned one in the way that objectives are tailored to emergency situations instead of normal driving and in the way constraints are handled to formulate the optimization problem. Hard constraints are transformed and integrated into the objective function by means of terms that penalize the violation of the constraints. The nonlinear program is hence transferred to an approximate, non-restricted optimization problem, which then results in a nonlinear least-squares problem that has to be solved. Coming from the same root Ulbrich et al. (2017) and Ulbrich et al. (2018) present a similar approach tested on different scenarios.

For explicit consideration of vehicle stability and feasibility in the process of trajectory planning, knowledge of system dynamics in form of a vehicle model is needed. The explicit consideration of a vehicle model to perform trajectory generation is followed in (nonlinear) model predictive control approaches.

Werling and Liscardo (2012) propose a model predictive approach for combined steering and braking maneuvers. Nonlinear coupling effects in the vehicle dynamics are considered by a model that is subject to acceleration constraints, which thus limit the combined steering and braking force. As per design the approach always performs at these limits as the analyzed simulations on the scenario of pedestrian collision avoidance show. Further work in this field is provided by Bauer et al. (2012), who use a single-track model for trajectory generation combined with a subordinated tracking controller and Frasch et al. (2013) who introduce a nonlinear MPC approach using a double track model for autonomous vehicle guidance. Publications in this field have been made for emergency scenarios using a single and double track model, respectively (Götte et al. 2016d; Götte et al. 2015a).

Because maneuver and trajectory planning both tackle (different parts of) the motion planning problem these tasks are deeply interleaved. Hence, it is not surprising that several approaches present a coordinated solution for maneuver and trajectory planning. Nilsson et al. (2017) present an approach for lane change maneuvers that select an appropriate inter-vehicle traffic gap and generates a safe and smooth lane change trajectory by solving a quadratic program. Yi et al. (2019) find different driving maneuvers in terms of homotopy classes first (combinatorial framework) and then use model predictive methods to determine the optimal trajectory in each class. The final global optimal trajectory is then found by comparison among all classes.

Other interesting contributions in this field are presented by e.g. Qian et al. (2016), Miller et al. (2018), Pek and Althoff (2018), Bergman and Axehill (2018), Nietzschmann et al. (2018), and Rösmann et al. (2021). In previous work a combination of curve interpolation and numerical optimization for trajectory planning has been presented (Götte et al. 2017b; Götte et al. 2017c; Götte et al. 2017d).

The major advantage of optimization based approaches is the ability to find the optimal solution for various maneuvers even in complex scenarios. On the other hand optimization algorithms suffer from high computational burden.

Trajectory Rollout

As already mentioned trajectory rollout can be seen as a special case of graph search in which the graph, or even more specific a rooted tree (start state of the vehicle is given) is generated, such that the best solution is found by comparing each candidate trajectory to each other. In its simplest version of trajectory rollout the expansion is stopped after one step, leading to a tree of height one.

In the trajectory set generation phase, sampling is typically performed in action space by uniformly sampling across the range of possible inputs. The trajectory is generated with the assumption that chosen inputs are held constant over a certain future time interval. Still, it is also possible to define the set of trajectories via state space sampling. Either way both sampling strategies finally lead to an exhaustive search, which is dedicated to find the best solution. This additionally contains the task of checking each candidate with respect to the given constraints.

Montemerlo et al. (2008) roll out trajectories with lateral shifts with respect to the smoothed center of the lane. To generate the trajectory an internal vehicle simulation with different steering parameters is performed. Ferguson et al. (2008b) and Urmson et al. (2009) generate trajectories in model predictive fashion by connecting the initial state to a set of desired terminal states (position and orientation), that vary in lateral offset. The vehicle controls are parameterized with a time based linear velocity function and an arc-length based curvature function and numerical optimization for the curvature profile is conducted in terms of an iterative shooting method. Then the best trajectory is chosen according to an evaluation function.

Werling et al. (2010) and Werling et al. (2012) derive an optimal control based solution for the trajectory generation problem, that leads to quintic polynomials for an unconstrained problem minimizing the jerk. The strategy is to first generate lateral and longitudinal trajectory sets, which are subsequently combined by superposition. The generation of trajectories as well as the evaluation are dependent on the desired high level behavior. Another approach that performs trajectory rollout is presented in Schwesinger et al. (2013), wherein sample states are connected via forward simulation of a vehicle model in combination with a feedback control. This corresponds to the idea of closed-loop prediction that is also applied in the context of incremental search (cmp. *Closed-loop Rapidly-exploring Random Tree*) and returns drivable trajectories in terms of accounting for vehicle dynamics. Candidate trajectories are hence denoted as system-compliant. The advantage of closed-loop prediction can be summarized as good way to easily include system dynamics and such generating only candidates which are drivable. Generally, the approach of Schwesinger et al. (2013) could be categorized as a graph search algorithm, but just a single tree level was chosen as edges within deeper tree levels were marked as an approximation of future costs, but not contributing to the overall behavior due to replanning.

Instead of sampling in state space another way to generate candidate trajectories is

by sampling in action space, as for example assuming continuous curvature (Hundelshausen et al. 2008) leading to the *Tentacles* approach that is also addressed by Mouhagir et al. (2020), or by sampling control inputs of a vehicle model (Keller et al. 2015). Schlechtriemen et al. (2016) exploit the differential flatness of a single track model and generate jerk-optimal trajectory candidates. The tasks of maneuver planning and trajectory planning are deeply interleaved, because the trajectory set is generated by sampling in action spaces that are defined from the occupancies detected in the traffic scenario. For trajectory planning this means that constraints are generated in terms of checkpoints (breakpoints), which the vehicle has to pass through. Then jerk optimal interpolation in the flat output space is performed to generate the trajectories. Therefore, the start and goal state are transformed to the flat output space and interpolated along with each intermediate checkpoint. The evaluation takes then place in the state and input space.

Some approaches combine the trajectory rollout with subsequent optimization to benefit from the advantages of both methods (see e.g. Xu et al. (2012) and Kunz and Dietmayer (2016)). In connection with an underlying interpolation strategy an approach to trajectory rollout has been proposed that features online generation and evaluation of trajectory candidates (Lienke et al. 2018b).

Graph Search

Graph search methods discretize the state space into a grid to find the optimal solution to the motion planning problem. The planning problem is thus represented as a search for a solution in a known finite graph. Given is a graph that represents the problem and the task is then to find a path from start to the goal region with minimal costs (also referred to as shortest path). The expansion strategy is chosen to minimize the number of expansions required to guarantee that the optimal cost path is under the least-costs path examined. In contrast to uninformed search strategies that expand states in the order of the costs of the best path found so far from start state to the query state (e.g. Dijkstra (1959)), a heuristic function can be utilized to perform an informed search, where an additional estimate of the cost of a least-cost path from the query state to the goal state is considered (e.g. Hart et al. (1968)).

State-of-the-art graph search algorithms for automated driving applications share the basic principle and advance the basic ideas in different aspects. In particular approaches were extended to run in an anytime fashion in order to meet run time requirements or were particularly developed to deal with dynamic environments. Anytime planning algorithms like the *Anytime Repairing A** (ARA*) (Likhachev et al. 2003) try to find the best plan within the available amount of time. Therefore, they follow a two-step strategy: Find a fast and possibly highly suboptimal plan first and then improve the initial solution within the remaining time. The ARA* algorithm relies on the fact, that the environment does not change and hence the graph remains unchanged. In dynamic environments (or in general for imperfect environment information) the graph needs to be updated over time and a replanning is necessary. Algorithms like the *Anytime Dynamic A** (AD*) improve the solution over time (anytime property) and

repair its solution in case that parts of the graph are affected by changes (incremental² property). Likhachev and Ferguson (2008) and Likhachev and Ferguson (2009) show the application of AD* in several parking and driving scenarios. Dolgov et al. (2010) propose *Hybrid A** for planning in semi-structured environments (mainly navigating in parking lots) to account for the problem that because of their discrete nature, regular A* provide solutions that might not be executable for the vehicle. It improves on the A* algorithm by assigning a continuous vehicle coordinate to each discrete grid cell. Primary designed for path planning applications, but mentioned because of its popularity is the use of *Probabilistic Road Maps* introduced by Kavraki et al. (1996). They rely on a randomized sampling of graph nodes with each edge corresponding to feasible path between these. In the same spirit, but featuring a systematic sampling strategy is the application of state lattices, that build a directed graph in the vehicle's state space and solve a two boundary value problem to connect each state. Especially the latter task is closely connected to well-known optimal path planners like Dubins (1957) or Reeds and Shepp (1990) that provide the solution of the two boundary value problem, but do not take obstacles into consideration.

The yet mentioned approaches reflect the fact that graph search methods predominantly demonstrate their strength in semi-structured environments i.e. in parking applications. For on-road driving Ziegler and Stiller (2009) advance the idea of state lattices to trajectory planning approaches in dynamic environments by explicitly considering time, leading to the concept of spatiotemporal state lattices. They propose a lane adapted reparametrization of the space and generate a spatiotemporal state lattice, which connects each node with quintic polynomials. McNaughton et al. (2011) enhance the trajectory rollout approach of Ferguson et al. (2008b) to a graph based search in a spatiotemporal state lattice. A path is defined as a cubic polynomial spiral and a spatiotemporal lattice is generated to account for the dynamic environment. The problem of dimensionality resulting from the combinatorial way of sampling is addressed by proposing an efficient pruning of the search space.

For graph search approaches, which specify a trajectory planning problem that explicitly contains time information, it will generally result in a *Directed Acyclic Graph* (DAG). This is due to the fact that time is strictly monotonically increasing and connections that go back in time do not exist. The basic concept to find the shortest paths in a DAG is based on the idea to apply topological sorting in advance, such that the resulting algorithm is linear in time. This means that DAG shortest path algorithms perform faster than e.g. Dijkstra's algorithm (Cormen et al. 2009). McNaughton et al. (2011) argue that for automated driving applications an exhaustive search is necessary anyway, because the computation time should never exceed a certain threshold to meet real-time requirements. This should also hold for the worst case in which all nodes have to be expanded in order to find the optimal path. As for lattice planners in dynamic environments the occurrence of the worst case is very likely (a heuristic estimate can hardly be found) they conclude to use dynamic programming to search the entire graph. As in the case of trajectory rollout some contributions apply graph search with subsequent optimization (Hesse et al. 2010)

²In this context the incremental property is related to the capability of repairing a previous solution of a graph search and should not be confused with the class of incremental search approaches.

Graph based search algorithms allow for a simplified handling of constraints, but suffer from the fact that for a fixed resolution, the memory and computation time grow exponentially with the number of dimensions of the search space. Further drawbacks go along with the discretization of the state space as the choice of resolution will certainly lead to suboptimal results.

Incremental Search

Incremental search techniques rely on random or deterministic sampling to incrementally build a graph or tree in the state space. Algorithms in this class generally follow a three step procedure: Take a random sample from the free state space, find the nearest state in the current graph and apply a local planner to generate a connection to add the random sample (the last step can also be replaced by adding a new state in the direction of the random sample, as it is not required to connect two states exactly). Finally, this will result in an increasingly finer discretization of the state space, in which the solution can be found at the time a path through the graph connecting start state and goal region exists.

The most prominent approach representing incremental search is the *Rapidly-exploring Random Tree* (RRT), which has been introduced by LaValle (1998). In the context of automated driving Ma et al. (2015) apply an RRT based on various rule templates for different maneuvers and an aggressive expansion strategy. Kuwata et al. (2008) and Kuwata et al. (2009) introduce the *Closed-loop Rapidly-exploring Random Tree* (CL-RRT), which extends the original RRT approach by a low level controller to follow a given reference by forward simulation using a vehicle model. In a subsequent step the generated state trajectory is checked against constraints like obstacle avoidance. Another variant, the CL – RRT[#], has been developed and introduced by Arslan et al. (2017). The approach relies on the closed-loop dynamics to connect nodes and additionally ensures asymptotic optimality. The main drawback of the RRT approach is that it certainly is not going to return an optimal solution. Karaman and Frazzoli (2011) introduce a variant named RRT* (optimal RRT) which achieves asymptotic optimality, by additionally performing a rewiring of the tree after a new state has been added to the graph. In the field of automated driving several approaches using RRT* have been presented (e.g. Karaman et al. (2011) and Jeon et al. (2013)). The characteristics of incremental search support its application in parking scenarios, such that various publications have been made within this area (see e.g. Banzhaf et al. (2018a) and Banzhaf et al. (2018b)).

Incremental search algorithms tend to be probabilistically complete, which means that the algorithm will find a solution, if one exists, with probability approaching one for increasing computation time. In contrast to graph search methods incremental search planners do not require to connect two states exactly, which facilitates the construction of the graph. A major drawback of incremental and graph search methods is the significant increase of the computational burden with increasing dimension of the state space, while completeness (see A.1) cannot be guaranteed for real-time applications.

3

A Concept for Trajectory Planning in Dynamic Environments

The complex task of automated driving necessitates a precise design of the overall system, in which the part of trajectory planning has to be integrated appropriately. Because of the diversity and difficulty of the driving task the functional architecture of automated vehicles is mostly structured hierarchically. To define the interfaces for each module it is important to specify the basic requirements and capabilities with respect to the functional interaction leading to the desired synergy. Hence, it is crucial to adjust the capabilities of the planning module based on the quality and integrity of the provided data, when designing the concept for trajectory planning in dynamic environments.

3.1. Incorporation in the Functional Architecture of Automated Vehicles

Since the functional architecture is core to structure the technical development, as well as to divide the complex problem into several sub-tasks it directly impacts the overall performance of the system. The functional architecture of automated vehicles is not standardized, but still it is evident that it is in general composed of several basic components. Namely this is perception, planning and action. Because the characteristics of architectures are as diverse, this thesis waives to pinpoint the exact details of each component, but proposes a possible design of the system architecture focusing on a seamless integration of trajectory planning. A prototype of a functional architecture accounting for several aspects of the driving task is shown in Figure 3.1.

Although not shown in the figure it is worth to mention, that modules in the functional architecture of automated vehicles do not only work in feed forward fashion, but have to gainfully feed back their obtained information at characteristic stages. The example of a functional architecture of automated vehicles, consists of modules for perception, situation analysis, mission planning, decision making and trajectory planning as well as for vehicle dynamics control. The perception module generates a description of the actual traffic situation (environment model). The environment model incorporates

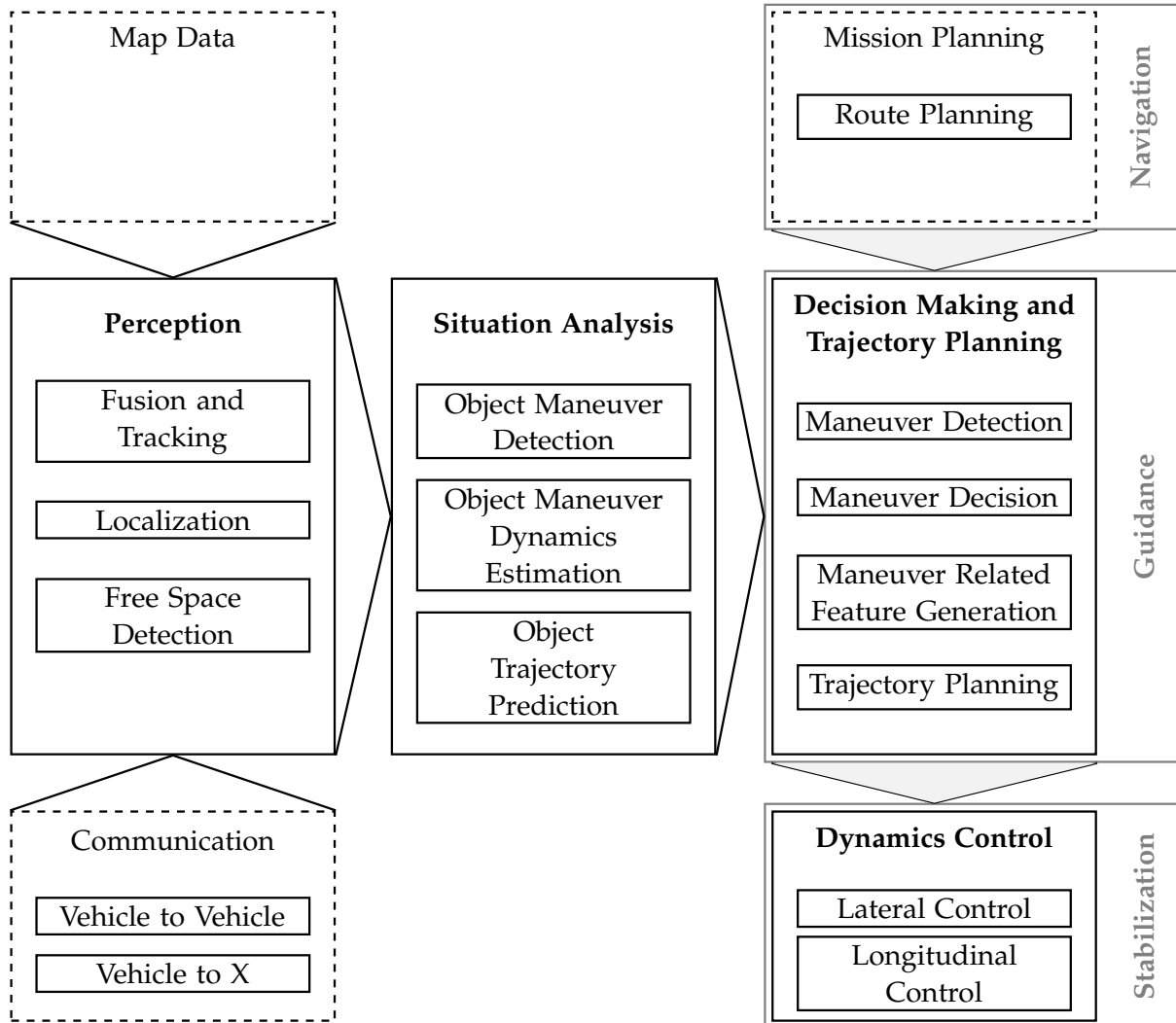


Figure 3.1.: Example of a functional architecture of an automated vehicle.

the estimated states of all obstacle vehicles and objects as well as a representation of the road topology. Map data and communication can rigorously support the data acquisition by contributing profound information. Situation analysis enhances the information with intention estimation of observed traffic participants and prediction of the traffic situation. The predicted trajectories complete the environment model to a predictive version that can be used by the maneuver and trajectory planning modules. During mission planning a static route is planned that neglects for example obstacle vehicles. In addition vehicle dynamics are not taken into account at that stage. During driving a decision making module manages upcoming events and infers appropriate maneuvers. At the trajectory planning level system constraints are taken into consideration, as well as other traffic participants are considered for the purpose of collision avoidance¹. Subsequently a reference trajectory is commanded to the low level vehicle controller, that takes care of realizing the planned trajectory. Because

¹Note that these characteristics also have to be considered in some way during decision making, which in general necessitates an interleaved design between decision making, situation analysis and trajectory planning.

of sudden changes in the environment frequent replanning is essential for safe and comfortable driving.

The main path of the shown functional architecture follows the principle of perception (pure acquisition of data), planning (situation analysis, decision making and trajectory planning) and action (vehicle dynamics control). The planning part has been subdivided to account for other-directed impact factors covered by the situation analysis on the one hand and to emphasize the self-determined part of decision making and planning on the other hand. This separation also allows to augment Figure 3.1 with respect to the hierarchy of driving tasks according to Donges (see e.g. Donges (2016)). A brief but more detailed description of the components of the functional architecture can be found in chapter A.2.

3.1.1. Prerequisites and Realization

The interconnection of components within the functional architecture builds up a complex system, that can be designed in many different ways revealing and emphasizing different characteristics of the automated vehicle performance. For a common understanding of the trajectory planning task, it is important to sketch a clear idea of the functional architecture in that respect.

Especially the complex task of decision making and trajectory planning is deeply interleaved. The problem is that a comprehensive decision making is dependent on detailed and precise modeling of all relevant aspects regarding the scenario assessment leading to the desired high accuracy and performance. This explicitly entails the task of an accurate trajectory planning with appropriate modeling of the vehicle dynamics. Although the number of potential maneuver classes in a scenario is finite, the number of maneuver trajectory candidates is infinite. Having in mind that computation time is limited for automotive applications, this fact leads to the conclusion that with nowadays hardware a hierarchical structure seems to be inevitable. In other words, the decision making is set up with maneuver modeling in a sufficient level of detail with some simplifications to match calculation time requirements, whereas trajectory planning as a subsequent module benefits from the global knowledge and implements the motion problem with a higher accuracy.

The trajectory planning algorithms presented in this thesis are developed under the following set of assumptions.

- The ego vehicle is equipped with a sensor system providing relevant data, such as e.g. relative positions and velocities of surrounding traffic participants and road boundaries.
- The ego vehicle is equipped with a situation analysis module that provides predicted trajectories of surrounding traffic participants. In this work the prediction of other traffic participants relies on the approach from Wissing et al. (2018).
- The ego vehicle is equipped with a decision making module that provides high level demands for the trajectory planner. The maneuver detection and the utility

function based maneuver decision own similar characteristics as the approach presented by Wissing et al. (2017) in the context of driver modeling.

- The ego vehicle is equipped with a low-level control system, which is capable of following the planned trajectory.

All these aspects are reflected in the given example of a functional architecture shown in Figure 3.1. In this work the process of decision making is separated into maneuver detection and decision. The maneuver detection mainly corresponds to a maneuver request and categorizes the desired maneuver based on the current traffic situation. Then the maneuver request is reviewed by means of a gap acceptance strategy to finally select a suitable maneuver class. Thus, the maneuver decision acts on the basic level of motion primitives such as lane keeping and lane changing. Because also more sophisticated maneuvers like overtaking can be decomposed into a set of these maneuver classes this interface is sufficient to manage even complex scenarios. Moreover, limitations arising from traffic rules, e.g. speed limits, stop signs or traffic lights are handled in this context.

3.1.2. Interfaces for Trajectory Planning

Based on the prerequisites mentioned before the interfaces between adjacent modules can be defined. With respect to the given example of a functional architecture illustrated in Figure 3.1 the question is how to generate a trajectory from the output of the maneuver decision and what is needed for the vehicle dynamics controller to work appropriately.

Interface to the Behavior Layer

The decision making within the functional architecture is designed in a way that it implements a general interface for the trajectory planning module. Beside many variants there are basically two options how the decision making output can be defined: Either the decision is propagated on maneuver class level, or a representative maneuver trajectory is provided. From trajectory planning perspective the favorable possibility is the generation of a maneuver trajectory, since this kind of pre-planning appends valuable information to the maneuver decision. Nevertheless, the developed trajectory planners should be able to deal with both kinds of input. This necessitates the implementation of an intermediate step to transform the maneuver decision to a suitable maneuver trajectory. This is achieved by means of the environment-aware maneuver planning described in chapter 4.

The input of the trajectory planning module is hence a maneuver trajectory, that functions as a representative prototype trajectory of the selected maneuver class. Furthermore, a spatiotemporal target region can be extracted, which involves topological knowledge about the environment. This resembles the demand on a desired maneuver providing a target position, orientation, velocity as well as further region-related information.

Interface to the Control Layer

The interface to the control layer is straightforward. It is assumed that a trajectory following controller is apparent, such that a trajectory is provided from the trajectory planner to the controller. The trajectory should primarily contain the coupled lateral and longitudinal position expressed in vehicle coordinates. Together with the inherent time information of the generated trajectory this allows to derive arbitrary reference signals (e.g. a reference velocity or reference yaw-rate) facilitating a variable controller design. For a high control quality a continuous description of the trajectory or at least a sufficiently high resolution in time and position is preferred. The control strategy can be designed as a fragment execution approach. The current trajectory is therefore considered as reference and the controller takes care about the realization of (at least some parts of) the trajectory. The sequence of trajectories does not necessarily contribute to an overall continuous driving, because deviations between the planned and realized trajectory are very likely to occur. The direct execution approach on the other hand could permanently provide updated trajectories to the controller. In this work the direct execution approach is chosen.

3.2. Requirements

The definition of requirements with respect to trajectory planning is twofold. First the requirements related to the planning approach itself have to be specified assigning the desired characteristics to the approach from development perspective. Secondly, demands on the characteristics of the trajectory in the context of automated driving need to be defined shaping the outlook of the planned trajectory that is directly coupled with the expected driving experience. The design of the trajectory planning approach should comply with the following considerations:

Requirements on the Trajectory Planning Approach

1. The requirement of temporal consistency accounts for the fact that the generated trajectory between two subsequent planning cycles should not change significantly, assumed that changes in the surrounding environment are consistent.
2. The approach should be complete, which means that the algorithm terminates in finite time and returns a valid solution to the trajectory planning problem or ascertains that no valid solution exists.
3. The approach should be real-time capable to be able to react to sudden changes in the perceived environment. This means that the computation time should be bounded from above in a way that it accounts for the challenges arising from the environment. An approximation of a suitable calculation time for one process cycle (including perception, situation analysis and decision making) can be derived from the dynamics of the environment. Kuwata et al. (2008) propose that the planning interval must be less than 0.1 s.

4. Arbitrary trajectory shapes should be realizable. This requirement is equivalent to the capability of covering driving tasks in static, unstructured environment as well as driving in dynamic and structured environment involving use-cases such as lane changing or lane keeping.
5. The approach should be optimal with respect to predefined criteria. The objectives should be derived from requirements on the desired ego trajectory for the respective situation. Additionally, the trajectory planning approach should be hybrid, which is defined as being aware of the global situation, while converging to a local minimum. This requirement is equivalent to finding the globally optimal solution.
6. The trajectory planning approach should be adjustable in a reasonable way. This requirement relates to the estimated application effort that is necessary to tune the developed approach in order to generate the desired behavior. Effectively, higher adjustability reflects a reduced effort for applying the proposed approach in practice.
7. It should be possible to integrate superordinate decisions in the trajectory planning process.

Requirements on the Trajectory

1. *Drivability*: The trajectory should consider nonholonomic constraints arising from the kinematics of the vehicle. This requirement is equivalent with the desire of a system compliant behavior and can be extended to the consideration of all kind of vehicle limitations.
2. *Collision Avoidance*: The trajectory should provably be collision free. This is especially addressing the capability of coping with dynamic traffic participants. Furthermore, safety distances should be satisfied in compliance with the passengers sense of security.
3. *High Level Behavior*: The trajectory should reflect the ability to drive in correspondence to the prevailing traffic rules (speed limits, lane discrete driving, etc.) and should be capable of reaching arbitrary goals (e.g. demanded maneuver). In particular two important functionalities should be covered: lane keeping with respect to lateral guidance as well as adaptive cruise control like longitudinal behavior.
4. *Comfort*: The trajectory should provide a comfortable drive for the passengers in the ego vehicle.

3.3. Derivation of a Trajectory Planning Concept

To derive a consistent concept the stated aspects with respect to the functional architecture and the requirements have to be merged into one comprehensive trajectory

planning framework.

To successfully accomplish the driving task a cyclic replanning is vital. This is first and foremost due to the dynamic character of the environment, since previous assumptions about the future development of the traffic situation might become invalid and need to be updated with the latest sensor data. Moreover, because of practical reasons the planning horizon is finite, which finally leads to a receding horizon approach that likewise relies on frequent updates. With permanent replanning there is a risk that due to e.g. suboptimality or unexpected events a discrepancy in consecutive planned trajectories arises, which might lead to undesired effects like overshoot or oscillations (Werling et al. 2012). This issue is addressed by the requirement of temporal consistency, because if temporal consistency holds then stability is implied and the resulting trajectory is independent of the replanning frequency. Whereas unexpected events (because of the uncertain prediction a certain unexpectedness is inherently expected) or wrong decisions aggravate the adherence of temporal consistency since they infer inconsistent restrictions, the aspect of optimality is explicitly considered in the design of the trajectory planning approach. This refers to the choice of an optimal function class, as well as to the overall optimization scheme. The necessary fundamentals are explained in chapter 6 that thoroughly investigate the choice of the trajectory representation in terms of optimality.

Overall, to solve the constrained trajectory optimization problem (see section 1.2) in accordance with the defined requirements on the trajectory this work relies on an optimization heuristic as a fundamental basis. The underlying idea is to perform a search within the set of optimal solutions to an unconstrained subproblem and choose the best solution of the associated high level problem², which fulfills the restrictions in order to derive a solution to the constrained problem. In principle the procedure of solving an unconstrained problem first and then complementing it by the consideration of restrictions in a second step is unique to discrete trajectory planning algorithms, but in consequence of the applied optimal interpolation strategy it is possible to transfer this idea to continuous optimization techniques. Hence, the proposed interpolation strategy paves the way to create a common basis for a generic trajectory planning approach to automated driving.

Finally, this leads to a two level hierarchical trajectory optimization framework in which the low level behavior is covered by the optimal solution of the unconstrained minimum kinematics problem, given by a spline-based trajectory representation with an optimal interpolation strategy. In order to satisfy the constraints like drivability and collision avoidance and to account for the desired high level behavior a superordinated optimization is performed. This way low level optimality with respect to basic driving functionality is assured, while simultaneously high level behavior is still prioritized. The separation into two different levels moreover addresses the problem of conflicting planning objectives and is completely compliant with the requirements on the trajectory. In particular, it is very difficult to define optimality in the context of the desired high level behavior. This is because of subjective assessment of driving maneuvers on the one hand and diverging interests in terms of maneuver demands on the other hand.

²The connection is established via the high level cost function, which is designed as a function of the low level solution.

A lane change for example might require a different set of optimality criteria than a lane keeping maneuver. By the application of the two-level hierarchical framework this conflict vanishes since an optimal low level behavior in terms of passenger comfort is always guaranteed by the optimal interpolation strategy, while the optimizer can solely take care to achieve the desired specifications given by high level maneuver requirements, covering a universally valid description of the automated driving task. Figure 3.2 shows the proposed concept of a two level hierarchical trajectory optimization framework. As a matter of fact relevant information in terms of the requirements

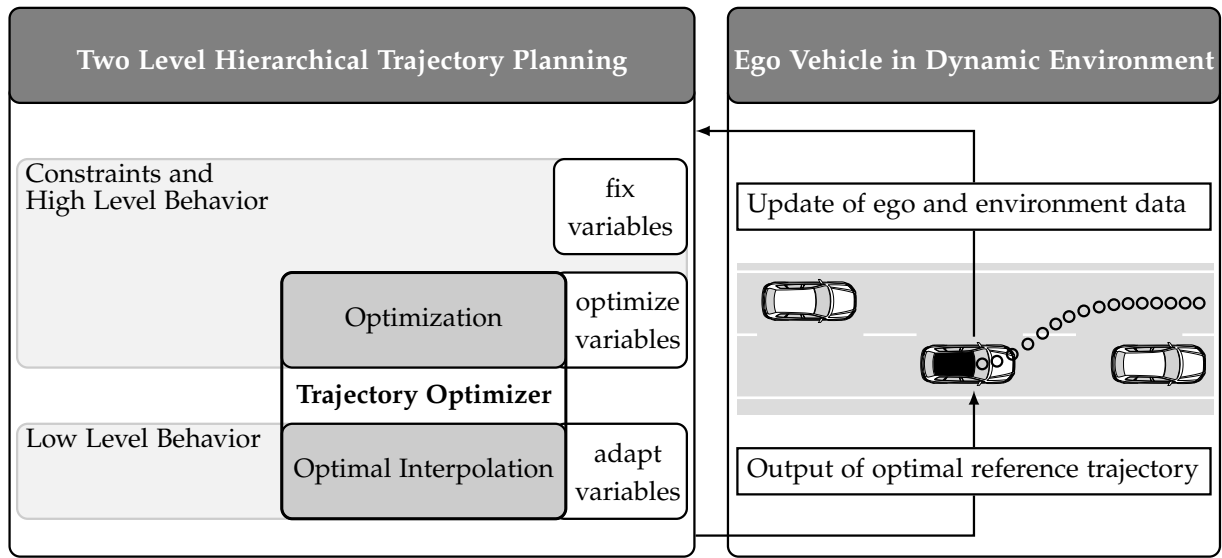


Figure 3.2.: Sketch of the proposed trajectory planning concept. Cyclic replanning in receding horizon fashion with the two level hierarchical trajectory optimization framework.

can be processed in three different ways, based on the prior knowledge about the desired trajectory. If for example some trajectory quantities can be directly inferred from the current traffic situation it is vain effort to search for the optimal solution in that respect. For this reason the developed trajectory planning approach allows to fix certain trajectory quantities (i.e. variables in terms of breakpoint elements) that are already known in advance. Remaining breakpoint elements can then be used as optimization parameters to shape the trajectory to comply to the desired behavior. As mentioned before a further subdivision is proposed to beneficially separate low and high level behavior. Low level decision variables hence adapt to the result obtained by optimization on high level. In case of the low level behavior a minimum kinematics trajectory generation approach is presented that relies on a spline-based trajectory representation in combination with an optimal interpolation. Details are given in chapter 6. To account for the high level behavior classical trajectory planning approaches are applied. In this thesis a discrete and a continuous approach are followed. A detailed description of the discrete trajectory optimization can be found in section 7.3 and for the continuous trajectory optimization in section 7.4. A separate spline for each direction is specified, yielding a lateral and longitudinal trajectory³.

³It is common practice to choose the number of separate splines in correspondence to the number of flat output variables of the system.

However, the evaluation of objectives and constraints is performed on the combined trajectory accounting for the coupled dynamics of the system.

The two level hierarchical trajectory optimization framework is seamlessly embedded into the functional architecture and intended to work in receding horizon fashion implying cyclic replanning. In general, to design the cyclic replanning strategy there are two options to choose the start state of the ego vehicle for the next planning cycle. The first option is to proceed from the state obtained from the previous planning solution. This enables the possibility to account for updates from the environment or new maneuver demands, however in this case feedback is realized on controller level. The overall performance (especially in terms of stability) is then mainly based on the subordinate trajectory following controller. The second option is to consider the updated measured quantities in the new planning cycle. Hence, at the start state deviations between the planned and realized trajectory introduced by inaccuracies of the low level controller will be eliminated as per design. Feedback is then realized on trajectory planning level likewise model predictive control approaches. In this work a similar variant of the latter approach is pursued.

The key features of the proposed approach can be summarized in the choice of the trajectory representation as well as in the design of the optimization scheme. A spline-based trajectory representation with optimal interpolation strategy allows for arbitrary shapes of the trajectory and also enhances computational efficiency. The optimization scheme is set up in order to find the optimal solution to the trajectory planning problem, leading to the concept of a two level hierarchical trajectory optimization. Beside other advantages the hierarchical framework promotes adjustability, since it enables the possibility to incorporate objectives such as comfort and high level decisions on different levels and allows to reduce the number of optimization parameters to the relevant parameters for the respective use-case. This leads to a highly versatile design that is capable to adaptively reshape in correspondence to demanded development features, resulting in a generic trajectory planning concept for automated driving.

3.4. Comparison to Other Trajectory Planning Approaches

Looking at the taxonomy of trajectory planning approaches (see section 2.1), there is numerical optimization, incremental search, trajectory rollout and graph search. In particular randomized, incremental search algorithms show some undesirable properties with respect to on-road driving, since randomized approaches do to a large extent not benefit from exploiting the structure of the environment. Moreover, it cannot be guaranteed that a solution is found within finite computation time. Since asymptotical optimality is also not an inherent feature of randomized approaches, the overall result might in addition be heavily suboptimal. The major disadvantage is the complexity that comes along with the dimension of time, which is vital for dynamic environments. This is because of the extension of the sampling space on the one hand that directly links to an increase of the computational effort, but it also complicates the connection between a node of the existing tree and the sampled query state on the other hand.

In the context of graph search algorithms a well chosen heuristic estimate will have a positive impact on the overall runtime of the algorithm. But in general such an heuristic that is valid for various scenarios is hard to find and worst-case assumptions should not be ignored as well. Furthermore, graph search algorithms are usually constructed to reach a fixed goal position. This is not necessarily the case for on-road driving scenarios, such that as a consequence a realization is complicated and might lead to undesired behavior. Jerky solutions for example possibly exclude drivability in terms of the non-holonomic constraints of the vehicle, but in any case decrease driving comfort. In that respect the application of trajectory rollout and numerical optimization approaches seems to represent the more promising way. A common approach to trajectory planning for automated driving in dynamic environments is the generation of multiple trajectories that fan out to cover the basic driving options of the vehicle. From this set the optimal trajectory in terms of a cost function is chosen. The popular work of Werling et al. (2010) for example is inspired by the theory of optimal control, in which the cost function is tuned to fit the subsequent trajectory selection process. It is shown that by this design desirable characteristics such as stability and temporal consistency are attained. Moreover, it enables a simplified handling of constraints. Polynomials are identified to represent the optimal function class in that respect (also see section 6.1). The usage of polynomials to represent longitudinal and lateral trajectories coincides with a trajectory rollout that stops expansion after the first step. In total this limits the general scope of maneuvers that can be performed, because it restricts the shape of the trajectories (e.g. a complete overtaking maneuver cannot be planned). For these kinds of maneuvers optimality cannot be declared. For this reason polynomial splines are applied in this work, allowing for arbitrary trajectory shapes and extending the scope of plannable maneuvers. Nevertheless, optimality is not neglected as the presented minimum kinematics trajectory generation features an optimal interpolation strategy on low level and on high level dedicated optimization approaches are applied. Therefore, it is assumed that a spline is capable of matching the optimal function class with respect to the high level problem. The adaptiveness of a spline suggests this capability, since the high level cost function merely concentrates on kinematic objectives, while trajectory smoothness induced by spline continuity conditions moreover relates to the non-holonomic characteristics of the ego vehicle. The work of Ziegler et al. (2014) in the context of the Bertha Benz drive represents one milestone in the history of automated vehicles. The approach basically follows the procedure of numerical optimization techniques. The decision variables are represented by longitudinal and lateral position and hence directly correspond to the discretized states of the trajectory. All objectives of the comprehensive objective function are handled on the same level balanced by manually adjusted weights. In this thesis the developed trajectory planning framework introduces a hierarchy that explicitly accounts for conflicting objectives, leading to less tuning effort. Moreover, as the trajectory is represented by a spline a significant reduction of the number of decision variables can be achieved. To sum up, from theoretical perspective the developed approach retains desirable features of well-known approaches, but advances the underlying ideas in some particular points. Namely this is the spline-based trajectory representation embedded in the proposed two level hierarchical framework.

4

Environment-aware Maneuver Planning

As mentioned before a major objective for trajectory planning is the realization of the desired vehicle behavior. In this chapter an environment-aware maneuver planning approach is described. Based on a high level decision a maneuver reference trajectory and spatial bounds are provided, which are used for trajectory planning in a subsequent step. For the sake of simplicity the level of detail with respect to the description of vehicle motion is reduced to the essential characteristics. Complex interdependencies in terms of the lateral and longitudinal vehicle dynamics are hence neglected. Calculations are done in curvilinear coordinates and obstacle vehicles are associated in a lane discrete fashion. The output can be used for initialization in optimization-based trajectory planning approaches or as a directive to generate the sampling set for discrete trajectory planners. Hence, it is worth to mention that this kind of pre-planning highly impacts the overall performance of the trajectory planning approach. For this reason and in order to give each investigated trajectory planning approach a common base a maneuver planning approach has been implemented in prototype fashion. The characteristics of the designed maneuver planning approach align with the proposed functional architecture given in section 3.1.

To show how the maneuver planning works, the approach is explained at the example of the situation depicted in Figure 4.1. The ego vehicle is surrounded by four obstacle

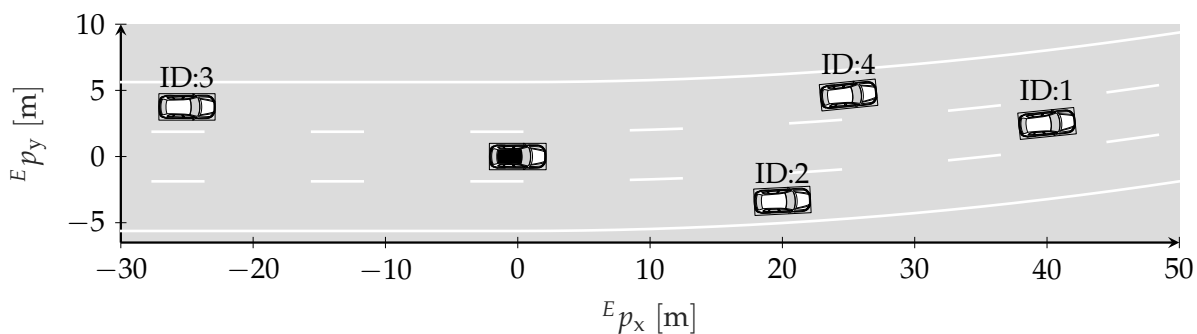


Figure 4.1.: Example situation to show the general functioning of the maneuver planning approach.

vehicles that are traveling along the road at different speeds. The start configuration of the test case is summarized in Table 4.1. In chapter A.3 another use case is shown.

Table 4.1.: Test case definition with start conditions of the ego vehicle and obstacle vehicles. The target lane is defined with respect to the ego vehicle, with 1 for the left lane, 2 for the center lane and 3 for the right lane. This means that the ego vehicle intends to perform a lane change to the left, whereas the obstacle vehicles stay on their respective current lane.

ID []	$^F p_x$ [m]	$^F p_y$ [m]	$^F v$ [km/h]	$^F \psi$ [rad]	target lane []	$^F v_{des}$ [km/h]
ego	0	0	100	0	1	130
1	40	2.5	100	0.1	2	not known
2	20	-3.35	90	0.05	3	not known
3	-25	3.75	110	0	1	not known
4	25	4.65	120	0.1	1	not known

Therefore, the test case definition is slightly altered leading to a cut-in and cut-out maneuver of obstacle vehicle ID:2 in front of the ego vehicle.

At first the surrounding environment has to be analyzed with respect to arising spatial limitations for the ego vehicle. Then the lateral and longitudinal maneuver trajectory can be calculated and finally be composed to the maneuver trajectory. As another result the maneuver planning further provides a target region.

4.1. Extraction of Maneuver Relevant Information

By considering only the relevant information with respect to the intended ego maneuver, the complexity of maneuver planning can significantly be reduced. Therefore, it is necessary to extract the maneuver relevant vehicles (which will be referred to as impeding vehicles) as well as the corresponding position bounds. The spatial relation between the ego vehicle and the ego maneuver relevant obstacle vehicles is summarized in the definition of impeding vehicles.

Impeding vehicles Impeding vehicles are classified with respect to the ego vehicle position as the closest obstacle vehicle on the respective lane. The closest obstacle vehicle to the front is denoted as lead, whereas the closest vehicle at the rear is denoted as tail vehicle. Hence, the definition covers the spatial relations between the ego vehicle and the other traffic participants in the scenario for each individual lane.

As the behavior decision is based on the current time step, the maneuver planning assumes that the current impeding vehicle constellation is binding and should also hold true for the future evolution of the scenario until a lane change of one of the obstacle vehicles is detected. This means that if a lane change is demanded, the current lead and tail vehicle on the target lane represent the spatial gap in which the ego vehicle should drive. The same holds for lane keeping as obviously in this case the lead vehicle limits the reachable space of the ego vehicle. In a first step other vehicles are predicted, such that the impeding vehicles, which are relevant for the intended maneuver can be identified. Then, in dependence on the desired maneuver, the position bounds for the

ego vehicle arising from the surrounding vehicles can be extracted.

Impeding Vehicles Prediction

The prediction of impeding vehicles is based on the lane association of the predicted obstacle vehicles (see section 3.1.1). The identification of future impeding vehicles is mandatory for predictive driving and enables the ego vehicle to handle for example cut-in maneuvers.

Normally, impeding vehicles are detected for the current time step based on the current constellation. With the aim to extract the impeding vehicles for future time steps the problem is that the constellation changes over time. This change is mainly caused by two reasons:

1. difference velocities between the other traffic participants that change their spatial relations to each other and the ego vehicle respectively,
2. lane changes of obstacle vehicles that lead to a different lane association.

It is obvious that the motion of the ego vehicle highly impacts the correct extraction of the impeding vehicles, as this metric is defined with respect to the position of the ego vehicle. The problem is that the future ego vehicle position is unknown at this stage of the maneuver planning. Based on the behavior plan it is not sufficient to assume constant velocity for the ego vehicle as this might be in contradiction to the assumption that the current impeding vehicle constellation is decisive as it reflects the conditions with respect to the desired maneuver. It could for example be necessary to accelerate to drive in the desired spatial gap on the target lane, but still the ego vehicle is not allowed to overtake the lead vehicle on the target lane.

It is important that the impeding vehicles reflect and correspond to the desired ego vehicle behavior. A consistent evolution of the situation has to be assured, which implies that the situation has to be evaluated for all behavior hypotheses of the ego vehicle (i.e. for all ego vehicle target lanes). It is presumed that the identified impeding vehicles constellation does not change until a lane change of one of the obstacle vehicles is detected. Until then the ego vehicle velocity is considered to be in the interval of the lead and tail vehicle velocity on the respective target lane¹. This measure contributes to the desired ego vehicle behavior to take care of realizing the decision to take the current gap on the target lane. The situation is reassessed when a lane change of one of the obstacle vehicles is detected. At that moment the process of determining the impeding vehicles is triggered again and it is decided if the lane changing vehicle merges in front or behind the ego vehicle and thus represents a potential lead or tail vehicle. The detection of impeding vehicles at respective lane change times can be regarded as a new decision that has not been made in advance, but approximates the development of the situation reasonably well.

For the considered example the predicted impeding vehicles are shown in Figure 4.2

¹It is assumed that the ego vehicle velocity at the predicted time step is independent of the past time steps.

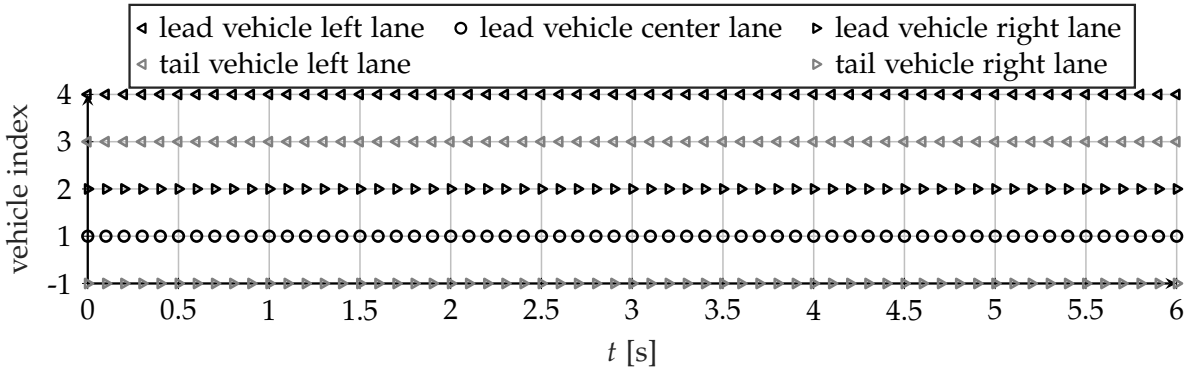


Figure 4.2.: Predicted impeding vehicles shown for the considered example test case. In correspondence to the selected test case (obstacle vehicles do not change lanes) the assignment with respect to lead or tail vehicle for the respective lane does not change. As no tail vehicle for the right lane exists it is mapped to a vehicle index of -1.

It is assumed that the tail vehicle velocity is less than or equal to the lead vehicle velocity on the target lane (which is a rather evident assumption as the tail cannot overtake the lead vehicle). Still, this shows that in general it is preferable to consider the interaction of obstacle vehicles during prediction. Furthermore, the relation to another vehicle in the future depends on a rough estimate of the ego vehicle velocity and the correctness of the impeding vehicles prediction is thus subject to high uncertainty. To cover this kind of problem a more advanced approach is necessary, most likely requiring a description in probabilistic fashion. Still, for most of the scenarios encountered in dynamic traffic the presented impeding vehicles prediction provides accurate results and definitively meets the requirements within the scope of maneuver trajectory generation.

Extraction of Ego Maneuver Position Bounds

From the detected lead and tail vehicles on each lane the resulting consequences for the ego vehicle are derived. This includes the consideration of the desired ego vehicle behavior determined during the stage of decision making.

To break the variety of possible situations down into a generalized description, the relevant information for the ego vehicle with respect to the current and desired target lane is merged into one representation. This is achieved by means of a lower position bound, an upper position bound and a lane change initiating vehicle position. The latter serves the purpose of providing an indication for the lateral motion of the ego vehicle maneuver and adds relevant information from the current ego lane, whereas the lower and upper position bound refer to the target lane. Generally speaking, based on the desired ego maneuver and the previously determined predicted impeding vehicles, up to three relevant obstacle vehicles are extracted. In case of a lane change two vehicles on the target lane, which define the merge-in gap and the lead vehicle on the ego lane are considered. In lane keeping case the number of considered obstacles reduces to the lead vehicle on the ego lane. To get the ego maneuver position bounds the relevant impeding vehicles are selected based on the desired ego vehicle behavior

(i.e. the ego vehicle target lane) and matched to the corresponding position of the respective obstacle vehicle. This means that for every predicted time step the identified predicted impeding vehicles position is extracted. Additionally, in lane change state, the lane change initiating vehicle is of importance as it defines the position in time at which the ego vehicle should reach the center of the target lane. The latter particularly addresses the requirement to prevent a collision with the lead vehicle on the initial ego lane during lane change. For the considered example the ego maneuver position bounds are shown in Figure 4.3. As the ego vehicle intends to perform a lane change

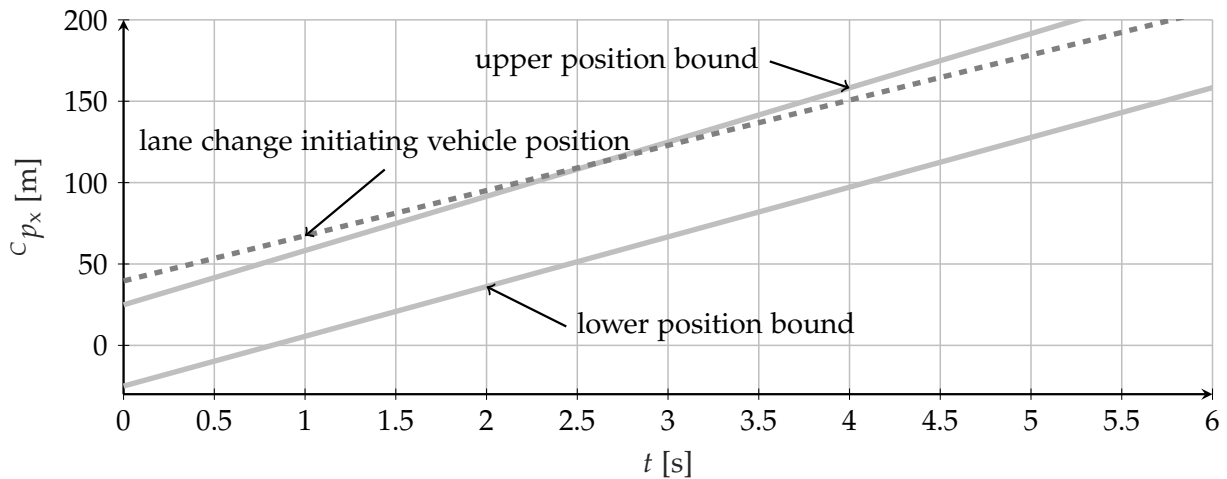


Figure 4.3.: Merged relevant information for the ego vehicle. The lower position bound arising from obstacle vehicle ID:3 on the target lane, the upper position bound arising from obstacle vehicle ID:4 on the target lane and the lane change initiating vehicle position arising from obstacle vehicle ID:1 on the ego lane.

to the left the relevant impeding vehicles on the target lane provide the upper (obstacle vehicle ID:4) and lower position bound (obstacle vehicle ID:3). The lead vehicle on the ego lane determines the position at which the lane change should be finished (lane change initiating vehicle position given by obstacle vehicle ID:1).

The maneuver planning approach also covers the case of overtaking by applying the simplified assumption that an overtaking maneuver has to be performed if oncoming traffic on the target lane is detected. It should be noted that in this thesis the case of planning a complete overtaking maneuver as such is considered as a special case, since in general it demands different requirement specifications² like e.g. a longer planning horizon (also cmp. section 3.2). Still, some ideas how to deal with this type of maneuver are given. This mainly relates to the definition of the maximum pass time for longitudinal and the target lane time zone for lateral maneuver planning (see section 4.3).

The maximum pass time is particularly defined for the case of overtaking and affects the time at which the lower position bound becomes active. It is considered to be the point in time when the ego lead vehicle position and target lane lead vehicle position intersect. If no intersection is discovered the maximum pass time is set to

²A change in the requirements with respect to the scope of the trajectory planning approach will also affect the design of the functional architecture.

be the prediction horizon of the maneuver planning approach. From the variety of possibilities only one relevant case with respect to overtaking is considered during maneuver planning. In this case the distance of the approaching target lane lead vehicle is larger than the distance to the ego lane lead vehicle. Then the lane change initiating vehicle position is set to be the ego lead vehicle position. The lower position bound is given by the ego lead vehicle position activated after the maximum pass time. With the assumption that the overtaking maneuver is completed before the maximum pass time, an upper position bound is not necessary. Table 4.2 gives a summary of the relation between the ego maneuver relevant position bounds and the respective impeding vehicle.

Table 4.2.: Summary of ego position bounds with respect to the intended ego maneuver and driving direction on the target lane.

oncoming traffic on target lane	ego vehicle maneuver	upper position bound	lower position bound	lane change initiating vehicle
no	lane keeping	ego lane lead vehicle	ego lane tail vehicle	-
	lane change	target lane lead vehicle	target lane tail vehicle	ego lane lead vehicle
yes	overtaking	-	ego lane lead vehicle	ego lane lead vehicle

4.2. Longitudinal Maneuver Planning

After the impeding vehicles and the respective ego maneuver position bounds have been identified a representative longitudinal prototype trajectory as well as a range of validity are derived in correspondence to the desired ego maneuver. To plan the longitudinal maneuver trajectory all states are given in curvilinear coordinates and thus describe the progress of the ego vehicle and other obstacle vehicles along the road. For a given behavior it is now of interest to determine the longitudinal target region (i.e. spatial gap between the lead and tail vehicle on the target lane) and the desired longitudinal position for the ego vehicle over time.

To determine the longitudinal target region three aspects are taken into account: limits arising from the ego vehicle system dynamics considered in terms of drivability (drivable region), spatial limitations arising from other surrounding traffic participants (admissible region) and the consideration of a safety distance to the vehicles on the proposed target lane (unimpeded region). In a subsequent step the ego demands in terms of a desired reference velocity are taken into account for longitudinal reference generation. The process consists of three steps: calculation of a reference position in order to reach the desired velocity (reference states generation), calculation of a reference position constrained by other traffic participants (bounded reference states generation)

and the calculation of an optimal reference position based on a least squares formulation subject to kinematic constraints (optimal reference states generation).

Drivable Region

The drivable region corresponds to the reachable space, that is identified by assuming maximum acceleration and deceleration of the ego vehicle, respectively. Furthermore, the drivable region is bounded by the maximum and minimum velocity. It is intended to cover a coarse description of the ego vehicle dynamics regardless of constraints arising from the current surrounding environment. To account for the torque-speed characteristic (that is proportional to the acceleration-velocity characteristic) of the vehicle the maximum acceleration is described as a function of the current velocity (cmp. Figure 7.2). The maximum deceleration remains at constant -9 m/s^2 for positive velocities. For the considered example the drivable region is depicted in Figure 4.4. The lower region bound corresponds to a full braking maneuver bringing the vehicle to a standstill, whereas the upper region bound reflects the position that is reached with maximum acceleration.

Admissible Region

The admissible region includes the constraints arising from the dynamic environment. It hence limits the space accounting for obstacles and a desired longitudinal stop position, respectively. Figure 4.4 shows how the admissible region is derived. Therefore, the drivable region is narrowed by the maneuver relevant lead and tail vehicle position. For the given example the drivable region is only constrained by the lower position bound as the upper position bound is larger than the upper bound of the drivable region.

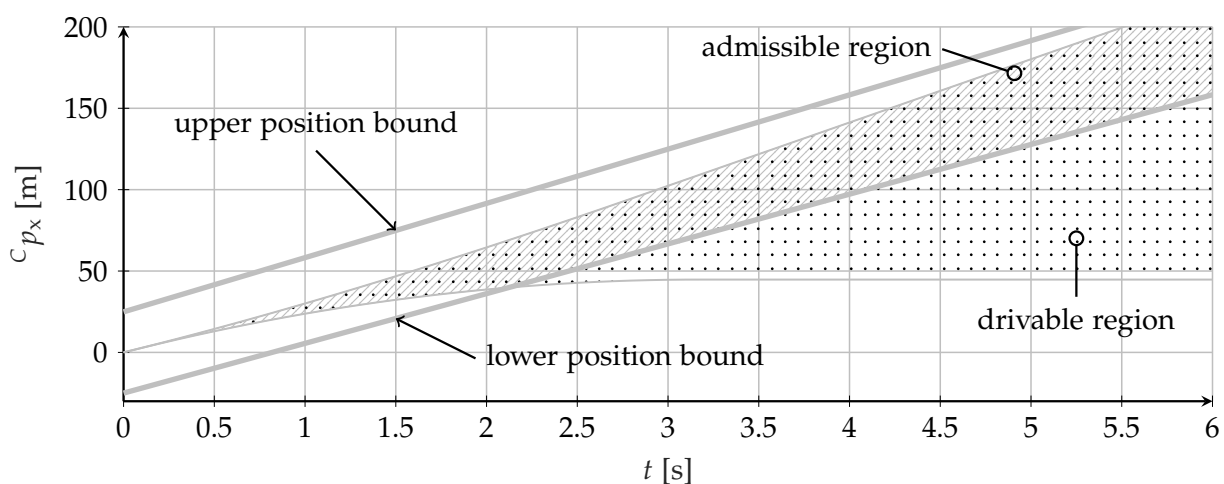


Figure 4.4.: The drivable region and the upper (derived from the lead vehicle on the target lane) and lower position bound (derived from the tail vehicle on the target lane) constitute to the admissible region. Note that the drivable region fully contains the admissible region.

Unimpeded Region

As it might be desirable to maintain a certain safe distance to other vehicles, the unimpeded region marks the area in which the ego vehicle can drive freely, without being disturbed by other traffic participants. Hence, the unimpeded region further narrows the admissible region as a safety distance to the respective lead and tail vehicle on the target lane is taken into account. Safety distances are derived in ACC fashion using a *Constant Time Gap Policy* (see e.g. Swaroop and Rajagopal (2001)) to describe the safety distance to the lead and tail vehicle. A lead and tail vehicle safe position are introduced, which represent the respective position bound for the ego vehicle arising from the desired safety margin to the lead and tail vehicle, respectively. For the considered test case the result is depicted in Figure 4.5.

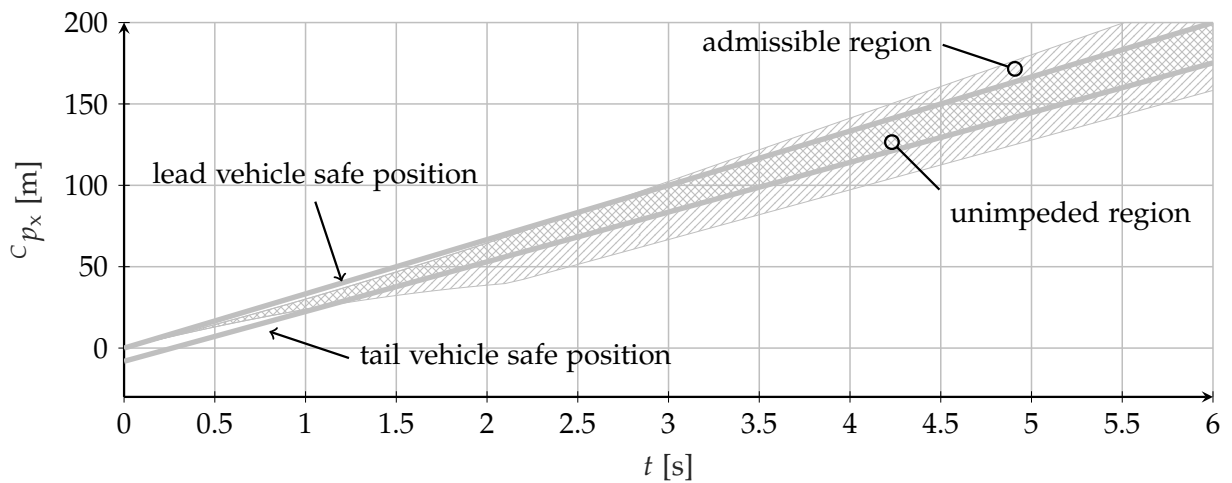


Figure 4.5.: The admissible region and the lead and tail vehicle safe position, which account for the desired safety margin to the lead and tail vehicle, constitute to the unimpeded region.

Reference States Generation

Yet, the available space has been defined and vehicle limits, limits from the dynamic environment and maneuver demands are already considered. In a first step the desired ego vehicle behavior is defined without consideration of the other traffic participants, but only acting in self-interested sense. The reference states are calculated by intending to reach the set speed. A predefined comfortable acceleration is used and by integration the corresponding position is obtained. Then the procedure is as follow:

- Identify the desired acceleration by checking if the ego vehicle has to accelerate or brake to reach the set speed. Apply a predefined comfortable value for acceleration \ddot{a}_x^+ and deceleration \ddot{a}_x^- , respectively.
- Integrate the constant desired acceleration to calculate the reference velocity. Limit the desired acceleration in dependence on the velocity to take the vehicle limits into account (cmp. Figure 7.2).

- Bound the reference velocity to the desired set speed and the vehicle’s speed limit.
- Integrate the reference velocity to obtain the corresponding reference position.

The reference states generation hence provides a reference longitudinal position and a reference velocity.

Bounded Reference States Generation

In a second step limitations from the system dynamics and the environment are considered. As depicted in Figure 4.6 the reference position is therefore constrained by the unimpeded region (system limits and safety distance to predicted obstacle vehicles position) yielding a bounded reference position. Moreover, the bounded reference states generation provides the bounded reference velocity (see Figure 4.7), which is derived from the bounded reference position by taking the difference quotient. The bounded reference velocity is then limited in a way that it prevents the ego vehicle from driving faster than the reference velocity or driving backwards. It is furthermore

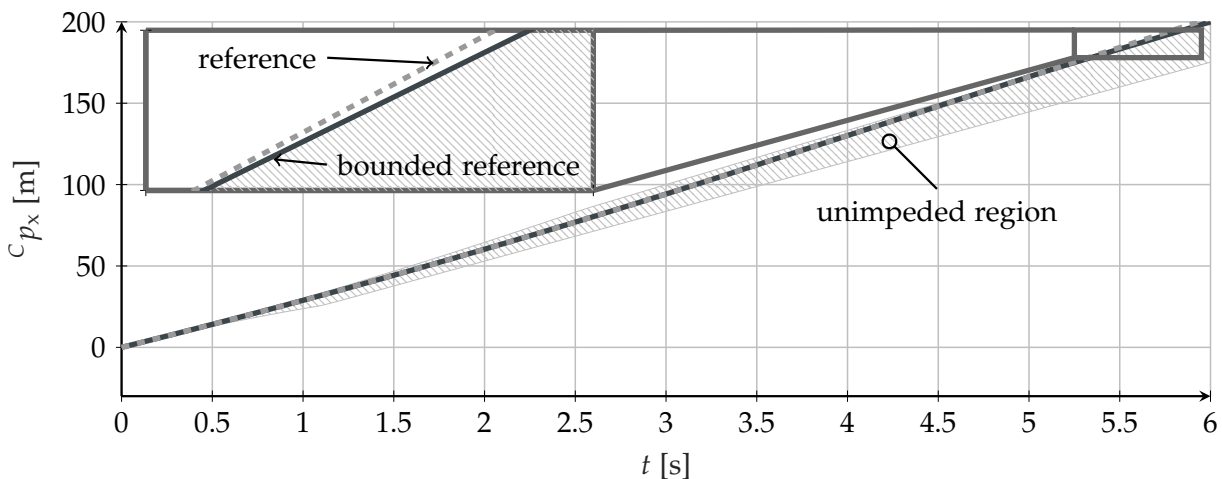


Figure 4.6.: The reference position is obtained by integration with the aim to reach the set speed with a desired comfortable acceleration. The reference position is then bounded by the unimpeded region yielding the bounded reference position.

assumed that the prevailing speed limit has already been considered in the stage of set speed determination.

Optimal Reference States Generation

In the third and final step the optimal reference position is calculated by formulating a constrained least squares problem, in which a least squares fit on the bounded reference velocity is performed. The solution is obtained by solving a quadratic programming problem subject to linear constraints on the position, velocity and acceleration.

The problem with respect to the decision variables vector $\hat{\mathbf{p}}$ reads:

$$\min_{\hat{\mathbf{p}}} 0.5\hat{\mathbf{p}}^T\mathbf{Q}\hat{\mathbf{p}} \quad (4.2.1a)$$

$$\text{s.t.} \quad \begin{bmatrix} \mathbf{v}_{\text{des}} \\ \mathbf{p}_{\text{min}} \\ \mathbf{v}_{\text{min}} \\ \mathbf{a}_{\text{min}} \end{bmatrix} \leq \begin{bmatrix} \mathbf{A}_{\text{lsq}} \\ \mathbf{A}_{\text{pos}} \\ \mathbf{A}_{\text{vel}} \\ \mathbf{A}_{\text{acc}} \end{bmatrix} \hat{\mathbf{p}} \leq \begin{bmatrix} \mathbf{v}_{\text{des}} \\ \mathbf{p}_{\text{max}} \\ \mathbf{v}_{\text{max}} \\ \mathbf{a}_{\text{max}} \end{bmatrix}. \quad (4.2.1b)$$

The desired velocity \mathbf{v}_{des} is the bounded reference velocity and subject to the least squares formulation. The maximum and minimum position and velocity bounds are derived from the admissible region, with the maximum and minimum acceleration given by configurable parameters and set to a comfortable acceleration \ddot{a}_x^+ and a braking limit \ddot{a}_x^- . Matrices \mathbf{Q} and \mathbf{A}_{lsq} constitute the least squares problem, whereas matrices \mathbf{A}_{pos} , \mathbf{A}_{vel} and \mathbf{A}_{acc} contain finite difference approximations of the derivatives to account for the respective kinematic constraint. The detailed structure of the matrices is given in section A.3.

Figure 4.7 shows the velocity obtained from the bounded reference generation (bounded reference velocity) as well as the solution from the formulated constrained least squares problem (the optimal reference velocity is given by differentiation of the optimal reference position). It can be seen that the ego vehicle accelerates from its initial velocity of 100 km/h to the desired velocity of 130 km/h. Because of the slower lead vehicle on the target lane the ego vehicle finally has to adapt to a speed of 120 km/h.

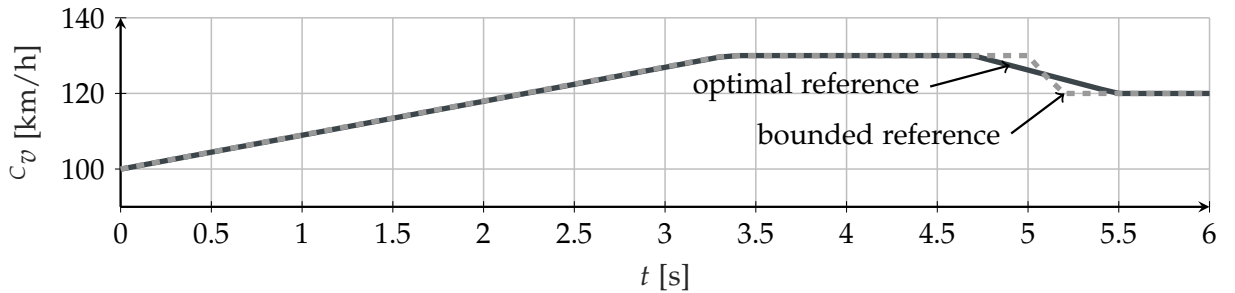


Figure 4.7.: The optimal reference position is calculated by a least squares fit on the bounded reference velocity subject to position constraints (from admissible region), velocity constraints (from drivable region) and acceleration constraints (comfort acceleration/deceleration).

4.3. Lateral Maneuver Planning

The lateral maneuver planning is performed subsequent to the longitudinal planning and hence is restricted to the predefined longitudinal motion. The task of lateral maneuver planning can be seen in the generation of a corresponding lateral motion that does not interfere with the goal of the maneuver planning. This means that the only requirement is not to collide with other obstacles, which should be ensured by the time the ego vehicle reaches the target lane. This could be generalized to a time

zone in which the ego vehicle is supposed to drive on the target lane, denoted as the target lane time zone.

Target Lane Time Zone

The target lane time zone defines the start and end point in time between which the ego vehicle should be located on the selected target lane. This definition also generalizes to the special case of planning a complete overtaking maneuver (cmp. Extraction of Ego Maneuver Bounds in section 4.1). The start point is inferred from the positions obtained by the longitudinal maneuver planning. Therefore, the ego vehicle reference position is compared to the lane change initiating vehicle position to identify the point in time at which the ego vehicle should be on the target lane. The point of intersection of the bounded optimal reference position and the lane change initiating vehicle position then marks the start time of the target lane time zone. For lane changes the end time of the target lane time zone equals the planning horizon, whereas for the special case of an overtaking maneuver the end time would be below (set in dependence on the desired distance to the lane change initiating vehicle when driving back to the origin lane).

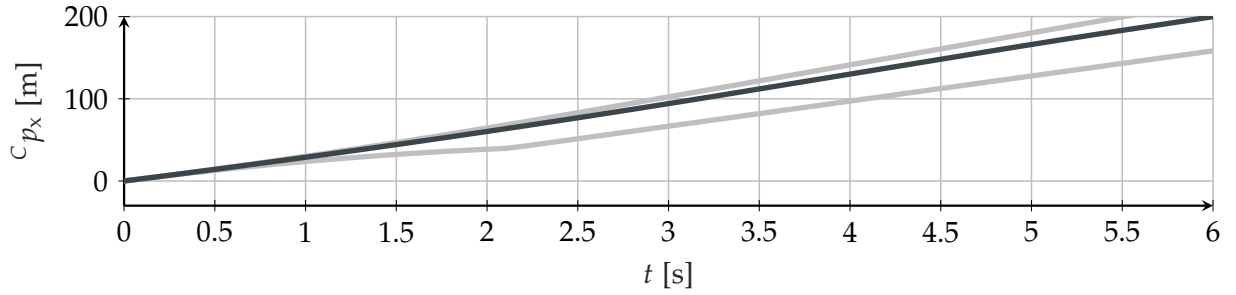
Lateral Reference Trajectory Generation

The application of curvilinear coordinates drastically simplifies the calculation of the respective lateral boundaries to determine the desired lateral translation. The lateral motion is divided into three parts, in which the first one moves the ego vehicle from the actual to the desired lateral offset derived from the target lane. The second and third part are designed in correspondence to the target lane time zone. Hence, for lane changes both parts adhere to the lateral position of the target lane center, whereas for the special case of overtaking the third part moves the ego vehicle back to the origin lane. The resulting lateral position profile is interpolated linearly in each of the three cases. In this context it should be noted that the final maneuver decision of the superordinated decision making module is assumed to be imminent, which excludes delayed lane changes after an earlier lane keeping phase for target gap alignment.

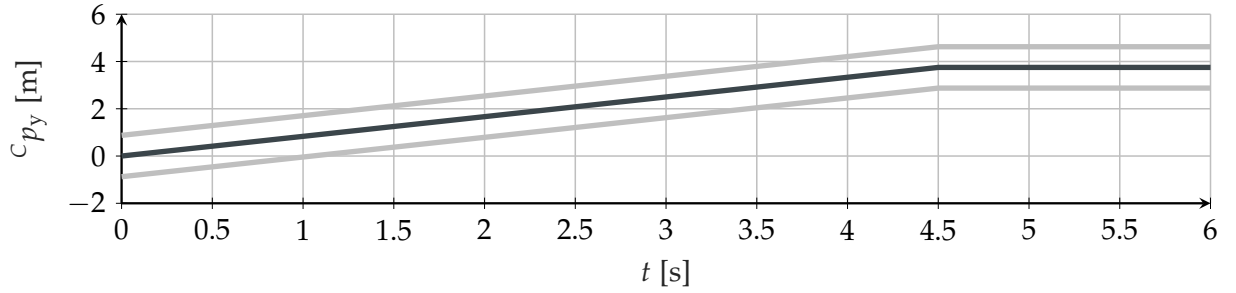
4.4. Maneuver Trajectory Generation

The maneuver trajectory is finally composed of the longitudinal (corresponds to the optimal reference position) and lateral reference trajectories, also fusing the information about the respective target regions. Figure 4.8 shows the result obtained for the considered example.

The coupling between lateral and longitudinal planning is accomplished by the lateral target lane time zone, which permits to consider only the lead and tail vehicle on the target lane for longitudinal planning. One problem with respect to this might be that as the longitudinal planning determines the target lane time zone for the lateral planning and no feedback is implemented it could lead to unfeasible low lane change times. It is therefore assumed that the decision algorithm already roughly covers the



(a) Longitudinal maneuver trajectory and longitudinal target region.



(b) Lateral maneuver trajectory and lateral target region.

Figure 4.8.: Final maneuver trajectory and target region of the maneuver planning approach.

characteristic of a lane change. Finally, the subsequent trajectory planning algorithm will take care of precise, combined lateral and longitudinal planning with respect to all obstacles. As already mentioned in section 4.1 especially the prediction of the future development of a situation is dependent on many impact factors. This aggravates the deterministic representation and requires a probabilistic description of the evolution of the situation to improve on the maneuver planning output. A possible alternative that brings maneuver planning to a wider scope is given by Schmidt et al. (2019b). In the work at hand the purpose of the maneuver planning approach is the derivation of an initial trajectory³ as well as the definition of a sampling region. For this purpose it can be concluded that the developed environment-aware maneuver planning approach is entirely suitable.

³Especially for the task of initialization it is mostly important to find the correct homotopy class.

5

Modeling and Objectives for Trajectory Planning

For trajectory planning in the context of automated driving several aspects have to be taken into account. Key properties encompass the consideration of system dynamics and the surrounding environment. In this chapter the modeling of the ego vehicle motion and the modeling of the environment are introduced in detail. Moreover, this chapter will elaborate on the trajectory planning objectives.

5.1. Modeling of Vehicle Motion

To describe vehicle dynamics there exist a lot of various options. In previous work (Götte et al. 2016a; Götte et al. 2016b) the suitability of vehicle models with respect to emergency steering maneuvers has been analyzed. In this work the focus is on normal driving situations, which poses different requirements to the modeling of lateral and longitudinal vehicle dynamics. It is assumed that the complex vehicle dynamics is addressed by the low level controller. Nevertheless, it is essential that the trajectory planning approach provides a roughly drivable trajectory, meaning that vehicle constraints are thoroughly considered.

5.1.1. System Modeling

The driving characteristics of a vehicle are typically split in longitudinal, vertical and lateral dynamics. In order to derive an adequate theoretic vehicle model for trajectory planning especially the longitudinal and lateral dynamics have to be considered. Whereas longitudinal dynamics can be captured on kinematic level with sufficient quality, the lateral dynamics, which play an important role in cornering, can be modeled at different levels of complexity rigorously affecting the overall performance of the vehicle model. The modeling of lateral dynamics involves the challenging aspect of capturing the non-holonomic constraints of the vehicle. As the description of lateral dynamics is crucial they will be introduced in more detail.

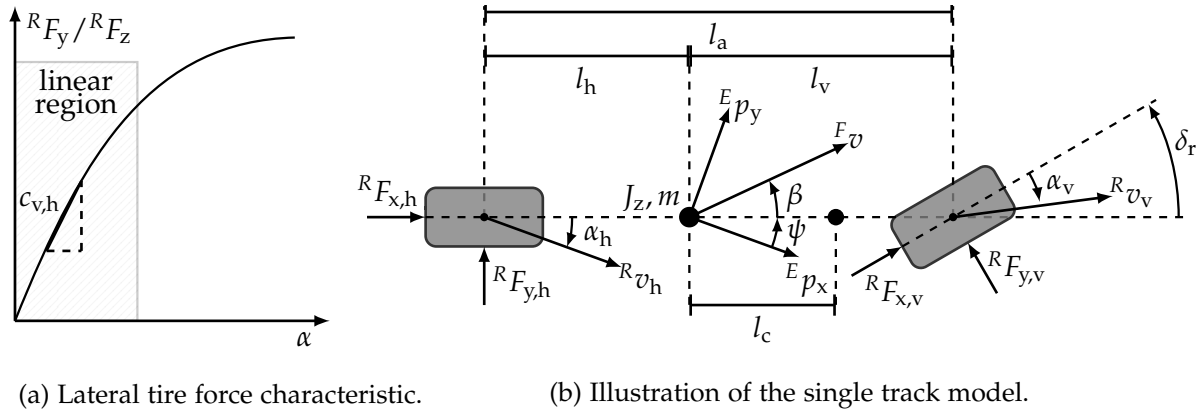


Figure 5.1.: Major components for the modeling of vehicle dynamics.

Lateral Tire Forces

The interaction between tire and roadway is an important aspect to be considered in the modeling of vehicle dynamics, since the contact between vehicle and roadway represents the primary possibility to actively affect the motion of the vehicle. For the purpose of vehicle modeling the complexity varies from simple linear to complex nonlinear models. To generate a lateral force a tire slip angle α is necessary, which results if the tire's direction of motion is different to the rolling direction. In general the tires of the vehicle follow a nonlinear characteristic, but in dependence on the application simplifying assumption can be made. For normal driving situations it is reasonable to use a linear tire model that well describes the vehicle behavior in the linear area of the tire characteristics (see Figure 5.1a).

In the linear model the lateral tire force is proportional to the tire slip angle. With the assumption of small tire slip angles $|\alpha| < 4^\circ$ (see Meywerk (2015, p. 180)) the linear tire forces at the front and rear axle read:

$${}^R F_{y,v} = c_v \alpha_v, \quad (5.1.1)$$

$${}^R F_{y,h} = c_h \alpha_h. \quad (5.1.2)$$

For more insight in the world of tire modeling providing a comprehensive description of the lateral and longitudinal tire behavior, especially for tire models covering the nonlinear region, see Pacejka (2007) and Schramm et al. (2018).

Equations of Motion of the Single Track Model

The single track model enables comprehensive analysis of the vehicle motion and allows to draw elementary conclusions on the lateral vehicle dynamics. Since it offers an excellent compromise between modeling effort and model accuracy, its use is widely spread among several application areas.

To derive the equations of motion for the single track model it is basically assumed that the vehicle's center of gravity lies in the plane on which the vehicle is traveling along the trajectory, e.g. road level. By this the wheel load transfer is neglected and the front and rear wheels are consolidated to one single wheel at the front and rear

axle respectively. Figure 5.1b shows a schematic sketch of the single track model. The vehicle mass m and the moment of inertia around the vertical vehicle axis J_z are combined in the vehicle's center of gravity. The geometric parameters l_v and l_h describe the distances from the center of gravity to the front and rear axle, the inter-axle distance is given by l_a . The single track model is modeled with front steering, where the front wheel can be turned by δ_r . Furthermore, tire slip angles α_v and α_h , side slip angle β and yaw angle ψ are depicted.

Note that beside the translational part the tire and vehicle coordinate system primarily differ at the front wheel, since this is rotated by the steering angle δ_r . For the following considerations the latter permits to omit the notation of coordinate systems for the sake of readability. In the same line, where appropriate, time arguments are omitted in this chapter as well. The equations of motion for the single track model read:

$$m \frac{v^2}{\rho_{cc}} \sin(\beta) - m\dot{v} \cos(\beta) + F_{x,h} - F_{a,x} + F_{x,v} \cos(\delta_r) - F_{y,v} \sin(\delta_r) = 0, \quad (5.1.3)$$

$$m \frac{v^2}{\rho_{cc}} \cos(\beta) + m\dot{v} \sin(\beta) - F_{y,h} + F_{a,y} - F_{x,v} \sin(\delta_r) - F_{y,v} \cos(\delta_r) = 0, \quad (5.1.4)$$

$$J_z \ddot{\psi} - (F_{y,v} \cos(\delta_r) + F_{x,v} \sin(\delta_r)) l_v + F_{y,h} l_h + F_{a,y} l_c = 0 \quad (5.1.5)$$

with the radius of curvature ρ_{cc} , aerodynamic drag forces $F_{a,x}$ and $F_{a,y}$ and the distance l_c between the center of gravity and the center of pressure.

Relation of Angles

It is obvious that the velocity components along the longitudinal vehicle axis have to be equal, as the vehicle does not elongate:

$$v \cos(\beta) = v_h \cos(\alpha_h), \quad (5.1.6)$$

$$v \cos(\beta) = v_v \cos(\delta_r - \alpha_v), \quad (5.1.7)$$

whereas the lateral velocity components deviate by the yaw rate:

$$v_h \sin(\alpha_h) = l_h \dot{\psi} - v \sin(\beta), \quad (5.1.8)$$

$$v_v \sin(\delta_r - \alpha_v) = l_v \dot{\psi} + v \sin(\beta). \quad (5.1.9)$$

This leads to the following equations:

$$\tan(\alpha_h) = \frac{l_h \dot{\psi} - v \sin(\beta)}{v \cos(\beta)}, \quad (5.1.10)$$

$$\tan(\delta_r - \alpha_v) = \frac{l_v \dot{\psi} + v \sin(\beta)}{v \cos(\beta)}, \quad (5.1.11)$$

which can further be simplified and rearranged to get the tire slip angles α at the front and rear tire respectively:

$$\alpha_v \approx \delta_r - \beta - l_v \frac{\dot{\psi}}{v}, \quad (5.1.12)$$

$$\alpha_h \approx -\beta + l_h \frac{\dot{\psi}}{v}. \quad (5.1.13)$$

Circle of Curvature

The circle of curvature locally approximates the vehicle trajectory at one point at which the velocity v is a tangent to the vehicle trajectory. The direction of velocity v changes with the course rate $\dot{\lambda} = \dot{\beta} + \dot{\psi}$. Hence it yields:

$$v = \rho_{cc}(\dot{\beta} + \dot{\psi}). \quad (5.1.14)$$

The lateral (i.e. centripetal) acceleration for a vehicle that moves with speed v along a circular path is with the assumption of a small side slip angle β given by:

$$a_y = v^2 / \rho_{cc}. \quad (5.1.15)$$

It should be noted that in general the vehicle rotates around the instantaneous center of rotation \mathbf{p}_{cr} . The circle of curvature locally approximates the motion of the center of gravity. For now this description is sufficient as in the relevant case of circular driving at a constant speed the instantaneous center of rotation \mathbf{p}_{cr} and the center of curvature \mathbf{p}_{cc} coincide (Meywerk 2015; Mitschke and Wallentowitz 2014).

Circular Driving at a Constant Speed

Because of the steady state indicating constant yaw rate and side slip it is:

$$\dot{\beta} = 0, \quad (5.1.16)$$

$$\ddot{\psi} = 0. \quad (5.1.17)$$

The tire slip angle difference $\alpha_v - \alpha_h$ can be calculated via equations 5.1.1, 5.1.2 and

$$F_{y,v} = \frac{l_h}{l_a} m \frac{v^2}{\rho_{cc}}, \quad (5.1.18)$$

$$F_{y,h} = \frac{l_v}{l_a} m \frac{v^2}{\rho_{cc}}, \quad (5.1.19)$$

which finally yields:

$$\alpha_v - \alpha_h = m \frac{c_h l_h - c_v l_v}{c_v c_h l_a} \frac{v^2}{\rho_{cc}}. \quad (5.1.20)$$

In the stationary case ($\dot{\beta} = 0$ in equation 5.1.14) and with equations 5.1.12, 5.1.13 and equation 5.1.20 the steering angle δ_r can be expressed by:

$$\delta_r = \frac{l_a}{\rho_{cc}} + m \frac{c_h l_h - c_v l_v}{c_v c_h l_a} \frac{v^2}{\rho_{cc}}. \quad (5.1.21)$$

At low velocities the so called Ackermann steer angle δ_A is:

$$\delta_A = \lim_{v \rightarrow 0} \delta_r = \frac{l_a}{\rho_{cc}}. \quad (5.1.22)$$

The relation between steering angle δ_r and yaw rate $\dot{\psi}$ at steady state can be derived from equation 5.1.14 and equation 5.1.21:

$$\dot{\psi} = \frac{v}{l_a(1 + (v/v_{ch})^2)} \delta_r \quad (5.1.23)$$

in which v_{ch} is the characteristic velocity

$$v_{ch} = \sqrt{\frac{l_a^2 c_v c_h}{m(c_h l_h - c_h l_v)}}. \quad (5.1.24)$$

For further reading see Mitschke and Wallentowitz (2014) and Schramm et al. (2018).

Vehicle Model with Steady State Yaw Dynamics of a Linear Single Track Model

With regard to the application of trajectory planning for normal driving situations a vehicle model with steady state yaw dynamics of a linear single track model is chosen as the best compromise between computational effort and accuracy.

The applied vehicle model reads:

$$\dot{p}_x = v \cos(\psi), \quad (5.1.25a)$$

$$\dot{p}_y = v \sin(\psi), \quad (5.1.25b)$$

$$\dot{\psi} = \frac{v}{l_a(1 + (v/v_{ch})^2)} \delta_r, \quad (5.1.25c)$$

$$\dot{v} = a \quad (5.1.25d)$$

with states $\mathbf{x} = [p_x \ p_y \ v \ \psi]^T$ and control inputs $\mathbf{u} = [a \ \delta_r]^T$.

Note that because of the steady state assumption the course rate $\dot{\lambda}$ equals the yaw rate $\dot{\psi}$ and hence can be used interchangeably. Furthermore, as will be shown in the next subsection, the trajectory planning approach benefits from the chosen vehicle model as it is differentially flat, facilitating the integration of system dynamics into the trajectory planning approach.

Limitations Arising from the Choice of Vehicle Model

From the derivation it is obvious that the validity of the chosen vehicle model is restricted to a certain field of application. Consequently, neither the behavior for larger steering angles nor for higher lateral accelerations is covered. The scope of application can be inferred from the way the lateral tire forces are handled, as the tire force characteristics can in general be linearized up to one third of the maximum lateral tire force $F_{y,max}$. This yields a lateral acceleration of

$$\frac{v^2}{\rho_{cc} a_g} \leq \frac{1}{3} \mu_t, \quad (5.1.26)$$

which is equivalent to the definition of a normal driving situation in chapter A.1. On a dry road surface a vehicle lateral acceleration of $0.4a_g$ marks the range of validity for the consideration of the linearized single track model.

From the construction design of German roads stated in RAS-L (1995) the minimum curve radius ρ_{\min} is given in dependence on the velocity v , the exploitation of the maximum radial adhesion coefficient ζ and the cross slope ξ (see Table 5.1).

Table 5.1.: Rounded minimum curve radius as defined in RAS-L (1995) for road construction. The calculations to derive the values can be found in chapter A.6.

v [km/h]		50	60	70	80	90	100	120
ρ_{\min} [m]	for $\xi = 7\%$, $\zeta = 50\%$	80	120	180	250	340	450	720
	for $\xi = 2.5\%$, $\zeta = 10\%$	320	490	700	980	1400	1700	2700

Via equation 5.1.15, it can hence be inferred that from the constructional point of view the lateral acceleration is below the admissible limit for the validity of the model.

Although it should be noted that the lateral acceleration caused by a lane change has to be taken into account on top of this considerations, it can be concluded that the derived model is valid for normal driving situations and thus fulfills the requirement.

5.1.2. Control Input Calculation

To calculate the control input values from the trajectory (i.e. solve the inverse dynamics problem) the property of flatness can be exploited. Flatness was first defined by Fliess et al. (1992) and Fliess et al. (1995) and generalizes the basic idea of controllability from linear systems to nonlinear system dynamics¹. Hence, systems of that kind represent an important subclass of nonlinear control systems denoted as differentially flat systems or just flat systems, providing a unique relation between trajectories in the flat output space \mathcal{Z} and the state space \mathcal{X} and control input space \mathcal{U} .

A nonlinear system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ is flat if the states $\mathbf{x} \in \mathcal{X}$ and control inputs $\mathbf{u} \in \mathcal{U}$ can be expressed by a new variable $\mathbf{z} \in \mathcal{Z}$ and a finite number of its derivatives:

$$\mathbf{z} = \Phi(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}, \dots, \mathbf{u}^{(r_u)}), \quad r_u \geq 1, \quad (5.1.27)$$

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \Psi(\mathbf{z}, \dot{\mathbf{z}}, \dots, \mathbf{z}^{(r_z)}), \quad r_z \geq 1, \quad (5.1.28)$$

where variable \mathbf{z} is denoted as flat output. The property of differential flatness enables efficient analytic calculation of the control inputs of a trajectory represented in the flat output space. The advantages of differential flatness can beneficially be combined with a representation of the flat output variables expressed as a sequence of differentiable trajectories. This is because states and inputs of a flat system can be mapped algebraically to the flat output variables and their derivatives and vice versa. To obtain the control inputs this means that the derivatives of the flat output have to be calculated, which promotes the choice of a differentiable flat output representation. On that

¹E.g. all linear, controllable systems are flat. In fact it can be shown that any system that can be transformed into a linear system by changes of coordinates or static or dynamic feedback transformations is also flat (Martin et al. 2003).

account chapter 6 is dedicated to the choice of a spline-based trajectory representation, giving details with respect to the general concept and the interpolation strategy.

The following calculations to determine the necessary algebraic functions are similar to Schlechtriemen et al. (2016), but differ in the important aspect of lateral vehicle dynamics modeling. In particular, the difference lies in the description of the yaw dynamics as Schlechtriemen et al. (2016) assume Ackermann steering, which is only valid at very low speeds. For more details in that respect it is referred to chapter A.5 in which the corresponding equations for the kinematic vehicle model with Ackermann steering are given.

In this work a vehicle model with steady state yaw dynamics of a linear single track model as given in equation 5.1.25 is used. This vehicle model is differentially flat with respect to the outputs p_x and p_y . The flat transformation which maps the states \mathbf{x} and control input \mathbf{u} to the flat output variables \mathbf{z}_x and \mathbf{z}_y is then given by:

$$\Phi : \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} \mapsto \begin{bmatrix} \mathbf{z}_x \\ \mathbf{z}_y \end{bmatrix},$$

$$\begin{bmatrix} z_{x,1} \\ z_{x,2} \\ z_{x,3} \\ z_{y,1} \\ z_{y,2} \\ z_{y,3} \end{bmatrix} = \begin{bmatrix} p_x \\ \dot{p}_x \\ \ddot{p}_x \\ p_y \\ \dot{p}_y \\ \ddot{p}_y \end{bmatrix} = \begin{bmatrix} p_x \\ v_x \\ a_x \\ p_y \\ v_y \\ a_y \end{bmatrix} = \begin{bmatrix} p_x \\ \cos(\psi)v \\ \cos(\psi)a - \sin(\psi)\dot{\psi}v \\ p_y \\ \sin(\psi)v \\ \sin(\psi)a + \cos(\psi)\dot{\psi}v \end{bmatrix}. \quad (5.1.29)$$

In accordance to the applied vehicle model as given in equation 5.1.25 the inverse transform to calculate the control inputs yields:

$$\Psi : \begin{bmatrix} \mathbf{z}_x \\ \mathbf{z}_y \end{bmatrix} \mapsto \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix},$$

$$\begin{bmatrix} p_x \\ p_y \\ v \\ \psi \\ a \\ \delta_r \end{bmatrix} = \begin{bmatrix} z_{x,1} \\ z_{y,1} \\ \frac{z_{x,2}}{\cos(\arctan(z_{y,2}/z_{x,2}))} \\ \arctan(z_{y,2}/z_{x,2}) \\ \frac{z_{x,3}}{\cos(\psi)} + \frac{\sin(\psi)v}{\cos(\psi)} \frac{z_{y,3} - \tan(\psi)z_{x,3}}{\tan(\psi)\sin(\psi)v + \cos(\psi)v} \\ \frac{z_{y,3} - \tan(\psi)z_{x,3}}{\tan(\psi)\sin(\psi)v + \cos(\psi)v} \frac{l_a(1+(v/v_{ch})^2)}{v} \end{bmatrix}. \quad (5.1.30)$$

Finally, this means that the control inputs can be derived analytically from the output trajectory. For further details with respect to the derivation of the equations for the inverse flat transform it is referred to chapter A.7.

5.2. Modeling of the Environment

For comfortable and safe driving knowledge about the surrounding environment is essential. The transition from perceived sensor data to trajectory planning is performed

by generating an environment model. The modeling of the environment is hence crucial for the performance of the trajectory planning approach. The chosen environment model and chosen solution approach to trajectory planning go hand in hand, tuned to match the requirements from numerical perspective as well as with respect to the overall performance. In the following an overview on environment modeling for collision avoidance in trajectory planning is given. The choice and performance of the environment model is dependent on the provided data from perception, as well as the situation analysis module. In most cases dynamic obstacles like other vehicles, or moving pedestrians are provided as an dynamic object list. The current as well as the predicted states of each detected traffic participant are given, mainly focusing on the relevant quantities like position and velocity. Lane boundaries on the other hand can be regarded as a kind of static obstacles, which among other static objects contribute to the derivation of a description of a drivable free space. For trajectory planning in particular it is important to take a rather accurate environment model into account that simultaneously does not complicate the overall process of trajectory generation. In dependence on the respective category of the trajectory planning method, some approaches even limit the number of usable environment models, since they require the presence of certain features.

One possibility to represent the environment is the use of line models. Such kind of description is especially valid for static environments, still there are some approaches that show some kind of consolidated models. Ziegler et al. (2014) for example choose to model the environment as polygons. Therefore, the static environment is imposed by the driving corridor, which is derived as a sequence of lane segments. Static objects are incorporated by building constraint polygons from sensor data. Likewise the motion of moving objects is predicted into the future to form a polygon for every discrete time interval. To incorporate obstacle vehicles into the description of lane boundaries Nilsson et al. (2015) adjust the forward collision avoidance constraint and the rear collision avoidance constraint in dependence on the driving lane of the obstacle. A prevalent holistic representation of the environment is the use of occupancy grids, which are introduced by Elfes (1989). In general 2D occupancy grids are designed to model the static environment. Nevertheless, several subsequent ideas deal with the development of an extension for dynamic objects. Tanzmeister and Wollherr (2017) for example, propose a grid-based tracking and mapping algorithm that simultaneously estimates the static and the dynamic features of the environment. With regard to trajectory planning it should be mentioned that occupancy grids rely on their representation as discrete cells with fixed sizes, which leads to a discontinuous description that is e.g. not appropriate for continuous optimization approaches. For this reason they can only be used for sampling based trajectory planning approaches like presented by Mouhagir et al. (2016). In the same line as occupancy grids, a potential field is another common way to represent the environment. The basic idea can be summarized to be similar to a hazard map in which a high potential denotes a high risk and vice versa. Hence, high and low potentials represent non negotiable and negotiable areas, respectively. Bauer et al. (2012) introduce a potential field within an approach to collision avoidance, which is composed of a static and a dynamic potential part. Wang et al. (2015) present a comprehensive version of a potential field, denoted as driving safety field.

It is divided in a potential field, a kinetic field and a behavior field, intended to take the static and dynamic environment as well as the (estimated) driver behavior into account. Brandt et al. (2007) introduce a potential field hazard map for lane keeping and collision avoidance.

The two essential, if not even the most important parts in the context of environment modeling for trajectory planning are the aspects of collision checking and distance calculation. It is obvious that the choice of a representative model has a direct impact on the choice of a suitable collision detection and distance calculation approach and vice versa. In particular for the modeling of lane boundaries the shortest distance is of high interest. Dynamic obstacle modeling on the other hand is mainly concerned with finding an accurate geometric shape that is moreover suitable for applying fast collision checks. For some general remarks with respect to the topic of collision detection it is referred to Jimenez et al. (2006). In the following modeling and collision checking strategies for the relevant environment components will be discussed in more detail.

Modeling of Lane Boundaries and Static Collision Checking

A very important aspect with respect to automated driving in structured environments is the choice of a suitable lane boundary representation, which will affect the accuracy and the computational burden (which explicitly includes memory consumption) with respect to vital functionality. One major aspect in that regard is the determination of the shortest distance from the ego vehicle to the lane boundaries, which is e.g. crucial for lane boundary collision checking as well as for curvilinear transformations.

A polynomial description of lane markers is beneficial in terms of memory consumption and also allows for simple evaluation to derive essential position bounds. On the other hand the accuracy can suffer from this kind of representation, since the approximation of the real lane boundaries is committed to the polynomial shape. Obviously, to attain a certain quality the polynomial order is of importance. An analytic solution to the shortest distance calculation problem that calculates the shortest distance from a point to the lane boundary polynomial is non-existent for a polynomial order higher than two. Nevertheless, numerical calculations can be done to get approximate results. For further reading see e.g. Willemsen et al. (2003).

Polygonal chains on the other hand are disadvantageously in the point of memory consumption, but are beneficial in accurately fitting the lane marker and fast shortest distance calculation. For these reasons the polyline description is chosen in this work. To account for discontinuities at the intersection of consecutive polyline segments a pseudo distance transform can be applied. Finally, this allows to derive a fast solution to the problem of shortest distance calculation. For details with respect to the calculations see Bender et al. (2014). In this work static collision checking to detect a collision with road boundaries, is performed by calculating the closest distance between the ego vehicle's bounding box and the respective lane boundary. The road model containing the respective lane markers considers at most three lanes: the ego lane and if present the left and/or right neighboring lane. Figure 5.2 shows the utilized road model with lane markers represented by polygonal chains. Lane markers are labeled in vehicle coordinate frame from the left to the right, starting with lane marker index $\ell = 0$ for

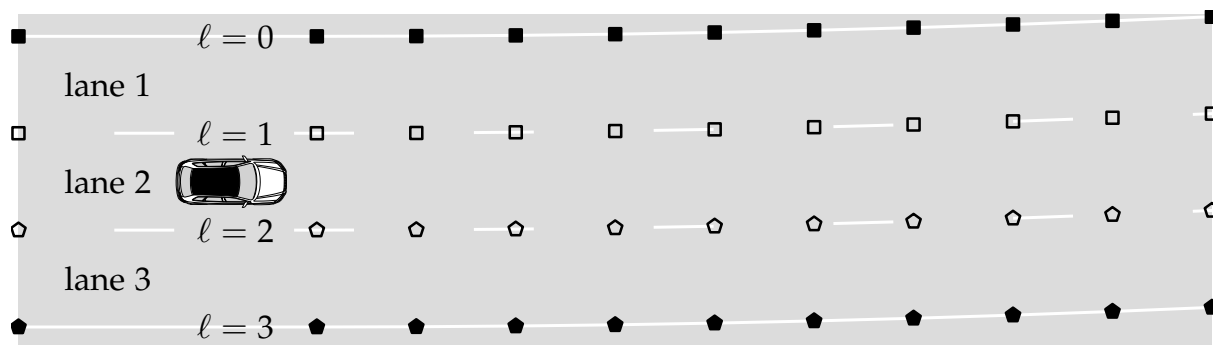


Figure 5.2.: Applied road model for static collision checking.

the left lane marker of the left lane 1, index $\ell = 1$ for the left and index $\ell = 2$ for the right lane marker of center lane 2, as well as index $\ell = 3$ for the right lane marker of the right neighboring lane 3.

Modeling of Dynamic Obstacles and Dynamic Collision Checking

In order to check for collision avoidance, an accurate geometric modeling of the vehicle shape is essential. Therefore two main representations can be identified that build up on basic geometric shapes. Namely the vehicle shape is modeled with either rectangles or by circles. The aspect of accuracy with respect to both kinds of vehicle shape modeling is discussed controversially, since a suitable arrangement with multiple circles will lead to a good coverage of the vehicle shape. However, it seems more intuitive to model the spatial dimensions of vehicle by means of a rectangular bounding box.

In general, to account for the dynamic character of the environment collision checking has to be performed in space-time domain. One approach in that regard is to check each ego candidate trajectory point-wise against all obstacle trajectories and repeatedly apply a static interference test. This is hence denoted as multiple interference test. It should be noted that the reliability of the performed collision checks is hence subject to the resolution of the trajectory, since with a too coarse sampling occurring collisions might be missed. In general to reduce run time with respect to the elaborate task of collision checking, strategies like *Hierarchical Pruning* (Ferguson et al. 2008a) or the *Bounding Volume Hierarchy* (Schwesinger et al. 2015) have been applied to improve upon the efficiency.

Typically, in dependence on the choice of geometric modeling, several characteristics are exploited to perform collision detection. The decomposition of the vehicle shape into rotation invariant primitives using multiple circles is for example shown by Ziegler and Stiller (2010). With this kind of approximation collision detection can be performed via distance calculations and comparatively simple threshold evaluations. Therefore, at each time instance the distances between the ego vehicle position and each center of the approximating circles is checked whether it is below the radius of the respective circle. The *Separating Axis Theorem*, which is accurate for convex shapes can be utilized to check for intersections between two rectangles in time. To detect a collision a projection onto the aligning axis of the ego vehicle and the respective object in consideration is performed. The two objects are not colliding, if there exists a line onto which

the projections of the ego vehicle and the object do not overlap. The *Separating Axis Theorem* is a pure collision detection approach and does not provide any information with respect to the actual distance between two objects. The *Gilbert-Johnson-Keerthi Algorithm* (Gilbert et al. 1988) for collision detection on the other hand computes the distance between two convex shapes. It works with an implicit representation of convex shapes through support functions, Minkowski sums and simplexes. If the Minkowski difference contains the origin of the space, the shapes intersect. Otherwise the minimum distance between the origin and the Minkowski difference represents the distance between the two objects.

In this work a circular approximation is chosen likewise the one presented by Ziegler and Stiller (2010). Then the radius $\mathring{\rho}$ of each approximating circle and the distance \mathring{b} between two consecutive circle centers is given by:

$$\mathring{\rho} = 0.5 \sqrt{\frac{l^2}{\mathring{\eta}^2} + w^2}, \quad \mathring{b} = 2 \sqrt{\mathring{\rho}^2 - \frac{w^2}{4}}. \quad (5.2.1)$$

Each vehicle (i.e. the ego and obstacle vehicles) is approximated by $\mathring{\eta}$ circles, as illustrated in Figure 5.3. A vehicle can hence be approximated by for example $\mathring{\eta} = 3$ circles

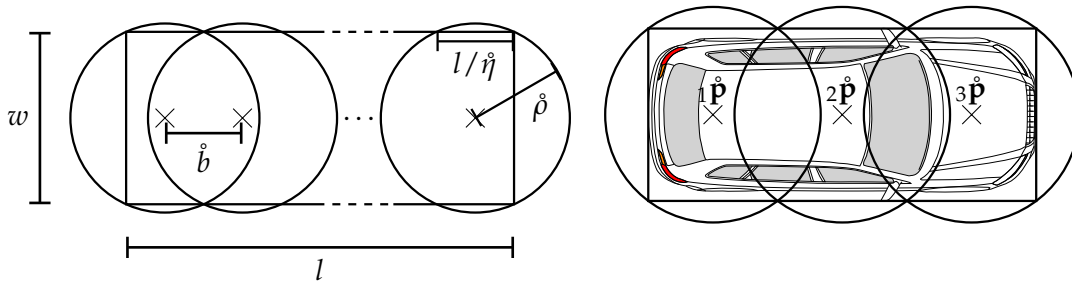


Figure 5.3.: Circle approximation of the vehicle shape.

with circle center positions $1\mathring{p}$, $2\mathring{p}$ and $3\mathring{p}$, placed in relation to the vehicle's center. As can be seen this setup adds a certain margin to the vehicle shape. Obviously, a higher number of approximating circles leads to a more accurate approximation of the vehicle shape, but increases the calculation effort for collision checking at the same time.

5.3. Trajectory Planning Objectives

When designing the overall cost function each objective has to be chosen thoughtfully. One reason for this is that the cost function must consider both safety and comfort at the same time, while the aggregated objectives should comply to the idea of mitigating the effect of mutual interference. The main challenge is to find a set of objectives that describe the desired behavior of the ego vehicle in a general way, but simultaneously promote the optimal behavior for each specific traffic situation as well. It is hence necessary to carefully balance each objective against each other, since it might sometimes be better to increase or decrease some quantity of interest in dependence on the current situation. This issue is particularly addressed in the overall trajectory planning

concept by proposing the two-level hierarchical trajectory optimization framework as presented in section 3.3. Still, to derive an explicit solution to the trajectory planning problem distinctive features have to be identified. Unfortunately, the definition of optimality in the context of automated driving is not unique and can be expressed in a range of various instances of a maneuver that would lead to the desired outcome. However, some general purpose objectives can be defined.

Distance Keeping

A very basic feature that should be captured by the trajectory planning approach is the objective of distance keeping to other obstacle vehicles. In contrast to collision avoidance the safety distance that supports the overall sense of security cannot be defined straightforward. First of all, safety distance is a subjective quantity that varies with respect to the individual passenger as well as in regard to the current situation. For a consistent driving style it is moreover essential that the case of driving below a certain safety distance is permitted for at least some time. The distance keeping objective is based on the constant time gap assumption similar to adaptive cruise control functionality. The idea is generalized in a way that lead and tail vehicle are identified for each trajectory point and the costs are then given with respect to the calculated distance in relation to the required time gap. Note that on the current ego lane only the lead vehicle is considered in order to prevent from being inordinately impacted by the tail vehicle driving too close to the ego vehicle.

Reference Speed Control

The ego vehicle should be able to travel at a desired set speed. This objective is on the one hand intended to promote progress along the route, but also covers the aspect of adherence to speed limits.

Reference Path Following

This objective is dedicated to the lateral guidance of the ego vehicle and intended to provide information about the preferred driving lane. It hence incorporates knowledge from the high level maneuver decision and is formulated as a penalty on the deviation to a desired lateral reference path.

Comfort Driving

This objective originates from a soft-constraint formulation for the comfort lateral and longitudinal acceleration, acting as a penalty in case of exceeding a certain comfort acceleration threshold. This means that the ego vehicle should obey the desired comfortable acceleration as long as it is appropriate with respect to the current situation. The integration into the objective function accounts for the fact that it is desired, but not mandatory to perform motions below the comfort acceleration threshold.

6

Spline-based Trajectory Representation

The spline-based trajectory representation constitutes the base frame of this work. It rigorously supports the idea of the two level hierarchical frame work as described in section 3.3. To catch up the considerations made with respect to the trajectory planning concept the assumption is made that a spline is capable of matching the optimal function class. This is not a daring hypothesis, since the flexibility and the adaptiveness of a spline are evident. Moreover, as shown in this chapter, a polynomial representation yields the optimal solution to a minimization of kinematics without constraints.

With regard to the idea of exploiting the property of flatness to incorporate system dynamics into the trajectory planning approach, it is beneficial to express the flat output variables as a sequence of differentiable trajectories such as polynomials. This is perfectly matching to the considerations made in chapter 5.1.2, since derivatives are needed for the respective calculations. Finally, trajectory optimization over the spline coefficients results in a lower-dimensional and hence more efficient solution. Especially for flat systems it is common to specify a separate spline for each of the flat output variables of the system. Hence, in this work one time parameterized trajectory for each flat output variable is planned, covering each dimension of the flat output space \mathcal{Z} . In a first step the lateral and longitudinal motion are considered independently and finally connected via discrete sampling in each direction with fixed sampling time T_s . For spline-based trajectory planning the trajectory is composed of several segments:

$$S(t) = \begin{cases} P_0(t) & t_0 \leq t \leq t_1 \\ P_1(t - t_1) & t_1 \leq t \leq t_2 \\ P_2(t - t_2) & t_2 \leq t \leq t_3 \\ \vdots \\ P_{\mathcal{K}}(t - t_{\mathcal{K}}) & t_{\mathcal{K}} \leq t \leq t_{\mathcal{K}+1} \end{cases} \quad (6.0.1)$$

with a polynomial P_{κ} of degree \mathcal{V} given as:

$$P_{\kappa}(t) = c_{0,\kappa} + c_{1,\kappa}t + c_{2,\kappa}t^2 + \dots + c_{\mathcal{V}-1,\kappa}t^{\mathcal{V}-1} + c_{\mathcal{V},\kappa}t^{\mathcal{V}}. \quad (6.0.2)$$

The coefficients $c_{\iota,\kappa}$ with $\iota = 0, 1, \dots, \mathcal{V}$ for each spline segment $\kappa = 0, 1, \dots, \mathcal{K}$ are written in a vector \mathbf{c}_{κ} , which yields $\mathbf{c} = [\mathbf{c}_0^T, \mathbf{c}_1^T, \dots, \mathbf{c}_{\mathcal{K}}^T]^T$ for the coefficients of the

entire spline. The link between two segments is denoted as spline breakpoint \mathbf{z} , also referred to as spline knot. Figure 6.1 illustrates an example in which three breakpoints for longitudinal and four breakpoints for lateral direction have been configured.

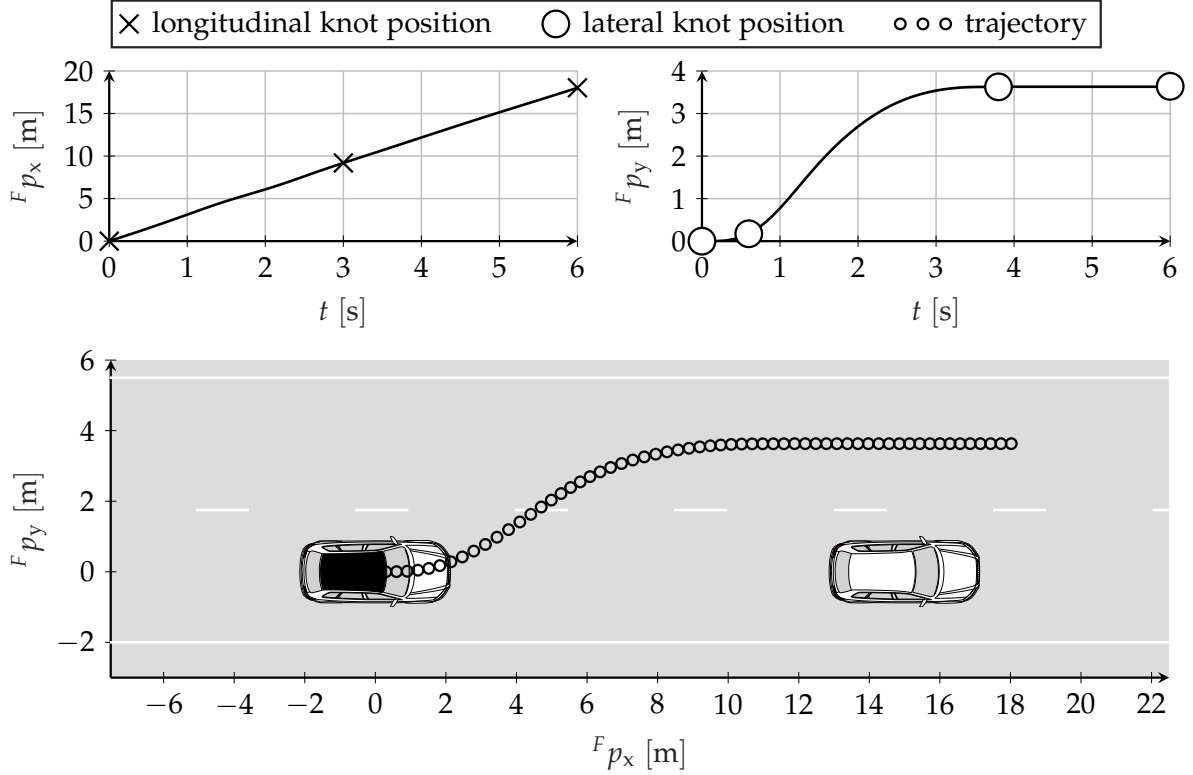


Figure 6.1.: Illustration of the spline-based trajectory representation. The trajectory (bottom) is composed of one spline for longitudinal (top left) and one for lateral direction (top right).

The continuous spline-based formulation allows the choice of resolution in time T_s independently of the number of segments \mathcal{K} and the fixed planning horizon T_p . Thus, by interpolation and evaluation of the splines for longitudinal and lateral direction $K = T_p/T_s + 1$ trajectory points $\mathbf{p}_k = [^F p_{x,k}, ^F p_{y,k}]^T$ are gained yielding the ego vehicle trajectory:

$$\mathbf{P}_{\text{ego}} = [\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_k, \dots, \mathbf{p}_K]^T \quad (6.0.3)$$

with constant time intervals $t \in [t_k, t_{k+1}]$ for $k = 0, 1, \dots, K$.

6.1. Variational Formulation

In cases without any constraints arising i.a. from obstacles or the vehicle dynamics, the variational problem of trajectory planning can be solved analytically. The goal is to find the optimal solution for position $p(t)$ by minimizing a function \mathcal{L} that interrelates the unknown function $p(t)$ to its derivatives:

$$p^*(t) = \arg \min_{p(t)} \int_0^{T_p} \mathcal{L}(p(t)^{(r)}, p(t)^{(r-1)}, \dots, \dot{p}(t), p(t), t) dt . \quad (6.1.1)$$

The calculus of variation aims at finding the extrema e.g. the stationary functions of a functional. The functional \mathcal{F} is given as the integral of function \mathcal{L} :

$$\mathcal{F}[p(t)] = \int_0^{T_p} \mathcal{L}(p(t)^{(r)}, p(t)^{(r-1)}, \dots, \dot{p}(t), p(t), t) dt . \quad (6.1.2)$$

The optimal solution can now be obtained via the Euler-Lagrange equation¹. For an integrand \mathcal{L} with derivatives up to order r the Euler-Lagrange equation reads:

$$\underbrace{\frac{\partial \mathcal{L}}{\partial p}}_{=0} - \frac{d}{dt} \underbrace{\left(\frac{\partial \mathcal{L}}{\partial \dot{p}} \right)}_{=0} + \frac{d^2}{dt^2} \underbrace{\left(\frac{\partial \mathcal{L}}{\partial \ddot{p}} \right)}_{=0} - \dots + (-1)^r \frac{d^r}{dt^r} \left(\frac{\partial \mathcal{L}}{\partial p^{(r)}} \right) = 0. \quad (6.1.3)$$

The solutions of equation 6.1.3 are the functions for which the functional \mathcal{F} is stationary and hence indicates optimality. In case of an integrand $\mathcal{L} = \left(p^{(r)} \right)^2$ the first r terms vanish to zero and it yields:

$$(-1)^r \frac{d^r}{dt^r} \left(\frac{\partial \mathcal{L}}{\partial p^{(r)}} \right) = (-1)^r \frac{d^r}{dt^r} \left(2p^{(r)} \right) = (-1)^r \left(2p^{(2r)} \right) = 0, \quad (6.1.4)$$

which finally results in the following differential equation that has to be solved:

$$p^{(2r)} = 0. \quad (6.1.5)$$

The solution of this differential equation, with coefficients c_i is given as:

$$p(t) = c_0 + c_1 t + c_2 t^2 + \dots + c_{2r-1} t^{(2r-1)}. \quad (6.1.6)$$

This means that for an unconstrained problem, the optimal solution with respect to the r -th derivative of the vehicle position $p(t)$ is given as a polynomial of $(2r - 1)$ -th order. Table 6.1 summarizes the relationship between trajectory optimality and the polynomial order of the corresponding solution.

Table 6.1.: Overview of the relationship between trajectory optimality and polynomial order of the corresponding solution.

cost functional derivative	trajectory optimality	polynomial order
$r = 1$	minimum velocity	line with $\mathcal{V} = 1$
$r = 2$	minimum acceleration	cubic polynomial with $\mathcal{V} = 3$
$r = 3$	minimum jerk	quintic polynomial with $\mathcal{V} = 5$
$r = 4$	minimum snap	septic polynomial with $\mathcal{V} = 7$

Note that the derivation has been shown for a one-dimensional case, but can easily be extended to more dimensions. For example in a two-dimensional case the solution results in a system of two independent differential equations that have to be solved.

¹For the sake of readability arguments indicating time dependence are omitted in the following derivation.

The remaining problem consists of finding the coefficients for the polynomial with respect to given constraints on the trajectory. To increase the scope of action for the trajectory planning algorithm in terms of flexibility, instead of finding the optimal coefficients for one single polynomial a sequence of polynomials is optimized, which yields the spline-based representation for the trajectory.

6.2. Distinct Analytic Spline Interpolation

For a distinct analytic spline interpolation the coefficients are calculated analytically. As can be inferred from Table 6.1 the polynomial order to minimize the respective kinematic quantity is an odd number, which hence means that an even number of coefficients needs to be determined. It can be found that considering the correlating derivative start and end conditions for each respective segment can be derived yielding the required number of equations to set up a square system of linear equations². Since this is done for each single spline segment this allows for a formulation in terms of spline breakpoints.

For segments $\kappa = 0, 1, \dots, \mathcal{K}$ and the required derivative $r_\gamma = (\mathcal{V} - 1)/2$ to allow for analytic determination of the coefficients $c_{l,\kappa}$ it is:

$$\begin{aligned}
 c_{0,\kappa} + c_{1,\kappa} t_{\vdash,\kappa} + \dots + c_{\mathcal{V}-1,\kappa} t_{\vdash,\kappa}^{\mathcal{V}-1} + c_{\mathcal{V},\kappa} t_{\vdash,\kappa}^{\mathcal{V}} &= p_{\vdash,\kappa}, \\
 c_{0,\kappa} + c_{1,\kappa} t_{\dashv,\kappa} + \dots + c_{\mathcal{V}-1,\kappa} t_{\dashv,\kappa}^{\mathcal{V}-1} + c_{\mathcal{V},\kappa} t_{\dashv,\kappa}^{\mathcal{V}} &= p_{\dashv,\kappa}, \\
 c_{1,\kappa} + \dots + (\mathcal{V} - 1) c_{\mathcal{V}-1,\kappa} t_{\vdash,\kappa}^{\mathcal{V}-2} + \mathcal{V} c_{\mathcal{V},\kappa} t_{\vdash,\kappa}^{\mathcal{V}-1} &= \dot{p}_{\vdash,\kappa}, \\
 c_{1,\kappa} + \dots + (\mathcal{V} - 1) c_{\mathcal{V}-1,\kappa} t_{\dashv,\kappa}^{\mathcal{V}-2} + \mathcal{V} c_{\mathcal{V},\kappa} t_{\dashv,\kappa}^{\mathcal{V}-1} &= \dot{p}_{\dashv,\kappa}, \\
 &\vdots \\
 \sum_{l=r_\gamma}^{\mathcal{V}} (\prod_{\epsilon=0}^{r_\gamma} (l - \epsilon)) c_{l,\kappa} t_{\vdash,\kappa}^{l-r_\gamma} &= p_{\vdash,\kappa}^{(r_\gamma)}, \\
 \sum_{l=r_\gamma}^{\mathcal{V}} (\prod_{\epsilon=0}^{r_\gamma} (l - \epsilon)) c_{l,\kappa} t_{\dashv,\kappa}^{l-r_\gamma} &= p_{\dashv,\kappa}^{(r_\gamma)}
 \end{aligned} \tag{6.2.1}$$

where start and end conditions are denoted by the subscript \vdash and \dashv respectively. Note that the specified start and end conditions coincide for connecting segments. Following this construction the distinct analytic spline interpolation is inherently accompanied by the following characteristics:

- The spline passes through the specified breakpoints $\underline{\mathbf{z}}$, which coincide with the start and end conditions of the respective spline segments in terms of the desired values for each derivative,
- the spline is continuous between each spline segment up to the r_γ -th derivative.

Solving the linear system of equations analytically for the coefficients is comparatively simple and fast. However, it is not easy to determine the corresponding values for the breakpoints and its derivatives (i.e. start and end conditions of each polynomial) to result in an overall optimal behavior of the concatenated spline segments.

²In that context this means a system with an equal number of equations and unknowns.

6.3. Minimum Kinematics Trajectory Generation

The term minimum kinematics accounts for the idea of calculating the coefficients with respect to a cost function with a kinematic derivative of arbitrary order. In dependence on the desired output it is possible to minimize acceleration as well as to generate a minimum jerk trajectory.

Bry et al. (2015) introduce a piecewise polynomial joint optimization with the aim of generating optimal trajectories that should pass through a sequence of given waypoints (i.e spline breakpoints). This corresponds to an optimal spline interpolation strategy that will be used as an underlying minimum kinematics trajectory scheme within the developed trajectory planning concept (see section 3.3). In contrast to the trajectory generation presented in this work Bry et al. (2015) propose an A* or RRT* algorithm for waypoint generation to enable quadrotor flights in mostly static environments. Schmidt et al. (2019b) use this idea and transfers it to the context of automated driving for the calculation of longitudinal trajectories.

For the sake of flexibility and thus coinciding with optimality of the trajectory with respect to a given traffic situation, a sequence of multiple polynomial trajectory segments is optimized. The trajectory between each pair of breakpoints is considered as an individual polynomial trajectory segment, for which a joint optimization promises lower costs for the resulting optimized trajectory.

6.3.1. Formulation of the Optimal Interpolation Problem

To find the optimal coefficients cost function \hat{f} for one spline segment is defined as:

$$\hat{f}_\kappa = \int_{t=+}^{-} \omega_0 P_\kappa(t)^2 + \omega_1 \dot{P}_\kappa(t)^2 + \omega_2 \ddot{P}_\kappa(t)^2 + \dots + \omega_\mathcal{V} P_\kappa^{(\mathcal{V})}(t)^2 dt \quad (6.3.1)$$

in which ω_r are the weights that balance the desired penalty among each derivative. In order to define the costs for the entire spline with $\kappa = 0, 1, \dots, \mathcal{K}$ spline segments this can be rewritten in terms of the spline coefficients \mathbf{c} , which yields:

$$\hat{f}(\mathbf{c}) = \mathbf{c}^T \mathbf{Q} \mathbf{c}. \quad (6.3.2)$$

In the following the construction of the cost matrix \mathbf{Q} is described. Furthermore, it is shown how to impose several constraints on the spline.

Generation of the Cost Matrix

The cost matrix \mathbf{Q} contains the desired penalty on each derivative of the spline segments. For the r -th derivative the cost matrix element of one spline segment reads:

$$r Q_\kappa^{i,j} = \begin{cases} 2(\prod_{\varepsilon=0}^{r-1} (t - \varepsilon)(j - \varepsilon) \frac{t_{-\kappa}^{i+j-2r+1}}{i+j-2r+1}) & i \geq r \wedge j \geq r \\ 0 & i < r \vee j < r \end{cases}, \quad (6.3.3)$$

where $t_{-\kappa}$ is the spline segment duration and $i = 0, 1, \dots, \mathcal{V}$ indicates the row and $j = 0, 1, \dots, \mathcal{V}$ the column of the cost matrix. The complete cost matrix for each spline

segment κ is a weighted sum of Hessian matrices for each derivative of the polynomial:

$$\mathbf{Q}_\kappa = \sum_{r=0}^{\mathcal{V}} \omega_r {}_r\mathbf{Q}_\kappa. \quad (6.3.4)$$

Finally, the cost matrix for the entire spline is composed as a block diagonal matrix of matrices \mathbf{Q}_κ for each segment:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_0 & & & \\ & \mathbf{Q}_1 & & \\ & & \ddots & \\ & & & \mathbf{Q}_\kappa \end{bmatrix}. \quad (6.3.5)$$

Generation of the Constraint Matrix

The constraint matrix \mathbf{A} is built to represent the desired behavior in terms of specified values for the position and its derivatives.

For each spline segment κ one row vector element of the constraint matrix for start and end conditions denoted by the subscript \vdash and \dashv respectively is constructed via:

$${}_r a_{\vdash,\kappa}^j = \begin{cases} \prod_{\varepsilon=0}^{r-1} (j - \varepsilon) & r = j \\ 0 & r \neq j \end{cases}, \quad {}_r a_{\dashv,\kappa}^j = \begin{cases} (\prod_{\varepsilon=0}^{r-1} (j - \varepsilon)) t_{\vdash}^{j-r} & r \leq j \\ 0 & r > j \end{cases}. \quad (6.3.6)$$

The column index $j = 0, 1, \dots, \mathcal{V}$ thereby denotes the respective coefficient of the spline segment. The complete row vectors ${}_r \mathbf{a}_{\vdash,\kappa}$, ${}_r \mathbf{a}_{\dashv,\kappa}$ for a spline segment κ correspond to a constraint on the r -th derivative and basically map the coefficients to the derivatives of the polynomial.

To build the constraint matrix two different types of constraints are considered. The first type ensures adherence to specified values for the position p and its derivatives, which is also referred to as value constraints. The second type cares about continuity between subsequent spline segments and is henceforth termed continuity constraints. To constrain the shape of the spline segment κ either a value constraint or a continuity constraint is specified. For the r -th derivative it is ${}_r b_{\vdash,\kappa}$ for a start condition, ${}_r b_{\dashv,\kappa}$ for an end condition and ${}_r b_{\vdash\vdash,\kappa}$ to enforce continuity³:

$${}_r b_{\vdash,\kappa} = P^{(r)}(0), \quad {}_r b_{\dashv,\kappa} = P^{(r)}(t_{\vdash}), \quad {}_r b_{\vdash\vdash,\kappa} = 0. \quad (6.3.7)$$

Then a value constraint on the r -th derivative at the start of the respective spline segment reads:

$${}_r \mathbf{a}_{\vdash,\kappa}^{\text{val}} \mathbf{c}_\kappa = {}_r b_{\vdash,\kappa} \quad (6.3.8)$$

and an end condition is defined by:

$${}_r \mathbf{a}_{\dashv,\kappa}^{\text{val}} \mathbf{c}_\kappa = {}_r b_{\dashv,\kappa}. \quad (6.3.9)$$

³Note that Bry et al. (2015) elaborate on the possibility to specify nonzero offsets by setting ${}_r b_{\vdash\vdash,\kappa}$ to a value other than zero. Since clear continuity should be attained this is no option in this work.

To enforce continuity between two consecutive spline segments with respect to the r -th derivative one row of the constraint matrix is given by:

$$\begin{bmatrix} r\mathbf{a}_{-,k}^{\text{con}} & -r\mathbf{a}_{+,k+1}^{\text{con}} \end{bmatrix} \begin{bmatrix} \mathbf{c}_k \\ \mathbf{c}_{k+1} \end{bmatrix} = r\mathbf{b}_{+,k}. \quad (6.3.10)$$

It should be noted that a value constraint at an intermediate segment infers a continuity constraint by setting up a value constraint for the connecting segment as well. A continuity constraint on the other hand is designated for segment transitions, which are not constrained to a certain value. Finally, the overall constraint matrix \mathbf{A} , as well as the constraint value vector \mathbf{b} are given by:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{+,0}^{\text{val}} \\ \mathbf{A}_{-,0}^{\text{val}} \\ \mathbf{A}_{-,0}^{\text{con}} & -\mathbf{A}_{+,1}^{\text{con}} \\ & \mathbf{A}_{+,1}^{\text{val}} \\ & \mathbf{A}_{-,1}^{\text{val}} \\ \mathbf{A}_{-,1}^{\text{con}} & -\mathbf{A}_{+,2}^{\text{con}} \\ & & \ddots \\ & & & \mathbf{A}_{+,K-1}^{\text{val}} \\ & & & \mathbf{A}_{-,K-1}^{\text{val}} \\ & & & \mathbf{A}_{-,K-1}^{\text{con}} & -\mathbf{A}_{+,K}^{\text{con}} \\ & & & & \mathbf{A}_{+,K}^{\text{val}} \\ & & & & \mathbf{A}_{-,K}^{\text{val}} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_{+,0} \\ \mathbf{b}_{-,0} \\ \mathbf{b}_{+,0} \\ \mathbf{b}_{+,1} \\ \mathbf{b}_{-,1} \\ \mathbf{b}_{+,1} \\ \vdots \\ \mathbf{b}_{+,K-1} \\ \mathbf{b}_{-,K-1} \\ \mathbf{b}_{+,K-1} \\ \mathbf{b}_{+,K} \\ \mathbf{b}_{-,K} \end{bmatrix} \quad (6.3.11)$$

in which matrices \mathbf{A}^{val} and \mathbf{A}^{con} are composed in accordance to the desired behavior of the spline.

6.3.2. Solution of the Optimal Interpolation Problem

The cost matrix \mathbf{Q} and constraint matrix \mathbf{A} can now be set up in accordance to the desired interpolation behavior. It is hence determined which kinematic quantities are minimized in order to perform the optimal interpolation with respect to the requested reference values.

Quadratic Programming Solution

To solve the optimal interpolation problem in terms of the cost function 6.3.2 subject to the constraints specified in equation 6.3.11 the following quadratic programming problem has to be solved:

$$\min_{\mathbf{c}} \quad \mathbf{c}^T \mathbf{Q} \mathbf{c} \quad (6.3.12a)$$

$$\text{s.t.} \quad \mathbf{A} \mathbf{c} - \mathbf{b} = \mathbf{0}. \quad (6.3.12b)$$

In this case the problem is solved directly for the optimal coefficients \mathbf{c}^* , since the decision variables are the polynomial coefficients of all spline segments.

Analysis of the Optimal Interpolation Strategy

So far all necessary ingredients for applying the minimum kinematics trajectory generation in terms of the optimal interpolation have been introduced. To better understand the process of matrix generation and the differences with respect to the chosen interpolation order an analysis is performed.

The general functioning is illustrated by an example, choosing a spline with 2 ($\mathcal{K} = 1$) segments. The constraint matrix is composed in accordance to the values specified in Table 6.2. Moreover, continuity should hold up to the second derivative $r_c = 2$ and a minimum jerk trajectory should be generated, by constructing the cost matrix for the third derivative $r_f = 3$.

Table 6.2.: Setup for constraint generation.

spline knot		\mathbf{z}_0	\mathbf{z}_1	\mathbf{z}_2
position [m]	${}^F p$	0	1	8
velocity [m/s]	${}^F v$	0	□	□
acceleration [m/s ²]	${}^F a$	0	□	□
time [s]	t	$t_0 = 0$	$t_1 = 1$	$t_2 = 3$

□ : Free value to be determined via optimal interpolation

The construction of the constraints matrix needs special attention. In this work the automated process of constraints generation is implemented in a way that at maximum $2(r_c + 1)(\mathcal{K} + 1)$ conditions are generated. From the given setup it can be concluded that six value constraints⁴ and two continuity constraints for the velocity and the acceleration at $t_1 = 1$ s can be derived.

The generation of the constraint matrix is shown in detail for interpolation order $\mathcal{V} = 4$. For the two segments this means that in total 10 coefficients need to be determined. The constraints read:

$$\begin{bmatrix}
 1 & 1t_{+,0} & 1t_{+,0}^2 & 1t_{+,0}^3 & 1t_{+,0}^4 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 2t_{+,0} & 3t_{+,0}^2 & 4t_{+,0}^3 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 2 & 6t_{+,0} & 12t_{+,0}^2 & 0 & 0 & 0 & 0 & 0 \\
 1 & 1t_{-,0} & 1t_{-,0}^2 & 1t_{-,0}^3 & 1t_{-,0}^4 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 1t_{+,1} & 1t_{+,1}^2 & 1t_{+,1}^3 & 1t_{+,1}^4 \\
 0 & 0 & 0 & 0 & 0 & 1 & 1t_{-,1} & 1t_{-,1}^2 & 1t_{-,1}^3 & 1t_{-,1}^4 \\
 0 & 1 & 2t_{-,0} & 3t_{-,0}^2 & 4t_{-,0}^3 & 0 & -1 & -2t_{+,1} & -3t_{+,1}^2 & -4t_{+,1}^3 \\
 0 & 0 & 2 & 6t_{-,0} & 12t_{-,0}^2 & 0 & 0 & -2 & -6t_{+,1} & -12t_{+,1}^2
 \end{bmatrix}
 \begin{bmatrix}
 c_{0,0} \\
 c_{0,1} \\
 c_{0,2} \\
 c_{0,3} \\
 c_{0,4} \\
 c_{1,0} \\
 c_{1,1} \\
 c_{1,2} \\
 c_{1,3} \\
 c_{1,4}
 \end{bmatrix}
 =
 \begin{bmatrix}
 0 \\
 0 \\
 0 \\
 1 \\
 1 \\
 8 \\
 0 \\
 0
 \end{bmatrix}$$

As each spline interval is defined from $t \in [0, t_{+,k}]$, for the given example it is $t_{+,0} = 0$ s, $t_{-,0} = t_1 - t_0 = 1$ s, $t_{+,1} = 0$ s and $t_{-,1} = t_2 - t_1 = 2$ s. With cost matrix \mathbf{Q} for $r_f = 3$ in

⁴The position value constraint at the intermediate point $t_1 = 1$ s infers two separate value constraints in the constraint matrix to account for the continuity between the first and second segment as well.

order to derive a minimal jerk trajectory this finally leads to:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 4 & 8 & 16 \\ 0 & 1 & 2 & 3 & 4 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 6 & 12 & 0 & 0 & -2 & 0 & 0 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 8 \\ 0 \\ 0 \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 72 & 144 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 144 & 384 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 144 & 576 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 576 & 3072 & 0 & 0 \end{bmatrix}.$$

Then the optimal coefficients can be obtained by solving the quadratic program 6.3.12. The resulting coefficients are shown in Table 6.3.

Table 6.3.: Resulting coefficients of the considered example with interpolation order $\mathcal{V} = 4$.

$c_{0,0}$	$c_{0,1}$	$c_{0,2}$	$c_{0,3}$	$c_{0,4}$	$c_{1,0}$	$c_{1,1}$	$c_{1,2}$	$c_{1,3}$	$c_{1,4}$
0	0	0	1.6429	-0.6429	1	2.3571	1.0714	-0.3571	0.0536

Obviously, the starting conditions can directly be extracted from coefficients $c_{0,0}$, $c_{0,1}$ and $2c_{0,2}$. The same holds for coefficient $c_{1,0}$ that reflects the position constraint at the intermediate breakpoint at $t_1 = 1$ s. The corresponding optimal trajectory with interpolation order $\mathcal{V} = 4$ is visualized in Figure 6.2a.

In a subsequent experiment the interpolation order is varied. Since the setup as given in

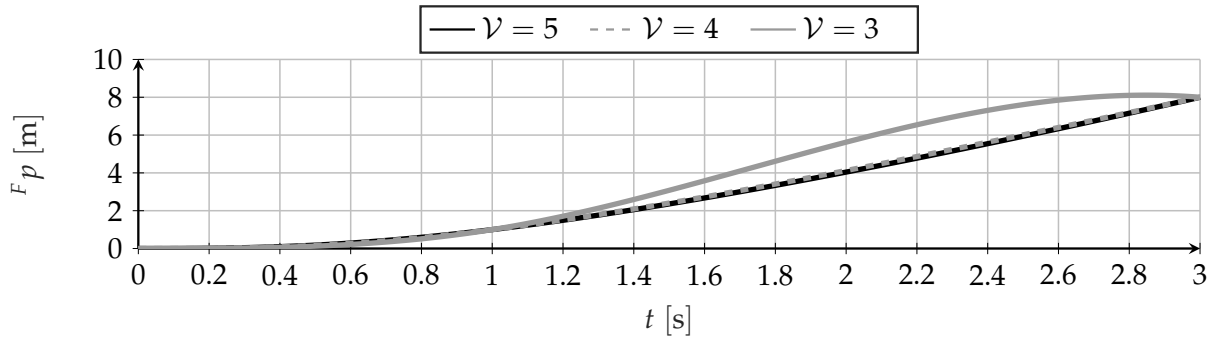
Table 6.4.: Resulting costs with respect to a different interpolation order. Since the number of segments is fix, the number of variables exclusively corresponds to the interpolation order.

interpolation order	number of variables	optimal costs []
$\mathcal{V} = 3$	8	344.25
$\mathcal{V} = 4$	10	54.00
$\mathcal{V} = 5$	12	50.55
$\mathcal{V} = 6$	14	50.55
$\mathcal{V} = 7$	16	50.55

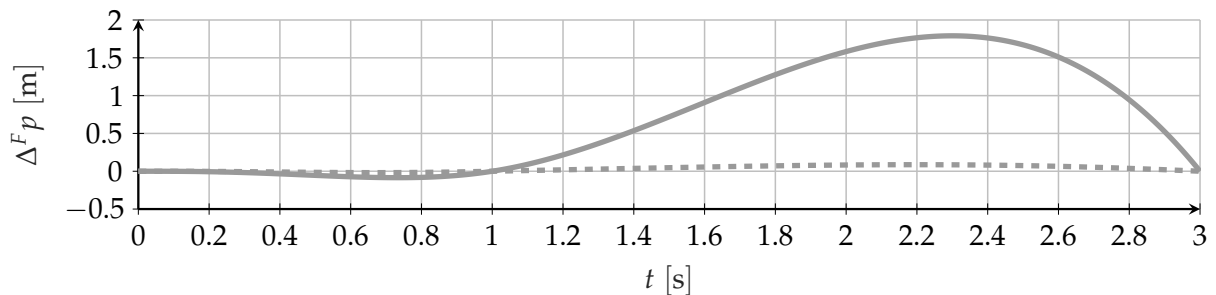
Table 6.2 is still holding, the number of constraints remains unchanged. However, with the varying interpolation order the number of variables (i.e. the number of coefficients) changes.

Moreover, the results of Table 6.4 confirm the findings of section 6.1 that a fifth order polynomial is the optimal choice with respect to generating a minimum jerk trajectory. In particular, it is shown that this aspect can be transferred to a spline that is composed of several segments as an interpolation order beyond $\mathcal{V} = 5$ does not improve up on the obtained solution. Figure 6.2 shows the suboptimal solutions for interpolation

order $\mathcal{V} = 3$ and $\mathcal{V} = 4$ in comparison to the optimal one. The deviation, as depicted in Figure 6.2b, shows that an interpolation with interpolation order $\mathcal{V} = 4$ already yields comparable accuracy in terms of the position, whereas an interpolation order of $\mathcal{V} = 3$ leads to deviations of up to 1.79 m.



(a) Resulting trajectories shown for interpolation order $\mathcal{V} = 3$, $\mathcal{V} = 4$ and $\mathcal{V} = 5$.



(b) Deviation of the trajectory with interpolation order $\mathcal{V} = 3$ and $\mathcal{V} = 4$ with respect to the optimal result generated with interpolation order $\mathcal{V} = 5$.

Figure 6.2.: Result of the optimal interpolation for the considered example.

To sum up it can be concluded that the optimal interpolation strategy supports the idea of the two level hierarchical frame work. Based on the considerations made with respect to the variational problem of trajectory planning without constraints, the theory of minimum kinematics trajectory generation expands the scope to a spline-based trajectory representation. This promises a higher flexibility and better results in terms of convenient trajectory generation. Nevertheless, the analysis shows that the setup has to be chosen carefully to obtain a meaningful result.

7

Trajectory Planning for Automated Driving

This chapter deals with the transition from the trajectory planning concept as derived in section 3.3 to a mathematical problem formulation, that allows to determine a suitable trajectory for automated driving applications. The proposed two level hierarchical trajectory optimization framework is therefore mapped to a bilevel programming problem, with the low level minimum kinematics trajectory generation. Finally, the optimal trajectory is found by applying a discrete or a continuous trajectory optimization approach to the high level optimization problem.

7.1. Bilevel Programming

Bilevel programming refers to a class of mathematical problems, in which one optimization problem is nested within another. The inner and outer optimization task are connected by the fact that the constraints of the outer optimization problem are defined in part by a nested inner parametric optimization problem. Hence, only those solutions are valid that are optimal in terms of the inner optimization problem, satisfying the constraints of the outer optimization problem at the same time (Sinha et al. 2018). In general such hierarchical optimization problems are represented by a two-level hierarchical system, in which the two interconnected entities make their decisions with respect to their own profit and constraints, but on different levels. As the two levels have their own objectives and constraints, the set of decision variables is partitioned accordingly. Although in literature related to bilevel programming the different levels are commonly referenced as upper and lower level, in compliance to the terminology used in section 3.3 they will coherently be referred to as high and low level, respectively. Alternatively, an illustrative description is given in terms of a leader-follower policy. For a high level decision vector $\hat{\mathbf{z}}_h$ chosen by the leader, the follower reacts optimally by selecting the low level decision vector $\hat{\mathbf{z}}_l$ in order to minimize the low level objective function $\hat{f}(\hat{\mathbf{z}}_h, \hat{\mathbf{z}}_l)$ subject to the constraints $\hat{g}(\hat{\mathbf{z}}_h, \hat{\mathbf{z}}_l)$. Let now $S(\hat{\mathbf{z}}_h)$ denote the solution set of the lower level problem for a given high level decision vector $\hat{\mathbf{z}}_h$ and $\hat{\mathbf{z}}_l(\hat{\mathbf{z}}_h)$ be an element of $S(\hat{\mathbf{z}}_h)$. Then, assuming that the lower level solution has at most one optimal solution¹, the leader chooses $\hat{\mathbf{z}}_h$ such that the high level objective

¹For more details on how to find a solution to the problem in case of a non-unique lower level optimal solution see e.g. Sinha et al. (2018).

function $\hat{F}(\hat{\mathbf{z}}_h, \mathbf{S}(\hat{\mathbf{z}}_h))$ is minimized subject to the constraints $\hat{H}(\hat{\mathbf{z}}_h, \mathbf{S}(\hat{\mathbf{z}}_h))$ (Ye and Zhu 2010). Formally the described bilevel programming problem reads²:

$$\min_{\hat{\mathbf{z}}_h, \hat{\mathbf{z}}_l} \hat{F}(\hat{\mathbf{z}}_h, \hat{\mathbf{z}}_l) \quad (7.1.1a)$$

$$\text{s.t.} \quad \hat{\mathbf{z}}_l \in \mathbf{S}(\hat{\mathbf{z}}_h), \quad (7.1.1b)$$

$$\hat{H}_j(\hat{\mathbf{z}}_h, \hat{\mathbf{z}}_l) \geq 0, \quad j = 1, 2, \dots, \Gamma \quad (7.1.1c)$$

with the corresponding lower level problem

$$\arg \min_{\hat{\mathbf{z}}_l} \hat{f}(\hat{\mathbf{z}}_h, \hat{\mathbf{z}}_l) \quad (7.1.2a)$$

$$\text{s.t.} \quad \hat{g}_i(\hat{\mathbf{z}}_h, \hat{\mathbf{z}}_l) = 0, \quad i = 1, 2, \dots, \Omega. \quad (7.1.2b)$$

The theory of bilevel programming constitutes the mathematical background to the proposed hierarchical trajectory optimization framework from section 3.3. It is hence represented by the bilevel programming problem as stated in equation 7.1.1 in which the lower level problem 7.1.2 basically corresponds to the minimum kinematics formulation of the optimal interpolation as given in equation 6.3.12.

As already mentioned the hierarchical trajectory optimization framework allows for the separation of breakpoint elements into three different types. For already known values prior to the trajectory generation there is no need to optimize, such that as a consequence these values are no part of the decision variables. This can for example be the case in a stopping scenario with a fix target position, in which the velocity and successive derivatives have to be zero. The set of decision variables to generate the trajectory is partitioned between low level decision variables $\hat{\mathbf{z}}_l$ and high level decision variables $\hat{\mathbf{z}}_h$, which are determined to serve optimally on their level of hierarchy. The high level decision variables represent the main source for optimization, essentially determining the shape of the ego vehicle trajectory. The low level optimization problem is then solved with respect to the low level decision vector, while the high level decision variables act as parameters. An advantage of the hierarchical trajectory optimization framework in terms of bilevel programming is that in contrast to bicriteria optimization, no compromise between objectives has to be found. Instead the adjustability increases since the clear separation in hierarchical levels obviates the joint consideration of high and low objectives that might (in part) be contradictory.

7.2. Formulation of the Trajectory Optimization Problem

The overall goal is to set up and solve a trajectory optimization problem in a way that it provides the best trajectory in terms of the desired functionality with respect to automated driving applications.

As shown in the previous section the idea of the proposed two level hierarchical trajectory optimization framework that involves the separation in high and low level

²The notation of the constraints of the bilevel programming problem has been tailored to the considered use case. Generally, equality and inequality constraints can be considered for the high level, as well as for the low level optimization problem.

problems disembogues in a bilevel programming problem. In this work a spline-based trajectory representation is chosen and the trajectory is then generated by optimal interpolation. The low level problem 7.1.2 hence corresponds to the minimum kinematics formulation as described in section 6.3, providing a unique optimal solution to the interpolation problem. According to Dempe (2002) in case the lower level problem has a unique optimal solution for all parameter values, the bilevel programming problem 7.1.1 is equivalent to a single level optimization problem that has an implicitly defined objective function. The bilevel programming problem can then be seen as the task to find the best solution in the set of the optimal solutions to the lower level problem with respect to the high level objective function. Since the optimal low level decision variables $\hat{\mathbf{z}}_l$ can basically be considered as a function of the high level decision variables $\hat{\mathbf{z}}_h$, a single level problem can be constructed omitting $\hat{\mathbf{z}}_l$ completely (Sinha et al. 2018):

$$\min_{\hat{\mathbf{z}}_h} \hat{F}(\hat{\mathbf{z}}_h) \quad (7.2.1a)$$

$$\text{s.t.} \quad \hat{H}_j(\hat{\mathbf{z}}_h) \geq 0, \quad j = 1, 2, \dots, \Gamma. \quad (7.2.1b)$$

From the considerations made in section 5.1.2 system dynamics can be expressed in the flat output space. This enables to formulate the trajectory optimization problem with respect to the flat output variable z . As stated in chapter 6 it is beneficial to apply a spline-based formulation of the trajectory, since the required derivatives to obtain the control inputs via equation 5.1.30 can be derived straight forward. The high level decision variables $\hat{\mathbf{z}}_h$ hence correspond to the respective spline breakpoint elements. For the two level hierarchical trajectory planning a separate spline for longitudinal and lateral direction is chosen, covering a two-dimensional plane for vehicle motion. The trajectory is then generated by optimal interpolation and combining the longitudinal and lateral result (see chapter 6).

Looking at the remaining problem 7.2.1 to solve in that respect the objective function \hat{F} is composed of the objectives mentioned in section 5.3. The combined costs read:

$$\hat{F}(\hat{\mathbf{z}}_h) = \omega_d \hat{F}_d(\hat{\mathbf{z}}_h) + \omega_v \hat{F}_v(\hat{\mathbf{z}}_h) + \omega_p \hat{F}_p(\hat{\mathbf{z}}_h) + \omega_c \hat{F}_c(\hat{\mathbf{z}}_h) \quad (7.2.2)$$

with costs for distance keeping \hat{F}_d , reference speed control \hat{F}_v , reference path following \hat{F}_p and comfort driving \hat{F}_c and their respective weights ω to balance the importance between the objectives. To ensure a certain safety distance to other vehicles, an objective is introduced that produces ACC-like behavior. Therefore, the distance to the respective predicted impeding vehicles (cmp. section 4.1) for the planned ego lane are taken.

The minimum reference distance that should be obtained to the lead vehicle ${}^C\check{d}_l$ and to the tail vehicle ${}^C\check{d}_t$ is given by:

$${}^C\check{d}_{l,k}(\hat{\mathbf{z}}_h) = d_{\min} + T_l {}^C v_{x,k}(\hat{\mathbf{z}}_h), \quad (7.2.3)$$

$${}^C\check{d}_{t,k}(\hat{\mathbf{z}}_h) = -(d_{\min} + T_t {}^C v_{x,k}(\hat{\mathbf{z}}_h)) \quad (7.2.4)$$

with the respective constant time gaps T_l and T_t and minimum safety distance d_{\min} . Note that the ego velocity v is expressed in curvilinear coordinates to merely account for the velocity that is directed along the course of the road, which has been identified

to be the most appropriate measure for the relevant distance in the considered case. The ego velocity is dependent on the high level decision variables $\hat{\mathbf{z}}_h$, since these effect the general shape of the trajectory and hence the planned velocity of the ego vehicle. The objective of distance keeping is then set up by comparing the actual distances to the lead and tail vehicle ${}^c d_l$ and ${}^c d_t$ to the reference distances calculated in equation 7.2.3 and equation 7.2.4:

$$\bar{d}_{l,k}(\hat{\mathbf{z}}_h) = \begin{cases} ({}^c \check{d}_{l,k}(\hat{\mathbf{z}}_h) - {}^c d_{l,k}(\hat{\mathbf{z}}_h)) / {}^c \check{d}_{l,k}(\hat{\mathbf{z}}_h) & {}^c \check{d}_{l,k}(\hat{\mathbf{z}}_h) - {}^c d_{l,k}(\hat{\mathbf{z}}_h) > 0 \\ 0 & \text{otherwise} \end{cases}, \quad (7.2.5)$$

$$\bar{d}_{t,k}(\hat{\mathbf{z}}_h) = \begin{cases} ({}^c d_{t,k}(\hat{\mathbf{z}}_h) - {}^c \check{d}_{t,k}(\hat{\mathbf{z}}_h)) / {}^c \check{d}_{t,k}(\hat{\mathbf{z}}_h) & {}^c d_{t,k}(\hat{\mathbf{z}}_h) - {}^c \check{d}_{t,k}(\hat{\mathbf{z}}_h) > 0 \\ 0 & \text{otherwise} \end{cases}, \quad (7.2.6)$$

also taking into account that only costs need to be considered, which violate the reference safety distance. Because both, the actual and the reference distance to the lead and tail vehicle are based on the planned trajectory derived via the high level decision variables $\hat{\mathbf{z}}_h$ the objective of distance keeping introduces a predictive behavior of the ego vehicle. The accumulated costs for the distance keeping objective are:

$$\hat{F}_d(\hat{\mathbf{z}}_h) = \sum_{k=0}^K \bar{d}_{l,k}^2(\hat{\mathbf{z}}_h) + \sum_{k=0}^K \bar{d}_{t,k}^2(\hat{\mathbf{z}}_h). \quad (7.2.7)$$

To attain the desired reference speed the objective reads:

$$\hat{F}_v(\hat{\mathbf{z}}_h) = \sum_{k=0}^K ({}^c v_{x,k}(\hat{\mathbf{z}}_h) - {}^c \tilde{v}_{x,k})^2, \quad (7.2.8)$$

which penalizes the deviation of the planned longitudinal ego velocity v_x from reference speed profile \tilde{v}_x provided by the environment-aware maneuver planning (see chapter 4). The difference to penalizing a deviation of absolute velocities is hence that by taking the curvilinear longitudinal velocity into account, the motion of the ego vehicle is directed along the road. That way a desired progress along the road should be ensured, inferring that the vehicle orientation with respect to the lanes is minimized as well.

The objective to reach a lateral reference position encompasses the determination of a reasonable time at which the desired lateral position should be attained. This allows for an uncommitted transition, considering that the maximum requested lateral offset $\tilde{p}_{y,\max}$ cannot be reached instantaneously. Then, with comfort acceleration a_{comf} and floor function $\lfloor \cdot \rfloor$ the target lane reaching time T_r and the corresponding trajectory index k_r are:

$$T_r = \sqrt{\frac{2|\tilde{p}_{y,\max}|}{a_{\text{comf}}}}, \quad k_r = \lfloor (T_r/T_s + 1) + 0.5 \rfloor. \quad (7.2.9)$$

The objective to reach the desired lateral position is intended to incorporate high level maneuver decisions in terms of the preferred driving lane. In the particular case considered in this thesis the reference lateral path is deduced from the maneuver

trajectory provided by the environment-aware maneuver planning (see chapter 4). The objective reads:

$$\hat{F}_p(\hat{\mathbf{z}}_h) = \sum_{k=k_r}^K {}^C p_{y,k}^2(\hat{\mathbf{z}}_h) = \sum_{k=k_r}^K \left({}^C p_{y,k}(\hat{\mathbf{z}}_h) - {}^C \tilde{p}_y({}^C p_{x,k}(\hat{\mathbf{z}}_h)) \right)^2. \quad (7.2.10)$$

Here, ${}^C p_y$ is the path-trajectory deviation in terms of the lateral offset of the ego trajectory in curvilinear coordinates with the lateral reference path serving as the curvilinear basis. This way it is ensured that the shortest distance is taken into consideration exclusively. An alternative formulation is given by performing a comparison of lateral ego vehicle position ${}^C p_y$ and reference position ${}^C \tilde{p}_y$ in curvilinear coordinates with respect to the same curvilinear basis (i.e. the course of the road), retaining the property of only considering the shortest lateral distance between the reference path and the ego trajectory. To calculate the path-trajectory deviation the effective lateral reference position of the reference path in the corresponding curvilinear frame ${}^C \tilde{p}_y$ is given in dependence of the curvilinear longitudinal position of the ego vehicle ${}^C p_x$.

The objective of comfort driving is given in terms of acceleration and makes use of:

$$\bar{a}_{x,k}(\hat{\mathbf{z}}_h) = \begin{cases} (|{}^N a_{x,k}(\hat{\mathbf{z}}_h)| - \check{a}_x) / \check{a}_x & |{}^N a_{x,k}(\hat{\mathbf{z}}_h)| - \check{a}_x > 0 \\ 0 & \text{otherwise} \end{cases}, \quad (7.2.11)$$

$$\bar{a}_{y,k}(\hat{\mathbf{z}}_h) = \begin{cases} (|{}^N a_{y,k}(\hat{\mathbf{z}}_h)| - \check{a}_y) / \check{a}_y & |{}^N a_{y,k}(\hat{\mathbf{z}}_h)| - \check{a}_y > 0 \\ 0 & \text{otherwise} \end{cases} \quad (7.2.12)$$

to incorporate a penalty in case the comfort longitudinal acceleration \check{a}_x or the comfort lateral acceleration \check{a}_y is exceeded, respectively. The longitudinal acceleration ${}^N a_x$ and lateral acceleration ${}^N a_y$ are derived from the planned ego trajectory by a transformation into natural coordinates. This basically corresponds to a two-dimensional rotation around the planned heading angle of the ego vehicle. The objective then reads:

$$\hat{F}_c(\hat{\mathbf{z}}_h) = \sum_{k=0}^K \bar{a}_{x,k}^2(\hat{\mathbf{z}}_h) + \sum_{k=0}^K \bar{a}_{y,k}^2(\hat{\mathbf{z}}_h). \quad (7.2.13)$$

The calculation of objectives involves a curvilinear transformation of the ego trajectory with respect to a reference lane marker. This is necessary without any alternative to formulate the desired behavior with accurate precision. Particularly the determination of distances profits from curvilinear representation, as relevant measures naturally coincide with the traffic flow.

Inequality constraints arise from the vehicle model as derived in section 5.1 and the environment as given in section 5.2. Because of the fact that the breakpoint time might be part of the decision variables it is essential to require monotonically increasing time. For breakpoint time t_κ with spline segment index $\kappa = 0, 1, \dots, \mathcal{K}$ and the minimum desired time difference T_m this is covered by:

$${}_x \hat{H}_{1,\kappa}(\hat{\mathbf{z}}_h) = t_{x,\kappa+1} - t_{x,\kappa} - T_m, \quad (7.2.14)$$

$${}_y \hat{H}_{1,\kappa}(\hat{\mathbf{z}}_h) = t_{y,\kappa+1} - t_{y,\kappa} - T_m \quad (7.2.15)$$

for the lateral and longitudinal spline, respectively.

For obstacles $q = 1, 2, \dots, F$ the inequality accounting for collision avoidance reads:

$${}_q\hat{H}_{2,k}(\hat{\mathbf{z}}_h) = {}_q d_k(\hat{\mathbf{z}}_h), \quad k = 0, 1, \dots, K \quad (7.2.16)$$

with distance function $d_k(\hat{\mathbf{z}}_h)$ that calculates the shortest distance to each obstacle vehicle q . As described in section 5.2 there are several possibilities to approximate the vehicle shape in combination with an efficient distance calculation.

The circle approximation has been chosen, because it provides an efficient and accurate way to perform collision checking. Relevant calculations are therefore given in equation 5.2.1. For $\mathring{\eta}$ approximating circles collision checking hence disembogues into a distance calculation between each approximating circle center ${}^F_{\varkappa}\hat{\mathbf{p}} = [{}^F_{\varkappa}\hat{p}_x \quad {}^F_{\varkappa}\hat{p}_y]^T$ with $\varkappa = 1, 2, \dots, \mathring{\eta}$ for the ego vehicle and obstacle approximating circle centers ${}^F_{\nu}\hat{\mathbf{p}} = [{}^F_{\nu}\hat{p}_x \quad {}^F_{\nu}\hat{p}_y]^T$ with $\nu = 1, 2, \dots, \mathring{\eta}$ for each obstacle vehicle q :

$${}_{\varkappa,\nu,q}d_{x,k}(\hat{\mathbf{z}}_h) = {}^F_{\varkappa}\hat{p}_{x,k}(\hat{\mathbf{z}}_h) - {}^F_{\nu,q}\hat{p}_{x,k}, \quad (7.2.17)$$

$${}_{\varkappa,\nu,q}d_{y,k}(\hat{\mathbf{z}}_h) = {}^F_{\varkappa}\hat{p}_{y,k}(\hat{\mathbf{z}}_h) - {}^F_{\nu,q}\hat{p}_{y,k}. \quad (7.2.18)$$

Because of the geometric properties of a circle the final inequality condition for collision avoidance is given by:

$${}_q d_k(\hat{\mathbf{z}}_h) = \min_{\varkappa,\nu} \sqrt{{}_{\varkappa,\nu,q}d_{x,k}^2(\hat{\mathbf{z}}_h) + {}_{\varkappa,\nu,q}d_{y,k}^2(\hat{\mathbf{z}}_h)} - (\hat{\rho}_{\text{ego}} + \hat{\rho}_q). \quad (7.2.19)$$

Note that because of the decomposition of the vehicle shape into multiple circles, the inequality constraint ${}_q\hat{H}_{2,k}(\hat{\mathbf{z}}_h)$ thus becomes dependent on each approximating circle \varkappa and ν for ego and obstacle vehicles, respectively. In this work each vehicle is represented by $\mathring{\eta} = 3$ circles, which leads to $3 \cdot 3 = 9$ distance calculations that have to be performed for each trajectory time step k .

The formulation of the road boundary constraints corresponds to the definition of a driving corridor that denotes the negotiable space i.e. the available lanes for the ego vehicle considering the intended driving direction.

It relies on the road model as given in section 5.2, in which all lane markers are represented by polygonal chains. The shortest distance between the restricting lane marker and a bounding box corner point of the ego vehicle ${}^F_{\omega}\hat{\mathbf{p}}$ with $\omega = 1, 2, 3, 4$ is hence given by the lateral component in curvilinear coordinates calculated with respect to the corresponding lane marker³. It is:

$${}^C_l\check{p}_{y,k}(\hat{\mathbf{z}}_h) = \max_{\omega} {}^C_{l,\omega}\check{p}_{y,k}(\hat{\mathbf{z}}_h), \quad (7.2.20)$$

$${}^C_r\check{p}_{y,k}(\hat{\mathbf{z}}_h) = \min_{\omega} {}^C_{r,\omega}\check{p}_{y,k}(\hat{\mathbf{z}}_h), \quad (7.2.21)$$

which finally leads to a constraint for the left lane and the right lane boundary:

$${}_l\hat{H}_{3,k}(\hat{\mathbf{z}}_h) = -{}^C_l\check{p}_{y,k}(\hat{\mathbf{z}}_h), \quad (7.2.22)$$

$${}_r\hat{H}_{3,k}(\hat{\mathbf{z}}_h) = {}^C_r\check{p}_{y,k}(\hat{\mathbf{z}}_h). \quad (7.2.23)$$

³Note the relation to the definition of curvilinear coordinates, since the orthogonal component in curvilinear frame corresponds to the shortest distance to the reference lane marker.

The minus sign for the left boundary accounts for the direction dependency as calculated orthogonal distances are counted positive, if they are on the left side of the polygonal chain. Figure 7.1 illustrates the road boundary constraint calculation.

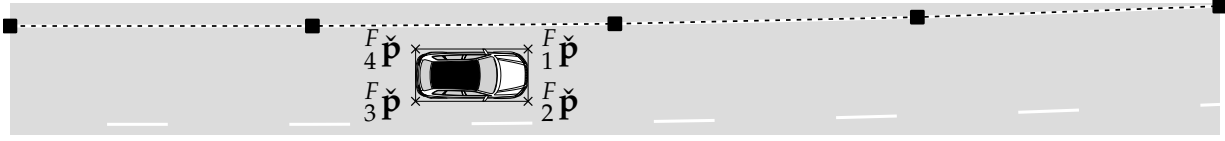


Figure 7.1.: Road boundary constraints are defined via a shortest distance calculation between the polygonal chain lane marker and the corner points of the ego vehicle bounding box.

For the system dynamics inequality constraints the control inputs are calculated as described in section 5.1.2. This enables to consider system input bounds that can be inferred from the vehicle system. Constraints accounting for the limited longitudinal acceleration $a_{\hat{H}_4}$ and limited steering angle $\delta_{\hat{H}_4}$ read:

$$a_{\hat{H}_4,k}(\hat{\mathbf{z}}_h) = a_{\max}(^F v_k) - {}^N a_{x,k}(\hat{\mathbf{z}}_h), \quad (7.2.24)$$

$$\delta_{\hat{H}_4,k}(\hat{\mathbf{z}}_h) = \delta_{\max} - |\delta_{r,k}(\hat{\mathbf{z}}_h)|. \quad (7.2.25)$$

The potential longitudinal acceleration that is available for the ego vehicle is given by a_{\max} and dependent on the current ego velocity $^F v$. It is designed to follow the speed-torque characteristic of the vehicle engine and hence limits the ability to further accelerate at higher speeds. Figure 7.2 shows the chosen characteristic. Lateral

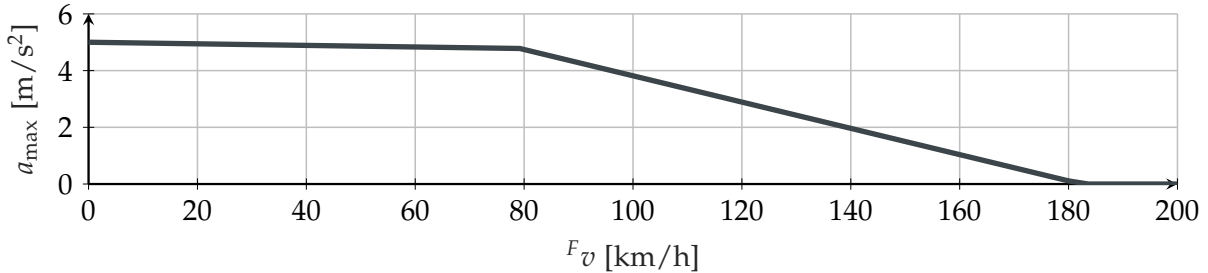


Figure 7.2.: Approximation of maximum longitudinal acceleration over the vehicle speed.

motion is limited by the maximum steering angle δ_{\max} . To account for the combined longitudinal and lateral restriction in terms of acceleration, the limit a_g is based on considerations with respect to the circle of forces (see e.g. Pacejka (2007)) leading to:

$${}_g \hat{H}_{4,k}(\hat{\mathbf{z}}_h) = a_g - |{}^N a_k(\hat{\mathbf{z}}_h)|. \quad (7.2.26)$$

Especially the property of an unambiguous optimal interpolation solution allows to set up a single level optimization problem 7.2.1 that is left as the remaining challenge to generate the optimal trajectory. At this point the similarity between a nonlinear program that can be deduced from the corresponding optimal control problem and the final bilevel programming problem becomes obvious.

Since the final bilevel problem 7.2.1 matches the ordinary formulation of a trajectory optimization problem, classical approaches as stated in chapter 2 can be applied for trajectory generation. In this thesis a discrete as well as a continuous approach are presented.

7.3. Discrete Trajectory Optimization

The class of sampling-based planning approaches is referred to as discrete trajectory optimization, which relates to the process of constrained numerical optimization over a discretized parameter space. The challenge is to find a suitable set of trajectory candidates by applying a sampling strategy in order to discretize the state space in form of a graph. As the sampling of longitudinal and lateral trajectory breakpoints explicitly considers time information the resulting graph naturally becomes a *Directed Acyclic Graph* (DAG). This is because subsequent breakpoints can only be connected given that the time is increasing. Since for trajectory generation in the presence of the two level hierarchical trajectory optimization framework a separate spline for longitudinal and lateral direction is defined, the sampling strategy follows this policy accordingly. Longitudinal and lateral trajectories are sampled independently and then combined at a later step. Finally, trajectory candidates need to be evaluated and the optimal trajectory with respect to given high level objectives and constraints has to be found within the set of sampled trajectory candidates. Hence, the chosen approach can be labeled as trajectory rollout (see section 2.1).

7.3.1. Previous Work

In previous work a sample-based planner for on-road automated driving has been presented that performs online generation and evaluation of suitable trajectory candidates (Lienke et al. 2018b). The idea is to generate a meaningful set of trajectory candidates starting from the current vehicle state and using a distinct analytic interpolation strategy and curvilinear transform of breakpoints to increase performance in terms of efficiency and generalizability for different scenarios.

The chosen architecture gives more responsibility to the trajectory planning layer intended to avoid inconsistencies between behavior and trajectory planning. The maneuver planning is performed in terms of a target state generation that represents the desired high level behavior. The chosen trajectory sampling strategy further features the generation of maneuver hypotheses defined with respect to the planned lateral motion of the ego vehicle. This way the sampling set contains several lane keeping and lane changing maneuvers. Selecting the best trajectory in terms of the objective function on trajectory level then goes along with a maneuver decision, as each sampled trajectory is associated to a dedicated maneuver class. To enable a reliable decision of the automated vehicle it is hence important to design the trajectory objective function in a comprehensive manner.

For trajectory set generation a mixed sampling strategy is applied, sampling from action as well as from state space. By sampling the longitudinal acceleration input a better sense of feasibility is established on the one hand, whereas lateral sampling with respect to the detected lanes on the other hand permits to exploit the structure of the environment. The idea behind the latter aspect is to determine breakpoints in close relation to the course of the road, which helps to generate expressive trajectory candidates. The use of a curvilinear coordinate system is ideally suited for this purpose. Sampling is hence performed in curvilinear coordinates and polynomial lane markers

then serve as reference for the transformation in the vehicle frame. To keep the number of curvilinear transformations as low as possible, candidate trajectories are interpolated and evaluated in vehicle frame. Instead of transforming every trajectory point of each sampled trajectory this allows to transform only sampled breakpoints and additionally obviates time consuming point wise transformation of obstacle trajectories. The curvilinear transformation establishes a basic universality, since the planned trajectories are independent of road curvature and work in straight as well as in curved lane scenarios.

The design of the architecture is intended to cover maneuver and trajectory planning in one integrated step by purposely generating trajectory samples for different lateral maneuver classes. Although not followed in this thesis the architecture as shown in section 3.1 would also allow to plan for different maneuver hypothesis⁴ to account for disadvantageous high level decisions. However, the focus of the presented trajectory planning approach in this thesis lies on determining the optimal trajectory with respect to the desired behavior, which is *inter alia* derived from long-term considerations.

Another salient difference can be found in the proposed sampling strategy. Since in the previous work a distinct interpolation strategy is applied, it is important that all kinematic quantities are sampled in consistency to each other in order to generate a meaningful trajectory sample. Otherwise this will likely lead to uncomfortable or undrivable trajectories. This restricts the approach to a constant acceleration assumption that accounts for the fact that the choice of a varying acceleration over time increases the number of possible solutions, while complicating the process of determining an appropriate sampling strategy for each scenario at the same time. The application of optimal interpolation as described in section 6.3 significantly improves on the applied sampling strategy. The major advantage can be seen in the fact that it enables the possibility to harmonize all kinematic quantities at each knot, because they are able to automatically adapt in accordance to the minimum kinematics formulation. This leads to a more flexible design of sampling, allowing e.g. varying accelerations, without taking the risk of producing uncomfortable or even undrivable trajectories. All in all this leads to a larger set of expressive trajectory candidates and a more intuitive design of trajectory sampling.

7.3.2. Sampling Strategy and Trajectory Evaluation

An adequate sampling strategy is vital in order to tackle the arising single level problem that finally represents the mathematical framework of the two level hierarchical trajectory optimization. The task is hence to find the optimal high level decision variables that minimize the high level objective function subject to the high level constraints. With respect to the discrete trajectory optimization approach, the challenge is to find a suitable discretization that leads to expressive trajectory candidates.

In terms of the two level hierarchy the sampling strategy needs to consider the separation in high and low level decision variables, which is relevant to appropriately account for the performed interpolation. The spline-based trajectory representation moreover

⁴This would mean to include feedback from the trajectory planning module to the high level decision making in some way.

offers the possibility to perform the sampling in terms of breakpoints. Obviously, as sampling is done before the interpolation, it is vital to sample coherent breakpoints. The connection of kinematic values of one breakpoint should hence account for this, which means that position, velocity and acceleration should be chosen in order to reflect the desired driving behavior subject to the restrictions of the vehicle dynamics. To enforce the independence of the sampling strategy with respect to road curvature, breakpoints are sampled in curvilinear coordinates. Lane markers then serve as reference for transformation in the vehicle frame. There are several possibilities of handling the chronology of optimal interpolation, transformation and evaluation. However, the most promising one is to transform the breakpoints and interpolate and evaluate the trajectories in vehicle frame afterwards, since it prevents from several time consuming point wise transformations (i.e. obstacle and/or ego trajectories).

The sampling strategy makes use of an adaptive discretization which has been proven to work well in several applications with respect to automated driving (Lienke et al. 2018b; Homann et al. 2018; Homann et al. 2019). For more details it is referred to section A.8.

Longitudinal Sampling Strategy

A deterministic sampling is performed in order to build a tree structure that represents the set of longitudinal trajectories. The tree structure accounts for the fact that a preceding node of a preceding tree level influences the driving behavior, since due to the applied interpolation strategy it affects the shape of the trajectory in the respective segment. This linkage forbids to arbitrarily combine each and every node, because this will likely lead to undesired driving behavior or undrivable trajectories and hence increase the number of inexpressive candidates.

To properly map a sampling strategy to the task of high level optimization of spline breakpoint elements without changing the number of segments of the spline, a tree level is built for each spline breakpoint. Each breakpoint hence corresponds to a tree level, which itself is subdivided into several time layers. This is important because only nodes of successive tree levels are allowed to be connected in order to ensure an increasing breakpoint time yielding a DAG, while maintaining the total number of spline segments at the same time. Since time takes on a special role the focus lies on kinematic quantities like position, velocity and acceleration as meaningful sampling quantities. One basic guideline to build the tree is that each node should represent a variation in only one of these kinematic quantities. This measure improves on the tracking of variations and simplifies the generation of suitable node sequences. The strategy for node generation is now that each high level decision variable is varied by sampling, leading to one node for each sample with the kinematic quantities chosen to yield consistent node values. Connections are established exclusively between nodes of the respective kinematic quantity, but the applied strategy could also be extended to work for all nodes (generated by the samples of each kinematic quantity) of the previous tree level in general. For node generation and connection time takes on a special role in the sampling strategy. This is because it obviously has a major impact on the shape of the trajectory and it is hence necessary to thoroughly consider this

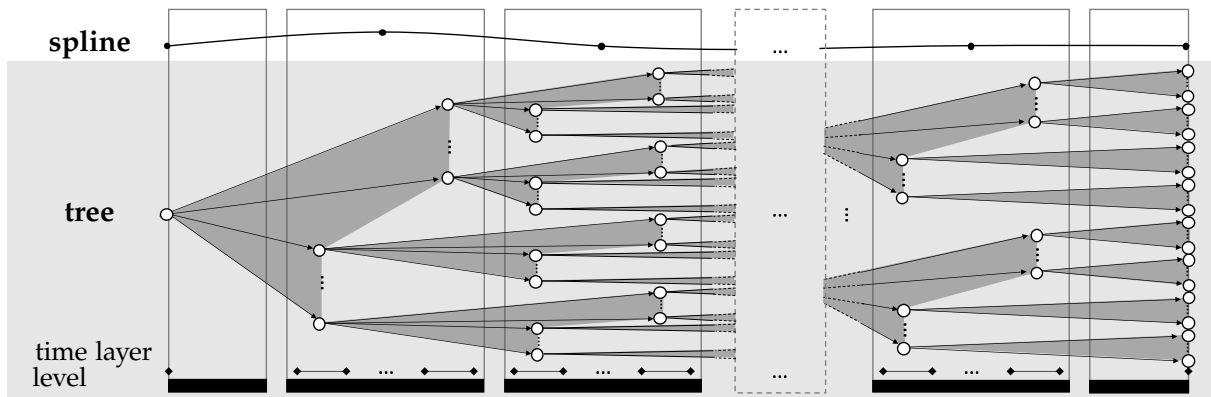


Figure 7.3.: Longitudinal Kinematic Sampling Tree.

relation with respect to the choice of the kinematic sampling quantities. To account for this aspect the sampling process is defined in a way that the sampled time instance influences the choice of the kinematic quantity values. Generally, the concept of adaptive discretization is used to calculate the sampling values. For sampling of the kinematic quantities the start values given by a node of the preceding tree level should be considered for the same reason as for the choice of tree structure, since the intention of both is to ensure coherent node sequences that generate appropriate trajectory segments. The determination of a sampling range for a kinematic quantity is supported by the use of a quasi-reachability analysis⁵ that approximates the states that can be realized from the current start values considering a given acceleration input range. This way the aspect of drivability is included in the process of trajectory candidates generation. When sampling a kinematic quantity the corresponding values of the remaining kinematic quantities of a node are found by integration or derivation, respectively. However, these values can be regarded as some kind of initial values and will not be considered (since they are configured to belong to the low level decision variables they will change with respect to the optimal interpolation), but used as start values for the subsequent tree level.

Figure 7.3 illustrates the longitudinal kinematic sampling tree. In dependence on the number of spline breakpoints a predefined number of samples is generated for each associated tree level. The set of longitudinal trajectory candidates is finally composed of all trajectory samples that are extracted as all paths that go from the root to a leaf of the longitudinal kinematic sampling tree.

Lateral Sampling Strategy

The use of curvilinear coordinates rigorously simplifies the sampling for the lateral quantities, because the respective values can be set independently from the course of the road. This measure hence generalizes the applied lateral sampling strategy to a

⁵In contrast to a reachability analysis as defined by Althoff (2010), the intention is not to determine a set of all possible states that the system (i.e. ego vehicle) can reach, but rather to provide a certain range of meaningful kinematic values in order to obtain an appropriate sequence of nodes through the tree.

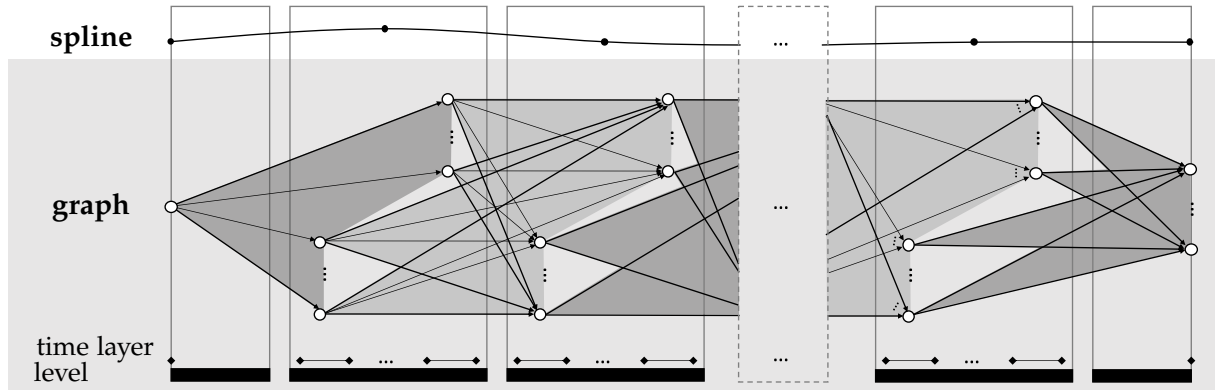


Figure 7.4.: Lateral Kinematic Sampling Graph.

wide range of various scenarios. For the lateral sampling strategy the assumption that the vehicle dynamics restrict the ego vehicle's motion is assumed to play a subordinated role. As a consequence, the consideration of quasi-reachability can be neglected and nodes of different levels can be combined arbitrarily. This leads to the characteristic of fully connected levels, such that finally a lateral kinematic sampling graph is generated. The discretization of kinematic values is done analogous to the longitudinal sampling strategy. The kinematic quantities such as position, velocity and acceleration are adaptively discretized around the actual breakpoint values. The difference to the generation of longitudinal sampling nodes is that lateral nodes are generated as a combination of all sampled kinematic quantity values. Figure 7.4 illustrates the resulting lateral kinematic sampling graph.

Trajectory Evaluation

Although lateral and longitudinal trajectories are constructed as a tree and a graph, no dedicated approach is applied for evaluation purposes. The major reason can be seen in the combinatorial problem as each longitudinal trajectory is combined with each lateral trajectory and not evaluated independently of each other. This finally results in an exhaustive search that has to be performed for each generated sample trajectory. Trajectories, which violate the constraints are eliminated from the trajectory sampling set, whereas remaining trajectories are ranked in accordance to the objective function costs. The trajectory with the lowest costs is then regarded as the optimal solution to the specified trajectory optimization problem 7.2.1.

7.4. Continuous Trajectory Optimization

Optimization-based planning approaches are within this thesis also referred to as continuous trajectory optimization to account for the continuous character of the optimal parameters. Hence, the solution space for trajectories tends towards infinity and accuracy is not restricted by the quantization error coming along with discretization. This chapter elaborates on a continuous trajectory optimization approach to automated

driving in dynamic environments, giving some details about techniques for nonlinear constrained optimization.

7.4.1. Previous Work

In previous work related to continuous trajectory optimization a major research effort has been dedicated to trajectory planning with spline-based trajectory representation using a distinct interpolation strategy (Götte et al. 2017c; Götte et al. 2017d). The trajectory optimization problem is solved by integrating the constraints into the objective function. This way the nonlinear program is transferred to an approximate, non-restricted optimization problem. The resulting nonlinear least-squares problem is solved with the *Levenberg-Marquardt* algorithm (Levenberg 1944; Marquardt 1963). Especially the comparison of trajectory optimization with interpolation strategy and without shows that it is possible to improve on the convergence. This means that while reducing the computational burden due to the consideration of less decision variables, faster convergence is achieved at the same time.

As continuous optimization approaches require certain characteristics with respect to the problem formulation, some effort has been expended to develop a suitable environment model (Lienke et al. 2018c; Lienke et al. 2019b). A potential field is introduced, which is composed of a static and a dynamic part. The latter exploits the results of the situation analysis as it incorporates the predicted trajectories of the current situation, affecting the safety distance that has to be satisfied. It is shown that without any regard to the developing situation a predetermined safety distance (oversized or undersized) will lead to uncomfortable or risky behavior, respectively. The predictive manner of the developed environment potential field hence significantly improves safety and comfort in complex traffic scenarios.

This thesis improves on several aspects of previous research. The definition of constraints and objectives in their entirety is vital to achieve the desired outcome on the one hand, but aggravates the task of parameter tuning on the other. Although this is a general problem, taken measures reveal the advantages of the proposed two level hierarchical trajectory planning framework. Therein, an optimal interpolation strategy replaces the formerly applied distinct interpolation. The two level hierarchical trajectory planning framework hence addresses the issue of conflicting objectives and facilitates the overall adjustment of the approach. In contrast to previous work the chosen nonlinear optimization algorithm refrains from using a penalty method to integrate constraints into the objective function and considers specified limitations as hard constraints instead. A major aspect in the context of trajectory planning for automated driving can be seen in the modeling of specific requirements leading to dedicated objectives and constraints. The potential field for environment modeling has proven to be beneficial for safe driving, but has not been designed to allow for exact collision checking. It is moreover relying on physical considerations and not defined with respect to the course of the road, such that it can only approximate ACC-like behavior. As a consequence this work adapts the definition of the widely accepted ACC functionality as an objective, paired with a constraint for accurate collision checking. Furthermore, the inclusion of a vehicle model by exploiting differential flatness leads

to a more precise modeling of the vehicle motion since it enables the possibility to account for constraints on the control input variables.

7.4.2. Nonlinear Constrained Optimization

Algorithms for nonlinear constrained optimization problems cover a broad range of applications in e.g engineering, science and economics. The solution of complex nonlinear problems appears to be a difficult task, since the existence of equality and inequality constraints generally complicates the solution procedure. In the last years a lot of research has been done to develop numerical approaches to solve such kind of constrained optimization problems.

According to Nocedal and Wright (2006) there is no standard taxonomy for constrained nonlinear optimization algorithms. Nevertheless, it appears that especially active set and interior point methods are largely present among popular solution approaches. The former approach considers inequalities, which represent one of the main challenges with respect to nonlinear programming, by making a distinction between active and inactive constraints. As a consequence active constraints are consolidated in a set of constraints that are satisfied as equalities at a solution. This leads to the so-called active set that consists of the equality constraints and inequalities for which $\hat{H} = 0$ holds. An inequality constraint is hence said to be active if it is $\hat{H} = 0$ and labeled as inactive if inequality $\hat{H} \geq 0$ is strictly satisfied. Active set methods estimate the active set in order to find a solution of the reduced problem, in which active set constraints are treated as equalities. Interior point methods are also referred to as barrier methods since the constraints are modeled as barrier functions. In an iterative procedure intermediate solutions are generated that stay within the boundaries of the feasible region defined by the inequality constraints. The influence of the barrier term is controlled by the barrier parameter. In that respect barrier function parameters are varied and as soon as the solution is approached, barrier effects are mitigated to increase the accuracy of the solution until the optimum is reached.

In the following approaches to quadratic and nonlinear programming as used in this thesis are presented. A *Sequential Quadratic Programming* (SQP) approach is chosen in order to tackle the high level problem of the proposed two level hierarchical trajectory planning framework. Quadratic programming techniques are essential to solve the emerging SQP subproblems as well as for the low level problem solution represented by the minimum kinematics trajectory generation (see section 6.3.2). For deeper insight into the topic of nonlinear optimization in general it is referred to Nocedal and Wright (2006).

Quadratic Programming

Quadratic programming is concerned with solving a linearly constrained quadratic optimization problem, also denoted as a quadratic program. The quadratic objective function $\hat{f}(\hat{\mathbf{p}}) = 0.5\hat{\mathbf{p}}^T\mathbf{Q}\hat{\mathbf{p}} + \mathbf{q}^T\hat{\mathbf{p}}$ with decision variables $\hat{\mathbf{p}} \in \mathbb{R}^{n_{\hat{\mathbf{p}}}}$ is given by cost matrix $\mathbf{Q} \in \mathbb{R}^{n_{\hat{\mathbf{p}}} \times n_{\hat{\mathbf{p}}}}$ and cost vector $\mathbf{q} \in \mathbb{R}^{n_{\hat{\mathbf{p}}}}$. The linear equality constraints $\hat{g}_i(\hat{\mathbf{p}})$ for $i = 1, 2, \dots, \Omega$ are expressed by constraint matrix $\mathbf{A}_{\hat{\mathbf{g}}} \in \mathbb{R}^{\Omega \times n_{\hat{\mathbf{p}}}}$ and constraint vector

$\mathbf{b}_{\hat{g}} \in \mathbb{R}^{\Omega}$ and inequality constraints $\hat{h}_j(\hat{\mathbf{p}})$ for $j = 1, 2, \dots, \Theta$ are given by inequality constraint matrix $\mathbf{A}_{\hat{h}} \in \mathbb{R}^{\Theta \times n_{\hat{p}}}$ and vector $\mathbf{b}_{\hat{h}} \in \mathbb{R}^{\Theta}$. The quadratic programming problem reads:

$$\min_{\hat{\mathbf{p}}} 0.5\hat{\mathbf{p}}^T \mathbf{Q} \hat{\mathbf{p}} + \mathbf{q}^T \hat{\mathbf{p}} \quad (7.4.1a)$$

$$\text{s.t.} \quad \mathbf{A}_{\hat{g}} \hat{\mathbf{p}} - \mathbf{b}_{\hat{g}} = \mathbf{0}, \quad (7.4.1b)$$

$$\mathbf{A}_{\hat{h}} \hat{\mathbf{p}} - \mathbf{b}_{\hat{h}} \geq \mathbf{0}. \quad (7.4.1c)$$

Quadratic programming problems often arise as subproblems within methods that deal with finding a solution for general constrained optimization problems. As indicated by the name this is especially true for sequential quadratic programming techniques, which are designed to cope with nonlinear programming problems.

There are several approaches to quadratic programming. Betts (2010) for example outlines an active set method for solving a quadratic program. Stellato et al. (2020) propose an *Operator Splitting Quadratic Program* (OSQP) solver that is based on the *Alternating Direction Method of Multipliers* (ADMM). According to Stellato et al. (2020) the most computationally expensive part in the algorithm is the step of solving the linear system that originates from iteratively solving the optimization problem with ADMM. The linear system is solved by either a direct or indirect method. The latter relies on an iterative procedure, whereas the direct method involves the factorization of the coefficient matrix followed by forward and backward substitution.

Since benchmark results show that OSQP outperforms state of the art quadratic programming solvers in terms of timing and failure rate, OSQP becomes the algorithm of choice for quadratic programming in this thesis.

Sequential Quadratic Programming

A general constrained optimization problem that has nonlinear terms in either the objective or its constraint function is given by:

$$\min_{\hat{\mathbf{z}}} \hat{F}(\hat{\mathbf{z}}) \quad (7.4.2a)$$

$$\text{s.t.} \quad \hat{G}_i(\hat{\mathbf{z}}) = 0, \quad i = 1, 2, \dots, \Lambda, \quad (7.4.2b)$$

$$\hat{H}_j(\hat{\mathbf{z}}) \geq 0, \quad j = 1, 2, \dots, \Gamma. \quad (7.4.2c)$$

Approaches like active set sequential quadratic programming (SQP) and interior point methods that deal with such optimization problems belong to the field of nonlinear programming. In this thesis an SQP algorithm is applied and the following explanatory notes will hence focus on this approach.

Briefly sketched, an SQP approach models problem 7.4.2 by a quadratic programming subproblem whose solution defines the search direction to determine the next iterate.

The quadratic programming subproblem at iterate $\hat{\mathbf{z}}_i$ reads:

$$\min_{\hat{\mathbf{p}}} \hat{F}(\hat{\mathbf{z}}_i) + \nabla \hat{F}(\hat{\mathbf{z}}_i)^T \hat{\mathbf{p}} + 0.5\hat{\mathbf{p}}^T \nabla_{\hat{\mathbf{z}}}^2 \mathcal{L}(\hat{\mathbf{z}}_i, \hat{\gamma}_i, \hat{\sigma}_i) \hat{\mathbf{p}} \quad (7.4.3a)$$

$$\text{s.t.} \quad \nabla \hat{G}_i(\hat{\mathbf{z}}_i)^T \hat{\mathbf{p}} + \hat{G}_i(\hat{\mathbf{z}}_i) = 0 \quad i = 1, 2, \dots, \Omega, \quad (7.4.3b)$$

$$\nabla \hat{H}_j(\hat{\mathbf{z}}_i)^T \hat{\mathbf{p}} + \hat{H}_j(\hat{\mathbf{z}}_i) \geq 0 \quad j = 1, 2, \dots, \Theta \quad (7.4.3c)$$

with the lagrangian of the nonlinear program given by:

$$\mathcal{L}(\hat{\mathbf{z}}, \hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\sigma}}) = \hat{F}(\hat{\mathbf{z}}) - \sum_{i=1}^{\Lambda} \hat{\gamma}_i \hat{G}_i(\hat{\mathbf{z}}) - \sum_{j=1}^{\Gamma} \hat{\sigma}_j \hat{H}_j(\hat{\mathbf{z}}). \quad (7.4.4)$$

The hessian of lagrangian $\nabla_{\hat{\mathbf{z}}}^2 \mathcal{L}(\hat{\mathbf{z}}, \hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\sigma}})$ contains the second derivatives of the objective function and constraints. For complicated problems it might be useful to replace the hessian matrix by a quasi-Newton approximation.

Inspired by the *Broyden–Fletcher–Goldfarb–Shanno* (BFGS) algorithm a damped BFGS update is applied to iteratively approximate the hessian. Therefore, in a first step vectors $\hat{\mathbf{s}}_i$ and $\hat{\mathbf{y}}_i$ are defined as follows:

$$\hat{\mathbf{s}}_i = \hat{\mathbf{z}}_{i+1} - \hat{\mathbf{z}}_i, \quad \hat{\mathbf{y}}_i = \nabla_{\hat{\mathbf{z}}} \mathcal{L}(\hat{\mathbf{z}}_{i+1}, \hat{\boldsymbol{\gamma}}_{i+1}, \hat{\boldsymbol{\sigma}}_{i+1}) - \nabla_{\hat{\mathbf{z}}} \mathcal{L}(\hat{\mathbf{z}}_i, \hat{\boldsymbol{\gamma}}_{i+1}, \hat{\boldsymbol{\sigma}}_{i+1}). \quad (7.4.5)$$

Then it is:

$$\hat{\mathbf{r}}_i = \hat{\theta}_i \hat{\mathbf{y}}_i + (1 - \hat{\theta}_i) \hat{\mathbf{B}}_i \hat{\mathbf{s}}_i \quad (7.4.6)$$

with scalar $\hat{\theta}$ for balancing the two terms of $\hat{\mathbf{r}}_i$

$$\hat{\theta} = \begin{cases} 1 & \hat{\mathbf{s}}_i^T \hat{\mathbf{y}}_i \geq 0.2 \hat{\mathbf{s}}_i^T \hat{\mathbf{B}}_i \hat{\mathbf{s}}_i \\ (0.8 \hat{\mathbf{s}}_i^T \hat{\mathbf{B}}_i \hat{\mathbf{s}}_i) / (\hat{\mathbf{s}}_i^T \hat{\mathbf{B}}_i \hat{\mathbf{s}}_i - \hat{\mathbf{s}}_i^T \hat{\mathbf{y}}_i) & \hat{\mathbf{s}}_i^T \hat{\mathbf{y}}_i < 0.2 \hat{\mathbf{s}}_i^T \hat{\mathbf{B}}_i \hat{\mathbf{s}}_i \end{cases}. \quad (7.4.7)$$

The BFGS update formula is finally given by:

$$\hat{\mathbf{B}}_{i+1} = \hat{\mathbf{B}}_i - \frac{\hat{\mathbf{B}}_i \hat{\mathbf{s}}_i \hat{\mathbf{s}}_i^T \hat{\mathbf{B}}_i}{\hat{\mathbf{s}}_i^T \hat{\mathbf{B}}_i \hat{\mathbf{s}}_i} + \frac{\hat{\mathbf{r}}_i \hat{\mathbf{r}}_i^T}{\hat{\mathbf{s}}_i^T \hat{\mathbf{r}}_i}. \quad (7.4.8)$$

With respect to the step computation and evaluation of a merit function, inequality constraints are often converted to equalities using slack variables. For the following considerations it is hence assumed that all constraints are in the form of equalities expressed by vector $\hat{\mathbf{a}}$. Likewise the lagrange multipliers for equality and inequality constraints are consolidated in $\hat{\boldsymbol{\lambda}}$. In the applied line search method an ℓ_1 merit function controls the size of the step, helping to decide whether a step is accepted or not. With penalty parameter $\hat{\mu} > 0$ the ℓ_1 merit function $\hat{\phi}$ is defined by:

$$\hat{\phi}(\hat{\mathbf{z}}, \hat{\mu}) = \hat{F}(\hat{\mathbf{z}}) + \hat{\mu} \|\hat{\mathbf{a}}(\hat{\mathbf{z}})\|_1. \quad (7.4.9)$$

A step $\hat{\alpha}_i \hat{\mathbf{p}}_i$ with step length parameter $\hat{\alpha}$ is accepted if the following condition holds:

$$\hat{\phi}(\hat{\mathbf{z}}_i + \hat{\alpha}_i \hat{\mathbf{p}}_i, \hat{\mu}_i) \leq \hat{\phi}(\hat{\mathbf{z}}_i, \hat{\mu}_i) + \hat{\eta} \hat{\alpha}_i \hat{D}(\hat{\phi}(\hat{\mathbf{z}}_i, \hat{\mu}_i), \hat{\mathbf{p}}_i), \quad \hat{\eta} \in [0, 1] \quad (7.4.10)$$

with the directional derivative of merit function $\hat{\phi}$ in the direction of $\hat{\mathbf{p}}_i$ given by:

$$\hat{D}(\hat{\phi}(\hat{\mathbf{z}}_i, \hat{\mu}_i), \hat{\mathbf{p}}_i) = \nabla \hat{F}(\hat{\mathbf{z}}_i)^T \hat{\mathbf{p}}_i - \hat{\mu}_i \|\hat{\mathbf{a}}(\hat{\mathbf{z}})\|_1. \quad (7.4.11)$$

The utilized SQP algorithm is outlined in chapter A.9. To solve subproblem 7.4.3 an approach to quadratic programming as described above is applied.

A major benefit of SQP algorithms for nonlinear programming is that the problem is typically solved in very few iterations. This is a desirable feature, since in dependence on the complexity of the specified problem function evaluations (i.e. collision checking) are presumably computational expensive. An SQP algorithm was for example successfully applied for trajectory planning on the *Bertha Benz drive* (Ziegler et al. 2015), showing the capabilities with respect to the application in the field of automated driving.

8

Evaluation in a Dynamic Environment

This chapter provides details with respect to the general applicability of the developed approaches to automated driving in dynamic environments. The discrete trajectory optimization (see section 7.4) and the continuous trajectory optimization approach (see section 7.3) are evaluated in expressive test cases representing typical on-road traffic scenarios. Finally, the obtained results are summarized and discussed to draw a comprehensive picture.

8.1. Setup of the Trajectory Planning Approaches

The developed algorithms feature a large variety of setting options. This enables precise tuning for custom application, but since performance is strongly connected to the chosen settings on the other hand a well-conceived setup is vital.

Since the trajectory representation is chosen to be a spline, relevant parameters are given in Table 8.1. The number of segments (and thus the number of breakpoints) and the chosen interpolation order affect the flexibility and adaptiveness of the spline. The higher the number of segments, the higher is the expected flexibility. Optimality in terms of minimum kinematics trajectory generation for interpolation accounts for the desired low level behavior and is reflected in the choice of the cost and continuity derivative. For longitudinal direction a 7th order spline composed of two segments is set up. To account for challenges with respect to lateral positioning due to road curvature and potential lateral maneuvers (lane change, overtaking, etc.) the lateral spline is composed of three segments. For both directions continuity in jerk is stipulated. In contrast, with penalty weight ω_r chosen as $\omega_r = 1$ for $r = r_f$ and $\omega_r = 0$ otherwise,

Table 8.1.: Spline definition.

	number of segments \mathcal{K}	interpolation order \mathcal{V}	cost derivative r_f	continuity derivative r_c
longitudinal	2	7	2	3
lateral	3	7	3	3

the interpolated longitudinal trajectory is set up to result in a minimum acceleration solution on low level, whereas in lateral direction the generated minimum kinematics trajectory is jerk optimal. The ego vehicle trajectory \mathbf{P}_{ego} is then characterized by a chosen resolution of $T_s = 0.1$ s with a planning horizon of $T_p = 5$ s. The choice of the trajectory sample time accounts for the aspect of accuracy in terms of related distance calculations and has proven to work well in previous work (see e.g. Götte et al. (2017c) and Lienke et al. (2018b)). The chosen planning horizon is in the range of the duration of single lane changes (see e.g. Toledo and Zohar (2007) and Mullakkal-Babu et al. (2020)). However, with respect to the capabilities of the onboard perception system, the planning horizon can be considered to be quite long as it needs to cope with issues arising from the limited sensor view range.

The decisive setting with respect to the two level hierarchical framework is the choice of low level $\hat{\mathbf{z}}_l$ and high level decision variables $\hat{\mathbf{z}}_h$. In the same line fix values for known reference quantities need to be set. The hierarchical framework configuration strongly relates to performance and run time of the trajectory optimization approaches. Moreover, a use case dependent configuration is implemented to account for the individual demands of a certain maneuver. It is distinguished between a configuration for driving (see Table 8.2) and for target braking (i.e. stopping, see Table 8.3). The

 Table 8.2.: Hierarchical framework configuration *Co-D* for driving.

spline knot		\mathbf{z}_0	\mathbf{z}_1	\mathbf{z}_2		\mathbf{z}_0	\mathbf{z}_1	\mathbf{z}_2	\mathbf{z}_3
position [m]	${}^F p_x$	0	■	■	${}^F p_y$	0	■	■	■
velocity [m/s]	${}^F v_x$	–	□	□	${}^F v_y$	–	□	□	□
acceleration [m/s ²]	${}^F a_x$	–	□	□	${}^F a_y$	–	□	□	0
jerk [m/s ³]	${}^F \dot{a}_x$	–	□	0	${}^F \dot{a}_y$	–	□	□	0
time [s]	t_x	0	■	T_p	t_y	0	■	■	T_p

– : Current value □ : Low level optimization ■ : High level optimization

 Table 8.3.: Hierarchical framework configuration *Co-S* for stop maneuver.

spline knot		\mathbf{z}_0	\mathbf{z}_1	\mathbf{z}_2		\mathbf{z}_0	\mathbf{z}_1	\mathbf{z}_2	\mathbf{z}_3
position [m]	${}^F p_x$	0	□	$p_{x,s}$	${}^F p_y$	0	■	■	$p_{y,s}$
velocity [m/s]	${}^F v_x$	–	□	0	${}^F v_y$	–	□	□	0
acceleration [m/s ²]	${}^F a_x$	–	□	0	${}^F a_y$	–	□	□	0
jerk [m/s ³]	${}^F \dot{a}_x$	–	□	0	${}^F \dot{a}_y$	–	□	□	0
time [s]	t_x	0	■	T_p	t_y	0	■	■	T_p

– : Current value □ : Low level optimization ■ : High level optimization

basic difference between both configurations is that for stop configuration *Co-S* the end knot is configured to be fixed with stop position ${}^F \mathbf{p}_s = [p_{x,s} \ p_{y,s}]^T$. Due to the fact that a continuous decrease in velocity is desired, an intermediate intervention is not necessary as well. Hence, the longitudinal position of breakpoint \mathbf{z}_1 is not optimized on high level. For configuration *Co-D* the end knot of the longitudinal spline \mathbf{z}_2 is

configured to induce a constant acceleration continuation in order to account for the limited planning horizon, whereas at the end knot \underline{z}_3 of the lateral spline constant velocity is assumed for the further course.

Due to their characteristic both trajectory optimization approaches own unique, distinctive features that need to be addressed. For the continuous trajectory optimization approach the setup mainly reduces to the maximum number of SQP solver iterations. In chapter A.10 the convergence behavior of the continuous approach is shown for a situation in which the initial and final desired trajectory differ to a large extent. It is hence expected that a higher number of SQP iterations is needed to converge to the optimal solution. As can be seen, after $\hat{i} = 10$ iterations the result is already very close to the optimal one, leading to the conclusion that for an adequate initialization (which is explicitly addressed by the introduced environment-aware maneuver planning that provides a suitable, situation dependent maneuver trajectory) fewer iterations are needed to achieve convergence. The chosen setup can hence be regarded as a rather conservative setup, intended to focus more on the aspect of trajectory optimality than on the overall run time. The result of the discrete trajectory optimization approach is dependent on the quantization of high level decision variables. This is achieved via the number of samples in addition to the provided range extracted from the target region of the environment-aware maneuver planning. The corresponding setup for the high level optimization approaches is given in Table 8.4. From the given

Table 8.4.: Setup of trajectory optimization approaches.

description	variable	value
Continuous Trajectory Optimization		
Maximum number of optimization iterations	\hat{i}	10
Discrete Trajectory Optimization		
Number of samples for high level decision variable p_x	n_{px}	5
Number of samples for high level decision variable t_x	n_{tx}	3
Number of samples for high level decision variable p_y	n_{py}	3
Number of samples for high level decision variable t_y	n_{ty}	3

configurations it is now possible to deduce the number of nodes, edges and paths of the Longitudinal Kinematic Sampling Tree and the Lateral Kinematic Sampling Graph. The number of nodes on each level is given by vector \mathbf{h} and can be derived from the respective number of samples (see Table 8.4) in combination with the information if a quantity is configured as a high level decision variable (see Table 8.2 and Table 8.3, respectively). The vector of the cumulative product \mathbf{e} is given by $e_l = \prod_{j=1}^l h_j$ with $1 \leq l \leq \mathcal{K} + 1$. Table 8.5 summarizes the characteristics of the sampling structures for discrete trajectory optimization in dependence on the chosen configuration of the hierarchical framework.

In order to obtain the desired vehicle behavior the high level objective function weights are chosen as follows: $\omega_d = 5000$, $\omega_v = 10$, $\omega_p = 500$, $\omega_c = 5000$. This reflects the distinction between preferable behavior (low weights) and penalization of intolerable

Table 8.5.: Sampling structures for driving configuration *Co-D* and stop configuration *Co-S*.

Longitudinal Kinematic Sampling Tree					Lateral Kinematic Sampling Graph				
\hbar	\mathbf{e}	nodes	edges	paths	\hbar	nodes	edges	paths	
		$\sum_{i=1}^{\mathcal{K}+1} e_i$	$\sum_{i=2}^{\mathcal{K}+1} e_i$	$e_{\mathcal{K}+1}$		$\sum_{i=1}^{\mathcal{K}+1} \hbar_i$	$\sum_{i=1}^{\mathcal{K}} \hbar_i \hbar_{i+1}$	$\prod_{j=1}^{\mathcal{K}+1} \hbar_j$	
<i>Co-D</i>	[1 15 5]	[1 15 75]	91	90	75	[1 9 9 3]	22	117	243
<i>Co-S</i>	[1 3 1]	[1 3 3]	7	6	3	[1 9 9 1]	20	99	81

behavior (high weights) and is also designed to represent a hierarchical order in respect to priority (the higher the weight the more priority is given to the respective objective). To complete this section the vehicle model is set up in accordance to data of an *Audi A3 Sportback 2009* and solver settings for constrained optimization are set to default. For a holistic overview with respect to the chosen parameters of the vehicle model, the high level optimization parameters and solver settings it is referred to chapter A.12.

8.2. Analysis and Evaluation

For evaluation in dynamic environment a profound simulation and/or vehicle equipment is necessary to appropriately record relevant data. Especially the testing in real world environments is challenging, since it represents the highest level of difficulty with respect to the performance of the overall system. The strength of simulation-based testing can evidently be seen in the repeatability and reproducibility of test cases.

In general all fused and tracked objects are given by an object list. The road topology is majorly represented by lane markers given as third order polynomials. Moreover, information about the lane marker type is provided, which allows to account for additional traffic relevant aspects. With regard to the road model some preprocessing is done to generate the required information (see section 5.2). The preprocessing of the road model corresponds to a transfer from polynomial representation to polygonal chains, enhanced by a circular extrapolation to increase the overall view range.

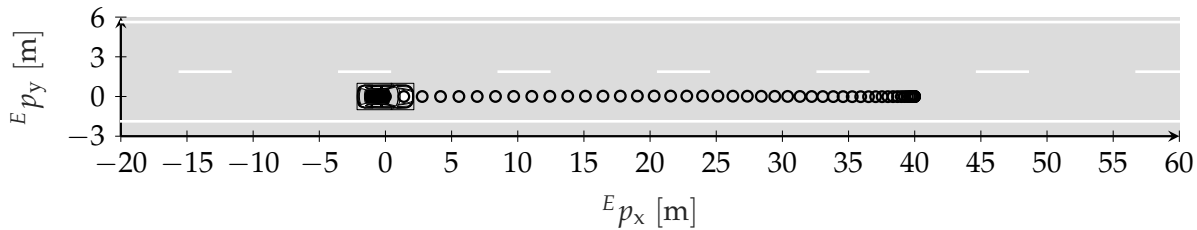
Situation-based testing

To focus on the capabilities of the developed trajectory planning approach situation based testing for one single time step is performed, assuming for example no uncertainty and exact representation of the static and the predicted dynamic environment. Results can hence be analyzed independent of e.g. perception and controller performance. To test the basic maneuver capabilities of the developed approaches a stop, a passing and a merge test case are chosen for detailed analysis.

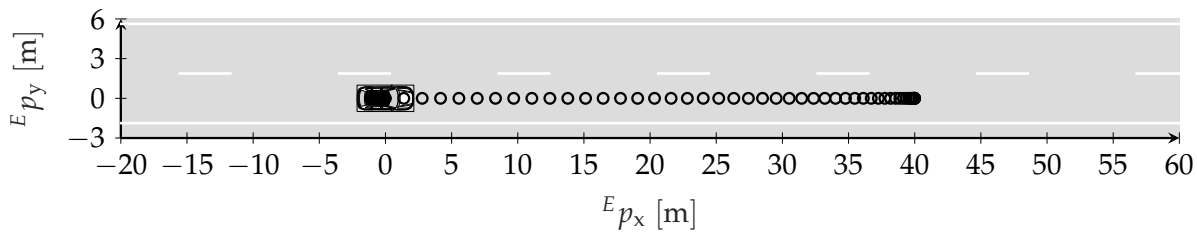
The stop test case *Si-S-1* is defined in Table 8.6 demanding a target break at stop position ${}^F \mathbf{p}_s = [40 \ 0]^T$ m. The target lane is set to two which means that the ego vehicle should follow the center of the current ego lane. Figure 8.1 shows the results for test case *Si-S-1* obtained with configuration *Co-S*.

Table 8.6.: Straight road test case definition *Si-S-1* for a stop maneuver.

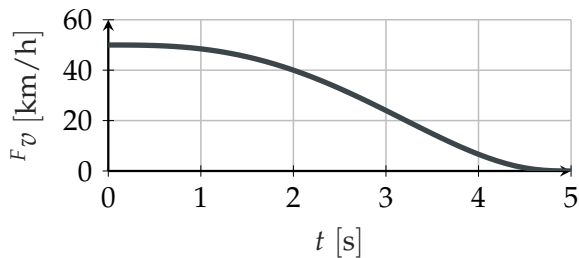
ID []	$F p_x$ [m]	$F p_y$ [m]	$F v$ [km/h]	$F \psi$ [rad]	target lane []	$F p_{x,s}$ [m]	$F p_{y,s}$ [m]
ego	0	0	50	0	2	40	0



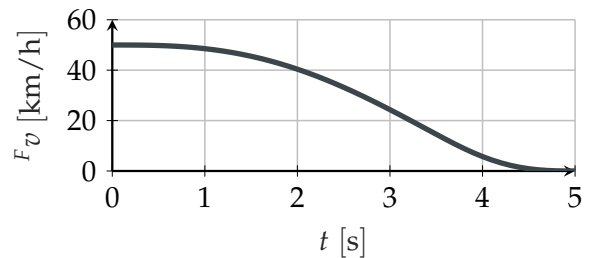
(a) Result of the continuous approach.



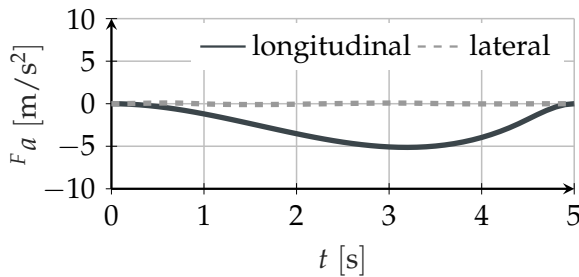
(b) Result of the discrete approach.



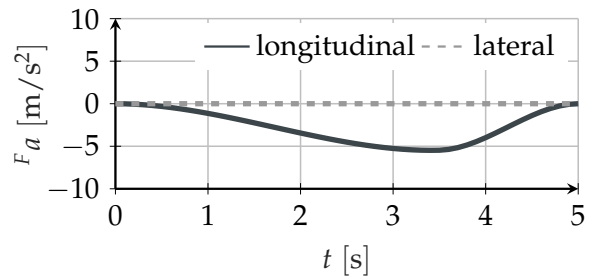
(c) Velocity of the continuous approach.



(d) Velocity of the discrete approach.



(e) Acceleration of the continuous approach.



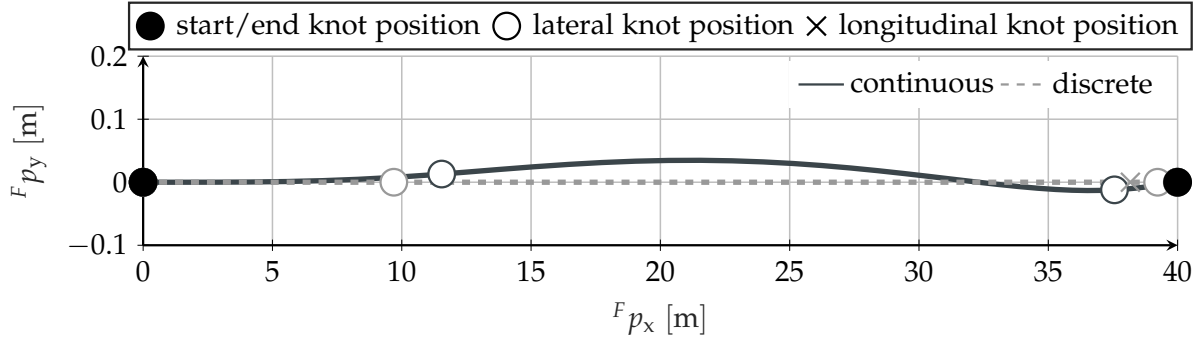
(f) Acceleration of the discrete approach.

Figure 8.1.: Results in stop test case *Si-S-1*.

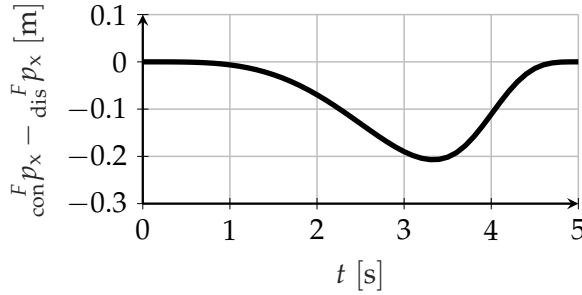
The results show only small differences between the solution of the continuous and the discrete approach. Both continuously decrease the initial ego velocity of 50 km/h to come to a stop at the desired stop position. The maximum longitudinal deceleration is therefore at around -5 m/s^2 . As the ego vehicle should follow the current lane center to stop at the desired stop position the lateral acceleration is zero.

The comparison as depicted in Figure 8.2 reveals that the continuous approach deviates by at maximum 4 cm from the ideal straight solution provided by the discrete approach.

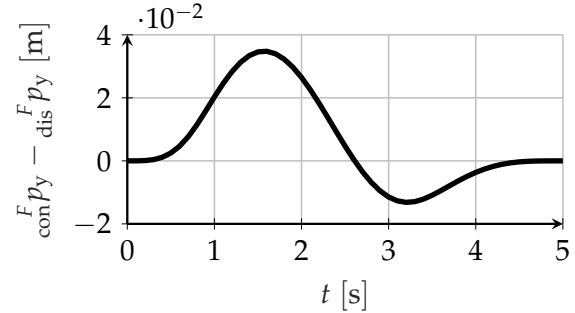
This is because the continuous approach tries to further reduce the costs caused by the required longitudinal braking acceleration exceeding the chosen comfort threshold of 3.5 m/s^2 as given in equation 7.2.11. In comparison to the discrete approach, the continuous approach is thus still better in terms of the defined objective function costs as it is shown in Table 8.9. Obviously, since start and end knot are fixed, the deviation in lateral and longitudinal position for $t = 0 \text{ s}$ and $t = 5 \text{ s}$ is zero. Because configuration



(a) Comparison of the lateral and longitudinal positions.



(b) Deviation in longitudinal position.



(c) Deviation in lateral position.

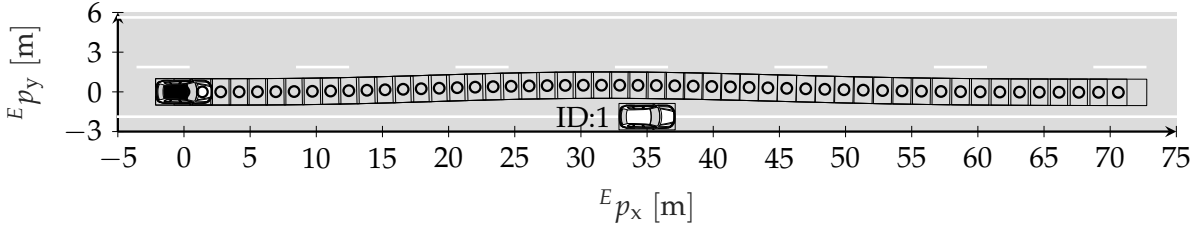
Figure 8.2.: Comparison of the continuous and the discrete approach in stop test case *Si-S-1*.

Co-S allows for low level optimization of the longitudinal position of the intermediate knot, the longitudinal behavior is basically determined by one single spline segment. In both solutions the longitudinal position of the intermediate knot \mathbf{z}_1 is hence located at the end of the trajectory. For the continuous approach the longitudinal knot position is 39.90 m at $t = 4.40 \text{ s}$ and 38.17 m at $t = 3.50 \text{ s}$ for the discrete approach.

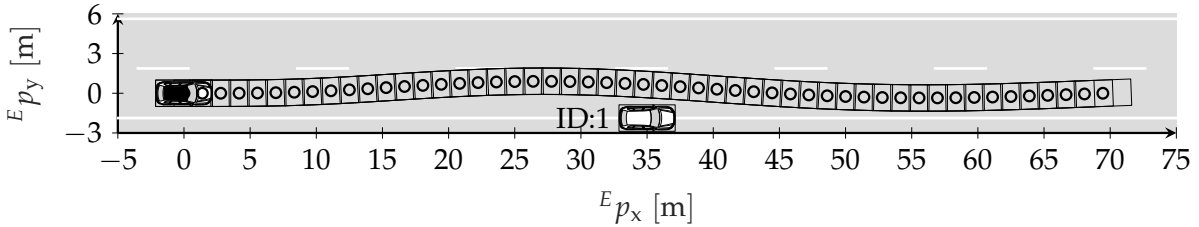
To illustrate the comprehensive capabilities of the developed approach, a passing test case is analyzed that requires an adapted, intuitive behavior of the ego vehicle. This is because the ego vehicle needs to deviate from the ideal center line of the ego lane to prevent from colliding with the parking vehicle. Although this is contrary to the intentionally defined objectives, it emphasizes the compliance with respect to the requirement of collision avoidance incorporated as an inequality constraint. Table 8.7 shows the corresponding test case definition *Si-P-1*. As depicted in Figure 8.3 the continuous as well as the discrete approach master the passing maneuver and avoid a collision with parking obstacle vehicle ID:1, which is illustrated by showing the ego vehicle bounding box for all trajectory points $k = 0, 1, \dots, K$. In comparison to the discrete approach the continuous approach deviates less from the desired lane center with the maximum deviation in close proximity to the parking obstacle ID:1. Thus, the discrete approach causes higher costs in terms of the objective function (see Table 8.9).

Table 8.7.: Straight road test case definition *Si-P-1* for a passing maneuver.

ID []	$F p_x$ [m]	$F p_y$ [m]	$F v$ [km/h]	$F \psi$ [rad]	target lane []	$F v_{des}$ [km/h]
ego	0	0	50	0	2	50
1	35	-1.875	0	0	3	0



(a) Result of the continuous approach.



(b) Result of the discrete approach.

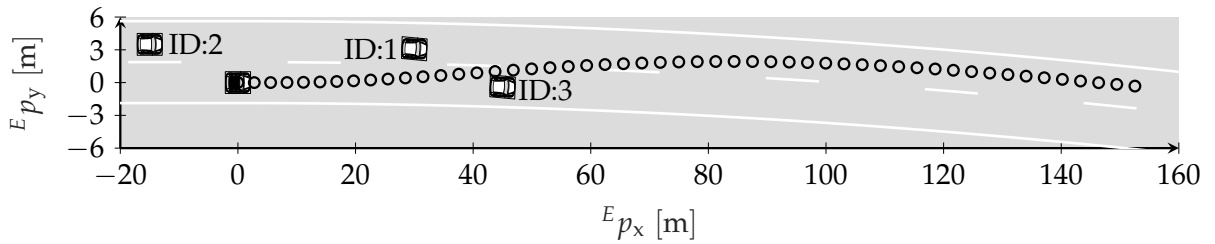
Figure 8.3.: Results in passing test case *Si-P-1*.

The test case definition for merge maneuver *Si-M-1* in a right curve is given in Table 8.8. It is designed to show the capabilities of the developed approaches with respect to realizing a lane change into a gap between two obstacle vehicles (ID:1 and ID:2) on the neighboring lane in order to overtake obstacle vehicle ID:3. This setup is reflected in the respective target lane, which is defined with respect to the current ego lane. Here, the ego vehicle target lane of one indicates a desired lane change to the left. Figure 8.4 shows the corresponding results for the continuous approach and the discrete approach using configuration *Co-D*. The general characteristic of the planning result is similar for the continuous as well as the discrete approach. The continuous approach realizes the merge-in maneuver with a lower maximum lateral acceleration, meaning that the discrete approach reaches the target lane faster on the other hand. For longitudinal planning no significant difference between the continuous and the discrete approach can be observed. The mentioned differences are also highlighted in Figure 8.5.

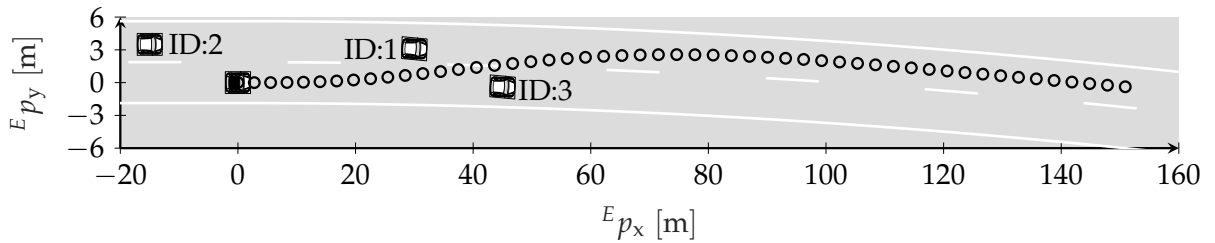
With results for test cases *Si-S-2* and *Si-M-2* shown in chapter A.11 altered variations of test cases *Si-S-1* and *Si-M-1* are analyzed. Comparing the absolute objective function costs as depicted in Table 8.9 it can clearly be seen that the continuous approach consistently performs better in that respect. It emphasizes the suboptimal character of the discrete approach due to the quantization of high level decision variables, indicating an insufficient resolution.

Table 8.8.: Right curve test case definition *Si-M-1* for a merge maneuver.

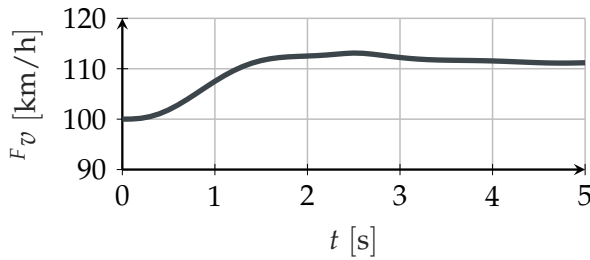
ID []	$F p_x$ [m]	$F p_y$ [m]	$F v$ [km/h]	$F \psi$ [rad]	target lane []	$F v_{des}$ [km/h]
ego	0	0	100	0	1	130
1	30	3.10	110	-0.05	1	110
2	-15	3.50	110	0	1	110
3	45	-0.40	85	-0.05	2	85



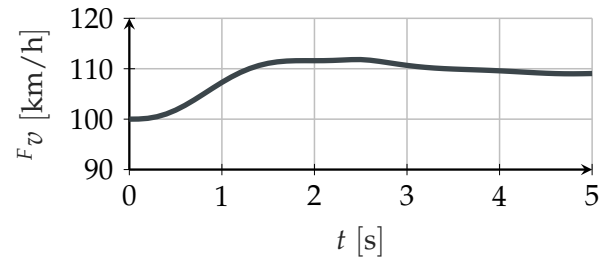
(a) Result of the continuous approach.



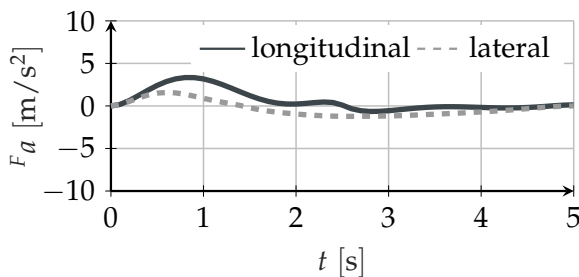
(b) Result of the discrete approach.



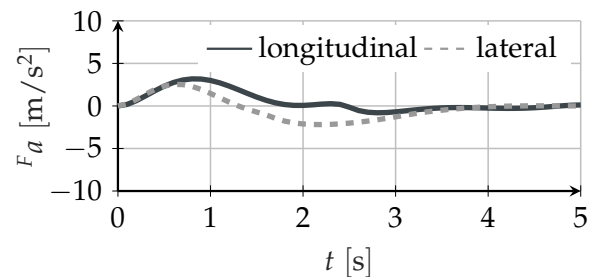
(c) Velocity of the continuous approach.



(d) Velocity of the discrete approach.

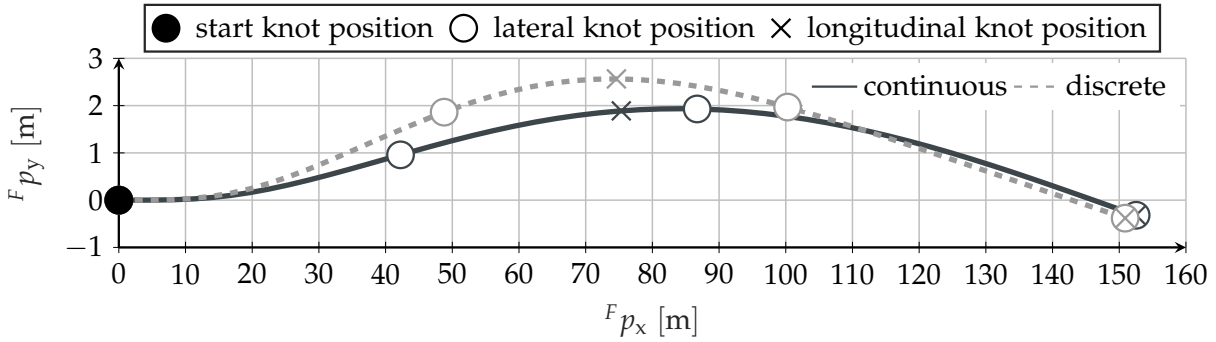


(e) Acceleration of the continuous approach.

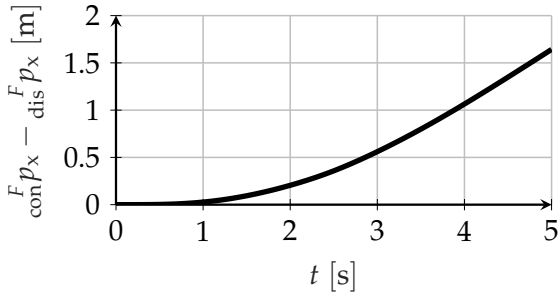


(f) Acceleration of the discrete approach.

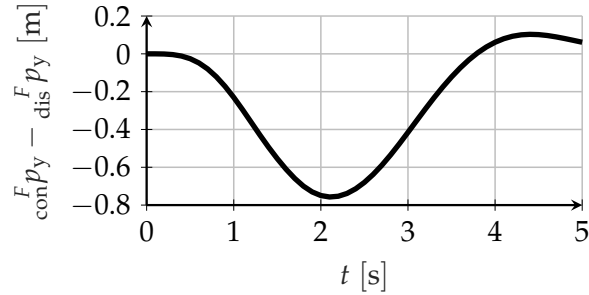
Figure 8.4.: Results in merge test case *Si-M-1*.



(a) Comparison of the lateral and longitudinal positions.



(b) Deviation in longitudinal position.



(c) Deviation in lateral position.

Figure 8.5.: Comparison of the continuous and the discrete approach in merge test case *Si-M-1*.

Table 8.9.: Comparison of final objective function costs.

	<i>Si-S-1</i>	<i>Si-S-2</i>	<i>Si-M-1</i>	<i>Si-M-2</i>	<i>Si-P-1</i>
continuous approach	58 088.02	59 562.76	28 508.14	19 404.07	2067.44
discrete approach	63 205.97	72 415.27	32 759.96	19 681.79	5491.10

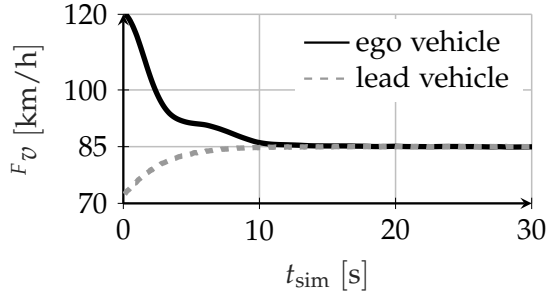
Scenario-based testing

A dynamic environment simulation (Wissing et al. 2016) is used to evaluate the developed algorithms on scenario basis (in closed-loop). To prevent an overestimation of the trajectory planning capabilities driven by results gained from a simulation environment it is vital to model certain effects of the perception module. Therefore, the simulated ego vehicle is equipped with a front camera and radar sensors to the front, rear and sides. The environment model contains a holistic description of the static world and dynamic objects. However, the final model is not considered to be ideal, since the detection of the prevalent lane structure mainly relies on the camera data and is hence prone to inaccurate measurements and low view ranges. Analogous effects are considered for dynamic object detection providing realistic results with respect to the modeling of the dynamic environment. At first, to evaluate the performance for lateral and longitudinal vehicle motion separately, desired high level behavior in terms of distance keeping and lane centering is tested.

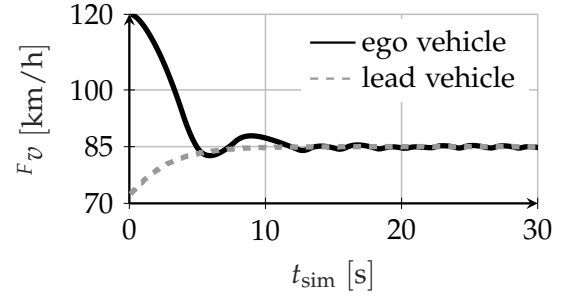
To test the distance keeping performance defined via a constant time gap, test case *Sc-A-1* as given in Table 8.10 is defined. A slower lead vehicle ID:1 is placed 70 m

Table 8.10.: Straight road test case definition *Sc-A-1*.

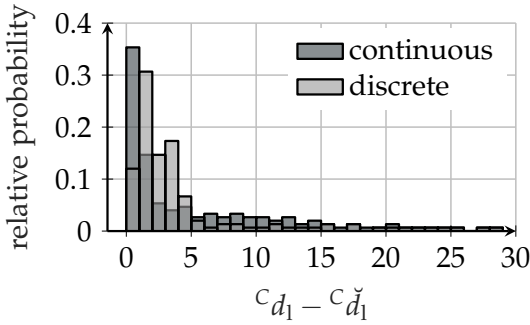
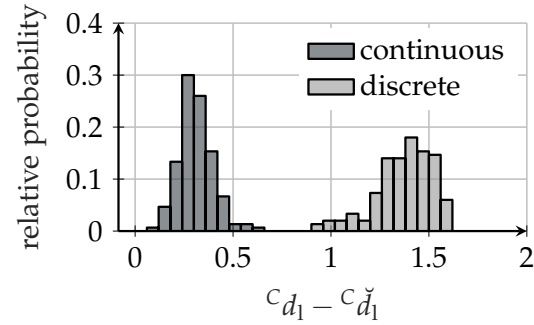
ID []	$F p_x$ [m]	$F p_y$ [m]	$F v$ [km/h]	$F \psi$ [rad]	target lane []	$F v_{des}$ [km/h]
ego	0	0	120	0	2	120
1	70	0	72	0	2	85



(a) Velocity of the continuous approach.



(b) Velocity of the discrete approach.

(c) Distribution for time interval $t_{sim} = [0 \ 15]$ s.(d) Distribution for time interval $t_{sim} =]15 \ 30]$ s.Figure 8.6.: Results of the continuous approach and the discrete approach in test case *Sc-A-1*.

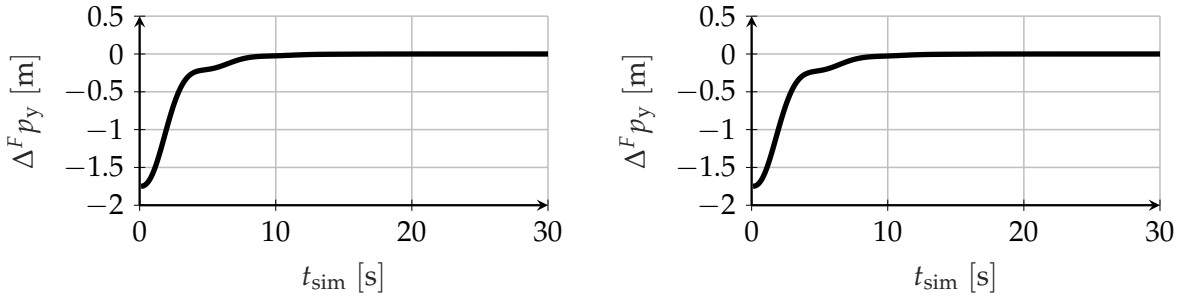
ahead of the ego vehicle. The lead vehicle accelerates to reach 85 km/h, whereas the ego vehicle drives at a desired speed of 120 km/h. As target lanes indicate the ego vehicle as well as the obstacle vehicle are supposed to follow the current ego lane.

Figure 8.6 shows the results of test case *Sc-A-1* for the continuous as well as the discrete approach. Although the desired speed of the ego vehicle is $F v_{des} = 120$ km/h, the ego vehicle decelerates to adapt to the speed of the lead vehicle (see Figure 8.6a and Figure 8.6b). In this test a difference between the continuous and the discrete approach becomes obvious, as the velocity of the discrete approach contains undesired oscillations. The prevalent reason for this can be seen in the suboptimality of the discrete approach, which leads to a volatile convergence towards the ideal lead vehicle distance. Because of the adaptive discretization strategy that features a stronger dependence on the realized trajectory of the previous planning cycle, the reciprocal action between the planning and control part also contributes to the observed effect. To account for the different phases during the ego maneuver the scenario is divided in two parts. Hence, Figure 8.6c and Figure 8.6d show the relative probability of the difference between the actual lead vehicle distance C_{d_1} and the minimum desired reference distance $C_{d_1}^{\check{}}$ for

two different time intervals. The first part within time interval $t_{\text{sim}} = [0 \ 15] \text{ s}$ represents the transition phase, in which the ego vehicle approaches the slower lead vehicle. The discrete approach decelerates less in the beginning, such that the distance to the lead vehicle decreases faster. The second part for time interval $t_{\text{sim}} =]15 \ 30] \text{ s}$ then shows the accuracy with respect to the distance keeping objective. In the latter case the discrete approach has a mean and standard deviation of $(1.375 \pm 0.146) \text{ m}$, whereas for the continuous approach a mean and standard deviation of $(0.310 \pm 0.086) \text{ m}$ are identified. As a matter of fact the result shows that the difference between the actual lead vehicle distance ${}^C d_1$ and the minimum desired reference distance ${}^C \check{d}_1$ is never below zero, which shows that the minimum desired safety distance is always satisfied. Although the velocity objective is conflicting with the distance keeping objective, the ego vehicle performs reasonable reflecting the priority in terms of the chosen weights. The lane centering performance is evaluated in test case *Sc-L-1* (see Table 8.11) with an initial deviation to the lane center of $\Delta^F p_y = -1.75 \text{ m}$. The ego vehicle target lane is two, which means that the ego vehicle should center in its current lane.

 Table 8.11.: Straight road test case definition *Sc-L-1*.

ID []	$\Delta^F p_y$ [m]	${}^F v$ [km/h]	${}^F \psi$ [rad]	target lane []	${}^F v_{\text{des}}$ [km/h]
ego	-1.75	100	0	2	100



(a) Lane center deviation for continuous approach. (b) Deviation to lane center for discrete approach.

 Figure 8.7.: Result of continuous approach and discrete approach in test case *Sc-L-1*.

The results shown in Figure 8.7 reveal only slight differences between the continuous and the discrete approach. Interestingly, the depicted results share the same characteristic with respect to the transition phase, in which convergence of the lateral position slows down after reaching $\Delta^F p_y = -0.2 \text{ m}$. One explanation for this behavior is that based on equation 7.2.9 the objective for lateral positioning is dependent on the relative position of the ego vehicle to the desired lateral position. Especially in the beginning high deviations to the desired lateral position contribute to higher lateral velocities to decrease these costs. Taking into account the results of *Sc-A-1* as well (cmp. 8.6a), it seems that the observed characteristic is hence caused by a power shift between effective cost terms of the high level objective function.

Finally, a typical highway scenario is investigated in simulation. Test case *Sc-H-1*, representing a part of the German highway A2 from Gelsenkirchen-Buer to Herten is considered. The course of the road highway is modeled starting with ${}^E p_x = 0$ m connected by an arbitrary highway entry. The desired ego vehicle velocity is 100 km/h for the highway entry part, valid until ${}^E p_x = 100$ m. Then, for highway driving the desired velocity is 120 km/h, changing to 140 km/h at the moment the ego vehicle passes ${}^E p_x = 300$ m. To follow the driven path of the ego vehicle the ego vehicle trace is visualized as a black dotted line. Figure 8.8 shows the result for the continuous approach. As can be seen, the velocity adapts to the dynamic traffic, but by overtaking slower vehicles in front of the ego vehicle the desired target speed is finally reached. Figure 8.8 also shows some characteristic snapshots of the ego vehicle drive for selected time instances. At $t_{\text{sim}} = 5.7$ s the ego vehicle performs a lane change to merge on the highway leaving the ending entrance lane. To reach the modified set speed of 120 km/h, at $t_{\text{sim}} = 14.3$ s the ego vehicle does a left lane change to overtake obstacle vehicle ID:10. Since obstacle vehicle ID:9 is driving slower than the second time updated desired set speed of 140 km/h, after some time at around $t_{\text{sim}} = 23.1$ s the ego vehicle safely merges in the gap between obstacle vehicle ID:8 and obstacle vehicle ID:11. The current lead vehicle ID:11 at that time also prevents the ego vehicle to realize the desired target speed. When obstacle vehicle ID:11 changes its lane to the right the ego vehicle accelerates and at the moment the ego vehicle passes obstacle vehicle ID:11 a lane change to the right is performed ($t_{\text{sim}} = 50.1$ s). At $t_{\text{sim}} = 66.8$ s the ego vehicle continues to overtake obstacle vehicle ID:5 and performs a single lane change to use the free lane to the right of the current ego lane. For a comprehensive understanding of the development and a more detailed description of the particular characteristics that can be observed, corresponding sequences to each of the five mentioned situations are shown in chapter A.11 (see Figure A.14, Figure A.15, Figure A.16, Figure A.17 and Figure A.18).

As depicted in Figure 8.9 the discrete approach shows a quite similar result with respect to highway test case *Sc-H-1*. Since the trajectory planning solution is not exactly the same as for the continuous approach, also simulated surrounding obstacle vehicles react accordingly. However, the basic evolution of the scenario is the same. The velocity plot reveals that acceleration of the discrete approach is lower, but the desired target speeds of 100 km/h, 120 km/h and 140 km/h are generally reached. Figure 8.9 also shows the result of the discrete approach for selected time instances. At $t_{\text{sim}} = 5.4$ s the ego vehicle plans to merge on the highway lane in front of obstacle vehicle ID:7. To accelerate up to the desired 120 km/h, at $t_{\text{sim}} = 14.0$ s the ego vehicle does a lane change to the left, intending to overtake obstacle vehicle ID:10. At $t_{\text{sim}} = 23.9$ s the ego vehicle accelerates to merge behind obstacle vehicle ID:11 and in front of approaching vehicle ID:8 to overtake obstacle vehicle ID:9, trying to reach the increased set speed of 140 km/h. With modified set speed of 140 km/h at $t_{\text{sim}} = 50.5$ s the ego vehicle then overtakes obstacle vehicle ID:11. Finally, at $t_{\text{sim}} = 66.8$ s to finish the overtaking maneuver of obstacle ID:5 the ego vehicle performs a single lane change to the right. Analogous to the results shown for the continuous approach, details and sequences illustrating the temporal context are given in chapter A.11 (see Figure A.19, Figure A.20, Figure A.21, Figure A.22 and Figure A.23).

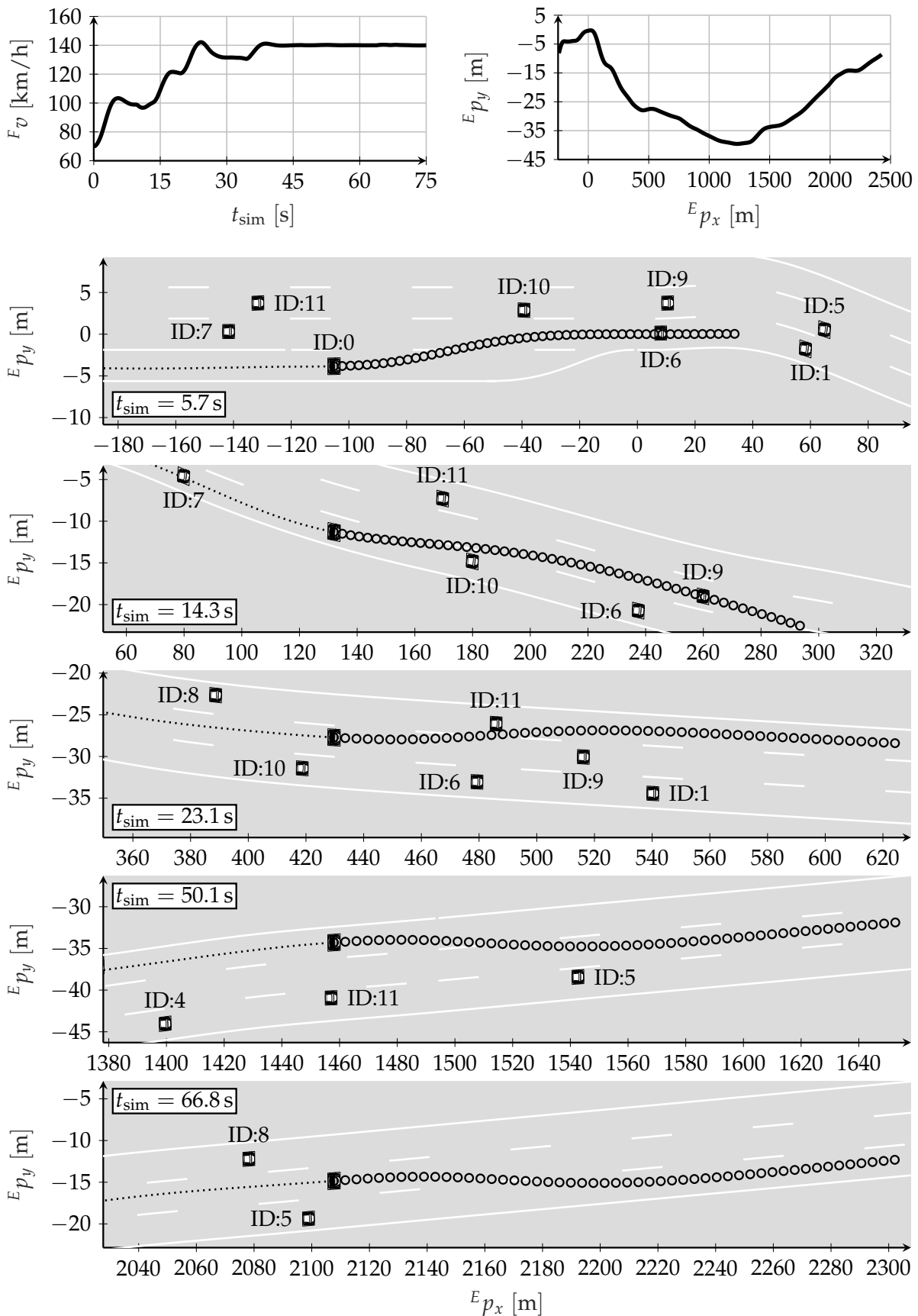


Figure 8.8.: Result of continuous approach in test case Sc-H-1.

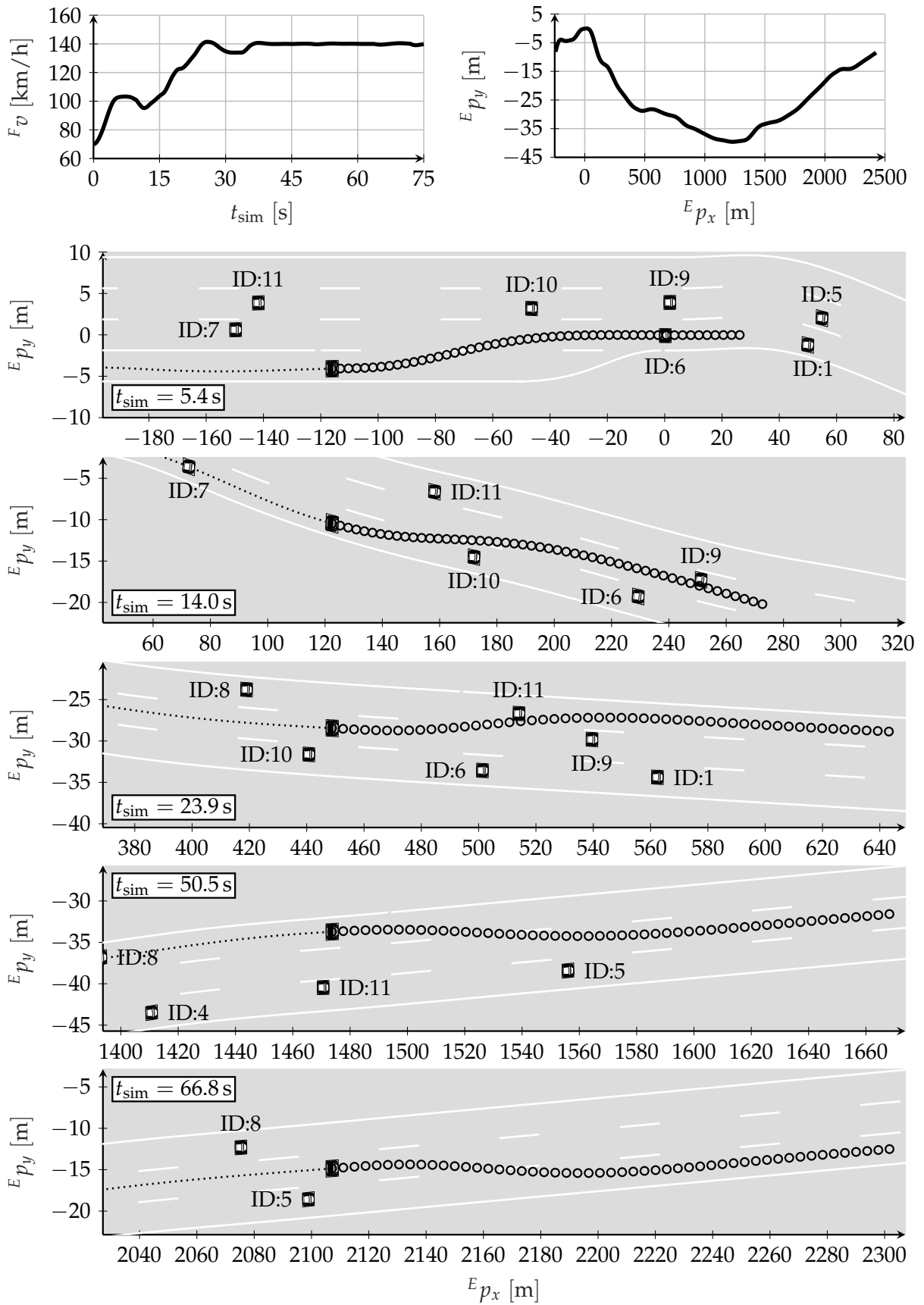
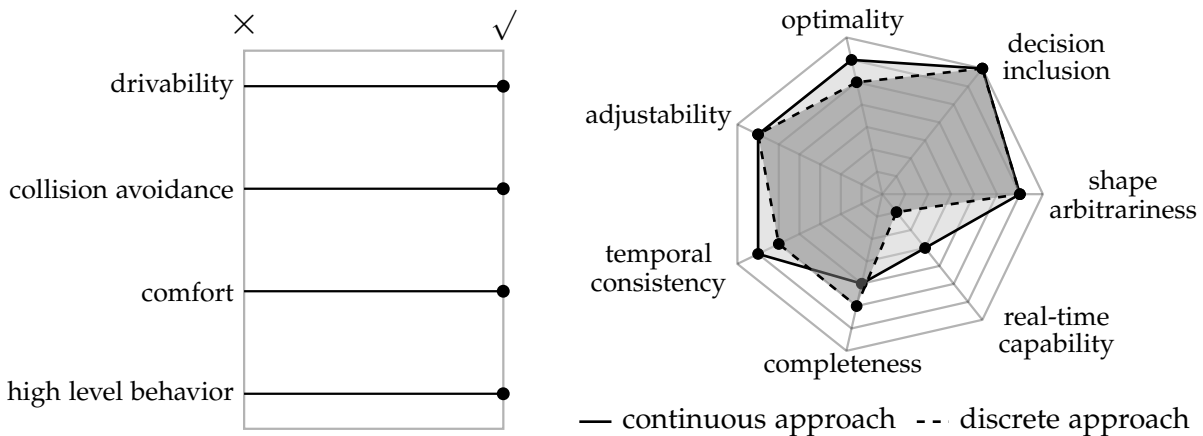


Figure 8.9.: Result of discrete approach in test case Sc-H-1.

8.3. Discussion of the Results

The fundamental goal that is targeted during development is the application in a vehicle in real world traffic. Test drives related to previous work (see section 7.4.1) attested the general functioning of the continuous trajectory optimization approach with distinct spline interpolation for highway driving. However, experience has shown that especially the correct balance for general purpose is hard to find. Tuning of lane changes affected the lane keeping performance possibly accompanied by other unwanted effects like e.g. curve cutting, complicating the application in real world traffic. Therefore, in this thesis special attention has inter alia been directed to the requirement of adjustability that is particularly reflected in the design of the developed planning approach.

In order to assess the results the achievements with respect to the requirements defined in section 3.2 are collated. Figure 8.10 summarizes the achievements with respect to the requirements on the trajectory as well as on the trajectory approach. As shown in



(a) Requirements on the trajectory.

(b) Requirements on the trajectory approach.

Figure 8.10.: Evaluation with respect to the requirements.

Figure 8.10a the requirements on the trajectory are universally met, irrespective of the chosen high level optimization approach. The drivability requirement is considered by the integration of the vehicle dynamics model introduced in section 5.1. Collision avoidance is covered by the integration of a comprehensive environment model that accounts for static as well as dynamic collision checking as described in section 5.2. Since comfort is uniquely related to the low level problem of the proposed two level hierarchical framework, the minimum kinematic trajectory generation in section 6.3 ensures the fulfillment of this requirement. The high level behavior as the counterpart within the proposed framework is considered via section 7.2.

The fulfillment of requirements with respect to the trajectory planning approach is rated by means of a spider web chart depicted in Figure 8.10b. Since in this context the continuous trajectory optimization and discrete trajectory optimization approach own different characteristics the estimated score is shown for both approaches. High values correspond to a better realization of the respective characteristic. This means

that the more area is covered by an approach the more suitable it is for the purpose of trajectory planning in dynamic environments. As can also be inferred from the chart, the aspects of adjustability, shape arbitrariness and decision inclusion are determined by the basic design of the framework, whereas optimality, real time capability, temporal consistency and completeness are ultimately affected by the chosen approach to the high level optimization problem.

The requirement of adjustability relates to the estimated application effort that is necessary to tune the developed approach with respect to the desired behavior. This in particular includes high and low level objective function and constraints parameters (cmp. Table A.7). To put adjustability in concrete terms it is defined in a way that a higher adjustability is reached with less parameters, moreover demanding a certain expressiveness of the parameters to minimize interrelated dependencies. This means that for good compliance it is important to reduce the number of conflicting objectives as well as the number of parameters itself. In comparison to previous work a noteworthy improvement has been achieved in that respect. Especially the elimination of conflicting objectives due to the hierarchical structure of the developed framework characterizes a beneficial effect. However, the developed approach still relies on a few parameters that need to be tuned. For this reason the characteristic is not rated with the highest score. Since adjustability is in some way contrary to the aspect of customization (which is not particularly listed as a requirement) it is worth to mention that the latter is beneficially addressed via the existing parameters. It can further be concluded that a good compromise between adjustability and customization has been found.

Within the proposed two level hierarchical framework the requirement of shape arbitrariness is mainly connected to the chosen spline representation. Since prevailing non-holonomic constraints inherently limit the representable trajectory shapes the focus of this requirement is directed to the aggregate of all drivable trajectories. This means that chosen representation should be valid for all possible traffic scenarios. The flexibility and adaptiveness of a spline are evident and because moreover the number of breakpoints and spline continuity conditions can be configured there are no essential restrictions apparent. Still, to take the interpolating behavior into account the assigned score is not matched to the best value.

The requirement of decision inclusion is fulfilled to its full extent. Not only that superordinated decisions are covered in terms of a reference, decisions can also be reflected in the setup of decision variables within the hierarchical framework that allows to distinguish between fixed, optimized and adapted quantities.

By convention an algorithm is denoted as complete if it is guaranteed that a solution is found if one exists and failure is returned otherwise. With respect to the developed continuous and discrete approach a proof is not given as the number of possible scenarios is too high to prove completeness by trial or any other approach. On the contrary continuous approaches can end up in local minima, which may cause the algorithm to terminate without finding a trajectory when one exists. Discrete algorithms on the other hand sacrifice completeness since a solution may be hidden in the interstice. But there are also some points to mention that have a positive effect with respect to the requirement of completeness. First of all, the amount of scenarios in

which a valid solution can be found is expected to be much higher than the ones where no solution exists. In addition, for most of the cases it is even presumed that several solutions exist that only differ in their costs with respect to the defined objectives. This aspect rigorously simplifies the search for a valid solution in general. The problem of local minima for the continuous approach is mitigated by the inclusion of the result of the higher level maneuver decision. For the discrete approach it is assumed that the solution is rather suboptimal than that the applied sampling strategy with its deterministic pattern prevents to find a valid solution in general. As completeness is not referring to the degree of optimality of the solution, in comparison to the continuous approach the discrete approach is ranked slightly better.

Results show that with respect to the requirement of optimality the discrete approach lacks performance, since suboptimality is inherently expected due to the chosen resolution in the sampling process. Although the continuous approach is prone to end up in a local minimum, because of the continuous character of the decision variables a larger scope for action to realize optimality demands can be inferred. Finally, in comparison to the discrete approach this leads to a higher score. Unfortunately, there is no universally valid guarantee of convergence for real-time application. However, in chapter A.10 convergence behavior of the continuous trajectory optimization approach is shown for a dedicated scenario. It can hence be concluded that a fundamental characteristic is represented by the trade off between optimality and run time, as for the continuous approach optimization iterations are limited to a certain number and for the discrete planning approach the number of sampled trajectories is ultimately restricted as well.

Both the continuous and the discrete approach do yet not meet the requirement of real-time capability to its full extent, which is also reflected in the respective score. In comparison to the discrete approach the run time of the continuous approach of around 300 ms to 400 ms is considerably closer to the target value of less than 100 ms. In fact in chapter A.10 it is shown that for the presented approach mean run times of 75 ms are achievable. Still, improvements with respect to the run time per iteration are vital. In this context a native C/C++ implementation will certainly enhance run time performance in a significant manner. To estimate the run time capabilities of the discrete approach a related approach introduced by Schlechtriemen et al. (2016) can be adduced. Therein run time is stated to be approximately 50 ms for 3000 sequentially processed trajectories. As for the discrete approach it is moreover possible to parallelize the computation task, switching to graphics processing units as target hardware will support significant speed-up in computation time. Hence, with the contrivable improvement in run time the overall potential of the developed continuous as well as of the discrete approach is estimated to be quite high.

The requirement of temporal consistency is strongly related to the aspect of optimality, since suboptimality might induce undesired effects between consecutive trajectory planning cycles. From that perspective temporal consistency is rated analogous to the optimality score particularly accounting for this connection.

9

Conclusion and Outlook

This thesis is dedicated to the development of a trajectory planning approach to automated driving. Hereinafter the key features and insights gained through thorough investigations will be summarized. Moreover, some thoughts are shared on how to tackle remaining challenges with respect to trajectory planning for automated vehicles.

9.1. Summary of the Work

The development of automated driving functions in order to increase traffic safety and comfort *inter alia* relies on high performance approaches to trajectory planning. Because of the broad variety of approaches to trajectory planning, which is mainly driven by the requirements originating from diverse applications, a taxonomy is presented that groups existing approaches to provide a conclusive overview. However, in general it is vital to define the role of trajectory planning within the functional architecture of the automated vehicle. To narrow the scope of the trajectory planning approach, in a first step requirements and interfaces need to be defined. Especially the interface to the superordinated modules is important, because it has a considerable impact on the performance of the trajectory planner. For this reason a generic interface to the maneuver decision is defined that is able to act on maneuver decision as well as on maneuver planning level. To harmonize the direct interface between trajectory planning and the high level decision making an approach to maneuver planning is sketched that enables trajectory planning in accordance to the scope of this work. Furthermore, the environment-aware maneuver planning is important to provide an appropriate initialization and starting point for the presented discrete and continuous trajectory planning approach. Obviously, a discrete approach can deal better with no initialization. However, for the sake of comparability this aspect is not focused in this thesis, such that both developed approaches have to solve the same planning problem. In this thesis an efficient and scaleable, general approach to trajectory planning is presented. The core of the developed approach is represented by the proposed two level hierarchical trajectory planning framework. The separation in high level and low level behavior accounts for the fact that the desired high level behavior changes in dependence on the current situation, whereas the low level behavior should basically adjust optimally without totally enforcing, but still pursuing its own objectives (i.e.

driving comfort). As a result a generic trajectory planning concept is introduced that explicitly considers the problem of conflicting planning objectives via its hierarchical character. The idea of the hierarchical trajectory planning framework can be mapped to the theory of bilevel programming. For the particular case that the lower level problem has a unique optimal solution, the bilevel programming problem is equivalent to a single level optimization problem. As a consequence common approaches to trajectory optimization can be applied in order to tackle the trajectory planning problem on high level. To solve the high level problem two distinct approaches are introduced. Referring to the characteristic of the optimal parameters a discrete and a continuous trajectory optimization approach are developed. Evidently, results show that the presented approaches share the advantages and disadvantages of optimization and sampling-based trajectory planning approaches, respectively. Whereas the continuous approach relies on numerical optimization, the developed discrete trajectory planning approach belongs to the class of trajectory rollout. One key feature of the presented trajectory planning framework is the spline-based trajectory representation. On low level an advanced approach to optimal interpolation for spline-based trajectory planning is used, denoted as minimum kinematics trajectory generation. The formulated low level problem can efficiently be solved in an analytical way or by using quadratic programming techniques.

In general a spline can solve the constrained trajectory planning problem in an optimal fashion, since it is able to adapt to the objectives of the high level cost function. Desired behavior can hence be reproduced if the corresponding reference is chosen appropriately. The underlying low level problem, which supports the idea of minimum kinematics trajectory generation for spline interpolation is inherently designed to be optimal. The spline representation offers planning versatility, while reducing the number of decision variables at the same time. As an analytic solution is provided continuous evaluation of each spline is enabled as well, facilitating the coupling of longitudinal and lateral dynamics. Moreover, interpolation permits to prolong the prediction horizon without increasing the number of decision variables and can hence be regarded as an efficient way of formulating the trajectory planning problem.

The spline representation and the choice of the vehicle model synthesize to a beneficial approach as the integration of system dynamics is accomplished by means of accurate and fast analytic calculations. Whereas longitudinal motion is captured on kinematic level, the modeling of lateral vehicle dynamics is thoughtfully chosen to find a compromise between an accurate solution for the intended use cases and a reasonable computational effort. A vehicle model with steady state yaw dynamics of a linear single track model is chosen that provides the relation between steering angle and yaw rate at steady state. Analogous to the choice of a suitable vehicle model, the applied approaches to static and dynamic collision checking need to balance the aspects of accuracy and computational effort. Since collision checking is directly connected to the aspect of safety, accuracy is prioritized to a large extent. For static collision checking this means that the ego vehicle shape is modeled by its bounding box and the road model is built from perceived lane markers represented by polygonal chains. Then a pseudo distance transform is used that allows for a fast shortest distance calculation. For dynamic collision checking vehicles are approximated by a predefined number

of circles, accounting for the uncertainty in object detection, since a certain margin is added to the vehicle shape. Dynamic collision checking reduces to simple distance calculations and threshold evaluations, which ensures a low computational burden. All taken measures and features of the developed trajectory planning framework engage with each other, since the spline-based representation allows for optimal minimum kinematics trajectory generation and enables the analytic calculation of kinematics, such that the property of flatness can be exploited to integrate vehicle dynamics. A seamlessly embedded solution is presented covering key aspects of trajectory planning for on-road driving in dynamic environments, such as accurate collision checking and appropriate vehicle dynamics modeling. This leads to a generic trajectory planning concept that addresses applicability for a broad range of scenarios, since it provides an optimal solution in terms of the formulated hierarchical trajectory planning problem.

9.2. Outlook

The search for a suitable trajectory planning approach is restricted by real-time requirements on the processing time. In particular, a high replanning frequency leads to a reactive layer that is capable of dealing with changes in the dynamic environment. As results and previous discussion have shown the latter aspect leaves some room for improvement. The spline-based design enables a versatile usage in terms of the number of breakpoints and could dynamically be adjusted to account for the current situation. In order to increase the accuracy for lane centering for example, the number of breakpoints for the lateral spline could be adjusted with respect to the road curvature. From the general considerations made, the proposed framework should be readily adaptable to enhance the scope to urban scenarios and parking applications. Still, some work will be necessary to establish an associated proof of concept. The dependency of trajectory planning on higher level modules like perception and decision making is evident. Clearly, the inclusion of map data for example would further enhance the quality of the overall results. In order to find the best trajectory among several maneuvers, within maneuver decision the space might be segmented into regions that will most likely contribute to the best solution. The procedure can be extended by evaluating different maneuver hypothesis in parallel, planning the optimal trajectory for each considered possibility and finally choosing the overall optimal solution. To broaden the scope of maneuver decision and maneuver planning as presented in this thesis a more advanced solution is necessary. However, the development of an approach related to these kinds of problems represents a separate research topic. Another interesting research direction is the development of cooperative motion planning approaches that own a certain kind of interaction awareness. Although hard to capture this is a very desirable feature, since it reflects the impact of the ego vehicle on the surrounding obstacle vehicles and vice versa. Moreover, it could be worth considering a probabilistic trajectory representation. This is left for further work. To finally launch highly automated driving functions it will require combined resources and substantial progress in all related high technology fields. Hence, industry and research need to show creativity and ingenuity to advance the state of the art.

Bibliography

- Althoff, M. (2010):** “Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars”. Dissertation. Munich: Technical University of Munich.
- Ardelt, M., C. Coester, and N. Kaempchen (2012):** “Highly Automated Driving on Freeways in Real Traffic Using a Probabilistic Framework”. In: *IEEE Transactions on Intelligent Transportation Systems* 13.4, pp. 1576–1585.
- Arslan, O., K. Berntorp, and P. Tsiotras (2017):** “Sampling-based algorithms for optimal motion planning using closed-loop prediction”. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4991–4996.
- Banzhaf, H., N. Berinpanathan, D. Nienhüser, and J. Zöllner (2018a):** “From G 2 to G 3 Continuity: Continuous Curvature Rate Steering Functions for Sampling-Based Nonholonomic Motion Planning”. In: *IEEE Intelligent Vehicles Symposium (IV)*, pp. 326–333.
- Banzhaf, H., M. Dolgov, J. Stellet, and J. M. Zöllner (2018b):** “From Footprints to Beliefprints: Motion Planning under Uncertainty for Maneuvering Automated Vehicles in Dense Scenarios”. In: *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1680–1687.
- Bauer, E., F. Lotz, M. Pfromm, M. Schreier, S. Cieler, A. Eckert, A. Hohm, S. Lüke, P. Rieth, B. Abendroth, V. Willert, J. Adamy, R. Bruder, U. Konigorski, and H. Winner (2012):** “PRORETA 3: An Integrated Approach to Collision Avoidance and Vehicle Automation”. In: *at - Automatisierungstechnik* 60.12, pp. 755–765.
- Bellman, R. (1957):** *Dynamic Programming*. 1st ed. Princeton, NJ, USA: Princeton University Press.
- Bender, P., J. Ziegler, and C. Stiller (2014):** “Lanelets: Efficient map representation for autonomous driving”. In: *IEEE Intelligent Vehicles Symposium (IV)*, pp. 420–425.
- Bergman, K. and D. Axehill (2018):** “Combining Homotopy Methods and Numerical Optimal Control to Solve Motion Planning Problems”. In: *IEEE Intelligent Vehicles Symposium (IV)*, pp. 347–354.
- Betts, J. T. (2010):** *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. 2nd ed. Society for Industrial and Applied Mathematics.
- Brandt, T., T. Sattel, and M. Böhm (2007):** “Combining haptic human-machine interaction with predictive path planning for lane-keeping and collision avoidance systems”. In: *IEEE Intelligent Vehicles Symposium (IV)*, pp. 582–587.

- Bry, A., C. Richter, A. Bachrach, and N. Roy (2015):** "Aggressive Flight of Fixed-Wing and Quadrotor Aircraft in Dense Indoor Environments". In: *The International Journal of Robotics Research* 34.7, pp. 969–1002.
- Buehler, M., K. Iagnemma, and S. Singh (2009):** *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, George Air Force Base, Victorville, California, USA. Vol. 56. Springer.
- Campbell, M., M. Egerstedt, J. P. How, and R. M. Murray (2010):** "Autonomous driving in urban environments: approaches, lessons and challenges". In: *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences* 368.1928, pp. 4649–4672.
- Cormen, T. H., C. E. Leiserson, R. L. Rivest, and C. Stein (2009):** *Introduction to Algorithms*. 3rd ed. The MIT Press.
- De Luca Alessandro and Oriolo, G. and C. Samson (1998):** "Feedback control of a nonholonomic car-like robot". In: *Robot Motion Planning and Control*. Ed. by J.-P. Laumond. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 171–253.
- Dempe, S. (2002):** *Foundations of Bilevel Programming*. 1st ed. Vol. 61. Nonconvex Optimization and Its Applications. Boston MA: Springer.
- Dickmanns, E. D. and A. Zapp (1987):** "Autonomous High Speed Road Vehicle Guidance by Computer Vision". In: *IFAC Proceedings Volumes* 20.5, Part 4. 10th Triennial IFAC Congress on Automatic Control - 1987 Volume IV, Munich, Germany, 27-31 July, pp. 221–226.
- Diehl, M., H. G. Bock, H. Diedam, and P.-B. Wieber (2005):** "Fast Direct Multiple Shooting Algorithms for Optimal Robot Control". In: *Fast Motions in Biomechanics and Robotics*. Vol. 340. Heidelberg: Springer Berlin Heidelberg, pp. 65–93.
- Dijkstra, E. W. (1959):** "A Note on Two Problems in Connexion with Graphs". In: *Numerische Mathematik* 1.1, pp. 269–271.
- Dolgov, D., S. Thrun, M. Montemerlo, and J. Diebel (2010):** "Path Planning for Autonomous Vehicles in Unknown Semi-structured Environments". In: *The International Journal of Robotics Research* 29.5, pp. 485–501.
- Donges, E. (2016):** "Driver Behavior Models". In: *Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort*. Ed. by H. Winner, S. Hakuli, F. Lotz, and C. Singer. Springer International Publishing, pp. 19–33.
- Dubins, L. E. (1957):** "On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents". In: *American Journal of Mathematics* 79.3, pp. 497–516.
- Elfes, A. (1989):** "Using occupancy grids for mobile robot perception and navigation". In: *Computer* 22.6, pp. 46–57.

- Ferguson, D., M. Darms, C. Urmson, and S. Kolski (2008a):** "Detection, prediction, and avoidance of dynamic obstacles in urban environments". In: *IEEE Intelligent Vehicles Symposium (IV)*, pp. 1149–1154.
- Ferguson, D., T. M. Howard, and M. Likhachev (2008b):** "Motion planning in urban environments". In: *Journal of Field Robotics* 25.11-12, pp. 939–960.
- Fliess, M., J. Lévine, P. Martin, and P. Rouchon (1992):** "Sur les systèmes non linéaires différentiellement plats". In: *C.R. Acad. Sci. Paris* 315, pp. 619–624.
- Fliess, M., J. Lévine, P. Martin, and P. Rouchon (1995):** "Flatness and defect of non-linear systems: introductory theory and examples". In: *International Journal of Control* 61.6, pp. 1327–1361.
- Frasch, J. V., A. Gray, M. Zanon, H. J. Ferreau, S. Sager, F. Borrelli, and M. Diehl (2013):** "An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles". In: *European Control Conference (ECC)*, pp. 4136–4141.
- Gilbert, E. G., D. W. Johnson, and S. S. Keerthi (1988):** "A fast procedure for computing the distance between complex objects in three-dimensional space". In: *IEEE Journal on Robotics and Automation* 4.2, pp. 193–203.
- Gonzalez, D., J. Perez, V. Milanés, and F. Nashashibi (2016):** "A Review of Motion Planning Techniques for Automated Vehicles". In: *IEEE Transactions on Intelligent Transportation Systems* 17.4, pp. 1135–1145.
- Götte, C., M. Keller, C. Hass, and T. Bertram (2016a):** "Model Predictive Planning and Control Applied to Critical Traffic Situations". In: *ATZ worldwide eMagazines* 2016-09, pp. 64–68.
- Götte, C., M. Keller, C. Hass, and T. Bertram (2016b):** "Modellprädiktive Planung und Regelung im Anwendungsfall kritischer Verkehrssituationen". In: *ATZ - Automobiltechnische Zeitschrift* 9, pp. 68–73.
- Götte, C., M. Keller, C. Haß, K.-H. Glander, A. Seewald, and T. Bertram (2015a):** "A model predictive combined planning and control approach for guidance of automated vehicles". In: *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pp. 69–74.
- Götte, C., M. Keller, T. Nattermann, C. Haß, K.-H. Glander, and T. Bertram (2017b):** "Ein polynombasierter Planungsansatz zur Generierung einer optimalen Trajektorie: A Polynomial based Planning Approach to the Generation of an Optimal Trajectory". In: *VDI/VDE Mechatroniktagung*. Dresden, pp. 134–139.
- Götte, C., M. Keller, T. Nattermann, C. Haß, K.-H. Glander, and T. Bertram (2017c):** "Spline-based Motion Planning for Automated Driving". In: *Proceedings of the 20th IFAC World Congress*, pp. 9444–9449.
- Götte, C., M. Keller, C. Rösmann, T. Nattermann, C. Haß, K.-H. Glander, A. Seewald, and T. Bertram (2016d):** "A Real-Time Capable Model Predictive Approach to

- Lateral Vehicle Guidance". In: *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1908–1913.
- Götte, C., C. Wissing, M. Keller, T. Nattermann, C. Haß, K.-H. Glander, and T. Bertram (2017d):** "Simulationsbasierte Entwicklung eines Trajektorienplanungsverfahrens für automatisiertes Fahren: Simulation based Development of a Trajectory Planning Approach for Automated Driving". In: *AUTOREG 2017, Automatisiertes Fahren und vernetzte Mobilität*. 8. VDI/VDE-Fachtagung. VDI Berichte 2292. Berlin, pp. 247–259.
- Hart, P. E., N. J. Nilsson, and B. Raphael (1968):** "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". In: *IEEE Transactions on Systems Science and Cybernetics* 4.2, pp. 100–107.
- Hesse, T., D. Heß, and T. Sattel (2010):** "Motion planning for passenger vehicles - Force field trajectory optimization for automated driving". In: *Proceedings of the IASTED International Conference on Robotics and Applications*.
- Hilgert, J., K. Hirsch, T. Bertram, and M. Hiller (2003):** "Emergency path planning for autonomous vehicles using elastic band theory". In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 1390–1395.
- Homann, A., M. Buss, M. Keller, K.-H. Glander, and T. Bertram (2018):** "Multi Stage Model Predictive Trajectory Set Approach for Collision Avoidance". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 945–950.
- Homann, A., C. Lienke, M. Keller, M. Buss, M. Mohamed, and T. Bertram (2019):** "Sampling-Based Trajectory Planning and Control for a Collision Avoidance System". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 2956–2962.
- Hundelshausen, F. v., M. Himmelsbach, F. Hecker, A. Mueller, and H.-J. Wünsche (2008):** "Driving with tentacles: Integral structures for sensing and motion". In: *Journal of Field Robotics* 25.9, pp. 640–673.
- Jeon, J. h., R. V. Cowlagi, S. C. Peters, S. Karaman, E. Frazzoli, P. Tsiotras, and K. Iagnemma (2013):** "Optimal motion planning with the half-car dynamical model for autonomous high-speed driving". In: *American Control Conference (ACC)*, pp. 188–193.
- Jimenez, P., F. Thomas, and C. Torras (2006):** "Collision detection algorithms for motion planning". In: *Robot Motion Planning and Control*. Ed. by J.-P. Laumond. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 305–343.
- Karaman, S. and E. Frazzoli (2011):** "Sampling-based algorithms for optimal motion planning". In: *The International Journal of Robotics Research* 30.7, pp. 846–894.
- Karaman, S., M. Walter, A. Perez, E. Frazzoli, and S. Teller (2011):** "Anytime Motion Planning using the RRT*". In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1478–1483.

- Katrakazas, C., M. Quddus, W.-H. Chen, and L. Deka (2015):** “Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions”. In: *Transportation Research Part C: Emerging Technologies* 60, pp. 416–442.
- Kavraki, L. E., P. Svestka, J.-C. Latombe, and M. H. Overmars (1996):** “Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces”. In: *IEEE Transactions on Robotics and Automation* 12, pp. 566–580.
- Keller, M., C. Hass, A. Seewald, and T. Bertram (2015):** “A Model Predictive Approach to Emergency Maneuvers in Critical Traffic Situations”. In: *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 369–374.
- Keller, M., F. Hoffmann, T. Bertram, C. Hass, and A. Seewald (2014):** “Planning of Optimal Collision Avoidance Trajectories with Timed Elastic Bands”. In: *Proceedings of the 19th IFAC World Congress*, pp. 9822–9827.
- Kelly, M. (2017):** “An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation”. In: *SIAM Review* 59.4, pp. 849–904.
- Klomp, M., M. Jonasson, L. Laine, L. Henderson, E. Regolin, and S. Schumi (2019):** “Trends in vehicle motion control for automated driving on public roads”. In: *Vehicle System Dynamics* 57.7, pp. 1028–1061.
- Kunz, F. and K. Dietmayer (2016):** “Hybrid Discrete-Parametric Optimization for Trajectory Planning in On-Road Driving Scenarios”. In: *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 802–807.
- Kuwata, Y., J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How (2009):** “Real-Time Motion Planning With Applications to Autonomous Urban Driving”. In: *IEEE Transactions on Control Systems Technology* 17.5, pp. 1105–1118.
- Kuwata, Y., J. Teo, S. Karaman, G. Fiore, E. Frazzoli, and J. How (2008):** “Motion Planning in Complex Environments Using Closed-loop Prediction”. In: *AIAA Guidance, Navigation and Control Conference and Exhibit*.
- LaValle, S. M. (1998):** “Rapidly-exploring random trees: A new tool for path planning”. PhD thesis. Iowa State University.
- LaValle, S. M. (2006):** *Planning Algorithms*. Cambridge University Press.
- Lefèvre, S., D. Vasquez, and C. Laugier (2014):** “A survey on motion prediction and risk assessment for intelligent vehicles”. In: *Robomech Journal* 1.
- Levenberg, K. (1944):** “A Method for the Solution of Certain Non-Linear Problems in Least Squares”. In: *Quarterly of Applied Mathematics* 2.2, pp. 164–168.
- Lienke, C., M. Keller, K.-H. Glander, and T. Bertram (2018b):** “An Ad-hoc Sampling-based Planner for On-road Automated Driving”. In: *IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 2371–2376.

- Lienke, C., M. Keller, K.-H. Glander, and T. Bertram (2018c):** “Environment Modeling for the Application in Optimization-based Trajectory Planning”. In: *IEEE Intelligent Vehicles Symposium (IV)*, pp. 498–503.
- Lienke, C., M. Schmidt, C. Wissing, M. Keller, C. Manna, T. Nattermann, and T. Bertram (2019a):** “Core components of automated driving – algorithms for situation analysis, decision-making, and trajectory planning”. In: *ATZ live – Automatisiertes Fahren*.
- Lienke, C., C. Wissing, M. Keller, T. Nattermann, and T. Bertram (2019b):** “Predictive Driving: Fusing Prediction and Planning for Automated Highway Driving”. In: *IEEE Transactions on Intelligent Vehicles* 4.3, pp. 456–467.
- Likhachev, M. and D. Ferguson (2008):** “Planning Long Dynamically-Feasible Maneuvers For Autonomous Vehicles”. In: *Proceedings of Robotics: Science and Systems (RSS)*.
- Likhachev, M. and D. Ferguson (2009):** “Planning Long Dynamically-Feasible Maneuvers For Autonomous Vehicles”. In: *The International Journal of Robotics Research* 28 (8), pp. 933–945.
- Likhachev, M., G. Gordon, and S. Thrun (2003):** “ARA*: Anytime A* with Provable Bounds on Sub-Optimality”. In: *Proceedings of Advances in Neural Information Processing Systems (NIPS)*. Vol. 16.
- Lynch, K. M. and F. C. Park (2017):** *Modern Robotics: Mechanics, Planning, and Control*. 1st ed. Cambridge University Press.
- Ma, L., J. Xue, K. Kawabata, J. Zhu, C. Ma, and N. Zheng (2015):** “Efficient Sampling-Based Motion Planning for On-Road Autonomous Driving”. In: *IEEE Transactions on Intelligent Transportation Systems* 16.4, pp. 1961–1976.
- Marquardt, D. W. (1963):** “An Algorithm for Least-Squares Estimation of Nonlinear Parameters”. In: *Journal of the Society for Industrial and Applied Mathematics* 11.2, pp. 431–441.
- Marti, E., M. A. de Miguel, F. Garcia, and J. Perez (2019):** “A Review of Sensor Technologies for Perception in Automated Driving”. In: *IEEE Intelligent Transportation Systems Magazine* 11.4, pp. 94–108.
- Martin, P., R. Murray, and P. Rouchon (2003):** “Flat Systems, Equivalence and Trajectory Generation”. In: CDS Technical Report, pp. 1–79.
- McNaughton, M., C. Urmson, J. M. Dolan, and J.-W. Lee (2011):** “Motion planning for autonomous driving with a conformal spatiotemporal lattice”. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4889–4895.
- Meywerk, M. (2015):** *Vehicle Dynamics*. Chichester: John Wiley & Sons Ltd.
- Miller, C., C. Pek, and M. Althoff (2018):** “Efficient Mixed-Integer Programming for Longitudinal and Lateral Motion Planning of Autonomous Vehicles”. In: *IEEE Intelligent Vehicles Symposium (IV)*, pp. 1954–1961.

- Mitschke, M. and H. Wallentowitz (2014):** *Dynamik der Kraftfahrzeuge*. Wiesbaden: Springer Vieweg.
- Montemerlo, M., J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, and S. Thrun (2008):** “Junior: The Stanford Entry in the Urban Challenge”. In: *Journal of Field Robotics* 25, pp. 569–597.
- Mouhagir, H., R. Talj, V. Cherfaoui, F. Aioun, and F. Guillemard (2016):** “Integrating safety distances with trajectory planning by modifying the occupancy grid for autonomous vehicle navigation”. In: *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1114–1119.
- Mouhagir, H., R. Talj, V. Cherfaoui, F. Aioun, and F. Guillemard (2020):** “Evidential-Based Approach for Trajectory Planning With Tentacles, for Autonomous Vehicles”. In: *IEEE Transactions on Intelligent Transportation Systems* 21.8, pp. 3485–3496.
- Mullakkal-Babu, F. A., M. Wang, B. van Arem, and R. Happee (2020):** “Empirics and Models of Fragmented Lane Changes”. In: *IEEE Open Journal of Intelligent Transportation Systems* 1, pp. 187–200.
- Nietzschmann, C., S. Klaudt, C. Klas, D. Will, and L. Eckstein (2018):** “Trajectory optimization for Car-Like Vehicles in Structured and Semi-Structured Environments”. In: *IEEE Intelligent Vehicles Symposium (IV)*, pp. 504–510.
- Nilsson, J. (2016):** “Automated Driving Maneuvers - Trajectory Planning via Convex Optimization in the Model Predictive Control Framework”. PhD thesis. Chalmers University of Technology.
- Nilsson, J., M. Brännström, E. Coelingh, and J. Fredriksson (2017):** “Lane Change Maneuvers for Automated Vehicles”. In: *IEEE Transactions on Intelligent Transportation Systems* 18.5, pp. 1087–1096.
- Nilsson, J., P. Falcone, M. Ali, and J. Sjöberg (2015):** “Receding horizon maneuver generation for automated highway driving”. In: *Control Engineering Practice* 41, pp. 124–133.
- Nocedal, J. and S. J. Wright (2006):** *Numerical Optimization*. 2nd ed. Springer Series in Operation Research and Financial Engineering. New York, NY, USA: Springer.
- Pacejka, H. B. (2007):** *Tyre and vehicle dynamics*. 2nd ed. Oxford: Butterworth-Heinemann.
- Paden, B., M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli (2016):** “A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles”. In: *IEEE Transactions on Intelligent Vehicles* 1.1, pp. 33–55.
- Pek, C. and M. Althoff (2018):** “Computationally Efficient Fail-safe Trajectory Planning for Self-driving Vehicles Using Convex Optimization”. In: *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1447–1454.

- Qian, X., F. Altché, P. Bender, C. Stiller, and A. de La Fortelle (2016):** “Optimal Trajectory Planning for Autonomous Driving Integrating Logical Constraints: An MIQP Perspective”. In: *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 205–210.
- Rajamani, R. (2012):** *Vehicle Dynamics and Control*. 2nd ed. New York, Dordrecht, Heidelberg, London: Springer Verlag.
- RAS-L (1995):** *Richtlinien für die Anlage von Straßen RAS, Teil: Linienführung RAS-L*. Forschungsgesellschaft für Straßen- und Verkehrswesen, Arbeitsgruppe Straßentwurf.
- Reeds, J. A. and L. A. Shepp (1990):** “Optimal paths for a car that goes both forwards and backwards”. In: vol. 145. 2, pp. 367–393.
- Rösmann, C., A. Makarow, and T. Bertram (2021):** “Online Motion Planning Based on Nonlinear Model Predictive Control with Non-Euclidean Rotation Groups”. In: *European Control Conference (ECC)*, pp. 1583–1590.
- SAE (2018):** *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. J3016.
- Sattel, T. and T. Brandt (2005):** “Ground vehicle guidance along collision-free trajectories using elastic bands”. In: *Proceedings of the 2005 American Control Conference*, pp. 4991–4996.
- Schlechtriemen, J., K. P. Wabersich, and K.-D. Kuhnert (2016):** “Wiggling through complex traffic: Planning trajectories constrained by predictions”. In: *IEEE Intelligent Vehicles Symposium (IV)*, pp. 1293–1300.
- Schmidt, M., M. Krüger, C. Lienke, M. Oeljeklaus, T. Nattermann, M. Mohamed, F. Hoffmann, and T. Bertram (2019a):** “Lane detection for automated driving using Deep Learning”. In: *at – Automatisierungstechnik* 67.10, pp. 866–878.
- Schmidt, M., C. Lienke, M. Oeljeklaus, M. Krüger, T. Nattermann, M. Mohamed, F. Hoffmann, and T. Bertram (2018):** “Fahrspurerkennung mit Deep Learning für automatisierte Fahrfunktionen”. In: 28. *Workshop Computational Intelligence*.
- Schmidt, M., C. Manna, J. H. Braun, C. Wissing, M. Mohamed, and T. Bertram (2019b):** “An Interaction-Aware Lane Change Behavior Planner for Automated Vehicles on Highways Based on Polygon Clipping”. In: *IEEE Robotics and Automation Letters* 4.2, pp. 1876–1883.
- Schramm, D., M. Hiller, and R. Bardini (2018):** *Vehicle Dynamics*. Berlin Heidelberg: Springer Verlag.
- Schwesinger, U., M. Rufli, P. Furgale, and R. Siegwart (2013):** “A sampling-based partial motion planning framework for system-compliant navigation along a reference path”. In: *IEEE Intelligent Vehicles Symposium (IV)*, pp. 391–396.
- Schwesinger, U., R. Siegwart, and P. Furgale (2015):** “Fast collision detection through bounding volume hierarchies in workspace-time space for sampling-based motion

- planners". In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 63–68.
- Sinha, A., P. Malo, and K. Deb (2018):** "A Review on Bilevel Optimization: From Classical to Evolutionary Approaches and Applications". In: *IEEE Transactions on Evolutionary Computation* 22.2, pp. 276–295.
- Stellato, B., G. Banjac, P. Goulart, A. Bemporad, and S. Boyd (2020):** "OSQP: an operator splitting solver for quadratic programs". In: *Mathematical Programming Computation* 12.4, pp. 637–672.
- Swaroop, D. and K. R. Rajagopal (2001):** "A review of constant time headway policy for automatic vehicle following". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 65–69.
- Tanzmeister, G. and D. Wollherr (2017):** "Evidential Grid-Based Tracking and Mapping". In: *IEEE Transactions on Intelligent Transportation Systems* 18.6, pp. 1454–1467.
- Toledo, T. and D. Zohar (2007):** "Modeling Duration of Lane Changes". In: *Transportation Research Record* 1999.1, pp. 71–78.
- Ulbrich, F., D. Goehring, T. Langner, Z. Boroujeni, and R. Rojas (2017):** "Stable Timed Elastic Band with Loose Ends". In: *IEEE Intelligent Vehicles Symposium (IV)*, pp. 186–192.
- Ulbrich, F., S. Sundermann, T. Langner, D. Goehring, and R. Rojas (2018):** "Following Cars With Elastic Bands". In: *IEEE Intelligent Vehicles Symposium (IV)*, pp. 1529–1536.
- Ulbrich, S. M. (2018):** "Towards Tactical Lane Change Behavior Planning for Automated Vehicles". PhD thesis.
- Urmson, C., J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. Howard, S. Kolski, A. Kelly, M. Likhachev, M. McNaughton, N. Miller, and D. Ferguson (2009):** "Autonomous Driving in Urban Environments: Boss and the Urban Challenge". In: *Journal of Field Robotics* 25, pp. 1–59.
- Wang, J., J. Wu, and Y. Li (2015):** "The Driving Safety Field Based on Driver–Vehicle–Road Interactions". In: *IEEE Transactions on Intelligent Transportation Systems* 16.4, pp. 2203–2214.
- Werling, M. (2011):** "Ein neues Konzept für die Trajektoriengenerierung und -stabilisierung in zeitkritischen Verkehrsszenarien". German. PhD thesis. 141 pp.
- Werling, M., S. Kammel, J. Ziegler, and L. Groell (2012):** "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds". In: *The International Journal of Robotics Research* 31, pp. 346–359.
- Werling, M. and D. Liccardo (2012):** "Automatic collision avoidance using model-predictive online optimization". In: *IEEE 51st Conference on Decision and Control (CDC)*, pp. 6309–6314.

- Werling, M., J. Ziegler, S. Kammel, and S. Thrun (2010):** “Optimal trajectory generation for dynamic street scenarios in a Frenét Frame”. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 987–993.
- Willemsen, P., J. K. Kearney, and H. Wang (2003):** “Ribbon networks for modeling navigable paths of autonomous agents in virtual urban environments”. In: *IEEE Virtual Reality*, pp. 79–86.
- Wissing, C., K.-H. Glander, C. Haß, T. Nattermann, and T. Bertram (2017):** “Development and test of a lane change prediction algorithm for automated driving”. In: *Fahrerassistenzsysteme 2017*. Ed. by R. Isermann. Wiesbaden: Springer Fachmedien Wiesbaden, pp. 385–402.
- Wissing, C., T. Nattermann, K.-H. Glander, and T. Bertram (2018):** “Trajectory Prediction for Safety Critical Maneuvers in Automated Highway Driving”. In: *IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 131–136.
- Wissing, C., T. Nattermann, K.-H. Glander, A. Seewald, and T. Bertram (2016):** “Environment Simulation for the Development, Evaluation and Verification of Underlying Algorithms for Automated Driving”. In: *AmE 2016 - Automotive meets Electronics; 7th GMM-Symposium*, pp. 9–14.
- Xu, W., J. Wei, J. M. Dolan, H. Zhao, and H. Zha (2012):** “A Real-Time Motion Planner with Trajectory Optimization for Autonomous Vehicles”. In: *IEEE International Conference on Robotics and Automation*, pp. 2061–2067.
- Ye, J. J. and D. Zhu (2010):** “New necessary optimality conditions for bilevel programs by combining MPEC and the value function approach”. In: *SIAM Journal on Optimization* 20.4, pp. 1885–1905.
- Yi, B., P. Bender, F. Bonarens, and C. Stiller (2019):** “Model Predictive Trajectory Planning for Automated Driving”. In: *IEEE Transactions on Intelligent Vehicles* 4.1, pp. 24–38.
- Ziegler, J., P. Bender, T. Dang, and C. Stiller (2014):** “Trajectory planning for Bertha — A local, continuous method”. In: *IEEE Intelligent Vehicles Symposium (IV)*, pp. 450–457.
- Ziegler, J., P. Bender, M. Schreiber, H. Lategahn, T. Strauß, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. Keller, E. Kaus, R. Herrtwich, C. Rabe, D. Pfeiffer, F. Lindner, F. Stein, F. Erbs, M. Enzweiler, C. Knoepfel, and E. Zeeb (2015):** “Making Bertha Drive — An Autonomous Journey on a Historic Route”. In: *IEEE Intelligent Transportation Systems Magazine* 6, pp. 8–20.
- Ziegler, J. and C. Stiller (2009):** “Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1879–1884.
- Ziegler, J. and C. Stiller (2010):** “Fast collision checking for intelligent vehicle motion planning”. In: *IEEE Intelligent Vehicles Symposium (IV)*, pp. 518–522.

Peer-Reviewed Publications

With first authorship

- Lienke, C., M. Schmidt, C. Wissing, M. Keller, C. Manna, T. Nattermann, and T. Bertram (2019a):** “Core components of automated driving – algorithms for situation analysis, decision-making, and trajectory planning”. In: *ATZ live – Automatisiertes Fahren*.
- Lienke, C., C. Wissing, M. Keller, T. Nattermann, and T. Bertram (2019b):** “Predictive Driving: Fusing Prediction and Planning for Automated Highway Driving”. In: *IEEE Transactions on Intelligent Vehicles* 4.3, pp. 456–467.
- Lienke, C., M. Keller, K.-H. Glander, and T. Bertram (2018b):** “An Ad-hoc Sampling-based Planner for On-road Automated Driving”. In: *IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 2371–2376.
- Lienke, C., M. Keller, K.-H. Glander, and T. Bertram (2018c):** “Environment Modeling for the Application in Optimization-based Trajectory Planning”. In: *IEEE Intelligent Vehicles Symposium (IV)*, pp. 498–503.
- Götte, C., M. Keller, T. Nattermann, C. Haß, K.-H. Glander, and T. Bertram (2017b):** “Ein polynombasierter Planungsansatz zur Generierung einer optimalen Trajektorie: A Polynomial based Planning Approach to the Generation of an Optimal Trajectory”. In: *VDI/VDE Mechatroniktagung*. Dresden, pp. 134–139.
- Götte, C., M. Keller, T. Nattermann, C. Haß, K.-H. Glander, and T. Bertram (2017c):** “Spline-based Motion Planning for Automated Driving”. In: *Proceedings of the 20th IFAC World Congress*, pp. 9444–9449.
- Götte, C., C. Wissing, M. Keller, T. Nattermann, C. Haß, K.-H. Glander, and T. Bertram (2017d):** “Simulationsbasierte Entwicklung eines Trajektorienplanungsverfahrens für automatisiertes Fahren: Simulation based Development of a Trajectory Planning Approach for Automated Driving”. In: *AUTOREG 2017, Automatisiertes Fahren und vernetzte Mobilität*. 8. VDI/VDE-Fachtagung. VDI Berichte 2292. Berlin, pp. 247–259.
- Götte, C., M. Keller, C. Hass, and T. Bertram (2016a):** “Model Predictive Planning and Control Applied to Critical Traffic Situations”. In: *ATZ worldwide eMagazines* 2016-09, pp. 64–68.
- Götte, C., M. Keller, C. Hass, and T. Bertram (2016b):** “Modellprädiktive Planung und Regelung im Anwendungsfall kritischer Verkehrssituationen”. In: *ATZ - Automobiltechnische Zeitschrift* 9, pp. 68–73.
- Götte, C., M. Keller, C. Rösmann, T. Nattermann, C. Haß, K.-H. Glander, A. Seewald, and T. Bertram (2016d):** “A Real-Time Capable Model Predictive Approach to Lateral Vehicle Guidance”. In: *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1908–1913.

Götte, C., M. Keller, C. Haß, K.-H. Glander, A. Seewald, and T. Bertram (2015a): “A model predictive combined planning and control approach for guidance of automated vehicles”. In: *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pp. 69–74.

With co-authorship

Homann, A., C. Lienke, M. Keller, M. Buss, M. Mohamed, and T. Bertram (2019): “Sampling-Based Trajectory Planning and Control for a Collision Avoidance System”. In: *IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 2956–2962.

Schmidt, M., M. Krüger, C. Lienke, M. Oeljeklaus, T. Nattermann, M. Mohamed, F. Hoffmann, and T. Bertram (2019a): “Lane detection for automated driving using Deep Learning”. In: *at – Automatisierungstechnik* 67.10, pp. 866–878.

Schmidt, M., C. Lienke, M. Oeljeklaus, M. Krüger, T. Nattermann, M. Mohamed, F. Hoffmann, and T. Bertram (2018): “Fahrspurerkennung mit Deep Learning für automatisierte Fahrfunktionen”. In: *28. Workshop Computational Intelligence*.

Additional Publications and Poster Presentations

Lienke, C., M. Keller, and T. Bertram (2018a): “Umfeldmodellierung für Trajektorienplanung in dynamischer Umgebung. Environment Modeling for Trajectory Planning in Dynamic Environments”. In: *13. DortmunderAutoTag (DAT)*.

Götte, C., M. Keller, C. Haß, K.-H. Glander, and T. Bertram (2017a): “Online-Trajektorienoptimierung mit Timed Elastic Splines. Online Trajectory Optimization with Timed Elastic Splines”. In: *12. DortmunderAutoTag (DAT)*.

Götte, C., M. Keller, C. Rösmann, T. Nattermann, C. Haß, K.-H. Glander, and T. Bertram (2016c): “Ein modellprädiktiver Ansatz zur Querführung automatisierter Fahrzeuge. A Model Predictive Approach to Lateral Guidance of Automated Vehicles”. In: *11. DortmunderAutoTag (DAT)*.

Götte, C., M. Keller, C. Haß, K.-H. Glander, A. Seewald, and T. Bertram (2015b): “Ein modellprädiktiver Ansatz zur Trajektorienplanung und Regelung von automatisierten Manövern mit Lenk- und radselektivem Bremseneingriff. A Model Predictive Approach to Trajectory Planning and Control for Automated Maneuvers using Steering Intervention and Wheel Independent Braking”. In: *10. DortmunderAutoTag (DAT)*.

Patent Applications

Keller, M., C. Wissing, C. Lienke, M. Schmidt, A. Homann, N. Stannartz, and T. Bertram (2021): “Steuerungssystem und Steuerungsverfahren für einen hierarchischen Ansatz zum Ermitteln einer Trajektorie für ein Kraftfahrzeug (DE)”. Patent DE 10 2019 128 787 A1.

- Bertram, T., K.-H. Glander, A. Homann, M. Keller, C. Lienke, T. Nattermann, M. Schmidt, and C. Wissing (2020a):** "Steuerungssystem und Steuerungsverfahren für einen hybriden Ansatz zum Ermitteln einer möglichen Trajektorie für ein Kraftfahrzeug (DE), Control System and Control Method for a Hybrid Approach to Determining a Possible Trajectory for a Motor Vehicle (EN)". Patent DE 10 2018 009 927 A1, WO 002020127011 A1.
- Bertram, T., K.-H. Glander, A. Homann, M. Keller, C. Lienke, M. Schmidt, and C. Wissing (2020b):** "Verfahren zum Betreiben eines wenigstens teilweise automatisierten Fahrzeugs (DE), Method for Operating an at least Partially Automated Vehicle (EN)". Patent DE 10 2018 123 896 A1, WO 002020002100 A1.
- Buß, M., M. Keller, K.-H. Glander, C. Lienke, C. Wissing, M. Schmidt, A. Homann, and T. Bertram (2020):** "Steuerungssystem und Steuerungsverfahren zum sampling-basierten Planen möglicher Trajektorien für Kraftfahrzeuge (DE), Control System and Control Method for Sampling-Based Planning of Possible Trajectories for Motor Vehicles (EN)". Patent DE 10 2018 008 624 A1, WO 002020089311 A1.

Supervised Theses

- Dorpmüller, P. (2019):** "Entwicklung eines optimierungsbasierten Trajektorienplanungsverfahrens für automatisierte Fahrzeuge". Master's thesis.
- Sarmiento, D. (2019):** "Development of an Incremental Search Trajectory Planner for Automated Driving in Structured Environments". Master's thesis.
- Vetter, S. (2018):** "Kombination einer Sample-Based Trajektorienplanung mit einer modellprädikativen Trajektorienfolgeregelung zur Kollisionvermeidung automatisierter Fahrzeuge". Bachelor's thesis.
- Dorpmüller, P. (2017):** "Analyse und Vergleich von numerischen Optimierungsalgorithmen für die Trajektorienplanung automatisierter Fahrzeuge". Bachelor's thesis.
- Parsiegel, F. (2017):** "Vergleich von Fahrdynamikmodellen zur Anwendung von modellprädiktiven Verfahren in automatisierten Fahrfunktionen". Bachelor's thesis.
- Schmidt, M. (2017):** "Online-Kalibrierung eines Kamerasystems in Straßenverkehrsszenarien zur Validierung automatisierter Fahrfunktionen". Master's thesis.
- Wunsch, C. (2017):** "Linearisierung eines Einspurmodells zur kombinierten Längs- und Querregelung eines Fahrzeuges". Bachelor's thesis.
- Diehl, C. (2016):** "Anwendung eines Trajektorienscharverfahrens für die Planungsaufgabe automatisierter Fahrzeuge". Bachelor's thesis.
- Spitzlei, D. (2016):** "Entwicklung, Analyse und Vergleich von Fahrdynamikregelungen zur Trajektorienfolge". Master's thesis.
- Homann, A. (2015):** "Ein Verfahren zur Fahrzeugführung mittels echtzeitfähiger Trajektorienoptimierung und Fahrdynamikregelung". Master's thesis.

Supervised Student Project Groups

Bartsch, K., Q. T. Bui, C. Diehl, A. Dikarew, A. Hugenroth, T. Kiratli, and Y. Sun (2017): "Entwicklung automatisierter Fahrerassistenzsysteme".

Overmann, A. S., B. Möllenbeck, E. Mares, F. Albers, J. Bruchhaus, M. A. Ali, M. Hamza, M. S. Irshad, and W. Schulz (2016): "Development of a multistep test environment for advanced driver assistance systems".

A

Appendix

A.1. Terminology and Definitions

Configuration Space A vehicle configuration o completely specifies the position relative to a fixed coordinate system. The configuration space is the set of all possible configurations and its dimension matches the number of degrees of freedom of the vehicle. Thus, it can also be denoted as the space of the system's degrees of freedom. The link between the configuration and the state space is given by $o = \kappa(x)$ with function $\kappa : \mathcal{X} \rightarrow \mathcal{C}$, which returns the configuration $o \in \mathcal{C}$ associated with state $x \in \mathcal{X}$. In dynamic environments, the vehicle's configuration space needs to include a notion of time (LaValle 2006; Lynch and Park 2017).

Holonomic and Non-Holonomic Systems Holonomic constraints are differential constraints that are integrable, which means that holonomic constraints restrict the vehicle's configuration space. Non-holonomic constraints on the other hand are differential constraints, that cannot be integrated. This means they cannot be converted into constraints that involve no derivatives and thus cannot be expressed as constraints on the vehicle's pose. Accordingly they do not restrict the reachable configuration space, but the space of differential motions. A non-holonomic constraint for vehicles arises from the rolling-without-slipping property, that prevents from instantaneous lateral motion (LaValle 2006, p. 791).

Normal Driving Situation A normal driving situation is defined in terms of acting tire forces within the linearized area of tire characteristics, which approximately is one third of the maximum lateral tire force. For dry roads this corresponds to circa 4 m/s^2 .

Trajectory versus Path Planning In literature a very confusing aspect is that the terms path planning and trajectory planning are often used interchangeably. At this point it should be clarified that there is a strict separation between these two terms as a trajectory considers time information explicitly, while path planning relates to a purely geometric problem in the configuration space.

Maneuver The term maneuver is used to summarize a sequence of actions that drive the vehicle from one steady state to another (Ulbrich 2018, p. 19). A maneuver is defined with respect to the given road topology, which basically includes lateral vehicle

motion like lane keeping and lane changing, but can also refer to the longitudinal motion component and the spatial relation to other vehicles (e.g. overtaking).

Scene, Situation and Scenario The traffic scene covers all directly measurable characteristics of the surrounding environment e.g. scenery and dynamic elements. A scene describes the environment at one point in time and *inter alia* suffers from incompleteness and uncertainty. A situation on the other hand supplements a scene with some kind of interpretation to reveal relevant additional information, such that a more comprehensive description of the environment can be derived. A scenario describes the temporal development between several scenes. Definitions are based on the work of Ulbrich (2018).

Motion Planning The term motion planning is used with respect to problems that involve obstacle avoidance and potentially other constraints. Hence, motion planning represents the generic term for path and trajectory planning.

Behavior In accordance to Ulbrich (2018) the term behavior is used to group actions or activities that result in observable outcomes. In this context it mainly relates to the question of how something is performed (i.e. formulating the driving style in a descriptive manner), covering aspects that may relate to different hierarchical levels of the driving task.

Completeness A planning algorithm is denoted as complete, if it terminates in finite time and returns a valid solution to the trajectory planning problem or ascertains that no valid solution exists. In the context of sampling-based trajectory planning approaches it is often referred to the term resolution completeness, which accounts for the discretized space.

Trajectory Optimization and Optimal Control In literature the terms trajectory optimization and optimal control are used interchangeably (see e.g. Kelly (2017)). In this thesis the term trajectory optimization is predominantly connected, but not necessarily restricted to the description of solving a nonlinear program.

On-road and off-road driving In this thesis the term off-road is used to denote navigation in unstructured environments. On-road driving on the other hand refers to navigating in structured (driving applications) and semi-structured (parking applications) environments.

A.2. Components of the Functional Architecture

Taking a closer look at the functional architecture of automated vehicles, in this section all modules related to the ego vehicle driving task are described in more detail. For an approach that shows an exemplary solution with respect to the interplay between selected components it is referred to Lienke et al. (2019a). In the following each component of the functional architecture is outlined to provide further information with respect to the general purpose and underlying concepts.

Perception

The part of perception deals with the important task to generate a preferably high quality reflection of the real world, using supplementary sensor concepts. Since there is no standardized hardware configuration for automated vehicles, the sensor setup is subject to thorough assessment, taking for example performance or cost aspects into consideration. Especially the choice and arrangement of exteroceptive¹ sensors largely contribute to the overall perception quality. As subsequent modules in the functional architecture rely on reliable and accurate environment data, the task of perception becomes even more important. Common sensors used in this context comprise cameras, radars as well as lidars and ultrasound sensors. A major challenge represents the fact, that the sensors have to cope with a wide range of external conditions, which is why pros and cons of the respective technology have to be balanced carefully against each other. Marti et al. (2019) give a summary on sensor technologies for perception in automated driving. An approach to lane detection using deep learning techniques has been described by Schmidt et al. (2018) and Schmidt et al. (2019a).

To consolidate the sensor measurements objects are fused to one common representation and tracked over a number of cycles to trace the development over time. In addition a representation of the road topology is deduced from detected lane boundaries inferring the available free space. The description of the surrounding environment can be enriched by map data and information gained from vehicle-to-x communication. Both types provide data with high accuracy and also extend the range of validity, since the distribution of cloud-based information and communication services represent a counterpart to measurements conducted by on-board sensors. Still, this requires the ability of precise localization within the current setting. Finally, with all the information obtained, an environment model is generated.

Situation Analysis

The environment model obtained from the perception module represents the basis for a further analysis of the traffic scene. The topics of intention prediction and estimation of future trajectories with respect to other traffic participants need to be addressed to extract meaningful knowledge about the future development of the current situation. Therefore, the task of situation analysis can be subdivided into maneuver detection, maneuver dynamics estimation and trajectory prediction. Especially the latter part is vital for the subsequent maneuver and trajectory planning, as it enhances the environment model with assumptions on the future evolution of the situation (predictive environment model) enabling anticipatory driving. A comprehensive review on motion prediction techniques is given by Lefèvre et al. (2014).

Mission Planning

At the beginning a mission goal connected to a specific task or event has to be provided inferring a desired destination for the automated vehicle. Based on that mission

¹In contrast to proprioceptive sensors, which relate to the ego vehicle state, measuring e.g. the velocity and accelerations.

several waypoints can be derived, which are connected via the route planning module. A route on a road network is planned neglecting dynamic events (e.g. obstacles, traffic lights) and vehicle constraints. Among others possible objectives for route planning involve travel time, distance or energy consumption. Route planning could also consider predicted or live data indicating the actual traffic flow comprising as well traffic congestions or blockages (macroscopic point of view).

Decision Making

The decision making module is a high level maneuver planner designed for the task of tactical reasoning with topological awareness. Therefore, distinct maneuver classes have to be identified during the maneuver detection process and to distinguish between different classes the spatiotemporal topology of the surrounding environment has to be analyzed. This allows the maneuver planner to make discrete decisions, whether for example to pass an obstacle vehicle to the right or left. Beside lane change and lane keeping commands, possible maneuvers are for example parking or stopping. The process of decision making starts with a maneuver request, which is followed by a maneuver analysis and decision. A maneuver request can be initiated by an automated process (e.g. as a static event triggered by the mission planning or as a reaction on the current traffic situation) or by the human driver (e.g. by activating the indicator). Furthermore, a request can be categorized as mandatory or discretionary. Whereas the former is due to infrastructural and mission related aspects the latter corresponds to expected improvements regarding the actual situation. For maneuver analysis there are two prevalent approaches in which the complexity of the detection part varies significantly. The first approach simplifies the possible set of maneuver classes to lane keeping and lane change maneuvers and deals with the detection of available gaps to perform a lane change. A more sophisticated solution is the application of maneuver pattern analysis that is often connected with sampling of maneuver trajectory prototypes. In this approach time augmented topological information is gathered to identify spatiotemporal distinct maneuver variants. According to Ardel et al. (2012), there are two main approaches to decision making in the context of automated driving that can be summarized as rule or utility function based decision making. In the sense of gap acceptance models the detected available gaps are compared with respect to the smallest acceptable gap, also denoted as critical gap. This critical gap is in general dependent on kinematic quantities such as the relative speed to the relevant obstacle vehicles. A key feature of decision making is the spatiotemporal topological analysis, which does not necessarily include the question of reachability. The trajectory planning benefits, as the property of topological awareness allows for example the efficient reduction of the search space for sampling-based methods by the formulation of sampling ranges as well as it significantly helps to provide a meaningful initialization for optimization-based methods.

Trajectory Planning

Based on the future driving strategy provided by the decision making a motion plan has to be generated accordingly, translating the desired behavior into a trajectory that

can be tracked by a low-level vehicle dynamics controller. In the context of automated driving the trajectory planner has to cope with imperfect and incomplete knowledge of the surrounding environment or rather the highly uncertain motion of other predicted traffic participants obtained from situation analysis. The resulting trajectory must be dynamically feasible and consider other obstacle vehicles and traffic participants to avoid collisions. Especially at high velocities (e.g. highway driving) this is not a trivial task, since the reachability of a safe state must be guaranteed at all times. Moreover, to allow for a pleasant drive, the aspect of passenger comfort needs to be addressed. To cover dynamic scenarios a time parameterized representation of the future motion of the ego vehicle is preferred over a purely spatial description in terms of a path. According to Werling (2011, p. 5) static traffic scenarios like parking without dynamic traffic participants can be seen as a special case of dynamic scenarios. It can hence be concluded that with a suitable trajectory planning and stabilization approach all traffic maneuvers are formally manageable. It is moreover important to consider the coupled lateral and longitudinal characteristics of the vehicle, since particularly overtaking and merging maneuvers result in a coupled lateral and longitudinal motion planning problem. Differential constraints arising from the vehicle dynamics complicate the process of trajectory planning. According to LaValle (2006) one approach is to ignore the differential constraints in the planning phase and assume that the subsequent controller will handle them appropriately. A better alternative can be seen in the direct inclusion of vehicle dynamics within the planning process, resulting in a system compliant motion plan.

Vehicle Dynamics Control

Since the exact modeling of vehicle dynamics cannot be considered within the planning phase at reasonable expense, control algorithms account for the remaining modeling errors of the trajectory planner. An introduction of control related topics in automated driving is e.g. given by Klomp et al. (2019).

Approaches to trajectory following control for normal driving situations are often decomposed into separate lateral and longitudinal control concepts. Still, in applications in which the coupled lateral and longitudinal dynamics come into effect, a comprehensive controller design is necessary. From the vast number of control approaches to trajectory and path tracking three main categories stand out: Proportional-integral-derivative (PID) controller, state feedback controller and model predictive control concepts. PID controllers and related concepts are used as cascaded or parallel version to control position and yaw rate. Because parameter tuning works quite intuitive they are widely distributed among lateral and longitudinal control approaches. Another established control concept can be seen in state feedback control, which includes e.g. the use of dynamic feedback linearization, and differentially flat outputs design. De Luca and Samson (1998) give an overview with respect to feedback control of a nonholonomic vehicle. Model predictive control uses an internal model to predict the future motion of the vehicle. By this means the optimal control inputs can be derived and system constraints can be handled explicitly. The major drawback with respect to this kind of control certainly is the intense use of computational resources.

A.3. Supplements to Environment-aware Maneuver Planning

Further Maneuver Planning Example

To give another example with respect to the functioning of the environment-aware maneuver planning the test case shown in chapter 4 is changed. This is leading to a cut-in and cut-out maneuver in front of the ego vehicle, which is now in lane keeping mode instead of performing a lane change maneuver. The obstacle vehicle ID:2 is configured to make a double lane change, all other obstacle vehicles stay in their respective lanes. The test case definition as shown in Table A.1 is similar to the one given in Table 4.1. Note that solely target lanes for the ego vehicle and obstacle vehicle ID:2 have changed. This means that the ego vehicle and obstacle vehicles ID:1, ID:3

Table A.1.: Test case definition with start conditions of the ego vehicle and obstacle vehicles.

ID []	$F p_x$ [m]	$F p_y$ [m]	$F v$ [km/h]	$F \psi$ [rad]	target lane []	$F v_{des}$ [km/h]
ego	0	0	100	0	2	130
1	40	2.5	100	0.1	2	not known
2	20	-3.35	90	0.05	1	not known
3	-25	3.75	110	0	1	not known
4	25	4.65	120	0.1	1	not known

and ID:4 intend to stay on their current lane, whereas obstacle vehicle ID:2 performs a double lane change from the right to the left lane given with respect to the ego vehicle. The impeding vehicles prediction as shown in Figure A.1 predicts obstacle vehicle ID:2 to become the lead vehicle at 1.9 s and to become the tail vehicle on the left at 4.3 s. This means that obstacle vehicle ID:2 cuts in in front of the ego vehicle and merges into the gap between obstacle vehicles ID:3 and ID:4 on the left lane. On the other hand, since obstacle vehicle ID:2 is driving at a slower speed than the ego vehicle, in a first step it is preliminarily assumed that the ego vehicle passes obstacle vehicle ID:2.

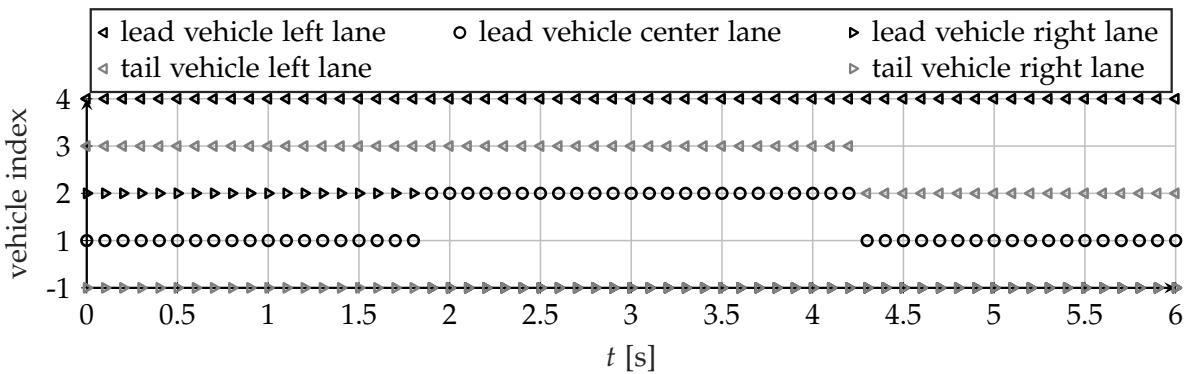


Figure A.1.: Predicted impeding vehicles shown for the considered example test case. As obstacle vehicle ID:2 is changing lanes respective lead and tail vehicles change accordingly. In case that no tail vehicle for the right or left lane exists it is mapped to a vehicle index of -1.

For lane keeping the only relevant position bound is constituted by the lead vehicle position on the current ego lane. The corresponding upper position bound for the considered example is shown in Figure A.2. Clearly, the cut-in and cut-out of obstacle

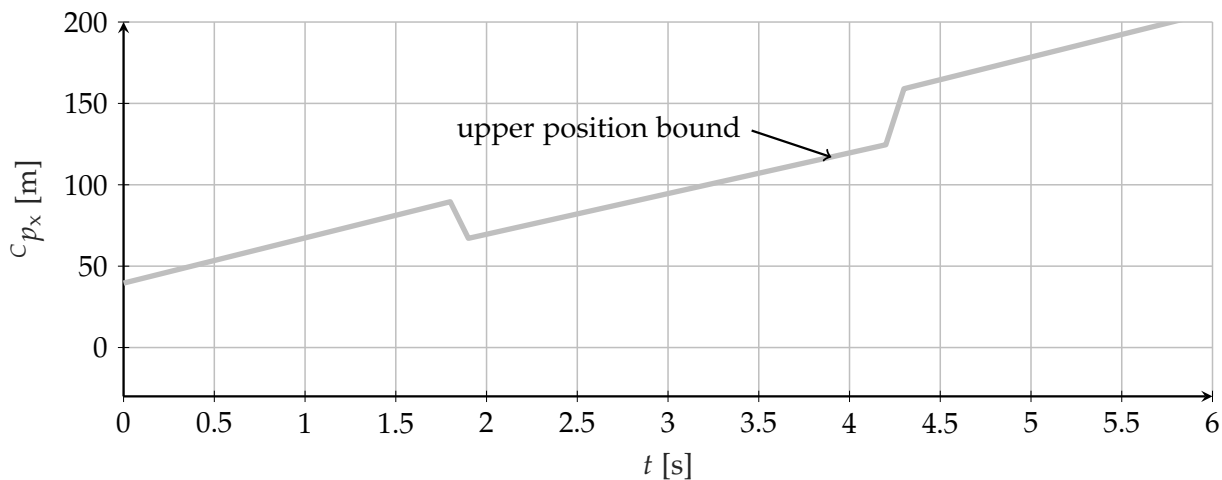


Figure A.2.: Merged relevant information for the ego vehicle. Since the ego vehicle is performing a lane keeping maneuver there is no lower position bound as no tail vehicle for the center lane is recorded. The upper position bound arises from obstacle vehicle ID:1 on the center lane and obstacle vehicle ID:2 cutting in and out in front of the ego vehicle at $t = 1.9$ s and $t = 4.3$ s, respectively.

vehicle ID:2 is visible, as during the time of $t = 1.9$ s and $t = 4.3$ s the upper position bound is reduced.

Analogous to the procedure described in chapter 4 the drivable, the admissible and the unimpeded region are identified to label areas that fulfill certain characteristics with respect to the ego vehicle position. The drivable region refers to the capabilities of the ego vehicle in terms of the vehicle dynamics. The admissible region and the unimpeded region take surrounding obstacle vehicles into account in order to integrate collision avoidance into the maneuver plan and to consider desired safety margins. The composition of the admissible region is shown in Figure A.3. With respect to the possible ego vehicle motion the impact of the limiting upper position bound is relatively small, as only a small part of the drivable region is excluded.

The unimpeded region depicted in Figure A.4 maintains the characteristics of the admissible region and adds a desired safety margin to other relevant vehicles. For the considered test case this is achieved in terms of the lead vehicle safe position. This time, in comparison to the impact of the upper position bound, the impact of the lead vehicle safe position adds more difficult constraints on the desired motion of the ego vehicle. Especially at around $t = 1.9$ s the range of corresponding longitudinal positions, which adhere to the desired safety margin is quite small.

The generation of bounded reference states is solely following a predetermined policy to restrict the longitudinal ego vehicle position, paying less regard to the actual drivability. The resulting reference that in the considered case is in fact bounded by the lead vehicle safe position is shown in Figure A.5. As already mentioned, the aspect of drivability is not considered in a first step, but taken into account via the generation

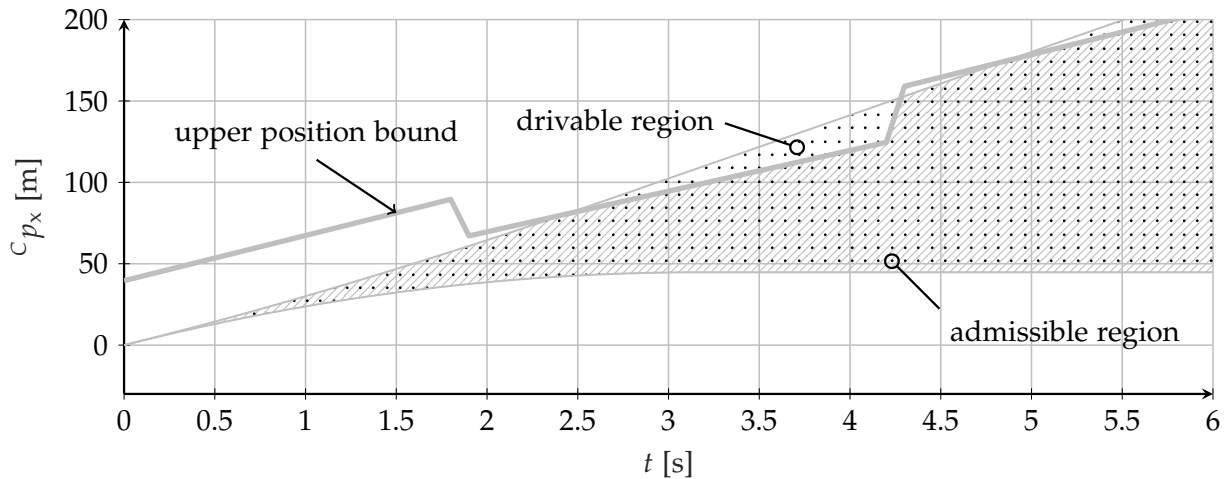


Figure A.3.: The admissible region is constituted by limiting the drivable region by the upper position bound. Note that in general the lower position bound might also be a limiting factor, hence representing a potential border of the admissible region.

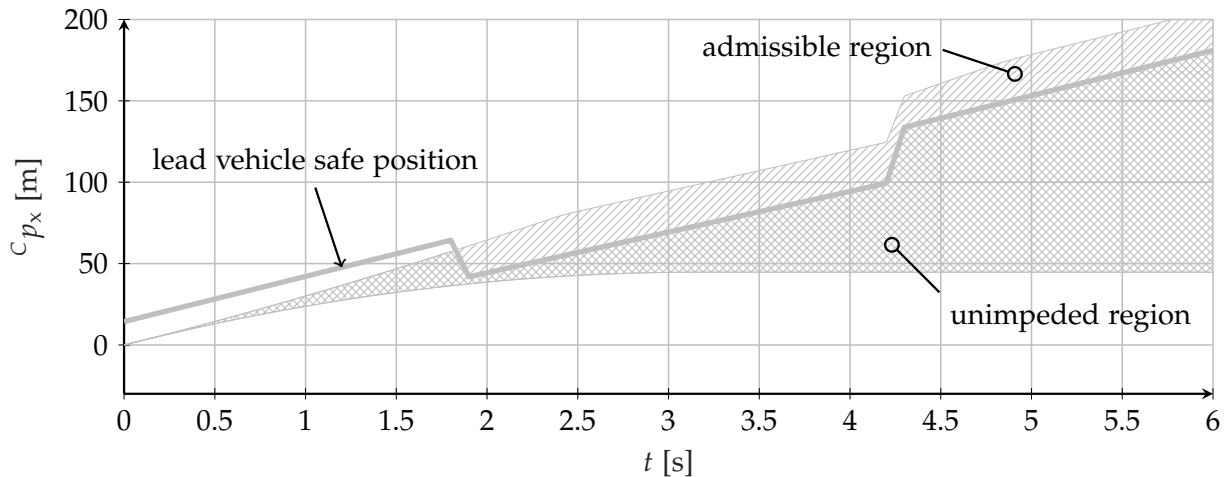


Figure A.4.: The admissible region and the lead vehicle vehicle safe position constitute to the unimpeded region.

of an optimal reference position. The least squares fit on the velocity maintains the desired characteristic of the ego demands, while accelerations comply to drivability by explicitly constraining the solution of the quadratic program in that respect. The optimal reference position for the considered example is given in Figure A.6. As can be seen, the application of a least-squares fit to generate the optimal reference velocity rigorously smoothens the extreme characteristic of the bounded reference velocity. The fact that in the first part of the maneuver the optimal reference velocity is mainly attracted by the desired ego vehicle set speed of 130 km/h explains the increase of the ego velocity, which accelerates from 100 km/h to approximately 107 km/h. To account for slower obstacle vehicle ID:2 the ego vehicle speed is then reduced to 90 km/h. Finally, after the cut-out of obstacle vehicle ID:2 the ego vehicle adapts its speed to 100 km/h in accordance to the new lead vehicle ID:1. Instead of directly optimizing the longitudinal position as such, the chosen procedure to take the bounded velocity

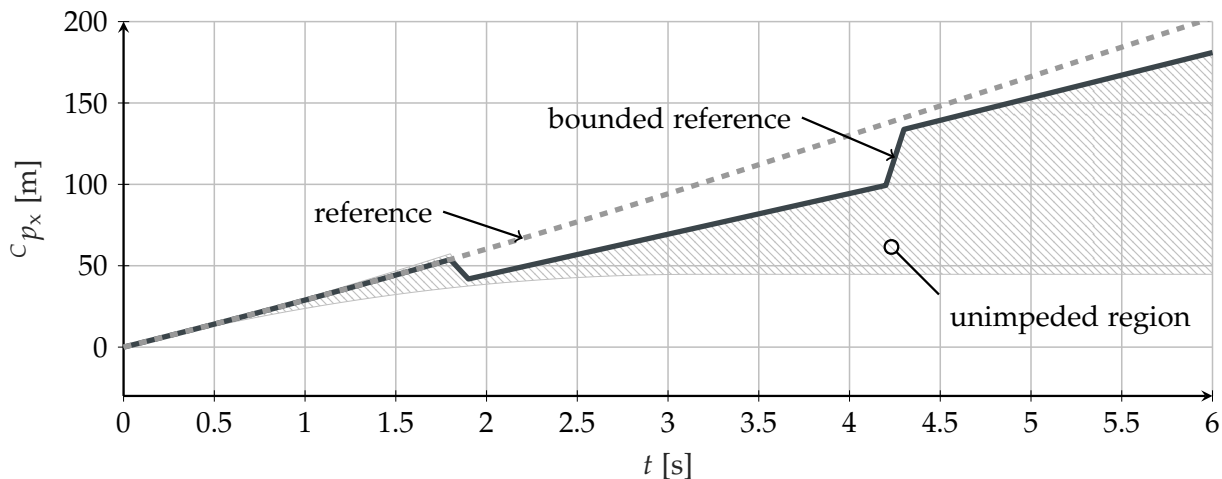


Figure A.5.: The reference position is obtained by integration with the aim to reach the set speed with a desired comfortable acceleration. The reference position is then bounded by the unimpeded region yielding the bounded reference position.

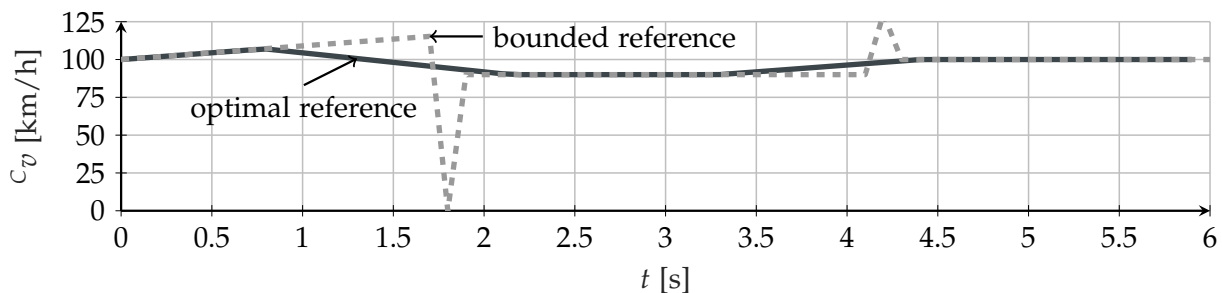
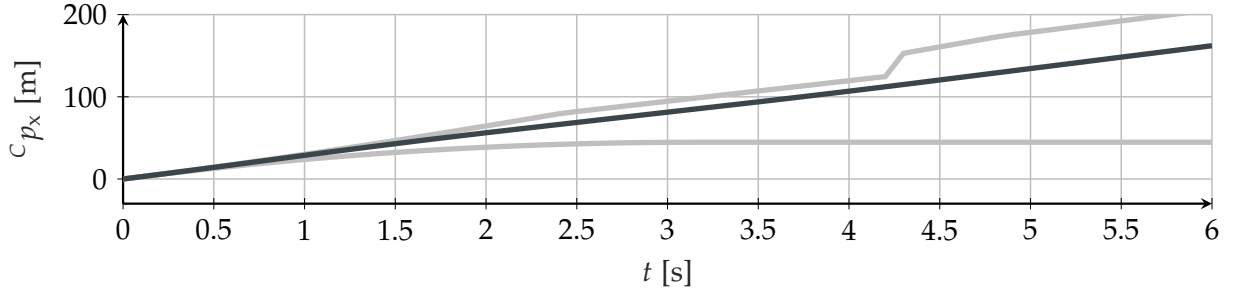


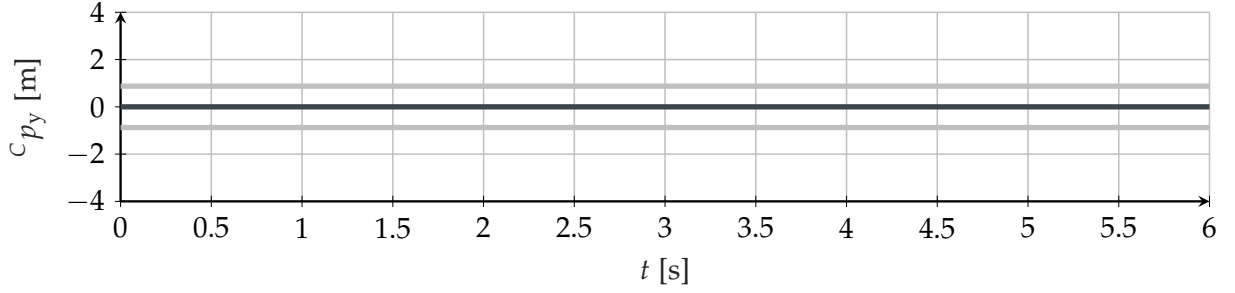
Figure A.6.: The optimal reference position is derived from the optimal reference velocity that is calculated by a least squares fit on the bounded reference velocity subject to position constraints (from admissible region), velocity constraints (from drivable region) and acceleration constraints (comfort acceleration/deceleration).

as reference facilitates to find a valid solution on the one hand and enables to find a final solution as close as possible to the initially desired ego velocity (i.e. ego vehicle set speed) on the other. As a consequence the ego vehicle smoothly adapts its speed with respect to the cut-in maneuver of obstacle vehicle ID:2, accepting to violate the bounds given by the lead vehicle safe position. The latter can be regarded as a desired feature, since strong reactions to the behavior of other traffic participants in terms of keeping a safety margin even in case of a cut-in maneuver of an obstacle vehicle could be considered as unnecessary. Because the admissible region is taken to constrain the longitudinal position of the ego vehicle, collision avoidance is still guaranteed by means of the chosen definition of the quadratic programming problem 4.2.1.

Finally, the longitudinal maneuver trajectory and target region as well as the lateral maneuver trajectory and target region are generated. The overall result is shown in Figure A.7. In the first part of the ego vehicle lane keeping maneuver the longitudinal ego vehicle motion is predominantly limited by the aspect of drivability (imposed in terms of the drivable region). In the further course the ego vehicle then manages to



(a) Longitudinal maneuver trajectory and longitudinal target region.



(b) Lateral maneuver trajectory and lateral target region.

Figure A.7.: Final maneuver trajectory and target region of the maneuver planning approach.

retain at least a small safety distance. Obviously, since the ego vehicle performs lane keeping the lateral maneuver trajectory in curvilinear coordinates is constantly zero.

Details on Optimal Reference States Calculation

For optimal reference calculation the quadratic programming problem 4.2.1 is set up. The elements of respective matrices are chosen to describe a least squares fit with respect to the bounded reference velocity, subject to linear constraints on position, velocity and acceleration. The decision variable vector $\hat{\mathbf{p}}$ can be subdivided into two parts. The first part is represented by the longitudinal position vector \mathbf{p}_x , whereas the second part represents the actual variables used to realize an optimal least squares fit at each trajectory point $k = 0, 1, \dots, K$:

$$\hat{\mathbf{p}} = [\mathbf{p}_x^T, r_1, r_2, \dots, r_k, \dots, r_K]^T. \quad (\text{A.3.1})$$

The cost matrix is given by:

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 1 \end{bmatrix}, \quad (\text{A.3.2})$$

whereas the cost vector \mathbf{q} is zero vector, such that the linear cost term can be omitted. The matrix dedicated to the least squares formulation is:

$$\mathbf{A}_{\text{lsq}} = \begin{bmatrix} \frac{-1}{\Delta t} & \frac{1}{\Delta t} & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & \frac{-1}{\Delta t} & \frac{1}{\Delta t} & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & 0 & 0 & \ddots & \vdots & \vdots \\ \vdots & \vdots & 0 & \frac{-1}{\Delta t} & \frac{1}{\Delta t} & \vdots & \vdots & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}. \quad (\text{A.3.3})$$

The matrix composition clearly reveals the least squares fit in terms of the velocity, since the first derivative of the position is calculated with respect to time by applying the difference quotient. Moreover, the two separate parts of the matrix are visible, highlighting again the subdivision of the decision vector on account of the chosen problem formulation. Finally, the constraint matrices for the position, velocity and acceleration are given as follows:

$$\mathbf{A}_{\text{pos}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & \vdots & \vdots & \vdots & 0 & \ddots & \vdots & \vdots \\ \vdots & \vdots & 0 & 1 & 0 & \vdots & \vdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (\text{A.3.4})$$

$$\mathbf{A}_{\text{vel}} = \begin{bmatrix} \frac{-1}{\Delta t} & \frac{1}{\Delta t} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{-1}{\Delta t} & \frac{1}{\Delta t} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & 0 & 0 & \ddots & \vdots & \vdots \\ \vdots & \vdots & 0 & \frac{-1}{\Delta t} & \frac{1}{\Delta t} & \vdots & \vdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (\text{A.3.5})$$

$$\mathbf{A}_{\text{acc}} = \begin{bmatrix} \frac{1}{\Delta t} & \frac{-2}{\Delta t} & \frac{1}{\Delta t} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\Delta t} & \frac{-2}{\Delta t} & \frac{1}{\Delta t} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \vdots & 0 & 0 & \ddots & \vdots & \vdots \\ \vdots & \vdots & 0 & \frac{1}{\Delta t} & \frac{-2}{\Delta t} & \frac{1}{\Delta t} & \vdots & \vdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (\text{A.3.6})$$

It can be seen that the matrices represent the prerequisite mapping of the position to the respective derivative. Numerical differentiation is applied to approximate the first and second derivative of the position via the difference quotient.

A.4. Linear Single Track Model

To obtain the linear single track model equations the equations 5.1.1 and 5.1.2 are substituted and the resulting equations are further linearized, which yields:

$$m\dot{v} = F_{x,v} + F_{x,h} - F_{a,x}, \quad (\text{A.4.1})$$

$$m(\dot{\beta} + \dot{\psi}) + m\dot{v}\beta = c_v(-\beta + \delta_r - l_v \frac{\dot{\psi}}{v}) + c_h(-\beta + l_h \frac{\dot{\psi}}{v}) - F_{a,y}, \quad (\text{A.4.2})$$

$$J_z \ddot{\psi} = c_v l_v (-\beta + \delta_r - l_v \frac{\dot{\psi}}{v}) - c_h l_h (-\beta + l_h \frac{\dot{\psi}}{v}) - F_{a,y} l_c. \quad (\text{A.4.3})$$

In the case of driving at a constant speed $\dot{v} = 0$ and furthermore neglecting the aerodynamic forces the model can be written as:

$$\dot{\beta} = -\frac{c_v + c_h}{mv} \beta + \frac{mv^2 - (c_h l_h - c_v l_v)}{mv^2} \dot{\psi} - \frac{c_v}{mv} \delta_r, \quad (\text{A.4.4})$$

$$\ddot{\psi} = -\frac{c_h l_h - c_v l_v}{J_z} \beta - \frac{c_h l_h^2 + c_v l_v^2}{J_z v} \dot{\psi} + \frac{c_v l_v}{J_z} \delta_r. \quad (\text{A.4.5})$$

A.5. Kinematic Vehicle Model with Ackermann Steering

A kinematic vehicle model describes the motion without the consideration of present forces (Rajamani 2012, p. 20). In fact this is the case at small speeds, when the lateral forces disappear. Then the tire slip angles at both wheels are zero and the following assumption holds:

$$\tan(\delta_A) \approx \frac{l_a}{\rho_{cc}}. \quad (\text{A.5.1})$$

This equation is related to equation 5.1.22 in which additionally a small angle assumption has been made. With steady state assumption ($\dot{\beta} = 0$ in equation 5.1.14) this leads to:

$$\dot{\psi} = \frac{v}{l_a} \tan(\delta_A) \quad (\text{A.5.2})$$

for the relation between yaw rate $\dot{\psi}$ and the Ackermann steer angle δ_A .

A.6. Calculation of the Minimum Curve Radius according to RAS-L

The minimum cross slope is defined as $\xi_{\min} = 2.5\%$, whereas the maximum cross slope is $\xi_{\max} = 8\%$ in general, but limited to $\xi_{\lim} = 7\%$ when calculating the minimum curve radius. In the design principles the exploitation of the maximum radial adhesion coefficient is defined as $\zeta_{\max} = 50\%$ in case of ξ_{\max} and $\zeta_{\min} = 10\%$ in case of ξ_{\min} :

$$\rho_{\min} = \frac{v^2}{3.6^2 a_g (\mu_r \zeta + \xi)}, \quad (\text{A.6.1})$$

$$\mu_t = 0.241(0.01 v)^2 - 0.721(0.01 v) + 0.708, \quad (\text{A.6.2})$$

$$\mu_r = 0.925 \mu_t. \quad (\text{A.6.3})$$

A description of the utilized variables is given in Table A.2.

Table A.2.: Variable description.

variable	unit	description
ρ_{\min}	[m]	minimum curve radius
v	[km/h]	velocity
a_g	[m/s ²]	gravitation
μ_t	[]	maximum tangential adhesion coefficient
μ_r	[]	maximum radial adhesion coefficient
ζ	[]	exploitation of the maximum radial adhesion coefficient
ξ	[]	cross slope

A.7. Inverse Flat Transform

The derivation of the longitudinal position p_x and lateral position p_y is straightforward since the flat output exactly corresponds to these quantities:

$$p_x = z_{x,1}, \quad (\text{A.7.1})$$

$$p_y = z_{y,1}. \quad (\text{A.7.2})$$

Starting from equation 5.1.29 to derive the velocity and yaw rate, it is:

$$z_{x,2} = \cos(\psi)v, \quad (\text{A.7.3})$$

$$z_{y,2} = \sin(\psi)v, \quad (\text{A.7.4})$$

which can be rearranged to

$$v = \frac{z_{x,2}}{\cos(\psi)} = \frac{z_{y,2}}{\sin(\psi)}. \quad (\text{A.7.5})$$

When solving this for yaw angle ψ it yields:

$$\psi = \arctan\left(\frac{z_{y,2}}{z_{x,2}}\right) \quad (\text{A.7.6})$$

and the velocity can be determined via equation A.7.5 to be:

$$v = \frac{z_{x,2}}{\cos\left(\arctan\left(\frac{z_{y,2}}{z_{x,2}}\right)\right)}. \quad (\text{A.7.7})$$

To derive the control inputs the starting point is once again given by equation 5.1.29:

$$z_{x,3} = \cos(\psi)a - \sin(\psi)\dot{\psi}v, \quad (\text{A.7.8})$$

$$z_{y,3} = \sin(\psi)a + \cos(\psi)\dot{\psi}v, \quad (\text{A.7.9})$$

From equation A.7.8

$$a = \frac{z_{x,3} + \sin(\psi)\dot{\psi}v}{\cos(\psi)} \quad (\text{A.7.10})$$

can directly be derived. Inserting this result into equation A.7.9 yields:

$$z_{y,3} = \sin(\psi) \left(\frac{z_{x,3} + \sin(\psi)\dot{\psi}v}{\cos(\psi)} \right) + \cos(\psi)\dot{\psi}v \quad (\text{A.7.11})$$

and after some algebraic transformations it finally is:

$$\dot{\psi} = \frac{z_{y,3} - \tan(\psi)z_{x,3}}{\tan(\psi)\sin(\psi)v + \cos(\psi)v}. \quad (\text{A.7.12})$$

Now, by substituting equation A.7.12 into equation A.7.10 the acceleration is given by:

$$a = \frac{z_{x,3}}{\cos(\psi)} + \frac{\sin(\psi)v}{\cos(\psi)} \frac{z_{y,3} - \tan(\psi)z_{x,3}}{\tan(\psi)\sin(\psi)v + \cos(\psi)v}. \quad (\text{A.7.13})$$

With equation A.7.12 and the help of equation 5.1.25c from the system dynamics of the applied vehicle model the steering angle is then given by:

$$\delta_r = \frac{z_{y,3} - \tan(\psi)z_{x,3}}{\tan(\psi)\sin(\psi)v + \cos(\psi)v} \frac{l_a(1 + (v/v_{ch})^2)}{v}. \quad (\text{A.7.14})$$

A.8. Adaptive Sampling Strategy

The adaptive sampling strategy has successfully been applied in various applications related to automated driving (Lienke et al. 2018b; Homann et al. 2018; Homann et al. 2019). Assuming that the optimal sampling variable for the current planning step is close to the one of the last planning cycle a finer resolution around the actual value is desired. Nevertheless, minimum and maximum values should always be considered. Two parabolas \mathcal{U}_1 and \mathcal{U}_2 are utilized to transform the linearly discretized values $\bar{\delta}$ online and in dependence on actual value $\bar{\delta}_{act}$. The respective coefficients are determined from conditions:

$$\mathcal{U}_1(\bar{\delta}_{min}) = \bar{\delta}_{min}, \quad \mathcal{U}_1(\bar{\delta}_{mid}) = \bar{\delta}_{act}, \quad \mathcal{U}'_1(\bar{\delta}_{mid}) = 0, \quad (\text{A.8.1})$$

$$\mathcal{U}_2(\bar{\delta}_{max}) = \bar{\delta}_{max}, \quad \mathcal{U}_2(\bar{\delta}_{mid}) = \bar{\delta}_{act}, \quad \mathcal{U}'_2(\bar{\delta}_{mid}) = 0, \quad (\text{A.8.2})$$

with $\bar{\delta}_{mid} = (\bar{\delta}_{max} - \bar{\delta}_{min})/2$. Then the adaptive value $\mathcal{U}(\bar{\delta})$ is:

$$\mathcal{U}(\bar{\delta}) = \begin{cases} \mathcal{U}_1(\bar{\delta}) & \bar{\delta} < (\bar{\delta}_{max} - \bar{\delta}_{mid}) \\ \mathcal{U}_2(\bar{\delta}) & \bar{\delta} \geq (\bar{\delta}_{max} - \bar{\delta}_{mid}) \end{cases}. \quad (\text{A.8.3})$$

The result of the proposed transformation yielding the adaptive discretization is shown in Figure A.8.

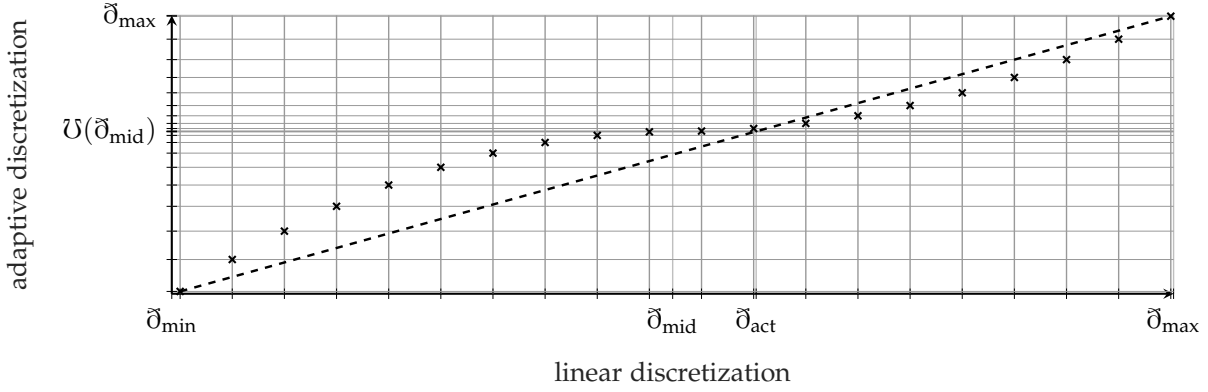


Figure A.8.: Nonlinear mapping from linear to adaptive discretization for actual value $\bar{\delta}_{\text{act}}$. The minor grid lines show the desired higher resolution around $\mathcal{U}(\bar{\delta}_{\text{mid}}) = \bar{\delta}_{\text{act}}$ for the adaptive discretization. Note, $\mathcal{U}(\bar{\delta})$ is changing with respect to $\bar{\delta}_{\text{act}}$, which yields the adaptive character.

A.9. Sequential Quadratic Programming Algorithm

The utilized approach given by Algorithm A.1 basically follows the *Line Search SQP Algorithm* as described in Nocedal and Wright (2006).

Algorithm A.1.: Sequential Quadratic Programming

Constraints and corresponding lagrange multipliers: $\hat{\mathbf{a}}, \hat{\boldsymbol{\lambda}}$

Jacobian of the lagrangian: $\hat{\mathbf{A}}(\hat{\mathbf{z}})^T = [\nabla \hat{a}_1(\hat{\mathbf{z}}), \nabla \hat{a}_2(\hat{\mathbf{z}}), \dots, \nabla \hat{a}_t(\hat{\mathbf{z}}), \dots, \nabla \hat{a}_\Xi(\hat{\mathbf{z}})]$

1: **function** LINE SEARCH SQP ALGORITHM

2: Choose parameters $\hat{\eta} \in [0, 0.5]$, $\hat{\tau} \in [0, 1]$ and initial values $\hat{\mathbf{p}}_0, \hat{\boldsymbol{\lambda}}_0$

3: Evaluate $\hat{F}(\hat{\mathbf{z}}_0)$, $\nabla \hat{F}(\hat{\mathbf{z}}_0)$, $\hat{\mathbf{a}}(\hat{\mathbf{z}}_0)$, $\hat{\mathbf{A}}(\hat{\mathbf{z}}_0)$

4: Choose initial symmetric positive definite hessian approximation $\hat{\mathbf{B}}_0 \in \mathbb{R}^{n_{\hat{\mathbf{B}}} \times n_{\hat{\mathbf{B}}}}$

5: **repeat**

6: Compute $\hat{\mathbf{p}}_i$ by solving the quadratic programming problem 7.4.3

7: Choose $\hat{\mu}_i$ to satisfy $\hat{\mu} > \|\hat{\boldsymbol{\lambda}}_{i+1}\|_\infty$ so that $\hat{\mathbf{p}}_i$ is descent direction for $\hat{\phi}$

8: $\hat{\alpha}_i \leftarrow 1$

9: **while** $\hat{\phi}(\hat{\mathbf{z}}_i + \hat{\alpha}_i \hat{\mathbf{p}}_i, \hat{\mu}_i) > \hat{\phi}(\hat{\mathbf{z}}_i, \hat{\mu}_i) + \hat{\eta} \hat{\alpha}_i \hat{D}(\hat{\phi}(\hat{\mathbf{z}}_i, \hat{\mu}_i), \hat{\mathbf{p}}_i)$ **do**

10: Reset $\hat{\alpha}_i \leftarrow \hat{\tau}_{\hat{\alpha}} \hat{\alpha}_i$ for $\hat{\tau}_{\hat{\alpha}} \in [0, \hat{\tau}]$

11: **end while**

12: Set $\hat{\mathbf{z}}_{i+1} \leftarrow \hat{\mathbf{z}}_i + \hat{\alpha}_i \hat{\mathbf{p}}_i$

13: Evaluate $\hat{F}(\hat{\mathbf{z}}_{i+1})$, $\nabla \hat{F}(\hat{\mathbf{z}}_{i+1})$, $\hat{\mathbf{a}}(\hat{\mathbf{z}}_{i+1})$, $\hat{\mathbf{A}}(\hat{\mathbf{z}}_{i+1})$

14: Set $\hat{\mathbf{s}}_i \leftarrow \hat{\mathbf{z}}_{i+1} - \hat{\mathbf{z}}_i$ and $\hat{\mathbf{y}}_i \leftarrow \nabla_{\hat{\mathbf{z}}} \mathcal{L}(\hat{\mathbf{z}}_{i+1}, \hat{\boldsymbol{\lambda}}_{i+1}) - \nabla_{\hat{\mathbf{z}}} \mathcal{L}(\hat{\mathbf{z}}_i, \hat{\boldsymbol{\lambda}}_{i+1})$

15: Calculate $\hat{\mathbf{B}}_{i+1}$ by updating $\hat{\mathbf{B}}_i$ with the BFGS update formula 7.4.8

16: **until** convergence test is satisfied or maximum number of iterations reached

17: **end function**

The algorithm addresses the aspects of step calculation by quadratic programming, backtracking line search using an ℓ_1 merit function and iterative updating of the hessian matrix as discussed in section 7.4.2.

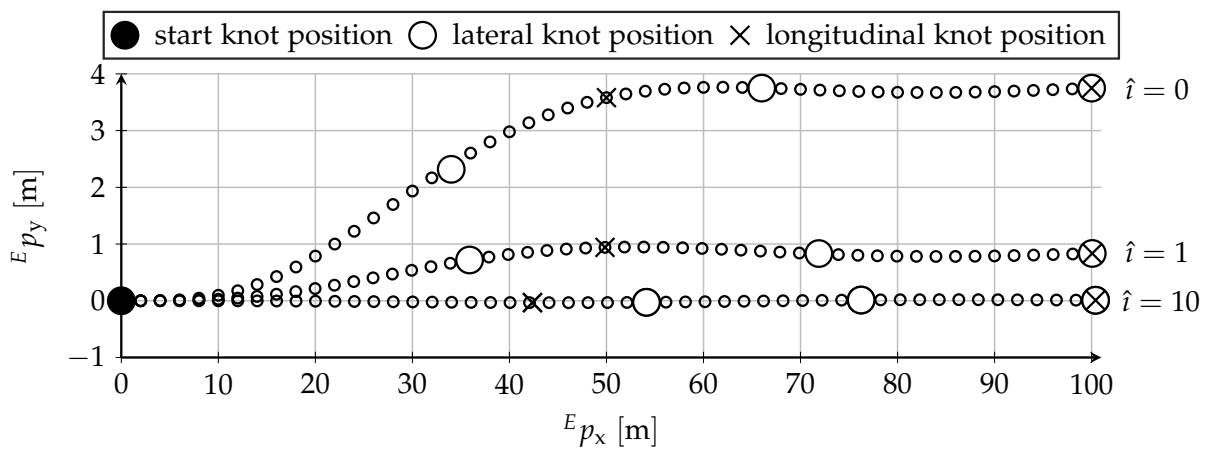
A.10. Convergence Test and Runtime Assessment for Continuous Approach

To get an impression of the general functioning the convergence of the continuous trajectory optimization approach is tested by way of experiment. Since the optimal trajectory is generally not known as it depends on subjective assessment the trajectory optimization problem is reversed for the simple case of driving straight ahead without any dynamic obstacles. For this example the optimal solution is known beforehand. A test case *Si-C-1* as shown in Table A.3 is created, in which the initial trajectory is a lane change to the left. Figure A.9 shows the convergence behavior of the continuous

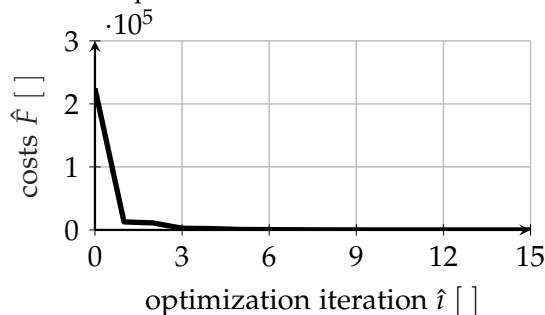
Table A.3.: Test case definition *Si-C-1*.

ID []	$F p_x$ [m]	$F p_y$ [m]	$F v$ [km/h]	$F \psi$ [rad]	target lane []	$F v_{des}$ [km/h]
ego	0	0	72	0	1	72

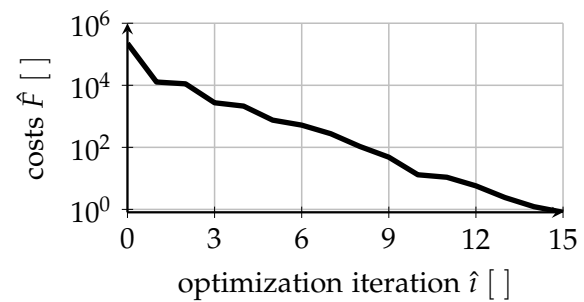
approach. The largest decrease in the objective function value can be observed in the first optimization iteration $\hat{i} = 1$. At round $\hat{i} = 6$ iterations the result is already very close to the final solution. The final result after $\hat{i} = 15$ iterations is depicted in



(a) Trajectory evolution with respect to optimization iterations exemplarily shown for initial trajectory $\hat{i} = 0$ and optimization iterations $\hat{i} = 1$ and $\hat{i} = 10$.



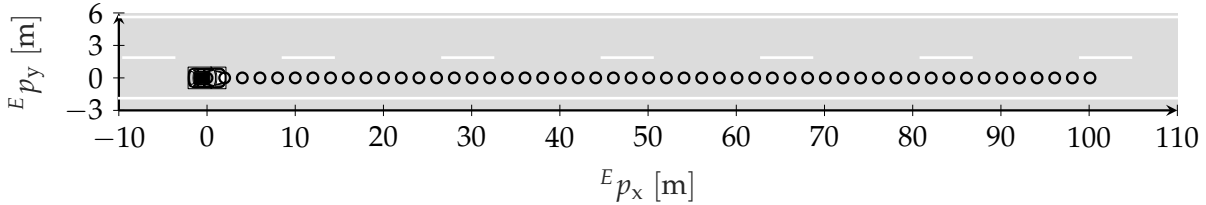
(b) Objective function value.



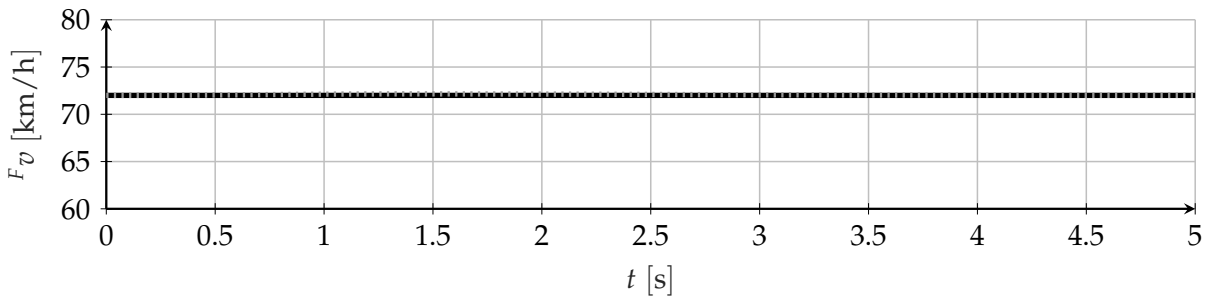
(c) Semi-log plot with costs on a logarithmic scale.

Figure A.9.: Convergence of the continuous approach in test case *Si-C-1*.

Figure A.10. As can be seen, the optimal trajectory converges to the expected straight line with constant velocity. This is no general proof of convergence, but at least an indication towards validity of the continuous variant of the two level hierarchical trajectory optimization framework.



(a) Final result after $\hat{i} = 15$ optimization iterations.



(b) Velocity of final result.

Figure A.10.: Final result of the continuous approach in test case *Si-C-1*.

A question that raises in this regard is related to the compromise between the quality of the solution and the run time taken to find the optimal solution. For the results shown in chapter 8 the chosen number of maximum iterations is set to 10, which is a rather conservative setup considering that the results of test case *Si-C-1* represent an artificial case. This allows to implement a more optimistic setup by reducing the maximum number of SQP optimization iterations to $\hat{i} = 5$ without suffering significant loss of performance. Additionally, with `time_limit = 0.0001` an OSQP parameter is added that permits to limit the run time of the underlying quadratic programming solver. Table A.4 shows the measurements taken on a single core of an Intel Core i5@3.30GHz with 6MB cache in a representative scenario including lead vehicle following and an overtaking sequence. It has to be noted that measurements are not taken in a real time environment, which is why given run times need to be rated carefully.

mean run time [ms]	standard deviation run time [ms]	mean SQP iterations []
75.197	43.757	2.8646

Table A.4.: Run time assessment.

With the chosen setup it can be seen that on average the optimizer converges even faster than the permitted number of five SQP iterations. However, the run time standard deviation shows significant potential for improvement as the run time of one single iteration should be decreased, instead of relying on a low number of iterations. On the positive side it can be concluded that mean run times of 75 ms are achievable.

A.11. Further Results

In this section supplemental results of the developed trajectory planning approaches are presented. Analogous to section 8.2 experiments are subdivided into situation-based and scenario-based tests in order to enable a thorough analysis with respect to diverse characteristics.

For situation-based testing a variant of the stop and a variant of the merge maneuver are shown. Moreover, for scenario-based testing sequences augmented with details on the in- and output of the trajectory planning approach are depicted, illustrating the temporal evolution and allowing an extended analysis of the trajectory planning results.

Situation-based testing

First, a variant of the stop test case is analyzed. Therefore, test case *Si-S-1* is changed in a way that a lateral motion of the ego vehicle is required to reach the target position. The definition for test case *Si-S-2* is given in Table A.5. In coherence to the defined

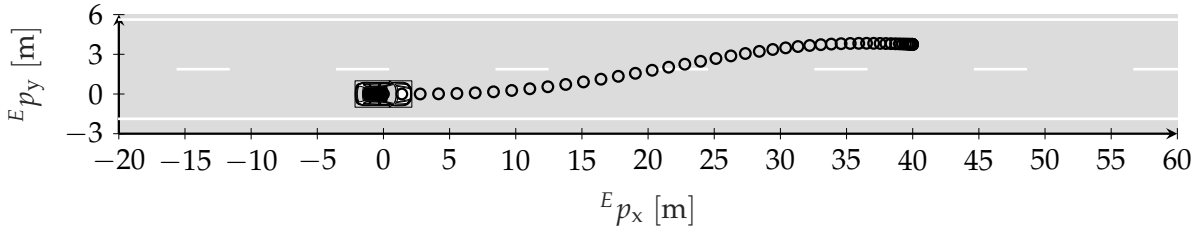
Table A.5.: Straight road test case definition *Si-S-2* for a stop maneuver.

ID []	${}^F p_x$ [m]	${}^F p_y$ [m]	${}^F v$ [km/h]	${}^F \psi$ [rad]	target lane []	${}^F p_{x,s}$ [m]	${}^F p_{y,s}$ [m]
ego	0	0	50	0	1	40	3.75

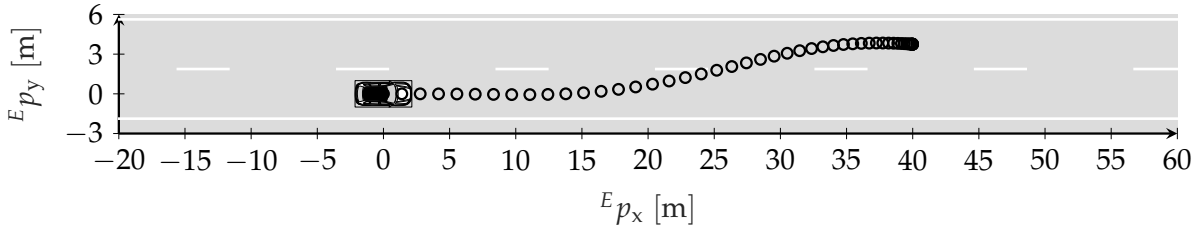
target lane the lateral stop position is set to the center of the left ego lane. The challenge is hence to find a valid solution, especially with regard to lateral vehicle dynamics at lower speeds in order to maintain the non-holonomic characteristics of the vehicle motion.

Figure A.11 shows the results for the continuous and the discrete approach in test case *Si-S-2* using configuration *Co-S*. Both, the continuous and the discrete approach succeed in realizing the demanded stop maneuver. The difference is mainly visible in the lateral motion as the continuous approach behaves smoothly and reaches the target lane earlier in time. The discrete approach realizes the stop maneuver at the expense of higher maximum lateral accelerations. In fact, the maximum absolute value of the lateral acceleration reached by the discrete approach is 3.54 m/s^2 at $t = 1.2 \text{ s}$, whereas for the continuous approach it is 2.71 m/s^2 at $t = 0.6 \text{ s}$. In comparison to the results of test case *Si-S-1* shown in Figure 8.1, the velocities of the continuous approach and discrete approach are basically the same. This is not surprising, as the longitudinal stop distance is the same and stop configuration *Co-S* leaves no significant freedom to attain further improvements with respect to the overall result.

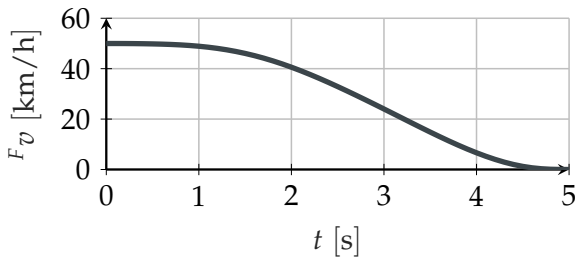
In the variant of the merge maneuver test case *Si-M-1* the initial velocity of the ego vehicle is increased to 120 km/h instead of driving at an initial speed of 100 km/h. The resulting test case definition for test case *Si-M-2* is summarized in Table A.6. Since the initial velocity is higher than the velocity of obstacle vehicles ID:1 and ID:2 the ego vehicle needs to decelerate to position itself in the gap on the left lane.



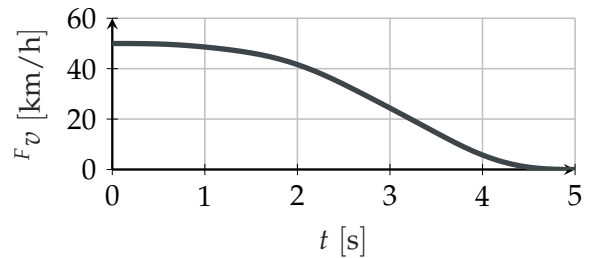
(a) Result of the continuous approach.



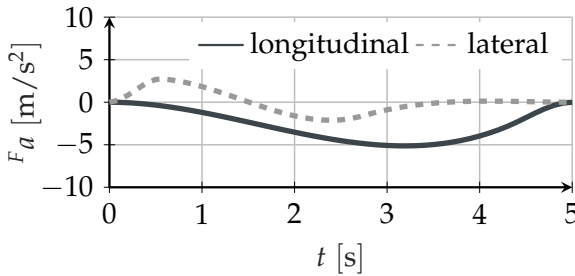
(b) Result of the discrete approach.



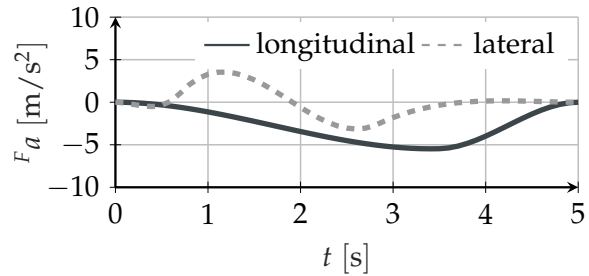
(c) Velocity of the continuous approach.



(d) Velocity of the discrete approach.



(e) Acceleration of the continuous approach.



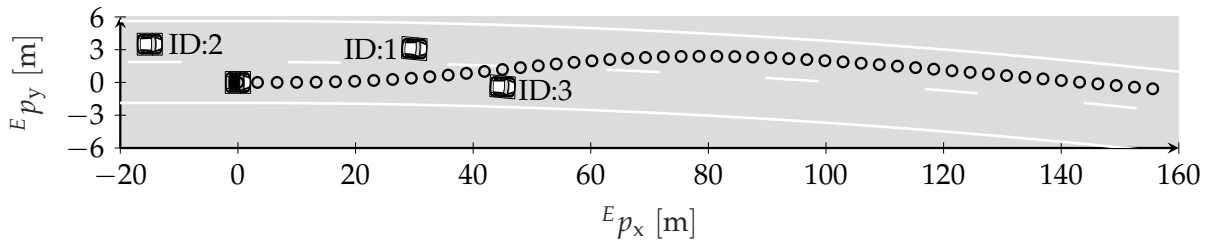
(f) Acceleration of the discrete approach.

Figure A.11.: Results in stop test case *Si-S-2*.

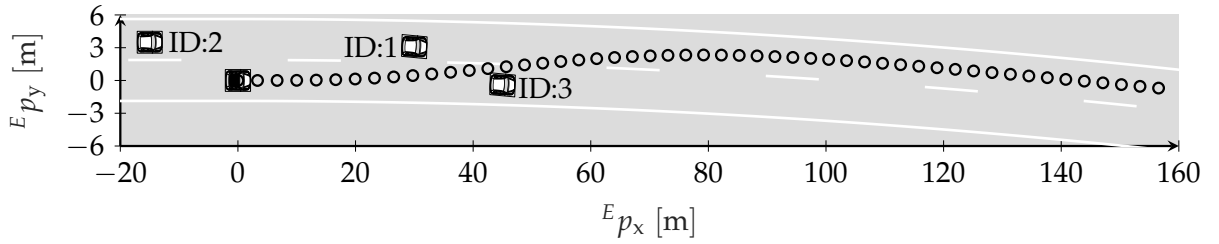
The corresponding results for test case *Si-M-2* using configuration *Co-D* are shown in Figure A.12. The results show that the continuous as well as the discrete approach succeed in realizing a lane change to the left in a right curve, while maintaining proper safety distances to lead vehicle ID:1 and tail vehicle ID:2. The latter is indicated by the decrease of the ego velocity, showing that the ego vehicle adapts to the speed of obstacle vehicles ID:1 and ID:2 that form the gap for the merge-in maneuver. In contrast to the results of test case *Si-M-1* the lateral acceleration shows that the planned lateral motion of the continuous and the discrete approach is nearly identical. The continuous approach decelerates faster to adapt to the speed of 110 km/h of obstacle vehicles ID:1 and ID:2 and in comparison the longitudinal velocity of the discrete approach between $t = 4$ s and $t = 5$ s is higher than the one of the continuous approach. However, with respect to the overall result the effective difference in longitudinal position is neglectable.

Table A.6.: Right curve test case definition *Si-M-2* for a merge maneuver.

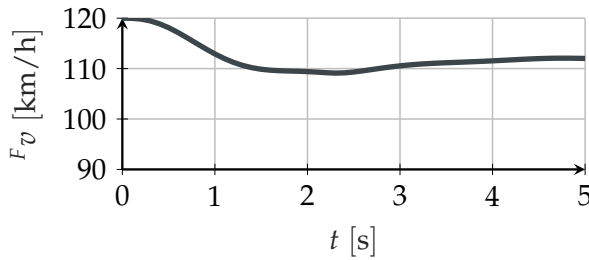
ID []	$F p_x$ [m]	$F p_y$ [m]	$F v$ [km/h]	$F \psi$ [rad]	target lane []	$F v_{des}$ [km/h]
ego	0	0	120	0	1	130
1	30	3.10	110	-0.05	1	110
2	-15	3.50	110	0	1	110
3	45	-0.40	85	-0.05	2	85



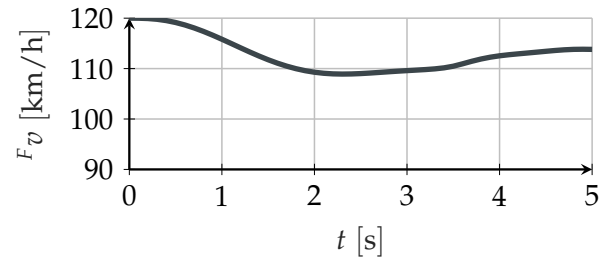
(a) Result of the continuous approach.



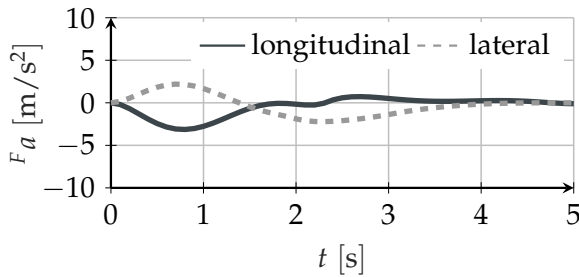
(b) Result of the discrete approach.



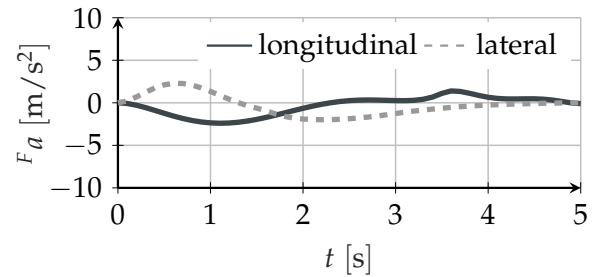
(c) Velocity of the continuous approach.



(d) Velocity of the discrete approach.



(e) Acceleration of the continuous approach.



(f) Acceleration of the discrete approach.

Figure A.12.: Results in merge test case *Si-M-2*.

Scenario-based testing

For scenario-based testing some more figures are shown with respect to highway test case *Sc-H-1* to give a comprehensive understanding of the temporal progress and a more detailed analysis of the particular characteristics that can be observed.

Especially the impact of perception with respect to accuracy and completeness of the generated environment model should be investigated. Since sensor data is prone to various effects it is expected that imperfect perception results mitigate the overall performance of the trajectory planning approach. The quality of lane marker detection by the camera is directly connected to the generation of the road model used to plan the trajectory. Limited view ranges and occlusion are the basic reasons why inaccuracies in that respect might occur. These inaccuracies are very likely to increase with higher distances. In combination with the relatively high planning horizon of $T_p = 5$ s the issue with respect to the necessity of lane marker extrapolation is evident. For this reason some measures are taken to extend the view range of detected lane boundaries, by assuming for example parallelism of respective lane boundaries and performing a circular extrapolation based on the detected curvature. In the same sense the object detection results will largely impact the planned trajectory with respect to desired distance behavior and collision avoidance. Figure A.13 shows the legend that accounts for the visualization of aforementioned aspects related to the environment. The object

Environment	
◆	object track position
—	camera - road boundary
—	road model - road boundary
⋯	camera - dashed lane marker
⋯	road model - dashed lane marker
Ego Vehicle	
⋯⋯⋯	ego vehicle trace
○	lateral knot position
×	longitudinal knot position

Figure A.13.: Legend for highway scenario analysis.

track position originates from fused radar and camera data. It is important to mention that the accuracy of each track improves over time, such that especially in the initial phase of detection larger deviations to the ground truth position of the object might occur. In addition, the object track position is dominated by the radar reflection point and lacks camera support for vehicles driving in the rear (as already mentioned in section 8.2 the simulated ego vehicle is equipped with a front facing camera only). However, all relevant lane data can be extracted for the purpose of trajectory planning. It is primarily distinguished between road boundaries and dashed lane markers. Whereas the former represents all lane marker types, which do not allow the ego vehicle to pass, the latter denotes all lane marker types that allow the ego vehicle to cross. For a deeper understanding of the trajectory planning result it is moreover vital to investigate the processed input related to the static environment. Hence, the road model that is composed based on the camera data is visualized. This enables an informative comparison between camera and road model data and contributes to a better understanding of the planning results. In the closed-loop simulation environment the realized path that corresponds to the ego vehicle position of all previous

simulation cycles is given in terms of the ego vehicle trace. The characteristics of the planned trajectory in terms of knot positions of the lateral and the longitudinal spline are visualized as well. In the following results of highway test case *Sc-H-1* will first be discussed for the continuous approach. Afterwards, the results for the discrete approach will be shown.

The highway entering is one of the most challenging maneuvers with respect to highway driving, because the ending lane forces the ego vehicle to make a lane change. As depicted in Figure A.14 the ego vehicle accelerates to merge in front of obstacle vehicle ID:7. In addition the ego vehicle should keep track of obstacle vehicle ID:10, which performs a lane change to the ego target lane and thus becomes the new lead vehicle.

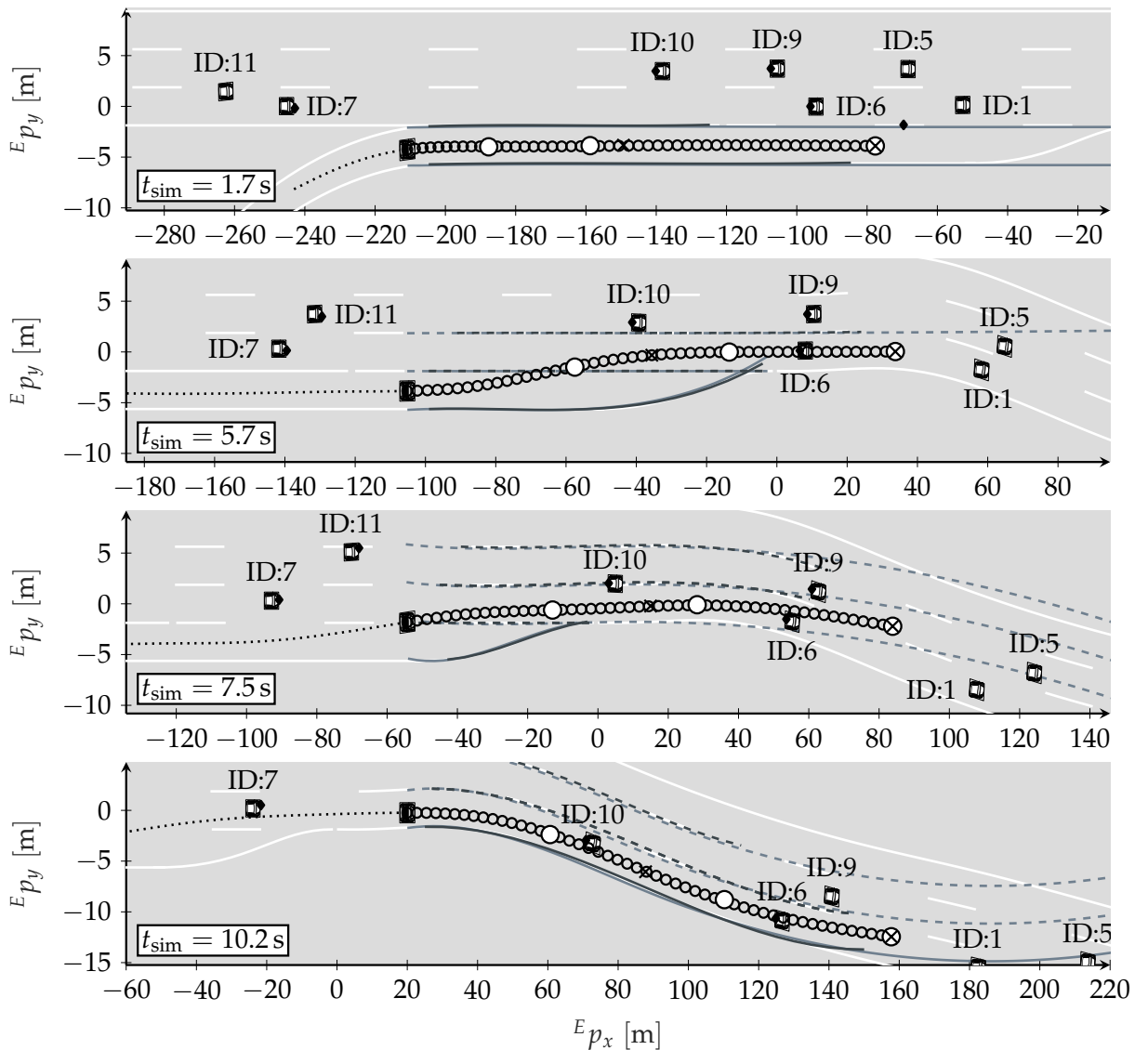


Figure A.14.: Result of the continuous approach for selected time instances, showing the entering of the highway.

The problem of perception data is in particular revealed in this part of the highway maneuver as changing curvature and a limited view range contribute to a large mismatch

of the road model with respect to the ground truth lane markers. As a consequence spline knots that might seem to be placed inaccurately with respect to the lateral position, show to be accurately positioned with respect to the applicable road model. A misdetection of an object can be observed at simulation time $t_{\text{sim}} = 2.1$ s leading to a ghost object at position ${}^E \mathbf{p} = [-69.60 \ -1.82]^T$ m. The reason for this is an inaccurate object track of obstacle vehicle ID:5. While first the accuracy improves over time, the related object track completely vanishes after a few more simulation cycles. Thus, from planning perspective the observed issue has no substantial effect.

Once the ego vehicle has entered the highway the increased set speed of 120 km/h after passing ${}^E p_x = 100$ m is leading to an overtaking maneuver. The corresponding sequence that shows the lane change to the left in order to overtake obstacle vehicle ID:10 is depicted in Figure A.15.

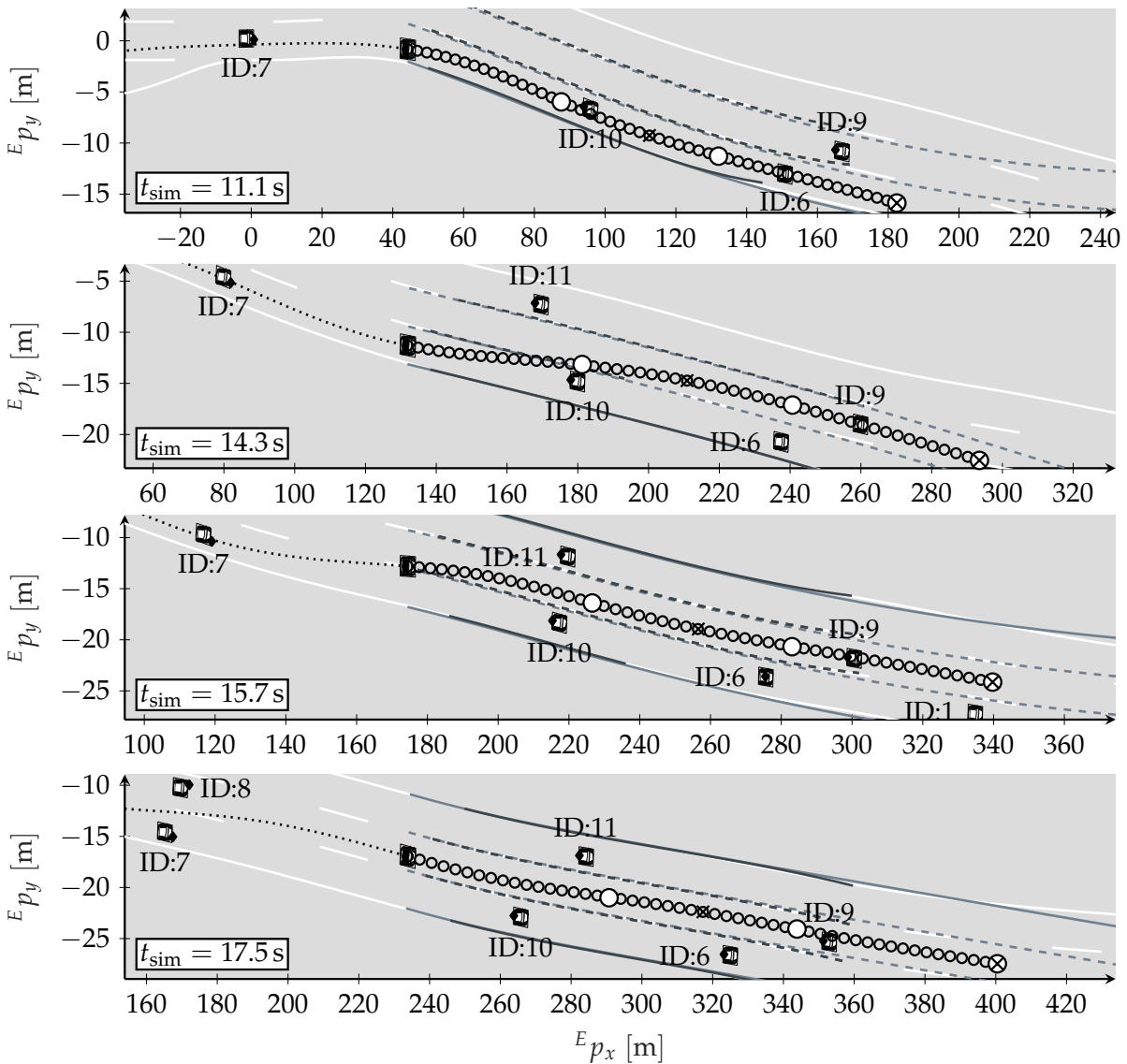


Figure A.15.: Result of the continuous approach for selected time instances, showing the lane change to start overtaking of obstacle vehicle ID:10.

After passing $E p_x = 300$ m the desired set speed is updated to 140 km/h. Although obstacle vehicle ID:9 in the ego lane is driving at a slower speed, the ego vehicle is not changing its lane immediately. This is because vehicle ID:8 is approaching from behind, narrowing the gap between obstacle vehicle ID:8 and ID:11 on the left ego lane. Only at the time, when obstacle vehicle ID:8 has adapted its velocity to approximately 130 km/h while keeping a constant distance to obstacle vehicle ID:11, the ego vehicle starts to perform a single lane change to the left. The merge-in maneuver of the ego vehicle is depicted in Figure A.16.

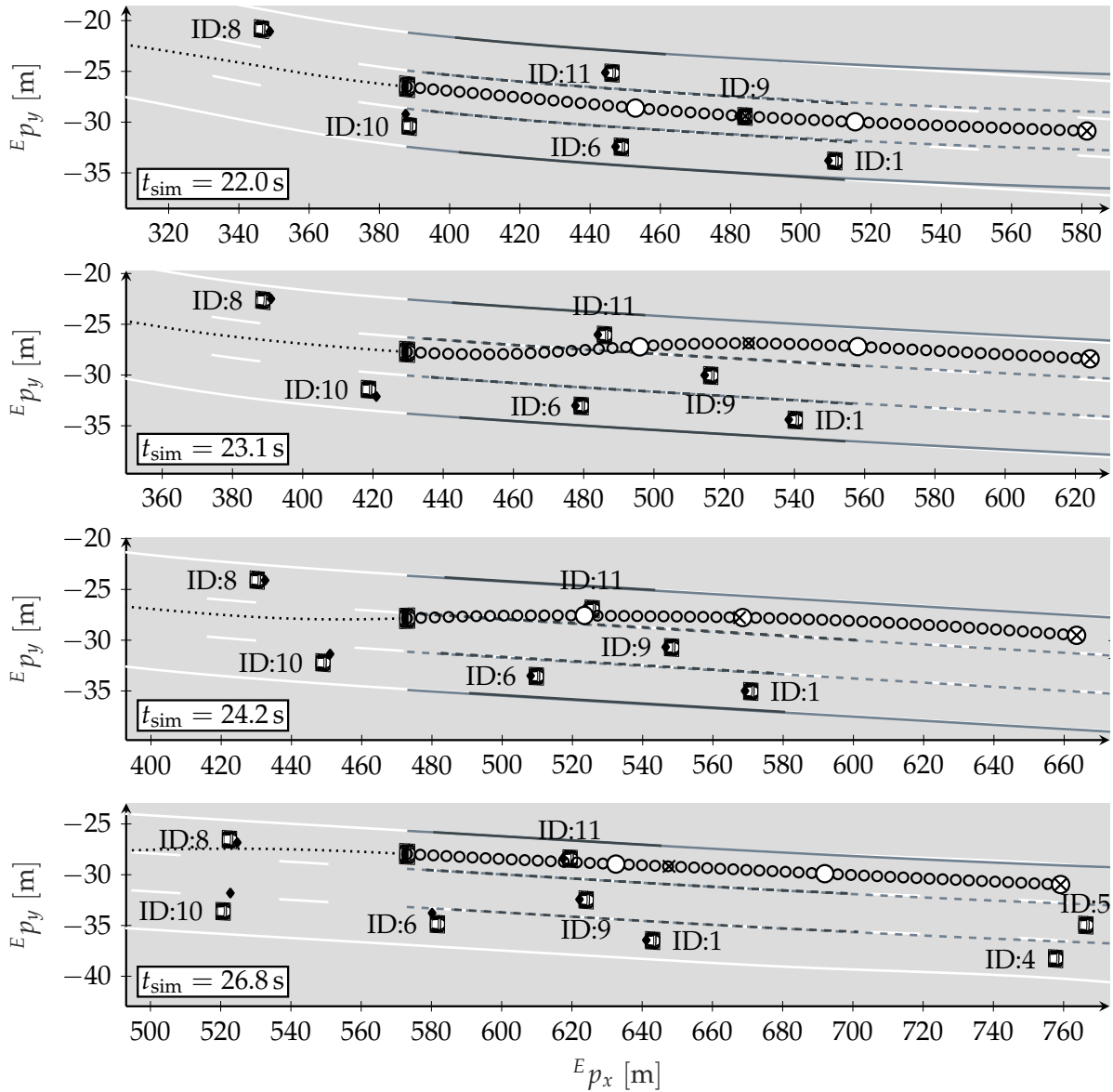


Figure A.16.: Result of the continuous approach for selected time instances, showing the lane change to start overtaking of obstacle vehicle ID:9.

The ego vehicle then follows obstacle vehicle ID:11, maintaining a constant time gap. As soon as obstacle vehicle ID:11 does a lane change to the right, the ego vehicle can now accelerate to the desired set speed. Restrictions imposed by distance keeping with

respect to preceding vehicles are no longer applicable as there is no impeding lead vehicle. This finally leads to an overtaking maneuver, which is shown in Figure A.17. During the maneuver, obstacle vehicle ID:11 performs a second lane change to the right in order to drive onto the rightmost lane. Subsequently, to complete the overtaking of obstacle vehicle ID:11 the ego vehicle performs a single lane change to the right.

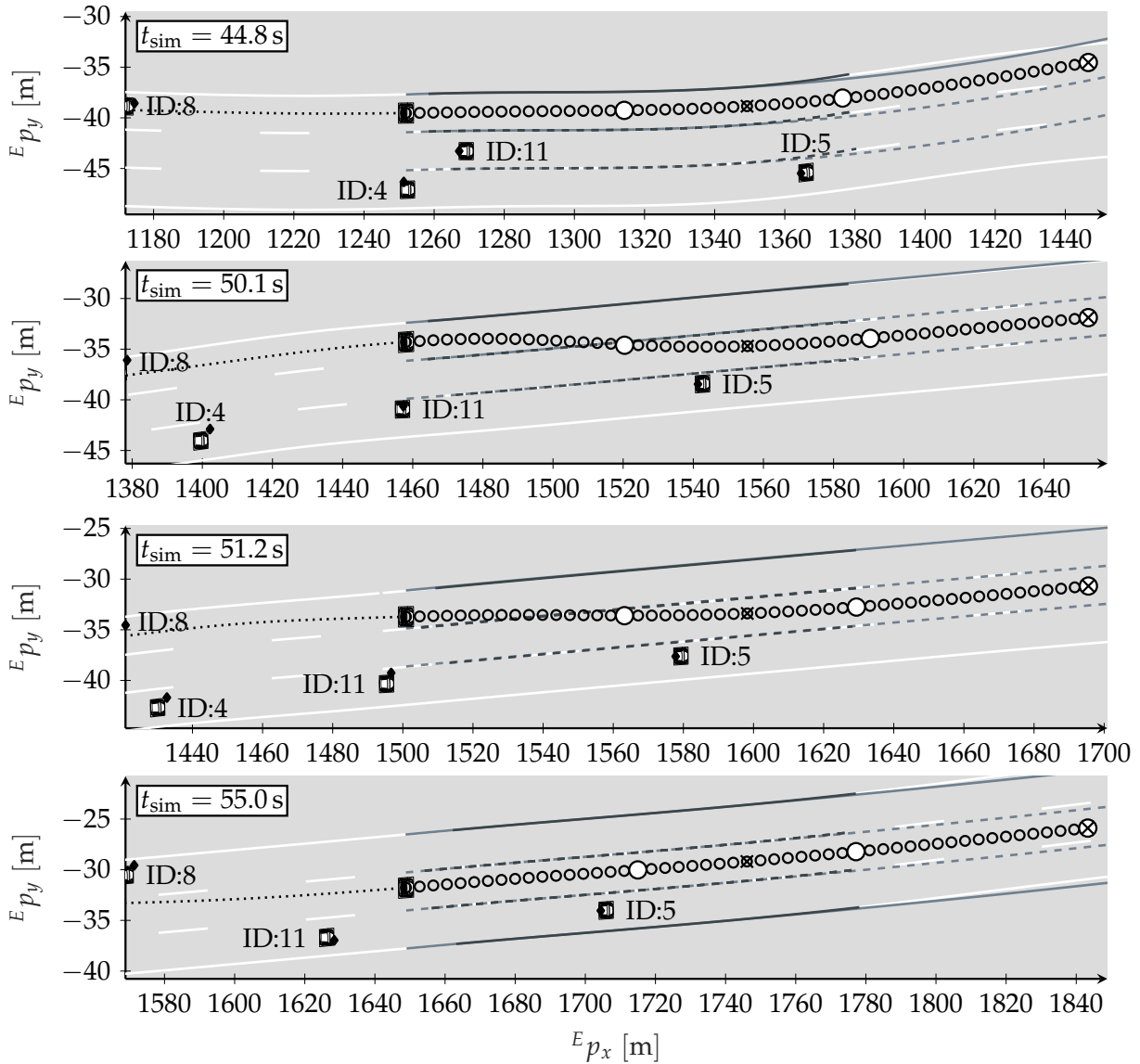


Figure A.17.: Result of the continuous approach for selected time instances, showing the overtaking of obstacle vehicle ID:11.

In the last part of the highway scenario, shown in Figure A.18, the ego vehicle overtakes obstacle vehicle ID:5. After the ego vehicle passes obstacle vehicle ID:5, the ego vehicle performs a single lane change to the right. Since lanes are more or less straight and view ranges are comparably high the road model is accurately matching the course of the road.

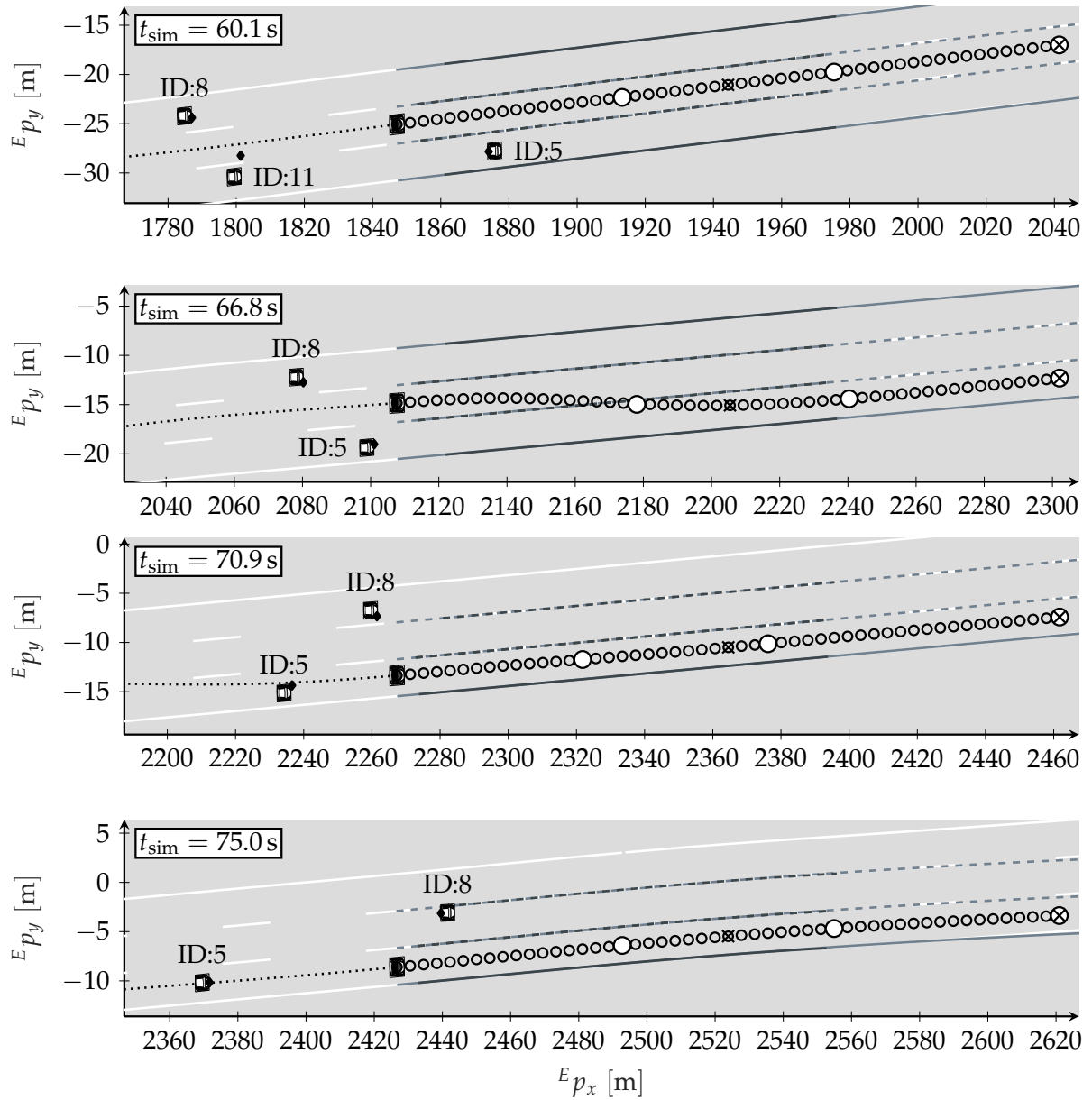


Figure A.18.: Result of the continuous approach for selected time instances, showing the overtaking of obstacle vehicle ID:5.

For the discrete approach observed characteristics in terms of the perception results apply analogously to the continuous approach. It should be mentioned that because of the different planning solutions regarding the continuous and the discrete trajectory planning approach, the two simulation runs are similar, but not exactly equal to each other.

In general, as will be shown hereinafter, the applied sampling approach proves as a convenient strategy for driving on curved roads. The curvilinear transform of sampled breakpoints into vehicle coordinates with subsequent interpolation enables a suitable approach to lateral positioning with respect to the road model, being independent of road curvature.

The highway entry as a result of the discrete approach is shown in Figure A.19.

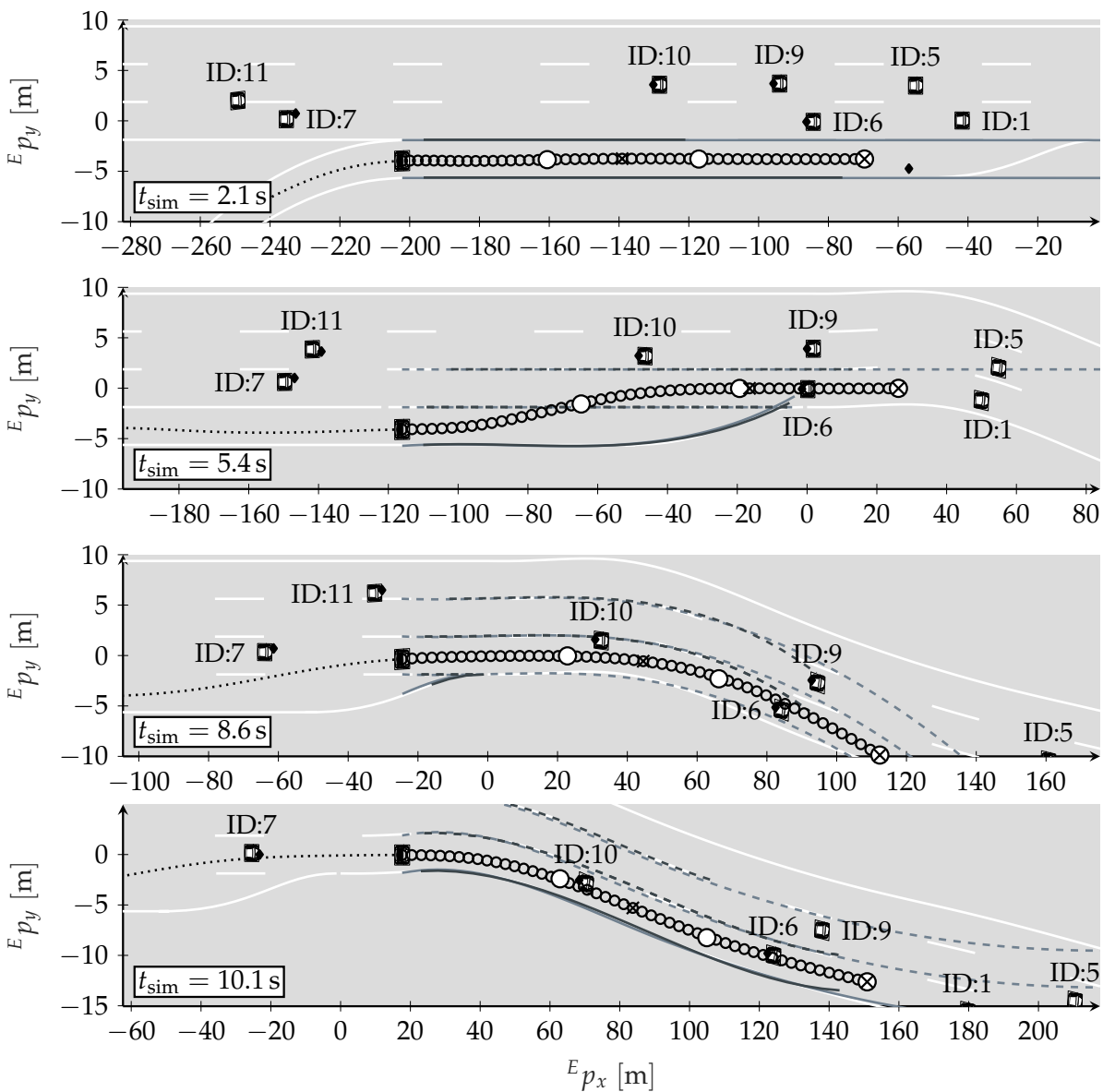


Figure A.19.: Result of the discrete approach for selected time instances, showing the highway entry of the ego vehicle.

At the time the ego vehicle intends to enter the highway obstacle vehicle ID:10 needs to be considered that likewise performs a lane change to the ego target lane, becoming the lead vehicle at around $t_{\text{sim}} = 8.6$ s.

In the next sequence depicted in Figure A.20 the ego vehicle overtakes obstacle vehicle ID:10. Therefore, the ego vehicle accelerates and performs a single lane change to the left. This is because of the increased set speed of 120 km/h that is applicable after $E p_x = 100$ m has been passed. At that time the ego vehicle set speed is higher than the current velocity of obstacle vehicle ID:10 that drives at a speed of 100 km/h. As can be seen at $t_{\text{sim}} = 14.0$ s the view range prevents from detecting the correct curvature (especially in terms of the correct sign, i.e. direction), leading to a deviation in the road model and the real course of the road. However, in this case the impact is quite low as the model is still quite accurate in the range of the planned trajectory.

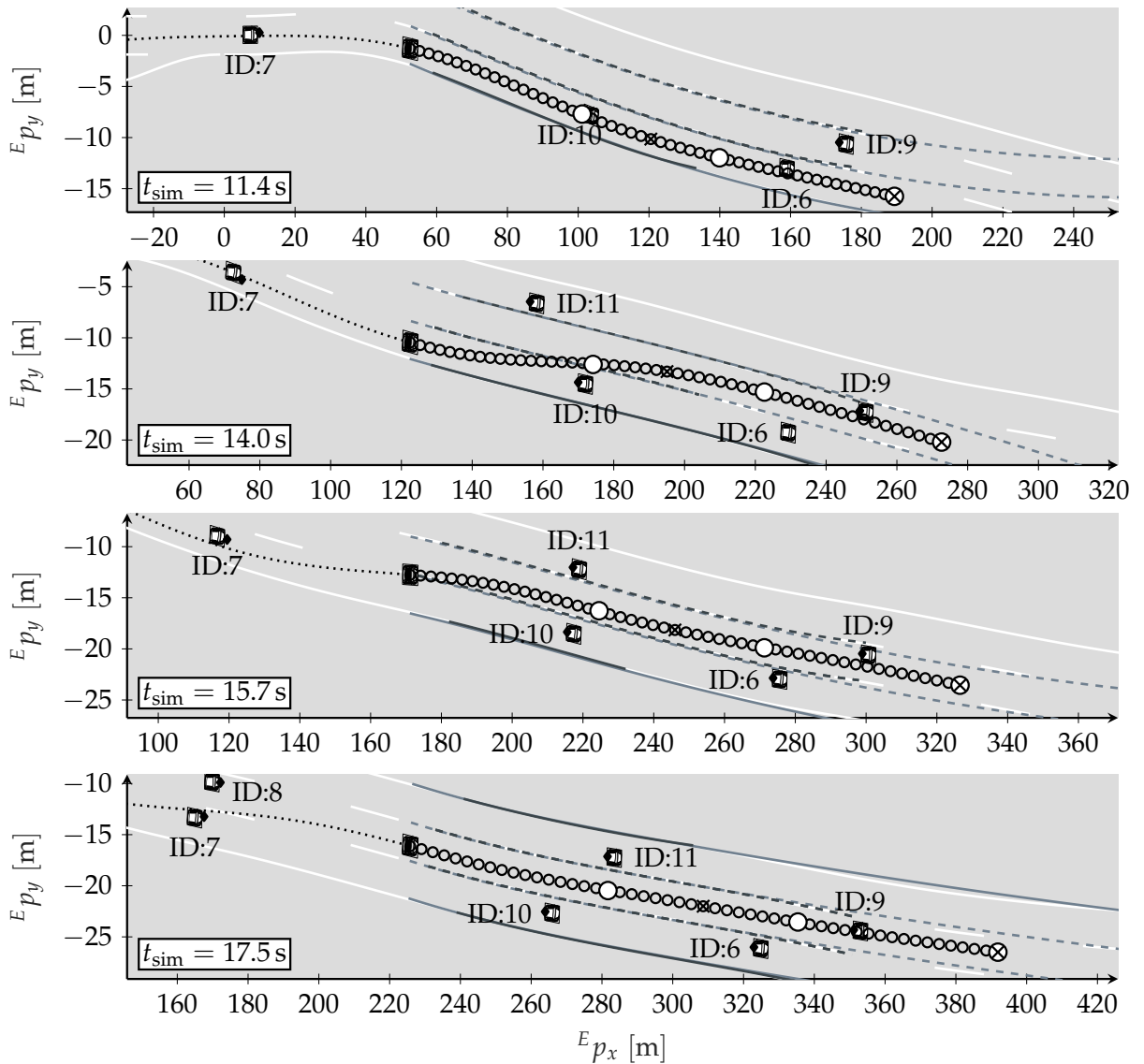


Figure A.20.: Result of the discrete approach for selected time instances, showing the lane change to start overtaking of obstacle vehicle ID:10.

In Figure A.21 the ego vehicle is surrounded by several obstacle vehicles. With desired set speed of 140 km/h and slower lead vehicle ID:9 the ego vehicle has to perform a lane change to the left to start the overtaking maneuver. In the considered case this means that the ego vehicle needs to merge into the gap between obstacle vehicle ID:8 and ID:11. After obstacle vehicle ID:9 has been passed, the ego vehicle follows obstacle vehicle ID:11 on the leftmost lane.

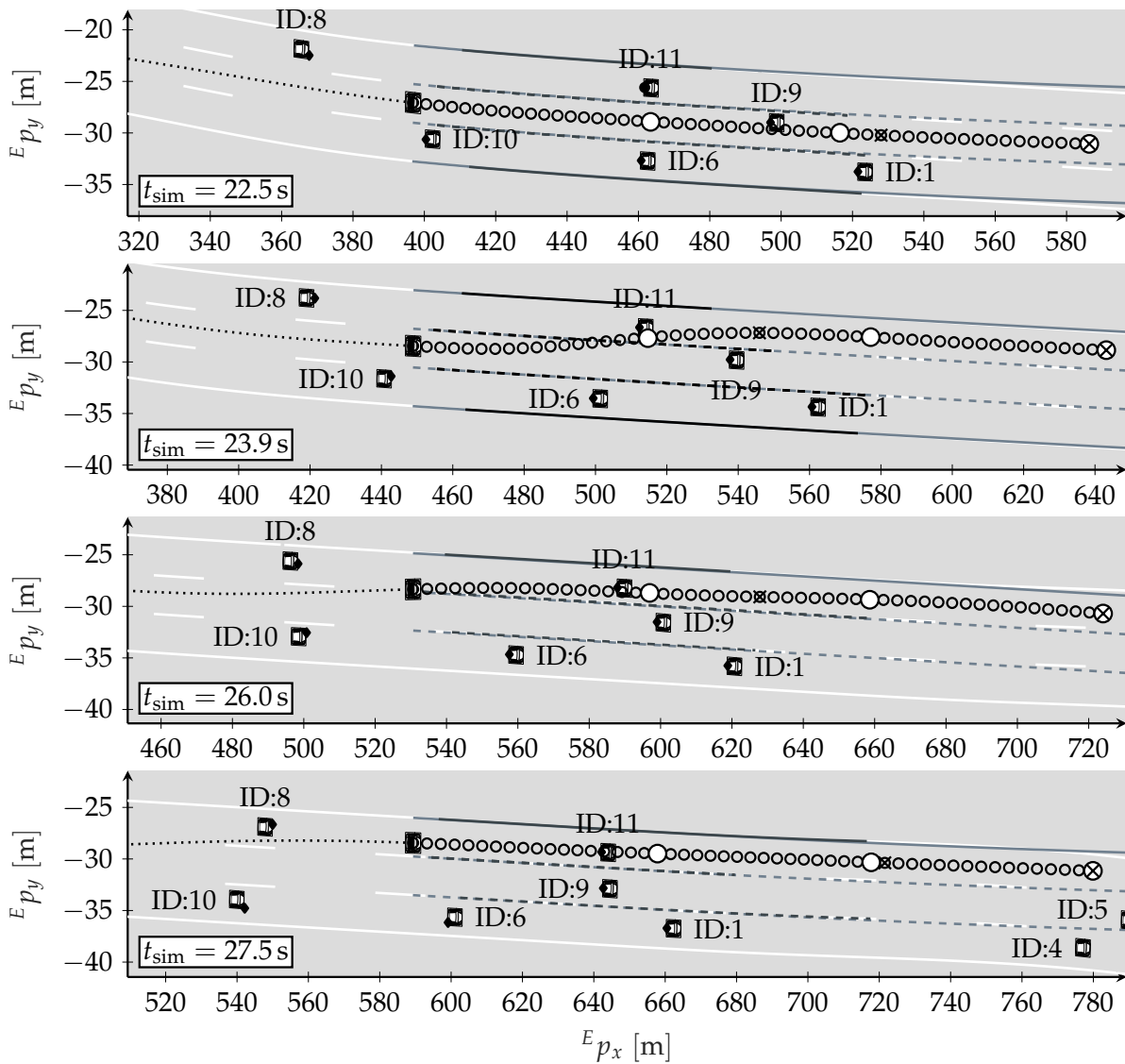


Figure A.21.: Result of the discrete approach for selected time instances, showing the lane change to start overtaking of obstacle vehicle ID:9.

Figure A.22 shows the performed overtaking maneuver of the ego vehicle, once that obstacle vehicle ID:11 has left the current ego lane. Since at this time there is no relevant lead vehicle anymore, the ego vehicle can pursue the aim to reach the set speed of 140 km/h. Obstacle vehicle ID:8 follows the ego vehicle and intends to overtake obstacle vehicle ID:11 as well.

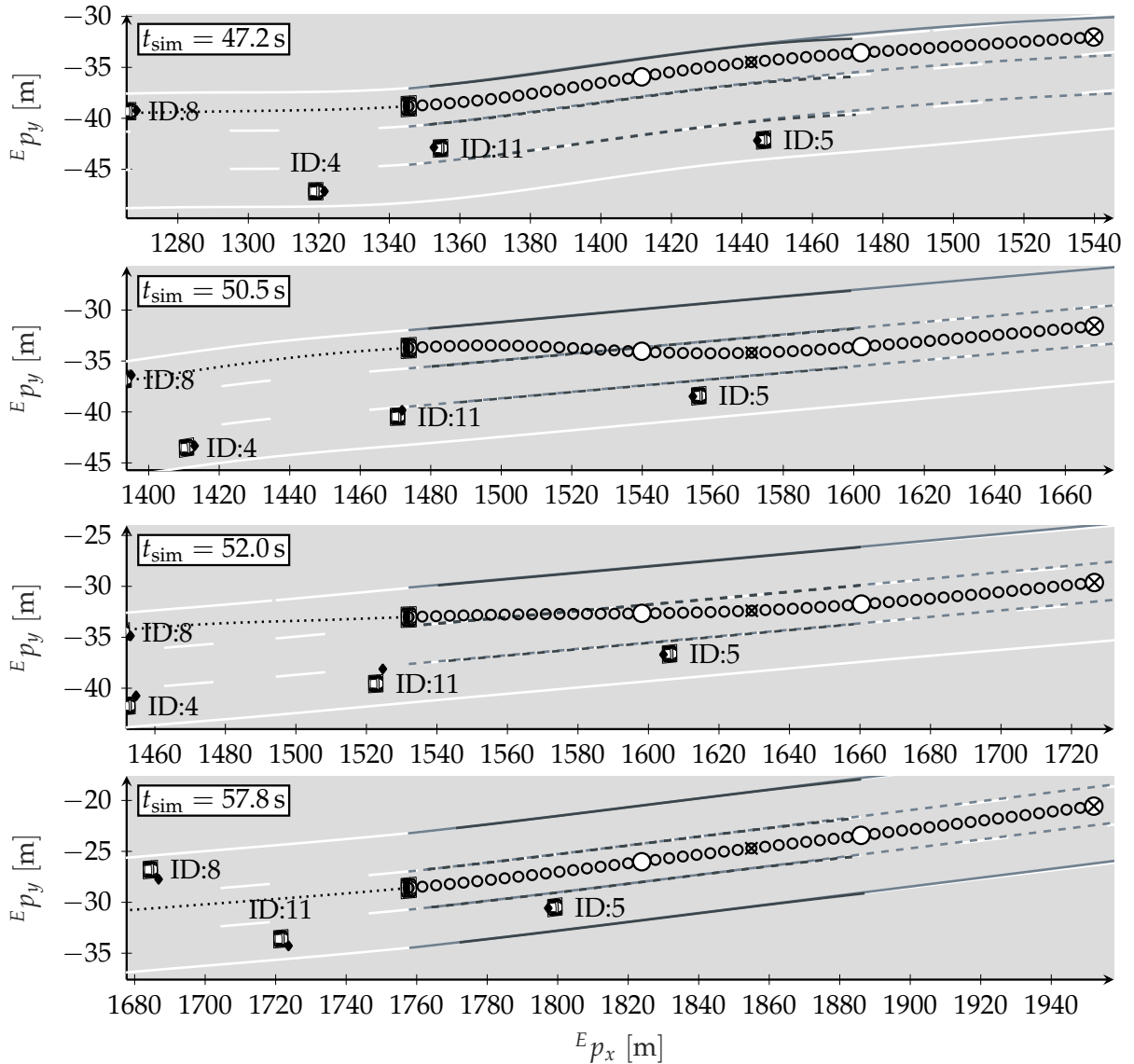


Figure A.22.: Result of the discrete approach for selected time instances, showing the single lane change to finish the overtaking of obstacle vehicle ID:11.

In the final sequence, depicted in Figure A.23, the ego vehicle overtakes obstacle vehicle ID:5 and drives to the rightmost lane by doing a lane change to the right. From the results it can be concluded that the continuous as well as the discrete approach are suitable for highway driving. The chosen simulation environment permits to draw a conclusive picture of the expected performance in real world applications. This addresses in particular the observed impact of inaccurate perception results. Due to the long planning horizon the trajectory goes beyond the view range of the perceived lane markers in almost every case. Still, it can be shown that the extrapolated road model mitigates the impact on the trajectory planning result, taking into account that especially in the close proximity of the ego vehicle perception results are adequately accurate. Having a closer look on the object tracking result, it can be found that especially in the back of the ego vehicle inaccuracies appear to a larger extend. However, for

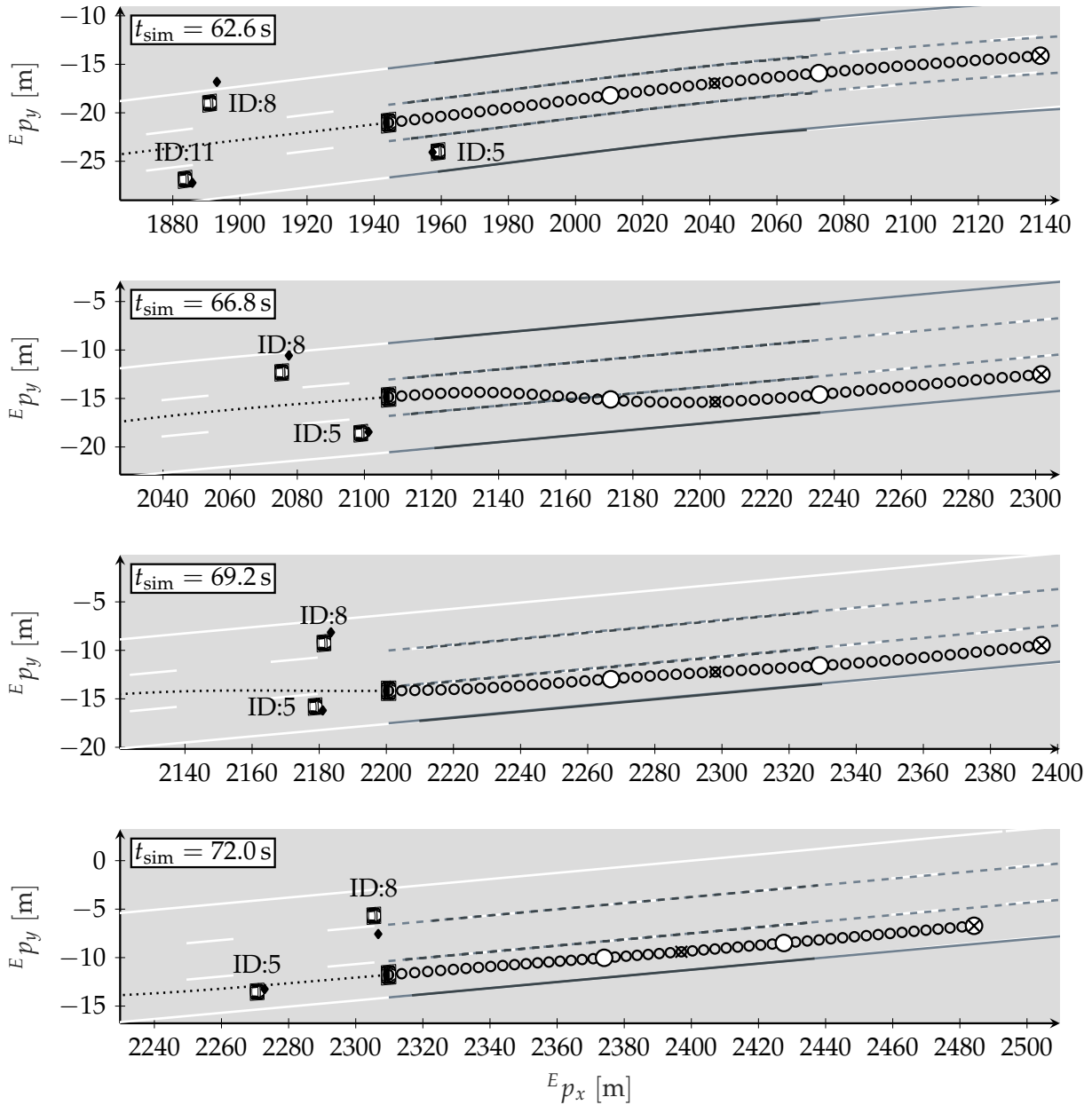


Figure A.23.: Result of the discrete approach for selected time instances, showing the lane change to finish the overtaking of obstacle vehicle ID:5.

ego trajectory planning the distance keeping objective reflects the desired behavior in terms of safe driving. It ensures to prevent dangerous situations in advance, effectively compensating for minor perception deviations. The receding horizon characteristic certainly contributes to react appropriately on unforeseen and even incorrectly predicted events. As can be seen from the scenario-based tests, when it comes to driving in dynamic environments combined with an imperfect perception the latter is a valuable characteristic of the developed approach.

A.12. Settings and Configuration

The high level optimization parameters as shown in Table A.7 summarize all relevant setting options for high level optimization, i.e. objectives and inequalities. They hence account for the setup of weights and constraint values manipulating the desired behavior of the vehicle.

Table A.7.: High level optimization parameters.

description	variable	value	unit
distance keeping weight	ω_d	5000	[]
reference velocity weight	ω_v	10	[]
reference lateral position weight	ω_p	500	[]
maximum acceleration weight	ω_c	5000	[]
constant time gap to lead vehicle	T_l	1.0	[s]
constant time gap to tail vehicle	T_t	0.5	[s]
minimum safety distance	d_{\min}	3	[m]
comfort acceleration	a_{comf}	1.5	[m/s ²]
comfort longitudinal acceleration limit	\check{a}_x	3.5	[m/s ²]
comfort lateral acceleration limit	\check{a}_y	2.5	[m/s ²]
minimum breakpoint time difference	T_m	0.5	[s]
maximum longitudinal acceleration	a_{\max}	see Figure 7.2	[m/s ²]
maximum steering angle	δ_{\max}	0.64	[rad]
maximum total acceleration	a_g	9.0	[m/s ²]

In order to obtain an accurate description of the vehicle motion, vehicle model parameters need to be chosen appropriately. The parameter values as given in Table A.8 are based on the *Audi A3 Sportback 2009*.

Table A.8.: Vehicle parameters.

description	variable	value	unit
ego vehicle length	l_{ego}	4.292	[m]
ego vehicle width	w_{ego}	1.995	[m]
inter-axle distance	l_a	2.5780	[m]
distance of CoG to front axle	l_v	1.057	[m]
distance of CoG to rear axle	l_h	1.521	[m]
vehicle mass	m	1400	[kg]
cornering stiffness front	c_v	117 800	[N/rad]
cornering stiffness rear	c_h	127 900	[N/rad]
characteristic velocity	v_{ch}	31.9604	[m/s]

The solver settings relate to the utilized underlying nonlinear constrained optimization approaches. Since the optimal interpolation relies on the result of the quadratic

programming solver it is vital for the continuous as well as for the discrete trajectory optimization approach. On top of that, for the continuous trajectory optimization approach, the parameters of the SQP algorithm need to be set. Because of the basic relevance of the constrained optimization algorithms for the developed trajectory optimization approaches, these parameters have a decisive impact on the overall result. The OSQP solver settings as shown in Table A.9 are taken from Stellato et al. (2020) and hence represent the default values for the solver.

Table A.9.: OSQP solver settings.

description	parameter name	value
ADMM overrelaxation parameter	α	1.6
ADMM rho step	ρ	0.1
ADMM sigma step	σ	$1 \cdot 10^{-6}$
primal infeasibility tolerance	eps_prim_inf	$1 \cdot 10^{-4}$
dual infeasibility tolerance	eps_dual_inf	$1 \cdot 10^{-4}$
relative tolerance	eps_rel	$1 \cdot 10^{-3}$
absolute tolerance	eps_abs	$1 \cdot 10^{-3}$
maximum number of iterations	max_iter	4000

The SQP solver setting is given in Table A.10 providing corresponding line search parameters as well as tolerance values for the definition of reasonable termination criteria.

Table A.10.: SQP solver settings.

description	variable	value
relative tolerance		$1.49 \cdot 10^{-8}$
absolute tolerance		$1.49 \cdot 10^{-8}$
line search acceptance factor	$\hat{\eta}$	0.25
line search step size factor maximum	$\hat{\tau}$	0.5
line search step size factor	$\hat{\tau}_{\hat{\alpha}}$	0.45